NEC

Preliminary User's Manual

μPD789835 Subseries

8-Bit Single-Chip Microcontrollers

μPD789832
μPD789833
μPD789834
μPD789835
μPD78F9835

Document No. U15559EJ1V1UD00 (1st edition)
Date Published June 2002 N CP(K)

[MEMO]

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

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

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

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

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

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

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

License not needed:	μPD78F9835
The customer must judge the need for license:	μPD789832, 789833, 789834, 789835

- **The information contained in this document is being issued in advance of the production cycle for the device. The parameters for the device may change before final production or NEC Corporation, at its own discretion, may withdraw the device prior to its production.**
- **Not all devices/types available in every country. Please check with local NEC 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 Corporation. NEC Corporation assumes no responsibility for any errors which may appear in this document.
- NEC Corporation does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from use of a device described herein or any other liability arising from use of such device. No license, either express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Corporation 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 the customer's equipment shall be done under the full responsibility of the customer. NEC Corporation assumes no responsibility for any losses incurred by the customer or third parties arising from the use of these circuits, software, and information.
- While NEC Corporation has been making continuous effort to enhance the reliability of its semiconductor devices, the possibility of defects cannot be eliminated entirely. To minimize risks of damage or injury to persons or property arising from a defect in an NEC semiconductor device, customers must incorporate sufficient safety measures in its design, such as redundancy, fire-containment, and anti-failure features.
- NEC devices are classified into the following three quality grades:
  "Standard", "Special", and "Specific". The Specific quality grade applies only to devices developed based on a customer designated "quality assurance program" for a specific application. The recommended applications of a device depend on its quality grade, as indicated below. Customers must check the quality grade of each device 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 or medical equipment for life support, etc.

  The quality grade of NEC devices is "Standard" unless otherwise specified in NEC's Data Sheets or Data Books. If customers intend to use NEC devices for applications other than those specified for Standard quality grade, they should contact an NEC sales representative in advance.

M5D 98. 12

Regional Information

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

J02.4

INTRODUCTION

Target Readers

This manual is intended for users who wish to understand the functions of the μPD789835 Subseries and to design and develop application systems and programs using these microcontrollers.

Purpose

This manual is intended for users to understand the functions described in the Organization below.

Organization

The μPD789835 Subseries User's Manual is divided into two parts: this manual and instructions (common to the 78K/0S Series).

μPD789835 Subseries User's Manual	78K/0S Series Instructions User's Manual
• Pin functions • Internal block functions • Interrupt functions • Other on-chip peripheral functions • Electrical specifications (target)	• CPU function • Instruction set • Explanation of each instruction

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 the bit whose number is enclosed in brackets, its bit name is defined as a reserved word in the assembler, and in the C compiler, already defined in the header file named sfrbit.h.
- When you know a register name and want to confirm its details:
  → Read **APPENDIX B REGISTER INDEX**.
- To know the 78K/0S Series instruction function in detail:
  → Read **78K/0S Series Instructions User's Manual (U11047E)**.
- To know the electrical specifications of the μPD789835 Subseries:
  → Read **CHAPTER 22 ELECTRICAL SPECIFICATIONS (TARGET)**.

Conventions

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

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

Documents Related to Devices

Document Name	Document No.
μPD789835 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
SM78K0S, SM78K0 System Simulator Ver.2.10 or Later	Operation (Windows™ Based)	U14611E
SM78K Series System Simulator Ver.2.10 or Later	External Part User Open Interface Specifications	U15006E
ID78K0-NS, ID78K0S-NS Integrated Debugger Ver.2.20 or Later	Operation (Windows Based)	U14910E
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-789835-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

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 & Packages	X13769E
Semiconductor Device Mounting Technology Manual	C10535E
Quality Grades on NEC Semiconductor Devices	C11531E
NEC Semiconductor Device Reliability/Quality Control System	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E

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

CONTENTS

LIST OF FIGURES (1/5)

LIST OF FIGURES (2/5)

LIST OF FIGURES (3/5)

LIST OF FIGURES (4/5)

LIST OF FIGURES (5/5)

LIST OF TABLES (1/3)

LIST OF TABLES (2/3)

LIST OF TABLES (3/3)

CHAPTER 1 GENERAL

1.1 Features

- ROM and RAM capacities

Item / Part Number	Program Memory (ROM)		Data Memory		
			Internal High-Speed RAM	Internal Low-Speed RAM	LCD Display RAM
μPD789832	24 KB	Mask ROM	1024 bytes	1216 bytes	288 bytes × 2
μPD789833	32 KB				
μPD789834	48 KB			2240 bytes	
μPD789835	60 KB				
μPD78F9835	60 KB	Flash memory			

- Ceramic/crystal oscillation or RC oscillation is selectable for the oscillator.
- I/O ports: 37
- Timer: 8 channels
  - 8-bit timer: 6 channels
  - Watch timer: 1 channel
  - Watchdog timer: 1 channel
- Sound generator: 16-level volume, 3-octave scale
- 8-bit resolution A/D converter: 3 channels
- Serial interface: 1 channel
- LCD controller/driver
  - Four display modes selectable (48 × 48, 64 × 32, 80 × 16, 80 × 8)
- Power supply voltage: V_{DD} = 1.8 to 3.6 V (Mask ROM version)
  V_{DD} = 3.0 to 3.6 V (Flash memory version)

1.2 Applications

LCD games, remote controllers, pagers, etc.

1.3 Ordering Information

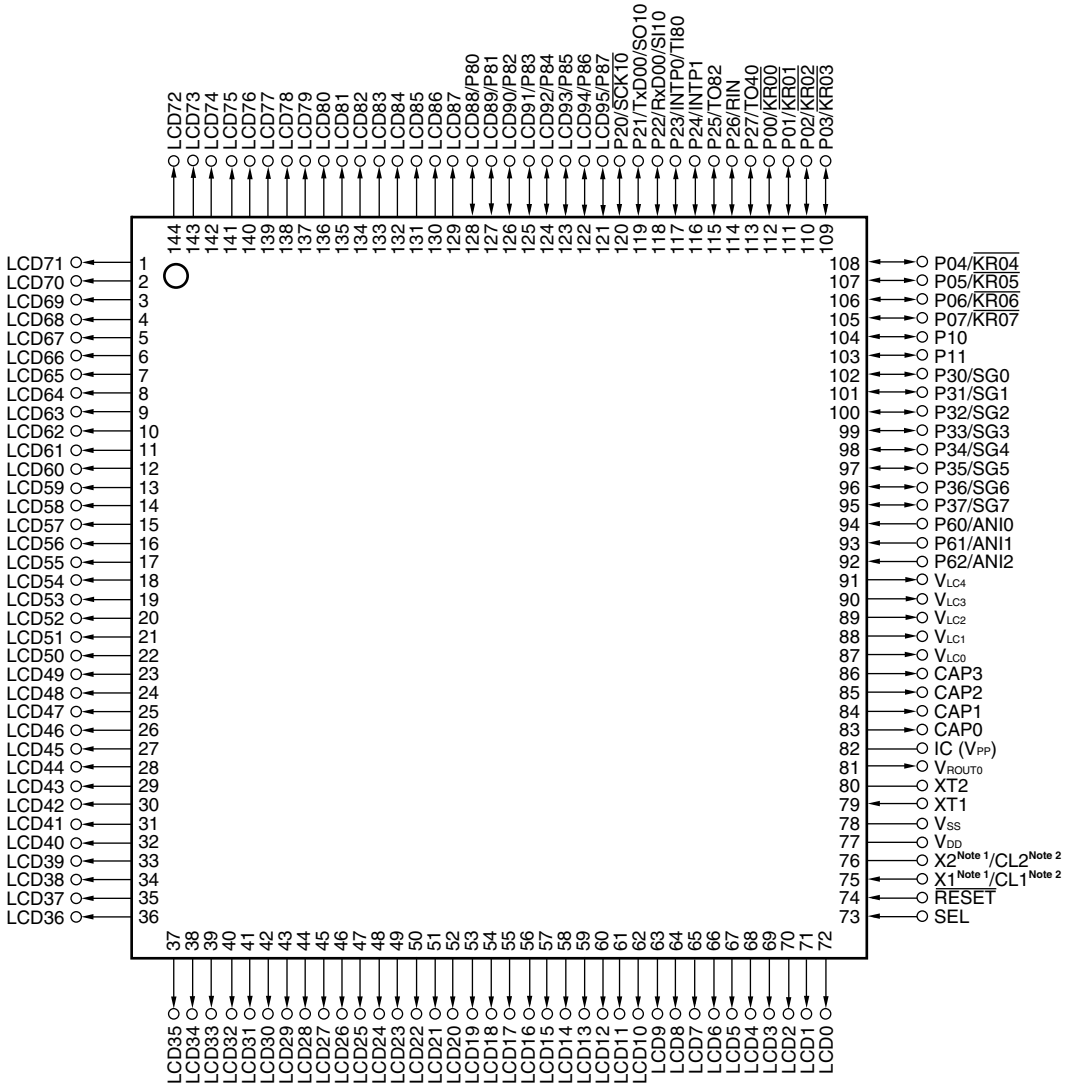
Part Number	Package	Internal ROM
μPD789832GJ-×××-UEN	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM
μPD789833GJ-×××-UEN	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM
μPD789834GJ-×××-UEN	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM
μPD789835GJ-×××-UEN	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM
μPD78F9835GJ-8ET	144-pin plastic LQFP (fine pitch) (20 × 20)	Flash memory

Remark ××× indicates ROM code suffix.

1.4 Pin Configuration (Top View)

144-pin plastic LQFP (fine pitch) (20 × 20)

μPD789832GJ-×××-UEN
μPD789833GJ-×××-UEN
μPD789834GJ-×××-UEN
μPD789835GJ-×××-UEN
μPD78F9835GJ-UEN



Notes 1. When ceramic/crystal oscillation is selected

2. When RC oscillation is selected

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

2. Ceramic/crystal oscillation and RC oscillation can be switched by the SEL pin. Connect the SEL pin to the V_{SS} pin when using ceramic/crystal oscillation, and to the V_{DD} pin when using RC oscillation.

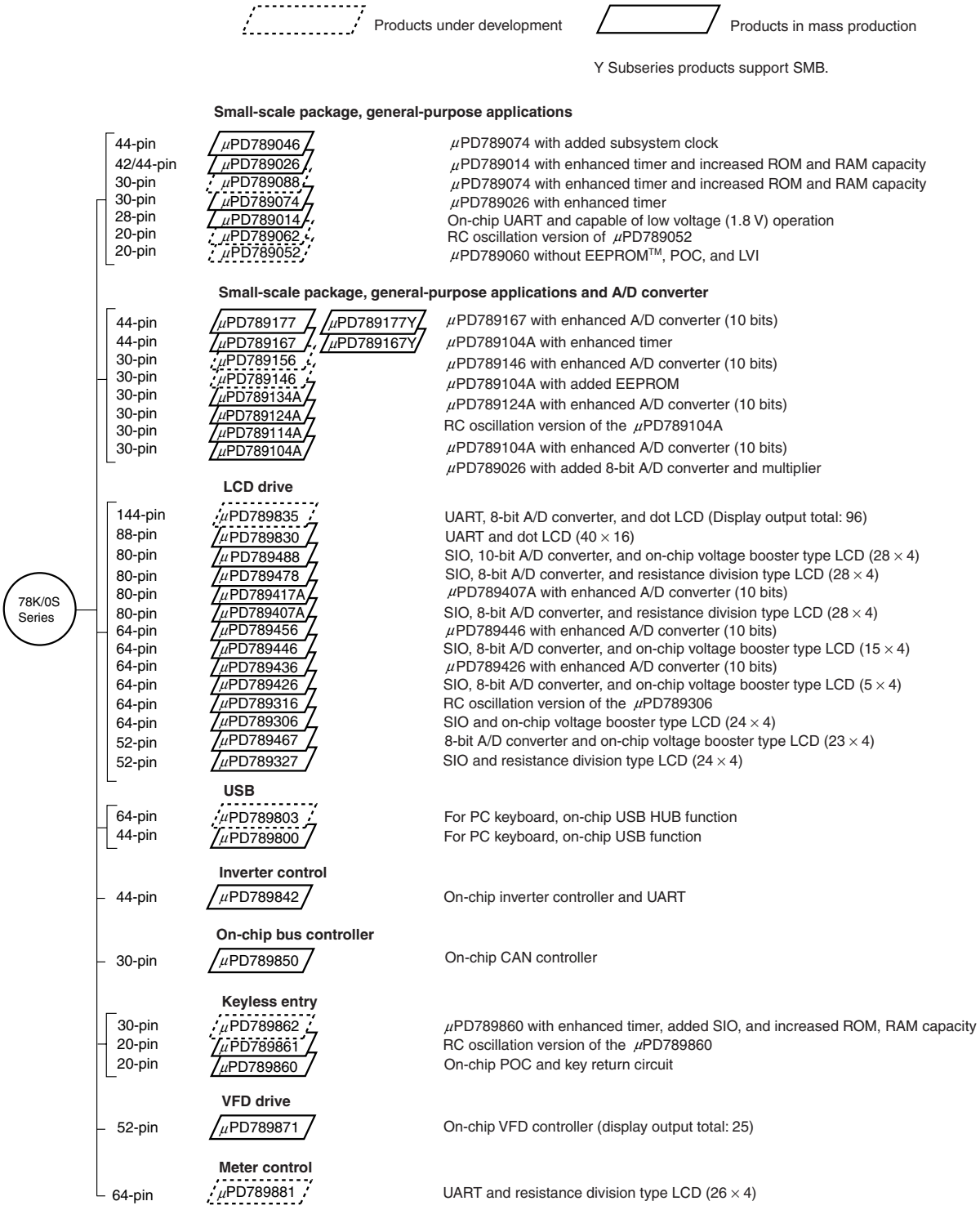
Remark The parenthesized values apply to μPD78F9835.

ANI0 to ANI2:	Analog Input	RIN:	Remote timer input
CAP0 to CAP3:	Output of booster	RxD00:	Receive data
CL1, CL2:	RC oscillator (main system clock)	SEL:	Main system clock selector
IC:	Internally connected	SG0 to SG7:	Sound generator
INTP0, INTP1:	Interrupt from peripherals	TI80:	Timer input
LCD0 to LCD95	Segment/common signal for LCD	TO40, TO82:	Timer output
$\overline{KR00}$ to $\overline{KR07}$:	Key return input	TxD00:	Transmit data
P00 to P07:	Port 0	V_{DD}:	Power supply
P10, P11:	Port 1	V_{LC0} to V_{LC4}:	Output of booster
P20 to P27:	Port 2	V_{PP}:	Programming power supply
P30 to P37:	Port 3	V_{ROUT0}:	Output of regulator
P60 to P62:	Port 6	V_{SS}:	Ground
P80 to P87:	Port 8	X1, X2:	Crystal (main system clock)
$\overline{RESET}$:	Reset	XT1, XT2:	Crystal (subsystem clock)

1.5 78K/0S Series Lineup

78K/0S Series Lineup

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



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

The major functional differences among the subseries are listed below.

Series for LCD drive, general-purpose applications

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

Note Flash memory version: 3.0 V

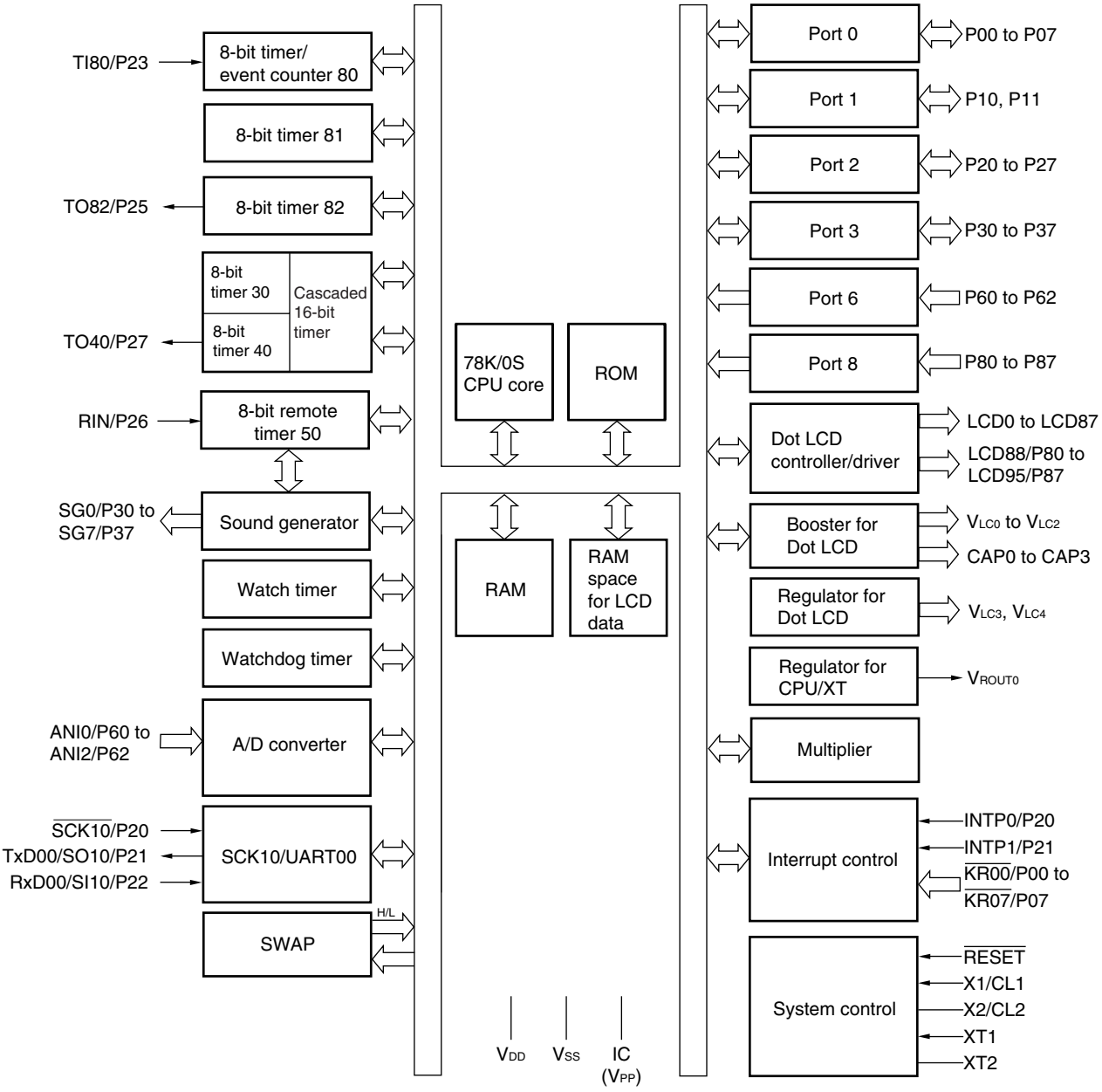
Series for ASSP

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

Notes 1. 10-bit timer: 1 channel

2. Flash memory version: 3.0 V

1.6 Block Diagram



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

2. The parenthesized values apply to μPD78F9835.

1.7 Overview of Functions

Item		μPD789832	μPD789833	μPD789834	μPD789835	μPD78F9835
Internal memory	ROM	Mask ROM				Flash memory
		24 KB	32 KB	48 KB	60 KB	
	High-speed RAM	1024 bytes				
	Low-speed RAM	1216 bytes		2240 bytes		
	LCD display RAM	288 bytes × 2				
Minimum instruction execution time	Ceramic/crystal oscillation	0.4 μs/1.6 μs (@ 5.0 MHz operation with main system clock)				
	RC oscillation	1.0 μs/4.0 μs (@ 2.0 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 operation • Bit manipulation (set, reset, test)				
I/O ports		Total: 37 • CMOS I/O: 26 • CMOS input: 11				
Timers		• 16-bit timer: 6 channels • Watch timer: 1 channel • Watchdog timer: 1 channel				
Sound generator		Volume: 16 levels, scale: 3 octaves				
A/D converter		8-bit resolution × 3 channels				
Serial interface		Switchable between UART and 3-wire serial I/O modes: 1 channel				
LCD controller/driver		4 display modes selectable (48 × 48, 64 × 32, 80 × 16, 80 × 8)				
Multiplier		8 bits × 8 bits = 16 bits				
SWAP		Higher and lower 4 bits of an 8-bit register can be swapped.				
Vectored interrupt sources	Maskable	Internal: 15, External: 5				
	Non-maskable	Internal: 1				
Power supply voltage (V_{DD})		1.8 to 3.6 V				3.0 to 3.6 V
Operating ambient temperature		T_A = –40 to +85°C				
Package		144-pin plastic LQFP (fine pitch) (20 × 20)				

An outline of the timer is shown below.

		8-Bit Timer/Event Counter 80	8-Bit Timer 81	8-Bit Timer 82	8-Bit Timer 30	8-Bit Timer 40	8-Bit Remote Control Timer 50	Watch Timer	Watchdog Timer
Operation mode	Interval timer	1 channel	1 channel	1 channel	1 channel	1 channel	1 channel	1 channel[Note 1]	1 channel[Note 2]
	External event counter	1 channel	–	–	–	–	–	–	–
Function	Timer outputs	–	–	1 output	–	1 output	–	–	–
	Square-wave outputs	–	–	1 output	–	1 output	–	–	–
	Capture	–	–	–	–	–	–	–	–
	Interrupt sources	1	1	1	1	1	3	2	2

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

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

CHAPTER 2 PIN FUNCTIONS

2.1 List of Pin Functions

(1) Port pins

Pin Name	I/O	Function	After Reset	Alternate Function
P00 to P07	I/O	Port 0. 8-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 0 (PU0).	Input	$\overline{KR00}$ to $\overline{KR07}$
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, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 0 (PU0).	Input	—
P20	I/O	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register B2 (PUB2).	Input	$\overline{SCK10}$
P21				TxD00/SO10
P22				RxD00/SI10
P23				INTP0/TI80
P24				INTP1
P25				TO82
P26				RIN
P27				TO40
P30 to P37	I/O	Port 3. 8-bit I/O port. Input/output can be specified in 1-bit units. When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register B3 (PUB3).	Input	SG0 to SG7
P60 to P62	Input	Port 6. 3-bit input port.	Input	ANI0 to ANI2
P80 to P87	Input	Port 8. 8-bit input-only port when used as a general-purpose port. Output-only pins when used for alternate function (LCD display).	Output	LCD88 to LCD95

(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	P23/TI80
INTP1				P24
$\overline{KR00}$ to $\overline{KR07}$	Input	Key return signal detection	Input	P00 to P07
SI10	Input	Serial interface serial data input	Input	P22/RxD00
SO10	Output	Serial interface serial data output	Input	P21/TxD00
$\overline{SCK10}$	I/O	Serial interface serial clock input/output	Input	P20
RxD00	Input	Serial data input for asynchronous serial interface	Input	P22/SI10
TxD00	Output	Serial data output for asynchronous serial interface	Input	P21/SO10
TI80	Input	External count clock input to 8-bit timer/event counter 80	Input	P23/INTP0
TO82	Output	8-bit timer 82 output	Input	P25
RIN	Input	Input to 8-bit remote control timer 50	Input	P26
TO40	Output	8-bit timer 40 output	Input	P27
SG0 to SG7	Output	Frequency output for sound generator	Input	P30 to P37
ANI0 to ANI2	Input	A/D converter analog input	Input	P60 to P62
LCD0 to LCD87	Output	Segment/common signal output	Output	—
LCD88 to LCD95				P80 to P87
V_{LC0} to V_{LC4}	—	LCD drive voltage	—	—
CAP0 to CAP3	—	Connection pin for LCD drive capacitor	—	—
V_{ROUT0}	—	Regulator output for subsystem clock oscillation. Connect to V_{SS} via a 0.1 μF capacitor.	—	—
SEL	Input	Input to switch ceramic/crystal oscillation or RC oscillation. Connect to V_{SS} when using ceramic/crystal oscillation, and connect to V_{DD} when using RC oscillation.	—	—
X1	Input	Connecting ceramic/crystal resonator for main system clock oscillation	—	—
X2	—		—	—
CL1	Input	Connecting resistor (R) and capacitor (C) for main system clock oscillation	—	—
CL2	—		—	—
XT1	Input	Connecting crystal resonator for subsystem clock oscillation	—	—
XT2	—		—	—
$\overline{RESET}$	Input	System reset input	Input	—
V_{DD}	—	Positive power supply for ports	—	—
V_{SS}	—	Ground potential	—	—
IC	—	Internally connected. Connect directly to V_{SS}.	—	—
V_{PP}	—	Sets flash memory programming mode. Applies high voltage when a program is written or verified.	—	—

2.2 Description of Pin Functions

2.2.1 P00 to P07 (Port 0)

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

(1) Port mode

These pins constitute an 8-bit I/O port and can be set in the input or output port mode in 1-bit units by port mode register 0 (PM0). When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 0 (PU0) in port units.

(2) Control mode

In this mode, P00 to P07 function as key return signal detection pins ($\overline{KR00}$ to $\overline{KR07}$).

2.2.2 P10, P11 (Port 1)

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

2.2.3 P20 to P27 (Port 2)

These pins constitute an 8-bit I/O port. In addition, these pins enable serial interface data I/O, clock I/O, external interrupt input, and timer I/O.
Port 2 can be specified in the following operation modes in 1-bit units.

(1) Port mode

In this mode, P20 to P27 function as an 8-bit I/O port. Port 2 can be set in the input or output port mode in 1-bit units by port mode register 2 (PM2). When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register B2 (PUB2).

(2) Control mode

In this mode, P20 to P27 function as the serial interface data I/O, clock I/O, external interrupt input, and timer I/O.

(a) SI10, SO10

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

(b) $\overline{SCK10}$

This is the serial clock I/O pin of the serial interface.

(c) RxD00, TxD00

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

(d) TI80

This is the external clock input pin for 8-bit timer/event counter 80.

(e) TO40, TO82

These are the output pins of the 8-bit timer.

(f) RIN

This is the input pin for 8-bit remote control timer 50.

(g) INTP0, INTP1

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

Caution When using P20 to P27 as serial interface pins, the I/O mode and output latch must be set according to the functions to be used. For the details of the setting, refer to Table 13-2 Settings of Serial Interface Operating Mode.

2.2.4 P30 to P37 (Port 3)

These pins constitute an 8-bit I/O port. In addition, they also function as frequency output for sound generator.

Port 3 can be specified in the following operation mode by P3 function register (PF3).

(1) Port mode

In this mode, port 3 functions as an 8-bit I/O port. Port 3 can be set in the input or output port mode in 1-bit units by port mode register 3 (PM3). When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register B3 (PUB3) in 1-bit units.

(2) Control mode

In this mode, the pins function as frequency output for sound generator (SG0 to SG7).

2.2.5 P60 to P62 (Port 6)

These pins constitute a 3-bit input-only port. In addition, they also function as A/D converter input.

(1) Port mode

In this mode, port 6 functions as a 3-bit input-only port.

(2) Control mode

In this mode, the pins of port 6 function as A/D converter analog inputs (ANI0 to ANI2).

2.2.6 P80 to P87 (Port 8)

These pins constitute an 8-bit input port. In addition, they also function as LCD controller/driver segment signal output.

Port 8 can be specified in the following operation mode in 8-bit units by LCD display mode register 20 (LCDM20).

(1) Control mode

In this mode, the pins function as LCD controller/driver segment/common signal output (LCD88 to LCD95).

(2) Port mode

In this mode, port 8 functions as an 8-bit input port.

2.2.7 LCD0 to LCD87

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

2.2.8 CAP0 to CAP3

These pins are the capacitor output pins used to drive the LCD. Connect a 0.1 μF capacitor.

2.2.9 V_{ROUT0}

This pin is the regulator output pin for subsystem clock oscillation. Connect to V_{SS} via a 0.1 μF capacitor.

2.2.10 V_{LC0} to V_{LC4}

These pins are the power supply voltage pins to drive the LCD. Connect a 0.1 μF capacitor.

2.2.11 $\overline{\text{RESET}}$

This pin inputs an active-low system reset signal.

2.2.12 SEL

This pin is the input pin for switching between ceramic/crystal oscillation and RC oscillation. Connect to V_{SS} when using ceramic/crystal oscillation, and connect to V_{DD} when using RC oscillation.

2.2.13 X1, X2

These pins are used to connect a crystal resonator for main system clock oscillation when ceramic/crystal oscillation is selected.

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

2.2.14 CL1, CL2

These pins are used to connect a resistor (R) and a capacitor (C) for main system clock oscillation when RC oscillation is selected.

To supply an external clock, input the clock to CL1 and input the inverted signal to CL2.

2.2.15 XT1, XT2

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

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

2.2.16 V_{DD}

This is the positive power supply pin.

2.2.17 V_{SS}

This is the ground pin.

2.2.18 V_{PP} (μPD78F9835 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.

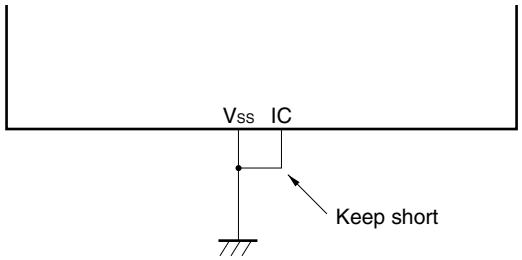
- Independently connect a 10 kΩ pull-down resistor to V_{PP}.
- Use the jumper on the board to connect V_{PP} to the dedicated flash programmer or V_{SS}, in programming mode or normal operation mode, respectively.

2.2.19 IC (mask ROM version only)

The IC (Internally Connected) pin is used to set the μPD789832, 789833, 789834, and 789835 in the test mode before shipment. In the normal operation mode, directly connect this pin to the V_{SS} pin with as short a wiring length as possible.

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

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



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 P07	5-A	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P10, P11			
P20/$\overline{SCK10}$	8-A		
P21/TxD00/SO10	5-A		
P22/RxD00/SI10	8-A		
P23/INTP0/TI80			
P24/INTP1			
P25/TO82	5-A		
P26/RIN	8-A		
P27/TO40	5-AA		
P30/SG0, P34/SG4	5-AB		
P31/SG1 to P33/SG3, P35/SG5 to P37/SG7	5-A		
P60/ANI0 to P62/ANI2	9-C	Input	Connect directly to V_{DD} or V_{SS}.
P80/LCD88 to P87/LCD95	17-J	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
LCD0 to LCD87	17-K	Output	Leave open.
SEL	2	Input	—
XT1	—		Connect to V_{SS}.
XT2	—	—	Leave open.
$\overline{RESET}$	2	Input	—
IC (mask ROM version)	—	—	Connect directly to V_{SS}.
V_{PP} (μPD78F9835)			Independently connect a 10 kΩ pull-down resistor to V_{PP} or directly connect to V_{SS}.

Figure 2-1. Pin I/O Circuits (1/2)

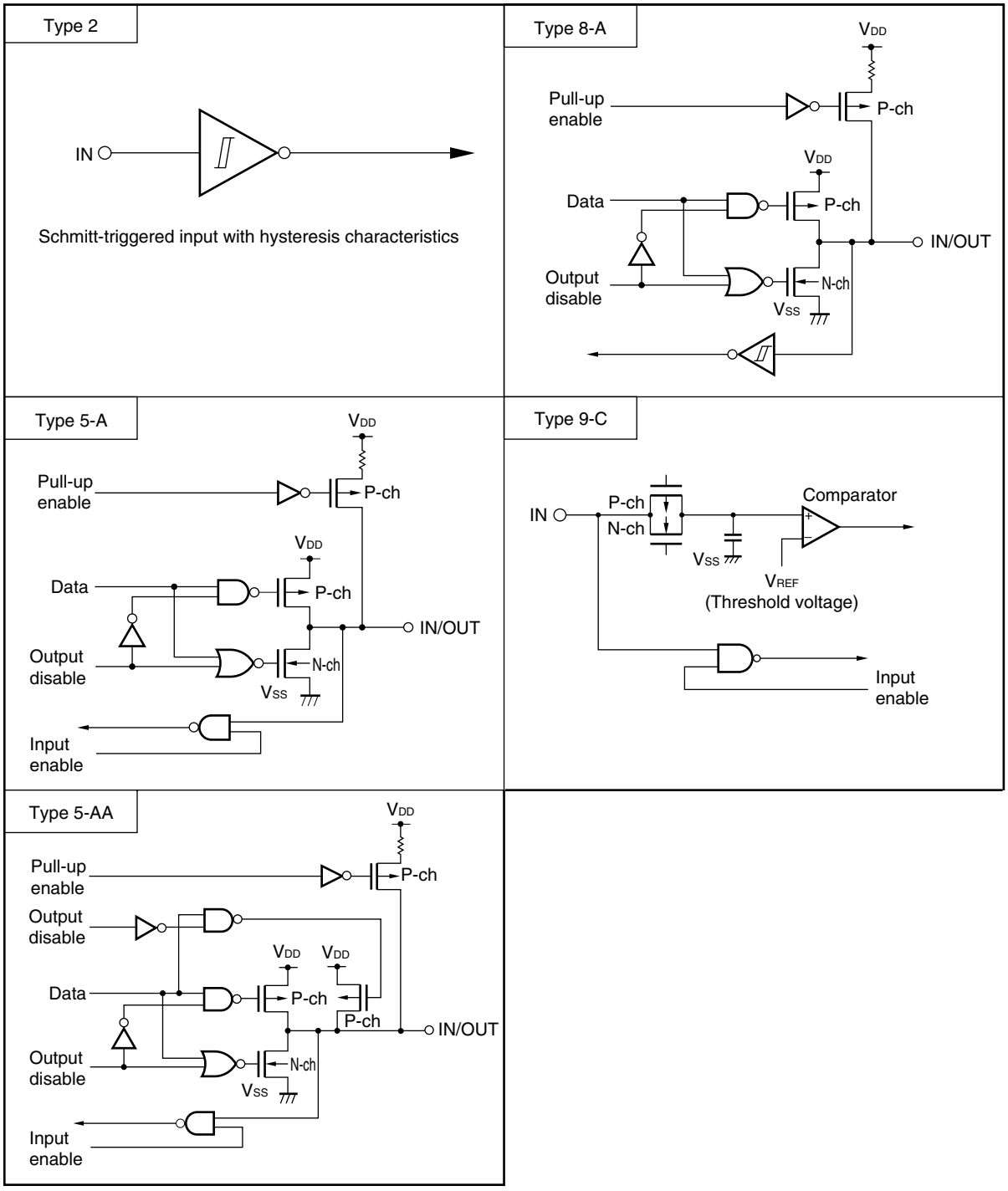
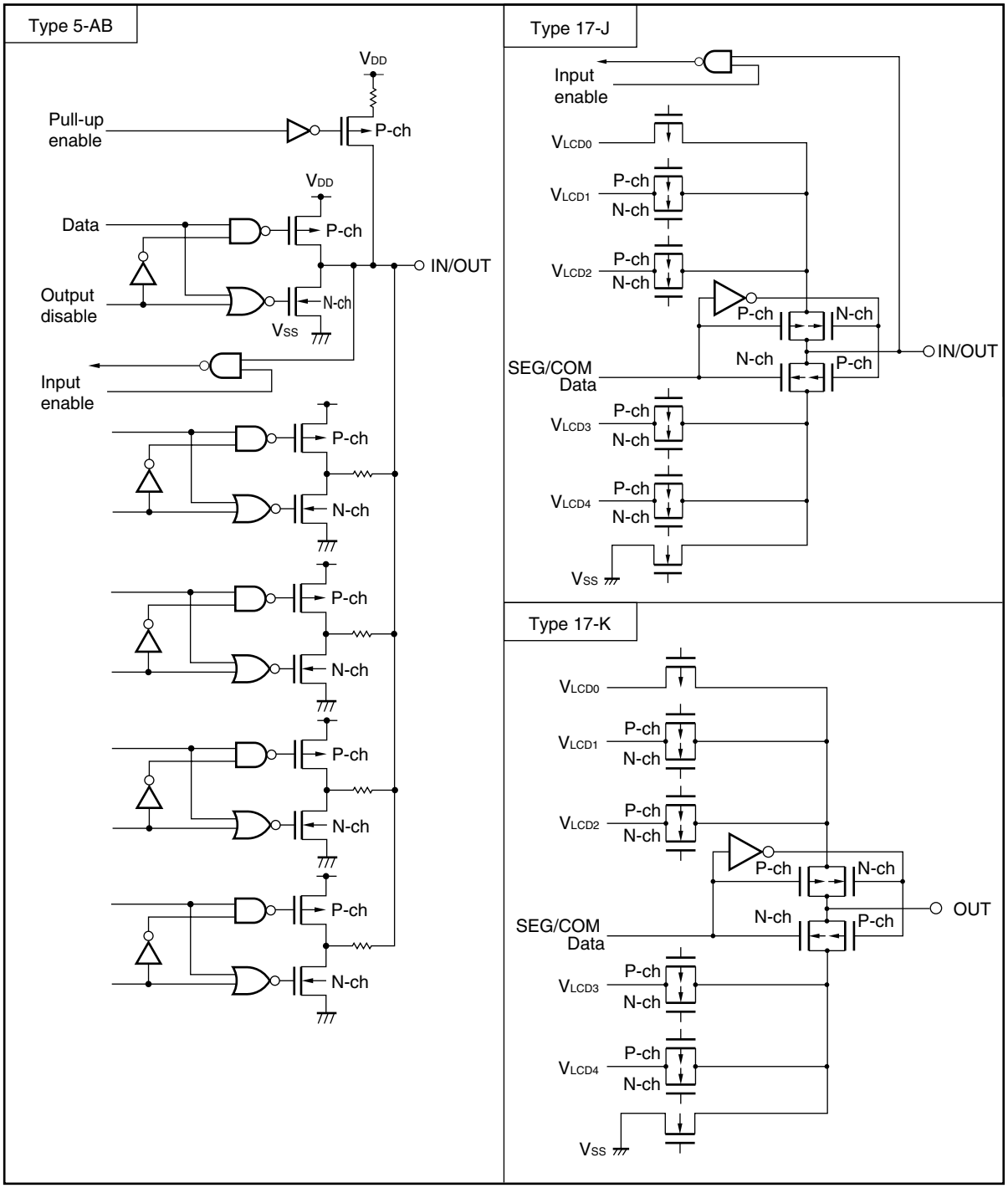


Figure 2-1. Pin I/O Circuits (2/2)

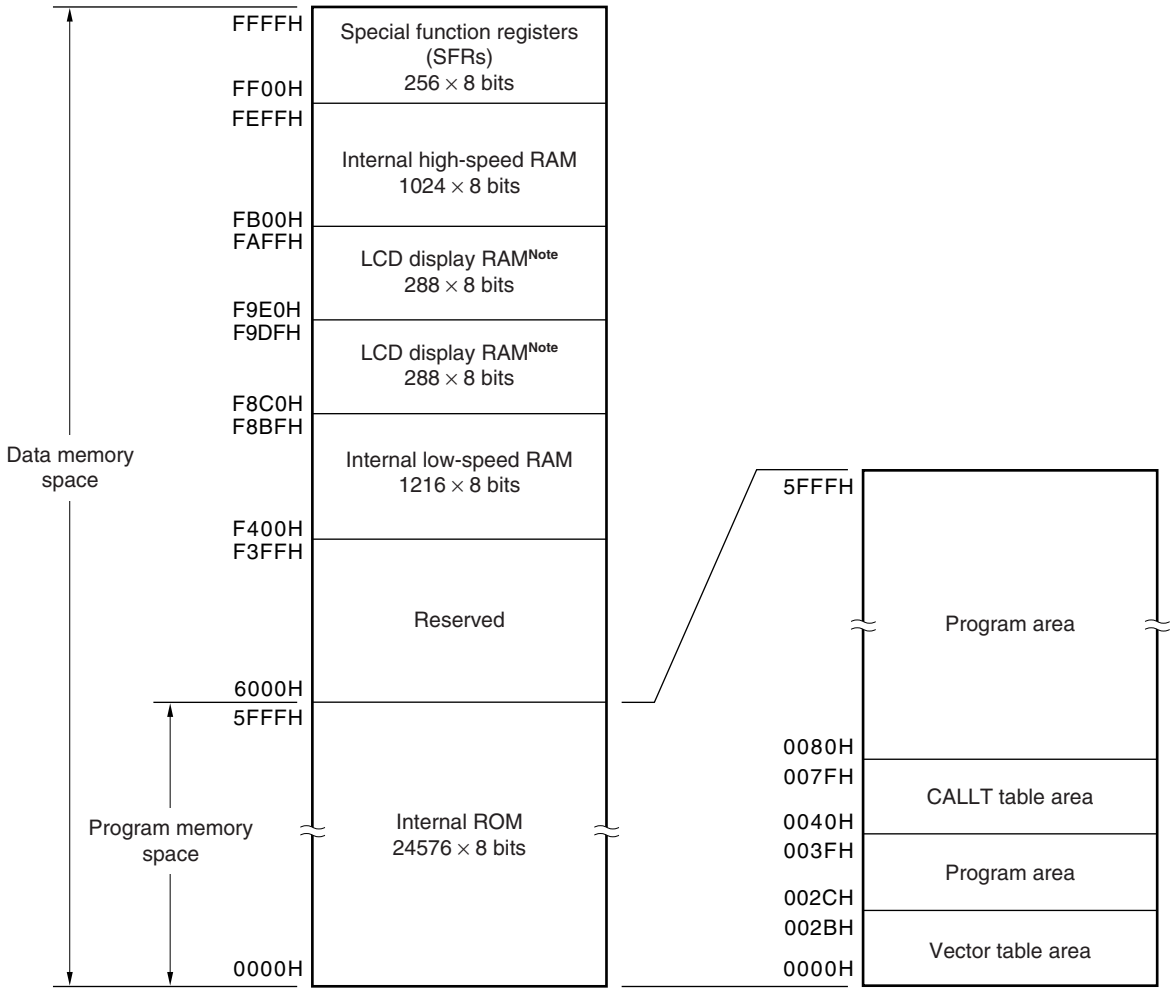


CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

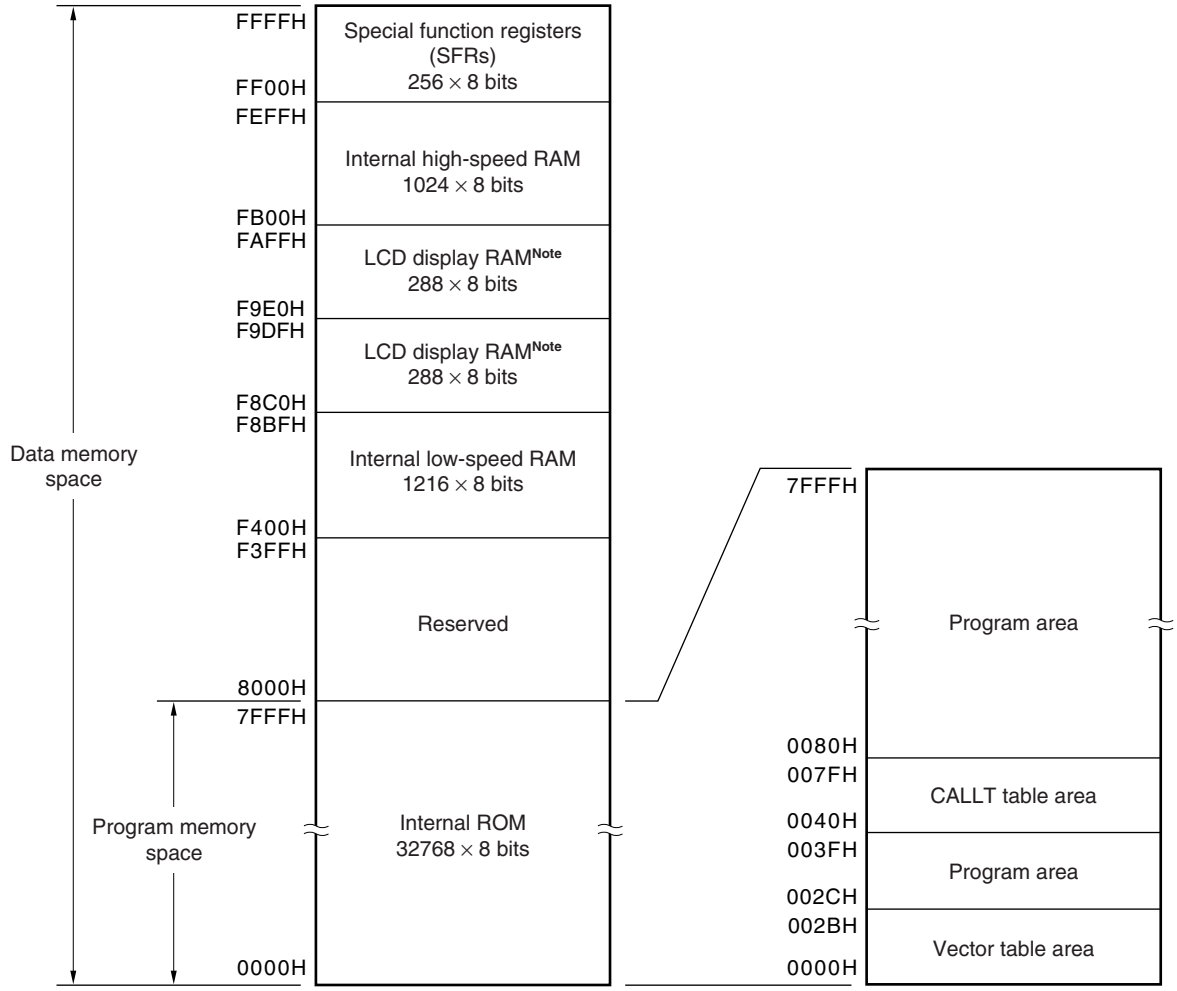
The μPD789835 Subseries can access 64 KB of memory space. Figures 3-1 through 3-5 show the memory maps.

Figure 3-1. Memory Map (μPD789832)



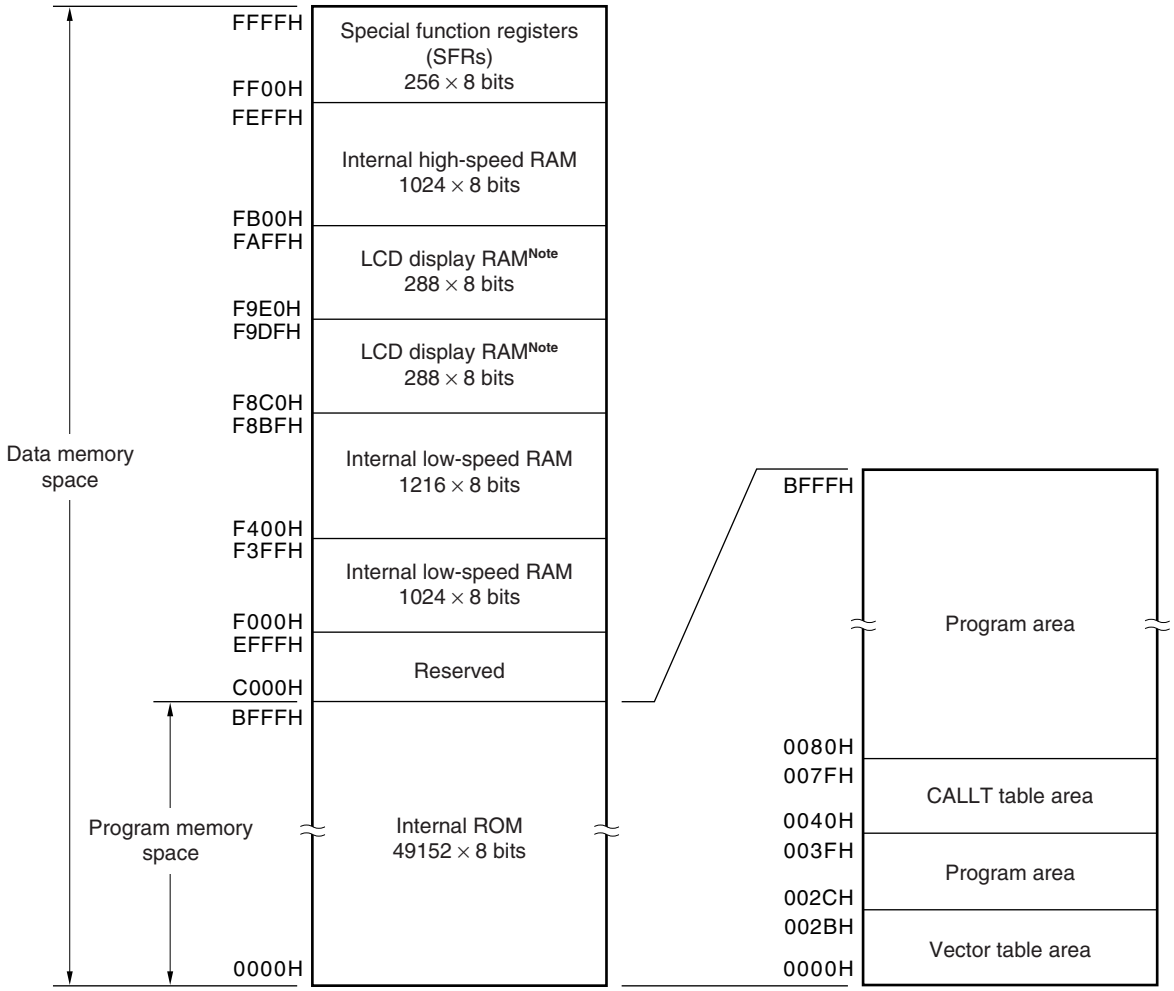
Note Can be used as normal RAM when not being used for LCD display.

Figure 3-2. Memory Map (μPD789833)



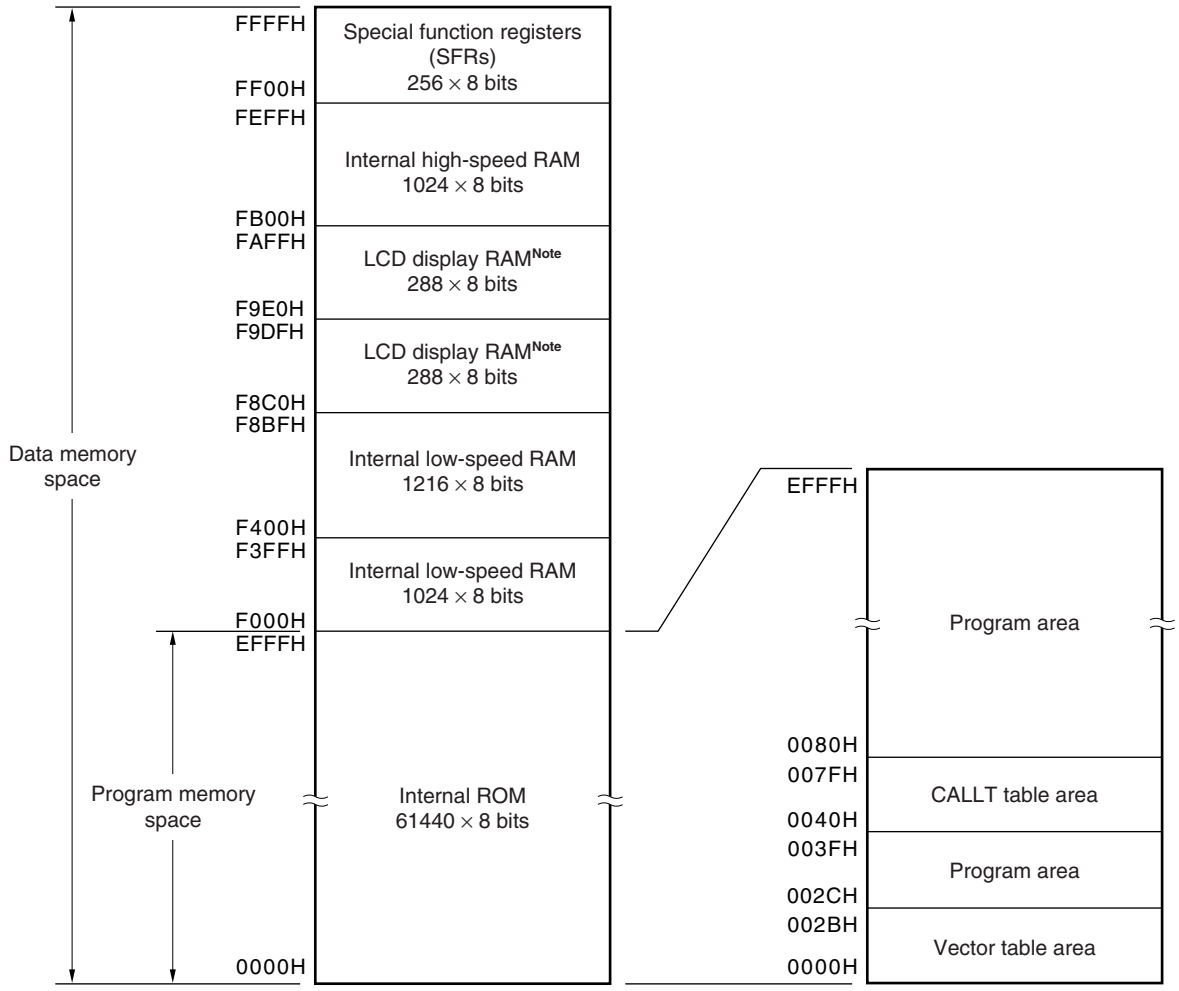
Note Can be used as normal RAM when not being used for LCD display.

Figure 3-3. Memory Map (μPD789834)



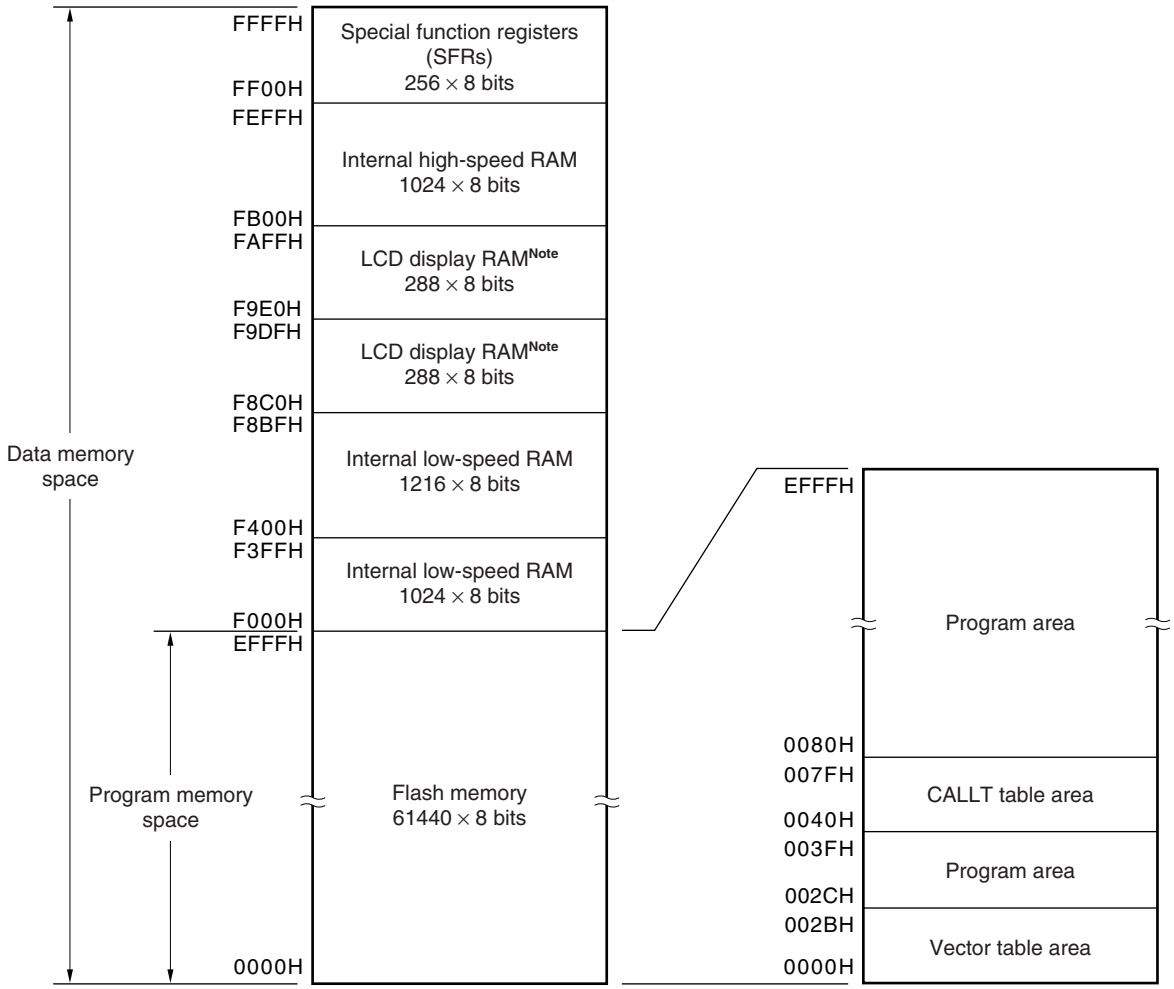
Note Can be used as normal RAM when not being used for LCD display.

Figure 3-4. Memory Map (μPD789835)



Note Can be used as normal RAM when not being used for LCD display.

Figure 3-5. Memory Map (μPD78F9835)



Note Can be used as normal RAM when not being used for LCD display.

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 μPD789835 Subseries provide internal ROM (or flash memory) with the following capacity for each product.

Table 3-1. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD789832	Mask ROM	24576 × 8 bits
μPD789833		32768 × 8 bits
μPD789834		49152 × 8 bits
μPD789835		61440 × 8 bits
μPD78F9835	Flash memory	61440 × 8 bits

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

(1) Vector table area

The 44-byte area of addresses 0000H to 002BH is reserved as a vector table area. This area stores program start addresses to be used when branching by the $\overline{\text{RESET}}$ input or an interrupt request generation. Of a 16-bit program address, the lower 8 bits are stored in an even address, and the higher 8 bits are stored in an odd address.

Table 3-2. Vector Table

Vector Table Address	Interrupt Request	Vector Table Address	Interrupt Request
0000H	$\overline{\text{RESET}}$ input	0018H	INTWT
0004H	INTWDT1	001AH	INTWTI
0006H	INTP0	001CH	INTTM80
0008H	INTP1	001EH	INTTM81
000AH	INTCSI10	0020H	INTTM82
000CH	INTSER00	0022H	INTTM30
000EH	INTSR00	0024H	INTTM40
0010H	INTST00	0026H	INTTMSG0
0012H	INTTM50	0028H	INTAD
0014H	INTTM51	002AH	INTKR00
0016H	INTTM52		

(2) CALLT instruction table area

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

3.1.2 Internal data memory space

The μPD789835 Subseries products incorporate the following RAM.

(1) Internal RAM

The μPD789835 Subseries incorporate the following RAM.

Table 3-3. Internal RAM Capacity

Part Number	Capacity
μPD789832	2240 bytes (high-speed RAM 1024 bytes + low-speed RAM 1216 bytes)
μPD789833	
μPD789834	3264 bytes (high-speed RAM 1024 bytes + low-speed RAM 2240 bytes)
μPD789835	
μPD78F9835	

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

(2) LCD display RAM

LCD display RAM is allocated in the area between F8C0H and FAFFH. The LCD display RAM can also be used as ordinary RAM.

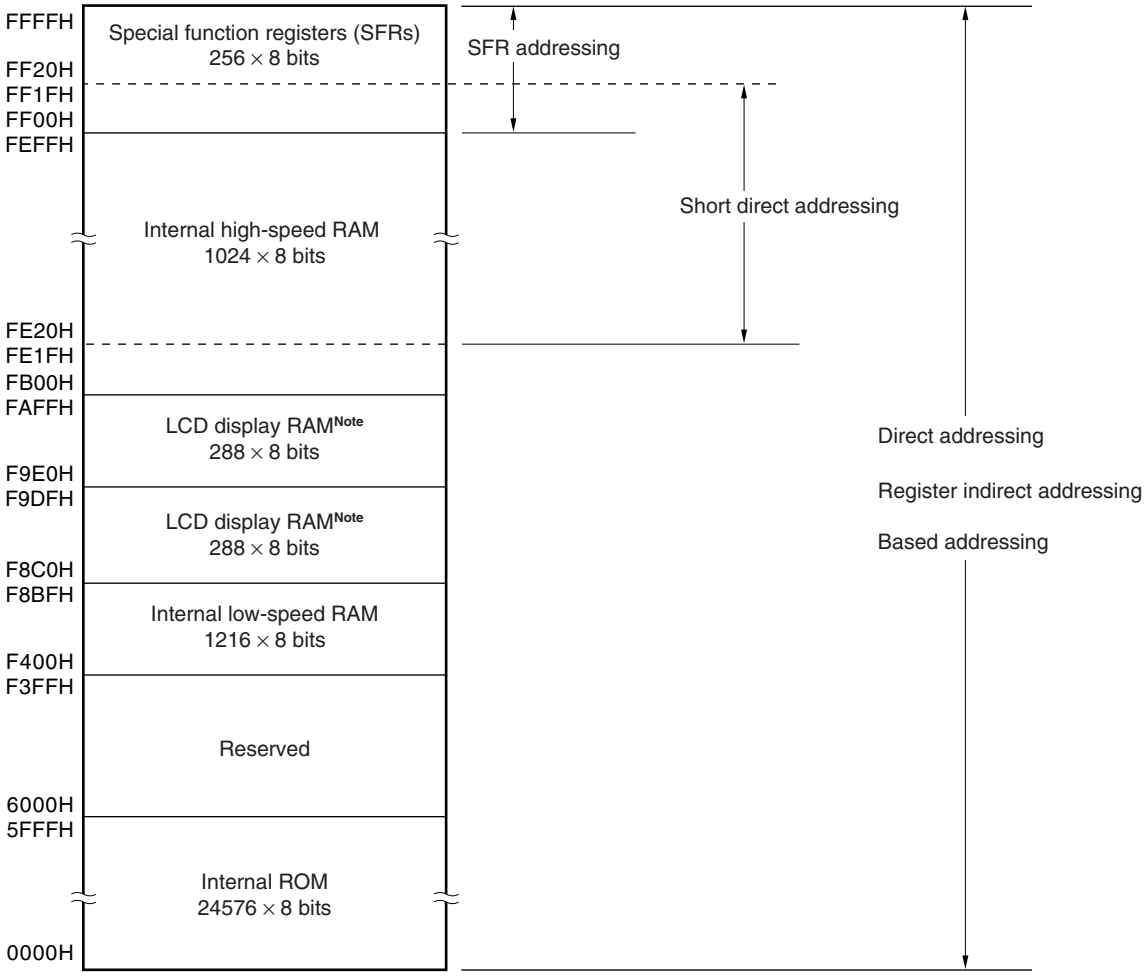
3.1.3 Special function register (SFR) area

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

3.1.4 Data memory addressing

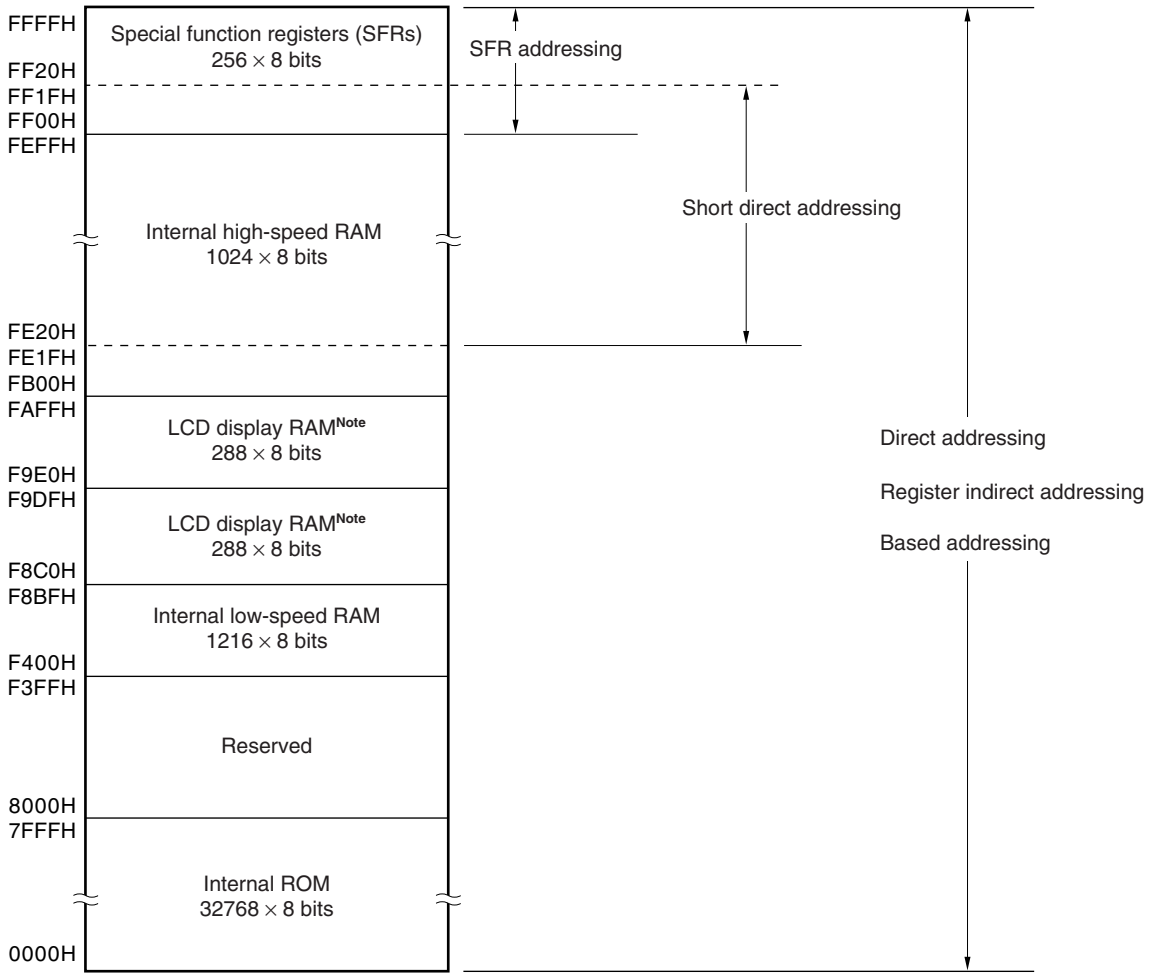
The μPD789835 Subseries are provided with a variety of addressing modes to make memory manipulation as efficient as possible. At the addresses corresponding to data memory area especially, specific addressing modes that correspond to the particular function an area, such as the special function registers are available. Figures 3-6 through 3-10 show the data memory addressing modes.

Figure 3-6. Data Memory Addressing (μPD789832)



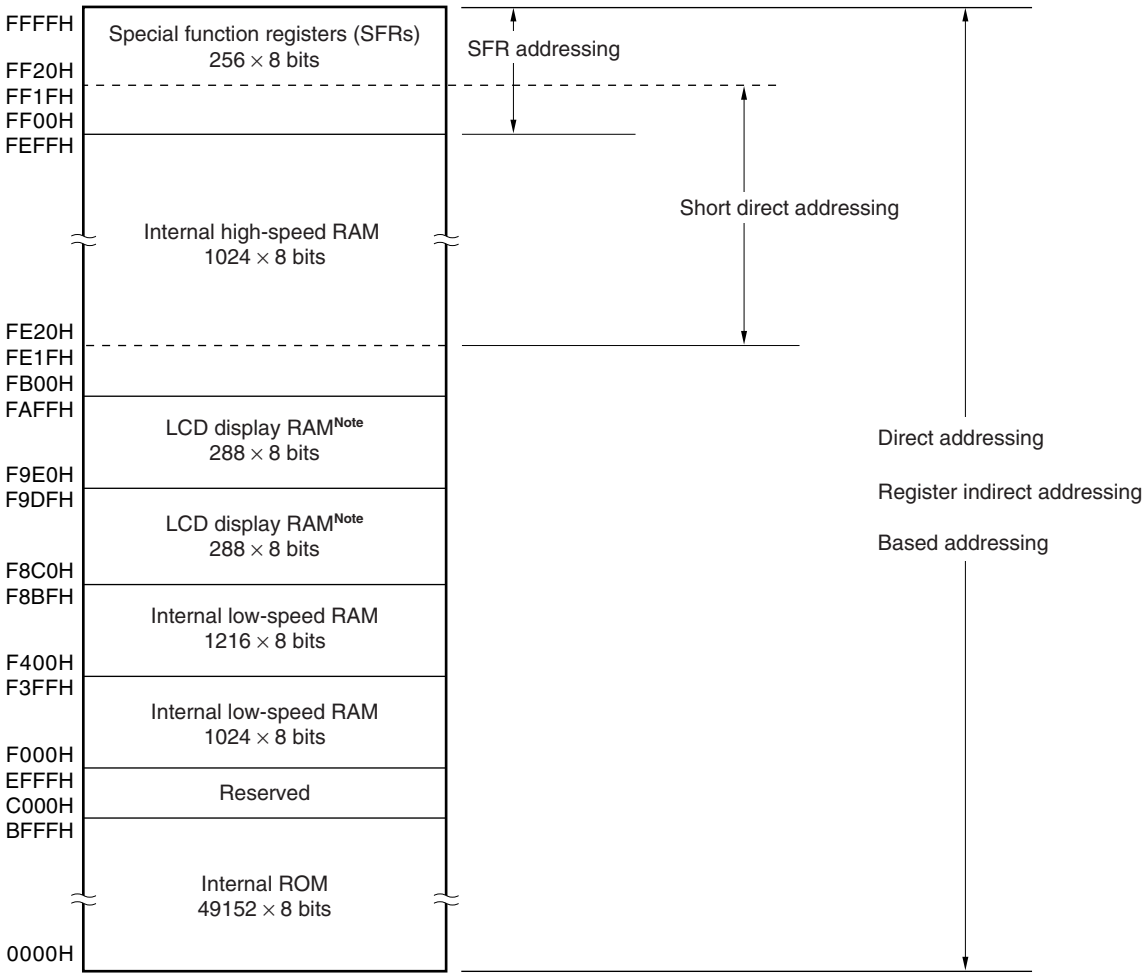
Note Can be used as normal RAM when not being used for LCD display.

Figure 3-7. Data Memory Addressing (μPD789833)



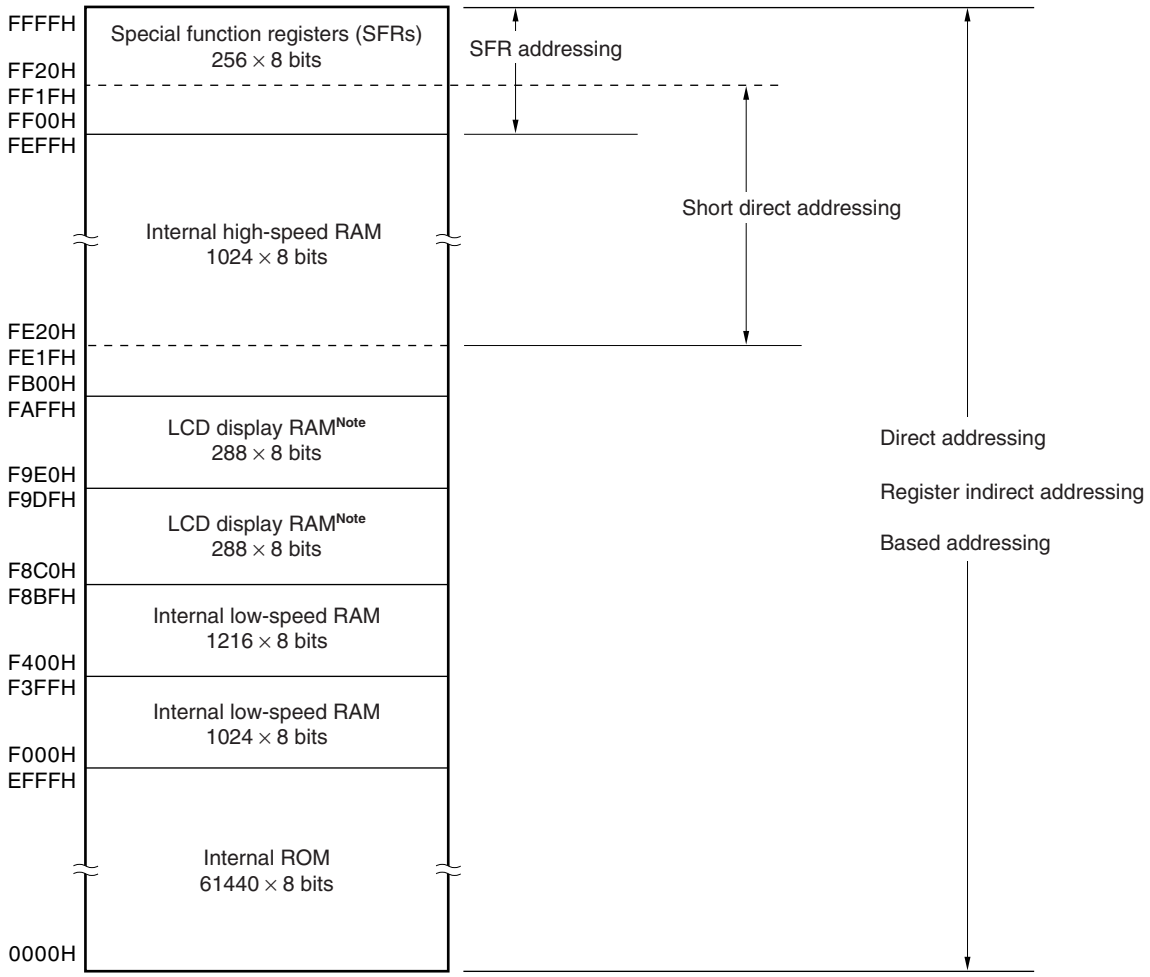
Note Can be used as normal RAM when not being used for LCD display.

Figure 3-8. Data Memory Addressing (μPD789834)



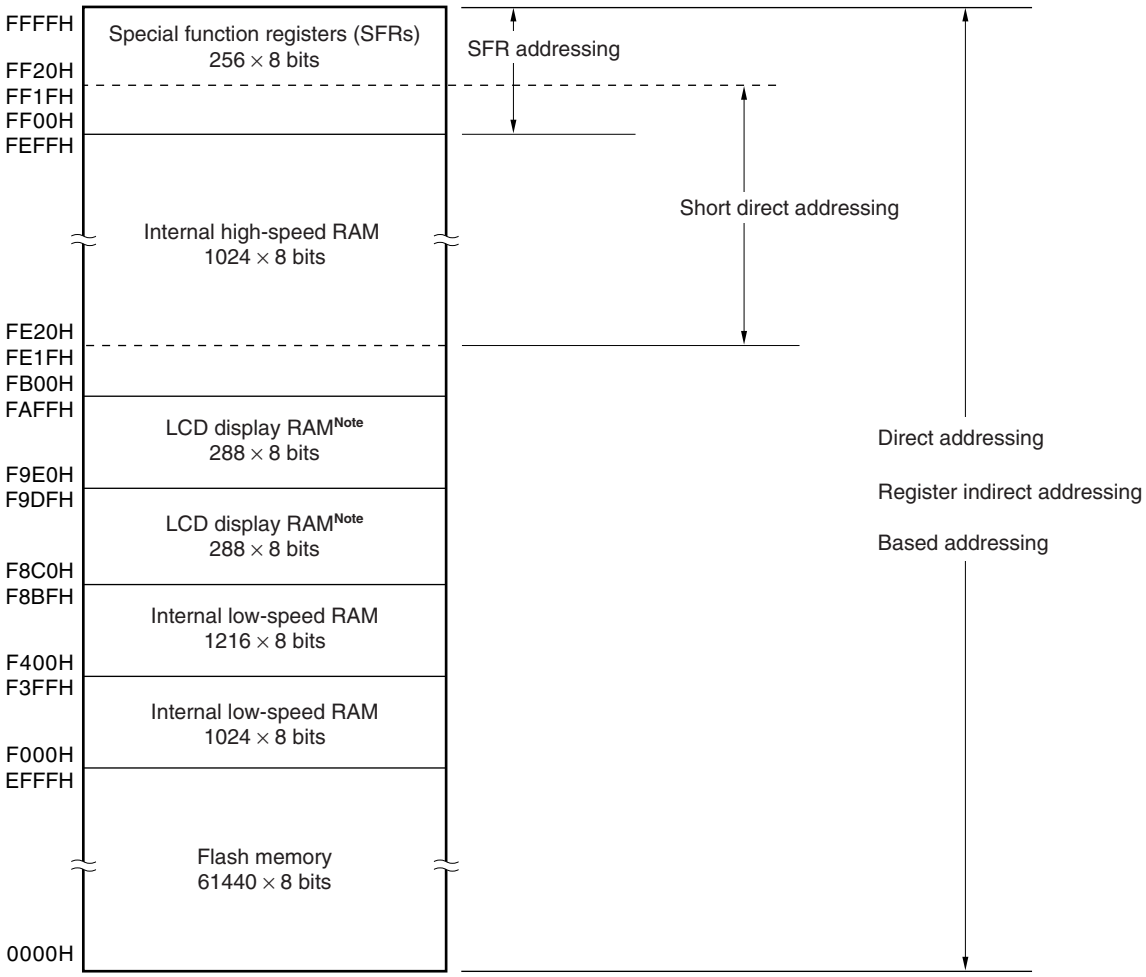
Note Can be used as normal RAM when not being used for LCD display.

Figure 3-9. Data Memory Addressing (μPD789835)



Note Can be used as normal RAM when not being used for LCD display.

Figure 3-10. Data Memory Addressing (μPD78F9835)



Note Can be used as normal RAM when not being used for LCD display.

3.2 Processor Registers

The μPD789835 Subseries provide 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 interrupt request acknowledgement operations of the CPU.

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

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

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

(b) Zero flag (Z)

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

(c) Auxiliary carry flag (AC)

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

(d) Carry flag (CY)

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

(3) Stack pointer (SP)

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

Figure 3-13. Stack Pointer Configuration

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

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

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

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

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

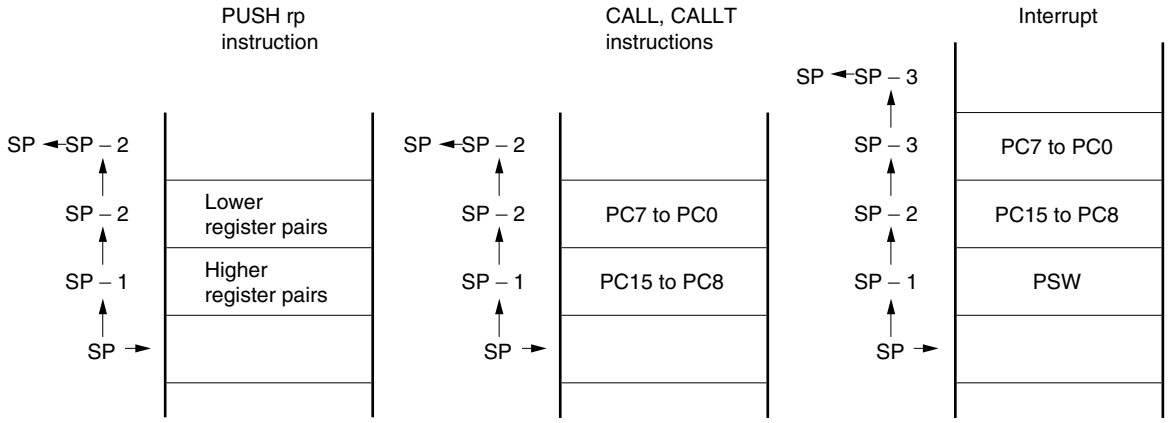
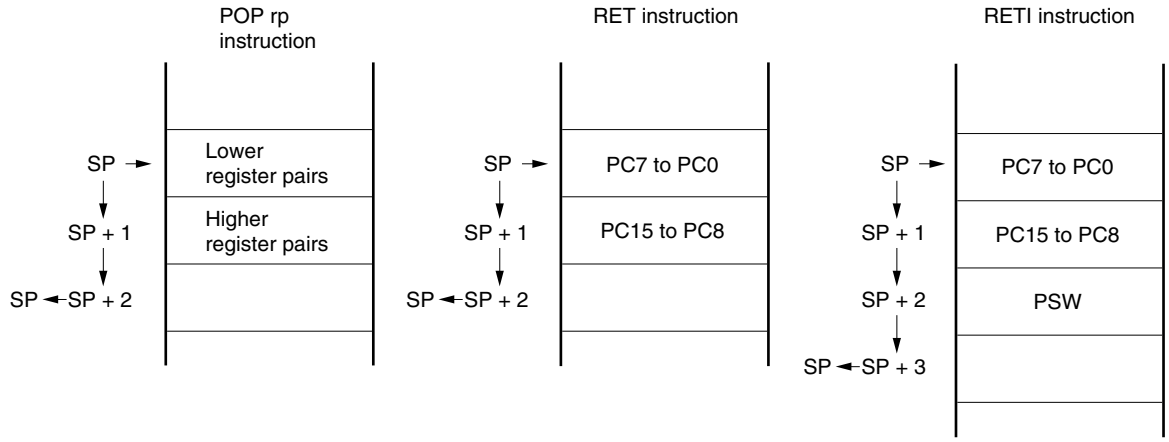


Figure 3-15. Data to Be Restored from Stack Memory



3.2.2 General-purpose registers

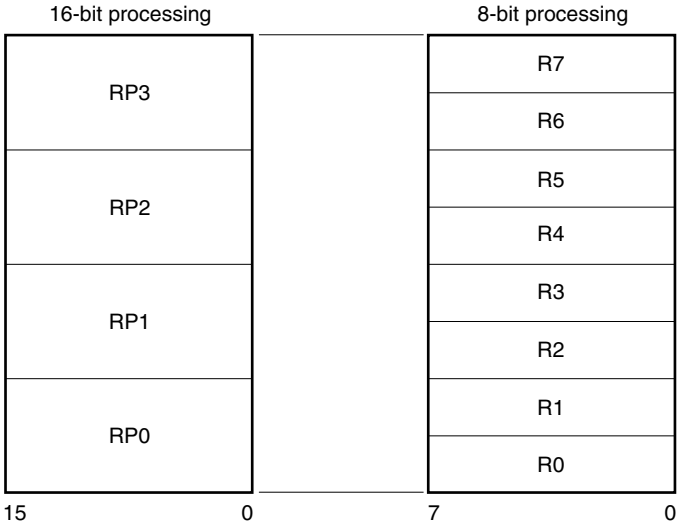
The general-purpose registers consist of eight 8-bit registers (X, A, C, B, E, D, L, and H).

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

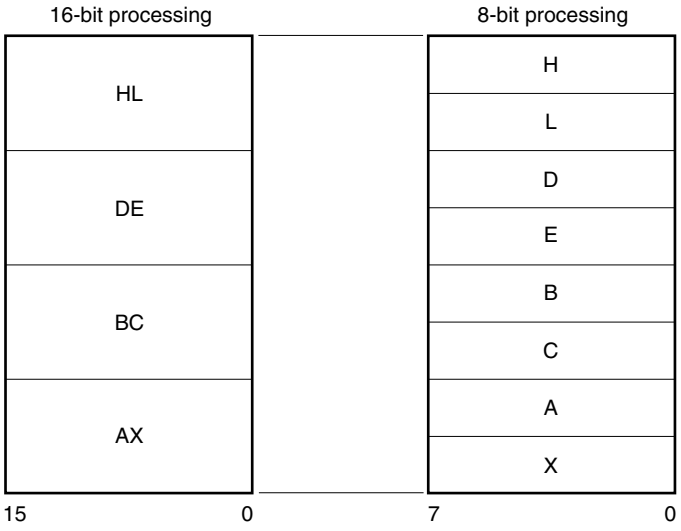
General-purpose registers can be described in terms of function names (X, A, C, B, E, D, L, H, AX, BC, DE, or HL) or absolute names (R0 to R7 and RP0 to RP3).

Figure 3-16. General-Purpose Register Configuration

(a) Absolute names



(b) Function names



3.2.3 Special function registers (SFRs)

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

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

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

The manipulatable bits can be specified as follows.

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

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

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

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

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

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

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

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

Table 3-4. Special Function Registers (1/2)

Address	Special Function Register (SFR) Name	Symbol		R/W	Bit Units for Manipulation			After Reset
					1 Bit	8 Bits	16 Bits	
FF00H	Port 0	P0		R/W	√	√	—	00H
FF01H	Port 1	P1			√	√	—	
FF02H	Port 2	P2			√	√	—	
FF03H	Port 3	P3			√	√	—	
FF06H	Port 6	P6		R	√	√	—	Undefined
FF08H	Port 8	P8			√	√	—	
FF11H	A/D conversion result register	ADCR			—	√	—	
FF14H	16-bit multiplication result store register 0	MUL0[Note 1]	MUL0L		—	√	√[Note 2]	
FF15H			MUL0H		—	√		
FF20H	Port mode register 0	PM0		R/W	√	√	—	FFH
FF21H	Port mode register 1	PM1			√	√	—	
FF22H	Port mode register 2	PM2			√	√	—	
FF23H	Port mode register 3	PM3			√	√	—	
FF32H	Pull-up resistor option register B2	PUB2			√	√	—	00H
FF33H	Pull-up resistor option register B3	PUB3			√	√	—	
FF42H	Watchdog timer clock selection register	WDCS			—	√	—	
FF4AH	Watch timer mode control register	WTM			√	√	—	
FF4BH	Watchdog timer mode register	WDTM			√	√	—	
FF53H	P3 function register	PF3			√	√	—	
FF54H	Remote control timer control register 50	TMC50			√	√	—	
FF55H	Remote control timer capture register 50	CP50		R	—	√	—	
FF56H	Remote control timer capture register 51	CP51			—	√	—	
FF57H	8-bit compare register 80	CR80		W	—	√	—	Undefined
FF58H	8-bit timer counter 80	TM80		R	—	√	—	00H
FF59H	8-bit timer mode control register 80	TMC80		R/W	√	√	—	
FF5AH	8-bit compare register 81	CR81		W	—	√	—	Undefined
FF5BH	8-bit timer counter 81	TM81		R	—	√	—	00H
FF5CH	8-bit timer mode control register 81	TMC81		R/W	√	√	—	
FF5DH	8-bit compare register 82	CR82		W	—	√	—	Undefined
FF5EH	8-bit timer counter 82	TM82		R	—	√	—	00H
FF5FH	8-bit timer mode control register 82	TMC82		R/W	√	√	—	
FF60H	8-bit compare register 30	CR30		W	—	√	—	Undefined
FF61H	8-bit timer counter 30	TM30		R	—	√	—	00H
FF62H	8-bit timer mode control register 30	TMC30		R/W	√	√	—	
FF63H	8-bit compare register 40	CR40		W	—	√	—	Undefined
FF64H	8-bit H width compare register 40	CRH40			—	√	—	

Notes 1. This symbol is for 16-bit access only.

2. 16-bit access is enabled for short direct addressing only.

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
FF65H	8-bit timer counter 40	TM40	R	—	√	—	00H
FF66H	8-bit timer mode control register 40	TMC40	R/W	√	√	—	
FF67H	Carrier generator output control register 40	TCA40	W	—	√	—	
FF68H	8-bit compare register SG0	CRSG0		—	√	—	Undefined
FF6AH	8-bit timer counter SG0	TMSG0	R	—	√	—	00H
FF6BH	8-bit timer mode control register SG0	TMCSG0	R/W	√	√	—	
FF6CH	Carrier generator output control register SG0	TCASG0	W	—	√	—	
FF6DH	Sound generator frequency setting register 00	SGFC00	R/W	—	√	—	
FF72H	Serial operation mode register 10	CSIM10		√	√	—	
FF74H	Transmission/reception shift register 10	SIO10	W	—	√	—	Undefined
			R	—	√	—	00H
FF80H	A/D converter mode register	ADM	R/W	√	√	—	
FF84H	A/D input selection register	ADS		√	√	—	
FFA0H	Asynchronous serial interface mode register 00	ASIM00		√	√	—	
FFA1H	Baud rate generator control register 00	BRGC00		—	√	—	
FFA2H	Transmit shift register 00	TXS00	W	—	√	—	Undefined
	Receive buffer register 00	RXB00	R	—	√	—	FFH
FFA3H	Asynchronous serial interface status register 00	ASIS00		√	√	—	00H
FFABH	Swapping function register 0	SWP0[Note]	W	—	√	—	Undefined
			R	—	√	—	00H
FFAFH	Power supply control register 0	PSC0	R/W	√	√	—	
FFB0H	LCD20 display mode register	LCDM20		√	√	—	
FFB2H	LCD20 clock control register	LCDC20		—	√	—	
FFB3H	LCD boost voltage level setting register 00	VLCD00		—	√	—	
FFD0H	Multiplication data register A0	MRA0	W	—	√	—	Undefined
FFD1H	Multiplication data register B0	MRB0		—	√	—	
FFD2H	Multiplier control register 0	MULC0	R/W	√	√	—	00H
FFE0H	Interrupt request flag register 0	IF0		√	√	—	
FFE1H	Interrupt request flag register 1	IF1		√	√	—	
FFE2H	Interrupt request flag register 2	IF2		√	√	—	
FFE4H	Interrupt mask flag register 0	MK0		√	√	—	FFH
FFE5H	Interrupt mask flag register 1	MK1		√	√	—	
FFE6H	Interrupt mask flag register 2	MK2		√	√	—	
FFECH	External interrupt mode register 0	INTM0		—	√	—	00H
FFF0H	Subsystem clock oscillation mode register	SCKM		√	√	—	
FFF2H	Subsystem clock control register	CSS		√	√	—	
FFF5H	Key return mode register 00	KRM00		√	√	—	
FFF7H	Pull-up resistor option register 0	PU0		√	√	—	
FFFAH	Oscillation stabilization time selection register	OSTS		—	√	—	04H
FFFBH	Processor clock control register	PCC		√	√	—	02H

Note The initial value in read mode differs from that in write mode. For details, refer to **CHAPTER 16 SWAPPING (SWAP)**.

3.3 Instruction Address Addressing

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

3.3.1 Relative addressing

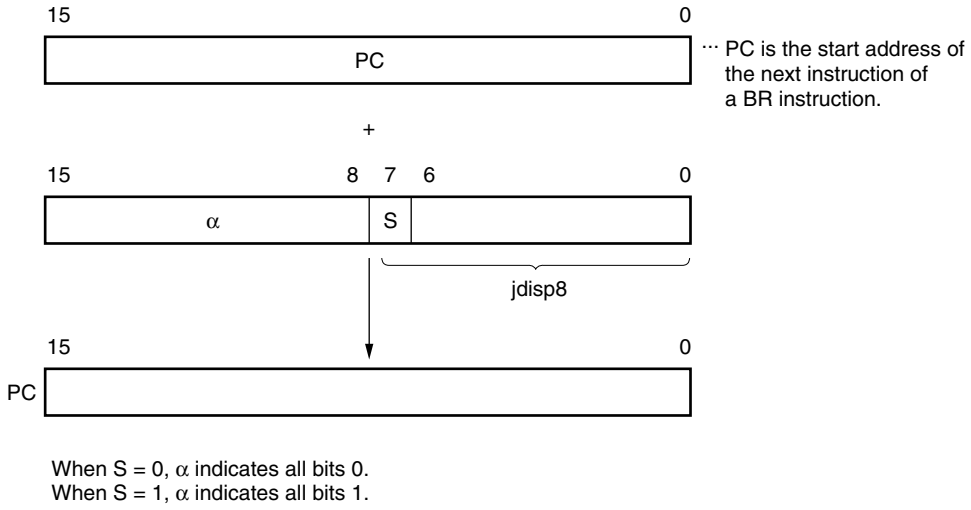
[Function]

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

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

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

[Illustration]



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

3.3.2 Immediate addressing

[Function]

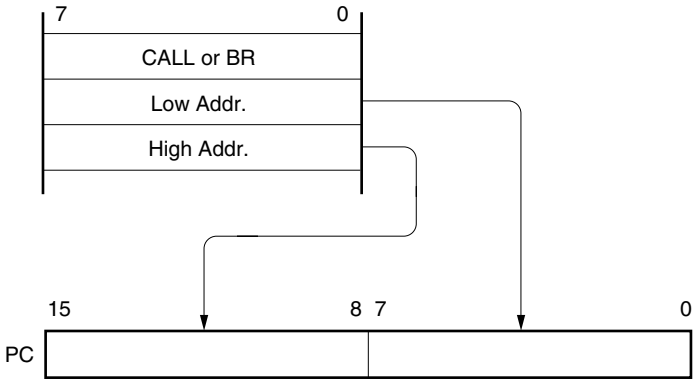
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

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

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

[Illustration]

In case of CALL !addr16 and BR !addr16 instructions



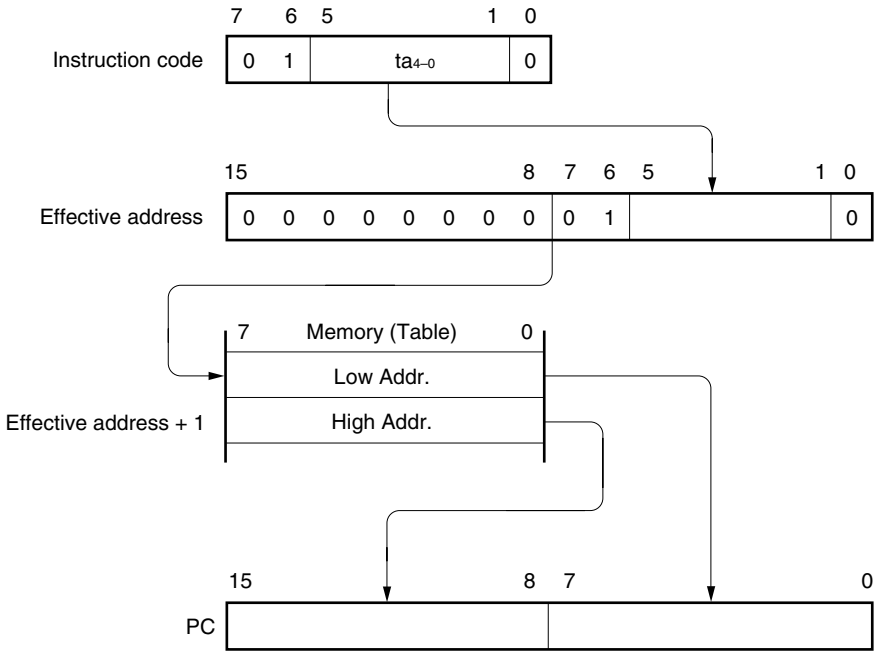
3.3.3 Table indirect addressing

[Function]

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

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

[Illustration]



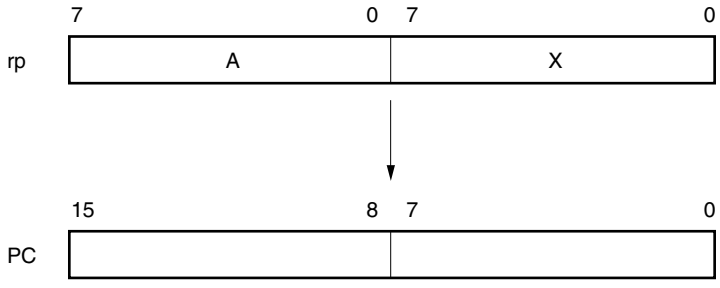
3.3.4 Register addressing

[Function]

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

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

[Illustration]



3.4 Operand Address Addressing

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

3.4.1 Direct addressing

[Function]

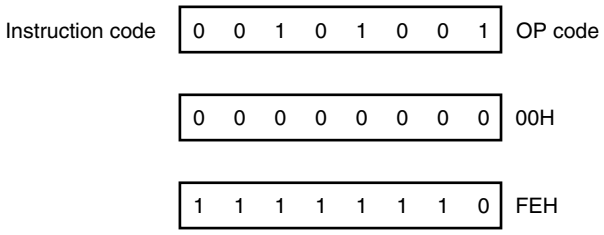
The memory indicated with immediate data in an instruction word is directly addressed.

[Operand format]

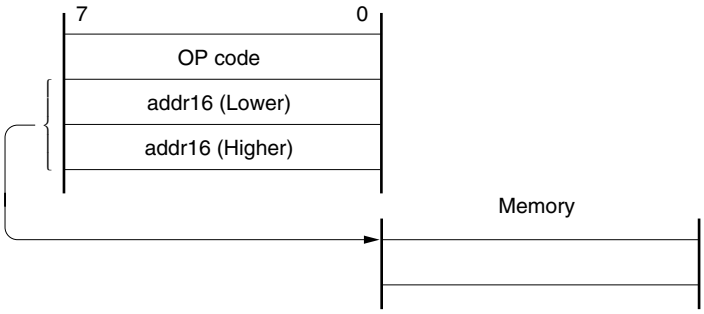
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

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



[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.

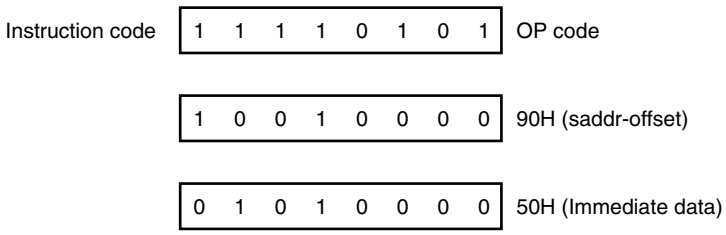
When 8-bit immediate data is at 20H to FFH, bit 8 of an effective address is set to 0. When it is at 00H to 1FH, bit 8 is set to 1. See **[Illustration]** below.

[Operand format]

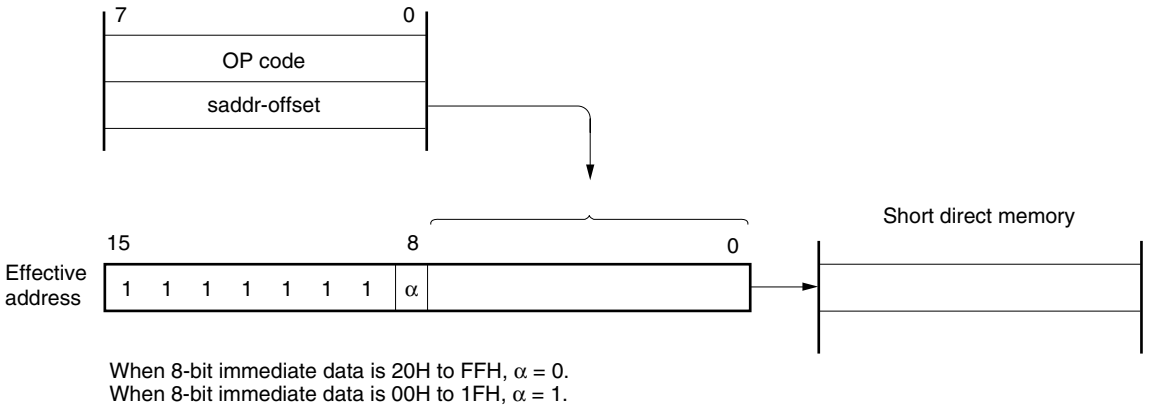
Identifier	Description
saddr	Label or FE20H to FF1FH immediate data
saddrp	Label or FE20H to FF1FH immediate data (even address only)

[Description example]

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



[Illustration]



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

3.4.3 Special function register (SFR) addressing

[Function]

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

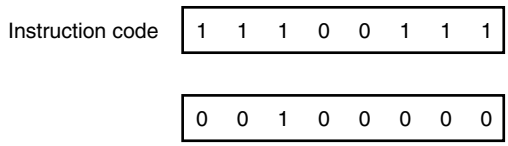
This addressing is applied to the 256-byte space FF00H to FFFFH. However, the SFRs mapped at FF00H to FF1FH can also be accessed with short direct addressing.

[Operand format]

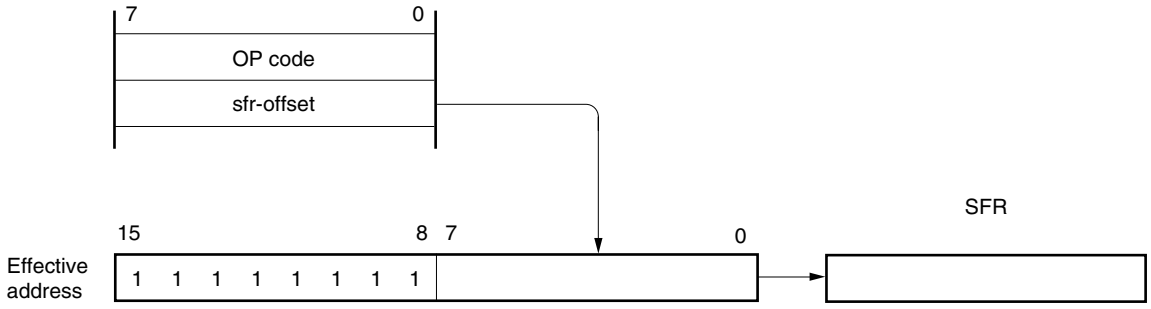
Identifier	Description
sfr	Special function register name

[Description example]

MOV PM0, A; When selecting PM0 for sfr



[Illustration]



3.4.4 Register addressing

[Function]

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

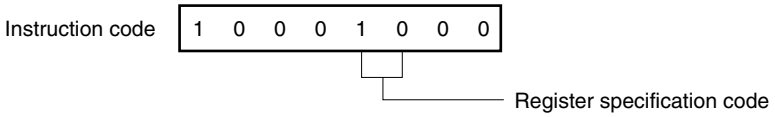
[Operand format]

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

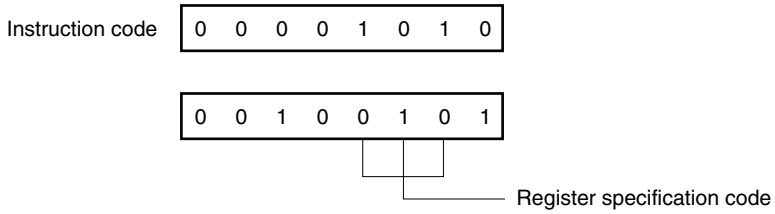
r and rp can be described with absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[Description example]

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



INCW DE; When selecting the DE register pair for rp



3.4.5 Register indirect addressing

[Function]

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

[Operand format]

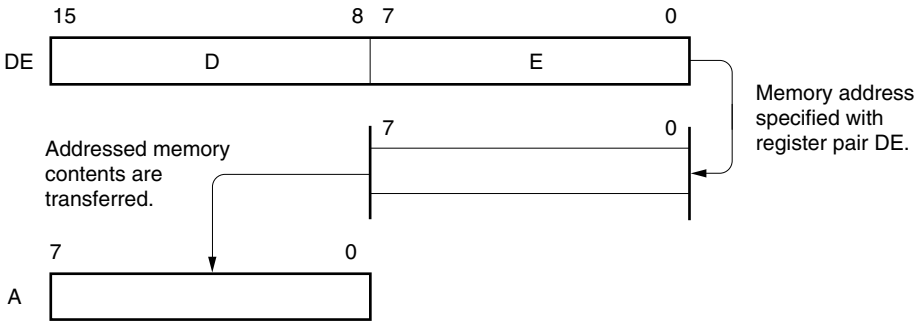
Identifier	Description
–	[DE], [HL]

[Description example]

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

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

[Illustration]



3.4.6 Based addressing

[Function]

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

[Operand format]

Identifier	Description
–	[HL+byte]

[Description example]

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

Instruction code

0	0	1	0	1	1	0	1

0	0	0	1	0	0	0	0

3.4.7 Stack addressing

[Function]

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

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

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

[Description example]

In the case of PUSH DE

Instruction code

1	0	1	0	1	0	1	0

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The μPD789835 Subseries provide the ports shown in Figure 4-1, enabling various methods of control. Table 4-1 shows the functions of the ports.

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

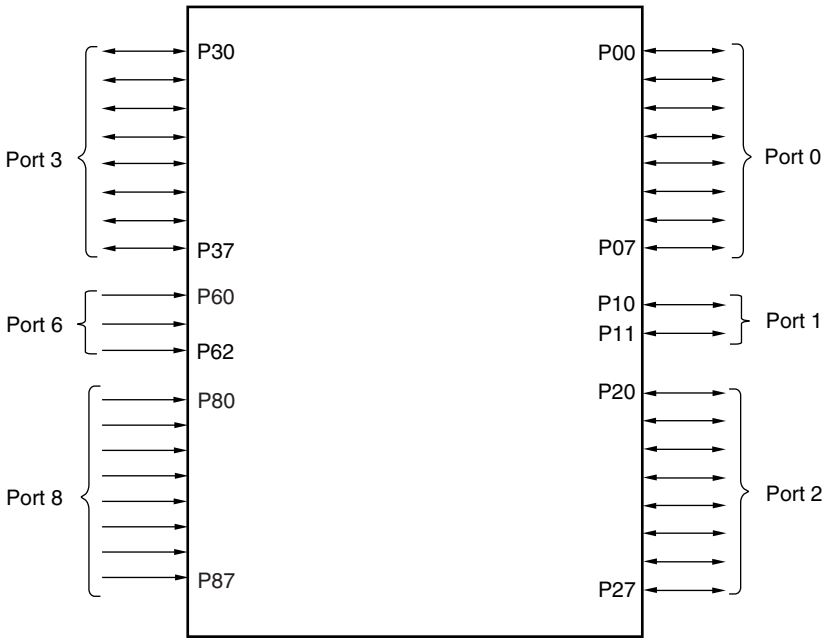


Table 4-1. Port Functions

Port Name	Pin Name	Function
Port 0	P00 to P07	This is an I/O port for which input and output can be specified in 1-bit units. When used as an input port, 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, on-chip pull-up resistors can be specified using pull-up resistor option register 0 (PU0).
Port 2	P20 to P27	This is an I/O port for which input and output can be specified in 1-bit units. When used as an input port, on-chip pull-up resistors can be specified using pull-up resistor option register B2 (PUB2).
Port 3	P30 to P37	This is an I/O port for which input and output can be specified in 1-bit units. When used as an input port, on-chip pull-up resistors can be specified using pull-up resistor option register B3 (PUB3).
Port 6	P60 to P62	This is an input port.
Port 8	P80 to P87	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 (PM0 to PM3) Pull-up resistor option registers (PU0, PUB2, PUB3)
Ports	Total: 37 (CMOS I/O: 26, CMOS input: 11)
Pull-up resistors	Total: 26 (software control: 26)

4.2.1 Port 0

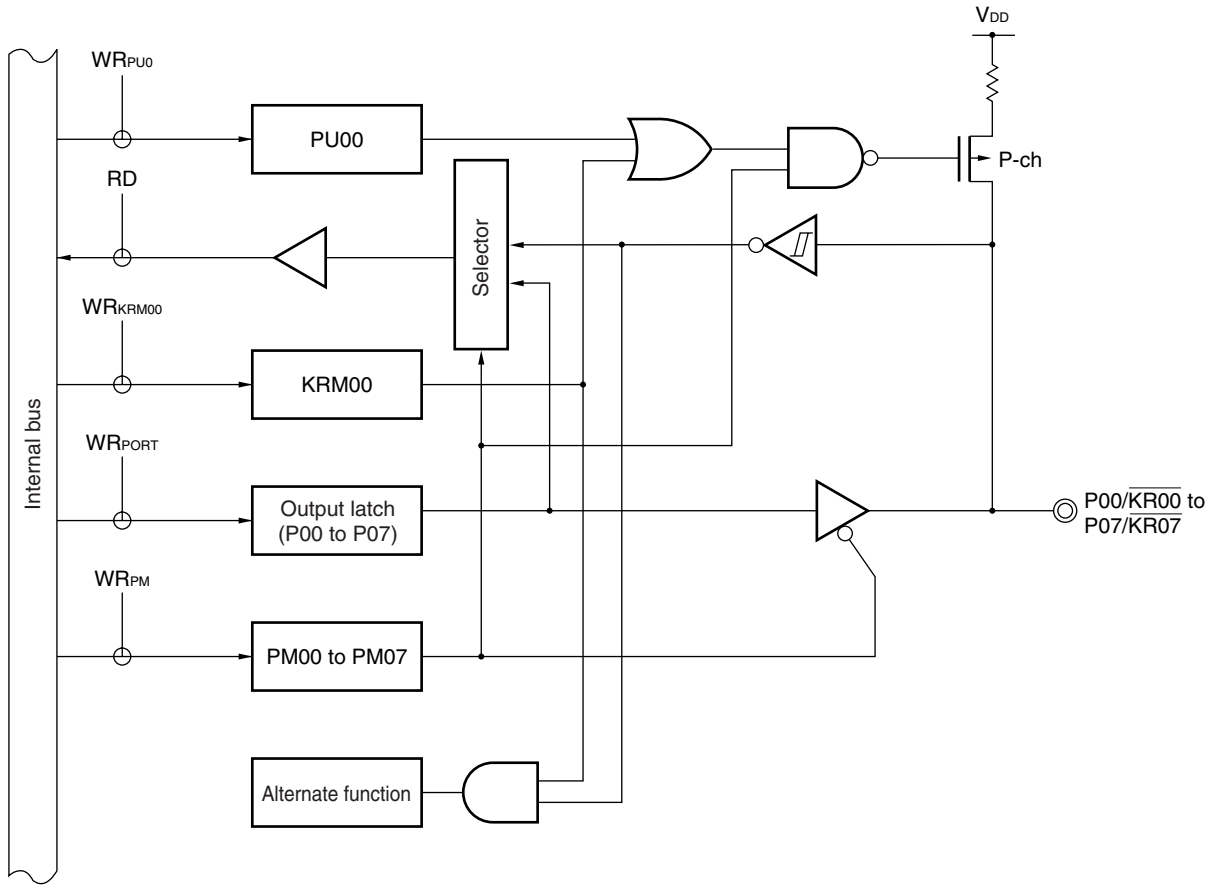
This is an 8-bit I/O port with an output latch. Port 0 can be specified in the input or output mode in 1-bit units by using the port mode register 0 (PM0). When the P00 to P07 pins are used as input port pins, on-chip pull-up resistors can be connected in 8-bit units by using pull-up resistor option register 0 (PU0).

This port is also used for key return input.

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 P07



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

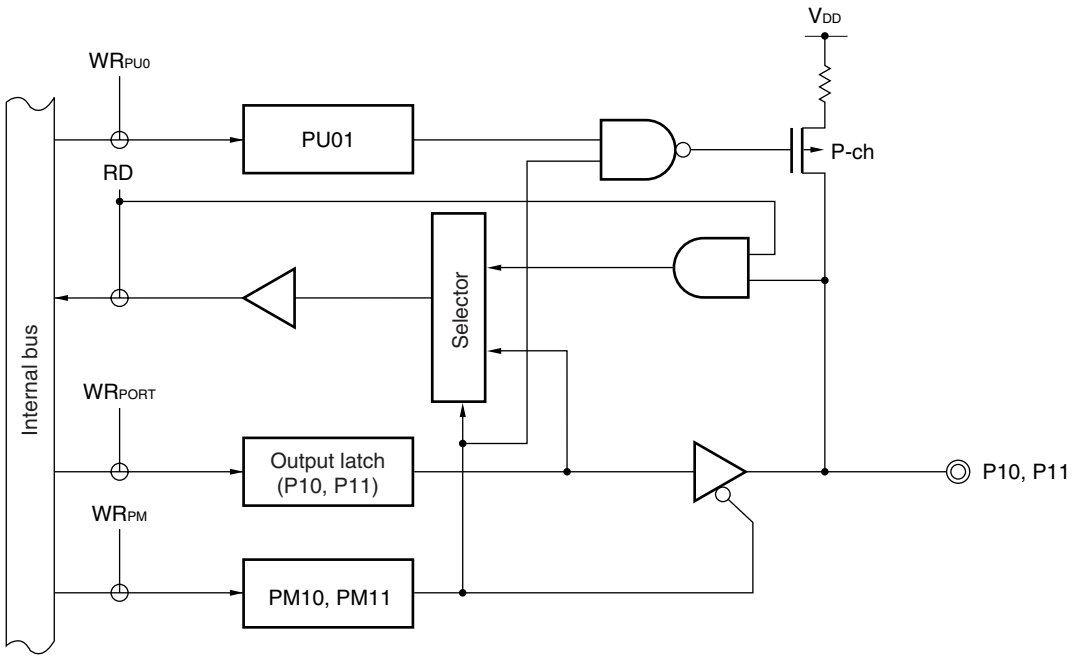
4.2.2 Port 1

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

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

Figure 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 2

This is an 8-bit I/O port with an output latch. Port 2 can be specified in the input or output mode in 1-bit units by using port mode register 2 (PM2). When using the P20 to P22 pins as input port pins, on-chip pull-up resistors can be connected in 1-bit units by using pull-up resistor option register B2 (PUB2).

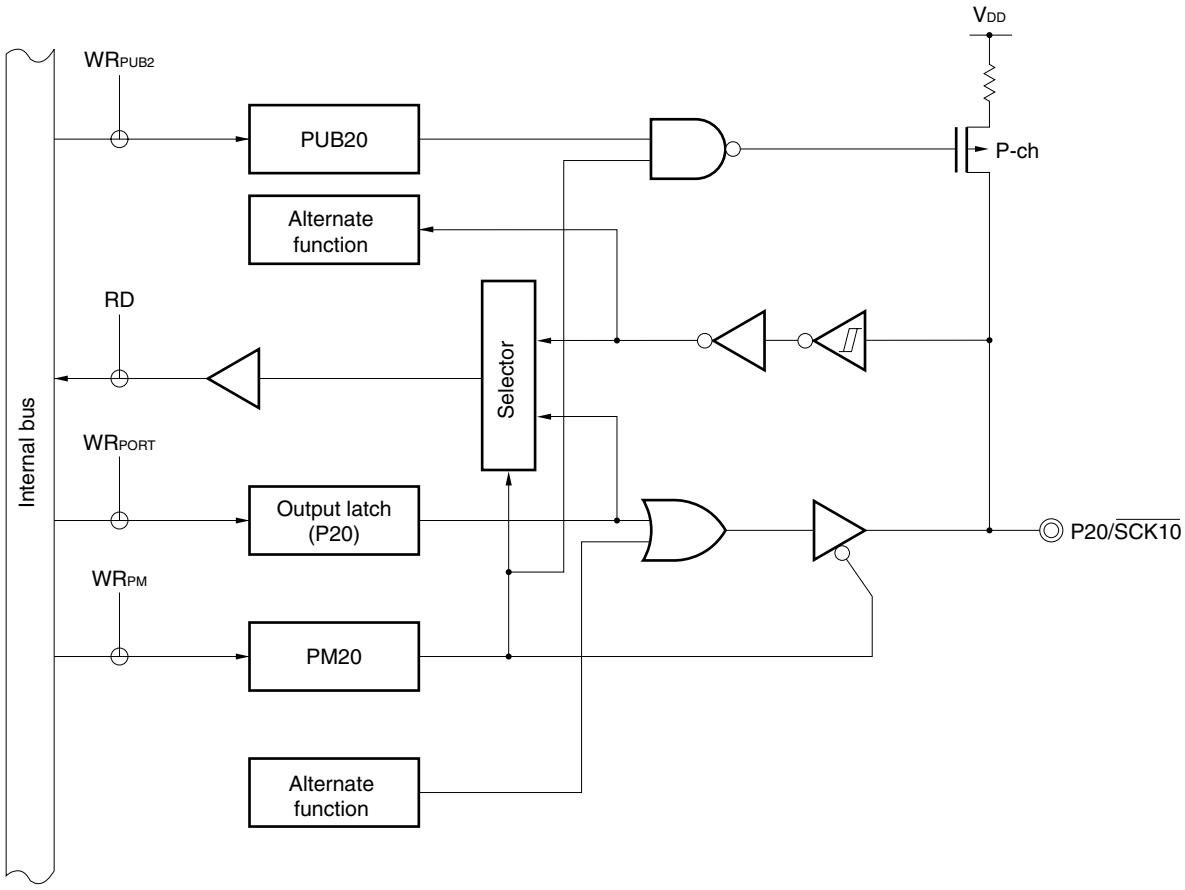
The port is also used as a data I/O to and from the serial interface, external interrupt input, and timer I/O.

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

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

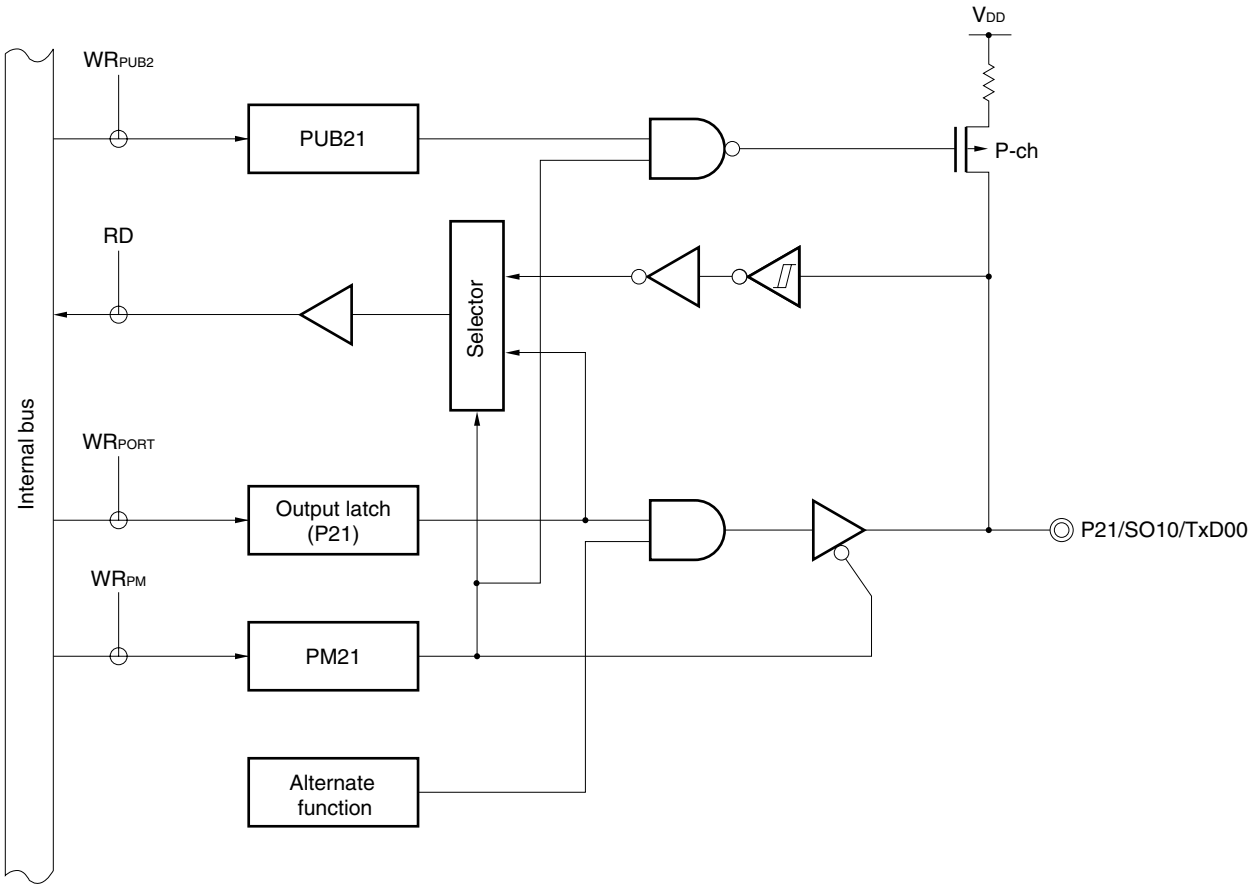
Caution When using the pins of port 2 as the serial interface, the I/O or output latch must be set according to the function to be used. For how to set the latches, see Table 13-2 Settings of Serial Interface Operating Mode.

Figure 4-4. Block Diagram of P20



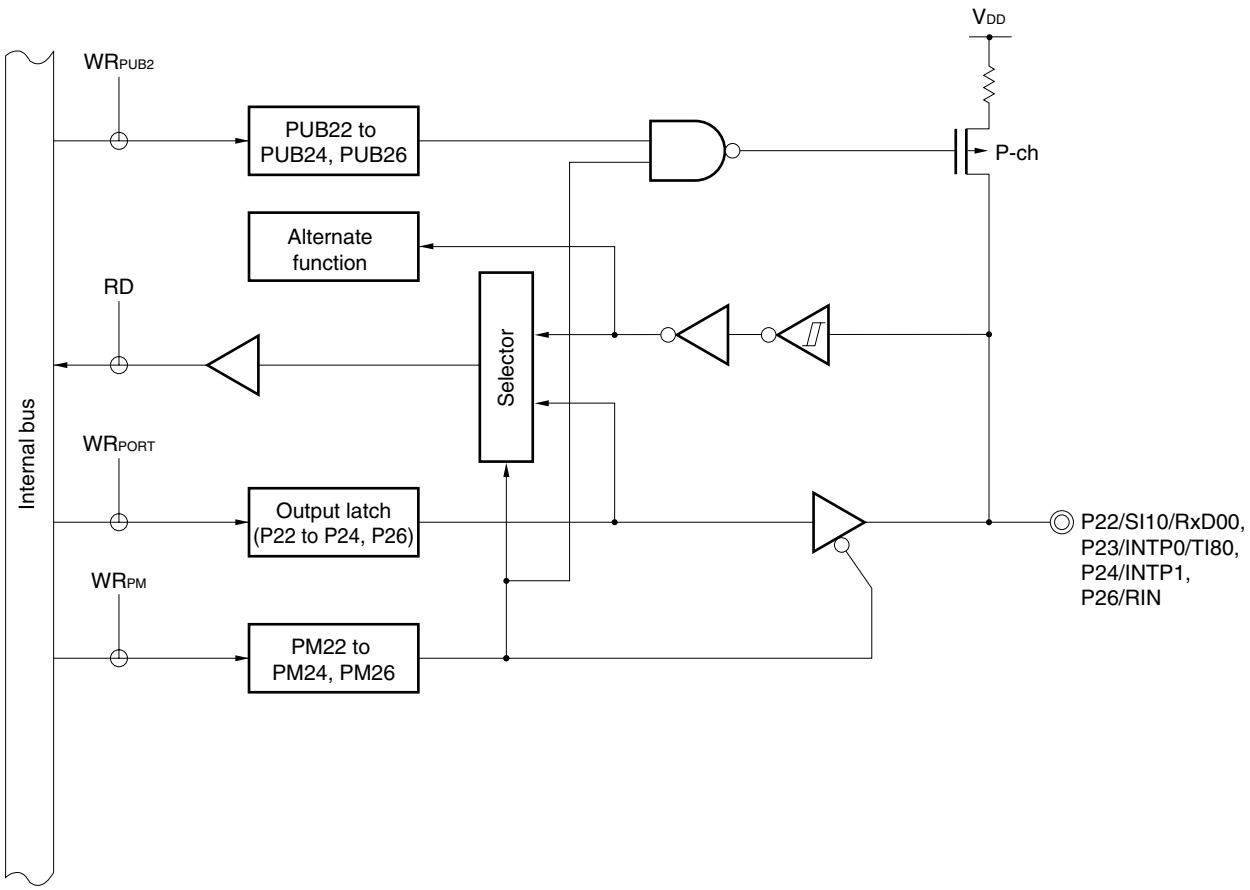
PUB2: Pull-up resistor option register B2
PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

Figure 4-5. Block Diagram of P21



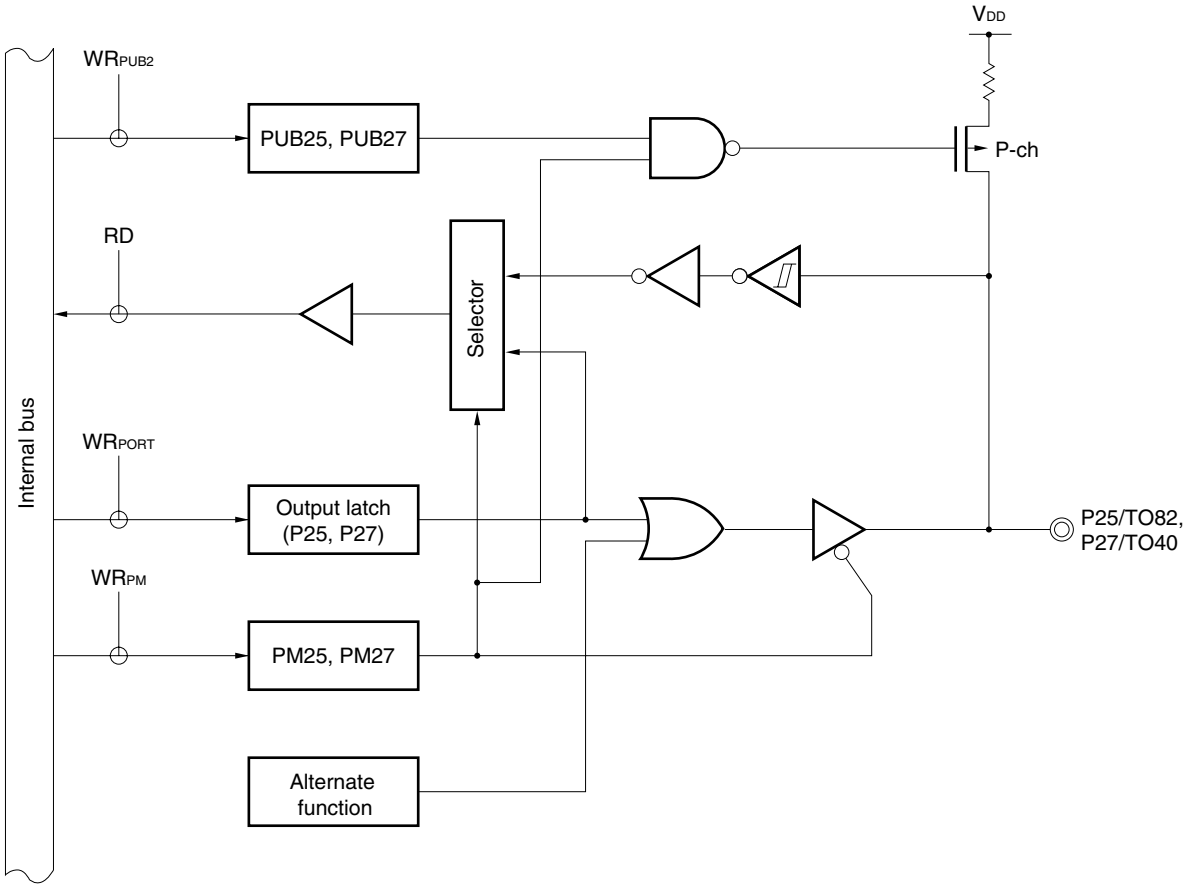
PUB2: Pull-up resistor option register B2
PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

Figure 4-6. Block Diagram of P22 to P24 and P26



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

Figure 4-7. Block Diagram of P25 and P27



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

4.2.4 Port 3

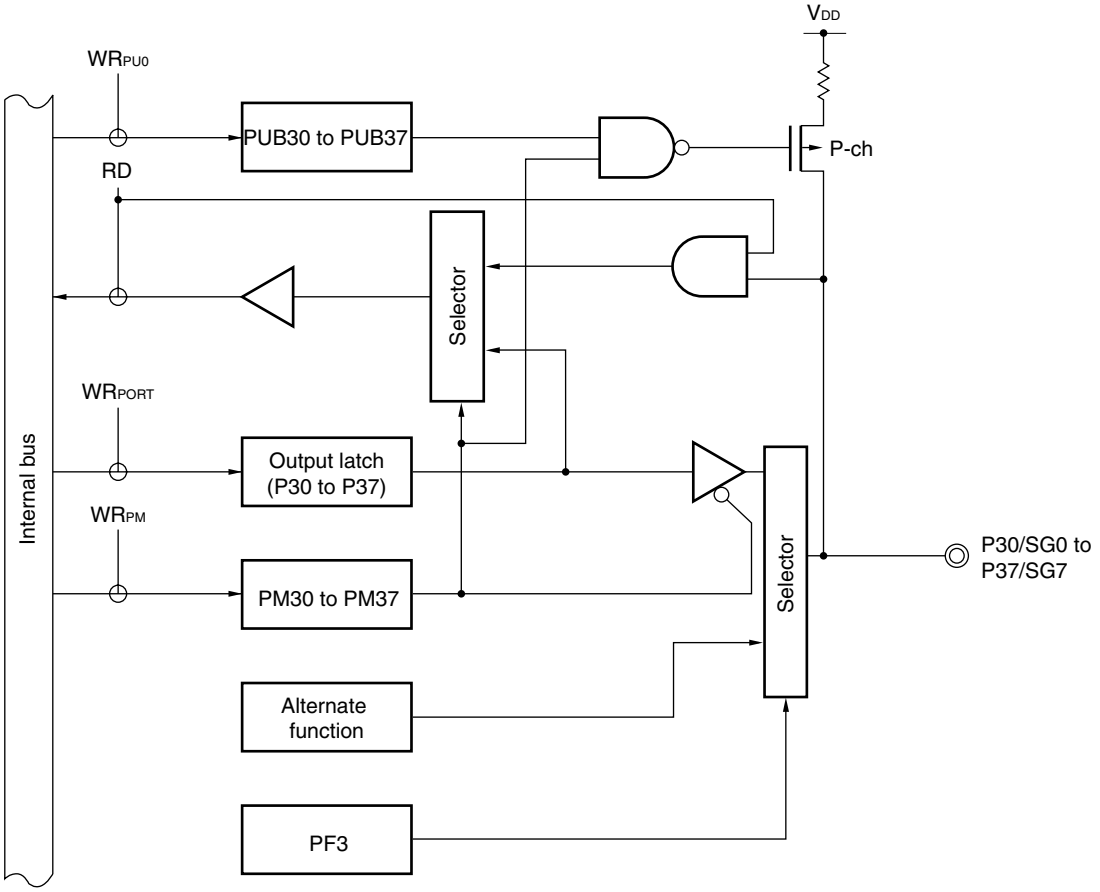
This is an 8-bit I/O port with an output latch. Only the bits specified for the port function by the P3 function register (PF3) can be used. Port 3 can be specified in the input or output mode in 1-bit units by using port mode register 3 (PM3). When using the P30 to P37 pins as input port pins, on-chip pull-up resistors can be connected in 1-bit units by using pull-up resistor option register B3 (PUB3).

This port is also used as a sound generator output.

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

Figure 4-8 shows block diagram of port 3.

Figure 4-8. Block Diagram of P30 to P37



PF3: P3 function register
PUB3: Pull-up resistor option register B3
PM: Port mode register
RD: Port 3 read signal
WR: Port 3 write signal

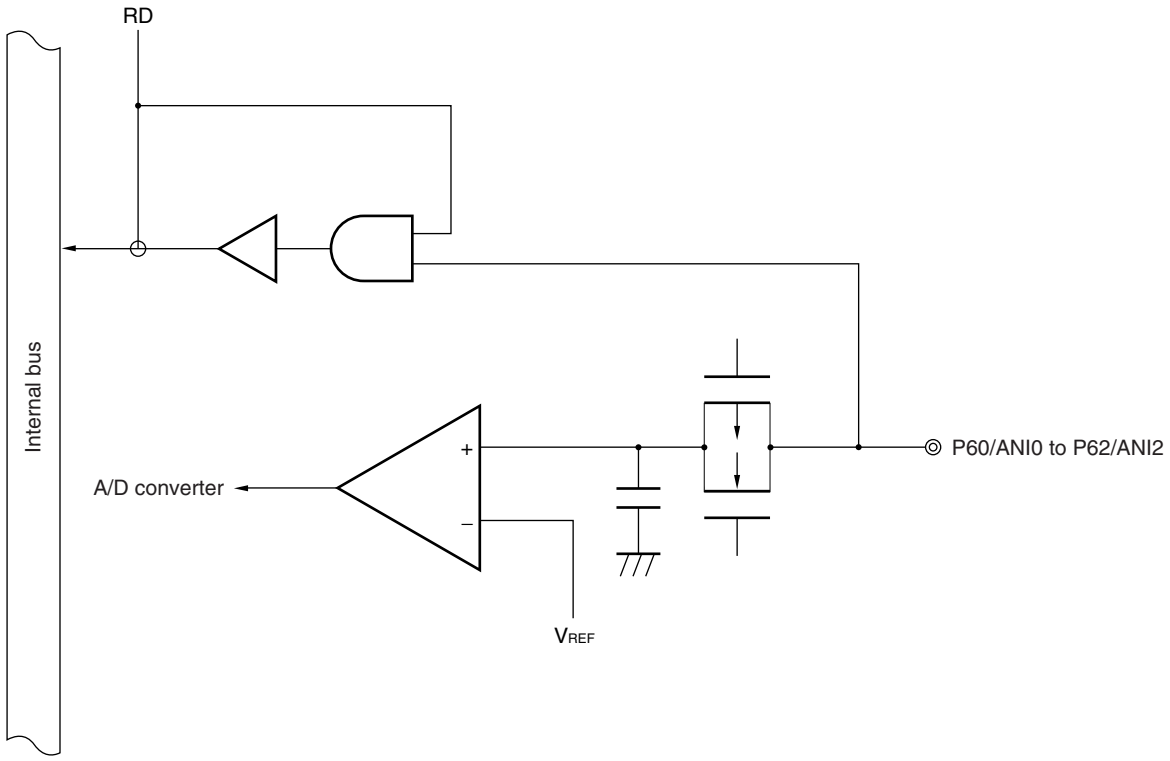
4.2.5 Port 6

This is a 3-bit input-only port.

This port is also used as the analog input of an A/D converter.

Figure 4-9 shows a block diagram of port 6.

Figure 4-9. Block Diagram of P60 to P62



RD: Port 6 read signal

4.2.6 Port 8

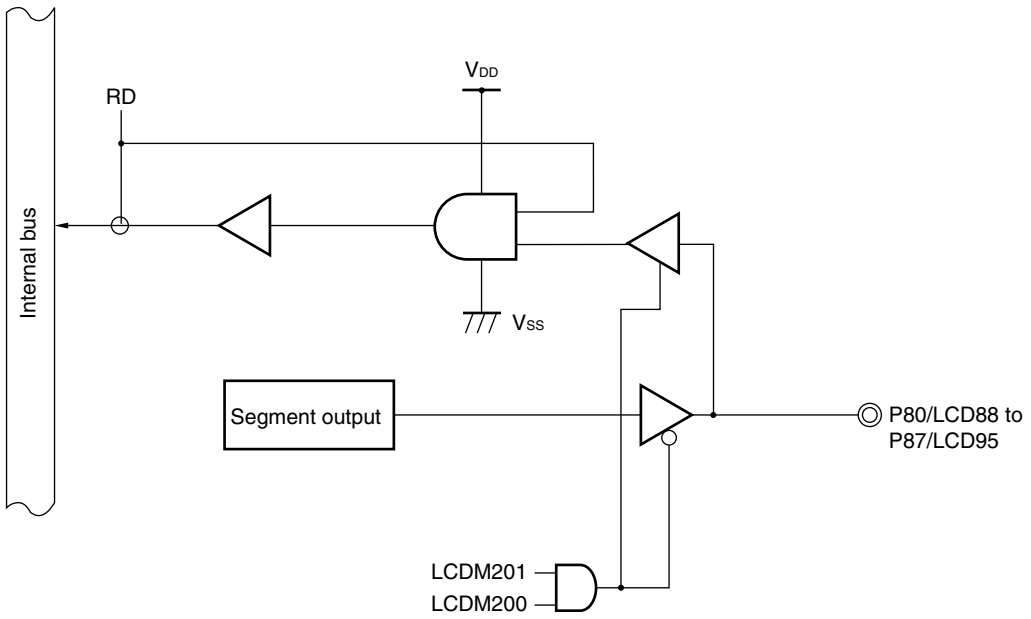
This is an 8-bit input-only port.

This port is also used for segment output. When used for segment output, P80/LCD88 to P87/LCD95 become output-only pins.

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

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

Figure 4-10. Block Diagram of P80 to P87



LCDM201: Bit 1 of the LCD20 display mode register (LCDM20)

LCDM200: Bit 0 of the LCD20 display mode register (LCDM20)

RD: Port 8 read signal

4.3 Registers Controlling Port Function

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

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

(1) Port mode registers (PM0 to PM3)

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 Port 2 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 4-11. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FF20H	FFH	R/W
PM1	1	1	1	1	1	1	PM11	PM10	FF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W

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

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

Pin Name	Alternate Function		PM××	P××
	Name	I/O		
P00 to P07	$\overline{KR00}$ to $\overline{KR07}$	Input	1	×
P23	INTP0	Input	1	×
	TI80	Input	1	×
P24	INTP1	Input	1	×
P25	TO82	Output	0	0
P26	RIN	Input	1	×
P27	TO40	Output	0	0
P30 to P37	SG0 to SG7[Note 1]	Output	—	—
P60 to P62	ANI0 to ANI2	Input	—	—
P80 to P87	LCD88 to LCD95[Note 2]	Output	—	—

Notes 1. These are set by the P3 function register (PF3) when used as alternate-function pins. For details of these settings, see **Figure 9-5 Format of P3 Function Register**.

2. These are set by the LCD20 display mode register (LCDM20) when used as alternate-function pins. For details of these settings, see **Figure 14-2 Format of LCD20 Display Mode Register**.

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

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

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

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

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

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

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

Figure 4-12. Format of Pull-Up Resistor Option Register 0

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

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

Caution Bits 2 to 7 must be set to 0.

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

This register is used to set whether or not to use an on-chip pull-up resistor for P20 to P27. At the pins where use of a pull-up resistor is specified by PUB2, an on-chip pull-up resistor can be used only for bits that have been set to input mode. An on-chip pull-up resistor cannot be used for any bit that has been set to output mode, regardless of the PUB2 setting. This is also true for pins used as output pins for alternate functions.

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

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

Figure 4-13. Format of Pull-up Resistor Option Register B2

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
PUB2	PUB27	PUB26	PUB25	PUB24	PUB23	PUB22	PUB21	PUB20	FF32H	00H	R/W

PUB2n	Selection of On-chip Pull-up Resistor for P2n (n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

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

This register is used to set whether or not to use an on-chip pull-up resistor for P30 to P37. At the pins where use of a pull-up resistor is specified by PUB3, an on-chip pull-up resistor can be used only for bits that have been set to input mode. An on-chip pull-up resistor cannot be used for any bit that has been set to output mode, regardless of the PUB3 setting. This is also true for pins used as output pins for alternate functions.

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

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

Figure 4-14. Format of Pull-up Resistor Option Register B3

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
PUB3	PUB37	PUB36	PUB35	PUB34	PUB33	PUB32	PUB31	PUB30	FF33H	00H	R/W

PUB3n	Selection of On-chip Pull-up Resistor for P3n (n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

4.4 Port Function Operation

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

4.4.1 Writing to I/O port

(1) In output mode

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

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

(2) In input mode

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

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

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

4.4.2 Reading from I/O port

(1) In output mode

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

(2) In input mode

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

4.4.3 Arithmetic operation of I/O port

(1) In output mode

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

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

(2) In input mode

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

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

CHAPTER 5 CLOCK GENERATOR

5.1 Clock Generator Functions

The clock generator generates the clock to be supplied to the CPU and peripheral hardware.
The following three 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).
- **Main system clock (RC) oscillator**
  This circuit oscillates at 0.4 to 2.0 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).

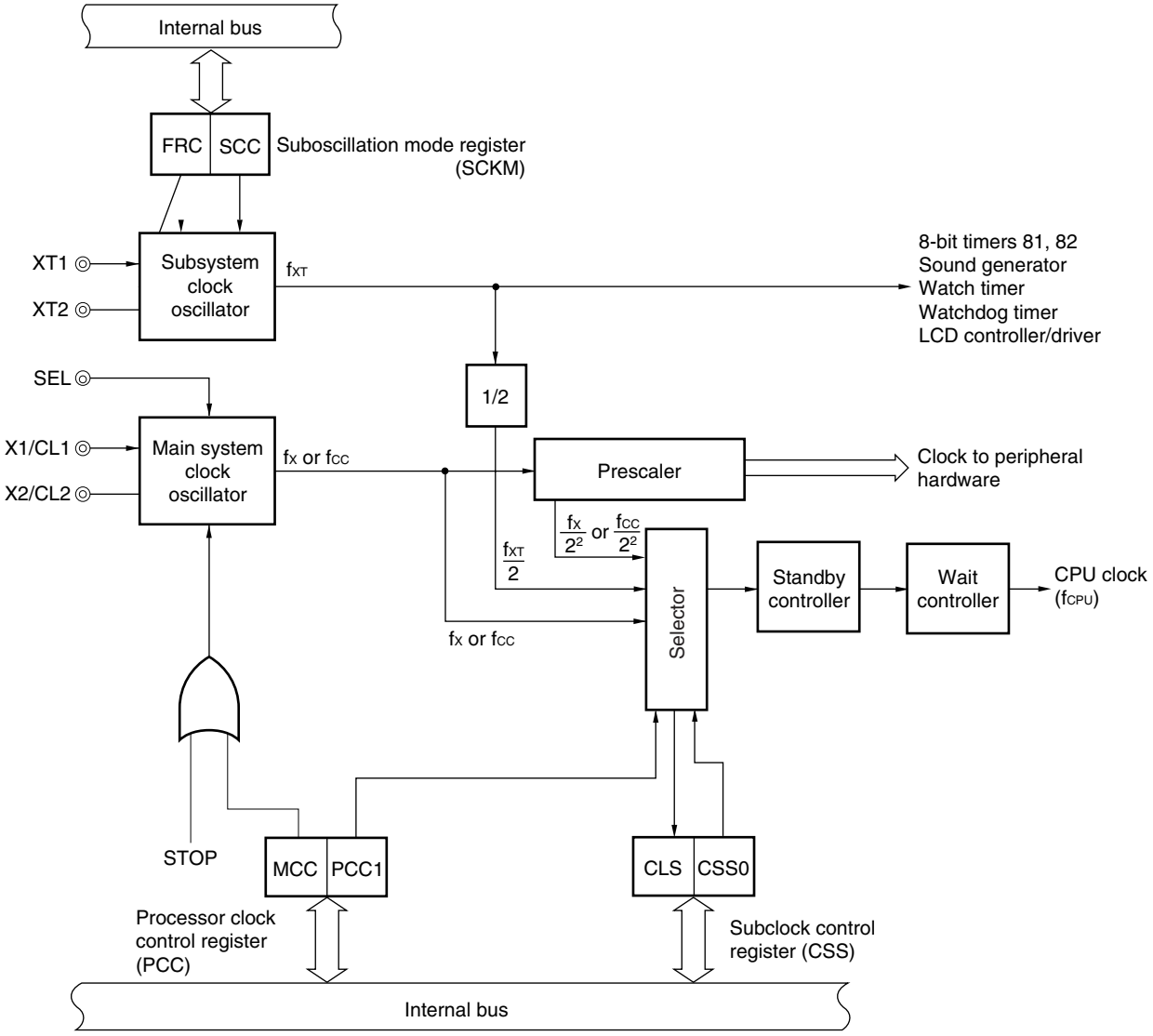
5.2 Clock Generator Configuration

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



Caution Connect the SEL pin to V_{SS} when these pins are used as the X1 and X2 pins (ceramic/crystal oscillation (f_X)), and to V_{DD} when used as the CL1 and CL2 pins (RC oscillation (fCC)).

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	Selection of CPU clock (f_{CPU})[Note]	
		@ f_X = 5.0 MHz and f_{XT} = 32.768 kHz operation	@ f_{CC} = 2.0 MHz and f_{XT} = 32.768 kHz operation
0	0	f_X (0.2 μs)	f_{CC} (0.5 μs)
0	1	$f_X/2^2$ (0.8 μs)	$f_{CC}/2^2$ (2.0 μs)
1	0	$f_{XT}/2$ (61 μs)	
1	1		

Note The CPU clock is selected according to a combination of the PCC and CSS flags (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.

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

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

3. f_{XT}: Subsystem clock oscillation frequency

CPU clock (f_{CPU}) × 2 indicates the minimum instruction execution time. The following table shows minimum instruction execution time based on each setting value.

CSS0	PCC1	Minimum instruction execution time	
		@ f_X = 5.0 MHz and f_{XT} = 32.768 kHz operation	@ f_{CC} = 2.0 MHz and f_{XT} = 32.768 kHz operation
0	0	0.4 μs	1.0 μs
0	1	1.6 μs	4.0 μs
1	0	122 μs	
1	1		

(2) Suboscillation mode register (SCKM)

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

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

$\overline{\text{RESET}}$ input 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

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

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

Caution Bits 2 to 7 must be set to 0.

(3) Subclock control register (CSS)

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

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

$\overline{\text{RESET}}$ input 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

There are two types of system clock oscillators: the main system clock oscillator and the subsystem clock oscillator.

With the μPD789835 Subseries, the main system clock oscillator can be switched between ceramic/crystal oscillation and RC oscillation. Switch between ceramic/crystal oscillation and RC oscillation using the SEL pin. When using ceramic/crystal oscillation, connect the SEL pin to V_{SS}. When using RC oscillation, connect it to V_{DD}.

5.4.1 Main system clock oscillator (crystal/ceramic oscillation)

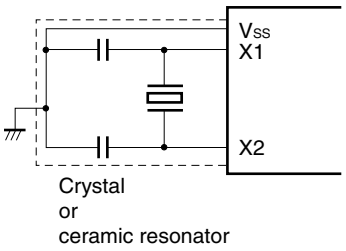
This oscillator is oscillated by the crystal or ceramic resonator (5.0 MHz TYP.) connected across the X1 and X2 pins.

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.

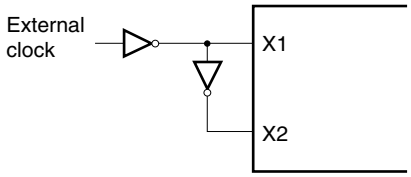
Figure 5-5 shows the external circuit of the main system clock oscillator.

Figure 5-5. External Circuit of Main System Clock Oscillator (Ceramic/Crystal Oscillation)

(a) Crystal or ceramic oscillation



(b) External clock



5.4.2 Main system clock oscillator (RC oscillation)

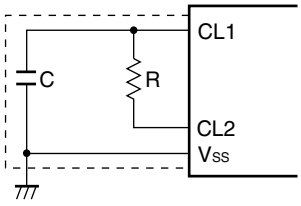
This oscillator is oscillated by the resistor (R) and capacitor (C) (2.0 MHz TYP.) connected across the CL1 and CL2 pins.

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

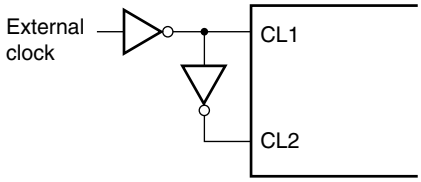
Figure 5-6 shows the external circuit of the main system clock oscillator.

Figure 5-6. External Circuit of Main System Clock Oscillator (RC Oscillation)

(a) RC oscillation



(b) External clock



5.4.3 Subsystem clock oscillator

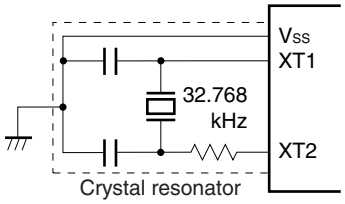
The subsystem clock oscillator is oscillated by the crystal resonator (32.768 kHz TYP.) connected across the XT1 and XT2 pins.

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.

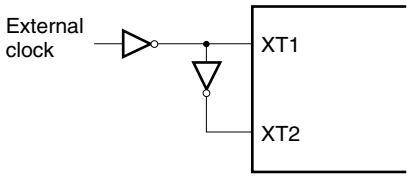
Figure 5-7 shows the external circuit of the subsystem clock oscillator.

Figure 5-7. 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 to 5-7 to avoid an adverse effect from wiring capacitance.

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

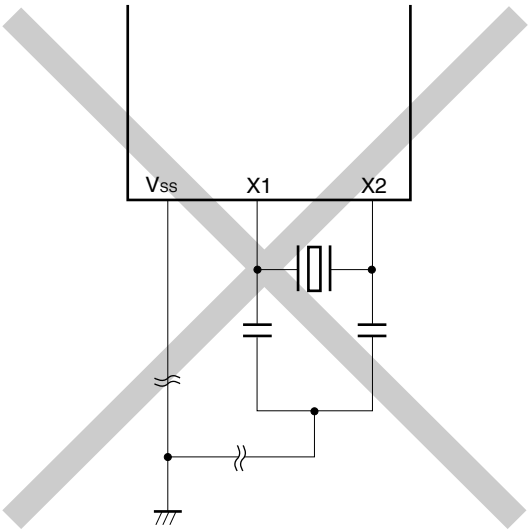
When using the subsystem clock, particular care is required because the subsystem clock oscillator is designed as a low-amplitude circuit for reducing current consumption.

5.4.4 Example of incorrect resonator connection

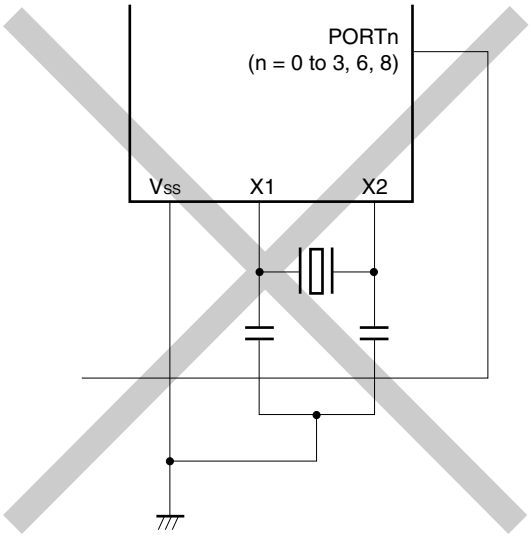
Figure 5-8 shows an example of incorrect connection for ceramic/crystal oscillation and Figure 5-9 shows an example for RC oscillation.

Figure 5-8. Examples of Incorrect Connection for Ceramic/Crystal Oscillation (1/2)

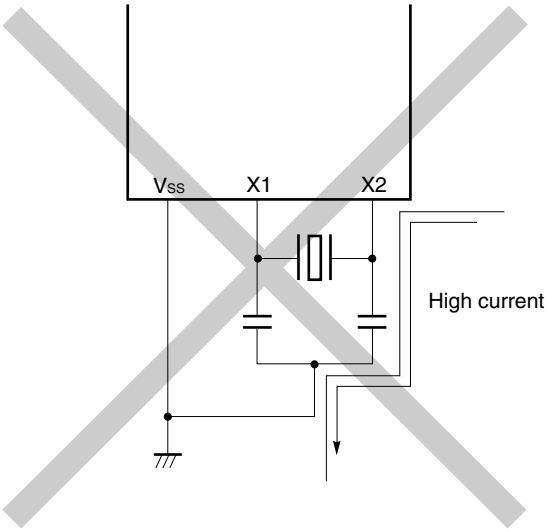
(a) Too long wiring



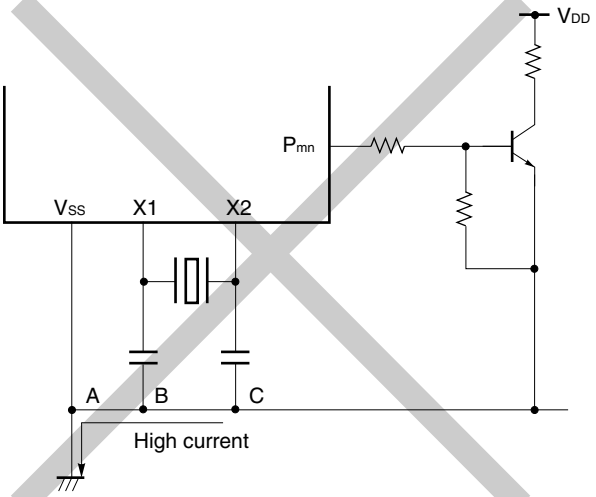
(b) Crossed signal line



(c) Wiring near high fluctuating current



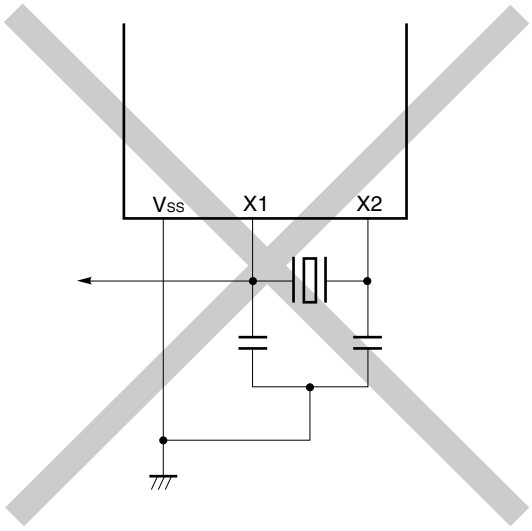
(d) Current flowing through ground line of oscillator (potential at points A, B, and C fluctuates)



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

Figure 5-8. Examples of Incorrect Connection for Ceramic/Crystal Oscillation (2/2)

(e) Signal is fetched

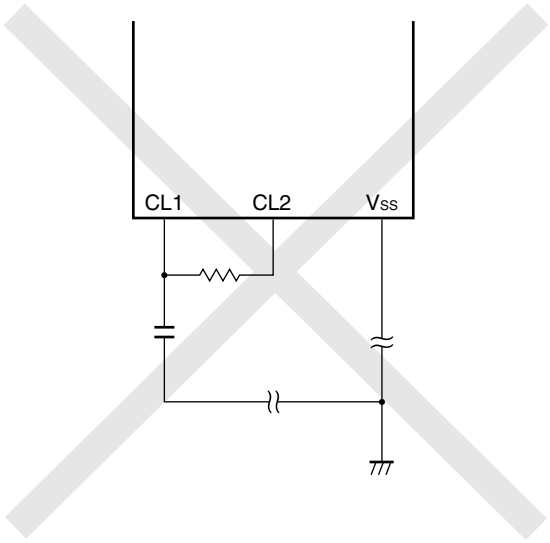


Remark When using the subsystem clock, read X1 and X2 as XT1 and XT2, respectively, and connect a resistor to the XT2 in series.

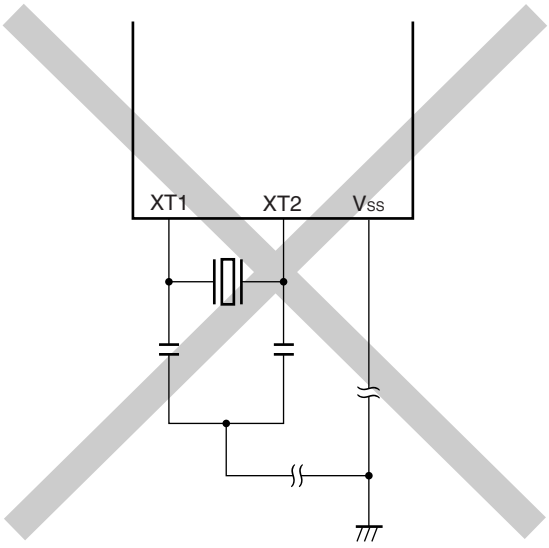
Figure 5-9. Examples of Incorrect Connection for RC Oscillation (1/3)

(a) Too long wiring

- **Main system clock**

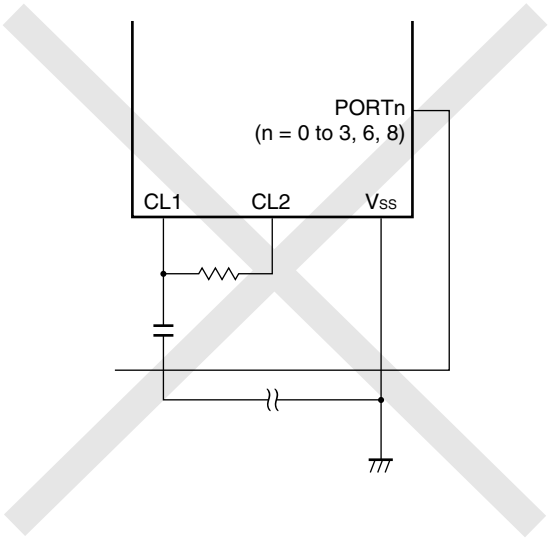


- **Subsystem clock**



(b) Crossed signal line

- **Main system clock**



- **Subsystem clock**

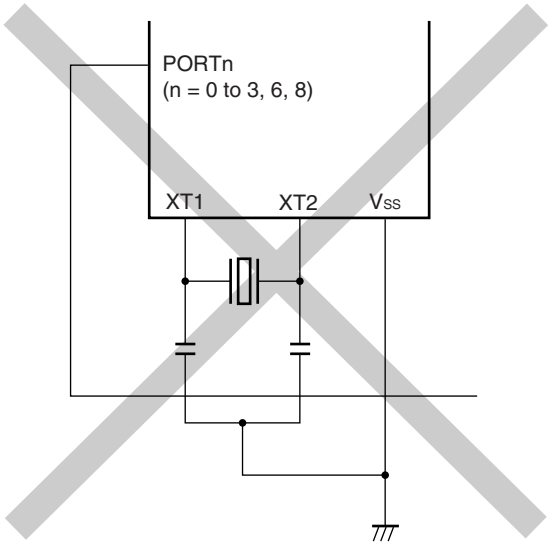
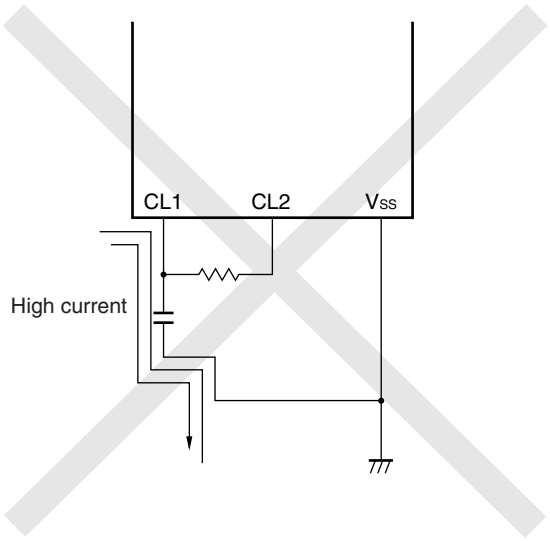


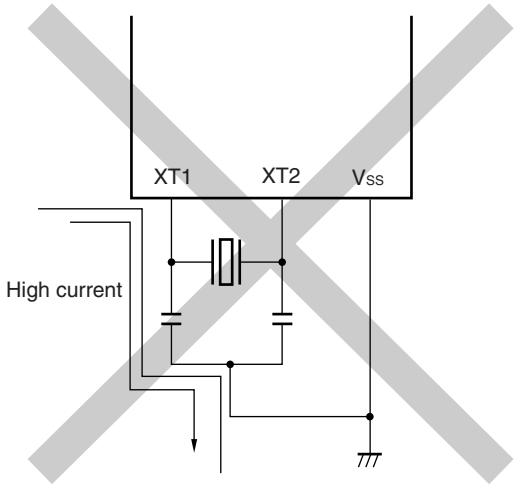
Figure 5-9. Examples of Incorrect Connection for RC Oscillation (2/3)

(c) Wiring near high fluctuating current

- **Main system clock**

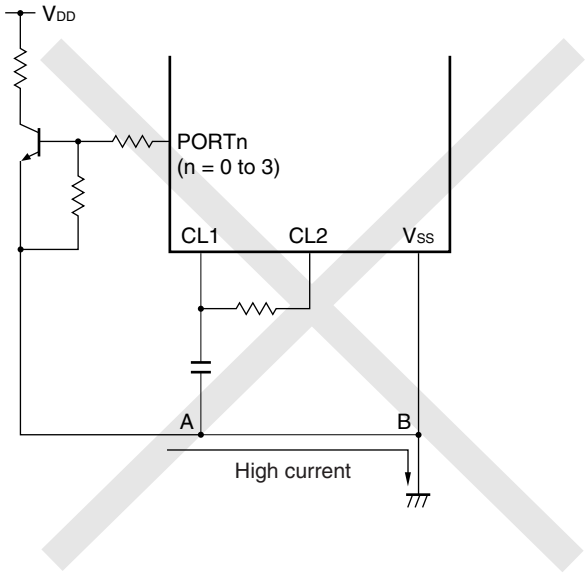


- **Subsystem clock**



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

- **Main system clock**



- **Subsystem clock**

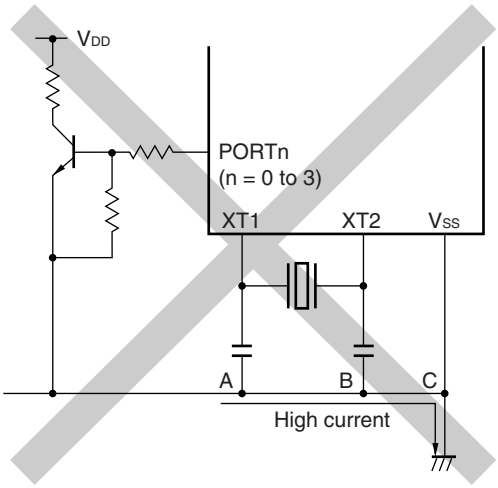
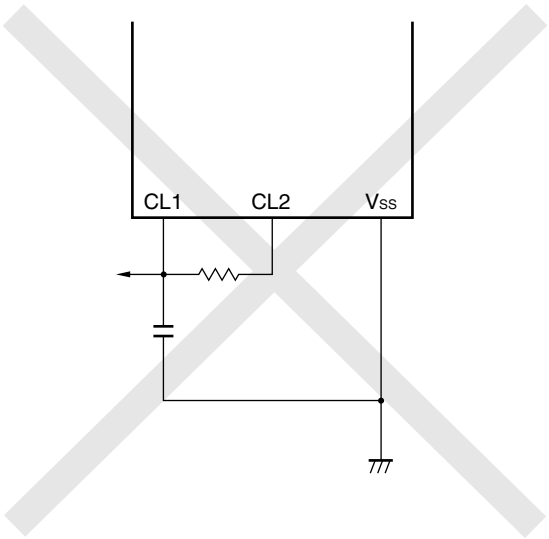


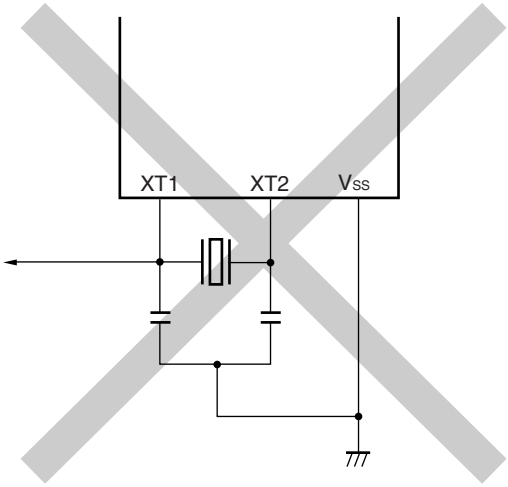
Figure 5-9. Examples of Incorrect Connection for RC Oscillation (3/3)

(e) Signal is fetched

- **Main system clock**



- **Subsystem clock**



5.4.5 Divider

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

5.4.6 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 or f_{CC}
- Subsystem clock f_{XT}
- CPU clock f_{CPU}
- Clock to peripheral hardware

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

(a) The low-speed mode $2f_{CPU}$ of the main system clock is selected when the $\overline{\text{RESET}}$ signal is generated (PCC = 02H). While a low level is input to the $\overline{\text{RESET}}$ pin, oscillation of the main system clock is stopped.

(b) Three types of CPU clocks f_{CPU} (for details, see **Figure 5-2 Format of Processor Clock Control Register**) can be selected by the PCC, SCKM, and CSS settings.

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

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

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

(f) The clock pulse for the peripheral hardware is generated by dividing the frequency of the main system clock, but the subsystem clock pulse is only supplied to the 8-bit timers 81, 82, sound generator, watch timer, watchdog timer, and LCD controller/driver. The 8-bit timers 81, 82, sound generator, watch timer, watchdog 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).

Actually, the specified clock is not selected immediately after the setting of PCC has been changed, and the old clock is used for the duration of several instructions after that (see **Tables 5-2** and **5-3**).

Table 5-2. Maximum Time Required for Switching CPU Clock (When Ceramic/Crystal Oscillation Is Selected)

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

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

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

3. ×: Don't care

Table 5-3. Maximum Time Required for Switching CPU Clock (When RC Oscillation Is Selected)

Set Value Before Switching		Set Value After Switching					
CSS0	PCC1	CSS0	PCC1	CSS0	PCC1	CSS0	PCC1
		0	0	0	1	1	×
0	0			4 clocks		$2f_{CC}/f_{XT}$ clocks (122 clocks)	
	1	2 clocks				$f_{CC}/2f_{XT}$ clocks (31 clocks)	
1	×	2 clocks		2 clocks			

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

2. The parenthesized values apply to operation at f_{CC} = 2.0 MHz or f_{XT} = 32.768 kHz.

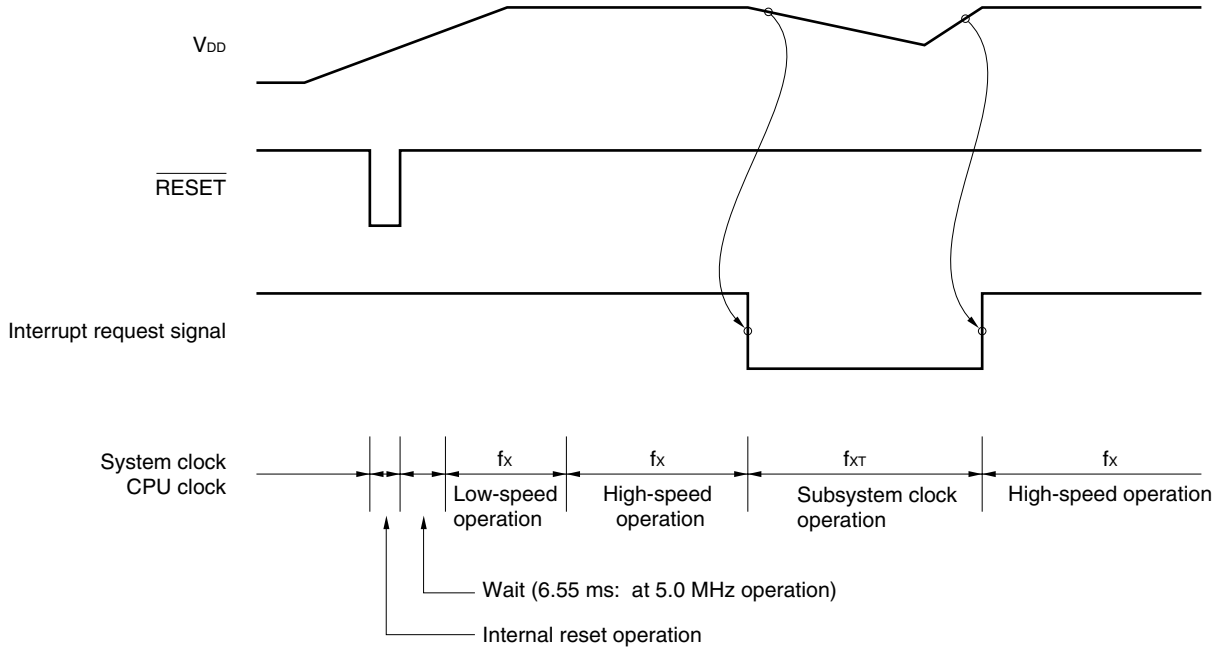
3. ×: Don't care

5.6.2 Switching between system clock and CPU clock

(1) When ceramic/crystal oscillation is selected

The following describes switching between the system clock and CPU clock when ceramic/crystal oscillation is selected for the main system clock.

Figure 5-10. Switching Between System Clock and CPU Clock (Ceramic/Crystal Oscillation)



<1> The CPU is reset when the $\overline{\text{RESET}}$ pin is made low on power application. The effect of resetting is released when the $\overline{\text{RESET}}$ pin is later made high, and the main system clock starts oscillating. At this time, the oscillation stabilization time (2^{15}/fx) is automatically secured.

After that, the CPU starts instruction execution at the slow speed of the main system clock (1.6 μs: at 5.0 MHz operation).

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

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

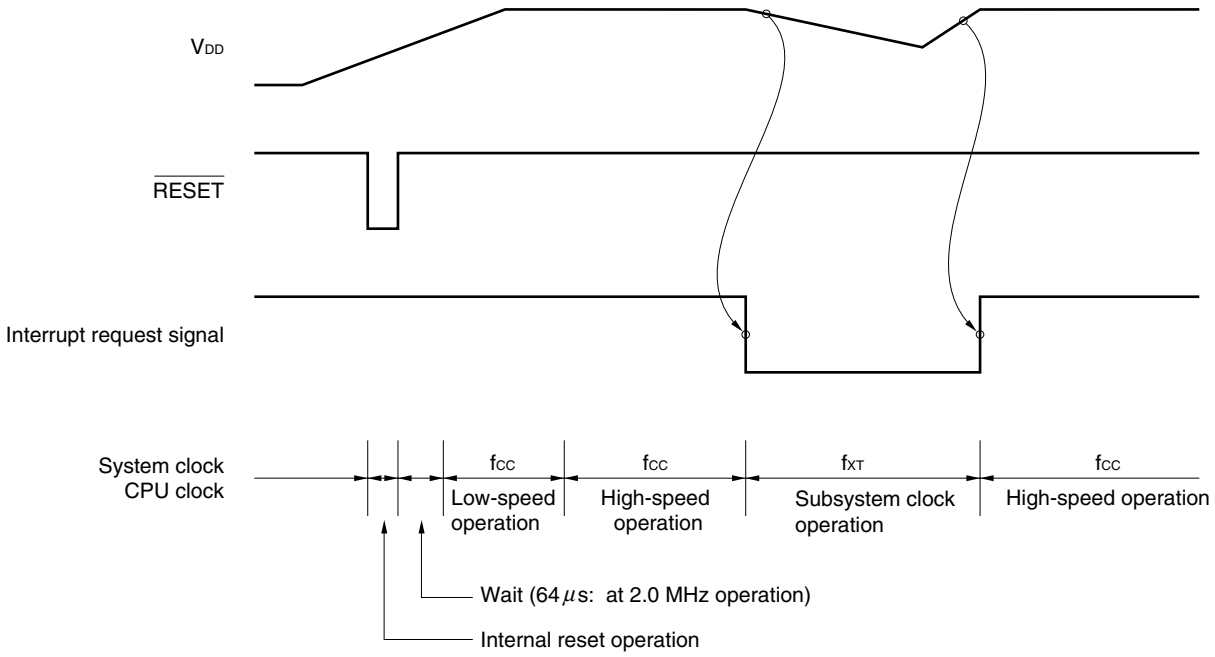
<4> A recover of the V_{DD} voltage is detected with an interrupt request signal. Bit 7 (MCC) of PCC is set to 0, and then the main system clock starts oscillating. After the time required for the oscillation to stabilize has elapsed, PCC1 and CSS0 are rewritten so that high-speed operation can be selected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

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

(2) When RC oscillation is selected

The following describes switching between the system clock and CPU clock when RC oscillation is selected for the main system clock.

Figure 5-11. Switching Between System Clock and CPU Clock (RC Oscillation)



<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 (2^7/fCC) is automatically secured.

After that, the CPU starts instruction execution at the slow speed of the main system clock (4.0 μs: at 2.0 MHz operation).

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

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

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

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

CHAPTER 6 8-BIT TIMER/EVENT COUNTERS 80 to 82

6.1 8-Bit Timer/Event Counters 80 to 82 Functions

The 8-bit timer/event counters (timer 80, timer 81, and timer 82) have the following functions:

- Interval timers (timer 80, timer 81, and timer 82)
- External event counter (timer 80 only)
- Square wave output (timer 82 only)

The μPD789835 Subseries is provided with a channel (timer 80) of an 8-bit timer/event counter and two channels (timer 81 and timer 82) of an 8-bit timer. When reading the description of timer 81 and timer 82, an 8-bit timer/event counter should be read as referring to an 8-bit timer.

(1) 8-bit interval timer

When the 8-bit timer/event counter is used as an interval timer, it generates an interrupt at any time intervals set in advance.

Table 6-1. Interval Time of 8-Bit Timer/Event Counter 80

	Minimum Interval Time	Maximum Interval Time	Resolution
@ f_X = 5.0 MHz operation	$2^6/f_X$ (12.8 μs)	$2^{14}/f_X$ (3.28 ms)	$2^6/f_X$ (12.8 μs)
	$2^9/f_X$ (102 μs)	$2^{17}/f_X$ (26.2 ms)	$2^9/f_X$ (102 μs)
@ f_{CC} = 2.0 MHz operation	$2^6/f_{CC}$ (32 μs)	$2^{14}/f_{CC}$ (8.19 ms)	$2^6/f_{CC}$ (32 μs)
	$2^9/f_{CC}$ (256 μs)	$2^{17}/f_{CC}$ (65.5 ms)	$2^9/f_{CC}$ (256 μs)

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

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

Table 6-2. Interval Time of 8-Bit Timer 81

	Minimum Interval Time	Maximum Interval Time	Resolution
@ f_X = 5.0 MHz and f_{XT} = 32.768 kHz operation	$2^4/f_X$ (3.2 μs)	$2^{12}/f_X$ (819 μs)	$2^4/f_X$ (3.2 μs)
	$2^8/f_X$ (51.2 μs)	$2^{16}/f_X$ (13.1 ms)	$2^8/f_X$ (51.2 μs)
	$1/f_{XT}$ (30.5 μs)	$2^8/f_{XT}$ (7.81 ms)	$1/f_{XT}$ (30.5 μs)
@ f_{CC} = 2.0 MHz and f_{XT} = 32.768 kHz operation	$2^4/f_{CC}$ (8.0 μs)	$2^{12}/f_{CC}$ (2.05 ms)	$2^4/f_{CC}$ (8.0 μs)
	$2^8/f_{CC}$ (128 μs)	$2^{16}/f_{CC}$ (32.8 ms)	$2^8/f_{CC}$ (128 μs)
	$1/f_{XT}$ (30.5 μs)	$2^8/f_{XT}$ (7.81 ms)	$1/f_{XT}$ (30.5 μs)

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

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

3. f_{XT}: Subsystem clock oscillation frequency

Table 6-3. Interval Time of 8-Bit Timer 82

	Minimum Interval Time	Maximum Interval Time	Resolution
@ f_X = 5.0 MHz and f_{XT} = 32.768 kHz operation	$2^3/f_X$ (1.6 μs)	$2^{11}/f_X$ (410 μs)	$2^3/f_X$ (1.6 μs)
	$2^{10}/f_X$ (205 μs)	$2^{18}/f_X$ (52.4 ms)	$2^{10}/f_X$ (205 μs)
	$1/f_{XT}$ (30.5 μs)	$2^8/f_{XT}$ (7.81 ms)	$1/f_{XT}$ (30.5 μs)
@ f_{CC} = 2.0 MHz and f_{XT} = 32.768 kHz operation	$2^3/f_{CC}$ (4.0 μs)	$2^{11}/f_{CC}$ (1.02 ms)	$2^3/f_{CC}$ (4.0 μs)
	$2^{10}/f_{CC}$ (512 μs)	$2^{18}/f_{CC}$ (131 ms)	$2^{10}/f_{CC}$ (512 μs)
	$1/f_{XT}$ (30.5 μs)	$2^8/f_{XT}$ (7.81 ms)	$1/f_{XT}$ (30.5 μs)

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

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

3. f_{XT}: Subsystem clock oscillation frequency

(2) External event counter

The number of pulses of an externally input signal can be measured.

(3) Square wave output

A square wave of any frequency can be output.

Table 6-4. Square Wave Output Range of 8-Bit Timer 82

	Minimum Interval Time	Maximum Interval Time	Resolution
@ f_X = 5.0 MHz and f_{XT} = 32.768 kHz operation	$2^3/f_X$ (1.6 μs)	$2^{11}/f_X$ (410 μs)	$2^3/f_X$ (1.6 μs)
	$2^{10}/f_X$ (205 μs)	$2^{18}/f_X$ (52.4 ms)	$2^{10}/f_X$ (205 μs)
	$1/f_{XT}$ (30.5 μs)	$2^8/f_{XT}$ (7.81 ms)	$1/f_{XT}$ (30.5 μs)
@ f_{CC} = 2.0 MHz and f_{XT} = 32.768 kHz operation	$2^3/f_{CC}$ (4.0 μs)	$2^{11}/f_{CC}$ (1.02 ms)	$2^3/f_{CC}$ (4.0 μs)
	$2^{10}/f_{CC}$ (512 μs)	$2^{18}/f_{CC}$ (131 ms)	$2^{10}/f_{CC}$ (512 μs)
	$1/f_{XT}$ (30.5 μs)	$2^8/f_{XT}$ (7.81 ms)	$1/f_{XT}$ (30.5 μs)

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

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

3. f_{XT}: Subsystem clock oscillation frequency

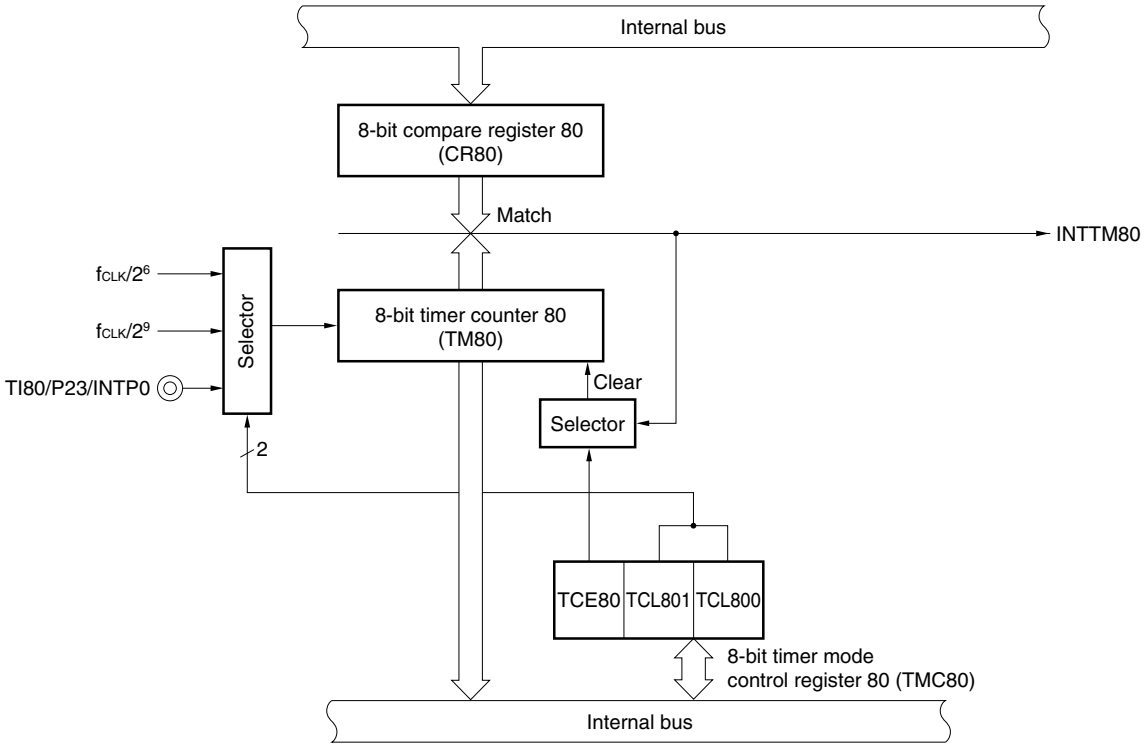
6.2 8-Bit Timer/Event Counters 80 to 82 Configuration

The 8-bit timer/event counters 80 to 82 consist of the following hardware configuration.

Table 6-5. Configuration of 8-Bit Timer/Event Counters 80 to 82

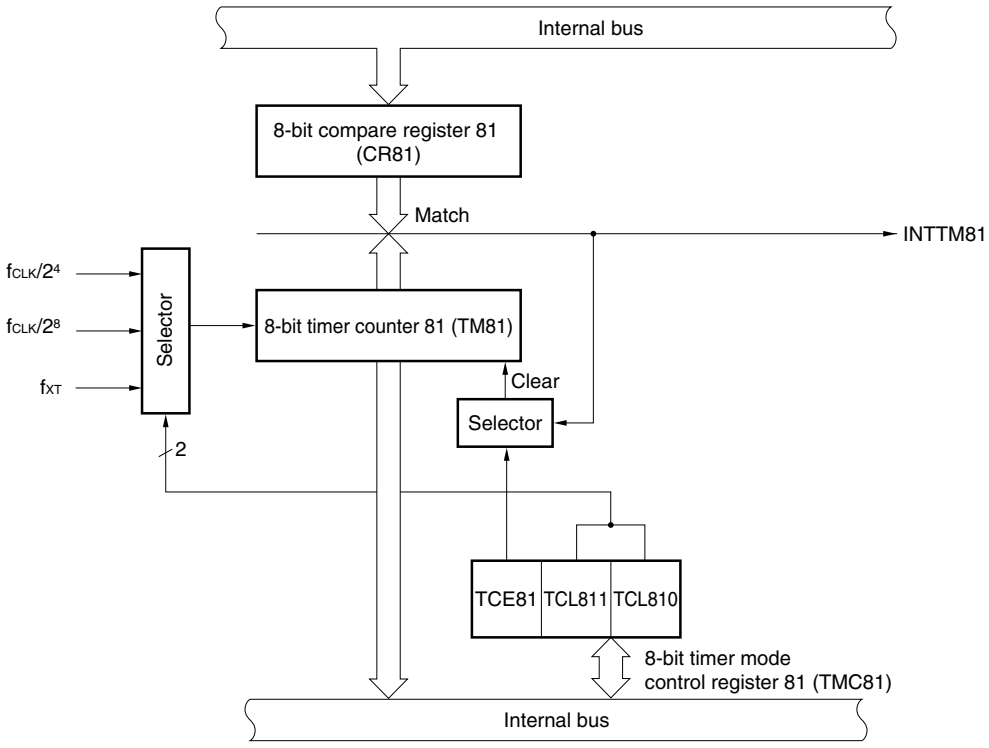
Item	Configuration
Timer counter	8 bits × 3 (TM80, TM81, and TM82)
Register	Compare register: 8 bits × 3 (CR80, CR81, and CR82)
Timer output	1 (TO82)
Control register	8-bit timer mode control registers 80, 81, and 82 (TMC80, TMC81, and TMC82) Port mode register 2 (PM2)

Figure 6-1. Block Diagram of 8-Bit Timer/Event Counter 80



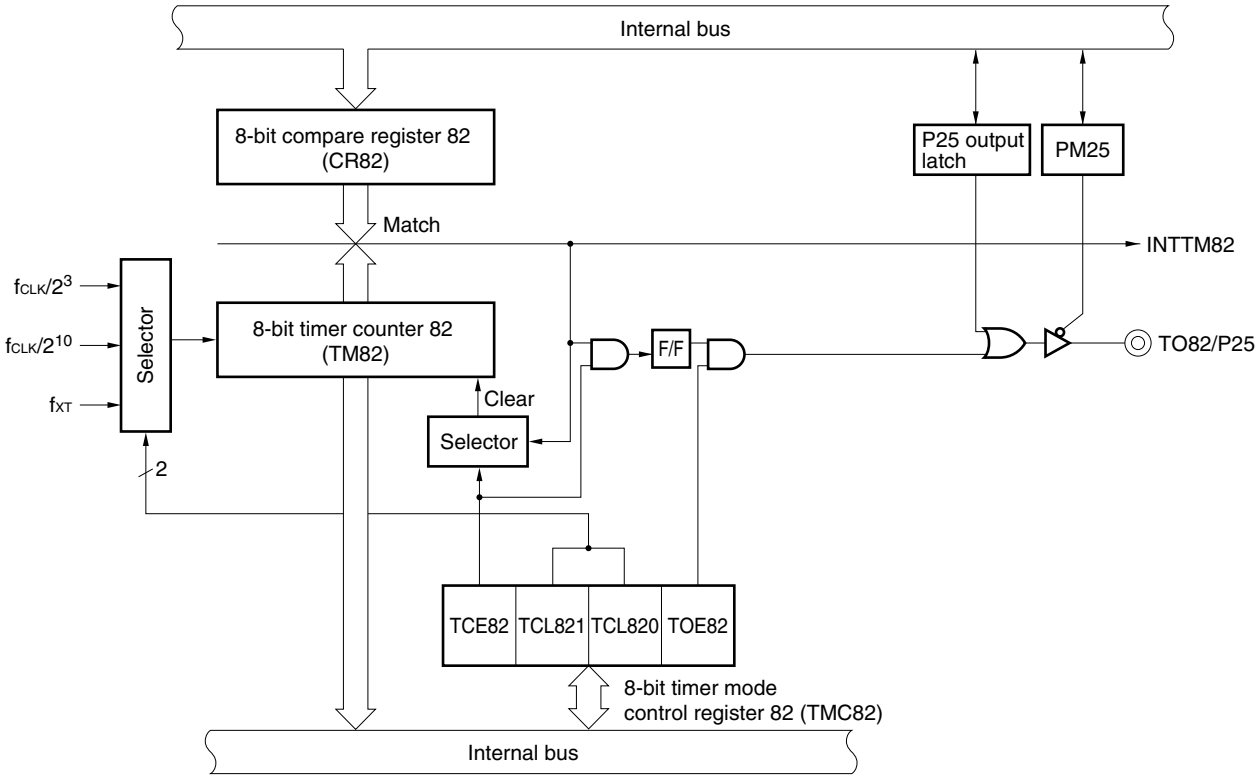
Remark f_{CLK}: fx or fcc

Figure 6-2. Block Diagram of 8-Bit Timer 81



Remark f_{CLK}: fx or fcc

Figure 6-3. Block Diagram of 8-Bit Timer 82



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

(1) 8-bit compare register 8n (CR8n)

This is an 8-bit register to compare the value set to CR8n with 8-bit timer counter 8n (TM8n) count value, and if they match, generates an interrupt request (INTTM8n).

CR8n is set with an 8-bit memory manipulation instruction. The 00H to FFH values can be set.

$\overline{\text{RESET}}$ input sets CR8n undefined.

Caution Be sure to stop the operation of the timer before rewriting CR8n. If CR8n is rewritten while the timer is operation-enabled, an interrupt request match signal may be generated at the time of the rewrite.

Remark n = 0 to 2

(2) 8-bit timer counter 8n (TM8n)

This is an 8-bit register to count pulses.

TM8n is read with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TM8n to 00H.

Remark n = 0 to 2

6.3 Registers Controlling 8-Bit Timer/Event Counters 80 to 82

The following two types of registers are used to control the 8-bit timer/event counters 80 to 82.

- 8-bit timer mode control registers 80, 81, and 82 (TMC80, TMC81, and TMC82)
- Port mode register 2 (PM2)

(1) 8-bit timer mode control register 80 (TMC80)

This register enables/stops operation of 8-bit timer counter 80 (TM80) and sets the count clock of TM80.

TMC80 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TMC80 to 00H.

Figure 6-4. Format of 8-Bit Timer Mode Control Register 80

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
TMC80	TCE80	0	0	0	0	TCL801	TCL800	0	FF59H	00H	R/W

TCE80	8-bit timer counter 80 operation control
0	Operation stopped (TM80 is cleared to 00H)
1	Operation enabled

TCL801	TCL800	8-bit timer/event counter 80 count clock selection	
		@f_X = 5.0 MHz operation	@f_{CC} = 2.0 MHz operation
0	0	$f_X/2^6$ (78.1 kHz)	$f_{CC}/2^6$ (31.3 kHz)
0	1	$f_X/2^9$ (9.76 kHz)	$f_{CC}/2^9$ (3.91 kHz)
1	0	Rising edge of TI80	
1	1	Falling edge of TI80	

Caution Be sure to stop the operation of the timer before setting TMC80.

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

2. f_{CC}: Main system clock oscillation frequency (RC oscillation)
3. The parenthesized values apply to operation at f_X = 5.0 MHz or f_{CC} = 2.0 MHz.

(2) 8-bit timer mode control register 81 (TMC81)

TMC01 determines whether to enable or stop operation of 8-bit timer counter 81 (TM81) and specifies the count clock for 8-bit timer 81.

TMC81 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TMC81 to 00H.

Figure 6-5. Format of 8-Bit Timer Mode Control Register 81

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
TMC81	TCE81	0	0	0	0	TCL811	TCL810	0	FF5CH	00H	R/W

TCE81	8-bit timer counter 81 operation control
0	Operation stopped (TM81 is cleared to 00H)
1	Operation enabled

TCL811	TCL810	8-bit timer 81 count clock selection	
		@f_X = 5.0 MHz and f_{XT} = 32.768 kHz operation	@f_{CC} = 2.0 MHz and f_{XT} = 32.768 kHz operation
0	0	$f_X/2^4$ (312.5 kHz)	$f_{CC}/2^4$ (125 kHz)
0	1	$f_X/2^8$ (19.5 kHz)	$f_{CC}/2^8$ (7.81 kHz)
1	0	f_{XT} (32.768 kHz)	
1	1		

Caution Be sure to stop the operation of the timer before setting TMC81.

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

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

3. f_{XT}: Subsystem clock oscillation frequency

4. The parenthesized values apply to operation at f_X = 5.0 MHz, f_{CC} = 2.0 MHz, or f_{XT} = 32.768 kHz.

(3) 8-bit timer mode control register 82 (TMC82)

TMC82 determines whether to enable or stop operation of 8-bit timer counter 82 (TM82) and specifies the count clock for 8-bit timer 82. It also controls the operation of the output control circuit.

TMC82 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TMC82 to 00H.

Figure 6-6. Format of 8-Bit Timer Mode Control Register 82

Symbol	<7>	6	5	4	3	2	1	<0>	Address	After reset	R/W
TMC82	TCE02	0	0	0	0	TCL821	TCL820	TOE82	FF5FH	00H	R/W

TCE82	8-bit timer counter 82 operation control
0	Operation stopped (TM82 is cleared to 00H)
1	Operation enabled

TCL821	TCL820	8-bit timer 82 count clock selection	
		@f_X = 5.0 MHz and f_{XT} = 32.768 kHz operation	@f_{CC} = 2.0 MHz and f_{XT} = 32.768 kHz operation
0	0	$f_X/2^3$ (625 kHz)	$f_{CC}/2^3$ (250 kHz)
0	1	$f_X/2^{10}$ (4.88 kHz)	$f_{CC}/2^{10}$ (1.95 kHz)
1	0	f_{XT} (32.768 kHz)	
1	1		

TOE82	8-bit timer 82 output control
0	Output disabled (port mode)
1	Output enabled

Caution Be sure to stop the operation of the timer before setting TMC82.

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

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

3. f_{XT}: Subsystem clock oscillation frequency

4. The parenthesized values apply to operation at f_X = 5.0 MHz, f_{CC} = 2.0 MHz, or f_{XT} = 32.768 kHz.

(4) Port mode register 2 (PM2)

This register sets port 2 input/output in 1-bit units.

When using the P23/INTP0/TI80 pin for timer input, set PM23 to 1.

When using the P25/TO82 pin for timer output, set PM25 and the output latch of P25 to 0.

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

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

Figure 6-7. Format of Port Mode Register 2

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W

PM2n	P2n pin input/output mode selection (n = 0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

6.4 8-Bit Timer/Event Counters 80 to 82 Operation

6.4.1 Operation as interval timer

Interval timer repeatedly generates an interrupt at time intervals specified by the count value set to 8-bit compare registers 80, 81, and 82 (CR80, CR81, and CR82) in advance.

To operate 8-bit timer/event counters 80 to 82 as an interval timer, make the settings in the following order.

<1> Set 8-bit timer counter 8n (TM8n) to operation-disabled (TCE8n (bit 7 of 8-bit timer mode control register 8n (TMC8n)) = 0)
<2> Select the count clock of the 8-bit timer/event counter (see **Tables 6-6** to **6-11**)
<3> Set the count value to CR8n
<4> Set TM8n to operation-enabled (TCE8n = 1)

When the count value of the 8-bit timer counter 8n (TM8n) matches the value set to CR8n, the value of TM8n is cleared to 00H and TM8n continues counting. At the same time, an interrupt request signal (INTTM8n) is generated.

Tables 6-6 through 6-11 show interval time, and Figures 6-8 and 6-9 show the timing of interval timer operation.

Caution When the setting of the count clock with TMC8n and the setting of the TM8n to operation-enable with an 8-bit memory manipulation instruction are performed at the same time, an error of one clock or more may occur in the first cycle after the timer is started. Because of this, when the 8-bit timer/event counter operates as an interval timer, be sure to make the settings in the order described above.

Remark n = 0 to 2

Table 6-6. Interval Time of 8-Bit Timer/Event Counter 80 (at f_X = 5.0 MHz Operation)

TCL801	TCL800	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	$2^6/f_X$ (12.8 μs)	$2^{14}/f_X$ (3.28 ms)	$2^6/f_X$ (12.8 μs)
0	1	$2^9/f_X$ (102 μs)	$2^{17}/f_X$ (26.2 ms)	$2^9/f_X$ (102 μs)
1	0	TI80 input cycle	$2^8 \times$ TI80 input cycle	TI80 input edge cycle
1	1			

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

Table 6-7. Interval Time of 8-Bit Timer/Event Counter 80 (at f_{CC} = 2.0 MHz Operation)

TCL801	TCL800	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	$2^6/f_{CC}$ (32 μs)	$2^{14}/f_{CC}$ (8.19 ms)	$2^6/f_{CC}$ (32 μs)
0	1	$2^9/f_{CC}$ (256 μs)	$2^{17}/f_{CC}$ (65.5 ms)	$2^9/f_{CC}$ (256 μs)
1	0	TI80 input cycle	$2^8 \times$ TI80 input cycle	TI80 input edge cycle
1	1			

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

Table 6-8. Interval Time of 8-Bit Timer 81 (at f_X = 5.0 MHz and f_{XT} = 32.768 kHz Operation)

TCL811	TCL810	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	$2^4/f_X$ (3.2 μs)	$2^{12}/f_X$ (819 μs)	$2^4/f_X$ (3.2 μs)
0	1	$2^8/f_X$ (51.2 μs)	$2^{16}/f_X$ (13.1 ms)	$2^8/f_X$ (51.2 μs)
1	0	$1/f_{XT}$ (30.5 μs)	$2^8/f_{XT}$ (7.81 ms)	$1/f_X$ (30.5 μs)
1	1			

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

2. f_{XT}: Subsystem clock oscillation frequency

Table 6-9. Interval Time of 8-Bit Timer 81 (at f_{CC} = 2.0 MHz and f_{XT} = 32.768 kHz Operation)

TCL811	TCL810	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	$2^4/f_{CC}$ (8.0 μs)	$2^{12}/f_{CC}$ (2.05 ms)	$2^4/f_{CC}$ (8.0 μs)
0	1	$2^8/f_{CC}$ (128 μs)	$2^{16}/f_{CC}$ (32.8 ms)	$2^8/f_{CC}$ (128 μs)
1	0	$1/f_{XT}$ (30.5 μs)	$2^8/f_{XT}$ (7.81 ms)	$1/f_{XT}$ (30.5 μs)
1	1			

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

2. f_{XT}: Subsystem clock oscillation frequency

Table 6-10. Interval Time of 8-Bit Timer 82 (at f_X = 5.0 MHz and f_{XT} = 32.768 kHz Operation)

TCL821	TCL820	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	$2^3/f_X$ (1.6 μs)	$2^{11}/f_X$ (410 μs)	$2^3/f_X$ (1.6 μs)
0	1	$2^{10}/f_X$ (205 μs)	$2^{18}/f_X$ (52.4 ms)	$2^{10}/f_X$ (205 μs)
1	0	$1/f_{XT}$ (30.5 μs)	$2^8/f_{XT}$ (7.81 ms)	$1/f_{XT}$ (30.5 μs)
1	1			

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

2. f_{XT}: Subsystem clock oscillation frequency

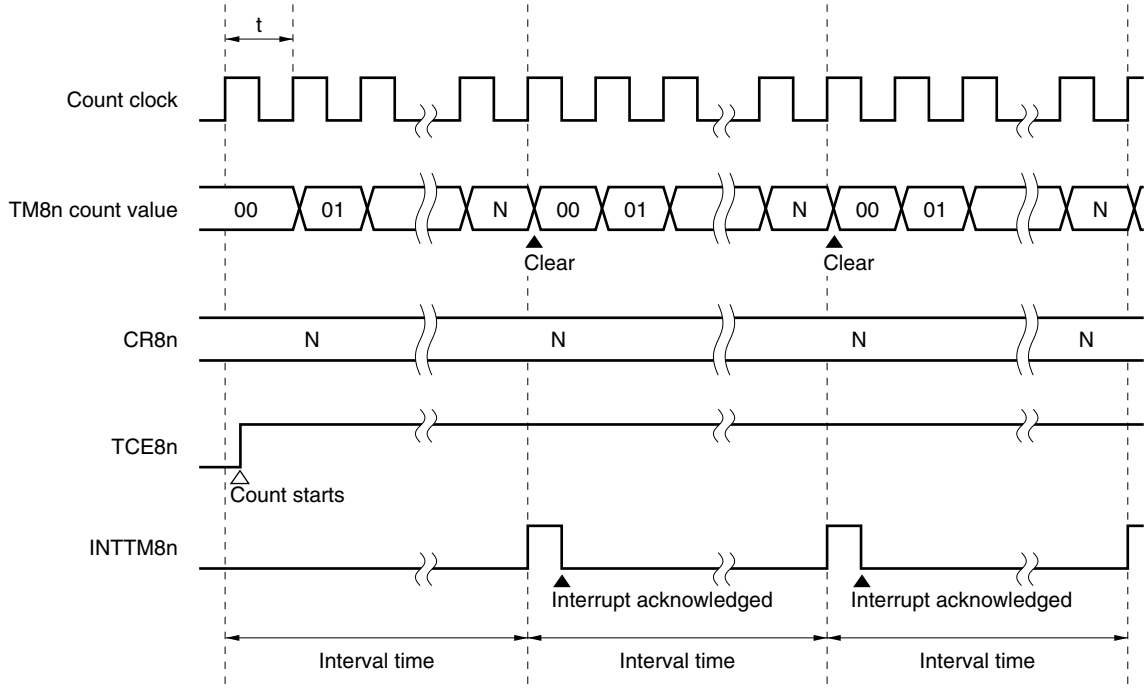
Table 6-11. Interval Time of 8-Bit Timer 82 (at f_{CC} = 2.0 MHz and f_{XT} = 32.768 kHz Operation)

TCL821	TCL820	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	$2^3/f_{CC}$ (4.0 μs)	$2^{11}/f_{CC}$ (1.02 ms)	$2^3/f_{CC}$ (4.0 μs)
0	1	$2^{10}/f_{CC}$ (512 μs)	$2^{18}/f_{CC}$ (131 ms)	$2^{10}/f_{CC}$ (512 μs)
1	0	$1/f_{XT}$ (30.5 μs)	$2^8/f_{XT}$ (7.81 ms)	$1/f_{XT}$ (30.5 μs)
1	1			

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

2. f_{XT}: Subsystem clock oscillation frequency

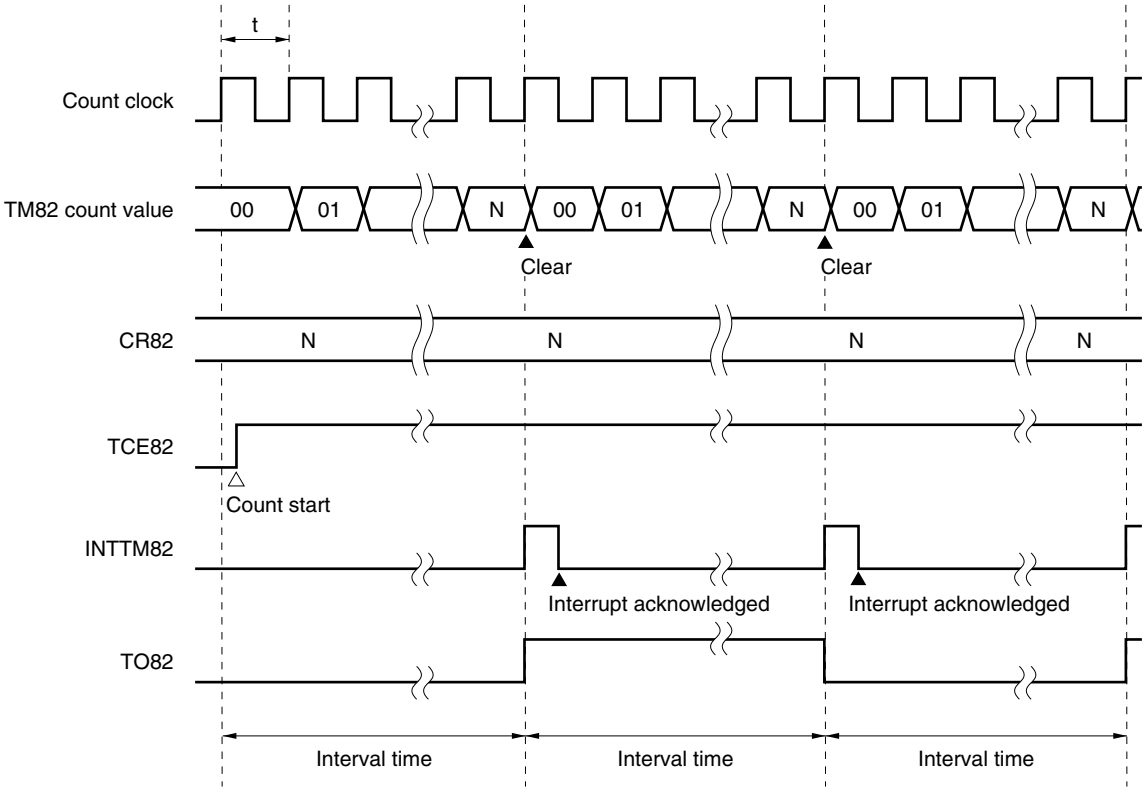
Figure 6-8. Interval Timer Operation Timing of TM80 and TM81



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

2. n = 0, 1

Figure 6-9. Interval Timer Operation Timing of TM82



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

6.4.2 Operation as external event counter (timer 80 only)

The external event counter counts the number of external clock pulses input to the TI80/P23/INTP0 pin by using 8-bit timer counter 80 (TM80).

To operate 8-bit timer/event counter 80 as an external event counter, make the settings in the following order.

<1> Set P23 to input mode (PM23 = 1)
<2> Set 8-bit timer counter 80 (TM80) to operation-disabled (TCE80 (bit 7 of 8-bit timer mode control register 80 (TMC80)) = 0)
<3> Specify the rising edge/falling edge of TI80 (see **Figure 6-4**)
<4> Set the count value to CR80
<5> Set TM80 to operation-enabled (TCE80 = 1)

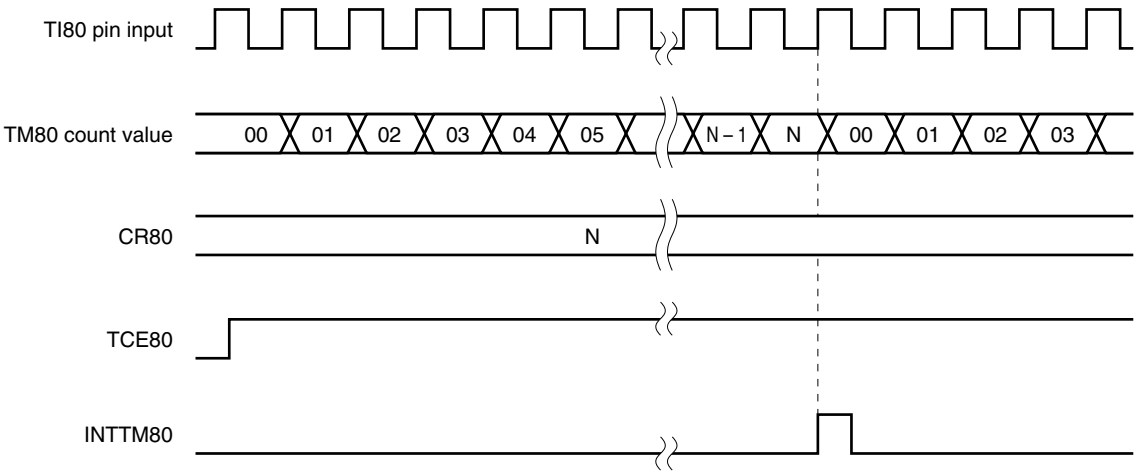
Each time the valid edge specified by the bit 1 (TCL800) of TMC80 is input, the value of 8-bit timer counter 80 (TM80) is incremented.

When the count value of the TM80 matches the value set to CR80, the value of TM80 is cleared to 00H and TM80 continues counting. At the same time, an interrupt request signal (INTTM80) is generated.

Figure 6-10 shows the timing of external event counter operation (with rising edge specified).

Caution When the setting of the count clock with TMC80 and the setting of the TM80 to operation-enable with an 8-bit memory manipulation instruction are performed at the same time, an error of one clock or more may occur in the first cycle after the timer is started. Because of this, when the 8-bit timer/event counter operates as an external event counter, be sure to make the settings in the order described above.

Figure 6-10. External Event Counter Operation Timing (with Rising Edge Specified)



Remark N = 00H to FFH

6.4.3 Operation as square wave output (timer 82 only)

The 8-bit timer/event counter can generate the output square waves of any frequency at intervals specified by the count value set to 8-bit compare register 82 (CR82) in advance.

To operate the 8-bit timer 82 as square wave output, make the settings in the following order.

<1> Set P25 to output mode (PM25 = 0), and set the output latch of P25 to 0
<2> Set 8-bit timer counter 82 (TM82) to operation-disabled (TCE82 (bit 7 of 8-bit timer mode control register 82 (TMC82)) = 0)
<3> Set the count clock of 8-bit timer 82 (see **Tables 6-12** and **6-13**), and set TO82 to output-enabled (TOE82 (bit 0 of TMC82) = 1)
<4> Set the count value to CR82
<5> Set TM82 to operation-enabled (TCE82 = 1)

When the count value of an 8-bit timer counter 82 (TM82) matches the value set in CR82, the TO82/P25 pin output will be inverted, respectively. Through application of this mechanism, square waves of any frequency can be output. As soon as a match occurred, the TM82 value will be cleared to 00H then resume to count, generating an interrupt request signal (INTTM82).

Setting 0 to the bit 7 in TMC82, that is, TCE82 makes the square-wave output clear to 0.

Tables 6-12 and 6-13 list square wave output range, and Figure 6-11 shows timing of square wave output.

Caution When the setting of the count clock with TMC82 and the setting of the TM82 to operation-enabled with an 8-bit memory manipulation instruction are performed at the same time, an error of one clock or more may occur in the first cycle after the timer is started. Because of this, when the 8-bit timer/event counter operates as a square-wave output, be sure to make the settings in the order described above.

Table 6-12. Square Wave Output Range of 8-Bit Timer 82 (at f_X = 5.0 MHz and f_{XT} = 32.768 kHz Operation)

TCL821	TCL820	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	$2^3/f_X$ (1.6 μs)	$2^{11}/f_X$ (409 μs)	$2^3/f_X$ (1.6 μs)
0	1	$2^{10}/f_X$ (205 μs)	$2^{18}/f_X$ (52.4 ms)	$2^{10}/f_X$ (205 μs)
1	0	$1/f_{XT}$ (30.5 μs)	$2^8/f_{XT}$ (7.81 ms)	$1/f_{XT}$ (30.5 μs)
1	1			

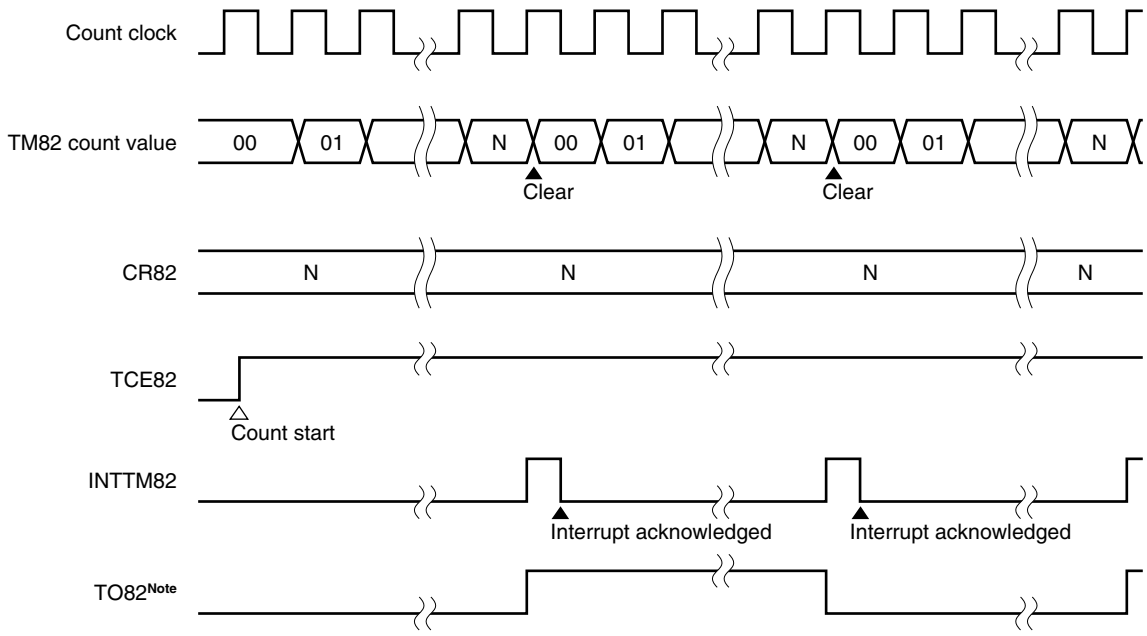
Remarks 1. f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)
2. f_{XT}: Subsystem clock oscillation frequency

Table 6-13. Square Wave Output Range of 8-Bit Timer 82 (at f_{CC} = 2.0 MHz and f_{XT} = 32.768 kHz Operation)

TCL821	TCL820	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	$2^3/f_{CC}$ (4.0 μs)	$2^{11}/f_{CC}$ (1.02 ms)	$2^3/f_{CC}$ (4.0 μs)
0	1	$2^{10}/f_{CC}$ (512 μs)	$2^{18}/f_{CC}$ (131 ms)	$2^{10}/f_{CC}$ (512 μs)
1	0	$1/f_{XT}$ (30.5 μs)	$2^8/f_{XT}$ (7.81 ms)	$1/f_{XT}$ (30.5 μs)
1	1			

Remarks 1. f_{CC}: Main system clock oscillation frequency (RC oscillation)
2. f_{XT}: Subsystem clock oscillation frequency

Figure 6-11. Square Wave Output Timing



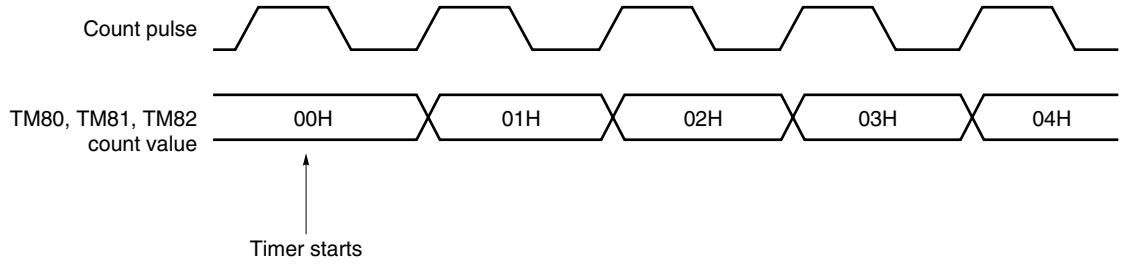
Note The initial value of TO82 at output enable (TOE82 = 1) becomes low-level.

6.5 Cautions Related to 8-Bit Timer/Event Counters 80 to 82

(1) Error on starting timer

An error of up to 1 clock occurs after the timer has been started until a coincidence signal is generated. This is because 8-bit timer counters 80, 81, and 82 (TM80, TM81, and TM82) started in asynchronization with the count pulse.

Figure 6-12. Start Timing of 8-Bit Timer Counters 80, 81, and 82

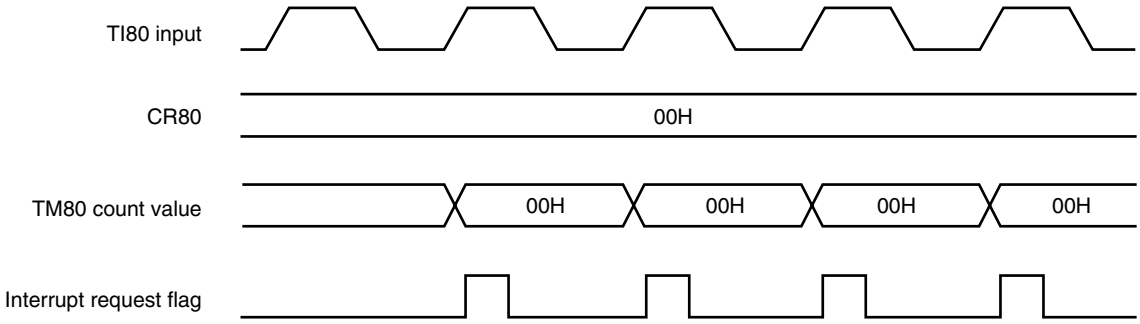


(2) Setting of 8-bit compare register

8-bit compare registers 80, 81, and 82 (CR80, CR81, and CR82) can be set to 00H.

Therefore, one pulse can be counted when the timer 80 operates as an event counter.

Figure 6-13. External Event Counter Operation Timing



7.1 8-Bit Timers 30, 40 Functions

The 8-bit timers in the μPD789835 Subseries have 2 channels (timer 30 and timer 40). The operation modes listed in the following table can be set via mode register settings.

Table 7-1. 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.

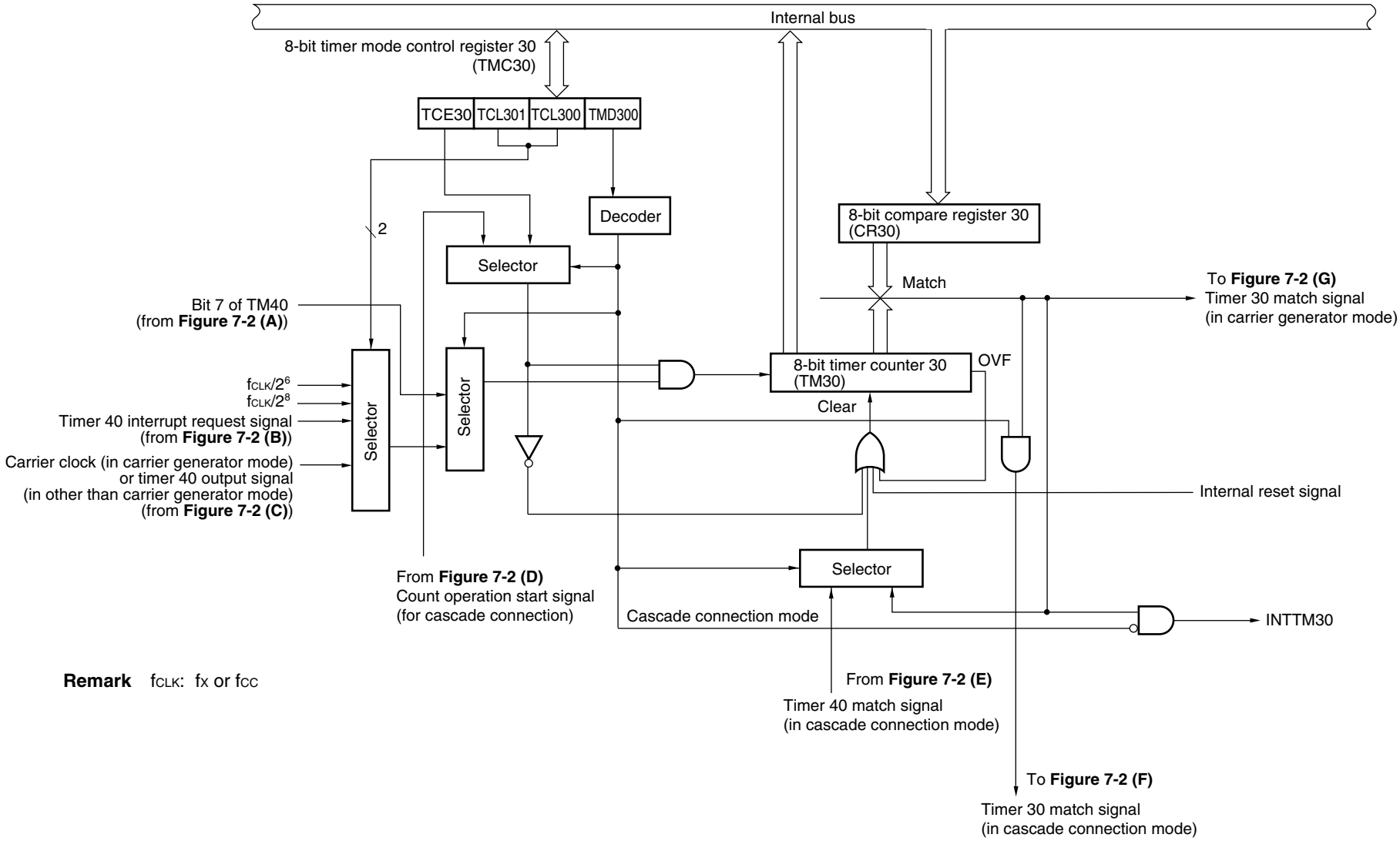
7.2 8-Bit Timers 30, 40 Configuration

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

Table 7-2. Configuration of 8-Bit Timers 30, 40

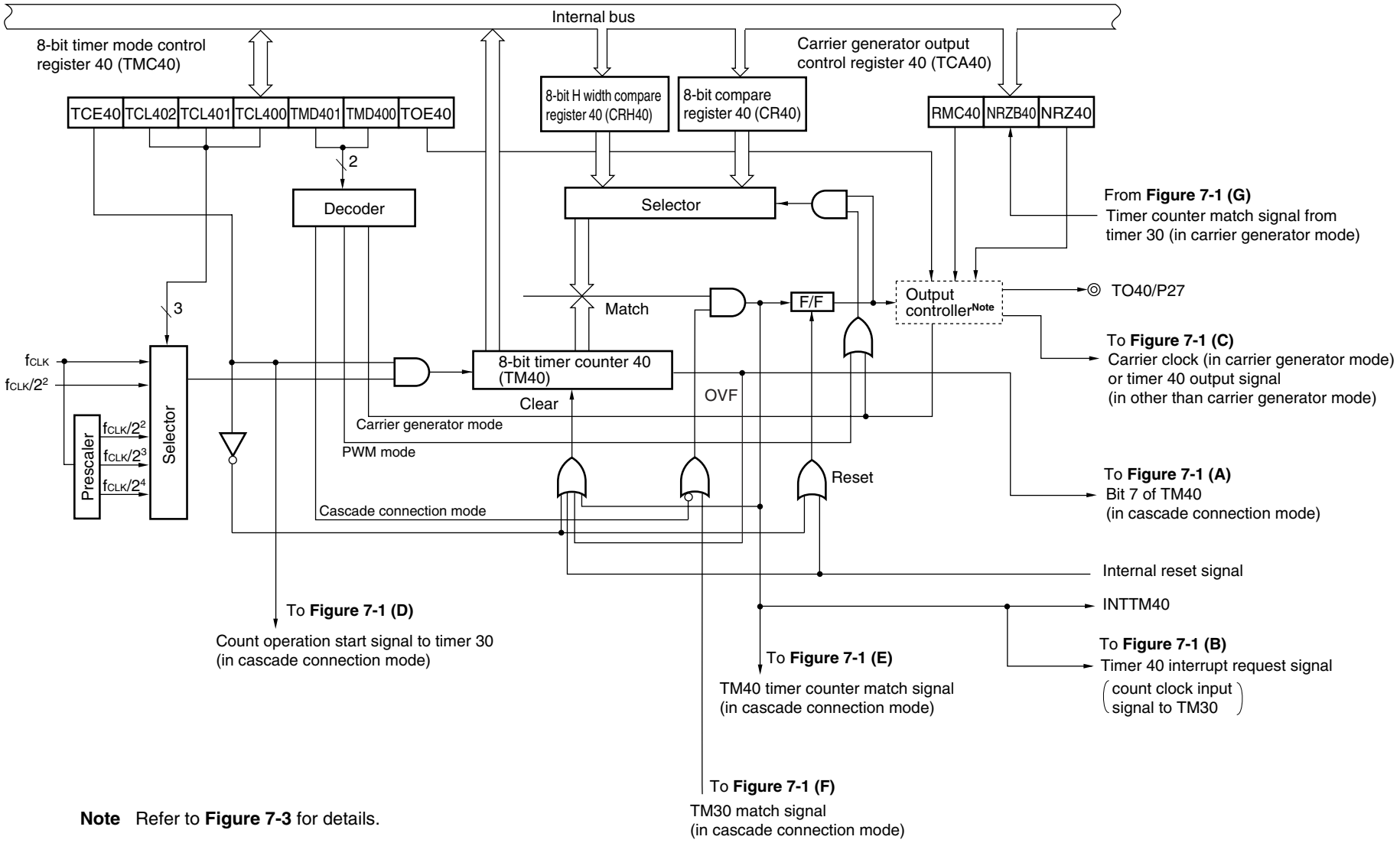
Item	Configuration
Timer counters	8 bits × 2 (TM30, TM40)
Registers	Compare registers: 8 bits × 3 (CR30, CR40, CRH40)
Timer output	1 (TO40)
Control registers	8-bit timer mode control register 30 (TMC30) 8-bit timer mode control register 40 (TMC40) Carrier generator output control register 40 (TCA40) Port mode register 2 (PM2)

Figure 7-1. Block Diagram of Timer 30



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

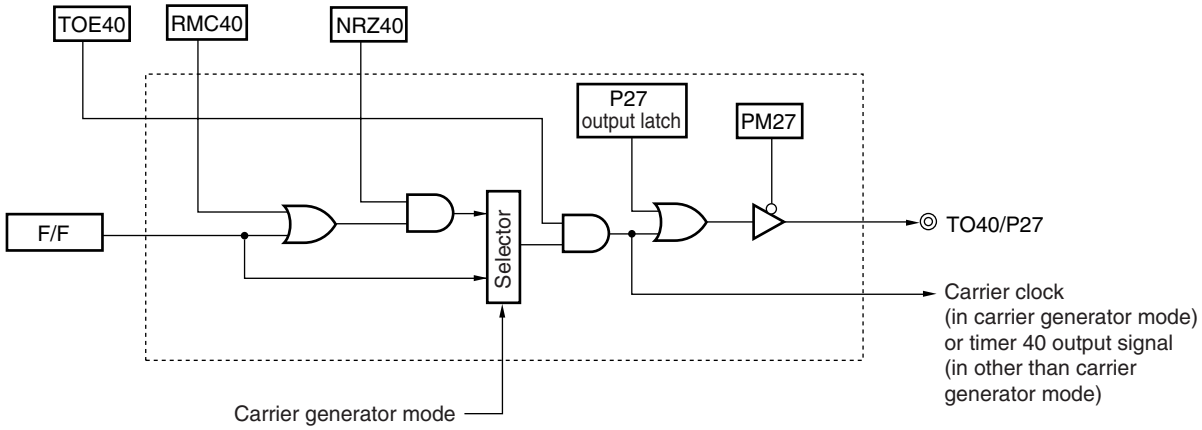
Figure 7-2. Block Diagram of Timer 40



Note Refer to **Figure 7-3** for details.

Remark f_{CLK}: fx or fCC

Figure 7-3. 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).

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)

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

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

7.3 Registers Controlling 8-Bit Timers 30, 40

The 8-bit timers 30 and 40 are controlled by the following four 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 2 (PM2)

(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 7-4. 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	FF62H	00H	R/W

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

TCL301	TCL300	Selection of timer 30 count clock	
		@fx = 5.0 MHz operation	@fcc = 2.0 MHz operation
0	0	$f_X/2^6$ (78.1 kHz)	$f_{CC}/2^6$ (31.3 kHz)
0	1	$f_X/2^8$ (19.5 kHz)	$f_{CC}/2^8$ (7.81 kHz)
1	0	Timer 40 match signal	
1	1	Carrier clock (in carrier generator mode) or timer 40 output signal (in other than carrier generator mode)	

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

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

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

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

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

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

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

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

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

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

Figure 7-5. 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	Clears TM40 count value and stops operation (the count value is also cleared for TM30 during cascade connection mode)
1	Starts count operation (the count operation is also started for TM30 during cascade connection mode)

TCL402	TCL401	TCL400	Selection of timer 40 count clock	
			@f_X = 5.0 MHz operation	@f_{CC} = 2.0 MHz operation
0	0	0	f_X (5.0 MHz)	f_{CC} (2.0 MHz)
0	0	1	$f_X/2^2$ (1.25 MHz)	$f_{CC}/2^2$ (500 kHz)
0	1	0	f_X (5.0 MHz)	f_{CC} (2.0 MHz)
0	1	1	$f_X/2^2$ (1.25 MHz)	$f_{CC}/2^2$ (500 kHz)
1	0	0	$f_X/2^3$ (625 kHz)	$f_{CC}/2^3$ (250 kHz)
1	0	1	$f_X/2^4$ (313 kHz)	$f_{CC}/2^4$ (125 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 (bit 7 of TMC40) in cascade connection mode, any setting for TCE30 is ignored.

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

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

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

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

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

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

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

Figure 7-6. 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	FF67H	00H	W

RMC40	Control of remote control output
0	When NRZB40 = 1, a carrier pulse is output to TO40/P27 pin
1	When NRZB40 = 1, high-level signal is output to TO40/P27 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	Outputs low-level signal (carrier clock is stopped)
1	Outputs carrier pulse or high-level signal

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

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

(4) Port mode register 2 (PM2)

This register is used to set the I/O mode of port 2 in 1-bit units.

When using the P27/TO40 pin as a timer output, set the PM27 and P27 output latch to 0.

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

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

Figure 7-7. Format of Port Mode Register 2

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W

PM2n	I/O mode of P2n pin (n = 0 to 7)
0	Output mode (output buffer is on)
1	Input mode (output buffer is off)

7.4 8-Bit Timers 30, 40 Operation

7.4.1 Operation as 8-bit timer counter

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

- Interval timer with 8-bit resolution
- 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 REM (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 7-4** and **7-5**).
<5> Set the count clock for timer n0 (see **Tables 7-3** to **7-6**).
<6> Enable the operation of TMn0 (TCEn0 = 1).

When the count value of 8-bit timer counter n0 (TMn0) matches the value set in CRn0, TMn0 is cleared to 00H and continues counting. At the same time, an interrupt request signal (INTTMn0) is generated.
Tables 7-3 to 7-6 show interval time, and Figures 7-8 to 7-13 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 7-3. Interval Time of Timer 30 (at f_X = 5.0 MHz Operation)

TCL301	TCL300	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	$2^6/f_X$ (12.8 μs)	$2^{14}/f_X$ (3.28 ms)	$2^6/f_X$ (12.8 μs)
0	1	$2^8/f_X$ (51.2 μs)	$2^{16}/f_X$ (13.1 ms)	$2^8/f_X$ (51.2 μs)
1	0	Input cycle of timer 40 match signal	Input cycle of timer 40 match signal $\times 2^8$	Input cycle of timer 40 match signal
1	1	Input cycle of timer 40 output	Input cycle of timer 40 output $\times 2^8$	Input cycle of timer 40

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

Table 7-4. Interval Time of Timer 30 (at f_{CC} = 2.0 MHz Operation)

TCL301	TCL300	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	$2^6/f_{CC}$ (32 μs)	$2^{14}/f_{CC}$ (8.19 ms)	$2^6/f_{CC}$ (32 μs)
0	1	$2^8/f_{CC}$ (128 μs)	$2^{16}/f_{CC}$ (32.8 ms)	$2^8/f_{CC}$ (128 μs)
1	0	Input cycle of timer 40 match signal	Input cycle of timer 40 match signal $\times 2^8$	Input cycle of timer 40 match signal
1	1	Input cycle of timer 40 output	Input cycle of timer 40 output $\times 2^8$	Input cycle of timer 40 output

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

Table 7-5. Interval Time of Timer 40 (at f_X = 5.0 MHz Operation)

TCL402	TCL401	TCL400	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	$1/f_X$ (0.2 μs)	$2^8/f_X$ (51.2 μs)	$1/f_X$ (0.2 μs)
0	0	1	$2^2/f_X$ (0.8 μs)	$2^{10}/f_X$ (205 μs)	$2^2/f_X$ (0.8 μs)
0	1	0	$1/f_X$ (0.2 μs)	$2^8/f_X$ (51.2 μs)	$1/f_X$ (0.2 μs)
0	1	1	$2^2/f_X$ (0.8 μs)	$2^{10}/f_X$ (205 μs)	$2^2/f_X$ (0.8 μs)
1	0	0	$2^3/f_X$ (1.6 μs)	$2^{11}/f_X$ (410 μs)	$2^3/f_X$ (1.6 μs)
1	0	1	$2^4/f_X$ (3.2 μs)	$2^{12}/f_X$ (819 μs)	$2^4/f_X$ (3.2 μs)

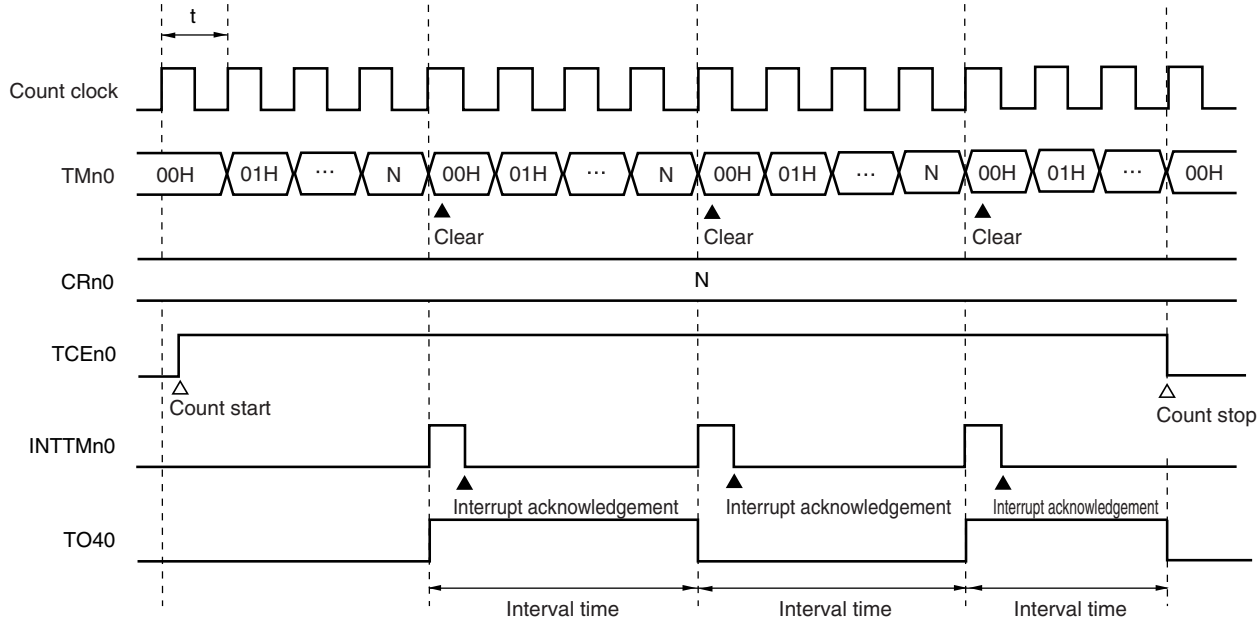
Remark f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

Table 7-6. Interval Time of Timer 40 (at f_{CC} = 2.0 MHz Operation)

TCL402	TCL401	TCL400	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	$1/f_{CC}$ (0.5 μs)	$2^8/f_{CC}$ (128 μs)	$1/f_{CC}$ (0.5 μs)
0	0	1	$2^2/f_{CC}$ (2.0 μs)	$2^{10}/f_{CC}$ (512 μs)	$2^2/f_{CC}$ (2.0 μs)
0	1	0	$1/f_{CC}$ (0.5 μs)	$2^8/f_{CC}$ (128 μs)	$1/f_{CC}$ (0.5 μs)
0	1	1	$2^2/f_{CC}$ (2.0 μs)	$2^{10}/f_{CC}$ (512 μs)	$2^2/f_{CC}$ (2.0 μs)
1	0	0	$2^3/f_{CC}$ (4.0 μs)	$2^{11}/f_{CC}$ (1.02 ms)	$2^3/f_{CC}$ (4.0 μs)
1	0	1	$2^4/f_{CC}$ (8.0 μs)	$2^{12}/f_{CC}$ (4.10 ms)	$2^4/f_{CC}$ (8.0 μs)

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

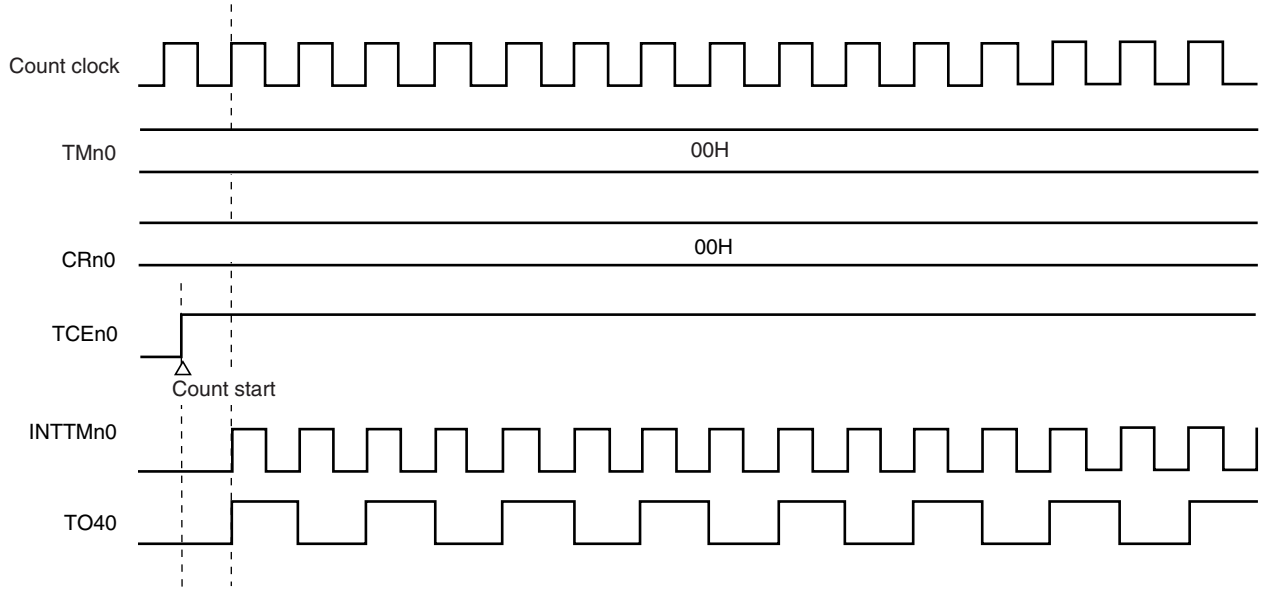
Figure 7-8. Timing of Interval Timer Operation with 8-Bit Resolution (Basic Operation)



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

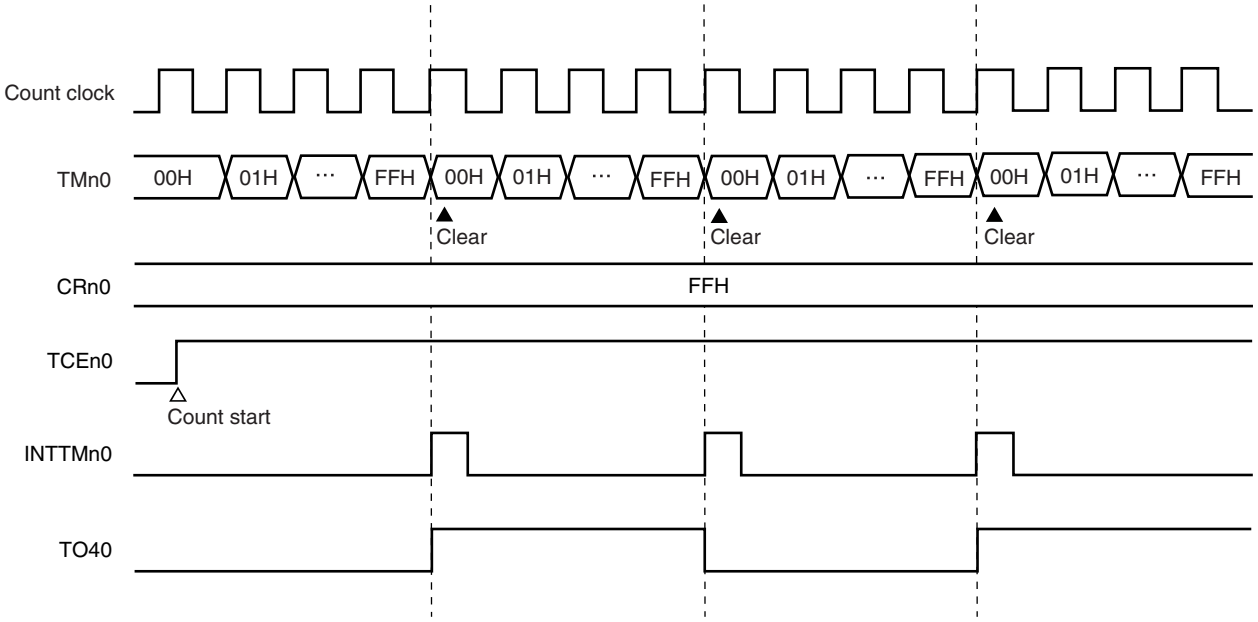
2. n = 3, 4

Figure 7-9. Timing of Interval Timer Operation with 8-Bit Resolution (When CRn0 Is Set to 00H)



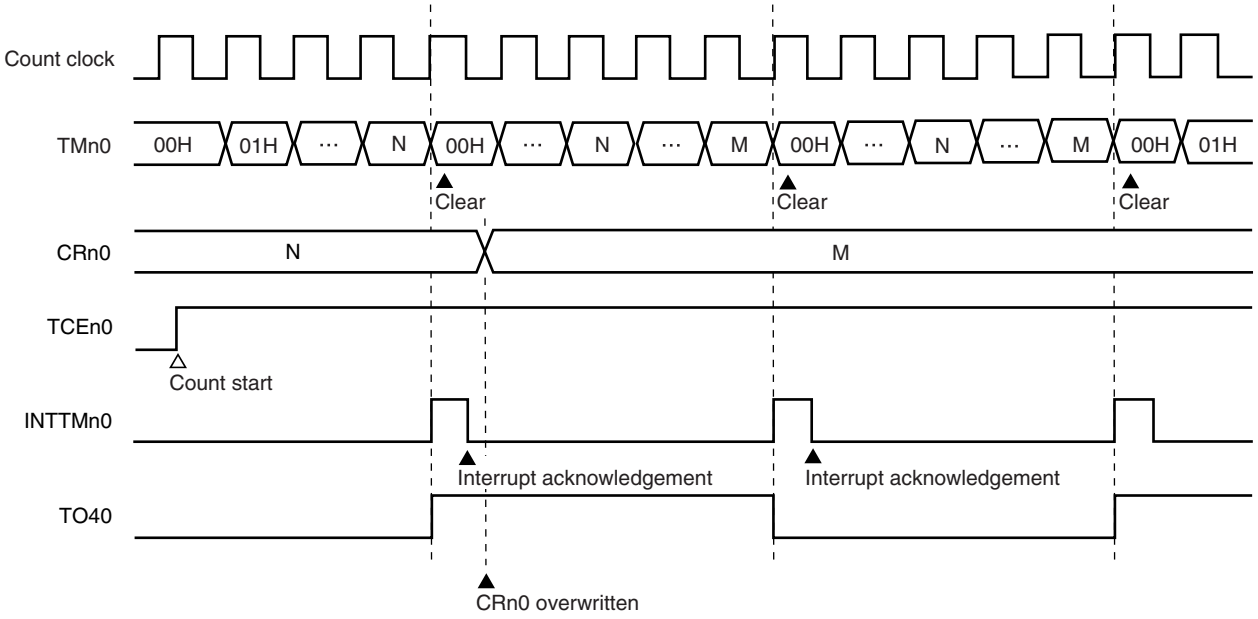
Remark n = 3, 4

Figure 7-10. Timing of Interval Timer Operation with 8-Bit Resolution (When CRn0 Is Set to FFH)



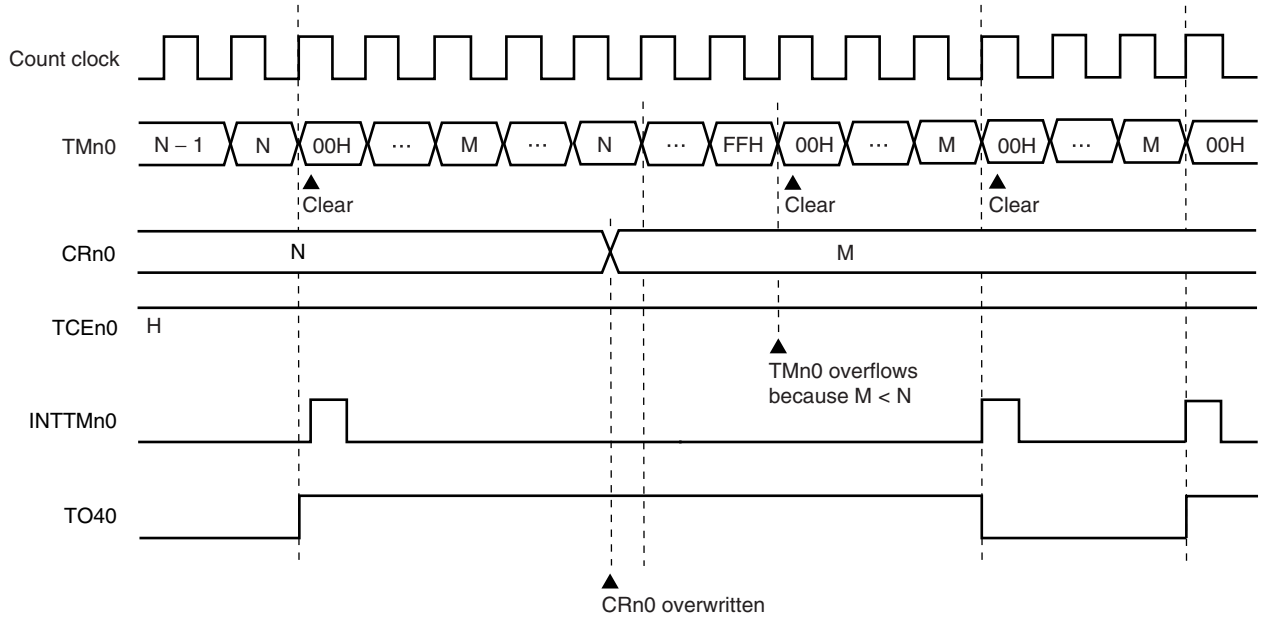
Remark n = 3, 4

Figure 7-11. Timing of Interval Timer Operation with 8-Bit Resolution (When CRn0 Changes from N to M (N < M))



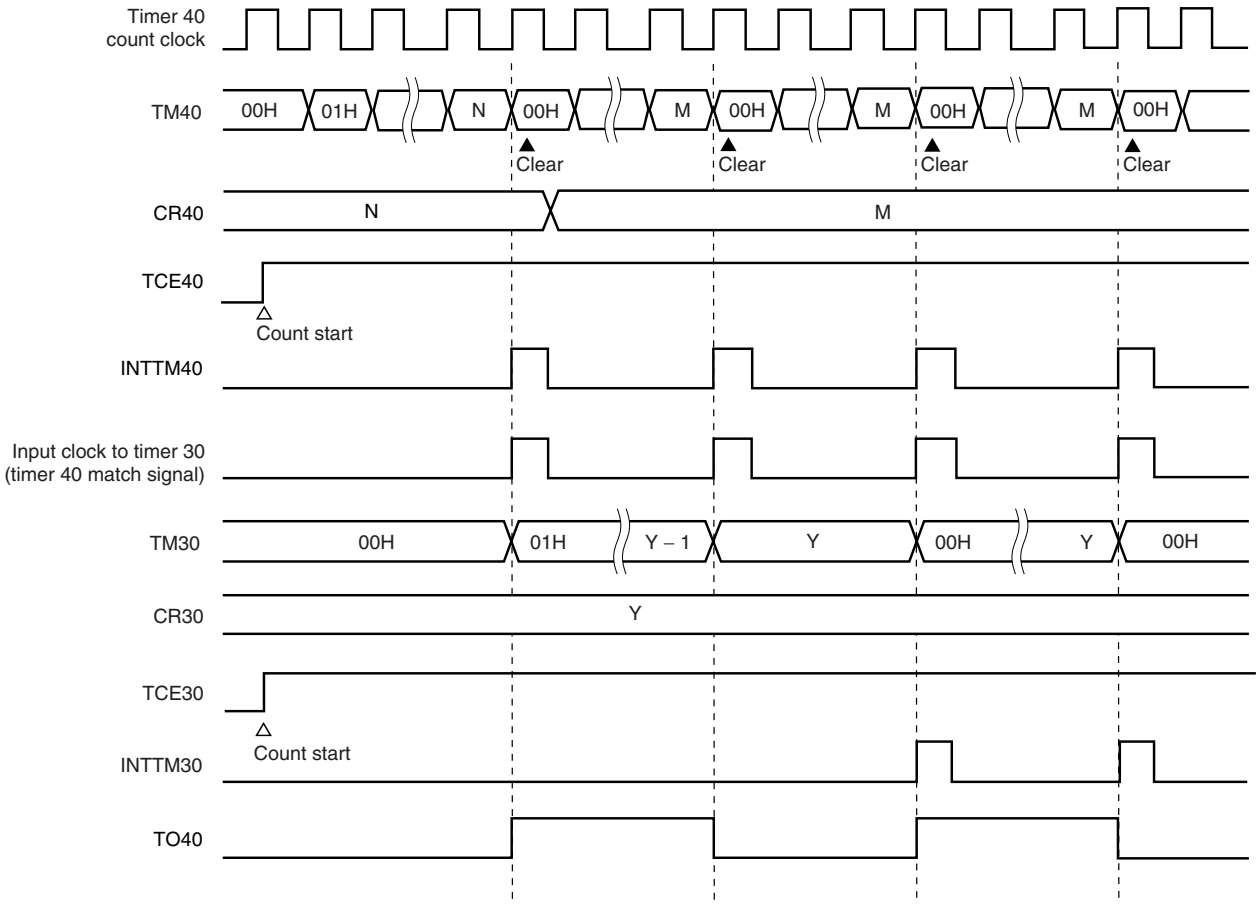
Remark n = 3, 4

Figure 7-12. Timing of Interval Timer Operation with 8-Bit Resolution (When CRn0 Changes from N to M (N > M))



Remark n = 3, 4

Figure 7-13. 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 P27 to output mode (PM27 = 0).
<2> Set the output latch of P27 to 0.
<3> Disable operation of 8-bit 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.

Tables 7-7 and 7-8 show the square-wave output range, and Figure 7-14 shows the timing of square-wave output.

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

Table 7-7. Square-Wave Output Range of Timer 40 (at f_X = 5.0 MHz Operation)

TCL402	TCL401	TCL400	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	0	$1/f_X$ (0.2 μs)	$2^8/f_X$ (51.2 μs)	$1/f_X$ (0.2 μs)
0	0	1	$2^2/f_X$ (0.8 μs)	$2^{10}/f_X$ (205 μs)	$2^2/f_X$ (0.8 μs)
0	1	0	$1/f_X$ (0.2 μs)	$2^8/f_X$ (51.2 μs)	$1/f_X$ (0.2 μs)
0	1	1	$2^2/f_X$ (0.8 μs)	$2^{10}/f_X$ (205 μs)	$2^2/f_X$ (0.8 μs)
1	0	0	$2^3/f_X$ (1.6 μs)	$2^{11}/f_X$ (410 μs)	$2^3/f_X$ (1.6 μs)
1	0	1	$2^4/f_X$ (3.2 μs)	$2^{12}/f_X$ (819 μs)	$2^4/f_X$ (3.2 μs)

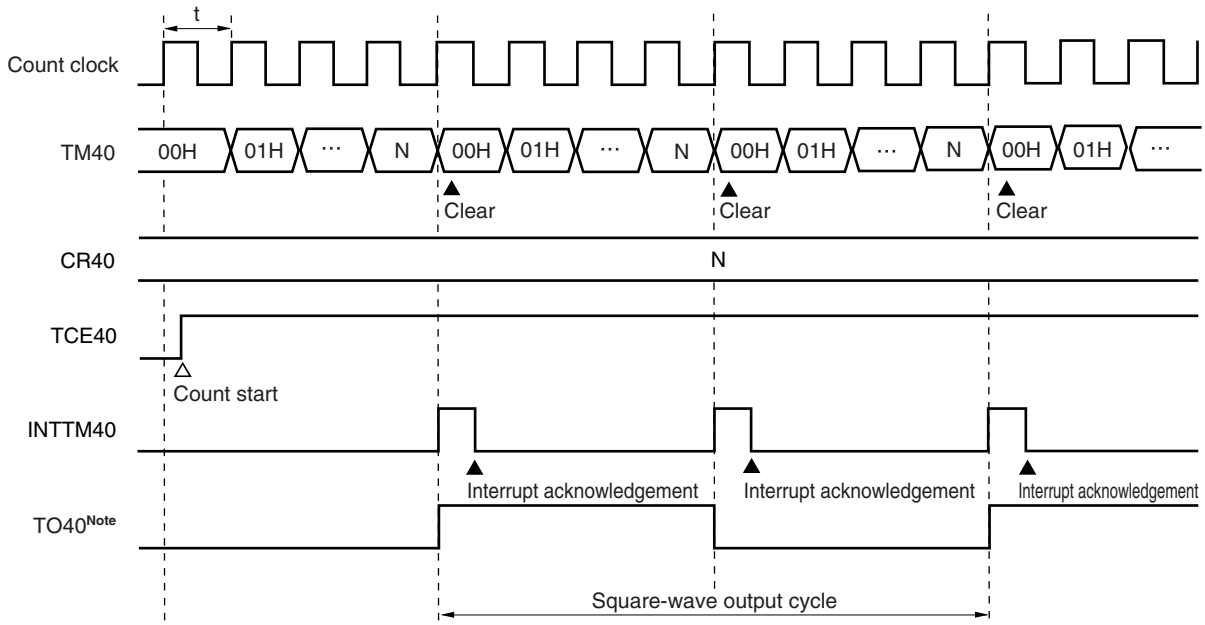
Remark f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

Table 7-8. Square-Wave Output Range of Timer 40 (at f_{CC} = 2.0 MHz Operation)

TCL402	TCL401	TCL400	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	0	$1/f_{CC}$ (0.5 μs)	$2^8/f_{CC}$ (128 μs)	$1/f_{CC}$ (0.5 μs)
0	0	1	$2^2/f_{CC}$ (2.0 μs)	$2^{10}/f_{CC}$ (512 μs)	$2^2/f_{CC}$ (2.0 μs)
0	1	0	$1/f_{CC}$ (0.5 μs)	$2^8/f_{CC}$ (128 μs)	$1/f_{CC}$ (0.5 μs)
0	1	1	$2^2/f_{CC}$ (2.0 μs)	$2^{10}/f_{CC}$ (512 μs)	$2^2/f_{CC}$ (2.0 μs)
1	0	0	$2^3/f_{CC}$ (4.0 μs)	$2^{11}/f_{CC}$ (1.02 ms)	$2^3/f_{CC}$ (4.0 μs)
1	0	1	$2^4/f_{CC}$ (8.0 μs)	$2^{12}/f_{CC}$ (4.10 ms)	$2^4/f_{CC}$ (8.0 μs)

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

Figure 7-14. 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 where N = 00H to FFH

7.4.2 Operation as 16-bit timer counter

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

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

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

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

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

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

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

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

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

Tables 7-9 and 7-10 show interval time, and Figure 7-15 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 7-9. 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	1/fx (0.2 μs)	2^{16}/fx (13.1 ms)	1/fx (0.2 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)

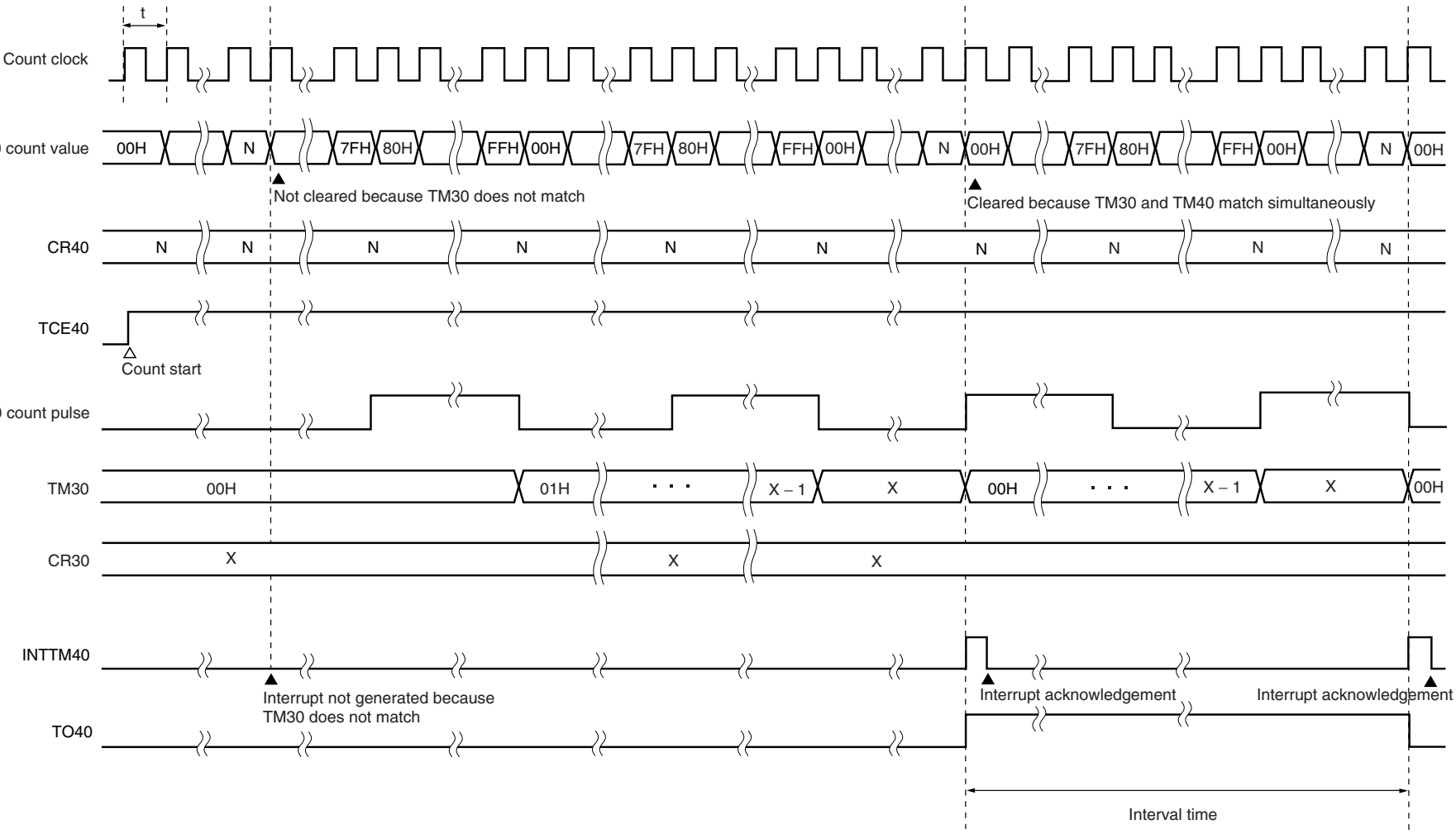
Remark fx: Main system clock oscillation frequency (ceramic/crystal oscillation)

Table 7-10. Interval Time with 16-Bit Resolution (at f_{CC} = 2.0 MHz Operation)

TCL402	TCL401	TCL400	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	$1/f_{CC}$ (0.5 μs)	$2^{16}/f_{CC}$ (32.8 ms)	$1/f_{CC}$ (0.5 μs)
0	0	1	$2^{2}/f_{CC}$ (2.0 μs)	$2^{18}/f_{CC}$ (131 ms)	$2^{2}/f_{CC}$ (2.0 μs)
0	1	0	$1/f_{CC}$ (0.5 μs)	$2^{16}/f_{CC}$ (32.8 ms)	$1/f_{CC}$ (0.5 μs)
0	1	1	$2^{2}/f_{CC}$ (2.0 μs)	$2^{18}/f_{CC}$ (131 ms)	$2^{2}/f_{CC}$ (2.0 μs)
1	0	0	$2^{3}/f_{CC}$ (4.0 μs)	$2^{19}/f_{CC}$ (262 ms)	$2^{3}/f_{CC}$ (4.0 μs)
1	0	1	$2^{4}/f_{CC}$ (8.0 μs)	$2^{20}/f_{CC}$ (524 ms)	$2^{4}/f_{CC}$ (8.0 μs)

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

Figure 7-15. Timing of Interval Timer Operation with 16-Bit Resolution



Remark Interval time = $(256X + N + 1) \times t$ where 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 P27 to output mode (PM27 = 0) and P27 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.

Tables 7-11 and 7-12 show the square wave output range, and Figure 7-16 shows timing of square wave output.

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

Table 7-11. Square-Wave Output Range with 16-Bit Resolution (at f_X = 5.0 MHz Operation)

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

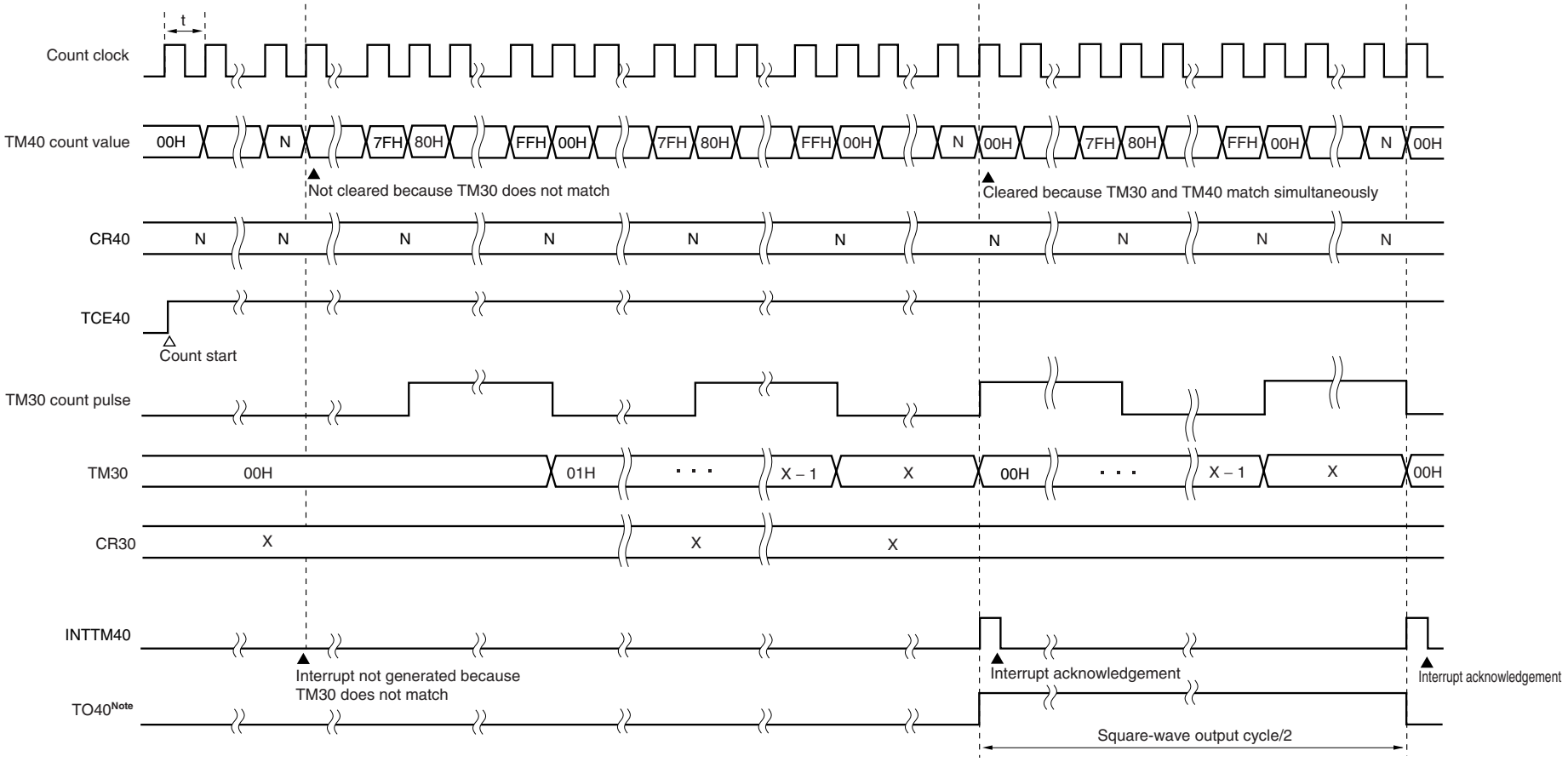
Remark f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

Table 7-12. Square-Wave Output Range with 16-Bit Resolution (at f_{CC} = 2.0 MHz Operation)

TCL402	TCL401	TCL400	Minimum Pulse Width	Maximum Pulse Width	Resolution
0	0	0	$1/f_{CC}$ (0.5 μs)	$2^{16}/f_{CC}$ (32.8 ms)	$1/f_{CC}$ (0.5 μs)
0	0	1	$2^2/f_{CC}$ (2.0 μs)	$2^{18}/f_{CC}$ (131 ms)	$2^2/f_{CC}$ (2.0 μs)
0	1	0	$1/f_{CC}$ (0.5 μs)	$2^{16}/f_{CC}$ (32.8 ms)	$1/f_{CC}$ (0.5 μs)
0	1	1	$2^2/f_{CC}$ (2.0 μs)	$2^{18}/f_{CC}$ (131 ms)	$2^2/f_{CC}$ (2.0 μs)
1	0	0	$2^3/f_{CC}$ (4.0 μs)	$2^{19}/f_{CC}$ (262 ms)	$2^3/f_{CC}$ (4.0 μs)
1	0	1	$2^4/f_{CC}$ (8.0 μs)	$2^{20}/f_{CC}$ (524 ms)	$2^4/f_{CC}$ (8.0 μs)

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

Figure 7-16. 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 where X = 00H to FFH, N = 00H to FFH

7.4.3 Operation as carrier generator

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

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

<1> Disable operation of TM30 and TM40 (TCE30 = 0, TCE40 = 0).
<2> Disable timer output of 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 7-4** and **7-5**).
<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 P27 to output mode (PM27 = 0) and the P27 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).

The operation of the carrier generator is as follows.

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

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

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

Figures 7-17 to 7-19 show the operation timing of the carrier generator.

Figure 7-17. Timing of Carrier Generator Operation (When CR40 = N, CRH40 = M (M > N))

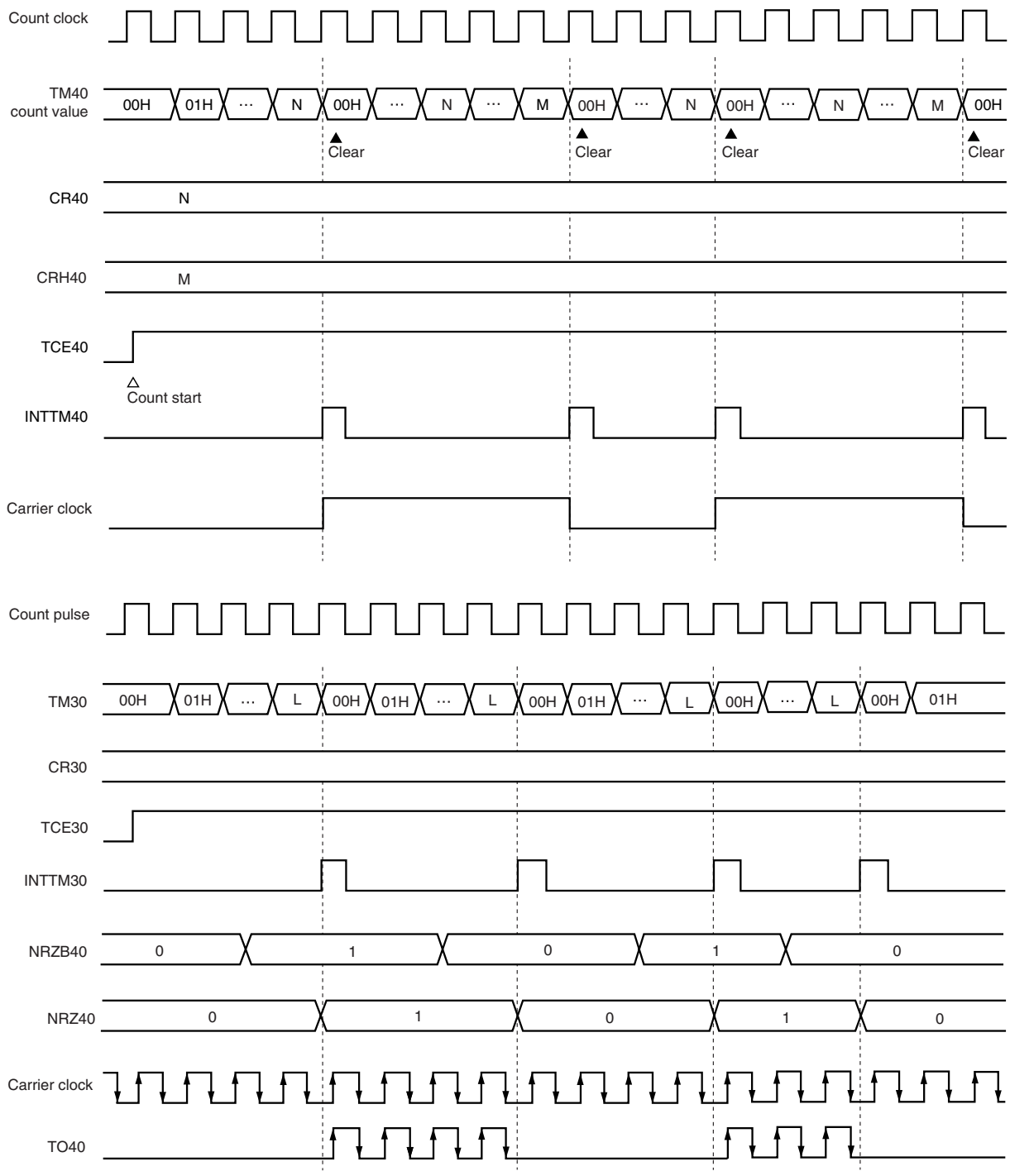


Figure 7-18. Timing of Carrier Generator Operation
(When CR40 = N, CRH40 = M (M < N), Phases of Carrier Clock and NRZ40 Are Asynchronous)

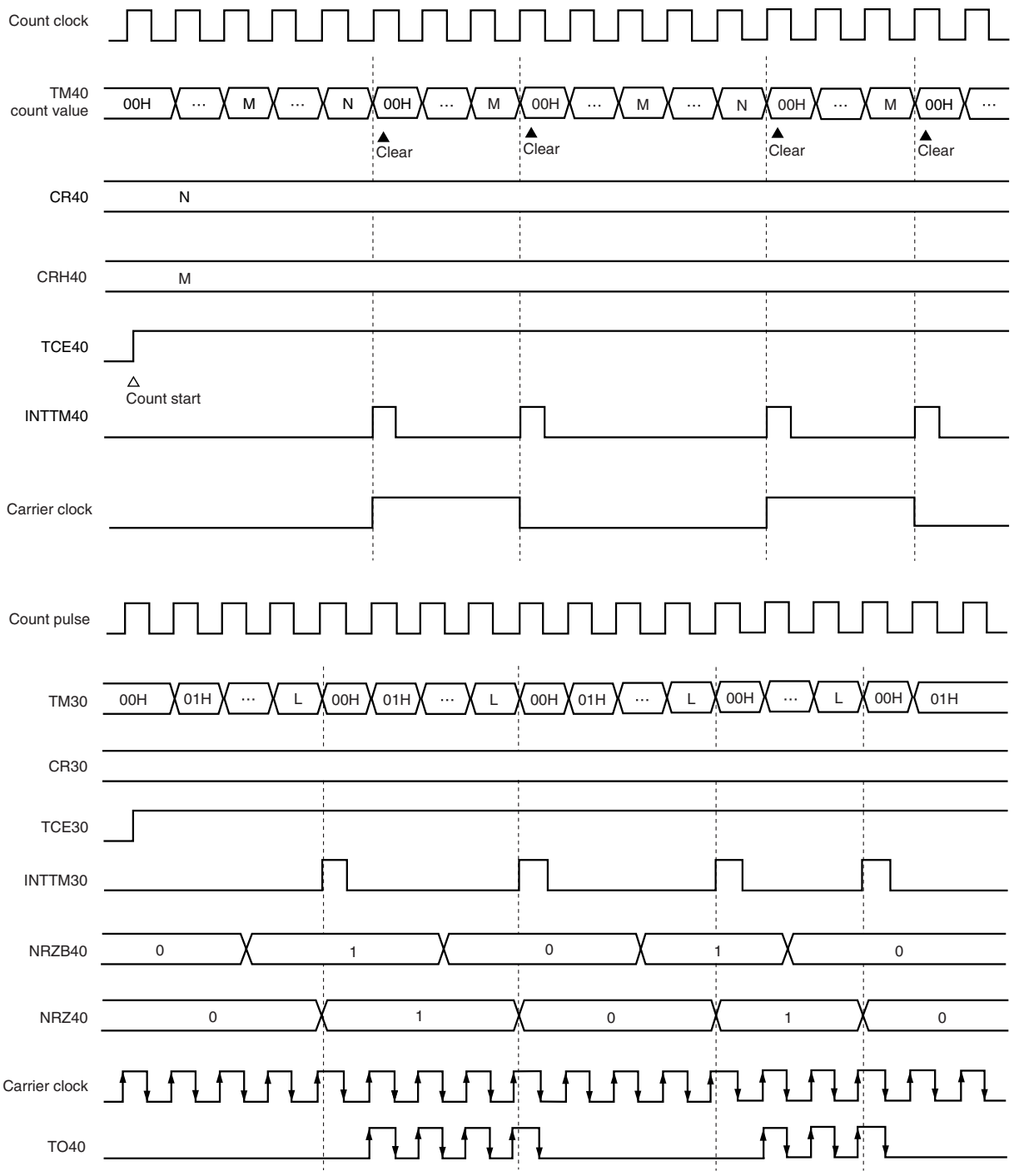
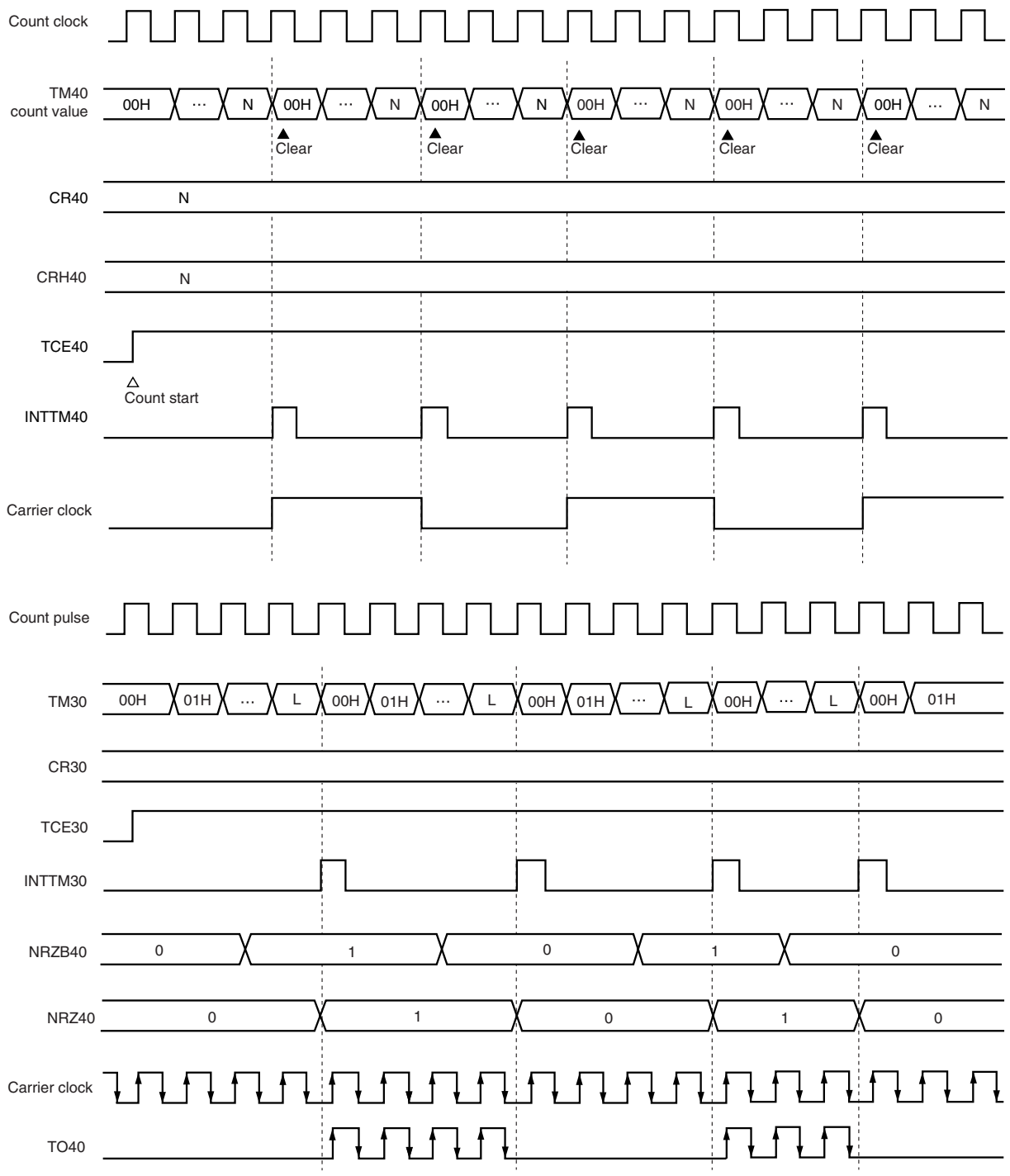


Figure 7-19. Timing of Carrier Generator Operation (When CR40 = CRH40 = N)



7.4.4 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.

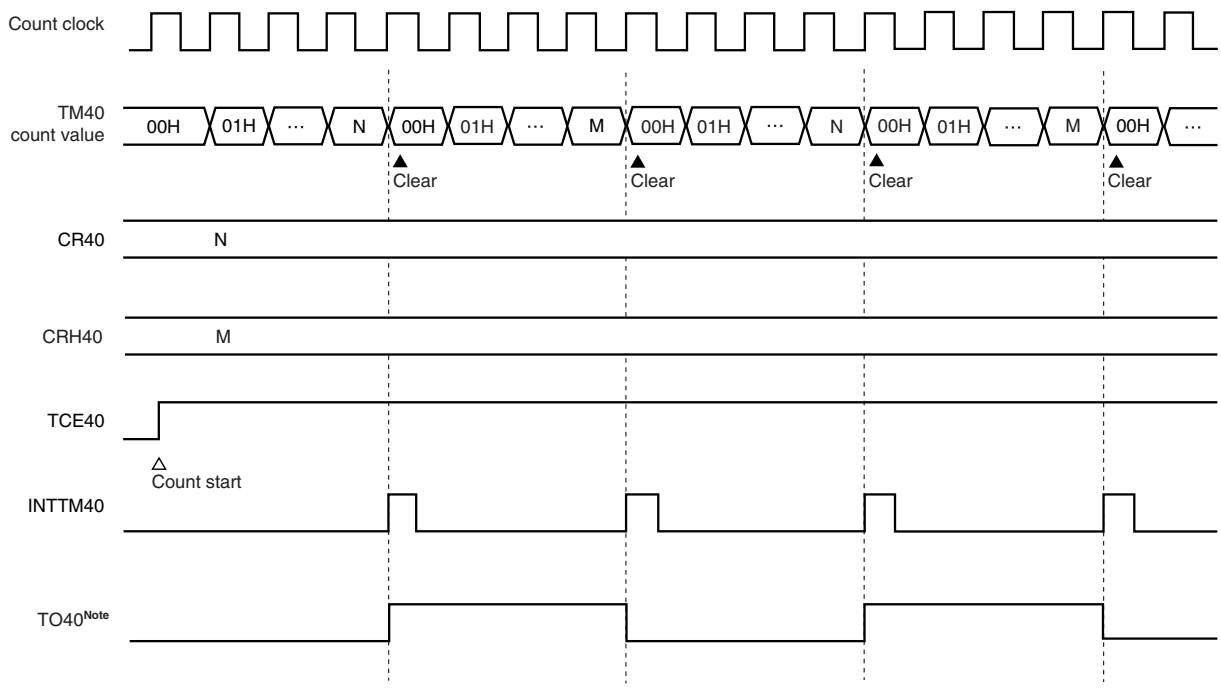
<1> Disable operation of TM40 (TCE40 = 0).
<2> Disable timer output of TO40 (TOE40 = 0).
<3> Set count values in CR40 and CRH40.
<4> Set the operation mode of timer 40 to carrier generator mode (see **Figure 7-5**).
<5> Set the count clock for timer 40.
<6> Set P27 to output mode (PM27 = 0) and the P27 output latch to 0 and enable timer output of TO40 (TOE40 = 1).
<7> Enable the operation of TM40 (TCE40 = 1).

The operation in the PWM output mode is as follows.

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

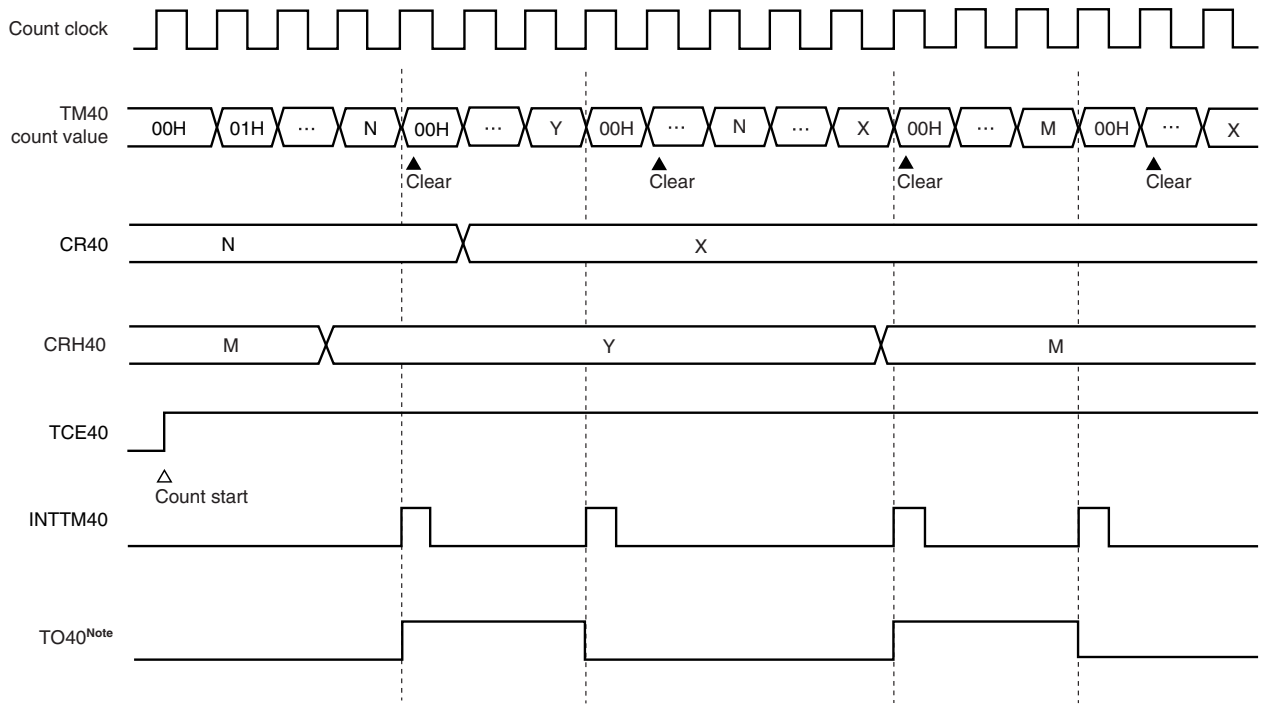
A pulse of any duty ratio is output by repeating <1> to <4> above. Figures 7-20 and 7-21 show the operation timing in the PWM output mode.

Figure 7-20. PWM Output Mode Timing (Basic Operation)



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

Figure 7-21. PWM Output Mode Timing (When CR40 and CRH40 Are Overwritten)



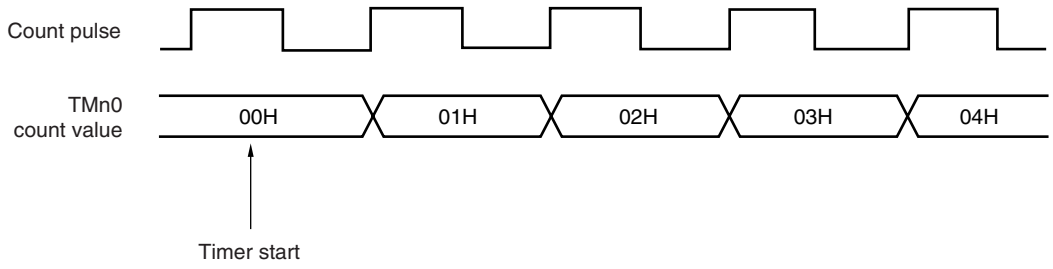
Note The initial value of TO40 is low level when output is enabled (TOE40 = 1).

7.5 Notes on Using 8-Bit Timers 30, 40

(1) Error on starting timer

An error of up to 1 clock is included in the time between the timer being started and a match signal being generated. This is because 8-bit timer counter n0 (TMn0) is started asynchronously to the count pulse.

Figure 7-22. Start Timing of 8-Bit Timer Counter



Remark n = 3, 4

8.1 8-Bit Remote Control Timer 50 Functions

The 8-bit remote control timer 50 has a pulse width measurement function with a resolution of 8 bits.

Pulse width is measured from a difference in count value when the valid edge has been detected while the timer operates in the free-running mode.

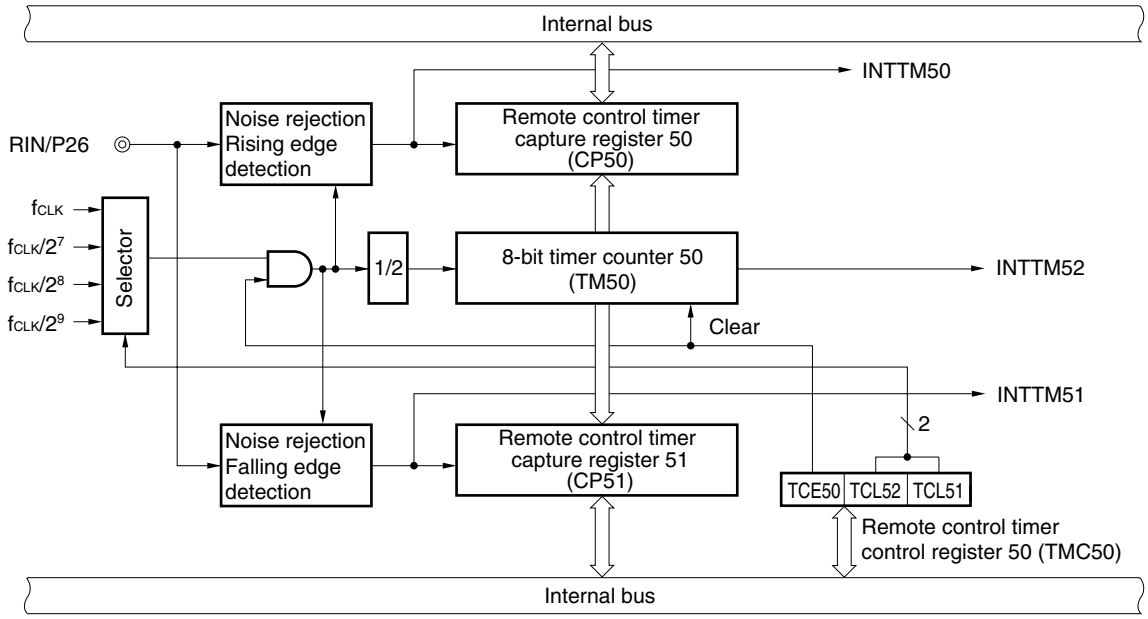
8.2 8-Bit Remote Control Timer 50 Configuration

The 8-bit remote control timer 50 consists of the following hardware.

Table 8-1. Configuration of 8-Bit Remote Control Timer 50

Item	Configuration
Timer counter	8 bits × 1
Register	Remote control timer capture register: × 2 (CP50 and CP51)
Control register	Remote control timer control register 50 (TMC50)

Figure 8-1. Block Diagram of 8-Bit Remote Control Timer 50



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

(1) Remote control timer capture registers (CP50 and CP51)

These 8-bit registers capture the contents of the 8-bit timer counter 50 (TM50).

The capture operation is performed in synchronization with the valid edge input to the RIN pin (capture trigger). The contents of CP50 are retained until the next rising edge of the RIN pin is detected. The contents of CP51 are retained until the next falling edge of the RIN pin is detected.

CP50 and CP51 can be read by using an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CP50 and CP51 to 00H.

(2) 8-bit timer counter 50 (TM50)

This 8-bit register counts the count pulse.

$\overline{\text{RESET}}$ input or clearing the TCE50 bit clears TM50 to 00H.

8.3 Registers Controlling 8-Bit Remote Control Timer 50

The following register controls the 8-bit remote control timer.

(1) Remote control timer control register 50 (TMC50)

This register enables or disables the operation of the 8-bit timer counter 50 (TM50), and sets the count clock.

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

$\overline{\text{RESET}}$ input clears TMC50 to 00H.

Figure 8-2. Format of Remote Control Timer Control Register 50

Symbol	< 7 >	6	5	4	3	2	1	0	Address	After reset	R/W
TMC50	TCE50	0	0	0	0	0	TCL501	TCL500	FF54H	00H	R/W

TCE50	TM50 count operation control
0	Clears counter to 0 and stops operation
1	Starts count operation

TCL501	TCL500	TM50 count clock selection	
		@f_X = 5.0 MHz operation	@f_{CC} = 2.0 MHz operation
0	0	$f_X/2^{10}$ (4.88 kHz)	$f_{CC}/2^{10}$ (1.95 kHz)
0	1	$f_X/2^9$ (9.77 kHz)	$f_{CC}/2^9$ (3.91 kHz)
1	0	$f_X/2^8$ (19.5 kHz)	$f_{CC}/2^8$ (7.81 kHz)
1	1	$f_X/2$ (2.5 kHz)	$f_{CC}/2$ (1.0 MHz)

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

2. Be sure to stop the timer operation (TCE50 = 0) before changing the count clock.

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

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

8.4 Operation of 8-Bit Remote Control Timer 50

The 8-bit remote control timer 50 operates as a pulse width measuring circuit.

The width of a high-level or low-level external pulse input to the RIN pin is measured by operating the 8-bit timer counter 50 (TM50) in the free-running mode.

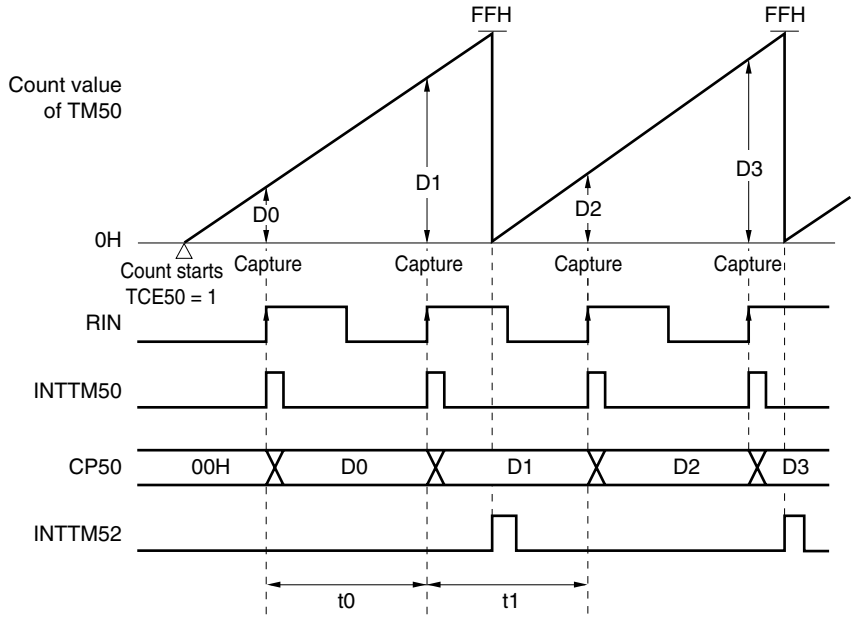
Noise with short pulse width can be detected since the detection of the valid edge is sampled every 2 cycles of the count clock selected by TCL500 and TCL501, and the capture operation is not performed until the valid level has been detected two times. Therefore, the pulse width input to the TI pin must be 5 or more of the count clock set by TCL500 and TCL501, regardless of whether the level is high or low. If the pulse width is less than 5 clocks, it cannot be detected, and the capture operation is not performed.

The value of timer counter 50 (TM50) being counted is loaded to and retained in the capture registers (CP50 and CP51) in synchronization with the valid edge of the pulse input to the RIN pin, as shown in Figure 8-3.

Figure 8-3 shows the timing of pulse width measurement.

Figure 8-3. Pulse Width Measurement Timing (1/2)

(1) To measure pulse width in synchronization with rising edge



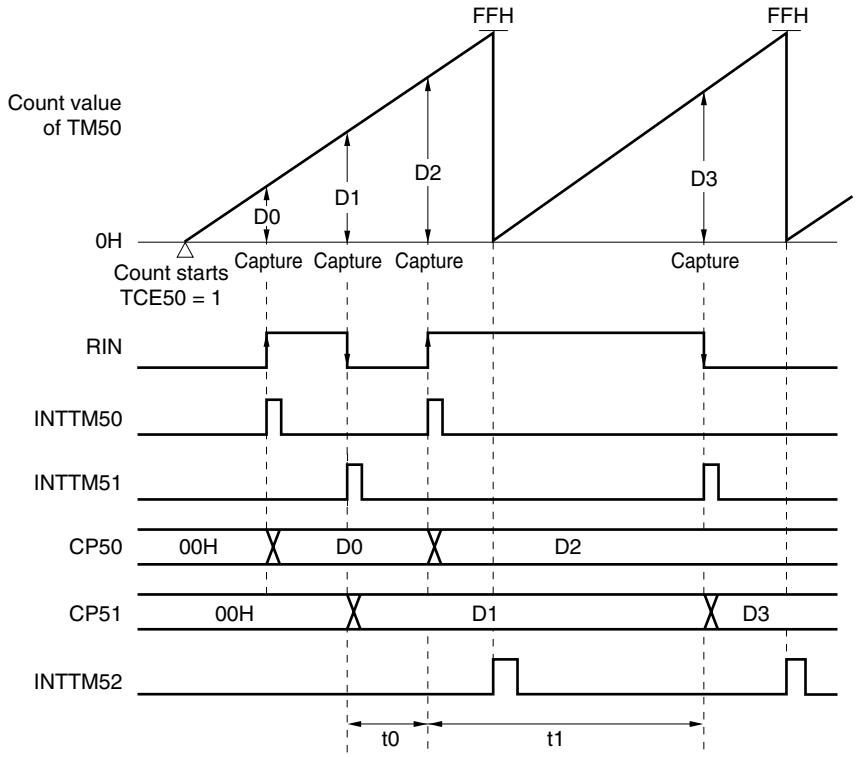
Remark $t0 = (D1 - D0) \times 1/f_{COUNT}$

$t1 = (100H - D1 + D2) \times 1/f_{COUNT}$

f_{COUNT}: Count clock frequency set by TCL500 and TCL501

Figure 8-3. Pulse Width Measurement Timing (2/2)

(2) Measure pulse width in synchronization with both rising and falling edges



Remark $t0 = (D2 - D1) \times 1/f_{COUNT}$

$t1 = (100H - D2 + D3) \times 1/f_{COUNT}$

f_{COUNT}: Count clock frequency set by TCL500 and TCL501

CHAPTER 9 SOUND GENERATOR

9.1 Functions of Sound Generator

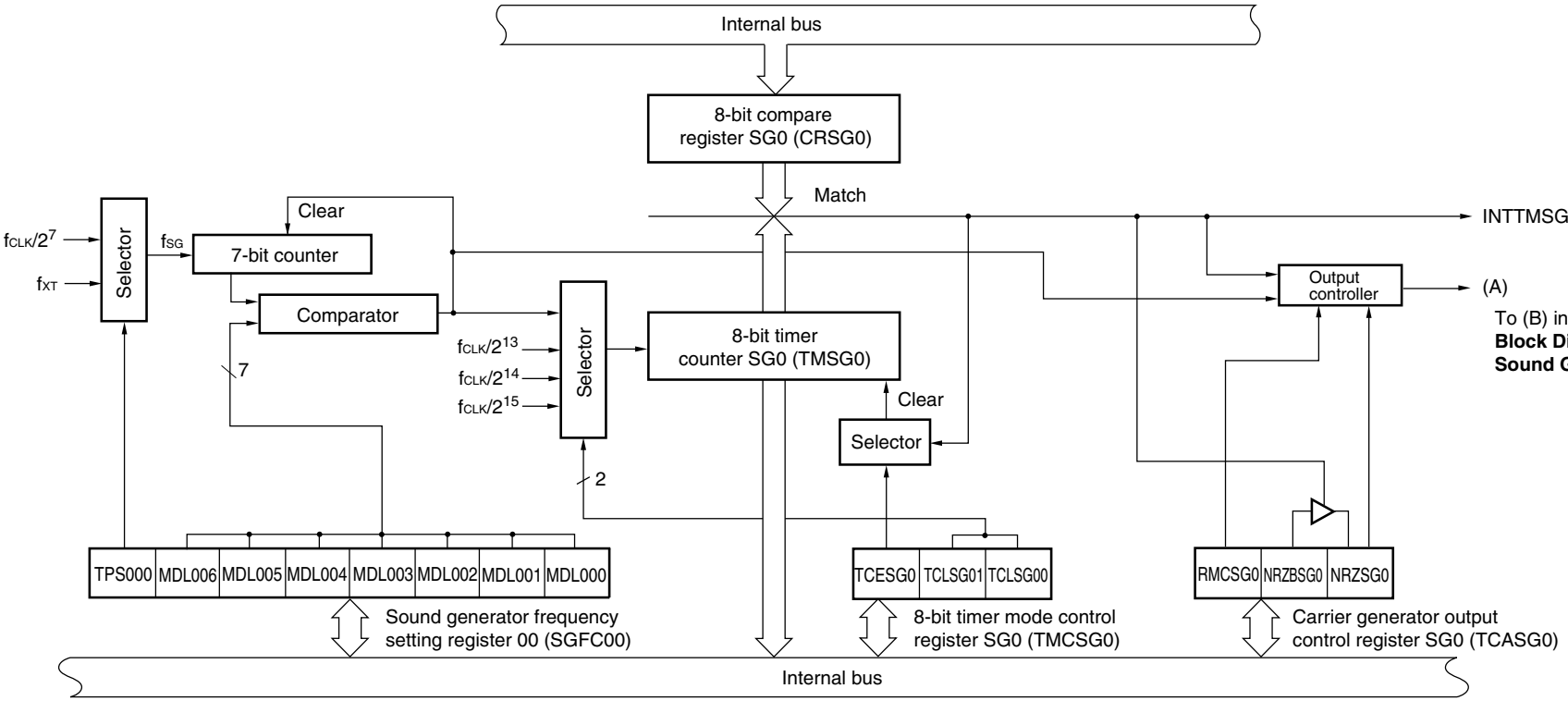
The second generator outputs an interrupt based on a match between a compare register and a timer counter. It also outputs the sound generator frequency from an arbitrary port.

9.2 Configuration of Sound Generator

Table 9-1. Configuration of Sound Generator

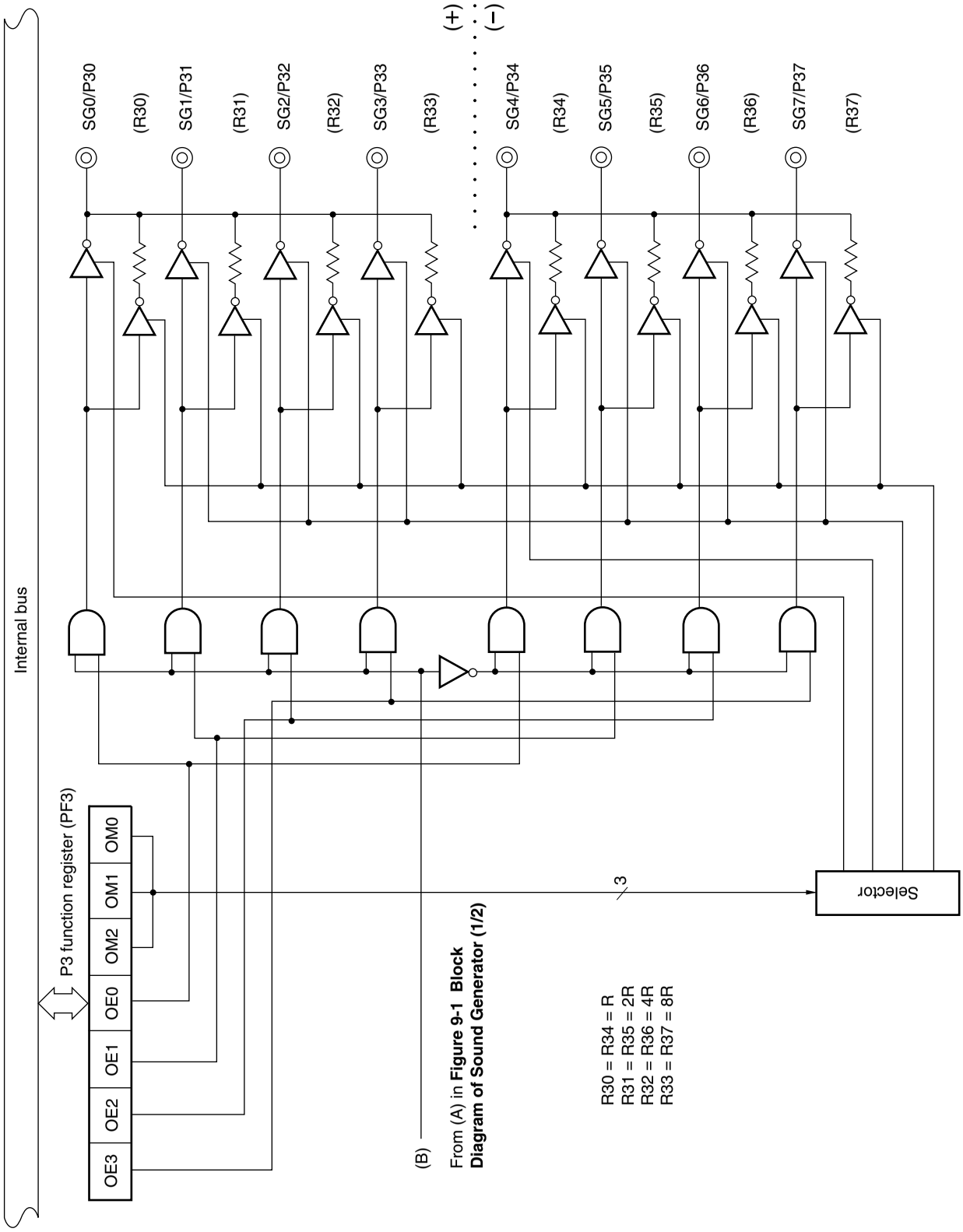
Item	Configuration
Timer counter	8 bits × 1 (TMSG0)
Registers	Compare registers: 8 bits × 1 (CRSG0)
Control registers	Sound generator frequency setting register 00 (SGFC00) Carrier generator output control register SG0 (TCASG0) 8-bit timer mode control register SG0 (TMCSG0) P3 function register (PF3)

Figure 9-1. Block Diagram of Sound Generator (1/2)



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

Figure 9-1. Block Diagram of Sound Generator (2/2)



(1) 8-bit compare register SG0 (CRSG0)

This 8-bit register is used to continually compare the value set to the CRSG0 with the count value in 8-bit timer counter SG0 (TMSG0) and to issue an interrupt request (INTTMSG0) when a match occurs.

CRSG0 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes CRSG0 undefined.

(2) 8-bit timer counter SG0 (TMSG0)

This 8-bit register is used to count a count pulse.

TMSG0 is read with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TMSG0 to 00H.

9.3 Control Registers for Sound Generator

The following four registers are used to control sound generator.

- 8-bit timer mode control register SG0 (TMCSG0)
- Carrier generator output control register SG0 (TCASG0)
- Sound generator frequency setting register 00 (SGFC00)
- P3 function register (PF3)

(1) 8-bit timer mode control register SG0 (TMCSG0)

This register enables/stops operation of 8-bit timer counter SG0 (TMSG0) and sets the count clock of TMSG0.

TMCSG0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TMCSG0 to 00H.

Figure 9-2. Format of 8-Bit Timer Mode Control Register SG0

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
TMCSG0	TCESG0	0	0	TCLSG01	TCLSG00	0	0	0	FF6BH	00H	R/W

TCESG0	Control of TMSG0 count operation
0	TMSG0 count value cleared and operation stopped
1	Count operation started

TCLSG01	TCLSG00	Selection of TMSG0 count clock	
		@ fx = 5.0 MHz operation	@ fcc = 2.0 MHz operation
0	0	$f_x/2^{13}$ (610 Hz)	$f_{cc}/2^{13}$ (244 Hz)
0	1	$f_x/2^{14}$ (305 Hz)	$f_{cc}/2^{14}$ (122 Hz)
1	0	$f_x/2^{15}$ (152 Hz)	$f_{cc}/2^{15}$ (61 Hz)
1	1	Carrier clock specified by SGFC00	

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

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

(2) Carrier generator output control register SG0 (TCASG0)

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

TCASG0 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TCASG0 to 00H.

Figure 9-3. Format of Carrier Generator Output Control Register SG0

Symbol	7	6	5	4	3	<2>	<1>	<0>	Address	After reset	R/W
TCASG0	0	0	0	0	0	RMCSG0	NRZBSG0	NRZSG0	FF6CH	00H	W

RMCSG0	NRZSG0	Control of remote control output
0	0	Low-level output
0	1	Carrier output specified by SGFC00
1	0	Low-level output
1	1	High-level output

NRZBSG0	This is the bit that stores the next NRZSG0 data to be output. When a match signal occurs (for a match between TMSG0 and CRSG0), the data is transferred to NRZSG0.

Caution Be sure to set bits 3 to 7 to 0.

(3) Sound generator frequency setting register 00 (SGFC00)

This register is used to set a frequency for the sound generator.

SGFC00 is set with AN 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets SGFC00 to 00H.

Figure 9-4. Format of Sound Generator Frequency Setting Register 00 (1/5)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
SGFC00	TPS000	MDL006	MDL005	MDL004	MDL003	MDL002	MDL001	MDL000	FF6DH	00H	R/W

TPS00	Selection of 7-bit counter source clock	
	@ fx = 5.0 MHz and fxT = 32.768 kHz operation	@ fCC = 2.0 MHz and fxT = 32.768 kHz operation
0	fxT (32.768 kHz)	
1	$f_X/2^7$ (39.1 kHz)	$f_{CC}/2^7$ (15.6 kHz)

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

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

3. fxT: Subsystem clock oscillation frequency

Figure 9-4. Format of Sound Generator Frequency Setting Register 00 (2/5)

MDL006	MDL005	MDL004	MDL003	MDL002	MDL001	MDL000	Setting of sound generator frequency
0	0	0	0	0	0	0	f_{SG} (32.768 kHz)
0	0	0	0	0	0	1	$f_{SG}/2$ (16.384 kHz)
0	0	0	0	0	1	0	$f_{SG}/3$ (10.923 kHz)
0	0	0	0	0	1	1	$f_{SG}/4$ (8.192 kHz)
0	0	0	0	1	0	0	$f_{SG}/5$ (6.554 kHz)
0	0	0	0	1	0	1	$f_{SG}/6$ (5.461 kHz)
0	0	0	0	1	1	0	$f_{SG}/7$ (4.681 kHz)
0	0	0	0	1	1	1	$f_{SG}/8$ (4.096 kHz)
0	0	0	1	0	0	0	$f_{SG}/9$ (3.641 kHz)
0	0	0	1	0	0	1	$f_{SG}/10$ (3.277 kHz)
0	0	0	1	0	1	0	$f_{SG}/11$ (2.979 kHz)
0	0	0	1	0	1	1	$f_{SG}/12$ (2.731 kHz)
0	0	0	1	1	0	0	$f_{SG}/13$ (2.521 kHz)
0	0	0	1	1	0	1	$f_{SG}/14$ (2.341 kHz)
0	0	0	1	1	1	0	$f_{SG}/15$ (2.185 kHz)
0	0	0	1	1	1	1	$f_{SG}/16$ (2.048 kHz) (do)
0	0	1	0	0	0	0	$f_{SG}/17$ (1.928 kHz) (ti)
0	0	1	0	0	0	1	$f_{SG}/18$ (1.820 kHz)
0	0	1	0	0	1	0	$f_{SG}/19$ (1.725 kHz) (la)
0	0	1	0	0	1	1	$f_{SG}/20$ (1.638 kHz)
0	0	1	0	1	0	0	$f_{SG}/21$ (1.560 kHz) (so)
0	0	1	0	1	0	1	$f_{SG}/22$ (1.489 kHz)
0	0	1	0	1	1	0	$f_{SG}/23$ (1.425 kHz)
0	0	1	0	1	1	1	$f_{SG}/24$ (1.365 kHz) (fa)
0	0	1	1	0	0	0	$f_{SG}/25$ (1.311 kHz) (mi)
0	0	1	1	0	0	1	$f_{SG}/26$ (1.260 kHz)
0	0	1	1	0	1	0	$f_{SG}/27$ (1.214 kHz)
0	0	1	1	0	1	1	$f_{SG}/28$ (1.170 kHz) (re)
0	0	1	1	1	0	0	$f_{SG}/29$ (1.130 kHz)
0	0	1	1	1	0	1	$f_{SG}/30$ (1.092 kHz)
0	0	1	1	1	1	0	$f_{SG}/31$ (1.057 kHz) (do)
0	0	1	1	1	1	1	$f_{SG}/32$ (1.024 kHz)

Remarks 1. f_{SG}: Oscillation frequency for sound generator

2. Values in parentheses apply to the case when $f_{SG} = f_{XT}$ (32.768 kHz) is selected.

Figure 9-4. Format of Sound Generator Frequency Setting Register 00 (3/5)

MDL006	MDL005	MDL004	MDL003	MDL002	MDL001	MDL000	Setting of sound generator frequency
0	1	0	0	0	0	0	$f_{SG}/33$ (993 Hz)
0	1	0	0	0	0	1	$f_{SG}/34$ (964 Hz)
0	1	0	0	0	1	0	$f_{SG}/35$ (936 Hz)
0	1	0	0	0	1	1	$f_{SG}/36$ (910 Hz)
0	1	0	0	1	0	0	$f_{SG}/37$ (886 Hz)
0	1	0	0	1	0	1	$f_{SG}/38$ (862 Hz)
0	1	0	0	1	1	0	$f_{SG}/39$ (840 Hz)
0	1	0	0	1	1	1	$f_{SG}/40$ (819 Hz)
0	1	0	1	0	0	0	$f_{SG}/41$ (799 Hz)
0	1	0	1	0	0	1	$f_{SG}/42$ (780 Hz)
0	1	0	1	0	1	0	$f_{SG}/43$ (762 Hz)
0	1	0	1	0	1	1	$f_{SG}/44$ (745 Hz)
0	1	0	1	1	0	0	$f_{SG}/45$ (728 Hz)
0	1	0	1	1	0	1	$f_{SG}/46$ (712 Hz)
0	1	0	1	1	1	0	$f_{SG}/47$ (697 Hz)
0	1	0	1	1	1	1	$f_{SG}/48$ (683 Hz)
0	1	1	0	0	0	0	$f_{SG}/49$ (669 Hz)
0	1	1	0	0	0	1	$f_{SG}/50$ (655 Hz)
0	1	1	0	0	1	0	$f_{SG}/51$ (643 Hz)
0	1	1	0	0	1	1	$f_{SG}/52$ (630 Hz)
0	1	1	0	1	0	0	$f_{SG}/53$ (618 Hz)
0	1	1	0	1	0	1	$f_{SG}/54$ (607 Hz)
0	1	1	0	1	1	0	$f_{SG}/55$ (596 Hz)
0	1	1	0	1	1	1	$f_{SG}/56$ (585 Hz)
0	1	1	1	0	0	0	$f_{SG}/57$ (575 Hz)
0	1	1	1	0	0	1	$f_{SG}/58$ (565 Hz)
0	1	1	1	0	1	0	$f_{SG}/59$ (555 Hz)
0	1	1	1	0	1	1	$f_{SG}/60$ (546 Hz)
0	1	1	1	1	0	0	$f_{SG}/61$ (537 Hz)
0	1	1	1	1	0	1	$f_{SG}/62$ (529 Hz)
0	1	1	1	1	1	0	$f_{SG}/63$ (520 Hz)
0	1	1	1	1	1	1	$f_{SG}/64$ (512 Hz)

Remarks 1. f_{SG}: Oscillation frequency for sound generator

2. Values in parentheses apply to the case when $f_{SG} = f_{XT}$ (32.768 kHz) is selected.

Figure 9-4. Format of Sound Generator Frequency Setting Register 00 (4/5)

MDL006	MDL005	MDL004	MDL003	MDL002	MDL001	MDL000	Setting of sound generator frequency
1	0	0	0	0	0	0	$f_{SG}/65$ (504 Hz)
1	0	0	0	0	0	1	$f_{SG}/66$ (496 Hz)
1	0	0	0	0	1	0	$f_{SG}/67$ (489 Hz)
1	0	0	0	0	1	1	$f_{SG}/68$ (482 Hz)
1	0	0	0	1	0	0	$f_{SG}/69$ (475 Hz)
1	0	0	0	1	0	1	$f_{SG}/70$ (468 Hz)
1	0	0	0	1	1	0	$f_{SG}/71$ (462 Hz)
1	0	0	0	1	1	1	$f_{SG}/72$ (455 Hz)
1	0	0	1	0	0	0	$f_{SG}/73$ (449 Hz)
1	0	0	1	0	0	1	$f_{SG}/74$ (443 Hz)
1	0	0	1	0	1	0	$f_{SG}/75$ (437 Hz)
1	0	0	1	0	1	1	$f_{SG}/76$ (431 Hz)
1	0	0	1	1	0	0	$f_{SG}/77$ (426 Hz)
1	0	0	1	1	0	1	$f_{SG}/78$ (420 Hz)
1	0	0	1	1	1	0	$f_{SG}/79$ (415 Hz)
1	0	0	1	1	1	1	$f_{SG}/80$ (410 Hz)
1	0	1	0	0	0	0	$f_{SG}/81$ (405 Hz)
1	0	1	0	0	0	1	$f_{SG}/82$ (400 Hz)
1	0	1	0	0	1	0	$f_{SG}/83$ (395 Hz)
1	0	1	0	0	1	1	$f_{SG}/84$ (390 Hz)
1	0	1	0	1	0	0	$f_{SG}/85$ (386 Hz)
1	0	1	0	1	0	1	$f_{SG}/86$ (381 Hz)
1	0	1	0	1	1	0	$f_{SG}/87$ (377 Hz)
1	0	1	0	1	1	1	$f_{SG}/88$ (372 Hz)
1	0	1	1	0	0	0	$f_{SG}/89$ (368 Hz)
1	0	1	1	0	0	1	$f_{SG}/90$ (364 Hz)
1	0	1	1	0	1	0	$f_{SG}/91$ (360 Hz)
1	0	1	1	0	1	1	$f_{SG}/92$ (356 Hz)
1	0	1	1	1	0	0	$f_{SG}/93$ (352 Hz)
1	0	1	1	1	0	1	$f_{SG}/94$ (349 Hz)
1	0	1	1	1	1	0	$f_{SG}/95$ (345 Hz)
1	0	1	1	1	1	1	$f_{SG}/96$ (341 Hz)

Remarks 1. f_{SG}: Oscillation frequency for sound generator

2. Values in parentheses apply to the case when $f_{SG} = f_{XT}$ (32.768 kHz) is selected.

Figure 9-4. Format of Sound Generator Frequency Setting Register 00 (5/5)

MDL006	MDL005	MDL004	MDL003	MDL002	MDL001	MDL000	Setting of sound generator frequency
1	1	0	0	0	0	0	$f_{SG}/97$ (338 Hz)
1	1	0	0	0	0	1	$f_{SG}/98$ (334 Hz)
1	1	0	0	0	1	0	$f_{SG}/99$ (331 Hz)
1	1	0	0	0	1	1	$f_{SG}/100$ (328 Hz)
1	1	0	0	1	0	0	$f_{SG}/101$ (324 Hz)
1	1	0	0	1	0	1	$f_{SG}/102$ (321 Hz)
1	1	0	0	1	1	0	$f_{SG}/103$ (318 Hz)
1	1	0	0	1	1	1	$f_{SG}/104$ (315 Hz)
1	1	0	1	0	0	0	$f_{SG}/105$ (312 Hz)
1	1	0	1	0	0	1	$f_{SG}/106$ (309 Hz)
1	1	0	1	0	1	0	$f_{SG}/107$ (306 Hz)
1	1	0	1	0	1	1	$f_{SG}/108$ (303 Hz)
1	1	0	1	1	0	0	$f_{SG}/109$ (300 Hz)
1	1	0	1	1	0	1	$f_{SG}/110$ (298 Hz)
1	1	0	1	1	1	0	$f_{SG}/111$ (295 Hz)
1	1	0	1	1	1	1	$f_{SG}/112$ (293 Hz)
1	1	1	0	0	0	0	$f_{SG}/113$ (290 Hz)
1	1	1	0	0	0	1	$f_{SG}/114$ (287 Hz)
1	1	1	0	0	1	0	$f_{SG}/115$ (285 Hz)
1	1	1	0	0	1	1	$f_{SG}/116$ (282 Hz)
1	1	1	0	1	0	0	$f_{SG}/117$ (280 Hz)
1	1	1	0	1	0	1	$f_{SG}/118$ (278 Hz)
1	1	1	0	1	1	0	$f_{SG}/119$ (275 Hz)
1	1	1	0	1	1	1	$f_{SG}/120$ (273 Hz)
1	1	1	1	0	0	0	$f_{SG}/121$ (271 Hz)
1	1	1	1	0	0	1	$f_{SG}/122$ (269 Hz)
1	1	1	1	0	1	0	$f_{SG}/123$ (266 Hz)
1	1	1	1	0	1	1	$f_{SG}/124$ (264 Hz)
1	1	1	1	1	0	0	$f_{SG}/125$ (262 Hz)
1	1	1	1	1	0	1	$f_{SG}/126$ (260 Hz)
1	1	1	1	1	1	0	$f_{SG}/127$ (258 Hz)
1	1	1	1	1	1	1	$f_{SG}/128$ (256 Hz)

Remarks 1. f_{SG}: Oscillation frequency for sound generator

2. Values in parentheses apply to the case when $f_{SG} = f_{XT}$ (32.768 kHz) is selected.

(4) P3 function register (PF3)

This register is used to specify a buzzer output pin for the sound generator.

PF3 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PF3 to 00H.

Figure 9-5. Format of P3 Function Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PF3	OE3	OE2	OE1	OE0	0	OM2	OM1	OM0	FF53H	00H	R/W

OEn	Control of P30/SG0 to P37/SG7 output
0	Output of SGn and SGn+4 pins disabled
1	Output of SGn and SGn+4 pins enabled

OM2	OM1	OM0	Selection of P30/SG0 to P37/SG7 output mode
0	0	0	General-purpose port mode (P30 to P37)
0	0	1	Buzzer mode 0 Note 1
0	1	0	Buzzer mode 1 Note 2
0	1	1	Buzzer mode 2 Note 3
1	0	0	Buzzer mode 3 Note 4
Other than above			Setting prohibited

Notes 1. Pin P30/SG0 is used for buzzer output. The other pins, P31/SG1 to P37/SG7, are used as ordinary general-purpose port pins.

2. Pins P30/SG0 and P34/SG4 are used for buzzer output. P34/SG4 outputs the inverted level of P30/SG0. On-chip resistors are not connected for pins P30/SG0 to P37/SG7. The other pins, P31/SG1 to P33/SG3 and P35/SG5 to P37/SG7, are used as ordinary general-purpose port pins.

3. Pins P30/SG0 and P34/SG4 are used for buzzer output. P34/SG4 outputs the inverted level of P30/SG0. On-chip resistors are connected for pins P30/SG0 to P37/SG7. The other pins, P31/SG1 to P33/SG3 and P35/SG5 to P37/SG7, are used as ordinary general-purpose port pins.

4. Pins P30/SG0 to P37/SG7 are used for buzzer output. P34/SG4, P35/SG5, P36/SG6, and P37/SG7 output the inverted level of P30/SG0, P31/SG1, P32/SG2, and P33/SG3, respectively. On-chip resistors are not connected for these pins (P30/SG0 to P37/SG7).

Remark n: 0 to 3

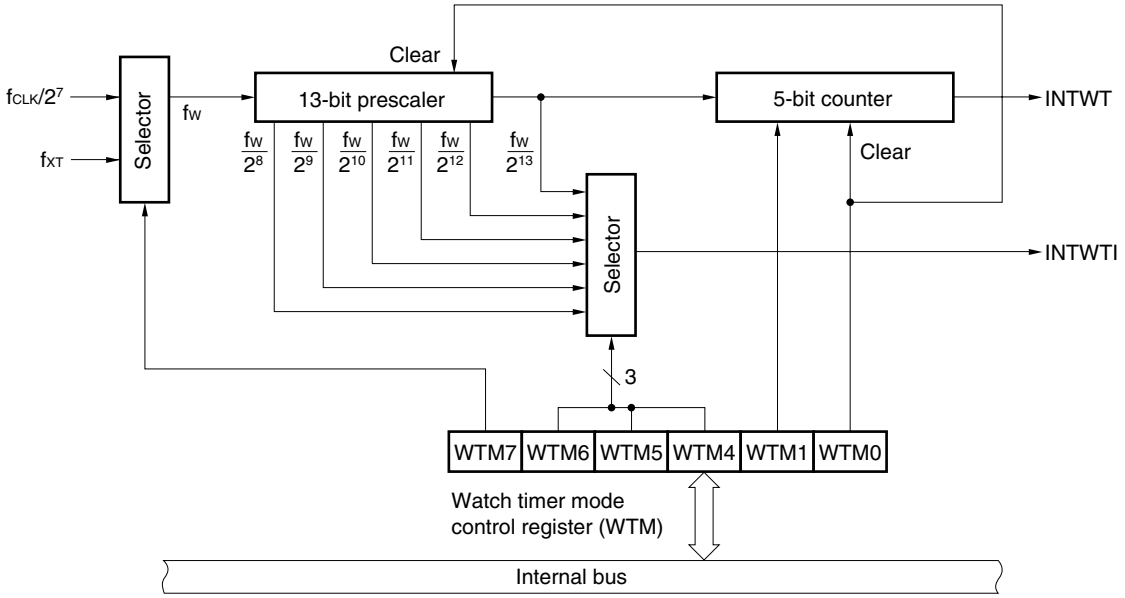
10.1 Watch Timer Functions

The watch timer has the following functions.

- Watch timer
- Interval timer

The watch and interval timers can be used at the same time.
Figure 10-1 shows a block diagram of the watch timer.

Figure 10-1. Block Diagram of Watch Timer



Remark f_{CLK}: fx or fCC

(1) Watch timer

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

Caution When the main system clock is operating at 5.0 MHz or 2.0 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 10-1. Interval Time of Interval Timer

Interval Time	@ f_X = 5.0 MHz Operation	@ f_X = 4.19 MHz Operation	@ f_{CC} = 2.0 MHz Operation	@ f_{XT} = 32.768 kHz Operation
$2^8 \times 1/f_W$	6.55 ms	7.81 ms	16.4 ms	7.81 ms
$2^9 \times 1/f_W$	13.1 ms	15.6 ms	32.8 ms	15.6 ms
$2^{10} \times 1/f_W$	26.2 ms	31.3 ms	65.5 ms	31.3 ms
$2^{11} \times 1/f_W$	52.4 ms	62.5 ms	131 ms	62.5 ms
$2^{12} \times 1/f_W$	104.9 ms	125 ms	262 ms	125 ms
$2^{13} \times 1/f_W$	209.7 ms	250 ms	524 ms	250 ms

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

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

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

4. f_{XT}: Subsystem clock oscillation frequency

10.2 Watch Timer Configuration

The watch timer includes the following hardware.

Table 10-2. Configuration of Watch Timer

Item	Configuration
Counter	5 bits × 1
Prescaler	13 bits × 1
Control register	Watch timer mode control register (WTM)

10.3 Register Controlling Watch Timer

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

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

Figure 10-2. 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 (f_W) selection	
	@f_X = 5.0 MHz and f_{XT} = 32.768 kHz operation	@f_{CC} = 2.0 MHz and f_{XT} = 32.768 kHz operation
0	$f_X/2^7$ (39.1 kHz)	$f_{CC}/2^7$ (15.6 kHz)
1	f_{XT} (32.768 kHz)	

WTM6	WTM5	WTM4	Prescaler interval selection
0	0	0	$2^8/f_W$
0	0	1	$2^9/f_W$
0	1	0	$2^{10}/f_W$
0	1	1	$2^{11}/f_W$
1	0	0	$2^{12}/f_W$
1	0	1	$2^{13}/f_W$
Other than above			Setting prohibited

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

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

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

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

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

4. f_{XT}: Subsystem clock oscillation frequency

10.4 Watch Timer Operation

10.4.1 Operation as watch timer

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

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

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

When the interval timer also operates at the same time, only the watch timer can be started from 0 seconds by setting WTM1 to 0. However, an error of up to $2^{13} \times 1/f_W$ seconds may occur for the first overflow of the watch timer (INTWT) after a 0-second start because the 13-bit prescaler is not cleared in this case.

10.4.2 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 10-3. Interval Time of Interval Timer

Interval Time	@ f_X = 5.0 MHz Operation	@ f_X = 4.19 MHz Operation	@ f_{CC} = 2.0 MHz Operation	@ f_{XT} = 32.768 kHz Operation
$2^8 \times 1/f_W$	6.55 ms	7.81 ms	16.4 ms	7.81 ms
$2^9 \times 1/f_W$	13.1 ms	15.6 ms	32.8 ms	15.6 ms
$2^{10} \times 1/f_W$	26.2 ms	31.3 ms	65.5 ms	31.3 ms
$2^{11} \times 1/f_W$	52.4 ms	62.5 ms	131 ms	62.5 ms
$2^{12} \times 1/f_W$	104.9 ms	125 ms	262 ms	125 ms
$2^{13} \times 1/f_W$	209.7 ms	250 ms	524 ms	250 ms

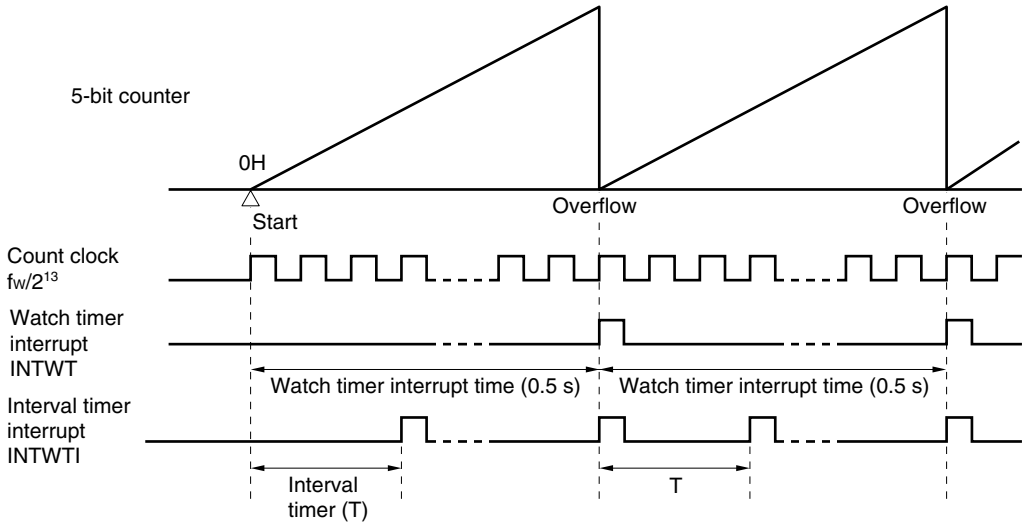
Remarks 1. f_W: Watch timer clock frequency ($f_X/2^7$, $f_{CC}/2^7$, or f_{XT})

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

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

4. f_{XT}: Subsystem clock oscillation frequency

Figure 10-3. Watch Timer/Interval Timer Operation Timing



Caution When operation of the watch timer and 5-bit counter has been enabled by setting the watch timer mode control register (WTM) (setting WTM0 (bit 0 of WTM) to 1), the time until the first interrupt request after this setting will not be exactly the same as the time set by WTM3 (bit 3 of WTM). This is because the 5-bit counter starts counting one cycle after the output of the 13-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 11 WATCHDOG TIMER

11.1 Watchdog Timer Functions

The watchdog timer has the following functions.

- Watchdog timer
- Interval timer

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

(1) Watchdog timer

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

Table 11-1. Inadvertent Program Loop Detection Time of Watchdog Timer

Inadvertent Program Loop Detection Time	@ f_X = 5.0 MHz Operation	@ f_{CC} = 2.0 MHz Operation	@ f_{XT} = 32.768 kHz Operation
$2^{13} \times 1/f_W$	210 ms	524 ms	250 ms
$2^{14} \times 1/f_W$	419 ms	1.05 s	500 ms

Remarks 1. f_W: $f_X/2^7$, $f_{CC}/2^7$, or f_{XT}

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

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

4. f_{XT}: Subsystem clock oscillation frequency

(2) Interval timer

The interval timer generates an interrupt at any preset intervals.

Table 11-2. Interval Time of Watchdog Timer

Interval Time	@ f_X = 5.0 MHz Operation	@ f_{CC} = 2.0 MHz Operation	@ f_{XT} = 32.768 kHz Operation
$2^{13} \times 1/f_W$	210 ms	524 ms	250 ms
$2^{14} \times 1/f_W$	419 ms	1.05 s	500 ms

Remarks 1. f_W: $f_X/2^7$, $f_{CC}/2^7$, or f_{XT}

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

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

4. f_{XT}: Subsystem clock oscillation frequency

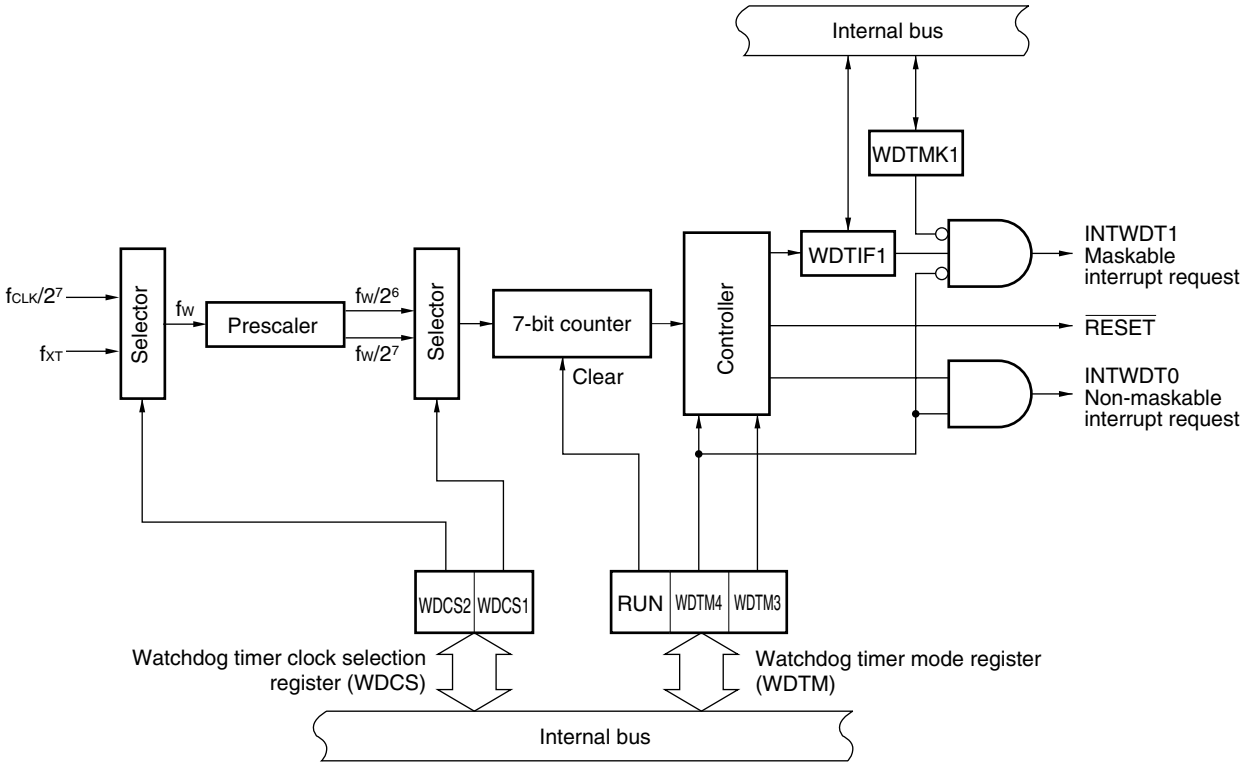
11.2 Watchdog Timer Configuration

The watchdog timer includes the following hardware.

Table 11-3. Configuration of Watchdog Timer

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

Figure 11-1. Block Diagram of Watchdog Timer



Remark f_{CLK}: fx or fCC

11.3 Registers Controlling Watchdog Timer

The watchdog timer is controlled by the following two registers.

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

(1) Watchdog timer clock selection register (WDCS)

This register sets the watchdog timer count clock.

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

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

Figure 11-2. Format of Watchdog Timer Clock Selection Register

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

WDCS2	WDCS1	Selection of count clock	
		@ f_X = 5.0 MHz and f_{XT} = 32.768 kHz operation	@ f_{CC} = 2.0 MHz and f_{XT} = 32.768 kHz operation
0	0	$f_X/2^{13}$ (610 Hz)	$f_{CC}/2^{13}$ (244 Hz)
0	1	$f_X/2^{14}$ (305 Hz)	$f_{CC}/2^{14}$ (122 Hz)
1	0	$f_{XT}/2^6$ (512 Hz)	
1	1	$f_{XT}/2^7$ (256 Hz)	

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

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

3. f_{XT}: Subsystem clock oscillation frequency

Table 11-4. Inadvertent Program Loop Detection Timer or Interval Time of Watchdog Timer

WDCS2	WDCS1	Inadvertent Program Loop Detection Time or Interval Time	
		@ f_X = 5.0 MHz and f_{XT} = 32.768 kHz Operation	@ f_{CC} = 2.0 MHz and f_{XT} = 32.768 kHz Operation
0	0	$2^{20}/f_X$ (210 ms)	$2^{20}/f_{CC}$ (524 ms)
0	1	$2^{21}/f_X$ (419 ms)	$2^{21}/f_{CC}$ (1.05 s)
1	0	$2^{13}/f_{XT}$ (250 ms)	
1	1	$2^{14}/f_{XT}$ (500 ms)	

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

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

3. f_{XT}: Subsystem clock oscillation frequency

(2) Watchdog timer mode register (WDTM)

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

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

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

Figure 11-3. 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	FF4BH	00H	R/W

RUN	Selection of operation of watchdog timer[Note 1]
0	Stops counting
1	Clears counter and starts counting

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

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

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

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

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

2. In watchdog timer mode 1 or 2, set WDTM4 to 1 after confirming that the WDTIF1 (bit 0 of interrupt request flag register 0 (IF0)) is set to 0. While WDTIF1 is 1, a non-maskable interrupt is generated upon write completion if watchdog timer mode 1 or 2 is selected.

11.4 Watchdog Timer Operation

11.4.1 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 (inadvertent program loop detection time interval) of the watchdog timer can be selected by bits 1 and 2 (WDCS1 and WDCS2) of the watchdog timer clock selection register (WDCS). By setting bit 7 (RUN) of WDTM to 1, the watchdog timer is started. Set RUN to 1 within the set inadvertent 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 inadvertent program loop detection time is exceeded, the system is reset or a non-maskable interrupt is generated by the value of bit 3 (WDTM3) of WDTM.

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

Cautions 1. The actual inadvertent 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 operation even though the main system clock is oscillating.

Table 11-5. Inadvertent Program Loop Detection Time of Watchdog Timer

WDCS2	WDCS1	Inadvertent Program Loop Detection Time	
		@ f_X = 5.0 MHz and f_{XT} = 32.768 kHz Operation	@ f_{CC} = 2.0 MHz and f_{XT} = 32.768 kHz Operation
0	0	$2^{20}/f_X$ (210 ms)	$2^{20}/f_{CC}$ (524 ms)
0	1	$2^{21}/f_X$ (419 ms)	$2^{21}/f_{CC}$ (1.05 s)
1	0	$2^{13}/f_{XT}$ (250 ms)	
1	1	$2^{14}/f_{XT}$ (500 ms)	

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

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

3. f_{XT}: Subsystem clock oscillation frequency

11.4.2 Operation as interval timer

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

Select a count clock (or interval time) by setting bits 1 and 2 (WDCS1 and WDCS2) of the watchdog timer clock selection register (WDCS). The watchdog timer starts operation as an interval timer when the RUN bit (bit 7 of WDTM) is set to 1.

In the interval timer mode, the interrupt mask flag (WDTMK1) is valid, and a maskable interrupt (INTWDT1) can be generated. The priority of INTWDT1 is set as the highest of all the maskable interrupts.

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

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

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

Table 11-6. Interval Time of Watchdog Timer

WDCS2	WDCS1	Interval Time	
		@ f_X = 5.0 MHz and f_{XT} = 32.768 kHz Operation	@ f_{CC} = 2.0 MHz and f_{XT} = 32.768 kHz Operation
0	0	$2^{20}/f_X$ (210 ms)	$2^{20}/f_{CC}$ (524 ms)
0	1	$2^{21}/f_X$ (419 ms)	$2^{21}/f_{CC}$ (1.05 s)
1	0	$2^{13}/f_{XT}$ (250 ms)	
1	1	$2^{14}/f_{XT}$ (500 ms)	

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

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

3. f_{XT}: Subsystem clock oscillation frequency

CHAPTER 12 8-BIT A/D CONVERTER

12.1 8-Bit A/D Converter Functions

The 8-bit A/D converter converts input analog voltages in to digital signals with an 8-bit resolution. It can control up to three analog input channels (ANI0 to ANI2).

A/D conversion can be started only by software.

One of analog inputs ANI0 to ANI2 is selected for A/D conversion. A/D conversion is performed repeatedly, with an interrupt request (INTAD) being issued each time an A/D conversion is completed.

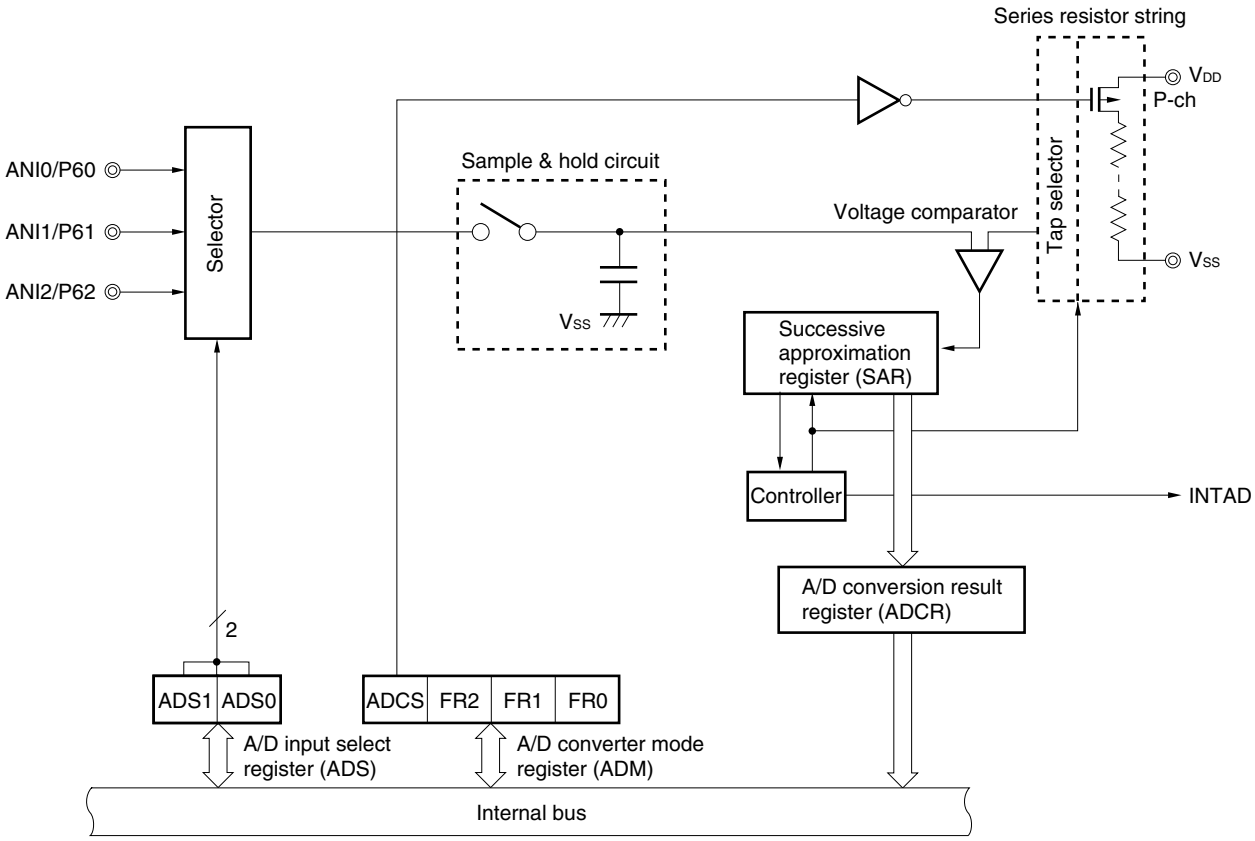
12.2 8-Bit A/D Converter Configuration

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

Table 12-1. Configuration of 8-Bit A/D Converter

Item	Configuration
Analog input	3 channels (ANI0 to ANI2)
Register	Successive approximation register (SAR) A/D conversion result register (ADCR)
Control register	A/D converter mode register (ADM) A/D input select register (ADS)

Figure 12-1. Block Diagram of 8-Bit A/D Converter



(1) Successive approximation register (SAR)

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

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

(2) A/D conversion result register (ADCR)

ADCR 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 ADCR, which is an 8-bit register that holds the result of A/D conversion.

ADCR is read with an 8-bit memory manipulation instruction.

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

(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 to ANI2 pins

Pins ANI0 to ANI2 are analog input pins for the three-channel A/D converter. They are used to receive the analog signals to be A/D converted.

Caution Do not supply pins ANI0 to ANI2 with voltages that fall outside the rated range. If a voltage equal to or greater than V_{DD} or equal to or less than V_{SS} (even if within the absolute maximum rating) is supplied to any of these pins, the conversion value for the corresponding channel will be undefined. Furthermore, the conversion values for the other channels may also be affected.

12.3 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 (ADM)
- A/D input select register (ADS)

(1) A/D converter mode register (ADM)

ADM specifies the conversion time for analog inputs. It also specifies whether to enable conversion.

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

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

Figure 12-2. Format of A/D Converter Mode Register

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
ADM	ADCS	0	FR2	FR1	FR0	0	0	0	FF80H	00H	R/W

ADCS	A/D conversion control
0	Conversion stopped
1	Conversion enabled

FR2	FR1	FR0	A/D conversion time selection[Note]	
			@fx = 5.0 MHz operation	@fcc = 2.0 MHz operation
0	0	0	288/fx (57.6 μs)	288/fcc (144 μs)
0	0	1	240/fx (48 μs)	240/fcc (120 μs)
0	1	0	192/fx (38.4 μs)	192/fcc (96 μs)
1	0	0	144/fx (28.8 μs)	144/fcc (72 μs)
1	0	1	120/fx (24 μs)	120/fcc (60 μs)
1	1	0	96/fx (19.2 μs)	96/fcc (48 μs)
Other settings			Setting prohibited	

Note The specifications of FR2, FR1, and FR0 must be such that the A/D conversion time is at least 14 μs.

Cautions 1. The result of conversion performed immediately after bit 7 (ADCS) is set is undefined.

2. The result of conversion performed after ADCS is cleared may be undefined (see 12.5 (5) Timing that makes the A/D conversion result undefined for details).

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

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

(2) A/D input select register (ADS)

ADS specifies the port used to input the analog voltages to be converted to a digital signal.

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

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

Figure 12-3. Format of A/D Input Select Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ADS	0	0	0	0	0	0	ADS1	ADS0	FF84H	00H	R/W

ADS1	ADS0	Analog input channel specification
0	0	ANI0
0	1	ANI1
1	0	ANI2
1	1	Setting prohibited

Caution Bits 2 to 7 must be fixed to 0.

12.4 8-Bit A/D Converter Operation

12.4.1 Basic operation of 8-bit A/D converter

<1> Select a channel for A/D conversion, using the A/D input select register (ADS).

<2> The voltage supplied to the selected analog input channel is sampled using the sample and hold circuit.

<3> After sampling continues for a certain period of time, the sample and 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 of V_{DD}.

<5> The series resistor string tap voltage is compared with the analog input voltage using the voltage comparator. If the analog input voltage is higher than half of V_{DD}, the MSB of the SAR is left set. If it is lower than half of 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 tap voltage is compared with the analog input voltage. Bit 6 is set or reset according to the result of comparison.

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

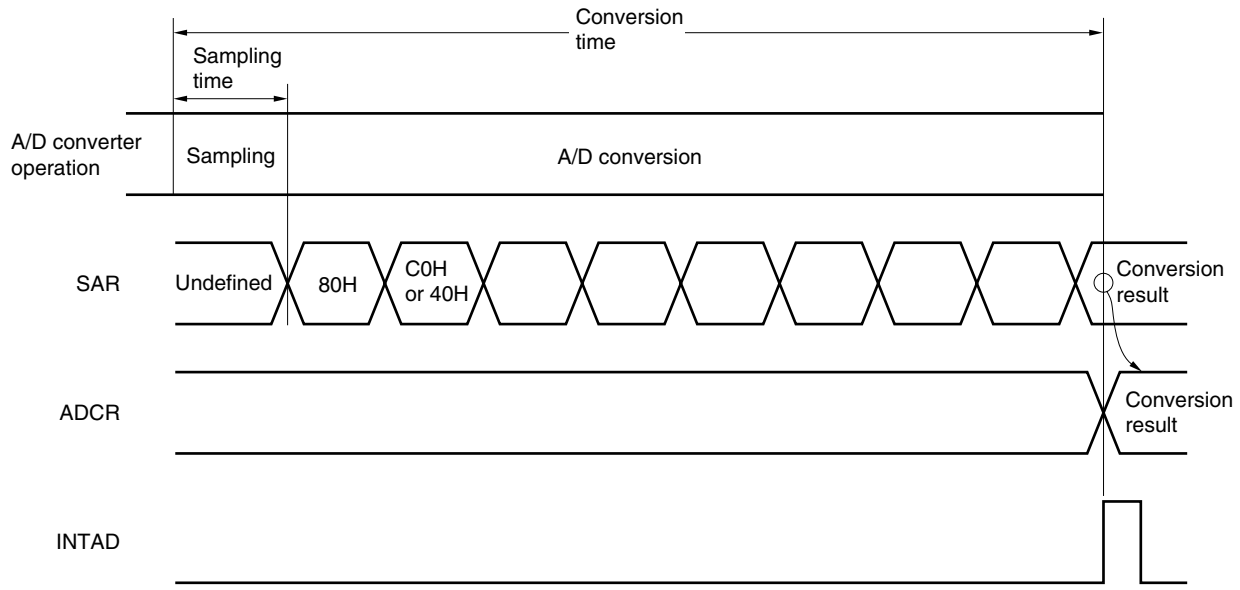
<7> Comparison is repeated until bit 0 of the SAR is reached.

<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 the A/D conversion result register (ADCR). At the same time, it is possible to generate an A/D conversion end interrupt request (INTAD).

Cautions 1. The first A/D conversion value immediately following the start of A/D conversion may be undefined.

2. When the A/D converter enters the standby mode, it stops operating.

Figure 12-4. Basic Operation of 8-Bit A/D Converter



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

If an attempt is made to write to ADM or the A/D input select register (ADS) during A/D conversion, the A/D conversion operation is canceled. In this case, operation A/D conversion is restarted from the beginning, if the ADCS is set (1).

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

12.4.2 Input voltage and conversion result

The relationship between the analog input voltage at the analog input pins (ANI0 to ANI2) and the A/D conversion result (A/D conversion result register (ADCR)) is represented by:

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

or

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

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

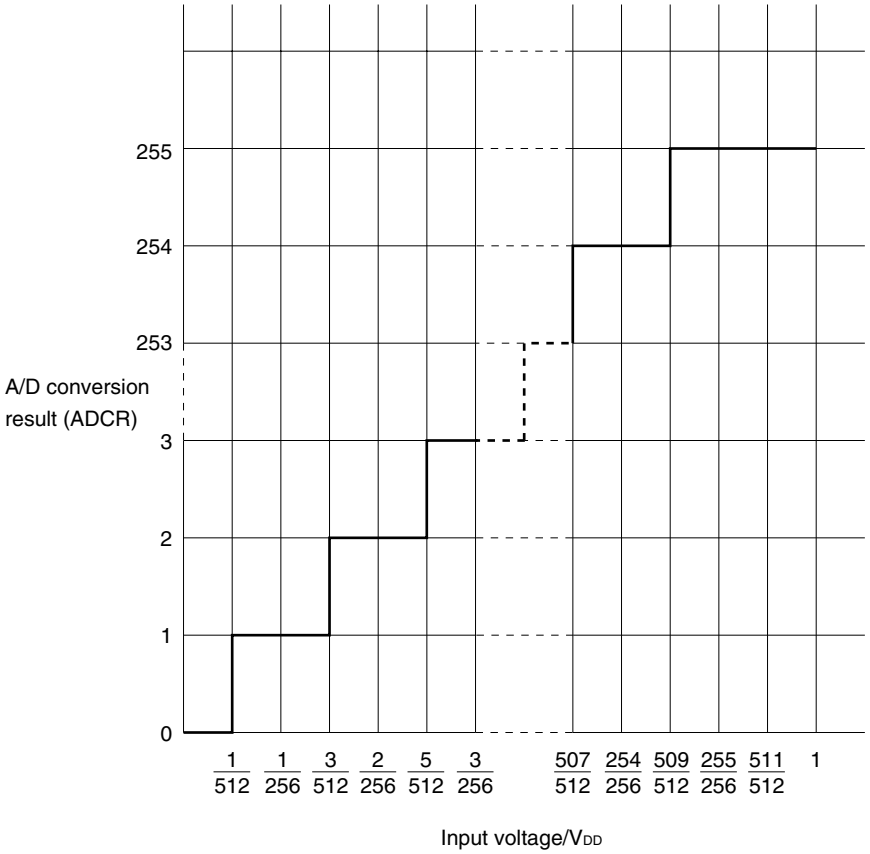
V_{IN}: Analog input voltage

V_{DD}: V_{DD} pin voltage

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

Figure 12-5 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 12-5. Relationship Between Analog Input Voltage and A/D Conversion Result



12.4.3 Operation mode of 8-bit A/D converter

The 8-bit A/D converter is initially in the select mode. In this mode, the A/D input select register (ADS) is used to select an analog input channel from ANI0 to ANI2 for A/D conversion.

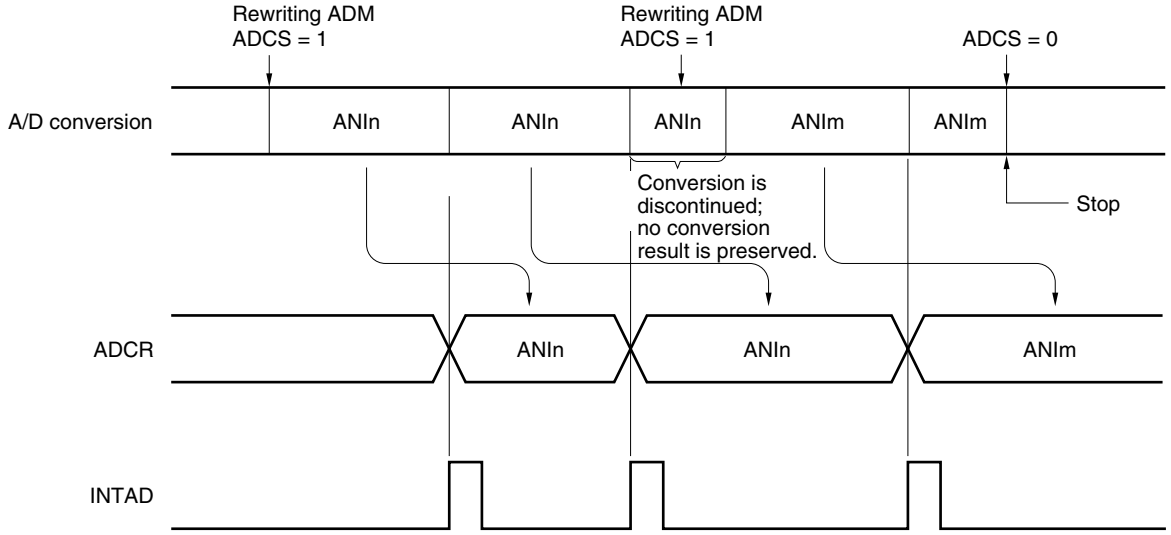
A/D conversion can be started only by software, that is, by setting the A/D converter mode register (ADM).

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

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

  Setting bit 7 (ADCS) of the A/D converter mode register (ADM) triggers A/D conversion for a voltage applied to the analog input pin specified in the A/D input select register (ADS). Upon completion of A/D conversion, the conversion result is saved to the A/D conversion result register (ADCR). At the same time, an interrupt request signal (INTAD) is generated. Once A/D conversion is activated, and completed, another session of A/D conversion is started. A/D conversion is repeated until new data is written to the ADM. If data where ADCS is 1 is written to ADM again during A/D conversion, the current session of A/D conversion is discontinued, and a new session of A/D conversion begins for the new data. If data where ADCS is 0 is written to ADM again during A/D conversion, A/D conversion is stopped immediately.

Figure 12-6. Software-Started A/D Conversion



Remarks 1. n = 0 to 2

2. m = 0 to 2

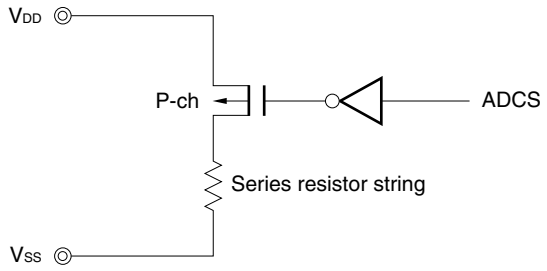
12.5 Cautions Related to 8-Bit A/D Converter

(1) Current consumption in the standby mode

When the A/D converter enters the standby mode, it stops operating. Stopping conversion (bit 7 (ADCS) of the A/D converter mode register (ADM) = 0) can reduce the current consumption.

Figure 12-7 shows how to reduce the current consumption in the standby mode.

Figure 12-7. How to Reduce Current Consumption in Standby Mode



(2) Input range for the ANI0 to ANI2 pins

Be sure to keep the input voltage at ANI0 to ANI2 within the rated value. If a voltage equal to or greater than V_{DD} or equal to or less than V_{SS} (even within the absolute maximum rating) is input to a conversion channel, the conversion output of the channel becomes undefined. This may also affect the conversion output of the other channels.

(3) Conflict

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

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

<2> Conflict between writing to ADCR at the end of conversion and writing to the A/D converter mode register (ADM) or the A/D input select register (ADS)

Writing to ADM or ADS takes precedence. A request to write to ADCR is ignored. No A/D conversion end interrupt request signal (INTAD) 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 conversion may be undefined. Be sure to poll the A/D conversion end interrupt request (INTAD) and perform processing such as discarding the first 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 at which the A/D converter stops operating 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 conversion 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 12-8 and 12-9.

Figure 12-8. Conversion Result Readout Timing (When Conversion Result Is Undefined Value)

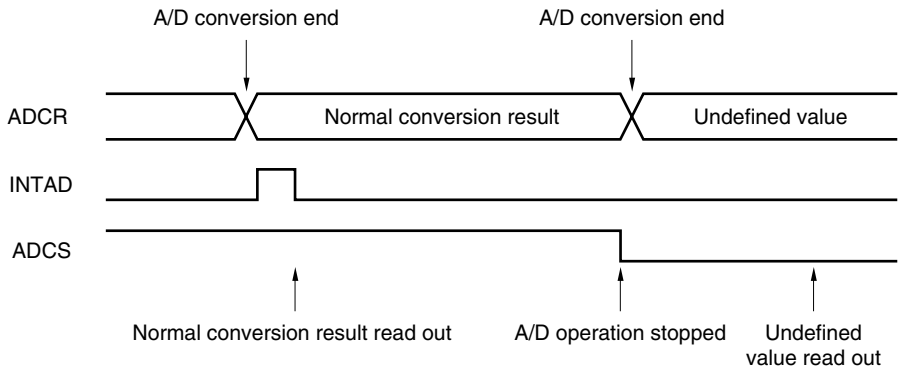
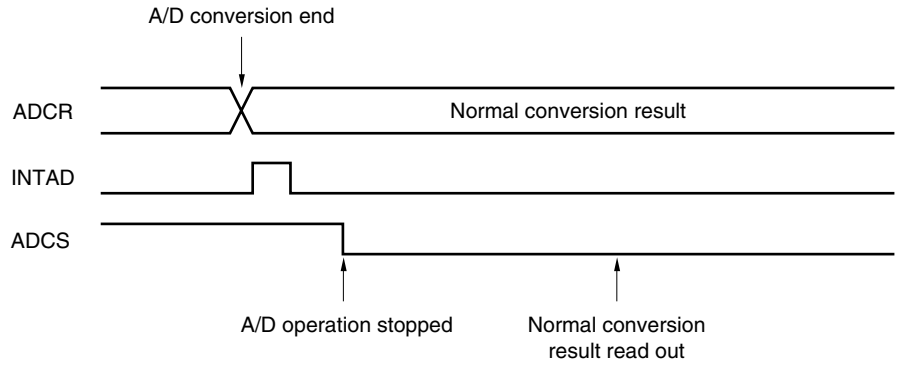


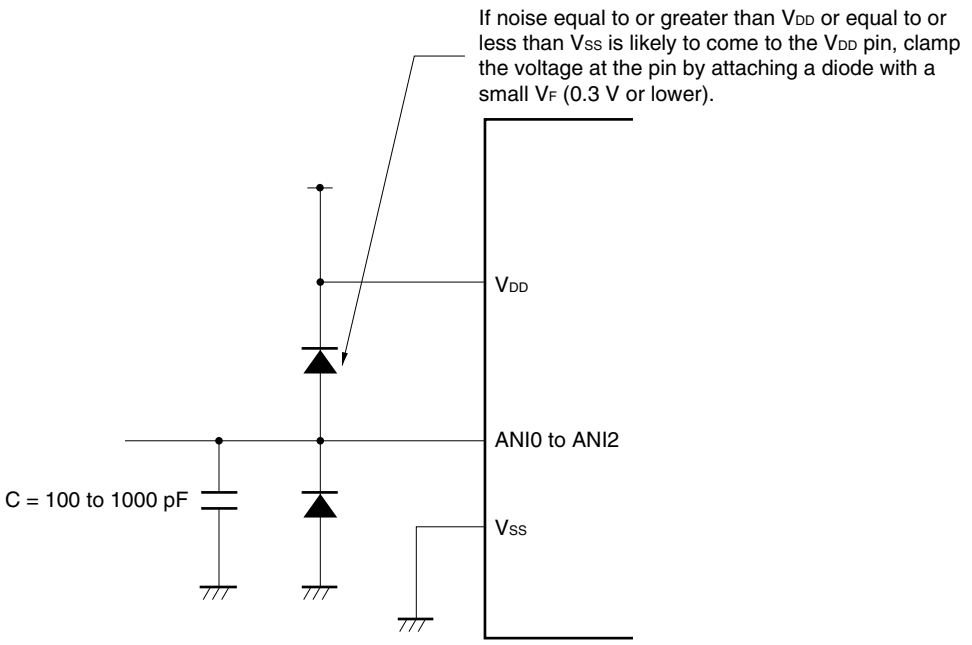
Figure 12-9. Conversion Result Readout Timing (When Conversion Result Is Normal Value)



(6) Noise prevention

To maintain a resolution of 8 bits, be careful of noise at the V_{DD} and ANI0 to ANI2 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 12-10.

Figure 12-10. Analog Input Pin Handling



(7) ANI0 to ANI2

The analog input pins (ANI0 to ANI2) are alternate-function pins. They are also used as port pins (P60 to P62).

If any of ANI0 to ANI2 has been selected for A/D conversion, do not execute input instructions for the ports; otherwise the conversion resolution may become lower.

If a digital pulse is applied to a pin adjacent to the analog input pins during A/D conversion, 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 pins during A/D conversion.

(8) Input impedance of the ANI0 to ANI2 pins

This A/D converter charges the internal sampling capacitor for about 1/10 of the conversion time, and performs sampling.

Therefore at times other than sampling, only the leak current is output. During sampling, the current for charging the capacitor is also output, so the input impedance fluctuates and has no meaning.

However, to ensure adequate sampling, it is recommended that the output impedance of the analog input source be set to below 10 kΩ, or a 100 pF capacitor be connected to the ANI0 to ANI2 pins (see **Figure 12-10**).

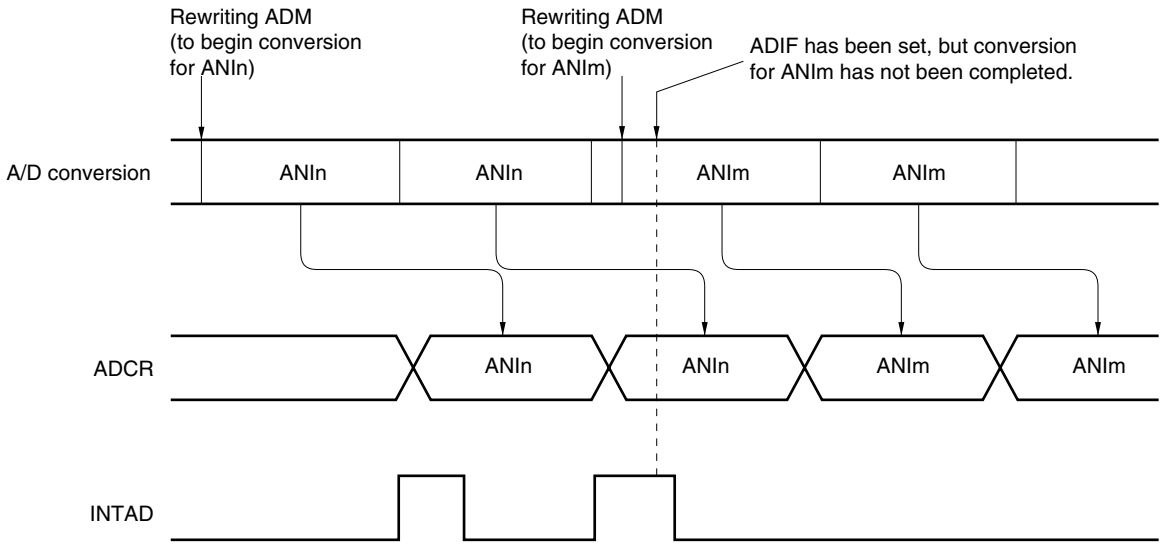
(9) Interrupt request flag (ADIF)

Changing the contents of the A/D converter mode register (ADM) does not clear the interrupt request flag (ADIF).

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

In addition, when A/D conversion is restarted, ADIF must be cleared beforehand.

Figure 12-11. A/D Conversion End Interrupt Request Generation Timing



Remarks 1. n = 0 to 2

2. m = 0 to 2

CHAPTER 13 SERIAL INTERFACE

13.1 Serial Interface Functions

The serial interface has the following three modes.

- Operation stop mode
- Asynchronous serial interface (UART) mode
- 3-wire serial I/O mode

(1) Operation stop mode

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

(2) Asynchronous serial interface (UART) mode

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

A dedicated UART baud rate generator is incorporated, allowing communication over a wide range of baud rates. In addition, the baud rate can be defined by dividing the clock input to the ASCK pin.

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

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

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

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

The 3-wire serial I/O mode is effective for connecting display controllers and peripheral I/Os such as the 75XL Series, 78K Series, and 17K Series that include a conventional clocked serial interface.

Figures 13-1 and 13-2 show the block diagrams of the serial interface.

Figure 13-1. Block Diagram of Serial Interface (SIO10)

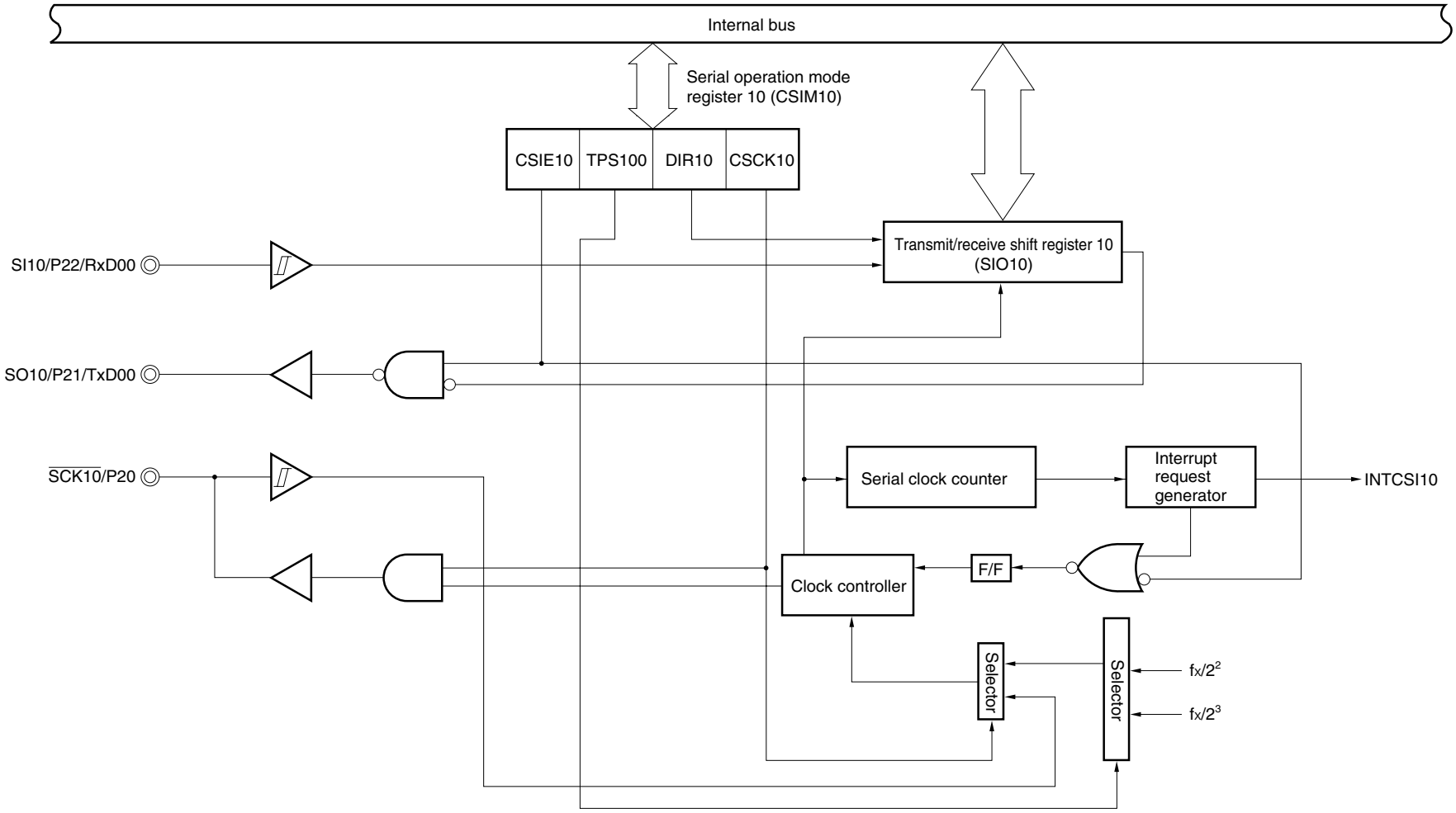
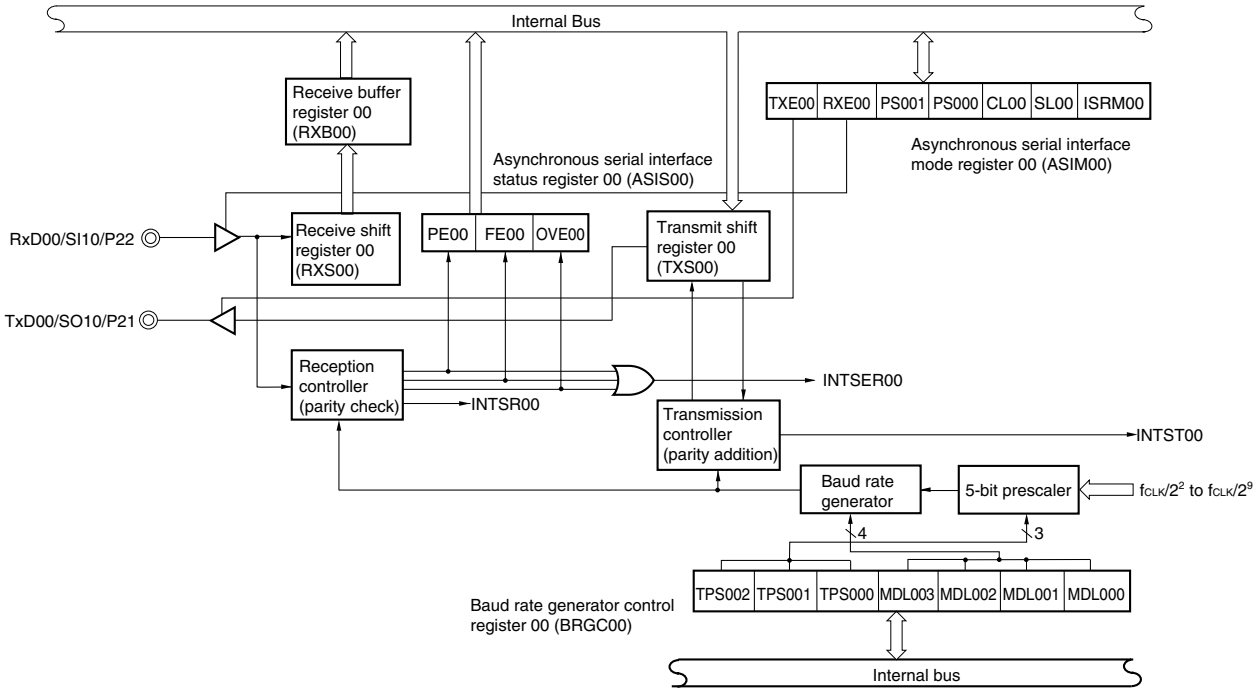


Figure 13-2. Block Diagram of Serial Interface (UART00)



13.2 Serial Interface Configuration

The serial interface has the following hardware configuration.

Table 13-1. Configuration of Serial Interface

Item	Configuration
Registers	Transmit/receive shift register 10 (SIO10) Transmit shift register 00 (TXS00) Receive shift register 00 (RXS00) Receive buffer register 00 (RXB00)
Control registers	Serial operation mode register 10 (CSIM10) Asynchronous serial interface mode register 00 (ASIM00) Asynchronous serial interface status register 00 (ASIS00) Baud rate generator control register 00 (BRGC00)

(1) Transmit/receive shift register 10 (SIO10)

This is an 8-bit register used for parallel-to-serial conversion and to perform serial data transmission/reception in synchronization with the serial clock in the 3-wire serial I/O mode.

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

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

(2) Transmit shift register 00 (TXS00)

This register is used to specify the data to be transmitted in the UART mode. Data written to TXS00 is transmitted as serial data.

If the data length is specified as 7 bits, bits 0 to 6 of the data written to TXS00 are transferred as the transmit data. The transmit operation is started by writing data to TXS00.

TXS00 is written to with an 8-bit memory manipulation instruction. It cannot be read.

$\overline{\text{RESET}}$ input sets TXS00 to FFH.

Caution During a transmit operation, do not write to TXS00.
TXS00 and receive buffer register 00 (RXB00) are allocated to the same address, and when reading is performed, RXB00 values are read.

(3) Receive shift register 00 (RXS00)

This register is used to convert serial data input to the RxD pin into parallel data in the UART mode. Each time one byte of data is received, it is transferred to receive buffer register 00 (RXB00).

RXS00 cannot be manipulated directly by program.

(4) Receive buffer register 00 (RXB00)

This register is used to hold received data in the UART mode. Each time one byte of data is received, a new byte of data is transferred from receive shift register 00 (RXS00).

If the data length is specified as 7 bits, receive data is transferred to bits 0 to 6 of RXB00, and the MSB of RXB00 always becomes 0.

RXB00 can be read with an 8-bit memory manipulation instruction. It cannot be written to.

$\overline{\text{RESET}}$ input becomes undefined.

Caution RXB00 and transmit shift register 00 (TXS00) are allocated to the same address, and when writing is performed, the values are written to TXS00.

(5) Transmit controller

This controller controls transmit operations by adding a start bit, parity bit, and stop bit to data written to transmit shift register 00 (TXS00), according to the data set to asynchronous serial interface mode register 00 (ASIM00).

(6) Receive controller

This controller controls receive operations according to the data set to asynchronous serial interface mode register 00 (ASIM00). It also performs parity error check, etc., during receive operations, and when an error is detected, it sets the value to asynchronous serial interface status register 00 (ASIS00) depending on the nature of the error.

13.3 Registers Controlling Serial Interface

The following four registers are used to control the serial interface.

- Serial operation mode register 10 (CSIM10)
- Asynchronous serial interface mode register 00 (ASIM00)
- Asynchronous serial interface status register 00 (ASIS00)
- Baud rate generator control register 00 (BRGC00)

(1) Serial operation mode register 10 (CSIM10)

This register is set when using the serial interface in the 3-wire serial I/O mode.

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

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

Figure 13-3. Format of Serial Operation Mode Register 10

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM10	CSIE10	0	0	TPS100	0	DIR10	CSCK10	0	FF72H	00H	R/W

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

TPS100	Count clock selection when internal clock is selected	
	@ f_X = 5.0 MHz operation	@ f_{CC} = 2.0 MHz operation
0	$f_X/2^2$ (1.25 MHz)	$f_{CC}/2^2$ (500 kHz)
1	$f_X/2^3$ (625 kHz)	$f_{CC}/2^3$ (250 kHz)

DIR10	First bit specification
0	MSB
1	LSB

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

Cautions 1. Bits 0, 3, 5, and to 6 must be fixed to 0.

2. Set CSIM10 to 00H in the UART mode.

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

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

(2) Asynchronous serial interface mode register 00 (ASIM00)

This register is set when using the serial interface in the asynchronous serial interface mode.

ASIM00 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets the ASIM00 to 00H.

Figure 13-4. Format of Asynchronous Serial Interface Mode Register 00

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After reset	R/W
ASIM00	TXE00	RXE00	PS001	PS000	CL00	SL00	ISRM00	0	FFA0H	00H	R/W

TXE00	Transmit operation control
0	Transmit operation stopped
1	Transmit operation enabled

RXE00	Receive operation control
0	Receive operation stopped
1	Receive operation enabled

PS001	PS000	Parity bit specification
0	0	No parity
0	1	0 parity always added at transmission Parity check is not performed at reception (no parity error is generated)
1	0	Odd parity
1	1	Even parity

CL00	Character length specification
0	7 bits
1	8 bits

SL00	Transmit data stop bit length specification
0	1 bit
1	2 bits

ISRM00	Reception completion interrupt control at error occurrence
0	A reception completion interrupt request is generated when an error occurs.
1	A reception completion interrupt request is not generated when an error occurs.

Cautions 1. Bit 0 must be fixed to 0.

2. Set ASIM00 to 00H in the 3-wire serial I/O mode.

3. Switching operation modes must be performed after the serial transmit/receive operation has been stopped.

Table 13-2. Settings of Serial Interface Operating Mode

(1) Operation stop mode

ASIM00		CSIM10			PM22	P22	PM21	P21	PM20	P20	First Bit	Shift Clock	P22/SI10/RxD00 Pin Function	P21/SO10/TxD00 Pin Function	P20/$\overline{SCK10}$ Pin Function
TXE00	RXE00	CSIE10	DIR10	CSCK10											
0	0	0	×	×	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	—	—	P22	P21	P20
Other than above											Setting prohibited				

(2) Asynchronous serial interface mode

ASIM00		CSIM10			PM22	P22	PM21	P21	PM20	P20	First Bit	Shift Clock	P22/SI10/RxD00 Pin Function	P21/SO10/TxD00 Pin Function	P20/$\overline{SCK10}$ Pin Function
TXE00	RXE00	CSIE10	DIR10	CSCK10											
1	0	0	0	0	×Note 1	×Note 1	0	1	×Note 1	×Note 1	LSB	Internal clock	P22	TxD00 (CMOS output)	P20
0	1	0	0	0	1	×	×Note 1	×Note 1	×Note 1	×Note 1			RxD00	P21	
1	1	0	0	0	1	×	0	1	×Note 1	×Note 1				TxD00 (CMOS output)	
Other than above											Setting prohibited				

(3) 3-wire serial I/O mode

ASIM00		CSIM10			PM22	P22	PM21	P21	PM20	P20	First Bit	Shift Clock	P22/SI10/RxD00 Pin Function	P21/SO10/TxD00 Pin Function	P20/$\overline{SCK10}$ Pin Function
TXE00	RXE00	CSIE10	DIR10	CSCK10											
0	0	1	0	0	1Note 2	×Note 2	0	1	1	×	MSB	External clock	SI10 Note 2	SO10 (CMOS output)	$\overline{SCK10}$ input
				1					0	1		Internal clock			$\overline{SCK10}$ output
		1	1	0					1	×	LSB	External clock			$\overline{SCK10}$ input
				1					0	1		Internal clock			$\overline{SCK10}$ output
Other than above											Setting prohibited				

Notes 1. Can be used as port function.

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

Remark ×: Don't care

(3) Asynchronous serial interface status register 00 (ASIS00)

This register is used to display the type of receive error, if it occurs while asynchronous serial interface mode is set.

ASIS00 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIS00 to 00H.

Figure 13-5. Format of Asynchronous Serial Interface Status Register 00

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ASIS00	0	0	0	0	0	PE00	FE00	OVE00	FFA3H	00H	R

PE00	Parity error flag
0	Parity error not generated
1	Parity error generated (when specified parity of transmit data does not match receive data parity)

FE00	Framing error flag
0	Framing error not generated
1	Framing error generated[Note 1] (when stop bit is not detected)

OVE00	Overrun error flag
0	Overrun error not generated
1	Overrun error generated[Note 2] (when the next receive operation is completed before the data is read from the receive buffer register)

Notes 1. Even when the stop bit length is set to 2 bits by setting bit 2 (SL00) of asynchronous serial interface mode register 00 (ASIM00), the stop bit detection in the case of reception is performed with 1 bit.

2. Be sure to read receive buffer register 00 (RXB00) when an overrun error occurs. If not, an overrun error will occur every time data is received.

Caution Be sure to set bits 3 to 7 to 0.

(4) Baud rate generator control register 00 (BRGC00)

This register is used to specify the serial clock for the serial interface.

BRGC00 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets BRGC00 to 00H.

Figure 13-6. Format of Baud Rate Generator Control Register 00

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
BRGC00	0	TPS002	TPS001	TPS000	MDL003	MDL002	MDL001	MDL000	FFA1H	00H	R/W

TPS002	TPS001	TPS000	Selection of 5-bit counter source clock (f_{SCK})		
			@ f_X = 5.0 MHz operation	@ f_{CC} = 2.0 MHz operation	n
0	0	0	$f_X/2^2$ (1.25 MHz)	$f_{CC}/2^2$ (500 kHz)	1
0	0	1	$f_X/2^3$ (625 kHz)	$f_{CC}/2^3$ (250 kHz)	2
0	1	0	$f_X/2^4$ (313 kHz)	$f_{CC}/2^4$ (125 kHz)	3
0	1	1	$f_X/2^5$ (156 kHz)	$f_{CC}/2^5$ (62.5 kHz)	4
1	0	0	$f_X/2^6$ (78.1 kHz)	$f_{CC}/2^6$ (31.3 kHz)	5
1	0	1	$f_X/2^7$ (39.1 kHz)	$f_{CC}/2^7$ (15.6 kHz)	6
1	1	0	$f_X/2^8$ (19.5 kHz)	$f_{CC}/2^8$ (7.81 kHz)	7
1	1	1	$f_X/2^9$ (9.8 kHz)	$f_{CC}/2^9$ (3.91 kHz)	8

MDL003	MDL002	MDL001	MDL000	Selection of baud rate generator input clock	k
0	0	0	0	$f_{SCK}/16$	0
0	0	0	1	$f_{SCK}/17$	1
0	0	1	0	$f_{SCK}/18$	2
0	0	1	1	$f_{SCK}/19$	3
0	1	0	0	$f_{SCK}/20$	4
0	1	0	1	$f_{SCK}/21$	5
0	1	1	0	$f_{SCK}/22$	6
0	1	1	1	$f_{SCK}/23$	7
1	0	0	0	$f_{SCK}/24$	8
1	0	0	1	$f_{SCK}/25$	9
1	0	1	0	$f_{SCK}/26$	10
1	0	1	1	$f_{SCK}/27$	11
1	1	0	0	$f_{SCK}/28$	12
1	1	0	1	$f_{SCK}/29$	13
1	1	1	0	$f_{SCK}/30$	14
1	1	1	1	Setting prohibited	—

Cautions 1. Be sure to set bit 7 to 0.

2. When writing to BRGC00 is performed during a communication operation, the output of the baud rate generator is disrupted and communications cannot be performed normally. Be sure not to write to BRGC00 during a communication operation.

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

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

3. fSCK: Source clock of 5-bit counter

4. n: Value determined by the setting of TPS000 to TPS002 ($1 \leq n \leq 8$)

5. k: Value determined by the setting of MDL000 to MDL003 ($0 \leq k \leq 14$)

The baud rate transmit/receive clock to be generated is either a divided main system clock signal, or a divided signal of the clock input from the ASCK pin.

(a) Generation of baud rate transmit/receive clock by means of main system clock

The transmit/receive clock is generated by dividing the main system clock. The baud rate generated from the main system clock is estimated by using the following expression.

$$[\text{Baud rate}] = \frac{f_{CLK}}{2^{n+2} \times (k + 16)} \quad [\text{Hz}]$$

f_{CLK}: main system clock oscillation frequency (f_X or f_{CC})

n: Value in Figure 13-7 determined by the settings of TPS000 to TPS002 ($1 \leq n \leq 8$)

k: Value in Figure 13-7 determined by the setting of MDL000 to MDL003 ($0 \leq k \leq 14$)

Table 13-3. Example of Relationship Between Main System Clock and Baud Rate (When f_X = 5.0 MHz)

Baud Rate (bps)	BRGC00 Set Value	n	k	Error (%)
				f_X = 5.0 MHz
300	70H	8	0	1.67
600	60H	7	0	
1200	50H	6	0	
2400	40H	5	0	
4800	30H	4	0	
9600	20H	3	0	
19200	10H	2	0	
31250	14H	1	4	0.00
38400	00H	1	0	1.67

Table 13-4. Example of Relationship Between Main System Clock and Baud Rate (When f_X = 4.9152 MHz)

Baud Rate (bps)	BRGC00 Set Value	n	k	Error (%)
				f_X = 4.9152 MHz
300	6FH	7	16	0.00
600	5FH	6	16	
1200	4FH	5	16	
2400	3FH	4	16	
4800	2FH	3	16	
9600	1FH	2	16	
19200	0FH	1	16	
31250	01H	1	4	1.7
38400	–	–	–	–

Table 13-5. Example of Relationship Between Main System Clock and Baud Rate (When fx = 4.1943 MHz)

Baud Rate (bps)	BRGC00 Set Value	n	k	Error (%)
				fx = 4.1943 MHz
300	6BH	7	11	1.67
600	5BH	6	11	
1200	4BH	5	11	
2400	3BH	4	11	
4800	2BH	3	11	
9600	1BH	2	11	
19200	0BH	1	11	
31250	00H	1	0	4.9
38400	–	–	–	–

Table 13-6. Example of Relationship Between Main System Clock and Baud Rate (When fx = 4.00 MHz)

Baud Rate (bps)	BRGC00 Set Value	n	k	Error (%)
				fx = 4.00 MHz
300	6AH	7	10	0.2
600	5AH	6	10	
1200	4AH	5	10	
2400	3AH	4	10	
4800	2AH	3	10	
9600	1AH	2	10	
19200	0AH	1	10	
31250	01H	1	0	0.0
38400	–	–	–	–

13.4 Serial Interface Operation

The serial interface provides the following three modes.

- Operation stop mode
- Asynchronous serial interface (UART) mode
- 3-wire serial I/O mode

13.4.1 Operation stop mode

In the operation stop mode, serial transfer is not executed, enabling a reduction in the power consumption.
The P20/$\overline{\text{SCK10}}$, P21/SO10/TxD00, and P22/SI10/RxD00 pins can be used as normal I/O ports.

(1) Register setting

Operation stop mode is set by serial operation mode register 10 (CSIM10) and asynchronous serial interface mode register 00 (ASIM00).

(a) Serial operation mode register 10 (CSIM10)

CSIM10 is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets CSIM00 to 00H.

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM10	CSIE10	0	0	TPS100	0	DIR10	CSCK10	0	FF72H	00H	R/W

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

Caution Bits 0, 3, 5, and 6 must be fixed to 0.

(b) Asynchronous serial interface mode register 00 (ASIM00)

ASIM00 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIM00 to 00H.

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After reset	R/W
ASIM00	TXE00	RXE00	PS001	PS000	CL00	SL00	ISRM00	0	FFA0H	00H	R/W

TXE00	Transmit operation control
0	Transmit operation stopped
1	Transmit operation enabled

RXE00	Receive operation control
0	Receive operation stopped
1	Receive operation enabled

Caution Bit 0 must be fixed to 0.

13.4.2 Asynchronous serial interface (UART) mode

In this mode, the one-byte data following the start bit is transmitted/received enabling full-duplex communications.

This device incorporates a UART-dedicated baud rate generator, allowing communication at a wide range of baud rates.

The UART-dedicated baud rate generator also can output the 31.25-kbps baud rate that complies with the MIDI standard.

(1) Register setting

UART mode is set by serial operation mode register 10 (CSIM10), asynchronous serial interface mode register 00 (ASIM00), asynchronous serial interface status register 00 (ASIS00), and baud rate generator control register 00 (BRGC00).

(a) Serial operation mode register 10 (CSIM10)

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

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

Set CSIM10 to 00H in the UART mode.

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM10	CSIE10	0	0	TPS100	0	DIR10	CSCK10	0	FF72H	00H	R/W

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

TPS100	Count clock selection when internal clock is selected	
	@ f_X = 5.0 MHz operation	@ f_{CC} = 2.0 MHz operation
0	$f_X/2^2$ (1.25 MHz)	$f_{CC}/2^2$ (500 kHz)
1	$f_X/2^3$ (625 kHz)	$f_{CC}/2^3$ (250 kHz)

DIR10	Start bit specification
0	MSB
1	LSB

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

Caution Bits 0, 3, 5, and 6 must be fixed to 0.

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

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

(b) Asynchronous serial interface mode register 00 (ASIM00)

ASIM00 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIM00 to 00H.

Caution When using the serial interface function (UART mode), set the related output latches to 0, and the port mode registers (PMxx) as follows.

- **For reception**
  Set P22 (RxD00) to input mode (PM22 = 1).
- **For transmission**
  Set P21 (TxD00) to output mode (PM21 = 0).
- **For transmission and reception**
  Set P22 and P21 to input and output mode, respectively.

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After reset	R/W
ASIM00	TXE00	RXE00	PS001	PS000	CL00	SL00	ISRM00	0	FFA0H	00H	R/W

TXE00	RXE00	Operation mode	Function of RxD00/SI10/P22 Pin	Function of TxD00/SO10/P21 pin
0	0	Operation disabled	Port function (P22)	Port function (P21)
0	1	UART mode (reception only)	Serial function (RxD00)	
1	0	UART mode (transmission only)	Port function (P22)	Serial function (TxD00)
1	1	UART mode (transmission and reception)	Serial function (RxD00)	

PS001	PS000	Parity bit specification
0	0	No parity
0	1	At transmission, the parity bit is fixed to 0. At reception, a parity check is not made; no parity error is reported.
1	0	Odd parity
1	1	Even parity

CL00	Character length specification
0	7 bits
1	8 bits

SL00	Transmission data stop bit length specification
0	1 bit
1	2 bits

ISRM00	Reception completion interrupt control at error occurrence
0	A reception completion interrupt request is generated when an error occurs.
1	A reception completion interrupt request is not generated when an error occurs.

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

2. Switch the operation mode after stopping serial transmission/reception.

(c) Asynchronous serial interface status register 00 (ASIS00)

ASIS00 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIS00 to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ASIS00	0	0	0	0	0	PE00	FE00	OVE00	FFA3H	00H	R

PE00	Parity error flag
0	Parity error not generated
1	Parity error generated (when specified parity of transmit data does not match receive data parity)

FE00	Framing error flag
0	Framing error not generated
1	Framing error generated[Note 1] (when stop bit is not detected)

OVE00	Overrun error flag
0	Overrun error not generated
1	Overrun error generated[Note 2] (when the next receive operation is completed before the data is read from the receive buffer register)

Notes 1. Even when the stop bit length is set to 2 bits by setting bit 2 (SL00) of asynchronous serial interface mode register 00 (ASIM00), the stop bit detection in the case of reception is performed with 1 bit.

2. Be sure to read receive buffer register 00 (RXB00) when an overrun error occurs. If not, an overrun error will occur every time data is received.

Caution Be sure to set bits 3 to 7 to 0.

(d) Baud rate generator control register 00 (BRGC00)

BRGC00 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets BRGC00 to 00H.

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
BRGC00	0	TPS002	TPS001	TPS000	MDL003	MDL002	MDL001	MDL000	FFA1H	00H	R/W

TPS002	TPS001	TPS000	Selection of 5-bit counter source clock (f_{SCK})		
			@ f_X = 5.0 MHz operation	@ f_{CC} = 2.0 MHz operation	n
0	0	0	$f_X/2^2$ (1.25 MHz)	$f_{CC}/2^2$ (500 kHz)	1
0	0	1	$f_X/2^3$ (625 kHz)	$f_{CC}/2^3$ (250 kHz)	2
0	1	0	$f_X/2^4$ (313 kHz)	$f_{CC}/2^4$ (125 kHz)	3
0	1	1	$f_X/2^5$ (156 kHz)	$f_{CC}/2^5$ (62.5 kHz)	4
1	0	0	$f_X/2^6$ (78.1 kHz)	$f_{CC}/2^6$ (31.3 kHz)	5
1	0	1	$f_X/2^7$ (39.1 kHz)	$f_{CC}/2^7$ (15.6 kHz)	6
1	1	0	$f_X/2^8$ (19.5 kHz)	$f_{CC}/2^8$ (7.81 kHz)	7
1	1	1	$f_X/2^9$ (9.8 kHz)	$f_{CC}/2^9$ (3.91 kHz)	8

MDL003	MDL002	MDL001	MDL000	Selection of baud rate generator input clock	k
0	0	0	0	$f_{SCK}/16$	0
0	0	0	1	$f_{SCK}/17$	1
0	0	1	0	$f_{SCK}/18$	2
0	0	1	1	$f_{SCK}/19$	3
0	1	0	0	$f_{SCK}/20$	4
0	1	0	1	$f_{SCK}/21$	5
0	1	1	0	$f_{SCK}/22$	6
0	1	1	1	$f_{SCK}/23$	7
1	0	0	0	$f_{SCK}/24$	8
1	0	0	1	$f_{SCK}/25$	9
1	0	1	0	$f_{SCK}/26$	10
1	0	1	1	$f_{SCK}/27$	11
1	1	0	0	$f_{SCK}/28$	12
1	1	0	1	$f_{SCK}/29$	13
1	1	1	0	$f_{SCK}/30$	14
1	1	1	1	Setting prohibited	—

Cautions 1. Be sure to set bit 7 to 0.

2. When writing to BRGC00 is performed during a communication operation, the output of the baud rate generator is disrupted and communications cannot be performed normally. Be sure not to write to BRGC00 during a communication operation.

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

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

3. f_{SCK}: Source clock of 5-bit counter

4. n: Value determined by the setting of TPS000 to TPS002 ($1 \leq n \leq 8$)

5. k: Value determined by the setting of MDL000 to MDL003 ($0 \leq k \leq 14$)

The baud rate transmit/receive clock to be generated is a divided main system clock signal.

- **Generation of baud rate transmit/receive clock by means of main system clock**
  The transmit/receive clock is generated by dividing the main system clock. The baud rate generated from the main system clock is estimated by using the following expression.

$$[\text{Baud rate}] = \frac{f_{CLK}}{2^{n+2}(k+16)} \quad [\text{Hz}]$$

f_{CLK}: fx or f_{CC}
fx: Main system clock oscillation frequency (ceramic/crystal oscillation)
f_{CC}: Main system clock oscillation frequency (RC oscillation)

Table 13-7 shows the relationship between the source clock of the 5-bit counter assigned to bits 4 to 6 (TPS000 to TPS002) of BRGC00 and value n, while Table 13-8 shows that between the input clock of the baud rate generator and value k.

Table 13-7. Relationship Between Source Clock of 5-Bit Counter and Value n

TPS002	TPS001	TPS000	5-Bit Counter Source Clock Selection	n
0	0	0	$f_{CLK}/2^2$	1
0	0	1	$f_{CLK}/2^3$	2
0	1	0	$f_{CLK}/2^4$	3
0	1	1	$f_{CLK}/2^5$	4
1	0	0	$f_{CLK}/2^6$	5
1	0	1	$f_{CLK}/2^7$	6
1	1	0	$f_{CLK}/2^8$	7
1	1	1	$f_{CLK}/2^9$	8

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

Table 13-8. Relationship Between Input Clock of Baud Rate Generator and Value k

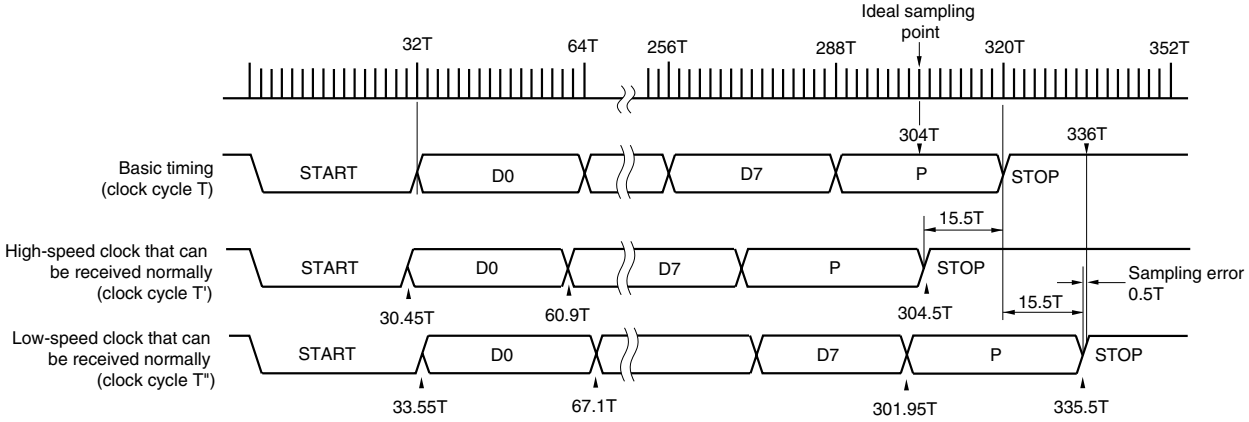
MDL003	MDL002	MDL001	MDL000	Selection of Baud Rate Generator Input Clock	k
0	0	0	0	f_{SCK}/16	0
0	0	0	1	f_{SCK}/17	1
0	0	1	0	f_{SCK}/18	2
0	0	1	1	f_{SCK}/19	3
0	1	0	0	f_{SCK}/20	4
0	1	0	1	f_{SCK}/21	5
0	1	1	0	f_{SCK}/22	6
0	1	1	1	f_{SCK}/23	7
1	0	0	0	f_{SCK}/24	8
1	0	0	1	f_{SCK}/25	9
1	0	1	0	f_{SCK}/26	10
1	0	1	1	f_{SCK}/27	11
1	1	0	0	f_{SCK}/28	12
1	1	0	1	f_{SCK}/29	13
1	1	1	0	f_{SCK}/30	14
1	1	1	1	Setting prohibited	—

Remark f_{SCK}: Source clock of 5-bit counter

- **Permissible error range of baud rate**

  The permissible error range of the baud rate is dependent upon the number of bits of one frame and the division ratio of the counter [1/(16 + k)].

Figure 13-7. Permissible Error in Baud Rate Allowing for Sampling Error (Where k = 0)



Remark T: Source clock cycle of 5-bit counter

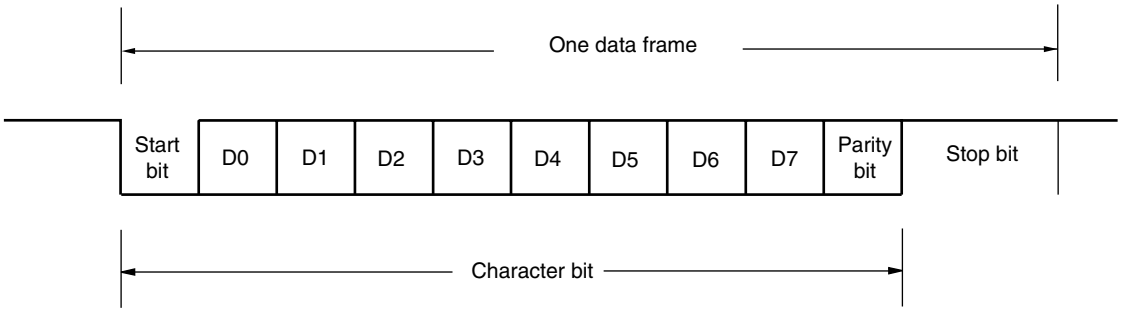
Permissible error range of baud rate (where k = 0) = $\frac{\pm 15.5}{320} \times 100 = 4.8438$ (%)

(2) Communication operation

(a) Data format

The transmission/reception data format is as shown in Figure 13-8.

Figure 13-8. Asynchronous Serial Interface Transmission/Reception Data Format



One data frame consists of the following bits.

- Start bit: 1 bit
- Character bits: 7 bits/8 bits
- Parity bits: Even parity/odd parity/0 parity/no parity
- Stop bit(s): 1 bit/2 bits

The specification of character bit length, parity selection, and specification of stop bit length for each data frame is carried out using asynchronous serial interface mode register 00 (ASIM00).

When 7 bits is selected as the number of character bits, only the lower 7 bits (bits 0 to 6) are valid; in transmission the most significant bit (bit 7) is ignored, and in reception the most significant bit (bit 7) is always “0”.

The serial transfer rate is selected by means of ASIM00 and baud rate generator control register 00 (BRGC00).

If a serial data receive error is generated, the receive error contents can be determined by reading the status of asynchronous serial interface status register 00 (ASIS00).

(b) Parity types and operation

The parity bit is used to detect a bit error in the communication data. Normally, the same parity bit is used on the transmitting side and the receiving side. With even parity and odd parity, a 1-bit (odd number) error can be detected. With 0 parity and no parity, an error cannot be detected.

(i) Even parity

- **Transmission**

  The transmission operation is controlled so that the number of bits with a value of “1” in the transmission data including parity bit is even. The parity bit value should be as follows.

  The number of bits with a value of “1” is an odd number in transmission data: 1
  The number of bits with a value of “1” is an even number in transmission data: 0

- **Reception**

  The number of bits with a value of “1” in the reception data including parity bit is counted, and if the number is odd, a parity error is generated.

(ii) Odd parity

- **Transmission**

  As opposed to even parity, the transmission operation is controlled so that the number of bits with a value of “1” in the transmission data including parity bit is odd. The parity bit value should be as follows.

  The number of bits with a value of “1” is an odd number in transmission data: 0
  The number of bits with a value of “1” is an even number in transmission data: 1

- **Reception**

  The number of bits with a value of “1” in the reception data including parity bit is counted, and if the number is even, a parity error is generated.

(iii) 0 parity

When transmitting, the parity bit is set to “0” irrespective of the transmission data.

At reception, a parity bit check is not performed. Therefore, a parity error is not generated, irrespective of whether the parity bit is set to “0” or “1”.

(iv) No parity

A parity bit is not added to the transmission data. At reception, data is received assuming that there is no parity bit. Since there is no parity bit, a parity error is not generated.

(c) Transmission

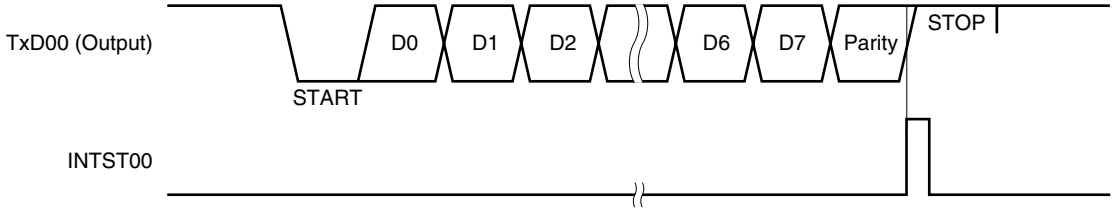
A transmit operation is enabled by setting bit 7 (TXE00) of asynchronous serial interface mode register 00 (ASIM00) to 1 and is started by writing transmit data to transmission shift register 00 (TXS00). The start bit, parity bit, and stop bit(s) are added automatically.

When the transmit operation starts, the data in TXS00 is shifted out, and when TXS00 is empty, a transmission completion interrupt request (INTST00) is generated.

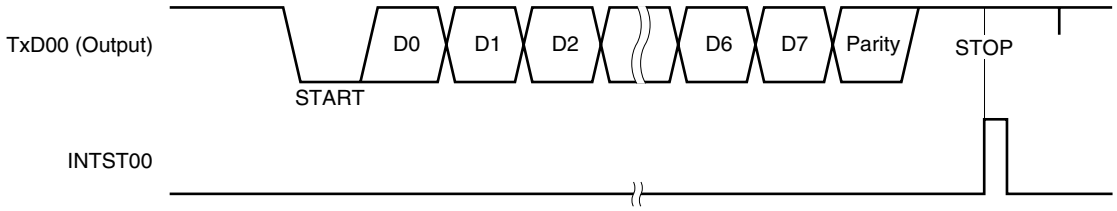
The transmission completion interrupt timing is shown in Figure 13-9.

Figure 13-9. Asynchronous Serial Interface Transmission Completion Interrupt Timing

(i) Stop bit length: 1



(ii) Stop bit length: 2



Caution Do not overwrite asynchronous serial interface mode register 00 (ASIM00) during a transmit operation. If the ASIM00 register is overwritten during transmission, subsequent transmission may not be performed (the normal state is restored by $\overline{\text{RESET}}$ input).

It is possible to determine whether transmission is in progress by software by using a transmission completion interrupt request (INTST00) or the interrupt request flag (STIF00) set by INTST00.

(d) Reception

When bit 6 (RXE00) of asynchronous serial interface mode register 00 (ASIM00) is set to 1, a receive operation is enabled and sampling of the RxD00 pin input is performed.

RxD00 pin input sampling is performed using the serial clock specified by ASIM00.

When the RxD00 pin input becomes low, the 5-bit counter of the baud rate generator starts counting, and when half the time determined by the specified baud rate has passed, the data sampling start timing signal is output. If the RxD00 pin input sampled again as a result of this start timing signal is low, it is identified as a start bit, the 5-bit counter is initialized and starts counting, and data sampling is performed. When character data, a parity bit, and one stop bit are detected after the start bit, reception of one frame of data ends.

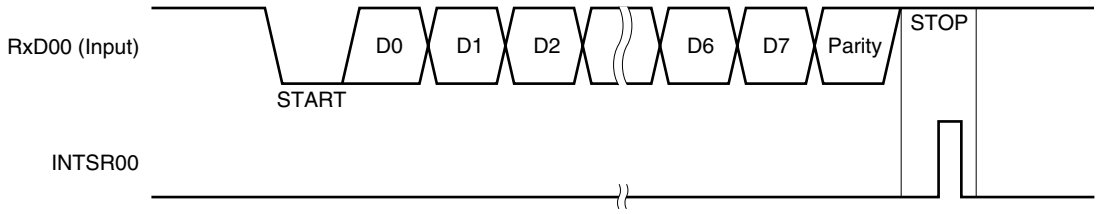
When one frame of data has been received, the receive data in the shift register is transferred to receive buffer register 00 (RXB00), and a reception completion interrupt request (INTSR00) is generated.

Even if an error occurs, the receive data in which the error occurred is still transferred to RXB00. If bit 1 (ISRM00) of ASIM00 is cleared to 0 when an error occurs, INTSR00 is generated (see **Figure 13-11**). If the ISRM00 bit is set to 1, INTSR00 is not generated.

If the RXE00 bit is cleared to 0 during the receive operation, the receive operation is stopped immediately. In this case, the contents of RXB00 and ASIS00 are not changed, and INTSR00 and INTSER00 are not generated.

Figure 13-10 shows the asynchronous serial interface reception completion interrupt timing.

Figure 13-10. Asynchronous Serial Interface Reception Completion Interrupt Timing



Caution Be sure to read reception buffer register 00 (RXB00) even if a receive error occurs. If RXB00 is not read, an overrun error will be generated when the next data is received, and the receive error state will continue indefinitely.

(e) Receive errors

The following three errors may occur during a receive operation: a parity error, framing error, or overrun error. If the error flag in asynchronous serial interface status register 00 (ASIS00) is set to 1 as a result of data reception, a reception error interrupt request (INTSER00) is generated. The reception error interrupt occurs before the reception completion interrupt request (INTSR00). Receive error causes are shown in Table 13-9.

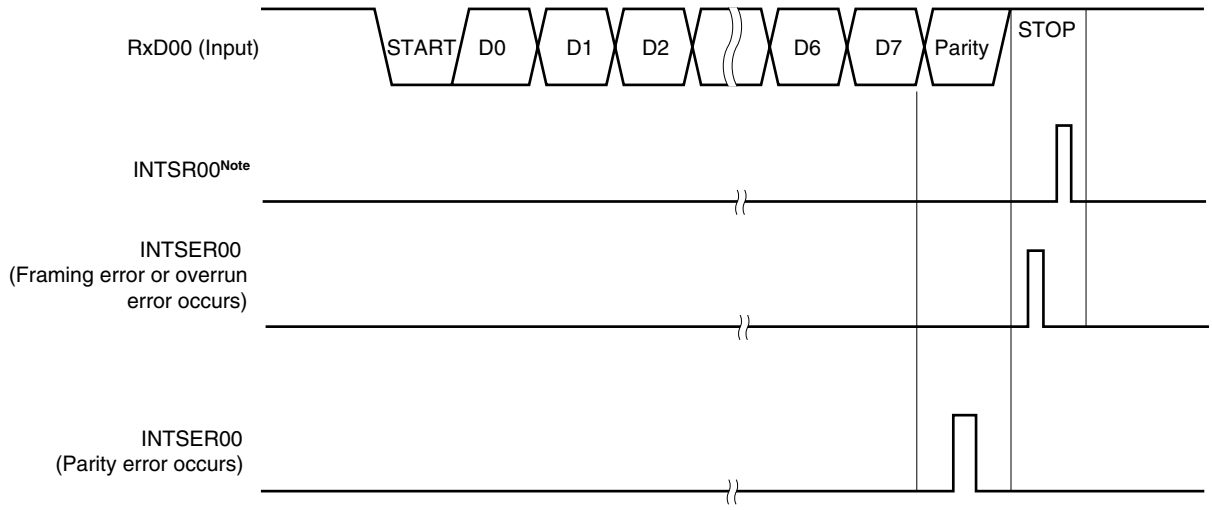
It is possible to determine what kind of error occurred during reception by reading the contents of ASIS00 in the reception error interrupt servicing (INTSER00) (see **Table 13-9** and **Figure13-11**).

The contents of ASIS00 are cleared to 0 by reading receive buffer register 00 (RXB00) or receiving the next data (if there is an error in the next data, the corresponding error flag is set).

Table 13-9. Receive Error Causes

Receive Error	Cause	ASIS00 Value
Parity error	Parity specification at transmission and reception data parity do not match	04H
Framing error	Stop bit not detected	02H
Overrun error	Reception of next data is completed before data is read from receive buffer register 00	01H

Figure 13-11. Receive Error Timing



Note If the receive error occurs when the ISRM00 bit is set to 1, INTSR00 is not generated.

Cautions 1. The contents of asynchronous serial interface status register 00 (ASIS00) are cleared to 0 by reading receive buffer register 00 (RXB00) or receiving the next data. To ascertain the error contents, read ASIS00 before reading RXB00.

2. Be sure to read receive buffer register 00 (RXB00) even if a receive error occurs. If RXB00 is not read, an overrun error will be generated when the next data is received, and the receive error state will continue indefinitely.

13.4.3 3-wire serial I/O mode

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

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

(1) Register setting

3-wire serial I/O mode settings are performed using serial operation mode register 10 (CSIM10), asynchronous serial interface mode register 00 (ASIM00), and baud rate generator control register 00 (BRGC00).

(a) Serial operation mode register 10 (CSIM10)

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

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

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM10	CSIE10	0	0	TPS100	0	DIR10	CSCK10	0	FF72H	00H	R/W

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

TPS100	Count clock selection when internal clock is selected	
	@ f_X = 5.0 MHz operation	@ f_{CC} = 2.0 MHz operation
0	$f_X/2^2$ (1.25 MHz)	$f_{CC}/2^2$ (500 kHz)
1	$f_X/2^3$ (625 kHz)	$f_{CC}/2^3$ (250 kHz)

DIR10	First bit specification
0	MSB
1	LSB

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

Caution Bits 0, 3, 5, and to 6 must be fixed to 0.

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

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

(b) Asynchronous serial interface mode register 00 (ASIM00)

ASIM00 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASIM00 to 00H.

Set ASIM00 to 00H in the 3-wire serial I/O mode.

Symbol	<7>	<6>	5	4	3	2	1	0	Address	After reset	R/W
ASIM00	TXE00	RXE00	PS001	PS000	CL00	SL00	ISRM00	0	FFA0H	00H	R/W

TXE00	RXE00	Operation mode	Function of RxD00/P22 pin	Function of TxD00/P21 pin
0	0	Operation stopped	Port function (P22)	Port function (P21)
0	1	UART mode (reception only)	Serial function (RxD00)	
1	0	UART mode (transmission only)	Port function (P22)	Serial function (TxD00)
1	1	UART mode (transmission and reception)	Serial function (RxD00)	

PS001	PS000	Parity bit specification
0	0	No parity
0	1	At transmission, the parity bit is fixed to 0. At reception, a parity check is not made; no parity error is reported.
1	0	Odd parity
1	1	Even parity

CL00	Character length specification
0	7 bits
1	8 bits

SL00	Transmission data stop bit length specification
0	1 bit
1	2 bits

ISRM00	Reception completion interrupt control at error occurrence
0	A reception completion interrupt request is generated when an error occurs.
1	A reception completion interrupt request is not generated when an error occurs.

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

2. Switch the operation mode after stopping serial transmission/reception.

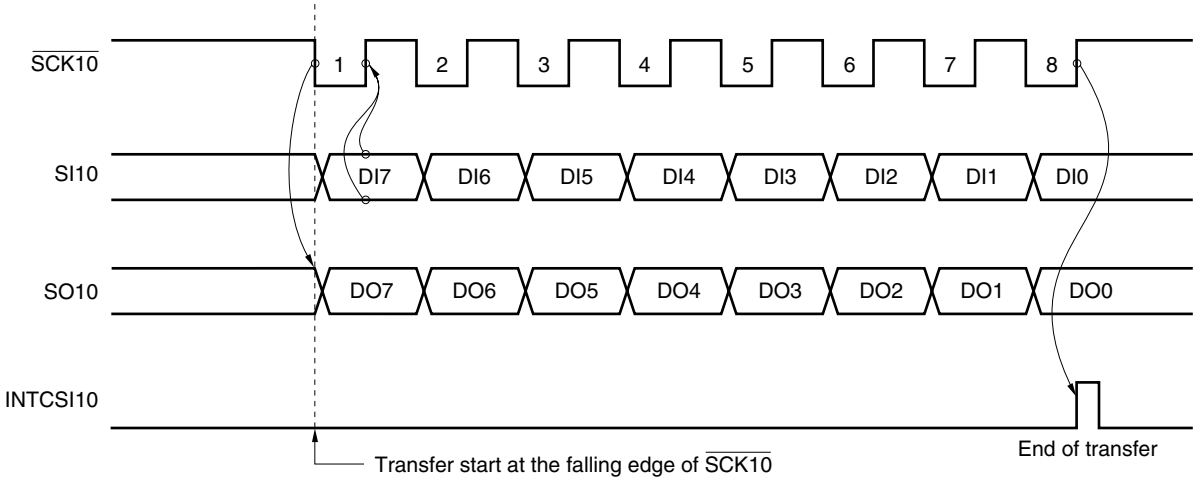
(2) Communication operation

In the 3-wire serial I/O mode, data transmission/reception is performed in 8-bit units. Data is transmitted/received in 1-bit units in synchronization with the serial clock.

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

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

Figure 13-12. 3-Wire Serial I/O Mode Timing



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

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

3. Once data has been written to SIO10 with the serial clock selected (CSCK10 = 0), overwriting the data does not update the contents of SIO10.

4. When CSIM10 is operated during data transmission/reception, data cannot be transmitted or received normally.

5. When SIO10 is operated during data transmission/reception, the data cannot be transmitted or received normally.

(3) Transfer start

Serial transfer is started by setting transfer data to the transmit/receive shift register 10 (SIO10) when the following two conditions are satisfied.

- Bit 7 (CSIE10) of serial operation mode register 10 (CSIM10) = 1
- Internal serial clock is stopped or $\overline{SCK10}$ is a high level after 8-bit serial transfer.

An end of 8-bit transfer stops the serial transfer automatically and generates the interrupt request signal (INTCSI10).

CHAPTER 14 LCD CONTROLLER/DRIVER

14.1 Functions of LCD Controller/Driver

The LCD controller/driver has the following functions.

(1) Enables automatic output of segment signals and common signals by automatically reading from display data memory.
(2) Four types of display modes can be selected:
- 1/8 duty (1/5 bias) (80 segments × 8 commons)
- 1/16 duty (1/5 bias) (80 segments × 16 commons)
- 1/32 duty (1/5 bias) (64 segments × 32 commons)
- 1/48 duty (1/5 bias) (48 segments × 48 commons)

(3) Any of four frame frequency settings can be selected for each display mode.
(4) Operation using the subsystem clock is also supported.

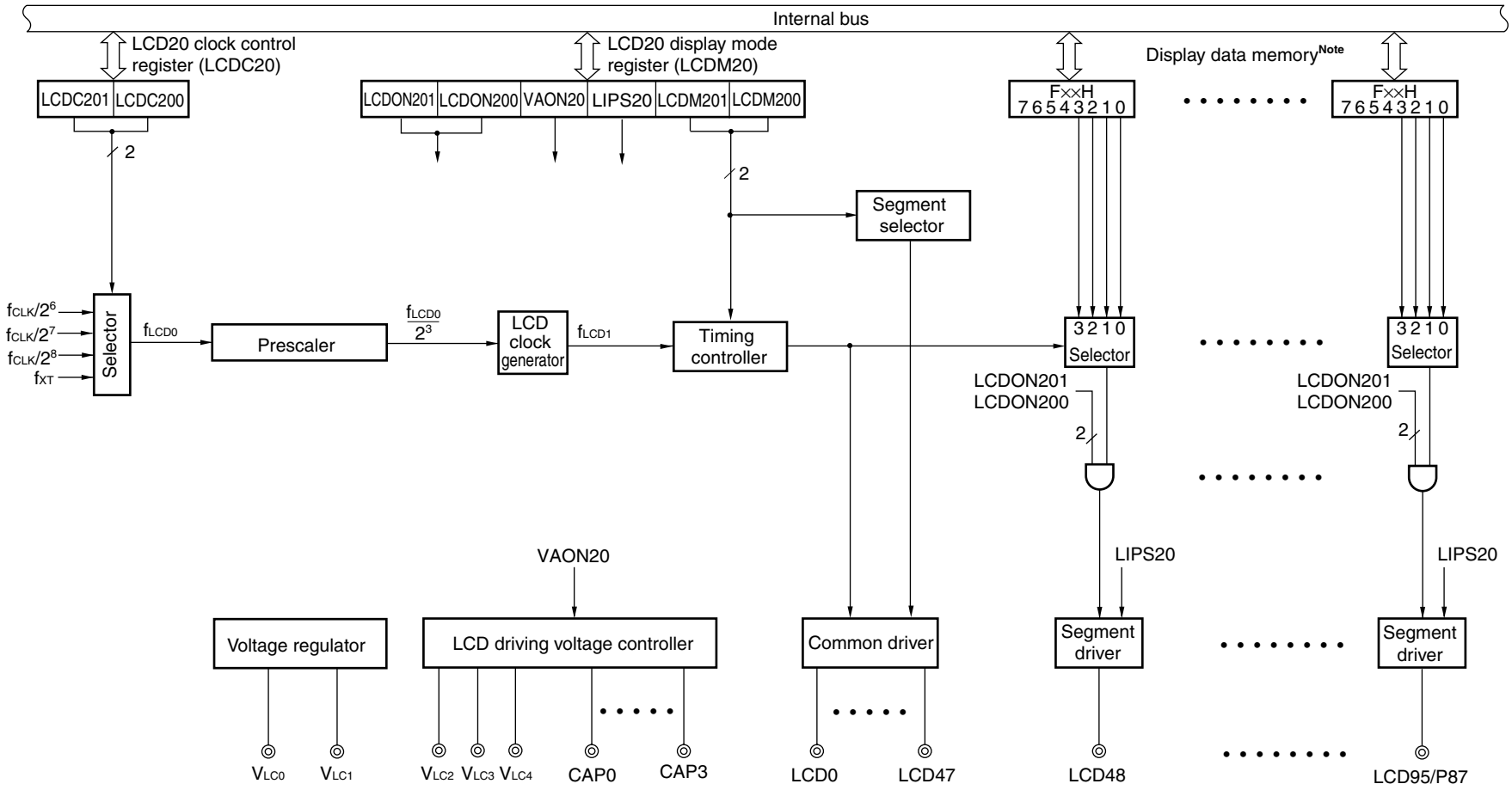
14.2 Configuration of LCD Controller/Driver

The LCD controller/driver includes the following hardware.

Table 14-1. Configuration of LCD Controller/Driver

Item	Configuration
Display outputs	Segment/common signals: 96 (LCD0 to LCD95)
Control registers	LCD20 display mode register (LCDM20) LCD20 clock control register (LCDC20) LCD boost voltage level setting register (VLCD00)

Figure 14-1. Block Diagram of LCD Controller/Driver



Note The display area changes depending on the LCD display mode.
For details, refer to **Figure 14-2 Format of LCD20 Display Mode Register**.

14.3 Registers Controlling LCD Controller/Driver

The following three registers are used to control the LCD controller/driver.

- LCD20 display mode register (LCDM20)
- LCD20 clock control register (LCDC20)
- LCD boost voltage level setting register 00 (VLCD00)

(1) LCD20 display mode register (LCDM20)

This register is used to set the display operation enabled/disabled status, the operation mode, the LCD drive power supply, and the display mode.

LCDM20 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets LCDM20 to 00H.

Figure 14-2. Format of LCD20 Display Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
LCDM20	LCDON201	LCDON200	VAON20	LIPS20	0	0	LCDM201	LCDM200	FFB0H	00H	R/W

LCDON201	LCDON200	LCD display enable/disable
0	0	Display is OFF (all segment outputs are non-select signals)
0	1	Display is ON (pattern A)Note 1
1	0	Setting prohibited
1	1	Display is ON (pattern B)Note 1

VAON20	Operation mode of LCD controller/driverNote 2
0	No internal voltage boost (normal operation)
1	Internal voltage boost is enabled (low-voltage operation)

LIPS20	Segment/common signal output enable/disable
0	Segment/common signal output disable
1	Segment/common signal output enable

LCDM201	LCDM200	Selection of LCD controller/driver display mode	
0	0	48 × 48 mode (1/48 duty)	1/5 bias
0	1	64 × 32 mode (1/32 duty)	
1	0	80 × 16 mode (1/16 duty)	
1	1	80 × 8 mode (1/8 duty)Note 3	

Notes 1. In pattern A, the RAM area is displayed between F8C0H and F9DFH, and in pattern B, between F9E0H and FAFFH. For the memory map, refer to **CHAPTER 3 CPU ARCHITECTURE**.

2. When an LCD display is not needed, set VAON20 and LIPS20 to 0 to reduce power consumption.

3. P80/LCD88 to P87/LCD95 are used as general-purpose input ports.

Cautions 1. Be sure to set bits 2 and 3 to 0.

2. Before manipulating VAON20, be sure to clear LIPS20, LCDON201, and LCDON200 to 0, and then set the LCD display to OFF.

(2) LCD20 clock control register (LCDC20)

This register is used to set the LCD clock and frame frequency.

LCDC20 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets LCDC20 to 00H.

Figure 14-3. Format of LCD20 Clock Control Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
LCDC20	0	0	0	0	0	0	LCDC201	LCDC200	FFB2H	00H	R/W

LCDC201	LCDC200	Selection of LCD source clock (f_{LCD0})	
		@ f_X = 5.0 MHz and f_{XT} = 32.768 kHz operation	@ f_{CC} = 2.0 MHz and f_{XT} = 32.768 kHz operation
0	0	$f_X/2^6$ (78.1 kHz)	$f_{CC}/2^6$ (31.3 kHz)
0	1	$f_X/2^7$ (39.1 kHz)	$f_{CC}/2^7$ (15.6 kHz)
1	0	$f_X/2^8$ (19.5 kHz)	$f_{CC}/2^8$ (7.81 kHz)
1	1	f_{XT} (32.768 kHz)	

Caution Be sure to set bits 2 to 7 to 0.

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

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

3. f_{XT}: Subsystem clock oscillation frequency

The LCD clock frequency (f_{LCD1}) is the LCD source clock divided by 2^3.

Table 14-2 lists the frame frequency values when f_{XT} (32.768 kHz) is selected as the LCD source clock (f_{LCD0}).

Table 14-2. Frame Frequency (Hz)

LCDC201	LCDC200	LCDM201	LCDM200	LCD Clock Frequency (f_{LCD1})	Frame Frequency
1	1	0	0	4096 Hz	43.7 Hz
		0	1		64 Hz
		1	0		128 Hz
		1	1		256 Hz

(3) LCD boost voltage level setting register 00 (VLCD00)

This register is used to set the voltage level of the LCD controller/driver booster.

VLCD00 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets VLCD00 to 00H.

Figure 14-4. Format of LCD Boost Voltage Level Setting Register 00

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
VLCD00	0	0	0	0	VLCD3	VLCD2	VLCD1	VLCD0	FFB3H	00H	R/W

VLCD3	VLCD2	VLCD1	VLCD0	Voltage level selection for LCD display	V_{LC0} voltage (TYP.) (V)
0	0	0	0	Voltage level 0 (light)	4.0
0	0	0	1	Voltage level 1	4.1
0	0	1	0	Voltage level 2	4.2
0	0	1	1	Voltage level 3	4.3
0	1	0	0	Voltage level 4	4.4
0	1	0	1	Voltage level 5	4.5
0	1	1	0	Voltage level 6	4.6
0	1	1	1	Voltage level 7 (medium)	4.7
1	0	0	0	Voltage level 8	4.8
1	0	0	1	Voltage level 9	4.9
1	0	1	0	Voltage level 10	5.0
1	0	1	1	Voltage level 11	5.1
1	1	0	0	Voltage level 12	5.2
1	1	0	1	Voltage level 13	5.3
1	1	1	0	Voltage level 14	5.4
1	1	1	1	Voltage level 15 (dark)	5.5

14.4 Common and Segment Signals

The LCD0 to LCD95 pins serve as common signal and segment signals. The assignment of the LCD pins differs depending on the display mode.

Table 14-3 shows the assignment of the LCD0 to LCD95 pins in each display mode.

Table 14-3. Assignment of LCD0 to LCD95 Pins in Each Display Mode (1/3)

Pin No.	Pin Name	LCD Display Mode			
		48×48 Mode (LCDM200, 201 = 00B)	64×32 Mode (LCDM200, 201 = 10B)	80×16 Mode (LCDM200, 201 = 01B)	80×8Mode (LCDM200, 201 = 11B)
1	LCD71	COM47	S39	S55	S63
2	LCD70	COM46	S38	S54	S62
3	LCD69	COM45	S37	S53	S61
4	LCD68	COM44	S36	S52	S60
5	LCD67	COM43	S35	S51	S59
6	LCD66	COM42	S34	S50	S58
7	LCD65	COM41	S33	S49	S57
8	LCD64	COM40	S32	S48	S56
9	LCD63	COM39	COM31	S47	S55
10	LCD62	COM38	COM30	S46	S54
11	LCD61	COM37	COM29	S45	S53
12	LCD60	COM36	COM28	S44	S52
13	LCD59	COM35	COM27	S43	S51
14	LCD58	COM34	COM26	S42	S50
15	LCD57	COM33	COM25	S41	S49
16	LCD56	COM32	COM24	S40	S48
17	LCD55	COM31	COM23	COM15	S47
18	LCD54	COM30	COM22	COM14	S46
19	LCD53	COM29	COM21	COM13	S45
20	LCD52	COM28	COM20	COM12	S44
21	LCD51	COM27	COM19	COM11	S43
22	LCD50	COM26	COM18	COM10	S42
23	LCD49	COM25	COM17	COM9	S41
24	LCD48	COM24	COM16	COM8	S40
25	LCD47	COM23	COM15	COM7	COM7
26	LCD46	COM22	COM14	COM6	COM6
27	LCD45	COM21	COM13	COM5	COM5
28	LCD44	COM20	COM12	COM4	COM4
29	LCD43	COM19	COM11	COM3	COM3
30	LCD42	COM18	COM10	COM2	COM2
31	LCD41	COM17	COM9	COM1	COM1
32	LCD40	COM16	COM8	COM0	COM0
33	LCD39	COM15	COM7	S39	S39
34	LCD38	COM14	COM6	S38	S38
35	LCD37	COM13	COM5	S37	S37

Table 14-3. Assignment of LCD0 to LCD95 Pins in Each Display Mode (2/3)

Pin No.	Pin Name	LCD Display Mode			
		48×48 Mode (LCDM200, 201 = 00B)	64×32 Mode (LCDM200, 201 = 10B)	80×16 Mode (LCDM200, 201 = 01B)	80×8 Mode (LCDM200, 201 = 11B)
36	LCD36	COM12	COM4	S36	S36
37	LCD35	COM11	COM3	S35	S35
38	LCD34	COM10	COM2	S34	S34
39	LCD33	COM9	COM1	S33	S33
40	LCD32	COM8	COM0	S32	S32
41	LCD31	COM7	S31	S31	S31
42	LCD30	COM6	S30	S30	S30
43	LCD29	COM5	S29	S29	S29
44	LCD28	COM4	S28	S28	S28
45	LCD27	COM3	S27	S27	S27
46	LCD26	COM2	S26	S26	S26
47	LCD25	COM1	S25	S25	S25
48	LCD24	COM0	S24	S24	S24
49	LCD23	S23	S23	S23	S23
50	LCD22	S22	S22	S22	S22
51	LCD21	S21	S21	S21	S21
52	LCD20	S20	S20	S20	S20
53	LCD19	S19	S19	S19	S19
54	LCD18	S18	S18	S18	S18
55	LCD17	S17	S17	S17	S17
56	LCD16	S16	S16	S16	S16
57	LCD15	S15	S15	S15	S15
58	LCD14	S14	S14	S14	S14
59	LCD13	S13	S13	S13	S13
60	LCD12	S12	S12	S12	S12
61	LCD11	S11	S11	S11	S11
62	LCD10	S10	S10	S10	S10
63	LCD9	S9	S9	S9	S9
64	LCD8	S8	S8	S8	S8
65	LCD7	S7	S7	S7	S7
66	LCD6	S6	S6	S6	S6
67	LCD5	S5	S5	S5	S5
68	LCD4	S4	S4	S4	S4
69	LCD3	S3	S3	S3	S3
70	LCD2	S2	S2	S2	S2
71	LCD1	S1	S1	S1	S1
72	LCD0	S0	S0	S0	S0
121	LCD95	S47	S63	S79	P87
122	LCD94	S46	S62	S78	P86
123	LCD93	S45	S61	S77	P85
124	LCD92	S44	S60	S76	P84
125	LCD91	S43	S59	S75	P83

Table 14-3. Assignment of LCD0 to LCD95 Pins in Each Display Mode (3/3)

Pin No.	Pin Name	LCD Display Mode			
		48×48 Mode (LCDM200, 201 = 00B)	64×32 Mode (LCDM200, 201 = 10B)	80×16 Mode (LCDM200, 201 = 01B)	80×8 Mode (LCDM200, 201 = 11B)
126	LCD90	S42	S58	S74	P82
127	LCD89	S41	S57	S73	P81
128	LCD88	S40	S56	S72	P80
129	LCD87	S39	S55	S71	S79
130	LCD86	S38	S54	S70	S78
131	LCD85	S37	S53	S69	S77
132	LCD84	S36	S52	S68	S76
133	LCD83	S35	S51	S67	S75
134	LCD82	S34	S50	S66	S74
135	LCD81	S33	S49	S65	S73
136	LCD80	S32	S48	S64	S72
137	LCD79	S31	S47	S63	S71
138	LCD78	S30	S46	S62	S70
139	LCD77	S29	S45	S61	S69
140	LCD76	S28	S44	S60	S68
141	LCD75	S27	S43	S59	S67
142	LCD74	S26	S42	S58	S66
143	LCD73	S25	S41	S57	S65
144	LCD72	S24	S40	S56	S64

14.5 Setting LCD Controller/Driver

Set the LCD display using the following procedure.

(1) Set the initial data to the LCD display data memory (F8C0H to F9DFH, F9E0H to FAFFH).
(2) Set the LCD clock using the LCD20 clock control register (LCDC20).
(3) Set the LCD display mode and operation mode using the LCD20 display mode register (LCDM20).
(4) Enable segment/common signal output using the LCD20 display mode register (LCDM20), and set display ON.

14.6 LCD Display Data Memory

The LCD display data memory is mapped at F8C0H to F9DFH (pattern A) and F9E0H to FAFFH (pattern B). When the LCD display data memory is not used for LCD display, it can be used as a normal RAM.

Figure 14-5 shows the assignment of the LCD display data memory in each display mode, and Table 14-4 shows the relation ship between the LCD display data memory and segment/common output.

Figure 14-5. Assignment of LCD Display Data Memory in Each Display Mode

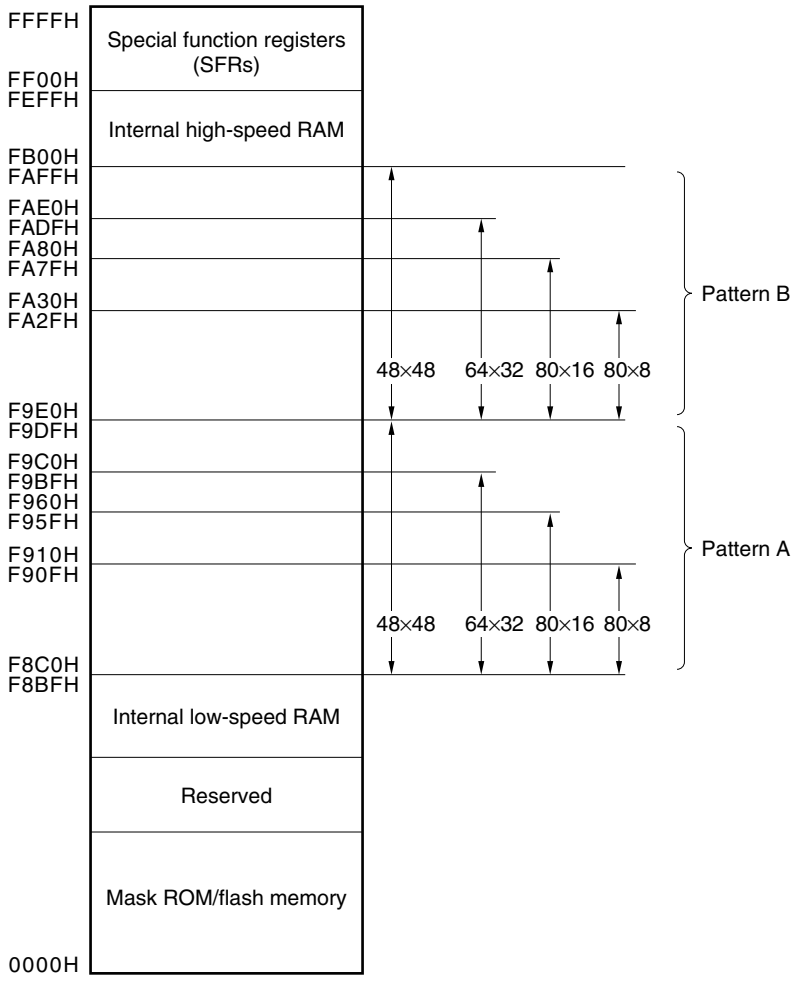


Table 14-4. Relationship Between LCD Display Data Memory and Segment/Common Output (1/7)

(i) Display mode: 48 × 48 mode, address: Pattern A (F8C0H to F9DFH)

Bit	Segment Pin						Common Pin
	S0	S1	S2	S3	• • •	S47	
b0							COM0
b1							COM1
b2							COM2
b3							COM3
b4							COM4
b5							COM5
b6							COM6
b7							COM7
Address	F8C0H	F8C6H	F8CDH	F8D3H	• • •	F9DAH	

	S0	S1	S2	S3	• • •	S47	
b0							COM8
b1							COM9
b2							COM10
b3							COM11
b4							COM12
b5							COM13
b6							COM14
b7							COM15
Address	F8C1H	F8C7H	F8CEH	F8D4H	• • •	F9DBH	

⋮

	S0	S1	S2	S3	• • •	S47	
b0							COM40
b1							COM41
b2							COM42
b3							COM43
b4							COM44
b5							COM45
b6							COM46
b7							COM47
Address	F8C5H	F8CCH	F8D2H	F8D8H	• • •	F9DFH	

Table 14-4. Relationship Between LCD Display Data Memory and Segment/Common Output (2/7)

(ii) Display mode: 48 × 48 mode, address: Pattern B (F9E0H to FAFFH)

Bit	Segment Pin						Common Pin
	S0	S1	S2	S3	• • •	S47	
b0							COM0
b1							COM1
b2							COM2
b3							COM3
b4							COM4
b5							COM5
b6							COM6
b7							COM7
Address	F9E0H	F9E6H	F9EDH	F9F3H	• • •	FAFAH	

	S0	S1	S2	S3	• • •	S47	
b0							COM8
b1							COM9
b2							COM10
b3							COM11
b4							COM12
b5							COM13
b6							COM14
b7							COM15
Address	F9E1H	F9E7H	F9EEH	F9F4H	• • •	FAFBH	

⋮

	S0	S1	S2	S3	• • •	S47	
b0							COM40
b1							COM41
b2							COM42
b3							COM43
b4							COM44
b5							COM45
b6							COM46
b7							COM47
Address	F9E5H	F9ECH	F9F2H	F9F8H	• • •	FAFFH	

Table 14-4. Relationship Between LCD Display Data Memory and Segment/Common Output (3/7)

(iii) Display mode: 64 × 32 mode, address: Pattern A (F8C0H to F9BFH)

Bit	Segment Pin						Common Pin
	S0	S1	S2	S3	• • •	S63	
b0							COM0
b1							COM1
b2							COM2
b3							COM3
b4							COM4
b5							COM5
b6							COM6
b7							COM7
Address	F8C0H	F8C4H	F8C8H	F8CCH	• • •	F9BCH	

	S0	S1	S2	S3	• • •	S63	
b0							COM8
b1							COM9
b2							COM10
b3							COM11
b4							COM12
b5							COM13
b6							COM14
b7							COM15
Address	F8C1H	F8C5H	F8C9H	F8CDH	• • •	F9BDH	

⋮

	S0	S1	S2	S3	• • •	S63	
b0							COM24
b1							COM25
b2							COM26
b3							COM27
b4							COM28
b5							COM29
b6							COM30
b7							COM31
Address	F8C3H	F8C7H	F8CBH	F8CFH	• • •	F9BFH	

Table 14-4. Relationship Between LCD Display Data Memory and Segment/Common Output (4/7)

(iv) Display mode: 64 × 32 mode, address: Pattern B (F9E0H to FADFH)

Bit	Segment Pin						Common Pin
	S0	S1	S2	S3	• • •	S63	
b0							COM0
b1							COM1
b2							COM2
b3							COM3
b4							COM4
b5							COM5
b6							COM6
b7							COM7
Address	F9E0H	F9E4H	F9E8H	F9ECH	• • •	FADCH	

	S0	S1	S2	S3	• • •	S63	
b0							COM8
b1							COM9
b2							COM10
b3							COM11
b4							COM12
b5							COM13
b6							COM14
b7							COM15
Address	F9E1H	F9E5H	F9E9H	F9EDH	• • •	FADDH	

⋮

	S0	S1	S2	S3	• • •	S63	
b0							COM24
b1							COM25
b2							COM26
b3							COM27
b4							COM28
b5							COM29
b6							COM30
b7							COM31
Address	F9E3H	F9E7H	F9EBH	F9EFH	• • •	FADFH	

Table 14-4. Relationship Between LCD Display Data Memory and Segment/Common Output (5/7)

(v) Display mode: 80 × 16 mode, address: Pattern A (F8C0H to F95FH)

Bit	Segment Pin						Common Pin
	S0	S1	S2	S3	• • •	S79	
b0							COM0
b1							COM1
b2							COM2
b3							COM3
b4							COM4
b5							COM5
b6							COM6
b7							COM7
Address	F8C0H	F8C2H	F8C4H	F8C6H	• • •	F95EH	

	S0	S1	S2	S3	• • •	S79	
b0							COM8
b1							COM9
b2							COM10
b3							COM11
b4							COM12
b5							COM13
b6							COM14
b7							COM15
Address	F8C1H	F8C3H	F8C5H	F8C7H	• • •	F95FH	

Table 14-4. Relationship Between LCD Display Data Memory and Segment/Common Output (6/7)

(vi) Display mode: 80 × 16 mode, address: Pattern B (F9E0H to FA7FH)

Bit	Segment Pin						Common Pin
	S0	S1	S2	S3	• • •	S79	
b0							COM0
b1							COM1
b2							COM2
b3							COM3
b4							COM4
b5							COM5
b6							COM6
b7							COM7
Address	F9E0H	F9E2H	F9E4H	F9E6H	• • •	FA7EH	

	S0	S1	S2	S3	• • •	S79	
b0							COM8
b1							COM9
b2							COM10
b3							COM11
b4							COM12
b5							COM13
b6							COM14
b7							COM15
Address	F9E1H	F9E3H	F9E5H	F9E7H	• • •	FA7FH	

Table 14-4. Relationship Between LCD Display Data Memory and Segment/Common Output (7/7)

(vii) Display mode: 80 × 8 mode, address: Pattern A (F8C0H to F90FH)

Bit	Segment Pin						Common Pin
	S0	S1	S2	S3	• • •	S79	
b0							COM0
b1							COM1
b2							COM2
b3							COM3
b4							COM4
b5							COM5
b6							COM6
b7							COM7
Address	F8C0H	F8C1H	F8C2H	F8C3H	• • •	F90FH	

(viii) Display mode: 80 × 8 mode, address: Pattern B (F9E0H to FA2FH)

Bit	Segment Pin						Common Pin
	S0	S1	S2	S3	• • •	S79	
b0							COM0
b1							COM1
b2							COM2
b3							COM3
b4							COM4
b5							COM5
b6							COM6
b7							COM7
Address	F9E0H	F9E1H	F9E2H	F9E3H	• • •	FA2FH	

14.7 Display Modes

14.7.1 80 × 8 mode (1/8 duty)

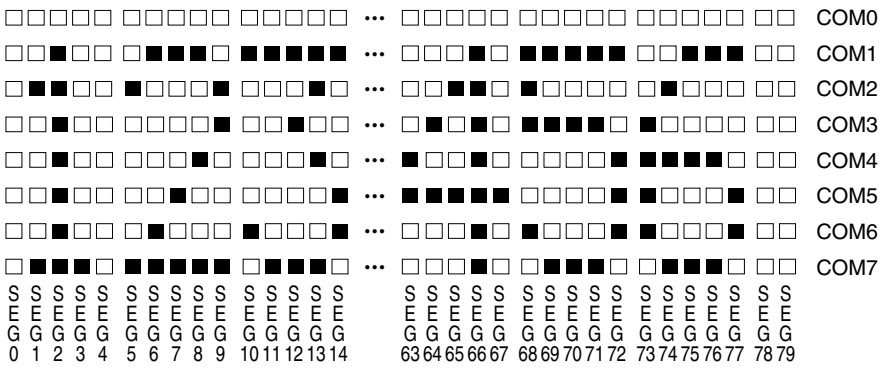
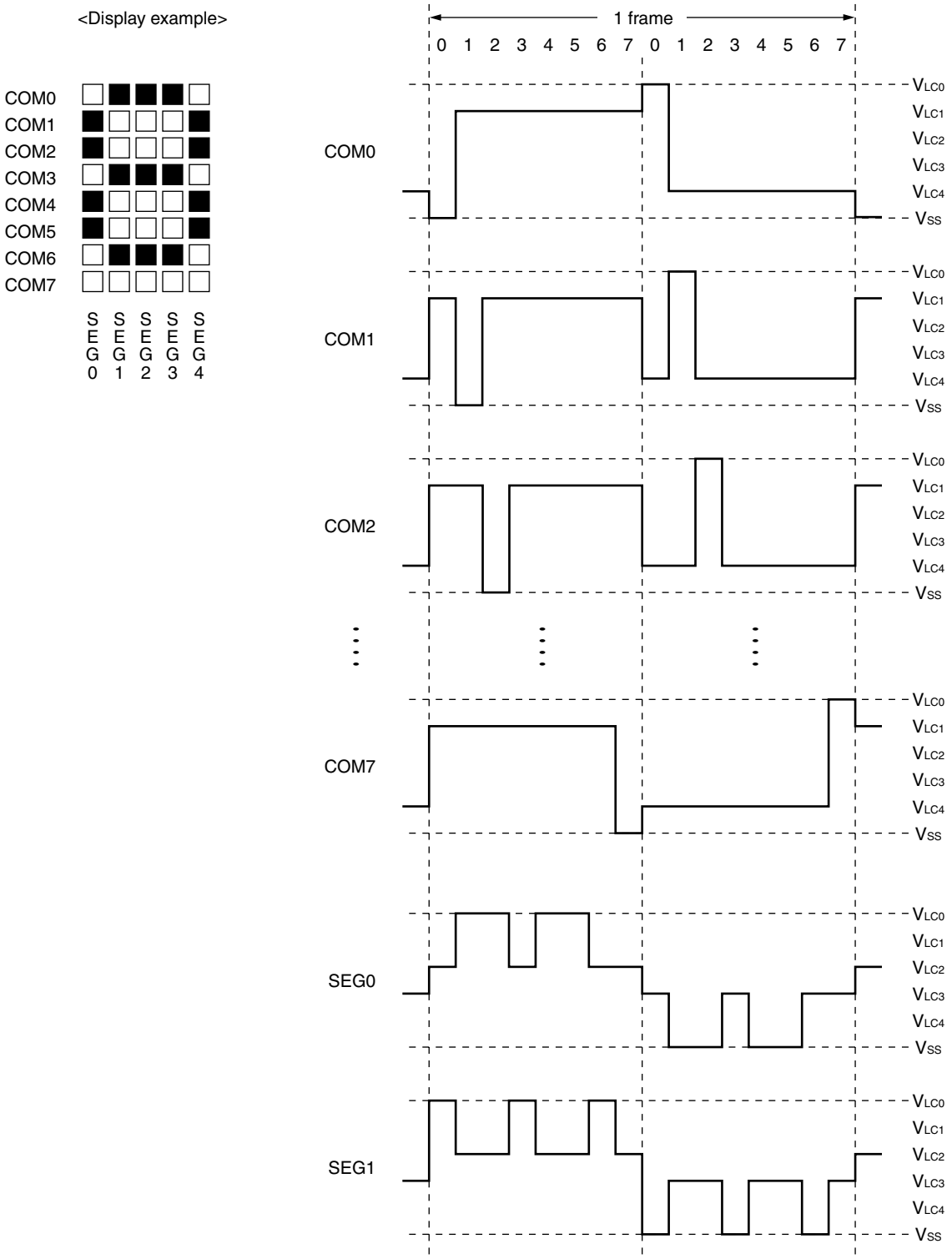


Figure 14-6. LCD Drive Waveform Examples (1/8 Duty)



14.7.2 80 × 16 mode (1/16 duty)

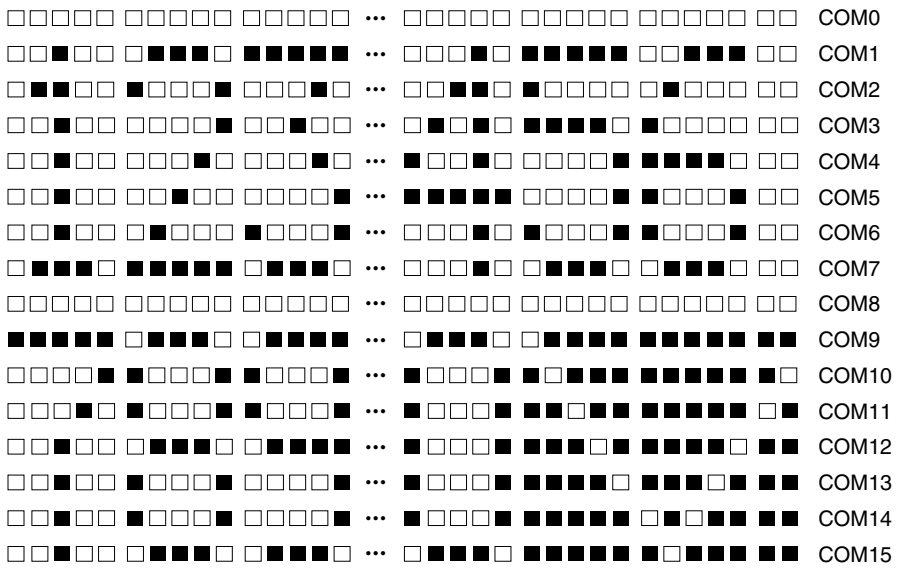
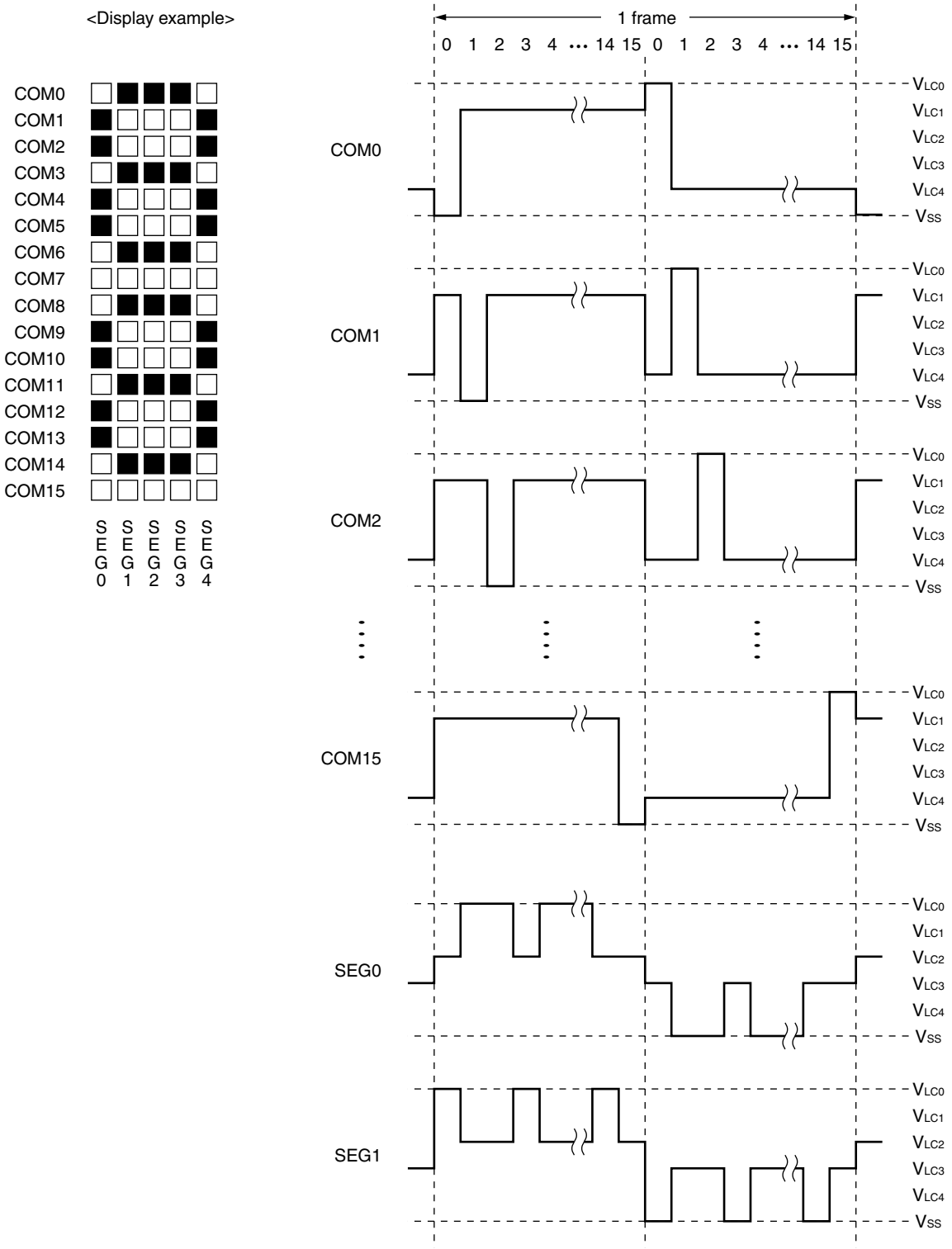


Figure 14-7. LCD Drive Waveform Examples (1/16 Duty)



14.7.3 64 × 32 mode (1/32 duty)

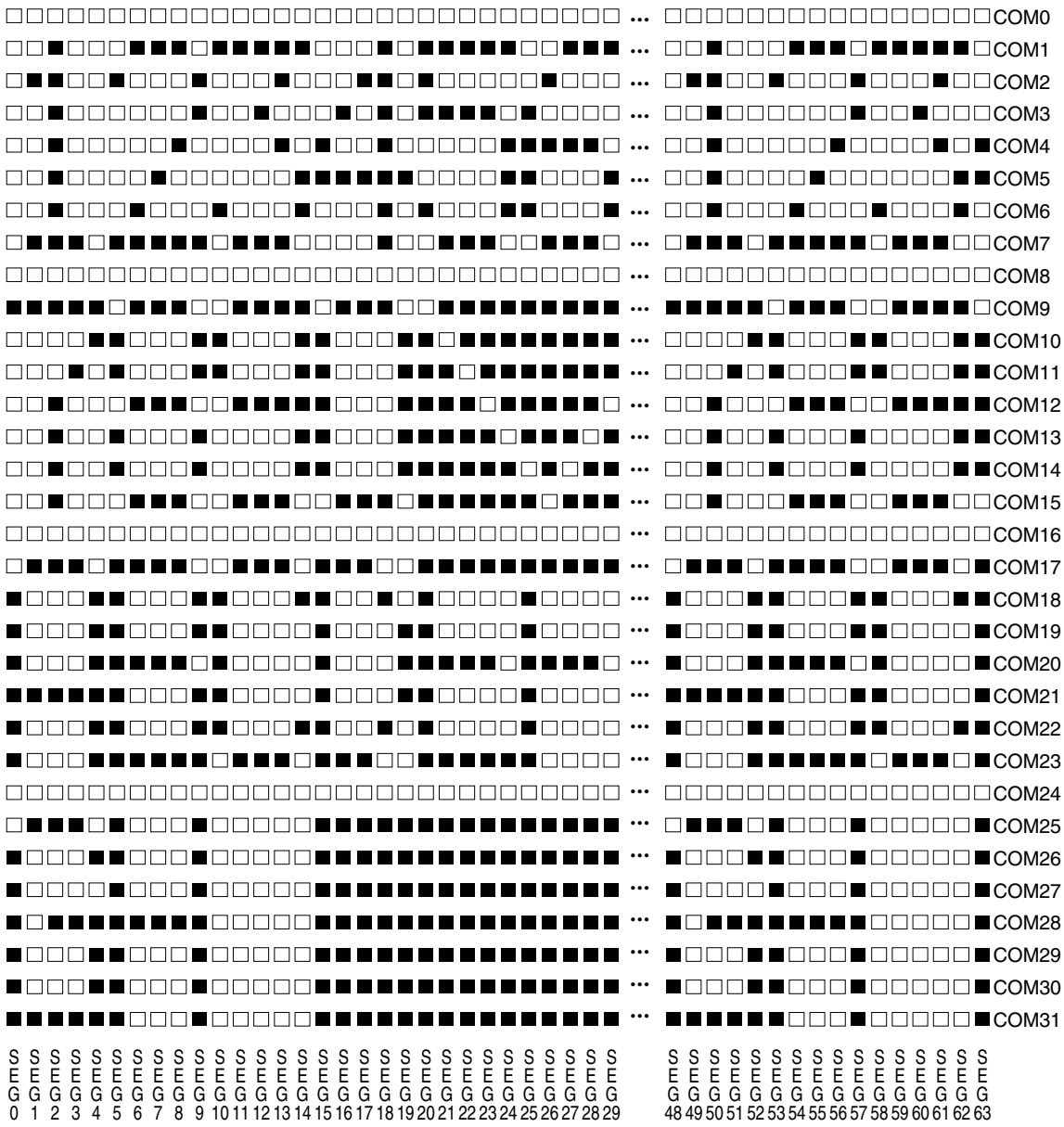
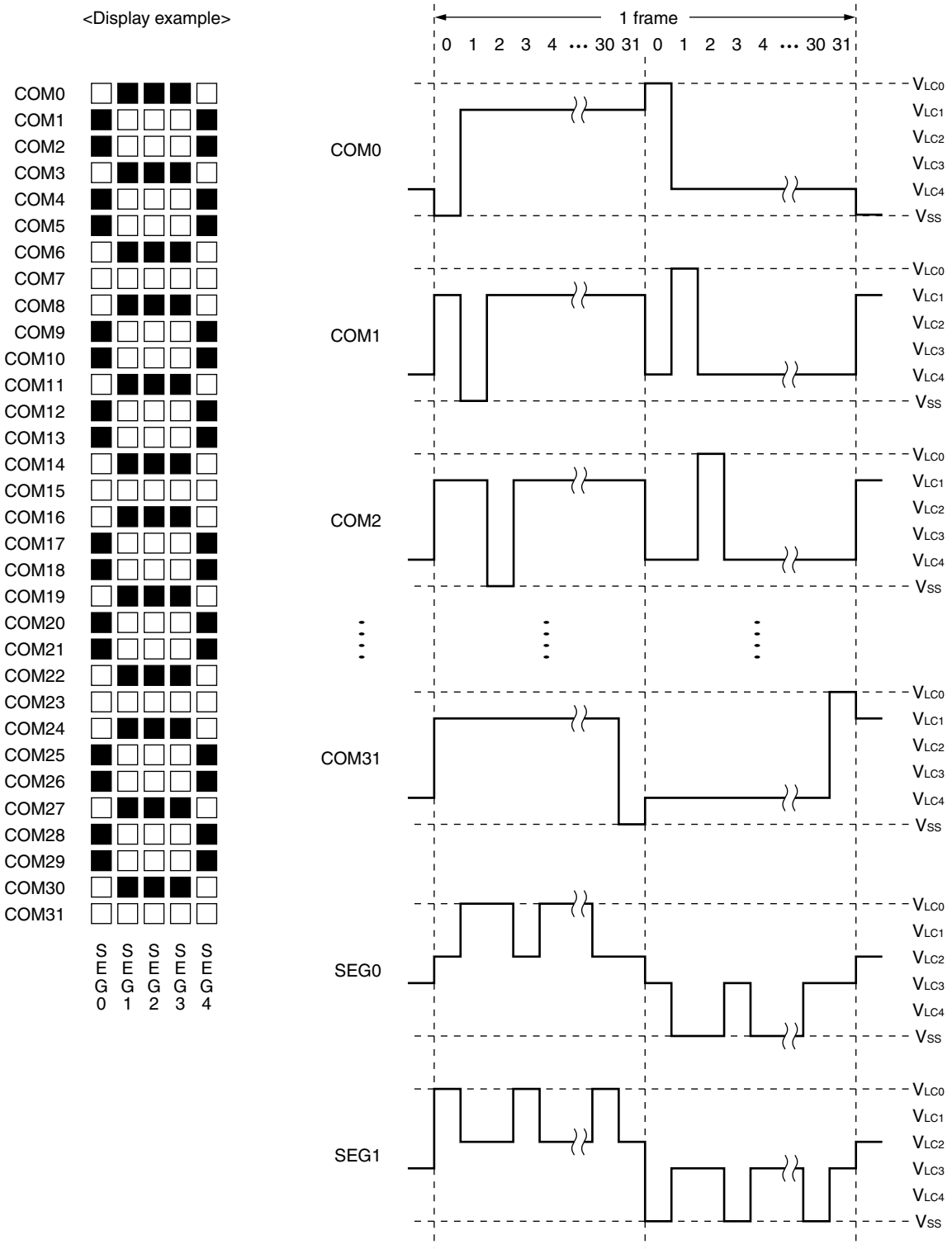


Figure 14-8. LCD Drive Waveform Examples (1/32 Duty)



14.7.4 48 × 48 mode (1/48 duty)

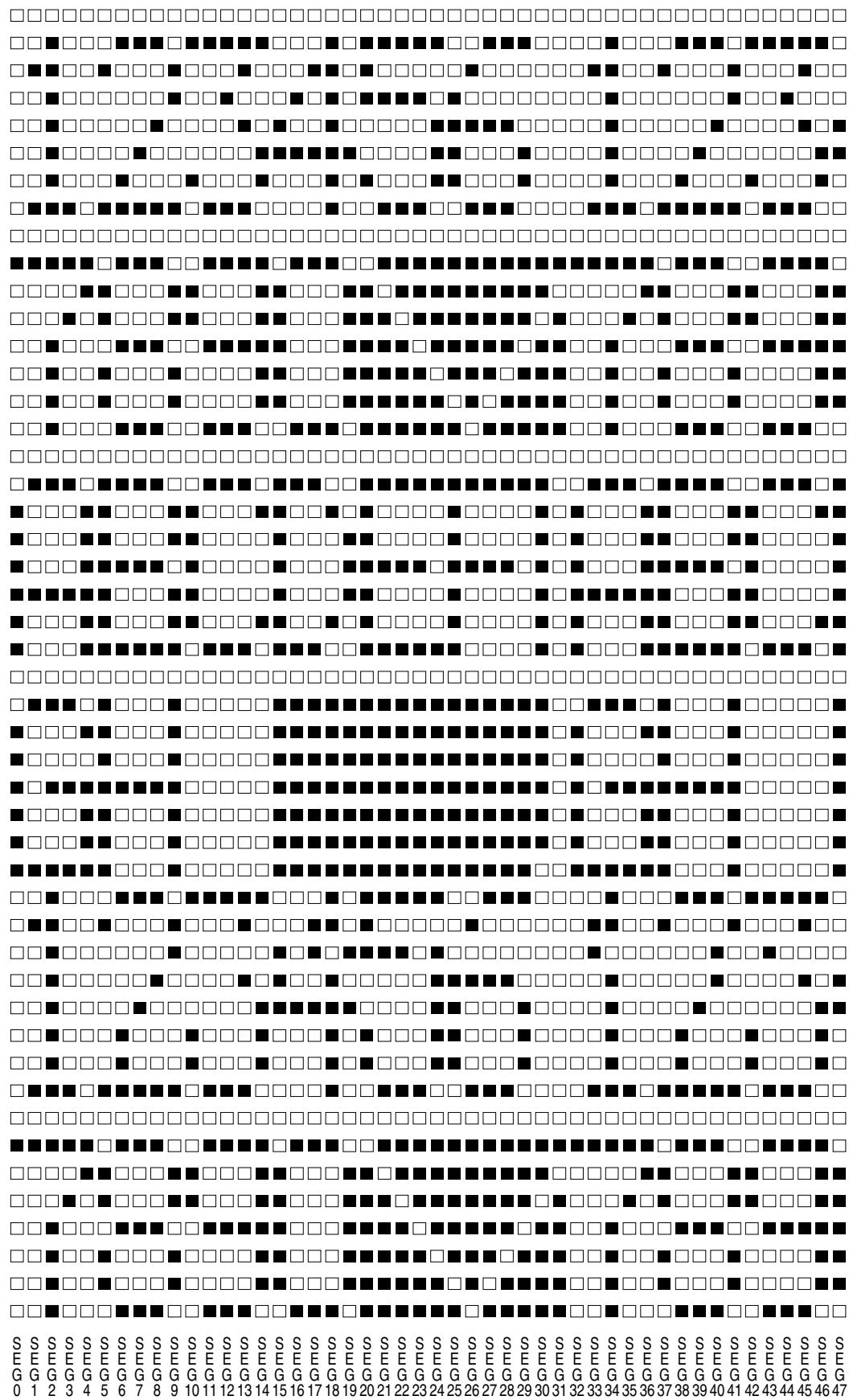
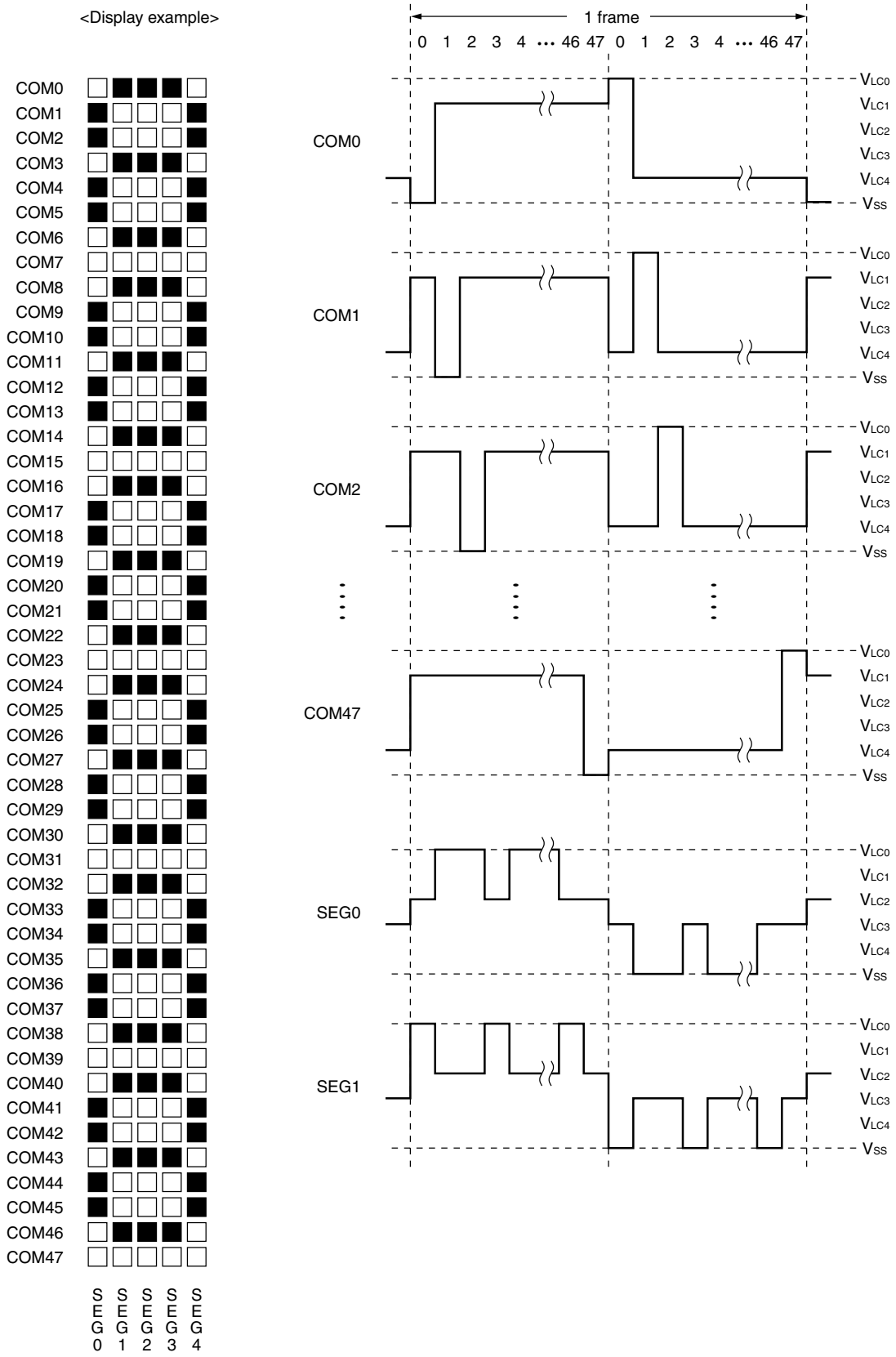


Figure 14-9. LCD Drive Waveform Examples (1/48 Duty)



CHAPTER 15 MULTIPLIER

15.1 Multiplier Function

The multiplier has the following function.

- Calculation of 8 bits × 8 bits = 16 bits

15.2 Multiplier Configuration

(1) 16-bit multiplication result storage register 0 (MUL0)

This register stores the 16-bit result of multiplication.

This register holds the result of multiplication after 16 CPU clocks have elapsed.

MUL0 is set with a 16-bit memory manipulation instruction.

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

Caution Although this register is manipulated with a 16-bit memory manipulation instruction, it can also be manipulated with an 8-bit memory manipulation instruction. When using an 8-bit memory manipulation instruction, however, access the register by means of direct addressing.

(2) Multiplication data registers A and B (MRA0 and MRB0)

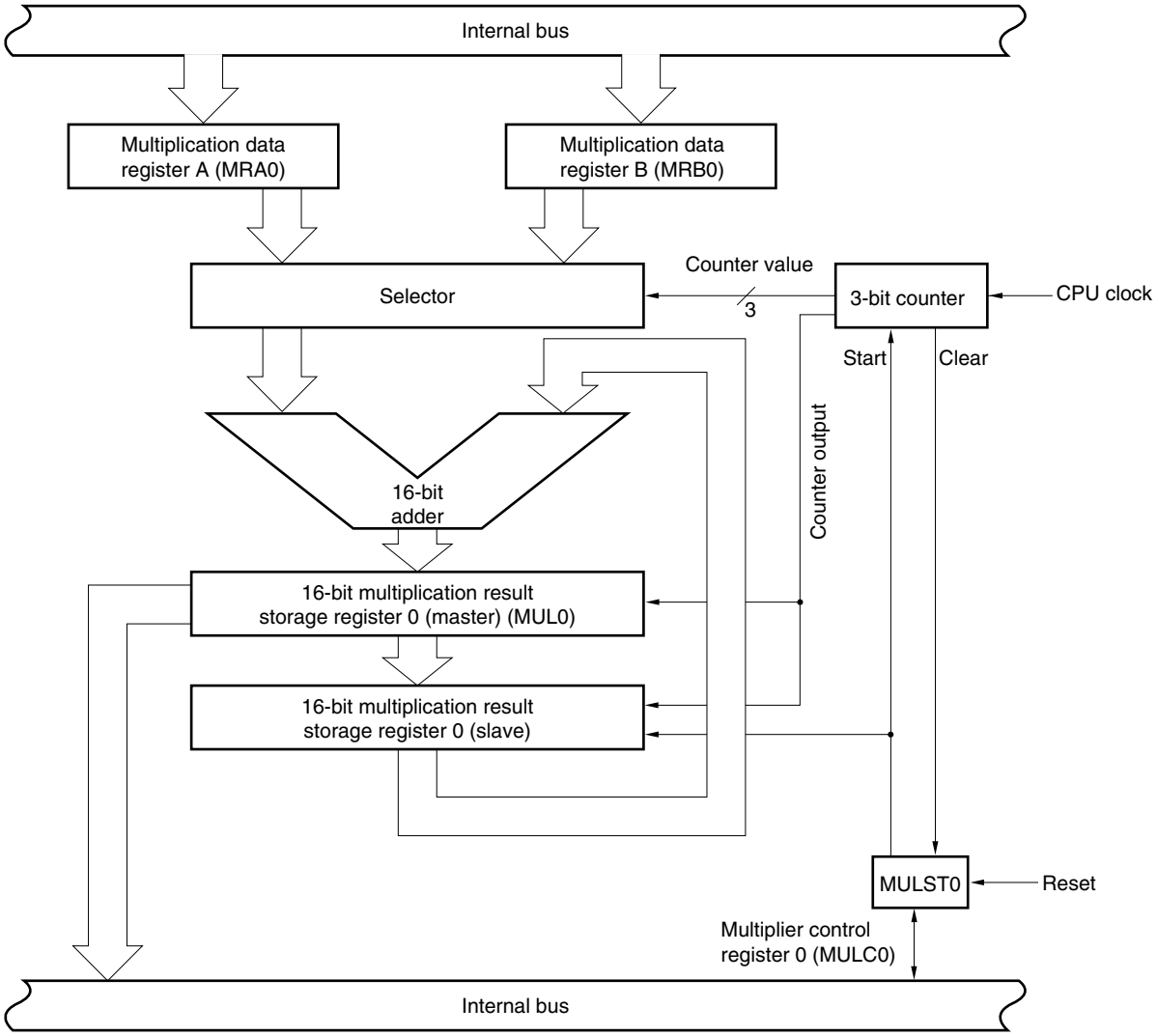
These are 8-bit multiplication data storage registers. The multiplier multiplies the values of MRA0 and MRB0.

MRA0 and MRB0 are set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes these registers undefined.

Figure 15-1 shows the block diagram of the multiplier.

Figure 15-1. Block Diagram of Multiplier



15.3 Multiplier Control Register

The multiplier is controlled by the following register.

- Multiplier control register 0 (MULC0)

(1) Multiplier control register 0 (MULC0)

MULC0 indicates the operating status of the multiplier after operation, as well as controls the multiplier.

MULC0 is set with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 15-2. Format of Multiplier Control Register 0

Symbol	7	6	5	4	3	2	1	<0>	Address	After reset	R/W
MULC0	0	0	0	0	0	0	0	MULST0	FFD2H	00H	R/W

MULST0	Multiplier operation start control bit	Operating status of multiplier
0	Stops operation after resetting counter to 0.	Operation stops
1	Enables operation	Operation in progress

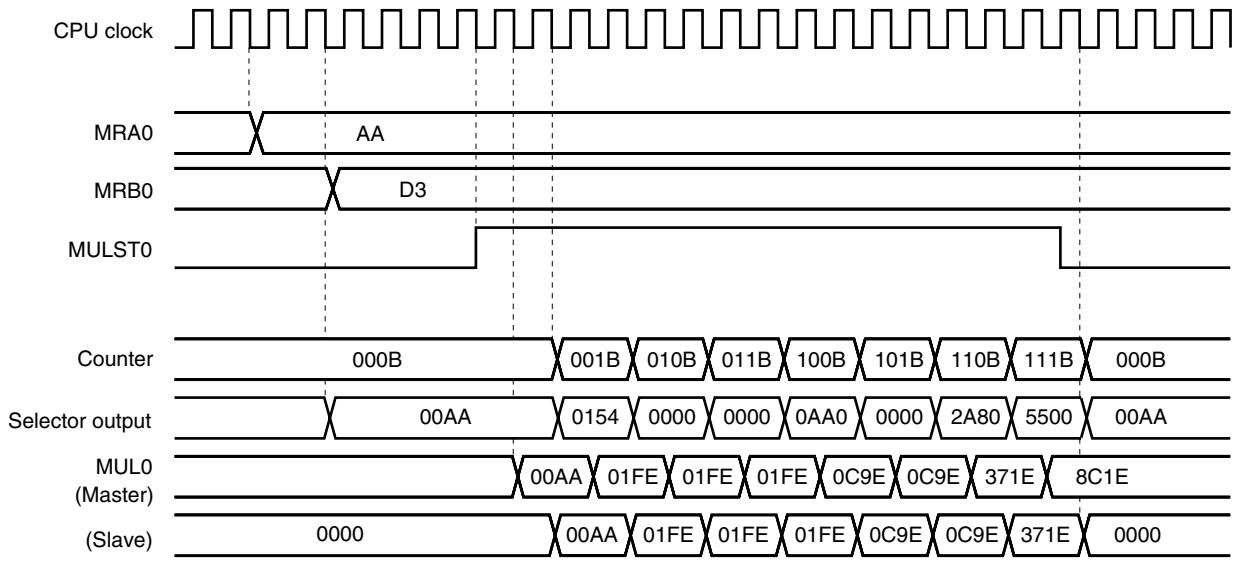
Caution Be sure to set bits 1 to 7 to 0.

15.4 Multiplier Operation

The multiplier of the μPD789835 Subseries can execute the calculation of 8 bits × 8 bits = 16 bits. Figure 15-3 shows the operation timing of the multiplier where MRA0 is set to AAH and MRB0 is set to D3H.

<1> Counting is started by setting MULST0.
<2> The data generated by the selector is added to the data of MUL0 at each CPU clock, and the counter value is incremented by one.
<3> If MULST0 is cleared when the counter value is 111B, the operation is stopped. At this time, MUL0 holds the data.
<4> While MULST0 is low, the counter and slave are cleared.

Figure 15-3. Multiplier Operation Timing (Example of AAH × D3H)

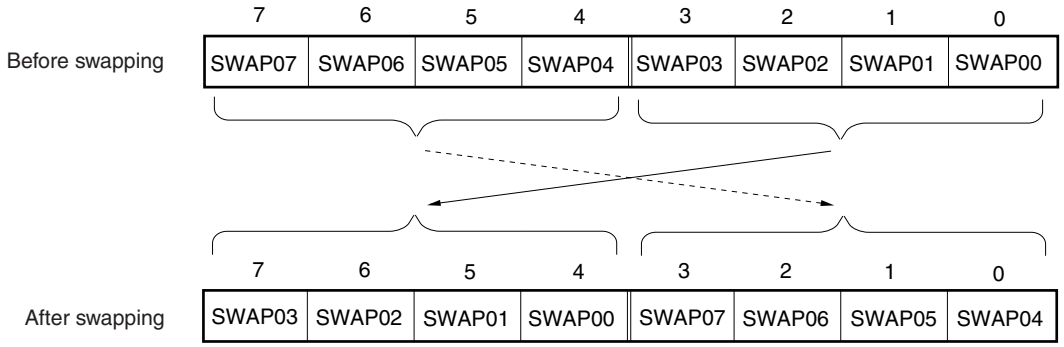


CHAPTER 16 SWAPPING (SWAP)

16.1 Function of SWAP

By performing four shift operations, it is possible to switch the contents of the higher four bits of swapping function register 0 (SWP0) with the lower four bits. Figure 16-1 shows an example of swapping.

Figure 16-1. Example of Swapping



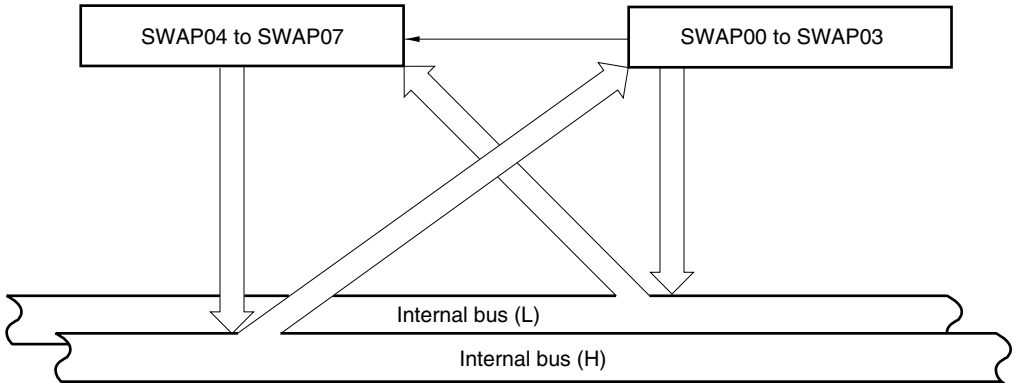
16.2 Configuration of SWAP

SWAP consists of the following hardware.

Table 16-1. SWAP Configuration

Item	Configuration
Register	Swapping function register 0 (SWP0)

Figure 16-2. SWAP Block Diagram



(1) Swapping function register 0 (SWP0)

By writing data to SWP0 and subsequently reading it back, the contents of the higher four bits and the lower four bits can be swapped.

SWP0 is set with an 8-bit memory manipulation instruction.

In write mode, $\overline{\text{RESET}}$ input makes SWP0 undefined.

In read mode, $\overline{\text{RESET}}$ input sets SWP0 to 00H.

CHAPTER 17 INTERRUPT FUNCTIONS

17.1 Interrupt Function Types

The following two types of interrupt functions are used.

(1) Non-maskable interrupt

This interrupt is acknowledged unconditionally. It does not undergo interrupt priority control and is given top priority over all other interrupt requests.

A standby release signal is generated.

One interrupt source from the watchdog timer is incorporated as a non-maskable interrupt.

(2) Maskable interrupt

This interrupt undergoes mask control. If two or more interrupts with the same priority are simultaneously generated, each interrupt has a predetermined priority as shown in Table 17-1.

A standby release signal is generated.

5 external and 15 internal interrupt sources are incorporated as maskable interrupts.

17.2 Interrupt Sources and Configuration

A total of 21 non-maskable and maskable interrupts are incorporated as interrupt sources (see **Table 17-1**).

Table 17-1. Interrupt Source List

Interrupt Type	Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Non-maskable	–	INTWDT0	Watchdog timer overflow (with watchdog timer mode 1 selected)	Internal	0002H	(A)
Maskable	0	INTWDT1	Watchdog timer overflow (with interval timer mode selected)		0004H	(B)
	1	INTP0	Pin input edge detection	External	0006H	(C)
	2	INTP1			0008H	
	3	INTCSI10	End of 3-wire serial I/O (SIO10) transmission/reception	Internal	000AH	(B)
	4	INTSER00	Occurrence of asynchronous serial interface (UART00) receive error		000CH	
	5	INTSR00	End of UART00 reception		000EH	
	6	INTST00	End of UART00 transmission		0010H	
	7	INTTM50	RIN pin input rising edge detection	External	0012H	(D)
	8	INTTM51	RIN pin input falling edge detection		0014H	
	9	INTTM52	8-bit remote control timer 50 overflow signal	Internal	0016H	(B)
	10	INTWT	Watch timer interrupt		0018H	
	11	INTWTI	Watch timer interval timer interrupt		001AH	
	12	INTTM80	Generation of match signal of 8-bit timer/event counter 80		001CH	
	13	INTTM81	Generation of match signal of 8-bit timer 81		001EH	
	14	INTTM82	Generation of match signal of 8-bit timer 82		0020H	
	15	INTTM30	Generation of match signal of 8-bit timer 30		0022H	
	16	INTTM40	Generation of match signal of 8-bit timer 40		0024H	
	17	INTTMSG0	Generation of match signal of 8-bit timer SG0		0026H	
	18	INTAD	A/D conversion completion signal		0028H	
	19	INTKR00	Key return signal detection	External	002AH	(C)

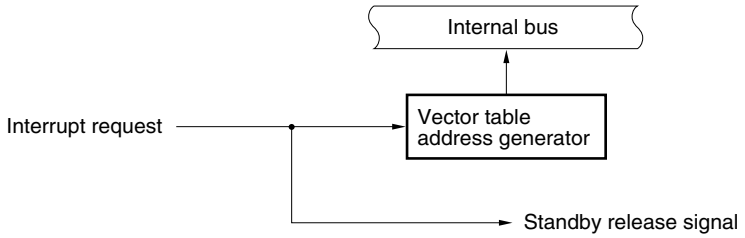
Notes 1. Priority is the priority order when more than one maskable interrupt request is generated at the same time. 0 is the highest priority and 19 is the lowest.

2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 17-1.

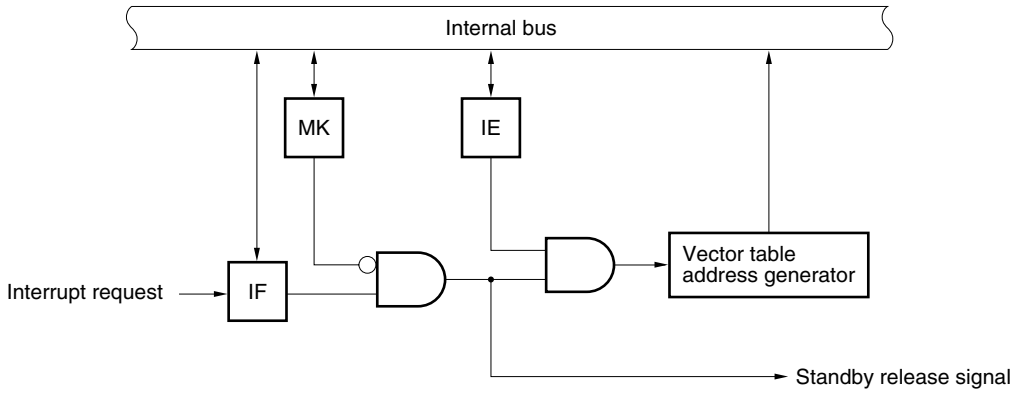
Remark There are two interrupt sources for the watchdog timer: a non-maskable interrupt (INTWDT0) and a maskable interrupt (internal) (INTWDT1). Either one (but not both) should be selected for actual use.

Figure 17-1. Basic Configuration of Interrupt Function

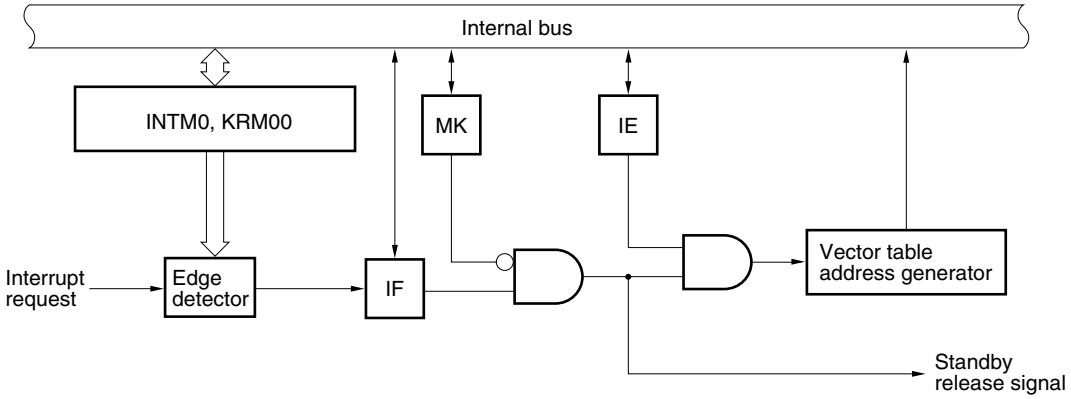
(A) Internal non-maskable interrupt



(B) Internal maskable interrupt

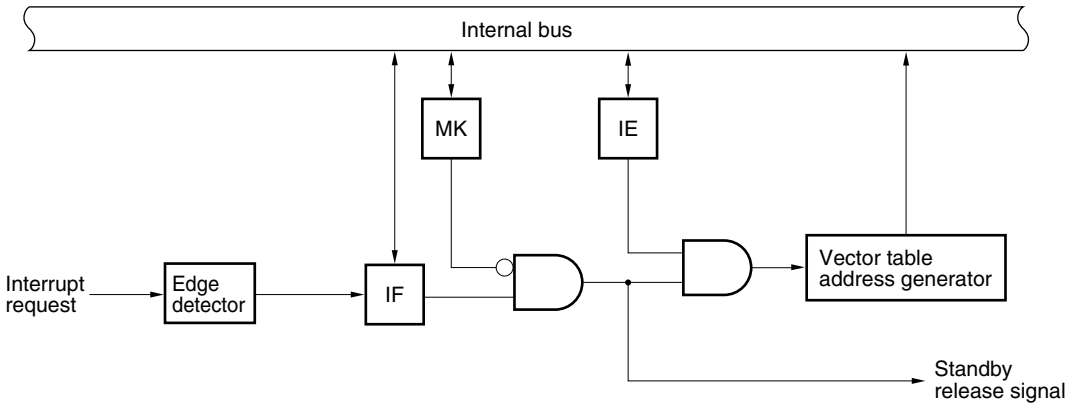


(C) External maskable interrupt



INTP0: External interrupt mode register 0
KRM00: Key return mode register 00
IF: Interrupt request flag
IE: Interrupt enable flag
MK: Interrupt mask flag

(D) External maskable interrupt (INTTM50, INTTM51)



IF: Interrupt request flag
IE: Interrupt enable flag
MK: Interrupt mask flag

17.3 Registers Controlling Interrupt Function

The following five types of registers are used to control the interrupt functions.

- Interrupt request flag registers 0 to 2 (IF0 to IF2)
- Interrupt mask flag registers 0 to 2 (MK0 to MK2)
- External interrupt mode register 0 (INTM0)
- Program status word (PSW)
- Key return mode register 00 (KRM00)

Table 17-2 lists the interrupt request flag and interrupt mask flag names corresponding to the interrupt requests.

Table 17-2. Flags Corresponding to Interrupt Request Signal Name

Interrupt Request Signal Name	Interrupt Request Flag	Interrupt Mask Flag
INTWDT1	WDTIF1	WDTMK1
INTP0	PIF0	PMK0
INTP1	PIF1	PMK1
INTCSI10	CSIIF10	CSIMK10
INTSER00	SERIF00	SERMK00
INTSR00	SRIF00	SRMK00
INTST00	STIF00	STMK00
INTWT	WTIF	WTMK
INTWTI	WTIIF	WTIMK
INTTM50	TMIF50	TMMK50
INTTM51	TMIF51	TMMK51
INTTM52	TMIF52	TMMK52
INTTM80	TMIF80	TMMK80
INTTM81	TMIF81	TMMK81
INTTM82	TMIF82	TMMK82
INTTM30	TMIF30	TMMK30
INTTM40	TMIF40	TMMK40
INTTMSG0	TMIFSG0	TMMKSG0
INTAD	ADIF	ADMK
INTKR00	KRIF00	KRMK00

(1) Interrupt request flag registers 0 to 2 (IF0 to IF2)

An interrupt request flag is set (1) when the corresponding interrupt request is generated, or when an instruction is executed. It is cleared (0) when the interrupt request is acknowledged, when the $\overline{\text{RESET}}$ signal is input, or when an instruction is executed.

IF0 to IF2 are set with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 17-2. Format of Interrupt Request Flag Registers

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
IF0	TMIF50	STIF00	SRIF00	SERIF00	CSIIF10	PIF1	PIF0	WDTIF1	FFE0H	00H	R/W
	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>			
IF1	TMIF30	TMIF82	TMIF81	TMIF80	WTIIF	WTIF	TMIF52	TMIF51	FFE1H	00H	R/W
	7	6	5	4	<3>	<2>	<1>	<0>			
IF2	0	0	0	0	KRIF00	ADIF	TMIFSG0	TMIF40	FFE2H	00H	R/W

××IF×	Interrupt request flag
0	Interrupt request signal not generated
1	Interrupt request signal generated, interrupt request status entered

Cautions 1. The WDTIF flag is R/W enabled only when the watchdog timer is used as an interval timer. If watchdog timer mode 1 or 2 is used, clear the WDTIF flag to 0.

2. Since port 2 has an alternate function as the external interrupt input, when the output level is changed by specifying the output mode of the port function, an interrupt request flag is set. Therefore, the corresponding interrupt mask flag should be set to 1 before using the output mode.

3. When an interrupt is acknowledged, the interrupt request flag is automatically cleared before entering the interrupt routine.

(2) Interrupt mask flag registers 0 to 2 (MK0 to MK2)

The interrupt mask flag is used to enable/disable the corresponding maskable interrupt servicing.

MK0 to MK2 are set with a 1-bit or 8-bit memory manipulation instruction.

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

Figure 17-3. Format of Interrupt Mask Flag Registers

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
MK0	TMMK50	STMK00	SRMK00	SERMK00	CSIMK10	PMK1	PMK0	WDTMK1	FFE4H	FFH	R/W
	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>			
MK1	TMMK30	TMMK82	TMMK81	TMMK80	WTIMK	WTMK	TMMK52	TMMK51	FFE5H	FFH	R/W
	7	6	5	4	<3>	<2>	<1>	<0>			
MK2	1	1	1	1	KRMK00	ADMK	TMMKSG0	TMMK40	FFE6H	FFH	R/W

××MK×	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Cautions 1. If the WDTMK flag is read when the watchdog timer is used in watchdog timer mode 1 or 2, its value becomes undefined.

2. Since port 2 has an alternate function as the external interrupt input, when the output level is changed by specifying the output mode of the port function, an interrupt request flag is set. Therefore, the corresponding interrupt mask flag should be set to 1 before using the output mode.

(3) External interrupt mode register 0 (INTM0)

This register is used to set the valid edge of INTP0 and INTP1.

INTM0 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets INTM0 to 00H.

Figure 17-4. Format of External Interrupt Mode Register 0

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
INTM0	0	0	ES11	ES10	ES01	ES00	0	0	FFECH	00H	R/W

ES11	ES10	Selection of INTP1 valid edge
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES01	ES00	Selection of INTP0 valid edge
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

Cautions 1. Be sure to set bits 0, 1, 6, and 7 to 0.

2. Before setting the INTM0 register, be sure to set the corresponding interrupt mask flag to 1 to disable interrupts.
Afterward, clear the interrupt request flag to 0, then clear the interrupt mask flag to 0 to enable interrupts.

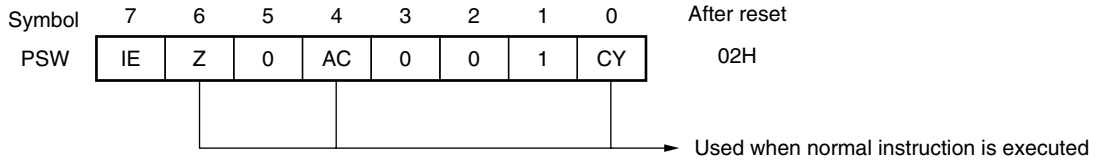
(4) Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current status for interrupt requests. The IE flag to set maskable interrupt enable/disable is mapped in the PSW.

Besides 8-bit unit read/write, this register can carry out operations with a bit manipulation instruction and dedicated instructions (EI, DI). When a vectored interrupt is acknowledged, the PSW is automatically saved into a stack, and the IE flag is reset to 0.

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

Figure 17-5. Configuration of Program Status Word



IE	Interrupt acknowledgement enabled/disabled
0	Disabled
1	Enabled

(5) Key return mode register 00 (KRM00)

This register sets the pin that is used to detect the key return signal (falling edge of port 0).

KRM00 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets KRM00 to 00H.

Figure 17-6. Format of Key Return Mode Register 00

Symbol	<7>	<6>	<5>	<4>	3	2	1	<0>	Address	After reset	R/W
KRM00	KRM007	KRM006	KRM005	KRM004	0	0	0	KRM000	FFF5H	00H	R/W

KRM000	Control of key return signal detection
0	Key return signal not detected
1	Key return signal detected (P00 to P03 falling edge detection)

KRM00n	Control of key return signal detection
0	Key return signal not detected
1	Key return signal detected (P0n falling edge detection)

Remark n = 4 to 7

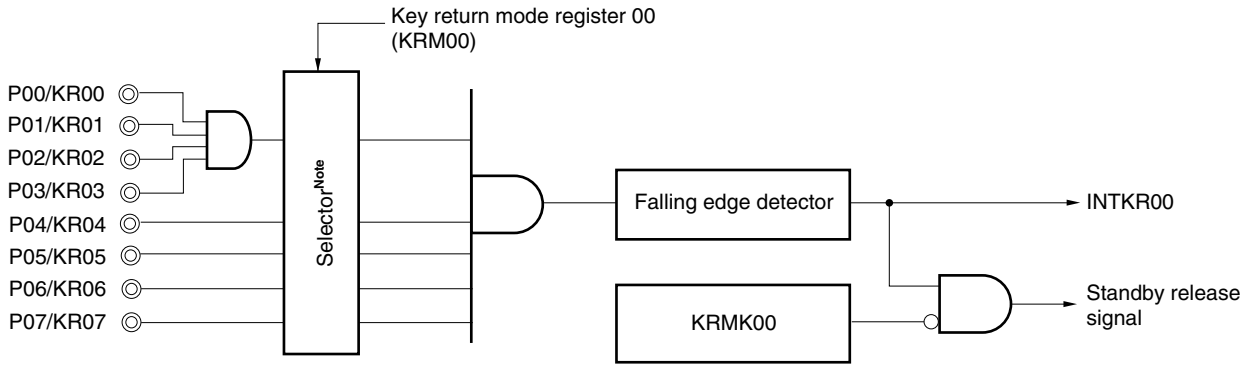
Cautions 1. Bits 1 to 3 must all be set to 0.

2. Before setting KRM00, set (1) bit 7 (KRMK00) of MK0 to disable interrupts. After KRM00 is set, clear (0) KRMK00 after clearing (0) bit 7 (KRIF00) of IF0 to enable interrupts.

3. On-chip pull-up resistors are automatically connected in the input mode to the pins specified for key return signal detection (P00 to P07). Although these resistors are disconnected when the mode changes to output, key return signal detection continues unchanged.

4. A key return signal cannot be detected while any one of the pins specified for key return detection is low level even when a falling edge occurs at another pin.

Figure 17-7. Block Diagram of Key Return Signal Detector



Note For selecting the pin to be used as the falling edge input.

17.4 Interrupt Servicing Operation

17.4.1 Non-maskable interrupt request acknowledgment operation

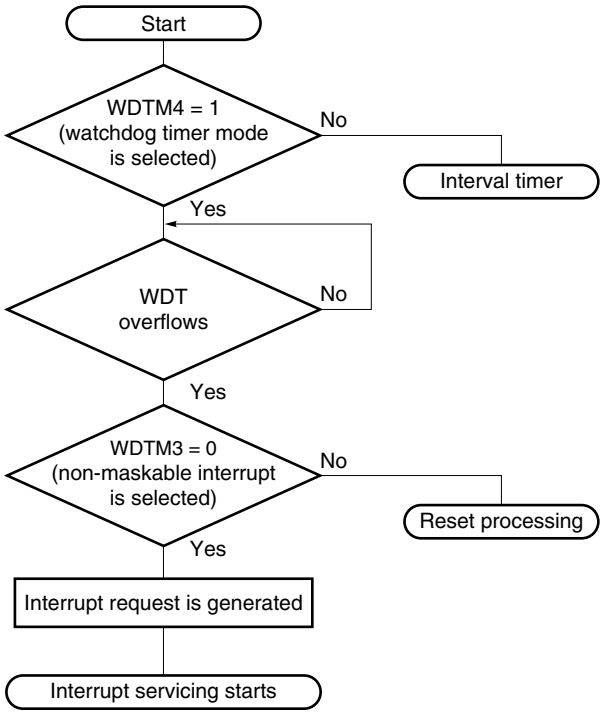
A non-maskable interrupt request is acknowledged unconditionally even when interrupts are disabled. It is not subject to interrupt priority control and takes precedence over all other interrupts.

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 17-8 shows the flow from non-maskable interrupt request generation to acknowledgement, Figure 17-9 shows the timing of non-maskable interrupt acknowledgement, and Figure 17-10 shows the acknowledgement operation when multiple non-maskable interrupts are generated.

Caution During non-maskable interrupt service program execution, do not input another non-maskable interrupt request; otherwise, the service program will be interrupted and the new non-maskable interrupt request will be acknowledged.

Figure 17-8. Flow from Generation of Non-Maskable Interrupt Request to Acknowledgment



WDTM: Watchdog timer mode register

WDT: Watchdog timer

Figure 17-9. Timing of Non-Maskable Interrupt Request Acknowledgment

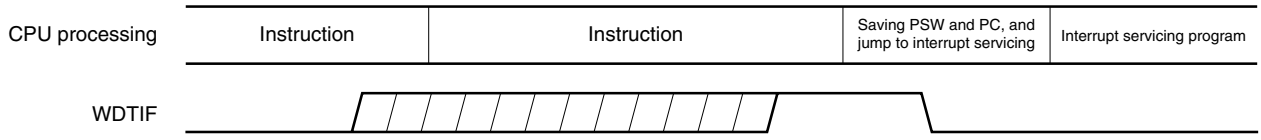
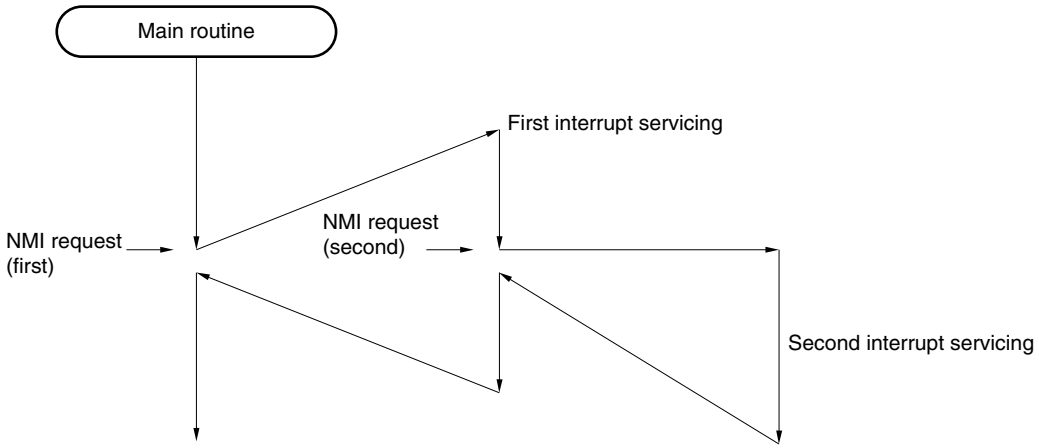


Figure 17-10. Non-Maskable Interrupt Request Acknowledgment



17.4.2 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 17-3.

Refer to **Figures 17-12** and **17-13** for the timing of interrupt request acknowledgement.

Table 17-3. 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.

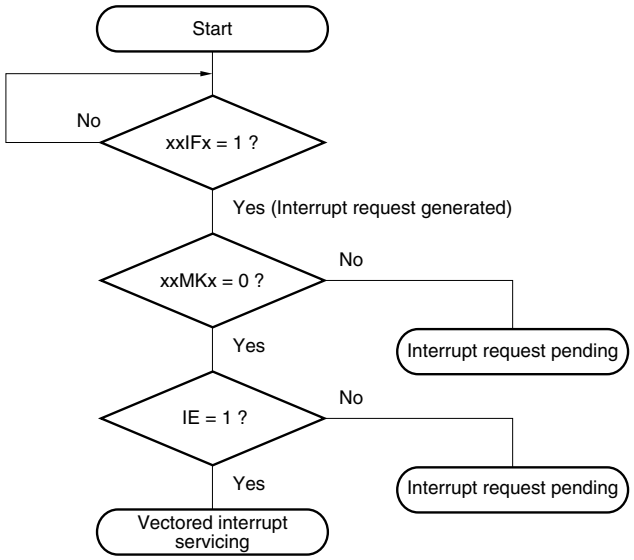
A pending interrupt is acknowledged when the status in which it can be acknowledged is set.

Figure 17-11 shows the algorithm of interrupt request acknowledgement.

When a maskable interrupt request is acknowledged, the PSW and PC are saved to the stack in that order, the IE flag is reset to 0, and the data in the vector table determined for each interrupt request is loaded to the PC, and execution branches.

To return from interrupt servicing, use the RETI instruction.

Figure 17-11. Interrupt Request Acknowledgment Program Algorithm

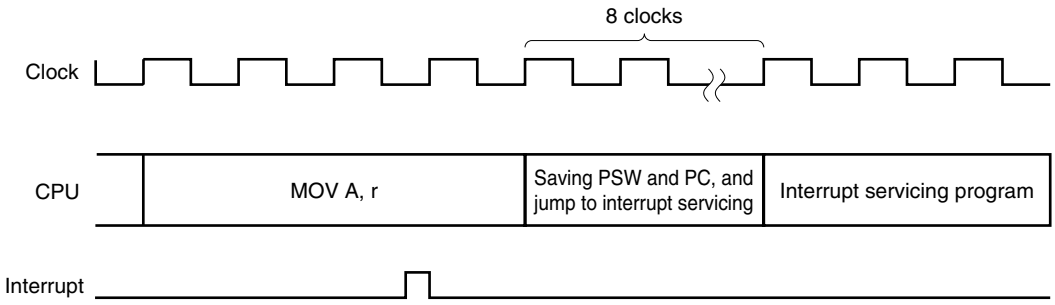


xxIFx: Interrupt request flag

xxMKx: Interrupt mask flag

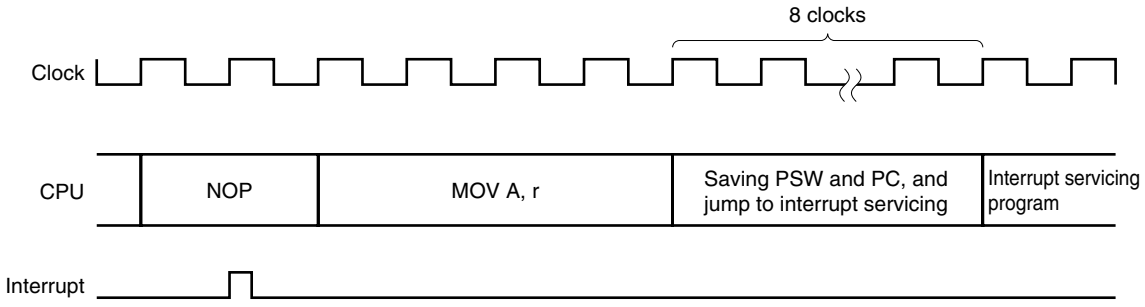
IE: Flag to control maskable interrupt request acknowledgement (1 = enable, 0 = disable)

Figure 17-12. Interrupt Request Acknowledgment Timing (Example: MOV A, r)



If the interrupt request has generated an interrupt request flag (xxIFx) 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 17-12 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 17-13. Interrupt Request Acknowledgment Timing (When Interrupt Request Flag Is Generated in Final Clock Under Execution)



If the interrupt request flag (xxIFx) is generated in the final clock of the instruction, interrupt request acknowledgment processing will begin after execution of the next instruction is complete.

Figure 17-13 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.

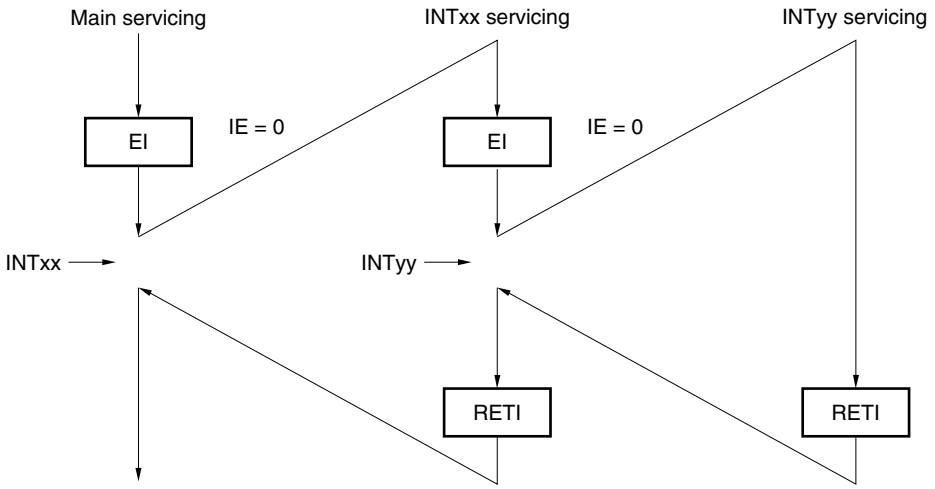
Caution When interrupt request flag registers 0 to 2 (IF0 to IF2), or interrupt mask flag registers 0 to 2 (MK0 to MK2) are being accessed, interrupt requests will be held pending.

17.4.3 Multiple interrupt servicing

Multiple interrupts, in which another interrupt request is acknowledged while an interrupt request being serviced, can be serviced using the priority order. If multiple interrupts are generated at the same time, they are serviced in the order of the priority assigned to each interrupt request in advance (refer to **Table 17-1**).

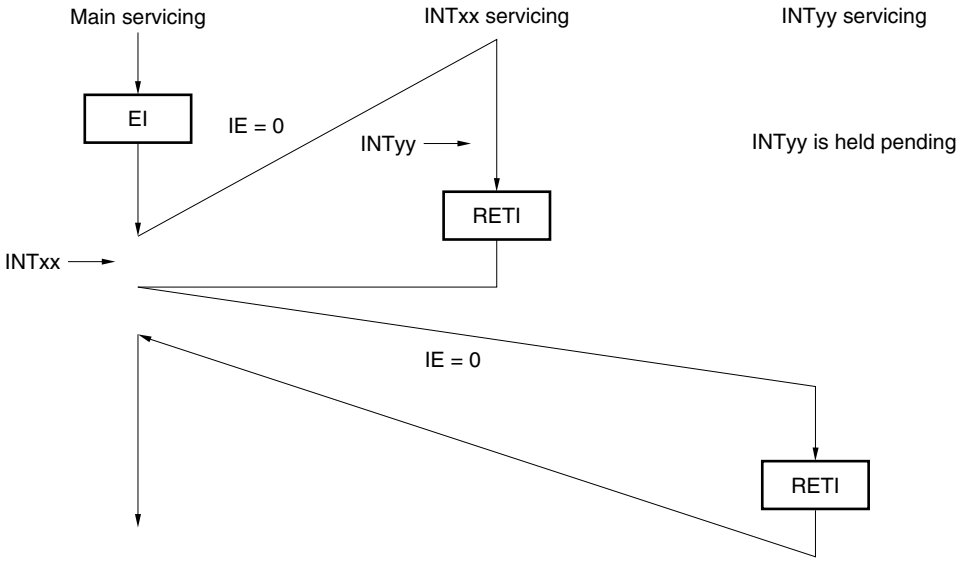
Figure 17-14. Example of Multiple Interrupts

Example 1. Acknowledging multiple interrupts



The interrupt request INTyy is acknowledged during the servicing of interrupt INTxx and multiple interrupts are performed. Before each interrupt request is acknowledged, the EI instruction is issued and the interrupt request is enabled.

Example 2. Multiple interrupts are not performed because interrupts are disabled



Because interrupt requests are disabled (the EI instruction has not been issued) in interrupt INTxx servicing, the interrupt request INTyy is not acknowledged and multiple interrupts are not performed. INTyy is held pending and is acknowledged after INTxx servicing is completed.

IE = 0: Interrupt requests disabled

17.4.4 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. These instructions (interrupt request pending instructions) are as follows.

- Instructions that manipulate interrupt request flag registers 0 to 2 (IF0 to IF2)
- Instructions that manipulate interrupt mask flag registers 0 to 2 (MK0 to MK2)

CHAPTER 18 STANDBY FUNCTION

18.1 Standby Function and Configuration

18.1.1 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 that this mode can be used for intermittent operation. However, some time is required until the system clock oscillator stabilizes after the STOP mode has been released. If processing must be resumed immediately by using an interrupt request, therefore, use the HALT mode.

In both modes, the previous contents of the registers, flags, and data memory before setting the standby mode are all retained. In addition, the statuses of the output latch of the I/O ports and output buffer are also retained.

Caution To set the STOP mode, be sure to stop the operations of the peripheral hardware, and then execute the STOP instruction.

18.1.2 Registers controlling standby function

The standby function is controlled by the following two registers.

- Oscillation stabilization time selection register (OSTS) (only when ceramic/crystal oscillation is selected)
- Power supply control register 0 (PSC0)

(1) Oscillation stabilization time selection register (OSTS) (only when ceramic/crystal oscillation is selected)

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 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}/fx, and does not depend on OSTS.

Caution When RC oscillation is selected, the oscillation stabilization time cannot be selected using OSTS.

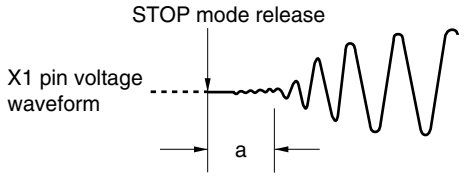
In the case of RC oscillation, the oscillation stabilization time is fixed to 2^7/fcc.

Figure 18-1. Format of Oscillation Stabilization Time Selection Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0	FFFAH	04H	R/W

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection
0	0	0	2^{12}/fx (819 μs)
0	1	0	2^{15}/fx (6.55 ms)
1	0	0	2^{17}/fx (26.2 ms)
Other than above			Setting prohibited

Caution The wait time after the STOP mode is released does not include the time from STOP mode release to clock oscillation start ("a" in the figure below), regardless of whether STOP mode is released by $\overline{\text{RESET}}$ input or by interrupt generation.



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

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

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

(2) Power supply control register 0 (PSC0)

This register is used to switch the power supply of the subsystem clock. By setting the power supply of the subsystem clock to V_{ROUT0}, the power consumption can be reduced.

PSC0 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{RESET}$ input sets PSC0 to 00H.

Figure 18-2. Format of Power Supply Control Register 0

Symbol	7	6	5	4	3	2	1		Address	After reset	R/W
PSC0	0	0	0	0	0	0	0	PSC00	FFAFH	00H	R/W

PSC00	Selection of subsystem clock power supply
0	V_{DD}
1	V_{ROUT0} (1.4 V TYP.)

Note Bits 1 to 7 must be set to 0.

18.2 Standby Function Operation

18.2.1 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 18-1. Operation Statuses in HALT Mode

Item		Operation Status in HALT Mode Set While Main System Clock Is Operating		Operation Status in HALT Mode Set While Subsystem Clock Is Operating	
		Subsystem Clock Operating	Subsystem Clock Stopped	Main System Clock Operating	Main System Clock Stopped
Main system clock		Oscillation enabled			Oscillation stopped
CPU		Operation stopped			
Port (output latch)		Status prior to setting HALT mode is retained			
8-bit timer/event counter 80		Operation enabled			Operation enabled[Note 1]
8-bit timer 81		Operation enabled	Operation enabled[Note 2]	Operation enabled	Operation enabled[Note 3]
8-bit timer 82		Operation enabled	Operation enabled[Note 2]	Operation enabled	Operation enabled[Note 3]
8-bit timer 30		Operation enabled			Operation stopped
8-bit timer 40					
8-bit remote control timer 50		Operation stopped			
Sound generator		Operation enabled	Operation enabled[Note 2]	Operation enabled	Operation enabled[Note 3]
Watch timer		Operation enabled	Operation enabled[Note 2]	Operation enabled	Operation enabled[Note 3]
Watchdog timer		Operation enabled	Operation enabled[Note 2]	Operation enabled	Operation enabled[Note 3]
Serial interface	SIO	Operation enabled			Operation enabled[Note 1]
	UART	Operation enabled			Operation stopped
A/D converter		Operation stopped			
LCD controller/driver		Operation enabled	Operation enabled[Note 2]	Operation enabled	Operation enabled[Note 3]
External interrupt		Operation enabled[Note 4]			

Notes 1. Operation is enabled only when an external clock has been selected.

2. Operation is enabled when the main system clock has been selected.

3. Operation is enabled when the subsystem clock has been selected.

4. Non-masked maskable interrupt

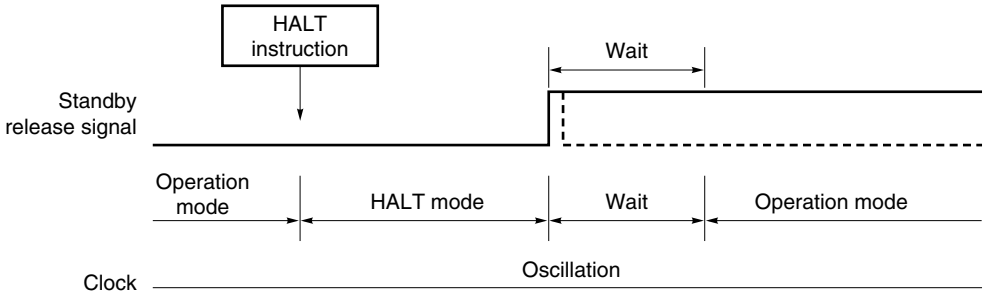
(2) Releasing HALT mode

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 18-3. Releasing HALT Mode by Interrupt



Remarks 1. The broken line indicates the case where the interrupt request that has released the standby mode is acknowledged.

2. The wait time is as follows:

- When vectored interrupt servicing is performed: 9 to 10 clocks
- When vectored interrupt servicing is not performed: 1 to 2 clocks

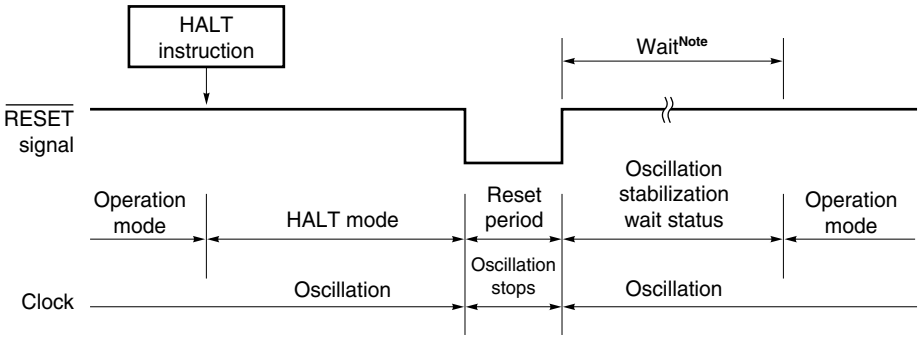
(b) Releasing by non-maskable interrupt request

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 18-4. Releasing HALT Mode by $\overline{\text{RESET}}$ Input



Note The wait time differs depending on the main system clock selected.
When ceramic/crystal oscillation is selected: $2^{15}/f_X$ (6.55 ms)
When RC oscillation is selected: $2^7/f_{CC}$ (64 μs)

Remarks 1. f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)
2. f_{CC}: Main system clock oscillation frequency (RC oscillation)
3. The parenthesized values apply to operation at f_X = 5.0 MHz or f_{CC} = 2.0 MHz.

Table 18-2. Operation After Releasing HALT Mode

Releasing Source	MK××	IE	Operation
Maskable interrupt request	0	0	Executes next address instruction
	0	1	Executes interrupt servicing
	1	×	Retains HALT mode
Non-maskable interrupt request	–	×	Executes interrupt servicing
$\overline{\text{RESET}}$ input	–	–	Reset processing

×: Don't care

18.2.2 STOP mode

(1) Setting and operation status of STOP mode

The STOP mode is set by executing the STOP instruction.

Caution Because the standby mode can be released by an interrupt request signal, the standby mode is released as soon as it is set if there is an interrupt source whose interrupt request flag is set and interrupt mask flag is reset. When the STOP mode is set, therefore, the HALT mode is set immediately after the STOP instruction has been executed, the wait time set by the oscillation stabilization time selection register (OSTS) elapses, and then an operation mode is set.

The operation statuses in the STOP mode are shown in the following table.

Table 18-3. Operation Statuses in STOP Mode

Item		Operation Status in STOP Mode	
		Subsystem Clock Operating	Subsystem Clock Stopped
Main system clock		Oscillation stopped	
CPU		Operation stopped	
Port (output latch)		Status prior to setting STOP mode is retained	
8-bit timer/event counter 80		Operation enabled[Note 1]	
8-bit timer 81		Operation enabled[Note 2]	Operation stopped
8-bit timer 82		Operation enabled[Note 2]	Operation stopped
8-bit timer 30		Operation stopped	
8-bit timer 40			
8-bit remote control timer 50		Operation stopped	
Sound generator		Operation enabled[Note 2]	Operation stopped
Watch timer		Operation enabled[Note 2]	Operation stopped
Watchdog timer		Operation enabled[Note 2]	Operation stopped
Serial interface	SIO	Operation enabled[Note 1]	
	UART	Operation stopped	
A/D converter		Operation stopped	
LCD controller/driver		Operation enabled[Note 2]	Operation stopped
External interrupt		Operation enabled[Note 3]	

Notes 1. Operation is enabled only when an external clock has been selected.

2. Operation is enabled when the subsystem clock has been selected.

3. Non-masked maskable interrupt

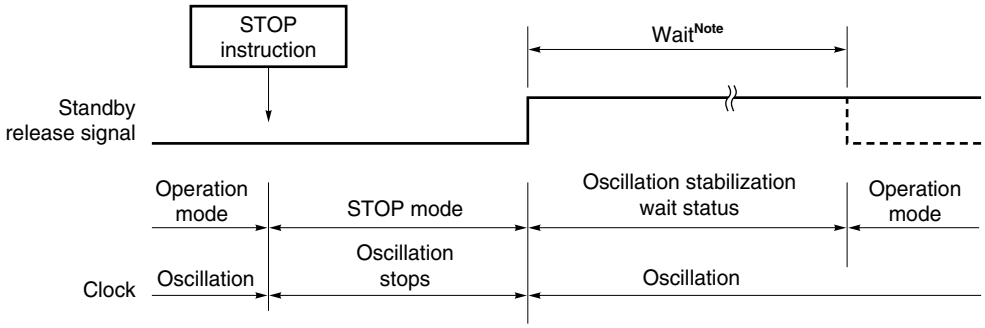
(2) Releasing STOP mode

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 18-5. Releasing STOP Mode by Interrupt



Note The wait time differs depending on the main system clock selected.

When ceramic/crystal oscillation is selected: Can be selected by OSTS register (for details, refer to **Figure 18-1 Format of Oscillation Stabilization Time Selection Register**)

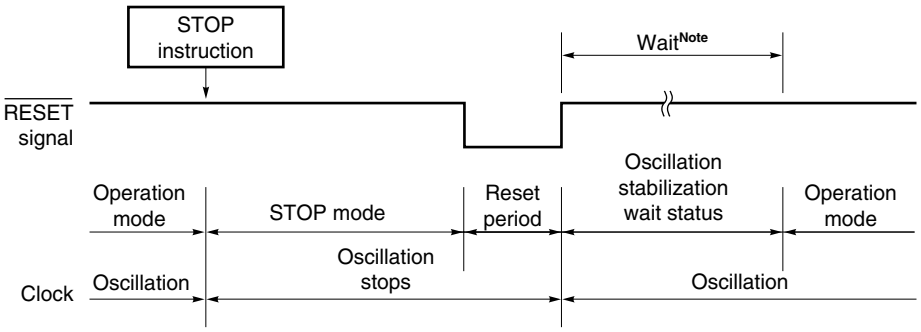
When RC oscillation is selected: Fixed at $2^7/f_{CC}$

Remark The broken line indicates the case where the interrupt request that has released the standby mode is acknowledged.

(b) Releasing by $\overline{\text{RESET}}$ input

When the STOP mode is released by the $\overline{\text{RESET}}$ signal, the reset operation is performed after the oscillation stabilization time has elapsed.

Figure 18-6. Releasing STOP Mode by $\overline{\text{RESET}}$ Input



Note The wait time differs depending on the main system clock selected.
When ceramic/crystal oscillation is selected: 2^{15}/fx (6.55 ms)
When RC oscillation is selected: 2^{7}/fcc (64 μs)

Remarks 1. fx: Main system clock oscillation frequency (ceramic/crystal oscillation)
2. fcc: Main system clock oscillation frequency (RC oscillation)
3. The parenthesized values apply to operation at fx = 5.0 MHz or fcc = 2.0 MHz.

Table 18-4. Operation After Releasing STOP Mode

Releasing Source	MK××	IE	Operation
Maskable interrupt request	0	0	Executes next address instruction
	0	1	Executes interrupt servicing
	1	×	Retains STOP mode
$\overline{\text{RESET}}$ input	–	–	Reset processing

×: Don't care

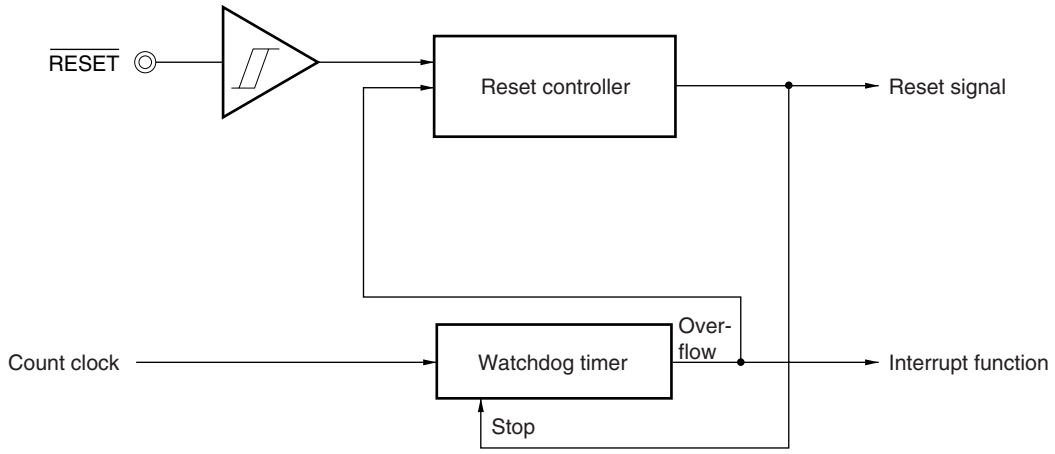
CHAPTER 19 RESET FUNCTION

There are two ways to issue a reset signal.

(1) External reset input via $\overline{\text{RESET}}$ pin

(2) Internal reset by watchdog timer program loop time detection

Figure 19-1. Block Diagram of Reset Function



(1) External reset input via $\overline{\text{RESET}}$ pin

Program execution starts at the address written to 0000H and 0001H after the $\overline{\text{RESET}}$ signal is input.

When a low level is input to the $\overline{\text{RESET}}$ pin, a reset is triggered and all hardware is set to the status shown in Table 19-1. Each pin has high impedance during reset input or during the oscillation stabilization time immediately after a reset is released.

When a high level is input to the $\overline{\text{RESET}}$ pin, the reset is released and program execution is started after the oscillation stabilization time has elapsed.

(2) Internal reset by watchdog timer runaway time detection

There are no functional differences between external and internal resets, since in both cases program execution starts at the address written to 0000H and 0001H after the $\overline{\text{RESET}}$ signal is input.

When the watchdog timer overflows, a reset is triggered and all hardware is set to the status shown in Table 19-1. Each pin has high impedance during reset input or during the oscillation stabilization time immediately after a reset is released.

The reset is automatically released after a reset, and program execution is started after the oscillation stabilization time has elapsed.

Cautions 1. For an external reset, input a low level for at least 10 μs to the $\overline{\text{RESET}}$ pin.

2. When the STOP mode is released by reset, the STOP mode contents are retained during reset input. However, the port pins are set to high impedance.

3. The oscillation stabilization time for ceramic/crystal oscillation after a reset is released differs from that for RC oscillation.

Ceramic/crystal oscillation: $2^{15}/f_X$

RC oscillation: $2^7/f_{CC}$

Figure 19-2. Reset Timing by $\overline{\text{RESET}}$ Input

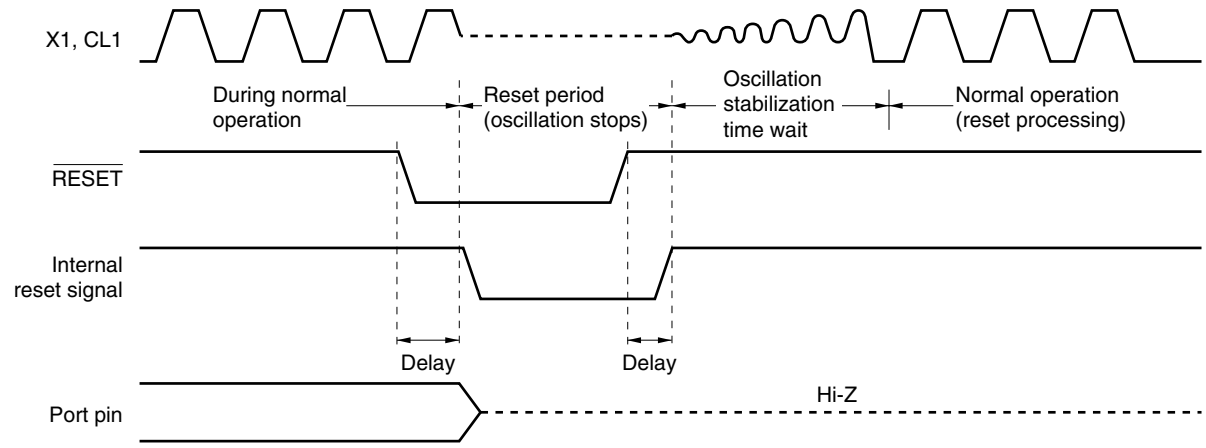


Figure 19-3. Reset Timing by Overflow in Watchdog Timer

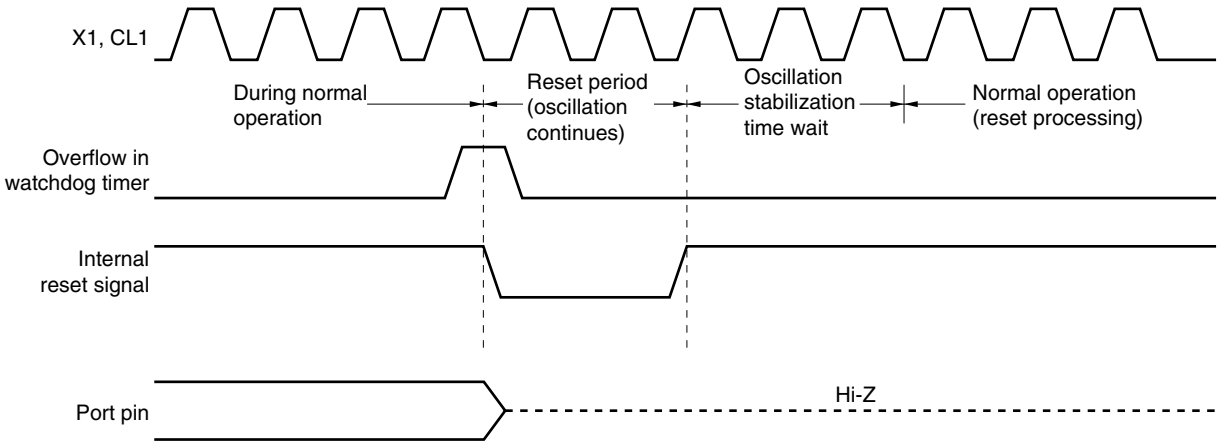


Figure 19-4. Reset Timing by $\overline{\text{RESET}}$ Input in STOP Mode

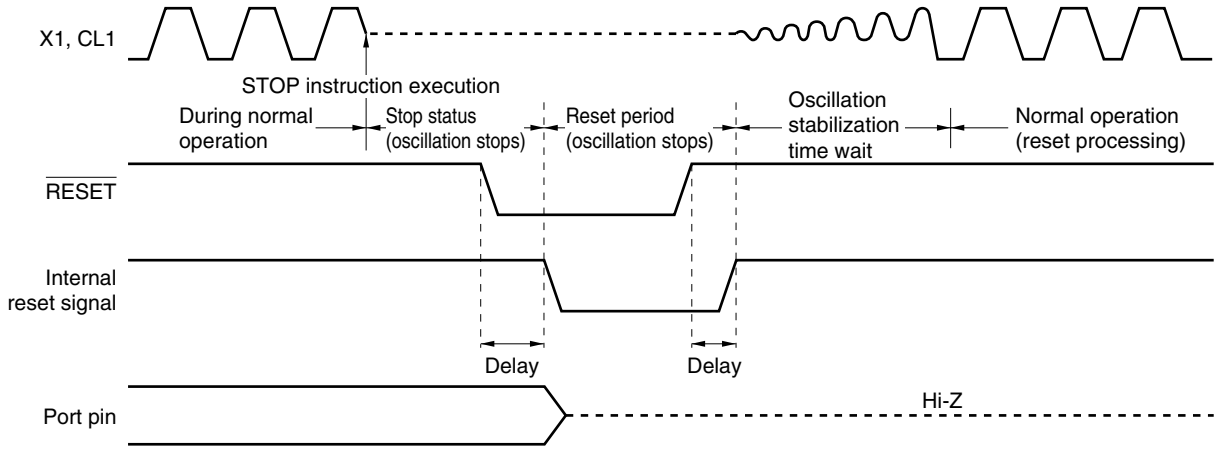


Table 19-1. Status of Each Hardware After Reset (1/2)

Hardware		Status After Reset
Program counter (PC)[Note 1]		Contents of reset vector table (0000H, 0001H) are set
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined[Note 2]
	General-purpose registers	Undefined[Note 2]
Ports (P0 to P3) (output latch)		00H
Port mode registers (PM0 to PM3)		FFH
Pull-up resistor option registers (PU0, PUB2, PUB3)		00H
Processor clock control register (PCC)		02H
Subsystem clock oscillation mode register (SCKM)		00H
Subsystem clock control register (CSS)		00H
Oscillation stabilization time selection register (OSTS)		04H
Power supply control register 0 (PSC0)		00H
8-bit timer	Timer counters (TM30, TM40, TM80 to TM82)	00H
	Compare registers (CR30, CR40, CRH40, CR80 to CR82)	Undefined
	Mode control registers (TMC30, TMC40, TMC80 to TMC82)	00H
	Carrier generator output control register (TCA40)	00H
8-bit remote control timer	Timer counter (TM50)	00H
	Capture register (CP50, CP51)	00H
	Control register (TMC50)	00H
Sound generator	Timer counter (TMSG0)	00H
	Compare register (CRSG0)	Undefined
	Frequency setting register (SGFC00)	00H
	Carrier generator output control register (TCASG0)	00H
	Mode control register (TMCSG0)	00H
	P3 function register (PF3)	00H

Notes 1. During reset input and oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined.

For all other hardware, the status during reset input and oscillation stabilization time wait remains unchanged after reset.

2. If the reset signal is input in the standby mode, the status before reset is retained even after reset.

Table 19-1. Status of Each Hardware After Reset (2/2)

Hardware		Status After Reset
Watch timer	Mode control register (WTM)	00H
Watchdog timer	Clock selection register (WDCS)	00H
	Mode register (WDTM)	00H
Serial interface	Mode register (CSIM10)	00H
	Transmit/receive shift register (SIO10)	Undefined
	Transmit shift register (TXS00)	Undefined
	Receive buffer register (RXB00)	FFH
	Asynchronous serial interface mode register (ASIM00)	00H
	Asynchronous serial interface status register (ASIS00)	00H
	Baud rate generator control register (BRGC00)	00H
A/D converter	Conversion result register (ADCR)	Undefined
	Mode register (ADM)	00H
	Input selection register (ADS)	00H
LCD controller/driver	Display mode register (LCDM20)	00H
	Clock control register (LCDC20)	00H
	Boost voltage setting register (VLCD00)	00H
Multiplier	Multiplication result storage register (MUL0)	Undefined
	Data registers (MRA0, MRB0)	Undefined
	Control register (MULC0)	00H
SWAP	Swap function register (SWP0)	**Note**
Interrupt	Request flag registers (IF0 to IF2)	00H
	Mask flag registers (MK0 to MK2)	FFH
	External interrupt mode register (INTM0)	00H
	Key return mode register (KRM00)	00H

Note The status after reset in read mode differs from that in write mode. For details, refer to **CHAPTER 16 SWAPPING**.

CHAPTER 20 μPD78F9835

The μPD78F9835 is available as the flash memory version of the μPD789835 Subseries.

The μPD78F9835 is a version with the internal ROM of the μPD789832, 789833, 789834, 789835 replaced with flash memory. The differences between the μPD78F9835 and the mask ROM versions are shown in Table 20-1.

Table 20-1. Differences Between μPD78F9835 and Mask ROM Versions

Item \ Part Number		Flash Memory Version	Mask ROM Version			
		μPD78F9835	μPD789832	μPD789833	μPD789834	μPD789835
Internal memory	ROM	60 KB (flash memory)	24 KB	32 KB	48 KB	60 KB
	High-speed RAM	1024 bytes				
	Low-speed RAM	2240 bytes	1216 bytes		2240 bytes	
	LCD display RAM	288 bytes × 2				
IC0 pin		Not provided	Provided			
V_{PP} pin		Provided	Not provided			
Supply voltage (V_{DD})		3.0 to 3.6 V	1.8 to 3.6 V			
Electrical specifications		Refer to **CHAPTER 22 ELECTRICAL SPECIFICATIONS (TARGET)**.				

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.

20.1 Flash Memory Characteristics

Flash memory programming is performed by connecting a dedicated flash programmer (Flashpro III (part no. FL-PR3, PG-FP3)/Flashpro IV[Note] (part no. FL-PR4, PG-FP4)) to the target system with the flash memory mounted on the target system (on-board). A flash memory writing adapter (program adapter), which is a target board used exclusively for programming, is also provided.

Note Under development

Remark FL-PR3, FL-PR4, and the program adapter are products made by Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191).

Programming using flash memory has the following advantages.

- Software can be modified after the microcontroller is solder-mounted on the target system
- Distinguishing software facilities low-quantity, varied model production
- Easy data adjustment when starting mass production

20.1.1 Programming environment

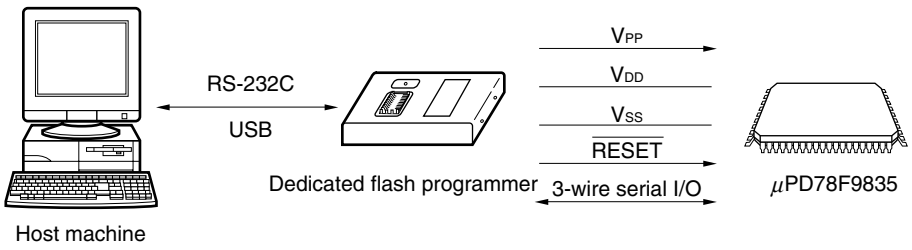
The following shows the environment required for μPD78F9835 flash memory programming.

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 manual for Flashpro III/Flashpro IV.

Remark USB is supported by Flashpro IV only.

Figure 20-1. Environment for Writing Program to Flash Memory



20.1.2 Communication mode

Use the communication mode shown in Table 20-2 to perform communication between the dedicated flash programmer and μPD78F9835.

Table 20-2. Communication Mode List

Communication Mode	TYPE Setting[Note 1]					Pins Used	Number of V_{PP} Pulses
	COMM PORT	SIO Clock	CPU Clock	Flash Clock	Multiple Rate		
3-wire serial I/O	SIO ch-0 (3 wired, sync.)	100 Hz to 1.25 MHz[Note 2]	Optional	1 to 5 MHz[Note 2]	1.0	$\overline{SCK10}$/P20 SO10/P21 SI1/P22	0

Notes 1. Selection items for TYPE settings on the dedicated flash programmer (Flashpro III (part no. FL-PR3, PG-FP3)/Flashpro IV (Part no. FL-PR4, PG-FP4)).

2. The possible setting range differs depending on the voltage. For details, refer to **CHAPTER 22 ELECTRICAL SPECIFICATIONS (TARGET)**.

Caution Be sure to select the communication mode according to the number of V_{PP} pulses shown in Table 20-2.

Figure 20-2. Communication Mode Selection Format

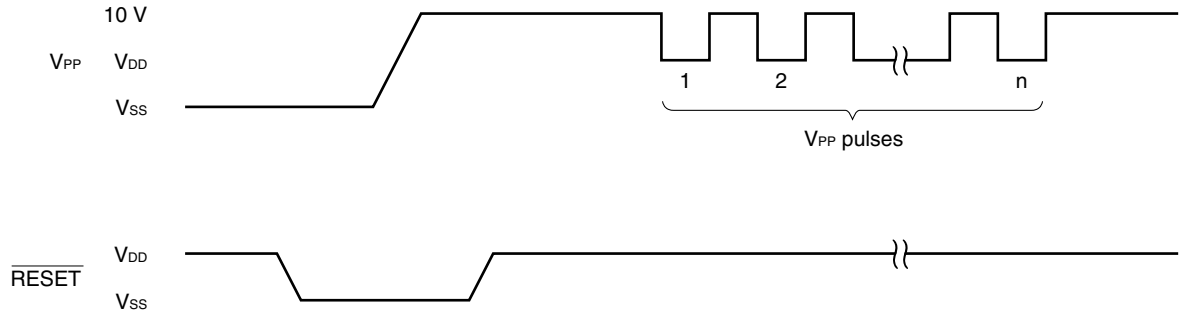
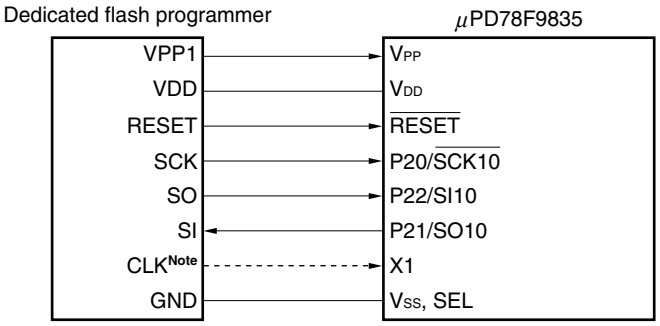


Figure 20-3. Example of Connection with Dedicated Flash Programmer



Note Connect this pin when the system clock is supplied by Flashpro III. If an oscillator is already connected to the X1 pin, the CLK pin does not need to be connected.

Caution The V_{DD} pin, if already connected to the power supply, must be connected to the VDD pin of the dedicated flash programmer. Before using the power supply connected to the V_{DD} pin, supply voltage before starting programming.

If Flashpro III (part no. FL-PR3, PG-FP3)/Flashpro IV is used as a dedicated flash programmer, the following signals are generated for the μPD78F9835. For details, refer to the manual of Flashpro III/Flashpro IV.

Table 20-3. Pin Connection List

Signal Name	I/O	Pin Function	Pin Name	3-Wire Serial I/O
VPP1	Output	Write voltage	V_{PP}	◎
VPP2	–	–		×
VDD	I/O	V_{DD} voltage generation/voltage monitoring	V_{DD}	◎Note
GND	–	Ground	V_{SS}, SEL	◎
CLK	Output	Clock output	X1	○
RESET	Output	Reset signal	$\overline{\text{RESET}}$	◎
SI	Input	Reception signal	P21/SO10	◎
SO	Output	Transmit signal	P22/SI10	◎
SCK	Output	Transfer clock	P20/$\overline{\text{SCK10}}$	◎
HS	–	–	–	×

Note V_{DD} voltage must be supplied before programming is started.

Remark ◎: Pin must be connected.
○: If the signal is supplied on the target board, pin need not be connected.
×: Pin need not be connected.

20.1.3 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.

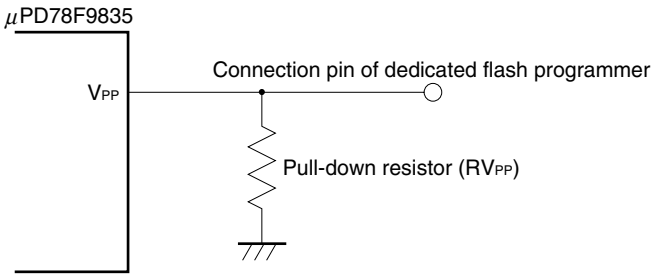
<V_{PP} pin>

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 20-4. V_{PP} Pin Connection Example



<Serial interface pin>

The following shows the pins used by the serial interface.

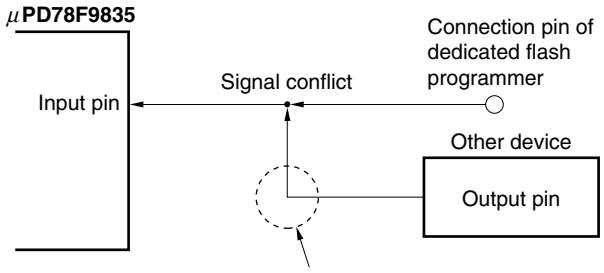
Serial Interface	Pins Used
3-wire serial I/O	$\overline{SCK10}$/P20, SO10/P21, SI10/P22

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 20-5. Signal Conflict (Input Pin of Serial Interface)

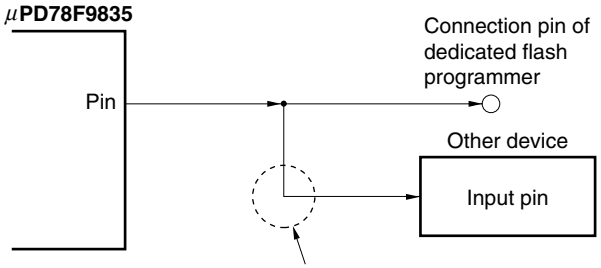


In the flash memory programming mode, the signal output by another device and the signal sent by the dedicated flash programmer conflict, therefore, isolate the signal of the other device.

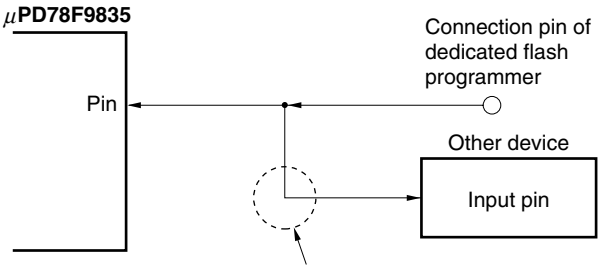
(2) Abnormal operation of other device

If the dedicated flash programmer (output or input) is connected to a serial interface pin (input or output) that is connected to another device (input), a signal is output to the device, and this may cause an abnormal operation. To prevent this abnormal operation, isolate the connection with the other device or set so that the input signals to the other device are ignored.

Figure 20-6. Abnormal Operation of Other Device



If the signal output by the μPD78F9835 affects another device in the flash memory programming mode, isolate the signals of the other device.



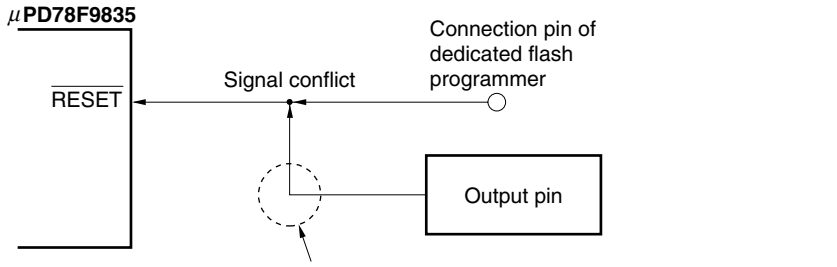
If the signal output by the dedicated flash programmer affects another device in the flash memory programming mode, isolate the signals of the other device.

<$\overline{\text{RESET}}$ 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 20-7. Signal Conflict ($\overline{\text{RESET}}$ Pin)



<Port pins>

When the μPD78F9468 enters the flash memory programming mode, all the pins other than those that communicate in flash memory programming are in the same status as immediately after reset.

If the external device does not recognize initial statuses such as the output high impedance status, therefore, connect the external device to V_{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. Operation of sub-clocks conforms that in the normal operation mode.

<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.

<Other pins>

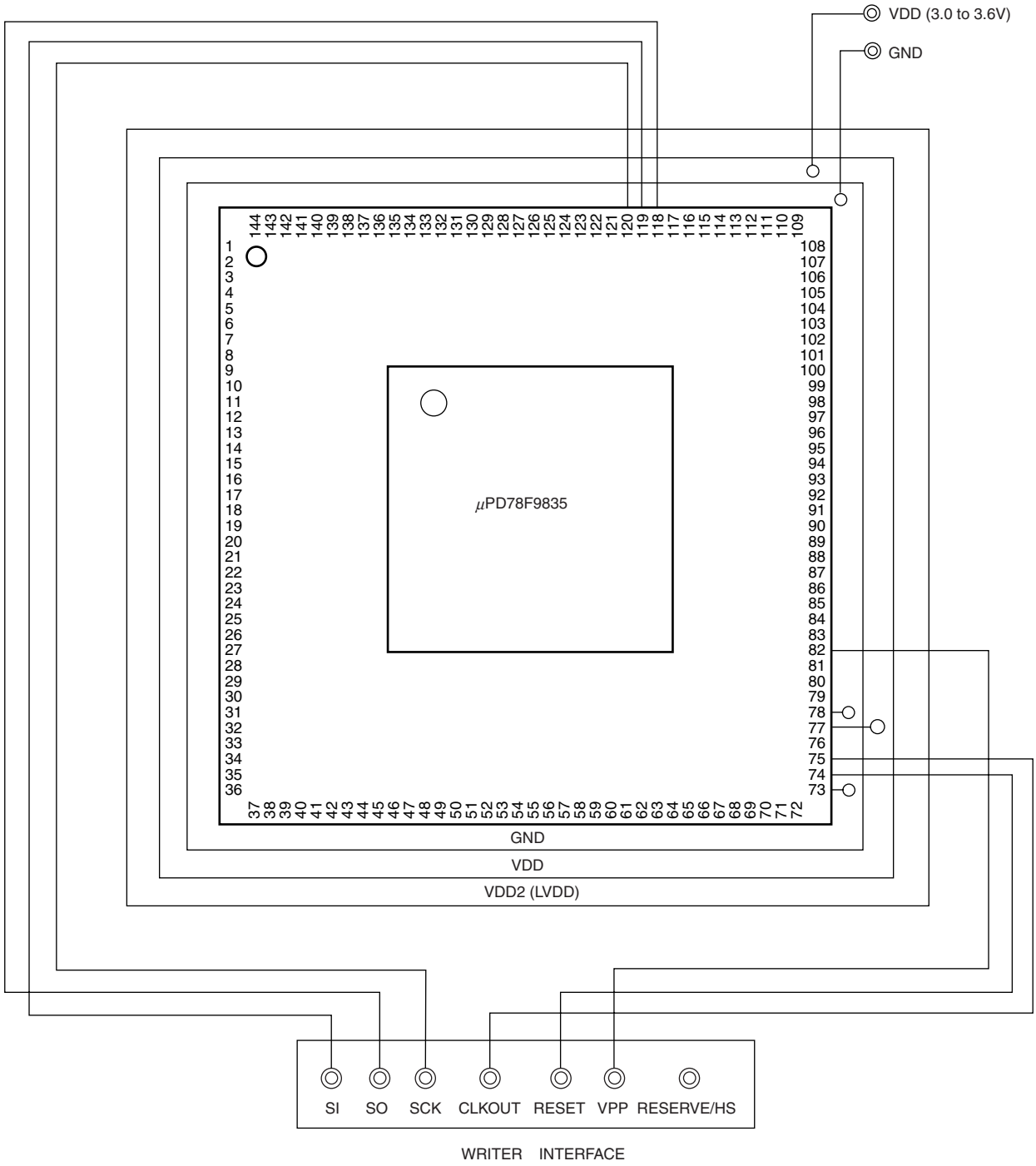
Connect the SEL pin to V_{SS}.

Process the other pins (LCD0 to LCD87, V_{LC0} to V_{LC4}, V_{ROUT0}, and CAP0 to CAP3) in the same manner as in the normal operation mode.

20.1.4 Connection on flash memory writing adapter

The following shows an example of the recommended connection when using the flash memory writing adapter.

Figure 20-8. Wiring Example of Flash Memory Writing Adapter Using 3-Wire Serial I/O Mode



CHAPTER 21 INSTRUCTION SET

This chapter lists the instruction set of the μPD789835 Subseries. For details of the operation and machine language (instruction code) of each instruction, refer to **78K/0S Series Instructions User's Manual (U11047E)**.

21.1 Operation

21.1.1 Operand identifiers and description methods

Operands are described in the "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 key words and are described as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: Absolute address specification
- $: Relative address specification
- [ ]: Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, $, and [ ] symbols.

For the 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 21-1. Operand Identifiers and Description Methods

Identifier	Description Method
r rp sfr	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special-function register symbol
saddr saddrp	FE20H to FF1FH Immediate data or label FE20H to FF1FH Immediate data or label (even addresses only)
addr16 addr5	0000H to FFFFH Immediate data or label (only even addresses for 16-bit data transfer instructions) 0040H to 007FH Immediate data or label (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 3-4 Special Function Registers** for symbols of special function registers.

21.1.2 Description of “Operation” column

A:	A register; 8-bit accumulator
X:	X register
B:	B register
C:	C register
D:	D register
E:	E register
H:	H register
L:	L register
AX:	AX register pair; 16-bit accumulator
BC:	BC register pair
DE:	DE register pair
HL:	HL register pair
PC:	Program counter
SP:	Stack pointer
PSW:	Program status word
CY:	Carry flag
AC:	Auxiliary carry flag
Z:	Zero flag
IE:	Interrupt request enable flag
NMIS:	Flag indicating non-maskable interrupt servicing in progress
( ):	Memory contents indicated by address or register contents in parenthesis
X_H, X_L:	Higher 8 bits and lower 8 bits of 16-bit register
∧:	Logical product (AND)
∨:	Logical sum (OR)
$\veebar$:	Exclusive logical sum (exclusive OR)
$\overline{\ }$:	Inverted data
addr16:	16-bit immediate data or label
jdisp8:	Signed 8-bit data (displacement value)

21.1.3 Description of “Flag” column

(Blank):	Unchanged
0:	Cleared to 0
1:	Set to 1
x:	Set/cleared according to the result
R:	Previously saved value is restored

21.2 Operation List

Mnemonic	Operands	Byte	Clock	Operation	Flag
					Z AC CY
MOV	r, #byte	3	6	r ← byte	
	saddr, #byte	3	6	(saddr) ← byte	
	sfr, #byte	3	6	sfr ← byte	
	A, r Note 1	2	4	A ← r	
	r, A Note 1	2	4	r ← A	
	A, saddr	2	4	A ← (saddr)	
	saddr, A	2	4	(saddr) ← A	
	A, sfr	2	4	A ← sfr	
	sfr, A	2	4	sfr ← A	
	A, !addr16	3	8	A ← (addr16)	
	!addr16, A	3	8	(addr16) ← A	
	PSW, #byte	3	6	PSW ← byte	x x x
	A, PSW	2	4	A ← PSW	
	PSW, A	2	4	PSW ← A	x x x
	A, [DE]	1	6	A ← (DE)	
	[DE], A	1	6	(DE) ← A	
	A, [HL]	1	6	A ← (HL)	
	[HL], A	1	6	(HL) ← A	
	A, [HL+byte]	2	6	A ← (HL + byte)	
	[HL+byte], A	2	6	(HL + byte) ← A	
XCH	A, X	1	4	A ↔ X	
	A, r Note 2	2	6	A ↔ r	
	A, saddr	2	6	A ↔ (saddr)	
	A, sfr	2	6	A ↔ sfr	
	A, [DE]	1	8	A ↔ (DE)	
	A, [HL]	1	8	A ↔ (HL)	
	A, [HL+byte]	2	8	A ↔ (HL + byte)	

Notes 1. Except r = A.

2. Except r = A, X.

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operands	Byte	Clock	Operation	Flag
					Z AC CY
MOVW	rp, #word	3	6	rp ← word	
	AX, saddrp	2	6	AX ← (saddrp)	
	saddrp, AX	2	8	(saddrp) ← AX	
	AX, rp Note	1	4	AX ← rp	
	rp, AX Note	1	4	rp ← AX	
XCHW	AX, rp Note	1	8	AX ↔ rp	
ADD	A, #byte	2	4	A, CY ← A + byte	x x x
	saddr, #byte	3	6	(saddr), CY ← (saddr) + byte	x x x
	A, r	2	4	A, CY ← A + r	x x x
	A, saddr	2	4	A, CY ← A + (saddr)	x x x
	A, !addr16	3	8	A, CY ← A + (addr16)	x x x
	A, [HL]	1	6	A, CY ← A + (HL)	x x x
	A, [HL+byte]	2	6	A, CY ← A + (HL + byte)	x x x
ADDC	A, #byte	2	4	A, CY ← A + byte + CY	x x x
	saddr, #byte	3	6	(saddr), CY ← (saddr) + byte + CY	x x x
	A, r	2	4	A, CY ← A + r + CY	x x x
	A, saddr	2	4	A, CY ← A + (saddr) + CY	x x x
	A, !addr16	3	8	A, CY ← A + (addr16) + CY	x x x
	A, [HL]	1	6	A, CY ← A + (HL) + CY	x x x
	A, [HL+byte]	2	6	A, CY ← A + (HL + byte) + CY	x x x
SUB	A, #byte	2	4	A, CY ← A – byte	x x x
	saddr, #byte	3	6	(saddr), CY ← (saddr) – byte	x x x
	A, r	2	4	A, CY ← A – r	x x x
	A, saddr	2	4	A, CY ← A – (saddr)	x x x
	A, !addr16	3	8	A, CY ← A – (addr16)	x x x
	A, [HL]	1	6	A, CY ← A – (HL)	x x x
	A, [HL+byte]	2	6	A, CY ← A – (HL + byte)	x x x

Note Only when rp = BC, DE, or HL.

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operands	Byte	Clock	Operation	Flag		
					Z	AC	CY
SUBC	A, #byte	2	4	A, CY ← A – byte – CY	x	x	x
	saddr, #byte	3	6	(saddr), CY ← (saddr) – byte – CY	x	x	x
	A, r	2	4	A, CY ← A – r – CY	x	x	x
	A, saddr	2	4	A, CY ← A – (saddr) – CY	x	x	x
	A, !addr16	3	8	A, CY ← A – (addr16) – CY	x	x	x
	A, [HL]	1	6	A, CY ← A – (HL) – CY	x	x	x
	A, [HL+byte]	2	6	A, CY ← A – (HL + byte) – CY	x	x	x
AND	A, #byte	2	4	A ← A ∧ byte	x		
	saddr, #byte	3	6	(saddr) ← (saddr) ∧ byte	x		
	A, r	2	4	A ← A ∧ r	x		
	A, saddr	2	4	A ← A ∧ (saddr)	x		
	A, !addr16	3	8	A ← A ∧ (addr16)	x		
	A, [HL]	1	6	A ← A ∧ (HL)	x		
	A, [HL+byte]	2	6	A ← A ∧ (HL + byte)	x		
OR	A, #byte	2	4	A ← A ∨ byte	x		
	saddr, #byte	3	6	(saddr) ← (saddr) ∨ byte	x		
	A, r	2	4	A ← A ∨ r	x		
	A, saddr	2	4	A ← A ∨ (saddr)	x		
	A, !addr16	3	8	A ← A ∨ (addr16)	x		
	A, [HL]	1	6	A ← A ∨ (HL)	x		
	A, [HL+byte]	2	6	A ← A ∨ (HL + byte)	x		
XOR	A, #byte	2	4	A ← A ∀ byte	x		
	saddr, #byte	3	6	(saddr) ← (saddr) ∀ byte	x		
	A, r	2	4	A ← A ∀ r	x		
	A, saddr	2	4	A ← A ∀ (saddr)	x		
	A, !addr16	3	8	A ← A ∀ (addr16)	x		
	A, [HL]	1	6	A ← A ∀ (HL)	x		
	A, [HL+byte]	2	6	A ← A ∀ (HL + byte)	x		

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operands	Byte	Clock	Operation	Flag
					Z AC CY
CMP	A, #byte	2	4	A – byte	x x x
	saddr, #byte	3	6	(saddr) – byte	x x x
	A, r	2	4	A – r	x x x
	A, saddr	2	4	A – (saddr)	x x x
	A, !addr16	3	8	A – (addr16)	x x x
	A, [HL]	1	6	A – (HL)	x x x
	A, [HL+byte]	2	6	A – (HL + byte)	x x x
ADDW	AX, #word	3	6	AX, CY ← AX + word	x x x
SUBW	AX, #word	3	6	AX, CY ← AX – word	x x x
CMPW	AX, #word	3	6	AX – word	x x x
INC	r	2	4	r ← r + 1	x x
	saddr	2	4	(saddr) ← (saddr) + 1	x x
DEC	r	2	4	r ← r – 1	x x
	saddr	2	4	(saddr) ← (saddr) – 1	x x
INCW	rp	1	4	rp ← rp + 1	
DECW	rp	1	4	rp ← rp – 1	
ROR	A, 1	1	2	(CY, $A_7 \leftarrow A_0$, $A_{m-1} \leftarrow A_m$) × 1	x
ROL	A, 1	1	2	(CY, $A_0 \leftarrow A_7$, $A_{m+1} \leftarrow A_m$) × 1	x
RORC	A, 1	1	2	(CY ← A_0, $A_7 \leftarrow$ CY, $A_{m-1} \leftarrow A_m$) × 1	x
ROLC	A, 1	1	2	(CY ← A_7, $A_0 \leftarrow$ CY, $A_{m+1} \leftarrow A_m$) × 1	x
SET1	saddr.bit	3	6	(saddr.bit) ← 1	
	sfr.bit	3	6	sfr.bit ← 1	
	A.bit	2	4	A.bit ← 1	
	PSW.bit	3	6	PSW.bit ← 1	x x x
	[HL].bit	2	10	(HL).bit ← 1	
CLR1	saddr.bit	3	6	(saddr.bit) ← 0	
	sfr.bit	3	6	sfr.bit ← 0	
	A.bit	2	4	A.bit ← 0	
	PSW.bit	3	6	PSW.bit ← 0	x x x
	[HL].bit	2	10	(HL).bit ← 0	
SET1	CY	1	2	CY ← 1	1
CLR1	CY	1	2	CY ← 0	0
NOT1	CY	1	2	CY ← $\overline{CY}$	x

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operands	Byte	Clock	Operation	Flag
					Z AC CY
CALL	!addr16	3	6	$(SP-1) \leftarrow (PC+3)_H$, $(SP-2) \leftarrow (PC+3)_L$, $PC \leftarrow$ addr16, $SP \leftarrow SP-2$	
CALLT	[addr5]	1	8	$(SP-1) \leftarrow (PC+1)_H$, $(SP-2) \leftarrow (PC+1)_L$, $PC_H \leftarrow$ (00000000, addr5 + 1), $PC_L \leftarrow$ (00000000, addr5), $SP \leftarrow SP-2$	
RET		1	6	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, $SP \leftarrow SP+2$	
RETI		1	8	$PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$, $PSW \leftarrow (SP+2)$, $SP \leftarrow SP+3$, NMIS $\leftarrow 0$	R R R
PUSH	PSW	1	2	$(SP-1) \leftarrow PSW$, $SP \leftarrow SP-1$	
	rp	1	4	$(SP-1) \leftarrow rp_H$, $(SP-2) \leftarrow rp_L$, $SP \leftarrow SP-2$	
POP	PSW	1	4	$PSW \leftarrow (SP)$, $SP \leftarrow SP+1$	R R R
	rp	1	6	$rp_H \leftarrow (SP+1)$, $rp_L \leftarrow (SP)$, $SP \leftarrow SP+2$	
MOVW	SP, AX	2	8	$SP \leftarrow AX$	
	AX, SP	2	6	$AX \leftarrow SP$	
BR	!addr16	3	6	PC ← addr16	
	$addr16	2	6	$PC \leftarrow PC+2+$ jdisp8	
	AX	1	6	$PC_H \leftarrow A$, $PC_L \leftarrow X$	
BC	$saddr16	2	6	$PC \leftarrow PC+2+$ jdisp8 if CY = 1	
BNC	$saddr16	2	6	$PC \leftarrow PC+2+$ jdisp8 if CY = 0	
BZ	$saddr16	2	6	$PC \leftarrow PC+2+$ jdisp8 if Z = 1	
BNZ	$saddr16	2	6	$PC \leftarrow PC+2+$ jdisp8 if Z = 0	
BT	saddr.bit, $addr16	4	10	$PC \leftarrow PC+4+$ jdisp8 if (saddr.bit) = 1	
	sfr.bit, $addr16	4	10	$PC \leftarrow PC+4+$ jdisp8 if sfr.bit = 1	
	A.bit, $addr16	3	8	$PC \leftarrow PC+3+$ jdisp8 if A.bit = 1	
	PSW.bit, $addr16	4	10	$PC \leftarrow PC+4+$ jdisp8 if PSW.bit = 1	
BF	saddr.bit, $addr16	4	10	$PC \leftarrow PC+4+$ jdisp8 if (saddr.bit) = 0	
	sfr.bit, $addr16	4	10	$PC \leftarrow PC+4+$ jdisp8 if sfr.bit = 0	
	A.bit, $addr16	3	8	$PC \leftarrow PC+3+$ jdisp8 if A.bit = 0	
	PSW.bit, $addr16	4	10	$PC \leftarrow PC+4+$ jdisp8 if PSW.bit = 0	
DBNZ	B, $addr16	2	6	$B \leftarrow B-1$, then $PC \leftarrow PC+2+$ jdisp8 if $B \neq 0$	
	C, $addr16	2	6	$C \leftarrow C-1$, then $PC \leftarrow PC+2+$ jdisp8 if $C \neq 0$	
	saddr, $addr16	3	8	$(saddr) \leftarrow (saddr)-1$, then $PC \leftarrow PC+3+$ jdisp8 if $(saddr) \neq 0$	
NOP		1	2	No Operation	
EI		3	6	IE ← 1 (Enable interrupt)	
DI		3	6	IE ← 0 (Disable interrupt)	
HALT		1	2	Set HALT mode	
STOP		1	2	Set STOP mode	

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

21.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, INC, DEC, ROR, ROL, RORC, ROLC, PUSH, POP, DBNZ

2nd Operand / 1st Operand	#byte	A	r	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL+byte]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV[Note] XCH[Note] ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											
[HL+byte]		MOV											

Note Except r = A.

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

1st Operand \ 2nd Operand	#word	AX	rp[Note]	saddrp	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	
rp	MOVW	MOVW[Note]				INCW DECW PUSH POP
saddrp		MOVW				
SP		MOVW				

Note Only when rp = BC, DE, or HL.

(3) Bit manipulation instructions

SET1, CLR1, NOT1, BT, BF

1st Operand \ 2nd Operand	$addr16	None
A.bit	BT BF	SET1 CLR1
sfr.bit	BT BF	SET1 CLR1
saddr.bit	BT BF	SET1 CLR1
PSW.bit	BT BF	SET1 CLR1
[HL].bit		SET1 CLR1
CY		SET1 CLR1 NOT1

(4) Call instructions/branch instructions

CALL, CALLT, BR, BC, BNC, BZ, BNZ, DBNZ

2nd Operand / 1st Operand	AX	!addr16	[addr5]	$addr16
Basic Instructions	BR	CALL BR	CALLT	BR BC BNC BZ BNZ
Compound Instructions				DBNZ

(5) Other instructions

RET, RETI, NOP, EI, DI, HALT, STOP

CHAPTER 22 ELECTRICAL SPECIFICATIONS (TARGET)

The following values are target specifications (target values) only, and may not be satisfied in mass-produced products.

Absolute Maximum Ratings (T_A = 25°C)

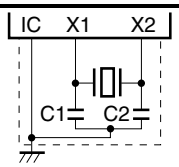
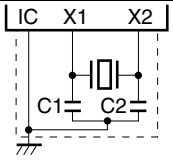
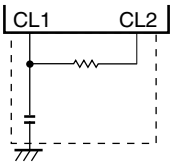
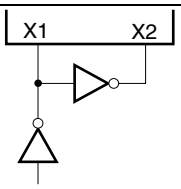
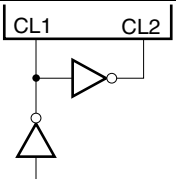
Parameter	Symbol	Conditions	Ratings	Unit
Power supply voltage	V_{DD}		–0.3 to +4.6	V
Input voltage	V_{I1}		–0.3 to V_{DD} + 0.3	V
Output voltage	V_{O1}	Other than below	–0.3 to V_{DD} + 0.3	V
	V_{O2}	LCD0 to LCD95, CAP0 to CAP3, V_{LC0} to V_{LC4}	–0.3 to +6.5	V
Output current, high	I_{OH}	1 pin	–10	mA
		TO40	–24	mA
		Total for all pins	–30	mA
Output current, low	I_{OL}	1 pin	30	mA
		Total for all pins	90	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
		μPD78F9835	–40 to +125	°C

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless otherwise specified, the characteristics of alternate-function pins are the same as those of port pins.

Main System Clock Oscillator Characteristics

(T_A = –40 to +85°C, V_{DD} = 1.8 to 3.6 V, (Mask ROM Version), V_{DD} = 3.0 to 3.6 V (μPD78F9835))

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		Oscillation frequency (f_X)[Note 1]		1.0		5.0	MHz
		Oscillation stabilization time[Note 2]	After V_{DD} reaches oscillation voltage range MIN.			4	ms
Crystal resonator		Oscillation frequency[Note 1]		1.0		5.0	MHz
		Oscillation stabilization time[Note 2]				10	ms
RC resonator		Oscillation frequency (f_{CC})		0.4		2.0	MHz
		Oscillation stabilization time[Note 2]	After V_{DD} reaches oscillation voltage range MIN.			10	μs
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
		CL1 input frequency (f_{CC})[Note 1]		1.0		5.0	MHz
		CL1 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.

Cautions 1. When using the main system clock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

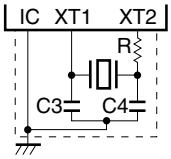
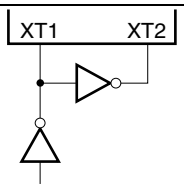
- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines.**
- **Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS}.**
-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

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

Remark For the resonator selection and oscillator constant, customers are required to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

Subsystem Clock Oscillator Characteristics

(T_A = –40 to +85°C, V_{DD} = 1.8 to 3.6 V (Mask ROM Version), V_{DD} = 3.0 to 3.6 V (μPD78F9835))

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]				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. Time required to stabilize oscillation after V_{DD} reaches oscillation voltage range MIN.

Cautions 1. When using the subsystem clock oscillator, wire as follows in the area enclosed by the broken lines in the above figure to avoid an adverse effect from wiring capacitance.

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

2. The subsystem clock oscillator is designed as a low-amplitude circuit for reducing current consumption, and is more prone to malfunction due to noise than the main system clock oscillator. Particular care is therefore required with the wiring method when the subsystem clock is used.

Remark For the resonator selection and oscillator constant, customers are required to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

DC Characteristics (1/2)

(T_A = –40 to +85°C, V_{DD} = 1.8 to 3.6 V (Mask ROM Version), V_{DD} = 3.0 to 3.6 V (μPD78F9835))

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low	I_{OL}	1 pin				10	mA
		All pins				80	mA
		P30/SG0, P34/SG4 (when buzzer mode 2 is selected) V_{OL} = V_{SS}	OE0 = 1, OEn = 0 (n = 1, 2, 3)		3		mA
			OE1 = 1, OEn = 0 (n = 0, 2, 3)		1.5		mA
			OE2 = 1, OEn = 0 (n = 0, 1, 3)		0.75		mA
			OE3 = 1, OEn = 0 (n = 0, 1, 2)		0.38		mA
Output current, high	I_{OH}	1 pin				–1	mA
		TO40/P27	V_{OH} = 1.0 V, V_{DD} = 3.0 V	–7	–15	–24	mA
		All pins				–15	mA
		P30/SG0, P34/SG4 (when buzzer mode 2 is selected) V_{OH} = V_{DD}	OE0 = 1, OEn = 0 (n = 1, 2, 3)		–3.0		mA
			OE1 = 1, OEn = 0 (n = 0, 2, 3)		–1.5		mA
			OE2 = 1, OEn = 0 (n = 0, 1, 3)		–0.75		mA
			OE3 = 1, OEn = 0 (n = 0, 1, 2)		–0.38		mA
Input voltage, high	V_{IH1}	P00 to P07, P10, P11, P21, P25, P27, P30 to P37, P60 to P62, P80 to P87		0.7V_{DD}		V_{DD}	V
	V_{IH2}	$\overline{\text{RESET}}$, P20, P22 to P24, P26		0.8V_{DD}		V_{DD}	V
	V_{IH3}	X1 (CL1), X2 (CL2)		V_{DD} – 0.1		V_{DD}	V
	V_{IH4}	XT1, XT2		V_{OH} – 0.1		V_{OH}	V
Input voltage, low	V_{IL1}	P00 to P07, P10, P11, P21, P25, P27, P30 to P37, P60 to P62, P80 to P87		0		0.3V_{DD}	V
	V_{IL2}	$\overline{\text{RESET}}$, P20, P22 to P24, P26		0		0.2V_{DD}	V
	V_{IL3}	X1 (CL1), X2 (CL2)		0		0.1	V
	V_{IL4}	XT1, XT2		0		0.1	V
Output voltage, high	V_{OH}	I_{OH} = –1 mA P00 to P07, P10, P11, P20 to P27, P30 to P37		V_{DD} – 1.0			V
Output voltage, low	V_{OL}	I_{OL} = 10 mA P00 to P07, P10, P11, P20 to P27, P30 to P37				1.0	V

Remarks 1. Unless otherwise specified, the characteristics of alternate-function pins are the same as those of port pins.

2. Pin names enclosed in parentheses are for RC oscillation.

DC Characteristics (2/2)

(T_A = –40 to +85°C, V_{DD} = 1.8 to 3.6 V (Mask ROM Version), V_{DD} = 3.0 to 3.6 V (μPD78F9835))

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	$V_{IN} = V_{DD}$	Other than X1 (CL1), X2 (CL2), XT1, XT2			3	μA
	I_{LIH2}		X1 (CL1), X2 (CL2)			20	μA
	I_{LIH3}	$V_{IN} = V_{OH}$	XT1, XT2			3	μA
Input leakage current, low	I_{LIL1}	V_{IN} = 0 V	Other than X1 (CL1), X2 (CL2), XT1, XT2			–3	μA
	I_{LIL2}		X1 (CL1), X2 (CL2), 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
Regulator voltage	V_{ROUT0}	Mask ROM version			1.4		V
		μPD78F9835			V_{DD}		V
Software pull-up resistor	R_1	V_{IN} = 0 V	P00 to P07, P10, P11, P20 to P27, P30 to P37	50	100	200	kΩ
Power supply current[Note 1] (mask ROM version)	I_{DD1}	5.0 MHz crystal oscillation operation mode	PCC = 00H			3.0	mA
			PCC = 02H			2.0	mA
	I_{DD2}	5.0 MHz crystal oscillation HALT mode				1.5	mA
	I_{DD3}	32.768 kHz crystal oscillation operation mode[Note 2]	VAON20 = 1		15		μA
			VAON20 = 0		10		μA
	I_{DD4}	STOP mode	When POC is operating		3.0		μA
			When POC is stopped		1.0		μA
	I_{DD5}	5.0 MHz crystal oscillation A/D operation mode	PCC = 00H			6.5	mA
			PCC = 02H			5.5	mA
Power supply current[Note 1] (μPD78F9835)	I_{DD1}	5.0 MHz crystal oscillation operation mode	PCC = 00H			3.5	mA
			PCC = 02H			2.5	mA
	I_{DD2}	5.0 MHz crystal oscillation HALT mode				2.0	mA
	I_{DD3}	32.768 kHz crystal oscillation operation mode[Note 2]			300		μA
	I_{DD4}	STOP mode			20		μA
	I_{DD5}	5.0 MHz crystal oscillation A/D operation mode	PCC = 00H			7.5	mA
			PCC = 02H			6.5	mA

Notes 1. The port current (including the current that flows to the on-chip pull-up resistors) is not included.

2. When the main system clock is stopped.

Remark Unless otherwise specified, the characteristics of alternate-function pins are the same as those of port pins.

AC Characteristics

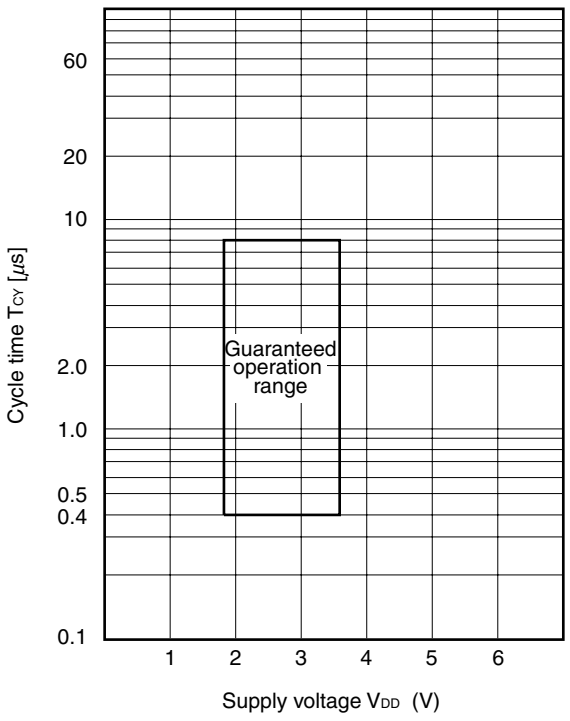
(1) Basic operation

(T_A = –40 to +85°C, V_{DD} = 1.8 to 3.6 V (mask ROM version), V_{DD} = 3.0 to 3.6 V (μPD78F9835))

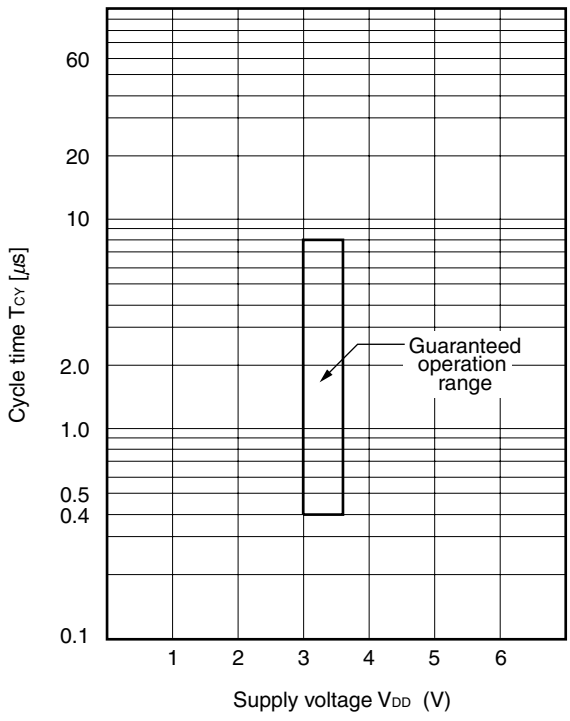
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Cycle time (minimum instruction execution time)	T_{CY}	Operating with main system clock	Crystal oscillation	0.4		8.0	μs
			RC oscillation	1.0		20	μs
		Operating with subsystem clock		114	122	125	μs
TI80 input frequency	f_{TI}			0		4	MHz
TI80 input high-/low-level width	t_{TIH}, t_{TIL}			0.1			μs
Interrupt input high-/low-level width	t_{INTH}, t_{INTL}	INTP0, INTP1		10			μs
Key return input low-level width	t_{KRL}	$\overline{KR00}$ to $\overline{KR07}$		10			μs
$\overline{RESET}$ low-level width	t_{RSL}			10			μs

T_{CY} vs. V_{DD} (Main system clock: ceramic/crystal oscillation)

(a) Mask ROM version



(b) μPD78F9835



(2) Serial interface (UART00)

(T_A = –40 to +85°C, V_{DD} = 1.8 to 3.6 V (mask ROM version), V_{DD} = 3.0 to 3.6 V (μPD78F9835))

Dedicated baud rate generator output

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		When fx = 5.0 MHz selected			115200	bps

(3) Serial interface (SIO10)

(T_A = –40 to +85°C, V_{DD} = 1.8 to 3.6 V (mask ROM version), V_{DD} = 3.0 to 3.6 V (μPD78F9835))

(a) 3-wire serial I/O mode (internal clock output)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCK10}$ cycle time	t_{KCY1}	V_{DD} = 2.7 to 3.6 V		800			ns
		V_{DD} = 1.8 to 3.6 V		3200			ns
$\overline{SCK10}$ high-/low-level width	t_{KH1}, t_{KL1}	V_{DD} = 2.7 to 3.6 V		$t_{KCY1}/2-50$			ns
		V_{DD} = 1.8 to 3.6 V		$t_{KCY1}/2-150$			ns
SI10 setup time (to $\overline{SCK10}$↑)	t_{SIK1}	V_{DD} = 2.7 to 3.6 V		150			ns
		V_{DD} = 1.8 to 3.6 V		500			ns
SI10 hold time (from $\overline{SCK10}$↑)	t_{KSI1}	V_{DD} = 2.7 to 3.6 V		400			ns
		V_{DD} = 1.8 to 3.6 V		600			ns
SO10 output delay time from $\overline{SCK10}$↓	t_{KSO1}	R = 1 kΩ, C = 100 pF[Note]	V_{DD} = 2.7 to 3.6 V	0		250	ns
			V_{DD} = 1.8 to 3.6 V	0		1000	ns

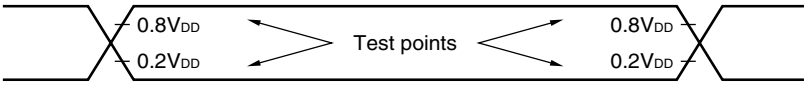
Note R and C are the load resistance and load capacitance of the SO10 output line.

(b) 3-wire serial I/O mode (external clock input)

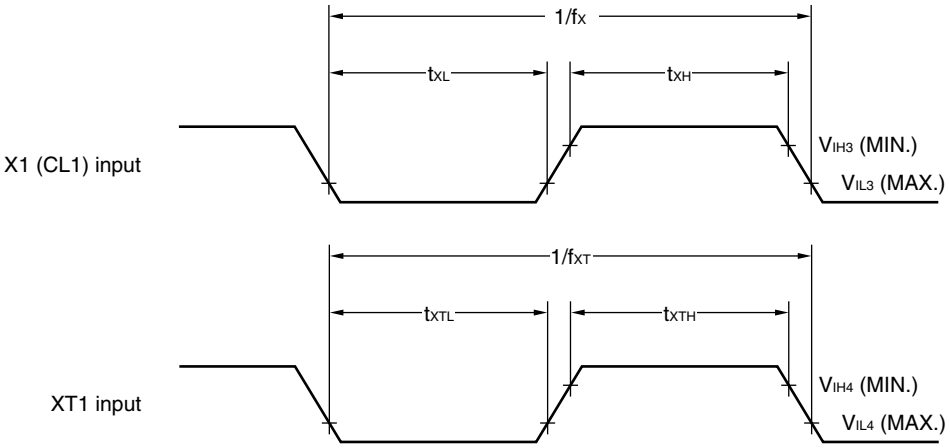
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCK10}$ cycle time	t_{KCY2}	V_{DD} = 2.7 to 3.6 V		800			ns
		V_{DD} = 1.8 to 3.6 V		3200			ns
$\overline{SCK10}$ high-/low-level width	t_{KH2}, t_{KL2}	V_{DD} = 2.7 to 3.6 V		400			ns
		V_{DD} = 1.8 to 3.6 V		1600			ns
SI10 setup time (to $\overline{SCK10}$↑)	t_{SIK2}	V_{DD} = 2.7 to 3.6 V		100			ns
		V_{DD} = 1.8 to 3.6 V		150			ns
SI10 hold time (from $\overline{SCK10}$↑)	t_{KSI2}	V_{DD} = 2.7 to 3.6 V		400			ns
		V_{DD} = 1.8 to 3.6 V		600			ns
SO10 output delay time from $\overline{SCK10}$↓	t_{KSO2}	R = 1 kΩ, C = 100 pF[Note]	V_{DD} = 2.7 to 3.6 V	0		300	ns
			V_{DD} = 1.8 to 3.6 V	0		1000	ns

Note R and C are the load resistance and load capacitance of the SO10 output line.

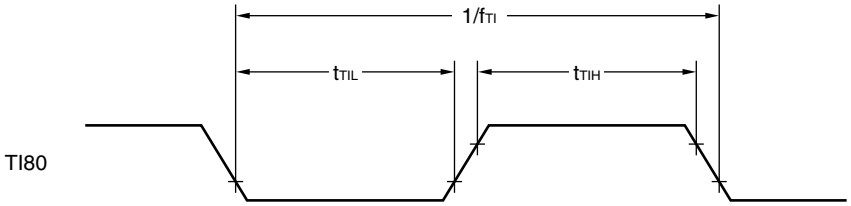
AC Timing Test Points (Excluding X1 and XT1 Inputs)



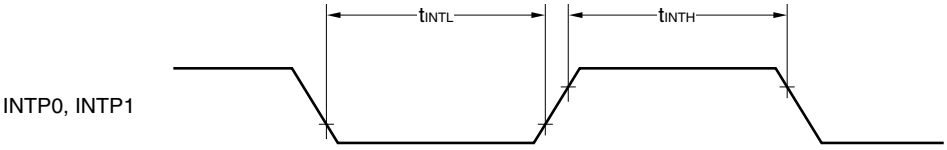
Clock Timing



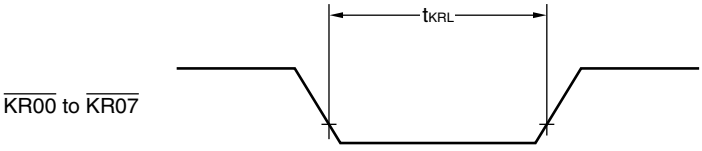
TI80 Timing



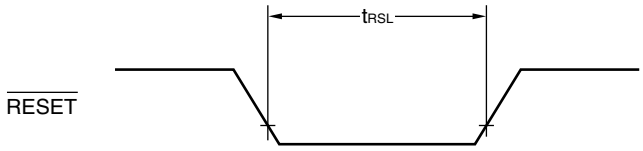
Interrupt Input Timing



Key Return Input Timing

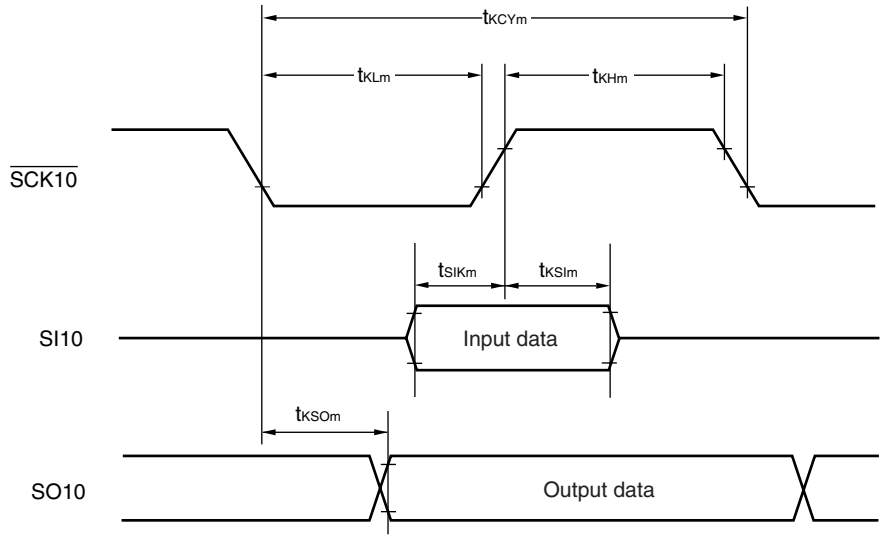


$\overline{\text{RESET}}$ Input Timing



Serial Transfer Timing

3-wire serial I/O mode:



Remark m = 1, 2

LCD Characteristics

(T_A = –40 to +85°C, V_{DD} = 1.8 to 3.6 V (Mask ROM Version), V_{DD} = 3.0 to 3.6 V (μPD78F9835))

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD output voltage differential (common)	V_{ODC}	Io = ±5 μA	0		±0.2	V
LCD output voltage differential (segment)	V_{ODS}	Io = ±5 μA	0		±0.2	V
Boost voltage set time	t_B	C = 0.47 μF, after LIPS20 is set		300		ms
LCD voltage	V_{LC0}	VLCD00 = 00H		4.0		V
		VLCD00 = 01H		4.1		V
		VLCD00 = 02H		4.2		V
		VLCD00 = 03H		4.3		V
		VLCD00 = 04H		4.4		V
		VLCD00 = 05H		4.5		V
		VLCD00 = 06H		4.6		V
		VLCD00 = 07H		4.7		V
		VLCD00 = 08H		4.8		V
		VLCD00 = 09H		4.9		V
		VLCD00 = 0AH		5.0		V
		VLCD00 = 0BH		5.1		V
		VLCD00 = 0CH		5.2		V
		VLCD00 = 0DH		5.3		V
		VLCD00 = 0EH		5.4		V
		VLCD00 = 0FH		5.5		V
	V_{LC1}			4/5 V_{LC0}		V
	V_{LC2}			3/5 V_{LC0}		V
	V_{LC3}			2/5 V_{LC0}		V
	V_{LC4}			1/5 V_{LC0}		V
Segment output voltage	V_{ODS}	Output level = V_{LC0}		V_{LC0}		V
		Output level = V_{LC2}		3/5 V_{LC0}		V
		Output level = V_{LC3}		2/5 V_{LC0}		V
Common signal output	V_{ODC}	Output level = V_{LC0}		V_{LC0}		V
		Output level = V_{LC1}		4/5 V_{LC0}		V
		Output level = V_{LC4}		1/5 V_{LC0}		V

8-Bit A/D Converter Characteristics

(T_A = –40 to +85°C, 2.2 V ≤ V_{DD} ≤ 3.6 V (Mask ROM Version), 3.0 V ≤ V_{DD} ≤ 3.6 V (μPD78F9835))

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			8	8	8	bit
Overall error[Note]					1.5	LSB
Conversion time	t_{CONV}		14			μs
Analog input voltage	V_{IAN}		0		V_{DD}	V

Note Excludes quantization error (±1/2 LSB).

Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics

(T_A = –40 to +85°C, V_{DD} = 1.8 to 3.6 V (Mask ROM Version), 3.0 V ≤ V_{DD} ≤ 3.6 V (μPD78F9835))

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Data retention power supply voltage	V_{DDDR}			1.8		3.6	V
Release signal set time	t_{SREL}			0			μs
Oscillation stabilization wait time[Note 1]	t_{WAIT}	Release by $\overline{RESET}$	Ceramic/crystal oscillation		$2^{15}/f_X$		s
			RC oscillation		$2^7/f_{CC}$		s
		Release by interrupt	Ceramic/crystal oscillation		**Note 2**		s
			RC oscillation		$2^7/f_{CC}$		s

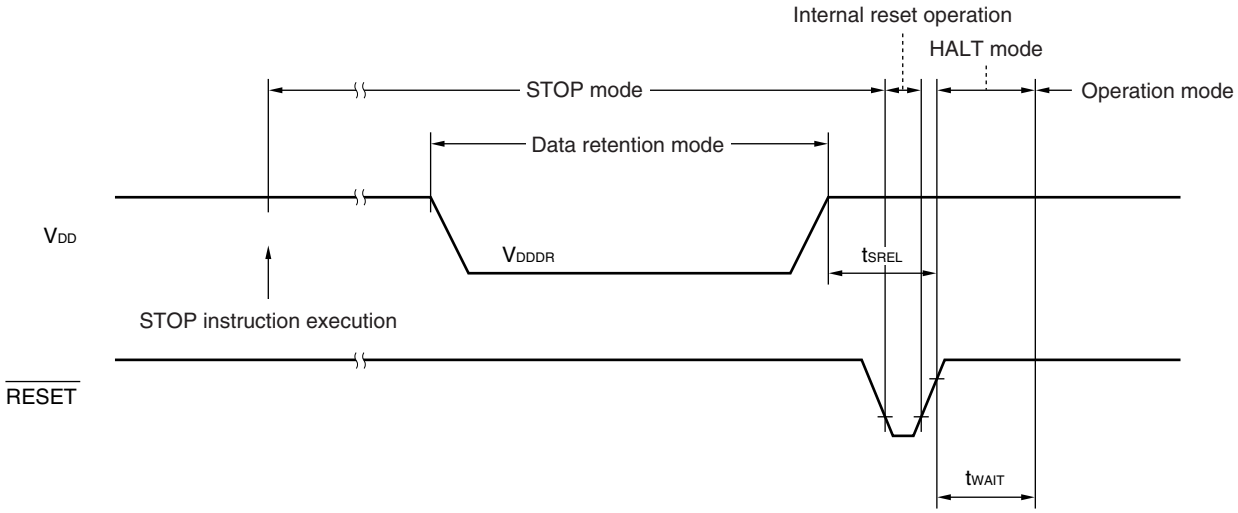
Notes 1. Use a resonator whose oscillation stabilizes within the oscillation stabilization wait time.

2. Selection of $2^{12}/f_X$, $2^{15}/f_X$, or $2^{17}/f_X$ is possible with bits 0 to 2 (OSTS0 to OSTS2) of the oscillation stabilization time selection register (OSTS). For details, see **18.1.2 Registers controlling standby function**.

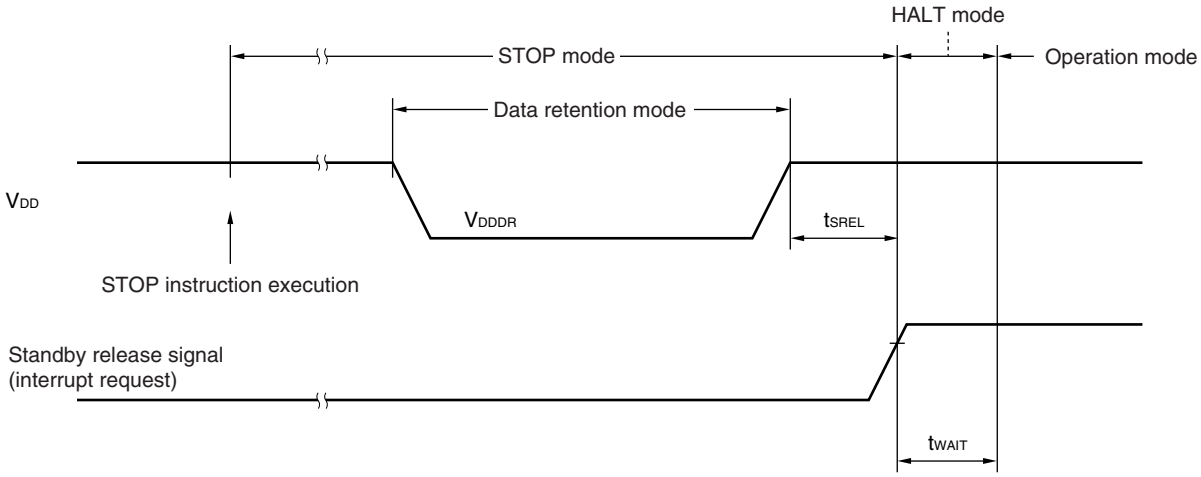
Remarks 1. f_X: Main system clock oscillation frequency (ceramic/crystal oscillation)

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

Data Retention Timing (STOP Mode Release by $\overline{RESET}$)



Data Retention Timing (Standby Release Signal: STOP Mode Release by Interrupt Request Signal)

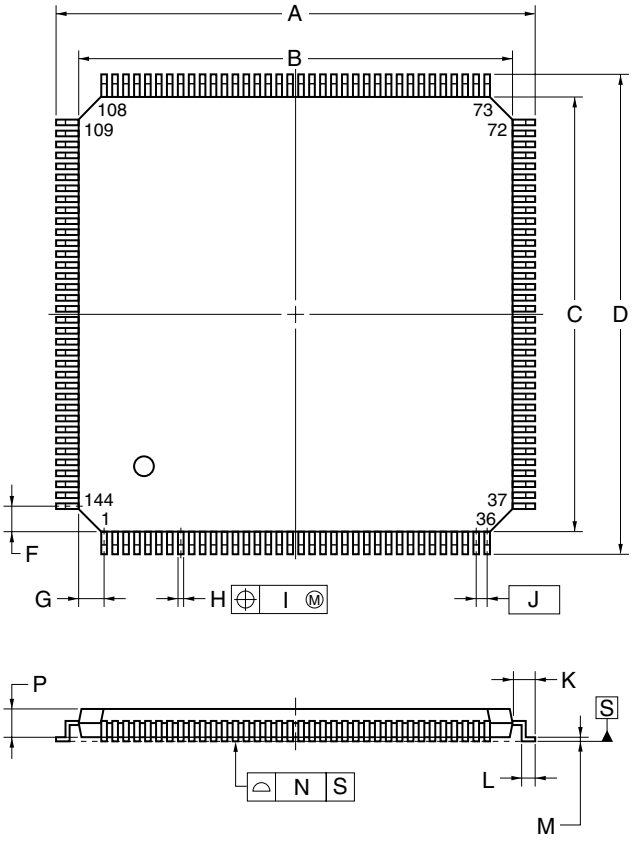


Flash Memory Writing and Erasing Characteristics (T_A = 10 to 40°C, V_{DD} = 3.0 to 3.6 V) (μPD78F9835 only)

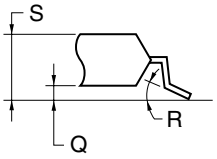
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Write/erase operating frequency	fx	3.0 V ≤ V_{DD} ≤ 3.6 V	1.0		5	MHz
Write current (V_{DD} pin)[Note]	I_{DDW}	When V_{PP} supply voltage = V_{PP1} (at 5.0 MHz operation)			7	mA
Write current (V_{PP} pin)[Note]	I_{PPW}	When V_{PP} supply voltage = V_{PP1}			13	mA
Erase current (V_{DD} pin)[Note]	I_{DDE}	When V_{PP} supply voltage = V_{PP1} (at 5.0 MHz operation)			7	mA
Erase current (V_{PP} pin)[Note]	I_{PPE}	When V_{PP} supply voltage = V_{PP1}			100	mA
Unit erase time	t_{er}		0.5	1	1	s
Total erase time	t_{era}				20	s
Number of overwrites		Erase and write is considered as 1 cycle			20	Times
V_{PP} supply voltage	V_{PP0}	During normal operation	0		0.2V_{DD}	V
	V_{PP1}	During flash memory programming	9.7	10.0	10.3	V

Note Excludes current flowing through ports (including on-chip pull-up resistors)

144-PIN PLASTIC LQFP (FINE PITCH) (20x20)



detail of lead end



NOTE

Each lead centerline is located within 0.08 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	22.0±0.2
B	20.0±0.2
C	20.0±0.2
D	22.0±0.2
F	1.25
G	1.25
H	0.22±0.05
I	0.08
J	0.5 (T.P.)
K	1.0±0.2
L	0.5±0.2
M	$0.17^{+0.03}_{-0.07}$
N	0.08
P	1.4
Q	0.10±0.05
R	$3^{\circ}{}^{+4^{\circ}}_{-3^{\circ}}$
S	1.5±0.1

S144GJ-50-UEN

APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for development of systems using the μPD789835 Subseries. Figure A-1 shows the development tools.

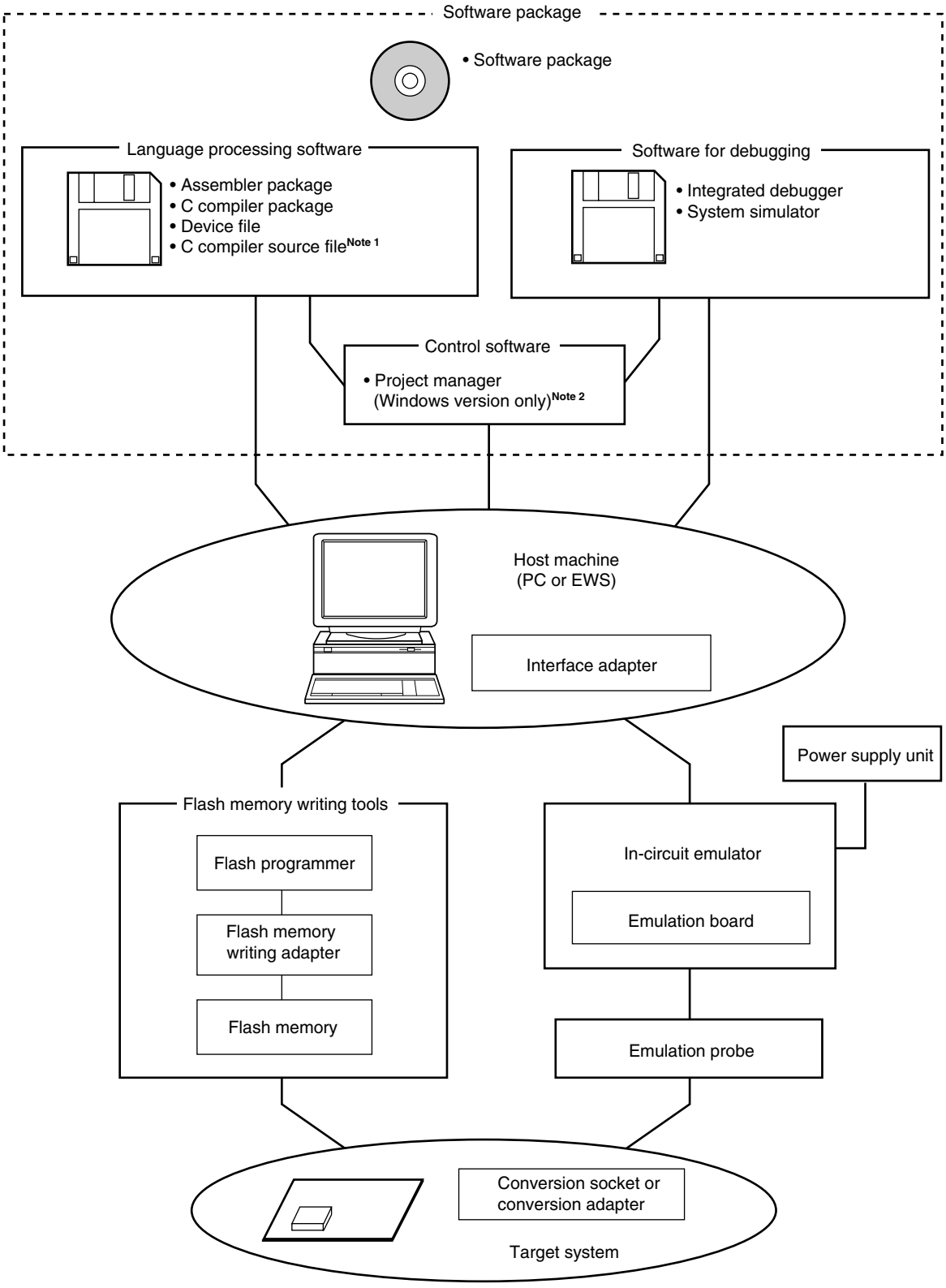
- Support of PC98-NX Series

  Unless specified otherwise, 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 compiler 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, ID78K0-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 (DF789835) (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 (DF789835) (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
DF789835[Note 1] Device file	File containing the information specific to the device. Used in combination with the RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S (sold separately).
	Part number: μS××××DF789835
CC78K0S-L[Note 2] C compiler source file	Source file of functions for generating object library included in C compiler package. Necessary for changing object library included in C compiler package according to customer's specifications. Since this is a source file, its working environment does not depend on any particular operating system.
	Part number: μS××××CC78K0S-L

Notes 1. DF789835 is a file that can be used commonly 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.4.1.1), Solaris™ (Rel.2.5.1)	

μS××××DF789835
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.4.1.1), 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 (Part No. FL-PR3, PG-FP3) Flashpro IV (Part No. FL-PR4, PG-FP4) Flash programmer	Dedicated flash programmer for microcontrollers incorporating flash memory
FA-144GJ-UEN Flash memory writing adapter	Adapter for writing to flash memory and connected to Flashpro III. FA-144GJ-UEN: for 144-pin plastic LQFP (GJ-UEN type)

Remark The FL-PR3, FL-PR4, and FA-144GJ-UEN are products made by Naito Densei Machida Mfg. Co., Ltd. (TEL +81-45-475-4191).

A.5 Debugging Tools (Hardware)

IE-78K0S-NS In-circuit emulator		In-circuit emulator for debugging hardware and software of application system using 78K/0S Series. Supports integrated debugger (ID78K0S-NS). Used in combination with AC adapter, emulation probe, and interface adapter for connecting the host machine.
IE-78K0S-NS-A In-circuit emulator		IE-78K0S-NS with added coverage function and enhanced the debug function, 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-789835-NS-EM1 Emulation board		Board for emulating peripheral hardware specific to device. Used in combination with in-circuit emulator.
SWEX-144SD-1 Emulation probe		Probe for connecting in-circuit emulator and target system. Used in combination with NQPACK144SD and YQSOCKET144SDF.
	NQPACK144SD Conversion connector	Conversion connector to connect SWEX-144SD-1 and target system board on which 144-pin plastic LQFP (GJ-UEN type) can be mounted
	YQSOCKET144SDF Conversion socket	Conversion socket to connect SWEX-144SD-1 and target system board on which 144-pin plastic LQFP (GJ-UEN type) can be mounted

Remark The SWEX-144SD-1, NQPACK144SD and YQSOCKET144SDF are products made by TOKYO ELETECH CORPORATION.

For further information, contact: Daimaru Kogyo, Ltd.
Tokyo Electronics Department (TEL +81-3-3820-7112)
Osaka Electronics Department (TEL +81-6-6244-6672)

A.6 Debugging Tools (Software)

ID78K0S-NS Integrated debugger	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 (DF789835) (sold separately).
	Part number: μS××××ID78K0S-NS
SM78K0S System simulator	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 (DF789835) (sold separately).
	Part number: μS××××SM78K0S
DF789835[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××××DF789835

Note DF789835 is a file that can be used commonly 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	

APPENDIX B REGISTER INDEX

B.1 Register Index (Alphabetic Order of Register Name)

B.2 Register Index (Alphabetic Order of Register Symbol)

NEC

Facsimile Message

From:

Name

Company

Tel. FAX

Address

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

Thank you for your kind support.

North America
NEC Electronics Inc.
Corporate Communications Dept.
Fax: +1-800-729-9288
+1-408-588-6130

Europe
NEC Electronics (Europe) GmbH
Market Communication Dept.
Fax: +49-211-6503-274

South America
NEC do Brasil S.A.
Fax: +55-11-6462-6829

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

Korea
NEC Electronics Hong Kong Ltd.
Seoul Branch
Fax: +82-2-528-4411

P.R. China
NEC Electronics Shanghai, Ltd.
Fax: +86-21-6841-1137

Taiwan
NEC Electronics Taiwan Ltd.
Fax: +886-2-2719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Japan
NEC Semiconductor Technical Hotline
Fax: +81- 44-435-9608

I would like to report the following error/make the following suggestion:

Document title: _____

Document number: _____ Page number: _____

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	❑	❑	❑	❑
Technical Accuracy	❑	❑	❑	❑
Organization	❑	❑	❑	❑