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

78K0/LF3

8-Bit Single-Chip Microcontrollers

μPD78F0471	μPD78F0481	μPD78F0491
μPD78F0472	μPD78F0482	μPD78F0492
μPD78F0473	μPD78F0483	μPD78F0493
μPD78F0474	μPD78F0484	μPD78F0494
μPD78F0475	μPD78F0485	μPD78F0495

Document No. U18329EJ4V0UD00 (4th edition)
Date Published August 2008 NS

© NEC Electronics Corporation 2007
Printed in Japan

[MEMO]

NOTES FOR CMOS DEVICES

① **VOLTAGE APPLICATION WAVEFORM AT INPUT PIN**

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

② **HANDLING OF UNUSED INPUT PINS**

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

③ **PRECAUTION AGAINST ESD**

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

④ **STATUS BEFORE INITIALIZATION**

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

⑤ **POWER ON/OFF SEQUENCE**

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

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

⑥ **INPUT OF SIGNAL DURING POWER OFF STATE**

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

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

EEPROM is a trademark of NEC Electronics Corporation.

SuperFlash is a registered trademark of Silicon Storage Technology, Inc. in several countries including the United States and Japan.

Caution: This product uses SuperFlash® technology licensed from Silicon Storage Technology, Inc.

- **The information in this document is current as of June, 2008. 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

INTRODUCTION

Readers

This manual is intended for user engineers who wish to understand the functions of the 78K0/LF3 and design and develop application systems and programs for these devices. The target products are as follows.

78K0/LF3: μPD78F0471, 78F0472, 78F0473, 78F0474, 78F0475,
μPD78F0481, 78F0482, 78F0483, 78F0484, 78F0485,
μPD78F0491, 78F0492, 78F0493, 78F0494, 78F0495

Purpose

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

Organization

The 78K0/LF3 manual is separated into two parts: this manual and the instructions edition (common to the 78K0 microcontrollers).

78K0/LF3 User's Manual (This Manual)	**78K/0 Series User's Manual Instructions**
• Pin functions • Internal block functions • Interrupts • Other on-chip peripheral functions • Electrical specifications	• CPU functions • Instruction set • Explanation of each instruction

How to Read This Manual

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

- To gain a general understanding of functions:
  → Read this manual in the order of the **CONTENTS**. The mark "<R>" shows major revised points. The revised points can be easily searched by copying an "<R>" in the PDF file and specifying it in the "Find what:" field.
- How to interpret the register format:
  → For a bit number enclosed in angle brackets, the bit name is defined as a reserved word in the RA78K0, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0.
- To know details of the 78K0 microcontroller instructions:
  → Refer to the separate document **78K/0 Series Instructions User's Manual (U12326E)**.

Conventions	Data significance:	Higher digits on the left and lower digits on the right
	Active low representations:	$\overline{\times\times\times}$ (overscore over pin and signal name)
	Note:	Footnote for item marked with **Note** in the text
	Caution:	Information requiring particular attention
	Remark:	Supplementary information
	Numerical representations:	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.
78K0/LF3 User's Manual	This manual
78K/0 Series Instructions User's Manual	U12326E
78K0 Microcontrollers Self Programming Library Type01 User's Manual[Note]	U18274E
78K0 Microcontrollers EEPROM™ Emulation Library Type01 User's Manual[Note]	U18275E

Note This document is under engineering management. For details, consult an NEC Electronics sales representative.

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

Document Name		Document No.
RA78K0 Ver. 3.80 Assembler Package	Operation	U17199E
	Language	U17198E
	Structured Assembly Language	U17197E
CC78K0 Ver. 3.70 C Compiler	Operation	U17201E
	Language	U17200E
ID78K0-QB Ver. 3.00 Integrated Debugger	Operation	U18492E
PM+ Ver. 6.30		U18416E

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

Document Name	Document No.
QB-78K0LX3 In-Circuit Emulator	U18511E
QB-MINI2 On-Chip Debug Emulator with Programming Function	U18371E

Documents Related to Flash Memory Programming

Document Name	Document No.
PG-FP5 Flash Memory Programme	U18865E

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

Other Documents

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

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

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

CONTENTS

1.1 Features

- Minimum instruction execution time can be changed from high speed (0.2 μs: @ 10 MHz operation with high-speed system clock) to ultra low-speed (122 μs: @ 32.768 kHz operation with subsystem clock)
- General-purpose register: 8 bits × 32 registers (8 bits × 8 registers × 4 banks)
- ROM, RAM capacities

Part Number \ Item	Program Memory (ROM)		Data Memory		
			Internal High-Speed RAM[Note 1]	Internal Expansion RAM[Note 1]	LCD Display RAM
μPD78F0471, 78F0481, 78F0491	Flash memory[Note 1]	16 KB	768 bytes	–	<μPD78F047x, 78F048x>
μPD78F0472, 78F0482, 78F0492		24 KB	1 KB		40 × 4 bits (36 × 8 bits)
μPD78F0473, 78F0483, 78F0493		32 KB			[36 × 4 bits (32 × 8 bits)][Note 2]
μPD78F0474, 78F0484, 78F0494		48 KB		1 KB	<μPD78F049x>
μPD78F0475, 78F0485, 78F0495		60 KB			32 × 4 bits (28 × 8 bits) [28 × 4 bits (24 × 8 bits)][Note 2]

Notes 1. The internal flash memory, internal high-speed RAM capacities, and internal expansion RAM capacities can be changed using the internal memory size switching register (IMS) and the internal expansion RAM size switching register (IXS).

<R> **2.** The items in parentheses are applicable when 8com is used.
The items in square brackets are applicable when using the UART6 pins (RxD6, TxD6) on the bottom side.

- On-chip single-power-supply flash memory
- Self-programming (with boot swap function)
- On-chip debug function
- On-chip power-on-clear (POC) circuit and low-voltage detector (LVI)
- On-chip watchdog timer (operable with internal low-speed oscillation clock)
- LCD controller/driver (external resistance division and internal resistance division are switchable)

μPD78F047x: Segment signals: 36, Common signals: 8 (1/4 bias)
: Segment signals: 40, Common signals: 4 (1/3 bias)
: Segment signals: 40, Common signals: 3 (1/3, 1/2 bias)
: Segment signals: 40, Common signals: 2 (1/2 bias)
: Segment signals: 40, Common signals: 1 (Static)

μPD78F048x: Segment signals: 36, Common signals: 8 (1/4 bias)
: Segment signals: 40, Common signals: 4 (1/3 bias)
: Segment signals: 40, Common signals: 3 (1/3, 1/2 bias)
: Segment signals: 40, Common signals: 2 (1/2 bias)
: Segment signals: 40, Common signals: 1 (Static)

μPD78F049x: Segment signals: 28, Common signals: 8 (1/4 bias)
: Segment signals: 32, Common signals: 4 (1/3 bias)
: Segment signals: 32, Common signals: 3 (1/3, 1/2 bias)
: Segment signals: 32, Common signals: 2 (1/2 bias)
: Segment signals: 32, Common signals: 1 (Static)

<R> ○ On-chip segment key scan function: 8 channels

○ On-chip key interrupt function: 8 channels

○ On-chip clock output/buzzer output controller

○ I/O ports: 62

○ Timer: 9 channels

- 16-bit timer/event counter: 1 channel
- 8-bit timer/event counter: 3 channels
- 8-bit timer: 3 channels
- Real-time counter (RTC): 1 channel
- Watchdog timer: 1 channel

○ Serial interface: 3 channels

- UART (LIN (Local Interconnect Network)-bus supported): 1 channel
- CSI/UART[Note]: 1 channel
- CSI with automatic transmit/receive function: 1 channel

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

○ 16-bit $\Delta\Sigma$ type A/D converter[Note]: 3 channels (μPD78F049x only)

○ 10-bit successive approximation type A/D converter: 8 channels (μPD78F048x and 78F049x only)

○ Remote controller receiver

○ Manchester code generator

○ Power supply voltage: V_{DD} = 1.8 to 5.5 V

○ Operating ambient temperature: T_A = −40 to +85°C

Note The specifications of the 16-bit $\Delta\Sigma$ A/D converter may have been changed.
For details of the specifications, contact an NEC Electronics sales representative or authorized dealer.

1.2 Applications

Digital cameras, AV equipments, household electrical appliances, utility meters, health care equipments, and measurement equipment, etc.

1.3 Ordering Information

- **Flash memory version (Lead-free products)**

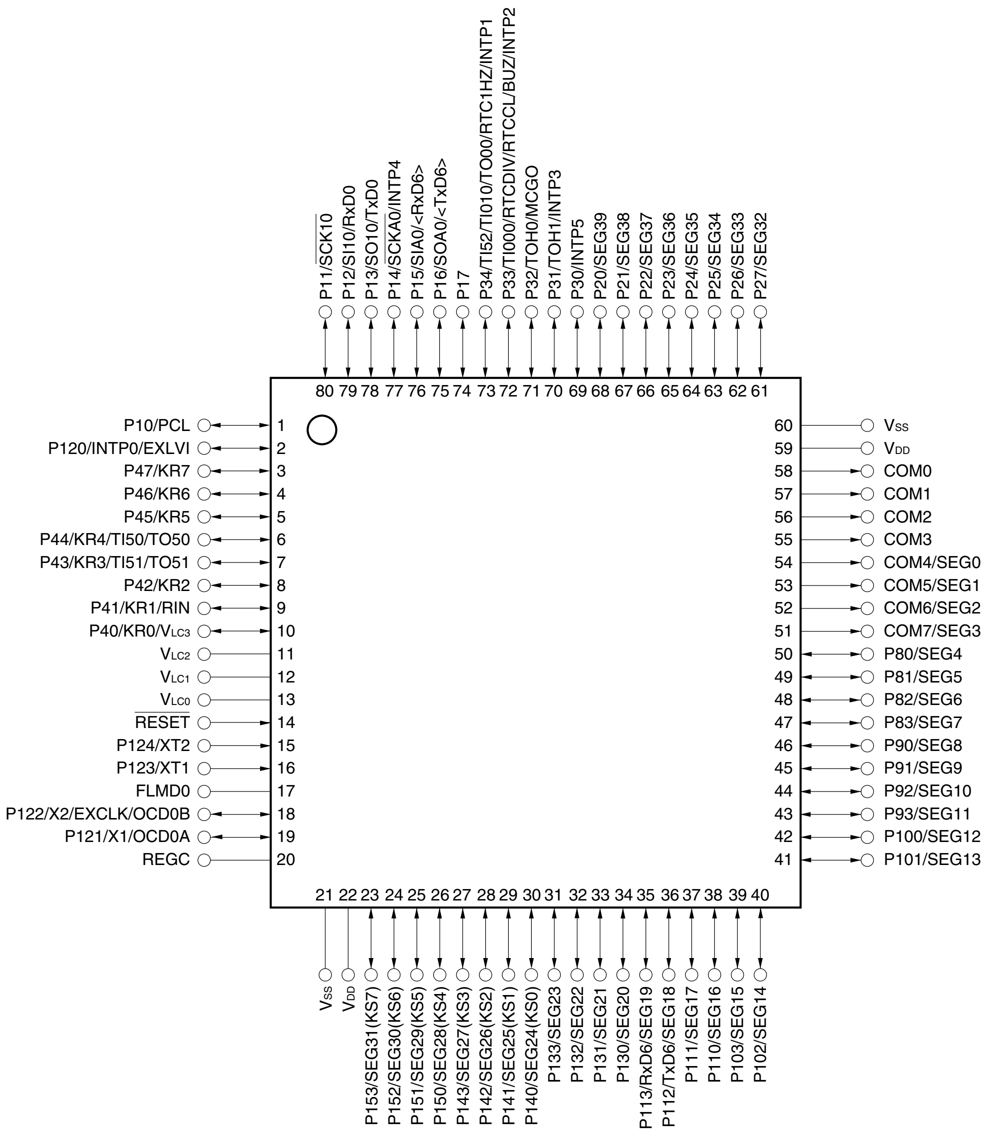
Part Number	Package
μPD78F0471GC-GAD-AX	80-pin plastic LQFP (14 × 14)
μPD78F0472GC-GAD-AX	80-pin plastic LQFP (14 × 14)
μPD78F0473GC-GAD-AX	80-pin plastic LQFP (14 × 14)
μPD78F0474GC-GAD-AX	80-pin plastic LQFP (14 × 14)
μPD78F0475GC-GAD-AX	80-pin plastic LQFP (14 × 14)
μPD78F0471GK-GAK-AX	80-pin plastic LQFP (fine pitch) (12 × 12)
μPD78F0472GK-GAK-AX	80-pin plastic LQFP (fine pitch) (12 × 12)
μPD78F0473GK-GAK-AX	80-pin plastic LQFP (fine pitch) (12 × 12)
μPD78F0474GK-GAK-AX	80-pin plastic LQFP (fine pitch) (12 × 12)
μPD78F0475GK-GAK-AX	80-pin plastic LQFP (fine pitch) (12 × 12)
μPD78F0481GC-GAD-AX	80-pin plastic LQFP (14 × 14)
μPD78F0482GC-GAD-AX	80-pin plastic LQFP (14 × 14)
μPD78F0483GC-GAD-AX	80-pin plastic LQFP (14 × 14)
μPD78F0484GC-GAD-AX	80-pin plastic LQFP (14 × 14)
μPD78F0485GC-GAD-AX	80-pin plastic LQFP (14 × 14)
μPD78F0481GK-GAK-AX	80-pin plastic LQFP (fine pitch) (12 × 12)
μPD78F0482GK-GAK-AX	80-pin plastic LQFP (fine pitch) (12 × 12)
μPD78F0483GK-GAK-AX	80-pin plastic LQFP (fine pitch) (12 × 12)
μPD78F0484GK-GAK-AX	80-pin plastic LQFP (fine pitch) (12 × 12)
μPD78F0485GK-GAK-AX	80-pin plastic LQFP (fine pitch) (12 × 12)
μPD78F0491GC-GAD-AX[Note]	80-pin plastic LQFP (14 × 14)
μPD78F0492GC-GAD-AX[Note]	80-pin plastic LQFP (14 × 14)
μPD78F0493GC-GAD-AX[Note]	80-pin plastic LQFP (14 × 14)
μPD78F0494GC-GAD-AX[Note]	80-pin plastic LQFP (14 × 14)
μPD78F0495GC-GAD-AX[Note]	80-pin plastic LQFP (14 × 14)
μPD78F0491GK-GAK-AX[Note]	80-pin plastic LQFP (fine pitch) (12 × 12)
μPD78F0492GK-GAK-AX[Note]	80-pin plastic LQFP (fine pitch) (12 × 12)
μPD78F0493GK-GAK-AX[Note]	80-pin plastic LQFP (fine pitch) (12 × 12)
μPD78F0494GK-GAK-AX[Note]	80-pin plastic LQFP (fine pitch) (12 × 12)
μPD78F0495GK-GAK-AX[Note]	80-pin plastic LQFP (fine pitch) (12 × 12)

Note Under development

1.4 Pin Configuration (Top View)

(1) μPD78F0471, 78F0472, 78F0473, 78F0474, 78F0475

- 80-pin plastic LQFP (14 × 14)
- 80-pin plastic LQFP (fine pitch) (12 × 12)



Cautions 1. Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF: recommended).

2. Only the bottom side pins (pin numbers 35 and 36) correspond to the UART6 pins (RxD6 and TxD6) when writing by a flash memory programmer. Writing cannot be performed by the top side pins (pin numbers 76 and 75).

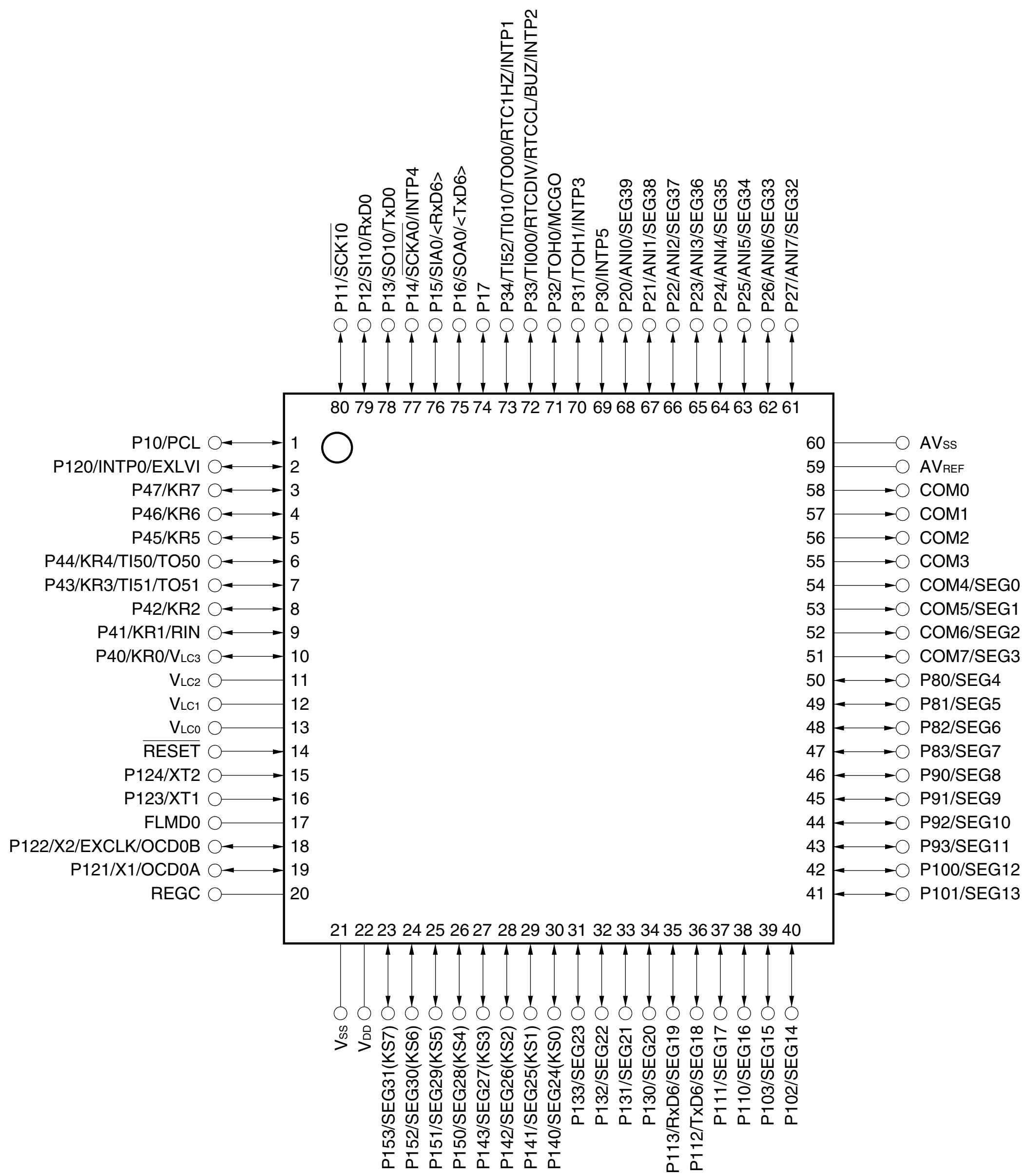
3. Make V_{DD} (pin number 22) and V_{DD} (pin number 59), V_{SS} (pin number 21) and V_{SS} (pin number 60) the same potential.

Remarks 1. The functions within arrowheads (< >) can be assigned by setting the input switch control register (ISC).

<R> **2.** The functions within parentheses can be used by setting the LCD mode register (LCDMD).

(2) μPD78F0481, 78F0482, 78F0483, 78F0484, 78F0485

- 80-pin plastic LQFP (14 × 14)
- 80-pin plastic LQFP (fine pitch) (12 × 12)



Cautions 1. Connect the AVSS pin to VSS.

2. Connect the REGC pin to VSS via a capacitor (0.47 to 1 μF: recommended).

3. ANI0/P20 to ANI7/P27 are set in the analog input mode after release of reset.

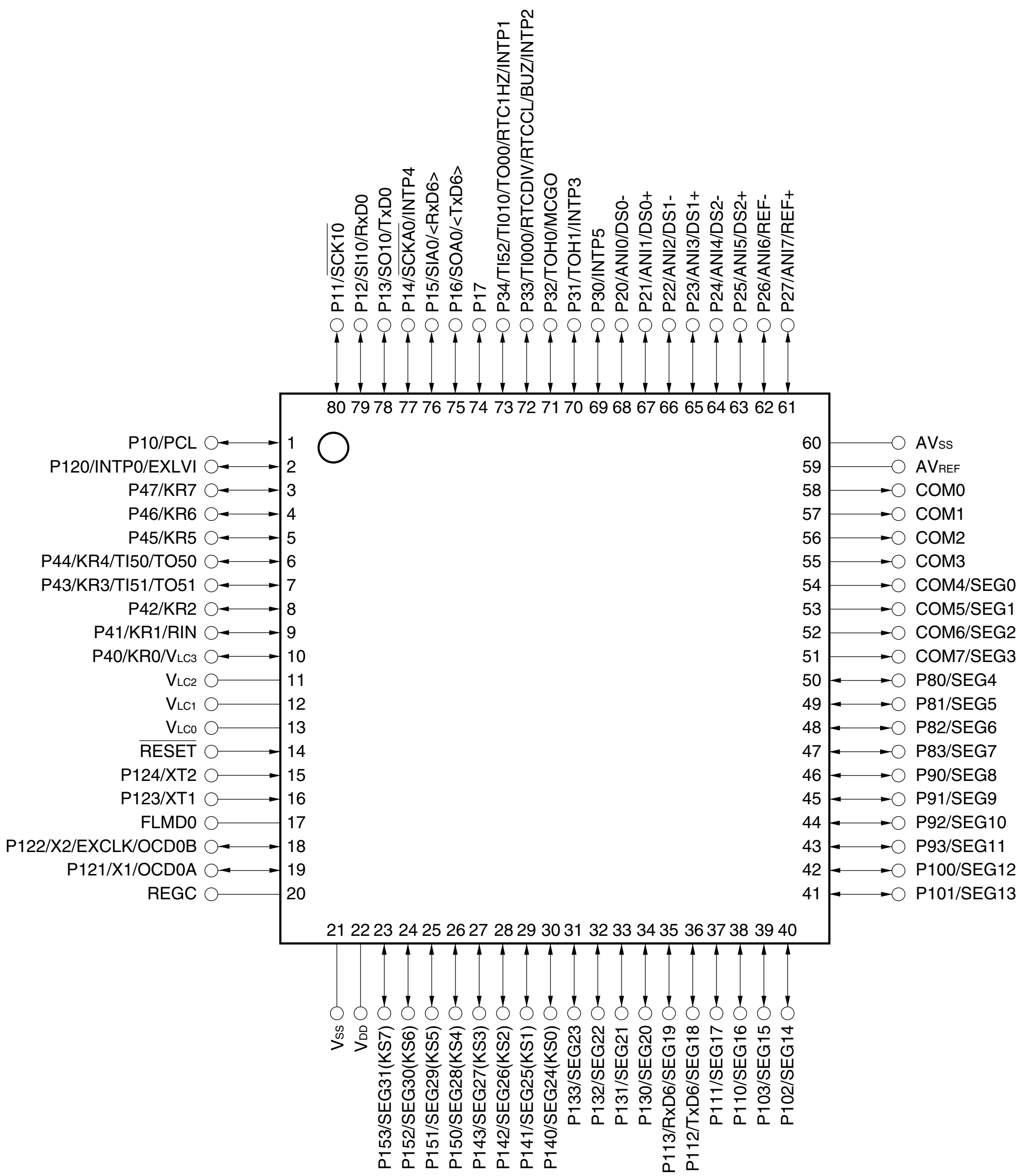
4. Only the bottom side pins (pin numbers 35 and 36) correspond to the UART6 pins (RxD6 and TxD6) when writing by a flash memory programmer. Writing cannot be performed by the top side pins (pin numbers 76 and 75).

Remarks 1. The functions within arrowheads (< >) can be assigned by setting the input switch control register (ISC).

<R> **2.** The functions within parentheses can be used by setting the LCD mode register (LCDMD).

(3) μPD78F0491, 78F0492, 78F0493, 78F0494, 78F0495

- 80-pin plastic LQFP (14 × 14)
- 80-pin plastic LQFP (fine pitch) (12 × 12)



Cautions 1. Connect the AVSS pin to VSS.

2. Connect the REGC pin to VSS via a capacitor (0.47 to 1 μF: recommended).

3. ANI0/P20 to ANI7/P27 are set in the analog input mode after release of reset.

4. Only the bottom side pins (pin numbers 35 and 36) correspond to the UART6 pins (RxD6 and TxD6) when writing by a flash memory programmer. Writing cannot be performed by the top side pins (pin numbers 76 and 75).

Remarks 1. The functions within arrowheads (< >) can be assigned by setting the input switch control register (ISC).

<R> **2.** The functions within parentheses can be used by setting the LCD mode register (LCDMD).

Pin Identification

ANI0 to ANI7[Note 1]:	Analog input
AV_{REF}[Note 1]:	Analog reference voltage
AV_{SS}[Note 1]:	Analog ground
BUZ:	Buzzer output
COM0 to COM7:	Common output
DS0+ to DS2+[Note 2]:	ΔΣ Analog input (+)
DS0− to DS2−[Note 2]:	ΔΣ Analog input (−)
EXCLK:	External clock input (main system clock)
EXLVI:	External potential input for low-voltage detector
FLMD0:	Flash programming mode
INTP0 to INTP5:	External interrupt input
KR0 to KR7:	Key return
MCGO:	Manchester code generator output
OCD0A, OCD0B:	On chip debug input/output
P10 to P17:	Port 1
P20 to P27:	Port 2
P30 to P34:	Port 3
P40 to P47:	Port 4
P80 to P83:	Port 8
P90 to P93:	Port 9
P100 to P103:	Port 10
P110 to P113:	Port 11
P120 to P124:	Port 12
P130 to P133:	Port 13
P140 to P143:	Port 14
P150 to P153:	Port 15
PCL:	Programmable clock output
REGC	Regulator capacitance
$\overline{\text{RESET}}$:	Reset
RxD0, RxD6:	Receive data
REF+[Note 2]:	ΔΣ Analog reference voltage (+)
REF−[Note 2]:	ΔΣ Analog reference voltage (−)
RIN:	Remote control input
RTC1HZ:	Real-time counter correction clock (1 Hz) output
RTCCL:	Real-time counter clock (32.768 kHz original oscillation) output
RTCDIV:	Real-time counter clock (32.768 kHz divided frequency) output
SEG0 to SEG31:	Segment output
SEG32 to SEG39[Note 3]:	Segment output
SEG24 (KS0) to SEG31 (KS7):	Segment key scan
$\overline{\text{SCK10}}$:	Serial clock input/output
$\overline{\text{SCKA0}}$:	Serial clock input/output
SI10:	Serial data input
SIA0:	Serial data input
SO10:	Serial data output
SOA0:	Serial data output
TI000, TI010:	Timer input
TI50, TI51, TI52:	Timer input
TO00:	Timer output
TO50, TO51:	Timer output
TOH0, TOH1:	Timer output
TxD0, TxD6:	Transmit data
V_{DD}:	Power supply
V_{SS}:	Ground
V_{LC0} to V_{LC3}:	LCD power supply
X1, X2:	Crystal oscillator (main system clock)
XT1, XT2:	Crystal oscillator (subsystem clock)

Notes 1. μPD78F048x and 78F049x only.
2. μPD78F049x only.
3. μPD78F047x and 78F048x only.

1.5 78K0/Lx3 Microcontroller Series Lineup

ROM	RAM	78K0/LC3	78K0/LD3	78K0/LE3	78K0/LF3
		48 Pins	52 Pins	64 Pins	80 Pins
60 KB	2 KB	–	–	μPD78F0465 μPD78F0455 μPD78F0445	μPD78F0495 μPD78F0485 μPD78F0475
48 KB	2 KB	–	–	μPD78F0464 μPD78F0454 μPD78F0444	μPD78F0494 μPD78F0484 μPD78F0474
32 KB	1 KB	μPD78F0413 μPD78F0403	μPD78F0433 μPD78F0423	μPD78F0463 μPD78F0453 μPD78F0443	μPD78F0493 μPD78F0483 μPD78F0473
24 KB	1 KB	μPD78F0412 μPD78F0402	μPD78F0432 μPD78F0422	μPD78F0462 μPD78F0452 μPD78F0442	μPD78F0492 μPD78F0482 μPD78F0472
16 KB	768 B	μPD78F0411 μPD78F0401	μPD78F0431 μPD78F0421	μPD78F0461 μPD78F0451 μPD78F0441	μPD78F0491 μPD78F0481 μPD78F0471
8 KB	512 B	μPD78F0410 μPD78F0400	μPD78F0430 μPD78F0420	–	–

The list of functions in the 78K0/Lx3 Microcontrollers is shown below. (1/3)

Item	Part Number																
		78K0/LC3								78K0/LD3							
		μPD78F040x				μPD78F041x				μPD78F042x				μPD78F043x			
		48 Pins								52 Pins							
Flash memory (KB)		8	16	24	32	8	16	24	32	8	16	24	32	8	16	24	32
RAM (KB)		0.5	0.75	1	1	0.5	0.75	1	1	0.5	0.75	1	1	0.5	0.75	1	1
Power supply voltage		V_{DD} = 1.8 to 5.5 V															
Regulator		Provided															
Minimum instruction execution time		0.2 μs (10 MHz: V_{DD} = 2.7 to 5.5 V)/ 0.4 μs (5 MHz: V_{DD} = 1.8 to 5.5 V)															
Clock (Main)	High-speed system clock	10 MHz: V_{DD} = 2.7 to 5.5 V/5 MHz: V_{DD} = 1.8 to 5.5 V															
Clock (Main)	Internal high-speed oscillation clock	8 MHz (TYP.): V_{DD} = 1.8 to 5.5 V															
Clock	Subclock	32.768 kHz (TYP.): V_{DD} = 1.8 to 5.5 V															
Clock	Internal low-speed oscillation clock	240 kHz (TYP.): V_{DD} = 1.8 to 5.5 V															
Port	Total	30								34							
Timer	16 bits (TM0)	1 ch															
Timer	8 bits (TM5)	3 ch															
Timer	8 bits (TMH)	3 ch															
Timer	RTC	1 ch															
Timer	WDT	1 ch															
Serial interface	3-wire CSI	–								1 ch Note 1							
Serial interface	UART	1 ch								1 ch Note 1							
Serial interface	UART supporting LIN-bus	1 ch Note 2								1 ch Note 3							
LCD	Type	External resistance division and internal resistance division are switchable.															
LCD	Segment signal	22 (18) [20 (16)] Note 4, 5								24 (20) [21 (17)] Note 4, 5							
LCD	Common signal	4 (8) Note 4															
10-bit successive approximation type A/D		–				6 ch				–				6 ch			
16-bit ΔΣ type A/D		–															
Interrupt	External	5															
Interrupt	Internal	17				18				19				20			
Segment key source signal output		8 ch								8 ch							
Key interrupt		3 ch								5 ch							
Reset	$\overline{\text{RESET}}$ pin	Provided															
Reset	POC	1.59 V ±0.15 V (Time for rising up to 1.8 V : 3.6 ms (MAX.))															
Reset	LVI	The detection level of the supply voltage is selectable in 16 steps.															
Reset	WDT	Provided															
Clock output		–															
Buzzer output		Provided															
Remote controller receiver		–								Provided							
MCG		Provided															
On-chip debug function		Provided															
Operating ambient temperature		T_A = –40 to +85°C															

Notes 1. Since 3-wire CSI and UART are used as alternate-function pins, they must be assigned to either of the functions for use.
2. The LIN-bus supporting UART pins can be changed to the UART pins (pin numbers 47 and 48).
3. The LIN-bus supporting UART pins can be changed to the 3-wire CSI/UART pins (pin numbers 50 and 51).
4. The values in parentheses are the number of signal outputs when 8com is used.
<R> **5.** The values in square brackets are the number of signal outputs when using the UART6 pins (RxD6, TxD6) on the bottom side.

(2/3)

Item \ Part Number	78K0/LE3														
	μPD78F044x					μPD78F045x					μPD78F046x				
	64 Pins														
Flash memory (KB)	16	24	32	48	60	16	24	32	48	60	16	24	32	48	60
RAM (KB)	0.75	1	1	2	2	0.75	1	1	2	2	0.75	1	1	2	2
Power supply voltage	V_{DD} = 1.8 to 5.5 V														
Regulator	Provided														
Minimum instruction execution time	0.2 μs (10 MHz: V_{DD} = 2.7 to 5.5 V)/ 0.4 μs (5 MHz: V_{DD} = 1.8 to 5.5 V)														
Clock – Main – High-speed system clock	10 MHz: V_{DD} = 2.7 to 5.5 V/5 MHz: V_{DD} = 1.8 to 5.5 V														
Clock – Main – Internal high-speed oscillation clock	8 MHz (TYP.): V_{DD} = 1.8 to 5.5 V														
Clock – Subclock	32.768 kHz (TYP.): V_{DD} = 1.8 to 5.5 V														
Clock – Internal low-speed oscillation clock	240 kHz (TYP.): V_{DD} = 1.8 to 5.5 V														
Port – Total	46														
Timer – 16 bits (TM0)	1 ch														
Timer – 8 bits (TM5)	3 ch														
Timer – 8 bits (TMH)	3 ch														
Timer – RTC	1 ch														
Timer – WDT	1 ch														
Serial interface – 3-wire CSI/UART Note 1	1 ch														
Serial interface – UART supporting LIN-bus Note 2	1 ch														
LCD – Type	External resistance division and internal resistance division are switchable.														
LCD – Segment signal	32 (28) [28 (24)] Note 3, 4										24 (20) [20 (16)] Note 3, 4				
LCD – Common signal	4 (8) Note 3														
10-bit successive approximation type A/D	–					8 ch									
16-bit ΔΣ type A/D	–										3 ch				
Interrupt – External	6														
Interrupt – Internal	19					20					21				
Segment key source signal output	8 ch														
Key interrupt	5 ch														
Reset – $\overline{RESET}$ pin	Provided														
Reset – POC	1.59 V ±0.15 V (Time for rising up to 1.8 V : 3.6 ms (MAX.))														
Reset – LVI	The detection level of the supply voltage is selectable in 16 steps.														
Reset – WDT	Provided														
Clock output	–														
Buzzer output	Provided														
Remote controller receiver	Provided														
MCG	Provided														
On-chip debug function	Provided														
Operating ambient temperature	T_A = –40 to +85°C														

Notes 1. Select either of the functions of these alternate-function pins.

2. The LIN-bus supporting UART pins can be changed to the 3-wire CSI/UART pins (pin numbers 62 and 63).

3. The values in parentheses are the number of signal outputs when 8com is used.

<R> **4.** The values in square brackets are the number of signal outputs when using the UART6 pins (RxD6, TxD6) on the bottom side.

(3/3)

Item \ Part Number	78K0/LF3														
	μPD78F047x					μPD78F048x					μPD78F049x				
	80 Pins														
Flash memory (KB)	16	24	32	48	60	16	24	32	48	60	16	24	32	48	60
RAM (KB)	0.75	1	1	2	2	0.75	1	1	2	2	0.75	1	1	2	2
Power supply voltage	V_{DD} = 1.8 to 5.5 V														
Regulator	Provided														
Minimum instruction execution time	0.2 μs (10 MHz: V_{DD} = 2.7 to 5.5 V)/ 0.4 μs (5 MHz: V_{DD} = 1.8 to 5.5 V)														
Clock – Main – High-speed system clock	10 MHz: V_{DD} = 2.7 to 5.5 V/5 MHz: V_{DD} = 1.8 to 5.5 V														
Clock – Main – Internal high-speed oscillation clock	8 MHz (TYP.): V_{DD} = 1.8 to 5.5 V														
Clock – Subclock	32.768 kHz (TYP.): V_{DD} = 1.8 to 5.5 V														
Clock – Internal low-speed oscillation clock	240 kHz (TYP.): V_{DD} = 1.8 to 5.5 V														
Port – Total	62														
Timer – 16 bits (TM0)	1 ch														
Timer – 8 bits (TM5)	3 ch														
Timer – 8 bits (TMH)	3 ch														
Timer – RTC	1 ch														
Timer – WDT	1 ch														
Serial interface – 3-wire CSI/UART Note 1	1 ch														
Serial interface – Automatic transmit/ receive 3-wire CSI	1 ch														
Serial interface – UART supporting LIN-bus Note 2	1 ch														
LCD – Type	External resistance division and internal resistance division are switchable.														
LCD – Segment signal	40 (36) [36 (32)] Note 3, 4										32 (28) [28 (24)] Note 3, 4				
LCD – Common signal	4 (8) Note 3														
10-bit successive approximation type A/D	–					8 ch									
16-bit ΔΣ type A/D	–										3 ch				
Interrupt – External	7														
Interrupt – Internal	20					21					22				
Segment key source signal output	8 ch														
Key interrupt	8 ch														
Reset – $\overline{RESET}$ pin	Provided														
Reset – POC	1.59 V ±0.15 V (Time for rising up to 1.8 V : 3.6 ms (MAX.))														
Reset – LVI	The detection level of the supply voltage is selectable in 16 steps.														
Reset – WDT	Provided														
Clock output/ Buzzer output	Provided														
Remote controller receiver	Provided														
MCG	Provided														
On-chip debug function	Provided														
Operating ambient temperature	T_A = –40 to +85°C														

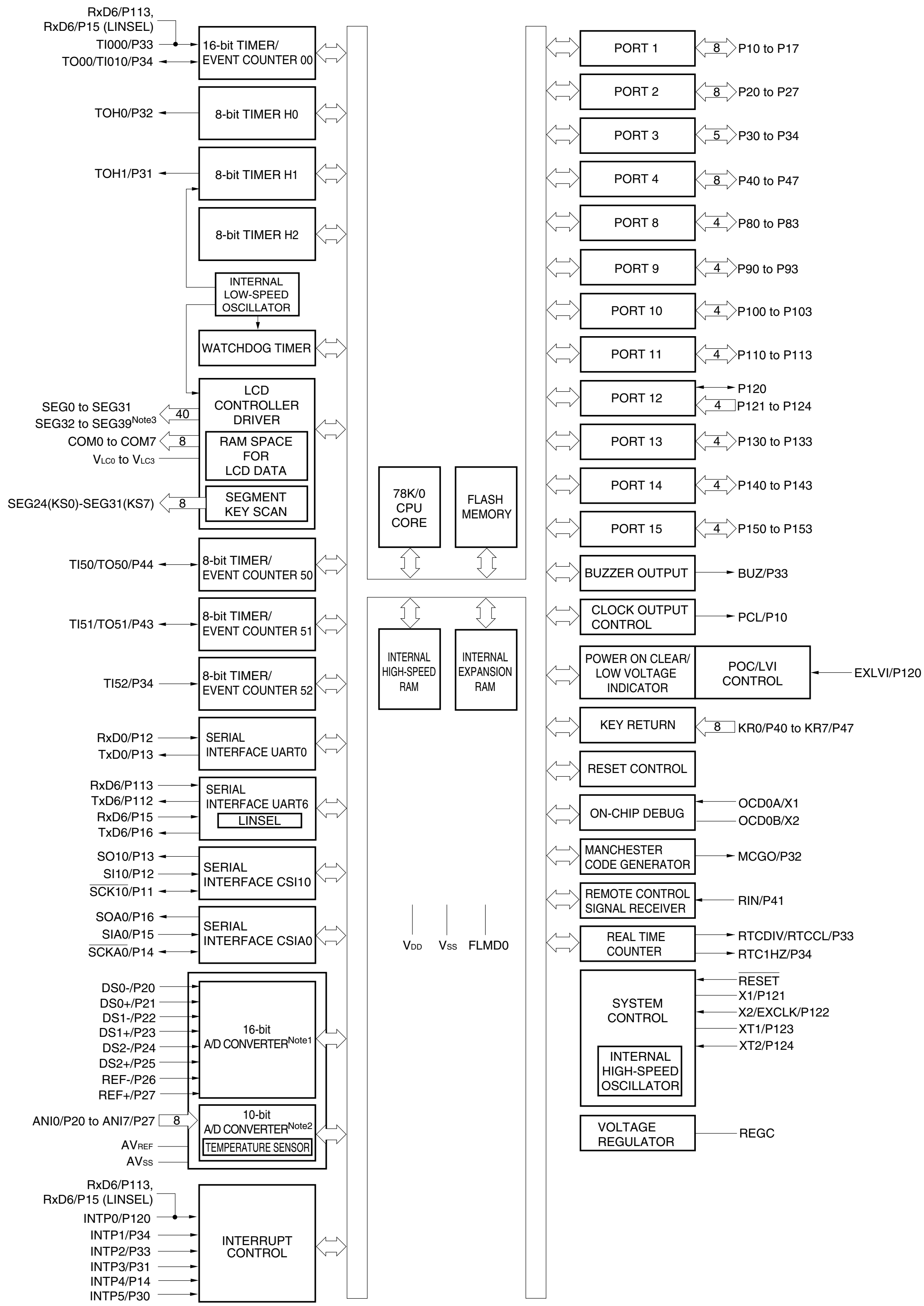
Notes 1. Select either of the functions of these alternate-function pins.

2. The LIN-bus supporting UART pins can be changed to the Automatic transmit/receive 3-wire CSI/UART pins (pin numbers 75 and 76).

3. The values in parentheses are the number of signal outputs when 8com is used.

<R> **4.** The values in square brackets are the number of signal outputs when using the UART6 pins (RxD6, TxD6) on the bottom side.

1.6 Block Diagram



Notes 1. μPD78F049x only.

2. μPD78F048x and 78F049x only.

3. μPD78F047x and 78F048x only.

1.7 Outline of Functions (μPD78F047x)

(1/2)

<table>
<tr><th colspan="2">Item</th><th>μPD78F0471</th><th>μPD78F0472</th><th>μPD78F0473</th><th>μPD78F0474</th><th>μPD78F0475</th></tr>
<tr><td rowspan="4">Internal memory</td><td>Flash memory (self-programming supported)[Note]</td><td>16 KB</td><td>24 KB</td><td>32 KB</td><td>48 KB</td><td>60 KB</td></tr>
<tr><td>High-speed RAM[Note]</td><td>768 bytes</td><td colspan="4">1 KB</td></tr>
<tr><td>Expansion RAM[Note]</td><td colspan="3">–</td><td colspan="2">1 KB</td></tr>
<tr><td>LCD display RAM</td><td colspan="5">40 × 4 bits (with 4 com) or 36 × 8 bits (with 8 com)</td></tr>
<tr><td colspan="2">Memory space</td><td colspan="5">64 KB</td></tr>
<tr><td rowspan="2">Main system clock (oscillation frequency)</td><td>High-speed system clock</td><td colspan="5">X1 (crystal/ceramic) oscillation, external main system clock input (EXCLK)
2 to 10 MHz: V_{DD} = 2.7 to 5.5 V,
2 to 5 MHz: V_{DD} = 1.8 to 5.5 V</td></tr>
<tr><td>Internal high-speed oscillation clock</td><td colspan="5">Internal oscillation
8 MHz (TYP.): V_{DD} = 1.8 to 5.5 V</td></tr>
<tr><td colspan="2">Subsystem clock (oscillation frequency)</td><td colspan="5">XT1 (crystal) oscillation
32.768 kHz (TYP.): V_{DD} = 1.8 to 5.5 V</td></tr>
<tr><td colspan="2">Internal low-speed oscillation clock (for TMH1, WDT)</td><td colspan="5">Internal oscillation
240 kHz (TYP.): V_{DD} = 1.8 to 5.5 V</td></tr>
<tr><td colspan="2">General-purpose registers</td><td colspan="5">8 bits × 32 registers (8 bits × 8 registers × 4 banks)</td></tr>
<tr><td colspan="2" rowspan="3">Minimum instruction execution time</td><td colspan="5">0.2 μs (high-speed system clock: @ f_{XH} = 10 MHz operation)</td></tr>
<tr><td colspan="5">0.25 μs (internal high-speed oscillation clock: @ f_{RH} = 8 MHz (TYP.) operation)</td></tr>
<tr><td colspan="5">122 μs (subsystem clock: @ f_{SUB} = 32.768 kHz operation)</td></tr>
<tr><td colspan="2">Instruction set</td><td colspan="5">• 8-bit operation and 16-bit operation
• Bit manipulate (set, reset, test, and Boolean operation)
• BCD adjust, etc.</td></tr>
<tr><td colspan="2">I/O ports</td><td colspan="5">Total: 62
CMOS I/O: 58
CMOS input: 4</td></tr>
<tr><td colspan="2">Timers</td><td colspan="5">• 16-bit timer/event counter: 1 channel
• 8-bit timer/event counter: 3 channels (out of which 2 channels can perform PWM output)
• 8-bit timer: 3 channels (out of which 2 channels can perform PWM output)
• Real-time counter: 1 channel
• Watchdog timer: 1 channel</td></tr>
<tr><td></td><td>Timer outputs</td><td colspan="5">5 (PWM output: 4 and PPG output: 1)</td></tr>
<tr><td></td><td>RTC outputs</td><td colspan="5">2
• 1 Hz (Subsystem clock: f_{SUB} = 32.768 kHz)
• 512 Hz or 16.384 kHz or 32.768 kHz (Subsystem clock: f_{SUB} = 32.768 kHz)</td></tr>
<tr><td colspan="2">Clock output</td><td colspan="5">• 156.25 kHz, 312.5 kHz, 625 kHz, 1.25 MHz, 5 MHz, 10 MHz
(peripheral hardware clock: @ f_{PRS} = 10 MHz operation)
• 32.768 kHz (subsystem clock: @ f_{SUB} = 32.768 kHz operation)</td></tr>
<tr><td colspan="2">Buzzer output</td><td colspan="5">• 1.22 kHz, 2.44 kHz, 4.88 kHz, 9.77 MHz
(peripheral hardware clock: @ f_{PRS} = 10 MHz operation)</td></tr>
</table>

Note The internal flash memory capacity, internal high-speed RAM capacity, and internal expansion RAM capacity can be changed using the internal memory size switching register (IMS) and the internal expansion RAM size switching register (IXS).

(2/2)

Item		μPD78F0471	μPD78F0472	μPD78F0473	μPD78F0474	μPD78F0475
10-bit successive approximation type A/D converter		–				
16-bit ΔΣ type A/D converter		–				
Serial interface		• UART supporting LIN-bus[Note 1]: 1 channel • 3-wire serial I/O/UART[Note 2]: 1 channel • Automatic transmit/receive 3-wire CSI: 1 channel				
LCD controller/driver		• External resistance division and internal resistance division are switchable. • Segment signal outputs: 40 (36) [36 (32)] [Note 3, 4] • Common signal outputs: 4 (8)[Note 3]				
Remote controller receiver		Provided				
Manchester code generator		Provided				
Vectored interrupt sources	Internal	20				
	External	7				
<R> Segment key source signal output		Segment key source signal outputs: 8 (SEG24(KS0)-SEG31(KS7))				
Key interrupt		Key interrupt (INTKR) occurs by detecting falling edge of key input pins (KR0 to KR7).				
Reset		• Reset using $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-clear • Internal reset by low-voltage detector				
On-chip debug function		Provided				
Power supply voltage		V_{DD} = 1.8 to 5.5 V				
Operating ambient temperature		T_A = –40 to +85°C				
Package		• 80-pin plastic LQFP (14 × 14) • 80-pin plastic LQFP (fine pitch) (12 × 12)				

Notes 1. The LIN-bus supporting UART pins can be changed to the automatic transmit/receive 3-wire CSI pins (pin numbers 75 and 76).

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

3. The values in parentheses are the number of signal outputs when 8com is used.

<R> **4.** The values in square brackets are the number of signal outputs when using the UART6 pins (RxD6, TxD6) on the bottom side.

An outline of the timer is shown below.

		16-Bit Timer/ Event Counters 00	8-Bit Timer/ Event Counters 50, 51, and 52			8-Bit Timers H0, H1, and H2			Real-time Counter	Watchdog Timer
		TM00	TM50	TM51	TM52	TMH0	TMH1	TMH2		
Function	Interval timer	1 channel	1 channel	1 channel	1 channel	1 channel	1 channel	1 channel	1 channel [Note 1]	–
	External event counter	1 channel [Note 2]	1 channel	1 channel	1 channel [Note 2]	–	–	– [Note 2]	–	–
	PPG output	1 output	–	–	–	–	–	–	–	–
	PWM output	–	1 output	1 output	–	1 output	1 output	–	–	–
	Pulse width measurement	2 inputs	–	–	–	–	–	–	–	–
	Square-wave output	1 output	1 output	1 output	–	1 output	1 output	–	–	–
	Carrier generator	–	–	– [Note 3]	–	–	1 output [Note 3]	–	–	–
	Calendar function	–	–	–	–	–	–	–	1 channel [Note 1]	
	RTC output	–	–	–	–	–	–	–	2 outputs [Note 4]	–
	Watchdog timer	–	–	–	–	–	–	–	–	1 channel
Interrupt source		2	1	1	1	1	1	1	1	–

Notes 1. In the real-time counter, the Interval timer function and calendar function can be used simultaneously.

2. TM52 and TM00 can be connected in cascade to be used as a 24-bit counter. Also, the external event input of TM52 can be input enable-controlled via TMH2.

3. TM51 and TMH1 can be used in combination as a carrier generator mode.

4. A 1 Hz output can be used as one output and a 512 Hz, 16.384 kHz, or 32.768 kHz output can be used as one output.

1.8 Outline of Functions (μPD78F048x)

(1/2)

Item		μPD78F0481	μPD78F0482	μPD78F0483	μPD78F0484	μPD78F0485
Internal memory	Flash memory (self-programming supported)[Note]	16 KB	24 KB	32 KB	48 KB	60 KB
	High-speed RAM[Note]	768 bytes	1 KB			
	Expansion RAM[Note]	–			1 KB	
	LCD display RAM	40 × 4 bits (with 4 com) or 36 × 8 bits (with 8 com)				
Memory space		64 KB				
Main system clock (oscillation frequency)	High-speed system clock	X1 (crystal/ceramic) oscillation, external main system clock input (EXCLK) 2 to 10 MHz: V_{DD} = 2.7 to 5.5 V, 2 to 5 MHz: V_{DD} = 1.8 to 5.5 V				
	Internal high-speed oscillation clock	Internal oscillation 8 MHz (TYP.): V_{DD} = 1.8 to 5.5 V				
Subsystem clock (oscillation frequency)		XT1 (crystal) oscillation 32.768 kHz (TYP.): V_{DD} = 1.8 to 5.5 V				
Internal low-speed oscillation clock (for TMH1, WDT)		Internal oscillation 240 kHz (TYP.): V_{DD} = 1.8 to 5.5 V				
General-purpose registers		8 bits × 32 registers (8 bits × 8 registers × 4 banks)				
Minimum instruction execution time		0.2 μs (high-speed system clock: @ f_{XH} = 10 MHz operation)				
		0.25 μs (internal high-speed oscillation clock: @ f_{RH} = 8 MHz (TYP.) operation)				
		122 μs (subsystem clock: @ f_{SUB} = 32.768 kHz operation)				
Instruction set		• 8-bit operation and 16-bit operation • Bit manipulate (set, reset, test, and Boolean operation) • BCD adjust, etc.				
I/O ports		Total: 62 CMOS I/O: 58 CMOS input: 4				
Timers		• 16-bit timer/event counter: 1 channel • 8-bit timer/event counter: 3 channels (out of which 2 channels can perform PWM output) • 8-bit timer: 3 channels (out of which 2 channels can perform PWM output) • Real-time counter: 1 channel • Watchdog timer: 1 channel				
	Timer outputs	5 (PWM output: 4 and PPG output: 1)				
	RTC outputs	2 • 1 Hz (Subsystem clock: f_{SUB} = 32.768 kHz) • 512 Hz or 16.384 kHz or 32.768 kHz (Subsystem clock: f_{SUB} = 32.768 kHz)				
Clock output		• 156.25 kHz, 312.5 kHz, 625 kHz, 1.25 MHz, 5 MHz, 10 MHz (peripheral hardware clock: @ f_{PRS} = 10 MHz operation) • 32.768 kHz (subsystem clock: @ f_{SUB} = 32.768 kHz operation)				
Buzzer output		• 1.22 kHz, 2.44 kHz, 4.88 kHz, 9.77 MHz (peripheral hardware clock: @ f_{PRS} = 10 MHz operation)				

Note The internal flash memory capacity, internal high-speed RAM capacity, and internal expansion RAM capacity can be changed using the internal memory size switching register (IMS) and the internal expansion RAM size switching register (IXS).

(2/2)

Item		μPD78F0481	μPD78F0482	μPD78F0483	μPD78F0484	μPD78F0485
10-bit successive approximation type A/D converter		10-bit resolution × 8 channels (AV_{REF} = 2.3 to 5.5 V)				
16-bit ΔΣ type A/D converter		–				
Serial interface		• UART supporting LIN-bus[Note 1]: 1 channel • 3-wire serial I/O/UART[Note 2]: 1 channel • Automatic transmit/receive 3-wire CSI:1 channel				
LCD controller/driver		• External resistance division and internal resistance division are switchable. • Segment signal outputs: 40 (36) [36 (32)] [Note 3 4] • Common signal outputs: 4 (8)[Note 3]				
Remote controller receiver		Provided				
Manchester code generator		Provided				
Vectored interrupt sources	Internal	21				
	External	7				
<R> Segment key source signal output		Segment key source signal outputs: 8 (SEG24(KS0)-SEG31(KS7))				
Key interrupt		Key interrupt (INTKR) occurs by detecting falling edge of key input pins (KR0 to KR7).				
Reset		• Reset using $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-clear • Internal reset by low-voltage detector				
On-chip debug function		Provided				
Power supply voltage		V_{DD} = 1.8 to 5.5 V				
Operating ambient temperature		T_A = –40 to +85°C				
Package		• 80-pin plastic LQFP (14 × 14) • 80-pin plastic LQFP (fine pitch) (12 × 12)				

Notes 1. The LIN-bus supporting UART pins can be changed to the automatic transmit/receive 3-wire CSI pins (pin numbers 75 and 76).

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

3. The values in parentheses are the number of signal outputs when 8com is used.

<R> **4.** The values in square brackets are the number of signal outputs when using the UART6 pins (RxD6, TxD6) on the bottom side.

An outline of the timer is shown below.

		16-Bit Timer/ Event Counters 00	8-Bit Timer/ Event Counters 50, 51, and 52			8-Bit Timers H0, H1, and H2			Real-time Counter	Watchdog Timer
		TM00	TM50	TM51	TM52	TMH0	TMH1	TMH2		
Function	Interval timer	1 channel	1 channel	1 channel	1 channel	1 channel	1 channel	1 channel	1 channel[Note 1]	–
	External event counter	1 channel[Note 2]	1 channel	1 channel	1 channel[Note 2]	–	–	–[Note 2]	–	–
	PPG output	1 output	–	–	–	–	–	–	–	–
	PWM output	–	1 output	1 output	–	1 output	1 output	–	–	–
	Pulse width measurement	2 inputs	–	–	–	–	–	–	–	–
	Square-wave output	1 output	1 output	1 output	–	1 output	1 output	–	–	–
	Carrier generator	–	–	–[Note 3]	–	–	1 output[Note 3]	–	–	–
	Calendar function	–	–	–	–	–	–	–	1 channel[Note 1]	
	RTC output	–	–	–	–	–	–	–	2 outputs[Note 4]	–
	Watchdog timer	–	–	–	–	–	–	–	–	1 channel
Interrupt source		2	1	1	1	1	1	1	1	–

Notes 1. In the real-time counter, the Interval timer function and calendar function can be used simultaneously.

2. TM52 and TM00 can be connected in cascade to be used as a 24-bit counter. Also, the external event input of TM52 can be input enable-controlled via TMH2.

3. TM51 and TMH1 can be used in combination as a carrier generator mode.

4. A 1 Hz output can be used as one output and a 512 Hz, 16.384 kHz, or 32.768 kHz output can be used as one output.

1.9 Outline of Functions (μPD78F049x)

(1/2)

Item		μPD78F0491	μPD78F0492	μPD78F0493	μPD78F0494	μPD78F0495
Internal memory	Flash memory (self-programming supported)[Note]	16 KB	24 KB	32 KB	48 KB	60 KB
	High-speed RAM[Note]	768 bytes	1 KB			
	Expansion RAM[Note]	–			1 KB	
	LCD display RAM	32 × 4 bits (with 4 com) or 28 × 8 bits (with 8 com)				
Memory space		64 KB				
Main system clock (oscillation frequency)	High-speed system clock	X1 (crystal/ceramic) oscillation, external main system clock input (EXCLK) 2 to 10 MHz: V_{DD} = 2.7 to 5.5 V, 2 to 5 MHz: V_{DD} = 1.8 to 5.5 V				
	Internal high-speed oscillation clock	Internal oscillation 8 MHz (TYP.): V_{DD} = 1.8 to 5.5 V				
Subsystem clock (oscillation frequency)		XT1 (crystal) oscillation 32.768 kHz (TYP.): V_{DD} = 1.8 to 5.5 V				
Internal low-speed oscillation clock (for TMH1, WDT)		Internal oscillation 240 kHz (TYP.): V_{DD} = 1.8 to 5.5 V				
General-purpose registers		8 bits × 32 registers (8 bits × 8 registers × 4 banks)				
Minimum instruction execution time		0.2 μs (high-speed system clock: @ f_{XH} = 10 MHz operation)				
		0.25 μs (internal high-speed oscillation clock: @ f_{RH} = 8 MHz (TYP.) operation)				
		122 μs (subsystem clock: @ f_{SUB} = 32.768 kHz operation)				
Instruction set		• 8-bit operation and 16-bit operation • Bit manipulate (set, reset, test, and Boolean operation) • BCD adjust, etc.				
I/O ports		Total: 62 CMOS I/O: 58 CMOS input: 4				
Timers		• 16-bit timer/event counter: 1 channel • 8-bit timer/event counter: 3 channels (out of which 2 channels can perform PWM output) • 8-bit timer: 3 channels (out of which 2 channels can perform PWM output) • Real-time counter: 1 channel • Watchdog timer: 1 channel				
	Timer outputs	5 (PWM output: 4 and PPG output: 1)				
	RTC outputs	2 • 1 Hz (Subsystem clock: f_{SUB} = 32.768 kHz) • 512 Hz or 16.384 kHz or 32.768 kHz (Subsystem clock: f_{SUB} = 32.768 kHz)				
Clock output		• 156.25 kHz, 312.5 kHz, 625 kHz, 1.25 MHz, 5 MHz, 10 MHz (peripheral hardware clock: @ f_{PRS} = 10 MHz operation) • 32.768 kHz (subsystem clock: @ f_{SUB} = 32.768 kHz operation)				
Buzzer output		• 1.22 kHz, 2.44 kHz, 4.88 kHz, 9.77 MHz (peripheral hardware clock: @ f_{PRS} = 10 MHz operation)				

Note The internal flash memory capacity, internal high-speed RAM capacity, and internal expansion RAM capacity can be changed using the internal memory size switching register (IMS) and the internal expansion RAM size switching register (IXS).

(2/2)

Item		μPD78F0491	μPD78F0492	μPD78F0493	μPD78F0494	μPD78F0495
10-bit successive approximation type A/D converter		10-bit resolution × 8 channels (AV_{REF} = 2.3 to 5.5 V)				
16-bit ΔΣ type[Note 1] A/D converter		16-bit resolution × 3 channels (AV_{REF} = 2.7 to 5.5 V)				
Serial interface		• UART supporting LIN-bus[Note 2]: 1 channel • 3-wire serial I/O/UART[Note 3]: 1 channel • Automatic transmit/receive 3-wire CSI:1 channel				
LCD controller/driver		• External resistance division and internal resistance division are switchable. • Segment signal outputs: 32 (28) [28 (24)][Note 4, 5] • Common signal outputs: 4 (8)[Note 4]				
Remote controller receiver		Provided				
Manchester code generator		Provided				
Vectored interrupt sources	Internal	22				
	External	7				
<R> Segment key source signal output		Segment key source signal outputs: 8 (SEG24(KS0)-SEG31(KS7))				
Key interrupt		Key interrupt (INTKR) occurs by detecting falling edge of key input pins (KR0 to KR7).				
Reset		• Reset using $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-clear • Internal reset by low-voltage detector				
On-chip debug function		Provided				
Power supply voltage		V_{DD} = 1.8 to 5.5 V				
Operating ambient temperature		T_A = −40 to +85°C				
Package		• 80-pin plastic LQFP (14 × 14) • 80-pin plastic LQFP (fine pitch) (12 × 12)				

Notes 1. The specifications of the 16-bit ΔΣ A/D converter may have been changed.
For details of the specifications, contact an NEC Electronics sales representative or authorized dealer.

2. The LIN-bus supporting UART pins can be changed to the automatic transmit/receive 3-wire CSI pins (pin numbers 75 and 76).

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

4. The values in parentheses are the number of signal outputs when 8com is used.

<R> **5.** The values in square brackets are the number of signal outputs when using the UART6 pins (RxD6, TxD6) on the bottom side.

An outline of the timer is shown below.

		16-Bit Timer/ Event Counters 00	8-Bit Timer/ Event Counters 50, 51, and 52			8-Bit Timers H0, H1, and H2			Real-time Counter	Watchdog Timer
		TM00	TM50	TM51	TM52	TMH0	TMH1	TMH2		
Function	Interval timer	1 channel	1 channel	1 channel	1 channel	1 channel	1 channel	1 channel	1 channel Note 1	–
	External event counter	1 channel Note 2	1 channel	1 channel	1 channel Note 2	–	–	– Note 2	–	–
	PPG output	1 output	–	–	–	–	–	–	–	–
	PWM output	–	1 output	1 output	–	1 output	1 output	–	–	–
	Pulse width measurement	2 inputs	–	–	–	–	–	–	–	–
	Square-wave output	1 output	1 output	1 output	–	1 output	1 output	–	–	–
	Carrier generator	–	–	– Note 3	–	–	1 output Note 3	–	–	–
	calendar function	–	–	–	–	–	–	–	1 channel Note 1	
	RTC output	–	–	–	–	–	–	–	2 outputs Note 4	–
	Watchdog timer	–	–	–	–	–	–	–	–	1 channel
Interrupt source		2	1	1	1	1	1	1	1	–

Notes 1. In the real-time counter, the Interval timer function and calendar function can be used simultaneously.

2. TM52 and TM00 can be connected in cascade to be used as a 24-bit counter. Also, the external event input of TM52 can be input enable-controlled via TMH2.

3. TM51 and TMH1 can be used in combination as a carrier generator mode.

4. A 1 Hz output can be used as one output and a 512 Hz, 16.384 kHz, or 32.768 kHz output can be used as one output.

2.1 Pin Function List

There are three types of pin I/O buffer power supplies: AV_{REF}[Note 1], V_{LC0}, and V_{DD}. The relationship between these power supplies and the pins is shown below.

Table 2-1. Pin I/O Buffer Power Supplies

Power Supply	Corresponding Pins
AV_{REF}[Note 1]	P20 to P27
V_{LC0}	COM0 to COM7, SEG0 to SEG31, SEG32 to SEG39[Note 2], V_{LC0} to V_{LC3}
V_{DD}	Pins other than above

Notes 1. μPD78F048x and 78F049x only. The power supply is V_{DD} with μPD78F047x.

2. μPD78F047x and 78F048x only.

(1) Port pins **(1/3)**

Function Name	I/O	Function	After Reset	Alternate Function
P10	I/O	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	PCL
P11				$\overline{SCK10}$
P12				SI10/RxD0
P13				SO10/TxD0
P14				$\overline{SCKA0}$/INTP4
P15				SIA0/<RxD6>
P16				SOA0/<TxD6>
P17				−

Remark The functions within arrowheads (< >) can be assigned by setting the input switch control register (ISC).

(1) Port pins **(2/3)**

Function Name	I/O	Function	After Reset	Alternate Function
P20	I/O	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	SEG39[Note 1]/ANI0[Note 2]/DS0−[Note 3]
P21				SEG38[Note 1]/ANI1[Note 2]/DS0+[Note3]
P22				SEG37[Note 1]/ANI2[Note 2]/DS1−[Note3]
P23				SEG36[Note 1]/ANI3[Note 2]/DS1+[Note3]
P24				SEG35[Note 1]/ANI4[Note 2]/DS2−[Note 3]
P25				SEG34[Note 1]/ANI5[Note 2]/DS2+[Note 3]
P26				SEG33[Note 1]/ANI6[Note 2]/REF−[Note 3]
P27				SEG32[Note 1]/ANI7[Note 2]/REF+[Note 3]
P30	I/O	Port 3. 5-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP5
P31				TOH1/INTP3
P32				TOH0/MCGO
P33				TI000/RTCDIV/RTCCL/BUZ/INTP2
P34				TI52/TI010/TO00/RTC1HZ/INTP1
P40	I/O	Port 4. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	V_{LC3}/KR0
P41				RIN/KR1
P42				KR2
P43				TO51/TI51/KR3
P44				TO50/TI50/KR4
P45 to P47				KR5 to KR7
P80 to P83	I/O	Port 8. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SEG4 to SEG7
P90 to P93	I/O	Port 9. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SEG8 to SEG11

Notes 1. μPD78F047x and 78F048x only.
2. μPD78F048x and 78F049x only.
3. μPD78F049x only.

(1) Port pins **(3/3)**

Function Name	I/O	Function	After Reset	Alternate Function
P100 to P103	I/O	Port 10. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SEG12 to SEG15
P110, P111	I/O	Port 11. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SEG16, SEG17
P112				SEG18/TxD6
P113				SEG19/RxD6
P120	I/O	Port 12. 1-bit I/O port and 4-bit input port. Only for P120, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP0/EXLVI
P121	Input			X1/OCD0A
P122				X2/EXCLK/OCD0B
P123				XT1
P124				XT2
P130 to P133	I/O	Port 13. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SEG20 to SEG23
P140 to P143	I/O	Port 14. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SEG24 (KS0) to SEG27 (KS3)
P150 to P153	I/O	Port 15. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SEG28 (KS4) to SEG31 (KS7)

(2) Non-port pins **(1/4)**

Function Name	I/O	Function	After Reset	Alternate Function
ANI0[Note 2]	Input	10-bit successive approximation type A/D converter analog input.	Digital input port	P20/SEG39[Note 1]/ DS0−[Note 3]
ANI1[Note 2]				P21/SEG38[Note 1]/ DS0+[Note 3]
ANI2[Note 2]				P22/SEG37[Note 1]/ DS1−[Note 3]
ANI3[Note 2]				P23/SEG36[Note 1]/ DS1+[Note 3]
ANI4[Note 2]				P24/SEG35[Note 1]/ DS2−[Note 3]
ANI5[Note 2]				P25/SEG34[Note 1]/ DS2+[Note 3]
ANI6[Note 2]				P26/SEG33[Note 1]/ REF−[Note 3]
ANI7[Note 2]				P27/SEG32[Note 1]/ REF+[Note 3]
DS0−[Note 3]	Input	16-bit ΔΣ type A/D converter analog input.	Digital input port	P20 /ANI0[Note 2]
DS0+[Note 3]				P21/ANI1[Note 2]
DS1−[Note 3]				P22/ANI2[Note 2]
DS1+[Note 3]				P23/ANI3[Note 2]
DS2−[Note 3]				P24/ANI4[Note 2]
DS2+[Note 3]				P25/ANI5[Note 2]
REF−[Note 3]		16-bit ΔΣ type A/D converter reference voltage input. Make the same potential as V_{SS} and AV_{SS}.		P26/ANI6[Note 2]
REF+[Note 3]		16-bit ΔΣ type A/D converter reference voltage input. Make the same potential as AV_{REF}.		P27/ANI7[Note 2]
AV_{REF}[Note 2]	Input	10-bit successive approximation type A/D converter reference voltage input, positive power supply for port 2, and 16-bit ΔΣ type A/D converter [Note3]	–	–
AV_{SS}[Note 2]	–	A/D converter ground potential. Make the same potential as V_{SS}.	–	–

Notes 1. μPD78F047x and 78F048x only.

2. μPD78F048x and 78F049x only.

3. μPD78F049x only.

(2) Non-port pins **(2/4)**

	Function Name	I/O	Function	After Reset	Alternate Function
	SEG0 to SEG3	Output	LCD controller/driver segment signal outputs	Output	COM4 to COM7
	SEG4 to SEG7			Input port	P80 to P83
	SEG8 to SEG11				P90 to P93
	SEG12 to SEG15				P100 to P103
	SEG16, SEG17				P110, P111
	SEG18				P112/TxD6
	SEG19				P113/RxD6
	SEG20 to SEG23				P130 to P133
<R>	SEG24 (KS0) to SEG27 (KS3)		LCD controller/driver segment signal outputs. Segment key source signal can be simultaneously output.		P140 to P143
<R>	SEG28 (KS4) to SEG31 (KS7)				P150 to P153
	SEG32[Note 1]		LCD controller/driver segment signal outputs	Digital input port	P27/ANI7[Note 2]
	SEG33[Note 1]				P26/ANI6[Note 2]
	SEG34[Note 1]				P25/ANI5[Note 2]
	SEG35[Note 1]				P24/ANI4[Note 2]
	SEG36[Note 1]				P23/ANI3[Note 2]
	SEG37[Note 1]				P22/ANI2[Note 2]
	SEG38[Note 1]				P21/ANI1[Note 2]
	SEG39[Note 1]				P20/ANI0[Note 2]
	COM0 to COM3	Output	LCD controller/driver common signal outputs	Output	–
	COM4 to COM7				SEG0 to SEG3
	V_{LC0} to V_{LC2}	–	LCD drive voltage	–	–
	V_{LC3}			Input port	P40/KR0

Notes 1. μPD78F047x and 78F048x only.

2. μPD78F048x and 78F049x only.

(2) Non-port pins (3/4)

Function Name	I/O	Function	After Reset	Alternate Function
BUZ	Output	Buzzer output	Input port	P33/TI000/RTCDIV /RTCCL/INTP2
INTP0	Input	External interrupt request input for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified	Input port	P120/EXLVI
INTP1				P34/TI52/TI010/ TO00/RTC1HZ
INTP2				P33/TI000/RTCDIV /RTCCL/BUZ
INTP3				P31/TOH1
INTP4				P14/$\overline{\text{SCKA0}}$
INTP5				P30
<R> KR0	Input	Key interrupt input or segment key scan input	Input port	P40/V_{LC3}
KR1				P41/RIN
KR2				P42
KR3				P43/TO51/TI51
KR4				P44/TO50/TI50
KR5 to KR7				P45 to P47
MCGO	Output	Manchester code output	Input port	P32/TOH0
PCL	Output	Clock output	Input port	P10
REGC	–	Connecting regulator output (2.4 V) stabilization capacitance for internal operation. Connect to V_{SS} via a capacitor (0.47 to 1 μF: recommended).	–	–
$\overline{\text{RESET}}$	Input	System reset input	–	–
RIN	Input	Remote control reception data input	Input port	P41/KR1
RTCDIV	Output	Real-time counter clock (32 kHz divided frequency) output	Input port	P33/TI000/RTCCL /BUZ/INTP2
RTCCL	Output	Real-time counter clock (32 kHz original oscillation) output	Input port	P33/TI000/RTCDIV /BUZ/INTP2
RTC1HZ	Output	Real-time counter clock (1 Hz) output	Input port	P34/TI52/TI010/ TO00/INTP1
RxD0	Input	Serial data input to asynchronous serial interface	Input port	P12/SI10
RxD6				P113/SEG19
<RxD6>				P15/SIA0
SI10	Input	Serial data input to CSI10	Input port	P12/RxD0
SIA0	Input	Serial data input to CSIA0	Input port	P15/<RxD6>
SO10	Output	Serial data output from CSI10	Input port	P13/TxD0
SOA0	Output	Serial data output from CSIA0	Input port	P16/<TxD6>
$\overline{\text{SCK10}}$	I/O	Clock input/output for CSI10	Input port	P11
$\overline{\text{SCKA0}}$	I/O	Clock input/output for CSIA0	Input port	P14/INTP4

Remark The functions within arrowheads (< >) can be assigned by setting the input switch control register (ISC).

(2) Non-port pins **(4/4)**

Function Name	I/O	Function	After Reset	Alternate Function
TI000	Input	External count clock input to 16-bit timer/event counter 00 Capture trigger input to capture registers (CR000, CR010) of 16-bit timer/event counter 00	Input port	P33/RTCDIV/RTCCL/BUZ/INTP2
TI010		Capture trigger input to capture register (CR000) of 16-bit timer/event counter 00		P34/TI52/TO00/RTC1HZ/INTP1
TI50	Input	External count clock input to 8-bit timer/event counter 50	Input port	P44/TO50/KR4
TI51		External count clock input to 8-bit timer/event counter 51		P43/TO51/KR3
TI52		External count clock input to 8-bit timer/event counter 52		P34/TI010/TO00/RTC1HZ/INTP1
TO00	Output	16-bit timer/event counter 00 output	Input port	P34/TI52/TI010/RTC1HZ/INTP1
TO50	Output	8-bit timer/event counter 50 output	Input port	P44/TI50/KR4
TO51		8-bit timer/event counter 51 output		P43/TI51/KR3
TOH0	Output	8-bit timer H0 output	Input port	P32/MCGO
TOH1		8-bit timer H1 output		P31/INTP3
TxD0	Output	Serial data output from asynchronous serial interface	Input port	P13/SO10
TxD6				P112/SEG18
<TxD6>				P16/SOA0
EXLVI	Input	Potential input for external low-voltage detection	Input port	P120/INTP0
X1	Input	Connecting resonator for main system clock	Input port	P121/OCD0A
X2	–			P122/EXCLK/OCD0B
EXCLK	Input	External clock input for main system clock	Input port	P122/X2/OCD0B
XT1	Input	Connecting resonator for subsystem clock	Input port	P123
XT2	–			P124
V_{DD}	–	Positive power supply	–	–
V_{SS}	–	Ground potential	–	–
FLMD0	–	Flash memory programming mode setting	–	–
OCD0A	Input	On-chip debug mode setting connection	Input port	P121/X1
OCD0B	–			P122/X2/EXCLK

Remark The functions within arrowheads (< >) can be assigned by setting the input switch control register (ISC).

2.2 Description of Pin Functions

2.2.1 P10 to P17 (port 1)

P10 to P17 function as an 8-bit I/O port. These pins also function as pins for external interrupt request input, serial interface data I/O, clock I/O, and clock output. P13 and P16 can be selected to function as pins, using port function register 1 (PF1) (see Figure 4-30).

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

(1) Port mode

P10 to P17 function as an 8-bit I/O port. P10 to P17 can be set to input or output port in 1-bit units using port mode register 1 (PM1). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 1 (PU1).

(2) Control mode

P10 to P17 function as external interrupt request input, serial interface data I/O, clock I/O, and clock output.

(a) SI10, SIA0

These are serial interface serial data input pins.

(b) SO10, SOA0

These are serial interface serial data output pins.

(c) $\overline{\text{SCK10}}$, $\overline{\text{SCKA0}}$

These are serial interface serial clock I/O pins.

(d) RxD0, RxD6

These are the serial data input pins of the asynchronous serial interface.

(e) TxD0, TxD6

These are the serial data output pins of the asynchronous serial interface.

(f) PCL

This is a clock output pin.

(g) INTP4

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

2.2.2 P20 to P27 (port 2)

P20 to P27 function as an 8-bit I/O port. These pins also function as pins for segment signal output pins for the LCD controller/driver, 10-bit successive approximation type A/D converter analog input (μPD78F048x and 78F049x only), 16-bit $\Delta\Sigma$ type A/D converter analog input, and reference voltage input (μPD78F049x only). Either I/O port function or segment signal output function can be selected using port function register 2 (PF2).

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

(1) Port mode

P20 to P27 function as an 8-bit I/O port. P20 to P27 can be set to input or output port in 1-bit units using port mode register 2 (PM2).

(2) Control mode

P20 to P27 function as segment signal output for the LCD controller/driver, 10-bit successive approximation type A/D converter analog input (μPD78F048x and 78F049x only), 16-bit $\Delta\Sigma$ type A/D converter analog input (μPD78F049x only), and reference voltage input.

(a) SEG32 to SEG39

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

(b) ANI0 to ANI7 (μPD78F048x and 78F049x only)

These are 10-bit successive approximation type A/D converter analog input pins. When using these pins as analog input pins, see **(5) ANI0/SEG39/P20 to ANI7/SEG32/P27 pins (μPD78F048x), ANI0/DS0−/P20 to ANI7/REF+/P27 pins (μPD78F049x)** in **12.6 Cautions for 10-bit successive approximation type A/D Converter**.

(c) DS0−, DS0+, DS1−, DS1+, DS2−, DS2+, REF−, and REF+ (μPD78F049x only)

These are 16-bit $\Delta\Sigma$ type A/D converter analog input pins, and reference voltage input pins.
Set REF− to the same potential as V_{SS} and AV_{SS}.
Set REF+ to the same potential as AV_{REF}.

Caution P20 to P27 are set in the analog input mode after release of reset.

2.2.3 P30 to P34 (port 3)

P30 to P34 function as a 5-bit I/O port. These pins also function as pins for external interrupt request input, timer I/O, buzzer output, real-time counter output, and manchester code output.

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

(1) Port mode

P30 to P34 function as a 5-bit I/O port. P30 to P34 can be set to input or output port in 1-bit units using port mode register 3 (PM3). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 3 (PU3).

(2) Control mode

P30 to P34 function as external interrupt request input, timer I/O, buzzer output, real-time counter output, and manchester code output.

(a) INTP1 to INTP3 and INTP5

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

(b) TO00, TOH0, TOH1

These are timer output pin.

(c) TI000

This is a pin for inputting an external count clock to 16-bit timer/event counters 00 and is also for inputting a capture trigger signal to the capture registers (CR000 or CR010) of 16-bit timer/event counters 00.

(d) TI010

This is a pin for inputting a capture trigger signal to the capture register (CR000) of 16-bit timer/event counters 00.

(e) TI52

This is the pin for inputting an external count clock to 8-bit timer/event counter 52.

(f) BUZ

This is a buzzer output pin.

(g) RTCDIV

This is a real-time counter clock (32 kHz, divided) output pin.

(h) RTCCL

This is a real-time counter clock (32 kHz, original oscillation) output pin.

(i) RTC1HZ

This is a real-time counter correction clock (1 Hz) output pin.

(j) MCGO

This is a Manchester code output pin.

2.2.4 P40 to P47 (port 4)

P40 to P47 function as an 8-bit I/O port. These pins also function as pins for key interrupt input, segment key scan input, timer I/O, remote control receive data input, and power supply voltage for driving the LCD.

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

(1) Port mode

P40 to P47 function as an 8-bit I/O port. P40 and P47 can be set to input port or output port in 1-bit units using port mode register 4 (PM4). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 4 (PU4).

(2) Control mode

P40 and P47 function as key interrupt input, segment key scan input, timer I/O, remote control receive data input, and power supply voltage for driving the LCD.

(a) **KR0 to KR7**

<R> These are the key interrupt input or segment key scan input pins.

(b) **TO50, TO51**

These are the timer output pin from 8-bit timer/event counter 50 and 51.

(c) **TI50, TI51**

These are the pins for inputting an external count clock to 8-bit timer/event counter 50 and 51.

(d) **RIN**

This is the data input pin of the remote controller receiver.

(e) **V_{LC3}**

This is the power supply voltage pins for driving the LCD.

2.2.5 P80 to P83 (port 8)

P80 to P83 function as a 4-bit I/O port. These pins also function as segment signal output pins for the LCD controller/driver. Either I/O port function or segment signal output function can be selected using port function register ALL (PFALL).

(1) Port mode

P80 to P83 function as a 4-bit I/O port. P80 to P83 can be set to input or output port in 1-bit units using port mode register 8 (PM8). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 8 (PU8).

(2) Control mode

P80 to P83 function as segment signal output for the LCD controller/driver.

(a) **SEG4 to SEG7**

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

2.2.6 P90 to P93 (port 9)

P90 to P93 function as a 4-bit I/O port. These pins also function as segment signal output pins for the LCD controller/driver. Either I/O port function or segment signal output function can be selected using port function register ALL (PFALL).

(1) Port mode

P90 to P93 function as a 4-bit I/O port. P90 to P93 can be set to input or output port in 1-bit units using port mode register 9 (PM9). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 9 (PU9).

(2) Control mode

P90 to P93 function as segment signal output for the LCD controller/driver.

(a) **SEG8 to SEG11**

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

2.2.7 P100 to P103 (port 10)

P100 to P103 function as a 4-bit I/O port. These pins also function as segment signal output pins for the LCD controller/driver. Either I/O port function or segment signal output function can be selected using port function register ALL (PFALL).

(1) Port mode

P100 to P103 function as a 4-bit I/O port. P100 to P103 can be set to input or output port in 1-bit units using port mode register 10 (PM10). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 10 (PU10).

(2) Control mode

P100 to P103 function as segment signal output for the LCD controller/driver.

(a) SEG12 to SEG15

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

2.2.8 P110 to P113 (port 11)

P110 to P113 function as a 4-bit I/O port. These pins also function as pins for segment signal output for the LCD controller/driver and serial interface data I/O. Either I/O port function (other than segment signal output) or segment signal output function can be selected using port function register ALL (PFALL).

(1) Port mode

P110 to P113 function as a 4-bit I/O port. P110 to P113 can be set to input or output port in 1-bit units using port mode register 11 (PM11). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 11 (PU11).

(2) Control mode

P110 to P113 function as segment signal output for the LCD controller/driver and serial interface data I/O.

(a) SEG16 to SEG19

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

(b) RxD6

This is a serial data input pin of serial interface UART6.

(c) TxD6

This is a serial data output pin of serial interface UART6.

2.2.9 P120 to P124 (port 12)

P120 functions as a 1-bit I/O port. P121 to P124 function as a 4-bit input port. These pins also function as pins for external interrupt request input, potential input for external low-voltage detection, resonator for main system clock connection, resonator for subsystem clock connection, and external clock input. The following operation modes can be specified in 1-bit units.

(1) Port mode

P120 functions as a 1-bit I/O port and P121 to P124 function as a 4-bit input port. Only for P120, can be set to input or output port using port mode register 12 (PM12). Only for P120, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

(2) Control mode

P120 to P124 function as an external interrupt request input, potential input for external low-voltage detection, resonator for main system clock connection, resonator for subsystem clock connection, and external clock input.

(a) INTP0

This functions as an external interrupt request input (INTP0) for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

(b) EXLVI

This is a potential input pin for external low-voltage detection.

(c) X1, X2

These are the pins for connecting a resonator for main system clock.

(d) EXCLK

This is an external clock input pin for main system clock.

(e) XT1, XT2

These are the pins for connecting a resonator for subsystem clock.

Remark X1 and X2 can be used as on-chip debug mode setting pins (OCD0A, OCD0B) when the on-chip debug function is used. For detail, see **CHAPTER 29 ON-CHIP DEBUG FUNCTION.**

2.2.10 P130 to P133 (port 13)

P130 to P133 function as a 4-bit I/O port. These pins also function as pins for segment signal output pins for the LCD controller/driver and serial interface data I/O. Either I/O port function or segment signal output function can be selected using port function register ALL (PFALL).

(1) Port mode

P130 to P133 function as a 4-bit I/O port. P130 to P133 can be set to input or output port in 1-bit units using port mode register 13 (PM13). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 13 (PU13).

(2) Control mode

P130 to P133 function as segment signal output for the LCD controller/driver.

(a) SEG20 to SEG23

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

2.2.11 P140 to P143 (port 14)

P140 to P143 function as a 4-bit I/O port. These pins also function as pins for segment signal output and simultaneous output of segment key source signal for the LCD controller/driver. Either I/O port function or segment signal output function can be selected using port function register ALL (PFALL).

(1) Port mode

P140 to P143 function as a 4-bit I/O port. P140 to P143 can be set to input or output port in 1-bit units using port mode register 14 (PM14). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 14 (PU14).

(2) Control mode

P140 to P143 function as segment signal output and simultaneous output of segment key source signal for the LCD controller/driver.

<R> **(a) SEG24 (KS0) to SEG27 (KS3)**

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

The segment key source signal output can be simultaneously used by setting the LCD mode register (LCDMD).

2.2.12 P150 to P153 (port 15)

P150 to P153 function as a 4-bit I/O port. These pins also function as pins for segment signal output and simultaneous output of segment key source signal for the LCD controller/driver. Either I/O port function or segment signal output function can be selected using port function register ALL (PFALL).

(1) Port mode

P150 to P153 function as a 4-bit I/O port. P150 to P153 can be set to input or output port in 1-bit units using port mode register 15 (PM15). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 15 (PU15).

(2) Control mode

P150 to P153 function as segment signal output and simultaneous output of segment key source signal for the LCD controller/driver.

<R> **(a) SEG28 (KS4) to SEG31 (KS7)**

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

The segment key source signal output can be simultaneously used by setting the LCD mode register (LCDMD).

2.2.13 AV_{REF} (μPD78F048x and 78F049x only)

This is the 10-bit successive approximation type A/D converter reference voltage input pin and the positive power supply pin of port 2 and 16-bit $\Delta\Sigma$ type A/D converter.

When the A/D converter is not used, connect this pin directly to V_{DD} Note.

Note When one or more of the pins of port 2 is used as the digital port pins or for segment output, make AV_{REF} the same potential as V_{DD}.

2.2.14 AV_{SS} (μPD78F048x and 78F049x only)

This is the A/D converter ground potential pin. Even when the A/D converter is not used, always use this pin with the same potential as the V_{SS} pin.

2.2.15 COM0 to COM7

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

2.2.16 V_{LC0} to V_{LC3}

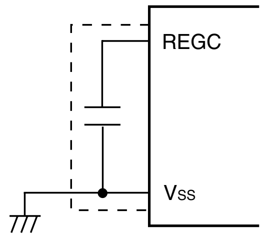
These pins are the power supply voltage pins for driving the LCD.

2.2.17 $\overline{\text{RESET}}$

This is the active-low system reset input pin.

2.2.18 REGC

This is the pin for connecting regulator output (2.4 V) stabilization capacitance for internal operation. Connect this pin to V_{SS} via a capacitor (0.47 to 1 μF: recommended).



Caution Keep the wiring length as short as possible in the area enclosed by the broken lines in the above figures.

2.2.19 V_{DD}

This is the positive power supply pin.

2.2.20 V_{SS}

This is the ground potential pin.

2.2.21 FLMD0

This is a pin for setting flash memory programming mode.

Connect FLMD0 to V_{SS} in the normal operation mode.

In flash memory programming mode, connect this pin to the flash memory programmer.

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

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

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

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P10/PCL	5-AG	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P11/$\overline{\text{SCK10}}$	5-AH		
P12/SI10/RxD0			
P13/SO10/TxD0			
P14/$\overline{\text{SCKA0}}$/INTP4			
P15/SIA0/<RxD6>			
P16/SOA0/<TxD6>	5-AG		
P17			
P20/SEG39/ANI0/DS0- to P27/SEG32/ANI7/REF+ Notes 1, 2, 3, 4	17-R		<Analog setting> Connect to AV_{REF} or AV_{SS}. <Digital setting> Input: Independently connect to AV_{REF} or AV_{SS} via a resistor.Note 5 Output: Leave open. <Segment setting> Leave open.
P30/INTP5	5-AH		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P31/TOH1/INTP3			
P32/TOH0/MCGO	5-AG		
P33/TI000/RTCDIV/ RTCCL/BUZ/INTP2	5-AH		
P34/TI52/TI010/TO00/ RTC1HZ/INTP1			
P40/V_{LC3}/KR0	5-AO		
P41/RIN/KR1	5-AH		
P42/KR2			
P43/TO51/TI51/KR3			
P44/TO50/TI50/KR4			
P45/KR5 to P47/KR7			
P80/SEG4 to P83/SEG7	17-P		<Port setting> Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open. <Segment setting> Leave open.
P90/SEG8 to P93/SEG11			
P100/SEG12 to P103/SEG15			

Notes 1. SEGx is provided to the μPD78F047x and 78F048x only.
2. ANIx is provided to the μPD78F048x and 78F049x only.
3. DSx and REFx are provided to the 78F049x only.
4. P20/SEG39/ANI0/DS0- to P27/SEG32/ANI7/REF+ are set in the digital input mode after release of reset.
5. With μPD78F047x, independently connect to V_{DD} or V_{SS} via a resistor.

Remark The functions within arrowheads (< >) can be assigned by setting the input switch control register (ISC).

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

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P110/SEG16, P111/SEG17	17-P	I/O	<Port setting> Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open. <Segment setting> Leave open.
P112/SEG18/TxD6			
P113/SEG19/RxD6	17-Q		
P120/INTP0/EXLVI	5-AH		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P121/X1/OCD0A[Note 1]	37-A	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P122/X2/EXCLK/OCD0B[Note 1]			
P123/XT1[Note 1]			
P124/XT2[Note 1]			
P130/SEG20 to P133/SEG23	17-P	I/O	<Port setting> Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open. <Segment setting> Leave open.
P140/SEG24 (KS0) to P143/SEG27 (KS3)			
P150/SEG28 (KS4) to P153/SEG31 (KS7)			
COM0 to COM3	18-E	Output	Leave open.
COM4/SEG0 to COM7/SEG3	18-F		
V_{LC0} to V_{LC2}	–	–	
$\overline{RESET}$	2	Input	Connect directly or via a resistor to V_{DD}.
FLMD0	38		Connect to V_{SS}.[Note 3]
AV_{REF}[Note 2]	–	–	Connect directly to V_{DD}.[Note 4]
AV_{SS}[Note 2]			Connect directly to V_{SS}.

Notes 1. Use recommended connection above in I/O port mode (see **Figure 5-2 Format of Clock Operation Mode Select Register (OSCCTL)**) when these pins are not used.

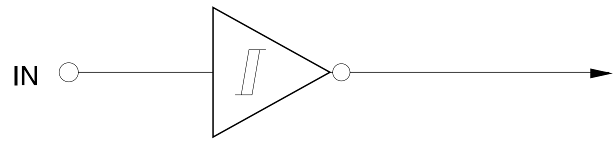
2. μPD78F048x and 78F049x only.

3. FLMD0 is a pin used when writing data to flash memory. When rewriting flash memory data on-board or performing on-chip debugging, connect this pin to V_{SS} via a resistor (10 kΩ: recommended).

4. When using port 2 as a digital port or for segment output, set it to the same potential as that of V_{DD}.

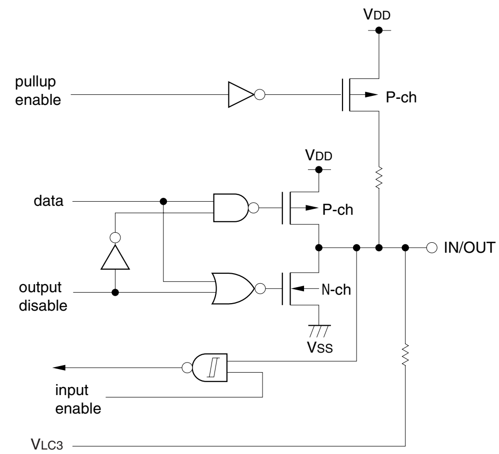
Figure 2-1. Pin I/O Circuit List (1/2)

Type 2

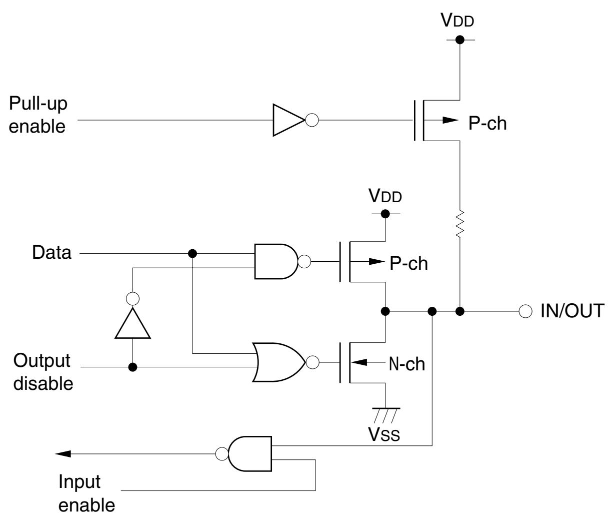


Schmitt-triggered input with hysteresis characteristics

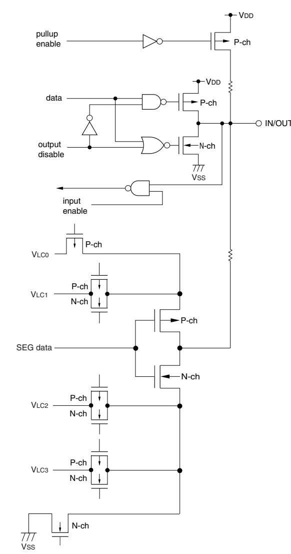
Type 5-AO



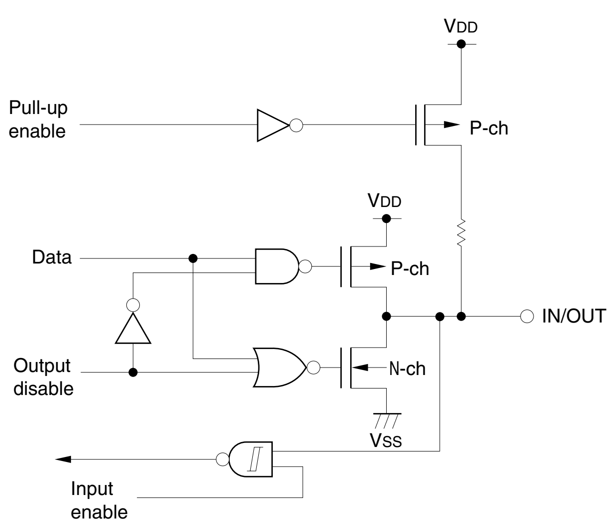
Type 5-AG



Type 17-P



Type 5-AH



Type 17-Q

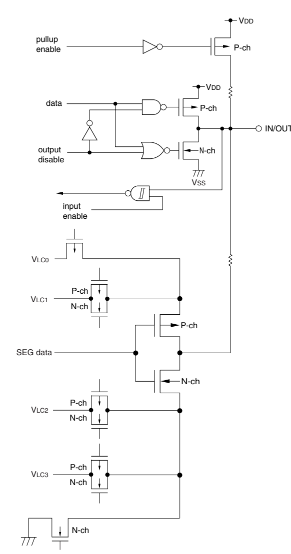
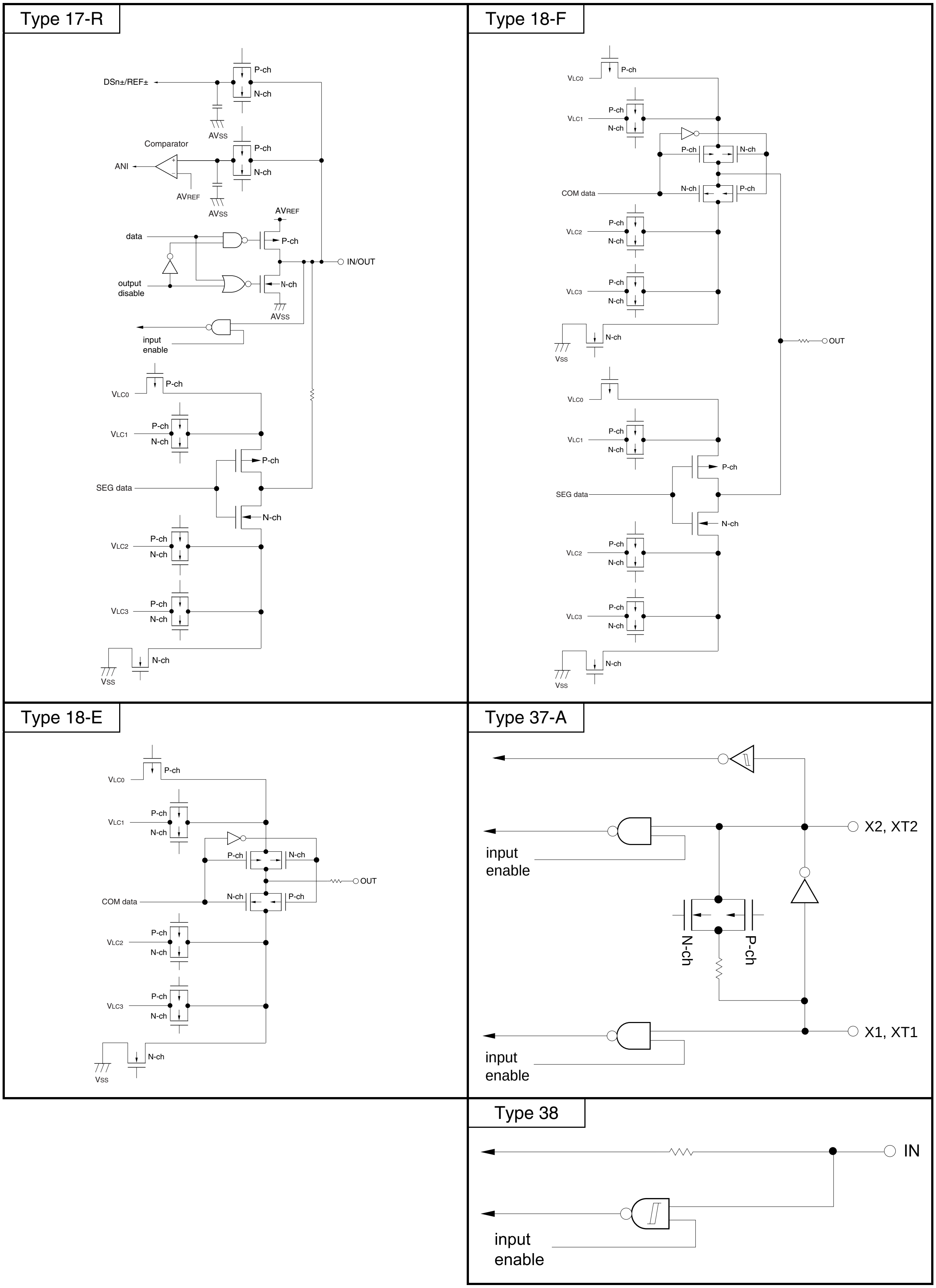


Figure 2-1. Pin I/O Circuit List (2/2)



3.1 Memory Space

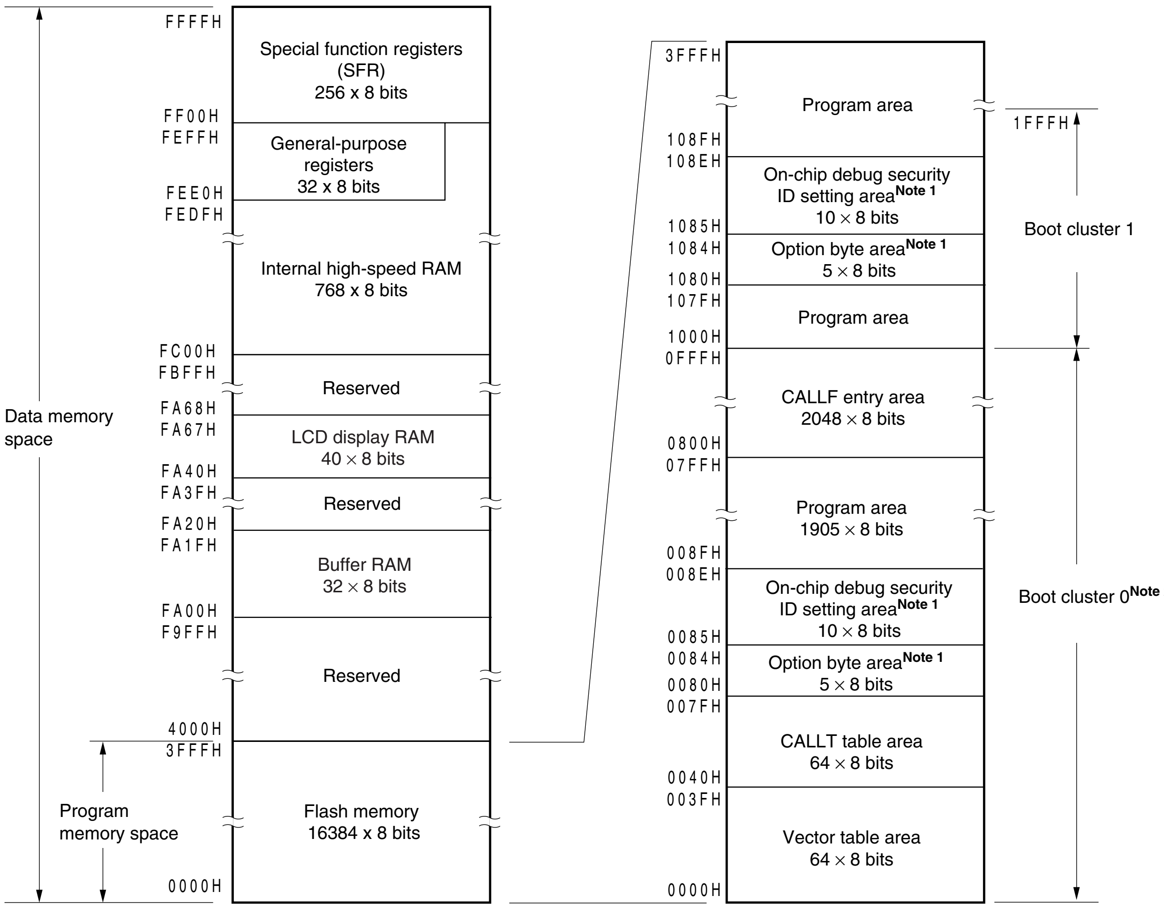
Each products in the 78K0/LF3 can access a 64 KB memory space. Figures 3-1 to 3-10 show the memory maps.

Caution Regardless of the internal memory capacity, the initial values of the internal memory size switching register (IMS) and internal expansion RAM size switching register (IXS) of all products in the 78K0/LF3 are fixed (IMS = CFH, IXS = 0CH). Therefore, set the value corresponding to each product as indicated below.

Table 3-1. Set Values of Internal Memory Size Switching Register (IMS) and Internal Expansion RAM Size Switching Register (IXS)

Flash Memory Version (78K0/LF3)	IMS	IXS	ROM Capacity	Internal High-Speed RAM Capacity	Internal Expansion RAM Capacity
μPD78F0471, 78F0481, 78F0491	04H	0CH	16 KB	768 bytes	−
μPD78F0472, 78F0482, 78F0492	C6H		24 KB	1 KB	
μPD78F0473, 78F0483, 78F0493	C8H		32 KB		
μPD78F0474, 78F0484, 78F0494	CCH	0AH	48 KB		1 KB
μPD78F0475, 78F0485, 78F0495	CFH		60 KB		

Figure 3-1. Memory Map (μPD78F0471, 78F0481)



Notes 1. When boot swap is not used: Set the option bytes to 0080H to 0084H, and the on-chip debug security IDs to 0085H to 008EH.

When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H, and the on-chip debug security IDs to 0085H to 008EH and 1085H to 108EH.

2. Writing boot cluster 0 can be prohibited depending on the setting of security (see **28.8 Security Setting**).

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 3-2 Correspondence Between Address Values and Block Numbers in Flash Memory**.

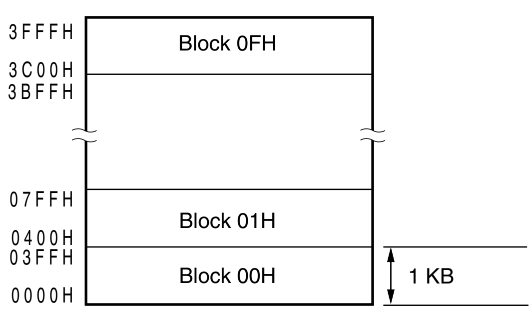
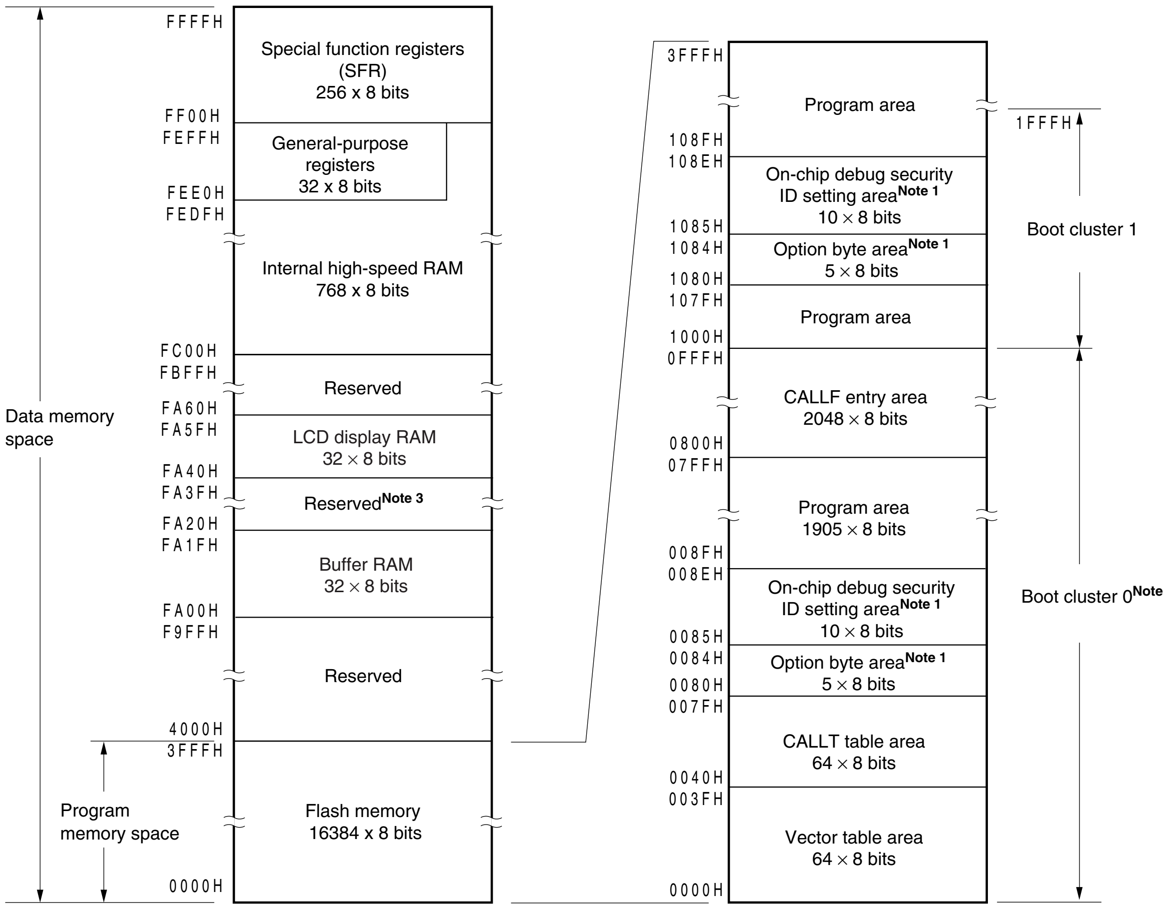


Figure 3-2. Memory Map (μPD78F0491)



Notes 1. When boot swap is not used: Set the option bytes to 0080H to 0084H, and the on-chip debug security IDs to 0085H to 008EH.

When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H, and the on-chip debug security IDs to 0085H to 008EH and 1085H to 108EH.

2. Writing boot cluster 0 can be prohibited depending on the setting of security (see **28.8 Security Setting**).

<R> **3.** However, FA26H and FA27H can be used (See **13.3 Registers Used in 16-Bit ΔΣ Type A/D Converter**).

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 3-2 Correspondence Between Address Values and Block Numbers in Flash Memory**.

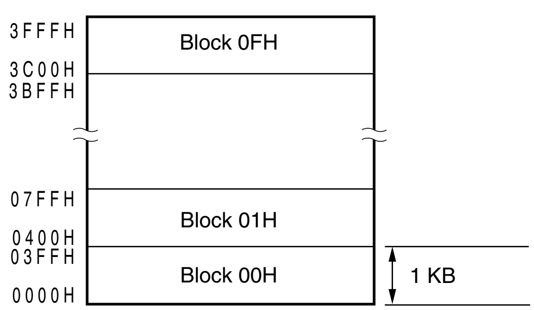
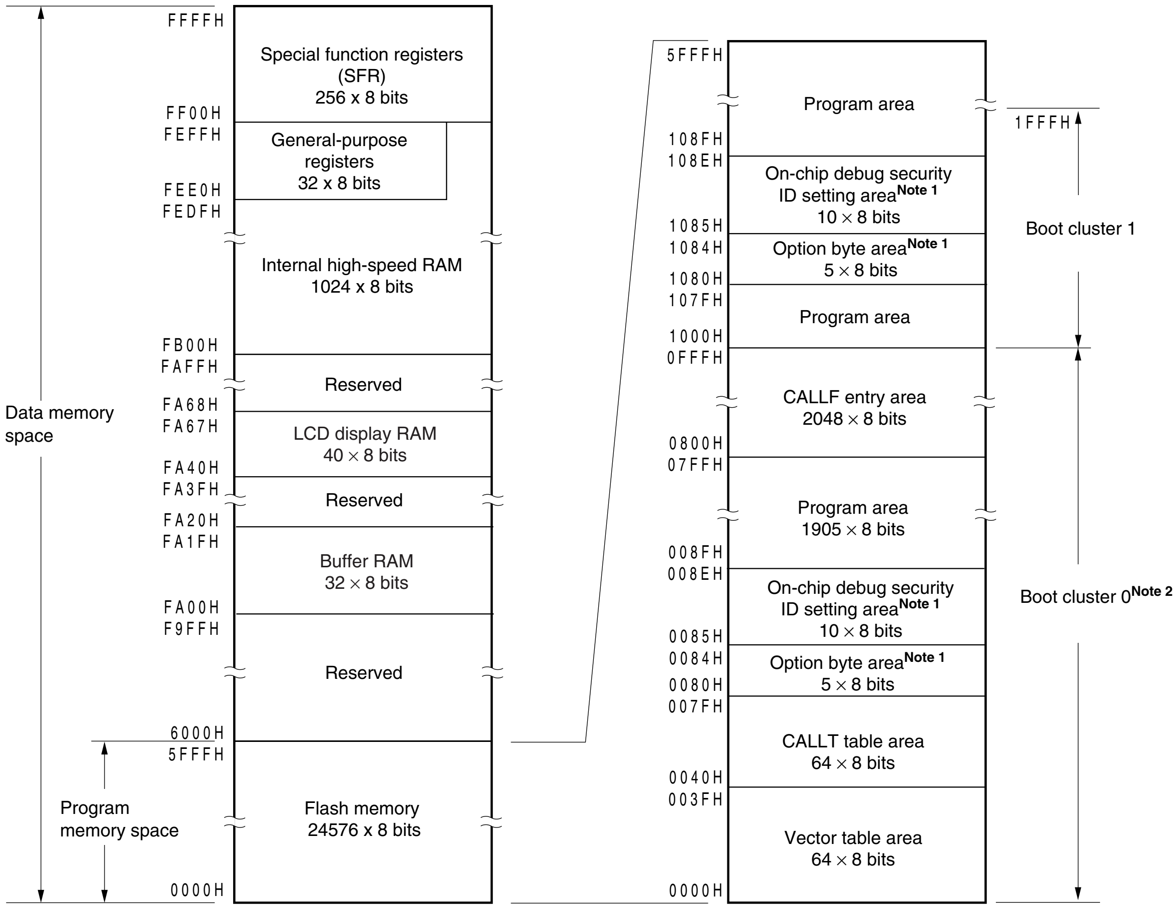


Figure 3-3. Memory Map (μPD78F0472, 78F0482)



Notes 1. When boot swap is not used: Set the option bytes to 0080H to 0084H, and the on-chip debug security IDs to 0085H to 008EH.

When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H, and the on-chip debug security IDs to 0085H to 008EH and 1085H to 108EH.

2. Writing boot cluster 0 can be prohibited depending on the setting of security (see **28.8 Security Setting**).

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 3-2 Correspondence Between Address Values and Block Numbers in Flash Memory**.

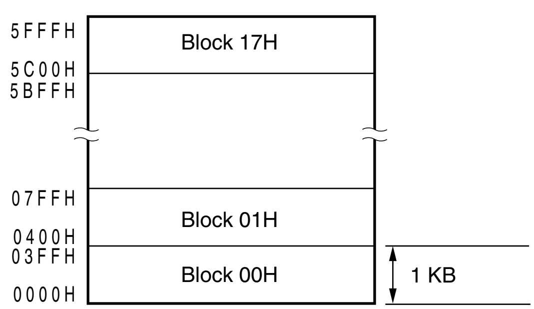
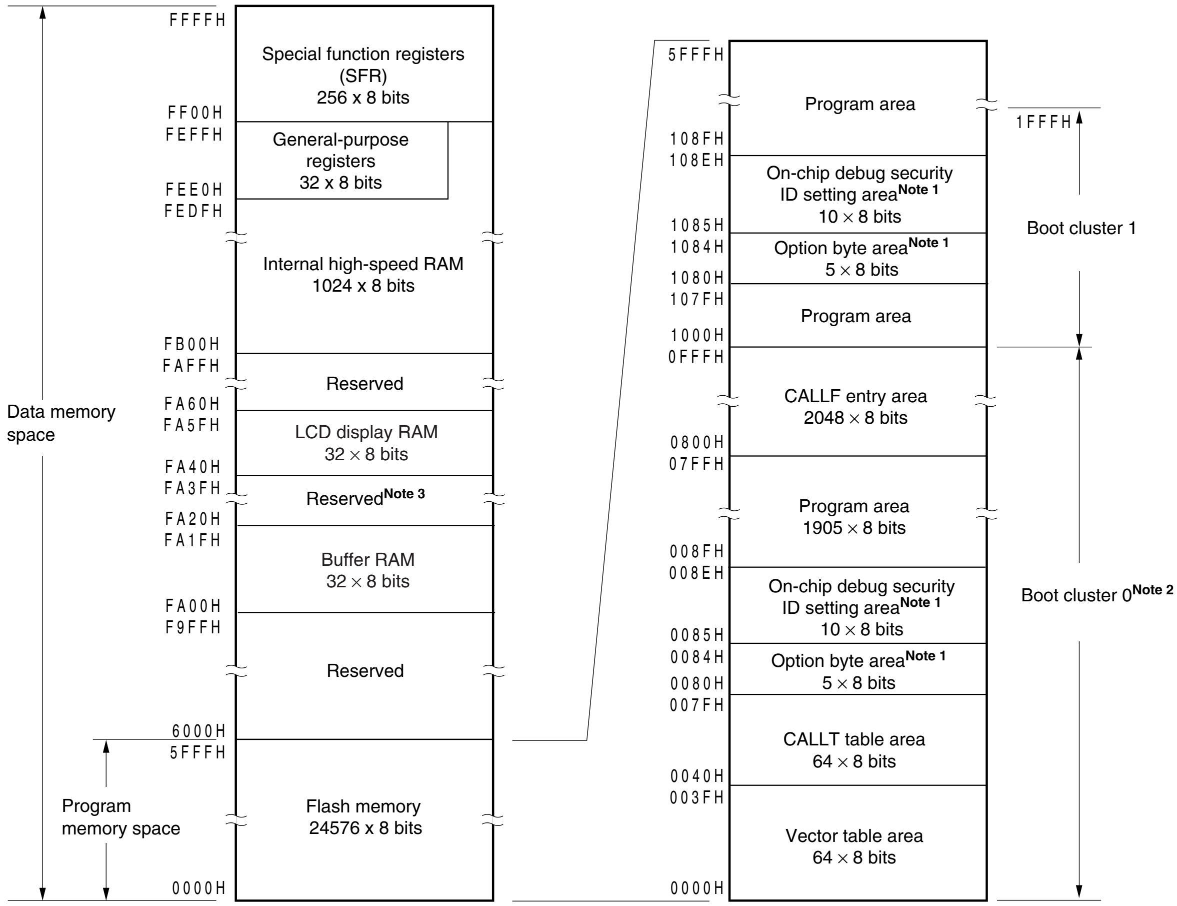


Figure 3-4. Memory Map (μPD78F0492)



Notes 1. When boot swap is not used: Set the option bytes to 0080H to 0084H, and the on-chip debug security IDs to 0085H to 008EH.

When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H, and the on-chip debug security IDs to 0085H to 008EH and 1085H to 108EH.

2. Writing boot cluster 0 can be prohibited depending on the setting of security (see **28.8 Security Setting**).

<R> **3.** However, FA26H and FA27H can be used (See **13.3 Registers Used in 16-Bit ΔΣ Type A/D Converter**).

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 3-2 Correspondence Between Address Values and Block Numbers in Flash Memory**.

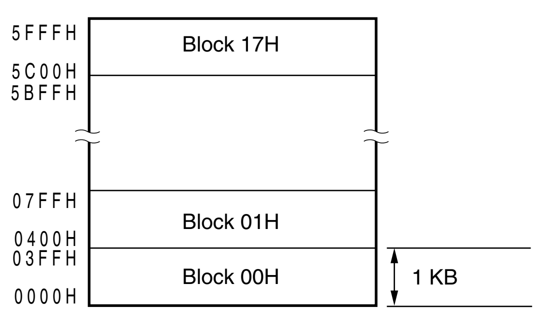
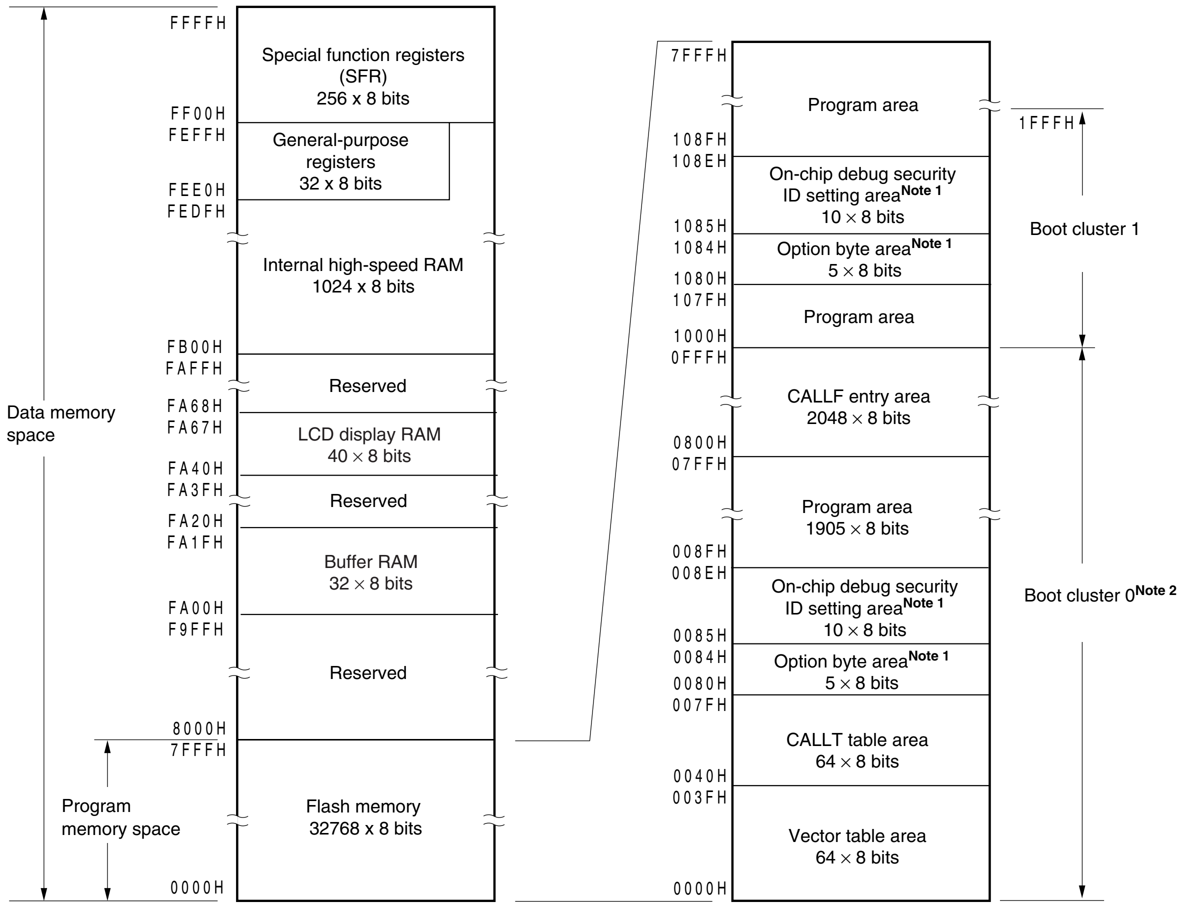


Figure 3-5. Memory Map (μPD78F0473, 78F0483)



Notes 1. When boot swap is not used: Set the option bytes to 0080H to 0084H, and the on-chip debug security IDs to 0085H to 008EH.

When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H, and the on-chip debug security IDs to 0085H to 008EH and 1085H to 108EH.

2. Writing boot cluster 0 can be prohibited depending on the setting of security (see **28.8 Security Setting**).

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 3-2 Correspondence Between Address Values and Block Numbers in Flash Memory**.

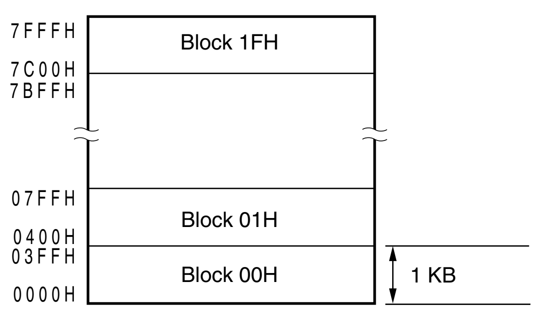
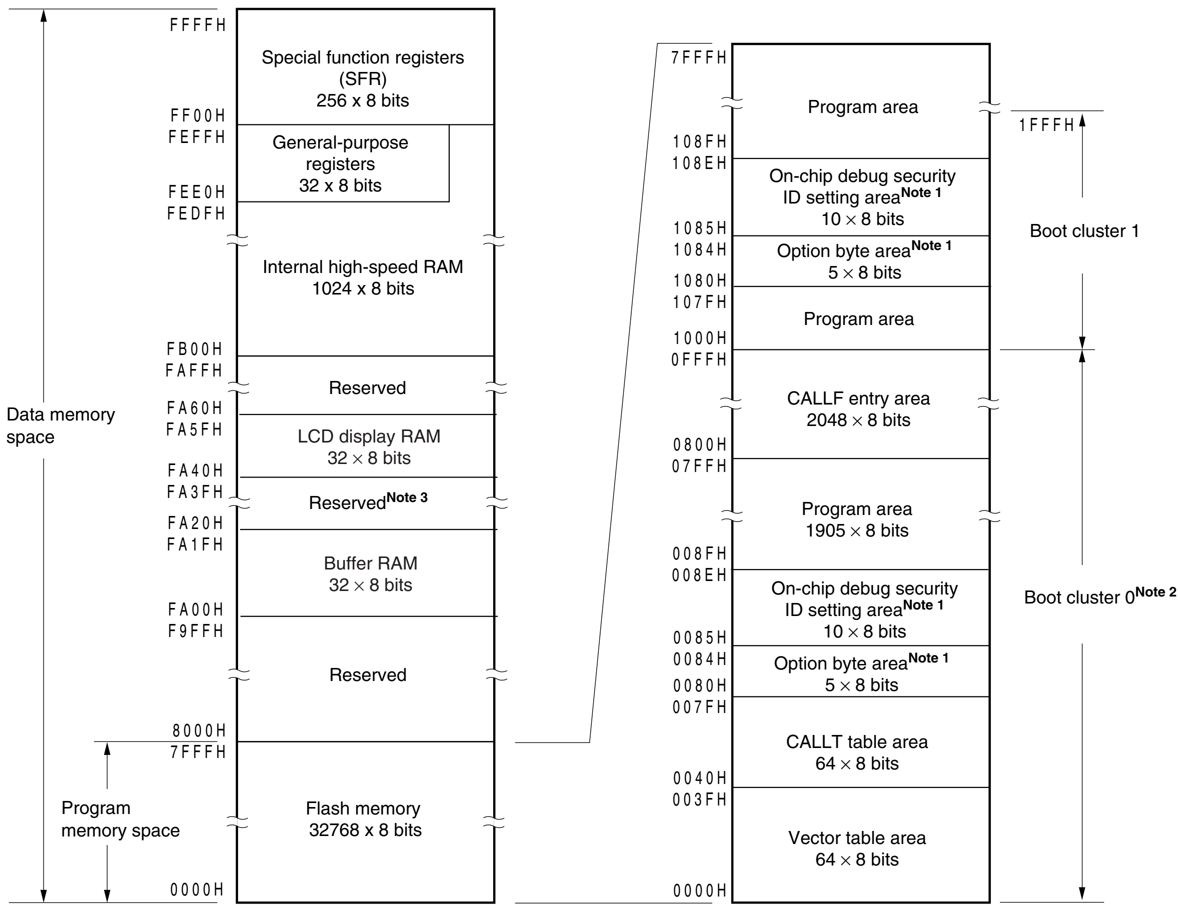


Figure 3-6. Memory Map (μPD78F0493)



Notes 1. When boot swap is not used: Set the option bytes to 0080H to 0084H, and the on-chip debug security IDs to 0085H to 008EH.

When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H, and the on-chip debug security IDs to 0085H to 008EH and 1085H to 108EH.

2. Writing boot cluster 0 can be prohibited depending on the setting of security (see **28.8 Security Setting**).

<R> **3.** However, FA26H and FA27H can be used (See **13.3 Registers Used in 16-Bit ΔΣ Type A/D Converter**).

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 3-2 Correspondence Between Address Values and Block Numbers in Flash Memory**.

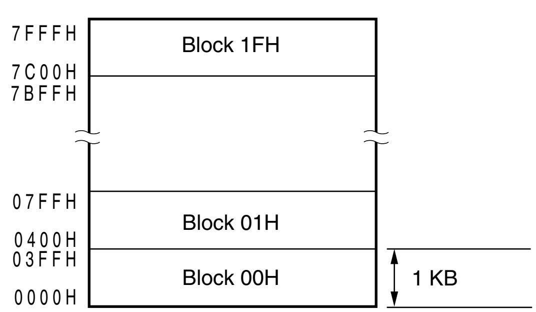
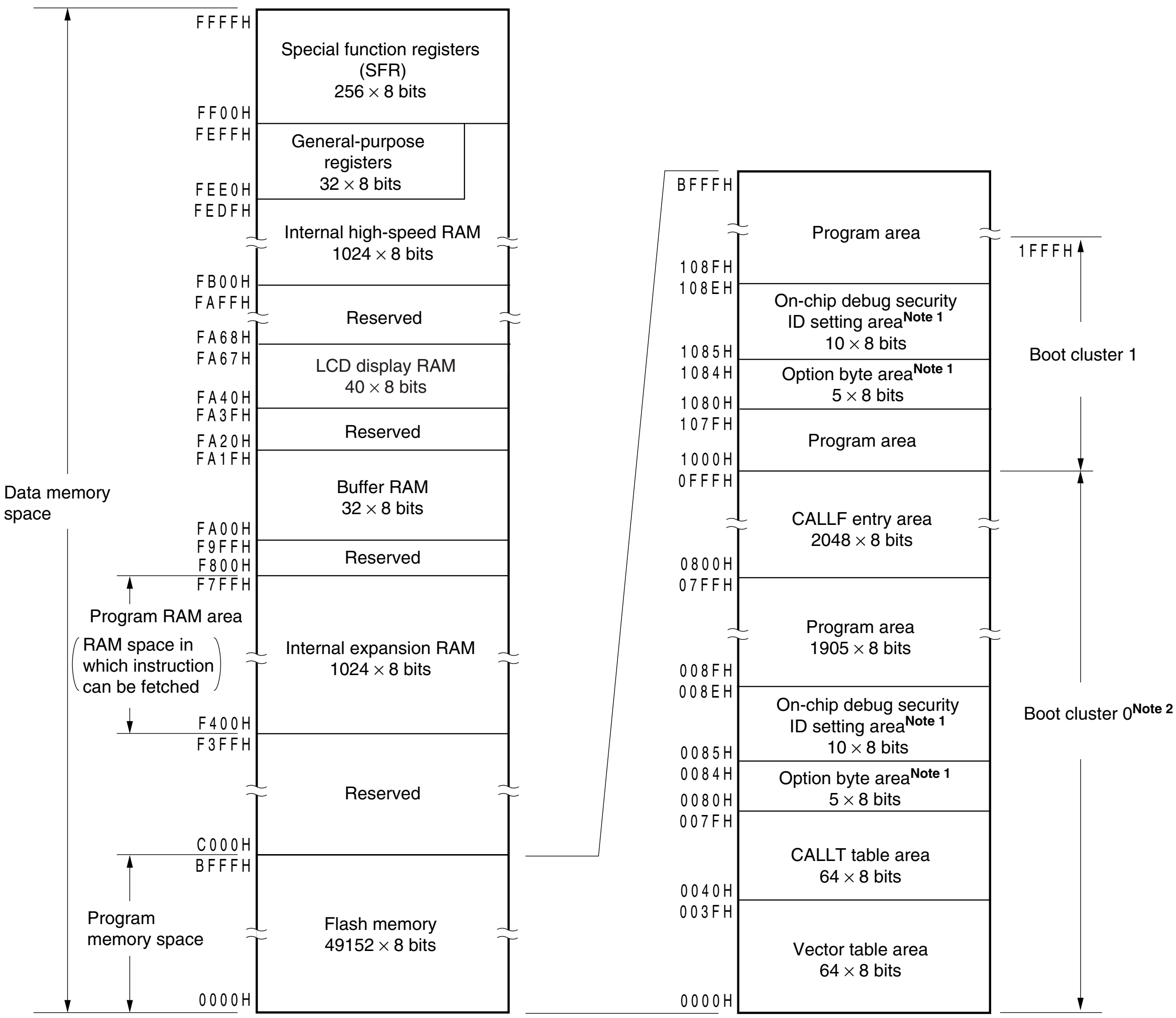


Figure 3-7. Memory Map (μPD78F0474, 78F0484)



Notes 1. When boot swap is not used: Set the option bytes to 0080H to 0084H, and the on-chip debug security IDs to 0085H to 008EH.

When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H, and the on-chip debug security IDs to 0085H to 008EH and 1085H to 108EH.

2. Writing boot cluster 0 can be prohibited depending on the setting of security (see **28.8 Security Setting**).

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 3-2 Correspondence Between Address Values and Block Numbers in Flash Memory**.

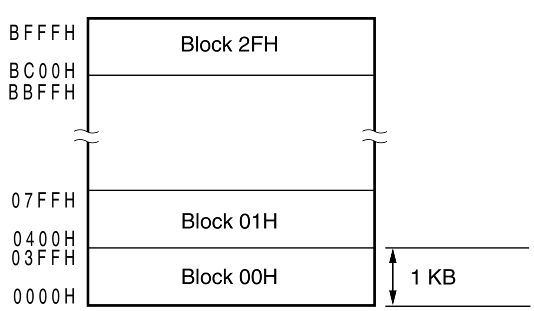
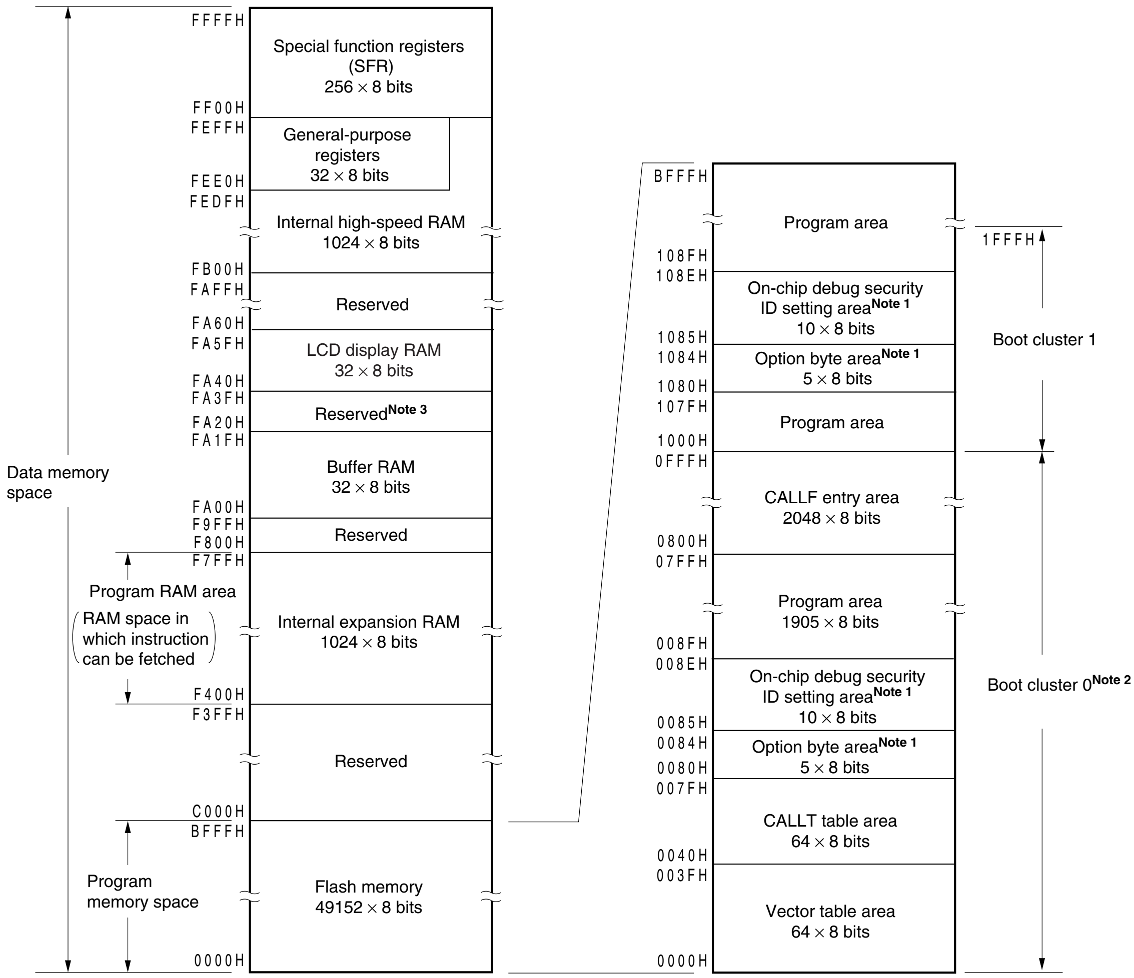


Figure 3-8. Memory Map (μPD78F0494)



Notes 1. When boot swap is not used: Set the option bytes to 0080H to 0084H, and the on-chip debug security IDs to 0085H to 008EH.

When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H, and the on-chip debug security IDs to 0085H to 008EH and 1085H to 108EH.

2. Writing boot cluster 0 can be prohibited depending on the setting of security (see **28.8 Security Setting**).

<R> **3.** However, FA26H and FA27H can be used (See **13.3 Registers Used in 16-Bit ΔΣ Type A/D Converter**).

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 3-2 Correspondence Between Address Values and Block Numbers in Flash Memory**.

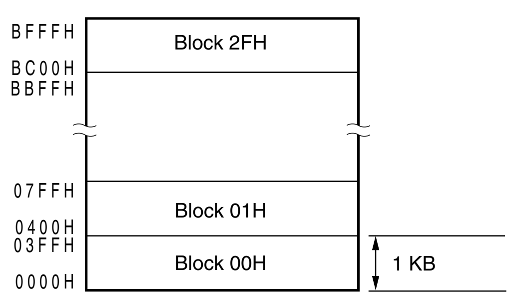
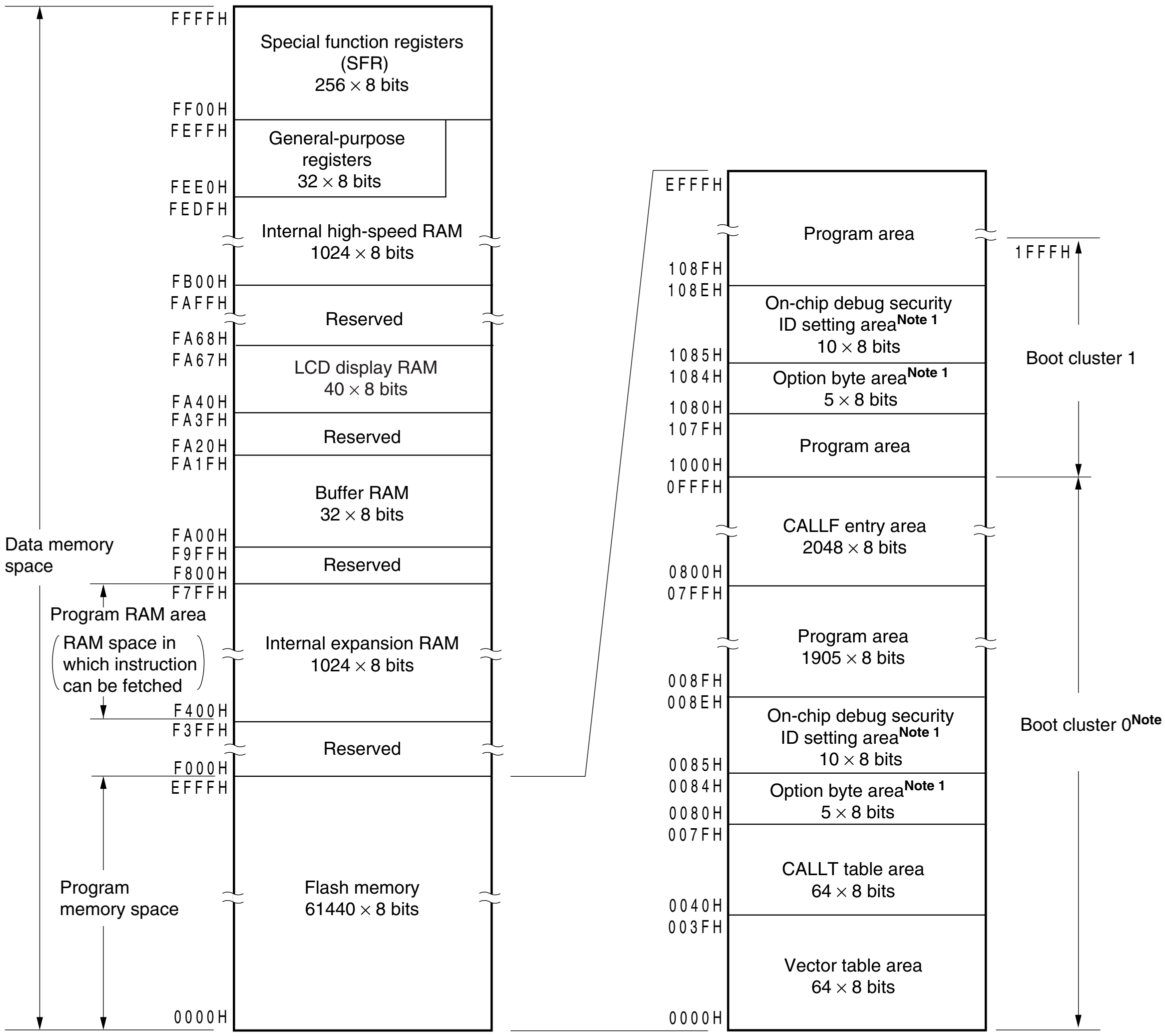


Figure 3-9. Memory Map (μPD78F0475, 78F0485)



Notes 1. When boot swap is not used: Set the option bytes to 0080H to 0084H, and the on-chip debug security IDs to 0085H to 008EH.

When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H, and the on-chip debug security IDs to 0085H to 008EH and 1085H to 108EH.

2. Writing boot cluster 0 can be prohibited depending on the setting of security (see **28.8 Security Setting**).

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 3-2 Correspondence Between Address Values and Block Numbers in Flash Memory**.

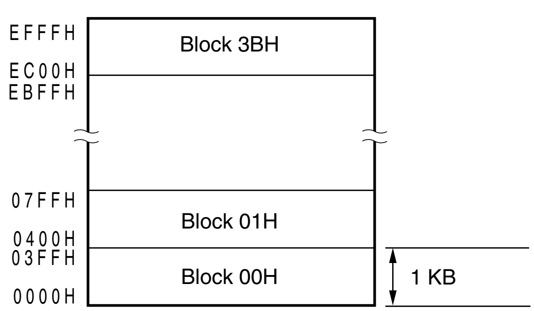
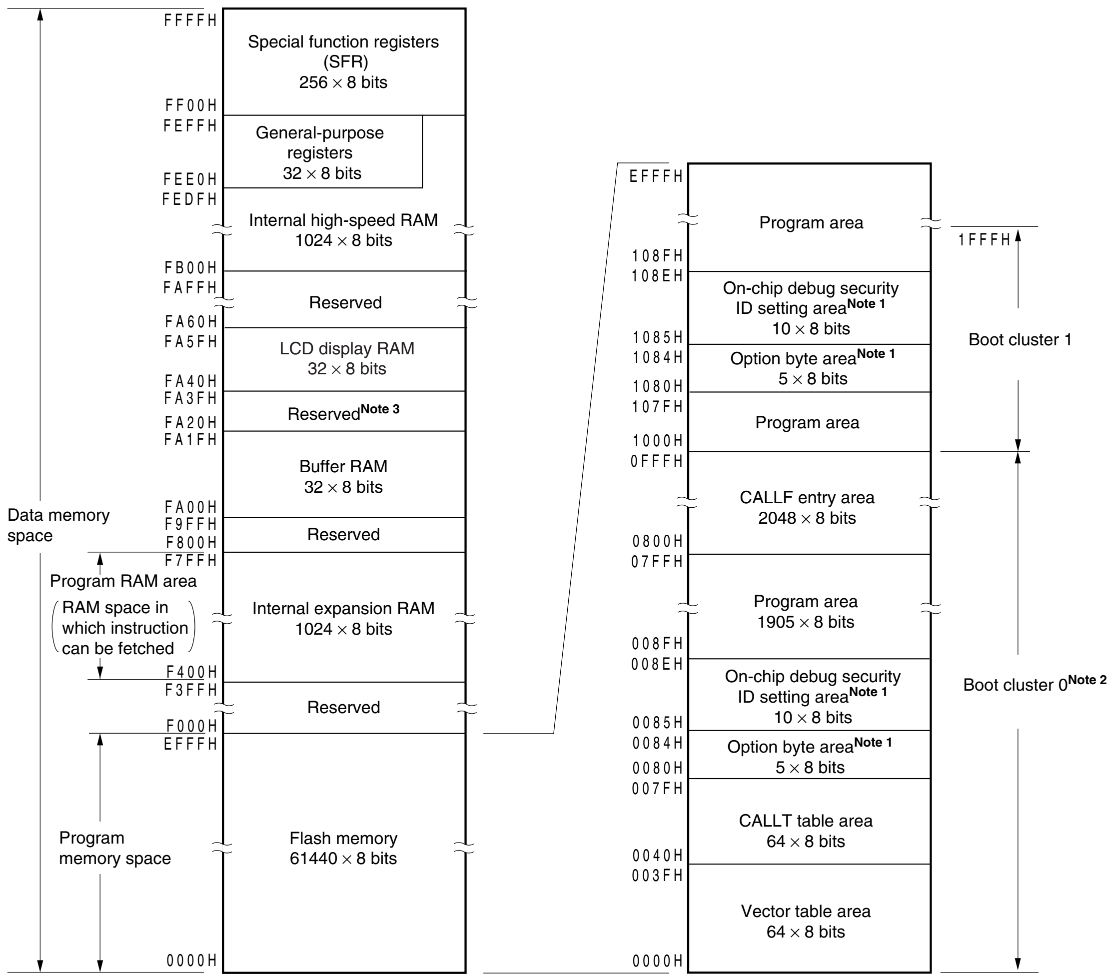


Figure 3-10. Memory Map (μPD78F0495)



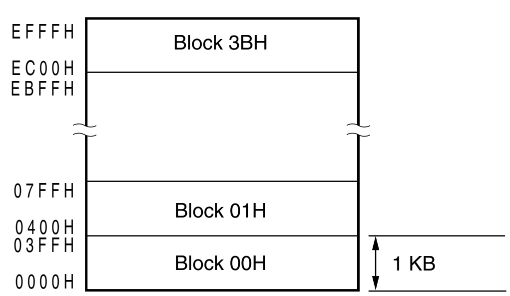
Notes 1. When boot swap is not used: Set the option bytes to 0080H to 0084H, and the on-chip debug security IDs to 0085H to 008EH.

When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H, and the on-chip debug security IDs to 0085H to 008EH and 1085H to 108EH.

2. Writing boot cluster 0 can be prohibited depending on the setting of security (see **28.8 Security Setting**).

<R> **3.** However, FA26H and FA27H can be used (See **13.3 Registers Used in 16-Bit ΔΣ Type A/D Converter**).

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 3-2 Correspondence Between Address Values and Block Numbers in Flash Memory**.



Correspondence between the address values and block numbers in the flash memory are shown below.

Table 3-2. Correspondence Between Address Values and Block Numbers in Flash Memory

Address Value	Block Number	Address Value	Block Number	Address Value	Block Number	Address Value	Block Number
0000H to 03FFH	00H	4000H to 43FFH	10H	8000H to 83FFH	20H	C000H to C3FFH	30H
0400H to 07FFH	01H	4400H to 47FFH	11H	8400H to 87FFH	21H	C400H to C7FFH	31H
0800H to 0BFFH	02H	4800H to 4BFFH	12H	8800H to 8BFFH	22H	C800H to CBFFH	32H
0C00H to 0FFFH	03H	4C00H to 4FFFH	13H	8C00H to 8FFFH	23H	CC00H to CFFFH	33H
1000H to 13FFH	04H	5000H to 53FFH	14H	9000H to 93FFH	24H	D000H to D3FFH	34H
1400H to 17FFH	05H	5400H to 57FFH	15H	9400H to 97FFH	25H	D400H to D7FFH	35H
1800H to 1BFFH	06H	5800H to 5BFFH	16H	9800H to 9BFFH	26H	D800H to DBFFH	36H
1C00H to 1FFFH	07H	5C00H to 5FFFH	17H	9C00H to 9FFFH	27H	DC00H to DFFFH	37H
2000H to 23FFH	08H	6000H to 63FFH	18H	A000H to A3FFH	28H	E000H to E3FFH	38H
2400H to 27FFH	09H	6400H to 67FFH	19H	A400H to A7FFH	29H	E400H to E7FFH	39H
2800H to 2BFFH	0AH	6800H to 6BFFH	1AH	A800H to ABFFH	2AH	E800H to EBFFH	3AH
2C00H to 2FFFH	0BH	6C00H to 6FFFH	1BH	AC00H to AFFFH	2BH	EC00H to EFFFH	3BH
3000H to 33FFH	0CH	7000H to 73FFH	1CH	B000H to B3FFH	2CH		
3400H to 37FFH	0DH	7400H to 77FFH	1DH	B400H to B7FFH	2DH		
3800H to 3BFFH	0EH	7800H to 7BFFH	1EH	B800H to BBFFH	2EH		
3C00H to 3FFFH	0FH	7C00H to 7FFFH	1FH	BC00H to BFFFH	2FH		

Remark μPD78F0471, 78F0481, 78F0491: Block numbers 00H to 0FH
μPD78F0472, 78F0482, 78F0492: Block numbers 00H to 17H
μPD78F0473, 78F0483, 78F0493: Block numbers 00H to 1FH
μPD78F0474, 78F0484, 78F0494: Block numbers 00H to 2FH
μPD78F0475, 78F0485, 78F0495: Block numbers 00H to 3BH

3.1.1 Internal program memory space

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

78K0/LF3 products incorporate internal ROM (flash memory), as shown below.

Table 3-3. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD78F0471, 78F0481, 78F0491	Flash memory	16384 × 8 bits (0000H to 3FFFH)
μPD78F0472, 78F0482, 78F0492		24576 × 8 bits (0000H to 5FFFH)
μPD78F0473, 78F0483, 78F0493		32768 × 8 bits (0000H to 7FFFH)
μPD78F0474, 78F0484, 78F0494		49152 × 8 bits (0000H to BFFFH)
μPD78F0475, 78F0485, 78F0495		61440 × 8 bits (0000H to EFFFH)

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

(1) Vector table area

The 64-byte area 0000H to 003FH is reserved as a vector table area. The program start addresses for branch upon reset or generation of each interrupt request are stored in the vector table area.

Of the 16-bit address, the lower 8 bits are stored at even addresses and the higher 8 bits are stored at odd addresses.

Table 3-4. Vector Table

Vector Table Address	Interrupt Source	Vector Table Address	Interrupt Source
0000H	$\overline{\text{RESET}}$ input, POC, LVI, WDT	0022H	INTTM010
0004H	INTLVI	0024H[Note 1]	INTAD[Note 1]
0006H	INTP0	0026H	INTSR0
0008H	INTP1	0028H	INTRTC
000AH	INTP2	002AH	INTTM51
000CH	INTP3	002CH	INTKR
000EH	INTP4	002EH	INTRTCI
0010H	INTP5	0030H[Note 2]	INTDSAD[Note 2]
0012H	INTSRE6	0032H	INTTM52
0014H	INTSR6	0034H	INTTMH2
0016H	INTST6	0036H	INTMCG
0018H	INTCSI10/INTST0	0038H	INTRIN
001AH	INTTMH1	003AH	INTRERR/INTGP/INTREND /INTDFULL
001CH	INTTMH0	003CH	INTACSI
001EH	INTTM50	003EH	BRK
0020H	INTTM000		

Notes 1. μPD78F048x and 78F049x only.

2. μPD78F049x only.

(2) CALLT instruction table area

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

(3) Option byte area

A 5-byte area of 0080H to 0084H and 1080H to 1084H can be used as an option byte area. Set the option byte at 0080H to 0084H when the boot swap is not used, and at 0080H to 0084H and 1080H to 1084H when the boot swap is used. For details, see **CHAPTER 27 OPTION BYTE**.

(4) CALLF instruction entry area

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

(5) On-chip debug security ID setting area

A 10-byte area of 0085H to 008EH and 1085H to 108EH can be used as an on-chip debug security ID setting area. Set the on-chip debug security ID of 10 bytes at 0085H to 008EH when the boot swap is not used and at 0085H to 008EH and 1085H to 108EH when the boot swap is used. For details, see **CHAPTER 29 ON-CHIP DEBUG FUNCTION**.

3.1.2 Internal data memory space

78K0/LF3 products incorporate the following RAMs.

(1) Internal high-speed RAM

Table 3-5. Internal High-Speed RAM Capacity

Part Number	Internal High-Speed RAM
μPD78F0471, 78F0481, 78F0491	768 × 8 bits (FC00H to FEFFH)
μPD78F0472, 78F0482, 78F0492	1024 × 8 bits (FB00H to FEFFH)
μPD78F0473, 78F0483, 78F0493	
μPD78F0474, 78F0484, 78F0494	
μPD78F0475, 78F0485, 78F0495	

This area cannot be used as a program area in which instructions are written and executed.
The internal high-speed RAM can also be used as a stack memory.

(2) Internal expansion RAM

Table 3-6. Internal Expansion RAM Capacity

Part Number	Internal Expansion RAM
μPD78F0471, 78F0481, 78F0491	–
μPD78F0472, 78F0482, 78F0492	
μPD78F0473, 78F0483, 78F0493	
μPD78F0474, 78F0484, 78F0494	1024 × 8 bits (F400H to F7FFH)
μPD78F0475, 78F0485, 78F0495	

The internal expansion RAM can also be used as a normal data area similar to the internal high-speed RAM, as well as a program area in which instructions can be written and executed.
The internal expansion RAM cannot be used as a stack memory.

(3) LCD display RAM

LCD display RAM is incorporated in the LCD controller/driver (see Figure 18-5 LCD Display RAM).

Table 3-7. LCD Display RAM Capacity

Part Number	Internal Expansion RAM
μPD78F047x, 78F048x	40 × 8 bits (FA40H to FA67H)
μPD78F049x	32 × 8 bits (FA40H to FA5FH)

3.1.3 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area FF00H to FFFFH (see **Table 3-8 Special Function Register List** in **3.2.3 Special function registers (SFRs)**).

Caution Do not access addresses to which SFRs are not assigned.

3.1.4 Data memory addressing

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

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the 78K0/LF3, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of special function registers (SFR) and general-purpose registers are available for use. Figures 3-11 to 3-20 show correspondence between data memory and addressing. For details of each addressing mode, see **3.4 Operand Address Addressing**.

Figure 3-11. Correspondence Between Data Memory and Addressing (μPD78F0471, 78F0481)

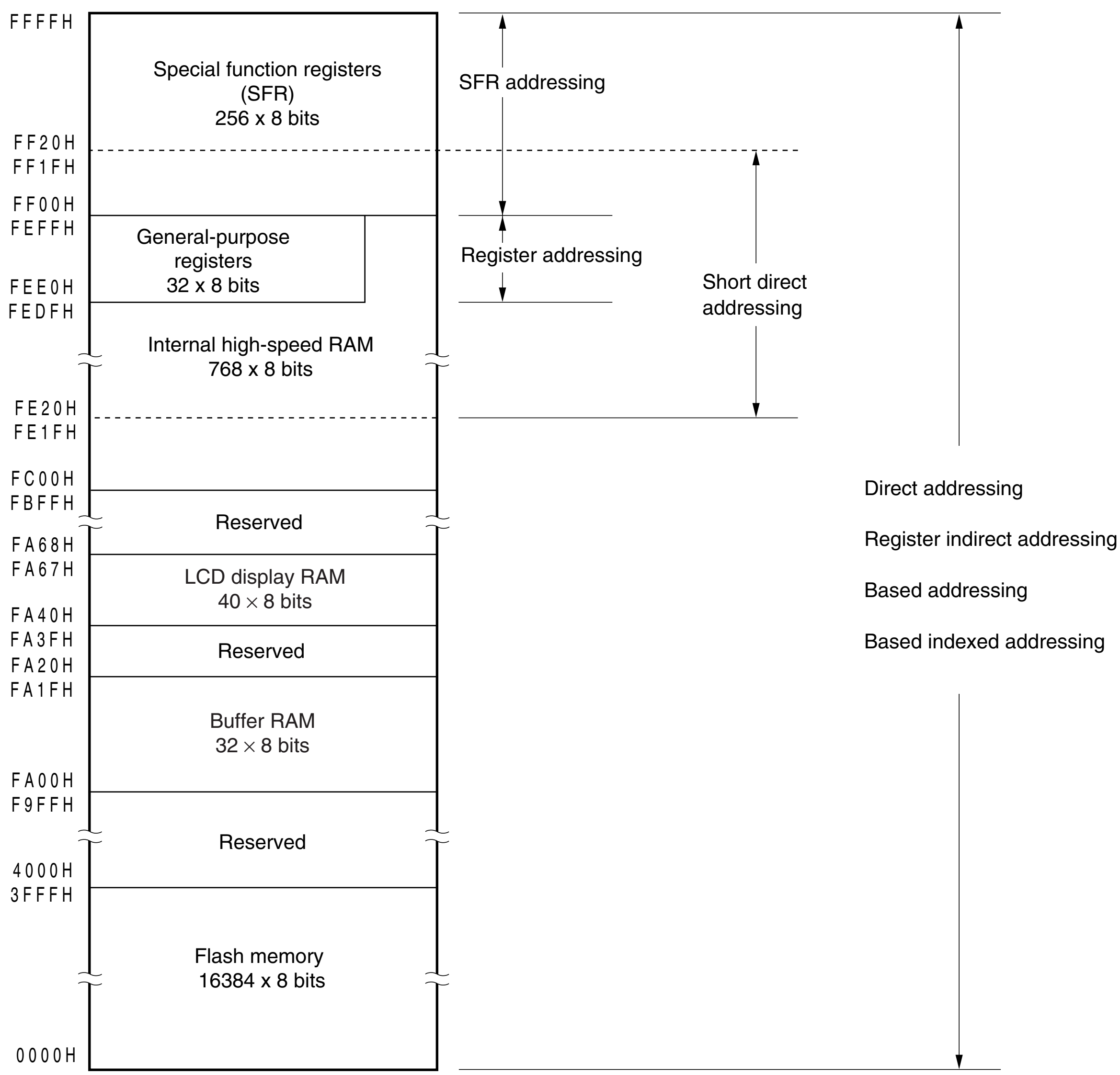
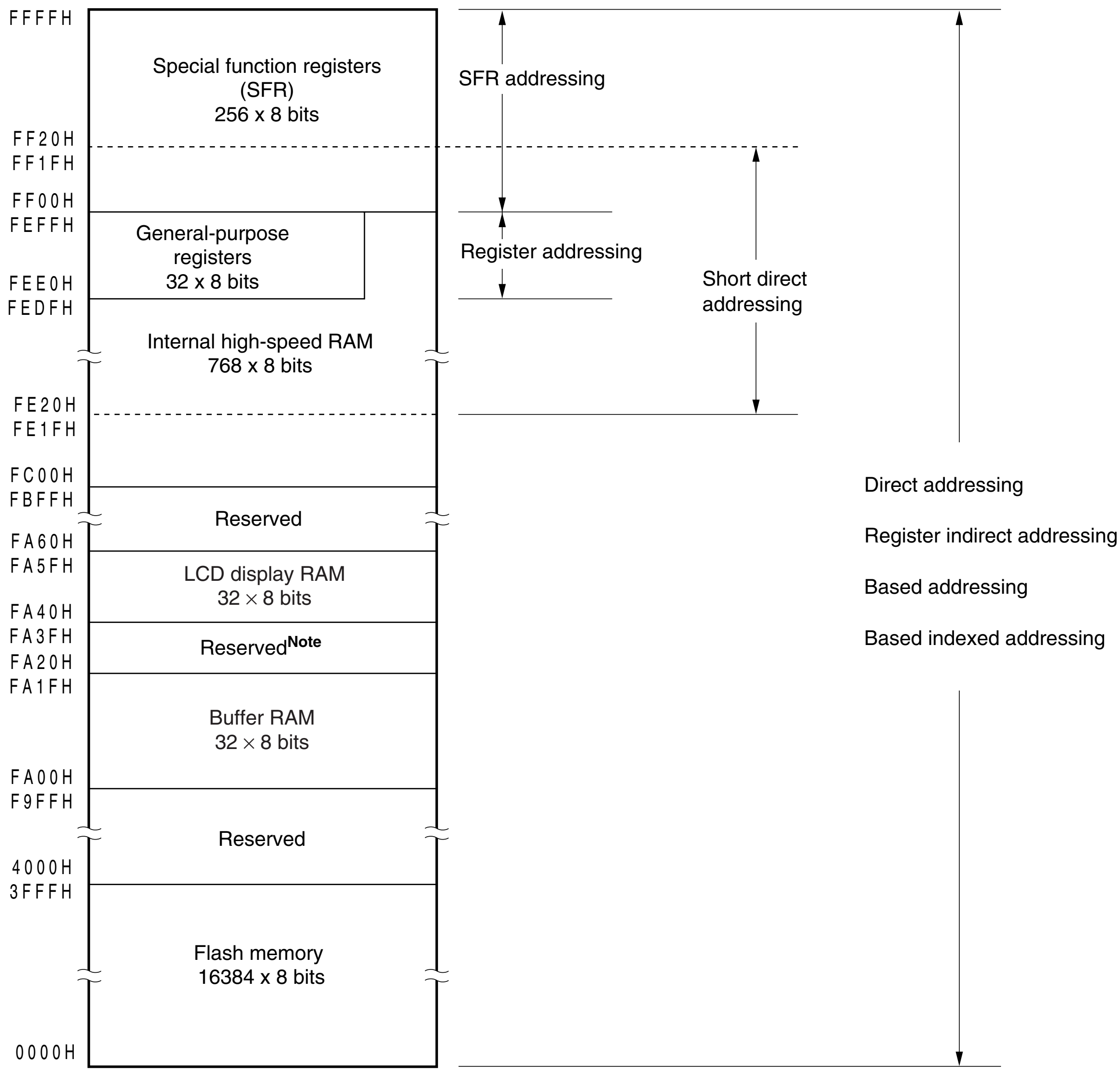


Figure 3-12. Correspondence Between Data Memory and Addressing (μPD78F0491)



<R> **Note** However, FA26H and FA27H can be used (See **13.3 Registers Used in 16-Bit ΔΣ Type A/D Converter**).

Figure 3-13. Correspondence Between Data Memory and Addressing (μPD78F0472, 78F0482)

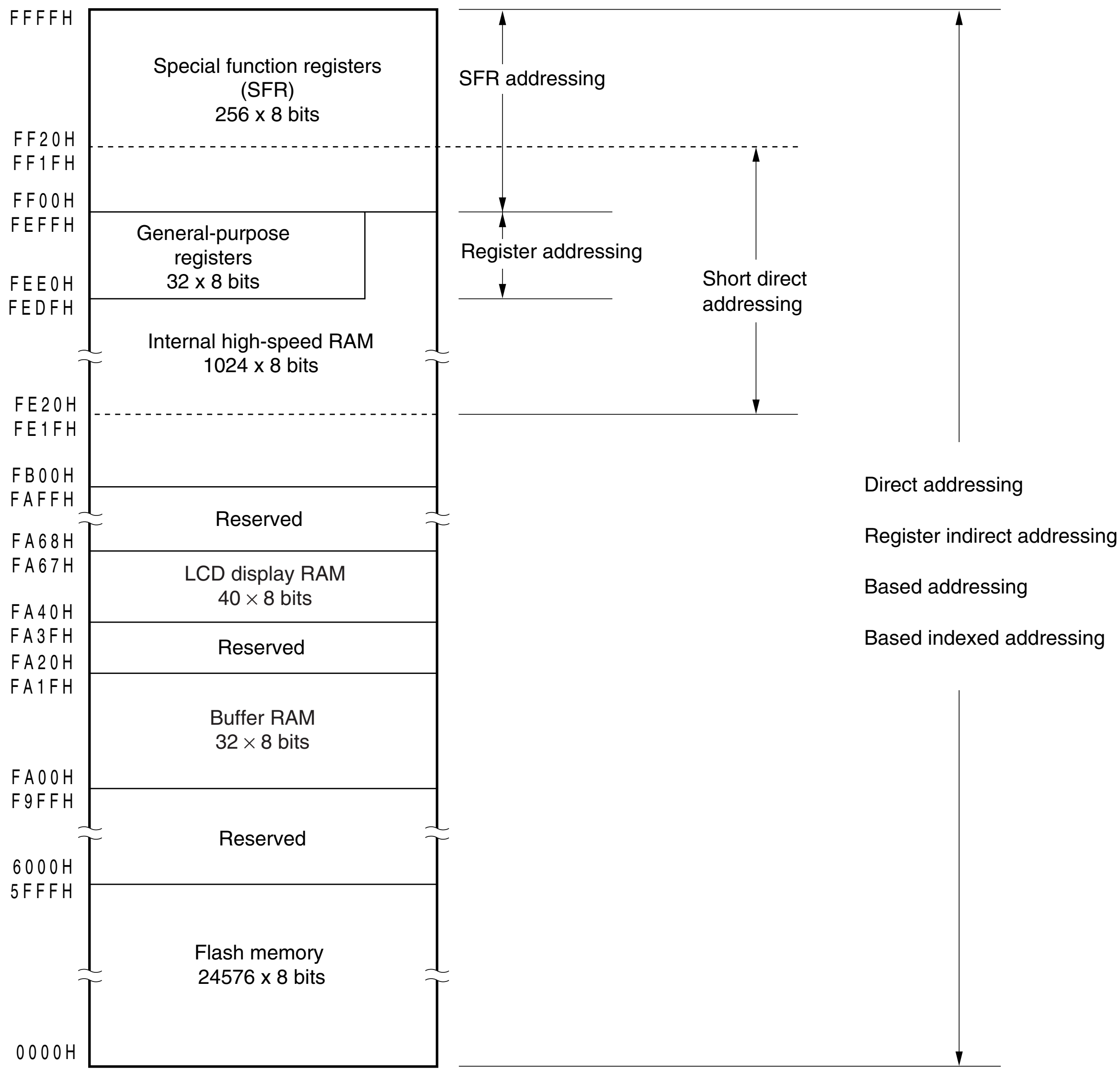
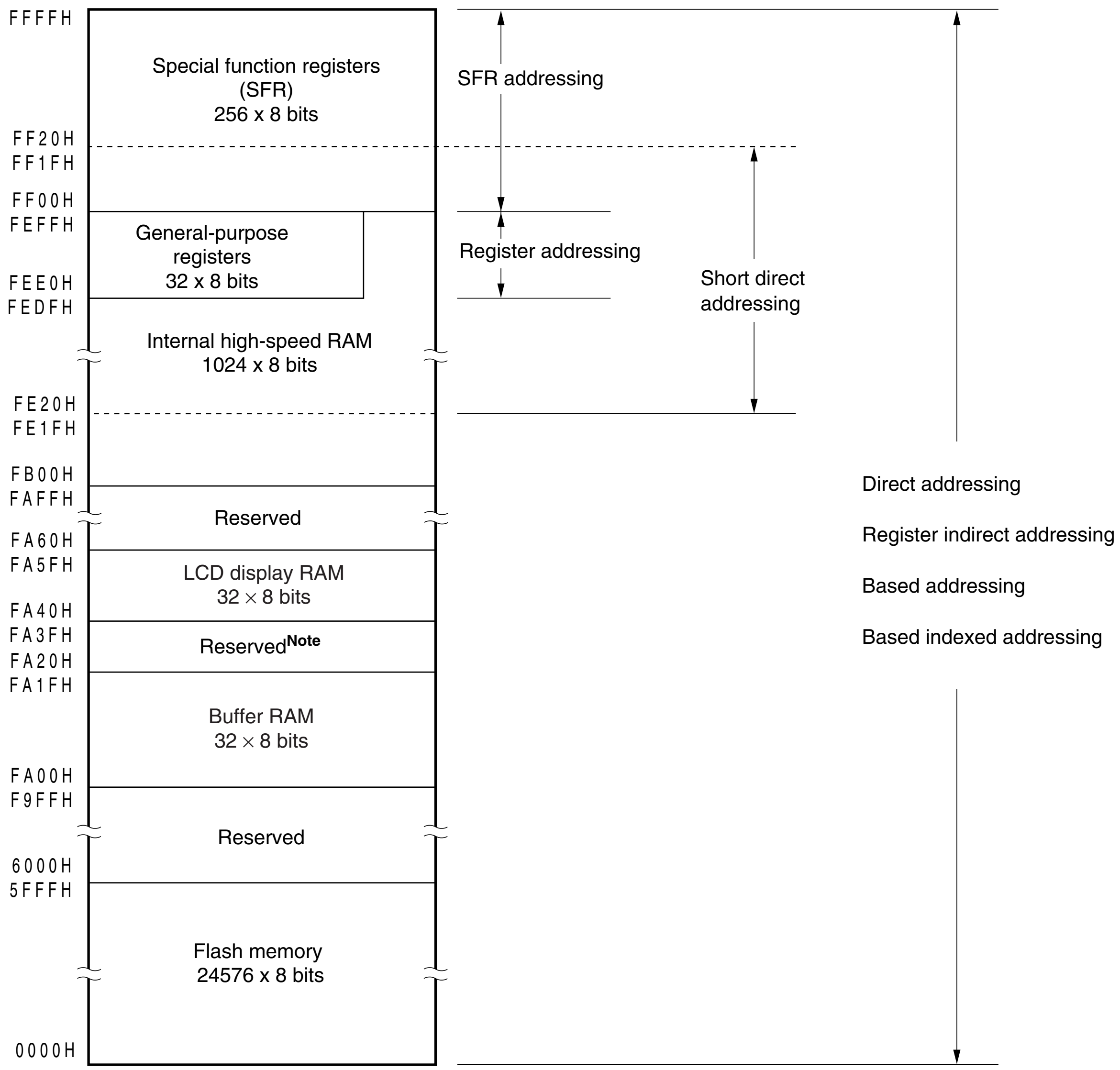


Figure 3-14. Correspondence Between Data Memory and Addressing (μPD78F0492)



<R> **Note** However, FA26H and FA27H can be used (See **13.3 Registers Used in 16-Bit ΔΣ Type A/D Converter**).

Figure 3-15. Correspondence Between Data Memory and Addressing (μPD78F0473, 78F0483)

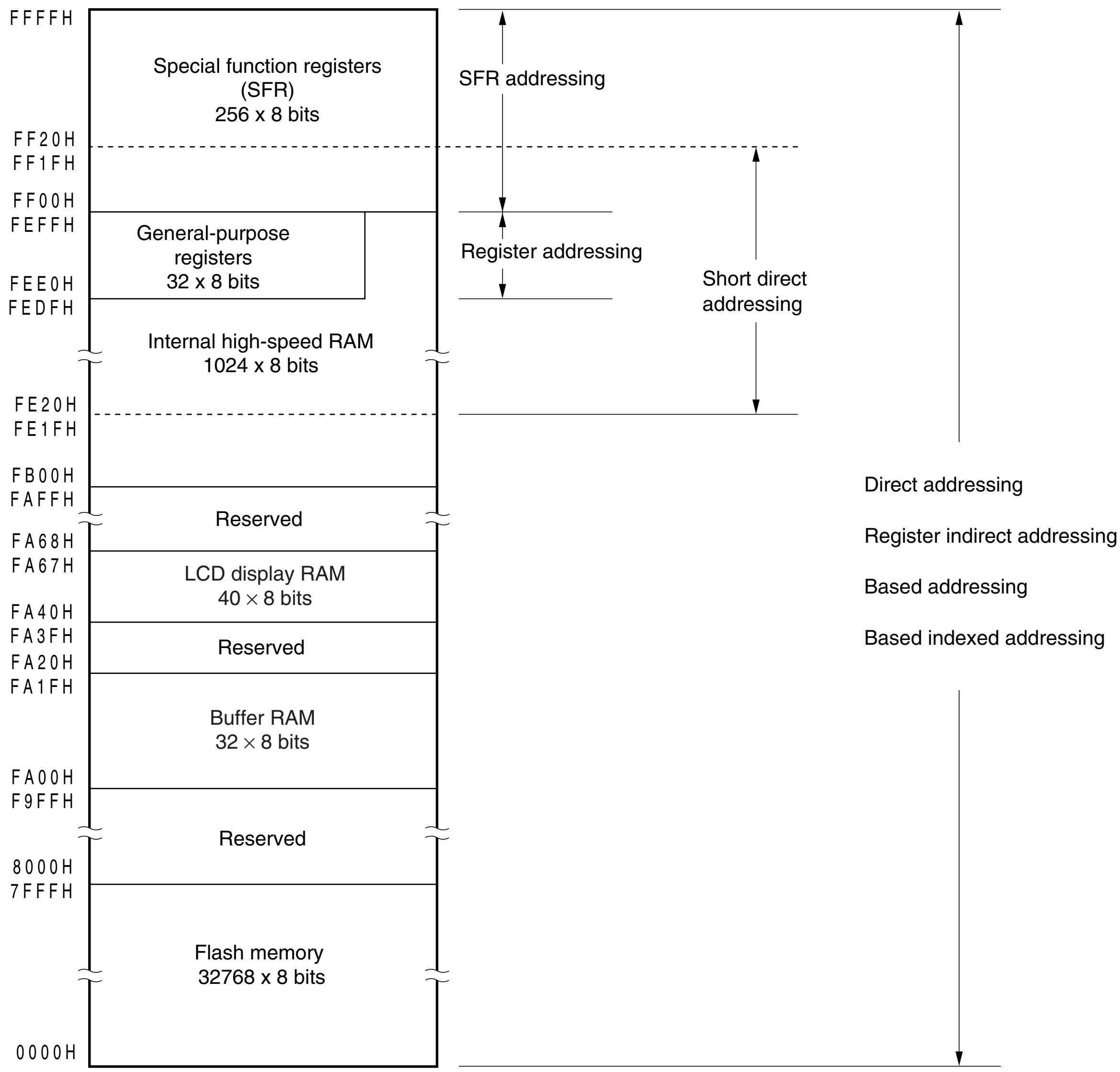
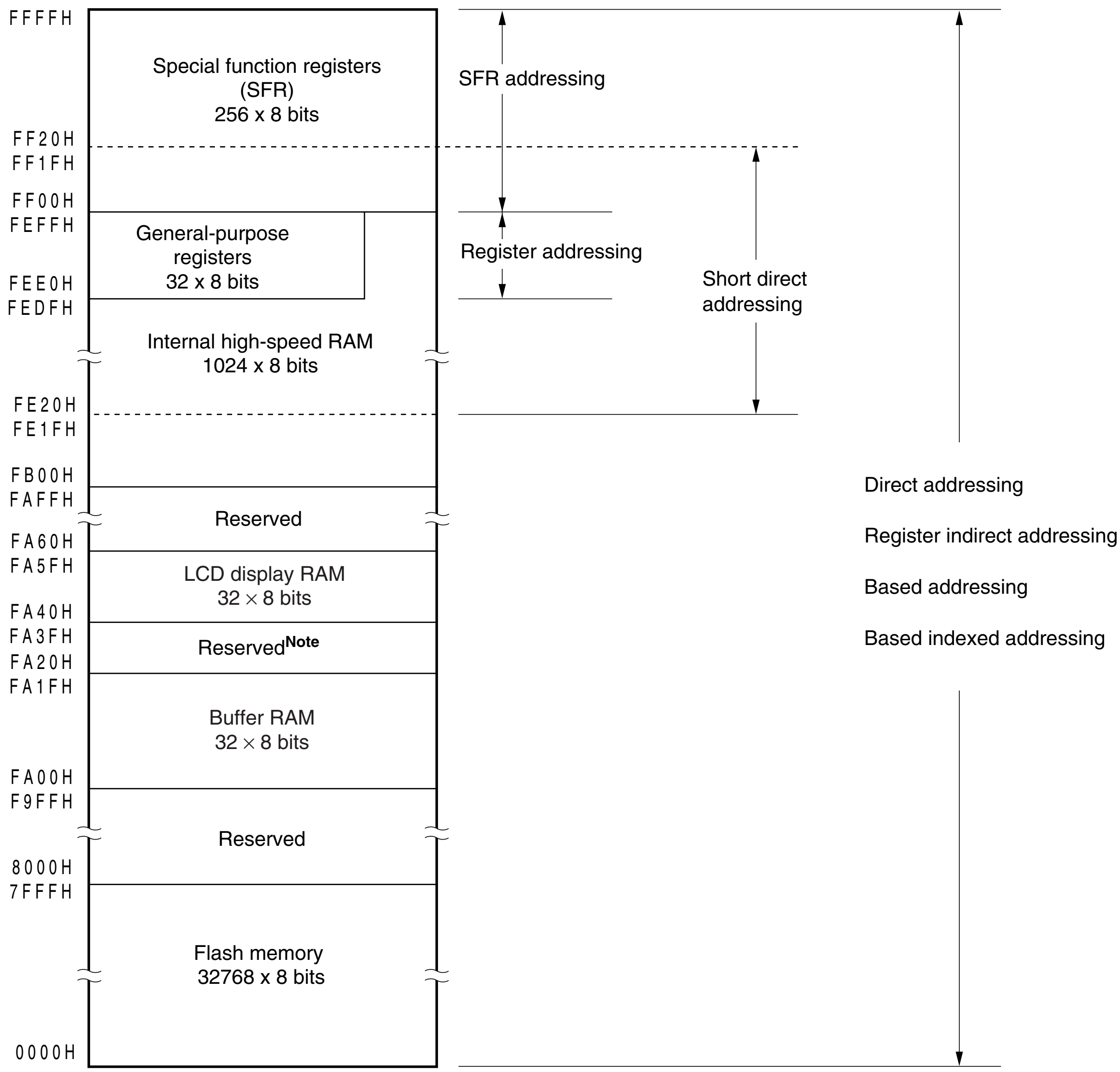


Figure 3-16. Correspondence Between Data Memory and Addressing (μPD78F0493)



<R> **Note** However, FA26H and FA27H can be used (See **13.3 Registers Used in 16-Bit ΔΣ Type A/D Converter**).

Figure 3-17. Correspondence Between Data Memory and Addressing (μPD78F0474, 78F0484)

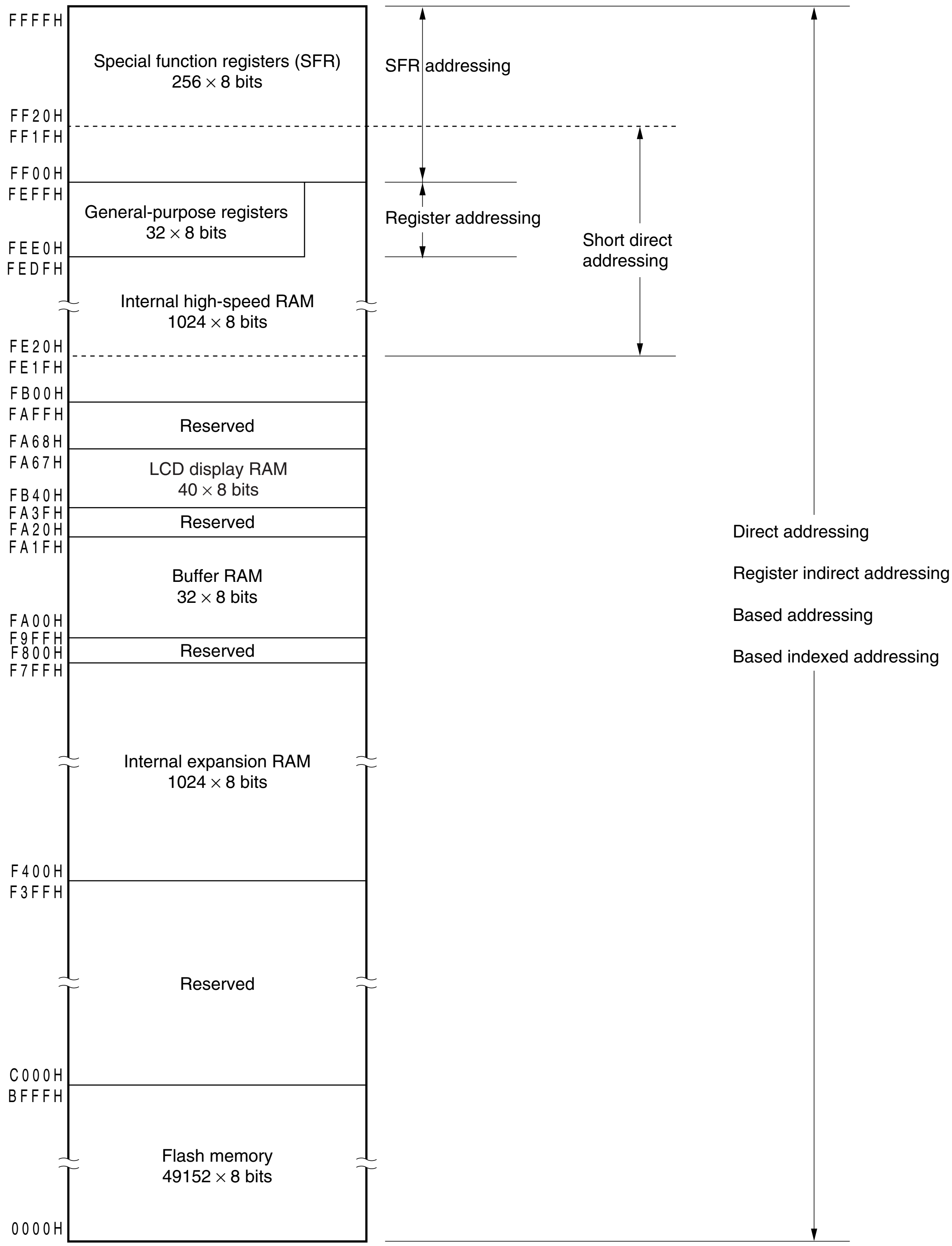
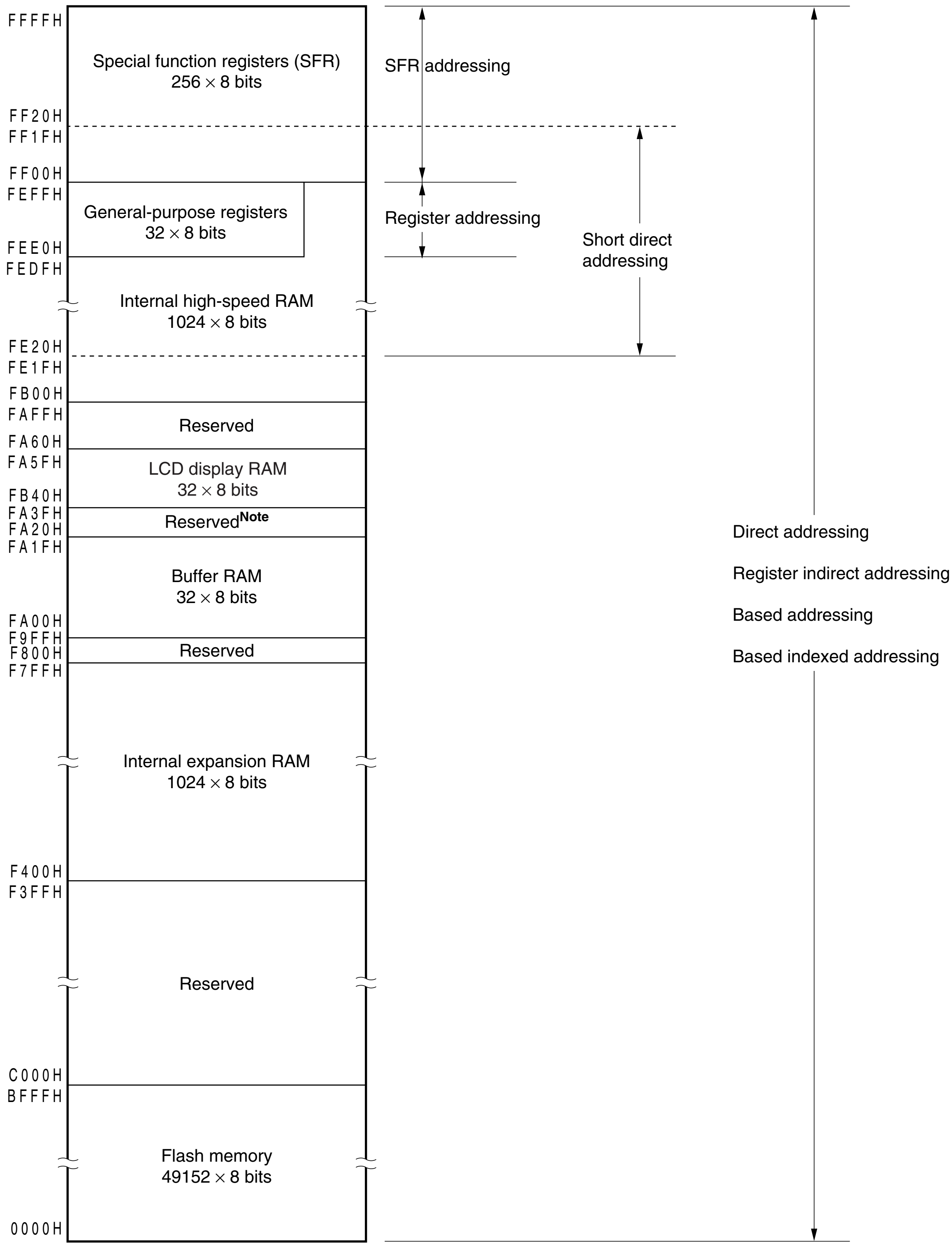


Figure 3-18. Correspondence Between Data Memory and Addressing (μPD78F0494)



<R> **Note** However, FA26H and FA27H can be used (See **13.3 Registers Used in 16-Bit ΔΣ Type A/D Converter**).

Figure 3-19. Correspondence Between Data Memory and Addressing (μPD78F0475, 78F0485)

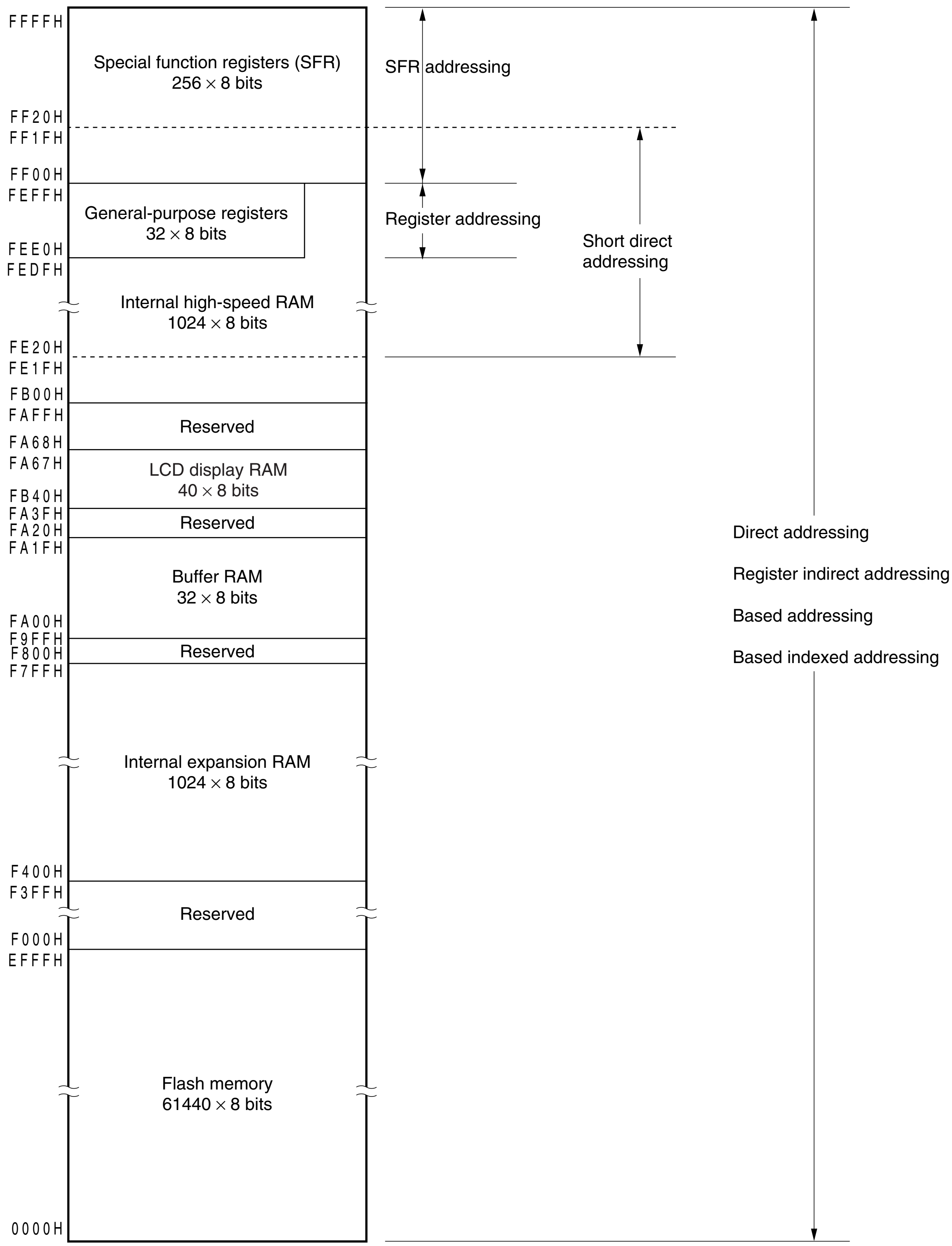
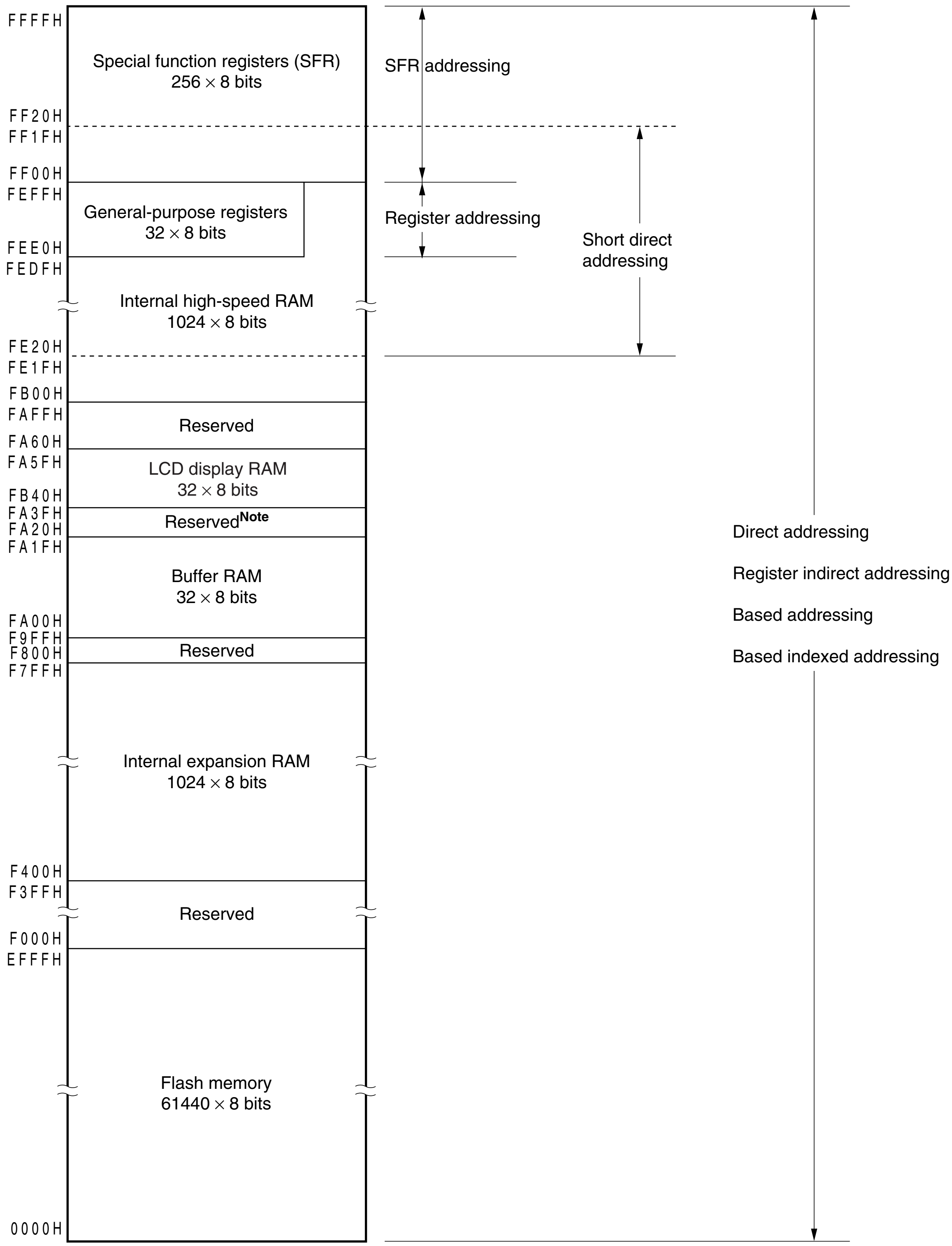


Figure 3-20. Correspondence Between Data Memory and Addressing (μPD78F0495)



<R> **Note** However, FA26H and FA27H can be used (See **13.3 Registers Used in 16-Bit ΔΣ Type A/D Converter**).

3.2 Processor Registers

The 78K0/LF3 products incorporate the following processor registers.

3.2.1 Control registers

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

(1) Program counter (PC)

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

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

Reset signal generation sets the reset vector table values at addresses 0000H and 0001H to the program counter.

Figure 3-21. Format of Program Counter

	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 set/reset by instruction execution.

Program status word contents are stored in the stack area upon vector interrupt request acknowledgment or PUSH PSW instruction execution and are restored upon execution of the RETB, RETI and POP PSW instructions.

Reset signal generation sets PSW to 02H.

Figure 3-22. Format of Program Status Word

	7							0
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	CY

(a) Interrupt enable flag (IE)

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

When 0, the IE flag is set to the interrupt disabled (DI) state, and all maskable interrupt requests are disabled.

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

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

(b) Zero flag (Z)

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

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

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

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

(d) Auxiliary carry flag (AC)

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

(e) In-service priority flag (ISP)

This flag manages the priority of acknowledgeable maskable vectored interrupts. When this flag is 0, low-level vectored interrupt requests specified by a priority specification flag register (PR0L, PR0H, PR1L, PR1H) (see **21.3 (3) Priority specification flag registers (PR0L, PR0H, PR1L, PR1H)**) can not be acknowledged. Actual request acknowledgment is controlled by the interrupt enable flag (IE).

(f) Carry flag (CY)

This flag stores overflow and underflow upon add/subtract instruction execution. It stores the shift-out value upon rotate instruction execution and functions as a bit accumulator during bit operation 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 3-23. Format of Stack Pointer

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

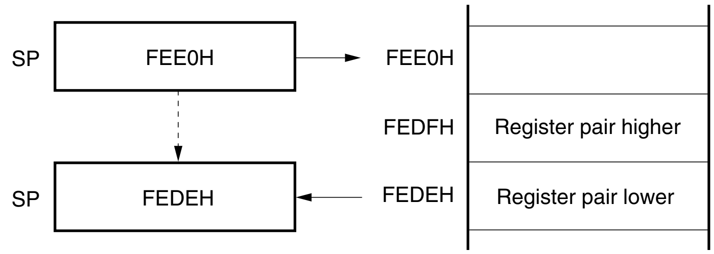
The SP is decremented ahead of write (save) to the stack memory and is incremented after read (restored) from the stack memory.

Each stack operation saves/restores data as shown in Figures 3-24 and 3-25.

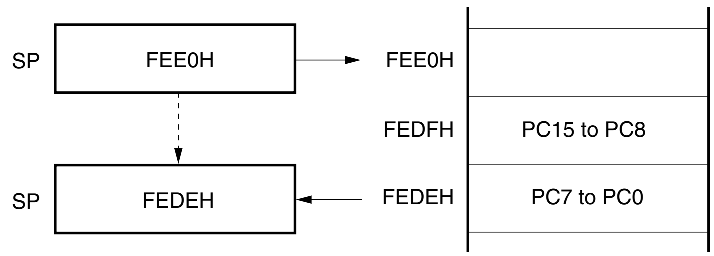
Caution Since reset signal generation makes the SP contents undefined, be sure to initialize the SP before using the stack.

Figure 3-24. Data to Be Saved to Stack Memory

(a) PUSH rp instruction (when SP = FEE0H)



(b) CALL, CALLF, CALLT instructions (when SP = FEE0H)



(c) Interrupt, BRK instructions (when SP = FEE0H)

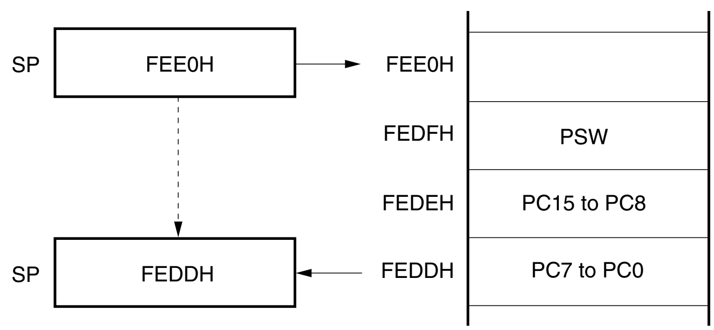
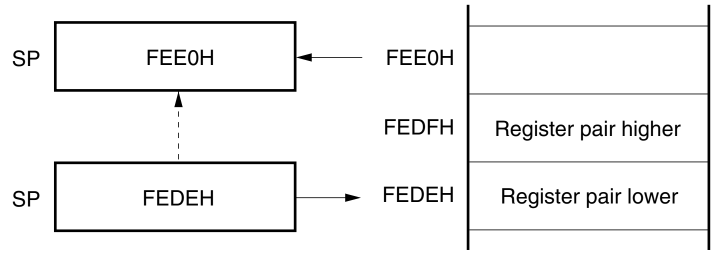
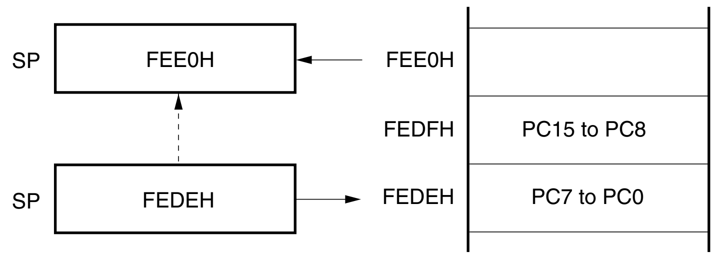


Figure 3-25. Data to Be Restored from Stack Memory

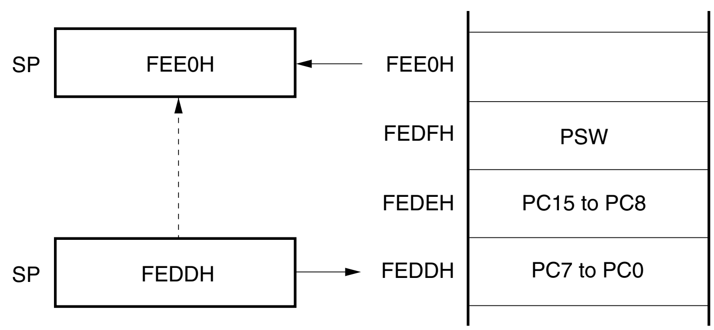
(a) POP rp instruction (when SP = FEDEH)



(b) RET instruction (when SP = FEDEH)



(c) RETI, RETB instructions (when SP = FEDDH)



3.2.2 General-purpose registers

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

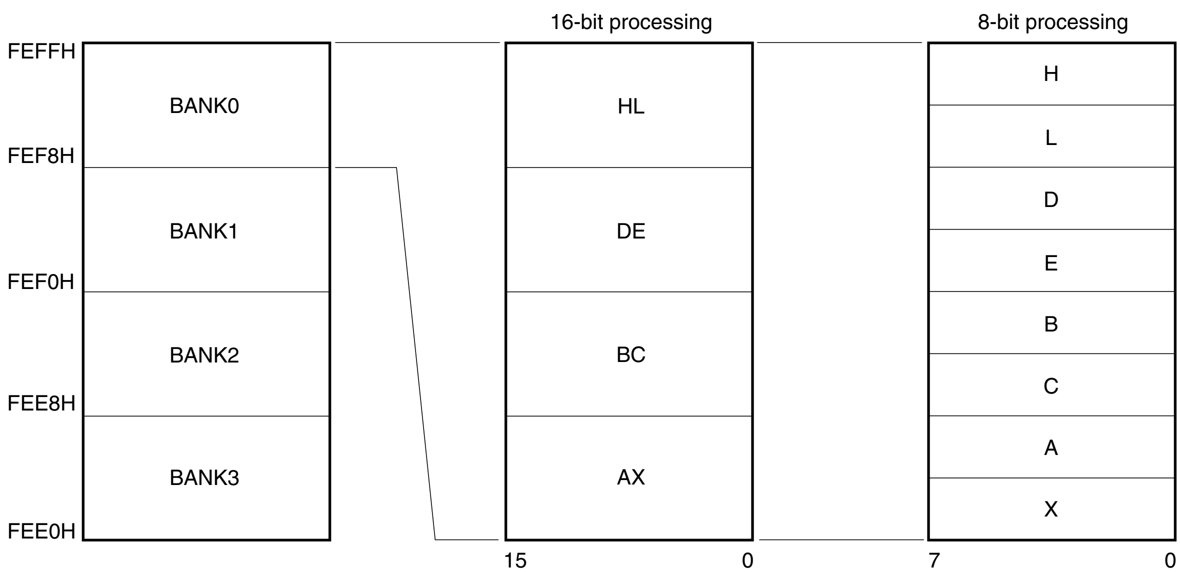
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

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

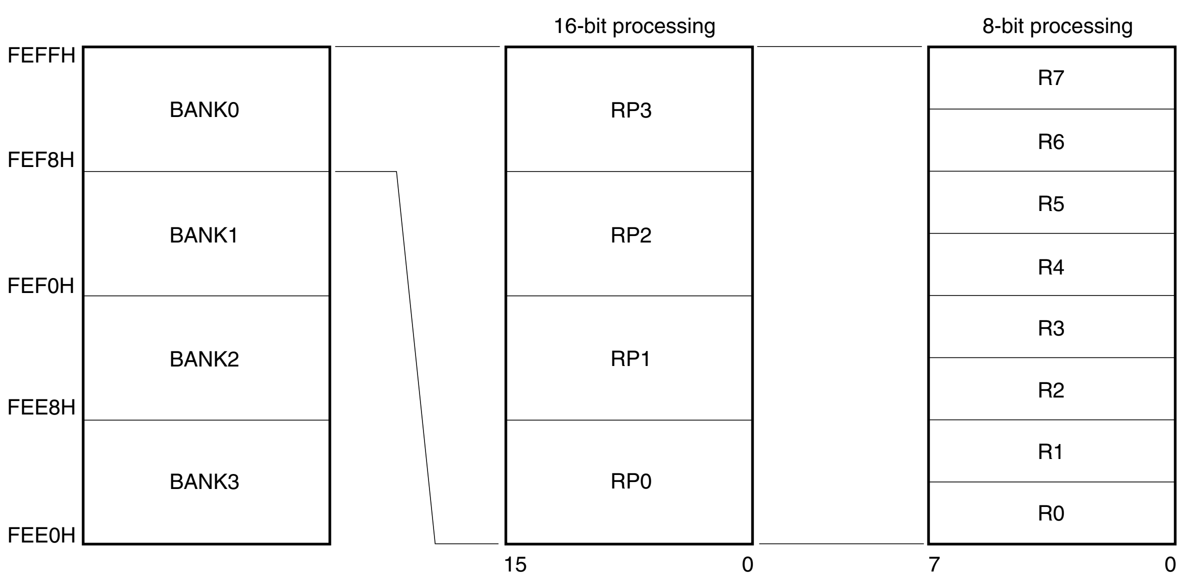
Register banks to be used for instruction execution are set by the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupts for each bank.

Figure 3-26. Configuration of General-Purpose Registers

(a) Function name



(b) Absolute name



3.2.3 Special function registers (SFRs)

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

SFRs are allocated to the FF00H to FFFFH areas in the CPU, and are allocated to the 00H to 03H areas of LCDCTL in the LCD controller/driver.

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

Each manipulation bit unit can be specified as follows.

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

Table 3-8 gives a list of the special function registers. The meanings of items in the table are as follows.

- Symbol
  Symbol indicating the address of a special function register. It is a reserved word in the RA78K0, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0. When using the RA78K0, ID78K0-QB, and SM+, symbols can be written as an instruction operand.
- R/W
  Indicates whether the corresponding special function register can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulatable bit units
  Indicates the manipulatable bit unit (1, 8, or 16). “–” indicates a bit unit for which manipulation is not possible.
- After reset
  Indicates each register status upon reset signal generation.

Table 3-8. Special Function Register List (1/5)

Address	Special Function Register (SFR) Name		Symbol	R/W	Manipulatable Bit Unit			After Reset
					1 Bit	8 Bits	16 Bits	
FF00H	Receive buffer register 6		RXB6	R	–	√	–	FFH
FF01H	Port register 1		P1	R/W	√	√	–	00H
FF02H	Port register 2		P2	R/W	√	√	–	00H
FF03H	Port register 3		P3	R/W	√	√	–	00H
FF04H	Port register 4		P4	R/W	√	√	–	00H
FF05H	Transmit buffer register 6		TXB6	R/W	–	√	–	FFH
FF06H	10-bit A/D conversion result register[Note]		ADCR	R	–	–	√	0000H
FF07H		8-bit A/D conversion result register H[Note]	ADCRH	R	–	√	–	00H
FF08H	Port register 8		P8	R/W	√	√	–	00H
FF09H	Port register 9		P9	R/W	√	√	–	00H
FF0AH	Port register 10		P10	R/W	√	√	–	00H
FF0BH	Port register 11		P11	R/W	√	√	–	00H
FF0CH	Port register 12		P12	R/W	√	√	–	00H
FF0DH	Port register 13		P13	R/W	√	√	–	00H
FF0EH	Port register 14		P14	R/W	√	√	–	00H
FF0FH	Port register 15		P15	R/W	√	√	–	00H
FF10H	16-bit timer counter 00		TM00	R	–	–	√	0000H
FF11H								
FF12H	16-bit timer capture/compare register 000		CR000	R/W	–	–	√	0000H
FF13H								
FF14H	16-bit timer capture/compare register 010		CR010	R/W	–	–	√	0000H
FF15H								
FF16H	8-bit timer counter 50		TM50	R	–	√	–	00H
FF17H	8-bit timer compare register 50		CR50	R/W	–	√	–	00H
FF18H	8-bit timer H compare register 00		CMP00	R/W	–	√	–	00H
FF19H	8-bit timer H compare register 10		CMP10	R/W	–	√	–	00H
FF1AH	8-bit timer H compare register 01		CMP01	R/W	–	√	–	00H
FF1BH	8-bit timer H compare register 11		CMP11	R/W	–	√	–	00H
FF1FH	Serial I/O shift register 10		SIO10	R	–	√	–	00H
FF20H	Port function register 1		PF1	R/W	√	√	–	00H
FF21H	Port mode register 1		PM1	R/W	√	√	–	FFH
FF22H	Port mode register 2		PM2	R/W	√	√	–	FFH
FF23H	Port mode register 3		PM3	R/W	√	√	–	FFH
FF24H	Port mode register 4		PM4	R/W	√	√	–	FFH
FF28H	Port mode register 8		PM8	R/W	√	√	–	FFH
FF29H	Port mode register 9		PM9	R/W	√	√	–	FFH
FF2AH	Port mode register 10		PM10	R/W	√	√	–	FFH
FF2BH	Port mode register 11		PM11	R/W	√	√	–	FFH
FF2CH	Port mode register 12		PM12	R/W	√	√	–	FFH
FF2DH	Port mode register 13		PM13	R/W	√	√	–	FFH
FF2EH	Port mode register 14		PM14	R/W	√	√	–	FFH
FF2FH	Port mode register 15		PM15	R/W	√	√	–	FFH

Note μPD78F048x and 78F049x only.

Table 3-8. Special Function Register List (2/5)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF30H	Internal high-speed oscillation trimming register	HIOTRM	R/W	–	√	–	10H
FF31H	Pull-up resistor option register 1	PU1	R/W	√	√	–	00H
FF33H	Pull-up resistor option register 3	PU3	R/W	√	√	–	00H
FF34H	Pull-up resistor option register 4	PU4	R/W	√	√	–	00H
FF38H	Pull-up resistor option register 8	PU8	R/W	√	√	–	00H
FF39H	Pull-up resistor option register 9	PU9	R/W	√	√	–	00H
FF3AH	Pull-up resistor option register 10	PU10	R/W	√	√	–	00H
FF3BH	Pull-up resistor option register 11	PU11	R/W	√	√	–	00H
FF3CH	Pull-up resistor option register 12	PU12	R/W	√	√	–	00H
FF3DH	Pull-up resistor option register 13	PU13	R/W	√	√	–	00H
FF3EH	Pull-up resistor option register 14	PU14	R/W	√	√	–	00H
FF3FH	Pull-up resistor option register 15	PU15	R/W	√	√	–	00H
FF40H	Clock output selection register	CKS	R/W	√	√	–	00H
FF41H	8-bit timer compare register 51	CR51	R/W	–	√	–	00H
FF42H	8-bit timer H mode register 2	TMHMD2	R/W	√	√	–	00H
FF43H	8-bit timer mode control register 51	TMC51	R/W	√	√	–	00H
FF44H	8-bit timer H compare register 02	CMP02	R/W	–	√	–	00H
FF45H	8-bit timer H compare register 12	CMP12	R/W	–	√	–	00H
FF47H	MCG status register	MC0STR	R	√	√	–	00H
FF48H	External interrupt rising edge enable register	EGP	R/W	√	√	–	00H
FF49H	External interrupt falling edge enable register	EGN	R/W	√	√	–	00H
FF4AH	MCG transmit buffer register	MC0TX	R/W	–	√	–	FFH
FF4BH	MCG transmit bit count specification register	MC0BIT	R/W	–	√	–	07H
FF4CH	MCG control register 0	MC0CTL0	R/W	√	√	–	10H
FF4DH	MCG control register 1	MC0CTL1	R/W	–	√	–	00H
FF4EH	MCG control register 2	MC0CTL2	R/W	–	√	–	1FH
FF4FH	Input switch control register	ISC	R/W	√	√	–	00H
FF50H	Asynchronous serial interface operation mode register 6	ASIM6	R/W	√	√	–	01H
FF51H	8-bit timer counter 52	TM52	R	–	√	–	00H
FF53H	Asynchronous serial interface reception error status register 6	ASIS6	R	–	√	–	00H
FF54H	Real-time counter clock selection register	RTCCL	R/W	√	√	–	00H
FF55H	Asynchronous serial interface transmission status register 6	ASIF6	R	–	√	–	00H
FF56H	Clock selection register 6	CKSR6	R/W	–	√	–	00H
FF57H	Baud rate generator control register 6	BRGC6	R/W	–	√	–	FFH
FF58H	Asynchronous serial interface control register 6	ASICL6	R/W	√	√	–	16H
FF59H	8-bit timer compare register 52	CR52	R/W	–	√	–	00H
FF5BH	Timer clock selection register 52	TCL52	R/W	√	√	–	00H
FF5CH	8-bit timer mode control register 52	TMC52	R/W	√	√	–	00H

Table 3-8. Special Function Register List (3/5)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF60H	Sub-count register	RSUBC	R	–	–	√	0000H
FF61H							
FF62H	Second count register	SEC	R/W	–	√	–	00H
FF63H	Minute count register	MIN	R/W	–	√	–	00H
FF64H	Hour count register	HOUR	R/W	–	√	–	12H
FF65H	Week count register	WEEK	R/W	–	√	–	00H
FF66H	Day count register	DAY	R/W	–	√	–	01H
FF67H	Month count register	MONTH	R/W	–	√	–	01H
FF68H	Year count register	YEAR	R/W	–	√	–	00H
FF69H	8-bit timer H mode register 0	TMHMD0	R/W	√	√	–	00H
FF6AH	Timer clock selection register 50	TCL50	R/W	√	√	–	00H
FF6BH	8-bit timer mode control register 50	TMC50	R/W	√	√	–	00H
FF6CH	8-bit timer H mode register 1	TMHMD1	R/W	√	√	–	00H
FF6DH	8-bit timer H carrier control register 1	TMCYC1	R/W	√	√	–	00H
FF6EH	Key return mode register	KRM	R/W	√	√	–	00H
FF6FH	8-bit timer counter 51	TM51	R	–	√	–	00H
FF70H	Asynchronous serial interface operation mode register 0	ASIM0	R/W	√	√	–	01H
FF71H	Baud rate generator control register 0	BRGC0	R/W	–	√	–	1FH
FF72H	Receive buffer register 0	RXB0	R	–	√	–	FFH
FF73H	Asynchronous serial interface reception error status register 0	ASIS0	R	–	√	–	00H
FF74H	Transmit shift register 0	TXS0	W	–	√	–	FFH
FF75H	16-bit ΔΣ A/D conversion end channel register[Note 1]	ADDSTR	R	–	√	–	00H
FF7CH	ΔΣ A/D converter control register 0[Note 1]	ADDCTL0	R/W	√	√	–	00H
FF7DH	ΔΣ A/D converter control register 1[Note 1]	ADDCTL1	R/W	√	√	–	00H
FF7EH	16-bit ΔΣ A/D conversion result register[Note 1]	ADDCR	R	–	–	√	0000H
FF7FH	8-bit ΔΣ A/D conversion result register[Note 1]	ADDCRH	R	–	√	–	00H
FF80H	Serial operation mode register 10	CSIM10	R/W	√	√	–	00H
FF81H	Serial clock selection register 10	CSIC10	R/W	√	√	–	00H
FF82H	Watch error correction register	SUBCUD	R/W	√	√	–	00H
FF84H	Transmit buffer register 10	SOTB10	R/W	–	√	–	00H
FF86H	Alarm minute register	ALARMWM	R/W	–	√	–	00H
FF87H	Alarm hour register	ALARMWH	R/W	–	√	–	12H
FF88H	Alarm week register	ALARMWW	R/W	–	√	–	00H
FF89H	Real-time counter control register 0	RTCC0	R/W	√	√	–	00H
FF8AH	Real-time counter control register 1	RTCC1	R/W	√	√	–	00H
FF8BH	Real-time counter control register 2	RTCC2	R/W	√	√	–	00H
FF8CH	Timer clock selection register 51	TCL51	R/W	√	√	–	00H
FF8DH	A/D converter mode register[Note 2]	ADM	R/W	√	√	–	00H
FF8EH	Analog input channel specification register[Note 2]	ADS	R/W	√	√	–	00H
FF8FH	A/D port configuration register 0[Note 2]	ADPC0	R/W	√	√	–	08H

Notes 1. μPD78F049x only.

2. μPD78F048x and 78F049x only.

Table 3-8. Special Function Register List (4/5)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
FF90H	Serial operation mode specification register 0	CSIMA0	R/W	√	√	–	00H
FF91H	Serial status register 0	CSIS0	R/W	√	√	–	00H
FF92H	Serial trigger register 0	CSIT0	R/W	√	√	–	00H
FF93H	Division value selection register 0	BRGCA0	R/W	–	√	–	03H
FF94H	Automatic data transfer address point specification register 0	ADTP0	R/W	–	√	–	00H
FF95H	Automatic data transfer interval specification register 0	ADTI0	R/W	–	√	–	00H
FF96H	Serial I/O shift register 0	SIOA0	R/W	–	√	–	00H
FF97H	Automatic data transfer address count register 0	ADTC0	R	–	√	–	00H
FF99H	Watchdog timer enable register	WDTE	R/W	–	√	–	Note 1 1AH/9AH
FF9AH	Remote controller receive control register	RMCN	R/W	√	√	–	00H
FF9BH	Remote controller receive data register	RMDR	R	–	√	–	00H
FF9CH	Remote controller shift register receive counter register	RMSCR	R	–	√	–	00H
FF9FH	Clock operation mode select register	OSCCTL	R/W	√	√	–	00H
FFA0H	Internal oscillation mode register	RCM	R/W	√	√	–	80H Note 2
FFA1H	Main clock mode register	MCM	R/W	√	√	–	00H
FFA2H	Main OSC control register	MOC	R/W	√	√	–	80H
FFA3H	Oscillation stabilization time counter status register	OSTC	R	√	√	–	00H
FFA4H	Oscillation stabilization time select register	OSTS	R/W	–	√	–	05H
FFA5H	Remote controller receive GPHS compare register	RMGPHS	R/W	–	√	–	00H
FFA6H	Remote controller receive GPHL compare register	RMGPHL	R/W	–	√	–	00H
FFA7H	Remote controller receive DLS compare register	RMDLS	R/W	–	√	–	00H
FFA8H	Remote controller receive DLL compare register	RMDLL	R/W	–	√	–	00H
FFA9H	Remote controller receive DH0S compare register	RMDH0S	R/W	–	√	–	00H
FFAAH	Remote controller receive DH0L compare register	RMDH0L	R/W	–	√	–	00H
FFABH	Remote controller receive shift register	RMSR	R	–	√	–	00H
FFACH	Reset control flag register	RESF	R	–	√	–	00H Note 3
FFADH	Remote controller receive DH1S compare register	RMDH1S	R/W	–	√	–	00H
FFAEH	Remote controller receive DH1L compare register	RMDH1L	R/W	–	√	–	00H
FFAFH	Remote controller receive end width select register	RMER	R/W	–	√	–	00H

Notes 1. The reset value of WDTE is determined by the setting of the option byte.

2. The value of this register is 00H immediately after a reset release but automatically changes to 80H after oscillation accuracy stabilization of high-speed internal oscillator has been waited.

3. The reset value of RESF varies depending on the reset source.

Table 3-8. Special Function Register List (5/5)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1 Bit	8 Bits	16 Bits	
FFB0H	LCD mode register	LCDMD		R/W	√	√	–	00H
FFB1H	LCD display mode register	LCDM		R/W	√	√	–	00H
FFB2H	LCD clock control register 0	LCDC0		R/W	√	√	–	00H
FFB5H	Port function register 2[Note 1]	PF2		R/W	√	√	–	00H
FFB6H	Port function register ALL	PFALL		R/W	√	√	–	00H
FFBAH	16-bit timer mode control register 00	TMC00		R/W	√	√	–	00H
FFBBH	Prescaler mode register 00	PRM00		R/W	√	√	–	00H
FFBCH	Capture/compare control register 00	CRC00		R/W	√	√	–	00H
FFBDH	16-bit timer output control register 00	TOC00		R/W	√	√	–	00H
FFBEH	Low-voltage detection register	LVIM		R/W	√	√	–	00H[Note 2]
FFBFH	Low-voltage detection level selection register	LVIS		R/W	√	√	–	00H[Note 2]
FFE0H	Interrupt request flag register 0L	IF0	IF0L	R/W	√	√	√	00H
FFE1H	Interrupt request flag register 0H		IF0H	R/W	√	√		00H
FFE2H	Interrupt request flag register 1L	IF1	IF1L	R/W	√	√	√	00H
FFE3H	Interrupt request flag register 1H		IF1H	R/W	√	√		00H
FFE4H	Interrupt mask flag register 0L	MK0	MK0L	R/W	√	√	√	FFH
FFE5H	Interrupt mask flag register 0H		MK0H	R/W	√	√		FFH
FFE6H	Interrupt mask flag register 1L	MK1	MK1L	R/W	√	√	√	FFH
FFE7H	Interrupt mask flag register 1H		MK1H	R/W	√	√		FFH
FFE8H	Priority specification flag register 0L	PR0	PR0L	R/W	√	√	√	FFH
FFE9H	Priority specification flag register 0H		PR0H	R/W	√	√		FFH
FFEAH	Priority specification flag register 1L	PR1	PR1L	R/W	√	√	√	FFH
FFEBH	Priority specification flag register 1H		PR1H	R/W	√	√		FFH
FFF0H	Internal memory size switching register[Note 3]	IMS		R/W	–	√	–	CFH
FFF4H	Internal expansion RAM size switching register[Note 3]	IXS		R/W	–	√	–	0CH
FFF9H	Remote controller receive interrupt status register	INTS		R	√	√	–	00H
FFFAH	Remote controller receive interrupt status clear register	INTC		R/W	√	√	–	00H
FFFBH	Processor clock control register	PCC		R/W	√	√	–	01H

Notes 1. μPD78F047x and 78F048x only.

2. The reset values of LVIM and LVIS vary depending on the reset source.

3. Regardless of the internal memory capacity, the initial values of the internal memory size switching register (IMS) and internal expansion RAM size switching register (IXS) of all products in the 78K0/LF3 are fixed (IMS = CFH, IXS = 0CH). Therefore, set the value corresponding to each product as indicated below.

Flash Memory Version (78K0/LF3)	IMS	IXS	ROM Capacity	Internal High-Speed RAM Capacity	Internal Expansion RAM Capacity
μPD78F0471, 78F0481, 78F0491	04H	0CH	16 KB	768 bytes	–
μPD78F0472, 78F0482, 78F0492	C6H		24 KB	1 KB	
μPD78F0473, 78F0483, 78F0493	C8H		32 KB		
μPD78F0474, 78F0484, 78F0494	CCH	0AH	48 KB		1 KB
μPD78F0475, 78F0485, 78F0495	CFH		60 KB		

3.3 Instruction Address Addressing

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

3.3.1 Relative addressing

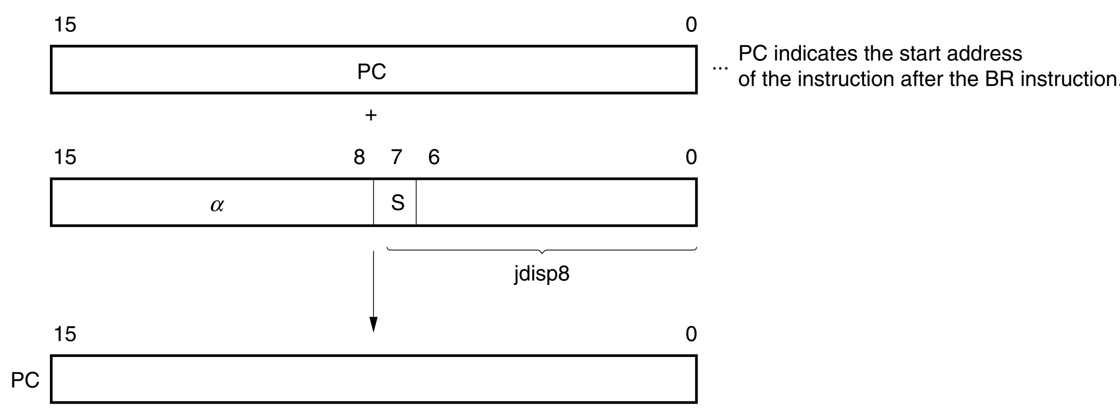
[Function]

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

In other words, relative addressing consists of relative branching from the start address of the following instruction to the −128 to +127 range.

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

[Illustration]



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

3.3.2 Immediate addressing

[Function]

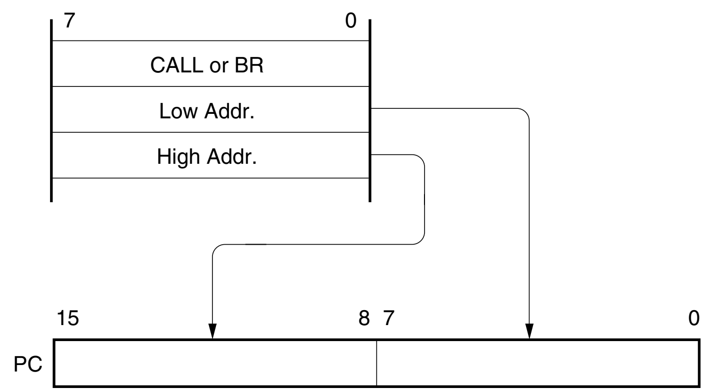
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

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

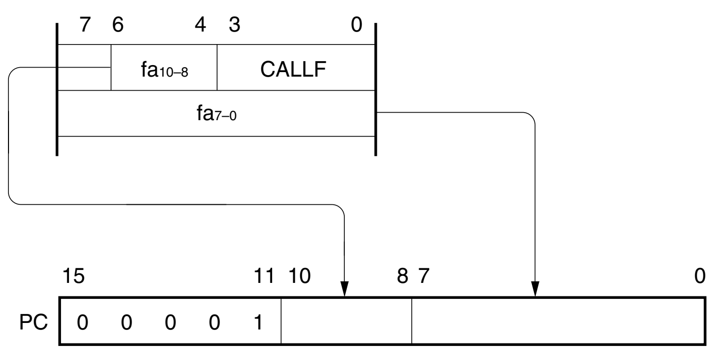
CALL !addr16 and BR !addr16 instructions can be branched to the entire memory space. The CALLF !addr11 instruction is branched to the 0800H to 0FFFH area.

[Illustration]

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



In the case of CALLF !addr11 instruction



3.3.3 Table indirect addressing

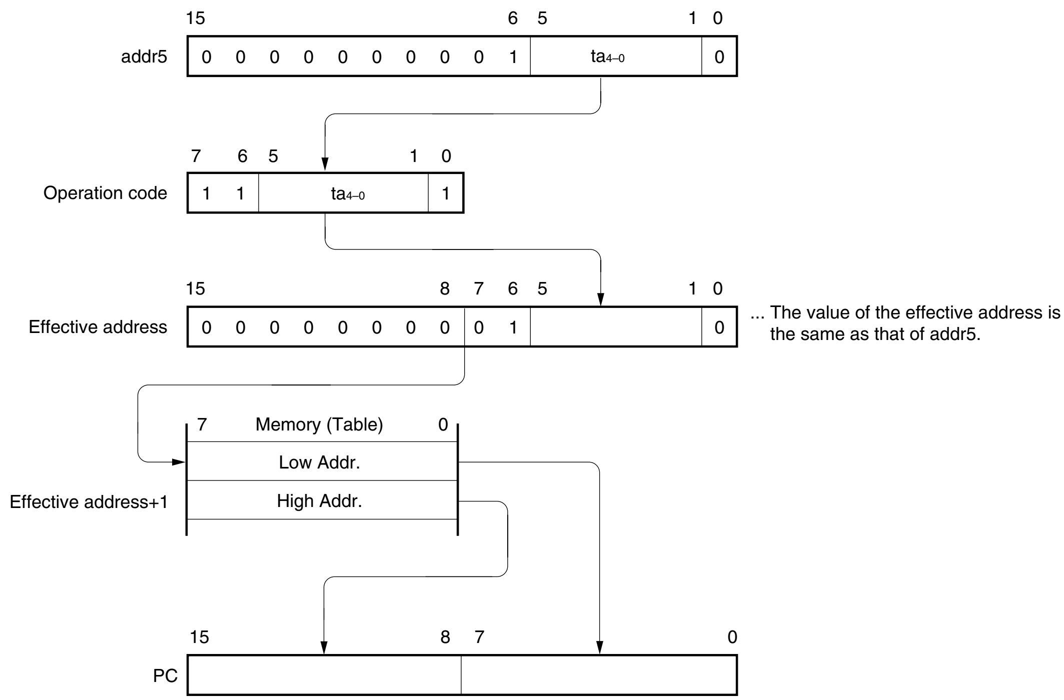
[Function]

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

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

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

<R> **[Illustration]**



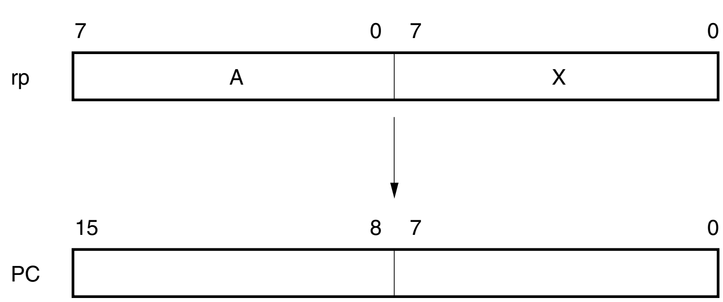
3.3.4 Register addressing

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

3.4.1 Implied addressing

[Function]

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

Of the 78K0/LF3 instruction words, the following instructions employ implied addressing.

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

[Operand format]

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

[Description example]

In the case of MULU X

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

3.4.2 Register addressing

[Function]

The general-purpose register to be specified is accessed as an operand with the register bank select flags (RBS0 to RBS1) and the register specify codes of an operation code.

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

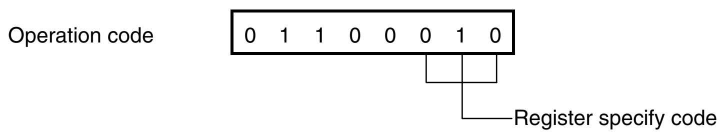
[Operand format]

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

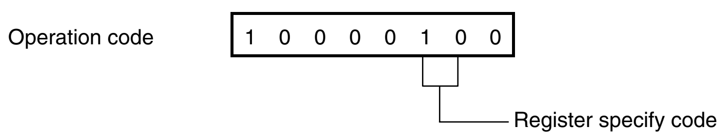
'r' and 'rp' can be described by absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[Description example]

MOV A, C; when selecting C register as r



INCW DE; when selecting DE register pair as rp



3.4.3 Direct addressing

[Function]

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

[Operand format]

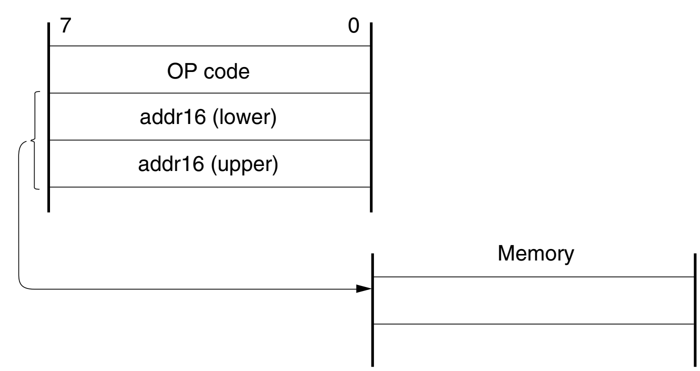
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

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

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

[Illustration]



3.4.4 Short direct addressing

[Function]

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

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

The SFR area (FF00H to FF1FH) where short direct addressing is applied is a part of the overall SFR area. Ports that are frequently accessed in a program and compare and capture registers of the timer/event counter are mapped in this area, allowing SFRs to 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 the **[Illustration]** shown below.

[Operand format]

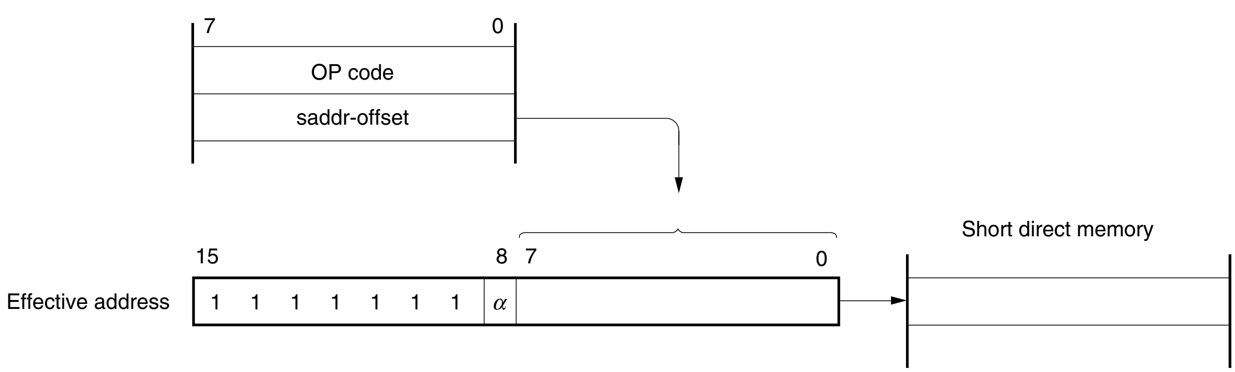
Identifier	Description
saddr	Immediate data that indicate label or FE20H to FF1FH
saddrp	Immediate data that indicate label or FE20H to FF1FH (even address only)

[Description example]

MOV 0FE30H, A ; When transferring the value of A register to the saddr (FE30H)

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

[Illustration]



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

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

3.4.5 Special function register (SFR) addressing

[Function]

A 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 SFRs mapped at FF00H to FF1FH can be accessed with short direct addressing.

[Operand format]

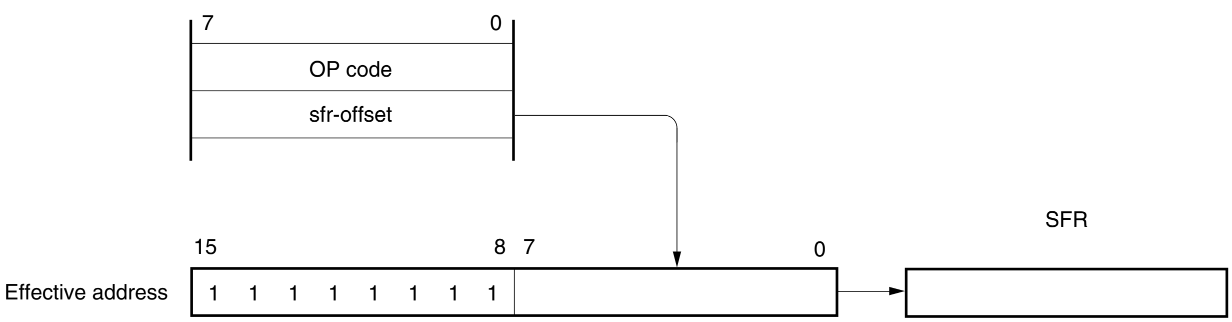
Identifier	Description
sfr	Special function register name
sfrp	16-bit manipulatable special function register name (even address only)

[Description example]

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

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

[Illustration]



3.4.6 Register indirect addressing

[Function]

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

[Operand format]

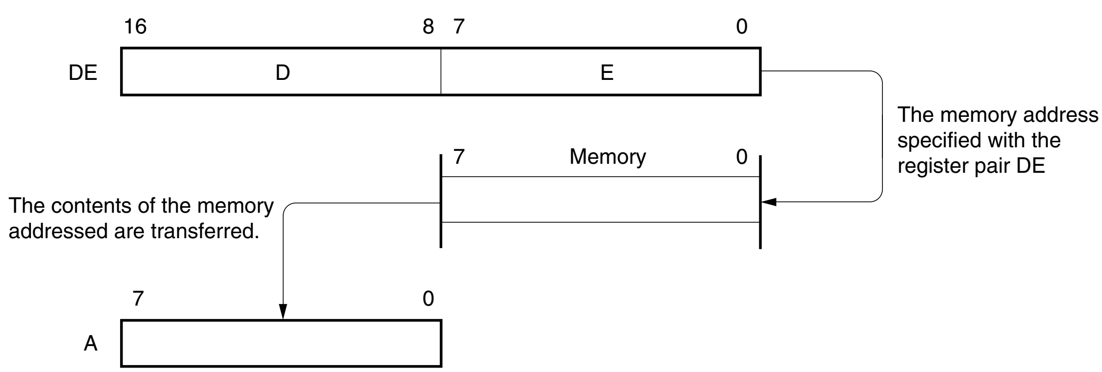
Identifier	Description
–	[DE], [HL]

[Description example]

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

Operation code 1 0 0 0 0 1 0 1

[Illustration]



3.4.7 Based addressing

[Function]

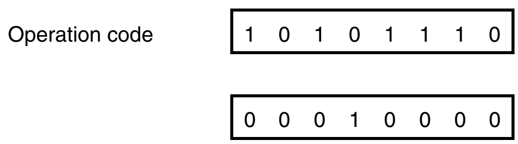
8-bit immediate data is added as offset data to the contents of the base register, that is, the HL register pair in the register bank specified by the register bank select flag (RBS0 and RBS1), and the sum is used to address the memory. Addition is performed by expanding the offset data as a positive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

[Operand format]

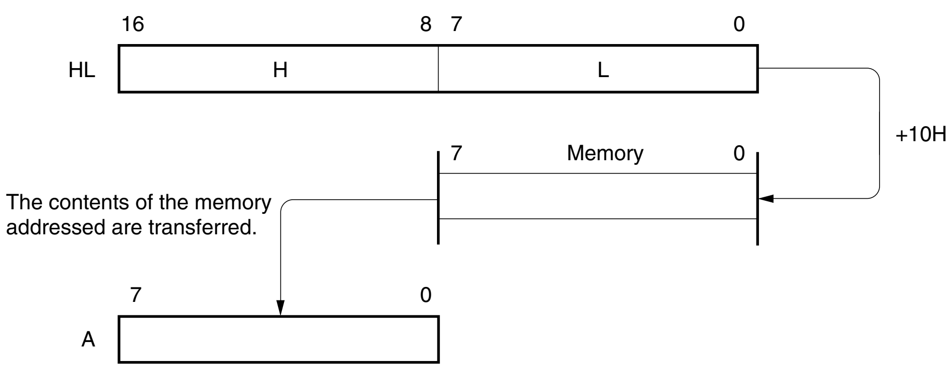
Identifier	Description
–	[HL + byte]

[Description example]

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



[Illustration]



3.4.8 Based indexed addressing

[Function]

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

[Operand format]

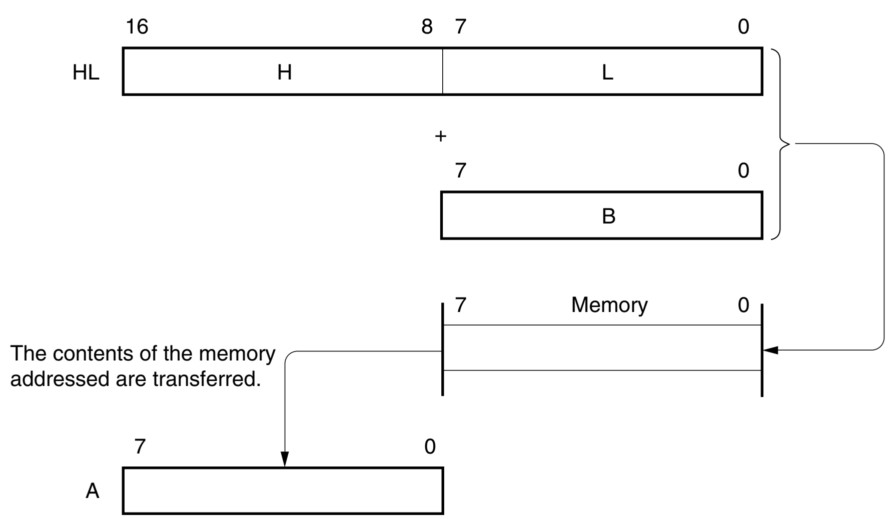
Identifier	Description
–	[HL + B], [HL + C]

[Description example]

MOV A, [HL +B]; when selecting B register

Operation code	1 0 1 0 1 0 1 1

[Illustration]



3.4.9 Stack addressing

[Function]

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

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

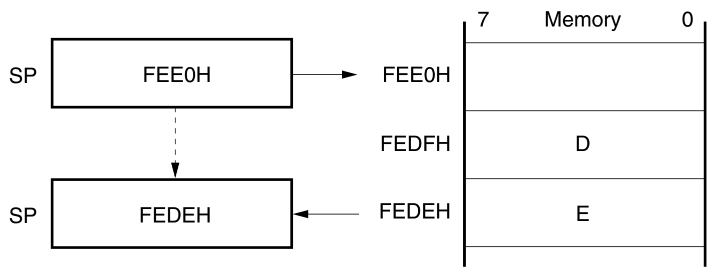
With stack addressing, only the internal high-speed RAM area can be accessed.

[Description example]

PUSH DE; when saving DE register

Operation code	1 0 1 1 0 1 0 1

[Illustration]



CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

There are two types of pin I/O buffer power supplies: AV_{REF}[Note] and V_{DD}. The relationship between these power supplies and the pins is shown below.

Table 4-1. Pin I/O Buffer Power Supplies

Power Supply	Corresponding Pins
AV_{REF}[Note]	P20 to P27
V_{DD}	Port pins other than P20 to P27

Note μPD78F048x and 78F049x only. The power supply is V_{DD} with μPD78F047x.

78K0/LF3 products are provided with the ports shown in Figure 4-1, which enable variety of control operations. The functions of each port are shown in Table 4-2.

In addition to the function as digital I/O ports, these ports have several alternate functions. For details of the alternate functions, see **CHAPTER 2 PIN FUNCTIONS**.

Figure 4-1. Port Types

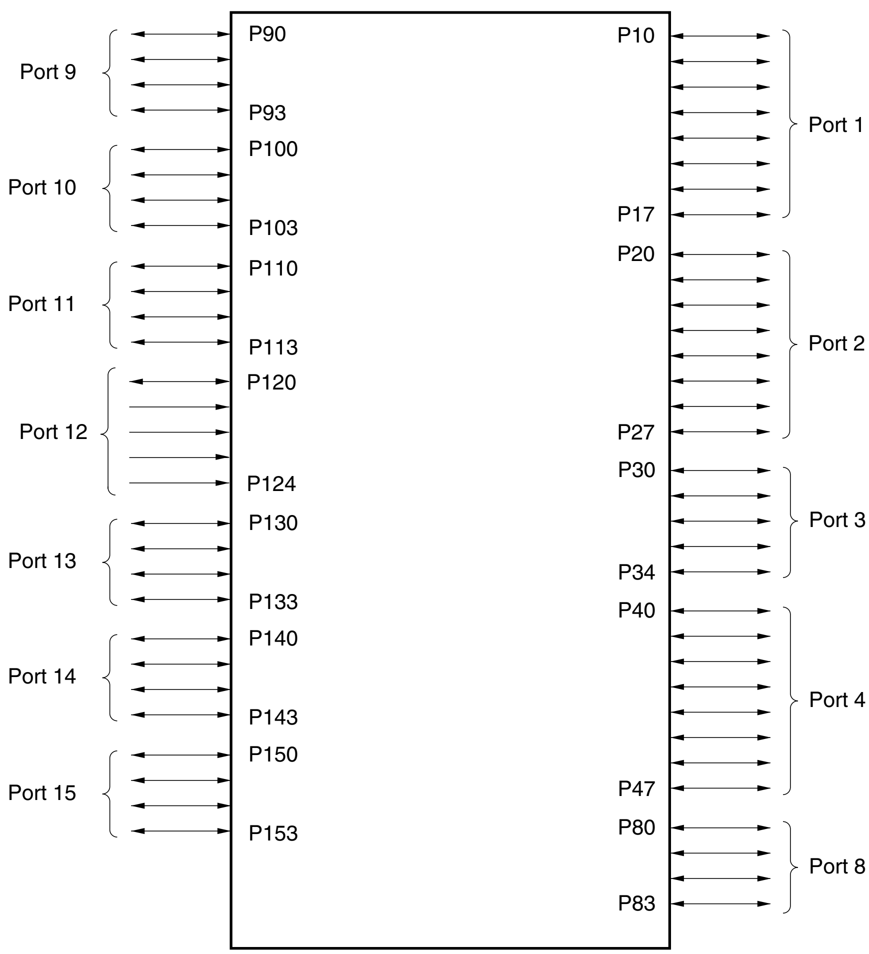


Table 4-2. Port Functions (1/2)

Function Name	I/O	Function	After Reset	Alternate Function
P10	I/O	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	PCL
P11				$\overline{\text{SCK10}}$
P12				SI10/RxD0
P13				SO10/TxD0
P14				$\overline{\text{SCKA0}}$/INTP4
P15				SIA0/<RxD6>
P16				SOA0/<TxD6>
P17				−
P20	I/O	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	SEG39[Note1]/ANI0[Note2]/DS0−[Note3]
P21				SEG38[Note1]/ANI1[Note2]/DS0+[Note3]
P22				SEG37[Note1]/ANI2[Note2]/DS1−[Note3]
P23				SEG36[Note1]/ANI3[Note2]/DS1+[Note3]
P24				SEG35[Note1]/ANI4[Note2]/DS2−[Note3]
P25				SEG34[Note1]/ANI5[Note2]/DS2+[Note3]
P26				SEG33[Note1]/ANI6[Note2]/REF−[Note3]
P27				SEG32[Note1]/ANI7[Note2]/REF+[Note3]
P30	I/O	Port 3. 5-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP5
P31				TOH1/INTP3
P32				TOH0/MCGO
P33				TI000/RTCDIV/RTCCL/BUZ/INTP2
P34				TI52/TI010/TO00/RTC1HZ/INTP1
P40	I/O	Port 4. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	V_{LC3}/KR0
P41				RIN/KR1
P42				KR2
P43				TO51/TI51/KR3
P44				TO50/TI50/KR4
P45 to P47				KR5 to KR7

Notes **1.** μPD78F047x and 78F048x only.
2. μPD78F048x and 78F049x only.
3. μPD78F049x only.

Remark The functions within arrowheads (< >) can be assigned by setting the input switch control register (ISC).

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

Function Name	I/O	Function	After Reset	Alternate Function
P80 to P83	I/O	Port 8. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SEG4 to SEG7
P90 to P93	I/O	Port 9. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SEG8 to SEG11
P100 to P103	I/O	Port 10. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SEG12 to SEG15
P110, P111	I/O	Port 11. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SEG16, SEG17
P112				SEG18/TxD6
P113				SEG19/RxD6
P120	I/O	Port 12. 1-bit I/O port and 4-bit input port. Only for P120, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP0/EXLVI
P121	Input			X1/OCD0A
P122				X2/EXCLK/OCD0B
P123				XT1
P124				XT2
P130 to P133	I/O	Port 13. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SEG20 to SEG23
P140 to P143	I/O	Port 14. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SEG24 (KS0) to SEG27 (KS3)
P150 to P153	I/O	Port 15. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SEG28 (KS4) to SEG31 (KS7)

4.2 Port Configuration

Ports include the following hardware.

Table 4-3. Port Configuration

Item	Configuration
Control registers	Port mode register (PM1 to PM4, PM8 to PM15) Port register (P1 to P4, P8 to P15) Pull-up resistor option register (PU1, PU3, PU4, PU8 to PU15) Port function register 1 (PF1) Port function register 2 (PF2)[Note 1] Port function register ALL (PFALL) A/D port configuration register 0 (ADPC0)[Note 2]
Port	Total: 62
Pull-up resistor	Total: 50

Notes 1. μPD78F047x and 78F048x only

2. μPD78F048x and 78F049x only

4.2.1 Port 1

Port 1 is an 8-bit I/O port with an output latch. Port 1 can be set to the input mode or output mode in 1-bit units using port mode register 1 (PM1). When the P10 to P17 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 1 (PU1).

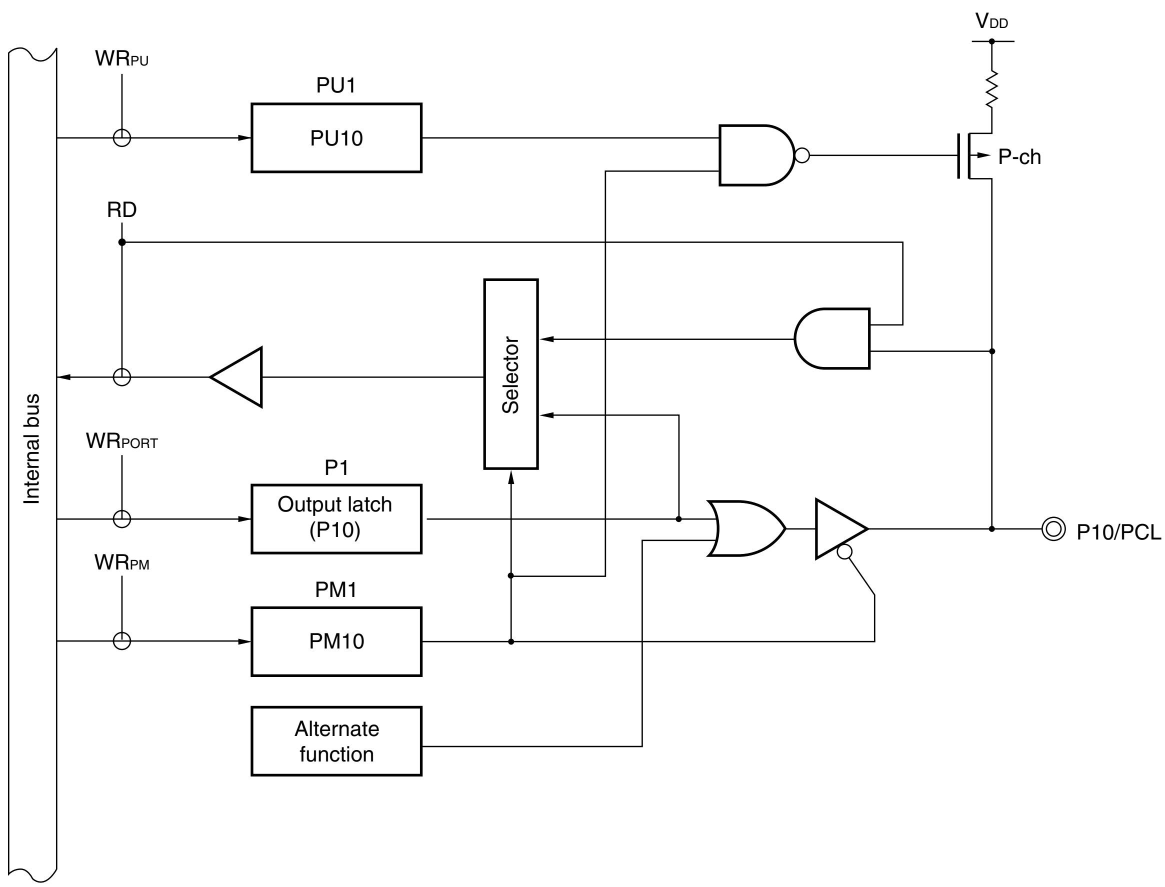
This port can also be used for serial clock I/O, serial interface data I/O, and maskable external interrupt input.

Reset signal generation sets port 1 to input mode.

Figures 4-2 to 4-7 show block diagrams of port 1.

Caution To use P11/$\overline{\text{SCK10}}$, P12/SI10/RxD0, and P13/SO10/TxD0 as general-purpose ports, set serial operation mode register 10 (CSIM10) and serial clock selection register 10 (CSIC10) to the default status (00H).

Figure 4-2. Block Diagram of P10



P1: Port register 1

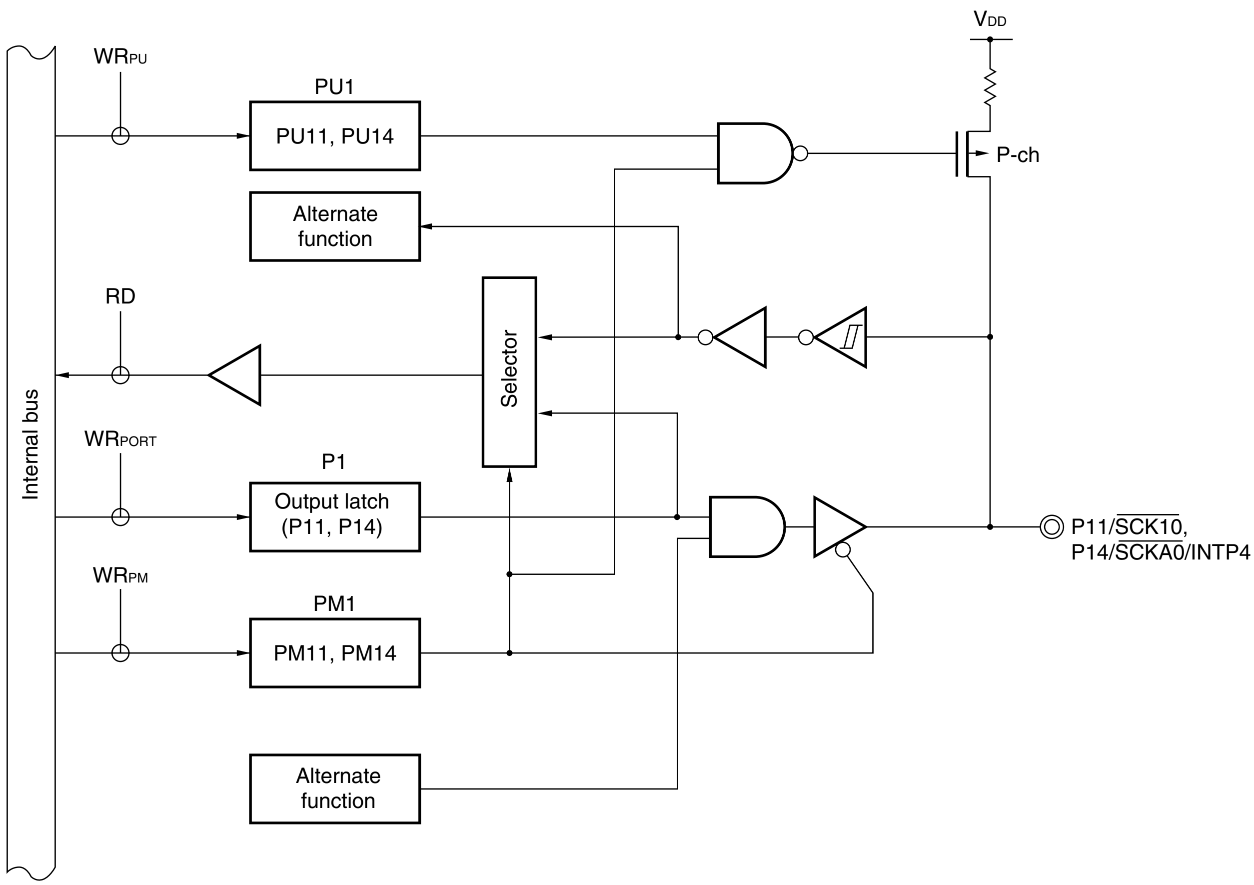
PU1: Pull-up resistor option register 1

PM1: Port mode register 1

RD: Read signal

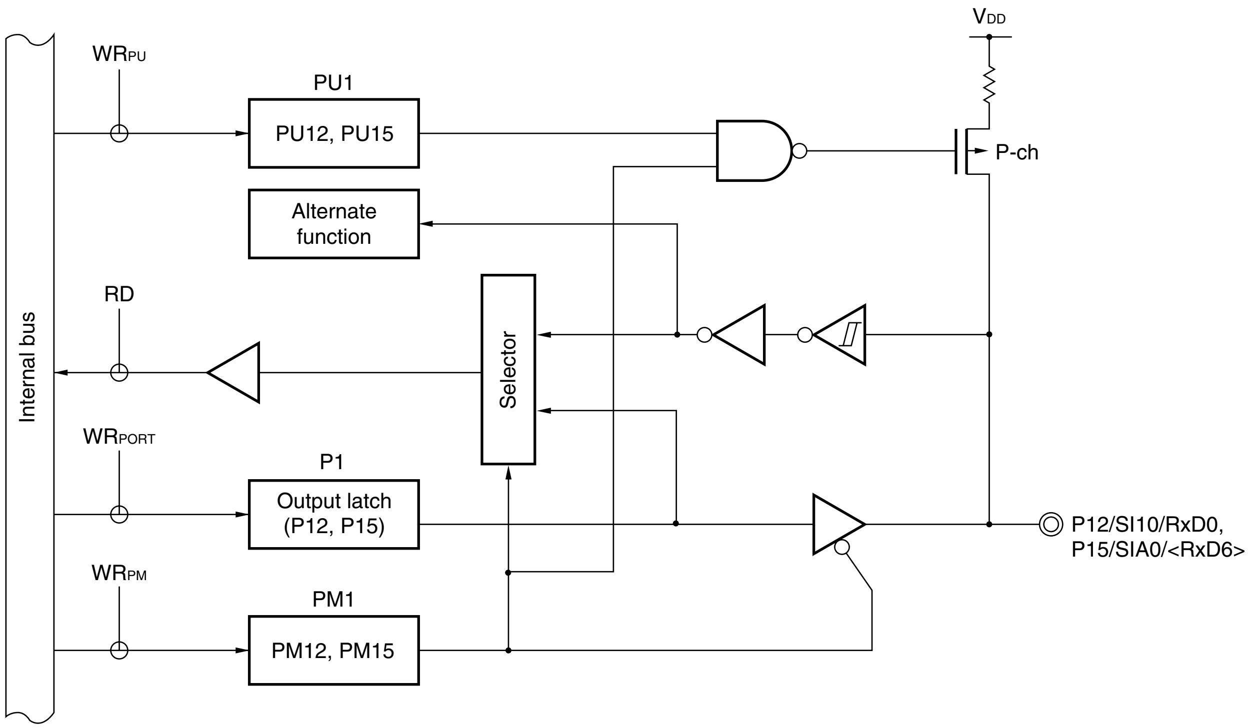
WR××: Write signal

Figure 4-3. Block Diagram of P11 and P14



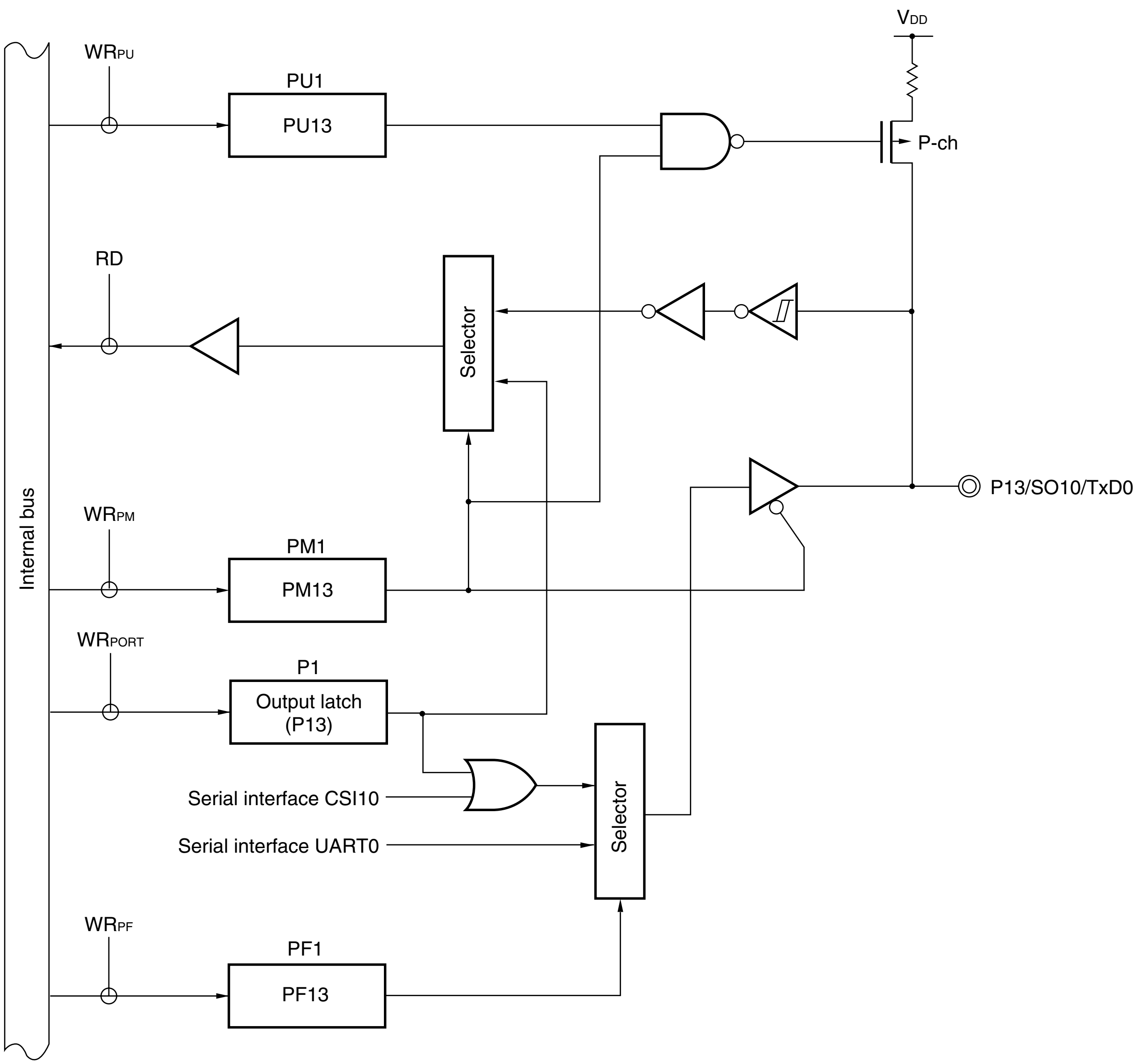
P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
RD: Read signal
WR××: Write signal

Figure 4-4. Block Diagram of P12 and P15



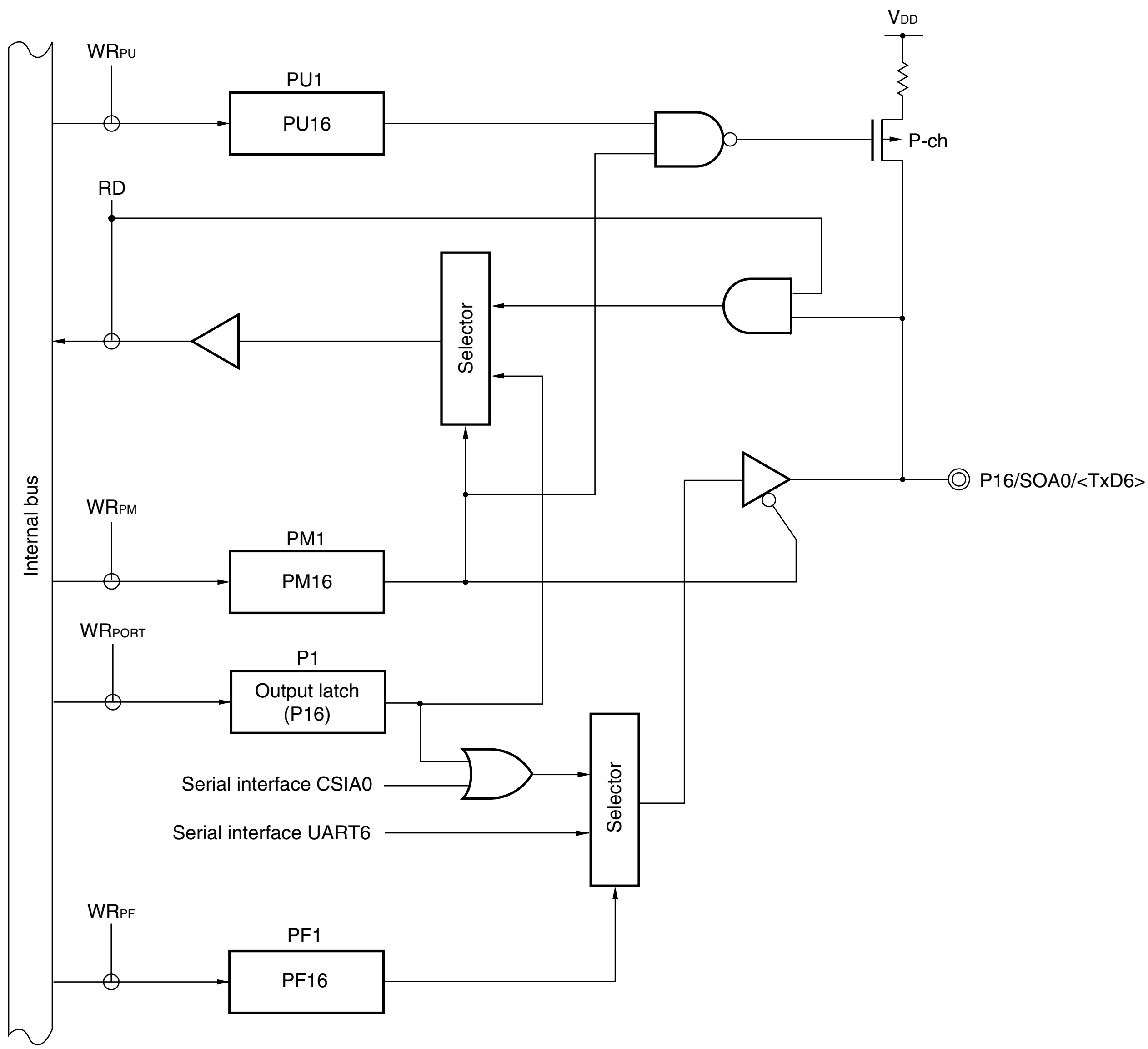
P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
RD: Read signal
WR××: Write signal

Figure 4-5. Block Diagram of P13



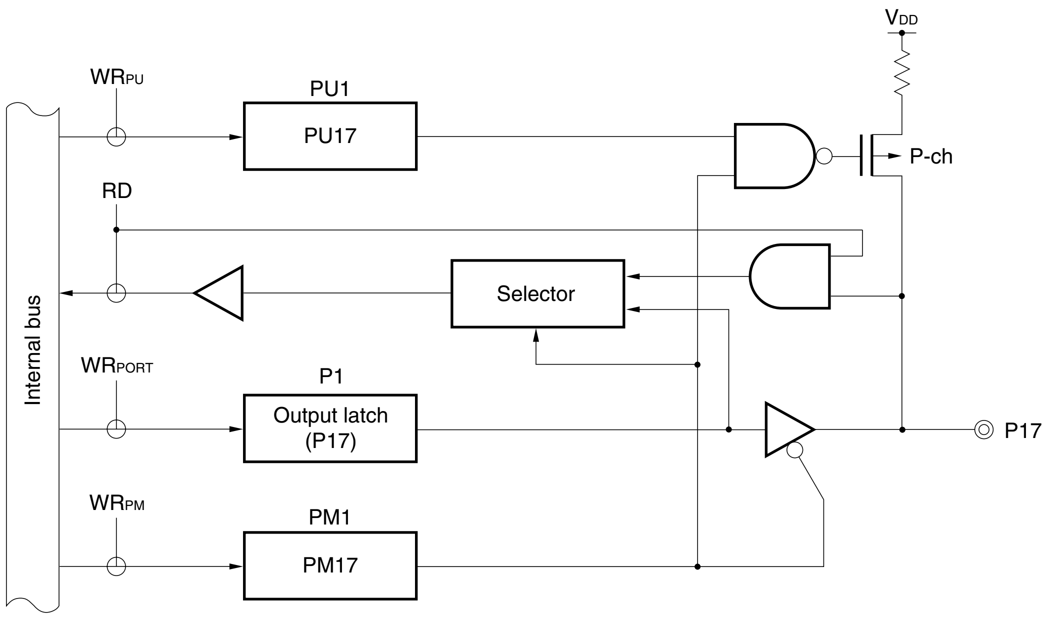
P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
PF1: Port function register 1
RD: Read signal
WR××: Write signal

Figure 4-6. Block Diagram of P16



P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
PF1: Port function register 1
RD: Read signal
WR××: Write signal

Figure 4-7. Block Diagram of P17



P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
RD: Read signal
WR××: Write signal

4.2.2 Port 2

Port 2 is an 8-bit I/O port with an output latch. Port 2 can be set to the input mode or output mode in 1-bit units using port mode register 2 (PM2).

This port can also be used for 10-bit successive approximation type A/D converter, 16-bit ΔΣ type A/D converter analog input, and segment output.

To use P20/ANI0/DS0−, P21/ANI1/DS0+, P22/ANI2/DS1−, P23/ANI3/DS1+, P24/ANI4/DS2−, P25/ANI5/DS2+, P26/ANI6/REF−, and P27/ANI7/REF+ as digital input pins, set them to port function (other than segment output) by using the port function register 2 (PF2), to digital I/O by using ADPC0, and to input mode by using PM2. Use these pins starting from the lower bit.

P20/ANI0/DS0−, P21/ANI1/DS0+, P22/ANI2/DS1−, P23/ANI3/DS1+, P24/ANI4/DS2−, P25/ANI5/DS2+, P26/ANI6/REF−, and P27/ANI7/REF+ as digital output pins, set them to port function (other than segment output) by using the port function register 2 (PF2), to digital I/O by using ADPC0, and to output mode by using PM2. Use these pins starting from the lower bit.

Reset signal generation sets port 1 to input mode.

Figure 4-8 shows block diagrams of port 2.

Table 4-4. Setting Functions of P20/SEG39[Note 1]/ANI0[Note 2]/DS0−[Note 3] to P27/SEG32[Note 1]/ANI7[Note 2]/REF+[Note 3] Pins

PF2	ADPC0	PM2	ADS	ADDCTL0	P20/SEG39[Note 1]/ANI0[Note 2]/DS0−[Note 3] to P27/SEG32[Note 1]/ANI7[Note 2]/REF+[Note 3] Pins
Digital/Analog selection	Analog input selection	Input mode	Does not select ANI.	Does not select DSn±.	Analog input (not to be converted)
			Selects ANI.	Does not select DSn±.	Analog input (to be converted by successive approximation type A/D converter)
			Does not select ANI.	Selects DSn±.	Analog input (to be converted by ΔΣ type A/D converter)
			Selects ANI.	Selects DSn±.	Setting prohibited
		Output mode	−		Setting prohibited
	Digital I/O selection	Input mode	−		Digital input
		Output mode	−		Digital output
SEG output selection[Note 1]	−	−	−		Segment output[Note 1]

Notes 1. μPD78F047x and 78F048x only.

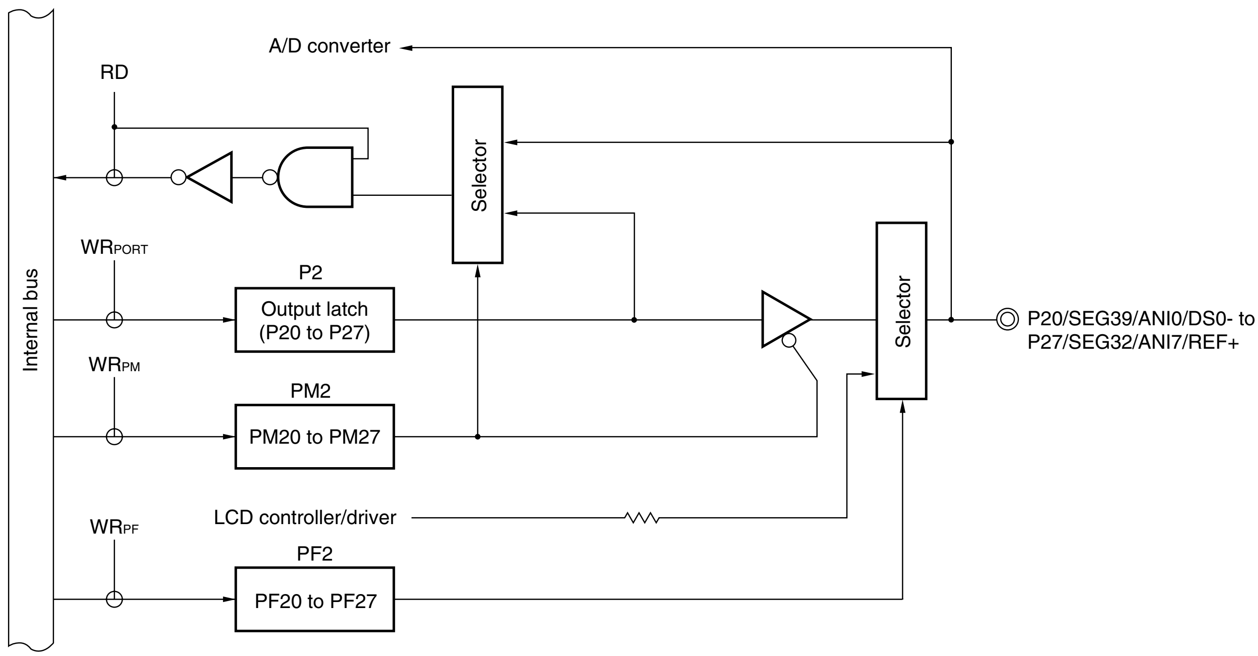
2. μPD78F048x and 78F049x only.

3. μPD78F049x only.

Remark n = 0 to 2

<R>

Figure 4-8. Block Diagram of P20 to P27



P2: Port register 2
PM2: Port mode register 2
PF2: Port function register 2
RD: Read signal
WR××: Write signal

4.2.3 Port 3

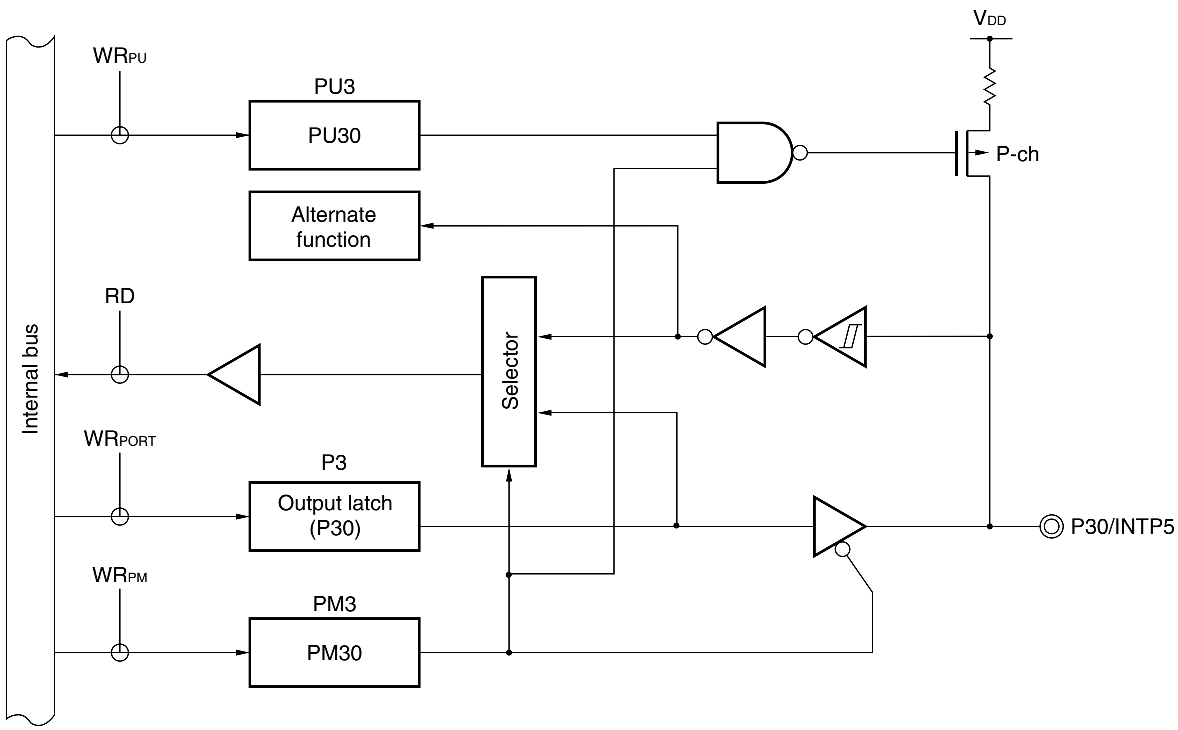
Port 3 is a 5-bit I/O port with an output latch. Port 3 can be set to the input mode or output mode in 1-bit units using port mode register 3 (PM3). When the P30 to P34 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 3 (PU3).

This port can also be used for external interrupt request input, timer I/O, manchester code generator output, real-time counter output, and buzzer output.

Reset signal generation sets port 3 to input mode.

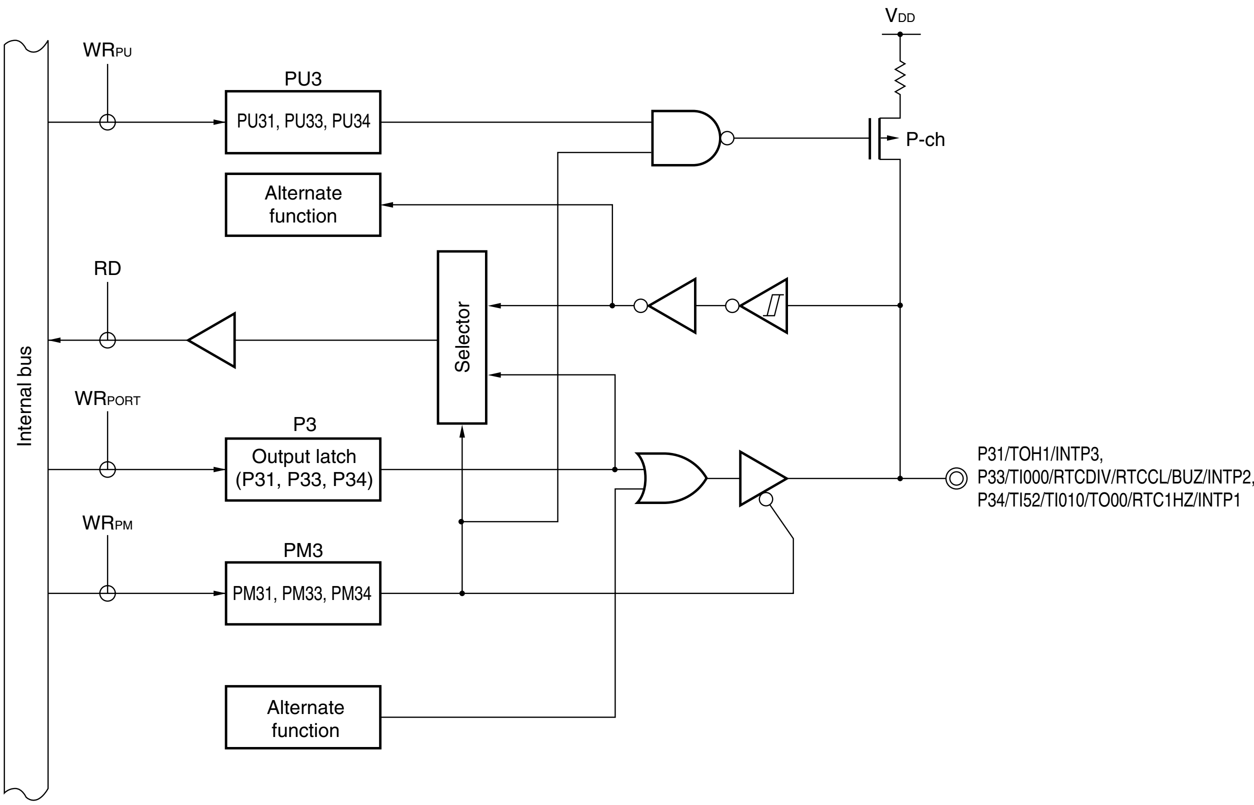
Figures 4-9 and 4-11 show block diagrams of port 3.

Figure 4-9. Block Diagram of P30



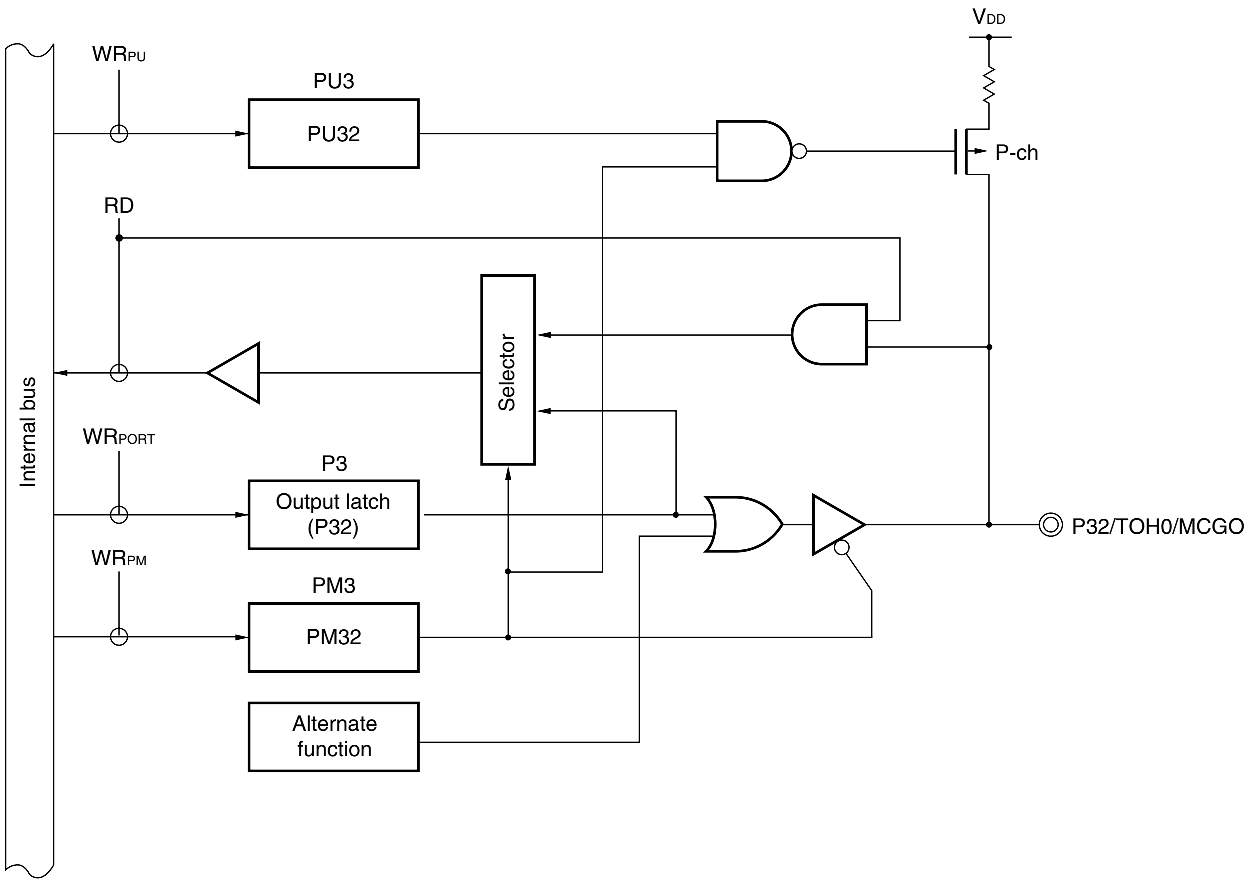
P3: Port register 3
PU3: Pull-up resistor option register 3
PM3: Port mode register 3
RD: Read signal
WR××: Write signal

Figure 4-10. Block Diagram of P31, P33, P34



P3: Port register 3
PU3: Pull-up resistor option register 3
PM3: Port mode register 3
RD: Read signal
WR××: Write signal

Figure 4-11. Block Diagram of P32



P3: Port register 3
PU3: Pull-up resistor option register 3
PM3: Port mode register 3
RD: Read signal
WR××: Write signal

4.2.4 Port 4

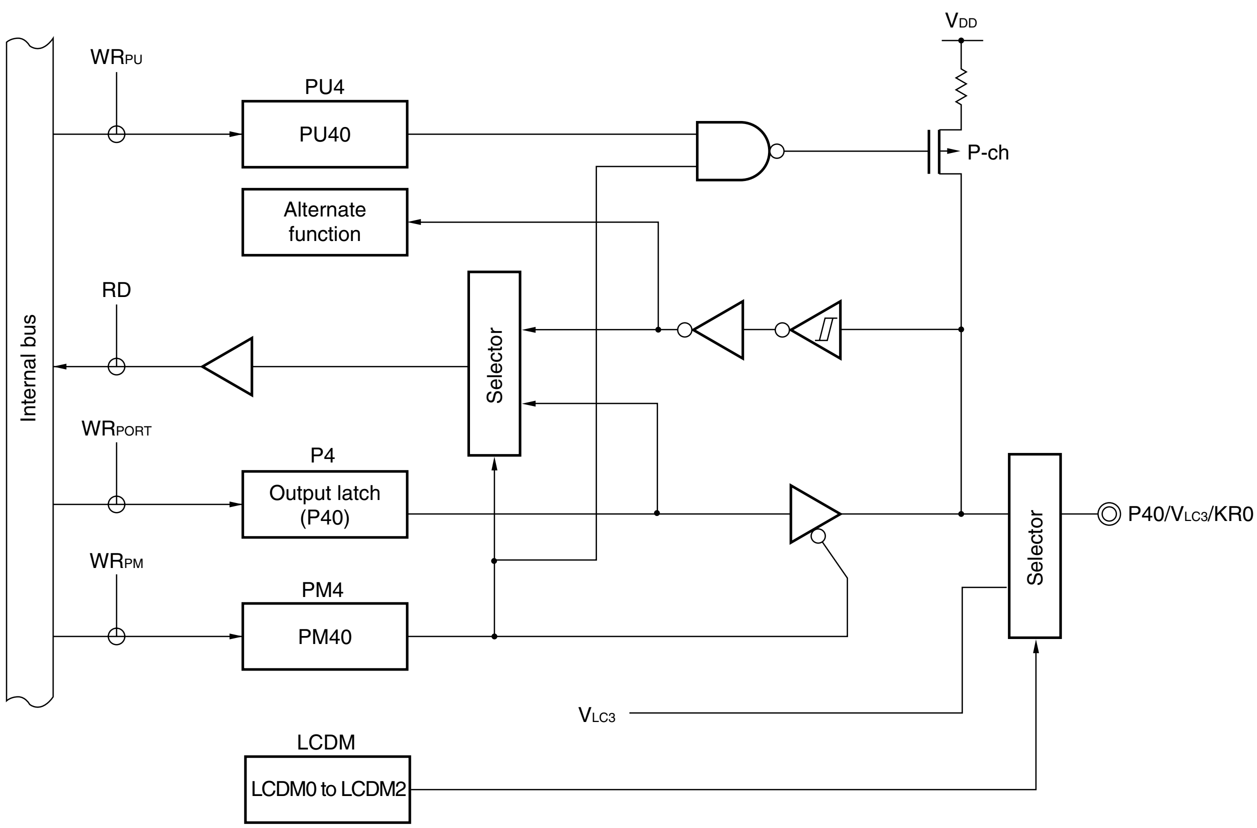
Port 4 is an 8-bit I/O port with an output latch. Port 4 can be set to the input mode or output mode in 1-bit units using port mode register 4 (PM4). When the P40 to P47 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 4 (PU4).

This port can also be used for key interrupt input, segment key scan input, timer I/O, remote control receive data input, and power supply voltage for driving the LCD.

Reset signal generation sets port 4 to input mode.

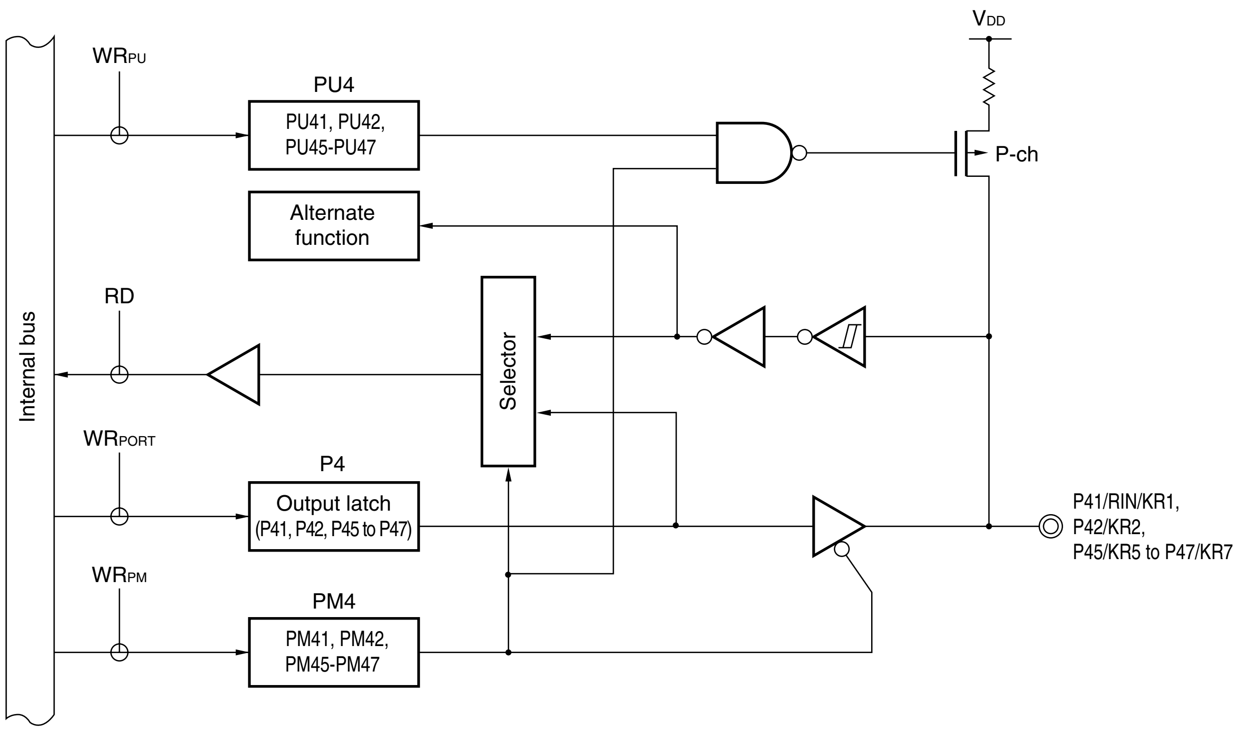
Figures 4-12 to 4-14 show a block diagram of port 4.

Figure 4-12. Block Diagram of P40



P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
LCDM: LCD display mode register
RD: Read signal
WR××: Write signal

Figure 4-13. Block Diagram of P41, P42, P45 to P47



P4: Port register 4

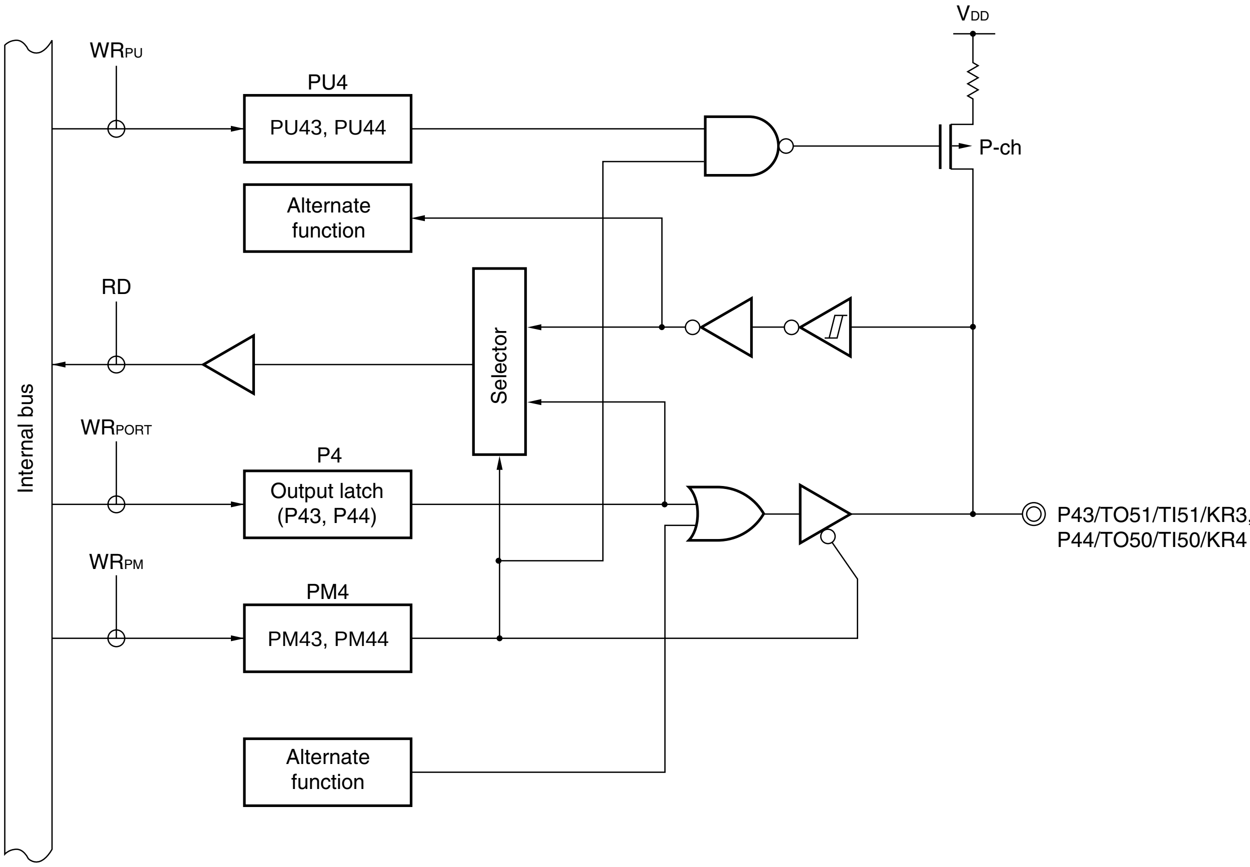
PU4: Pull-up resistor option register 4

PM4: Port mode register 4

RD: Read signal

WR××: Write signal

Figure 4-14. Block Diagram of P43 and P44



P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
RD: Read signal
WR××: Write signal

4.2.5 Port 8

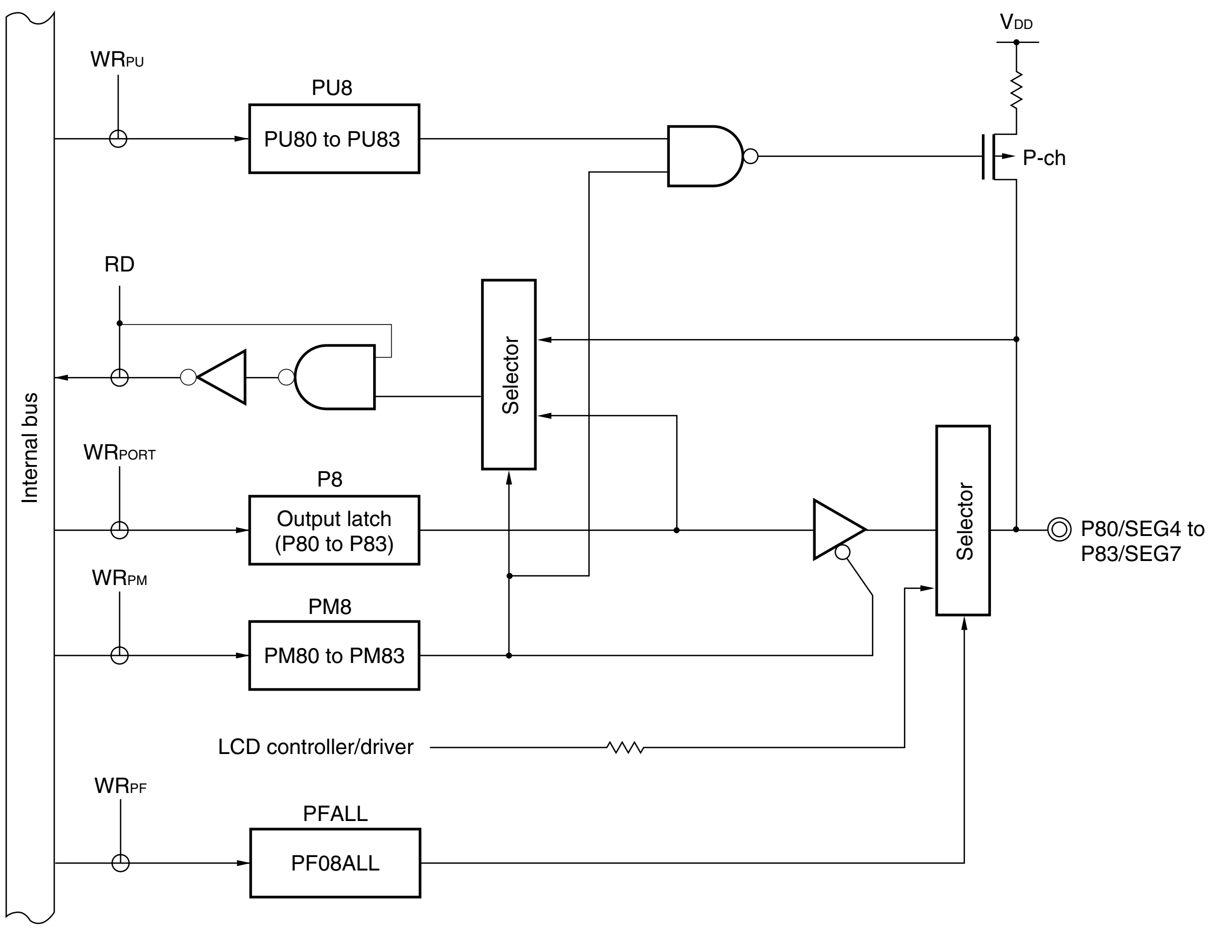
Port 8 is a 4-bit I/O port with an output latch. Port 8 can be set to the input mode or output mode in 1-bit units using port mode register 8 (PM8). When the P80 to P83 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 8 (PU8).

This port can also be used for segment output.

Reset signal generation sets port 8 to input mode.

Figure 4-15 shows a block diagram of port 8.

Figure 4-15. Block Diagram of P80 to P83



P8: Port register 8
PU8: Pull-up resistor option register 8
PM8: Port mode register 8
PFALL: Port function register ALL
RD: Read signal
WR××: Write signal

4.2.6 Port 9

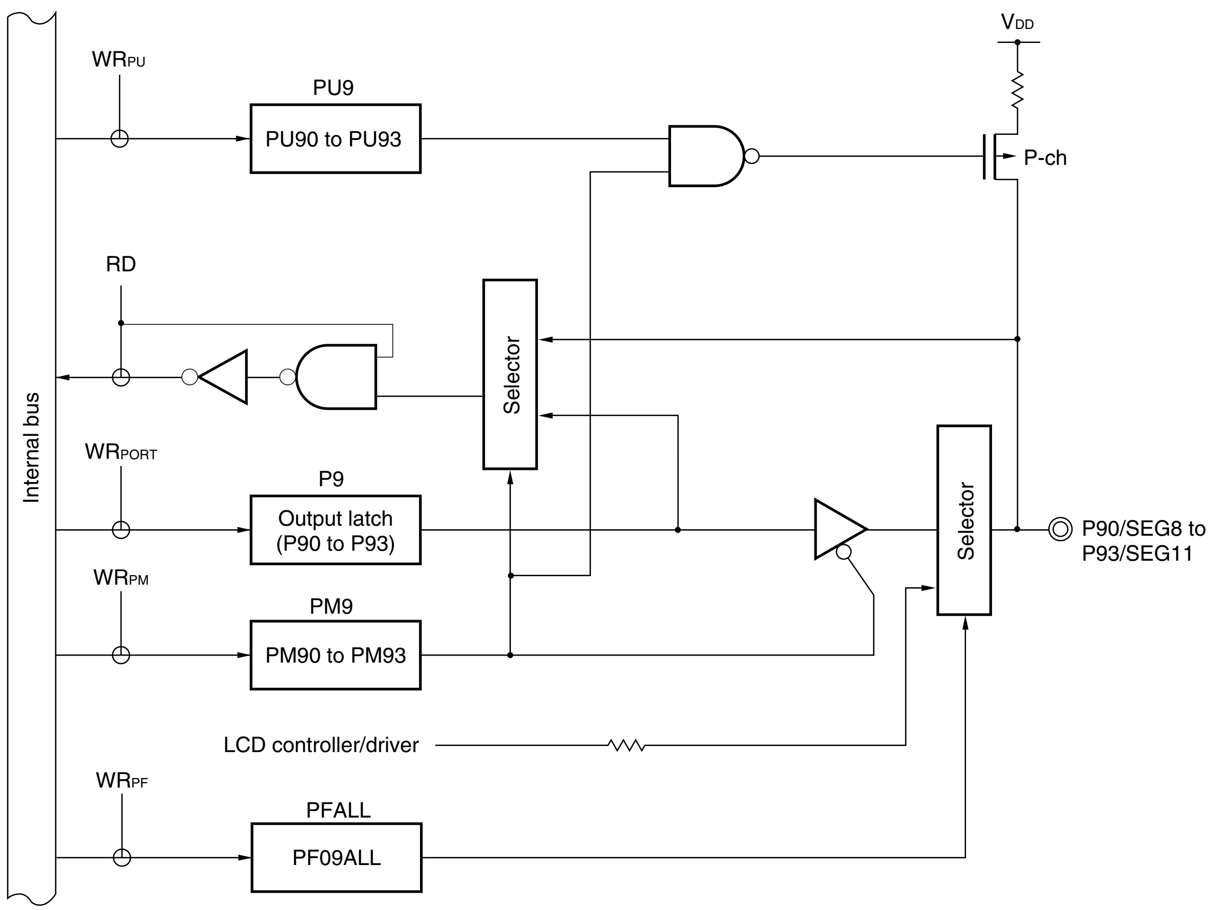
Port 9 is a 4-bit I/O port with an output latch. Port 9 can be set to the input mode or output mode in 1-bit units using port mode register 9 (PM9). When the P90 to P93 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 9 (PU9).

This port can also be used for segment output.

Reset signal generation sets port 9 to input mode.

Figure 4-16 shows block diagrams of port 9.

Figure 4-16. Block Diagram of P90 to P93



P9: Port register 9
PU9: Pull-up resistor option register 9
PM9: Port mode register 9
PFALL: Port function register ALL
RD: Read signal
WR××: Write signal

4.2.7 Port 10

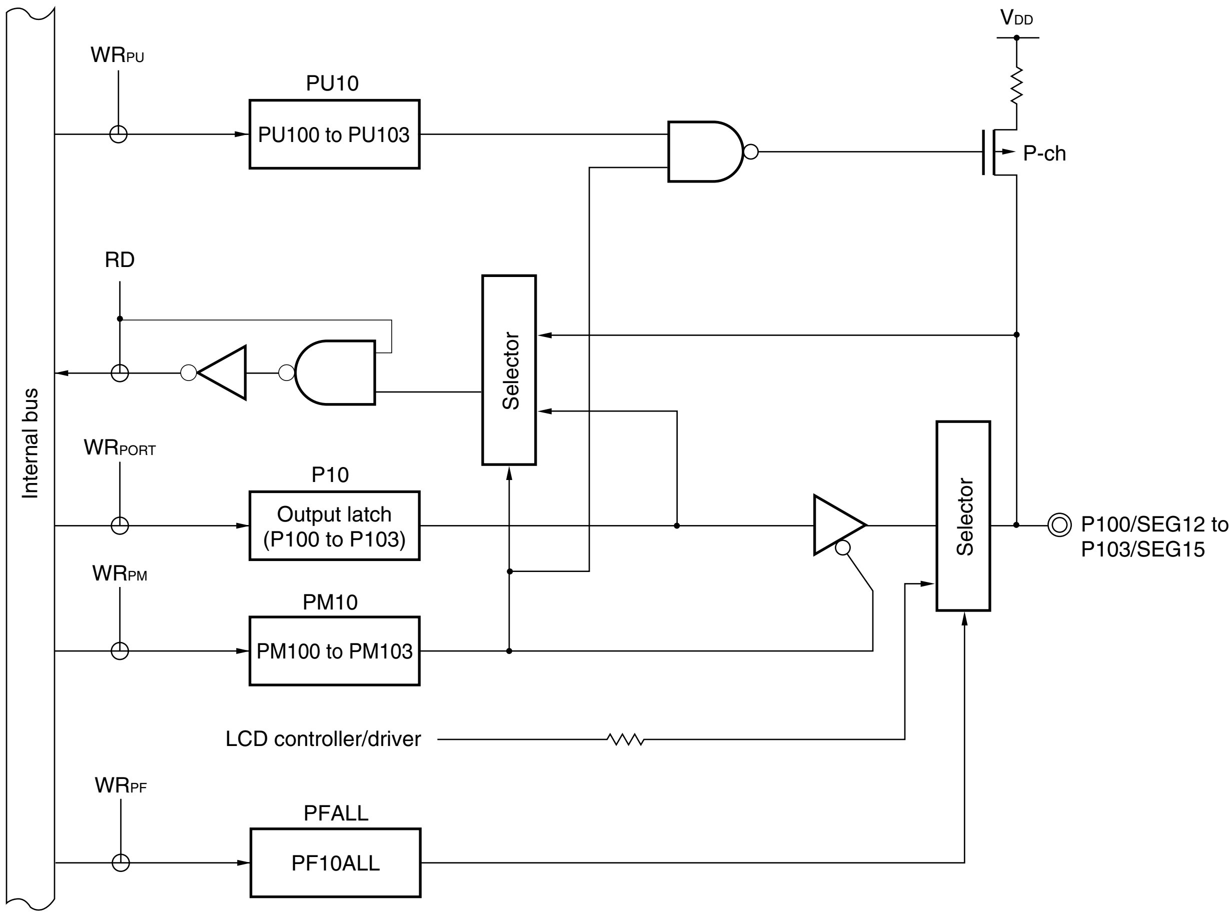
Port 10 is a 4-bit I/O port with an output latch. Port 10 can be set to the input mode or output mode in 1-bit units using port mode register 10 (PM10). When the P100 to P103 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 10 (PU10).

This port can also be used for segment output.

Reset signal generation sets port 10 to input mode.

Figure 4-17 shows a block diagram of port 10.

Figure 4-17. Block Diagram of P100 to P103



P10: Port register 10
PU10: Pull-up resistor option register 10
PM10: Port mode register 10
PFALL: Port function register ALL
RD: Read signal
WR××: Write signal

4.2.8 Port 11

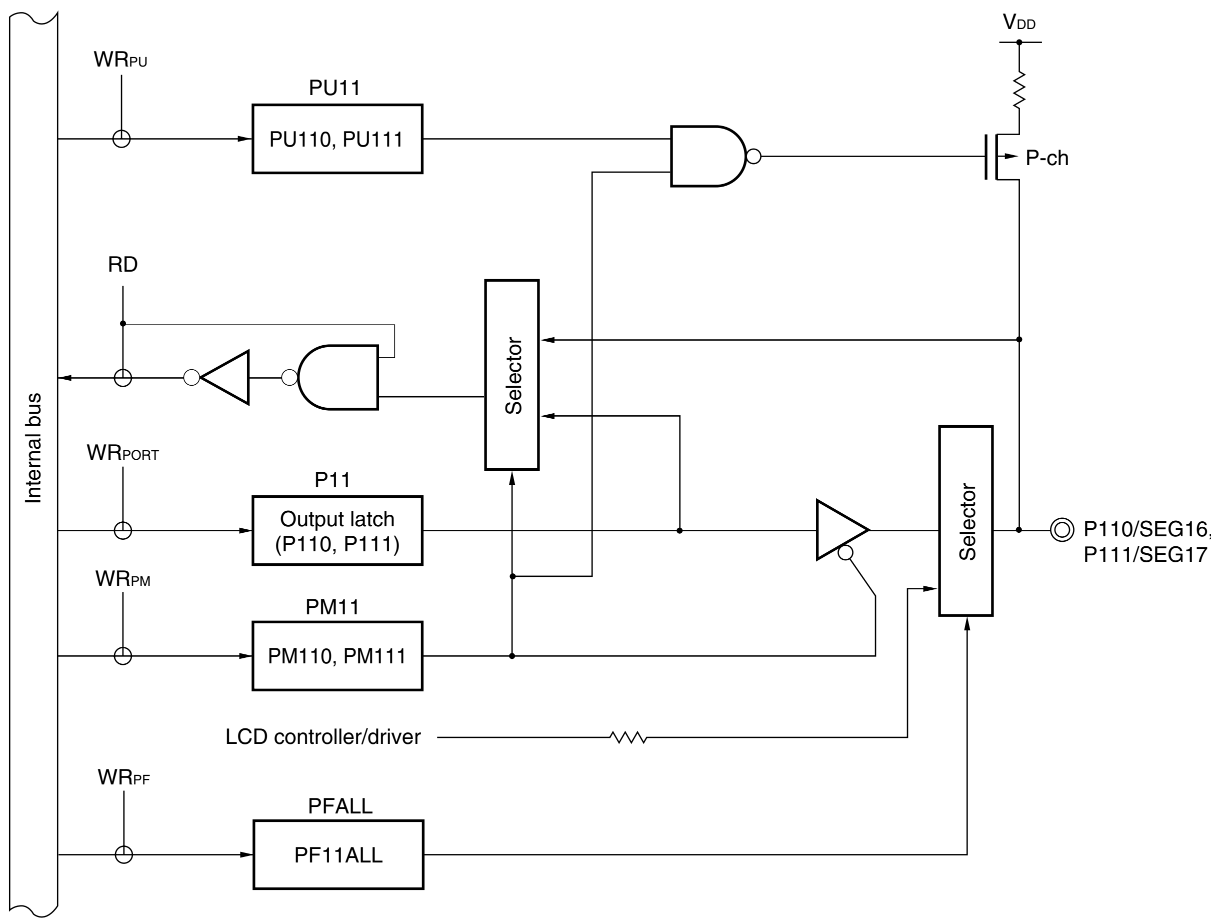
Port 11 is a 4-bit I/O port with an output latch. Port 11 can be set to the input mode or output mode in 1-bit units using port mode register 11 (PM11). When the P110 to P113 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 11 (PU11).

This port can also be used for segment output and serial interface data I/O.

Reset signal generation sets port 11 to input mode.

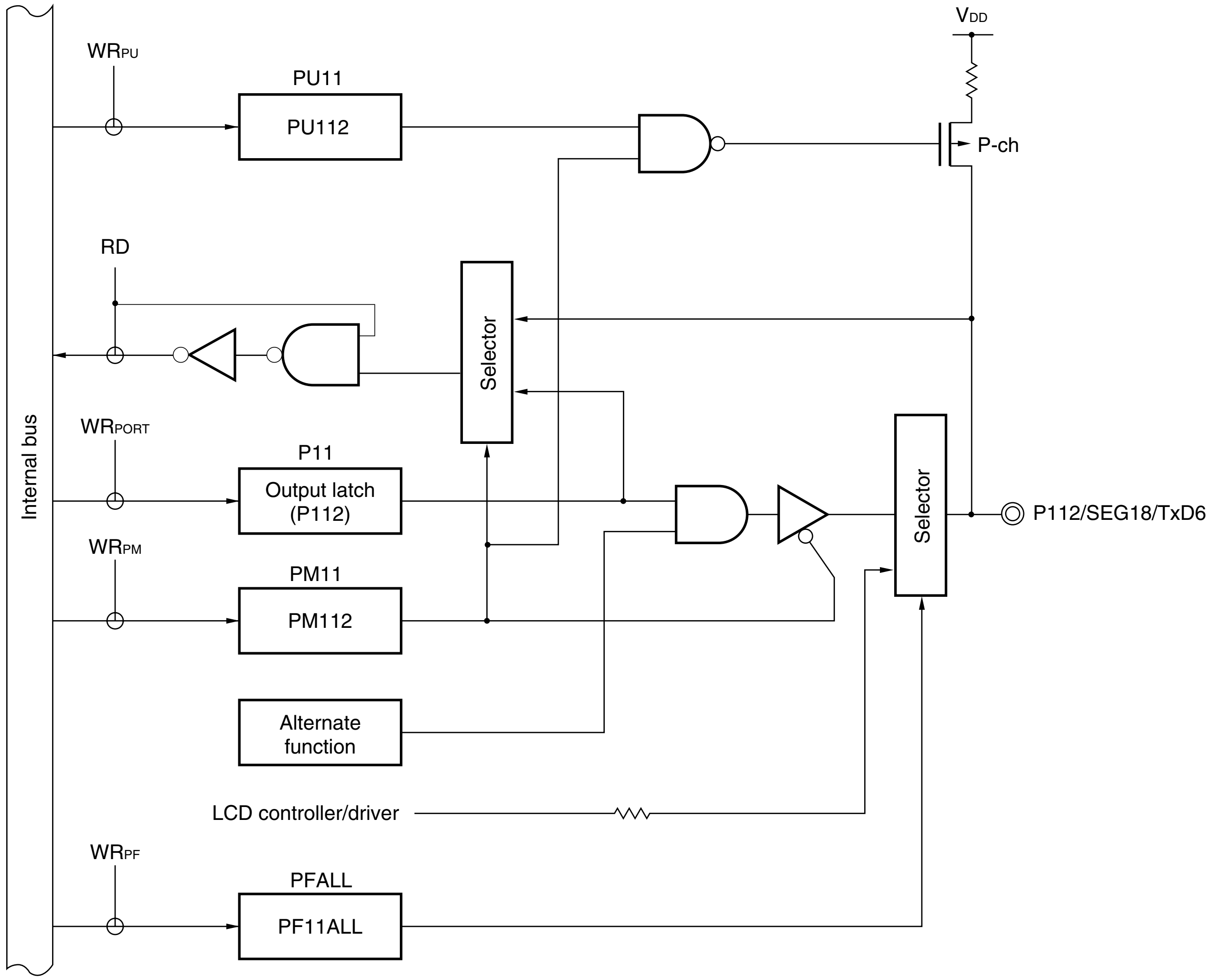
Figures 4-18 to 4-20 show a block diagram of port 11.

Figure 4-18. Block Diagram of P110 and P111



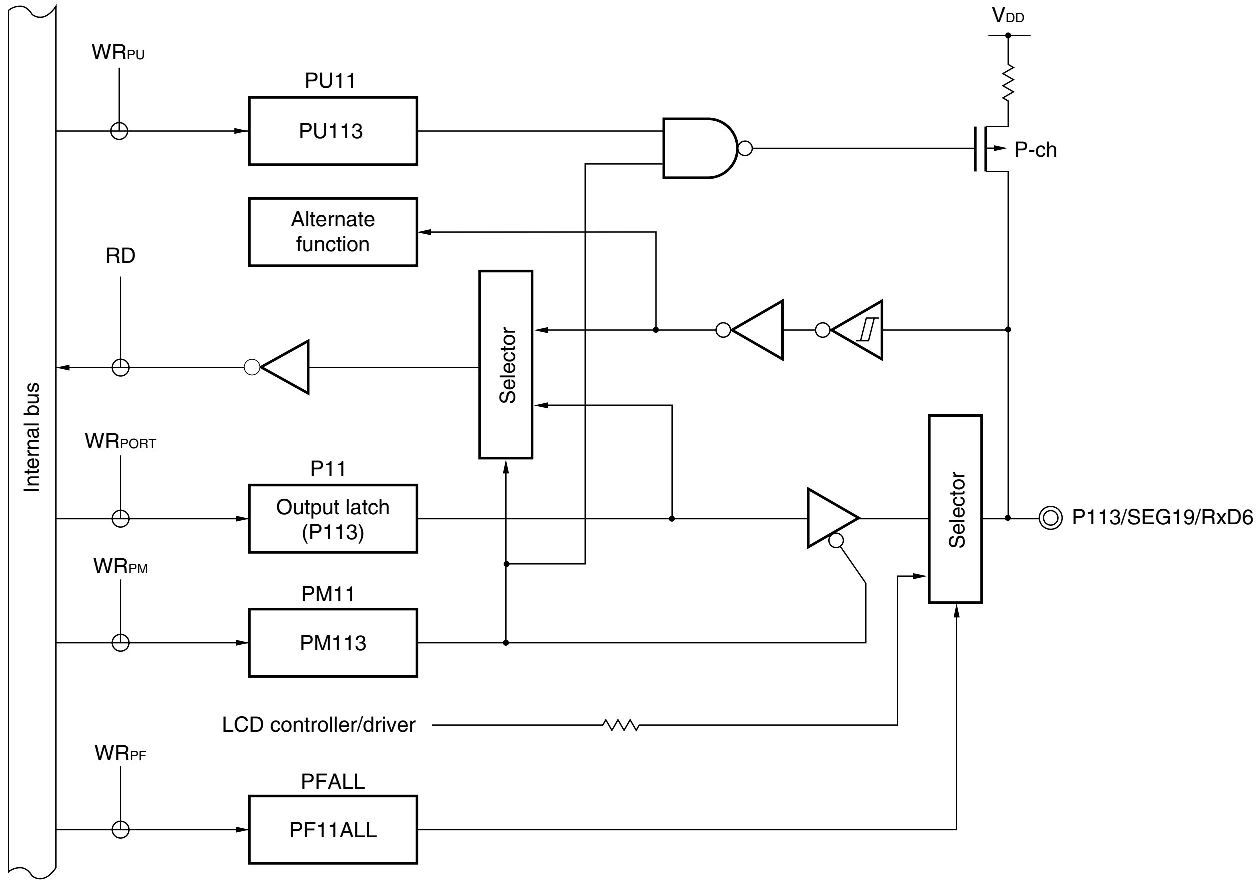
P11: Port register 11
PU11: Pull-up resistor option register 11
PM11: Port mode register 11
PFALL: Port function register ALL
RD: Read signal
WR××: Write signal

Figure 4-19. Block Diagram of P112



P11: Port register 11
PU11: Pull-up resistor option register 11
PM11: Port mode register 11
PFALL: Port function register ALL
RD: Read signal
WR××: Write signal

Figure 4-20. Block Diagram of P113



P11: Port register 11
PU11: Pull-up resistor option register 11
PM11: Port mode register 11
PFALL: Port function register ALL
RD: Read signal
WR××: Write signal

4.2.9 Port 12

Port 12 is a 1-bit I/O port with an output latch and a 4-bit input port. Only P120 can be set to the input mode or output mode in 1-bit units using port mode register 12 (PM12). When used as an input port only for P120, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

This port can also be used as pins for external interrupt request input, potential input for external low-voltage detection, connecting resonator for main system clock, connecting resonator for subsystem clock, and external clock input for main system clock.

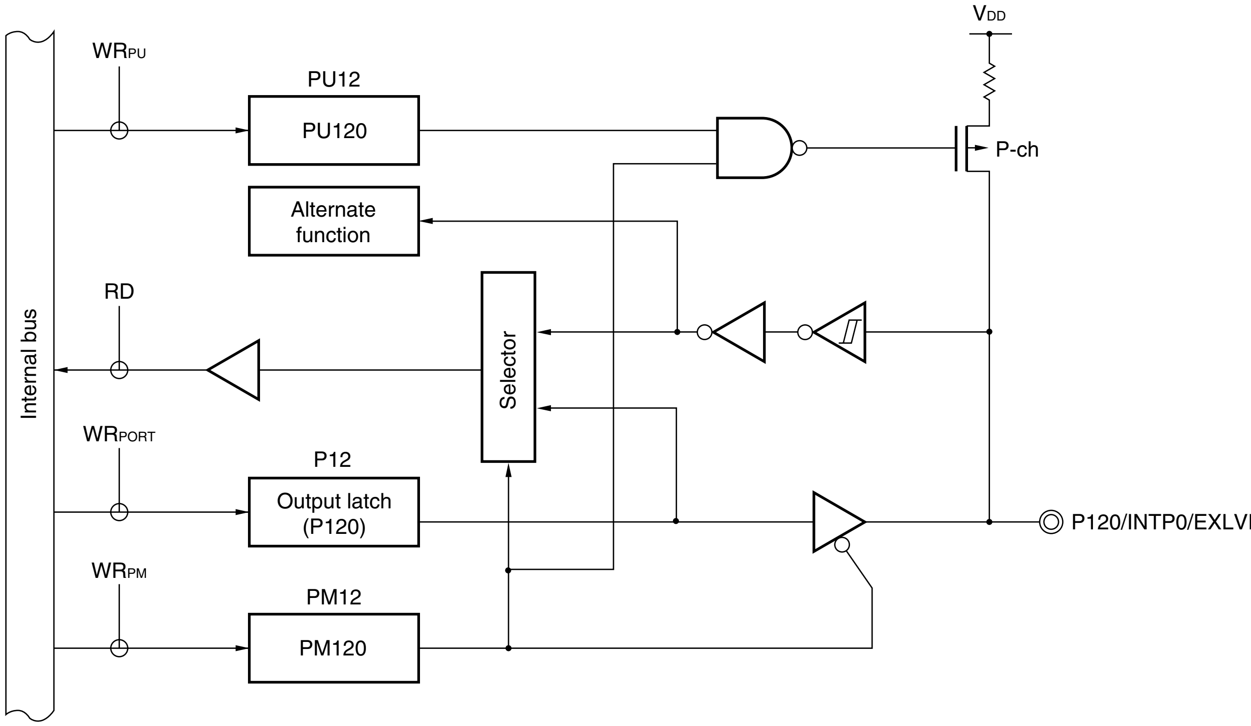
Reset signal generation sets port 12 to input mode.

Figures 4-21 to 4-23 show block diagrams of port 12.

Caution When using the P121 to P124 pins to connect a resonator for the main system clock (X1, X2) or subsystem clock (XT1, XT2), or to input an external clock for the main system clock (EXCLK), the X1 oscillation mode, XT1 oscillation mode, or external clock input mode must be set by using the clock operation mode select register (OSCCTL) (for details, see 5.3 (1) Clock operation mode select register (OSCCTL) and (3) Setting of operation mode for subsystem clock pin). The reset value of OSCCTL is 00H (all of the P121 to P124 pins are input port pins).

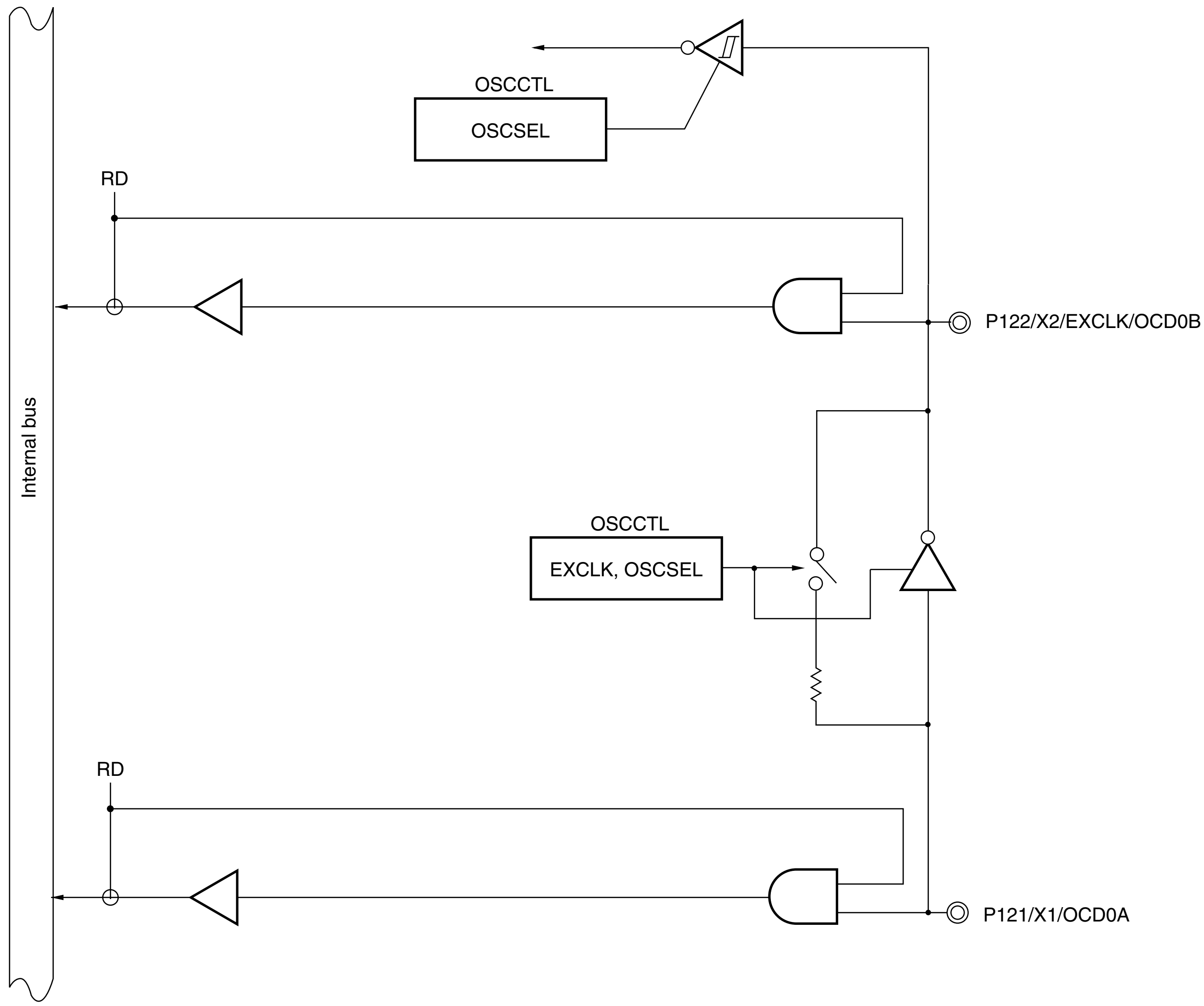
Remark P121 and P122 can be used as on-chip debug mode setting pins (OCD0A, OCD0B) when the on-chip debug function is used. For detail, see **CHAPTER 29 ON-CHIP DEBUG FUNCTION.**

Figure 4-21. Block Diagram of P120



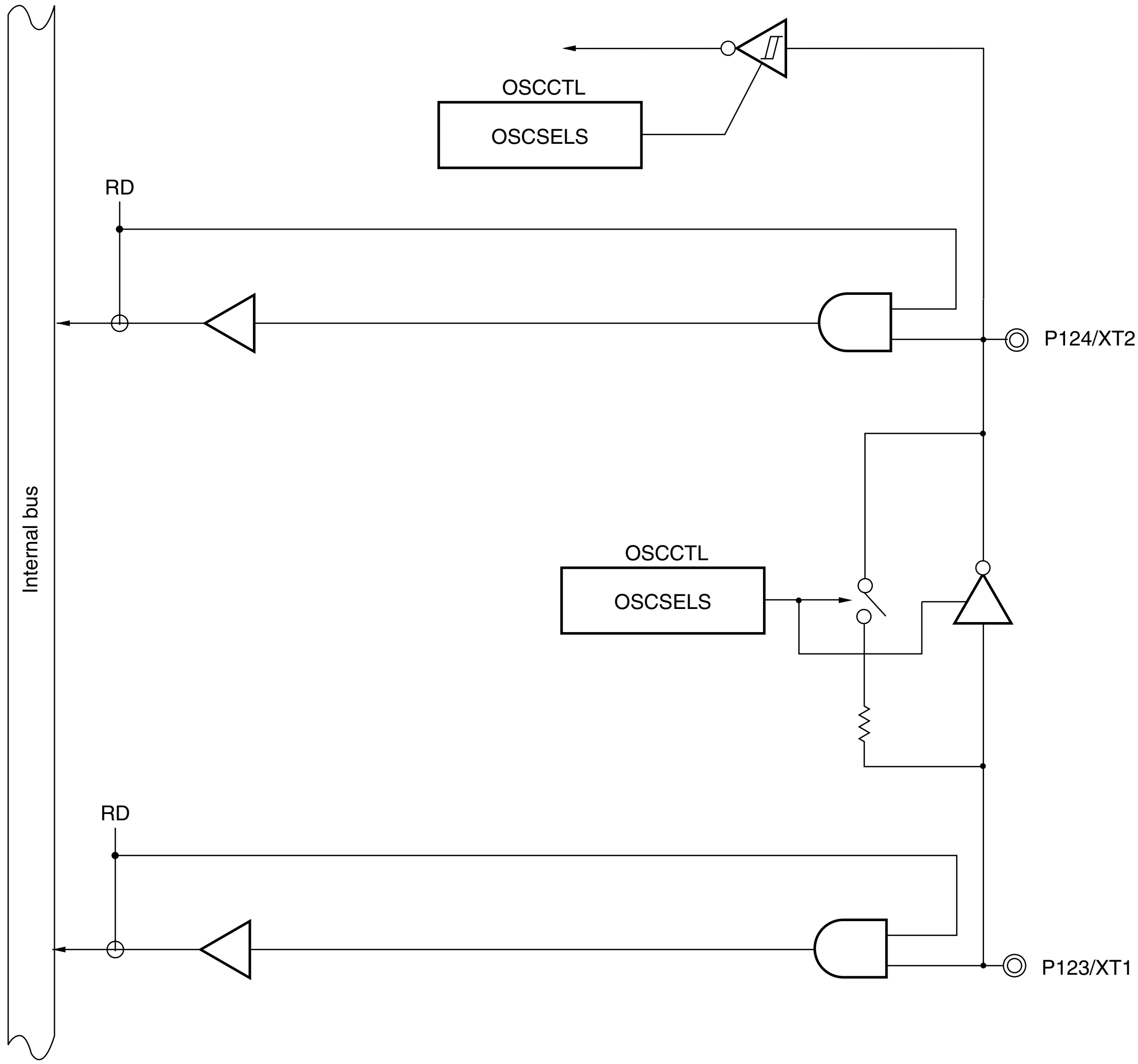
P12: Port register 12
PU12: Pull-up resistor option register 12
PM12: Port mode register 12
RD: Read signal
WR××: Write signal

Figure 4-22. Block Diagram of P121 and P122



OSCCTL: Clock operation mode select register
RD: Read signal

Figure 4-23. Block Diagram of P123 and P124



OSCCTL: Clock operation mode select register
RD: Read signal

4.2.10 Port 13

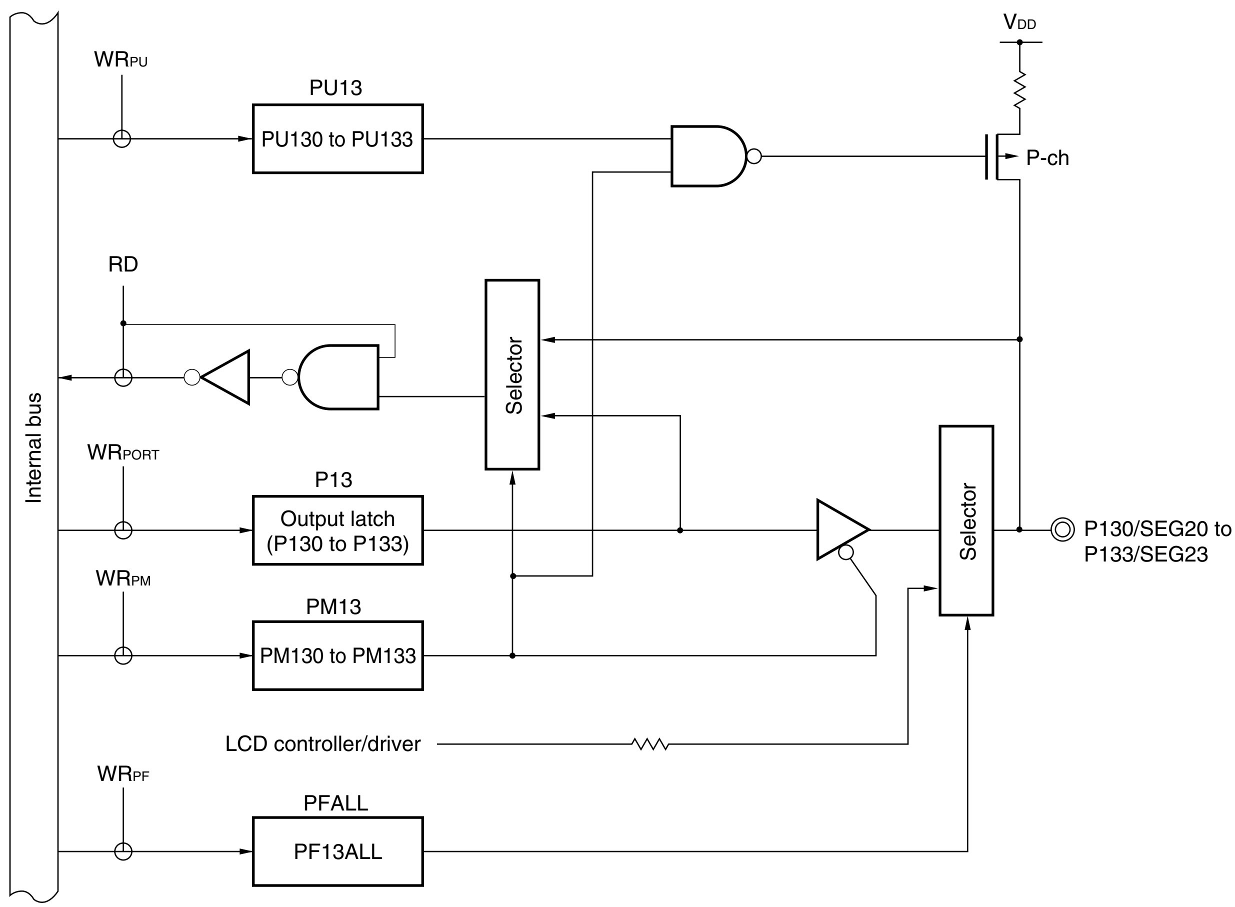
Port 13 is a 4-bit I/O port with an output latch. Port 13 can be set to the input mode or output mode in 1-bit units using port mode register 13 (PM13). When the P130 to P133 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 13 (PU13).

This port can also be used for segment output.

Reset signal generation sets port 13 to input mode.

Figure 4-24 shows a block diagram of port 13.

Figure 4-24. Block Diagram of P130 to P133



P13: Port register 13
PU13: Pull-up resistor option register 13
PM13: Port mode register 13
PFALL: Port function register ALL
RD: Read signal
WR××: Write signal

4.2.11 Port 14

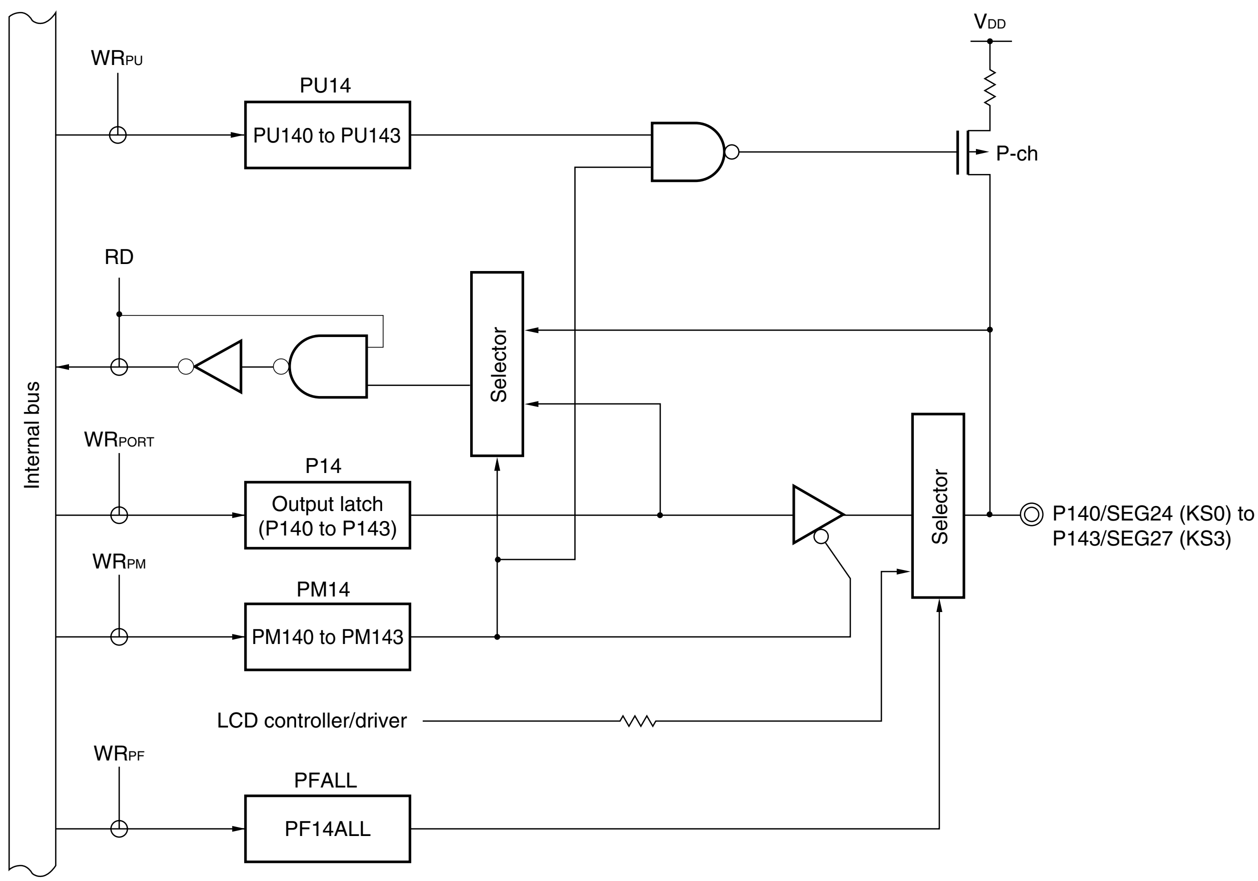
Port 14 is a 4-bit I/O port with an output latch. Port 14 can be set to the input mode or output mode in 1-bit units using port mode register 14 (PM14). When the P140 to P143 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 14 (PU14).

This port can also be used for segment output.

Reset signal generation sets port 14 to input mode.

Figure 4-25 shows a block diagram of port 14.

Figure 4-25. Block Diagram of P140 and P143



P14: Port register 14
PU14: Pull-up resistor option register 14
PM14: Port mode register 14
PFALL: Port function register ALL
RD: Read signal
WR××: Write signal

4.2.12 Port 15

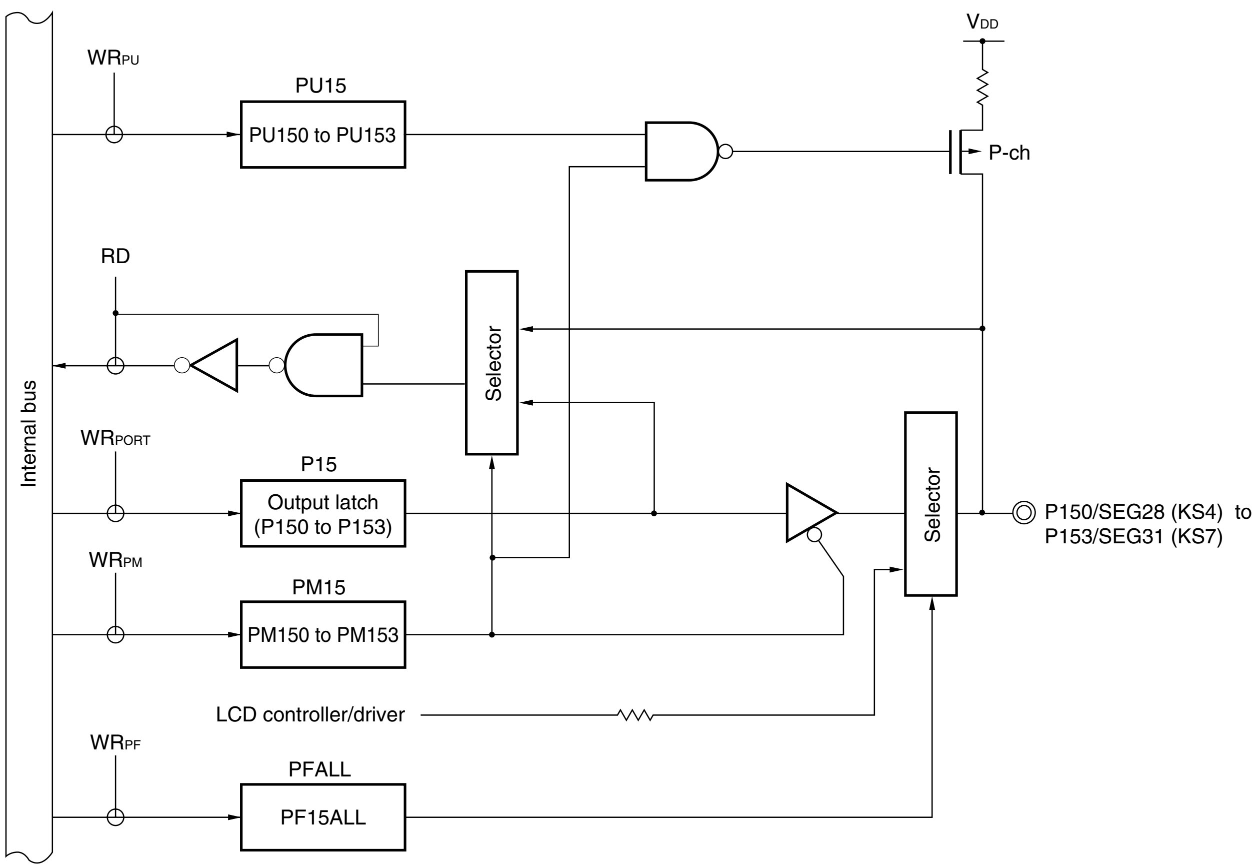
Port 15 is a 4-bit I/O port with an output latch. Port 15 can be set to the input mode or output mode in 1-bit units using port mode register 15 (PM15). When the P150 to P153 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 15 (PU15).

This port can also be used for segment output.

Reset signal generation sets port 15 to input mode.

Figure 4-26 shows a block diagram of port 15.

Figure 4-26. Block Diagram of P150 and P153



P15: Port register 15
PU15: Pull-up resistor option register 15
PM15: Port mode register 15
PFALL: Port function register ALL
RD: Read signal
WR××: Write signal

4.3 Registers Controlling Port Function

Port functions are controlled by the following seven types of registers.

- Port mode registers (PM1 to PM4, PM8 to PM15)
- Port registers (P1 to P4, P8 to P15)
- Pull-up resistor option registers (PU1, PU3, PU4, PU8 to PU15)
- Port function register 1 (PF1)
- Port function register 2 (PF2)[Note 1]
- Port function register ALL (PFALL)
- A/D port configuration register 0 (ADPC0)[Note 2]

Notes 1. μPD78F047x and 78F048x only
2. μPD78F048x and 78F049x only

(1) Port mode registers (PM1 to PM4, PM8 to PM15)

These registers specify input or output mode for the port in 1-bit units.
These registers can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets these registers to FFH.
When port pins are used as alternate-function pins, set the port mode register by referencing **4.5 Settings of PFALL, PF2, PF1, ISC, Port Mode Register, and Output Latch When Using Alternate Function**.

Figure 4-27. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM2	PM27	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	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	FF24H	FFH	R/W
PM8	1	1	1	1	PM83	PM82	PM81	PM80	FF28H	FFH	R/W
PM9	1	1	1	1	PM93	PM92	PM91	PM90	FF29H	FFH	R/W
PM10	1	1	1	1	PM103	PM102	PM101	PM100	FF2AH	FFH	R/W
PM11	1	1	1	1	PM113	PM112	PM111	PM110	FF2BH	FFH	R/W
PM12	1	1	1	1	1	1	1	PM120	FF2CH	FFH	R/W
PM13	1	1	1	1	PM133	PM132	PM131	PM130	FF2DH	FFH	R/W
PM14	1	1	1	1	PM143	PM142	PM141	PM140	FF2EH	FFH	R/W
PM15	1	1	1	1	PM153	PM152	PM151	PM150	FF2FH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 1 to 4, 8 to 15; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set bits 5 to 7 of PM3, bits 4 to 7 of PM8, bits 4 to 7 of PM9, bits 4 to 7 of PM10, bits 4 to 7 of PM11, bits 1 to 7 of PM12, bits 4 to 7 of PM13, bits 4 to 7 of PM14, and bits 4 and 7 of PM15 to "1".

(2) Port registers (P1 to P4, P8 to P15)

These registers write the data that is output from the chip when data is output from a port.

If the data is read in the input mode, the pin level is read. If it is read in the output mode, the output latch value is read.

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

Reset signal generation clears these registers to 00H.

Figure 4-28. Format of Port Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	FF01H	00H (output latch)	R/W
P2	P27	P26	P25	P24	P23	P22	P21	P20	FF02H	00H (output latch)	R/W
P3	0	0	0	P34	P33	P32	P31	P30	FF03H	00H (output latch)	R/W
P4	P47	P46	P45	P44	P43	P42	P41	P40	FF04H	00H (output latch)	R/W
P8	0	0	0	0	P83	P82	P81	P80	FF08H	00H (output latch)	R/W
P9	0	0	0	0	P93	P92	P91	P90	FF09H	00H (output latch)	R/W
P10	0	0	0	0	P103	P102	P101	P100	FF0AH	00H (output latch)	R/W
P11	0	0	0	0	P113	P112	P111	P110	FF0BH	00H (output latch)	R/W
<R> P12	0	0	0	P124[Note 2]	P123[Note 2]	P122[Note 2]	P121[Note 2]	P120	FF0CH	00H[Note 1] (output latch)	R/W[Note 1]
P13	0	0	0	0	P133	P132	P131	P130	FF0DH	00H (output latch)	R/W
<R> P14	PK143[Note 3]	PK142[Note 3]	PK141[Note 3]	PK140[Note 3]	P143	P142	P141	P140	FF0EH	00H (output latch)	R/W
<R> P15	PK153[Note 3]	PK152[Note 3]	PK151[Note 3]	PK150[Note 3]	P153	P152	P151	P150	FF0FH	00H (output latch)	R/W

Pmn	m = 1 to 4, 8 to 15; n = 0 to 7	
	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

<R> **Notes 1.** P121 to P124 are read-only. These become undefined at reset.

<R> **2.** When the operation mode of the pin is the clock input mode, 0 is always read.

<R> **3.** This bit is used for the segment key scan function. For details, see **18.3 Registers Controlling LCD Controller/Driver**.

(3) Pull-up resistor option registers (PU1, PU3, PU4, PU8 to PU15)

These registers specify whether the on-chip pull-up resistors of P10 to P17, P30 to P34, P40 to P47, P80 to P83, P90 to P93, P100 to P103, P110 to P113, P120, P130 to P133, P140 to P143, or P150 to P153 are to be used or not. On-chip pull-up resistors can be used in 1-bit units only for the bits set to input mode of the pins to which the use of an on-chip pull-up resistor has been specified in PU1, PU3, PU4, and PU8 to PU15. On-chip pull-up resistors cannot be connected to bits set to output mode and bits used as alternate-function output pins, regardless of the settings of PU1, PU3, PU4, and PU8 to PU15.

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

Reset signal generation clears these registers to 00H.

Figure 4-29. Format of Pull-up Resistor Option Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	FF31H	00H	R/W
PU3	0	0	0	PU34	PU33	PU32	PU31	PU30	FF33H	00H	R/W
PU4	PU47[Note]	PU46[Note]	PU45[Note]	PU44[Note]	PU43[Note]	PU42[Note]	PU41[Note]	PU40[Note]	FF34H	00H	R/W
PU8	0	0	0	0	PU83	PU82	PU81	PU80	FF38H	00H	R/W
PU9	0	0	0	0	PU93	PU92	PU91	PU90	FF39H	00H	R/W
PU10	0	0	0	0	PU103	PU102	PU101	PU100	FF3AH	00H	R/W
PU11	0	0	0	0	PU113	PU112	PU111	PU110	FF3BH	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	FF3CH	00H	R/W
PU13	0	0	0	0	PU133	PU132	PU131	PU130	FF3DH	00H	R/W
PU14	0	0	0	0	PU143	PU142	PU141	PU140	FF3EH	00H	R/W
PU15	0	0	0	0	PU153	PU152	PU151	PU150	FF3FH	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 1, 3, 4, 8 to 15; n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

<R> **Note** For setting when using the segment key scan function, see **18.3 Registers Controlling LCD Controller/Driver**.

(4) Port function register 1 (PF1)

This register sets the pin functions of P13/SO10/TxD0 and P16/SOA0/TxD6 pins.

PF1 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PF1 to 00H.

Figure 4-30. Format of Port Function Register 1 (PF1)

Address: FF20H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PF1	0	PF16	0	0	PF13	0	0	0

PF16	Port (P16), CSIA0, and UART6 output specification
0	Used as P16 or SOA0
1	Used as TxD6

PF13	Port (P13), CSI10, and UART0 output specification
0	Used as P13 or SO10
1	Used as TxD0

(5) Port function register 2 (PF2) (μPD78F047x and 78F048x only)

This register sets whether to use pins P20 to P27 as port pins (other than segment output pins) or segment output pins.

PF2 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PF2 to 00H.

Figure 4-31. Format of Port Function Register 2 (PF2)

Address: FFB5H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PF2	PF27	PF26	PF25	PF24	PF23	PF22	PF21	PF20

PF2n	Port/segment output specification
0	Used as port (other than segment output)
1	Used as segment output

Remark n = 0 to 7

(6) Port function register ALL (PFALL)

This register sets whether to use pins P8 to P11 and P13 to P15 as port pins (other than segment output pins) or segment output pins.

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

Reset signal generation sets PFALL to 00H.

Figure 4-32. Format of Port Function Register ALL (PFALL)

Address: FFB6H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PFALL	0	PF15ALL	PF14ALL	PF13ALL	PF11ALL	PF10ALL	PF09ALL	PF08ALL

PFnALL	Port/segment output specification
0	Used as port (other than segment output)
1	Used as segment output

Remark n = 08 to 11, 13 to 15

(7) A/D port configuration register 0 (ADPC0) (μPD78F048x and 78F049x only)

This register switches the P20/ANI0 to P27/ANI7 pins to analog input of A/D converter or digital I/O of port.

ADPC0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 08H.

Figure 4-33. Format of A/D Port Configuration Register 0 (ADPC0)

Address: FF8FH After reset: 08H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC0	0	0	0	0	ADPC03	ADPC02	ADPC01	ADPC00

ADPC03	ADPC02	ADPC01	ADPC00	Digital I/O (D)/analog input (A: successive approximation type, Δ: ΔΣ type) switching							
				P27/ANI7/REF+	P26/ANI6/REF-	P25/ANI5/DS2+	P24/ANI4/DS2-	P23/ANI3/DS1+	P22/ANI2/DS1-	P21/ANI1/DS0+	P20/ANI0/DS0-
0	0	0	0	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ
0	0	0	1	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A	D
0	0	1	0	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	D	D
0	0	1	1	A/Δ	A/Δ	A/Δ	A/Δ	A	D	D	D
0	1	0	0	A/Δ	A/Δ	A/Δ	A/Δ	D	D	D	D
0	1	0	1	A	A	A	D	D	D	D	D
0	1	1	0	A	A	D	D	D	D	D	D
0	1	1	1	A	D	D	D	D	D	D	D
1	0	0	0	D	D	D	D	D	D	D	D
Other than above				Setting prohibited							

Cautions 1. Set the channel used for A/D conversion to the input mode by using port mode register 2 (PM2).

2. The pin to be set as a digital I/O via ADPC, must not be set via ADS, ADDS1 or ADDS0.

3. If data is written to ADPC0, a wait cycle is generated. Do not write data to ADPC0 when the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped. For details, see CHAPTER 34 CAUTIONS FOR WAIT.

4. If pins ANI0/P20/SEG39 to ANI7/P27/SEG32 are set to segment output via the PF2 register, output is set to segment output, regardless of the ADPC0 setting (for μPD78F048x only).

4.4 Port Function Operations

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

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

4.4.1 Writing to I/O port

(1) Output mode

A value is written to the output latch by a transfer instruction, and the output latch contents are output from the pin.
Once data is written to the output latch, it is retained until data is written to the output latch again.
The data of the output latch is cleared when a reset signal is generated.

(2) Input mode

A value is written to the output latch by a transfer instruction, but since the output buffer is off, the pin status does not change.
Once data is written to the output latch, it is retained until data is written to the output latch again.

4.4.2 Reading from I/O port

(1) Output mode

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

(2) Input mode

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

4.4.3 Operations on I/O port

(1) Output mode

An operation is performed on the output latch contents, and the result is written to the output latch. The output latch contents are output from the pins.
Once data is written to the output latch, it is retained until data is written to the output latch again.
The data of the output latch is cleared when a reset signal is generated.

(2) Input mode

The pin level is read and an operation is performed on its contents. The result of the operation is written to the output latch, but since the output buffer is off, the pin status does not change.

4.5 Settings of PFALL, PF2, PF1, ISC, Port Mode Register, and Output Latch When Using Alternate Function

To use the alternate function of a port pin, set the PFALL, PF2, PF1, ISC, port mode register, and output latch as shown in Table 4-5.

Table 4-5. Settings of PFALL, PF2, PF1, ISC, Port Mode Register, and Output Latch When Using Alternate Function (1/2)

Pin Name	Alternate Function		PFALL, PF2[Note 4]	PF1	ISC	PM××	P××
	Function Name	I/O					
P10	PCL	Output	–			0	0
P11	$\overline{\text{SCK10}}$	Input	–			1	×
		Output	–			0	1
P12	SI10	Input	–			1	×
	RxD0	Input	–			1	×
P13[Note 10]	SO10	Output	–	PF13 = 0		0	0
	TxD0	Output	–	PF13 = 1		0	×
P14	$\overline{\text{SCKA0}}$	Input	–			1	×
		Output	–			0	1
	INTP4	Input	–			1	×
P15	SIA0	Input	–			1	×
	<RxD6>	Input	–		ISC4 = 1[Note 5, 7], ISC5 = 0	1	×
P16[Note 11]	SOA0	Output	–	PF16 = 0		0	0
	<TxD6>	Output	–	PF16 = 1	ISC4 = 1, ISC5 = 0	0	×
P20 to P27[Note 2]	SEG39 to SEG32[Note 12]	Output	1			×	×
	ANI0 to ANI7[Note 1]	Input	0			1	×
	DS0± to DS2±[Note 8]	Input	0			1	×
	REF±[Note 8]	Input	0			1	×
P30	INTP5	Input	–			1	×
P31	TOH1	Output	–			0	0
	INTP3	Input	–			1	×
P32	TOH0	Output	–			0	0
	MCGO	Output	–			0	0
P33	TI000	Input	–		ISC1 = 0	1	×
	RTCDIV	Output	–			0	0
	RTCCL	Output	–			0	0
	BUZ	Output	–			0	0
	INTP2	Input	–			1	×
P34	TI52	Input	–		**Note 6**	1	×
	TI010	Input	–			1	×
	TO00	Output	–			0	0
	RTC1HZ	Output	–			0	0
	INTP1	Input	–			1	×

(Note and Remark are listed on the page after next.)

Table 4-5. Settings of PFALL, PF2, PF1, ISC, Port Mode Register, and Output Latch When Using Alternate Function (2/2)

Pin Name	Alternate Function		PFALL, PF2[Note 4]	ISC	PM××	P××
	Function Name	I/O				
P40	KR0	Input	–		1	×
	V_{LC3}[Note 9]	Input	–		×	×
P41	KR1	Input	–		1	×
	RIN	Input	–		1	×
P42	KR2	Input	–		1	×
P43	KR3	Input	–		1	×
	TI51	Input	–		1	×
	TO51	Output	–		0	0
P44	KR4	Input	–		1	×
	TI50	Input	–		1	×
	TO50	Output	–		0	0
P45	KR5	Input	–		1	×
P46	KR6	Input	–		1	×
P47	KR7	Input	–		1	×
P80 to P83	SEG4 to SEG7	Output	1		×	×
P90 to P93	SEG8 to SEG11	Output	1		×	×
P100 to P103	SEG12 to SEG15	Output	1		×	×
P110	SEG16	Output	1	ISC3 = 0	×	×
P111	SEG17	Output	1	ISC3 = 0	×	×
P112	SEG18	Output	1	ISC3 = 0	×	×
	TxD6	Output	0	ISC3 = 1, ISC4 = ISC5 = 0	0	1
P113	SEG19	Output	1	ISC3 = 0	×	×
	RxD6	Input	0	ISC3 = 1, ISC4 = ISC5 = 0 [Notes 5, 7]	1	×
P120	EXLVI	Input	–		1	×
	INTP0	Input	–	ISC0 = 0	1	×
P121	X1[Note 3]	–	–		×	×
	OCD0A	–	–		×	×
P122	X2[Note 3]	–	–		×	×
	EXCLK[Note 3]	Input	–		×	×
	OCD0B	–	–		×	×
P123	XT1[Note 3]	–	–		×	×
P124	XT2[Note 3]	–	–		×	×
P130 to P133	SEG20 to SEG23	Output	1		×	×
P140 to P143	SEG24 (KS0) to SEG27 (KS3)	Output	1		×	×
P150 to P153	SEG28 (KS4) to SEG31 (KS7)	Output	1		×	×

(Note and Remark are listed on the next page.)

Notes 1. μPD78F048x and 78F049x only.

2. The functions of the P20/ANI0/DS0–, P21/ANI1/DS0+, P22/ANI2/DS1–, P23/ANI3/DS1+, P24/ANI4/DS2–, P25/ANI5/DS2+, P26/ANI6/REF–, and P27/ANI7/REF+ pins are determined according to the settings of port function register 2 (PF2), A/D port configuration register 0 (ADPC0), port mode register 2 (PM2), analog input channel specification register (ADS), and $\Delta\Sigma$ A/D converter mode register 0 (ADDCTL0).

Table 4-6. Setting Functions of P20/SEG39[Note 1]/ANI0[Note 2]/DS0–[Note 3] to P27/SEG32[Note 1]/ANI7[Note 2]/REF+[Note 3] Pins

PF2[Note 1]	ADPC0	PM2	ADS	ADDCTL0	P20/SEG39[Note 1]/ANI0[Note 2]/DS0–[Note 3] to P27/SEG32[Note 1]/ANI7[Note 2]/REF+[Note 3] Pins
Digital/Analog selection	Analog input selection	Input mode	Does not select ANI.	Does not select DSn±.	Analog input (not to be converted)
			Selects ANI.	Does not select DSn±.	Analog input (to be converted by successive approximation type A/D converter)
			Does not select ANI.	Selects DSn±.	Analog input (to be converted by 16-bit $\Delta\Sigma$ type A/D converter)
			Selects ANI.	Selects DSn±.	Setting prohibited
		Output mode	–		Setting prohibited
	Digital I/O selection	Input mode	–		Digital input
		Output mode	–		Digital output
SEG output selection[Note 1]	–	–	–		Segment output[Note 1]

Notes 1. μPD78F047x and 78F048x only.

2. μPD78F048x and 78F049x only.

3. μPD78F049x only.

Remark n = 0 to 2

3. When using the P121 to P124 pins to connect a resonator for the main system clock (X1, X2) or subsystem clock (XT1, XT2), or to input an external clock for the main system clock (EXCLK), the X1 oscillation mode, XT1 oscillation mode, or external clock input mode must be set by using the clock operation mode select register (OSCCTL) (for details, see **5.3 (1) Clock operation mode select register (OSCCTL)** and **(3) Setting of operation mode for subsystem clock pin**). The reset value of OSCCTL is 00H (all of the P121 to P124 are Input port pins).

4. Targeted at registers corresponding to each port.

5. RxD6 can be set as the input source for TI000 by setting ISC1 = 1.

6. Input enable of TM52 via TMH2 can be controlled by setting ISC2 = 1.

7. RxD6 can be set as the input source for INTP0 by setting ISC0 = 1.

8. μPD78F049x only.

9. When the P40/KR0/V_{LC3} pin is set to the 1/4 bias method, it is used as V_{LC3}. When the pin is set to another bias method, it is used for the port function (P40) or the key interrupt function (KR0).

10. Set PF13 = 0 when using as port function.

11. Set PF16 = 0 when using as port function.

12. μPD78F047x and 78F048x only.

Remarks 1. ×: Don't care

–: Does not apply.

PM××: Port mode register

P××: Port output latch

2. The functions within arrowheads (< >) can be assigned by setting the input switch control register (ISC).

3. X1, X2 pins can be used as on-chip debug mode setting pins (OCD1A, OCD1B) when the on-chip debug function is used. For detail, see **CHAPTER 29 ON-CHIP DEBUG FUNCTION.**

CHAPTER 5 CLOCK GENERATOR

5.1 Functions of Clock Generator

The clock generator generates the clock to be supplied to the CPU and peripheral hardware.
The following three kinds of system clocks and clock oscillators are selectable.

(1) Main system clock

<1> X1 oscillator

This circuit oscillates a clock of f_X = 2 to 10 MHz by connecting a resonator to X1 and X2.
Oscillation can be stopped by executing the STOP instruction or using the main OSC control register (MOC).

<2> Internal high-speed oscillator

This circuit oscillates a clock of f_{RH} = 8 MHz (TYP.). After a reset release, the CPU always starts operating with this internal high-speed oscillation clock. Oscillation can be stopped by executing the STOP instruction or using the internal oscillation mode register (RCM).

An external main system clock (f_{EXCLK} = 2 to 10 MHz) can also be supplied from the OCD0B/EXCLK/X2/P122 pin. An external main system clock input can be disabled by executing the STOP instruction or using RCM.
As the main system clock, a high-speed system clock (X1 clock or external main system clock) or internal high-speed oscillation clock can be selected by using the main clock mode register (MCM).

(2) Subsystem clock

- **Subsystem clock oscillator**

This circuit oscillates at a frequency of f_{XT} = 32.768 kHz by connecting a 32.768 kHz resonator across XT1 and XT2. Oscillation can be stopped by using the processor clock control register (PCC) and clock operation mode select register (OSCCTL).

Remarks 1. f_X: X1 clock oscillation frequency

2. f_{RH}: Internal high-speed oscillation clock frequency

3. f_{EXCLK}: External main system clock frequency

4. f_{XT}: XT1 clock oscillation frequency

(3) Internal low-speed oscillation clock (clock for watchdog timer)

- **Internal low-speed oscillator**

This circuit oscillates a clock of f_{RL} = 240 kHz (TYP.). After a reset release, the internal low-speed oscillation clock always starts operating.

Oscillation can be stopped by using the internal oscillation mode register (RCM) when "internal low-speed oscillator can be stopped by software" is set by option byte.

The internal low-speed oscillation clock cannot be used as the CPU clock. The following hardware operates with the internal low-speed oscillation clock.

- Watchdog timer
- 8-bit timer H1 (if f_{RL}, $f_{RL}/2^7$ or $f_{RL}/2^9$ is selected as the count clock)
- LCD controller/driver (if $f_{RL}/2^3$ is selected as the LCD source clock)

Remark f_{RL}: Internal low-speed oscillation clock frequency

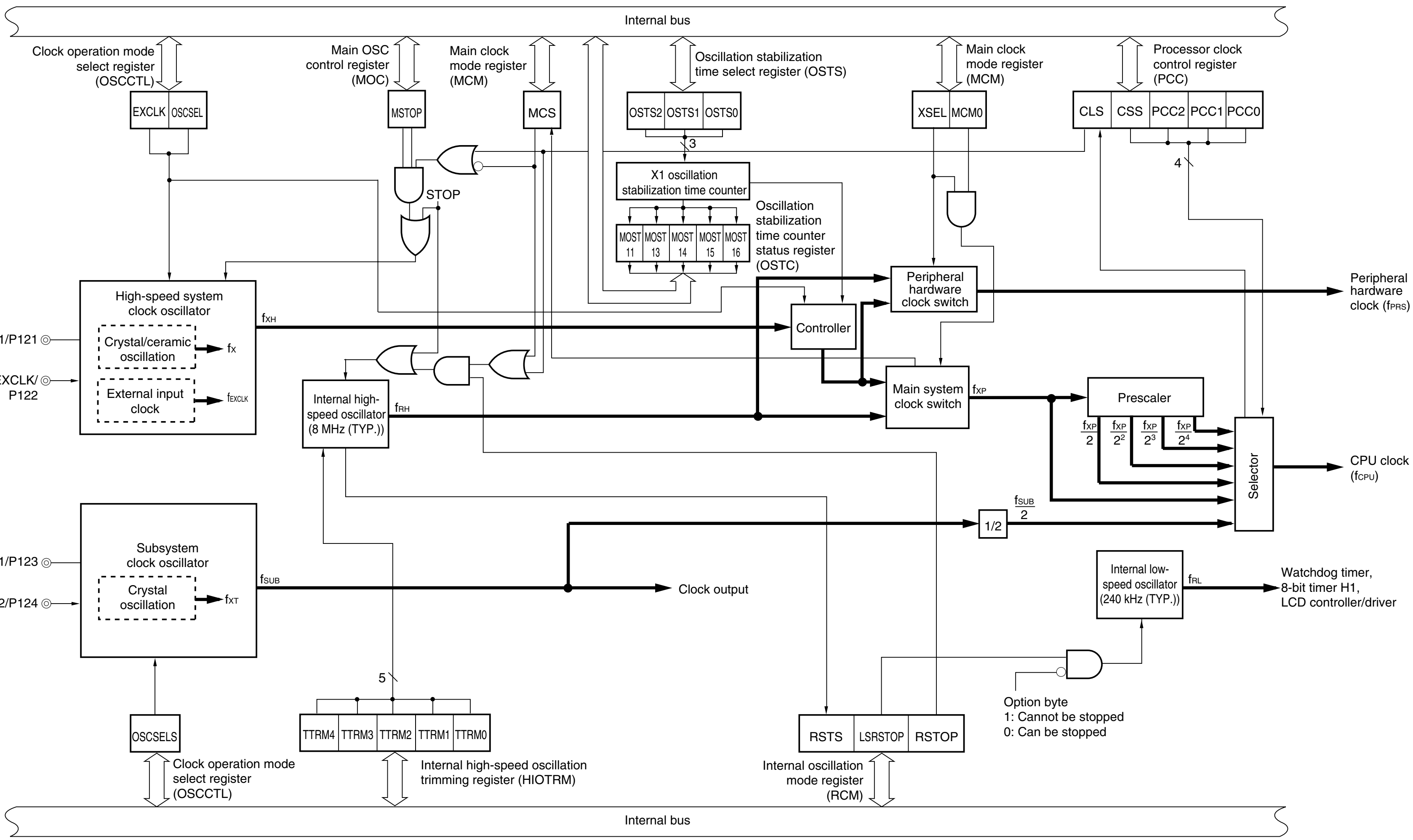
5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
Control registers	Clock operation mode select register (OSCCTL) Processor clock control register (PCC) Internal oscillation mode register (RCM) Main OSC control register (MOC) Main clock mode register (MCM) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS) Internal high-speed oscillation trimming register (HIOTRM)
Oscillators	X1 oscillator XT1 oscillator Internal high-speed oscillator Internal low-speed oscillator

Figure 5-1. Block Diagram of Clock Generator



Remarks 1. f_X: X1 clock oscillation frequency
2. f_{RH}: Internal high-speed oscillation clock frequency
3. f_{EXCLK}: External main system clock frequency
4. f_{XH}: High-speed system clock frequency
5. f_{XP}: Main system clock frequency
6. f_{PRS}: Peripheral hardware clock frequency
7. f_{CPU}: CPU clock frequency
8. f_{XT}: XT1 clock oscillation frequency
9. f_{SUB}: Subsystem clock frequency
10. f_{RL}: Internal low-speed oscillation clock frequency

5.3 Registers Controlling Clock Generator

The following eight registers are used to control the clock generator.

- Clock operation mode select register (OSCCTL)
- Processor clock control register (PCC)
- Internal oscillation mode register (RCM)
- Main OSC control register (MOC)
- Main clock mode register (MCM)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- Internal high-speed oscillation trimming register (HIOTRM)

(1) Clock operation mode select register (OSCCTL)

This register selects the operation modes of the high-speed system and subsystem clocks, and the gain of the on-chip oscillator.
OSCCTL can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 5-2. Format of Clock Operation Mode Select Register (OSCCTL)

Address: FF9FH After reset: 00H R/W

Symbol	<7>	<6>	5	<4>	3	2	1	0
OSCCTL	EXCLK	OSCSEL	0	OSCSELS	0	0	0	0

EXCLK	OSCSEL	High-speed system clock pin operation mode	P121/X1 pin	P122/X2/EXCLK pin
0	0	Input port mode	Input port	
0	1	X1 oscillation mode	Crystal/ceramic resonator connection	
1	0	Input port mode	Input port	
1	1	External clock input mode	Input port	External clock input

Caution **To change the value of EXCLK and OSCSEL, be sure to confirm that bit 7 (MSTOP) of the main OSC control register (MOC) is 1 (the X1 oscillator stops or the external clock from the EXCLK pin is disabled).**
Be sure to clear bits 0 to 3, and 5 to "0".

Remark f_{XH}: High-speed system clock oscillation frequency

(2) Processor clock control register (PCC)

This register is used to select the CPU clock, the division ratio, and operation mode for subsystem clock.

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

Reset signal generation sets PCC to 01H.

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

Address: FFFBH After reset: 01H R/W[Note 1]

Symbol	7	6	<5>	<4>	3	2	1	0
PCC	0	0	CLS	CSS	0	PCC2	PCC1	PCC0

CLS	CPU clock status
0	Main system clock
1	Subsystem clock

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

Note Bit 5 is read-only.

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

Remarks 1. f_{XP}: Main system clock oscillation frequency

2. f_{SUB}: Subsystem clock oscillation frequency

The fastest instruction can be executed in 2 clocks of the CPU clock in the 78K0/LF3. Therefore, the relationship between the CPU clock (f_{CPU}) and the minimum instruction execution time is as shown in Table 5-2.

Table 5-2. Relationship Between CPU Clock and Minimum Instruction Execution Time

CPU Clock (f_{CPU})	Minimum Instruction Execution Time: $2/f_{CPU}$		
	Main System Clock		Subsystem Clock
	High-Speed System Clock[Note]	Internal High-Speed Oscillation Clock[Note]	
	At 10 MHz Operation	At 8 MHz (TYP.) Operation	At 32.768 kHz Operation
f_{XP}	0.2 μs	0.25 μs (TYP.)	–
$f_{XP}/2$	0.4 μs	0.5 μs (TYP.)	–
$f_{XP}/2^2$	0.8 μs	1.0 μs (TYP.)	–
$f_{XP}/2^3$	1.6 μs	2.0 μs (TYP.)	–
$f_{XP}/2^4$	3.2 μs	4.0 μs (TYP.)	–
$f_{SUB}/2$	–	–	122.1 μs

Note The main clock mode register (MCM) is used to set the main system clock supplied to CPU clock (high-speed system clock/internal high-speed oscillation clock) (see **Figure 5-6**).

(3) Setting of operation mode for subsystem clock pin

The operation mode for the subsystem clock pin can be set by using bit 4 (OSCSELS) of the clock operation mode select register (OSCCTL) in combination.

Table 5-3. Setting of Operation Mode for Subsystem Clock Pin

Bit 4 of OSCCTL	Subsystem Clock Pin Operation Mode	P123/XT1 Pin	P124/XT2 Pin
OSCSELS			
0	Input port mode	Input port	
1	XT1 oscillation mode	Crystal resonator connection	

Caution Confirm that bit 5 (CLS) of the processor clock control register (PCC) is 0 (CPU is operating with main system clock) when changing the current values of OSCSELS.

(4) Internal oscillation mode register (RCM)

This register sets the operation mode of internal oscillator.

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

Reset signal generation sets this register to 80H[Note 1].

Figure 5-4. Format of Internal Oscillation Mode Register (RCM)

Address: FFA0H After reset: 80H[Note 1] R/W[Note 2]

Symbol	<7>	6	5	4	3	2	<1>	<0>
RCM	RSTS	0	0	0	0	0	LSRSTOP	RSTOP

RSTS	Status of internal high-speed oscillator
0	Waiting for accuracy stabilization of internal high-speed oscillator
1	Stability operating of internal high-speed oscillator

LSRSTOP	Internal low-speed oscillator oscillating/stopped
0	Internal low-speed oscillator oscillating
1	Internal low-speed oscillator stopped

RSTOP	Internal high-speed oscillator oscillating/stopped
0	Internal high-speed oscillator oscillating
1	Internal high-speed oscillator stopped

Notes 1. The value of this register is 00H immediately after a reset release but automatically changes to 80H after internal high-speed oscillator has been stabilized.

2. Bit 7 is read-only.

Caution When setting RSTOP to 1, be sure to confirm that the CPU operates with a clock other than the internal high-speed oscillation clock. Specifically, set under either of the following conditions.

- **When MCS = 1 (when CPU operates with the high-speed system clock)**
- **When CLS = 1 (when CPU operates with the subsystem clock)**

In addition, stop peripheral hardware that is operating on the internal high-speed oscillation clock before setting RSTOP to 1.

(5) Main OSC control register (MOC)

This register selects the operation mode of the high-speed system clock.

This register is used to stop the X1 oscillator or to disable an external clock input from the EXCLK pin when the CPU operates with a clock other than the high-speed system clock.

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

Reset signal generation sets this register to 80H.

Figure 5-5. Format of Main OSC Control Register (MOC)

Address: FFA2H After reset: 80H R/W

Symbol	<7>	6	5	4	3	2	1	0
MOC	MSTOP	0	0	0	0	0	0	0

MSTOP	Control of high-speed system clock operation	
	X1 oscillation mode	External clock input mode
0	X1 oscillator operating	External clock from EXCLK pin is enabled
1	X1 oscillator stopped	External clock from EXCLK pin is disabled

Cautions 1. When setting MSTOP to 1, be sure to confirm that the CPU operates with a clock other than the high-speed system clock. Specifically, set under either of the following conditions.

- **When MCS = 0 (when CPU operates with the internal high-speed oscillation clock)**
- **When CLS = 1 (when CPU operates with the subsystem clock)**

In addition, stop peripheral hardware that is operating on the high-speed system clock before setting MSTOP to 1.

2. Do not clear MSTOP to 0 while bit 6 (OSCSEL) of the clock operation mode select register (OSCCTL) is 0 (I/O port mode).

3. The peripheral hardware cannot operate when the peripheral hardware clock is stopped. To resume the operation of the peripheral hardware after the peripheral hardware clock has been stopped, initialize the peripheral hardware.

(6) Main clock mode register (MCM)

This register selects the main system clock supplied to CPU clock and clock supplied to peripheral hardware clock.

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

Reset signal generation clears this register to 00H.

Figure 5-6. Format of Main Clock Mode Register (MCM)

Address: FFA1H After reset: 00H R/W[Note]

Symbol	7	6	5	4	3	<2>	<1>	<0>
MCM	0	0	0	0	0	XSEL	MCS	MCM0

XSEL	MCM0	Selection of clock supplied to main system clock and peripheral hardware	
		Main system clock (f_{XP})	Peripheral hardware clock (f_{PRS})
0	0	Internal high-speed oscillation clock (f_{RH})	Internal high-speed oscillation clock (f_{RH})
0	1		
1	0		High-speed system clock (f_{XH})
1	1	High-speed system clock (f_{XH})	

MCS	Main system clock status
0	Operates with internal high-speed oscillation clock
1	Operates with high-speed system clock

Note Bit 1 is read-only.

Cautions 1. XSEL can be changed only once after a reset release.

2. A clock other than f_{PRS} is supplied to the following peripheral functions regardless of the setting of XSEL and MCM0.

- **Watchdog timer (operates with internal low-speed oscillation clock)**
- **When "f_{RL}", "$f_{RL}/2^7$", or "$f_{RL}/2^9$" is selected as the count clock for 8-bit timer H1 (operates with internal low-speed oscillation clock)**
- **When "$f_{RL}/2^3$" is selected as the LCD source clock for LCD controller/driver (operates with internal low-speed oscillation clock)**
- **Peripheral hardware selects the external clock as the clock source (Except when the external count clock of TM00 is selected (TI000 pin valid edge))**

(7) Oscillation stabilization time counter status register (OSTC)

This is the register that indicates the count status of the X1 clock oscillation stabilization time counter. When X1 clock oscillation starts with the internal high-speed oscillation clock or subsystem clock used as the CPU clock, the X1 clock oscillation stabilization time can be checked.

OSTC can be read by a 1-bit or 8-bit memory manipulation instruction.

When reset is released (reset by $\overline{\text{RESET}}$ input, POC, LVI, and WDT), the STOP instruction and MSTOP (bit 7 of MOC register) = 1 clear OSTC to 00H.

Figure 5-7. Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFA3H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
OSTC	0	0	0	MOST11	MOST13	MOST14	MOST15	MOST16

MOST 11	MOST 13	MOST 14	MOST 15	MOST 16	Oscillation stabilization time status			
						fx = 2 MHz	fx = 5 MHz	fx = 10 MHz
1	0	0	0	0	2^{11}/fx min.	1.02 ms min.	409.6 μs min.	204.8 μs min.
1	1	0	0	0	2^{13}/fx min.	4.10 ms min.	1.64 ms min.	819.2 μs min.
1	1	1	0	0	2^{14}/fx min.	8.19 ms min.	3.27 ms min.	1.64 ms min.
1	1	1	1	0	2^{15}/fx min.	16.38 ms min.	6.55 ms min.	3.27 ms min.
1	1	1	1	1	2^{16}/fx min.	32.77 ms min.	13.11 ms min.	6.55 ms min.

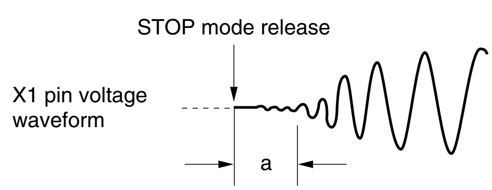
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST11 and remain 1.

2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock, set the oscillation stabilization time as follows.

- **Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS**

Note, therefore, that only the status up to the oscillation stabilization time set by OSTS is set to OSTC after STOP mode is released.

3. The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts (“a” below).



Remark fx: X1 clock oscillation frequency

(8) Oscillation stabilization time select register (OSTS)

This register is used to select the X1 clock oscillation stabilization wait time when the STOP mode is released.

When the X1 clock is selected as the CPU clock, the operation waits for the time set using OSTS after the STOP mode is released.

When the internal high-speed oscillation clock is selected as the CPU clock, confirm with OSTC that the desired oscillation stabilization time has elapsed after the STOP mode is released. The oscillation stabilization time can be checked up to the time set using OSTC.

OSTS can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets OSTS to 05H.

Figure 5-8. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFA4H After reset: 05H R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection			
				fx = 2 MHz	fx = 5 MHz	fx = 10 MHz
0	0	1	2^{11}/fx	1.02 ms	409.6 μs	204.8 μs
0	1	0	2^{13}/fx	4.10 ms	1.64 ms	819.2 μs
0	1	1	2^{14}/fx	8.19 ms	3.27 ms	1.64 ms
1	0	0	2^{15}/fx	16.38 ms	6.55 ms	3.27 ms
1	0	1	2^{16}/fx	32.77 ms	13.11 ms	6.55 ms
Other than above			Setting prohibited			

Cautions 1. To set the STOP mode when the X1 clock is used as the CPU clock, set OSTS before executing the STOP instruction.

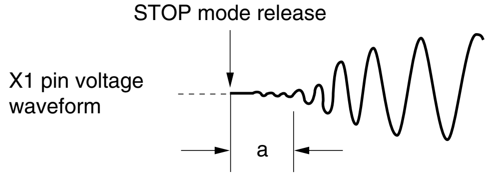
2. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

3. The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock, set the oscillation stabilization time as follows.

- **Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS**

Note, therefore, that only the status up to the oscillation stabilization time set by OSTS is set to OSTC after STOP mode is released.

4. The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts (“a” below).



Remark fx: X1 clock oscillation frequency

(9) Internal high-speed oscillation trimming register (HIOTRM)

This register corrects the accuracy of the internal high-speed oscillator. The accuracy can be corrected by self-measuring the frequency of the internal high-speed oscillator, using a subsystem clock using a crystal resonator or using a timer with high-accuracy external clock input, such as a real-time counter.

HIOTRM can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets HIOTRM to 10H.

Caution If the temperature or V_{DD} pin voltage is changed after accuracy correction, the frequency will fluctuate. Also, if a value other than the initial value (10H) is set to the HIOTRM register, the oscillation accuracy of the internal high-speed oscillation clock may exceed the MIN. and MAX. values described in CHAPTER 31 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS) due to the subsequent fluctuation in the temperature or V_{DD} voltage, or HIOTRM register setting value. If the temperature or V_{DD} voltage fluctuates, accuracy correction must be executed either before frequency accuracy will be required or regularly.

<R> **Figure 5-9. Format of Internal High-speed Oscillation Trimming Register (HIOTRM)**

Address: FF30H After reset: 10H R/W

Symbol	7	6	5	4	3	2	1	0
HIOTRM	0	0	0	TTRM4	TTRM3	TTRM2	TTRM1	TTRM0

TTRM4	TTRM3	TTRM2	TTRM1	TTRM0	Clock correction value ($2.7\ V \le V_{DD} \le 5.5\ V$)		
					MIN.	TYP.	MAX.
0	0	0	0	0	−5.54%	−4.88%	−4.02%
0	0	0	0	1	−5.28%	−4.62%	−3.76%
0	0	0	1	0	−4.99%	−4.33%	−3.47%
0	0	0	1	1	−4.69%	−4.03%	−3.17%
0	0	1	0	0	−4.39%	−3.73%	−2.87%
0	0	1	0	1	−4.09%	−3.43%	−2.57%
0	0	1	1	0	−3.79%	−3.13%	−2.27%
0	0	1	1	1	−3.49%	−2.83%	−1.97%
0	1	0	0	0	−3.19%	−2.53%	−1.67%
0	1	0	0	1	−2.88%	−2.22%	−1.36%
0	1	0	1	0	−2.23%	−1.91%	−1.31%
0	1	0	1	1	−1.92%	−1.60%	−1.28%
0	1	1	0	0	−1.60%	−1.28%	−0.96%
0	1	1	0	1	−1.28%	−0.96%	−0.64%
0	1	1	1	0	−0.96%	−0.64%	−0.32%
0	1	1	1	1	−0.64%	−0.32%	±0%
1	0	0	0	0	±0% (default)		
1	0	0	0	1	±0%	+0.32%	+0.64%
1	0	0	1	0	+0.33%	+0.65%	+0.97%
1	0	0	1	1	+0.66%	+0.98%	+1.30%
1	0	1	0	0	+0.99%	+1.31%	+1.63%
1	0	1	0	1	+1.32%	+1.64%	+1.96%
1	0	1	1	0	+1.38%	+1.98%	+2.30%
1	0	1	1	1	+1.46%	+2.32%	+2.98%
1	1	0	0	0	+1.80%	+2.66%	+3.32%
1	1	0	0	1	+2.14%	+3.00%	+3.66%
1	1	0	1	0	+2.48%	+3.34%	+4.00%
1	1	0	1	1	+2.83%	+3.69%	+4.35%
1	1	1	0	0	+3.18%	+4.04%	+4.70%
1	1	1	0	1	+3.53%	+4.39%	+5.05%
1	1	1	1	0	+3.88%	+4.74%	+5.40%
1	1	1	1	1	+4.24%	+5.10%	+5.76%

Caution The internal high-speed oscillation frequency will increase in speed if the HIOTRM register value is incremented above a specific value, and will decrease in speed if decremented below that specific value. A reversal, such that the frequency decreases in speed by incrementing the value, or increases in speed by decrementing the value, will not occur.

5.4 System Clock Oscillator

5.4.1 X1 oscillator

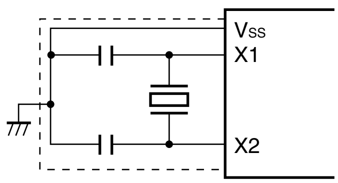
The X1 oscillator oscillates with a crystal resonator or ceramic resonator (2 to 10 MHz) connected to the X1 and X2 pins.

An external clock can also be input. In this case, input the clock signal to the EXCLK pin.

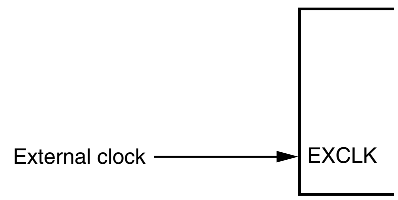
Figure 5-10 shows an example of the external circuit of the X1 oscillator.

Figure 5-10. Example of External Circuit of X1 Oscillator

(a) Crystal or ceramic oscillation



(b) External clock



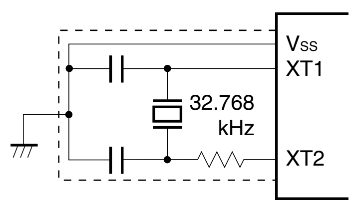
5.4.2 XT1 oscillator

The XT1 oscillator oscillates with a crystal resonator (standard: 32.768 kHz) connected to the XT1 and XT2 pins.

Figure 5-11 shows an example of the external circuit of the XT1 oscillator.

Figure 5-11. Example of External Circuit of XT1 Oscillator

(a) Crystal oscillation



Caution 1. When using the X1 oscillator and XT1 oscillator, wire as follows in the area enclosed by the broken lines in the Figures 5-10 and 5-11 to avoid an adverse effect from wiring capacitance.

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

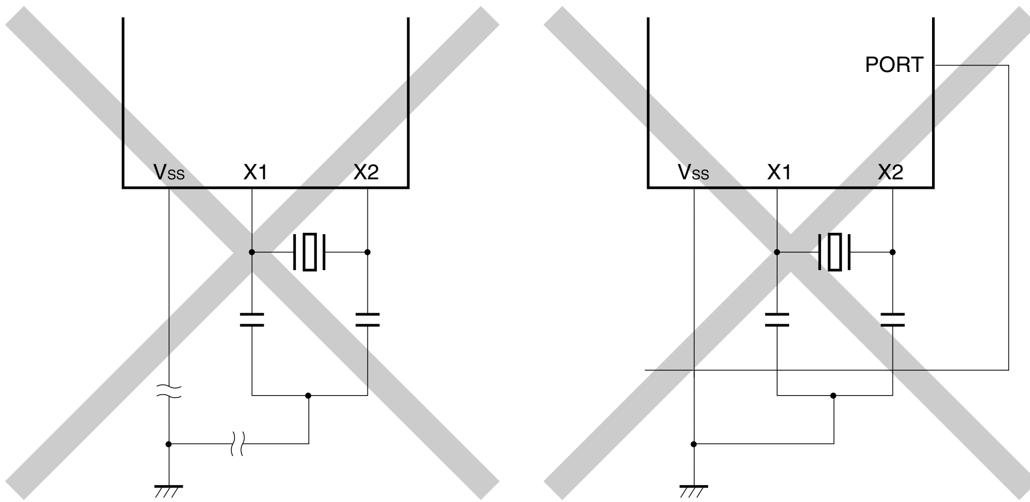
Note that the XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption.

Figure 5-12 shows examples of incorrect resonator connection.

Figure 5-12. Examples of Incorrect Resonator Connection (1/2)

(a) Too long wiring

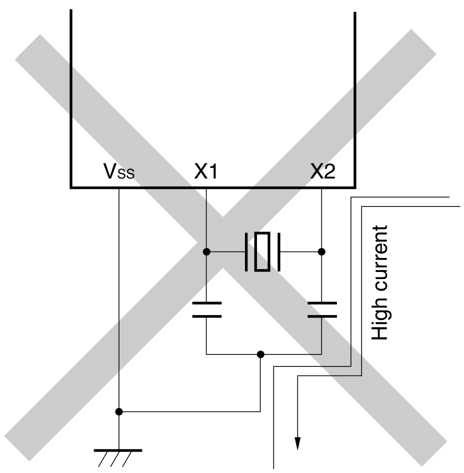
(b) Crossed signal line



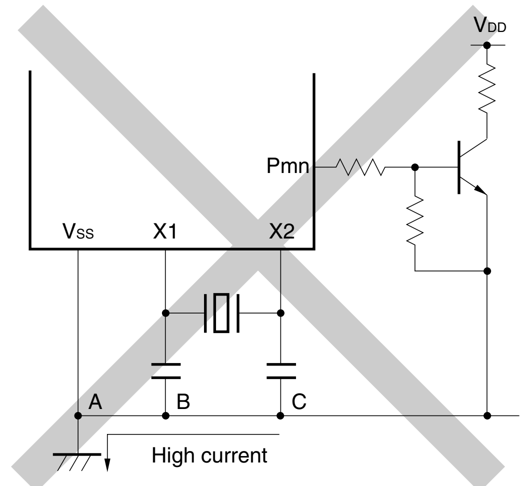
Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

Figure 5-12. Examples of Incorrect Resonator Connection (2/2)

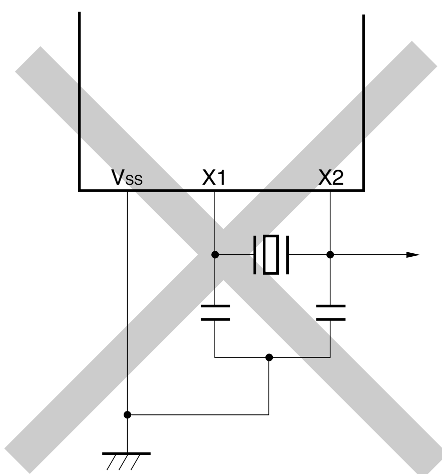
(c) Wiring near high alternating current



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



(e) Signals are fetched



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

Caution 2. When X2 and XT1 are wired in parallel, the crosstalk noise of X2 may increase with XT1, resulting in malfunctioning.

5.4.3 When subsystem clock is not used

If it is not necessary to use the subsystem clock for low power consumption operations, or if not using the subsystem clock as an I/O port, set the XT1 and XT2 pins to Input port mode (OSCSELS = 0) and independently connect to V_{DD} or V_{SS} via a resistor.

Remark OSCSELS: Bit 4 of clock operation mode select register (OSCCTL)

5.4.4 Internal high-speed oscillator

The internal high-speed oscillator is incorporated in the 78K0/LF3. Oscillation can be controlled by the internal oscillation mode register (RCM).

After a reset release, the internal high-speed oscillator automatically starts oscillation (8 MHz (TYP.)).

5.4.5 Internal low-speed oscillator

The internal low-speed oscillator is incorporated in the 78K0/LF3.

The internal low-speed oscillation clock is only used as the clock of the watchdog timer, 8-bit timer H1, and LCD controller/driver. The internal low-speed oscillation clock cannot be used as the CPU clock.

“Can be stopped by software” or “Cannot be stopped” can be selected by the option byte. When “Can be stopped by software” is set, oscillation can be controlled by the internal oscillation mode register (RCM).

After a reset release, the internal low-speed oscillator automatically starts oscillation, and the watchdog timer is driven (240 kHz (TYP.)) if the watchdog timer operation is enabled using the option byte.

5.4.6 Prescaler

The prescaler generates various clocks by dividing the main system clock when the main system clock is selected as the clock to be supplied to the CPU.

5.5 Clock Generator Operation

The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode (see **Figure 5-1**).

- Main system clock f_{XP}
  - High-speed system clock f_{XH}
    X1 clock f_X
    External main system clock f_{EXCLK}
  - Internal high-speed oscillation clock f_{RH}
- Subsystem clock f_{SUB}
  - XT1 clock f_{XT}
- Internal low-speed oscillation clock f_{RL}
- CPU clock f_{CPU}
- Peripheral hardware clock f_{PRS}

The CPU starts operation when the internal high-speed oscillator starts outputting after a reset release in the 78K0/LF3, thus enabling the following.

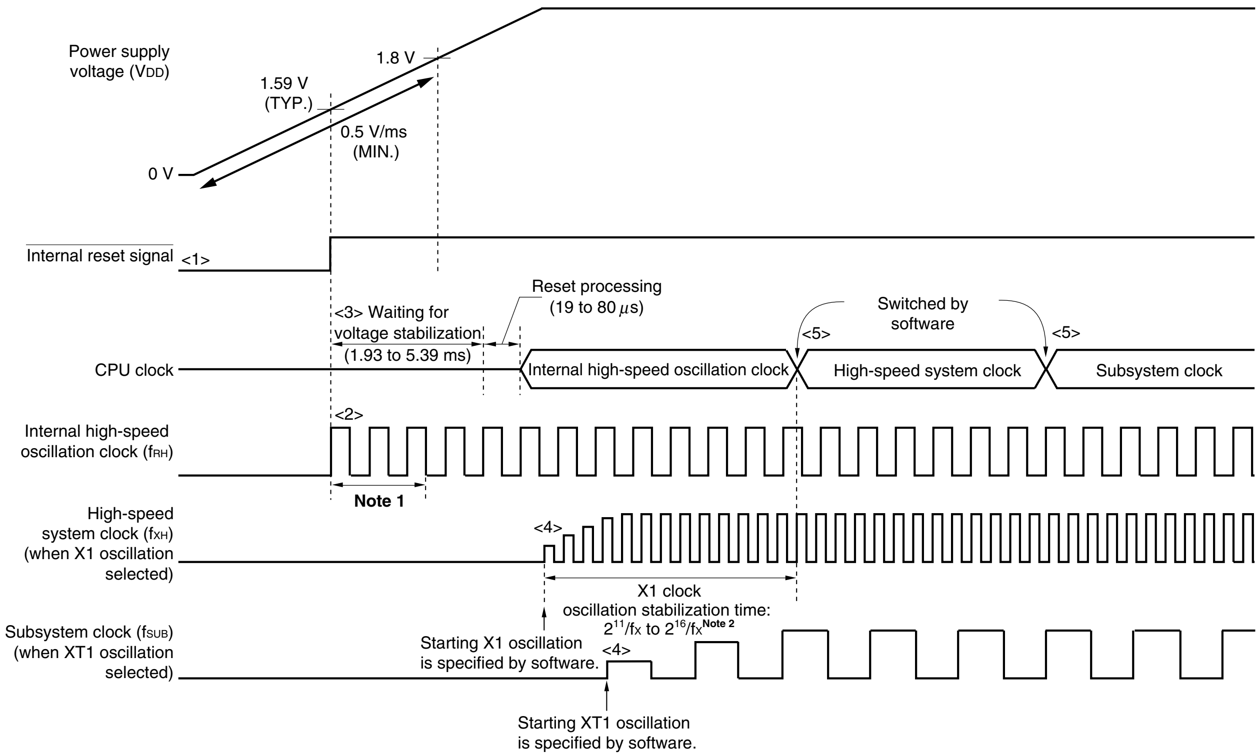
(1) Enhancement of security function

When the X1 clock is set as the CPU clock by the default setting, the device cannot operate if the X1 clock is damaged or badly connected and therefore does not operate after reset is released. However, the start clock of the CPU is the internal high-speed oscillation clock, so the device can be started by the internal high-speed oscillation clock after a reset release. Consequently, the system can be safely shut down by performing a minimum operation, such as acknowledging a reset source by software or performing safety processing when there is a malfunction.

(2) Improvement of performance

Because the CPU can be started without waiting for the X1 clock oscillation stabilization time, the total performance can be improved.
When the power supply voltage is turned on, the clock generator operation is shown in Figure 5-13.

<R> **Figure 5-13. Clock Generator Operation When Power Supply Voltage Is Turned On (When 1.59 V POC Mode Is Set (Option Byte: POCMODE = 0))**



<1> When the power is turned on, an internal reset signal is generated by the power-on-clear (POC) circuit.

<2> When the power supply voltage exceeds 1.59 V (TYP.), the reset is released and the internal high-speed oscillator automatically starts oscillation.

<3> When the power supply voltage rises with a slope of 0.5 V/ms (MIN.), the CPU starts operation on the internal high-speed oscillation clock after the reset is released and after the stabilization times for the voltage of the power supply and regulator have elapsed, and then reset processing is performed.

<4> Set the start of oscillation of the X1 or XT1 clock via software (see **(1)** in **5.6.1 Example of controlling high-speed system clock** and **(1)** in **5.6.3 Example of controlling subsystem clock)**.

<5> When switching the CPU clock to the X1 or XT1 clock, wait for the clock oscillation to stabilize, and then set switching via software (see **(3)** in **5.6.1 Example of controlling high-speed system clock** and **(3)** in **5.6.3 Example of controlling subsystem clock**).

Notes 1. The internal voltage stabilization time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

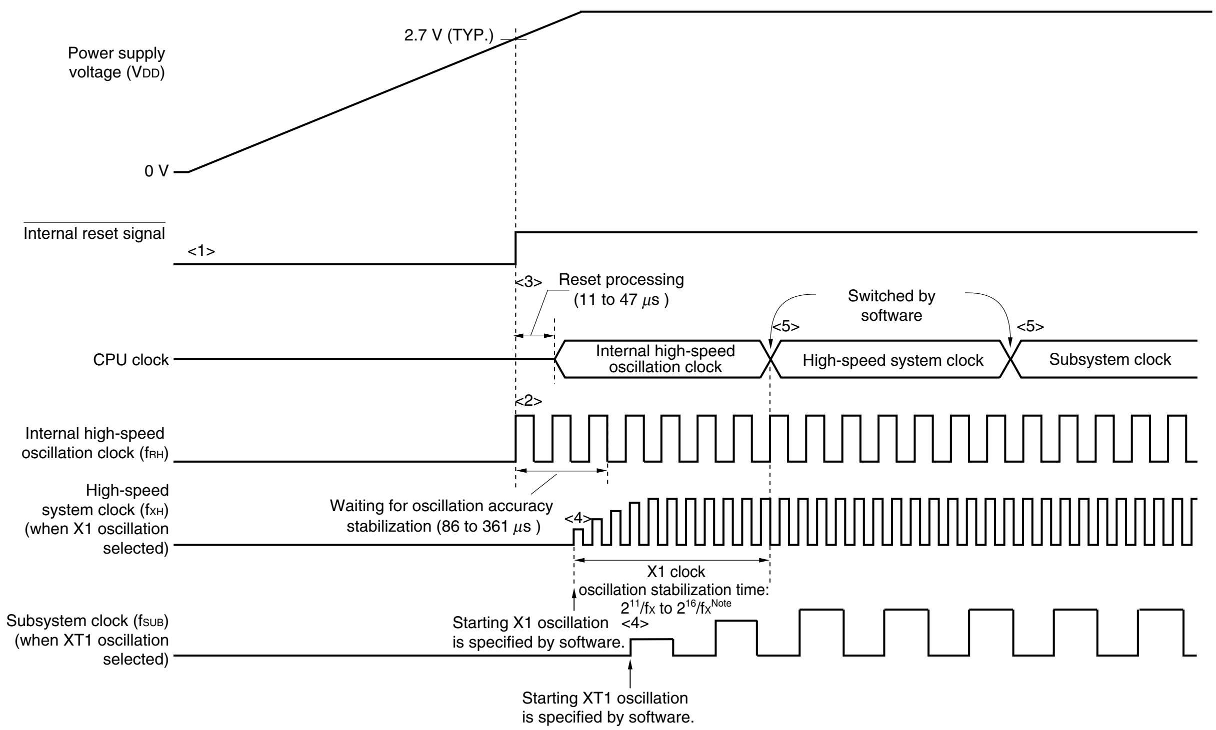
2. When releasing a reset (above figure) or releasing STOP mode while the CPU is operating on the internal high-speed oscillation clock, confirm the oscillation stabilization time for the X1 clock using the oscillation stabilization time counter status register (OSTC). If the CPU operates on the high-speed system clock (X1 oscillation), set the oscillation stabilization time when releasing STOP mode using the oscillation stabilization time select register (OSTS).

Cautions 1. If the voltage rises with a slope of less than 0.5 V/ms (MIN.) from power application until the voltage reaches 1.8 V, input a low level to the $\overline{\text{RESET}}$ pin from power application until the voltage reaches 1.8 V, or set the 2.7 V/1.59 V POC mode by using the option byte (POCMODE = 1) (see Figure 5-14). By doing so, the CPU operates with the same timing as <2> and thereafter in Figure 5-13 after reset release by the $\overline{\text{RESET}}$ pin.

2. It is not necessary to wait for the oscillation stabilization time when an external clock input from the EXCLK pin is used.

Remark While the microcontroller is operating, a clock that is not used as the CPU clock can be stopped via software settings. The internal high-speed oscillation clock and high-speed system clock can be stopped by executing the STOP instruction (see **(4)** in **5.6.1 Example of controlling high-speed system clock**, **(3)** in **5.6.2 Example of controlling internal high-speed oscillation clock**, and **(4)** in **5.6.3 Example of controlling subsystem clock**).

Figure 5-14. Clock Generator Operation When Power Supply Voltage Is Turned On (When 2.7 V/1.59 V POC Mode Is Set (Option Byte: POCMODE = 1))



<1> When the power is turned on, an internal reset signal is generated by the power-on-clear (POC) circuit.

<2> When the power supply voltage exceeds 2.7 V (TYP.), the reset is released and the internal high-speed oscillator automatically starts oscillation.

<3> After the reset is released and reset processing is performed, the CPU starts operation on the internal high-speed oscillation clock.

<4> Set the start of oscillation of the X1 or XT1 clock via software (see **(1)** in **5.6.1 Example of controlling high-speed system clock** and **(1)** in **5.6.3 Example of controlling subsystem clock)**.

<5> When switching the CPU clock to the X1 or XT1 clock, wait for the clock oscillation to stabilize, and then set switching via software (see **(3)** in **5.6.1 Example of controlling high-speed system clock** and **(3)** in **5.6.3 Example of controlling subsystem clock**).

Note When releasing a reset (above figure) or releasing STOP mode while the CPU is operating on the internal high-speed oscillation clock, confirm the oscillation stabilization time for the X1 clock using the oscillation stabilization time counter status register (OSTC). If the CPU operates on the high-speed system clock (X1 oscillation), set the oscillation stabilization time when releasing STOP mode using the oscillation stabilization time select register (OSTS).

<R> **Cautions 1. A voltage oscillation stabilization time of 1.93 to 5.39 ms is required after the supply voltage reaches 1.59 V (TYP.). If the time the supply voltage rises from 1.59 V (TYP.) to 2.7 V (TYP.) is within 1.93 to 5.39 ms, a power supply stabilization wait time of 0 to 5.39 ms occurs automatically before reset processing, and the reset processing time becomes 19 to 80 μs.**

2. It is not necessary to wait for the oscillation stabilization time when an external clock input from the EXCLK pin is used.

Remark While the microcontroller is operating, a clock that is not used as the CPU clock can be stopped via software settings. The internal high-speed oscillation clock and high-speed system clock can be stopped by executing the STOP instruction (see **(4)** in **5.6.1 Example of controlling high-speed system clock**, **(3)** in **5.6.2 Example of controlling internal high-speed oscillation clock**, and **(4)** in **5.6.3 Example of controlling subsystem clock**).

5.6 Controlling Clock

5.6.1 Example of controlling high-speed system clock

The following two types of high-speed system clocks are available.

- X1 clock: Crystal/ceramic resonator is connected across the X1 and X2 pins.
- External main system clock: External clock is input to the EXCLK pin.

When the high-speed system clock is not used, the OCD0A/X1/P121 and OCD0B/X2/EXCLK/P122 pins can be used as I/O port pins.

Caution The OCD0A/X1/P121 and OCD0B/X2/EXCLK/P122 pins are in the I/O port mode after a reset release.

The following describes examples of setting procedures for the following cases.

(1) When oscillating X1 clock
(2) When using external main system clock
(3) When using high-speed system clock as CPU clock and peripheral hardware clock
(4) When stopping high-speed system clock

(1) Example of setting procedure when oscillating the X1 clock

<1> Setting P121/X1 and P122/X2/EXCLK pins and selecting X1 clock or external clock (OSCCTL register)
When EXCLK is cleared to 0 and OSCSEL is set to 1, the mode is switched from port mode to X1 oscillation mode.

EXCLK	OSCSEL	Operation Mode of High-Speed System Clock Pin	P121/X1 Pin	P122/X2/EXCLK Pin
0	1	X1 oscillation mode	Crystal/ceramic resonator connection	

<2> Controlling oscillation of X1 clock (MOC register)
If MSTOP is cleared to 0, the X1 oscillator starts oscillating.

<3> Waiting for the stabilization of the oscillation of X1 clock
Check the OSTC register and wait for the necessary time.
During the wait time, other software processing can be executed with the internal high-speed oscillation clock.

Cautions 1. Do not change the value of EXCLK and OSCSEL while the X1 clock is operating.
2. Set the X1 clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 31 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)).

(2) Example of setting procedure when using the external main system clock

<1> Setting P121/X1 and P122/X2/EXCLK pins and selecting operation mode (OSCCTL register)

When EXCLK and OSCSEL are set to 1, the mode is switched from port mode to external clock input mode.

EXCLK	OSCSEL	Operation Mode of High-Speed System Clock Pin	P121/X1 Pin	P122/X2/EXCLK Pin
1	1	External clock input mode	I/O port	External clock input

<2> Controlling external main system clock input (MOC register)

When MSTOP is cleared to 0, the input of the external main system clock is enabled.

Cautions 1. Do not change the value of EXCLK and OSCSEL while the external main system clock is operating.

2. Set the external main system clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 31 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)).

(3) Example of setting procedure when using high-speed system clock as CPU clock and peripheral hardware clock

<1> Setting high-speed system clock oscillation[Note]

(See **5.6.1 (1) Example of setting procedure when oscillating the X1 clock** and **(2) Example of setting procedure when using the external main system clock.**)

Note The setting of <1> is not necessary when high-speed system clock is already operating.

<2> Setting the high-speed system clock as the main system clock (MCM register)

When XSEL and MCM0 are set to 1, the high-speed system clock is supplied as the main system clock and peripheral hardware clock.

XSEL	MCM0	Selection of Main System Clock and Clock Supplied to Peripheral Hardware	
		Main System Clock (f_{XP})	Peripheral Hardware Clock (f_{PRS})
1	1	High-speed system clock (f_{XH})	High-speed system clock (f_{XH})

Caution If the high-speed system clock is selected as the main system clock, a clock other than the high-speed system clock cannot be set as the peripheral hardware clock.

<3> Setting the main system clock as the CPU clock and selecting the division ratio (PCC register)

When CSS is cleared to 0, the main system clock is supplied to the CPU. To select the CPU clock division ratio, use PCC0, PCC1, and PCC2.

CSS	PCC2	PCC1	PCC0	CPU Clock (f_{CPU}) Selection
0	0	0	0	f_{XP}
	0	0	1	$f_{XP}/2$ (default)
	0	1	0	$f_{XP}/2^2$
	0	1	1	$f_{XP}/2^3$
	1	0	0	$f_{XP}/2^4$
	Other than above			Setting prohibited

(4) Example of setting procedure when stopping the high-speed system clock

The high-speed system clock can be stopped in the following two ways.

- Executing the STOP instruction to set the STOP mode
- Setting MSTOP to 1 and stopping the X1 oscillation (disabling clock input if the external clock is used)

(a) To execute a STOP instruction

<1> Setting to stop peripheral hardware

Stop peripheral hardware that cannot be used in the STOP mode (for peripheral hardware that cannot be used in STOP mode, see **CHAPTER 23 STANDBY FUNCTION**).

<2> Setting the X1 clock oscillation stabilization time after standby release

When the CPU is operating on the X1 clock, set the value of the OSTS register before the STOP instruction is executed.

<3> Executing the STOP instruction

When the STOP instruction is executed, the system is placed in the STOP mode and X1 oscillation is stopped (the input of the external clock is disabled).

(b) To stop X1 oscillation (disabling external clock input) by setting MSTOP to 1

<1> Confirming the CPU clock status (PCC and MCM registers)

Confirm with CLS and MCS that the CPU is operating on a clock other than the high-speed system clock.

When CLS = 0 and MCS = 1, the high-speed system clock is supplied to the CPU, so change the CPU clock to the subsystem clock or internal high-speed oscillation clock.

CLS	MCS	CPU Clock Status
0	0	Internal high-speed oscillation clock
0	1	High-speed system clock
1	×	Subsystem clock

<2> Stopping the high-speed system clock (MOC register)

When MSTOP is set to 1, X1 oscillation is stopped (the input of the external clock is disabled).

Caution Be sure to confirm that MCS = 0 or CLS = 1 when setting MSTOP to 1. In addition, stop peripheral hardware that is operating on the high-speed system clock.

5.6.2 Example of controlling internal high-speed oscillation clock

The following describes examples of clock setting procedures for the following cases.

(1) When restarting oscillation of the internal high-speed oscillation clock

(2) When using internal high-speed oscillation clock as CPU clock, and internal high-speed oscillation clock or high-speed system clock as peripheral hardware clock

(3) When stopping the internal high-speed oscillation clock

(1) Example of setting procedure when restarting oscillation of the internal high-speed oscillation clock[Note 1]

<1> Setting restart of oscillation of the internal high-speed oscillation clock (RCM register)
When RSTOP is cleared to 0, the internal high-speed oscillation clock starts operating.

<2> Waiting for the oscillation accuracy stabilization time of internal high-speed oscillation clock (RCM register)
Wait until RSTS is set to 1[Note 2].

Notes 1. After a reset release, the internal high-speed oscillator automatically starts oscillating and the internal high-speed oscillation clock is selected as the CPU clock.

2. This wait time is not necessary if high accuracy is not necessary for the CPU clock and peripheral hardware clock.

(2) Example of setting procedure when using internal high-speed oscillation clock as CPU clock, and internal high-speed oscillation clock or high-speed system clock as peripheral hardware clock

<1>
- Restarting oscillation of the internal high-speed oscillation clock[Note]
  (See **5.6.2 (1) Example of setting procedure when restarting internal high-speed oscillation clock**).
- Oscillating the high-speed system clock[Note]
  (This setting is required when using the high-speed system clock as the peripheral hardware clock. See **5.6.1 (1) Example of setting procedure when oscillating the X1 clock** and **(2) Example of setting procedure when using the external main system clock.**)

Note The setting of <1> is not necessary when the internal high-speed oscillation clock or high-speed system clock is already operating.

<2> Selecting the clock supplied as the main system clock and peripheral hardware clock (MCM register)
Set the main system clock and peripheral hardware clock using XSEL and MCM0.

XSEL	MCM0	Selection of Main System Clock and Clock Supplied to Peripheral Hardware	
		Main System Clock (f_{XP})	Peripheral Hardware Clock (f_{PRS})
0	0	Internal high-speed oscillation clock (f_{RH})	Internal high-speed oscillation clock (f_{RH})
0	1		
1	0		High-speed system clock (f_{XH})

<3> Selecting the CPU clock division ratio (PCC register)
When CSS is cleared to 0, the main system clock is supplied to the CPU. To select the CPU clock division ratio, use PCC0, PCC1, and PCC2.

CSS	PCC2	PCC1	PCC0	CPU Clock (f_{CPU}) Selection
0	0	0	0	f_{XP}
	0	0	1	$f_{XP}/2$ (default)
	0	1	0	$f_{XP}/2^2$
	0	1	1	$f_{XP}/2^3$
	1	0	0	$f_{XP}/2^4$
	Other than above			Setting prohibited

(3) Example of setting procedure when stopping the internal high-speed oscillation clock

The internal high-speed oscillation clock can be stopped in the following two ways.

- Executing the STOP instruction to set the STOP mode
- Setting RSTOP to 1 and stopping the internal high-speed oscillation clock

(a) To execute a STOP instruction

<1> Setting of peripheral hardware

Stop peripheral hardware that cannot be used in the STOP mode (for peripheral hardware that cannot be used in STOP mode, see **CHAPTER 23 STANDBY FUNCTION**).

<2> Setting the X1 clock oscillation stabilization time after standby release

When the CPU is operating on the X1 clock, set the value of the OSTS register before the STOP instruction is executed.

<3> Executing the STOP instruction

When the STOP instruction is executed, the system is placed in the STOP mode and internal high-speed oscillation clock is stopped.

(b) To stop internal high-speed oscillation clock by setting RSTOP to 1

<1> Confirming the CPU clock status (PCC and MCM registers)

Confirm with CLS and MCS that the CPU is operating on a clock other than the internal high-speed oscillation clock.

When CLS = 0 and MCS = 0, the internal high-speed oscillation clock is supplied to the CPU, so change the CPU clock to the high-speed system clock or subsystem clock.

CLS	MCS	CPU Clock Status
0	0	Internal high-speed oscillation clock
0	1	High-speed system clock
1	×	Subsystem clock

<2> Stopping the internal high-speed oscillation clock (RCM register)

When RSTOP is set to 1, internal high-speed oscillation clock is stopped.

Caution Be sure to confirm that MCS = 1 or CLS = 1 when setting RSTOP to 1. In addition, stop peripheral hardware that is operating on the internal high-speed oscillation clock.

5.6.3 Example of controlling subsystem clock

The following two types of subsystem clocks are available.

- XT1 clock: Crystal/ceramic resonator is connected across the XT1 and XT2 pins.

When the subsystem clock is not used, the XT1/P123 and XT2/P124 pins can be used as Input port pins.

Caution The XT1/P123 and XT2/P124 pins are in the Input port mode after a reset release.

The following describes examples of setting procedures for the following cases.

(1) When oscillating XT1 clock

(2) When using subsystem clock as CPU clock

(3) When stopping subsystem clock

(1) Example of setting procedure when oscillating the XT1 clock

<1> Setting XT1 and XT2 pins and selecting operation mode (PCC and OSCCTL registers)

When OSCSELS is set as any of the following, the mode is switched from port mode to XT1 oscillation mode.

OSCSELS	Operation Mode of Subsystem Clock Pin	P123/XT1 Pin	P124/XT2 Pin
1	XT1 oscillation mode	Crystal/ceramic resonator connection	

<2> Waiting for the stabilization of the subsystem clock oscillation

Wait for the oscillation stabilization time of the subsystem clock by software, using a timer function.

Caution Do not change the value of OSCSELS while the subsystem clock is operating.

(2) Example of setting procedure when using the subsystem clock as the CPU clock

<1> Setting subsystem clock oscillation[Note]

(See **5.6.3 (1) Example of setting procedure when oscillating the XT1 clock**)

Note The setting of <1> is not necessary when while the subsystem clock is operating.

<2> Switching the CPU clock (PCC register)

When CSS is set to 1, the subsystem clock is supplied to the CPU.

CSS	PCC2	PCC1	PCC0	CPU Clock (f_{CPU}) Selection
1	0	0	0	$f_{SUB}/2$
	0	0	1	
	0	1	0	
	0	1	1	
	1	0	0	
	Other than above			Setting prohibited

(3) Example of setting procedure when stopping the subsystem clock

<1> Confirming the CPU clock status (PCC and MCM registers)

Confirm with CLS and MCS that the CPU is operating on a clock other than the subsystem clock.

When CLS = 1, the subsystem clock is supplied to the CPU, so change the CPU clock to the internal high-speed oscillation clock or high-speed system clock.

CLS	MCS	CPU Clock Status
0	0	Internal high-speed oscillation clock
0	1	High-speed system clock
1	×	Subsystem clock

<2> Stopping the subsystem clock (OSCCTL register)

When OSCSELS is cleared to 0, XT1 oscillation is stopped.

Cautions 1. Be sure to confirm that CLS = 0 when clearing OSCSELS to 0. In addition, stop the peripheral hardware if it is operating on the subsystem clock.

2. The subsystem clock oscillation cannot be stopped using the STOP instruction.

5.6.4 Example of controlling internal low-speed oscillation clock

The internal low-speed oscillation clock cannot be used as the CPU clock.

Only the following peripheral hardware can operate with this clock.

- Watchdog timer
- 8-bit timer H1 (if f_{RL}, $f_{RL}/2^7$ or $f_{RL}/2^9$ is selected as the count clock)
- LCD controller/driver (if $f_{RL}/2^3$ is selected as the LCD source clock)

In addition, the following operation modes can be selected by the option byte.

- Internal low-speed oscillator cannot be stopped
- Internal low-speed oscillator can be stopped by software

The internal low-speed oscillator automatically starts oscillation after a reset release, and the watchdog timer is driven (240 kHz (TYP.)) if the watchdog timer operation has been enabled by the option byte.

(1) Example of setting procedure when stopping the internal low-speed oscillation clock

<1> Setting LSRSTOP to 1 (RCM register)

When LSRSTOP is set to 1, the internal low-speed oscillation clock is stopped.

(2) Example of setting procedure when restarting oscillation of the internal low-speed oscillation clock

<1> Clearing LSRSTOP to 0 (RCM register)

When LSRSTOP is cleared to 0, the internal low-speed oscillation clock is restarted.

Caution If "Internal low-speed oscillator cannot be stopped" is selected by the option byte, oscillation of the internal low-speed oscillation clock cannot be controlled.

5.6.5 Clocks supplied to CPU and peripheral hardware

The following table shows the relation among the clocks supplied to the CPU and peripheral hardware, and setting of registers.

Table 5-4. Clocks Supplied to CPU and Peripheral Hardware, and Register Setting

Supplied Clock		XSEL	CSS	MCM0	EXCLK
Clock Supplied to CPU	Clock Supplied to Peripheral Hardware				
Internal high-speed oscillation clock		0	0	×	×
Internal high-speed oscillation clock	X1 clock	1	0	0	0
	External main system clock	1	0	0	1
X1 clock		1	0	1	0
External main system clock		1	0	1	1
Subsystem clock	Internal high-speed oscillation clock	0	1	×	×
	X1 clock	1	1	0	0
		1	1	1	0
	External main system clock	1	1	0	1
		1	1	1	1

Remarks 1. XSEL: Bit 2 of the main clock mode register (MCM)

2. CSS: Bit 4 of the processor clock control register (PCC)

3. MCM0: Bit 0 of MCM

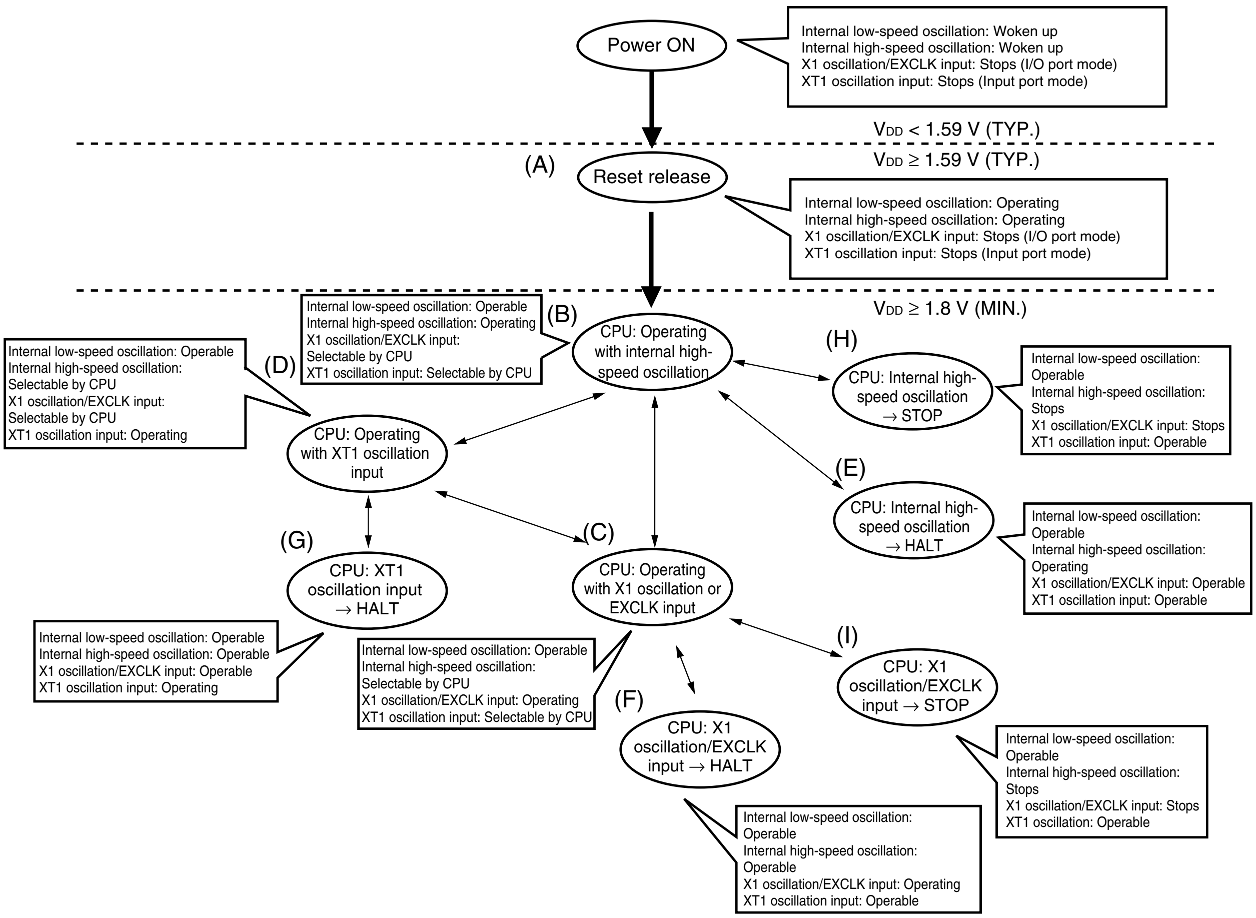
4. EXCLK: Bit 7 of the clock operation mode select register (OSCCTL)

5. ×: don't care

5.6.6 CPU clock status transition diagram

Figure 5-15 shows the CPU clock status transition diagram of this product.

Figure 5-15. CPU Clock Status Transition Diagram
(When 1.59 V POC Mode Is Set (Option Byte: POCMODE = 0))



Remark In the 2.7 V/1.59 V POC mode (option byte: POCMODE = 1), the CPU clock status changes to (A) in the above figure when the supply voltage exceeds 2.7 V (TYP.), and to (B) after reset processing (11 to 47 μs (TYP.)).

Table 5-5 shows transition of the CPU clock and examples of setting the SFR registers.

Table 5-5. CPU Clock Transition and SFR Register Setting Examples (1/4)

(1) CPU operating with internal high-speed oscillation clock (B) after reset release (A)

Status Transition	SFR Register Setting
(A) → (B)	SFR registers do not have to be set (default status after reset release).

(2) CPU operating with high-speed system clock (C) after reset release (A)
(The CPU operates with the internal high-speed oscillation clock immediately after a reset release (B).)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	EXCLK	OSCSEL	MSTOP	OSTC Register	XSEL	MCM0
(A) → (B) → (C) (X1 clock)	0	1	0	Must be checked	1	1
(A) → (B) → (C) (external main clock)	1	1	0	Must not be checked	1	1

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 31 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)).

(3) CPU operating with subsystem clock (D) after reset release (A)
(The CPU operates with the internal high-speed oscillation clock immediately after a reset release (B).)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	OSCSELS	Waiting for Oscillation Stabilization	CSS
(A) → (B) → (D)	1	Necessary	1

Remarks 1. (A) to (I) in Table 5-5 correspond to (A) to (I) in Figure 5-15.
2. EXCLK, OSCSEL, OSCSELS:
Bits 7, 6, and 4 of the clock operation mode select register (OSCCTL)
MSTOP: Bit 7 of the main OSC control register (MOC)
XSEL, MCM0: Bits 2 and 0 of the main clock mode register (MCM)
CSS: Bit 4 of the processor clock control register (PCC)

Table 5-5. CPU Clock Transition and SFR Register Setting Examples (2/4)

(4) CPU clock changing from internal high-speed oscillation clock (B) to high-speed system clock (C)

(Setting sequence of SFR registers) →

Status Transition \ Setting Flag of SFR Register	EXCLK	OSCSEL	MSTOP	OSTC Register	XSEL[Note]	MCM0
(B) → (C) (X1 clock)	0	1	0	Must be checked	1	1
(B) → (C) (external main clock)	1	1	0	Must not be checked	1	1

EXCLK, OSCSEL: Unnecessary if these registers are already set

MSTOP, OSTC Register: Unnecessary if the CPU is operating with the high-speed system clock

Note The value of this flag can be changed only once after a reset release. This setting is not necessary if it has already been set.

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 31 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)).

(5) CPU clock changing from internal high-speed oscillation clock (B) to subsystem clock (D)

(Setting sequence of SFR registers) →

Status Transition \ Setting Flag of SFR Register	OSCSELS	Waiting for Oscillation Stabilization	CSS
(B) → (D)	1	Necessary	1

Remarks 1. (A) to (I) in Table 5-5 correspond to (A) to (I) in Figure 5-15.

2. EXCLK, OSCSEL, OSCSELS:
Bits 7, 6, and 4 of the clock operation mode select register (OSCCTL)
MSTOP: Bit 7 of the main OSC control register (MOC)
XSEL, MCM0: Bits 2 and 0 of the main clock mode register (MCM)
CSS: Bit 4 of the processor clock control register (PCC)

Table 5-5. CPU Clock Transition and SFR Register Setting Examples (3/4)

(6) CPU clock changing from high-speed system clock (C) to internal high-speed oscillation clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	RSTOP	RSTS	MCM0
(C) → (B)	0	Confirm this flag is 1.	0

(RSTOP, RSTS:) Unnecessary if the CPU is operating with the internal high-speed oscillation clock

(7) CPU clock changing from high-speed system clock (C) to subsystem clock (D)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	OSCSELS	Waiting for Oscillation Stabilization	CSS
(C) → (D)	1	Necessary	1

(OSCSELS, Waiting for Oscillation Stabilization:) Unnecessary if the CPU is operating with the subsystem clock

(8) CPU clock changing from subsystem clock (D) to internal high-speed oscillation clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	RSTOP	RSTS	MCM0	CSS
(D) → (B)	0	Confirm this flag is 1.	0	0

(RSTOP, RSTS:) Unnecessary if the CPU is operating with the internal high-speed oscillation clock

(MCM0:) ↑ Unnecessary if XSEL is 0

Remarks 1. (A) to (I) in Table 5-5 correspond to (A) to (I) in Figure 5-15.

2. MCM0: Bit 0 of the main clock mode register (MCM)

OSCSELS: Bit 4 of the clock operation mode select register (OSCCTL)

RSTS, RSTOP: Bits 7 and 0 of the internal oscillation mode register (RCM)

CSS: Bit 4 of the processor clock control register (PCC)

Table 5-5. CPU Clock Transition and SFR Register Setting Examples (4/4)

(9) CPU clock changing from subsystem clock (D) to high-speed system clock (C)

(Setting sequence of SFR registers) →

Status Transition \ Setting Flag of SFR Register	EXCLK	OSCSEL	MSTOP	OSTC Register	XSEL[Note]	MCM0	CSS
(D) → (C) (X1 clock)	0	1	0	Must be checked	1	1	0
(D) → (C) (external main clock)	1	1	0	Must not be checked	1	1	0

EXCLK, OSCSEL: Unnecessary if these registers are already set

MSTOP, OSTC Register: Unnecessary if the CPU is operating with the high-speed system clock

Note The value of this flag can be changed only once after a reset release. This setting is not necessary if it has already been set.

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 31 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)).

(10) • HALT mode (E) set while CPU is operating with internal high-speed oscillation clock (B)
• HALT mode (F) set while CPU is operating with high-speed system clock (C)
• HALT mode (G) set while CPU is operating with subsystem clock (D)

Status Transition	Setting
(B) → (E) (C) → (F) (D) → (G)	Executing HALT instruction

(11) • STOP mode (H) set while CPU is operating with internal high-speed oscillation clock (B)
• STOP mode (I) set while CPU is operating with high-speed system clock (C)

(Setting sequence) →

Status Transition	Setting	
(B) → (H) (C) → (I)	Stopping peripheral functions that cannot operate in STOP mode	Executing STOP instruction

Remarks 1. (A) to (I) in Table 5-5 correspond to (A) to (I) in Figure 5-15.

2. EXCLK, OSCSEL: Bits 7 and 6 of the clock operation mode select register (OSCCTL)
MSTOP: Bit 7 of the main OSC control register (MOC)
XSEL, MCM0: Bits 2 and 0 of the main clock mode register (MCM)
CSS: Bit 4 of the processor clock control register (PCC)

5.6.7 Condition before changing CPU clock and processing after changing CPU clock

Condition before changing the CPU clock and processing after changing the CPU clock are shown below.

Table 5-6. Changing CPU Clock

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
Internal high-speed oscillation clock	X1 clock	Stabilization of X1 oscillation • MSTOP = 0, OSCSEL = 1, EXCLK = 0 • After elapse of oscillation stabilization time	• Internal high-speed oscillator can be stopped (RSTOP = 1).
	External main system clock	Enabling input of external clock from EXCLK pin • MSTOP = 0, OSCSEL = 1, EXCLK = 1	
X1 clock	Internal high-speed oscillation clock	Oscillation of internal high-speed oscillator • RSTOP = 0	X1 oscillation can be stopped (MSTOP = 1).
External main system clock			External main system clock input can be disabled (MSTOP = 1).
Internal high-speed oscillation clock	XT1 clock	Stabilization of XT1 oscillation • OSCSELS = 1 • After elapse of oscillation stabilization time	Operating current can be reduced by stopping internal high-speed oscillator (RSTOP = 1).
X1 clock			X1 oscillation can be stopped (MSTOP = 1).
External main system clock			External main system clock input can be disabled (MSTOP = 1).
XT1 clock	Internal high-speed oscillation clock	Oscillation of internal high-speed oscillator and selection of internal high-speed oscillation clock as main system clock • RSTOP = 0, MCS = 0	XT1 oscillation can be stopped (OSCSELS = 0).
	X1 clock	Stabilization of X1 oscillation and selection of high-speed system clock as main system clock • MSTOP = 0, OSCSEL = 1, EXCLK = 0 • After elapse of oscillation stabilization time • MCS = 1	
	External main system clock	Enabling input of external clock from EXCLK pin and selection of high-speed system clock as main system clock • MSTOP = 0, OSCSEL = 1, EXCLK = 1 • MCS = 1	

5.6.8 Time required for switchover of CPU clock and main system clock

By setting bits 0 to 2 (PCC0 to PCC2) and bit 4 (CSS) of the processor clock control register (PCC), the CPU clock can be switched (between the main system clock and the subsystem clock) and the division ratio of the main system clock can be changed.

The actual switchover operation is not performed immediately after rewriting to PCC; operation continues on the pre-switchover clock for several clocks (see **Table 5-7**).

Whether the CPU is operating on the main system clock or the subsystem clock can be ascertained using bit 5 (CLS) of the PCC register.

Table 5-7. Time Required for Switchover of CPU Clock and Main System Clock Cycle Division Factor

Set Value Before Switchover				Set Value After Switchover					
CSS	PCC2	PCC1	PCC0	CSS PCC2 PCC1 PCC0	CSS PCC2 PCC1 PCC0	CSS PCC2 PCC1 PCC0	CSS PCC2 PCC1 PCC0	CSS PCC2 PCC1 PCC0	CSS PCC2 PCC1 PCC0
				0 0 0 0	0 0 0 1	0 0 1 0	0 0 1 1	0 1 0 0	1 × × ×
0	0	0	0		16 clocks	16 clocks	16 clocks	16 clocks	$2f_{XP}/f_{SUB}$ clocks
	0	0	1	8 clocks		8 clocks	8 clocks	8 clocks	f_{XP}/f_{SUB} clocks
	0	1	0	4 clocks	4 clocks		4 clocks	4 clocks	$f_{XP}/2f_{SUB}$ clocks
	0	1	1	2 clocks	2 clocks	2 clocks		2 clocks	$f_{XP}/4f_{SUB}$ clocks
	1	0	0	1 clock	1 clock	1 clock	1 clock		$f_{XP}/8f_{SUB}$ clocks
1	×	×	×	2 clocks	2 clocks	2 clocks	2 clocks	2 clocks	

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

Remarks 1. The number of clocks listed in Table 5-7 is the number of CPU clocks before switchover.

2. When switching the CPU clock from the main system clock to the subsystem clock, calculate the number of clocks by rounding up to the next clock and discarding the decimal portion, as shown below.

Example When switching CPU clock from $f_{XP}/2$ to $f_{SUB}/2$ (@ oscillation with f_{XP} = 10 MHz, f_{SUB} = 32.768 kHz)

$f_{XP}/f_{SUB} = 10000/32.768 \cong 305.1 \rightarrow 306$ clocks

By setting bit 0 (MCM0) of the main clock mode register (MCM), the main system clock can be switched (between the internal high-speed oscillation clock and the high-speed system clock).

The actual switchover operation is not performed immediately after rewriting to MCM0; operation continues on the pre-switchover clock for several clocks (see **Table 5-8**).

Whether the CPU is operating on the internal high-speed oscillation clock or the high-speed system clock can be ascertained using bit 1 (MCS) of MCM.

Table 5-8. Maximum Time Required for Main System Clock Switchover

Set Value Before Switchover	Set Value After Switchover	
MCM0	MCM0	
	0	1
0		1 + $2f_{RH}/f_{XH}$ clock
1	1 + $2f_{XH}/f_{RH}$ clock	

Caution When switching the internal high-speed oscillation clock to the high-speed system clock, bit 2 (XSEL) of MCM must be set to 1 in advance. The value of XSEL can be changed only once after a reset release.

Remarks 1. The number of clocks listed in Table 5-8 is the number of main system clocks before switchover.

2. Calculate the number of clocks in Table 5-8 by removing the decimal portion.

Example When switching the main system clock from the internal high-speed oscillation clock to the high-speed system clock (@ oscillation with f_{RH} = 8 MHz, f_{XH} = 10 MHz)

$1 + 2f_{RH}/f_{XH} = 1 + 2 \times 8/10 = 1 + 2 \times 0.8 = 1 + 1.6 = 2.6 \rightarrow 2$ clocks

5.6.9 Conditions before clock oscillation is stopped

The following lists the register flag settings for stopping the clock oscillation (disabling external clock input) and conditions before the clock oscillation is stopped.

Table 5-9. Conditions Before the Clock Oscillation Is Stopped and Flag Settings

Clock	Conditions Before Clock Oscillation Is Stopped (External Clock Input Disabled)	Flag Settings of SFR Register
Internal high-speed oscillation clock	MCS = 1 or CLS = 1 (The CPU is operating on a clock other than the internal high-speed oscillation clock)	RSTOP = 1
X1 clock	MCS = 0 or CLS = 1 (The CPU is operating on a clock other than the high-speed system clock)	MSTOP = 1
External main system clock		
XT1 clock	CLS = 0 (The CPU is operating on a clock other than the subsystem clock)	OSCSELS = 0

5.6.10 Peripheral hardware and source clocks

The following lists peripheral hardware and source clocks incorporated in the 78K0/LF3.

Table 5-10. Peripheral Hardware and Source Clocks

Peripheral Hardware \ Source Clock		Peripheral Hardware Clock (f_{PRS})	Subsystem Clock (f_{SUB})	Internal Low-Speed Oscillation Clock (f_{RL})	TM50 Output	TM52 Output	TMH1 Output	External Clock from Peripheral Hardware Pins
16-bit timer/ event counter	00	Y	Y	N	N	Y	N	Y (TI000 pin)Note
8-bit timer/ event counter	50	Y	N	N	N	N	N	Y (TI50 pin)Note
	51	Y	N	N	N	N	Y	Y (TI51 pin)Note
	52	Y	N	N	N	N	N	Y (TI52 pin)Note
8-bit timer	H0	Y	N	N	Y	N	N	N
	H1	Y	N	Y	N	N	N	N
	H2	Y	N	N	N	N	N	N
Real-time counter		Y	Y	N	N	N	N	N
Watchdog timer		N	N	Y	N	N	N	N
Buzzer output		Y	N	N	N	N	N	N
Clock output		Y	Y	N	N	N	N	N
Successive approximation type A/D converter		Y	N	N	N	N	N	N
ΔΣ type A/D converter		Y	Y	N	N	N	N	N
Serial interface	UART0	Y	N	N	Y	N	N	N
	UART6	Y	N	N	Y	N	N	N
	CSI10	Y	N	N	N	N	N	Y ($\overline{SCK10}$ pin)Note
	CSIA0	Y	N	N	N	N	N	Y ($\overline{SCKA0}$ pin)Note
LCD controller/driver		Y	Y	Y	N	N	N	N
Manchester code generator		Y	N	N	N	N	N	N
Remote controller receiver		Y	Y	N	N	N	N	N

Note When the CPU is operating on the subsystem clock and the internal high-speed oscillation clock has been stopped, do not start operation of these functions on the external clock input from peripheral hardware pins.

Remark Y: Can be selected, N: Cannot be selected

6.1 Functions of 16-Bit Timer/Event Counter 00

16-bit timer/event counter 00 has the following functions.

(1) Interval timer

16-bit timer/event counter 00 generates an interrupt request at the preset time interval.

(2) Square-wave output

16-bit timer/event counter 00 can output a square wave with any selected frequency.

(3) External event counter

16-bit timer/event counter 00 can measure the number of pulses of an externally input signal.

(4) One-shot pulse output

16-bit timer event counter 00 can output a one-shot pulse whose output pulse width can be set freely.

(5) PPG output

16-bit timer/event counter 00 can output a rectangular wave whose frequency and output pulse width can be set freely.

(6) Pulse width measurement

16-bit timer/event counter 00 can measure the pulse width of an externally input signal.

(7) External 24-bit event counter

16-bit timer/event counter 00 can be operated to function as an external 24-bit event counter, by connecting 16-bit timer 00 and 8-bit timer/event counter 52 in cascade, and using the external event counter function of 8-bit timer/event counter 52.

When using it as an external 24-bit event counter, external event input gate enable can be controlled via 8-bit timer counter H2 output.

6.2 Configuration of 16-Bit Timer/Event Counter 00

16-bit timer/event counter 00 includes the following hardware.

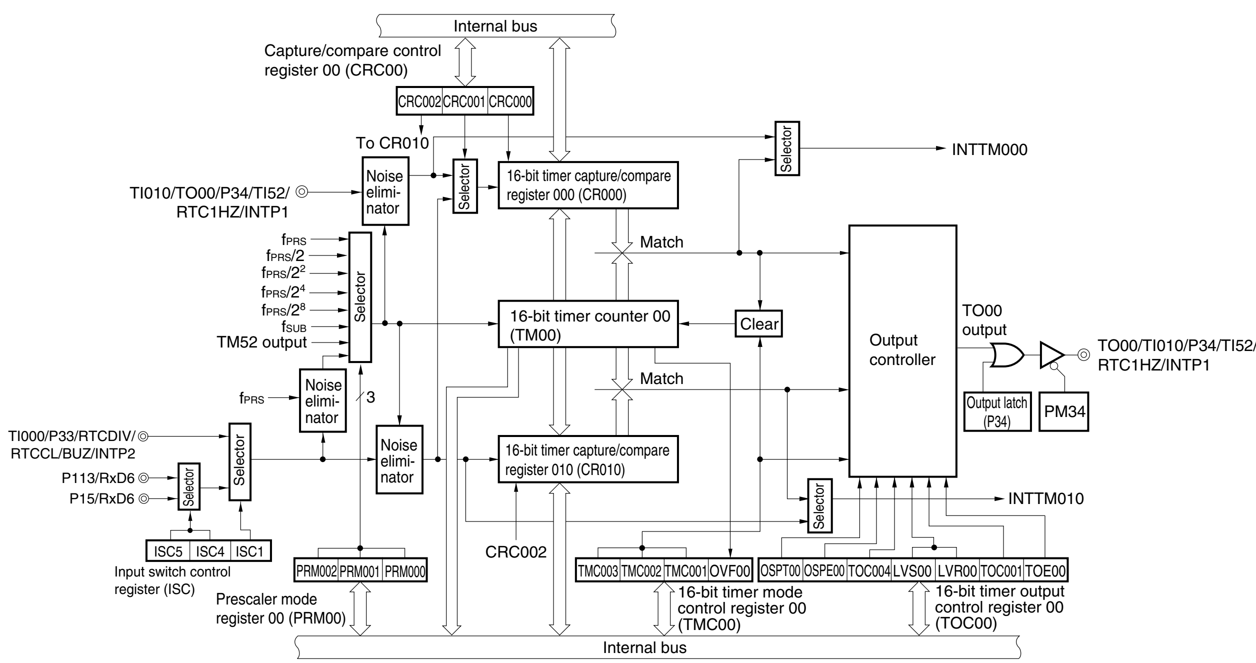
Table 6-1. Configuration of 16-Bit Timer/Event Counter 00

Item	Configuration
Time/counter	16-bit timer counter 00 (TM00)
Register	16-bit timer capture/compare registers 000, 010 (CR000, CR010)
Timer input	TI000, TI010 pins
Timer output	TO00 pin, output controller
Control registers	16-bit timer mode control register 00 (TMC00) Capture/compare control register 00 (CRC00) 16-bit timer output control register 00 (TOC00) Prescaler mode register 00 (PRM00) Input switch control register (ISC) Port mode register 3 (PM3) Port register 3 (P3)

Remark When using 16-bit timer/event counter 00 as an external 24-bit event counter, 8-bit timer/event counter 52 (TM52) and 8-bit timer counter H2 (TMH2) are used. For details, see **6.4.9 External 24-bit event counter operation**.

Figures 6-1 shows the block diagrams.

Figure 6-1. Block Diagram of 16-Bit Timer/Event Counter 00



Cautions 1. The valid edge of TI010 and timer output (TO00) cannot be used for the P34 pin at the same time. Select either of the functions.

Cautions 2. If clearing of bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) to 00 and input of the capture trigger conflict, then the captured data is undefined.

3. To change the mode from the capture mode to the comparison mode, first clear the TMC003 and TMC002 bits to 00, and then change the setting.
A value that has been once captured remains stored in CR000 unless the device is reset. If the mode has been changed to the comparison mode, be sure to set a comparison value.

(1) 16-bit timer counter 00 (TM00)

TM00 is a 16-bit read-only register that counts count pulses.
The counter is incremented in synchronization with the rising edge of the count clock.

Figure 6-2. Format of 16-Bit Timer Counter 00 (TM00)

Address: FF10H, FF11H After reset: 0000H R

	FF11H								FF10H							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TM00																

The count value of TM00 can be read by reading TM00 when the value of bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) is other than 00. The value of TM00 is 0000H if it is read when TMC003 and TMC002 = 00.
The count value is reset to 0000H in the following cases.

- At reset signal generation
- If TMC003 and TMC002 are cleared to 00
- If the valid edge of the TI000 pin is input in the mode in which the clear & start occurs when inputting the valid edge to the TI000 pin
- If TM00 and CR000 match in the mode in which the clear & start occurs when TM00 and CR000 match
- OSPT00 is set to 1 in one-shot pulse output mode or the valid edge is input to the TI000 pin

Caution Even if TM00 is read, the value is not captured by CR010.

(2) 16-bit timer capture/compare register 000 (CR000), 16-bit timer capture/compare register 010 (CR010)

CR000 and CR010 are 16-bit registers that are used with a capture function or comparison function selected by using CRC00.
Change the value of CR000 while the timer is stopped (TMC003 and TMC002 = 00).
The value of CR010 can be changed during operation if the value has been set in a specific way. For details, see **6.5.1 Rewriting CR010 during TM00 operation**.
These registers can be read or written in 16-bit units.
Reset signal generation sets these registers to 0000H.

Figure 6-3. Format of 16-Bit Timer Capture/Compare Register 000 (CR000)

Address: FF12H, FF13H After reset: 0000H R/W

	FF13H								FF12H							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR000																

(i) When CR000 is used as a compare register

The value set in CR000 is constantly compared with the TM00 count value, and an interrupt request signal (INTTM000) is generated if they match. The value is held until CR000 is rewritten.

Caution CR000 does not perform the capture operation when it is set in the comparison mode, even if a capture trigger is input to it.

(ii) When CR000 is used as a capture register

The count value of TM00 is captured to CR000 when a capture trigger is input.

As the capture trigger, an edge of a phase reverse to that of the TI000 pin or the valid edge of the TI010 pin can be selected by using CRC00 or PRM00.

Figure 6-4. Format of 16-Bit Timer Capture/Compare Register 010 (CR010)

Address: FF14H, FF15H After reset: 0000H R/W

	FF15H								FF14H							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR010																

(i) When CR010 is used as a compare register

The value set in CR010 is constantly compared with the TM00 count value, and an interrupt request signal (INTTM010) is generated if they match.

Caution CR010 does not perform the capture operation when it is set in the comparison mode, even if a capture trigger is input to it.

(ii) When CR010 is used as a capture register

The count value of TM00 is captured to CR010 when a capture trigger is input.

It is possible to select the valid edge of the TI000 pin as the capture trigger. The TI000 pin valid edge is set by PRM00.

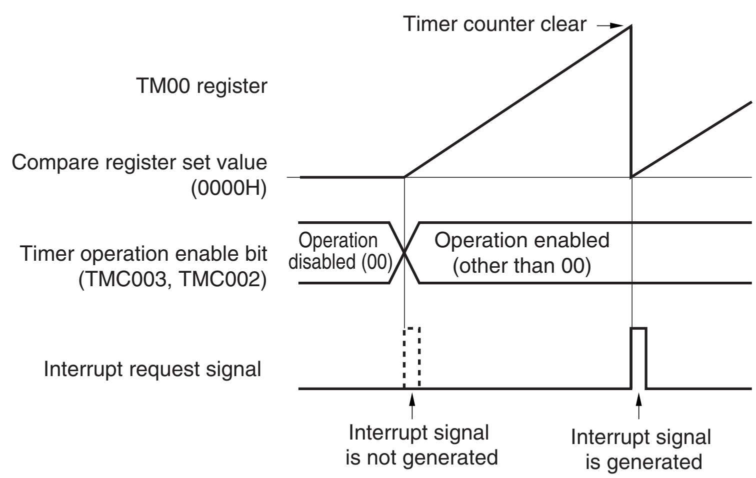
(iii) Setting range when CR000 or CR010 is used as a compare register

When CR000 or CR010 is used as a compare register, set it as shown below.

Operation	CR000 Register Setting Range	CR010 Register Setting Range
Operation as interval timer	0000H < N ≤ FFFFH	0000H[Note] ≤ M ≤ FFFFH Normally, this setting is not used. Mask the match interrupt signal (INTTM010).
Operation as square-wave output		
Operation as external event counter		
Operation in the clear & start mode entered by TI000 pin valid edge input	0000H[Note] ≤ N ≤ FFFFH	0000H[Note] ≤ M ≤ FFFFH
Operation as free-running timer		
Operation as PPG output	M < N ≤ FFFFH	0000H[Note] ≤ M < N
Operation as one-shot pulse output	0000H[Note] ≤ N ≤ FFFFH (N ≠ M)	0000H[Note] ≤ M ≤ FFFFH (M ≠ N)

Note When 0000H is set, a match interrupt immediately after the timer operation does not occur and timer output is not changed, and the first match timing is as follows. A match interrupt occurs at the timing when the timer counter (TM00 register) is changed from 0000H to 0001H.

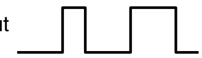
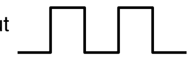
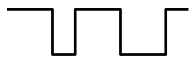
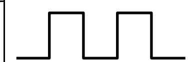
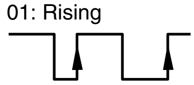
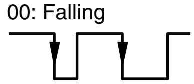
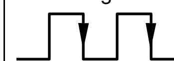
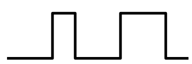
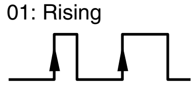
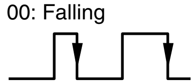
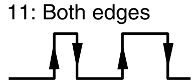
- When the timer counter is cleared due to overflow
- When the timer counter is cleared due to TI000 pin valid edge (when clear & start mode is entered by TI000 pin valid edge input)
- When the timer counter is cleared due to compare match (when clear & start mode is entered by match between TM00 and CR000 (CR000 = other than 0000H, CR010 = 0000H))



Remarks 1. N: CR000 register set value, M: CR010 register set value

2. For details of TMC003 and TMC002, see **6.3 (1) 16-bit timer mode control register 00 (TMC00)**.

Table 6-2. Capture Operation of CR000 and CR010

External Input Signal / Capture Operation	TI000 Pin Input 		TI010 Pin Input 	
Capture operation of CR000	CRC001 = 1 TI000 pin input (reverse phase) 	Set values of ES001 and ES000 Position of edge to be captured	CRC001 bit = 0 TI010 pin input 	Set values of ES101 and ES100 Position of edge to be captured
		01: Rising 		01: Rising 
		00: Falling 		00: Falling 
		11: Both edges (cannot be captured)		11: Both edges 
	Interrupt signal	INTTM000 signal is not generated even if value is captured.	Interrupt signal	INTTM000 signal is generated each time value is captured.
Capture operation of CR010	TI000 pin input[Note] 	Set values of ES001 and ES000 Position of edge to be captured		
		01: Rising 		
		00: Falling 		
		11: Both edges 		
	Interrupt signal	INTTM010 signal is generated each time value is captured.		

Note The capture operation of CR010 is not affected by the setting of the CRC001 bit.

Caution To capture the count value of the TM00 register to the CR000 register by using the phase reverse to that input to the TI000 pin, the interrupt request signal (INTTM000) is not generated after the value has been captured. If the valid edge is detected on the TI010 pin during this operation, the capture operation is not performed but the INTTM000 signal is generated as an external interrupt signal. To not use the external interrupt, mask the INTTM000 signal.

Remark CRC001: See **6.3 (2) Capture/compare control register 00 (CRC00)**.
ES101, ES100, ES001, ES000: See **6.3 (4) Prescaler mode register 00 (PRM00)**.

6.3 Registers Controlling 16-Bit Timer/Event Counter 00

Registers used to control 16-bit timer/event counter 00 are shown below.

- 16-bit timer mode control register 00 (TMC00)
- Capture/compare control register 00 (CRC00)
- 16-bit timer output control register 00 (TOC00)
- Prescaler mode register 00 (PRM00)
- Input switch control register (ISC)
- Port mode register 3 (PM3)
- Port register 3 (P3)

(1) 16-bit timer mode control register 00 (TMC00)

TMC00 is an 8-bit register that sets the 16-bit timer/event counter 00 operation mode, TM00 clear mode, and output timing, and detects an overflow.

Rewriting TMC00 is prohibited during operation (when TMC003 and TMC002 = other than 00). However, it can be changed when TMC003 and TMC002 are cleared to 00 (stopping operation) and when OVF00 is cleared to 0.

TMC00 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets TMC00 to 00H.

Caution 16-bit timer/event counter 00 starts operation at the moment TMC002 and TMC003 are set to values other than 00 (operation stop mode), respectively. Set TMC002 and TMC003 to 00 to stop the operation.

Figure 6-5. Format of 16-Bit Timer Mode Control Register 00 (TMC00)

Address: FFBAH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
TMC00	0	0	0	0	TMC003	TMC002	TMC001	OVF00

TMC003	TMC002	Operation enable of 16-bit timer/event counter 00
0	0	Disables 16-bit timer/event counter 00 operation. Stops supplying operating clock. Clears 16-bit timer counter 00 (TM00).
0	1	Free-running timer mode
1	0	Clear & start mode entered by TI000 pin valid edge input[Note]
1	1	Clear & start mode entered upon a match between TM00 and CR000

TMC001	Condition to reverse timer output (TO00)
0	• Match between TM00 and CR000 or match between TM00 and CR010
1	• Match between TM00 and CR000 or match between TM00 and CR010 • Trigger input of TI000 pin valid edge

OVF00	TM00 overflow flag
Clear (0)	Clears OVF00 to 0 or TMC003 and TMC002 = 00
Set (1)	Overflow occurs.
OVF00 is set to 1 when the value of TM00 changes from FFFFH to 0000H in all the operation modes (free-running timer mode, clear & start mode entered by TI000 pin valid edge input, and clear & start mode entered upon a match between TM00 and CR000). It can also be set to 1 by writing 1 to OVF00.	

Note The TI000 pin valid edge is set by bits 5 and 4 (ES001, ES000) of prescaler mode register 00 (PRM00).

(2) Capture/compare control register 00 (CRC00)

CRC00 is the register that controls the operation of CR000 and CR010.
Changing the value of CRC00 is prohibited during operation (when TMC003 and TMC002 = other than 00).
CRC00 can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears CRC00 to 00H.

Figure 6-6. Format of Capture/Compare Control Register 00 (CRC00)

Address: FFBCH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRC00	0	0	0	0	0	CRC002	CRC001	CRC000

CRC002	CR010 operating mode selection
0	Operates as compare register
1	Operates as capture register

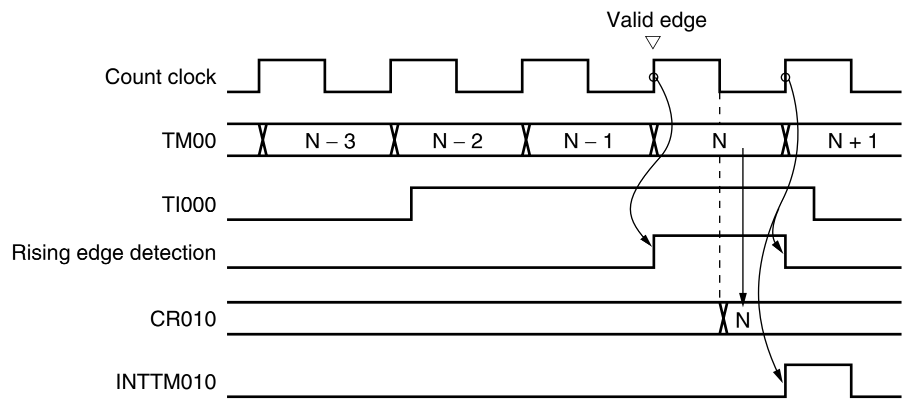
CRC001	CR000 capture trigger selection
0	Captures on valid edge of TI010 pin
1	Captures on valid edge of TI000 pin by reverse phase[Note]
The valid edge of the TI010 and TI000 pin is set by PRM00. If ES001 and ES000 are set to 11 (both edges) when CRC001 is 1, the valid edge of the TI000 pin cannot be detected.	

CRC000	CR000 operating mode selection
0	Operates as compare register
1	Operates as capture register
If TMC003 and TMC002 are set to 11 (clear & start mode entered upon a match between TM00 and CR000), be sure to set CRC000 to 0.	

Note When the valid edge is detected from the TI010 pin, the capture operation is not performed but the INTTM000 signal is generated as an external interrupt signal.

Caution To ensure that the capture operation is performed properly, the capture trigger requires a pulse two cycles longer than the count clock selected by prescaler mode register 00 (PRM00).

Figure 6-7. Example of CR010 Capture Operation (When Rising Edge Is Specified)



(3) 16-bit timer output control register 00 (TOC00)

TOC00 is an 8-bit register that controls TO00 output.

TOC00 can be rewritten while only OSPT00 is operating (when TMC003 and TMC002 = other than 00). Rewriting the other bits is prohibited during operation.

However, TOC004 can be rewritten during timer operation as a means to rewrite CR010 (see **6.5.1 Rewriting CR010 during TM00 operation**).

TOC00 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TOC00 to 00H.

Caution Be sure to set TOC00 using the following procedure.
<1> Set TOC004 and TOC001 to 1.
<2> Set only TOE00 to 1.
<3> Set either of LVS00 or LVR00 to 1.

Figure 6-8. Format of 16-Bit Timer Output Control Register 00 (TOC00)

Address: FFBDH After reset: 00H R/W

Symbol	7	<6>	<5>	4	<3>	<2>	1	<0>
TOC00	0	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00

OSPT00	One-shot pulse output trigger via software
0	–
1	One-shot pulse output
The value of this bit is always “0” when it is read. Do not set this bit to 1 in a mode other than the one-shot pulse output mode. If it is set to 1, TM00 is cleared and started.	

OSPE00	One-shot pulse output operation control
0	Successive pulse output
1	One-shot pulse output
One-shot pulse output operates correctly in the free-running timer mode or clear & start mode entered by TI000 pin valid edge input. The one-shot pulse cannot be output in the clear & start mode entered upon a match between TM00 and CR000.	

TOC004	TO00 output control on match between CR010 and TM00
0	Disables inversion operation
1	Enables inversion operation
The interrupt signal (INTTM010) is generated even when TOC004 = 0.	

LVS00	LVR00	Setting of TO00 pin output status
0	0	No change
0	1	Initial value of TO00 output is low level (TO00 output is cleared to 0).
1	0	Initial value of TO00 output is high level (TO00 output is set to 1).
1	1	Setting prohibited

- LVS00 and LVR00 can be used to set the initial value of the TO00 output level. If the initial value does not have to be set, leave LVS00 and LVR00 as 00.
- Be sure to set LVS00 and LVR00 when TOE00 = 1.
  LVS00, LVR00, and TOE00 being simultaneously set to 1 is prohibited.
- LVS00 and LVR00 are trigger bits. By setting these bits to 1, the initial value of the TO00 output level can be set. Even if these bits are cleared to 0, TO00 output is not affected.
- The values of LVS00 and LVR00 are always 0 when they are read.
- For how to set LVS00 and LVR00, see **6.5.2 Setting LVS00 and LVR00**.
- The actual TO00/TI010/P34/TI52/RTC1HZ/INTP1 pin output is determined depending on PM34 and P34, besides TO00 output.

TOC001	TO00 output control on match between CR000 and TM00
0	Disables inversion operation
1	Enables inversion operation
The interrupt signal (INTTM000) is generated even when TOC001 = 0.	

TOE00	TO00 output control
0	Disables output (TO00 output fixed to low level)
1	Enables output

(4) Prescaler mode register 00 (PRM00)

PRM00 is the register that sets the TM00 count clock and TI000 and TI010 pin input valid edges.
Rewriting PRM00 is prohibited during operation (when TMC003 and TMC002 = other than 00).
PRM00 can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets PRM00 to 00H.

Cautions 1. Do not apply the following setting when setting the PRM001 and PRM000 bits to 11 (to specify the valid edge of the TI000 pin as a count clock).

- **Clear & start mode entered by the TI000 pin valid edge**
- **Setting the TI000 pin as a capture trigger**

2. If the operation of the 16-bit timer/event counter 00 is enabled when the TI000 or TI010 pin is at high level and when the valid edge of the TI000 or TI010 pin is specified to be the rising edge or both edges, the high level of the TI000 or TI010 pin is detected as a rising edge. Note this when the TI000 or TI010 pin is pulled up. However, the rising edge is not detected when the timer operation has been once stopped and then is enabled again.

3. The valid edge of TI010 and timer output (TO00) cannot be used for the P34 pin at the same time. Select either of the functions.

Figure 6-9. Format of Prescaler Mode Register 00 (PRM00)

Address: FFBBH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM00	ES101	ES100	ES001	ES000	0	PRM002	PRM001	PRM000

ES101	ES100	TI010 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES001	ES000	TI000 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

PRM002	PRM001	PRM000	Count clock selection[Note 1]			
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz
0	0	0	f_{PRS}[Note 2]	2 MHz	5 MHz	10 MHz
0	0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz
0	1	0	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz
0	1	1	$f_{PRS}/2^4$	1.25 MHz	2.5 MHz	625 kHz
1	0	0	$f_{PRS}/2^8$	7.81 kHz	19.53 kHz	39.06 kHz
1	0	1	f_{SUB}	32.768 kHz		
1	1	0	TI000 valid edge[Note 3]			
1	1	1	TM52 output			

Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{XH}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 2.7 to 5.5 V: $f_{PRS} \leq 10$ MHz
- V_{DD} = 1.8 to 2.7 V: $f_{PRS} \leq 5$ MHz

2. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{RH}) (XSEL = 0), when 1.8 V ≤ V_{DD} < 2.7 V, the setting of PRM002 = PRM001 = PRM000 = 0 (count clock: f_{PRS}) is prohibited.

3. The external clock from the TI000 pin requires a pulse longer than twice the cycle of the peripheral hardware clock (f_{PRS}).

Caution Do not select the valid edge of TI000 as the count clock during the pulse width measurement.

Remarks 1. 8-bit timer/event counter 52 (TM52) output can be selected as the TM00 count clock by setting PRM002, PRM001, PRM000 = 1, 1, 1. Any frequency can be set as the 16-bit timer (TM00) count clock, depending on the TM52 count clock and compare register setting values.

2. f_{PRS}: Peripheral hardware clock frequency

f_{SUB}: Subsystem clock frequency

(5) Input switch control register (ISC)

The input source to TI000 becomes the input signal from the P33/TI000 pin, by setting ISC1 to 0.
ISC can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets ISC to 00H.

Figure 6-10. Format of Input Switch Control Register (ISC)

Address: FF4FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	ICS5	ICS4	ICS3	ICS2	ICS1	ICS0

ICS5	ICS4	TxD6, RxD6 input source selection
0	0	TxD6:P112, RxD6: P113
0	1	TxD6:P16, RxD6: P15
Other than above		Setting prohibited

ISC3	RxD6/P113 input enabled/disabled
0	RxD6/P113 input disabled
1	RxD6/P113 input enabled

ISC2	TI52 input source control
0	No enable control of TI52 input (P34)
1	Enable controlled of TI52 input (P34)Note 1

ISC1	TI000 input source selection
0	TI000 (P33)
1	RxD6 (P15 or P113Note 2)

ISC0	INTP0 input source selection
0	INTP0 (P120)
1	RxD6 (P15 or P113Note 2)

Notes 1. TI52 input is controlled by TOH2 output signal.

<R> **2.** This is selected by ISC5 and ISC4.

(6) Port mode register 3 (PM3)

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

When using the P34/TI52/TI010/TO00/RTC1HZ/INTP1 pin for timer output, set PM34 and the output latches of P34 to 0.

When using the P33/TI000/RTCDIV/RTCCL/BUZ/INTP2 and P34/TI52/TI010/TO00/RTC1HZ/INTP1 pins for timer input, set PM33 and PM34 to 1. At this time, the output latches of P33 and P34 may be 0 or 1.

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

Reset signal generation sets PM3 to FFH.

Figure 6-11. Format of Port Mode Register 3 (PM3)

Address: FF23H After reset: FFH R/W

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

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

6.4 Operation of 16-Bit Timer/Event Counter 00

6.4.1 Interval timer operation

If bits 3 and 2 (TMC003 and TMC002) of the 16-bit timer mode control register (TMC00) are set to 11 (clear & start mode entered upon a match between TM00 and CR000), the count operation is started in synchronization with the count clock.

When the value of TM00 later matches the value of CR000, TM00 is cleared to 0000H and a match interrupt signal (INTTM000) is generated. This INTTM000 signal enables TM00 to operate as an interval timer.

Remarks 1. For the setting of I/O pins, see **6.3 (6) Port mode register 3 (PM3)**.

2. For how to enable the INTTM000 interrupt, see **CHAPTER 21 INTERRUPT FUNCTIONS**.

Figure 6-12. Block Diagram of Interval Timer Operation

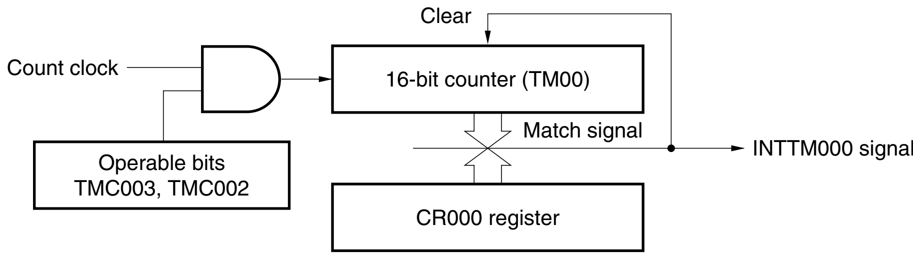


Figure 6-13. Basic Timing Example of Interval Timer Operation

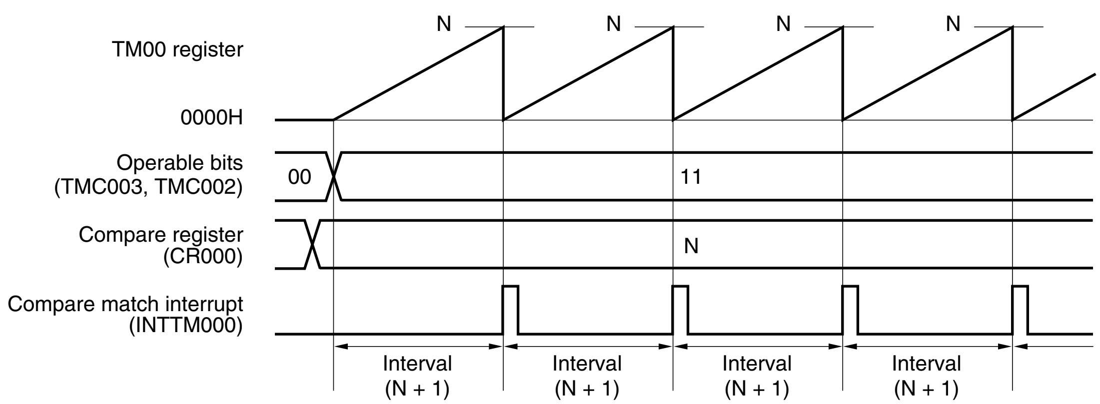


Figure 6-14. Example of Register Settings for Interval Timer Operation

(a) 16-bit timer mode control register 00 (TMC00)

				TMC003	TMC002	TMC001	OVF00
0	0	0	0	1	1	0	0

TMC003, TMC002 (1, 1): Clears and starts on match between TM00 and CR000.

(b) Capture/compare control register 00 (CRC00)

					CRC002	CRC001	CRC000
0	0	0	0	0	0	0	0

CRC000 (0): CR000 used as compare register

(c) 16-bit timer output control register 00 (TOC00)

	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
0	0	0	0	0	0	0	0

(d) Prescaler mode register 00 (PRM00)

ES101	ES100	ES001	ES000	3	PRM002	PRM001	PRM000
0	0	0	0	0	0/1	0/1	0/1

PRM002, PRM001, PRM000: Selects count clock

(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

If M is set to CR000, the interval time is as follows.

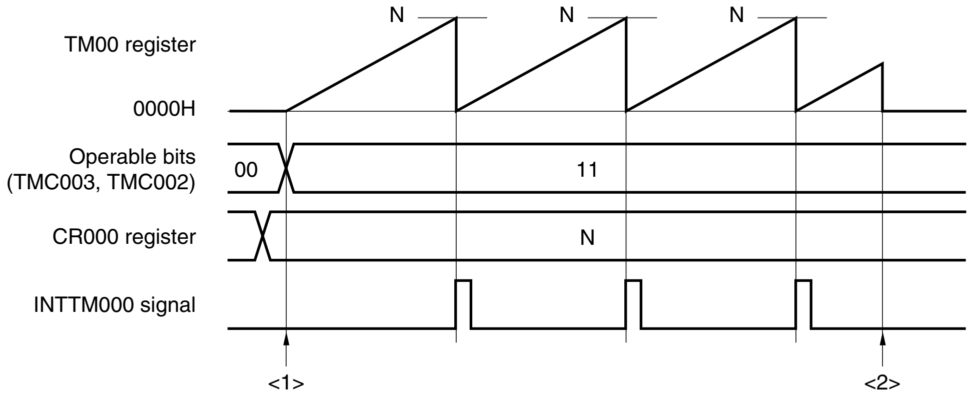
- Interval time = $(M + 1) \times$ Count clock cycle

Setting CR000 to 0000H is prohibited.

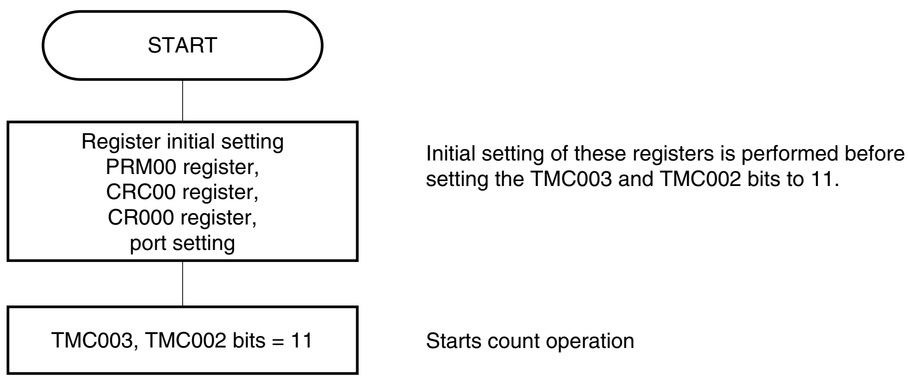
(g) 16-bit capture/compare register 010 (CR010)

Usually, CR010 is not used for the interval timer function. However, a compare match interrupt (INTTM010) is generated when the set value of CR010 matches the value of TM00.
Therefore, mask the interrupt request by using the interrupt mask flag (TMMK010).

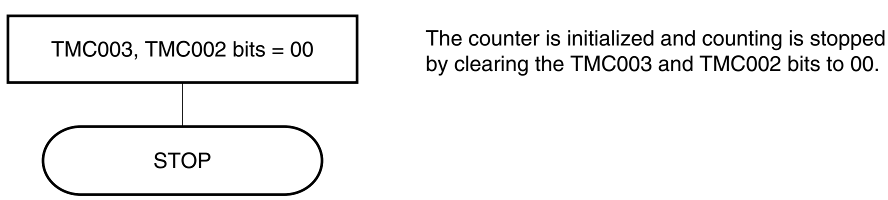
Figure 6-15. Example of Software Processing for Interval Timer Function



<1> Count operation start flow



<2> Count operation stop flow



6.4.2 Square wave output operation

When 16-bit timer/event counter 00 operates as an interval timer (see **6.4.1**), a square wave can be output from the TO00 pin by setting the 16-bit timer output control register 00 (TOC00) to 03H.

When TMC003 and TMC002 are set to 11 (count clear & start mode entered upon a match between TM00 and CR000), the counting operation is started in synchronization with the count clock.

When the value of TM00 later matches the value of CR000, TM00 is cleared to 0000H, an interrupt signal (INTTM000) is generated, and TO00 output is inverted. This TO00 output that is inverted at fixed intervals enables TO00 to output a square wave.

Remarks 1. For the setting of I/O pins, see **6.3 (6) Port mode register 3 (PM3)**.

2. For how to enable the INTTM000 signal interrupt, see **CHAPTER 21 INTERRUPT FUNCTIONS**.

Figure 6-16. Block Diagram of Square Wave Output Operation

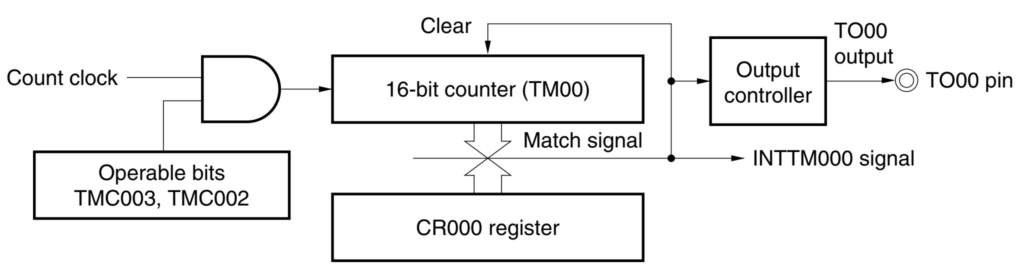


Figure 6-17. Basic Timing Example of Square Wave Output Operation

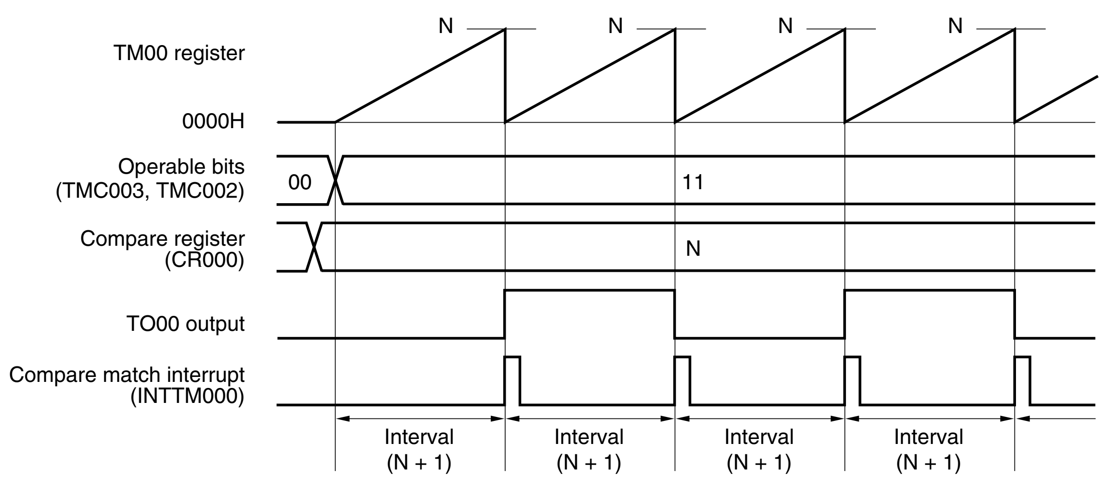


Figure 6-18. Example of Register Settings for Square Wave Output Operation

(a) 16-bit timer mode control register 00 (TMC00)

				TMC003	TMC002	TMC001	OVF00
0	0	0	0	1	1	0	0

TMC003, TMC002: Clears and starts on match between TM00 and CR000.

(b) Capture/compare control register 00 (CRC00)

					CRC002	CRC001	CRC000
0	0	0	0	0	0	0	0

CRC000: CR000 used as compare register

(c) 16-bit timer output control register 00 (TOC00)

	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
0	0	0	0	0/1	0/1	1	1

TOE00: Enables TO00 output.

TOC001: Inverts TO00 output on match between TM00 and CR000.

LVS00, LVR00: Specifies initial value of TO00 output F/F

(d) Prescaler mode register 00 (PRM00)

ES101	ES100	ES001	ES000	3	PRM002	PRM001	PRM000
0	0	0	0	0	0/1	0/1	0/1

PRM002, PRM001, PRM000: Selects count clock

(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

If M is set to CR000, the interval time is as follows.

- Square wave frequency = 1 / [2 × (M + 1) × Count clock cycle]

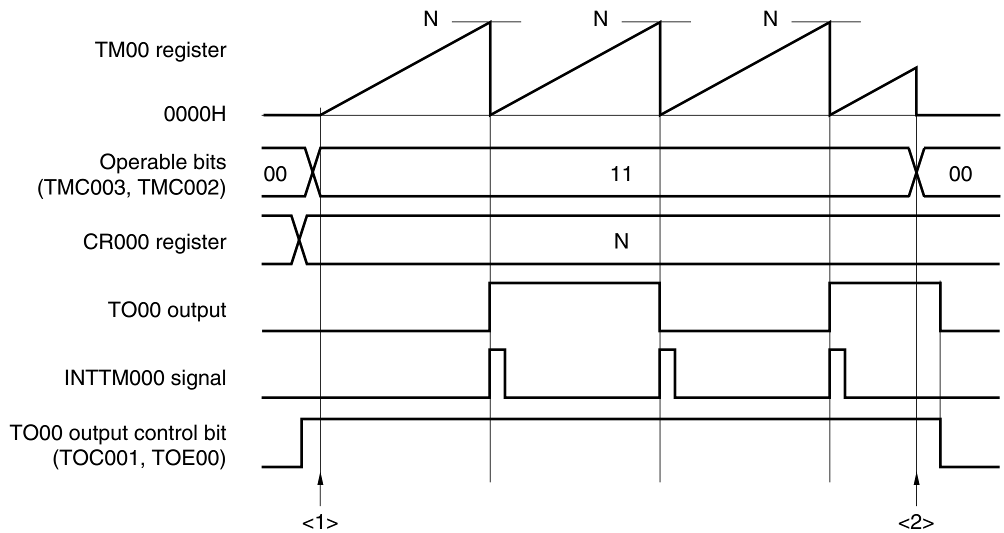
Setting CR000 to 0000H is prohibited.

(g) 16-bit capture/compare register 010 (CR010)

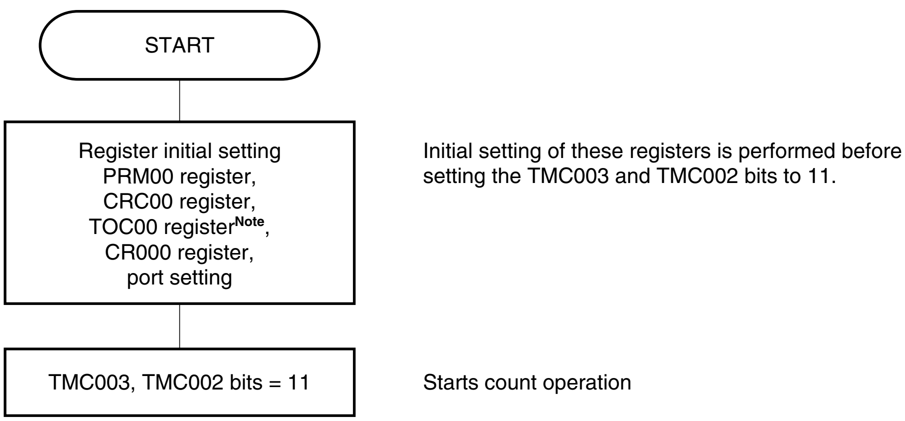
Usually, CR010 is not used for the square wave output function. However, a compare match interrupt (INTTM010) is generated when the set value of CR010 matches the value of TM00.

Therefore, mask the interrupt request by using the interrupt mask flag (TMMK010).

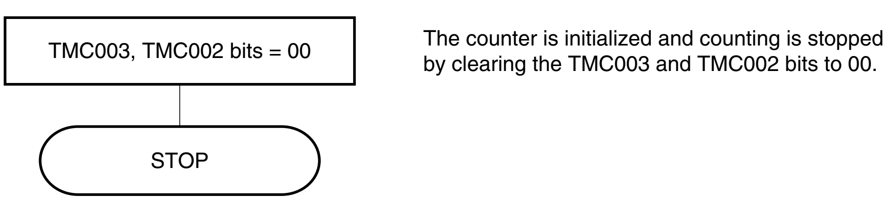
Figure 6-19. Example of Software Processing for Square Wave Output Function



<1> Count operation start flow



<2> Count operation stop flow



Note Care must be exercised when setting TOC00. For details, see **6.3 (3) 16-bit timer output control register 00 (TOC00)**.

6.4.3 External event counter operation

When bits 1 and 0 (PRM001 and PRM000) of the prescaler mode register 00 (PRM00) are set to 11 (for counting up with the valid edge of the TI000 pin) and bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) are set to 11, the valid edge of an external event input is counted, and a match interrupt signal indicating matching between TM00 and CR000 (INTTM000) is generated.

To input the external event, the TI000 pin is used. Therefore, the timer/event counter cannot be used as an external event counter in the clear & start mode entered by the TI000 pin valid edge input (when TMC003 and TMC002 = 10).

The INTTM000 signal is generated with the following timing.

- Timing of generation of INTTM000 signal (second time or later)
  = Number of times of detection of valid edge of external event × (Set value of CR000 + 1)

However, the first match interrupt immediately after the timer/event counter has started operating is generated with the following timing.

- Timing of generation of INTTM000 signal (first time only)
  = Number of times of detection of valid edge of external event input × (Set value of CR000 + 2)

To detect the valid edge, the signal input to the TI000 pin is sampled during the clock cycle of f_{PRS}. The valid edge is not detected until it is detected two times in a row. Therefore, a noise with a short pulse width can be eliminated.

Remarks 1. For the setting of I/O pins, see **6.3 (6) Port mode register 3 (PM3)**.

2. For how to enable the INTTM000 signal interrupt, see **CHAPTER 21 INTERRUPT FUNCTIONS**.

Figure 6-20. Block Diagram of External Event Counter Operation

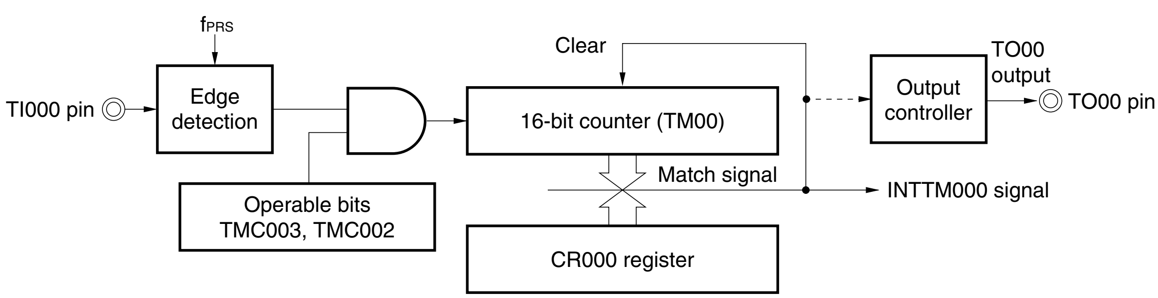


Figure 6-21. Example of Register Settings in External Event Counter Mode (1/2)

(a) 16-bit timer mode control register 00 (TMC00)

				TMC003	TMC002	TMC001	OVF00
0	0	0	0	1	1	0	0

TMC003, TMC002: Clears and starts on match between TM00 and CR000.

(b) Capture/compare control register 00 (CRC00)

					CRC002	CRC001	CRC000
0	0	0	0	0	0	0	0

CRC000: CR000 used as compare register

(c) 16-bit timer output control register 00 (TOC00)

	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
0	0	0	0/1	0/1	0/1	0/1	0/1

TOE00:
0: Disables TO00 output
1: Enables TO00 output

LVS00, LVR00: Specifies initial value of TO00 output F/F

TOC004, TOC001:
00: Does not invert TO00 output on match between TM00 and CR000/CR010.
01: Inverts TO00 output on match between TM00 and CR000.
10: Inverts TO00 output on match between TM00 and CR010.
11: Inverts TO00 output on match between TM00 and CR000/CR010.

(d) Prescaler mode register 00 (PRM00)

ES101	ES100	ES001	ES000	3	PRM002	PRM001	PRM000
0	0	0/1	0/1	0	1	1	0

PRM002, PRM001, PRM000: Selects count clock (specifies valid edge of TI000).

ES001, ES000:
00: Falling edge detection
01: Rising edge detection
10: Setting prohibited
11: Both edges detection

Figure 6-21. Example of Register Settings in External Event Counter Mode (2/2)

(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

If M is set to CR000, the interrupt signal (INTTM000) is generated when the number of external events reaches (M + 1).

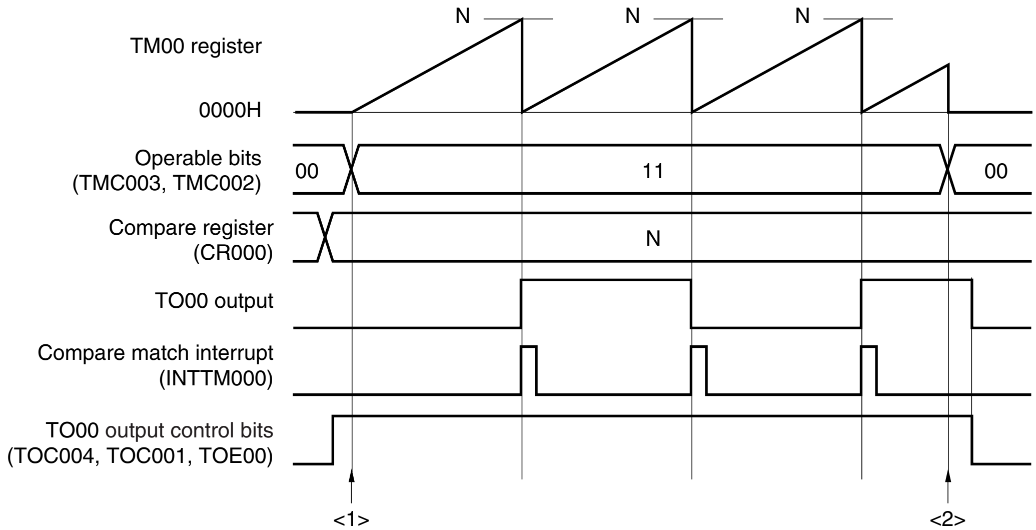
Setting CR000 to 0000H is prohibited.

(g) 16-bit capture/compare register 010 (CR010)

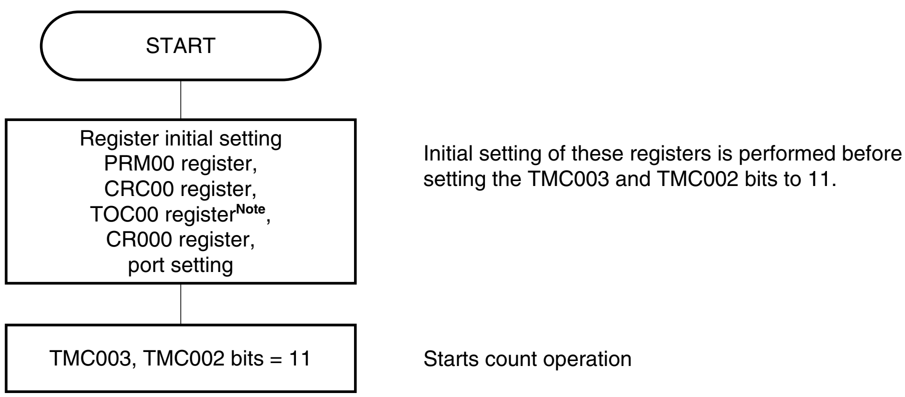
Usually, CR010 is not used in the external event counter mode. However, a compare match interrupt (INTTM010) is generated when the set value of CR010 matches the value of TM00.

Therefore, mask the interrupt request by using the interrupt mask flag (TMMK010).

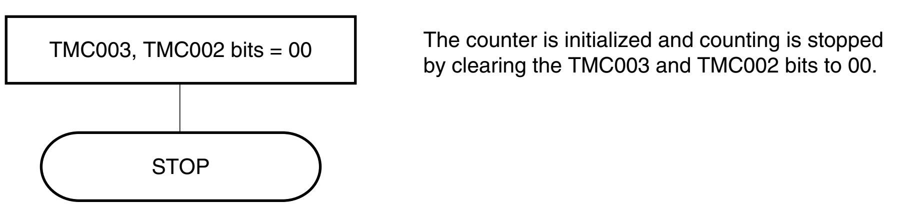
Figure 6-22. Example of Software Processing in External Event Counter Mode



<1> Count operation start flow



<2> Count operation stop flow



Note Care must be exercised when setting TOC00. For details, see **6.3 (3) 16-bit timer output control register 00 (TOC00)**.

6.4.4 Operation in clear & start mode entered by TI000 pin valid edge input

When bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) are set to 10 (clear & start mode entered by the TI000 pin valid edge input) and the count clock (set by PRM00) is supplied to the timer/event counter, TM00 starts counting up. When the valid edge of the TI000 pin is detected during the counting operation, TM00 is cleared to 0000H and starts counting up again. If the valid edge of the TI000 pin is not detected, TM00 overflows and continues counting.

The valid edge of the TI000 pin is a cause to clear TM00. Starting the counter is not controlled immediately after the start of the operation.

CR000 and CR010 are used as compare registers and capture registers.

(a) When CR000 and CR010 are used as compare registers

Signals INTTM000 and INTTM010 are generated when the value of TM00 matches the value of CR000 and CR010.

(b) When CR000 and CR010 are used as capture registers

The count value of TM00 is captured to CR000 and the INTTM000 signal is generated when the valid edge is input to the TI010 pin (or when the phase reverse to that of the valid edge is input to the TI000 pin).

When the valid edge is input to the TI000 pin, the count value of TM00 is captured to CR010 and the INTTM010 signal is generated. As soon as the count value has been captured, the counter is cleared to 0000H.

Caution Do not set the count clock as the valid edge of the TI000 pin (PRM002, PRM001, and PRM000 = 110). When PRM002, PRM001, and PRM000 = 110, TM00 is cleared.

Remarks 1. For the setting of the I/O pins, see **6.3 (6) Port mode register 3 (PM3)**.

2. For how to enable the INTTM000 signal interrupt, see **CHAPTER 21 INTERRUPT FUNCTIONS**.

(1) Operation in clear & start mode entered by TI000 pin valid edge input (CR000: compare register, CR010: compare register)

Figure 6-23. Block Diagram of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Compare Register, CR010: Compare Register)

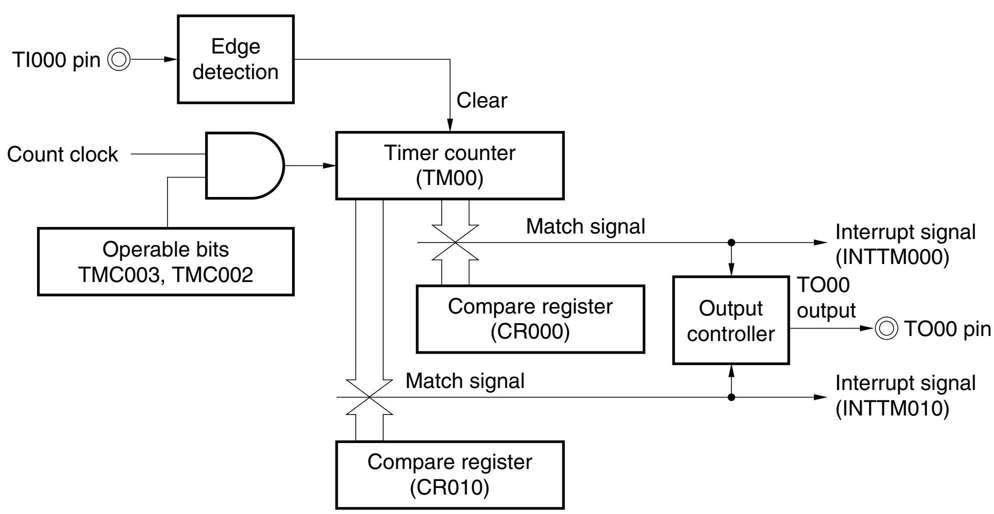
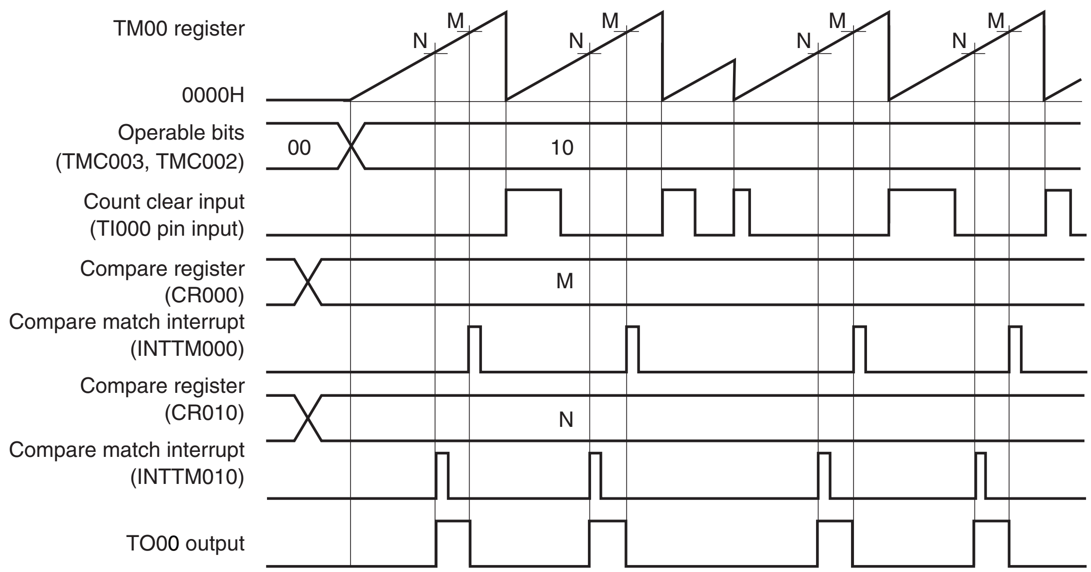
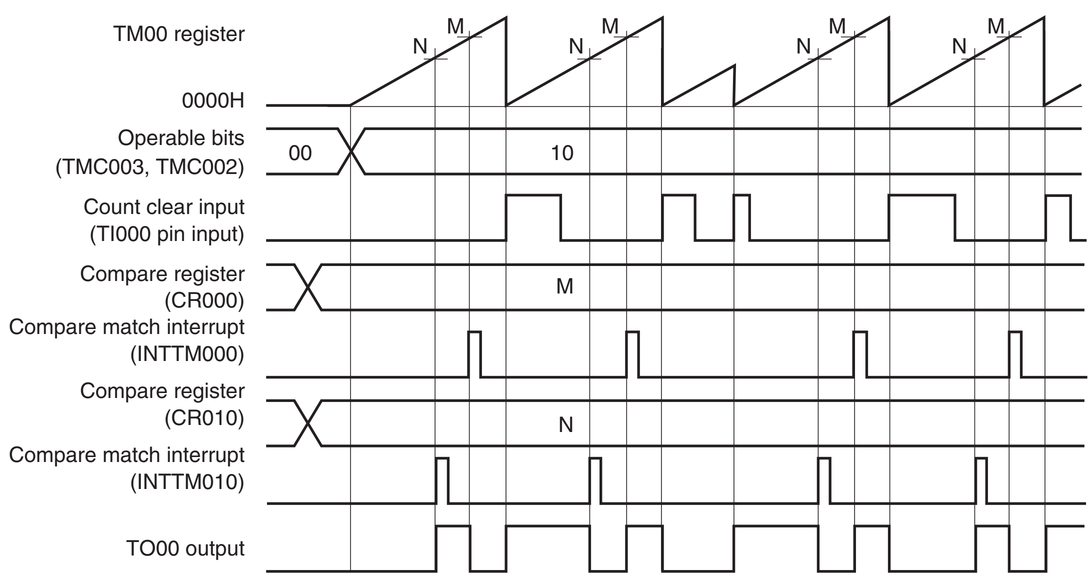


Figure 6-24. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Compare Register, CR010: Compare Register)

(a) TOC00 = 13H, PRM00 = 10H, CRC00, = 00H, TMC00 = 08H



(b) TOC00 = 13H, PRM00 = 10H, CRC00, = 00H, TMC00 = 0AH



(a) and (b) differ as follows depending on the setting of bit 1 (TMC001) of the 16-bit timer mode control register 01 (TMC00).

(a) The TO00 output level is inverted when TM00 matches a compare register.

(b) The TO00 output level is inverted when TM00 matches a compare register or when the valid edge of the TI000 pin is detected.

(2) Operation in clear & start mode entered by TI000 pin valid edge input (CR000: compare register, CR010: capture register)

Figure 6-25. Block Diagram of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Compare Register, CR010: Capture Register)

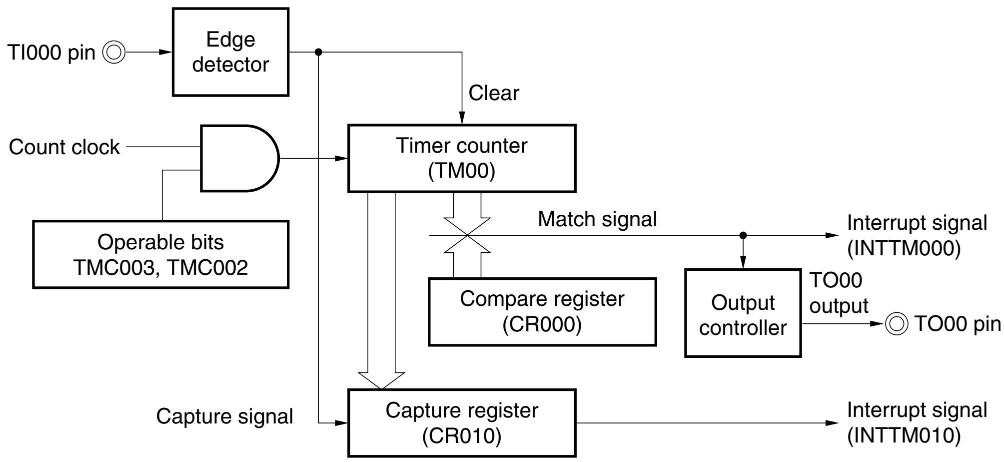
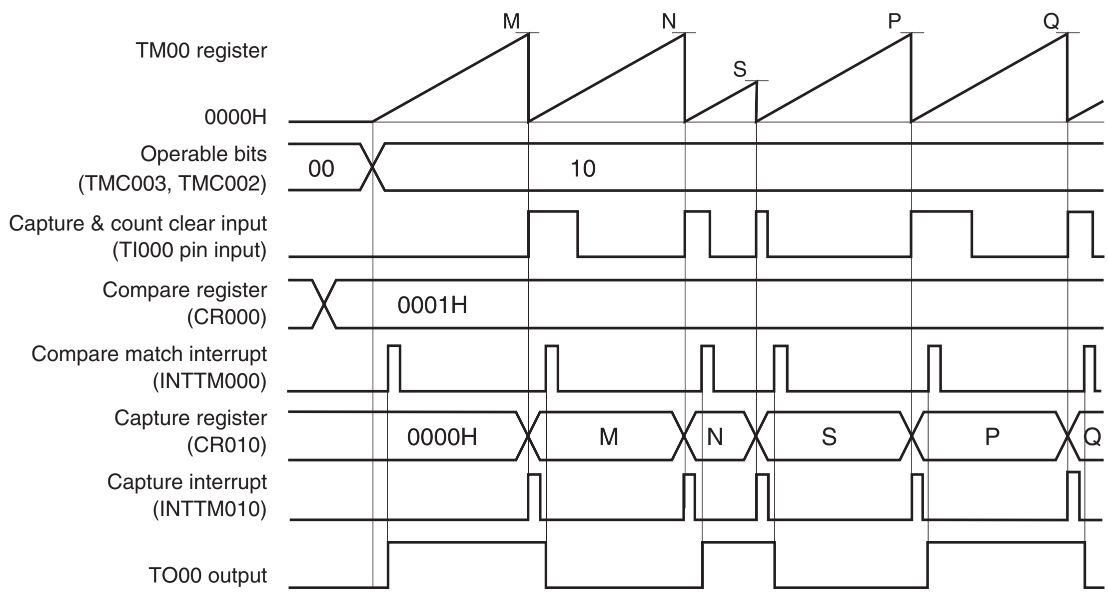


Figure 6-26. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Compare Register, CR010: Capture Register) (1/2)

(a) TOC00 = 13H, PRM00 = 10H, CRC00, = 04H, TMC00 = 08H, CR000 = 0001H

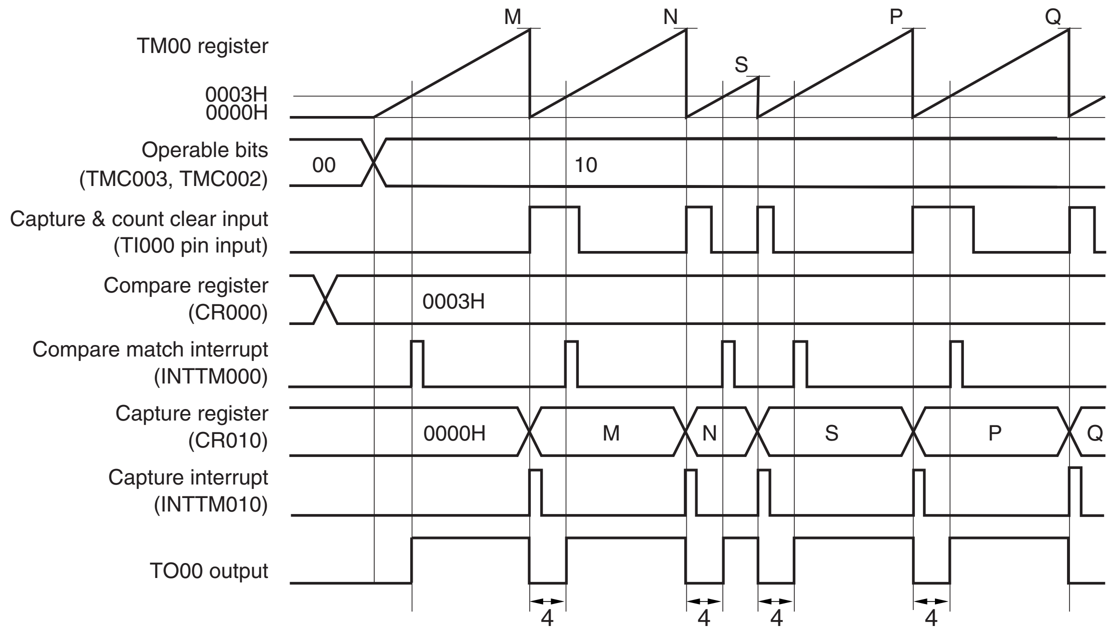


This is an application example where the TO00 output level is inverted when the count value has been captured & cleared.

The count value is captured to CR010 and TM00 is cleared (to 0000H) when the valid edge of the TI000 pin is detected. When the count value of TM00 is 0001H, a compare match interrupt signal (INTTM000) is generated, and the TO00 output level is inverted.

Figure 6-26. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Compare Register, CR010: Capture Register) (2/2)

(b) TOC00 = 13H, PRM00 = 10H, CRC00, = 04H, TMC00 = 0AH, CR000 = 0003H



This is an application example where the width set to CR000 (4 clocks in this example) is to be output from the TO00 pin when the count value has been captured & cleared.

The count value is captured to CR010, a capture interrupt signal (INTTM010) is generated, TM00 is cleared (to 0000H), and the TO00 output is inverted when the valid edge of the TI000 pin is detected. When the count value of TM00 is 0003H (four clocks have been counted), a compare match interrupt signal (INTTM000) is generated and the TO00 output level is inverted.

(3) Operation in clear & start mode by entered TI000 pin valid edge input (CR000: capture register, CR010: compare register)

Figure 6-27. Block Diagram of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Compare Register)

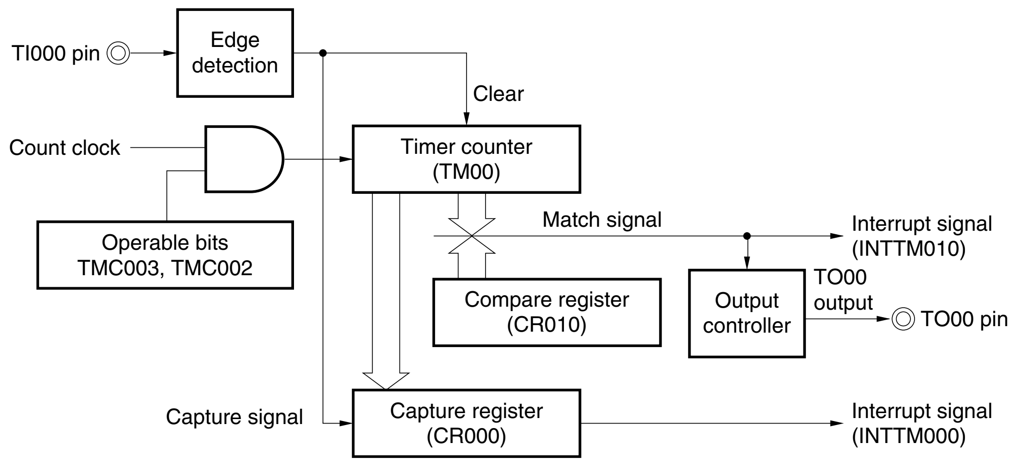
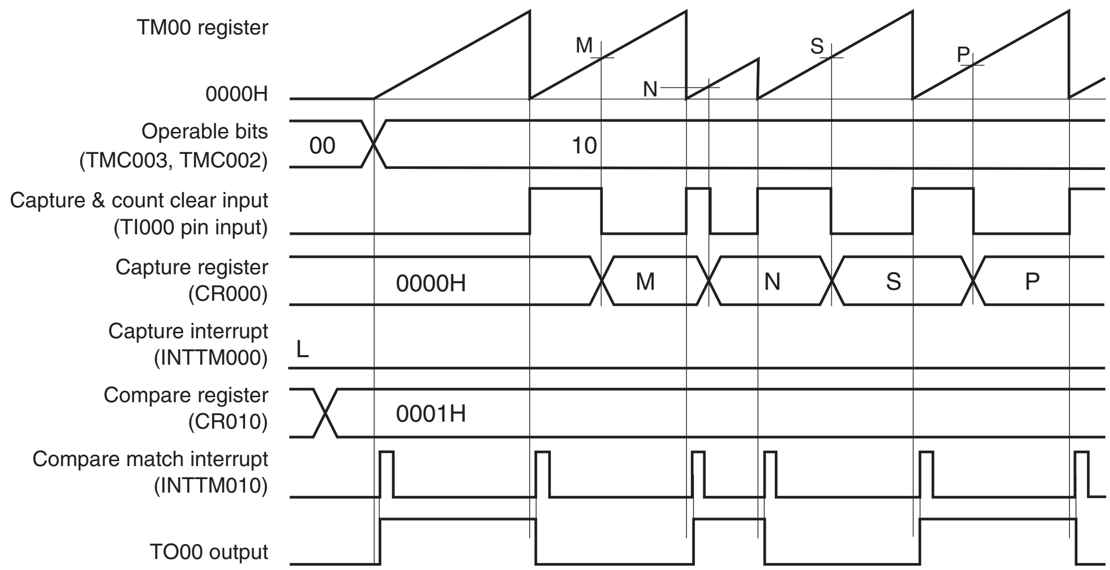


Figure 6-28. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Compare Register) (1/2)

(a) TOC00 = 13H, PRM00 = 10H, CRC00, = 03H, TMC00 = 08H, CR010 = 0001H



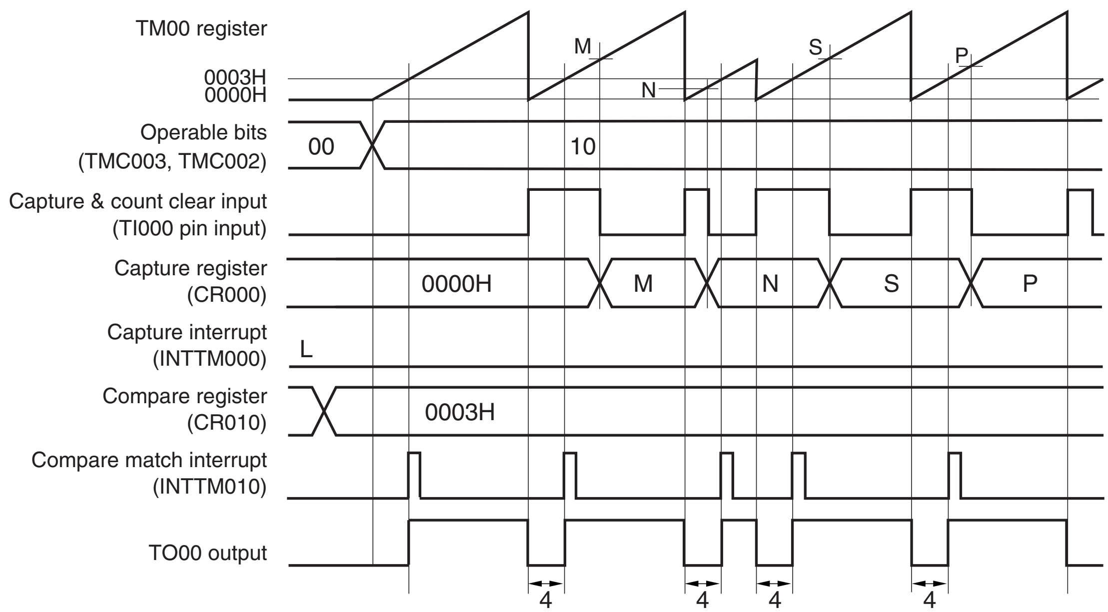
This is an application example where the TO00 output level is to be inverted when the count value has been captured & cleared.

TM00 is cleared at the rising edge detection of the TI000 pin and it is captured to CR000 at the falling edge detection of the TI000 pin.

When bit 1 (CRC001) of capture/compare control register 00 (CRC00) is set to 1, the count value of TM00 is captured to CR000 in the phase reverse to that of the signal input to the TI000 pin, but the capture interrupt signal (INTTM000) is not generated. However, the INTTM000 signal is generated when the valid edge of the TI010 pin is detected. Mask the INTTM000 signal when it is not used.

Figure 6-28. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Compare Register) (2/2)

(b) TOC00 = 13H, PRM00 = 10H, CRC00, = 03H, TMC00 = 0AH, CR010 = 0003H



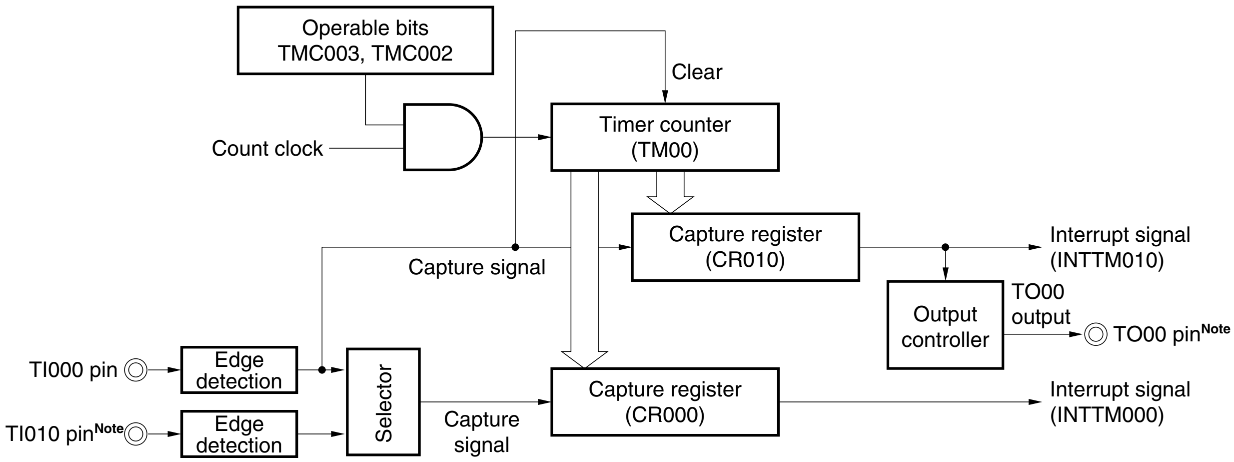
This is an application example where the width set to CR010 (4 clocks in this example) is to be output from the TO00 pin when the count value has been captured & cleared.

TM00 is cleared (to 0000H) at the rising edge detection of the TI000 pin and captured to CR000 at the falling edge detection of the TI000 pin. The TO00 output is inverted when TM00 is cleared (to 0000H) because the rising edge of the TI000 pin has been detected or when the value of TM00 matches that of a compare register (CR010).

When bit 1 (CRC001) of capture/compare control register 00 (CRC00) is 1, the count value of TM00 is captured to CR000 in the phase reverse to that of the input signal of the TI000 pin, but the capture interrupt signal (INTTM000) is not generated. However, the INTTM000 interrupt is generated when the valid edge of the TI010 pin is detected. Mask the INTTM000 signal when it is not used.

(4) Operation in clear & start mode entered by TI000 pin valid edge input (CR000: capture register, CR010: capture register)

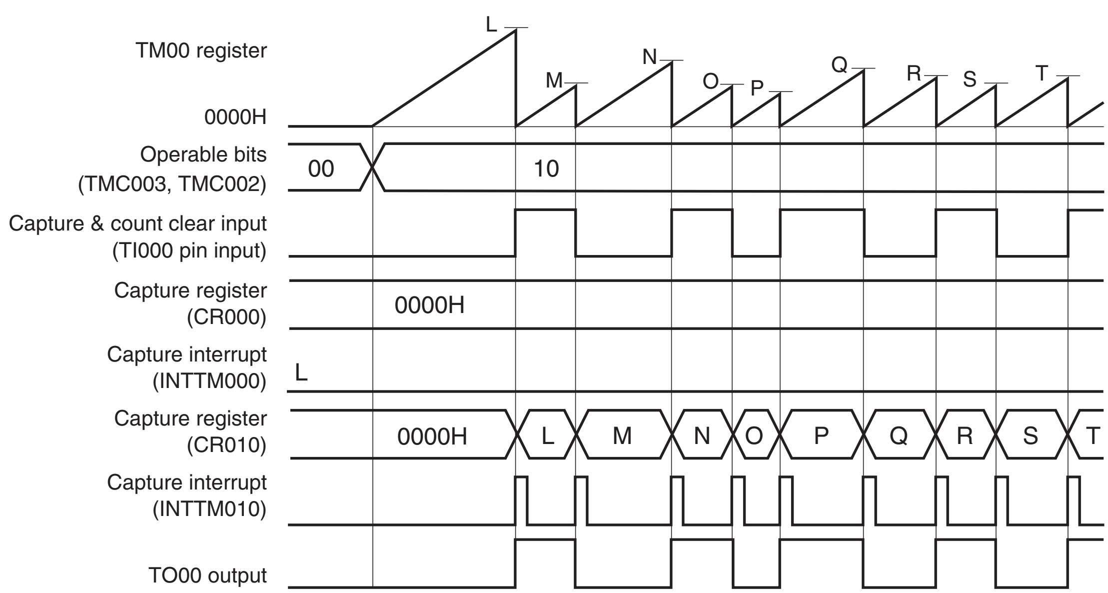
Figure 6-29. Block Diagram of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Capture Register)



Note The timer output (TO00) cannot be used when detecting the valid edge of the TI010 pin is used.

Figure 6-30. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Capture Register) (1/3)

(a) TOC00 = 13H, PRM00 = 30H, CRC00 = 05H, TMC00 = 0AH

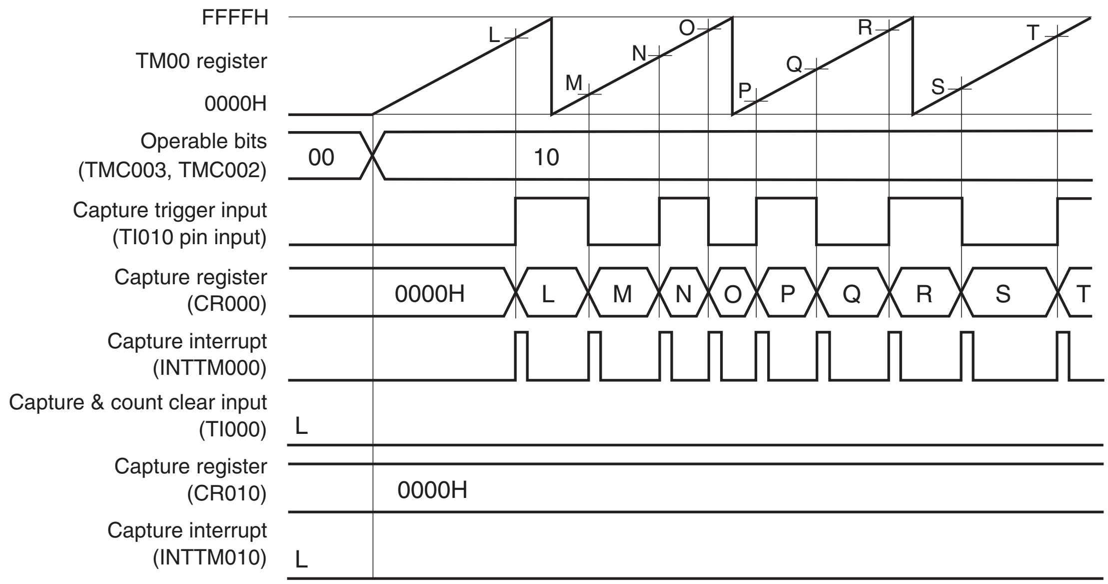


This is an application example where the count value is captured to CR010, TM00 is cleared, and the TO00 output is inverted when the rising or falling edge of the TI000 pin is detected.

When the edge of the TI010 pin is detected, an interrupt signal (INTTM000) is generated. Mask the INTTM000 signal when it is not used.

Figure 6-30. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Capture Register) (2/3)

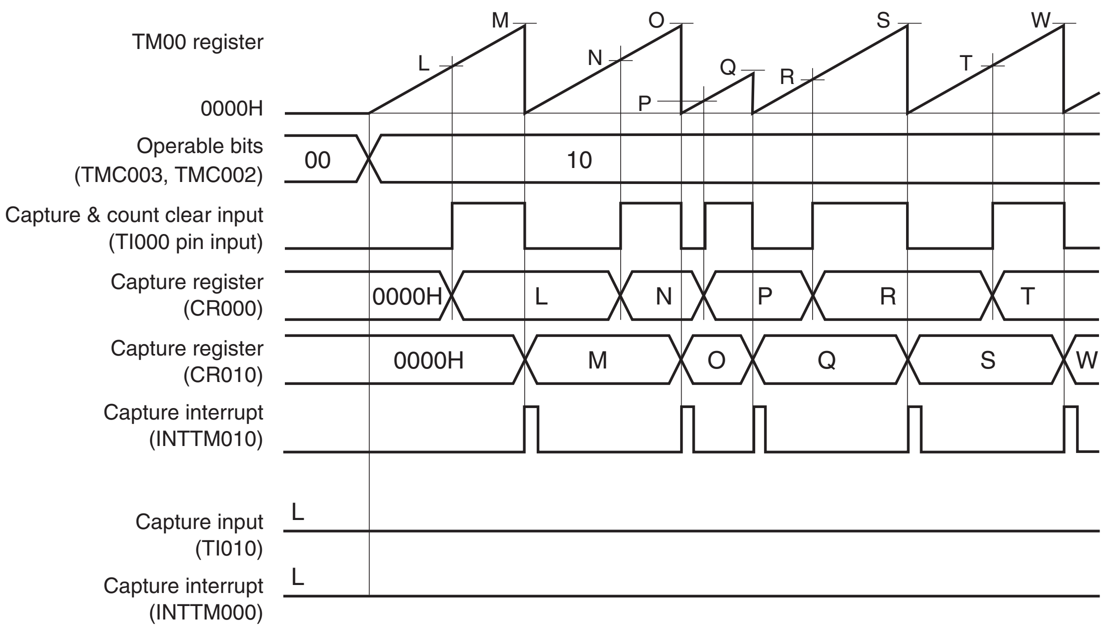
(b) TOC00 = 13H, PRM00 = C0H, CRC00 = 05H, TMC00 = 0AH



This is a timing example where an edge is not input to the TI000 pin, in an application where the count value is captured to CR000 when the rising or falling edge of the TI010 pin is detected.

Figure 6-30. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Capture Register, CR010: Capture Register) (3/3)

(c) TOC00 = 13H, PRM00 = 00H, CRC00 = 07H, TMC00 = 0AH



This is an application example where the pulse width of the signal input to the TI000 pin is measured.
By setting CRC00, the count value can be captured to CR000 in the phase reverse to the falling edge of the TI000 pin (i.e., rising edge) and to CR010 at the falling edge of the TI000 pin.
The high- and low-level widths of the input pulse can be calculated by the following expressions.

- High-level width = [CR010 value] – [CR000 value] × [Count clock cycle]
- Low-level width = [CR000 value] × [Count clock cycle]

If the reverse phase of the TI000 pin is selected as a trigger to capture the count value to CR000, the INTTM000 signal is not generated. Read the values of CR000 and CR010 to measure the pulse width immediately after the INTTM010 signal is generated.
However, if the valid edge specified by bits 6 and 5 (ES101 and ES100) of prescaler mode register 00 (PRM00) is input to the TI010 pin, the count value is not captured but the INTTM000 signal is generated. To measure the pulse width of the TI000 pin, mask the INTTM000 signal when it is not used.

Figure 6-31. Example of Register Settings in Clear & Start Mode Entered by TI000 Pin Valid Edge Input (1/2)

(a) 16-bit timer mode control register 00 (TMC00)

				TMC003	TMC002	TMC001	OVF00
0	0	0	0	1	0	0/1	0

TMC001:
- 0: Inverts TO00 output on match between CR000 and CR010.
- 1: Inverts TO00 output on match between CR000 and CR010 and valid edge of TI000 pin.

TMC003, TMC002: Clears and starts at valid edge input of TI000 pin.

(b) Capture/compare control register 00 (CRC00)

					CRC002	CRC001	CRC000
0	0	0	0	0	0/1	0/1	0/1

CRC000:
- 0: CR000 used as compare register
- 1: CR000 used as capture register

CRC001:
- 0: TI010 pin is used as capture trigger of CR000.
- 1: Reverse phase of TI000 pin is used as capture trigger of CR000.

CRC002:
- 0: CR010 used as compare register
- 1: CR010 used as capture register

(c) 16-bit timer output control register 00 (TOC00)

	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
0	0	0	0/1	0/1	0/1	0/1	0/1

TOE00:
- 0: Disables TO00 output[Note]
- 1: Enables TO00 output

LVS00, LVR00: Specifies initial value of TO00 output F/F

TOC004, TOC001:
- 00: Does not invert TO00 output on match between TM00 and CR000/CR010.
- 01: Inverts TO00 output on match between TM00 and CR000.
- 10: Inverts TO00 output on match between TM00 and CR010.
- 11: Inverts TO00 output on match between TM00 and CR000/CR010.

Note The timer output (TO00) cannot be used when detecting the valid edge of the TI010 pin is used.

Figure 6-31. Example of Register Settings in Clear & Start Mode Entered by TI000 Pin Valid Edge Input (2/2)

(d) Prescaler mode register 00 (PRM00)

ES101	ES100	ES001	ES000	3	PRM002	PRM001	PRM000
0/1	0/1	0/1	0/1	0	0/1	0/1	0/1

PRM002, PRM001, PRM000: Count clock selection
(setting TI000 valid edge is prohibited)

ES001, ES000:
00: Falling edge detection
01: Rising edge detection
10: Setting prohibited
11: Both edges detection
(setting prohibited when CRC001 = 1)

ES101, ES100:
00: Falling edge detection
01: Rising edge detection
10: Setting prohibited
11: Both edges detection

(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

When this register is used as a compare register and when its value matches the count value of TM00, an interrupt signal (INTTM000) is generated. The count value of TM00 is not cleared.

To use this register as a capture register, select either the TI000 or TI010 pin[Note] input as a capture trigger. When the valid edge of the capture trigger is detected, the count value of TM00 is stored in CR000.

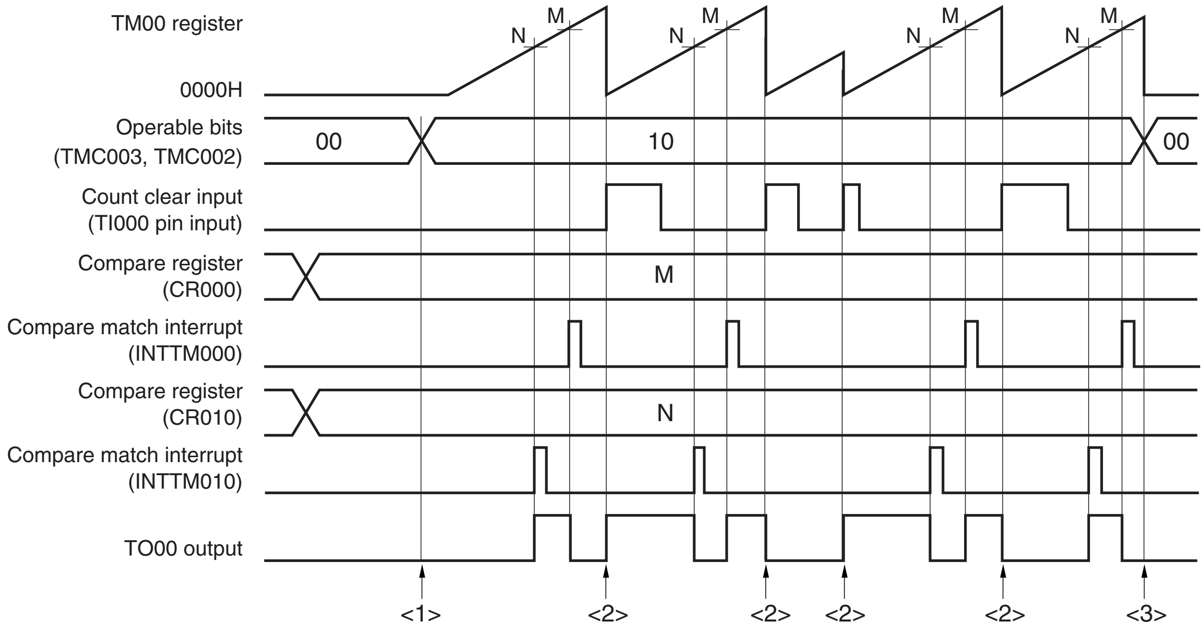
Note The timer output (TO00) cannot be used when detection of the valid edge of the TI010 pin is used.

(g) 16-bit capture/compare register 010 (CR010)

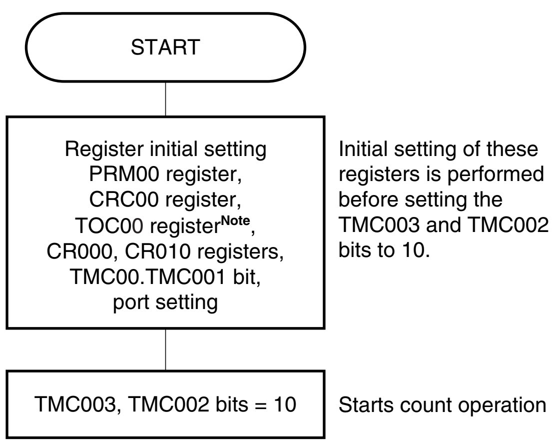
When this register is used as a compare register and when its value matches the count value of TM00, an interrupt signal (INTTM010) is generated. The count value of TM00 is not cleared.

When this register is used as a capture register, the TI000 pin input is used as a capture trigger. When the valid edge of the capture trigger is detected, the count value of TM00 is stored in CR010.

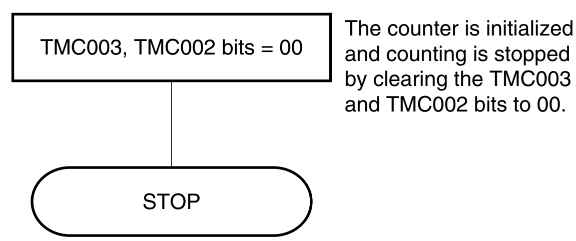
Figure 6-32. Example of Software Processing in Clear & Start Mode Entered by TI000 Pin Valid Edge Input



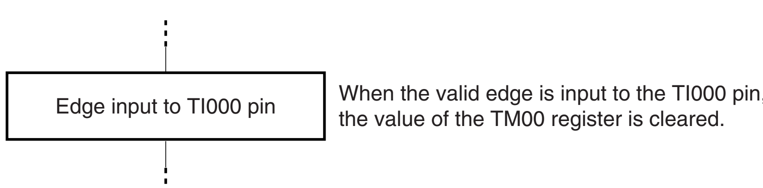
<1> Count operation start flow



<3> Count operation stop flow



<2> TM00 register clear & start flow



Note Care must be exercised when setting TOC00. For details, see **6.3 (3) 16-bit timer output control register 00 (TOC00)**.

6.4.5 Free-running timer operation

When bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) are set to 01 (free-running timer mode), 16-bit timer/event counter 00 continues counting up in synchronization with the count clock. When it has counted up to FFFFH, the overflow flag (OVF00) is set to 1 at the next clock, and TM00 is cleared (to 0000H) and continues counting. Clear OVF00 to 0 by executing the CLR instruction via software.

The following three types of free-running timer operations are available.

- Both CR000 and CR010 are used as compare registers.
- One of CR000 or CR010 is used as a compare register and the other is used as a capture register.
- Both CR000 and CR010 are used as capture registers.

Remarks 1. For the setting of the I/O pins, see **6.3 (6) Port mode register 3 (PM3)**.

2. For how to enable the INTTM000 signal interrupt, see **CHAPTER 21 INTERRUPT FUNCTIONS**.

(1) Free-running timer mode operation (CR000: compare register, CR010: compare register)

Figure 6-33. Block Diagram of Free-Running Timer Mode (CR000: Compare Register, CR010: Compare Register)

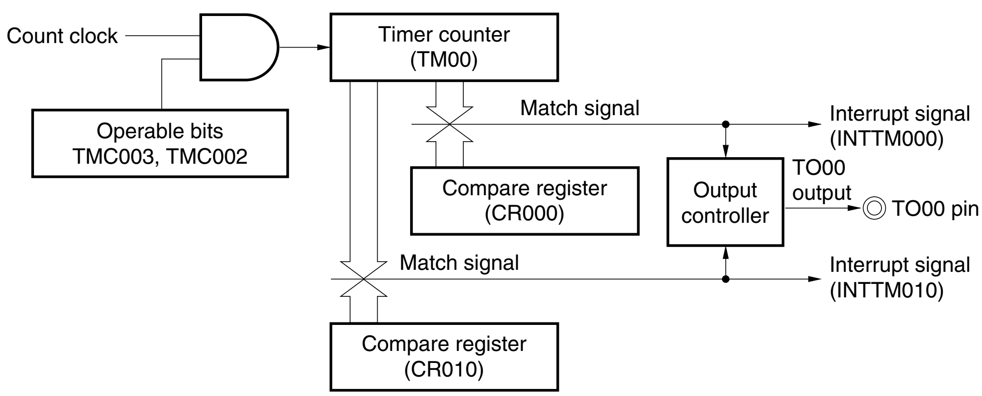
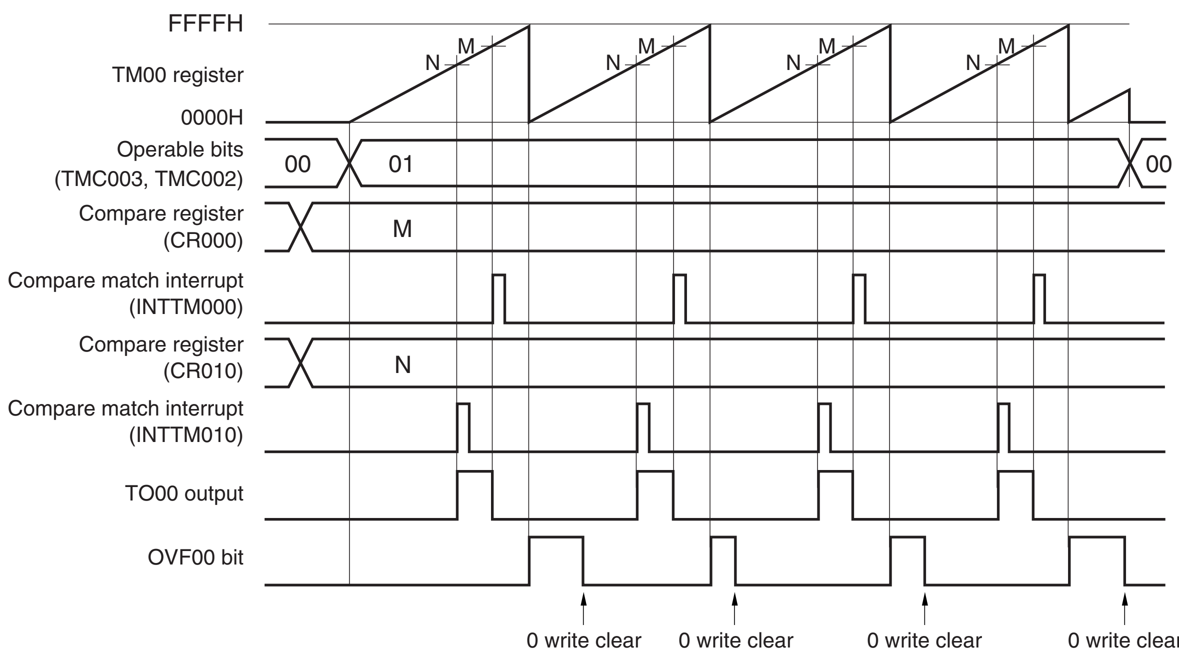


Figure 6-34. Timing Example of Free-Running Timer Mode (CR000: Compare Register, CR010: Compare Register)

- **TOC00 = 13H, PRM00 = 00H, CRC00 = 00H, TMC00 = 04H**



This is an application example where two compare registers are used in the free-running timer mode.
The TO00 output level is reversed each time the count value of TM00 matches the set value of CR000 or CR010.
When the count value matches the register value, the INTTM000 or INTTM010 signal is generated.

(2) Free-running timer mode operation (CR000: compare register, CR010: capture register)

Figure 6-35. Block Diagram of Free-Running Timer Mode (CR000: Compare Register, CR010: Capture Register)

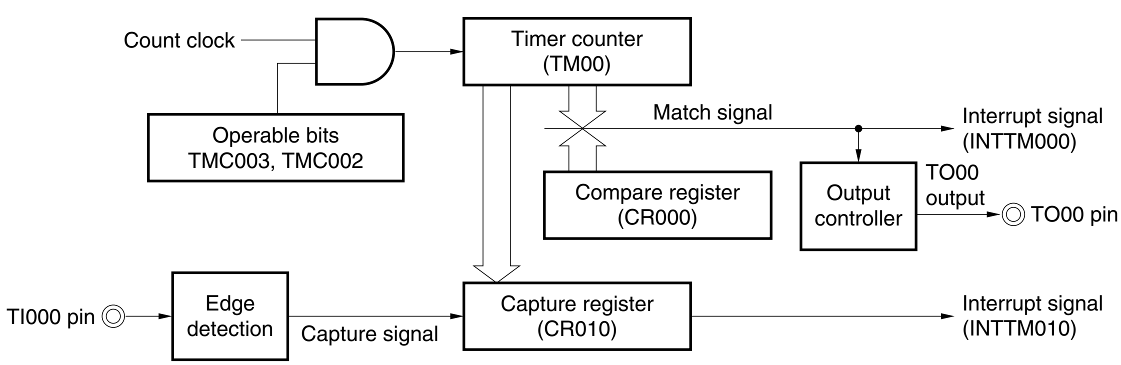
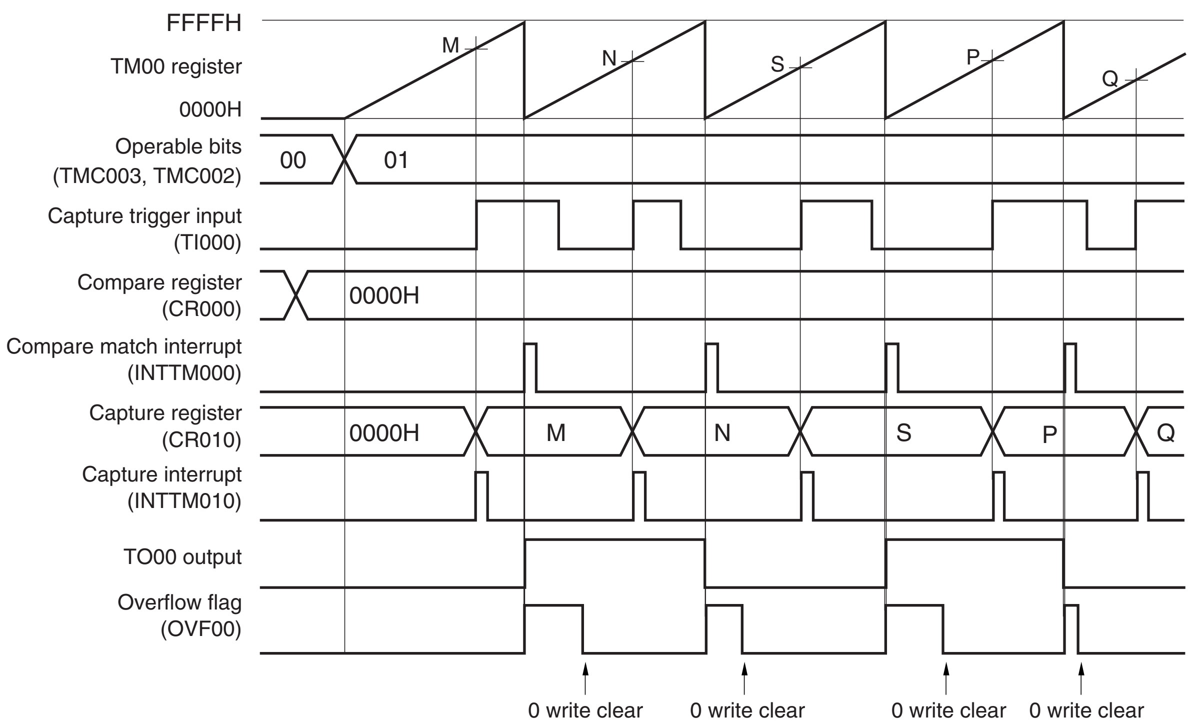


Figure 6-36. Timing Example of Free-Running Timer Mode (CR000: Compare Register, CR010: Capture Register)

• **TOC00 = 13H, PRM00 = 10H, CRC00 = 04H, TMC00 = 04H**

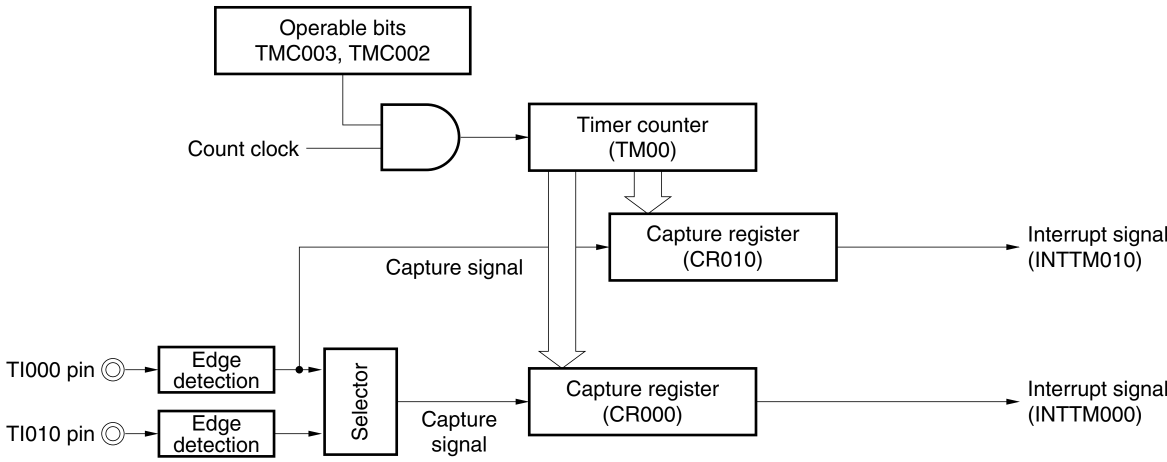


This is an application example where a compare register and a capture register are used at the same time in the free-running timer mode.

In this example, the INTTM000 signal is generated and the TO00 output is reversed each time the count value of TM00 matches the set value of CR000 (compare register). In addition, the INTTM010 signal is generated and the count value of TM00 is captured to CR010 each time the valid edge of the TI000 pin is detected.

(3) Free-running timer mode operation
(CR000: capture register, CR010: capture register)

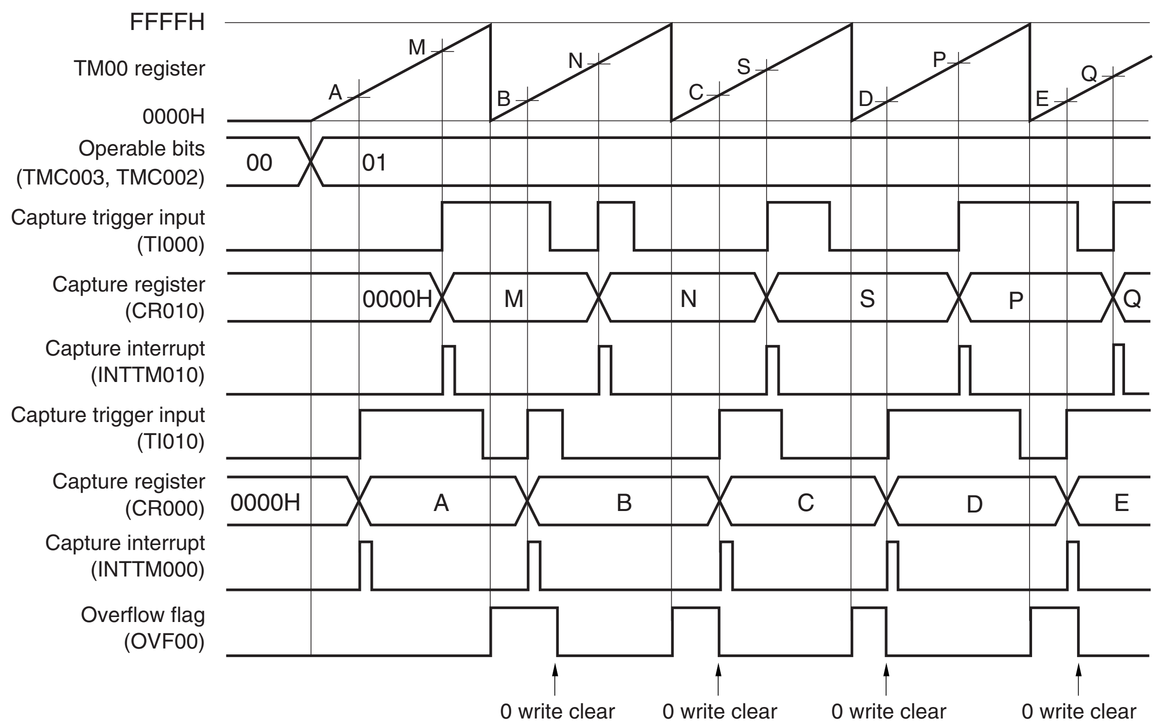
Figure 6-37. Block Diagram of Free-Running Timer Mode
(CR000: Capture Register, CR010: Capture Register)



Remark If both CR000 and CR010 are used as capture registers in the free-running timer mode, the TO00 output level is not inverted.
However, it can be inverted each time the valid edge of the TI000 pin is detected if bit 1 (TMC001) of 16-bit timer mode control register 00 (TMC00) is set to 1.

Figure 6-38. Timing Example of Free-Running Timer Mode
(CR000: Capture Register, CR010: Capture Register) (1/2)

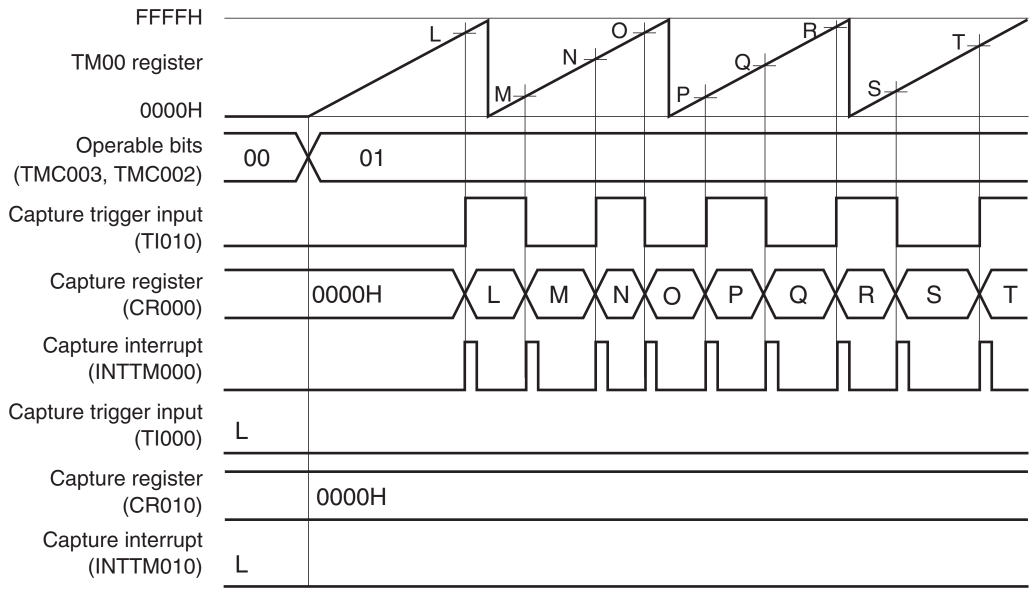
(a) TOC00 = 13H, PRM00 = 50H, CRC00 = 05H, TMC00 = 04H



This is an application example where the count values that have been captured at the valid edges of separate capture trigger signals are stored in separate capture registers in the free-running timer mode.
The count value is captured to CR010 when the valid edge of the TI000 pin input is detected and to CR000 when the valid edge of the TI010 pin input is detected.

Figure 6-38. Timing Example of Free-Running Timer Mode
(CR000: Capture Register, CR010: Capture Register) (2/2)

(b) TOC00 = 13H, PRM00 = C0H, CRC00 = 05H, TMC00 = 04H



This is an application example where both the edges of the TI010 pin are detected and the count value is captured to CR000 in the free-running timer mode.

When both CR000 and CR010 are used as capture registers and when the valid edge of only the TI010 pin is to be detected, the count value cannot be captured to CR010.

Figure 6-39. Example of Register Settings in Free-Running Timer Mode (1/2)

(a) 16-bit timer mode control register 00 (TMC00)

				TMC003	TMC002	TMC001	OVF00
0	0	0	0	0	1	0/1	0

TMC001:
0: Inverts TO00 output on match between TM00 and CR000/CR010.
1: Inverts TO00 output on match between TM00 and CR000/CR010 and valid edge of TI000 pin.

TMC003, TMC002: Free-running timer mode

(b) Capture/compare control register 00 (CRC00)

					CRC002	CRC001	CRC000
0	0	0	0	0	0/1	0/1	0/1

CRC000:
0: CR000 used as compare register
1: CR000 used as capture register

CRC001:
0: TI010 pin is used as capture trigger of CR000.
1: Reverse phase of TI000 pin is used as capture trigger of CR000.

CRC002:
0: CR010 used as compare register
1: CR010 used as capture register

(c) 16-bit timer output control register 00 (TOC00)

	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
0	0	0	0/1	0/1	0/1	0/1	0/1

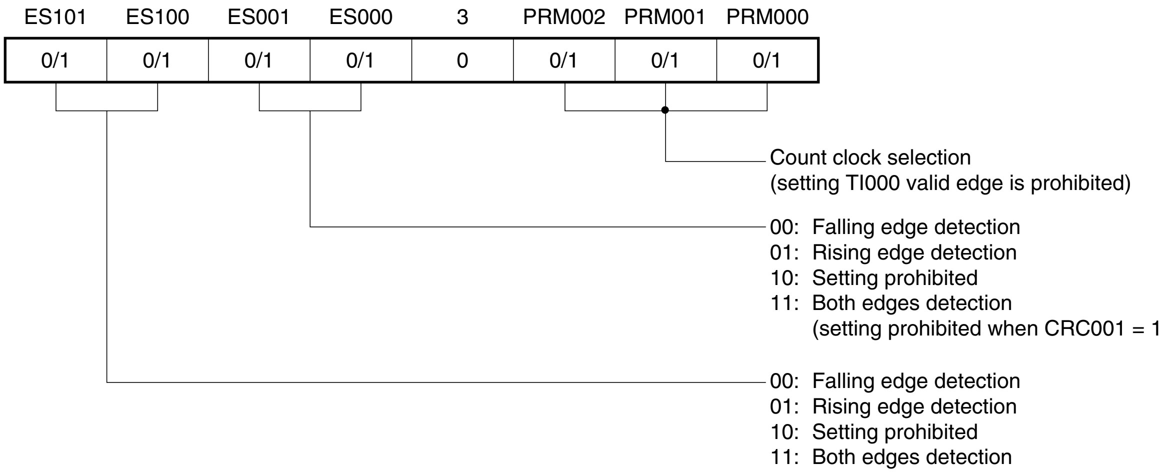
TOE00:
0: Disables TO00 output
1: Enables TO00 output

LVS00, LVR00: Specifies initial value of TO00 output F/F

TOC004, TOC001:
00: Does not invert TO00 output on match between TM00 and CR000/CR010.
01: Inverts TO00 output on match between TM00 and CR000.
10: Inverts TO00 output on match between TM00 and CR010.
11: Inverts TO00 output on match between TM00 and CR000/CR010.

Figure 6-39. Example of Register Settings in Free-Running Timer Mode (2/2)

(d) Prescaler mode register 00 (PRM00)



(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

When this register is used as a compare register and when its value matches the count value of TM00, an interrupt signal (INTTM000) is generated. The count value of TM00 is not cleared.

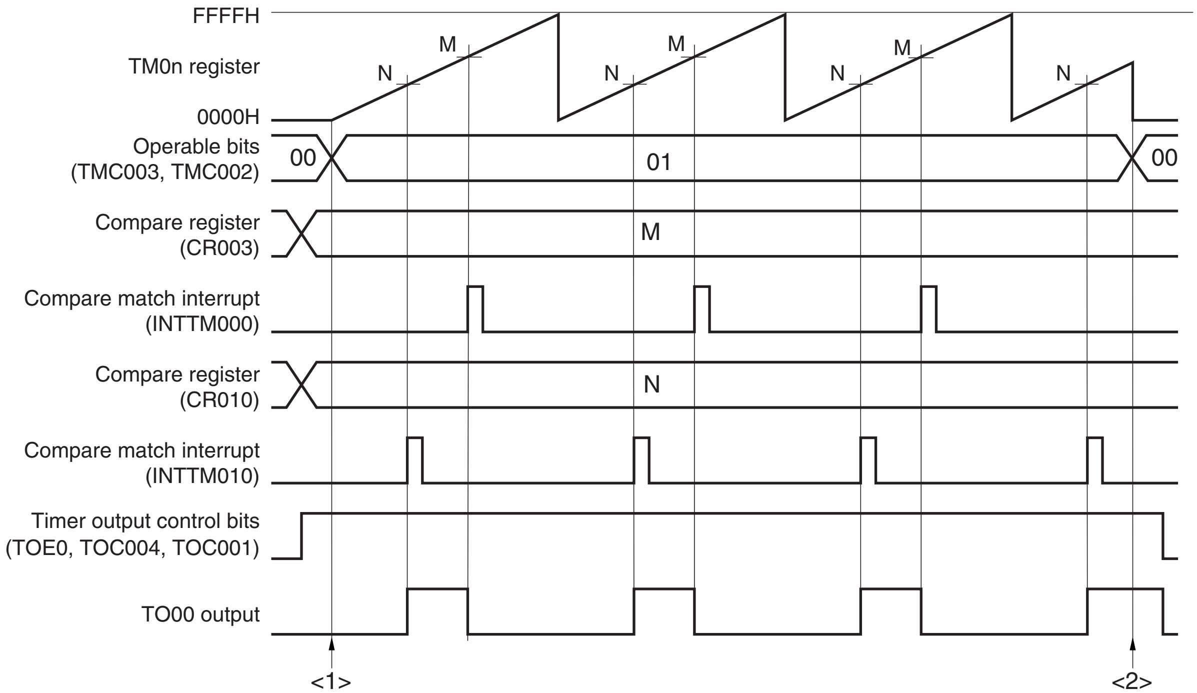
To use this register as a capture register, select either the TI000 or TI010 pin input as a capture trigger. When the valid edge of the capture trigger is detected, the count value of TM00 is stored in CR000.

(g) 16-bit capture/compare register 010 (CR010)

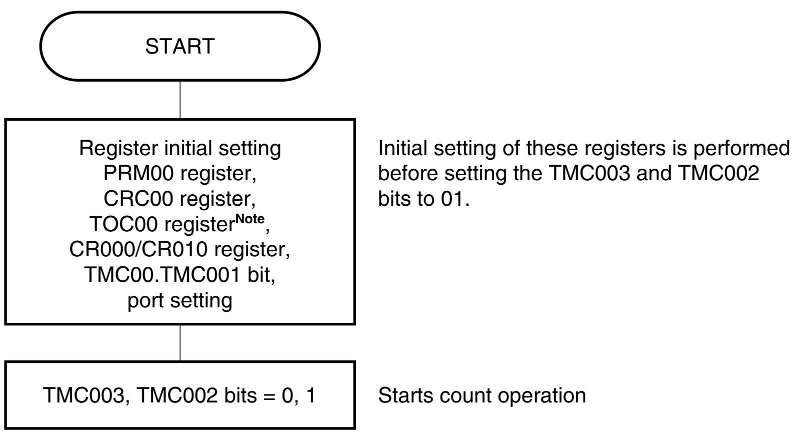
When this register is used as a compare register and when its value matches the count value of TM00, an interrupt signal (INTTM010) is generated. The count value of TM00 is not cleared.

When this register is used as a capture register, the TI000 pin input is used as a capture trigger. When the valid edge of the capture trigger is detected, the count value of TM00 is stored in CR010.

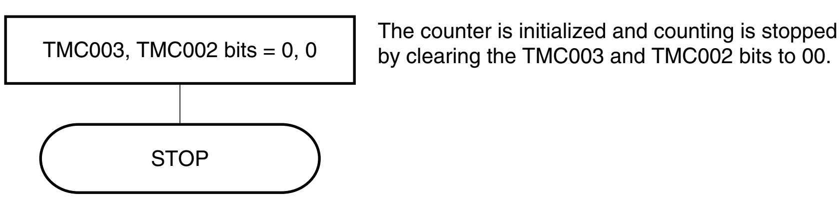
Figure 6-40. Example of Software Processing in Free-Running Timer Mode



<1> Count operation start flow



<2> Count operation stop flow



Note Care must be exercised when setting TOC00. For details, see **6.3 (3) 16-bit timer output control register 00 (TOC00)**.

6.4.6 PPG output operation

A square wave having a pulse width set in advance by CR010 is output from the TO00 pin as a PPG (Programmable Pulse Generator) signal during a cycle set by CR000 when bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) are set to 11 (clear & start upon a match between TM00 and CR000).

The pulse cycle and duty factor of the pulse generated as the PPG output are as follows.

- Pulse cycle = (Set value of CR000 + 1) × Count clock cycle
- Duty = (Set value of CR010 + 1) / (Set value of CR000 + 1)

Caution To change the duty factor (value of CR010) during operation, see 6.5.1 Rewriting CR010 during TM00 operation.

Remarks 1. For the setting of I/O pins, see **6.3 (6) Port mode register 0 (PM0)**.

2. For how to enable the INTTM000 signal interrupt, see **CHAPTER 21 INTERRUPT FUNCTIONS**.

Figure 6-41. Block Diagram of PPG Output Operation

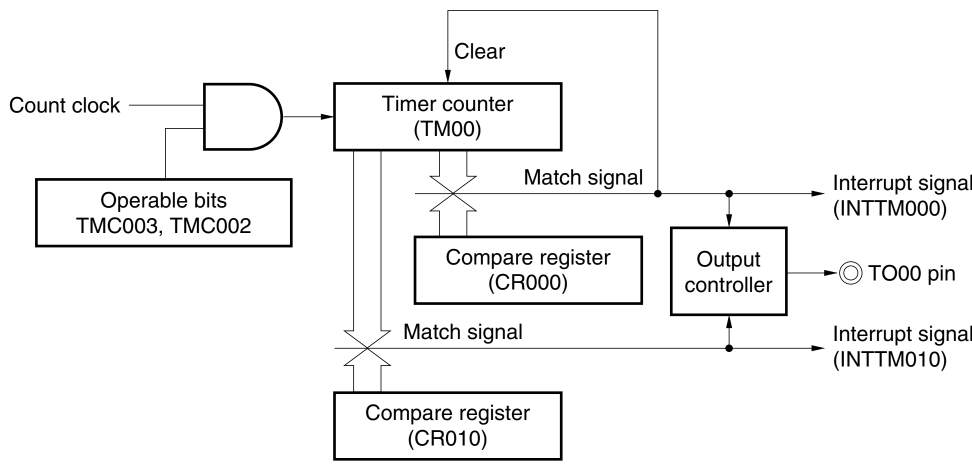


Figure 6-42. Example of Register Settings for PPG Output Operation

(a) 16-bit timer mode control register 00 (TMC00)

				TMC003	TMC002	TMC001	OVF00
0	0	0	0	1	1	0	0

TMC003, TMC002 = 11: Clears and starts on match between TM00 and CR000.

(b) Capture/compare control register 00 (CRC00)

					CRC002	CRC001	CRC000
0	0	0	0	0	0	0	0

CRC000 = 0: CR000 used as compare register

CRC002 = 0: CR010 used as compare register

(c) 16-bit timer output control register 00 (TOC00)

	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
0	0	0	1	0/1	0/1	1	1

TOE00 = 1: Enables TO00 output

LVS00, LVR00: Specifies initial value of TO00 output F/F

TOC004, TOC001 = 11: Inverts TO00 output on match between TM00 and CR000/CR010.

OSPT00, OSPE00 = 00: Disables one-shot pulse output

(d) Prescaler mode register 00 (PRM00)

ES101	ES100	ES001	ES000	3	PRM002	PRM001	PRM000
0	0	0	0	0	0/1	0/1	0/1

PRM002, PRM001, PRM000: Selects count clock

(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

An interrupt signal (INTTM000) is generated when the value of this register matches the count value of TM00.

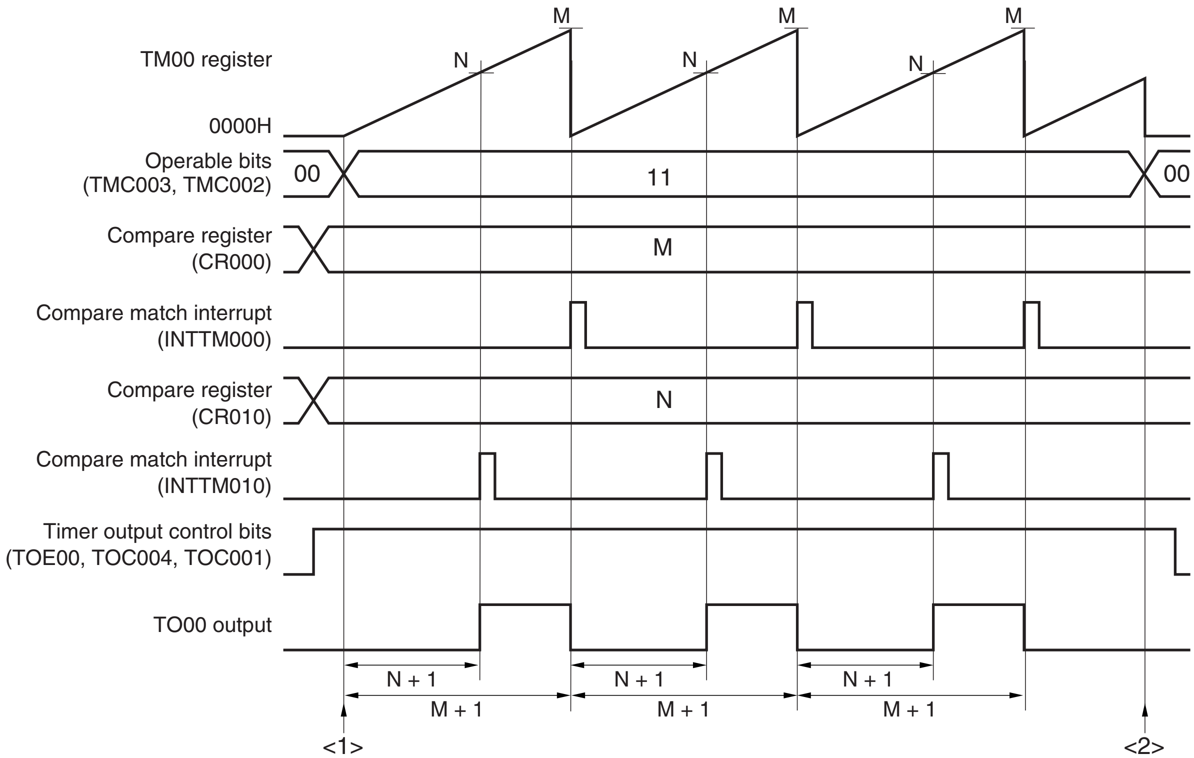
<R> The count value of TM00 is cleared.

(g) 16-bit capture/compare register 010 (CR010)

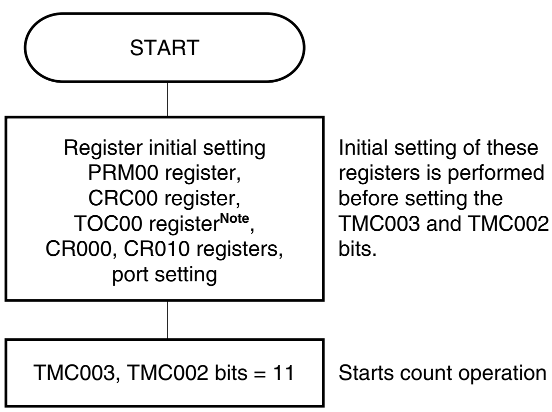
An interrupt signal (INTTM010) is generated when the value of this register matches the count value of TM00. The count value of TM00 is not cleared.

Caution Set values to CR000 and CR010 such that the condition 0000H ≤ CR010 < CR000 ≤ FFFFH is satisfied.

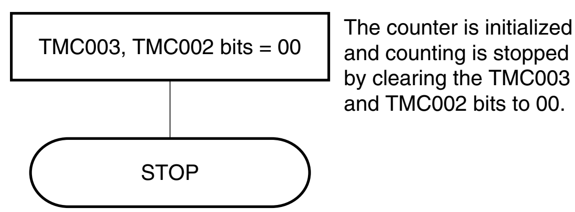
Figure 6-43. Example of Software Processing for PPG Output Operation



<1> Count operation start flow



<2> Count operation stop flow



Note Care must be exercised when setting TOC00. For details, see **6.3 (3) 16-bit timer output control register 00 (TOC00)**.

Remark PPG pulse cycle = (M + 1) × Count clock cycle
PPG duty = (N + 1)/(M + 1)

6.4.7 One-shot pulse output operation

A one-shot pulse can be output by setting bits 3 and 2 (TMC003 and TMC002) of the 16-bit timer mode control register 00 (TMC00) to 01 (free-running timer mode) or to 10 (clear & start mode entered by the TI000 pin valid edge) and setting bit 5 (OSPE00) of 16-bit timer output control register 00 (TOC00) to 1.

When bit 6 (OSPT00) of TOC00 is set to 1 or when the valid edge is input to the TI000 pin during timer operation, clearing & starting of TM00 is triggered, and a pulse of the difference between the values of CR000 and CR010 is output only once from the TO00 pin.

Cautions 1. Do not input the trigger again (setting OSPT00 to 1 or detecting the valid edge of the TI000 pin) while the one-shot pulse is output. To output the one-shot pulse again, generate the trigger after the current one-shot pulse output has completed.

2. To use only the setting of OSPT00 to 1 as the trigger of one-shot pulse output, do not change the level of the TI000 pin or its alternate function port pin. Otherwise, the pulse will be unexpectedly output.

Remarks 1. For the setting of the I/O pins, see **6.3 (6) Port mode register 3 (PM3)**.

2. For how to enable the INTTM000 signal interrupt, see **CHAPTER 21 INTERRUPT FUNCTIONS**.

Figure 6-44. Block Diagram of One-Shot Pulse Output Operation

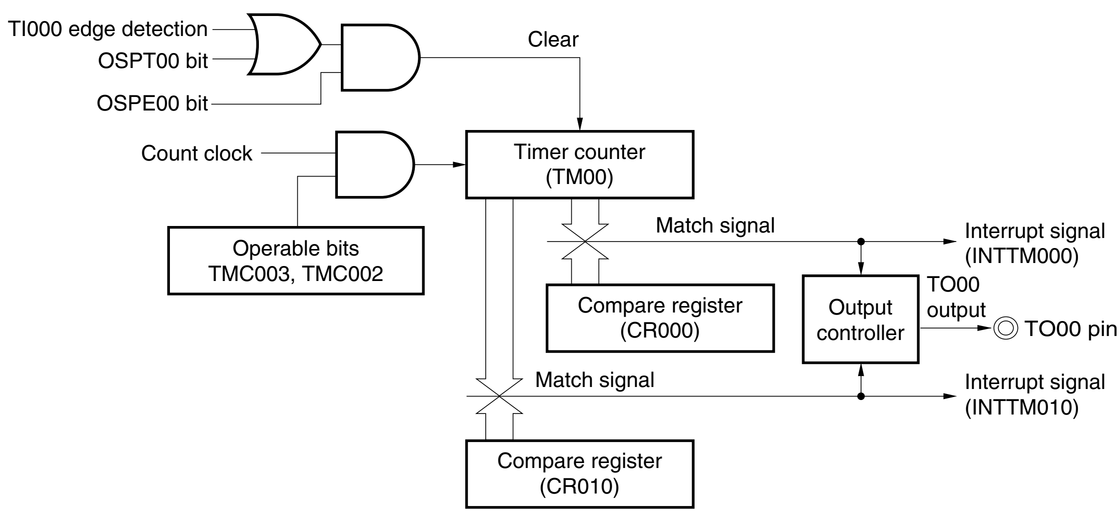


Figure 6-45. Example of Register Settings for One-Shot Pulse Output Operation (1/2)

(a) 16-bit timer mode control register 00 (TMC00)

				TMC003	TMC002	TMC001	OVF00
0	0	0	0	0/1	0/1	0	0

TMC003, TMC002:
01: Free running timer mode
10: Clear and start mode by valid edge of TI000 pin.

(b) Capture/compare control register 00 (CRC00)

					CRC002	CRC001	CRC000
0	0	0	0	0	0	0	0

CRC000: CR000 used as compare register

CRC002: CR010 used as compare register

(c) 16-bit timer output control register 00 (TOC00)

	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
0	0/1	1	1	0/1	0/1	1	1

TOE00: Enables TO00 output

LVS00, LVR00: Specifies initial value of TO00 output

TOC004, TOC001: Inverts TO00 output on match between TM00 and CR000/CR010.

OSPE00: Enables one-shot pulse output

OSPT00: Software trigger is generated by writing 1 to this bit (operation is not affected even if 0 is written to it).

(d) Prescaler mode register 00 (PRM00)

ES101	ES100	ES001	ES000	3	PRM002	PRM001	PRM000
0	0	0	0	0	0/1	0/1	0/1

PRM002, PRM001, PRM000: Selects count clock

Figure 6-45. Example of Register Settings for One-Shot Pulse Output Operation (2/2)

(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

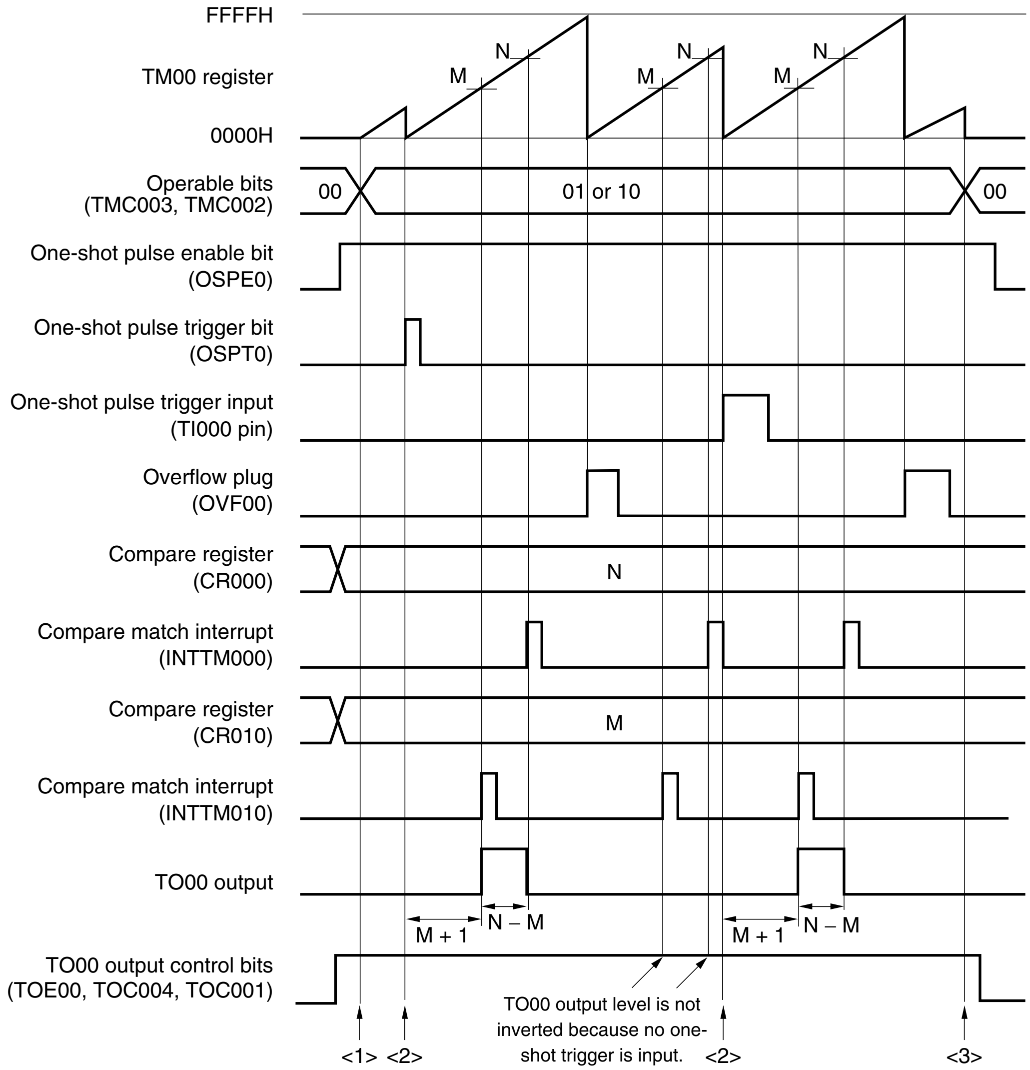
This register is used as a compare register when a one-shot pulse is output. When the value of TM00 matches that of CR000, an interrupt signal (INTTM000) is generated and the TO00 output level is inverted.

(g) 16-bit capture/compare register 010 (CR010)

This register is used as a compare register when a one-shot pulse is output. When the value of TM00 matches that of CR010, an interrupt signal (INTTM010) is generated and the TO00 output level is inverted.

Caution Do not set the same value to CR000 and CR010.

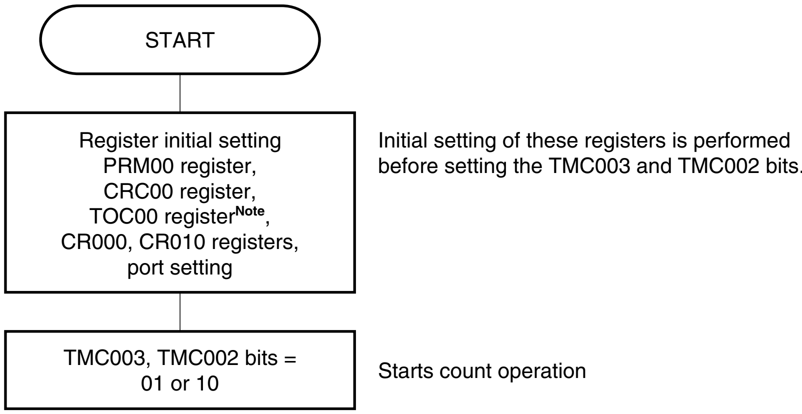
Figure 6-46. Example of Software Processing for One-Shot Pulse Output Operation (1/2)



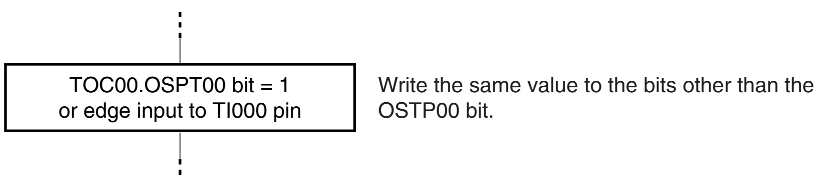
- Time from when the one-shot pulse trigger is input until the one-shot pulse is output
  = (M + 1) × Count clock cycle
- One-shot pulse output active level width
  = (N − M) × Count clock cycle

Figure 6-46. Example of Software Processing for One-Shot Pulse Output Operation (2/2)

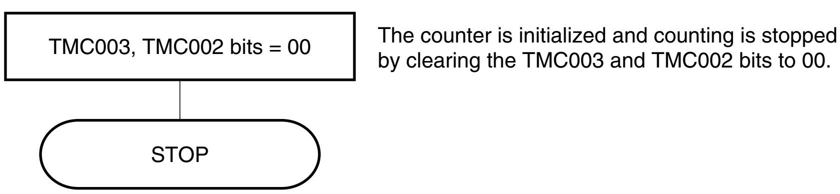
<1> Count operation start flow



<2> One-shot trigger input flow



<3> Count operation stop flow



Note Care must be exercised when setting TOC00. For details, see **6.3 (3) 16-bit timer output control register 00 (TOC00)**.

6.4.8 Pulse width measurement operation

TM00 can be used to measure the pulse width of the signal input to the TI000 and TI010 pins.

Measurement can be accomplished by operating the 16-bit timer/event counter 00 in the free-running timer mode or by restarting the timer in synchronization with the signal input to the TI000 pin.

When an interrupt is generated, read the value of the valid capture register and measure the pulse width. Check bit 0 (OVF00) of 16-bit timer mode control register 00 (TMC00). If it is set (to 1), clear it to 0 by software.

Figure 6-47. Block Diagram of Pulse Width Measurement (Free-Running Timer Mode)

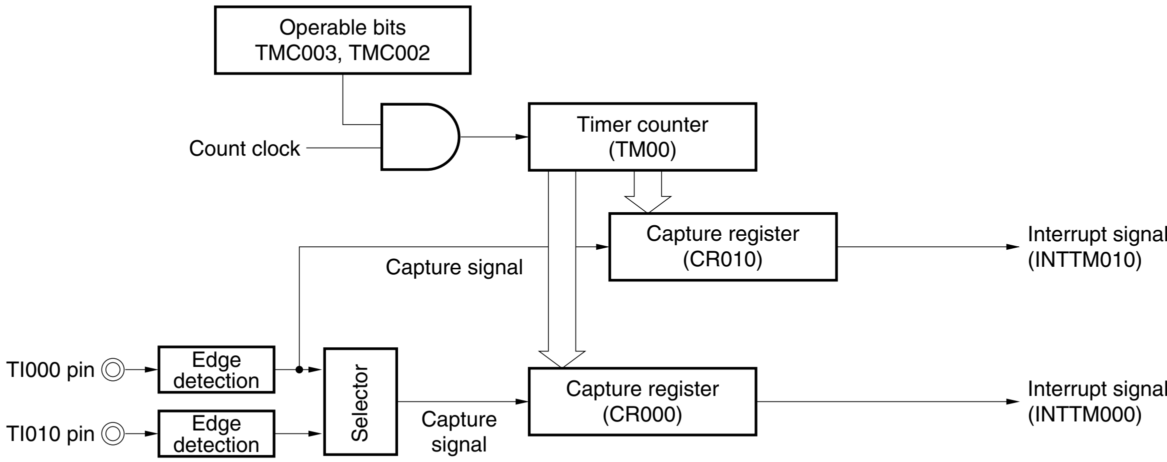
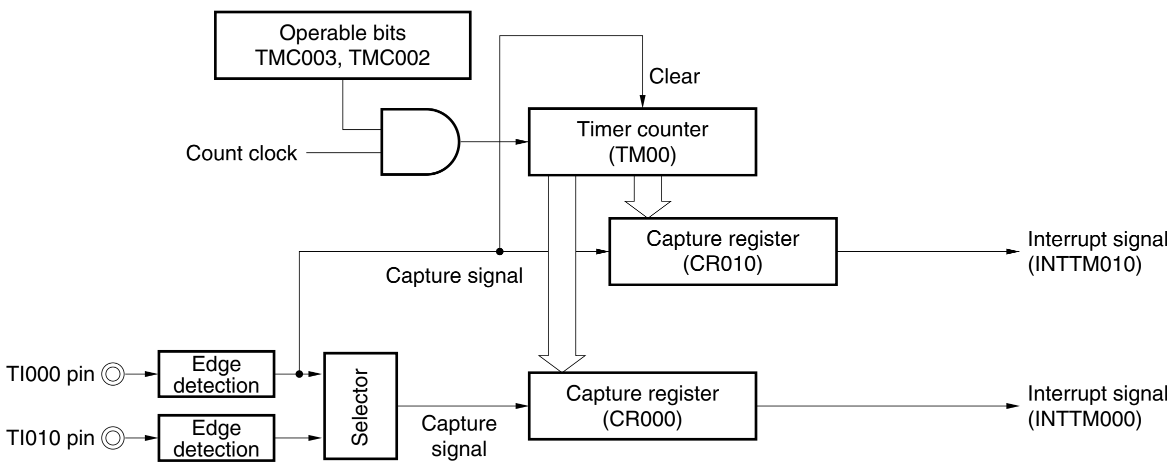


Figure 6-48. Block Diagram of Pulse Width Measurement
(Clear & Start Mode Entered by TI000 Pin Valid Edge Input)



A pulse width can be measured in the following three ways.

- Measuring the pulse width by using two input signals of the TI000 and TI010 pins (free-running timer mode)
- Measuring the pulse width by using one input signal of the TI000 pin (free-running timer mode)
- Measuring the pulse width by using one input signal of the TI000 pin (clear & start mode entered by the TI000 pin valid edge input)

Caution Do not select the TI000 valid edge as the count clock when measuring the pulse width.

Remarks 1. For the setting of the I/O pins, see **6.3 (6) Port mode register 3 (PM3)**.

2. For how to enable the INTTM000 signal interrupt, see **CHAPTER 21 INTERRUPT FUNCTIONS**.

(1) Measuring the pulse width by using two input signals of the TI000 and TI010 pins (free-running timer mode)

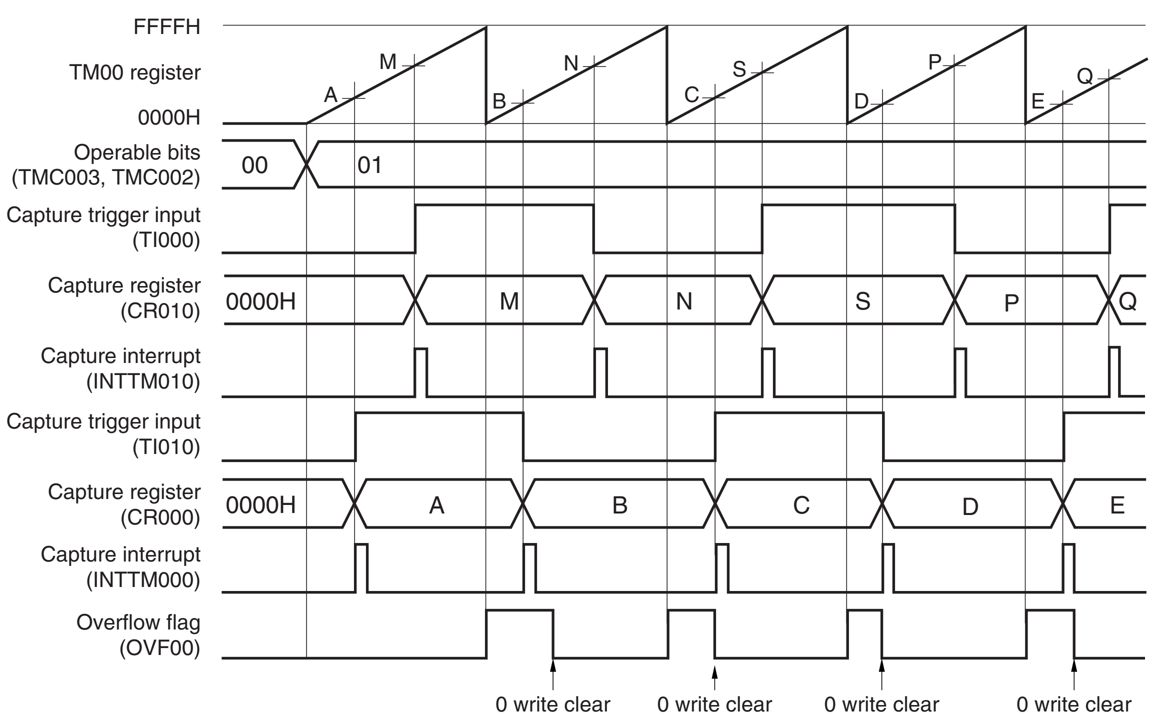
Set the free-running timer mode (TMC003 and TMC002 = 01). When the valid edge of the TI000 pin is detected, the count value of TM00 is captured to CR010. When the valid edge of the TI010 pin is detected, the count value of TM00 is captured to CR000. Specify detection of both the edges of the TI000 and TI010 pins.

By this measurement method, the previous count value is subtracted from the count value captured by the edge of each input signal. Therefore, save the previously captured value to a separate register in advance.

If an overflow occurs, the value becomes negative if the previously captured value is simply subtracted from the current captured value and, therefore, a borrow occurs (bit 0 (CY) of the program status word (PSW) is set to 1). If this happens, ignore CY and take the calculated value as the pulse width. In addition, clear bit 0 (OVF00) of 16-bit timer mode control register 00 (TMC00) to 0.

Figure 6-49. Timing Example of Pulse Width Measurement (1)

• TMC00 = 04H, PRM00 = F0H, CRC00 = 05H



(2) Measuring the pulse width by using one input signal of the TI000 pin (free-running mode)

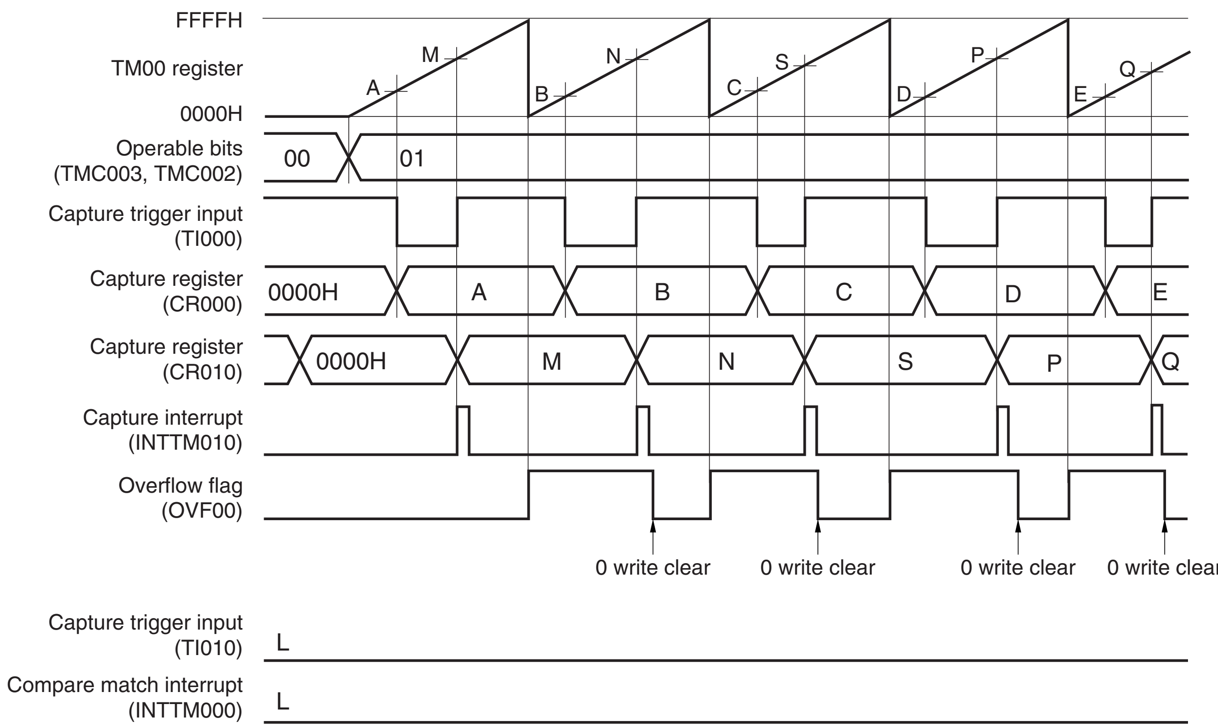
Set the free-running timer mode (TMC003 and TMC002 = 01). The count value of TM00 is captured to CR000 in the phase reverse to the valid edge detected on the TI000 pin. When the valid edge of the TI000 pin is detected, the count value of TM00 is captured to CR010.

By this measurement method, values are stored in separate capture registers when a width from one edge to another is measured. Therefore, the capture values do not have to be saved. By subtracting the value of one capture register from that of another, a high-level width, low-level width, and cycle are calculated.

If an overflow occurs, the value becomes negative if one captured value is simply subtracted from another and, therefore, a borrow occurs (bit 0 (CY) of the program status word (PSW) is set to 1). If this happens, ignore CY and take the calculated value as the pulse width. In addition, clear bit 0 (OVF00) of 16-bit timer mode control register 00 (TMC00) to 0.

Figure 6-50. Timing Example of Pulse Width Measurement (2)

• TMC00 = 04H, PRM00 = 10H, CRC00 = 07H



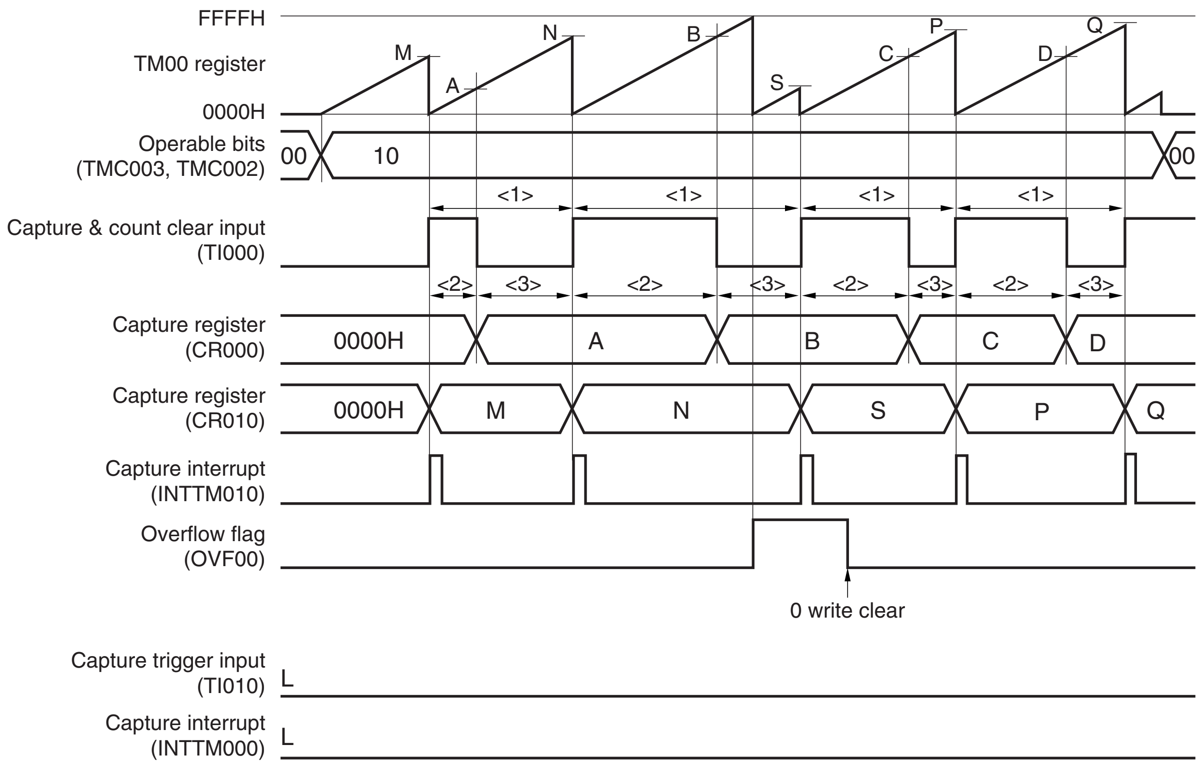
(3) Measuring the pulse width by using one input signal of the TI000 pin (clear & start mode entered by the TI000 pin valid edge input)

Set the clear & start mode entered by the TI000 pin valid edge (TMC003 and TMC002 = 10). The count value of TM00 is captured to CR000 in the phase reverse to the valid edge of the TI000 pin, and the count value of TM00 is captured to CR010 and TM00 is cleared (0000H) when the valid edge of the TI000 pin is detected. Therefore, a cycle is stored in CR010 if TM00 does not overflow.

If an overflow occurs, take the value that results from adding 10000H to the value stored in CR010 as a cycle. Clear bit 0 (OVF00) of 16-bit timer mode control register 00 (TMC00) to 0.

Figure 6-51. Timing Example of Pulse Width Measurement (3)

• TMC00 = 08H, PRM00 = 10H, CRC00 = 07H



<1>	Pulse cycle =	(10000H × Number of times OVF00 bit is set to 1 + Captured value of CR010) × Count clock cycle
<2>	High-level pulse width =	(10000H × Number of times OVF00 bit is set to 1 + Captured value of CR000) × Count clock cycle
<3>	Low-level pulse width =	(Pulse cycle − High-level pulse width)

Figure 6-52. Example of Register Settings for Pulse Width Measurement (1/2)

(a) 16-bit timer mode control register 00 (TMC00)

				TMC003	TMC002	TMC001	OVF00
0	0	0	0	0/1	0/1	0	0

TMC003, TMC002:
- 01: Free running timer mode
- 10: Clear and start mode entered by valid edge of TI000 pin.

(b) Capture/compare control register 00 (CRC00)

					CRC002	CRC001	CRC000
0	0	0	0	0	1	0/1	1

CRC000:
- 1: CR000 used as capture register

CRC001:
- 0: TI010 pin is used as capture trigger of CR000.
- 1: Reverse phase of TI000 pin is used as capture trigger of CR000.

CRC002:
- 1: CR010 used as capture register

(c) 16-bit timer output control register 00 (TOC00)

	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
0	0	0	0	0	0	0	0

(d) Prescaler mode register 00 (PRM00)

ES101	ES100	ES001	ES000	3	PRM002	PRM001	PRM000
0/1	0/1	0/1	0/1	0	0/1	0/1	0/1

PRM002, PRM001, PRM000:
Selects count clock
(setting valid edge of TI000 is prohibited)

ES001, ES000:
- 00: Falling edge detection
- 01: Rising edge detection
- 10: Setting prohibited
- 11: Both edges detection
  (setting when CRC001 = 1 is prohibited)

ES101, ES100:
- 00: Falling edge detection
- 01: Rising edge detection
- 10: Setting prohibited
- 11: Both edges detection

Figure 6-52. Example of Register Settings for Pulse Width Measurement (2/2)

(e) 16-bit timer counter 00 (TM00)

By reading TM00, the count value can be read.

(f) 16-bit capture/compare register 000 (CR000)

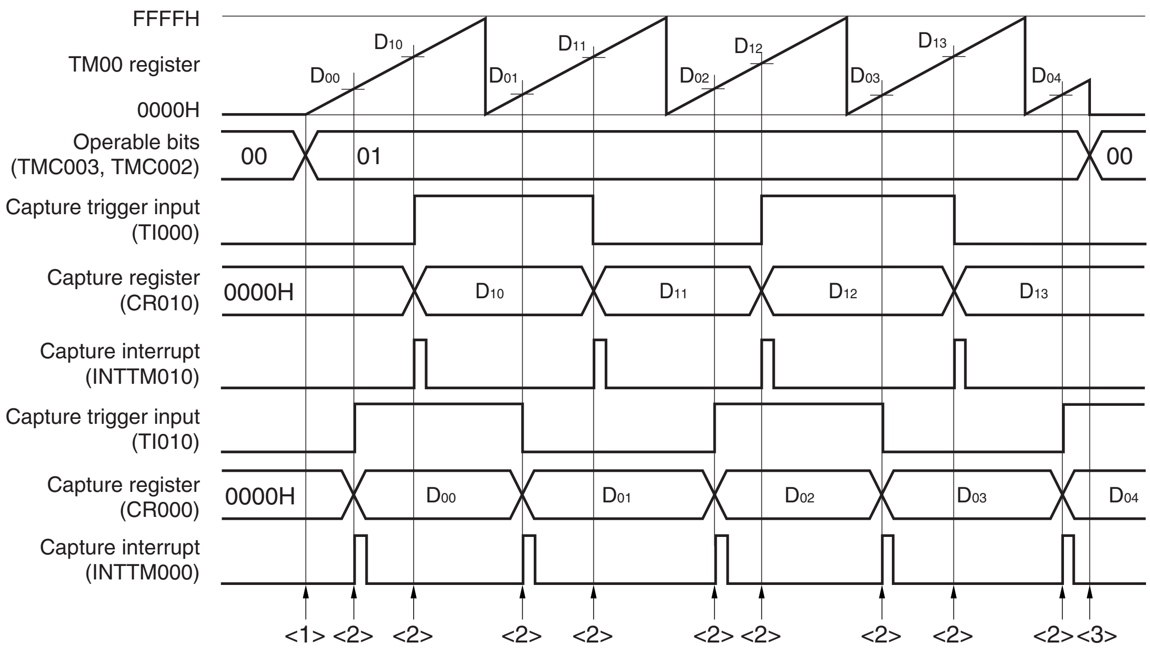
This register is used as a capture register. Either the TI000 or TI010 pin is selected as a capture trigger. When a specified edge of the capture trigger is detected, the count value of TM00 is stored in CR000.

(g) 16-bit capture/compare register 010 (CR010)

This register is used as a capture register. The signal input to the TI000 pin is used as a capture trigger. When the capture trigger is detected, the count value of TM00 is stored in CR010.

Figure 6-53. Example of Software Processing for Pulse Width Measurement (1/2)

(a) Example of free-running timer mode



(b) Example of clear & start mode entered by TI000 pin valid edge

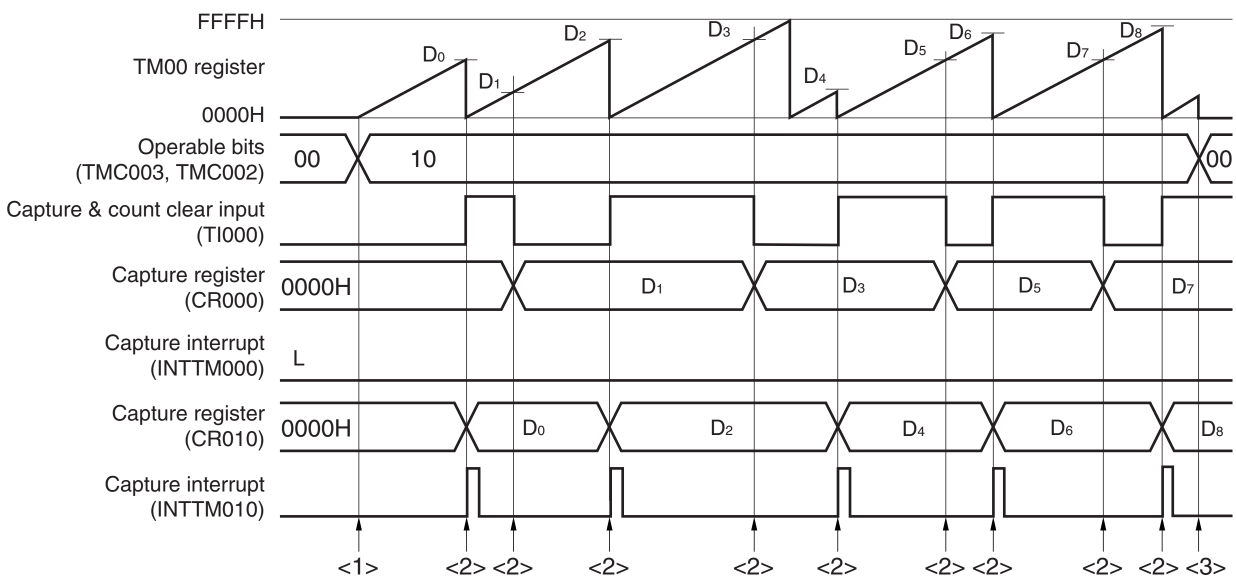
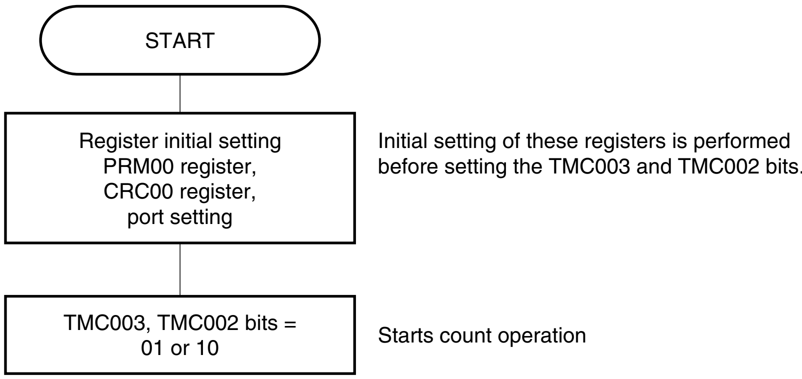
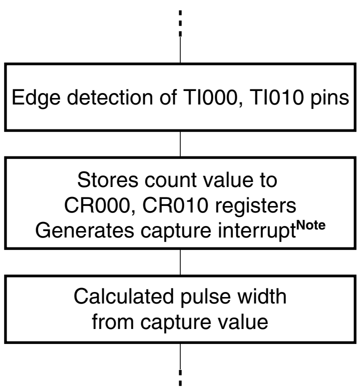


Figure 6-53. Example of Software Processing for Pulse Width Measurement (2/2)

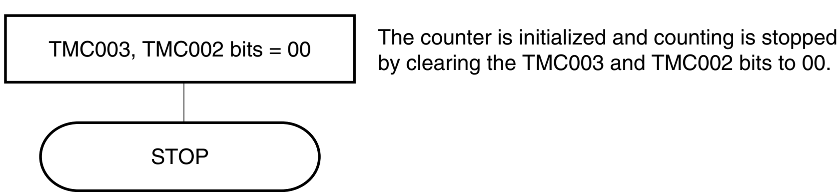
<1> Count operation start flow



<2> Capture trigger input flow



<3> Count operation stop flow



Note The capture interrupt signal (INTTM000) is not generated when the reverse-phase edge of the TI000 pin input is selected to the valid edge of CR000.

6.4.9 External 24-bit event counter operation

16-bit timer/event counter 00 can be operated to function as an external 24-bit event counter, by connecting 16-bit timer/event counter 00 and 8-bit timer/event counter 52 in cascade, and using the external event counter function of 8-bit timer/event counter 52.

It operates as an external 24-bit event counter, by counting the number of external clock pulses input to the TI52 pin via 8-bit timer counter 52 (TM52), and counting the signal which has been output upon a match between the TM52 count value and 8-bit timer compare register 52 (CR52 = FFH[Note]) via 16-bit timer counter 00 (TM00).

When using 16-bit timer/event counter 00 as an external 24-bit event counter, external event input enable can be controlled via 8-bit timer counter H2 output.

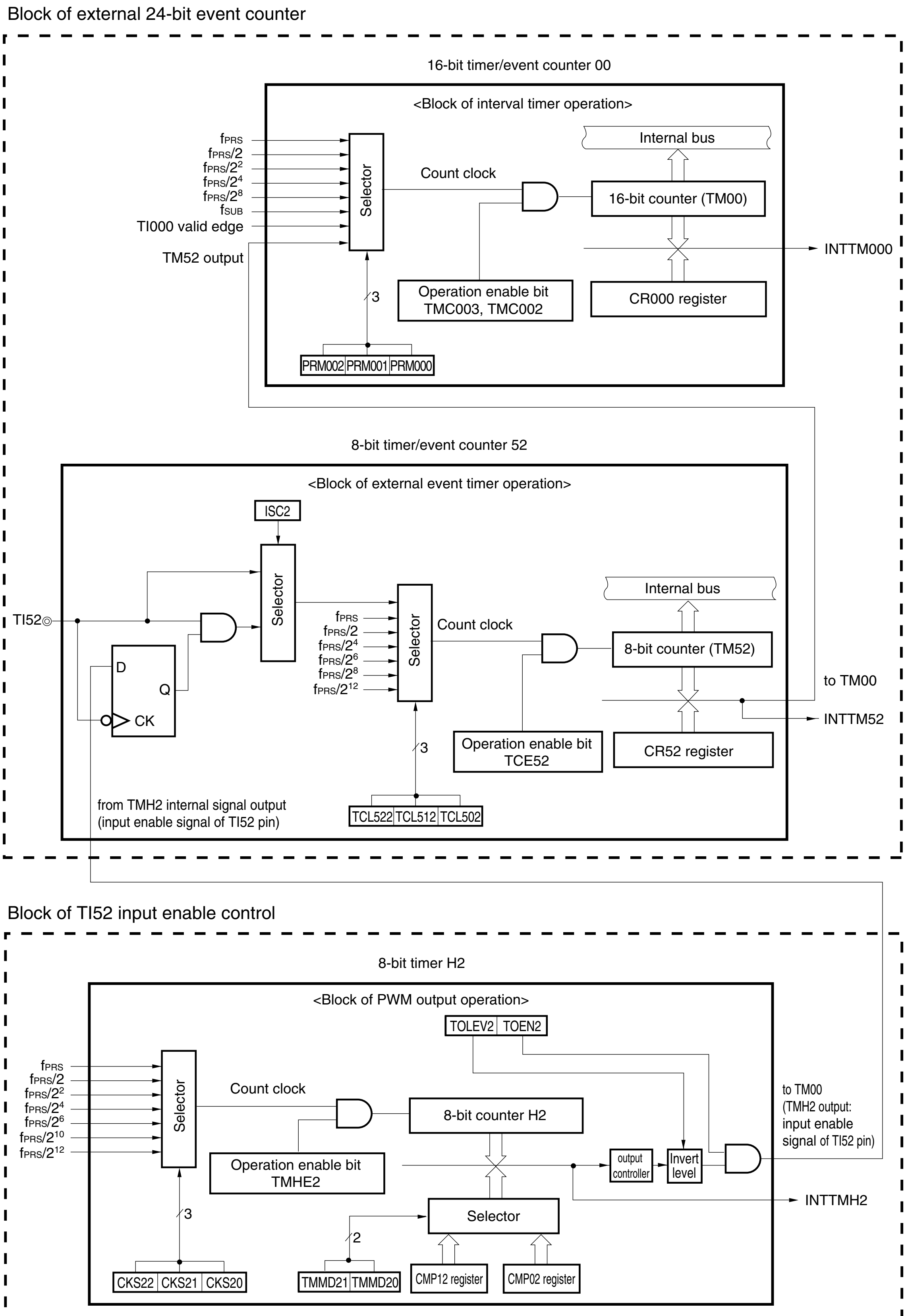
The valid edge of the input to the TI52 pin can be specified by timer clock selection register 52 (TCL52) of 8-bit timer counter 52 (TM52). Also, input enable for TM52 external event input can be controlled via 8-bit timer counter H2 output, by setting bit 2 (ISC2) of the input switch control register (ISC) to “1”.

Count operation using 8-bit timer 52 output as the count clock is started, by setting bits 2, 1, and 0 (PRM002, PRM001, and PRM000) of prescaler mode register 00 (PRM00) of 16-bit timer/event counter 00 to “1”, “1”, and “1” (TM52 output is selected as a count clock), and bits 3 and 2 (TMC003 and TMC002) of 16-bit timer mode control register 00 (TMC00) to “1” and “1” (count clear & start mode entered upon a match between TM00 and CR000). TM00 is cleared to “0” and an interrupt request signal (INTTM000) is generated upon a match between the TM00 count value and 16-bit timer compare register 000 (CR000) value.

Subsequently, INTTM000 is generated upon every match between the TM00 and CR000 values.

Note When operating 16-bit timer/event counter 00 as an external 24-bit event counter, the 8-bit timer compare register 52 (CR52) value must be set to FFH. Also, the TM52 interrupt request signal (INTTM52) must be masked (TMMK52 = 1).

Figure 6-54. Configuration Diagram of External 24-bit Event Counter



Setting

<1> Each mode of TM00 and TM52 is set.

(a) Set TM00 as an interval timer. Select TM52 output as the count clock.

- TMC00: Set to operation prohibited.
  (TMC00 = 00000000B)
- CRC00: Set to operation as a compare register.
  (CRC00 = 000000x0B, x = don't care)
- TOC00: Setting TO00 pin output is prohibited upon a match between CR000 and TM00
  (TOC00 = 00000000B)
- PRM00: TM52 output selected as a count clock.
  (PRM00 = 00000111B)
- CR000: Set the compare value to FFFFH.
  If the compare value is set to M, TM00 will only count up to M.
- CR010: Normally, CR010 is not used, however, a compare match interrupt (INTTM010) is generated upon a match between the CR010 setting value and TM00 value. Therefore, mask the interrupt request by using the interrupt mask flag (TMMK010).

(b) Set TM52 as an external event counter.

- TCL52: Edge selection of TI52 pin input
  Falling edge of TI52 pin → TCL52 = 00H
  Rising edge of TI52 pin → TCL52 = 01H
- CR52: Set the compare register value to FFH.
- TMC52: Count operation is stopped.
  (TMC52 = 00000000B)
- TMIF52: Clear this register.

Caution When operating 16-bit timer/event counter 00 as an external 24-bit event counter, INTTM52 must be masked (TMMK52 = 1). Also, the compare register 52 (CR52) value must be set to FFH.

(c) Set TMH2 to the input enable width adjust mode (PWM mode) for the TI52 pin.[Note]

- TMHMD2: Count operation is stopped, the count clock is selected, the mode is set to input enable width adjust mode (PWM mode), the timer output level default value is set to high level, and timer output is set to enable (TMHMD2 = 0xxx1011B, x = set based on usage conditions).
- CMP02: Compare value (N) frequency setting
- CMP12: Compare value (M) duty setting
  Remark 00H ≤ CMP12 (M) < CMP02 (N) ≤ FFH
- ISC2: Set to ISC2 = 1 (TI52 pin input enable controlled)

Note This setting is not required if input enable for the TI52 pin is not controlled.

<2> TM00, TM52, and TMH2 count operation is started. Timer operation must be started in accordance with the following procedure.

(a) Start TM00 counter operation by setting the TMC003 and TMC002 bits to 1 and 1.
(b) Start TM52 counter operation by setting TCE52 to 1.
(c) Start TMH2 counter operation by setting TMHE2 to 1.[Note]

Note This setting is not required if input enable for the TI52 pin is not controlled.

<3> When the TM52 and CR52 (= FFH) values match, TM52 is cleared to 00, and the match signal causes TM000 to start counting up. Then, when the TM000 and CR000 values match, TM00 is cleared to 0000H, and a match interrupt signal (INTTM000) is generated.
If input enable for the TI52 pin is controlled, external event count values within the input enable periods for the TI52 pin can be measured, by reading TM52, the TM00 count value, and TMIF52 via interrupt servicing by the TMH2 interrupt request signal (INTTMH2).

Figure 6-55. Operation Timing of External 24-bit Event Counter

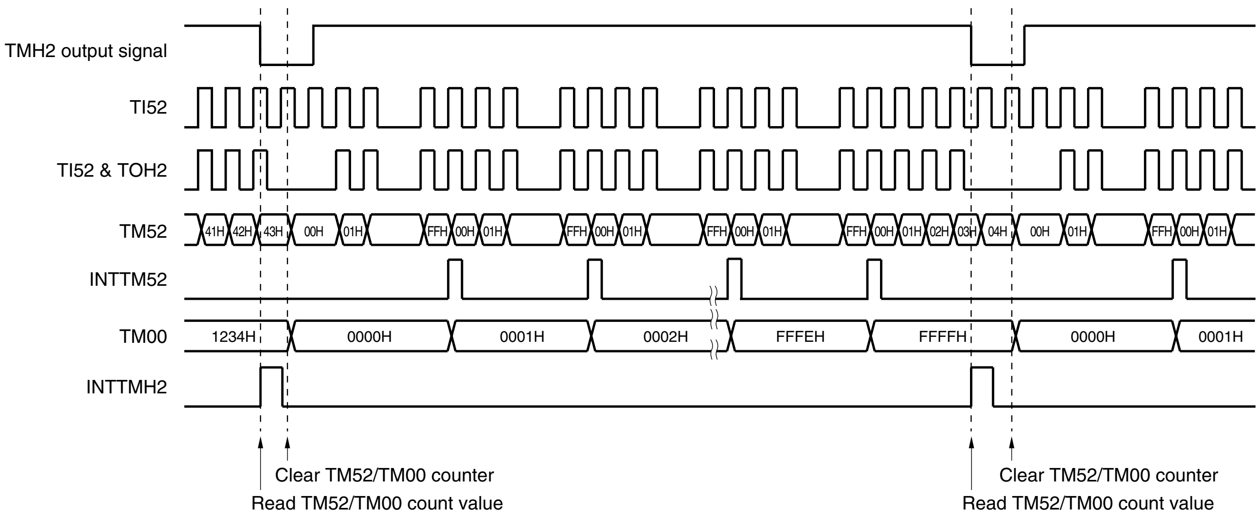
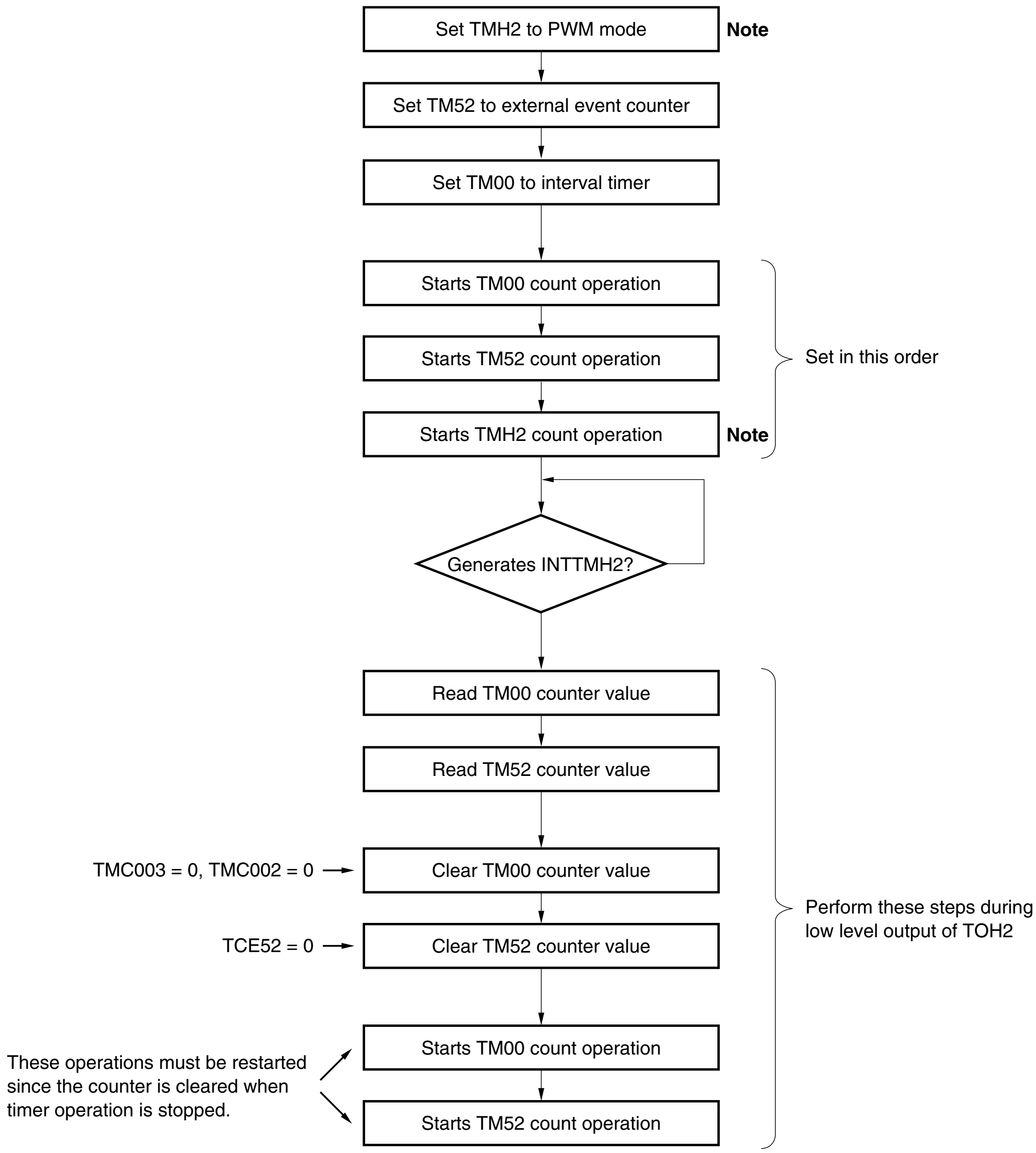


Figure 6-56. Operation Flowchart of External 24-bit Event Counter



Note This setting is not required if input enable for the TI52 pin is not controlled.

6.4.10 Cautions for external 24-bit event counter

(1) 8-bit timer counter H2 output signal

The output level control (default value) of 8-bit timer H2 which is used to control input enable for the TI52 pin, must be set to high level (TOLEV2 = 1). Consequently, an interrupt request signal (INTTMH2) is generated while the input enable signal to the TI52 pin is disabled (TMH2 output: low level), and the TM52 and TM00 count values (= external event count value in input enable period) can be read via servicing of this interrupt.

Note with caution that the input enable signal to the TI52 pin is at high level (enable status) until the TMH2 and CMP02 register values match, after 8-bit timer H2 operation has been enabled (TMHE2 = 1) via this setting (TOLEV2 = 1).

(2) Cautions for input enable control for TI52 pin

The input enable control signal (TMH2 output signal) for the TI52 pin is synchronized by the TI52 pin input clock, as described in **Figure 6-54 Configuration Diagram of External 24-bit Event Counter** and **Figure 6-55 Operation Timing of External 24-bit Event Counter**. Thus, when the counter is operated as an external event counter, an error up to one count may be caused.

(3) Cautions for 16-bit timer/event counter 00 count up during external 24-bit event counter operation

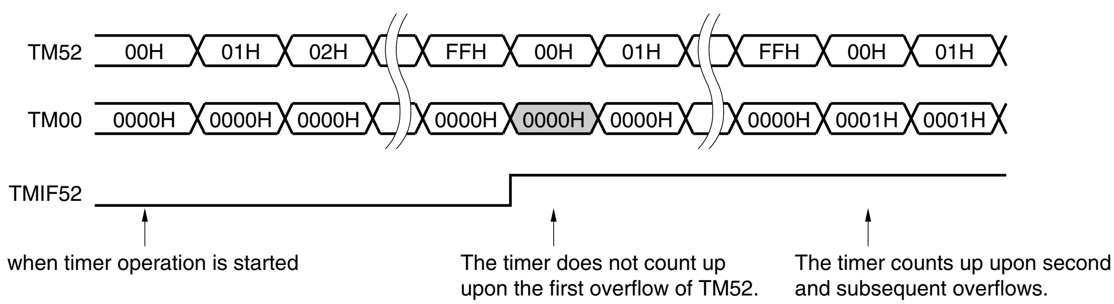
16-bit timer/event counter 00 has an internal synchronization circuit to eliminate noise when starting operation, and the first clock immediately after operation start is not counted.

When using the counter as a 24-bit counter, by setting 16-bit timer/event counter 00 and 8-bit timer/event counter 52 as the higher and lower timer and connecting them in cascade, the interrupt request flag of 8-bit timer/event counter 52 which is the lower timer must be checked as described below, in order to accurately read the 24-bit count values.

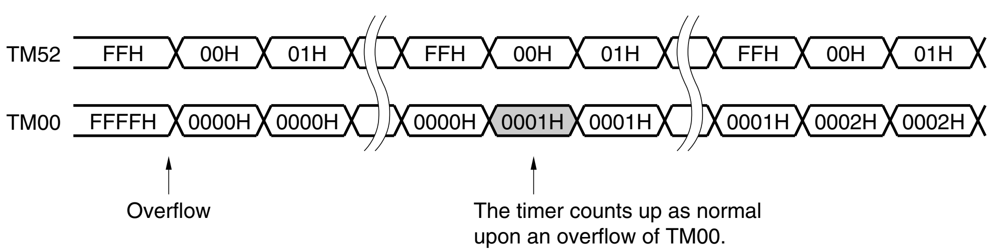
- If TMIF52 = 1 when TM52 and TM00 are read:
  The actual TM00 count value is “read value of TM00 + 1”.
- If TMIF52 = 0 when TM52 and TM00 are read:
  The read value is the correct value.

This phenomenon of 16-bit timer/event counter 00 occurs only when operation is started. A count delay will not occur when 16-bit timer/event counter 00 overflows and the count is restarted from 0000H, since synchronization has already been implemented.

<When starting operation>



<Overflow of higher timer>



6.5 Special Use of TM00

6.5.1 Rewriting CR010 during TM00 operation

In principle, rewriting CR000 and CR010 of the 78K0/LF3 when they are used as compare registers is prohibited while TM00 is operating (TMC003 and TMC002 = other than 00).

However, the value of CR010 can be changed, even while TM00 is operating, using the following procedure if CR010 is used for PPG output and the duty factor is changed (when setting CR010 to a smaller or larger value than the current value, rewrite the CR010 value immediately after a match between CR010 and TM00 or between CR000 and TM00. When CR010 is rewritten immediately before a match between CR010 and TM00 or between CR000 and TM00, an unexpected operation may be performed).

Procedure for changing value of CR010

<1> Disable interrupt INTTM010 (TMMK010 = 1).
<2> Disable reversal of the timer output when the value of TM00 matches that of CR010 (TOC004 = 0).
<3> Change the value of CR010.
<4> Wait for one cycle of the count clock of TM00.
<5> Enable reversal of the timer output when the value of TM00 matches that of CR010 (TOC004 = 1).
<6> Clear the interrupt flag of INTTM010 (TMIF010 = 0) to 0.
<7> Enable interrupt INTTM010 (TMMK010 = 0).

Remark For TMIF010 and TMMK010, see **CHAPTER 21 INTERRUPT FUNCTIONS**.

6.5.2 Setting LVS00 and LVR00

(1) Usage of LVS00 and LVR00

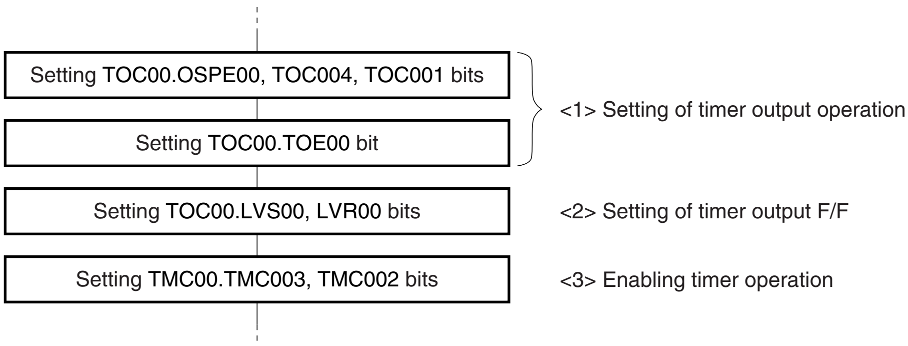
LVS00 and LVR00 are used to set the default value of the TO00 output and to invert the timer output without enabling the timer operation (TMC003 and TMC002 = 00). Clear LVS00 and LVR00 to 00 (default value: low-level output) when software control is unnecessary.

LVS00	LVR00	Timer Output Status
0	0	Not changed (low-level output)
0	1	Cleared (low-level output)
1	0	Set (high-level output)
1	1	Setting prohibited

(2) Setting LVS00 and LVR00

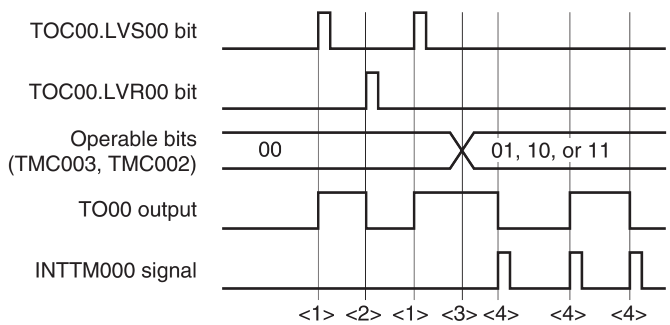
Set LVS00 and LVR00 using the following procedure.

Figure 6-57. Example of Flow for Setting LVS00 and LVR00 Bits



Caution Be sure to set LVS00 and LVR00 following steps <1>, <2>, and <3> above. Step <2> can be performed after <1> and before <3>.

Figure 6-58. Timing Example of LVR00 and LVS00



<1> The TO00 output goes high when LVS00 and LVR00 = 10.

<2> The TO00 output goes low when LVS00 and LVR00 = 01 (the pin output remains unchanged from the high level even if LVS00 and LVR00 are cleared to 00).

<3> The timer starts operating when TMC003 and TMC002 are set to 01, 10, or 11. Because LVS00 and LVR00 were set to 10 before the operation was started, the TO00 output starts from the high level. After the timer starts operating, setting LVS00 and LVR00 is prohibited until TMC003 and TMC002 = 00 (disabling the timer operation).

<4> The TO00 output level is inverted each time an interrupt signal (INTTM000) is generated.

6.6 Cautions for 16-Bit Timer/Event Counter 00

(1) Restrictions for each channel of 16-bit timer/event counter 00

Table 6-3 shows the restrictions for each channel.

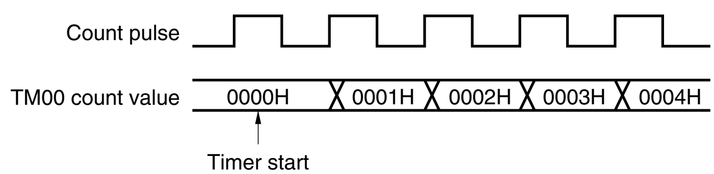
Table 6-3. Restrictions for Each Channel of 16-Bit Timer/Event Counter 00

Operation	Restriction
As interval timer	–
As square wave output	
As external event counter	
As clear & start mode entered by TI000 pin valid edge input	Using timer output (TO00) is prohibited when detection of the valid edge of the TI010 pin is used. (TOC00 = 00H)
As free-running timer	–
As PPG output	0000H ≤ CR010 < CR000 ≤ FFFFH
As one-shot pulse output	Setting the same value to CR000 and CR010 is prohibited.
As pulse width measurement	Using timer output (TO00) is prohibited (TOC00 = 00H)

(2) Timer start errors

An error of up to one clock may occur in the time required for a match signal to be generated after timer start. This is because counting TM00 is started asynchronously to the count pulse.

Figure 6-59. Start Timing of TM00 Count



(3) Setting of CR000 and CR010 (clear & start mode entered upon a match between TM00 and CR000)

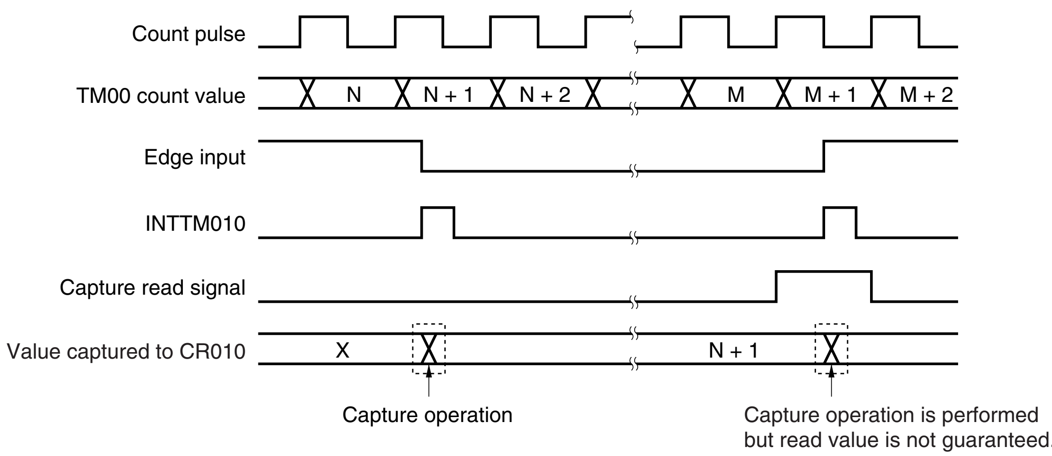
Set a value other than 0000H to CR000 and CR010 (TM00 cannot count one pulse when it is used as an external event counter).

(4) Timing of holding data by capture register

(a) When the valid edge is input to the TI000/TI010 pin and the reverse phase of the TI000 pin is detected while CR000/CR010 is read, CR010 performs a capture operation but the read value of CR000/CR010 is not guaranteed. At this time, an interrupt signal (INTTM000/INTTM010) is generated when the valid edge of the TI000/TI010 pin is detected (the interrupt signal is not generated when the reverse-phase edge of the TI000 pin is detected).

When the count value is captured because the valid edge of the TI000/TI010 pin was detected, read the value of CR000/CR010 after INTTM000/INTTM010 is generated.

Figure 6-60. Timing of Holding Data by Capture Register



(b) The values of CR000 and CR010 are not guaranteed after 16-bit timer/event counter 00 stops.

(5) Setting valid edge

Set the valid edge of the TI000 pin while the timer operation is stopped (TMC003 and TMC002 = 00). Set the valid edge by using ES000 and ES001.

(6) Re-triggering one-shot pulse

Make sure that the trigger is not generated while an active level is being output in the one-shot pulse output mode. Be sure to input the next trigger after the current active level is output.

(7) Operation of OVF00 flag

(a) Setting OVF00 flag (1)

The OVF00 flag is set to 1 in the following case, as well as when TM00 overflows.

Select the clear & start mode entered upon a match between TM00 and CR000.

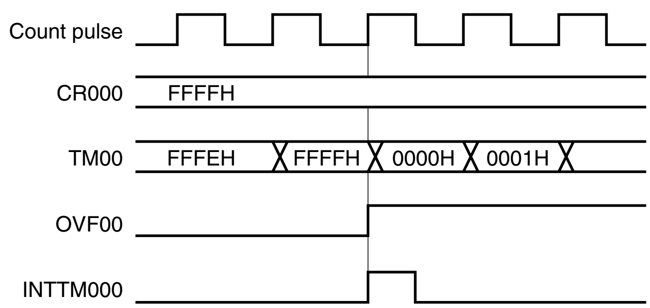
↓

Set CR000 to FFFFH.

↓

When TM00 matches CR000 and TM00 is cleared from FFFFH to 0000H

Figure 6-61. Operation Timing of OVF00 Flag



(b) Clearing OVF00 flag

Even if the OVF00 flag is cleared to 0 after TM00 overflows and before the next count clock is counted (before the value of TM00 becomes 0001H), it is set to 1 again and clearing is invalid.

(8) One-shot pulse output

One-shot pulse output operates correctly in the free-running timer mode or the clear & start mode entered by the TI000 pin valid edge. The one-shot pulse cannot be output in the clear & start mode entered upon a match between TM00 and CR000.

(9) Capture operation

(a) When valid edge of TI000 is specified as count clock

When the valid edge of TI000 is specified as the count clock, the capture register for which TI000 is specified as a trigger does not operate correctly.

(b) Pulse width to accurately capture value by signals input to TI010 and TI000 pins

To accurately capture the count value, the pulse input to the TI000 and TI010 pins as a capture trigger must be wider than two count clocks selected by PRM00 (see **Figure 6-7**).

(c) Generation of interrupt signal

The capture operation is performed at the falling edge of the count clock but the interrupt signals (INTTM000 and INTTM010) are generated at the rising edge of the next count clock (see **Figure 6-7**).

(d) Note when CRC001 (bit 1 of capture/compare control register 00 (CRC00)) is set to 1

When the count value of the TM00 register is captured to the CR000 register in the phase reverse to the signal input to the TI000 pin, the interrupt signal (INTTM000) is not generated after the count value is captured. If the valid edge is detected on the TI010 pin during this operation, the capture operation is not performed but the INTTM000 signal is generated as an external interrupt signal. Mask the INTTM000 signal when the external interrupt is not used.

(10) Edge detection

(a) Specifying valid edge after reset

If the operation of the 16-bit timer/event counter 00 is enabled after reset and while the TI000 or TI010 pin is at high level and when the rising edge or both the edges are specified as the valid edge of the TI000 or TI010 pin, then the high level of the TI000 or TI010 pin is detected as the rising edge. Note this when the TI000 or TI010 pin is pulled up. However, the rising edge is not detected when the operation is once stopped and then enabled again.

(b) Sampling clock for eliminating noise

The sampling clock for eliminating noise differs depending on whether the valid edge of TI000 is used as the count clock or capture trigger. In the former case, the sampling clock is fixed to f_{PRS}. In the latter, the count clock selected by PRM00 is used for sampling.

When the signal input to the TI000 pin is sampled and the valid level is detected two times in a row, the valid edge is detected. Therefore, noise having a short pulse width can be eliminated (see **Figure 6-7**).

(11) Timer operation

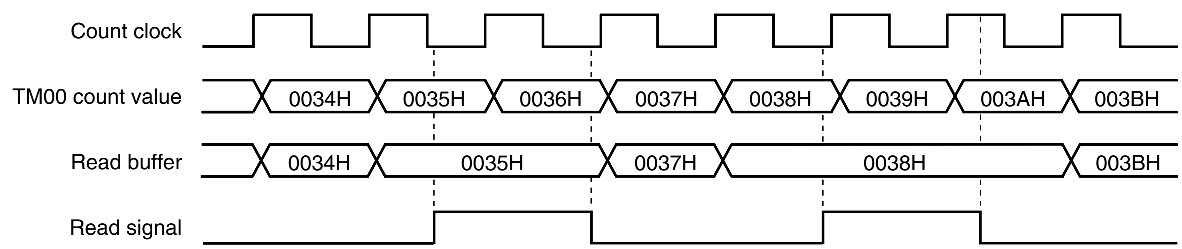
The signal input to the TI000/TI010 pin is not acknowledged while the timer is stopped, regardless of the operation mode of the CPU.

Remark f_{PRS}: Peripheral hardware clock frequency

<R> **(12) Reading of 16-bit timer counter 00 (TM00)**

TM00 can be read without stopping the actual counter, because the count values captured to the buffer are fixed when it is read. The buffer, however, may not be updated when it is read immediately before the counter counts up, because the buffer is updated at the timing the counter counts up.

Figure 6-62. 16-bit Timer Counter 00 (TM00) Read Timing



CHAPTER 7 8-BIT TIMER/EVENT COUNTERS 50, 51, AND 52

7.1 Functions of 8-Bit Timer/Event Counters 50, 51, and 52

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

- Interval timer
- External event counter[Note 1]
- Square-wave output[Note 2]
- PWM output[Note 2]

Notes 1. TM52 and TM00 can be connected in cascade to be used as an external 24-bit event counter. Also, the external event input of TM52 can be input enable-controlled via TMH2. For details, see **CHAPTER 6 16-BIT TIMER/EVENT COUNTER 00**.

2. TM50 and TM51 only.

7.2 Configuration of 8-Bit Timer/Event Counters 50, 51, and 52

8-bit timer/event counters 50, 51, and 52 include the following hardware.

Table 7-1. Configuration of 8-Bit Timer/Event Counters 50, 51, and 52

Item	Configuration
Timer register	8-bit timer counter 5n (TM5n)
Register	8-bit timer compare register 5n (CR5n)
Timer input	TI5n
Timer output	TO50, TO51
Control registers	Timer clock selection register 5n (TCL5n) 8-bit timer mode control register 5n (TMC5n) Input switch control register (ISC) Port mode register 3 (PM3) or port mode register 4 (PM4) Port register 3 (P3) or port register 4 (P4)

Remark n = 0 to 2

Figures 7-1 to 7-3 show the block diagrams of 8-bit timer/event counters 50, 51, and 52.

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

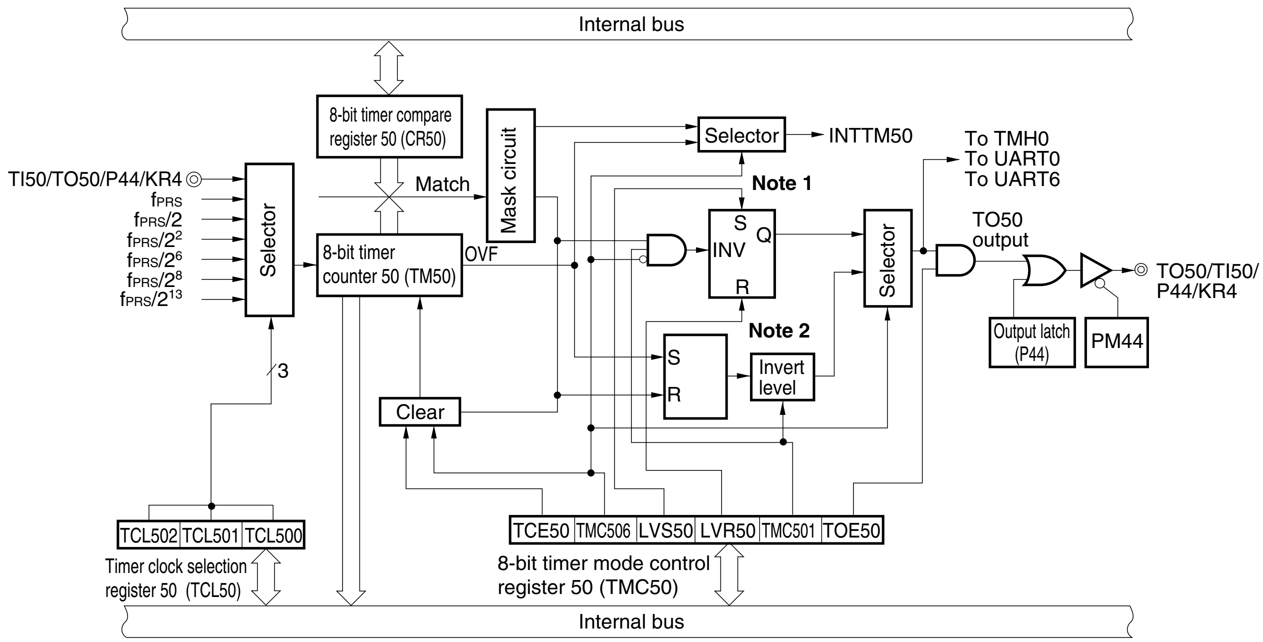
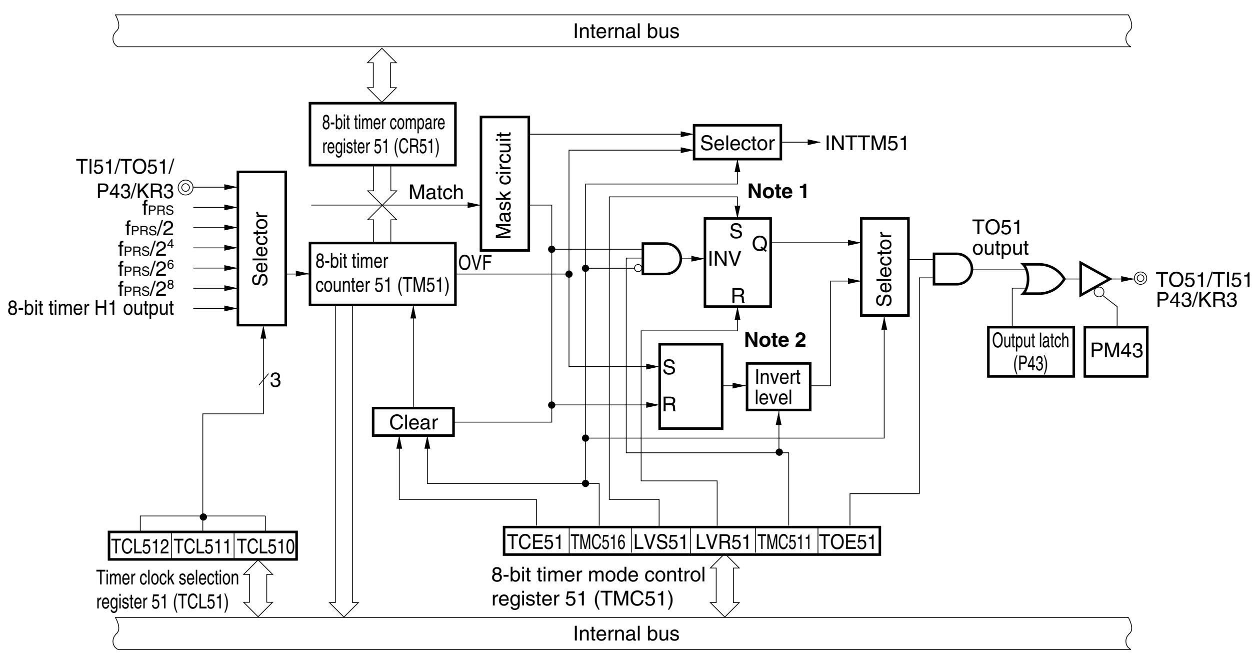


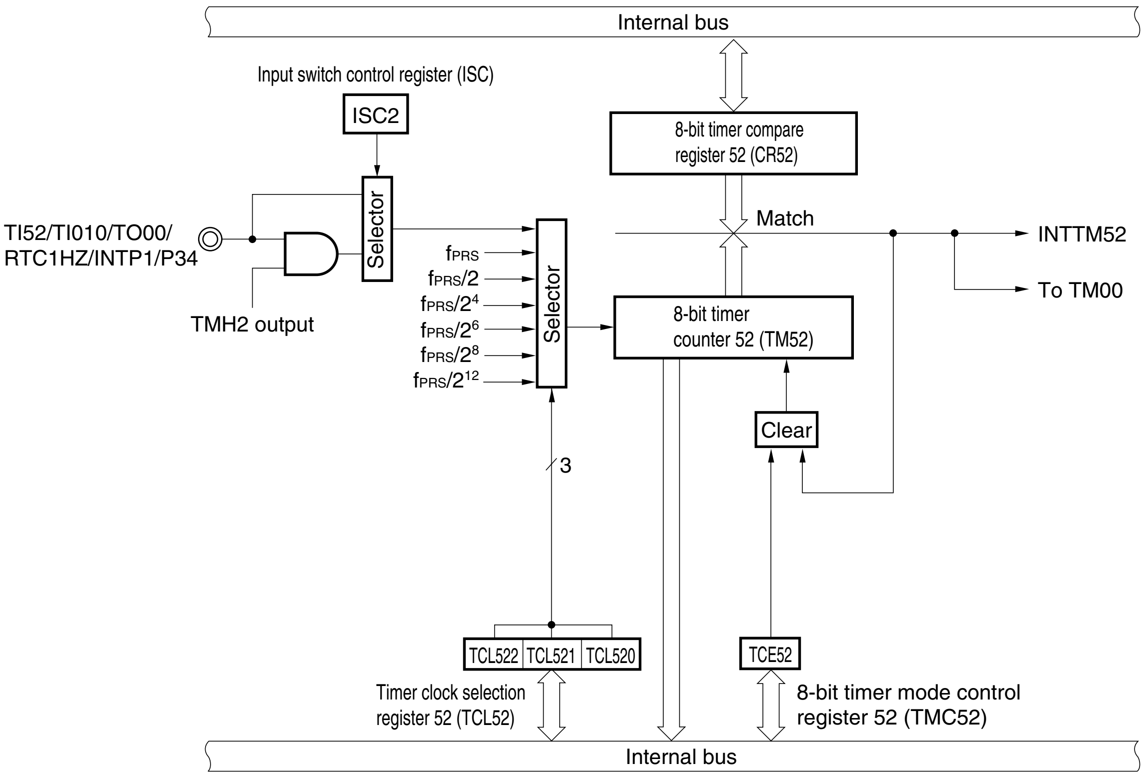
Figure 7-2. Block Diagram of 8-Bit Timer/Event Counter 51



Notes 1. Timer output F/F

2. PWM output F/F

Figure 7-3. Block Diagram of 8-Bit Timer/Event Counter 52



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

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

The counter is incremented in synchronization with the rising edge of the count clock.

Figure 7-4. Format of 8-Bit Timer Counter 5n (TM5n)

Address: FF16H (TM50), FF6FH (TM51), FF51H (TM52) After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
TM5n (n = 0-2)								

In the following situations, the count value is cleared to 00H.

<1> Reset signal generation

<2> When TCE5n is cleared

<3> When TM5n and CR5n match in the mode in which clear & start occurs upon a match of the TM5n and CR5n.

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

CR5n can be read and written by an 8-bit memory manipulation instruction.

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

In the PWM mode, the TO5n output becomes inactive when the values of TM5n and CR5n match, but no interrupt is generated.

The value of CR5n can be set within 00H to FFH.

Reset signal generation sets CR5n to 00H.

Figure 7-5. Format of 8-Bit Timer Compare Register 5n (CR5n)

Address: FF17H (CR50), FF41H (CR51), FF59H (CR52) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CR5n (n = 0-2)								

Cautions 1. In the mode in which clear & start occurs on a match of TM5n and CR5n (TMC5n6 = 0), do not write other values to CR5n during operation.

2. In PWM mode, make the CR5n rewrite period 3 count clocks of the count clock (clock selected by TCL5n) or more.

Remark n = 0 to 2

7.3 Registers Controlling 8-Bit Timer/Event Counters 50, 51, and 52

The following five registers are used to control 8-bit timer/event counters 50, 51, and 52.

- Timer clock selection register 5n (TCL5n)
- 8-bit timer mode control register 5n (TMC5n)
- Input switch control register (ISC)
- Port mode register 3 (PM3) or port mode register 4 (PM4)
- Port register 3 (P3) or port register 4 (P4)

(1) Timer clock selection register 5n (TCL5n)

This register sets the count clock of 8-bit timer/event counter 5n and the valid edge of the TI5n pin input.
TCL5n can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets TCL5n to 00H.

Remark n = 0 to 2

Figure 7-6. Format of Timer Clock Selection Register 50 (TCL50)

Address: FF6AH After reset: 00H R/W

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

TCL502	TCL501	TCL500	Count clock selection[Note 1]			
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz
0	0	0	TI50 pin falling edge			
0	0	1	TI50 pin rising edge			
0	1	0	f_{PRS}[Note 2]	2 MHz	5 MHz	10 MHz
0	1	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz
1	0	0	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz
1	0	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz
1	1	0	$f_{PRS}/2^8$	7.81 kHz	19.53 kHz	39.06 kHz
1	1	1	$f_{PRS}/2^{13}$	0.24 kHz	0.61 kHz	1.22 kHz

Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{XH}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 2.7 to 5.5 V: $f_{PRS} \leq 10$ MHz
- V_{DD} = 1.8 to 2.7 V: $f_{PRS} \leq 5$ MHz

2. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{RH}) (XSEL = 0), when 1.8 V $\leq V_{DD} <$ 2.7 V, the setting of TCL502, TCL501, TCL500 = 0, 1, 0 (count clock: f_{PRS}) is prohibited.

Cautions 1. When rewriting TCL50 to other data, stop the timer operation beforehand.

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

Remark f_{PRS}: Peripheral hardware clock frequency

Figure 7-7. Format of Timer Clock Selection Register 51 (TCL51)

Address: FF8CH After reset: 00H R/W

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

TCL512	TCL511	TCL510	Count clock selection[Note 1]			
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz
0	0	0	TI51 pin falling edge			
0	0	1	TI51 pin rising edge			
0	1	0	f_{PRS}[Note 2]	2 MHz	5 MHz	10 MHz
0	1	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz
1	0	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz
1	0	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz
1	1	0	$f_{PRS}/2^8$	7.81 kHz	19.53 kHz	39.06 kHz
1	1	1	Timer H1 output signal			

Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{XH}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 2.7 to 5.5 V: $f_{PRS} \leq$ 10 MHz
- V_{DD} = 1.8 to 2.7 V: $f_{PRS} \leq$ 5 MHz

2. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{RH}) (XSEL = 0), when 1.8 V ≤ V_{DD} < 2.7 V, the setting of TCL512, TCL511, TCL510 = 0, 1, 0 (count clock: f_{PRS}) is prohibited.

Cautions 1. When rewriting TCL51 to other data, stop the timer operation beforehand.

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

Figure 7-8. Format of Timer Clock Selection Register 52 (TCL52)

Address: FF5BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL52	0	0	0	0	0	TCL522	TCL521	TCL520

TCL522	TCL521	TCL520	Count clock selection[Note 1]			
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz
0	0	0	Falling edge of clock selected by ISC2			
0	0	1	Rising edge of clock selected by ISC2			
0	1	0	f_{PRS}[Note 2]	2 MHz	5 MHz	10 MHz
0	1	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz
1	0	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz
1	0	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz
1	1	0	$f_{PRS}/2^8$	7.81 kHz	19.53 kHz	39.06 kHz
1	1	1	$f_{PRS}/2^{12}$	0.49 kHz	1.22 kHz	2.44 kHz

Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{XH}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 2.7 to 5.5 V: $f_{PRS} \leq 10$ MHz
- V_{DD} = 1.8 to 2.7 V: $f_{PRS} \leq 5$ MHz

2. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{RH}) (XSEL = 0), when 1.8 V ≤ V_{DD} < 2.7 V, the setting of TCL522, TCL521, TCL520 = 0, 1, 0 (count clock: f_{PRS}) is prohibited.

Cautions 1. When rewriting TCL52 to other data, stop the timer operation beforehand.

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

Remark f_{PRS}: Peripheral hardware clock frequency

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

TMC5n is a register that performs the following five types of settings.

<1> 8-bit timer counter 5n (TM5n) count operation control
<2> 8-bit timer counter 5n (TM5n) operating mode selection[Note]
<3> Timer output F/F (flip flop) status setting[Note]
<4> Active level selection in timer F/F control or PWM (free-running) mode[Note]
<5> Timer output control[Note]

TMC5n can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to 00H.

Note TM50 and TM51 only.

Remark n = 0 to 2

Figure 7-9. Format of 8-Bit Timer Mode Control Register 50 (TMC50)

Address: FF6BH After reset: 00H R/W[Note]

Symbol	<7>	6	5	4	<3>	<2>	1	<0>
TMC50	TCE50	TMC506	0	0	LVS50	LVR50	TMC501	TOE50

TCE50	TM50 count operation control
0	After clearing to 0, count operation disabled (counter stopped)
1	Count operation start

TMC506	TM50 operating mode selection
0	Mode in which clear & start occurs on a match between TM50 and CR50
1	PWM (free-running) mode

LVS50	LVR50	Timer output F/F status setting
0	0	No change
0	1	Timer output F/F clear (0) (default value of TO50 output: low level)
1	0	Timer output F/F set (1) (default value of TO50 output: high level)
1	1	Setting prohibited

TMC501	In other modes (TMC506 = 0)	In PWM mode (TMC506 = 1)
	Timer F/F control	Active level selection
0	Inversion operation disabled	Active-high
1	Inversion operation enabled	Active-low

TOE50	Timer output control
0	Output disabled (TO50 output is low level)
1	Output enabled

Note Bits 2 and 3 are write-only.

(**Cautions** and **Remarks** are listed on the next page.)

Figure 7-10. Format of 8-Bit Timer Mode Control Register 51 (TMC51)

Address: FF43H After reset: 00H R/W[Note]

Symbol	<7>	6	5	4	<3>	<2>	1	<0>
TMC51	TCE51	TMC516	0	0	LVS51	LVR51	TMC511	TOE51

TCE51	TM51 count operation control
0	After clearing to 0, count operation disabled (counter stopped)
1	Count operation start

TMC516	TM51 operating mode selection
0	Mode in which clear & start occurs on a match between TM51 and CR51
1	PWM (free-running) mode

LVS51	LVR51	Timer output F/F status setting
0	0	No change
0	1	Timer output F/F clear (0) (default value of TO51 output: low)
1	0	Timer output F/F set (1) (default value of TO51 output: high)
1	1	Setting prohibited

TMC511	In other modes (TMC516 = 0)	In PWM mode (TMC516 = 1)
	Timer F/F control	Active level selection
0	Inversion operation disabled	Active-high
1	Inversion operation enabled	Active-low

TOE51	Timer output control
0	Output disabled (TO51 output is low level)
1	Output enabled

Note Bits 2 and 3 are write-only.

Cautions 1. The settings of LVS5n and LVR5n are valid in other than PWM mode.

2. Perform <1> to <4> below in the following order, not at the same time.

<1> Set TMC5n1, TMC5n6: Operation mode setting

<2> Set TOE5n to enable output: Timer output enable

<3> Set LVS5n, LVR5n (see Caution 1): Timer F/F setting

<4> Set TCE5n

3. When TCE5n = 1, setting the other bits of TMC5n is prohibited.

4. The actual TO50/TI50/P44/KR4 and TO51/TI51/P43/KR3 pin outputs are determined depending on PM44 and P44, and PM43 and P43, besides TO5n output.

Remarks 1. In PWM mode, PWM output is made inactive by clearing TCE5n to 0.

2. If LVS5n and LVR5n are read, the value is 0.

3. The values of the TMC5n6, LVS5n, LVR5n, TMC5n1, and TOE5n bits are reflected at the TO5n pin regardless of the value of TCE5n.

4. n = 0, 1

Figure 7-11. Format of 8-Bit Timer Mode Control Register 52 (TMC52)

Address: FF5CH After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
TMC52	TCE52	0	0	0	0	0	0	0

TCE52	TM52 count operation control
0	After clearing to 0, count operation disabled (counter stopped)
1	Count operation start

Caution Be sure to clear bits 0 to 6 to 0.

(3) Input switch control register (ISC)

By setting ISC2 to 1, the TI52 input signal can be controlled via the TOH2 output signal.
This register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to 00H.

Figure 7-12. Format of Input Switch Control Register (ISC)

Address: FF4FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	ISC5	ISC4	ISC3	ISC2	ISC1	ISC0

ISC5	ISC4	TxD6, RxD6 input source selection
0	0	TxD6:P112, RxD6: P113
0	1	TxD6:P16, RxD6: P15
Other than above		Setting prohibited

ISC3	RxD6/P113 input enabled/disabled
0	RxD6/P113 input disabled
1	RxD6/P113 input enabled

ISC2	TI52 input source control
0	No enable control of TI52 input (P34)
1	Enable controlled of TI52 input (P34) Note 1

ISC1	TI000 input source selection
0	TI000 (P33)
1	RxD6 (P15 or P113 Note 2)

ISC0	INTP0 input source selection
0	INTP0 (P120)
1	RxD6 (P15 or P113 Note 2)

Notes 1. TI52 input is controlled by TOH2 output signal.

<R> **2.** This is selected by ISC5 and ISC4.

(4) Port mode registers 3 and 4 (PM3, PM4)

These registers set port 3 and 4 input/output in 1-bit units.

When using the P44/TO50/TI50/KR4 and P43/TO51/TI51/KR3 pins for timer output, clear PM44 and PM43 and the output latches of P44 and P43 to 0.

When using the P44/TO50/TI50/KR4, P43/TO51/TI51/KR3, and P34/TI52/TI010/TO00/RTC1HZ/INTP1 pins for timer input, set PM44, PM43, and PM34 to 1. The output latches of P44, PM43, and PM34 at this time may be 0 or 1.

PM3 and PM4 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 7-13. Format of Port Mode Register 3 (PM3)

Address: FF23H After reset: FFH R/W

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

PM3n	P1n pin I/O mode selection (n = 0 to 4)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Figure 7-14. Format of Port Mode Register 4 (PM4)

Address: FF24H After reset: FFH R/W

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

PM4n	P4n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

7.4 Operations of 8-Bit Timer/Event Counters 50, 51, and 52

7.4.1 Operation as interval timer

8-bit timer/event counter 5n operates as an interval timer that generates interrupt requests repeatedly at intervals of the count value preset to 8-bit timer compare register 5n (CR5n).

When the count value of 8-bit timer counter 5n (TM5n) matches the value set to CR5n, counting continues with the TM5n value cleared to 0 and an interrupt request signal (INTTM5n) is generated.

The count clock of TM5n can be selected with bits 0 to 2 (TCL5n0 to TCL5n2) of timer clock selection register 5n (TCL5n).

Setting

<1> Set the registers.
- TCL5n: Select the count clock.
- CR5n: Compare value
- TMC5n: Stop the count operation, select the mode in which clear & start occurs on a match of TM5n and CR5n.
  (TMC5n = 0000×××0B × = Don't care)

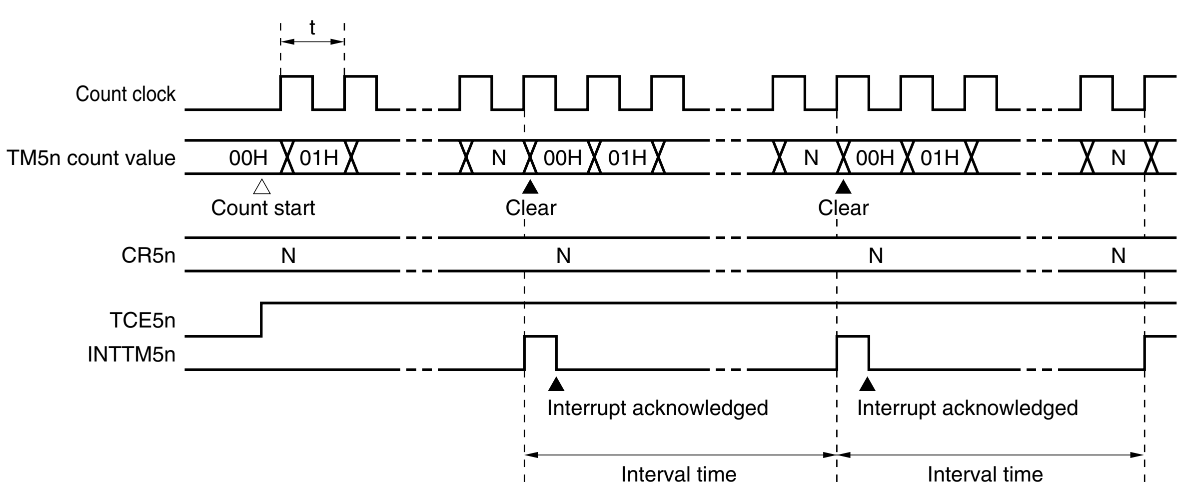
<2> After TCE5n = 1 is set, the count operation starts.
<3> If the values of TM5n and CR5n match, INTTM5n is generated (TM5n is cleared to 00H).
<4> INTTM5n is generated repeatedly at the same interval.
Set TCE5n to 0 to stop the count operation.

Caution Do not write other values to CR5n during operation.

Remarks 1. For how to enable the INTTM5n signal interrupt, see **CHAPTER 21 INTERRUPT FUNCTIONS**.
2. n = 0 to 2

Figure 7-15. Interval Timer Operation Timing (1/2)

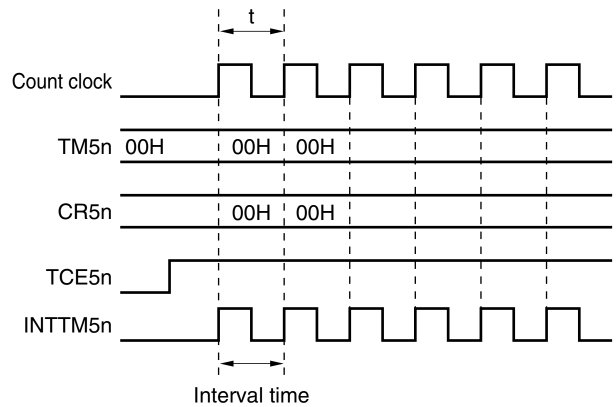
(a) Basic operation



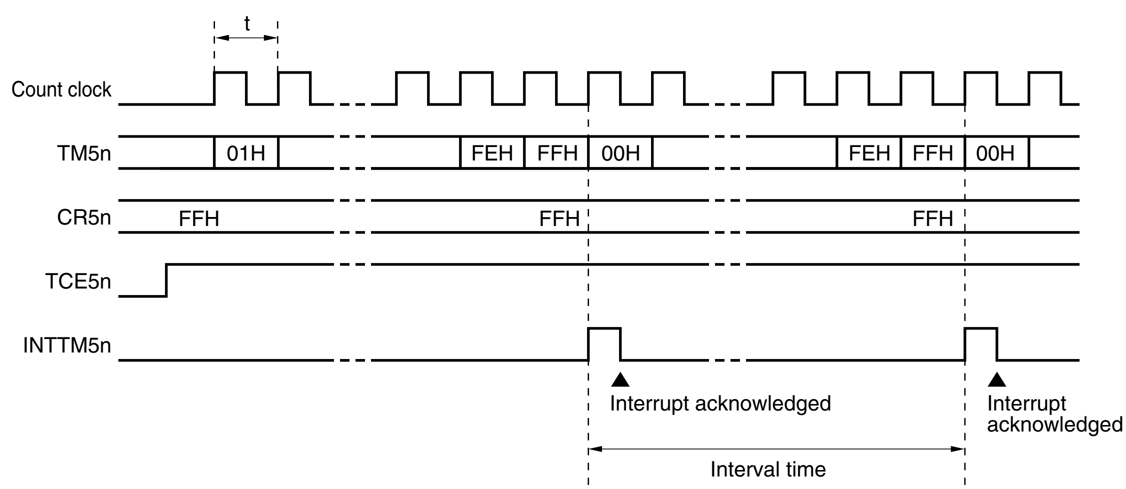
Remark Interval time = (N + 1) × t
N = 01H to FFH
n = 0 to 2

Figure 7-15. Interval Timer Operation Timing (2/2)

(b) When CR5n = 00H



(c) When CR5n = FFH



Remark n = 0 to 2

7.4.2 Operation as external event counter

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

TM5n is incremented each time the valid edge specified by timer clock selection register 5n (TCL5n) is input. Either the rising or falling edge can be selected.

When the TM5n count value matches the value of 8-bit timer compare register 5n (CR5n), TM5n is cleared to 0 and an interrupt request signal (INTTM5n) is generated.

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

Setting

<1> Set each register.

- Set the port mode register (PM44, PM43, or PM34)[Note] to 1.
- TCL5n: Select TI5n pin input edge.
  TI5n pin falling edge → TCL5n = 00H
  TI5n pin rising edge → TCL5n = 01H
- CR5n: Compare value
- TMC5n: Stop the count operation, select the mode in which clear & start occurs on match of TM5n and CR5n, disable the timer F/F inversion operation, disable timer output.
  (TMC5n = 0000××00B × = Don't care)

<2> When TCE5n = 1 is set, the number of pulses input from the TI5n pin is counted.

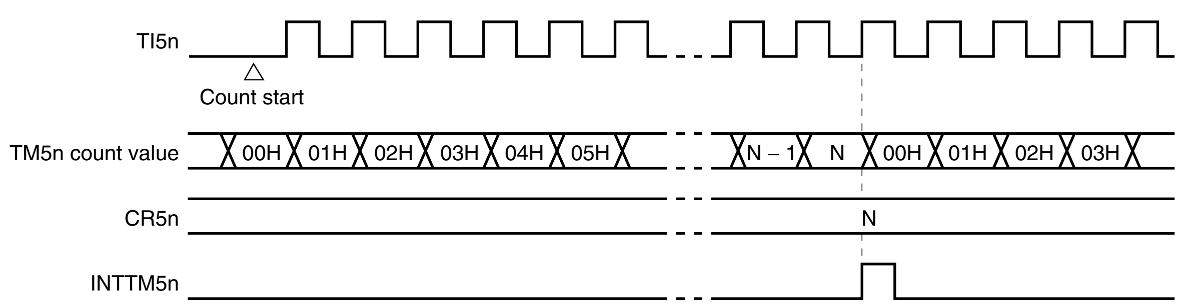
<3> When the values of TM5n and CR5n match, INTTM5n is generated (TM5n is cleared to 00H).

<4> After these settings, INTTM5n is generated each time the values of TM5n and CR5n match.

Note 8-bit timer/event counter 50: PM44
8-bit timer/event counter 51: PM43
8-bit timer/event counter 52: PM34

Remark For how to enable the INTTM5n signal interrupt, see **CHAPTER 21 INTERRUPT FUNCTIONS**.

Figure 7-16. External Event Counter Operation Timing (with Rising Edge Specified)



Remark 1. 8-bit timer/event counter 52 (TM52) can be used as an external 24-bit event counter, by connecting it with 16-bit timer/event counter (TM00) in cascade. Also, input enable of TM52 can be controlled via TMH2. For details, see **6.4.9 External 24-bit event counter operation**.

2. N = 00H to FFH, n = 0 to 2

7.4.3 Square-wave output operation

A square wave with any selected frequency is output at intervals determined by the value preset to 8-bit timer compare register 5n (CR5n).

The TO5n pin output status is inverted at intervals determined by the count value preset to CR5n by setting bit 0 (TOE5n) of 8-bit timer mode control register 5n (TMC5n) to 1. This enables a square wave with any selected frequency to be output (duty = 50%).

Setting

<1> Set each register.

- Clear the port output latch (P44 or P43)[Note] and port mode register (PM44 or PM43)[Note] to 0.
- TCL5n: Select the count clock.
- CR5n: Compare value
- TMC5n: Stop the count operation, select the mode in which clear & start occurs on a match of TM5n and CR5n.

LVS5n	LVR5n	Timer Output F/F Status Setting
1	0	Timer output F/F clear (0) (default value of TO5n output: low level)
0	1	Timer output F/F set (1) (default value of TO5n output: high level)

Timer output enabled
(TMC5n = 00001011B or 00000111B)

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

<3> The timer output F/F is inverted by a match of TM5n and CR5n. After INTTM5n is generated, TM5n is cleared to 00H.

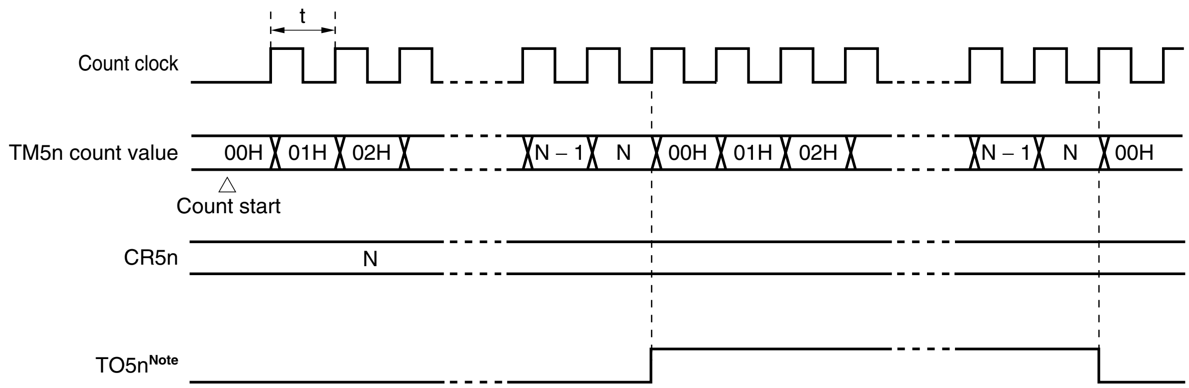
<4> After these settings, the timer output F/F is inverted at the same interval and a square wave is output from TO5n.
The frequency is as follows.
• Frequency = 1/2t (N + 1)
(N: 00H to FFH)

Note 8-bit timer/event counter 50: P44, PM44
8-bit timer/event counter 51: P43, PM43

Caution Do not write other values to CR5n during operation.

Remarks 1. For how to enable the INTTM5n signal interrupt, see **CHAPTER 21 INTERRUPT FUNCTIONS**.
2. n = 0, 1

Figure 7-17. Square-Wave Output Operation Timing



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

7.4.4 PWM output operation

8-bit timer/event counter 5n operates as a PWM output when bit 6 (TMC5n6) of 8-bit timer mode control register 5n (TMC5n) is set to 1.

The duty pulse determined by the value set to 8-bit timer compare register 5n (CR5n) is output from TO5n.

Set the active level width of the PWM pulse to CR5n; the active level can be selected with bit 1 (TMC5n1) of TMC5n.

The count clock can be selected with bits 0 to 2 (TCL5n0 to TCL5n2) of timer clock selection register 5n (TCL5n).

PWM output can be enabled/disabled with bit 0 (TOE5n) of TMC5n.

Caution In PWM mode, make the CR5n rewrite period 3 count clocks of the count clock (clock selected by TCL5n) or more.

Remark n = 0, 1

(1) PWM output basic operation

Setting

<1> Set each register.

- Clear the port output latch (P44 or P43)[Note] and port mode register (PM44 or PM43)[Note] to 0.
- TCL5n: Select the count clock.
- CR5n: Compare value
- TMC5n: Stop the count operation, select PWM mode.
  The timer output F/F is not changed.

TMC5n1	Active Level Selection
0	Active-high
1	Active-low

Timer output enabled
(TMC5n = 01000001B or 01000011B)

<2> The count operation starts when TCE5n = 1.
Clear TCE5n to 0 to stop the count operation.

Note 8-bit timer/event counter 50: P44, PM43
8-bit timer/event counter 51: P43, PM43

PWM output operation

<1> PWM output (output from TO5n) outputs an inactive level until an overflow occurs.

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

<3> After the CR5n matches the count value, the inactive level is output until an overflow occurs again.

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

<5> When the count operation is stopped with TCE5n = 0, PWM output becomes inactive.
For details of timing, see **Figures 7-18** and **7-19**.
The cycle, active-level width, and duty are as follows.

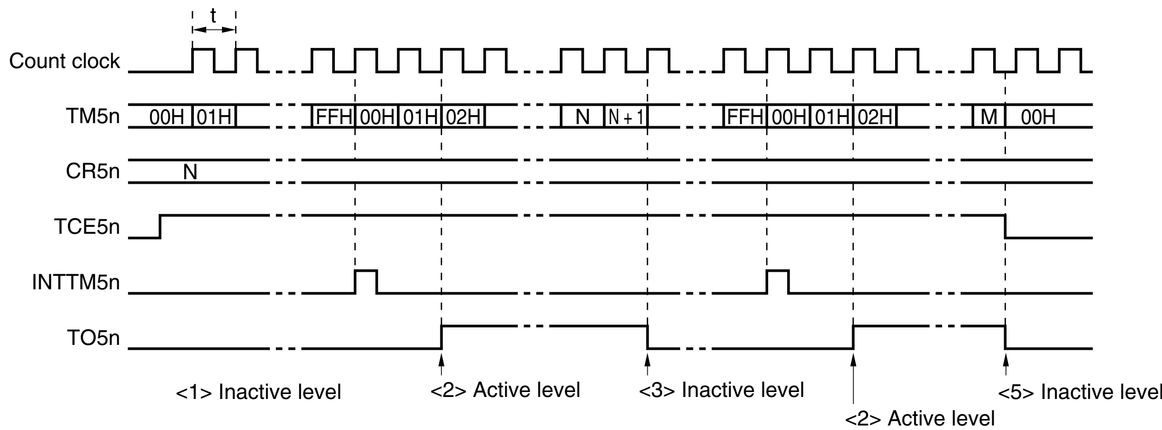
- Cycle = 2^8t
- Active-level width = Nt
- Duty = $N/2^8$

(N = 00H to FFH)

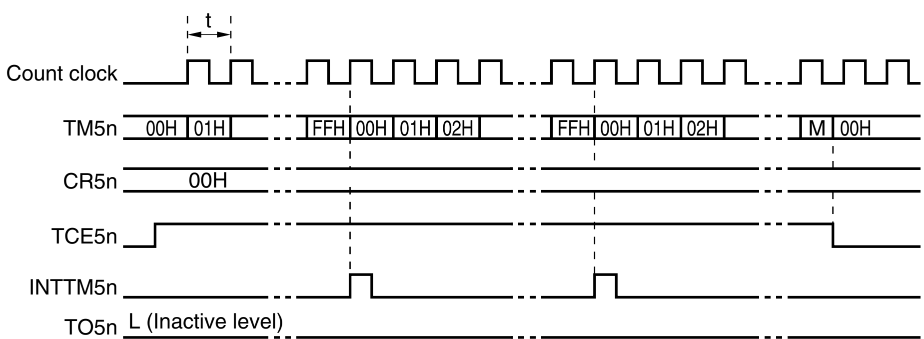
Remark n = 0, 1

Figure 7-18. PWM Output Operation Timing

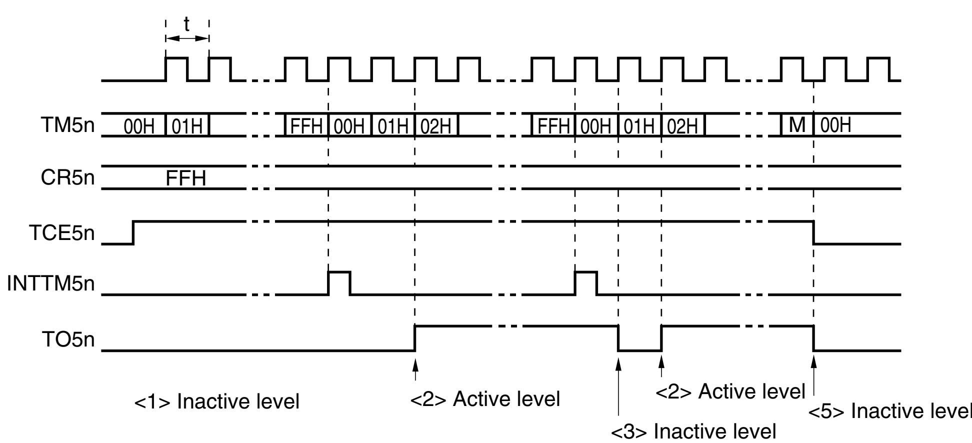
(a) Basic operation (active level = H)



(b) CR5n = 00H



(c) CR5n = FFH



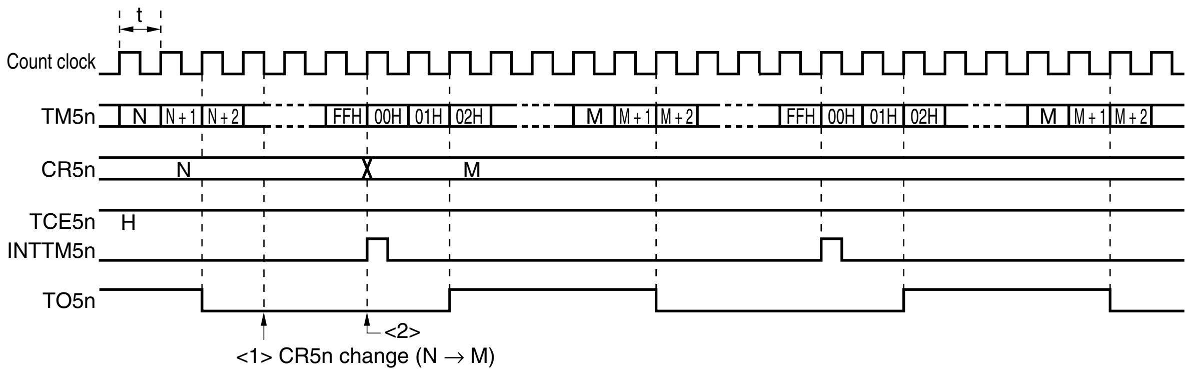
Remarks 1. <1> to <3> and <5> in Figure 7-18 (a) correspond to <1> to <3> and <5> in PWM output operation in **7.4.4 (1) PWM output basic operation**.

2. n = 0, 1

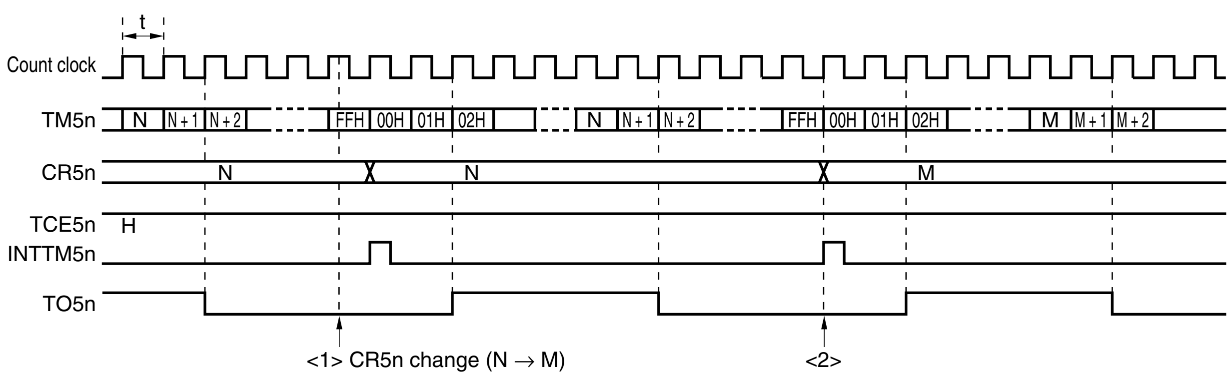
(2) Operation with CR5n changed

Figure 7-19. Timing of Operation with CR5n Changed

(a) CR5n value is changed from N to M before clock rising edge of FFH
→ Value is transferred to CR5n at overflow immediately after change.



(b) CR5n value is changed from N to M after clock rising edge of FFH
→ Value is transferred to CR5n at second overflow.



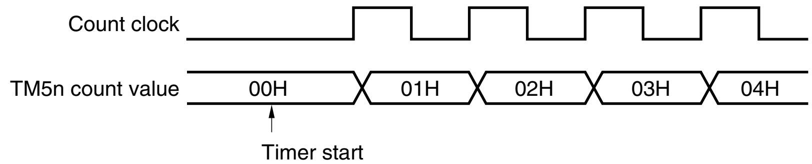
Caution When reading from CR5n between <1> and <2> in Figure 7-19, the value read differs from the actual value (read value: M, actual value of CR5n: N).

7.5 Cautions for 8-Bit Timer/Event Counters 50, 51, and 52

(1) Timer start error

An error of up to one clock may occur in the time required for a match signal to be generated after timer start. This is because 8-bit timer counters 50, 51, and 52 (TM50, TM51, and TM52) are started asynchronously to the count clock.

Figure 7-20. 8-Bit Timer Counter 5n Start Timing



Remark n = 0 to 2

<R> **(2) Cautions for 16-bit timer/event counter 00 count up during external 24-bit event counter operation**

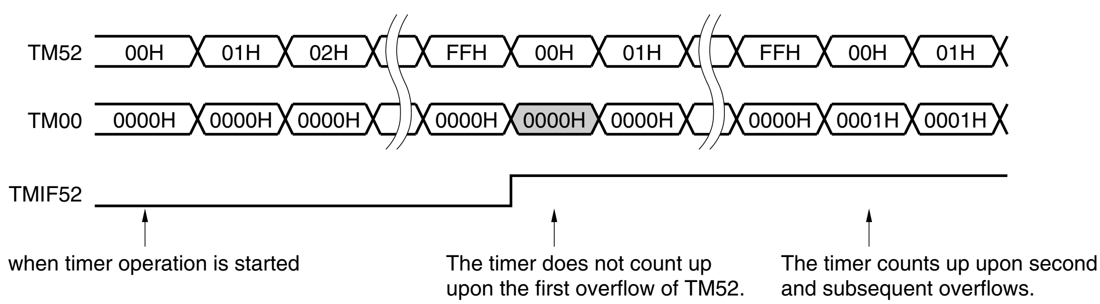
16-bit timer/event counter 00 has an internal synchronization circuit to eliminate noise when starting operation, and the first clock immediately after operation start is not counted.

When using the counter as a 24-bit counter, by setting 16-bit timer/event counter 00 and 8-bit timer/event counter 52 as the higher and lower timer and connecting them in cascade, the interrupt request flag of 8-bit timer/event counter 52 which is the lower timer must be checked as described below, in order to accurately read the 24-bit count values.

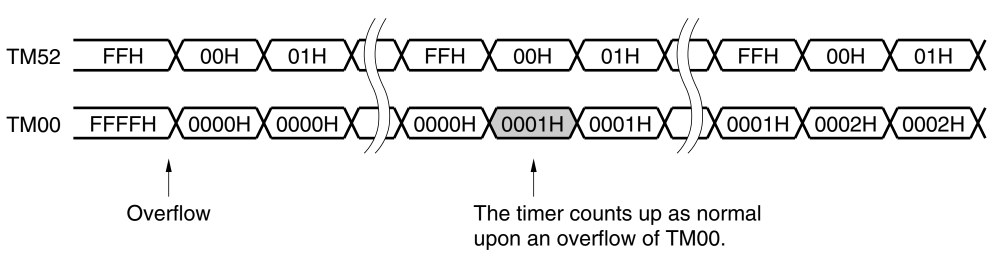
- If TMIF52 = 1 when TM52 and TM00 are read:
  The actual TM00 count value is "read value of TM00 + 1".
- If TMIF52 = 0 when TM52 and TM00 are read:
  The read value is the correct value.

This phenomenon of 16-bit timer/event counter 00 occurs only when operation is started. A count delay will not occur when 16-bit timer/event counter 00 overflows and the count is restarted from 0000H, since synchronization has already been implemented.

<When starting operation>



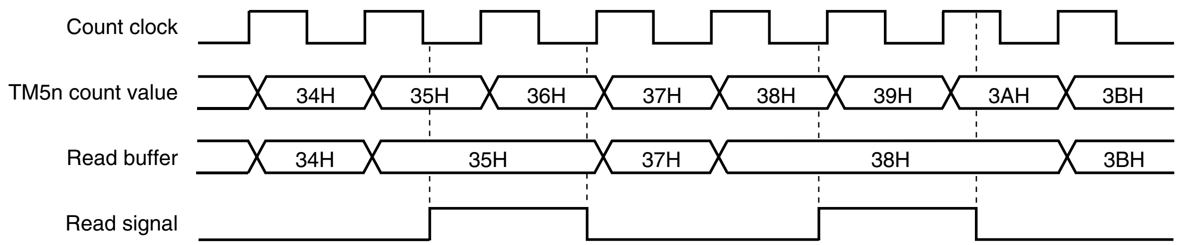
<Overflow of higher timer>



<R> **(3) Reading of 8-bit timer counter 5n (TM5n)**

TM5n can be read without stopping the actual counter, because the count values captured to the buffer are fixed when it is read. The buffer, however, may not be updated when it is read immediately before the counter counts up, because the buffer is updated at the timing the counter counts up.

Figure 7-21. 8-bit Timer Counter 5n (TM5n) Read Timing



Remark n = 0 to 2

8.1 Functions of 8-Bit Timers H0, H1, and H2

8-bit timers H0, H1, and H2 have the following functions.

- Interval timer
- Square-wave output[Note 1]
- PWM output[Note 2]
- Carrier generator (8-bit timer H1 only)[Note 3]

Notes 1. TMH0 and TMH1 only.
2. However, TOH0 and TOH1 only for TOHn
3. TMH1 only. TM51 and TMH1 can be used in combination as a carrier generator mode.

8.2 Configuration of 8-Bit Timers H0, H1, and H2

8-bit timers H0, H1, and H2 include the following hardware.

Table 8-1. Configuration of 8-Bit Timers H0, H1, and H2

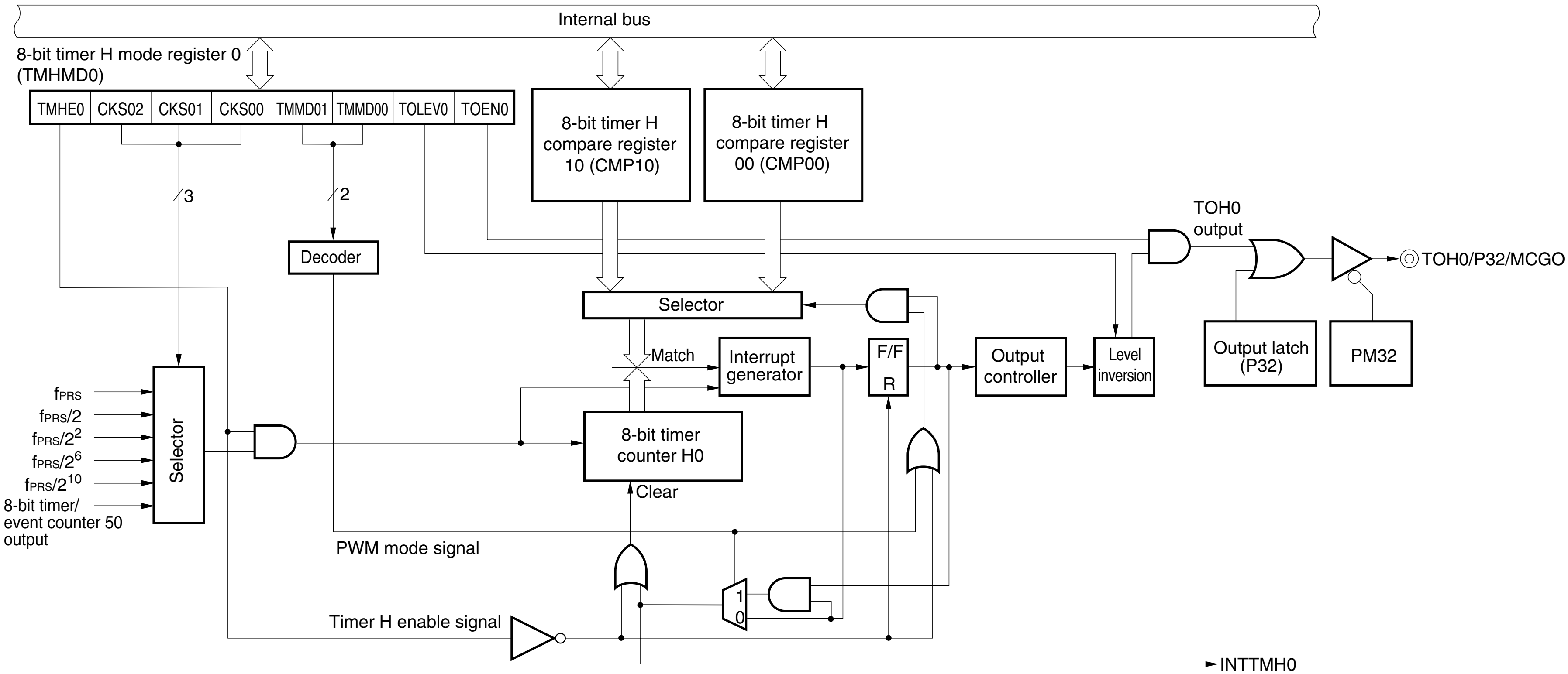
Item	Configuration
Timer register	8-bit timer counter Hn
Registers	8-bit timer H compare register 0n (CMP0n) 8-bit timer H compare register 1n (CMP1n)
Timer output	TOHn[Note 1], output controller
Control registers	8-bit timer H mode register n (TMHMDn) 8-bit timer H carrier control register 1 (TMCYC1)[Note 2] Port mode register 3 (PM3) Port register 3 (P3)

Notes 1. TMH2 does not have an output pin (TOH2). It can only be used as an internal interrupt (INTTMH2) or an external event input enable signal for the TI52 pin.
2. 8-bit timer H1 only

Remark n = 0-2, however, TOH0 and TOH1 only for TOHn

Figures 8-1 and 8-3 show the block diagrams.

Figure 8-1. Block Diagram of 8-Bit Timer H0



<R>
Figure 8-2. Block Diagram of 8-Bit Timer H1

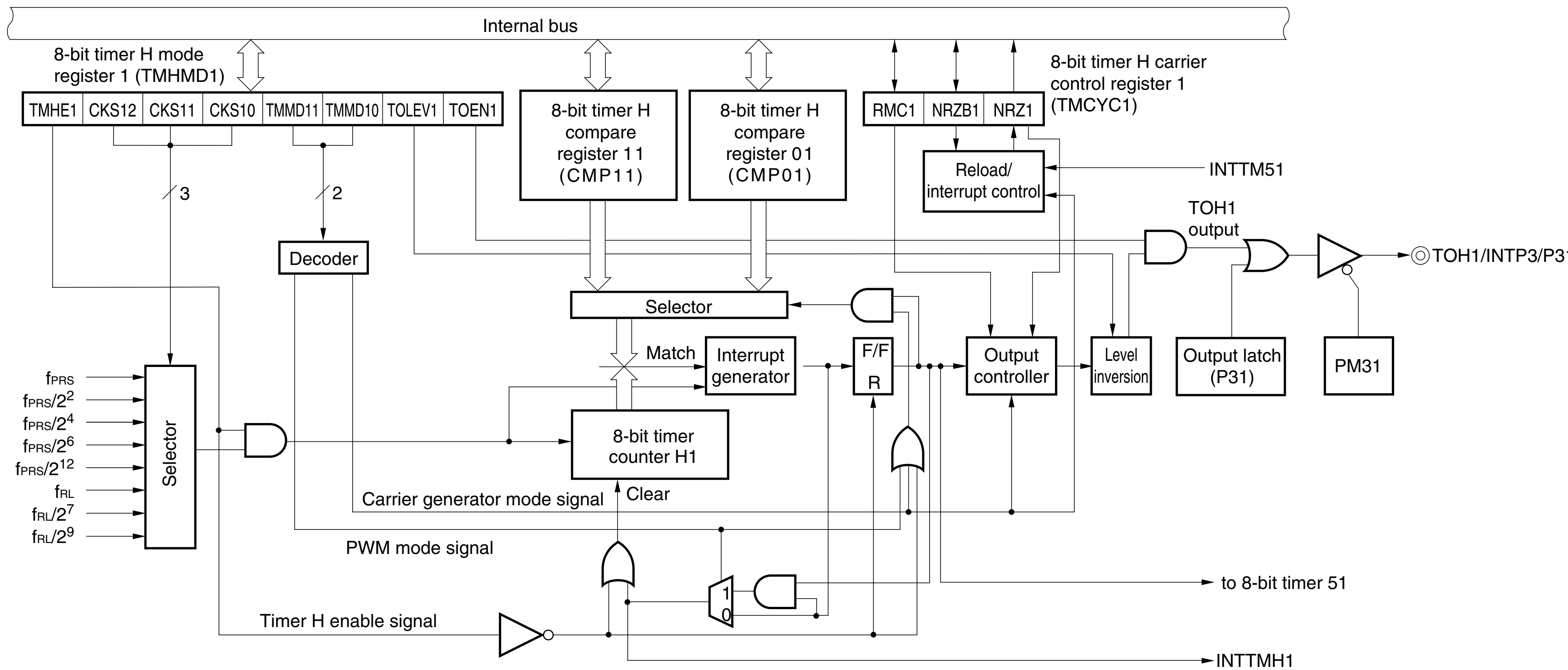
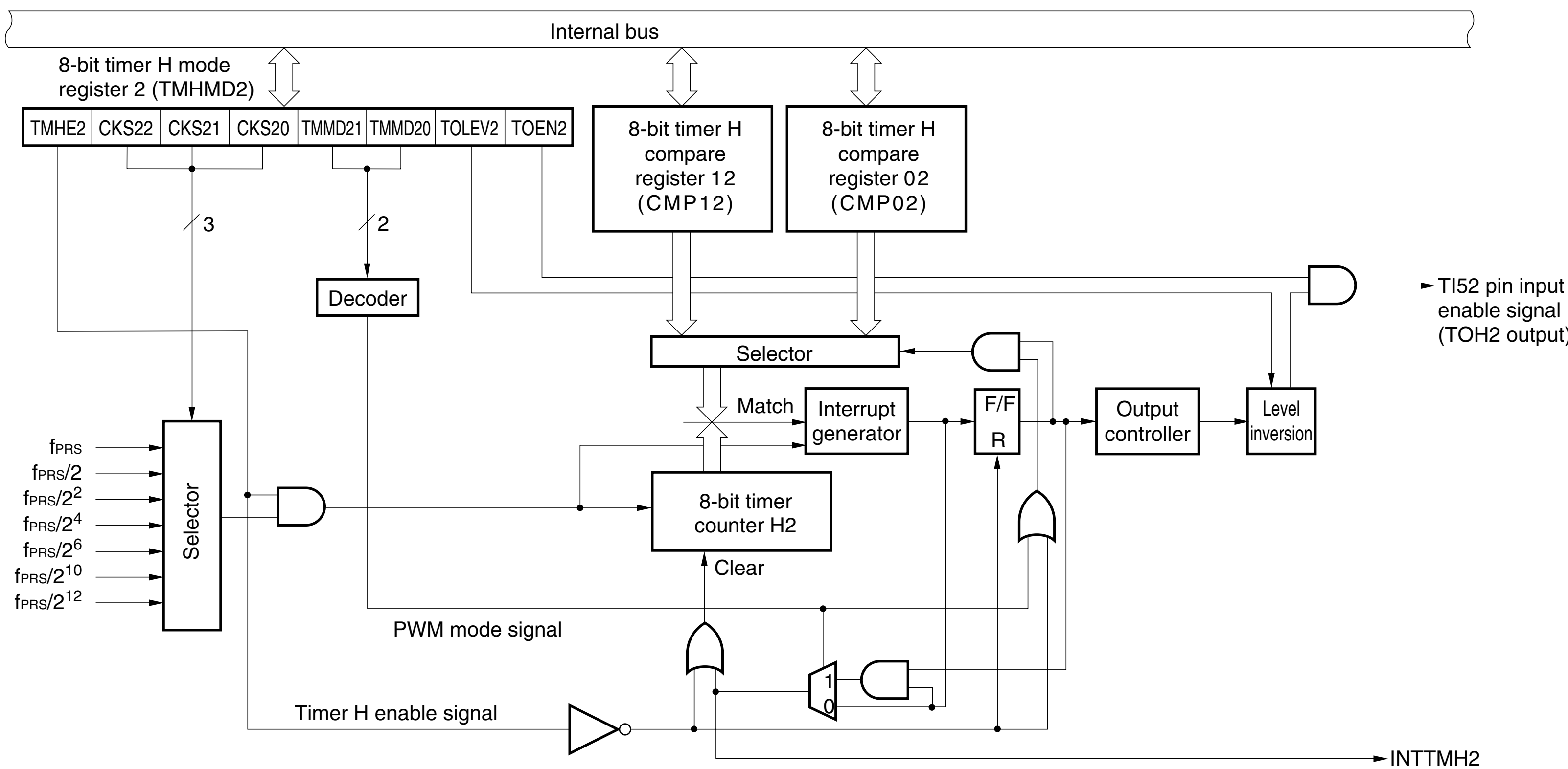


Figure 8-3. Block Diagram of 8-Bit Timer H2



(1) 8-bit timer H compare register 0n (CMP0n)

This register can be read or written by an 8-bit memory manipulation instruction. This register is used in all of the timer operation modes.

This register constantly compares the value set to CMP0n with the count value of the 8-bit timer counter Hn and, when the two values match, generates an interrupt request signal (INTTMHn) and inverts the output level of TOHn.

Rewrite the value of CMP0n while the timer is stopped (TMHEn = 0).

A reset signal generation sets this register to 00H.

Figure 8-4. Format of 8-Bit Timer H Compare Register 0n (CMP0n)

Address: FF18H (CMP00), FF1AH (CMP01), FF44H (CMP02) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMP0n (n = 0 to 2)								

Caution CMP0n cannot be rewritten during timer count operation. CMP0n can be refreshed (the same value is written) during timer count operation.

(2) 8-bit timer H compare register 1n (CMP1n)

This register can be read or written by an 8-bit memory manipulation instruction. This register is used in the PWM output mode and carrier generator mode.

In the PWM output mode, this register constantly compares the value set to CMP1n with the count value of the 8-bit timer counter Hn and, when the two values match, inverts the output level of TOHn. No interrupt request signal is generated.

In the carrier generator mode, the CMP1n register always compares the value set to CMP1n with the count value of the 8-bit timer counter Hn and, when the two values match, generates an interrupt request signal (INTTMHn). At the same time, the count value is cleared.

CMP1n can be rewritten during timer count operation.

If the value of CMP1n is rewritten while the timer is operating, the new value is latched and transferred to CMP1n when the count value of the timer matches the old value of CMP1n, and then the value of CMP1n is changed to the new value. If matching of the count value and the CMP1n value and writing a value to CMP1n conflict, the value of CMP1n is not changed.

A reset signal generation sets this register to 00H.

Figure 8-5. Format of 8-Bit Timer H Compare Register 1n (CMP1n)

Address: FF19H (CMP10), FF1BH (CMP11), FF45H (CMP12) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMP1n (n = 0 to 2)								

Caution In the PWM output mode and carrier generator mode, be sure to set CMP1n when starting the timer count operation (TMHEn = 1) after the timer count operation was stopped (TMHEn = 0) (be sure to set again even if setting the same value to CMP1n).

Remark n = 0 to 2, however, TOH0 and TOH1 only for TOHn

8.3 Registers Controlling 8-Bit Timers H0, H1, and H2

The following four registers are used to control 8-bit timers H0, H1, and H2.

- 8-bit timer H mode register n (TMHMDn)
- 8-bit timer H carrier control register 1 (TMCYC1)[Note]
- Port mode register 3 (PM3)
- Port register 3 (P3)

Note 8-bit timer H1 only

(1) 8-bit timer H mode register n (TMHMDn)

This register controls the mode of timer H.
This register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to 00H.

Remark n = 0 to 2

Figure 8-6. Format of 8-Bit Timer H Mode Register 0 (TMHMD0)

Address: FF69H After reset: 00H R/W

	<7>	6	5	4	3	2	<1>	<0>
TMHMD0	TMHE0	CKS02	CKS01	CKS00	TMMD01	TMMD00	TOLEV0	TOEN0

TMHE0	Timer operation enable
0	Stops timer count operation (counter is cleared to 0)
1	Enables timer count operation (count operation started by inputting clock)

CKS02	CKS01	CKS00	Count clock selection[Note 1]			
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz
0	0	0	f_{PRS}[Note 2]	2 MHz	5 MHz	10 MHz
0	0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz
0	1	0	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz
0	1	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz
1	0	0	$f_{PRS}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz
1	0	1	TM50 output[Note 3]			
Other than above			Setting prohibited			

TMMD01	TMMD00	Timer operation mode
0	0	Interval timer mode
1	0	PWM mode
Other than above		Setting prohibited

<R>

TOLEV0	Timer output level control (in default mode)
0	Low level
1	High level

TOEN0	Timer output control
0	Disables output
1	Enables output

Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{XH}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 2.7 to 5.5 V: $f_{PRS} \leq 10$ MHz
- V_{DD} = 1.8 to 2.7 V: $f_{PRS} \leq 5$ MHz

2. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{RH}) (XSEL = 0), when 1.8 V $\leq V_{DD} <$ 2.7 V, the setting of CKS02 = CKS01 = CKS00 = 0 (count clock: f_{PRS}) is prohibited.

Notes 3. Note the following points when selecting the TM50 output as the count clock.

- Mode in which the count clock is cleared and started upon a match of TM50 and CR50 (TMC506 = 0)
  Start the operation of the 8-bit timer/event counter 50 first and then enable the timer F/F inversion operation (TMC501 = 1).
- PWM mode (TMC506 = 1)
  Start the operation of the 8-bit timer/event counter 50 first and then set the count clock to make the duty = 50%.

It is not necessary to enable (TOE50 = 1) TO50 output in any mode.

Cautions 1. When TMHE0 = 1, setting the other bits of TMHMD0 is prohibited. However, TMHMD0 can be refreshed (the same value is written).

2. In the PWM output mode, be sure to set the 8-bit timer H compare register 10 (CMP10) when starting the timer count operation (TMHE0 = 1) after the timer count operation was stopped (TMHE0 = 0) (be sure to set again even if setting the same value to CMP10).

3. The actual TOH0/P32/MCGO pin output is determined depending on PM32 and P32, besides TOH0 output.

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. TMC506: Bit 6 of 8-bit timer mode control register 50 (TMC50)
TMC501: Bit 1 of TMC50

Figure 8-7. Format of 8-Bit Timer H Mode Register 1 (TMHMD1)

Address: FF6CH After reset: 00H R/W

	<7>	6	5	4	3	2	<1>	<0>
TMHMD1	TMHE1	CKS12	CKS11	CKS10	TMMD11	TMMD10	TOLEV1	TOEN1

TMHE1	Timer operation enable
0	Stops timer count operation (counter is cleared to 0)
1	Enables timer count operation (count operation started by inputting clock)

CKS12	CKS11	CKS10	Count clock selection[Note 1]			
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz
0	0	0	f_{PRS}[Note 2]	2 MHz	5 MHz	10 MHz
0	0	1	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz
0	1	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz
0	1	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz
1	0	0	$f_{PRS}/2^{12}$	0.49 kHz	1.22 kHz	2.44 kHz
1	0	1	$f_{RL}/2^7$	1.88 kHz (TYP.)		
1	1	0	$f_{RL}/2^9$	0.47 kHz (TYP.)		
1	1	1	f_{RL}	240 kHz (TYP.)		

TMMD11	TMMD10	Timer operation mode
0	0	Interval timer mode
0	1	Carrier generator mode
1	0	PWM output mode
1	1	Setting prohibited

TOLEV1	Timer output level control (in default mode)
0	Low level
1	High level

TOEN1	Timer output control
0	Disables output
1	Enables output

Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{XH}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 2.7 to 5.5 V: $f_{PRS} \leq$ 10 MHz
- V_{DD} = 1.8 to 2.7 V: $f_{PRS} \leq$ 5 MHz

2. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{RH}) (XSEL = 0), when 1.8 V $\leq V_{DD} <$ 2.7 V, the setting of CKS12 = CKS11 = CKS10 = 0 (count clock: f_{PRS}) is prohibited.

Cautions 1. When TMHE1 = 1, setting the other bits of TMHMD1 is prohibited. However, TMHMD1 can be refreshed (the same value is written).

2. In the PWM output mode and carrier generator mode, be sure to set the 8-bit timer H compare register 11 (CMP11) when starting the timer count operation (TMHE1 = 1) after the timer count operation was stopped (TMHE1 = 0) (be sure to set again even if setting the same value to CMP11).

3. When the carrier generator mode is used, set so that the count clock frequency of TMH1 becomes more than 6 times the count clock frequency of TM51.

4. The actual TOH1/P31/INTP3 pin output is determined depending on PM31 and P31, besides TOH1 output.

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. f_{RL}: Internal low-speed oscillation clock frequency

Figure 8-8. Format of 8-Bit Timer H Mode Register 2 (TMHMD2)

Address: FF42H After reset: 00H R/W

	<7>	6	5	4	3	2	<1>	<0>
TMHMD2	TMHE2	CKS22	CKS21	CKS20	TMMD21	TMMD20	TOLEV2	TOEN2

TMHE2	Timer operation enable
0	Stops timer count operation (counter is cleared to 0)
1	Enables timer count operation (count operation started by inputting clock)

CKS22	CKS21	CKS20	Count clock selection[Note 1]			
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz
0	0	0	f_{PRS}[Note 2]	2 MHz	5 MHz	10 MHz
0	0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz
0	1	0	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz
0	1	1	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz
1	0	0	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz
1	0	1	$f_{PRS}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz
1	1	0	$f_{PRS}/2^{12}$	0.49 kHz	1.22 kHz	2.44 kHz
Other than above			Setting prohibited			

TMMD21	TMMD20	Timer operation mode
0	0	Interval timer mode
1	0	Input enable width adjust mode for pins (PWM mode)
Other than above		Setting prohibited

TOLEV2	Timer output level control (in default mode)
0	Low level
1	High level

TOEN2	Timer output control
0	Disables output
1	Enables output[Note 3]

Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{XH}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.
- V_{DD} = 2.7 to 5.5 V: $f_{PRS} \leq 10$ MHz
- V_{DD} = 1.8 to 2.7 V: $f_{PRS} \leq 5$ MHz

2. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{RH}) (XSEL = 0), when 1.8 V ≤ V_{DD} < 2.7 V, the setting of CKS22 = CKS21 = CKS20 = 0 (count clock: f_{PRS}) is prohibited.

3. The timer output of TMH2 can only be used as an external event input enable signal of TM52. No pins for external output are available.

Caution When TMHE2 = 1, setting the other bits of TMHMD2 is prohibited.

Remark f_{PRS}: Peripheral hardware clock frequency

(2) 8-bit timer H carrier control register 1 (TMCYC1)

This register controls the remote control output and carrier pulse output status of 8-bit timer H1.

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

Reset signal generation sets this register to 00H.

Figure 8-9. Format of 8-Bit Timer H Carrier Control Register 1 (TMCYC1)

Address: FF6DH After reset: 00H R/W[Note]

Symbol	7	6	5	4	3	2	1	<0>
TMCYC1	0	0	0	0	0	RMC1	NRZB1	NRZ1

RMC1	NRZB1	Remote control output
0	0	Low-level output
0	1	High-level output at rising edge of INTTM51 signal input
1	0	Low-level output
1	1	Carrier pulse output at rising edge of INTTM51 signal input

NRZ1	Carrier pulse output status flag
0	Carrier output disabled status (low-level status)
1	Carrier output enabled status (RMC1 = 1: Carrier pulse output, RMC1 = 0: High-level status)

Note Bit 0 is read-only.

Caution Do not rewrite RMC1 when TMHE1 = 1. However, TMCYC1 can be refreshed (the same value is written).

(3) Port mode register 3 (PM3)

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

When using the P32/TOH0/MCGO and P31/TOH1/INTP3 pins for timer output, clear PM32 and PM31 and the output latches of P32 and P31 to 0.

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

Reset signal generation sets this register to FFH.

Figure 8-10. Format of Port Mode Register 3 (PM3)

Address: FF23H After reset: FFH R/W

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

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

8.4 Operation of 8-Bit Timers H0, H1 and H2

8.4.1 Operation as interval timer/square-wave output

When the 8-bit timer counter Hn and compare register 0n (CMP0n) match, an interrupt request signal (INTTMHn) is generated and the 8-bit timer counter Hn is cleared to 00H.

Compare register 1n (CMP1n) is not used in interval timer mode. Since a match of the 8-bit timer counter Hn and the CMP1n register is not detected even if the CMP1n register is set, timer output is not affected.

By setting bit 0 (TOENn) of timer H mode register n (TMHMDn) to 1, a square wave of any frequency (duty = 50%) is output from TOHn.

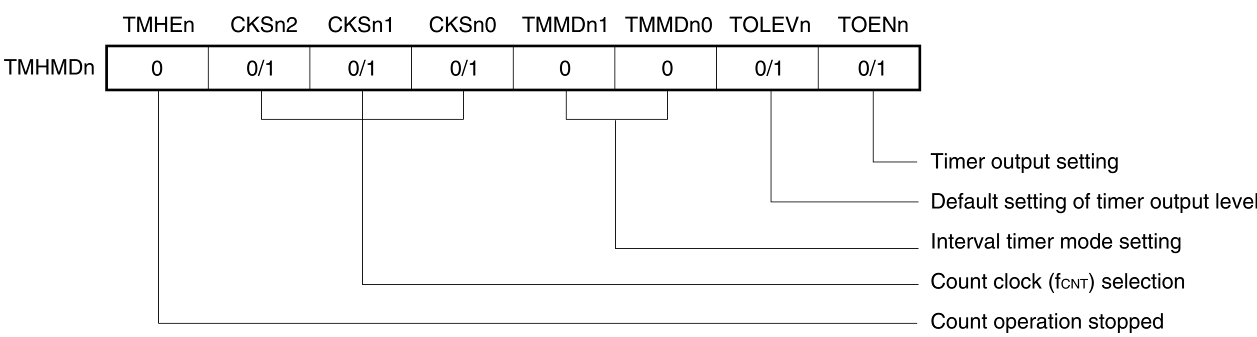
The timer output of TMH2 can only be used as an external event input enable signal of TM52. Note, no pins for external output are available.

Setting

<1> Set each register.

Figure 8-11. Register Setting During Interval Timer/Square-Wave Output Operation

(i) Setting timer H mode register n (TMHMDn)



(ii) CMP0n register setting

The interval time is as follows if N is set as a comparison value.

- Interval time = (N +1)/f_{CNT}

<2> Count operation starts when TMHEn = 1.

<3> When the values of the 8-bit timer counter Hn and the CMP0n register match, the INTTMHn signal is generated and the 8-bit timer counter Hn is cleared to 00H.

<4> Subsequently, the INTTMHn signal is generated at the same interval. To stop the count operation, clear TMHEn to 0.

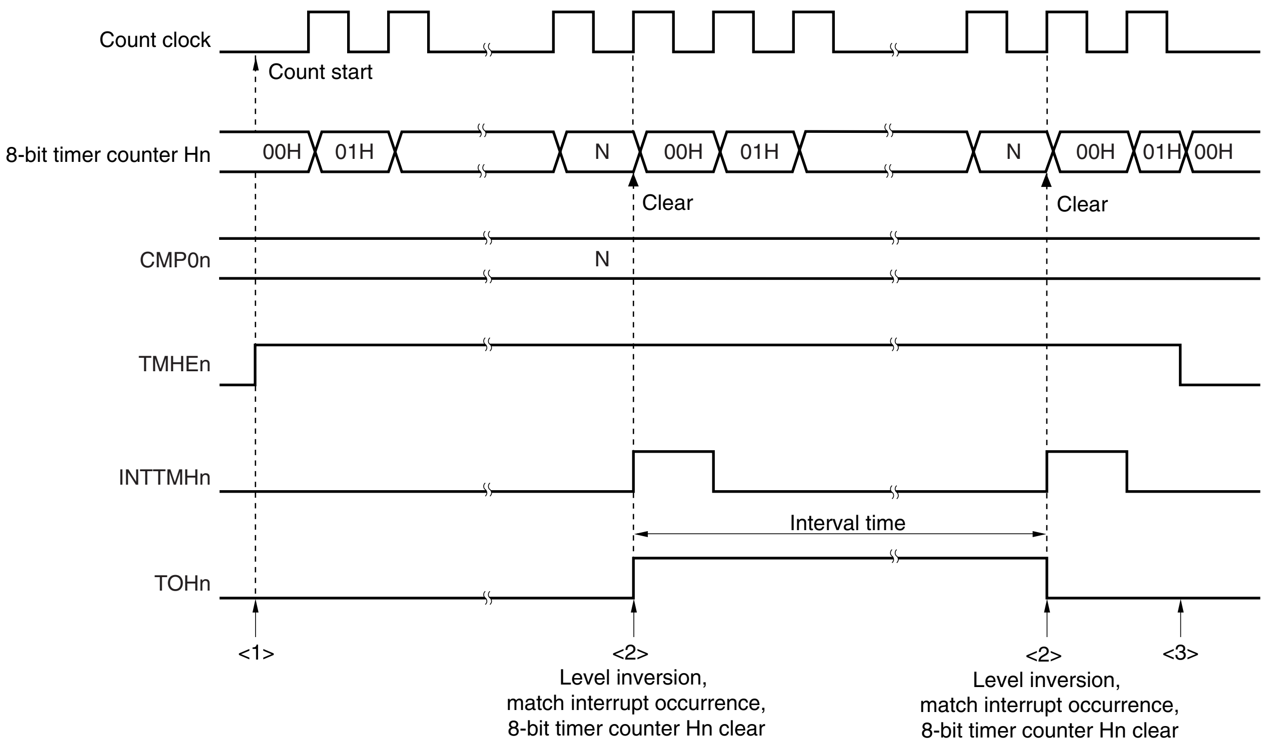
Remarks 1. For the setting of the output pin, see **8.3 (3) Port mode register 3 (PM3)**.

2. For how to enable the INTTMHn signal interrupt, see **CHAPTER 21 INTERRUPT FUNCTIONS**.

3. n = 0 to 2, however, TOH0 and TOH1 only for TOHn

Figure 8-12. Timing of Interval Timer/Square-Wave Output Operation (1/2)

(a) Basic operation (Operation When 01H ≤ CMP0n ≤ FEH)



<1> The count operation is enabled by setting the TMHEn bit to 1. The count clock starts counting no more than 1 clock after the operation is enabled.

<2> When the value of the 8-bit timer counter Hn matches the value of the CMP0n register, the value of the timer counter is cleared, and the level of the TOHn output is inverted. In addition, the INTTMHn signal is output at the rising edge of the count clock.

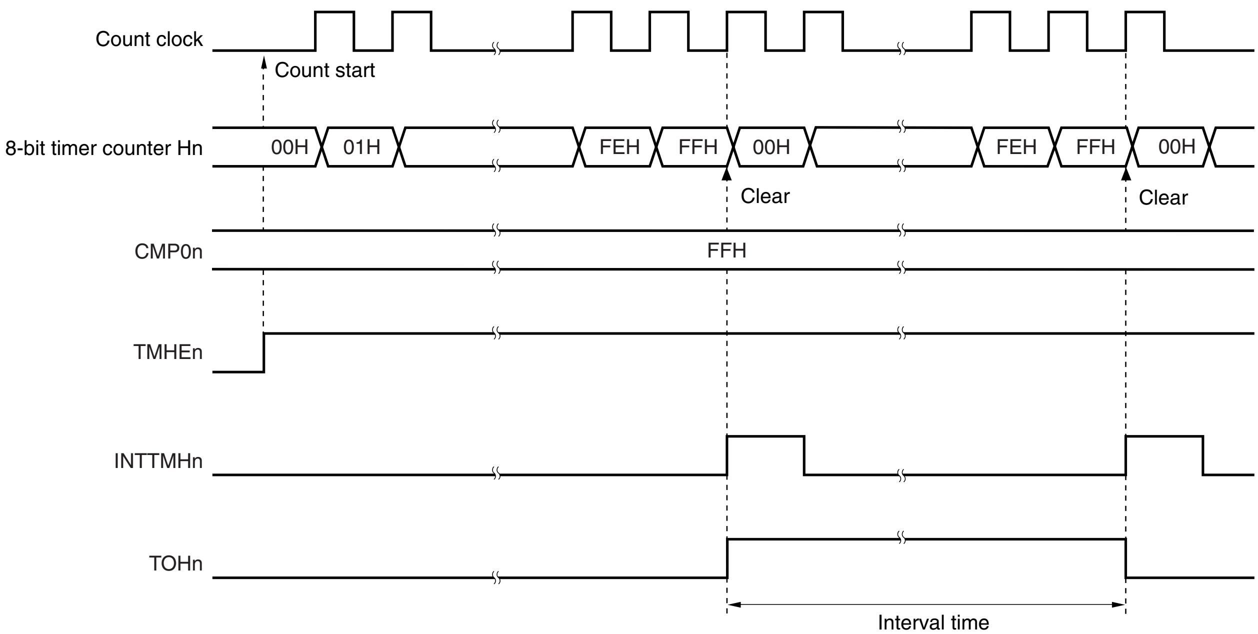
<3> If the TMHEn bit is cleared to 0 while timer H is operating, the INTTMHn signal and TOHn output are set to the default level. If they are already at the default level before the TMHEn bit is cleared to 0, then that level is maintained.

Remarks 1. n = 0 to 2, however, TOH0 and TOH1 only for TOHn

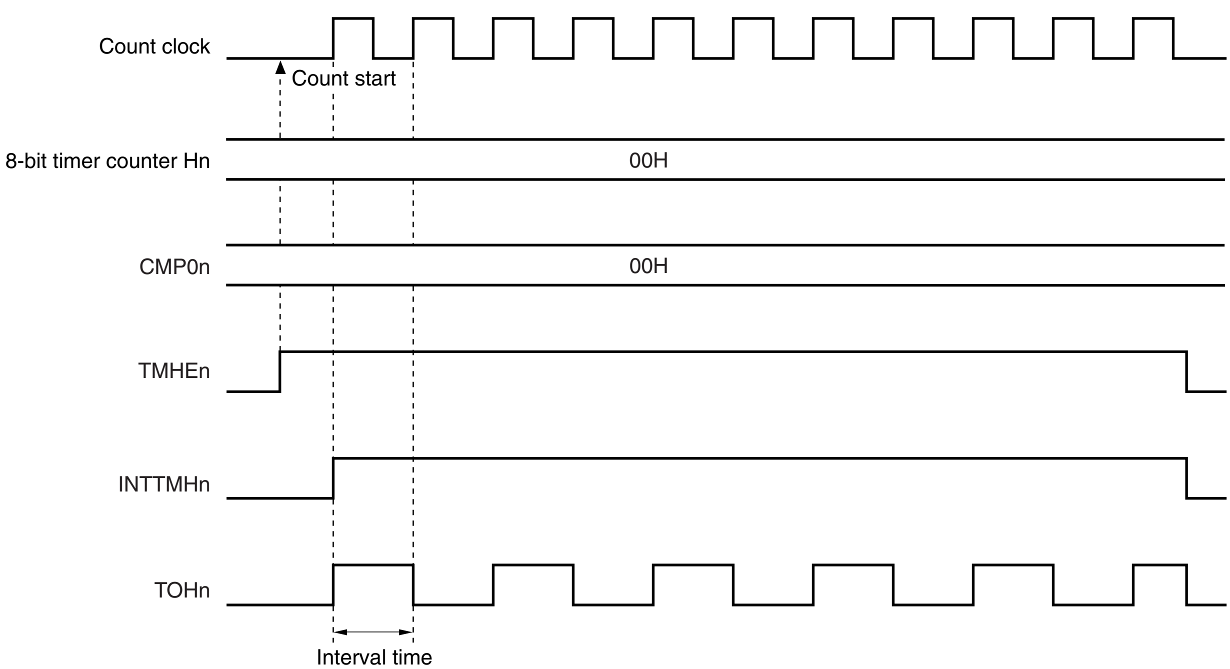
2. 01H ≤ N ≤ FEH

Figure 8-12. Timing of Interval Timer/Square-Wave Output Operation (2/2)

(b) Operation when CMP0n = FFH



(c) Operation when CMP0n = 00H



Remark n = 0 to 2, however, TOH0 and TOH1 only for TOHn

8.4.2 Operation as PWM output

In PWM output mode, a pulse with an arbitrary duty and arbitrary cycle can be output.

The 8-bit timer compare register 0n (CMP0n) controls the cycle of timer output (TOHn). Rewriting the CMP0n register during timer operation is prohibited.

The 8-bit timer compare register 1n (CMP1n) controls the duty of timer output (TOHn). Rewriting the CMP1n register during timer operation is possible.

The operation in PWM output mode is as follows.

PWM output (TOHn output) outputs an active level and 8-bit timer counter Hn is cleared to 0 when 8-bit timer counter Hn and the CMP0n register match after the timer count is started. PWM output (TOHn output) outputs an inactive level when 8-bit timer counter Hn and the CMP1n register match.

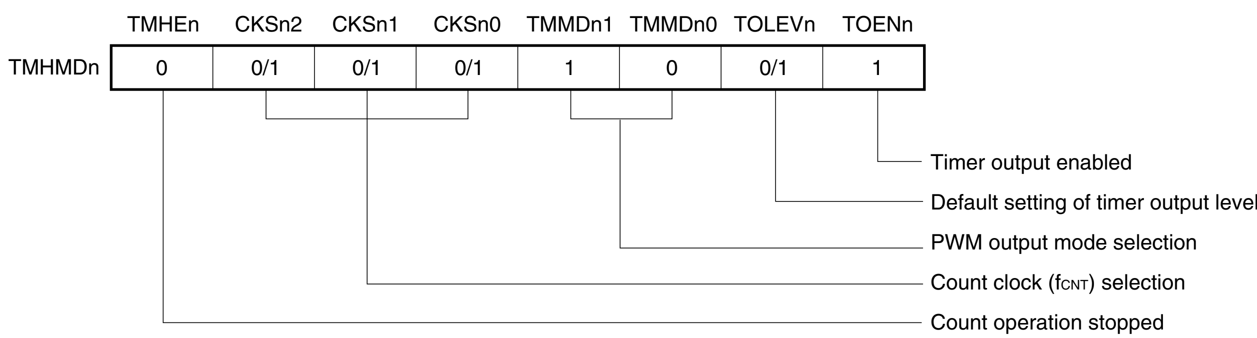
The timer output of TMH2 (PWM output) can only be used as an external event input enable signal of TM52. Note, no pins for external output are available.

Setting

<1> Set each register.

Figure 8-13. Register Setting in PWM Output Mode

(i) Setting timer H mode register n (TMHMDn)



(ii) Setting CMP0n register

- Compare value (N): Cycle setting

(iii) Setting CMP1n register

- Compare value (M): Duty setting

Remarks 1. n = 0 to 2, however, TOH0 and TOH1 only for TOHn

2. 00H ≤ CMP1n (M) < CMP0n (N) ≤ FFH

<2> The count operation starts when TMHEn = 1.

<3> The CMP0n register is the compare register that is to be compared first after counter operation is enabled. When the values of the 8-bit timer counter Hn and the CMP0n register match, the 8-bit timer counter Hn is cleared, an interrupt request signal (INTTMHn) is generated, an active level is output. At the same time, the compare register to be compared with the 8-bit timer counter Hn is changed from the CMP0n register to the CMP1n register.

<4> When the 8-bit timer counter Hn and the CMP1n register match, an inactive level is output and the compare register to be compared with 8-bit timer counter Hn is changed from the CMP1n register to the CMP0n register. At this time, 8-bit timer counter Hn is not cleared and the INTTMHn signal is not generated.

<5> By performing procedures <3> and <4> repeatedly, a pulse with an arbitrary duty can be obtained.

<6> To stop the count operation, set TMHEn = 0.

If the setting value of the CMP0n register is N, the setting value of the CMP1n register is M, and the count clock frequency is f_{CNT}, the PWM pulse output cycle and duty are as follows.

- PWM pulse output cycle = (N + 1)/f_{CNT}
- Duty = (M + 1)/(N + 1)

Cautions 1. The set value of the CMP1n register can be changed while the timer counter is operating. However, this takes a duration of three operating clocks (signal selected by the CKSn2 to CKSn0 bits of the TMHMDn register) from when the value of the CMP1n register is changed until the value is transferred to the register.

2. Be sure to set the CMP1n register when starting the timer count operation (TMHEn = 1) after the timer count operation was stopped (TMHEn = 0) (be sure to set again even if setting the same value to the CMP1n register).

3. Make sure that the CMP1n register setting value (M) and CMP0n register setting value (N) are within the following range.

00H ≤ CMP1n (M) < CMP0n (N) ≤ FFH

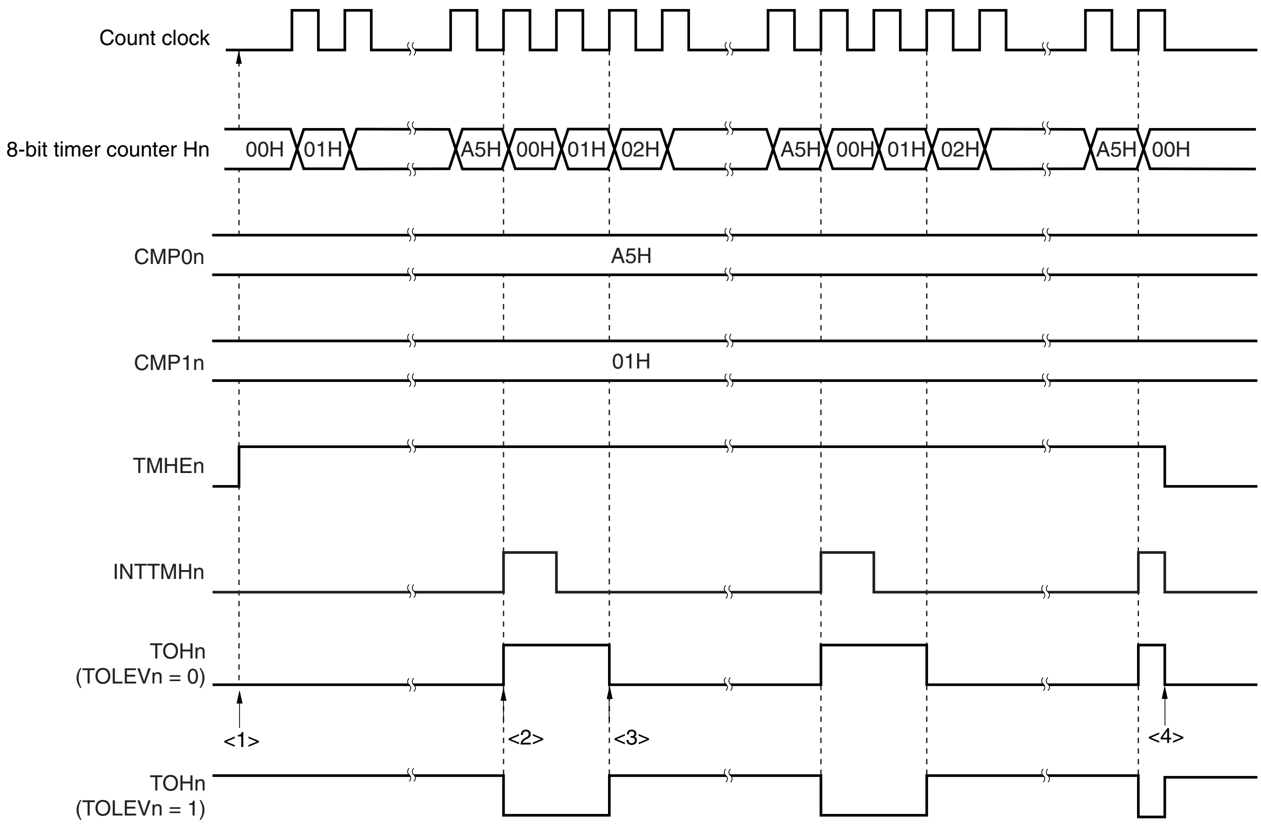
Remarks 1. For the setting of the output pin, see **8.3 (3) Port mode register 3 (PM3)**.

2. For details on how to enable the INTTMHn signal interrupt, see **CHAPTER 21 INTERRUPT FUNCTIONS**.

3. n = 0 to 2, however, TOH0 and TOH1 only for TOHn

Figure 8-14. Operation Timing in PWM Output Mode (1/4)

(a) Basic operation



<1> The count operation is enabled by setting the TMHEn bit to 1. Start 8-bit timer counter Hn by masking one count clock to count up. At this time, PWM output outputs an inactive level.

<2> When the values of 8-bit timer counter Hn and the CMP0n register match, an active level is output. At this time, the value of 8-bit timer counter Hn is cleared, and the INTTMHn signal is output.

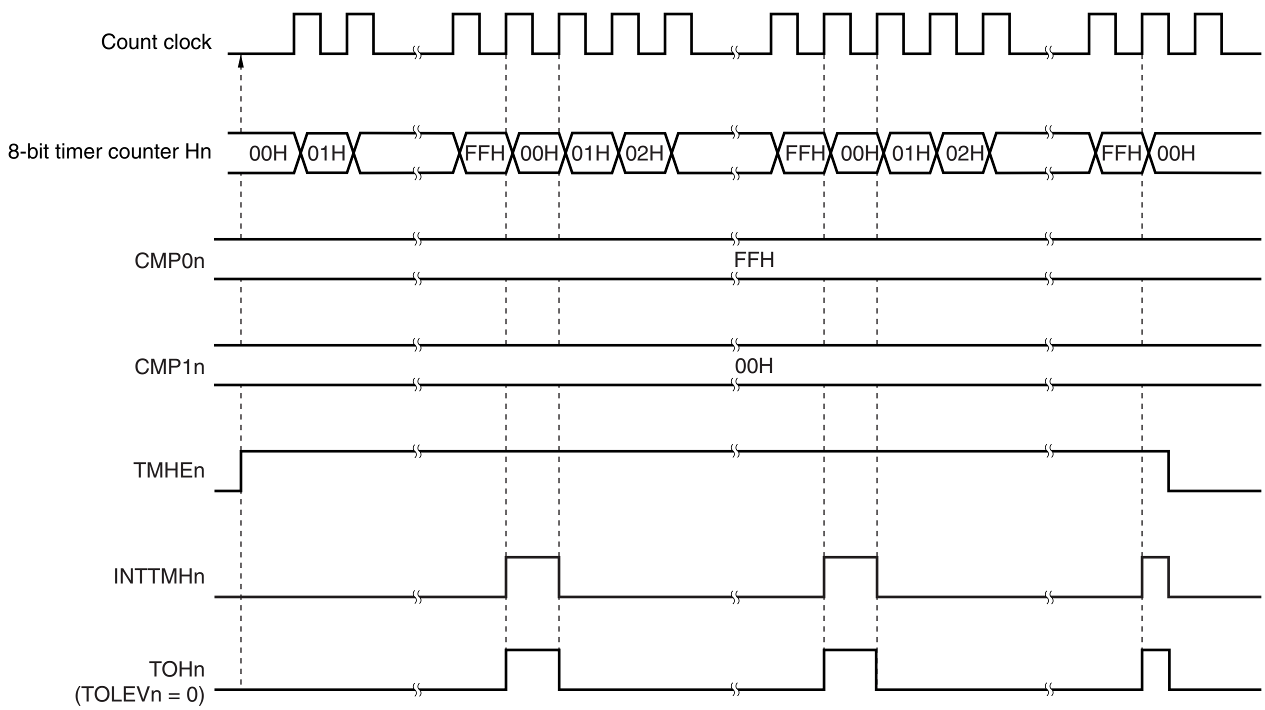
<3> When the values of 8-bit timer counter Hn and the CMP1n register match, an inactive level is output. At this time, the 8-bit counter value is not cleared and the INTTMHn signal is not output.

<4> Clearing the TMHEn bit to 0 during timer Hn operation sets the INTTMHn signal to the default and PWM output to an inactive level.

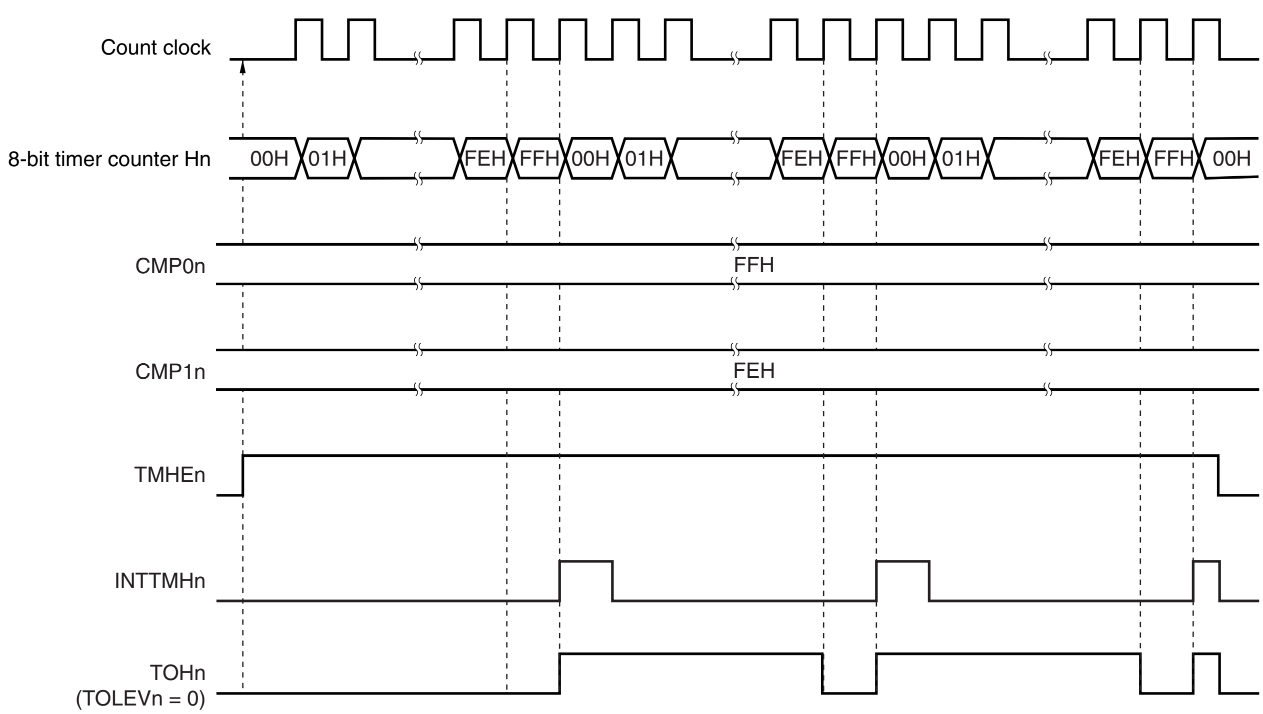
Remark n = 0 to 2, however, TOH0 and TOH1 only for TOHn

Figure 8-14. Operation Timing in PWM Output Mode (2/4)

(b) Operation when CMP0n = FFH, CMP1n = 00H



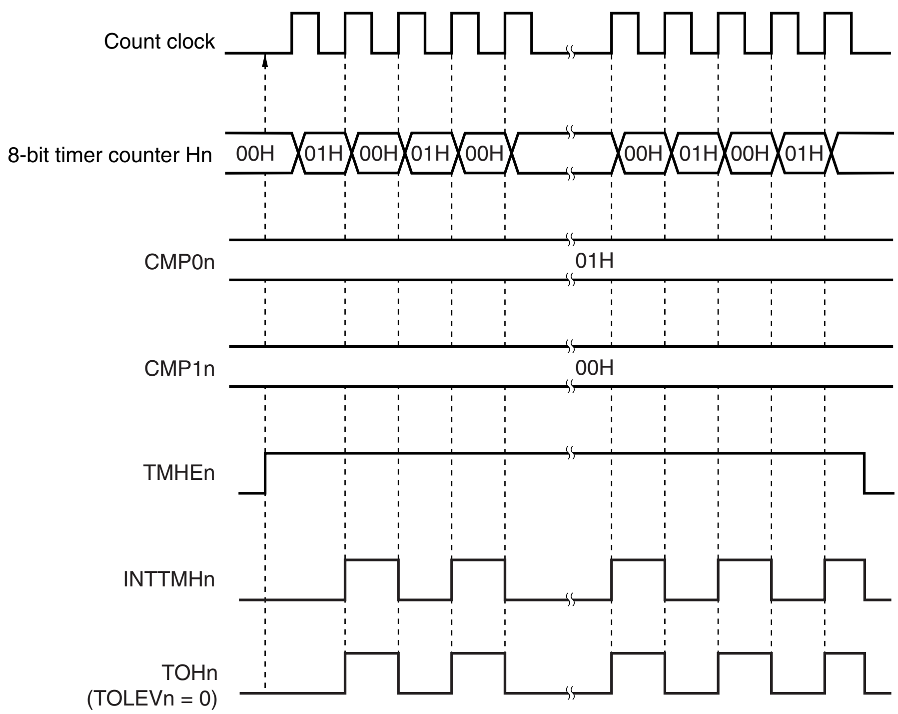
(c) Operation when CMP0n = FFH, CMP1n = FEH



Remark n = 0 to 2, however, TOH0 and TOH1 only for TOHn

Figure 8-14. Operation Timing in PWM Output Mode (3/4)

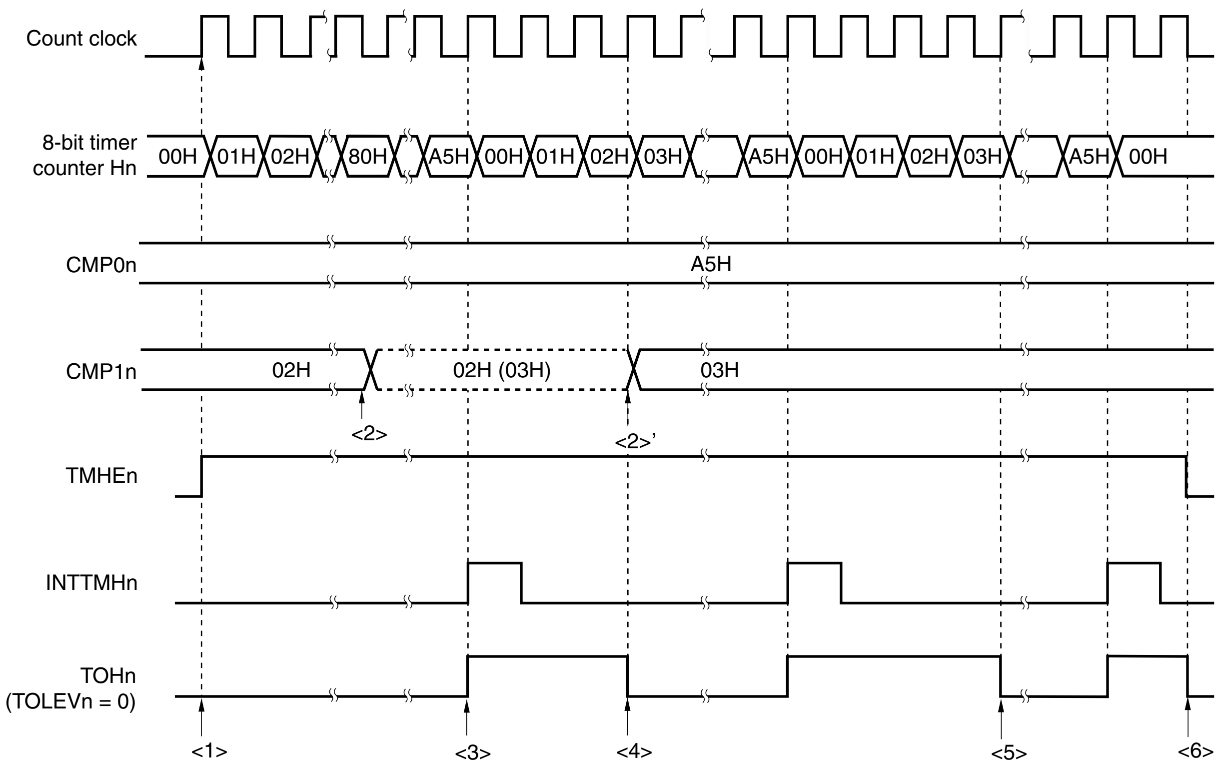
(d) Operation when CMP0n = 01H, CMP1n = 00H



Remark n = 0 to 2, however, TOH0 and TOH1 only for TOHn

Figure 8-14. Operation Timing in PWM Output Mode (4/4)

(e) Operation by changing CMP1n (CMP1n = 02H → 03H, CMP0n = A5H)



<1> The count operation is enabled by setting TMHEn = 1. Start 8-bit timer counter Hn by masking one count clock to count up. At this time, PWM output outputs an inactive level.

<2> The CMP1n register value can be changed during timer counter operation. This operation is asynchronous to the count clock.

<3> When the values of 8-bit timer counter Hn and the CMP0n register match, the value of 8-bit timer counter Hn is cleared, an active level is output, and the INTTMHn signal is output.

<4> If the CMP1n register value is changed, the value is latched and not transferred to the register. When the values of the 8-bit timer counter Hn and the CMP1n register before the change match, the value is transferred to the CMP1n register and the CMP1n register value is changed (<2>').
However, three count clocks or more are required from when the CMP1n register value is changed to when the value is transferred to the register. If a match signal is generated within three count clocks, the changed value cannot be transferred to the register.

<5> When the values of 8-bit timer counter Hn and the CMP1n register after the change match, an inactive level is output. 8-bit timer counter Hn is not cleared and the INTTMHn signal is not generated.

<6> Clearing the TMHEn bit to 0 during timer Hn operation sets the INTTMHn signal to the default and PWM output to an inactive level.

Remark n = 0 to 2, however, TOH0 and TOH1 only for TOHn

8.4.3 Carrier generator operation (8-bit timer H1 only)

In the carrier generator mode, the 8-bit timer H1 is used to generate the carrier signal of an infrared remote controller, and the 8-bit timer/event counter 51 is used to generate an infrared remote control signal (time count).

The carrier clock generated by the 8-bit timer H1 is output in the cycle set by the 8-bit timer/event counter 51.

In carrier generator mode, the output of the 8-bit timer H1 carrier pulse is controlled by the 8-bit timer/event counter 51, and the carrier pulse is output from the TOH1 output.

(1) Carrier generation

In carrier generator mode, the 8-bit timer H compare register 01 (CMP01) generates a low-level width carrier pulse waveform and the 8-bit timer H compare register 11 (CMP11) generates a high-level width carrier pulse waveform.

Rewriting the CMP11 register during the 8-bit timer H1 operation is possible but rewriting the CMP01 register is prohibited.

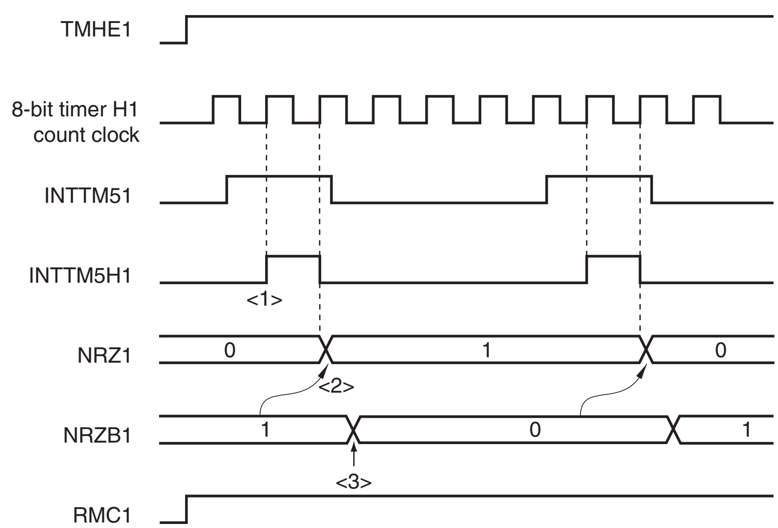
(2) Carrier output control

Carrier output is controlled by the interrupt request signal (INTTM51) of the 8-bit timer/event counter 51 and the NRZB1 and RMC1 bits of the 8-bit timer H carrier control register (TMCYC1). The relationship between the outputs is shown below.

RMC1 Bit	NRZB1 Bit	Output
0	0	Low-level output
0	1	High-level output at rising edge of INTTM51 signal input
1	0	Low-level output
1	1	Carrier pulse output at rising edge of INTTM51 signal input

To control the carrier pulse output during a count operation, the NRZ1 and NRZB1 bits of the TMCYC1 register have a master and slave bit configuration. The NRZ1 bit is read-only but the NRZB1 bit can be read and written. The INTTM51 signal is synchronized with the 8-bit timer H1 count clock and is output as the INTTM5H1 signal. The INTTM5H1 signal becomes the data transfer signal of the NRZ1 bit, and the NRZB1 bit value is transferred to the NRZ1 bit. The timing for transfer from the NRZB1 bit to the NRZ1 bit is as shown below.

Figure 8-15. Transfer Timing



<1> The INTTM51 signal is synchronized with the count clock of the 8-bit timer H1 and is output as the INTTM5H1 signal.

<2> The value of the NRZB1 bit is transferred to the NRZ1 bit at the second clock from the rising edge of the INTTM5H1 signal.

<3> Write the next value to the NRZB1 bit in the interrupt servicing program that has been started by the INTTM5H1 interrupt or after timing has been checked by polling the interrupt request flag. Write data to count the next time to the CR51 register.

Cautions 1. Do not rewrite the NRZB1 bit again until at least the second clock after it has been rewritten, or else the transfer from the NRZB1 bit to the NRZ1 bit is not guaranteed.

2. When the 8-bit timer/event counter 51 is used in the carrier generator mode, an interrupt is generated at the timing of <1>. When the 8-bit timer/event counter 51 is used in a mode other than the carrier generator mode, the timing of the interrupt generation differs.

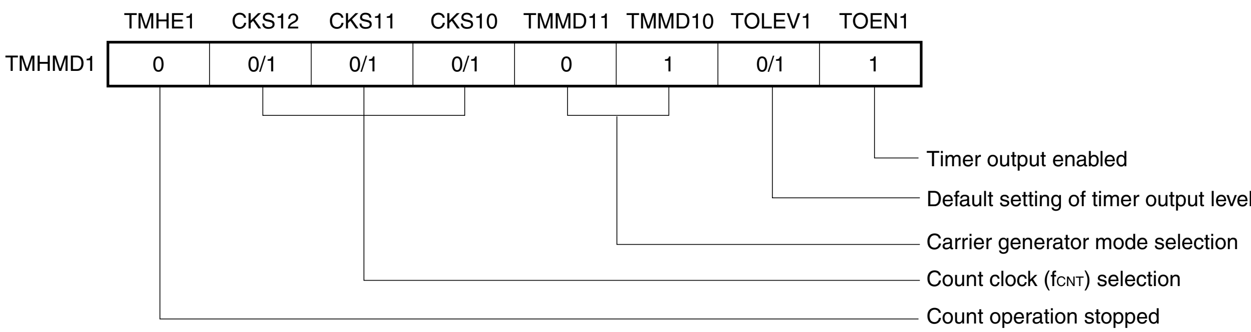
Remark INTTM5H1 is an internal signal and not an interrupt source.

Setting

<1> Set each register.

Figure 8-16. Register Setting in Carrier Generator Mode

(i) Setting 8-bit timer H mode register 1 (TMHMD1)



(ii) CMP01 register setting

- Compare value

(iii) CMP11 register setting

- Compare value

(iv) TMCYC1 register setting

- RMC1 = 1 ... Remote control output enable bit
- NRZB1 = 0/1 ... carrier output enable bit

(v) TCL51 and TMC51 register setting

- See **7.3 Registers Controlling 8-Bit Timer/Event Counters 50, 51, and 52**.

<2> When TMHE1 = 1, the 8-bit timer H1 starts counting.

<3> When TCE51 of the 8-bit timer mode control register 51 (TMC51) is set to 1, the 8-bit timer/event counter 51 starts counting.

<4> After the count operation is enabled, the first compare register to be compared is the CMP01 register. When the count value of the 8-bit timer counter H1 and the CMP01 register value match, the INTTMH1 signal is generated, the 8-bit timer counter H1 is cleared. At the same time, the compare register to be compared with the 8-bit timer counter H1 is switched from the CMP01 register to the CMP11 register.

<5> When the count value of the 8-bit timer counter H1 and the CMP11 register value match, the INTTMH1 signal is generated, the 8-bit timer counter H1 is cleared. At the same time, the compare register to be compared with the 8-bit timer counter H1 is switched from the CMP11 register to the CMP01 register.

<6> By performing procedures <4> and <5> repeatedly, a carrier clock is generated.

<7> The INTTM51 signal is synchronized with count clock of the 8-bit timer H1 and output as the INTTM5H1 signal. The INTTM5H1 signal becomes the data transfer signal for the NRZB1 bit, and the NRZB1 bit value is transferred to the NRZ1 bit.

<8> Write the next value to the NRZB1 bit in the interrupt servicing program that has been started by the INTTM5H1 interrupt or after timing has been checked by polling the interrupt request flag. Write data to count the next time to the CR51 register.

<9> When the NRZ1 bit is high level, a carrier clock is output by TOH1 output.

<10> By performing the procedures above, an arbitrary carrier clock is obtained. To stop the count operation, clear TMHE1 to 0.

If the setting value of the CMP01 register is N, the setting value of the CMP11 register is M, and the count clock frequency is f_{CNT}, the carrier clock output cycle and duty are as follows.

- Carrier clock output cycle = (N + M + 2)/f_{CNT}
- Duty = High-level width/carrier clock output width = (M + 1)/(N + M + 2)

Cautions 1. Be sure to set the CMP11 register when starting the timer count operation (TMHE1 = 1) after the timer count operation was stopped (TMHE1 = 0) (be sure to set again even if setting the same value to the CMP11 register).

2. Set so that the count clock frequency of TMH1 becomes more than 6 times the count clock frequency of TM51.

3. Set the values of the CMP01 and CMP11 registers in a range of 01H to FFH.

4. The set value of the CMP11 register can be changed while the timer counter is operating. However, it takes the duration of three operating clocks (signal selected by the CKS12 to CKS10 bits of the TMHMD1 register) since the value of the CMP11 register has been changed until the value is transferred to the register.

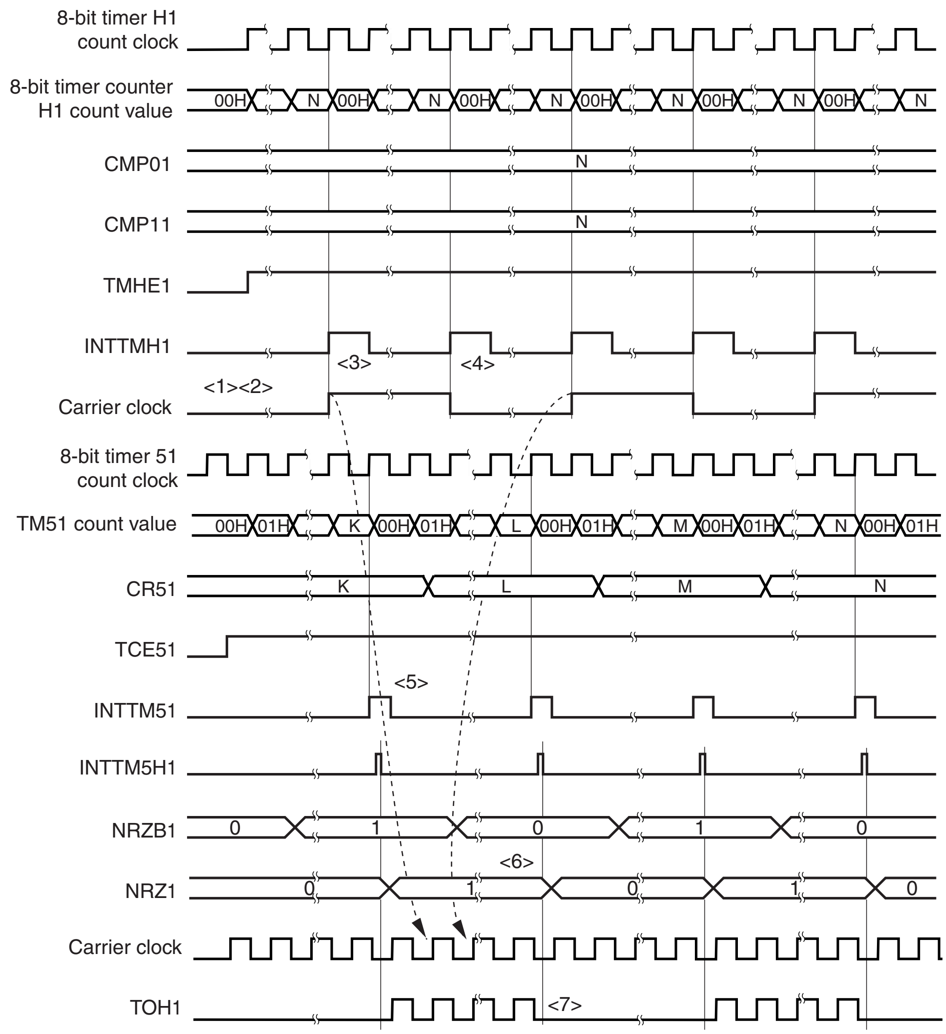
5. Be sure to set the RMC1 bit before the count operation is started.

Remarks 1. For the setting of the output pin, see **8.3 (3) Port mode register 3 (PM3)**.

2. For how to enable the INTTMH1 signal interrupt, see **CHAPTER 21 INTERRUPT FUNCTIONS**.

Figure 8-17. Carrier Generator Mode Operation Timing (1/3)

(a) Operation when CMP01 = N, CMP11 = N



<1> When TMHE1 = 0 and TCE51 = 0, the 8-bit timer counter H1 operation is stopped.

<2> When TMHE1 = 1 is set, the 8-bit timer counter H1 starts a count operation. At that time, the carrier clock remains default.

<3> When the count value of the 8-bit timer counter H1 matches the CMP01 register value, the first INTTMH1 signal is generated, the carrier clock signal is inverted, and the compare register to be compared with the 8-bit timer counter H1 is switched from the CMP01 register to the CMP11 register. The 8-bit timer counter H1 is cleared to 00H.

<4> When the count value of the 8-bit timer counter H1 matches the CMP11 register value, the INTTMH1 signal is generated, the carrier clock signal is inverted, and the compare register to be compared with the 8-bit timer counter H1 is switched from the CMP11 register to the CMP01 register. The 8-bit timer counter H1 is cleared to 00H. By performing procedures <3> and <4> repeatedly, a carrier clock with duty fixed to 50% is generated.

<5> When the INTTM51 signal is generated, it is synchronized with the 8-bit timer H1 count clock and is output as the INTTM5H1 signal.

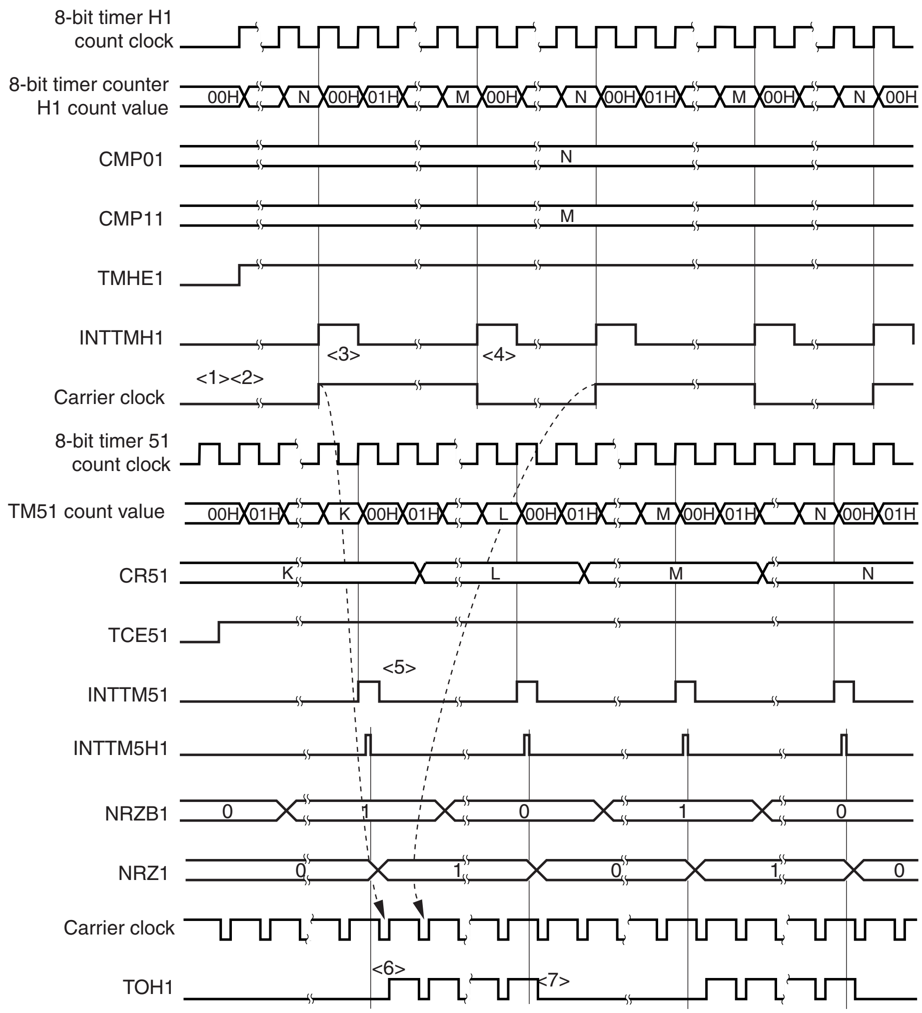
<6> The INTTM5H1 signal becomes the data transfer signal for the NRZB1 bit, and the NRZB1 bit value is transferred to the NRZ1 bit.

<7> When NRZ1 = 0 is set, the TOH1 output becomes low level.

Remark INTTM5H1 is an internal signal and not an interrupt source.

Figure 8-17. Carrier Generator Mode Operation Timing (2/3)

(b) Operation when CMP01 = N, CMP11 = M



<1> When TMHE1 = 0 and TCE51 = 0, the 8-bit timer counter H1 operation is stopped.

<2> When TMHE1 = 1 is set, the 8-bit timer counter H1 starts a count operation. At that time, the carrier clock remains default.

<3> When the count value of the 8-bit timer counter H1 matches the CMP01 register value, the first INTTMH1 signal is generated, the carrier clock signal is inverted, and the compare register to be compared with the 8-bit timer counter H1 is switched from the CMP01 register to the CMP11 register. The 8-bit timer counter H1 is cleared to 00H.

<4> When the count value of the 8-bit timer counter H1 matches the CMP11 register value, the INTTMH1 signal is generated, the carrier clock signal is inverted, and the compare register to be compared with the 8-bit timer counter H1 is switched from the CMP11 register to the CMP01 register. The 8-bit timer counter H1 is cleared to 00H. By performing procedures <3> and <4> repeatedly, a carrier clock with duty fixed to other than 50% is generated.

<5> When the INTTM51 signal is generated, it is synchronized with the 8-bit timer H1 count clock and is output as the INTTM5H1 signal.

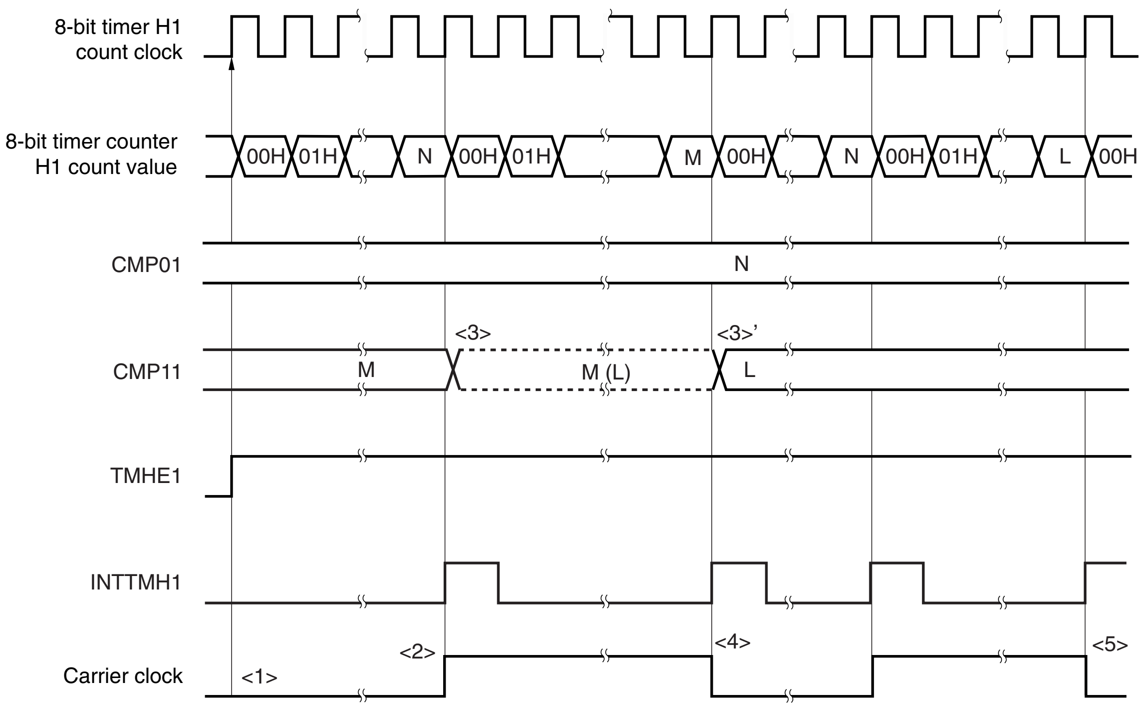
<6> A carrier signal is output at the first rising edge of the carrier clock if NRZ1 is set to 1.

<7> When NRZ1 = 0, the TOH1 output is held at the high level and is not changed to low level while the carrier clock is high level (from <6> and <7>, the high-level width of the carrier clock waveform is guaranteed).

Remark INTTM5H1 is an internal signal and not an interrupt source.

Figure 8-17. Carrier Generator Mode Operation Timing (3/3)

(c) Operation when CMP11 is changed



<1> When TMHE1 = 1 is set, the 8-bit timer H1 starts a count operation. At that time, the carrier clock remains default.

<2> When the count value of the 8-bit timer counter H1 matches the value of the CMP01 register, the INTTMH1 signal is output, the carrier signal is inverted, and the timer counter is cleared to 00H. At the same time, the compare register whose value is to be compared with that of the 8-bit timer counter H1 is changed from the CMP01 register to the CMP11 register.

<3> The CMP11 register is asynchronous to the count clock, and its value can be changed while the 8-bit timer H1 is operating. The new value (L) to which the value of the register is to be changed is latched. When the count value of the 8-bit timer counter H1 matches the value (M) of the CMP11 register before the change, the CMP11 register is changed (<3>').
However, it takes three count clocks or more since the value of the CMP11 register has been changed until the value is transferred to the register. Even if a match signal is generated before the duration of three count clocks elapses, the new value is not transferred to the register.

<4> When the count value of 8-bit timer counter H1 matches the value (M) of the CMP1 register before the change, the INTTMH1 signal is output, the carrier signal is inverted, and the timer counter is cleared to 00H. At the same time, the compare register whose value is to be compared with that of the 8-bit timer counter H1 is changed from the CMP11 register to the CMP01 register.

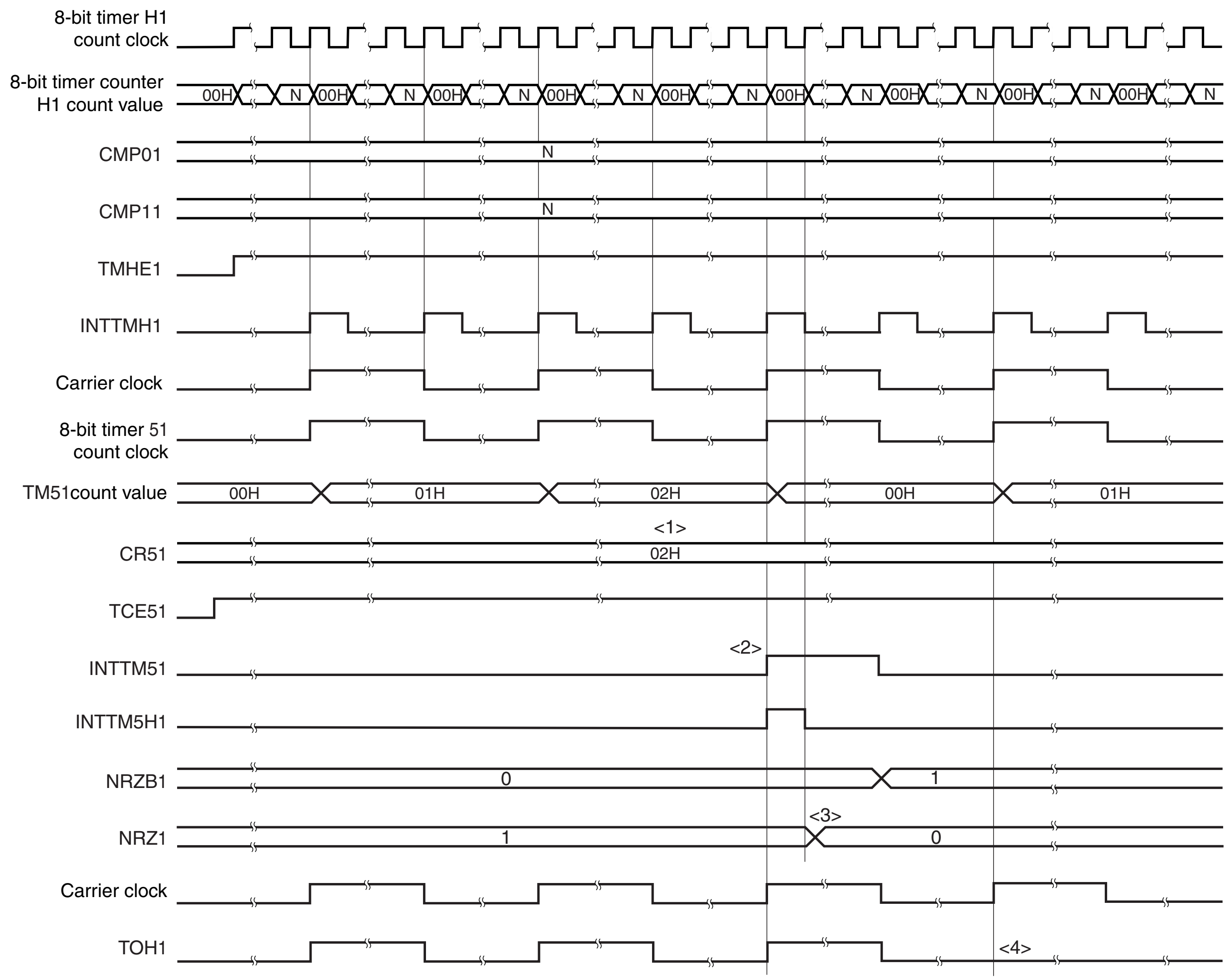
<5> The timing at which the count value of the 8-bit timer counter H1 and the CMP11 register value match again is indicated by the value after the change (L).

<R> 8.4.4 Control of number of carrier clocks by timer 51 counter

The number of carrier clocks to be output from the TOH1 pin can be controlled by selecting the timer H1 output signal for the 8-bit timer 51 count clock.

Figure 8-18 shows an example of control when three carrier clocks are to be output from the TOH1 pin

Figure 8-18. Example of Controlling Number of Carrier Clocks by Timer 51 Counter (Setting Timer H1 Output Signal for Timer 51 Count Clock (TCL51 = 07H))



<1> Set the CR51 register to 02H when three carrier clocks are to be output from the TOH1 pin.

<2> The INTTM51 signal is generated when the TM51 count value and the CR51 register value (02H) have matched. The signal is synchronized with the 8-bit timer H1 count clock and is output as the INTTM5H1 signal.

<3> The INTTM5H1 signal becomes the data transfer signal for the NRZB1 bit, and the NRZB1 bit value is transferred to the NRZ1 bit.

The transfer timing at this time is one timer H1 count clock after a rise of the INTTM5H1 signal.

<4> By setting NRZ1 to 0, the TOH1 output becomes low level after having output the third carrier clock.

Remark INTTM5H1 is an internal signal and not an interrupt source.

CHAPTER 9 REAL-TIME COUNTER

9.1 Functions of Real-Time Counter

The real-time counter has the following features.

- Having counters of year, month, week, day, hour, minute, and second, and can count up to 99 years.
- Constant-period interrupt function (period: 1 month to 0.5 seconds)
- Alarm interrupt function (alarm: week, hour, minute)
- Interval interrupt function
- Pin output function of 1 Hz
- Pin output function of 512 Hz or 16.384 kHz or 32.768 kHz

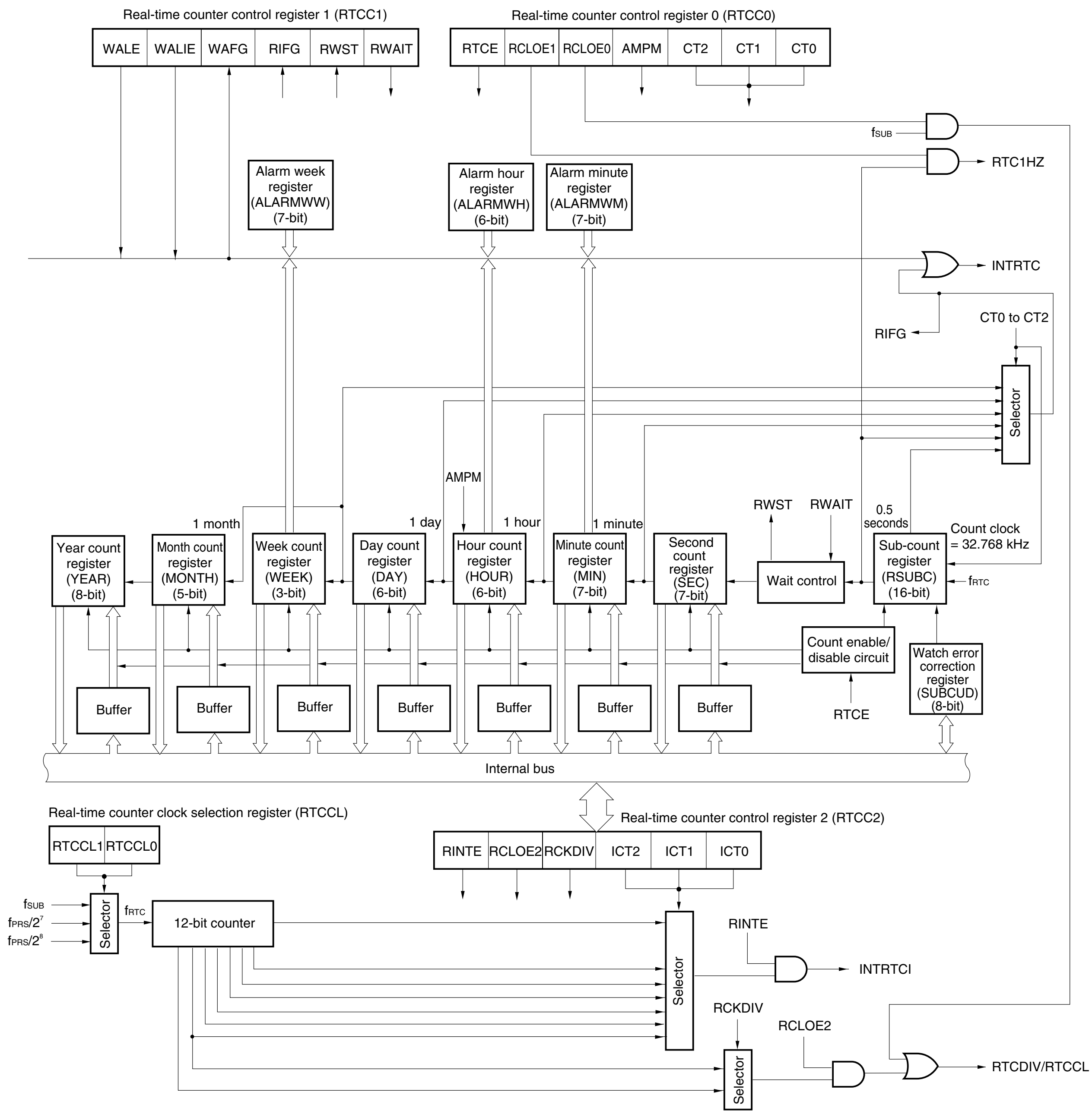
9.2 Configuration of Real-Time Counter

The real-time counter includes the following hardware.

Table 9-1. Configuration of Real-Time Counter

Item	Configuration
Control registers	Real-time counter clock selection register (RTCCL) Real-time counter control register 0 (RTCC0) Real-time counter control register 1 (RTCC1) Real-time counter control register 2 (RTCC2) Sub-count register (RSUBC) Second count register (SEC) Minute count register (MIN) Hour count register (HOUR) Day count register (DAY) Week count register (WEEK) Month count register (MONTH) Year count register (YEAR) Watch error correction register (SUBCUD) Alarm minute register (ALARMWM) Alarm hour register (ALARMWH) Alarm week register (ALARMWW)

Figure 9-1. Block Diagram of Real-Time Counter



9.3 Registers Controlling Real-Time Counter

Timer real-time counter is controlled by the following 16 registers.

(1) Real-time counter clock selection register (RTCCL)

This register controls the mode of real-time counter.

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

Reset signal generation clears this register to 00H.

Figure 9-2. Format of Real-time Counter Clock Selection Register (RTCCL)

Address: FF54H After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
RTCCL	0	0	0	0	0	0	RTCCL1	RTCCL0

RTCCL1	RTCCL0	Control of real-time counter (RTC) input clock (f_{RTC})
0	0	f_{SUB}
0	1	$f_{PRS}/2^7$
1	0	$f_{PRS}/2^8$
1	1	Setting prohibited

Remark • When f_{PRS} = 4.19 MHz, $f_{RTC} = f_{PRS}/2^7$ = 32.768 kHz
• When f_{PRS} = 8.38 MHz, $f_{RTC} = f_{PRS}/2^8$ = 32.768 kHz

(2) Real-time counter control register 0 (RTCC0)

The RTCC0 register is an 8-bit register that is used to start or stop the real-time counter operation, control the RTCCL and RTC1HZ pins, and set a 12- or 24-hour system and the constant-period interrupt function.

RTCC0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 9-3. Format of Real-Time Counter Control Register 0 (RTCC0)

Address: FF89H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	3	2	1	0
RTCC0	RTCE	0	RCLOE1	RCLOE0	AMPM	CT2	CT1	CT0

RTCE	Real-time counter operation control
0	Stops counter operation.
1	Starts counter operation.

RCLOE1	RTC1HZ pin output control
0	Disables output of RTC1HZ pin (1 Hz).
1	Enables output of RTC1HZ pin (1 Hz).

RCLOE0[Note]	RTCCL pin output control
0	Disables output of RTCCL pin (32.768 kHz).
1	Enables output of RTCCL pin (32.768 kHz).

AMPM	Selection of 12-/24-hour system
0	12-hour system (a.m. and p.m. are displayed.)
1	24-hour system
• To change the value of AMPM, set RWAIT (bit 0 of RTCC1) to 1, and re-set the hour count register (HOUR). • Table 9-2 shows the displayed time digits.	

CT2	CT1	CT0	Constant-period interrupt (INTRTC) selection
0	0	0	Does not use constant-period interrupt function.
0	0	1	Once per 0.5 s (synchronized with second count up)
0	1	0	Once per 1 s (same time as second count up)
0	1	1	Once per 1 m (second 00 of every minute)
1	0	0	Once per 1 hour (minute 00 and second 00 of every hour)
1	0	1	Once per 1 day (hour 00, minute 00, and second 00 of every day)
1	1	×	Once per 1 month (Day 1, hour 00 a.m., minute 00, and second 00 of every month)
After changing the values of CT2 to CT0, clear the interrupt request flag.			

Note RCLOE0 and RCLOE2 must not be enabled at the same time.

Caution If RCLOE0 and RCLOE1 are changed when RTCE = 1, a pulse with a narrow width may be generated on the 32.768 kHz and 1 Hz output signals.

Remark ×: don't care

(3) Real-time counter control register 1 (RTCC1)

The RTCC1 register is an 8-bit register that is used to control the alarm interrupt function and the wait time of the counter.

RTCC1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 9-4. Format of Real-Time Counter Control Register 1 (RTCC1) (1/2)

Address: FF8AH After reset: 00H R/W

Symbol	<7>	<6>	5	<4>	<3>	2	<1>	<0>
RTCC1	WALE	WALIE	0	WAFG	RIFG	0	RWST	RWAIT

WALE	Alarm operation control
0	Match operation is invalid.
1	Match operation is valid.
To set the registers of alarm (WALIE flag of RTCC1, ALARMWM register, ALARMWH register, and ALARMWW register), disable WALE (clear it to “0”).	

WALIE	Control of alarm interrupt (INTRTC) function operation
0	Does not generate interrupt on matching of alarm.
1	Generates interrupt on matching of alarm.

WAFG	Alarm detection status flag
0	Alarm mismatch
1	Detection of matching of alarm
This is a status flag that indicates detection of matching with the alarm. It is valid only when WALE = 1 and is set to “1” one clock (32.768 kHz) after matching of the alarm is detected. This flag is cleared when “0” is written to it. Writing “1” to it is invalid.	

RIFG	Constant-period interrupt status flag
0	Constant-period interrupt is not generated.
1	Constant-period interrupt is generated.
This flag indicates the status of generation of the constant-period interrupt. When the constant-period interrupt is generated, it is set to “1”. This flag is cleared when “0” is written to it. Writing “1” to it is invalid.	

RWST	Wait status flag of real-time counter
0	Counter is operating.
1	Mode to read or write counter value
This status flag indicates whether the setting of RWAIT is valid. Before reading or writing the counter value, confirm that the value of this flag is 1.	

Figure 9-4. Format of Real-Time Counter Control Register 1 (RTCC1) (2/2)

RWAIT	Wait control of real-time counter
0	Sets counter operation.
1	Stops SEC to YEAR counters. Mode to read or write counter value
This bit controls the operation of the counter. Be sure to write "1" to it to read or write the counter value. Because RSUBC continues operation, complete reading or writing of it in 1 second, and clear this bit back to 0. When RWAIT = 1, it takes up to 1 clock (32.768 kHz) until the counter value can be read or written. If RSUBC overflows when RWAIT = 1, it counts up after RWAIT = 0. If the second count register is written, however, it does not count up because RSUBC is cleared.	

<R> **Caution The RIFG and WAFG flags may be cleared when the RTCC1 register is written by using a 1-bit manipulation instruction. Use, therefore, an 8-bit manipulation instruction in order to write to the RTCC1 register. To prevent the RIFG and WAFG flags from being cleared during writing, disable writing by setting "1" to the corresponding bit. When the value may be rewritten because the RIFG and WAFG flags are not being used, the RTCC1 register may be written by using a 1-bit manipulation instruction.**

Remark Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

(4) Real-time counter control register 2 (RTCC2)

The RTCC2 register is an 8-bit register that is used to control the interval interrupt function and the RTCDIV pin.

RTCC2 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 9-5. Format of Real-Time Counter Control Register 2 (RTCC2)

Address: FF8BH After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
RTCC2	RINTE	RCLOE2	RCKDIV	0	0	ICT2	ICT1	ICT0

RINTE	ICT2	ICT1	ICT0	Interval interrupt (INTRTCI) selection
0	×	×	×	Interval interrupt is not generated.
1	0	0	0	$2^6/f_{RTC}$ (1.953125 ms)
1	0	0	1	$2^7/f_{RTC}$ (3.90625 ms)
1	0	1	0	$2^8/f_{RTC}$ (7.8125 ms)
1	0	1	1	$2^9/f_{RTC}$ (15.625 ms)
1	1	0	0	$2^{10}/f_{RTC}$ (31.25 ms)
1	1	0	1	$2^{11}/f_{RTC}$ (62.5 ms)
1	1	1	×	$2^{12}/f_{RTC}$ (125 ms)

RCLOE2[Note]	RTCDIV pin output control
0	Output of RTCDIV pin is disabled.
1	Output of RTCDIV pin is enabled.

RCKDIV	Selection of RTCDIV pin output frequency
0	RTCDIV pin outputs 512 Hz (1.95 ms).
1	RTCDIV pin outputs 16.384 kHz (0.061 ms).

Note RCLOE0 and RCLOE2 must not be enabled at the same time.

Cautions 1. Change ICT2, ICT1, and ICT0 when RINTE = 0.

2. When the output from RTCDIV pin is stopped, the output continues after a maximum of two clocks of f_{RTC} and enters the low level. While 512 Hz is output, and when the output is stopped immediately after entering the high level, a pulse of at least one clock width of f_{XT} may be generated.

(5) Sub-count register (RSUBC)

The RSUBC register is a 16-bit register that counts the reference time of 1 second of the real-time counter. It takes a value of 0000H to 7FFFH and counts 1 second with a clock of 32.768 kHz.

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

Reset signal generation clears this register to 0000H.

Cautions 1. When a correction is made by using the SUBCUD register, the value may become 8000H or more.

2. This register is also cleared by reset effected by writing the second count register.

3. The value read from this register is not guaranteed if it is read during operation, because a value that is changing is read.

Figure 9-6. Format of Sub-Count Register (RSUBC)

Address: FF60H After reset: 0000H R

Symbol	7	6	5	4	3	2	1	0
RSUBC	SUBC7	SUBC6	SUBC5	SUBC4	SUBC3	SUBC2	SUBC1	SUBC0

Address: FF61H After reset: 0000H R

Symbol	7	6	5	4	3	2	1	0
RSUBC	SUBC15	SUBC14	SUBC13	SUBC12	SUBC11	SUBC10	SUBC9	SUBC8

(6) Second count register (SEC)

The SEC register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of seconds.

It counts up when the sub-counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 59 to this register in BCD code. If a value outside this range is set, the register value returns to the normal value after 1 period.

SEC can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 9-7. Format of Second Count Register (SEC)

Address: FF62H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

(7) Minute count register (MIN)

The MIN register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of minutes.

It counts up when the second counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 59 to this register in BCD code. If a value outside this range is set, the register value returns to the normal value after 1 period.

MIN can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 9-8. Format of Minute Count Register (MIN)

Address: FF63H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

(8) Hour count register (HOUR)

The HOUR register is an 8-bit register that takes a value of 00 to 23 or 01 to 12, 21 to 32 (decimal) and indicates the count value of hours.

It counts up when the minute counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 23 or 01 to 12, 21 to 32 to this register in BCD code. If a value outside this range is set, the register value returns to the normal value after 1 period.

HOUR can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 12H.

However, the value of this register is 00H if the AMPM bit is set to 1 after reset.

Figure 9-9. Format of Hour Count Register (HOUR)

Address: FF64H After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

Caution Bit 5 (HOUR20) of HOUR indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

Table 9-2 shows the relationship between the setting value of the AMPM bit, the HOUR register value, and time.

<R>

Table 9-2. Displayed Time Digits

24-Hour Display (AMPM bit = 1)		12-Hour Display (AMPM bit = 0)	
Time	HOUR Register	Time	HOUR Register
0	00H	0 a.m.	12H
1	01H	1 a.m.	01H
2	02H	2 a.m.	02H
3	03H	3 a.m.	03H
4	04H	4 a.m.	04H
5	05H	5 a.m.	05H
6	06H	6 a.m.	06H
7	07H	7 a.m.	07H
8	08H	8 a.m.	08H
9	09H	9 a.m.	09H
10	10H	10 a.m.	10H
11	11H	11 a.m.	11H
12	12H	0 p.m.	32H
13	13H	1 p.m.	21H
14	14H	2 p.m.	22H
15	15H	3 p.m.	23H
16	16H	4 p.m.	24H
17	17H	5 p.m.	25H
18	18H	6 p.m.	26H
19	19H	7 p.m.	27H
20	20H	8 p.m.	28H
21	21H	9 p.m.	29H
22	22H	10 p.m.	30H
23	23H	11 p.m.	31H

The HOUR register value is set to 12-hour display when the AMPM bit is “0” and to 24-hour display when the AMPM bit is “1”.

In 12-hour display, the fifth bit of the HOUR register displays 0 for AM and 1 for PM.

(9) Day count register (DAY)

The DAY register is an 8-bit register that takes a value of 1 to 31 (decimal) and indicates the count value of days.

It counts up when the hour counter overflows.

This counter counts as follows.

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February, leap year)
- 01 to 28 (February, normal year)

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 01 to 31 to this register in BCD code. If a value outside this range is set, the register value returns to the normal value after 1 period.

DAY can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 01H.

Figure 9-10. Format of Day Count Register (DAY)

Address: FF66H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

(10) Week count register (WEEK)

The WEEK register is an 8-bit register that takes a value of 0 to 6 (decimal) and indicates the count value of weekdays.

It counts up in synchronization with the day counter.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 06 to this register in BCD code. If a value outside this range is set, the register value returns to the normal value after 1 period.

WEEK can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 9-11. Format of Week Count Register (WEEK)

Address: FF65H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

<R> **Caution The value corresponding to the month count register or the day count register is not stored in the week count register automatically. After reset release, set the week count register as follow.**

Day	WEEK
Sunday	00H
Monday	01H
Tuesday	02H
Wednesday	03H
Thursday	04H
Friday	05H
Saturday	06H

(11) Month count register (MONTH)

The MONTH register is an 8-bit register that takes a value of 1 to 12 (decimal) and indicates the count value of months.

It counts up when the day counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 01 to 12 to this register in BCD code. If a value outside this range is set, the register value returns to the normal value after 1 period.

MONTH can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 01H.

Figure 9-12. Format of Month Count Register (MONTH)

Address: FF67H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

(12) Year count register (YEAR)

The YEAR register is an 8-bit register that takes a value of 0 to 99 (decimal) and indicates the count value of years.

It counts up when the month counter overflows.

Values 00, 04, 08, …, 92, and 96 indicate a leap year.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 99 to this register in BCD code. If a value outside this range is set, the register value returns to the normal value after 1 period.

YEAR can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 9-13. Format of Year Count Register (YEAR)

Address: FF68H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

<R> **(13) Watch error correction register (SUBCUD)**

This register is used to correct the watch with high accuracy when it is slow or fast by changing the value (reference value: 7FFFH) that overflows from the sub-count register (RSUBC) to the second count register.

SUBCUD can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 9-14. Format of Watch Error Correction Register (SUBCUD)

Address: FF82H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SUBCUD	DEV	F6	F5	F4	F3	F2	F1	F0

DEV	Setting of watch error correction timing
0	Corrects watch error when the second digits are at 00, 20, or 40 (every 20 seconds).
1	Corrects watch error only when the second digits are at 00 (every 60 seconds).

F6	Setting of watch error correction value
0	Increases by {(F5, F4, F3, F2, F1, F0) – 1} × 2.
1	Decreases by {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2.

When (F6, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, *), the watch error is not corrected. * is 0 or 1.
/F5 to /F0 are the inverted values of the corresponding bits (000011 when 111100).
Range of correction value: (when F6 = 0) 2, 4, 6, 8, … , 120, 122, 124
(when F6 = 1) –2, –4, –6, –8, … , –120, –122, –124

The range of value that can be corrected by using the watch error correction register (SUBCUD) is shown below.

	DEV = 0 (correction every 20 seconds)	DEV = 1 (correction every 60 seconds)
Correctable range	–189.2 ppm to 189.2 ppm	–63.1 ppm to 63.1 ppm
Maximum excludes quantization error	± 1.53 ppm	± 0.51 ppm
Minimum resolution	± 3.05 ppm	± 1.02 ppm

Remark Set DEV to 0 when the correction range is –63.1 ppm or less, or 63.1 ppm or more.

(14) Alarm minute register (ALARMWM)

This register is used to set minutes of alarm.

ALARMWM can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Caution Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 9-15. Format of Alarm Minute Register (ALARMWM)

Address: FF86H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

(15) Alarm hour register (ALARMWH)

This register is used to set hours of alarm.

ALARMWH can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 12H.

<R> However, the value of this register is 00H if the AMPM bit is set to 1 after reset.

Caution Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 9-16. Format of Alarm Hour Register (ALARMWH)

Address: FF87H After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWH	0	0	WH20	WH10	WH8	WH4	WH2	WH1

Caution Bit 5 (WH20) of ALARMWH indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

(16) Alarm week register (ALARMWW)

This register is used to set date of alarm.
ALARMWW can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 9-17. Format of Alarm Week Register (ALARMWW)

Address: FF88H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

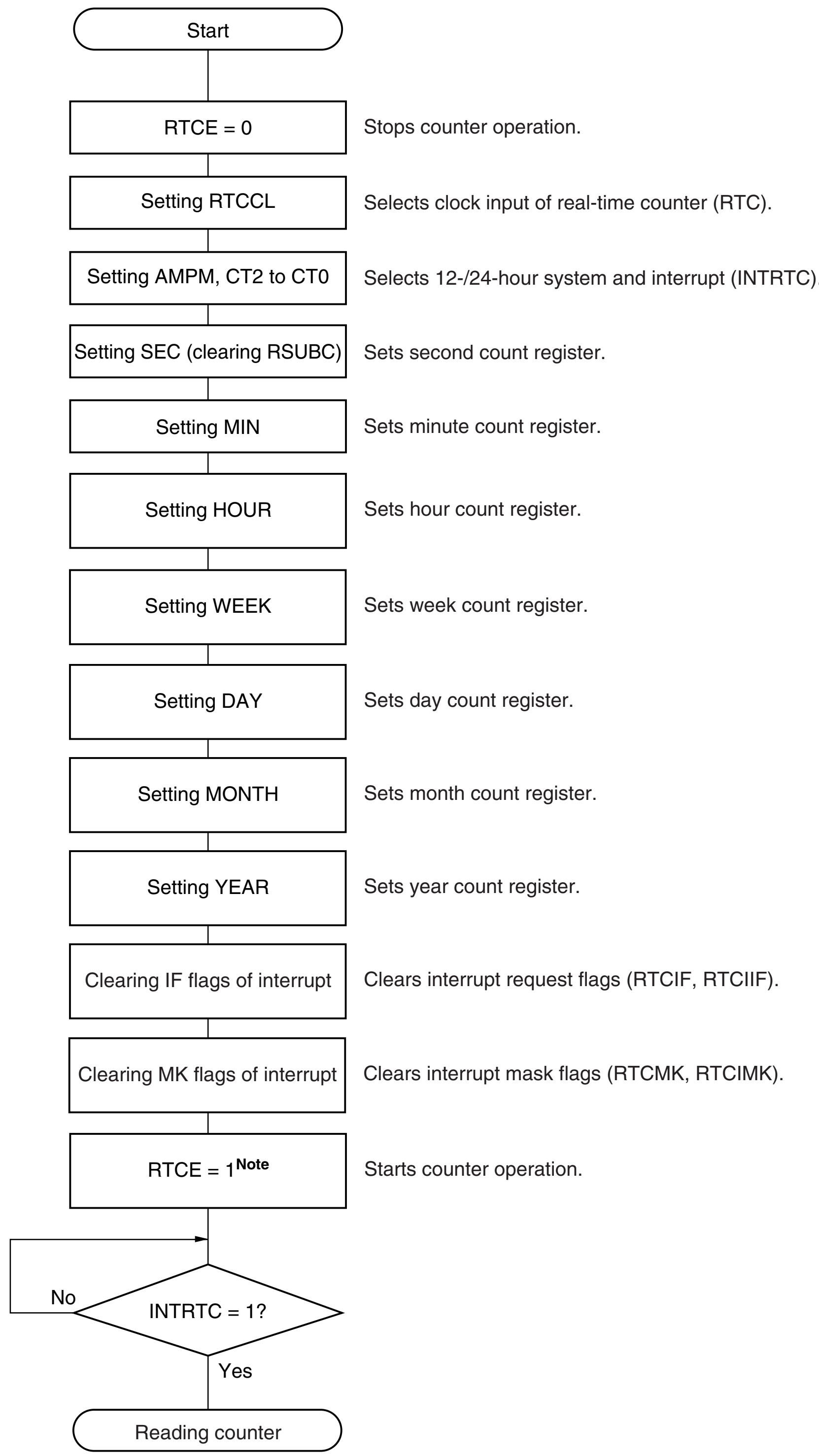
Here is an example of setting the alarm.

Time of Alarm	Day							12-Hour Display				24-Hour Display			
	Sunday WW0	Monday WW1	Tuesday WW2	Wednesday WW3	Thursday WW4	Friday WW5	Saturday WW6	Hour 10	Hour 1	Minute 10	Minute 1	Hour 10	Hour 1	Minute 10	Minute 1
Every day, 0:00 a.m.	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
Every day, 1:30 a.m.	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
Every day, 11:59 a.m.	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
Monday through Friday, 0:00 p.m.	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
Sunday, 1:30 p.m.	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
Monday, Wednesday, Friday, 11:59 p.m.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

9.4 Real-Time Counter Operation

9.4.1 Starting operation of real-time counter

Figure 9-18. Procedure for Starting Operation of Real-Time Counter



<R> **Note** Confirm the procedure described in **9.4.2 Shifting to STOP mode after starting operation** when shifting to STOP mode without waiting for INTRTC = 1 after RTCE = 1.

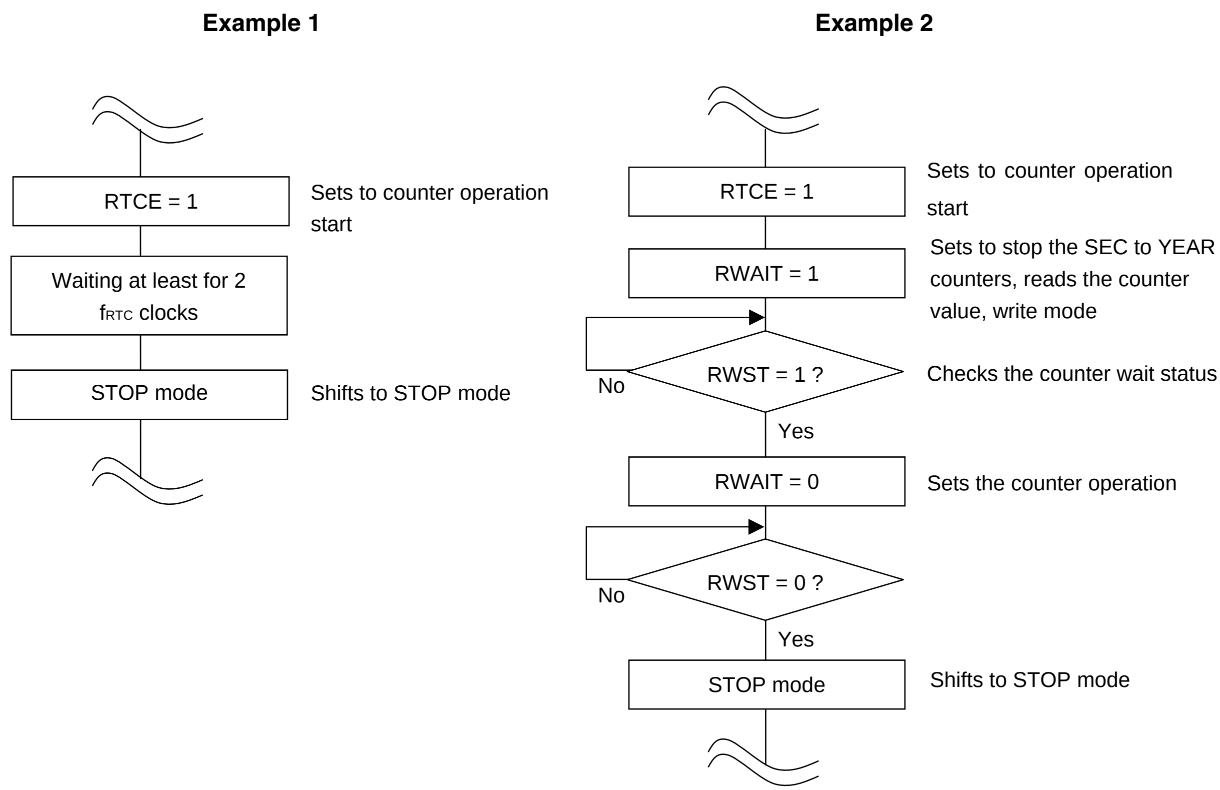
<R> 9.4.2 Shifting to STOP mode after starting operation

Perform one of the following processing when shifting to STOP mode immediately after setting RTCE to 1.

However, after setting RTCE to 1, this processing is not required when shifting to STOP mode after the first INTRTC interrupt has occurred.

- Shifting to STOP mode when at least two input clocks (f_{RTC}) have elapsed after setting RTCE to 1 (see **Figure 9-19**, **Example 1**).
- Checking by polling RWST to become 1, after setting RTCE to 1 and then setting RWAIT to 1. Afterward, setting RWAIT to 0 and shifting to STOP mode after checking again by polling that RWST has become 0 (see **Figure 9-19**, **Example 2**).

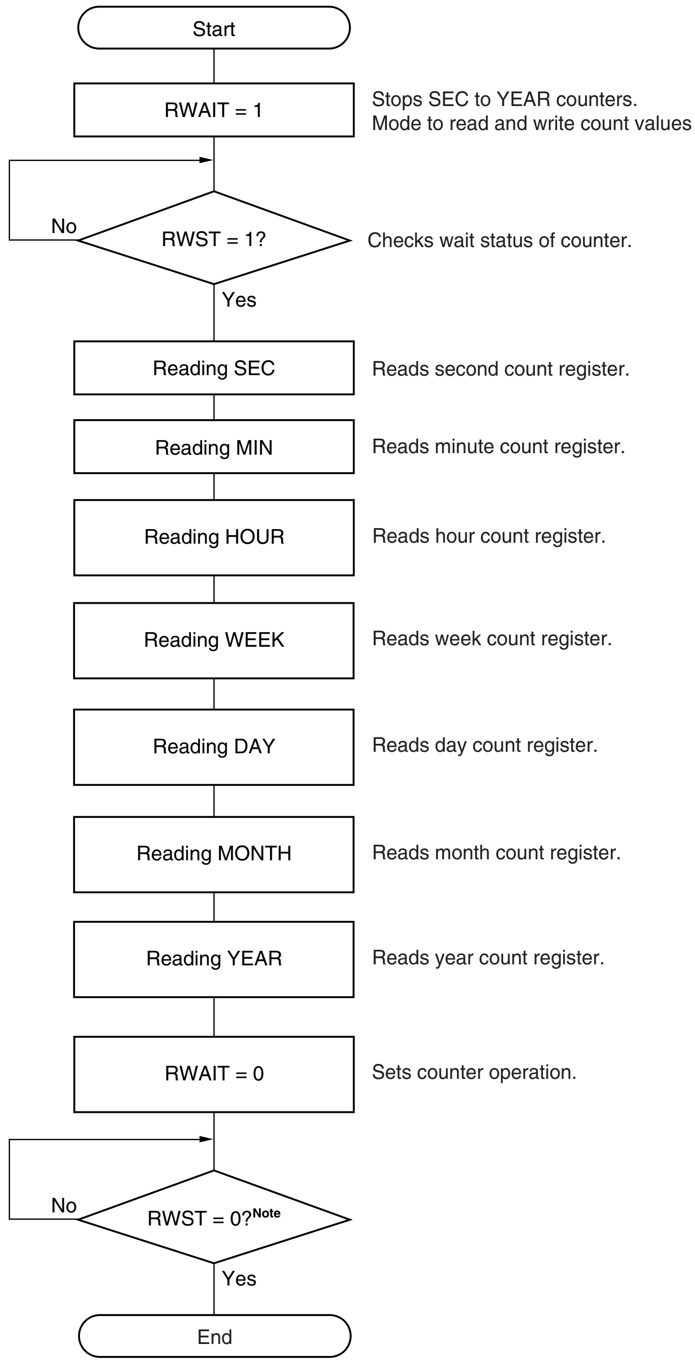
Figure 9-19. Procedure for Shifting to STOP Mode After Setting RTCE to 1



9.4.3 Reading/writing real-time counter

Read or write the counter after setting 1 to RWAIT first.

Figure 9-20. Procedure for Reading Real-Time Counter

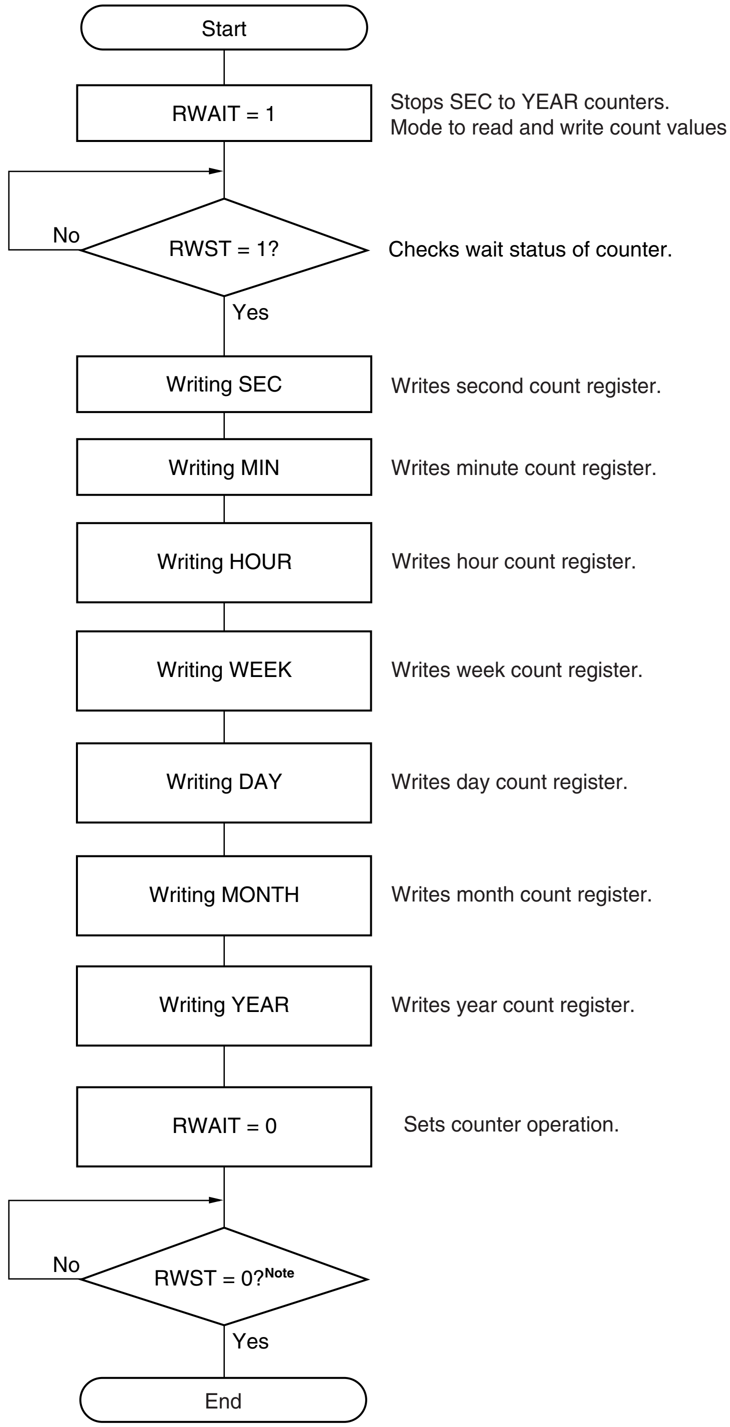


Note Be sure to confirm that RWST = 0 before setting STOP mode.

Caution Complete the series of operations of setting RWAIT to 1 to clearing RWAIT to 0 within 1 second.

Remark SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR may be read in any sequence.
All the registers do not have to be set and only some registers may be read.

Figure 9-21. Procedure for Writing Real-Time Counter



Note Be sure to confirm that RWST = 0 before setting STOP mode.

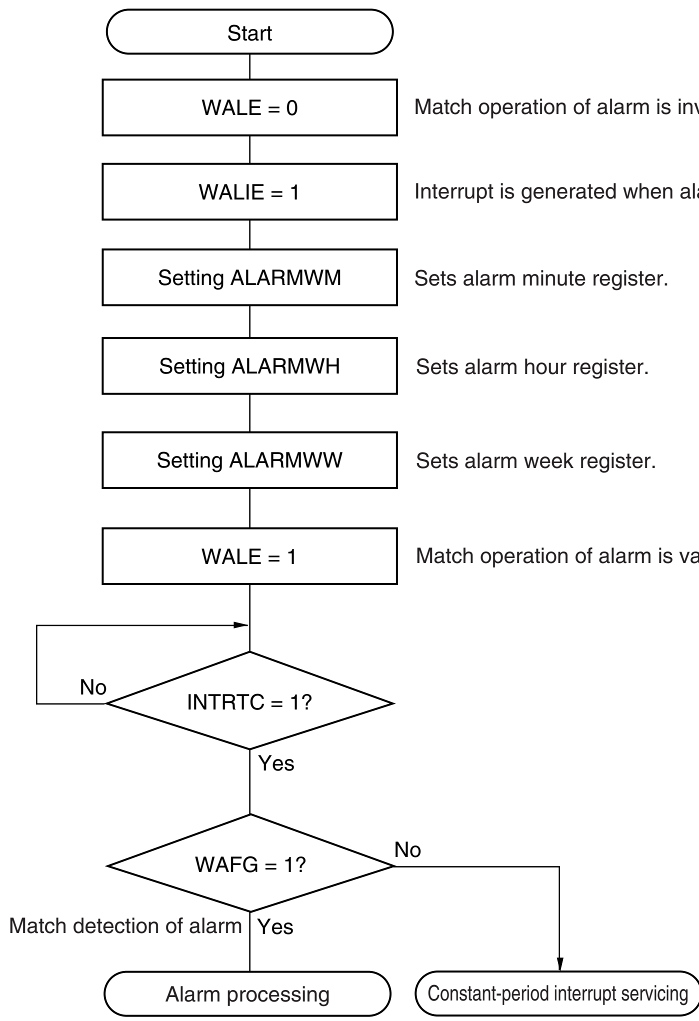
Caution Complete the series of operations of setting RWAIT to 1 to clearing RWAIT to 0 within 1 second.

Remark SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR may be written in any sequence.
All the registers do not have to be set and only some registers may be written.

9.4.4 Setting alarm of real-time counter

Set time of alarm after setting 0 to WALE first.

Figure 9-22. Alarm Setting Procedure



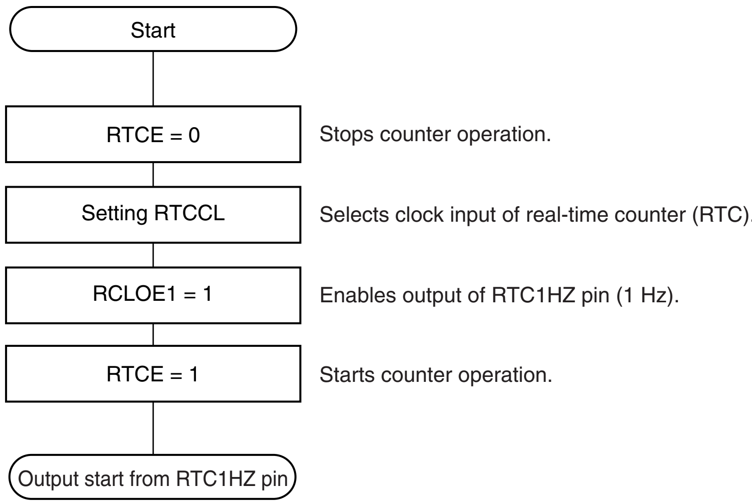
Remarks 1. ALARMWM, ALARMWH, and ALARMWW may be written in any sequence.

2. Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

<R> 9.4.5 1 Hz output of real-time counter

Set 1 Hz output after setting 0 to RTCE first.

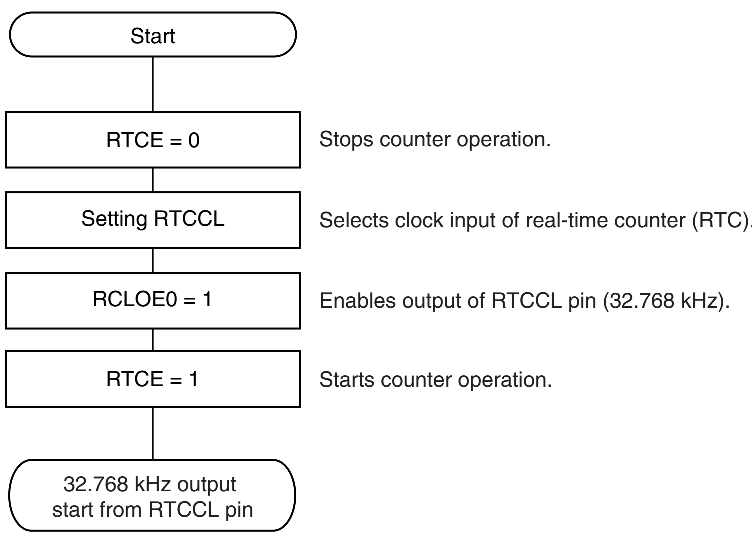
Figure 9-23. 1 Hz Output Setting Procedure



<R> 9.4.6 32.768 kHz output of real-time counter

Set 32.768 kHz output after setting 0 to RTCE first.

Figure 9-24. 32.768 kHz Output Setting Procedure

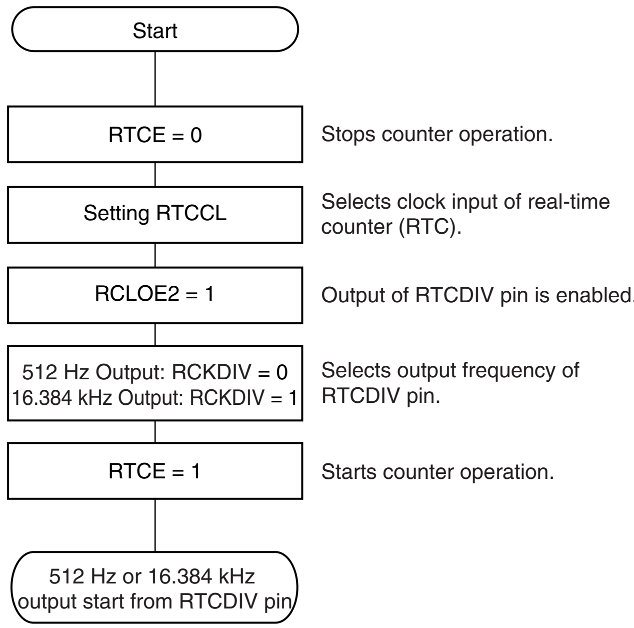


<R>

9.4.7 512 Hz, 16.384 kHz output of real-time counter

Set 512 Hz or 16.384 kHz output after setting 0 to RTCE first.

Figure 9-25. 512 Hz, 16.384 kHz output Setting Procedure



<R> **9.4.8 Example of watch error correction of real-time counter**

The watch can be corrected with high accuracy when it is slow or fast, by setting a value to the watch error correction register.

Example of calculating the correction value

The correction value used when correcting the count value of the sub-count register (RSUBC) is calculated by using the following expression.

Set DEV to 0 when the correction range is −63.1 ppm or less, or 63.1 ppm or more.

(When DEV = 0)

Correction value[Note] = Number of correction counts in 1 minute ÷ 3 = (Oscillation frequency ÷ Target frequency − 1) × 32768 × 60 ÷ 3

(When DEV = 1)

Correction value[Note] = Number of correction counts in 1 minute = (Oscillation frequency ÷ Target frequency − 1) × 32768 × 60

Note The correction value is the watch error correction value calculated by using bits 6 to 0 of the watch error correction register (SUBCUD).

(When F6 = 0) Correction value = {(F5, F4, F3, F2, F1, F0) − 1} × 2

(When F6 = 1) Correction value = − {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2

When (F6, F5, F4, F3, F2, F1, F0) is (*, 0, 0, 0, 0, 0, *), watch error correction is not performed. "*" is 0 or 1.

/F5 to /F0 are bit-inverted values (000011 when 111100).

Remarks 1. The correction value is 2, 4, 6, 8, … 120, 122, 124 or −2, −4, −6, −8, … −120, −122, −124.

2. The oscillation frequency is the input clock (f_{RTC}) value of the real-time counter (RTC).
It can be calculated from the 32 kHz output frequency of the RTCCL pin or the output frequency of the RTC1HZ pin × 32768 when the watch error correction register is set to its initial value (00H).

3. The target frequency is the frequency resulting after correction performed by using the watch error correction register.

Correction example <1>

Example of correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz − 131.2 ppm)

[Measuring the oscillation frequency]

The oscillation frequency[Note] of each product is measured by outputting about 32 kHz from the RTCCL pin or outputting about 1 Hz from the RTC1HZ pin when the watch error correction register is set to its initial value (00H).

Note See **9.4.5 1 Hz output of real-time counter** for the setting procedure of outputting about 1 Hz from the RTC1HZ pin, and **9.4.6 32.768 kHz output of real-time counter** for the setting procedure of outputting about 32 kHz from the RTCCL pin.

[Calculating the correction value]

(When the output frequency from the RTCCL pin is 32772.3 Hz)

If the target frequency is assumed to be 32768 Hz (32772.3 Hz − 131.2 ppm), the correction range for −131.2 ppm is −63.1 ppm or less, so assume DEV to be 0.

The expression for calculating the correction value when DEV is 0 is applied.

Correction value = Number of correction counts in 1 minute ÷ 3
= (Oscillation frequency ÷ Target frequency − 1) × 32768 × 60 ÷ 3
= (32772.3 ÷ 32768 − 1) × 32768 × 60 ÷ 3
= 86

[Calculating the values to be set to (F6 to F0)]

(When the correction value is 86)

If the correction value is 0 or more (when delaying), assume F6 to be 0.

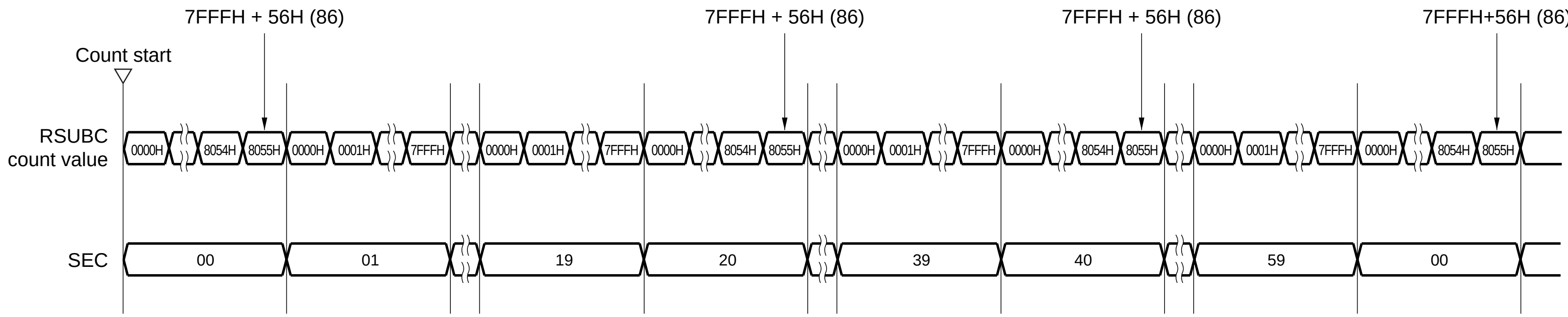
Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

{ (F5, F4, F3, F2, F1, F0) − 1} × 2 = 86
(F5, F4, F3, F2, F1, F0) = 44
(F5, F4, F3, F2, F1, F0) = (1, 0, 1, 1, 0, 0)

Consequently, when correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz − 131.2 ppm), setting the correction register such that DEV is 0 and the correction value is 86 (bits 6 to 0 of SUBCUD: 0101100) results in 32768 Hz (0 ppm).

Figure 9-26 shows the operation when (DEV, F6, F5, F4, F3, F2, F1, F0) is (0, 0, 1, 0, 1, 1, 0, 0).

Figure 9-26. Operation when (DEV, F6, F5, F4, F3, F2, F1, F0) = (0, 0, 1, 0, 1, 1, 0, 0)



Correction example <2>

Example of correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm)

[Measuring the oscillation frequency]

The oscillation frequency[Note] of each product is measured by outputting about 32 kHz from the RTCCL pin or outputting about 1 Hz from the RTC1HZ pin when the watch error correction register is set to its initial value (00H).

Note See **9.4.5 1 Hz output of real-time counter** for the setting procedure of outputting about 1 Hz from the RTC1HZ pin, and **9.4.6 32.768 kHz output of real-time counter** for the setting procedure of outputting about 32 kHz from the RTCCL pin.

[Calculating the correction value]

(When the output frequency from the RTCCL pin is 0.9999817 Hz)

Oscillation frequency = 32768 × 0.9999817 ≈ 32767.4 Hz

Assume the target frequency to be 32768 Hz (32767.4 Hz + 18.3 ppm) and DEV to be 1.

The expression for calculating the correction value when DEV is 1 is applied.

Correction value = Number of correction counts in 1 minute
= (Oscillation frequency ÷ Target frequency − 1) × 32768 × 60
= (32767.4 ÷ 32768 − 1) × 32768 × 60
= −36

[Calculating the values to be set to (F6 to F0)]

(When the correction value is −36)

If the correction value is 0 or less (when speeding up), assume F6 to be 1.

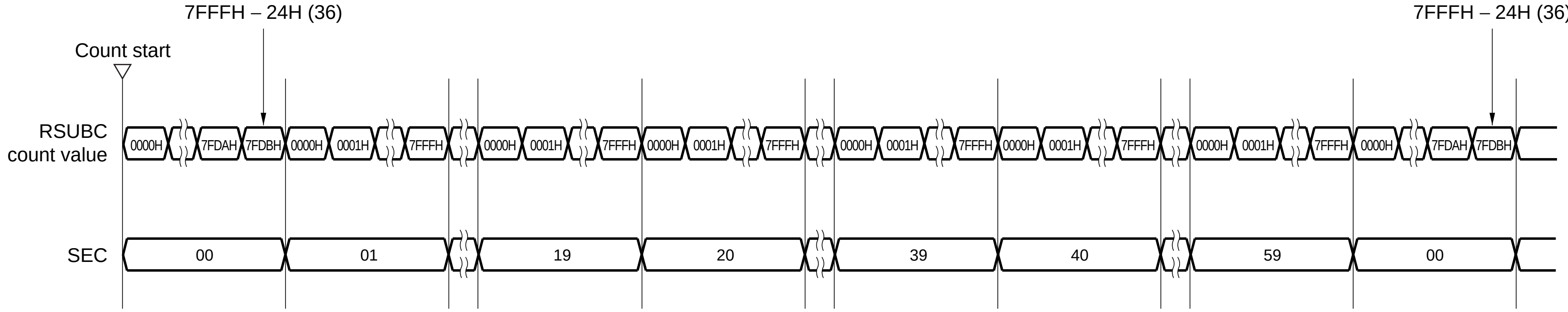
Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

− {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2	= −36
(/F5, /F4, /F3, /F2, /F1, /F0)	= 17
(/F5, /F4, /F3, /F2, /F1, /F0)	= (0, 1, 0, 0, 0, 1)
(F5, F4, F3, F2, F1, F0)	= (1, 0, 1, 1, 1, 0)

Consequently, when correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm), setting the correction register such that DEV is 1 and the correction value is −36 (bits 6 to 0 of SUBCUD: 1101110) results in 32768 Hz (0 ppm).

Figure 9-27 shows the operation when (DEV, F6, F5, F4, F3, F2, F1, F0) is (1, 1, 1, 0, 1, 1, 1, 0).

Figure 9-27. Operation when (DEV, F6, F5, F4, F3, F2, F1, F0) = (1, 1, 1, 0, 1, 1, 1, 0)



CHAPTER 10 WATCHDOG TIMER

10.1 Functions of Watchdog Timer

The watchdog timer operates on the internal low-speed oscillation clock.

The watchdog timer is used to detect an inadvertent program loop. If a program loop is detected, an internal reset signal is generated.

Program loop is detected in the following cases.

- If the watchdog timer counter overflows
- If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
- If data other than “ACH” is written to WDTE
- If data is written to WDTE during a window close period
- If the instruction is fetched from an area not set by the IMS and IXS registers (detection of an invalid check while the CPU hangs up)
- If the CPU accesses an area that is not set by the IMS and IXS registers (excluding FB00H to FFFFH) by executing a read/write instruction (detection of an abnormal access during a CPU program loop)

When a reset occurs due to the watchdog timer, bit 4 (WDTRF) of the reset control flag register (RESF) is set to 1. For details of RESF, see **CHAPTER 24 RESET FUNCTION**.

10.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 10-1. Configuration of Watchdog Timer

Item	Configuration
Control register	Watchdog timer enable register (WDTE)

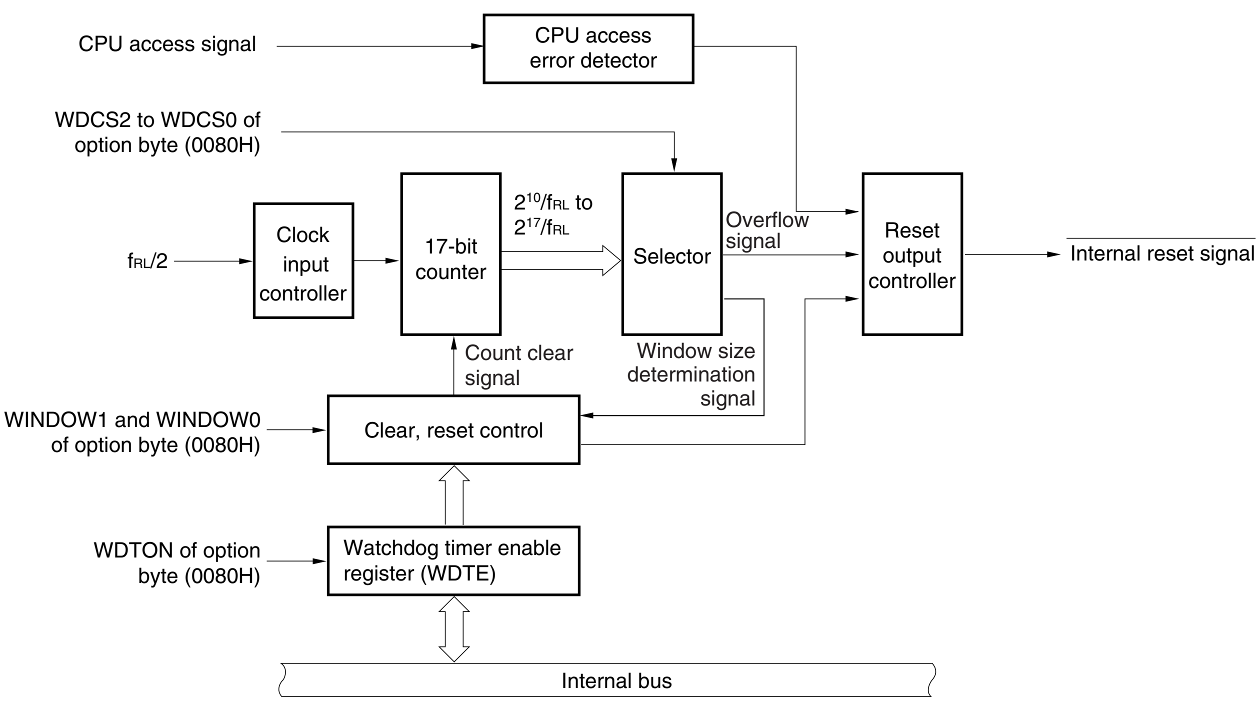
How the counter operation is controlled, overflow time, and window open period are set by the option byte.

Table 10-2. Setting of Option Bytes and Watchdog Timer

Setting of Watchdog Timer	Option Byte (0080H)
Window open period	Bits 6 and 5 (WINDOW1, WINDOW0)
Controlling counter operation of watchdog timer	Bit 4 (WDTON)
Overflow time of watchdog timer	Bits 3 to 1 (WDCS2 to WDCS0)

Remark For the option byte, see **CHAPTER 27 OPTION BYTE**.

Figure 10-1. Block Diagram of Watchdog Timer



10.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

(1) Watchdog timer enable register (WDTE)

Writing ACH to WDTE clears the watchdog timer counter and starts counting again.

This register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 9AH or 1AH[Note].

Figure 10-2. Format of Watchdog Timer Enable Register (WDTE)

Address: FF99H After reset: 9AH/1AH[Note] R/W

Symbol	7	6	5	4	3	2	1	0
WDTE								

Note The WDTE reset value differs depending on the WDTON setting value of the option byte (0080H). To operate watchdog timer, set WDTON to 1.

WDTON Setting Value	WDTE Reset Value
0 (watchdog timer count operation disabled)	1AH
1 (watchdog timer count operation enabled)	9AH

Cautions 1. If a value other than ACH is written to WDTE, an internal reset signal is generated. If the source clock to the watchdog timer is stopped, however, an internal reset signal is generated when the source clock to the watchdog timer resumes operation.

2. If a 1-bit memory manipulation instruction is executed for WDTE, an internal reset signal is generated. If the source clock to the watchdog timer is stopped, however, an internal reset signal is generated when the source clock to the watchdog timer resumes operation.

3. The value read from WDTE is 9AH/1AH (this differs from the written value (ACH)).

10.4 Operation of Watchdog Timer

10.4.1 Controlling operation of watchdog timer

1. When the watchdog timer is used, its operation is specified by the option byte (0080H).

    - Enable counting operation of the watchdog timer by setting bit 4 (WDTON) of the option byte (0080H) to 1 (the counter starts operating after a reset release) (for details, see **CHAPTER 27**).

WDTON	Operation Control of Watchdog Timer Counter/Illegal Access Detection
0	Counter operation disabled (counting stopped after reset), illegal access detection operation disabled
1	Counter operation enabled (counting started after reset), illegal access detection operation enabled

    - Set an overflow time by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (0080H) (for details, see **10.4.2** and **CHAPTER 27**).
    - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (0080H) (for details, see **10.4.3** and **CHAPTER 27**).

2. After a reset release, the watchdog timer starts counting.
3. By writing "ACH" to WDTE after the watchdog timer starts counting and before the overflow time set by the option byte, the watchdog timer is cleared and starts counting again.
4. After that, write WDTE the second time or later after a reset release during the window open period. If WDTE is written during a window close period, an internal reset signal is generated.
5. If the overflow time expires without "ACH" written to WDTE, an internal reset signal is generated.
   A internal reset signal is generated in the following cases.

    - If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
    - If data other than "ACH" is written to WDTE
    - If the instruction is fetched from an area not set by the IMS and IXS registers (detection of an invalid check during a CPU program loop)
    - If the CPU accesses an area not set by the IMS and IXS registers (excluding FB00H to FFFFH) by executing a read/write instruction (detection of an abnormal access during a CPU program loop)

Cautions 1. The first writing to WDTE after a reset release clears the watchdog timer, if it is made before the overflow time regardless of the timing of the writing, and the watchdog timer starts counting again.

2. If the watchdog timer is cleared by writing "ACH" to WDTE, the actual overflow time may be different from the overflow time set by the option byte by up to 2/f_{RL} seconds.

3. The watchdog timer can be cleared immediately before the count value overflows (FFFFH).

Cautions 4. The operation of the watchdog timer in the HALT and STOP modes differs as follows depending on the set value of bit 0 (LSROSC) of the option byte.

	LSROSC = 0 (Internal Low-Speed Oscillator Can Be Stopped by Software)	LSROSC = 1 (Internal Low-Speed Oscillator Cannot Be Stopped)
In HALT mode	Watchdog timer operation stops.	Watchdog timer operation continues.
In STOP mode		

If LSROSC = 0, the watchdog timer resumes counting after the HALT or STOP mode is released. At this time, the counter is not cleared to 0 but starts counting from the value at which it was stopped.

If oscillation of the internal low-speed oscillator is stopped by setting LSRSTOP (bit 1 of the internal oscillation mode register (RCM) = 1) when LSROSC = 0, the watchdog timer stops operating. At this time, the counter is not cleared to 0.

5. The watchdog timer continues its operation during self-programming and EEPROM emulation of the flash memory. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

10.4.2 Setting overflow time of watchdog timer

Set the overflow time of the watchdog timer by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (0080H).

If an overflow occurs, an internal reset signal is generated. The present count is cleared and the watchdog timer starts counting again by writing “ACH” to WDTE during the window open period before the overflow time.

The following overflow time is set.

Table 10-3. Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer
0	0	0	$2^{10}/f_{RL}$ (3.88 ms)
0	0	1	$2^{11}/f_{RL}$ (7.76 ms)
0	1	0	$2^{12}/f_{RL}$ (15.52 ms)
0	1	1	$2^{13}/f_{RL}$ (31.03 ms)
1	0	0	$2^{14}/f_{RL}$ (62.06 ms)
1	0	1	$2^{15}/f_{RL}$ (124.12 ms)
1	1	0	$2^{16}/f_{RL}$ (248.24 ms)
1	1	1	$2^{17}/f_{RL}$ (496.48 ms)

Cautions 1. The combination of WDCS2 = WDCS1 = WDCS0 = 0 and WINDOW1 = WINDOW0 = 0 is prohibited.

2. The watchdog timer continues its operation during self-programming and EEPROM emulation of the flash memory. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

Remarks 1. f_{RL}: Internal low-speed oscillation clock frequency

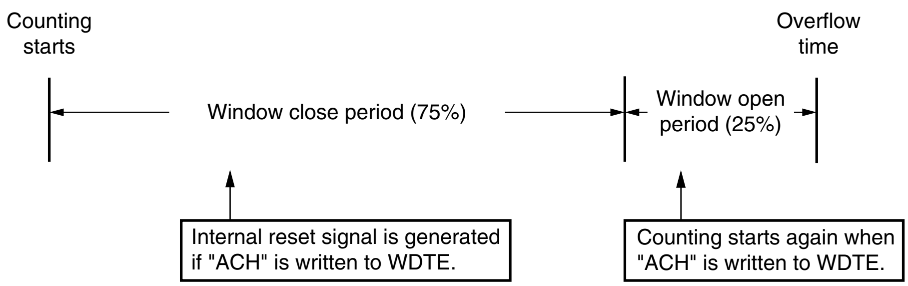
2. ( ): f_{RL} = 264 kHz (MAX.)

10.4.3 Setting window open period of watchdog timer

Set the window open period of the watchdog timer by using bits 6 and 5 (WINDOW1, WINDOW0) of the option byte (0080H). The outline of the window is as follows.

- If "ACH" is written to WDTE during the window open period, the watchdog timer is cleared and starts counting again.
- Even if "ACH" is written to WDTE during the window close period, an abnormality is detected and an internal reset signal is generated.

Example: If the window open period is 25%



Caution The first writing to WDTE after a reset release clears the watchdog timer, if it is made before the overflow time regardless of the timing of the writing, and the watchdog timer starts counting again.

The window open period to be set is as follows.

Table 10-4. Setting Window Open Period of Watchdog Timer

WINDOW1	WINDOW0	Window Open Period of Watchdog Timer
0	0	25%
0	1	50%
1	0	75%
1	1	100%

Cautions 1. The combination of WDCS2 = WDCS1 = WDCS0 = 0 and WINDOW1 = WINDOW0 = 0 is prohibited.

2. The watchdog timer continues its operation during self-programming and EEPROM emulation of the flash memory. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

<R>

Remark If the overflow time is set to $2^{10}/f_{RL}$, the window close time and open time are as follows.

(2.6 V≤V_{DD}≤5.5 V)

	Setting of Window Open Period			
	25%	50%	75%	100%
Window close time	0 to 3.56 ms	0 to 2.37 ms	0 to 1.19 ms	None
Window open time	3.56 to 3.88 ms	2.37 to 3.88 ms	1.19 to 3.88 ms	0 to 3.88 ms

<When window open period is 25%>

- Overflow time:
  $2^{10}/f_{RL}$ (MAX.) = 2^{10}/264 kHz (MAX.) = 3.88 ms
- Window close time:
  0 to $2^{10}/f_{RL}$ (MIN.) × (1 − 0.25) = 0 to 2^{10}/216 kHz (MIN.) × 0.75 = 0 to 3.56 ms
- Window open time:
  $2^{10}/f_{RL}$ (MIN.) × (1 − 0.25) to $2^{10}/f_{RL}$ (MAX.) = 2^{10}/216 kHz (MIN.) × 0.75 to 2^{10}/264 kHz (MAX.)
  = 3.56 to 3.88 ms

CHAPTER 11 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

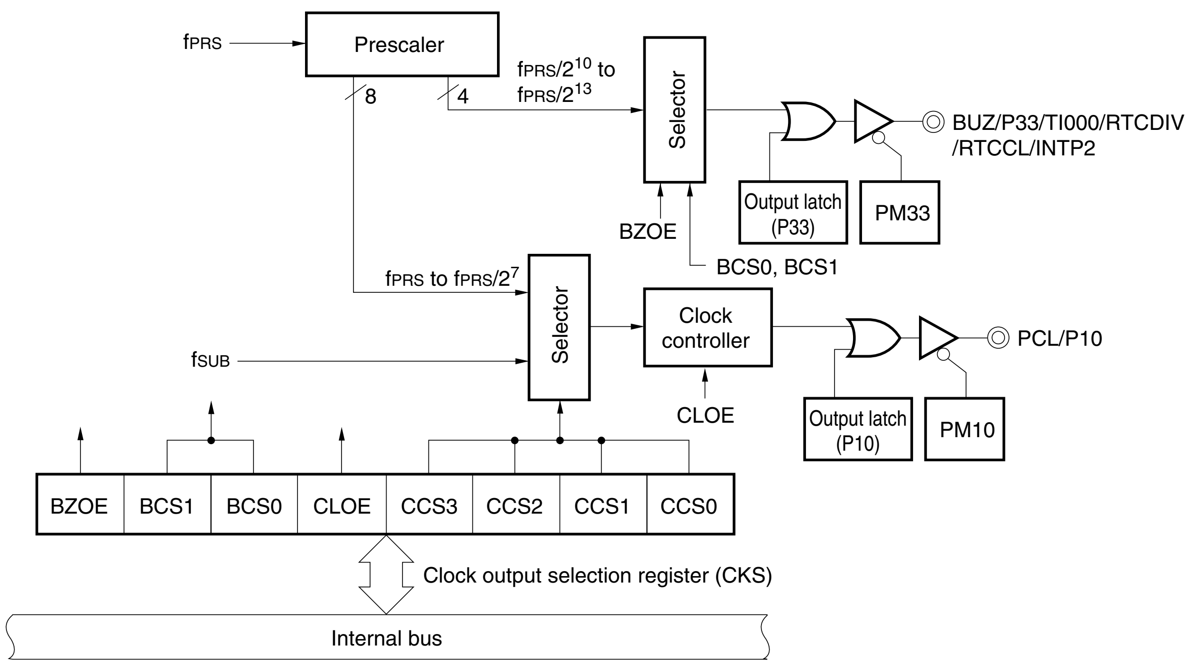
11.1 Functions of Clock Output/Buzzer Output Controller

The clock output controller is intended for carrier output during remote controlled transmission and clock output for supply to peripheral ICs. The clock selected with the clock output selection register (CKS) is output.

In addition, the buzzer output is intended for square-wave output of buzzer frequency selected with CKS.

Figure 11-1 shows the block diagram of clock output/buzzer output controller.

Figure 11-1. Block Diagram of Clock Output/Buzzer Output Controller



11.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 11-1. Configuration of Clock Output/Buzzer Output Controller

Item	Configuration
Control registers	Clock output selection register (CKS) Port mode register 3 (PM3) Port register 3 (P3) Port mode register 1 (PM1) Port register 1 (P1)

11.3 Registers Controlling Clock Output/Buzzer Output Controller

The following two registers are used to control the clock output/buzzer output controller.

- Clock output selection register (CKS)
- Port mode register 3 (PM3)
- Port mode register 1 (PM1)

(1) Clock output selection register (CKS)

This register sets output enable/disable for clock output (PCL) and for the buzzer frequency output (BUZ), and sets the output clock.

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

Reset signal generation clears CKS to 00H.

Figure 11-2. Format of Clock Output Selection Register (CKS)

Address: FF40H After reset: 00H R/W

Symbol	<7>	6	5	<4>	3	2	1	0
CKS	BZOE	BCS1	BCS0	CLOE	CCS3	CCS2	CCS1	CCS0

BZOE	BUZ output enable/disable specification
0	Clock division circuit operation stopped. BUZ fixed to low level.
1	Clock division circuit operation enabled. BUZ output enabled.

BCS1	BCS0	BUZ output clock selection		
			f_{PRS} = 5 MHz	f_{PRS} = 10 MHz
0	0	$f_{PRS}/2^{10}$	4.88 kHz	9.77 kHz
0	1	$f_{PRS}/2^{11}$	2.44 kHz	4.88 kHz
1	0	$f_{PRS}/2^{12}$	1.22 kHz	2.44 kHz
1	1	$f_{PRS}/2^{13}$	0.61 kHz	1.22 kHz

CLOE	PCL output enable/disable specification
0	Clock division circuit operation stopped. PCL fixed to low level.
1	Clock division circuit operation enabled. PCL output enabled.

CCS3	CCS2	CCS1	CCS0	PCL output clock selection^Note 1			
					f_{SUB} = 32.768 kHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz
0	0	0	0	f_{PRS}^Note 2	–	5 MHz	10 MHz
0	0	0	1	$f_{PRS}/2$		2.5 MHz	5 MHz
0	0	1	0	$f_{PRS}/2^2$		1.25 MHz	2.5 MHz
0	0	1	1	$f_{PRS}/2^3$		625 kHz	1.25 MHz
0	1	0	0	$f_{PRS}/2^4$		312.5 kHz	625 kHz
0	1	0	1	$f_{PRS}/2^5$		156.25 kHz	312.5 kHz
0	1	1	0	$f_{PRS}/2^6$		78.125 kHz	156.25 kHz
0	1	1	1	$f_{PRS}/2^7$		39.062 kHz	78.125 kHz
1	0	0	0	f_{SUB}	32.768 kHz	–	
Other than above				Setting prohibited			

Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{XH}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.
- V_{DD} = 2.7 to 5.5 V: $f_{PRS} \leq$ 10 MHz
- V_{DD} = 1.8 to 2.7 V: $f_{PRS} \leq$ 5 MHz

2. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{RH}) (XSEL = 0), when 1.8 V $\leq V_{DD} <$ 2.7 V, the setting of CCS3 = CCS2 = CCS1 = CCS0 = 0 (output clock of PCL: f_{PRS}) is prohibited.

Cautions 1. Set BCS1 and BCS0 when the buzzer output operation is stopped (BZOE = 0).

2. Set CCS3 to CCS0 while the clock output operation is stopped (CLOE = 0).

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. f_{SUB}: Subsystem clock frequency

(2) Port mode register 3 (PM3)

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

When using the P33/TI000/RTCDIV/RTCCL/BUZ/INTP2 pin for buzzer output, clear PM33 and the output latches of P33 to 0.

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

Reset signal generation sets PM3 to FFH.

Figure 11-3. Format of Port Mode Register 3 (PM3)

Address: FF23H After reset: FFH R/W

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

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

(3) Port mode register 1 (PM1)

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

When using the P10/PCL pin for clock output, clear PM10 and the output latches of P10 to 0.

PM1 is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PM1 to FFH.

Figure 11-4. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

11.4 Operations of Clock Output/Buzzer Output Controller

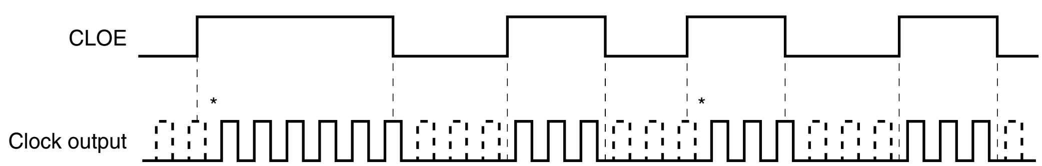
11.4.1 Operation as clock output

The clock pulse is output as the following procedure.

<1> Select the clock pulse output frequency with bits 0 to 3 (CCS0 to CCS3) of the clock output selection register (CKS) (clock pulse output in disabled status).
<2> Set bit 4 (CLOE) of CKS to 1 to enable clock output.

Remark The clock output controller is designed not to output pulses with a small width during output enable/disable switching of the clock output. As shown in Figure 11-5, be sure to start output from the low period of the clock (marked with * in the figure). When stopping output, do so after the high-level period of the clock.

Figure 11-5. Remote Control Output Application Example



11.4.2 Operation as buzzer output

The buzzer frequency is output as the following procedure.

<1> Select the buzzer output frequency with bits 5 and 6 (BCS0, BCS1) of the clock output selection register (CKS) (buzzer output in disabled status).
<2> Set bit 7 (BZOE) of CKS to 1 to enable buzzer output.

CHAPTER 12 10-BIT SUCCESSIVE APPROXIMATION TYPE A/D CONVERTER (μPD78F048x and 78F049x only)

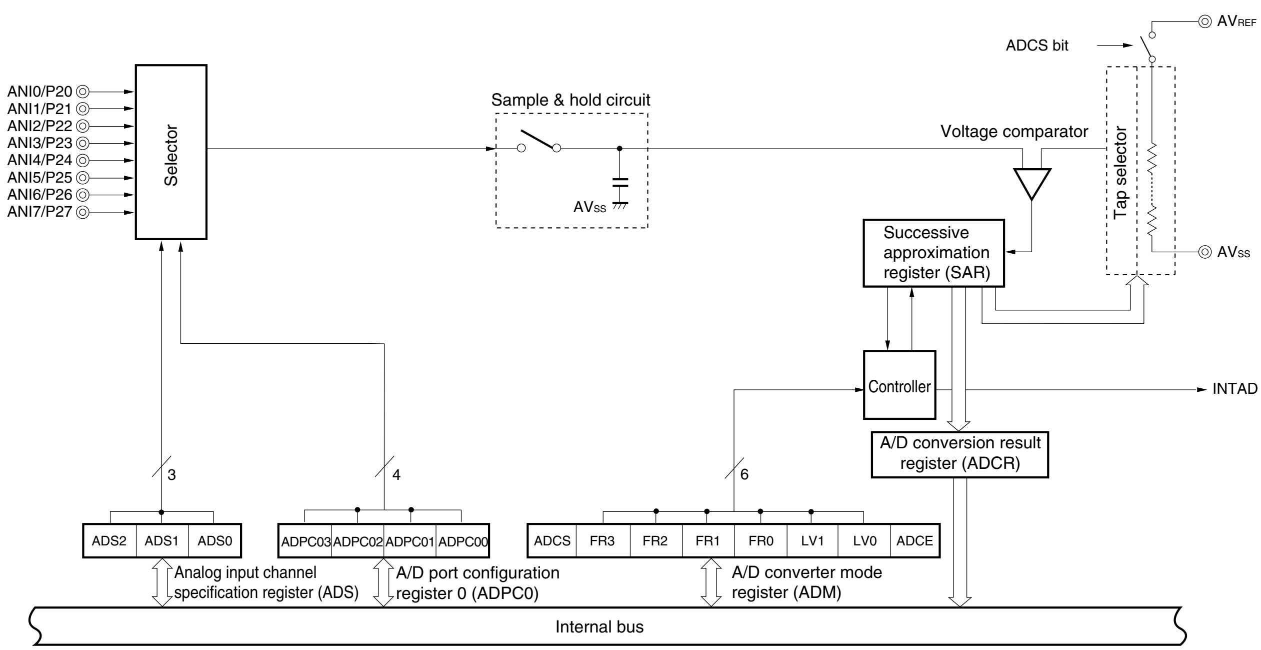
12.1 Function of 10-Bit Successive Approximation Type A/D Converter

The 10-bit successive approximation type A/D converter converts an analog input signal into a digital value, and consists of up to eight channels (ANI0 to ANI7) with a resolution of 10 bits.

The A/D converter has the following function.

- **10-bit resolution A/D conversion**
  10-bit resolution A/D conversion is carried out repeatedly for one analog input channel selected from ANI0 to ANI7. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

Figure 12-1. Block Diagram of 10-Bit Successive Approximation type A/D Converter



12.2 Configuration of 10-Bit Successive Approximation Type A/D Converter

The 10-bit successive approximation type A/D converter includes the following hardware.

(1) ANI0 to ANI7 pins

These are the 8-channel analog input pins of the 10-bit successive approximation type A/D converter. They input analog signals to be converted into digital signals. Pins other than the one selected as the analog input pin can be used as I/O port pins or segment output pins (μPD78F048x only).

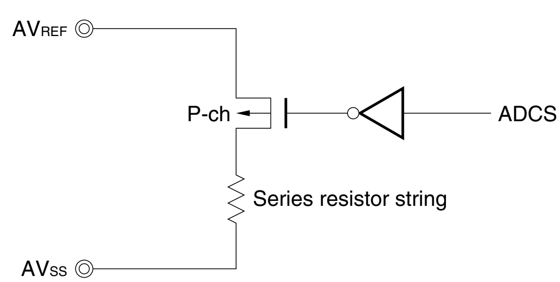
(2) Sample & hold circuit

The sample & hold circuit samples the input voltage of the analog input pin selected by the selector when A/D conversion is started, and holds the sampled voltage value during A/D conversion.

(3) Series resistor string

The series resistor string is connected between AV_{REF} and AV_{SS}, and generates a voltage to be compared with the sampled voltage value.

Figure 12-2. Circuit Configuration of Series Resistor String



(4) Voltage comparator

The voltage comparator compares the sampled voltage value and the output voltage of the series resistor string.

(5) Successive approximation register (SAR)

This register converts the result of comparison by the voltage comparator, starting from the most significant bit (MSB).

When the voltage value is converted into a digital value down to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register are transferred to the A/D conversion result register (ADCR).

(6) 10-bit A/D conversion result register (ADCR)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCR register holds the A/D conversion result in its higher 10 bits (the lower 6 bits are fixed to 0).

(7) 8-bit A/D conversion result register (ADCRH)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCRH register stores the higher 8 bits of the A/D conversion result.

Caution When data is read from ADCR and ADCRH, a wait cycle is generated. Do not read data from ADCR and ADCRH when the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped. For details, see CHAPTER 34 CAUTIONS FOR WAIT.

(8) Controller

This circuit controls the conversion time of an input analog signal that is to be converted into a digital signal, as well as starting and stopping of the conversion operation. When A/D conversion has been completed, this controller generates INTAD.

(9) AV_{REF} pin

This pin inputs an analog power/reference voltage to the A/D converter. When using at least one port of port 2 as a digital port or for segment output, set it to the same potential as the V_{DD} pin.

The signal input to ANI0 to ANI7 is converted into a digital signal, based on the voltage applied across AV_{REF} and AV_{SS}.

(10) AV_{SS} pin

This is the ground potential pin of the A/D converter. Always use this pin at the same potential as that of the V_{SS} pin even when the A/D converter is not used.

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

This register is used to set the conversion time of the analog input signal to be converted, and to start or stop the conversion operation.

(12) A/D port configuration register 0 (ADPC0)

This register switches the ANI0/P20 to ANI7/P27 pins to analog input (analog input of 16-bit ΔΣ type A/D converter or analog input of 10-bit successive approximation type A/D converter) or digital I/O of port.

(13) Analog input channel specification register (ADS)

This register is used to specify the port that inputs the analog voltage to be converted into a digital signal.

(14) Port mode register 2 (PM2)

This register switches the ANI0/P20 to ANI7/P27 pins to input or output.

(15) Port function register 2 (PF2) (μPD78F048x only)

This register switches the ANI0/P20 to ANI7/P27 pins to I/O of port, analog input of A/D converter, or segment output.

12.3 Registers Used in 10-Bit Successive Approximation Type A/D Converter

The A/D converter uses the following seven registers.

- A/D converter mode register (ADM)
- A/D port configuration register 0 (ADPC0)
- Analog input channel specification register (ADS)
- Port function register 2 (PF2) (μPD78F048x only)
- Port mode register 2 (PM2)
- 10-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)

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

This register sets the conversion time for analog input to be A/D converted, and starts/stops conversion.
ADM can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 12-3. Format of A/D Converter Mode Register (ADM)

Address: FF8DH After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM	ADCS	FR3[Note 1]	FR2[Note 1]	FR1[Note 1]	FR0[Note 1]	LV1[Note 1]	LV0[Note 1]	ADCE

ADCS	A/D conversion operation control
0	Stops conversion operation
1	Enables conversion operation

ADCE	Comparator operation control[Note 2]
0	Stops comparator operation
1	Enables comparator operation

Notes 1. For details of FR3 to FR0, LV1, LV0, and A/D conversion, see **Table 12-2 A/D Conversion Time Selection**.

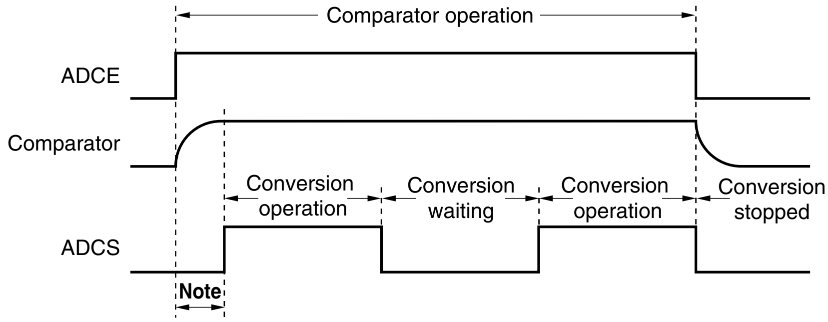
2. The operation of the comparator is controlled by ADCS and ADCE, and it takes 1 μs from operation start to operation stabilization. Therefore, when ADCS is set to 1 after 1 μs or more has elapsed from the time ADCE is set to 1, the conversion result at that time has priority over the first conversion result. Otherwise, ignore data of the first conversion.

Table 12-1. Settings of ADCS and ADCE

ADCS	ADCE	A/D Conversion Operation
0	0	Stop status (DC power consumption path does not exist)
0	1	Conversion waiting mode (comparator operation, only comparator consumes power)
1	0	Conversion mode (comparator operation stopped[Note])
1	1	Conversion mode (comparator operation)

Note Ignore data of the first conversion.

Figure 12-4. Timing Chart When Comparator Is Used



Note To stabilize the internal circuit, the time from the rising of the ADCE bit to the falling of the ADCS bit must be 1 μs or longer.

Cautions 1. A/D conversion must be stopped before rewriting bits FR0 to FR3, LV1, and LV0 to values other than the identical data.

2. If data is written to ADM, a wait cycle is generated. Do not write data to ADM when the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped. For details, see CHAPTER 34 CAUTIONS FOR WAIT.

Table 12-2. A/D Conversion Time Selection

(1) 2.7 V ≤ AV_{REF} ≤ 5.5 V

A/D Converter Mode Register (ADM)						Conversion Time Selection				Conversion Clock (f_{AD})
FR3	FR2	FR1	FR0	LV1	LV0		f_{PRS} = 2 MHz	f_{PRS} = 8 MHz	f_{PRS} = 10 MHz	
1	×	×	×	0	0	352/f_{PRS}	Setting prohibited	44.0 μs	35.2 μs	f_{PRS}/16
0	0	0	0	0	0	264/f_{PRS}		33.0 μs	26.4 μs	f_{PRS}/12
0	0	0	1	0	0	176/f_{PRS}		22.0 μs	17.6 μs	f_{PRS}/8
0	0	1	0	0	0	132/f_{PRS}		16.5 μs	13.2 μs	f_{PRS}/6
0	0	1	1	0	0	88/f_{PRS}	44.0 μs	11.0 μs[Note]	8.8 μs[Note]	f_{PRS}/4
0	1	0	0	0	0	66/f_{PRS}	33.0 μs	8.3 μs[Note]	6.6 μs[Note]	f_{PRS}/3
0	1	0	1	0	0	44/f_{PRS}	22.0 μs	Setting prohibited	Setting prohibited	f_{PRS}/2
Other than above						Setting prohibited				

Note This can be set only when 4.0 V ≤ AV_{REF} ≤ 5.5 V.

(2) 2.3 V ≤ AV_{REF} < 2.7 V

A/D Converter Mode Register (ADM)						Conversion Time Selection				Conversion Clock (f_{AD})
FR3	FR2	FR1	FR0	LV1	LV0		f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 8 MHz	
0	0	0	0	0	1	480/f_{PRS}	Setting prohibited	Setting prohibited	60.0 μs	f_{PRS}/12
0	0	0	1	0	1	320/f_{PRS}		64.0 μs	40.0 μs	f_{PRS}/8
0	0	1	0	0	1	240/f_{PRS}		48.0 μs	30.0 μs	f_{PRS}/6
0	0	1	1	0	1	160/f_{PRS}		32.0 μs	Setting prohibited	f_{PRS}/4
0	1	0	0	0	1	120/f_{PRS}	60.0 μs	Setting prohibited		f_{PRS}/3
0	1	0	1	0	1	80/f_{PRS}	40.0 μs			f_{PRS}/2
Other than above						Setting prohibited				

<R> **Cautions 1. Set the conversion times with the following conditions.**

- **4.0 V ≤ AV_{REF} ≤ 5.5 V: f_{AD} = 0.6 to 3.6 MHz**
- **2.7 V ≤ AV_{REF} < 4.0 V: f_{AD} = 0.6 to 1.8 MHz**
- **2.3 V ≤ AV_{REF} < 2.7 V: f_{AD} = 0.6 to 1.48 MHz**

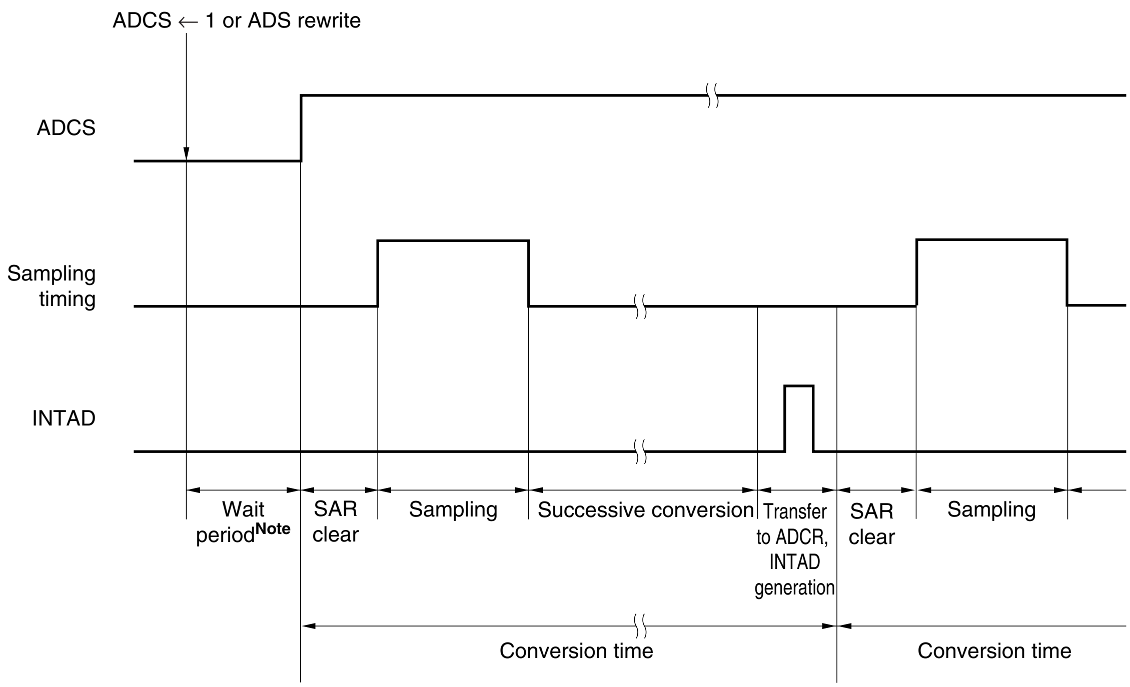
2. When rewriting FR3 to FR0, LV1, and LV0 to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

3. Change LV1 and LV0 from the default value, when 2.3 V ≤ AV_{REF} < 2.7 V.

4. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

Remark f_{PRS}: Peripheral hardware clock frequency

Figure 12-5. A/D Converter Sampling and A/D Conversion Timing



Note For details of wait period, see **CHAPTER 34 CAUTIONS FOR WAIT**.

(2) 10-bit A/D conversion result register (ADCR)

This register is a 16-bit register that stores the A/D conversion result. The lower 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register. The higher 8 bits of the conversion result are stored in FF07H and the lower 2 bits are stored in the higher 2 bits of FF06H.

ADCR can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 12-6. Format of 10-Bit A/D Conversion Result Register (ADCR)

Address: FF06H, FF07H After reset: 0000H R

Symbol	FF07H								FF06H							
ADCR											0	0	0	0	0	0

Cautions 1. When writing to the A/D converter mode register (ADM), analog input channel specification register (ADS), and A/D port configuration register 0 (ADPC0), the contents of ADCR may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC0. Using timing other than the above may cause an incorrect conversion result to be read.

2. If data is read from ADCR, a wait cycle is generated. Do not read data from ADCR when the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped. For details, see CHAPTER 34 CAUTIONS FOR WAIT.

(3) 8-bit A/D conversion result register (ADCRH)

This register is an 8-bit register that stores the A/D conversion result. The higher 8 bits of 10-bit resolution are stored.

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

Reset signal generation clears this register to 00H.

Figure 12-7. Format of 8-Bit A/D Conversion Result Register (ADCRH)

Address: FF07H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADCRH								

Cautions 1. When writing to the A/D converter mode register (ADM), analog input channel specification register (ADS), and A/D port configuration register 0 (ADPC0), the contents of ADCRH may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC0. Using timing other than the above may cause an incorrect conversion result to be read.

2. If data is read from ADCRH, a wait cycle is generated. Do not read data from ADCRH when the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped. For details, see CHAPTER 34 CAUTIONS FOR WAIT.

(4) Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted.

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

Reset signal generation clears this register to 00H.

Figure 12-8. Format of Analog Input Channel Specification Register (ADS)

Address: FF8EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	0	0	0	0	0	ADS2	ADS1	ADS0

ADS2	ADS1	ADS0	Analog input channel specification
0	0	0	ANI0
0	0	1	ANI1
0	1	0	ANI2
0	1	1	ANI3
1	0	0	ANI4
1	0	1	ANI5
1	1	0	ANI6
1	1	1	ANI7

Cautions 1. Be sure to clear bits 3 to 7 to “0”.

2. Set a channel to be used for A/D conversion in the input mode by using port mode register 2 (PM2).

3. Do not set a pin to be used as a digital I/O pin with ADPC with ADS.

4. A pin whose channel has been selected as a 10-bit successive approximation type A/D converter input must not be selected as a 16-bit ΔΣ type A/D converter input.

5. If data is written to ADS, a wait cycle is generated. Do not write data to ADS when the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped. For details, see CHAPTER 34 CAUTIONS FOR WAIT.

(5) A/D port configuration register 0 (ADPC0)

This register switches the ANI0/P20 to ANI7/P27 pins to analog input (analog input of 16-bit ΔΣ type A/D converter or analog input of 10-bit successive approximation type A/D converter) or digital I/O of port.

ADPC0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 08H.

Figure 12-9. Format of A/D Port Configuration Register 0 (ADPC0)

Address: FF8FH After reset: 08H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC0	0	0	0	0	ADPC03	ADPC02	ADPC01	ADPC00

<R>

<μPD78F048x>

ADPC03	ADPC02	ADPC01	ADPC00	Digital I/O (D)/analog input (A) switching							
				P27/ANI7/SEG32	P26/ANI6/SEG33	P25/ANI5/SEG34	P24/ANI4/SEG35	P23/ANI3/SEG36	P22/ANI2/SEG37	P21/ANI1/SEG38	P20/ANI0/SEG39
0	0	0	0	A	A	A	A	A	A	A	A
0	0	0	1	A	A	A	A	A	A	A	D
0	0	1	0	A	A	A	A	A	A	D	D
0	0	1	1	A	A	A	A	A	D	D	D
0	1	0	0	A	A	A	A	D	D	D	D
0	1	0	1	A	A	A	D	D	D	D	D
0	1	1	0	A	A	D	D	D	D	D	D
0	1	1	1	A	D	D	D	D	D	D	D
1	0	0	0	D	D	D	D	D	D	D	D
Other than above				Setting prohibited							

<μPD78F049x>

ADPC03	ADPC02	ADPC01	ADPC00	Digital I/O (D)/analog input (A: successive approximation type, Δ: ΔΣ type) switching							
				P27/ANI7/REF+	P26/ANI6/REF-	P25/ANI5/DS2+	P24/ANI4/DS2-	P23/ANI3/DS1+	P22/ANI2/DS1-	P21/ANI1/DS0+	P20/ANI0/DS0-
0	0	0	0	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ
0	0	0	1	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A	D
0	0	1	0	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	D	D
0	0	1	1	A/Δ	A/Δ	A/Δ	A/Δ	A	D	D	D
0	1	0	0	A/Δ	A/Δ	A/Δ	A/Δ	D	D	D	D
0	1	0	1	A	A	A	D	D	D	D	D
0	1	1	0	A	A	D	D	D	D	D	D
0	1	1	1	A	D	D	D	D	D	D	D
1	0	0	0	D	D	D	D	D	D	D	D
Other than above				Setting prohibited							

Cautions 1. Set the channel used for A/D conversion to the input mode by using port mode register 2 (PM2).

2. Do not set the pin set by ADPC0 as digital I/O by ADS, ADDS1, or ADDS0.

3. If data is written to ADPC0, a wait cycle is generated. Do not write data to ADPC0 when the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped. For details, see CHAPTER 34 CAUTIONS FOR WAIT.

4. If pins ANI0/P20/SEG39 to ANI7/P27/SEG32 are set to segment output pins via the PF2 register, output is set to segment output, regardless of the ADPC0 setting (for μPD78F048x only).

(6) Port mode register 2 (PM2)

When using the ANI0/P20 to ANI7/P27 pins for analog input port, set PM20 to PM27 to 1. The output latches of P20 to P27 at this time may be 0 or 1.

If PM20 to PM27 are set to 0, they cannot be used as analog input port pins.

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

Reset signal generation sets this register to FFH.

Figure 12-10. Format of Port Mode Register 2 (PM2)

Address: FF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20

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

ANI0/P20 to ANI7/P27 pins are as shown below depending on the settings of PF2, ADPC0, PM2, ADS, and ADDCTL0.

Table 12-3. Setting Functions of P20/ANI0 to P27/ANI7 Pins

(a) μPD78F048x

PF2	ADPC0	PM2	ADS	P20/SEG39/ANI0 to P27/SEG32/ANI7 Pins
Digital/Analog selection	Analog input selection	Input mode	Does not select ANI.	Analog input (not to be converted)
			Selects ANI.	Analog input (to be converted by successive approximation type A/D converter)
		Output mode	–	Setting prohibited
	Digital I/O selection	Input mode	–	Digital input
		Output mode	–	Digital output
SEG output selection	–	–	–	Segment output

(b) μPD78F049x

ADPC0	PM2	ADS	ADDCTL0	P20/ANI0/DS0- to P27/ANI7/REF+ Pins
Analog input selection	Input mode	Does not select ANI.	Does not select DSn±.	Analog input (not to be converted)
		Selects ANI.	Does not select DSn±.	Analog input (to be converted by successive approximation type A/D converter)
		Does not select ANI.	Selects DSn±.	Analog input (to be converted by ΔΣ type A/D converter)
		Selects ANI.	Selects DSn±.	Setting prohibited
	Output mode	–		Setting prohibited
Digital I/O selection	Input mode	–		Digital input
	Output mode	–		Digital output

12.4 10-Bit Successive Approximation Type A/D Converter Operations

12.4.1 Basic operations of A/D converter

<1> Set bit 0 (ADCE) of the A/D converter mode register (ADM) to 1 to start the operation of the comparator.
<2> Set channels for A/D conversion to analog input by using the A/D port configuration register (ADPC0) and set to input mode by using port mode register 2 (PM2).
<3> Set A/D conversion time by using bits 6 to 1 (FR3 to FR0, LV1, and LV0) of ADM.
<4> Select one channel for A/D conversion using the analog input channel specification register (ADS).
<5> Start the conversion operation by setting bit 7 (ADCS) of ADM to 1.
(<6> to <12> are operations performed by hardware.)
<6> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.
<7> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the sampled voltage is held until the A/D conversion operation has ended.
<8> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) AV_{REF} by the tap selector.
<9> The voltage difference between the series resistor string voltage tap and sampled voltage is compared by the voltage comparator. If the analog input is greater than (1/2) AV_{REF}, the MSB of SAR remains set to 1. If the analog input is smaller than (1/2) AV_{REF}, the MSB is reset to 0.
<10> Next, bit 8 of SAR is automatically set to 1, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 9, as described below.
- Bit 9 = 1: (3/4) AV_{REF}
- Bit 9 = 0: (1/4) AV_{REF}

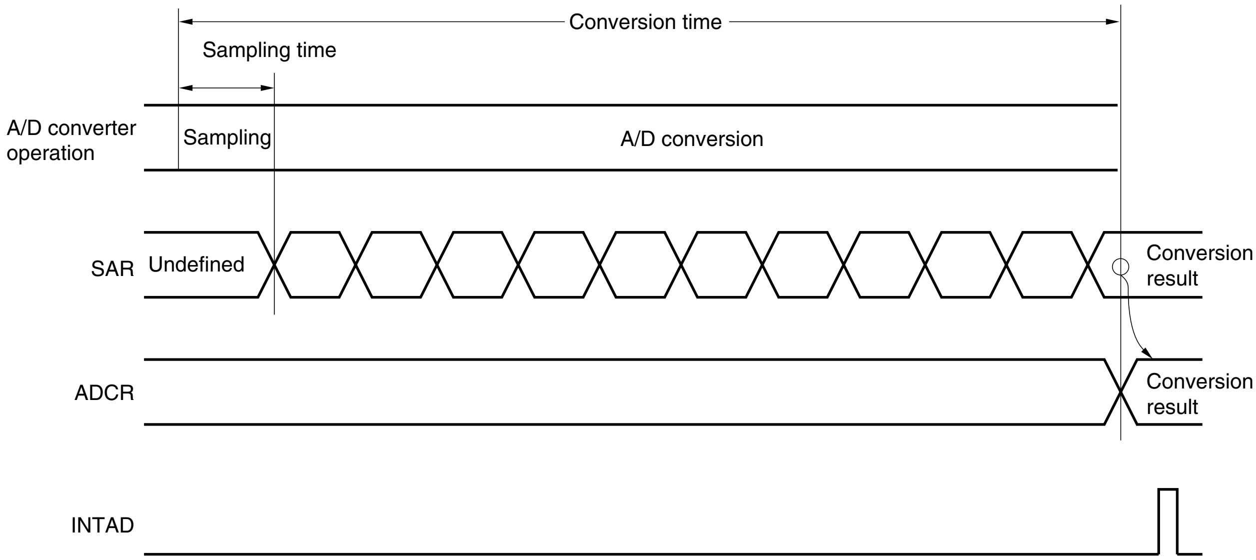
The voltage tap and sampled voltage are compared and bit 8 of SAR is manipulated as follows.
- Analog input voltage ≥ Voltage tap: Bit 8 = 1
- Analog input voltage < Voltage tap: Bit 8 = 0

<11> Comparison is continued in this way up to bit 0 of SAR.
<12> Upon completion of the comparison of 10 bits, an effective digital result value remains in SAR, and the result value is transferred to the A/D conversion result register (ADCR, ADCRH) and then latched.
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.
<13> Repeat steps <6> to <12>, until ADCS is cleared to 0.
To stop the A/D converter, clear ADCS to 0.
To restart A/D conversion from the status of ADCE = 1, start from <5>. To start A/D conversion again when ADCE = 0, set ADCE to 1, wait for 1 μs or longer, and start <5>. To change a channel of A/D conversion, start from <4>.

Caution Make sure the period of <1> to <5> is 1 μs or more.

Remark Two types of A/D conversion result registers are available.
- ADCR (16 bits): Store 10-bit A/D conversion value
- ADCRH (8 bits): Store 8-bit A/D conversion value

Figure 12-11. Basic Operation of A/D Converter



A/D conversion operations are performed continuously until bit 7 (ADCS) of the A/D converter mode register (ADM) is reset (0) by software.

If a write operation is performed to the analog input channel specification register (ADS) during an A/D conversion operation, the conversion operation is initialized, and if the ADCS bit is set (1), conversion starts again from the beginning.

Reset signal generation clears the A/D conversion result register (ADCR, ADCRH) to 0000H or 00H.

12.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI7) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$SAR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5\right)$$

$$ADCR = SAR \times 64$$

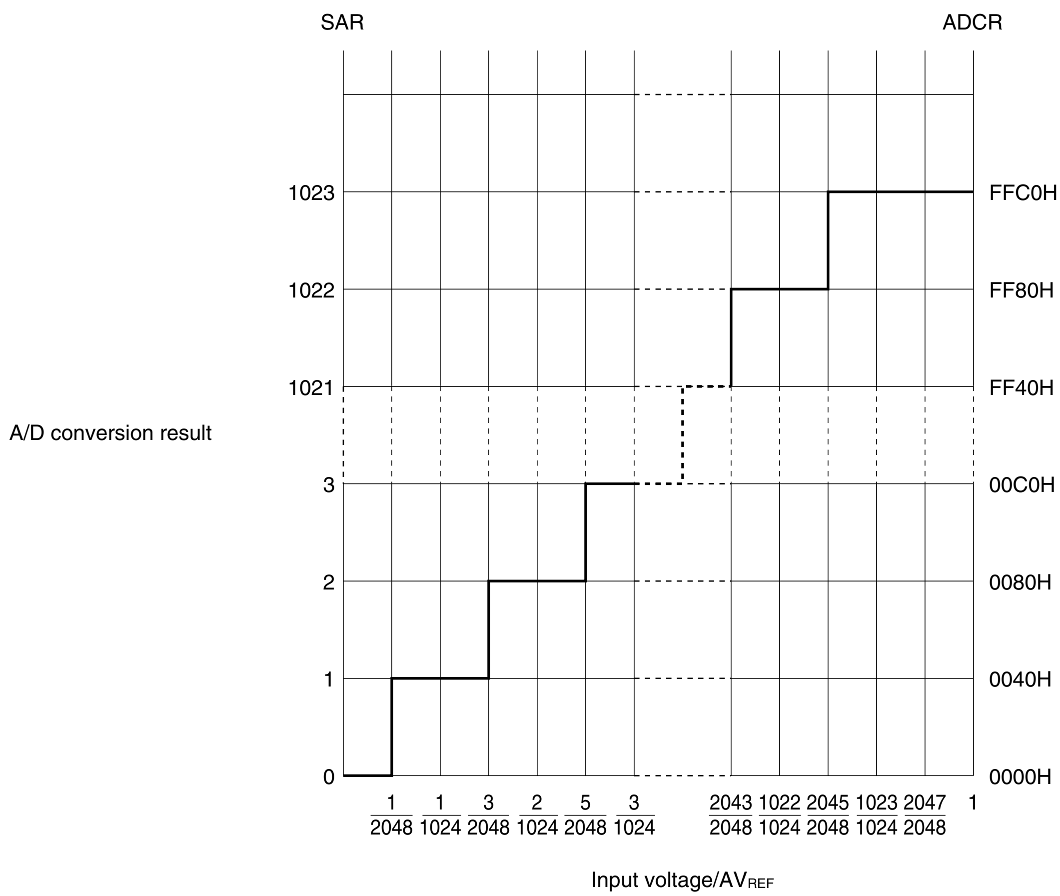
or

$$\left(\frac{ADCR}{64} - 0.5\right) \times \frac{AV_{REF}}{1024} \le V_{AIN} < \left(\frac{ADCR}{64} + 0.5\right) \times \frac{AV_{REF}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses
V_{AIN}: Analog input voltage
AV_{REF}: AV_{REF} pin voltage
ADCR: A/D conversion result register (ADCR) value
SAR: Successive approximation register

Figure 12-12 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 12-12. Relationship Between Analog Input Voltage and A/D Conversion Result



12.4.3 A/D converter operation mode

The operation mode of the A/D converter is the select mode. One channel of analog input is selected from ANI0 to ANI7 by the analog input channel specification register (ADS) and A/D conversion is executed.

(1) A/D conversion operation

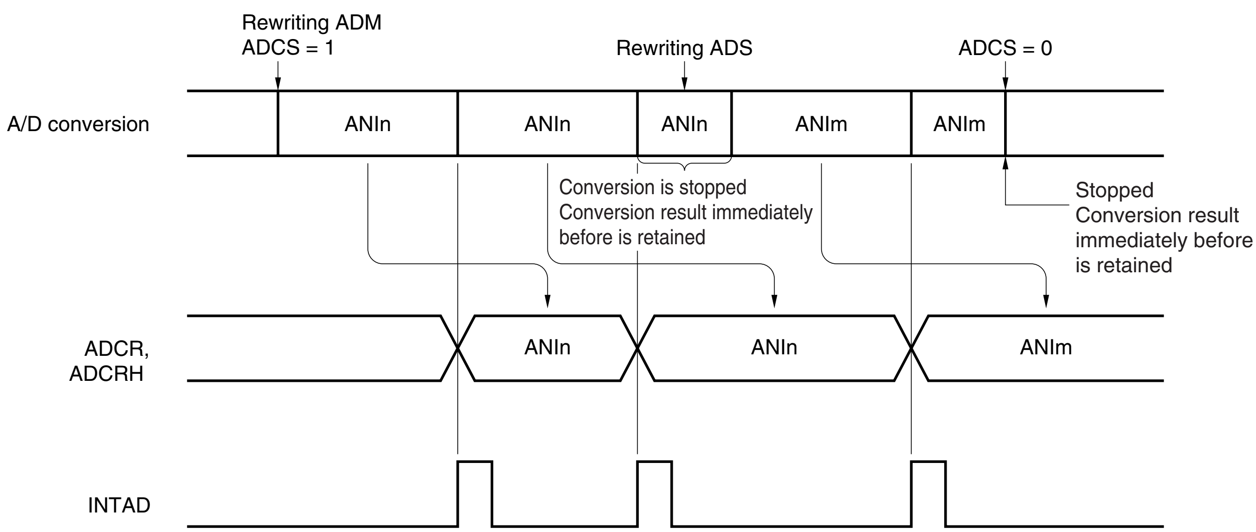
By setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 1, the A/D conversion operation of the voltage, which is applied to the analog input pin specified by the analog input channel specification register (ADS), is started.

When A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR), and an interrupt request signal (INTAD) is generated. When one A/D conversion has been completed, the next A/D conversion operation is immediately started.

If ADS is rewritten during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning.

If 0 is written to ADCS during A/D conversion, A/D conversion is immediately stopped. At this time, the conversion result immediately before is retained.

Figure 12-13. A/D Conversion Operation



Remarks 1. n = 0 to 7

2. m = 0 to 7

The setting methods are described below.

<1> Set bit 0 (ADCE) of the A/D converter mode register (ADM) to 1.
<2> Set the channel to be used in the analog input mode by using bits 3 to 0 (ADPC03 to ADPC00) of the A/D port configuration register 0 (ADPC0) and bits 7 to 0 (PM27 to PM20) of port mode register 2 (PM2).
<3> Select conversion time by using bits 6 to 1 (FR3 to FR0, LV1, and LV0) of ADM.
<4> Select a channel to be used by using bits 2 to 0 (ADS2 to ADS0) of the analog input channel specification register (ADS).
<5> Set bit 7 (ADCS) of ADM to 1 to start A/D conversion.
<6> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.
<7> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH).

<Change the channel>

<8> Change the channel using bits 2 to 0 (ADS2 to ADS0) of ADS to start A/D conversion.
<9> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.
<10> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH).

<Complete A/D conversion>

<11> Clear ADCS to 0.
<12> Clear ADCE to 0.

Cautions 1. Make sure the period of <1> to <5> is 1 μs or more.

2. <1> may be done between <2> and <4>.

3. <1> can be omitted. However, ignore data of the first conversion after <5> in this case.

4. The period from <6> to <9> differs from the conversion time set using bits 6 to 1 (FR3 to FR0, LV1, LV0) of ADM. The period from <8> to <9> is the conversion time set using FR3 to FR0, LV1, and LV0.

12.5 How to Read Successive Approximation Type A/D Converter Characteristics Table

Here, special terms unique to the successive approximation type A/D converter are explained.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

1LSB is as follows when the resolution is 10 bits.

$$1LSB = 1/2^{10} = 1/1024$$
$$= 0.098\%FSR$$

Accuracy has no relation to resolution, but is determined by overall error.

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.

Zero-scale error, full-scale error, integral linearity error, and differential linearity errors that are combinations of these express the overall error.

Note that the quantization error is not included in the overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, a ±1/2LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2LSB is converted to the same digital code, so a quantization error cannot be avoided.

Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

Figure 12-14. Overall Error

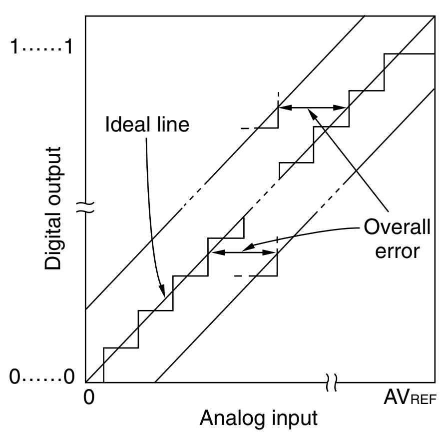
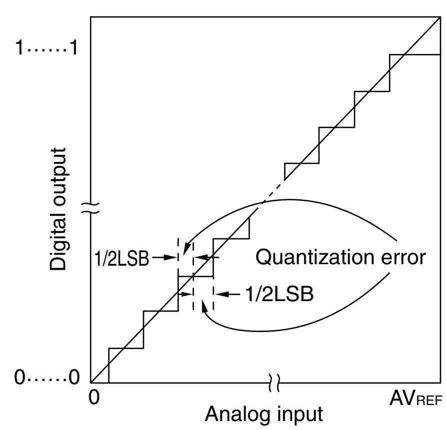


Figure 12-15. Quantization Error



(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2LSB) when the digital output changes from 0......000 to 0......001.

If the actual measurement value is greater than the theoretical value, it shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0……001 to 0……010.

(5) Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (Full-scale − 3/2LSB) when the digital output changes from 1......110 to 1......111.

(6) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

(7) Differential linearity error

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value.

Figure 12-16. Zero-Scale Error

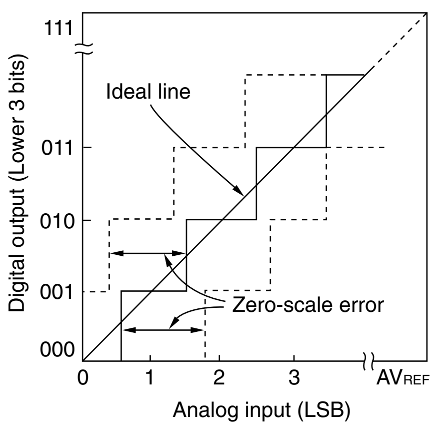


Figure 12-17. Full-Scale Error

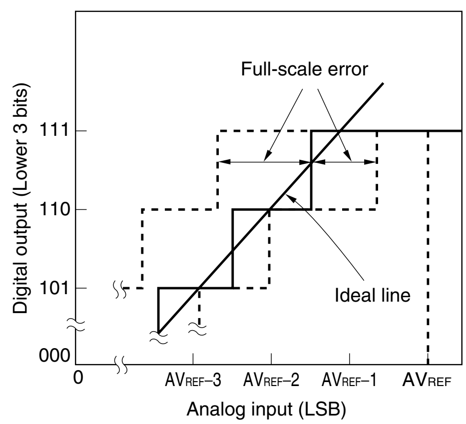


Figure 12-18. Integral Linearity Error

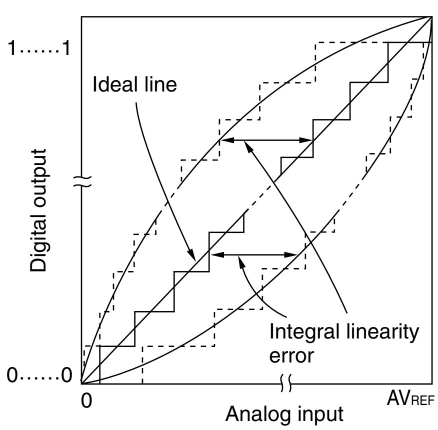
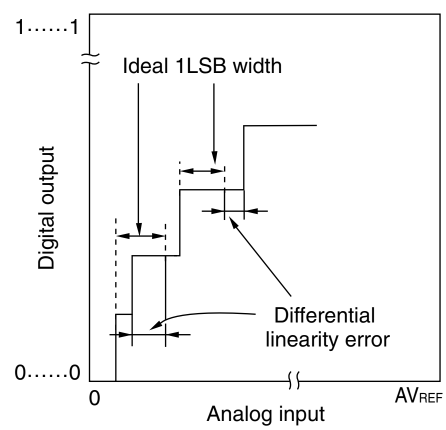


Figure 12-19. Differential Linearity Error



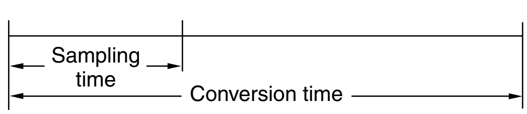
(8) Conversion time

This expresses the time from the start of sampling to when the digital output is obtained.

The sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.



12.6 Cautions for 10-bit successive approximation type A/D Converter

(1) Operating current in STOP mode

The A/D converter stops operating in the STOP mode. At this time, the operating current can be reduced by clearing bit 7 (ADCS) and bit 0 (ADCE) of the A/D converter mode register (ADM) to 0.

To restart from the standby status, clear bit 0 (ADIF) of interrupt request flag register 1L (IF1L) to 0 and start operation.

(2) Input range of ANI0 to ANI7

Observe the rated range of the ANI0 to ANI7 input voltage. If a voltage of AV_{REF} or higher and AV_{SS} or lower (even in the range of absolute maximum ratings) is input to an analog input channel, the converted value of that channel becomes undefined. In addition, the converted values of the other channels may also be affected.

(3) Conflicting operations

<1> Conflict between A/D conversion result register (ADCR, ADCRH) write and ADCR or ADCRH read by instruction upon the end of conversion

ADCR or ADCRH read has priority. After the read operation, the new conversion result is written to ADCR or ADCRH.

<2> Conflict between ADCR or ADCRH write and A/D converter mode register (ADM) write, analog input channel specification register (ADS), or A/D port configuration register 0 (ADPC0) write upon the end of conversion

ADM, ADS, or ADPC0 write has priority. ADCR or ADCRH write is not performed, nor is the conversion end interrupt signal (INTAD) generated.

(4) Noise countermeasures

To maintain the 10-bit resolution, attention must be paid to noise input to the AV_{REF} pin and pins ANI0 to ANI7.

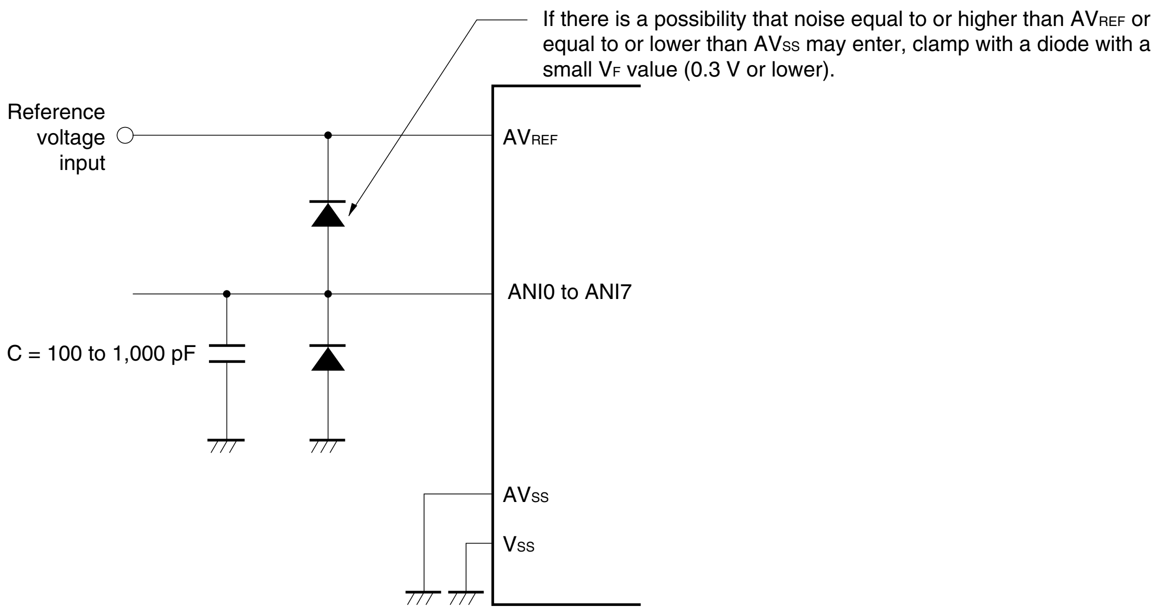
<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

<2> The higher the output impedance of the analog input source, the greater the influence. To reduce the noise, connecting external C as shown in Figure 12-20 is recommended.

<3> Do not switch these pins with other pins during conversion.

<4> The accuracy is improved if the HALT mode is set immediately after the start of conversion.

Figure 12-20. Analog Input Pin Connection



(5) ANI0/SEG39/P20 to ANI7/SEG32/P27 pins (μPD78F048x), ANI0/DS0−/P20 to ANI7/REF+/P27 pins (μPD78F049x)

<1> The analog input pins (ANI0 to ANI7) are also used as I/O port pins (P20 to P27).
When A/D conversion is performed with any of ANI0 to ANI7 selected, do not access P20 to P27 while conversion is in progress; otherwise the conversion resolution may be degraded. It is recommended to any pin of P20 to P27 used as digital I/O port starting with the ANI0/P20 that is the furthest from AV_{REF}.

<2> If a digital pulse is input or output, or segment-output to the pins adjacent to the pins currently used for A/D conversion, the expected value of the A/D conversion may not be obtained due to coupling noise. Therefore, do not input or output a pulse, or segment-output to the pins adjacent to the pin undergoing A/D conversion.

(6) Input impedance of ANI0 to ANI7 pins

This A/D converter charges a sampling capacitor for sampling during sampling time.
Therefore, only a leakage current flows when sampling is not in progress, and a current that charges the capacitor flows during sampling. Consequently, the input impedance fluctuates depending on whether sampling is in progress, and on the other states.
To make sure that sampling is effective, however, it is recommended to keep the output impedance of the analog input source to within 10 kΩ, and to connect a capacitor of about 100 pF to the ANI0 to ANI7 pins (see **Figure 12-20**).

(7) AV_{REF} pin input impedance

A series resistor string of several tens of kΩ is connected between the AV_{REF} and AV_{SS} pins.
Therefore, if the output impedance of the reference voltage source is high, this will result in a series connection to the series resistor string between the AV_{REF} and AV_{SS} pins, resulting in a large reference voltage error.

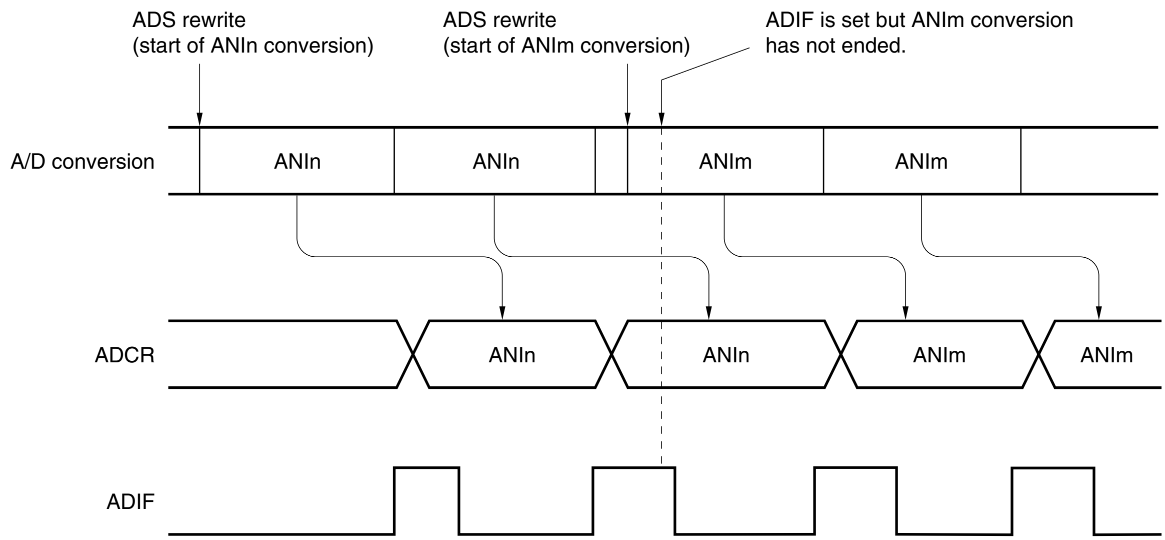
(8) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the analog input channel specification register (ADS) is changed.

Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and ADIF for the pre-change analog input may be set just before the ADS rewrite. Caution is therefore required since, at this time, when ADIF is read immediately after the ADS rewrite, ADIF is set despite the fact A/D conversion for the post-change analog input has not ended.

When A/D conversion is stopped and then resumed, clear ADIF before the A/D conversion operation is resumed.

Figure 12-21. Timing of A/D Conversion End Interrupt Request Generation



Remarks 1. n = 0 to 7

2. m = 0 to 7

(9) Conversion results just after A/D conversion start

The first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within 1 μs after the ADCE bit was set to 1, or if the ADCS bit is set to 1 with the ADCE bit = 0. Take measures such as polling the A/D conversion end interrupt request (INTAD) and removing the first conversion result.

(10) A/D conversion result register (ADCR, ADCRH) read operation

When a write operation is performed to the A/D converter mode register (ADM), analog input channel specification register (ADS), and A/D port configuration register 0 (ADPC0), the contents of ADCR and ADCRH may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC0. Using a timing other than the above may cause an incorrect conversion result to be read.

<R> **(11) Internal equivalent circuit**

The equivalent circuit of the analog input block is shown below.

Figure 12-22. Internal Equivalent Circuit of ANIn Pin

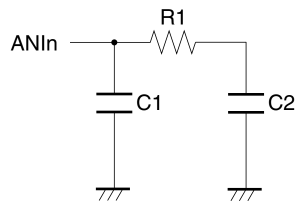


Table 12-4. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{REF}	R1	C1	C2
4.0 V ≤ AV_{REF} ≤ 5.5 V	8.1 kΩ	8 pF	5 pF
2.7 V ≤ AV_{REF} < 4.0 V	31 kΩ	8 pF	5 pF
2.3 V ≤ AV_{REF} < 2.7 V	381 kΩ	8 pF	5 pF

Remarks 1. The resistance and capacitance values shown in Table 12-4 are not guaranteed values.

2. n = 0 to 7

(12) Simultaneous use of the 10-bit successive approximation type A/D converter and the 16-bit ΔΣ type A/D converter (μPD78F049x only)

The A/D conversion accuracy may deteriorate when the 10-bit successive approximation type A/D converter and the 16-bit ΔΣ type A/D converter are used at the same time.

Stop the 16-bit ΔΣ type A/D converter during 10-bit successive approximation type A/D converter operation, because the accuracy cannot be guaranteed. Also, stop the 10-bit successive approximation type A/D converter during 16-bit ΔΣ type A/D converter operation. (Do not operate them simultaneously.)

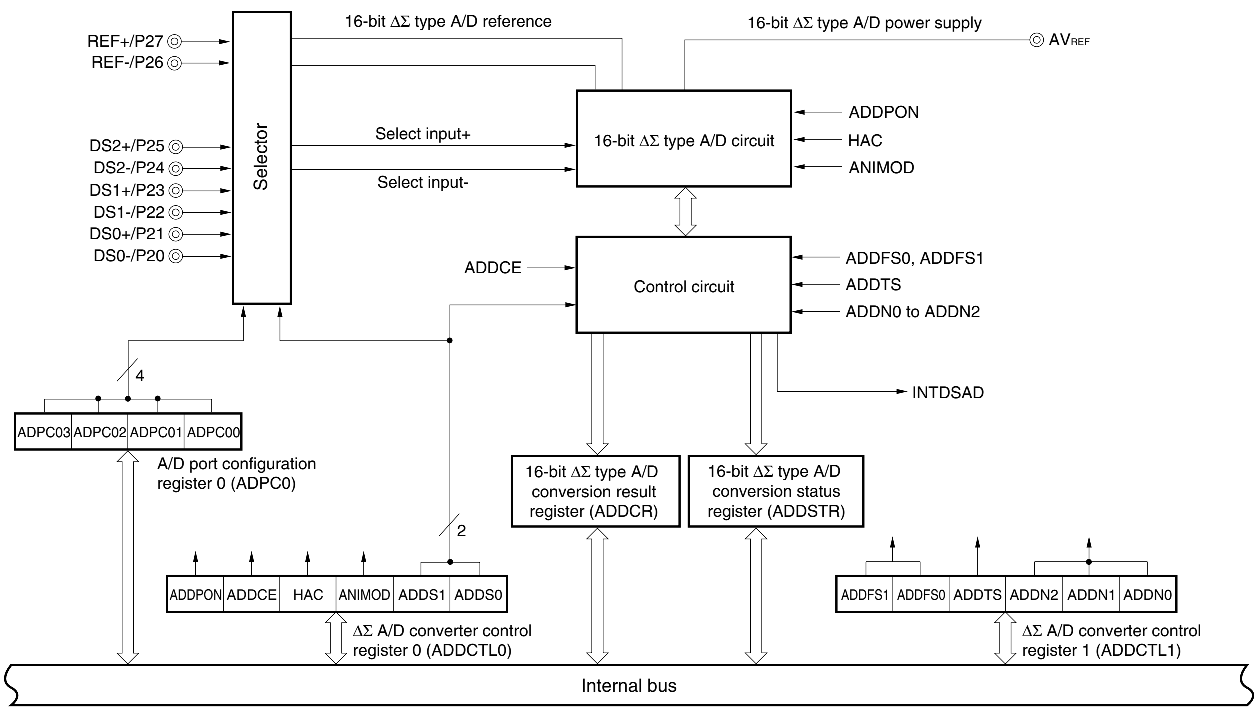
13.1 Function of 16-Bit ΔΣ Type A/D Converter

The 16-bit ΔΣ type A/D converter converts an analog input signal into a digital value, and consists of up to three channels (DS0−/DS0+, DS1−/DS1+, DS2−/DS2+) with a resolution of 16 bits.

The A/D converter has the following function.

- **16-bit resolution A/D conversion**
  16-bit resolution A/D conversion is carried out repeatedly for one analog input channel selected from DS0−/DS0+, DS1−/DS1+, and DS2−/DS2+. Each time an A/D conversion operation ends, an interrupt request (INTDSAD) is generated.
  The conversion time can be shortened by lowering the resolution.

Figure 13-1. Block Diagram of 16-Bit ΔΣ Type A/D Converter



13.2 Configuration of 16-Bit ΔΣ Type A/D Converter

The 16-bit ΔΣ type A/D converter includes the following hardware.

(1) DS0−/DS0+, DS1−/DS1+, and DS2−/DS2+ pins

These are the 3-channel analog input pins of the 16-bit ΔΣ type A/D converter. They input analog signals to be converted into digital signals. Pins other than the one selected as the analog input pin can be used as I/O port pins.

When using these pins in differential input mode, input analog signals to DS− and DS+. When using them in single input mode, input analog signals to DS+ and set DS− to the same potential as V_{SS} and AV_{SS}.

(2) 16-bit ΔΣ type A/D circuit

A 16-bit ΔΣ type A/D circuit converts voltage values sampled according to the reference voltage to digital values and outputs them to the control circuit.

(3) Control circuit

A control circuit controls the conversion time and starting/stopping of conversion operation of analog input to be A/D converted. When A/D conversion is completed, the conversion result is transferred to the 16-bit ΔΣ type A/D conversion result register (ADDCR) whereby an interrupt (INTDSAD) is generated.

(4) 16-bit ΔΣ type A/D conversion result register (ADDCR)

The A/D conversion result is loaded from the control circuit to this register each time A/D conversion is completed, and the ADDCR register holds the A/D conversion result in its higher 16 bits.

(5) 8-bit ΔΣ type A/D conversion result register (ADDCRH)

The A/D conversion result is loaded from the control circuit to this register each time A/D conversion is completed, and the ADDCRH register stores the higher 8 bits of the A/D conversion result.

Caution When data is read from ADDCR and ADDCRH, a wait cycle is generated. Do not read data from ADDCR and ADDCRH when the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped.

(6) AV_{REF} pin

This pin inputs an analog power to the 16-bit ΔΣ type A/D circuit. When using at least one port of port 2 as a digital port or for segment output, set it to the same potential as the V_{DD} pin.

(7) REF− and REF+ pins

This pin inputs the reference voltage of the 16-bit ΔΣ type A/D converter. Signals input to DS0−/DS0+, DS1−/DS1+ and DS2−/DS2+ are converted to digital signals, according to the voltage applied across REF− and REF+. The REF− and REF+ pins must be used at the same potential as the V_{SS}/AV_{SS} and AV_{REF} pins, respectively.

(8) AV_{SS} pin

This is the ground potential pin of the A/D converter. Always use this pin at the same potential as that of the V_{SS} pin even when the A/D converter is not used.

(9) 16-bit ΔΣ type A/D converter control register 0 (ADDCTL0)

This register sets the 16-bit ΔΣ type A/D circuit or control circuit power on/off state, conversion start/stop state, high-accuracy mode on/off state, ΔΣ input mode control, and analog input channel.

(10) 16-bit ΔΣ type A/D converter control register 1 (ADDCTL1)

This register sets the sampling clock to be A/D converted, serial/parallel mode state and sampling count (resolution).

(11) 16-bit ΔΣ type A/D conversion status register (ADDSTR)

This register checks which channel has completed conversion, when a 16-bit ΔΣ type A/D conversion operation completion (conversion completion interrupt generation) and a conversion channel change occur at the same time.

(12) A/D port configuration register 0 (ADPC0)

This register switches the ANI0/P20/DS0− to ANI7/P27/REF+ pins to analog input (analog input of 16-bit ΔΣ type A/D converter or analog input of 10-bit successive approximation type A/D converter) or digital I/O of port.

(13) Port mode register 2 (PM2)

This register switches the ANI0/P20/DS0− to ANI7/P27/REF+ pins to input or output.

13.3 Registers Used in 16-Bit ΔΣ Type A/D Converter

The 16-bit ΔΣ type A/D converter uses the following nine registers.

- 16-bit ΔΣ type A/D converter control register 0 (ADDCTL0)
- 16-bit ΔΣ type A/D converter control register 1 (ADDCTL1)
- 16-bit ΔΣ type A/D conversion result register (ADDCR)
- 8-bit ΔΣ type A/D conversion result register (ADDCRH)
- 16-bit ΔΣ type A/D conversion status register (ADDSTR)
- A/D port configuration register 0 (ADPC0)
- 16-bit ΔΣ type A/D sampling delay time setting enable register
- 16-bit ΔΣ type A/D sampling delay time setting register
- Port mode register 2 (PM2)

(1) 16-bit ΔΣ type A/D converter control register 0 (ADDCTL0)

This register sets the 16-bit ΔΣ type A/D circuit or control circuit power on/off state, conversion start/stop state, high-accuracy mode on/off state, ΔΣ input mode control, and analog input channel.

ADDCTL0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 13-2. Format of 16-bit ΔΣ type A/D Converter Control Register 0 (ADDCTL0)

Address: FF7CH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
ADDCTL0	ADPON	ADDCE	HAC	AINMCD	0	0	ADDS1	ADDS0

ADDPON	16-bit ΔΣ type A/D circuit power supply control
0	Power supply OFF
1	Power supply ON

ADDCE	16-bit ΔΣ type A/D conversion operation control
0	Stops conversion operation
1	Starts conversion operation

HAC	Setting 16-bit ΔΣ type A/D conversion high-accuracy mode
0	High-accuracy mode OFF
1	High-accuracy mode ON

AINMOD	16-bit ΔΣ type A/D conversion input mode control
0	Single input
1	Differential input

ADDS1	ADDS0	16-bit ΔΣ type analog input specification
0	0	DS0+/DS0-
0	1	DS1+/DS1-
1	0	DS2+/DS2-
1	1	Setting prohibited

Cautions 1. Do not set the ADDPON and ADDCE bits to 1 at the same time. ADDCE must be set to 1, at least 1.2 μs after ADDPON has been set to 1.

2. Setting the ΔΣ analog input channel to be set by ADDS1 and ADDS0 to a pin which has been selected to be used in the analog input mode by the ADPC0 register is prohibited.

3. Operating 16-bit ΔΣ type A/D conversion and 10-bit successive approximation type A/D conversions at the same time (ADDCE = 1 and ADCS = 1) is prohibited.

4. If ADDCTL0 is rewritten (including identical data), A/D conversion operation is resumed after it has been initialized.

5. Set the input voltage in accordance with Table 13-4 Input Voltage Range.

6. When executing a STOP instruction, power to the 16-bit ΔΣ type A/D converter must be turned off (ADDPON = 0).

(2) 16-bit ΔΣ type A/D converter control register 1 (ADDCTL1)

This register sets the sampling clock to be 16-bit ΔΣ type A/D converted, serial/parallel mode state and sampling count (resolution).

ADDCTL1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 13-3. Format of 16-Bit ΔΣ Type A/D Converter Control Register (ADDCTL1)

Address: FF7DH After reset: 00H R/W

Symbol	7	6	<5>	4	3	2	1	0
ADDCTL1	ADDFS1	ADDFS0	ADDTS	0	0	ADDN2	ADDN1	ADDN0

ADDFS1	ADDFS0	16-bit ΔΣ type A/D sampling clock (f_{VP}) selection
0	0	$f_{PRS}/4$
0	1	$f_{PRS}/8$
1	0	$f_{PRS}/16$
1	1	$f_{SUB}/2$

ADDTS	Setting 16-bit ΔΣ type A/D serial/parallel mode
0	Serial mode
1	Parallel mode

ADDN2	ADDN1	ADDN0	Number of times of 16-bit ΔΣ type A/D sampling N (resolution)
0	0	0	256 (8 bits)
0	0	1	1024 (10 bits)
0	1	0	2048 (11 bits)
0	1	1	4096 (12 bits)
1	0	0	8192 (13 bits)
1	0	1	16384 (14 bits)
1	1	0	32768 (15 bits)
1	1	1	65536 (16 bits)

Cautions 1. Set the sampling clock (conversion time) so that it satisfies the conditions in Table 13-1. When selecting the conversion time, take clock frequency errors into consideration.

2. Writing to the ADDCTL1 register during 16-bit ΔΣ type A/D conversion operation is prohibited. Be sure to write after 16-bit ΔΣ type A/D conversion operation has been stopped (ADDCE = 0).

3. Setting the parallel mode is prohibited when f_{SUB} is selected as the 16-bit ΔΣ type A/D sampling clock (f_{VP}).

The conversion time can be derived from the sampling clock (f_{VP}) and sampling count (N) via the following calculations.

Sampling time = $1/f_{VP} \times N$
Initialization time = $1/\text{operation clock} + 1/f_{VP} \times 256$

Operation clock
ADDFS1-0 selected as 1, 1: f_{SUB}
ADDFS1-0 selected as other than the above: f_{PRS}

In serial mode

<First conversion>

Conversion time = Initialization time + sampling time
= $(1/\text{operation clock} + 1/f_{VP} \times 256) + (1/f_{VP} \times N)$

<After second conversion>

Conversion time = Sampling time
= $1/f_{VP} \times N$

In parallel mode

<First conversion>

Conversion time = Initialization time + sampling time
= $(1/\text{operation clock} + 1/f_{VP} \times 256) + (1/f_{VP} \times N)$

<After second conversion>

Conversion time = Sampling time/4
= $1/f_{VP} \times N/4$

f_{VP}: sampling clock, N: 16-bit ΔΣ type A/D sampling count

Caution If ADDCTL0 is rewritten (including the same values), conversion is assumed to have been restarted from that point and the conversion time of the first conversion is applied.

Table 13-1. Sampling Clock (Sampling Time) Setting Conditions

ADDN2	AV_{REF} Condition	Sampling Clock: f_{VP} (Conversion Time in 16-bit Resolution)
Differential input	3.5 V ≤ AV_{REF} ≤ 5.5 V	1.25 MHz max. (52.42 ms min.)
	2.7 V ≤ AV_{REF} < 3.5 V	625 kHz max. (104.85 ms min.)
Single input	2.85 V ≤ AV_{REF} ≤ 5.5 V	625 kHz max. (104.85 ms min.)
	2.7 V ≤ AV_{REF} < 2.85 V	525 kHz max. (124.83 ms min.)

Table 13-2. Examples of Sampling Time under Setting Conditions

f_{PRS}	f_{VP}	Number of Times of 16-bit ΔΣ Type A/D Sampling: N (Resolution) 65536 (16-bit)	32768 (15-bit)	16384 (14-bit)	8192 (13-bit)	4096 (12-bit)	2048 (11-bit)	1024 (10-bit)	256 (8-bit)
10 MHz	$f_{PRS}/4$	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
	$f_{PRS}/8$ Note 1	52.42 ms	26.21 ms	13.10 ms	6.55 ms	3.27 ms	1.63 ms	0.81 ms	0.20 ms
	$f_{PRS}/16$ Note 2	104.85 ms	52.42 ms	26.21 ms	13.10 ms	6.55 ms	3.27 ms	1.63 ms	0.41 ms
8 MHz	$f_{PRS}/4$	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
	$f_{PRS}/8$ Note 1	65.53 ms	32.76 ms	16.38 ms	8.19 ms	4.09 ms	2.04 ms	1.02 ms	0.25 ms
	$f_{PRS}/16$	131.07 ms	65.53 ms	32.76 ms	16.38 ms	8.19 ms	4.09 ms	2.04 ms	0.51 ms
5 MHz	$f_{PRS}/4$ Note 1	52.42 ms	26.21 ms	13.10 ms	6.55 ms	3.27 ms	1.63 ms	0.81 ms	0.40 ms
	$f_{PRS}/8$ Note 2	104.85 ms	52.42 ms	26.21 ms	13.10 ms	6.55 ms	3.27 ms	1.63 ms	0.81 ms
	$f_{PRS}/16$	209.71 ms	104.85 ms	52.42 ms	26.21 ms	13.10 ms	6.55 ms	3.27 ms	1.63 ms
4 MHz	$f_{PRS}/4$ Note 1	65.53 ms	32.76 ms	16.38 ms	8.19 ms	4.09 ms	2.04 ms	1.02 ms	0.25 ms
	$f_{PRS}/8$	131.07 ms	65.53 ms	32.76 ms	16.38 ms	8.19 ms	4.09 ms	2.04 ms	0.51 ms
	$f_{PRS}/16$	262.14 ms	131.07 ms	65.53 ms	32.76 ms	16.38 ms	8.19 ms	4.09 ms	1.02 ms
2 MHz	$f_{PRS}/4$	131.07 ms	65.53 ms	32.76 ms	16.38 ms	8.19 ms	4.09 ms	2.04 ms	0.51 ms
	$f_{PRS}/8$	262.14 ms	131.07 ms	65.53 ms	32.76 ms	16.38 ms	8.19 ms	4.09 ms	1.02 ms
	$f_{PRS}/16$	524.28 ms	262.14 ms	131.07 ms	65.53 ms	32.76 ms	16.38 ms	8.19 ms	2.04 ms
–	$f_{SUB}/2$	4 s	2 s	1 s	500 ms	250 ms	125 ms	62.5 ms	15.62 ms

Notes 1. Setting the differential input mode (2.7 V ≤ AV_{REF} < 3.5 V) and single input mode is prohibited since the sampling time conditions are not satisfied in these modes.

2. Setting the single input mode (2.7 V ≤ AV_{REF} < 2.85 V) is prohibited since the sampling time conditions are not satisfied in this modes.

(3) 16-bit ΔΣ type A/D conversion result register (ADDCR)

This register is a 16-bit register that stores the A/D conversion result. Each time A/D conversion ends, the conversion result is loaded from the ΔΣ A/D circuit. The higher 8 bits of the conversion result are stored in FF7FH and the lower 8 bits are stored in FF7EH.

ADDCR can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 13-4. Format of 16-Bit ΔΣ type A/D Conversion Result Register (ADDCR)

Address: FF7EH, FF7FH After reset: 0000H R

Symbol	FF7FH								FF7EH							
ADDCR																

Cautions 1. When N-bit resolution is set, conversion results are stored starting from the higher bits and the remaining 16-N bits are fixed to "0".

2. If the conversion completion interrupt and conversion result read operation conflict, the conversion result may be undefined. Read the conversion result after the generation of the conversion result completion interrupt, and before the next conversion completion.

(4) 8-bit ΔΣ type A/D conversion result register (ADDCRH)

This register is an 8-bit register that stores the A/D conversion result. The higher 8 bits of 16-bit resolution are stored.

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

Reset signal generation clears this register to 00H.

Figure 13-5. Format of 8-Bit ΔΣ type A/D Conversion Result Register (ADDCRH)

Address: FF7FH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADDCRH								

Caution If the conversion completion interrupt and conversion result read operation conflict, the read value of the conversion result may be undefined. Read the conversion result after the generation of the conversion result completion interrupt, and before the next conversion completion.

(5) 16-bit ΔΣ type A/D conversion status register (ADDSTR)

This register holds the channel for which A/D conversion has been completed. It also checks which channel has completed conversion.

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

Reset signal generation clears this register to 00H.

Figure 13-6. Format of 16-Bit ΔΣ type A/D Conversion Status Register (ADDSTR)

Address: FF75H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADDSTR	0	0	0	0	0	0	ADDIT1	ADDIT0

ADDIT1	ADDIT0	Channel converted by 16-bit ΔΣ type A/D conversion	
		When differential input is selected	When single input is selected
0	0	DS0+/DS0-	DS0+
0	1	DS1+/DS1-	DS1+
1	0	DS2+/DS2-	DS2+

(6) A/D port configuration register 0 (ADPC0)

This register switches the ANI0/P20/DS0- to ANI7/P27/REF+ pins to analog input (analog input of 16-bit ΔΣ type A/D converter or analog input of 10-bit successive approximation type A/D converter) or digital I/O of port.

ADPC0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 08H.

Figure 13-7. Format of A/D Port Configuration Register 0 (ADPC0)

Address: FF8FH After reset: 08H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC0	0	0	0	0	ADPC03	ADPC02	ADPC01	ADPC00

ADPC03	ADPC02	ADPC01	ADPC00	Digital I/O (D)/analog input (A: successive approximation type, Δ: ΔΣ type) switching							
				P27/ ANI7/ REF+	P26/ ANI6/ REF-	P25/ ANI5/ DS2+	P24/ ANI4/ DS2-	P23/ ANI3/ DS1+	P22/ ANI2/ DS1-	P21/ ANI1/ DS0+	P20/ ANI0/ DS0-
0	0	0	0	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ
0	0	0	1	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A	D
0	0	1	0	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	D	D
0	0	1	1	A/Δ	A/Δ	A/Δ	A/Δ	A	D	D	D
0	1	0	0	A/Δ	A/Δ	A/Δ	A/Δ	D	D	D	D
0	1	0	1	A	A	A	D	D	D	D	D
0	1	1	0	A	A	D	D	D	D	D	D
0	1	1	1	A	D	D	D	D	D	D	D
1	0	0	0	D	D	D	D	D	D	D	D
Other than above				Setting prohibited							

Cautions 1. Set the channel used for A/D conversion to the input mode by using port mode register 2 (PM2).

2. Do not set the pin set by ADPC0 as digital I/O by ADS, ADDS1, or ADDS0.

3. If data is written to ADPC0, a wait cycle is generated. Do not write data to ADPC0 when the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped. For details, see CHAPTER 34 CAUTIONS FOR WAIT.

(7) 16-bit ΔΣ type A/D sampling delay time setting enable register

This register enables the setting of the sampling delay time.

The address of this register is directly specified and set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Remark This register is not required to be set if the A/D conversion accuracy is sufficient.

Figure 13-8. Format of 16-bit ΔΣ type A/D sampling delay time setting enable register

Address: FA26H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
	ADD0TEN	0	0	0	0	0	0	0

ADD0TEN	Control to set delay time
0	Disables delay time setting
1	Enables delay time setting

Caution Be sure to set this register after turning off the power of the 16-bit ΔΣ type A/D circuit (ADDPON = 0) and stopping conversion operation (ADDCE = 0).

(8) 16-bit ΔΣ type A/D sampling delay time setting register

This register sets the sampling delay time.

The accuracy can be improved by setting the optimal delay time.

The address of this register is directly specified and set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 20H.

Remark This register is not required to be set if the A/D conversion accuracy is sufficient.

Figure 13-9. Format of 16-bit ΔΣ type A/D sampling delay time setting register

Address: FA27H After reset: 20H R/W

Symbol	7	6	5	4	3	2	1	0
	0	ADD0DLY2	ADD0DLY1	ADD0DLY0	0	0	0	0

ADD0DLY2	ADD0DLY1	ADD0DLY0	Setting delay time [nsec]
0	0	0	2
0	0	1	4
0	1	0	6 (default)
0	1	1	8
1	0	0	10
1	0	1	12
1	1	0	14
1	1	1	16

Caution Be sure to set this register after turning off the power of the 16-bit ΔΣ type A/D circuit (ADDPON = 0), stopping conversion operation (ADDCE = 0), and enabling the delay time setting (ADD0TEN = 1).

(9) Port mode register 2 (PM2)

When using the ANI0/P20/DS0− to ANI7/P27/REF+ pins for analog input port, set PM20 to PM27 to 1. The output latches of P20 to P27 at this time may be 0 or 1.

If PM20 to PM27 are set to 0, they cannot be used as analog input port pins.

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

Reset signal generation sets this register to FFH.

Figure 13-10. Format of Port Mode Register 2 (PM2)

Address: FF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20

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

P20/ANI0/DS0− to P27/ANI7/REF+ pins are as shown below depending on the settings of ADPC0, PM2, ADS, and ADDCTL0.

Table 13-3. Setting Functions of P20/ANI0/DS0− to P27/ANI7/REF+ Pins

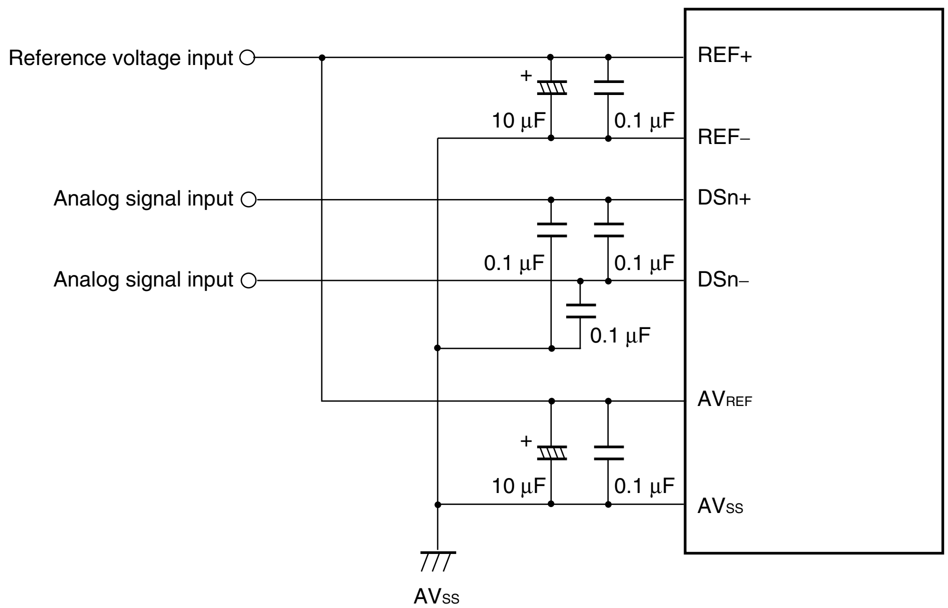
ADPC0	PM2	ADS	ADDCTL0	P20/ANI0/DS0− to P27/ANI7/REF+ Pins
Analog input selection	Input mode	Does not select ANI	Does not select DSn±.	Analog input (not to be converted)
		Selects ANI	Does not select DSn±.	Analog input (to be converted by successive approximation type A/D converter)
		Does not select ANI	Selects DSn±.	Analog input (to be converted by ΔΣ type A/D converter)
		Selects ANI	Selects DSn±.	Setting prohibited
	Output mode	−		Setting prohibited
Digital I/O selection	Input mode	−		Digital input
	Output mode	−		Digital output

Remark n = 0 to 2

13.4 Circuit Configuration Example of 16-Bit ΔΣ Type A/D Converter

Figures 13-11 shows the circuit configuration example when using the 16-bit ΔΣ type A/D converter.

Figure 13-11. Circuit Configuration Example When Using 16-Bit ΔΣ Type A/D Converter (Differential Input)



13.5 16-Bit ΔΣ Type A/D Converter Operations

13.5.1 Basic operations of 16-bit ΔΣ type A/D converter

<1> Set ADD0TEN[Note 1] = 1 (enable the delay time setting).[Note 2]
<2> Use ADD0DLY2 to ADD0DLY0[Note 1] to set the delay time.[Note 2]
<3> Set the ΔΣ A/D conversion target channel.
<4> Set ADDPON to 1 (ΔΣ A/D power-on).
<5> Set the conversion operation modes, such as the input mode, operation mode and sampling counts via the ADDCTL1 and ADDCTL0 registers.
<6> Conversion operation starts when ADDCE is set to 1, at least 1.2 μs after ADDPON has been set to 1. (Conversion operation also starts when ADDCE is set to 1 before 1.2 μs elapse after ADDPON has been set to 1, but the conversion result is not guaranteed in this case).[Note 3]
<7> When conversion is completed, an interrupt (INTDSAD) is generated and the result is stored in the ADDCR register. Read the ADDCR register value.
<8> If ADDCE is not to be set to 0 (conversion operation stop), repeat step 5. If conversion operation is to be stopped, set ADDCE to 0.
<9> When the current is to be reduced without using ΔΣ A/D, set ADDPON to 0 (ΔΣ A/D power-off).

Notes 1. Directly specify the addresses and write to ADD0TEN and ADD0DLY2 to ADD0DLY0 by using an 8-bit memory manipulation instruction.
2. Setting of the delay time is not required if the conversion accuracy is sufficient.
3. Writing to ADDCTL1 during conversion operation is prohibited.
When the pin settings subject to ΔΣ A/D conversion are altered during conversion operation, conversion results are not stored. When the target pin settings are altered, restart conversion operation.

13.5.2 Operation mode of 16-bit ΔΣ type A/D converter

Several operation modes can be set for the 16-bit ΔΣ type A/D converter.

(1) Differential input mode/single input mode
Differential input mode or single input mode can be selected as the input mode for the 16-bit ΔΣ type A/D converter. The accuracy is higher in differential input mode than in single input mode. When using the differential input mode, input analog signals to DSn- and DSn+. When using the single input mode, input analog signals to DSn+ and set DSn- to the same potential as V_{SS} and AV_{SS}. Make sure that the central values of the DSn- and DSn+ input voltages are 0.5 REF+ in differential input mode.

(2) 16-bit ΔΣ type A/D high-accuracy mode
High-accuracy mode ON or OFF can be selected as the conversion accuracy mode for the 16-bit ΔΣ type A/D converter. The accuracy is higher when set to high-accuracy mode ON than when set to high-accuracy mode OFF.

Table 13-4. Input Voltage Range

		Input Voltage Range to DSn+	Input Voltage Range to DSn-
Differential input	High-accuracy mode ON	0.5 × (REF+) + X1	0.5 × (REF+) − X1
	High-accuracy mode OFF	0.5 × (REF+) + X2	0.5 × (REF+) − X2
Single input	High-accuracy mode ON	0.1 × (REF+) to 0.9 × (REF+)	Fixed to AV_{SS}
	High-accuracy mode OFF	0 to REF+	

Remark X1 = −0.4 × (REF+) to 0.4 × (REF+)
X2 = −0.5 × (REF+) to 0.5 × (REF+)
n = 0 to 2

Figure 13-12. Example of Application circuit

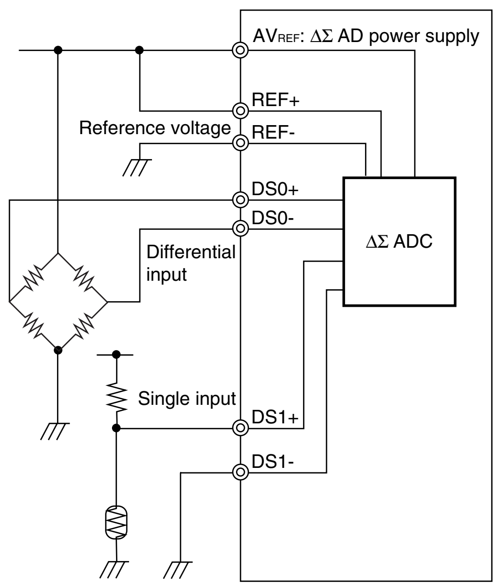
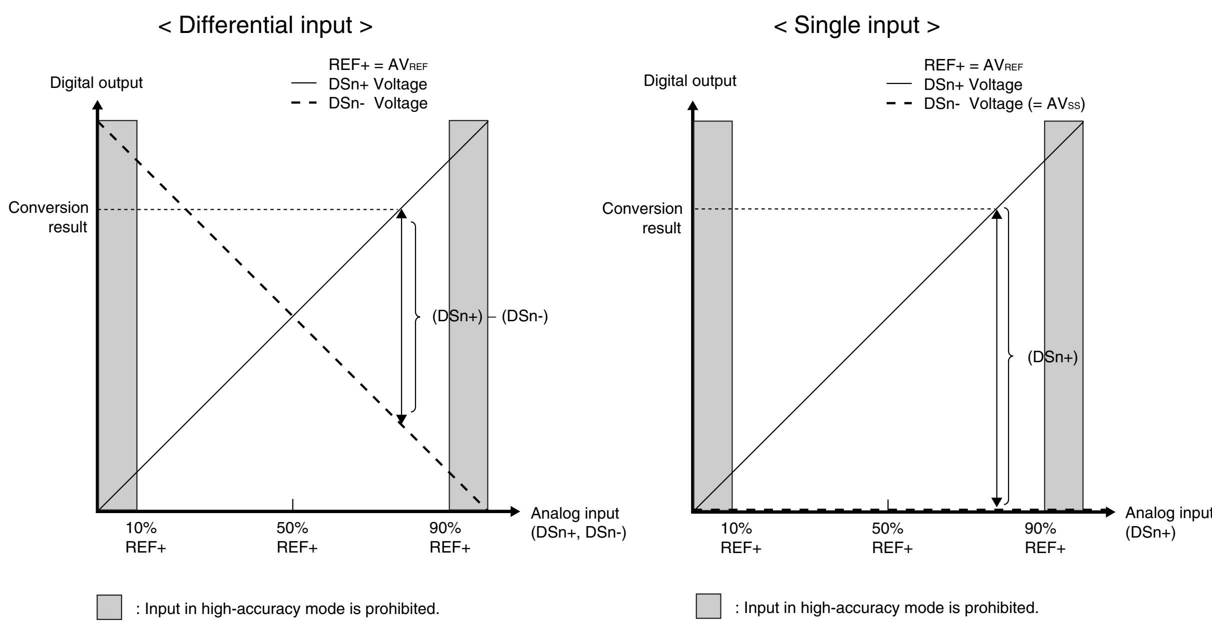


Figure 13-13. Enabled input range by A/D converter mode

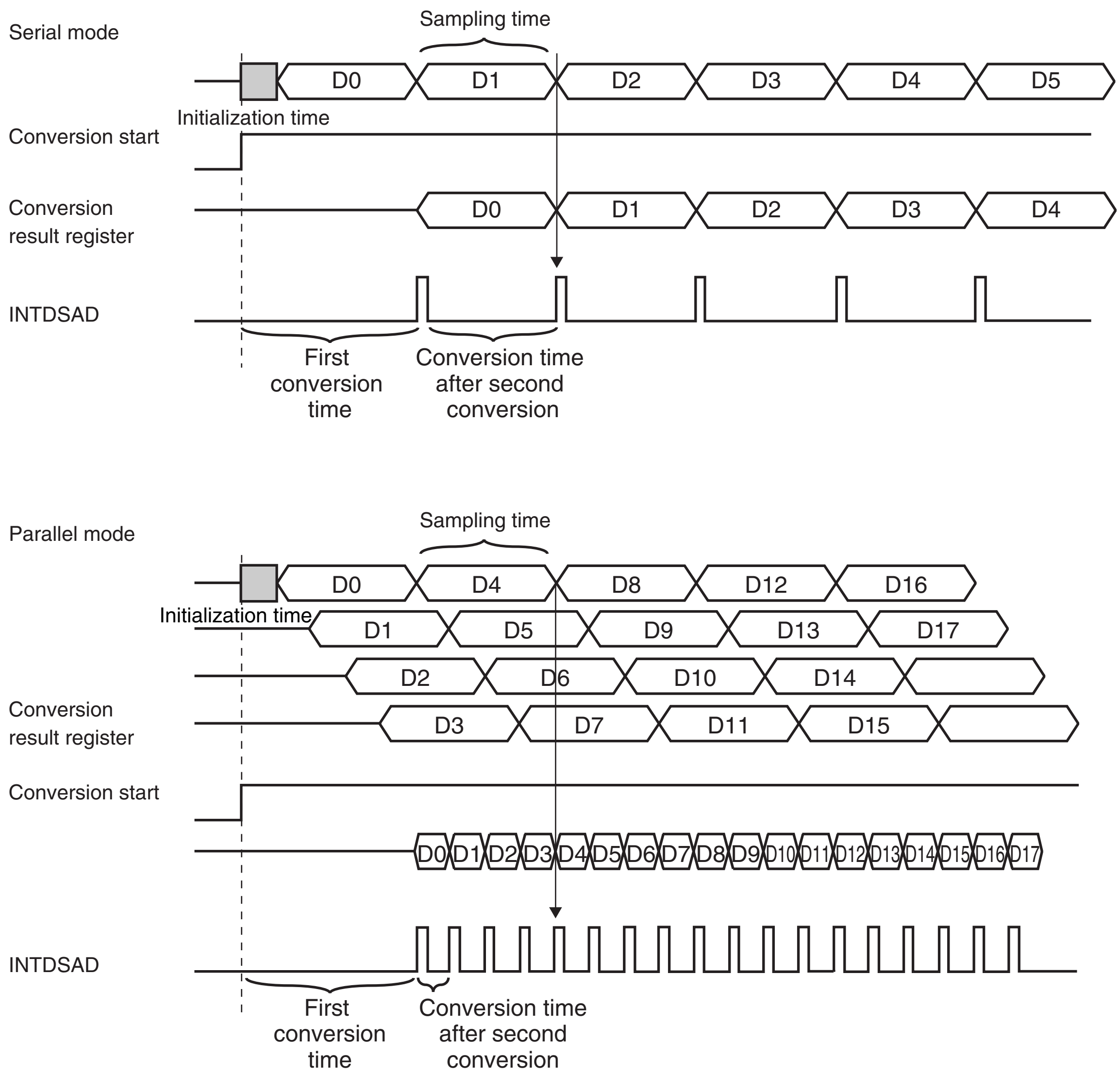


Remark n = 0 to 2

(3) Serial mode/parallel mode

Serial or parallel mode can be selected as the input mode for the 16-bit ΔΣ type A/D converter. The parallel mode can reduce the conversion time to a fourth of that in the serial mode. The conversion time of the first conversion, however, is the same as that in the serial mode. Also, the sampling time itself is the same as in the serial mode.

Figure 13-14. Conversion Time and Sampling Time



13.6 How to Read ΔΣ Type A/D Converter Characteristics Table

Here, special terms unique to the ΔΣ type A/D converter are explained.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

1LSB is as follows when the resolution is 16 bits.

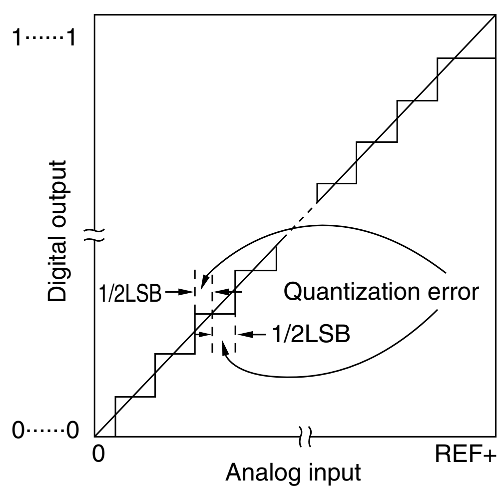
$1LSB = 1/2^{16} = 1/65536$
$\cong 0.0015\%FSR$

(2) Quantization error

When analog values are converted to digital values, a ±1/2LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2LSB is converted to the same digital code, so a quantization error cannot be avoided.

Note that the quantization error is not included in the offset, gain error, integral linearity error, and differential linearity error in the characteristics table.

Figure 13-15. Quantization Error

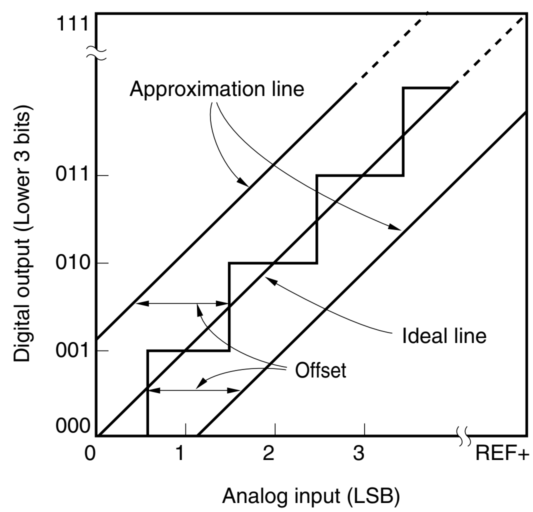


(3) Offset (Single input)

The offset represents the difference between the approximation line of the actually measured analog input voltage values and the theoretical values (1/2 LSB).

If the approximation line is greater than the theoretical values, it shows the difference between the approximation line of the actually measured analog input voltage values and the theoretical values (3/2 LSB).

Figure 13-16. Offset (Single Input)



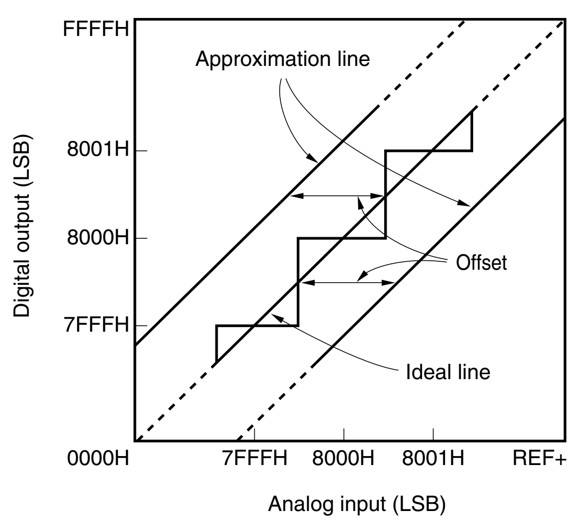
(4) Offset (Differential input)

The offset represents the difference between the approximation line of the actually measured analog input voltage values and the theoretical values (1/2 full-scale).

In the case of 16-bit resolution, the offset represents the difference between the approximation line of the actually measured analog input voltage values and the theoretical values (8000H − 1/2 LSB).

If the approximation line is greater than the theoretical values, it shows the difference between the approximation line of the actually measured analog input voltage values and the theoretical values (8000H + 1/2 LSB).

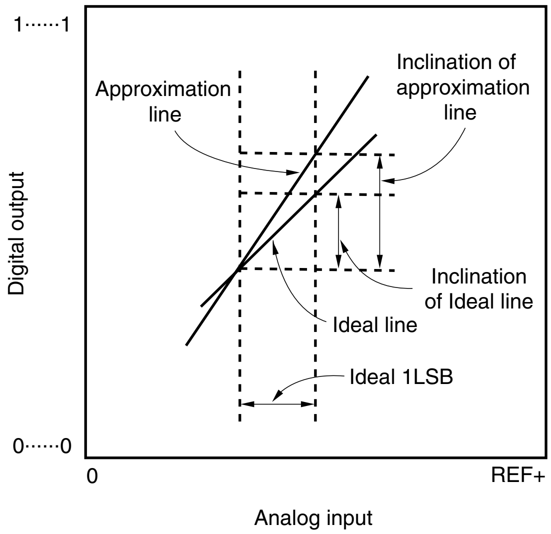
Figure 13-17. Offset (Differential Input)



(5) Gain error

The gain error is the ratio of the ideal inclination to the inclination of the approximation line.

Figure 13-18. Gain Error

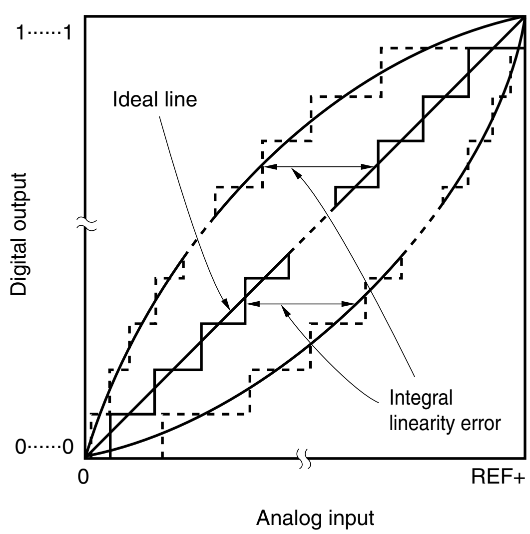


$$\text{Gain error} = \frac{\text{Inclination of approximation line} - 1}{\text{Inclination of Ideal line}} \times 100\ [\%]$$

(6) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the approximation line. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the offset and gain error are 0.

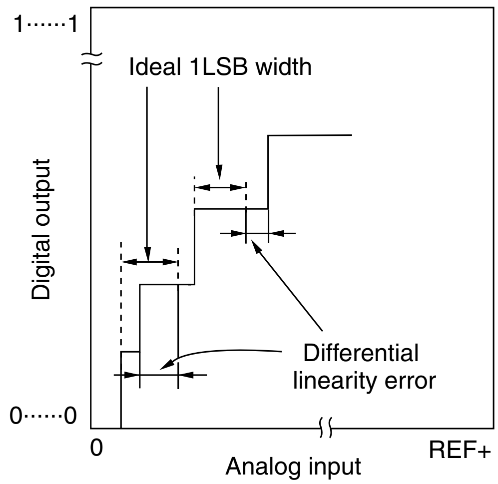
Figure 13-19. Integral linearity error



(7) Differential linearity error

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value. However, excluding the gain error.

Figure 13-20. Differential Linearity Error



(8) Conversion time

The conversion time is the time from starting conversion or obtaining the conversion result to obtaining the next conversion result.

For details, see **Figure 13-14. Conversion Time and Sampling Time**.

(9) Sampling time

The sampling time is the time required to perform one conversion.

For details, see **Figure 13-14. Conversion Time and Sampling Time**.

(10) Approximation line

The approximation line is the line defined by applying the least-squares method to the actually measured values.

13.7 Cautions for 16-Bit ΔΣ Type A/D Converter

(1) Setting standby mode

Clear bit 7 (ADDPON) and bit 6 (ADDCE) of the 16-bit ΔΣ type A/D converter control register 0 (ADDCTL0) to 0 before setting to the STOP mode.

To restart from the standby status, clear bit 6 (DSADIF) of interrupt request flag register 1L (IF1L) to 0 and start operation.

(2) Input range of DS0−, DS0+, DS1−, DS1+, DS2−, and DS2+

Observe the rated range of the DS0−, DS0+, DS1−, DS1+, DS2−, and DS2+ input voltage. If a voltage of REF+ (AV_{REF}) or higher, or REF− (AV_{SS}) or lower (even in the range of absolute maximum ratings) is input to an analog input channel, the converted value of that channel becomes undefined. In addition, the converted values of the other channels may also be affected.

(3) Conflicting operations

<1> Conflict between A/D conversion result register (ADDCR, ADDCRH) write and ADDCR or ADDCRH read by instruction upon the end of conversion
ADDCR or ADDCRH read has priority. After the read operation, the new conversion result is written to ADDCR or ADDCRH.

<2> Conflict between ADDCR or ADDCRH write and 16-bit ΔΣ type A/D converter control register 0 (ADDCTL0) write or A/D port configuration register 0 (ADPC0) write upon the end of conversion
ADDCTL0 or ADPC0 write has priority. ADDCR or ADDCRH write is not performed, nor is the conversion end interrupt signal (INTDSAD) generated.

(4) Noise countermeasures

To maintain the accuracy of specification, attention must be paid to noise input to the DS0−, DS0+, DS1−, DS1+, DS2−, DS2+, REF− (AV_{SS}), and REF+ (AV_{REF}) pins.

<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

<2> The higher the output impedance of the analog input source, the greater the influence. To reduce the noise, connecting external capacitor is recommended.

<3> Do not switch these pins with other pins during conversion.

<4> The accuracy may be improved if the HALT mode is set immediately after the start of conversion.

(5) DS0−/ANI0/P20, DS0+/ANI1/P21, DS1−/ANI2/P22, DS1+/ANI3/P23, DS2−/ANI4/P24, DS2+/ANI5/P25, REF−/ANI6/P26, and REF+/ANI7/P27

<1> The analog input pins (DS0−, DS0+, DS1−, DS1+, DS2−, DS2+, REF−, and REF+) are also used as I/O port pins (P20 to P27).
When 16-bit ΔΣ type A/D conversion is performed with any of DS0−/DS0+, DS1−/DS1+, or DS2−/DS2+ selected, do not access P20 to P27 while conversion is in progress; otherwise the accuracy may be degraded. It is recommended to any pin of P20 to P27 used as digital I/O port starting with the DS0−/ANI0/P20 that is the furthest from REF+.

<2> If any pin among pins P20 to P27 is used as a digital I/O port during 16-bit ΔΣ type A/D conversion, the expected value of the A/D conversion may not be obtained due to coupling noise. Make sure that digital pulses are not input to or output from pins P20 to P27 during A/D conversion.

(6) Input impedance of DS0+, DS1+, DS2+, DS0−, DS1−, and DS2− pins

This A/D converter charges a sampling capacitor for sampling during sampling time.

Therefore, only a leakage current flow when sampling is not in progress, and a current that charges the capacitor flows during sampling. Consequently, the input impedance fluctuates depending on whether sampling is in progress, and on the other states.

To make sure that sampling is effective, however, it is recommended to keep the output impedance of the analog input source to within 5 kΩ.

(7) AV_{REF} pin input impedance

A current flows to the AV_{REF} pin while the 16-bit ΔΣ type A/D converter operates.

If the output impedance of the reference voltage source is high, the error of the reference voltage between the AV_{REF} pin/REF+ pin and AV_{SS} pin/REF− pin increases.

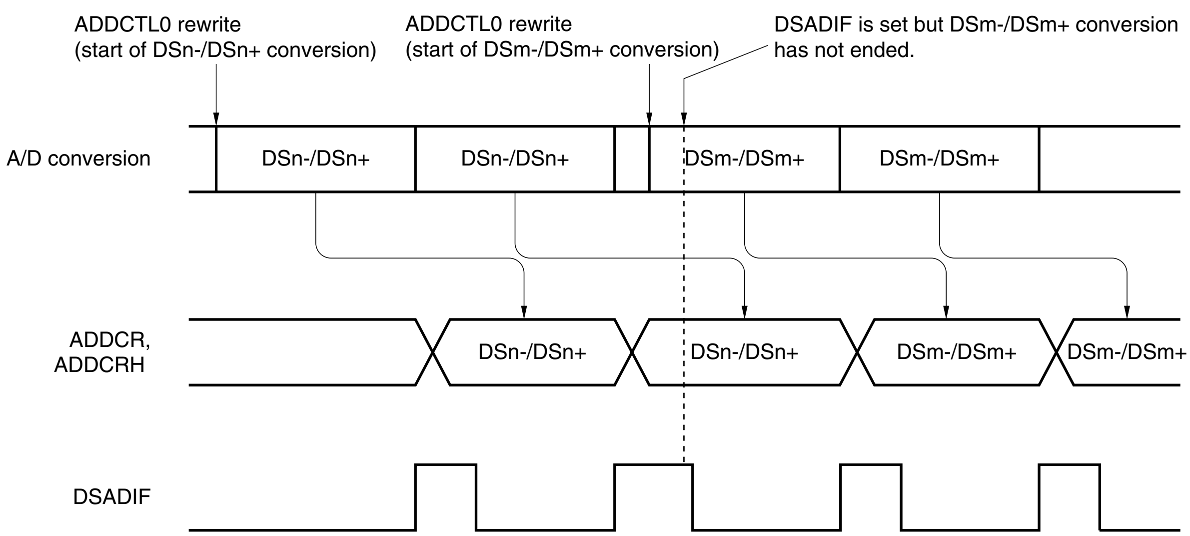
(8) Interrupt request flag (DSADIF)

The interrupt request flag (DSADIF) is not cleared even if bit 1 and 0 (ADDS1 and ADDS0) of the 16-bit ΔΣ type A/D converter control register 0 (ADDCTL0) is changed.

Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and DSADIF for the pre-change analog input may be set just before the ADDCTL0 rewrite. Caution is therefore required since, at this time, when DSADIF is read immediately after the ADDCTL0 rewrite, DSADIF is set despite the fact A/D conversion for the post-change analog input has not ended.

When A/D conversion is stopped and then resumed, clear DSADIF before the A/D conversion operation is resumed.

Figure 13-21. Timing of A/D Conversion End Interrupt Request Generation



Remarks 1. n = 0 to 2

2. m = 0 to 2

(9) Conversion results just after A/D conversion start

The first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADDCE bit is set to 1 within 1.2 μs after the ADDPON bit was set to 1, or if the ADDCE bit is set to 1 with the ADDPON bit = 0. Take measures such as polling the A/D conversion end interrupt request (INTDSAD) and removing the first conversion result.

(10) Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 13-22. Internal Equivalent Circuit of DSn− and DSn+ Pin

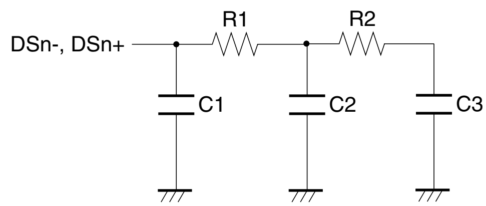


Table 13-5. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{REF}	R1	R2	C1	C2	C3
$4.0\ V \leq AV_{REF} \leq 5.5\ V$	8.1 kΩ	6.8 kΩ	8 pF	1.3 pF	0.22 pF
$2.7\ V \leq AV_{REF} < 4.0\ V$	31 kΩ	36 kΩ	8 pF	1.3 pF	0.22 pF

Remarks 1. The resistance and capacitance values shown in Table 13-5 are not guaranteed values.

2. n = 0 to 2

(11) Simultaneous use of the 10-bit successive approximation type A/D converter and the 16-bit ΔΣ type A/D converter

The A/D conversion accuracy may deteriorate when the 10-bit successive approximation type A/D converter and the 16-bit ΔΣ type A/D converter are used at the same time.

Stop the 16-bit ΔΣ type A/D converter during 10-bit successive approximation type A/D converter operation, because the accuracy cannot be guaranteed. Also, stop the 10-bit successive approximation type A/D converter during 16-bit ΔΣ type A/D converter operation. (Do not operate them simultaneously.)

CHAPTER 14 SERIAL INTERFACE UART0

14.1 Functions of Serial Interface UART0

Serial interface UART0 has the following two modes.

(1) Operation stop mode

This mode is used when serial communication is not executed and can enable a reduction in the power consumption.

For details, see **14.4.1 Operation stop mode**.

(2) Asynchronous serial interface (UART) mode

The functions of this mode are outlined below.

For details, see **14.4.2 Asynchronous serial interface (UART) mode** and **14.4.3 Dedicated baud rate generator**.

- Maximum transfer rate: 625 kbps
- Two-pin configuration TxD0: Transmit data output pin
  RxD0: Receive data input pin
- Length of communication data can be selected from 7 or 8 bits.
- Dedicated on-chip 5-bit baud rate generator allowing any baud rate to be set
- Transmission and reception can be performed independently (full-duplex operation).
- Fixed to LSB-first communication

Cautions 1. If clock supply to serial interface UART0 is not stopped (e.g., in the HALT mode), normal operation continues. If clock supply to serial interface UART0 is stopped (e.g., in the STOP mode), each register stops operating, and holds the value immediately before clock supply was stopped. The TxD0 pin also holds the value immediately before clock supply was stopped and outputs it. However, the operation is not guaranteed after clock supply is resumed. Therefore, reset the circuit so that POWER0 = 0, RXE0 = 0, and TXE0 = 0.

2. Set POWER0 = 1 and then set TXE0 = 1 (transmission) or RXE0 = 1 (reception) to start communication.

3. TXE0 and RXE0 are synchronized by the base clock (f_{XCLK0}) set by BRGC0. To enable transmission or reception again, set TXE0 or RXE0 to 1 at least two clocks of base clock after TXE0 or RXE0 has been cleared to 0. If TXE0 or RXE0 is set within two clocks of base clock, the transmission circuit or reception circuit may not be initialized.

4. Set transmit data to TXS0 at least one base clock (f_{XCLK0}) after setting TXE0 = 1.

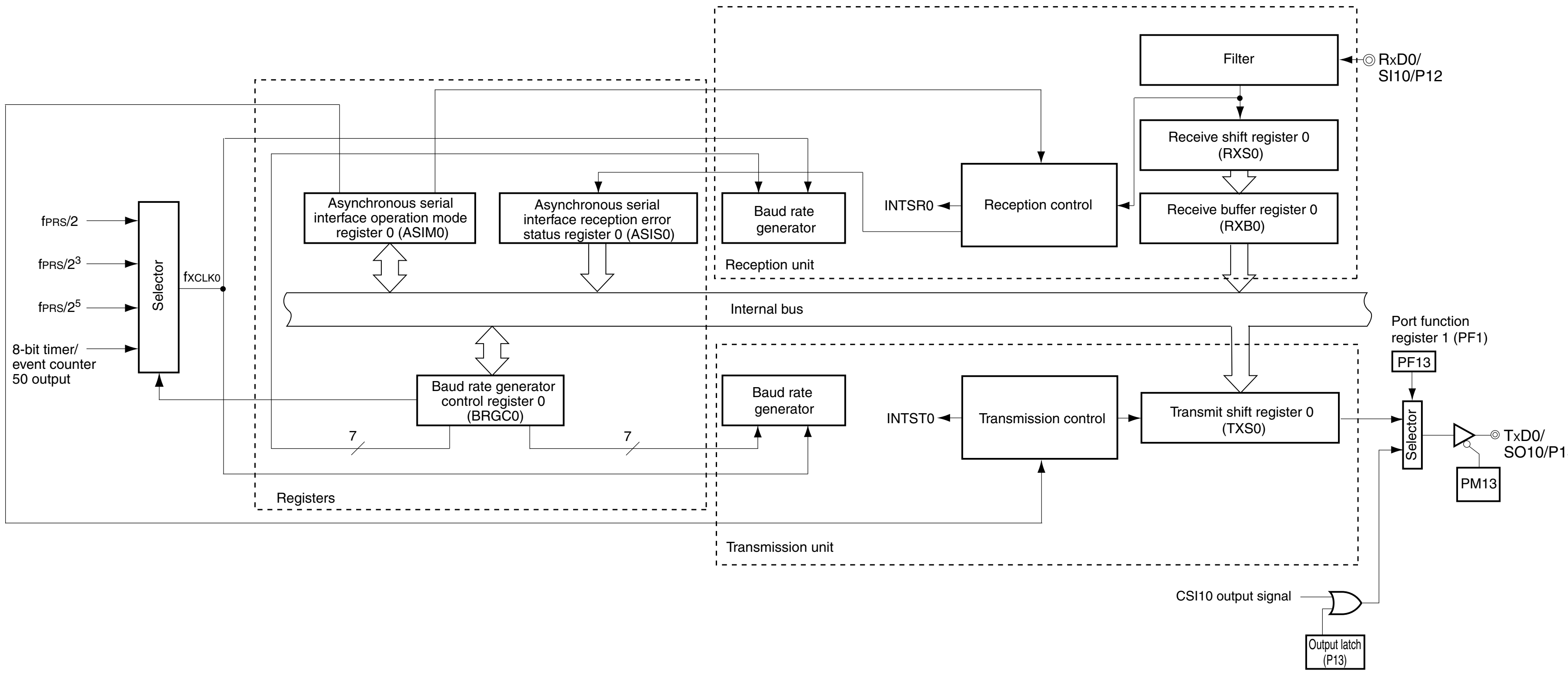
14.2 Configuration of Serial Interface UART0

Serial interface UART0 includes the following hardware.

Table 14-1. Configuration of Serial Interface UART0

Item	Configuration
Registers	Receive buffer register 0 (RXB0) Receive shift register 0 (RXS0) Transmit shift register 0 (TXS0)
Control registers	Asynchronous serial interface operation mode register 0 (ASIM0) Asynchronous serial interface reception error status register 0 (ASIS0) Baud rate generator control register 0 (BRGC0) Port function register 1 (PF1) Port mode register 1 (PM1) Port register 1 (P1)

Figure 14-1. Block Diagram of Serial Interface UART0



(1) Receive buffer register 0 (RXB0)

This 8-bit register stores parallel data converted by receive shift register 0 (RXS0).

Each time 1 byte of data has been received, new receive data is transferred to this register from receive shift register 0 (RXS0).

If the data length is set to 7 bits the receive data is transferred to bits 0 to 6 of RXB0 and the MSB of RXB0 is always 0.

If an overrun error (OVE0) occurs, the receive data is not transferred to RXB0.

RXB0 can be read by an 8-bit memory manipulation instruction. No data can be written to this register.

Reset signal generation and POWER0 = 0 set this register to FFH.

(2) Receive shift register 0 (RXS0)

This register converts the serial data input to the RxD0 pin into parallel data.

RXS0 cannot be directly manipulated by a program.

(3) Transmit shift register 0 (TXS0)

This register is used to set transmit data. Transmission is started when data is written to TXS0, and serial data is transmitted from the TxD0 pins.

TXS0 can be written by an 8-bit memory manipulation instruction. This register cannot be read.

Reset signal generation, POWER0 = 0, and TXE0 = 0 set this register to FFH.

Cautions 1. Set transmit data to TXS0 at least one base clock (f_{XCLK0}) after setting TXE0 = 1.

2. Do not write the next transmit data to TXS0 before the transmission completion interrupt signal (INTST0) is generated.

14.3 Registers Controlling Serial Interface UART0

Serial interface UART0 is controlled by the following six registers.

- Asynchronous serial interface operation mode register 0 (ASIM0)
- Asynchronous serial interface reception error status register 0 (ASIS0)
- Baud rate generator control register 0 (BRGC0)
- Port function register 1 (PF1)
- Port mode register 1 (PM1)
- Port register 1 (P1)

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

This 8-bit register controls the serial communication operations of serial interface UART0.
This register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to 01H.

Figure 14-2. Format of Asynchronous Serial Interface Operation Mode Register 0 (ASIM0) (1/2)

Address: FF70H After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM0	POWER0	TXE0	RXE0	PS01	PS00	CL0	SL0	1

POWER0	Enables/disables operation of internal operation clock
0[Note 1]	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit[Note 2].
1	Enables operation of the internal operation clock.

TXE0	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).
1	Enables transmission.

RXE0	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).
1	Enables reception.

Notes 1. The input from the RxD0 pin is fixed to high level when POWER0 = 0.

2. Asynchronous serial interface reception error status register 0 (ASIS0), transmit shift register 0 (TXS0), and receive buffer register 0 (RXB0) are reset.

Figure 14-2. Format of Asynchronous Serial Interface Operation Mode Register 0 (ASIM0) (2/2)

PS01	PS00	Transmission operation	Reception operation
0	0	Does not output parity bit.	Reception without parity
0	1	Outputs 0 parity.	Reception as 0 parity[Note]
1	0	Outputs odd parity.	Judges as odd parity.
1	1	Outputs even parity.	Judges as even parity.

CL0	Specifies character length of transmit/receive data
0	Character length of data = 7 bits
1	Character length of data = 8 bits

SL0	Specifies number of stop bits of transmit data
0	Number of stop bits = 1
1	Number of stop bits = 2

Note If "reception as 0 parity" is selected, the parity is not judged. Therefore, bit 2 (PE0) of asynchronous serial interface reception error status register 0 (ASIS0) is not set and the error interrupt does not occur.

Cautions
1. **To start the transmission, set POWER0 to 1 and then set TXE0 to 1. To stop the transmission, clear TXE0 to 0, and then clear POWER0 to 0.**
2. **To start the reception, set POWER0 to 1 and then set RXE0 to 1. To stop the reception, clear RXE0 to 0, and then clear POWER0 to 0.**
3. **Set POWER0 to 1 and then set RXE0 to 1 while a high level is input to the RxD0 pin. If POWER0 is set to 1 and RXE0 is set to 1 while a low level is input, reception is started.**
4. **TXE0 and RXE0 are synchronized by the base clock (f_{XCLK0}) set by BRGC0. To enable transmission or reception again, set TXE0 or RXE0 to 1 at least two clocks of base clock after TXE0 or RXE0 has been cleared to 0. If TXE0 or RXE0 is set within two clocks of base clock, the transmission circuit or reception circuit may not be initialized.**
5. **Set transmit data to TXS0 at least one base clock (f_{XCLK0}) after setting TXE0 = 1.**
6. **Clear the TXE0 and RXE0 bits to 0 before rewriting the PS01, PS00, and CL0 bits.**
7. **Make sure that TXE0 = 0 when rewriting the SL0 bit. Reception is always performed with "number of stop bits = 1", and therefore, is not affected by the set value of the SL0 bit.**
8. **Be sure to set bit 0 to 1.**

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

This register indicates an error status on completion of reception by serial interface UART0. It includes three error flag bits (PE0, FE0, OVE0).

This register is read-only by an 8-bit memory manipulation instruction.

Reset signal generation, or clearing bit 7 (POWER0) or bit 5 (RXE0) of ASIM0 to 0 clears this register to 00H. 00H is read when this register is read. If a reception error occurs, read ASIS0 and then read receive buffer register 0 (RXB0) to clear the error flag.

Figure 14-3. Format of Asynchronous Serial Interface Reception Error Status Register 0 (ASIS0)

Address: FF73H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS0	0	0	0	0	0	PE0	FE0	OVE0

PE0	Status flag indicating parity error
0	If POWER0 = 0 or RXE0 = 0, or if ASIS0 register is read.
1	If the parity of transmit data does not match the parity bit on completion of reception.

FE0	Status flag indicating framing error
0	If POWER0 = 0 or RXE0 = 0, or if ASIS0 register is read.
1	If the stop bit is not detected on completion of reception.

OVE0	Status flag indicating overrun error
0	If POWER0 = 0 and RXE0 = 0, or if ASIS0 register is read.
1	If receive data is set to the RXB0 register and the next reception operation is completed before the data is read.

Cautions 1. The operation of the PE0 bit differs depending on the set values of the PS01 and PS00 bits of asynchronous serial interface operation mode register 0 (ASIM0).

2. Only the first bit of the receive data is checked as the stop bit, regardless of the number of stop bits.

3. If an overrun error occurs, the next receive data is not written to receive buffer register 0 (RXB0) but discarded.

4. If data is read from ASIS0, a wait cycle is generated. Do not read data from ASIS0 when the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped. For details, see CHAPTER 34 CAUTIONS FOR WAIT.

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

This register selects the base clock of serial interface UART0 and the division value of the 5-bit counter.

BRGC0 can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 1FH.

Figure 14-4. Format of Baud Rate Generator Control Register 0 (BRGC0)

Address: FF71H After reset: 1FH R/W

Symbol	7	6	5	4	3	2	1	0
BRGC0	TPS01	TPS00	0	MDL04	MDL03	MDL02	MDL01	MDL00

TPS01	TPS00	Base clock (f_{XCLK0}) selection[Note 1]				
			f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 8 MHz	f_{PRS} = 10 MHz
0	0	TM50 output[Note 2]				
0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	4 MHz	5 MHz
1	0	$f_{PRS}/2^3$	250 kHz	625 kHz	1 MHz	1.25 MHz
1	1	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	250 kHz	312.5 kHz

MDL04	MDL03	MDL02	MDL01	MDL00	k	Selection of 5-bit counter output clock
0	0	×	×	×	×	Setting prohibited
0	1	0	0	0	8	$f_{XCLK0}/8$
0	1	0	0	1	9	$f_{XCLK0}/9$
0	1	0	1	0	10	$f_{XCLK0}/10$
• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •
1	1	0	1	0	26	$f_{XCLK0}/26$
1	1	0	1	1	27	$f_{XCLK0}/27$
1	1	1	0	0	28	$f_{XCLK0}/28$
1	1	1	0	1	29	$f_{XCLK0}/29$
1	1	1	1	0	30	$f_{XCLK0}/30$
1	1	1	1	1	31	$f_{XCLK0}/31$

Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{XH}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 2.7 to 5.5 V: $f_{PRS} \leq 10$ MHz
- V_{DD} = 1.8 to 2.7 V: $f_{PRS} \leq 5$ MHz

2. Note the following points when selecting the TM50 output as the base clock.

- Mode in which the count clock is cleared and started upon a match of TM50 and CR50 (TMC506 = 0)
  Start the operation of 8-bit timer/event counter 50 first and then enable the timer F/F inversion operation (TMC501 = 1).
- PWM mode (TMC506 = 1)
  Start the operation of 8-bit timer/event counter 50 first and then set the count clock to make the duty = 50%.

It is not necessary to enable (TOE50 = 1) TO50 output in any mode.

Cautions 1. Make sure that bit 6 (TXE0) and bit 5 (RXE0) of the ASIM0 register = 0 when rewriting the MDL04 to MDL00 bits.

2. The baud rate value is the output clock of the 5-bit counter divided by 2.

Remarks 1. f_{XCLK0}: Frequency of base clock selected by the TPS01 and TPS00 bits

2. f_{PRS}: Peripheral hardware clock frequency

3. k: Value set by the MDL04 to MDL00 bits (k = 8, 9, 10, ..., 31)

4. ×: Don't care

5. TMC506: Bit 6 of 8-bit timer mode control register 50 (TMC50)
TMC501: Bit 1 of TMC50

(4) Port function register 1 (PF1)

This register sets the pin functions of P13/SO10/TxD0 pin.

PF1 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PF1 to 00H.

Figure 14-5. Format of Port Function Register 1 (PF1)

Address: FF20H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PF1	0	PF16	0	0	PF13	0	0	0

PF16	Port (P16), CSIA0, and UART6 output specification
0	Used as P16 or SOA0
1	Used as TxD6

PF13	Port (P13), CSI10, and UART0 output specification
0	Used as P13 or SO10
1	Used as TxD0

(5) Port mode register 1 (PM1)

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

When using the P13/SO10/TxD0 pin for serial interface data output, clear PM13 to 0. The output latch of P13 at this time may be 0 or 1.

When using the P12/SI10/RxD0 pin for serial interface data input, set PM12 to 1. The output latch of P12 at this time may be 0 or 1.

PM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 14-6. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

14.4 Operation of Serial Interface UART0

Serial interface UART0 has the following two modes.

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

14.4.1 Operation stop mode

In this mode, serial communication cannot be executed, thus reducing the power consumption. In addition, the pins can be used as ordinary port pins in this mode. To set the operation stop mode, clear bits 7, 6, and 5 (POWER0, TXE0, and RXE0) of ASIM0 to 0.

(1) Register used

The operation stop mode is set by asynchronous serial interface operation mode register 0 (ASIM0).
ASIM0 can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to 01H.

Address: FF70H After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM0	POWER0	TXE0	RXE0	PS01	PS00	CL0	SL0	1

POWER0	Enables/disables operation of internal operation clock
0[Note 1]	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit[Note 2].

TXE0	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).

RXE0	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).

Notes 1. The input from the RxD0 pin is fixed to high level when POWER0 = 0.

2. Asynchronous serial interface reception error status register 0 (ASIS0), transmit shift register 0 (TXS0), and receive buffer register 0 (RXB0) are reset.

Caution Clear POWER0 to 0 after clearing TXE0 and RXE0 to 0 to set the operation stop mode.
To start the communication, set POWER0 to 1, and then set TXE0 or RXE0 to 1.

Remark To use the RxD0/SI10/P12 and TxD0/SO10/P13 pins as general-purpose port pins, see **CHAPTER 4 PORT FUNCTIONS**.

14.4.2 Asynchronous serial interface (UART) mode

In this mode, 1-byte data is transmitted/received following a start bit, and a full-duplex operation can be performed.

A dedicated UART baud rate generator is incorporated, so that communication can be executed at a wide range of baud rates.

(1) Registers used

- Asynchronous serial interface operation mode register 0 (ASIM0)
- Asynchronous serial interface reception error status register 0 (ASIS0)
- Baud rate generator control register 0 (BRGC0)
- Port mode register 1 (PM1)
- Port register 1 (P1)

The basic procedure of setting an operation in the UART mode is as follows.

<1> Set the BRGC0 register (see **Figure 14-4**).
<2> Set bits 1 to 4 (SL0, CL0, PS00, and PS01) of the ASIM0 register (see **Figure 14-2**).
<3> Set bit 7 (POWER0) of the ASIM0 register to 1.
<4> Set bit 6 (TXE0) of the ASIM0 register to 1. → Transmission is enabled.
Set bit 5 (RXE0) of the ASIM0 register to 1. → Reception is enabled.
<5> Write data to the TXS0 register. → Data transmission is started.

Caution Take relationship with the other party of communication when setting the port mode register and port register.

The relationship between the register settings and pins is shown below.

Table 14-2. Relationship Between Register Settings and Pins

POWER0	TXE0	RXE0	PM13	P13	PM12	P12	UART0 Operation	Pin Function	
								TxD0/SO10/P13	RxD0/SI10/P12
0	0	0	×[Note]	×[Note]	×[Note]	×[Note]	Stop	SO10/P13	SI10/P12
1	0	1	×[Note]	×[Note]	1	×	Reception	SO10/P13	RxD0
	1	0	0	×	×[Note]	×[Note]	Transmission	TxD0	SI10/P12
	1	1	0	×	1	×	Transmission/ reception	TxD0	RxD0

Note Can be set as port function or serial interface CSI10.

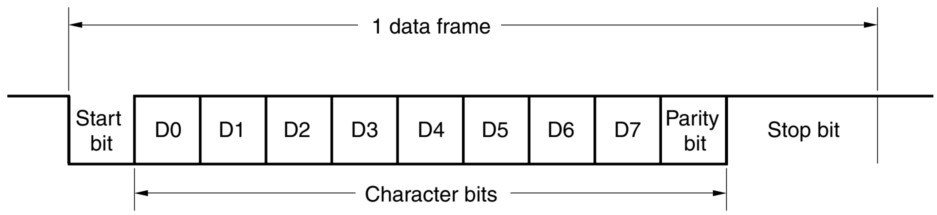
Remark ×: don't care
POWER0: Bit 7 of asynchronous serial interface operation mode register 0 (ASIM0)
TXE0: Bit 6 of ASIM0
RXE0: Bit 5 of ASIM0
PM1×: Port mode register
P1×: Port output latch

(2) Communication operation

(a) Format and waveform example of normal transmit/receive data

Figures 14-7 and 14-8 show the format and waveform example of the normal transmit/receive data.

Figure 14-7. Format of Normal UART Transmit/Receive Data



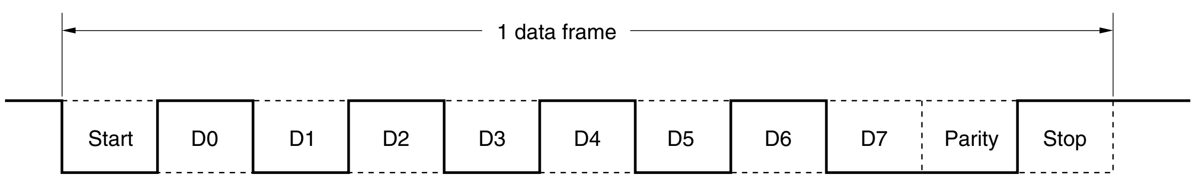
One data frame consists of the following bits.

- Start bit ... 1 bit
- Character bits ... 7 or 8 bits (LSB first)
- Parity bit ... Even parity, odd parity, 0 parity, or no parity
- Stop bit ... 1 or 2 bits

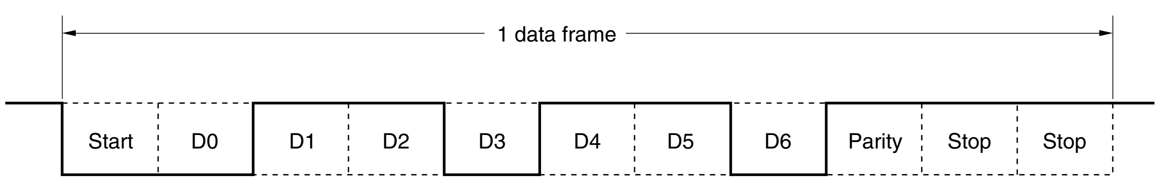
The character bit length, parity, and stop bit length in one data frame are specified by asynchronous serial interface operation mode register 0 (ASIM0).

Figure 14-8. Example of Normal UART Transmit/Receive Data Waveform

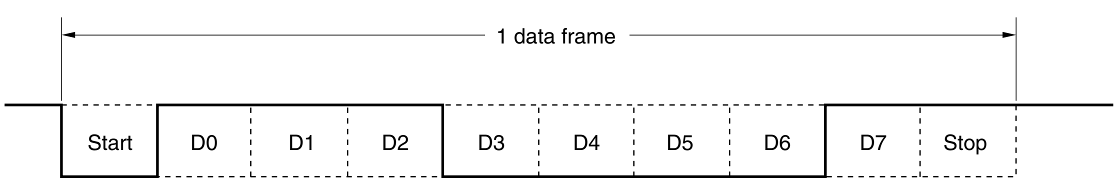
1. Data length: 8 bits, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H



2. Data length: 7 bits, Parity: Odd parity, Stop bit: 2 bits, Communication data: 36H



3. Data length: 8 bits, Parity: None, Stop bit: 1 bit, Communication data: 87H



(b) Parity types and operation

The parity bit is used to detect a bit error in communication data. Usually, the same type of parity bit is used on both the transmission and reception sides. With even parity and odd parity, a 1-bit (odd number) error can be detected. With zero parity and no parity, an error cannot be detected.

(i) Even parity

- Transmission

  Transmit data, including the parity bit, is controlled so that the number of bits that are “1” is even.
  The value of the parity bit is as follows.

  If transmit data has an odd number of bits that are “1”: 1
  If transmit data has an even number of bits that are “1”: 0

- Reception

  The number of bits that are “1” in the receive data, including the parity bit, is counted. If it is odd, a parity error occurs.

(ii) Odd parity

- Transmission

  Unlike even parity, transmit data, including the parity bit, is controlled so that the number of bits that are “1” is odd.

  If transmit data has an odd number of bits that are “1”: 0
  If transmit data has an even number of bits that are “1”: 1

- Reception

  The number of bits that are “1” in the receive data, including the parity bit, is counted. If it is even, a parity error occurs.

(iii) 0 parity

The parity bit is cleared to 0 when data is transmitted, regardless of the transmit data.
The parity bit is not detected when the data is received. Therefore, a parity error does not occur regardless of whether the parity bit is “0” or “1”.

(iv) No parity

No parity bit is appended to the transmit data.
Reception is performed assuming that there is no parity bit when data is received. Because there is no parity bit, a parity error does not occur.

(c) Transmission

If bit 7 (POWER0) of asynchronous serial interface operation mode register 0 (ASIM0) is set to 1 and bit 6 (TXE0) of ASIM0 is then set to 1, transmission is enabled. Transmission can be started by writing transmit data to transmit shift register 0 (TXS0). The start bit, parity bit, and stop bit are automatically appended to the data.

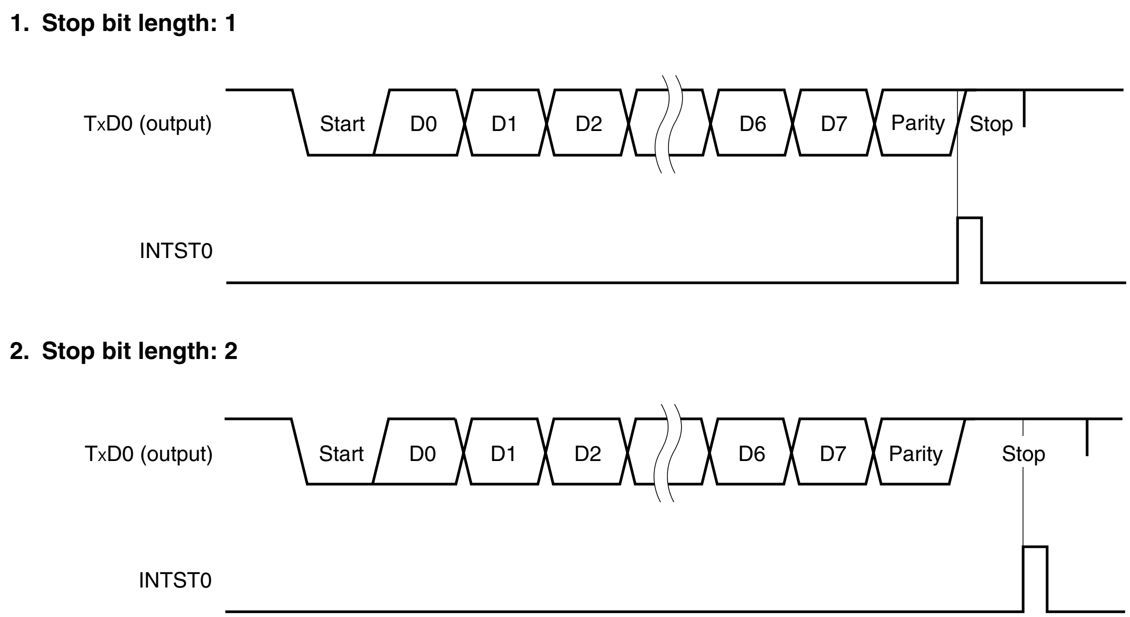
When transmission is started, the start bit is output from the TxD0 pin, and the transmit data is output followed by the rest of the data in order starting from the LSB. When transmission is completed, the parity and stop bits set by ASIM0 are appended and a transmission completion interrupt request (INTST0) is generated.

Transmission is stopped until the data to be transmitted next is written to TXS0.

Figure 14-9 shows the timing of the transmission completion interrupt request (INTST0). This interrupt occurs as soon as the last stop bit has been output.

Caution After transmit data is written to TXS0, do not write the next transmit data before the transmission completion interrupt signal (INTST0) is generated.

Figure 14-9. Transmission Completion Interrupt Request Timing



(d) Reception

Reception is enabled and the RxD0 pin input is sampled when bit 7 (POWER0) of asynchronous serial interface operation mode register 0 (ASIM0) is set to 1 and then bit 5 (RXE0) of ASIM0 is set to 1.

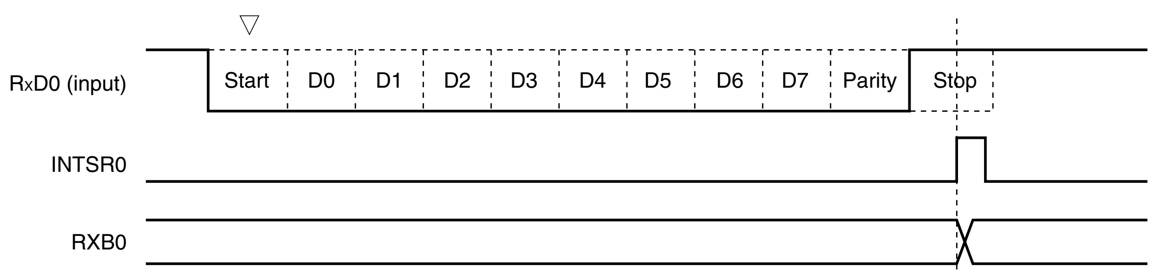
The 5-bit counter of the baud rate generator starts counting when the falling edge of the RxD0 pin input is detected. When the set value of baud rate generator control register 0 (BRGC0) has been counted, the RxD0 pin input is sampled again (▽in Figure 14-10). If the RxD0 pin is low level at this time, it is recognized as a start bit.

When the start bit is detected, reception is started, and serial data is sequentially stored in receive shift register 0 (RXS0) at the set baud rate. When the stop bit has been received, the reception completion interrupt (INTSR0) is generated and the data of RXS0 is written to receive buffer register 0 (RXB0). If an overrun error (OVE0) occurs, however, the receive data is not written to RXB0.

Even if a parity error (PE0) occurs while reception is in progress, reception continues to the reception position of the stop bit, and an reception error interrupt (INTSR0) is generated after completion of reception.

INTSR0 occurs upon completion of reception and in case of a reception error.

Figure 14-10. Reception Completion Interrupt Request Timing



Cautions 1. If a reception error occurs, read asynchronous serial interface reception error status register 0 (ASIS0) and then read receive buffer register 0 (RXB0) to clear the error flag. Otherwise, an overrun error will occur when the next data is received, and the reception error status will persist.

2. Reception is always performed with the “number of stop bits = 1”. The second stop bit is ignored.

(e) Reception error

Three types of errors may occur during reception: a parity error, framing error, or overrun error. If the error flag of asynchronous serial interface reception error status register 0 (ASIS0) is set as a result of data reception, a reception error interrupt (INTSR0) is generated.

Which error has occurred during reception can be identified by reading the contents of ASIS0 in the reception error interrupt (INTSR0) servicing (see **Figure 14-3**).

The contents of ASIS0 are cleared to 0 when ASIS0 is read.

Table 14-3. Cause of Reception Error

Reception Error	Cause
Parity error	The parity specified for transmission does not match the parity of the receive data.
Framing error	Stop bit is not detected.
Overrun error	Reception of the next data is completed before data is read from receive buffer register 0 (RXB0).

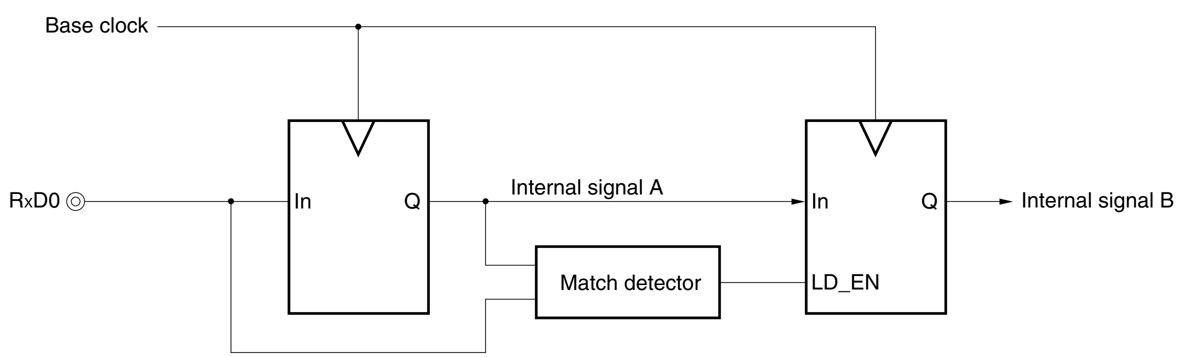
(f) Noise filter of receive data

The RxD0 signal is sampled using the base clock output by the prescaler block.

If two sampled values are the same, the output of the match detector changes, and the data is sampled as input data.

Because the circuit is configured as shown in Figure 14-11, the internal processing of the reception operation is delayed by two clocks from the external signal status.

Figure 14-11. Noise Filter Circuit



14.4.3 Dedicated baud rate generator

The dedicated baud rate generator consists of a source clock selector and a 5-bit programmable counter, and generates a serial clock for transmission/reception of UART0.

Separate 5-bit counters are provided for transmission and reception.

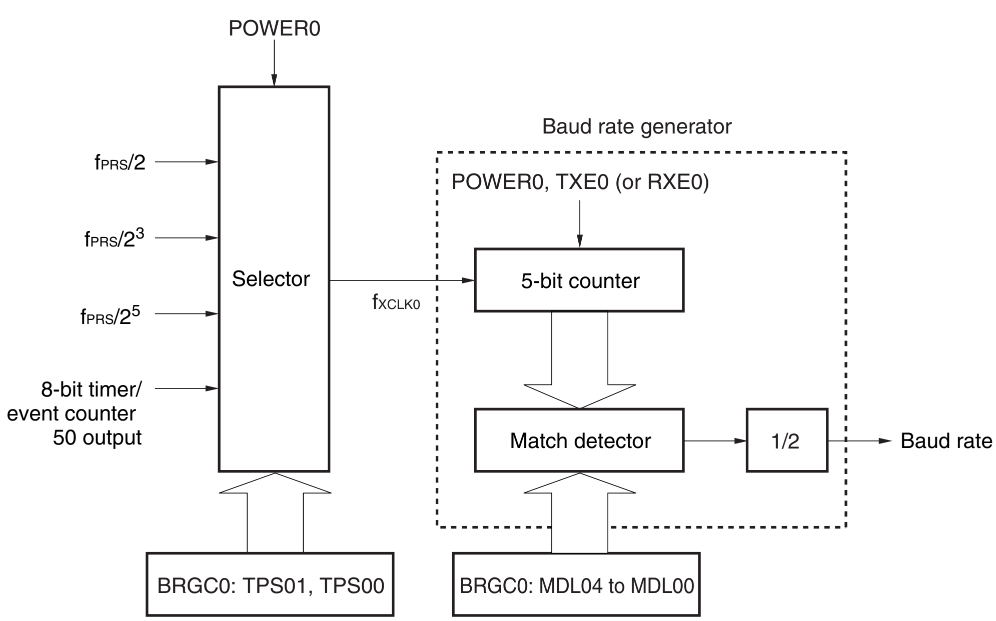
(1) Configuration of baud rate generator

- Base clock
  The clock selected by bits 7 and 6 (TPS01 and TPS00) of baud rate generator control register 0 (BRGC0) is supplied to each module when bit 7 (POWER0) of asynchronous serial interface operation mode register 0 (ASIM0) is 1. This clock is called the base clock and its frequency is called f_{XCLK0}. The base clock is fixed to low level when POWER0 = 0.

- Transmission counter
  This counter stops operation, cleared to 0, when bit 7 (POWER0) or bit 6 (TXE0) of asynchronous serial interface operation mode register 0 (ASIM0) is 0.
  It starts counting when POWER0 = 1 and TXE0 = 1.
  The counter is cleared to 0 when the first data transmitted is written to transmit shift register 0 (TXS0).

- Reception counter
  This counter stops operation, cleared to 0, when bit 7 (POWER0) or bit 5 (RXE0) of asynchronous serial interface operation mode register 0 (ASIM0) is 0.
  It starts counting when the start bit has been detected.
  The counter stops operation after one frame has been received, until the next start bit is detected.

Figure 14-12. Configuration of Baud Rate Generator



Remark POWER0: Bit 7 of asynchronous serial interface operation mode register 0 (ASIM0)
TXE0: Bit 6 of ASIM0
RXE0: Bit 5 of ASIM0
BRGC0: Baud rate generator control register 0

(2) Generation of serial clock

A serial clock to be generated can be specified by using baud rate generator control register 0 (BRGC0).

Select the clock to be input to the 5-bit counter by using bits 7 and 6 (TPS01 and TPS00) of BRGC0.

Bits 4 to 0 (MDL04 to MDL00) of BRGC0 can be used to select the division value ($f_{XCLK0}/8$ to $f_{XCLK0}/31$) of the 5-bit counter.

14.4.4 Calculation of baud rate

(1) Baud rate calculation expression

The baud rate can be calculated by the following expression.

- Baud rate = $\frac{f_{XCLK0}}{2 \times k}$ [bps]

f_{XCLK0}: Frequency of base clock selected by the TPS01 and TPS00 bits of the BRGC0 register

k: Value set by the MDL04 to MDL00 bits of the BRGC0 register (k = 8, 9, 10, ..., 31)

Table 14-4. Set Value of TPS01 and TPS00

TPS01	TPS00	Base clock (f_{XCLK0}) selection[Note 1]				
			f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 8 MHz	f_{PRS} = 10 MHz
0	0	TM50 output[Note 2]				
0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	4 MHz	5 MHz
1	0	$f_{PRS}/2^3$	250 kHz	625 kHz	1 MHz	1.25 MHz
1	1	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	250 kHz	312.5 kHz

Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{XH}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 2.7 to 5.5 V: $f_{PRS} \leq 10$ MHz
- V_{DD} = 1.8 to 2.7 V: $f_{PRS} \leq 5$ MHz

2. Note the following points when selecting the TM50 output as the base clock.

- Mode in which the count clock is cleared and started upon a match of TM50 and CR50 (TMC506 = 0)
  Start the operation of 8-bit timer/event counter 50 first and then enable the timer F/F inversion operation (TMC501 = 1).
- PWM mode (TMC506 = 1)
  Start the operation of 8-bit timer/event counter 50 first and then set the count clock to make the duty = 50%.

It is not necessary to enable (TOE50 = 1) TO50 output in any mode.

(2) Error of baud rate

The baud rate error can be calculated by the following expression.

- Error (%) = $\left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (correct baud rate)}} - 1\right) \times 100$ [%]

Cautions 1. Keep the baud rate error during transmission to within the permissible error range at the reception destination.

2. Make sure that the baud rate error during reception satisfies the range shown in (4) Permissible baud rate range during reception.

Example: Frequency of base clock = 2.5 MHz = 2,500,000 Hz
Set value of MDL04 to MDL00 bits of BRGC0 register = 10000B (k = 16)
Target baud rate = 76,800 bps

Baud rate $= 2.5\text{ M}/(2 \times 16)$
$= 2{,}500{,}000/(2 \times 16) = 78{,}125$ [bps]

Error $= (78{,}125/76{,}800 - 1) \times 100$
$= 1.725$ [%]

(3) Example of setting baud rate

Table 14-5. Set Data of Baud Rate Generator

	Baud Rate [bps]	f_{PRS} = 2.0 MHz				f_{PRS} = 5.0 MHz				f_{PRS} = 10.0 MHz			
		TPS01, TPS00	k	Calculated Value	ERR [%]	TPS01, TPS00	k	Calculated Value	ERR [%]	TPS01, TPS00	k	Calculated Value	ERR [%]
	1200	3H	26	1202	0.16	–	–	–	–	–	–	–	–
	2400	3H	13	2404	0.16	–	–	–	–	–	–	–	–
	4800	2H	26	4808	0.16	3H	16	4883	1.73	–	–	–	–
	9600	2H	13	9615	0.16	3H	8	9766	1.73	3H	16	9766	1.73
	10400	2H	12	10417	0.16	2H	30	10417	0.16	3H	15	10417	0.16
	19200	1H	26	19231	0.16	2H	16	19531	1.73	3H	8	19531	1.73
	24000	1H	21	23810	−0.79	2H	13	24038	0.16	2H	26	24038	0.16
	31250	1H	16	31250	0	2H	10	31250	0	2H	20	31250	0
<R>	33600	1H	15	33333	−0.79	2H	9	34722	3.34	2H	19	32895	−2.1
	38400	1H	13	38462	0.16	2H	8	39063	1.73	2H	16	39063	1.73
	56000	1H	9	55556	−0.79	1H	22	56818	1.46	2H	11	56818	1.46
	62500	1H	8	62500	0	1H	20	62500	0	2H	10	62500	0
	76800	–	–	–	–	1H	16	78125	1.73	2H	8	78125	1.73
	115200	–	–	–	–	1H	11	113636	−1.36	1H	22	113636	−1.36
	153600	–	–	–	–	1H	8	156250	1.73	1H	16	156250	1.73
<R>	312500	–	–	–	–	–	–	–	–	1H	8	312500	0

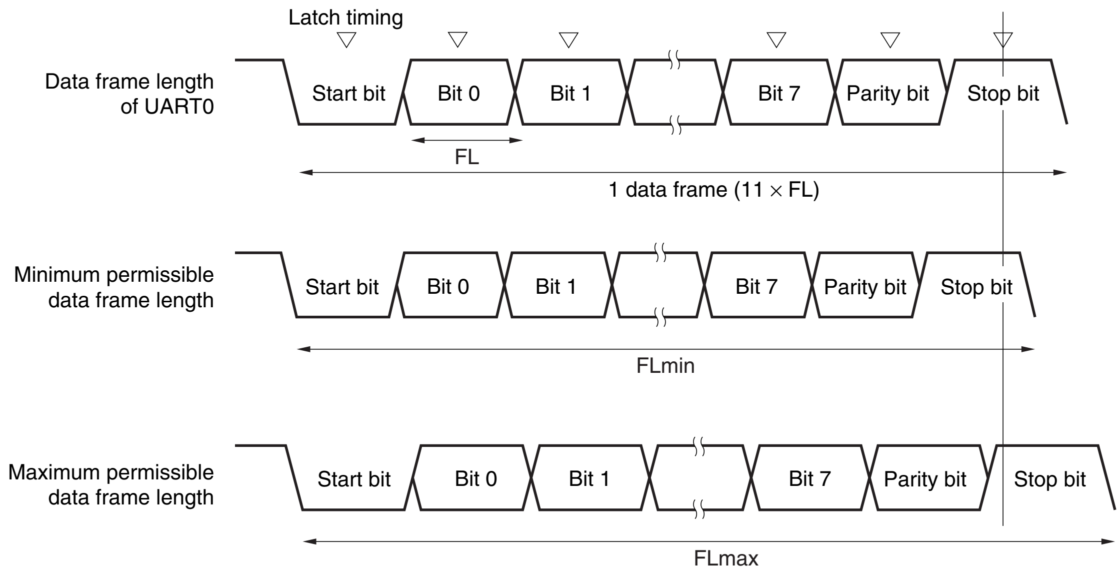
Remark TPS01, TPS00: Bits 7 and 6 of baud rate generator control register 0 (BRGC0) (setting of base clock (f_{XCLK0}))
k: Value set by the MDL04 to MDL00 bits of BRGC0 (k = 8, 9, 10, ..., 31)
f_{PRS}: Peripheral hardware clock frequency
ERR: Baud rate error

(4) Permissible baud rate range during reception

The permissible error from the baud rate at the transmission destination during reception is shown below.

Caution Make sure that the baud rate error during reception is within the permissible error range, by using the calculation expression shown below.

Figure 14-13. Permissible Baud Rate Range During Reception



As shown in Figure 14-13, the latch timing of the receive data is determined by the counter set by baud rate generator control register 0 (BRGC0) after the start bit has been detected. If the last data (stop bit) meets this latch timing, the data can be correctly received.

Assuming that 11-bit data is received, the theoretical values can be calculated as follows.

$FL = (Brate)^{-1}$

Brate: Baud rate of UART0

k: Set value of BRGC0

FL: 1-bit data length

Margin of latch timing: 2 clocks

Minimum permissible data frame length: $FLmin = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$

Therefore, the maximum receivable baud rate at the transmission destination is as follows.

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k+2} Brate$$

Similarly, the maximum permissible data frame length can be calculated as follows.

$$\frac{10}{11} \times FLmax = 11 \times FL - \frac{k+2}{2 \times k} \times FL = \frac{21k-2}{2 \times k} FL$$

$$FLmax = \frac{21k-2}{20k} FL \times 11$$

Therefore, the minimum receivable baud rate at the transmission destination is as follows.

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k-2} Brate$$

The permissible baud rate error between UART0 and the transmission destination can be calculated from the above minimum and maximum baud rate expressions, as follows.

Table 14-6. Maximum/Minimum Permissible Baud Rate Error

Division Ratio (k)	Maximum Permissible Baud Rate Error	Minimum Permissible Baud Rate Error
8	+3.53%	–3.61%
16	+4.14%	–4.19%
24	+4.34%	–4.38%
31	+4.44%	–4.47%

Remarks 1. The permissible error of reception depends on the number of bits in one frame, input clock frequency, and division ratio (k). The higher the input clock frequency and the higher the division ratio (k), the higher the permissible error.

2. k: Set value of BRGC0

CHAPTER 15 SERIAL INTERFACE UART6

15.1 Functions of Serial Interface UART6

Serial interface UART6 has the following two modes.

(1) Operation stop mode

This mode is used when serial communication is not executed and can enable a reduction in the power consumption.

For details, see **15.4.1 Operation stop mode**.

(2) Asynchronous serial interface (UART) mode

This mode supports the LIN (Local Interconnect Network)-bus. The functions of this mode are outlined below.

For details, see **15.4.2 Asynchronous serial interface (UART) mode** and **15.4.3 Dedicated baud rate generator**.

- Maximum transfer rate: 625 kbps
- Two-pin configuration TxD6: Transmit data output pin
  RxD6: Receive data input pin
- TxD6/RxD6 pins can be selected from P112/P113 (default) or P16/P15 by using the registers.
- Data length of communication data can be selected from 7 or 8 bits.
- Dedicated internal 8-bit baud rate generator allowing any baud rate to be set
- Transmission and reception can be performed independently (full duplex operation).
- MSB- or LSB-first communication selectable
- Inverted transmission operation
- Sync break field transmission from 13 to 20 bits
- More than 11 bits can be identified for sync break field reception (SBF reception flag provided).

Cautions 1. The TxD6 output inversion function inverts only the transmission side and not the reception side. To use this function, the reception side must be ready for reception of inverted data.

2. If clock supply to serial interface UART6 is not stopped (e.g., in the HALT mode), normal operation continues. If clock supply to serial interface UART6 is stopped (e.g., in the STOP mode), each register stops operating, and holds the value immediately before clock supply was stopped. The TxD6 pin also holds the value immediately before clock supply was stopped and outputs it. However, the operation is not guaranteed after clock supply is resumed. Therefore, reset the circuit so that POWER6 = 0, RXE6 = 0, and TXE6 = 0.

3. Set POWER6 = 1 and then set TXE6 = 1 (transmission) or RXE6 = 1 (reception) to start communication.

4. TXE6 and RXE6 are synchronized by the base clock (f_{XCLK6}) set by CKSR6. To enable transmission or reception again, set TXE6 or RXE6 to 1 at least two clocks of the base clock after TXE6 or RXE6 has been cleared to 0. If TXE6 or RXE6 is set within two clocks of the base clock, the transmission circuit or reception circuit may not be initialized.

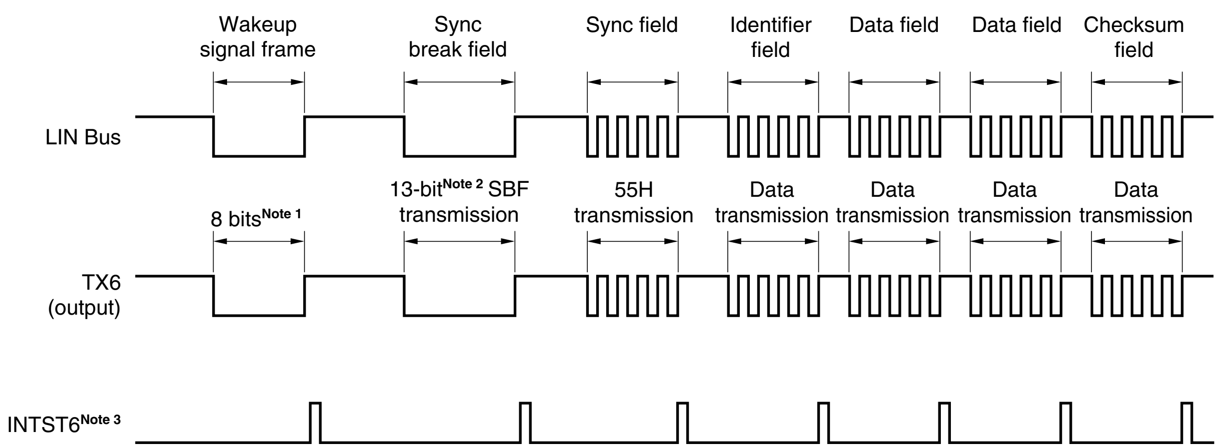
5. Set transmit data to TXB6 at least one base clock (f_{XCLK6}) after setting TXE6 = 1.

6. If data is continuously transmitted, the communication timing from the stop bit to the next start bit is extended two operating clocks of the macro. However, this does not affect the result of communication because the reception side initializes the timing when it has detected a start bit. Do not use the continuous transmission function if the interface is used in LIN communication operation.

Remark LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol intended to aid the cost reduction of an automotive network.
LIN communication is single-master communication, and up to 15 slaves can be connected to one master.
The LIN slaves are used to control the switches, actuators, and sensors, and these are connected to the LIN master via the LIN network.
Normally, the LIN master is connected to a network such as CAN (Controller Area Network).
In addition, the LIN bus uses a single-wire method and is connected to the nodes via a transceiver that complies with ISO9141.
In the LIN protocol, the master transmits a frame with baud rate information and the slave receives it and corrects the baud rate error. Therefore, communication is possible when the baud rate error in the slave is ±15% or less.

Figures 15-1 and 15-2 outline the transmission and reception operations of LIN.

Figure 15-1. LIN Transmission Operation



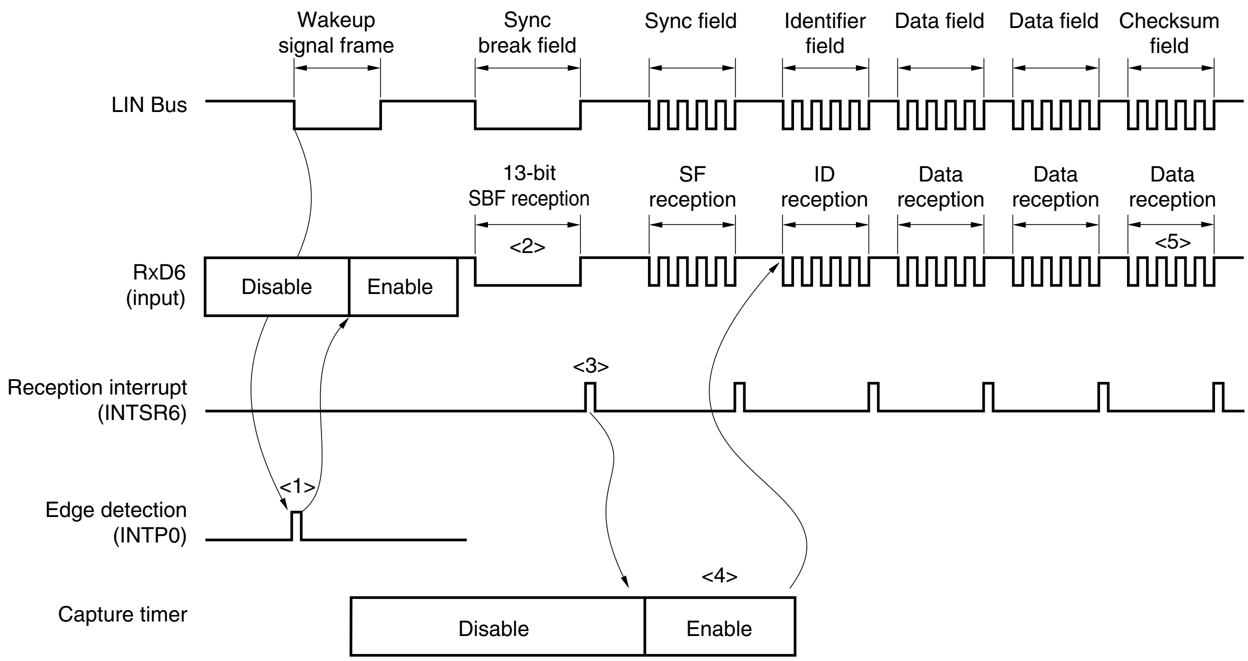
Notes 1. The wakeup signal frame is substituted by 80H transmission in the 8-bit mode.

2. The sync break field is output by hardware. The output width is the bit length set by bits 4 to 2 (SBL62 to SBL60) of asynchronous serial interface control register 6 (ASICL6) (see **15.4.2 (2) (h) SBF transmission**).

3. INTST6 is output on completion of each transmission. It is also output when SBF is transmitted.

Remark The interval between each field is controlled by software.

Figure 15-2. LIN Reception Operation



Reception processing is as follows.

<1> The wakeup signal is detected at the edge of the pin, and enables UART6 and sets the SBF reception mode.

<2> Reception continues until the STOP bit is detected. When an SBF with low-level data of 11 bits or more has been detected, it is assumed that SBF reception has been completed correctly, and an interrupt signal is output. If an SBF with low-level data of less than 11 bits has been detected, it is assumed that an SBF reception error has occurred. The interrupt signal is not output and the SBF reception mode is restored.

<3> If SBF reception has been completed correctly, an interrupt signal is output. Start 16-bit timer/event counter 00 by the SBF reception end interrupt servicing and measure the bit interval (pulse width) of the sync field (see **6.4.8 Pulse width measurement operation**). Detection of errors OVE6, PE6, and FE6 is suppressed, and error detection processing of UART communication and data transfer of the shift register and RXB6 is not performed. The shift register holds the reset value FFH.

<4> Calculate the baud rate error from the bit interval of the sync field, disable UART6 after SF reception, and then re-set baud rate generator control register 6 (BRGC6).

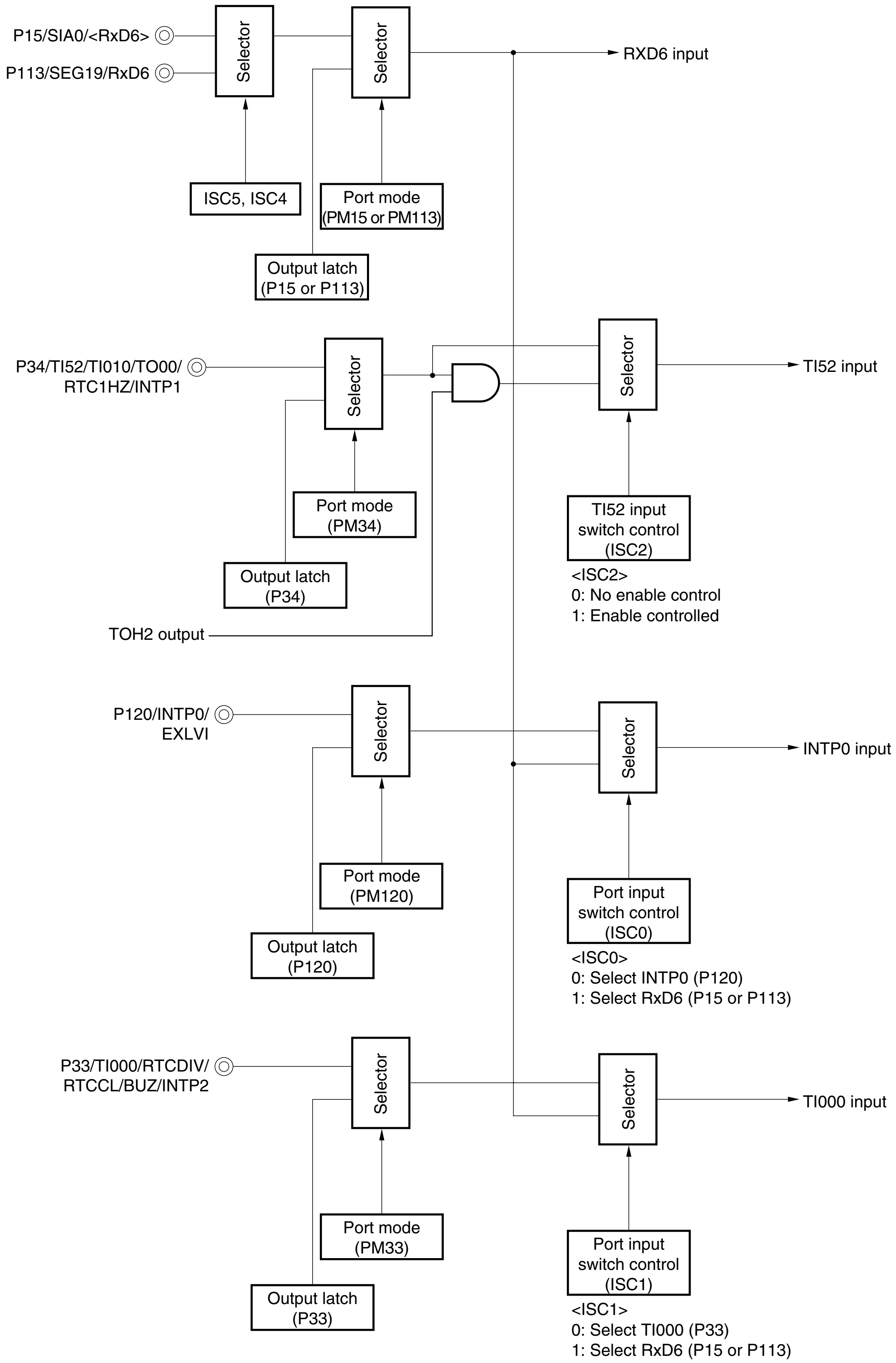
<5> Distinguish the checksum field by software. Also perform processing by software to initialize UART6 after reception of the checksum field and to set the SBF reception mode again.

Figure 15-3 shows the port configuration for LIN reception operation.

The wakeup signal transmitted from the LIN master is received by detecting the edge of the external interrupt (INTP0). The length of the sync field transmitted from the LIN master can be measured using the external event capture operation of 16-bit timer/event counter 00, and the baud rate error can be calculated.

The input source of the reception port input (RxD6) can be input to the external interrupt (INTP0) and 16-bit timer/event counter 00 by port input switch control (ISC0/ISC1), without connecting RxD6 and INTP0/TI000 externally.

Figure 15-3. Port Configuration for LIN Reception Operation



Remark ISC0, ISC1, ISC2, ISC4, ISC5: Bits 0, 1, 2, 4 and 5 of the input switch control register (ISC) (see **Figure 15-11**)

The peripheral functions used in the LIN communication operation are shown below.

<Peripheral functions used>

- External interrupt (INTP0); wakeup signal detection
  Use: Detects the wakeup signal edges and detects start of communication.
- 16-bit timer/event counter 00 (TI000); baud rate error detection
  Use: Detects the baud rate error (measures the TI000 input edge interval in the capture mode) by detecting the sync field (SF) length and divides it by the number of bits.
- Serial interface UART6

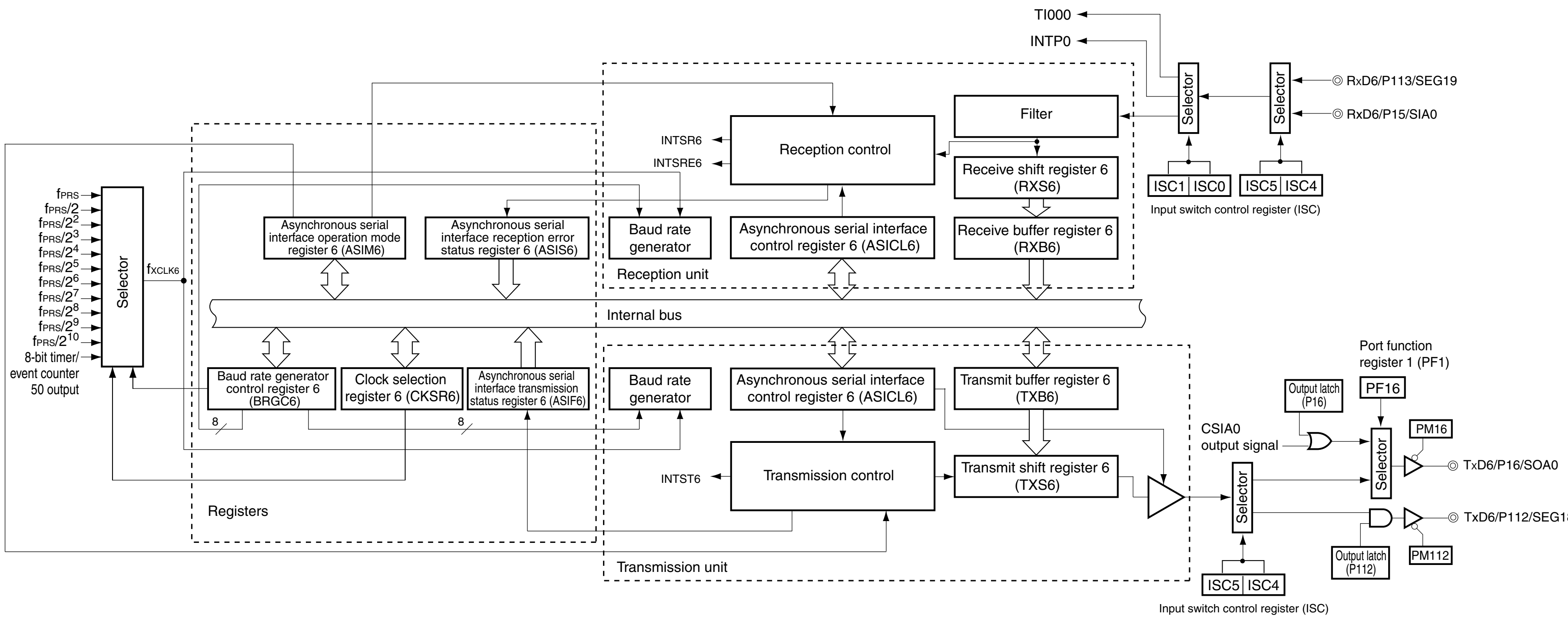
15.2 Configuration of Serial Interface UART6

Serial interface UART6 includes the following hardware.

Table 15-1. Configuration of Serial Interface UART6

Item	Configuration
Registers	Receive buffer register 6 (RXB6) Receive shift register 6 (RXS6) Transmit buffer register 6 (TXB6) Transmit shift register 6 (TXS6)
Control registers	Asynchronous serial interface operation mode register 6 (ASIM6) Asynchronous serial interface reception error status register 6 (ASIS6) Asynchronous serial interface transmission status register 6 (ASIF6) Clock selection register 6 (CKSR6) Baud rate generator control register 6 (BRGC6) Asynchronous serial interface control register 6 (ASICL6) Input switch control register (ISC) Port function register 1 (PF1) Port mode register 1 (PM1) Port register 1 (P1) Port mode register 11 (PM11) Port register 11 (P11)

Figure 15-4. Block Diagram of Serial Interface UART6



(1) Receive buffer register 6 (RXB6)

This 8-bit register stores parallel data converted by receive shift register 6 (RXS6).

Each time 1 byte of data has been received, new receive data is transferred to this register from RXS6. If the data length is set to 7 bits, data is transferred as follows.

- In LSB-first reception, the receive data is transferred to bits 0 to 6 of RXB6 and the MSB of RXB6 is always 0.
- In MSB-first reception, the receive data is transferred to bits 1 to 7 of RXB6 and the LSB of RXB6 is always 0.

If an overrun error (OVE6) occurs, the receive data is not transferred to RXB6.

RXB6 can be read by an 8-bit memory manipulation instruction. No data can be written to this register.

Reset signal generation sets this register to FFH.

(2) Receive shift register 6 (RXS6)

This register converts the serial data input to the RxD6 pin into parallel data.

RXS6 cannot be directly manipulated by a program.

(3) Transmit buffer register 6 (TXB6)

This buffer register is used to set transmit data. Transmission is started when data is written to TXB6.

This register can be read or written by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Cautions 1. Do not write data to TXB6 when bit 1 (TXBF6) of asynchronous serial interface transmission status register 6 (ASIF6) is 1.

2. Do not refresh (write the same value to) TXB6 by software during a communication operation (when bits 7 and 6 (POWER6, TXE6) of asynchronous serial interface operation mode register 6 (ASIM6) are 1 or when bits 7 and 5 (POWER6, RXE6) of ASIM6 are 1).

3. Set transmit data to TXB6 at least one base clock (f_{XCLK6}) after setting TXE6 = 1.

(4) Transmit shift register 6 (TXS6)

This register transmits the data transferred from TXB6 from the TxD6 pin as serial data. Data is transferred from TXB6 immediately after TXB6 is written for the first transmission, or immediately before INTST6 occurs after one frame was transmitted for continuous transmission. Data is transferred from TXB6 and transmitted from the TxD6 pin at the falling edge of the base clock.

TXS6 cannot be directly manipulated by a program.

15.3 Registers Controlling Serial Interface UART6

Serial interface UART6 is controlled by the following twelve registers.

- Asynchronous serial interface operation mode register 6 (ASIM6)
- Asynchronous serial interface reception error status register 6 (ASIS6)
- Asynchronous serial interface transmission status register 6 (ASIF6)
- Clock selection register 6 (CKSR6)
- Baud rate generator control register 6 (BRGC6)
- Asynchronous serial interface control register 6 (ASICL6)
- Input switch control register (ISC)
- Port function register 1 (PF1)
- Port mode register 1 (PM1)
- Port register 1 (P1)
- Port mode register 11 (PM11)
- Port register 11 (P11)

(1) Asynchronous serial interface operation mode register 6 (ASIM6)

This 8-bit register controls the serial communication operations of serial interface UART6.
This register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to 01H.

Remark ASIM6 can be refreshed (the same value is written) by software during a communication operation (when bits 7 and 6 (POWER6, TXE6) of ASIM6 = 1 or bits 7 and 5 (POWER6, RXE6) of ASIM6 = 1).

Figure 15-5. Format of Asynchronous Serial Interface Operation Mode Register 6 (ASIM6) (1/2)

Address: FF50H After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM6	POWER6	TXE6	RXE6	PS61	PS60	CL6	SL6	ISRM6

POWER6	Enables/disables operation of internal operation clock
0[Note 1]	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit[Note 2].
1	Enables operation of the internal operation clock

TXE6	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).
1	Enables transmission

RXE6	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).
1	Enables reception

Notes 1. The output of the TxD6 pin goes high level and the input from the RxD6 pin is fixed to the high level when POWER6 = 0 during transmission.

2. Asynchronous serial interface reception error status register 6 (ASIS6), asynchronous serial interface transmission status register 6 (ASIF6), bit 7 (SBRF6) and bit 6 (SBRT6) of asynchronous serial interface control register 6 (ASICL6), and receive buffer register 6 (RXB6) are reset.

Figure 15-5. Format of Asynchronous Serial Interface Operation Mode Register 6 (ASIM6) (2/2)

PS61	PS60	Transmission operation	Reception operation
0	0	Does not output parity bit.	Reception without parity
0	1	Outputs 0 parity.	Reception as 0 parity[Note]
1	0	Outputs odd parity.	Judges as odd parity.
1	1	Outputs even parity.	Judges as even parity.

CL6	Specifies character length of transmit/receive data
0	Character length of data = 7 bits
1	Character length of data = 8 bits

SL6	Specifies number of stop bits of transmit data
0	Number of stop bits = 1
1	Number of stop bits = 2

ISRM6	Enables/disables occurrence of reception completion interrupt in case of error
0	“INTSRE6” occurs in case of error (at this time, INTSR6 does not occur).
1	“INTSR6” occurs in case of error (at this time, INTSRE6 does not occur).

Note If “reception as 0 parity” is selected, the parity is not judged. Therefore, bit 2 (PE6) of asynchronous serial interface reception error status register 6 (ASIS6) is not set and the error interrupt does not occur.

Cautions
1. **To start the transmission, set POWER6 to 1 and then set TXE6 to 1. To stop the transmission, clear TXE6 to 0, and then clear POWER6 to 0.**
2. **To start the reception, set POWER6 to 1 and then set RXE6 to 1. To stop the reception, clear RXE6 to 0, and then clear POWER6 to 0.**
3. **Set POWER6 to 1 and then set RXE6 to 1 while a high level is input to the RxD6 pin. If POWER6 is set to 1 and RXE6 is set to 1 while a low level is input, reception is started.**
4. **TXE6 and RXE6 are synchronized by the base clock (f_{XCLK6}) set by CKSR6. To enable transmission or reception again, set TXE6 or RXE6 to 1 at least two clocks of the base clock after TXE6 or RXE6 has been cleared to 0. If TXE6 or RXE6 is set within two clocks of the base clock, the transmission circuit or reception circuit may not be initialized.**
5. **Set transmit data to TXB6 at least one base clock (f_{XCLK6}) after setting TXE6 = 1.**
6. **Clear the TXE6 and RXE6 bits to 0 before rewriting the PS61, PS60, and CL6 bits.**
7. **Fix the PS61 and PS60 bits to 0 when used in LIN communication operation.**
8. **Clear TXE6 to 0 before rewriting the SL6 bit. Reception is always performed with “the number of stop bits = 1”, and therefore, is not affected by the set value of the SL6 bit.**
9. **Make sure that RXE6 = 0 when rewriting the ISRM6 bit.**

(2) Asynchronous serial interface reception error status register 6 (ASIS6)

This register indicates an error status on completion of reception by serial interface UART6. It includes three error flag bits (PE6, FE6, OVE6).

This register is read-only by an 8-bit memory manipulation instruction.

Reset signal generation, or clearing bit 7 (POWER6) or bit 5 (RXE6) of ASIM6 to 0 clears this register to 00H. 00H is read when this register is read. If a reception error occurs, read ASIS6 and then read receive buffer register 6 (RXB6) to clear the error flag.

Figure 15-6. Format of Asynchronous Serial Interface Reception Error Status Register 6 (ASIS6)

Address: FF53H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS6	0	0	0	0	0	PE6	FE6	OVE6

PE6	Status flag indicating parity error
0	If POWER6 = 0 or RXE6 = 0, or if ASIS6 register is read
1	If the parity of transmit data does not match the parity bit on completion of reception

FE6	Status flag indicating framing error
0	If POWER6 = 0 or RXE6 = 0, or if ASIS6 register is read
1	If the stop bit is not detected on completion of reception

OVE6	Status flag indicating overrun error
0	If POWER6 = 0 or RXE6 = 0, or if ASIS6 register is read
1	If receive data is set to the RXB6 register and the next reception operation is completed before the data is read.

Cautions 1. The operation of the PE6 bit differs depending on the set values of the PS61 and PS60 bits of asynchronous serial interface operation mode register 6 (ASIM6).

2. For the stop bit of the receive data, only the first stop bit is checked regardless of the number of stop bits.

3. If an overrun error occurs, the next receive data is not written to receive buffer register 6 (RXB6) but discarded.

4. If data is read from ASIS6, a wait cycle is generated. Do not read data from ASIS6 when the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped. For details, see CHAPTER 34 CAUTIONS FOR WAIT.

(3) Asynchronous serial interface transmission status register 6 (ASIF6)

This register indicates the status of transmission by serial interface UART6. It includes two status flag bits (TXBF6 and TXSF6).

Transmission can be continued without disruption even during an interrupt period, by writing the next data to the TXB6 register after data has been transferred from the TXB6 register to the TXS6 register.

This register is read-only by an 8-bit memory manipulation instruction.

Reset signal generation, or clearing bit 7 (POWER6) or bit 6 (TXE6) of ASIM6 to 0 clears this register to 00H.

Figure 15-7. Format of Asynchronous Serial Interface Transmission Status Register 6 (ASIF6)

Address: FF55H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIF6	0	0	0	0	0	0	TXBF6	TXSF6

TXBF6	Transmit buffer data flag
0	If POWER6 = 0 or TXE6 = 0, or if data is transferred to transmit shift register 6 (TXS6)
1	If data is written to transmit buffer register 6 (TXB6) (if data exists in TXB6)

TXSF6	Transmit shift register data flag
0	If POWER6 = 0 or TXE6 = 0, or if the next data is not transferred from transmit buffer register 6 (TXB6) after completion of transfer
1	If data is transferred from transmit buffer register 6 (TXB6) (if data transmission is in progress)

Cautions 1. To transmit data continuously, write the first transmit data (first byte) to the TXB6 register. Be sure to check that the TXBF6 flag is "0". If so, write the next transmit data (second byte) to the TXB6 register. If data is written to the TXB6 register while the TXBF6 flag is "1", the transmit data cannot be guaranteed.

2. To initialize the transmission unit upon completion of continuous transmission, be sure to check that the TXSF6 flag is "0" after generation of the transmission completion interrupt, and then execute initialization. If initialization is executed while the TXSF6 flag is "1", the transmit data cannot be guaranteed.

(4) Clock selection register 6 (CKSR6)

This register selects the base clock of serial interface UART6.

CKSR6 can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Remark CKSR6 can be refreshed (the same value is written) by software during a communication operation (when bits 7 and 6 (POWER6, TXE6) of ASIM6 = 1 or bits 7 and 5 (POWER6, RXE6) of ASIM6 = 1).

Figure 15-8. Format of Clock Selection Register 6 (CKSR6)

Address: FF56H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CKSR6	0	0	0	0	TPS63	TPS62	TPS61	TPS60

TPS63	TPS62	TPS61	TPS60	Base clock (f_{XCLK6}) selection[Note 1]				
					f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 8 MHz	f_{PRS} = 10 MHz
0	0	0	0	f_{PRS}[Note 2]	2 MHz	5 MHz	8 MHz	10 MHz
0	0	0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	4 MHz	5 MHz
0	0	1	0	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2 MHz	2.5 MHz
0	0	1	1	$f_{PRS}/2^3$	250 kHz	625 kHz	1 MHz	1.25 MHz
0	1	0	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	500 kHz	625 kHz
0	1	0	1	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	250 kHz	312.5 kHz
0	1	1	0	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	125 kHz	156.25 kHz
0	1	1	1	$f_{PRS}/2^7$	15.625 kHz	39.06 kHz	62.5 kHz	78.13 kHz
1	0	0	0	$f_{PRS}/2^8$	7.813 kHz	19.53 kHz	31.25 kHz	39.06 kHz
1	0	0	1	$f_{PRS}/2^9$	3.906 kHz	9.77 kHz	15.625 kHz	19.53 kHz
1	0	1	0	$f_{PRS}/2^{10}$	1.953 kHz	4.88 kHz	7.513 kHz	9.77 kHz
1	0	1	1	TM50 output[Note 3]				
Other than above				Setting prohibited				

Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{XH}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 2.7 to 5.5 V: $f_{PRS} \leq 10$ MHz
- V_{DD} = 1.8 to 2.7 V: $f_{PRS} \leq 5$ MHz

2. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{RH}) (XSEL = 0), when 1.8 V ≤ V_{DD} < 2.7 V, the setting of TPS63 = TPS62 = TPS61 = TPS60 = 0 (base clock: f_{PRS}) is prohibited.

3. Note the following points when selecting the TM50 output as the base clock.

- Mode in which the count clock is cleared and started upon a match of TM50 and CR50 (TMC506 = 0)
  Start the operation of 8-bit timer/event counter 50 first and then enable the timer F/F inversion operation (TMC501 = 1).
- PWM mode (TMC506 = 1)
  Start the operation of 8-bit timer/event counter 50 first and then set the count clock to make the duty = 50%.

It is not necessary to enable (TOE50 = 1) TO50 output in any mode.

Caution Make sure POWER6 = 0 when rewriting TPS63 to TPS60.

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. TMC506: Bit 6 of 8-bit timer mode control register 50 (TMC50)
TMC501: Bit 1 of TMC50

(5) Baud rate generator control register 6 (BRGC6)

This register sets the division value of the 8-bit counter of serial interface UART6.
BRGC6 can be set by an 8-bit memory manipulation instruction.
Reset signal generation sets this register to FFH.

Remark BRGC6 can be refreshed (the same value is written) by software during a communication operation (when bits 7 and 6 (POWER6, TXE6) of ASIM6 = 1 or bits 7 and 5 (POWER6, RXE6) of ASIM6 = 1).

Figure 15-9. Format of Baud Rate Generator Control Register 6 (BRGC6)

Address: FF57H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
BRGC6	MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60

MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60	k	Output clock selection of 8-bit counter
0	0	0	0	0	0	×	×	×	Setting prohibited
0	0	0	0	0	1	0	0	4	$f_{XCLK6}/4$
0	0	0	0	0	1	0	1	5	$f_{XCLK6}/5$
0	0	0	0	0	1	1	0	6	$f_{XCLK6}/6$
• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •
1	1	1	1	1	1	0	0	252	$f_{XCLK6}/252$
1	1	1	1	1	1	0	1	253	$f_{XCLK6}/253$
1	1	1	1	1	1	1	0	254	$f_{XCLK6}/254$
1	1	1	1	1	1	1	1	255	$f_{XCLK6}/255$

Cautions 1. Make sure that bit 6 (TXE6) and bit 5 (RXE6) of the ASIM6 register = 0 when rewriting the MDL67 to MDL60 bits.

2. The baud rate is the output clock of the 8-bit counter divided by 2.

Remarks 1. f_{XCLK6}: Frequency of base clock selected by the TPS63 to TPS60 bits of CKSR6 register

2. k: Value set by MDL67 to MDL60 bits (k = 4, 5, 6, ..., 255)

3. ×: Don't care

(6) Asynchronous serial interface control register 6 (ASICL6)

This register controls the serial communication operations of serial interface UART6.

ASICL6 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 16H.

Caution ASICL6 can be refreshed (the same value is written) by software during a communication operation (when bits 7 and 6 (POWER6, TXE6) of ASIM6 = 1 or bits 7 and 5 (POWER6, RXE6) of ASIM6 = 1). However, do not set both SBRT6 and SBTT6 to 1 by a refresh operation during SBF reception (SBRT6 = 1) or SBF transmission (until INTST6 occurs since SBTT6 has been set (1)), because it may re-trigger SBF reception or SBF transmission.

Figure 15-10. Format of Asynchronous Serial Interface Control Register 6 (ASICL6) (1/2)

Address: FF58H After reset: 16H R/W[Note]

Symbol	<7>	<6>	5	4	3	2	1	0
ASICL6	SBRF6	SBRT6	SBTT6	SBL62	SBL61	SBL60	DIR6	TXDLV6

SBRF6	SBF reception status flag
0	If POWER6 = 0 and RXE6 = 0 or if SBF reception has been completed correctly
1	SBF reception in progress

SBRT6	SBF reception trigger
0	–
1	SBF reception trigger

SBTT6	SBF transmission trigger
0	–
1	SBF transmission trigger

Note Bit 7 is read-only.

Figure 15-10. Format of Asynchronous Serial Interface Control Register 6 (ASICL6) (2/2)

SBL62	SBL61	SBL60	SBF transmission output width control
1	0	1	SBF is output with 13-bit length.
1	1	0	SBF is output with 14-bit length.
1	1	1	SBF is output with 15-bit length.
0	0	0	SBF is output with 16-bit length.
0	0	1	SBF is output with 17-bit length.
0	1	0	SBF is output with 18-bit length.
0	1	1	SBF is output with 19-bit length.
1	0	0	SBF is output with 20-bit length.

DIR6	First-bit specification
0	MSB
1	LSB

TXDLV6	Enables/disables inverting TxD6 output
0	Normal output of TxD6
1	Inverted output of TxD6

Cautions 1. In the case of an SBF reception error, the mode returns to the SBF reception mode. The status of the SBRF6 flag is held (1).

2. Before setting the SBRT6 bit, make sure that bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1. After setting the SBRT6 bit to 1, do not clear it to 0 before SBF reception is completed (before an interrupt request signal is generated).

3. The read value of the SBRT6 bit is always 0. SBRT6 is automatically cleared to 0 after SBF reception has been correctly completed.

4. Before setting the SBTT6 bit to 1, make sure that bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1. After setting the SBTT6 bit to 1, do not clear it to 0 before SBF transmission is completed (before an interrupt request signal is generated).

5. The read value of the SBTT6 bit is always 0. SBTT6 is automatically cleared to 0 at the end of SBF transmission.

6. Do not set the SBRT6 bit to 1 during reception, and do not set the SBTT6 bit to 1 during transmission.

7. Before rewriting the DIR6 and TXDLV6 bits, clear the TXE6 and RXE6 bits to 0.

(7) Input switch control register (ISC)

By setting ISC4 to 1, the UART6 I/O pins are switched from P113/SEG19/RxD6 and P112/SEG18/TxD6 to P15/SIA0/<RxD6> and P16/SOA0/<TxD6>.

By setting ISC3 to 1, the P113/SEG19/RxD6 pin is enabled for input. When ISC3 is cleared to 0, external input is not acknowledged. Thus, after release of reset, a generation of a through current due to an undetermined input state until an output setting is performed is prevented.

The input switch control register (ISC) is used to receive a status signal transmitted from the master during LIN (Local Interconnect Network) reception.

By setting ISC0 and ISC1 to 1, the input sources of INTP0 and TI000 are switched to input signals from the P15/SIA0/RxD6 or P113/SEG19/RxD6 pin.

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

Reset signal generation sets this register to 00H.

Figure 15-11. Format of Input Switch Control Register (ISC)

Address: FF4FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	ISC5	ISC4	ISC3	ISC2	ISC1	ISC0

ISC5	ISC4	TxD6, RxD6 input source selection
0	0	TxD6:P112, RxD6: P113
0	1	TxD6:P16, RxD6: P15
Other than above		Setting prohibited

ISC3	RxD6/P113 input enabled/disabled
0	RxD6/P113 input disabled
1	RxD6/P113 input enabled

ISC2	TI52 input source control
0	No enable control of TI52 input (P34)
1	Enable controlled of TI52 input (P34) Note

ISC1	TI000 input source selection
0	TI000 (P33)
1	RxD6 (P15 or P113)

ISC0	INTP0 input source selection
0	INTP0 (P120)
1	RxD6 (P15 or P113)

Note TI52 input is controlled by TOH2 output signal.

Caution When using the P113/RxD6/SEG19 pin as the P113 or RxD6 pin, set PF11ALL to 0 and ISC3 to 1, after release of reset.
When using the P113/RxD6/SEG19 pin as the SEG19 pin, set PF11ALL to 1 and ISC3 to 0, after release of reset.

(8) Port function register 1 (PF1)

This register sets the pin functions of P16/SOA0/TxD6 pin.

PF1 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PF1 to 00H.

Figure 15-12. Format of Port Function Register 1 (PF1)

Address: FF20H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PF1	0	PF16	0	0	PF13	0	0	0

PF16	Port (P16), CSIA0, and UART6 output specification
0	Used as P16 or SOA0
1	Used as TxD6

PF13	Port (P13), CSI10, and UART0 output specification
0	Used as P13 or SO10
1	Used as TxD0

(9) Port mode register 1 (PM1)

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

When using the P16/SOA0/TxD6 pin for serial interface data output, clear PM16 to 0. The output latch of P16 at this time may be 0 or 1.

When using the P15/SIA0/RxD6 pin for serial interface data input, set PM15 to 1. The output latch of P15 at this time may be 0 or 1.

PM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 15-13. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

(10) Port mode register 11 (PM11)

This register sets port 11 input/output in 1-bit units.

When using the P112/SEG18/TxD6 pin for serial interface data output, clear PM112 to 0 and set the output latch of P112 to 1.

When using the P113/SEG19/RxD6 pin for serial interface data input, set PM113 to 1. The output latch of P113 at this time may be 0 or 1.

PM11 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 15-14. Format of Port Mode Register 11 (PM11)

Address: FF2BH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM11	1	1	1	1	PM113	PM112	PM111	PM110

PM11n	P11n pin I/O mode selection (n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

15.4 Operation of Serial Interface UART6

Serial interface UART6 has the following two modes.

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

15.4.1 Operation stop mode

In this mode, serial communication cannot be executed; therefore, the power consumption can be reduced. In addition, the pins can be used as ordinary port pins in this mode. To set the operation stop mode, clear bits 7, 6, and 5 (POWER6, TXE6, and RXE6) of ASIM6 to 0.

(1) Register used

The operation stop mode is set by asynchronous serial interface operation mode register 6 (ASIM6).
ASIM6 can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to 01H.

Address: FF50H After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM6	POWER6	TXE6	RXE6	PS61	PS60	CL6	SL6	ISRM6

POWER6	Enables/disables operation of internal operation clock
0[Note 1]	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit[Note 2].

TXE6	Enables/disables transmission
0	Disables transmission operation (synchronously resets the transmission circuit).

RXE6	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).

Notes 1. The output of the TxD6 pin goes high and the input from the RxD6 pin is fixed to high level when POWER6 = 0 during transmission.

2. Asynchronous serial interface reception error status register 6 (ASIS6), asynchronous serial interface transmission status register 6 (ASIF6), bit 7 (SBRF6) and bit 6 (SBRT6) of asynchronous serial interface control register 6 (ASICL6), and receive buffer register 6 (RXB6) are reset.

Caution Clear POWER6 to 0 after clearing TXE6 and RXE6 to 0 to stop the operation.
To start the communication, set POWER6 to 1, and then set TXE6 or RXE6 to 1.

Remark To use the RxD6/P15 and TxD6/P16 or RxD6/P113 and TxD6/P112 pins as general-purpose port pins, see **CHAPTER 4 PORT FUNCTIONS**.

15.4.2 Asynchronous serial interface (UART) mode

In this mode, data of 1 byte is transmitted/received following a start bit, and a full-duplex operation can be performed.

A dedicated UART baud rate generator is incorporated, so that communication can be executed at a wide range of baud rates.

(1) Registers used

- Asynchronous serial interface operation mode register 6 (ASIM6)
- Asynchronous serial interface reception error status register 6 (ASIS6)
- Asynchronous serial interface transmission status register 6 (ASIF6)
- Clock selection register 6 (CKSR6)
- Baud rate generator control register 6 (BRGC6)
- Asynchronous serial interface control register 6 (ASICL6)
- Input switch control register (ISC)
- Port mode register 1 (PM1)
- Port register 1 (P1)
- Port mode register 11 (PM11)
- Port register 11 (P11)

The basic procedure of setting an operation in the UART mode is as follows.

<1> Set the CKSR6 register (see **Figure 15-8**).
<2> Set the BRGC6 register (see **Figure 15-9**).
<3> Set bits 0 to 4 (ISRM6, SL6, CL6, PS60, PS61) of the ASIM6 register (see **Figure 15-5**).
<4> Set bits 0 and 1 (TXDLV6, DIR6) of the ASICL6 register (see **Figure 15-10**).
<5> Set bit 7 (POWER6) of the ASIM6 register to 1.
<6> Set bit 6 (TXE6) of the ASIM6 register to 1. → Transmission is enabled.
Set bit 5 (RXE6) of the ASIM6 register to 1. → Reception is enabled.
<7> Write data to transmit buffer register 6 (TXB6). → Data transmission is started.

Caution Take relationship with the other party of communication when setting the port mode register and port register.

The relationship between the register settings and pins is shown below.

Table 15-2. Relationship Between Register Settings and Pins

(a) When the P16 and P15 are selected as the UART6 pins using the bits 4, 5 (ISC4, ISC5) of the ISC register

POWER6	TXE6	RXE6	PM16	P16	PM15	P15	UART6 Operation	Pin Function	
								TxD6/SOA0/P16	RxD6/SIA0/P15
0	0	0	×Note	×Note	×Note	×Note	Stop	SOA0/P16	SIA0/P15
1	0	1	×Note	×Note	1	×	Reception	SOA0/P16	RxD6
	1	0	0	×	×Note	×Note	Transmission	TxD6	SIA0/P15
	1	1	0	×	1	×	Transmission/ reception	TxD6	RxD6

Note Can be set as port function or serial interface CSIA0.

Caution TxD6/SEG18/P112 and RxD6/SEG19/P113 pins function as the SEG18/P112 and SEG19/P113.

Remark ×: don't care
POWER6: Bit 7 of asynchronous serial interface operation mode register 6 (ASIM6)
TXE6: Bit 6 of ASIM6
RXE6: Bit 5 of ASIM6
PM1×: Port mode register
P1×: Port output latch

(b) When the P112 and P113 are selected as the UART6 pins using the bits 4, 5 (ISC4, ISC5) of the ISC register

POWER6	TXE6	RXE6	PM112	P112	PM113	P113	UART6 Operation	Pin Function	
								TxD6/SEG18/P112	RxD6/SEG19/P113
0	0	0	×Note	×Note	×Note	×Note	Stop	SEG18/P112	SEG19/P113
1	0	1	×Note	×Note	1	×	Reception	SEG18/P112	RxD6
	1	0	0	1	×Note	×Note	Transmission	TxD6	SEG19/P113
	1	1	0	1	1	×	Transmission/ reception	TxD6	RxD6

Note Can be set as port function or segment output.

Caution TxD6/SOA0/P16 and RxD6/SIA0/P15 pins function as the SOA0/P16 and SIA0/P15.

Remark ×: don't care
POWER6: Bit 7 of asynchronous serial interface operation mode register 6 (ASIM6)
TXE6: Bit 6 of ASIM6
RXE6: Bit 5 of ASIM6
PM11×: Port mode register
P11×: Port output latch

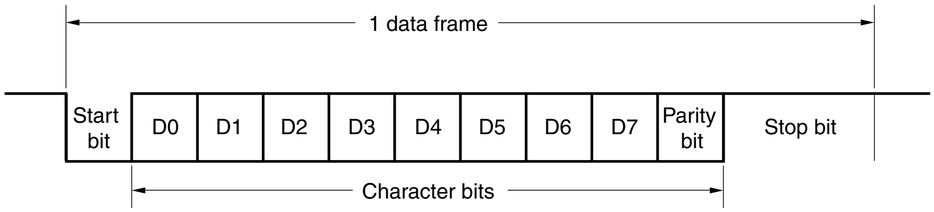
(2) Communication operation

(a) Format and waveform example of normal transmit/receive data

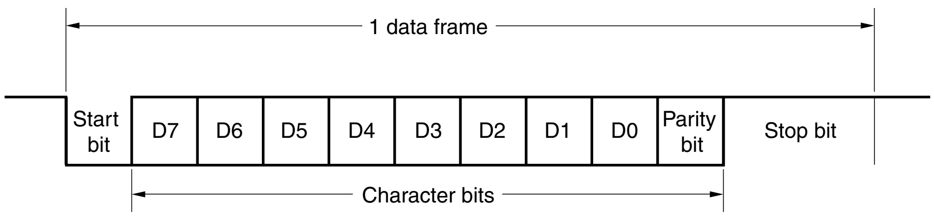
Figures 15-15 and 15-16 show the format and waveform example of the normal transmit/receive data.

Figure 15-15. Format of Normal UART Transmit/Receive Data

1. LSB-first transmission/reception



2. MSB-first transmission/reception



One data frame consists of the following bits.

- Start bit ... 1 bit
- Character bits ... 7 or 8 bits
- Parity bit ... Even parity, odd parity, 0 parity, or no parity
- Stop bit ... 1 or 2 bits

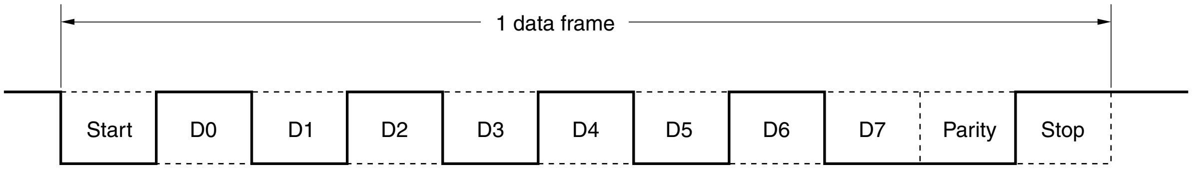
The character bit length, parity, and stop bit length in one data frame are specified by asynchronous serial interface operation mode register 6 (ASIM6).

Whether data is communicated with the LSB or MSB first is specified by bit 1 (DIR6) of asynchronous serial interface control register 6 (ASICL6).

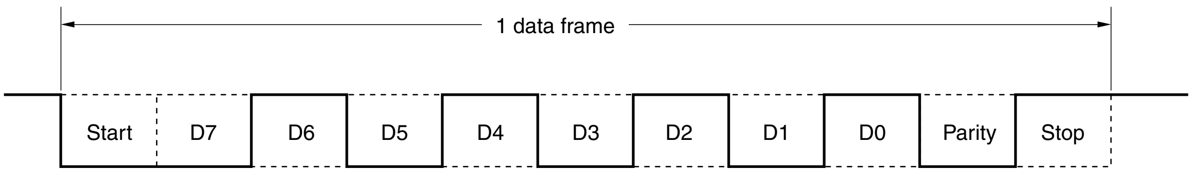
Whether the TxD6 pin outputs normal or inverted data is specified by bit 0 (TXDLV6) of ASICL6.

Figure 15-16. Example of Normal UART Transmit/Receive Data Waveform

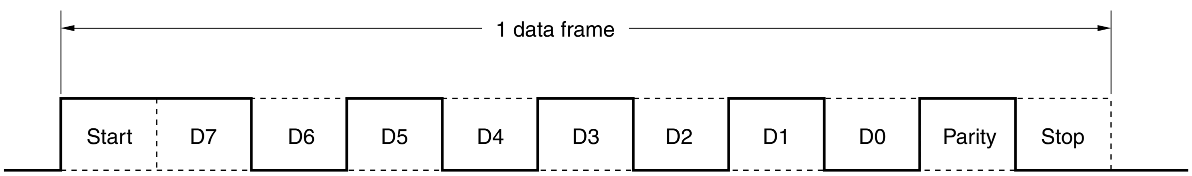
1. **Data length: 8 bits, LSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H**



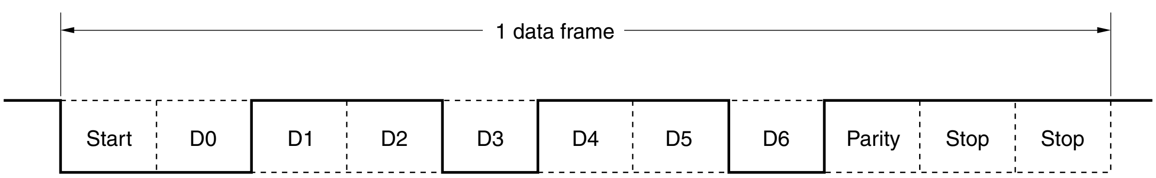
2. **Data length: 8 bits, MSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H**



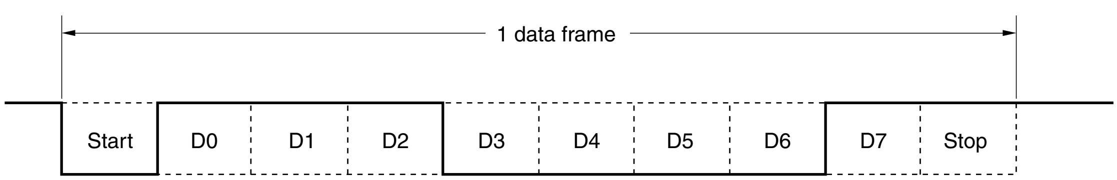
3. **Data length: 8 bits, MSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H, TxD6 pin inverted output**



4. **Data length: 7 bits, LSB first, Parity: Odd parity, Stop bit: 2 bits, Communication data: 36H**



5. **Data length: 8 bits, LSB first, Parity: None, Stop bit: 1 bit, Communication data: 87H**



(b) Parity types and operation

The parity bit is used to detect a bit error in communication data. Usually, the same type of parity bit is used on both the transmission and reception sides. With even parity and odd parity, a 1-bit (odd number) error can be detected. With zero parity and no parity, an error cannot be detected.

Caution Fix the PS61 and PS60 bits to 0 when the device is used in LIN communication operation.

(i) Even parity

- Transmission

  Transmit data, including the parity bit, is controlled so that the number of bits that are “1” is even.
  The value of the parity bit is as follows.

  If transmit data has an odd number of bits that are “1”: 1
  If transmit data has an even number of bits that are “1”: 0

- Reception

  The number of bits that are “1” in the receive data, including the parity bit, is counted. If it is odd, a parity error occurs.

(ii) Odd parity

- Transmission

  Unlike even parity, transmit data, including the parity bit, is controlled so that the number of bits that are “1” is odd.

  If transmit data has an odd number of bits that are “1”: 0
  If transmit data has an even number of bits that are “1”: 1

- Reception

  The number of bits that are “1” in the receive data, including the parity bit, is counted. If it is even, a parity error occurs.

(iii) 0 parity

The parity bit is cleared to 0 when data is transmitted, regardless of the transmit data.
The parity bit is not detected when the data is received. Therefore, a parity error does not occur regardless of whether the parity bit is “0” or “1”.

(iv) No parity

No parity bit is appended to the transmit data.
Reception is performed assuming that there is no parity bit when data is received. Because there is no parity bit, a parity error does not occur.

(c) Normal transmission

When bit 7 (POWER6) of asynchronous serial interface operation mode register 6 (ASIM6) is set to 1 and bit 6 (TXE6) of ASIM6 is then set to 1, transmission is enabled. Transmission can be started by writing transmit data to transmit buffer register 6 (TXB6). The start bit, parity bit, and stop bit are automatically appended to the data.

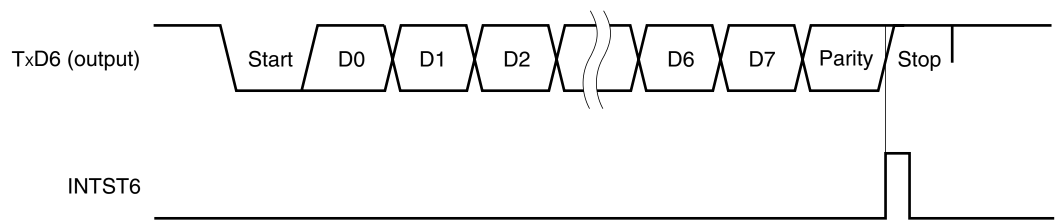
When transmission is started, the data in TXB6 is transferred to transmit shift register 6 (TXS6). After that, the transmit data is sequentially output from TXS6 to the TxD6 pin. When transmission is completed, the parity and stop bits set by ASIM6 are appended and a transmission completion interrupt request (INTST6) is generated.

Transmission is stopped until the data to be transmitted next is written to TXB6.

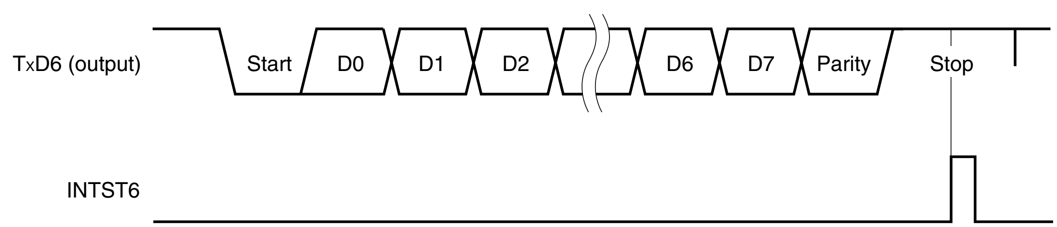
Figure 15-17 shows the timing of the transmission completion interrupt request (INTST6). This interrupt occurs as soon as the last stop bit has been output.

Figure 15-17. Normal Transmission Completion Interrupt Request Timing

1. Stop bit length: 1



2. Stop bit length: 2



(d) Continuous transmission

The next transmit data can be written to transmit buffer register 6 (TXB6) as soon as transmit shift register 6 (TXS6) has started its shift operation. Consequently, even while the INTST6 interrupt is being serviced after transmission of one data frame, data can be continuously transmitted and an efficient communication rate can be realized. In addition, the TXB6 register can be efficiently written twice (2 bytes) without having to wait for the transmission time of one data frame, by reading bit 0 (TXSF6) of asynchronous serial interface transmission status register 6 (ASIF6) when the transmission completion interrupt has occurred.

To transmit data continuously, be sure to reference the ASIF6 register to check the transmission status and whether the TXB6 register can be written, and then write the data.

Cautions 1. The TXBF6 and TXSF6 flags of the ASIF6 register change from "10" to "11", and to "01" during continuous transmission. To check the status, therefore, do not use a combination of the TXBF6 and TXSF6 flags for judgment. Read only the TXBF6 flag when executing continuous transmission.

2. When the device is use in LIN communication operation, the continuous transmission function cannot be used. Make sure that asynchronous serial interface transmission status register 6 (ASIF6) is 00H before writing transmit data to transmit buffer register 6 (TXB6).

TXBF6	Writing to TXB6 Register
0	Writing enabled
1	Writing disabled

Caution To transmit data continuously, write the first transmit data (first byte) to the TXB6 register. Be sure to check that the TXBF6 flag is "0". If so, write the next transmit data (second byte) to the TXB6 register. If data is written to the TXB6 register while the TXBF6 flag is "1", the transmit data cannot be guaranteed.

The communication status can be checked using the TXSF6 flag.

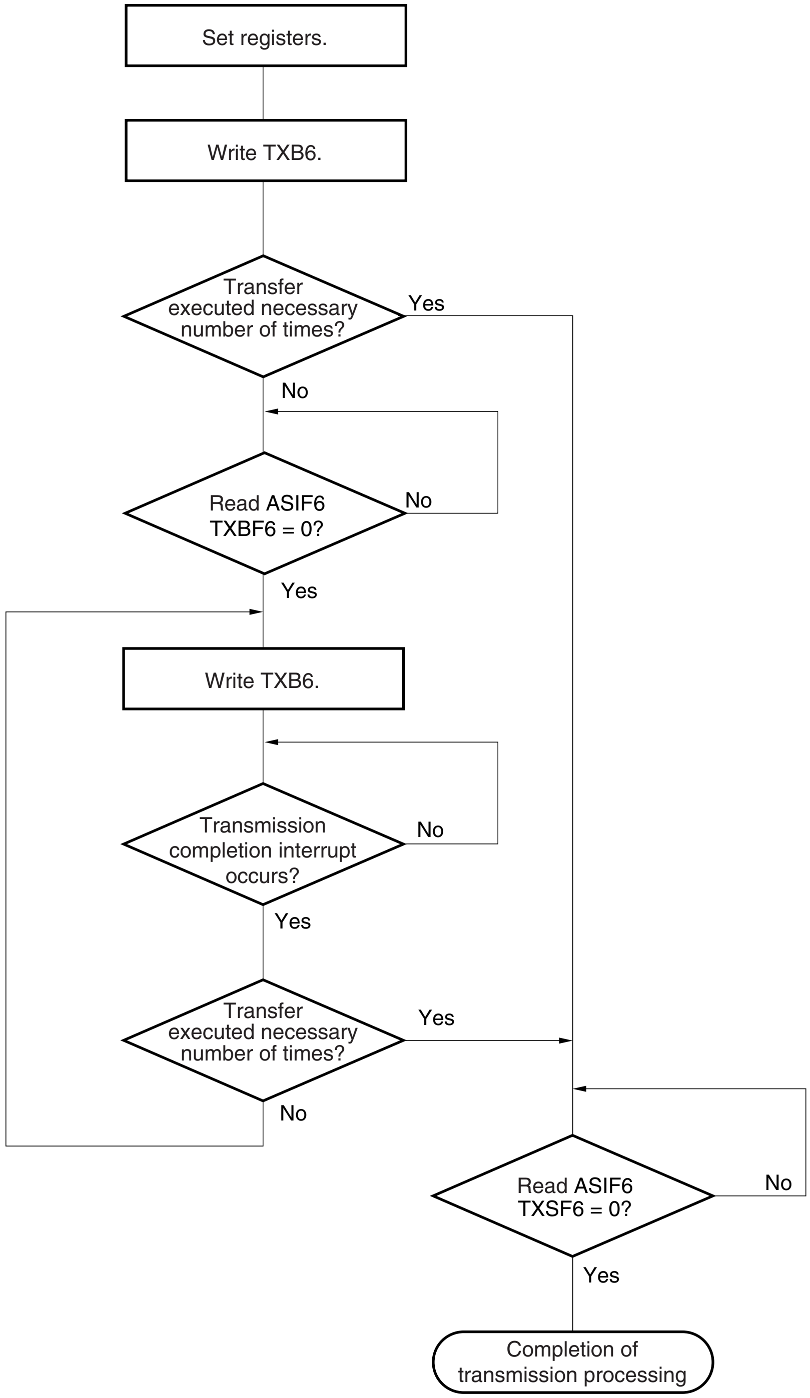
TXSF6	Transmission Status
0	Transmission is completed.
1	Transmission is in progress.

Cautions 1. To initialize the transmission unit upon completion of continuous transmission, be sure to check that the TXSF6 flag is "0" after generation of the transmission completion interrupt, and then execute initialization. If initialization is executed while the TXSF6 flag is "1", the transmit data cannot be guaranteed.

2. During continuous transmission, the next transmission may complete before execution of INTST6 interrupt servicing after transmission of one data frame. As a countermeasure, detection can be performed by developing a program that can count the number of transmit data and by referencing the TXSF6 flag.

Figure 15-18 shows an example of the continuous transmission processing flow.

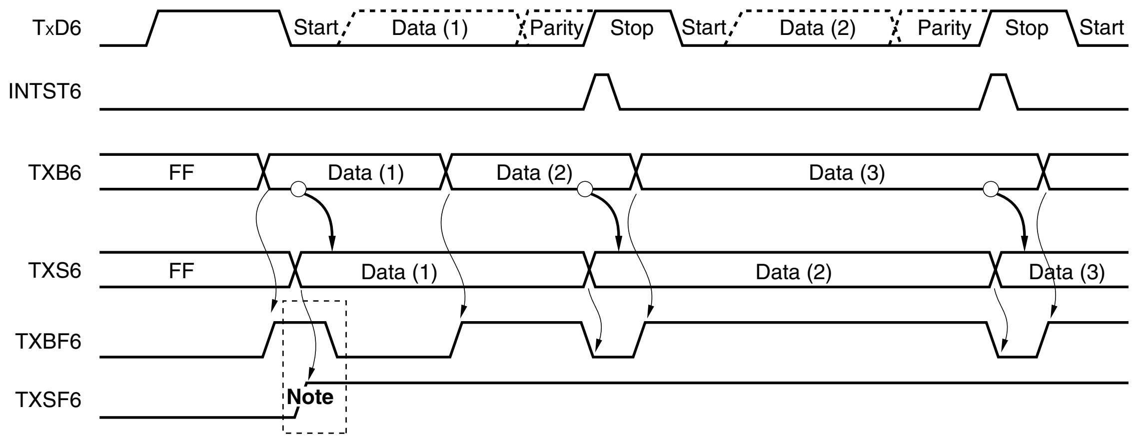
Figure 15-18. Example of Continuous Transmission Processing Flow



Remark TXB6: Transmit buffer register 6
ASIF6: Asynchronous serial interface transmission status register 6
TXBF6: Bit 1 of ASIF6 (transmit buffer data flag)
TXSF6: Bit 0 of ASIF6 (transmit shift register data flag)

Figure 15-19 shows the timing of starting continuous transmission, and Figure 15-20 shows the timing of ending continuous transmission.

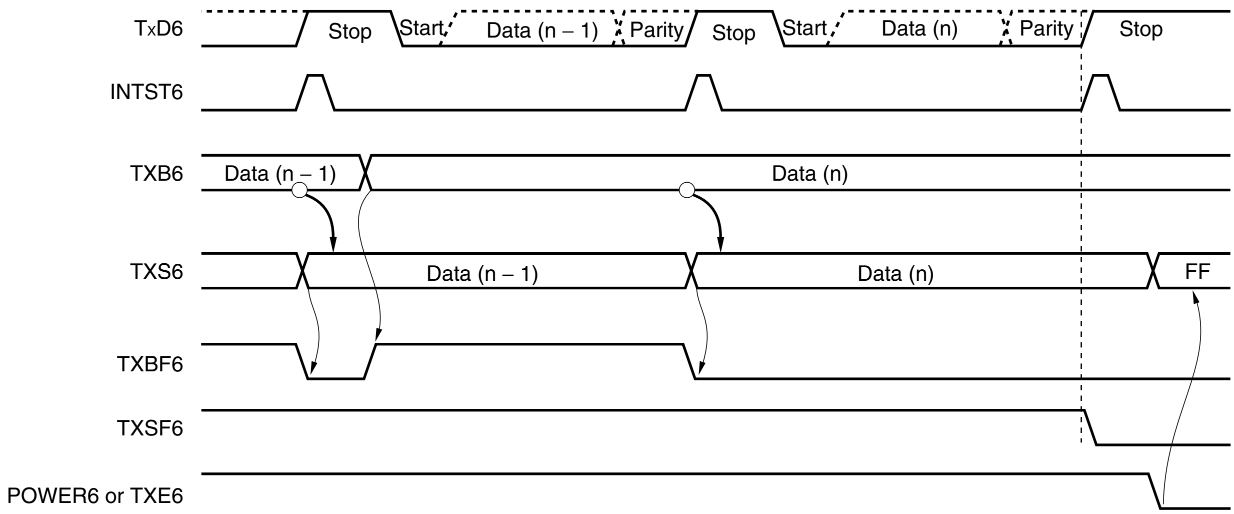
Figure 15-19. Timing of Starting Continuous Transmission



Note When ASIF6 is read, there is a period in which TXBF6 and TXSF6 = 1, 1. Therefore, judge whether writing is enabled using only the TXBF6 bit.

Remark
TxD6: TxD6 pin (output)
INTST6: Interrupt request signal
TXB6: Transmit buffer register 6
TXS6: Transmit shift register 6
ASIF6: Asynchronous serial interface transmission status register 6
TXBF6: Bit 1 of ASIF6
TXSF6: Bit 0 of ASIF6

Figure 15-20. Timing of Ending Continuous Transmission



Remark

TxD6:	TxD6 pin (output)
INTST6:	Interrupt request signal
TXB6:	Transmit buffer register 6
TXS6:	Transmit shift register 6
ASIF6:	Asynchronous serial interface transmission status register 6
TXBF6:	Bit 1 of ASIF6
TXSF6:	Bit 0 of ASIF6
POWER6:	Bit 7 of asynchronous serial interface operation mode register (ASIM6)
TXE6:	Bit 6 of asynchronous serial interface operation mode register (ASIM6)

(e) Normal reception

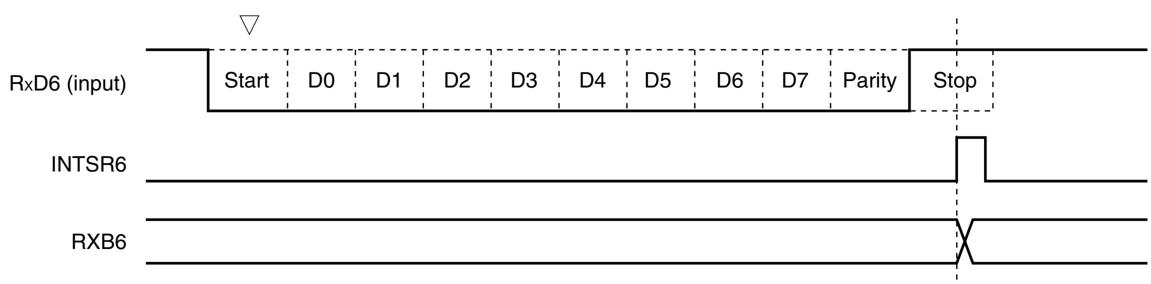
Reception is enabled and the RxD6 pin input is sampled when bit 7 (POWER6) of asynchronous serial interface operation mode register 6 (ASIM6) is set to 1 and then bit 5 (RXE6) of ASIM6 is set to 1.

The 8-bit counter of the baud rate generator starts counting when the falling edge of the RxD6 pin input is detected. When the set value of baud rate generator control register 6 (BRGC6) has been counted, the RxD6 pin input is sampled again (▽ in Figure 15-21). If the RxD6 pin is low level at this time, it is recognized as a start bit.

When the start bit is detected, reception is started, and serial data is sequentially stored in the receive shift register (RXS6) at the set baud rate. When the stop bit has been received, the reception completion interrupt (INTSR6) is generated and the data of RXS6 is written to receive buffer register 6 (RXB6). If an overrun error (OVE6) occurs, however, the receive data is not written to RXB6.

Even if a parity error (PE6) occurs while reception is in progress, reception continues to the reception position of the stop bit, and a reception error interrupt (INTSR6/INTSRE6) is generated on completion of reception.

Figure 15-21. Reception Completion Interrupt Request Timing



Cautions 1. If a reception error occurs, read ASIS6 and then RXB6 to clear the error flag. Otherwise, an overrun error will occur when the next data is received, and the reception error status will persist.

2. Reception is always performed with the “number of stop bits = 1”. The second stop bit is ignored.

3. Be sure to read asynchronous serial interface reception error status register 6 (ASIS6) before reading RXB6.

(f) Reception error

Three types of errors may occur during reception: a parity error, framing error, or overrun error. If the error flag of asynchronous serial interface reception error status register 6 (ASIS6) is set as a result of data reception, a reception error interrupt request (INTSR6/INTSRE6) is generated.

Which error has occurred during reception can be identified by reading the contents of ASIS6 in the reception error interrupt (INTSR6/INTSRE6) servicing (see **Figure 15-6**).

The contents of ASIS6 are cleared to 0 when ASIS6 is read.

Table 15-3. Cause of Reception Error

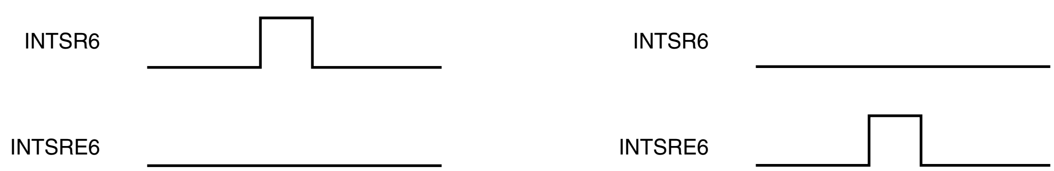
Reception Error	Cause
Parity error	The parity specified for transmission does not match the parity of the receive data.
Framing error	Stop bit is not detected.
Overrun error	Reception of the next data is completed before data is read from receive buffer register 6 (RXB6).

The reception error interrupt can be separated into reception completion interrupt (INTSR6) and error interrupt (INTSRE6) by clearing bit 0 (ISRM6) of asynchronous serial interface operation mode register 6 (ASIM6) to 0.

Figure 15-22. Reception Error Interrupt

1. If ISRM6 is cleared to 0 (reception completion interrupt (INTSR6) and error interrupt (INTSRE6) are separated)

(a) No error during reception **(b) Error during reception**



2. If ISRM6 is set to 1 (error interrupt is included in INTSR6)

(a) No error during reception **(b) Error during reception**

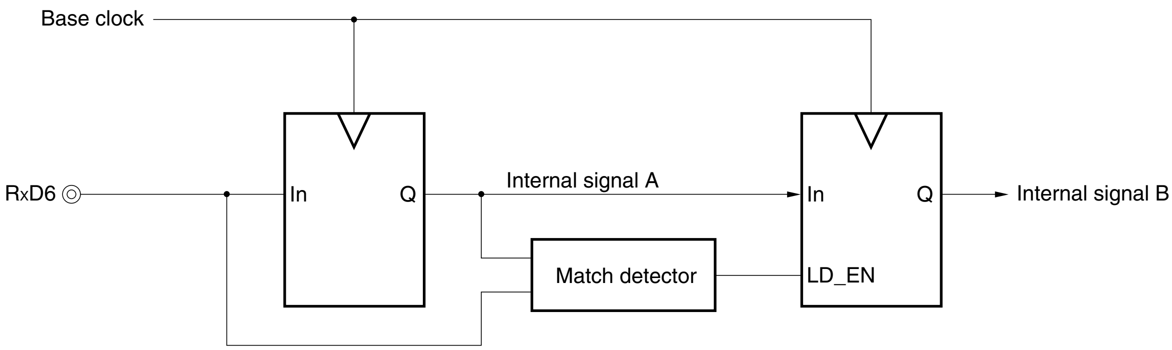
(g) Noise filter of receive data

The RxD6 signal is sampled with the base clock output by the prescaler block.

If two sampled values are the same, the output of the match detector changes, and the data is sampled as input data.

Because the circuit is configured as shown in Figure 15-23, the internal processing of the reception operation is delayed by two clocks from the external signal status.

Figure 15-23. Noise Filter Circuit



(h) SBF transmission

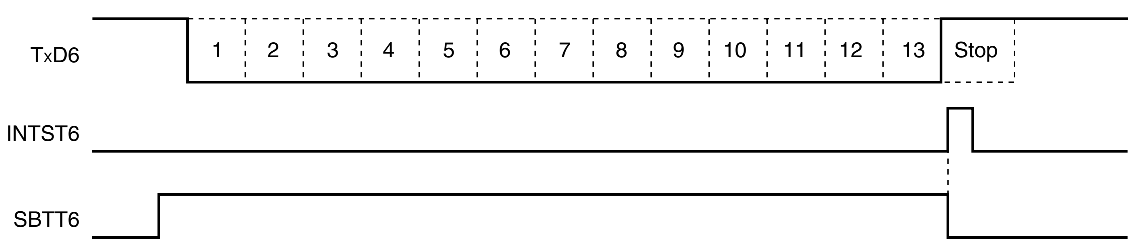
When the device is use in LIN communication operation, the SBF (Synchronous Break Field) transmission control function is used for transmission. For the transmission operation of LIN, see **Figure 15-1 LIN Transmission Operation**.

When bit 7 (POWER6) of asynchronous serial interface mode register 6 (ASIM6) is set to 1, the TxD6 pin outputs high level. Next, when bit 6 (TXE6) of ASIM6 is set to 1, the transmission enabled status is entered, and SBF transmission is started by setting bit 5 (SBTT6) of asynchronous serial interface control register 6 (ASICL6) to 1.

Thereafter, a low level of bits 13 to 20 (set by bits 4 to 2 (SBL62 to SBL60) of ASICL6) is output. Following the end of SBF transmission, the transmission completion interrupt request (INTST6) is generated and SBTT6 is automatically cleared. Thereafter, the normal transmission mode is restored.

Transmission is suspended until the data to be transmitted next is written to transmit buffer register 6 (TXB6), or until SBTT6 is set to 1.

Figure 15-24. SBF Transmission



Remark TxD6: TxD6 pin (output)
INTST6: Transmission completion interrupt request
SBTT6: Bit 5 of asynchronous serial interface control register 6 (ASICL6)

(i) SBF reception

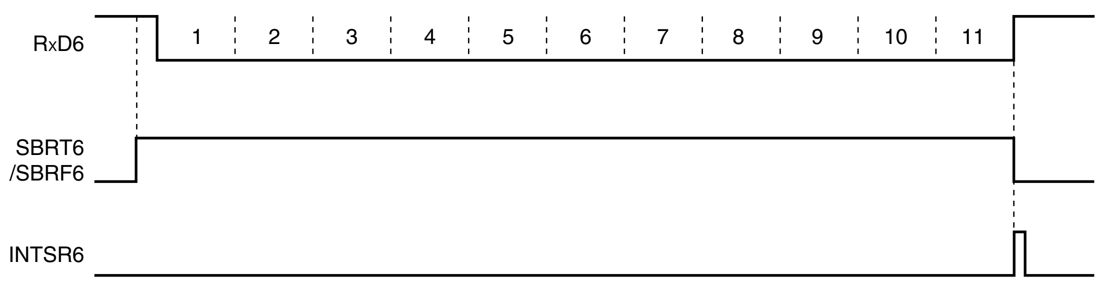
When the device is used in LIN communication operation, the SBF (Synchronous Break Field) reception control function is used for reception. For the reception operation of LIN, see **Figure 15-2 LIN Reception Operation**.

Reception is enabled when bit 7 (POWER6) of asynchronous serial interface operation mode register 6 (ASIM6) is set to 1 and then bit 5 (RXE6) of ASIM6 is set to 1. SBF reception is enabled when bit 6 (SBRT6) of asynchronous serial interface control register 6 (ASICL6) is set to 1. In the SBF reception enabled status, the RxD6 pin is sampled and the start bit is detected in the same manner as the normal reception enable status.

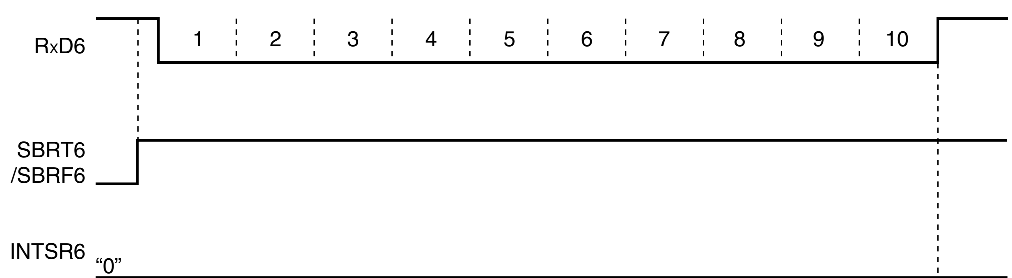
When the start bit has been detected, reception is started, and serial data is sequentially stored in the receive shift register 6 (RXS6) at the set baud rate. When the stop bit is received and if the width of SBF is 11 bits or more, a reception completion interrupt request (INTSR6) is generated as normal processing. At this time, the SBRF6 and SBRT6 bits are automatically cleared, and SBF reception ends. Detection of errors, such as OVE6, PE6, and FE6 (bits 0 to 2 of asynchronous serial interface reception error status register 6 (ASIS6)) is suppressed, and error detection processing of UART communication is not performed. In addition, data transfer between receive shift register 6 (RXS6) and receive buffer register 6 (RXB6) is not performed, and the reset value of FFH is retained. If the width of SBF is 10 bits or less, an interrupt does not occur as error processing after the stop bit has been received, and the SBF reception mode is restored. In this case, the SBRF6 and SBRT6 bits are not cleared.

Figure 15-25. SBF Reception

1. Normal SBF reception (stop bit is detected with a width of more than 10.5 bits)



2. SBF reception error (stop bit is detected with a width of 10.5 bits or less)



Remark RxD6: RxD6 pin (input)
SBRT6: Bit 6 of asynchronous serial interface control register 6 (ASICL6)
SBRF6: Bit 7 of ASICL6
INTSR6: Reception completion interrupt request

15.4.3 Dedicated baud rate generator

The dedicated baud rate generator consists of a source clock selector and an 8-bit programmable counter, and generates a serial clock for transmission/reception of UART6.

Separate 8-bit counters are provided for transmission and reception.

(1) Configuration of baud rate generator

- Base clock

  The clock selected by bits 3 to 0 (TPS63 to TPS60) of clock selection register 6 (CKSR6) is supplied to each module when bit 7 (POWER6) of asynchronous serial interface operation mode register 6 (ASIM6) is 1. This clock is called the base clock and its frequency is called f_{XCLK6}. The base clock is fixed to low level when POWER6 = 0.

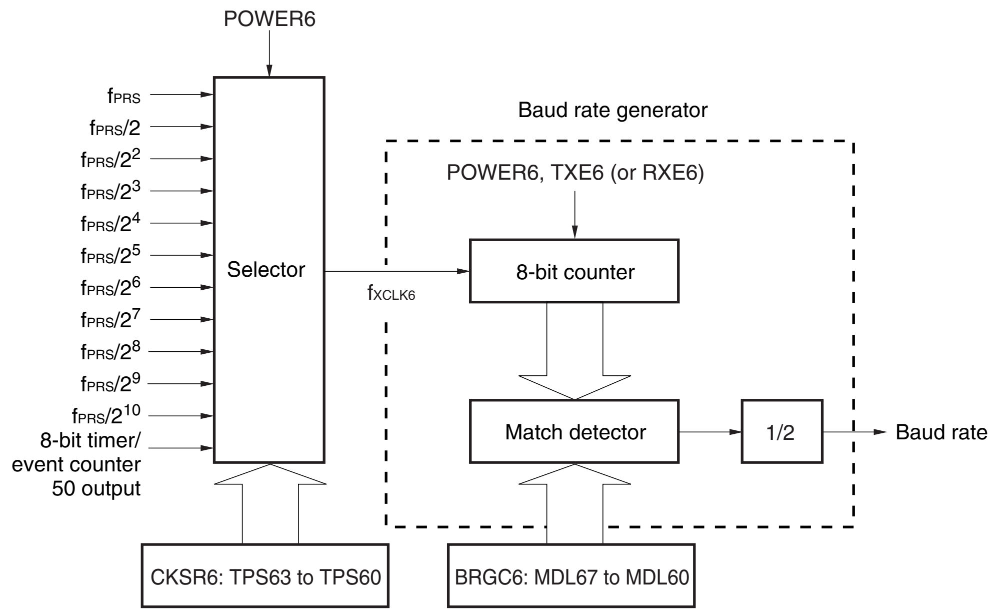
- Transmission counter

  This counter stops operation, cleared to 0, when bit 7 (POWER6) or bit 6 (TXE6) of asynchronous serial interface operation mode register 6 (ASIM6) is 0.
  It starts counting when POWER6 = 1 and TXE6 = 1.
  The counter is cleared to 0 when the first data transmitted is written to transmit buffer register 6 (TXB6).
  If data are continuously transmitted, the counter is cleared to 0 again when one frame of data has been completely transmitted. If there is no data to be transmitted next, the counter is not cleared to 0 and continues counting until POWER6 or TXE6 is cleared to 0.

- Reception counter

  This counter stops operation, cleared to 0, when bit 7 (POWER6) or bit 5 (RXE6) of asynchronous serial interface operation mode register 6 (ASIM6) is 0.
  It starts counting when the start bit has been detected.
  The counter stops operation after one frame has been received, until the next start bit is detected.

Figure 15-26. Configuration of Baud Rate Generator



Remark POWER6: Bit 7 of asynchronous serial interface operation mode register 6 (ASIM6)
TXE6: Bit 6 of ASIM6
RXE6: Bit 5 of ASIM6
CKSR6: Clock selection register 6
BRGC6: Baud rate generator control register 6

(2) Generation of serial clock

A serial clock to be generated can be specified by using clock selection register 6 (CKSR6) and baud rate generator control register 6 (BRGC6).

The clock to be input to the 8-bit counter can be set by bits 3 to 0 (TPS63 to TPS60) of CKSR6 and the division value ($f_{XCLK6}/4$ to $f_{XCLK6}/255$) of the 8-bit counter can be set by bits 7 to 0 (MDL67 to MDL60) of BRGC6.

15.4.4 Calculation of baud rate

(1) Baud rate calculation expression

The baud rate can be calculated by the following expression.

- Baud rate = $\frac{f_{XCLK6}}{2 \times k}$ [bps]

f_{XCLK6}: Frequency of base clock selected by TPS63 to TPS60 bits of CKSR6 register

k: Value set by MDL67 to MDL60 bits of BRGC6 register (k = 4, 5, 6, ..., 255)

Table 15-4. Set Value of TPS63 to TPS60

TPS63	TPS62	TPS61	TPS60	Base Clock (f_{XCLK6}) Selection[Note 1]				
					f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 8 MHz	f_{PRS} = 10 MHz
0	0	0	0	f_{PRS}[Note 2]	2 MHz	5 MHz	8 MHz	10 MHz
0	0	0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	4 MHz	5 MHz
0	0	1	0	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2 MHz	2.5 MHz
0	0	1	1	$f_{PRS}/2^3$	250 kHz	625 kHz	1 MHz	1.25 MHz
0	1	0	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	500 kHz	625 kHz
0	1	0	1	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	250 kHz	312.5 kHz
0	1	1	0	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	125 kHz	156.25 kHz
0	1	1	1	$f_{PRS}/2^7$	15.625 kHz	39.06 kHz	62.5 kHz	78.13 kHz
1	0	0	0	$f_{PRS}/2^8$	7.813 kHz	19.53 kHz	31.25 kHz	39.06 kHz
1	0	0	1	$f_{PRS}/2^9$	3.906 kHz	9.77 kHz	15.625 kHz	19.53 kHz
1	0	1	0	$f_{PRS}/2^{10}$	1.953 kHz	4.88 kHz	7.813 kHz	9.77 kHz
1	0	1	1	TM50 output[Note 3]				
Other than above				Setting prohibited				

Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{XH}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.
- V_{DD} = 2.7 to 5.5 V: $f_{PRS} \leq 10$ MHz
- V_{DD} = 1.8 to 2.7 V: $f_{PRS} \leq 5$ MHz

2. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{RH}) (XSEL = 0), when 1.8 V ≤ V_{DD} < 2.7 V, the setting of TPS63 = TPS62 = TPS61 = TPS60 = 0 (base clock: f_{PRS}) is prohibited.

3. Note the following points when selecting the TM50 output as the base clock.
- Mode in which the count clock is cleared and started upon a match of TM50 and CR50 (TMC506 = 0)
  Start the operation of 8-bit timer/event counter 50 first and then enable the timer F/F inversion operation (TMC501 = 1).
- PWM mode (TMC506 = 1)
  Start the operation of 8-bit timer/event counter 50 first and then set the count clock to make the duty = 50%.

It is not necessary to enable (TOE50 = 1) TO50 output in any mode.

(2) Error of baud rate

The baud rate error can be calculated by the following expression.

- $\text{Error (\%)} = \left(\dfrac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (correct baud rate)}} - 1 \right) \times 100\ [\%]$

Cautions 1. Keep the baud rate error during transmission to within the permissible error range at the reception destination.

2. Make sure that the baud rate error during reception satisfies the range shown in (4) Permissible baud rate range during reception.

Example: Frequency of base clock = 10 MHz = 10,000,000 Hz
Set value of MDL67 to MDL60 bits of BRGC6 register = 00100001B (k = 33)
Target baud rate = 153600 bps

Baud rate = 10 M / (2 × 33)
= 10000000 / (2 × 33) = 151,515 [bps]

Error = (151515/153600 − 1) × 100
= −1.357 [%]

(3) Example of setting baud rate

Table 15-5. Set Data of Baud Rate Generator

Baud Rate [bps]	f_{PRS} = 2.0 MHz				f_{PRS} = 5.0 MHz				f_{PRS} = 10.0 MHz			
	TPS63-TPS60	k	Calculated Value	ERR [%]	TPS63-TPS60	k	Calculated Value	ERR [%]	TPS63-TPS60	k	Calculated Value	ERR [%]
300	8H	13	301	0.16	7H	65	301	0.16	8H	65	301	0.16
600	7H	13	601	0.16	6H	65	601	0.16	7H	65	601	0.16
1200	6H	13	1202	0.16	5H	65	1202	0.16	6H	65	1202	0.16
2400	5H	13	2404	0.16	4H	65	2404	0.16	5H	65	2404	0.16
4800	4H	13	4808	0.16	3H	65	4808	0.16	4H	65	4808	0.16
9600	3H	13	9615	0.16	2H	65	9615	0.16	3H	65	9615	0.16
19200	2H	13	19231	0.16	1H	65	19231	0.16	2H	65	19231	0.16
24000	1H	21	23810	−0.79	3H	13	24038	0.16	4H	13	24038	0.16
31250	1H	16	31250	0	4H	5	31250	0	5H	5	31250	0
38400	1H	13	38462	0.16	0H	65	38462	0.16	1H	65	38462	0.16
48000	0H	21	47619	−0.79	2H	13	48077	0.16	3H	13	48077	0.16
76800	0H	13	76923	0.16	0H	33	75758	−1.36	0H	65	76923	0.16
115200	0H	9	111111	−3.55	1H	11	113636	−1.36	0H	43	116279	0.94
153600	–	–	–	–	1H	8	156250	1.73	0H	33	151515	−1.36
312500	–	–	–	–	0H	8	312500	0	1H	8	312500	0
625000	–	–	–	–	0H	4	625000	0	1H	4	625000	0

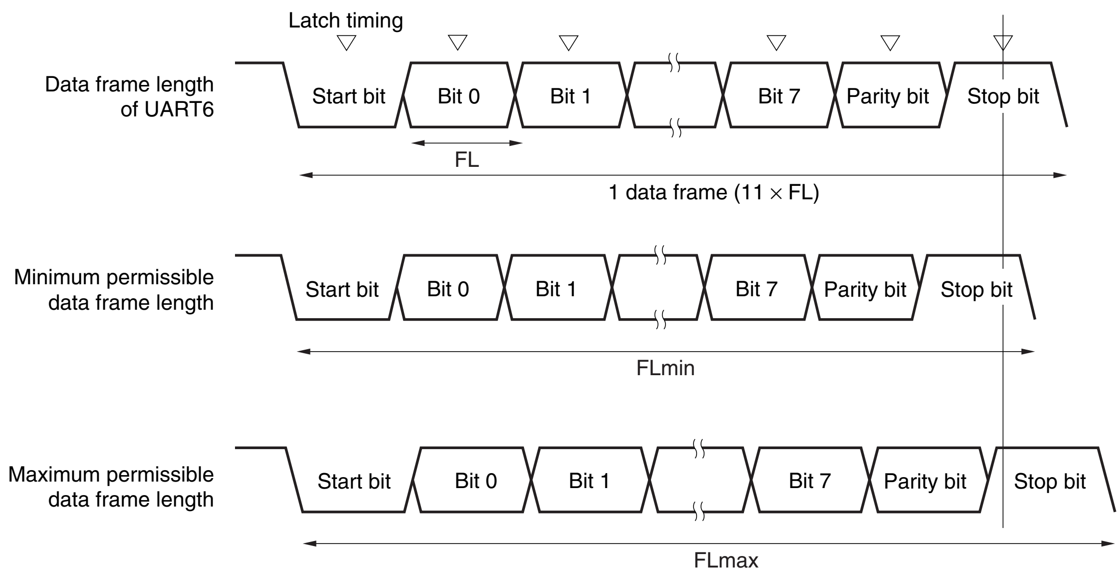
Remark TPS63 to TPS60: Bits 3 to 0 of clock selection register 6 (CKSR6) (setting of base clock (f_{XCLK6}))
k: Value set by MDL67 to MDL60 bits of baud rate generator control register 6 (BRGC6) (k = 4, 5, 6, ..., 255)
f_{PRS}: Peripheral hardware clock frequency
ERR: Baud rate error

(4) Permissible baud rate range during reception

The permissible error from the baud rate at the transmission destination during reception is shown below.

Caution Make sure that the baud rate error during reception is within the permissible error range, by using the calculation expression shown below.

Figure 15-27. Permissible Baud Rate Range During Reception



As shown in Figure 15-27, the latch timing of the receive data is determined by the counter set by baud rate generator control register 6 (BRGC6) after the start bit has been detected. If the last data (stop bit) meets this latch timing, the data can be correctly received.

Assuming that 11-bit data is received, the theoretical values can be calculated as follows.

$FL = (Brate)^{-1}$

Brate: Baud rate of UART6

k: Set value of BRGC6

FL: 1-bit data length

Margin of latch timing: 2 clocks

Minimum permissible data frame length: $FLmin = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$

Therefore, the maximum receivable baud rate at the transmission destination is as follows.

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k+2} \text{ Brate}$$

Similarly, the maximum permissible data frame length can be calculated as follows.

$$\frac{10}{11} \times FLmax = 11 \times FL - \frac{k+2}{2 \times k} \times FL = \frac{21k-2}{2 \times k} FL$$

$$FLmax = \frac{21k-2}{20k} FL \times 11$$

Therefore, the minimum receivable baud rate at the transmission destination is as follows.

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k-2} \text{ Brate}$$

The permissible baud rate error between UART6 and the transmission destination can be calculated from the above minimum and maximum baud rate expressions, as follows.

Table 15-6. Maximum/Minimum Permissible Baud Rate Error

Division Ratio (k)	Maximum Permissible Baud Rate Error	Minimum Permissible Baud Rate Error
4	+2.33%	−2.44%
8	+3.53%	−3.61%
20	+4.26%	−4.31%
50	+4.56%	−4.58%
100	+4.66%	−4.67%
255	+4.72%	−4.73%

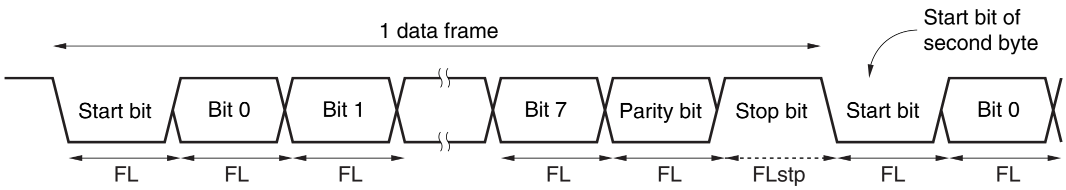
Remarks 1. The permissible error of reception depends on the number of bits in one frame, input clock frequency, and division ratio (k). The higher the input clock frequency and the higher the division ratio (k), the higher the permissible error.

2. k: Set value of BRGC6

(5) Data frame length during continuous transmission

When data is continuously transmitted, the data frame length from a stop bit to the next start bit is extended by two clocks of base clock from the normal value. However, the result of communication is not affected because the timing is initialized on the reception side when the start bit is detected.

Figure 15-28. Data Frame Length During Continuous Transmission



Where the 1-bit data length is FL, the stop bit length is FLstp, and base clock frequency is f_{XCLK6}, the following expression is satisfied.

$$FLstp = FL + 2/f_{XCLK6}$$

Therefore, the data frame length during continuous transmission is:

$$\text{Data frame length} = 11 \times FL + 2/f_{XCLK6}$$

CHAPTER 16 SERIAL INTERFACE CSI10

16.1 Functions of Serial Interface CSI10

Serial interface CSI10 has the following two modes.

(1) Operation stop mode

This mode is used when serial communication is not performed and can enable a reduction in the power consumption.

For details, see **16.4.1 Operation stop mode**.

(2) 3-wire serial I/O mode (MSB/LSB-first selectable)

This mode is used to communicate 8-bit data using three lines: a serial clock line ($\overline{SCK10}$) and two serial data lines (SI10 and SO10).

The processing time of data communication can be shortened in the 3-wire serial I/O mode because transmission and reception can be simultaneously executed.

In addition, whether 8-bit data is communicated with the MSB or LSB first can be specified, so this interface can be connected to any device.

The 3-wire serial I/O mode is used for connecting peripheral ICs and display controllers with a clocked serial interface.

For details, see **16.4.2 3-wire serial I/O mode**.

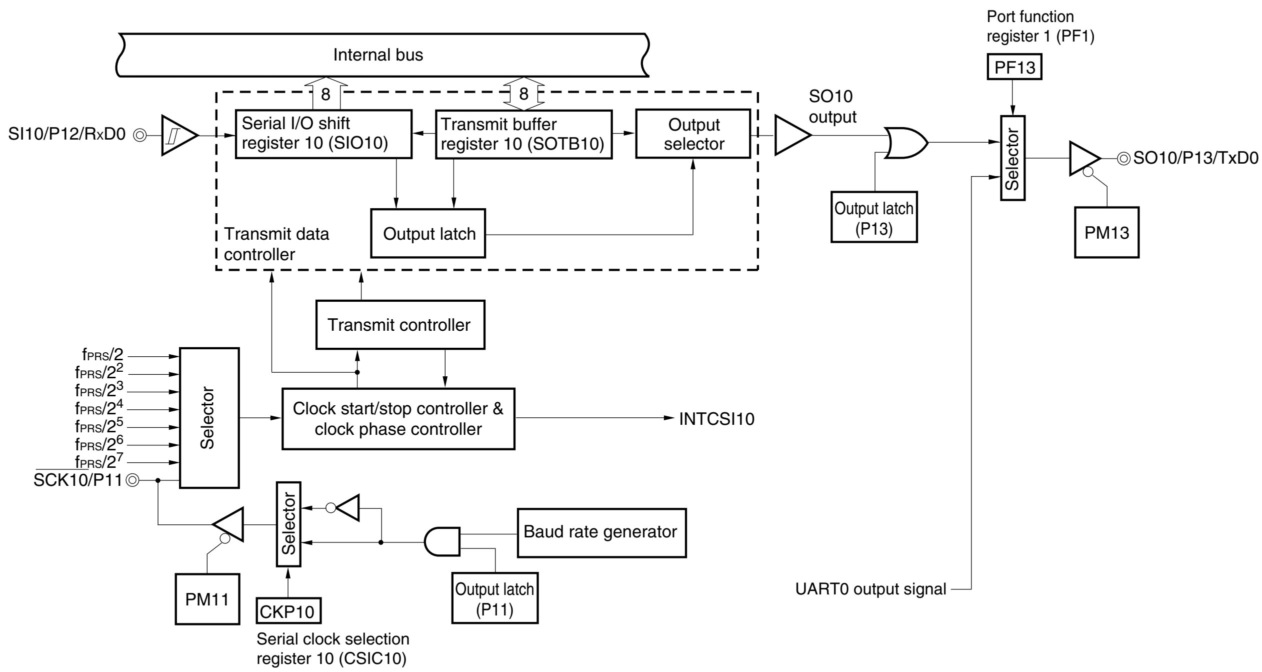
16.2 Configuration of Serial Interface CSI10

Serial interface CSI10 includes the following hardware.

Table 16-1. Configuration of Serial Interface CSI10

Item	Configuration
Controller	Transmit controller Clock start/stop controller & clock phase controller
Registers	Transmit buffer register 10 (SOTB10) Serial I/O shift register 10 (SIO10)
Control registers	Serial operation mode register 10 (CSIM10) Serial clock selection register 10 (CSIC10) Port function register 1 (PF1) Port mode register 1 (PM1) Port register 1 (P1)

Figure 16-1. Block Diagram of Serial Interface CSI10



(1) Transmit buffer register 10 (SOTB10)

This register sets the transmit data.

Transmission/reception is started by writing data to SOTB10 when bit 7 (CSIE10) and bit 6 (TRMD10) of serial operation mode register 10 (CSIM10) is 1.

The data written to SOTB10 is converted from parallel data into serial data by serial I/O shift register 10, and output to the serial output pin (SO10).

SOTB10 can be written or read by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Caution Do not access SOTB10 when CSOT10 = 1 (during serial communication).

(2) Serial I/O shift register 10 (SIO10)

This is an 8-bit register that converts data from parallel data into serial data and vice versa.

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

Reception is started by reading data from SIO10 if bit 6 (TRMD10) of serial operation mode register 10 (CSIM10) is 0.

During reception, the data is read from the serial input pin (SI10) to SIO10.

Reset signal generation sets this register to 00H.

Caution Do not access SIO10 when CSOT10 = 1 (during serial communication).

16.3 Registers Controlling Serial Interface CSI10

Serial interface CSI10 is controlled by the following five registers.

- Serial operation mode register 10 (CSIM10)
- Serial clock selection register 10 (CSIC10)
- Port function register 1 (PF1)
- Port mode register 1 (PM1)
- Port register 1 (P1)

(1) Serial operation mode register 10 (CSIM10)

CSIM10 is used to select the operation mode and enable or disable operation.
CSIM10 can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to 00H.

Figure 16-2. Format of Serial Operation Mode Register 10 (CSIM10)

Address: FF80H After reset: 00H R/W[Note 1]

Symbol	<7>	6	5	4	3	2	1	0
CSIM10	CSIE10	TRMD10	0	DIR10	0	0	0	CSOT10

CSIE10[Note 2]	Operation control in 3-wire serial I/O mode
0	Disables operation and asynchronously resets the internal circuit[Note 3].
1	Enables operation

TRMD10[Note 4]	Transmit/receive mode control
0[Note 5]	Receive mode (transmission disabled).
1	Transmit/receive mode

DIR10[Note 6]	First bit specification
0	MSB
1	LSB

CSOT10	Communication status flag
0	Communication is stopped.
1	Communication is in progress.

Notes 1. Bit 0 is a read-only bit.

2. To use P11/$\overline{\text{SCK10}}$, P12/SI10/RxD0, and P13/SO10/TxD0 as general-purpose port, clear CSIE10 to 0.

3. Bit 0 (CSOT10) of CSIM10 and serial I/O shift register 10 (SIO10) are reset.

4. Do not rewrite TRMD10 when CSOT10 = 1 (during serial communication).

5. The SO10 output (see **Figure 16-1**) is fixed to the low level when TRMD10 is 0. Reception is started when data is read from SIO10.

6. Do not rewrite DIR10 when CSOT10 = 1 (during serial communication).

Cautions 1. When resuming operation from standby status, do so after having cleared (0) bit 2 (CSIIF10) of interrupt request flag register 0H (IF0H).

2. Be sure to clear bit 5 to 0.

(2) Serial clock selection register 10 (CSIC10)

This register specifies the timing of the data transmission/reception and sets the serial clock.

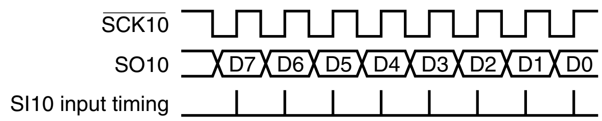
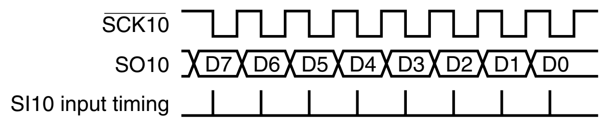
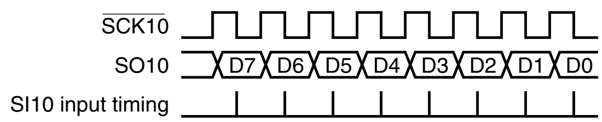
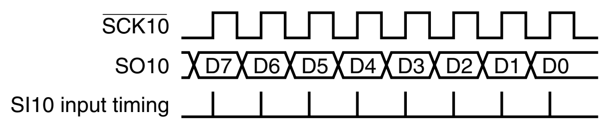
CSIC10 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Figure 16-3. Format of Serial Clock Selection Register 10 (CSIC10)

Address: FF81H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIC10	0	0	0	CKP10	DAP10	CKS102	CKS101	CKS100

CKP10	DAP10	Specification of data transmission/reception timing	Type
0	0		1
0	1		2
1	0		3
1	1		4

CKS102	CKS101	CKS100	CSI10 serial clock selection[Notes 1, 2]					Mode
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 8 MHz	f_{PRS} = 10 MHz	
0	0	0	$f_{PRS}/2$	1 MHz	2.5 MHz	4 MHz	Setting prohibited	Master mode
0	0	1	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2 MHz	2.5 MHz	
0	1	0	$f_{PRS}/2^3$	250 kHz	625 kHz	1 MHz	1.25 MHz	
0	1	1	$f_{PRS}/2^4$	125 kHz	312.5 kHz	500 kHz	625 kHz	
1	0	0	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	250 kHz	312.5 kHz	
1	0	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	125 kHz	156.25 kHz	
1	1	0	$f_{PRS}/2^7$	15.63 kHz	39.06 kHz	62.5 kHz	78.13 kHz	
1	1	1	External clock input to $\overline{SCK10}$					Slave mode

Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{XH}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 2.7 to 5.5 V: $f_{PRS} \leq 10$ MHz
- V_{DD} = 1.8 to 2.7 V: $f_{PRS} \leq 5$ MHz

Notes 2. Set the serial clock to satisfy the following conditions.

- V_{DD} = 2.7 to 5.5 V: serial clock ≤ 4 MHz
- V_{DD} = 1.8 to 2.7 V: serial clock ≤ 2 MHz

Cautions 1. Do not write to CSIC10 while CSIE10 = 1 (operation enabled).

2. To use P11/$\overline{\text{SCK10}}$ and P13/SO10/TxD0 as general-purpose ports, set CSIC10 in the default status (00H).

3. The phase type of the data clock is type 1 after reset.

Remark f_{PRS}: Peripheral hardware clock oscillation frequency

(3) Port function register 1 (PF1)

This register sets the pin functions of P13/SO10/TxD0 pin.

PF1 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PF1 to 00H.

Figure 16-4. Format of Port Function Register 1 (PF1)

Address: FF20H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PF1	0	PF16	0	0	PF13	0	0	0

PF16	Port (P16), CSIA0, and UART6 output specification
0	Used as P16 or SOA0
1	Used as TxD6

PF13	Port (P13), CSI10, and UART0 output specification
0	Used as P13 or SO10
1	Used as TxD0

(4) Port mode register 1 (PM1)

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

When using P11/$\overline{SCK10}$ as the clock output pin of the serial interface, clear PM11 to 0, and set the output latches of P11 to 1.

When using P13/SO10/TxD0 as the data output pin of the serial interface, clear PM13 and the output latches of P13 to 0.

When using P11/$\overline{SCK10}$ as the clock input pin of the serial interface and P12/SI10/RxD0 as the data input pin, set PM11 and PM12 to 1. At this time, the output latches of P11 and P12 may be 0 or 1.

PM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 16-5. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

16.4 Operation of Serial Interface CSI10

Serial interface CSI10 can be used in the following two modes.

- Operation stop mode
- 3-wire serial I/O mode

16.4.1 Operation stop mode

Serial communication is not executed in this mode. Therefore, the power consumption can be reduced. In addition, the P11/SCK10, P12/SI10/RxD0, and P13/SO10/TxD0 pins can be used as ordinary I/O port pins in this mode.

(1) Register used

The operation stop mode is set by serial operation mode register 10 (CSIM10).

To set the operation stop mode, clear bit 7 (CSIE10) of CSIM10 to 0.

(a) Serial operation mode register 10 (CSIM10)

CSIM10 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets CSIM10 to 00H.

Address: FF80H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CSIM10	CSIE10	TRMD10	0	DIR10	0	0	0	CSOT10

CSIE10	Operation control in 3-wire serial I/O mode
0	Disables operation[Note 1] and asynchronously resets the internal circuit[Note 2].

Notes 1. To use P11/SCK10, P12/SI10/RxD0, and P13/SO10/TxD0 as general-purpose ports, set CSIM10 in the default status (00H).

2. Bit 0 (CSOT10) of CSIM10 and serial I/O shift register 10 (SIO10) are reset.

16.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode is used for connecting peripheral ICs and display controllers with a clocked serial interface.

In this mode, communication is executed by using three lines: the serial clock (SCK10), serial output (SO10), and serial input (SI10) lines.

(1) Registers used

- Serial operation mode register 10 (CSIM10)
- Serial clock selection register 10 (CSIC10)
- Port mode register 1 (PM1)
- Port register 1 (P1)

The basic procedure of setting an operation in the 3-wire serial I/O mode is as follows.

<1> Set the CSIC10 register (see **Figures 16-3**).
<2> Set bits 4 and 6 (DIR10 and TRMD10) of the CSIM10 register (see **Figures 16-2**).
<3> Set bit 7 (CSIE10) of the CSIM10 register to 1. → Transmission/reception is enabled.
<4> Write data to transmit buffer register 10 (SOTB10). → Data transmission/reception is started.
Read data from serial I/O shift register 10 (SIO10). → Data reception is started.

Caution Take relationship with the other party of communication when setting the port mode register and port register.

The relationship between the register settings and pins is shown below.

Table 16-2. Relationship Between Register Settings and Pins

CSIE10	TRMD10	PM12	P12	PM13	P13	PM11	P11	CSI10 Operation	Pin Function		
									SI10/RxD0/ P12	SO10/ TxD0/P13	$\overline{\text{SCK10}}$/ P11
0	×	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	Stop	RxD0/P12	TxD0/P13	P11[Note 2]
1	0	1	×	×[Note 1]	×[Note 1]	1	×	Slave reception[Note 3]	SI10	TxD0/P13	$\overline{\text{SCK10}}$ (input)[Note 3]
1	1	×[Note 1]	×[Note 1]	0	0	1	×	Slave transmission[Note 3]	RxD0/P12	SO10	$\overline{\text{SCK10}}$ (input)[Note 3]
1	1	1	×	0	0	1	×	Slave transmission/ reception[Note 3]	SI10	SO10	$\overline{\text{SCK10}}$ (input)[Note 3]
1	0	1	×	×[Note 1]	×[Note 1]	0	1	Master reception	SI10	TxD0/P13	$\overline{\text{SCK10}}$ (output)
1	1	×[Note 1]	×[Note 1]	0	0	0	1	Master transmission	RxD0/P12	SO10	$\overline{\text{SCK10}}$ (output)
1	1	1	×	0	0	0	1	Master transmission/ reception	SI10	SO10	$\overline{\text{SCK10}}$ (output)

Notes 1. Can be set as port function.
2. To use P11/$\overline{\text{SCK10}}$ as port pins, clear CKP10 to 0.
3. To use the slave mode, set CKS102, CKS101, and CKS100 to 1, 1, 1.

Remark
×: don’t care
CSIE10: Bit 7 of serial operation mode register 10 (CSIM10)
TRMD10: Bit 6 of CSIM10
CKP10: Bit 4 of serial clock selection register 10 (CSIC10)
CKS102, CKS101, CKS100: Bits 2 to 0 of CSIC10
PM1×: Port mode register
P1×: Port output latch

(2) Communication operation

In the 3-wire serial I/O mode, data is transmitted or received in 8-bit units. Each bit of the data is transmitted or received in synchronization with the serial clock.

Data can be transmitted or received if bit 6 (TRMD10) of serial operation mode register 10 (CSIM10) is 1. Transmission/reception is started when a value is written to transmit buffer register 10 (SOTB10). In addition, data can be received when bit 6 (TRMD10) of serial operation mode register 10 (CSIM10) is 0.

Reception is started when data is read from serial I/O shift register 10 (SIO10).

After communication has been started, bit 0 (CSOT10) of CSIM10 is set to 1. When communication of 8-bit data has been completed, a communication completion interrupt request flag (CSIIF10) is set, and CSOT10 is cleared to 0. Then the next communication is enabled.

Caution Do not access the control register and data register when CSOT10 = 1 (during serial communication).

Figure 16-6. Timing in 3-Wire Serial I/O Mode (1/2)

(a) Transmission/reception timing (Type 1: TRMD10 = 1, DIR10 = 0, CKP10 = 0, DAP10 = 0)

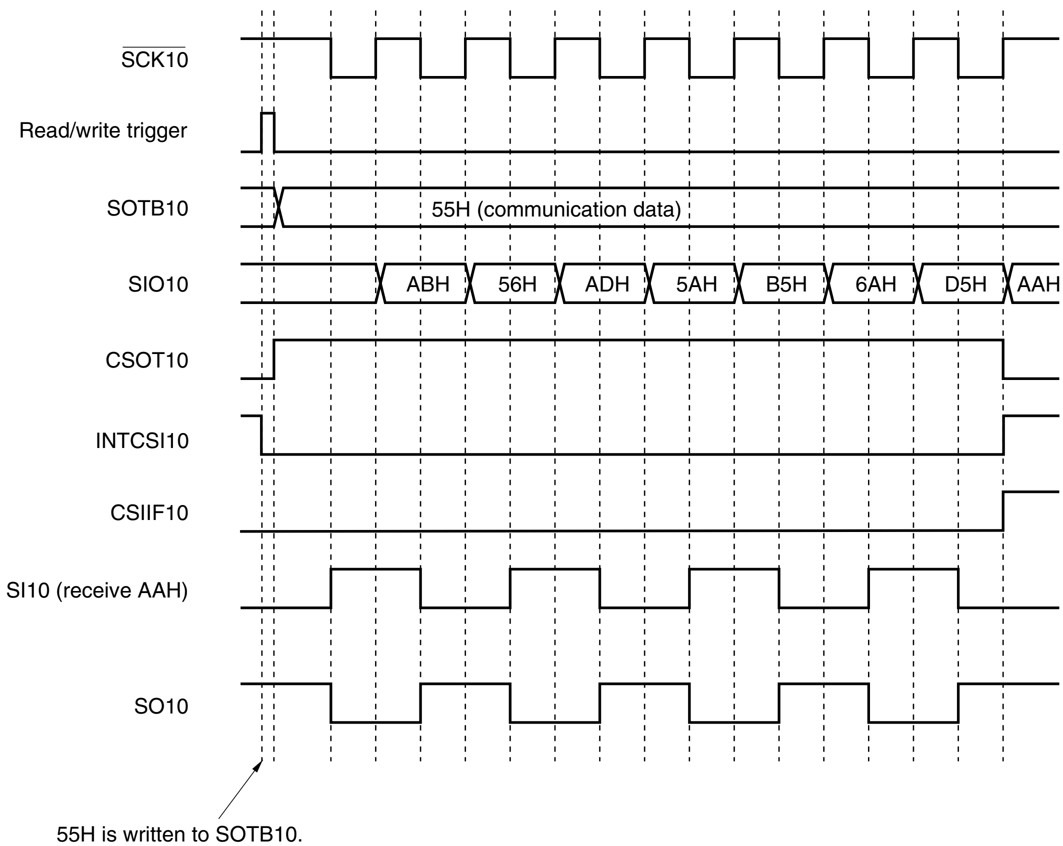


Figure 16-6. Timing in 3-Wire Serial I/O Mode (2/2)

(b) Transmission/reception timing (Type 2: TRMD10 = 1, DIR10 = 0, CKP10 = 0, DAP10 = 1)

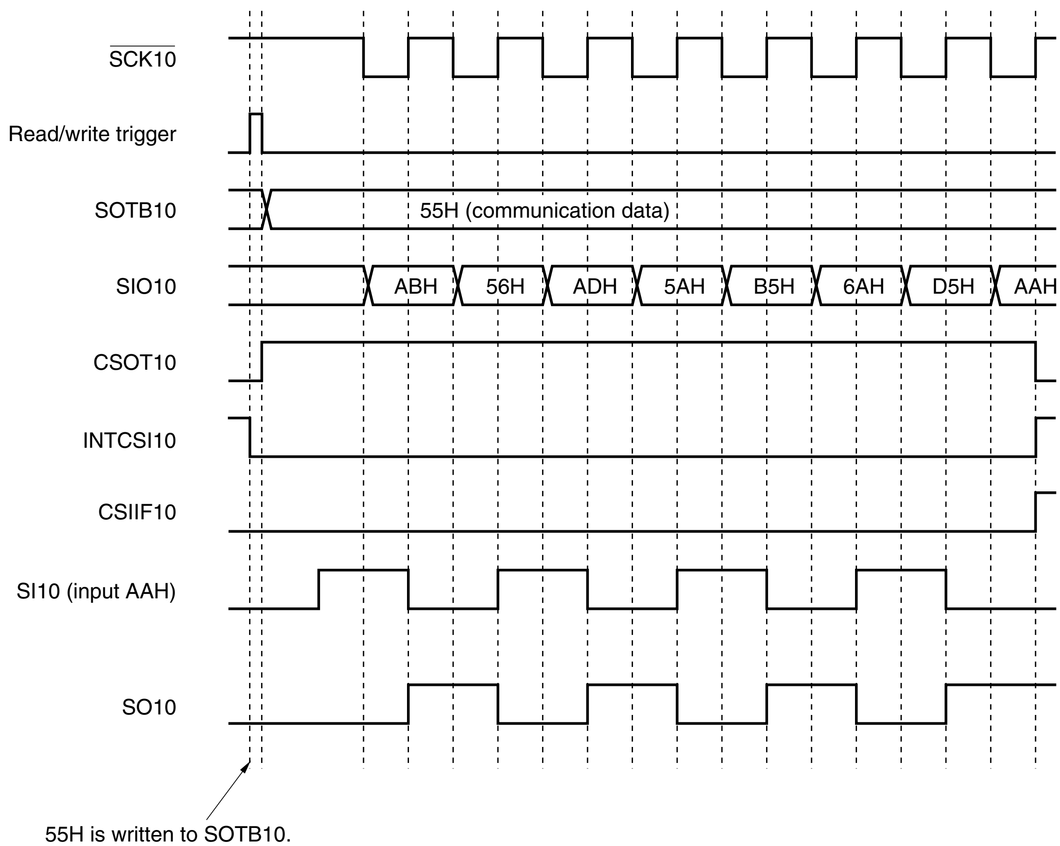
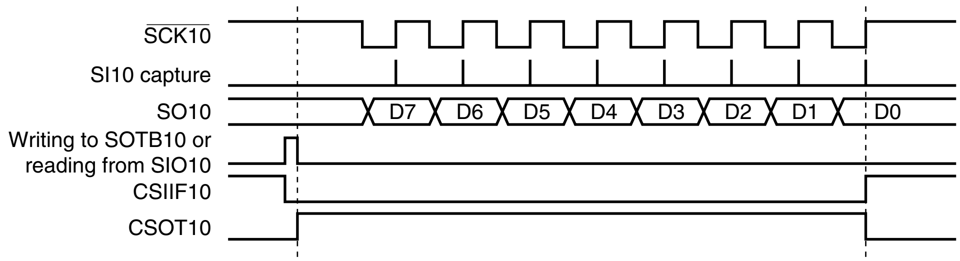
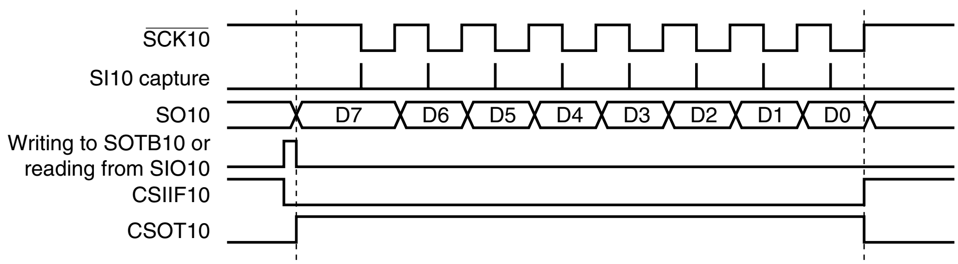


Figure 16-7. Timing of Clock/Data Phase

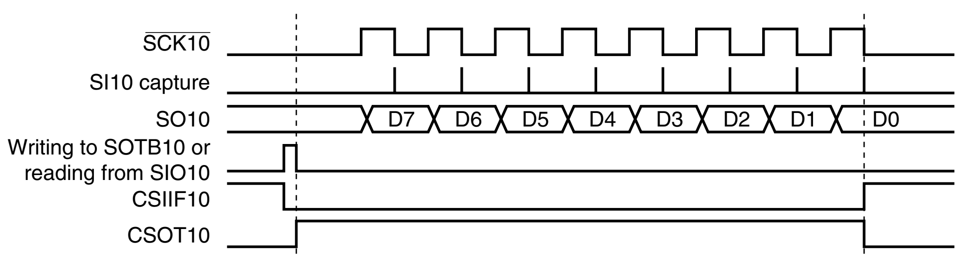
(a) Type 1: CKP10 = 0, DAP10 = 0, DIR10 = 0



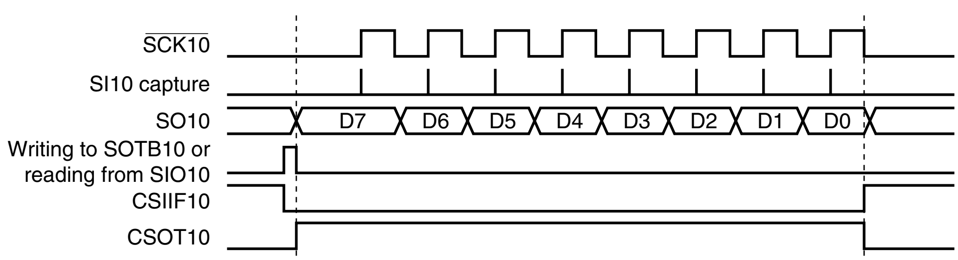
(b) Type 2: CKP10 = 0, DAP10 = 1, DIR10 = 0



(c) Type 3: CKP10 = 1, DAP10 = 0, DIR10 = 0



(d) Type 4: CKP10 = 1, DAP10 = 1, DIR10 = 0



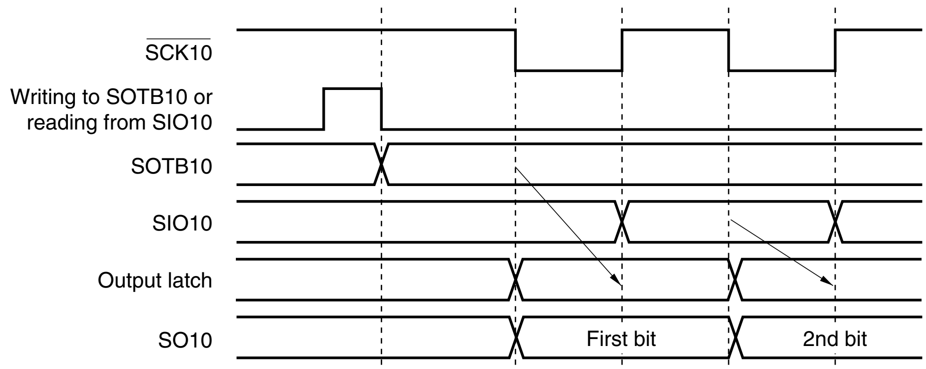
Remark The above figure illustrates a communication operation where data is transmitted with the MSB first.

(3) Timing of output to SO10 pin (first bit)

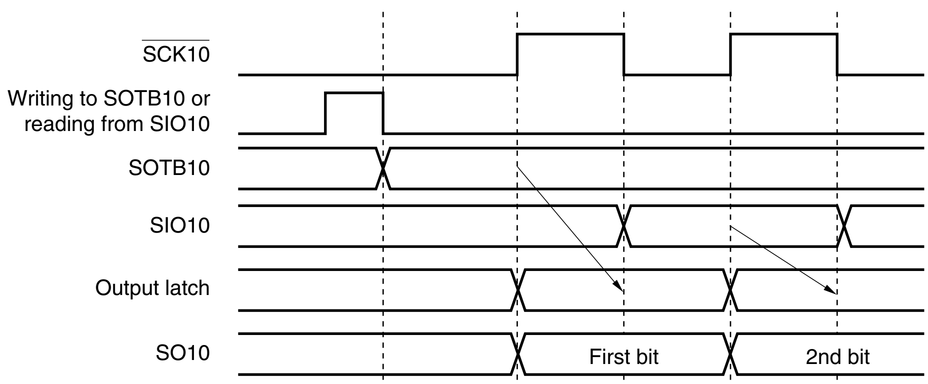
When communication is started, the value of transmit buffer register 10 (SOTB10) is output from the SO10 pin. The output operation of the first bit at this time is described below.

Figure 16-8. Output Operation of First Bit (1/2)

(a) Type 1: CKP10 = 0, DAP10 = 0



(b) Type 3: CKP10 = 1, DAP10 = 0

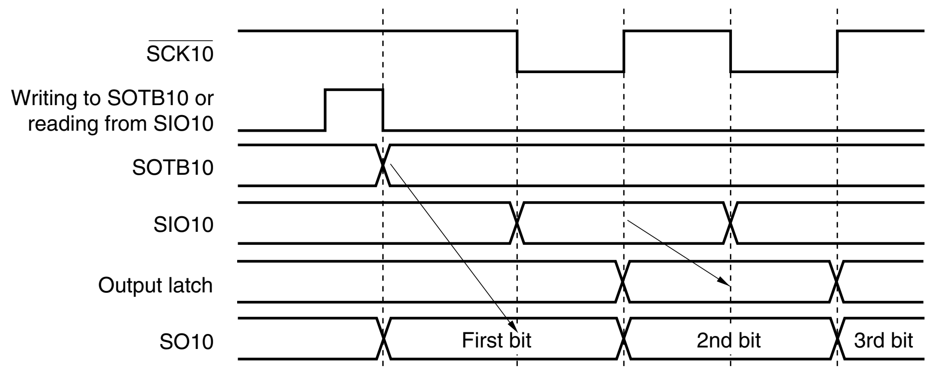


The first bit is directly latched by the SOTB10 register to the output latch at the falling (or rising) edge of $\overline{SCK10}$, and output from the SO10 pin via an output selector. Then, the value of the SOTB10 register is transferred to the SIO10 register at the next rising (or falling) edge of $\overline{SCK10}$, and shifted one bit. At the same time, the first bit of the receive data is stored in the SIO10 register via the SI10 pin.

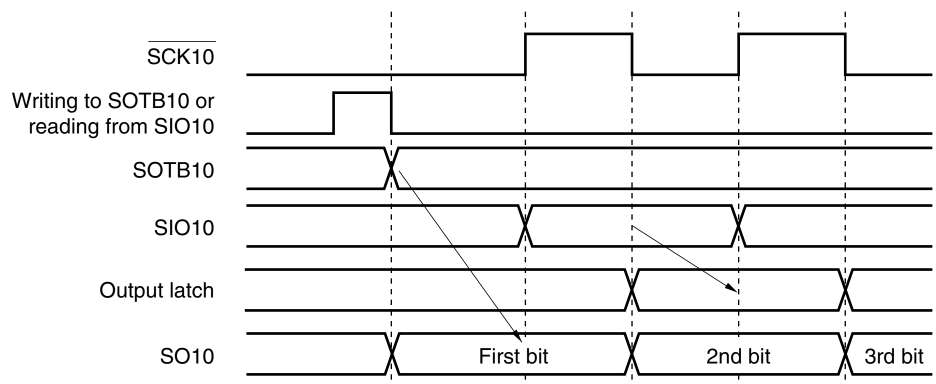
The second and subsequent bits are latched by the SIO10 register to the output latch at the next falling (or rising) edge of $\overline{SCK10}$, and the data is output from the SO10 pin.

Figure 16-8. Output Operation of First Bit (2/2)

(c) Type 2: CKP10 = 0, DAP10 = 1



(d) Type 4: CKP10 = 1, DAP10 = 1



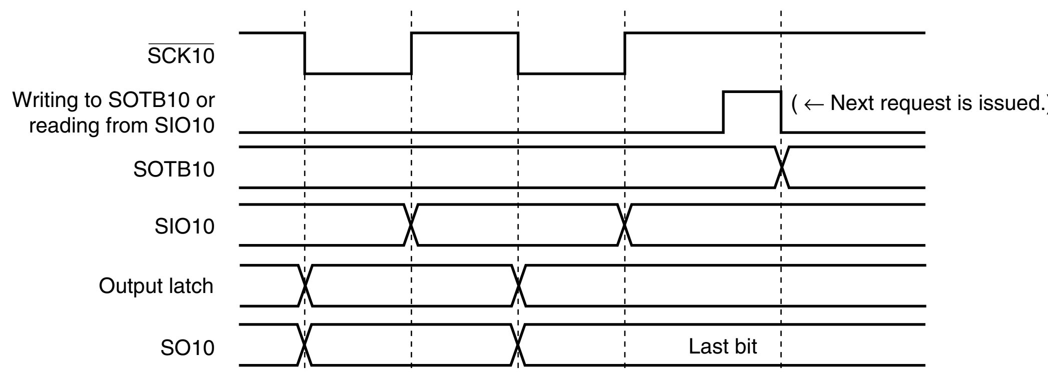
The first bit is directly latched by the SOTB10 register at the falling edge of the write signal of the SOTB10 register or the read signal of the SIO10 register, and output from the SO10 pin via an output selector. Then, the value of the SOTB10 register is transferred to the SIO10 register at the next falling (or rising) edge of $\overline{\text{SCK10}}$, and shifted one bit. At the same time, the first bit of the receive data is stored in the SIO10 register via the SI10 pin.
The second and subsequent bits are latched by the SIO10 register to the output latch at the next rising (or falling) edge of $\overline{\text{SCK10}}$, and the data is output from the SO10 pin.

(4) Output value of SO10 pin (last bit)

After communication has been completed, the SO10 pin holds the output value of the last bit.

Figure 16-9. Output Value of SO10 Pin (Last Bit) (1/2)

(a) Type 1: CKP10 = 0, DAP10 = 0



(b) Type 3: CKP10 = 1, DAP10 = 0

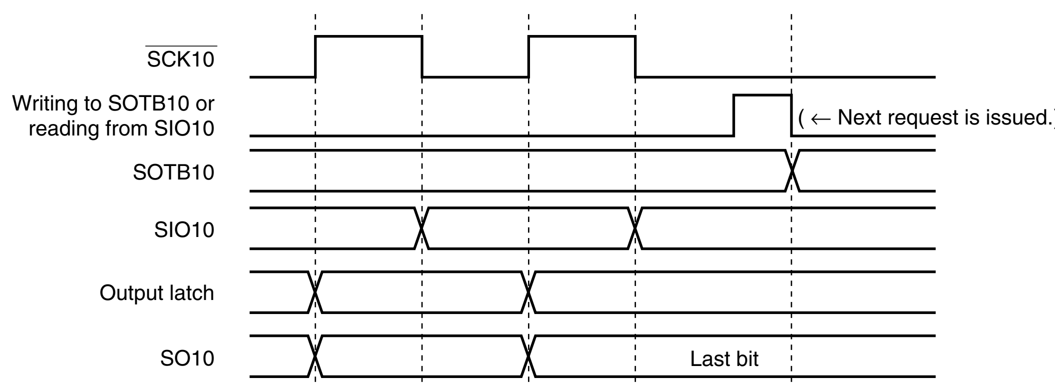
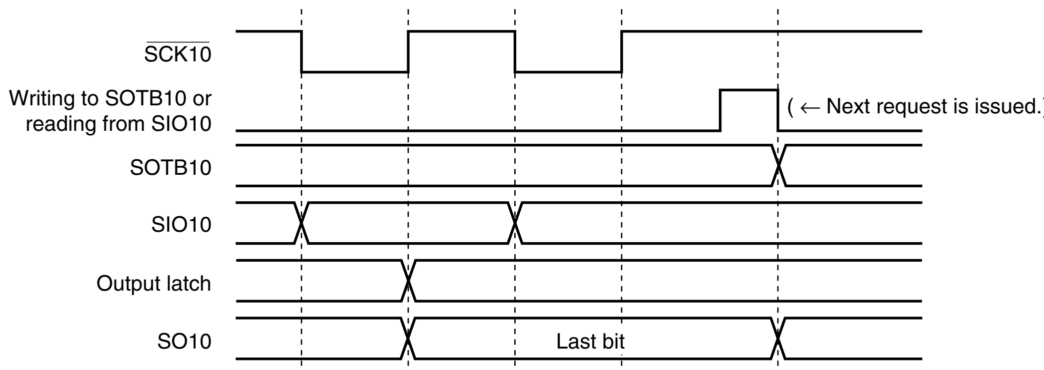
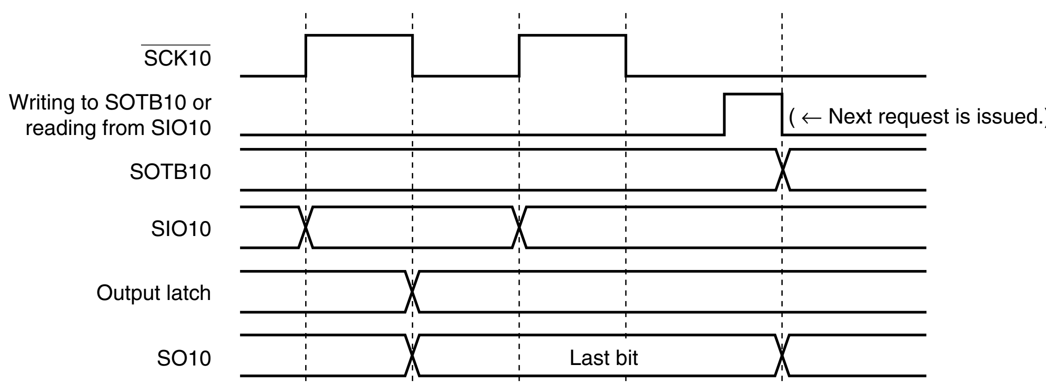


Figure 16-9. Output Value of SO10 Pin (Last Bit) (2/2)

(c) Type 2: CKP10 = 0, DAP10 = 1



(d) Type 4: CKP10 = 1, DAP10 = 1



(5) SO10 output (see Figure 16-1)

The status of the SO10 output is as follows if bit 7 (CSIE10) of serial operation mode register 10 (CSIM10) is cleared to 0.

Table 16-3. SO10 Output Status

TRMD10	DAP10	DIR10	SO10 Output[Note 1]
TRMD10 = 0[Note 2]	–	–	Outputs low level[Note 2]
TRMD10 = 1	DAP10 = 0	–	Value of SO10 latch (low-level output)
	DAP10 = 1	DIR10 = 0	Value of bit 7 of SOTB10
		DIR10 = 1	Value of bit 0 of SOTB10

Notes 1. The actual output of the SO10/P13 pin is determined according to PM13 and P13, as well as the SO10 output.

2. Status after reset

Caution If a value is written to TRMD10, DAP10, and DIR10, the output value of SO10 changes.

CHAPTER 17 SERIAL INTERFACE CSIA0

17.1 Functions of Serial Interface CSIA0

Serial interface CSIA0 has the following three modes.

(1) Operation stop mode

This mode is used when serial communication is not performed and can enable a reduction in the power consumption.

For details, see **17.4.1 Operation stop mode**.

(2) 3-wire serial I/O mode (MSB/LSB-first selectable)

This mode is to communicate data successively in 8-bit units, by using three lines: serial clock ($\overline{SCKA0}$) and serial data (SIA0 and SOA0) lines.

The processing time of data communication can be shortened in the 3-wire serial I/O mode because transmission and reception can be simultaneously executed.

In addition, whether 8-bit data is communicated MSB or LSB first can be specified, so this interface can be connected to any device.

For details, see **17.4.2 3-wire serial I/O mode**.

(3) 3-wire serial I/O mode with automatic transmit/receive function (MSB/LSB-first selectable)

This mode is used to communicate data continuously in 8-bit units using three lines: a serial clock line ($\overline{SCKA0}$) and two serial data lines (SIA0 and SOA0).

The processing time of data communication can be shortened in the 3-wire serial I/O mode with automatic transmit/receive function because transmission and reception can be simultaneously executed.

In addition, whether 8-bit data is communicated MSB or LSB first can be specified, so this interface can be connected to any device.

Data can be communicated to/from a display driver etc. without using software since a 32-byte transfer buffer RAM is incorporated.

For details, see **17.4.3 3-wire serial I/O mode with automatic transmit/receive function**.

The features of serial interface CSIA0 are as follows.

- Master mode/slave mode selectable
- Communication data length: 8 bits
- MSB/LSB-first selectable for communication data
- Automatic transmit/receive function:
  Number of transfer bytes can be specified between 1 and 32
  Transfer interval can be specified (0 to 63 clocks)
  Single communication/repeat communication selectable
  Internal 32-byte buffer RAM
- On-chip dedicated baud rate generator (6/8/16/32 divisions)
- 3-wire SOA0: Serial data output
  SIA0: Serial data input
  $\overline{SCKA0}$: Serial clock I/O
- Transmission/reception completion interrupt: INTACSI

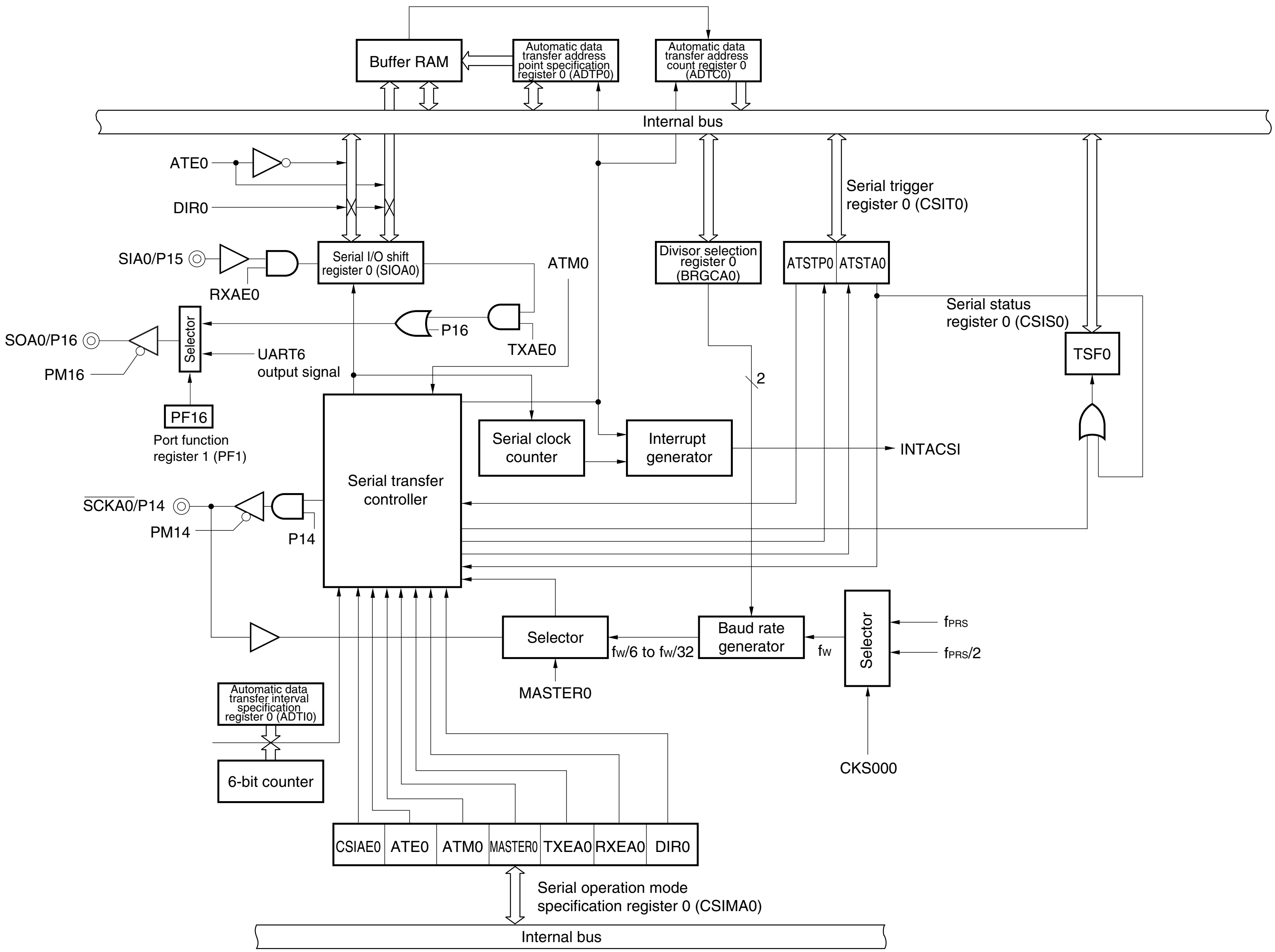
17.2 Configuration of Serial Interface CSIA0

Serial interface CSIA0 consists of the following hardware.

Table 17-1. Configuration of Serial Interface CSIA0

Item	Configuration
Controller	Serial transfer controller
Registers	Serial I/O shift register 0 (SIOA0)
Control registers	Serial operation mode specification register 0 (CSIMA0) Serial status register 0 (CSIS0) Serial trigger register 0 (CSIT0) Divisor selection register 0 (BRGCA0) Automatic data transfer address point specification register 0 (ADTP0) Automatic data transfer interval specification register 0 (ADTI0) Automatic data transfer address count register 0 (ADTC0) Port function register 1 (PF1) Port mode register 1 (PM1) Port register 1 (P1)

Figure 17-1. Block Diagram of Serial Interface CSIA0



(1) Serial I/O shift register 0 (SIOA0)

This is an 8-bit register used to store transmit/receive data in 1-byte transfer mode (bit 6 (ATE0) of serial operation mode specification register 0 (CSIMA0) = 0). Writing transmit data to SIOA0 starts the communication. In addition, after a communication completion interrupt request (INTACSI) is output (bit 0 (TSF0) of serial status register 0 (CSIS0) = 0), data can be received by reading data from SIOA0.

This register can be written or read by an 8-bit memory manipulation instruction. However, writing to SIOA0 is prohibited when bit 0 (TSF0) of serial status register 0 (CSIS0) = 1.

Reset signal generation clears this register to 00H.

Cautions 1. A communication operation is started by writing to SIOA0. Consequently, when transmission is disabled (bit 3 (TXEA0) of CSIMA0 = 0), write dummy data to the SIOA0 register to start the communication operation, and then perform a receive operation.

2. Do not write data to SIOA0 while the automatic transmit/receive function is operating.

17.3 Registers Controlling Serial Interface CSIA0

Serial interface CSIA0 is controlled by the following ten registers.

- Serial operation mode specification register 0 (CSIMA0)
- Serial status register 0 (CSIS0)
- Serial trigger register 0 (CSIT0)
- Divisor selection register 0 (BRGCA0)
- Automatic data transfer address point specification register 0 (ADTP0)
- Automatic data transfer interval specification register 0 (ADTI0)
- Automatic data transfer address count register 0 (ADTC0)
- Port function register 1 (PF1)
- Port mode register 1 (PM1)
- Port register 1 (P1)

(1) Serial operation mode specification register 0 (CSIMA0)

This is an 8-bit register used to control the serial communication operation.

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

Reset signal generation clears this register to 00H.

Figure 17-2. Format of Serial Operation Mode Specification Register 0 (CSIMA0)

Address: FF90H After reset: 00H R/W

Symbol	<7>	6	5	4	<3>	<2>	1	0
CSIMA0	CSIAE0	ATE0	ATM0	MASTER0	TXEA0	RXEA0	DIR0	0

CSIAE0	Control of CSIA0 operation enable/disable
0	CSIA0 operation disabled (SOA0: Low level, $\overline{\text{SCKA0}}$: High level) and asynchronously resets the internal circuit[Note].
1	CSIA0 operation enabled

ATE0	Control of automatic communication operation enable/disable
0	1-byte communication mode
1	Automatic communication mode

ATM0	Automatic communication mode specification
0	Single transfer mode (stops at the address specified by the ADTP0 register)
1	Repeat transfer mode (after transfer is complete, clear the ADTC0 register to 00H to resume transfer)

MASTER0	CSIA0 master/slave mode specification
0	Slave mode (synchronous with $\overline{\text{SCKA0}}$ input clock)
1	Master mode (synchronous with internal clock)

TXEA0	Control of transmit operation enable/disable
0	Transmit operation disabled (SOA0: Low level)
1	Transmit operation enabled

RXEA0	Control of receive operation enable/disable
0	Receive operation disabled
1	Receive operation enabled

DIR0	First bit specification
0	MSB
1	LSB

Note Automatic data transfer address count register 0 (ADTC0), serial trigger register 0 (CSIT0), serial I/O shift register 0 (SIOA0), and bit 0 (TSF0) of serial status register 0 (CSIS0) are reset.

Cautions 1. When CSIAE0 = 0, the buffer RAM cannot be accessed.

2. When CSIAE0 is changed from 1 to 0, the registers and bits mentioned in Note above are asynchronously initialized. To set CSIAE0 = 1 again, be sure to re-set the initialized registers.

3. When CSIAE0 is re-set to 1 after CSIAE0 is changed from 1 to 0, it is not guaranteed that the value of the buffer RAM will be retained.

(2) Serial status register 0 (CSIS0)

This is an 8-bit register used to select the base clock, control the communication operation, and indicate the status of serial interface CSIA0.

This register can be set by a 1-bit or 8-bit memory manipulation instruction. However, rewriting CSIS0 is prohibited when bit 0 (TSF0) is 1.

Reset signal generation clears this register to 00H.

Figure 17-3. Format of Serial Status Register 0 (CSIS0)

Address: FF91H After reset: 00H R/W[Note 1]

Symbol	7	6	5	4	3	2	1	0
CSIS0	0	CKS00	0	0	0	0	0	TSF0

CKS00	Base clock (f_W) selection[Note 2]				
		f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 8 MHz	f_{PRS} = 10 MHz
0	f_{PRS}[Note 3]	2 MHz	5 MHz	8 MHz	10 MHz
1	$f_{PRS}/2$	1 MHz	2.5 MHz	4 MHz	5 MHz

TSF0	Transfer status detection flag
0	• Bit 7 (CSIAE0) of serial operation mode specification register 0 (CSIMA0) = 0 • At reset input • At the end of the specified transfer • When transfer is stopped by setting bit 1 (ATSTP0) of serial trigger register 0 (CSIT0) to 1
1	From the transfer start to the end of the specified transfer

Notes 1. Bit 0 is read-only.

2. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{XH}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 2.7 to 5.5 V: $f_{PRS} \leq 10$ MHz
- V_{DD} = 1.8 to 2.7 V: $f_{PRS} \leq 5$ MHz

3. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{RH}) (XSEL = 0), when 1.8 V ≤ V_{DD} < 2.7 V, the setting of CKS00 = 0 (base clock: f_{PRS}) is prohibited.

Cautions 1. Be sure to clear bits 7 and 5 to 1.

2. During transfer (TSF0 = 1), rewriting serial operation mode specification register 0 (CSIMA0), serial status register 0 (CSIS0), divisor selection register 0 (BRGCA0), automatic data transfer address point specification register 0 (ADTP0), automatic data transfer interval specification register 0 (ADTI0), and serial I/O shift register 0 (SIOA0) are prohibited. However, these registers can be read and re-written to the same value. In addition, the buffer RAM can be rewritten during transfer.

Remark f_{PRS}: Peripheral hardware clock frequency

(3) Serial trigger register 0 (CSIT0)

This is an 8-bit register used to control execution/stop of automatic data transfer between buffer RAM and serial I/O shift register 0 (SIOA0).

This register can be set by a 1-bit or 8-bit memory manipulation instruction. This register can be set when bit 6 (ATE0) of serial operation mode specification register 0 (CSIMA0) is 1.

Reset signal generation clears this register to 00H.

Figure 17-4. Format of Serial Trigger Register 0 (CSIT0)

Address: FF92H After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
CSIT0	0	0	0	0	0	0	ATSTP0	ATSTA0

ATSTP0	Automatic data transfer stop
0	–
1	Automatic data transfer stopped

ATSTA0	Automatic data transfer start
0	–
1	Automatic data transfer started

Cautions 1. Even if ATSTP0 or ATSTA0 is set to 1, automatic transfer cannot be started/stopped until 1-byte transfer is complete.

2. ATSTP0 and ATSTA0 change to 0 automatically after the interrupt signal INTACSI is generated.

3. After automatic data transfer is stopped, the data address when the transfer stopped is stored in automatic data transfer address count register 0 (ADTC0). However, since no function to restart automatic data transfer is incorporated, when transfer is stopped by setting ATSTP0 = 1, start automatic data transfer by setting ATSTA0 to 1 after re-setting the registers.

(4) Divisor selection register 0 (BRGCA0)

This is an 8-bit register used to select the base clock divisor of CSIA0.

This register can be set by an 8-bit memory manipulation instruction. However, when bit 0 (TSF0) of serial status register 0 (CSIS0) is 1, rewriting BRGCA0 is prohibited.

Reset signal generation sets this register to 03H.

Figure 17-5. Format of Divisor Selection Register 0 (BRGCA0)

Address: FF93H After reset: 03H R/W

Symbol	7	6	5	4	3	2	1	0
BRGCA0	0	0	0	0	0	0	BRGCA01	BRGCA00

BRGCA01	BRGCA00	Selection of base clock (f_W) divisor of CSIA0[Note]					
			f_W = 1 MHz	f_W = 2 MHz	f_W = 2.5 MHz	f_W = 5 MHz	f_W = 10 MHz
0	0	$f_W/6$	166.67 kHz	333.3 kHz	416.67 kHz	833.33 kHz	1.67 MHz
0	1	$f_W/2^3$	125 kHz	250 kHz	312.5 kHz	625 kHz	1.25 MHz
1	0	$f_W/2^4$	62.5 kHz	125 kHz	156.25 kHz	312.5 kHz	625 kHz
1	1	$f_W/2^5$	31.25 kHz	62.5 kHz	78.125 kHz	156.25 kHz	312.5 kHz

Note Set the transfer clock so as to satisfy the following conditions.
- When 2.7 V ≤ V_{DD} < 4.0 V: transfer clock ≤ 833.33 kHz
- When 1.8 V ≤ V_{DD} < 2.7 V: transfer clock ≤ 555.56 kHz

Remark f_W: Base clock frequency selected by CKS00 bit of CSIS0 register
f_{PRS}: Peripheral hardware clock frequency

(5) Automatic data transfer address point specification register 0 (ADTP0)

This is an 8-bit register used to specify the buffer RAM address that ends transfer during automatic data transfer (bit 6 (ATE0) of serial operation mode specification register 0 (CSIMA0) = 1).

This register can be set by an 8-bit memory manipulation instruction. However, during transfer (TSF0 = 1), rewriting ADTP0 is prohibited.

In the 78K0/LF3, 00H to 1FH can be specified because 32 bytes of buffer RAM are incorporated.

Example When ADTP0 is set to 07H
8 bytes of FA00H to FA07H are transferred.

In repeat transfer mode (bit 5 (ATM0) of CSIMA0 = 1), transfer is performed repeatedly up to the address specified with ADTP0.

Example When ADTP0 is set to 07H (repeat transfer mode)
Transfer is repeated as FA00H to FA07H, FA00H to FA07H, … .

Figure 17-6. Format of Automatic Data Transfer Address Point Specification Register 0 (ADTP0)

Address: FF94H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTP0	0	0	0	ADTP04	ADTP03	ADTP02	ADTP01	ADTP00

Caution Be sure to clear bits 7 to 5 to “0”.

The relationship between transfer end buffer RAM address values and ADTP0 setting values is shown below.

Table 17-2. Relationship Between Transfer End Buffer RAM Address Values and ADTP0 Setting Values

Transfer End Buffer RAM Address Value	ADTP0 Setting Value
FAxxH	xxH

Remark xx: 00 to 1F

(6) Automatic data transfer interval specification register 0 (ADTI0)

This is an 8-bit register used to specify the interval time for byte data transfer during automatic data transfer (bit 6 (ATE0) of serial operation mode specification register 0 (CSIMA0) = 1).

Set this register when in master mode (bit 4 (MASTER0) of CSIMA0 = 1) (setting is unnecessary in slave mode). Setting in 1-byte communication mode (bit 6 (ATE0) of CSIMA0 = 0) is also valid. When the interval time specified by ADTI0 after the end of 1-byte communication has elapsed, an interrupt request signal (INTACSI) is output. The number of clocks for the interval can be set to between 0 and 63 clocks.

This register can be set by an 8-bit memory manipulation instruction. However, when bit 0 (TSF0) of serial status register 0 (CSIS0) is 1, rewriting ADTI0 is prohibited.

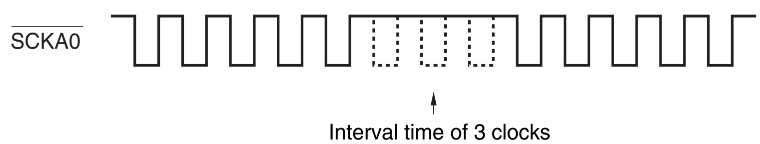
Figure 17-7. Format of Automatic Data Transfer Interval Specification Register 0 (ADTI0)

Address: FF95H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTI0	0	0	ADTI05	ADTI04	ADTI03	ADTI02	ADTI01	ADTI00

The specified interval time is the serial clock (specified by divisor selection register 0 (BRGCA0)) multiplied by an integer value.

Example When ADTI0 = 03H



(7) Automatic data transfer address count register 0 (ADTC0)

This is a register used to indicate buffer RAM addresses during automatic transfer. When automatic transfer is stopped, the data position when transfer stopped can be ascertained by reading ADTC0 register value.

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

Reset signal generation clears this register to 00H. However, reading from ADTC0 is prohibited when bit 0 (TSF0) of serial status register 0 (CSIS0) = 1.

Figure 17-8. Format of Automatic Data Transfer Address Count Register 0 (ADTC0)

Address: FF97H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADTC0	0	0	0	ADTC04	ADTC03	ADTC02	ADTC01	ADTP00

(8) Port function register 1 (PF1)

This register sets the pin functions of P16/SOA0/TxD6 pin.

PF1 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PF1 to 00H.

Figure 17-9. Format of Port Function Register 1 (PF1)

Address: FF20H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PF1	0	PF16	0	0	PF13	0	0	0

PF16	Port (P16), CSIA0, and UART6 output specification
0	Used as P16 or SOA0
1	Used as TxD6

PF13	Port (P13), CSI10, and UART0 output specification
0	Used as P13 or SO10
1	Used as TxD0

(9) Port mode register 1 (PM1)

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

When using P14/$\overline{\text{SCKA0}}$ pin as the clock output of the serial interface, clear PM14 to 0 and set the output latch of P14 to 1.

When using P16/SOA0 pin as the data output of the serial interface, clear PM16 and the output latches of P16 to 0.

When using P14/$\overline{\text{SCKA0}}$ and P15/SIA0 pins as the clock input, or data input of the serial interface, set PM14 and PM15 to 1. At this time, the output latches of P14 and P15 may be 0 or 1.

PM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 17-10. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

17.4 Operation of Serial Interface CSIA0

Serial interface CSIA0 has the following three modes.

- Operation stop mode
- 3-wire serial I/O mode
- 3-wire serial I/O mode with automatic transmit/receive function

17.4.1 Operation stop mode

Serial communication is not executed in this mode. Therefore, the power consumption can be reduced. In addition, the P14/$\overline{\text{SCKA0}}$, P15/SIA0, and P16/SOA0 pins can be used as ordinary I/O port pins in this mode.

(1) Register used

The operation stop mode is set by serial operation mode specification register 0 (CSIMA0). To set the operation stop mode, clear bit 7 (CSIAE0) of CSIMA0 to 0.

(a) Serial operation mode specification register 0 (CSIMA0)

This is an 8-bit register used to control the serial communication operation.
This register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Address: FF90H After reset: 00H R/W

	<7>	6	5	4	<3>	<2>	1	0
CSIMA0	CSIAE0	ATE0	ATM0	MASTER0	TXEA0	RXEA0	DIR0	0

CSIAE0	Control of CSIA0 operation enable/disable
0	CSIA0 operation disabled (SOA0: Low level, $\overline{\text{SCKA0}}$: High level) and asynchronously resets the internal circuit

17.4.2 3-wire serial I/O mode

The one-byte data transmission/reception is executed in the mode in which bit 6 (ATE0) of serial operation mode specification register 0 (CSIMA0) is cleared to 0.

The 3-wire serial I/O mode is useful for connecting peripheral ICs and display controllers with a clocked serial interface.

In this mode, communication is executed by using three lines: serial clock ($\overline{\text{SCKA0}}$), serial output (SOA0), and serial input (SIA0) lines.

(1) Registers used

- Serial operation mode specification register 0 (CSIMA0)[Note 1]
- Serial status register 0 (CSIS0)[Note 2]
- Divisor selection register 0 (BRGCA0)
- Port mode register 1 (PM1)
- Port register 1 (P1)

Notes 1. Bits 7, 6, and 4 to 1 (CSIAE0, ATE0, MASTER0, TXEA0, RXEA0, and DIR0) are used. Setting of bit 5 (ATM0) is invalid.

2. Only bit 0 (TSF0) and bit 6 (CKS00) are used.

The basic procedure of setting an operation in the 3-wire serial I/O mode is as follows.

<1> Set bit 6 (CKS00) of the CSIS0 register (see **Figure 17-3**)[Note 1].
<2> Set the BRGCA0 register (see **Figure 17-5**)[Note 1].
<3> Set bits 4 to 1 (MASTER0, TXEA0, RXEA0, and DIR0) of the CSIMA0 register (see **Figure 17-2**).
<4> Set bit 7 (CSIAE0) of the CSIMA0 register to 1 and clear bit 6 (ATE0) to 0.
<5> Write data to serial I/O shift register 0 (SIOA0). → Data transmission/reception is started[Note 2].

Notes 1. This register does not have to be set when the slave mode is specified (MASTER0 = 0).

2. Write dummy data to SIOA0 only for reception.

Caution Take relationship with the other party of communication when setting the port mode register and port register.

The relationship between the register settings and pins is shown below.

Table 17-3. Relationship Between Register Settings and Pins

CSIAE0	ATE0	MASTER0	PM15	P15	PM16	P16	PM14	P14	Serial I/O Shift Register 0 Operation	Serial Clock Counter Operation Control	Pin Function		
											SIA0/P15	SOA0/P16	$\overline{\text{SCKA0}}$/P14/INTP4
0	×	×	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	×Note 1	Operation stopped	Clear	P15	P16	P14/INTP4
1	0	0	1Note 2	×Note 2	0Note 3	0Note 3	1	×	Operation enabled	Count operation	SIA0Note 2	SOA0Note 3	$\overline{\text{SCKA0}}$ (input)
		1					0	1					$\overline{\text{SCKA0}}$ (output)

Notes 1. Can be set as port function.

2. Can be used as P15 when only transmission is performed. Clear bit 2 (RXEA0) of CSIMA0 to 0.

3. Can be used as P16 when only reception is performed. Clear bit 3 (TXEA0) of CSIMA0 to 0.

Remark ×: don't care

CSIAE0: Bit 7 of serial operation mode specification register 0 (CSIMA0)

ATE0: Bit 6 of CSIMA0

MASTER0: Bit 4 of CSIMA0

PM1×: Port mode register

P1×: Port output latch

(2) 1-byte transmission/reception communication operation

(a) 1-byte transmission/reception

When bit 7 (CSIAE0) and bit 6 (ATE0) of serial operation mode specification register 0 (CSIMA0) = 1, 0, respectively, if communication data is written to serial I/O shift register 0 (SIOA0), the data is output via the SOA0 pin in synchronization with the $\overline{\text{SCKA0}}$ falling edge, and stored in the SIOA0 register in synchronization with the rising edge 1 clock later.

Data transmission and data reception can be performed simultaneously.

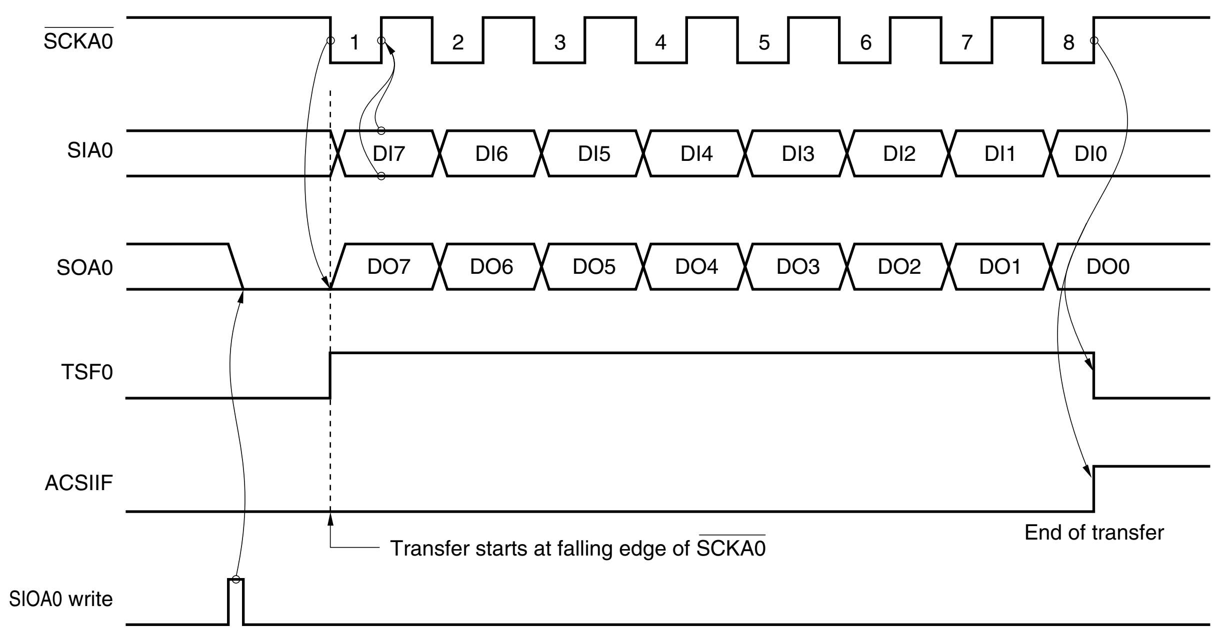
If only reception is to be performed, communication can only be started by writing a dummy value to the SIOA0 register.

When communication of 1 byte is complete, an interrupt request signal (INTACSI) is generated.

In 1-byte transmission/reception, the setting of bit 5 (ATM0) of CSIMA0 is invalid.

Be sure to read data after confirming that bit 0 (TSF0) of serial status register 0 (CSIS0) = 0.

Figure 17-11. 3-Wire Serial I/O Mode Timing



Caution The SOA0 pin becomes low level by an SIOA0 write.

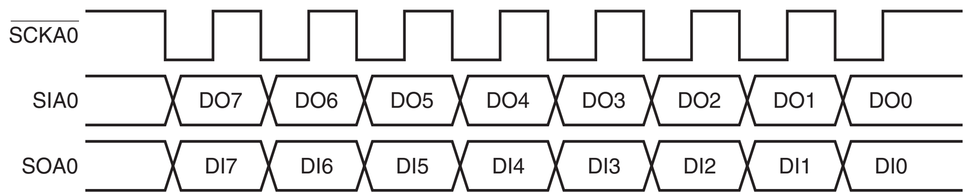
(b) Data format

In the data format, data is changed in synchronization with the $\overline{SCKA0}$ falling edge as shown below.

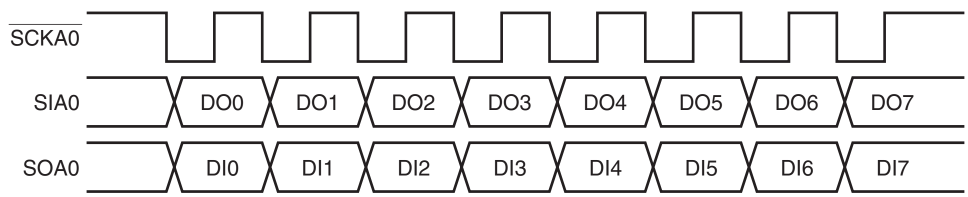
The data length is fixed to 8 bits and the data communication direction can be switched by the specification of bit 1 (DIR0) of serial operation mode specification register 0 (CSIMA0).

Figure 17-12. Format of Transmit/Receive Data

(a) MSB-first (DIR0 bit = 0)



(b) LSB-first (DIR0 bit = 1)

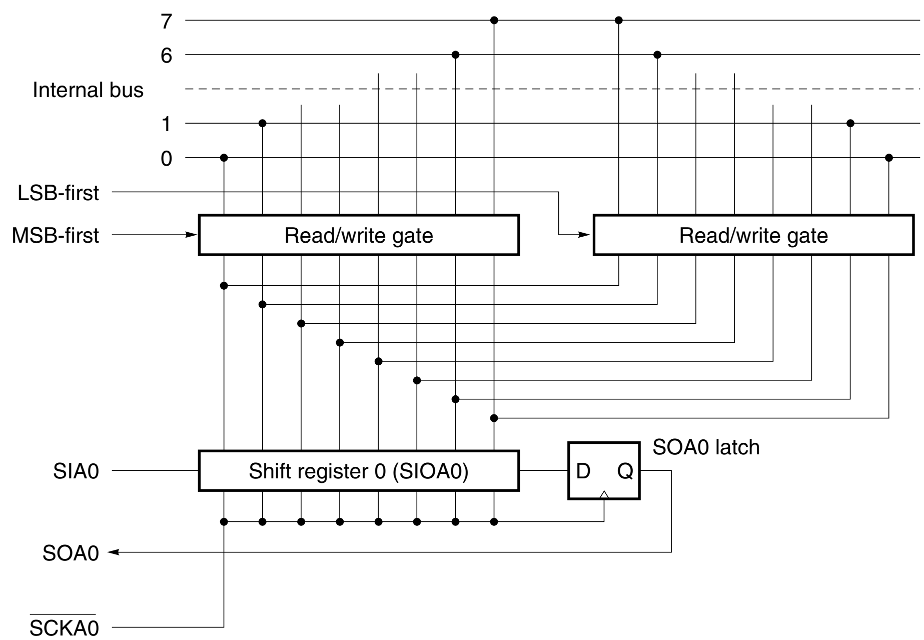


(c) Switching MSB/LSB as start bit

Figure 17-13 shows the configuration of serial I/O shift register 0 (SIOA0) and the internal bus. As shown in the figure, MSB/LSB can be read/written in reverse form.

Switching MSB/LSB as the start bit can be specified using bit 1 (DIR0) of serial operation mode specification register 0 (CSIMA0).

Figure 17-13. Transfer Bit Order Switching Circuit



Start bit switching is realized by switching the bit order for data written to SIOA0. The SIOA0 shift order remains unchanged.

Thus, switching between MSB-first and LSB-first must be performed before writing data to the shift register.

(d) Communication start

Serial communication is started by setting communication data to serial I/O shift register 0 (SIOA0) when the following two conditions are satisfied.

- Serial interface CSIA0 operation control bit (CSIAE0) = 1
- Serial communication is not in progress

Caution If CSIAE0 is set to 1 after data is written to SIOA0, communication does not start.

Upon termination of 8-bit communication, serial communication automatically stops and the interrupt request flag (ACSIIF) is set.

17.4.3 3-wire serial I/O mode with automatic transmit/receive function

Up to 32 bytes of data can be transmitted/received without using software in the mode in which bit 6 (ATE0) of serial operation mode specification register 0 (CSIMA0) is set to 1. After communication is started, only data of the set number of bytes stored in RAM in advance can be transmitted, and only data of the set number of bytes can be received and stored in RAM.

(1) Registers used

- Serial operation mode specification register 0 (CSIMA0)
- Serial status register 0 (CSIS0)
- Serial trigger register 0 (CSIT0)
- Divisor selection register 0 (BRGCA0)
- Automatic data transfer address point specification register 0 (ADTP0)
- Automatic data transfer interval specification register 0 (ADTI0)
- Port mode register 1 (PM1)
- Port register 1 (P1)

The relationship between the register settings and pins is shown below.

Caution A wait state may be generated when data is written to the buffer RAM. For details, see CHAPTER 34 CAUTIONS FOR WAIT.

The relationship between the register settings and pins is shown below.

Table 17-4. Relationship Between Register Settings and Pins

CSIAE0	ATE0	MASTER0	PM15	P15	PM16	P16	PM14	P14	Serial I/O Shift Register 0 Operation	Serial Clock Counter Operation Control	Pin Function		
											SIA0/P15	SOA0/P16	$\overline{SCKA0}$/P14 /INTP4
0	×	×	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	Operation stopped	Clear	P15	P16	P14/INTP4
1	0	0	1[Note 2]	×[Note 2]	0[Note 3]	0[Note 3]	1	×	Operation enabled	Count operation	SIA0[Note 2]	SOA0[Note 3]	$\overline{SCKA0}$ (input)
		1					0	1					$\overline{SCKA0}$ (output)

Notes 1. Can be set as port function.

2. Can be used as P15 when only transmission is performed. Clear bit 2 (RXEA0) of CSIMA0 to 0.

3. Can be used as P16 when only reception is performed. Clear bit 3 (TXEA0) of CSIMA0 to 0.

Remark
×: don't care
CSIAE0: Bit 7 of serial operation mode specification register 0 (CSIMA0)
ATE0: Bit 6 of CSIMA0
MASTER0: Bit 4 of CSIMA0
PM1×: Port mode register
P1×: Port output latch

(2) Automatic transmit/receive data setting

Here is an example of the procedure for successively transmitting/receiving data as the master.

<1> Enable CSIA0 to operate by setting bit 7 (CSIAE0) of serial operation mode specification register 0 (CSIMA0) to 1 (the buffer RAM can now be accessed).
<2> Select a serial clock by using serial status register 0 (CSIS0).
<3> Set the division ratio of the serial clock by using division value selection register 0 (BRGCA0), and specify a communication rate.
<4> Sequentially write data to be transmitted to the buffer RAM, starting from the least significant address FA00H, up to FA1FH. Data is transmitted from the lowest address, continuing on to higher addresses.
<5> Set "number of data items to be transmitted − 1" to automatic data transfer address point specification register 0 (ADTP0).
<6> Set bits 6 (ATE0) and 4 (MASTER0) of CSIMA0 to select a master operation in the automatic communication mode.
<7> Set bits 3 (TXEA0) and 2 (RXEA0) of CSIMA0 to 1 to enable transmission/reception.
<8> Set the transmission interval of data to the automatic data transfer interval specification register (ADTI0).
<9> Automatic transmit/receive processing is started when bit 0 (ATSTA0) of serial trigger register 0 (CSIT0) is set to 1.

Caution Take the relationship with the other communicating party into consideration when setting the port mode register and port register.

Operations <1> to <9> execute the following operation.

- After the buffer RAM data indicated by automatic data transfer address count register 0 (ADTC0) is transferred to SIOA0, transmission is carried out (start of automatic transmission/reception).
- The received data is written to the buffer RAM address indicated by ADTC0.
- ADTC0 is incremented and the next data transmission/reception is carried out. Data transmission/reception continues until the ADTC0 incremental output matches the set value of automatic data transfer address point specification register 0 (ADTP0) (end of automatic transmission/reception). However, if bit 5 (ATM0) of CSIMA0 is set to 1 (repeat mode), ADTC0 is cleared after a match between ADTP0 and ADTC0, and then repeated transmission/reception is started.
- When automatic transmission/reception is terminated, an interrupt request (INTACSI) is generated and bit 0 (TSF0) of CSIS0 is cleared.
- To continue transmitting the next data, set the new data to the buffer RAM, and set "number of data to be transmitted − 1" to ADTP0. After setting the number of data, set ATSTA0 to 1.

(3) Automatic transmission/reception communication operation

(a) Automatic transmission/reception mode

Automatic transmission/reception can be performed using buffer RAM.

The data stored in the buffer RAM is output from the SOA0 pin via the SIOA0 register in synchronization with the $\overline{\text{SCKA0}}$ falling edge by performing **(2) Automatic transmit/receive data setting**.

The receive data is stored in the buffer RAM via the SIOA0 register in synchronization with the $\overline{\text{SCKA0}}$ rising edge.

Data transfer ends if bit 0 (TSF0) of serial status register 0 (CSIS0) is set to 1 when any of the following conditions is met.

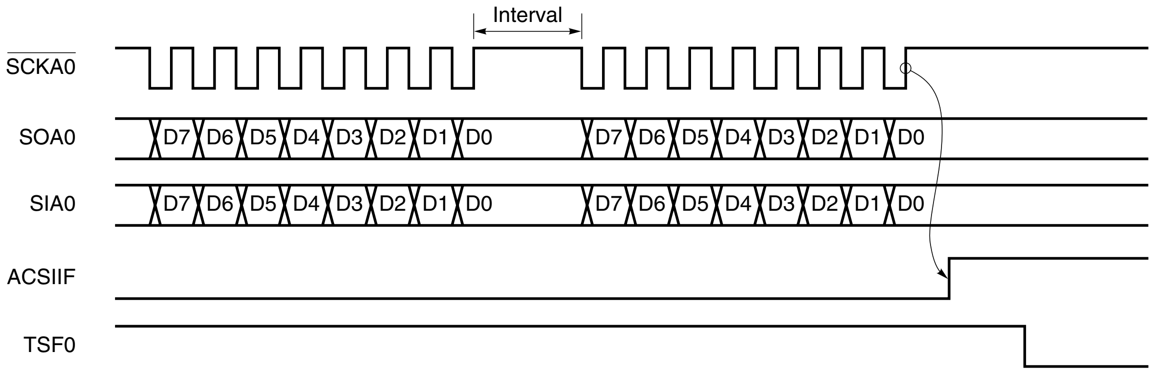
- Communication stop: Reset by clearing bit 7 (CSIAE0) of the CSIMA0 register to 0
- Communication suspension: Transfer of 1 byte is complete by setting bit 1 (ATSTP0) of the CSIT0 register to 1
- Transfer of the range specified by the ADTP0 register is complete

At this time, an interrupt request signal (INTACSI) is generated except when the CSIAE0 bit = 0.

If a transfer is terminated in the middle, transfer starting from the remaining data is not possible. Read automatic data transfer address count register 0 (ADTC0) to confirm how much of the data has already been transferred and re-execute transfer by performing **(2) Automatic transmit/receive data setting**.

Figure 17-14 shows the example of the operation timing in automatic transmission/reception mode and Figure 17-15 shows the operation flowchart. Figures 17-16 and 17-17 show the operation of internal buffer RAM when 6 bytes of data are transmitted/received.

Figure 17-14. Example of Automatic Transmission/Reception Mode Operation Timings



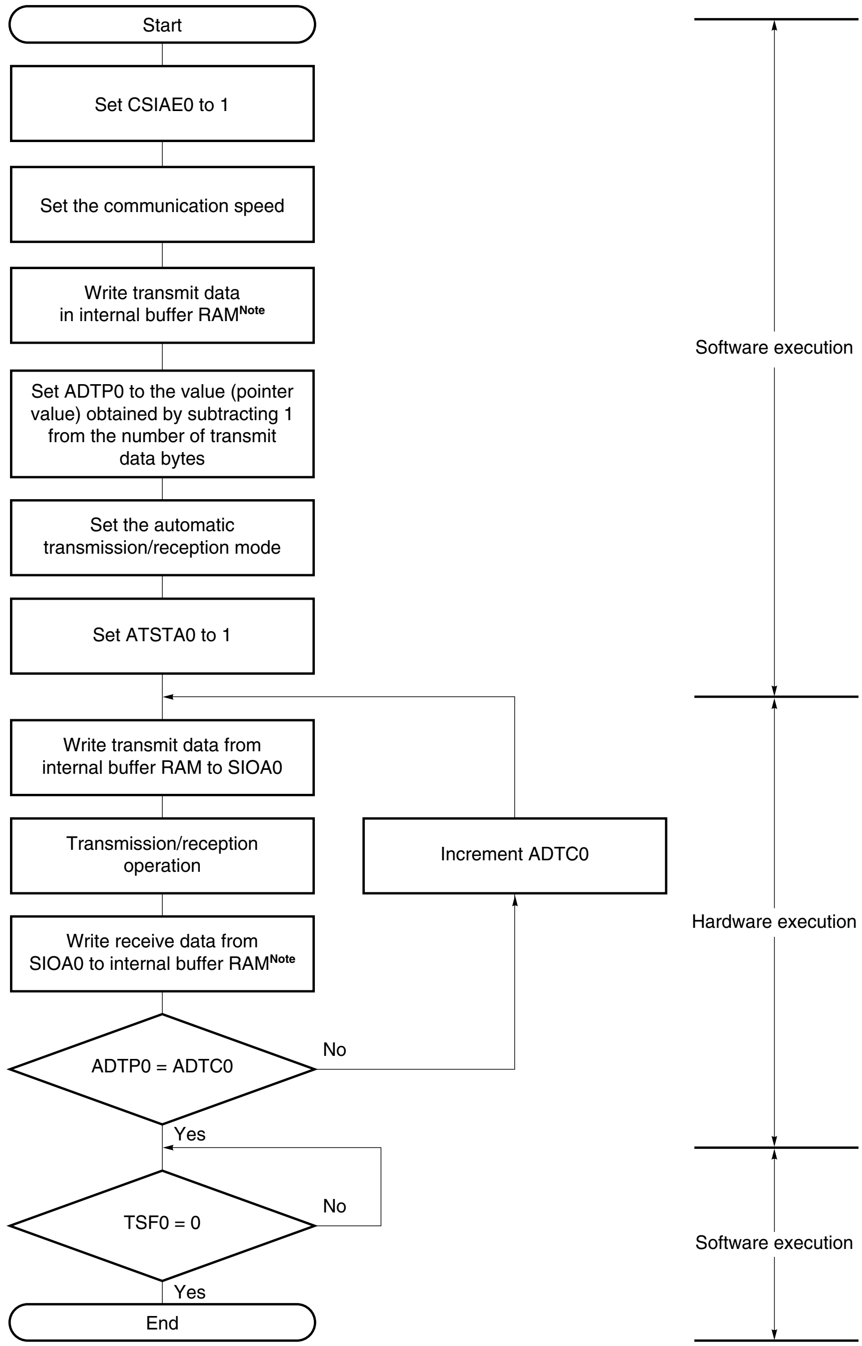
Cautions 1. Because, in the automatic transmission/reception mode, the automatic transmit/receive function writes/reads data to/from the internal buffer RAM after 1-byte transmission/reception, an interval is inserted until the next transmission/reception. As the buffer RAM write/read is performed at the same time as CPU processing, the interval is dependent upon the set value of automatic data transfer interval specification register 0 (ADTI0).

2. If an access to the buffer RAM by the CPU conflicts with an access to the buffer RAM by serial interface CSIA0 during the interval period, the interval time specified by automatic data transfer interval specification register 0 (ADTI0) may be extended.

Remark ACSIIF: Interrupt request flag

TSF0: Bit 0 of serial status register 0 (CSIS0)

Figure 17-15. Automatic Transmission/Reception Mode Flowchart



CSIAE0:	Bit 7 of serial operation mode specification register 0 (CSIMA0)
ADTP0:	Automatic data transfer address point specification register 0
ADTI0:	Automatic data transfer interval specification register 0
ATSTA0:	Bit 0 of serial trigger register 0 (CSIT0)
SIOA0:	Serial I/O shift register 0
ADTC0:	Automatic data transfer address count register 0
TSF0:	Bit 0 of serial status register 0 (CSIS0)

Note A wait state may be generated when data is written to the buffer RAM. For details, see **CHAPTER 34 CAUTIONS FOR WAIT**.

In 6-byte transmission/reception (ATM0 = 0, RXEA0 = 1, TXEA0 = 1, ATE0 = 1) in automatic transmission/reception mode, internal buffer RAM operates as follows.

(i) Starting automatic transmission/reception (see Figure 17-16)

<1> When bit 0 (ATSTA0) of serial trigger register 0 (CSIT0) is set to 1, transmit data 1 (T1) is transferred from the internal buffer RAM to SIOA0 and transmission/reception is started.

<2> When transmission of the first byte is completed, the receive data 1 (R1) is transferred from SIOA0 to the buffer RAM, and automatic data transfer address count register 0 (ADTC0) is incremented.

<3> Next, transmit data 2 (T2) is transferred from the internal buffer to SIOA0.

Figure 17-16. Internal Buffer RAM Operation in Automatic Transmission/Reception Mode (Starting Transmission/Reception) (1/2)

<1> Starting 1st byte transmission/reception

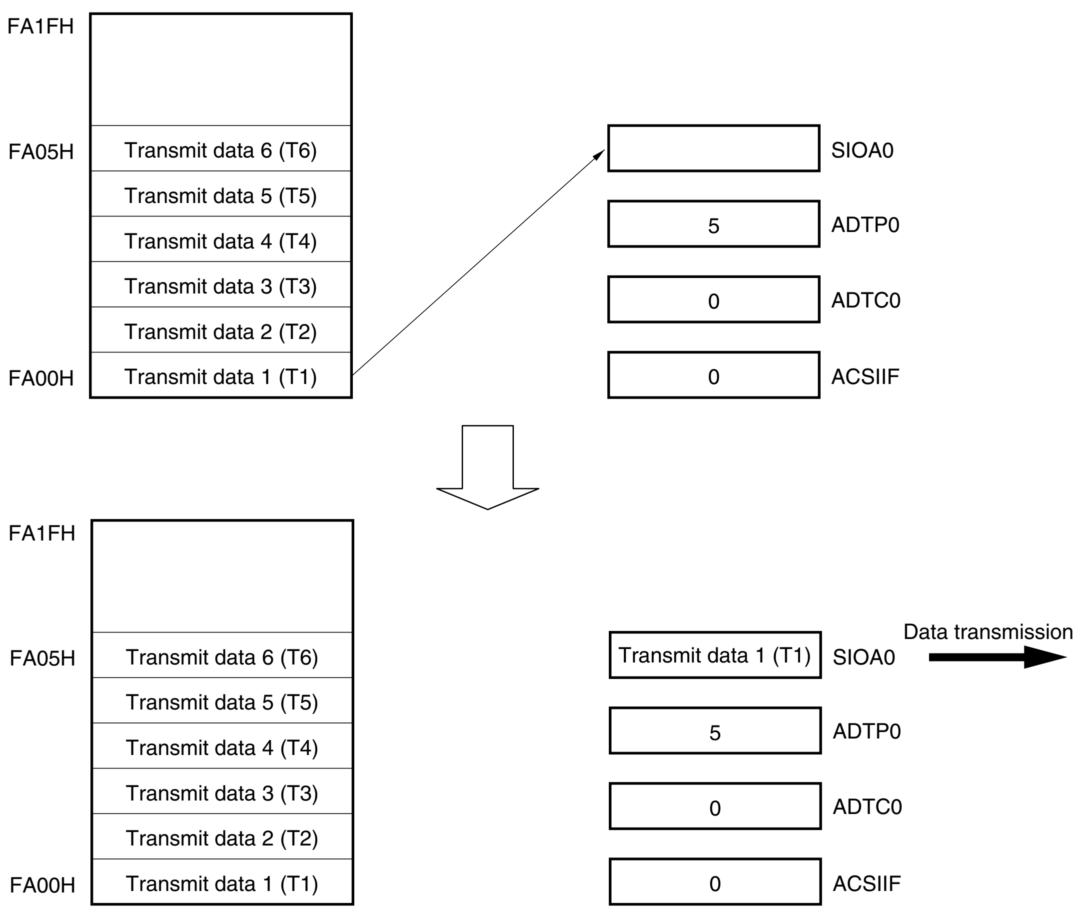
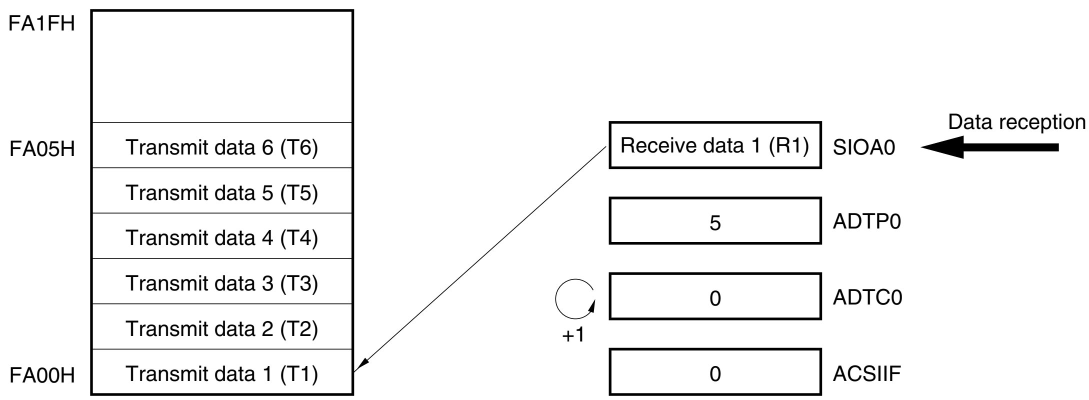
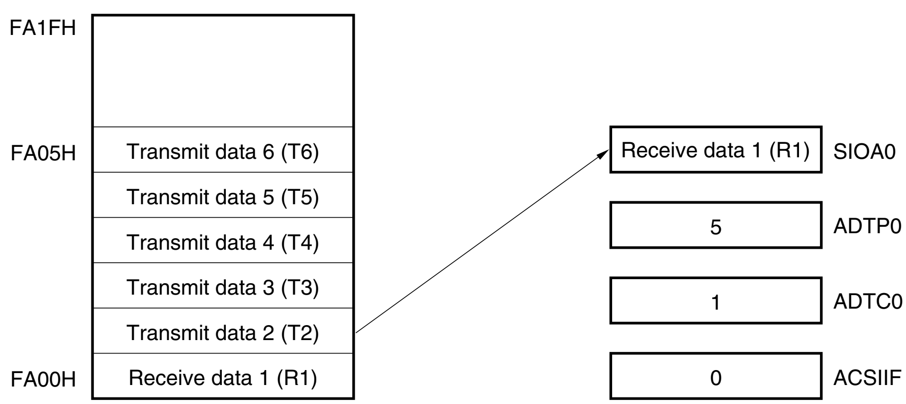


Figure 17-16. Internal Buffer RAM Operation in Automatic Transmission/Reception Mode (Starting Transmission/Reception) (2/2)

<2> End of 1st byte transmission/reception



<3> Starting of 2nd byte transmission/reception



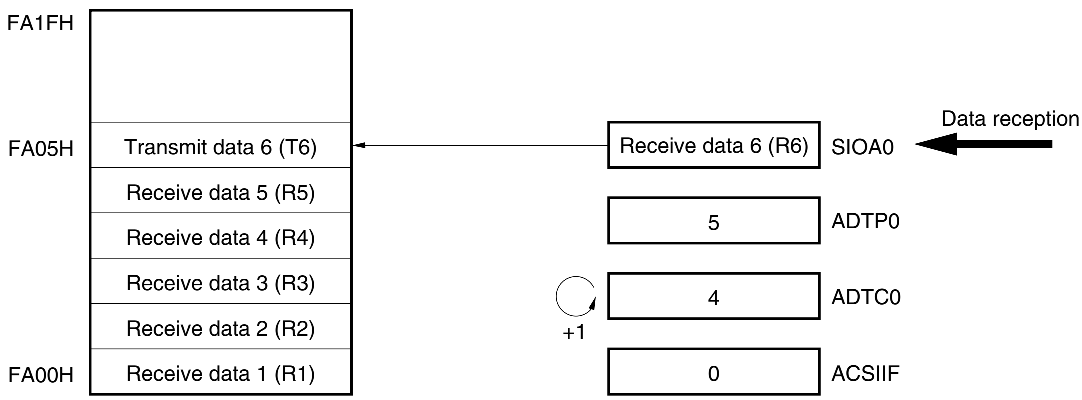
(ii) Completion of transmission/reception (see Figure 17-17)

<1> When transmission/reception of the sixth byte is completed, receive data 6 (R6) is transferred from SIOA0 to the internal buffer RAM and ADTC0 is incremented.

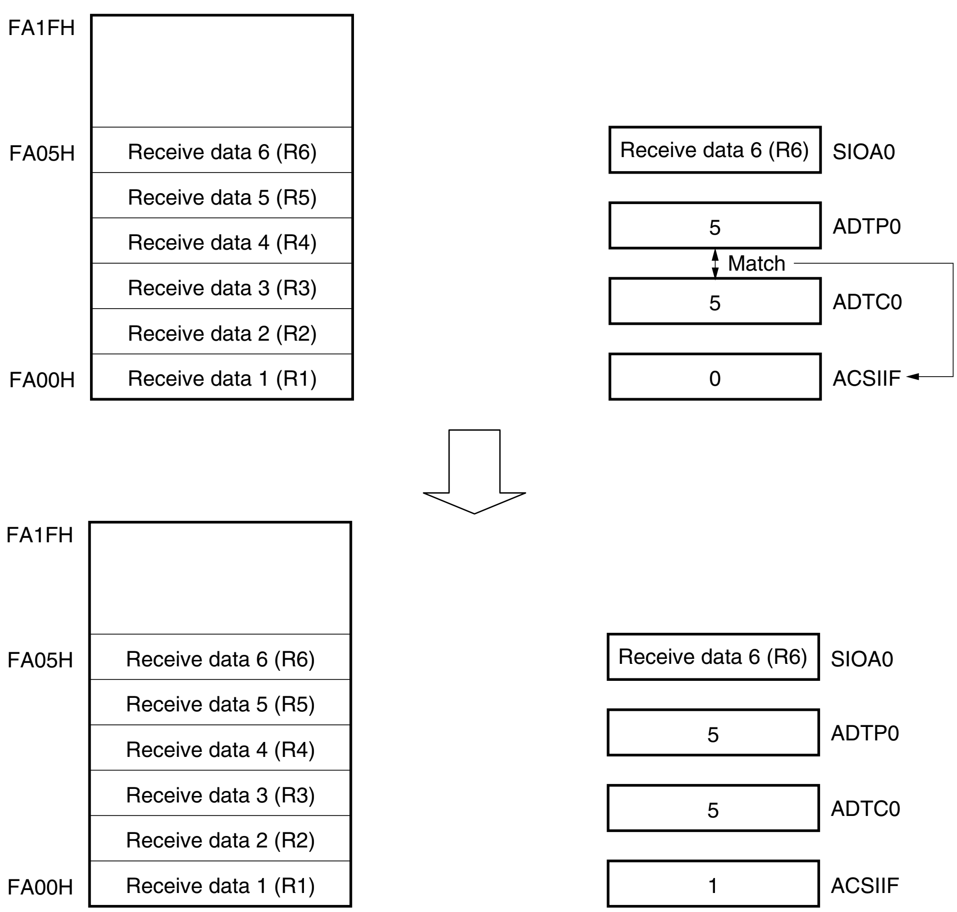
<2> When the value of ADPT0 and that of ADTC0 match, the automatic transmission/reception ends, and an interrupt request flag (ACSIIF) is set (INTACSI is generated). ADTC0 and bit 0 (TSF0) of serial status register 0 (CSIS0) are cleared to 0.

Figure 17-17. Internal Buffer RAM Operation in Automatic Transmission/Reception Mode (End of Transmission/Reception)

<1> End of 6th byte transmission/reception



<2> End of automatic transmission/reception



(b) Automatic transmission mode

In this mode, the specified data is transmitted in 8-bit unit.

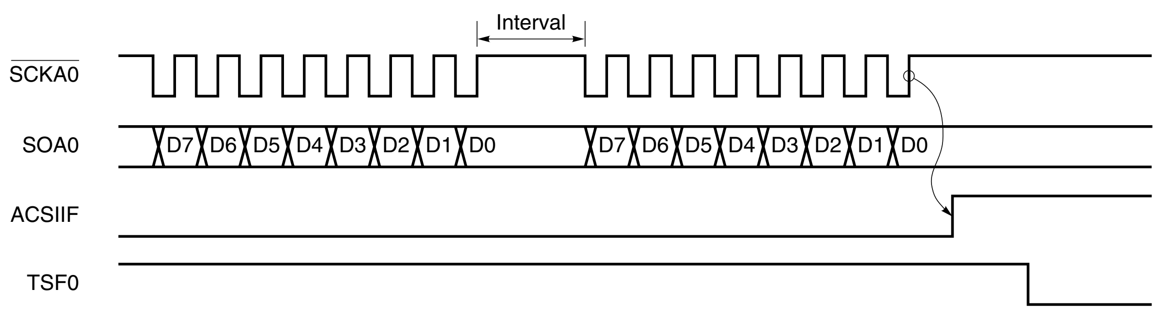
Serial communication is started when bit 0 (ATSTA0) of serial trigger register 0 (CSIT0) is set to 1 while bit 7 (CSIAE0), bit 6 (ATE0), and bit 3 (TXEA0) of serial operation mode specification register 0 (CSIMA0) are set to 1.

When the final byte has been transmitted, an interrupt request flag (ACSIIF) is set. The termination of automatic transmission can also be judged by bit 0 (TSF0) of serial status register 0 (CSIS0).

If a receive operation, busy control and strobe control are not executed, the SIA0/P15 pin can be used as normal I/O port pins.

Figure 17-18 shows the example of the automatic transmission mode operation timing, and Figure 17-19 shows the operation flowchart.

Figure 17-18. Example of Automatic Transmission Mode Operation Timing



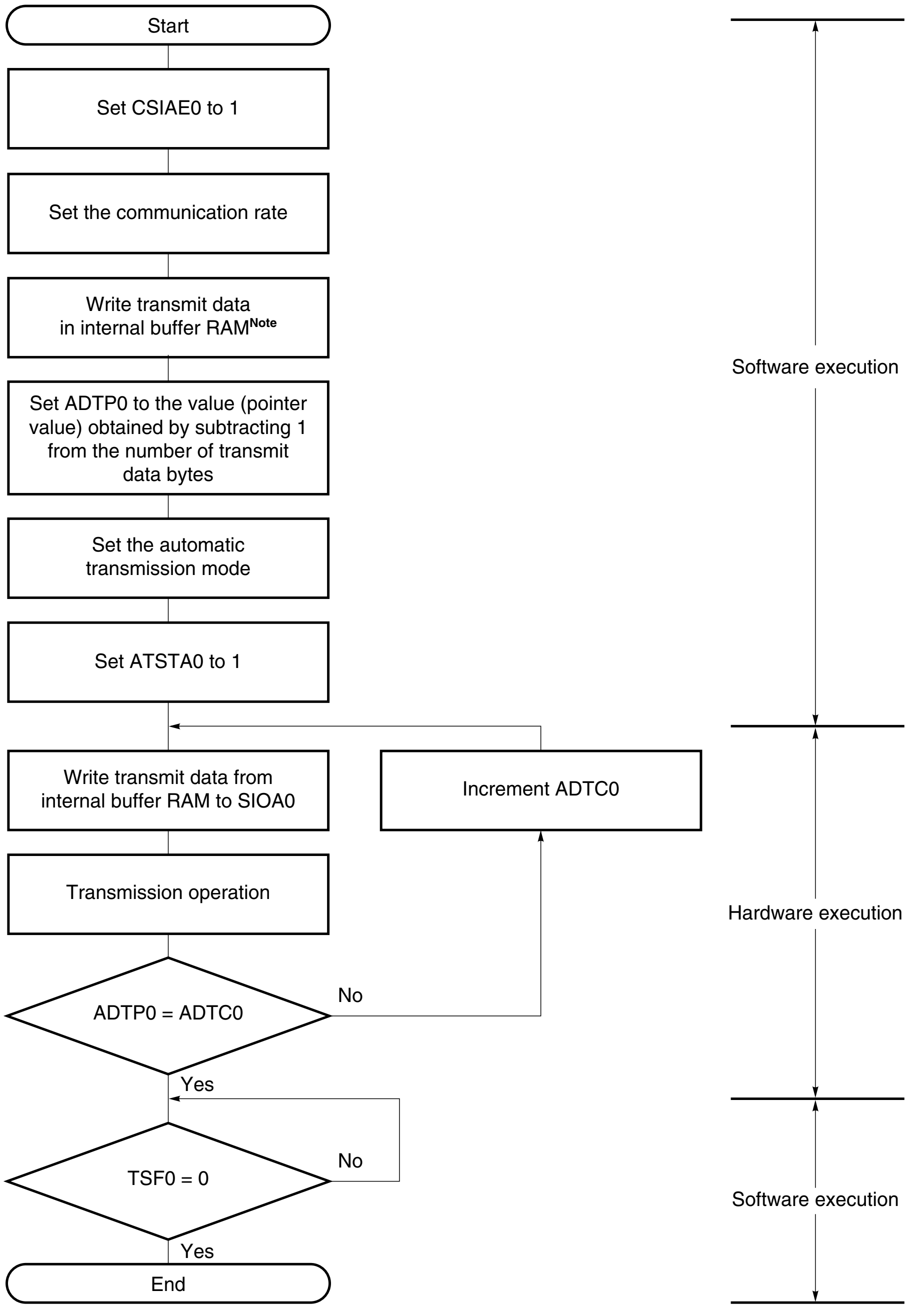
Cautions 1. Because, in the automatic transmission mode, the automatic transmit/receive function reads data from the internal buffer RAM after 1-byte transmission, an interval is inserted until the next transmission. As the buffer RAM read is performed at the same time as CPU processing, the interval is dependent upon the set value of automatic data transfer interval specification register 0 (ADTI0).

2. If an access to the buffer RAM by the CPU conflicts with an access to the buffer RAM by serial interface CSIA0 during the interval period, the interval time specified by automatic data transfer interval specification register 0 (ADTI0) may be extended.

Remark ACSIIF: Interrupt request flag

TSF0: Bit 0 of serial status register 0 (CSIS0)

Figure 17-19. Automatic Transmission Mode Flowchart



CSIAE0: Bit 7 of serial operation mode specification register 0 (CSIMA0)
ADTP0: Automatic data transfer address point specification register 0
ADTI0: Automatic data transfer interval specification register 0
ATSTA0: Bit 0 of serial trigger register 0 (CSIT0)
SIOA0: Serial I/O shift register 0
ADTC0: Automatic data transfer address count register 0
TSF0: Bit 0 of serial status register 0 (CSIS0)

Note A wait state may be generated when data is written to the buffer RAM. For details, see **CHAPTER 34 CAUTIONS FOR WAIT**.

(c) Repeat transmission mode

In this mode, data stored in the internal buffer RAM is transmitted repeatedly.

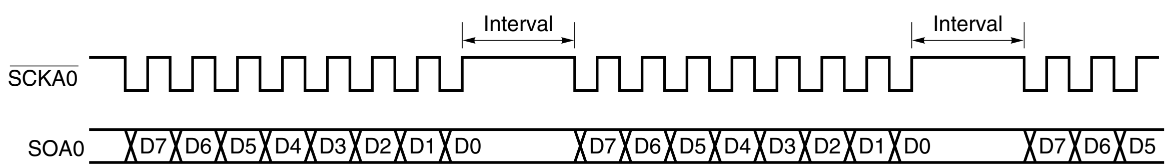
Serial communication is started when bit 0 (ATSTA0) of serial trigger register 0 (CSIT0) is set to 1 while bit 7 (CSIAE0), bit 6 (ATE0), bit 5 (ATM0), and bit 3 (TXEA0) of serial operation mode specification register 0 (CSIMA0) are set to 1.

Unlike the automatic transmission mode, after the number of setting bytes has been transmitted, the interrupt request flag (ACSIIF) is not set, automatic data transfer address count register 0 (ADTC0) is reset to 0, and the internal buffer RAM contents are transmitted again.

When a reception operation, busy control and strobe control are not performed, the SIA0/P15 pin can be used as ordinary I/O port pins.

The example of the repeat transmission mode operation timing is shown in Figure 17-20, and the operation flowchart in Figure 17-21.

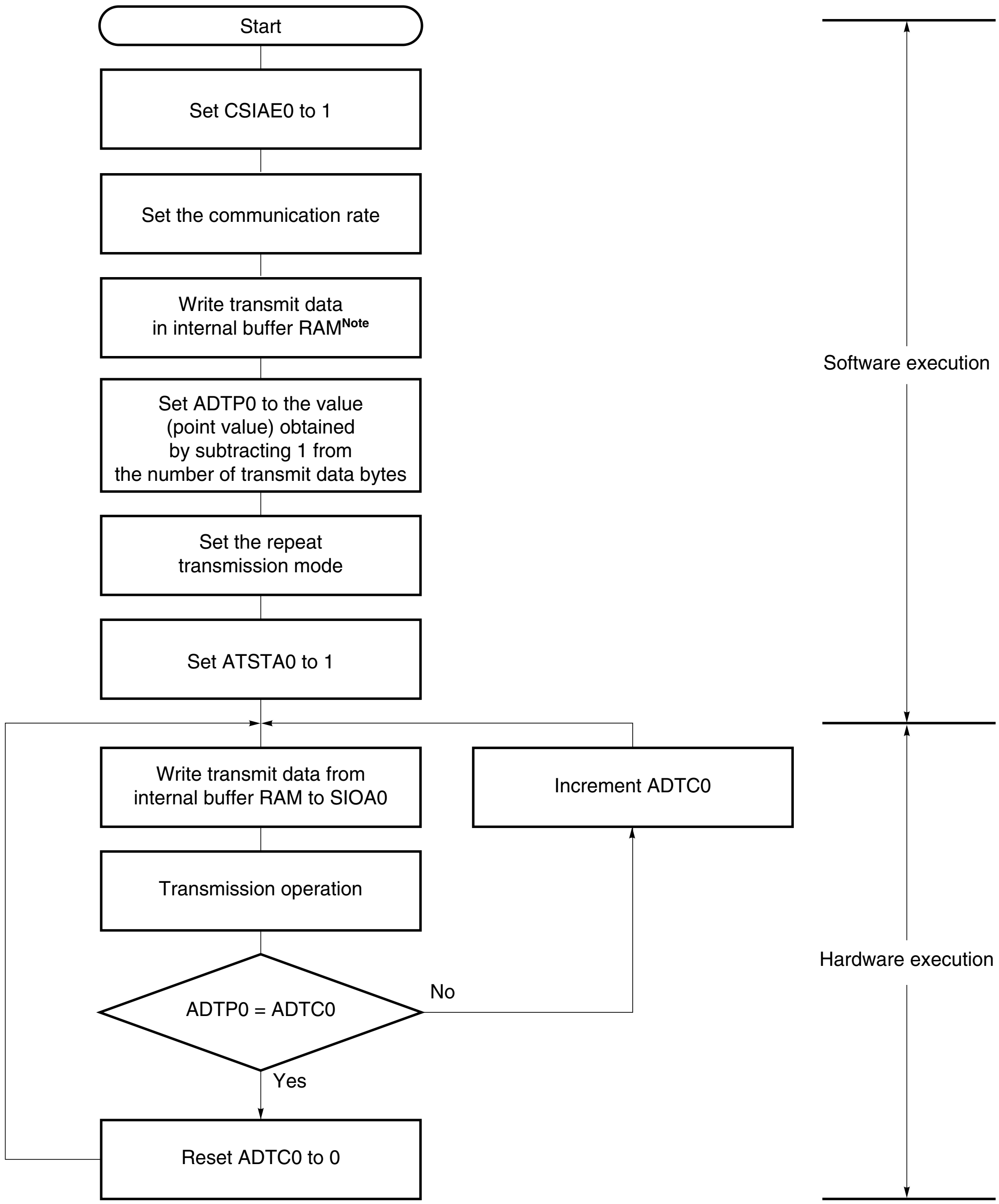
Figure 17-20. Example of Repeat Transmission Mode Operation Timing



Cautions 1. Because, in the repeat transmission mode, a read is performed on the buffer RAM after the transmission of one byte, the interval is included in the period up to the next transmission. As the buffer RAM read is performed at the same time as CPU processing, the interval is dependent upon the set value of automatic data transfer interval specification register 0 (ADTI0).

2. If an access to the buffer RAM by the CPU conflicts with an access to the buffer RAM by serial interface CSIA0 during the interval period, the interval time specified by automatic data transfer interval specification register 0 (ADTI0) may be extended.

Figure 17-21. Repeat Transmission Mode Flowchart



CSIAE0: Bit 7 of serial operation mode specification register 0 (CSIMA0)
ADTP0: Automatic data transfer address point specification register 0
ADTI0: Automatic data transfer interval specification register 0
ATSTA0: Bit 0 of serial trigger register 0 (CSIT0)
SIOA0: Serial I/O shift register 0
ADTC0: Automatic data transfer address count register 0

Note A wait state may be generated when data is written to the buffer RAM. For details, see **CHAPTER 34 CAUTIONS FOR WAIT**.

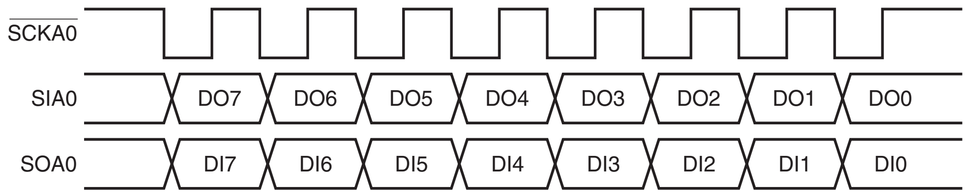
(d) Data format

Data is changed in synchronization with the $\overline{\text{SCKA0}}$ falling edge as shown below.

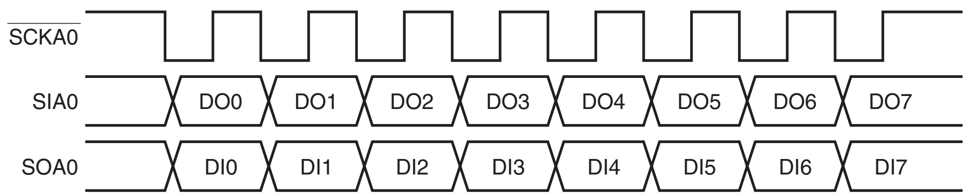
The data length is fixed to 8 bits and the data transfer direction can be switched by the specification of bit 1 (DIR0) of serial operation mode specification register 0 (CSIMA0).

Figure 17-22. Format of CSIA0 Transmit/Receive Data

(a) MSB-first (DIR0 bit = 0)



(b) LSB-first (DIR0 bit = 1)



(e) Automatic transmission/reception suspension and restart

Automatic transmission/reception can be temporarily suspended by setting bit 1 (ATSTP0) of serial trigger register 0 (CSIT0) to 1.

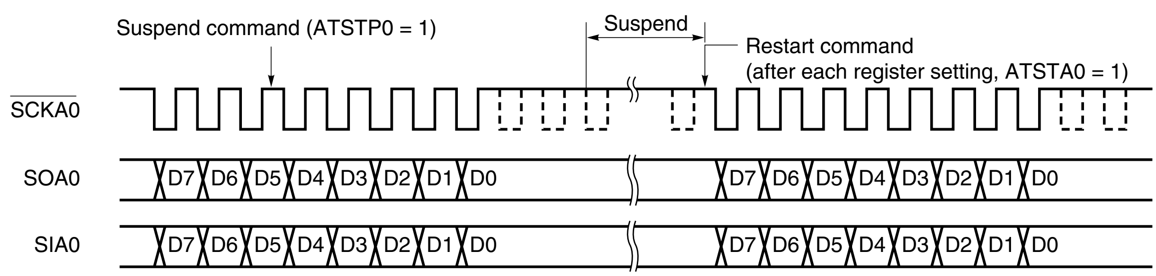
During 8-bit data communication, the transmission/reception is not suspended. It is suspended upon completion of 8-bit data communication.

When suspended, bit 0 (TSF0) of serial status register 0 (CSIS0) is cleared to 0 after transfer of the 8th bit.

Cautions 1. If the HALT instruction is executed during automatic transmission/reception, communication is suspended and the HALT mode is set if during 8-bit data communication. When the HALT mode is cleared, automatic transmission/reception is restarted from the suspended point.

2. When suspending automatic transmission/reception, do not change the operating mode to 3-wire serial I/O mode while TSF0 = 1.

Figure 17-23. Automatic Transmission/Reception Suspension and Restart



ATSTP0: Bit 1 of serial trigger register 0 (CSIT0)

ATSTA0: Bit 0 of CSIT0

<R>

CHAPTER 18 LCD CONTROLLER/DRIVER

18.1 Functions of LCD Controller/Driver

The functions of the LCD controller/driver in the 78K0/LF3 are as follows.

(1) The LCD driver voltage generator can switch external resistance division and internal resistance division.

(2) Automatic output of segment and common signals based on automatic display data memory read

(3) Six different display modes:
- Static
- 1/2 duty (1/2 bias)
- 1/3 duty (1/2 bias)
- 1/3 duty (1/3 bias)
- 1/4 duty (1/3 bias)
- 1/8 duty (1/4 bias)

(4) Six different frame frequencies, selectable in each display mode

(5) μPD78F047x: Segment signal outputs: 40[Note] (SEG0 to SEG39),
Common signal outputs: 8[Note] (COM0 to COM7)
μPD78F048x: Segment signal outputs: 40[Note] (SEG0 to SEG39),
Common signal outputs: 8[Note] (COM0 to COM7)
μPD78F049x: Segment signal outputs: 32[Note] (SEG0 to SEG31),
Common signal outputs: 8[Note] (COM0 to COM7)

(6) Output of LCD segment signals and time division output of segment key source signals in each display mode (except static mode)
Segment key source signal outputs: Max. 8 (SEG24 (KS0) to SEG31 (KS7))

Note The four segment signal outputs (SEG0 to SEG3) and four common signal outputs (COM4 to COM7) are alternate-function pins. COM4 to COM7 can be used only when eight-time-slice mode is selected by the setting of the LCD display mode register (LCDM).

Table 18-1 lists the maximum number of pixels that can be displayed in each display mode.

Table 18-1. Maximum Number of Pixels

(a) μPD78F047x, 78F048x

LCD Driver Voltage Generator	Bias Mode	Number of Time Slices	Common Signals Used	Number of Segments	Maximum Number of Pixels
• External resistance division • Internal resistance division	–	Static	COM0 (COM1 to COM3)	40	40 (40 segment signals, 1 common signal)[Note 2]
	1/2	2 [Note 1]	COM0, COM1		80 (40 segment signals, 2 common signals)[Note 3]
		3 [Note 1]	COM0 to COM2		120 (40 segment signals, 3 common signals)[Note 4]
	1/3	3 [Note 1]	COM0 to COM2		
		4 [Note 1]	COM0 to COM3		160 (40 segment signals, 4 common signals)[Note 5]
	1/4	8 [Note 1]	COM0 to COM7	36	288 (36 segment signals, 8 common signals)[Note 6]

Notes 1. When using the segment key scan function (KSON = 1), “number of time slices + 1” is added for segment key scan signal output.

2. 5-digit LCD panel, each digit having an 8-segment 8. configuration.

3. 10-digit LCD panel, each digit having a 4-segment 8. configuration.

4. 15-digit LCD panel, each digit having a 3-segment 8. configuration.

5. 20-digit LCD panel, each digit having a 2-segment 8. configuration.

6. 36-digit LCD panel, each digit having a 1-segment 8. configuration.

(b) μPD78F049x

LCD Driver Voltage Generator	Bias Mode	Number of Time Slices	Common Signals Used	Number of Segments	Maximum Number of Pixels
• External resistance division • Internal resistance division	–	Static	COM0 (COM1 to COM3)	32	32 (32 segment signals, 1 common signal)[Note 2]
	1/2	2 [Note 1]	COM0, COM1		64 (32 segment signals, 2 common signals)[Note 3]
		3 [Note 1]	COM0 to COM2		96 (32 segment signals, 3 common signals)[Note 4]
	1/3	3 [Note 1]	COM0 to COM2		
		4 [Note 1]	COM0 to COM3		128 (32 segment signals, 4 common signals)[Note 5]
	1/4	8 [Note 1]	COM0 to COM7	28	224 (28 segment signals, 8 common signals)[Note 6]

Notes 1. When using the segment key scan function (KSON = 1), “number of time slices + 1” is added for segment key scan signal output.

2. 4-digit LCD panel, each digit having an 8-segment 8. configuration.

3. 8-digit LCD panel, each digit having a 4-segment 8. configuration.

4. 12-digit LCD panel, each digit having a 3-segment 8. configuration.

5. 16-digit LCD panel, each digit having a 2-segment 8. configuration.

6. 28-digit LCD panel, each digit having a 1-segment 8. configuration.

18.2 Configuration of LCD Controller/Driver

The LCD controller/driver consists of the following hardware.

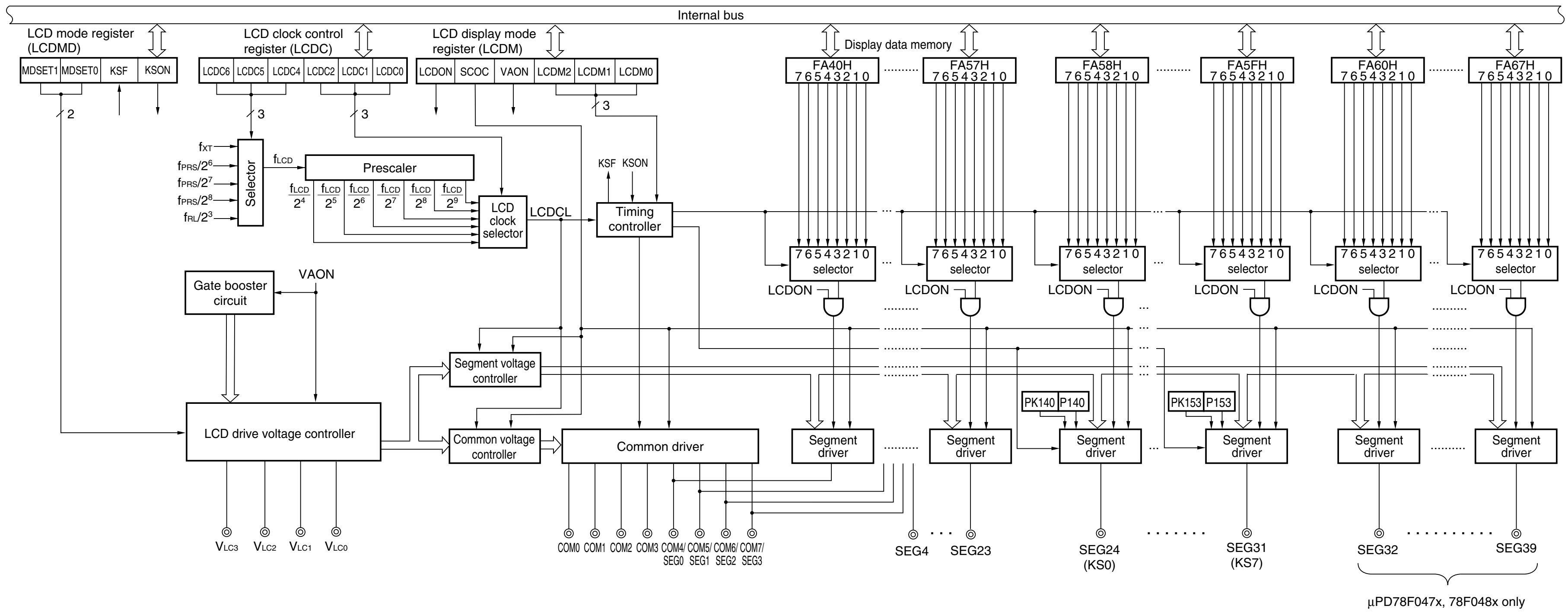
Table 18-2. Configuration of LCD Controller/Driver

Item	Configuration
Display outputs	μPD78F047x: 40 segment signals[Note 1] (SEG0-SEG39), 8 common signals[Note 1] (COM0 to COM7) μPD78F048x: 40 segment signals[Note 1] (SEG0-SEG39), 8 common signals[Note 1] (COM0 to COM7) μPD78F049x: 32 segment signals[Note 1] (SEG0-SEG31), 8 common signals[Note 1] (COM0 to COM7)
Segment key source output	Segment key source signal: 8 (SEG24 (KS0)-SEG31 (KS7))
Control registers	LCD mode register (LCDMD) LCD display mode register (LCDM) LCD clock control register (LCDC0) Port function register 2 (PF2)[Note 2] Port function register ALL (PFALL) Key return mode register (KRM) Port mode register 4 (PM4) Pull-up resistor option register4 (PU4) Port register 14 (P14) Port register 15 (P15)

Notes 1. The four segment signal outputs (SEG0 to SEG3) and four common signal outputs (COM4 to COM7) are alternate-function pins. COM4 to COM7 can be used only when eight-time-slice mode is selected by the setting of the LCD display mode register.

2. μPD78F047x and 78F048x only.

Figure 18-1. Block Diagram of LCD Controller/Driver



18.3 Registers Controlling LCD Controller/Driver

The following ten registers are used to control the LCD controller/driver.

- LCD mode register (LCDMD)
- LCD display mode register (LCDM)
- LCD clock control register (LCDC0)
- Port function register 2 (PF2)[Note]
- Port function register ALL (PFALL)
- Key return mode register (KRM)
- Port mode register 4 (PM4)
- Pull-up resistor option register4 (PU4)
- Port register 14 (P14)
- Port register 15 (P15)

Note μPD78F047x and 78F048x only.

(1) LCD mode register (LCDMD)

LCDMD sets the LCD drive voltage generator.

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

Reset signal generation sets LCDMD to 00H.

Figure 18-2. Format of LCD Mode Register

Address: FFB0H After reset: 00H R/W[note 1]

Symbol	7	6	5	4	3	2	1	0
LCDMD	0	0	MDSET1	MDSET0	0	0	KSF	KSON

MDSET1	MDSET0	LCD drive voltage generator selection
0	0	External resistance division method, internal resistor disconnection
0	1	Internal resistance division method, internal resistor connection (no step-down transforming, Used when $V_{LCD} = V_{DD}$)
1	1	Internal resistance division method, internal resistor connection (step-down transforming, Used when $V_{LCD} = 3/5V_{DD}$)
Other than above		Setting prohibited

KSF	Segment key scan status
0	LCD display signal being output
1	Segment key scan signal being output

KSON	Segment key scan function control
0	Segment key scan function is not used
1	Segment key scan function is used[note 2]

Notes 1. Bit 1 is read-only.

2. Use the segment key scan function if V_{DD} is equal to V_{LC0}.

Only the KR0 to KR7 pins can be used as input pins for the segment key scan function.

Caution Bits 0 to 2, 3, 6 and 7 must be set to 0.

(2) LCD display mode register (LCDM)

LCDM specifies whether to enable display operation. It also specifies whether to enable segment pin/common pin output, gate booster circuit control, and the display mode.

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

Reset signal generation sets LCDM to 00H.

Figure 18-3. Format of LCD Display Mode Register

Address: FFB1H After reset: 00H R/W

Symbol	<7>	<6>	5	<4>	3	2	1	0
LCDM	LCDON	SCOC	0	VAON	0	LCDM2	LCDM1	LCDM0

LCDON	LCD display enable/disable
0	Display off (all segment outputs are deselected.)
1	Display on

SCOC	Segment pin/common pin output control[Note 1]
0	Output ground level to segment/common pin
1	Output deselect level to segment pin and LCD waveform to common pin

VAON	Gate booster circuit control[Notes 1, 2]
0	No gate voltage boosting
1	Gate voltage boosting

LCDM2	LCDM1	LCDM0	LCD controller/driver display mode selection	
			Resistance division method	
			Number of time slices	Bias mode
1	1	1	8 [Note 3]	1/4 [Note 4]
0	0	0	4 [Note 3]	1/3
0	0	1	3 [Note 3]	1/3
0	1	0	2 [Note 3]	1/2
0	1	1	3 [Note 3]	1/2
1	0	0	Static	
Other than above			Setting prohibited	

(Note and Caution are listed on the next page.)

Notes 1. When LCD display is not to be performed or not required, power consumption can be reduced by using the following settings.
<1> Set both SCOC and VAON to 0.
<2> When the internal resistance division method is used, assume MDSET1, MDSET0 = (0, 0).
(The current flowing to the internal resistors can be reduced.)

2. This bit is used to control boosting of the internal gate signal of the LCD controller/driver.
If set to "Internal gate voltage boosting", the LCD drive performance can be enhanced.
Set VAON based on the following conditions.
<When set to the static display mode>
- When 2.0 V $\le V_{LCD} \le V_{DD} \le$ 5.5 V: VAON = 0
- When 1.8 V $\le V_{LCD} \le V_{DD} \le$ 3.6 V: VAON = 1

<When set to the 1/3 bias method>
- When 2.5 V $\le V_{LCD} \le V_{DD} \le$ 5.5 V: VAON = 0
- When 1.8 V $\le V_{LCD} \le V_{DD} \le$ 3.6 V: VAON = 1

<When set to the 1/2 bias method or 1/4 bias method >
- When 2.7 V $\le V_{LCD} \le V_{DD} \le$ 5.5 V: VAON = 0
- When 1.8 V $\le V_{LCD} \le V_{DD} \le$ 3.6 V: VAON = 1

3. When using the segment key scan function (KSON = 1), "number of time slices + 1" is added for segment key scan signal output.

4. When the P40/KR0/V_{LC3} pin is set to the 1/4 bias method, it is used as V_{LC3}. When the pin is set to another bias method, it is used for the port function (P40) or the key interrupt function (KR0).

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

2. When displaying in a mode with a large number of COMs, such as 8 COM, V_{LC0} may not be able to obtain sufficient contrast under the low-voltage conditions, depending on the panel characteristics. Use the LCD controller/driver after having performed thorough LCD display evaluation and confirmed that there are no problems regarding the display quality.

(3) LCD clock control register (LCDC0)

LCDC0 specifies the LCD source clock and LCD clock.

The frame frequency is determined according to the LCD clock and the number of time slices.

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

Reset signal generation sets LCDC0 to 00H.

Figure 18-4. Format of LCD Clock Control Register

Address: FFB2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDC0	0	LCDC6	LCDC5	LCDC4	0	LCDC2	LCDC1	LCDC0

LCDC6	LCDC5	LCDC4	LCD source clock (f_{LCD}) selection
0	0	0	f_{XT} (32.768 kHz)
0	0	1	$f_{PRS}/2^6$
0	1	0	$f_{PRS}/2^7$
0	1	1	$f_{PRS}/2^8$
1	0	0	$f_{RL}/2^3$
Other than above			Setting prohibited

LCDC2	LCDC1	LCDC0	LCD clock (LCDCL) selection
0	0	0	$f_{LCD}/2^4$
0	0	1	$f_{LCD}/2^5$
0	1	0	$f_{LCD}/2^6$
0	1	1	$f_{LCD}/2^7$
1	0	0	$f_{LCD}/2^8$
1	0	1	$f_{LCD}/2^9$
Other than above			Setting prohibited

Caution Bits 3 and 7 must be set to 0.

Remarks 1. f_{XT}: XT1 clock oscillation frequency

2. f_{PRS}: Peripheral hardware clock frequency

3. f_{RL}: Internal low-speed oscillation clock frequency

(4) Port function register 2 (PF2) (μPD78F047x and 78F048x only)

This register sets whether to use pins P20 to P27 as port pins (other than segment output pins) or segment output pins.

PF2 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PF2 to 00H.

Figure 18-5. Format of Port Function Register 2

Address: FFB5H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PF2	PF27	PF26	PF25	PF24	PF23	PF22	PF21	PF20

PF2n	Port/segment output specification
0	Used as port (other than segment output)
1	Used as segment output

Remark n = 0 to 7

(5) Port function register ALL (PFALL)

This register sets whether to use pins P8 to P11 and P13 to P15 as port pins (other than segment output pins) or segment output pins.

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

Reset signal generation sets PFALL to 00H.

Figure 18-6. Format of Port Function Register ALL

Address: FFB6H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PFALL	0	PF15ALL	PF14ALL	PF13ALL	PF11ALL	PF10ALL	PF09ALL	PF08ALL

PFnALL	Port/segment output specification
0	Used as port (other than segment output)
1	Used as segment output

Remark n = 08 to 11, 13 to 15

(6) Key return mode register (KRM)

This register is used to specify that a pin is to be used as a segment key scan input pin when using the segment key scan function.

See Figure 22-2 Format of Key Return Mode Register (KRM) when not using the segment key scan function.

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

Reset signal generation sets 00H.

Figure 18-7. Format of Key return mode register (KRM)

Address: FF6EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRM	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	Setting segment key scan input pin (n = 0 to 7)
0	Does not use specified pin as segment key scan input pin.
1	Uses specified pin as segment key scan input pin.

Cautions 1. If KRM is changed, the interrupt request flag may be set. Therefore, disable interrupts and then change the KRM register. Clear the interrupt request flag and enable interrupts.

2. When set not to use the specified pin as a segment key scan input pin (KRMn = 0), the corresponding P4n pin can be used as a normal port.

3. When using the P40/KR0/V_{LC3} pin for the key interrupt function (KR0), set the LCD display mode register (LCDM) to a setting other than the 1/4 bias method. When set to the 1/4 bias method, the P40/KR0/V_{LC3} pin functions as V_{LC3}.

(7) Port mode register 4 (PM4)

This register sets port 4 input/output in 1-bit units.

When using the segment key scan function, set the port mode register of the port to be used to 1 (PM4n = 1), in order to set the P4n pin as a key scan input pin.

PM4 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets FFH.

Figure 18-8. Format of Port mode register 4 (PM4)

Address: FF24H After reset: FFH R/W

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

PF4n	P4n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

(8) Pull-up resistor option register 4 (PU4)

This register is used to set whether to use the on-chip pull-up resistors of P40 to P47.

When using the segment key scan function, set the pull-up resistor option register of the port to be used to 0 (PU4n = 0), in order to set the P4n pin as a key scan input pin.

An external pull-up resistor cannot be used, because it affects the LCD display output.

PU4 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets 00H.

Figure 18-9. Format of Pull-up Resistor Option Register 4

Address: FF34H After reset: 00H R/W

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

PF4n	P4n pin on-chip pull-up resistor selection (n = 0 to 7)	
	Pin using segment key scan function	Pin not using segment key scan function
0	On-chip pull-up resistor connected only during segment key scan output period	On-chip pull-up resistor not connected
1	Setting prohibited	On-chip pull-up resistor connected

(9) Port register 14 (P14)

This register is used to perform the first half of KS0 to KS3 output control by using bits 0 to 3, and the latter half of KS0 to KS3 output control by using bits 4 to 7, when using the segment key scan function.

When using the P14n pin for segment key scan output, the P14n and PK14n bits are used for control.

See **Figure 4-28 Format of Port Register** when not using the segment key scan function.

P14 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets 00H.

Figure 18-10. Format of Port register 14 (P14)

Address: FF0EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
P14	PK143	PK142	PK141	PK140	P143	P142	P141	P140

P140-P143	First half of KS0 to KS3 output control
0	Low-level output
1	High-level output

PK140-PK143	Latter half of KS0 to KS3 output control
0	Low-level output
1	High-level output

(10) Port register 15 (P15)

This register is used to perform the first half of KS4 to KS7 output control by using bits 0 to 3, and the latter half of KS4 to KS7 output control by using bits 4 to 7, when using the segment key scan function.

When using the P15n pin for segment key scan output, the P15n and PK15n bits are used for control.

See **Figure 4-28 Format of Port Register** when not using the segment key scan function.

P15 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets 00H.

Figure 18-11. Format of Port register 15 (P15)

Address: FF0FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
P15	PK153	PK152	PK151	PK150	P153	P152	P151	P150

P150-P153	First half of KS4 to KS7 output control
0	Low-level output
1	High-level output

PK150-PK153	Latter half of KS4 to KS7 output control
0	Low-level output
1	High-level output

18.4 Setting LCD Controller/Driver

18.4.1 Setting method when not using segment key scan function (KSON = 0)

When not using the segment key scan function (KSON = 0), set the LCD controller/driver as follows. Set the LCD controller/driver using the following procedure.

<1> Set (VAON = 1) internal gate voltage boosting (bit 4 of the LCD display mode register (LCDM)).[Note]
<2> Set the resistance division method via MDSET0 and MDSET1 (bits 4 and 5 of the LCD mode register (LCDMD)) (MDSET0 = 0: external resistance division method, MDSET0 = 1: internal resistance division method).
<3> Set the pins to be used as segment outputs to the port function registers (PF2m, PFnALL).
<4> Set an initial value to the RAM for LCD display.
<5> Set the number of time slices via LCDM0 to LCDM2 (bits 0 to 2 of the LCD display mode register (LCDM)).
<6> Set the LCD source clock and LCD clock via LCD clock control register 0 (LCDC0).
<7> Set (SCOC = 1) SCOC (bit 6 of the LCD display mode register (LCDM)).
Deselect signals are output from all the segment and common pins, and the non-display status is entered.
<8> Start output corresponding to each data memory by setting (LCDON = 1) LCDON (bit 7 of LCDM).

Subsequent to this procedure, set the data to be displayed in the data memory.

Note Set VAON based on the following conditions.

<When set to the static display mode>
- When 2.0 V ≤ V_{LCD} ≤ V_{DD} ≤ 5.5 V: VAON = 0
- When 1.8 V ≤ V_{LCD} ≤ V_{DD} ≤ 3.6 V: VAON = 1

<When set to the 1/3 bias method>
- When 2.5 V ≤ V_{LCD} ≤ V_{DD} ≤ 5.5 V: VAON = 0
- When 1.8 V ≤ V_{LCD} ≤ V_{DD} ≤ 3.6 V: VAON = 1

<When set to the 1/2 bias method or 1/4 bias method >
- When 2.7 V ≤ V_{LCD} ≤ V_{DD} ≤ 5.5 V: VAON = 0
- When 1.8 V ≤ V_{LCD} ≤ V_{DD} ≤ 3.6 V: VAON = 1

Remarks 1. Use the following procedure to set to the display-off state and disconnect the internal resistors when using the internal resistance division method.
<1> Clear LCDON (bit 7 of LCDM) (LCDON = 0).
Deselect signals are output from all segment pins and common pins, and a non-display state is entered.
<2> Clear SCOC (bit 6 of the LCD display mode register (LCDM)) (SCOC = 0).
Ground levels are output from all segment pins and common pins.
<3> Assume MDSET0, MDSET1 (bits 4 and 5 of the LCD mode register (LCDMD)) = (0, 0) and set the resistance division method to the external resistance division method.

2. m = 0 to 7, n = 08 to 11, 13 to 15

Caution When displaying in a mode with a large number of COMs, such as 8 COM, V_{LC0} may not be able to obtain sufficient contrast under the low-voltage conditions, depending on the panel characteristics. Use the LCD controller/driver after having performed thorough LCD display evaluation and confirmed that there are no problems regarding the display quality.

18.4.2 Setting method when using segment key scan function (KSON = 1)

When using the segment key scan function (KSON = 1), set the LCD controller/driver as follows. Set the LCD controller/driver using the following procedure.

<1> Set (VAON = 1) internal gate voltage boosting (bit 4 of the LCD display mode register (LCDM)).[Note 1]

<2> Set the resistance division method via MDSET0 and MDSET1 (bits 4 and 5 of the LCD mode register (LCDMD)) (MDSET0 = 0: external resistance division method, MDSET0 = 1: internal resistance division method).
Set (KSON = 1) KSON (bit 0 of the LCD mode register (LCDMD)).

<3> Set the pins to be used as segment outputs to the port function registers (PF2m, PFnALL).

<4> Use port mode register 4 (PM4) to set the pin to be used as a key scan input pin[Note 2] to PM4m = 1 (input mode).

<5> Use pull-up resistor option register 4 (PU4) to set the pin to be used as a key scan input pin[Note 2] to PU4m = 0 (connects an on-chip pull-up resistor only during the segment key scan output period).

<6> Use the key return mode register (KRM) to set the pin to be used as a segment key scan input pin to KRMm = 1[Note 3].

<7> Set an initial value to the RAM for LCD display.

<8> Set an initial value of segment key scan output to P14, P15.

<9> Set the number of time slices via LCDM0 to LCDM2 (bits 0 to 2 of the LCD display mode register (LCDM)).

<10> Set the LCD source clock and LCD clock via LCD clock control register 0 (LCDC0).

<11> Set (SCOC = 1) SCOC (bit 6 of the LCD display mode register (LCDM)).
Deselect signals are output from all the segment and common pins, and the non-display status is entered.

<12> Start output corresponding to each data memory by setting (LCDON = 1) LCDON (bit 7 of LCDM).

Hereinafter, set data to the data memory according to the contents displayed, and perform segment key scan output settings for the port registers (P14, P15) according to the contents of the segment key scan output.

Notes 1. Set VAON based on the following conditions.

<When set to the 1/3 bias method>

- When 2.5 V $\le V_{LCD} = V_{DD} \le$ 5.5 V: VAON = 0
- When 1.8 V $\le V_{LCD} = V_{DD} \le$ 3.6 V: VAON = 1

<When set to the 1/2 bias method or 1/4 bias method >

- When 2.7 V $\le V_{LCD} = V_{DD} \le$ 5.5 V: VAON = 0
- When 1.8 V $\le V_{LCD} = V_{DD} \le$ 3.6 V: VAON = 1

2. When using the segment key scan function, be sure to set port 4 as a segment key scan input pin and the pull-up resistor option register of the port to be used to PU4m = 0 (connects an on-chip pull-up resistor only during the segment key scan output period).
An external pull-up resistor cannot be used, because it affects the LCD display output.

3. An interrupt request flag may be set when KRM has been changed. Consequently, change the KRM register after having disabled interrupts, and enable interrupts after having cleared the interrupt request flag.

Remarks 1. Use the following procedure to set to the display-off state and disconnect the internal resistors when using the internal resistance division method.

<1> Clear LCDON (bit 7 of LCDM) (LCDON = 0).
Deselect signals are output from all segment pins and common pins, and a non-display state is entered.

<2> Clear SCOC (bit 6 of the LCD display mode register (LCDM)) (SCOC = 0).
Ground levels are output from all segment pins and common pins.

<3> Assume MDSET0, MDSET1 (bits 4 and 5 of the LCD mode register (LCDMD)) = (0, 0) and set the resistance division method to the external resistance division method.

2. m = 0 to 7, n = 08 to 11, 13 to 15

Caution When displaying in a mode with a large number of COMs, such as 8 COM, V_{LC0} may not be able to obtain sufficient contrast under the low-voltage conditions, depending on the panel characteristics. Use the LCD controller/driver after having performed thorough LCD display evaluation and confirmed that there are no problems regarding the display quality.

18.5 LCD Display Data Memory

The LCD display data memory is mapped at addresses FA40H to FA67H (μPD78F047x and 78F048x) or FA40H to FA5FH (μPD78F049x). Data in the LCD display data memory can be displayed on the LCD panel using the LCD controller/driver.

Figure 18-12 shows the relationship between the contents of the LCD display data memory and the segment/common outputs.

The areas not to be used for display can be used as normal RAM.

Figure 18-12. Relationship between LCD Display Data Memory Contents and Segment/Common Outputs

(a) μPD78F047x, 78F048x

	b7	b6	b5	b4	b3	b2	b1	b0	
FA67H									SEG39
FA66H									SEG38
FA65H									SEG37
FA45H									SEG5
FA44H									SEG4
FA43H	0	0	0	0					SEG3
FA42H	0	0	0	0					SEG2
FA41H	0	0	0	0					SEG1
FA40H	0	0	0	0					SEG0
	COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0	

(b) μPD78F049x

	b7	b6	b5	b4	b3	b2	b1	b0	
FA5FH									SEG31
FA5EH									SEG30
FA5DH									SEG29
FA45H									SEG5
FA44H									SEG4
FA43H	0	0	0	0					SEG3
FA42H	0	0	0	0					SEG2
FA41H	0	0	0	0					SEG1
FA40H	0	0	0	0					SEG0
	COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0	

Caution No memory is allocated to the higher 4 bits of FA40H to FA43H. Be sure to set there bits to 0.

18.6 Common and Segment Signals

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

Applying DC voltage to the common and segment signals of an LCD panel causes deterioration. To avoid this problem, this LCD panel is driven by AC voltage.

(1) Common signals

Each common signal is selected sequentially according to a specified number of time slices at the timing listed in Table 18-3. In the static display mode, the same signal is output to COM0 to COM3.

When using the segment key scan output function (KSON = 1), segment key scan output will be performed for a period of one time slice after one LCD output cycle. The common signal generated at that time will not be displayed when output.

In the two-time-slice mode, leave the COM2 and COM3 pins open. In the three-time-slice mode, leave the COM3 pin open.

Use the COM4 to COM7 pins other than in the eight-time-slice mode as open or segment pins.

Table 18-3. COM Signals

COM Signal / Number of Time Slices	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7
Static display mode					**Note 2**	**Note 2**	**Note 2**	**Note 2**
Two-time-slice mode[Note 1]			Open	Open	**Note 2**	**Note 2**	**Note 2**	**Note 2**
Three-time-slice mode[Note 1]				Open	**Note 2**	**Note 2**	**Note 2**	**Note 2**
Four-time-slice mode[Note 1]					**Note 2**	**Note 2**	**Note 2**	**Note 2**
eight-time-slice mode[Note 1]								

Notes 1. When using the segment key scan output function (KSON = 1), non-display output will be performed for a period of one time slice after one LCD output cycle.

2. Use the pins as open or segment pins.

(2) Segment signals

(a) μPD78F047x, 78F048x

The segment signals correspond to 40 bytes of the LCD display data memory (FA40H to FA67H) during an LCD display period, bits 0, 1, 2, and 3 of each byte are read in synchronization with COM0, COM1, COM2, and COM3, respectively. If a bit is 1, it is converted to the select voltage, and if it is 0, it is converted to the deselect voltage. The conversion results are output to the segment pins (SEG0 to SEG39).

Furthermore, segment signals correspond to the setting values of port registers 14 and 15 during a segment key scan output period. Bits 0 to 3 and bits 4 to 7 of each port register will be read in synchronization with the first half and latter half of the segment key scan output period, respectively, and if the content of each bit is 1 or 0, high levels or low levels will be output to the segment pins (SEG24 to SEG31), respectively.

(b) μPD78F049x

The segment signals correspond to 32 bytes of the LCD display data memory (FA40H to FA5FH) during an LCD display period, bits 0, 1, 2, and 3 of each byte are read in synchronization with COM0, COM1, COM2, and COM3, respectively. If a bit is 1, it is converted to the select voltage, and if it is 0, it is converted to the deselect voltage. The conversion results are output to the segment pins (SEG0 to SEG31).

Furthermore, segment signals correspond to the setting values of port registers 14 and 15 during a segment key scan output period. Bits 0 to 3 and bits 4 to 7 of each port register will be read in synchronization with the first half and latter half of the segment key scan output period, respectively, and if the content of each bit is 1 or 0, high levels or low levels will be output to the segment pins (SEG24 to SEG31), respectively.

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

LCD display data memory bits 1 to 3, bits 2 and 3, and bit 3 are not used for LCD display in the static display, two-time slot, and three-time slot modes, respectively. So these bits can be used for purposes other than display.

The higher 4 bits of FA40H to FA43H are fixed to 0.

(3) Output waveforms of common signals and segment signals during LCD display signal output period

The voltages shown in Table 18-4 are output to the common signals and segment signals during the LCD display signal output period.

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

Table 18-4. LCD Drive Voltage

(a) Static display mode (during LCD display signal output period)

Segment Signal / Common Signal	Select Signal Level	Deselect Signal Level
	V_{SS}/V_{LC0}	V_{LC0}/V_{SS}
V_{LC0}/V_{SS}	$-V_{LCD}/+V_{LCD}$	0 V/0 V

(b) 1/2 bias method (during LCD display signal output period)

Segment Signal / Common Signal		Select Signal Level	Deselect Signal Level
		V_{SS}/V_{LC0}	V_{LC0}/V_{SS}
Select signal level	V_{LC0}/V_{SS}	$-V_{LCD}/+V_{LCD}$	0 V/0 V
Deselect signal level	$V_{LC1} = V_{LC2}$	$-\frac{1}{2}V_{LCD}/+\frac{1}{2}V_{LCD}$	$+\frac{1}{2}V_{LCD}/-\frac{1}{2}V_{LCD}$

(c) 1/3 bias method (during LCD display signal output period)

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

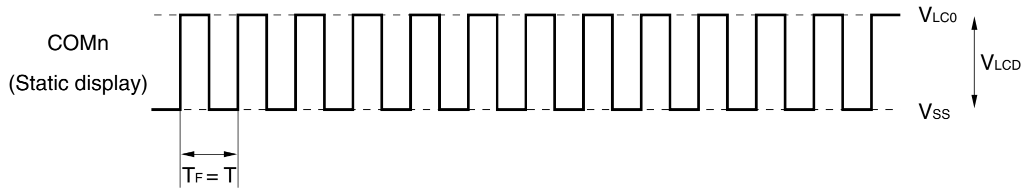
(d) 1/4 bias method (during LCD display signal output period)

Segment Signal / Common Signal		Select Signal Level	Deselect Signal Level
		V_{LC0}/V_{SS}	V_{LC1}/V_{LC2}
Select signal level	V_{SS}/V_{LC0}	$+V_{LCD}/-V_{LCD}$	$+\frac{1}{2}V_{LCD}/-\frac{1}{2}V_{LCD}$
Deselect signal level	V_{LC1}/V_{LC3}	$+\frac{1}{4}V_{LCD}/-\frac{1}{4}V_{LCD}$	$-\frac{1}{4}V_{LCD}/+\frac{1}{4}V_{LCD}$

Figure 18-13 shows the common signal waveforms, and Figure 18-14 shows the voltages and phases of the common and segment signals.

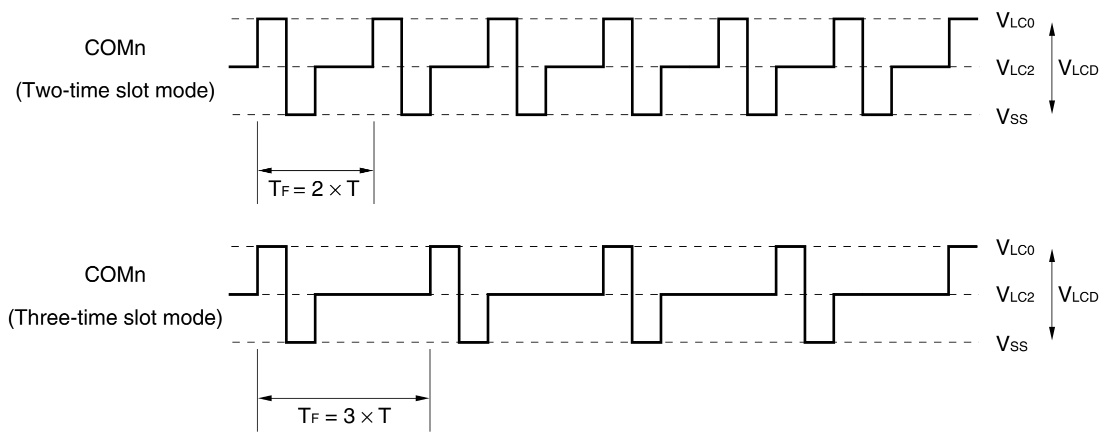
Figure 18-13. Common Signal Waveforms

(a) Static display mode



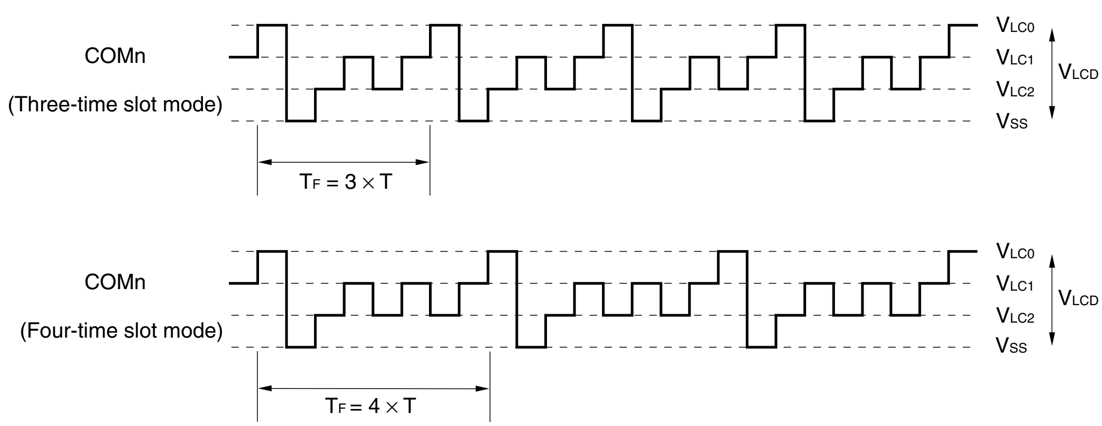
T: One LCD clock period TF: Frame frequency

(b) 1/2 bias method



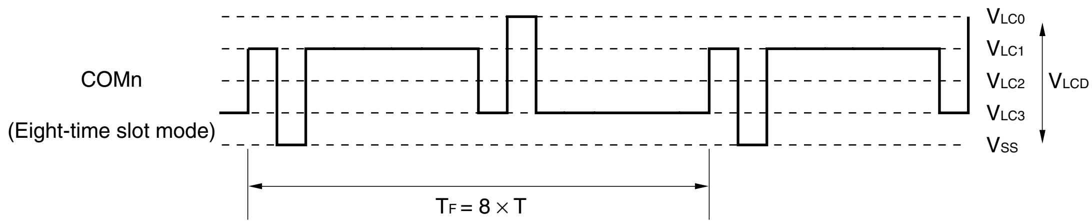
T: One LCD clock period TF: Frame frequency

(c) 1/3 bias method



T: One LCD clock period TF: Frame frequency

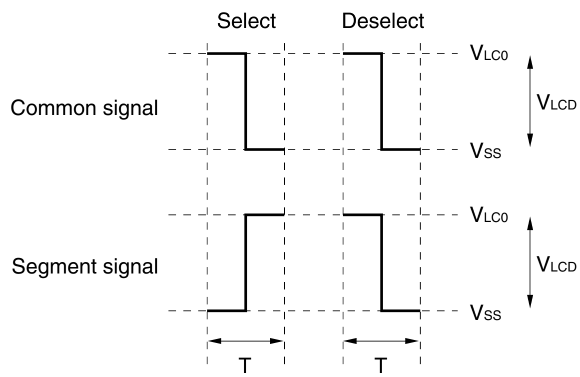
(d) 1/4 bias method



T: One LCD clock period T_F: Frame frequency

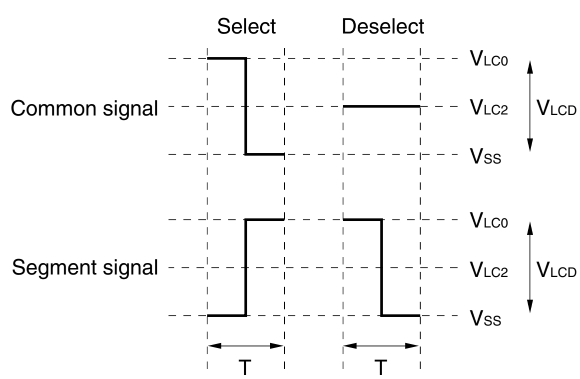
Figure 18-14 Voltages and Phases of Common and Segment Signals

(a) Static display mode



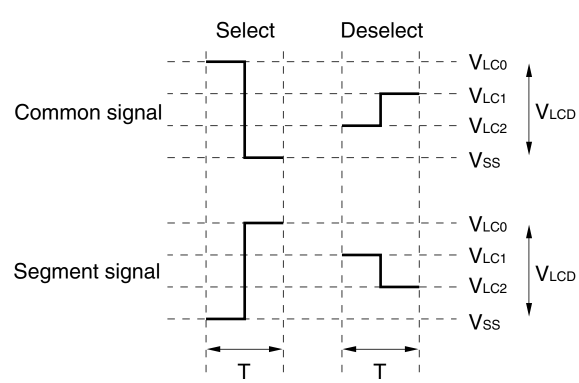
T: One LCD clock period

(b) 1/2 bias method



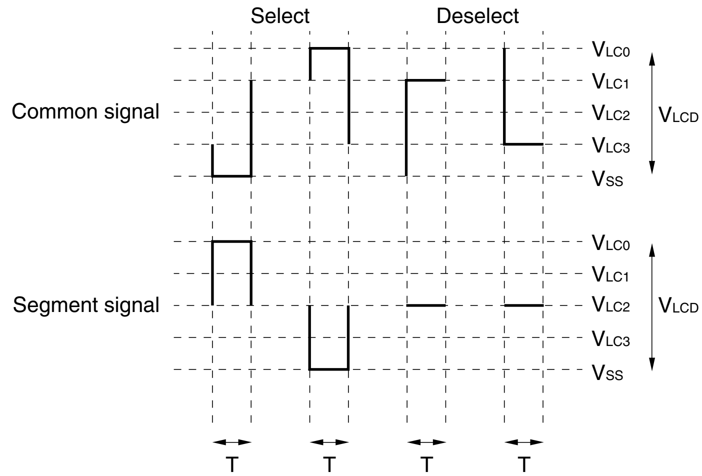
T: One LCD clock period

(c) 1/3 bias method



T: One LCD clock period

(d) 1/4 bias method



T: One LCD clock period

(4) Output waveforms of common and segment signals during segment key scan output period

The voltages shown in Table 18-5 are output to the common signals and segment signals during the segment key scan output period.

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

Table 18-5. LCD Drive Voltage

(a) 1/2 bias method (during segment key scan output period)

Common Signal \ Key scan Signal		P14x = P15x = 1	P14x = P15x = 0
		V_{D0}	V_{SS}
Deselect signal level	$V_{LC1} = V_{LC2}$	$+\frac{1}{2}V_{LCD}$	$-\frac{1}{2}V_{LCD}$

(b) 1/3 bias method (during segment key scan output period)

Common Signal \ Key scan Signal		P14x = P15x = 1	P14x = P15x = 0
		V_{DD}	V_{SS}
Deselect signal level	V_{LC2}/V_{LC1}	$+\frac{2}{3}V_{LCD}/+\frac{1}{3}V_{LCD}$	$-\frac{1}{3}V_{LCD}/-\frac{2}{3}V_{LCD}$

(c) 1/4 bias method (during segment key scan output period)

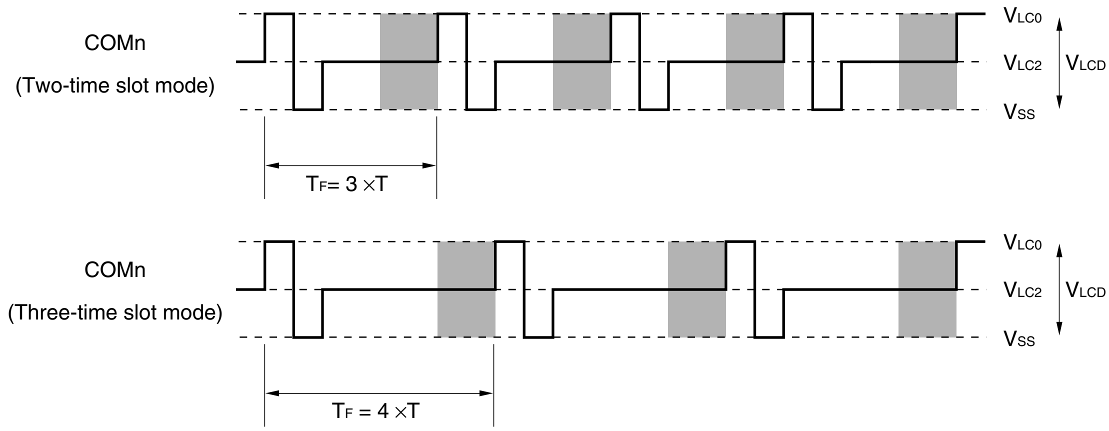
Common Signal \ Key scan Signal		P14x = P15x = 1	P14x = P15x = 0
		V_{DD}	V_{SS}
Deselect signal level	V_{LC1}/V_{LC3}	$+\frac{1}{4}V_{LCD}/+\frac{3}{4}V_{LCD}$	$-\frac{3}{4}V_{LCD}/-\frac{1}{4}V_{LCD}$

Remark The segment key scan output function cannot be used in the static display mode.

Figure 18-15 shows the common signal waveforms, and Figure 18-16 shows the voltages and phases of the common and segment signals.

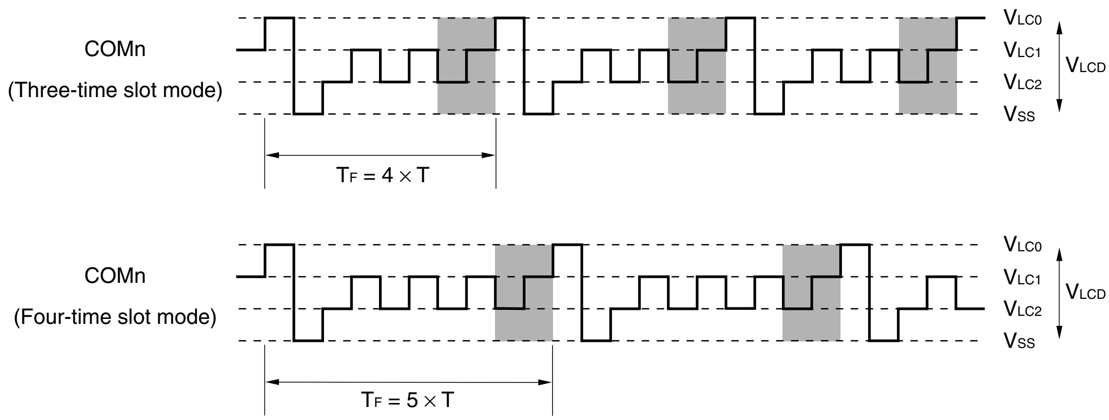
Figure 18-15 Common Signal Waveforms

(a) 1/2 bias method



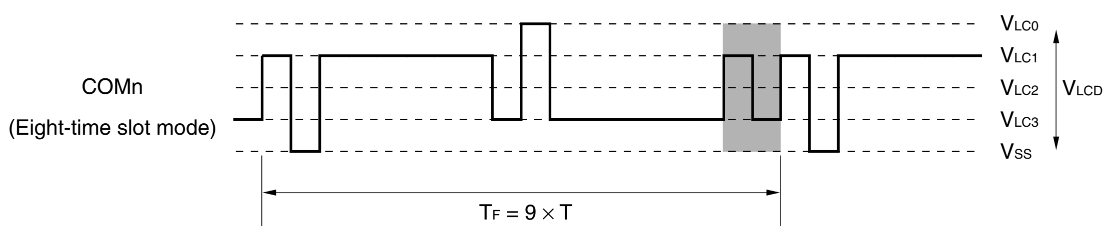
T: One LCD clock period TF: Frame frequency Shaded sections: Segment key scan output period

(b) 1/3 bias method



T: One LCD clock period TF: Frame frequency Shaded sections: Segment key scan output period

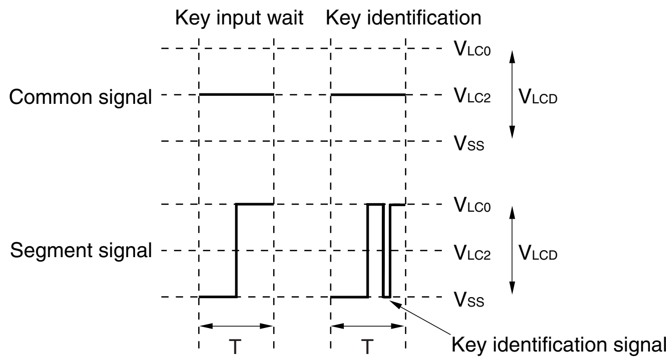
(c) 1/4 bias method



T: One LCD clock period TF: Frame frequency Shaded sections: Segment key scan output period

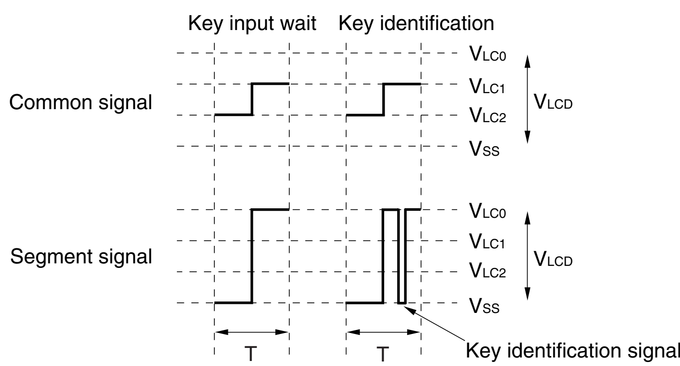
Figure 18-16 Voltages and Phases of Common and Segment Signals

(a) 1/2 bias method



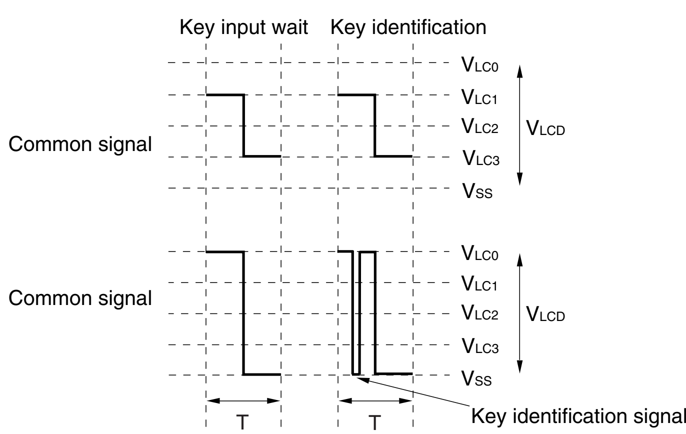
T: One LCD clock period

(b) 1/3 bias method



T: One LCD clock period

(c) 1/4 bias method



T: One LCD clock period

Remark Segment key scan signals must be set by using port registers 14 and 15 (P14, P15).

18.7 Display Modes

18.7.1 Static display example

Figure 18-18 shows how the three-digit LCD panel having the display pattern shown in Figure 18-17 is connected to the segment signals (SEG0 to SEG23) and the common signal (COM0) of the 78K0/LF3 chip. This example displays data "12.3" in the LCD panel. The contents of the display data memory (FA40H to FA57H) correspond to this display.

The following description focuses on numeral "2." (2.) displayed in the second digit. To display "2." in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG8 to SEG15 pins according to Table 18-6 at the timing of the common signal COM0; see **Figure 18-17** for the relationship between the segment signals and LCD segments.

Table 18-6. Select and Deselect Voltages (COM0)

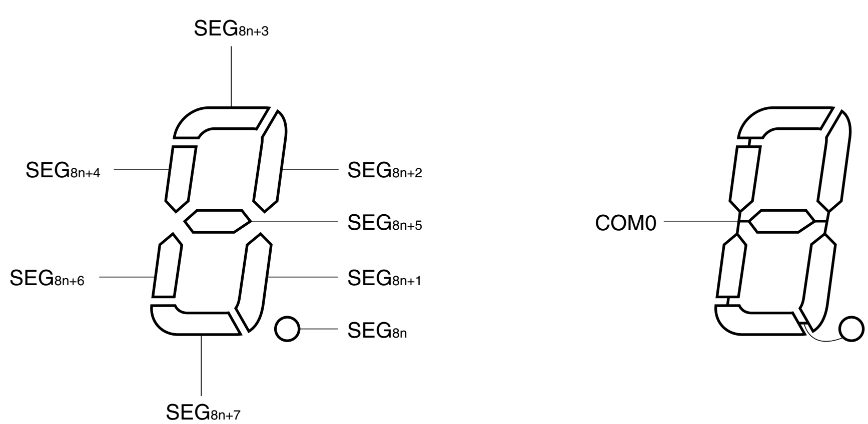
Segment / Common	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15
COM0	Select	Deselect	Select	Select	Deselect	Select	Select	Select

According to Table 18-6, it is determined that the bit-0 pattern of the display data memory locations (FA48H to FA4FH) must be 10110111.

Figure 18-19 shows the LCD drive waveforms of SEG11 and SEG12, and COM0. When the select voltage is applied to SEG11 at the timing of COM0, an alternate rectangle waveform, $+V_{LCD}/-V_{LCD}$, is generated to turn on the corresponding LCD segment.

COM1 to COM3 are supplied with the same waveform as for COM0. So, COM0 to COM3 may be connected together to increase the driving capacity.

Figure 18-17 Static LCD Display Pattern and Electrode Connections



Remark n = 0 to 2

Figure 18-18. Example of Connecting Static LCD Panel

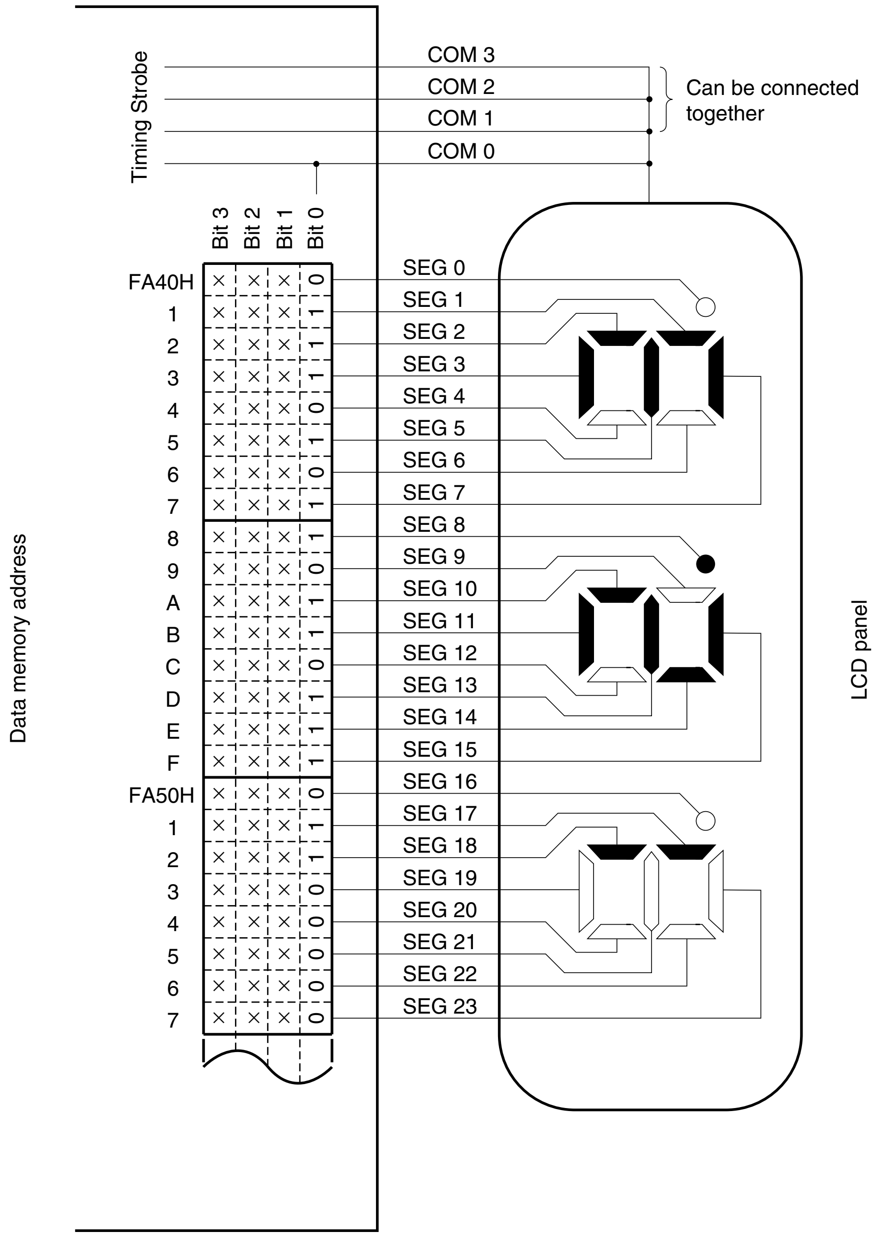
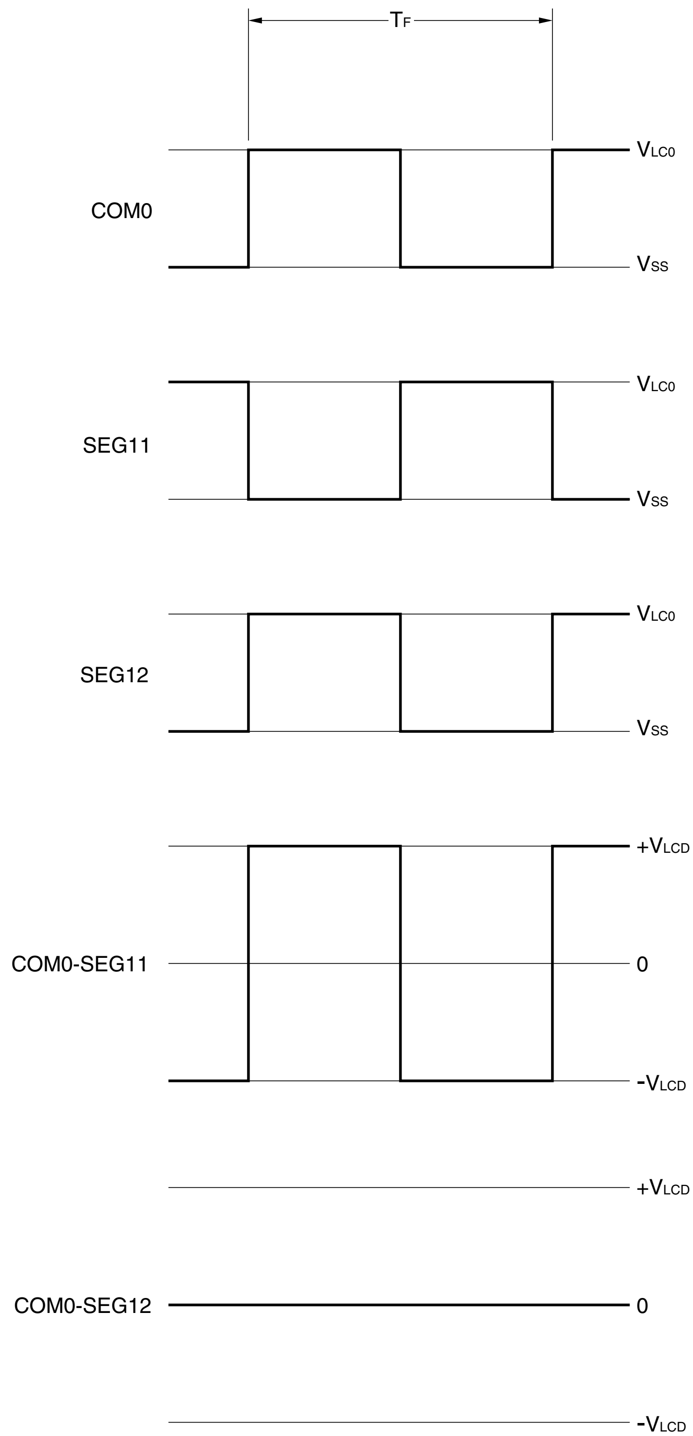


Figure 18-19. Static LCD Drive Waveform Examples



18.7.2 Two-time-slice display example

Figure 18-21 shows how the 6-digit LCD panel having the display pattern shown in Figure 18-20 is connected to the segment signals (SEG0 to SEG23) and the common signals (COM0 and COM1) of the 78K0/LF3 chip. This example displays data "12345.6" in the LCD panel. The contents of the display data memory (FA40H to FA57H) correspond to this display.

The following description focuses on numeral "3" (3) displayed in the fourth digit. To display "3" in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG12 to SEG15 pins according to Table 18-7 at the timing of the common signals COM0 and COM1; see Figure 18-20 for the relationship between the segment signals and LCD segments.

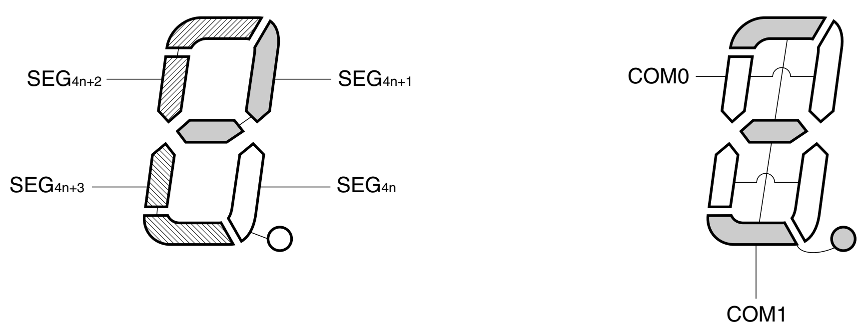
Table 18-7. Select and Deselect Voltages (COM0 and COM1)

Segment / Common	SEG12	SEG13	SEG14	SEG15
COM0	Select	Select	Deselect	Deselect
COM1	Deselect	Select	Select	Select

According to Table 18-7, it is determined that the display data memory location (FA4FH) that corresponds to SEG15 must contain xx10.

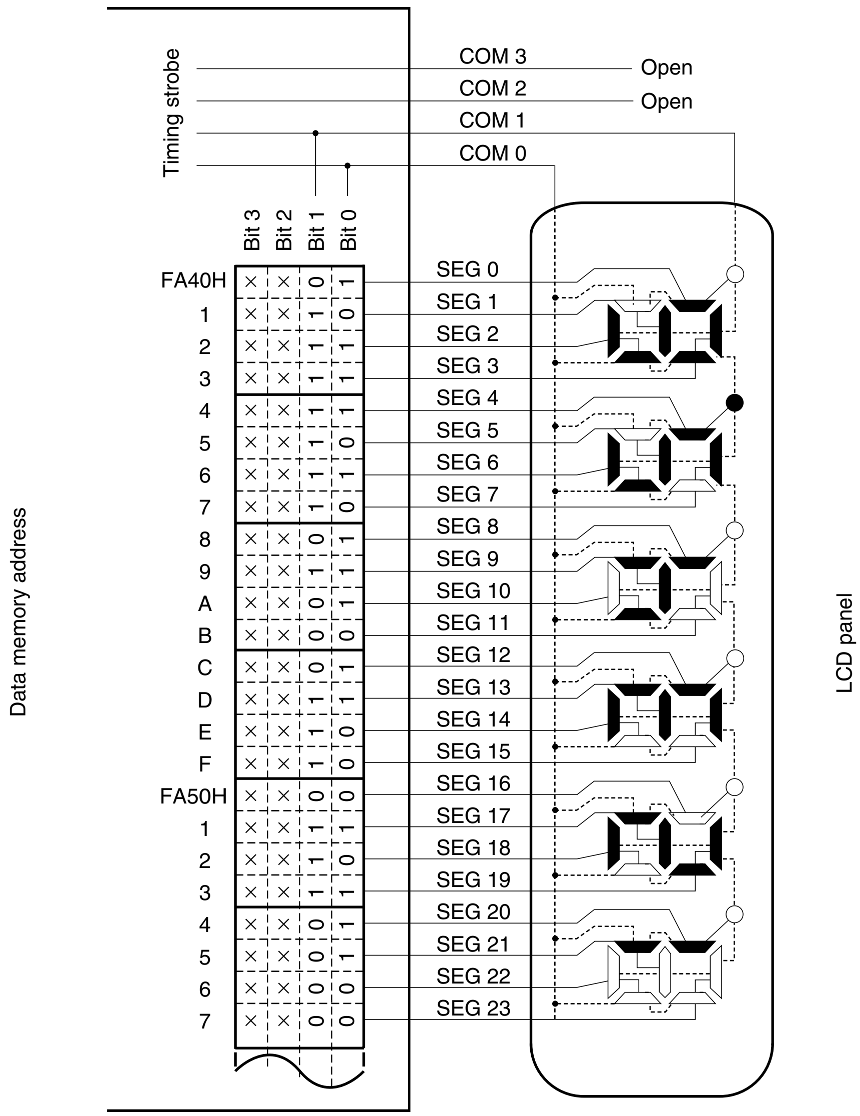
Figure 18-22 shows examples of LCD drive waveforms between the SEG15 signal and each common signal. When the select voltage is applied to SEG15 at the timing of COM1, an alternate rectangle waveform, $+V_{LCD}/-V_{LCD}$, is generated to turn on the corresponding LCD segment.

Figure 18-20. Two-Time-Slice LCD Display Pattern and Electrode Connections



Remark n = 0 to 5

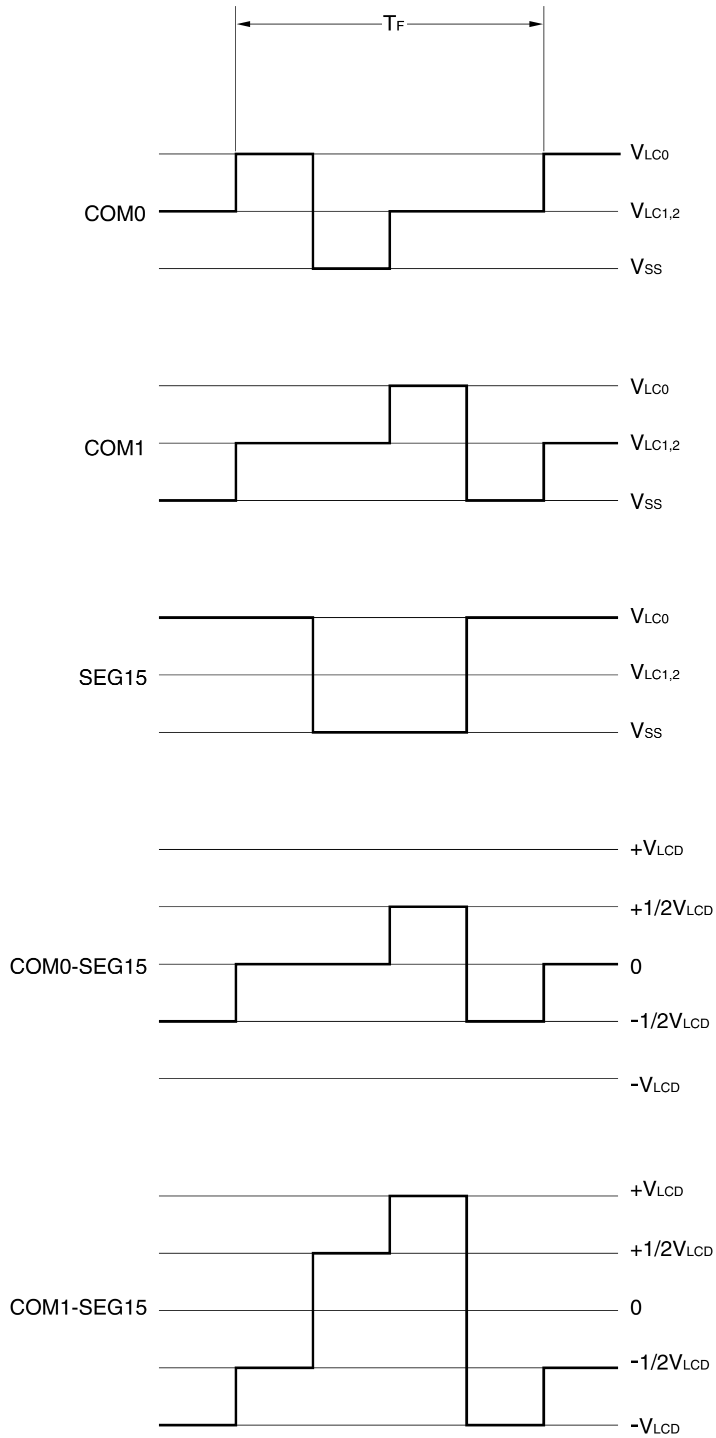
Figure 18-21. Example of Connecting Two-Time-Slice LCD Panel



×: Can always be used to store any data because the two-time-slice mode is being used.

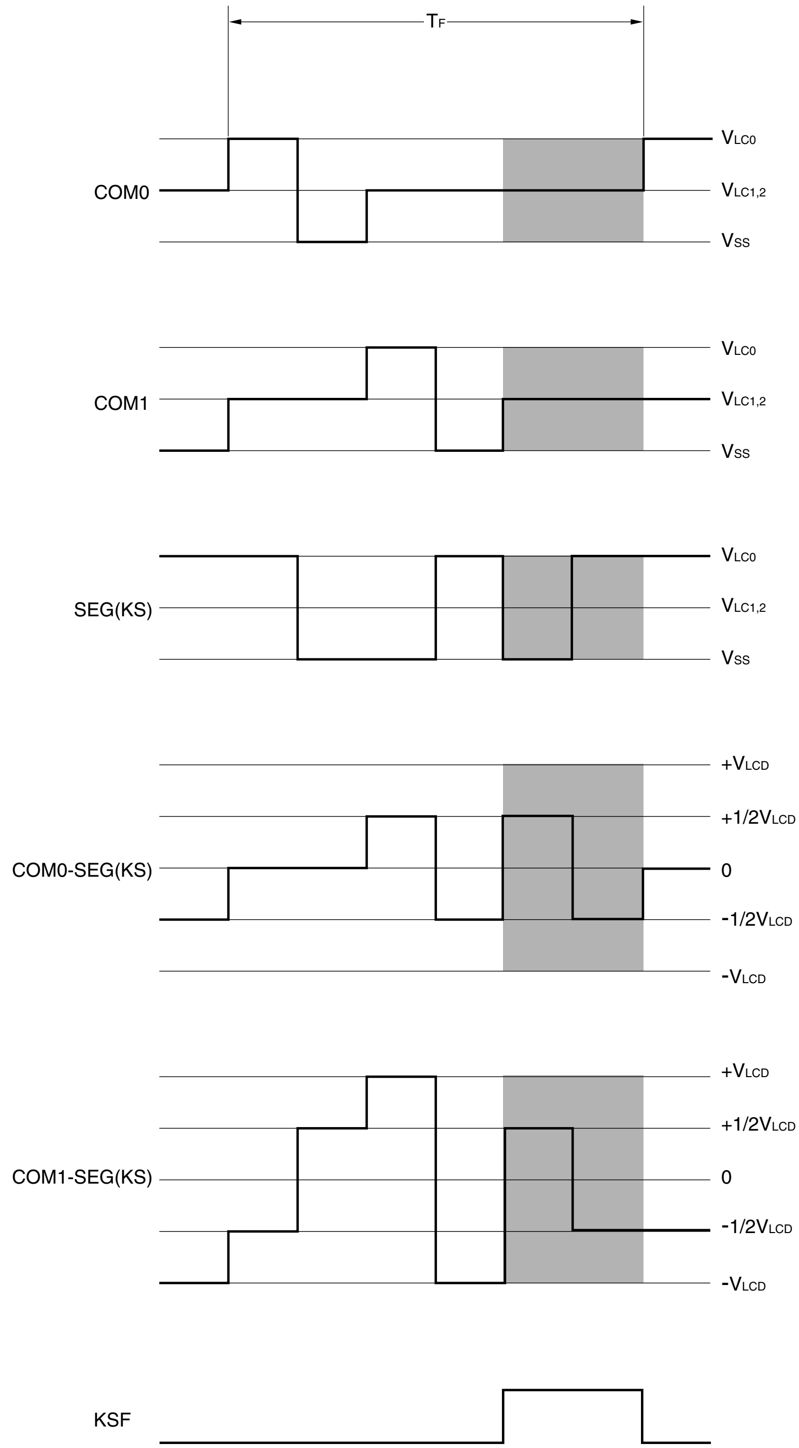
Figure 18-22. Two-Time-Slice LCD Drive Waveform Examples (1/2 Bias Method)

(a) When segment key scan function is not used (KSON = 0)



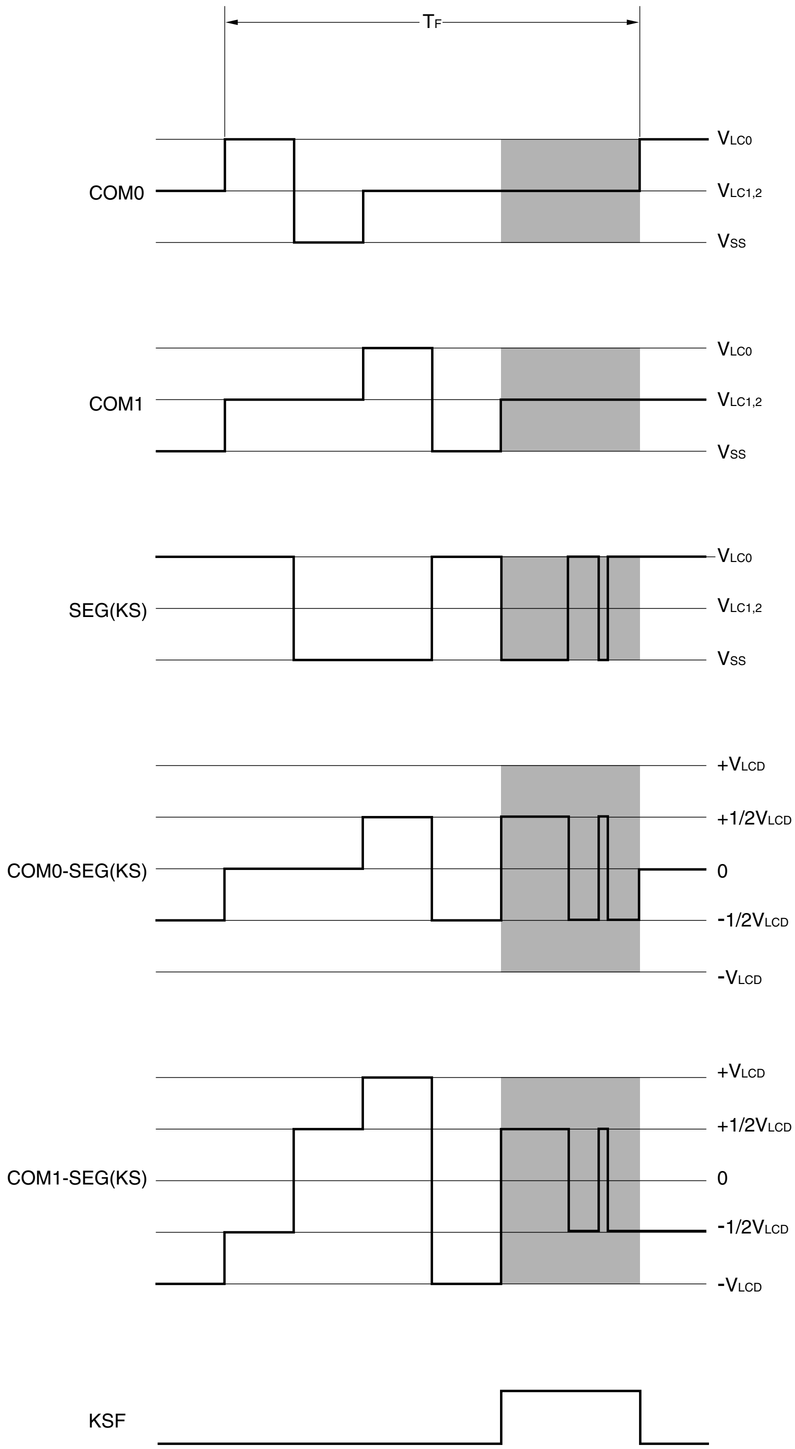
(b) When segment key scan function is used (KSON = 1)

<Key input wait>



Shaded sections: Segment key scan output period

<Key identification>



Shaded sections: Segment key scan output period

Remark During key identification, the residual charge of the LCD panel can be eliminated by outputting a signal with an inverted relationship between its first half and latter half.

18.7.3 Three-time-slice display example

Figure 18-24 shows how the 8-digit LCD panel having the display pattern shown in Figure 18-23 is connected to the segment signals (SEG0 to SEG23) and the common signals (COM0 to COM2) of the 78K0/LF3 chip. This example displays data "123456.78" in the LCD panel. The contents of the display data memory (addresses FA40H to FA57H) correspond to this display.

The following description focuses on numeral "6." (6.) displayed in the third digit. To display "6." in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG6 to SEG8 pins according to Table 18-8 at the timing of the common signals COM0 to COM2; see Figure 18-23 for the relationship between the segment signals and LCD segments.

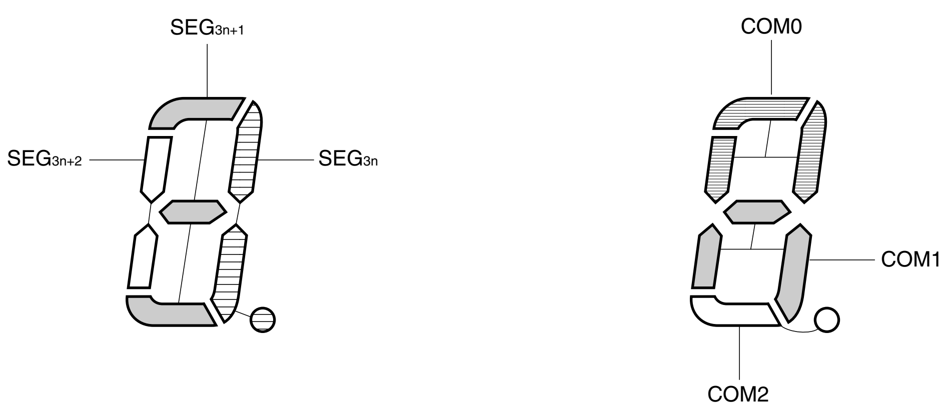
Table 18-8. Select and Deselect Voltages (COM0 to COM2)

Common \ Segment	SEG6	SEG7	SEG8
COM0	Deselect	Select	Select
COM1	Select	Select	Select
COM2	Select	Select	–

According to Table 18-8, it is determined that the display data memory location (FA46H) that corresponds to SEG6 must contain x110.

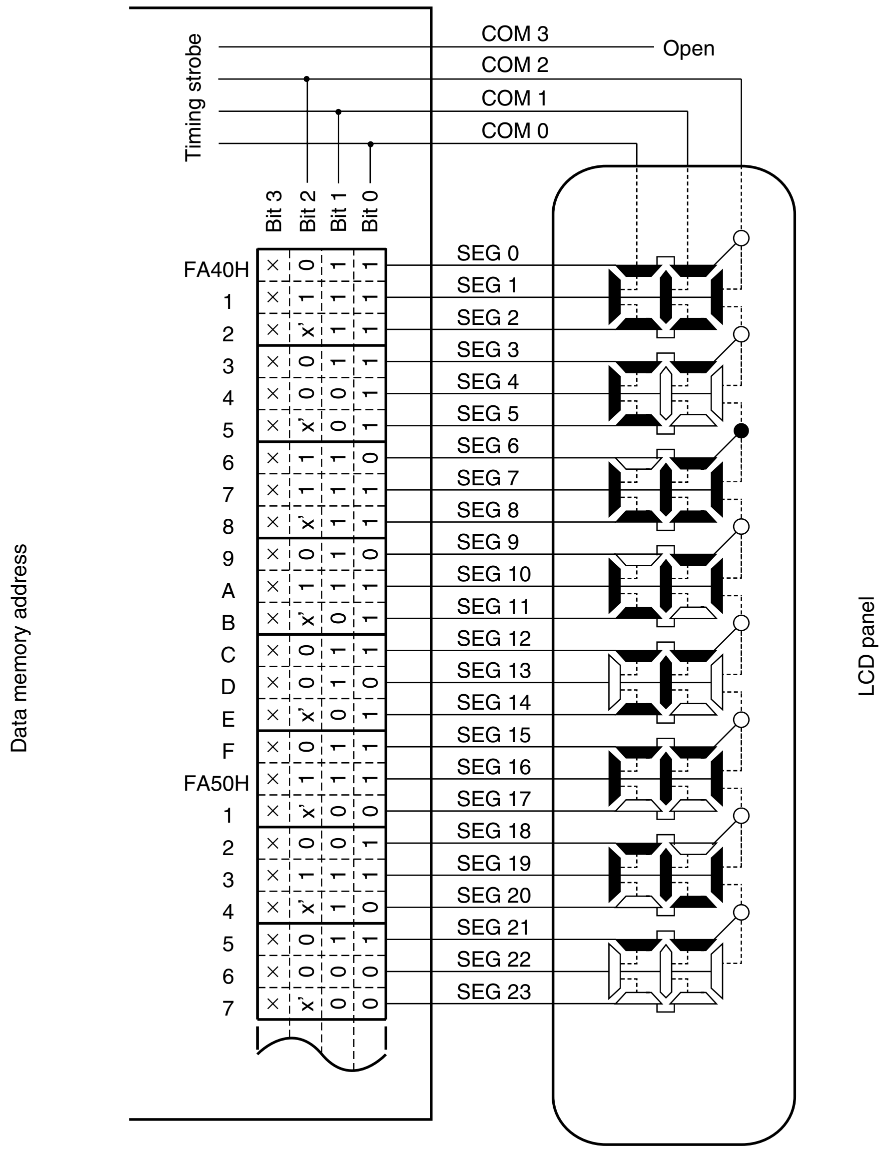
Figures 18-25 and 18-26 show examples of LCD drive waveforms between the SEG6 signal and each common signal in the 1/2 and 1/3 bias methods, respectively. When the select voltage is applied to SEG6 at the timing of COM1 or COM2, an alternate rectangle waveform, $+V_{LCD}/-V_{LCD}$, is generated to turn on the corresponding LCD segment.

Figure 18-23. Three-Time-Slice LCD Display Pattern and Electrode Connections



Remark n = 0 to 7

Figure 18-24. Example of Connecting Three-Time-Slice LCD Panel

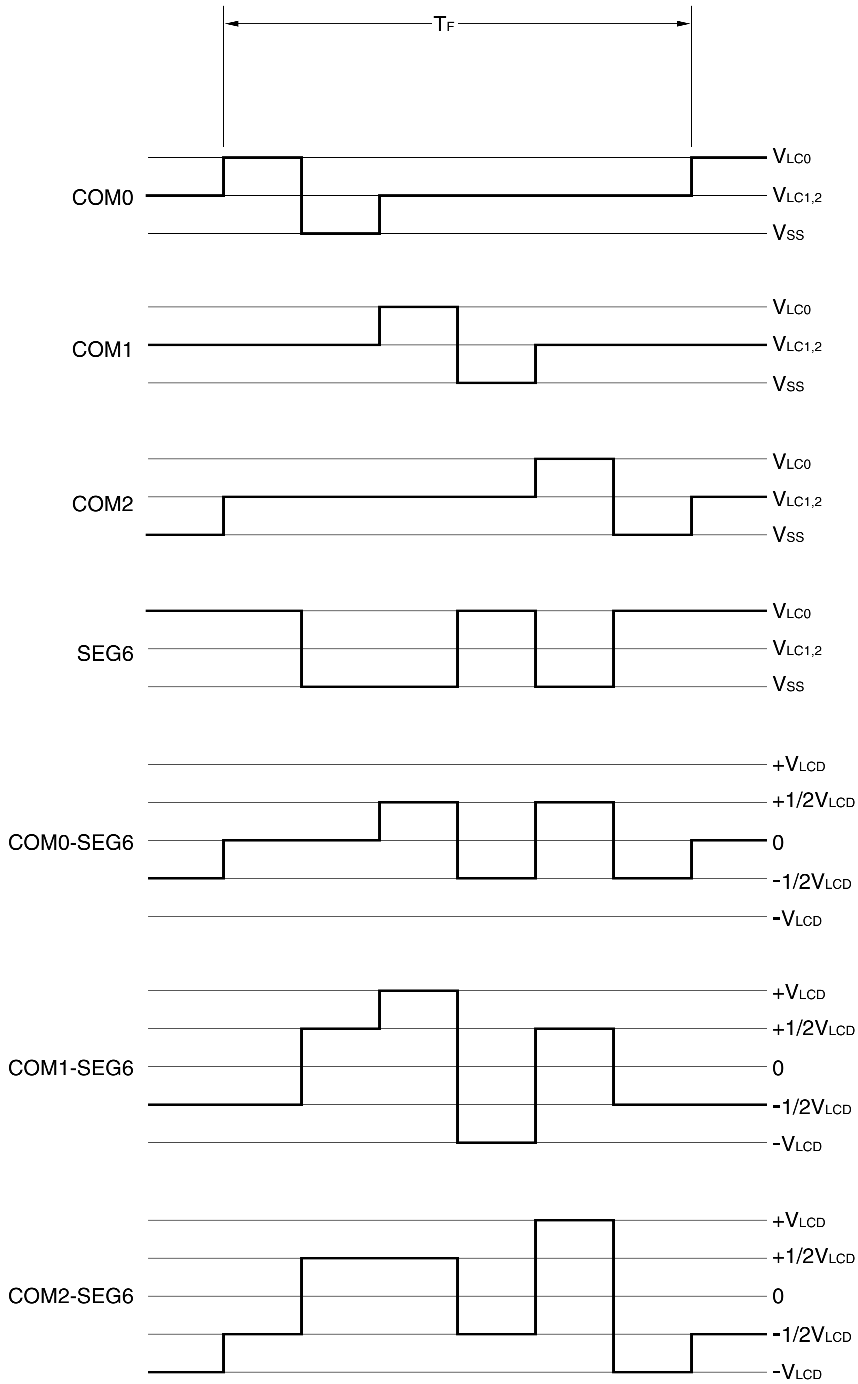


×': Can be used to store any data because there is no corresponding segment in the LCD panel.

×: Can always be used to store any data because the three-time-slice mode is being used.

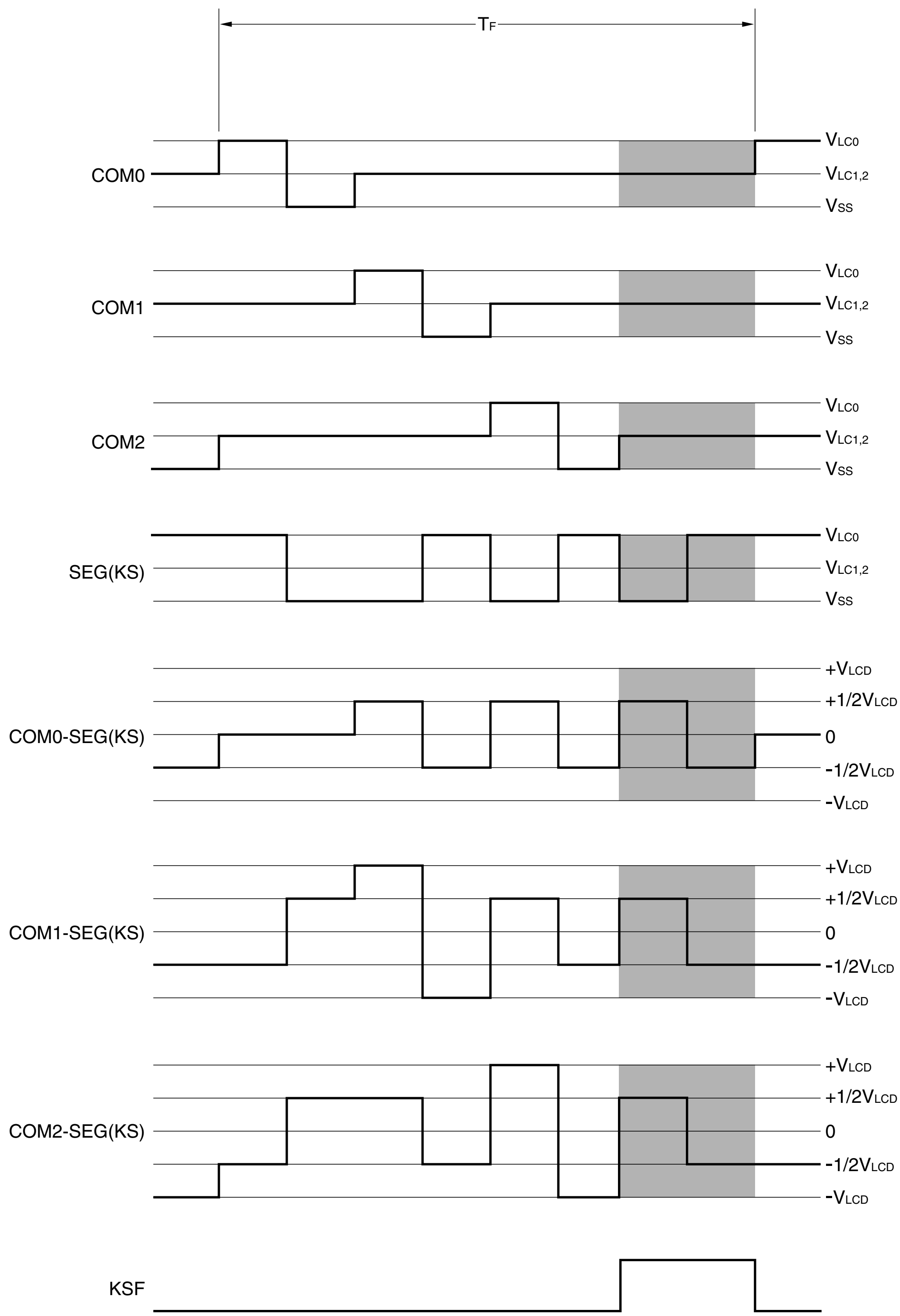
Figure 18-25. Three-Time-Slice LCD Drive Waveform Examples (1/2 Bias Method)

(a) When segment key scan function is not used (KSON = 0)



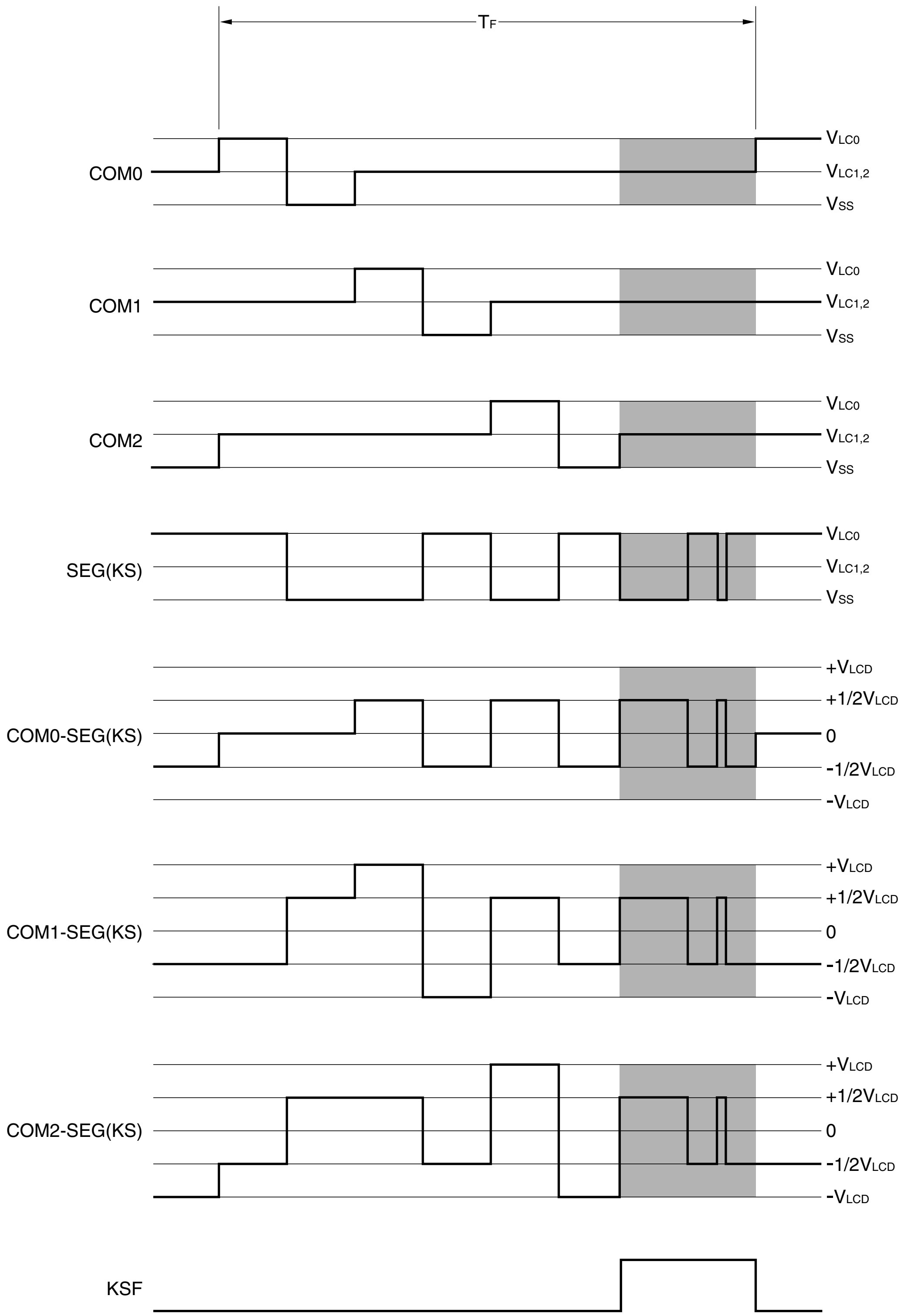
(b) When segment key scan function is used (KSON = 1)

<Key input wait>



Shaded sections: Segment key scan output period

<Key identification>

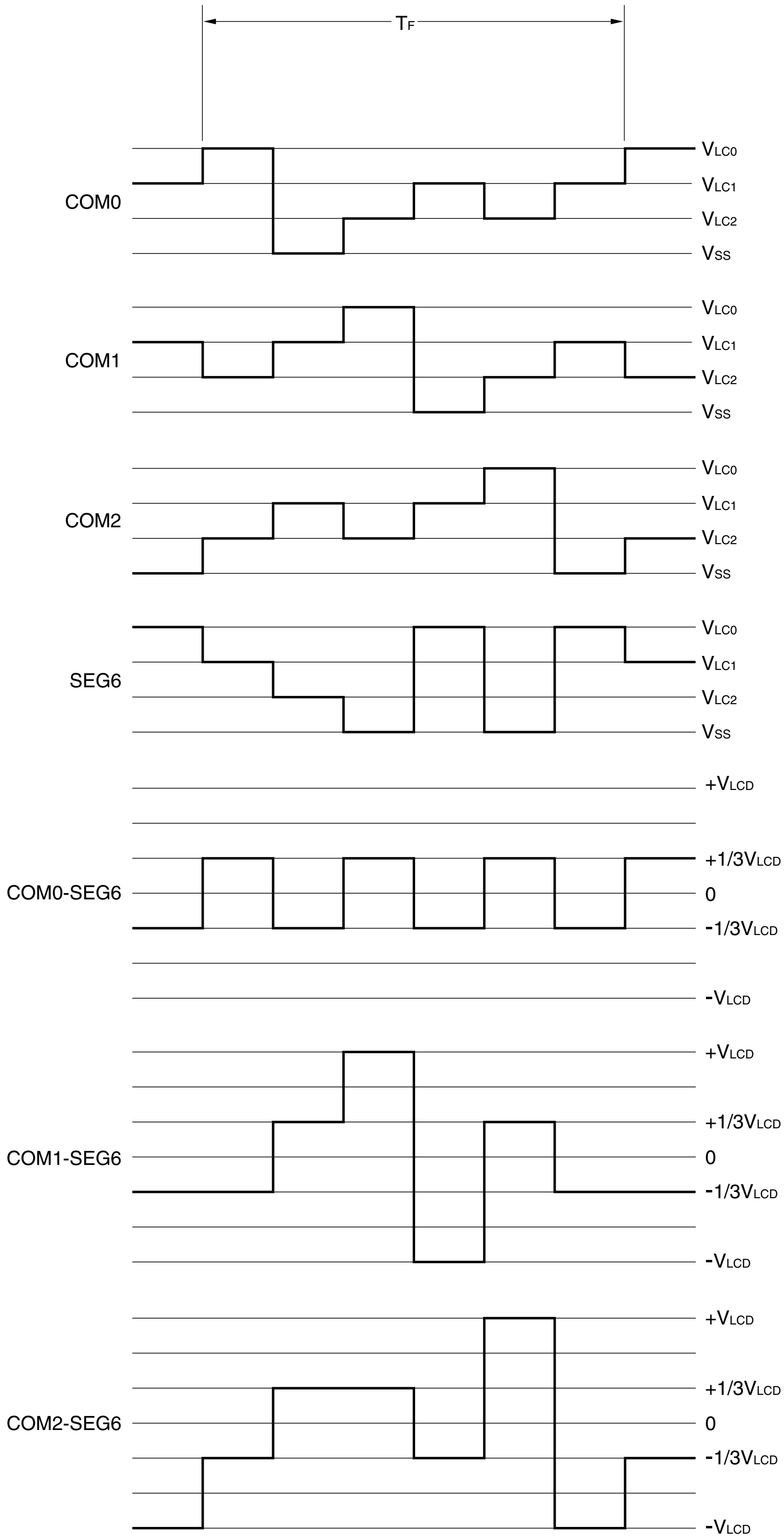


Shaded sections: Segment key scan output period

Remark During key identification, the residual charge of the LCD panel can be eliminated by outputting a signal with an inverted relationship between its first half and latter half.

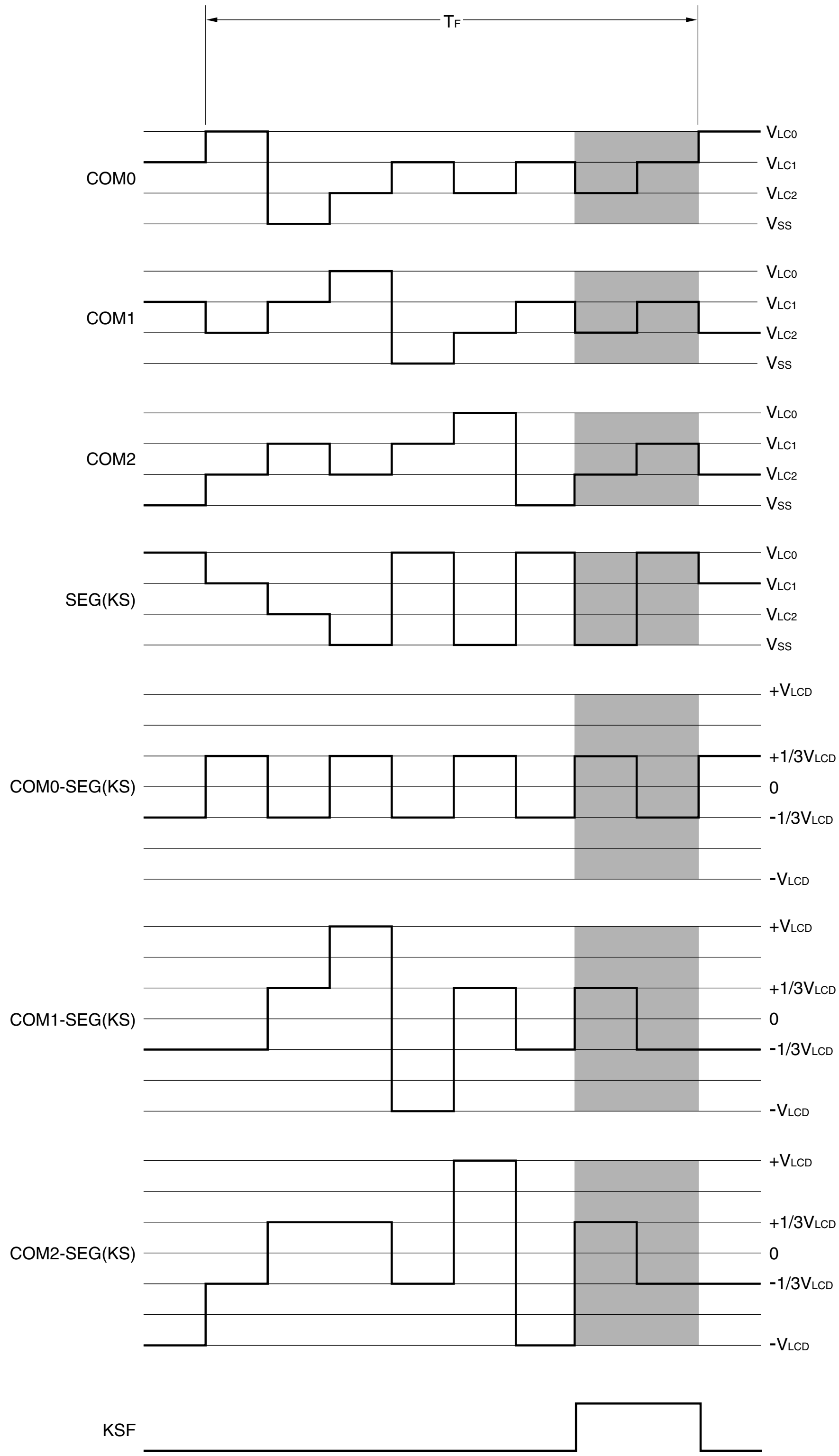
Figure 18-26. Three-Time-Slice LCD Drive Waveform Examples (1/3 Bias Method)

(a) When segment key scan function is not used (KSON = 0)



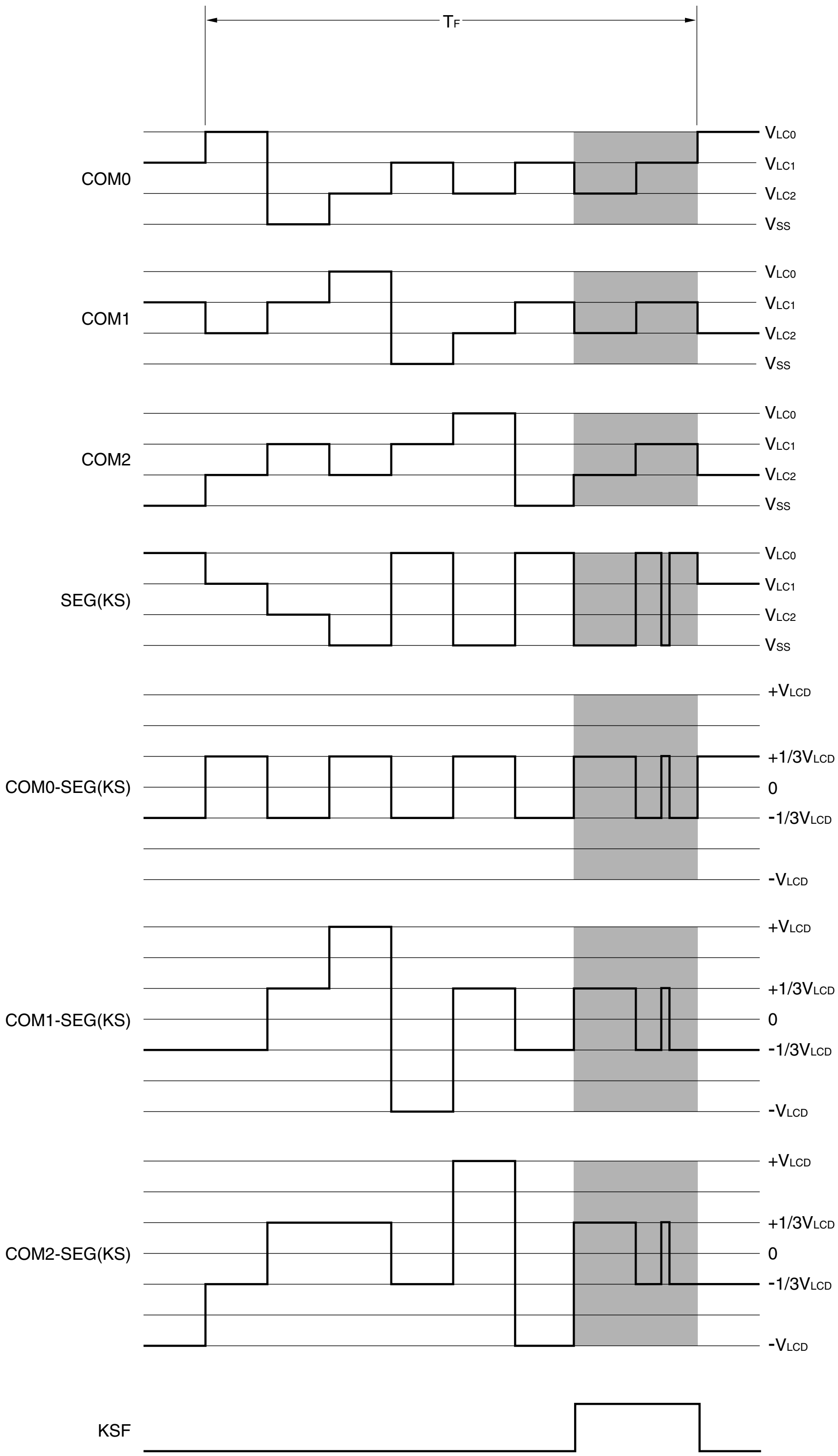
(b) When segment key scan function is used (KSON = 1)

<Key input wait>



Shaded sections: Segment key scan output period

<Key identification>



Shaded sections: Segment key scan output period

Remark During key identification, the residual charge of the LCD panel can be eliminated by outputting a signal with an inverted relationship between its first half and latter half.

18.7.4 Four-time-slice display example

Figure 18-28 shows how the 12-digit LCD panel having the display pattern shown in Figure 18-27 is connected to the segment signals (SEG0 to SEG23) and the common signals (COM0 to COM3) of the 78K0/LF3 chip. This example displays data "123456.789012" in the LCD panel. The contents of the display data memory (addresses FA40H to FA57H) correspond to this display.

The following description focuses on numeral "6." (6.) displayed in the seventh digit. To display "6." in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG12 and SEG13 pins according to Table 18-9 at the timing of the common signals COM0 to COM3; see Figure 18-27 for the relationship between the segment signals and LCD segments.

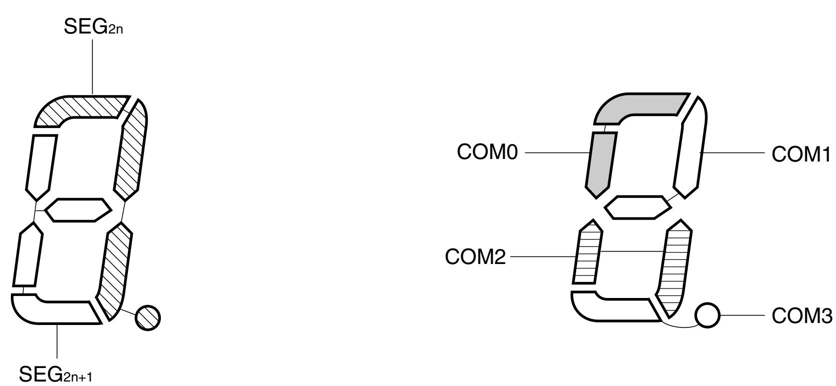
Table 18-9. Select and Deselect Voltages (COM0 to COM3)

Segment / Common	SEG12	SEG13
COM0	Select	Select
COM1	Deselect	Select
COM2	Select	Select
COM3	Select	Select

According to Table 18-9, it is determined that the display data memory location (FA4CH) that corresponds to SEG12 must contain 1101.

Figure 18-29 shows examples of LCD drive waveforms between the SEG12 signal and each common signal. When the select voltage is applied to SEG12 at the timing of COM0, an alternate rectangle waveform, $+V_{LCD}/-V_{LCD}$, is generated to turn on the corresponding LCD segment.

Figure 18-27. Four-Time-Slice LCD Display Pattern and Electrode Connections



Remark n = 0 to 11

Figure 18-28. Example of Connecting Four-Time-Slice LCD Panel

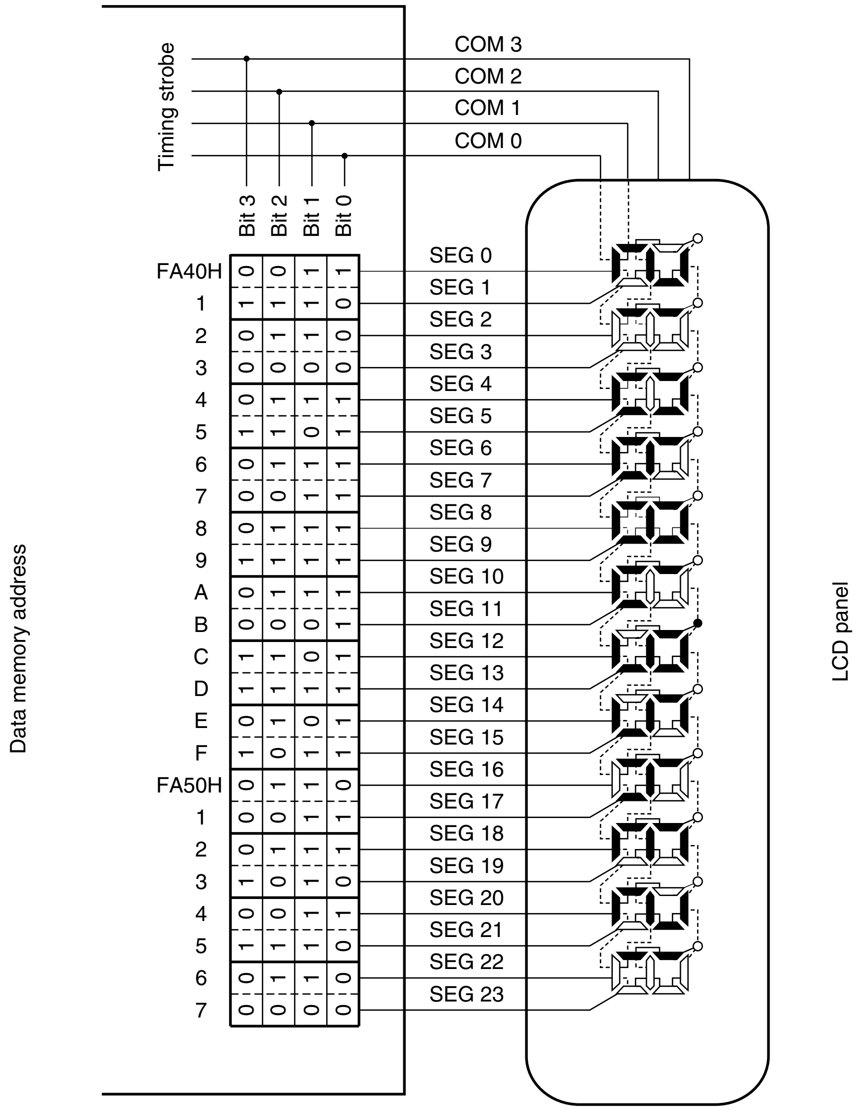
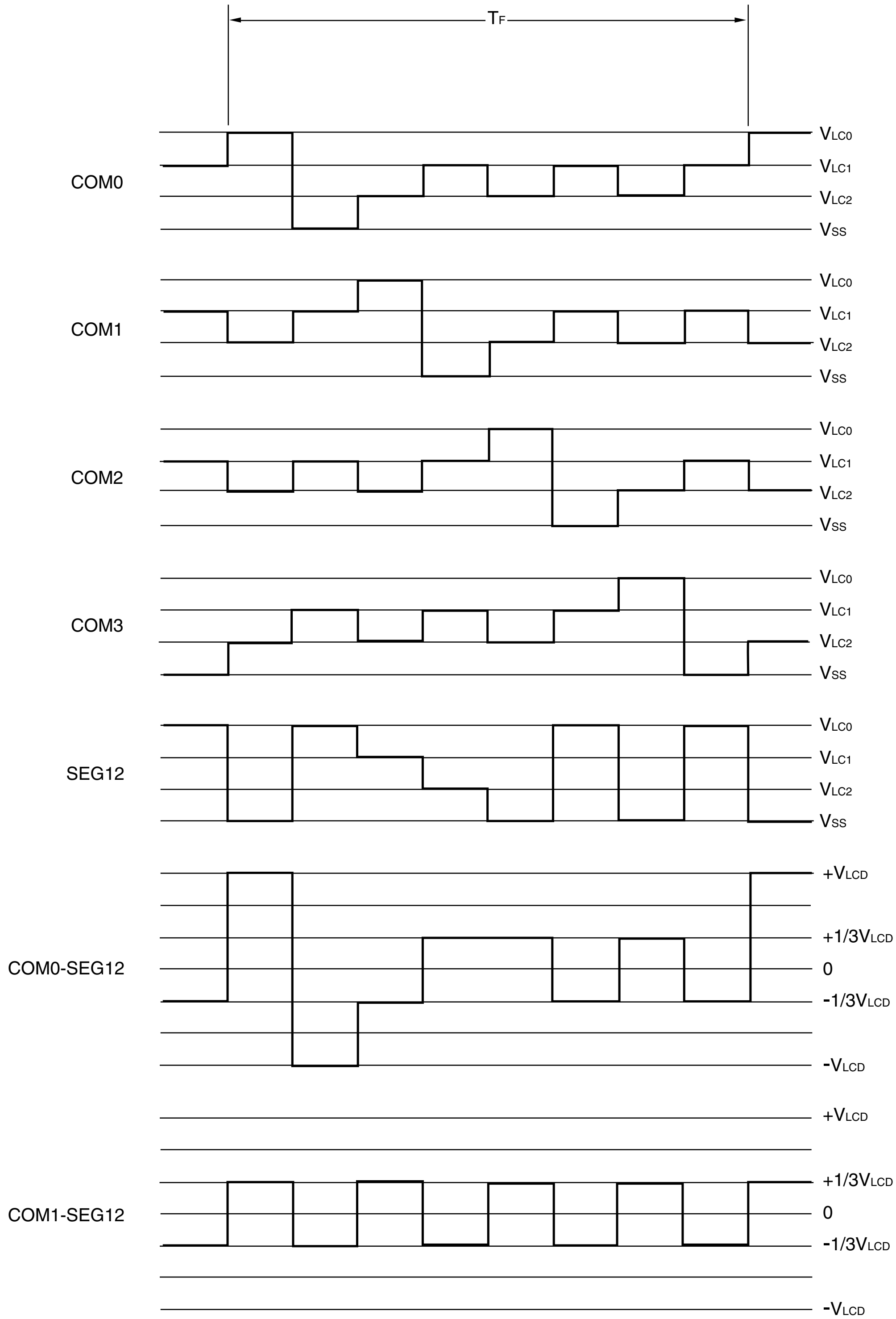


Figure 18-29. Four-Time-Slice LCD Drive Waveform Examples (1/3 Bias Method)

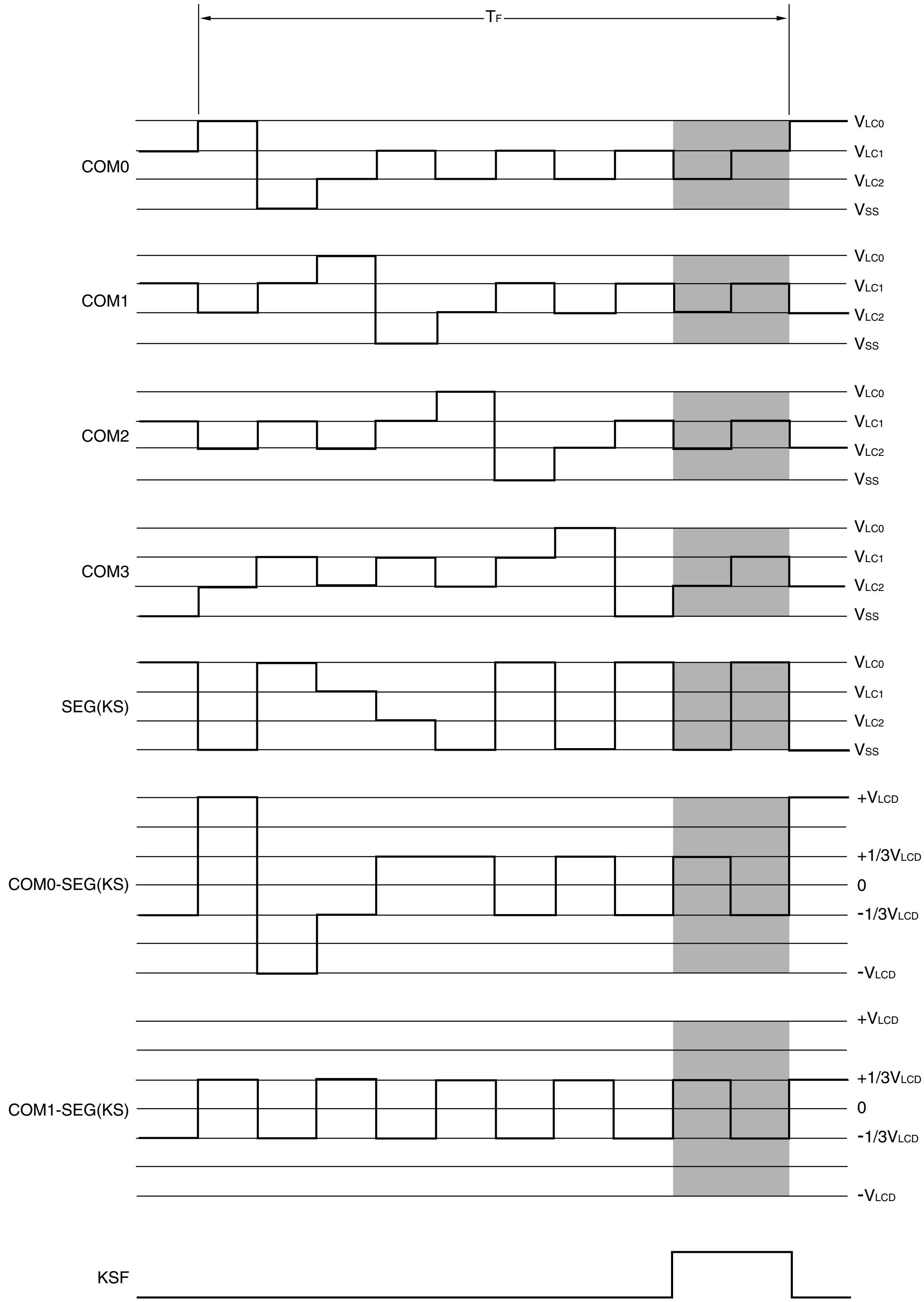
(a) When segment key scan function is not used (KSON = 0)



Remark The waveforms for COM2 to SEG12 and COM3 to SEG12 are omitted.

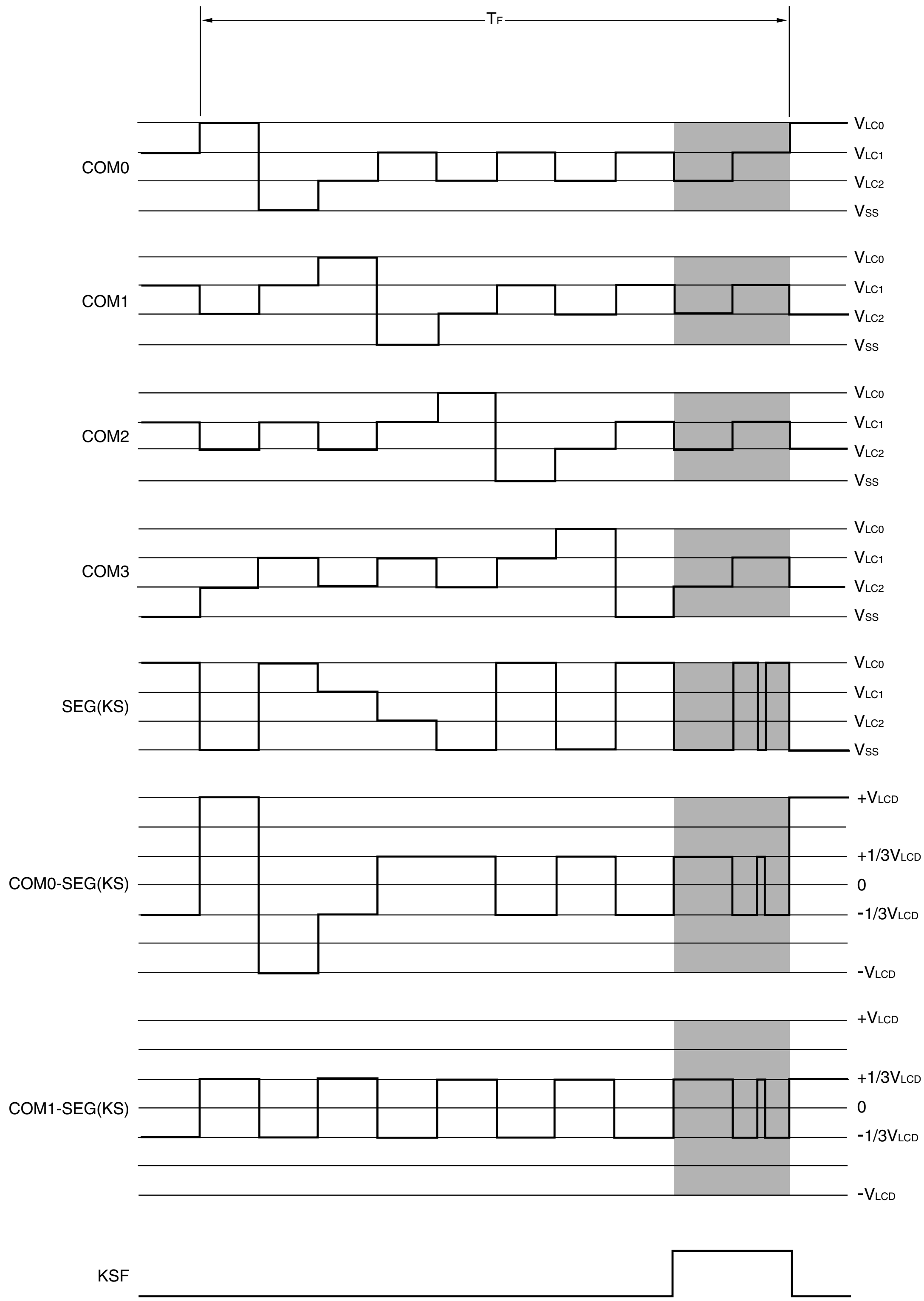
(b) When segment key scan function is used (KSON = 1)

<Key input wait>



Shaded sections: Segment key scan output period

<Key identification>



Shaded sections: Segment key scan output period

Remark During key identification, the residual charge of the LCD panel can be eliminated by outputting a signal with an inverted relationship between its first half and latter half.

18.7.5 Eight-time-slice display example

Figure 18-31 shows how the 15×8 dots LCD panel having the display pattern shown in Figure 18-30 is connected to the segment signals (SEG4 to SEG18) and the common signals (COM0 to COM7) of the 78K0/LF3 chip. This example displays data "123" in the LCD panel. The contents of the display data memory (addresses FA44H to FA52H) correspond to this display.

The following description focuses on numeral "3." (∃) displayed in the first digit. To display "3." in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG4 and SEG8 pins according to Table 18-10 at the timing of the common signals COM0 to COM7; see Figure 18-30 for the relationship between the segment signals and LCD segments.

Table 18-10. Select and Deselect Voltages (COM0 to COM7)

Segment / Common	SEG4	SEG5	SEG6	SEG7	SEG8
COM0	Select	Select	Select	Select	Select
COM1	Deselect	Select	Deselect	Deselect	Deselect
COM2	Deselect	Deselect	Select	Deselect	Deselect
COM3	Deselect	Select	Deselect	Deselect	Deselect
COM4	Select	Deselect	Deselect	Deselect	Deselect
COM5	Select	Deselect	Deselect	Deselect	Select
COM6	Deselect	Select	Select	Select	Deselect
COM7	Deselect	Deselect	Deselect	Deselect	Deselect

According to Table 18-10, it is determined that the display data memory location (FA44H) that corresponds to SEG4 must contain 00110001.

Figure 18-32 shows examples of LCD drive waveforms between the SEG4 signal and each common signal. When the select voltage is applied to SEG4 at the timing of COM0, a waveform is generated to turn on the corresponding LCD segment.

Figure 18-30. Eight-Time-Slice LCD Display Pattern and Electrode Connections

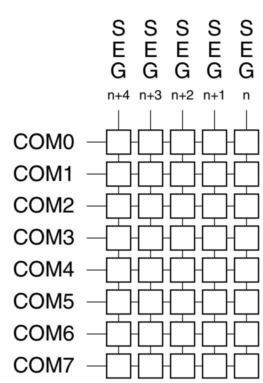


Figure 18-31. Example of Connecting Eight-Time-Slice LCD Panel

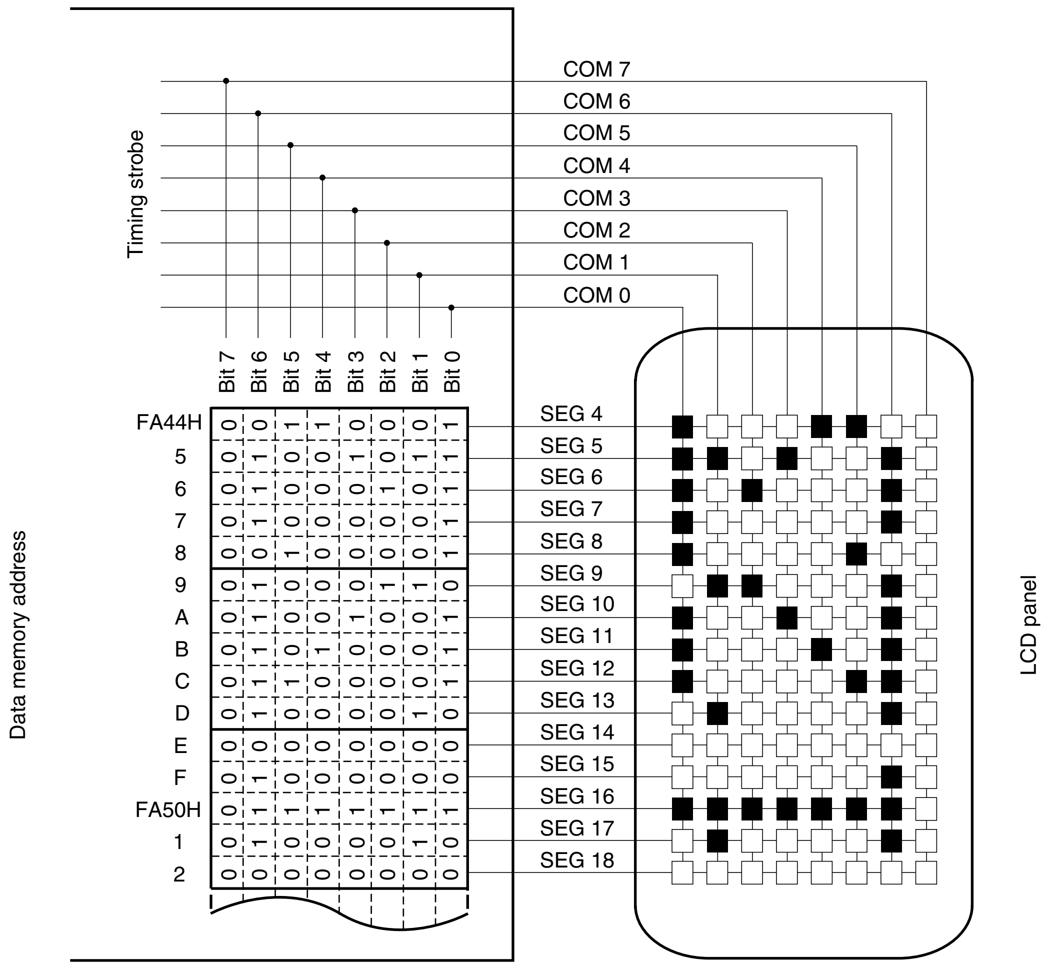
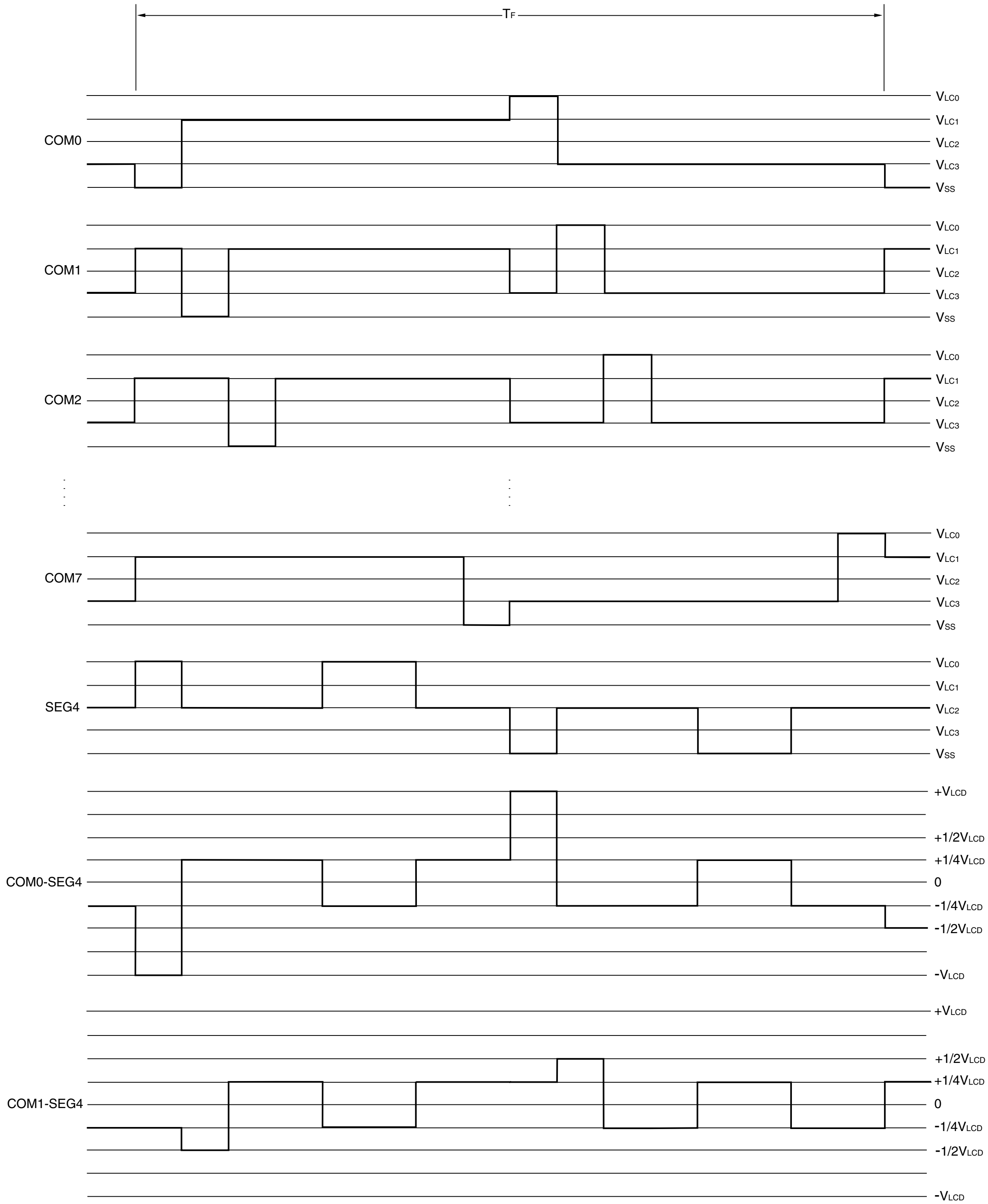


Figure 18-32. Eight-Time-Slice LCD Drive Waveform Examples (1/4 Bias Method)

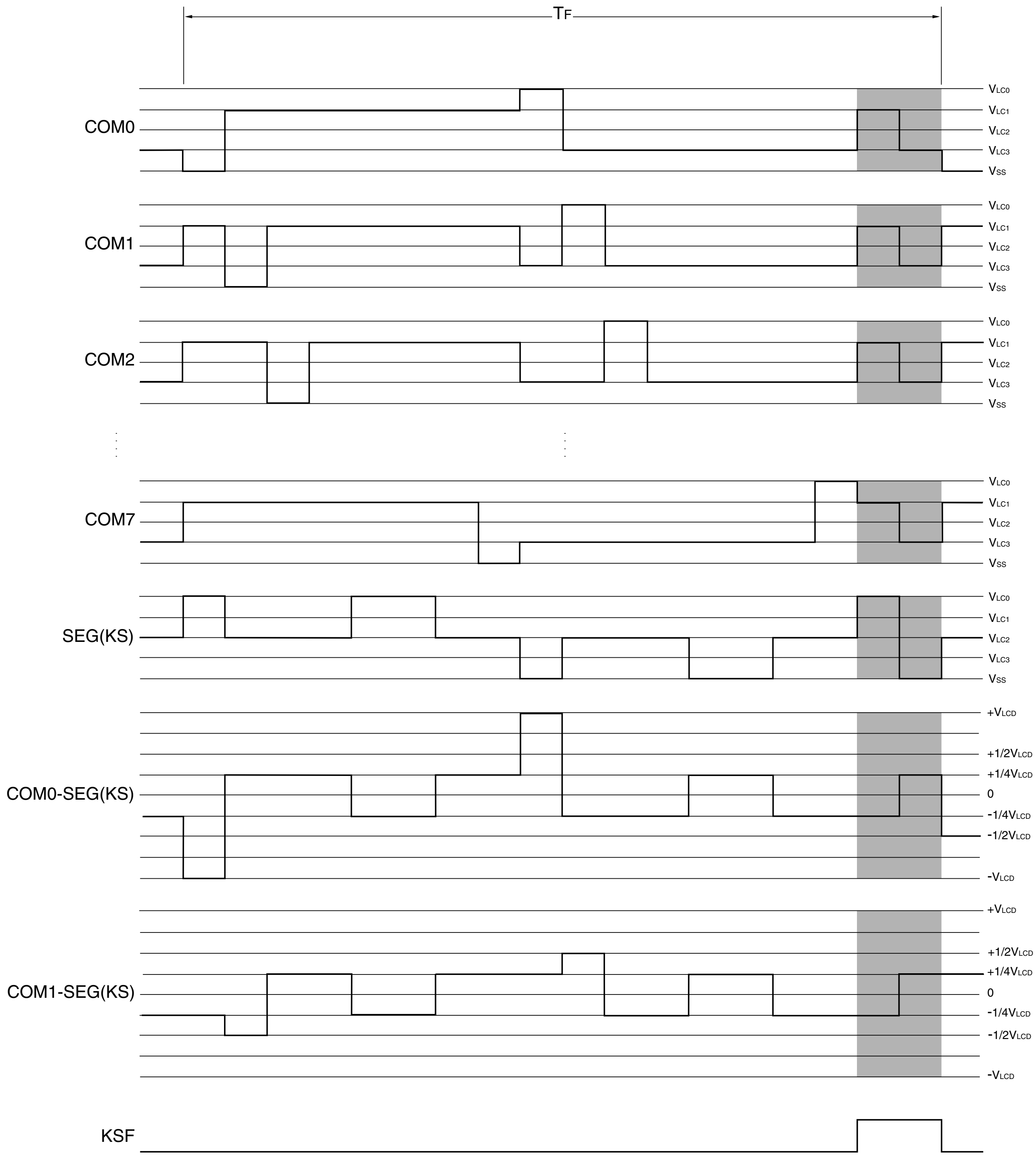
(a) When segment key scan function is not used (KSON = 0)



Remark The waveforms for COM3 to COM6, COM2 to SEG4 and COM7 to SEG4 are omitted.

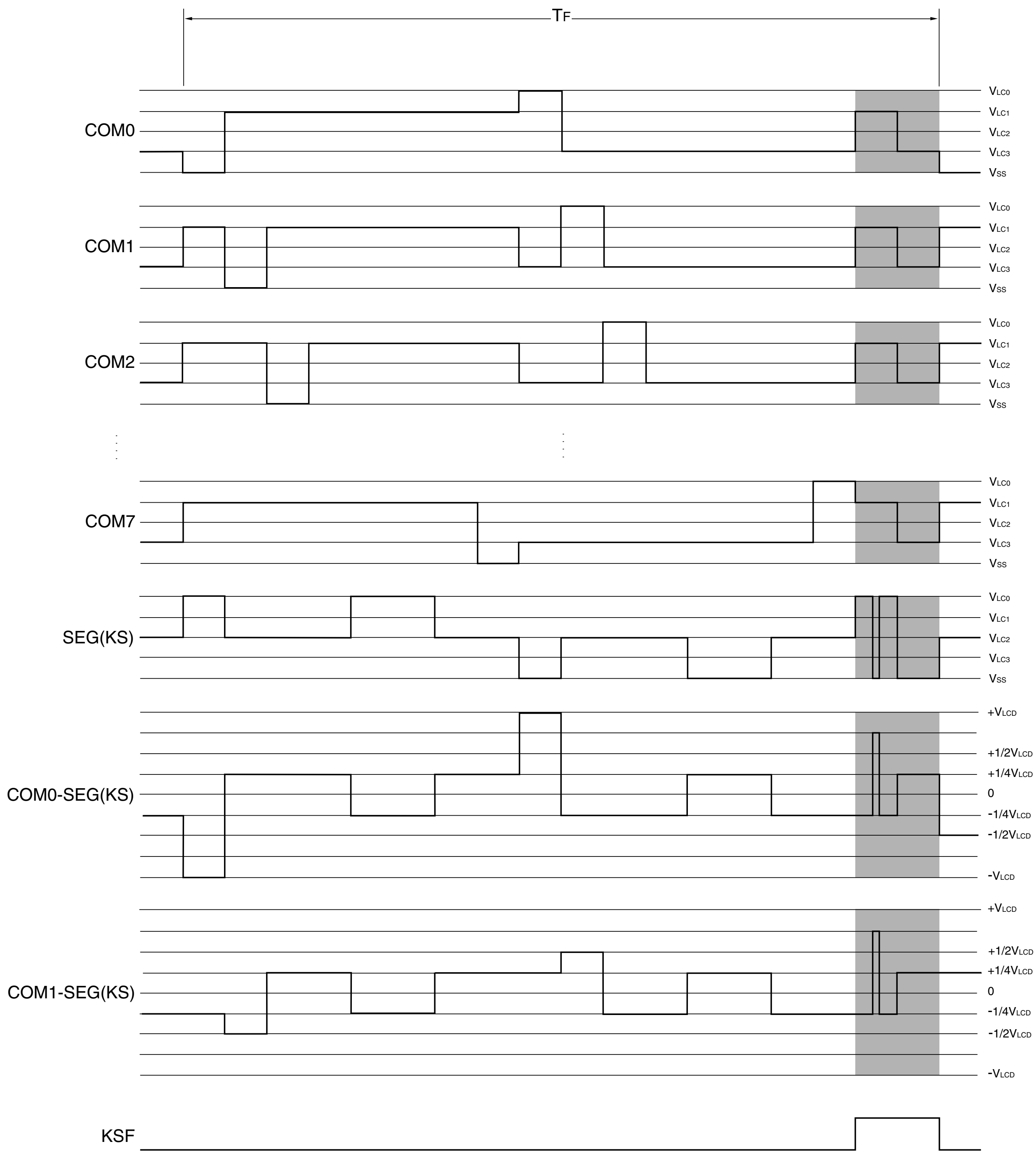
(b) When segment key scan function is used (KSON = 1)

<Key input wait>



Shaded sections: Segment key scan output period

<Key identification>



Shaded sections: Segment key scan output period

Remark During key identification, the residual charge of the LCD panel can be eliminated by outputting a signal with an inverted relationship between its first half and latter half.

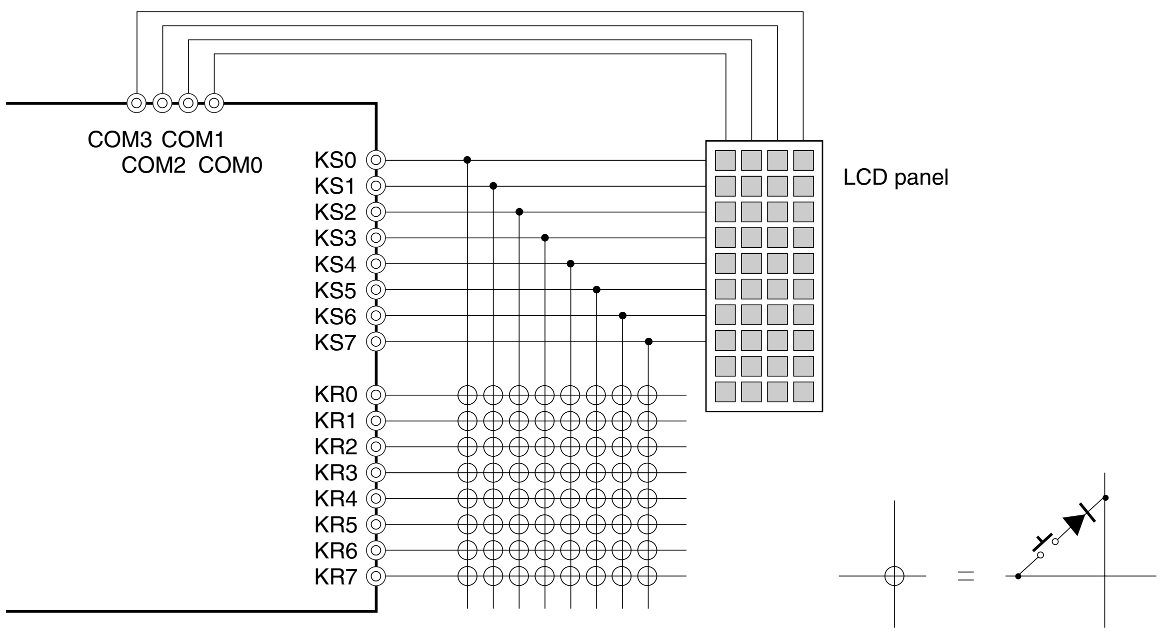
18.8 Operation of Segment Key Scan Function

The segment key scan function is used to reduce the number of pins used by outputting LCD display segment output and key scan signals from the same pin.

**Caution This function may affect the LCD panel, depending on how it is used.
Use the function after thorough evaluation.**

18.8.1 Circuit configuration example

Figure 18-33. Circuit configuration example



18.8.2 Example of procedure for using segment key scan function

Figure 18-34 shows the operation flow of the segment key scan and Figure 18-35 shows the key connection example.

Figure 18-34. Operation Flow of Segment Key Scan

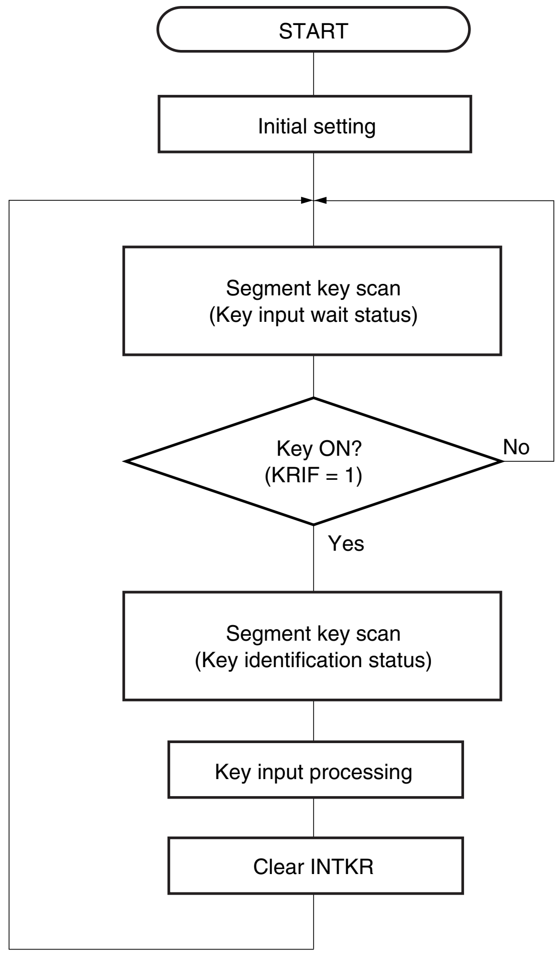
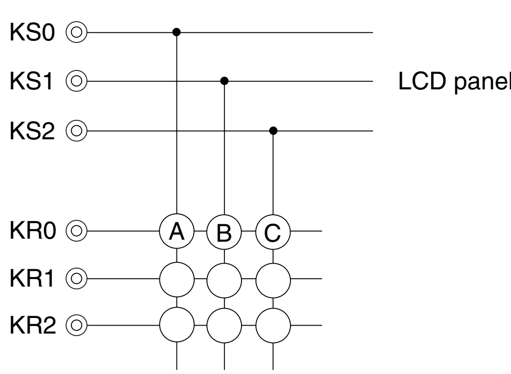
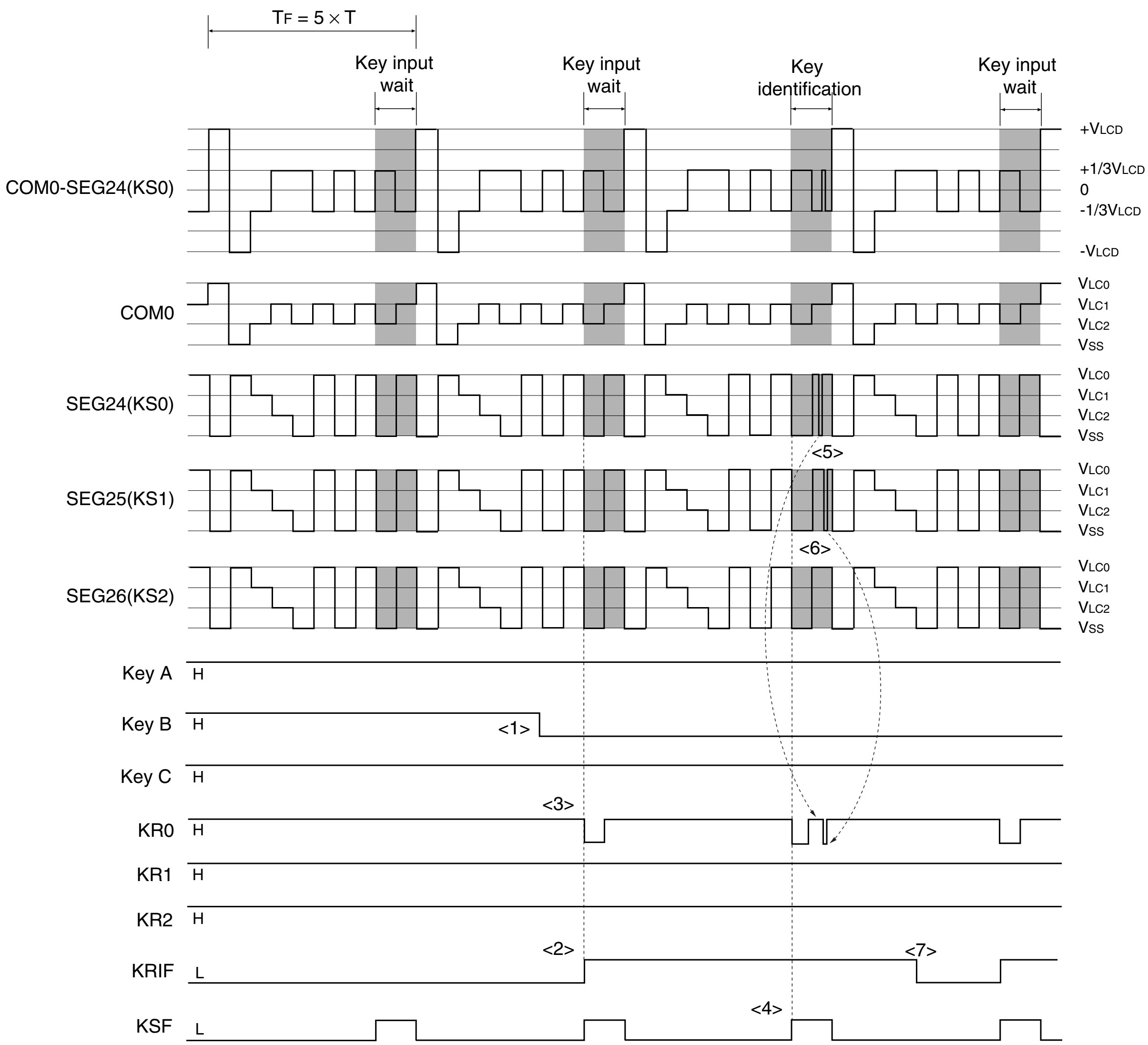


Figure 18-35. Key Connection Example



An example of the segment key scan operation when key B, shown in Figure 18-35, has been pressed is shown below.

Figure 18-36. Example of Segment Key Scan Operation Timing (Four-Time-Slice (1/3 Bias Method))



T: One LCD clock period T_F: Frame frequency Shaded sections: Segment key scan output period

<1> Assume key B has been pressed at this timing.

<2> KRIF becomes “1” and the key that has been pressed can be known.

<3> KR0 becomes low level, and whether key A, B, or C has been pressed can be known.
Input to the KR pin will be enabled after two f_{LCD} clocks from the rise of KSF. Consequently, KRIF becomes “1” after two f_{LCD} clocks from the rise of KSF.

<4> A segment key scan operation is started after it has been confirmed that KSF is “1”.

<5> It can be known that key A is not pressed, because KR0 was at high level when the SEG24 (KS0) pin outputs a low level.

<6> It can be known that key B is pressed, because KR0 was at low level when the SEG25 (KS1) pin outputs a low level. Perform the input processing of key B.

<7> Clear KRIF.

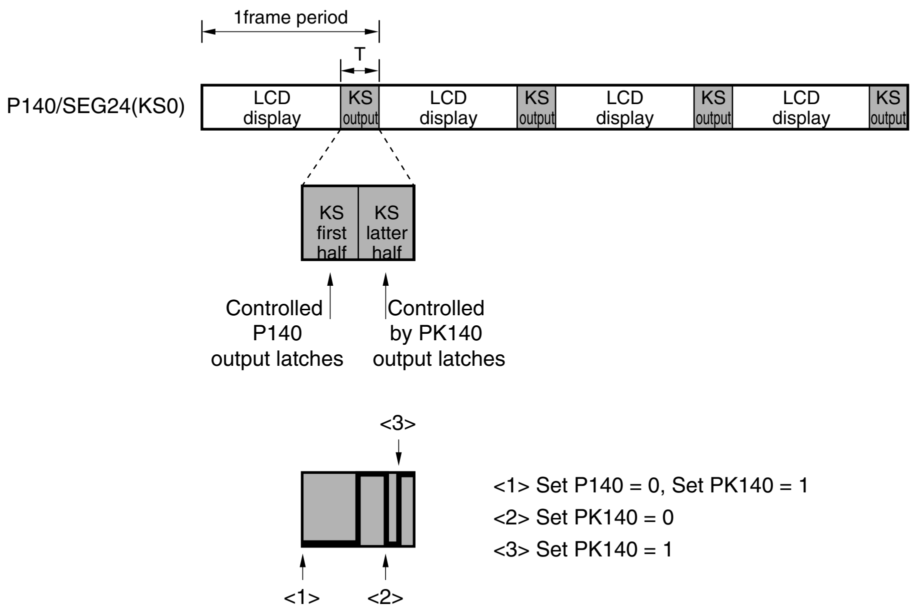
The output values of the SEG (KS) pin during the segment key scan output period correspond to the setting values of port registers 14 and 15, and can be controlled by using port registers 14 and 15.

Bits 0 to 3 and bits 4 to 7 of each port register are used to control the first half and latter half of the segment key scan output period, respectively (see **(9) Port register 14 (P14)** and **(10) Port register 15 (P15)** in **18.3**).

Figure 18-37 shows the relationship between port register 14 and the segment key scan output.

Figure 18-37. Relationship Between Port Register 14 and Segment Key Scan Output (for P140/SEG24(KS0) Pin)

<KS0N = 1>



T : For one period of the LCD clock
LCD display : LCD indication signal output period
KS output : Segment key scan output period

Remark During the segment key scan output period, COM will not be displayed when output.
See **Figures 18-15** and **18-16** for waveform details.

18.9 Cautions When Using Segment Key Scan Function

(1) Conditions for use

Use the segment key scan function if V_{DD} is equal to V_{LC0}.

(2) Segment key scan input pins

Only the KR0 to KR7 pins can be used as input pins for the segment key scan function.

Other pins cannot be used as input pins for the segment key scan function.

(3) Allowable input range of KR0 to KR7 pins

Due to a delay caused by a pull-up resistor, segment key scan input cannot be performed for the KR pin for a period of two f_{LCD} clocks from the start of the segment key scan output period.

Similarly, due to input end processing, segment key scan input cannot be performed for the KR pin for the period of the last f_{LCD} clock of the segment key scan output period.

(4) Key return mode register (KRM) setting

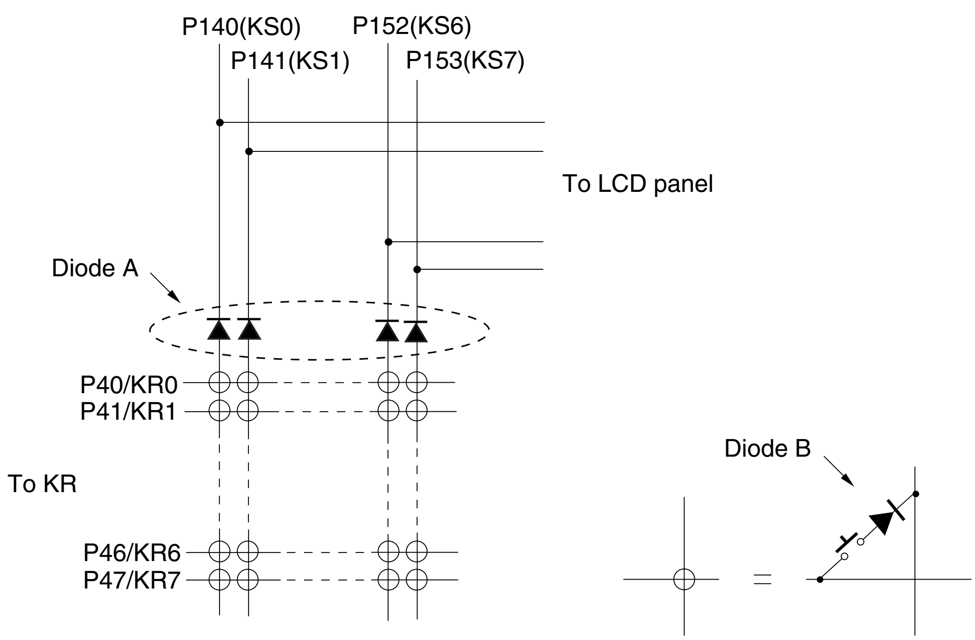
When the segment key scan function is used (KSON = 1), set KRMn to 1 or 0 to use or not use the KRn pin as a segment key scan input pin.

(5) Circuit configuration

When using the segment key scan function, at least diode A or diode B shown in Figure 18-38 is required.

The following problems will occur when diodes A and B are missing.

Figure 18-38. Key Matrix Configuration Example



(a) When both diodes A and B are missing

When both diodes A and B are missing, the segment key scan function cannot be used, because of the following.

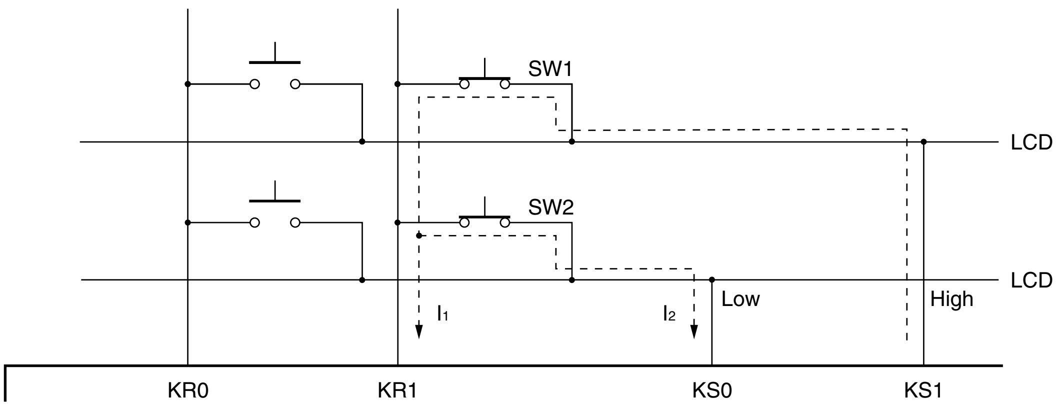
The following figure shows a circuit example when both diodes A and B are missing.

Assume, as shown in the figure below, that switches SW1 and SW2 are turned on, and a high level and a low level are output from the KS1 pin and KS0 pin, respectively.

When diode A is missing at this time, currents I_1 and I_2, shown as broken lines, will flow.

Consequently, the high level of KS1 and the low level of KS0 will not be output normally due to I_2, and the key input data of KR1 will become undefined.

Furthermore, the LCD display will not be turned on or off normally.



(b) When only diode A exists

When only diode A exists, whether switches are pressed simultaneously cannot be identified, due to the following.

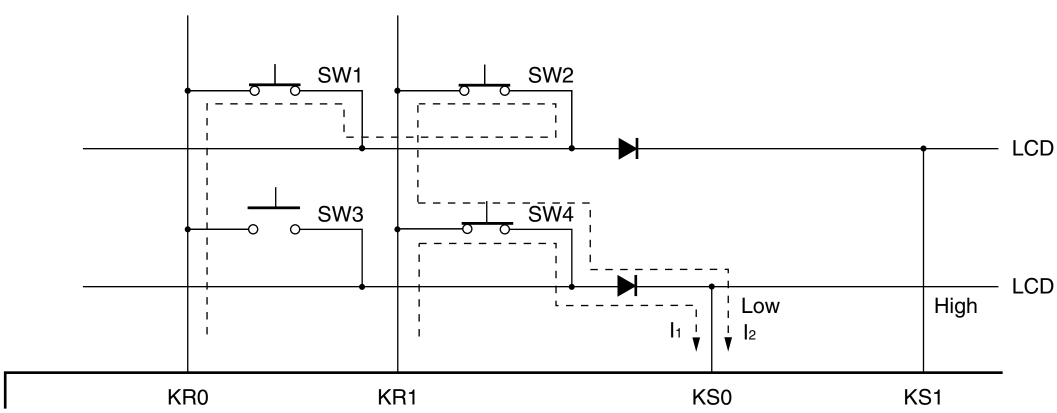
The following figure shows a circuit example when only diode A exists.

Assume, as shown in the figure below, that switches SW1, SW2, and SW4 are turned on, and a high level and a low level are output from the KS1 pin and KS0 pin, respectively.

At this time currents I_1 and I_2, shown as broken lines, will flow.

Consequently, even though SW3 is turned off, SW3 is identified to be turned on, because a low level is input to KR0 due to I_2.

There is no interference with the LCD display.



(c) When only diode B exists

Identification of at least three switches being pressed simultaneously can be performed.

There is no interference with the LCD display.

18.10 Supplying LCD Drive Voltages V_{LC0}, V_{LC1}, V_{LC2}, and V_{LC3}

With the 78K0/LF3, a LCD drive power supply can be generated using either of two types of methods: internal resistance division method or external resistance division method.

18.10.1 Internal resistance division method

The 78K0/LF3 incorporates voltage divider resistors for generating LCD drive power supplies. Using internal voltage divider resistors, a LCD drive power supply that meet each bias method listed in Table 18-11 can be generated, without using external voltage divider resistors.

Table 18-11. LCD Drive Voltages (with On-Chip Voltage Divider Resistors)

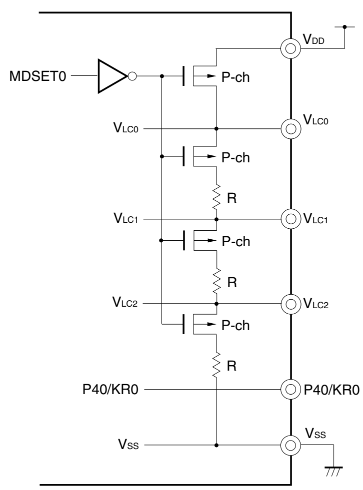
Bias Method / LCD Drive Voltage Pin	No Bias (Static)	1/2 Bias Method	1/3 Bias Method	1/4 Bias Method
V_{LC0}	V_{LCD}	V_{LCD}	V_{LCD}	V_{LCD}
V_{LC1}	$\frac{2}{3}V_{LCD}$	$\frac{1}{2}V_{LCD}$ Note	$\frac{2}{3}V_{LCD}$	$\frac{3}{4}V_{LCD}$
V_{LC2}	$\frac{1}{3}V_{LCD}$		$\frac{1}{3}V_{LCD}$	$\frac{2}{4}V_{LCD}$
V_{LC3}	V_{SS}	V_{SS}	V_{SS}	$\frac{1}{4}V_{LCD}$

Note For the 1/2 bias method, it is necessary to connect the V_{LC1} and V_{LC2} pins externally.

Figure 18-39 shows examples of generating LCD drive voltages internally according to Table 18-11.

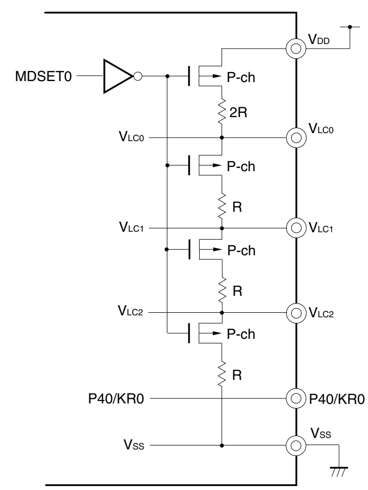
Figure 18-39. Examples of LCD Drive Power Connections (Internal Resistance Division Method) (1/2)

(a) 1/3 bias method and static display mode (MDSET1, MDSET0 = 0, 1) (example of V_{DD} = 5 V, V_{LC0} = 5 V)



$V_{LC0} = V_{DD}$

(b) 1/3 bias method and static display mode (MDSET1, MDSET0 = 1, 1) (example of V_{DD} = 5 V, V_{LC0} = 3 V)

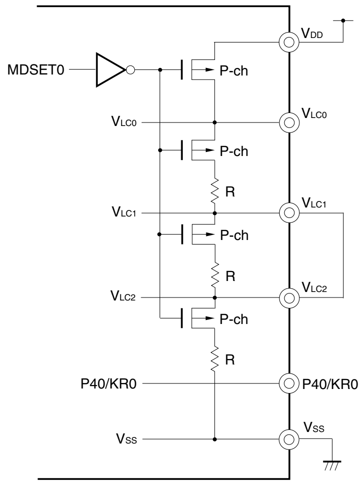


$V_{LC0} = \frac{3}{5} V_{DD}$

Remark It is recommended to use the external resistance division method when using the static display mode, in order to reduce power consumed by the voltage divider resistor.

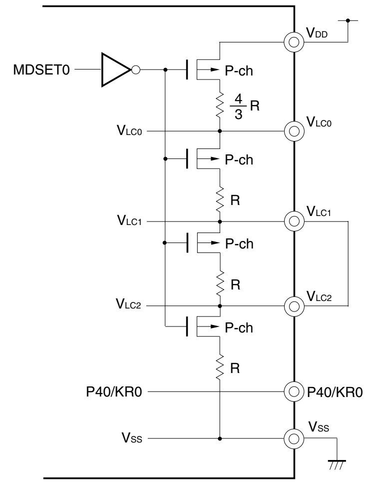
Figure 18-39. Examples of LCD Drive Power Connections (Internal Resistance Division Method) (2/2)

(c) 1/2 bias method
(MDSET1, MDSET0 = 0, 1)
(example of V_{DD} = 5 V, V_{LC0} = 5 V)



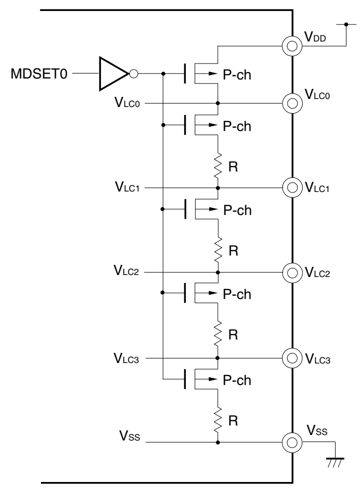
$V_{LC0} = V_{DD}$

(d) 1/4 bias method
(MDSET1, MDSET0 = 1, 1)
(example of V_{DD} = 5 V, V_{LC0} = 3 V)



$V_{LC0} = \frac{3}{5} V_{DD}$

(e) 1/4 bias method
(MDSET1, MDSET0 = 0, 1)
(example of V_{DD} = 5 V, V_{LC0} = 5 V)



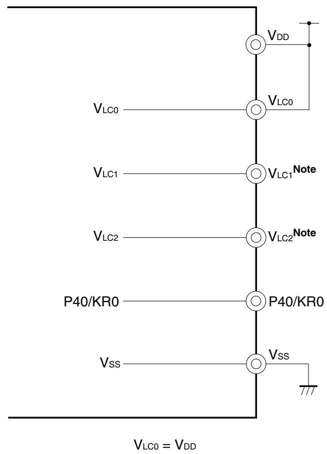
$V_{LC0} = V_{DD}$

18.10.2 External resistance division method

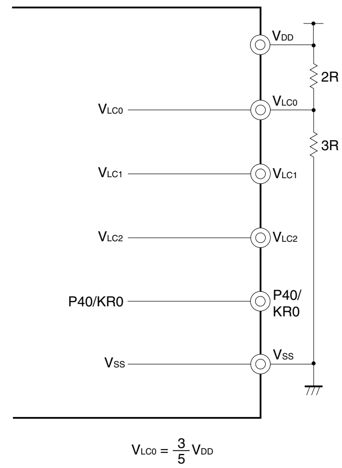
The 78K0/LF3 can also use external voltage divider resistors for generating LCD drive power supplies, without using internal resistors. Figure 18-40 shows examples of LCD drive voltage connection, corresponding to each bias method.

Figure 18-40. Examples of LCD Drive Power Connections (External Resistance Division Method) (1/2)

(a) Static display mode
(MDSET1, MDSET0 = 0, 0)
(example of V_{DD} = 5 V, V_{LC0} = 5 V)

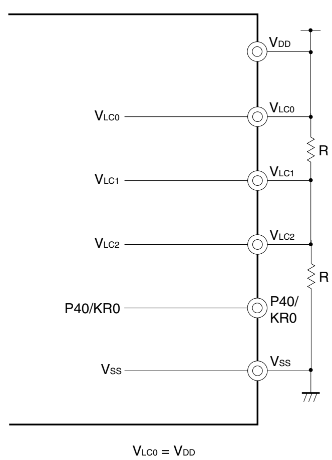


(b) Static display mode
(MDSET1, MDSET0 = 0, 0)
(example of V_{DD} = 5 V, V_{LC0} = 3 V)



Note Connect V_{LC1} and V_{LC2} directly to GND or V_{LC0}.

(c) 1/2 bias method
(MDSET1, MDSET0 = 0, 0)
(example of V_{DD} = 5 V, V_{LC0} = 5 V)



(d) 1/2 bias method
(MDSET1, MDSET0 = 0, 0)
(example of V_{DD} = 5 V, V_{LC0} = 3 V)

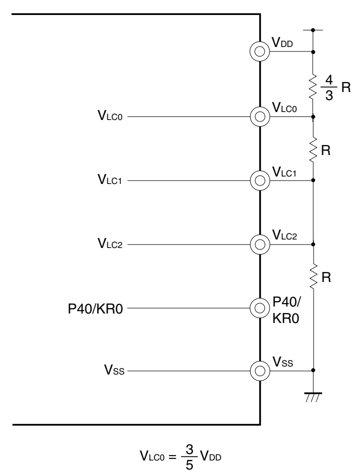
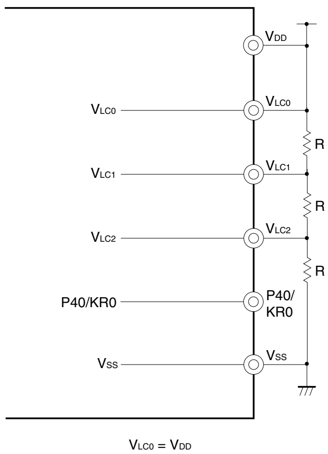
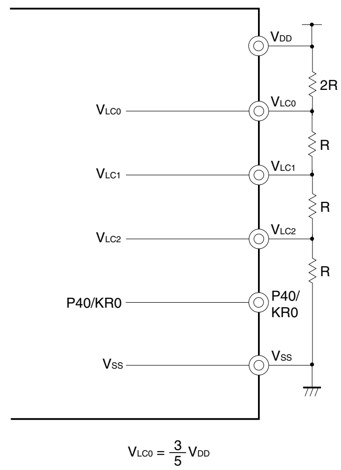


Figure 18-40. Examples of LCD Drive Power Connections (External Resistance Division Method) (2/2)

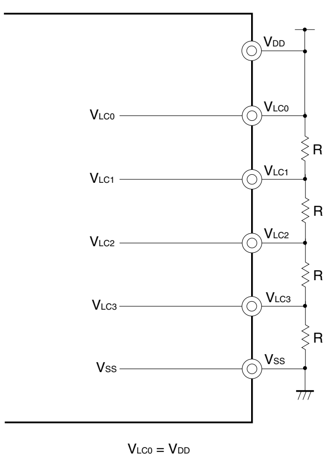
(e) 1/3 bias method
(MDSET1, MDSET0 = 0, 0)
(example of V_{DD} = 5 V, V_{LC0} = 5 V)



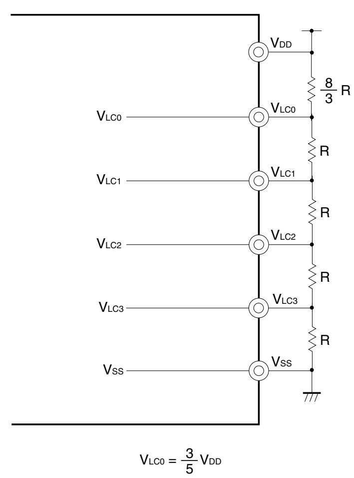
(f) 1/3 bias method
(MDSET1, MDSET0 = 0, 0)
(example of V_{DD} = 5 V, V_{LC0} = 3 V)



(g) 1/4 bias method
(MDSET1, MDSET0 = 0, 0)
(example of V_{DD} = 5 V, V_{LC0} = 5 V)



(h) 1/4 bias method
(MDSET1, MDSET0 = 0, 0)
(example of V_{DD} = 5 V, V_{LC0} = 3 V)



CHAPTER 19 MANCHESTER CODE GENERATOR

19.1 Functions of Manchester Code Generator

The following three types of modes are available for the Manchester code generator.

(1) Operation stop mode

This mode is used when output by the Manchester code generator/bit sequential buffer is not performed. This mode reduces the power consumption.

For details, refer to **19.4.1 Operation stop mode**.

(2) Manchester code generator mode

This mode is used to transmit Manchester code from the MCGO pin.

The transfer bit length can be set and transfers of various bit lengths are enabled. Also, the output level of the data transfer and LSB- or MSB-first can be set for 8-bit transfer data.

(3) Bit sequential buffer mode

This mode is used to transmit bit sequential data from the MCGO pin.

The transfer bit length can be set and transfers of various bit lengths are enabled. Also, the output level of the data transfer and LSB- or MSB-first can be set for 8-bit transfer data.

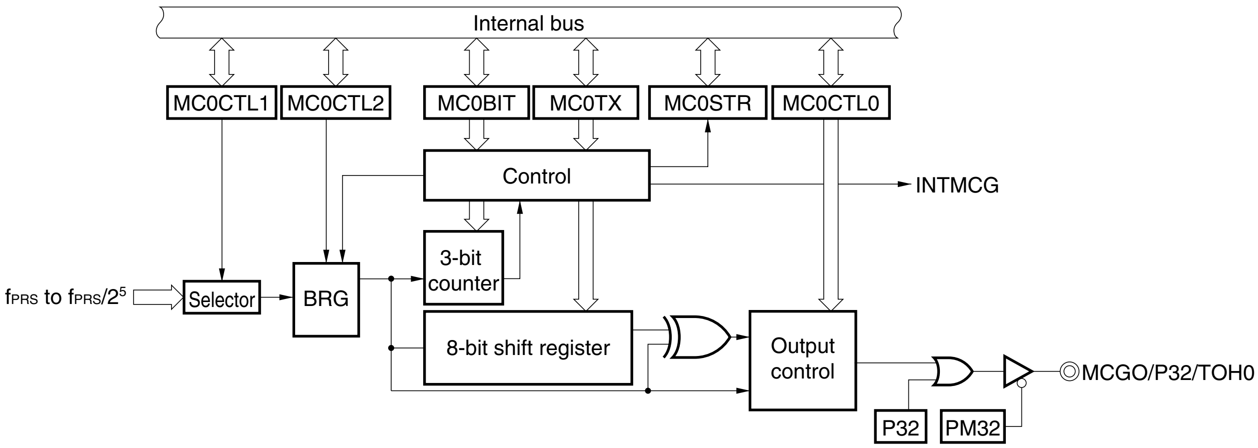
19.2 Configuration of Manchester Code Generator

The Manchester code generator includes the following hardware.

Table 19-1. Configuration of Manchester Code Generator

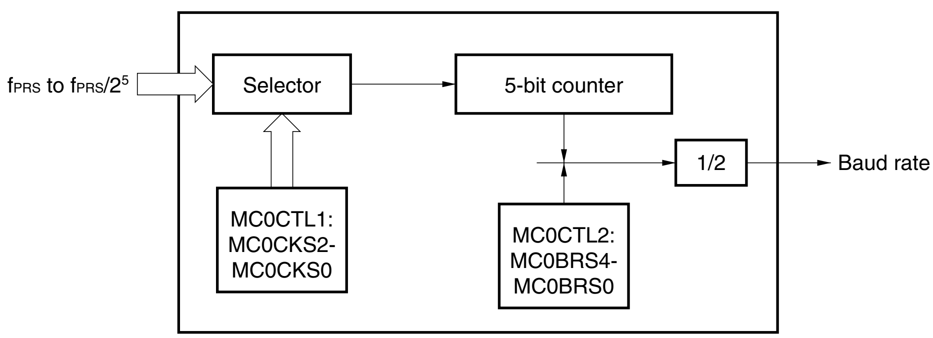
Item	Configuration
Registers	MCG transmit buffer register (MC0TX) MCG transmit bit count specification register (MC0BIT)
Control registers	MCG control register 0 (MC0CTL0) MCG control register 1 (MC0CTL1) MCG control register 2 (MC0CTL2) MCG status register (MC0STR) Port mode register 3 (PM3) Port register 3 (P3)

Figure 19-1. Block Diagram of Manchester Code Generator



Remark BRG: Baud rate generator
f_{PRS}: Peripheral hardware clock frequency
MC0BIT: MCG transmit bit count specification register
MC0CTL2 to MC0CTL0: MCG control registers 2 to 0
MC0STR: MCG status register
MC0TX: MCG transmit buffer register

Figure 19-2. Block Diagram of Baud Rate Generator



Remark f_{PRS}: Peripheral hardware clock frequency
MC0CTL2, MC0CTL 1: MCG control registers 2, 1
MC0CKS2 to MC0CKS0: Bits 2 to 0 of MC0CTL1 register
MC0BRS4 to MC0BRS0: Bits 4 to 0 of MC0CTL2 register

(1) MCG transmit buffer register (MC0TX)

This register is used to set the transmit data. A transmit operation starts when data is written to MC0TX while bit 7 (MC0PWR) of MCG control register 0 (MC0CTL0) is 1.

The data written to MC0TX is converted into serial data by the 8-bit shift register, and output to the MCGO pin.

Manchester code or bit sequential data can be set as the output code using bit 1 (MC0OSL) of MCG control register 0 (MC0CTL0).

This register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

(2) MCG transmit bit count specification register (MC0BIT)

This register is used to set the number of transmit bits.

Set the transmit bit count to this register before setting the transmit data to MC0TX.

In continuous transmission, the number of transmit bits to be transmitted next needs to be written after the occurrence of a transmission start interrupt (INTMCG). However, if the next transmit count is the same number as the previous transmit count, this register does not need to be written.

This register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 07H.

Figure 19-3. Format of MCG Transmit Bit Count Specification Register (MC0BIT)

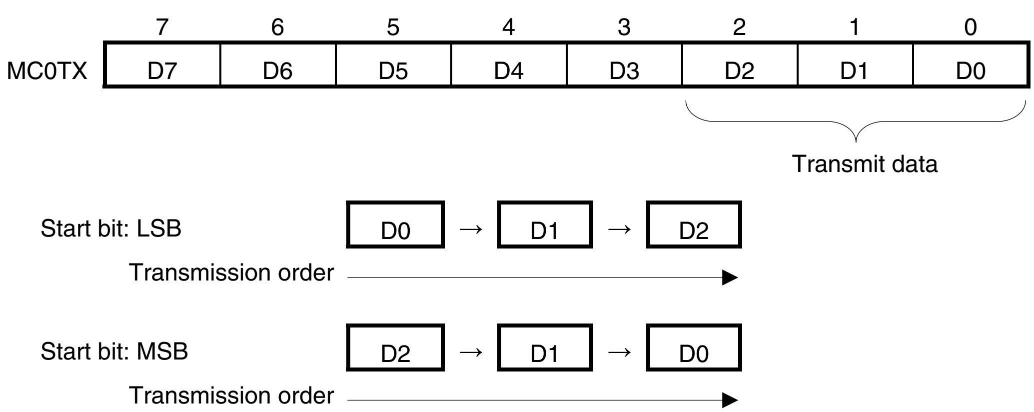
Address: FF4BH After reset: 07H R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
MC0BIT	0	0	0	0	0	MC0BIT2	MC0BIT1	MC0BIT0

MC0BIT2	MC0BIT1	MC0BIT0	Transmit bit count setting
0	0	0	1 bit
0	0	1	2 bits
0	1	0	3 bits
0	1	1	4 bits
1	0	0	5 bits
1	0	1	6 bits
1	1	0	7 bits
1	1	1	8 bits

Remark When the number of transmit bits is set as 7 bits or smaller, the lower bits are always transmitted regardless of MSB/LSB settings as the transmission start bit.

ex. When the number of transmit bits is set as 3 bits, and D7 to D0 are written to MCG transmit buffer register (MC0TX)



19.3 Registers Controlling Manchester Code Generator

The following six types of registers are used to control the Manchester code generator.

- MCG control register 0 (MC0CTL0)
- MCG control register 1 (MC0CTL1)
- MCG control register 2 (MC0CTL2)
- MCG status register (MC0STR)
- Port mode register 3 (PM3)
- Port register 3 (P3)

(1) MCG control register 0 (MC0CTL0)

This register is used to set the operation mode and to enable/disable the operation.
This register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to 10H.

Figure 19-4. Format of MCG Control Register 0 (MC0CTL0)

Address: FF4CH After reset: 10H R/W

Symbol	<7>	6	5	<4>	3	2	<1>	<0>
MC0CTL0	MC0PWR	0	0	MC0DIR	0	0	MC0OSL	MC0OLV

MC0PWR	Operation control
0	Operation stopped
1	Operation enabled

MC0DIR	First bit specification
0	MSB
1	LSB

MC0OSL	Data format
0	Manchester code
1	Bit sequential data

MC0OLV	Output level when transmission suspended
0	Low level
1	High level

Caution Clear (0) the MC0PWR bit before rewriting the MC0DIR, MC0OSL, and MC0OLV bits (it is possible to rewrite these bits by an 8-bit memory manipulation instruction at the same time when the MC0PWR bit is set (1)).

(2) MCG control register 1 (MC0CTL1)

This register is used to set the base clock of the Manchester code generator.

This register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 19-5. Format of MCG Control Register 1 (MC0CTL1)

Address: FF4DH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MC0CTL1	0	0	0	0	0	MC0CKS2	MC0CKS1	MC0CKS0

MC0CKS2	MC0CKS1	MC0CKS0	Base clock (f_{XCLK}) selection[Note 1]
0	0	0	f_{PRS}[Note 2] (10 MHz)
0	0	1	$f_{PRS}/2$ (5 MHz)
0	1	0	$f_{PRS}/2^2$ (2.5 MHz)
0	1	1	$f_{PRS}/2^3$ (1.25 MHz)
1	0	0	$f_{PRS}/2^4$ (625 kHz)
1	0	1	$f_{PRS}/2^5$ (312.5 kHz)
1	1	0	Setting prohibited
1	1	1	

Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{XH}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 2.7 to 5.5 V: $f_{PRS} \leq 10$ MHz
- V_{DD} = 1.8 to 2.7 V: $f_{PRS} \leq 5$ MHz

2. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{RH}) (XSEL = 0), when 1.8 V $\leq V_{DD} <$ 2.7 V, the setting of MC0CKS2 = MC0CKS1 = MC0CKS0 = 0 (base clock: f_{PRS}) is prohibited.

Caution Clear bit 7 (MC0PWR) of the MC0CTL0 register to 0 before rewriting the MC0CKS2 to MC0CKS0 bits.

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. Figures in parentheses are for operation with f_{PRS} = 10 MHz.

(3) MCG control register 2 (MC0CTL2)

This register is used to set the transmit baud rate.

This register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 1FH.

Figure 19-6. Format of MCG Control Register 2 (MC0CTL2)

Address: FF4EH After reset: 1FH R/W

Symbol	7	6	5	4	3	2	1	0
MC0CTL2	0	0	0	MC0BRS4	MC0BRS3	MC0BRS2	MC0BRS1	MC0BRS0

MC0BRS4	MC0BRS3	MC0BRS2	MC0BRS1	MC0BRS0	k	Output clock selection of 5-bit counter
0	0	0	×	×	4	$f_{XCLK}/4$
0	0	1	0	0	4	$f_{XCLK}/4$
0	0	1	0	1	5	$f_{XCLK}/5$
0	0	1	1	0	6	$f_{XCLK}/6$
0	0	1	1	1	7	$f_{XCLK}/7$
• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •
1	1	1	0	0	28	$f_{XCLK}/28$
1	1	1	0	1	29	$f_{XCLK}/29$
1	1	1	1	0	30	$f_{XCLK}/30$
1	1	1	1	1	31	$f_{XCLK}/31$

Cautions 1. Clear bit 7 (MC0PWR) of the MC0CTL0 register to 0 before rewriting the MC0BRS4 to MC0BRS0 bits.

2. The value from further dividing the output clock of the 5-bit counter by 2 is the baud rate value.

Remarks 1. f_{XCLK}: Frequency of the base clock selected by the MC0CKS2 to MC0CKS0 bits of the MC0CTL1 register

2. k: Value set by the MC0BRS4 to MC0BRS0 bits (k = 4, 5, 6, 7, …., 31)

3. ×: Don’t care

(4) MCG status register (MC0STR)

This register is used to indicate the operation status of the Manchester code generator.

This register can be read by a 1-bit or 8-bit memory manipulation instruction. Writing to this register is not possible.

Reset signal generation or setting MC0PWR = 0 clears this register to 00H.

Figure 19-7. Format of MCG Status Register (MC0STR)

Address: FF47H After reset: 00H R

Symbol	<7>	6	5	4	3	2	1	0
MC0STR	MC0TSF	0	0	0	0	0	0	0

MC0TSF	Data transmission status
0	• Reset signal generation • MC0PWR = 0 • If the next transfer data is not written to MC0TX when a transmission is completed
1	Transmission operation in progress

Caution This flag always indicates 1 during continuous transmission. Do not initialize a transmission operation without confirming that this flag has been cleared.

19.4 Operation of Manchester Code Generator

The Manchester code generator has the three modes described below.

- Operation stop mode
- Manchester code generator mode
- Bit sequential buffer mode

19.4.1 Operation stop mode

Transmissions are not performed in the operation stop mode. Therefore, the power consumption can be reduced. In addition, the P32/TOH0/MCGO pin is used as an ordinary I/O port in this mode.

(1) Register description

MCG control register 0 (MC0CTL0) is used to set the operation stop mode.

To set the operation stop mode, clear bit 7 (MC0PWR) of MC0CTL0 to 0.

(a) MCG control register 0 (MC0CTL0)

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

Reset signal generation sets this register to 10H.

Address: FF4CH After reset: 10H R/W

Symbol	<7>	6	5	<4>	3	2	<1>	<0>
MC0CTL0	MC0PWR	0	0	MC0DIR	0	0	MC0OSL	MC0OLV

MC0PWR	Operation control
0	Operation stopped

19.4.2 Manchester code generator mode

This mode is used to transmit data in Manchester code format using the MCGO pin.

(1) Register description

MCG control register 0 (MC0CTL0), MCG control register 1 (MC0CTL1), and MCG control register 2 (MC0CTL2) are used to set the Manchester code generator mode.

(a) MCG control register 0 (MC0CTL0)

This register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to 10H.

Address: FF4CH After reset: 10H R/W

Symbol	<7>	6	5	<4>	3	2	<1>	<0>
MC0CTL0	MC0PWR	0	0	MC0DIR	0	0	MC0OSL	MC0OLV

MC0PWR	Operation control
0	Operation stopped
1	Operation enabled

MC0DIR	First bit specification
0	MSB
1	LSB

MC0OSL	Data format
0	Manchester code
1	Bit sequential data

MC0OLV	Output level when transmission suspended
0	Low level
1	High level

Caution Clear (0) the MC0PWR bit before rewriting the MC0DIR, MC0OSL, and MC0OLV bits (it is possible to rewrite these bits by an 8-bit memory manipulation instruction at the same time when the MC0PWR bit is set (1)).

(b) MCG control register 1 (MC0CTL1)

This register is used to set the base clock of the Manchester code generator.

This register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Address: FF4DH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MC0CTL1	0	0	0	0	0	MC0CKS2	MC0CKS1	MC0CKS0

MC0CKS2	MC0CKS1	MC0CKS0	Base clock (f_{XCLK}) selection[Note 1]
0	0	0	f_{PRS}[Note 2] (10 MHz)
0	0	1	$f_{PRS}/2$ (5 MHz)
0	1	0	$f_{PRS}/2^2$ (2.5 MHz)
0	1	1	$f_{PRS}/2^3$ (1.25 MHz)
1	0	0	$f_{PRS}/2^4$ (625 kHz)
1	0	1	$f_{PRS}/2^5$ (312.5 kHz)
1	1	0	
1	1	1	

Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{XH}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 2.7 to 5.5 V: $f_{PRS} \leq 10$ MHz
- V_{DD} = 1.8 to 2.7 V: $f_{PRS} \leq 5$ MHz

2. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{RH}) (XSEL = 0), when 1.8 V $\leq V_{DD} <$ 2.7 V, the setting of MC0CKS2 = MC0CKS1 = MC0CKS0 = 0 (base clock: f_{PRS}) is prohibited.

Caution Clear bit 7 (MC0PWR) of the MC0CTL0 register to 0 before rewriting the MC0CKS2 to MC0CKS0 bits.

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. Figures in parentheses are for operation with f_{PRS} = 10 MHz.

(c) MCG control register 2 (MC0CTL2)

This register is used to set the transmit baud rate.

This register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 1FH.

Address: FF4EH After reset: 1FH R/W

Symbol	7	6	5	4	3	2	1	0
MC0CTL2	0	0	0	MC0BRS4	MC0BRS3	MC0BRS2	MC0BRS1	MC0BRS0

MC0BRS4	MC0BRS3	MC0BRS2	MC0BRS1	MC0BRS0	k	Output clock selection of 5-bit counter
0	0	0	×	×	4	$f_{XCLK}/4$
0	0	1	0	0	4	$f_{XCLK}/4$
0	0	1	0	1	5	$f_{XCLK}/5$
0	0	1	1	0	6	$f_{XCLK}/6$
0	0	1	1	1	7	$f_{XCLK}/7$
• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •
1	1	1	0	0	28	$f_{XCLK}/28$
1	1	1	0	1	29	$f_{XCLK}/29$
1	1	1	1	0	30	$f_{XCLK}/30$
1	1	1	1	1	31	$f_{XCLK}/31$

Cautions 1. Clear bit 7 (MC0PWR) of the MC0CTL0 register to 0 before rewriting the MC0BRS4 to MC0BRS0 bits.

2. The value from further dividing the output clock of the 5-bit counter by 2 is the baud rate value.

Remarks 1. f_{XCLK}: Frequency of the base clock selected by the MC0CKS2 to MC0CKS0 bits of the MC0CTL1 register

2. k: Value set by the MC0BRS4 to MC0BRS0 bits (k = 4, 5, 6, 7, …., 31)

3. ×: Don't care

<1> Baud rate

The baud rate can be calculated by the following expression.

- Baud rate = $\frac{f_{XCLK}}{2 \times k}$ [bps]

f_{XCLK}: Frequency of base clock selected by the MC0CKS2 to MC0CKS0 bits of the MC0CTL1 register

k: Value set by the MC0BRS4 to MC0BRS0 bits of the MC0CTL2 register (k = 4, 5, 6, ..., 31)

<2> Error of baud rate

The baud rate error can be calculated by the following expression.

- $\text{Error (\%)} = \left(\dfrac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (correct baud rate)}} - 1 \right) \times 100\ [\%]$

Caution Keep the baud rate error during transmission to within the permissible error range at the reception destination.

Example: Frequency of base clock = 2.5 MHz = 2,500,000 Hz
Set value of MC0BRS4 to MC0BRS0 bits of MC0CTL2 register = 10000B (k = 16)
Target baud rate = 76,800 bps

Baud rate = 2.5 M/(2 × 16)
= 2,500,000/(2 × 16) = 78125 [bps]

Error = (78,125/76,800 − 1) × 100
= 1.725 [%]

<3> Example of setting baud rate

Baud Rate [bps]	f_{PRS} = 10.0 MHz				f_{PRS} = 8.38 MHz				f_{PRS} = 8.0 MHz				f_{PRS} = 6.0 MHz			
	MC0CKS2 to MC0CKS0	k	Calculated Value	ERR [%]	MC0CKS2 to MC0CKS0	k	Calculated Value	ERR [%]	MC0CKS2 to MC0CKS0	k	Calculated Value	ERR [%]	MC0CKS2 to MC0CKS0	k	Calculated Value	ERR [%]
4800	–	–	–	–	5, 6, or 7	27	4850	1.03	5, 6, or 7	26	4808	0.16	5, 6, or 7	20	4688	–2.34
9600	5, 6, or 7	16	9766	1.73	4	27	9699	1.03	5, 6, or 7	13	9615	0.16	4	20	9375	–2.34
19200	5	8	19531	1.73	3	27	19398	1.03	4	13	19231	0.16	4	10	18750	–2.34
31250	4	10	31250	0	2	17	30809	–1.41	4	8	31250	0	2	24	31250	0
38400	4	8	39063	1.73	2	27	38796	1.03	3	13	38462	0.16	2	20	37500	–2.34
56000	3	11	56818	1.46	2	19	55132	–1.55	3	9	55556	–0.79	1	27	55556	–0.79
62500	2	20	62500	0	2	17	61618	–1.41	3	8	62500	0	2	12	62500	0
76800	2	16	78125	1.73	1	27	77592	1.03	2	13	76923	0.16	2	10	75000	–2.34
115200	1	22	113636	–1.36	2	9	116389	1.03	1	17	117647	2.12	1	13	115385	0.16
125000	1	20	125000	0	1	17	123235	–1.41	1	16	125000	0	1	12	125000	0
153600	1	16	156250	1.73	2	7	149643	–2.58	1	13	153846	0.16	1	10	150000	–2.34
250000	1	10	250000	0	1	8	261875	4.75	1	8	250000	0	1	6	250000	0
					0	17	246471	–1.41								

Remark MC0CKS2 to MC0CKS0: Bits 2 to 0 of MCG control register 1 (MC0CTL1) (setting of base clock (f_{XCLK}))
k: Value set by bits 4 to 0 (MC0BRS4 to MC0BRS0) of MCG control register 2 (MC0CTL2) (k = 4, 5, 6, …, 31)
f_{PRS}: Peripheral hardware clock frequency
ERR: Baud rate error

(d) Port mode register 3 (PM3)

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

When using the P32/TOH0/MCGO pin for Manchester code output, clear PM32 to 0 and clear the output latch of P32 to 0.

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

Reset signal generation sets these registers to FFH.

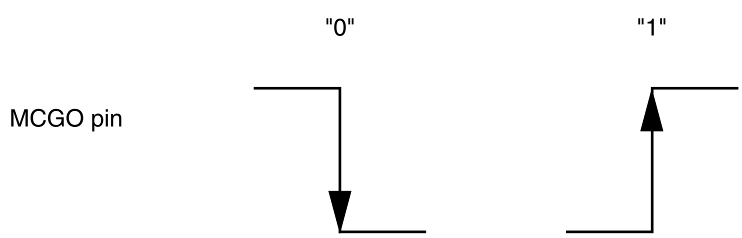
Address: FF23H After reset: FFH R/W

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

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

(2) Format of "0" and "1" of Manchester code output

The format of "0" and "1" of Manchester code output in 78K0/LF3 is as follows.



(3) Transmit operation

In Manchester code generator mode, data is transmitted in 1- to 8-bit units. Data bits are transmitted in Manchester code format. Transmission is enabled if bit 7 (MC0PWR) of MCG control register 0 (MC0CTL0) is set to 1.

The output value while a transmission is suspended can be set by using bit 0 (MC0OLV) of the MC0CTL0 register.

A transmission starts by writing a value to the MCG transmit buffer register (MC0TX) after setting the transmit data bit length to the MCG transmit bit count specification register (MC0BIT). At the transmission start timing, the MC0BIT value is transferred to the 3-bit counter and the data of MC0TX is transferred to the 8-bit shift register. An interrupt request signal (INTMCG) occurs at the timing that the MC0TX value is transferred to the 8-bit shift register. The 8-bit shift register is continuously shifted by the baud rate clock, and signal that is XORed with the baud rate clock is output from the MCGO pin.

When continuous transmission is executed, the next data is set to MC0BIT and MC0TX during data transmission after INTMCG occurs.

To transmit continuously, writing the next transfer data to MC0TX must be complete within the period (3) and (4) in Figure 19-8. Rewrite the MC0BIT before writing to MC0TX during continuous transmission.

Figure 19-8. Timing of Manchester Code Generator Mode (LSB First) (1/4)

(1) Transmit timing (MC0OLV = 1, total transmit bit length = 8 bits)

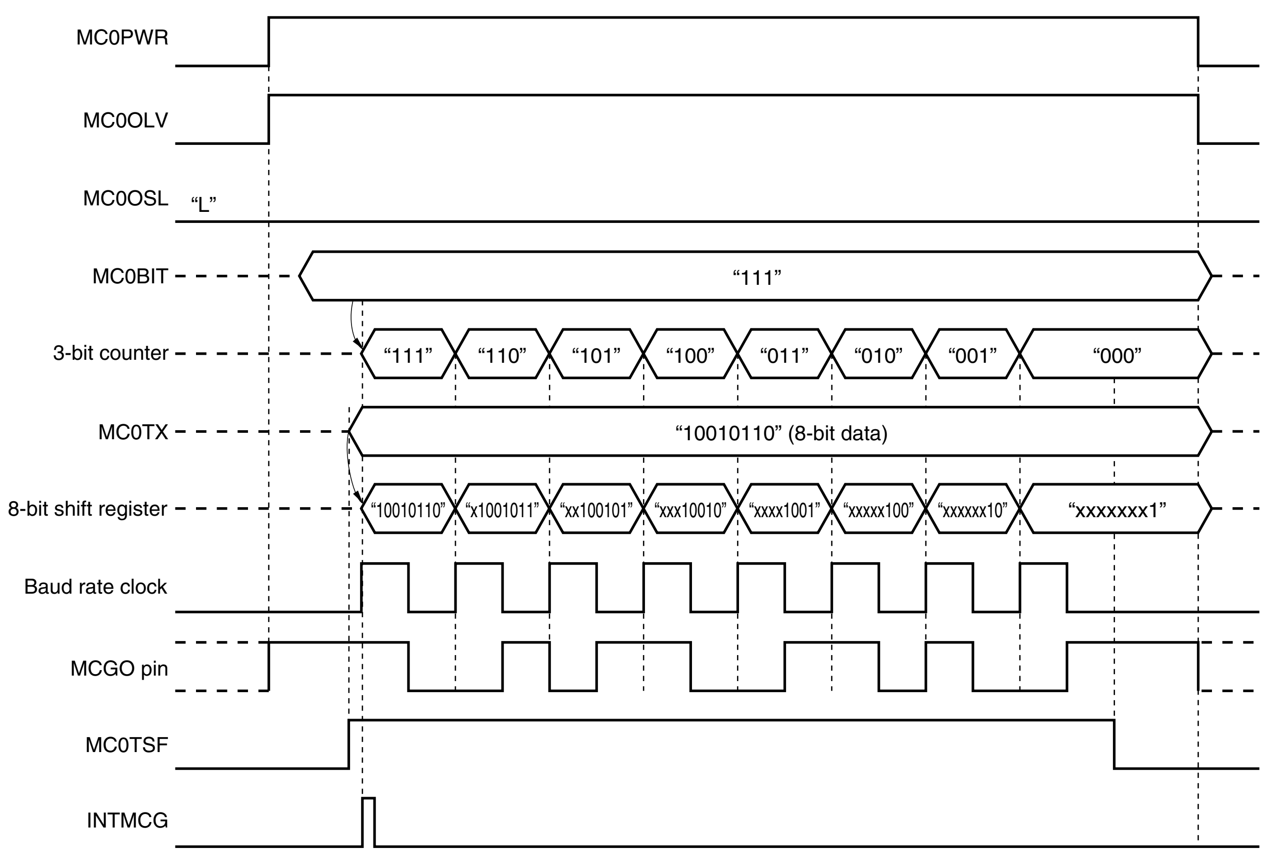


Figure 19-8. Timing of Manchester Code Generator Mode (LSB First) (2/4)

(2) Transmit timing (MC0OLV = 0, total transmit bit length = 8 bits)

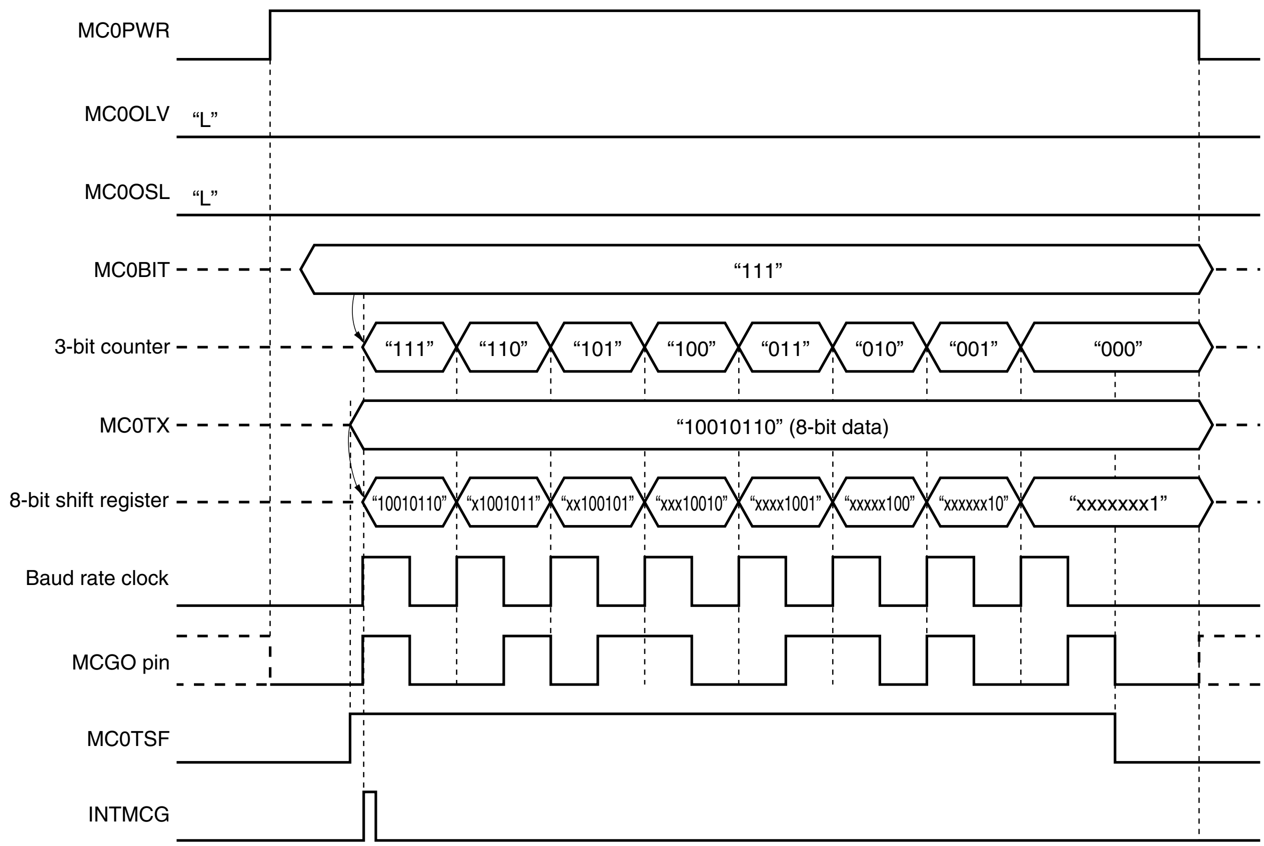
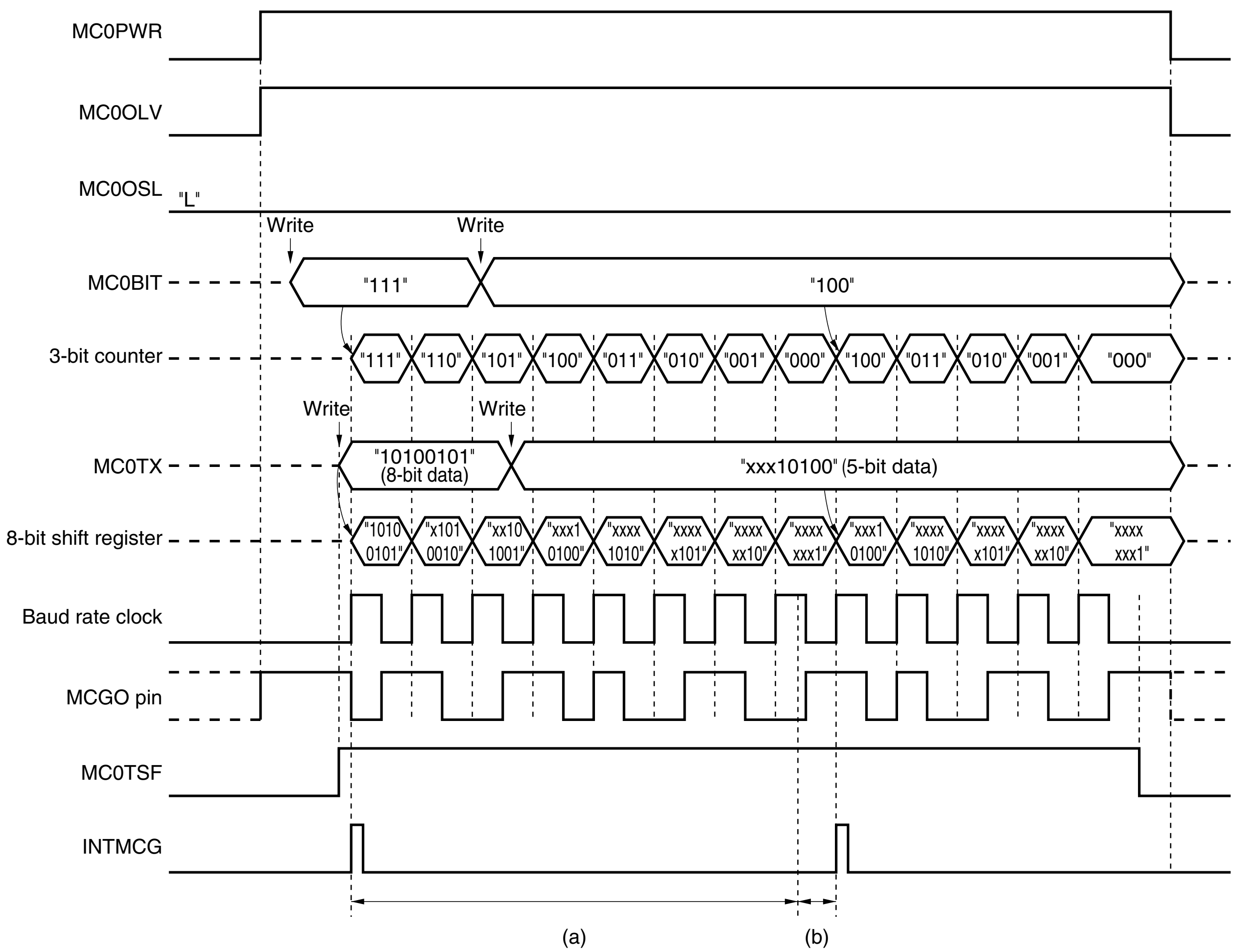


Figure 19-8. Timing of Manchester Code Generator Mode (LSB First) (3/4)

(3) Transmit timing (MC0OLV = 1, total transmit bit length = 13 bits)



(a): "8-bit transfer period" – (b)

(b): "1/2 cycle of baud rate" + 1 clock (f_{XCLK}) before the last bit of transmit data

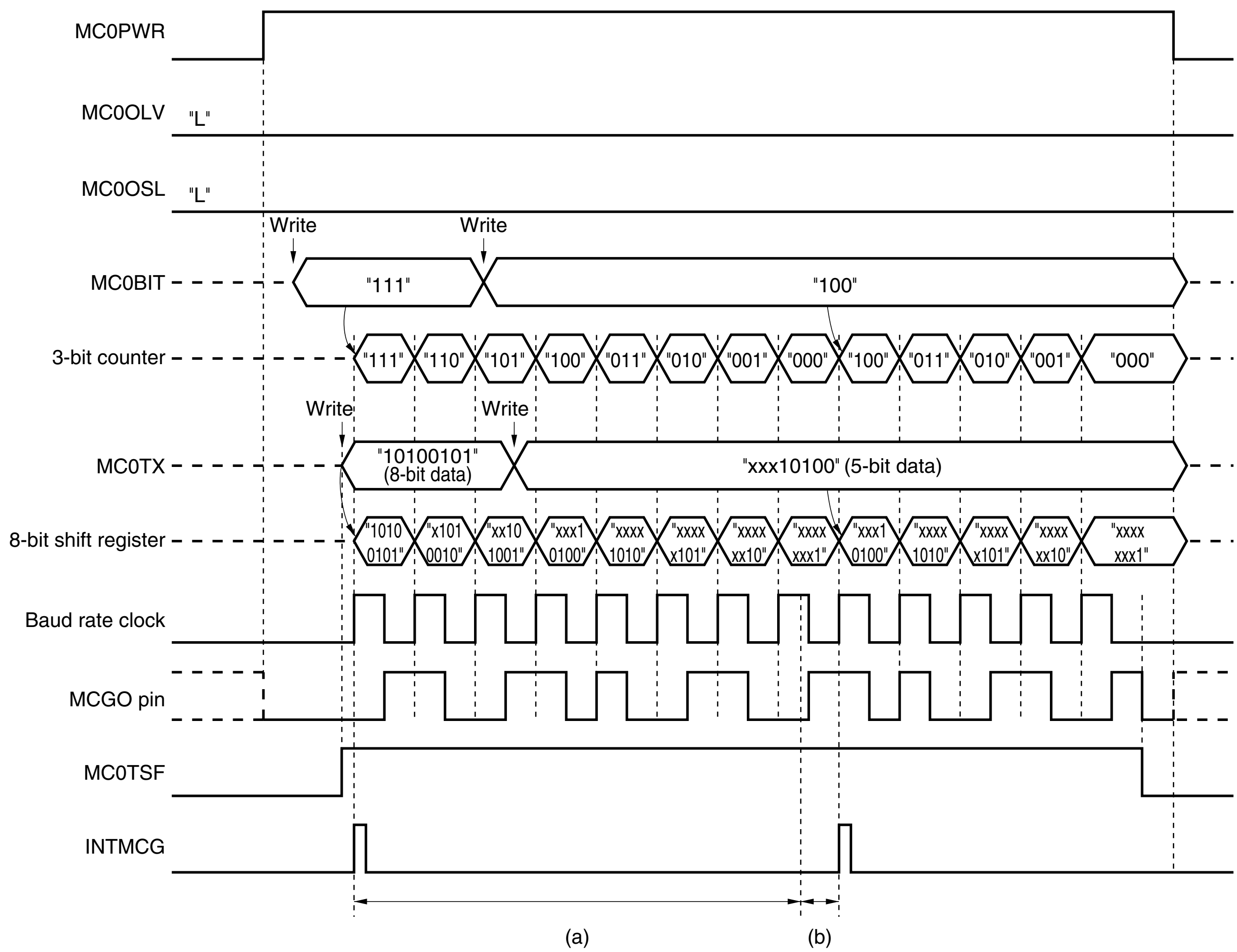
f_{XCLK}: Frequency of the operation base clock selected by using the MC0CKS2 to MC0CKS0 bits of the MC0CTL1 register

Last bit: Transfer bit when 3-bit counter = 000

Caution Writing the next transmit data to MC0TX must be complete within the period (a) during continuous transmission. If writing the next transmit data to MC0TX is executed in the period (b), the next data transmission starts 2 clocks (f_{XCLK}) after the last bit has been transmitted. Rewrite the MC0BIT before writing to MC0TX during continuous transmission.

Figure 19-8. Timing of Manchester Code Generator Mode (LSB First) (4/4)

(4) Transmit timing (MC0OLV = 0, total transmit bit length = 13 bits)



(a): "8-bit transfer period" – (b)

(b): "1/2 cycle of baud rate" + 1 clock (f_{XCLK}) before the last bit of transmit data

f_{XCLK}: Frequency of the operation base clock selected by using the MC0CKS2 to MC0CKS0 bits of the MC0CTL1 register

Last bit: Transfer bit when 3-bit counter = 000

Caution Writing the next transmit data to MC0TX must be complete within the period (a) during continuous transmission. If writing the next transmit data to MC0TX is executed in the period (b), the next data transmission starts 2 clocks (f_{XCLK}) after the last bit has been transmitted. Rewrite the MC0BIT before writing to MC0TX during continuous transmission.

19.4.3 Bit sequential buffer mode

The bit sequential buffer mode is used to output sequential signals using the MCGO pin.

(1) Register description

The MCG control register 0 (MC0CTL0), MCG control register 1 (MC0CTL1), and MCG control register 2 (MC0CTL2) are used to set the bit sequential buffer mode.

(a) MCG control register 0 (MC0CTL0)

This register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to 10H.

Address: FF4CH After reset: 10H R/W

Symbol	<7>	6	5	<4>	3	2	<1>	<0>
MC0CTL0	MC0PWR	0	0	MC0DIR	0	0	MC0OSL	MC0OLV

MC0PWR	Operation control
0	Operation stopped
1	Operation enabled

MC0DIR	First bit specification
0	MSB
1	LSB

MC0OSL	Data format
0	Manchester code
1	Bit sequential data

MC0OLV	Output level when transmission suspended
0	Low level
1	High level

Caution Clear (0) the MC0PWR bit before rewriting the MC0DIR, MC0OSL, and MC0OLV bits (it is possible to rewrite these bits by an 8-bit memory manipulation instruction at the same time when the MC0PWR bit is set (1)).

(b) MCG control register 1 (MC0CTL1)

This register is used to set the base clock of the Manchester code generator.

This register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Address: FF4DH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MC0CTL1	0	0	0	0	0	MC0CKS2	MC0CKS1	MC0CKS0

MC0CKS2	MC0CKS1	MC0CKS0	Base clock (f_{XCLK}) selection
0	0	0	f_{PRS} (10 MHz)
0	0	1	$f_{PRS}/2$ (5 MHz)
0	1	0	$f_{PRS}/2^2$ (2.5 MHz)
0	1	1	$f_{PRS}/2^3$ (1.25 MHz)
1	0	0	$f_{PRS}/2^4$ (625 kHz)
1	0	1	$f_{PRS}/2^5$ (312.5 kHz)
1	1	0	
1	1	1	

Caution Clear bit 7 (MC0PWR) of the MC0CTL0 register to 0 before rewriting the MC0CKS2 to MC0CKS0 bits.

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. Figures in parentheses are for operation with f_{PRS} = 10 MHz.

(c) MCG control register 2 (MC0CTL2)

This register is used to set the transmit baud rate.

This register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 1FH.

Address: FF4EH After reset: 1FH R/W

Symbol	7	6	5	4	3	2	1	0
MC0CTL2	0	0	0	MC0BRS4	MC0BRS3	MC0BRS2	MC0BRS1	MC0BRS0

MC0BRS4	MC0BRS3	MC0BRS2	MC0BRS1	MC0BRS0	k	Output clock selection of 5-bit counter
0	0	0	×	×	4	$f_{XCLK}/4$
0	0	1	0	0	4	$f_{XCLK}/4$
0	0	1	0	1	5	$f_{XCLK}/5$
0	0	1	1	0	6	$f_{XCLK}/6$
0	0	1	1	1	7	$f_{XCLK}/7$
• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •
1	1	1	0	0	28	$f_{XCLK}/28$
1	1	1	0	1	29	$f_{XCLK}/29$
1	1	1	1	0	30	$f_{XCLK}/30$
1	1	1	1	1	31	$f_{XCLK}/31$

Cautions 1. Clear bit 7 (MC0PWR) of the MC0CTL0 register to 0 before rewriting the MC0BRS4 to MC0BRS0 bits.

2. The value from further dividing the output clock of the 5-bit counter by 2 is the baud rate value.

Remarks 1. f_{XCLK}: Frequency of the base clock selected by the MC0CKS2 to MC0CKS0 bits of the MC0CTL1 register

2. k: Value set by the MC0BRS4 to MC0BRS0 bits (k = 4, 5, 6, 7, …., 31)

3. ×: Don't care

<1> Baud rate

The baud rate can be calculated by the following expression.

- Baud rate = $\frac{f_{XCLK}}{2 \times k}$ [bps]

f_{XCLK}: Frequency of base clock selected by the MC0CKS2 to MC0CKS0 bits of the MC0CTL1 register

k: Value set by the MC0BRS4 to MC0BRS0 bits of the MC0CTL2 register (k = 4, 5, 6, ..., 31)

<2> Error of baud rate

The baud rate error can be calculated by the following expression.

- $\text{Error (\%)} = \left(\dfrac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (correct baud rate)}} - 1 \right) \times 100\ [\%]$

Caution Keep the baud rate error during transmission to within the permissible error range at the reception destination.

Example: Frequency of base clock = 2.5 MHz = 2,500,000 Hz
Set value of MC0BRS4 to MC0BRS0 bits of MC0CTL2 register = 10000B (k = 16)
Target baud rate = 76,800 bps

Baud rate = 2.5 M/(2 × 16)
= 2,500,000/(2 × 16) = 78125 [bps]

Error = (78,125/76,800 − 1) × 100
= 1.725 [%]

<3> Example of setting baud rate

Baud Rate [bps]	f_{PRS} = 10.0 MHz				f_{PRS} = 8.38 MHz				f_{PRS} = 8.0 MHz				f_{PRS} = 6.0 MHz			
	MC0CKS2 to MC0CKS0	k	Calculated Value	ERR [%]	MC0CKS2 to MC0CKS0	k	Calculated Value	ERR [%]	MC0CKS2 to MC0CKS0	k	Calculated Value	ERR [%]	MC0CKS2 to MC0CKS0	k	Calculated Value	ERR [%]
4800	–	–	–	–	5, 6, or 7	27	4850	1.03	5, 6, or 7	26	4808	0.16	5, 6, or 7	20	4688	–2.34
9600	5, 6, or 7	16	9766	1.73	4	27	9699	1.03	5, 6, or 7	13	9615	0.16	4	20	9375	–2.34
19200	5	8	19531	1.73	3	27	19398	1.03	4	13	19231	0.16	4	10	18750	–2.34
31250	4	10	31250	0	2	17	30809	–1.41	4	8	31250	0	2	24	31250	0
38400	4	8	39063	1.73	2	27	38796	1.03	3	13	38462	0.16	2	20	37500	–2.34
56000	3	11	56818	1.46	2	19	55132	–1.55	3	9	55556	–0.79	1	27	55556	–0.79
62500	2	20	62500	0	2	17	61618	–1.41	3	8	62500	0	2	12	62500	0
76800	2	16	78125	1.73	1	27	77592	1.03	2	13	76923	0.16	2	10	75000	–2.34
115200	1	22	113636	–1.36	2	9	116389	1.03	1	17	117647	2.12	1	13	115385	0.16
125000	1	20	125000	0	1	17	123235	–1.41	1	16	125000	0	1	12	125000	0
153600	1	16	156250	1.73	2	7	149643	–2.58	1	13	153846	0.16	1	10	150000	–2.34
250000	1	10	250000	0	1	8	261875	4.75	1	8	250000	0	1	6	250000	0
					0	17	246471	–1.41								

Remark MC0CKS2 to MC0CKS0: Bits 2 to 0 of MCG control register 1 (MC0CTL1) (setting of base clock (f_{XCLK}))
k: Value set by bits 4 to 0 (MC0BRS4 to MC0BRS0) of MCG control register 2 (MC0CTL2) (k = 4, 5, 6, …, 31)
f_{PRS}: Peripheral hardware clock frequency
ERR: Baud rate error

(d) Port mode register 3 (PM3)

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

When using the P32/TOH0/MCGO pin for bit sequential data output, clear PM32 to 0 and clear the output latch of P32 to 0.

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

Reset signal generation sets these registers to FFH.

Address: FF23H After reset: FFH R/W

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

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

(2) Transmit operation

In bit sequential buffer mode, data is transmitted in 1- to 8-bit units. Transmission is enabled if bit 7 (MC0PWR) of MCG control register 0 (MC0CTL0) is set to 1.

The output value while transmission is suspended can be set by using bit 0 (MC0OLV) of the MC0CTL0 register.

A transmission starts by writing a value to the MCG transmit buffer register (MC0TX) after setting the transmit data bit length to the MCG transmit bit count specification register (MC0BIT). At the transmission start timing, the MC0BIT value is transferred to the 3-bit counter and data of MC0TX is transferred to the 8-bit shift register. An interrupt request signal (INTMCG) occurs at the timing that the MC0TX value is transferred to the 8-bit shift register. The 8-bit shift register is continuously shifted by the baud rate clock and is output from the MCGO pin.

When continuous transmission is executed, the next data is set to MC0BIT and MC0TX during data transmission after INTMCG occurs.

To transmit continuously, writing the next transfer data to MC0TX must be complete within the period (3) and (4) in Figure 19-9. Rewrite MC0BIT before writing to MC0TX during continuous transmission.

Figure 19-9. Timing of Bit Sequential Buffer Mode (LSB First) (1/4)

(1) Transmit timing (MC0OLV = 1, total transmit bit length = 8 bits)

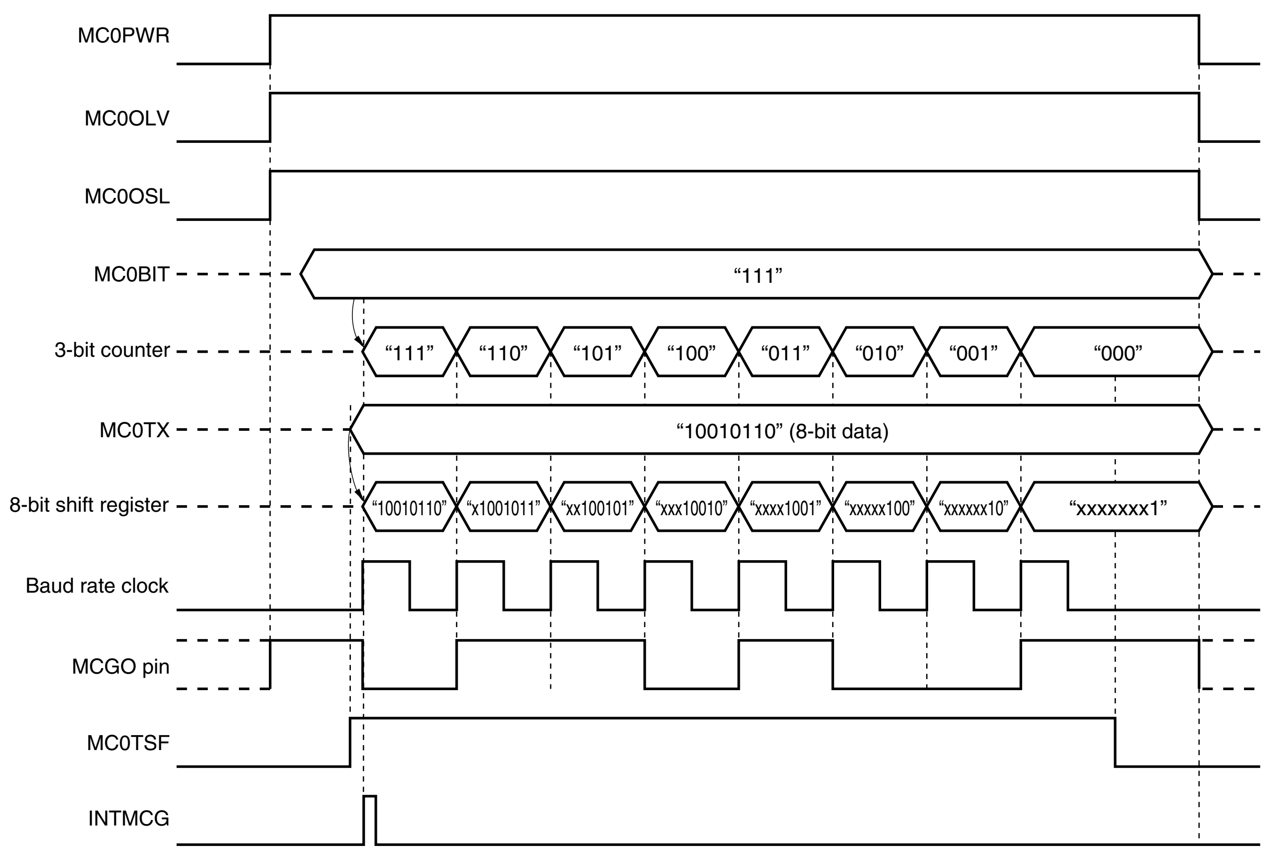


Figure 19-9. Timing of Bit Sequential Buffer Mode (LSB First) (2/4)

(2) Transmit timing (MC0OLV = 0, total transmit bit length = 8 bits)

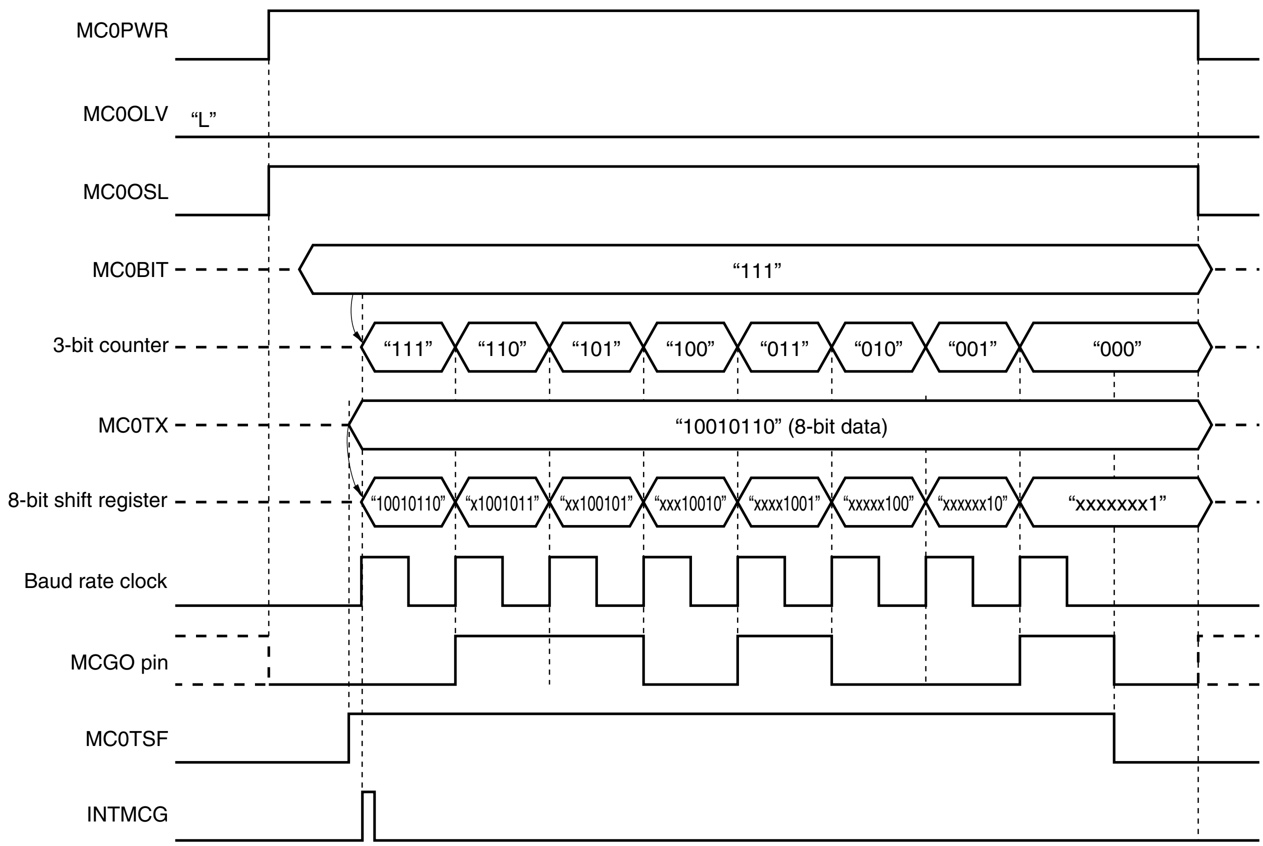
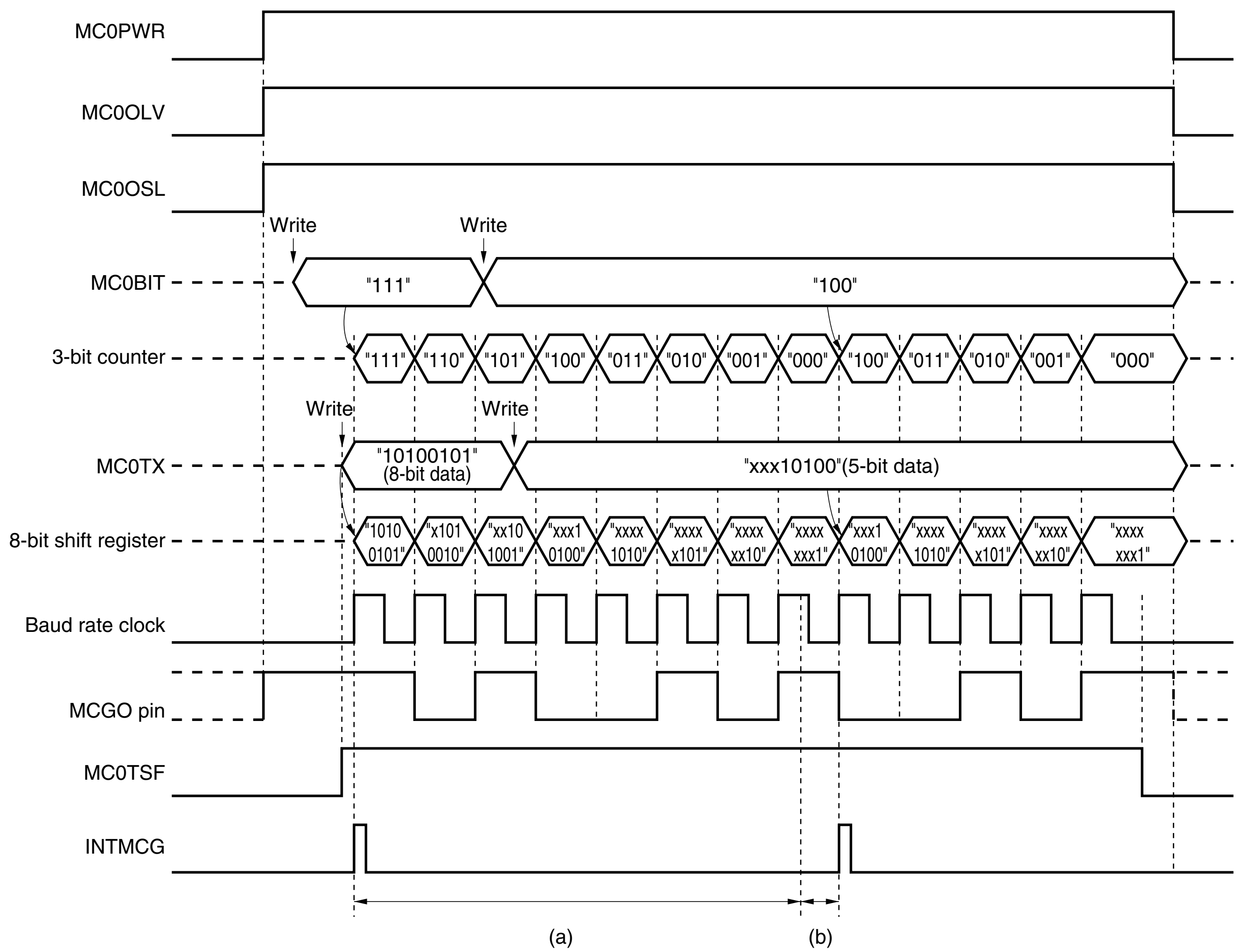


Figure 19-9. Timing of Bit Sequential Buffer Mode (LSB First) (3/4)

(3) Transmit timing (MC0OLV = 1, total transmit bit length = 13 bits)



(a): "8-bit transfer period" – (b)

(b): "1/2 cycle of baud rate" + 1 clock (f_{XCLK}) before the last bit of transmit data

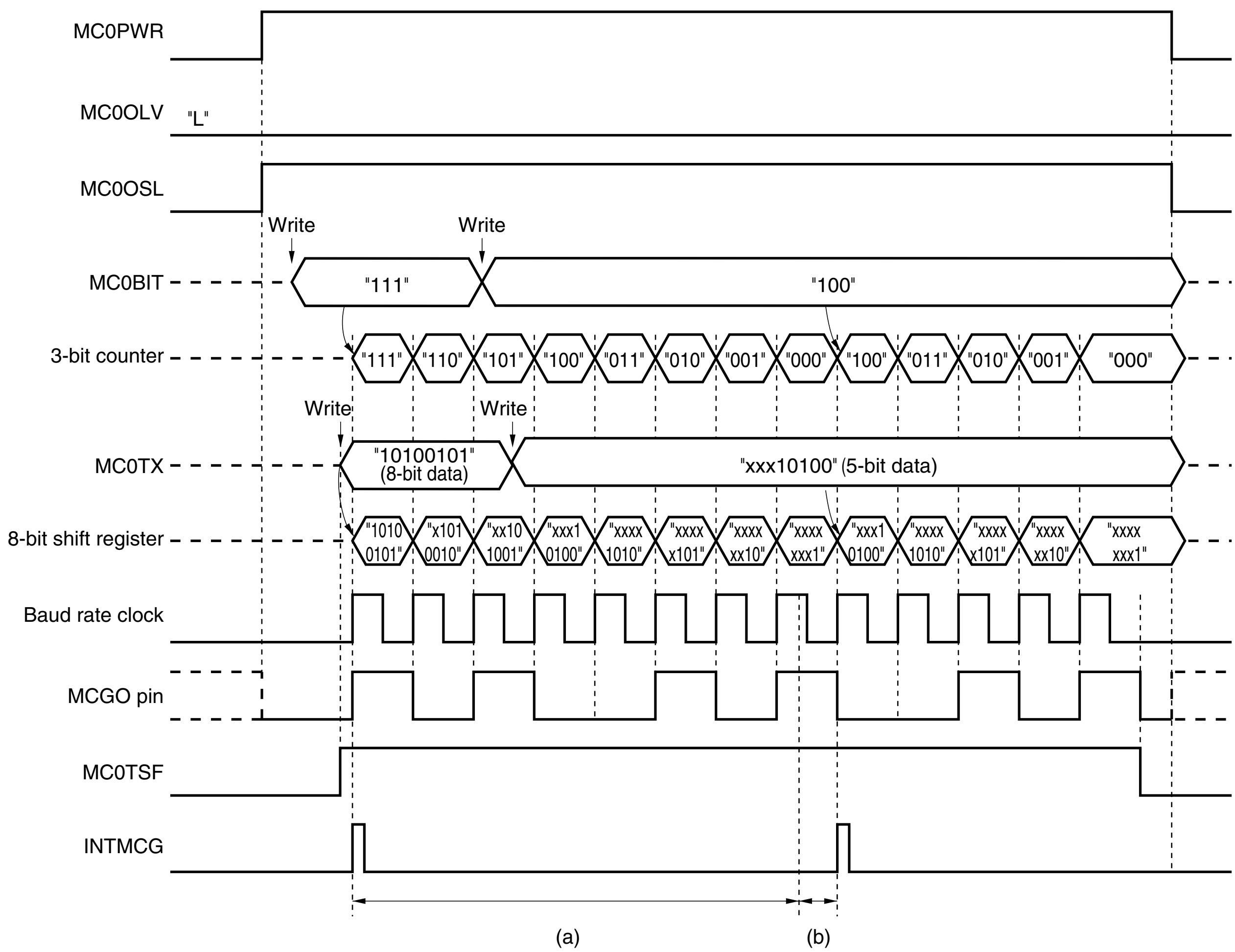
f_{XCLK}: Frequency of operation base clock selected by using the MC0CKS2 to MC0CKS0 bits of the MC0CTL1 register

Last bit: Transfer bit when 3-bit counter = 000

Caution Writing the next transmit data to MC0TX must be complete within the period (a) during continuous transmission. If writing the next transmit data to MC0TX is executed in the period (b), the next data transmission starts 2 clocks (f_{XCLK}) after the last bit has been transmitted.
Rewrite the MC0BIT before writing to MC0TX during continuous transmission.

Figure 19-9. Timing of Bit Sequential Buffer Mode (LSB First) (4/4)

(4) Transmit timing (MC0OLV = 0, total transmit bit length = 13 bits)



(a): "8-bit transfer period" – (b)

(b): "1/2 cycle of baud rate" + 1 clock (f_{XCLK}) before the last bit of transmit data

f_{XCLK}: Frequency of operation base clock selected by using the MC0CKS2 to MC0CKS0 bits of the MC0CTL1 register

Last bit: Transfer bit when 3-bit counter = 000

Caution Writing the next transmit data to MC0TX must be complete within the period (a) during continuous transmission. If writing the next transmit data to MC0TX is executed in the period (b), the next data transmission starts 2 clocks (f_{XCLK}) after the last bit has been transmitted. Rewrite the MC0BIT before writing to MC0TX during continuous transmission.

<R>

20.1 Remote Controller Receiver Functions

The remote controller receiver uses the following remote controller modes.

- Type A reception mode … Guide pulse (half clock) provided
- Type B reception mode … Guide pulse (1clock) provided
- Type C reception mode … Guide pulse not provided

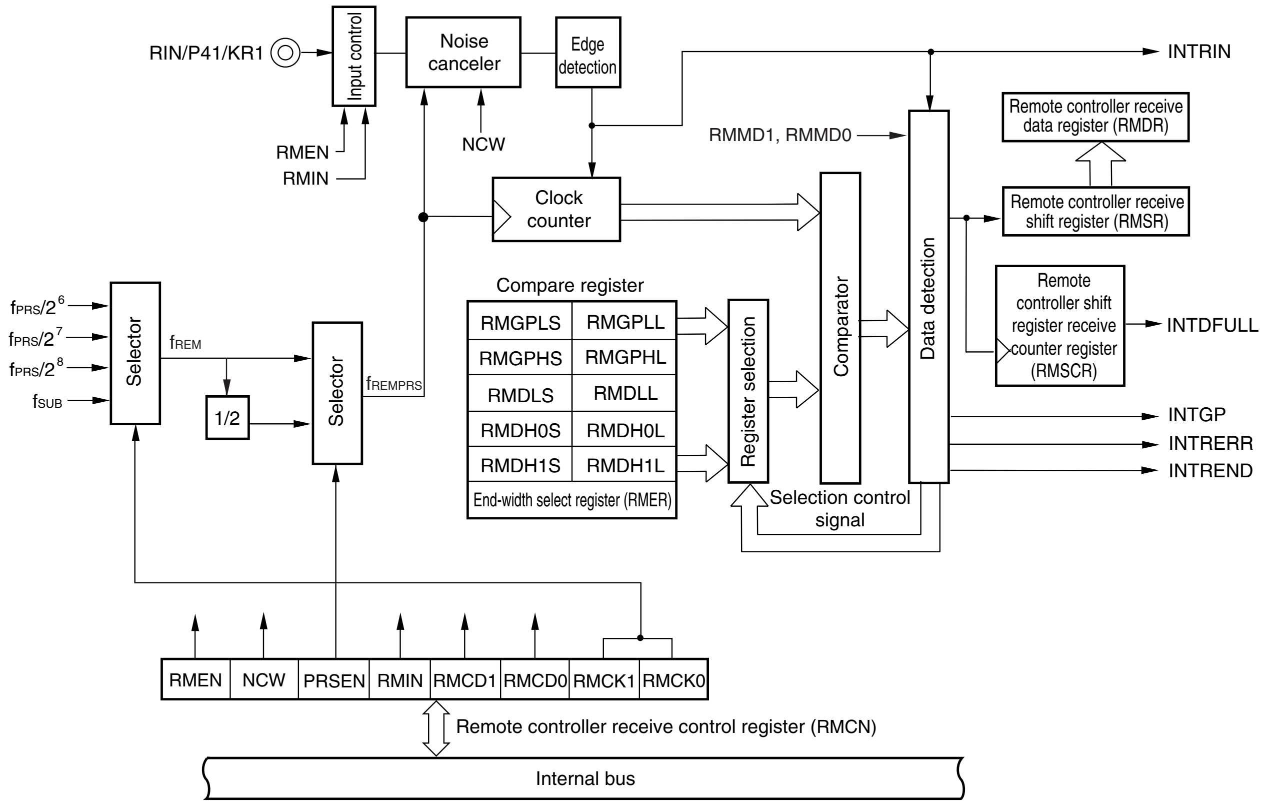
20.2 Remote Controller Receiver Configuration

The remote controller receiver includes the following hardware.

Table 20-1. Remote Controller Receiver Configuration

Item	Configuration
Registers	Remote controller receive shift register (RMSR) Remote controller receive data register (RMDR) Remote controller shift register receive counter register (RMSCR) Remote controller receive GPLS compare register (RMGPLS) Remote controller receive GPLL compare register (RMGPLL) Remote controller receive GPHS compare register (RMGPHS) Remote controller receive GPHL compare register (RMGPHL) Remote controller receive DLS compare register (RMDLS) Remote controller receive DLL compare register (RMDLL) Remote controller receive DH0S compare register (RMDH0S) Remote controller receive DH0L compare register (RMDH0L) Remote controller receive DH1S compare register (RMDH1S) Remote controller receive DH1L compare register (RMDH1L) Remote controller receive end width select register (RMER)
Control register	Remote controller receive interrupt status register (INTS) Remote controller receive interrupt status clear register (INTC) Remote controller receive control register (RMCN)

Figure 20-1. Block Diagram of Remote Controller Receiver



(1) Remote controller receive shift register (RMSR)

This is an 8-bit register for reception of remote controller data.

Data is stored in bit 7 first. Each time new data is stored, the stored data is shifted to the lower bits. Therefore, the latest data is stored in bit 7, and the first data is stored in bit 0.

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

Reset signal generation sets RMSR to 00H.

Also, RMSR is cleared to 00H under any of the following conditions.

- Remote controller stops operation (RMEN = 0).
- Error is detected (INTRERR is generated).
- INTDFULL is generated.
- RMSR is read after INTREND has been generated.

Caution Reading RMSR is disabled during remote controller reception. Complete reception, then read RMSR. When the reading operation is complete, RMSR is cleared. Therefore, values once read are not guaranteed.

(2) Remote controller receive data register (RMDR)

This register holds the remote controller reception data. When the remote controller receive shift register (RMSR) overflows, the data in RMSR is transferred to RMDR. Bit 7 stores the last data, and bit 0 stores the first data. INTDFULL is generated at the same time as data is transferred from RMSR to RMDR.

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

Reset signal generation sets RMDR to 00H.

When the remote controller operation is disabled (RMEN = 0), RMDR is cleared to 00H.

Caution When INTDFULL has been generated, read RMDR before the next 8-bit data is received. If the next INTDFULL is generated before the read operation is complete, RMDR is overwritten.

(3) Remote controller shift register receive counter register (RMSCR)

This is a 3-bit counter register used to indicate the number of valid bits remaining in the remote controller receive shift register (RMSR) when remote controller reception is complete (INTREND is generated). Reading the values of this register allows confirmation of the number of bits, even if the received data is in a format other than an integral multiple of 8 bits.

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

Reset signal generation sets RMSCR to 00H.

It is cleared to 00H under any of the following conditions.

- Remote controller stops operation (RMEN = 0).
- Error is detected (INTRERR is generated).
- RMSR is read after INTREND has been generated.

Caution When INTREND has been generated, immediately read RMSCR before reading RMSR. If reading occurs at another timing, the value is not guaranteed.

Figure 20-2. Operation Examples of RMSR, RMSCR, and RMDR Registers When Receiving 1010101011111111B (16 Bits)

	RMSR								RMSCR	RMDR
	7	6	5	4	3	2	1	0		
After reset	0	0	0	0	0	0	0	0	00H	00000000B
Receiving 1 bit	1	0	0	0	0	0	0	0	01H	00000000B
Receiving 2 bits	0	1	0	0	0	0	0	0	02H	00000000B
Receiving 3 bits	1	0	1	0	0	0	0	0	03H	00000000B
…	…	…	…	…	…	…	…	…	…	…
Receiving 7 bits	1	0	1	0	1	0	1	0	07H	00000000B
Receiving 8 bits ↓ RMDR transfer	0 ↓ 0	1 ↓ 0	0 ↓ 0	1 ↓ 0	0 ↓ 0	1 ↓ 0	0 ↓ 0	1 ↓ 0	00H ↓ 00H	00000000B ↓ 01010101B
Receiving 9 bits	1	0	0	0	0	0	0	0	01H	01010101B
Receiving 10 bits	1	1	0	0	0	0	0	0	02H	01010101B
…	…	…	…	…	…	…	…	…	…	…
Receiving 16 bits ↓ RMDR transfer	1 ↓ 0	1 ↓ 0	1 ↓ 0	1 ↓ 0	1 ↓ 0	1 ↓ 0	1 ↓ 0	1 ↓ 0	00H ↓ 00H	01010101B ↓ 11111111B

(4) Remote controller receive GPLS compare register (RMGPLS) (Type B reception mode)

This register is used to detect the low level of a remote controller guide pulse (short side).

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

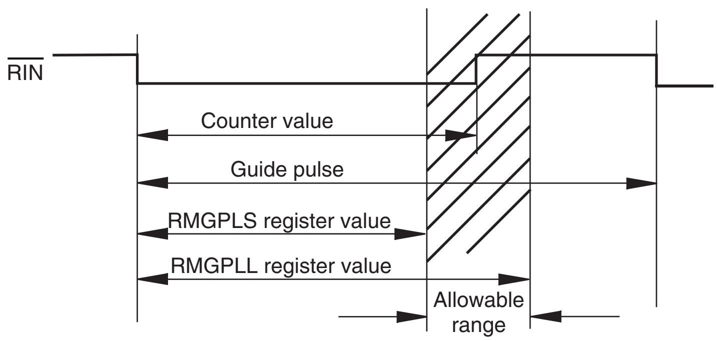
Reset signal generation sets RMGPLS to 00H.

(5) Remote controller receive GPLL compare register (RMGPLL) (Type B reception mode)

This register is used to detect the low level of a remote controller guide pulse (long side).

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

Reset signal generation sets RMGPLL to 00H.



If RMGPLS ≤ counter value < RMGPLL is satisfied, it is assumed that the low level of the guide pulse has been successfully received.

(6) Remote controller receive GPHS compare register (RMGPHS) (Type A, Type B reception mode only)

This register is used to detect the high level of a remote controller guide pulse (short side).

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

Reset signal generation sets RMGPHS to 00H.

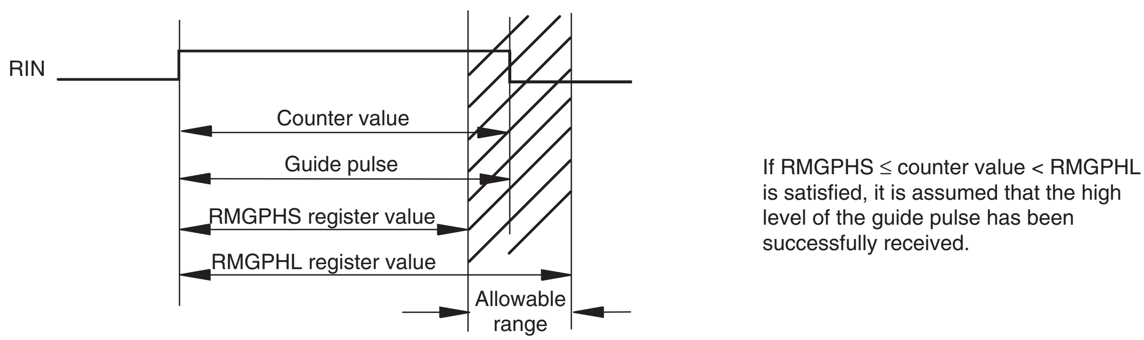
(7) Remote controller receive GPHL compare register (RMGPHL) (Type A, Type B reception mode only)

This register is used to detect the high level of a remote controller guide pulse (long side).

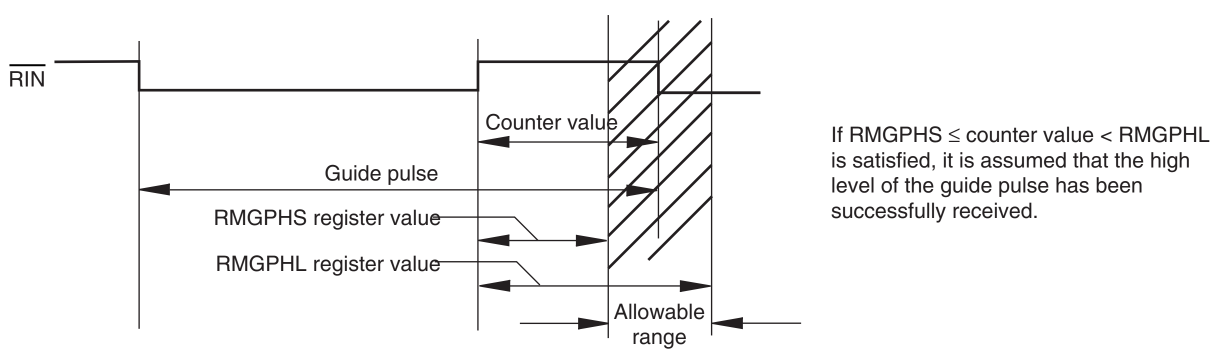
RMGPHL is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMGPHL to 00H.

(a) Type A reception mode



(b) Type B reception mode



(8) Remote controller receive DLS compare register (RMDLS)

This register is used to detect the low level of a remote controller data (short side).

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

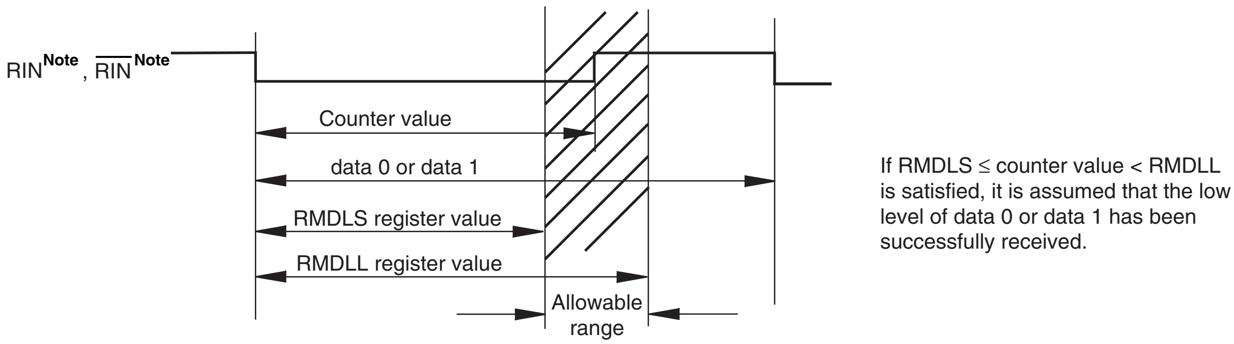
Reset signal generation sets RMDLS to 00H.

(9) Remote controller receive DLL compare register (RMDLL)

This register is used to detect the low level of a remote controller data (long side).

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

Reset signal generation sets RMDLL to 00H.



Note RIN is generated in type A reception mode, and $\overline{RIN}$ is generated in type B and type C reception modes.

(10) Remote controller receive DH0S compare register (RMDH0S)

This register is used to detect the high level of a remote controller data 0 (short side).

RMDH0S is set with an 8-bit memory manipulation instruction.

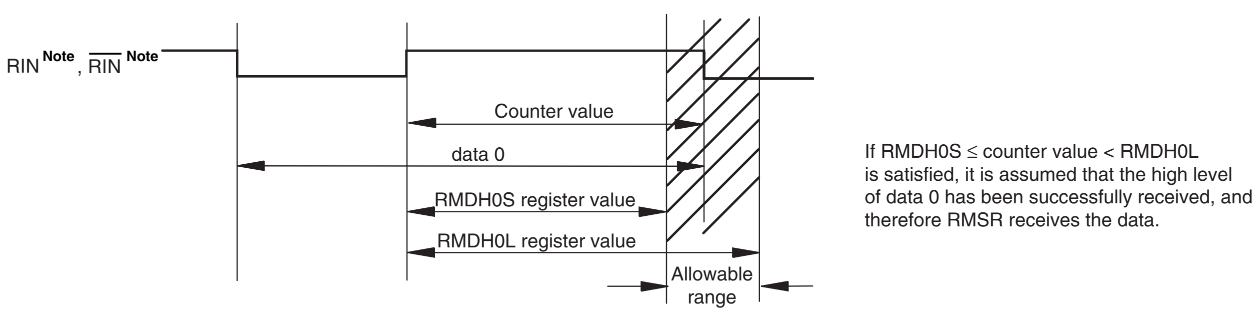
Reset signal generation sets RMDH0S to 00H.

(11) Remote controller receive DH0L compare register (RMDH0L)

This register is used to detect the high level of a remote controller data 0 (long side).

RMDH0L is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMDH0L to 00H.



Note RIN is generated in type A reception mode, and $\overline{RIN}$ is generated in type B and type C reception modes.

(12) Remote controller receive DH1S compare register (RMDH1S)

This register is used to detect the high level of remote controller data 1 (short side).

RMDH1S is set with an 8-bit memory manipulation instruction.

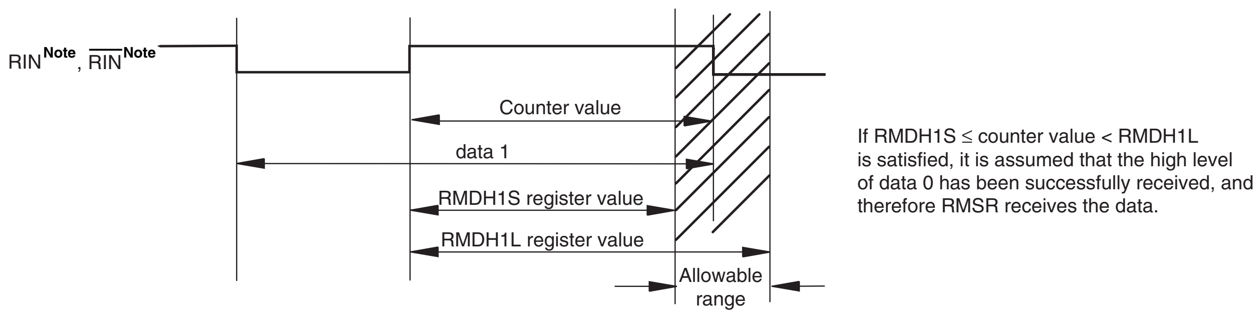
Reset signal generation sets RMDH1S to 00H.

(13) Remote controller receive DH1L compare register (RMDH1L)

This register is used to detect the high level of remote controller data 1 (long side).

RMDH1L is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMDH1L to 00H.



Note RIN is generated in type A reception mode, and $\overline{\text{RIN}}$ is generated in type B and type C reception modes.

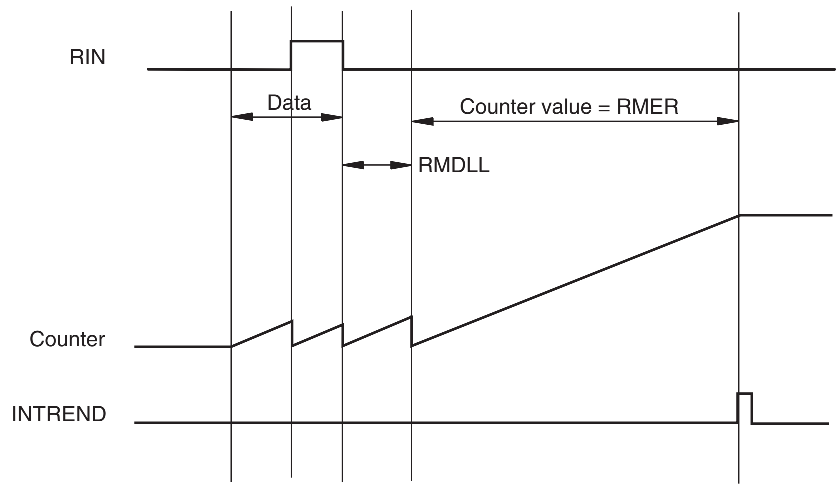
(14) Remote controller receive end-width select register (RMER)

This register determines the interval between the timing at which the INTREND signal is output.

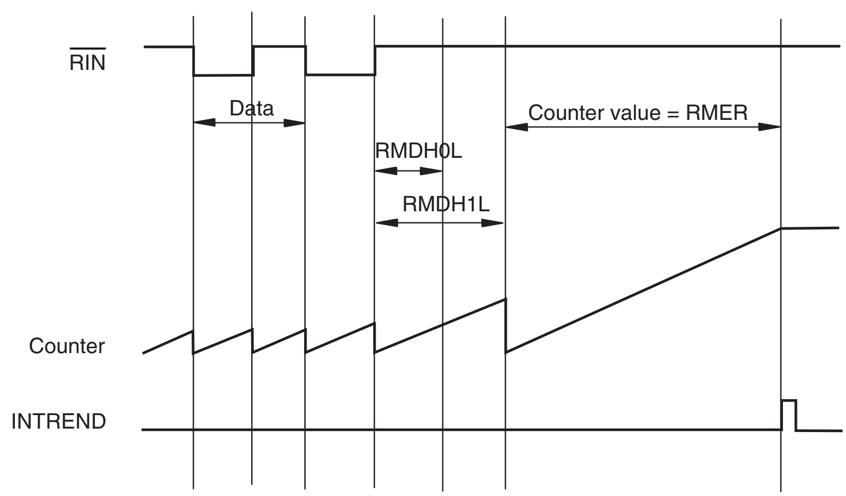
RMER is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMER to 00H.

(a) Type A reception mode



(b) Type B, Type C reception mode



Caution For RMER and all the remote controller receive compare registers (RMGPLS, RMGPLL, RMGPHS, RMGPHL, RMDLS, RMDLL, RMDH0S, RMDH0L, RMDH1S, and RMDH1L), disable remote controller reception (bit 7 (RMEN) of the remote controller receive control register (RMCN) = 0) first, and then change the value.

20.3 Registers to Control Remote Controller Receiver

The remote controller receiver is controlled by the following register.

- Remote controller receive interrupt status register (INTS)
- Remote controller receive interrupt status clear register (INTC)
- Remote controller receive control register (RMCN)

(1) Remote controller receive interrupt status register (INTS)

This register is used to identify which interrupt request among the remote control receive interrupts (INTRERR, INTGP, INTREND, INTDFULL) has occurred.

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

Reset signal generation sets INTS to 00H.

Figure 20-3. Format of Remote Controller Receive Interrupt Status Register (INTS)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
INTS	0	0	0	0	INTS DFULL	INTS REND	INTS GP	INTS RERR	FFF9H	00H	R

INTS DFULL	Interrupt request by reading of 8-bit shift data
0	Interrupt request by reading of 8-bit shift data has not occurred
1	Interrupt request by reading of 8-bit shift data has occurred

INTS REND	Request by data reception completion interrupt
0	Request by data reception completion interrupt has not occurred
1	Request by data reception completion interrupt has occurred

INTS GP	Guide pulse detection interrupt
0	Guide pulse detection interrupt request has not occurred
1	Guide pulse detection interrupt request has occurred

INTS RERR	Interrupt request by remote control receive error
0	Interrupt request by remote control receive error has not occurred
1	Interrupt request by remote control receive error has occurred

Caution The INTS register will not be cleared even if it is read. Use the INTC register to clear the INTS register.

(2) Remote controller receive interrupt status clear register (INTC)

This register is used to control the remote controller receive interrupt status register (INTS).

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

Reset signal generation sets INTC to 00H.

Figure 20-4. Format of Remote Controller Receive Interrupt Status Clear Register (INTC)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
INTC	0	0	0	0	INTC DFULL	INTC REND	INTC GP	INTC RERR	FFFAH	00H	R/W

INTC DFULL	Interrupt identification bit control by reading of 8-bit shift data
0	INTSDFULL bit not changed
1	INTSDFULL bit cleared

INTC REND	Data reception completion Interrupt identification bit control
0	INTSREND bit not changed
1	INTSREND bit cleared

INTC GP	Guide pulse detection interrupt identification bit control
0	INTSGP bit not changed
1	INTSGP bit cleared

INTC RERR	Interrupt identification bit control by remote control receive error
0	INTSRERR bit not changed
1	INTSRERR bit cleared

(3) Remote controller receive control register (RMCN)

This register is used to enable/disable remote controller reception and to set the noise elimination width, clock internal division, input invert signal, and source clock.

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

Reset signal generation sets RMCN to 00H.

Figure 20-5. Format of Remote Controller Receive Control Register (RMCN)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
RMCN	RMEN	NCW	PRSEN	RMIN	RMMD1	RMMD0	RMCK1	RMCK0	FF9AH	00H	R/W

RMEN	Control of remote controller receive operation
0	Disable remote controller reception
1	Enable remote controller reception

NCW	Noise elimination width control signal
0	Eliminate noise less than $1/f_{REMPRS}$
1	Eliminate noise less than $2/f_{REMPRS}$

PRSEN	Internal clock division control signal
0	Clock not divided internally ($f_{REMPRS} = f_{REM}$)
1	Clock internally divided into two ($f_{REMPRS} = f_{REM}/2$)

RMIN	Remote controller input invert signal
0	Input positive phase
1	Input negative phase

RMMD1	RMMD0	Remote controller reception mode
0	0	Type A reception mode (Guide pulse (half clock) provided)
0	1	Type B reception mode (Guide pulse (1 clock) provided)
1	0	Type C reception mode (Guide pulse not provided)
1	1	Setting prohibited

RMCK1	RMCK0	Selection of source clock (f_{REM}) of remote controller counter
0	0	$f_{PRS}/2^6$ (156.25 kHz)
0	1	$f_{PRS}/2^7$ (78.125 kHz)
1	0	$f_{PRS}/2^8$ (39.063 kHz)
1	1	f_{SUB} (32.768 kHz)

Caution To change the values of NCW, PRSEN, RMIN, RMMD1, RMMD0, RMCK1, and RMCK0, disable remote controller reception (RMEN = 0) first.

Remark 1. f_{REM}: Source clock of remote controller counter (selected by bits 0 and 1 (RMCK0 and RMCK1))

Remarks 2. f_{REMPRS}: Operation clock inside remote controller receiver

3. f_{PRS}: Peripheral hardware clock frequency

4. f_{SUB}: Oscillation frequency of subsystem clock

5. The parenthesized values apply to operation at f_{PRS} = 10 MHz and f_{SUB} = 32.768 kHz.

20.4 Operation of Remote Controller Receiver

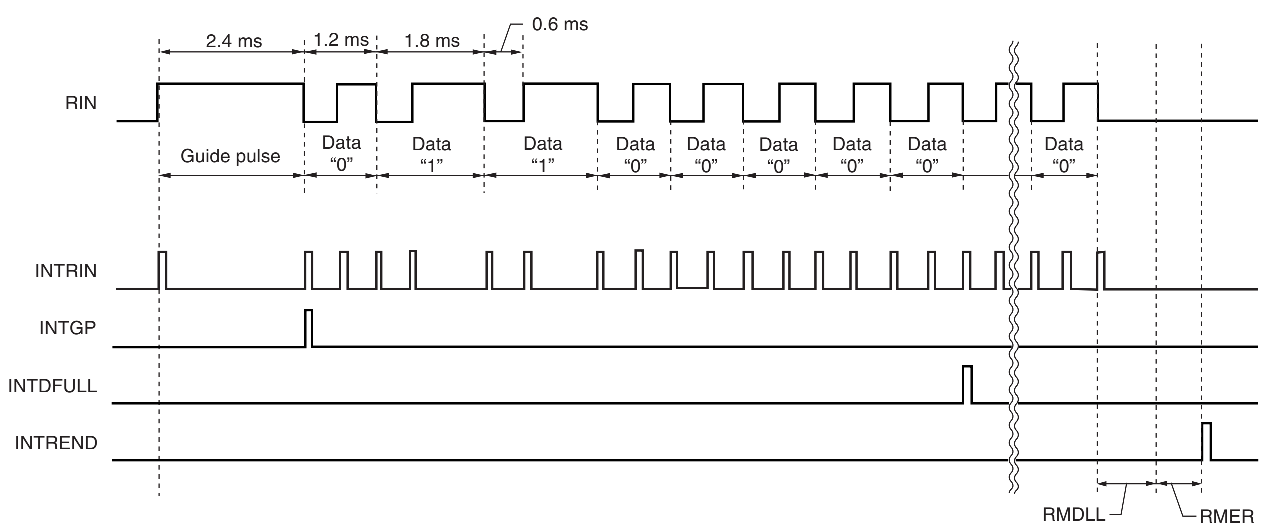
The following remote controller reception mode is used for this remote controller receiver.

- Type A reception mode with guide pulse (half clock)
- Type B reception mode with guide pulse (1clock)
- Type C reception mode without guide pulse

20.4.1 Format of type A reception mode

Figure 20-6 shows the data format for type A.

Figure 20-6. Example of Type A Data Format



20.4.2 Operation flow of type A reception mode

Figure 20-7 shows the operation flow.

Cautions 1. When INTRERR is generated, RMSR and RMSCR are automatically cleared immediately.

2. When data has been set to all the bits of RMSR, the following processing is automatically performed.

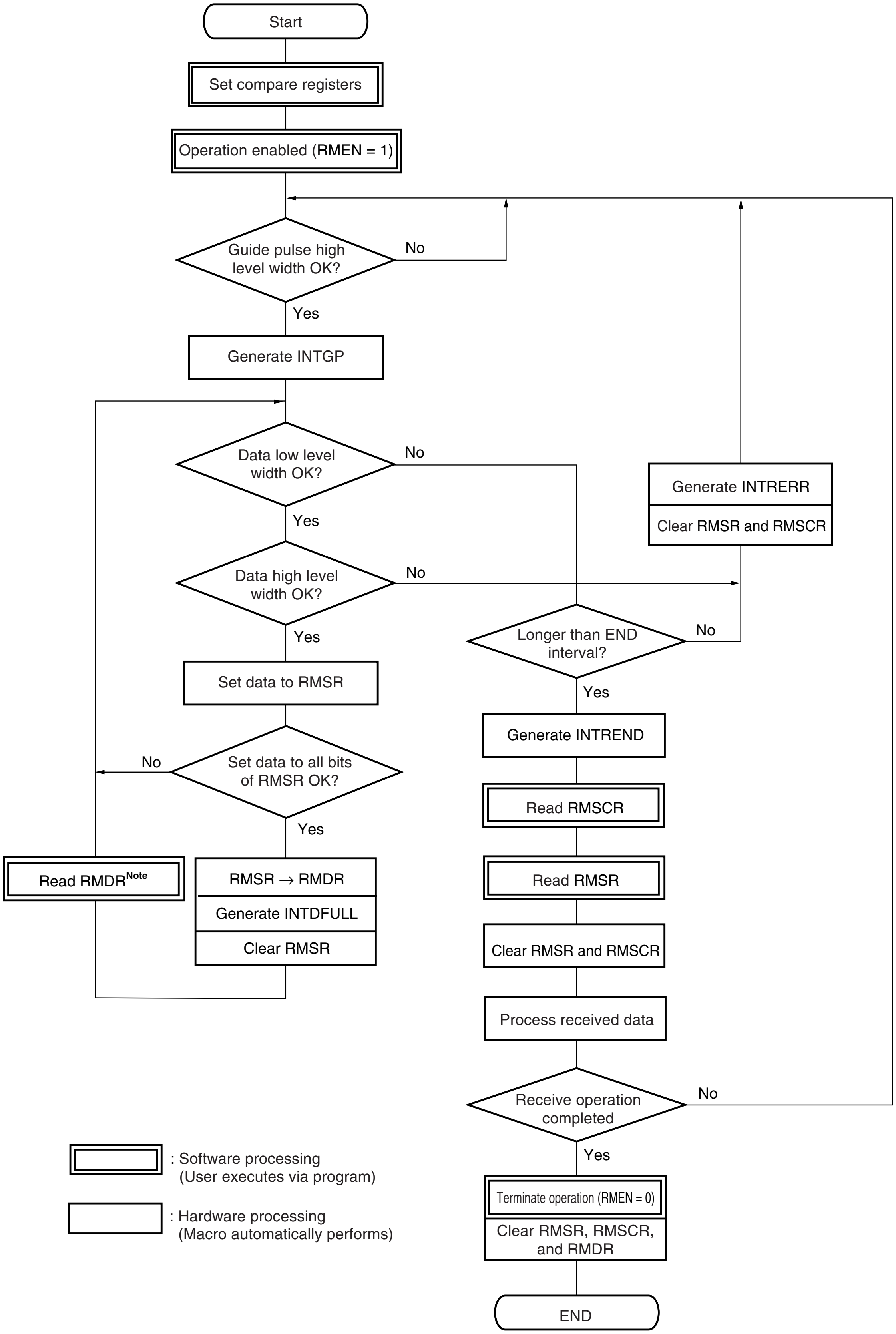
- **The value of RMSR is transferred to RMDR.**
- **INTDFULL is generated.**
- **RMSR is cleared.**

RMDR must then be read before the next data is set to all the bits of RMSR.

3. When INTREND has been generated, read RMSCR first followed by RMSR.
When RMSR has been read, RMSCR and RMSR are automatically cleared.
If INTREND is generated, the next data cannot be received until RMSR is read.

4. RMSR, RMSCR, and RMDR are cleared simultaneously to operation termination (RMEN = 0).

Figure 20-7. Operation Flow of Type A Reception Mode

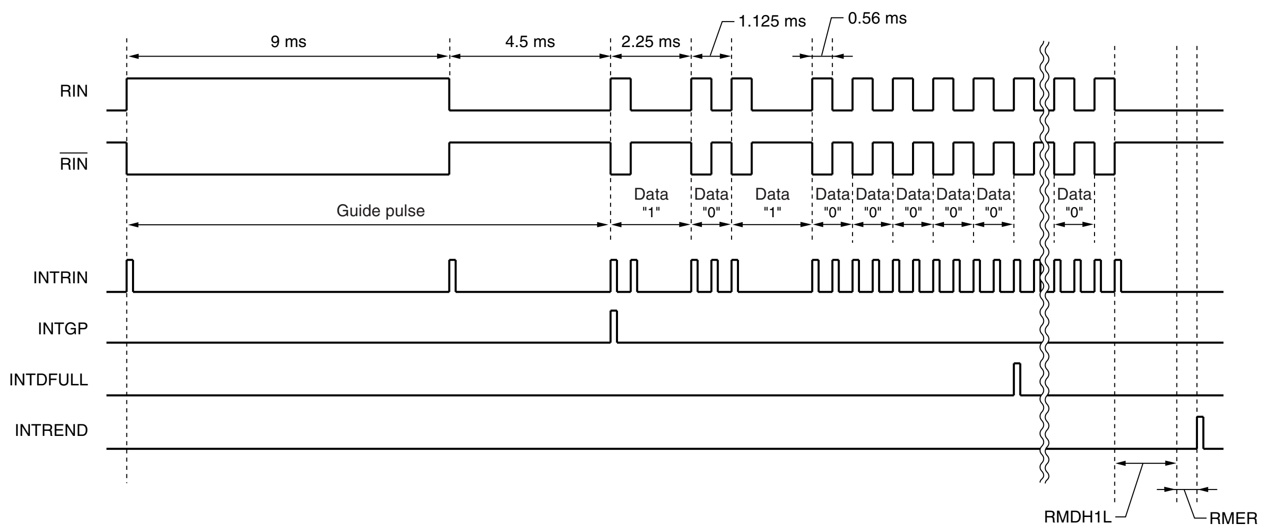


Note Read RMDR before data has been set to all the bits of RMSR.

20.4.3 Format of type B reception mode

Figure 20-8 shows the data format for type B.

Figure 20-8. Example of Type B Data Format



Remark RIN is the internally inverted signal of RIN.

20.4.4 Operation flow of type B reception mode

Figure 20-9 shows the operation flow.

Cautions 1. When INTRERR is generated, RMSR and RMSCR are automatically cleared immediately.

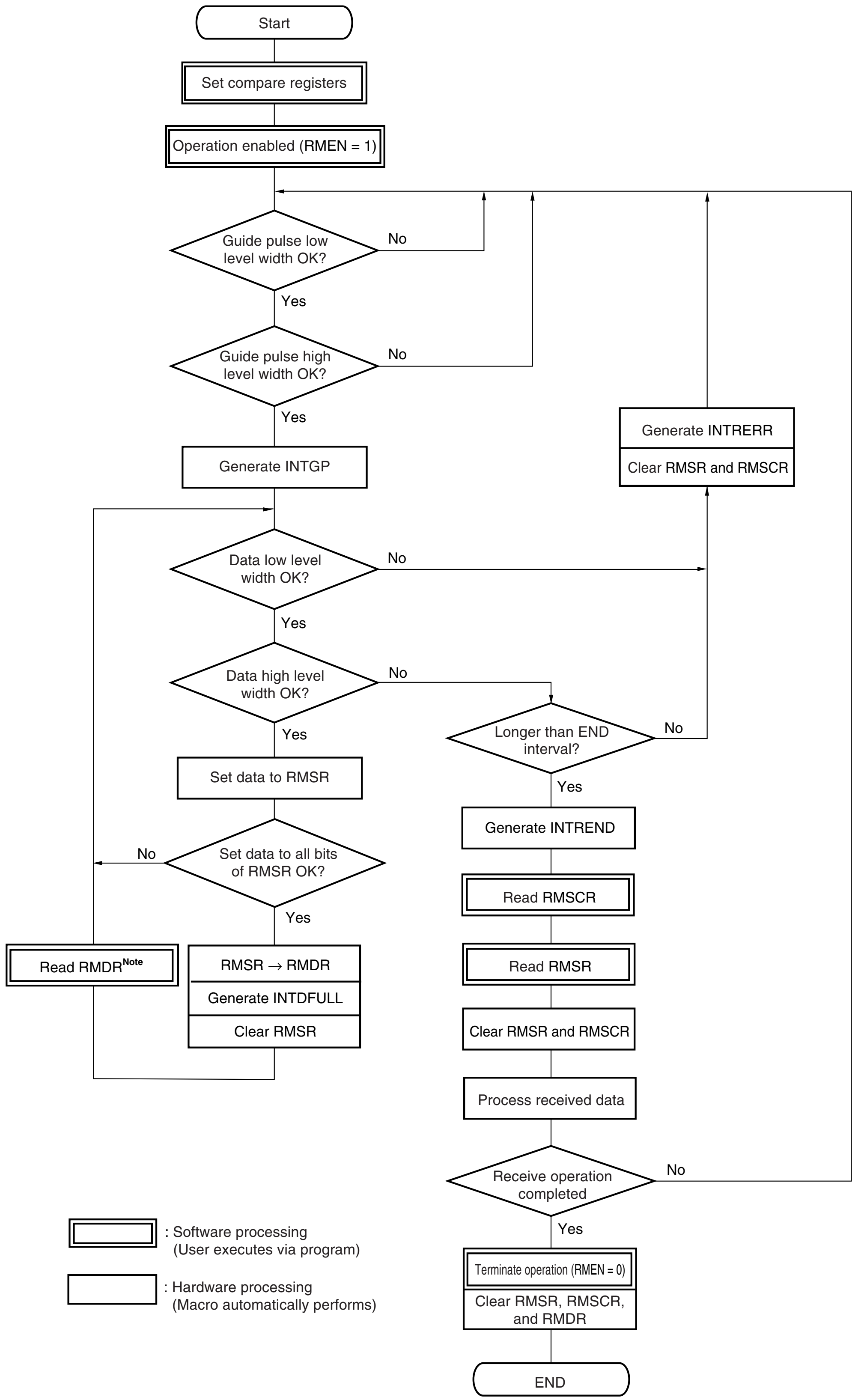
2. When data has been set to all the bits of RMSR, the following processing is automatically performed.
- **The value of RMSR is transferred to RMDR.**
- **INTDFULL is generated.**
- **RMSR is cleared.**

RMDR must then be read before the next data is set to all the bits of RMSR.

3. When INTREND has been generated, read RMSCR first followed by RMSR.
When RMSR has been read, RMSCR and RMSR are automatically cleared.
If INTREND is generated, the next data cannot be received until RMSR is read.

4. RMSR, RMSCR, and RMDR are cleared simultaneously to operation termination (RMEN = 0).

Figure 20-9. Operation Flow of Type B Reception Mode

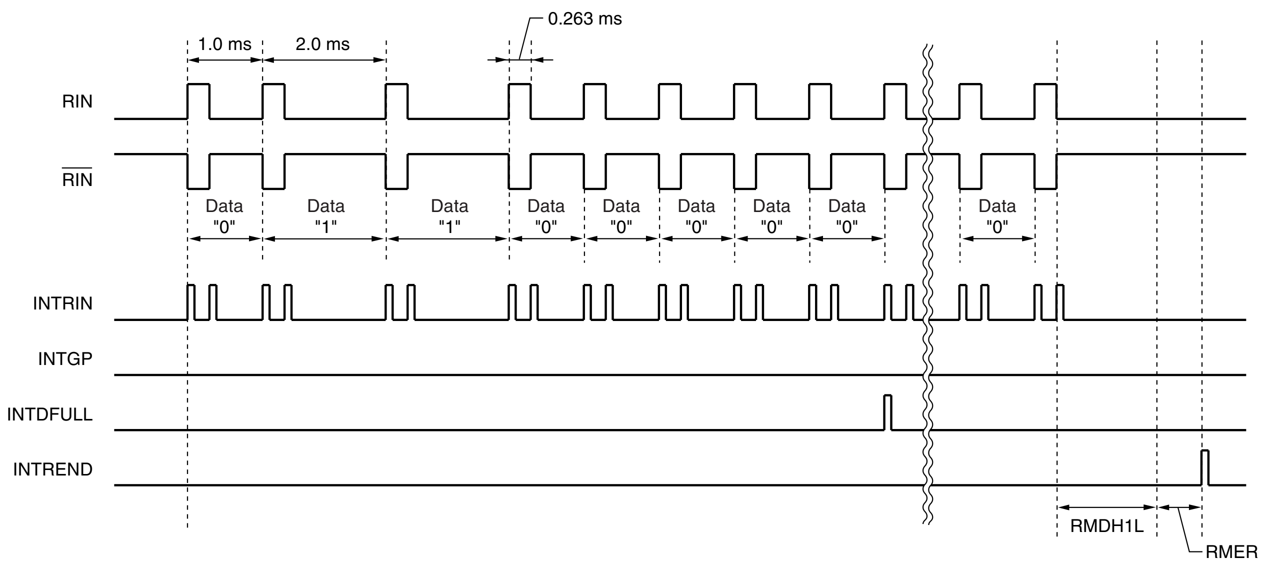


Note Read RMDR before data has been set to all the bits of RMSR.

20.4.5 Format of type C reception mode

Figure 20-10 shows the data format for type C.

Figure 20-10. Example of Type C Data Format



Remark RIN is the internally inverted signal of RIN.

20.4.6 Operation flow of type C reception mode

Figure 20-11 shows the operation flow.

Cautions 1. When INTRERR is generated, RMSR and RMSCR are automatically cleared immediately.

2. When data has been set to all the bits of RMSR, the following processing is automatically performed.

- **The value of RMSR is transferred to RMDR.**
- **INTDFULL is generated.**
- **RMSR is cleared.**

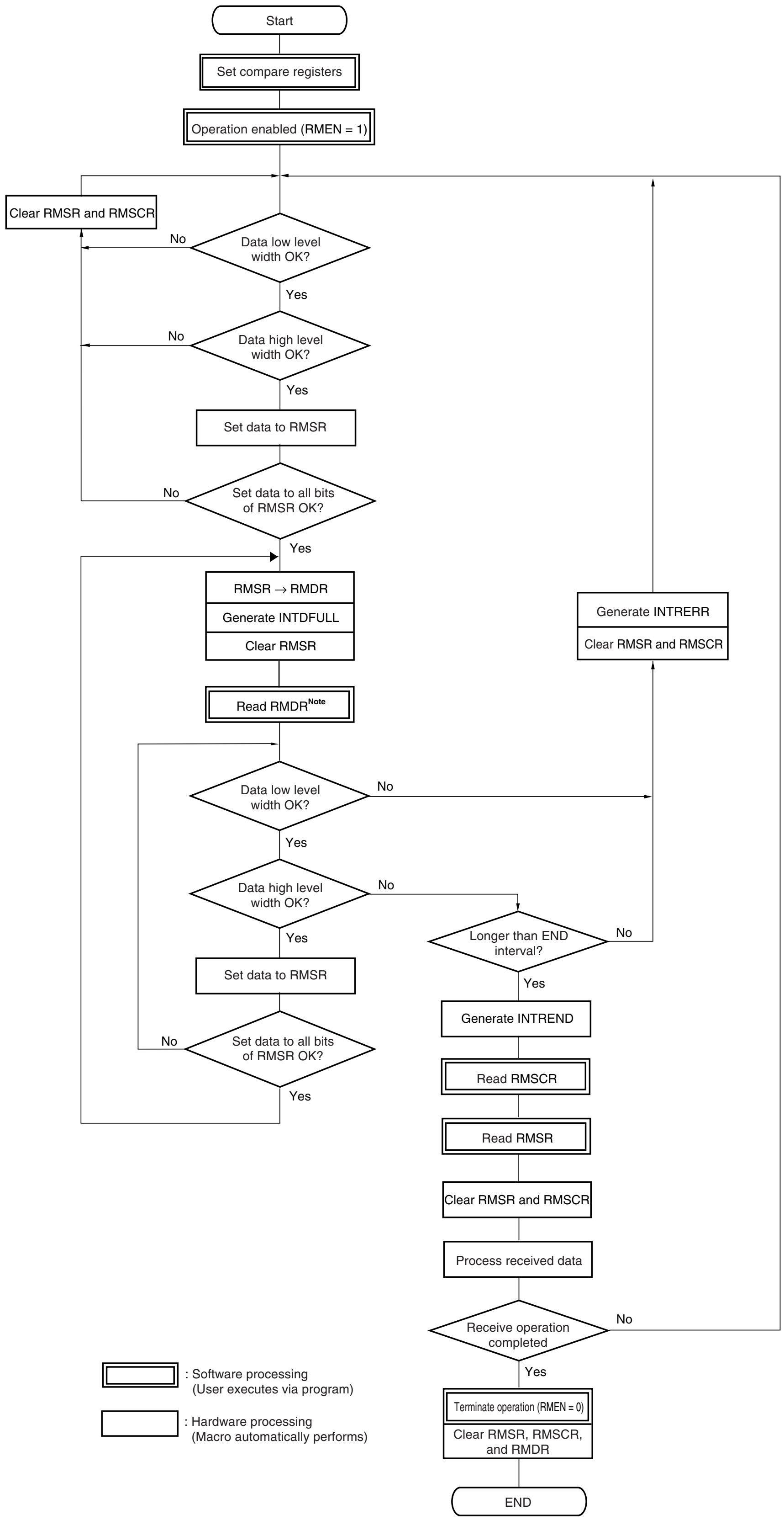
RMDR must then be read before the next data is set to all the bits of RMSR.

3. When INTREND has been generated, read RMSCR first followed by RMSR.
When RMSR has been read, RMSCR and RMSR are automatically cleared.
If INTREND is generated, the next data cannot be received until RMSR is read.

4. RMSR, RMSCR, and RMDR are cleared simultaneously to operation termination (RMEN = 0).

5. In type C reception mode, if the conditions for receiving a data low-/high-level width are not met before the first INTDFULL interrupt is generated, INTRERR and INTREND will not be generated. However, RMSR and RMSCR will be cleared.

Figure 20-11. Operation Flow of Type C Reception Mode

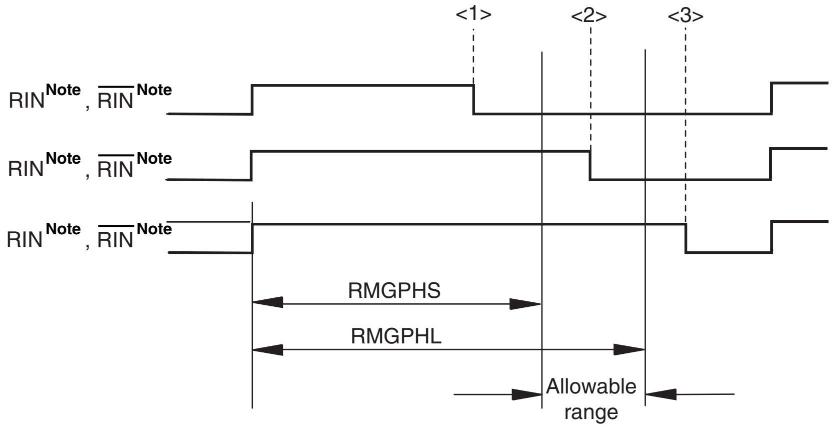


Note Read RMDR before data has been set to all the bits of RMSR.

20.4.7 Timing

Operation varies depending on the positions of the RIN input waveform below.

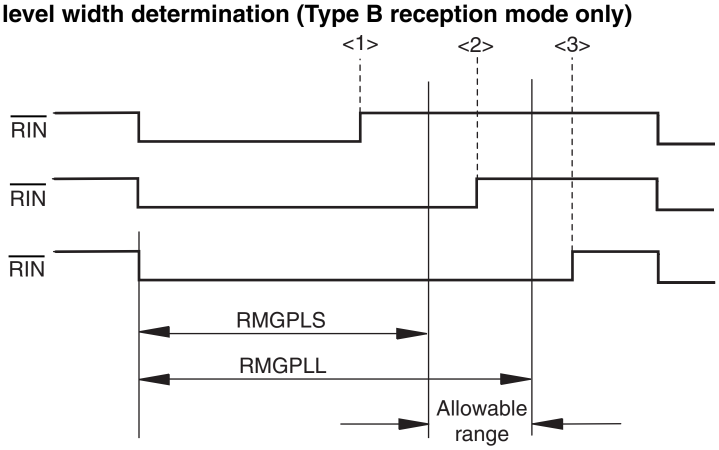
(1) Guide pulse high level width determination (Type A, Type B reception modes only)



Note RIN is generated in type A reception mode, and $\overline{RIN}$ is generated in type B reception mode.

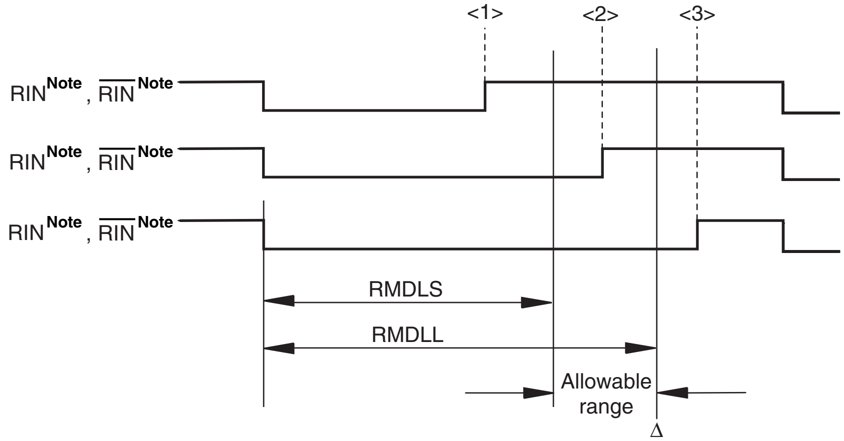
Relationship Between RMGPHS/RMGPHL/Counter	Position of Waveform	Corresponding Operation
Counter < RMGPHS	<1>: Short	Measuring guide pulse high-level width is started from the next rising edge.
RMGPHS ≤ counter < RMGPHL	<2>: Within the range	INTGP is generated. Data measurement is started.
RMGPHL ≤ counter	<3>: Long	Measuring guide pulse high-level width is started from the next rising edge.

(2) Guide pulse low level width determination (Type B reception mode only)



Relationship Between RMGPLS/RMGPLL/Counter	Position of Waveform	Corresponding Operation
Counter < RMGPLS	<1>: Short	Measuring guide pulse high-level width is started from the next rising edge.
RMGPLS ≤ counter < RMGPLL	<2>: Within the range	INTGP is generated. Data measurement is started.
RMGPLL ≤ counter	<3>: Long	Measuring guide pulse high-level width is started from the next rising edge.

(3) Data low level width determination

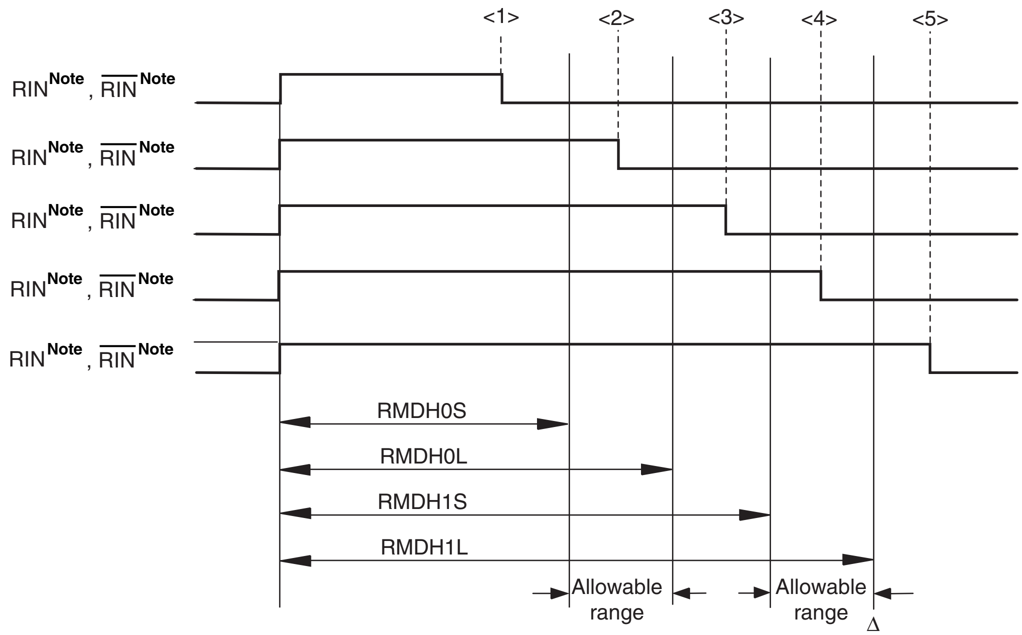


Note RIN is generated in type A reception mode, and $\overline{RIN}$ is generated in type B and type C reception modes.

Relationship Between RMDLS/RMDLL/Counter	Position of Waveform	Corresponding Operation
Counter < RMDLS	<1>: Short	Error interrupt INTRERR is generated[note]. Measuring guide pulse high-level width is started.
RMDLS ≤ counter < RMDLL	<2>: Within the range	Measuring data high-level width is started.
RMDLL ≤ counter	<3>: Long	(Type A reception mode) Measuring the end width is started from the Δ point. (Type B, Type C reception modes) Error interrupt INTRERR is generated at the Δ point.[note].

Note In type C reception mode, before the first INTDFULL interrupt is generated, INTRERR will not be generated. However, RMSR and RMSCR will be cleared.

(4) Data high level width determination



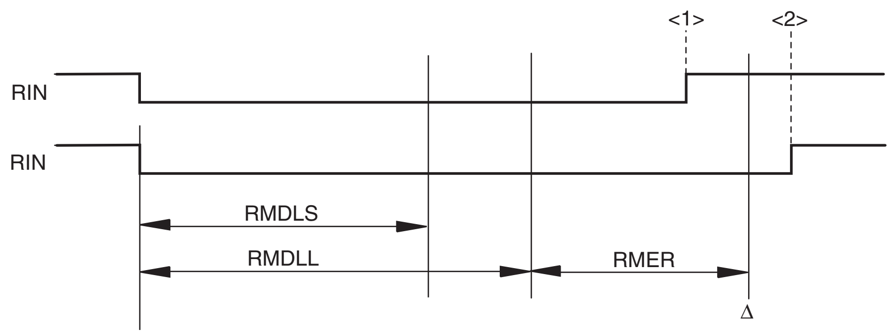
Note RIN is generated in type A reception mode, and $\overline{\text{RIN}}$ is generated in type B and type C reception modes.

Relationship Between RMDH0S/RMDH0L/RMDH1S/RMDH1L/Counter	Position of Waveform	Corresponding Operation
Counter < RMDH0S	<1>: Short	Error interrupt INTRERR is generated. Measuring the guide pulse high-level width is started at the next rising edge.
RMDH0S ≤ counter < RMDH0L	<2>: Within the range	Data 0 is received. Measuring data low-level width is started.
RMDH0L ≤ counter < RMDH1S	<3>: Outside of the range	Error interrupt INTRERR is generated. Measuring the guide pulse high-level width is started at the next rising edge.
RMDH1S ≤ counter < RMDH1L	<4>: Within the range	Data 1 is received. Measuring the data low-level width is started.
RMDH1L ≤ counter	<5>: Long	(Type A reception mode) Error interrupt INTRERR is generated at the Δ point. (Type B, Type C reception modes) Measuring the end width is started from the Δ point. Measuring the guide pulse high-level width is started at the next rising edge.

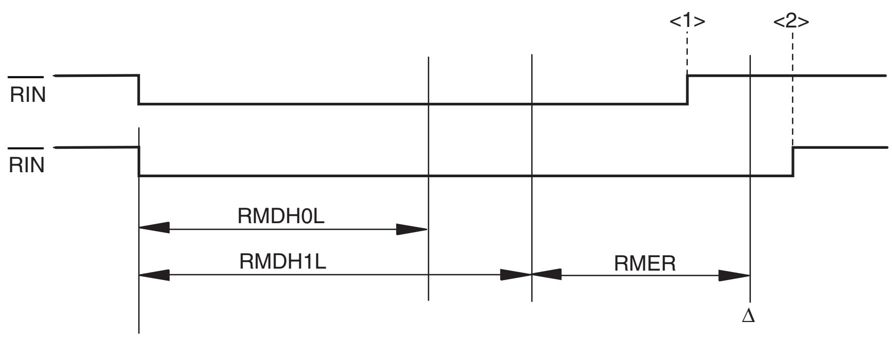
Note In type C reception mode, before the first INTDFULL interrupt is generated, INTRERR will not be generated. However, RMSR and RMSCR will be cleared.

(5) End width determination

(a) Type A reception mode



(b) Type B, Type C reception modes



Relationship Between RMER/Counter	Position of Waveform	Corresponding Operation
Counter < RMER	<1>: Short	Error interrupt INTRERR is generated[Note]. Measuring the guide pulse high-level width is started.
RMER ≤ counter	<2>: Long	INTREND is generated at the Δ point[Note]. Reception via circuit stops until RMSR is read.

Note In type C reception mode, before the first INTDFULL interrupt is generated, INTRERR and INTREND will not be generated. However, RMSR and RMSCR will be cleared.

20.4.8 Compare register setting

This remote controller receiver has the following 11 types of compare registers.

- Remote controller receive GPLS compare register (RMGPLS)
- Remote controller receive GPLL compare register (RMGPLL)
- Remote controller receive GPHS compare register (RMGPHS)
- Remote controller receive GPHL compare register (RMGPHL)
- Remote controller receive DLS compare register (RMDLS)
- Remote controller receive DLL compare register (RMDLL)
- Remote controller receive DH0S compare register (RMDH0S)
- Remote controller receive DH0L compare register (RMDH0L)
- Remote controller receive DH1S compare register (RMDH1S)
- Remote controller receive DH1L compare register (RMDH1L)
- Remote controller receive end width select register (RMER)

Use formulas (1) to (3) below to set the value of each compare register.

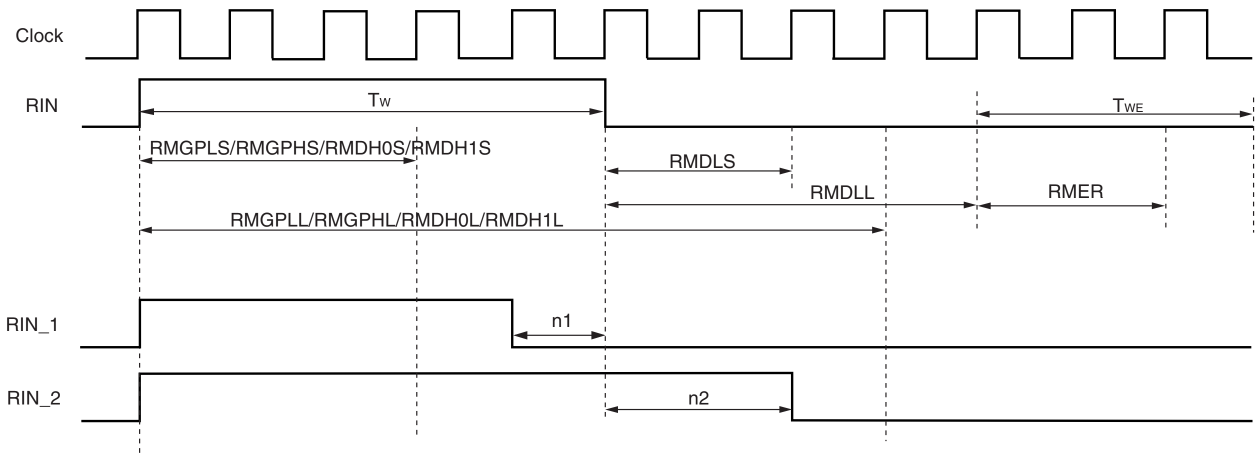
Making allowances for tolerance enables a normal reception operation, even if the RIN input waveform is RIN_1 or RIN_2 shown in Figure 20-12 due to the effect of noise.

Cautions 1. Always set each compare register while remote controller reception is disabled (RMEN = 0).

2. Set the set values so that they satisfy all the following four conditions.

- **RMGPLS < RMGPLL**
- **RMGPHS < RMGPHL**
- **RMDLS < RMDLL**
- **RMDH0S < RMDH0L ≤ RMDH1S < RMDH1L**

Figure 20-12. Setting Example (Where n1 = 1, n2 = 2)



(1) Formula for RMGPLS, RMGPHS, RMDLS, RMDH0S, and RMDH1S

$$\left[\frac{T_W \times (1 - a/100)}{1/f_{REMPRS}}\right]_{INT} - 2 - n1$$

(2) Formula for RMGPLL, RMGPHL, RMDLL, RMDH0L, and RMDH1L

$$\left[\frac{T_W \times (1 + a/100)}{1/f_{REMPRS}}\right]_{INT} + 1 + n2$$

(3) Formula for RMER

$$\left[\frac{T_{WE} \times (1 - a/100)}{1/f_{REMPRS}}\right]_{INT} - 1$$

T_W: Width of RIN input waveform
$1/f_{REMPRS}$: Width of internal operation clock cycle after division control by PRSEN
a: Tolerance (%)
$[\ ]_{INT}$: Round down the fractional portion of the value produced by the formula in the brackets.
n1, n2: Variables of waveform change caused by noise[Note1]
T_{WE}: End width of RIN input[Note2]

Notes 1. Set the values of n1 and n2 as required to meet the user's system specification.

2. This end width is counted after RMDLL.
The low-level width actually required after the last data has been received is as follows:
(RMDLL + 1 + RMER + 1) × (width of internal operation clock cycle after division control by PRSEN)

20.4.9 Error interrupt generation timing

(1) Type A reception mode

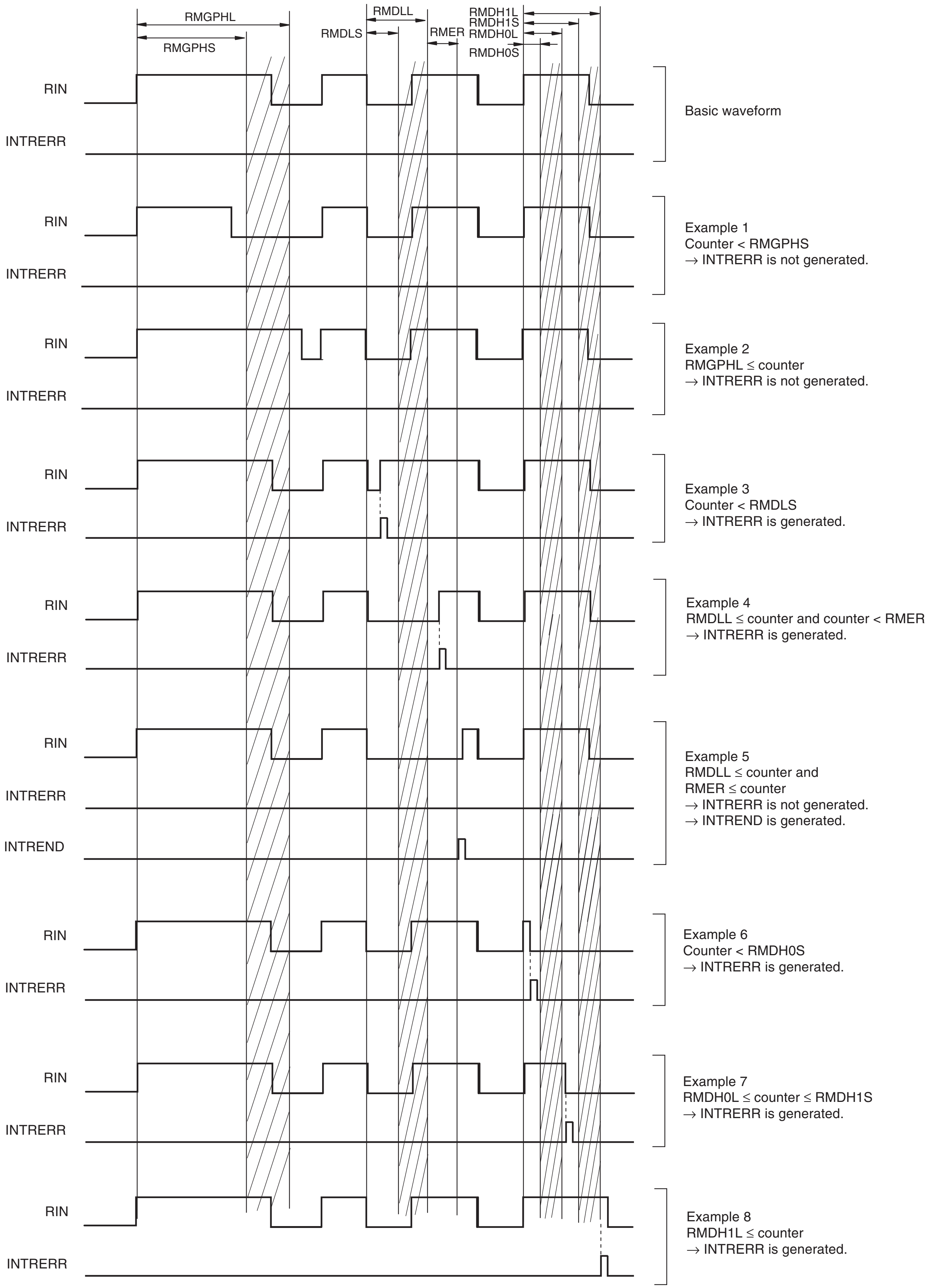
After the guide pulse has been detected normally, the INTRERR signal is generated under any of the following conditions.

- Counter $<$ RMDLS at the rising edge of RIN
- RMDLL $\leq$ counter and counter after RMDLL $<$ RMER at the rising edge of RIN
- Counter $<$ RMDH0S at the falling edge of RIN
- RMDH0L $\leq$ counter $<$ RMDH1S at the falling edge of RIN
- Register changes so that RMDH1L $\leq$ counter while RIN is at high level

The INTRERR signal is not generated until the guide pulse is detected.

Once the INTRERR signal has been generated, it will not be generated again until the next guide pulse is detected. The generation timing of the INTRERR signal is shown in Figure 20-13.

Figure 20-13. Generation Timing of INTRERR Signal (Type A reception mode)



(2) Type B reception mode

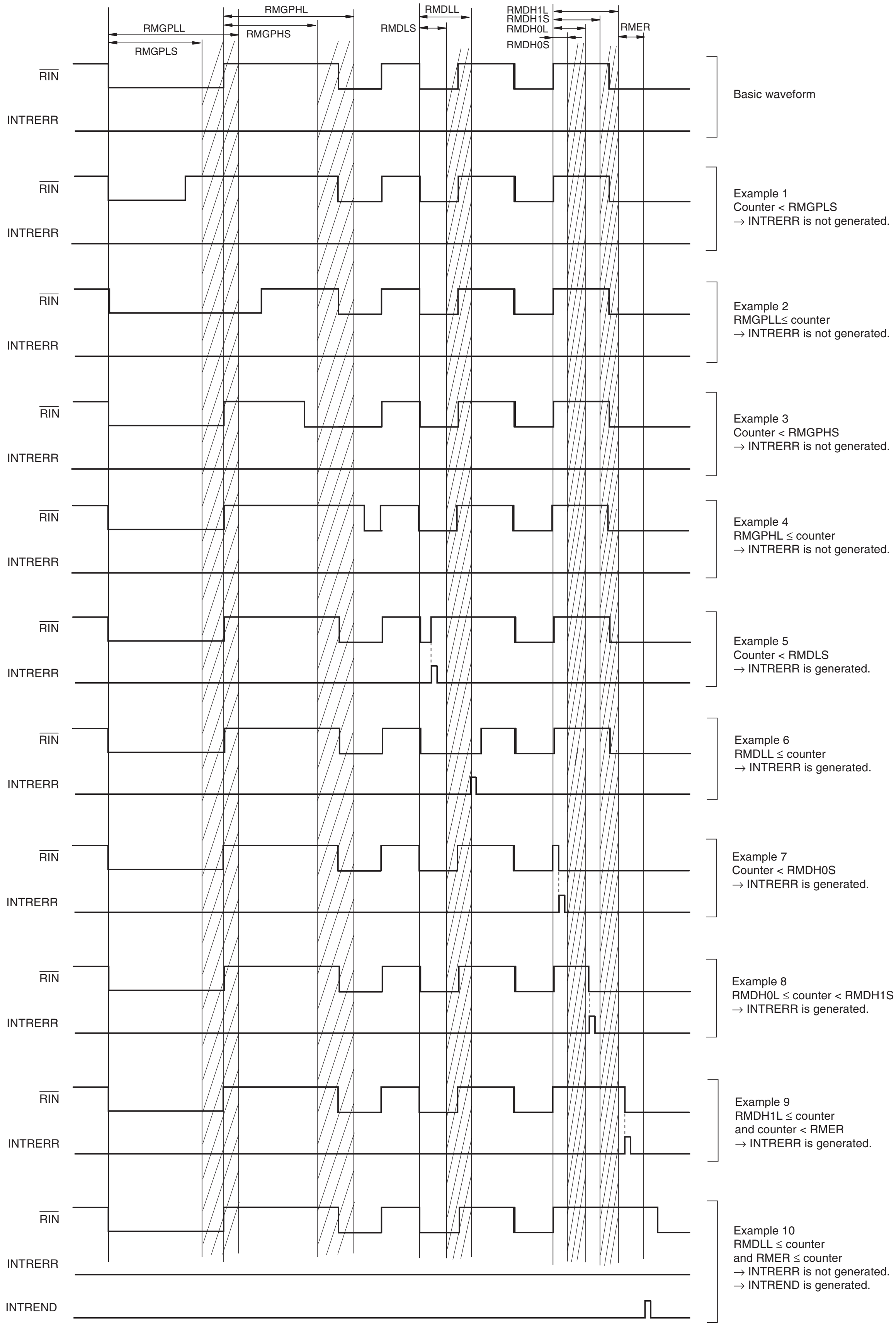
After the guide pulse has been detected normally, the INTRERR signal is generated under any of the following conditions.

- Counter < RMDLS at the rising edge of $\overline{\text{RIN}}$
- Register changes so that RMDLL ≤ counter while $\overline{\text{RIN}}$ is at low level
- Counter < RMDH0S at the falling edge of $\overline{\text{RIN}}$
- RMDH0L ≤ counter < RMDH1S at the falling edge of $\overline{\text{RIN}}$
- RMDH1L ≤ counter and counter after RMDH1L < RMER at the falling edge of $\overline{\text{RIN}}$

The INTRERR signal is not generated until the guide pulse is detected.

Once the INTRERR signal has been generated, it will not be generated again until the next guide pulse is detected. The generation timing of the INTRERR signal is shown in Figure 20-14.

Figure 20-14. Generation Timing of INTRERR Signal (Type B reception mode)



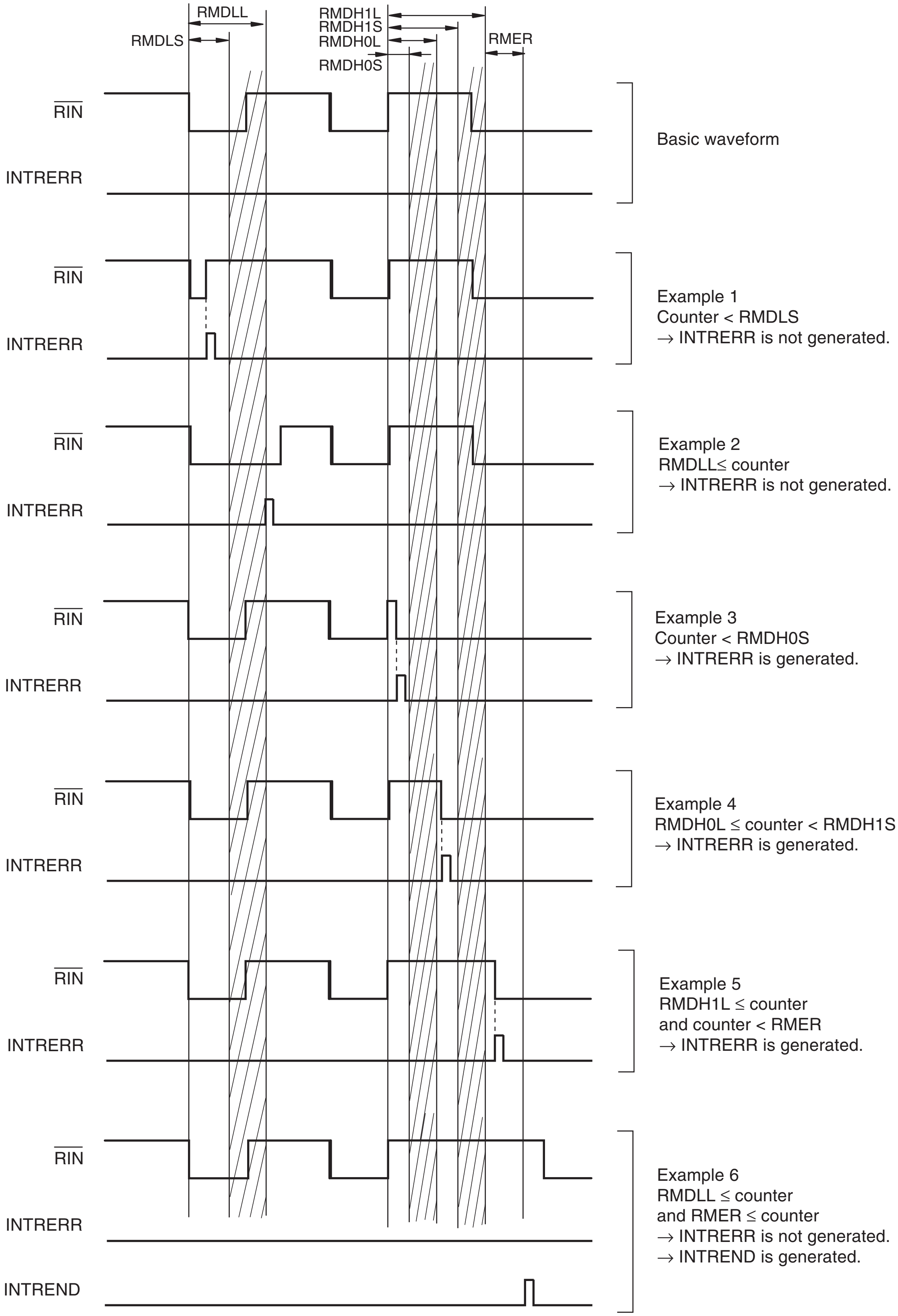
(3) Type C reception mode

The INTRERR signal is generated under any of the following conditions.

- Counter $<$ RMDLS at the rising edge of $\overline{\text{RIN}}$
- Register changes so that RMDLL $\leq$ counter while $\overline{\text{RIN}}$ is at low level
- Counter $<$ RMDH0S at the falling edge of $\overline{\text{RIN}}$
- RMDH0L $\leq$ counter $<$ RMDH1S at the falling edge of $\overline{\text{RIN}}$
- RMDH1L $\leq$ counter and counter after RMDH1L $<$ RMER at the falling edge of $\overline{\text{RIN}}$

However, before the first INTDFULL interrupt is generated, INTRERR signal will not be generated.
The generation timing of the INTRERR signal is shown in Figure 20-15.

Figure 20-15. Generation Timing of INTRERR Signal (Type C reception mode)



20.4.10 Noise elimination

This remote controller receiver provides a function that supplies the signals input from the outside to the RIN pin after eliminating noise.

Noise width can be eliminated by setting bit 5 (PRSEN) and bit 6 (NCW) of the remote controller receive control register (RMCN) as shown in Figure 20-2.

Table 20-2. Noise Elimination Width

PRSEN Division Control Signal	NCW Noise Elimination Width Control Signal	Internal Operation Clock Cycle After Division Control by PRSEN ($1/f_{REMPRS}$)	Eliminatable Noise Width
0	0	$1/f_{REM}$	Less than $1/f_{REM}$
0	1	$1/f_{REM}$	Less than $2/f_{REM}$
1	0	$2/f_{REM}$	Less than $2/f_{REM}$
1	1	$2/f_{REM}$	Less than $4/f_{REM}$

Remark f_{REM}: Source clock of remote controller counter

A noise elimination operation is performed by using the internal operation clock after division control by PRSEN.
Then, after the external input signal from RIN pin has been synchronized with the clock,
If NCW = 0, the signal after sampling is performed twice is processed as a RIN input in the circuit.
If NCW = 1, the signal after sampling is performed three times is processed as a RIN input in the circuit.

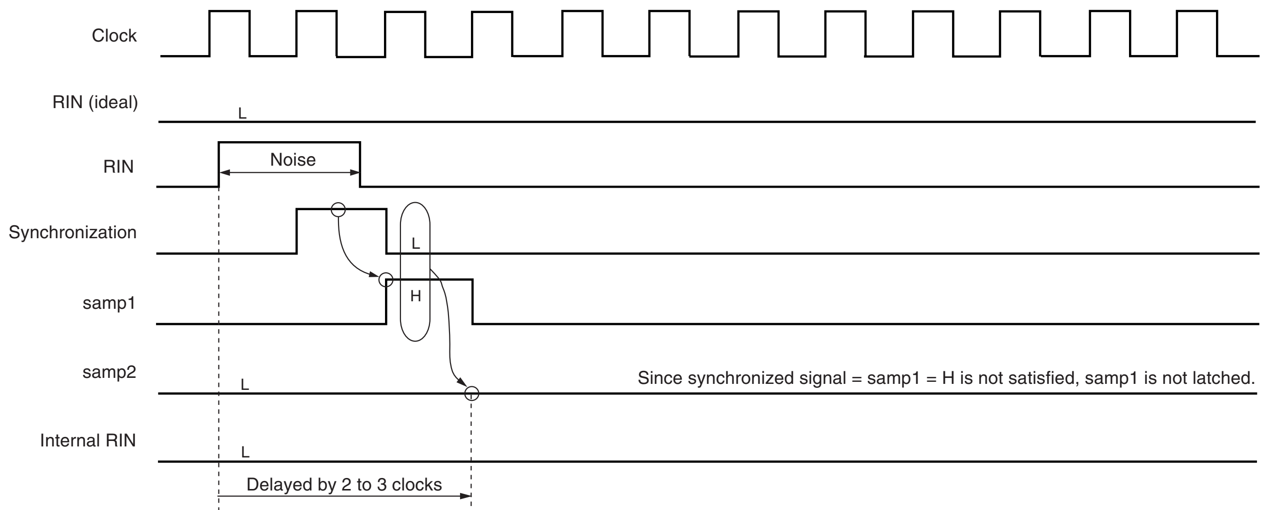
The following shows the flow of a noise elimination operation.

<1> Select whether or not the internal operation clock is divided by PRSEN.
PRSEN = 0: Not divided ($f_{REMPRS} = f_{REM}$)
PRSEN = 1: Divided ($f_{REMPRS} = f_{REM}/2$)
<2> Synchronize the external input signal from the RIN pin with the internal operation clock.
<3> Generate a signal (samp1) sampling the synchronized signal for the first time.
(The signal is later than the synchronized signal by one clock.)
<4> Generate a signal (samp2) sampling the synchronized signal and samp1 for the second time.
(When synchronized signal = samp1 = H, samp1 is latched.)
<5> Generate a signal (samp3) sampling the synchronized signal and samp2 for the third time.
(When synchronized signal = samp2 = H, samp2 is latched.)
<6> Select a signal to be the RIN input in the circuit using NCW.
NCW = 0: samp2 is processed as the RIN input in the circuit.
NCW = 1: samp3 is processed as the RIN input in the circuit.

Figure 20-16 shows an example of a noise elimination operation.

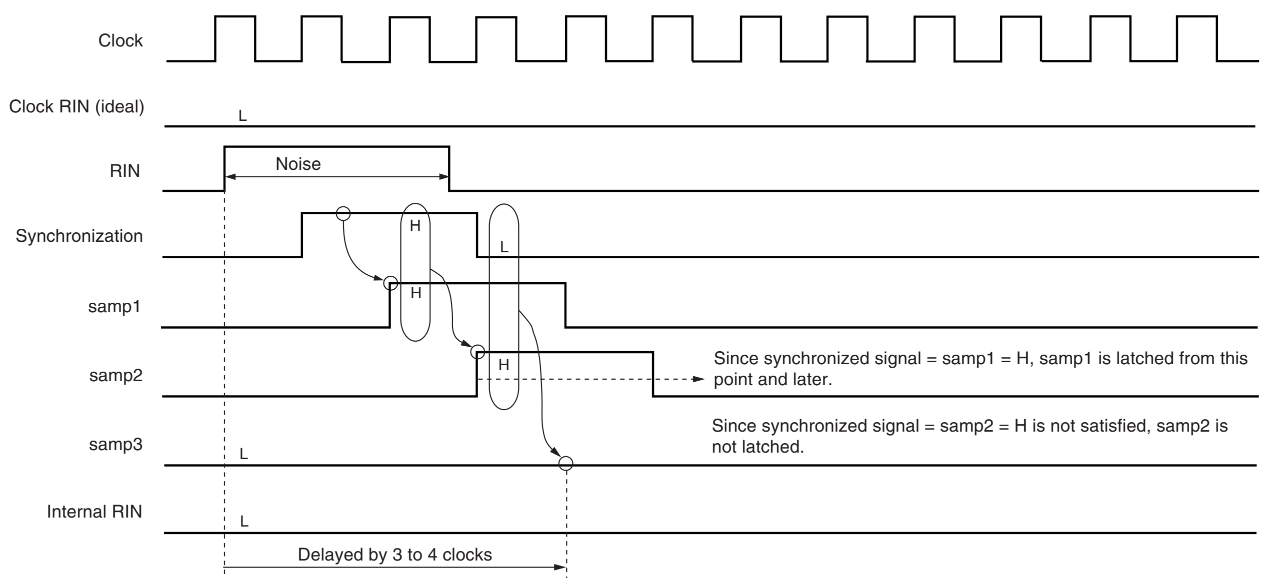
Figure 20-16. Noise Elimination Operation Example (1/2)

(a) 1-clock noise elimination (PRSEN = 0, NCW = 0)



Remark Internal RIN is a signal after synchronization and sampling are performed twice, and is therefore later than the actual signal input from the outside to the RIN pin by two to three clocks.

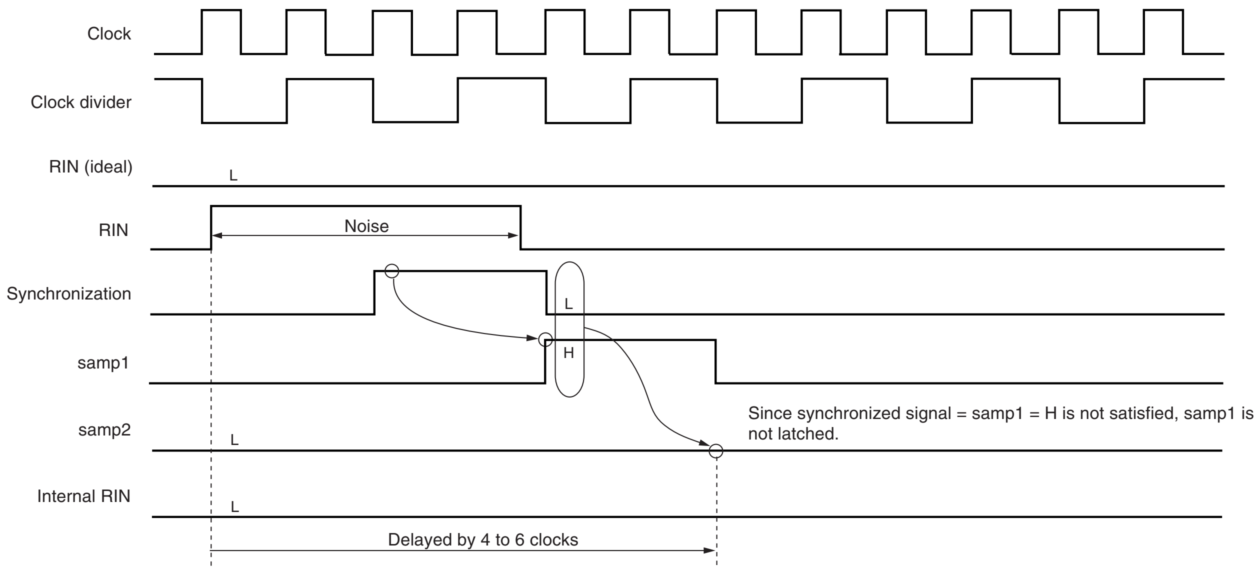
(b) 2-clock noise elimination (PRSEN = 0, NCW = 1)



Remark Internal RIN is a signal after synchronization and sampling are performed three times, and is therefore later than the actual signal input from the outside to the RIN pin by 3 to 4 clocks.

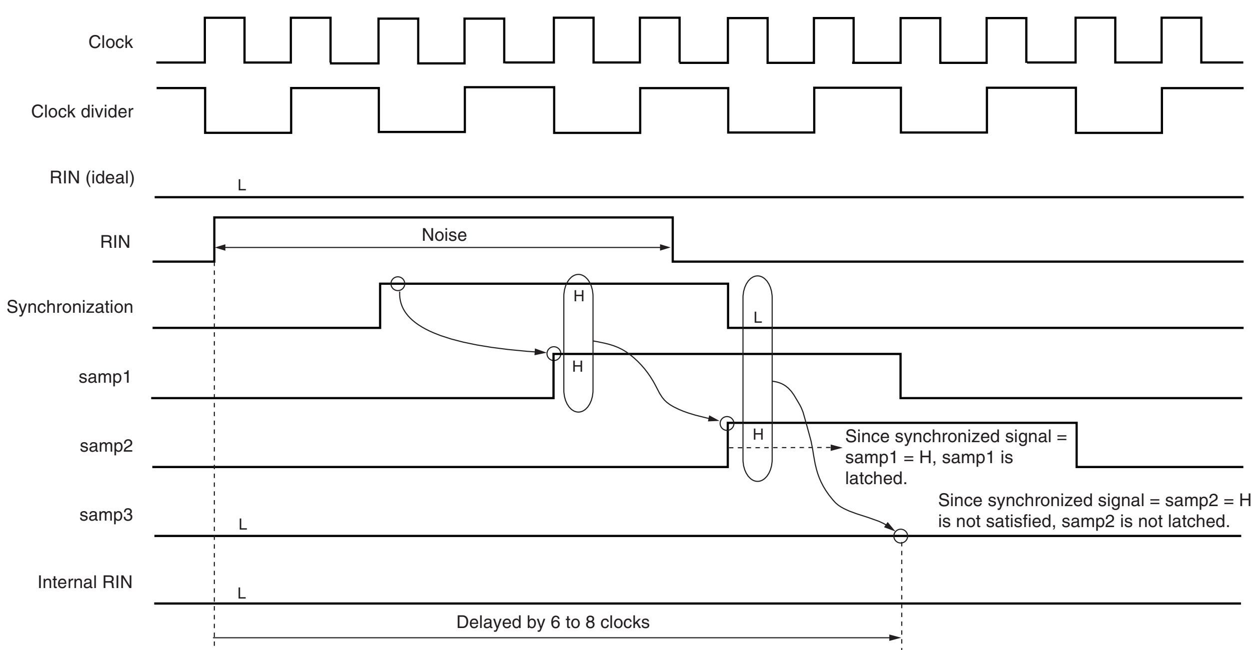
Figure 20-16. Noise Elimination Operation Example (2/2)

(c) 2-clock noise elimination (PRSEN = 1, NCW = 0)



Remark Internal RIN is a signal after synchronization and sampling are performed twice, and is therefore later than the actual signal input from the outside to the RIN pin by 4 to 6 clocks.

(d) 4-clock noise elimination (PRSEN = 1, NCW = 1)



Remark Internal RIN is a signal after synchronization and sampling are performed three times, and is therefore later than the actual signal input from the outside to the RIN pin by 6 to 8 clocks.

CHAPTER 21 INTERRUPT FUNCTIONS

21.1 Interrupt Function Types

The following two types of interrupt functions are used.

(1) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into a high interrupt priority group and a low interrupt priority group by setting the priority specification flag registers (PR0L, PR0H, PR1L, PR1H).

Multiple interrupt servicing can be applied to low-priority interrupts when high-priority interrupts are generated. If two or more interrupt requests, each having the same priority, are simultaneously generated, then they are processed according to the priority of vectored interrupt servicing. For the priority order, see **Table 21-1**.

A standby release signal is generated and STOP and HALT modes are released.

External interrupt requests and internal interrupt requests are provided as maskable interrupts.

- μPD78F047x
  External: 7, internal: 20
- μPD78F048x
  External: 7, internal: 21
- μPD78F049x
  External: 7, internal: 22

(2) Software interrupt

This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even when interrupts are disabled. The software interrupt does not undergo interrupt priority control.

21.2 Interrupt Sources and Configuration

The μPD78F047x has a total of 28 interrupt sources, the μPD78F048x has a total of 29 interrupt sources, and the μPD78F049x has a total of 30 interrupt sources, including maskable interrupts and software interrupts. In addition, they also have up to four reset sources (see **Table 21-1**).

Table 21-1. Interrupt Source List (1/2)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Maskable	0	INTLVI	Low-voltage detection[Note 3]	Internal	0004H	(A)
	1	INTP0	Pin input edge detection	External	0006H	(B)
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTP4			000EH	
	6	INTP5			0010H	
	7	INTSRE6	UART6 reception error generation	Internal	0012H	(A)
	8	INTSR6	End of UART6 reception		0014H	
	9	INTST6	End of UART6 transmission		0016H	
	10	INTCSI10/ INTST0	End of CSI10 communication/end of UART0 transmission		0018H	
	11	INTTMH1	Match between TMH1 and CMP01 (when compare register is specified)		001AH	
	12	INTTMH0	Match between TMH0 and CMP00 (when compare register is specified)		001CH	
	13	INTTM50	Match between TM50 and CR50 (when compare register is specified)		001EH	
	14	INTTM000	Match between TM00 and CR000 (when compare register is specified), TI010 pin valid edge detection (when capture register is specified)		0020H	
	15	INTTM010	Match between TM00 and CR010 (when compare register is specified), TI000 pin valid edge detection (when capture register is specified)		0022H	
	16	INTAD[Note 5]	End of 10-bit successive approximation type A/D conversion		0024H	
	17	INTSR0	End of UART0 reception or reception error generation		0026H	
	18	INTRTC	Fixed-cycle signal of real-time counter/alarm match detection		0028H	
	19	INTTM51 [Note 4]	Match between TM51 and CR51 (when compare register is specified)		002AH	
	20	INTKR	Key interrupt detection	External	002CH	(C)
	21	INTRTCI	Interval signal detection of real-time counter	Internal	002EH	(A)

Notes 1. The default priority determines the sequence of processing vectored interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 28 indicates the lowest priority.

2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 21-1.

3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is cleared to 0.

4. When 8-bit timer/event counter 51 and 8-bit timer H1 are used in the carrier generator mode, an interrupt is generated upon the timing when the INTTM5H1 signal is generated (see **Figure 8-15 Transfer Timing**).

5. μPD78F048x and 78F049x only.

Table 21-1. Interrupt Source List (2/2)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Maskable	22	INTDSAD[Note 4]	End of 16-bit ΔΣ type A/D conversion	Internal	0030H	(A)
	23	INTTM52	Match between TM52 and CR52 (when compare register is specified)		0032H	
	24	INTTMH2	Match between TMH2 and CRH2 (when compare register is specified)		0034H	
	25	INTMCG	End of Manchester code reception		0036H	
	26	INTRIN	Remote controller reception edge detection		0038H	
	27	INTRERR/ INTGP/ INTREND/ INTDFULL	Remote controller reception error occurrence Remote controller guide pulse detection Remote controller data reception completion Read request for remote controller 8-bit shift data		003AH	
	28	INTACSI	End of CSIA0 communication		003CH	
Software	–	BRK	BRK instruction execution	–	003EH	(D)
Reset	–	RESET	Reset input	–	0000H	–
		POC	Power-on clear			
		LVI	Low-voltage detection[Note 3]			
		WDT	WDT overflow			

Notes 1. The default priority determines the sequence of processing vectored interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 28 indicates the lowest priority.

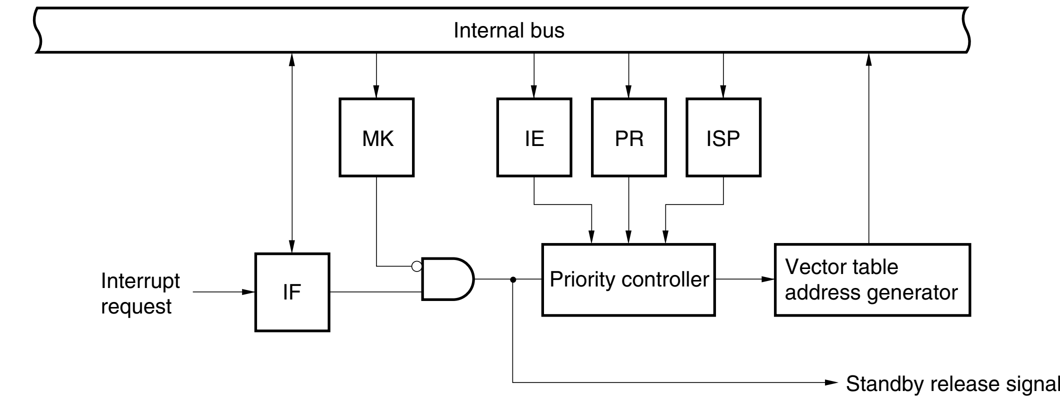
2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 21-1.

3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is set to 1.

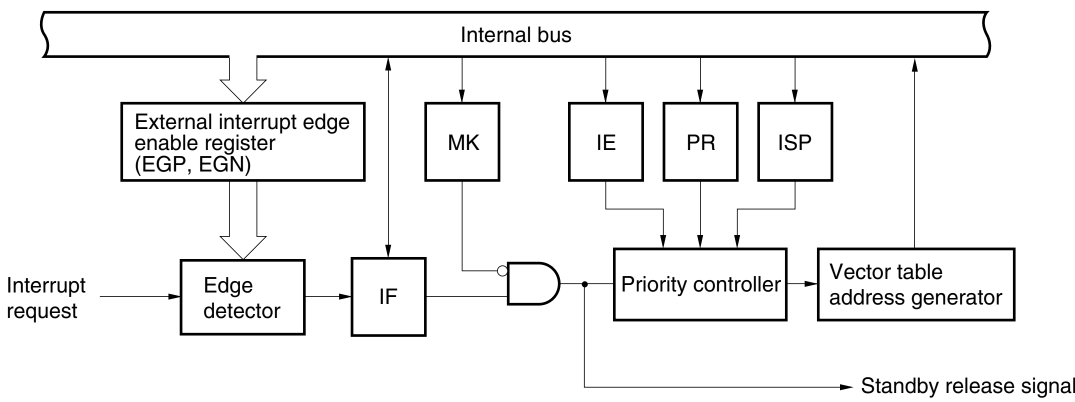
4. μPD78F049x only.

Figure 21-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal maskable interrupt



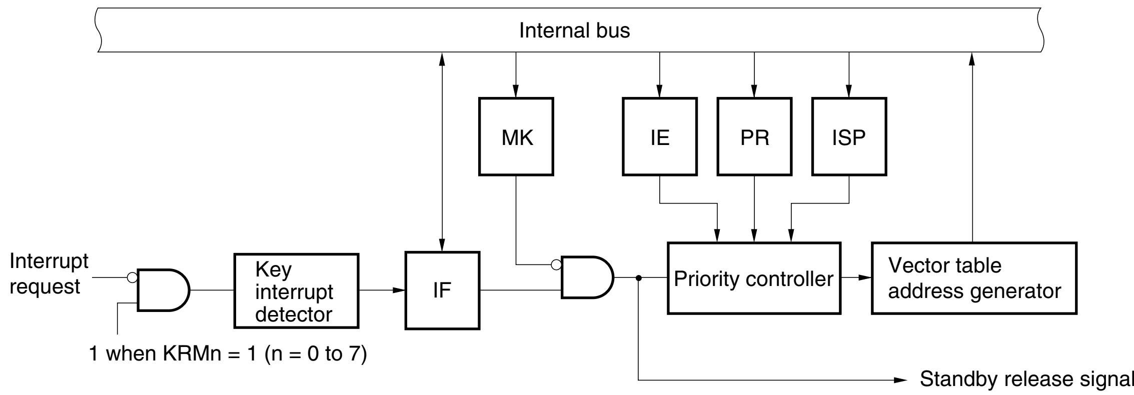
(B) External maskable interrupt (INTP0 to INTP5)



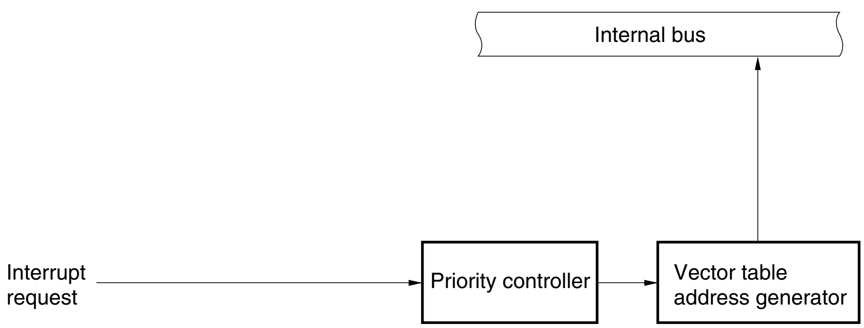
IF: Interrupt request flag
IE: Interrupt enable flag
ISP: In-service priority flag
MK: Interrupt mask flag
PR: Priority specification flag

Figure 21-1. Basic Configuration of Interrupt Function (2/2)

(C) External maskable interrupt (INTKR)



(D) Software interrupt



IF: Interrupt request flag
IE: Interrupt enable flag
ISP: In-service priority flag
MK: Interrupt mask flag
PR: Priority specification flag
KRM: Key return mode register

21.3 Registers Controlling Interrupt Functions

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag register (IF0L, IF0H, IF1L, IF1H)
- Interrupt mask flag register (MK0L, MK0H, MK1L, MK1H)
- Priority specification flag register (PR0L, PR0H, PR1L, PR1H)
- External interrupt rising edge enable register (EGP)
- External interrupt falling edge enable register (EGN)
- Program status word (PSW)

Table 21-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 21-2. Flags Corresponding to Interrupt Request Sources (1/2)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTLVI	LVIIF	IF0L	LVIMK	MK0L	LVIPR	PR0L
INTP0	PIF0		PMK0		PPR0	
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2		PMK2		PPR2	
INTP3	PIF3		PMK3		PPR3	
INTP4	PIF4		PMK4		PPR4	
INTP5	PIF5		PMK5		PPR5	
INTSRE6	SREIF6		SREMK6		SREPR6	
INTSR6	SRIF6	IF0H	SRMK6	MK0H	SRPR6	PR0H
INTST6	STIF6		STMK6		STPR6	
INTCSI10 INTST0	CSIIF10 Note 1 STIF0 Note 1		CSIMK10 Note 2 STMK0 Note 2		CSIPR10 Note 3 STPR0 Note 3	
INTTMH1	TMIFH1		TMMKH1		TMPRH1	
INTTMH0	TMIFH0		TMMKH0		TMPRH0	
INTTM50	TMIF50		TMMK50		TMPR50	
INTTM000	TMIF000		TMMK000		TMPR000	
INTTM010	TMIF010		TMMK010		TMPR010	

Notes 1. If either interrupt source INTCSI10 or INTST0 is generated, bit 2 of IF0H is set (1).
2. Bit 2 of MK0H supports both interrupt sources INTCSI10 and INTST0.
3. Bit 2 of PR0H supports both interrupt sources INTCSI10 and INTST0.

Table 21-2. Flags Corresponding to Interrupt Request Sources (2/2)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTAD[Note 1]	ADIF[Note 1]	IF1L	ADMK[Note 1]	MK1L	ADPR[Note 1]	PR1L
INTSR0	SRIF0		SRMK0		SRPR0	
INTRTC	RTCIF		RTCMK		RTCPR	
INTTM51[Note 2]	TMIF51		TMMK51		TMPR51	
INTKR	KRIF		KRMK		KRPR	
INTRTCI	RTCIIF		RTCIMK		RTCIPR	
INTDSAD[Note 3]	DSADIF[Note 3]		DSADMK[Note 3]		DASDPR[Note 3]	
INTTM52	TMIF52		TMMK52		TMPR52	
INTTMH2	TMHIF2	IF1H	TMHMK2	MK1H	TMHPR2	PR1H
INTMCG	MCGIF		MCGMK		MCGPR	
INTRIN	RINIF		RINMK		RINPR	
INTRERR INTGP INTREND INTDFULL	RERRIF[Note 4] GPIF[Note 4] RENDIF[Note 4] DFULLIF[Note 4]		RERRMK[Note 5] GPMK[Note 5] RENDMK[Note 5] DFULLMK[Note 5]		RERRPR[Note 6] GPPR[Note 6] RENDPR[Note 6] DFULLPR[Note 6]	
INTACSI	ACSIIF		ACSIMK		ACSIPR	

Notes 1. μPD78F048x and 78F049x only.

2. When 8-bit timer/event counter 51 and 8-bit timer H1 are used in the carrier generator mode, an interrupt is generated upon the timing when the INTTM5H1 signal is generated (see **Figure 8-15 Transfer Timing**).

3. μPD78F049x only.

4. If either interrupt source INTRERR, INTGP, INTREND, or INTDFULL is generated, bit 3 of IF1H is set (1).

5. Bit 3 of MK1H supports all of interrupt sources INTRERR, INTGP, INTREND, and INTDFULL.

6. Bit 3 of PR1H supports all of interrupt sources INTRERR, INTGP, INTREND, and INTDFULL.

(1) Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H)

The interrupt request flags are set to 1 when the corresponding interrupt request is generated or an instruction is executed. They are cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon reset signal generation.

When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.

IF0L, IF0H, IF1L, and IF1H are set by a 1-bit or 8-bit memory manipulation instruction. When IF0L and IF0H, and IF1L and IF1H are combined to form 16-bit registers IF0 and IF1, they are set by a 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 21-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H)

Address: FFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	SREIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIF

Address: FFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	TMIF010	TMIF000	TMIF50	TMIFH0	TMIFH1	CSIIF10 STIF0	STIF6	SRIF6

Address: FFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
<R> IF1L	TMIF52	DSADIF Note 2	RTCIIF	KRIF	TMIF51	RTCIF	SRIF0	ADIF Note 1

Address: FFE3H After reset: 00H R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
IF1H	0	0	0	ACSIIF	RERRIF GPIF RENIF DFULLIF	RINIF	MCGIF	TMHIF2

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Notes 1. μPD78F048x and 78F049x only.

2. μPD78F049x only.

Cautions 1. Be sure to clear bits 5 to 7 of IF1H to 0.

2. When operating a timer, serial interface, or A/D converter after standby release, operate it once after clearing the interrupt request flag. An interrupt request flag may be set by noise.

Cautions 3. When manipulating a flag of the interrupt request flag register, use a 1-bit memory manipulation instruction (CLR1). When describing in C language, use a bit manipulation instruction such as "IF0L.0 = 0;" or "_asm("clr1 IF0L, 0");" because the compiled assembler must be a 1-bit memory manipulation instruction (CLR1).
If a program is described in C language using an 8-bit memory manipulation instruction such as "IF0L &= 0xfe;" and compiled, it becomes the assembler of three instructions.

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

In this case, even if the request flag of another bit of the same interrupt request flag register (IF0L) is set to 1 at the timing between "mov a, IF0L" and "mov IF0L, a", the flag is cleared to 0 at "mov IF0L, a". Therefore, care must be exercised when using an 8-bit memory manipulation instruction in C language.

(2) Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing.

MK0L, MK0H, MK1L, and MK1H are set by a 1-bit or 8-bit memory manipulation instruction. When MK0L and MK0H, and MK1L and MK1H are combined to form 16-bit registers MK0 and MK1, they are set by a 16-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 21-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H)

Address: FFE4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	SREMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK

Address: FFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	TMMK010	TMMK000	TMMK50	TMMKH0	TMMKH1	CSIMK10 STMK0	STMK6	SRMK6

Address: FFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
<R> MK1L	TMMK52	DASDMK[Note 2]	RTCIMK	KRMK	TMMK51	RTCMK	SRMK0	ADMK[Note 1]

Address: FFE7H After reset: FFH R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
MK1H	1	1	1	ACSIMK	RERRMK GPMK RENDMK DFULLMK	RINMK	MCGMK	TMHMK2

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Notes 1. μPD78F048x and 78F049x only.

2. μPD78F049x only.

Caution Be sure to set bits 5 to 7 of MK1H to 1.

(3) Priority specification flag registers (PR0L, PR0H, PR1L, PR1H)

The priority specification flag registers are used to set the corresponding maskable interrupt priority order.

PR0L, PR0H, PR1L, and PR1H are set by a 1-bit or 8-bit memory manipulation instruction. If PR0L and PR0H, and PR1L and PR1H are combined to form 16-bit registers PR0 and PR1, they are set by a 16-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 21-4. Format of Priority Specification Flag Registers (PR0L, PR0H, PR1L, PR1H)

Address: FFE8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR0L	SREPR6	PPR5	PPR4	PPR3	PPR2	PPR1	PPR0	LVIPR

Address: FFE9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR0H	TMPR010	TMPR000	TMPR50	TMPRH0	TMPRH1	CSIPR10 STPR0	STPR6	SRPR6

Address: FFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
<R> PR1L	TMPR52	DSADPR[Note 2]	RTCIPR	KRPR	TMPR51	RTCPR	SRPR0	ADPR[Note 1]

Address: FFEBH After reset: FFH R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
PR1H	1	1	1	ACSIPR	RERRPR GPPR RENDPR DFULLPR	RINPR	MCGPR	TMHPR2

XXPRX	Priority level selection
0	High priority level
1	Low priority level

Notes 1. μPD78F048x and 78F049x only.

2. μPD78F049x only.

Caution Be sure to set bits 5 to 7 of PR1H to 1.

(4) External interrupt rising edge enable register (EGP), external interrupt falling edge enable register (EGN)

These registers specify the valid edge for INTP0 to INTP5.

EGP and EGN are set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 21-5. Format of External Interrupt Rising Edge Enable Register (EGP) and External Interrupt Falling Edge Enable Register (EGN)

Address: FF48H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP	0	0	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN	0	0	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 5)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 21-3 shows the ports corresponding to EGPn and EGNn.

Table 21-3. Ports Corresponding to EGPn and EGNn

Detection Enable Register		Edge Detection Port	Interrupt Request Signal
EGP0	EGN0	P120/EXLVI	INTP0
EGP1	EGN1	P34/TI52/TI010/TO00/RTC1HZ	INTP1
EGP2	EGN2	P33/TI000/RTCDIV/RTCCL/BUZ	INTP2
EGP3	EGN3	P31/TOH1	INTP3
EGP4	EGN4	P14/$\overline{\text{SCKA0}}$	INTP4
EGP5	EGN5	P30	INTP5

Caution Select the port mode by clearing EGPn and EGNn to 0 because an edge may be detected when the external interrupt function is switched to the port function.

Remark n = 0 to 5

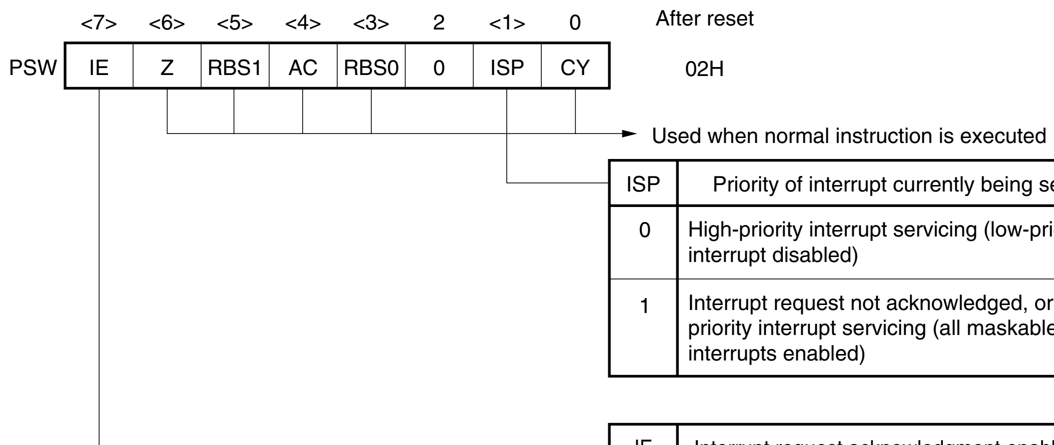
(5) Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current status for an interrupt request. The IE flag that sets maskable interrupt enable/disable and the ISP flag that controls multiple interrupt servicing are mapped to the PSW.

Besides 8-bit read/write, this register can carry out operations using bit manipulation instructions and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged, if the BRK instruction is executed, the contents of the PSW are automatically saved into a stack and the IE flag is reset to 0. If a maskable interrupt request is acknowledged, the contents of the priority specification flag of the acknowledged interrupt are transferred to the ISP flag. The PSW contents are also saved into the stack with the PUSH PSW instruction. They are restored from the stack with the RETI, RETB, and POP PSW instructions.

Reset signal generation sets PSW to 02H.

Figure 21-6. Format of Program Status Word



ISP	Priority of interrupt currently being serviced
0	High-priority interrupt servicing (low-priority interrupt disabled)
1	Interrupt request not acknowledged, or low-priority interrupt servicing (all maskable interrupts enabled)

IE	Interrupt request acknowledgment enable/disable
0	Disabled
1	Enabled

21.4 Interrupt Servicing Operations

21.4.1 Maskable interrupt acknowledgment

A maskable interrupt becomes acknowledgeable when the interrupt request flag is set to 1 and the mask (MK) flag corresponding to that interrupt request is cleared to 0. A vectored interrupt request is acknowledged if interrupts are in the interrupt enabled state (when the IE flag is set to 1). However, a low-priority interrupt request is not acknowledged during servicing of a higher priority interrupt request (when the ISP flag is reset to 0).

The times from generation of a maskable interrupt request until vectored interrupt servicing is performed are listed in Table 21-4 below.

For the interrupt request acknowledgment timing, see **Figures 21-8** and **21-9**.

Table 21-4. Time from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time[Note]
When ××PR = 0	7 clocks	32 clocks
When ××PR = 1	8 clocks	33 clocks

Note If an interrupt request is generated just before a divide instruction, the wait time becomes longer.

Remark 1 clock: $1/f_{CPU}$ (f_{CPU}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.

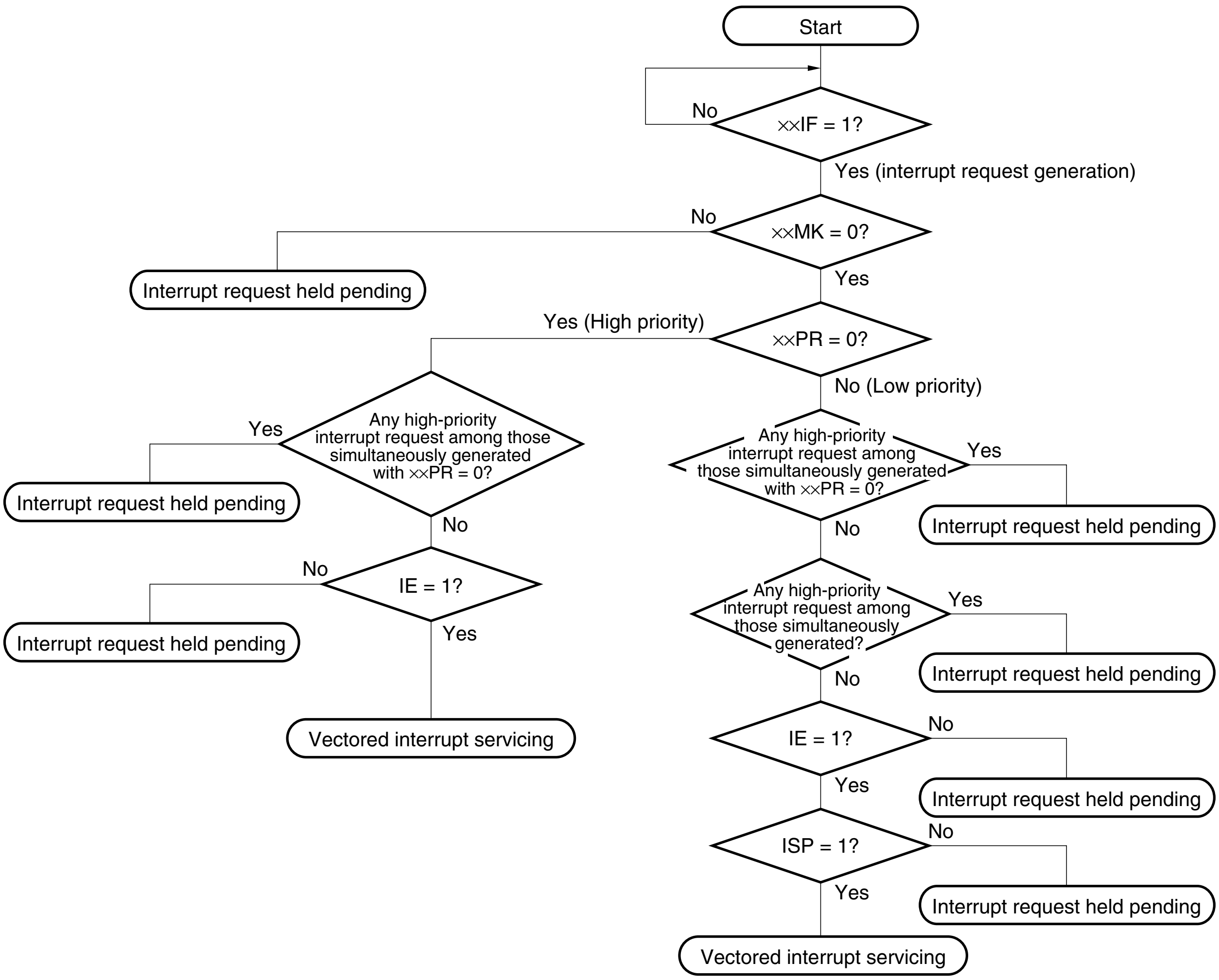
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 21-7 shows the interrupt request acknowledgment algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag is reset (0), and the contents of the priority specification flag corresponding to the acknowledged interrupt are transferred to the ISP flag. The vector table data determined for each interrupt request is the loaded into the PC and branched.

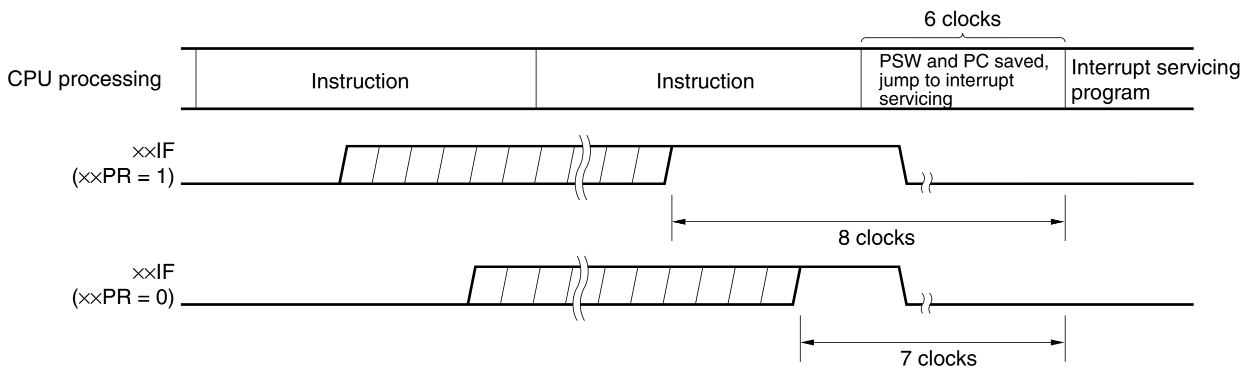
Restoring from an interrupt is possible by using the RETI instruction.

Figure 21-7. Interrupt Request Acknowledgment Processing Algorithm



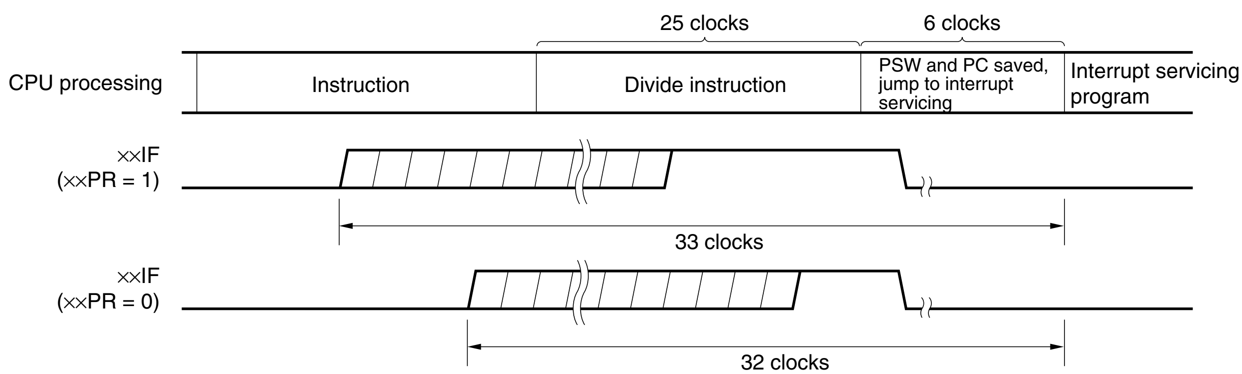
××IF: Interrupt request flag
××MK: Interrupt mask flag
××PR: Priority specification flag
IE: Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)
ISP: Flag that indicates the priority level of the interrupt currently being serviced (0 = high-priority interrupt servicing, 1 = No interrupt request acknowledged, or low-priority interrupt servicing)

Figure 21-8. Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

Figure 21-9. Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

21.4.2 Software interrupt request acknowledgment

A software interrupt acknowledge is acknowledged by BRK instruction execution. Software interrupts cannot be disabled.

If a software interrupt request is acknowledged, the contents are saved into the stacks in the order of the program status word (PSW), then program counter (PC), the IE flag is reset (0), and the contents of the vector table (003EH, 003FH) are loaded into the PC and branched.

Restoring from a software interrupt is possible by using the RETB instruction.

Caution Do not use the RETI instruction for restoring from the software interrupt.

21.4.3 Multiple interrupt servicing

Multiple interrupt servicing occurs when another interrupt request is acknowledged during execution of an interrupt.

Multiple interrupt servicing does not occur unless the interrupt request acknowledgment enabled state is selected (IE = 1). When an interrupt request is acknowledged, interrupt request acknowledgment becomes disabled (IE = 0). Therefore, to enable multiple interrupt servicing, it is necessary to set (1) the IE flag with the EI instruction during interrupt servicing to enable interrupt acknowledgment.

Moreover, even if interrupts are enabled, multiple interrupt servicing may not be enabled, this being subject to interrupt priority control. Two types of priority control are available: default priority control and programmable priority control. Programmable priority control is used for multiple interrupt servicing.

In the interrupt enabled state, if an interrupt request with a priority equal to or higher than that of the interrupt currently being serviced is generated, it is acknowledged for multiple interrupt servicing. If an interrupt with a priority lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for multiple interrupt servicing. Interrupt requests that are not enabled because interrupts are in the interrupt disabled state or because they have a lower priority are held pending. When servicing of the current interrupt ends, the pending interrupt request is acknowledged following execution of at least one main processing instruction execution.

Table 21-5 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 21-10 shows multiple interrupt servicing examples.

Table 21-5. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing

Multiple Interrupt Request / Interrupt Being Serviced		Maskable Interrupt Request				Software Interrupt Request
		PR = 0		PR = 1		
		IE = 1	IE = 0	IE = 1	IE = 0	
Maskable interrupt	ISP = 0	○	×	×	×	○
	ISP = 1	○	×	○	×	○
Software interrupt		○	×	○	×	○

Remarks 1. ○: Multiple interrupt servicing enabled

2. ×: Multiple interrupt servicing disabled

3. ISP and IE are flags contained in the PSW.

ISP = 0: An interrupt with higher priority is being serviced.

ISP = 1: No interrupt request has been acknowledged, or an interrupt with a lower priority is being serviced.

IE = 0: Interrupt request acknowledgment is disabled.

IE = 1: Interrupt request acknowledgment is enabled.

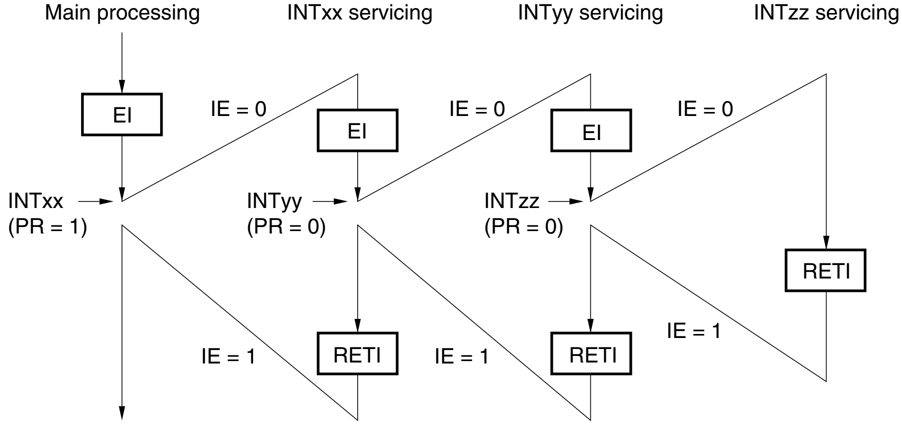
4. PR is a flag contained in PR0L, PR0H, PR1L, and PR1H.

PR = 0: Higher priority level

PR = 1: Lower priority level

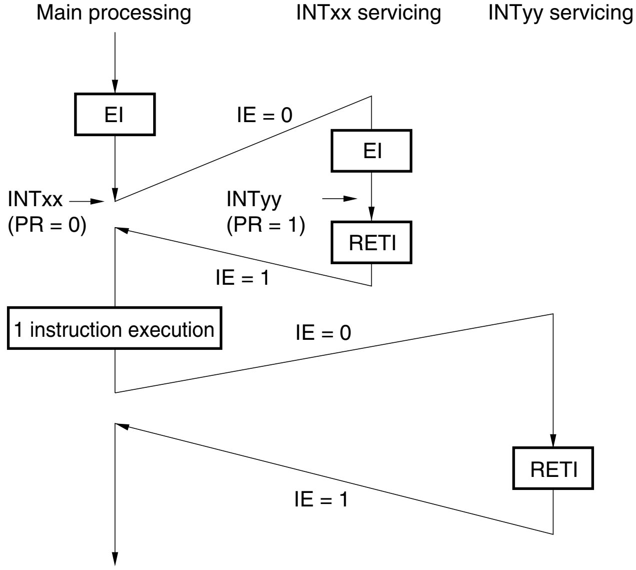
Figure 21-10. Examples of Multiple Interrupt Servicing (1/2)

Example 1. Multiple interrupt servicing occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledgment.

Example 2. Multiple interrupt servicing does not occur due to priority control

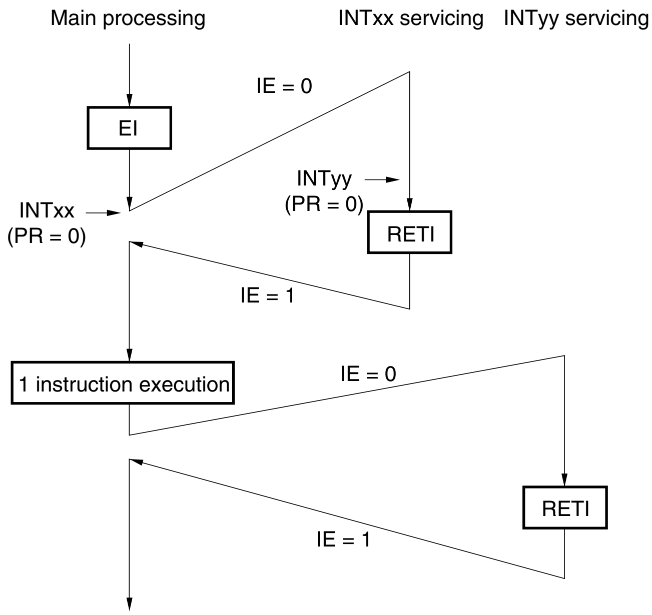


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level
PR = 1: Lower priority level
IE = 0: Interrupt request acknowledgment disabled

Figure 21-10. Examples of Multiple Interrupt Servicing (2/2)

Example 3. Multiple interrupt servicing does not occur because interrupts are not enabled



Interrupts are not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level
IE = 0: Interrupt request acknowledgment disabled

21.4.4 Interrupt request hold

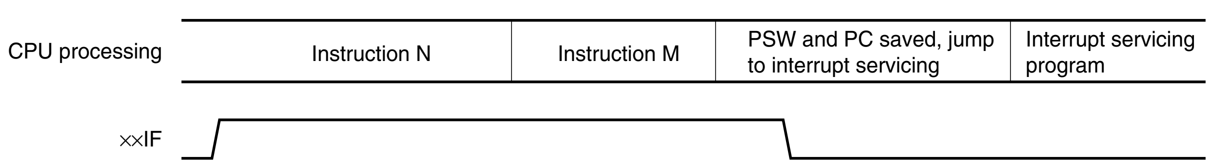
There are instructions where, even if an interrupt request is issued for them while another instruction is being executed, request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV A, PSW
- MOV PSW, A
- MOV1 PSW. bit, CY
- MOV1 CY, PSW. bit
- AND1 CY, PSW. bit
- OR1 CY, PSW. bit
- XOR1 CY, PSW. bit
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- PUSH PSW
- POP PSW
- BT PSW. bit, $addr16
- BF PSW. bit, $addr16
- BTCLR PSW. bit, $addr16
- EI
- DI
- Manipulation instructions for the IF0L, IF0H, IF1L, IF1H, MK0L, MK0H, MK1L, MK1H, PR0L, PR0H, PR1L, and PR1H registers.

Caution The BRK instruction is not one of the above-listed interrupt request hold instructions. However, the software interrupt activated by executing the BRK instruction causes the IE flag to be cleared. Therefore, even if a maskable interrupt request is generated during execution of the BRK instruction, the interrupt request is not acknowledged.

Figure 21-11 shows the timing at which interrupt requests are held pending.

Figure 21-11. Interrupt Request Hold



Remarks 1. Instruction N: Interrupt request hold instruction

2. Instruction M: Instruction other than interrupt request hold instruction

3. The ××PR (priority level) values do not affect the operation of ××IF (interrupt request).

CHAPTER 22 KEY INTERRUPT FUNCTION

22.1 Functions of Key Interrupt

A key interrupt (INTKR) can be generated by setting the key return mode register (KRM) and inputting a falling edge to the key interrupt input pins (KR0 to KR7).

Table 22-1. Assignment of Key Interrupt Detection Pins

Flag	Description
KRM0	Controls KR0 signal in 1-bit units.
KRM1	Controls KR1 signal in 1-bit units.
KRM2	Controls KR2 signal in 1-bit units.
KRM3	Controls KR3 signal in 1-bit units.
KRM4	Controls KR4 signal in 1-bit units.
KRM5	Controls KR5 signal in 1-bit units.
KRM6	Controls KR6 signal in 1-bit units.
KRM7	Controls KR7 signal in 1-bit units.

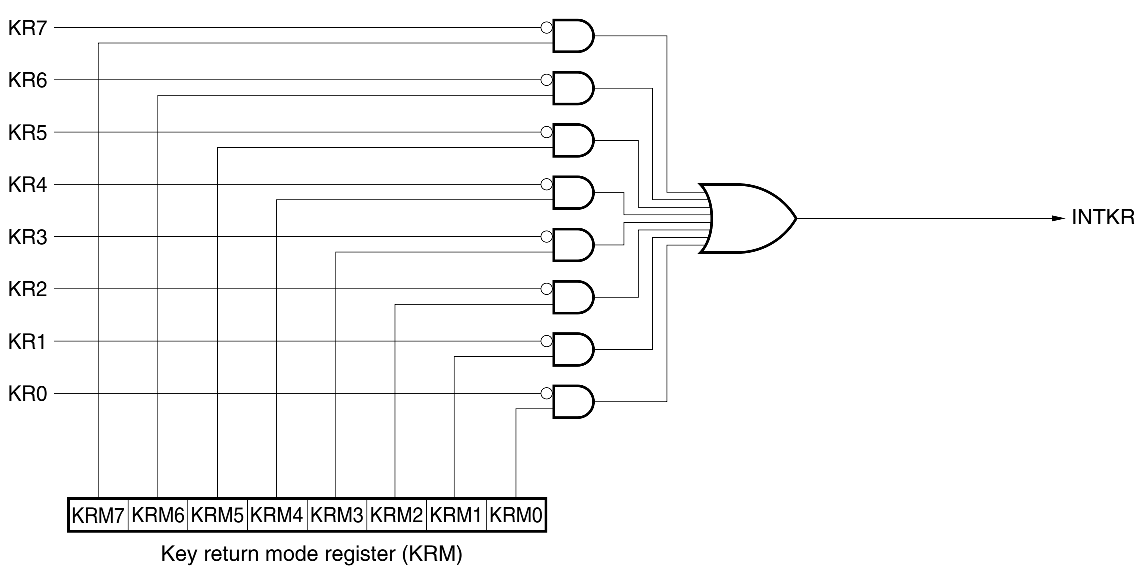
22.2 Configuration of Key Interrupt

The key interrupt includes the following hardware.

Table 22-2. Configuration of Key Interrupt

Item	Configuration
Control register	Key return mode register (KRM)

Figure 22-1. Block Diagram of Key Interrupt



22.3 Register Controlling Key Interrupt

(1) Key return mode register (KRM)

This register is used to set whether to detect a key interrupt (INTKR) when a falling edge of the key interrupt input pins (KR0 to KR7) is generated.

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

Reset signal generation clears KRM to 00H.

Figure 22-2. Format of Key Return Mode Register (KRM)

Address: FF6EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRM	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	Key interrupt mode control (n = 0-7)
0	Does not detect key interrupt signal
1	Detects key interrupt signal

Cautions 1. When setting the KRMn bit to 1 in order to use the key interrupt function, set the PU4n bit of the corresponding pull-up resistor option register 4 (PU4) to 1.

2. If KRM is changed, the interrupt request flag may be set. Therefore, disable interrupts and then change the KRM register. Clear the interrupt request flag and enable interrupts.

3. If detection of key interrupt signals is disabled (KRMn = 0), the corresponding P4n pin can be used as a normal port.

4. To use the P40/KR0/V_{LC3} pin for the key interrupt function (KR0), use the LCD display mode register (LCDM) to set it other than to the 1/4 bias method. If the P40/KR0/V_{LC3} pin is set to the 1/4 bias method, it will function as V_{LC3}.

<R> **5. When using the KRn pin as a segment key scan input pin while using the segment key scan function (KSON = 1), set KRMn to 1. When not using the KRn pin as a segment key scan input pin, set KRMn to 0 (see Figure 18-7).**

CHAPTER 23 STANDBY FUNCTION

23.1 Standby Function and Configuration

23.1.1 Standby function

The standby function is designed to reduce the operating current of the system. The following two modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. In the HALT mode, the CPU operation clock is stopped. If the high-speed system clock oscillator, internal high-speed oscillator, internal low-speed oscillator, or subsystem clock oscillator is operating before the HALT mode is set, oscillation of each clock continues. In this mode, the operating current is not decreased as much as in the STOP mode, but the HALT mode is effective for restarting operation immediately upon interrupt request generation and carrying out intermittent operations frequently.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the high-speed system clock oscillator and internal high-speed oscillator stop, stopping the whole system, thereby considerably reducing the CPU operating current.

Because this mode can be cleared by an interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure the oscillation stabilization time after the STOP mode is released when the X1 clock is selected, select the HALT mode if it is necessary to start processing immediately upon interrupt request generation.

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latches and output buffer statuses are also held.

Cautions 1. The STOP mode can be used only when the CPU is operating on the main system clock. The subsystem clock oscillation cannot be stopped. The HALT mode can be used when the CPU is operating on either the main system clock or the subsystem clock.

2. When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction.

3. The following sequence is recommended for operating current reduction of the 10-bit successive approximation type A/D converter when the standby function is used: First clear bit 7 (ADCS) and bit 0 (ADCE) of the A/D converter mode register (ADM) to 0 to stop the A/D conversion operation, and then execute the STOP instruction.

The following sequence is recommended for operating current reduction of the 16-bit ΔΣ type A/D converter when the standby function is used: First clear bit 7 (ADDPON) and bit 6 (ADDCE) of the 16-bit ΔΣ type A/D converter mode register (ADDCTL0) to 0 to stop the A/D conversion operation, and then execute the STOP instruction.

23.1.2 Registers controlling standby function

The standby function is controlled by the following two registers.

- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)

Remark For the registers that start, stop, or select the clock, see **CHAPTER 5 CLOCK GENERATOR**.

(1) Oscillation stabilization time counter status register (OSTC)

This is the register that indicates the count status of the X1 clock oscillation stabilization time counter. When X1 clock oscillation starts with the internal high-speed oscillation clock or subsystem clock used as the CPU clock, the X1 clock oscillation stabilization time can be checked.

OSTC can be read by a 1-bit or 8-bit memory manipulation instruction.

When reset is released (reset by $\overline{\text{RESET}}$ input, POC, LVI, and WDT), the STOP instruction and MSTOP (bit 7 of MOC register) = 1 clear OSTC to 00H.

Figure 23-1. Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFA3H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
OSTC	0	0	0	MOST11	MOST13	MOST14	MOST15	MOST16

MOST11	MOST13	MOST14	MOST15	MOST16	Oscillation stabilization time status	
						fx = 10 MHz
1	0	0	0	0	2^{11}/fx min.	204.8 μs min.
1	1	0	0	0	2^{13}/fx min.	819.2 μs min.
1	1	1	0	0	2^{14}/fx min.	1.64 ms min.
1	1	1	1	0	2^{15}/fx min.	3.27 ms min.
1	1	1	1	1	2^{16}/fx min.	6.55 ms min.

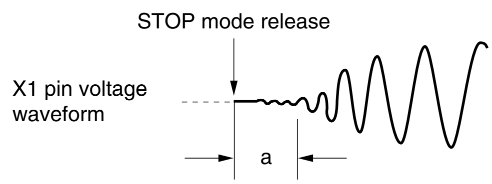
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST11 and remain 1.

2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock, set the oscillation stabilization time as follows.

- **Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS**

Note, therefore, that only the status up to the oscillation stabilization time set by OSTS is set to OSTC after STOP mode is released.

3. The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts (“a” below).



Remark fx: X1 clock oscillation frequency

(2) Oscillation stabilization time select register (OSTS)

This register is used to select the X1 clock oscillation stabilization wait time when the STOP mode is released.

When the X1 clock is selected as the CPU clock, the operation waits for the time set using OSTS after the STOP mode is released.

When the internal high-speed oscillation clock is selected as the CPU clock, confirm with OSTC that the desired oscillation stabilization time has elapsed after the STOP mode is released. The oscillation stabilization time can be checked up to the time set using OSTC.

OSTS can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets OSTS to 05H.

Figure 23-2. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFA4H After reset: 05H R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection	
				fx = 10 MHz
0	0	1	2^{11}/fx	204.8 μs
0	1	0	2^{13}/fx	819.2 μs
0	1	1	2^{14}/fx	1.64 ms
1	0	0	2^{15}/fx	3.27 ms
1	0	1	2^{16}/fx	6.55 ms
Other than above			Setting prohibited	

Cautions 1. To set the STOP mode when the X1 clock is used as the CPU clock, set OSTS before executing the STOP instruction.

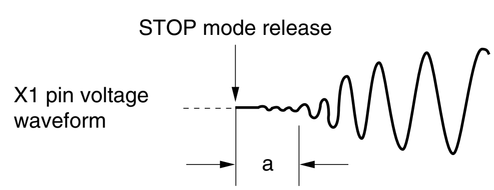
2. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

3. The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock, set the oscillation stabilization time as follows.

- **Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS**

Note, therefore, that only the status up to the oscillation stabilization time set by OSTS is set to OSTC after STOP mode is released.

4. The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts (“a” below).



Remark fx: X1 clock oscillation frequency

23.2 Standby Function Operation

23.2.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction. HALT mode can be set regardless of whether the CPU clock before the setting was the high-speed system clock, internal high-speed oscillation clock, or subsystem clock.

The operating statuses in the HALT mode are shown below.

Table 23-1. Operating Statuses in HALT Mode (1/2)

HALT Mode Setting / Item			When HALT Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU Is Operating on Internal High-Speed Oscillation Clock (f_{RH})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EXCLK})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{RH}	Operation continues (cannot be stopped)	Status before HALT mode was set is retained	
		f_X	Status before HALT mode was set is retained	Operation continues (cannot be stopped)	Status before HALT mode was set is retained
		f_{EXCLK}	Operates or stops by external clock input		Operation continues (cannot be stopped)
	Subsystem clock	f_{XT}	Status before HALT mode was set is retained		
	f_{RL}		Status before HALT mode was set is retained		
CPU			Operation stopped		
Flash memory			Operation stopped		
RAM			Status before HALT mode was set is retained		
Port (latch)			Status before HALT mode was set is retained		
16-bit timer/event counter 00			Operable		
8-bit timer/event counter	50				
	51				
	52				
8-bit timer	H0				
	H1				
	H2				
Real-time counter					
Watchdog timer			Operable. Clock supply to watchdog timer stops when "internal low-speed oscillator can be stopped by software" is set by option byte.		
Clock output			Operable		
Buzzer output					
10-bit successive approximation type A/D converter					
16-bit ΔΣ type A/D converter					
Serial interface	UART0				
	UART6				
	CSI10				
	CSIA0				
LCD controller/driver					
Manchester code generator					
Remote controller receiver					
Power-on-clear function					
Low-voltage detection function					
External interrupt					

Remark f_{RH}: Internal high-speed oscillation clock
f_X: X1 clock
f_{EXCLK}: External main system clock
f_{XT}: XT1 clock
f_{RL}: Internal low-speed oscillation clock

Table 23-1. Operating Statuses in HALT Mode (2/2)

Item \ HALT Mode Setting			When HALT Instruction Is Executed While CPU Is Operating on Subsystem Clock
			When CPU Is Operating on XT1 Clock (f_{XT})
System clock			Clock supply to the CPU is stopped
	Main system clock	f_{RH}	Status before HALT mode was set is retained
		f_X	
		f_{EXCLK}	Operates or stops by external clock input
	Subsystem clock	f_{XT}	Operation continues (cannot be stopped)
	f_{RL}		Status before HALT mode was set is retained
CPU			Operation stopped
Flash memory			Operation stopped
RAM			Status before HALT mode was set is retained
Port (latch)			Status before HALT mode was set is retained
16-bit timer/event counter 00[Note]			Operable
8-bit timer/event counter	50[Note]		
	51[Note]		
	52[Note]		
8-bit timer	H0		
	H1		
	H2		
Real-time counter			
Watchdog timer			Operable. Clock supply to watchdog timer stops when “internal low-speed oscillator can be stopped by software” is set by option byte.
Clock output			Operable
Buzzer output			Operable. However, operation disabled when peripheral hardware clock (f_{PRS}) is stopped.
10-bit successive approximation type A/D converter			
16-bit ΔΣ type A/D converter			Operable
Serial interface	UART0		
	UART6		
	CSI10[Note]		
	CSIA0[Note]		
LCD controller/driver			
Manchester code generator			
Remote controller receiver			
Power-on-clear function			
Low-voltage detection function			
External interrupt			

Note When the CPU is operating on the subsystem clock and the internal high-speed oscillation clock has been stopped, do not start operation of these functions on the external clock input from peripheral hardware pins.

Remark f_{RH}: Internal high-speed oscillation clock
f_X: X1 clock
f_{EXCLK}: External main system clock
f_{XT}: XT1 clock
f_{RL}: Internal low-speed oscillation clock

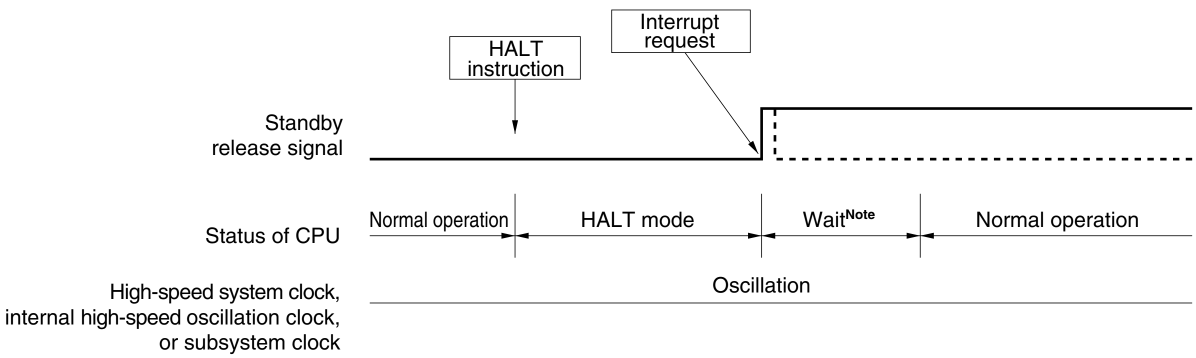
(2) HALT mode release

The HALT mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is released. If interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 23-3. HALT Mode Release by Interrupt Request Generation



Note The wait time is as follows:

- When vectored interrupt servicing is carried out: 8 or 9 clocks
- When vectored interrupt servicing is not carried out: 2 or 3 clocks

Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

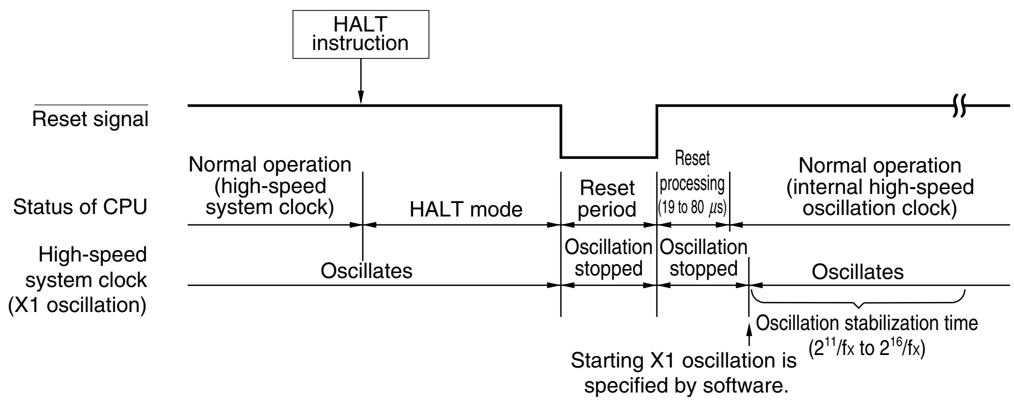
(b) Release by reset signal generation

When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

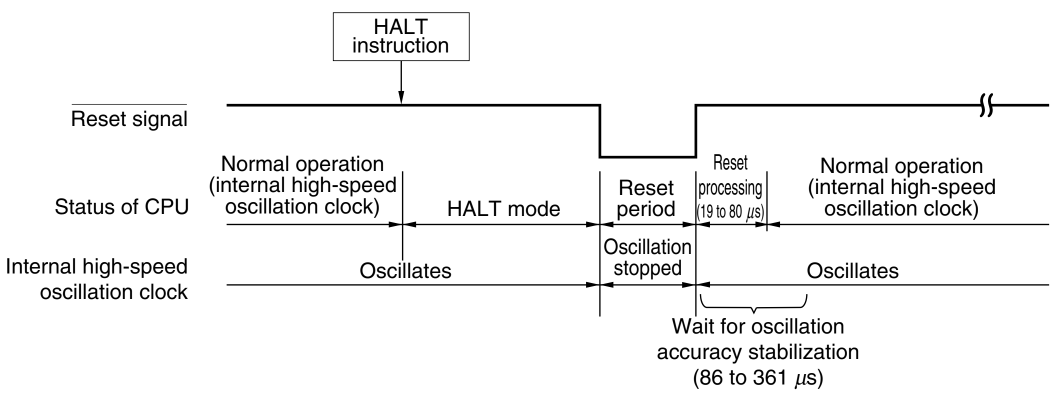
<R>

Figure 23-4. HALT Mode Release by Reset

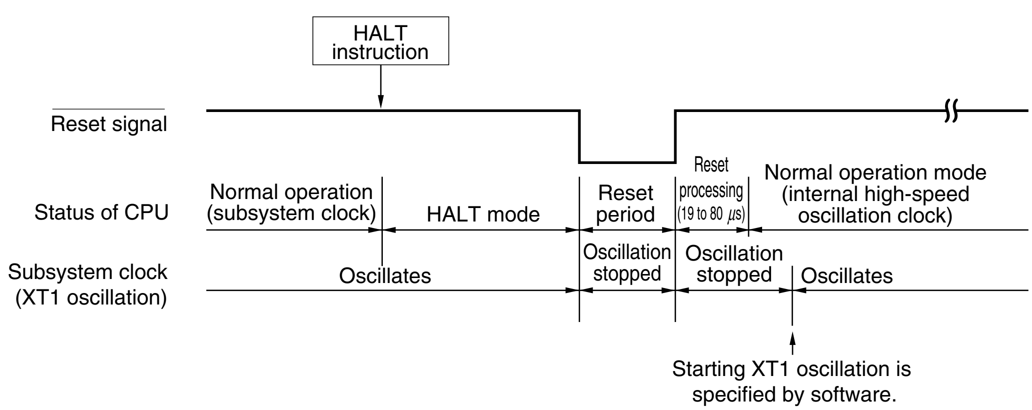
(1) When high-speed system clock is used as CPU clock



(2) When internal high-speed oscillation clock is used as CPU clock



(3) When subsystem clock is used as CPU clock



Remark fx: X1 clock oscillation frequency

Table 23-2. Operation in Response to Interrupt Request in HALT Mode

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt servicing execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt servicing execution
	1	×	×	×	HALT mode held
Reset	–	–	×	×	Reset processing

×: don't care

23.2.2 STOP mode

(1) STOP mode setting and operating statuses

The STOP mode is set by executing the STOP instruction, and it can be set only when the CPU clock before the setting was the main system clock.

Caution Because the interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction and the system returns to the operating mode as soon as the wait time set using the oscillation stabilization time select register (OSTS) has elapsed.

The operating statuses in the STOP mode are shown below.

Table 23-3. Operating Statuses in STOP Mode

STOP Mode Setting / Item			When STOP Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU Is Operating on Internal High-Speed Oscillation Clock (f_{RH})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EXCLK})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{RH}	Stopped		
		f_X			
		f_{EXCLK}	Input invalid		
	Subsystem clock	f_{XT}	Status before STOP mode was set is retained		
	f_{RL}		Status before STOP mode was set is retained		
CPU			Operation stopped		
Flash memory			Operation stopped		
RAM			Status before STOP mode was set is retained		
Port (latch)			Status before STOP mode was set is retained		
16-bit timer/event counter 00 Note 1			Operable only when TM52 output or TI000 is selected as the count clock		
8-bit timer/event counter	50 Note 1		Operable only when TI50 is selected as the count clock		
	51 Note 1		Operable only when TI51 is selected as the count clock		
	52 Note 1		Operable only when TI52 is selected as the count clock		
8-bit timer	H0		Operable only when TM50 output is selected as the count clock during 8-bit timer/event counter 50 operation		
	H1		Operable only when f_{RL}, $f_{RL}/2^7$, $f_{RL}/2^9$ is selected as the count clock		
	H2		Operation stopped		
Real-time counter			Operable only when subsystem clock is selected as the count clock		
Watchdog timer			Operable. Clock supply to watchdog timer stops when "internal low-speed oscillator can be stopped by software" is set by option byte.		
Clock output			Operable only when subsystem clock is selected as the count clock		
Buzzer output			Operation stopped		
10-bit successive approximation type A/D converter					
<R> 16-bit ΔΣ type A/D converter Note 2			Operable		
Serial interface	UART0		Operable only when TM50 output is selected as the serial clock during 8-bit timer/event counter 50 operation		
	UART6				
	CSI10 Note 1		Operable only when external clock is selected as the serial clock		
	CSIA0 Note 1		Operation stopped		
LCD controller/driver			Operable only when subsystem clock is selected as the count clock		
Manchester code generator			Operation stopped		
Remote controller receiver			Operable only when subsystem clock is selected as the count clock		
Power-on-clear function			Operable		
Low-voltage detection function					
External interrupt					

Notes 1. Do not start operation of these functions on the external clock input from peripheral hardware pins in the stop mode.

<R> **2.** Be sure to turn off the power (ADDPON = 0) of the 16-bit ΔΣ type A/D converter, when executing a STOP instruction upon selecting $f_{PRS}/4$, $f_{PRS}/8$, or $f_{PRS}/16$ for the sampling clock.

Remark f_{RH}: Internal high-speed oscillation clock
f_X: X1 clock
f_{EXCLK}: External main system clock
f_{XT}: XT1 clock
f_{RL}: Internal low-speed oscillation clock

Cautions 1. To use the peripheral hardware that stops operation in the STOP mode, and the peripheral hardware for which the clock that stops oscillating in the STOP mode after the STOP mode is released, restart the peripheral hardware.

2. Even if “internal low-speed oscillator can be stopped by software” is selected by the option byte, the internal low-speed oscillation clock continues in the STOP mode in the status before the STOP mode is set. To stop the internal low-speed oscillator’s oscillation in the STOP mode, stop it by software and then execute the STOP instruction.

<R> **3. To shorten oscillation stabilization time after the STOP mode is released when the CPU operates with the high-speed system clock (X1 oscillation), switch the CPU clock to the internal high-speed oscillation clock before the execution of the STOP instruction using the following procedure.**

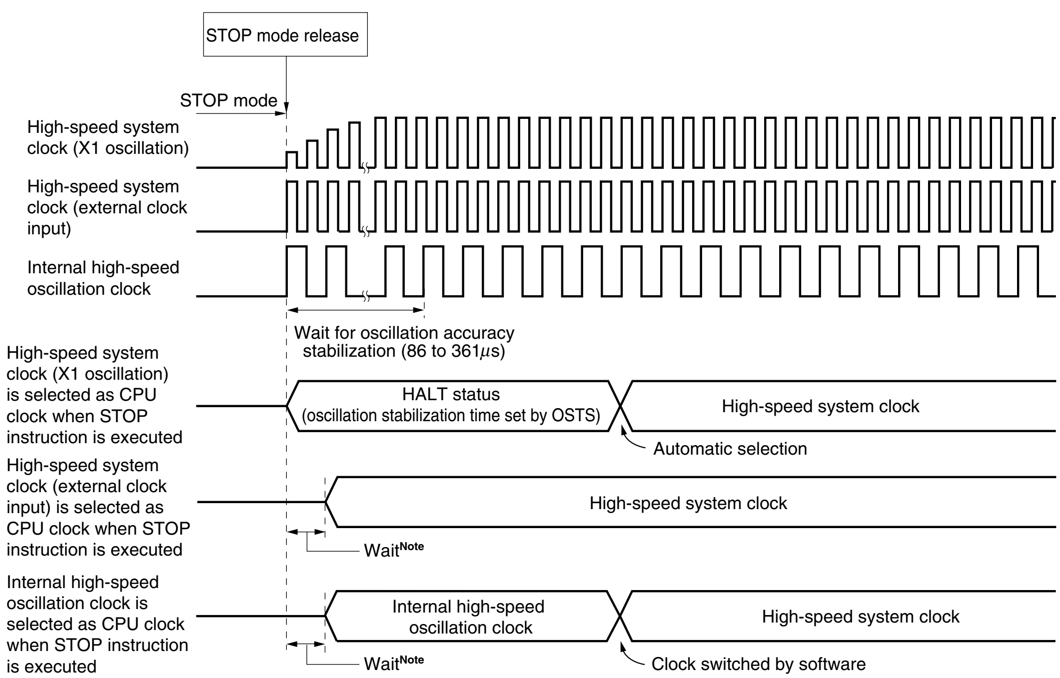
<1> Set RSTOP to 0 (starting oscillation of the internal high-speed oscillator) → <2> Set MCM0 to 0 (switching the CPU from X1 oscillation to internal high-speed oscillation) → <3> Check that MCS is 0 (checking the CPU clock) → <4> Check that RSTS is 1 (checking internal high-speed oscillation operation) → <5> Execute the STOP instruction

Before changing the CPU clock from the internal high-speed oscillation clock to the high-speed system clock (X1 oscillation) after the STOP mode is released, check the oscillation stabilization time with the oscillation stabilization time counter status register (OSTC).

<R> **4. Execute the STOP instruction after having confirmed that the internal high-speed oscillator is operating stably (RSTS = 1).**

(2) STOP mode release

Figure 23-5. Operation Timing When STOP Mode Is Released (When Unmasked Interrupt Request Is Generated)



Note The wait time is as follows:

- When vectored interrupt servicing is carried out: 8 or 9 clocks
- When vectored interrupt servicing is not carried out: 2 or 3 clocks

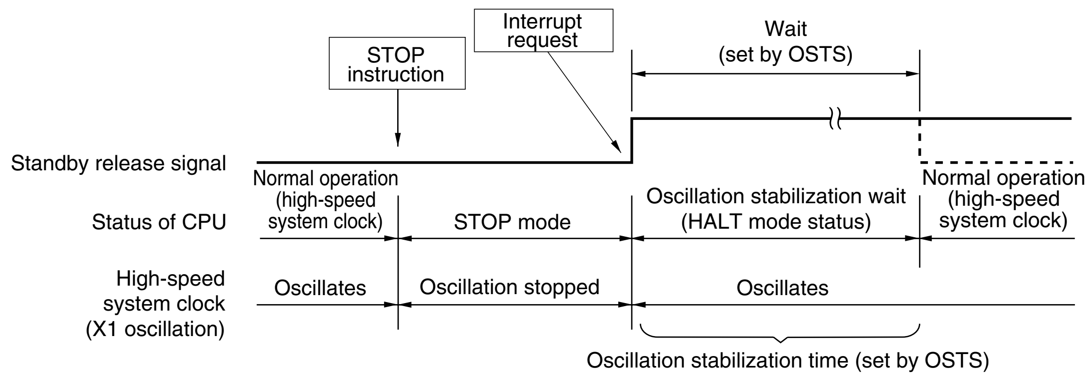
The STOP mode can be released by the following two sources.

(a) Release by unmasked interrupt request

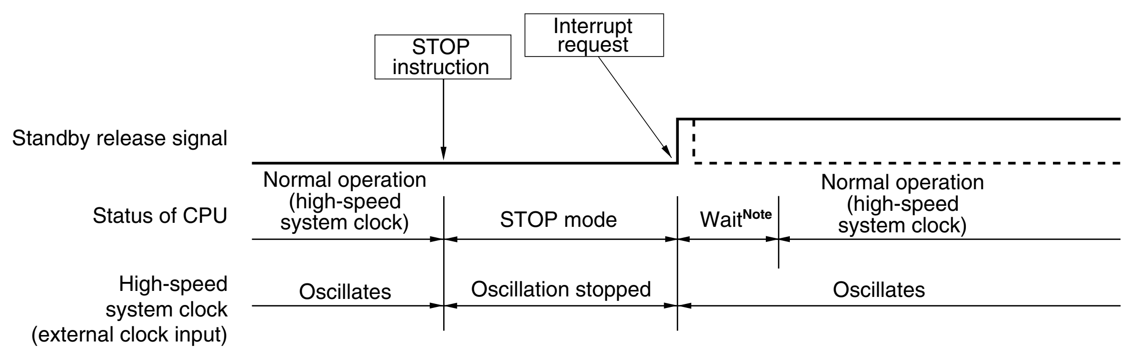
When an unmasked interrupt request is generated, the STOP mode is released. After the oscillation stabilization time has elapsed, if interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 23-6. STOP Mode Release by Interrupt Request Generation (1/2)

(1) When high-speed system clock (X1 oscillation) is used as CPU clock



(2) When high-speed system clock (external clock input) is used as CPU clock



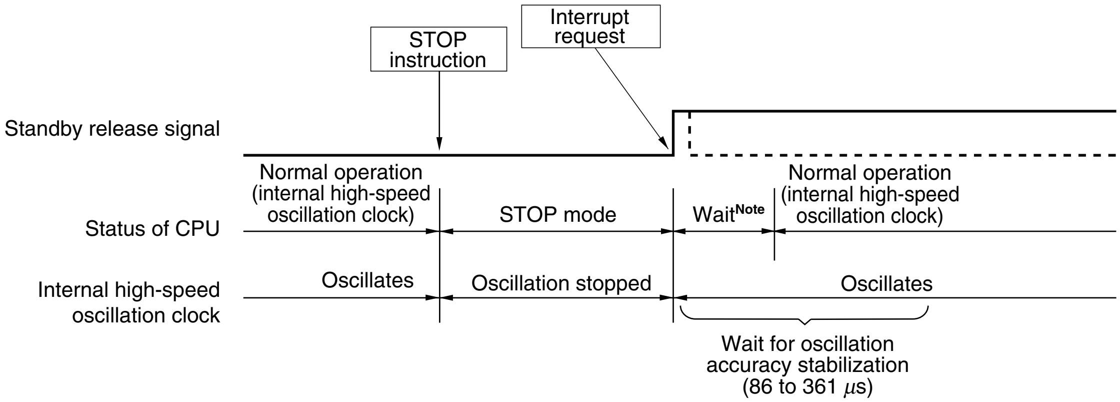
Note The wait time is as follows:

- When vectored interrupt servicing is carried out: 8 or 9 clocks
- When vectored interrupt servicing is not carried out: 2 or 3 clocks

Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

Figure 23-6. STOP Mode Release by Interrupt Request Generation (2/2)

(3) When internal high-speed oscillation clock is used as CPU clock



Note The wait time is as follows:

- When vectored interrupt servicing is carried out: 8 or 9 clocks
- When vectored interrupt servicing is not carried out: 2 or 3 clocks

Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

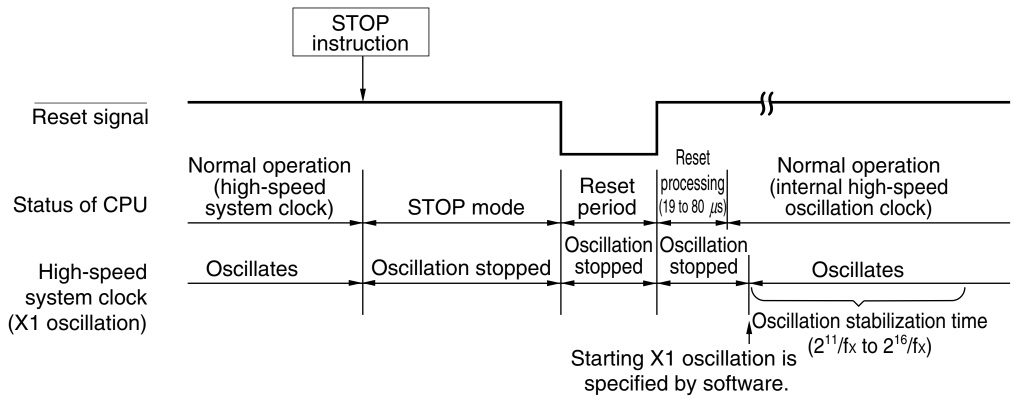
(b) Release by reset signal generation

When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

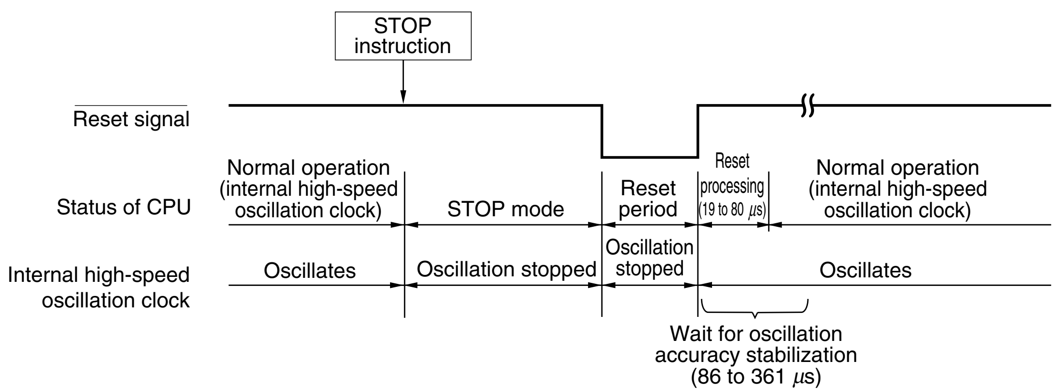
<R>

Figure 23-7. STOP Mode Release by Reset

(1) When high-speed system clock is used as CPU clock



(2) When internal high-speed oscillation clock is used as CPU clock



Remark fx: X1 clock oscillation frequency

Table 23-4. Operation in Response to Interrupt Request in STOP Mode

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt servicing execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt servicing execution
	1	×	×	×	STOP mode held
Reset	–	–	×	×	Reset processing

×: don't care

CHAPTER 24 RESET FUNCTION

The following four operations are available to generate a reset signal.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer program loop detection
(3) Internal reset by comparison of supply voltage and detection voltage of power-on-clear (POC) circuit
(4) Internal reset by comparison of supply voltage and detection voltage of low-power-supply detector (LVI)

External and internal resets have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H when the reset signal is generated.

A reset is applied when a low level is input to the $\overline{\text{RESET}}$ pin, the watchdog timer overflows, or by POC and LVI circuit voltage detection, and each item of hardware is set to the status shown in Tables 24-1 and 24-2. Each pin is high impedance during reset signal generation or during the oscillation stabilization time just after a reset release.

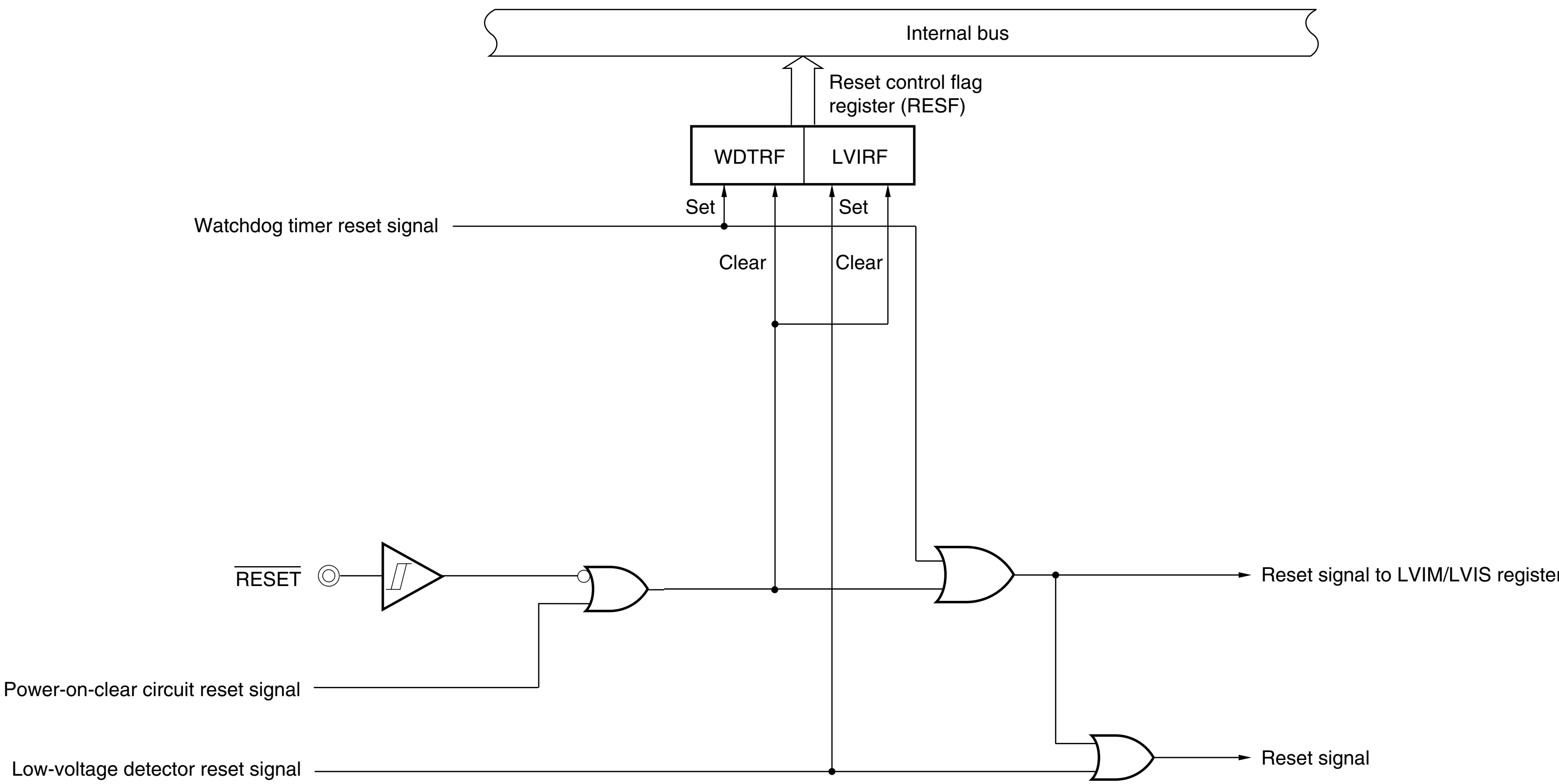
When a low level is input to the $\overline{\text{RESET}}$ pin, the device is reset. It is released from the reset status when a high level is input to the $\overline{\text{RESET}}$ pin and program execution is started with the internal high-speed oscillation clock after reset processing. A reset by the watchdog timer is automatically released, and program execution starts using the internal high-speed oscillation clock (see **Figures 24-2** to **24-4**) after reset processing. Reset by POC and LVI circuit power supply detection is automatically released when $V_{DD} \geq V_{POC}$ or $V_{DD} \geq V_{LVI}$ after the reset, and program execution starts using the internal high-speed oscillation clock (see **CHAPTER 25 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 26 LOW-VOLTAGE DETECTOR**) after reset processing.

Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.

2. During reset input, the X1 clock, XT1 clock, internal high-speed oscillation clock, and internal low-speed oscillation clock stop oscillating. External main system clock input becomes invalid.

3. When the STOP mode is released by a reset, the STOP mode contents are held during reset input. However, the port pins become high-impedance.

Figure 24-1. Block Diagram of Reset Function



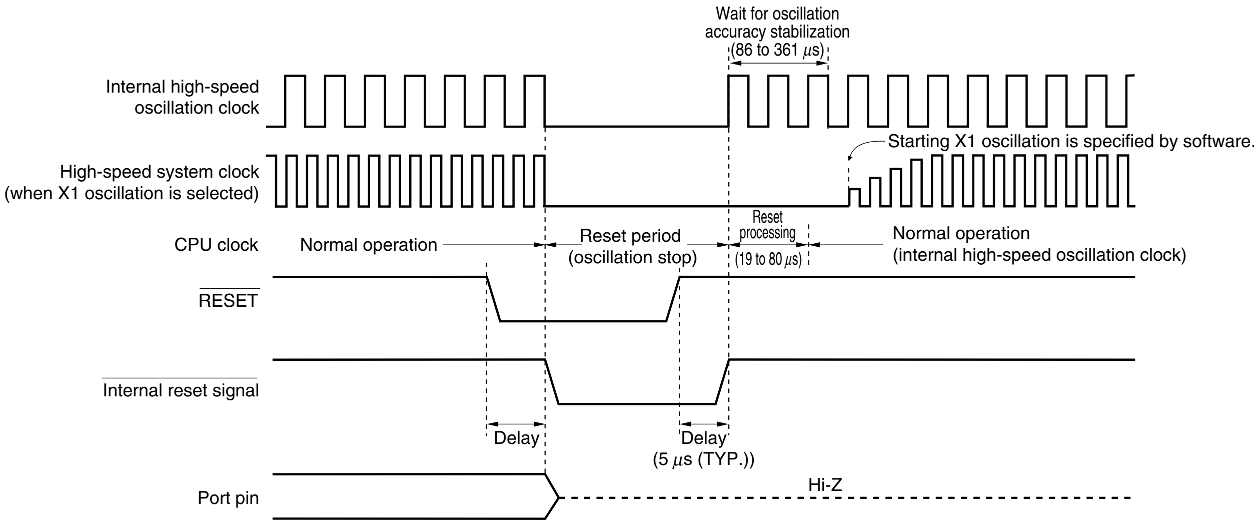
Caution An LVI circuit internal reset does not reset the LVI circuit.

Remarks 1. LVIM: Low-voltage detection register

2. LVIS: Low-voltage detection level selection register

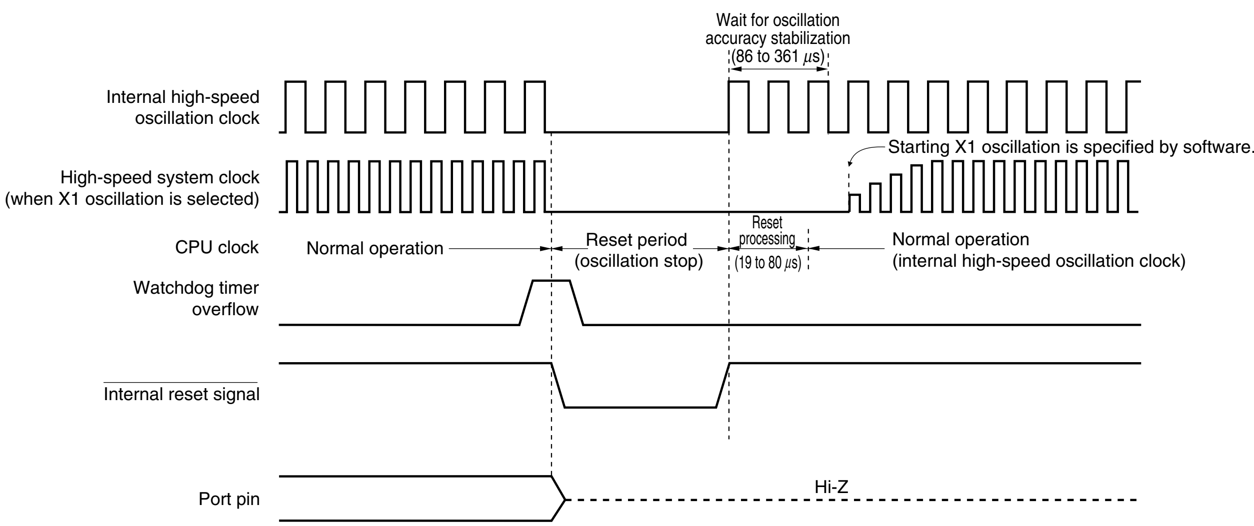
<R>

Figure 24-2. Timing of Reset by RESET Input



<R>

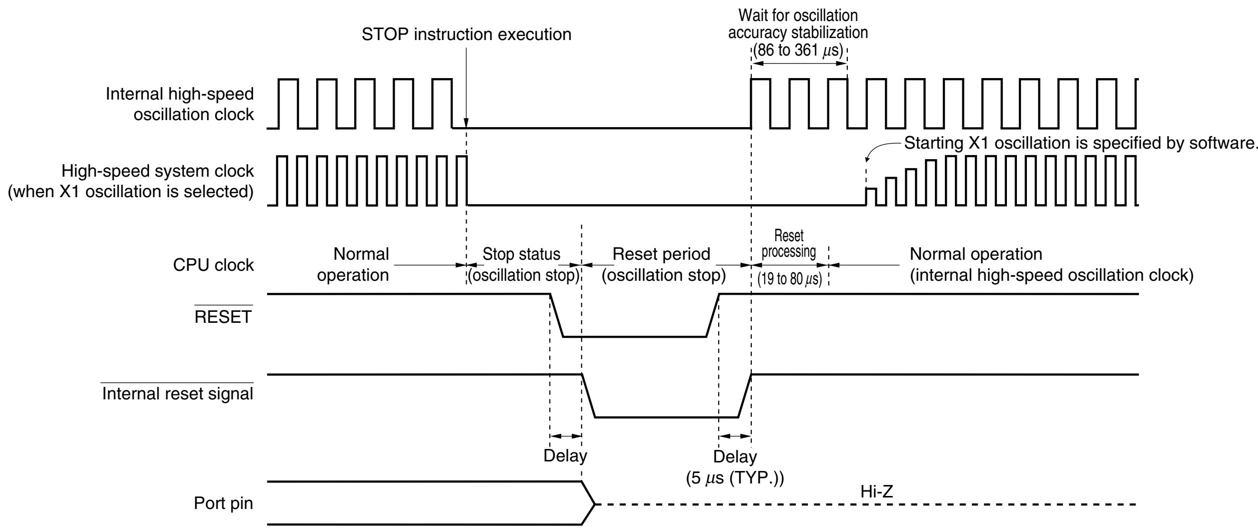
Figure 24-3. Timing of Reset Due to Watchdog Timer Overflow



Caution A watchdog timer internal reset resets the watchdog timer.

<R>

Figure 24-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input



Remark For the reset timing of the power-on-clear circuit and low-voltage detector, see **CHAPTER 25 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 26 LOW-VOLTAGE DETECTOR**.

Table 24-1. Operation Statuses During Reset Period

Item			During Reset Period
System clock			Clock supply to the CPU is stopped.
	Main system clock	f_{RH}	Operation stopped
		f_X	Operation stopped (pin is I/O port mode)
		f_{EXCLK}	Clock input invalid (pin is I/O port mode)
	Subsystem clock	f_{XT}	Operation stopped (pin is I/O port mode)
	f_{RL}		Operation stopped
CPU			
Flash memory			
RAM			
Port (latch)			
16-bit timer/event counter	00		
8-bit timer/event counter	50		
	51		
	52		
8-bit timer	H0		
	H1		
	H2		
Real-time counter			
Watchdog timer			
Clock output			
Buzzer output			
10-bit successive approximation type A/D converter			
16-bit ΔΣ type A/D converter			
Serial interface	UART0		
	UART6		
	CSI10		
	CSIA0		
LCD controller/driver			
Manchester code generator			
Remote controller receiver			
Power-on-clear function			Operable
Low-voltage detection function			Operation stopped
External interrupt			

Remark f_{RH}: Internal high-speed oscillation clock
f_X: X1 oscillation clock
f_{EXCLK}: External main system clock
f_{XT}: XT1 oscillation clock
f_{RL}: Internal low-speed oscillation clock

Table 24-2. Hardware Statuses After Reset Acknowledgment (1/4)

Hardware		After Reset Acknowledgment[Note 1]
Program counter (PC)		The contents of the reset vector table (0000H, 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined[Note 2]
	General-purpose registers	Undefined[Note 2]
Port registers (P1 to P4, P8 to P15) (output latches)		00H
Port mode registers (PM1 to PM4, PM8 to PM15)		FFH
Pull-up resistor option registers (PU1, PU3, PU4, PU8 to PU15)		00H
Port function register 1 (PF1)		00H
Port function register 2 (PF2)		00H
Port function register ALL (PFALL)		00H
Internal expansion RAM size switching register (IXS)		0CH[Note 3]
Internal memory size switching register (IMS)		CFH[Note 3]
Clock operation mode select register (OSCCTL)		00H
Processor clock control register (PCC)		01H
Internal oscillation mode register (RCM)		80H
Main OSC control register (MOC)		80H
Main clock mode register (MCM)		00H
Oscillation stabilization time counter status register (OSTC)		00H
Oscillation stabilization time select register (OSTS)		05H
Internal high-speed oscillation trimming register (HIOTRM)		10H
16-bit timer/event counters 00	Timer counters 00 (TM00)	0000H
	Capture/compare registers 000, 010 (CR000, CR010)	0000H
	Mode control registers 00 (TMC00)	00H
	Prescaler mode registers 00 (PRM00)	00H
	Capture/compare control registers 00 (CRC00)	00H
	Timer output control registers 00 (TOC00)	00H
8-bit timer/event counters 50, 51, 52	Timer counters 50, 51, 52 (TM50, TM51, TM52)	00H
	Compare registers 50, 51, 52 (CR50, CR51, CR52)	00H
	Timer clock selection registers 50, 51, 52 (TCL50, TCL51, TCL52)	00H
	Mode control registers 50, 51, 52 (TMC50, TMC51, TMC52)	00H

Notes 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. When a reset is executed in the standby mode, the pre-reset status is held even after reset.

3. The initial values of the internal memory size switching register (IMS) and internal expansion RAM size switching register (IXS) after a reset release are constant (IMS = CFH, IXS = 0CH) in all the 78K0/LF3 products, regardless of the internal memory capacity. Therefore, after a reset is released, be sure to set the following values for each product.

Flash Memory Version (78K0/LF3)	IMS	IXS
μPD78F0471, 78F0481, 78F0491	04H	0CH
μPD78F0472, 78F0482, 78F0492	C6H	
μPD78F0473, 78F0483, 78F0493	C8H	
μPD78F0474, 78F0484, 78F0494	CCH	0AH
μPD78F0475, 78F0485, 78F0495	CFH	

Table 24-2. Hardware Statuses After Reset Acknowledgment (2/4)

Hardware		Status After Reset Acknowledgment[Note 1]
8-bit timers H0, H1, H2	Compare registers 00, 10, 01, 11, 02, 12 (CMP00, CMP10, CMP01, CMP11, CMP02, CMP12)	00H
	Mode registers (TMHMD0, TMHMD1, TMHMD2)	00H
	Carrier control register 1 (TMCYC1)[Note 2]	00H
Real-time counter	Clock selection register (RTCCL)	00H
	Sub-count register (RSUBC)	0000H
	Second count register (SEC)	00H
	Minute count register (MIN)	00H
	Hour count register (HOUR)	12H
	Week count register (WEEK)	00H
	Day count register (DAY)	01H
	Month count register (MONTH)	01H
	Year count register (YEAR)	00H
	Watch error correction register (SUBCUD)	00H
	Alarm minute register (ALARMWM)	00H
	Alarm hour register (ALARMWH)	12H
	Alarm week register (ALARMWW)	00H
	Control register 0 (RTCC0)	00H
	Control register 1 (RTCC1)	00H
	Control register 2 (RTCC2)	00H
Clock output/buzzer output controller	Clock output selection register (CKS)	00H
Watchdog timer	Enable register (WDTE)	1AH/9AH[Note 3]
10-bit successive approximation type A/D converter	10-bit A/D conversion result register (ADCR)	0000H
	8-bit A/D conversion result register (ADCRH)	00H
	A/D converter mode register (ADM)	00H
	Analog input channel specification register (ADS)	00H
	A/D port configuration register 0 (ADPC0)	08H
16-bit ΔΣ type A/D converter	ΔΣ A/D converter control register 0 (ADDCTL0)	00H
	ΔΣ A/D converter control register 1 (ADDCTL1)	00H
	16-bit ΔΣ A/D conversion status register (ADDSTR)	00H
	16-bit ΔΣ A/D conversion result register (ADDCR)	0000H
	8-bit ΔΣ A/D conversion result register (ADDCRH)	00H
Serial interface UART0	Receive buffer register 0 (RXB0)	FFH
	Transmit shift register 0 (TXS0)	FFH
	Asynchronous serial interface operation mode register 0 (ASIM0)	01H
	Asynchronous serial interface reception error status register 0 (ASIS0)	00H
	Baud rate generator control register 0 (BRGC0)	1FH

Notes 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. 8-bit timer H1 only.

3. The reset value of WDTE is determined by the option byte setting.

Table 24-2. Hardware Statuses After Reset Acknowledgment (3/4)

Hardware		Status After Reset Acknowledgment[Note 1]
Serial interface UART6	Receive buffer register 6 (RXB6)	FFH
	Transmit buffer register 6 (TXB6)	FFH
	Asynchronous serial interface operation mode register 6 (ASIM6)	01H
	Asynchronous serial interface reception error status register 6 (ASIS6)	00H
	Asynchronous serial interface transmission status register 6 (ASIF6)	00H
	Clock selection register 6 (CKSR6)	00H
	Baud rate generator control register 6 (BRGC6)	FFH
	Asynchronous serial interface control register 6 (ASICL6)	16H
	Input switch control register (ISC)	00H
Serial interfaces CSI10	Transmit buffer registers 10 (SOTB10)	00H
	Serial I/O shift registers 10 (SIO10)	00H
	Serial operation mode registers 10 (CSIM10)	00H
	Serial clock selection registers 10 (CSIC10)	00H
Serial interface CSIA0	Serial operation mode specification register 0 (CSIMA0)	00H
	Serial status register 0 (CSIS0)	00H
	Serial trigger register 0 (CSIT0)	00H
	Divisor value selection register 0 (BRGCA0)	03H
	Automatic data transfer address point specification register 0 (ADTP0)	00H
	Automatic data transfer interval specification register 0 (ADTI0)	00H
	Serial I/O shift register 0 (SIOA0)	00H
	Automatic data transfer address count register 0 (ADTC0)	00H
LCD controller/driver	LCD mode register (LCDMD)	00H
	LCD display mode register (LCDM)	00H
	LCD clock control register 0 (LCDC0)	00H
Manchester code generator	Transmit buffer register (MC0TX)	FFH
	Transmit bit count specification register (MC0BIT)	07H
	Control register 0 (MC0CTL0)	10H
	Control register 1 (MC0CTL1)	00H
	Control register 2 (MC0CTL2)	1FH
	Status register (MC0STR)	00H
Key interrupt	Key return mode register (KRM)	00H
Reset function	Reset control flag register (RESF)	00H[Note 2]
Low-voltage detector	Low-voltage detection register (LVIM)	00H[Note 2]
	Low-voltage detection level selection register (LVIS)	00H[Note 2]

Notes 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. These values vary depending on the reset source.

Register \ Reset Source		$\overline{\text{RESET}}$ Input	Reset by POC	Reset by WDT	Reset by LVI
RESF	WDTRF bit	Cleared (0)	Cleared (0)	Set (1)	Held
	LVIRF bit			Held	Set (1)
LVIM		Cleared (00H)	Cleared (00H)	Cleared (00H)	Held
LVIS					

Table 24-2. Hardware Statuses After Reset Acknowledgment (4/4)

Hardware		Status After Reset Acknowledgment[Note]
Remote controller receiver	Remote controller receive shift register (RMSR)	00H
	Remote controller receive data register (RMDR)	00H
	Remote controller shift register receive counter register (RMSCR)	00H
	Remote controller receive GPHS compare register (RMGPHS)	00H
	Remote controller receive GPHL compare register (RMGPHL)	00H
	Remote controller receive DLS compare register (RMDLS)	00H
	Remote controller receive DLL compare register (RMDLL)	00H
	Remote controller receive DH0S compare register (RMDH0S)	00H
	Remote controller receive DH0L compare register (RMDH0L)	00H
	Remote controller receive DH1S compare register (RMDH1S)	00H
	Remote controller receive DH1L compare register (RMDH1L)	00H
	Remote controller receive end width select register (RMER)	00H
	Remote controller receive interrupt status register (INTS)	00H
	Remote controller receive interrupt status clear register (INTC)	00H
	Remote controller receive control register (RMCN)	00H
Interrupt	Request flag registers 0L, 0H, 1L, 1H (IF0L, IF0H, IF1L, IF1H)	00H
	Mask flag registers 0L, 0H, 1L, 1H (MK0L, MK0H, MK1L, MK1H)	FFH
	Priority specification flag registers 0L, 0H, 1L, 1H (PR0L, PR0H, PR1L, PR1H)	FFH
	External interrupt rising edge enable register (EGP)	00H
	External interrupt falling edge enable register (EGN)	00H

Note During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

24.1 Register for Confirming Reset Source

Many internal reset generation sources exist in the 78K0/LF3. The reset control flag register (RESF) is used to store which source has generated the reset request.

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

$\overline{\text{RESET}}$ input, reset by power-on-clear (POC) circuit, and reading RESF set RESF to 00H.

Figure 24-5. Format of Reset Control Flag Register (RESF)

Address: FFACH After reset: 00H[Note] R

Symbol	7	6	5	4	3	2	1	0
RESF	0	0	0	WDTRF	0	0	0	LVIRF

WDTRF	Internal reset request by watchdog timer (WDT)
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

LVIRF	Internal reset request by low-voltage detector (LVI)
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

Note The value after reset varies depending on the reset source.

Caution Do not read data by a 1-bit memory manipulation instruction.

The status of RESF when a reset request is generated is shown in Table 24-3.

Table 24-3. RESF Status When Reset Request Is Generated

Flag \ Reset Source	$\overline{\text{RESET}}$ Input	Reset by POC	Reset by WDT	Reset by LVI
WDTRF	Cleared (0)	Cleared (0)	Set (1)	Held
LVIRF			Held	Set (1)

CHAPTER 25 POWER-ON-CLEAR CIRCUIT

25.1 Functions of Power-on-Clear Circuit

The power-on-clear circuit (POC) has the following functions.

- Generates internal reset signal at power on.
  In the 1.59 V POC mode (option byte: POCMODE = 0), the reset signal is released when the supply voltage (V_{DD}) exceeds 1.59 V ±0.15 V.
  In the 2.7 V/1.59 V POC mode (option byte: POCMODE = 1), the reset signal is released when the supply voltage (V_{DD}) exceeds 2.7 V ±0.2 V.

- Compares supply voltage (V_{DD}) and detection voltage (V_{POC} = 1.59 V ±0.15 V), generates internal reset signal when $V_{DD} < V_{POC}$.

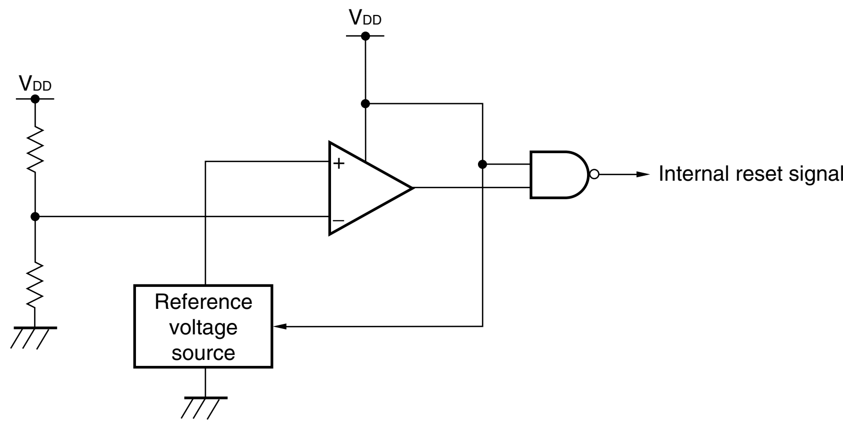
Caution If an internal reset signal is generated in the POC circuit, the reset control flag register (RESF) is cleared to 00H.

Remark 78K0/LF3 incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset source is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT) or low-voltage-detector (LVI). RESF is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by WDT or LVI.
For details of RESF, see **CHAPTER 24 RESET FUNCTION**.

25.2 Configuration of Power-on-Clear Circuit

The block diagram of the power-on-clear circuit is shown in Figure 25-1.

Figure 25-1. Block Diagram of Power-on-Clear Circuit



25.3 Operation of Power-on-Clear Circuit

(1) In 1.59 V POC mode (option byte: POCMODE = 0)

- An internal reset signal is generated on power application. When the supply voltage (V_{DD}) exceeds the detection voltage (V_{POC} = 1.59 V ±0.15 V), the reset status is released.
- The supply voltage (V_{DD}) and detection voltage (V_{POC} = 1.59 V ±0.15 V) are compared. When $V_{DD} < V_{POC}$, the internal reset signal is generated. It is released when $V_{DD} \geq V_{POC}$.

(2) In 2.7 V/1.59 V POC mode (option byte: POCMODE = 1)

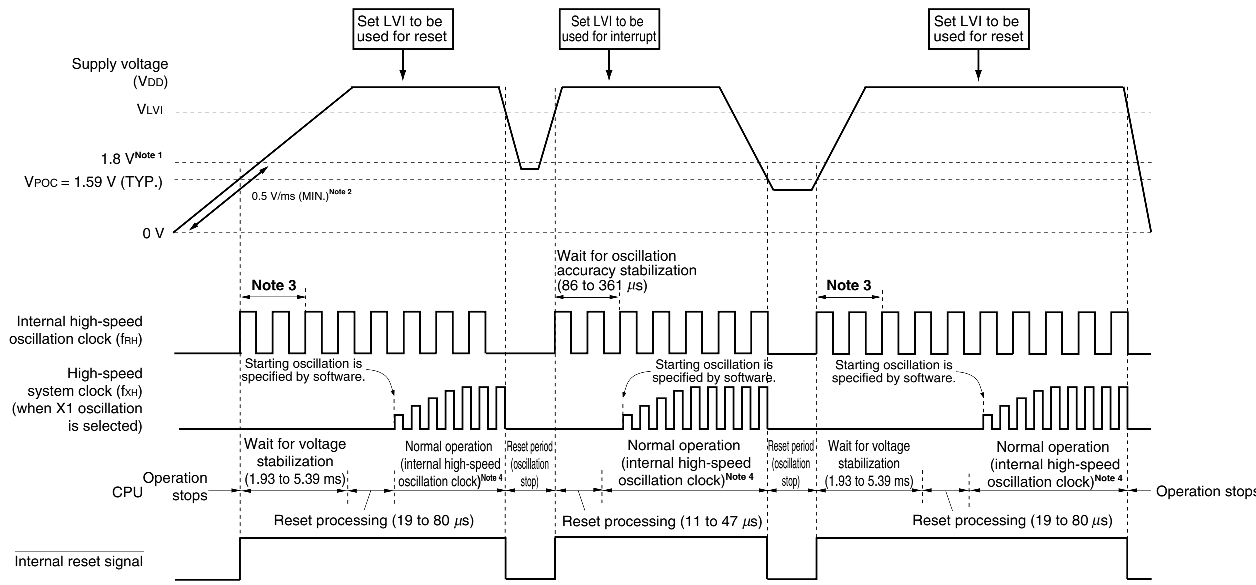
- An internal reset signal is generated on power application. When the supply voltage (V_{DD}) exceeds the detection voltage (V_{DDPOC} = 2.7 V ±0.2 V), the reset status is released.
- The supply voltage (V_{DD}) and detection voltage (V_{POC} = 1.59 V ±0.15 V) are compared. When $V_{DD} < V_{POC}$, the internal reset signal is generated. It is released when $V_{DD} \geq V_{DDPOC}$.

The timing of generation of the internal reset signal by the power-on-clear circuit and low-voltage detector is shown below.

<R>

Figure 25-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (1/2)

(1) In 1.59 V POC mode (option byte: POCMODE = 0)



Notes 1. The operation guaranteed range is 1.8 V ≤ V_{DD} ≤ 5.5 V. To make the state at lower than 1.8 V reset state when the supply voltage falls, use the reset function of the low-voltage detector, or input the low level to the $\overline{\text{RESET}}$ pin.

2. If the voltage rises to 1.8 V at a rate slower than 0.5 V/ms (MIN.) on power application, input a low level to the $\overline{\text{RESET}}$ pin after power application and before the voltage reaches 1.8 V, or set the 2.7 V/1.59 V POC mode by using an option byte (POCMODE = 1).

3. The internal voltage stabilization time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

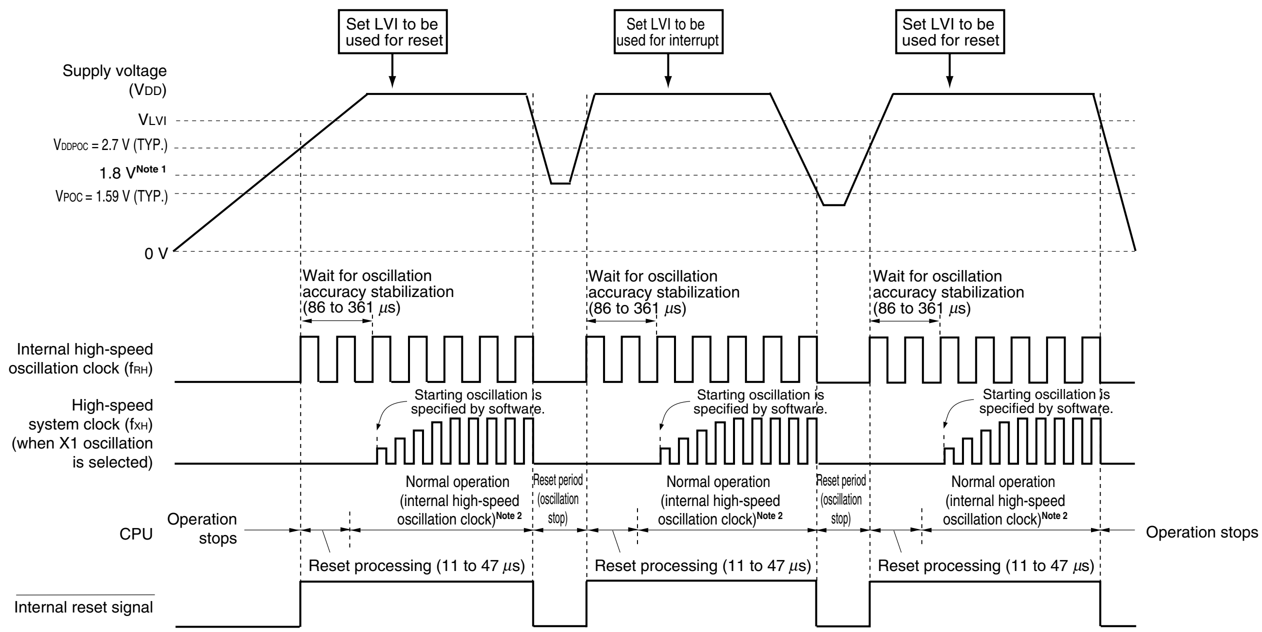
4. The internal high-speed oscillation clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the OSTC register to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

Caution Set the low-voltage detector by software after the reset status is released (see CHAPTER 26 LOW-VOLTAGE DETECTOR).

Remark V_{LVI}: LVI detection voltage
V_{POC}: POC detection voltage

Figure 25-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (2/2)

(2) In 2.7 V/1.59 V POC mode (option byte: POCMODE = 1)



Notes 1. The operation guaranteed range is 1.8 V ≤ V_{DD} ≤ 5.5 V. To make the state at lower than 1.8 V reset state when the supply voltage falls, use the reset function of the low-voltage detector, or input the low level to the $\overline{\text{RESET}}$ pin.

2. The internal high-speed oscillation clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the OSTC register to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

Cautions 1. Set the low-voltage detector by software after the reset status is released (see CHAPTER 26 LOW-VOLTAGE DETECTOR).

<R> **2. A voltage oscillation stabilization time of 1.93 to 5.39 ms is required after the supply voltage reaches 1.59 V (TYP.). If the time the supply voltage rises from 1.59 V (TYP.) to 2.7 V (TYP.) is within 1.93 to 5.39 ms, a power supply stabilization wait time of 0 to 5.39 ms occurs automatically before reset processing and the reset processing time becomes 19 to 80 μs.**

Remark V_{LVI}: LVI detection voltage
V_{POC}: POC detection voltage

25.4 Cautions for Power-on-Clear Circuit

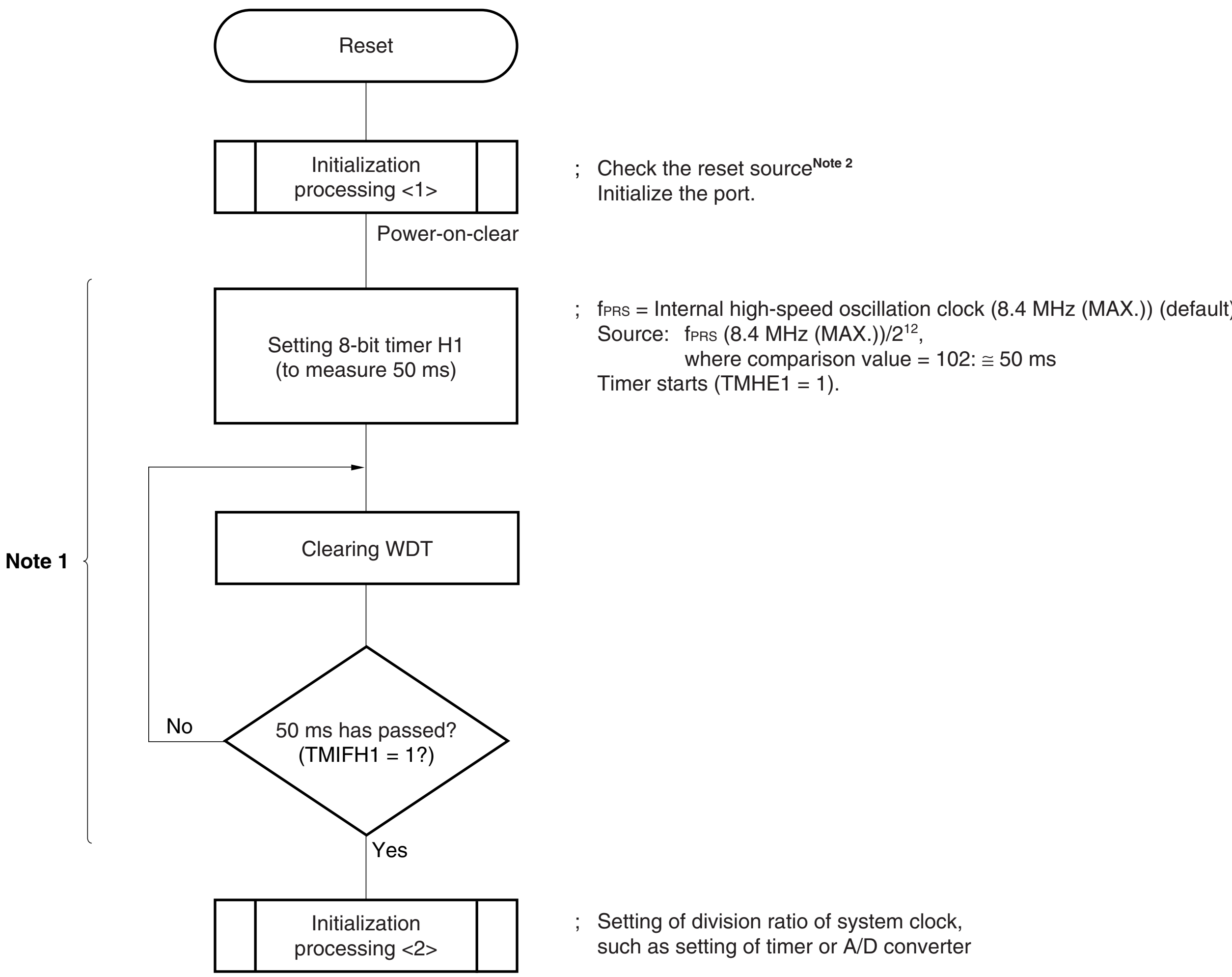
In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the POC detection voltage (V_{POC}), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking the following action.

<Action>

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

Figure 25-3. Example of Software Processing After Reset Release (1/2)

• If supply voltage fluctuation is 50 ms or less in vicinity of POC detection voltage

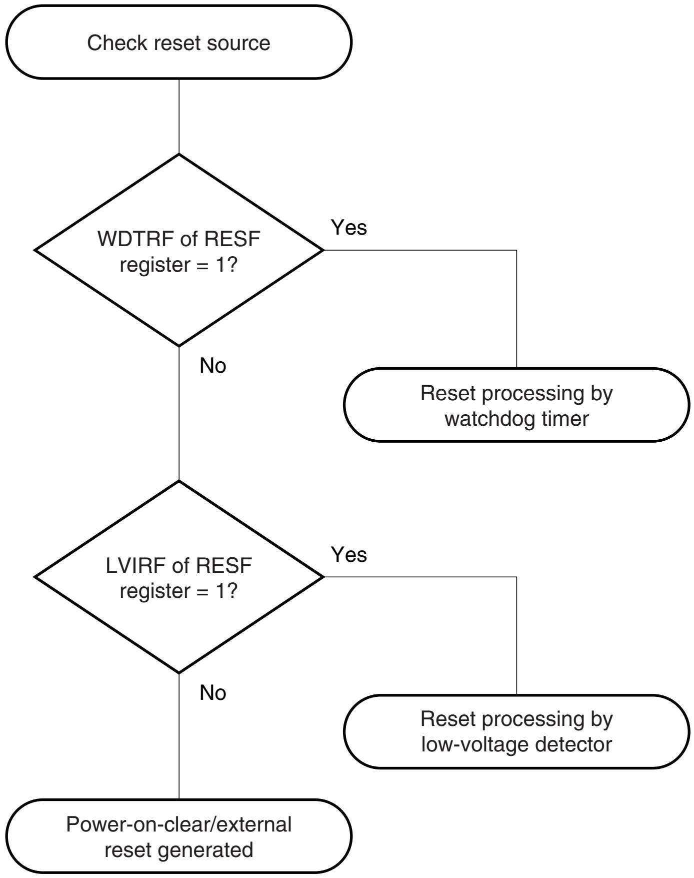


Notes 1. If reset is generated again during this period, initialization processing <2> is not started.

2. A flowchart is shown on the next page.

Figure 25-3. Example of Software Processing After Reset Release (2/2)

- Checking reset source



CHAPTER 26 LOW-VOLTAGE DETECTOR

26.1 Functions of Low-Voltage Detector

The low-voltage detector (LVI) has the following functions.

- The LVI circuit compares the supply voltage (V_{DD}) with the detection voltage (V_{LVI}) or the input voltage from an external input pin (EXLVI) with the detection voltage (V_{EXLVI} = 1.21 V (TYP.): fixed), and generates an internal reset or internal interrupt signal.
- The supply voltage (V_{DD}) or input voltage from an external input pin (EXLVI) can be selected by software.
- Reset or interrupt function can be selected by software.
- Detection levels (16 levels) of supply voltage can be changed by software.
- Operable in STOP mode.

The reset and interrupt signals are generated as follows depending on selection by software.

Selection of Level Detection of Supply Voltage (V_{DD}) (LVISEL = 0)		Selection Level Detection of Input Voltage from External Input Pin (EXLVI) (LVISEL = 1)	
Selects reset (LVIMD = 1).	Selects interrupt (LVIMD = 0).	Selects reset (LVIMD = 1).	Selects interrupt (LVIMD = 0).
Generates an internal reset signal when $V_{DD} < V_{LVI}$ and releases the reset signal when $V_{DD} \geq V_{LVI}$.	Generates an internal interrupt signal when V_{DD} drops lower than V_{LVI} ($V_{DD} < V_{LVI}$) or when V_{DD} becomes V_{LVI} or higher ($V_{DD} \geq V_{LVI}$).	Generates an internal reset signal when EXLVI < V_{EXLVI} and releases the reset signal when EXLVI ≥ V_{EXLVI}.	Generates an internal interrupt signal when EXLVI drops lower than V_{EXLVI} (EXLVI < V_{EXLVI}) or when EXLVI becomes V_{EXLVI} or higher (EXLVI ≥ V_{EXLVI}).

Remark LVISEL: Bit 2 of low-voltage detection register (LVIM)
LVIMD: Bit 1 of LVIM

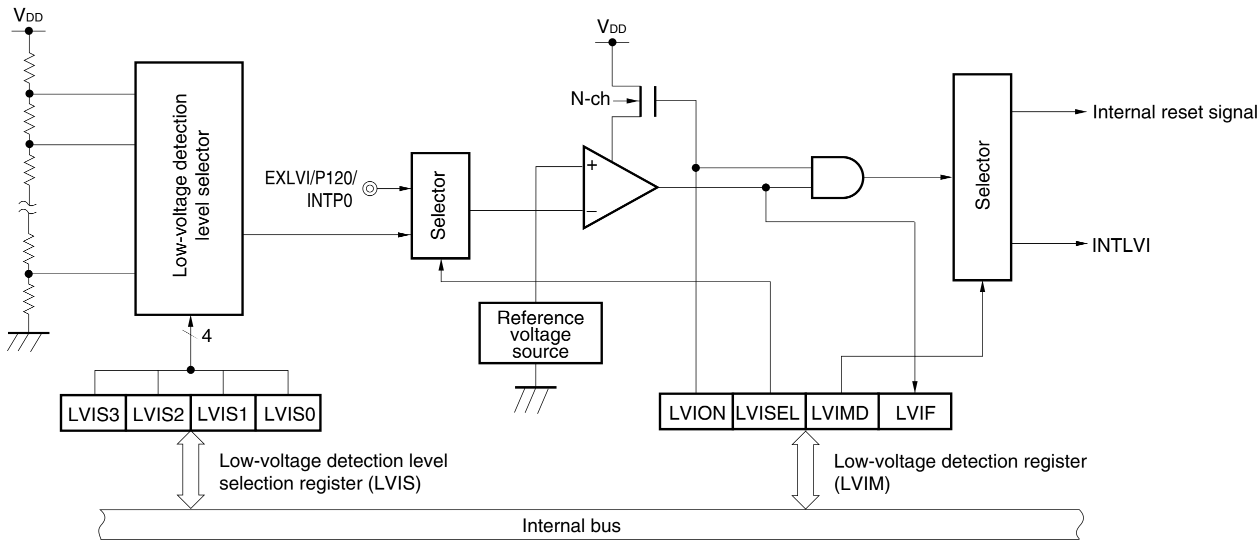
While the low-voltage detector is operating, whether the supply voltage or the input voltage from an external input pin is more than or less than the detection level can be checked by reading the low-voltage detection flag (LVIF: bit 0 of LVIM).

When the low-voltage detector is used to reset, bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of RESF, see **CHAPTER 24 RESET FUNCTION**.

26.2 Configuration of Low-Voltage Detector

The block diagram of the low-voltage detector is shown in Figure 26-1.

Figure 26-1. Block Diagram of Low-Voltage Detector



26.3 Registers Controlling Low-Voltage Detector

The low-voltage detector is controlled by the following registers.

- Low-voltage detection register (LVIM)
- Low-voltage detection level selection register (LVIS)
- Port mode register 12 (PM12)

(1) Low-voltage detection register (LVIM)

This register sets low-voltage detection and the operation mode.

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

The generation of a reset signal other than an LVI reset clears this register to 00H.

Figure 26-2. Format of Low-Voltage Detection Register (LVIM)

Address: FFBEH After reset: 00H[Note 1] R/W[Note 2]

Symbol	<7>	6	5	4	3	<2>	<1>	<0>
LVIM	LVION	0	0	0	0	LVISEL	LVIMD	LVIF

LVION[Notes 3, 4]	Enables low-voltage detection operation
0	Disables operation
1	Enables operation

LVISEL[Note 3]	Voltage detection selection
0	Detects level of supply voltage (V_{DD})
1	Detects level of input voltage from external input pin (EXLVI)

LVIMD[Note 3]	Low-voltage detection operation mode (interrupt/reset) selection
0	• LVISEL = 0: Generates an internal interrupt signal when the supply voltage (V_{DD}) drops lower than the detection voltage (V_{LVI}) ($V_{DD} < V_{LVI}$) or when V_{DD} becomes V_{LVI} or higher ($V_{DD} \geq V_{LVI}$). • LVISEL = 1: Generates an interrupt signal when the input voltage from an external input pin (EXLVI) drops lower than the detection voltage (V_{EXLVI}) (EXLVI $< V_{EXLVI}$) or when EXLVI becomes V_{EXLVI} or higher (EXLVI $\geq V_{EXLVI}$).
1	• LVISEL = 0: Generates an internal reset signal when the supply voltage (V_{DD}) < detection voltage (V_{LVI}) and releases the reset signal when $V_{DD} \geq V_{LVI}$. • LVISEL = 1: Generates an internal reset signal when the input voltage from an external input pin (EXLVI) < detection voltage (V_{EXLVI}) and releases the reset signal when EXLVI $\geq V_{EXLVI}$.

LVIF	Low-voltage detection flag
0	• LVISEL = 0: Supply voltage (V_{DD}) ≥ detection voltage (V_{LVI}), or when operation is disabled • LVISEL = 1: Input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI}), or when operation is disabled
1	• LVISEL = 0: Supply voltage (V_{DD}) < detection voltage (V_{LVI}) • LVISEL = 1: Input voltage from external input pin (EXLVI) < detection voltage (V_{EXLVI})

Notes 1. This bit is cleared to 00H upon a reset other than an LVI reset.

2. Bit 0 is read-only.

3. LVION, LVIMD, and LVISEL are cleared to 0 in the case of a reset other than an LVI reset. These are not cleared to 0 in the case of an LVI reset.

4. When LVION is set to 1, operation of the comparator in the LVI circuit is started. Use software to wait for an operation stabilization time (10 μs (MAX.)) from when LVION is set to 1 until operation is stabilized. After operation has stabilized, 200 μs (MIN.) are required from when a state below LVI detection voltage has been entered, until LVIF is set (1).

Cautions 1. To stop LVI, follow either of the procedures below.

- **When using 8-bit memory manipulation instruction: Write 00H to LVIM.**
- **When using 1-bit memory manipulation instruction: Clear LVION to 0.**

2. Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

Caution 3. When using LVI as an interrupt, if LVION is cleared (0) in a state below the LVI detection voltage, an INTLVI signal is generated and LVIIF becomes 1.

(2) Low-voltage detection level selection register (LVIS)

This register selects the low-voltage detection level.

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

The generation of a reset signal other than an LVI reset clears this register to 00H.

Figure 26-3. Format of Low-Voltage Detection Level Selection Register (LVIS)

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

Symbol	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	LVIS3	LVIS2	LVIS1	LVIS0

LVIS3	LVIS2	LVIS1	LVIS0	Detection level
0	0	0	0	V_{LVI0} (4.24 V ±0.1 V)
0	0	0	1	V_{LVI1} (4.09 V ±0.1 V)
0	0	1	0	V_{LVI2} (3.93 V ±0.1 V)
0	0	1	1	V_{LVI3} (3.78 V ±0.1 V)
0	1	0	0	V_{LVI4} (3.62 V ±0.1 V)
0	1	0	1	V_{LVI5} (3.47 V ±0.1 V)
0	1	1	0	V_{LVI6} (3.32 V ±0.1 V)
0	1	1	1	V_{LVI7} (3.16 V ±0.1 V)
1	0	0	0	V_{LVI8} (3.01 V ±0.1 V)
1	0	0	1	V_{LVI9} (2.85 V ±0.1 V)
1	0	1	0	V_{LVI10} (2.70 V ±0.1 V)
1	0	1	1	V_{LVI11} (2.55 V ±0.1 V)
1	1	0	0	V_{LVI12} (2.39 V ±0.1 V)
1	1	0	1	V_{LVI13} (2.24 V ±0.1 V)
1	1	1	0	V_{LVI14} (2.08 V ±0.1 V)
1	1	1	1	V_{LVI15} (1.93 V ±0.1 V)

Note The value of LVIS is not reset but retained as is, upon a reset by LVI. It is cleared to 00H upon other resets.

Cautions 1. Be sure to clear bits 4 to 7 to "0".

2. Do not change the value of LVIS during LVI operation.

3. When an input voltage from the external input pin (EXLVI) is detected, the detection voltage (V_{EXLVI} = 1.21 V (TYP.)) is fixed. Therefore, setting of LVIS is not necessary.

(3) Port mode register 12 (PM12)

When using the P120/EXLVI/INTP0 pin for external low-voltage detection potential input, set PM120 to 1. At this time, the output latch of P120 may be 0 or 1.

PM12 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PM12 to FFH.

Figure 26-4. Format of Port Mode Register 12 (PM12)

Address: FF2CH After reset: FFH R/W

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

PM120	P120 pin I/O mode selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

26.4 Operation of Low-Voltage Detector

The low-voltage detector can be used in the following two modes.

(1) Used as reset (LVIMD = 1)

- If LVISEL = 0, compares the supply voltage (V_{DD}) and detection voltage (V_{LVI}), generates an internal reset signal when $V_{DD} < V_{LVI}$, and releases internal reset when $V_{DD} \geq V_{LVI}$.
- If LVISEL = 1, compares the input voltage from external input pin (EXLVI) and detection voltage (V_{EXLVI} = 1.21 V (TYP.)), generates an internal reset signal when EXLVI < V_{EXLVI}, and releases internal reset when EXLVI ≥ V_{EXLVI}.

(2) Used as interrupt (LVIMD = 0)

- If LVISEL = 0, compares the supply voltage (V_{DD}) and detection voltage (V_{LVI}). When V_{DD} drops lower than V_{LVI} ($V_{DD} < V_{LVI}$) or when V_{DD} becomes V_{LVI} or higher ($V_{DD} \geq V_{LVI}$), generates an interrupt signal (INTLVI).
- If LVISEL = 1, compares the input voltage from external input pin (EXLVI) and detection voltage (V_{EXLVI} = 1.21 V (TYP.)). When EXLVI drops lower than V_{EXLVI} (EXLVI < V_{EXLVI}) or when EXLVI becomes V_{EXLVI} or higher (EXLVI ≥ V_{EXLVI}), generates an interrupt signal (INTLVI).

While the low-voltage detector is operating, whether the supply voltage or the input voltage from an external input pin is more than or less than the detection level can be checked by reading the low-voltage detection flag (LVIF: bit 0 of LVIM).

Remark LVIMD: Bit 1 of low-voltage detection register (LVIM)
LVISEL: Bit 2 of LVIM

26.4.1 When used as reset

(1) When detecting level of supply voltage (V_{DD})

- When starting operation

<1> Mask the LVI interrupt (LVIMK = 1).

<2> Clear bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 0 (detects level of supply voltage (V_{DD})) (default value).

<3> Set the detection voltage using bits 3 to 0 (LVIS3 to LVIS0) of the low-voltage detection level selection register (LVIS).

<4> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).

<5> Use software to wait for an operation stabilization time (10 μs (MAX.)).

<6> Wait until it is checked that (supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})) by bit 0 (LVIF) of LVIM.

<7> Set bit 1 (LVIMD) of LVIM to 1 (generates reset when the level is detected).

Figure 26-5 shows the timing of the internal reset signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <7> above.

Cautions 1. <1> must always be executed. When LVIMK = 0, an interrupt may occur immediately after the processing in <4>.

2. If supply voltage (V_{DD}) ≥ detection voltage (V_{LVI}) when LVIMD is set to 1, an internal reset signal is not generated.

- When stopping operation

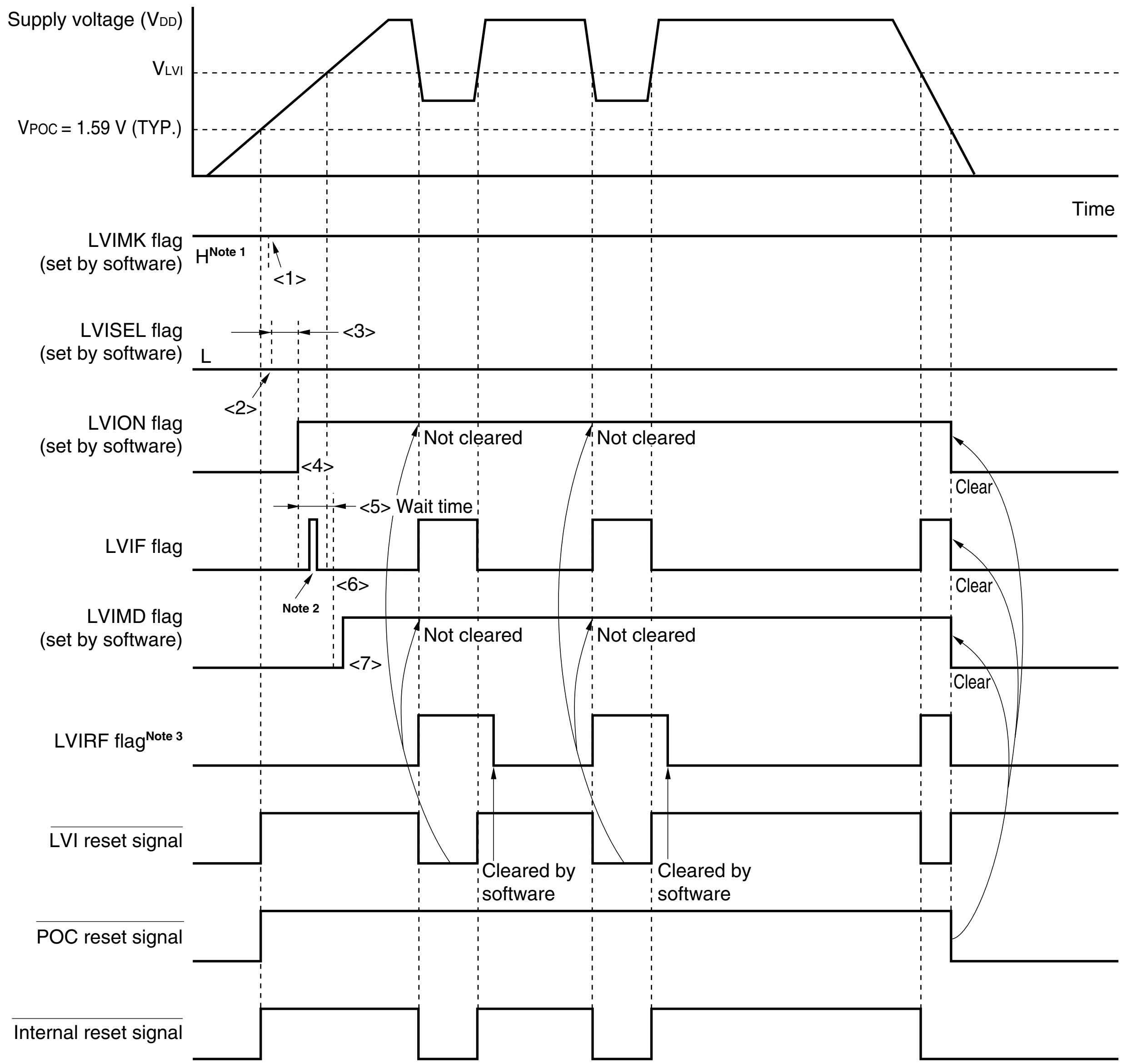
Either of the following procedures must be executed.

- When using 8-bit memory manipulation instruction:
  Write 00H to LVIM.

- When using 1-bit memory manipulation instruction:
  Clear LVIMD to 0 and then LVION to 0.

Figure 26-5. Timing of Low-Voltage Detector Internal Reset Signal Generation (Detects Level of Supply Voltage (V_{DD})) (1/2)

(1) In 1.59 V POC mode (option byte: POCMODE = 0)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

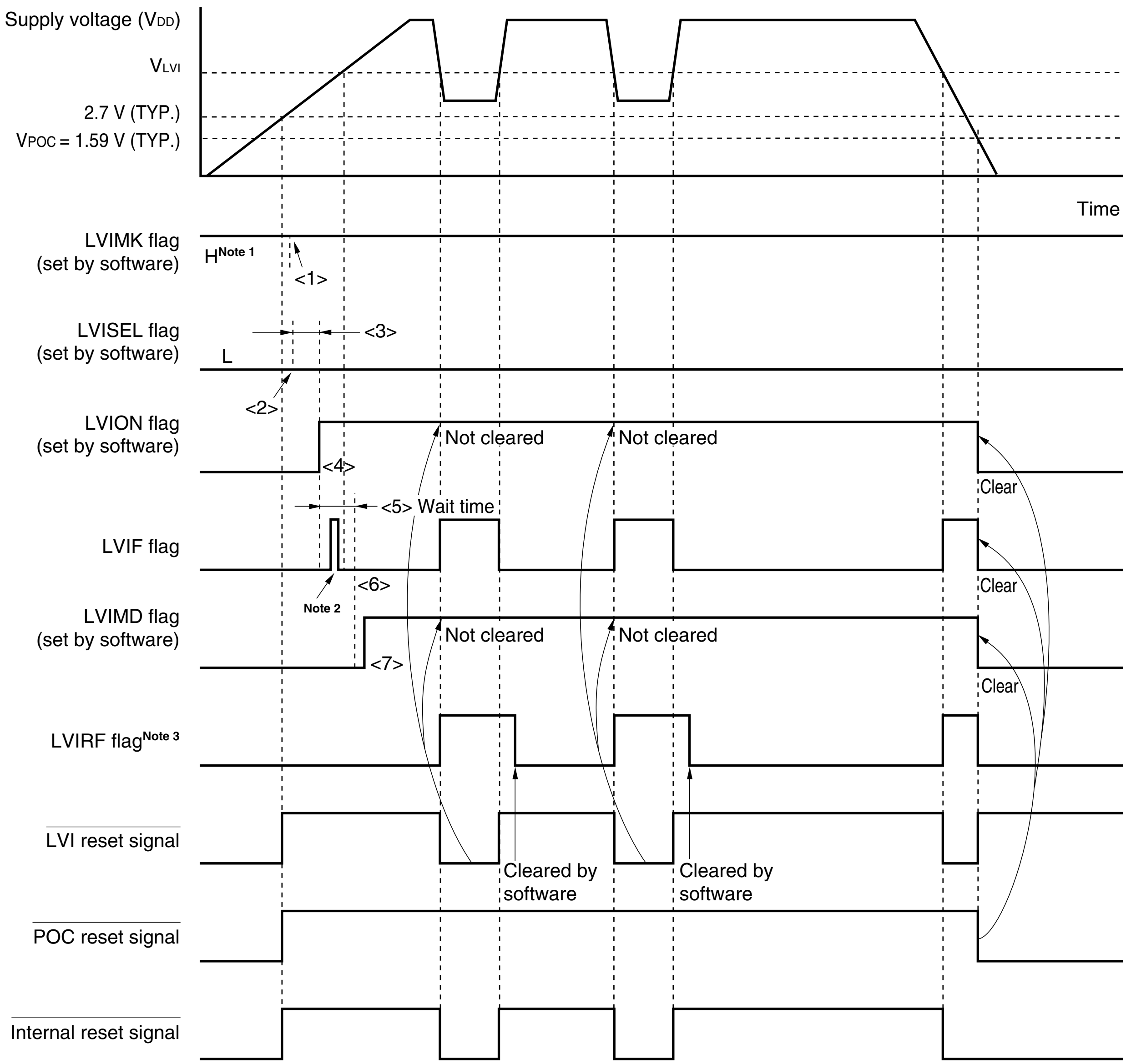
2. The LVIF flag may be set (1).

3. LVIRF is bit 0 of the reset control flag register (RESF). For details of RESF, see **CHAPTER 24 RESET FUNCTION**.

Remark <1> to <7> in Figure 26-5 above correspond to <1> to <7> in the description of "When starting operation" in **26.4.1 (1) When detecting level of supply voltage (V_{DD})**.

Figure 26-5. Timing of Low-Voltage Detector Internal Reset Signal Generation (Detects Level of Supply Voltage (V_{DD})) (2/2)

(2) In 2.7 V/1.59 V POC mode (option byte: POCMODE = 1)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. The LVIF flag may be set (1).

3. LVIRF is bit 0 of the reset control flag register (RESF). For details of RESF, see **CHAPTER 24 RESET FUNCTION**.

Remark <1> to <7> in Figure 26-5 above correspond to <1> to <7> in the description of "When starting operation" in **26.4.1 (1) When detecting level of supply voltage (V_{DD})**.

(2) When detecting level of input voltage from external input pin (EXLVI)

- When starting operation

<1> Mask the LVI interrupt (LVIMK = 1).
<2> Set bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 1 (detects level of input voltage from external input pin (EXLVI)).
<3> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
<4> Use software to wait for an operation stabilization time (10 μs (MAX.)).
<5> Wait until it is checked that (input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI} = 1.21 V (TYP.))) by bit 0 (LVIF) of LVIM.
<6> Set bit 1 (LVIMD) of LVIM to 1 (generates reset signal when the level is detected).

Figure 26-6 shows the timing of the internal reset signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <6> above.

Cautions 1. <1> must always be executed. When LVIMK = 0, an interrupt may occur immediately after the processing in <3>.

2. If input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI} = 1.21 V (TYP.)) when LVIMD is set to 1, an internal reset signal is not generated.

3. Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

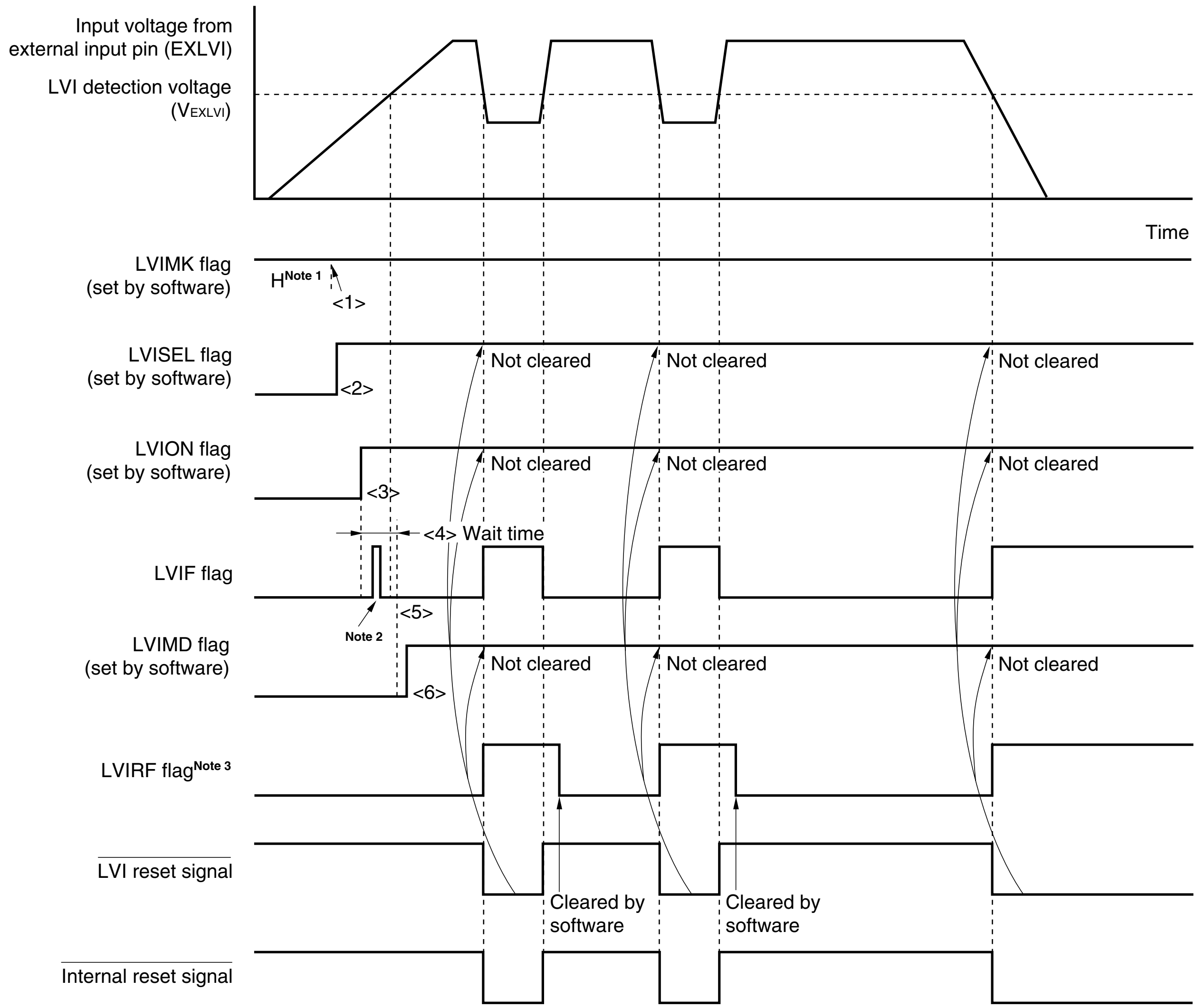
- When stopping operation

Either of the following procedures must be executed.

- When using 8-bit memory manipulation instruction:
  Write 00H to LVIM.

- When using 1-bit memory manipulation instruction:
  Clear LVIMD to 0 and then LVION to 0.

Figure 26-6. Timing of Low-Voltage Detector Internal Reset Signal Generation (Detects Level of Input Voltage from External Input Pin (EXLVI))



Notes 1. The LVIMK flag is set to “1” by reset signal generation.

2. The LVIF flag may be set (1).

3. LVIRF is bit 0 of the reset control flag register (RESF). For details of RESF, see **CHAPTER 24 RESET FUNCTION**.

Remark <1> to <6> in Figure 26-6 above correspond to <1> to <6> in the description of “When starting operation” in **26.4.1 (2) When detecting level of input voltage from external input pin (EXLVI)**.

26.4.2 When used as interrupt

(1) When detecting level of supply voltage (V_{DD})

- When starting operation
  <1> Mask the LVI interrupt (LVIMK = 1).
  <2> Clear bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 0 (detects level of supply voltage (V_{DD})) (default value).
  <3> Set the detection voltage using bits 3 to 0 (LVIS3 to LVIS0) of the low-voltage detection level selection register (LVIS).
  <4> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
  <5> Use software to wait for an operation stabilization time (10 μs (MAX.)).
  <6> Confirm that "supply voltage (V_{DD}) $\geq$ detection voltage (V_{LVI})" when detecting the falling edge of V_{DD}, or "supply voltage (V_{DD}) < detection voltage (V_{LVI})" when detecting the rising edge of V_{DD}, at bit 0 (LVIF) of LVIM.
  <7> Clear the interrupt request flag of LVI (LVIIF) to 0.
  <8> Release the interrupt mask flag of LVI (LVIMK).
  <9> Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).
  <10> Execute the EI instruction (when vector interrupts are used).

  Figure 26-7 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <9> above.

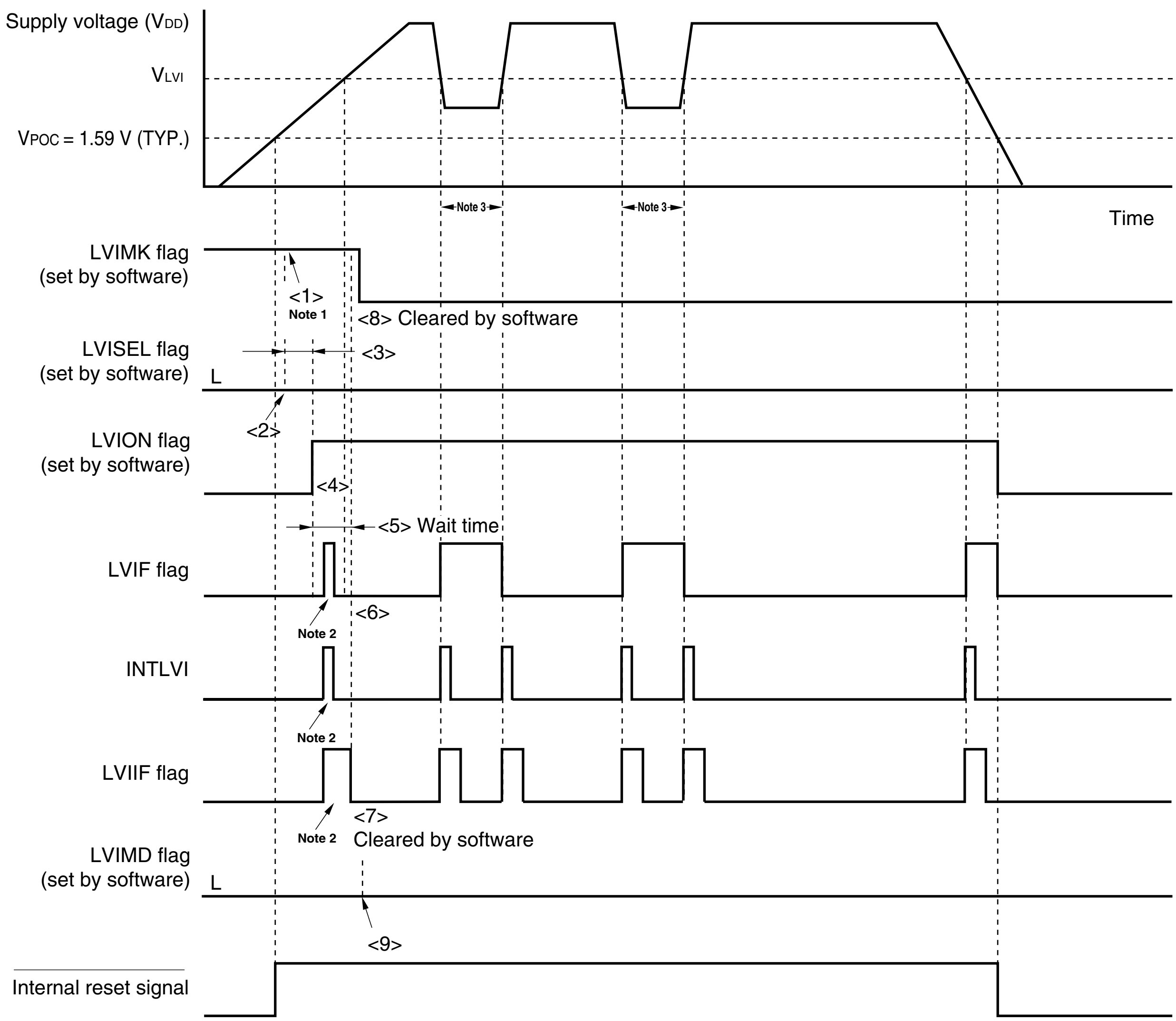
- When stopping operation
  Either of the following procedures must be executed.

  - When using 8-bit memory manipulation instruction:
    Write 00H to LVIM.

  - When using 1-bit memory manipulation instruction:
    Clear LVION to 0.

Figure 26-7. Timing of Low-Voltage Detector Interrupt Signal Generation (Detects Level of Supply Voltage (V_{DD})) (1/2)

(1) In 1.59 V POC mode (option byte: POCMODE = 0)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

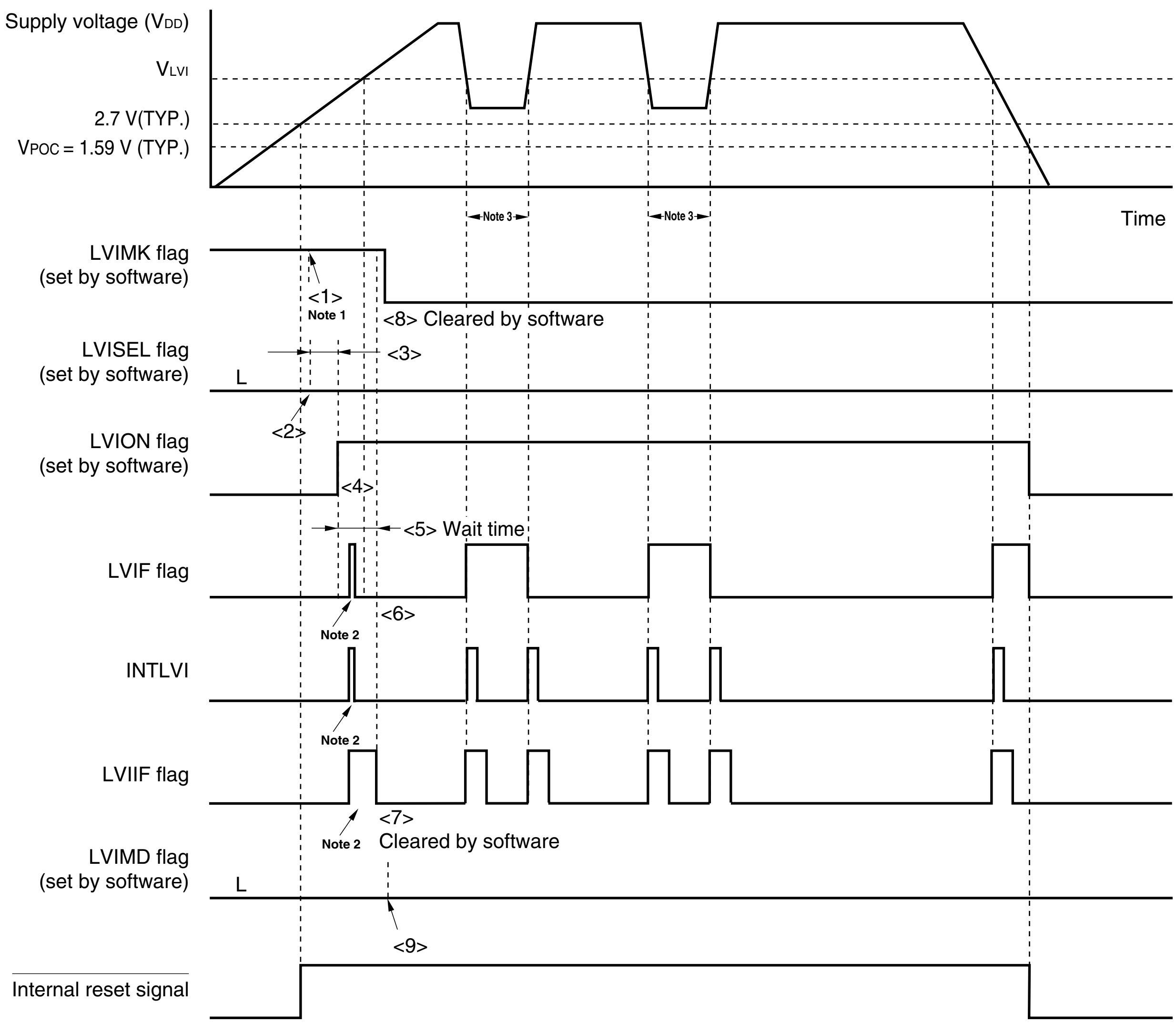
2. The interrupt request signal (INTLVI) is generated and the LVIF and LVIIF flags may be set (1).

3. If LVION is cleared (0) in a state below the LVI detection voltage, an INTLVI signal is generated and LVIIF becomes 1.

Remark <1> to <9> in Figure 26-7 above correspond to <1> to <9> in the description of "When starting operation" in **26.4.2 (1) When detecting level of supply voltage (V_{DD})**.

Figure 26-7. Timing of Low-Voltage Detector Interrupt Signal Generation (Detects Level of Supply Voltage (V_{DD})) (2/2)

(2) In 2.7 V/1.59 V POC mode (option byte: POCMODE = 1)



Notes 1. The LVIMK flag is set to “1” by reset signal generation.

2. The interrupt request signal (INTLVI) is generated and the LVIF and LVIIF flags may be set (1).

3. If LVION is cleared (0) in a state below the LVI detection voltage, an INTLVI signal is generated and LVIIF becomes 1.

Remark <1> to <9> in Figure 26-7 above correspond to <1> to <9> in the description of “When starting operation” in **26.4.2 (1) When detecting level of supply voltage (V_{DD})**.

(2) When detecting level of input voltage from external input pin (EXLVI)

- When starting operation

<1> Mask the LVI interrupt (LVIMK = 1).

<2> Set bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 1 (detects level of input voltage from external input pin (EXLVI)).

<3> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).

<4> Use software to wait for an operation stabilization time (10 μs (MAX.)).

<5> Confirm that "input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI} = 1.21 V (TYP.)" when detecting the falling edge of EXLVI, or "input voltage from external input pin (EXLVI) < detection voltage (V_{EXLVI} = 1.21 V (TYP.)" when detecting the rising edge of EXLVI, at bit 0 (LVIF) of LVIM.

<6> Clear the interrupt request flag of LVI (LVIIF) to 0.

<7> Release the interrupt mask flag of LVI (LVIMK).

<8> Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).

<9> Execute the EI instruction (when vector interrupts are used).

Figure 26-8 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <8> above.

Caution Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

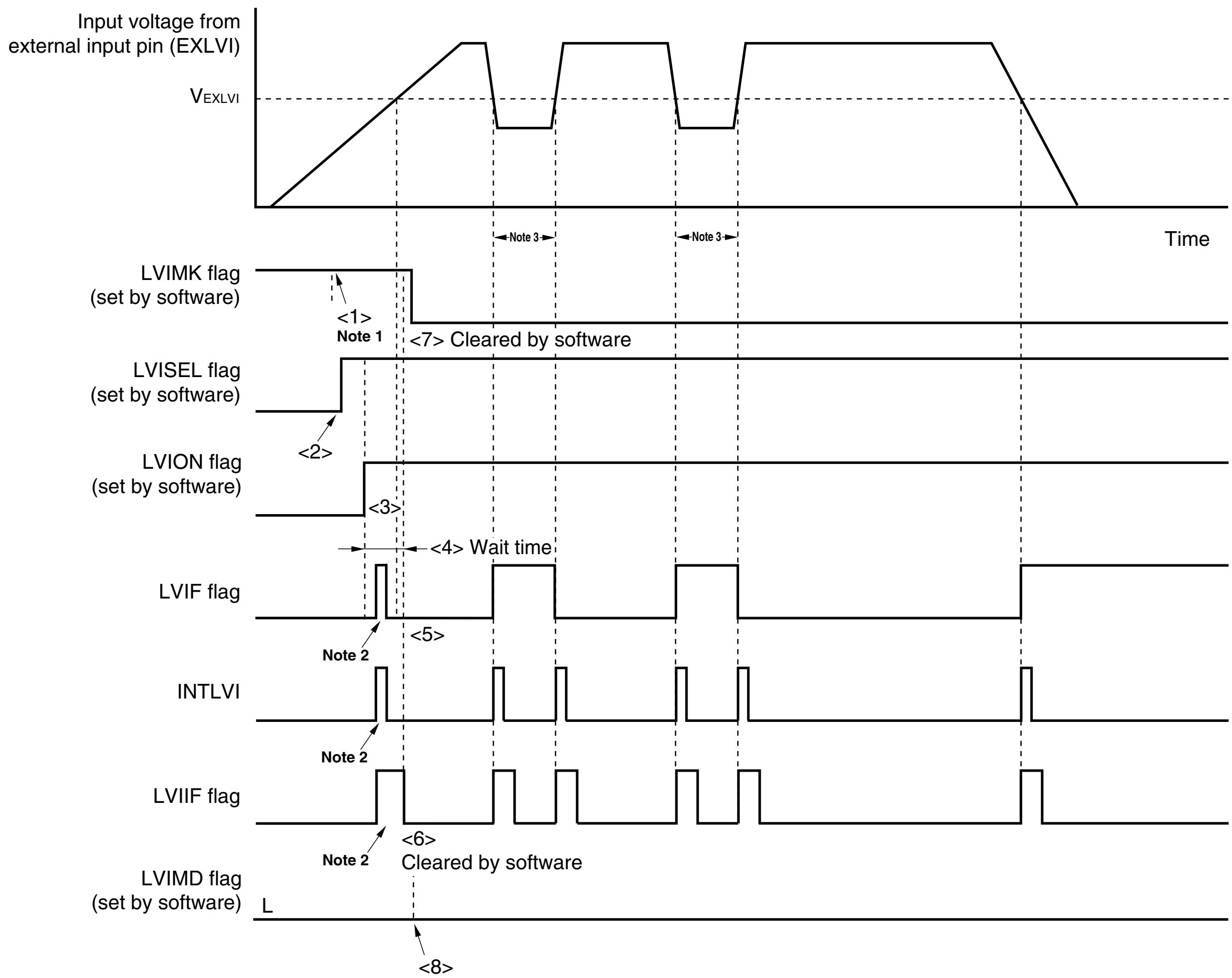
- When stopping operation

Either of the following procedures must be executed.

  - When using 8-bit memory manipulation instruction:
    Write 00H to LVIM.

  - When using 1-bit memory manipulation instruction:
    Clear LVION to 0.

Figure 26-8. Timing of Low-Voltage Detector Interrupt Signal Generation (Detects Level of Input Voltage from External Input Pin (EXLVI))



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. The interrupt request signal (INTLVI) is generated and the LVIF and LVIIF flags may be set (1).

3. If LVION is cleared (0) in a state below the LVI detection voltage, an INTLVI signal is generated and LVIIF becomes 1.

Remark <1> to <8> in Figure 26-8 above correspond to <1> to <8> in the description of "When starting operation" in **26.4.2 (2) When detecting level of input voltage from external input pin (EXLVI)**.

26.5 Cautions for Low-Voltage Detector

In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the LVI detection voltage (V_{LVI}), the operation is as follows depending on how the low-voltage detector is used.

(1) When used as reset

The system may be repeatedly reset and released from the reset status.

In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking action (1) below.

(2) When used as interrupt

Interrupt requests may be frequently generated. Take (b) of action (2) below.

<Action>

(1) When used as reset

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports (see **Figure 26-9**).

(2) When used as interrupt

(a) Confirm that "supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})" when detecting the falling edge of V_{DD}, or "supply voltage (V_{DD}) < detection voltage (V_{LVI})" when detecting the rising edge of V_{DD}, in the servicing routine of the LVI interrupt by using bit 0 (LVIF) of the low-voltage detection register (LVIM). Clear bit 0 (LVIIF) of interrupt request flag register 0L (IF0L) to 0.

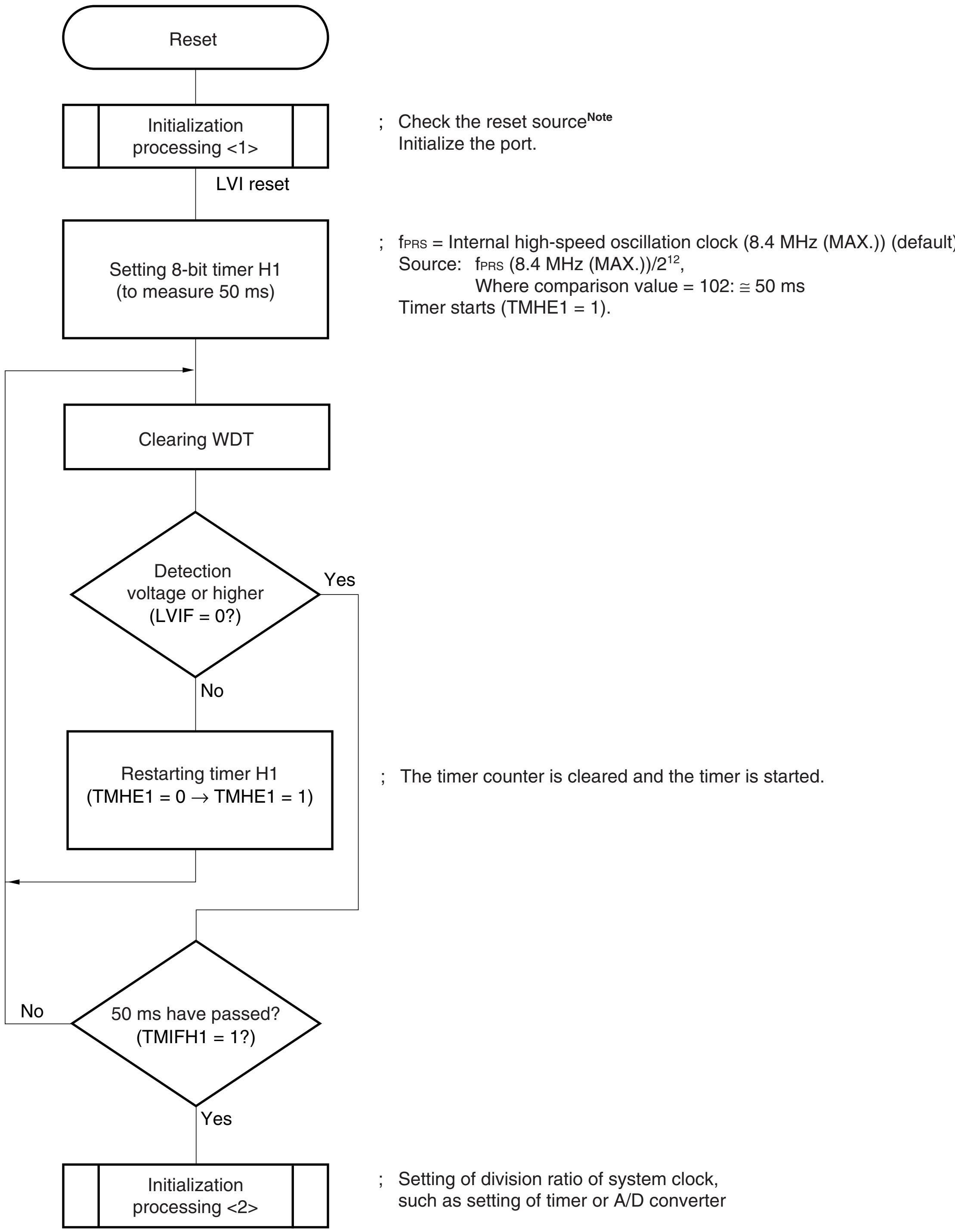
(b) In a system where the supply voltage fluctuation period is long in the vicinity of the LVI detection voltage, wait for the supply voltage fluctuation period, confirm that "supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})" when detecting the falling edge of V_{DD}, or "supply voltage (V_{DD}) < detection voltage (V_{LVI})" when detecting the rising edge of V_{DD}, using the LVIF flag, and clear the LVIIF flag to 0.

Remark If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to "1", the meanings of the above words change as follows.

- Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)
- Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)

Figure 26-9. Example of Software Processing After Reset Release (1/2)

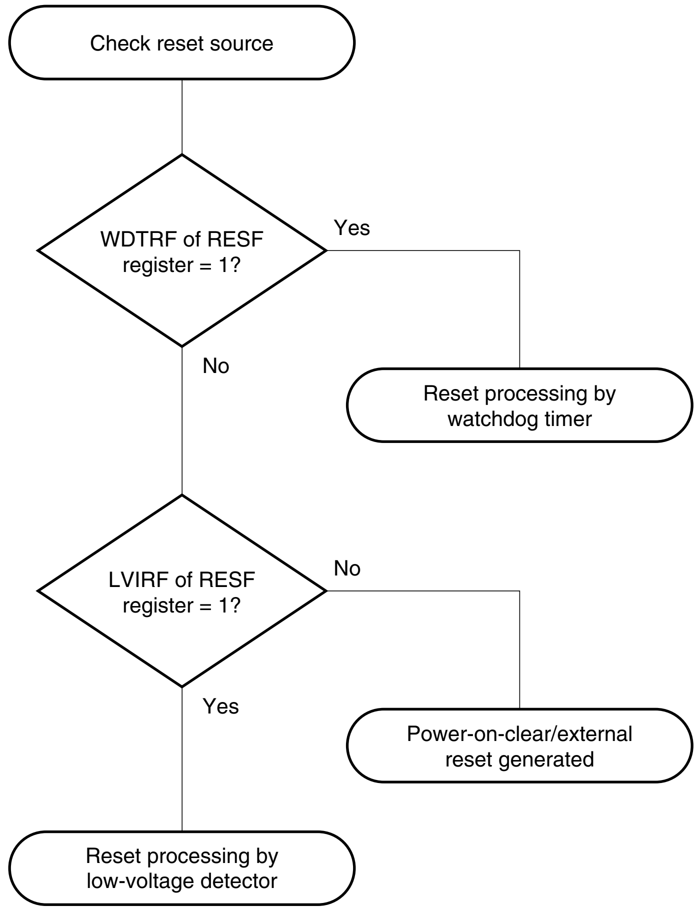
• If supply voltage fluctuation is 50 ms or less in vicinity of LVI detection voltage



Note A flowchart is shown on the next page.

<R> **Figure 26-9. Example of Software Processing After Reset Release (2/2)**

• Checking reset source



CHAPTER 27 OPTION BYTE

27.1 Functions of Option Bytes

The flash memory at 0080H to 0084H of the 78K0/LF3 is an option byte area. When power is turned on or when the device is restarted from the reset status, the device automatically references the option bytes and sets specified functions. When using the product, be sure to set the following functions by using the option bytes.

When the boot swap operation is used during self-programming, 0080H to 0084H are switched to 1080H to 1084H. Therefore, set values that are the same as those of 0080H to 0084H to 1080H to 1084H in advance.

Caution Be sure to set 00H to 0082H and 0083H (0082H/1082H and 0083H/1083H when the boot swap function is used).

(1) 0080H/1080H

○ Internal low-speed oscillator operation
- Can be stopped by software
- Cannot be stopped

○ Watchdog timer interval time setting

○ Watchdog timer counter operation
- Enabled counter operation
- Disabled counter operation

○ Watchdog timer window open period setting

Caution Set a value that is the same as that of 0080H to 1080H because 0080H and 1080H are switched during the boot swap operation.

(2) 0081H/1081H

○ Selecting POC mode
- During 2.7 V/1.59 V POC mode operation (POCMODE = 1)
  The device is in the reset state upon power application and until the supply voltage reaches 2.7 V (TYP.). It is released from the reset state when the voltage exceeds 2.7 V (TYP.). After that, POC is not detected at 2.7 V but is detected at 1.59 V (TYP.).
  If the supply voltage rises to 1.8 V after power application at a pace slower than 0.5 V/ms (MIN.), use of the 2.7 V/1.59 V POC mode is recommended.
- During 1.59 V POC mode operation (POCMODE = 0)
  The device is in the reset state upon power application and until the supply voltage reaches 1.59 V (TYP.). It is released from the reset state when the voltage exceeds 1.59 V (TYP.). After that, POC is detected at 1.59 V (TYP.), in the same manner as on power application.

Caution POCMODE can only be written by using a dedicated flash memory programmer. It cannot be set during self-programming or boot swap operation during self-programming (at this time, 1.59 V POC mode (default) is set). However, because the value of 1081H is copied to 0081H during the boot swap operation, it is recommended to set a value that is the same as that of 0081H to 1081H when the boot swap function is used.

(3) 0084H/1084H

○ On-chip debug operation control

- Disabling on-chip debug operation
- Enabling on-chip debug operation and erasing data of the flash memory in case authentication of the on-chip debug security ID fails
- Enabling on-chip debug operation and not erasing data of the flash memory even in case authentication of the on-chip debug security ID fails

Caution To use the on-chip debug function, set 02H or 03H to 0084H. Set a value that is the same as that of 0084H to 1084H because 0084H and 1084H are switched during the boot operation.

27.2 Format of Option Byte

The format of the option byte is shown below.

Figure 27-1. Format of Option Byte (1/2)

Address: 0080H/1080H[Note]

7	6	5	4	3	2	1	0
0	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	LSROSC

WINDOW1	WINDOW0	Watchdog timer window open period
0	0	25%
0	1	50%
1	0	75%
1	1	100%

WDTON	Operation control of watchdog timer counter/illegal access detection
0	Counter operation disabled (counting stopped after reset), illegal access detection operation disabled
1	Counter operation enabled (counting started after reset), illegal access detection operation enabled

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time
0	0	0	$2^{10}/f_{RL}$ (3.88 ms)
0	0	1	$2^{11}/f_{RL}$ (7.76 ms)
0	1	0	$2^{12}/f_{RL}$ (15.52 ms)
0	1	1	$2^{13}/f_{RL}$ (31.03 ms)
1	0	0	$2^{14}/f_{RL}$ (62.06 ms)
1	0	1	$2^{15}/f_{RL}$ (124.12 ms)
1	1	0	$2^{16}/f_{RL}$ (248.24 ms)
1	1	1	$2^{17}/f_{RL}$ (496.48 ms)

LSROSC	Internal low-speed oscillator operation
0	Can be stopped by software (stopped when 1 is written to bit 1 (LSRSTOP) of RCM register)
1	Cannot be stopped (not stopped even if 1 is written to LSRSTOP bit)

Note Set a value that is the same as that of 0080H to 1080H because 0080H and 1080H are switched during the boot swap operation.

Cautions 1. The combination of WDCS2 = WDCS1 = WDCS0 = 0 and WINDOW1 = WINDOW0 = 0 is prohibited.

2. The watchdog timer continues its operation during self-programming and EEPROM emulation of the flash memory. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

3. If LSROSC = 0 (oscillation can be stopped by software), the count clock is not supplied to the watchdog timer in the HALT and STOP modes, regardless of the setting of bit 1 (LSRSTOP) of the internal oscillation mode register (RCM).
When 8-bit timer H1 operates with the internal low-speed oscillation clock, the count clock is supplied to 8-bit timer H1 even in the HALT/STOP mode.

4. Be sure to clear bit 7 to 0.

Remarks 1. f_{RL}: Internal low-speed oscillation clock frequency

2. ( ): f_{RL} = 264 kHz (MAX.)

Figure 27-1. Format of Option Byte (2/2)

Address: 0081H/1081H[Notes 1, 2]

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	POCMODE

POCMODE	POC mode selection
0	1.59 V POC mode (default)
1	2.7 V/1.59 V POC mode

Notes 1. POCMODE can only be written by using a dedicated flash memory programmer. It cannot be set during self-programming or boot swap operation during self-programming (at this time, 1.59 V POC mode (default) is set). However, because the value of 1081H is copied to 0081H during the boot swap operation, it is recommended to set a value that is the same as that of 0081H to 1081H when the boot swap function is used.

2. To change the setting for the POC mode, set the value to 0081H again after batch erasure (chip erasure) of the flash memory. The setting cannot be changed after the memory of the specified block is erased.

Caution Be sure to clear bits 7 to 1 to “0”.

Address: 0082H/1082H, 0083H/1083H[Note]

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

Note Be sure to set 00H to 0082H and 0083H, as these addresses are reserved areas. Also set 00H to 1082H and 1083H because 0082H and 0083H are switched with 1082H and 1083H when the boot swap operation is used.

Address: 0084H/1084H[Note]

7	6	5	4	3	2	1	0
0	0	0	0	0	0	OCDEN1	OCDEN0

OCDEN1	OCDEN0	On-chip debug operation control
0	0	Operation disabled
0	1	Setting prohibited
1	0	Operation enabled. Does not erase data of the flash memory in case authentication of the on-chip debug security ID fails.
1	1	Operation enabled. Erases data of the flash memory in case authentication of the on-chip debug security ID fails.

Note To use the on-chip debug function, set 02H or 03H to 0084H. Set a value that is the same as that of 0084H to 1084H because 0084H and 1084H are switched during the boot swap operation.

Remark For the on-chip debug security ID, see **CHAPTER 29 ON-CHIP DEBUG FUNCTION**.

Here is an example of description of the software for setting the option bytes.

```
OPT      CSEG   AT 0080H
OPTION:  DB     30H          ; Enables watchdog timer operation (illegal access detection operation),
                             ; Window open period of watchdog timer: 50%,
                             ; Overflow time of watchdog timer: 2^10/fRL,
                             ; Internal low-speed oscillator can be stopped by software.
         DB     00H          ; 1.59 V POC mode
         DB     00H          ; Reserved area
         DB     00H          ; Reserved area
         DB     00H          ; On-chip debug operation disabled
```

Remark Referencing of the option byte is performed during reset processing. For the reset processing timing, see **CHAPTER 24 RESET FUNCTION**.

CHAPTER 28 FLASH MEMORY

The 78K0/LF3 incorporates the flash memory to which a program can be written, erased, and overwritten while mounted on the board.

28.1 Internal Memory Size Switching Register

The internal memory capacity can be selected using the internal memory size switching register (IMS).
IMS is set by an 8-bit memory manipulation instruction.
Reset signal generation sets IMS to CFH.

Caution Be sure to set each product to the values shown in Table 28-1 after a reset release.

Figure 28-1. Format of Internal Memory Size Switching Register (IMS)

Address: FFF0H After reset: CFH R/W

Symbol	7	6	5	4	3	2	1	0
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0

RAM2	RAM1	RAM0	Internal high-speed RAM capacity selection
0	0	0	768 bytes
1	1	0	1024 bytes
Other than above			Setting prohibited

ROM3	ROM2	ROM1	ROM0	Internal ROM capacity selection
0	1	0	0	16 KB
0	1	1	0	24 KB
1	0	0	0	32 KB
1	1	0	0	48 KB
1	1	1	1	60 KB
Other than above				Setting prohibited

Table 28-1. Internal Memory Size Switching Register Settings

Flash Memory Versions (78K0/LF3)	IMS Setting
μPD78F0471, 78F0481, 78F0491	04H
μPD78F0472, 78F0482, 78F0492	C6H
μPD78F0473, 78F0483, 78F0493	C8H
μPD78F0474, 78F0484, 78F0494	CCH
μPD78F0475, 78F0485, 78F0495	CFH

28.2 Internal Expansion RAM Size Switching Register

The internal expansion RAM capacity can be selected using the internal expansion RAM size switching register (IXS).

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

Reset signal generation sets IXS to 0CH.

Caution Be sure to set each product to the values shown in Table 28-2 after a reset release.

Figure 28-2. Format of Internal Expansion RAM Size Switching Register (IXS)

Address: FFF4H After reset: 0CH R/W

Symbol	7	6	5	4	3	2	1	0
IXS	0	0	0	IXRAM4	IXRAM3	IXRAM2	IXRAM1	IXRAM0

IXRAM4	IXRAM3	IXRAM2	IXRAM1	IXRAM0	Internal expansion RAM capacity selection
0	1	1	0	0	0 byte
0	1	0	1	0	1024 bytes
Other than above					Setting prohibited

Table 28-2. Internal Expansion RAM Size Switching Register Settings

Flash Memory Versions (78K0/LF3)	IXS Setting
μPD78F0471, 78F0481, 78F0491	0CH
μPD78F0472, 78F0482, 78F0492	
μPD78F0473, 78F0483, 78F0493	
μPD78F0474, 78F0484, 78F0494	0AH
μPD78F0475, 78F0485, 78F0495	

28.3 Writing with Flash memory programmer

Data can be written to the flash memory on-board or off-board, by using a dedicated flash memory programmer.

(1) On-board programming

The contents of the flash memory can be rewritten after the 78K0/LF3 has been mounted on the target system. The connectors that connect the dedicated flash memory programmer must be mounted on the target system.

(2) Off-board programming

Data can be written to the flash memory with a dedicated program adapter (FA series) before the 78K0/LF3 is mounted on the target system.

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

Table 28-3. Wiring Between 78K0/LF3 and Dedicated Flash memory programmer

Pin Configuration of Dedicated Flash memory programmer			With CSI10		With UART6	
Signal Name	I/O	Pin Function	Pin Name	Pin No.	Pin Name	Pin No.
SI/RxD	Input	Receive signal	SO10/TxD0/P13	78	TxD6/SEG18/P112	36
SO/TxD	Output	Transmit signal	SI10/RxD0/P12	79	RxD6/SEG19/P113	35
SCK	Output	Transfer clock	$\overline{SCK10}$/P11	80	–	–
CLK	Output	Clock to 78K0/LF3	–[Note 1]	–	**Note 2**	**Note 2**
/RESET	Output	Reset signal	$\overline{RESET}$	14	$\overline{RESET}$	14
FLMD0	Output	Mode signal	FLMD0	17	FLMD0	17
V_{DD}	I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	22	V_{DD}	22
			V_{DD}[Note 3]	59	V_{DD}[Note 3]	59
			AV_{REF}[Note 4]		AV_{REF}[Note 4]	
GND	–	Ground	V_{SS}	21	V_{SS}	21
			V_{SS}[Note 3]	60	V_{SS}[Note 3]	60
			AV_{SS}[Note 4]		AV_{SS}[Note 4]	

Notes 1. Only the internal high-speed oscillation clock (f_{RH}) can be used when CSI10 is used.

<R> **2.** Only the X1 clock (f_X), external main system clock (f_{EXCLK}), or internal high-speed oscillation clock (f_{RH}) can be used when UART6 is used. When using the clock output of the dedicated flash memory programmer, connect CLK of the PG-FP5 or FL-PR5 to EXCLK/X2/P122 (pin number 18).

3. μPD78F047x only.

4. μPD78F048x and 78F049x only.

<R> **Caution Only the bottom side pins (pin numbers 35 and 36) correspond to the UART6 pins (RxD6 and TxD6) when writing by a flash memory programmer. Writing cannot be performed by the top side pins (pin numbers 76 and 75).**

Examples of the recommended connection when using the adapter for flash memory writing are shown below.

Figure 28-3. Example of Wiring Adapter for Flash Memory Writing in 3-Wire Serial I/O (CSI10) Mode

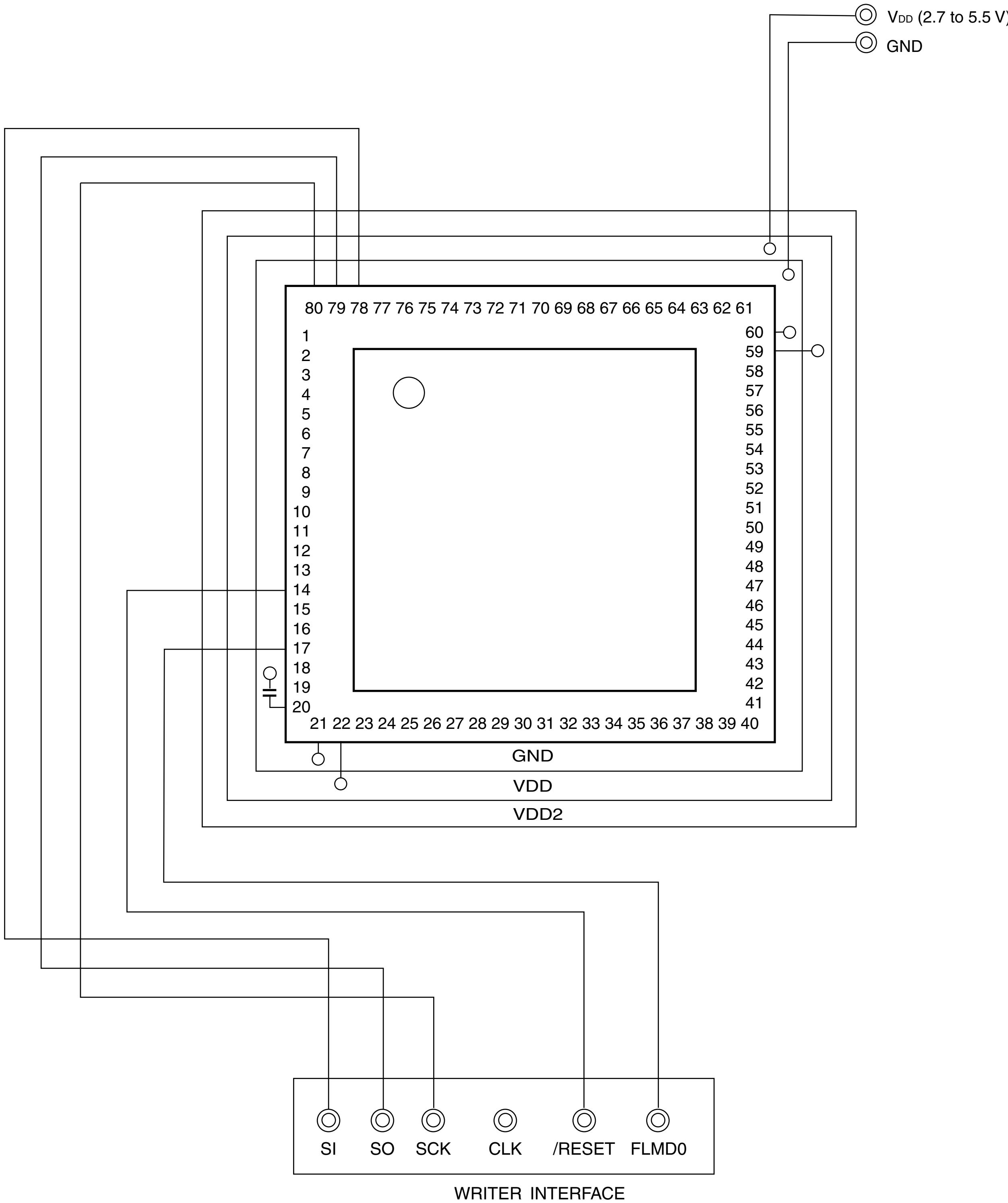
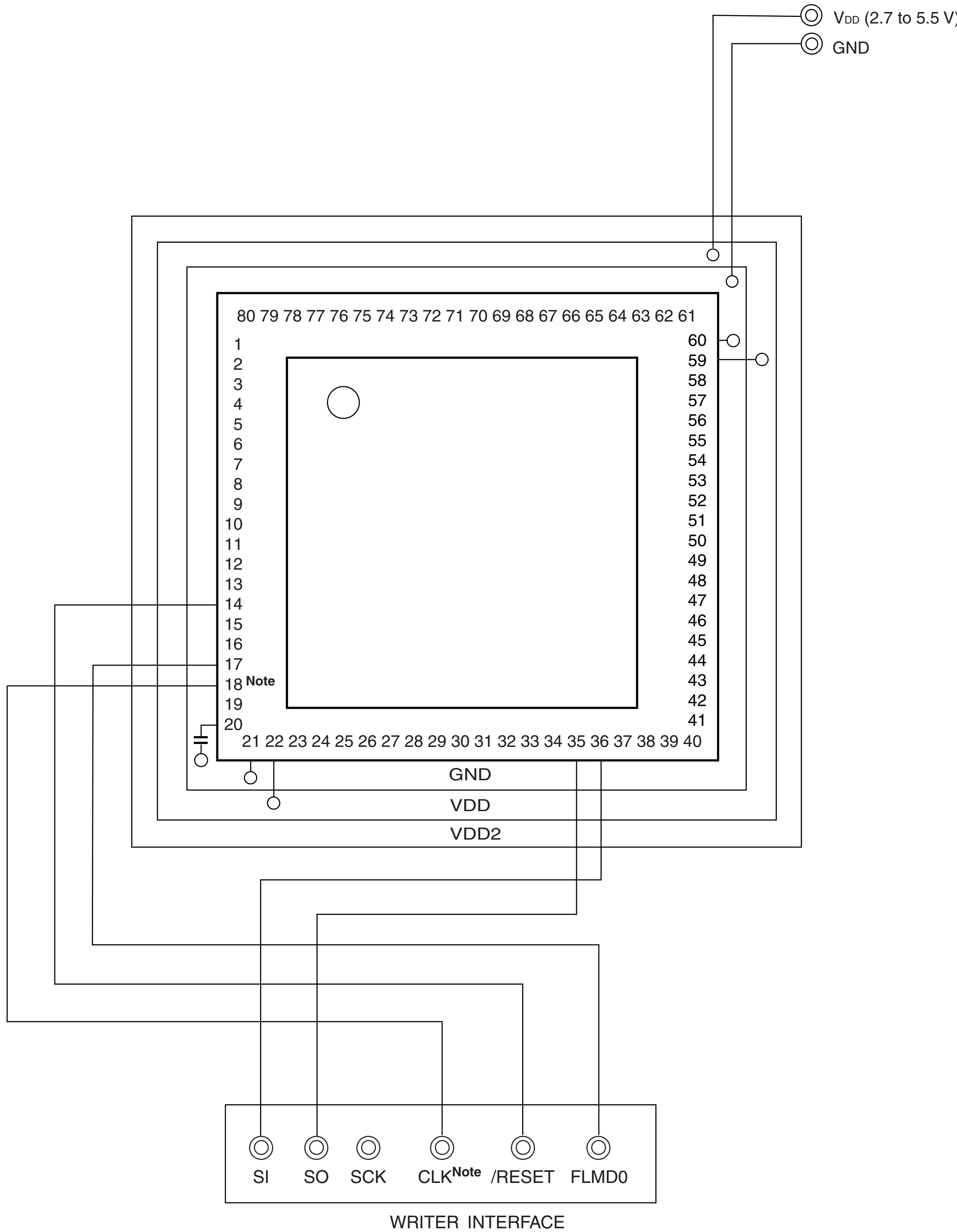


Figure 28-4. Example of Wiring Adapter for Flash Memory Writing in UART (UART6) Mode

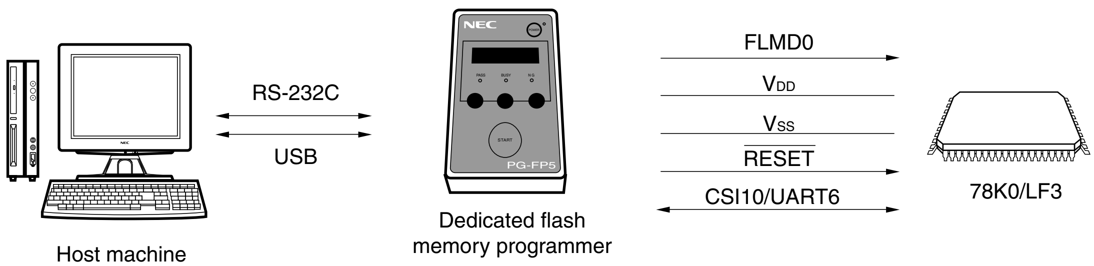


Note The above figure illustrates an example of wiring when using the clock output from the PG-FP5 or FL-PR5.

28.4 Programming Environment

The environment required for writing a program to the flash memory of the 78K0/LF3 is illustrated below.

Figure 28-5. Environment for Writing Program to Flash Memory



A host machine that controls the dedicated flash memory programmer is necessary.

To interface between the dedicated flash memory programmer and the 78K0/LF3, CSI10 or UART6 is used for manipulation such as writing and erasing. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

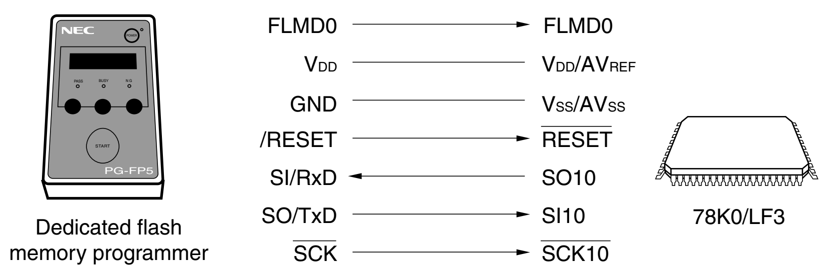
28.5 Communication Mode

Communication between the dedicated flash memory programmer and the 78K0/LF3 is established by serial communication via CSI10 or UART6 of the 78K0/LF3.

(1) CSI10

Transfer rate: 2.4 kHz to 2.5 MHz

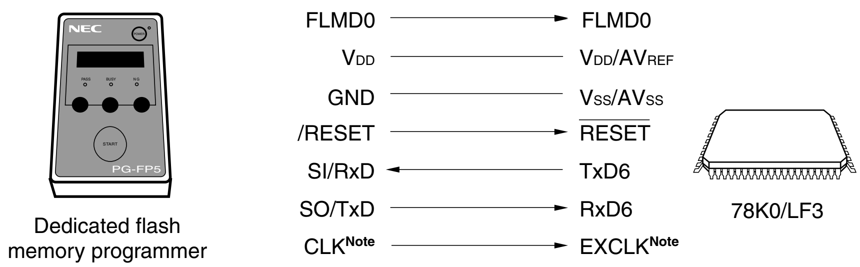
Figure 28-6. Communication with Dedicated Flash memory programmer (CSI10)



(2) UART6

Transfer rate: 115200 bps

Figure 28-7. Communication with Dedicated Flash memory programmer (UART6)



Note The above figure illustrates an example of wiring when using the clock output from the PG-FP5 or FL-PR5.

Caution Only the bottom side pins (pin numbers 35 and 36) correspond to the UART6 pins (RxD6 and TxD6) when writing by a flash memory programmer. Writing cannot be performed by the top side pins (pin numbers 76 and 75).

The dedicated flash memory programmer generates the following signals for the 78K0/LF3. For details, refer to the user's manual for the PG-FP5 or FL-PR5.

Table 28-4. Pin Connection

Dedicated Flash memory programmer			78K0/LF3	Connection	
Signal Name	I/O	Pin Function	Pin Name	CSI10	UART6
FLMD0	Output	Mode signal	FLMD0	◎	◎
V_{DD}	I/O	V_{DD} voltage generation/power monitoring	V_{DD}, AV_{REF}[Note 3]	◎	◎
GND	–	Ground	V_{SS}, AV_{SS}[Note 3]	◎	◎
CLK	Output	Clock output to 78K0/LF3	**Note 1**	×[Note 2]	○[Note 1]
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	◎	◎
SI/RxD	Input	Receive signal	SO10 or TxD6	◎	◎
SO/TxD	Output	Transmit signal	SI10 or RxD6	◎	◎
SCK	Output	Transfer clock	$\overline{\text{SCK10}}$	◎	×

<R> **Notes 1.** Only the X1 clock (f_X), external main system clock (f_{EXCLK}), or internal high-speed oscillation clock (f_{RH}) can be used when UART6 is used. When using the clock output of the dedicated flash memory programmer, connect CLK of the PG-FP5 or FL-PR5 to EXCLK/X2/P122.

2. Only the internal high-speed oscillation clock (f_{RH}) can be used when CSI10 is used.

3. μPD78F048x and 78F049x only.

Remark ◎: Be sure to connect the pin.

○: The pin does not have to be connected if the signal is generated on the target board.

×: The pin does not have to be connected.

28.6 Connection of Pins on Board

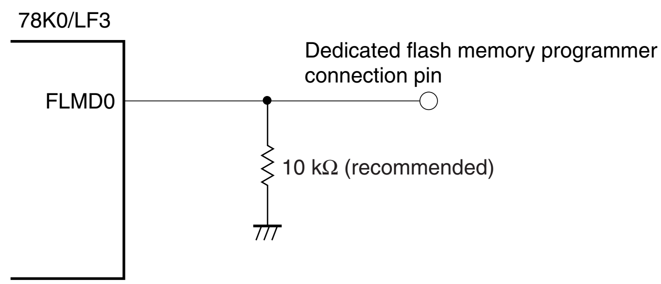
To write the flash memory on-board, connectors that connect the dedicated flash memory programmer must be provided on the target system. First provide a function that selects the normal operation mode or flash memory programming mode on the board.

When the flash memory programming mode is set, all the pins not used for programming the flash memory are in the same status as immediately after reset. Therefore, if the external device does not recognize the state immediately after reset, the pins must be handled as described below.

28.6.1 FLMD0 pin

In the normal operation mode, 0 V is input to the FLMD0 pin. In the flash memory programming mode, the V_{DD} write voltage is supplied to the FLMD0 pin. An FLMD0 pin connection example is shown below.

Figure 28-8. FLMD0 Pin Connection Example



28.6.2 Serial interface pins

The pins used by each serial interface are listed below.

Table 28-5. Pins Used by Each Serial Interface

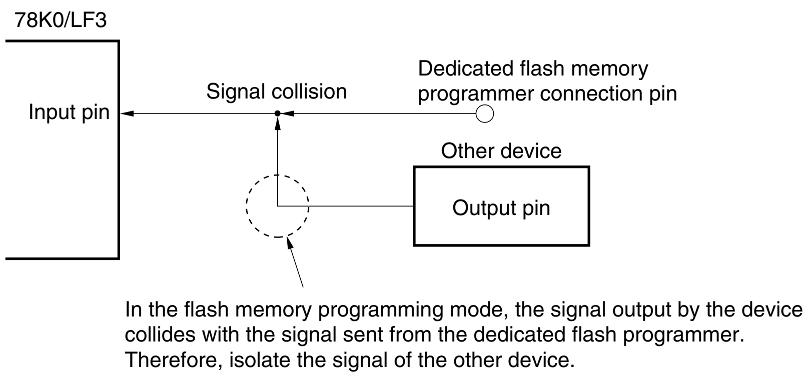
Serial Interface	Pins Used
CSI10	SO10, SI10, $\overline{SCK10}$
UART6	TxD6, RxD6

To connect the dedicated flash memory programmer to the pins of a serial interface that is connected to another device on the board, care must be exercised so that signals do not collide or that the other device does not malfunction.

(1) Signal collision

If the dedicated flash memory programmer (output) is connected to a pin (input) of a serial interface connected to another device (output), signal collision takes place. To avoid this collision, either isolate the connection with the other device, or make the other device go into an output high-impedance state.

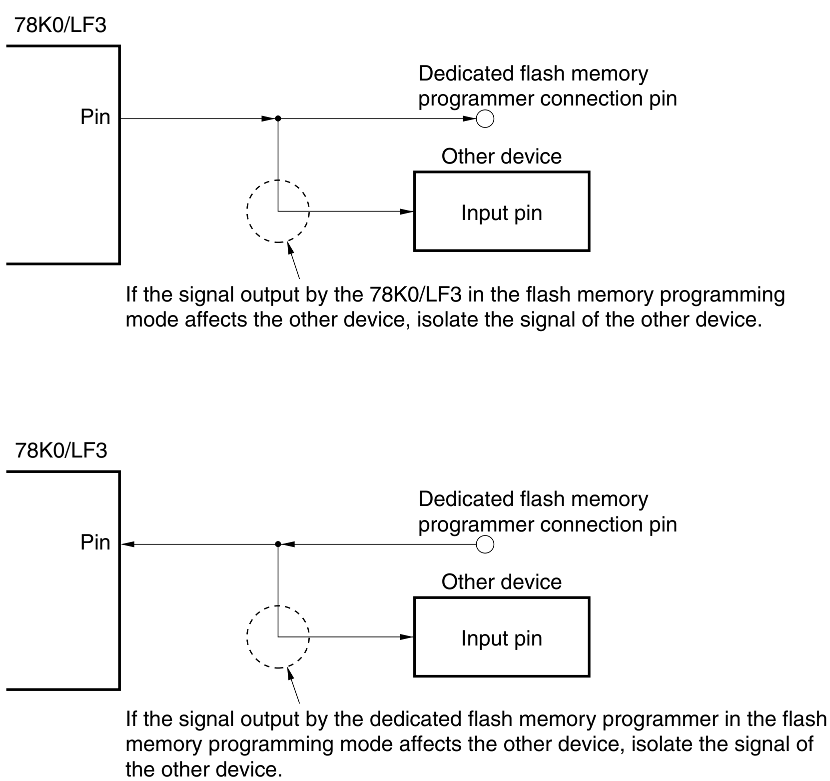
Figure 28-9. Signal Collision (Input Pin of Serial Interface)



(2) Malfunction of other device

If the dedicated flash memory programmer (output or input) is connected to a pin (input or output) of a serial interface connected to another device (input), a signal may be output to the other device, causing the device to malfunction. To avoid this malfunction, isolate the connection with the other device.

Figure 28-10. Malfunction of Other Device

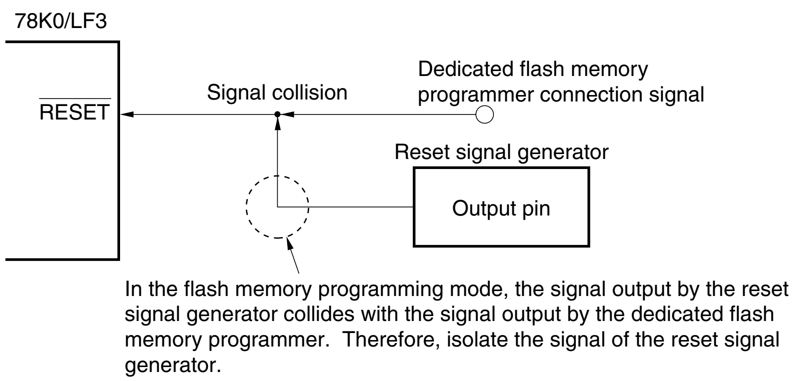


28.6.3 $\overline{\text{RESET}}$ pin

If the reset signal of the dedicated flash memory programmer is connected to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generator on the board, signal collision takes place. To prevent this collision, isolate the connection with the reset signal generator.

If the reset signal is input from the user system while the flash memory programming mode is set, the flash memory will not be correctly programmed. Do not input any signal other than the reset signal of the dedicated flash memory programmer.

Figure 28-11. Signal Collision ($\overline{\text{RESET}}$ Pin)



28.6.4 Port pins

When the flash memory programming mode is set, all the pins not used for flash memory programming enter the same status as that immediately after reset. If external devices connected to the ports do not recognize the port status immediately after reset, the port pin must be connected to V_{DD} or V_{SS} via a resistor.

28.6.5 REGC pin

Connect the REGC pin to GND via a capacitor (0.47 to 1 μF: recommended) in the same manner as during normal operation.

28.6.6 Other signal pins

Connect X1 and X2 in the same status as in the normal operation mode when using the on-board clock.

To input the operating clock from the dedicated flash memory programmer, however, connect CLK of the PG-FP5 or FL-PR5 to EXCLK/X2/P122.

Cautions 1. Only the internal high-speed oscillation clock (f_{RH}) can be used when CSI10 is used.

<R> **2. The X1 clock (f_X), external main system clock (f_{EXCLK}), or internal high-speed oscillation clock (f_{RH}) can be used when UART6 is used.**

28.6.7 Power supply

To use the supply voltage output of the flash memory programmer, connect the V_{DD} pin to V_{DD} of the flash memory programmer, and the V_{SS} pin to GND of the flash memory programmer.

To use the on-board supply voltage, connect in compliance with the normal operation mode.

However, be sure to connect the V_{DD} and V_{SS} pins to V_{DD} and GND of the flash memory programmer to use the power monitor function with the flash memory programmer, even when using the on-board supply voltage.

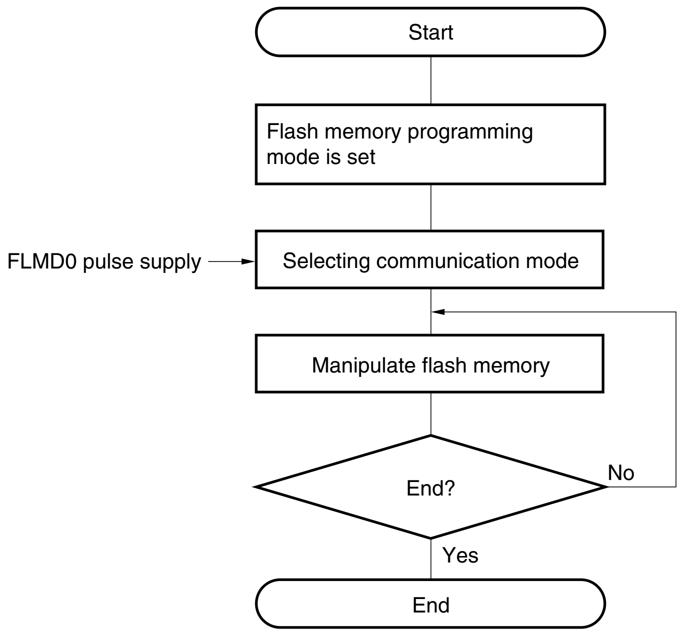
Supply the same other power supplies (AV_{REF} and AV_{SS}) as those in the normal operation mode.

28.7 Programming Method

28.7.1 Controlling flash memory

The following figure illustrates the procedure to manipulate the flash memory.

<R> **Figure 28-12. Flash Memory Manipulation Procedure**



28.7.2 Flash memory programming mode

To rewrite the contents of the flash memory by using the dedicated flash memory programmer, set the 78K0/LF3 in the flash memory programming mode. To set the mode, set the FLMD0 pin to V_{DD} and clear the reset signal.

Change the mode by using a jumper when writing the flash memory on-board.

Figure 28-13. Flash Memory Programming Mode

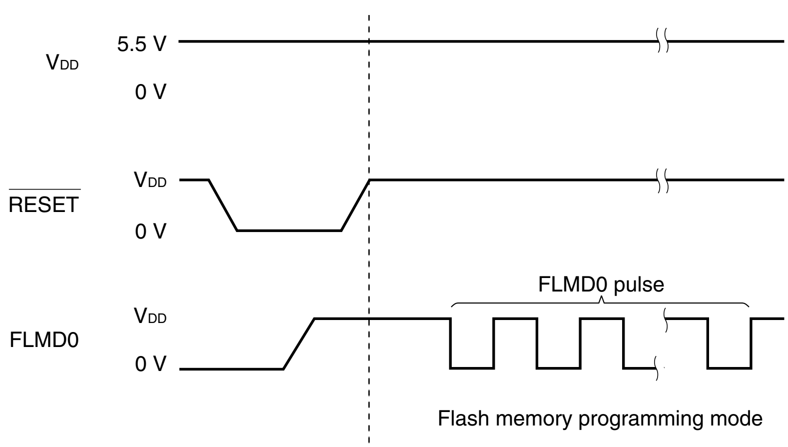


Table 28-6. Relationship Between FLMD0 Pin and Operation Mode After Reset Release

FLMD0	Operation Mode
0	Normal operation mode
V_{DD}	Flash memory programming mode

28.7.3 Selecting communication mode

In the 78K0/LF3, a communication mode is selected by inputting pulses to the FLMD0 pin after the dedicated flash memory programming mode is entered. These FLMD0 pulses are generated by the flash memory programmer.

The following table shows the relationship between the number of pulses and communication modes.

Table 28-7. Communication Modes

Communication Mode	Standard Setting[Note 1]				Pins Used	Peripheral Clock	Number of FLMD0 Pulses
	Port	Speed	Frequency	Multiply Rate			
UART (UART6)	UART-Ext-OSC	115,200 bps[Note 3]	2 M to 10 MHz[Note 2]	1.0	TxD6, RxD6	f_X	0
	UART-Ext-FP5CLK					f_{EXCLK}	3
<R>	UART-Internal-OSC		–			f_{RH}	5
3-wire serial I/O (CSI10)	CSI-Internal-OSC	2.4 kHz to 2.5 MHz	–		SO10, SI10, $\overline{SCK10}$	f_{RH}	8

Notes 1. Selection items for Standard settings on GUI of the flash memory programmer.

2. The possible setting range differs depending on the voltage. For details, see **CHAPTER 31 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)**.

3. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

Caution When UART6 is selected, the receive clock is calculated based on the reset command sent from the dedicated flash memory programmer after the FLMD0 pulse has been received.

Remark f_X: X1 clock

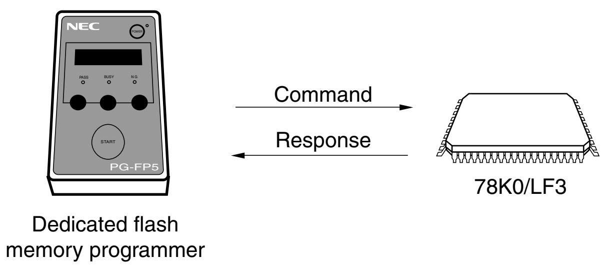
f_{EXCLK}: External main system clock

f_{RH}: Internal high-speed oscillation clock

28.7.4 Communication commands

The 78K0/LF3 communicates with the dedicated flash memory programmer by using commands. The signals sent from the flash memory programmer to the 78K0/LF3 are called commands, and the signals sent from the 78K0/LF3 to the dedicated flash memory programmer are called response.

Figure 28-14. Communication Commands



The flash memory control commands of the 78K0/LF3 are listed in the table below. All these commands are issued from the programmer and the 78K0/LF3 perform processing corresponding to the respective commands.

Table 28-8. Flash Memory Control Commands

Classification	Command Name	Function
Verify	Verify	Compares the contents of a specified area of the flash memory with data transmitted from the programmer.
Erase	Chip Erase	Erases the entire flash memory.
	Block Erase	Erases a specified area in the flash memory.
Blank check	Block Blank Check	Checks if a specified block in the flash memory has been correctly erased.
Write	Programming	Writes data to a specified area in the flash memory.
Getting information	Status	Gets the current operating status (status data).
	Silicon Signature	Gets 78K0/Lx3 information (such as the part number and flash memory configuration).
	Version Get	Gets the 78K0/Lx3 version and firmware version.
	Checksum	Gets the checksum data for a specified area.
Security	Security Set	Sets security information.
Others	Reset	Used to detect synchronization status of communication.
	Oscillating Frequency Set	Specifies an oscillation frequency.

The 78K0/LF3 return a response for the command issued by the dedicated flash memory programmer. The response names sent from the 78K0/LF3 are listed below.

Table 28-9. Response Names

Response Name	Function
ACK	Acknowledges command/data.
NAK	Acknowledges illegal command/data.

28.8 Security Settings

The 78K0/LF3 supports a security function that prohibits rewriting the user program written to the internal flash memory, so that the program cannot be changed by an unauthorized person.

The operations shown below can be performed using the Security Set command. The security setting is valid when the programming mode is set next.

- Disabling batch erase (chip erase)
  Execution of the block erase and batch erase (chip erase) commands for entire blocks in the flash memory is prohibited by this setting during on-board/off-board programming. Once execution of the batch erase (chip erase) command is prohibited, all of the prohibition settings (including prohibition of batch erase (chip erase)) can no longer be cancelled.

  Caution After the security setting for the batch erase is set, erasure cannot be performed for the device. In addition, even if a write command is executed, data different from that which has already been written to the flash memory cannot be written, because the erase command is disabled.

- Disabling block erase
  Execution of the block erase command for a specific block in the flash memory is prohibited during on-board/off-board programming. However, blocks can be erased by means of self programming.

- Disabling write
  Execution of the write and block erase commands for entire blocks in the flash memory is prohibited during on-board/off-board programming. However, blocks can be written by means of self programming.

- Disabling rewriting boot cluster 0
  Execution of the batch erase (chip erase) command, block erase command, and write command on boot cluster 0 (0000H to 0FFFH) in the flash memory is prohibited by this setting.

  Caution If a security setting that rewrites boot cluster 0 has been applied, boot cluster 0 of that device will not be rewritten.

The batch erase (chip erase), block erase, write commands, and rewriting boot cluster 0 are enabled by the default setting when the flash memory is shipped. Security can be set by on-board/off-board programming and self programming. Each security setting can be used in combination.

Prohibition of erasing blocks and writing is cleared by executing the batch erase (chip erase) command.

Table 28-10 shows the relationship between the erase and write commands when the 78K0/LF3 security function is enabled.

Table 28-10. Relationship Between Enabling Security Function and Command

(1) During on-board/off-board programming

Valid Security	Executed Command		
	Batch Erase (Chip Erase)	Block Erase	Write
Prohibition of batch erase (chip erase)	Cannot be erased in batch	Blocks cannot be erased.	Can be performed[Note].
Prohibition of block erase	Can be erased in batch.		Can be performed.
Prohibition of writing			Cannot be performed.
Prohibition of rewriting boot cluster 0	Cannot be erased in batch	Boot cluster 0 cannot be erased.	Boot cluster 0 cannot be written.

Note Confirm that no data has been written to the write area. Because data cannot be erased after batch erase (chip erase) is prohibited, do not write data if the data has not been erased.

(2) During self programming

Valid Security	Executed Command	
	Block Erase	Write
Prohibition of batch erase (chip erase)	Blocks can be erased.	Can be performed.
Prohibition of block erase		
Prohibition of writing		
Prohibition of rewriting boot cluster 0	Boot cluster 0 cannot be erased.	Boot cluster 0 cannot be written.

Table 28-11 shows how to perform security settings in each programming mode.

Table 28-11. Setting Security in Each Programming Mode

(1) On-board/off-board programming

Security	Security Setting	How to Disable Security Setting
Prohibition of batch erase (chip erase)	Set via GUI of dedicated flash memory programmer, etc.	Cannot be disabled after set.
Prohibition of block erase		Execute batch erase (chip erase) command
Prohibition of writing		
Prohibition of rewriting boot cluster 0		Cannot be disabled after set.

(2) Self programming

Security	Security Setting	How to Disable Security Setting
Prohibition of batch erase (chip erase)	Set by using information library.	Cannot be disabled after set.
Prohibition of block erase		Execute batch erase (chip erase) command during on-board/off-board programming (cannot be disabled during self programming)
Prohibition of writing		
Prohibition of rewriting boot cluster 0		Cannot be disabled after set.

<R> 28.9 Processing Time for Each Command When PG-FP5 Is Used (Reference)

The following table shows the processing time for each command (reference) when the PG-FP5 is used as a dedicated flash memory programmer.

Table 28-12. Processing Time for Each Command When PG-FP5 Is Used (Reference) (1/3)

(1) μPD78F0471, 78F0481, 78F0491 (Products with internal ROM: 16 KB)

Command of PG-FP5	Port: CSI-Internal-OSC (internal high-speed oscillation clock (f_{RH})) Speed: 2.5 MHz	Port: UART-Internal-OSC (internal high-speed oscillation clock (f_{RH})) Speed: 115200 bps	Port:UART-Ext-OSC (X1 clock (f_X)) Speed:115200 bps		Port: UART-Ext-FP5CLK (External main system clock (f_{EXCLK})), Speed:115200 bps	
			Frequency: 2.0 MHz	Frequency: 10 MHz	Frequency: 2.0 MHz	Frequency: 10 MHz
Signature	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)
Blankcheck	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)
Erase	1.5 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)
Program	3 s (TYP.)	4 s (TYP.)	4 s (TYP.)	4 s (TYP.)	4 s (TYP.)	4 s (TYP.)
Verify	2 s (TYP.)	3 s (TYP.)	3 s (TYP.)	3 s (TYP.)	3 s (TYP.)	3 s (TYP.)
E.P.V	3.5 s (TYP.)	4 s (TYP.)	4 s (TYP.)	4 s (TYP.)	4 s (TYP.)	4 s (TYP.)
Checksum	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)
Security	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)

(2) μPD78F0472, 78F0482, 78F0492 (Products with internal ROM: 24 KB)

Command of PG-FP5	Port: CSI-Internal-OSC (internal high-speed oscillation clock (f_{RH})) Speed: 2.5 MHz	Port: UART-Internal-OSC (internal high-speed oscillation clock (f_{RH})) Speed: 115200 bps	Port:UART-Ext-OSC (X1 clock (f_X)) Speed:115200 bps		Port:UART-Ext-FP5CLK (External main system clock (f_{EXCLK})) Speed:115200 bps	
			Frequency: 2.0 MHz	Frequency: 10 MHz	Frequency: 2.0 MHz	Frequency: 10 MHz
Signature	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)
Blankcheck	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)
Erase	1.5 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)
Program	4 s (TYP.)	5.5 s (TYP.)	5.5 s (TYP.)	5.5 s (TYP.)	5.5 s (TYP.)	5.5 s (TYP.)
Verify	2.5 s (TYP.)	4 s (TYP.)	4 s (TYP.)	4 s (TYP.)	4 s (TYP.)	4 s (TYP.)
E.P.V	4.5 s (TYP.)	5.5 s (TYP.)	5.5 s (TYP.)	5.5 s (TYP.)	5.5 s (TYP.)	5.5 s (TYP.)
Checksum	1.5 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)
Security	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)

Caution When executing boot swapping, do not use the E.P.V. command with the dedicated flash memory programmer.

Table 28-12. Processing Time for Each Command When PG-FP5 Is Used (Reference) (2/3)

(3) μ PD78F0473, 78F0483, 78F0493 (Products with internal ROM: 32 KB)

Command of PG-FP5	Port: CSI-Internal-OSC (internal high-speed oscillation clock (f_{RH})) Speed: 2.5 MHz	Port: UART-Internal-OSC (internal high-speed oscillation clock (f_{RH})) Speed: 115200 bps	Port:UART-Ext-OSC (X1 clock (f_X)) Speed:115200 bps		Port:UART-Ext-FP5CLK (External main system clock (f_{EXCLK})) Speed:115200 bps	
			Frequency: 2.0 MHz	Frequency: 10 MHz	Frequency: 2.0 MHz	Frequency: 10 MHz
Signature	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)
Blankcheck	1.5 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)
Erase	1.5 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)
Program	4.5 s (TYP.)	6.5 s (TYP.)	6.5 s (TYP.)	6.5 s (TYP.)	6.5 s (TYP.)	6.5 s (TYP.)
Verify	2.5 s (TYP.)	5 s (TYP.)	5 s (TYP.)	5 s (TYP.)	5 s (TYP.)	5 s (TYP.)
E.P.V	5.5 s (TYP.)	7 s (TYP.)	7 s (TYP.)	7 s (TYP.)	7 s (TYP.)	7 s (TYP.)
Checksum	1.5 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)
Security	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)

(4) μ PD78F0474, 78F0484, 78F0494 (Products with internal ROM: 48 KB)

Command of PG-FP5	Port: CSI-Internal-OSC (internal high-speed oscillation clock (f_{RH})) Speed: 2.5 MHz	Port: UART-Internal-OSC (internal high-speed oscillation clock (f_{RH})) Speed: 115200 bps	Port:UART-Ext-OSC (X1 clock (f_X)) Speed:115200 bps		Port:UART-Ext-FP5CLK (External main system clock (f_{EXCLK})) Speed:115200 bps	
			Frequency: 2.0 MHz	Frequency: 10 MHz	Frequency: 2.0 MHz	Frequency: 10 MHz
Signature	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)
Blankcheck	1 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)
Erase	1.5 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)
Program	6.5 s (TYP.)	9.5 s (TYP.)	9.5 s (TYP.)	9.5 s (TYP.)	9.5 s (TYP.)	9.5 s (TYP.)
Verify	3.5 s (TYP.)	7 s (TYP.)	7 s (TYP.)	7 s (TYP.)	7 s (TYP.)	7 s (TYP.)
E.P.V	7.5 s (TYP.)	10 s (TYP.)	10 s (TYP.)	10 s (TYP.)	10 s (TYP.)	10 s (TYP.)
Checksum	1.5 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)
Security	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)

Caution When executing boot swapping, do not use the E.P.V. command with the dedicated flash memory programmer.

Table 28-12. Processing Time for Each Command When PG-FP5 Is Used (Reference) (3/3)

(5) μPD78F0475, 78F0485, 78F0495 (Products with internal ROM: 60 KB)

Command of PG-FP5	Port: CSI-Internal-OSC (internal high-speed oscillation clock (f_{RH})) Speed: 2.5 MHz	Port: UART-Internal-OSC (internal high-speed oscillation clock (f_{RH})) Speed: 115200 bps	Port:UART-Ext-OSC (X1 clock (f_X)) Speed:115200 bps		Port:UART-Ext-FP5CLK (External main system clock (f_{EXCLK})) Speed:115200 bps	
			Frequency: 2.0 MHz	Frequency: 10 MHz	Frequency: 2.0 MHz	Frequency: 10 MHz
Signature	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)
Blankcheck	1.5 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)
Erase	2 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)
Program	8 s (TYP.)	12 s (TYP.)	12 s (TYP.)	11.5 s (TYP.)	12 s (TYP.)	11.5 s (TYP.)
Verify	4.5 s (TYP.)	8.5 s (TYP.)	8.5 s (TYP.)	8.5 s (TYP.)	8.5 s (TYP.)	8.5 s (TYP.)
E.P.V	9 s (TYP.)	12.5 s (TYP.)	12.5 s (TYP.)	12.5 s (TYP.)	12.5 s (TYP.)	12.5 s (TYP.)
Checksum	2 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)
Security	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)

Caution When executing boot swapping, do not use the E.P.V. command with the dedicated flash memory programmer.

<R>
28.10 Flash Memory Programming by Self-Programming

The 78K0/LF3 microcontrollers support a self-programming function that can be used to rewrite the flash memory via a user program. Because this function allows a user application to rewrite the flash memory by using the self-programming library, it can be used to upgrade the program in the field.

If an interrupt occurs during self-programming, self-programming can be temporarily stopped and interrupt servicing can be executed. To execute interrupt servicing, restore the normal operation mode after self-programming has been stopped, and execute the EI instruction. After the self-programming mode is later restored, self-programming can be resumed.

Cautions 1. The self-programming function cannot be used when the CPU operates with the subsystem clock.

2. Oscillation of the internal high-speed oscillator is started during self programming, regardless of the setting of the RSTOP flag (bit 0 of the internal oscillation mode register (RCM)). Oscillation of the internal high-speed oscillator cannot be stopped even if the STOP instruction is executed.

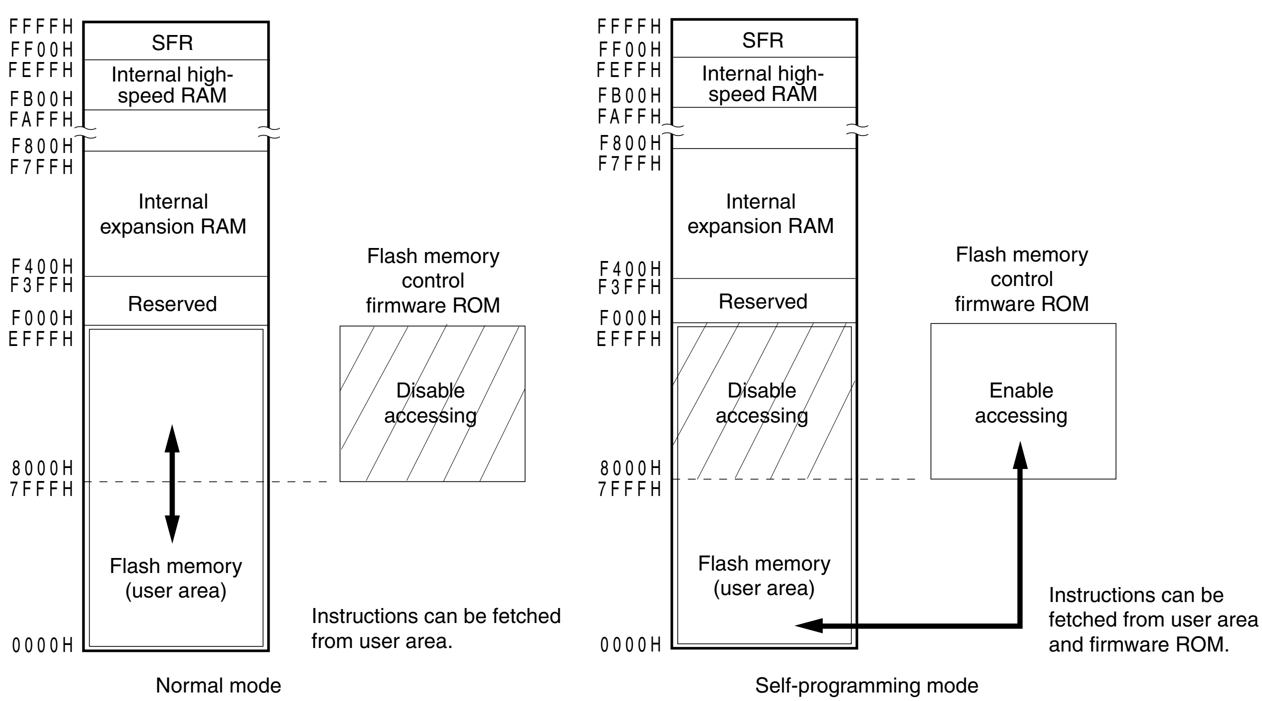
3. Input a high level to the FLMD0 pin during self-programming.

4. Be sure to execute the DI instruction before starting self-programming.
The self-programming function checks the interrupt request flags (IF0L, IF0H, IF1L, and IF1H). If an interrupt request is generated, self-programming is stopped.

5. Self-programming is also stopped by an interrupt request that is not masked even in the DI status. To prevent this, mask the interrupt by using the interrupt mask flag registers (MK0L, MK0H, MK1L, and MK1H).

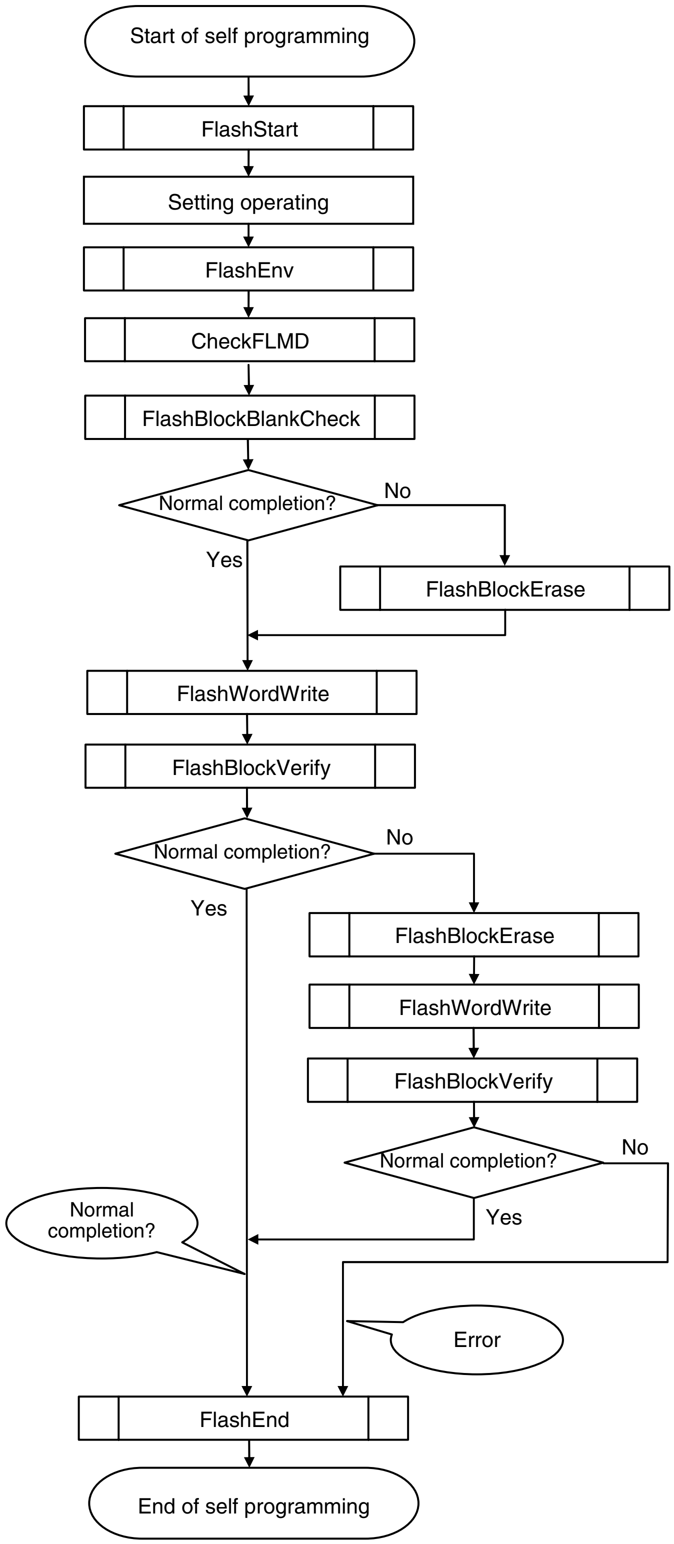
6. Allocate the entry program for self-programming in the 0000H to 7FFFH.

Figure 28-15. Operation Mode and Memory Map for Self-Programming (μPD78F0475)



The following figure illustrates a flow of rewriting the flash memory by using a self programming sample library.

Figure 28-16. Flow of Self Programming (Rewriting Flash Memory)



Remark For details of the self programming library, refer to **78K0 Microcontroller Self-Programming Library Type01 User's Manual (U18274E)**.

The following table shows the processing time and interrupt response time for the self-programming library.

Table 28-13. Processing Time and Interrupt Acknowledgment (1/4)
(When Normal Model Library and Entry RAM Are Allocated Outside Short Direct Addressing Range)

Function		Processing Time (Unit: μs)			Interrupt Acknowledgment
		RSTOP = 0 and RSTS = 1 (during stable operation of internal high-speed oscillator)		RSTOP = 1 (internal high-speed oscillator stopped)Note	
		MCS = 0 (internal high-speed oscillation clock)	MCS = 1 (high-speed system clock)	MCS = 1 (high-speed system clock)	
Self programming start function		34/fCPU	34/fCPU	34/fCPU	Disabled
Self programming end function		34/fCPU	34/fCPU	34/fCPU	Disabled
Initialize function		55/fCPU+1140	55/fCPU+1140	55/fCPU+1912	Disabled
Block erase function		179/fCPU+353193	179/fCPU+353193	179/fCPU+353965	Enabled
Word write function		333/fCPU+1154+ 2142×W	333/fCPU+1154+ 2142×W	333/fCPU+1927+ 2142×W	Enabled
Block verify function		179/fCPU+25596	179/fCPU+25596	179/fCPU+26369	Enabled
Block blank check function		179/fCPU+12805	179/fCPU+12805	179/fCPU+13578	Enabled
Get information function	Option value: 03H	180/fCPU+1065	180/fCPU+1065	180/fCPU+1838	Disabled
	Option value: 04H	190/fCPU+1056	190/fCPU+1056	190/fCPU+1829	Disabled
	Option value: 05H	350/fCPU+1041	350/fCPU+1041	350/fCPU+1813	Disabled
Set information function		80/fCPU+753218	80/fCPU+753218	80/fCPU+753990	Enabled
Mode check function		36/fCPU+952	36/fCPU+952	36/fCPU+1724	Disabled
EEPROM write function		333/fCPU+1297+ 2286×W	333/fCPU+1297+ 2286×W	333/fCPU+2069+ 2286×W	Enabled

Note This is the function processing time when the function is executed immediately after the self programming start function has been executed. The processing time after a function other than the self programming start function has been executed is the same as that of RSTOP = 0.

Remark RSTOP: Bit 0 of the internal oscillation mode register (RCM)
RSTS: Bit 7 of RCM
MCS: Bit 1 of main clock mode register (MCM)
fCPU: CPU clock frequency
W: Number of words to be written (1 word = 4 bytes)

Table 28-13. Processing Time and Interrupt Acknowledgment (2/4)
(When Normal Model Library and Entry RAM Are Allocated Within Short Direct Addressing Range)

Function		Processing Time (Unit: μs)			Interrupt Acknowledgment
		RSTOP = 0 and RSTS = 1 (during stable operation of internal high-speed oscillator)		RSTOP = 1 (internal high-speed oscillator stopped)Note	
		MCS = 0 (internal high-speed oscillation clock)	MCS = 1 (high-speed system clock)	MCS = 1 (high-speed system clock)	
Self programming start function		$34/f_{CPU}$	$34/f_{CPU}$	$34/f_{CPU}$	Disabled
Self programming end function		$34/f_{CPU}$	$34/f_{CPU}$	$34/f_{CPU}$	Disabled
Initialize function		$55/f_{CPU}+462$	$55/f_{CPU}+462$	$55/f_{CPU}+473$	Disabled
Block erase function		$179/f_{CPU}+352516$	$179/f_{CPU}+352516$	$179/f_{CPU}+352528$	Enabled
Word write function		$333/f_{CPU}+477+2142\times W$	$333/f_{CPU}+477+2142\times W$	$333/f_{CPU}+488+2142\times W$	Enabled
Block verify function		$179/f_{CPU}+24918$	$179/f_{CPU}+24918$	$179/f_{CPU}+24930$	Enabled
Block blank check function		$179/f_{CPU}+12128$	$179/f_{CPU}+12128$	$179/f_{CPU}+12139$	Enabled
Get information function	Option value: 03H	$180/f_{CPU}+388$	$180/f_{CPU}+388$	$180/f_{CPU}+399$	Disabled
	Option value: 04H	$190/f_{CPU}+378$	$190/f_{CPU}+378$	$190/f_{CPU}+390$	Disabled
	Option value: 05H	$350/f_{CPU}+363$	$350/f_{CPU}+363$	$350/f_{CPU}+375$	Disabled
Set information function		$80/f_{CPU}+752540$	$80/f_{CPU}+752540$	$80/f_{CPU}+753654$	Enabled
Mode check function		$36/f_{CPU}+274$	$36/f_{CPU}+274$	$36/f_{CPU}+286$	Disabled
EEPROM write function		$333/f_{CPU}+619+2286\times W$	$333/f_{CPU}+619+2286\times W$	$333/f_{CPU}+630+2286\times W$	Enabled

Note This is the function processing time when the function is executed immediately after the self programming start function has been executed. The processing time after a function other than the self programming start function has been executed is the same as that of RSTOP = 0.

Remark RSTOP: Bit 0 of the internal oscillation mode register (RCM)
RSTS: Bit 7 of RCM
MCS: Bit 1 of main clock mode register (MCM)
f_{CPU}: CPU clock frequency
W: Number of words to be written (1 word = 4 bytes)

Table 28-13. Processing Time and Interrupt Acknowledgment (3/4)
(When Static Model Library and Entry RAM Are Allocated Outside Short Direct Addressing Range)

Function		Processing Time (Unit: μs)			Interrupt Acknowledgment
		RSTOP = 0 and RSTS = 1 (during stable operation of internal high-speed oscillator)		RSTOP = 1 (internal high-speed oscillator stopped)Note	
		MCS = 0 (CPU operates with internal high-speed oscillation clock)	MCS = 1 (CPU operates with high-speed system clock)	MCS = 1 (CPU operates with high-speed system clock)	
Self programming start function		34/fCPU	34/fCPU	34/fCPU	Disabled
Self programming end function		34/fCPU	34/fCPU	34/fCPU	Disabled
Initialize function		55/fCPU+1140	55/fCPU+1140	55/fCPU+1912	Disabled
Block erase function		136/fCPU+353193	136/fCPU+353193	136/fCPU+353965	Enabled
Word write function		272/fCPU+1154+ 2142×W	272/fCPU+1154+ 2142×W	272/fCPU+1927+ 2142×W	Enabled
Block verify function		136/fCPU+25596	136/fCPU+25596	136/fCPU+26369	Enabled
Block blank check function		136/fCPU+12805	136/fCPU+12805	136/fCPU+13578	Enabled
Get information function	Option value: 03H	134/fCPU+1065	134/fCPU+1065	134/fCPU+1838	Disabled
	Option value: 04H	144/fCPU+1056	144/fCPU+1056	144/fCPU+1829	Disabled
	Option value: 05H	304/fCPU+1041	304/fCPU+1041	304/fCPU+1813	Disabled
Set information function		72/fCPU+753218	72/fCPU+753218	72/fCPU+753990	Enabled
Mode check function		30/fCPU+952	30/fCPU+952	30/fCPU+1724	Disabled
EEPROM write function		268/fCPU+1297+ 2286×W	268/fCPU+1297+ 2286×W	268/fCPU+2069+ 2286×W	Enabled

Note This is the function processing time when the function is executed immediately after the self programming start function has been executed. The processing time after a function other than the self programming start function has been executed is the same as that of RSTOP = 0.

Remark RSTOP: Bit 0 of the internal oscillation mode register (RCM)
RSTS: Bit 7 of RCM
MCS: Bit 1 of main clock mode register (MCM)
f_{CPU}: CPU clock frequency
W: Number of words to be written (1 word = 4 bytes)

Table 28-13. Processing Time and Interrupt Acknowledgment (4/4)
(When Static Model Library and Entry RAM Are Allocated Within Short Direct Addressing Range)

Function Name		Processing Time (Unit: μs)			Interrupt Acknowledgment
		RSTOP = 0 and RSTS = 1 (during stable operation of internal high-speed oscillator)		RSTOP = 1 (internal high-speed oscillator stopped)Note	
		MCS = 0 (CPU operates with internal high-speed oscillation clock)	MCS = 1 (CPU operates with high-speed system clock)	MCS = 1 (CPU operates with high-speed system clock)	
Self programming start function		34/f_{CPU}	34/f_{CPU}	34/f_{CPU}	Disabled
Self programming end function		34/f_{CPU}	34/f_{CPU}	34/f_{CPU}	Disabled
Initialize function		55/f_{CPU}+462	55/f_{CPU}+462	55/f_{CPU}+473	Disabled
Block erase function		136/f_{CPU}+352516	136/f_{CPU}+352516	136/f_{CPU}+352528	Enabled
Word write function		272/f_{CPU}+477+ 2142×W	272/f_{CPU}+477+ 2142×W	272/f_{CPU}+488+ 2142×W	Enabled
Block verify function		136/f_{CPU}+24918	136/f_{CPU}+24918	136/f_{CPU}+24930	Enabled
Block blank check function		136/f_{CPU}+12128	136/f_{CPU}+12128	136/f_{CPU}+12139	Enabled
Get information function	Option value: 03H	134/f_{CPU}+388	134/f_{CPU}+388	134/f_{CPU}+399	Disabled
	Option value: 04H	144/f_{CPU}+378	144/f_{CPU}+378	144/f_{CPU}+390	Disabled
	Option value: 05H	304/f_{CPU}+363	304/f_{CPU}+363	304/f_{CPU}+375	Disabled
Set information function		72/f_{CPU}+752540	72/f_{CPU}+752540	72/f_{CPU}+753654	Enabled
Mode check function		30/f_{CPU}+274	30/f_{CPU}+274	30/f_{CPU}+286	Disabled
EEPROM write function		268/f_{CPU}+619+ 2286×W	268/f_{CPU}+619+ 2286×W	268/f_{CPU}+630+ 2286×W	Enabled

Note This is the function processing time when the function is executed immediately after the self programming start function has been executed. The processing time after a function other than the self programming start function has been executed is the same as that of RSTOP = 0.

Remark RSTOP: Bit 0 of the internal oscillation mode register (RCM)
RSTS: Bit 7 of RCM
MCS: Bit 1 of main clock mode register (MCM)
f_{CPU}: CPU clock frequency
W: Number of words to be written (1 word = 4 bytes)

Table 28-14. Interrupt Response Time (Library for Normal Model) (1/2)

Function Name	Interrupt Response Time (Unit: μs)					
	When entry RAM is allocated outside short direct addressing range			When entry RAM is allocated within short direct addressing range		
	RSTOP = 0 and RSTS = 1 (during stable operation of internal high-speed oscillator)		RSTOP = 1 (internal high-speed oscillator stopped)[Note]	RSTOP = 0 and RSTS = 1 (during stable operation of internal high-speed oscillator)		RSTOP = 1 (internal high-speed oscillator stopped)[Note]
	MCS = 0 (CPU operates with internal high-speed oscillation clock)	MCS = 1 (CPU operates with high-speed system clock)	MCS = 1 (CPU operates with high-speed system clock)	MCS = 0 (CPU operates with internal high-speed oscillation clock)	MCS = 1 (CPU operates with high-speed system clock)	MCS = 1 (CPU operates with high-speed system clock)
Block erase function	$179/f_{CPU}+1269$	$179/f_{CPU}+1269$	$179/f_{CPU}+1912$	$179/f_{CPU}+703$	$179/f_{CPU}+703$	$179/f_{CPU}+713$
Word write function	$333/f_{CPU}+1098$	$333/f_{CPU}+1098$	$333/f_{CPU}+1742$	$333/f_{CPU}+533$	$333/f_{CPU}+533$	$333/f_{CPU}+543$
Block verify function	$179/f_{CPU}+1013$	$179/f_{CPU}+1013$	$179/f_{CPU}+1656$	$179/f_{CPU}+448$	$179/f_{CPU}+448$	$179/f_{CPU}+456$
Block blank check function	$179/f_{CPU}+993$	$179/f_{CPU}+993$	$179/f_{CPU}+1637$	$179/f_{CPU}+428$	$179/f_{CPU}+428$	$179/f_{CPU}+438$
Set information function	$80/f_{CPU}+833$	$80/f_{CPU}+833$	$80/f_{CPU}+1477$	$80/f_{CPU}+346$	$80/f_{CPU}+346$	$80/f_{CPU}+346$
EEPROM write function	$333/f_{CPU}+1107$	$333/f_{CPU}+1107$	$333/f_{CPU}+1751$	$333/f_{CPU}+542$	$333/f_{CPU}+542$	$333/f_{CPU}+552$

Note This is the function processing time when the function is executed immediately after the self programming start function has been executed. The processing time after a function other than the self programming start function has been executed is the same as that of RSTOP = 0.

Remark RSTOP: Bit 0 of the internal oscillation mode register (RCM)
RSTS: Bit 7 of RCM
MCS: Bit 1 of main clock mode register (MCM)
f_{CPU}: CPU clock frequency
W: Number of words to be written (1 word = 4 bytes)

Table 28-14. Interrupt Response Time (Library for Static Model) (2/2)

Function Name	Interrupt Response Time (μs)					
	When entry RAM is allocated outside short direct addressing range			When entry RAM is allocated within short direct addressing range		
	RSTOP = 0 and RSTS = 1 (during stable operation of internal high-speed oscillator)		RSTOP = 1 (internal high-speed oscillator stopped)[Note]	RSTOP = 0 and RSTS = 1 (during stable operation of internal high-speed oscillator)		RSTOP = 1 (internal high-speed oscillator stopped)[Note]
	MCS = 0 (CPU operates with internal high-speed oscillation clock)	MCS = 1 (CPU operates with high-speed system clock)	MCS = 1 (CPU operates with high-speed system clock)	MCS = 0 (CPU operates with internal high-speed oscillation clock)	MCS = 1 (CPU operates with high-speed system clock)	MCS = 1 (CPU operates with high-speed system clock)
Block erase function	136/f_{CPU}+1269	136/f_{CPU}+1269	136/f_{CPU}+1912	136/f_{CPU}+703	136/f_{CPU}+703	136/f_{CPU}+713
Word write function	272/f_{CPU}+1098	272/f_{CPU}+1098	272/f_{CPU}+1742	272/f_{CPU}+533	272/f_{CPU}+533	272/f_{CPU}+543
Block verify function	136/f_{CPU}+1013	136/f_{CPU}+1013	136/f_{CPU}+1656	136/f_{CPU}+448	136/f_{CPU}+448	136/f_{CPU}+456
Block blank check function	136/f_{CPU}+993	136/f_{CPU}+993	136/f_{CPU}+1637	136/f_{CPU}+428	136/f_{CPU}+428	136/f_{CPU}+438
Set information function	72/f_{CPU}+833	72/f_{CPU}+833	72/f_{CPU}+1477	72/f_{CPU}+346	72/f_{CPU}+346	72/f_{CPU}+346
EEPROM write function	268/f_{CPU}+1107	268/f_{CPU}+1107	268/f_{CPU}+1751	268/f_{CPU}+542	268/f_{CPU}+542	268/f_{CPU}+552

Note This is the function processing time when the function is executed immediately after the self programming start function has been executed. The processing time after a function other than the self programming start function has been executed is the same as that of RSTOP = 0.

Remark RSTOP: Bit 0 of the internal oscillation mode register (RCM)
RSTS: Bit 7 of RCM
MCS: Bit 1 of main clock mode register (MCM)
f_{CPU}: CPU clock frequency
W: Number of words to be written (1 word = 4 bytes)

28.10.1 Boot swap function

If rewriting the boot area has failed during self-programming due to a power failure or some other cause, the data in the boot area may be lost and the program may not be restarted by resetting.

The boot swap function is used to avoid this problem.

Before erasing boot cluster 0[Note], which is a boot program area, by self-programming, write a new boot program to boot cluster 1 in advance. When the program has been correctly written to boot cluster 1, swap this boot cluster 1 and boot cluster 0 by using the set information function of the firmware of the 78K0/LF3, so that boot cluster 1 is used as a boot area. After that, erase or write the original boot program area, boot cluster 0.

As a result, even if a power failure occurs while the boot programming area is being rewritten, the program is executed correctly because it is booted from boot cluster 1 to be swapped when the program is reset and started next.

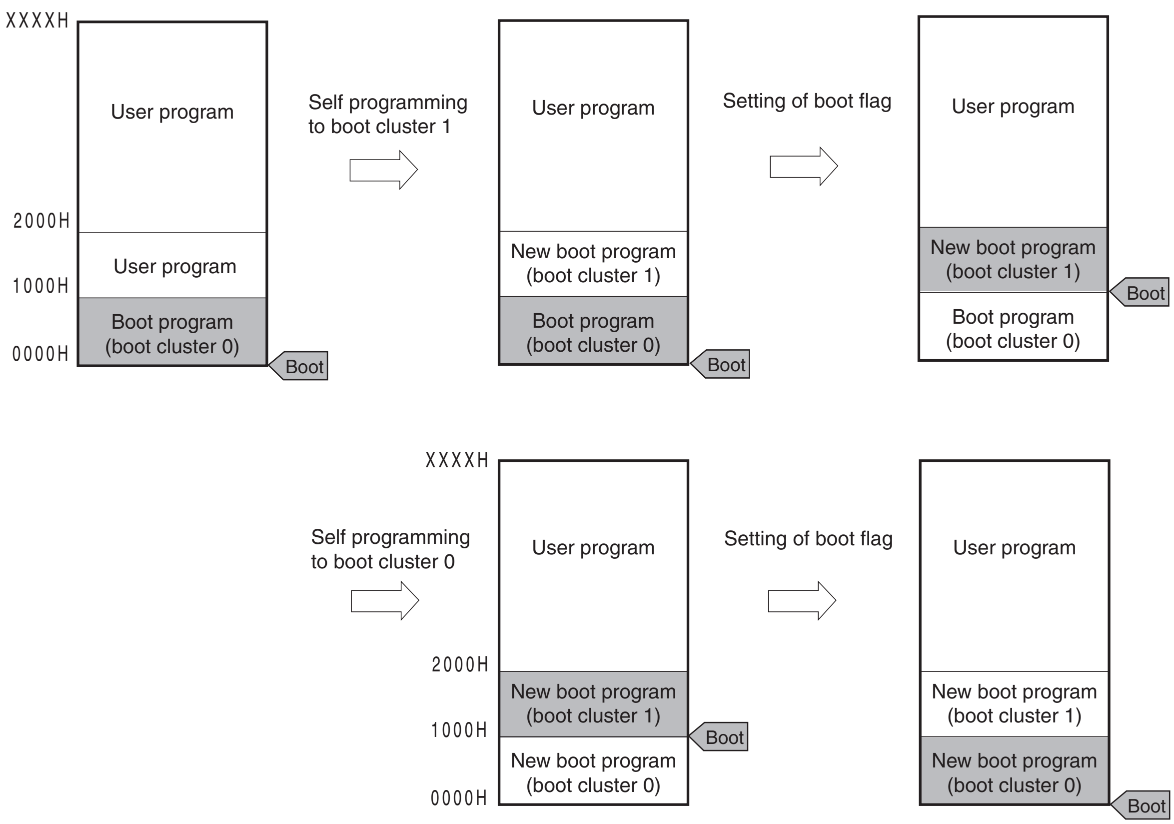
If the program has been correctly written to boot cluster 0, restore the original boot area by using the set information function of the firmware of the 78K0/LF3.

Note A boot cluster is a 4 KB area and boot clusters 0 and 1 are swapped by the boot swap function.

Boot cluster 0 (0000H to 0FFFH): Original boot program area

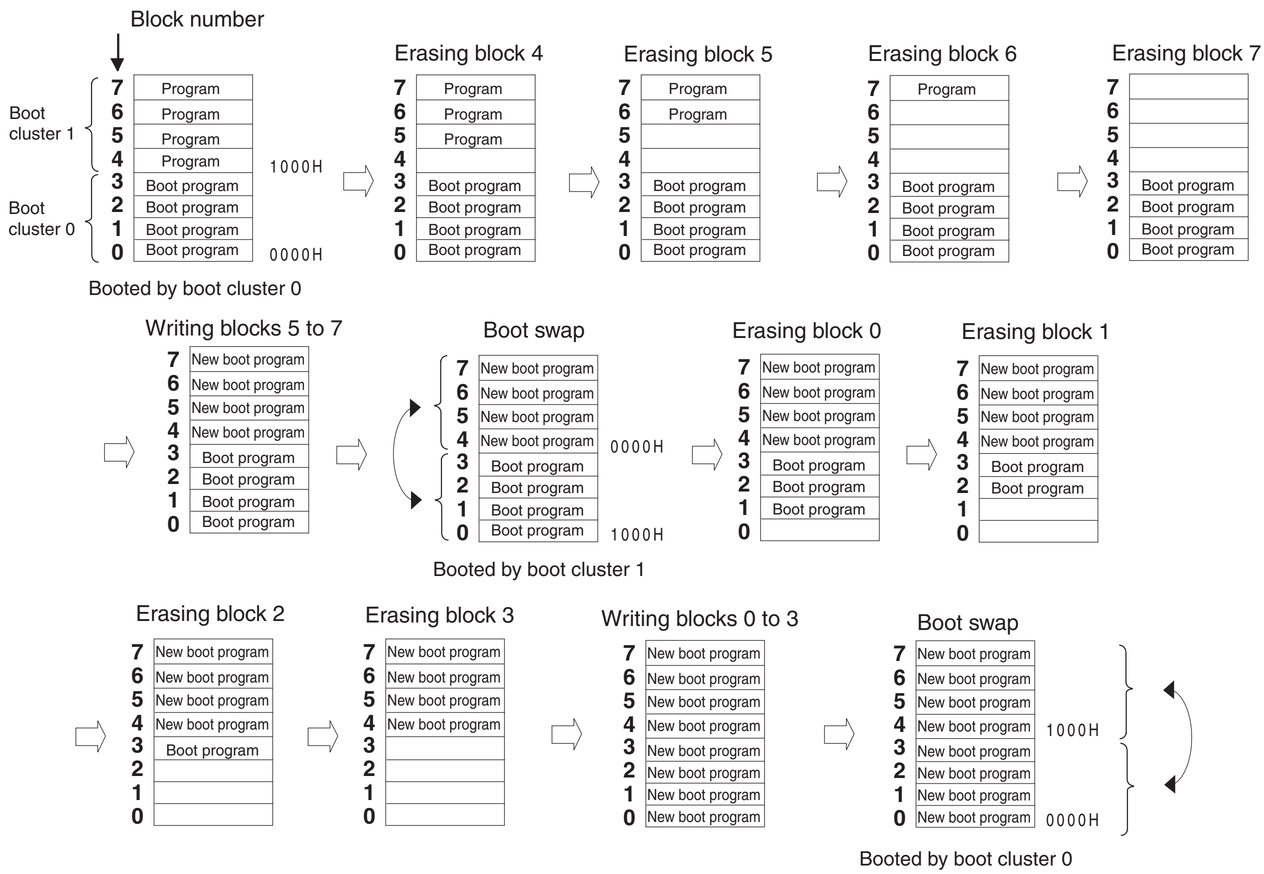
Boot cluster 1 (1000H to 1FFFH): Area subject to boot swap function

Figure 28-17. Boot Swap Function



Remark Boot cluster 1 becomes 0000H to 0FFFH when a reset is generated after the boot flag has been set.

Figure 28-18. Example of Executing Boot Swapping

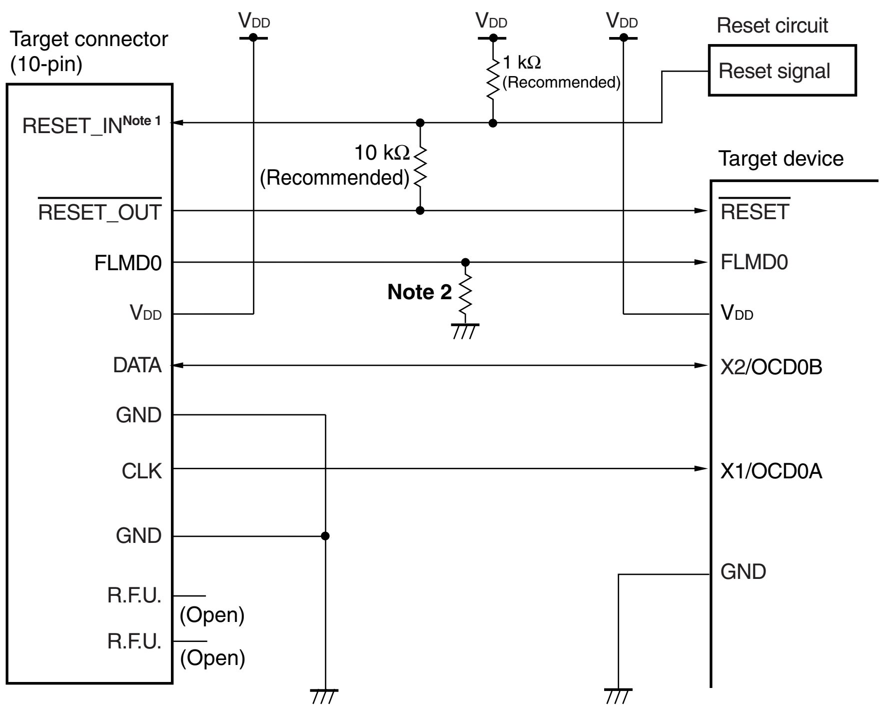


29.1 Connecting QB-MINI2 to 78K0/LF3

The 78K0/LF3 uses the V_{DD}, FLMD0, $\overline{\text{RESET}}$, OCD0A/X1, OCD0B/X2, and V_{SS} pins to communicate with the host machine via an on-chip debug emulator (QB-MINI2).

Caution The 78K0/LF3 has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. NEC Electronics is not liable for problems occurring when the on-chip debug function is used.

Figure 29-1. Connection Example of QB-MINI2 and 78K0/LF3



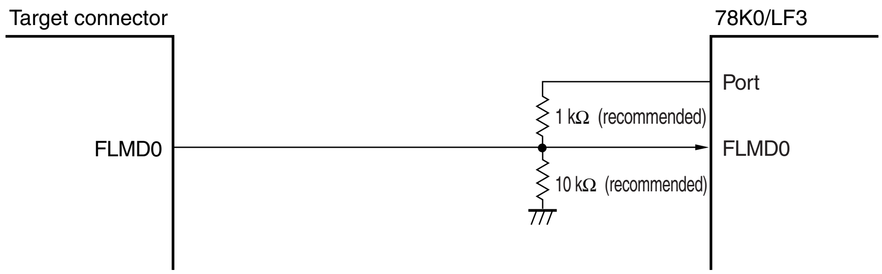
Notes 1. This connection is designed assuming that the reset signal is output from the N-ch open-drain buffer (output resistance: 100 Ω or less). For details, refer to **QB-MINI2 User's Manual (U18371E)**.

2. Make pull-down resistor 470 Ω or more (10 kΩ: recommended).

Caution Input the clock from the OCD0A/X1 pin during on-chip debugging.

Connect the FLMD0 pin as follows when performing self programming by means of on-chip debugging.

Figure 29-2. Connection of FLMD0 Pin for Self Programming by Means of On-Chip Debugging



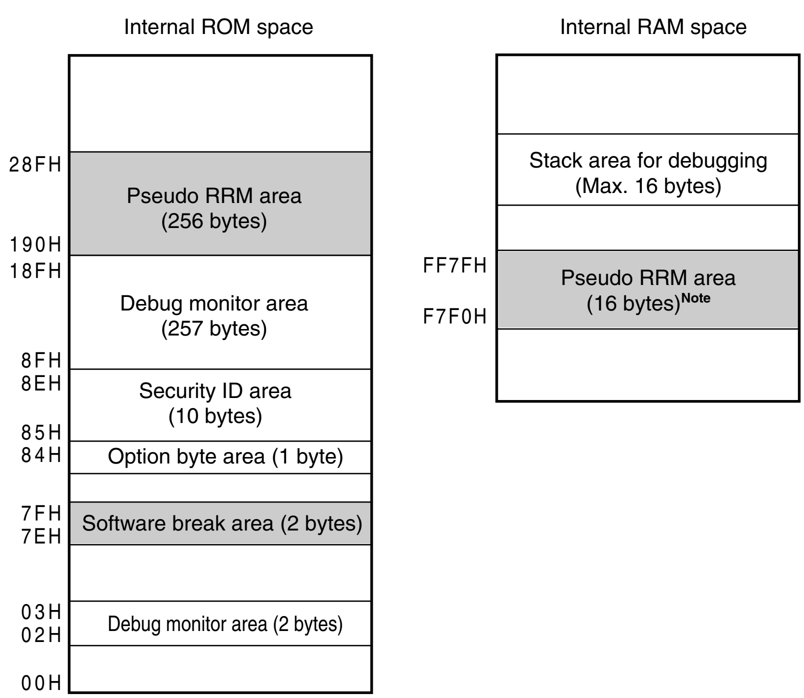
29.2 Reserved Area Used by QB-MINI2

QB-MINI2 uses the reserved areas shown in Figure 29-3 below to implement communication with the 78K0/LF3, or each debug function. The shaded reserved areas are used for the respective debug functions to be used, and the other areas are always used for debugging. These reserved areas can be secured by using user programs and compiler options.

When using a boot swap operation during self programming, set the same value to boot cluster 1 beforehand.

For details on reserved area, refer to **QB-MINI2 User's Manual (U18371E)**.

Figure 29-3. Reserved Area Used by QB-MINI2



Remark Shaded reserved areas: Area used for the respective debug functions to be used
Other reserved areas: Areas always used for debugging

This chapter lists each instruction set of the 78K0/LF3 in table form. For details of each operation and operation code, refer to the separate document **78K/0 Series Instructions User's Manual (U12326E)**.

30.1 Conventions Used in Operation List

30.1.1 Operand identifiers and specification methods

Operands are written in the "Operand" column of each instruction in accordance with the specification method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more methods, select one of them. Uppercase letters and the symbols #, !, $ and [ ] are keywords and must be written as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: Absolute address specification
- $: Relative address specification
- [ ]: Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to write the #, !, $, and [ ] symbols.

For operand register identifiers r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for specification.

Table 30-1. Operand Identifiers and Specification Methods

Identifier	Specification Method
r rp sfr sfrp	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[Note] Special function register symbol (16-bit manipulatable register even addresses only)[Note]
saddr saddrp	FE20H to FF1FH Immediate data or labels FE20H to FF1FH Immediate data or labels (even address only)
addr16 addr11 addr5	0000H to FFFFH Immediate data or labels (Only even addresses for 16-bit data transfer instructions) 0800H to 0FFFH Immediate data or labels 0040H to 007FH Immediate data or labels (even address only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label
RBn	RB0 to RB3

Note Addresses from FFD0H to FFDFH cannot be accessed with these operands.

Remark For special function register symbols, see **Table 3-6 Special Function Register List**.

30.1.2 Description of operation column

A:	A register; 8-bit accumulator
X:	X register
B:	B register
C:	C register
D:	D register
E:	E register
H:	H register
L:	L register
AX:	AX register pair; 16-bit accumulator
BC:	BC register pair
DE:	DE register pair
HL:	HL register pair
PC:	Program counter
SP:	Stack pointer
PSW:	Program status word
CY:	Carry flag
AC:	Auxiliary carry flag
Z:	Zero flag
RBS:	Register bank select flag
IE:	Interrupt request enable flag
( ):	Memory contents indicated by address or register contents in parentheses
X_H, X_L:	Higher 8 bits and lower 8 bits of 16-bit register
$\wedge$:	Logical product (AND)
$\vee$:	Logical sum (OR)
$\veebar$:	Exclusive logical sum (exclusive OR)
$\overline{}$:	Inverted data
addr16:	16-bit immediate data or label
jdisp8:	Signed 8-bit data (displacement value)

30.1.3 Description of flag operation column

(Blank):	Not affected
0:	Cleared to 0
1:	Set to 1
$\times$:	Set/cleared according to the result
R:	Previously saved value is restored

30.2 Operation List

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
8-bit data transfer	**MOV**	r, #byte	2	4	–	r ← byte			
		saddr, #byte	3	6	7	(saddr) ← byte			
		sfr, #byte	3	–	7	sfr ← byte			
		A, r Note 3	1	2	–	A ← r			
		r, A Note 3	1	2	–	r ← A			
		A, saddr	2	4	5	A ← (saddr)			
		saddr, A	2	4	5	(saddr) ← A			
		A, sfr	2	–	5	A ← sfr			
		sfr, A	2	–	5	sfr ← A			
		A, !addr16	3	8	9	A ← (addr16)			
		!addr16, A	3	8	9	(addr16) ← A			
		PSW, #byte	3	–	7	PSW ← byte	×	×	×
		A, PSW	2	–	5	A ← PSW			
		PSW, A	2	–	5	PSW ← A	×	×	×
		A, [DE]	1	4	5	A ← (DE)			
		[DE], A	1	4	5	(DE) ← A			
		A, [HL]	1	4	5	A ← (HL)			
		[HL], A	1	4	5	(HL) ← A			
		A, [HL + byte]	2	8	9	A ← (HL + byte)			
		[HL + byte], A	2	8	9	(HL + byte) ← A			
		A, [HL + B]	1	6	7	A ← (HL + B)			
		[HL + B], A	1	6	7	(HL + B) ← A			
		A, [HL + C]	1	6	7	A ← (HL + C)			
		[HL + C], A	1	6	7	(HL + C) ← A			
	XCH	A, r Note 3	1	2	–	A ↔ r			
		A, saddr	2	4	6	A ↔ (saddr)			
		A, sfr	2	–	6	A ↔ (sfr)			
		A, !addr16	3	8	10	A ↔ (addr16)			
		A, [DE]	1	4	6	A ↔ (DE)			
		A, [HL]	1	4	6	A ↔ (HL)			
		A, [HL + byte]	2	8	10	A ↔ (HL + byte)			
		A, [HL + B]	2	8	10	A ↔ (HL + B)			
		A, [HL + C]	2	8	10	A ↔ (HL + C)			

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	**MOVW**	rp, #word	3	6	–	rp ← word			
		saddrp, #word	4	8	10	(saddrp) ← word			
		sfrp, #word	4	–	10	sfrp ← word			
		AX, saddrp	2	6	8	AX ← (saddrp)			
		saddrp, AX	2	6	8	(saddrp) ← AX			
		AX, sfrp	2	–	8	AX ← sfrp			
		sfrp, AX	2	–	8	sfrp ← AX			
		AX, rp Note 3	1	4	–	AX ← rp			
		rp, AX Note 3	1	4	–	rp ← AX			
		AX, !addr16	3	10	12	AX ← (addr16)			
		!addr16, AX	3	10	12	(addr16) ← AX			
	XCHW	AX, rp Note 3	1	4	–	AX ↔ rp			
8-bit operation	**ADD**	A, #byte	2	4	–	A, CY ← A + byte	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte	×	×	×
		A, r Note 4	2	4	–	A, CY ← A + r	×	×	×
		r, A	2	4	–	r, CY ← r + A	×	×	×
		A, saddr	2	4	5	A, CY ← A + (saddr)	×	×	×
		A, !addr16	3	8	9	A, CY ← A + (addr16)	×	×	×
		A, [HL]	1	4	5	A, CY ← A + (HL)	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte)	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B)	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A + (HL + C)	×	×	×
	ADDC	A, #byte	2	4	–	A, CY ← A + byte + CY	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte + CY	×	×	×
		A, r Note 4	2	4	–	A, CY ← A + r + CY	×	×	×
		r, A	2	4	–	r, CY ← r + A + CY	×	×	×
		A, saddr	2	4	5	A, CY ← A + (saddr) + CY	×	×	×
		A, !addr16	3	8	9	A, CY ← A + (addr16) + C	×	×	×
		A, [HL]	1	4	5	A, CY ← A + (HL) + CY	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte) + CY	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B) + CY	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A + (HL + C) + CY	×	×	×

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Only when rp = BC, DE or HL

4. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	**SUB**	A, #byte	2	4	–	A, CY ← A – byte	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte	×	×	×
		A, r Note 3	2	4	–	A, CY ← A – r	×	×	×
		r, A	2	4	–	r, CY ← r – A	×	×	×
		A, saddr	2	4	5	A, CY ← A – (saddr)	×	×	×
		A, !addr16	3	8	9	A, CY ← A – (addr16)	×	×	×
		A, [HL]	1	4	5	A, CY ← A – (HL)	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A – (HL + byte)	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A – (HL + B)	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A – (HL + C)	×	×	×
	SUBC	A, #byte	2	4	–	A, CY ← A – byte – CY	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte – CY	×	×	×
		A, r Note 3	2	4	–	A, CY ← A – r – CY	×	×	×
		r, A	2	4	–	r, CY ← r – A – CY	×	×	×
		A, saddr	2	4	5	A, CY ← A – (saddr) – CY	×	×	×
		A, !addr16	3	8	9	A, CY ← A – (addr16) – CY	×	×	×
		A, [HL]	1	4	5	A, CY ← A – (HL) – CY	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A – (HL + byte) – CY	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A – (HL + B) – CY	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A – (HL + C) – CY	×	×	×
	AND	A, #byte	2	4	–	A ← A ∧ byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∧ byte	×		
		A, r Note 3	2	4	–	A ← A ∧ r	×		
		r, A	2	4	–	r ← r ∧ A	×		
		A, saddr	2	4	5	A ← A ∧ (saddr)	×		
		A, !addr16	3	8	9	A ← A ∧ (addr16)	×		
		A, [HL]	1	4	5	A ← A ∧ (HL)	×		
		A, [HL + byte]	2	8	9	A ← A ∧ (HL + byte)	×		
		A, [HL + B]	2	8	9	A ← A ∧ (HL + B)	×		
		A, [HL + C]	2	8	9	A ← A ∧ (HL + C)	×		

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
8-bit operation	**OR**	A, #byte	2	4	–	A ← A ∨ byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∨ byte	×		
		A, r Note 3	2	4	–	A ← A ∨ r	×		
		r, A	2	4	–	r ← r ∨ A	×		
		A, saddr	2	4	5	A ← A ∨ (saddr)	×		
		A, !addr16	3	8	9	A ← A ∨ (addr16)	×		
		A, [HL]	1	4	5	A ← A ∨ (HL)	×		
		A, [HL + byte]	2	8	9	A ← A ∨ (HL + byte)	×		
		A, [HL + B]	2	8	9	A ← A ∨ (HL + B)	×		
		A, [HL + C]	2	8	9	A ← A ∨ (HL + C)	×		
	XOR	A, #byte	2	4	–	A ← A ⊻ byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr) ⊻ byte	×		
		A, r Note 3	2	4	–	A ← A ⊻ r	×		
		r, A	2	4	–	r ← r ⊻ A	×		
		A, saddr	2	4	5	A ← A ⊻ (saddr)	×		
		A, !addr16	3	8	9	A ← A ⊻ (addr16)	×		
		A, [HL]	1	4	5	A ← A ⊻ (HL)	×		
		A, [HL + byte]	2	8	9	A ← A ⊻ (HL + byte)	×		
		A, [HL + B]	2	8	9	A ← A ⊻ (HL + B)	×		
		A, [HL + C]	2	8	9	A ← A ⊻ (HL + C)	×		
	CMP	A, #byte	2	4	–	A – byte	×	×	×
		saddr, #byte	3	6	8	(saddr) – byte	×	×	×
		A, r Note 3	2	4	–	A – r	×	×	×
		r, A	2	4	–	r – A	×	×	×
		A, saddr	2	4	5	A – (saddr)	×	×	×
		A, !addr16	3	8	9	A – (addr16)	×	×	×
		A, [HL]	1	4	5	A – (HL)	×	×	×
		A, [HL + byte]	2	8	9	A – (HL + byte)	×	×	×
		A, [HL + B]	2	8	9	A – (HL + B)	×	×	×
		A, [HL + C]	2	8	9	A – (HL + C)	×	×	×

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	**ADDW**	AX, #word	3	6	–	AX, CY ← AX + word	×	×	×
	SUBW	AX, #word	3	6	–	AX, CY ← AX – word	×	×	×
	CMPW	AX, #word	3	6	–	AX – word	×	×	×
Multiply/ divide	**MULU**	X	2	16	–	AX ← A × X			
	DIVUW	C	2	25	–	AX (Quotient), C (Remainder) ← AX ÷ C			
Increment/ decrement	**INC**	r	1	2	–	r ← r + 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) + 1	×	×	
	DEC	r	1	2	–	r ← r – 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) – 1	×	×	
	INCW	rp	1	4	–	rp ← rp + 1			
	DECW	rp	1	4	–	rp ← rp – 1			
Rotate	**ROR**	A, 1	1	2	–	(CY, A_7 ← A_0, A_{m-1} ← A_m) × 1 time			×
	ROL	A, 1	1	2	–	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1 time			×
	RORC	A, 1	1	2	–	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1 time			×
	ROLC	A, 1	1	2	–	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1 time			×
	ROR4	[HL]	2	10	12	A_{3-0} ← $(HL)_{3-0}$, $(HL)_{7-4}$ ← A_{3-0}, $(HL)_{3-0}$ ← $(HL)_{7-4}$			
	ROL4	[HL]	2	10	12	A_{3-0} ← $(HL)_{7-4}$, $(HL)_{3-0}$ ← A_{3-0}, $(HL)_{7-4}$ ← $(HL)_{3-0}$			
BCD adjustment	**ADJBA**		2	4	–	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		2	4	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit manipulate	**MOV1**	CY, saddr.bit	3	6	7	CY ← (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← sfr.bit			×
		CY, A.bit	2	4	–	CY ← A.bit			×
		CY, PSW.bit	3	–	7	CY ← PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← (HL).bit			×
		saddr.bit, CY	3	6	8	(saddr.bit) ← CY			
		sfr.bit, CY	3	–	8	sfr.bit ← CY			
		A.bit, CY	2	4	–	A.bit ← CY			
		PSW.bit, CY	3	–	8	PSW.bit ← CY	×	×	
		[HL].bit, CY	2	6	8	(HL).bit ← CY			

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Bit manipulate	**AND1**	CY, saddr.bit	3	6	7	CY ← CY ∧ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY ∧ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY ∧ A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY ∧ PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY ∧ (HL).bit			×
	OR1	CY, saddr.bit	3	6	7	CY ← CY ∨ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY ∨ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY ∨ A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY ∨ PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY ∨ (HL).bit			×
	XOR1	CY, saddr.bit	3	6	7	CY ← CY ⊻ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY ⊻ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY ⊻ A.bit			×
		CY, PSW. bit	3	–	7	CY ← CY ⊻ PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY ⊻ (HL).bit			×
	SET1	saddr.bit	2	4	6	(saddr.bit) ← 1			
		sfr.bit	3	–	8	sfr.bit ← 1			
		A.bit	2	4	–	A.bit ← 1			
		PSW.bit	2	–	6	PSW.bit ← 1	×	×	×
		[HL].bit	2	6	8	(HL).bit ← 1			
	CLR1	saddr.bit	2	4	6	(saddr.bit) ← 0			
		sfr.bit	3	–	8	sfr.bit ← 0			
		A.bit	2	4	–	A.bit ← 0			
		PSW.bit	2	–	6	PSW.bit ← 0	×	×	×
		[HL].bit	2	6	8	(HL).bit ← 0			
	SET1	CY	1	2	–	CY ← 1			1
	CLR1	CY	1	2	–	CY ← 0			0
	NOT1	CY	1	2	–	CY ← $\overline{\text{CY}}$			×

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Call/return	**CALL**	!addr16	3	7	–	(SP − 1) ← $(PC + 3)_H$, (SP − 2) ← $(PC + 3)_L$, PC ← addr16, SP ← SP − 2			
	CALLF	!addr11	2	5	–	(SP − 1) ← $(PC + 2)_H$, (SP − 2) ← $(PC + 2)_L$, PC_{15-11} ← 00001, PC_{10-0} ← addr11, SP ← SP − 2			
	CALLT	[addr5]	1	6	–	(SP − 1) ← $(PC + 1)_H$, (SP − 2) ← $(PC + 1)_L$, PC_H ← (addr5 + 1), PC_L ← (addr5), SP ← SP − 2			
	BRK		1	6	–	(SP − 1) ← PSW, (SP − 2) ← $(PC + 1)_H$, (SP − 3) ← $(PC + 1)_L$, PC_H ← (003FH), PC_L ← (003EH), SP ← SP − 3, IE ← 0			
	RET		1	6	–	PC_H ← (SP + 1), PC_L ← (SP), SP ← SP + 2			
	RETI		1	6	–	PC_H ← (SP + 1), PC_L ← (SP), PSW ← (SP + 2), SP ← SP + 3	R	R	R
	RETB		1	6	–	PC_H ← (SP + 1), PC_L ← (SP), PSW ← (SP + 2), SP ← SP + 3	R	R	R
Stack manipulate	**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	2	–	PSW ← (SP), SP ← SP + 1	R	R	R
		rp	1	4	–	rp_H ← (SP + 1), rp_L ← (SP), SP ← SP + 2			
	MOVW	SP, #word	4	–	10	SP ← word			
		SP, AX	2	–	8	SP ← AX			
		AX, SP	2	–	8	AX ← SP			
Unconditional branch	**BR**	!addr16	3	6	–	PC ← addr16			
		$addr16	2	6	–	PC ← PC + 2 + jdisp8			
		AX	2	8	–	PCH ← A, PC_L ← X			
Conditional branch	**BC**	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 0			

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
Conditional branch	**BT**	saddr.bit, $addr16	3	8	9	PC ← PC + 3 + jdisp8 if (saddr.bit) = 1	
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 1	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1	
		PSW.bit, $addr16	3	–	9	PC ← PC + 3 + jdisp8 if PSW.bit = 1	
		[HL].bit, $addr16	3	10	11	PC ← PC + 3 + jdisp8 if (HL).bit = 1	
	BF	saddr.bit, $addr16	4	10	11	PC ← PC + 4 + jdisp8 if (saddr.bit) = 0	
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 0	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 0	
		PSW.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if PSW. bit = 0	
		[HL].bit, $addr16	3	10	11	PC ← PC + 3 + jdisp8 if (HL).bit = 0	
	BTCLR	saddr.bit, $addr16	4	10	12	PC ← PC + 4 + jdisp8 if (saddr.bit) = 1 then reset (saddr.bit)	
		sfr.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit	
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit	
		PSW.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	× × ×
		[HL].bit, $addr16	3	10	12	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit	
	DBNZ	B, $addr16	2	6	–	B ← B – 1, then PC ← PC + 2 + jdisp8 if B ≠ 0	
		C, $addr16	2	6	–	C ← C –1, then PC ← PC + 2 + jdisp8 if C ≠ 0	
		saddr, $addr16	3	8	10	(saddr) ← (saddr) – 1, then PC ← PC + 3 + jdisp8 if (saddr) ≠ 0	
CPU control	**SEL**	RBn	2	4	–	RBS1, 0 ← n	
	NOP		1	2	–	No Operation	
	EI		2	–	6	IE ← 1 (Enable Interrupt)	
	DI		2	–	6	IE ← 0 (Disable Interrupt)	
	HALT		2	6	–	Set HALT Mode	
	STOP		2	6	–	Set STOP Mode	

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

30.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4, ROL4, PUSH, POP, DBNZ

Second Operand First Operand	#byte	A	r[Note]	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL + byte] [HL + B] [HL + C]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV ADD ADDC SUB SUBC AND OR XOR CMP											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											ROR4 ROL4
[HL + byte] [HL + B] [HL + C]		MOV											
X													MULU
C													DIVUW

Note Except “r = A”

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

Second Operand / First Operand	#word	AX	rp[Note]	sfrp	saddrp	!addr16	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW[Note]						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
!addr16		MOVW						
SP	MOVW	MOVW						

Note Only when rp = BC, DE, HL

(3) Bit manipulation instructions

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

Second Operand / First Operand	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	$addr16	None
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) Call instructions/branch instructions

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

Second Operand / First Operand	AX	!addr16	!addr11	[addr5]	$addr16
Basic instruction	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
Compound instruction					BT BF BTCLR DBNZ

(5) Other instructions

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

<R> **Caution The 78K0/LF3 has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. NEC Electronics is not liable for problems occurring when the on-chip debug function is used.**

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}		−0.5 to +6.5	V
	V_{SS}		−0.5 to +0.3	V
	AV_{REF}[Note 2]		−0.5 to V_{DD} + 0.3[Note 1]	V
	AV_{SS}[Note 2]		−0.5 to +0.3	V
REGC pin input voltage	V_{IREGC}		−0.5 to + 3.6 and −0.5 to V_{DD}	V
Input voltage	V_I	P10 to P17, P20 to P27, P30 to P34, P40 to P47, P80 to P83, P90 to P93, P100 to P103, P110 to P113, P120 to P124, P130 to P133, P140 to P143, P150 to P153, X1, X2, XT1, XT2, FLMD0, $\overline{\text{RESET}}$	−0.3 to V_{DD} + 0.3[Note 1]	V
Output voltage	V_O		−0.3 to V_{DD} + 0.3[Note 1]	V
Analog input voltage	V_{AN}	ANI0 to ANI7[Note 2], DS0− to DS2−[Note 3], DS0+ to DS2+[Note 3]	−0.3 to AV_{REF} + 0.3[Note 1] and −0.3 to V_{DD} + 0.3[Note 1]	V
	REF+[Note 3]		−0.5 to AV_{REF} + 0.3[Note 1]	V
	REF−[Note 3]		−0.5 to + 0.3	V

Notes 1. Must be 6.5 V or lower.

2. μPD78F048x and 78F049x only.

3. μPD78F049x only.

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.

Standard products

Absolute Maximum Ratings (T_A = 25°C) (2/2)

Parameter	Symbol	Conditions		Ratings	Unit
Output current, high	I_{OH1}	Per pin	P10 to P17, P30 to P34, P40 to P47, P80 to P83, P90 to P93, P100 to P103, P110 to P113, P120, P130 to P133, P140 to P143, P150 to P153	−10	mA
		Total of all pins −35 mA	P10 to P17, P30 to P34, P40 to P47, P120	−25	mA
			P80 to P83, P90 to P93, P100 to P103, P110 to P113, P130 to P133, P140 to P143, P150 to P153	−10	mA
	I_{OH2}	Per pin	P20 to P27	−0.5	mA
		Total of all pins		−2	mA
Output current, low	I_{OL}	Per pin	P10 to P17, P30 to P34, P40 to P47, P80 to P83, P90 to P93, P100 to P103, P110 to P113, P120, P130 to P133, P140 to P143, P150 to P153	30	mA
		Total of all pins 80 mA	P10 to P17, P30 to P34, P40 to P47, P120	40	mA
			P80 to P83, P90 to P93, P100 to P103, P110 to P113, P130 to P133, P140 to P143, P150 to P153	40	mA
		Per pin	P20 to P27	1	mA
		Total of all pins		5	mA
Operating ambient temperature	T_A			−40 to +85	°C
Storage temperature	T_{stg}			−65 to +150	°C

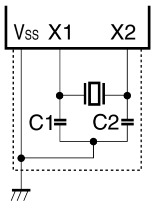
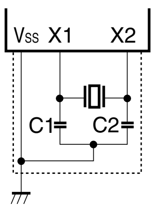
Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Standard products

X1 Oscillator Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
		X1 clock oscillation frequency (f_X)[Note]	2.7 V ≤ V_{DD} ≤ 5.5 V	2.0		10.0	MHz
			1.8 V ≤ V_{DD} < 2.7 V	2.0		5.0	
Crystal resonator		X1 clock oscillation frequency (f_X)[Note]	2.7 V ≤ V_{DD} ≤ 5.5 V	2.0		10.0	MHz
			1.8 V ≤ V_{DD} < 2.7 V	2.0		5.0	

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Cautions 1. When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

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

2. Since the CPU is started by the internal high-speed oscillation clock after a reset release, check the X1 clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) by the user. Determine the oscillation stabilization time of the OSTC register and oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.

Standard products

Internal Oscillator Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Resonator	Parameter	Conditions		MIN.	TYP.	MAX.	Unit
8 MHz internal oscillator	Internal high-speed oscillation clock frequency (f_{RH})Notes 1, 2	RSTS = 1	2.5 V ≤ V_{DD} ≤ 5.5 V	7.6	8.0	8.4	MHz
			1.8 V ≤ V_{DD} < 2.5 V	6.75	8.0	8.4	MHz
		RSTS = 0		2.48	5.6	9.86	MHz
240 kHz internal oscillator	Internal low-speed oscillation clock frequency (f_{RL})	2.6 V ≤ V_{DD} ≤ 5.5 V		216	240	264	kHz
		1.8 V ≤ V_{DD} < 2.6 V		192	240	264	kHz

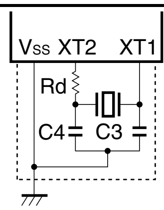
Notes 1. Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

2. When setting HIOTRM = 10H (±0%: default)

Remark RSTS: Bit 7 of the internal oscillation mode register (RCM)

XT1 Oscillator Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		XT1 clock oscillation frequency (f_{XT})Note		32	32.768	35	kHz

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Cautions 1. When using the XT1 oscillator, wire as follows in the area enclosed by the broken lines in the above figure to avoid an adverse effect from wiring capacitance.

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

2. The XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption, and is more prone to malfunction due to noise than the X1 oscillator. Particular care is therefore required with the wiring method when the XT1 clock is used.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

Standard products

<R> **Recommended Oscillator Constants**

(1) X1 Oscillator: Ceramic resonator (T_A = −40 to +85°C)

Manufacturer	Part Number	SMD/ Lead	Frequency (MHz)	Recommended circuit invariable		Oscillation Voltage Range	
				C1 (pF)	C2 (pF)	MIN.(V)	MAX.(V)
Murata Mfg.	CSTCC2M00G56-R0	SMD	2.00	Internal (47)	Internal (47)	1.8	5.5
	CSTLS4M00G56-B0	Lead	4.00	Internal (47)	Internal (47)		
	CSTCR4M00G55-R0	SMD		Internal (39)	Internal (39)		
	CSTLS4M19G56-B0	Lead	4.194	Internal (47)	Internal (47)		
	CSTCR4M19G55-R0	SMD		Internal (39)	Internal (39)		
	CSTLS4M91G56-B0	Lead	4.915	Internal (47)	Internal (47)	2.0	
	CSTCR4M91G55-R0	SMD		Internal (39)	Internal (39)	1.8	
	CSTLS5M00G56-B0	Lead	5.00	Internal (47)	Internal (47)	2.0	
	CSTCR5M00G55-R0	SMD		Internal (39)	Internal (39)	1.8	
	CSTLS6M00G56-B0	Lead	6.00	Internal (47)	Internal (47)	2.2	
	CSTCR6M00G55-R0	SMD		Internal (39)	Internal (39)	1.9	
	CSTLS8M00G56-B0	Lead	8.00	Internal (47)	Internal (47)	2.2	
	CSTCE8M00G55-R0	SMD		Internal (33)	Internal (33)	1.8	
	CSTLS8M38G56-B0	Lead	8.388	Internal (47)	Internal (47)	2.2	
	CSTCE8M38G55-R0	SMD		Internal (33)	Internal (33)	1.8	
	CSTLS10M0G53-B0	SMD	10.0	Internal (15)	Internal (15)	1.8	
	CSTCE10M0G55-R0	SMD		Internal (33)	Internal (33)	2.1	
Murata Mfg. (low-capacitance products)	CSTLS4M91G53-B0	Lead	4.915	Internal (15)	Internal (15)	1.8	5.5
	CSTLS5M00G53-B0	Lead	5.00	Internal (15)	Internal (15)	1.8	
	CSTCR6M00G53-R0	SMD	6.00	Internal (15)	Internal (15)	1.8	
	CSTLS6M00G53-B0	Lead		Internal (15)	Internal (15)	1.8	
	CSTLS8M00G53-B0	Lead	8.00	Internal (15)	Internal (15)	1.8	
	CSTLS8M38G53-B0	Lead	8.388	Internal (15)	Internal (15)	1.8	
	CSTCE10M0G52-R0	SMD	10.0	Internal (10)	Internal (10)	1.8	

Caution The oscillator constants shown above are reference values based on evaluation in a specific environment by the resonator manufacturer. If it is necessary to optimize the oscillator characteristics in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit. The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0/LF3 so that the internal operation conditions are within the specifications of the DC and AC characteristics.

Standard products

DC Characteristics (1/5)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high[Note1]	I_{OH1}	Per pin for P10 to P17, P30 to P34, P40 to P47, P120	4.0 V ≤ V_{DD} ≤ 5.5 V			–3.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–2.5	mA
			1.8 V ≤ V_{DD} < 2.7 V			–1.0	mA
		Per pin for P80 to P83, P90 to P93, P100 to P103, P110 to P113, P130 to P133, P140 to P143, P150 to P153	4.0 V ≤ V_{DD} ≤ 5.5 V			–0.1	mA
			2.7 V ≤ V_{DD} < 4.0 V			–0.1	mA
			1.8 V ≤ V_{DD} < 2.7 V			–0.1	mA
		Total[Note3] of P10 to P17, P30 to P34, P40 to P47, P120	4.0 V ≤ V_{DD} ≤ 5.5 V			–20.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–10.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			–5.0	mA
		Total[Note3] of P80 to P83, P90 to P93, P100 to P103, P110 to P113, P130 to P133, P140 to P143, P150 to P153	4.0 V ≤ V_{DD} ≤ 5.5 V			–2.8	mA
			2.7 V ≤ V_{DD} < 4.0 V			–2.8	mA
			1.8 V ≤ V_{DD} < 2.7 V			–2.8	mA
		Total[Note3] of all pins	4.0 V ≤ V_{DD} ≤ 5.5 V			–22.8	mA
			2.7 V ≤ V_{DD} < 4.0 V			–12.8	mA
			1.8 V ≤ V_{DD} < 2.7 V			–7.8	mA
	I_{OH2}	Per pin for P20 to P27	AV_{REF} = V_{DD}			–0.1	mA
Output current, low[Note2]	I_{OL1}	Per pin for P10 to P17, P30 to P34, P40 to P47, P120	4.0 V ≤ V_{DD} ≤ 5.5 V			8.5	mA
			2.7 V ≤ V_{DD} < 4.0 V			5.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			2.0	mA
		Per pin for P80 to P83, P90 to P93, P100 to P103, P110 to P113, P130 to P133, P140 to P143, P150 to P153	4.0 V ≤ V_{DD} ≤ 5.5 V			0.4	mA
			2.7 V ≤ V_{DD} < 4.0 V			0.4	mA
			1.8 V ≤ V_{DD} < 2.7 V			0.4	mA
		Total[Note3] of P10 to P17, P30 to P34, P40 to P47, P120	4.0 V ≤ V_{DD} ≤ 5.5 V			20.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			15.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			9.0	mA
		Total[Note3] of P80 to P83, P90 to P93, P100 to P103, P110 to P113, P130 to P133, P140 to P143, P150 to P153	4.0 V ≤ V_{DD} ≤ 5.5 V			11.2	mA
			2.7 V ≤ V_{DD} < 4.0 V			11.2	mA
			1.8 V ≤ V_{DD} < 2.7 V			11.2	mA
		Total[Note3] of all pins	4.0 V ≤ V_{DD} ≤ 5.5 V			31.2	mA
			2.7 V ≤ V_{DD} < 4.0 V			26.2	mA
			1.8 V ≤ V_{DD} < 2.7 V			20.2	mA
	I_{OL2}	Per pin for P20 to P27	AV_{REF} = V_{DD}			0.4	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from V_{DD} to an output pin.

2. Value of current at which the device operation is guaranteed even if the current flows from an output pin to GND.

3. Specification under conditions where the duty factor is 70% (time for which current is output is 0.7 × t and time for which current is not output is 0.3 × t, where t is a specific time). The total output current of the pins at a duty factor of other than 70% can be calculated by the following expression.

- Where the duty factor of I_{OH} is n%: Total output current of pins = (I_{OH} × 0.7)/(n × 0.01)

<Example> Where the duty factor is 50%, I_{OH} = 20.0 mA

Total output current of pins = (20.0 × 0.7)/(50 × 0.01) = 28.0 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Standard products

DC Characteristics (2/5)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P10, P16, P17, P32, P80 to P83, P90 to P93, P100 to P103, P110 to P112, P121 to P124, P130 to P133, P140 to P143, P150 to P153		0.7V_{DD}		V_{DD}	V
	V_{IH2}	P11 to P15, P30, P31, P33, P34, P40 to P47, P113, P120, $\overline{\text{RESET}}$, EXCLK		0.8V_{DD}		V_{DD}	V
	V_{IH3}	P20 to P27	AV_{REF} = V_{DD}	0.7AV_{REF}		AV_{REF}	V
Input voltage, low	V_{IL1}	P10, P16, P17, P32, P80 to P83, P90 to P93, P100 to P103, P110 to P112, P121 to P124, P130 to P133, P140 to P143, P150 to P153		0		0.3V_{DD}	V
	V_{IL2}	P11 to P15, P30, P31, P33, P34, P40 to P47, P113, P120, $\overline{\text{RESET}}$, EXCLK		0		0.2V_{DD}	V
	V_{IL3}	P20 to P27	AV_{REF} = V_{DD}	0		0.3AV_{REF}	V
Output voltage, high	V_{OH1}	P10 to P17, P30 to P34, P40 to P47, P120	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –3.0 mA	V_{DD} – 0.7			V
			2.7 V ≤ V_{DD} < 4.0 V, I_{OH1} = –2.5 mA	V_{DD} – 0.5			V
			1.8 V ≤ V_{DD} < 2.7 V, I_{OH1} = –1.0 mA	V_{DD} – 0.5			V
		P80 to P83, P90 to P93, P100 to P103, P110 to P113, P130 to P133, P140 to P143, P150 to P153	I_{OH1} = –0.1 mA	V_{DD} – 0.5			V
	V_{OH2}	P20 to P27	AV_{REF} = V_{DD}, I_{OH2} = –0.1 mA	V_{DD} – 0.5			V
Output voltage, low	V_{OL1}	P10 to P17, P30 to P34, P40 to P47, P120	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 8.5 mA			0.7	V
			2.7 V ≤ V_{DD} < 4.0 V, I_{OL1} = 5.0 mA			0.7	V
			1.8 V ≤ V_{DD} < 2.7 V, I_{OL1} = 2.0 mA			0.5	V
			1.8 V ≤ V_{DD} < 2.7 V, I_{OL1} = 1.0 mA			0.5	V
			1.8 V ≤ V_{DD} < 2.7 V, I_{OL1} = 0.5 mA			0.4	V
		P80 to P83, P90 to P93, P100 to P103, P110 to P113, P130 to P133, P140 to P143, P150 to P153	I_{OL1} = 0.4 mA			0.4	V
	V_{OL2}	P20 to P27	AV_{REF} = V_{DD}, I_{OL2} = 0.4 mA			0.4	V

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Caution The high-level and low-level input voltages of P122/EXCLK vary between the input port mode and external clock mode.

Standard products

DC Characteristics (3/5)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P10 to P17, P30 to P34, P40 to P47, P80 to P83, P90 to P93, P100 to P103, P110 to P113, P120, P130 to P133, P140 to P143, P150 to P153, FLMD0, $\overline{RESET}$	$V_I = V_{DD}$				1	μA
	I_{LIH2}	P20 to P27	$V_I = AV_{REF} = V_{DD}$				1	μA
	I_{LIH3}	P121 to 124 (X1, X2, XT1, XT2)	$V_I = V_{DD}$	I/O port mode			1	μA
				OSC mode			20	μA
Input leakage current, low	I_{LIL1}	P10 to P17, P30 to P34, P40 to P47, P80 to P83, P90 to P93, P100 to P103, P110 to P113, P120, P130 to P133, P140 to P143, P150 to P153, FLMD0, $\overline{RESET}$	$V_I = V_{SS}$				–1	μA
	I_{LIL2}	P20 to P27	$V_I = V_{SS}$, $AV_{REF} = V_{DD}$				–1	μA
	I_{LIL3}	P121 to 124 (X1, X2, XT1, XT2)	$V_I = V_{SS}$	I/O port mode			–1	μA
				OSC mode			–20	μA
Pull-up resistor	R_U	$V_I = V_{SS}$			10	20	100	kΩ
FLMD0 supply voltage	V_{IL}	In normal operation mode			0		0.2V_{DD}	V
	V_{IH}	In self-programming mode			0.8V_{DD}		V_{DD}	V

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Standard products

DC Characteristics (4/5)

(T_A = −40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current	I_{DD1} Note 1	Operating mode	f_{XH} = 10 MHz Note 2, V_{DD} = 5.0 V	Square wave input		1.6	3.0	mA
				Resonator connection		2.3	3.4	
			f_{XH} = 10 MHz Note 2, V_{DD} = 3.0 V	Square wave input		1.5	2.9	mA
				Resonator connection		2.2	3.3	
			f_{XH} = 5 MHz Note 2, V_{DD} = 3.0 V	Square wave input		0.9	1.7	mA
				Resonator connection		1.3	2.0	
			f_{XH} = 5 MHz Note 2, V_{DD} = 2.0 V	Square wave input		0.7	1.4	mA
				Resonator connection		1.0	1.6	
			f_{RH} = 8 MHz, V_{DD} = 5.0 V Note 3			1.4	2.3	mA
			f_{SUB} = 32.768 kHz Note 4, V_{DD} = 5.0 V	Resonator connection		6.7	26	μA
	I_{DD2} Note 1	HALT mode	f_{XH} = 10 MHz Note 2, V_{DD} = 5.0 V	Square wave input		0.4	1.4	mA
				Resonator connection		1.0	1.7	
			f_{XH} = 5 MHz Note 2, V_{DD} = 3.0 V	Square wave input		0.2	0.7	mA
				Resonator connection		0.5	1.0	
			f_{RH} = 8 MHz, V_{DD} = 5.0 V Note 3			0.4	1.2	mA
<R>			f_{SUB} = 32.768 kHz Note 5, V_{DD} = 5.0 V	Resonator connection		2.4	22	μA
	I_{DD3} Note 6	STOP mode	V_{DD} = 5.0 V			1	20	μA
			V_{DD} = 5.0 V, T_A = −40 to +70°C			1	10	μA

<R> **Notes 1.** Total current flowing into the internal power supply (V_{DD}), including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The MAX. values include the peripheral operation current. However, the current flowing into the pull-up resistors and the output current of the port are not included.

2. Not including the operating current of the 8 MHz internal oscillator, 240 kHz internal oscillator and XT1 oscillation, and the current flowing into the A/D converter, watchdog timer, LVI circuit and LCD controller/driver.

3. Not including the operating current of the X1 oscillation, XT1 oscillation and 240 kHz internal oscillator, and the current flowing into the A/D converter, watchdog timer, LVI circuit and LCD controller/driver.

4. Not including the operating current of the X1 oscillation, 8 MHz internal oscillator and 240 kHz internal oscillator, and the current flowing into the A/D converter, watchdog timer, LVI circuit and LCD controller/driver.

<R> **5.** Not including the operating current of the X1 oscillation, 8 MHz internal oscillator and 240 kHz internal oscillator, and the current flowing into the A/D converter, watchdog timer, LVI circuit, LCD controller/driver and real-time counter.

<R> **6.** Total current flowing into the internal power supply (V_{DD}), including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. However, the current flowing into the pull-up resistors and the output current of the port, the operating current of the 240 kHz internal oscillator and XT1 oscillation, and the current flowing into the A/D converter, watchdog timer, LVI circuit, LCD controller/driver and real-time counter are not included.

Remarks 1. f_{XH}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{RH}: Internal high-speed oscillation clock frequency

3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

Standard products

DC Characteristics (5/5)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

	Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
	Watchdog timer operating current	I_{WDT} Note 1	During 240 kHz internal low-speed oscillation clock operation			5	10	μA
	LVI operating current	I_{LVI} Note 2				9	18	μA
	Successive approximation type A/D converter operating current	I_{ADC1} Note 3		2.3 V ≤ AV_{REF} ≤ V_{DD}		0.86	1.9	mA
<R>	ΔΣ type A/D converter operating current	I_{ADC2} Note 3		2.7 V ≤ AV_{REF} ≤ V_{DD}		1.4	2.7	mA
	LCD operating current	I_{LCD1} Note 4	LCD display off (deselect signal output) (LCDON = 0, SCOC = 1)	V_{DD} = 5.0 V		3.0	8.0	μA
				V_{DD} = 3.0 V		2.0	5.0	μA
		I_{LCD2} Note 4	LCD display on (LCDON = 1, SCOC = 1)	V_{DD} = 5.0 V		3.0	8.0	μA
				V_{DD} = 3.0 V		2.0	5.0	μA

Notes 1. This includes only the current that flows through the watchdog timer (including the operating current of the 240 kHz internal oscillator). When the watchdog timer is operating in HALT mode or STOP mode, the current value of the 78K0/LF3 is obtained by adding I_{WDT} to I_{DD2} or I_{DD3}.

2. This includes only the current that flows through the LVI circuit. When the LVI circuit is operating in HALT mode or STOP mode, the current value of the 78K0/LF3 is obtained by adding I_{LVI} to I_{DD2} or I_{DD3}.

3. This includes only the current that flows through the A/D converter (AV_{REF}). When the A/D converter is operating in HALT mode or STOP mode, the current value of the 78K0/LF3 is obtained by adding I_{ADC1} or I_{ADC2} to I_{DD1} or I_{DD2}.

4. This includes only the current that flows through the LCD controller/driver. Not including the current that flows through the LCD divider resistor. The current value of the 78K0/LF3 is obtained by adding the LCD operating current (I_{LCD1} or I_{LCD2}) to the supply current (I_{DD1}, I_{DD2}, or I_{DD3}).

Standard products

AC Characteristics

(1) Basic operation

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

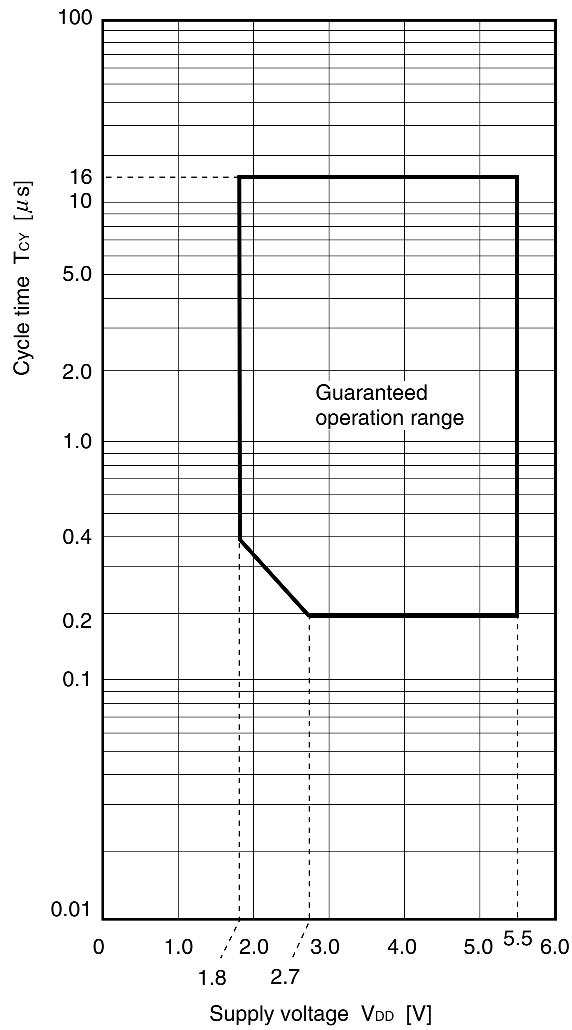
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{XP}) operation	2.7 V ≤ V_{DD} ≤ 5.5 V	0.2		16	μs
			1.8 V ≤ V_{DD} < 2.7 V	0.4		16	μs
		Subsystem clock (f_{SUB}) operation		114	122	125	μs
Peripheral hardware clock frequency	f_{PNS}	XSEL = 1	2.7 V ≤ V_{DD} ≤ 5.5 V			10	MHz
			1.8 V ≤ V_{DD} < 2.7 V			5	MHz
		XSEL = 0	2.7 V ≤ V_{DD} ≤ 5.5 V	7.6		8.4	MHz
			1.8 V ≤ V_{DD} < 2.7 V Note 1	6.75		8.4	MHz
External main system clock frequency	f_{EXCLK}	2.7 V ≤ V_{DD} ≤ 5.5 V		2.0		10.0	MHz
		1.8 V ≤ V_{DD} < 2.7 V		2.0		5.0	MHz
External main system clock input high-level width, low-level width	t_{EXCLKH}, t_{EXCLKL}	2.7 V ≤ V_{DD} ≤ 5.5 V		48		500	ns
		1.8 V ≤ V_{DD} < 2.7 V		96		500	ns
TI000 input high-level width, low-level width	t_{TIH0}, t_{TIL0}	2.7 V ≤ V_{DD} ≤ 5.5 V		$2/f_{sam}$ + 0.2 Note 2			μs
		1.8 V ≤ V_{DD} < 2.7 V		$2/f_{sam}$ + 0.5 Note 2			μs
TI50, TI51, TI52 input frequency	f_{TI5}	4.0 V ≤ V_{DD} ≤ 5.5 V	TI50, TI51			10	MHz
			TI52			16	MHz
		2.7 V ≤ V_{DD} < 4.0 V				10	MHz
		1.8 V ≤ V_{DD} < 2.7 V				5	MHz
TI50, TI51, TI52 input high-level width, low-level width	t_{TIH5}, t_{TIL5}	4.0 V ≤ V_{DD} ≤ 5.5 V	TI50, TI51	50			ns
			TI52	31.25			ns
		2.7 V ≤ V_{DD} < 4.0 V		50			ns
		1.8 V ≤ V_{DD} < 2.7 V		100			ns
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}			1			μs
Key return input low-level width	t_{KR}			250			ns
$\overline{\text{RESET}}$ low-level width	t_{RSL}			10			μs

Notes 1. A characteristic of the main system clock frequency. Set the clock divider to be set using a peripheral function to $f_{RH}/2$ or less.

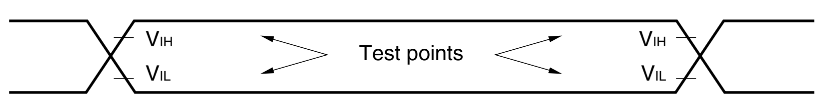
2. Selection of $f_{sam} = f_{PRS}$, $f_{PRS}/4$, $f_{PRS}/256$ is possible using bits 0 and 1 (PRM000, PRM001) of prescaler mode registers 00 (PRM00). Note that when selecting the TI000 valid edge as the count clock, $f_{sam} = f_{PRS}$.

Standard products

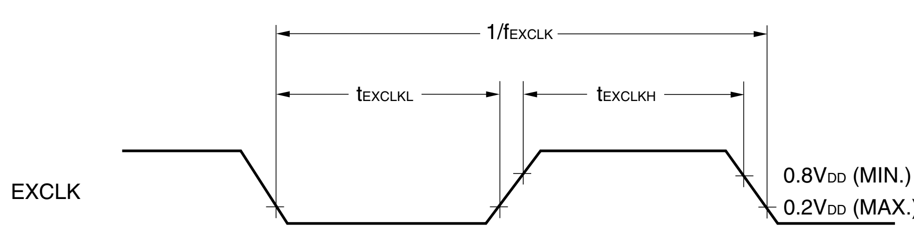
T_{CY} vs. V_{DD} (Main System Clock Operation)



AC Timing Test Points (Excluding External Main System Clock)

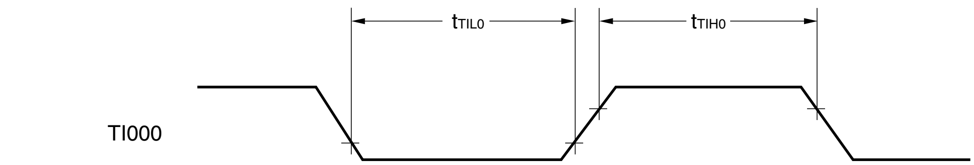


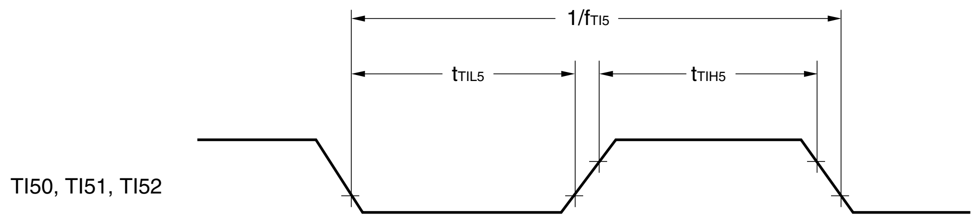
External Main System Clock Timing



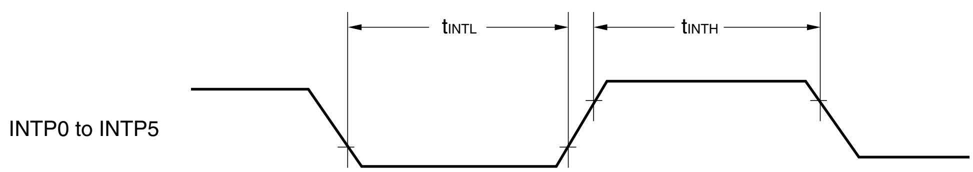
Standard products

TI Timing

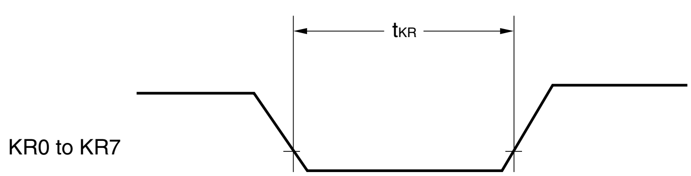




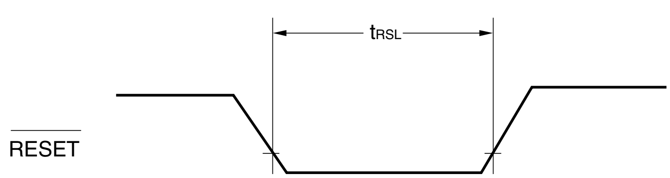
Interrupt Request Input Timing



Key Interrupt Input Timing



$\overline{\text{RESET}}$ Input Timing



(2) Serial interface
(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					250	kbps

(3) Serial interface
(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

(a) UART6 (Dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					625	kbps

(b) UART0 (Dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					625	kbps

Standard products

(c) CSI10 (Master mode, $\overline{SCK10}$... internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCK10}$ cycle time	t_{KCY1}	$2.7\ V \leq V_{DD} \leq 5.5\ V$	250			ns
		$1.8\ V \leq V_{DD} < 2.7\ V$	500			ns
$\overline{SCK10}$ high-/low-level width	t_{KH1}, t_{KL1}	$2.7\ V \leq V_{DD} \leq 5.5\ V$	$t_{KCY1}/2 - 25$ [Note 1]			ns
		$1.8\ V \leq V_{DD} < 2.7\ V$	$t_{KCY1}/2 - 50$ [Note 1]			ns
SI10 setup time (to $\overline{SCK10}$↑)	t_{SIK1}	$2.7\ V \leq V_{DD} \leq 5.5\ V$	80			ns
		$1.8\ V \leq V_{DD} < 2.7\ V$	170			ns
SI10 hold time (from $\overline{SCK10}$↑)	t_{KSI1}		30			ns
Delay time from $\overline{SCK10}$↓ to SO10 output	t_{KSO1}	C = 50 pF [Note 2]			40	ns

Notes 1. This value is when high-speed system clock (f_{XH}) is used.

2. C is the load capacitance of the $\overline{SCK10}$ and SO10 output lines.

(d) CSI10 (Slave mode, $\overline{SCK10}$... external clock input)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCK10}$ cycle time	t_{KCY2}			400			ns
$\overline{SCK10}$ high-/low-level width	t_{KH2}, t_{KL2}			$t_{KCY2}/2$			ns
SI10 setup time (to $\overline{SCK10}$↑)	t_{SIK2}			80			ns
SI10 hold time (from $\overline{SCK10}$↑)	t_{KSI2}			50			ns
Delay time from $\overline{SCK10}$↓ to SO10 output	t_{KSO2}	C = 50 pF [Note]	$2.7\ V \leq V_{DD} \leq 5.5\ V$			120	ns
			$1.8\ V \leq V_{DD} < 2.7\ V$			165	ns

Note C is the load capacitance of the SO10 output line.

Standard products

(e) AUTOCSI (Master mode, $\overline{\text{SCKA0}}$... internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCKA0}}$ cycle time	t_{KCY3}	4.0 V ≤ V_{DD} ≤ 5.5 V	600			ns
		2.7 V ≤ V_{DD} < 4.0 V	1200			ns
		1.8 V ≤ V_{DD} < 2.7 V	1800			ns
$\overline{\text{SCKA0}}$ high-/low-level width	t_{KH3}, t_{KL3}	4.0 V ≤ V_{DD} ≤ 5.5 V	t_{KCY3}/2 – 50			ns
		2.7 V ≤ V_{DD} < 4.0 V	t_{KCY3}/2 – 100			ns
		1.8 V ≤ V_{DD} < 2.7 V	t_{KCY3}/2 – 200			ns
SIA0 setup time (to $\overline{\text{SCKA0}}$↑)	t_{SIK3}	2.7 V ≤ V_{DD} ≤ 5.5 V	100			ns
		1.8 V ≤ V_{DD} < 2.7 V	200			ns
SIA0 hold time (from $\overline{\text{SCKA0}}$↑)	t_{KSI3}		300			ns
Delay time from $\overline{\text{SCKA0}}$↓ to SOA0 output	t_{KSO3}	C = 100 pF Note, 4.0 V ≤ V_{DD} ≤ 5.5 V			200	ns
		C = 100 pF Note, 2.7 V ≤ V_{DD} < 4.0 V			300	ns
		C = 100 pF Note, 1.8 V ≤ V_{DD} < 2.7 V			400	ns

Note C is the load capacitance of the $\overline{\text{SCKA0}}$ and SOA0 output lines.

(f) AUTOCSI (Slave mode, $\overline{\text{SCKA0}}$... external clock input)

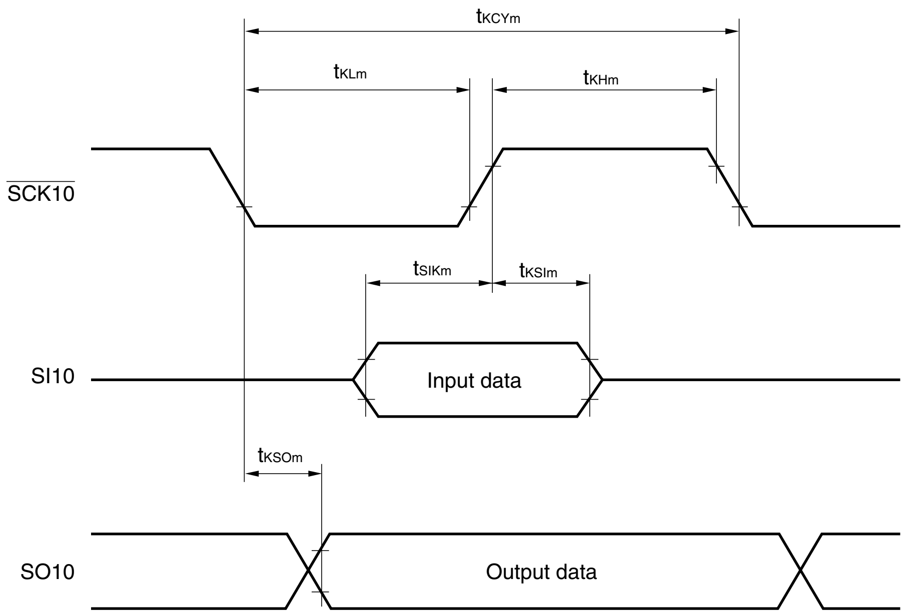
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCKA0}}$ cycle time	t_{KCY4}	4.0 V ≤ V_{DD} ≤ 5.5 V	600			ns
		2.7 V ≤ V_{DD} < 4.0 V	1200			ns
		1.8 V ≤ V_{DD} < 2.7 V	1800			ns
$\overline{\text{SCKA0}}$ high-/low-level width	t_{KH4}, t_{KL4}	4.0 V ≤ V_{DD} ≤ 5.5 V	300			ns
		2.7 V ≤ V_{DD} < 4.0 V	600			ns
		1.8 V ≤ V_{DD} < 2.7 V	900			ns
SIA0 setup time (to $\overline{\text{SCKA0}}$↑)	t_{SIK4}	4.0 V ≤ V_{DD} ≤ 5.5 V	100			ns
SIA0 hold time (from $\overline{\text{SCKA0}}$↑)	t_{KSI4}		2/f_W+100			ns
Delay time from $\overline{\text{SCKA0}}$↓ to SOA0 output	t_{KSO4}	C = 100 pF Note, 4.0 V ≤ V_{DD} ≤ 5.5 V			2/f_W+100	ns
		C = 100 pF Note, 2.7 V ≤ V_{DD} < 4.0 V			2/f_W+200	ns
		C = 100 pF Note, 1.8 V ≤ V_{DD} < 2.7 V			2/f_W+300	ns
$\overline{\text{SCKA0}}$ rise/fall time	t_{R4}, t_{F4}				1000	ns

Note C is the load capacitance of the SOA0 output line.

Standard products

Serial Transfer Timing (1/2)

CSI10:

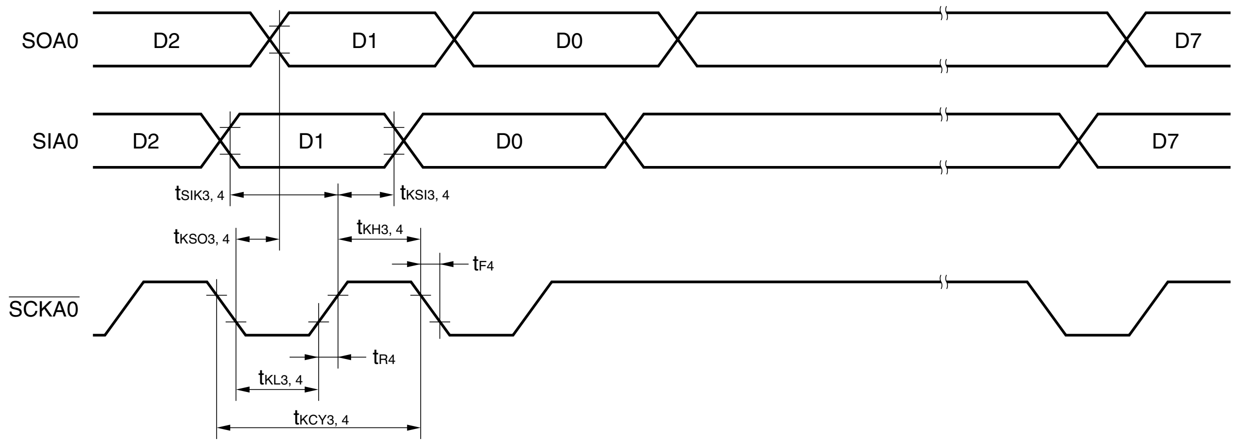


Remark m = 1, 2

Standard products

Serial Transfer Timing (2/2)

CSIA0:



Standard products

10-bit successive approximation type A/D Converter Characteristics (μPD78F048x and 78F049x only)
(T_A = –40 to +85°C, 2.3 V ≤ AV_{REF} ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	R_{ES1}				10	bit
Overall error[Notes 1, 2]	A_{INL}	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.6	%FSR
		2.3 V ≤ AV_{REF} < 2.7 V			±1.2	%FSR
Conversion time	t_{CONV}	4.0 V ≤ AV_{REF} ≤ 5.5 V	6.1		36.7	μs
		2.7 V ≤ AV_{REF} < 4.0 V	12.2		36.7	μs
		2.3 V ≤ AV_{REF} < 2.7 V	27		66.6	μs
Zero-scale error[Notes 1, 2]	E_{ZS}	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.6	%FSR
		2.3 V ≤ AV_{REF} < 2.7 V			±0.6	%FSR
Full-scale error[Notes 1, 2]	E_{FS}	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.6	%FSR
		2.3 V ≤ AV_{REF} < 2.7 V			±0.6	%FSR
Integral non-linearity error[Note 1]	I_{LE1}	4.0 V ≤ AV_{REF} ≤ 5.5 V			±2.5	LSB
		2.7 V ≤ AV_{REF} < 4.0 V			±4.5	LSB
		2.3 V ≤ AV_{REF} < 2.7 V			±6.5	LSB
Differential non-linearity error[Note 1]	D_{LE1}	4.0 V ≤ AV_{REF} ≤ 5.5 V			±1.5	LSB
		2.7 V ≤ AV_{REF} < 4.0 V			±2.0	LSB
		2.3 V ≤ AV_{REF} < 2.7 V			±2.0	LSB
Analog input voltage	V_{AIN1}		AV_{SS}		AV_{REF}	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

Standard products

<R> **16-bit ΔΣ type A/D Converter Characteristics (μPD78F049x only)**
(T_A = –40 to +85°C, 2.7 V ≤ AV_{REF} ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Resolution	R_{ES2}				8		16	bit
Sampling clock[Note 1]	f_{VP}	At differential input		3.5 V ≤ AV_{REF} ≤ 5.5 V	0.016		1.25	MHz
				2.7 V ≤ AV_{REF} < 3.5 V	0.016		0.625	MHz
		At single input		2.85 V ≤ AV_{REF} ≤ 5.5 V	0.016		0.625	MHz
				2.7 V ≤ AV_{REF} < 2.85 V	0.016		0.525	MHz
Integral non-linearity error (relative accuracy)	I_{LE2}	At differential input[Note 2]	14-bit resolution[Note 3]	AV_{REF} = 5.0 V		±1.0		LSB
				3.5 V ≤ AV_{REF} ≤ 5.5 V		±1.7		LSB
				2.7 V ≤ AV_{REF} < 3.5 V		±2.6		LSB
		At single input[Note 2]	12-bit resolution[Note 3]			±2.8		LSB
Differential non-linearity error (relative accuracy)	D_{LE2}	At differential input[Note 2]	14-bit resolution[Note 3]	AV_{REF} = 5.0 V		±1.0		LSB
				3.5 V ≤ AV_{REF} ≤ 5.5 V		±1.7		LSB
				2.7 V ≤ AV_{REF} < 3.5 V		±2.6		LSB
		At single input[Note 2]	12-bit resolution[Note 3]			±2.8		LSB
Offset	EOS	At differential input				±0.032		%FSR
		At single input				±0.16		%FSR
Gain error	GE	At differential input				±0.09		%
		At single input				±0.1		%
Reference voltage	REF+					AV_{REF}		V
	REF–					AV_{SS}		V
Analog input voltage	V_{AIN2}	In high-accuracy mode OFF			0		REF+	V
		In high-accuracy mode ON			0.1REF+		0.9REF+	V

Notes 1. The conversion time can be calculated by using the following expression, based on the sampling clock (f_{VP}) and set resolution (N bits).

Conversion time = $2^N / f_{VP}$

2. These values apply when the high-accuracy mode is set to be on during differential input, or when the high-accuracy mode is set to be off during single input.

3. The characteristics of resolutions (N bits) other than those stated as conditions in the integral linearity error (I_{LE2}) and differential linearity error (D_{LE2}) columns can be calculated by using the following expressions.

- During differential input
  I_{LE2} in N-bit resolution = I_{LE2} in 14-bit resolution × $2^{(N-14)}$
  D_{LE2} in N-bit resolution = D_{LE2} in 14-bit resolution × $2^{(N-14)}$
- During single input
  I_{LE2} in N-bit resolution = I_{LE2} in 12-bit resolution × $2^{(N-12)}$
  D_{LE2} in N-bit resolution = D_{LE2} in 12-bit resolution × $2^{(N-12)}$

Remark In the 16-bit ΔΣ type A/D converter characteristics, the approximation line is defined by the least-squares method.

Standard products

LCD Characteristics

(1) Resistance division method

(a) Static display mode (T_A = –40 to +85°C, 1.8 V ≤ V_{LCD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)[Note 3]

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD drive voltage	V_{LCD}	**Note 3**			V_{DD}	V
LCD divider resistor[Note 1]	R_{LCD}		60	100	150	kΩ
LCD output resistor[Note 2] (Common)	R_{ODC}				40	kΩ
LCD output resistor[Note 2] (Segment)	R_{ODS}				200	kΩ

(b) 1/3 bias method (T_A = –40 to +85°C, 1.8 V ≤ V_{LCD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)[Note 3]

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD drive voltage	V_{LCD}	**Note 3**			V_{DD}	V
LCD divider resistor[Note 1]	R_{LCD}		60	100	150	kΩ
LCD output resistor[Note 2] (Common)	R_{ODC}				40	kΩ
LCD output resistor[Note 2] (Segment)	R_{ODS}				200	kΩ

(c) 1/2 bias method, 1/4 bias method (T_A = –40 to +85°C, 1.8 V ≤ V_{LCD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)[Note 3]

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD drive voltage	V_{LCD}	**Note 3**			V_{DD}	V
LCD divider resistor[Note 1]	R_{LCD}		60	100	150	kΩ
LCD output resistor[Note 2] (Common)	R_{ODC}				40	kΩ
LCD output resistor[Note 2] (Segment)	R_{ODS}				200	kΩ

Notes 1. Internal resistance division method only.

2. The output resistor is a resistor connected between one of the V_{LC0}, V_{LC1}, V_{LC2} and V_{SS} pins, and either of the SEG and COM pins.

3. Set VAON based on the following conditions.

<When set to the static display mode>

- When 2.0V≤ V_{LCD} ≤ V_{DD} ≤ 5.5 V: VAON = 0
- When 1.8V≤ V_{LCD} ≤ V_{DD} ≤ 3.6 V: VAON = 1

<When set to the 1/3 bias method>

- When 2.5V≤ V_{LCD} ≤ V_{DD} ≤ 5.5 V: VAON = 0
- When 1.8V≤ V_{LCD} ≤ V_{DD} ≤ 3.6 V: VAON = 1

<R> <When set to the 1/2 bias method or 1/4 bias method>

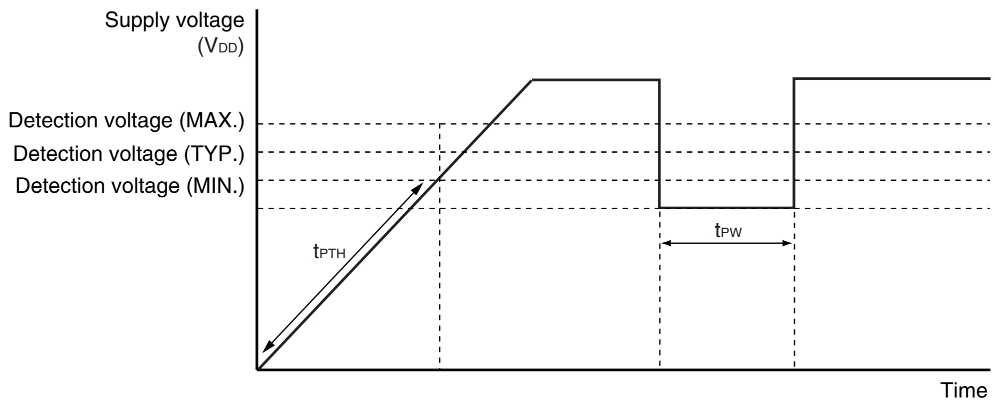
- When 2.7V≤ V_{LCD} ≤ V_{DD} ≤ 5.5 V: VAON = 0
- When 1.8V≤ V_{LCD} ≤ V_{DD} ≤ 3.6 V: VAON = 1

Standard products

1.59 V POC Circuit Characteristics (T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POC}		1.44	1.59	1.74	V
Power supply voltage rise inclination	t_{PTH}	V_{DD}: 0 V → change inclination of V_{POC}	0.5			V/ms
Minimum pulse width	t_{PW}		200			μs

POC Circuit Timing

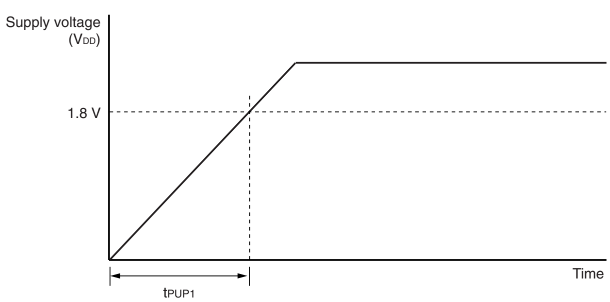


Supply Voltage Rise Time (T_A = –40 to +85°C, V_{SS} = 0 V)

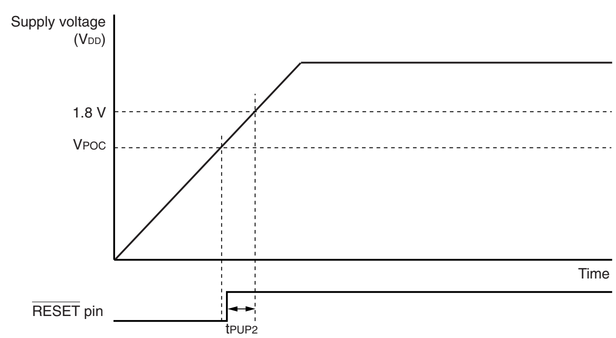
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Maximum time to rise to 1.8 V (V_{DD} (MIN.)) (V_{DD}: 0 V → 1.8 V)	t_{PUP1}	POCMODE (option byte) = 0, when $\overline{\text{RESET}}$ input is not used			3.6	ms
Maximum time to rise to 1.8 V (V_{DD} (MIN.)) (releasing $\overline{\text{RESET}}$ input → V_{DD}: 1.8 V)	t_{PUP2}	POCMODE (option byte) = 0, when $\overline{\text{RESET}}$ input is used			1.9	ms

Supply Voltage Rise Time Timing

- When $\overline{\text{RESET}}$ pin input is not used



- When $\overline{\text{RESET}}$ pin input is used



2.7 V POC Circuit Characteristics (T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage on application of supply voltage	V_{DDPOC}	POCMODE (option bye) = 1	2.50	2.70	2.90	V

Standard products

LVI Circuit Characteristics (T_A = –40 to +85°C, $V_{POC} \le V_{DD} \le$ 5.5 V, V_{SS} = 0 V)

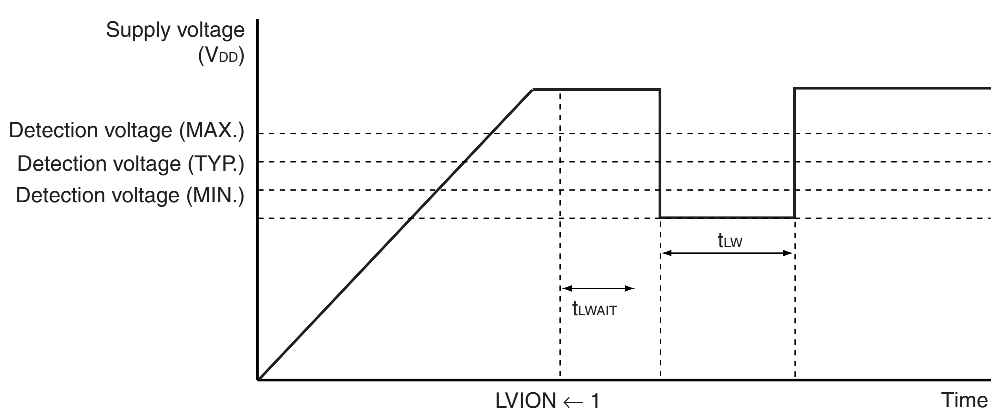
Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVI0}		4.14	4.24	4.34	V
		V_{LVI1}		3.99	4.09	4.19	V
		V_{LVI2}		3.83	3.93	4.03	V
		V_{LVI3}		3.68	3.78	3.88	V
		V_{LVI4}		3.52	3.62	3.72	V
		V_{LVI5}		3.37	3.47	3.57	V
		V_{LVI6}		3.22	3.32	3.42	V
		V_{LVI7}		3.06	3.16	3.26	V
		V_{LVI8}		2.91	3.01	3.11	V
		V_{LVI9}		2.75	2.85	2.95	V
		V_{LVI10}		2.60	2.70	2.80	V
		V_{LVI11}		2.45	2.55	2.65	V
		V_{LVI12}		2.29	2.39	2.49	V
		V_{LVI13}		2.14	2.24	2.34	V
		V_{LVI14}		1.98	2.08	2.18	V
		V_{LVI15}		1.83	1.93	2.03	V
	External input pin[Note 1]	EXLVI	EXLVI < V_{DD}, 1.8 V ≤ V_{DD} ≤ 5.5 V	1.11	1.21	1.31	V
Minimum pulse width		t_{LW}		200			μs
Operation stabilization wait time[Note 2]		t_{LWAIT}				10	μs

Notes 1. The EXLVI/P120/INTP0 pin is used.

2. Time required from setting bit 7 (LVION) of the low-voltage detection register (LVIM) to 1 to operation stabilization.

Remark $V_{LVI(n-1)} > V_{LVIn}$: n = 1 to 15

LVI Circuit Timing

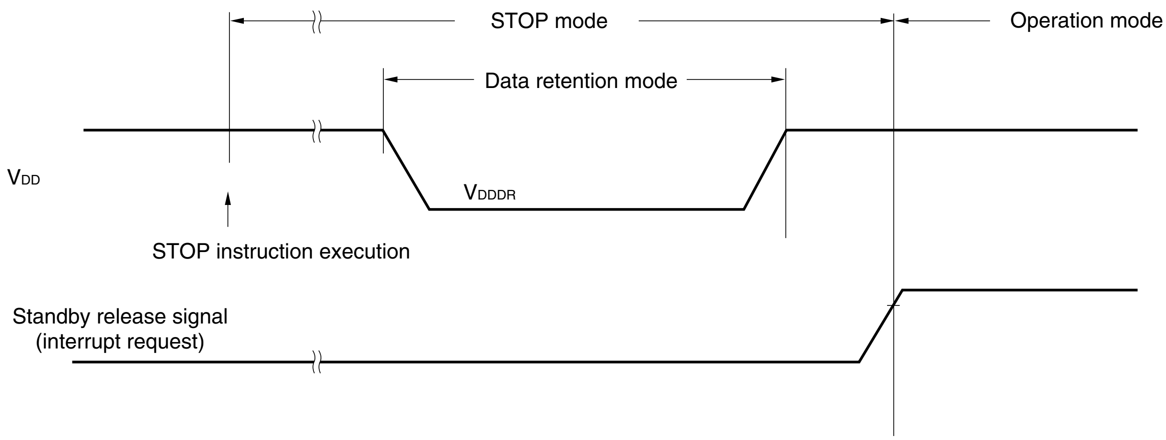


Standard products

Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics (T_A = –40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.44[Note]		5.5	V

Note The value depends on the POC detection voltage. When the voltage drops, the data is retained until a POC reset is effected, but data is not retained when a POC reset is effected.



Flash Memory Programming Characteristics
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

• **Basic characteristics**

Parameter		Symbol	Conditions			MIN.	TYP.	MAX.	Unit
V_{DD} supply current		I_{DD}					4.5	11.0	mA
Erase time[Note 1, 2]	All block	T_{eraca}					20	200	ms
	Block unit	T_{erasa}					20	200	ms
Write time (in 8-bit units)[Note 1]		T_{wrwa}					10	100	μs
<R> Number of rewrites per chip		C_{erwr}	1 erase + 1 write after erase = 1 rewrite[Note 3]	When a flash memory programmer is used, and the libraries provided by NEC Electronics are used	Retention: 15 years	1000			Times
				When the EEPROM emulation libraries provided by NEC Electronics are used, and the rewritable ROM size is 4 KB	Retention: 3 years[Note 4]	10000			Times

<R> **Notes 1.** Characteristic of the flash memory. For the characteristic when a dedicated flash programmer, PG-FP5, is used and the rewrite time during self programming, see **Tables 28-12** and **28-13**.

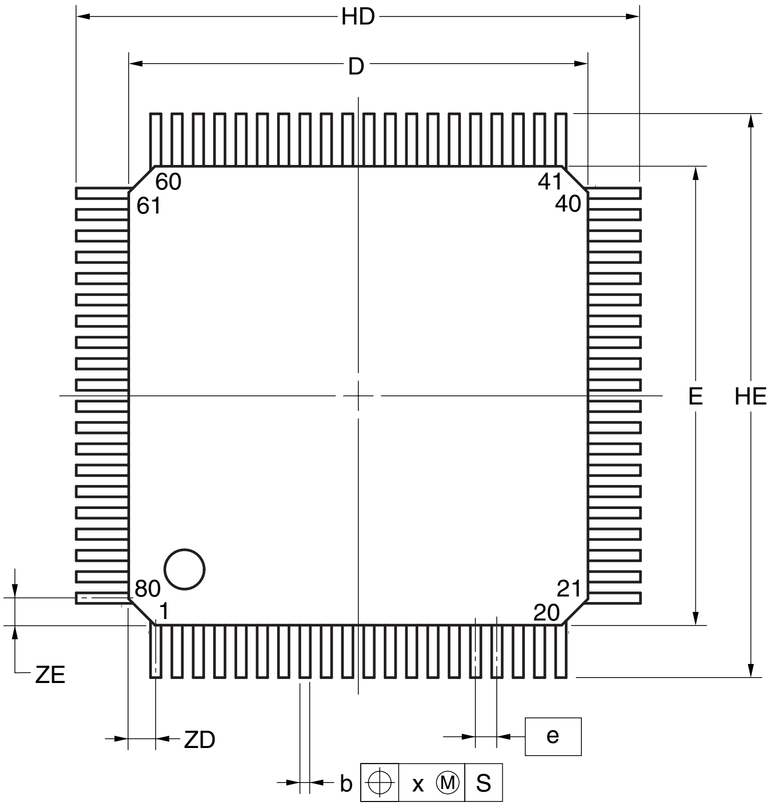
2. The prewrite time before erasure and the erase verify time (writeback time) are not included.

3. When a product is first written after shipment, "erase → write" and "write only" are both taken as one rewrite.

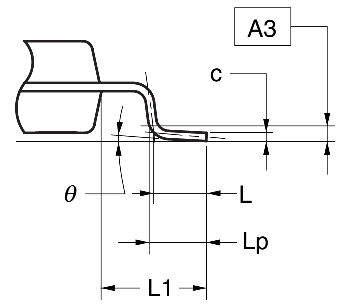
<R> **4.** Data retention is guaranteed for three years after data has been written. If rewriting has been performed, data retention is guaranteed for another three years thereafter.

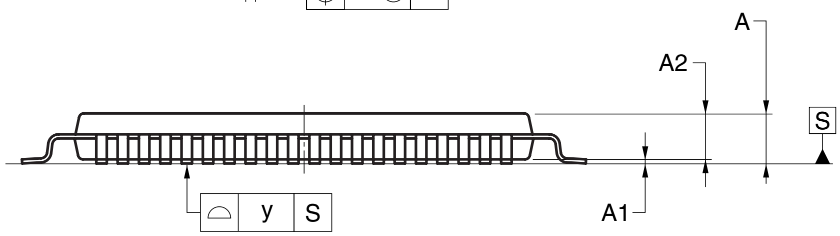
Remark f_{XP}: Main system clock oscillation frequency

80-PIN PLASTIC LQFP (14x14)



detail of lead end





(UNIT:mm)

ITEM	DIMENSIONS
D	14.00±0.20
E	14.00±0.20
HD	17.20±0.20
HE	17.20±0.20
A	1.70 MAX.
A1	0.125±0.075
A2	1.40±0.05
A3	0.25
b	$0.30^{+0.08}_{-0.04}$
c	$0.125^{+0.075}_{-0.025}$
L	0.80
Lp	0.886±0.15
L1	1.60±0.20
θ	$3°^{+5°}_{-3°}$
e	0.65
x	0.13
y	0.10
ZD	0.825
ZE	0.825

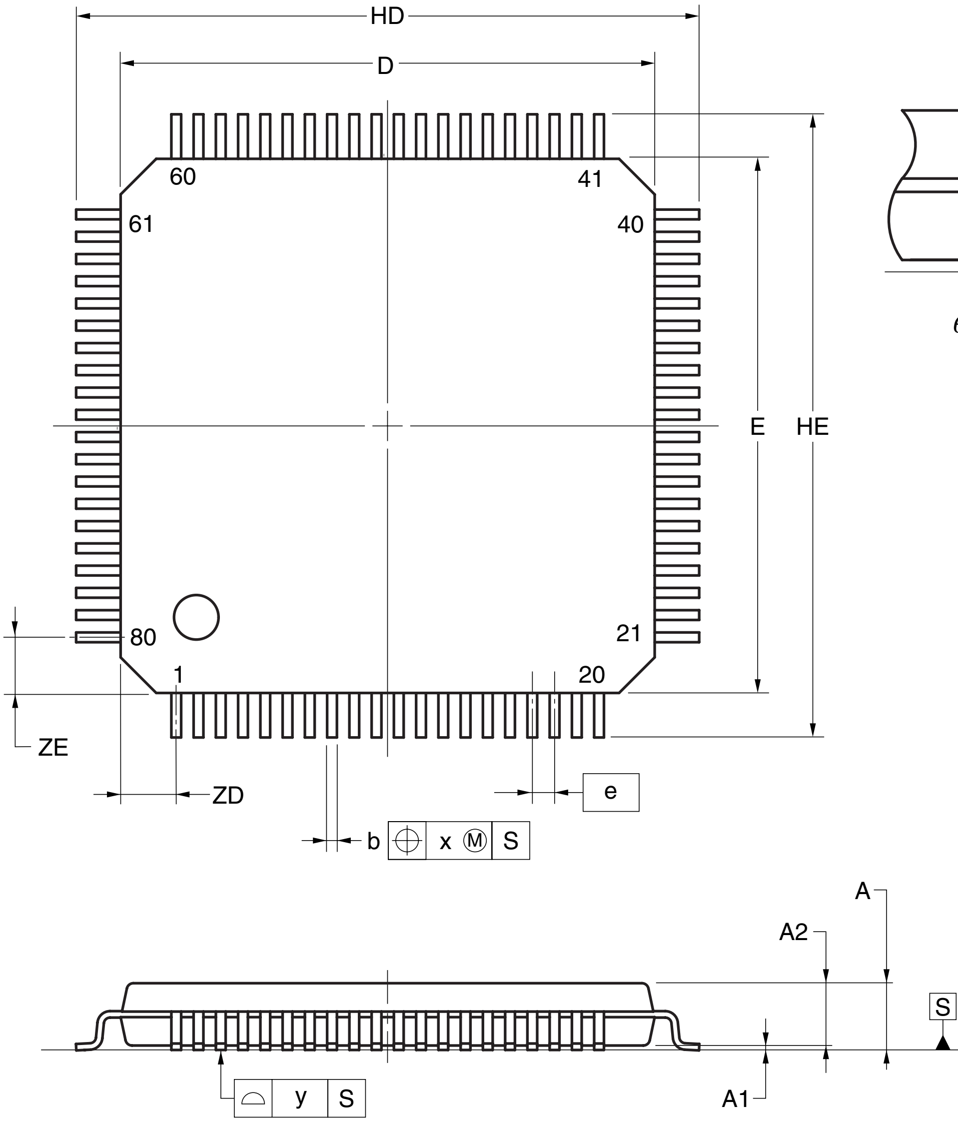
P80GC-65-GAD

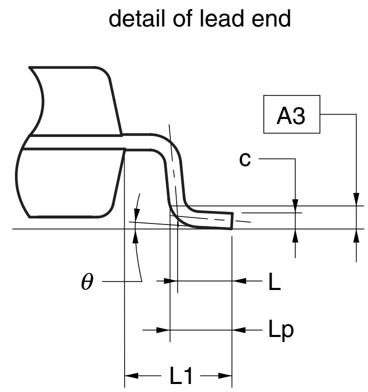
NOTE

Each lead centerline is located within 0.13 mm of its true position at maximum material condition.

© NEC Electronics Corporation 2005

80-PIN PLASTIC LQFP (FINE PITCH) (12x12)





(UNIT:mm)

ITEM	DIMENSIONS
D	12.00±0.20
E	12.00±0.20
HD	14.00±0.20
HE	14.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	$0.20^{+0.07}_{-0.03}$
c	$0.125^{+0.075}_{-0.025}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3°^{+5°}_{-3°}$
e	0.50
x	0.08
y	0.08
ZD	1.25
ZE	1.25

P80GK-50-GAK

NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

© NEC Electronics Corporation 2005

<R>

CHAPTER 33 RECOMMENDED SOLDERING CONDITIONS

These products should be soldered and mounted under the following recommended conditions.

For soldering methods and conditions other than those recommended below, please contact an NEC Electronics sales representative.

For technical information, see the following website.

Semiconductor Device Mount Manual (http://www.necel.com/pkg/en/mount/index.html)

Table 33-1. Surface Mounting Type Soldering Conditions

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 260°C, Time: 60 seconds max. (at 220°C or higher), Count: 3 times or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 to 72 hours)	IR60-107-3
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).

CHAPTER 34 CAUTIONS FOR WAIT

34.1 Cautions for Wait

This product has two internal system buses.

One is a CPU bus and the other is a peripheral bus that interfaces with the low-speed peripheral hardware.

Because the clock of the CPU bus and the clock of the peripheral bus are asynchronous, unexpected illegal data may be passed if an access to the CPU conflicts with an access to the peripheral hardware.

When accessing the peripheral hardware that may cause a conflict, therefore, the CPU repeatedly executes processing, until the correct data is passed.

As a result, the CPU does not start the next instruction processing but waits. If this happens, the number of execution clocks of an instruction increases by the number of wait clocks (for the number of wait clocks, see **Tables 34-1** and **34-2**). This must be noted when real-time processing is performed.

34.2 Peripheral Hardware That Generates Wait

Table 34-1 lists the registers that issue a wait request when accessed by the CPU, and the number of CPU wait clocks and Table 34-2 lists the RAM accesses that issue a wait request and the number of CPU wait clocks.

Table 34-1. Registers That Generate Wait and Number of CPU Wait Clocks

<table>
<tr><th>Peripheral Hardware</th><th>Register</th><th>Access</th><th>Number of Wait Clocks</th></tr>
<tr><td>Serial interface UART0</td><td>ASIS0</td><td>Read</td><td>1 clock (fixed)</td></tr>
<tr><td>Serial interface UART6</td><td>ASIS6</td><td>Read</td><td>1 clock (fixed)</td></tr>
<tr><td rowspan="5">10-bit successive approximation type A/D converter</td><td>ADM</td><td>Write</td><td rowspan="4">1 to 5 clocks (when $f_{AD} = f_{PRS}/2$ is selected)
1 to 7 clocks (when $f_{AD} = f_{PRS}/3$ is selected)
1 to 9 clocks (when $f_{AD} = f_{PRS}/4$ is selected)
2 to 13 clocks (when $f_{AD} = f_{PRS}/6$ is selected)
2 to 17 clocks (when $f_{AD} = f_{PRS}/8$ is selected)
2 to 25 clocks (when $f_{AD} = f_{PRS}/12$ is selected)</td></tr>
<tr><td>ADS</td><td>Write</td></tr>
<tr><td>ADPC</td><td>Write</td></tr>
<tr><td>ADCR</td><td>Read</td></tr>
<tr><td colspan="3">The above number of clocks is when the same source clock is selected for f_{CPU} and f_{PRS}. The number of wait clocks can be calculated by the following expression and under the following conditions.
<Calculating number of wait clocks>
• Number of wait clocks = $\frac{2 f_{CPU}}{f_{AD}} + 1$
* Fraction is truncated if the number of wait clocks ≤ 0.5 and rounded up if the number of wait clocks > 0.5.
f_{AD}: A/D conversion clock frequency ($f_{PRS}/2$ to $f_{PRS}/12$)
f_{CPU}: CPU clock frequency
f_{PRS}: Peripheral hardware clock frequency
f_{XP}: Main system clock frequency
<Conditions for maximum/minimum number of wait clocks>
• Maximum number of times: Maximum speed of CPU (f_{XP}), lowest speed of A/D conversion clock ($f_{PRS}/12$)
• Minimum number of times: Minimum speed of CPU ($f_{SUB}/2$), highest speed of A/D conversion clock ($f_{PRS}/2$)</td></tr>
</table>

Caution When the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped, do not access the registers listed above using an access method in which a wait request is issued.

Remark The clock is the CPU clock (f_{CPU}).

Table 34-2. RAM Accesses That Generate Wait and Number of CPU Wait Clocks

<table>
<tr><th>Area</th><th>Access</th><th>Number of Wait Clocks</th></tr>
<tr><td>Buffer RAM of CSIA0</td><td>Write</td><td>See the following calculation formula[Note]</td></tr>
<tr><td colspan="3"><Calculating maximum number of wait clocks>
• Maximum Number of wait clocks = $\frac{5 f_{CPU}}{f_W} + 1$
* Fraction is truncated if the number of wait clocks multiplied by ($1/f_{CPU}$) is equal or lower than t_{CPUL} and rounded up if higher than t_{CPUL}.
f_W: Frequency of base clock selected by CKS00 bit of CSIS0 register (CKS00 = 0: f_{PRS}, CKS00 = 1: $f_{PRS}/2$)
f_{CPU}: CPU clock frequency
t_{CPUL}: CPU clock low-level width
f_{PRS}: Peripheral hardware clock frequency</td></tr>
</table>

Note No waits are generated when five CSIA0 operating clocks or more are inserted between writing to the RAM from the CSIA0 and writing to the buffer RAM from the CPU.

<R>

APPENDIX A DEVELOPMENT TOOLS

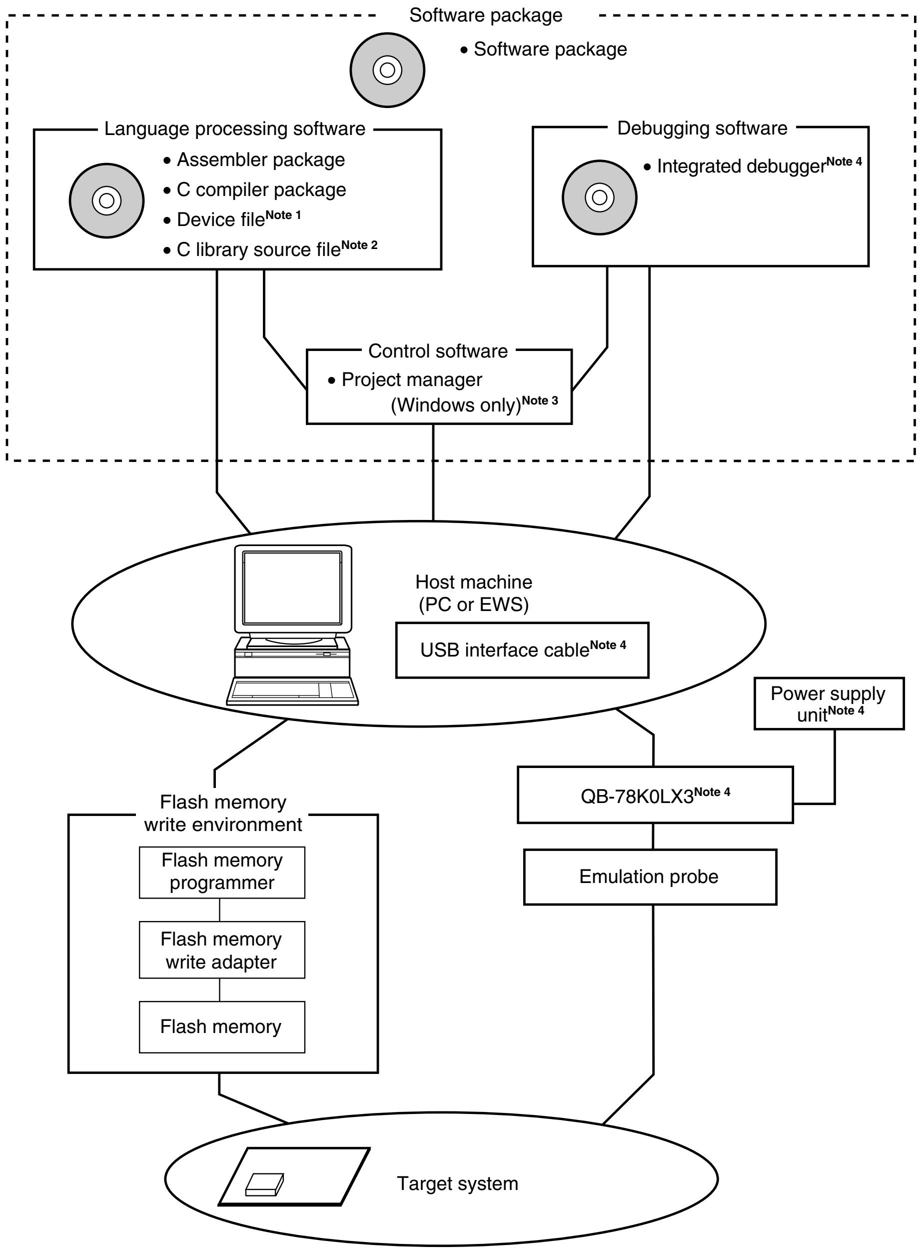
The following development tools are available for the development of systems that employ the 78K0/LF3
 Figure A-1 shows the development tool configuration.

- **Support for PC98-NX series**
  Unless otherwise specified, products supported by IBM PC/AT™ compatibles are compatible with PC98-NX series computers. When using PC98-NX series computers, refer to the explanation for IBM PC/AT compatibles.

- **Windows™**
  Unless otherwise specified, "Windows" means the following OSs.
  - Windows 98
  - Windows NT™
  - Windows 2000
  - Windows XP

Figure A-1. Development Tool Configuration (1/2)

(1) When using the in-circuit emulator QB-78K0LX3



Notes 1. Download the device file (DF780495) for the 78K0/LF3 from the download site for development tools (http://www.necel.com/micro/ods/eng/index.html).

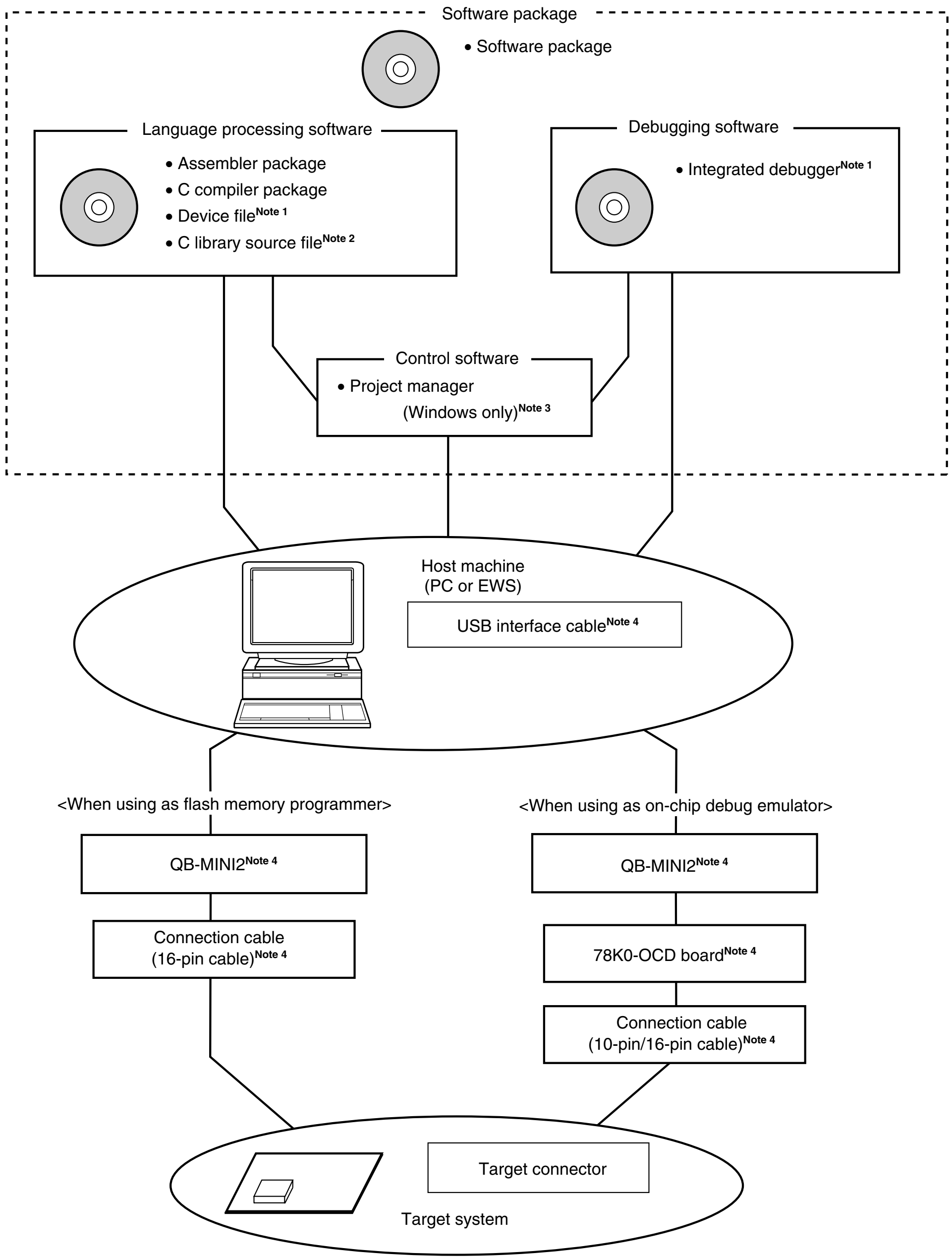
2. The C library source file is not included in the software package.

3. The project manager PM+ is included in the assembler package.
The PM+ is only used for Windows.

4. The QB-78K0LX3 is supplied with the integrated debugger ID78K0-QB, a USB interface cable, a power supply unit, the on-chip debug emulator QB-MINI2, connection cables (10-pin and 16-pin cables), and the 78K0-OCD board. Any other products are sold separately.
Download the software for operating the QB-MINI2 from the download site for development tools (http://www.necel.com/micro/ods/eng/index.html) when using the QB-MINI2.

Figure A-1. Development Tool Configuration (2/2)

(2) When using the on-chip debug emulator with programming function QB-MINI2



Notes 1. Download the device file (DF780495) for the 78K0/LF3 and the integrated debugger ID78K0-QB from the download site for development tools (http://www.necel.com/micro/ods/eng/index.html).

2. The C library source file is not included in the software package.

3. The project manager PM+ is included in the assembler package.
The PM+ is only used for Windows.

4. The QB-MINI2 is supplied with a USB interface cable, connection cables (10-pin and 16–pin cables), and the 78K0-OCD board. Any other products are sold separately.
Download the software for operating the QB-MINI2 from the download site for development tools (http://www.necel.com/micro/ods/eng/index.html).

A.1 Software Package

SP78K0 78K0 microcontroller software package	Development tools (software) common to the 78K0 microcontrollers are combined in this package.
	Part number: μS××××SP78K0

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SP78K0

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

A.2 Language Processing Software

RA78K0 Assembler package	This assembler converts programs written in mnemonics into object codes executable with a microcontroller. This assembler is also provided with functions capable of automatically creating symbol tables and branch instruction optimization. This assembler should be used in combination with a device file (DF780495) (sold separately). **<Precaution when using RA78K0 in PC environment>** This assembler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.
	Part number: μS××××RA78K0
CC78K0 C compiler package	This compiler converts programs written in C language into object codes executable with a microcontroller. This compiler should be used in combination with an assembler package and device file (both sold separately). **<Precaution when using CC78K0 in PC environment>** This C compiler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.
	Part number: μS××××CC78K0
DF780495 [Note 1] Device file	This file contains information peculiar to the device. This device file should be used in combination with a tool (RA78K0, CC78K0, and ID78K0-QB) (all sold separately). The corresponding OS and host machine differ depending on the tool to be used.
	Part number: μS××××DF780495
CC78K0-L [Note 2] C library source file	This is a source file of the functions that configure the object library included in the C compiler package. This file is required to match the object library included in the C compiler package to the user's specifications.
	Part number: μS××××CC78K0-L

Notes 1. The DF780495 can be used in common with the RA78K0, CC78K0, and ID78K0-QB. Download the DF780495 from the download site for development tools (http://www.necel.com/micro/ods/eng/index.html).

2. The CC78K0-L is not included in the software package (SP78K0).

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××RA78K0
μS××××CC78K0
μS××××CC78K0-L

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	
3P17	HP9000 series 700™	HP-UX™ (Rel. 10.10)	
3K17	SPARCstation™	SunOS™ (Rel. 4.1.4) Solaris™ (Rel. 2.5.1)	

μS××××DF780495

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	3.5-inch 2HD FD
BB13		Windows (English version)	

A.3 Control Software

PM+ Project manager	This is control software designed to enable efficient user program development in the Windows environment. All operations used in development of a user program, such as starting the editor, building, and starting the debugger, can be performed from the project manager. **<Caution>** The project manager is included in the assembler package (RA78K0). It can only be used in Windows.

A.4 Flash Memory Writing Tools

A.4.1 When using flash memory programmer PG-FP5 and FL-PR5

PG-FP5, FL-PR5 Flash memory programmer	Flash memory programmer dedicated to microcontrollers with on-chip flash memory.
FA-80GC-GAD-B FA-78F0495GC-GAD-RX FA-80GK-GAK-B FA-78F0495GK-GAK-RX Flash memory writing adapter	Flash memory writing adapter used connected to the flash memory programmer. • FA-80GC-GAD-B, FA-78F0495GC-GAD-RX: For 80pin plastic LQFP (GC-GAD type) • FA-80GK-GAK-B, FA-78F0495GK-GAK-RX: For 80-pin plastic LQFP (GK-GAK type)

Remarks 1. FL-PR5, FA-80GC-GAD-B, FA-78F0495GC-GAD-RX, FA-80GK-GAK-B, and FA-78F0495GK-GAK-RX are products of Naito Densei Machida Mfg. Co., Ltd.
TEL: +81-42-750-4172 Naito Densei Machida Mfg. Co., Ltd.

2. Use the latest version of the flash memory programming adapter.

A.4.2 When using on-chip debug emulator with programming function QB-MINI2

QB-MINI2 On-chip debug emulator with programming function	This is a flash memory programmer dedicated to microcontrollers with on-chip flash memory. It is available also as on-chip debug emulator which serves to debug hardware and software when developing application systems using the 78K0/Lx3. When using this as flash memory programmer, it should be used in combination with a connection cable (16-pin cable) and a USB interface cable that is used to connect the host machine.
Target connector specifications	16-pin general-purpose connector (2.54 mm pitch)

Remarks 1. The QB-MINI2 is supplied with a USB interface cable, connection cables (10-pin and 16-pin cables), and the 78K0-OCD board. The connection cable (10-pin cable) and the 78K0-OCD board are used only when using the on-chip debug function.

2. Download the software for operating the QB-MINI2 from the download site for development tools (http://www.necel.com/micro/ods/eng/index.html).

A.5 Debugging Tools (Hardware)

A.5.1 When using in-circuit emulator QB-78K0LX3

QB-78K0/Lx3 In-circuit emulator	This in-circuit emulator serves to debug hardware and software when developing application systems using the 78K0/Lx3. It supports to the integrated debugger (ID78K0-QB). This emulator should be used in combination with a power supply unit and emulation probe, and the USB is used to connect this emulator to the host machine.
QB-144-CA-01 Check pin adapter	This check pin adapter is used in waveform monitoring using the oscilloscope, etc.
QB-80-EP-01T Emulation probe	This emulation probe is flexible type and used to connect the in-circuit emulator and target system.
QB-80GC-EA-01T, QB-80GK-EA-01T Exchange adapter	This exchange adapter is used to perform pin conversion from the in-circuit emulator to target connector. • QB-80GC-EA-01T: For 80-pin plastic LQFP (GC-GAD type) • QB-80GK-EA-01T: For 80-pin plastic LQFP (GK-GAK type)
QB-80GC-YS-01T, QB-80GK-YS-01T Space adapter	This space adapter is used to adjust the height between the target system and in-circuit emulator. • QB-80GC-YS-01T: For 80-pin plastic LQFP (GC-GAD type) • QB-80GK-YS-01T: For 80-pin plastic LQFP (GK-GAK type)
QB-80GC-YQ-01T, QB-80GK-YQ-01T YQ connector	This YQ connector is used to connect the target connector and exchange adapter. • QB-80GC-YQ-01T: For 80-pin plastic LQFP (GC-GAD type) • QB-80GK-YQ-01T: For 80-pin plastic LQFP (GK-GAK type)
QB-80GC-HQ-01T, QB-80GK-HQ-01T Mount adapter	This mount adapter is used to mount the target device with socket. • QB-80GC-HQ-01T: For 80-pin plastic LQFP (GC-GAD type) • QB-80GK-HQ-01T: For 80-pin plastic LQFP (GK-GAK type)
QB-80GC-NQ-01T, QB-80GK-NQ-01T Target connector	This target connector is used to mount on the target system. • QB-80GC-NQ-01T: For 80-pin plastic LQFP (GC-GAD type) • QB-80GK-NQ-01T: For 80-pin plastic LQFP (GK-GAK type)

Remarks 1. The QB-78K0LX3 is supplied with the integrated debugger ID78K0-QB, a USB interface cable, a power supply unit, the on-chip debug emulator QB-MINI2, connection cables (10-pin and 16-pin cables), and the 78K0-OCD board.
Download the software for operating the QB-MINI2 from the download site for development tools (http://www.necel.com/micro/ods/eng/index.html) when using the QB-MINI2.

2. The packed contents differ depending on the part number, as follows.

Part Number \ Packed Contents	In-Circuit Emulator	Emulation Probe	Exchange Adapter	YQ Connector	Target Connector
QB-78K0LX3 -ZZZ	QB-78K0LX3	None			
QB-78K0LX3-T80GC		QB-80-EP-01T	QB-80GC-EA-01T	QB-80GC-YQ-01T	QB-80GC-NQ-01T
QB-78K0LX3-T80GK			QB-80GK-EA-01T	QB-80GK-YQ-01T	QB-80GK-NQ-01T

A.5.2 When using on-chip debug emulator with programming function QB-MINI2

QB-MINI2 On-chip debug emulator with programming function	This on-chip debug emulator serves to debug hardware and software when developing application systems using the 78K0/Lx3. It is available also as flash memory programmer dedicated to microcontrollers with on-chip flash memory. When using this as on-chip debug emulator, it should be used in combination with a connection cable (10-pin cable or 16-pin cable), a USB interface cable that is used to connect the host machine, and the 78K0-OCD board.
Target connector specifications	10-pin general-purpose connector (2.54 mm pitch) or 16-pin general-purpose connector (2.54 mm pitch)

Remarks 1. The QB-MINI2 is supplied with a USB interface cable, connection cables (10-pin and 16-pin cables), and the 78K0-OCD board. The connection cable (10-pin cable) and the 78K0-OCD board are used only when using the on-chip debug function.

2. Download the software for operating the QB-MINI2 from the download site for development tools (http://www.necel.com/micro/ods/eng/index.html).

A.6 Debugging Tools (Software)

ID78K0-QB Integrated debugger	This debugger supports the in-circuit emulators for the 78K0 microcontrollers. The ID78K0-QB is a 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. It should be used in combination with the device file (sold separately).
	Part number: μS××××ID78K0-QB

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××ID78K0-QB

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

APPENDIX B REVISION HISTORY

B.1 Major Revisions in This Edition

(1/3)

Page	Description	Classification
Throughout	• Addition of PG-FP5 and FL-PR5 • Addition of explanation for segment key scan function	(b, c)
CHAPTER 1 OUTLINE		
p. 17	Addition of **Note 2** to Table of ROM, RAM capacities **in 1.1 Features**	(d)
pp. 20 to 22	Addition of **Remark 2** to **1.4 Pin Configuration (Top View)**	(c)
pp. 25 to 27	Addition of **Note** to table of function list in **1.5 78K0/Lx3 Microcontroller Series Lineup**	(c)
p. 30	Addition of **Note 4** to **1.7 Outline of Functions (μPD78F047x)**	(c)
p. 33	Addition of **Note 4** to **1.8 Outline of Functions (μPD78F048x)**	(c)
p. 36	Addition of **Note 5** to **1.9 Outline of Functions (μPD78F049x)**	(c)
CHAPTER 3 CPU ARCHITECTURE		
pp. 59, 61, 63, 65, 67	Addition of **Note 3** to **Memory Map** of **Figure 3-2**, **Figure 3-4**, **Figure 3-6**, **Figure 3-8**, **and Figure 3-10**	(b)
pp. 73, 75, 77, 79, 81	Addition of **Note** to **Correspondence Between Data Memory and Addressing** of **Figure 3-12**, **Figure 3-14**, **Figure 3-16**, **Figure 3-18**, **and Figure 3-20**	(b)
p. 95	Change of Illustration in **3.3.3 Table indirect addressing**	(c)
CHAPTER 4 PORT FUNCTIONS		
p. 116	Change of **Figure 4-8. Block Diagram of P20 to P27**	(a)
p. 138	Change of **Figure 4-28. Format of Port Register** and addition of **Notes 1, 2**, and **3**	(b, c)
p. 139	Addition of **Note** to **Figure 4-29. Format of Pull-up Resistor Option Register**	(c)
CHAPTER 5 CLOCK GENERATOR		
p. 160	Change of **Figure 5-9. Format of Internal High-speed Oscillation Trimming Register (HIOTRM)**	(b)
p. 166	Change of **Figure 5-13. Clock Generator Operation When Power Supply Voltage Is Turned On**	(b)
p. 167	Change of **Caution 1** in **Figure 5-14. Clock Generator Operation When Power Supply Voltage Is Turned On**	(b)
CHAPTER 6 16-BIT TIMER/EVENT COUNTER 00		
p. 197	Addition of **Note 2** to **Figure 6-10. Format of Input Switch Control Register (ISC)**	(c)
p. 232	Modification of **Figure 6-42. (f) 16-bit capture/compare register 000 (CR000)**	(a)
p. 259	Addition of **6.6 (12) Reading of 16-bit timer counter 00 (TM00)**	(c)
CHAPTER 7 8-BIT TIMER/EVENT COUNTERS 50, 51, AND 52		
p. 270	Addition of **Note 2** to **Figure 7-12. Format of Input Switch Control Register (ISC)**	(c)
p. 280	Change of **7.5 (2) Cautions for 16-bit timer/event counter 00 count up during external 24-bit event counter operation**	(c)
p. 281	Addition of **7.5 (3) Reading of 8-bit timer counter 5n (TM5n)**	(c)
CHAPTER 8 8-BIT TIMERS H0, H1, AND H2		
p. 284	Change of **Figure 8-2. Block Diagram of 8-Bit Timer H1**	(a)
p. 288	Change of **Figure 8-6. Format of 8-Bit Timer H Mode Register 0 (TMHMD0)**	(c)
p. 310	Addition of **8.4.4 Control of number of carrier clocks by timer 51 counter**	(b, c)

Remark "Classification" in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note, (d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

(2/3)

Page	Description	Classification
CHAPTER 9 REAL-TIME COUNTER		
p. 316	Change of **Caution** in **Figure 9-4. Format of Real-Time Counter Control Register 1 (RTCC1)**	(b, c)
p. 320	Change of **Table 9-2. Displayed Time Digits**	(c)
p. 322	Addition of **Caution** to **Figure 9-11. Format of Week Count Register (WEEK)**	(c)
p. 324	Change of **(13) Watch error correction register (SUBCUD)**	(c)
p. 325	Addition of explanation to **(15) Alarm hour register (ALARMWH)**	(c)
p. 327	Addition of **Note** to **Figure 9-18. Procedure for Starting Operation of Real-Time Counter**	(c)
p. 328	Addition of **9.4.2 Shifting to STOP mode after starting operation**	(c)
p. 332	Addition of **9.4.5 1 Hz output of real-time counter**	(c)
p. 332	Addition of **9.4.6 32.768 kHz output of real-time counter**	(c)
p. 333	Addition of **9.4.7 512 Hz, 16.384 kHz output of real-time counter**	(c)
p. 334	Addition of **9.4.8 Example of watch error correction of real-time counter**	(c)
CHAPTER 10 WATCHDOG TIMER		
p. 345	Change of **Remark** in **10.4.3 Setting window open period of watchdog timer**	(a)
CHAPTER 12 10-BIT SUCCESSIVE APPROXIMATION TYPE A/D CONVERTER (μPD78F048x and 78F049x only)		
p. 356	Change of **Caution 1** in **Table 12-2. A/D Conversion Time Selection**	(c)
p. 360	Modification of **Figure 12-9. Format of A/D Port Configuration Register 0 (ADPC0)**	(a)
p. 372	Change of **12.6 (11) Internal equivalent circuit**	(b)
CHAPTER 13 16-BIT ΔΣ TYPE A/D CONVERTER (μPD78F049x only)		
pp. 373 to 396	Full-scale revision of chapter	(b, c)
CHAPTER 14 SERIAL INTERFACE UART0		
p. 416	Change of **Table 14-5. Set Data of Baud Rate Generator**	(b)
CHAPTER 18 LCD CONTROLLER/DRIVER		
pp. 509 to 570	Full-scale revision of chapter	(b, c)
CHAPTER 20 REMOTE CONTROLLER RECEIVER		
pp. 596 to 627	Full-scale revision of chapter	(b, c)
CHAPTER 21 INTERRUPT FUNCTIONS		
p. 635	Change of **Figure 21-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H)**	(a)
p. 637	Change of **Figure 21-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H)**	(a)
p. 638	Change of **Figure 21-4. Format of Priority Specification Flag Registers (PR0L, PR0H, PR1L, PR1H)**	(a)
CHAPTER 22 KEY INTERRUPT FUNCTION		
p. 649	Addition of **Caution 5** to **Figure 22-2. Format of Key Return Mode Register (KRM)**	(c)
CHAPTER 23 STANDBY FUNCTION		
p. 657	Change of **Figure 23-4. HALT Mode Release by Reset**	(c)
pp. 659, 660	Change of **Table 23-3. Operating Statuses in STOP Mode**, **Note 2**, and **Caution 3** and addition of **Caution 4**	(c)
p. 663	Change of **Figure 23-7. STOP Mode Release by Reset**	(c)

Remark "Classification" in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note, (d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

(3/3)

Page	Description	Classification
CHAPTER 24 RESET FUNCTION		
p. 666	Change of **Figure 24-2. Timing of Reset by $\overline{\text{RESET}}$ Input**	(c)
p. 666	Change of **Figure 24-3. Timing of Reset Due to Watchdog Timer Overflow**	(c)
p. 667	Change of **Figure 24-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input**	(c)
CHAPTER 25 POWER-ON-CLEAR CIRCUIT		
p. 676	Change of **Figure 25-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (1)**	(b)
p. 677	Change of **Caution 2** in **Figure 25-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (2)**	(b, c)
CHAPTER 26 LOW-VOLTAGE DETECTOR		
p. 697	Change of **Figure 26-9. Example of Software Processing After Reset Release (2/2)**	(c)
CHAPTER 28 FLASH MEMORY		
p. 705	Change of **Note 2** in and addition of **Caution** to **Table 28-3. Wiring Between 78K0/LF3 and Dedicated Flash memory programmer**	(c)
p. 709	Change of **Note 1** in **Table 28-4. Pin Connection**	(c)
p. 712	Change of **Caution 2** in **28.6.6 Other signal pins**	(c)
p. 713	Change of **Figure 28-12. Flash Memory Manipulation Procedure**	(b)
p. 714	Change of **Table 28-7. Communication Modes**	(b)
pp. 718 to 720	Addition of **28.9 Processing Time for Each Command When PG-FP5 Is Used (Reference)**	(b, c)
pp. 721 to 730	Revision of **28.10 Flash Memory Programming by Self-Programming**	(b, c)
CHAPTER 29 ON-CHIP DEBUG FUNCTION		
pp. 731, 732	Full-scale revision of chapter	(c)
CHAPTER 31 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)		
p. 746	Addition of **Caution**	(c)
p. 750	Addition of **Recommended Oscillator Constants**	(b)
p. 754	Change of **Supply current value** and **Notes 1**, **6** in and addition of **Note 5** to **DC Characteristics**	(b)
p. 755	Change of ΔΣ type A/D converter operating current value in **DC Characteristics**	(b)
p. 765	Change of **16-bit ΔΣ type A/D Converter Characteristics**	(b)
p. 766	Change of **Note 3** in **LCD Characteristics**	(b)
p. 769	Change of **Basic characteristics** in **Flash Memory Programming Characteristics** and addition of **Notes 1**, **4**	(b, c)
CHAPTER 33 RECOMMENDED SOLDERING CONDITIONS		
p. 772	Addition of chapter	(b, c)
APPENDIX A DEVELOPMENT TOOLS		
pp. 775 to 782	Addition of appendix A	(c)

Remark "Classification" in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note, (d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

B.2 Revision History up to Previous Editions

Here is the revision history of the preceding editions. Chapter indicates the chapter of each edition.

(1/3)

Edition	Description	Applied to:
2nd edition	Addition of **Note** to **1.3 Ordering Information**	**CHAPTER 1 OUTLINE**
	Addition of **Caution 3** to **(1)** in **1.4 Pin Configuration (Top View)**	
	Change of **Table 2-2. Pin I/O Circuit Types**	**CHAPTER 2 PIN FUNCTIONS**
	Modification of **Figure 3-9. Memory Map (μPD78F0475, 78F0485)**	**CHAPTER 3 CPU ARCHITECTURE**
	Modification of **Figure 3-10. Memory Map (μPD78F0495)**	
	Modification of **Figure 4-5. Block Diagram of P13**	**CHAPTER 4 PORT FUNCTIONS**
	Modification of **Figure 4-6. Block Diagram of P16**	
	Modification of **Figure 4-29. Format of Pull-up Resistor Option Register**	
	Change of **Table 4-5. Settings of PFALL, PF2, PF1, ISC, Port Mode Register, and Output Latch When Using Alternate Function**	
	Change of **5.3 (9) Internal high-speed oscillation trimming register (HIOTRM)**	**CHAPTER 5 CLOCK GENERATOR**
	Addition of explanation to **Table 5-4. Clocks Supplied to CPU and Peripheral Hardware, and Register Setting**	
	Addition of explanation to **Table 5-6. Changing CPU Clock**	
	• TO00 pin output → TO00 output • Addition of TO00 output in block diagram	**CHAPTER 6 16-BIT TIMER/EVENT COUNTER 00**
	Addition of explanation to **Figure 6-8 Format of 16-bit Timer Output Control Register 00 (TOC00)**	
	Addition of **Notes 1** and **2** to and change of **Note 3** in **Figure 6-9 Format of Prescaler Mode Register 00 (PRM00)**	
	Change of **Figure 6-54. Configuration Diagram of External 24-bit Event Counter**	
	Change of **Figure 6-55. Operation Timing of External 24-bit Event Counter**	
	• TO50 pin output → TO50 output, TO51 pin output → TO51 output • Addition of TO50, TO51 output in block diagram	**CHAPTER 7 8-BIT TIMER/EVENT COUNTERS 50, 51, AND 52**
	Addition of **Notes 1** and **2** to **Figure 7-6 Format of Timer Clock Selection Register 50 (TCL50)**	
	Addition of **Notes 1** and **2** to **Figure 7-7 Format of Timer Clock Selection Register 51 (TCL51)**	
	Addition of **Notes 1** and **2** to **Figure 7-8 Format of Timer Clock Selection Register 52 (TCL52)**	
	Chenge of **Caution 3** in and addition of **Caution 4** to **Figure 7-9 Format of 8-bit Timer Mode Control Register 50 (TMC50)** and **Figure 7-10 Format of 8-bit Timer Mode Control Register 51 (TMC51)**	
	Full-scale revision	**CHAPTER 8 8-BIT TIMERS H0, H1, AND H2**
	Addition of **Note 1** to **Figure 11-2 Format of Clock Output Selection Register (CKS)**	**CHAPTER 11 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER**

(2/3)

Edition	Description	Applied to:
2nd edition	Change of **Figure 14-1 Block Diagram of Serial Interface UART0**	**CHAPTER 14 SERIAL INTERFACE UART0**
	Addition of **Note 1** to **Figure 14-4 Format of Baud Rate Generator Control Register 0 (BRGC0)**	
	Change of **14.3 (5) Port mode register 1 (PM1)**	
	Change of **Table 14-2. Relationship Between Register Settings and Pins**	
	Addition of **Notes 1** and **2** to **Table 14-4 Set Value of TPS01 and TPS00**	
	Change of explanation in **15.1 Functions of Serial Interface UART6**	**CHAPTER 15 SERIAL INTERFACE UART6**
	Change of **Figure 15-4 Block Diagram of Serial Interface UART6**	
	Addition of **Notes 1** and **2** to **Figure 15-8 Format of Clock Selection Register 6 (CKSR6)**	
	Change of **15.3 (9) Port mode register 1 (PM1)**	
	Change of **(a)** in **Table 15-2. Relationship Between Register Settings and Pins**	
	Addition of **Notes 1**, **2**, and **3** to **Table 15-4 Set Value of TPS63 to TPS60**	
	Change of **Figure 16-1. Block Diagram of Serial Interface CSI10**	**CHAPTER 16 SERIAL INTERFACE CSI10**
	Addition of **Notes 1** and **2** to **Figure 16-3 Format of Serial Clock Selection Register 10 (CSIC10)**	
	Change of **Figure 17-1. Block Diagram of Serial Interface CSIA0**	**CHAPTER 17 SERIAL INTERFACE CSIA0**
	Addition of **Notes 2** and **3** to **Figure 17-3. Format of Serial Status Register 0 (CSIS0)**	
	Change of **Note 2** in **18.3 (2) LCD display mode register (LCDM)**	**CHAPTER 18 LCD CONTROLLER/DRIVER**
	Change of **Note** in **18.4 Setting LCD Controller/Driver**	
	Addition of **Notes 1** and **2** to **Figure 19-5. Format of MCG Control Register 1 (MC0CTL1)**	**CHAPTER 19 MANCHESTER CODE GENERATOR**
	Addition of **Notes 1** and **2** to **19.4.2 (b) MCG control register 1 (MC0CTL1)**	
	Change of **Note 4** in **Table 21-1. Interrupt Source List**	**CHAPTER 21 INTERRUPT FUNCTIONS**
	Change of **Note 2** in and addition of **Notes 4**, **5**, and **6** to **Table 21-2 Flags Corresponding to Interrupt Request Sources**	
	Change of **Caution 1** in **Figure 22-2 Format of Key Return Mode Register (KRM)**	**CHAPTER 22 KEY INTERRUPT FUNCTION**
	Addition of explanation to **Caution 3** in **23.1.1 Standby function**	**CHAPTER 23 STANDBY FUNCTION**
	Change of explanation in **26.3 (1) Low-voltage detection register (LVIM)**	**CHAPTER 26 LOW-VOLTAGE DETECTOR**
	Addition of **Notes 1** and **4** and **Cautions 3** and **4** to **Figure 26-2 Format of Low-Voltage Detection Register (LVIM)**	
	Change of explanation in **26.3 (2) Low-voltage detection level selection register (LVIS)**	
	Addition of **Note** and **Caution 4** to **Figure 26-3 Format of Low-Voltage Detection Level Selection Register (LVIS)**	
	Addition of **Note 3** to **Figure 26-7 Timing of Low-Voltage Detector Interrupt Signal Generation (Detects Level of Supply Voltage (V_{DD}))**	
	Addition of **Note 3** to **Figure 26-8 Timing of Low-Voltage Detector Interrupt Signal Generation (Detects Level of Input Voltage from External Input Pin (EXLVI))**	
	Change of **Figure 26-9 Example of Software Processing After Reset Release**	

(3/3)

Edition	Description	Applied to:
2nd edition	Change of and addition of **Remark** to **Figure 28-17 Boot Swap Function**	**CHAPTER 28 FLASH MEMORY**
	Change of **Figure 28-18 Example of Executing Boot Swapping**	
	Change to formal spec from target spec	**CHAPTER 31 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)**
	Addition of explanation to **Table 33-1. Registers That Generate Wait and Number of CPU Wait Clocks**	**CHAPTER 33 CAUTIONS FOR WAIT**
	Change of **Table 33-2. RAM Accesses That Generate Wait and Number of CPU Wait Clocks**	
	Addition of appendix	**APPENDIX REVISION HISTORY**

For further information, please contact:

NEC Electronics Corporation
1753, Shimonumabe, Nakahara-ku,
Kawasaki, Kanagawa 211-8668,
Japan
Tel: 044-435-5111
http://www.necel.com/

[America]

NEC Electronics America, Inc.
2880 Scott Blvd.
Santa Clara, CA 95050-2554, U.S.A.
Tel: 408-588-6000
800-366-9782
http://www.am.necel.com/

[Europe]

NEC Electronics (Europe) GmbH
Arcadiastrasse 10
40472 Düsseldorf, Germany
Tel: 0211-65030
http://www.eu.necel.com/

Hanover Office
Podbielskistrasse 166 B
30177 Hannover
Tel: 0 511 33 40 2-0

Munich Office
Werner-Eckert-Strasse 9
81829 München
Tel: 0 89 92 10 03-0

Stuttgart Office
Industriestrasse 3
70565 Stuttgart
Tel: 0 711 99 01 0-0

United Kingdom Branch
Cygnus House, Sunrise Parkway
Linford Wood, Milton Keynes
MK14 6NP, U.K.
Tel: 01908-691-133

Succursale Française
9, rue Paul Dautier, B.P. 52
78142 Velizy-Villacoublay Cédex
France
Tel: 01-3067-5800

Sucursal en España
Juan Esplandiu, 15
28007 Madrid, Spain
Tel: 091-504-2787

Tyskland Filial
Täby Centrum
Entrance S (7th floor)
18322 Täby, Sweden
Tel: 08 638 72 00

Filiale Italiana
Via Fabio Filzi, 25/A
20124 Milano, Italy
Tel: 02-667541

Branch The Netherlands
Steijgerweg 6
5616 HS Eindhoven
The Netherlands
Tel: 040 265 40 10

[Asia & Oceania]

NEC Electronics (China) Co., Ltd
7th Floor, Quantum Plaza, No. 27 ZhiChunLu Haidian
District, Beijing 100083, P.R.China
Tel: 010-8235-1155
http://www.cn.necel.com/

Shanghai Branch
Room 2509-2510, Bank of China Tower,
200 Yincheng Road Central,
Pudong New Area, Shanghai, P.R.China P.C:200120
Tel:021-5888-5400
http://www.cn.necel.com/

Shenzhen Branch
Unit 01, 39/F, Excellence Times Square Building,
No. 4068 Yi Tian Road, Futian District, Shenzhen,
P.R.China P.C:518048
Tel:0755-8282-9800
http://www.cn.necel.com/

NEC Electronics Hong Kong Ltd.
Unit 1601-1613, 16/F., Tower 2, Grand Century Place,
193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: 2886-9318
http://www.hk.necel.com/

NEC Electronics Taiwan Ltd.
7F, No. 363 Fu Shing North Road
Taipei, Taiwan, R. O. C.
Tel: 02-8175-9600
http://www.tw.necel.com/

NEC Electronics Singapore Pte. Ltd.
238A Thomson Road,
#12-08 Novena Square,
Singapore 307684
Tel: 6253-8311
http://www.sg.necel.com/

NEC Electronics Korea Ltd.
11F., Samik Lavied'or Bldg., 720-2,
Yeoksam-Dong, Kangnam-Ku,
Seoul, 135-080, Korea
Tel: 02-558-3737
http://www.kr.necel.com/

G0706