To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

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

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

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

User’s Manual

μPD789830 Subseries

8-Bit Single-Chip Microcontrollers

μPD789830
μPD78F9831

Document No. U13679EJ2V1UD00 (2nd edition)
Date Published August 2005 N CP(K)

[MEMO]

NOTES FOR CMOS DEVICES

① VOLTAGE APPLICATION WAVEFORM AT INPUT PIN

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

② HANDLING OF UNUSED INPUT PINS

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

③ PRECAUTION AGAINST ESD

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

④ STATUS BEFORE INITIALIZATION

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

⑤ POWER ON/OFF SEQUENCE

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

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

⑥ INPUT OF SIGNAL DURING POWER OFF STATE

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

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

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

- **The information in this document is current as of August, 2005. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".
  The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.
  "Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots.
  "Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).
  "Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

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

(Note)
(1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.
(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

M8E 02. 11-1

Regional Information

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

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

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

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

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

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

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

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

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

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

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

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

J05.6

Major Revisions in This Edition

Page	Description
U13679JJ1V0UD00 → U13679JJ2V0UD00	
p. 31	Addition of description on timer to **1.7 Overview of Functions**
p. 38	Addition of description on pin processing to **2.3.14 V_{PP}**
p. 41	Modification of **Table 2-1 Types of Pin I/O Circuits and Recommended Connection of Unused Pins (μPD789830)**
p. 42	Modification of **Table 2-2 Types of Pin I/O Circuits and Recommended Connection of Unused Pins (μPD78F9831)**
p. 87	Addition of **Note** to **5.3 (2) Suboscillation mode register (SCKM)**
p. 97	Addition of **Caution** to **6.2 (1) 16-bit compare register 40 (CR40)**
p. 99	Addition of **Caution** to **6.4.1 Operation as interval timer**
p. 105	Addition of description to **7.3 8-Bit Timer 00 Control Registers**
p. 117	Modification of **Caution** in **9.3 (2) Watchdog timer mode register (WDTM)**
p. 180	Overall modification of descriptions in **CHAPTER 16 μPD78F9831**
p. 198	Addition of **CHAPTER 18 ELECTRICAL SPECIFICATIONS**
p. 211	Addition of **CHAPTER 19 PACKAGE DRAWINGS**
p. 212	Addition of **CHAPTER 20 RECOMMENDED SOLDERING CONDITIONS**
p. 213	Overall modification of descriptions in **APPENDIX A DEVELOPMENT TOOLS**
p. 204	Addition of **APPENDIX C REVISION HISTORY**
p. 205 in 1st edition	Deletion of **APPENDIX B EMBEDDED SOFTWARE**
U13679JJ2V0UD00 → U13679JJ2V1UD00	
p. 21	Addition of lead-free products to **1.3 Ordering Information**
p. 212	Addition of lead-free products to **CHAPTER 20 RECOMMENDED SOLDERING CONDITIONS**

The mark ★ shows major revised points.

INTRODUCTION

Target Readers

This manual is intended for users who wish to understand the functions of the μPD789830 Subseries and to design and develop application systems and programs using these microcontrollers.

- μPD789830 Subseries: μPD789830 and μPD78F9831

Purpose

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

Organization

The μPD789830 Subseries User’s Manual is divided into two parts: this manual and instructions (common to the 78K/0S Series).

μPD789830 Subseries User's Manual (This manual)	78K/0S Series User's Manual Instructions
• Pin functions	• CPU functions
• Internal block functions	• Instruction set
• Interrupt functions	• Explanation of each instruction
• Other on-chip peripheral functions	
• Electrical specifications	

How to Read This Manual

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

To 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 with < >, its bit name is defined as a reserved word in the RA78K0S, and in the CC78K0S, 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 μPD789830 Subseries instruction functions in detail:
→ Refer to **78K/0S Series Instruction User's Manual (U11047E)**.

To know the electrical specifications of the μPD789830 Subseries
→ Refer to **CHAPTER 18 ELECTRICAL SPECIFICATIONS**.

Conventions

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

★ Related Documents

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

Documents Related to Devices

Document Name	Document No.
μPD789830 Subseries User's Manual	This manual
78K/0S Series Instruction User's Manual	U11047E

Documents Related to Development Software Tools (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 Specification	U15006E
ID78K Series Integrated Debugger Ver. 2.30 or Later	Operation (Windows Based)	U15185E
Project Manager Ver. 3.12 or Later (Windows Based)		U14610E

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

Document Name	Document No.
IE-78K0S-NS In-Circuit Emulator	U13549E
IE-78K0S-NS-A In-Circuit Emulator	U15207E
IE-789831-NS-EM1 Emulation Board	U14202E

Documents Related to Flash Memory Writing

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

Other Related Documents

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

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

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

CONTENTS

LIST OF FIGURES (1/4)

LIST OF FIGURES (2/4)

LIST OF FIGURES (3/4)

LIST OF FIGURES (4/4)

LIST OF TABLES (1/2)

LIST OF TABLES (2/2)

CHAPTER 1 GENERAL

1.1 Features

- ROM and RAM capacity

Item	Program Memory	Data Memory	
Product Name		RAM	LCD Display RAM
μPD789830	24 KB (mask ROM)	1 KB	80 KB
μPD78F9831	48 KB (flash memory)	2 KB	

- Minimum instruction execution time changeable from high speed (0.56 μs: Main system clock 3.58 MHz operation) to ultra-low speed (122 μs: Subsystem clock 32.768 kHz operation)
- I/O port
  - μPD789830: 30 (one N-ch open drain)
  - μPD78F9831: 38 (one N-ch open drain)
- Serial interface (UART00): 1 channel
- Timer: 4 channels
  - 16-bit timer: 1 channel
  - 8-bit timer: 1 channel
  - Watch timer: 1 channel
  - Watchdog timer: 1 channel
- Pulse output: clock output/buzzer output
- LCD controller/driver
  - Segment signal output: 40 lines MAX.
  - Common signal output: 16 lines MAX.
  - 1/5 bias mode
- Vectored interrupt source
  - μPD789830: 15
  - μPD78F9831: 17
- On-chip key return signal detector
- Supply voltage: V_{DD} = 2.7 to 5.5 V
- Supply format
  - μPD789830: 88-pin bare chip
  - μPD78F9831: 100-pin plastic LQFP (fine pitch)

1.2 Applications

Card readers, etc.

★

1.3 Ordering Information

Part Number	Supply Format	Internal ROM
μPD789830P-×××	88-pin bare chip	Mask ROM
μPD78F9831GC-8EU	100-pin plastic LQFP (fine pitch) (14 × 14)	Flash memory
μPD78F9831GC-8EU-A	100-pin plastic LQFP (fine pitch) (14 × 14)	Flash memory

Remarks 1. ××× indicates ROM code suffix.

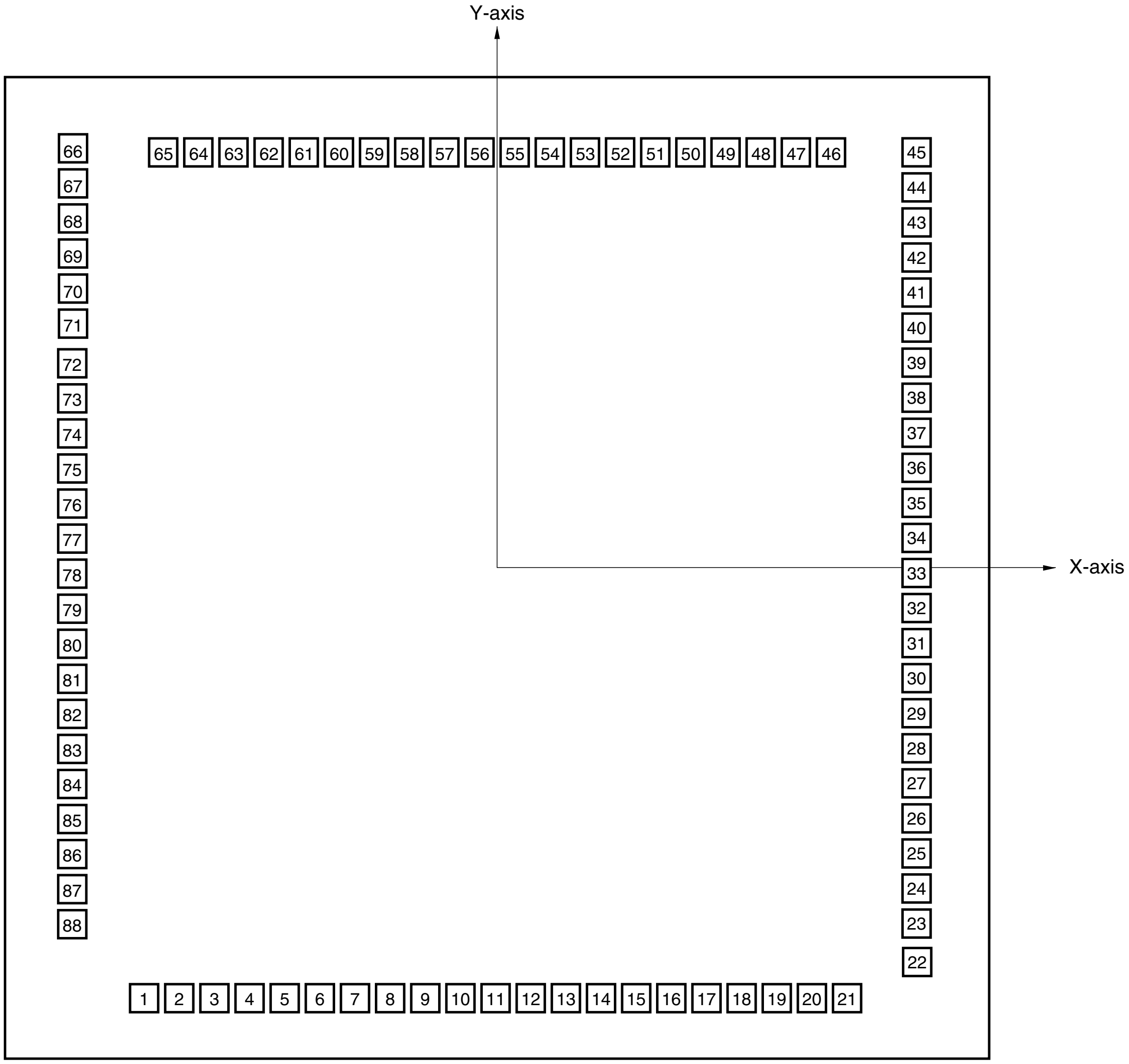
2. Products that have the part numbers suffixed by "-A" are lead-free products.

1.4 Pin Configuration (Top View)

(1) μPD789830

- **88-pin bare chip**
  μPD789830P-×××

Minimum pad pitch: 110.04 μm
Pad aperture: 80.04 μm



Pin Name

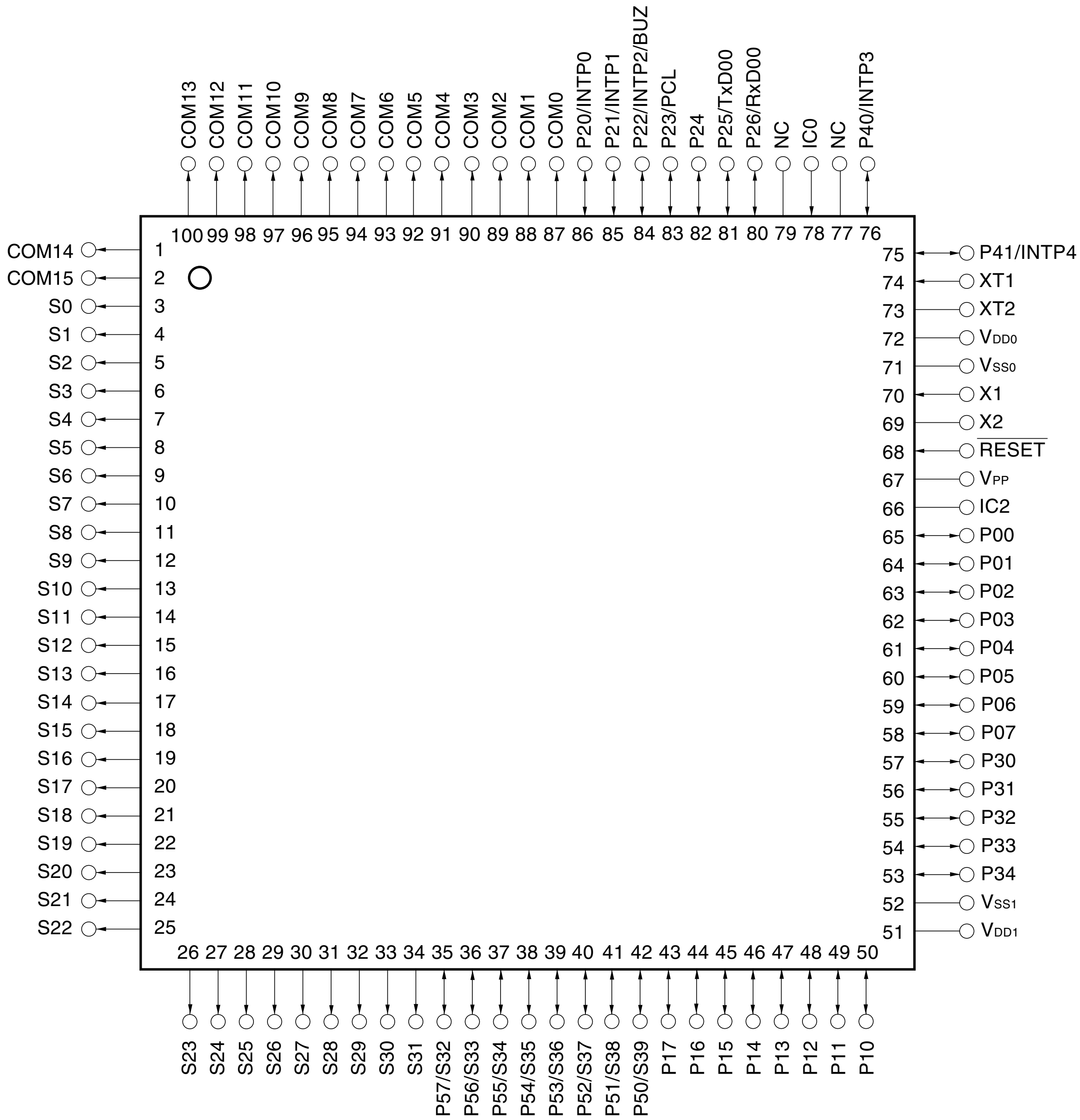
No.	Pin Name	No.	Pin Name	No.	Pin Name
1	COM14	31	S28	61	$\overline{\text{RESET}}$
2	COM15	32	S29	62	X2
3	S0	33	S30	63	X1
4	S1	34	S31	64	V_{SS0}
5	S2	35	P57/S32	65	V_{DD0}
6	S3	36	P56/S33	66	XT2
7	S4	37	P55/S34	67	XT1
8	S5	38	P54/S35	68	P26/RxD00
9	S6	39	P53/S36	69	P25/TxD00
10	S7	40	P52/S37	70	P24
11	S8	41	P51/S38	71	P23/PCL
12	S9	42	P50/S39	72	P22/INTP2/BUZ
13	S10	43	P11	73	P21/INTP1
14	S11	44	P10	74	P20/INTP0
15	S12	45	V_{DD1}	75	COM0
16	S13	46	V_{SS1}	76	COM1
17	S14	47	P34	77	COM2
18	S15	48	P33	78	COM3
19	S16	49	P32	79	COM4
20	S17	50	P31	80	COM5
21	S18	51	P30	81	COM6
22	S19	52	P07	82	COM7
23	S20	53	P06	83	COM8
24	S21	54	P05	84	COM9
25	S22	55	P04	85	COM10
26	S23	56	P03	86	COM11
27	S24	57	P02	87	COM12
28	S25	58	P01	88	COM13
29	S26	59	P00		
30	S27	60	IC0		

Remark For details of pin coordinates, contact an NEC sales representative.

BUZ:	Buzzer clock	PCL:	Programming clock
COM0 to COM15:	RC oscillator	$\overline{\text{RESET}}$:	Reset
IC0:	Internally connected	RxD00:	Receive data
INTP0 to INTP2:	Interrupt from peripherals	S0 to S39:	Segment output
P00 to P07:	Port 0	TxD00:	Transmit data
P10, P11:	Port 1	V_{DD0}, V_{DD1}:	Power supply
P20 to P26:	Port 2	V_{SS0}, V_{SS1}:	Ground
P30 to P34:	Port 3	X1, X2:	Crystal (Main system clock)
P50 to P57:	Port 5	XT1, XT2:	Crystal (Subsystem clock)

(2) μPD78F9831

- **100-pin plastic LQFP (fine pitch) (14 × 14)**
  μPD78F9831GC-8EU
  μPD78F9831GC-8EU-A



Cautions 1. Connect the V_{PP} pin directly to the V_{SS0} or V_{SS1} pin in normal operation mode.

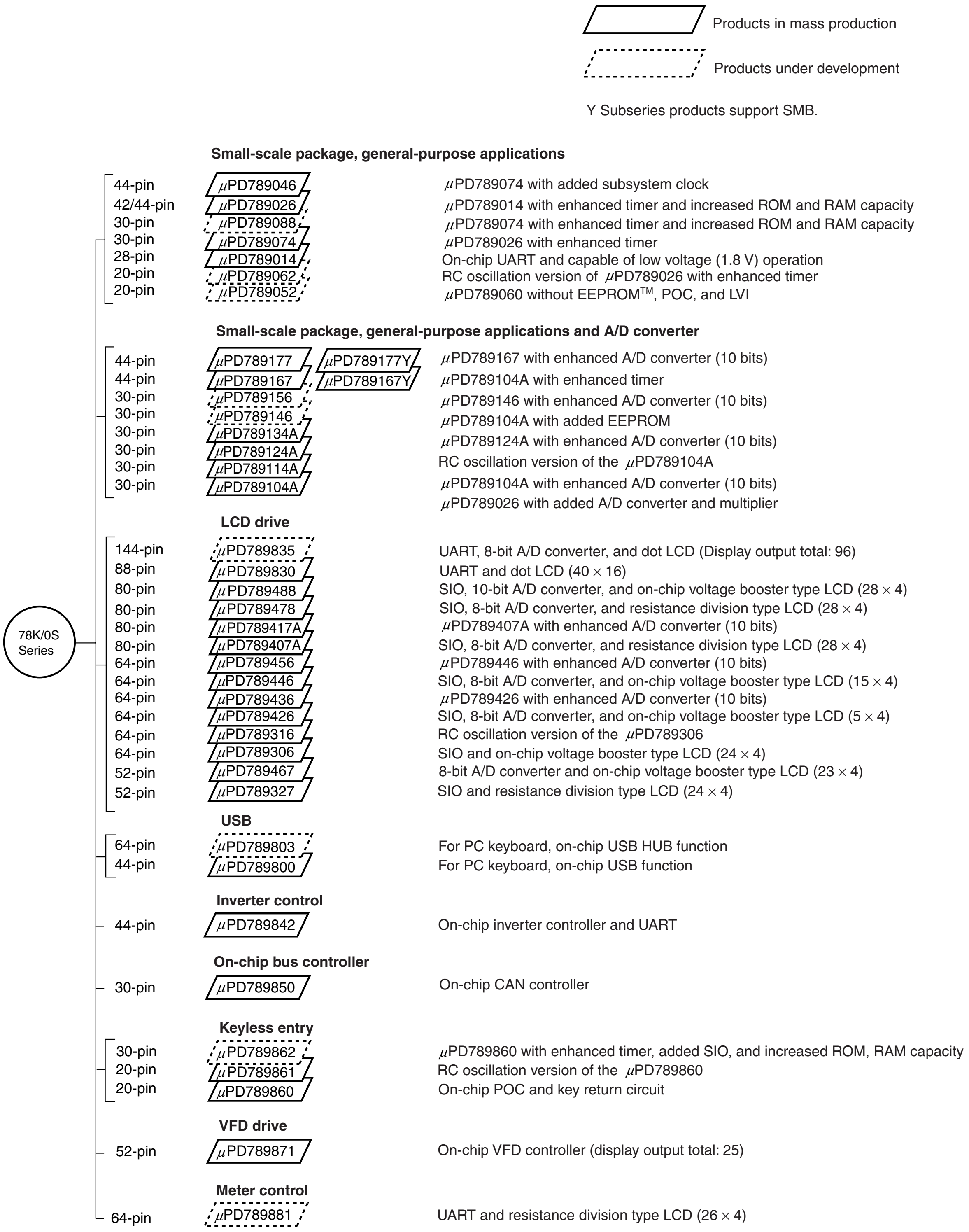
2. Connect the IC0 pin directly to the V_{SS0} or V_{SS1} pin.

3. Leave the IC2 pin open.

BUZ:	Buzzer clock
COM0 to COM15:	RC oscillator
IC0, IC2:	Internally connected
INTP0 to INTP4:	Interrupt from peripherals
NC:	Non-connection
P00 to P07:	Port 0
P10 to P17:	Port 1
P20 to P26:	Port 2
P30 to P34:	Port 3
P40, P41:	Port 4
P50 to P57:	Port 5
PCL:	Programming clock
$\overline{\text{RESET}}$:	Reset
RxD00:	Receive data
S0 to S39:	Segment output
TxD00:	Transmit data
V_{DD0}, V_{DD1}:	Power supply
V_{PP}:	Programming power supply
V_{SS0}, V_{SS1}:	Ground
X1, X2:	Crystal (Main system clock)
XT1, XT2:	Crystal (Subsystem clock)

★ 1.5 78K/0Series Lineup

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



Y Subseries products support SMB.

Small-scale package, general-purpose applications

44-pin	μPD789046	μPD789074 with added subsystem clock
42/44-pin	μPD789026	μPD789014 with enhanced timer and increased ROM and RAM capacity
30-pin	μPD789088	μPD789074 with enhanced timer and increased ROM and RAM capacity
30-pin	μPD789074	μPD789026 with enhanced timer
28-pin	μPD789014	On-chip UART and capable of low voltage (1.8 V) operation
20-pin	μPD789062	RC oscillation version of μPD789026 with enhanced timer
20-pin	μPD789052	μPD789060 without EEPROM™, POC, and LVI

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

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

LCD drive

144-pin	μPD789835	UART, 8-bit A/D converter, and dot LCD (Display output total: 96)
88-pin	μPD789830	UART and dot LCD (40 × 16)
80-pin	μPD789488	SIO, 10-bit A/D converter, and on-chip voltage booster type LCD (28 × 4)
80-pin	μPD789478	SIO, 8-bit A/D converter, and resistance division type LCD (28 × 4)
80-pin	μPD789417A	μPD789407A with enhanced A/D converter (10 bits)
80-pin	μPD789407A	SIO, 8-bit A/D converter, and resistance division type LCD (28 × 4)
64-pin	μPD789456	μPD789446 with enhanced A/D converter (10 bits)
64-pin	μPD789446	SIO, 8-bit A/D converter, and on-chip voltage booster type LCD (15 × 4)
64-pin	μPD789436	μPD789426 with enhanced A/D converter (10 bits)
64-pin	μPD789426	SIO, 8-bit A/D converter, and on-chip voltage booster type LCD (5 × 4)
64-pin	μPD789316	RC oscillation version of the μPD789306
64-pin	μPD789306	SIO and on-chip voltage booster type LCD (24 × 4)
52-pin	μPD789467	8-bit A/D converter and on-chip voltage booster type LCD (23 × 4)
52-pin	μPD789327	SIO and resistance division type LCD (24 × 4)

USB

64-pin	μPD789803	For PC keyboard, on-chip USB HUB function
44-pin	μPD789800	For PC keyboard, on-chip USB function

Inverter control

44-pin	μPD789842	On-chip inverter controller and UART

On-chip bus controller

30-pin	μPD789850	On-chip CAN controller

Keyless entry

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

VFD drive

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

Meter control

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

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

The major differences between subseries are shown 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	VDD	Remarks
			8-Bit	16-Bit	Watch	WDT					MIN. Value	
Small-scale package, general-purpose applications	μPD789046	16 KB	1 ch	1 ch	1 ch	1 ch	–	–	1 ch (UART: 1 ch)	34	1.8 V	–
	μPD789026	4 KB to 16 KB			–							
	μPD789088	16 KB to 32 KB	3 ch							24		
	μPD789074	2 KB to 8 KB	1 ch									
	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	1.8 V Note	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

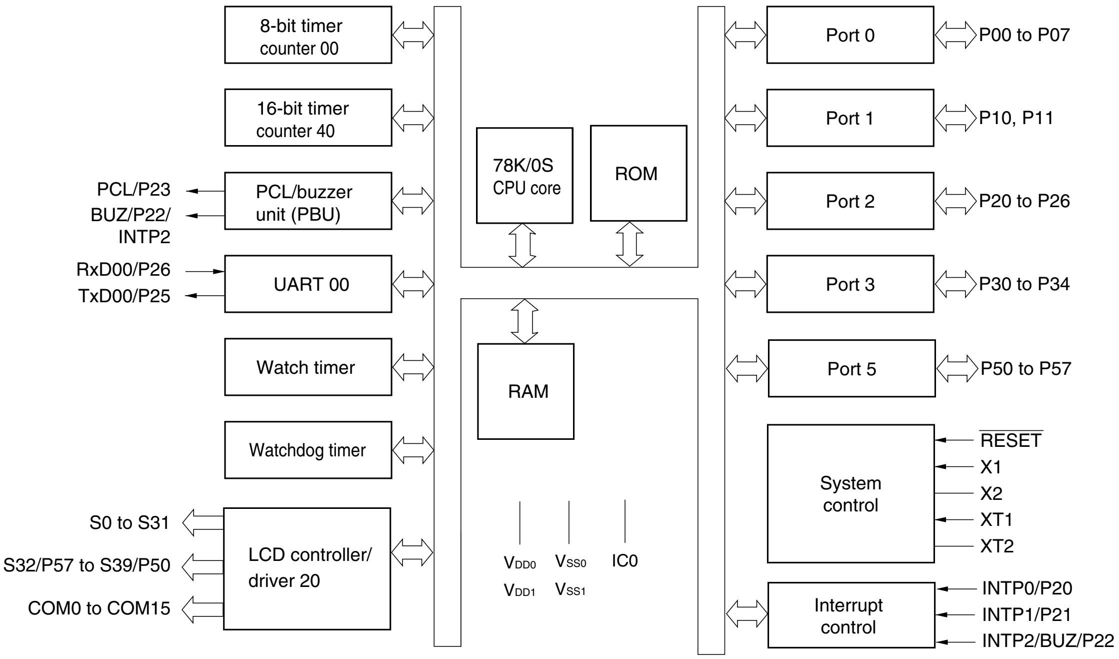
Function Subseries Name		ROM Capacity	Timer				8-Bit A/D	10-Bit A/D	Serial Interface	I/O	V_{DD}	Remarks
			8-Bit	16-Bit	Watch	WDT					MIN. Value	
USB	μPD789803	8 KB to 16 KB	2 ch	–	–	1 ch	–	–	2 ch (USB: 1 ch)	41	3.6 V	–
	μPD789800	8 KB								31	4.0 V	
Inverter control	μPD789842	8 KB to 16 KB	3 ch	**Note 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	2.7 V Note 2	–

Notes 1. 10-bit timer: 1 channel

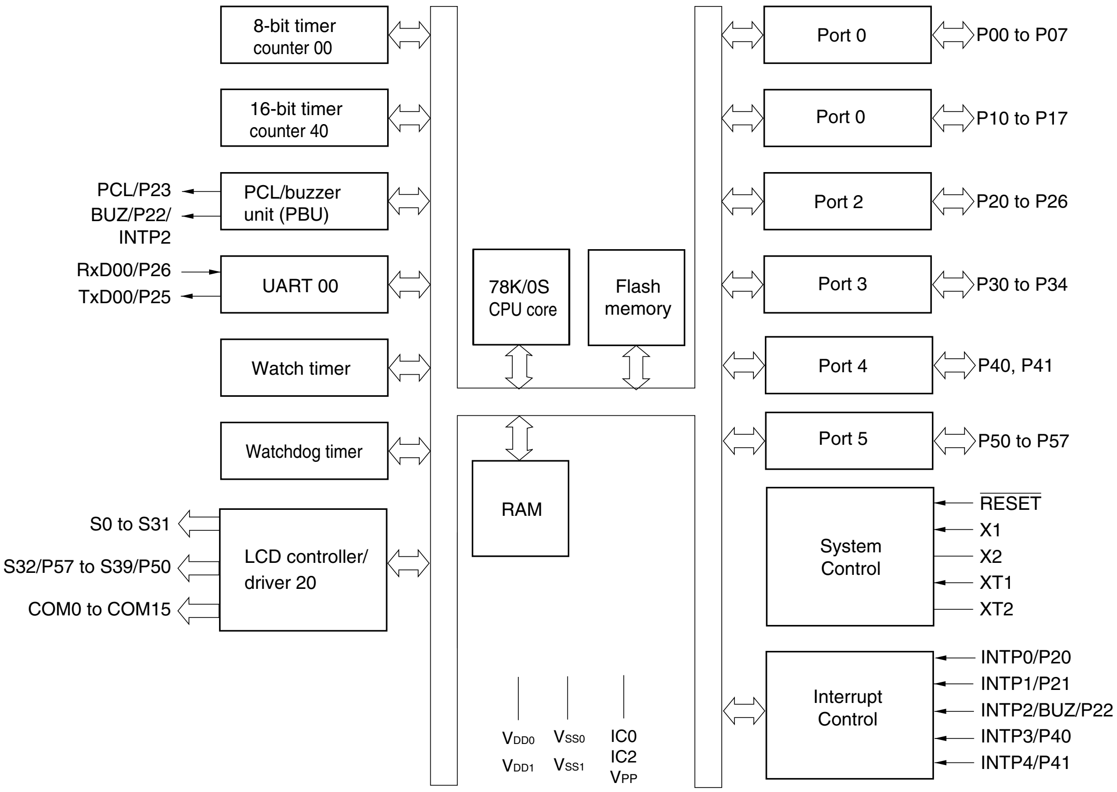
2. Flash memory version: 3.0 V

1.6 Block Diagram

(1) μPD789830



(2) μPD78F9831



1.7 Overview of Functions

Item		μPD789830	μPD78F9831
Internal memory	ROM structure	Mask ROM	Flash memory
	ROM	24 KB	48 KB
	RAM	1 KB	2 KB
	LCD display RAM	80 bytes	
Minimum instruction execution time		• 0.56/2.23 μs (@ 3.58 MHz operation with main system clock) • 122 μs (@ 32.768 kHz operation with subsystem clock)	
Instruction set		• 16-bit operations • Bit manipulations (such as set, reset, and test)	
I/O ports		Total of 30 port pins 29 CMOS I/O pins 1 N-ch open-drain I/O pin	Total of 38 port pins 37 CMOS I/O pins 1 N-ch open-drain I/O pin
Serial interface		UART: 1 channel	
Timers		• 16-bit timer: 1 channel • 8-bit timer: 1 channel • Watch timer: 1 channel • Watchdog timer: 1 channel	
Pulse output		Clock output/buzzer output	
LCD controller/driver		• Segment signal output: 40 lines MAX. • Common signal output: 16 lines MAX. • 1/5 bias mode	
Vectored interrupt sources	Maskable	Internal: 10, external: 4	Internal: 10, external: 6
	Non-maskable	Internal: 1	
Power supply voltage		V_{DD} = 2.7 to 5.5 V	
Operating ambient temperature		T_A = –20°C to +60°C	
Supply format		88-pin bare chip	100-pin plastic LQFP (fine pitch) (14 × 14)

★ The outline of the timer is as follows.

		16-Bit Timer 40	8-Bit Timer 00	Watch Timer	Watchdog Timer
Operating mode	Interval timer	1 channel	1 channel	1 channel[Note 1]	1 channel[Note 2]
	External event counter	–	–	–	–
Function	Timer outputs	–	–	–	–
	Square-wave outputs	–	–	–	–
	Capture	–	–	–	–
	Interrupt sources	3	1	2	2

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

2. The watchdog timer provides the watchdog timer function and interval timer function. Use either of the functions.

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

(1) Port pins (μPD789830)

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, connection of an on-chip pull-up resistor can be specified by setting pull-up resistor option register 0 (PU0).	Input	–
P10, P11	I/O	Port 1 2-bit I/O port Input/output can be specified in 1-bit units. When used as an input port, connection of an on-chip pull-up resistor can be specified by setting pull-up resistor option register 0 (PU0).	Input	–
P20	I/O	Port 2 7-bit I/O port Input/output can be specified in 1-bit units. P24 can be used as an N-ch open-drain I/O port pin.	Input	INTP0
P21				INTP1
P22				INTP2/BUZ
P23				PCL
P24				–
P25				TxD00
P26				RxD00
P30 to P34	I/O	Port 3 5-bit I/O port Input/output can be specified in 1-bit units. When used as an input port, connection of an on-chip pull-up resistor can be specified by setting pull-up resistor option register 0 (PU0).	Input	–
P50 to P57	I/O	Port 5 8-bit I/O port Input/output can be specified in 1-bit units.	Input	S39 to S32

(2) Port pins (μPD78F9831)

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, connection of an on-chip pull-up resistor can be specified by setting pull-up resistor option register 0 (PU0).	Input	–
P10 to P17	I/O	Port 1 8-bit I/O port Input/output can be specified in 1-bit units. When used as an input port, connection of an on-chip pull-up resistor can be specified by setting pull-up resistor option register 0 (PU0).	Input	–
P20	I/O	Port 2 7-bit I/O port Input/output can be specified in 1-bit units. P24 can be used as an N-ch open-drain I/O port pin.	Input	INTP0
P21				INTP1
P22				INTP2/BUZ
P23				PCL
P24				–
P25				TxD00
P26				RxD00
P30 to P34	I/O	Port 3 5-bit I/O port Input/output can be specified in 1-bit units. When used as an input port, connection of an on-chip pull-up resistor can be specified by setting pull-up resistor option register 0 (PU0).	Input	–
P40	I/O	Port 4 2-bit I/O port Input/output can be specified in 1-bit units.		INTP3
P41				INTP4
P50 to P57	I/O	Port 5 8-bit I/O port Input/output can be specified in 1-bit units.	Input	S39 to S32

(3) 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	P20
INTP1				P21
INTP2				P22/BUZ
INTP3[Note]				P40
INTP4[Note]				P41
RxD00	Input	Serial data input to asynchronous serial interface	Input	P26
TxD00	Output	Serial data output from asynchronous serial interface	Input	P25
BUZ	Output	Buzzer output	Input	P22/INTP2
PCL	Output	Clock output	Input	P23
S0 to S31	Output	Segment signal output from LCD controller/driver	Output	–
S32 to S39				P57 to P50
COM0 to COM15	Output	Common signal output from LCD controller/driver	Output	–
X1	Input	Connected to crystal for main system clock oscillation	–	–
X2	–		–	–
XT1	Input	Connected to crystal for subsystem clock oscillation	–	–
XT2	–		–	–
$\overline{\text{RESET}}$	Input	System reset input	Input	–
V_{DD0}	–	Positive supply voltage for ports	–	–
V_{DD1}		Positive supply voltage for circuits other than ports		–
V_{SS0}	–	Port section ground potential	–	–
V_{SS1}		Ground potential of circuits other than ports		–
IC0	–	This pin is internally connected. Connect this pin directly to the V_{SS0} or V_{SS1} pin.	–	–
IC2[Note]		This pin is internally connected. Leave this pin open.	–	–
NC[Note]	–	This pin is not internally connected. Connect this pin directly to the V_{SS0} or V_{SS1} pin (it can also be left open).	–	–
V_{PP}[Note]	–	This pin is used to set flash memory programming mode and applies a high voltage when a program is written or verified.	–	–

Note μPD78F9831 only

2.2 Description of Pin Functions (μPD789830)

2.2.1 P00 to P07 (Port 0)

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

2.2.2 P10, P11 (Port 1)

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

2.2.3 P20 to P26 (Port 2)

These pins constitute a 7-bit I/O port. In addition, these pins provide a function to perform input to the external interrupt, to output the buzzer, to output the pulse, and to input/output the data of UART.

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

(1) Port mode

In port mode, P20 to P26 function as a 7-bit I/O port. Port 2 can be set to input or output mode in 1-bit units by using port mode register 2 (PM2). P24 is an N-ch open-drain I/O port.

(2) Control mode

In this mode, P20 to P26 function as the external interrupt input, buzzer output, pulse output, and input/output of UART data.

(a) INTP0 to INTP2

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

(b) PCL

Pulse output pin of clock output circuit.

(c) BUZ

Buzzer output pin of clock output circuit.

(d) RxD00, TxD00

Serial data I/O pins of UART.

Caution When using P20 to P26 as data I/O pins of UART, the input/output mode and output latch must be set according to the functions to be used. For details of the setting, see (1) in Section 11.3.

2.2.4 P30 to P34 (Port 3)

These pins constitute a 5-bit I/O port. Port 3 can be set to input or output mode in 1-bit units by using port mode register 3 (PM3). When these pins are used as an input port, an on-chip pull-up resistor can be used by setting pull-up resistor option register 0 (PU0). This port can be also used to detect a key return signal in 1-bit units.

2.2.5 P50 to P57 (Port 5)

These pins constitute an 8-bit I/O port. In addition, these pins provide the function to output the LCD controller/driver segment signal. Port 5 can be specified in the following operation modes in 1-bit units.

(1) Port mode

In port mode, P50 to P57 function as an 8-bit I/O port. Port 5 can be set to input or output mode in 1-bit units by using port mode register 5 (PM5).

(2) Control mode

In this mode, P50 to P57 function as a segment signal output of the LCD controller/driver (S32 to S39).

2.2.6 S0 to S31

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

2.2.7 COM0 to COM15

These pins are used to output the common signal of the LCD controller/driver.

2.2.8 $\overline{\text{RESET}}$

This pin inputs an active-low system reset signal.

2.2.9 X1, X2

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

2.2.10 XT1, XT2

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

2.2.11 V_{DD0}, V_{DD1}

V_{DD0} supplies positive power to the ports.

V_{DD1} supplies positive power to circuits other than those of the ports.

2.2.12 V_{SS0}, V_{SS1}

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

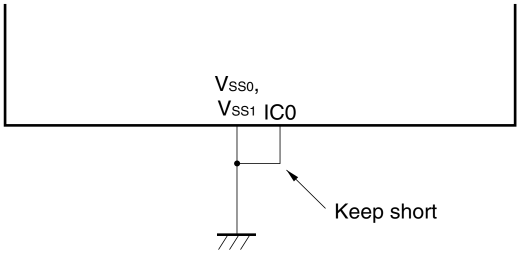
V_{SS1} is the ground pin for circuits other than those of the ports.

2.2.13 IC0

The IC0 (Internally Connected) pin is used to set the μPD789830 Subseries in test mode before shipment. In normal operation mode, directly connect this pin to the V_{SS0} or V_{SS1} pin with as short a wiring length as possible.

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

- Directly connect the IC0 pin to the V_{SS0} or V_{SS1} pin.



2.3 Description of Pin Functions (μPD78F9831)

2.3.1 P00 to P07 (Port 0)

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

2.3.2 P10 to P17 (Port 1)

These pins constitute an 8-bit I/O port and can be set to input or output port mode in 1-bit units by using port mode register 1 (PM1). When these pins are used as an input port, an on-chip pull-up resistor can be used by setting pull-up resistor option register 0 (PU0).

2.3.3 P20 to P26 (Port 2)

These pins constitute a 7-bit I/O port. In addition, these pins provide a function to perform input to the external interrupt, to output the buzzer, to output the pulse, and to input/output the data of UART.

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

(1) Port mode

In port mode, P20 to P26 function as a 7-bit I/O port. Port 2 can be set to input or output mode in 1-bit units by using port mode register 2 (PM2). P24 is an N-ch open-drain I/O port.

(2) Control mode

In this mode, P20 to P26 function as the external interrupt input, buzzer output, pulse output, and input/output of UART data.

(a) INTP0 to INTP2

External interrupt input pins whose valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

(b) PCL

Pulse output pin of clock output circuit.

(c) BUZ

Buzzer output pin of clock output circuit.

(d) RxD00, TxD00

Serial data I/O pins of UART.

Caution When using P20 to P26 as data I/O pins of UART, the input/output mode and output latch must be set according to the functions to be used. For details of the setting, see (1) in Section 11.3.

2.3.4 P30 to P34 (Port 3)

These pins constitute a 5-bit I/O port. Port 3 can be set to input or output mode in 1-bit units by using port mode register 3 (PM3). When these pins are used as an input port, an on-chip pull-up resistor can be used by setting pull-up resistor option register 0 (PU0). This port can be also used to detect a key return signal in 1-bit units.

2.3.5 P40, P41 (Port 4)

These pins constitute a 2-bit I/O port. In addition, these pins provide the function to perform input to the external interrupt. Port 4 can be specified in the following operation modes in 1-bit units.

(1) Port mode

In port mode, P40 and P41 function as a 2-bit I/O port. Port 4 can be set to input or output mode in 1-bit units by using port mode register 4 (PM4).

(2) Control mode

In this mode, P40 and P41 function as an external interrupt input (INTP3, INTP4).

2.3.6 P50 to P57 (Port 5)

These pins constitute an 8-bit I/O port. In addition, these pins provide the function to output the LCD controller/driver segment signal. Port 5 can be specified in the following operation modes in 1-bit units.

(1) Port mode

In port mode, P50 to P57 function as an 8-bit I/O port. Port 5 can be set to input or output mode in 1-bit units by using port mode register 5 (PM5).

(2) Control mode

In this mode, P50 to P57 function as a segment signal output of the LCD controller/driver (S32 to S39).

2.3.7 S0 to S31

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

2.3.8 COM0 to COM15

These pins are used to output the common signal of the LCD controller/driver.

2.3.9 $\overline{\text{RESET}}$

This pin inputs an active-low system reset signal.

2.3.10 X1, X2

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

2.3.11 XT1, XT2

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

2.3.12 V_{DD0}, V_{DD1}

V_{DD0} supplies positive power to the ports.

V_{DD1} supplies positive power to circuits other than those of the ports.

2.3.13 V_{SS0}, V_{SS1}

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

V_{SS1} is the ground pin for circuits other than those of the ports.

2.3.14 V_{PP}

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

★ Handle this pin in either of the following ways.

- Connect a 10 kΩ pull-down resistor to the pin.
- Provide a jumper on the board so that the pin is connected to a dedicated flash programmer in programming mode and to V_{SS0} or V_{SS1} directly in normal operation mode.

2.3.15 IC0

The IC0 pin is connected internally. Connect this pin to V_{SS0} or V_{SS1}.

2.3.16 IC2

The IC2 pin is connected internally. Leave this pin open.

2.3.17 NC

The NC (Non-connection) pin is not connected internally. Connect this pin to V_{SS0} or V_{SS1} (it can be also left open).

2.4 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 Tables 2-1 and 2-2. For the I/O circuit configuration of each type, refer to Figure 2-1.

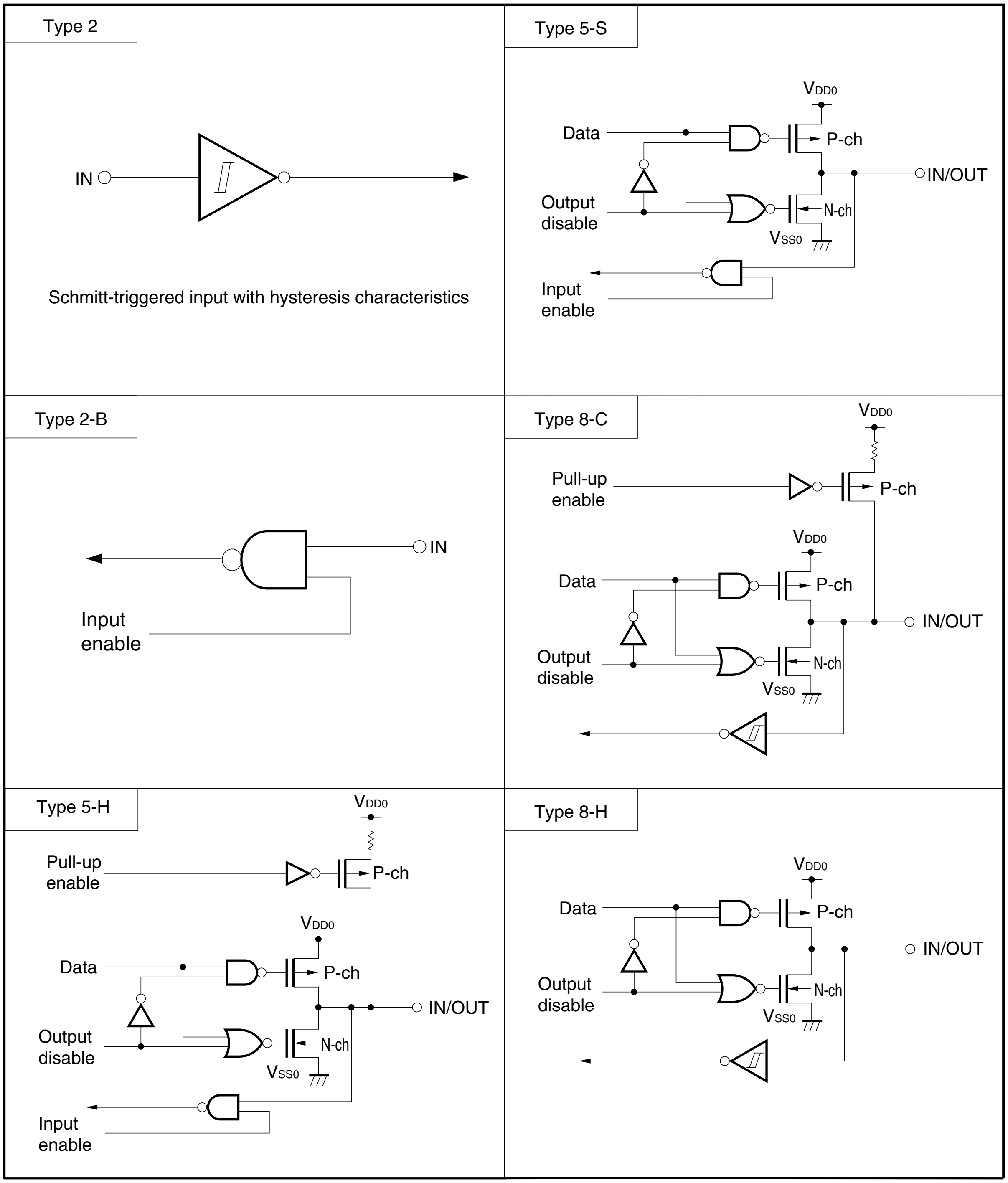
★ **Table 2-1. Types of Pin I/O Circuits and Recommended Connection of Unused Pins (μPD789830)**

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00 to P07	5-H	I/O	Input: Independently connect to V_{DD0}, V_{DD1}, V_{SS0}, or V_{SS1} via a resistor. Output: Leave open.
P10, P11			
P20/INTP0	8-H		Input: Independently connect to V_{SS0} or V_{SS1} via a resistor. Output: Leave open.
P21/INTP1			
P22/INTP2/BUZ			
P23/PCL	5-S		Input: Independently connect to V_{DD0}, V_{DD1}, V_{SS0}, or V_{SS1} via a resistor. Output: Leave open.
P24	13-AB		Input: Independently connect to V_{DD0} or V_{DD1} via a resistor. Output: Leave open.
P25/TxD00	5-S		Input: Independently connect to V_{DD0}, V_{DD1}, V_{SS0}, or V_{SS1} via a resistor. Output: Leave open.
P26/RxD00	8-H		
P30 to P34	8-C		
P50/S39 to P57/S32	17-I		
S0 to S31	17-H	Output	Leave open.
COM0 to COM15	18-C		
XT1	16	Input	Connect directly to V_{SS0} or V_{SS1}.
XT2		–	Leave open.
$\overline{\text{RESET}}$	2	Input	–
IC0	–	–	Connect directly to V_{SS0} or V_{SS1}.

★ **Table 2-2. Types of Pin I/O Circuits and Recommended Connection of Unused Pins (μPD78F9831)**

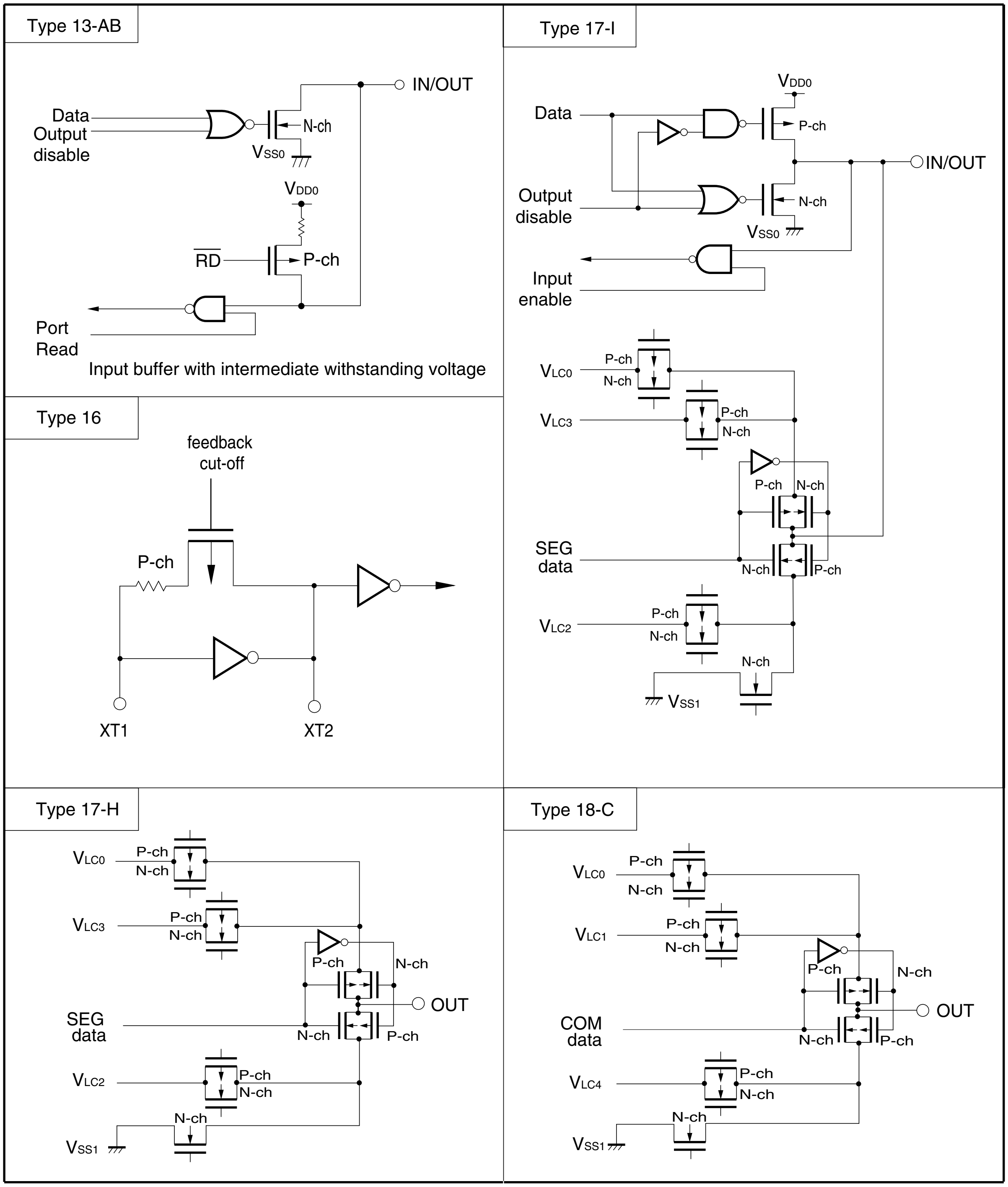
Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00 to P07	5-H	I/O	Input: Independently connect to V_{DD0}, V_{DD1}, V_{SS0}, or V_{SS1} via a resistor. Output: Leave open.
P10 to P17			
P20/INTP0	8-H		Input: Independently connect to V_{SS0} or V_{SS1} via a resistor. Output: Leave open.
P21/INTP1			
P22/INTP2/BUZ			
P23/PCL	5-S		Input: Independently connect to V_{DD0}, V_{DD1}, V_{SS0}, or V_{SS1} via a resistor. Output: Leave open.
P24	13-AB		Input: Independently connect to V_{DD0} or V_{DD1} via a resistor. Output: Leave open.
P25/TxD00	5-S		Input: Independently connect to V_{DD0}, V_{DD1}, V_{SS0}, or V_{SS1} via a resistor. Output: Leave open.
P26/RxD00	8-H		
P30 to P34	8-C		
P40/INTP3	8-H		Input: Independently connect to V_{SS0} or V_{SS1} via a resistor. Output: Leave open.
P41/INTP4			
P50/S39 to P57/S32	17-I		Input: Independently connect to V_{DD0}, V_{DD1}, V_{SS0}, or V_{SS1} via a resistor. Output: Leave open.
S0 to S31	17-H	Output	Leave open.
COM0 to COM15	18-C		
XT1	16	Input	Connect directly to V_{SS0} or V_{SS1}.
XT2		–	Leave open.
$\overline{\text{RESET}}$	2	Input	–
IC0	2-B		Connect directly to V_{SS0} or V_{SS1}.
IC2	–	–	Leave open.
NC			
V_{PP}			Independently connect a 10 kΩ pull-down resistor or connect directly to V_{SS0} or V_{SS1}.

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



Schmitt-triggered input with hysteresis characteristics

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



3.1 Memory Space

The μPD789830 Subseries can access up to 64 KB of memory space.

Figures 3-1 and 3-2 show the memory maps.

Figure 3-1. Memory Map (μPD789830)

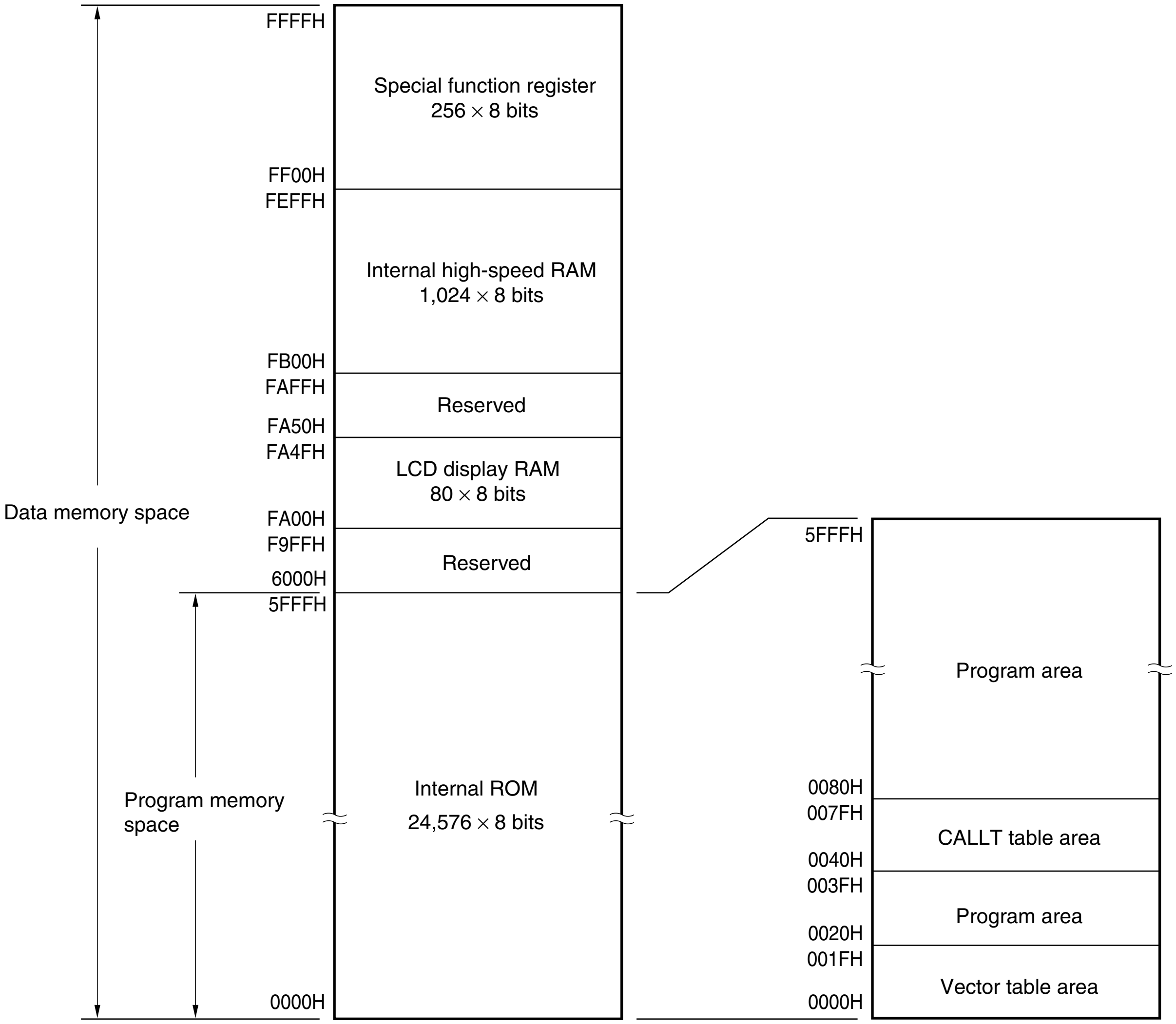
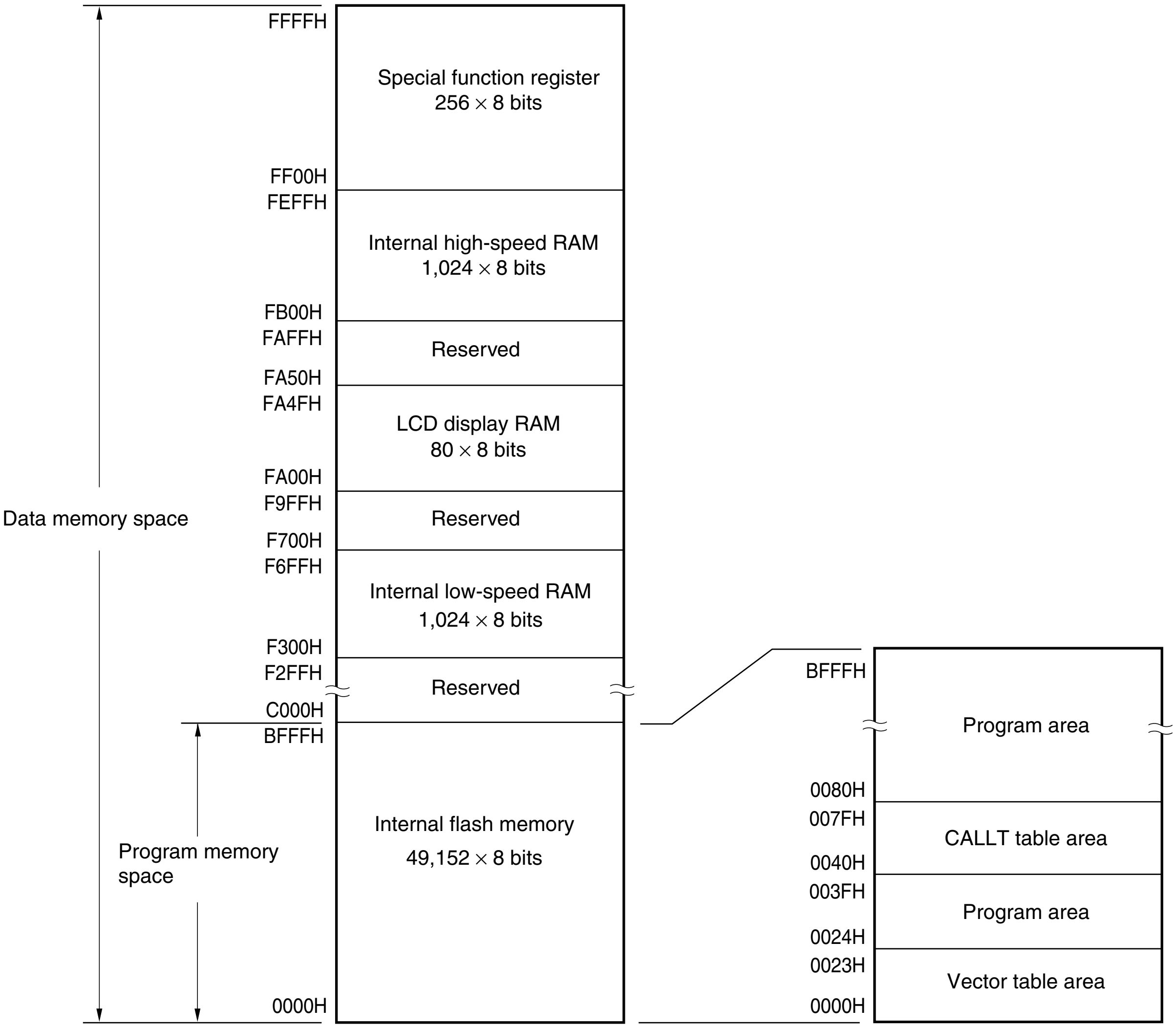


Figure 3-2. Memory Map (μPD78F9831)



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 μPD789830 Subseries provide the following internal ROMs (or flash memory) containing the following capacities.

Table 3-1. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD789830	Mask ROM	24,576 × 8 bits
μPD78F9831	Flash memory	49,152 × 8 bits

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

(1) Vector table area

A 32-byte area of addresses 0000H to 001FH (μPD789830) or a 36-byte area of address 0000H to 0023H (μPD78F9831) 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	0014H	INTTM41
0004H	INTWDT	0016H	INTTM4
0006H	INTP0	0018H	INTTM00
0008H	INTP1	001AH	INTWTI
000AH	INTP2	001CH	INTWT
000CH	INTSER00	001EH	INTKR00
000EH	INTSR00	0020H[Note]	INTP3[Note]
0010H	INTST00	0022H[Note]	INTP4[Note]
0012H	INTTM40		

Note μPD78F9831 only

(2) CALLT instruction table area

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

3.1.2 Internal data memory space

The μPD789830 Subseries provide the following RAM.

(1) Internal RAM

Each member of the μPD789830 Subseries provides the following internal RAM.

Table 3-3. Internal RAM Capacity

Part Number	Capacity	Address
μPD789830	1,024 × 8 bits	FB00H to FEFFH
μPD78F9831	2,048 × 8 bits	FB00H to FEFFH (1,024 × 8 bits)
		F300H to F6FFH (1,024 × 8 bits)

The internal RAM is also used as a stack.

(2) LCD display RAM

An LCD display RAM is allocated in the area between FA00H and FA4FH. 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 to an area of FF00H to FFFFH (see **Table 3-4**).

3.1.4 Data memory addressing

Each of the μPD789830 Subseries is provided with a wide range of addressing modes to make memory manipulation as efficient as possible. A data memory area (FB00H to FFFFH) can be accessed using a unique addressing mode according to its use, such as a special function register (SFR). Figures 3-3 and 3-4 illustrate the data memory addressing modes.

Figure 3-3. Data Memory Addressing (μPD789830)

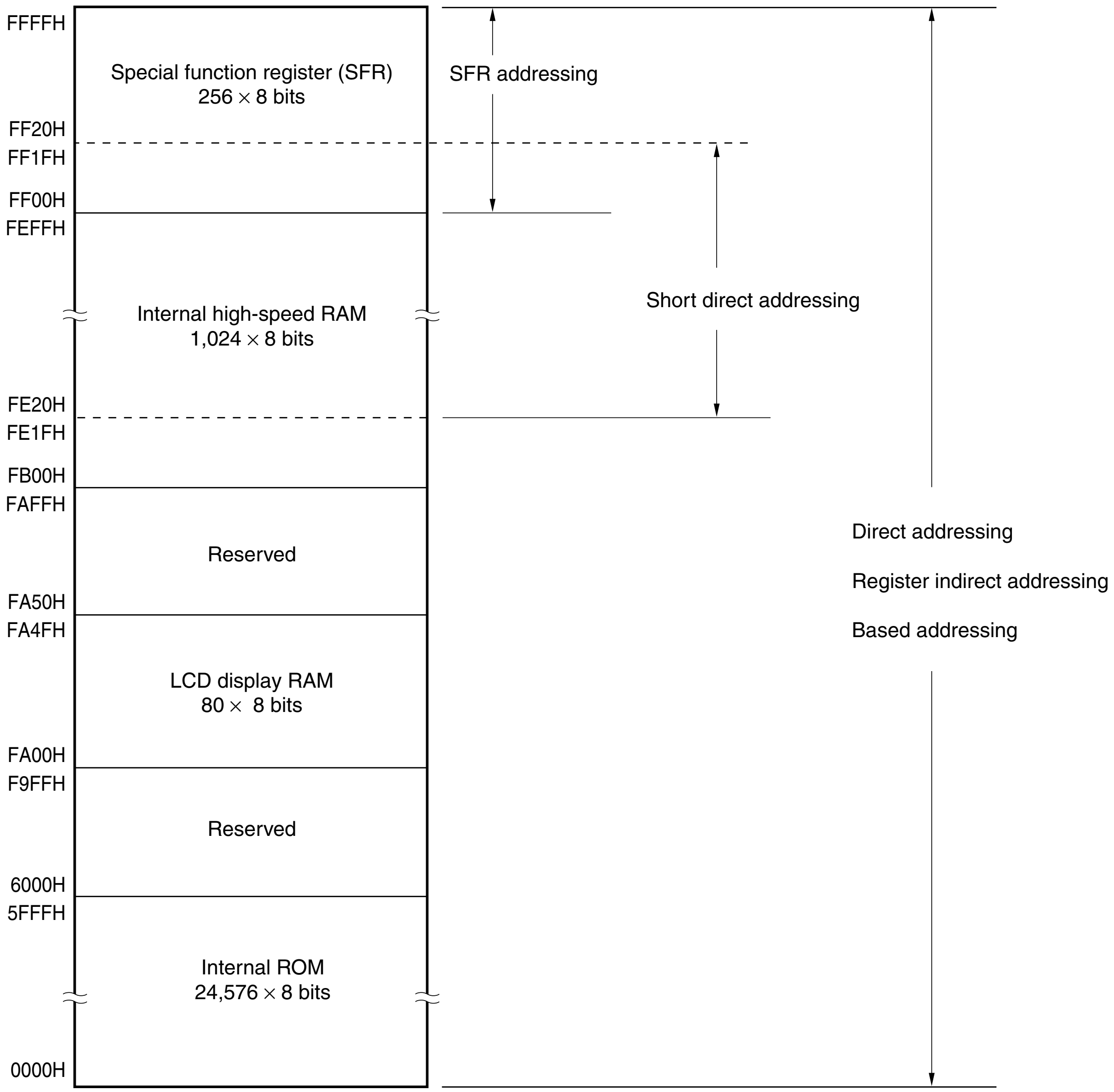
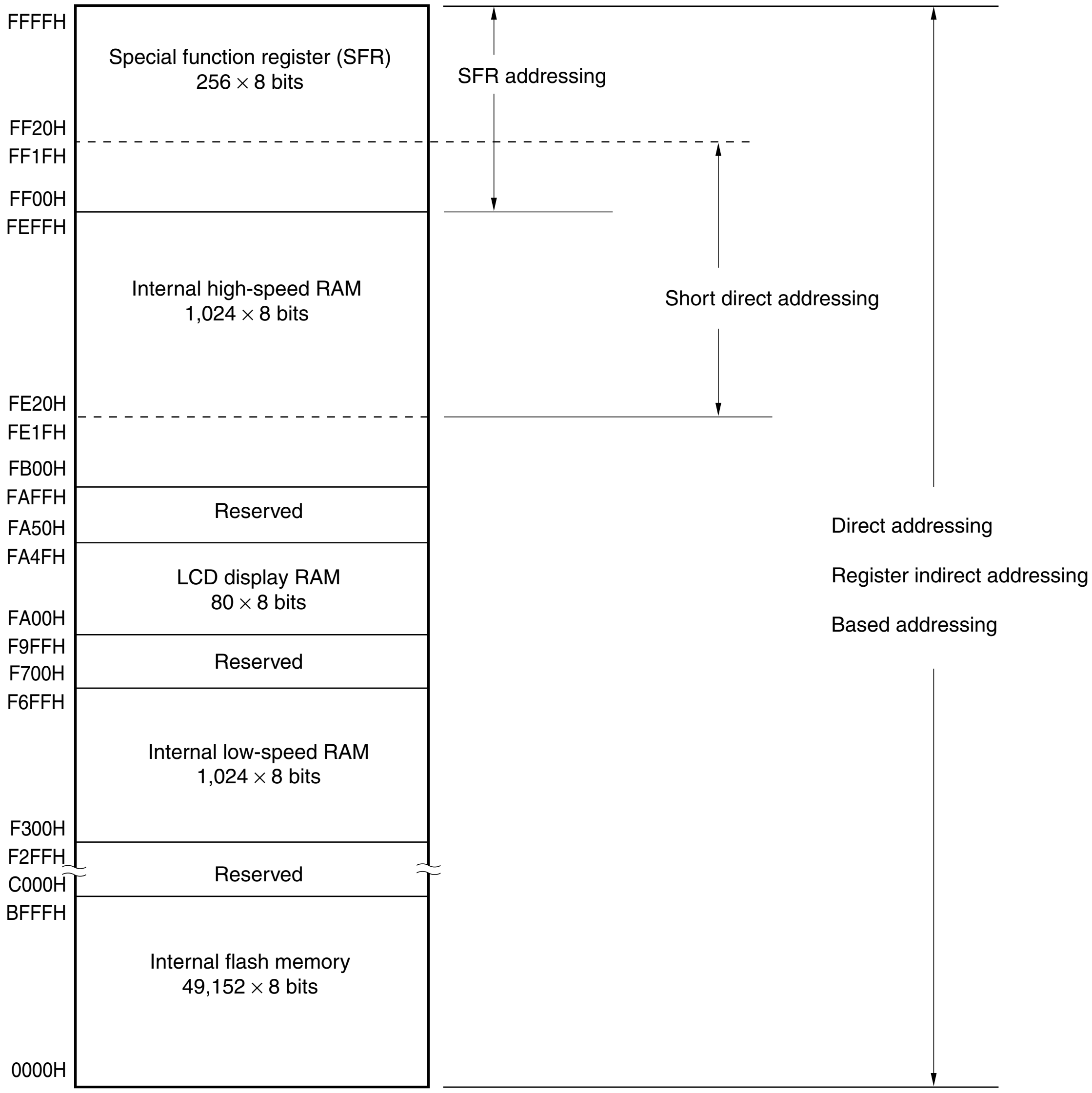


Figure 3-4. Data Memory Addressing (μPD78F9831)



3.2 Processor Registers

The μPD789830 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. A program counter, a program status word, and a stack pointer are control registers.

(1) Program counter (PC)

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

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

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

Figure 3-5. Program Counter Configuration

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

(2) Program status word (PSW)

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

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 the PSW to 02H.

Figure 3-6. Program Status Word Configuration

	7							0
PSW	IE	Z	0	AC	0	0	1	CY

(a) Interrupt enable flag (IE)

This flag controls interrupt request acknowledge operations of the CPU.

When IE = 0, the IE is set to interrupt disabled (DI) status. All interrupt requests except non-maskable interrupt are disabled.

When IE = 1, the IE is set to interrupt enabled (EI) status and interrupt request acknowledgment is controlled with an interrupt mask flag for various interrupt sources.

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

(b) Zero flag (Z)

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

(c) Auxiliary carry flag (AC)

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

(d) Carry flag (CY)

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

(3) Stack pointer (SP)

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

Figure 3-7. Stack Pointer Configuration

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

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

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

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

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

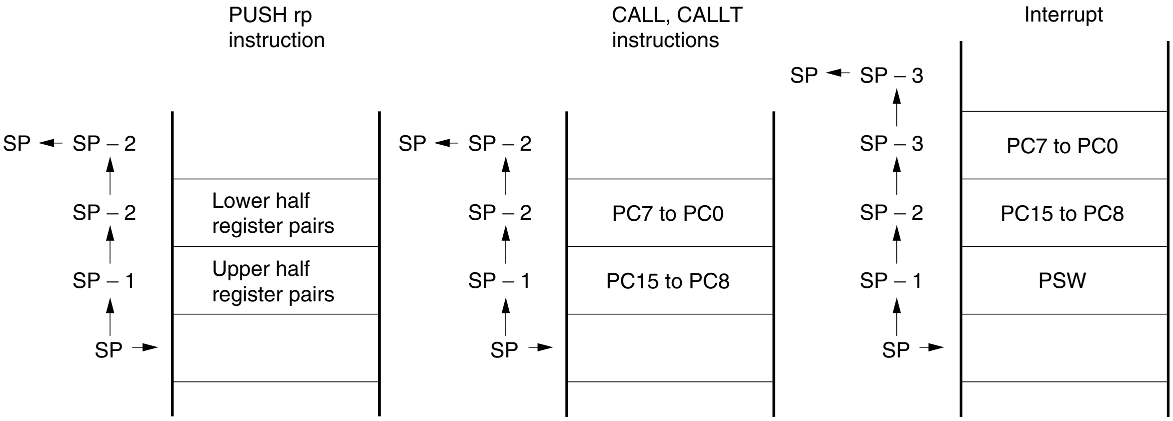
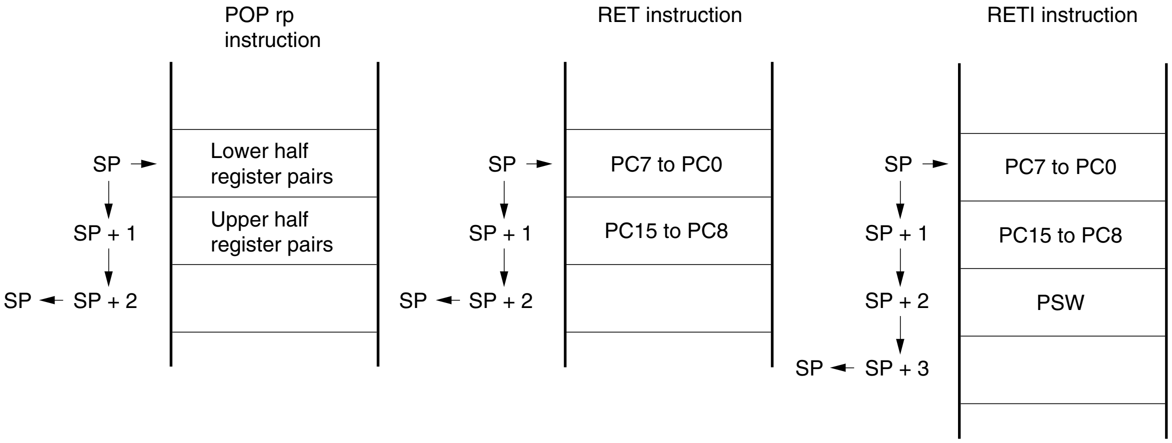


Figure 3-9. Data to Be Restored from Stack Memory



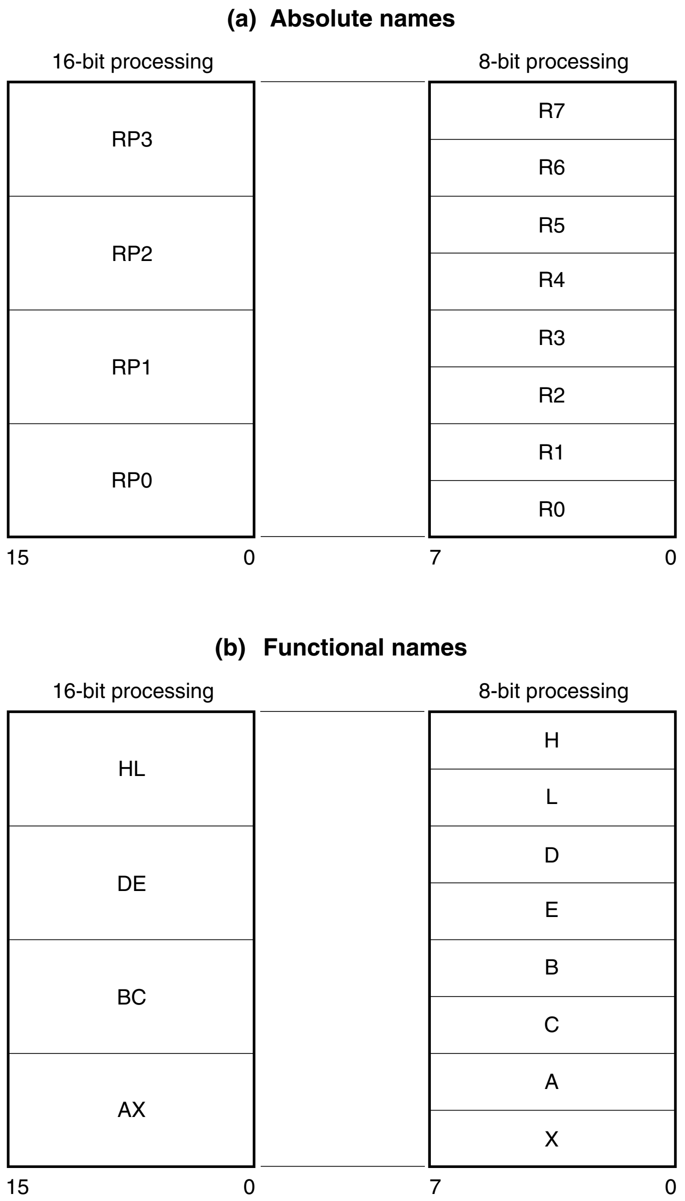
3.2.2 General-purpose registers

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

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

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

Figure 3-10. General-Purpose Register Configuration



(a) Absolute names

(b) Functional names

3.2.3 Special function register (SFR)

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

It is allocated in the 256-byte area FF00H to FFFFH.

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

Each manipulation bit unit can be specified as follows.

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

Table 3-4 lists the special function register. 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 C compiler. Therefore, these symbols can be used as instruction operands if 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

- Number of bits manipulated simultaneously
  Indicates the bit units (1, 8, and 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	Manipulatable Bit Unit			After Reset
					1 Bit	8 Bits	16 Bits	
FF00H	Port 0	P0		R/W	√	√	–	00H
FF01H	Port 1	P1			√	√	–	
FF02H	Port 2	P2			√	√	–	
FF03H	Port 3	P3			√	√	–	
FF04H	Port 4[Note 1]	P4			√	√	–	
FF05H	Port 5	P5			√	√	–	
FF16H	16-bit compare register 40	CR40L	CR40		–	√	√[Note 2]	0000H
FF17H		CR40H				√		
FF20H	Port mode register 0	PM0			√	√	–	FFH
FF21H	Port mode register 1	PM1			√	√	–	
FF22H	Port mode register 2	PM2			√	√	–	
FF23H	Port mode register 3	PM3			√	√	–	
FF24H	Port mode register 4[Note 1]	PM4			√	√	–	
FF25H	Port mode register 5	PM5			√	√	–	
FF40H	PCL/BUZ control register 0	PBS0			√	√	–	00H
FF42H	Timer clock selection register 2	TCL2			–	√	–	
FF4AH	Watch timer mode control register	WTM			√	√	–	
FF50H	8-bit compare register 00	CR00		W	–	√	–	Undefined
FF51H	8-bit timer counter 00	TM00		R	–	√	–	00H
FF53H	8-bit timer mode control register 00	TMC00		R/W	√	√	–	
FF5BH	Timer 40 control register	TMC40			√	√	–	
FFA0H	Asynchronous serial interface mode register 00	ASIM00			√	√	–	
FFA1H	Asynchronous serial interface status register 00	ASIS00		R	√	√	–	
FFA2H	Baud rate generator control register 00	BRGC00		R/W	–	√	–	
FFA3H	Transmission shift register 00	TXS00		W	–	√	–	FFH
	Reception buffer register 00	RXB00		R	–	√	–	
FFB0H	LCD20 mode register	LCDM20		R/W	√	√	–	00H
FFB1H	Alternate port function selection register	PF5			–	√	–	
FFB2H	LCD20 clock selection register	LCDC20			–	√	–	
FFE0H	Interrupt request flag register 0	IF0			√	√	–	
FFE1H	Interrupt request flag register 1	IF1			√	√	–	
FFE4H	Interrupt mask flag register 0	MK0			√	√	–	FFH
FFE5H	Interrupt mask flag register 1	MK1			√	√	–	

Notes 1. μPD78F9831 only

2. 16-bit access is allowed only with short direct addressing.

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FFECH	External interrupt mode register 0	INTM0	R/W	–	√	–	00H
FFEDH	External interrupt mode register 1[Note]	INTM1		–	√	–	
FFF0H	Suboscillation mode register	SCKM		√	√	–	
FFF2H	Subclock control register	CSS		√	√	–	
FFF5H	Key return mode register 00	KRM00		–	√	–	
FFF7H	Pull-up resistor option register 0	PU0		√	√	–	
FFF9H	Watchdog timer mode register	WDTM		√	√	–	
FFFAH	Oscillation stabilization time selection register	OSTS		–	√	–	04H
FFFBH	Processor clock control register	PCC		√	√	–	02H

Note μPD78F9831 only

3.3 Instruction Address Addressing

An instruction address is determined by program counter (PC) contents. 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 Instruction User's Manual (U11047E)**).

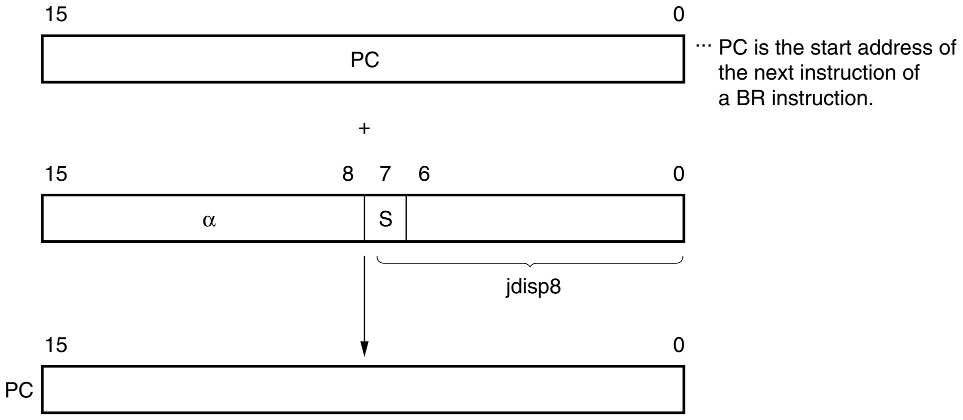
3.3.1 Relative addressing

[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. In other words, the range of branch in relative addressing is between –128 and +127 of the start address of the following instruction.

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

[Illustration]


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

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

3.3.2 Immediate addressing

[Function]

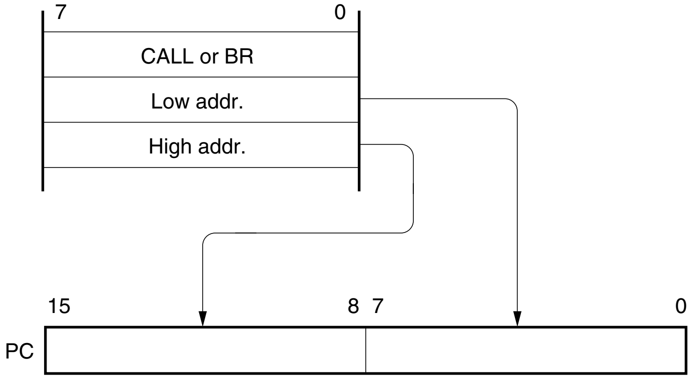
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

This function is carried out when the CALL !addr16 and BR !addr16 instructions are executed.

CALL !addr16 and BR !addr16 instructions can branch to all the memory spaces.

[Illustration]

In case of CALL !addr16 and BR !addr16 instructions



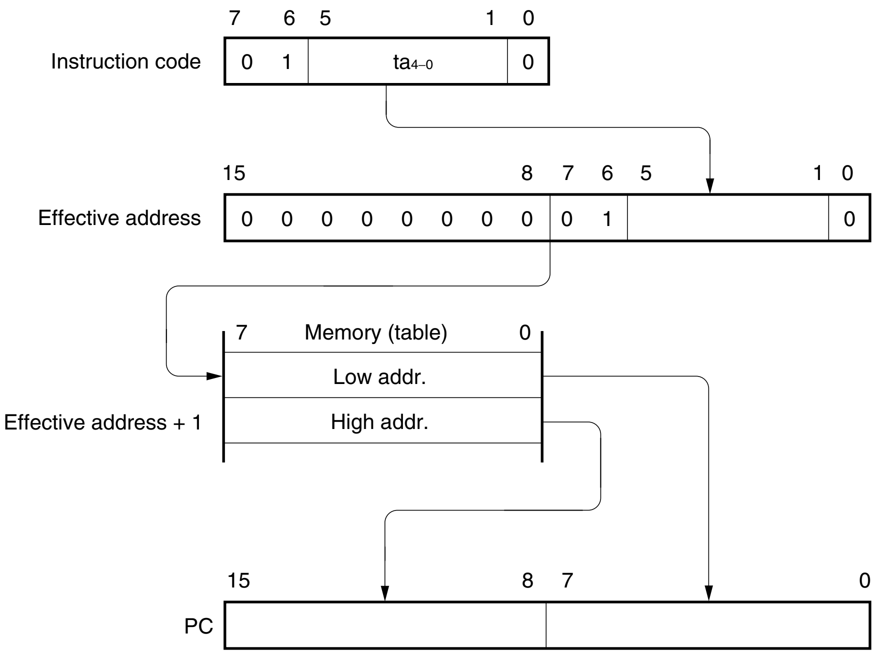
3.3.3 Table indirect addressing

[Function]

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

Table indirect addressing is carried out when the CALLT [addr5] instruction is executed. This instruction can refer to the address stored in the memory table 40H to 7FH and branch to all the memory spaces.

[Illustration]



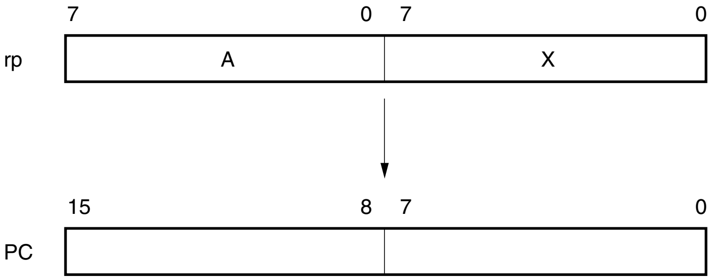
3.3.4 Register addressing

[Function]

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

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

[Illustration]



3.4 Operand Address Addressing

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

3.4.1 Direct addressing

[Function]

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

[Operand format]

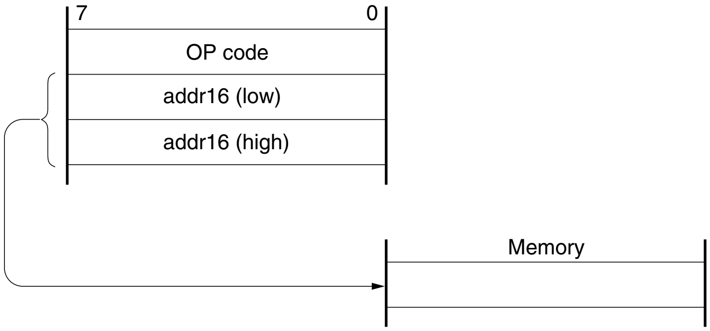
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

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

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

[Illustration]



3.4.2 Short direct addressing

[Function]

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

The fixed space where this addressing is applied is the 256-byte space FE20H to FF1FH. An internal high-speed RAM and a special function register (SFR) 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 all SFR areas. In this area, ports which are frequently accessed in a program and a compare register of the timer/event counter are mapped, and these SFRs can be manipulated with a small number of bytes and clocks.

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

[Operand format]

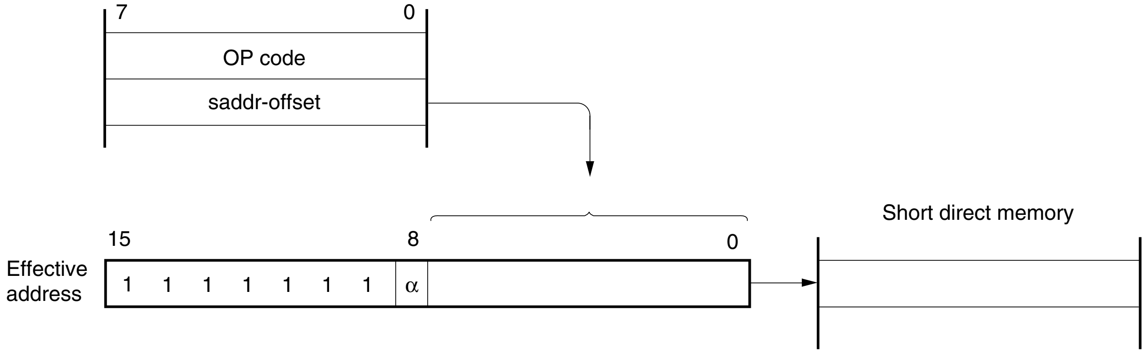
Identifier	Description
saddr	Label or FE20H to FF1FH immediate data
saddrp	Label or FE20H to FF1FH immediate data (even address only)

[Description example]

MOV FE30H, #50H; When setting saddr to FE30H and the immediate data to 50H

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

[Illustration]



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

3.4.3 Special function register (SFR) addressing

[Function]

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

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

[Operand format]

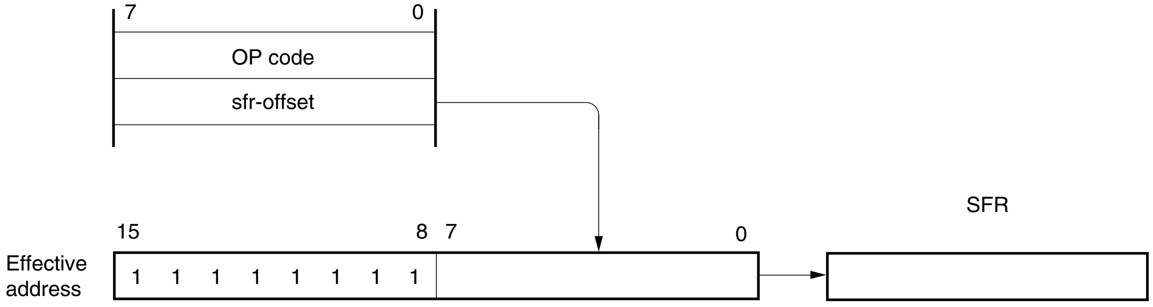
Identifier	Description
sfr	Special function register name

[Description example]

MOV PM0, A; When selecting PM0 for sfr

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

[Illustration]



3.4.4 Register addressing

[Function]

The general-purpose register is accessed as an operand. The general-purpose register to be accessed is specified with register specification code and 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 three bits in the instruction code.

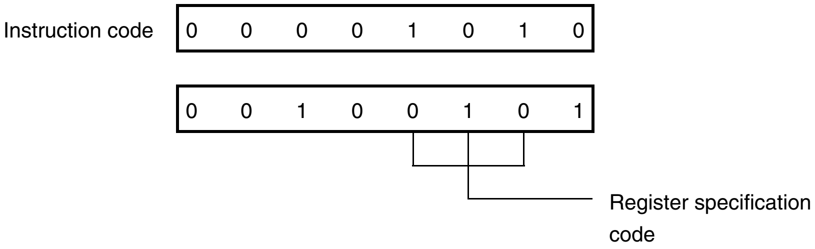
[Operand format]

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

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

[Description example]

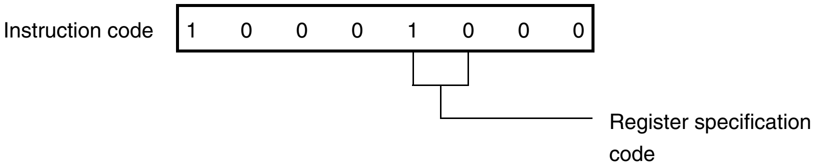
MOV A, C; When selecting the C register for r



Instruction code 0 0 0 0 1 0 1 0

0 0 1 0 0 1 0 1

INCW DE; When selecting the DE register pair for rp



Instruction code 1 0 0 0 1 0 0 0

3.4.5 Register indirect addressing

[Function]

The memory is addressed with the contents of the register pair specified as an operand. The register pair to be accessed is specified with the register pair specification code in the instruction code. This addressing can be carried out for all the memory spaces.

[Operand format]

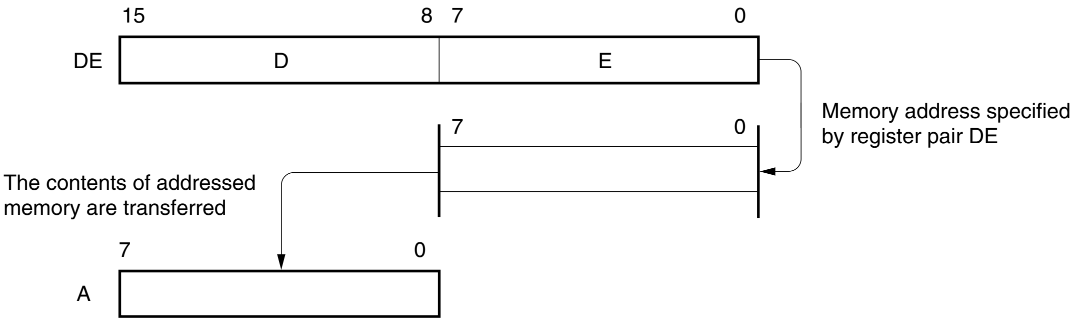
Identifier	Description
–	[DE], [HL]

[Description example]

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

Instruction code	0	0	1	0	1	0	1	1

[Illustration]


15
8
7
0
DE
D
E
Memory address specified by register pair DE
7
0
The contents of addressed memory are transferred
7
0
A

3.4.6 Based addressing

[Function]

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

[Operand format]

Identifier	Description
–	[HL+byte]

[Description example]

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

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

3.4.7 Stack addressing

[Function]

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

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

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

[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 μPD789830 Subseries is provided with the ports shown in Figures 4-1 and 4-2. These ports are used to enable several types of control. Tables 4-1 and 4-2 list the functions of each port.

These ports, while originally designed as digital input/output ports, can also be used for other functions, as summarized in **CHAPTER 2**.

Figure 4-1. Port Types (μPD789830)

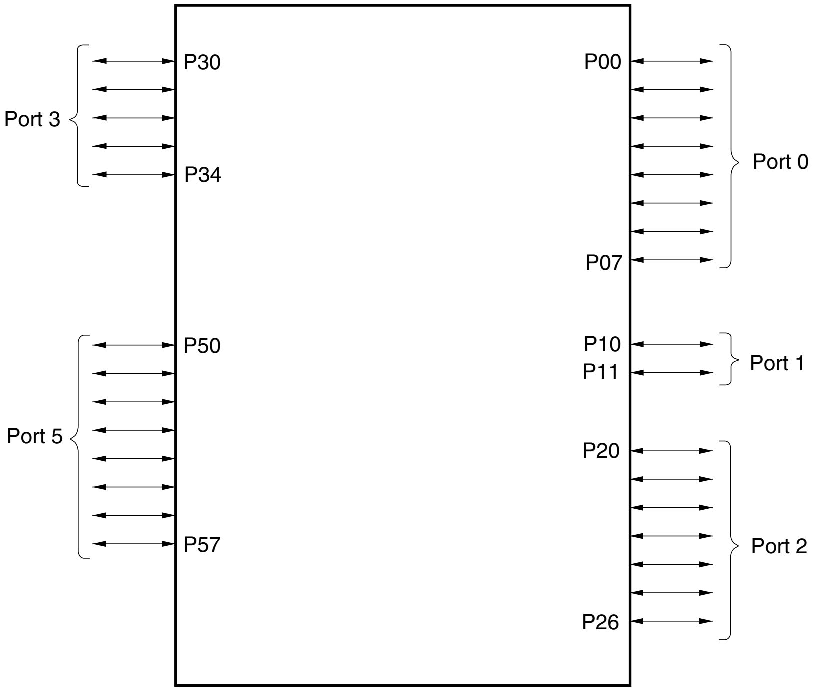


Figure 4-2. Port Types (μPD78F9831)

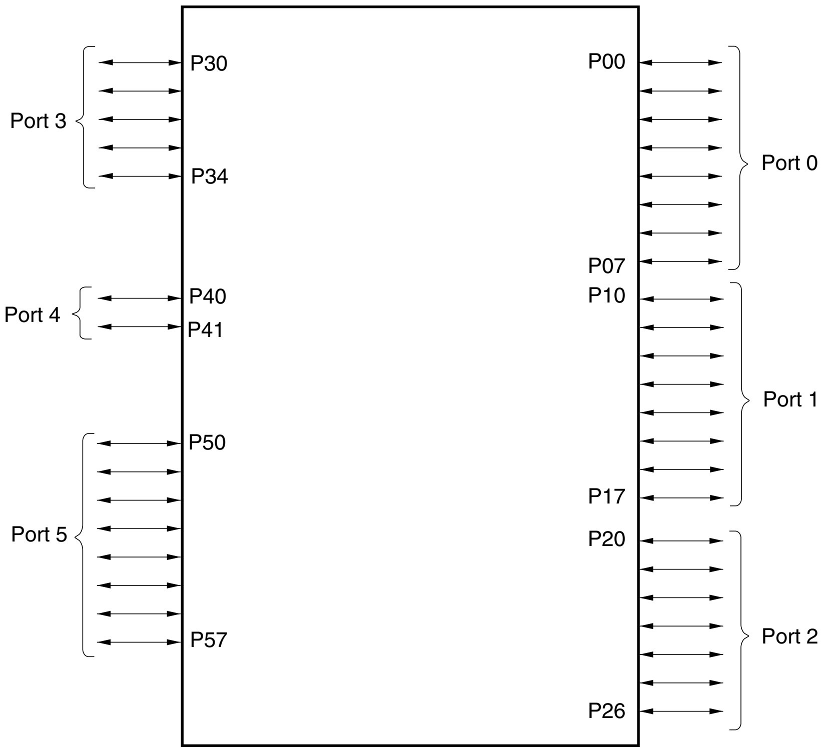


Table 4-1. Port Functions (μPD789830)

Name	Pin Name	Function
Port 0	P00 to P07	I/O port Input/output can be specified in 1-bit units. When used as an input port, connection of an on-chip pull-up resistor can be specified by setting pull-up resistor option register 0 (PU0).
Port 1	P10, P11	I/O port Input/output can be specified in 1-bit units. When used as an input port, connection of an on-chip pull-up resistor can be specified by setting pull-up resistor option register 0 (PU0).
Port 2	P20 to P26	I/O port Input/output can be specified in 1-bit units. P24 is an N-ch open-drain input/output port.
Port 3	P30 to P34	I/O port Input/output can be specified in 1-bit units. When used as an input port, connection of an on-chip pull-up resistor can be specified by setting pull-up resistor option register 0 (PU0).
Port 5	P50 to P57	I/O port Input/output can be specified in 1-bit units.

Table 4-2. Port Functions (μPD78F9831)

Name	Pin Name	Function
Port 0	P00 to P07	I/O port Input/output can be specified in 1-bit units. When used as an input port, connection of an on-chip pull-up resistor can be specified by setting pull-up resistor option register 0 (PU0).
Port 1	P10 to P17	I/O port Input/output can be specified in 1-bit units. When used as an input port, connection of an on-chip pull-up resistor can be specified by setting pull-up resistor option register 0 (PU0).
Port 2	P20 to P26	I/O port Input/output can be specified in 1-bit units. P24 is an N-ch open-drain input/output port.
Port 3	P30 to P34	I/O port Input/output can be specified in 1-bit units. When used as an input port, connection of an on-chip pull-up resistor can be specified by setting pull-up resistor option register 0 (PU0).
Port 4	P40, P41	I/O port Input/output can be specified in 1-bit units.
Port 5	P50 to P57	I/O port Input/output can be specified in 1-bit units.

4.2 Port Configuration

Ports include the following hardware.

Table 4-3. Configuration of Port

Item	Configuration

Control register	Port mode register (PMm)
	m = 0 to 3, 5 (μPD789830)
	m = 0 to 5 (μPD78F9831)

	Pull-up resistor option register (PU0)
Port	• μPD789830 Total: 30 (CMOS I/O: 29, N-ch open-drain I/O: 1) • μPD78F9831 Total: 38 (CMOS I/O: 37, N-ch open-drain I/O: 1)
Pull-up resistor	• μPD789830 Total: 15 (software control: 15) • μPD78F9831 Total: 21 (software control: 21)

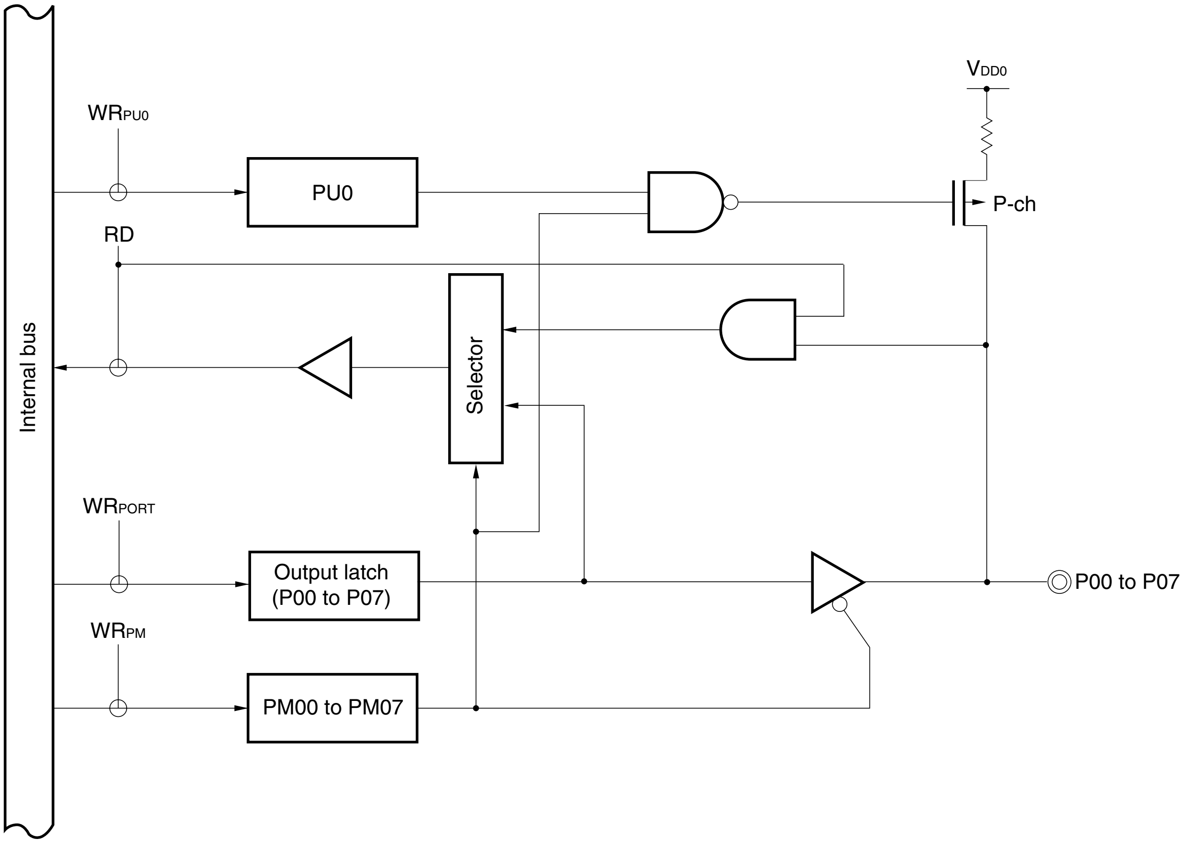
4.2.1 Port 0

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

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

Figure 4-3 shows the block diagram of port 0.

Figure 4-3. Block Diagram of P00 to P07



PU0: Pull-up resistor option register 0

PM: Port mode register

RD: Port 0 read signal

WR: Port 0 write signal

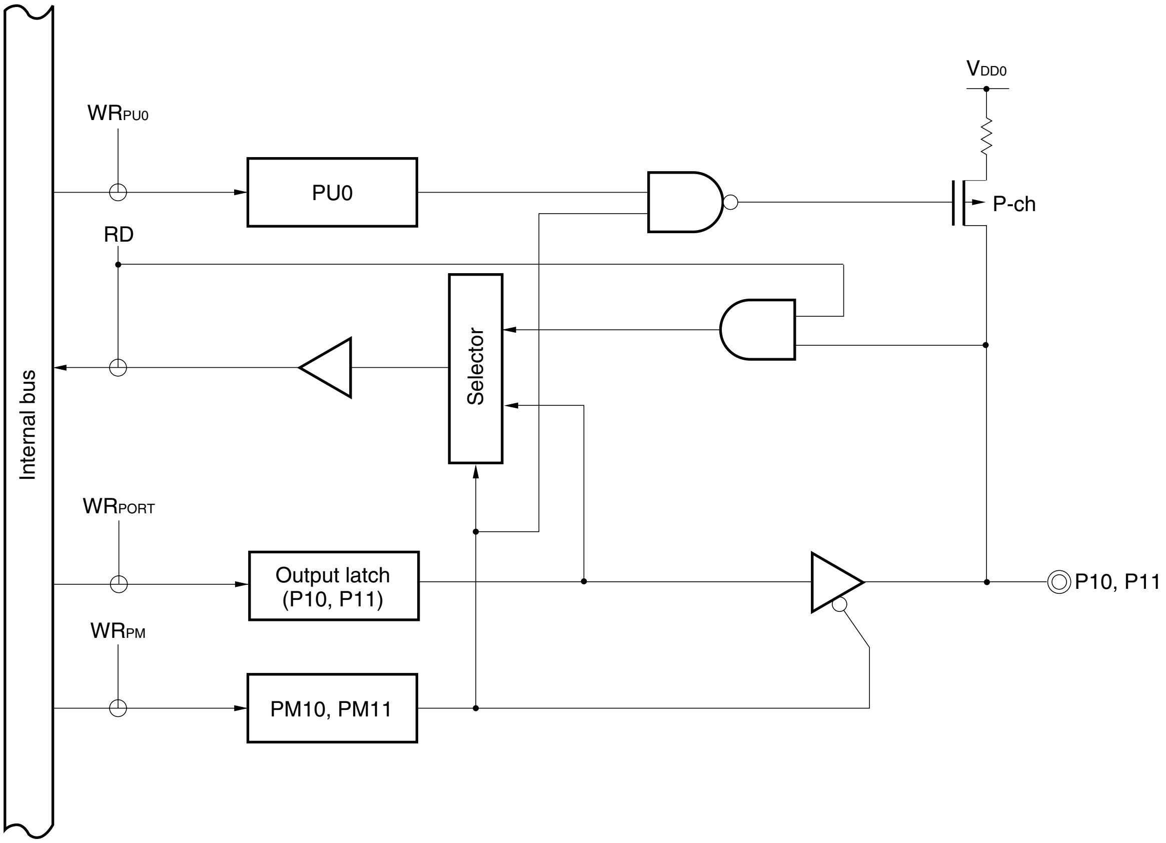
4.2.2 Port 1: μPD789830

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

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

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

Figure 4-4. Block Diagram of P10 and P11 (μPD789830)



PU0: Pull-up resistor option register 0
PM: Port mode register
RD: Port 1 read signal
WR: Port 1 write signal

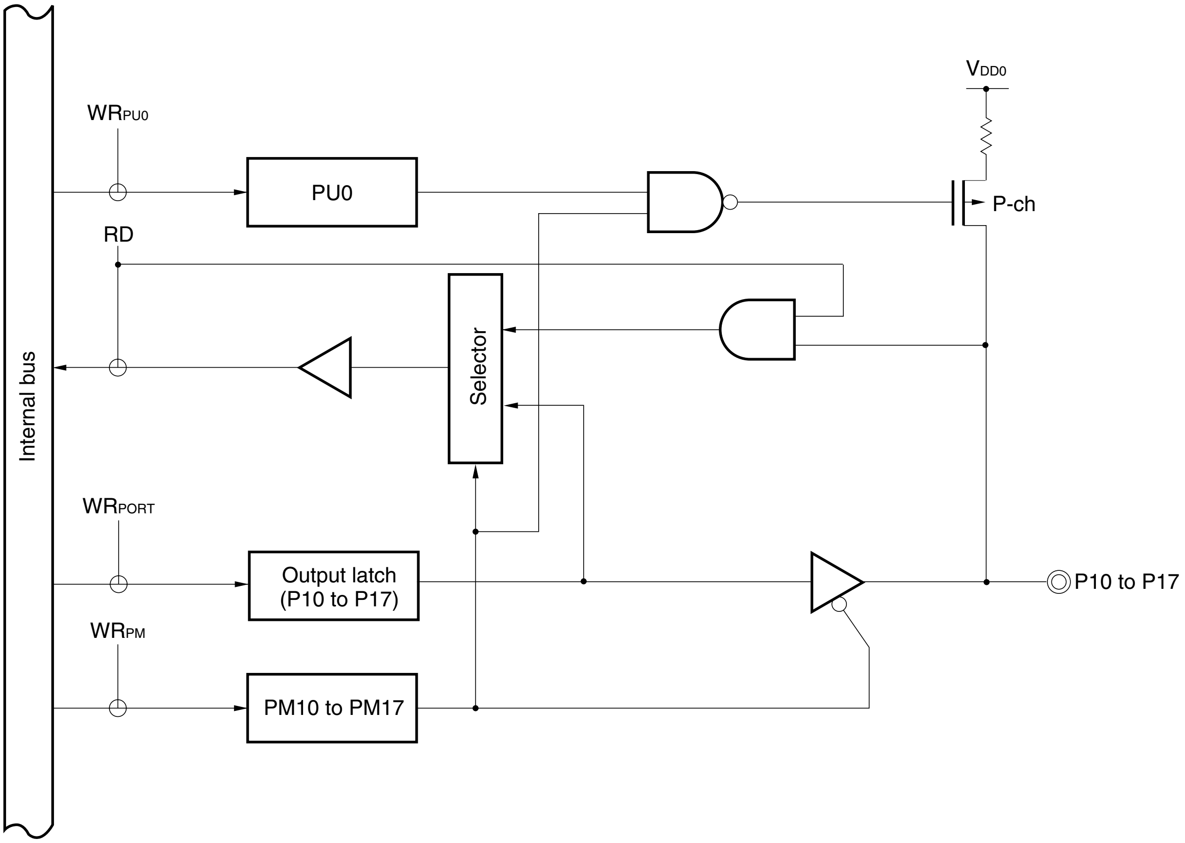
4.2.3 Port 1: μPD78F9831

This is an 8-bit I/O port with an output latch. Port 1 can be specified in input or output mode in 1-bit units by using port mode register 1 (PM1). When P10 to P17 pins are used as input port pins, on-chip pull-up resistors can be connected in 8-bit units by setting pull-up resistor option register 0 (PU0).

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

Figure 4-5 shows the block diagram of port 1.

Figure 4-5. Block Diagram of P10 to P17 (μPD78F9831)



PU0: Pull-up resistor option register 0

PM: Port mode register

RD: Port 1 read signal

WR: Port 1 write signal

4.2.4 Port 2

This is a 7-bit I/O port with an output latch. Port 2 can be specified in input or output mode in 1-bit units by using port mode register 2 (PM2).

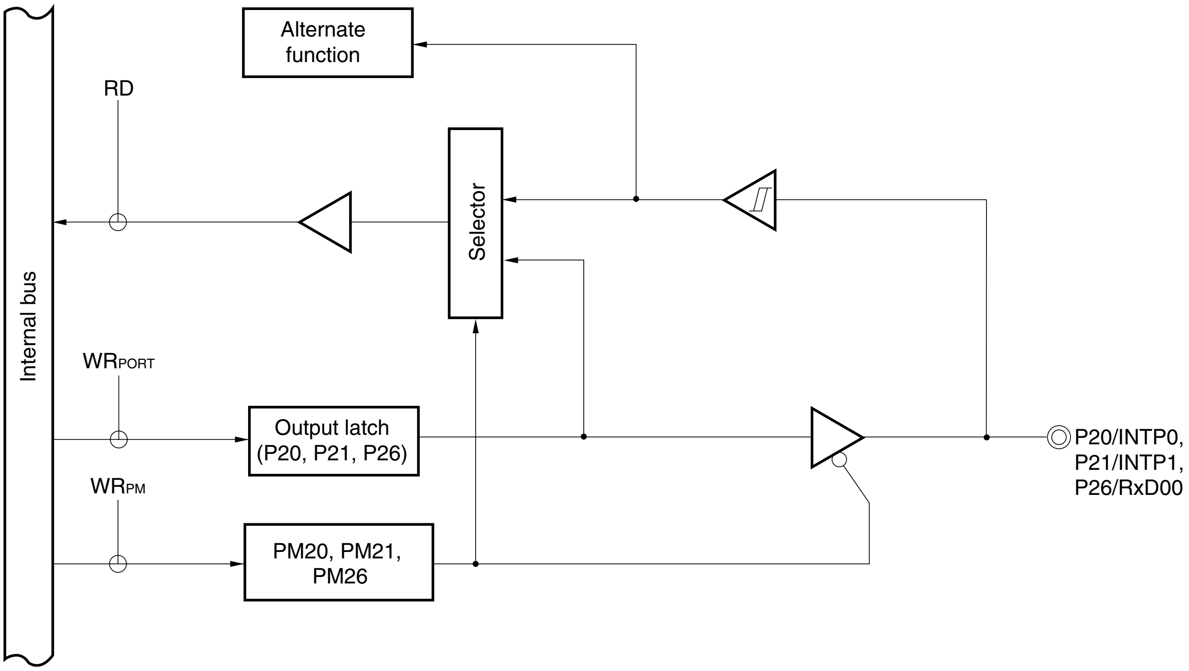
P24 is an N-ch open-drain I/O port. This port is pulled up to V_{DD0} when it is read.

The port is also used as an external interrupt input, pulse output, clock output, and a data I/O to and from the asynchronous serial interface.

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

Figures 4-6 to 4-9 show block diagrams of port 2.

Figure 4-6. Block Diagram of P20, P21, and P26



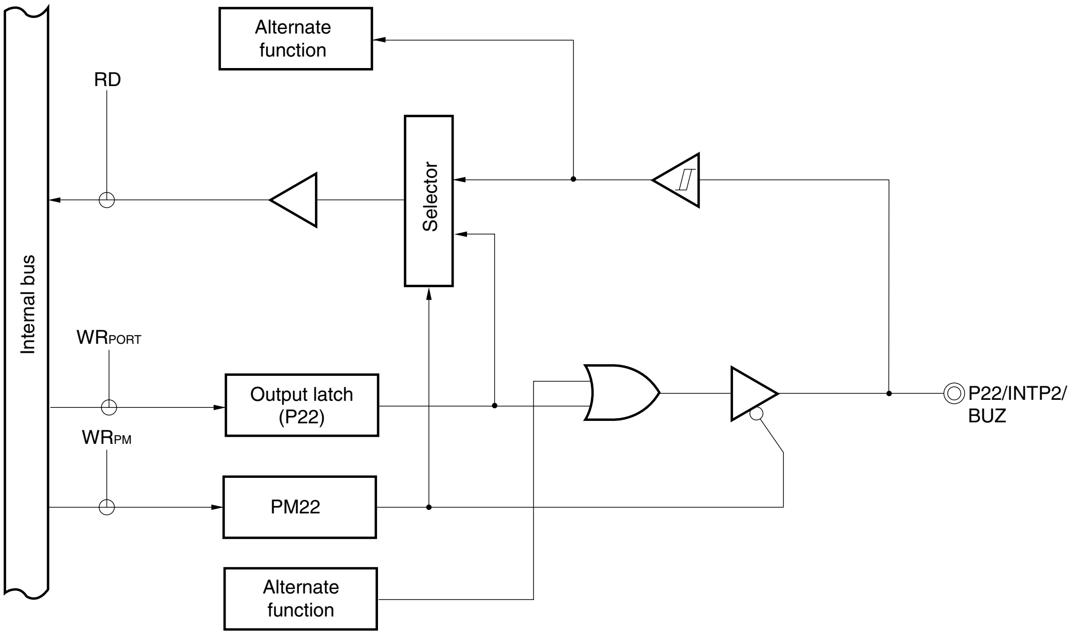
PU0: Pull-up resistor option register 0

PM: Port mode register

RD: Port 2 read signal

WR: Port 2 write signal

Figure 4-7. Block Diagram of P22

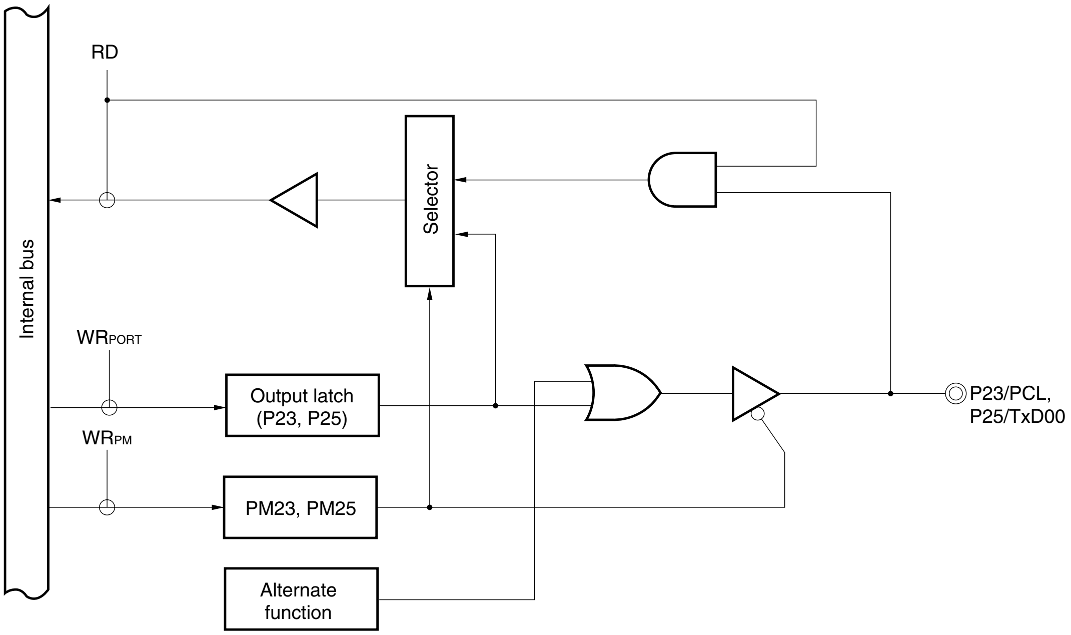


PM: Port mode register

RD: Port 2 read signal

WR: Port 2 write signal

Figure 4-8. Block Diagram of P23 and P25

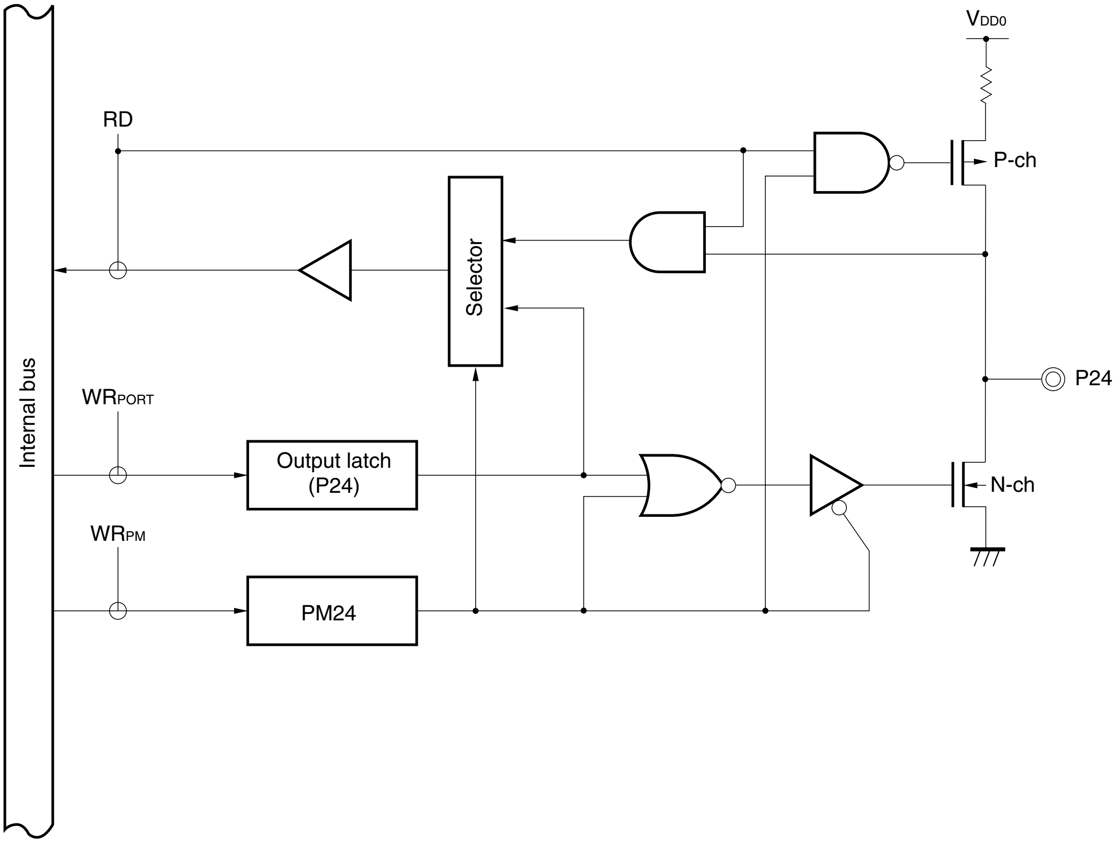


PM: Port mode register

RD: Port 2 read signal

WR: Port 2 write signal

Figure 4-9. Block Diagram of P24



PM: Port mode register
RD: Port 2 read signal
WR: Port 2 write signal

4.2.5 Port 3

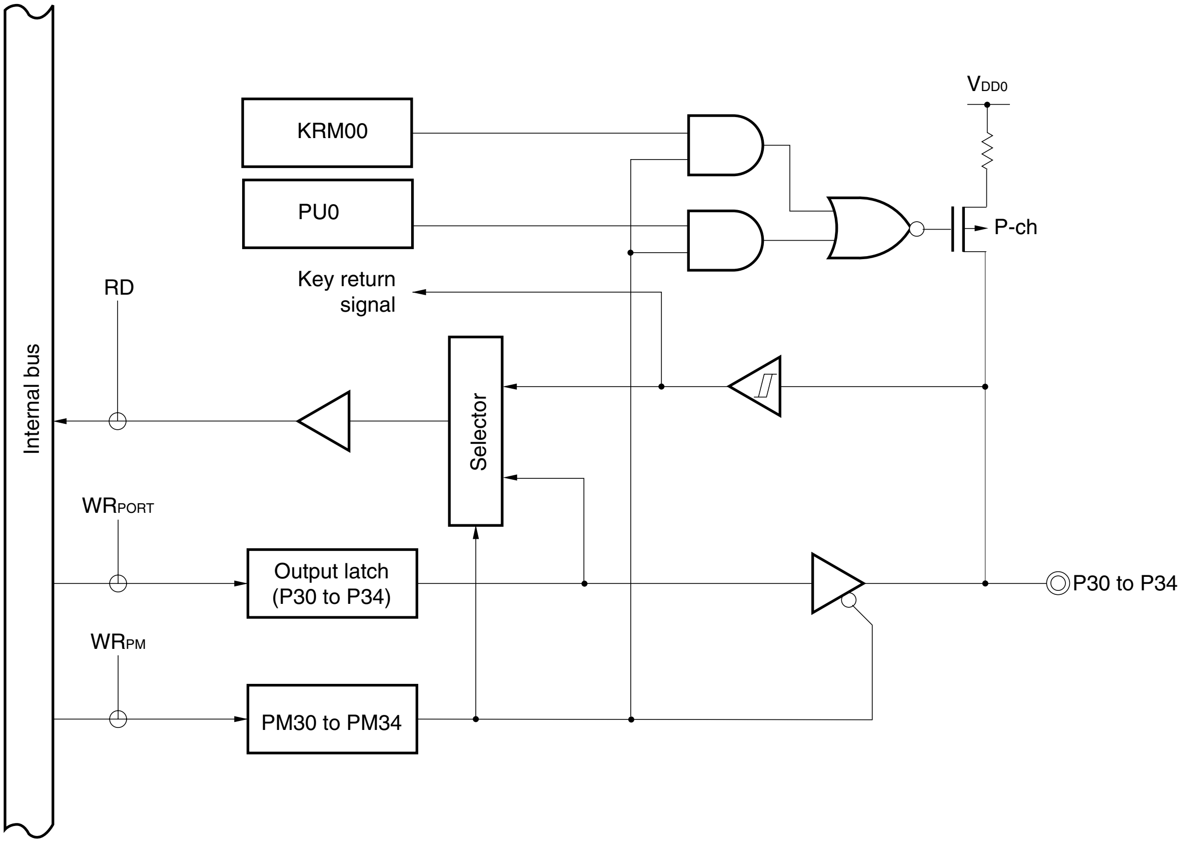
This is a 5-bit I/O port with an output latch. Port 3 can be specified in input or output mode in 1-bit units by using port mode register 3 (PM3).

This port can be also used to detect a key return signal in 1-bit units. For how to set the ports, see **(6)** in **Section 13.3**.

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

Figure 4-10 shows the block diagram of port 3.

Figure 4-10. Block Diagram of P30 to P34



PU0: Pull-up resistor option register 0
KRM00: Key return mode register 00
PM: Port mode register
RD: Port 3 read signal
WR: Port 3 write signal

4.2.6 Port 4: μPD78F9831

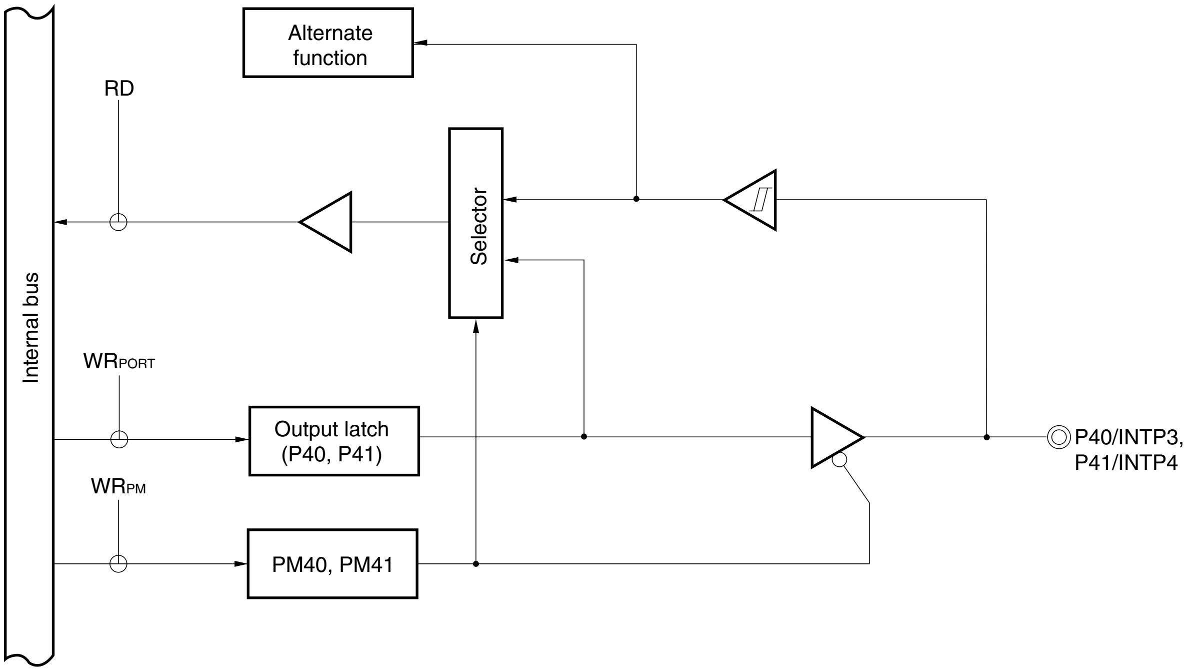
This is a 2-bit I/O port with an output latch. Port 4 can be specified in input or output mode in 1-bit units by using port mode register 4 (PM4).

The port is also used as an external interrupt input.

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

Figure 4-11 shows the block diagram of port 4.

Figure 4-11. Block Diagram of P40 and P41 (μPD78F9831)



PM: Port mode register

RD: Port 4 read signal

WR: Port 4 write signal

4.2.7 Port 5

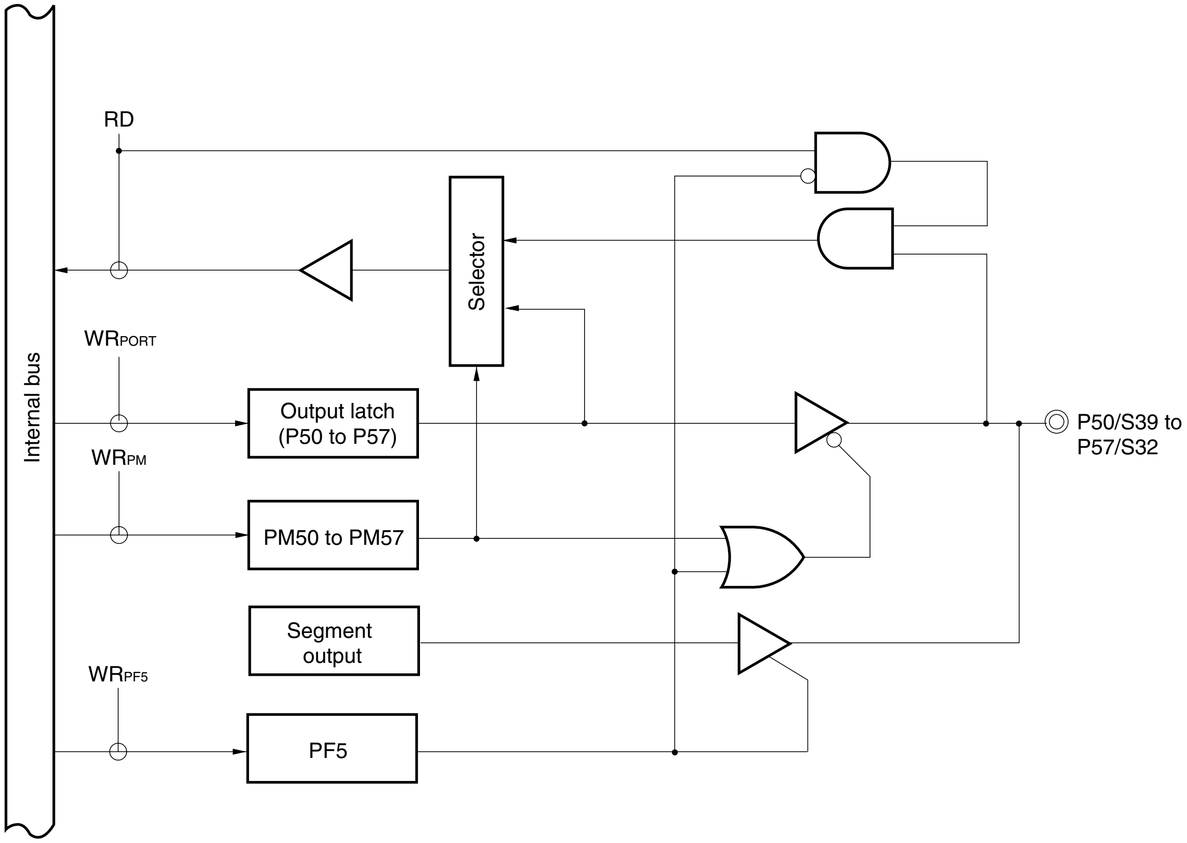
This is an 8-bit I/O port with an output latch. Port 5 can be specified in input or output mode in 1-bit units by using port mode register 5 (PM5).

The port is also used as a segment output.

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

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

Figure 4-12. Block Diagram of P50 to P57



PF5: Alternate port function switching register

PM: Port mode register

RD: Port 5 read signal

WR: Port 5 write signal

4.3 Port Function Control Registers

The following two types of registers are used to control the ports.

- Port mode registers (PM0 to PM5[Note])
- Pull-up resistor option register 0 (PU0)

(1) Port mode registers (PM0 to PM5[Note])

The port mode registers separately specify each port bit as being for input or output.

Each port mode register is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input writes FFH into the port mode registers.

When port pins are used for alternate functions, the corresponding port mode register and output latch must be set or reset as described in Table 4-4.

Caution When ports 2 and 4[Note] are acting as an output port, and its output level is changed, an interrupt request flag is set, because this port is also used as the input for an external interrupt. To use ports 2 and 4 in output mode, therefore, the interrupt mask flag must be set to 1 in advance.

Note PM4 and port 4 are provided to the μPD78F9831 only.

Table 4-4. Port Mode Register and Output Latch Settings for Using Alternate Functions

Pin Name	Alternate Function		PM××	P××
	Name	Input/Output		
P20	INTP0	Input	1	×
P21	INTP1	Input	1	×
P22	INTP2	Input	1	×
	BUZ	Output	0	0
P23	PCL	Output	0	0
P40[Note 1]	INTP3	Input	1	×
P41[Note 1]	INTP4	Input	1	×
P50 to P57	S39 to S32[Note 2]	Output	×	×

Notes 1. μPD78F9831 only

2. When using the alternate function, set the alternate port function switching register (PF5) to 1 (see **(2)** in **Section 12.3**).

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 (1) in Section 11.3.

Remark ×: Don't care

PM××: Port mode register

P××: Port output latch

Figure 4-13. Format of Port Mode Register (μPD789830)

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	1	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	1	1	1	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FF25H	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 3, 5 n = 0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

Figure 4-14. Format of Port Mode Register (μPD78F9831)

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	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM2	1	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	1	1	1	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
PM4	1	1	1	1	1	1	PM41	PM40	FF24H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FF25H	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 5 n = 0 to 7)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

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

This register specifies whether an on-chip pull-up resistor is connected to ports 0, 1, and 3.

On the port which is specified to use the on-chip pull-up resistor in PU0, the pull-up resistor can be internally used only for the bits set in input mode. No on-chip pull-up resistors can be used in the bits set in output mode in spite of setting PU0. On-chip pull-up resistors cannot be used even when the pins are used as the alternate-function output pins.

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

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

Figure 4-15. 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	PU03	0	PU01	PU00	FFF7H	00H	R/W

PU0m	Pm on-chip pull-up resistor selection (m = 0, 1, 3)
0	On-chip pull-up resistor not used
1	On-chip pull-up resistor used

Caution Bits 2, 4 to 7 must all be set to 0.

4.4 Operation of Port Functions

The operation of a port differs depending on whether the port is set in 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.

The 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.

The 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 input mode and not subject to manipulation become undefined.

4.4.2 Reading from I/O port

(1) In output mode

The contents of the 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.

The 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 input mode and not subject to manipulation become undefined

CHAPTER 5 CLOCK GENERATOR

5.1 Clock Generator Functions

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

- **Main system clock oscillator**
  This circuit oscillates at 2.0 to 5.0 MHz. Oscillation can be stopped by executing the STOP instruction or setting the processor clock control register (PCC).
- **Subsystem clock oscillator**
  This circuit oscillates at 32.768 kHz. Oscillation can be stopped by setting the suboscillation mode register (SCKM).

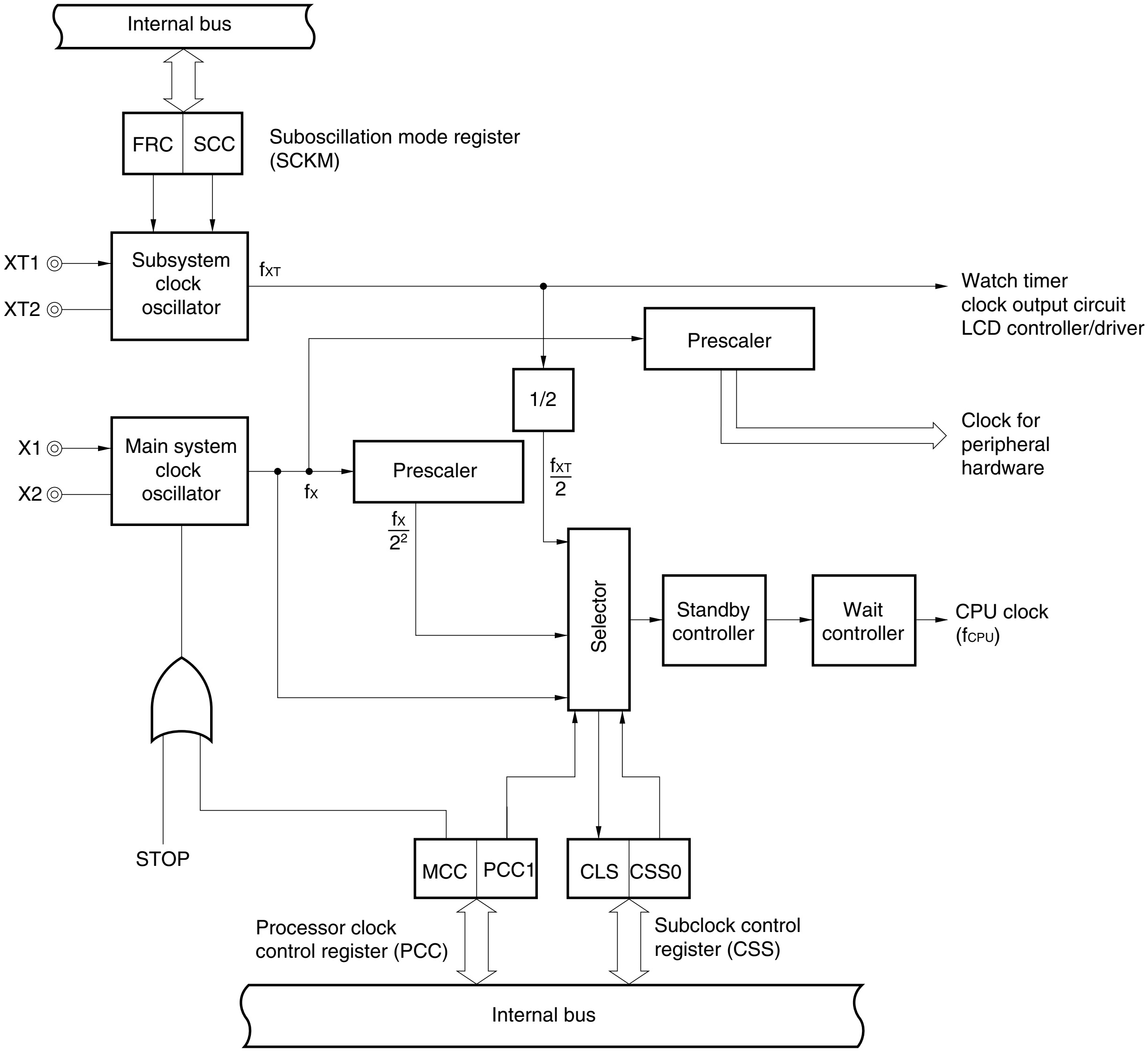
5.2 Clock Generator Configuration

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

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

Figure 5-1. Block Diagram of Clock Generator



5.3 Clock Generator Control Registers

The clock generator is controlled by the following register.

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

Figure 5-2. Format of Processor Clock Control Register

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

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

CSS0	PCC1	CPU clock (f_{CPU}) selection[Note]
0	0	f_X (0.28 μs)
0	1	$f_X/2^2$ (1.12 μs)
1	0	$f_{XT}/2$ (61 μs)
1	1	

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

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

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

3. If an external clock pulse is input, do not set MCC, because the X2 pin is pulled up to V_{DD0} or V_{DD1}.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. The parenthesized values apply to operation at f_X = 3.58 MHz or f_{XT} = 32.768 kHz.

4. Minimum instruction execution time: $2f_{CPU}$

- f_{CPU} = 0.28 μs: 0.56 μs
- f_{CPU} = 1.12 μs: 2.23 μs
- f_{CPU} = 61 μs: 122 μs

(2) Suboscillation mode register (SCKM)

SCKM selects a feedback resistor for the subsystem clock, and controls the oscillation of the clock.
SCKM is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears SCKM to 00H.

Figure 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 ooperation
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.

Cautions 1. Bits 2 to 7 must all be set to 0.

2. If an external clock pulse is input, do not set SCC, because the XT2 pin is pulled up to V_{DD0} or V_{DD1}.

(3) Subclock control register (CSS)

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

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

$\overline{\text{RESET}}$ input clears 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 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 all be set to 0.

5.4 System Clock Oscillators

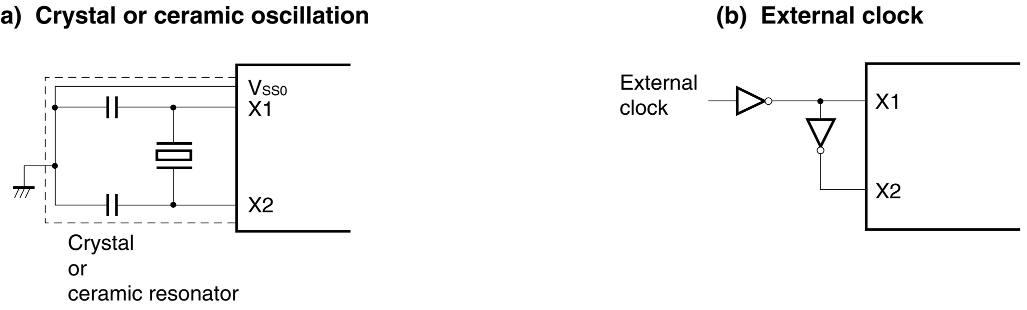
5.4.1 Main system clock oscillator

The main system clock oscillator is oscillated by the crystal or ceramic resonator (3.58 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 reversed 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



Cautions 1. If an external clock pulse is input, do not set the STOP instruction and MCC (bit 7 of the processor clock control register (PCC)) to 1, because doing so stops the main system clock, thus causing the X2 pin to be pulled up to V_{DD0} or V_{DD1}.

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

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with 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_{SS0}. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

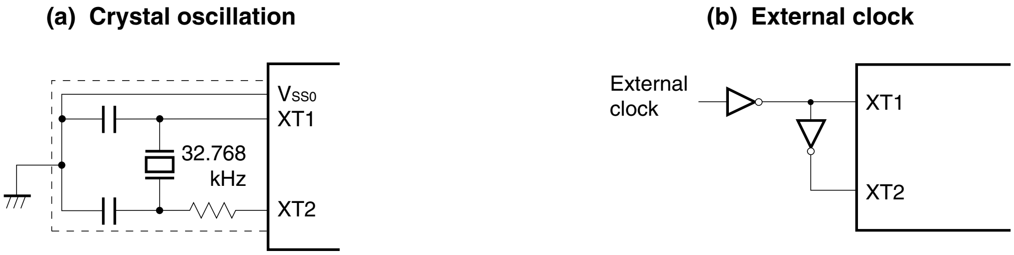
5.4.2 Subsystem clock oscillator

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

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

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

Figure 5-6. External Circuit of Subsystem Clock Oscillator



(a) Crystal oscillation

(b) External clock

Cautions 1. If an external clock pulse is input, do not set the SCC (bit 0 of the suboscillation mode register (SCKM)) to 1, because doing so causes the subsystem clock oscillator to stop operating, thus causing the XT2 pin to be pulled up to V_{DD0} or V_{DD1}.

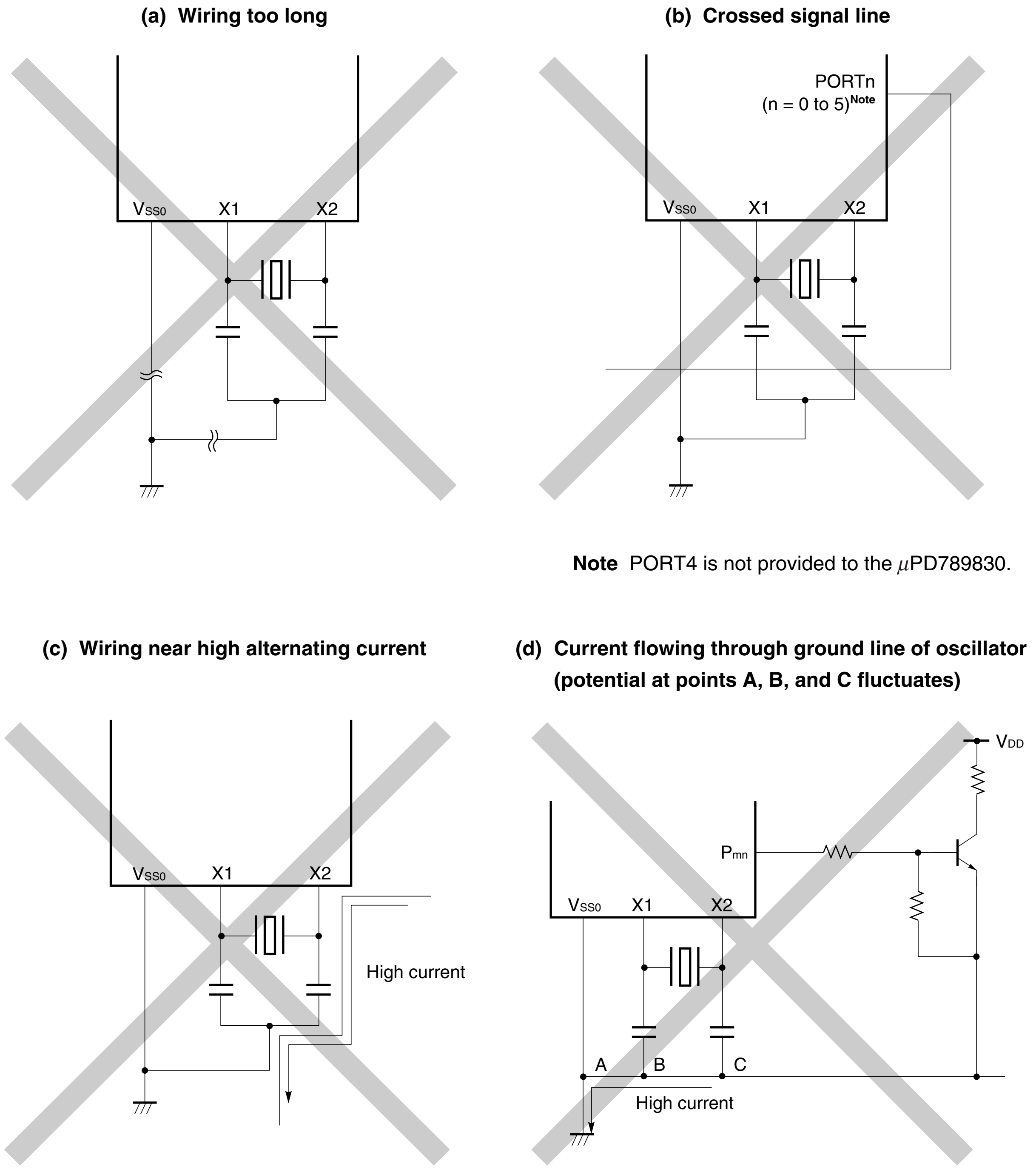
2. When using the main system clock oscillator and subsystem clock oscillator, wire as follows in the area enclosed by the broken lines in Figures 5-5 and 5-6 to avoid an adverse effect from wiring capacitance.

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with 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_{SS0}. 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, pay special attention because the subsystem clock oscillator has low amplification to minimize current consumption.

Figure 5-7 shows incorrect resonator connections.

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



(a) Wiring too long

(b) Crossed signal line

Note PORT4 is not provided to the μPD789830.

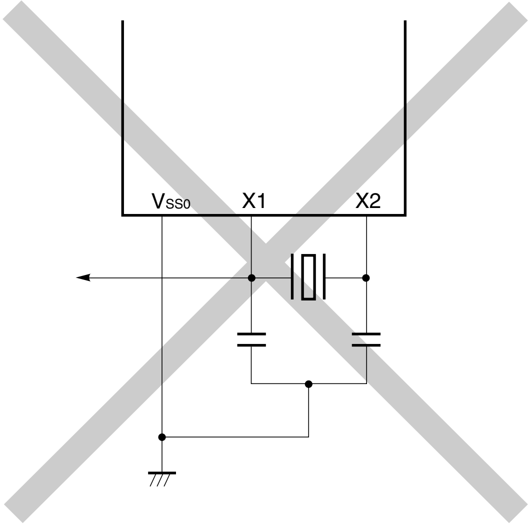
(c) Wiring near high alternating current

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

(e) Signal is extracted



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

5.4.3 Frequency divider

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

5.4.4 When no subsystem clocks are used

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

XT1: Connect to V_{SS0} or V_{SS1}
XT2: Open

In this state, however, some current may leak via the internal feedback resistor of the subsystem clock oscillator when the main system clock stops. To minimize the leakage current, the above internal feedback resistor can be removed by setting bit 1 (FRC) of the suboscillation mode register (SCKM). In this case, also connect the XT1 and XT2 pins as described above.

5.5 Clock Generator Operation

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

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

The operation 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 slow mode $2f_{CPU}$ (2.23 μs: at 3.58 MHz operation) of the main system clock is selected when the $\overline{\text{RESET}}$ signal is generated (PCC = 02H). While a low level is input to the $\overline{\text{RESET}}$ pin, oscillation of the main system clock is stopped.

(b) Three types of CPU clocks f_{CPU} (0.28 μs and 1.12 μs: main system clock (at 3.58 MHz operation), 61 μs: subsystem clock (at 32.768 kHz operation)) can be selected by the PCC, SCKM, and CSS settings.

(c) Two standby modes, STOP and HALT, can be used with the main system clock selected. In a system where no subsystem clock is used, setting bit 1 (FRC) of SCKM so that the on-chip feedback resistor cannot be used reduces current drain during 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 power dissipation operation is used (at 122 μs, 32.768 kHz operation).

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

(f) The clock pulse for the peripheral hardware is generated by dividing the frequency of the main system clock. The subsystem clock pulse is supplied to the clock output circuit, LCD controller/driver, and watch timer only. So, also at standby, the clock output circuit, LCD controller/driver, and clock function can keep running. The other hardware stops when the main system clock stops, because it runs based on the main system clock (except for external clock pulses).

5.6 Changing Setting of System Clock and CPU Clock

5.6.1 Time required for switching between system clock and CPU clock

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

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

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

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 (219 clocks)	
	1	2 clocks				$f_X/2f_{XT}$ clocks (55 clocks)	
1	×	2 clocks		2 clocks			

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

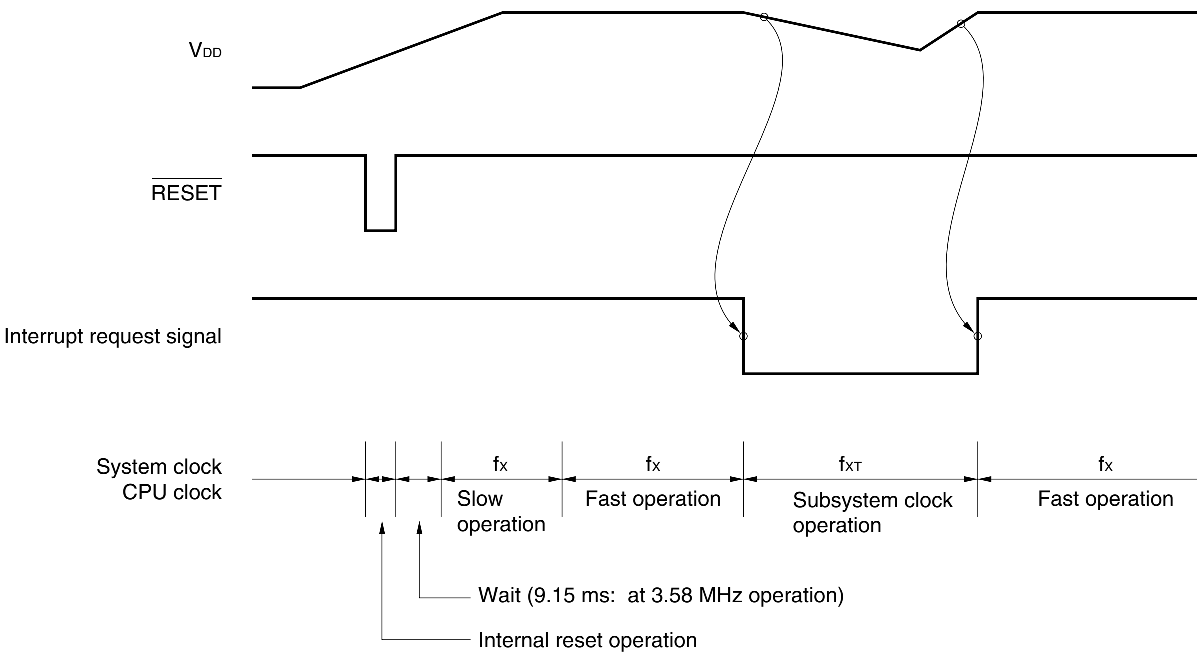
2. The parenthesized values apply to operation at f_X = 3.58 MHz or f_{XT} = 32.768 kHz.

3. ×: Don't care

5.6.2 Switching between system clock and CPU clock

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

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



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

After that, the CPU starts instruction execution at the slow speed of the main system clock (2.23 μs: at 3.58 MHz operation).

<2> After the time required for the V_{DD} voltage to rise to the level at which the CPU can operate at the 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 the 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 stabilized 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 subsystem clock is operating, allow sufficient time for the oscillation to stabilize by coding the program before switching again from the subsystem clock to the main system clock.

6.1 16-Bit Timer 40 Functions

16-bit timer 40 has the following functions.

(1) Interval timer

This timer generates an interrupt (INTTM40) if the TM40 count matches the comparison value.

Table 6-1. Interval Time of 16-Bit Timer 40

Minimum Interval Time	Maximum Interval Time	Resolution
1/fx (0.28 μs)	2^{16}/fx (18.3 ms)	1/fx (0.28 μs)
2^{2}/fx (1.12 μs)	2^{18}/fx (73.2 ms)	2^{2}/fx (1.12 μs)
2^{5}/fx (8.94 μs)	2^{21}/fx (585.8 ms)	2^{5}/fx (8.94 μs)
2^{10}/fx (286.0 μs)	2^{26}/fx (18.7 s)	2^{10}/fx (286.0 μs)

Remarks 1. fx: Main system clock oscillation frequency

2. The parenthesized values apply to operation at fx = 3.58 MHz.

(2) Free-running timer

This timer generates an interrupt (INTTM41) upon the occurrence of a timer overflow.

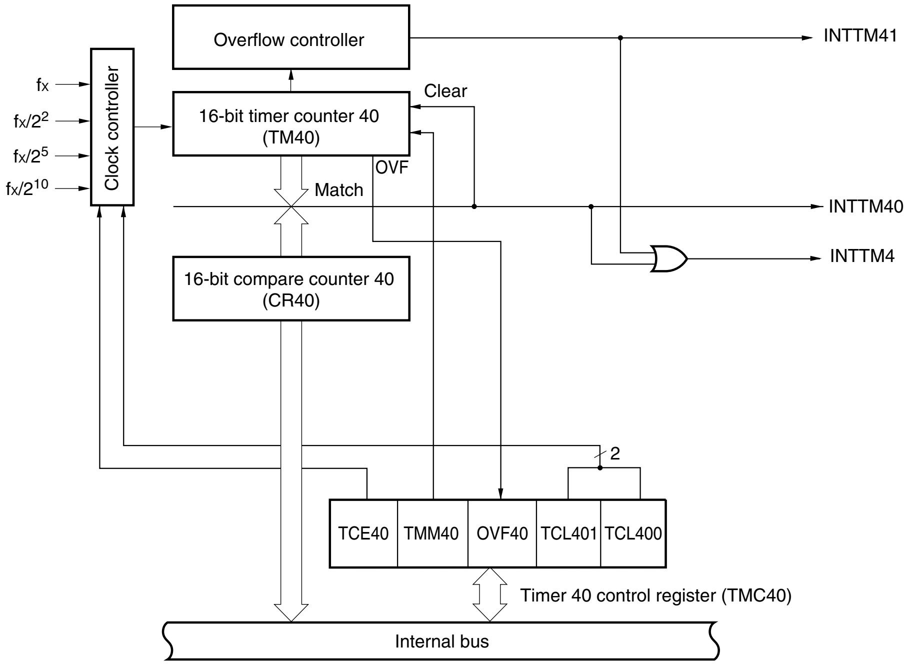
6.2 16-Bit Timer 40 Configuration

16-bit timer 40 includes the following hardware.

Table 6-2. 16-Bit Timer 40 Configuration

Item	Configuration
Timer counter	16 bits × 1 (TM40)
Register	Compare register: 16 bits × 1 (CR40)
Control register	Timer 40 control register (TMC40)

Figure 6-1. Block Diagram of 16-Bit Timer 40



(1) 16-bit compare register 40 (CR40)

A value specified in CR40 is compared with the count in 16-bit timer counter 40 (TM40). If they match, an interrupt request (INTTM40) is issued.

CR40 is set with a 16-bit memory manipulation instruction[Note]. Any value from 0000H to FFFFH can be set.

$\overline{\text{RESET}}$ input sets CR40 to 0000H.

Note For CR40, 16-bit access can be performed only in short direct addressing.

Cautions 1. Do not write to CR40 during the count operation; otherwise the timer counter may not operate normally.

2. When the 16-bit timer is used as a free-running timer, do not set CR40 to 0000H or FFFEH; otherwise noise may be superimposed on the interrupt request signal (INTTM4).

★ **3. If CR40 is overwritten during a count operation, prevent this from recurring by setting interrupt mask flag registers 0 and 1 (MK0 and MK1) to disable interrupts. If CR40 is overwritten while interrupts are enabled, an interrupt request may be issued at the point of overwrite.**

(2) 16-bit timer counter 40 (TM40)

TM40 is a 16-bit register used to count the number of pulses.

TM40 cannot be read from or written to.

TM40 is cleared to 0000H:

- when a $\overline{\text{RESET}}$ signal is input;
- when TCE40 (bit 7 of the timer 40 control register (TMC40)) is 0;
- immediately after the TM40 count matches the CR40 comparison value in clear and start mode (TMM40 (bit 6 of TMC40) = 0); or
- immediately after a TM40 overflow occurs in free-running mode (TMM40 = 1).

6.3 16-Bit Timer 40 Control Register

The following register is used to control 16-bit timer 40.

- Timer 40 control register (TMC40)

(1) Timer 40 control register (TMC40)

TMC40 controls the count clock and operation mode settings of 16-bit timer 40.
TMC40 is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears TMC40 to 00H.

Figure 6-2. Format of Timer 40 Control Register

Symbol	<7>	6	5	4	<3>	2	1	0	Address	After reset	R/W
TMC40	TCE40	TMM40	0	0	OVF40	0	TCL401	TCL400	FF5BH	00H	R/W[Note]

TCE40	16-bit timer 40 count operation control
0	TM40 count operation disabled (TM40 = 0000H)
1	TM40 count operation enabled

TMM40	16-bit timer 40 operation mode control
0	Clear (immediately after TM40 matches CR40) and start mode
1	Free-running mode

OVF40	Overflow status
0	A timer overflow occurs when OVF40 is 1.
1	A timer overflow occurs when OVF40 is 0.

TCL401	TCL400	16-bit timer 40 count clock selection
0	0	f_X (3.58 MHz)
0	1	$f_X/2^2$ (895 kHz)
1	0	$f_X/2^5$ (112 kHz)
1	1	$f_X/2^{10}$ (3.50 kHz)

Note Bit 3 is read-only.

Cautions 1. Bits 2, 4, and 5 must be fixed to 0.

2. Do not write to TMM40, TCL400, or TCL401 while the timer is operating. To write to these bits, first stop the TM40 count operation (TCE40 = 0).

Remarks 1. f_X : Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 3.58 MHz.

6.4 16-Bit Timer 40 Operation

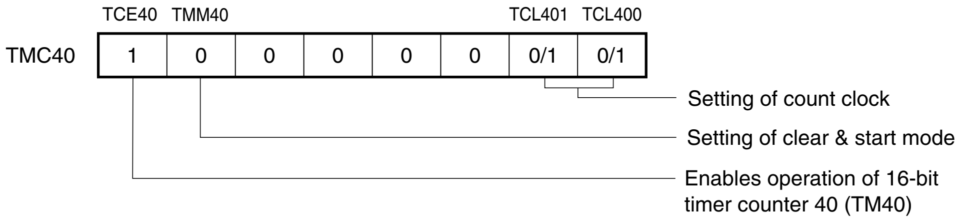
6.4.1 Operation as interval timer

The interval timer can repeatedly generate an interrupt at intervals specified by the preset count value of 16-bit compare register 40 (CR40).

To operate 16-bit timer 40 as an interval timer, the following settings are required.

- Set count values to CR40.
- Set timer 40 control register (TMC40) as shown in Figure 6-3.

Figure 6-3. Settings of Timer 40 Control Register at Interval Timer Operation


TCE40
TMM40
TCL401
TCL400
TMC40
1
0
0
0
0
0
0/1
0/1
Setting of count clock
Setting of clear & start mode
Enables operation of 16-bit timer counter 40 (TM40)

When the count value of 16-bit timer counter 40 (TM40) matches the set value of CR40, the value of TM40 is cleared to 0000H, TM40 continues counting, and an interrupt request signal (INTTM40) is generated.

Table 6-3 shows interval time, and Figure 6-4 shows timing of interval timer operation.

Cautions 1. When using 16-bit timer 40 as an interval timer, set 16-bit timer 40 to clear & start mode (TMM40 = 0).

★ **2. Be sure to execute the following processing when rewriting the value in CR40 during a count operation.**

• Set interrupt disabled (set TMMK40 (bit 7 of interrupt mask flag register 0 (MK0)) to 1).

If the value in CR40 is rewritten in the interrupt-enabled state, an interrupt request may occur at the moment of rewrite.

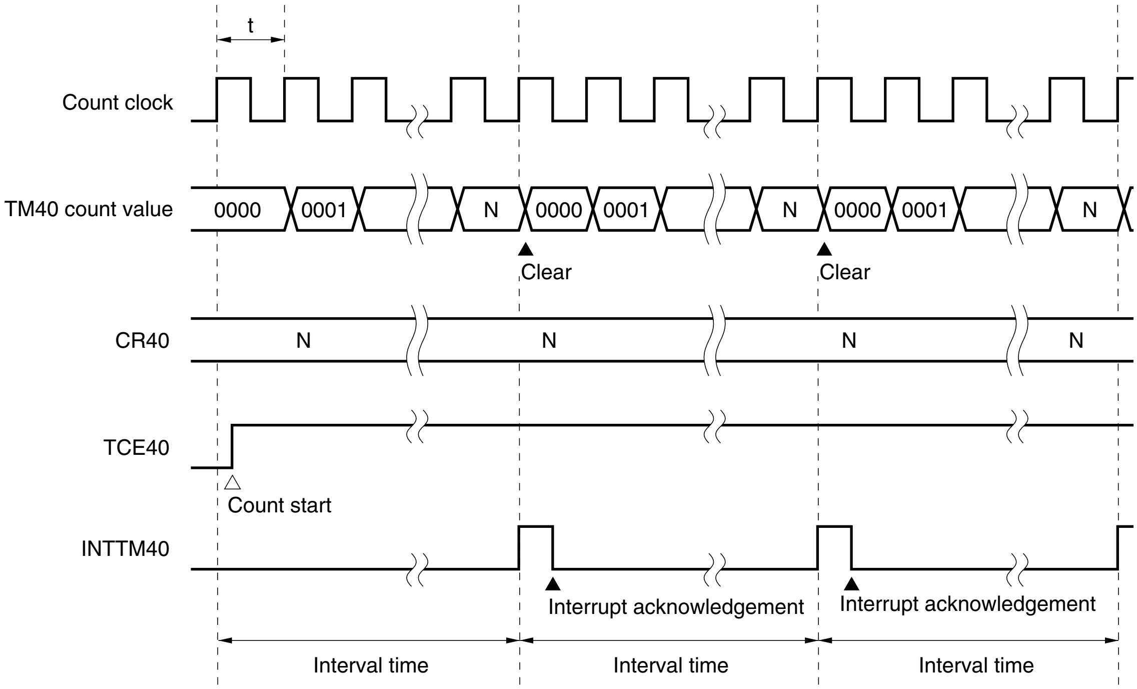
Table 6-3. Interval Time of 16-Bit Timer 40

TCL400	TCL401	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	1/f_X (0.28 μs)	2^{16}/f_X (18.3 ms)	1/f_X (0.28 μs)
0	1	2^{2}/f_X (1.12 μs)	2^{18}/f_X (73.2 ms)	2^{2}/f_X (1.12 μs)
1	0	2^{5}/f_X (8.94 μs)	2^{21}/f_X (586 ms)	2^{5}/f_X (8.94 μs)
1	1	2^{10}/f_X (286 μs)	2^{26}/f_X (18.7 s)	2^{10}/f_X (286 μs)

Remarks 1. f_X : Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 3.58 MHz.

Figure 6-4. Operating Timing of 16-Bit Timer 40 Used as Interval Timer



Remark Interval time = (N + 1) × t: N = 0000H to FFFFH

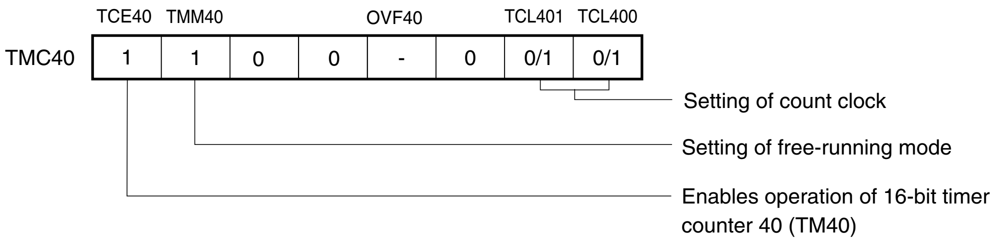
6.4.2 Operation as free-running timer

The free-running timer can repeatedly generate interrupts at the value set to 16-bit compare register 40 (CR40) in advance based on the intervals of the value set in TCL400 and TCL401.

To operate 16-bit timer 40 as a free-running timer, the following settings are required.

- Set the count value to CR40
- Set timer 40 control register (TMC40) as shown in Figure 6-5

Figure 6-5. Settings of Timer 40 Control Register at Free-Running Timer Operation



	TCE40	TMM40			OVF40		TCL401	TCL400
TMC40	1	1	0	0	-	0	0/1	0/1

Setting of count clock
Setting of free-running mode
Enables operation of 16-bit timer counter 40 (TM40)

When the count value of 16-bit timer counter 40 (TM40) matches the set value of CR40, TM40 continues counting, and an interrupt request signal (INTTM40) is generated. If TM40 continues counting and an overflow occurs as a result, another interrupt request signal (INTTM41) is generated. In addition, an interrupt request signal (INTTM4) is generated as a result of a logical OR between INTTM40 and INTTM41.

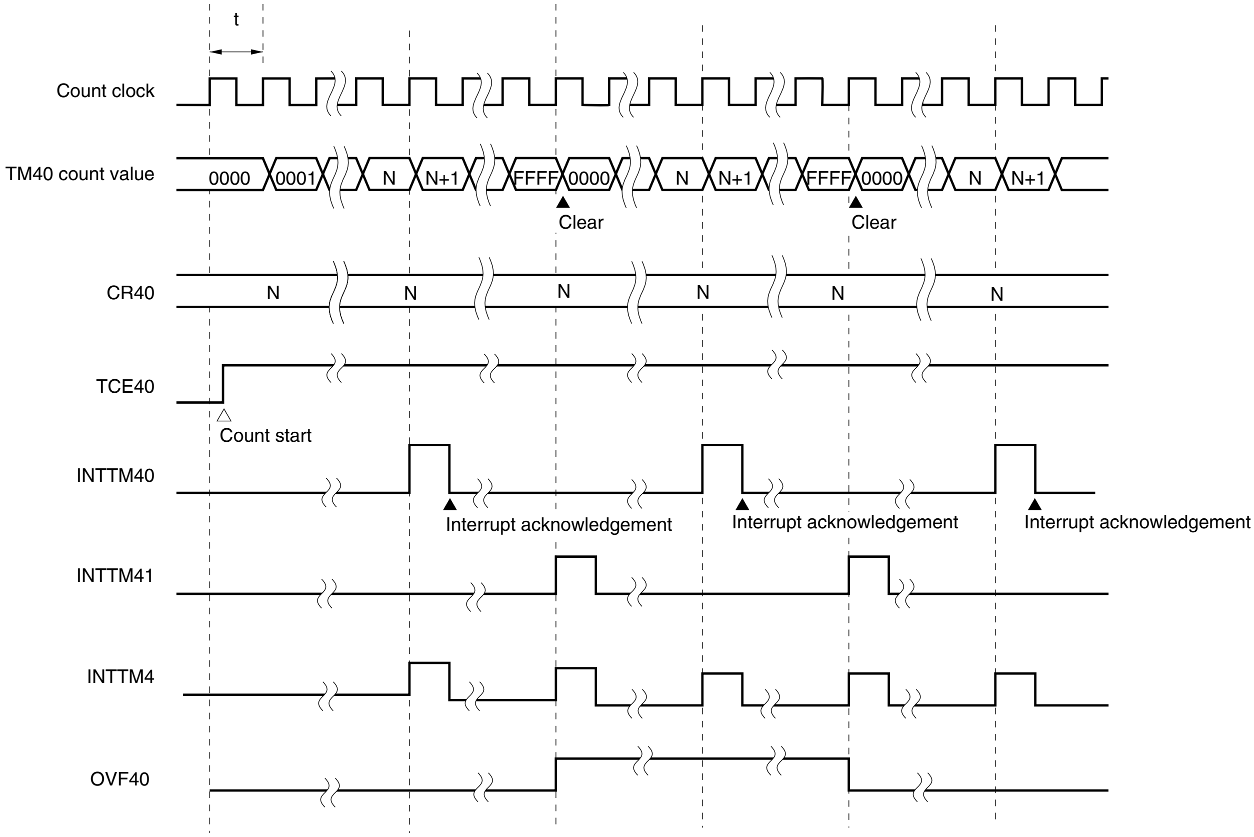
Cautions 1. When using 16-bit timer 40 as a free-running timer, set 16-bit timer 40 to free-running mode (TMM40 = 1).

2. When CR40 is set to FFFFH, the interrupt request signal (INTTM40) cannot be generated.

3. When 16-bit timer 40 is used as a free-running timer, do not set CR40 to 0000H or FFFEH; otherwise noise may be superimposed on the interrupt request signal (INTTM4).

Figure 6-6 shows the operating timing of the free-running timer.

Figure 6-6. Operating Timing of 16-Bit Timer 40 Used as Free-Running Timer

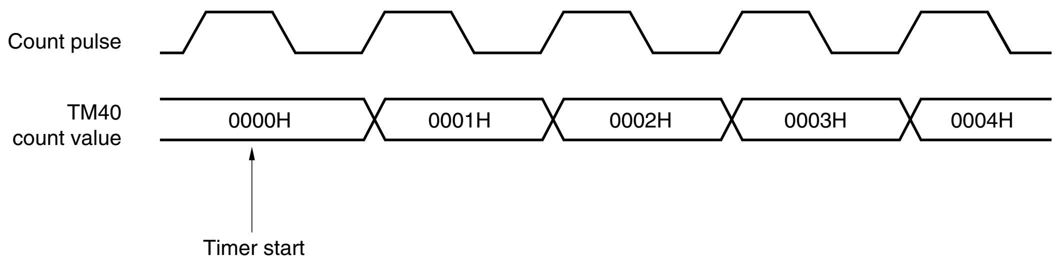


6.5 Notes on Using 16-Bit Timer 40

(1) Error on starting timer

An error of up to 1 clock occurs after the timer has been started until a match signal is generated. This is because 16-bit timer counter 40 (TM40) is started asynchronously to the count pulse.

Figure 6-7. Start Timing of 16-Bit Timer Counter 40



CHAPTER 7 8-BIT TIMER 00

7.1 8-Bit Timer 00 Functions

8-bit timer 00 has the following function.

★ **(1) Interval timer**

When 8-bit timer 00 is used as an interval timer, it generates an interrupt at any time intervals set in advance.

Table 7-1. Interval Time of 8-Bit Timer 00

Minimum Interval Time	Maximum Interval Time	Resolution
1/fx (0.28 μs)	2^8/fx (71.5 μs)	1/fx (0.28 μs)
2^5/fx (8.94 μs)	2^{13}/fx (2.23 ms)	2^5/fx (8.94 μs)

Remarks 1. fx: Main system clock oscillation frequency

2. The parenthesized values apply to operation at fx = 3.58 MHz.

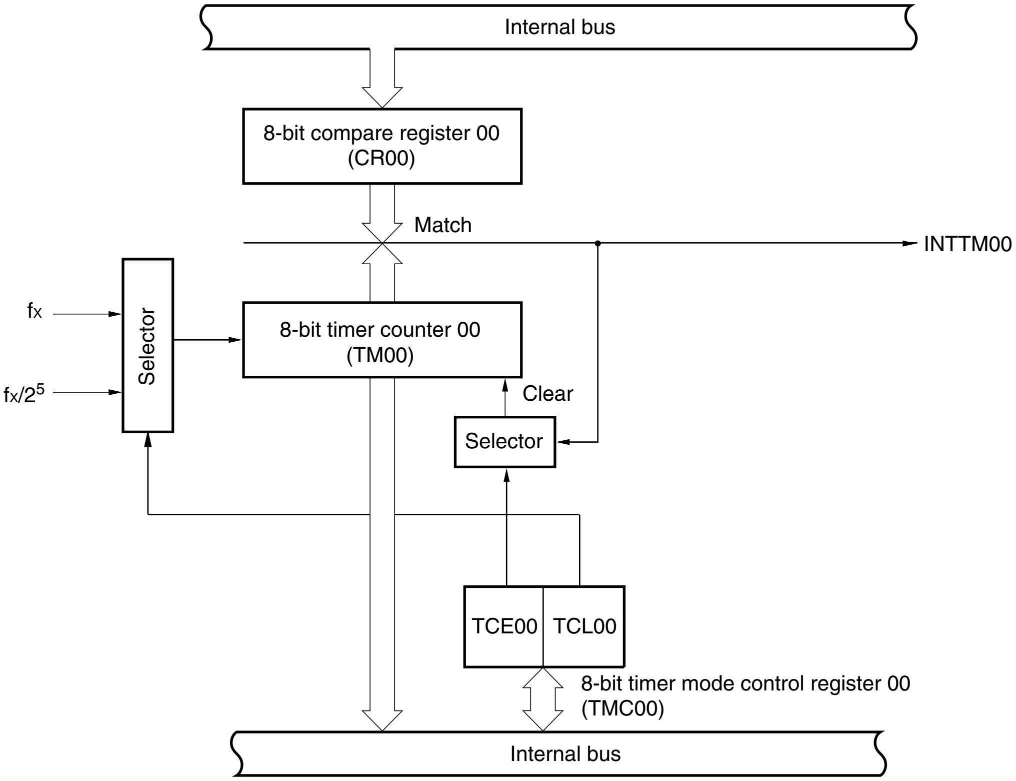
7.2 8-Bit Timer 00 Configuration

8-bit timer 00 includes the following hardware.

Table 7-2. Configuration of 8-Bit Timer 00

Item	Configuration
Timer counter	8 bits × 1 (TM00)
Register	Compare register: 8 bits × 1 (CR00)
Control register	8-bit timer control register 00 (TMC00)

Figure 7-1. Block Diagram of 8-Bit Timer 00



(1) 8-bit compare register 00 (CR00)

This is an 8-bit register to compare the value set to CR00 with 8-bit timer counter 00 (TM00) count value, and if they match, generates an interrupt request (INTTM00).

CR00 is set with an 8-bit memory manipulation instruction. The 00H to FFH values can be set.

$\overline{\text{RESET}}$ input makes CR00 undefined.

Caution Stop the timer operation before rewriting CR00; otherwise the match interrupt request signal may be generated immediately.

(2) 8-bit timer counter 00 (TM00)

This is an 8-bit register to count pulses.

TM00 is read with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TM00 to 00H.

7.3 8-Bit Timer 00 Control Registers

The following register is used to control 8-bit timer 00.

- 8-bit timer mode control register 00 (TMC00)

★ **(1) 8-bit timer mode control register 00 (TMC00)**

This is a register used to enable or disable operation of 8-bit timer counter 00 (TM00) and set the count clock of 8-bit timer 00.

TMC00 is set with a 1-bit or 8-bit memory manipulation instruction.

RESET input clears TMC00 to 00H.

Figure 7-2. 8-Bit Timer Mode Control Register 00 Format

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
TMC00	TCE00	0	0	0	0	0	TCL00	0	FF53H	00H	R/W

TCE00	8-bit timer 00 count operation control
0	Operation disabled (TM00 is cleared to 0.)
1	Operation enabled

TCL00	8-bit timer 00 operation mode control
0	f_X (3.58 MHz)
1	$f_X/2^5$ (112 kHz)

Caution Always stop the timer before setting TMC00.

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 3.58 MHz.

7.4 8-Bit Timer 00 Operation

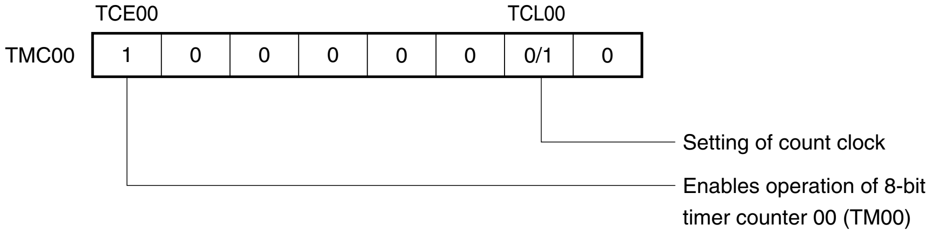
7.4.1 Operation as interval timer

The interval timer repeatedly generates an interrupt at time intervals specified by the count value set to 8-bit compare register 00 (CR00) in advance.

To operate 8-bit timer 00 as an interval timer, the following settings are required.

- Set count values to CR00.
- Set 8-bit timer mode control register 00 (TMC00) as shown in Figure 7-3.

Figure 7-3. Settings of 8-Bit Timer Mode Control Register 00 in Interval Timer Operation



When the count value of 8-bit timer counter 00 (TM00) matches the value set to CR00, the value of TM00 is cleared to 00H and TM00 continues counting. At the same time, an interrupt request signal (INTTM00) is generated.

Table 7-3 shows interval time, and Figure 7-4 shows the timing of interval timer operation.

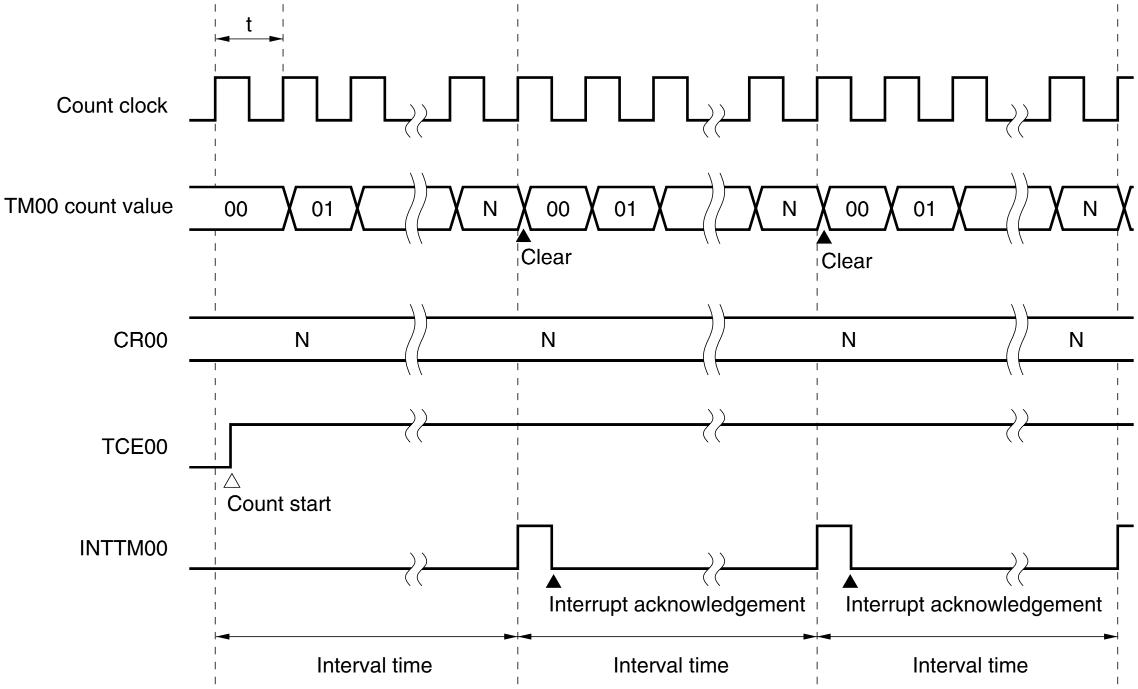
Table 7-3. Interval Time of 8-Bit Timer 00

TCL00	Minimum Interval Time	Maximum Interval Time	Resolution
0	1/fx (0.28 μs)	2^8/fx (71.5 μs)	1/fx (0.28 μs)
1	2^5/fx (8.94 μs)	2^{13}/fx (2.23 ms)	2^5/fx (8.94 μs)

Remarks 1. fx: Main system clock oscillation frequency

2. The parenthesized values apply to operation at fx = 3.58 MHz.

Figure 7-4. Operating Timing of 8-Bit Timer 00 Used as Interval Timer



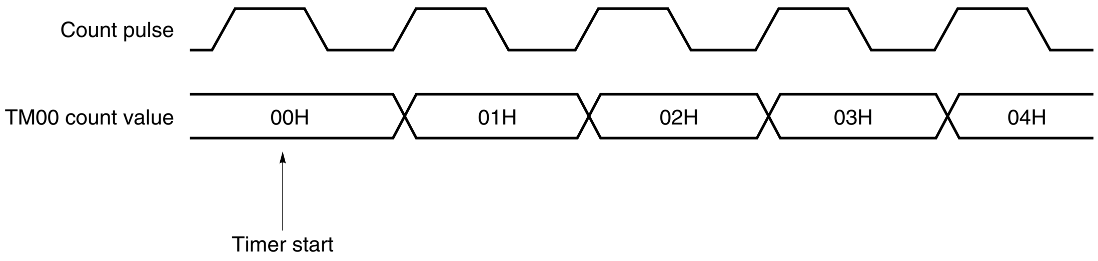
Remark Interval time = (N + 1) × t: N = 00H to FFH

7.5 Notes on Using 8-Bit Timer 00

(1) Error on starting timer

An error of up to 1 clock occurs after the timer has been started until a match signal is generated. This is because 8-bit timer counter 00 (TM00) is started asynchronously to the count pulse.

Figure 7-5. Start Timing of 8-Bit Timer Counter 00



8.1 Watch Timer Functions

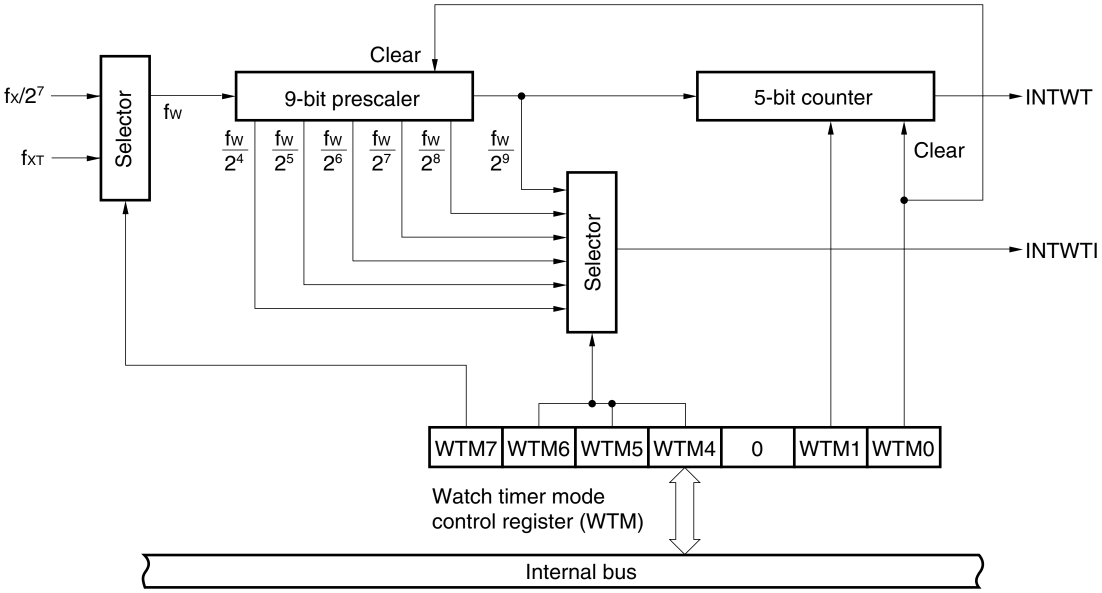
The watch timer has the following functions.

- Watch timer
- Interval timer

The watch and interval timers can be used at the same time.

Figure 8-1 is a block diagram of the watch timer.

Figure 8-1. Block Diagram of Watch Timer



(1) Watch timer

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

Caution When the main system clock is operating at 3.58 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 (INTWTI) at specified intervals.

Table 8-1. Interval Generated Using Interval Timer

Interval	At f_X = 3.58 MHz	At f_X = 4.19 MHz	At f_{XT} = 32.768 kHz
$2^4 \times 1/f_W$	572 μs	489 μs	488 μs
$2^5 \times 1/f_W$	1.14 ms	978 μs	977 μs
$2^6 \times 1/f_W$	2.29 ms	1.96 ms	1.95 ms
$2^7 \times 1/f_W$	4.58 ms	3.91 ms	3.91 ms
$2^8 \times 1/f_W$	9.15 ms	7.82 ms	7.81 ms
$2^9 \times 1/f_W$	18.3 ms	15.6 ms	15.6 ms

Remark f_W: Watch timer clock frequency ($f_X/2^7$ or f_{XT})
f_X: Main system clock oscillation frequency
f_{XT}: Subsystem clock oscillation frequency

8.2 Watch Timer Configuration

The watch timer includes the following hardware.

Table 8-2. Watch Timer Configuration

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

8.3 Watch Timer Control Register

The following register is used to control the watch timer.

- **Watch timer mode control register (WTM)**

  WTM selects a count clock for the watch timer and specifies whether to enable clocking of the timer. It also specifies the prescaler interval and how the 5-bit counter is controlled.

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

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

Figure 8-2. Format of Watch Timer Mode Control Register

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

WTM7	Watch timer count clock selection
0	$f_X/2^7$ (27.9 kHz)
1	f_{XT} (32.768 kHz)

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

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

WTM0	Watch timer operation
0	Operation disabled (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

3. f_{XT}: Subsystem clock oscillation frequency

4. The parenthesized values apply to operation at f_X = 3.58 MHz or f_{XT} = 32.768 kHz.

8.4 Watch Timer Operation

8.4.1 Operation as watch timer

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

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

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

Only the watch timer can be started form zero seconds by clearing WTM1 to 0 when the interval timer and watch timer operate at the same time. In this case, however, an error of up to $2^9 \times 1/f_W$ may occur in the overflow (INTWT) after the zero-second start of the watch timer because the 9-bit prescaler is not cleared to 0.

8.4.2 Operation as interval timer

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

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

Table 8-3. Interval Generated Using Interval Timer

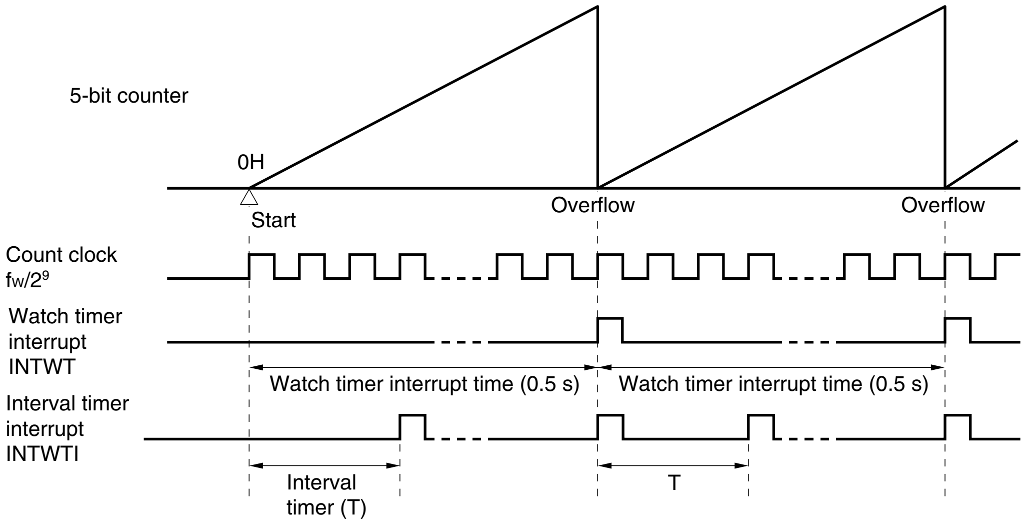
WTM6	WTM5	WTM4	Interval	At f_X = 3.58 MHz	At f_{XT} = 32.768 kHz
0	0	0	$2^4 \times 1/f_W$	572 μs	488 μs
0	0	1	$2^5 \times 1/f_W$	1.14 ms	977 μs
0	1	0	$2^6 \times 1/f_W$	2.29 ms	1.95 ms
0	1	1	$2^7 \times 1/f_W$	4.58 ms	3.91 ms
1	0	0	$2^8 \times 1/f_W$	9.15 ms	7.81 ms
1	0	1	$2^9 \times 1/f_W$	18.3 ms	15.6 ms
Other than above			Setting prohibited		

Remark f_X: Main system clock oscillation frequency

f_{XT}: Subsystem clock oscillation frequency

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

Figure 8-3. Watch Timer/Interval Timer Operation Timing



Remark fw: Watch timer clock frequency

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

CHAPTER 9 WATCHDOG TIMER

9.1 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 loops. When an inadvertent loop is detected, a non-maskable interrupt request or the $\overline{\text{RESET}}$ signal can be generated.

Table 9-1. Inadvertent Loop Detection Time of Watchdog Timer

Inadvertent Loop Detection Time	At fx = 3.58 MHz
$2^{11} \times 1/f_X$	572 μs
$2^{13} \times 1/f_X$	2.29 ms
$2^{15} \times 1/f_X$	9.15 ms
$2^{17} \times 1/f_X$	36.6 ms

fx: Main system clock oscillation frequency

(2) Interval timer

The interval timer generates an interrupt at a given interval set in advance.

Table 9-2. Interval Time

Interval	At fx = 3.58 MHz
$2^{11} \times 1/f_X$	572 μs
$2^{13} \times 1/f_X$	2.29 ms
$2^{15} \times 1/f_X$	9.15 ms
$2^{17} \times 1/f_X$	36.6 ms

fx: Main system clock oscillation frequency

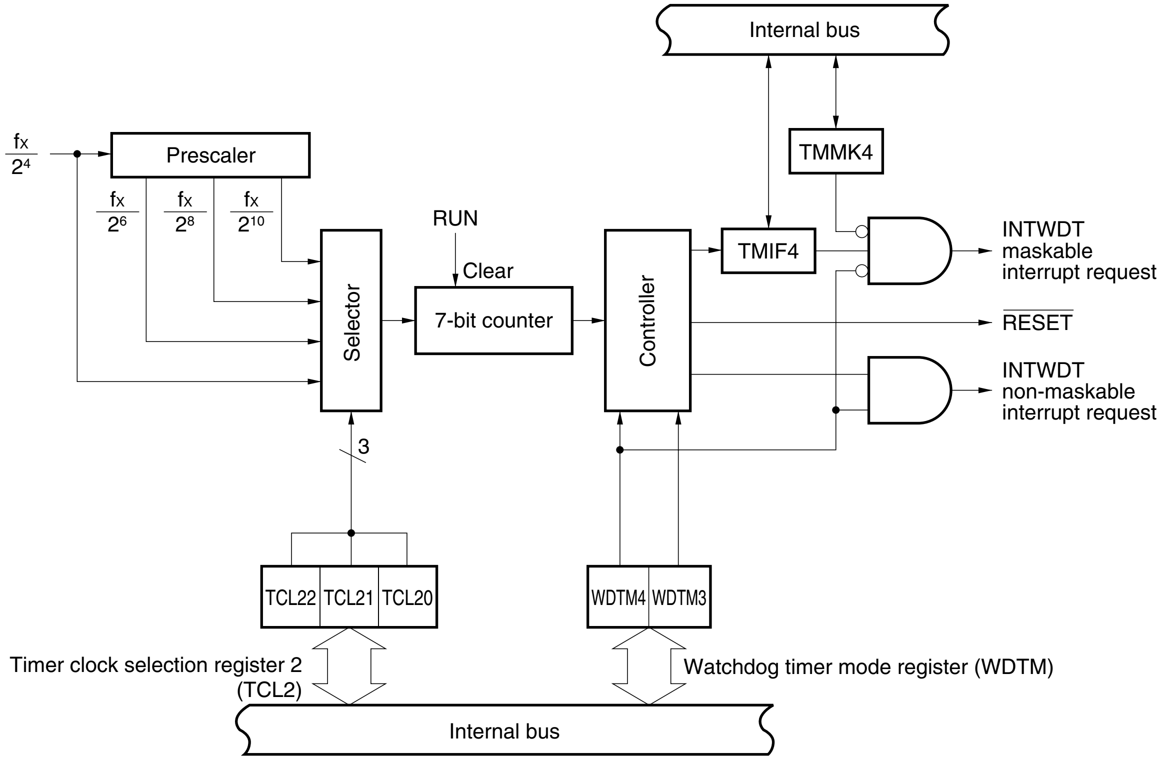
9.2 Watchdog Timer Configuration

The watchdog timer includes the following hardware.

Table 9-3. Configuration of Watchdog Timer

Item	Configuration
Control register	Timer clock selection register 2 (TCL2) Watchdog timer mode register (WDTM)

Figure 9-1. Block Diagram of Watchdog Timer



9.3 Watchdog Timer Control Registers

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

- Timer clock selection register 2 (TCL2)
- Watchdog timer mode register (WDTM)

(1) Timer clock selection register 2 (TCL2)

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

Figure 9-2. Format of Timer Clock Selection Register 2

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TCL2	0	0	0	0	0	TCL22	TCL21	TCL20	FF42H	00H	R/W

TCL22	TCL21	TCL20	Watchdog timer count clock selection	Interval
0	0	0	$f_X/2^4$ (223.8 kHz)	$2^{11}/f_X$ (572 μs)
0	1	0	$f_X/2^6$ (55.9 kHz)	$2^{13}/f_X$ (2.29 ms)
1	0	0	$f_X/2^8$ (14.0 kHz)	$2^{15}/f_X$ (9.15 ms)
1	1	0	$f_X/2^{10}$ (3.50 kHz)	$2^{17}/f_X$ (36.6 ms)
Other than above			Setting prohibited	

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 3.58 MHz.

(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 clears WDTM to 00H.

Figure 9-3. Format of Watchdog Timer Mode Register

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

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

WDTM4	WDTM3	Watchdog timer operation mode selection[Note 2]
0	0	Operation stop
0	1	Interval timer mode (overflow and maskable interrupt occur)[Note 3]
1	0	Watchdog timer mode 1 (overflow and non-maskable interrupt occur)
1	1	Watchdog timer mode 2 (overflow occurs and reset operation started)

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

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

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

Cautions 1. When the watchdog timer is cleared by setting 1 to RUN, the actual overflow time is up to 0.8% shorter than the time set by timer clock selection register 2 (TCL2).

2. In watchdog timer mode 1 or 2, set WDTM4 to 1 after confirming the WDTIF (bit 0 of interrupt request mask flag register 0 (IF0)) being set to 0. When watchdog timer mode 1 or 2 is selected under the condition where WDTIF is 1, a non-maskable interrupt occurs at the completion of rewriting.

9.4 Watchdog Timer Operation

9.4.1 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 loop detection time interval) of the watchdog timer can be selected by bits 0 to 2 (TCL20 to TCL22) of timer clock selection register 2 (TCL2). By setting bit 7 (RUN) of WDTM to 1, the watchdog timer is started. Set RUN to 1 within the set inadvertent 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 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 HALT mode, but stops in STOP mode. Therefore, set RUN to 1 before entering STOP mode to clear the watchdog timer, and then execute the STOP instruction.

Caution The actual inadvertent loop detection time may be up to 0.8% shorter than the set time.

Table 9-4. Inadvertent Loop Detection Time of Watchdog Timer

TCL22	TCL21	TCL20	Inadvertent Loop Detection Time	At fx = 3.58 MHz
0	0	0	$2^{11} \times 1/f_X$	572 μs
0	1	0	$2^{13} \times 1/f_X$	2.29 ms
1	0	0	$2^{15} \times 1/f_X$	9.15 ms
1	1	0	$2^{17} \times 1/f_X$	36.6 ms

fx: Main system clock oscillation frequency

9.4.2 Operation as interval timer

When bits 4 and 3 (WDTM4 and 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 intervals specified by a count value set in advance.

Select a count clock (or interval) by setting bits 0 to 2 (TCL20 to TCL22) of timer clock selection register 2 (TCL2). The watchdog timer starts operation as an interval timer when the RUN bit (bit 7 of WDTM) is set to 1.

In interval timer mode, the interrupt mask flag (WDTMK: bit 0 of interrupt mask flag register 0 (MK0)) is valid, and a maskable interrupt (INTWDT) can be generated. The priority of INTWDT is set as the highest of all the maskable interrupts.

The interval timer continues operation in HALT mode, but stops in STOP mode. Therefore, set RUN to 1 before entering 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 watchdog timer mode is selected), 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 9-5. Interval Time of Interval Timer

TCL22	TCL21	TCL20	Interval	At fx = 3.58 MHz
0	0	0	$2^{11} \times 1/f_X$	572 μs
0	1	0	$2^{13} \times 1/f_X$	2.29 ms
1	0	0	$2^{15} \times 1/f_X$	9.15 ms
1	1	0	$2^{17} \times 1/f_X$	36.6 ms

fx: Main system clock oscillation frequency

CHAPTER 10 CLOCK OUTPUT CIRCUIT

10.1 Clock Output Circuit Functions

The clock output circuit (PBU) has the following functions.

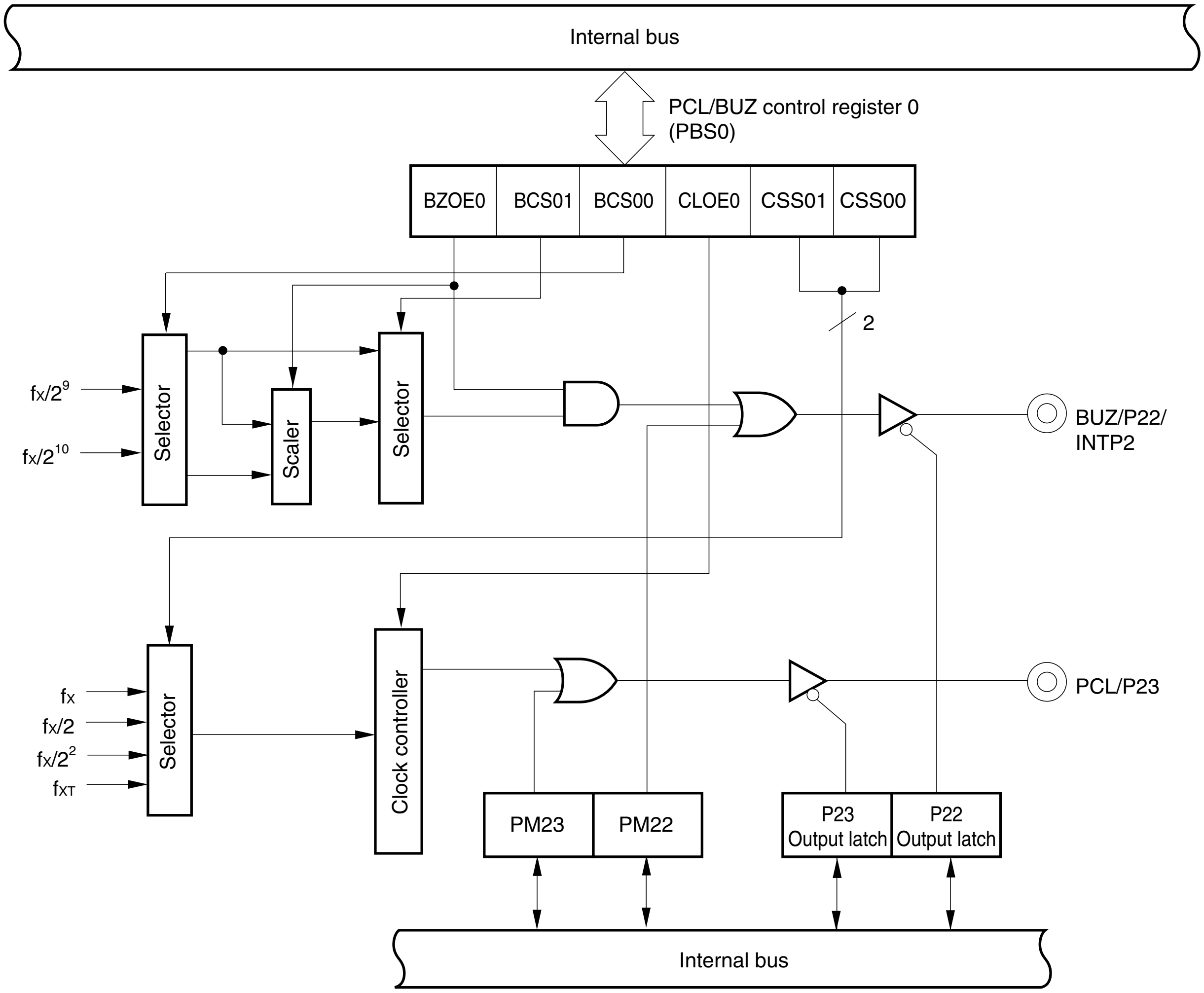
(1) PCL output
Pulse clocks are output from the PCL/P23 pin, and are supplied to peripheral LSIs.

(2) Buzzer output
A signal having the buzzer frequency is output from the BUZ/P22/INTP2 pin.

10.2 Clock Output Circuit Configuration

Figure 10-1 is a block diagram of the clock output circuit (PBU).

Figure 10-1. Block Diagram of Clock Output Circuit



10.3 Clock Output Circuit Control Registers

The following two types of registers are used to control the clock output circuit (PBU).

- PCL/BUZ control register 0 (PBS0)
- Port mode register 2 (PM2)

(1) PCL/BUZ control register 0 (PBS0)

PBS0 controls clock pulse output and buzzer output.
PBS0 is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears PBS0 to 00H.

Figure 10-2. Format of PCL/BUZ Control Register 0

Symbol	<7>	6	5	<4>	3	2	1	0	Address	After reset	R/W
PBS0	BZOE0	BCS01	BCS00	CLOE0	0	0	CSS01	CSS00	FF40H	00H	R/W

BZOE0	Buzzer output control
0	Clock divider operation and buzzer output disabled
1	Clock divider operation and buzzer output enabled

BCS01	Count clock selection
0	Count clock selected by BCS00 (bit 5 of PBS0)
1	BCS00-selected count clock divided-by-4

BCS00	Buzzer count clock selection
0	$f_X/2^9$ (6.9 kHz)
1	$f_X/2^{10}$ (3.5 kHz)

CLOE0	Pulse clock output control
0	Pulse clock output disabled
1	Pulse clock output enabled

CSS01	CSS00	Pulse clock count clock selection
0	0	f_X (3.58 MHz)
0	1	$f_X/2$ (1.79 MHz)
1	0	$f_X/2^2$ (895 MHz)
1	1	f_{XT} (32.768 MHz)

Cautions 1. Bits 2 and 3 must be fixed to 0.

2. Do not write to BCS01, BCS00, CSS01, and CSS00 while buzzer or clock pulses are being output. To change the count clocks, first disable buzzer output (BZOE0 = 0) and pulse clock output (CLOE0 = 0).

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. The parenthesized values apply to operation at f_X = 3.58 MHz or at f_{XT} = 32.768 kHz.

(2) Port mode register 2 (PM2)

PM2 sets each bit of port 2 to either input or output. To use the P22/BUZ/INTP2 pin for buzzer output, set both PM22 and the output latch for P22 to 0.

To use the P23/PCL pin for pulse clock output, set both PM23 and the output latch for P23 to 0.

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

$\overline{\text{RESET}}$ input clears PM2 to 00H.

Figure 10-3. Format of Port Mode Register 2

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

PM22	P22 pin I/O mode selection
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

PM23	P23 pin I/O mode selection
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

10.4 Clock Output Circuit Operation

10.4.1 PCL output operation

The PCL/P23 pin outputs a pulse clock to the peripheral LSI.

To use the clock output circuit for PCL output, make the following setting:

- Set P23 to output mode (PM23 = 0).
- Clear the output latch of P23 to 0.
- Set PCL/BUZ control register 0 (PBS0) as shown in Figure 10-4.

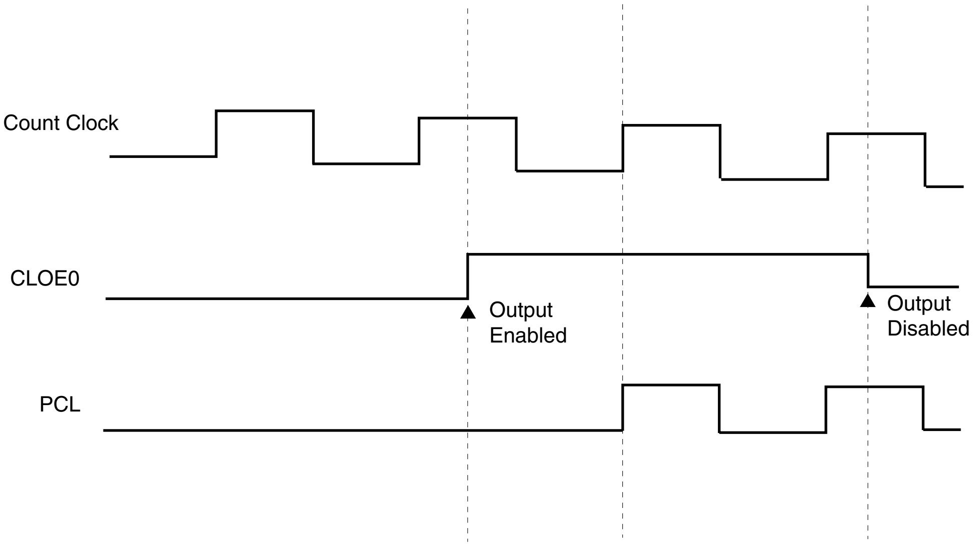
Figure 10-4. Setting of PCL/BUZ Control Register 0 for PCL Output Operation

				CLOE0			CSS01	CSS00
PBS0	–	–	–	1	0	0	0/1	0/1

Selects count clock
Enables output of pulse clock

Figure 10-5 shows the timing of PCL output.

Figure 10-5. PCL Output Timing



Because the pulse clock output enable signal is latched when the count clock goes low, PCL output is always started from the low level of the count clock even if the output is enabled asynchronously (CLOE0 = 1).

If the output is disabled asynchronously (CLOE0 = 0), the high level of the count clock is guaranteed before the output is stopped.

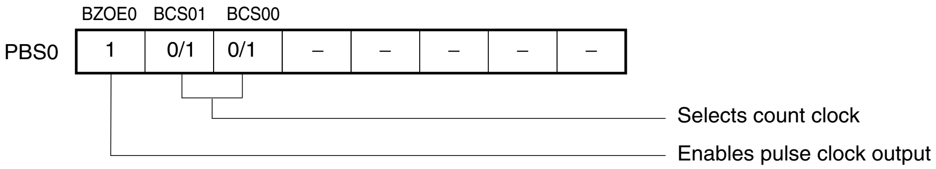
10.4.2 Buzzer output operation

The BUZ/P22/INTP2 pin can be used to output a buzzer frequency.

To use the clock output circuit for buzzer output, make the following setting:

- Set P22 to output mode (PM22 = 0).
- Clear the output latch of P22 to 0.
- Set PCL/BUZ control register 0 (PBS0) as shown in Figure 10-6.

Figure 10-6. Setting of PCL/BUZ Control Register 0 for Buzzer Output Operation



CHAPTER 11 SERIAL INTERFACE UART00

11.1 Serial Interface UART00 Functions

Serial interface UART00 has the following two types of modes.

- Operation stop mode
- Asynchronous serial interface (UART) mode

(1) Operation stop mode

This mode is used when serial transfer is not performed. Power consumption is minimized in this mode.

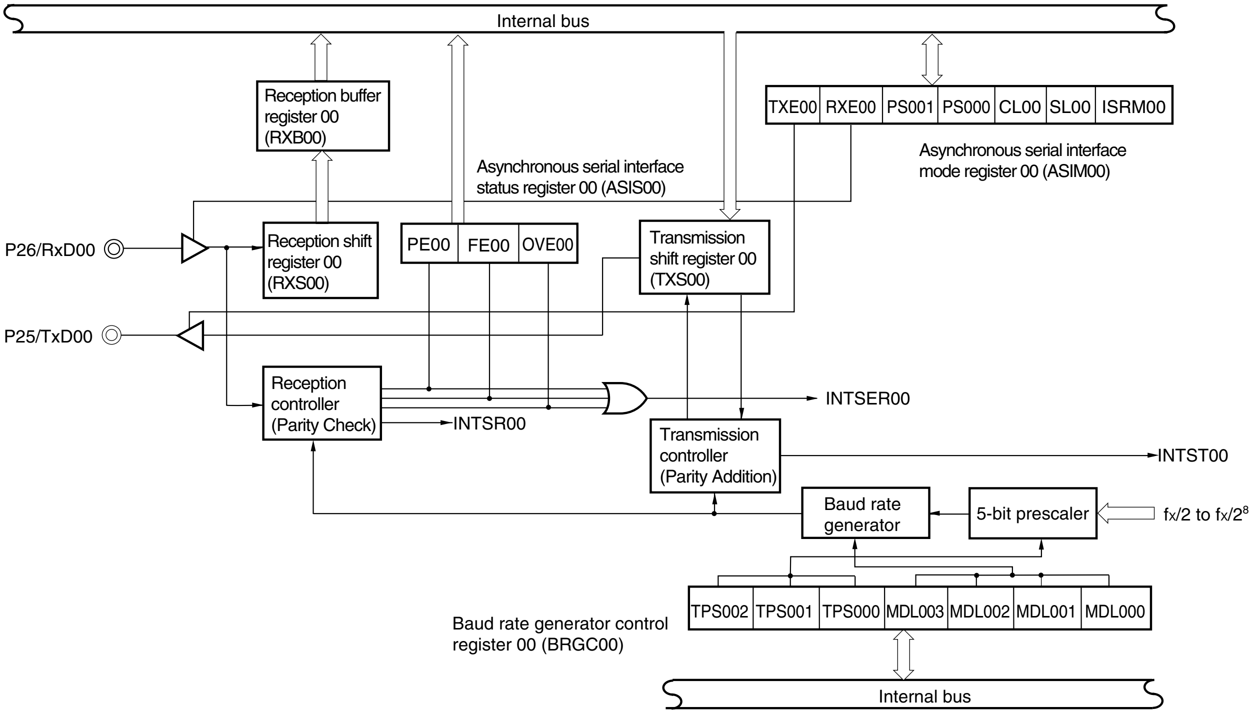
(2) Asynchronous serial interface (UART) mode

This mode is used to send and receive the one byte of data that follows a start bit. It supports full-duplex communication.

Serial interface UART00 contains a UART-dedicated baud rate generator, enabling communication over a wide range of baud rates. The UART-dedicated baud rate generator, for example, enables the use of a MIDI standard baud rate (31.25 kbps).

Figure 11-1 is a block diagram of serial interface UART00.

Figure 11-1. Block Diagram of Serial Interface UART00


Internal bus
Reception buffer register 00 (RXB00)
TXE00
RXE00
PS001
PS000
CL00
SL00
ISRM00
Asynchronous serial interface mode register 00 (ASIM00)
Asynchronous serial interface status register 00 (ASIS00)
P26/RxD00
Reception shift register 00 (RXS00)
PE00
FE00
OVE00
Transmission shift register 00 (TXS00)
P25/TxD00
Reception controller (Parity Check)
INTSER00
INTSR00
Transmission controller (Parity Addition)
INTST00
Baud rate generator
5-bit prescaler
fx/2 to fx/2⁸
TPS002
TPS001
TPS000
MDL003
MDL002
MDL001
MDL000
Baud rate generator control register 00 (BRGC00)
Internal bus

11.2 Serial Interface UART00 Configuration

Serial interface UART00 includes the following hardware.

Table 11-1. Serial Interface UART00 Configuration

Item	Configuration
Register	Transmission shift register 00 (TXS00) Reception shift register 00 (RXS00) Reception buffer register 00 (RXB00)
Control register	Asynchronous serial interface mode register 00 (ASIM00) Asynchronous serial interface status register 00 (ASIS00) Baud rate generator control register 00 (BRGC00)

(1) Transmission shift register 00 (TXS00)

TXS00 is a register in which transmission data is prepared. The transmission data is output from TXS00 bit-serially.

When the data length is seven bits, bits 0 to 6 of the data in TXS00 will be transmission data. Writing data to TXS00 triggers transmission.

TXS00 can be write-accessed, using an 8-bit memory manipulation instruction, but cannot be read-accessed.

$\overline{\text{RESET}}$ input sets TXS00 to FFH.

Caution Do not write to TXS00 during transmission.
TXS00 and reception buffer register 00 (RXB00) are mapped at the same address, such that any attempt to read from TXS00 results in a value being read from the RXB00.

(2) Reception shift register 00 (RXS00)

RXS00 is a register in which serial data, received at the RxD00 pin, is converted to parallel data. Once one entire byte has been received, RXS00 feeds the reception data to reception buffer register 00 (RXB00). RXS00 cannot be manipulated directly by a program.

(3) Reception buffer register 00 (RXB00)

RXB00 is used to hold reception data. Once reception shift register 00 (RXS00) has received one entire byte of data, it feeds that data into RXB00.

When the data length is seven bits, the reception data is sent to bits 0 to 6 of RXB00, in which the MSB is fixed to 0.

RXB00 can be read-accessed, using an 8-bit memory manipulation instruction, but cannot be write-accessed.

$\overline{\text{RESET}}$ input sets RXB00 to FFH.

Caution RXB00 and transmission shift register 00 (TXS00) are mapped at the same address, such that any attempt to write to RXB00 results in a value being written to TXS00.

(4) Transmission controller

The transmission controller controls transmission. For example, it adds start, parity, and stop bits to the data in transmission shift register 00 (TXS00), according to the setting of asynchronous serial interface mode register 00 (ASIM00).

(5) Reception controller

The reception controller controls reception according to the setting of asynchronous serial interface mode register 00 (ASIM00). It also checks for errors, such as parity errors, during reception. If an error is detected, asynchronous serial interface status register 00 (ASIS00) is set according to the status of the error.

11.3 Serial Interface UART00 Control Registers

The following three types of registers are used to control serial interface UART00.

- Asynchronous serial interface mode register 00 (ASIM00)
- Asynchronous serial interface status register 00 (ASIS00)
- Baud rate generator control register 00 (BRGC00)

(1) Asynchronous serial interface mode register 00 (ASIM00)

ASIM00 is an 8-bit register that is used to control the serial transfer operation of serial interface UART00.
ASIM00 is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears ASIM00 to 00H.

Caution When using the serial interface function in UART mode, set the related output latches to 0, and the port mode registers (PM××) as follows.

- **For reception**
  Set P26 (RxD00) to input mode (PM26 = 1).
- **For transmission**
  Set P25 (TxD00) to output mode (PM25 = 0).
- **For transmission and reception**
  Set P26 and P25 to input and output mode, respectively.

Figure 11-2. 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	RXE00	Operation mode	Function of RxD00/P26 pin	Function of TxD00/P25 pin
0	0	Operation disabled	Port function (P26)	Port function (P25)
0	1	UART mode (reception only)	Serial function (RxD00)	
1	0	UART mode (transmission only)	Port function (P26)	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	An interrupt request is generated at error occurrence.
1	An interrupt request is not generated at error occurrence.

Cautions 1. Bit 0 must be fixed to 0.

2. Switch operation mode from one mode to another after stopping both serial transmission and reception.

(2) Asynchronous serial interface status register 00 (ASIS00)

ASIS00 is used to display the type of a reception error, if it occurs while UART mode is set.
ASIS00 is set with a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears ASIS00 to 00H.

Figure 11-3. 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	FFA1H	00H	R

PE00	Parity error flag
0	Parity error not generated
1	Parity error generated (when the transmit parity and receive parity do not match)

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 reception buffer register 00)

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 reception buffer register 00 (RXB00) when an overrun error occurs. If not, every time the data is received an overrun error is generated.

(3) Baud rate generator control register 00 (BRGC00)

BRGC00 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 clears BRGC00 to 00H.

Figure 11-4. 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	FFA2H	00H	R/W

TPS002	TPS001	TPS000	5-bit counter source clock selection
0	0	0	$f_X/2$ (1.79 MHz)
0	0	1	$f_X/2^2$ (895 kHz)
0	1	0	$f_X/2^3$ (448 kHz)
0	1	1	$f_X/2^4$ (224 kHz)
1	0	0	$f_X/2^5$ (112 kHz)
1	0	1	$f_X/2^6$ (56 kHz)
1	1	0	$f_X/2^7$ (28 kHz)
1	1	1	$f_X/2^8$ (14 kHz)

MDL003	MDL002	MDL001	MDL000	Baud rate generator input clock selection
0	0	0	0	$f_{SCK}/16$
0	0	0	1	$f_{SCK}/17$
0	0	1	0	$f_{SCK}/18$
0	0	1	1	$f_{SCK}/19$
0	1	0	0	$f_{SCK}/20$
0	1	0	1	$f_{SCK}/21$
0	1	1	0	$f_{SCK}/22$
0	1	1	1	$f_{SCK}/23$
1	0	0	0	$f_{SCK}/24$
1	0	0	1	$f_{SCK}/25$
1	0	1	0	$f_{SCK}/26$
1	0	1	1	$f_{SCK}/27$
1	1	0	0	$f_{SCK}/28$
1	1	0	1	$f_{SCK}/29$
1	1	1	0	$f_{SCK}/30$
1	1	1	1	Setting prohibited

Cautions 1. Bit 7 must be fixed to 0.

2. When writing to BRGC00 is performed during a communication operation, the output of baud rate generator is disrupted and communications cannot be performed normally. Be sure not to write to BRGC00 during communication operation.

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 3.58 MHz.

3. f_{SCK}: Source clock of the 5-bit counter

11.4 Serial Interface UART00 Operation

Serial interface UART00 provides the following two types of modes.

- Operation stop mode
- Asynchronous serial interface (UART) mode

11.4.1 Operation stop mode

In operation stop mode, serial transfer is not executed, therefore, the power consumption can be reduced. In this mode, the pins can be used as normal I/O ports.

(1) Register setting

Operation mode is set by asynchronous serial interface mode register 00 (ASIM00).

ASIM00 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears 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	RXE00	Operation mode	Function of RxD00/P26 pin	Function of TxD00/P25 pin
0	0	Operation disabled	Port function (P26)	Port function (P25)
0	1	UART mode (reception only)	Serial function (RxD00)	
1	0	UART mode (transmission only)	Port function (P26)	Serial function (TxD00)
1	1	UART mode (transmission and reception)	Serial function (RxD00)	

Caution Switch operation mode from one mode to another after stopping both serial transmission and reception.

11.4.2 Asynchronous serial interface (UART) mode

In this mode, the one-byte data following the start bit is transmitted/received and thus full-duplex communication is possible.

The serial interface contains a UART-dedicated baud rate generator that enables communications at a desired baud rate from many options.

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 asynchronous serial interface mode register 00 (ASIM00), asynchronous serial interface status register 00 (ASIS00), and baud rate generator control register 00 (BRGC00).

(a) Asynchronous serial interface mode register 00 (ASIM00)

ASIM00 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIM00 to 00H.

Caution When using the asynchronous serial interface function (UART mode), set the related output latches to 0, and port mode register 2 (PM2) as follows.

- **For reception**
  Set P26 (RxD00) to input mode (PM26 = 1).
- **For transmission**
  Set P25 (TxD00) to output mode (PM25 = 0).
- **For transmission and reception**
  Set P26 and P25 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/P26 Pin	Function of TxD00/P25 Pin
0	0	Operation disabled	Port function (P26)	Port function (P25)
0	1	UART mode (reception only)	Serial function (RxD00)	
1	0	UART mode (transmission only)	Port function (P26)	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	An interrupt request is generated at error occurrence.
1	An interrupt request is not generated at error occurrence.

Cautions 1. Bit 0 must be fixed to 0.

2. Switch operation mode from one mode to another after stopping both serial transmission and reception.

(b) Asynchronous serial interface status register 00 (ASIS00)

ASIS00 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\mathrm{RESET}}$ input clears 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	FFA1H	00H	R

PE00	Parity error flag
0	Parity error not generated
1	Parity error generated (when the parity of transmission data does not match)

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 reception buffer register 00)

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 reception buffer register 00 (RXB00) when an overrun error occurs. If not, every time the data is received an overrun error is generated.

Caution Bits 3 to 7 must be fixed to 0.

(c) Baud rate generator control register 00 (BRGC00)

BRGC00 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears 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	FFA2H	00H	R/W

TPS002	TPS001	TPS000	5-bit counter source clock selection
0	0	0	$f_X/2$ (1.79 MHz)
0	0	1	$f_X/2^2$ (895 kHz)
0	1	0	$f_X/2^3$ (448 kHz)
0	1	1	$f_X/2^4$ (224 kHz)
1	0	0	$f_X/2^5$ (112 kHz)
1	0	1	$f_X/2^6$ (56 kHz)
1	1	0	$f_X/2^7$ (28 kHz)
1	1	1	$f_X/2^8$ (14 kHz)

MDL003	MDL002	MDL001	MDL000	Baud rate generator input clock selection
0	0	0	0	$f_{SCK}/16$
0	0	0	1	$f_{SCK}/17$
0	0	1	0	$f_{SCK}/18$
0	0	1	1	$f_{SCK}/19$
0	1	0	0	$f_{SCK}/20$
0	1	0	1	$f_{SCK}/21$
0	1	1	0	$f_{SCK}/22$
0	1	1	1	$f_{SCK}/23$
1	0	0	0	$f_{SCK}/24$
1	0	0	1	$f_{SCK}/25$
1	0	1	0	$f_{SCK}/26$
1	0	1	1	$f_{SCK}/27$
1	1	0	0	$f_{SCK}/28$
1	1	0	1	$f_{SCK}/29$
1	1	1	0	$f_{SCK}/30$
1	1	1	1	Setting prohibited

Cautions 1. Bit 7 must be fixed to 0.

2. When writing to BRGC00 is performed during a communication operation, the output of baud rate generator is disrupted and communications cannot be performed normally. Be sure not to write to BRGC00 during communication operation.

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 3.58 MHz.

3. f_{SCK}: Source clock of the 5-bit counter

The baud rate transmit/receive clock to be generated is a signal scaled from the main system clock.

- **Generation of baud rate transmit/receive clock by means of main system clock**

  The transmit/receive clock is generated by scaling the main system clock. The baud rate generated from the main system clock is estimated by using the following expression.

  $$[\text{Baud rate}] = \frac{f_X}{2^{n+1}(k + 16)} \quad [\text{Hz}]$$

  f_X: Main system clock oscillation frequency

Table 11-2 shows the relation between the source clock of the 5-bit counter assigned to bits 4 to 6 (TPS000 to TPS002) of BRGC00, and value n.

Table 11-2. Relation Between Source Clock of 5-Bit Counter and Value n

TPS002	TPS001	TPS000	5-Bit Counter Source Clock Selection	n
0	0	0	$f_X/2$ (1.79 MHz)	0
0	0	1	$f_X/2^2$ (895 kHz)	1
0	1	0	$f_X/2^3$ (448 kHz)	2
0	1	1	$f_X/2^4$ (224 kHz)	3
1	0	0	$f_X/2^5$ (112 kHz)	4
1	0	1	$f_X/2^6$ (56 kHz)	5
1	1	0	$f_X/2^7$ (28 kHz)	6
1	1	1	$f_X/2^8$ (14 kHz)	7

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 3.58 MHz.

- **Permissible error range of baud rate**

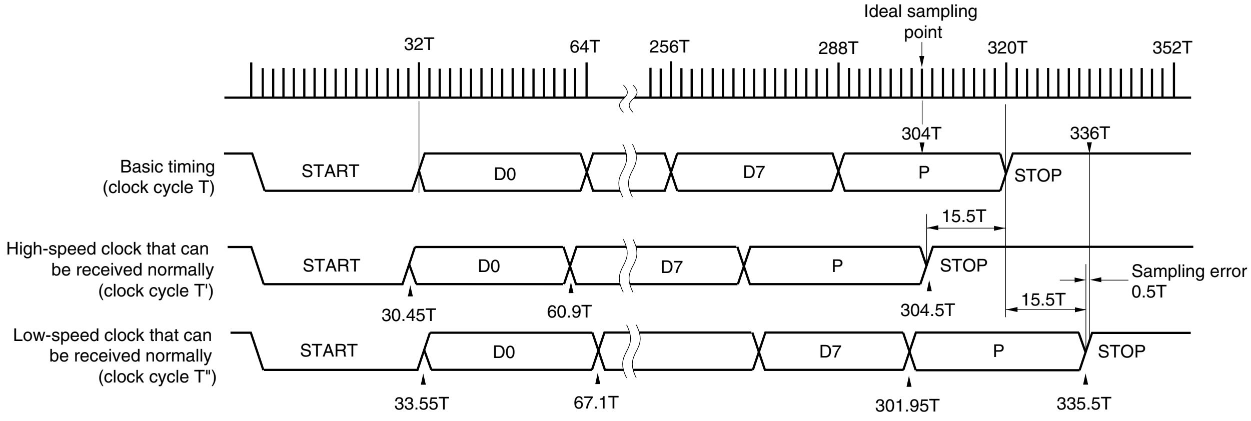
  The permissible error range of the baud rate is dependent upon the number of bits of one frame and the division ratio of the counter [1/(16 + k)]. Table 11-3 shows the relationship between the main system clock and baud rate, while Figure 11-5 shows an example of the permissible error in the baud rate.

Table 11-3. Example of Relationships Between Main System Clock and Baud Rate

Baud Rate [bps]	fx = 5.0 MHz		fx = 4.1943 MHz		fx = 3.58 MHz	
	BRGC00	Error (%)	BRGC00	Error (%)	BRGC00	Error (%)
75	–	–	–	–	–	–
110	–	–	–	–	–	–
150	–	–	–	–	–	–
300	–	–	7BH	1.14	77H	1.33
600	70H	1.73	6BH	1.14	67H	1.33
1,200	60H	1.73	5BH	1.14	57H	1.33
2,400	50H	1.73	4BH	1.14	47H	1.33
4,800	40H	1.73	3BH	1.14	37H	1.33
9,600	30H	1.73	2BH	1.14	27H	1.33
19,200	20H	1.73	1BH	1.14	17H	1.33
31,250	14H	0.00	11H	-1.31	0CH	-1.24
38,400	10H	1.73	0BH	1.14	07H	1.33
76,800	00H	1.73	–	–	–	–
115,200	–	–	–	–	–	–

Remark fx: Main system clock oscillation frequency

Figure 11-5. Permissible Error in Baud Rate with Sampling Error Considered (Where k = 0)



Remark T: Source clock cycle of 5-bit counter

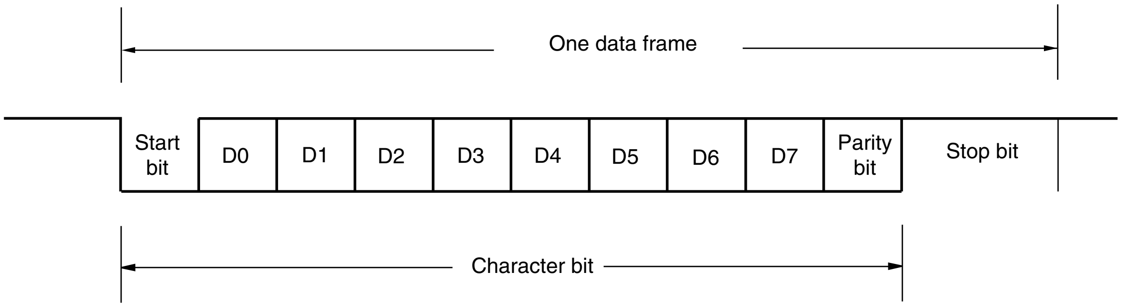
Permissible error range of baud rate (where k = 0) = $\frac{\pm 15.5}{320} \times 100 = 4.8438$ (%)

(2) Communication operation

(a) Data format

The transmission/reception data format is as shown in Figure 11-6.

Figure 11-6. 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 with asynchronous serial interface mode register 00 (ASIM00).

When 7 bits are selected as the number of character bits, only the lower 7 bits (bits 0 to 6) are valid; in transmission the most significant bit (bit 7) is ignored, and in reception the most significant bit (bit 7) is always "0".

The serial transfer rate is selected by means of 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 one-bit (odd number) error can be detected. With 0 parity and no parity, an error cannot be detected.

(i) Even parity

- **At transmission**

  The transmission operation is controlled so that the number of bits with a value of "1" in the transmission data including parity bit may be even. The parity bit value should be as follows.

  The number of bits with a value of "1" is an odd number in transmission data: 1
  The number of bits with a value of "1" is an even number in transmission data: 0

- **At 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

- **At transmission**

  Conversely to the even parity, the transmission operation is controlled so that the number of bits with a value of "1" in the transmission data including parity bit may be odd. The parity bit value should be as follows.

  The number of bits with a value of "1" is an odd number in transmission data: 0
  The number of bits with a value of "1" is an even number in transmission data: 1

- **At 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

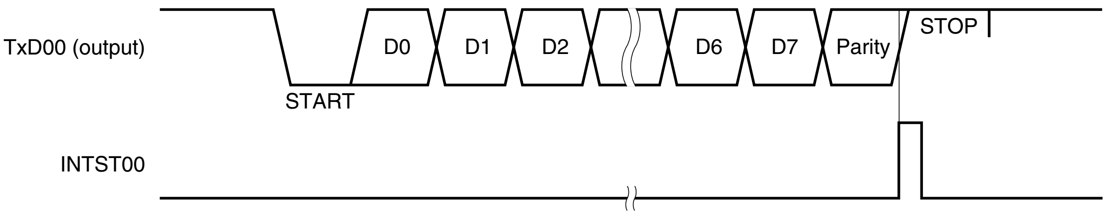
A transmit operation is started by writing transmission data to transmission shift register 00 (TXS00). The start bit, parity bit, and stop bit(s) are added automatically.

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.

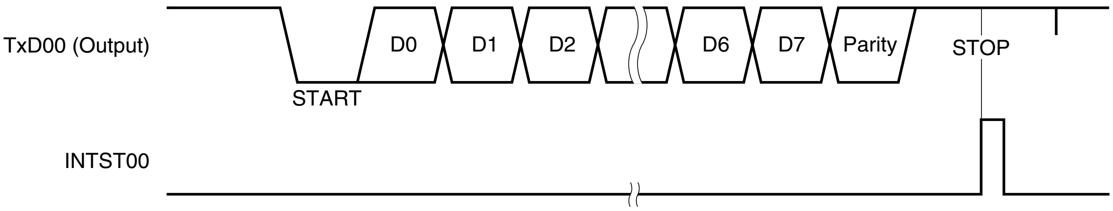
The transmission completion interrupt timing is shown in Figure 11-7.

Figure 11-7. Asynchronous Serial Interface Transmission Completion Interrupt Request Timing

(i) Stop bit length: 1



(ii) Stop bit length: 2



Caution Do not replace asynchronous serial interface mode register 00 (ASIM00) during a transmit operation. If the ASIM00 register is replaced 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 at the time when the half time determined by the specified baud rate has passed, the data sampling start timing signal is output. If the 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.

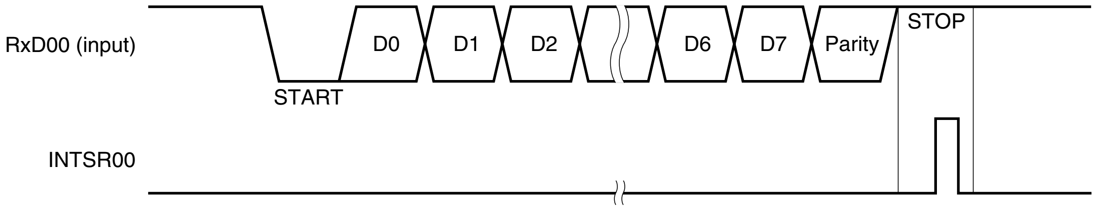
When one frame of data has been received, the reception data in the shift register is transferred to reception buffer register 00 (RXB00), and a reception completion interrupt request (INTSR00) is generated.

Even if an error is generated, the reception data in which the error was generated is still transferred to RXB00. If bit 1 (ISRM00) of ASIM00 is cleared to 0 when an error is generated, INTSR00 is generated (see Figure 11-9). 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 11-8 shows the asynchronous serial interface reception completion interrupt request timing.

Figure 11-8. Asynchronous Serial Interface Reception Completion Interrupt Request 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 11-4.

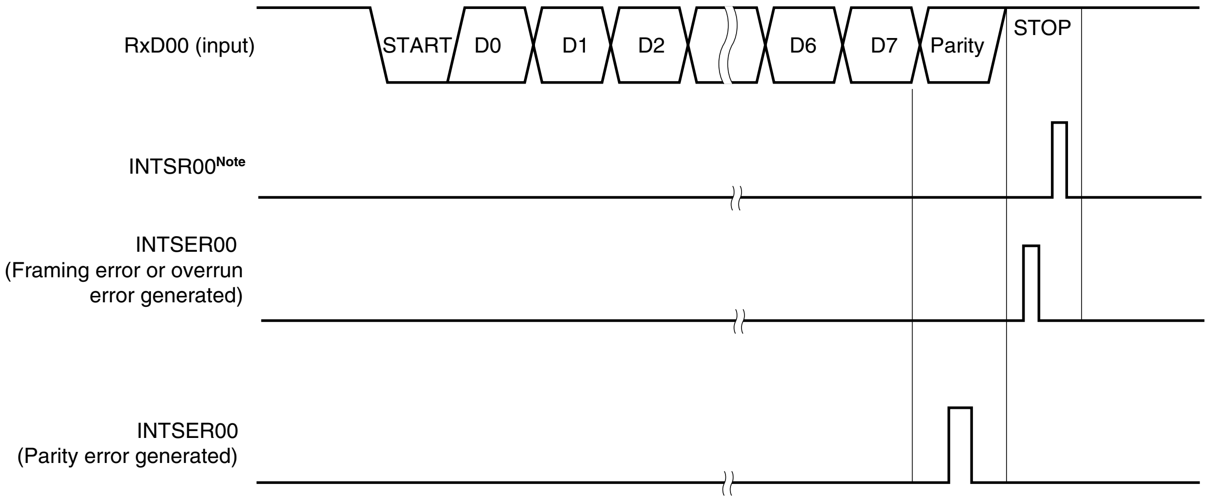
It is possible to determine what kind of error was generated during reception by reading the contents of ASIS00 in the reception error interrupt servicing (INTSER00) (see **Table 11-4** and **Figure11-9**).

The contents of ASIS00 are cleared to 0 by reading reception 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 11-4. Receive Error Causes

Receive Errors	Cause	ASIS00 Value
Parity error	Transmission-time parity specification 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 reception buffer register 00	01H

Figure 11-9. Receive Error Timing



Note INTSR00 is not generated if the receive error is generated when the ISRM00 bit is set to 1.

Cautions 1. The contents of asynchronous serial interface status register 00 (ASIS00) are cleared to 0 by reading reception buffer register 00 (RXB00) or receiving the next data. To ascertain the error contents, read ASIS00 before reading RXB00.

2. Be sure to read reception buffer register 00 (RXB00) even if a receive error is generated. 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.

CHAPTER 12 LCD CONTROLLER/DRIVER

12.1 LCD Controller/Driver Functions

The LCD controller/driver (LCD20) has the following functions:

(1) Automatic output of segment and common signals based on automatic display data memory read
(2) Operation in 1/16 duty (1/5 bias) display mode
Up to 640 pixels (40 segments × 16 commons)
(3) Four different frame frequencies selectable
(4) Up to 40 segment signal outputs (S0 to S39) and 16 common signal outputs (COM0 to COM15)
Of these segment signal outputs, 8 outputs can be switched to input/output ports bit by bit (P50/S39 to P57/S32).
(5) Operation with a subsystem clock

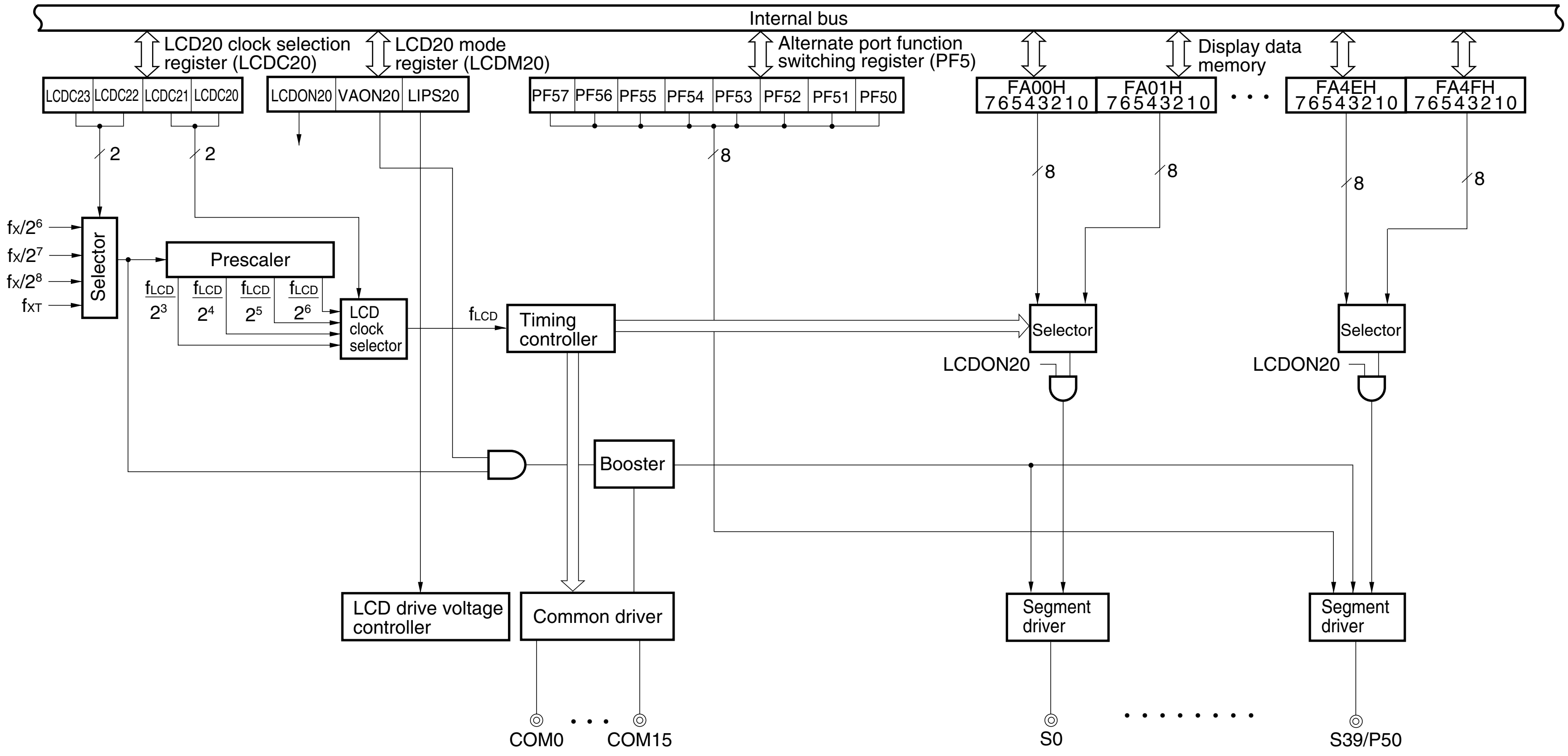
12.2 LCD Controller/Driver Configuration

The LCD controller/driver (LCD20) includes the following hardware.

Table 12-1. Configuration of LCD Controller/Driver

Item	Configuration
Display output	40 segment signals (32 dedicated segment signals and 8 segment and input/output port signals) 16 common signals
Control register	LCD20 mode register (LCDM20) Alternate port function switching register (PF5) LCD20 clock selection register (LCDC20)

Figure 12-1. Block Diagram of LCD Controller/Driver



12.3 LCD Controller/Driver Control Registers

The following three types of registers are used to control the LCD controller/driver (LCD20).

- LCD20 mode register (LCDM20)
- Alternate port function switching register (PF5)
- LCD20 clock selection register (LCDC20)

(1) LCD20 mode register (LCDM20)

LCDM20 specifies whether to enable display operation. It also specifies the operation mode and LCD drive power supply.

LCDM20 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears LCDM20 to 00H.

Figure 12-2. Format of LCD20 Mode Register

Symbol	<7>	<6>	5	<4>	3	2	1	0	Address	After reset	R/W
LCDM20	LCDON20	VAON20	0	LIPS20	0	0	0	0	FFB0H	00H	R/W

LCDON20	Control of LCD display
0	Display OFF
1	Display ON

VAON20	LCD controller/driver operation mode[Note]
0	Normal operation
1	Low-voltage operation

LIPS20	LCD drive power supply selection
0	LCD drive power is not supplied.
1	LCD drive power is supplied.

Note Clear VAON20 to 0 to reduce the power consumption when LCD display is not performed.

Cautions 1. Bits 0 to 3 and bit 5 must be fixed to 0.

2. When manipulating VAON20, clear LCDON20 to 0 and turn off the LCD display.

(2) Alternate port function switching register (PF5)

PF5 controls port and segment signal output switching.

PF5 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears PF5 to 00H.

Figure 12-3. Format of Alternate Port Function Switching Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PF5	PF57	PF56	PF55	PF54	PF53	PF52	PF51	PF50	FFB1H	00H	R/W

	PF57	PF56	PF55	PF54	PF53	PF52	PF51	PF50
	P57/S32	P56/S33	P55/S34	P54/S35	P53/S36	P52/S37	P51/S38	P50/S39
0	Used as ports (P5n)							
1	Used as segments (S×)							

Remark n = 0 to 7
× = 32 to 39

(3) LCD20 clock selection register (LCDC20)

LCDC20 specifies the LCD clock and LCD frame frequency.

LCDC20 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears LCDC20 to 00H.

Figure 12-4. Format of LCD20 Clock Selection Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
LCDC20	0	0	0	0	LCDC203	LCDC202	LCDC201	LCDC200	FFB2H	00H	R/W

LCDC203	LCDC202	LCD clock (f_{LCD}) selection
0	0	$f_X/2^6$ (55.9 kHz)
0	1	$f_X/2^7$ (28.0 kHz)
1	0	$f_X/2^8$ (14.0 kHz)
1	1	f_{XT} (32.768 kHz)

LCDC201	LCDC200	LCD clock frequency selection
0	0	$f_{LCD}/2^3$
0	1	$f_{LCD}/2^4$
1	0	$f_{LCD}/2^5$
1	1	$f_{LCD}/2^6$

Caution Bits 4 to 7 must be fixed to 0.

Remarks 1. f_X: Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. The parenthesized values apply to operation at f_X = 3.58 MHz or at f_{XT} = 32.768 kHz.

Table 12-2 lists the frame frequencies used when f_{XT} (32.768 kHz) is supplied to the LCD clock (f_{LCD}).

Table 12-2. LCD Frame Frequencies

LCDC201	LCDC200	LCD Clock Frequency	Frame Frequency
0	0	4,096 Hz	256 Hz
0	1	2,048 Hz	128 Hz
1	0	1,024 Hz	64 Hz
1	1	512 Hz	32 Hz

12.4 Setting Up LCD Controller/Driver

Set up the LCD controller/driver using the following procedure.

<1> Set initial values in display data memory (FA00H to FA4FH).
<2> Set pins to be used for segment output in the alternate port function switching register (PF5).
<3> Enable the LCD display and set the operation modes in the LCD20 mode register (LCDM20).
<4> Set the LCD clock in the LCD20 clock selection register (LCDC20).

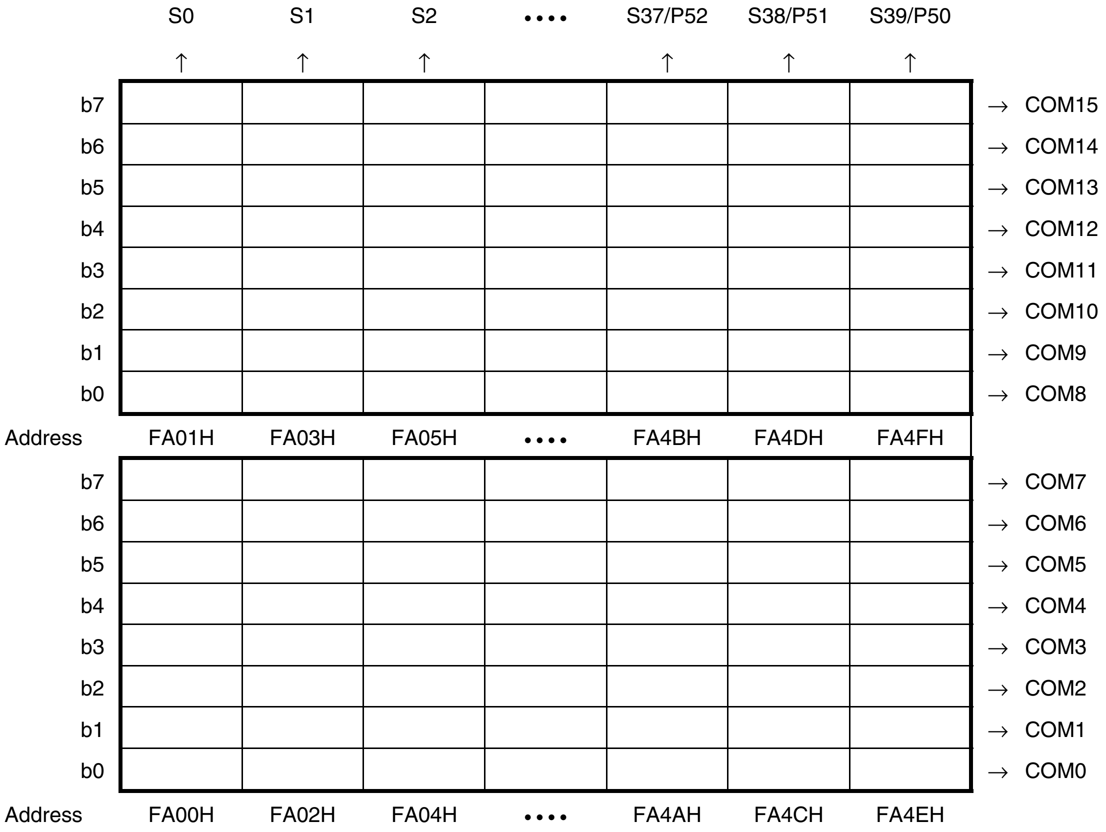
12.5 LCD Display Data Memory

The LCD display data memory is mapped at addresses FA00H to FA4FH. Data in the LCD display data memory can be displayed on the LCD panel using the LCD controller/driver.

Figure 12-5 shows the relationships between the contents of the LCD display data memory and the segment/common outputs.

That part of the display data memory which is not used for display can be used as ordinary RAM.

Figure 12-5. Relationships Between LCD Display Data Memory Contents and Segment/Common Outputs



Each pixel of the LCD panel becomes on when the potential difference between the corresponding common and segment signals becomes higher than a specific voltage ($2V_{LCD}$). It becomes off when the potential difference becomes lower than $2V_{LCD}$ (for details, see **(1)** below).

A segment signal is converted to a select voltage if the contents of the corresponding bit of each display data memory are 1; if the contents of the bit are 0, they are converted to an unselect voltage and output to a segment pin (S0 to S39). Note that S32 to S39 can be used also as an I/O port.

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

Applying DC voltage as the common and segment signals for an LCD panel would deteriorate it. To avoid this problem, this LCD panel is driven with AC voltage.

(1) Output waveforms of common and segment signals

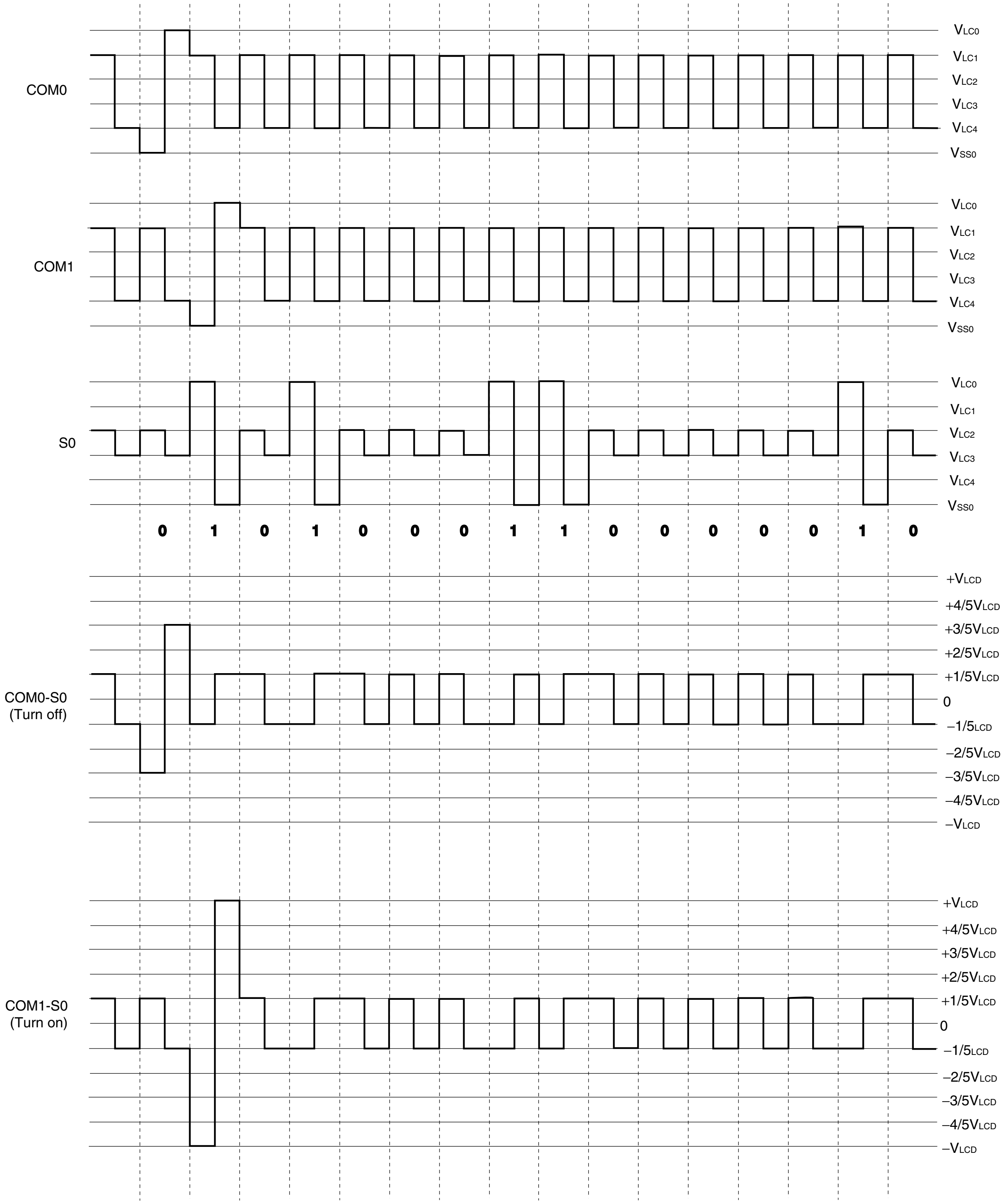
Voltages listed in Table 12-3 are output as common and segment signals.

When both common and segment signals are at the select voltage, a display on-voltage of $\pm V_{LCD}$ is obtained. The other combinations of the signals correspond to the display off-voltage. Figure 12-6 shows an example LCD drive waveform between a segment signal and a common signal.

Table 12-3. LCD Drive Voltage

Segment Signal Common Signal		Select Signal Level	Deselect Signal Level
		V_{SS0}/V_{LC0}	V_{LC3}/V_{LC2}
Select signal level	V_{LC0}/V_{SS0}	$+V_{LCD}/-V_{LCD}$	$+\frac{3}{5}V_{LCD}/-\frac{3}{5}V_{LCD}$
Deselect signal level	V_{LC4}/V_{LC1}	$+\frac{1}{5}V_{LCD}/-\frac{1}{5}V_{LCD}$	$-\frac{1}{5}V_{LCD}/+\frac{1}{5}V_{LCD}$

Figure 12-6. Example LCD Drive Waveform Between Segment Signal and Common Signal



12.6 Supplying LCD Drive Voltage

The μPD789830 Subseries has a divider resistor that is used to supply power to drive an LCD.

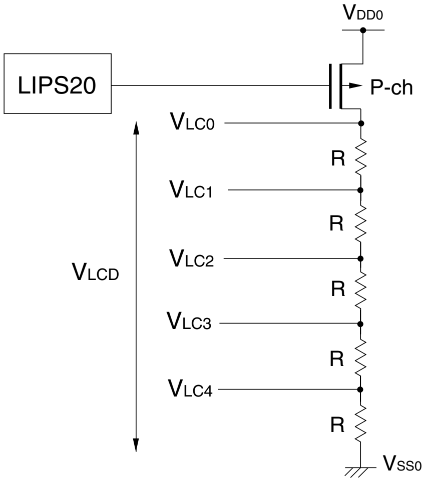
The LCD drive voltage is selected by manipulating LIPS20 (bit 4 of the LCD20 mode register (LCDM20)).

When LIPS20 = 0, the current flowing through the divider resistor is cut off.

When LIPS20 = 1, the supply voltage V_{DD} is divided by the divider resistor and LCD drive voltages V_{LCD0} through V_{LCD4} are supplied.

Figure 12-7 shows the connection of the power supply for the LCD drive.

Figure 12-7. Connection of Power Supply for LCD Drive



12.7 LCD Display

The μPD789830 Subseries can display 8 columns × 2 rows on an LCD panel. The first row displays 12345678, while the second row displays ABCDEFGH. These characters correspond to the contents of the display data memory (FA00H to FA4FH). Figure 12-8 shows the connection of the segment signals (S0 to S39) and common signals (COM0 to COM15) of an LCD panel having 8 columns × 2 rows.

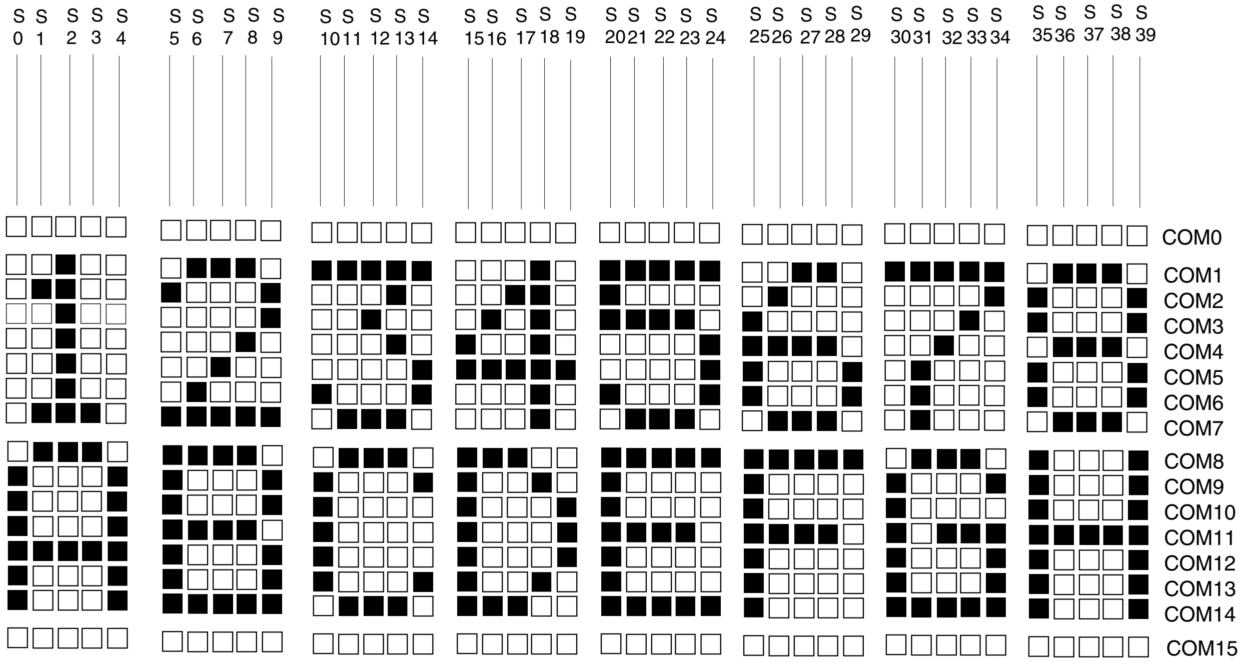
Figure 12-8. Example of Connecting LCD Panel

	FA00H	FA02H	FA04H	FA06H	FA08H	FA0AH	FA0CH	FA0EH	FA10H	FA12H	FA14H	FA16H	FA18H	FA1AH	FA1CH	FA1EH	FA20H	FA22H	FA24H	FA26H
BIT0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BIT1	0	0	1	0	0	0	1	1	1	0	1	1	1	1	1	0	0	0	1	0
BIT2	0	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	0
BIT3	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	1	0
BIT4	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	1	0
BIT5	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	1	1
BIT6	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
BIT7	0	1	1	1	0	1	1	1	1	1	0	1	1	1	0	0	0	0	1	0

	FA28H	FA2AH	FA2CH	FA2EH	FA30H	FA32H	FA34H	FA36H	FA38H	FA3AH	FA3CH	FA3EH	FA40H	FA42H	FA44H	FA46H	FA48H	FA4AH	FA4CH	FA4EH
BIT0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BIT1	1	1	1	1	1	0	0	1	1	0	1	1	1	1	1	0	1	1	1	0
BIT2	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1
BIT3	1	1	1	1	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	1
BIT4	0	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0	1	1	1	0
BIT5	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	1	0	0	0	1
BIT6	1	0	0	0	1	1	0	0	0	1	0	1	0	0	0	1	0	0	0	1
BIT7	0	1	1	1	0	0	1	1	1	0	0	1	0	0	0	0	1	1	1	0

	FA01H	FA03H	FA05H	FA07H	FA09H	FA0BH	FA0DH	FA0FH	FA11H	FA13H	FA15H	FA17H	FA19H	FA1BH	FA1DH	FA1FH	FA21H	FA23H	FA25H	FA27H
BIT0	0	1	1	1	0	1	1	1	1	0	0	1	1	1	0	1	1	1	0	0
BIT1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	1	0
BIT2	1	0	0	0	1	1	0	0	0	1	1	0	0	0	0	1	0	0	0	1
BIT3	1	0	0	0	1	1	1	1	1	0	1	0	0	0	0	1	0	0	0	1
BIT4	1	1	1	1	1	1	0	0	0	1	1	0	0	0	0	1	0	0	0	1
BIT5	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	1	0
BIT6	1	0	0	0	1	1	1	1	1	0	0	1	1	1	0	1	1	1	0	0
BIT7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	FA29H	FA2BH	FA2DH	FA2FH	FA31H	FA33H	FA35H	FA37H	FA39H	FA3BH	FA3DH	FA3FH	FA41H	FA43H	FA45H	FA47H	FA49H	FA4BH	FA4DH	FA4FH
BIT0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	0	0	0	1
BIT1	1	0	0	0	0	1	0	0	0	0	1	0	0	0	1	1	0	0	0	1
BIT2	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	1
BIT3	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1
BIT4	1	0	0	0	0	1	0	0	0	0	1	0	0	0	1	1	0	0	0	1
BIT5	1	0	0	0	0	1	0	0	0	0	1	0	0	0	1	1	0	0	0	1
BIT6	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	0	0	0	1
BIT7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



CHAPTER 13 INTERRUPT FUNCTIONS

13.1 Interrupt Function Types

The following two types of interrupt functions are used.

(1) Non-maskable interrupt

This interrupt is acknowledged unconditionally. It does not undergo interrupt priority control and is given top priority over all other interrupt requests.

A standby release signal is generated.

The non-maskable interrupt has one source of interrupt from the watchdog timer.

(2) Maskable interrupt

These interrupts undergo mask control. If two or more interrupts with the same priority are simultaneously generated, each interrupt has a predetermined priority (priority) as shown in Table 13-2.

A standby release signal is generated.

Table 13-1 lists the number of maskable sources in the μPD789830 Subseries.

Table 13-1. Maskable Interrupt Sources of μPD789830 Subseries

	External Interrupt Sources	Internal Interrupt Sources	Total
μPD789830	4	10	14
μPD78F9831	6		16

13.2 Interrupt Sources and Configuration

Table 13-2. Interrupt Sources

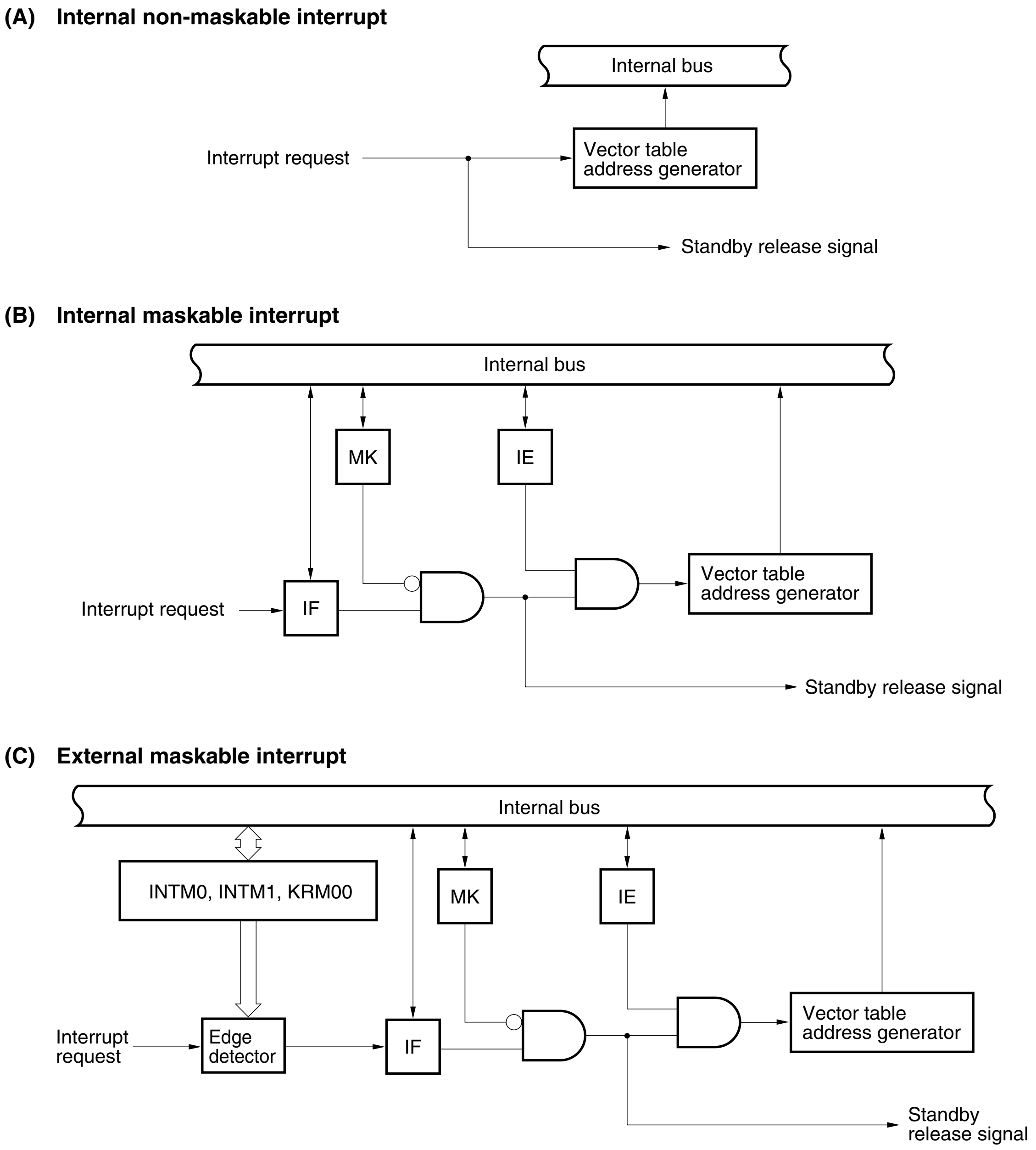
Interrupt Type	Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Non-maskable interrupt	–	INTWDT	Watchdog timer overflow (when watchdog timer mode 1 is selected)	Internal	0004H	(A)
Maskable interrupt	0	INTWDT	Watchdog timer overflow (when interval timer mode is selected)			(B)
	1	INTP0	Pin input edge detection	External	0006H	(C)
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTSER00	Occurrence of reception error of serial interface (UART00)	Internal	000CH	(B)
	5	INTSR00	Completion of reception by serial interface (UART00)		000EH	
	6	INTST00	Completion of transmission by serial interface (UART00)		0010H	
	7	INTTM40	Generation of match signal for 16-bit timer counter 40		0012H	
	8	INTTM41	Occurrence of overflow of 16-bit timer counter 40		0014H	
	9	INTTM4	Logical sum of match signal and overflow signal of 16-bit timer counter 40		0016H	
	10	INTTM00	Generation of match signal for 8-bit timer counter 00		0018H	
	11	INTWTI	Interval timer interrupt for watch timer		001AH	
	12	INTWT	Watch timer interrupt		001CH	
	13	INTKR00	Detection of key return signal	External	001EH	(C)
	14[Note 3]	INTP3	Pin input edge detection		0020H	
	15[Note 3]	INTP4			0022H	

Notes 1. The priority regulates which maskable interrupt is higher, when two or more maskable interrupts are requested simultaneously. With the μPD789830, 0 indicates the highest priority and 13 indicates the lowest priority. With the μPD78F9831, 0 indicates the highest priority, and 15 indicates the lowest priority.

2. Basic configuration types (A), (B), and (C) correspond to (A), (B), and (C) in Figure 13-1, respectively.

3. μPD78F9831 only

Figure 13-1. Basic Configuration of Interrupt Function



(A) Internal non-maskable interrupt

(B) Internal maskable interrupt

(C) External maskable interrupt

INTM0:	External interrupt mode register 0
INTM1[Note]:	External interrupt mode register 1
KRM00:	Key return mode register 00
IF:	Interrupt request flag
IE:	Interrupt enable flag
MK:	Interrupt mask flag

Note μPD78F9831 only

13.3 Interrupt Function Control Registers

The interrupt functions are controlled by the following registers.

- Interrupt request flag registers 0 and 1 (IF0, IF1)
- Interrupt mask flag registers 0 and 1 (MK0, MK1)
- External interrupt mode registers 0 and 1 (INTM0, INTM1[Note])
- Program status word (PSW)

Note μPD78F9831 only

Table 13-3 lists interrupt requests, the corresponding interrupt request flags, and interrupt mask flags.

Table 13-3. Interrupt Request Signals and Corresponding Flags

Interrupt Request Signal	Interrupt Request Flag	Interrupt Mask Flag
INTWDT	WDTIF	WDTMK
INTP0	PIF0	PMK0
INTP1	PIF1	PMK1
INTP2	PIF2	PMK2
INTSER00	SERIF00	SERMK00
INTSR00	SRIF00	SRMK00
INTST00	STIF00	STMK00
INTTM40	TMIF40	TMMK40
INTTM41	TMIF41	TMMK41
INTTM4	TMIF4	TMMK4
INTTM00	TMIF00	TMMK00
INTWTI	WTIIF	WTIMK
INTWT	WTIF	WTMK
INTKR00	KRIF00	KRMK00
INTP3[Note]	PIF3[Note]	PMK3[Note]
INTP4[Note]	PIF4[Note]	PMK4[Note]

Note μPD78F9831 only

(1) Interrupt request flag registers 0 and 1 (IF0, IF1)

An interrupt request flag is set to 1, when the corresponding interrupt request is issued, or when the related instruction is executed. It is cleared to 0, when the interrupt request is acknowledged, when a $\overline{\text{RESET}}$ signal is input, or when a related instruction is executed.

IF0 and IF1 are set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears IF0 and IF1 to 00H.

Figure 13-2. Format of Interrupt Request Flag Register (μPD789830)

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
IF0	TMIF40	STIF00	SRIF00	SERIF00	PIF2	PIF1	PIF0	WDTIF	FFE0H	00H	R/W
	7	6	<5>	<4>	<3>	<2>	<1>	<0>			
IF1	0	0	KRIF00	WTIF	WTIIF	TMIF00	TMIF4	TMIF41	FFE1H	00H	R/W

××IF	Interrupt request flag
0	No interrupt request signal has been issued.
1	An interrupt request signal has been issued; an interrupt request has been made.

Cautions 1. Bits 6 and 7 of IF1 must all be set to 0.

2. The WDTIF flag can be read- and write-accessed only when the watchdog timer is being used as an interval timer. It must be cleared to 0 if the watchdog timer is used in watchdog timer mode 1 or 2.

3. When port 2 is being used as an output port, and its output level is changed, an interrupt request flag is set, because this port is also used as an external interrupt input. To use port 2 in output mode, therefore, the interrupt mask flag must be set to 1 in advance.

Figure 13-3. Format of Interrupt Request Flag Register (μPD78F9831)

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
IF0	TMIF40	STIF00	SRIF00	SERIF00	PIF2	PIF1	PIF0	WDTIF	FFE0H	00H	R/W
	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>			
IF1	PIF4	PIF3	KRIF00	WTIF	WTIIF	TMIF00	TMIF4	TMIF41	FFE1H	00H	R/W

××IF	Interrupt request flag
0	No interrupt request signal has been issued.
1	An interrupt request signal has been issued; an interrupt request has been made.

Cautions 1. The WDTIF flag can be read- and write-accessed only when the watchdog timer is being used as an interval timer. It must be cleared to 0 if the watchdog timer is used in watchdog timer mode 1 or 2.

2. When ports 2 and 4 are being used as an output port, and its output level is changed, an interrupt request flag is set, because this port is also used as an external interrupt input. To use ports 2 and 4 in output mode, therefore, the interrupt mask flag must be set to 1 in advance.

(2) Interrupt mask flag registers 0 and 1 (MK0, MK1)

The interrupt mask flags are used to enable and disable the corresponding maskable interrupts.

MK0 and MK1 are set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets MK0 and MK1 to FFH.

Figure 13-4. Format of Interrupt Mask Flag Register (μPD789830)

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
MK0	TMMK40	STMK00	SRMK00	SERMK00	PMK2	PMK1	PMK0	WDTMK	FFE4H	FFH	R/W
	7	6	<5>	<4>	<3>	<2>	<1>	<0>			
MK1	1	1	KRMK00	WTMK	WTIMK	TMMK00	TMMK4	TMMK41	FFE5H	FFH	R/W

××MK	Interrupt servicing control
0	Enable interrupt servicing.
1	Disable interrupt servicing.

Cautions 1. Bits 6 and 7 of MK1 must all be set to 0.

2. The WDTMK flag can be read- and write-accessed only when the watchdog timer is being used as an interval timer. It must be cleared to 0 if the watchdog timer is used in watchdog timer mode 1 or 2.

3. When port 2 is being used as an output port, and its output level is changed, an interrupt request flag is set, because this port is also used as an external interrupt input. To use port 2 in output mode, therefore, the interrupt mask flag must be set to 1 in advance.

Figure 13-5. Format of Interrupt Mask Flag Register (μPD78F9831)

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	Address	After reset	R/W
MK0	TMMK40	STMK00	SRMK00	SERMK00	PMK2	PMK1	PMK0	WDTMK	FFE4H	FFH	R/W
	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>			
MK1	PMK4	PMK3	KRMK00	WTMK	WTIMK	TMMK00	TMMK4	TMMK41	FFE5H	FFH	R/W

××MK	Interrupt servicing control
0	Enable interrupt servicing.
1	Disable interrupt servicing.

Cautions 1. The WDTMK flag can be read- and write-accessed only when the watchdog timer is being used as an interval timer. It must be cleared to 0 if the watchdog timer is used in watchdog timer mode 1 or 2.

2. When ports 2 and 4 are being used as an output port, and its output level is changed, an interrupt request flag is set, because this port is also used as an external interrupt input. To use ports 2 and 4 in output mode, therefore, the interrupt mask flag must be set to 1 in advance.

(3) External interrupt mode register 0 (INTM0)

INTM0 is used to specify a valid edge for INTP0 to INTP2.

INTM0 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears INTM0 to 00H.

Figure 13-6. Format of External Interrupt Mode Register 0

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
INTM0	ES21	ES20	ES11	ES10	ES01	ES00	0	0	FFECH	00H	R/W

ES21	ES20	INTP2 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES11	ES10	INTP1 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES01	ES00	INTP0 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

Cautions 1. Bits 0 and 1 must all be set to 0.

2. Set the corresponding interrupt mask flag to 1 to disable interrupts before setting INTM0.

To enable interrupts, clear to 0 the corresponding interrupt request flag, then the corresponding interrupt mask flag.

(4) External interrupt mode register 1 (INTM1)[Note]

INTM1 is used to specify a valid edge for INTP3 and INTP4.

INTM1 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears INTM1 to 00H.

Note μPD78F9831 only

Figure 13-7. Format of External Interrupt Mode Register 1 (μPD78F9831)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
INTM1	0	0	0	0	ES41	ES40	ES31	ES30	FFEDH	00H	R/W

ES41	ES40	INTP4 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES31	ES30	INTP3 valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

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

2. Set the corresponding interrupt mask flag to 1 to disable interrupts before setting INTM1.

To enable interrupts, clear to 0 the corresponding interrupt request flag, then the corresponding interrupt mask flag.

(5) Program status word (PSW)

The program status word is used to hold the instruction execution result and the current status of the interrupt requests. The IE flag, used to enable and disable maskable interrupts, is mapped to PSW.

PSW can be read- and write-accessed in 8-bit units, as well as in 1-bit units when using bit manipulation instructions and dedicated instructions (EI and DI). When a vectored interrupt is acknowledged, PSW is automatically saved to a stack, and the IE flag is reset to 0.

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

Figure 13-8. Program Status Word Configuration

Symbol	7	6	5	4	3	2	1	0	After reset
PSW	IE	Z	0	AC	0	0	1	CY	02H

Used in the execution of ordinary instructions

IE	Whether to enable/disable interrupt acknowledgement
0	Disable
1	Enable

(6) Key return mode register 00 (KRM00)

KRM00 is used to select a pin that detects a key return signal (falling edge of port 3).

KRM00 is set with an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears KRM00 to 00H.

Figure 13-9. Format of Key Return Mode Register 00

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
KRM00	0	0	0	KRM004	KRM003	KRM002	KRM001	KRM000	FFF5H	00H	R/W

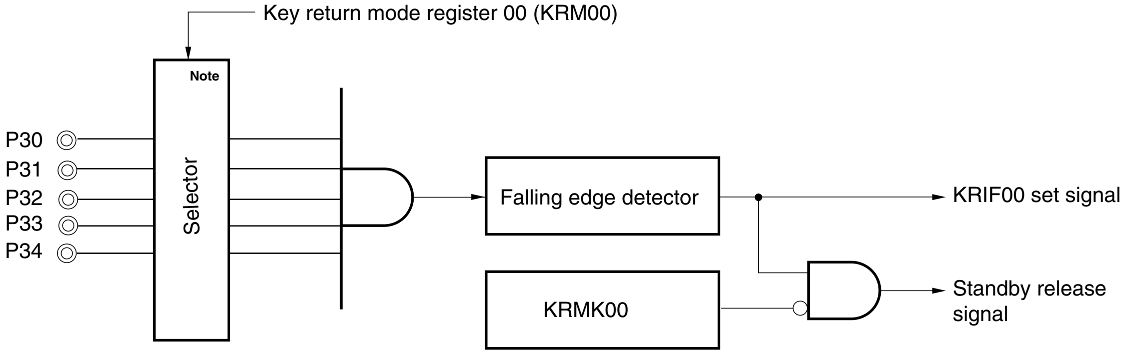
KRM00n	P3n key return signal detect selection
0	Not detected
1	Detected (falling edge of port 3)

Cautions 1. Bits 5 to 7 must all be set to 0.

2. Set bit 5 (KRMK00) of MK1 to 1 to disable interrupts before setting KRM00. After setting KRM00, set bit 5 (KRIF00) of IF1 to 1, then clear KRMK00 to 0 to enable the interrupt.

Remark n = 0 to 4

Figure 13-10. Block Diagram of Falling Edge Detector



Note Selector that selects a pin used to input the falling edge.

13.4 Interrupt Servicing Operation

13.4.1 Non-maskable interrupt request acknowledgement operation

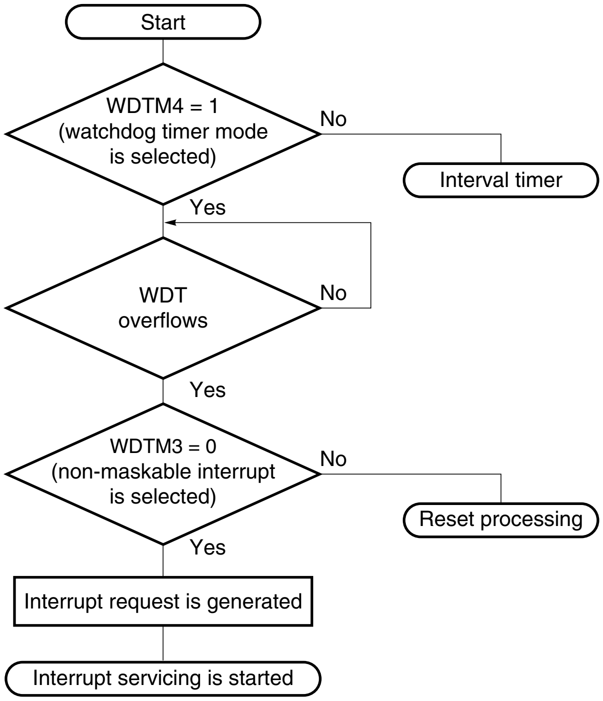
The non-maskable interrupt is unconditionally acknowledged even when interrupts are disabled. It is not subject to interrupt priority control and takes precedence over all other interrupts.

When the non-maskable interrupt request is acknowledged, PSW and PC are saved to the stack in that order, the IE flag is reset to 0, the contents of the vector table are loaded to the PC, and then program execution branches.

Figure 13-11 shows the flowchart from non-maskable interrupt request generation to acknowledgement. Figure 13-12 shows the timing of non-maskable interrupt request acknowledgement. Figure 13-13 shows the acknowledgement operation if multiple non-maskable interrupts are generated.

Caution During a non-maskable interrupt servicing program execution, do not input another non-maskable interrupt request; otherwise the interrupt servicing program will be interrupted and the new interrupt request will be acknowledged.

Figure 13-11. Flowchart from Non-Maskable Interrupt Request Generation to Acknowledgement



WDTM: Watchdog timer mode register

WDT: Watchdog timer

Figure 13-12. Timing of Non-Maskable Interrupt Request Acknowledgement

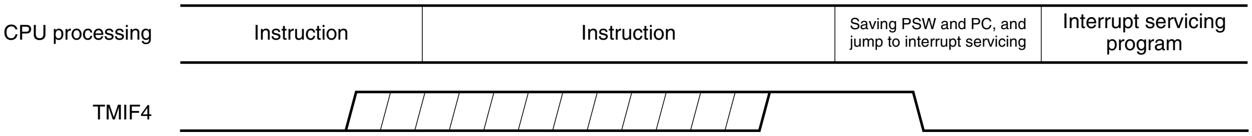
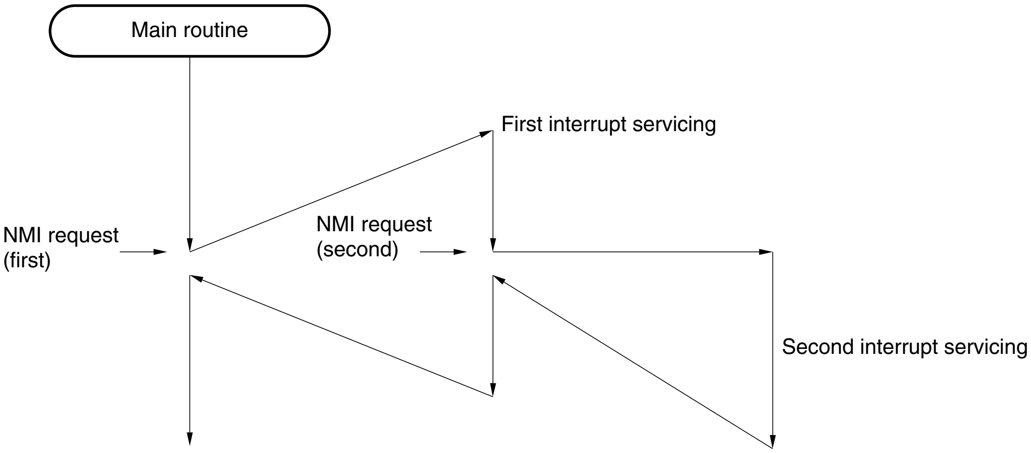


Figure 13-13. Acknowledgement of Non-Maskable Interrupt Request



13.4.2 Maskable interrupt acknowledgement operation

A maskable interrupt 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 13-4.

See Figures 13-15 and 13-16 for the interrupt request acknowledgement timing.

Table 13-4. Time from Generation of Maskable Interrupt Request to Servicing

Minimum Time	Maximum Time[Note]
9 clocks	19 clocks

Note The wait time is maximum when an interrupt request is generated immediately before BT and 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 interrupt request assigned the highest priority.

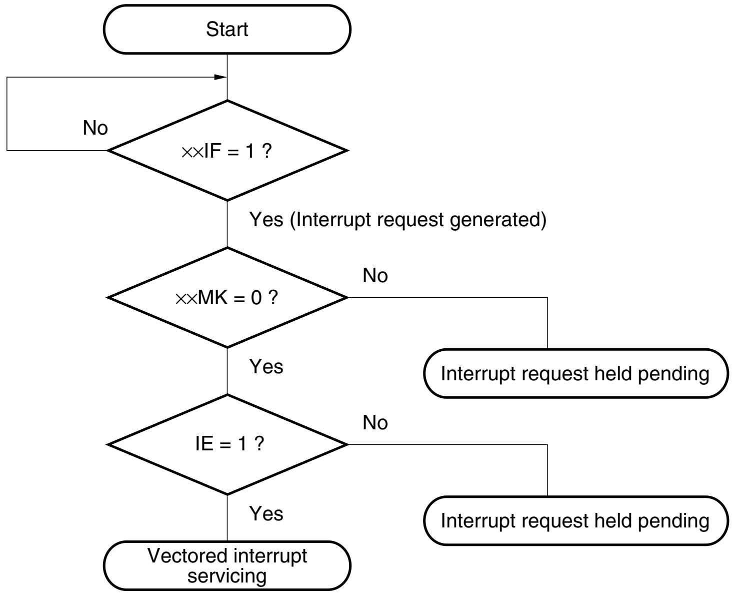
An interrupt that is held pending is acknowledged when the status where it can be acknowledged is set.

Figure 13-14 shows the algorithm of acknowledging interrupt requests.

When a maskable interrupt request is acknowledged, the contents of PSW and PC are saved to the stack in that order, the IE flag is reset to 0, and the data in the vector table determined for each interrupt request is loaded to the PC, and execution branches.

To return from interrupt servicing, use the RETI instruction.

Figure 13-14. Interrupt Request Acknowledgement Program Algorithm

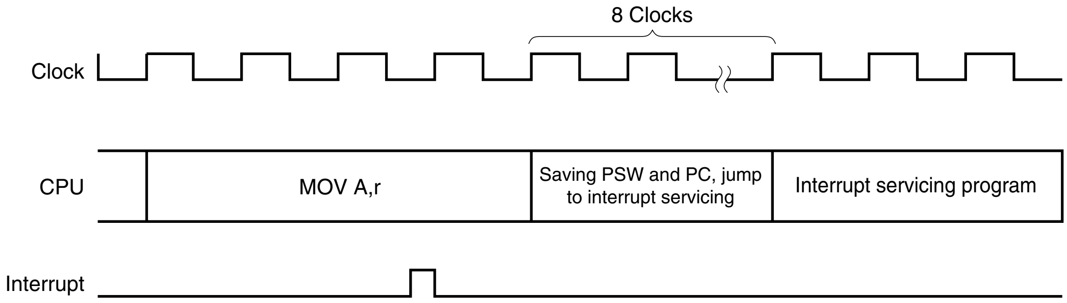

Start
No
××IF = 1 ?
Yes (Interrupt request generated)
××MK = 0 ?
No
Yes
Interrupt request held pending
IE = 1 ?
No
Yes
Interrupt request held pending
Vectored interrupt servicing

××IF: Interrupt request flag

××MK: Interrupt mask flag

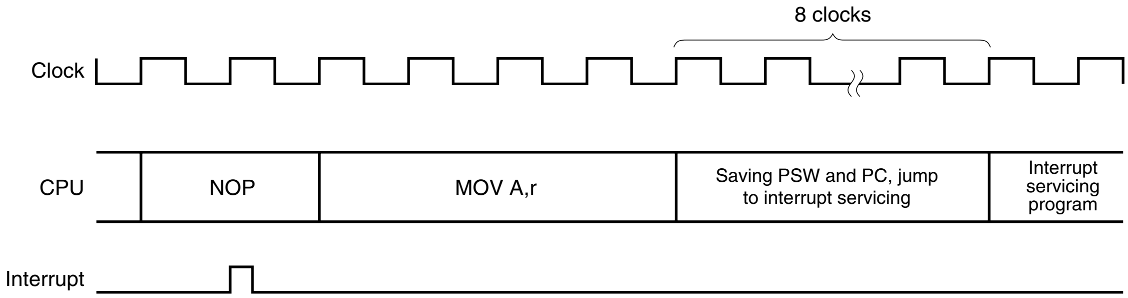
IE: Flag to control maskable interrupt request acknowledgement (1 = enable, 0 = disable)

Figure 13-15. Interrupt Request Acknowledgement Timing (Example of MOV A,r)


8 Clocks
Clock
CPU
MOV A,r
Saving PSW and PC, jump to interrupt servicing
Interrupt servicing program
Interrupt

If an interrupt request flag (××IF) is set before an instruction clock n (n = 4 to 10) under execution becomes n – 1, the interrupt is acknowledged after the instruction under execution completes. Figure 13-15 shows an example of the interrupt request acknowledgement timing for an 8-bit data transfer instruction MOV A,r. Since this instruction is executed for 4 clocks, if an interrupt occurs for 3 clocks after the execution starts, the interrupt acknowledgement processing is performed after the MOV A,r instruction is completed.

Figure 13-16. Interrupt Request Acknowledgement Timing (When Interrupt Request Flag Is Generated at the Last Clock During Instruction Execution)


8 clocks
Clock
CPU
NOP
MOV A,r
Saving PSW and PC, jump to interrupt servicing
Interrupt servicing program
Interrupt

If an interrupt request flag (××IF) is set at the last clock of the instruction, the interrupt acknowledgement processing starts after the next instruction is executed.

Figure 13-16 shows an example of the interrupt acknowledgement timing for an interrupt request flag that is set at the second clock of NOP (2-clock instruction). In this case, the MOV A,r instruction after the NOP instruction is executed, and then the interrupt acknowledgement processing is performed.

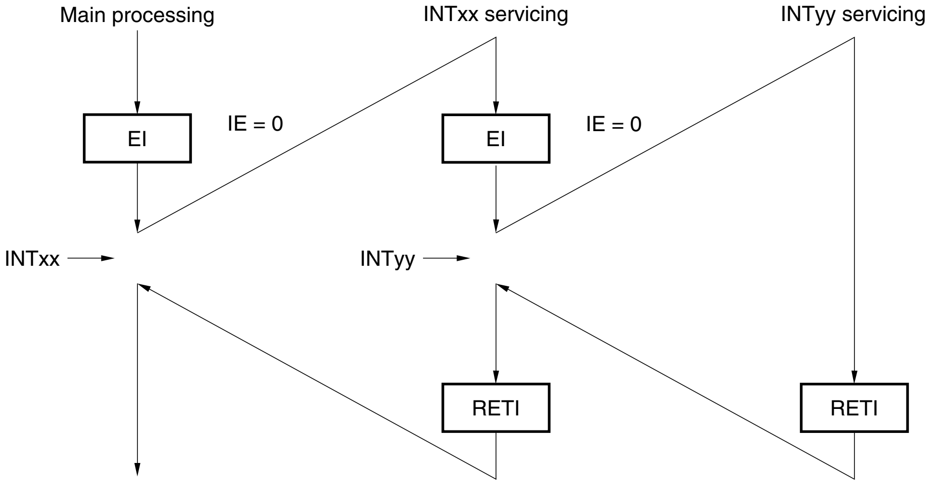
Caution Interrupt requests are reserved while interrupt request flag registers 0 or 1 (IF0 or IF1) or interrupt mask flag register 0 or 1 (MK0 or MK1) is being accessed.

13.4.3 Multiple interrupt servicing

Multiple interrupt servicing in which another interrupt is acknowledged while an interrupt is serviced can be processed by priority. When two or more interrupts are generated at once, interrupt servicing is performed according to the priority assigned to each interrupt request in advance (see **Table 13-2**).

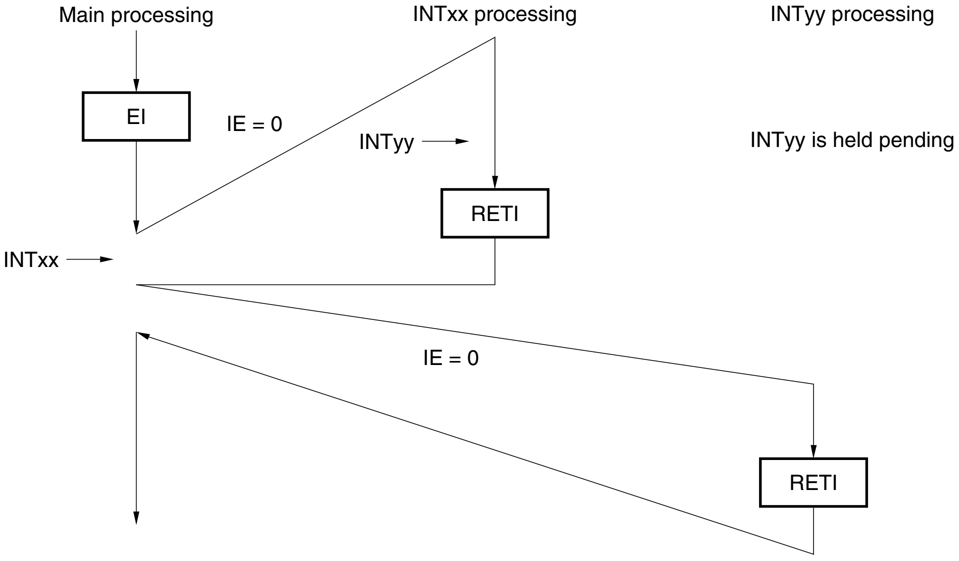
Figure 13-17. Example of Multiple Interrupt

Example 1. A multiple interrupt is acknowledged



During interrupt INTxx processing, interrupt request INTyy is acknowledged, and a multiple interrupt is generated. An EI instruction is issued before each interrupt request acknowledgement, and the interrupt request acknowledgement enable state is set.

Example 2. A multiple interrupt is not generated because interrupts are not enabled



Because interrupts are not enabled in interrupt INTxx servicing (an EI instruction is not issued), interrupt request INTyy is not acknowledged, and a multiple interrupt is not generated. The INTyy request is held pending and acknowledged after the INTxx servicing is performed.

IE = 0: Interrupt request acknowledgement

13.4.4 Pending interrupt request

Some instructions may hold the acknowledgement of an instruction request until the completion of the execution of the next instruction even if the interrupt request (maskable interrupt, non-maskable interrupt, and external interrupt) is generated during the execution. The following shows such instructions (interrupt request hold instruction).

- Manipulation instruction for interrupt request flag registers 0 and 1 (IF0, IF1)
- Manipulation instruction for interrupt mask flag registers 0 and 1 (MK0, MK1)

CHAPTER 14 STANDBY FUNCTION

14.1 Standby Function and Configuration

14.1.1 Standby function

The standby function is used to reduce the power consumption of the system and the following two modes are available:

(1) HALT mode

This mode is set when the HALT instruction is executed. 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 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. STOP mode stops the system clock oscillator and stops the entire system. The power consumption of the CPU can be substantially reduced in this mode.

The low voltage (V_{DD} = 1.8 V) of the data memory can be retained. Therefore, this mode is useful for retaining the contents of the data memory at a low current.

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 STOP mode has been released. If processing must be resumed immediately by using an interrupt request, therefore, use HALT mode.

In both modes, the previous contents of the registers, flags, and data memory before stabilizing 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 STOP mode, be sure to stop the operations of the peripheral hardware, and then execute the STOP instruction.

14.1.2 Standby function control register

The wait time after STOP mode is released upon interrupt request until the oscillation stabilizes is controlled with the oscillation stabilization time selection register (OSTS).

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

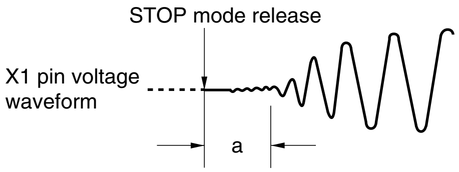
$\overline{\text{RESET}}$ input sets OSTS to 04H. However, the oscillation stabilization time after $\overline{\text{RESET}}$ input is $2^{15}/f_X$.

Figure 14-1. 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	$2^{12}/f_X$ (1.02 ms)
0	1	0	$2^{15}/f_X$ (8.19 ms)
1	0	0	$2^{17}/f_X$ (32.8 ms)
Other than above			Setting prohibited

Caution The wait time after STOP mode is released does not include the time from STOP mode release to clock oscillation start ("a" in the figure below), regardless of release by $\overline{\text{RESET}}$ input or by interrupt generation.



Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 3.58 MHz.

14.2 Standby Function Operation

14.2.1 HALT mode

(1) Setting and operation status of HALT mode

HALT mode is set by executing the HALT instruction.

The operation status in HALT mode is shown in the following table.

Table 14-1. Operation Statuses in HALT Mode

Item	HALT Mode Operation Status While Main System Clock Is Operating		HALT Mode Operation Status While Subsystem Clock Is Operating	
	While Subsystem Clock Is Operating	While Subsystem Clock Is Not Operating	While Main System Clock Is Operating	While Main System Clock Is Not Operating
Main system clock	Oscillation enabled			Oscillation disabled
CPU	Operation disabled			
Port (output latch)	Remains in the state existing before the selection of HALT mode			
16-bit timer counter	Operation enabled			Operation disabled
8-bit timer counter	Operation enabled			Operation disabled
Watch timer	Operation enabled	Operation enabled[Note 1]	Operation enabled	Operation enabled[Note 2]
Watchdog timer	Operation enabled		Operation disabled	
Clock output circuit	Operation enabled	Operation enabled[Note 1]	Operation enabled	Operation enabled[Note 3]
Serial interface	Operation enabled			Operation disabled
LCD controller/driver	Operation enabled	Operation enabled[Note 1]	Operation enabled	Operation enabled[Note 2]
External interrupt	Operation enabled[Note 4]			

Notes 1. Operation is enabled while the main system clock is selected.

2. Operation is enabled while the subsystem clock is selected.

3. Operation is enabled in the pulse clock output circuit while the subsystem clock is selected.

4. Maskable interrupt that is not masked

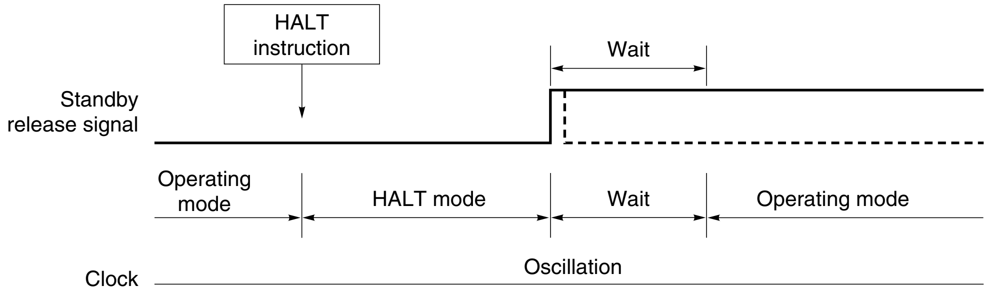
(2) Releasing HALT mode

HALT mode can be released by the following three types of sources.

(a) Releasing by unmasked interrupt request

HALT mode is released by an unmasked interrupt request. In this case, if the interrupt is enabled to be acknowledged, vectored interrupt servicing is performed. If the interrupt acknowledgement is disabled, the instruction at the next address is executed.

Figure 14-2. Releasing HALT Mode by Interrupt



Remarks 1. The broken line indicates the case where the interrupt request that has released 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

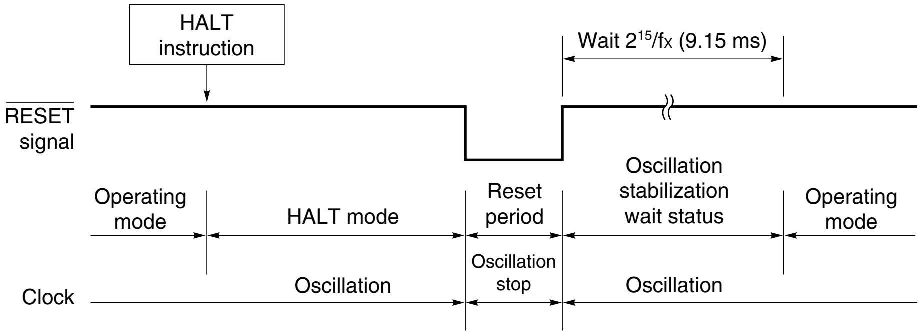
(b) Releasing by non-maskable interrupt request

HALT mode is released regardless of whether the interrupt is enabled or disabled, and vectored interrupt servicing is performed.

(c) Releasing by $\overline{\text{RESET}}$ input

When 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 14-3. Releasing HALT Mode by $\overline{\text{RESET}}$ Input



Remarks **1.** fx: Main system clock oscillation frequency

2. The parenthesized values apply to operation at fx = 3.58 MHz.

Table 14-2. Operation After Release of 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

14.2.2 STOP mode

(1) Setting and operation status of STOP mode

STOP mode is set by executing the STOP instruction.

Cautions 1. When STOP mode is set, the X2 pin is internally pulled up to V_{DD0} or V_{DD1} to suppress the current leakage of the oscillator block. Therefore, do not use STOP mode in a system where the external clock is used as the system clock.

2. Because standby mode can be released by an interrupt request signal, 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 STOP mode is set, therefore, 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 operating mode is set.

The operation status in STOP mode is shown in the following table.

Table 14-3. Operation Statuses in STOP Mode

Item	STOP Mode Operation Status While Main System Clock Is Operating	
	While Subsystem Clock Is Operating	While Subsystem Clock Is Not Operating
Main system clock	Oscillation disabled	
CPU	Operation disabled	
Port (output latch)	Remains in the state existing before the selection of STOP mode	
16-bit timer counter	Operation disabled	
8-bit timer counter	Operation disabled	
Watch timer	Operation enabled[Note 1]	Operation disabled
Watchdog timer	Operation disabled	
Clock output circuit	Operation enabled[Note 2]	Operation disabled
Serial interface	Operation disabled	
LCD controller/driver	Operation enabled[Note 1]	Operation disabled
External interrupt	Operation enabled[Note 3]	

Notes 1. Operation is enabled while the subsystem clock is selected.

2. Operation is enabled in the pulse clock output circuit while the subsystem clock is selected.

3. Maskable interrupt that is not masked

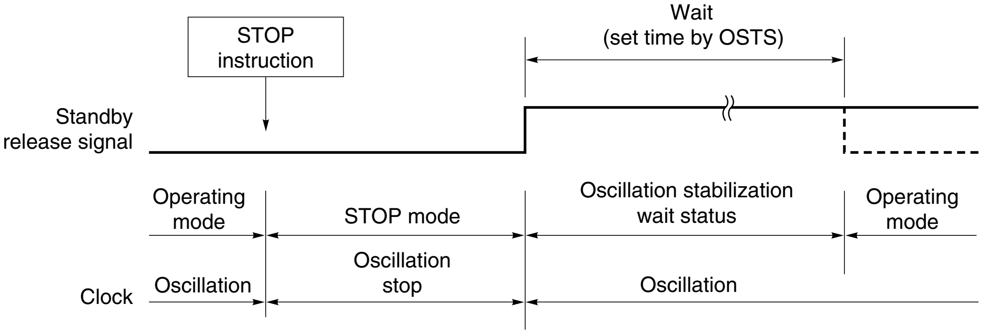
(2) Releasing STOP mode

STOP mode can be released by the following two types of sources.

(a) Releasing by unmasked interrupt request

STOP mode can be released by an unmasked interrupt request. In this case, if the interrupt is enabled to be acknowledged, vectored interrupt servicing is performed, after the oscillation stabilization time has elapsed. If the interrupt acknowledgement is disabled, the instruction at the next address is executed.

Figure 14-4. Releasing STOP Mode by Interrupt

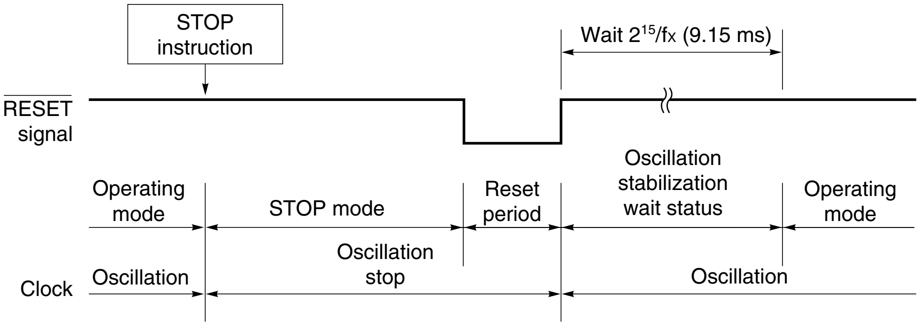


Remark The broken line indicates the case where the interrupt request that has released standby mode is acknowledged.

(b) Releasing by $\overline{\text{RESET}}$ input

When STOP mode is released by the $\overline{\text{RESET}}$ signal, the reset operation is performed after the oscillation stabilization time has elapsed.

Figure 14-5. Releasing STOP Mode by $\overline{\text{RESET}}$ Input



Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 3.58 MHz.

Table 14-4. Operation After Release of 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 15 RESET FUNCTION

The following two operations are available to generate reset signals.

(1) External reset input with $\overline{\text{RESET}}$ pin

(2) Internal reset by inadvertent program loop time detected with watchdog timer

External and internal reset have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H by reset signal input.

When a low level is input to the $\overline{\text{RESET}}$ pin or the watchdog timer overflows, a reset is applied and each hardware is set to the status shown in Table 15-1. Each pin has a high impedance during reset input or during oscillation stabilization time just after reset clear.

When a high level is input to the $\overline{\text{RESET}}$ pin, the reset is cleared and program execution is started after the oscillation stabilization time has elapsed (2^{15}/fx). The reset applied by the watchdog timer overflow is automatically cleared after reset, and program execution is started after the oscillation stabilization time has elapsed (2^{15}/fx) (see **Figures 15-2** to **15-4**).

Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.

2. When STOP mode is cleared by reset, STOP mode contents are held during reset input. However, the port pins become high impedance.

Figure 15-1. Block Diagram of Reset Function

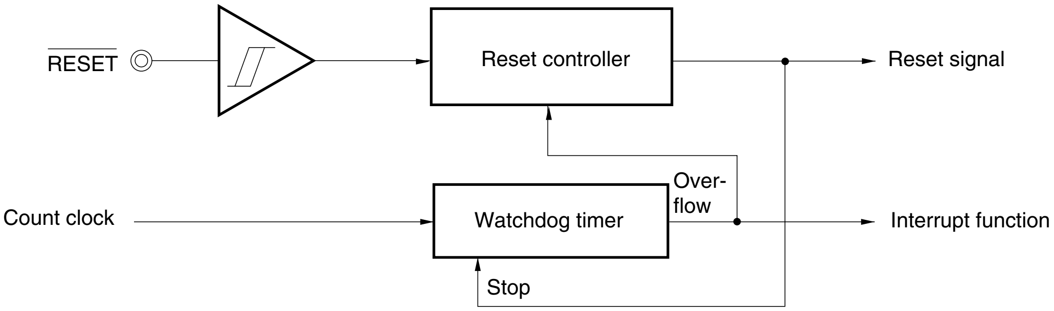


Figure 15-2. Reset Timing by $\overline{\text{RESET}}$ Input

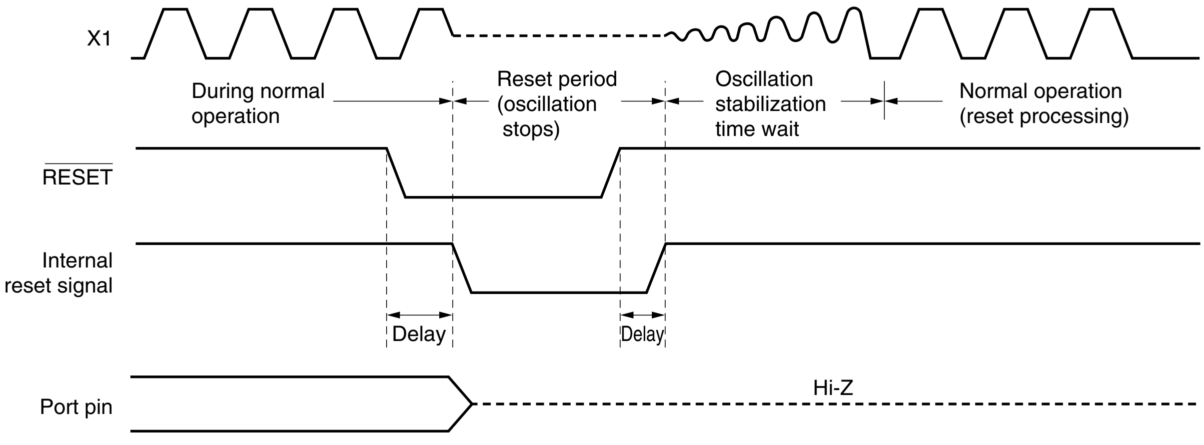

X1
During normal operation
Reset period (oscillation stops)
Oscillation stabilization time wait
Normal operation (reset processing)
RESET
Internal reset signal
Delay
Delay
Port pin
Hi-Z

Figure 15-3. Reset Timing by Overflow in Watchdog Timer

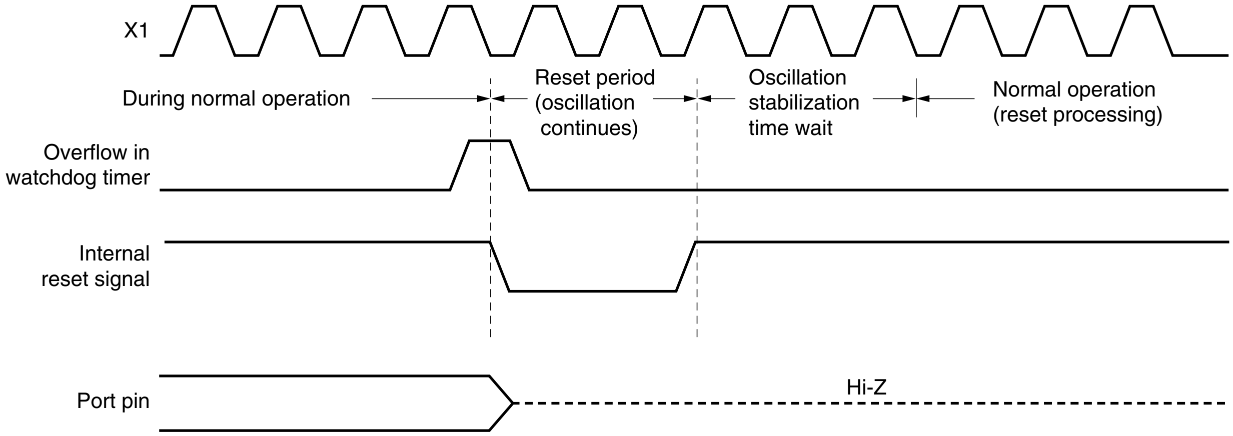

X1
During normal operation
Reset period (oscillation continues)
Oscillation stabilization time wait
Normal operation (reset processing)
Overflow in watchdog timer
Internal reset signal
Port pin
Hi-Z

Figure 15-4. Reset Timing by $\overline{\text{RESET}}$ Input in STOP Mode

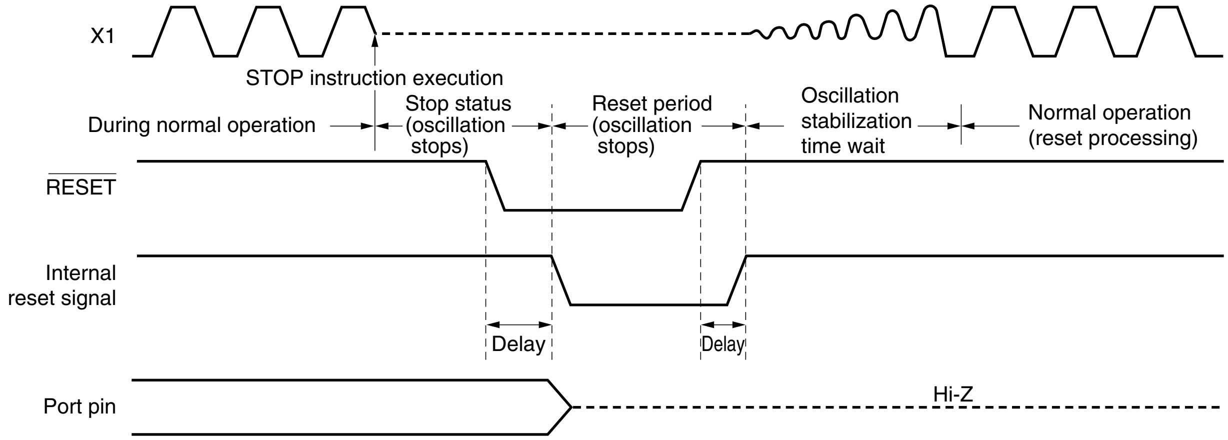

X1
STOP instruction execution
During normal operation
Stop status (oscillation stops)
Reset period (oscillation stops)
Oscillation stabilization time wait
Normal operation (reset processing)
RESET
Internal reset signal
Delay
Delay
Port pin
Hi-Z

Table 15-1. State of Hardware After Reset

Hardware		State After Reset
Program counter (PC)[Note 1]		Loaded with the contents of the reset vector table (0000H, 0001H)
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined[Note 2]
	General-purpose register	Undefined[Note 2]
Ports (P0 to P3, P4[Note 3], P5) (output latch)		00H
Port mode registers (PM0 to PM3, PM4[Note 3], PM5)		FFH
Pull-up resistor option register (PU0)		00H
Processor clock control register (PCC)		02H
Suboscillation mode register (SCKM)		00H
Subclock control register (CSS)		00H
Oscillation stabilization time selection register (OSTS)		04H
16-bit timer counter	Compare register (CR40)	0000H
	Control register (TMC40)	00H
8-bit timer counter	Timer register (TM00)	00H
	Compare register (CR00)	Undefined
	Mode control register (TMC00)	00H
Watch timer	Mode control register (WTM)	00H
Watchdog timer	Timer clock selection register (TCL2)	00H
	Mode register (WDTM)	00H
Clock output circuit	Control register (PBS0)	00H
Serial interface	Asynchronous serial interface mode register (ASIM00)	00H
	Asynchronous serial interface status register (ASIS00)	00H
	Baud rate generator control register (BRGC00)	00H
	Transmission shift register (TXS00)	FFH
	Reception buffer register (RXB00)	
LCD controller/driver	Mode register (LCDM20)	00H
	Alternate port function switching register (PF5)	00H
	Clock selection register (LCDC20)	00H
Interrupts	Request flag registers (IF0, IF1)	00H
	Mask flag registers (MK0, MK1)	FFH
	External interrupt mode register (INTM0)	00H
	External interrupt mode register (INTM1)[Note 3]	00H
	Key return mode register (KRM00)	00H

Notes 1. While a reset signal is being input, and during the oscillation stabilization period, the contents of the PC will be undefined, while the remainder of the hardware will be the same as after the reset.

2. In standby mode, the RAM enters the hold state after a reset.

3. μPD78F9831 only

CHAPTER 16 μPD78F9831

The μPD78F9831 is produced by replacing the internal ROM of the mask ROM version μPD789830 with larger flash memory, and by adding I/O ports to the μPD789830. Unlike bare chips of mask ROM versions, the shipped μPD78F9831 is contained in a 100-pin plastic LQFP package. Table 16-1 lists differences between the μPD78F9831 and μPD789830.

Table 16-1. Differences Between μPD78F9831 and μPD789830

Item		Flash Memory Version	Mask ROM Version
		μPD78F9831	μPD789830
Internal memory	ROM	48 KB (flash memory)	24 KB (mask ROM)
	RAM	2 KB	1 KB
	LCD display RAM	80 bytes	
I/O ports		Total: 38 port pins P00 to P07, P10 to P17, P20 to P26, P30 to P34, P40, P41, P50 to P57	Total: 30 port pins P00 to P07, P10, P11, P20 to P26, P30 to P34, P50 to P57
External interrupt input pins		Total: 5 pins INTP0 to INTP4	Total: 3 pins INTP0 to INTP2
V_{PP} pin		Provided	Not provided
Form of shipment		100-pin plastic LQFP	88-pin bare chip
Electrical characteristics		Refer to **CHAPTER 18 ELECTRICAL SPECIFICATIONS**.	

16.1 Flash Memory Programming

The on-chip program memory in the μPD78F9831 is a flash memory.

The flash memory can be written with the μPD78F9831 mounted on the target system (on-board). Connect the dedicated flash writer (Flashpro III (part number: PG-FR3)) to the host machine and target system to write the flash memory.

16.1.1 Selecting communication mode

The flash memory is written by using Flashpro III and by means of serial communication. Select a communication mode from those listed in Table 16-2. To select a communication mode, the format shown in Figure 16-1 is used. Each communication mode is selected by the number of V_{PP} pulses shown in Table 16-2.

Table 16-2. Communication Mode

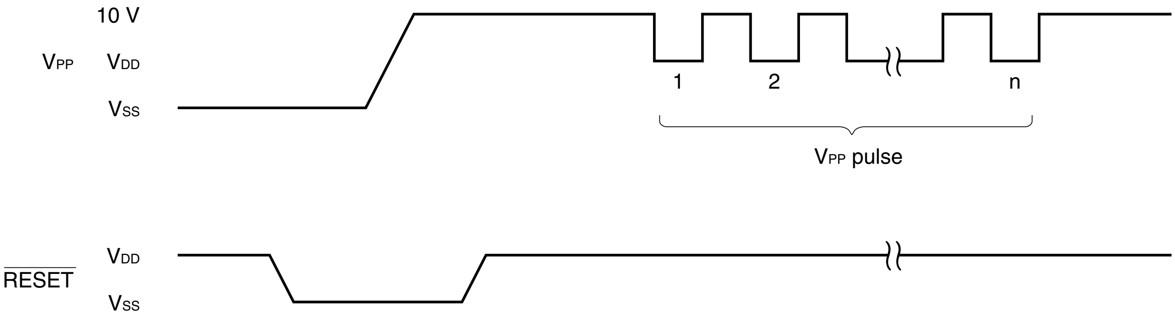
Communication Mode	Pins Used[Note 1]	Number of V_{PP} Pulses
UART	TxD00/P25 RxD00/P26	8
Pseudo 3-wire mode[Note 2]	P10 (Serial clock input) P11 (Serial data output) P12 (Serial data input)	12

★ **Notes 1.** Shifting to the flash memory programming mode sets all pins not used for flash memory programming to the same state as the immediately after reset. If the external device connected to each port does not acknowledge the state immediately after reset, pin handling such as connecting to V_{DD} or V_{SS} via a resistor is required.

2. Serial transfer is performed by controlling a port by software.

Caution Be sure to select a communication mode depending on the number of V_{PP} pulses shown in Table 16-2.

Figure 16-1. Format of Communication Mode Selection



16.1.2 Function of flash memory programming

By transmitting/receiving commands and data in the selected communication mode, operations such as writing to the flash memory are performed. Table 16-3 shows the major functions of flash memory programming.

Table 16-3. Functions of Flash Memory Programming

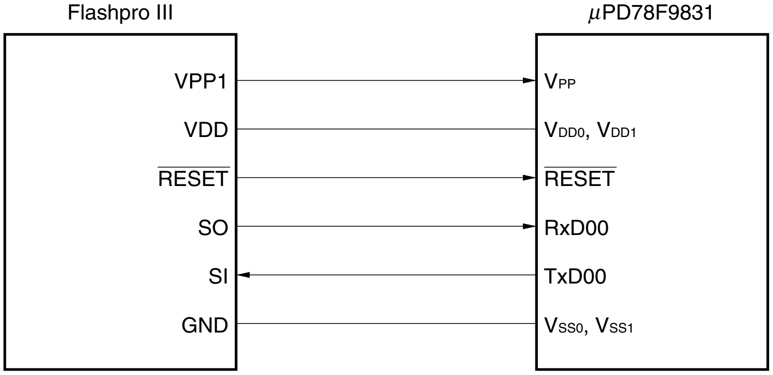
Function	Description
Batch erase	Erases all contents of memory
Batch blank check	Checks erased state of entire memory
Data write	Writes to flash memory based on write start address and number of data written (number of bytes)
Batch verify	Compares all contents of memory with input data

16.1.3 Flashpro III connection

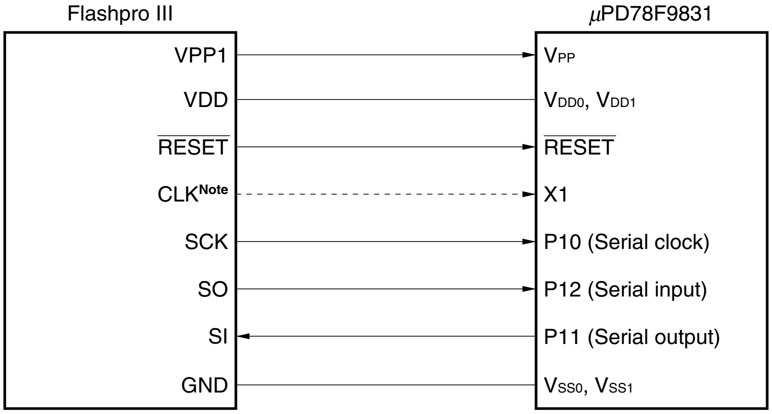
Connection between Flashpro III and the μPD78F9831 differs depending on the communication mode (UART or pseudo 3-wire mode). Figure 16-2 shows the connection in the respective modes.

★ **Figure 16-2. Flashpro III Connection Example**

(a) UART



(b) Pseudo 3-Wire Mode



Note Connect the CLK pin when the system clock is input from Flashpro III. When a resonator has already been connected to the X1 pin, there is no need to connect the CLK pin to the X1 pin.

Caution Be sure to connect the V_{DD} pin to the VDD pin of Flashpro III, even if it is already connected to the power supply. When using the power supply, be sure to apply voltage before starting programming.

Remark There is no need to connect the CLK pin to the X1 pin, because the clock of the resonator connected to the X1 pin is always used when UART is used in Flashpro III.

16.1.4 Setting with Flashpro III

Perform the setting shown in Table 16-4 when writing data to the flash memory by using Flashpro III.

Table 16-4. Setting with Flashpro III

Communication Mode	Setting with Flashpro III			V_{PP} Pulse Count[Note 1]
UART	Type		78 K(2)	8
	RAM		128	
	ROM		Flash	
	START ADDRESS		0	
	END ADDRESS		BFFF	
	COMM PORT		UART ch-0	
	CPU CLK		On Target Board	
	On Target Board		4.91/5.0 MHz	
	UART BPS		9,600 bps[Note 2]	
Pseudo 3-wire mode	Type		78 K(2)	12
	RAM		128	
	ROM		Flash	
	START ADDRESS		0	
	END ADDRESS		BFFF	
	COMM PORT		Port A	
	CPU CLK		On Target Board	
			In Flashpro	
	On Target Board		4.1/5.0 MHz	
	On Target Board: 5.0 MHz	SIO CLK	3.6 kHz (MAX.)	
	On Target Board: 4.1 MHz	SIO CLK	3.1 kHz (MAX.)	
	In Flashpro		1.56 MHz	
	SIO CLK		1 kHz	

Notes 1. The number of V_{PP} pulses supplied from Flashpro III when serial communication is initialized. These pulses determine the pins used for communication.

2. Select 9,600 bps, 19,200 bps, 38,400 bps, or 76,800 bps.

Remark COMM PORT: Selects serial port.
SIO CLK: Selects serial clock frequency.
CPU CLK: Selects source of CPU clock to be input.

★ ### 16.1.5 On-board pin connections

When programming on the target system, provide a connector on the target system to connect to the dedicated flash programmer.

There may be cases in which an on-board function that switches from the normal operation mode to flash memory programming mode is required.

<VPP pin>

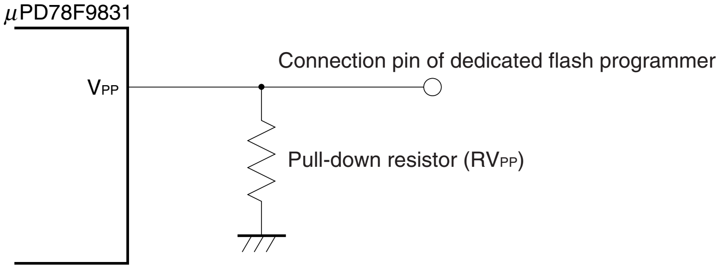
Input 0 V to the VPP pin in the normal operation mode. A writing voltage of 10.0 V (TYP.) is supplied to the VPP pin in the flash memory programming mode. Therefore, Handle this pin in either of the following ways (1) and (2).

(1) Connect a pull-down resistor of RVPP = 10 kΩ to the VPP pin.

(2) Set the jumper on the board to switch the input of VPP pin to the programmer side or directly to GND.

The following shows an example of VPP pin connection.

Figure 16-3. VPP Pin Connection Example


μPD78F9831
VPP
Connection pin of dedicated flash programmer
Pull-down resistor (RVPP)

<Serial interface pins>

The following shows the pins used by each serial interface.

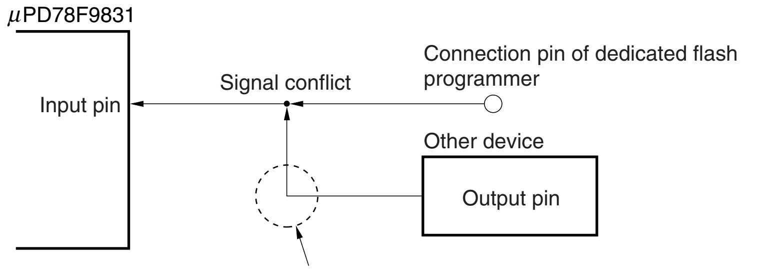
Serial Interface	Pins Used
UART	TxD00/P25, RxD00/P26
Pseudo 3-wire	P10, P11, P12

Note that signal conflict or malfunction of other devices may occur when an on-board serial interface pin that is connected to another device is connected to the dedicated flash programmer.

(1) Signal conflict

A signal conflict occurs if the dedicated flash programmer (output) is connected to a serial interface pin (input) connected to another device (output). To prevent this signal conflict, isolate the connection with the other device or put the other device in the output high impedance status.

Figure 16-4. Signal Conflict (Serial Interface Input Pin)

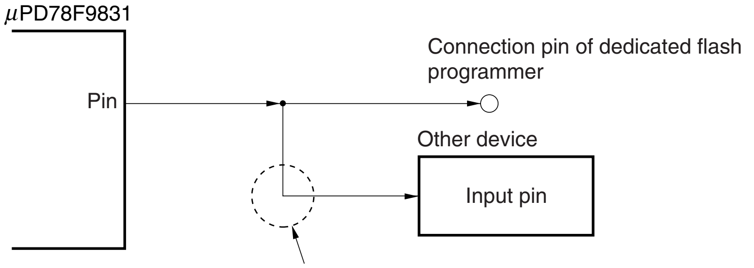


In the flash memory programming mode, the signal output by another device and the signal sent by the dedicated flash programmer conflict. To prevent this, isolate the signal on the device side.

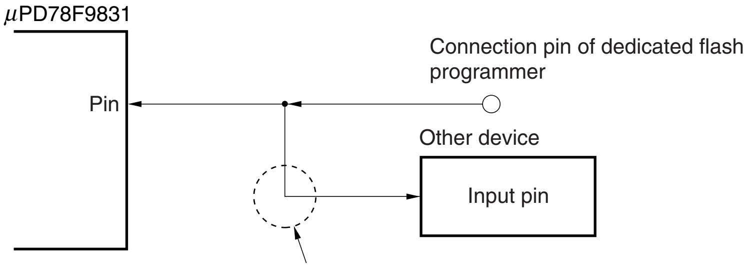
(2) Malfunction of another device

When the dedicated flash programmer (output or input) is connected to a serial interface pin (input or output) connected to another device (input), a signal may be output to the device, causing a malfunction. To prevent such malfunction, isolate the connection with other device or set so that the input signal to the device is ignored.

Figure 16-5. Malfunction of Another Device



If the signal output by the μPD78F9831 affects another device in the flash memory programming mode, isolate the signal on the device side.



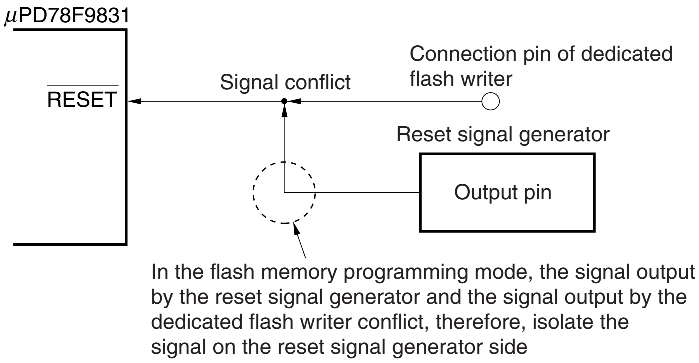
If the signal output by the dedicated flash programmer affects another device, isolate the signal on the device side.

<$\overline{\text{RESET}}$ pin>

When the reset signal of the dedicated flash programmer is connected to the $\overline{\text{RESET}}$ signal connected to the reset signal generator on the board, a signal conflict occurs. To prevent this signal conflict, isolate the connection with the reset signal generator.

If a reset signal is input from the user system in the flash memory programming mode, a normal programming operation will not be performed. Do not input signals other than reset signals from the dedicated flash programmer during this period.

Figure 16-6. Signal Conflict ($\overline{\text{RESET}}$ Pin)


μPD78F9831
$\overline{\text{RESET}}$
Signal conflict
Connection pin of dedicated flash writer
Reset signal generator
Output pin
In the flash memory programming mode, the signal output by the reset signal generator and the signal output by the dedicated flash writer conflict, therefore, isolate the signal on the reset signal generator side

<Port pins>

Shifting to the flash memory programming mode sets all the pins except those used for flash memory programming communication to the status immediately after reset.

Therefore, if the external device does not acknowledge an initial status such as the output high impedance status, connect the external device to V_{DD0} or V_{SS0} via a resistor.

<Oscillation pins>

When using an on-board clock, connection of X1 and X2 must conform to the methods in the normal operation mode.

When using the clock output of the flash programmer, directly connect it to the X1 pin with the on-board main resonator disconnected, and leave the X2 pin open. The subclock (XT1 and XT2) conforms to the method in the normal operation mode.

<Power supply>

To use the power output of the flash programmer, connect the V_{DD0} and V_{DD0} pins to VDD of the flash programmer, and the V_{SS0} and V_{SS1} pins to GND of the flash programmer.

To use the on-board power supply, connection must conform to that in the normal operation mode. However, because the voltage is monitored by the flash programmer, therefore, VDD of the flash programmer must be connected.

<Other pins>

Handle other pins (S0 to S31 and COM0 to COM15) in the same manner as is in the normal operation mode.

CHAPTER 17 INSTRUCTION SET

This chapter lists the instruction set of the μPD789830 Subseries. For the details of the operation and machine language (instruction code) of each instruction, refer to **78K/0S Series Instruction User's Manual (U11047E)**.

17.1 Operation

17.1.1 Operand identifiers and description methods

Operands are described in "Operand" column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for detail). When there are two or more description methods, select one of them. Alphabetic letters in capitals and symbols, #, !, $, and [ ] are key words and are described as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: Absolute address specification
- $: Relative address specification
- [ ]: Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, $ and [ ] symbols.

For operand register identifiers, r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for description.

Table 17-1. Operand Identifiers and Description Methods

Identifier	Description Method
r rp sfr	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special function register symbol
saddr saddrp	FE20H to FF1FH Immediate data or labels FE20H to FF1FH Immediate data or labels (even addresses only)
addr16 addr5	0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions) 0040H to 007FH Immediate data or labels (even addresses only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label

Remark See **Table 3-4** for symbols of special function registers.

17.1.2 Description of "Operation" column

A:	A register; 8-bit accumulator
X:	X register
B:	B register
C:	C register
D:	D register
E:	E register
H:	H register
L:	L register
AX:	AX register pair; 16-bit accumulator
BC:	BC register pair
DE:	DE register pair
HL:	HL register pair
PC:	Program counter
SP:	Stack pointer
PSW:	Program status word
CY:	Carry flag
AC:	Auxiliary carry flag
Z:	Zero flag
IE:	Interrupt request enable flag
NMIS:	Flag indicating non-maskable interrupt servicing in progress
( ):	Memory contents indicated by address or register contents in parentheses
×H, ×L:	Higher 8 bits and lower 8 bits of 16-bit register
∧:	Logical product (AND)
∨:	Logical sum (OR)
∀:	Exclusive logical sum (exclusive OR)
$\overline{\quad}$:	Inverted data
addr16:	16-bit immediate data or label
jdisp8:	Signed 8-bit data (displacement value)

17.1.3 Description of "flag operation" column

(Blank):	Unchanged
0:	Cleared to 0
1:	Set to 1
×	Set/cleared according to the result
R:	Previously saved value is stored

17.2 Operation List

Mnemonic	Operands	Byte	Clock	Operation	Flag		
					Z	AC	CY
MOV	r, #byte	3	6	r ← byte			
	saddr, #byte	3	6	(saddr) ← byte			
	sfr, #byte	3	6	sfr ← byte			
	A, r Note 1	2	4	A ← r			
	r, A Note 1	2	4	r ← A			
	A, saddr	2	4	A ← (saddr)			
	saddr, A	2	4	(saddr) ← A			
	A, sfr	2	4	A ← sfr			
	sfr, A	2	4	sfr ← A			
	A, !addr16	3	8	A ← (addr16)			
	!addr16, A	3	8	(addr16) ← A			
	PSW, #byte	3	6	PSW ← byte	×	×	×
	A, PSW	2	4	A ← PSW			
	PSW, A	2	4	PSW ← A	×	×	×
	A, [DE]	1	6	A ← (DE)			
	[DE], A	1	6	(DE) ← A			
	A, [HL]	1	6	A ← (HL)			
	[HL], A	1	6	(HL) ← A			
	A, [HL + byte]	2	6	A ← (HL + byte)			
	[HL + byte], A	2	6	(HL + byte) ← A			
XCH	A, X	1	4	A ↔ X			
	A, r Note 2	2	6	A ↔ r			
	A, saddr	2	6	A ↔ (saddr)			
	A, sfr	2	6	A ↔ sfr			
	A, [DE]	1	8	A ↔ (DE)			
	A, [HL]	1	8	A ↔ (HL)			
	A, [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	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) + byte	×	×	×
	A, r	2	4	A, CY ← A + r	×	×	×
	A, saddr	2	4	A, CY ← A + (saddr)	×	×	×
	A, !addr16	3	8	A, CY ← A + (addr16)	×	×	×
	A, [HL]	1	6	A, CY ← A + (HL)	×	×	×
	A, [HL + byte]	2	6	A, CY ← A + (HL + byte)	×	×	×
ADDC	A, #byte	2	4	A, CY ← A + byte + CY	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) + byte + CY	×	×	×
	A, r	2	4	A, CY ← A + r + CY	×	×	×
	A, saddr	2	4	A, CY ← A + (saddr) + CY	×	×	×
	A, !addr16	3	8	A, CY ← A + (addr16) + CY	×	×	×
	A, [HL]	1	6	A, CY ← A + (HL) + CY	×	×	×
	A, [HL + byte]	2	6	A, CY ← A + (HL + byte) + CY	×	×	×
SUB	A, #byte	2	4	A, CY ← A – byte	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) – byte	×	×	×
	A, r	2	4	A, CY ← A – r	×	×	×
	A, saddr	2	4	A, CY ← A – (saddr)	×	×	×
	A, !addr16	3	8	A, CY ← A – (addr16)	×	×	×
	A, [HL]	1	6	A, CY ← A – (HL)	×	×	×
	A, [HL + byte]	2	6	A, CY ← A – (HL + byte)	×	×	×

Note Only when rp = BC, DE, or HL.

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operands	Byte	Clock	Operation	Flag		
					Z	AC	CY
SUBC	A, #byte	2	4	A, CY ← A – byte – CY	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) – byte – CY	×	×	×
	A, r	2	4	A, CY ← A – r – CY	×	×	×
	A, saddr	2	4	A, CY ← A – (saddr) – CY	×	×	×
	A, !addr16	3	8	A, CY ← A – (addr16) – CY	×	×	×
	A, [HL]	1	6	A, CY ← A – (HL) – CY	×	×	×
	A, [HL + byte]	2	6	A, CY ← A – (HL + byte) – CY	×	×	×
AND	A, #byte	2	4	A ← A ∧ byte	×		
	saddr, #byte	3	6	(saddr) ← (saddr) ∧ byte	×		
	A, r	2	4	A ← A ∧ r	×		
	A, saddr	2	4	A ← A ∧ (saddr)	×		
	A, !addr16	3	8	A ← A ∧ (addr16)	×		
	A, [HL]	1	6	A ← A ∧ (HL)	×		
	A, [HL + byte]	2	6	A ← A ∧ (HL + byte)	×		
OR	A, #byte	2	4	A ← A ∨ byte	×		
	saddr, #byte	3	6	(saddr) ← (saddr) ∨ byte	×		
	A, r	2	4	A ← A ∨ r	×		
	A, saddr	2	4	A ← A ∨ (saddr)	×		
	A, !addr16	3	8	A ← A ∨ (addr16)	×		
	A, [HL]	1	6	A ← A ∨ (HL)	×		
	A, [HL + byte]	2	6	A ← A ∨ (HL + byte)	×		
XOR	A, #byte	2	4	A ← A ⊻ byte	×		
	saddr, #byte	3	6	(saddr) ← (saddr) ⊻ byte	×		
	A, r	2	4	A ← A ⊻ r	×		
	A, saddr	2	4	A ← A ⊻ (saddr)	×		
	A, !addr16	3	8	A ← A ⊻ (addr16)	×		
	A, [HL]	1	6	A ← A ⊻ (HL)	×		
	A, [HL + byte]	2	6	A ← A ⊻ (HL + byte)	×		

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operands	Byte	Clock	Operation	Flag		
					Z	AC	CY
CMP	A, #byte	2	4	A – byte	×	×	×
	saddr, #byte	3	6	(saddr) – byte	×	×	×
	A, r	2	4	A – r	×	×	×
	A, saddr	2	4	A – (saddr)	×	×	×
	A, !addr16	3	8	A – (addr16)	×	×	×
	A, [HL]	1	6	A – (HL)	×	×	×
	A, [HL + byte]	2	6	A – (HL + byte)	×	×	×
ADDW	AX, #word	3	6	AX, CY ← AX + word	×	×	×
SUBW	AX, #word	3	6	AX, CY ← AX – word	×	×	×
CMPW	AX, #word	3	6	AX – word	×	×	×
INC	r	2	4	r ← r + 1	×	×	
	saddr	2	4	(saddr) ← (saddr) + 1	×	×	
DEC	r	2	4	r ← r – 1	×	×	
	saddr	2	4	(saddr) ← (saddr) – 1	×	×	
INCW	rp	1	4	rp ← rp + 1			
DECW	rp	1	4	rp ← rp – 1			
ROR	A, 1	1	2	$(CY, A_7 \leftarrow A_0, A_{m-1} \leftarrow A_m) \times 1$			×
ROL	A, 1	1	2	$(CY, A_0 \leftarrow A_7, A_{m+1} \leftarrow A_m) \times 1$			×
RORC	A, 1	1	2	$(CY \leftarrow A_0, A_7 \leftarrow CY, A_{m-1} \leftarrow A_m) \times 1$			×
ROLC	A, 1	1	2	$(CY \leftarrow A_7, A_0 \leftarrow CY, A_{m+1} \leftarrow A_m) \times 1$			×
SET1	saddr.bit	3	6	(saddr.bit) ← 1			
	sfr.bit	3	6	sfr.bit ← 1			
	A.bit	2	4	A.bit ← 1			
	PSW.bit	3	6	PSW.bit ← 1	×	×	×
	[HL].bit	2	10	(HL).bit ← 1			
CLR1	saddr.bit	3	6	(saddr.bit) ← 0			
	sfr.bit	3	6	sfr.bit ← 0			
	A.bit	2	4	A.bit ← 0			
	PSW.bit	3	6	PSW.bit ← 0	×	×	×
	[HL].bit	2	10	(HL).bit ← 0			
SET1	CY	1	2	CY ← 1			1
CLR1	CY	1	2	CY ← 0			0
NOT1	CY	1	2	CY ← $\overline{CY}$			×

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

Mnemonic	Operands	Byte	Clock	Operation	Flag		
					Z	AC	CY
CALL	!addr16	3	6	(SP – 1) ← $(PC + 3)_H$, (SP – 2) ← $(PC + 3)_L$, PC ← addr16, SP ← SP – 2			
CALLT	[addr5]	1	8	(SP – 1) ← $(PC + 1)_H$, (SP – 2) ← $(PC + 1)_L$, PC_H ← (00000000, addr5 + 1), PC_L ← (00000000, addr5), SP ← SP – 2			
RET		1	6	PC_H ← (SP + 1), PC_L ← (SP), SP ← SP + 2			
RETI		1	8	PC_H ← (SP + 1), PC_L ← (SP), PSW ← (SP + 2), SP ← SP + 3, NMIS ← 0	R	R	R
PUSH	PSW	1	2	(SP – 1) ← PSW, SP ← SP – 1			
	rp	1	4	(SP – 1) ← rp_H, (SP – 2) ← rp_L, SP ← SP – 2			
POP	PSW	1	4	PSW ← (SP), SP ← SP + 1	R	R	R
	rp	1	6	rp_H ← (SP + 1), rp_L ← (SP), SP ← SP + 2			
MOVW	SP, AX	2	8	SP ← AX			
	AX, SP	2	6	AX ← SP			
BR	!addr16	3	6	PC ← addr16			
	$addr16	2	6	PC ← PC + 2 + jdisp8			
	AX	1	6	PC_H ← A, PC_L ← X			
BC	$saddr16	2	6	PC ← PC + 2 + jdisp8 if CY = 1			
BNC	$saddr16	2	6	PC ← PC + 2 + jdisp8 if CY = 0			
BZ	$saddr16	2	6	PC ← PC + 2 + jdisp8 if Z = 1			
BNZ	$saddr16	2	6	PC ← PC + 2 + jdisp8 if Z = 0			
BT	saddr.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if (saddr.bit) = 1			
	sfr.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
	A.bit, $addr16	3	8	PC ← PC + 3 + jdisp8 if A.bit = 1			
	PSW.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if PSW.bit = 1			
BF	saddr.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if (saddr.bit) = 0			
	sfr.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
	A.bit, $addr16	3	8	PC ← PC + 3 + jdisp8 if A.bit = 0			
	PSW.bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
DBNZ	B, $addr16	2	6	B ← B – 1, then PC ← PC + 2 + jdisp8 if B ≠ 0			
	C, $addr16	2	6	C ← C – 1, then PC ← PC + 2 + jdisp8 if C ≠ 0			
	saddr, $addr16	3	8	(saddr) ← (saddr) – 1, then PC ← PC + 3 + jdisp8 if (saddr) ≠ 0			
NOP		1	2	No Operation			
EI		3	6	IE ← 1 (Enable Interrupt)			
DI		3	6	IE ← 0 (Disable Interrupt)			
HALT		1	2	Set HALT Mode			
STOP		1	2	Set STOP Mode			

Remark One instruction clock cycle is one CPU clock cycle (f_{CPU}) selected by the processor clock control register (PCC).

17.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, INC, DEC, ROR, ROL, RORC, ROLC, PUSH, POP, DBNZ

2nd Operand 1st Operand	#byte	A	r	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL + byte]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV[Note] XCH[Note] ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											
[HL + byte]		MOV											

Note Except r = A.

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

2nd Operand 1st Operand	#word	AX	rp[Note]	saddrp	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	
rp	MOVW	MOVW[Note]				INCW DECW PUSH POP
saddrp		MOVW				
SP		MOVW				

Note Only when rp = BC, DE, or HL.

(3) Bit manipulation instructions

SET1, CLR1, NOT1, BT, BF

2nd Operand 1st Operand	$addr16	None
A.bit	BT BF	SET1 CLR1
sfr.bit	BT BF	SET1 CLR1
saddr.bit	BT BF	SET1 CLR1
PSW.bit	BT BF	SET1 CLR1
[HL].bit		SET1 CLR1
CY		SET1 CLR1 NOT1

(4) Call instructions/branch instructions

CALL, CALLT, BR, BC, BNC, BZ, BNZ, DBNZ

2nd Operand / 1st Operand	AX	!addr16	[addr5]	$addr16
Basic instructions	BR	CALL BR	CALLT	BR BC BNC BZ BNZ
Compound instructions				DBNZ

(5) Other instructions

RET, RETI, NOP, EI, DI, HALT, STOP

Absolute Maximum Ratings (T_A = 25°C) (μPD789830)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}		–0.3 to +6.5	V
Input voltage	V_{I1}	P00 to P07, P10, P11, P20 to P23, P25, P26, P30 to P34, P50 to P57, X1, X2, XT1, XT2, $\overline{\text{RESET}}$	–0.3 to V_{DD} + 0.3[Note]	V
	V_{I2}	P24 (N-ch open drain)	–0.3 to +13	V
Output voltage	V_O		–0.3 to V_{DD} + 0.3[Note]	V
Output current, high	I_{OH}	Per pin	–10	mA
		Total for all pins	–30	mA
Output current, low	I_{OL}	Per pin	30	mA
		Total for all pins	160	mA
Operating temperature	T_A		–20 to +60	°C
Storage temperature	T_{stg}		–65 to +150	°C

Note 6.5 V or lower

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Absolute Maximum Ratings (T_A = 25°C) (μPD78F9831)

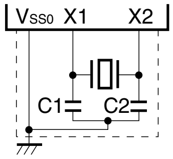
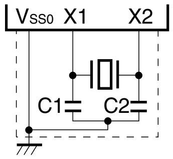
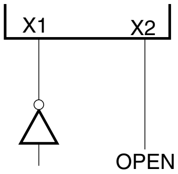
Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}		–0.3 to +6.5	V
	V_{PP}		–0.3 to +10.5	V
Input voltage	V_{I1}	P00 to P07, P10 to P17, P20 to P23, P25, P26, P30 to P34, P40, P41, P50 to P57, X1, X2, XT1, XT2, $\overline{\text{RESET}}$	–0.3 to V_{DD} + 0.3 Note	V
	V_{I2}	P24 (N-ch open drain)	–0.3 to +13	V
Output voltage	V_O		–0.3 to V_{DD} + 0.3 Note	V
Output current, high	I_{OH}	Per pin	–10	mA
		Total for all pins	–30	mA
Output current, low	I_{OL}	Per pin	30	mA
		Total for all pins	160	mA
Operating ambient temperature	T_A	During normal operation	–20 to +60	°C
		During flash memory programming	10 to 40	°C
Storage temperature	T_{stg}		–40 to +125	°C

Note 6.5 V or lower

Caution **Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.**

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Main System Clock Oscillator Characteristics (T_A = –20 to +60°C, V_{DD} = 2.7 to 5.5 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		Oscillator frequency (f_X)[Note 1]	V_{DD} oscillation voltage range	2.0		5.0	MHz
		Oscillation stabilization time[Note 2]	After V_{DD} reaches oscillation voltage range MIN.			4	ms
Crystal		Oscillator frequency (f_X)[Note 1]		2.0		5.0	MHz
		Oscillation stabilization time[Note 2]	V_{DD} = 4.5 to 5.5 V			10	ms
			V_{DD} = 2.7 to 5.5 V			30	ms
External clock		X1 input frequency (f_X)[Note 1]		2.0		5.0	MHz
		X1 input high-/low-level width (t_{XH}, t_{XL})		85		250	ns

Notes 1. Indicates only oscillator characteristics. Refer to **AC Characteristics** for the instruction execution time.

2. Time required to stabilize oscillation after reset or STOP mode release. Use the resonator that stabilizes oscillation within the oscillation wait time.

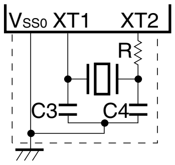
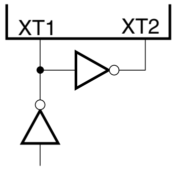
Cautions 1. When using the main system clock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with other signal lines.**
- **Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS0}.**
- **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, users are required to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

Subsystem Clock Oscillator Characteristics (T_A = –20 to +60°C, V_{DD} = 2.7 to 5.5 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		Oscillator frequency (f_{XT})[Note 1]		32	32.768	35	kHz
		Oscillation stabilization time[Note 2]	V_{DD} = 4.5 to 5.5 V		1.2	2	s
			V_{DD} = 2.7 to 5.5 V			10	s
External clock		XT1 input frequency (f_{XT})[Note 1]		32		35	kHz
		XT1 input high-/low-level width (t_{XTH}, t_{XTL})		14.3		15.6	μs

Notes 1. Indicates only oscillator characteristics. Refer to **AC Characteristics** for the instruction execution time.

2. Time required to stabilize oscillation after V_{DD} reaches oscillation voltage range MIN. Use the resonator that stabilizes oscillation within the oscillation wait time.

Cautions 1. When using the subsystem clock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with 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_{SS0}.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

2. The subsystem clock oscillator is designed as a low-amplitude circuit for reducing current consumption, and is more prone to malfunction due to noise than the main system clock oscillator. Particular care is therefore required with the wiring method when the subsystem clock is used.

Remark For the resonator selection and oscillator constant, users are required to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

DC Characteristics (T_A = –20 to +60°C, V_{DD} = 2.7 to 5.5 V) (μPD789830) (1/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high	I_{OH}	Per pin				–1	mA
		Total for all pins				–15	mA
Output current, low	I_{OL}	Per pin				10	mA
		Total for all pins				80	mA
Input voltage, high	V_{IH1}	P00 to P07, P10, P11, P23, P25, P50 to P57		0.7V_{DD}		V_{DD}	V
	V_{IH2}	$\overline{\text{RESET}}$, P20 to P22, P26, P30 to 34		0.8V_{DD}		V_{DD}	V
	V_{IH3}	P24 (N-ch open drain)		0.7V_{DD}		12	V
	V_{IH4}	X1, X2, XT1, XT2		V_{DD}–0.1		V_{DD}	V
Input voltage, high	V_{IL1}	P00 to P07, P10, P11, P23, P25, P50 to P57		0		0.3V_{DD}	V
	V_{IL2}	$\overline{\text{RESET}}$, P20 to P22, P26, P30 to P34		0		0.2V_{DD}	V
	V_{IL3}	P24 (N-ch open drain)		0		0.3V_{DD}	V
	V_{IL4}	X1, X2, XT1, XT2		0		0.1	V
Output voltage, high	V_{OH}	V_{DD} = 4.5 to 5.5 V, I_{OH} = –1 mA		V_{DD}–1.0			V
		I_{OH} = 2.7 to 5.5 V, I_{OH} = –100 μA		V_{DD}–0.5			V
Output voltage, low	V_{OL1}	Pins other than P24	V_{DD} = 4.5 to 5.5 V. I_{OL} = 10 mA			1.0	V
			V_{DD} = 2.7 to 5.5 V, I_{OL} = 400 μA			0.5	V
	V_{OL2}	P24 (N-ch open drain)	V_{DD} = 4.5 to 5.5 V, I_{OL} = 10 mA			1.0	V
			V_{DD} = 2.7 to 5.5 V, I_{OL} = 1.6 mA			0.4	V

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (T_A = –20 to +60 °C, V_{DD} = 2.7 to 5.5 V) (μPD789830) (2/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	$V_{IN} = V_{DD}$	P00 to P07, P10, P11, P20 to P23, P25, P26, P30 to P34, P50 to P57, $\overline{RESET}$			3	μA
	I_{LIH2}		X1, X2, XT1, XT2			20	μA
	I_{LIH3}	V_{IN} = 12 V	P24 (N-ch open drain)			20	μA
Input leakage current, low	I_{LIL1}	V_{IN} = 0 V	P00 to P07, P10, P11, P20 to P23, P25, P26, P30 to P34, P50 to P57, $\overline{RESET}$, P24 (N-ch open drain), except during read			–3	μA
	I_{LIL2}		X1, X2, XT1, XT2			–20	μA
	I_{LIL3}		P24 (N-ch open drain), during read			–30	μA
Output leakage current, high	I_{LOH}	$V_{OUT} = V_{DD}$				3	μA
Output leakage current, low	I_{LOL}	V_{OUT} = 0 V				–3	μA
Software pull-up resistor	R_1	V_{IN} = 0 V, P00 to P07, P10, P11, P30 to P34		50	100	200	kΩ
Power supply current[Note 1]	I_{DD1}	5.0 MHz crystal oscillation operating mode	V_{DD} = 5.0 V ±10%[Note 2]		1.7	3.2	mA
			V_{DD} = 3.0 V ±10%[Note 3]		0.45	0.9	mA
	I_{DD2}	5.0 MHz crystal oscillation HALT mode	V_{DD} = 5.0 V ±10%[Note 2]		0.6	1.2	mA
			V_{DD} = 3.0 V ±10%[Note 3]		0.3	0.6	mA
	I_{DD3}	32.768 kHz crystal oscillation operating mode[Note 4]	V_{DD} = 5.0 V ±10%		25	50	μA
			V_{DD} = 3.0 V ±10%		12	35	μA
	I_{DD4}	32.768 kHz crystal oscillation HALT mode[Note 4]	V_{DD} = 5.0 V ±10%		17	34	μA
			V_{DD} = 3.0 V ±10%		5	17	μA
	I_{DD5}	STOP mode	V_{DD} = 5.0 V ±10%		0.1	10	μA
			V_{DD} = 3.0 V ±10%		0.05	5.0	μA

Notes 1. Does not include the current when the LCD is operating (LCDON20 (bit 7 of LCD20 mode register (LCDM20)) = 1, LIPS20 (bit 4 of LCDM20) = 1) and the port current (including the current flowing through the on-chip pull-up resistors). For the current when the LCD is operating, refer to **LCD Operating Current** in **LCD Characteristics**.

2. High-speed mode operation (when processor clock control register (PCC) is set to 00H)

3. Low-speed mode operation (when PCC is set to 02H)

4. When the main system clock operation is stopped

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (T_A = –20 to +60°C, V_{DD} = 2.7 to 5.5 V) (μPD78F9831) (1/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high	I_{OH}	Per pin				–1	mA
		Total for all pins				–15	mA
Output current, low	I_{OL}	Per pin				10	mA
		Total for all pins				80	mA
Input voltage, high	V_{IH1}	P00 to P07, P10 to P17, P23, P25, P50 to P57		$0.7V_{DD}$		V_{DD}	V
	V_{IH2}	$\overline{\text{RESET}}$, P20 to P22, P26, P30 to P34, P40, P41		$0.8V_{DD}$		V_{DD}	V
	V_{IH3}	P24 (N-ch open drain)		$0.7V_{DD}$		12	V
	V_{IH4}	X1, X2, XT1, XT2	V_{DD} = 4.5 to 5.5 V	V_{DD} – 0.5		V_{DD}	V
			V_{DD} = 2.7 to 5.5 V	V_{DD} – 0.1		V_{DD}	V
Input voltage, low	V_{IL1}	P00 to P07, P10 to P17, P23, P25, P50 to P57		0		$0.3V_{DD}$	V
	V_{IL2}	$\overline{\text{RESET}}$, P20 to P22, P26, P30 to P34, P40, P41		0		$0.2V_{DD}$	V
	V_{IL3}	P24 (N-ch open drain)		0		$0.3V_{DD}$	V
	V_{IL4}	X1, X2, XT1, XT2	V_{DD} = 4.5 to 5.5 V	0		0.4	V
			V_{DD} = 2.7 to 5.5 V	0		0.1	V
Output voltage, high	V_{OH}	I_{OH} = –1 mA	V_{DD} = 4.5 to 5.5 V	V_{DD} – 1.0			V
		I_{OH} = –100 μA	V_{DD} = 2.7 to 5.5 V	V_{DD} – 0.5			V
Output voltage, low	V_{OL1}	Pins other than the P24 pin	$4.5 \leq V_{DD} \leq 5.5$ V, I_{OL} = 10 mA			1.0	V
			$2.7 \leq V_{DD} < 4.5$ V, I_{OL} = 400 μA			0.5	V
	V_{OL2}	P24 (N-ch open drain)	V_{DD} = 4.5 to 5.5 V, I_{OL} = 10 mA			1.0	V
			$2.7 \leq V_{DD} < 4.5$ V, I_{OL} = 1.6 mA			0.4	V
Input leakage current, high	I_{LIH1}	$V_{IN} = V_{DD}$	P00 to P07, P10 to P17, P20 to P23, P25, P26, P30 to P34, P40, P41, P50 to P57, $\overline{\text{RESET}}$			3	μA
	I_{LIH2}		X1, X2, XT1, XT2			20	μA
	I_{LIH3}	V_{IN} = 12 V	P24 (N-ch open drain)			20	μA
Input leakage current, low	I_{LIL1}	V_{IN} = 0 V	P00 to P07, P10 to P17, P20 to P23, P25, P26, P30 to P34, P40, P41, P50 to P57, $\overline{\text{RESET}}$, P24 (When an input instruction is not executed)			–3	μA
	I_{LIL2}		X1, X2, XT1, XT2			–20	μA
	I_{LIL3}		P24 (N-ch open drain) When an input instruction is executed			–30	μA

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (T_A = –20 to +60°C, V_{DD} = 2.7 to 5.5 V) (μPD78F9831) (2/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output leakage current, high	I_{LOH}	$V_{OUT} = V_{DD}$				3	μA
Output leakage current, low	I_{LOL}	V_{OUT} = 0 V				-3	μA
Software pull-up resistor	R_1	V_{IN} = 0 V, P00 to P07, P10 to P17, P30 to P34		50	100	200	kΩ
Power supply current[Note 1]	I_{DD1}	5.0 MHz crystal oscillation operating mode	V_{DD} = 5.0 V ±10%[Note 2]		5	10	MA
			V_{DD} = 3.0 V ±10%[Note 3]		3	6	mA
	I_{DD2}	5.0 MHz crystal oscillation HALT mode	V_{DD} = 5.0 V ±10%[Note 2]		0.8	1.6	mA
			V_{DD} = 3.0 V ±10%[Note 3]		0.4	0.8	mA
	I_{DD3}	32.768 kHz crystal oscillation operating mode[Note 4]	V_{DD} = 5.0 V ±10%		120	240	μA
			V_{DD} = 3.0 V ±10%		80	160	μA
	I_{DD4}	32.768 kHz crystal oscillation HALT mode[Note 4]	V_{DD} = 5.0 V ±10%		25	55	μA
			V_{DD} = 3.0 V ±10%		10	20	μA
	I_{DD5}	STOP mode	V_{DD} = 5.0 V ±10%		0.1	10	μA
			V_{DD} = 3.0 V ±10%		0.05	10	μA

Notes 1. Does not include the current when the LCD is operating (LCDON20 = 1, LIPS20 = 1) and the port current (including the current flowing through the on-chip pull-up resistors).

2. High-speed mode operation (when processor clock control register (PCC) is set to 00H)

3. Low-speed mode operation (when PCC is set to 02H)

4. When the main system clock operation is stopped

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

LCD Characteristics (T_A = –20 to +60°C, V_{DD} = 2.7 to 5.5 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
LCD drive voltage	V_{LCD}	$V_{DD} = V_{LCD}$	VAON20 = 0	3.5		5.5	V
			VAON20 = 1	2.7		5.5	V
Segment output voltage[Note 1]	V_{ODS}	When the output level is V_{LC0}			V_{LCD}		V
		When the output level is V_{LC2}			$3/5V_{LCD}$		V
		When the output level is V_{LC3}			$2/5V_{LCD}$		V
Common output voltage[Note 1]	V_{ODC}	When the output level is V_{LC0}			V_{LCD}		V
		When the output level is V_{LC1}			$4/5V_{LCD}$		V
		When the output level is V_{LC4}			$1/5V_{LCD}$		V
Segment output on resistance	R_{SEG}	$V_{LCn} \rightarrow$ Sp, I_O = | 20 μA |			5.0	12.5	kΩ
Common output on resistance	R_{COM}	$V_{LCn} \rightarrow$ COMq, I_O = | 20 μA |			4.0	10.0	kΩ
LCD input frequency	f_{LCD}	VAON20 = 1		32		78.13	kHz
		VAON20 = 0		7.81		78.13	kHz
LCD operating current[Note 2] (μPD789830)	I_{LCD1}	V_{DD} = 5.0 V ±10%, VAON20 = 0			25	50	μA
	I_{LCD2}	V_{DD} = 3.0 V ±10%, VAON20 = 1			13	30	μA
LCD operating current[Note 2] (μPD78F9831)	I_{LCD1}	V_{DD} = 5.0 V ±10%, VAON20 = 0			30	65	μA
	I_{LCD2}	V_{DD} = 3.0 V ±10%, VAON20 = 1			17	40	μA

Notes 1. Voltages when no load is applied

2. Total current flowing through the V_{DD0} pin (including the current flowing through the LCD divider resistor)

When LCDON20 = 0 and LIPS20 = 0 (the display is turned off and the internal drive power is not supplied), the power supply current is included in the power supply current I_{DD5} (STOP mode) in the **DC Characteristics**.

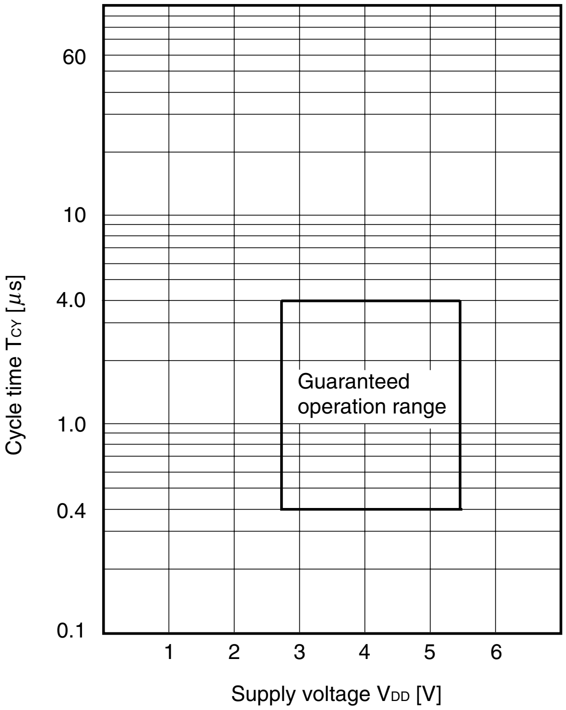
Remark n = 0 to 4
p = 0 to 39
q = 0 to 15

AC Characteristics

(1) Basic operation (T_A = –20 to +60 °C, V_{DD} = 2.7 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Cycle time (Min. instruction execution time)	T_{CY}	Operating with main system clock	0.4		4.0	μs
		Operating with subsystem clock	114	122	125	μs
Interrupt input high-/low-level width	t_{INTH}, t_{INTL}	INTP0 to INTP2 (μPD789830)	10			μs
		INTP0 to INTP4 (μPD78F9831)	10			μs
$\overline{\text{RESET}}$ input low-level width	t_{RSL}		10			μs

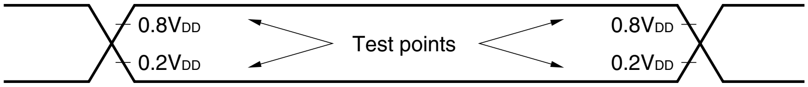
T_{CY} vs V_{DD} (main system clock)



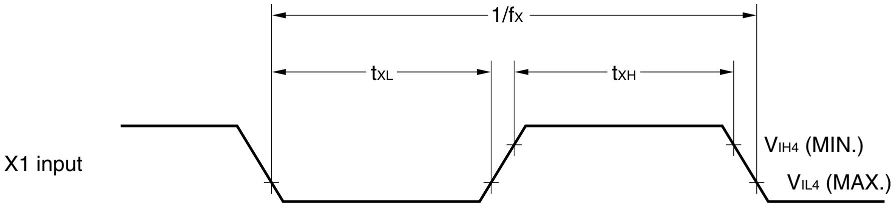
(2) Serial interface (UART00) (T_A = –20 to +60 °C, V_{DD} = 2.7 to 5.5 V)

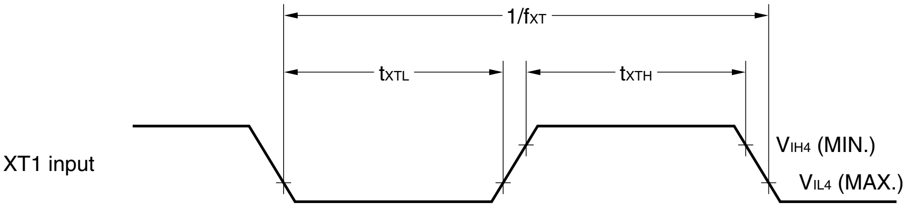
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		Operation at f_X = 5.0 MHz			78,125	bps

AC Timing Test Points (excluding X1, XT1 inputs)

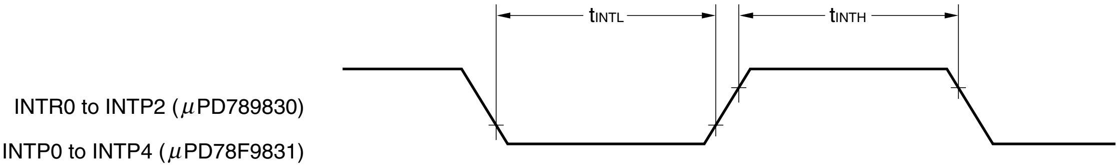


Clock Timing

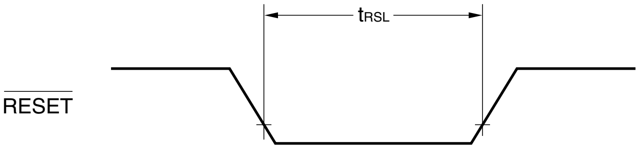




Interrupt Input Timing



$\overline{\text{RESET}}$ Input Timing



Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics (T_A = –20 to +60°C)

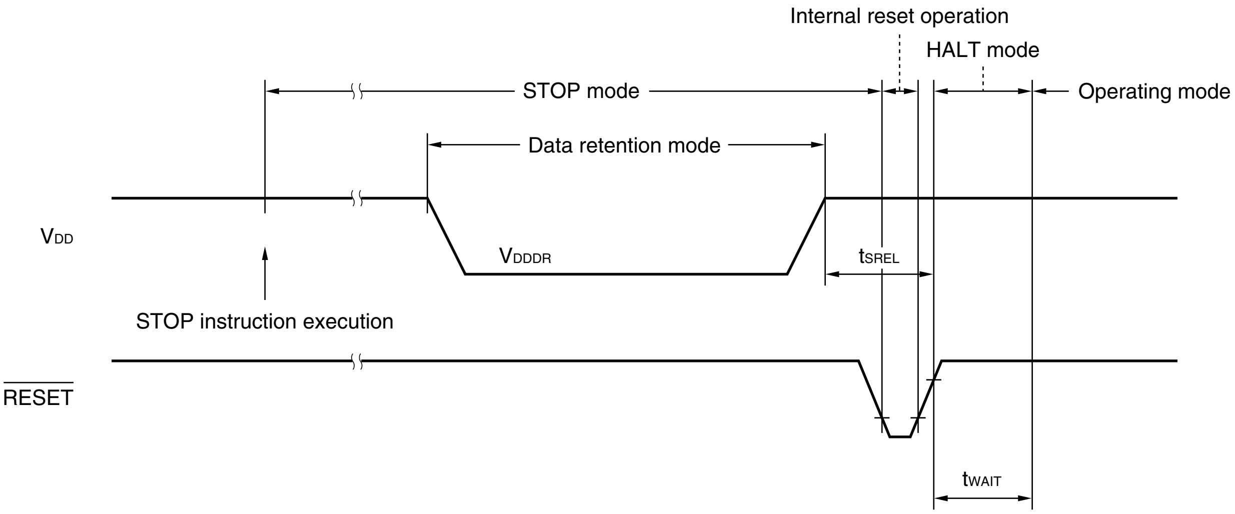
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention power supply voltage	V_{DDDR}		1.8		5.5	V
Release signal set time	t_{SREL}		0			μs
Oscillation stabilization wait time[Note 1]	t_{WAIT}	Release by $\overline{\text{RESET}}$		2^{15}/fx		ms
		Release by interrupt request		**Note 2**		ms

Notes 1. Oscillation stabilization wait time is a time for stopping the CPU operation to prevent the unstable operation when the oscillation is started.

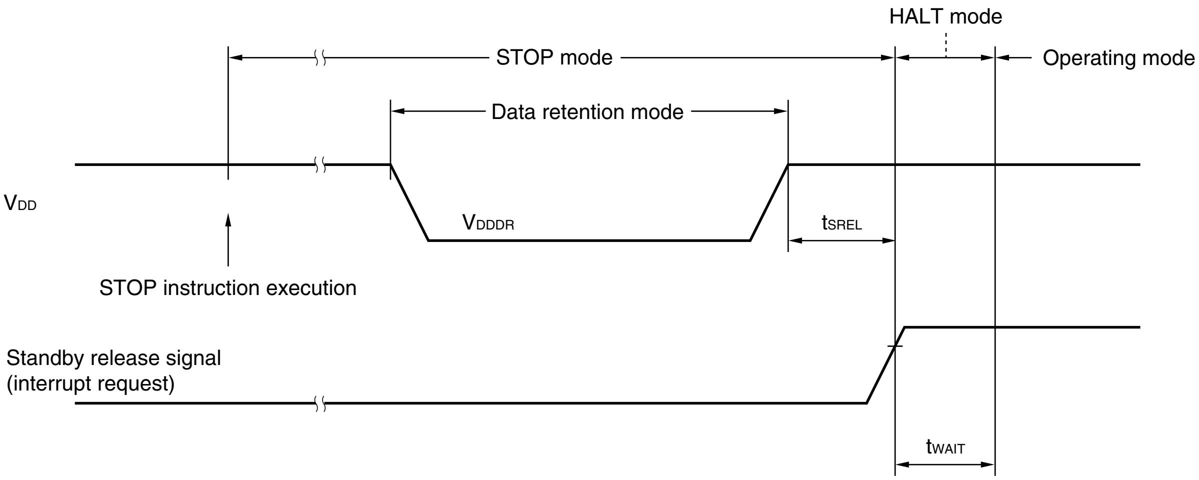
2. Selection of 2^{12}/fx, 2^{15}/fx, and 2^{17}/fx is possible with bits 0 to 2 (OSTS0 to OSTS2) of the oscillation stabilization time select register (OSTS).

Remark fx: Main system clock oscillation frequency

Data Retention Timing (STOP mode release by $\overline{\text{RESET}}$)



Data Retention Timing (Standby release signal: STOP mode release by interrupt signal)

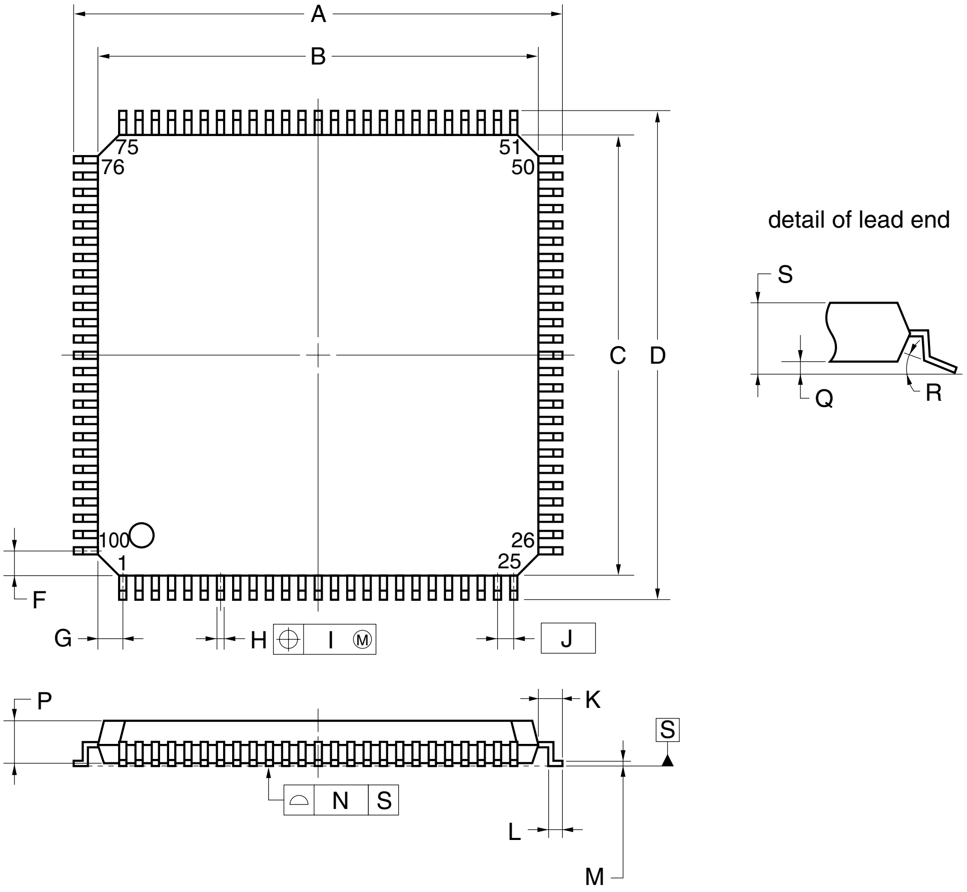


Flash Memory Write/Erase Characteristics (T_A = 10 to 40 °C, V_{DD} = 2.7 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Operating frequency	f_X		2		5	MHz
Write current[Note] (V_{DD} pin)	I_{DDW}	When V_{PP} supply voltage = V_{PP1} 5.0 MHz crystal oscillation operating mode			13	mA
Write current[Note] (V_{PP} pin)	I_{PPW}	When V_{PP} supply voltage = V_{PP1}			20	mA
Erase current[Note] (V_{DD} pin)	I_{DDE}	When V_{PP} supply voltage = V_{PP1} 5.0 MHz crystal oscillation operating mode			13	mA
Erase current[Note] (V_{PP} pin)	I_{PPE}	When V_{PP} supply voltage = V_{PP1}			100	mA
Erase time	t_{er}				20	s
Write count		Erase/write are regarded as 1 cycle.			20	Times
V_{PP} supply voltage	V_{PP0}	In normal operation	0		$0.2V_{DD}$	V
	V_{PP1}	During flash memory programming	9.7	10.0	10.3	V

Note The port current (including the current that flows to the on-chip pull-up resistors) is not included.

100-PIN PLASTIC LQFP (FINE PITCH) (14x14)



NOTE

Each lead centerline is located within 0.08 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	16.00±0.20
B	14.00±0.20
C	14.00±0.20
D	16.00±0.20
F	1.00
G	1.00
H	$0.22^{+0.05}_{-0.04}$
I	0.08
J	0.50 (T.P.)
K	1.00±0.20
L	0.50±0.20
M	$0.17^{+0.03}_{-0.07}$
N	0.08
P	1.40±0.05
Q	0.10±0.05
R	$3°^{+7°}_{-3°}$
S	1.60 MAX.

S100GC-50-8EU, 8EA-2

★

CHAPTER 20 RECOMMENDED SOLDERING CONDITIONS

The μPD78F9831 should be soldered and mounted under the following recommended conditions.

For soldering methods and conditions other than those recommended below, contact an NEC sales representative.

For technical information, see the following website.

Semiconductor Device Mount Manual (http://www.necel.com/pkg/en/mount/index.html)

Table 20-1. Surface Mounting Soldering Conditions

(1) μPD78F9831GC-8EU

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 235°C, Time: 30 seconds max. (at 210°C or higher), Count: Twice or less	IR35-00-2
VPS	Package peak temperature: 215°C, Time: 40 seconds max. (at 200°C or higher), Count: Twice or less	VP15-00-2
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

(2) μPD78F9831GC-8EU-A

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 260°C, Time: 60 seconds max. (at 220°C or higher), Count: Three times or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 20 to 72 hours)	IR60-207-3
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

Note After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Caution Do not use different soldering methods together (except for partial heating).

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

★

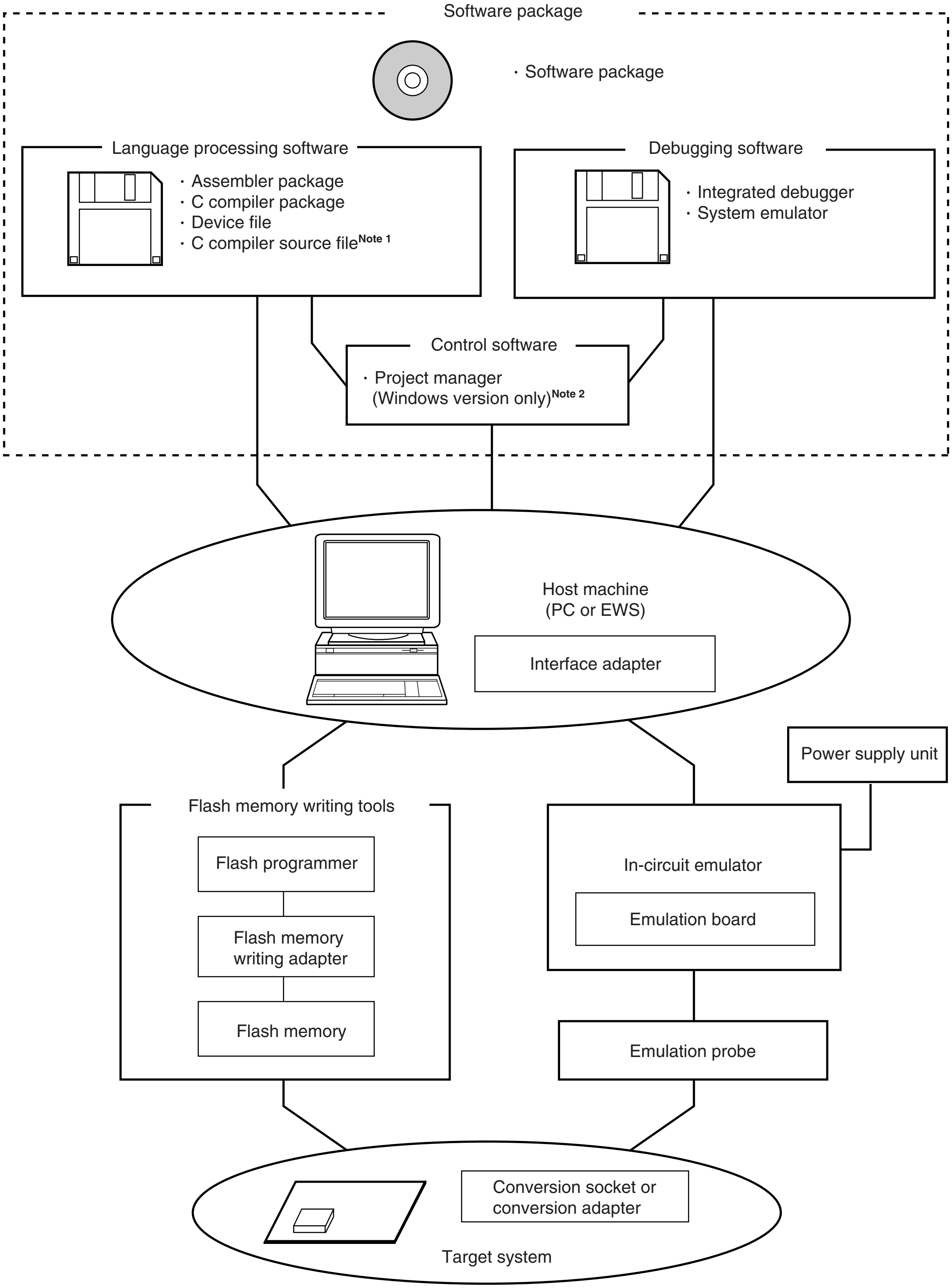
APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for development of systems using the μPD789830 Subseries. Figure A-1 shows development tools.

- Compatibility with PC98-NX Series
  Unless stated otherwise, products which are supported for IBM PC/AT™ and compatibles can also be used with the PC98-NX Series. When using the PC98-NX Series, therefore, refer to the explanations for IBM PC/AT and compatibles.

- Windows
  Unless stated otherwise, "Windows" refers to 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 and is available only for Windows.

A.1 Software Package

SP78K0S Software package	Various software tools for 78K/0S development are integrated in 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 and compatibles	Japanese Windows	CD-ROM
BB17		English Windows	

Note Also operates under the DOS environment

A.2 Language Processing Software

RA78K0S Assembler package	Program that converts program written in mnemonic into object code that can be executed by microcontroller. In addition, automatic functions to generate symbol table and optimize branch instructions are also provided. Used in combination with a device file (DF789831) (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 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 (DF789831) (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
DF789831[Note 1] Device file	File containing the information inherent to the device. Used in combination with other tools (RA78K0S, CC78K0S, ID78K0S-NS, SM78K0S) (all sold separately).
	Part number: μS××××DF789831
CC78K0S-L[Note 2] C compiler source file	Source file of functions constituting 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 the source file, its working environment does not depend on any particular operating system.
	Part number: μS××××CC78K0S-L

Notes 1. DF789831 is a common file that can be used with RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S.

2. CC78K0S-L is not included in the software package (SP78K0S).

Remark ×××× in the part number differs depending on the host machine and operating system to be used.

*μ*S××××RA78K0S
*μ*S××××CC78K0S

××××	Host Machine	OS	Supply Media
AB13	PC-9800 series, IBM PC/AT and 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)	

*μ*S××××DF789831
*μ*S××××CC78K0S-L

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT and 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 provided for an efficient user program development in the Windows environment. The Project Manager allows a series of tasks required for user program development to be performed, including starting the editor, building, and starting the debugger. <**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 number: FL-PR3, PG-FP3) Flash writer	Flash programmer dedicated to microcontrollers incorporating flash memory.
FA-100GC Flash memory writing adapter	Flash memory writing adapter. Used in connection with Flashpro III. 100-pin plastic LQFP

Remark FL-PR3 and FA-100GC are products of Naito Densei Machida Mfg. Co., Ltd.
For further information, contact: Naito Densei Machida Mfg. Co., Ltd. (+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 the 78K/0S Series. Supports an integrated debugger (ID78K0S-NS). Used in combination with an AC adapter, emulation probe, and interface adapter for connecting the host machine.
IE-78K0S-NS-A In-circuit emulator		In-circuit emulator with enhanced functions of the IE-78K0S-NS. The debug function is further enhanced by adding a coverage function and enhancing the tracer and timer functions.
IE-70000-MC-PS-B AC adapter		Adapter for supplying power from a 100 to 240 VAC outlet.
IE-70000-98-IF-C Interface adapter		Adapter required when using a PC-9800 series (except notebook type) as the host machine (C bus supported).
IE-70000-CD-IF-A PC card interface		PC card and interface cable required when using a notebook type PC as the host machine (PCMICA socket supported).
IE-70000-PC-IF-C Interface adapter		Adapter required when using an IBM PC/AT or compatible as the host machine (ISA bus supported).
IE-70000-PCI-IF-A Interface adapter		Adapter required when using a personal computer incorporating the PCI bus as the host machine.
IE-789831-NS-EM1 Emulation board		Emulation board for emulating the peripheral hardware inherent to the device. Used in combination with an in-circuit emulator.
NP-100GC Emulation probe		Cable for connecting the in-circuit emulator and target system. Used in combination with the TGC-100SDW when supporting a 100-pin plastic LQFP.
	TGC-100SDW Conversion adapter	Conversion adapter used to connect a target system board designed to allow mounting a 100-pin plastic LQFP and the NP-100GC.

Remarks 1. NP-100GC is a product made by Naito Densei Machida Mfg. Co., Ltd.
For further information, contact: Naito Densei Machida Mfg. Co., Ltd. (+81-45-475-4191)

2. The TGC-100SDW is a product made by TOKYO ELETECH CORPORATION.
For further information, contact: Daimaru Kogyo, Ltd.
Tokyo Electronics Department (TEL +81-3-3820-7112)
Osaka Electronics Department (TEL +81-6-6244-6672)

A.6 Debugging Tools (Software)

ID78K0S-NS Integrated debugger	This debugger supports the in-circuit emulators IE-78K0S-NS and IE-78K0S-NS-A for the 78K/0S Series. The ID78K0S-NS is Windows-based software. It has improved C-compatible debugging functions and can display the results of tracing with the source program using an integrating window function that associates the source program, disassemble display, and memory display with the trace result. Used in combination with a device file (DF789831) (sold separately).
	Part number: μS××××ID78K0S-NS
SM78K0S System simulator	This is a system simulator for the 78K/0S Series. The SM78K0S is Windows-based software. It can be used to debug the target system at C source level of assembler level while simulating the operation of the target system on the host machine. Using SM78K0S, the logic and performance of the application can be verified independently of hardware development. Therefore, the development efficiency can be enhanced and the software quality can be improved. Used in combination with a device file (DF789831) (sold separately).
	Part number: μS××××SM78K0S
DF789831[Note] Device file	File containing the information inherent to the device. Used in combination with other tools (RA78K0S, CC78K0S, ID78K0S-NS, SM78K0S) (all sold separately).
	Part number: μS××××DF789831

Note DF789831 is a common file that can be used with RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S.

Remark ×××× in the part number differs depending on the operating system to be used and the supply medium.

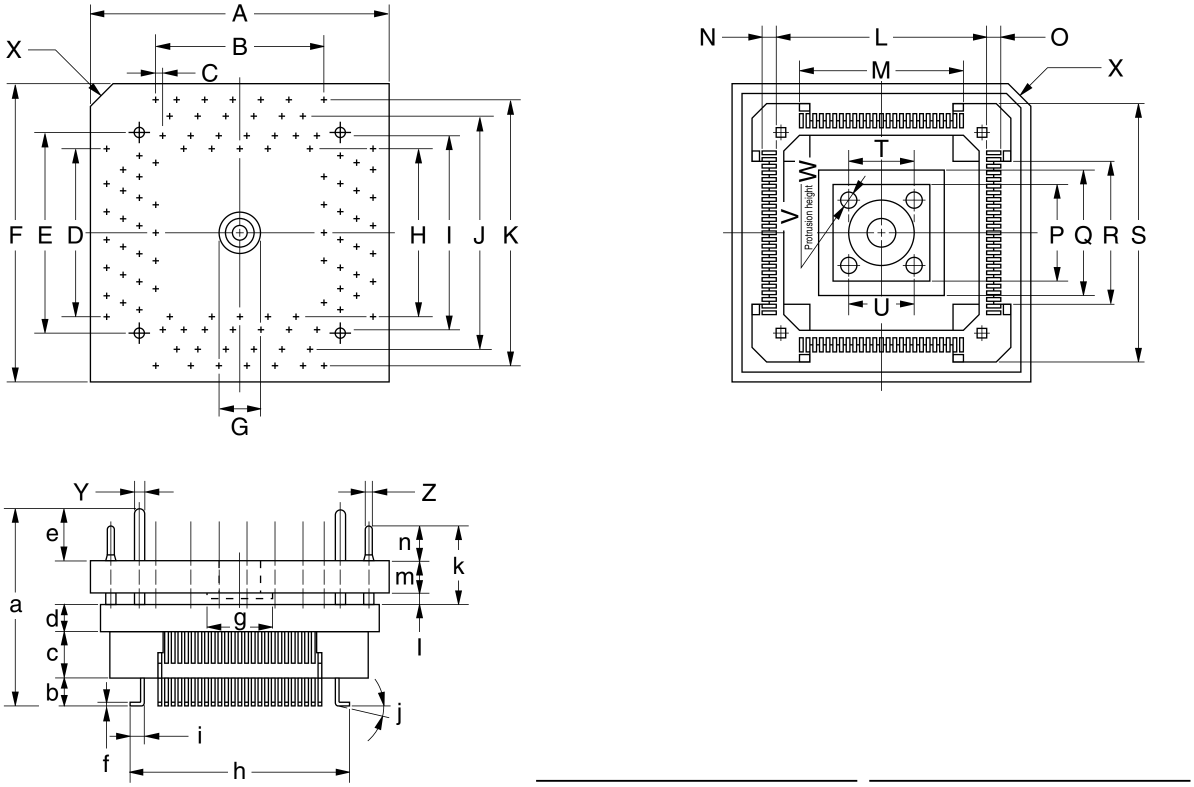
μS××××ID78K0S-NS
μS××××SM78K0S

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT and compatibles	Japanese Windows	3.5" 2HD FD
BB13		English Windows	
AB17		Japanese Windows	CD-ROM
BB17		English Windows	

A.7 Package Drawings of Conversion Adapter (TGC-100SDW)

Figure A-2. TGC-100SDW Package Drawings (for Reference) (Unit: mm)

TGC-100SDW (TQPACK100SD + TQSOCKET100SDW)
Package dimension (unit: mm)



ITEM	MILLIMETERS	INCHES
A	21.55	0.848
B	0.5x24=12	0.020x0.945=0.472
C	0.5	0.020
D	0.5x24=12	0.020x0.945=0.472
E	15.0	0.591
F	21.55	0.848
G	ϕ3.55	ϕ0.140
H	10.9	0.429
I	13.3	0.524
J	15.7	0.618
K	18.1	0.713
L	13.75	0.541
M	0.5x24=12.0	0.020x0.945=0.472
N	1.125±0.3	0.044±0.012
O	1.125±0.2	0.044±0.008
P	7.5	0.295
Q	10.0	0.394
R	11.3	0.445
S	18.1	0.713
T	ϕ5.0	ϕ0.197
U	5.0	0.197
V	4-ϕ1.3	4-ϕ0.051
W	1.8	0.071
X	C 2.0	C 0.079
Y	ϕ0.9	ϕ0.035
Z	ϕ0.3	ϕ0.012

ITEM	MILLIMETERS	INCHES
a	14.45	0.569
b	1.85±0.25	0.073±0.010
c	3.5	0.138
d	2.0	0.079
e	3.9	0.154
f	0.25	0.010
g	ϕ4.5	ϕ0.177
h	16.0	0.630
i	1.125±0.3	0.044±0.012
j	0~5°	0.000~0.197°
k	5.9	0.232
l	0.8	0.031
m	2.4	0.094
n	2.7	0.106

TGC-100SDW-G1E

note: Product by TOKYO ELETECH CORPORATION.

APPENDIX B REGISTER INDEX

B.1 Register Name Index (Alphabetic Order)

B.2 Register Symbol Index (Alphabetic Order)

APPENDIX C REVISION HISTORY

The following shows the revision history. “Chapter” refers to the chapters in the respective edition.

Edition	Description	Chapter
2nd edition	Addition of table of timer outline	**CHAPTER 1 GENERAL**
	Addition of description on V_{PP} pin processing	**CHAPTER 2 PIN FUNCTIONS**
	Modification of tables for types of pin I/O circuits and recommended connection of unused pins	
	Addition of **Note** to description on suboscillation mode register (SCKM)	**CHAPTER 5 CLOCK GENERATOR**
	Addition of **Caution** to description on 16-bit compare register 40 (CR40)	**CHAPTER 6 16-BIT TIMER 40**
	Addition of **Caution** to description on operation as interval timer	**CHAPTER 7 8-BIT TIMER 00**
	Modification of **Caution** in watchdog timer mode register (WDTM)	**CHAPTER 9 WATCHDOG TIMER**
	Overall modification of descriptions on μPD78F9831	**CHAPTER 16 μPD78F9831**
	Addition of electrical specifications	**CHAPTER 18 ELECTRICAL SPECIFICATIONS**
	Addition of package drawing	**CHAPTER 19 PACKAGE DRAWINGS**
	Addition of recommended soldering conditions	**CHAPTER 20 RECOMMENDED SOLDERING CONDITIONS**
	Overall modification of descriptions on development tools	**APPENDIX A DEVELOPMENT TOOLS**
	Addition of revision history	**APPENDIX C REVISION HISTORY**
	Deletion of **APPENDIX B EMBEDDED SOFTWARE**	–
2nd edition (Modification version)	Addition of lead-free products to ordering information	**CHAPTER 1 GENERAL**
	Addition of lead-free products to recommended soldering conditions	**CHAPTER 20 RECOMMENDED SOLDERING CONDITIONS**