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.

RENESAS

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.

Preliminary User's Manual

78K0/Dx2

8-Bit Single-Chip Microcontrollers

78K0/DE2:

μPD78F0836(A)
μPD78F0836(A2)
μPD78F0837(A)
μPD78F0837(A2)
μPD78F0844(A)
μPD78F0844(A2)
μPD78F0845(A)
μPD78F0845(A2)

78K0/DF2:

μPD78F0838(A)
μPD78F0838(A2)
μPD78F0839(A)
μPD78F0839(A2)
μPD78F0840(A)
μPD78F0840(A2)
μPD78F0841(A)
μPD78F0841(A2)
μPD78F0842(A)
μPD78F0842(A2)
μPD78F0843(A)
μPD78F0843(A2)
μPD78F0846(A)
μPD78F0846(A2)
μPD78F0847(A)
μPD78F0847(A2)
μPD78F0848(A)
μPD78F0848(A2)
μPD78F0849(A)
μPD78F0849(A2)

Document No. U19748EJ1V0UD00 (1st edition)
Date Published September 2009 NS

[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.

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

- **The information contained in this document is being issued in advance of the production cycle for the product. The parameters for the product may change before final production or NEC Electronics Corporation, at its own discretion, may withdraw the product prior to its production.**
- **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 and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. In addition, NEC Electronics products are not taken measures to prevent radioactive rays in the product design. When customers use NEC Electronics products with their products, customers shall, on their own responsibility, incorporate sufficient safety measures such as redundancy, fire-containment and anti-failure features to their products in order to avoid risks of the damages to property (including public or social property) or injury (including death) to persons, as the result of defects of NEC Electronics products.
- 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).

M5D0904E

INTRODUCTION

Readers

This manual is intended for user engineers who wish to understand the functions of the conventional-specification products of the 78K0/Dx2 and design and develop application systems and programs for these devices.
The target products are as follows.

<R> 78K0/DE2: μPD78F0836(A), 78F0837(A), 78F0844(A), 78F0845(A),
78F0836(A2), 78F0837(A2), 78F0844(A2), 78F0845(A2)
78K0/DF2: μPD78F0838(A), 78F0839(A), 78F0840(A), 78F0841(A), 78F0842(A),
78F0843(A), 78F0846(A), 78F0847(A), 78F0848(A), 78F0849(A),
78F0838(A2), 78F0839(A2), 78F0840(A2), 78F0841(A2), 78F0842(A2),
78F0843(A2), 78F0846(A2), 78F0847(A2), 78F0848(A2), 78F0849(A2)

Purpose

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

Organization

The 78K0/Dx2 manual is separated into two parts: this manual and the instructions edition (common to the 78K0 Series).

78K0/Dx2 **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.

<R>

- When using this manual as the manual for (A) and (A2) grade products:
  → Only the quality grade differs between (A) grade products and (A2) grade products. Read the part number as follows.
  - μPD78F0836 → μPD78F0836(A), 78F0836(A2)
  - μPD78F0837 → μPD78F0837(A), 78F0837(A2)
  - μPD78F0838 → μPD78F0838(A), 78F0838(A2)
  - μPD78F0839 → μPD78F0839(A), 78F0839(A2)
  - μPD78F0840 → μPD78F0840(A), 78F0840(A2)
  - μPD78F0841 → μPD78F0841(A), 78F0841(A2)
  - μPD78F0842 → μPD78F0842(A), 78F0842(A2)
  - μPD78F0843 → μPD78F0843(A), 78F0843(A2)
  - μPD78F0844 → μPD78F0844(A), 78F0844(A2)
  - μPD78F0845 → μPD78F0845(A), 78F0845(A2)
  - μPD78F0846 → μPD78F0846(A), 78F0846(A2)
  - μPD78F0847 → μPD78F0847(A), 78F0847(A2)
  - μPD78F0848 → μPD78F0848(A), 78F0848(A2)
  - μPD78F0849 → μPD78F0849(A), 78F0849(A2)

- 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 brackets, the bit name is defined as a reserved word in the assembler, and is already defined in the header file named sfrbit.h in the C compiler.
- To check the details of a register when you know the register name:
  → Refer to **APPENDIX C REGISTER INDEX**.
- 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/Dx2 User’s Manual	This manual
78K/0 Series Instructions User’s Manual	U12326E

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

<R>

Document Name	Document No.
QB-78K0DX2 In-Circuit Emulator	U19952E
QB-78K0MINI On-Chip Debug Emulator	U17029E
QB-MINI2 On-Chip Debug Emulator with Programming Function	U18371E

Documents Related to Flash Memory Programming

Document Name		Document No.
PG-FP5 Flash Memory Programmer User’s Manual		U18865E
PG-FP4 Flash Memory Programmer User’s Manual		U15260E
QB-Programmer Programming GUI Operation User’s Manual	Operation	U18527E

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

Documents Related to Development Tools (Software)

Document Name		Document No.
RA78K0 Ver.3.80 Assembler Package User's Manual[Note 1]	Operation	U17199E
	Language	U17198E
	Structured Assembly Language	U17197E
78K0 Assembler Package RA78K0 Ver.4.01 Operating Precautions (Notification Document)[Note 1]		ZUD-CD-07-0181-E
CC78K0 Ver.3.70 C Compiler User's Manual[Note 2]	Operation	U17201E
	Language	U17200E
78K0 C Compiler CC78K0 Ver. 4.00 Operating Precautions (Notification Document)[Note 2]		ZUD-CD-07-0103-E
SM+ System Simulator User's Manual	Operation	U18601E
	User Open Interface	U18212E
ID78K0-QB Ver.2.94 Integrated Debugger User's Manual	Operation	U18330E
ID78K0-QB Ver.3.00 Integrated Debugger User's Manual	Operation	U18492E
PM plus Ver.5.20[Note 3] User's Manual		U16934E
PM+ Ver.6.30[Note 4] User's Manual		U18416E

Notes 1. This document is installed into the PC together with the tool when installing RA78K0 Ver. 4.01. For descriptions not included in "78K0 Assembler Package RA78K0 Ver. 4.01 Operating Precautions", refer to the user's manual of RA78K0 Ver. 3.80.

2. This document is installed into the PC together with the tool when installing CC78K0 Ver. 4.00. For descriptions not included in "78K0 C Compiler CC78K0 Ver. 4.00 Operating Precautions", refer to the user's manual of CC78K0 Ver. 3.70.

3. PM plus Ver. 5.20 is the integrated development environment included with RA78K0 Ver. 3.80.

4. PM+ Ver. 6.30 is the integrated development environment included with RA78K0 Ver. 4.01. Software tool (assembler, C compiler, debugger, and simulator) products of different versions can be managed.

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

CHAPTER 1 OUTLINE

1.1 Features

- ○ Minimum instruction execution time can be changed from high speed (0.1 μs: @ 20 MHz operation with highspeed system clock) to ultra low-speed (114 μs: @ 32.768 kHz operation with subsystem clock)
- ○ General-purpose register: 8 bits × 32 registers (8 bits × 8 registers × 4 banks)
- ○ Internal CAN controller (for μPD78F0844, 78F0845, 78F0846, 78F0847, 78F0848, and 78F0849 only)
- ○ Stepper motor controller/driver with zero point detection (ZPD):
  2 channels (for 78K0/DE2) / 0 / 2 / 4 channels (for 78K0/DF2)
- ○ LCD controller/driver (seg × com): 24 × 4 (for 78K0/DE2) / 28 × 4 / 32 × 4 / 40 × 4 (for 78K0/DF2)
- ○ ROM, RAM capacities

Item Part Number		CAN [channel]	Stepper Motor Controller /driver [channels]	LCD controller/driver (seg × com)		Program Memory (ROM) [KB]		Data Memory [bytes]	
78K0/DE2 (64 pins)	78K0/DF2 (80 pins)			78K0/DE2	78K0/DF2			Internal High-speed RAM Note	Internal Expansion RAM Note
–	μPD78F0838	–	0	–	40 × 4	Flash memory Note	24	1024	1024
–	μPD78F0839						48		2048
μPD78F0836	μPD78F0840		2	24 × 4	32 × 4		24		1024
μPD78F0837	μPD78F0841						48		2048
–	μPD78F0842		4	–	28 × 4		24		1024
–	μPD78F0843						48		2048
<R> μPD78F0844	μPD78F0846	1	2	24 × 4	32 × 4		32		1024
μPD78F0845	μPD78F0847						60		2048
–	μPD78F0848		4	–	28 × 4		32		1024
–	μPD78F0849						60		2048

Note 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).

- ○ 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)
- ○ Short startup is possible via the CPU default start using the on-chip internal high-speed oscillator
- ○ On-chip watchdog timer (operable with on-chip internal low-speed oscillator clock)
- ○ On-chip sound generator
- ○ On-chip multiplier/divider
- ○ On-chip clock output/buzzer output controller
- ○ I/O ports: 47 (for 78K0/DE2) / 63 (for 78K0/DF2) (N-ch open drain: 2)
- ○ Timer: 9 channels

○ Serial interface: 3 (for 78K0/DE2) / 5 (for 78K0/DF2) channels
(CAN: 1 channel, UART (LIN (Local Interconnect Network)-bus supported): 1 channel (78K0/DF2 only), CSI/UART[Note]: 1 channel, CSI: 1 channel (78K0/DF2 only), I^2C: 1 channel)

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

○ 10-bit resolution A/D converter: 4 (for 78K0/DE2) / 8 (for 78K0/DF2) channels

○ Supply voltage: V_{DD} = 4.0 to 5.5 V when 20 MHz, V_{DD} = 2.7 to 5.5 V when 10 MHz
(with internal high-speed oscillator clock or subsystem clock: V_{DD} = 2.7 to 5.5 V)

<R> ○ Operating ambient temperature: T_A = –40 to +85°C ((A) grade products), –40 to +105°C ((A2) grade products)

Caution The 78K0/Dx2 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.

1.2 Applications

○ Automotive dashboard control

1.3 Ordering Information

- **Flash memory version**

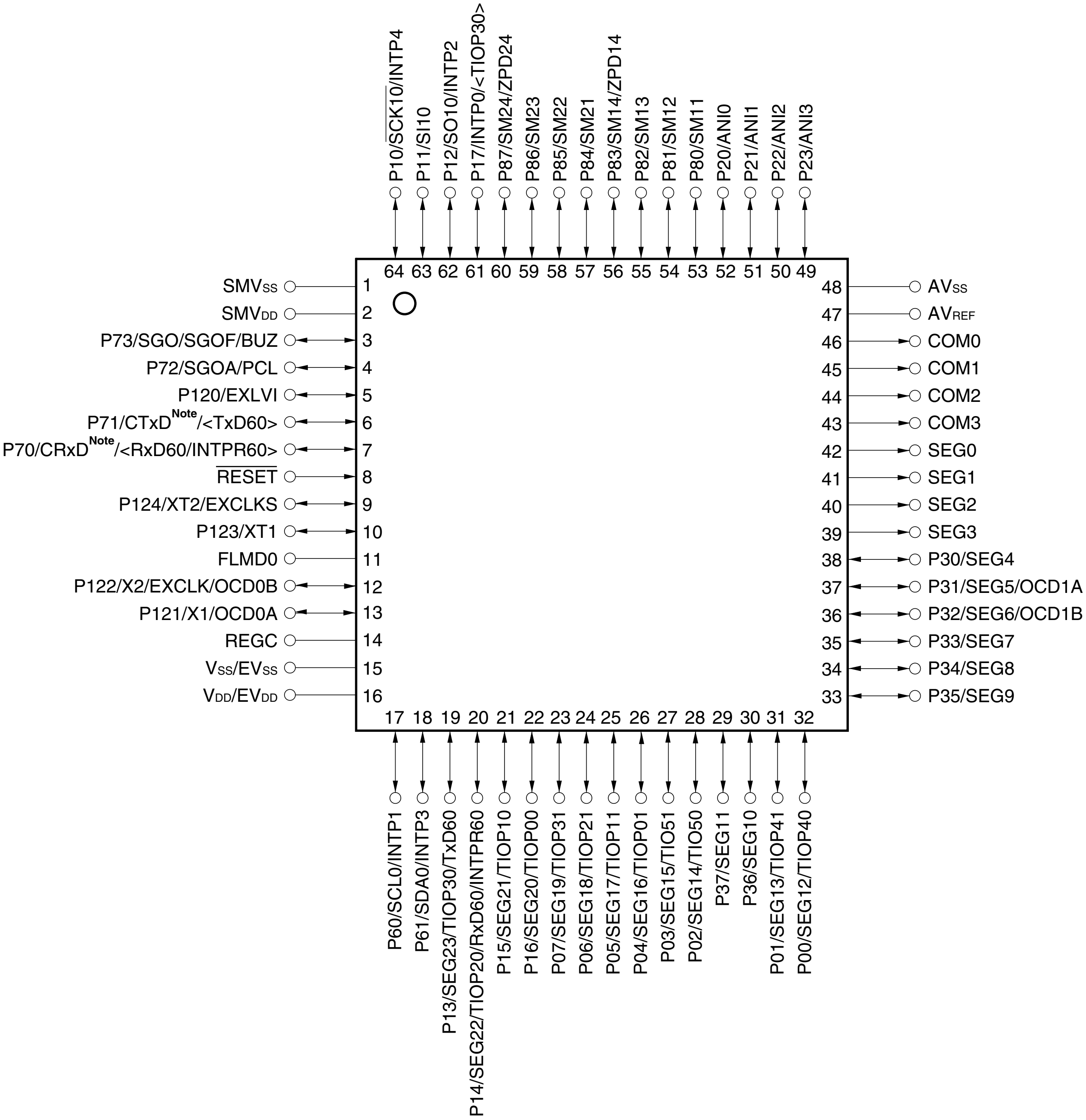
<R>

Part Number	Package	Quality Grade
μPD78F0836GBA-GAH-G	64-pin plastic LQFP (fine pitch) (10 × 10)	Special
μPD78F0836GBA2-GAH-G	64-pin plastic LQFP (fine pitch) (10 × 10)	Special
μPD78F0837GBA-GAH-G	64-pin plastic LQFP (fine pitch) (10 × 10)	Special
μPD78F0837GBA2-GAH-G	64-pin plastic LQFP (fine pitch) (10 × 10)	Special
μPD78F0838GKA-GAK-G	80-pin plastic LQFP (fine pitch) (12 × 12)	Special
μPD78F0838GKA2-GAK-G	80-pin plastic LQFP (fine pitch) (12 × 12)	Special
μPD78F0839GKA-GAK-G	80-pin plastic LQFP (fine pitch) (12 × 12)	Special
μPD78F0839GKA2-GAK-G	80-pin plastic LQFP (fine pitch) (12 × 12)	Special
μPD78F0840GKA-GAK-G	80-pin plastic LQFP (fine pitch) (12 × 12)	Special
μPD78F0840GKA2-GAK-G	80-pin plastic LQFP (fine pitch) (12 × 12)	Special
μPD78F0841GKA-GAK-G	80-pin plastic LQFP (fine pitch) (12 × 12)	Special
μPD78F0841GKA2-GAK-G	80-pin plastic LQFP (fine pitch) (12 × 12)	Special
μPD78F0842GKA-GAK-G	80-pin plastic LQFP (fine pitch) (12 × 12)	Special
μPD78F0842GKA2-GAK-G	80-pin plastic LQFP (fine pitch) (12 × 12)	Special
μPD78F0843GKA-GAK-G	80-pin plastic LQFP (fine pitch) (12 × 12)	Special
μPD78F0843GKA2-GAK-G	80-pin plastic LQFP (fine pitch) (12 × 12)	Special
μPD78F0844GBA-GAH-G	64-pin plastic LQFP (fine pitch) (10 × 10)	Special
μPD78F0844GBA2-GAH-G	64-pin plastic LQFP (fine pitch) (10 × 10)	Special
μPD78F0845GBA-GAH-G	64-pin plastic LQFP (fine pitch) (10 × 10)	Special
μPD78F0845GBA2-GAH-G	64-pin plastic LQFP (fine pitch) (10 × 10)	Special
μPD78F0846GKA-GAK-G	80-pin plastic LQFP (fine pitch) (12 × 12)	Special
μPD78F0846GKA2-GAK-G	80-pin plastic LQFP (fine pitch) (12 × 12)	Special
μPD78F0847GKA-GAK-G	80-pin plastic LQFP (fine pitch) (12 × 12)	Special
μPD78F0847GKA2-GAK-G	80-pin plastic LQFP (fine pitch) (12 × 12)	Special
μPD78F0848GKA-GAK-G	80-pin plastic LQFP (fine pitch) (12 × 12)	Special
μPD78F0848GKA2-GAK-G	80-pin plastic LQFP (fine pitch) (12 × 12)	Special
μPD78F0849GKA-GAK-G	80-pin plastic LQFP (fine pitch) (12 × 12)	Special
μPD78F0849GKA2-GAK-G	80-pin plastic LQFP (fine pitch) (12 × 12)	Special

Remark All these products are lead free products.

1.4 Pin Configuration (Top View)

- 78K0/DE2 (μPD78F0836, 78F0837, 78F0844, 78F0845)
  64-pin plastic LQFP (fine pitch) (10 × 10)



Note The alternate functions CTxD and CRxD are for the μPD78F0844, 78F0845 only.

Cautions 1. Make AVSS the same potential as VSS.

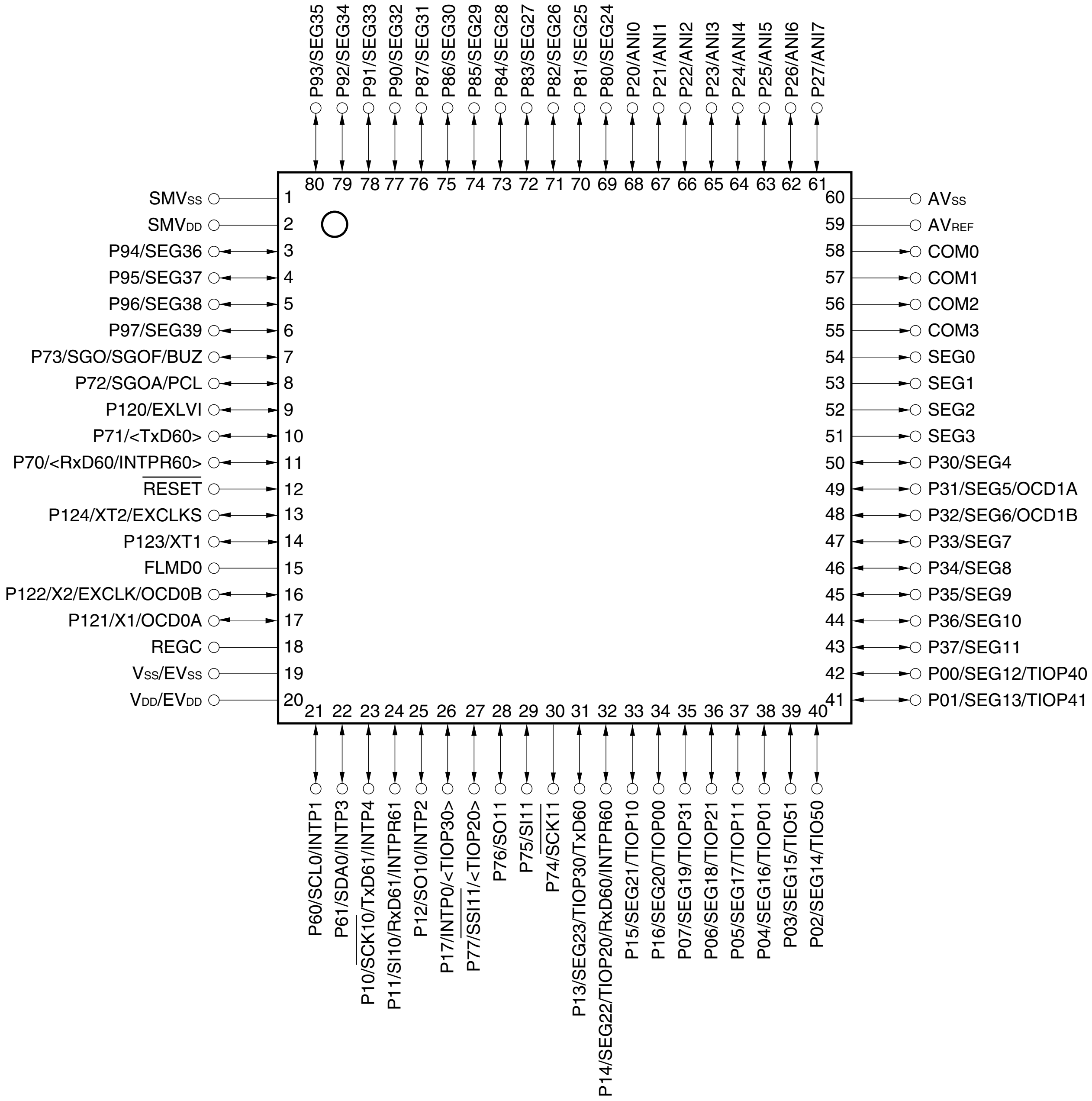
2. Connect the REGC pin to VSS via a capacitor (0.47 to 1 μF: recommended).

3. ANI0/P20 to ANI3/P23 are set in the analog input mode after release of reset.

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

- 78K0/DF2 (μPD78F0838, 78F0839)

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



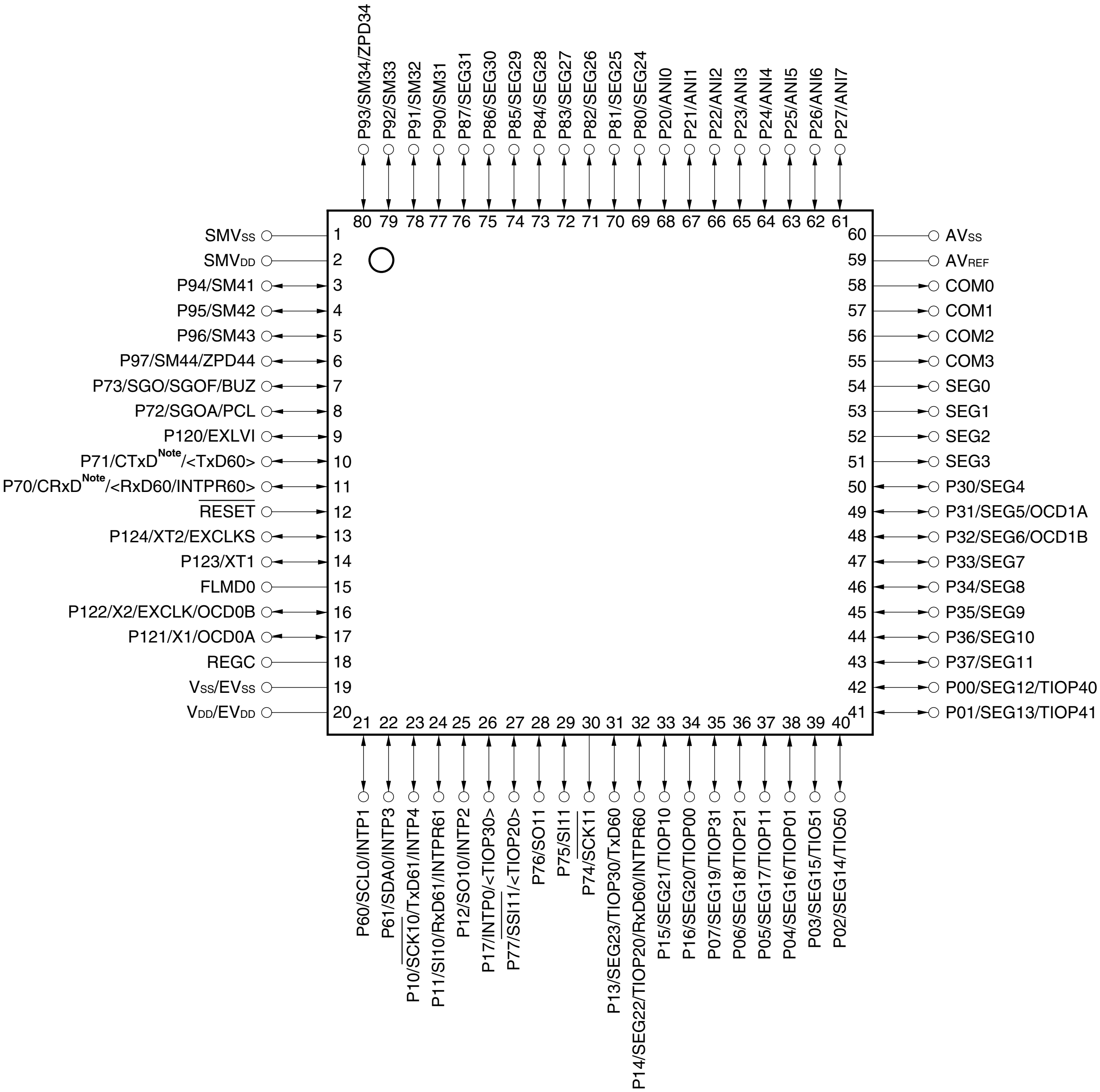
Cautions 1. Make AVSS the same potential as VSS.

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.

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

- 78K0/DF2 (μPD78F0840, 78F0841, 78F0846, 78F0847)
  80-pin plastic LQFP (fine pitch) (12 × 12)



Note The alternate functions CTxD and CRxD are for the μPD78F0846, 78F0847 only.

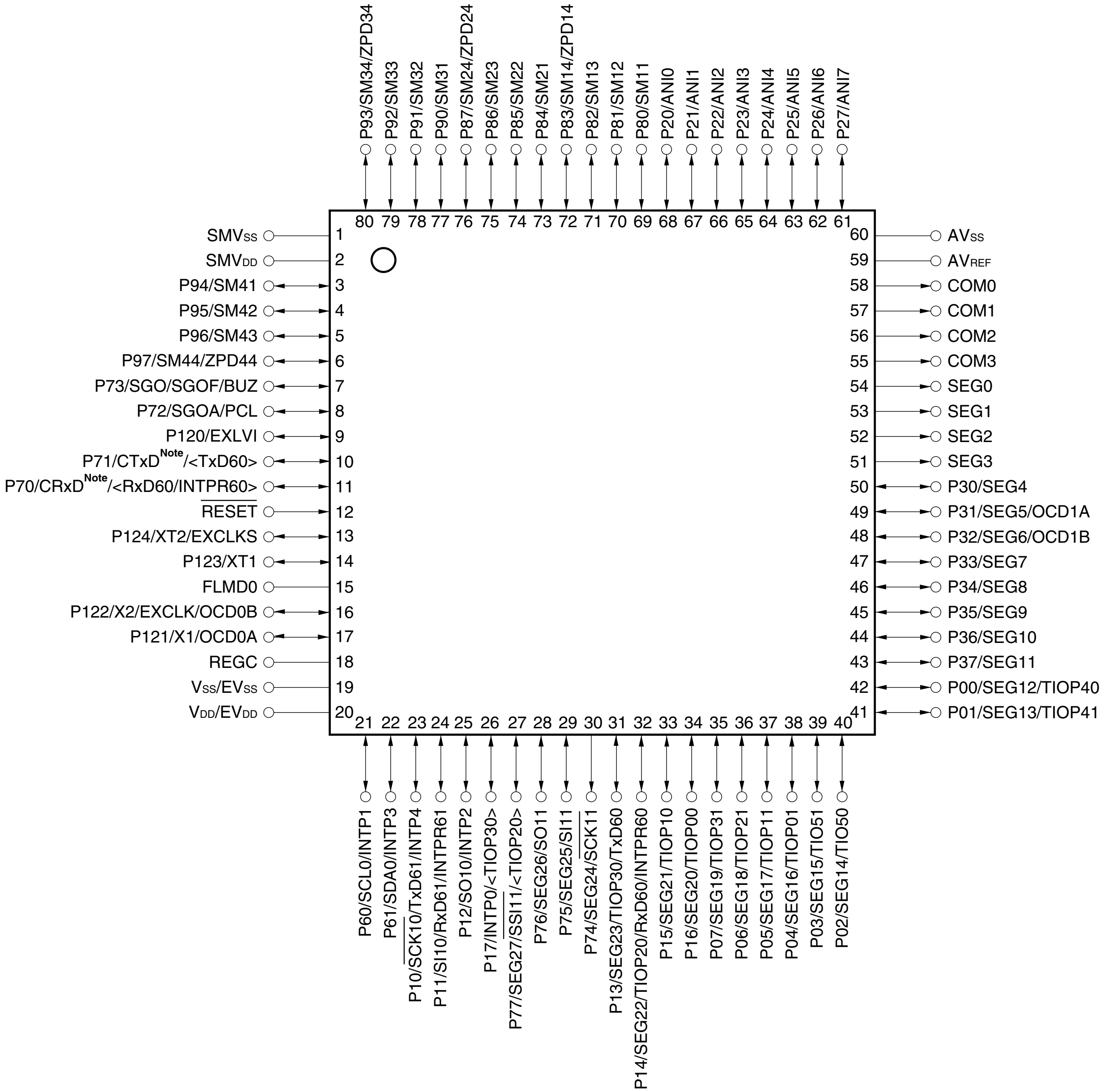
Cautions 1. Make AVSS the same potential as VSS.

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.

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

- 78K0/DF2 (μPD78F0842, 78F0843, 78F0848, 78F0849)
  80-pin plastic LQFP (fine pitch) (12 × 12)



Note The alternate functions CTxD and CRxD are for the μPD78F0848, 78F0849 only.

Cautions 1. Make AV$_{SS}$ the same potential as V$_{SS}$.

2. Connect the REGC pin to V$_{SS}$ 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.

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

Pin Identification

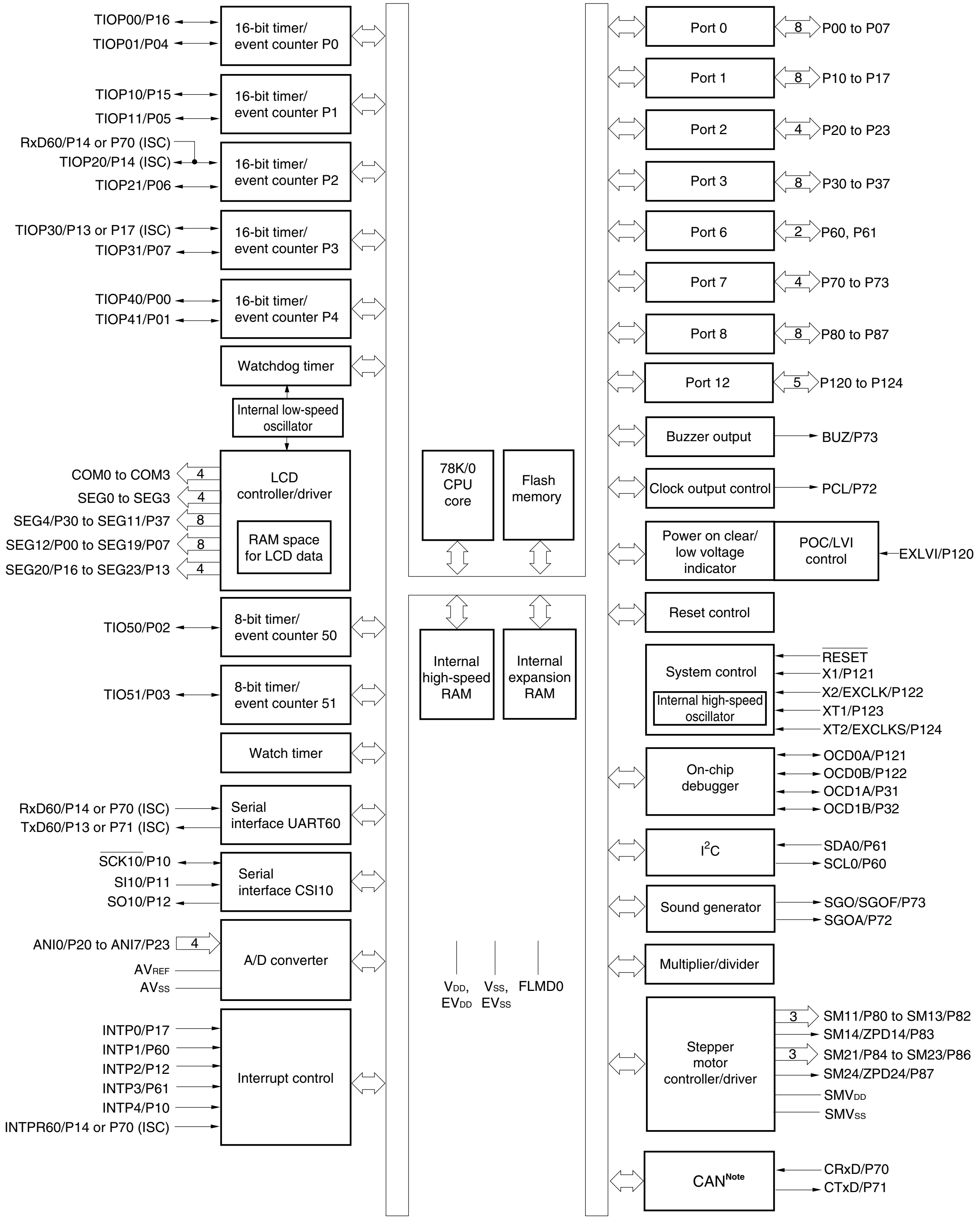
ANI0 to ANI7:	Analog input
AV$_{REF}$:	Analog reference voltage
AV$_{SS}$:	Analog ground
BUZ:	Buzzer output
COM0 to COM3:	Common output
CRxD:	Receive data for CAN
CTxD:	Transmit data for CAN
EV$_{DD}$:	Power supply for port
EV$_{SS}$:	Ground for port
EXCLK:	External clock input (Main system clock)
EXCLKS:	External clock input (Subsystem clock)
EXLVI:	External potential input for low-voltage detector
FLMD0:	Flash programming mode
INTP0 to INTP4, INTPR60, INTPR61:	External interrupt input
OCD0A, OCD0B, OCD1A OCD1B:	On-chip debug input/output
P00 to P07:	Port 0
P10 to P17:	Port 1
P20 to P27:	Port 2
P30 to P37:	Port 3
P60, P61:	Port 6
P70 to P77:	Port 7
P80 to P87:	Port 8
P90 to P97:	Port 9
P120 to P124	Port 12
PCL:	Programmable clock output
REGC:	Regulator capacitance
$\overline{\text{RESET}}$:	Reset
RxD60, RxD61:	Receive data
$\overline{\text{SCK10}}$, $\overline{\text{SCK11}}$:	Serial clock input/output
SCL0:	Serial clock input/output
SDA0:	Serial data input/output
SEG0 to SEG39:	Segment output
SGO:	Sound generator output
SGOA:	Sound generator amplitude PWM output
SGOF:	Sound generator frequency output
SI10, SI11:	Serial data input
SM11 to SM14, SM21 to SM24, SM31 to SM34, SM41 to SM44:	Stepper motor outputs
SMV$_{DD}$:	Stepper motor controller/driver supply voltage
SMV$_{SS}$:	Stepper motor controller/driver ground
SO10, SO11:	Serial data output
$\overline{\text{SSI11}}$:	Serial interface chip select input
TIO50, TIO51, TIOP00, TIOP01, TIOP10, TIOP11, TIOP20, TIOP21, TIOP30, TIOP31, TIOP40, TIOP41:	Timer input and timer output
TxD60, TxD61:	Transmit data
V$_{DD}$:	Power supply
V$_{SS}$:	Ground
X1, X2:	Crystal oscillator (High-speed system clock)
XT1, XT2:	Crystal oscillator (Subsystem clock)
ZPD14, ZPD24, ZPD34, ZPD44:	Zero point detection input

1.5 Block Diagram

- 78K0/DE2 (μPD78F0836, 78F0837, 78F0844, 78F0845)

<R>

<R>

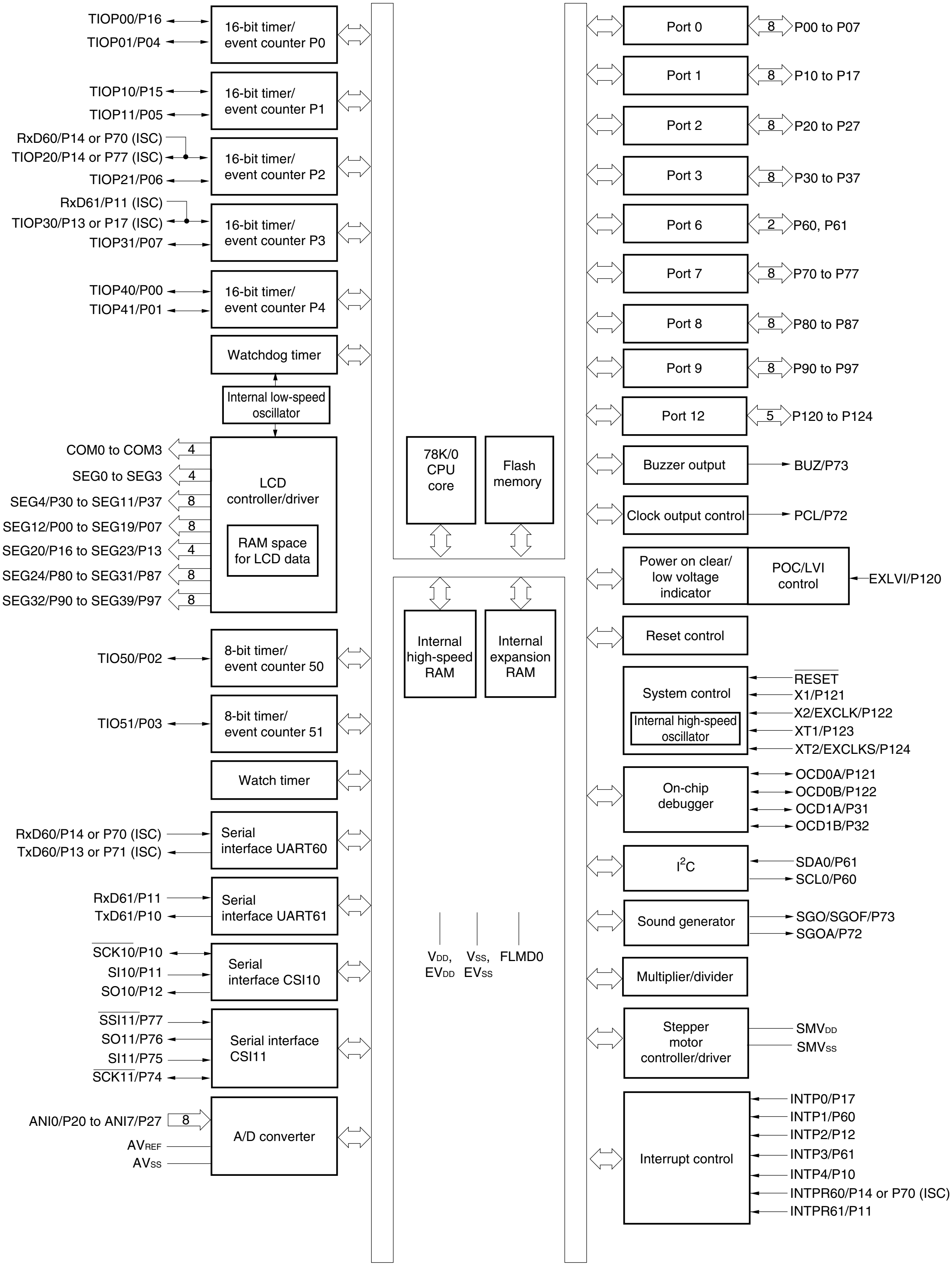


Note μPD78F0844, 78F0845 only.

- 78K0/DF2 (μPD78F0838, 78F0839)

<R>

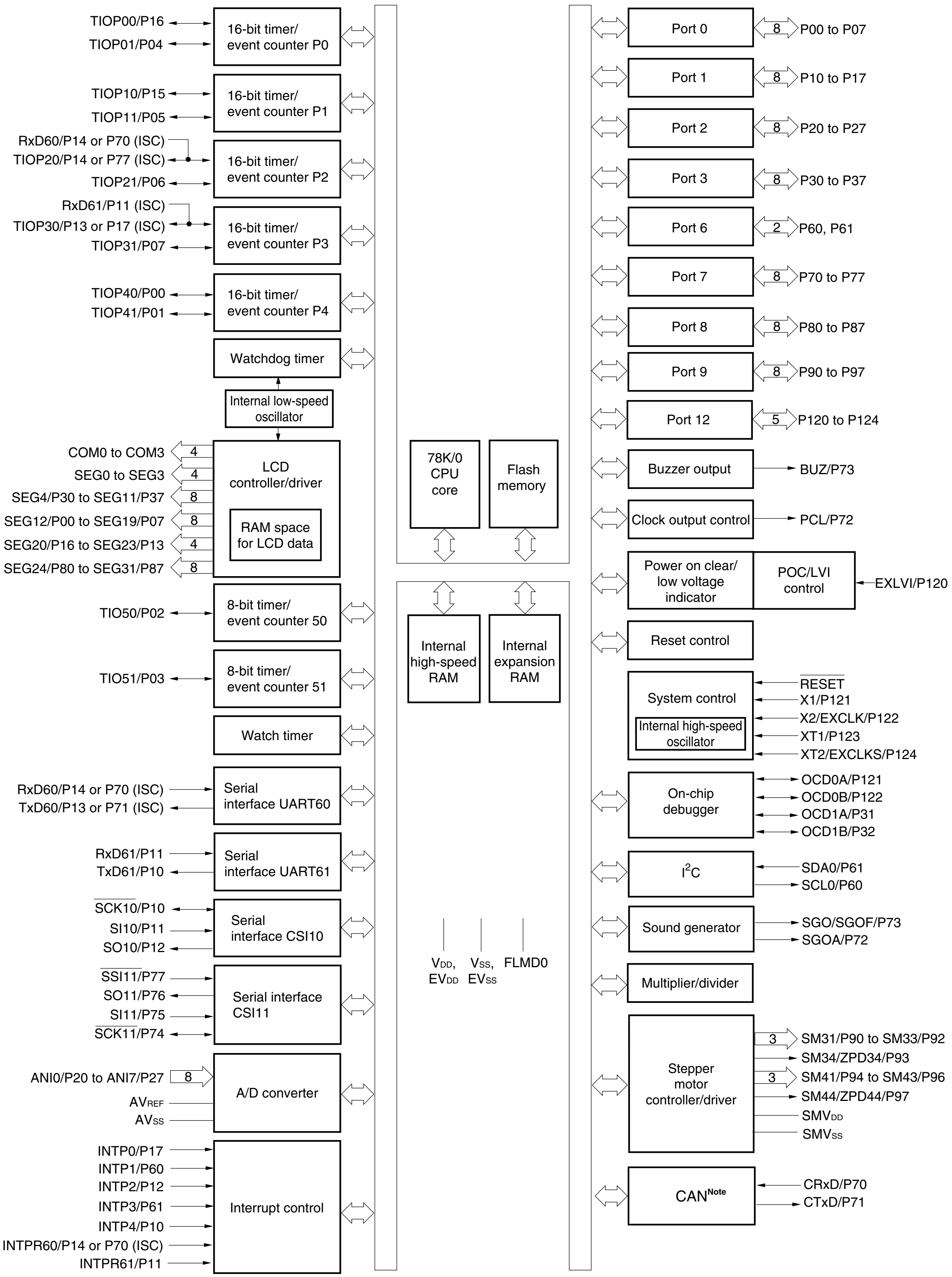
<R>



- 78K0/DF2 (μPD78F0840, 78F0841, 78F0846, 78F0847)

<R>

<R>

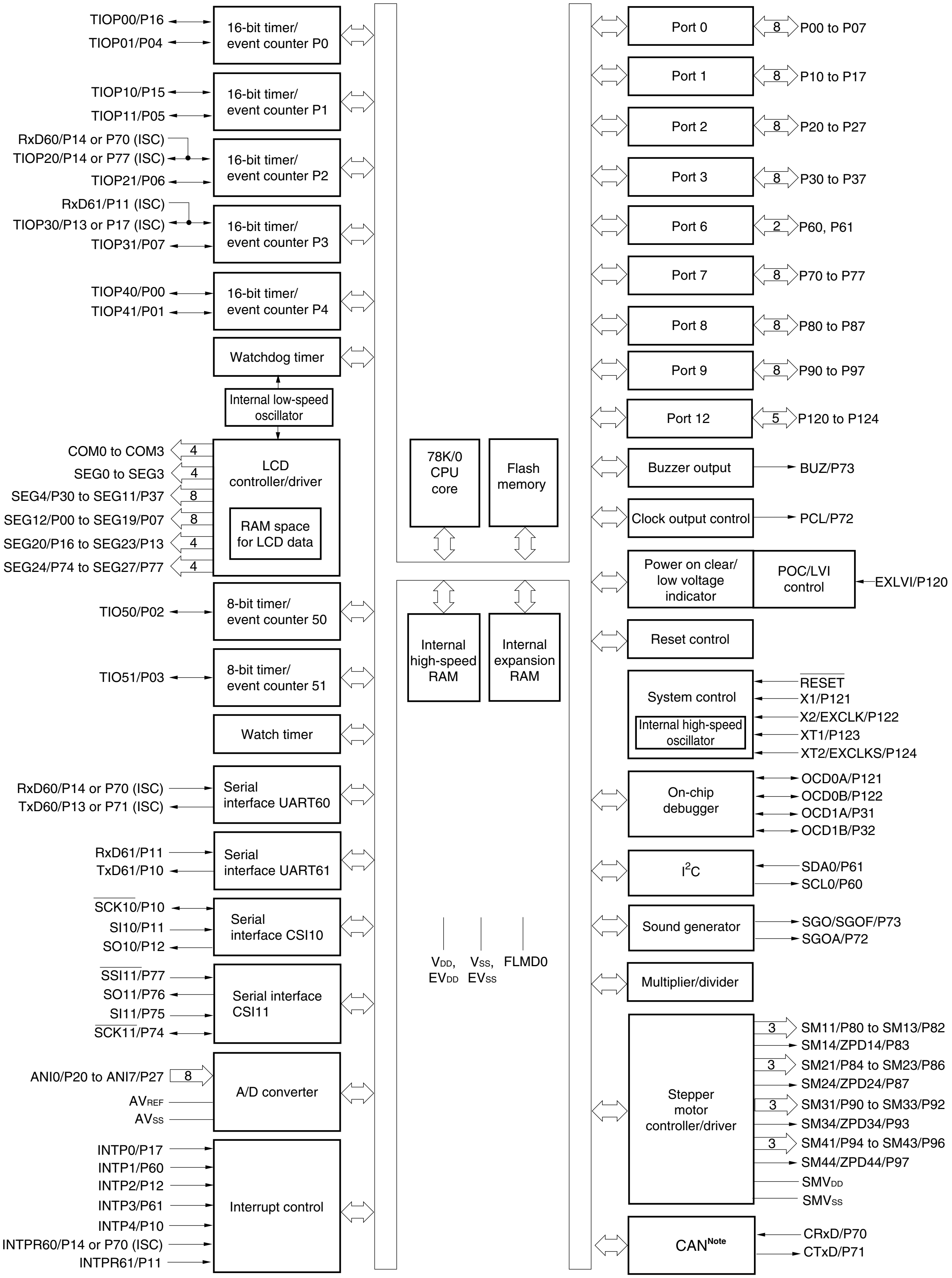


Note μPD78F0846, 78F0847 only.

- 78K0/DF2 (μPD78F0842, 78F0843, 78F0848, 78F0849)

<R>

<R>



Note μPD78F0848, 78F0849 only.

1.6 Outline of Functions

- 78K0/DE2 without CAN (μPD78F0836, 78F0837)

(1/2)

Item		μPD78F0836	μPD78F0837
Internal memory (bytes)	Flash memory (self-programming supported)[Note]	24 K	48 K
	High-speed RAM[Note]	1 K	
	Expansion RAM[Note]	1 K	2K
CAN buffer RAM		–	
High-speed system clock (oscillation frequency)		Crystal/ceramic oscillation (X1), external main system clock input (EXCLK) 4 to 20 MHz: V_{DD} = 4.0 to 5.5 V, 4 to 10 MHz: V_{DD} = 2.7 to 5.5 V	
Internal high-speed oscillation clock (oscillation frequency)		On-chip internal oscillation (8 MHz (TYP.): V_{DD} = 2.7 to 5.5 V)	
Internal low-speed oscillation clock (oscillation frequency)		On-chip internal oscillation (240 kHz (TYP.))	
Subsystem clock (oscillation frequency)		Crystal oscillation (XT1), external subsystem clock input (EXCLKS) (32.768 kHz: V_{DD} = 2.7 to 5.5 V)	
General-purpose registers		8 bits × 32 registers (8 bits × 8 registers × 4 banks)	
Minimum instruction execution time		0.1 μs/0.2 μs/0.4 μs/0.8 μs/1.6 μs (high-speed system clock: @ f_{MAIN} = 20 MHz operation)	
		0.25 μs/0.5 μs/1.0 μs/2.0 μs/4.0 μs (TYP.) (internal oscillator clock: @ f_{OSC8} = 8 MHz (TYP.) operation)	
		122 μs (subsystem clock: when operating at f_{XT} = 32.768 kHz)	
Instruction set		• 16-bit operation • Multiply/divide (16 bits × 16 bits, 32 bits ÷ 16 bits) • Bit manipulate (set, reset, test, and Boolean operation) • BCD adjust, etc.	
I/O ports		Total : 47 CMOS I/O 45 N-ch open-drain I/O 2 (5 V tolerant / N-ch open-drain output selectable)	
Timers		• 16-bit timer/event counter: 5 channels • 8-bit timer/event counter: 2 channels • Watch timer 1 channel • Watchdog timer: 1 channel	
<R>	Timer outputs	12 (PWM output: 7)	
Clock output		• 78.125 kHz, 156.25 kHz, 312.5 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz (high-speed system clock: 10 MHz) • 32.768 kHz (subsystem clock: 32.768 kHz)	
Buzzer output		1.22 kHz, 2.44 kHz, 4.88 kHz, 9.77 kHz (high-speed system clock: 10 MHz)	
A/D converter		10-bit resolution × 4 channels	

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		μPD78F0836	μPD78F0837
Serial interface	CAN	–	
	LIN-UART/CSI[Note 1]	1 ch	
	I²C bus	1 ch	
LCD controller/driver (seg × com)		24 × 4	
Sound generator		1 ch	
Stepper motor controller/driver (with ZPD)		2 ch	
Multiplier/divider		• 16 bit × 16 bit = 32 bit (Multiplication) • 32 bit ÷ 16 bit = 32 bit remainder of 16 bits (Division)	
Vectored interrupt sources	Internal	19	
<R>	External[Note 2]	6	
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	
Supply voltage		V_{DD} = 2.7 to 5.5 V	
Operating ambient temperature		T_A = –40 to +85°C, –40 to +105°C	
Package		64-pin plastic LQFP (fine pitch) (10 × 10)	

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

2. The external interrupt sources INTP3 and INTP4 can not be used at the same time.

- 78K0/DF2 without CAN (µPD78F0838, 78F0839, 78F0840, 78F0841, 78F0842, 78F0843)

(1/2)

Item		µPD78F0838 µPD78F0839	µPD78F0840 µPD78F0841	µPD78F0842 µPD78F0843
Internal memory (bytes)	Flash memory (self-programming supported)[Note]	24 K/48 K		
	High-speed RAM[Note]	1 K		
	Expansion RAM[Note]	1 K		
CAN buffer RAM		–		
High-speed system clock (oscillation frequency)		Crystal/ceramic oscillation (X1), external main system clock input (EXCLK) 4 to 20 MHz: V_{DD} = 4.0 to 5.5 V, 4 to 10 MHz: V_{DD} = 2.7 to 5.5 V		
Internal high-speed oscillation clock (oscillation frequency)		On-chip internal oscillation (8 MHz (TYP.): V_{DD} = 2.7 to 5.5 V)		
Internal low-speed oscillation clock (oscillation frequency)		On-chip internal oscillation (240 kHz (TYP.))		
Subsystem clock (oscillation frequency)		Crystal oscillation (XT1), external subsystem clock input (EXCLKS) (32.768 kHz: V_{DD} = 2.7 to 5.5 V)		
General-purpose registers		8 bits × 32 registers (8 bits × 8 registers × 4 banks)		
Minimum instruction execution time		0.1 µs/0.2 µs/0.4 µs/0.8 µs/1.6 µs (high-speed system clock: @ f_{MAIN} = 20 MHz operation)		
		0.25 µs/0.5 µs/1.0 µs/2.0 µs/4.0 µs (TYP.) (internal oscillator clock: @ f_{OSC8} = 8 MHz (TYP.) operation)		
		122 µs (subsystem clock: when operating at f_{XT} = 32.768 kHz)		
Instruction set		• 16-bit operation • Multiply/divide (16 bits × 16 bits, 32 bits ÷ 16 bits) • Bit manipulate (set, reset, test, and Boolean operation) • BCD adjust, etc.		
I/O ports		Total : 63 CMOS I/O 61 N-ch open-drain I/O 2 (5 V tolerant / N-ch open-drain output selectable)		
Timers		• 16-bit timer/event counter: 5 channels • 8-bit timer/event counter: 2 channels • Watch timer 1 channel • Watchdog timer: 1 channel		
<R>	Timer outputs	12 (PWM output: 7)		
Clock output		• 78.125 kHz, 156.25 kHz, 312.5 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz (high-speed system clock: 10 MHz) • 32.768 kHz (subsystem clock: 32.768 kHz)		
Buzzer output		1.22 kHz, 2.44 kHz, 4.88 kHz, 9.77 kHz (high-speed system clock: 10 MHz)		
A/D converter		10-bit resolution × 8 channels		

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		μPD78F0838 μPD78F0839	μPD78F0840 μPD78F0841	μPD78F0842 μPD78F0843
Serial interface	CAN	–		
	3-wire CSI	1 ch		
	LIN-UART	1 ch		
	LIN-UART/CSI[Note 1]	1 ch		
	I^{2}C bus	1 ch		
LCD controller/driver (seg × com)		40 × 4	32 × 4	28 × 4
Sound generator		1 ch		
Stepper motor controller/driver (with ZPD)		–	2 ch	4 ch
Multiplier/divider		• 16 bit × 16 bit = 32 bit (Multiplication) • 32 bit ÷ 16 bit = 32 bit remainder of 16 bits (Division)		
<R> Vectored interrupt sources	Internal	22		
	External[Note 2]	7		
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		
Supply voltage		V_{DD} = 2.7 to 5.5 V		
Operating ambient temperature		T_A = –40 to +85°C, –40 to +105°C		
Package		80-pin plastic LQFP (fine pitch) (12 × 12)		

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

2. The external interrupt sources INTP3 and INTP4 can not be used at the same time.

- 78K0/DE2 with CAN (µPD78F0844, 78F0845)

(1/2)

Item		µPD78F0844	µPD78F0845
Internal memory (bytes)	Flash memory (self-programming supported)[Note]	32 K	60 K
	High-speed RAM[Note]	1 K	
	Expansion RAM[Note]	1 K	2K
CAN buffer RAM		288 bytes	
High-speed system clock (oscillation frequency)		Crystal/ceramic oscillation (X1), external main system clock input (EXCLK) 4 to 20 MHz: V_{DD} = 4.0 to 5.5 V, 4 to 10 MHz: V_{DD} = 2.7 to 5.5 V	
Internal high-speed oscillation clock (oscillation frequency)		On-chip internal oscillation (8 MHz (TYP.): V_{DD} = 2.7 to 5.5 V)	
Internal low-speed oscillation clock (oscillation frequency)		On-chip internal oscillation (240 kHz (TYP.))	
Subsystem clock (oscillation frequency)		Crystal oscillation (XT1), external subsystem clock input (EXCLKS) (32.768 kHz: V_{DD} = 2.7 to 5.5 V)	
General-purpose registers		8 bits × 32 registers (8 bits × 8 registers × 4 banks)	
Minimum instruction execution time		0.1 µs/0.2 µs/0.4 µs/0.8 µs/1.6 µs (high-speed system clock: @ f_{MAIN} = 20 MHz operation)	
		0.25 µs/0.5 µs/1.0 µs/2.0 µs/4.0 µs (TYP.) (internal oscillator clock: @ f_{OSC8} = 8 MHz (TYP.) operation)	
		122 µs (subsystem clock: when operating at f_{XT} = 32.768 kHz)	
Instruction set		• 16-bit operation • Multiply/divide (16 bits × 16 bits, 32 bits ÷ 16 bits) • Bit manipulate (set, reset, test, and Boolean operation) • BCD adjust, etc.	
I/O ports		Total : 47 CMOS I/O 45 N-ch open-drain I/O 2 (5 V tolerant / N-ch open-drain output selectable)	
Timers		• 16-bit timer/event counter: 5 channels • 8-bit timer/event counter: 2 channels • Watch timer 1 channel • Watchdog timer: 1 channel	
<R>	Timer outputs	12 (PWM output: 7)	
Clock output		• 78.125 kHz, 156.25 kHz, 312.5 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz (high-speed system clock: 10 MHz) • 32.768 kHz (subsystem clock: 32.768 kHz)	
Buzzer output		1.22 kHz, 2.44 kHz, 4.88 kHz, 9.77 kHz (high-speed system clock: 10 MHz)	
A/D converter		10-bit resolution × 4 channels	

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		μPD78F0844	μPD78F0845
Serial interface	CAN	1 ch	
	LIN-UART/CSI[Note 1]	1 ch	
	I^{2}C bus	1 ch	
LCD controller/driver (seg × com)		24 × 4	
Sound generator		1 ch	
Stepper motor controller/driver (with ZPD)		2 ch	
Multiplier/divider		• 16 bit × 16 bit = 32 bit (Multiplication) • 32 bit ÷ 16 bit = 32 bit remainder of 16 bits (Division)	
Vectored interrupt sources	Internal	22	
<R>	External[Note 2]	6	
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	
Supply voltage		V_{DD} = 2.7 to 5.5 V	
Operating ambient temperature		T_A = –40 to +85°C, –40 to +105°C	
Package		64-pin plastic LQFP (fine pitch) (10 × 10)	

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

2. The external interrupt sources INTP3 and INTP4 can not be used at the same time.

- 78K0/DF2 with CAN (μPD78F0846, 78F0847, 78F0848, 78F0849)

(1/2)

Item		μPD78F0846 μPD78F0847	μPD78F0848 μPD78F0849
Internal memory (bytes)	Flash memory (self-programming supported)[Note]	32 K/60 K	
	High-speed RAM[Note]	1 K	
	Expansion RAM[Note]	2 K	
CAN buffer RAM		288 bytes	
High-speed system clock (oscillation frequency)		Crystal/ceramic oscillation (X1), external main system clock input (EXCLK) 4 to 20 MHz: V_{DD} = 4.0 to 5.5 V, 4 to 10 MHz: V_{DD} = 2.7 to 5.5 V	
Internal high-speed oscillation clock (oscillation frequency)		On-chip internal oscillation (8 MHz (TYP.): V_{DD} = 2.7 to 5.5 V)	
Internal low-speed oscillation clock (oscillation frequency)		On-chip internal oscillation (240 kHz (TYP.))	
Subsystem clock (oscillation frequency)		Crystal oscillation (XT1), external subsystem clock input (EXCLKS) (32.768 kHz: V_{DD} = 2.7 to 5.5 V)	
General-purpose registers		8 bits × 32 registers (8 bits × 8 registers × 4 banks)	
Minimum instruction execution time		0.1 μs/0.2 μs/0.4 μs/0.8 μs/1.6 μs (high-speed system clock: @ f_{MAIN} = 20 MHz operation)	
		0.25 μs/0.5 μs/1.0 μs/2.0 μs/4.0 μs (TYP.) (internal oscillator clock: @ f_{OSC8} = 8 MHz (TYP.) operation)	
		122 μs (subsystem clock: when operating at f_{XT} = 32.768 kHz)	
Instruction set		• 16-bit operation • Multiply/divide (16 bits × 16 bits, 32 bits ÷ 16 bits) • Bit manipulate (set, reset, test, and Boolean operation) • BCD adjust, etc.	
I/O ports		Total : 63 CMOS I/O 61 N-ch open-drain I/O 2 (5 V tolerant / N-ch open-drain output selectable)	
Timers		• 16-bit timer/event counter: 5 channels • 8-bit timer/event counter: 2 channels • Watch timer 1 channel • Watchdog timer: 1 channel	
<R>	Timer outputs	12 (PWM output: 7)	
Clock output		• 78.125 kHz, 156.25 kHz, 312.5 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz (high-speed system clock: 10 MHz) • 32.768 kHz (subsystem clock: 32.768 kHz)	
Buzzer output		1.22 kHz, 2.44 kHz, 4.88 kHz, 9.77 kHz (high-speed system clock: 10 MHz)	
A/D converter		10-bit resolution × 8 channels	

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		µPD78F0846 µPD78F0847	µPD78F0848 µPD78F0849
Serial interface	CAN	1 ch	
	3-wire CSI	1 ch	
	LIN-UART	1 ch	
	LIN-UART/CSI[Note 1]	1 ch	
	I^2C bus	1 ch	
LCD controller/driver (seg × com)		32 × 4	28 × 4
Sound generator		1 ch	
Stepper motor controller/driver (with ZPD)		2 ch	4 ch
Multiplier/divider		• 16 bit × 16 bit = 32 bit (Multiplication) • 32 bit ÷ 16 bit = 32 bit remainder of 16 bits (Division)	
<R> Vectored interrupt sources	Internal	25	
	External[Note 2]	7	
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	
Supply voltage		V_{DD} = 2.7 to 5.5 V	
Operating ambient temperature		T_A = –40 to +85°C, –40 to +105°C	
Package		80-pin plastic LQFP (fine pitch) (12 × 12)	

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

2. The external interrupt sources INTP3 and INTP4 can not be used at the same time.

- Outline of timers

An outline of the timer is shown below.

		16-Bit Timer/ Event Counters P0 to P4					8-Bit Timer/ Event Counters 50 and 51		Watch Timer	Watchdog Timer
		TMP0	TMP1	TMP2	TMP3	TMP4	TM50	TM51		
Operation mode	Interval timer	1 ch	1 ch	1 ch	1 ch	1 ch	1 ch	1 ch	1 channel[Note]	1 channel
	External event counter	1 ch	1 ch	1 ch	1 ch	1 ch	1 ch	1 ch	–	–
Function	Timer output	2	2	2	2	2	1	1	–	–
	External trigger pulse output	1	1	1	1	1	–	–	–	–
	PWM output	1	1	1	1	1	1	1	–	–
	Pulse width measurement	2	2	2	2	2	–	–	–	–
	Square-wave output	–	–	–	–	–	1	1	–	–
<R>	Interrupt source	2	2	3	3	2	1	1	1	–

<R> **Note** In the watch timer, the watch timer function and interval timer function cannot be used simultaneously.

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

There are four types of pin I/O buffer power supplies: AV_{REF}, EV_{DD}, SMV_{DD}, 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}	P20 to P23, P24 to P27[Note]
EV_{DD}	Port pins other than P20 to P27, P80 to P87, P90 to P97, and P121 to P124
SMV_{DD}	• P80 to P87 • P90 to P97[Note]
V_{DD}	• P121 to P124 • Non-port pins

Note 78K0/DF2 only.

This section explains the names and functions of the pins of the 78K0/Dx2.

(1) Port pins

- 78K0/DE2 (μPD78F0836, 78F0837, 78F0844, 78F0845)

<R>

Table 2-2. Port Pins for 78K0/DE2 (1/2)

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
P00	I/O	Port 0. 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.	Ext.: PD Int.: HZ	Input	SEG12/TIOP40
P01					SEG13/TIOP41
P02					SEG14/TIO50
P03					SEG15/TIO51
P04					SEG16/TIOP01
P05					SEG17/TIOP11
P06					SEG18/TIOP21
P07					SEG19/TIOP31
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.	HZ	Input	$\overline{\text{SCK10}}$/INTP4
P11					SI10
P12					SO10/INTP2
P13			Ext.: PD Int.: HZ		SEG23/TIOP30/TxD60
P14					SEG22/TIOP20/RxD60/ INTPR60
P15					SEG21/TIOP10
P16					SEG20/TIOP00
P17			HZ		INTP0/<TIOP30>
P20 to P23	I/O	Port 2 4-bit I/O port. Input/output can be specified in 1-bit units.	HZ	Input	ANI0 to ANI3
P30	I/O	Port 3. 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.	Ext.: PD Int.: HZ	Input	SEG4
P31					SEG5/OCD1A
P32					SEG6/OCD1B
P33 to P37					SEG7 to SEG11
P60	I/O	Port 6. 2-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.	HZ	Input	SCL0/INTP1
P61					SDA0/INTP3

Remarks 1. Ext. (external reset): POC reset or pin reset, Int. (internal reset): WDT reset or LVI reset, PD: Pull down, HZ: High impedance

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

Table 2-2. Port Pins for 78K0/DE2 (2/2)

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
P70	I/O	Port 7. 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.	HZ	Input	CRxD[Note]/ <RXD60/INTPR60>
P71					CTxD[Note]/<TxD60>
P72					SGOA/PCL
P73					SGO/SGOF/BUZ
P80 to P82	I/O	Port 8 8-bit I/O port. Input/output can be specified in 1-bit units.	Ext.: PD Int.: HZ	Input	SM11 to SM13
P83					SM14/ZPD14
P84 to P86					SM21 to SM23
P87					SM24/ZPD24
P120	I/O	Port 12. 5-bit I/O port. Only for P120, use of an on-chip pull-up resistor can be specified by a software setting.	HZ	Input	EXLVI
P121					X1/OCD0A
P122					X2/EXCLK/OCD0B
P123					XT1
P124					XT2/EXCLKS

Note μPD78F0844 and 78F0845 only.

Remarks 1. Ext. (external reset): POC reset or pin reset, Int. (internal reset): WDT reset or LVI reset, PD: Pull down, HZ: High impedance

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

- 78K0/DF2 (μPD78F0838, 78F0839, 78F0840, 78F0841, 78F0842, 78F0843, 78F0846, 78F0847, 78F0848, 78F0849)

<R> **Table 2-3. Port Pins for 78K0/DF2 (1/2)**

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
P00	I/O	Port 0. 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.	Ext.: PD Int.: HZ	Input	SEG12/TIOP40
P01					SEG13/TIOP41
P02					SEG14/TIO50
P03					SEG15/TIO51
P04					SEG16/TIOP01
P05					SEG17/TIOP11
P06					SEG18/TIOP21
P07					SEG19/TIOP31
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.	HZ	Input	$\overline{\text{SCK10}}$/TxD61/INTP4
P11					SI10/RxD61/INTPR61
P12					SO10/INTP2
P13			Ext.: PD Int.: HZ		SEG23/TIOP30/TxD60
P14					SEG22/TIOP20/RxD60/ INTPR60
P15					SEG21/TIOP10
P16					SEG20/TIOP00
P17			HZ		INTP0/<TIOP30>
P20 to P27	I/O	Port 2 8-bit I/O port. Input/output can be specified in 1-bit units.	HZ	Input	ANI0 to ANI7
P30	I/O	Port 3. 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.	Ext.: PD Int.: HZ	Input	SEG4
P31					SEG5/OCD1A
P32					SEG6/OCD1B
P33 to P37					SEG7 to SEG11
P60	I/O	Port 6. 2-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.	HZ	Input	SCL0/INTP1
P61					SDA0/INTP3

Note Ext. (external reset): POC reset or pin reset, Int. (internal reset): WDT reset or LVI reset, PD: Pull down, HZ: High impedance

Remarks 1. Ext. (external reset): POC reset or pin reset, Int. (internal reset): WDT reset or LVI reset, PD: Pull down, HZ: High impedance

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

Table 2-3. Port Pins for 78K0/DF2 (2/2)

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
P70	I/O	Port 7. 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.	HZ	Input	CRxD[Note1]/ <RXD60/INTPR60>
P71					CTxD[Note1]/<TxD60>
P72					SGOA/PCL
P73					SGO/SGOF/BUZ
P74			Ext.: PD Int.: HZ		SEG24[Note2]/$\overline{\text{SCK11}}$
P75					SEG25[Note2]/SI11
P76					SEG26[Note2]/SO11
P77					SEG27[Note2]/$\overline{\text{SSI11}}$/ <TIOP20>
P80 to P82	I/O	Port 8 8-bit I/O port. Input/output can be specified in 1-bit units.	Ext.: PD Int.: HZ	Input	SEG24 to SEG26[Note3] SM11 to SM13[Note2]
P83					SEG27[Note3] SM14/ZPD14[Note2]
P84 to P86					SEG28 to SEG30[Note3] SM21 to SM23[Note2]
P87					SEG31[Note3] SM24/ZPD24[Note2]
P90 to P92	I/O	Port 9 8-bit I/O port. Input/output can be specified in 1-bit units.	Ext.: PD Int.: HZ	Input	SEG32 to SEG34[Note4] SM31 to SM33[Note5]
P93					SEG35[Note4] SM34/ZPD34[Note5]
P94 to P96					SEG36 to SEG38[Note4] SM41 to SM43[Note5]
P97					SEG39[Note4] SM44/ZPD44[Note5]
P120	I/O	Port 12. 5-bit I/O port. Only for P120, use of an on-chip pull-up resistor can be specified by a software setting.	HZ	Input	EXLVI
P121					X1/OCD0A
P122					X2/EXCLK/OCD0B
P123					XT1
P124					XT2/EXCLKS

Notes 1. μPD78F0846, 78F0847, 78F0848, and 78F0849 only.

2. For μPD78F0842, 78F0843, 78F0848, and 78F0849.

3. For μPD78F0838 to 78F0841, 78F0846, and 78F0847.

4. For μPD78F0838 and 78F0839.

5. For μPD78F0840, 78F0841, 78F0842, 78F0843, 78F0846, 78F0847, 78F0848, and 78F0849.

Remarks 1. Ext. (external reset): POC reset or pin reset, Int. (internal reset): WDT reset or LVI reset, PD: Pull down, HZ: High impedance

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

(2) Non-port pins

- 78K0/DE2 (μPD78F0836, 78F0837, 78F0844, 78F0845)

<R> **Table 2-4. Non-port Pins for 78K0/DE2 (1/4)**

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
ANI0 to ANI3	Input	A/D converter analog input	HZ	Input	P20 to P23
AV$_{REF}$	–	A/D converter reference voltage input and positive power supply for port 2	–	–	–
AV$_{SS}$	–	A/D converter ground potential. Make the same potential as EV$_{SS}$ or V$_{SS}$.	–	–	–
BUZ	Output	Buzzer output	HZ	Input	P73/SGO/SGOF
COM0 to COM3	Output	LCD controller/driver common signal outputs	L	–	–
CRxD[Note]	Input	CAN receive data input	HZ	Input	P70/<RxD60/INTPR60>
CTxD[Note]	Output	CAN transmit data output	HZ	Input	P71/<TxD60>
EXCLK	Input	External clock input for main system clock	HZ	Input	P122/X2/OCD0B
EXCLKS	Input	External clock input for subsystem clock	HZ	Input	P124/XT2
EXLVI	Input	Potential input for external low-voltage detection	HZ	Input	P120
FLMD0	–	Flash memory programming mode setting	–	–	–
INTP0	Input	External interrupt request input for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.	HZ	Input	P17/<TIOP30>
INTP1					P60/SCL0
INTP2					P12/SO10
INTP3					P61/SDA0
INTP4					P10/$\overline{\text{SCK10}}$
INTPR60			Ext.: PD Int.: HZ		P14/SEG22/TIOP20/RxD60
<INTPR60>			HZ		P70/CRxD[Note]/<RxD60>
OCD0A	I/O	On-chip debug mode setting connection	HZ	Input	P121/X1
OCD0B					P122/X2/EXCLK
OCD1A			Ext.: PD Int.: HZ		P31/SEG5
OCD1B					P32/SEG6
PCL	Output	Clock output (for trimming of high-speed system clock, subsystem clock)	HZ	Input	P72/SGOA
REGC	–	This is the pin for connecting regulator output (2.5 V) stabilization capacitance for internal operation. Connect this pin to V$_{SS}$ via a capacitor (0.47 to 1 μF: recommended).	–	–	–

Note μPD78F0844 and 78F0845 only.

Remarks 1. Ext. (external reset): POC reset or pin reset, Int. (internal reset): WDT reset or LVI reset,
PD: Pull down, HZ: High impedance, L: Low level output

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

Table 2-4. Non-port Pins for 78K0/DE2 (2/4)

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
$\overline{\text{RESET}}$	Input	System reset input	–	–	–
RxD60	Input	Serial data input to asynchronous serial interface	Ext.: PD Int.: HZ	Input	P14/SEG22/TIOP20/ INTPR60
<RxD60>			HZ		P70/CRxD[Note]/ <INTPR60>
$\overline{\text{SCK10}}$	I/O	Clock input/output for serial interface	HZ	Input	P10/INTP4
SCL0	I/O	Clock input/output for I^{2}C 5 V tolerant/N-ch open-drain output selectable.	HZ	Input	P60/INTP1
SDA0	I/O	Serial data I/O for I^{2}C 5 V tolerant/N-ch open-drain output selectable.	HZ	Input	P61/INTP3
SEG0 to SEG3	Output	LCD controller/driver segment signal outputs	Ext.: PD Int.: L	–	–
SEG4			Ext.: PD Int.: HZ	Input	P30
SEG5					P31/OCD1A
SEG6					P32/OCD1B
SEG7 to SEG11					P33 to P37
SEG12					P00/TIOP40
SEG13					P01/TIOP41
SEG14					P02/TIO50
SEG15					P03/TIO51
SEG16					P04/TIOP01
SEG17					P05/TIOP11
SEG18					P06/TIOP21
SEG19					P07/TIOP31
SEG20					P16/TIOP00
SEG21					P15/TIOP10
SEG22					P14/TIOP20/RxD60/ INTPR60
SEG23					P13/TIOP30/TxD60
SGO	Output	Sound generator output	HZ	Input	P73/SGOF/BUZ
SGOA		Sound generator amplitude PWM output			P72/PCL
SGOF		Sound generator frequency output			P73/SGO/BUZ
SI10	Input	Serial data input to serial interface	HZ	Input	P11

Note μPD78F0844 and 78F0845 only.

Remarks 1. Ext. (external reset): POC reset or pin reset, Int. (internal reset): WDT reset or LVI reset, PD: Pull down, HZ: High impedance, L: Low level output

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

Table 2-4. Non-port Pins for 78K0/DE2 (3/4)

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
SM11	Output	Stepper motor 1 output sin +	Ext.: PD Int.: HZ	Input	P80
SM12		Stepper motor 1 output sin –			P81
SM13		Stepper motor 1 output cos +			P82
SM14		Stepper motor 1 output cos –			P83/ZPD14
SM21		Stepper motor 2 output sin +			P84
SM22		Stepper motor 2 output sin –			P85
SM23		Stepper motor 2 output cos +			P86
SM24		Stepper motor 2 output cos –			P87/ZPD24
SMV_{DD}	–	Stepper motor controller/driver supply voltage	–	–	–
SMV_{SS}	–	Stepper motor controller/driver ground potential	–	–	–
SO10	Output	Serial data output from serial interface	HZ	Input	P12/INTP2
TIO50	I/O	External count clock input/timer output (TM50)	Ext.: PD Int.: HZ	Input	P02/SEG14
TIO51		External count clock input/timer output (TM51)			P03/SEG15
TIOP00	I/O	External event count input/capture trigger input/external trigger input/timer output (TMP0)	Ext.: PD Int.: HZ	Input	P16/SEG20
TIOP01		Capture trigger input/timer output (TMP0)			P04/SEG16
TIOP10		External event count input/capture trigger input/external trigger input/timer output (TMP1)			P15/SEG21
TIOP11		Capture trigger input/timer output (TMP1)			P05/SEG17
TIOP20		External event count input/capture trigger input/external trigger input/timer output (TMP2)			P14/SEG22/RxD60/INTPR60
TIOP21		Capture trigger input/timer output (TMP2)			P06/SEG18
TIOP30		External event count input/capture trigger input/external trigger input/timer output (TMP3)			P13/SEG23/TxD60
<TIOP30>			HZ		P17/INTP0
TIOP31		Capture trigger input/timer output (TMP3)	Ext.: PD Int.: HZ		P07/SEG19
TIOP40		External event count input/capture trigger input/external trigger input/timer output (TMP4)			P00/SEG12
TIOP41		Capture trigger input/timer output (TMP4)			P01/SEG13
TxD60	Output	Serial data output from asynchronous serial interface	Ext.: PD Int.: HZ	Input	P13/SEG23/TIOP30
<TxD60>			HZ		P71/CTxD[Note]
V_{DD}/EV_{DD}	–	Positive power supply (except for ports)/positive power supply for ports	–	–	–
V_{SS}/EV_{SS}	–	Ground potential (except for ports)/ground potential for ports	–	–	–

Note μPD78F0844 and 78F0845 only.

Remarks 1. Ext. (external reset): POC reset or pin reset, Int. (internal reset): WDT reset or LVI reset, PD: Pull down, HZ: High impedance

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

Table 2-4. Non-port Pins for 78K0/DE2 (4/4)

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
X1	Input	Connecting resonator for high-speed system clock	HZ	Input	P121/OCD0A
X2	–				P122/EXCLK/OCD0B
XT1	Input	Connecting resonator for subsystem clock	HZ	Input	P123
XT2	–				P124/EXCLKS
ZPD14	Input	Zero point detection input	Ext.: PD Int.: HZ	Input	P83/SM14
ZPD24					P87/SM24

Remark Ext. (external reset): POC reset or pin reset, Int. (internal reset): WDT reset or LVI reset, PD: Pull down, HZ: High impedance

- 78K0/DF2 (μPD78F0838, 78F0839, 78F0840, 78F0841, 78F0842, 78F0843, 78F0846, 78F0847, 78F0848, 78F0849)

<R> **Table 2-5. Non-port Pins for 78K0/DF2 (1/4)**

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
ANI0 to ANI7	Input	A/D converter analog input	HZ	Input	P20 to P27
AV_{REF}	–	A/D converter reference voltage input and positive power supply for port 2	–	–	–
AV_{SS}	–	A/D converter ground potential. Make the same potential as EV_{SS} or V_{SS}.	–	–	–
BUZ	Output	Buzzer output	HZ	Input	P73/SGO/SGOF
COM0 to COM3	Output	LCD controller/driver common signal outputs	L	–	–
CRxD[Note]	Input	CAN receive data input	HZ	Input	P70/<RxD60/INTPR60>
CTxD[Note]	Output	CAN transmit data output	HZ	Input	P71/<TxD60>
EXCLK	Input	External clock input for main system clock	HZ	Input	P122/X2/OCD0B
EXCLKS	Input	External clock input for subsystem clock	HZ	Input	P124/XT2
EXLVI	Input	Potential input for external low-voltage detection	HZ	Input	P120
FLMD0	–	Flash memory programming mode setting	–	–	–
INTP0	Input	External interrupt request input for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified	HZ	Input	P17/<TIOP30>
INTP1					P60/SCL0
INTP2					P12/SO10
INTP3					P61/SDA0
INTP4					P10/$\overline{\text{SCK10}}$/TxD61
INTPR60			Ext.: PD Int.: HZ		P14/SEG22/TIOP20/ RxD60
<INTPR60>			HZ		P70/CRxD[Note]/<RxD60>
INTPR61					P11/SI10/RxD61
OCD0A	I/O	On-chip debug mode setting connection	HZ	Input	P121/X1
OCD0B					P122/X2/EXCLK
OCD1A			Ext.: PD Int.: HZ		P31/SEG5
OCD1B					P32/SEG6
PCL	Output	Clock output (for trimming of high-speed system clock, subsystem clock)	HZ	Input	P72/SGOA
REGC	–	This is the pin for connecting regulator output (2.5 V) stabilization capacitance for internal operation. Connect this pin to V_{SS} via a capacitor (0.47 to 1 μF: recommended).	–	–	–
$\overline{\text{RESET}}$	Input	System reset input	–	–	–

Note μPD78F0846, 78F0847, 78F0848, and 78F0849 only.

Remarks 1. Ext. (external reset): POC reset or pin reset, Int. (internal reset): WDT reset or LVI reset, PD: Pull down, HZ: High impedance, L: Low level output

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

Table 2-5. Non-port Pins for 78K0/DF2 (2/4)

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
RxD60	Input	Serial data input to asynchronous serial interface	Ext.: PD Int.: HZ	Input	P14/SEG22/TIOP20/INTPR60
<RxD60>			HZ		P70/CRxD[Note1]/<INTPR60>
RxD61					P11/SI10/INTPR61
$\overline{\text{SCK10}}$	I/O	Clock input/output for serial interface	HZ	Input	P10/TxD61/INTP4
$\overline{\text{SCK11}}$			Ext.: PD Int.: HZ		P74/SEG24[Note2]
SCL0	I/O	Clock input/output for I^{2}C 5 V tolerant/N-ch open-drain output selectable.	HZ	Input	P60/INTP1
SDA0	I/O	Serial data I/O for I^{2}C 5 V tolerant/N-ch open-drain output selectable.	HZ	Input	P61/INTP3
SEG0 to SEG3	Output	LCD controller/driver segment signal outputs	Ext.: PD Int.: L	–	–
SEG4			Ext.: PD Int.: HZ	Input	P30
SEG5					P31/OCD1A
SEG6					P32/OCD1B
SEG7 to SEG11					P33 to P37
SEG12					P00/TIOP40
SEG13					P01/TIOP41
SEG14					P02/TIO50
SEG15					P03/TIO51
SEG16					P04/TIOP01
SEG17					P05/TIOP11
SEG18					P06/TIOP21
SEG19					P07/TIOP31
SEG20					P16/TIOP00
SEG21					P15/TIOP10
SEG22					P14/TIOP20/RxD60/INTPR60
SEG23					P13/TIOP30/TxD60
SEG24					P74/$\overline{\text{SCK11}}$[Note2] P80[Note3]
SEG25					P75/SI11[Note2] P81[Note3]

Notes 1. μPD78F0846, 78F0847, 78F0848, and 78F0849 only.

2. For μPD78F0842, 78F0843, 78F0848, and 78F0849.

3. For μPD78F0838, 78F0839, 78F0840, 78F0841, 78F0846, and 78F0847.

Remarks 1. Ext. (external reset): POC reset or pin reset, Int. (internal reset): WDT reset or LVI reset,
PD: Pull down, HZ: High impedance, L: Low level output

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

Table 2-5. Non-port Pins for 78K0/DF2 (3/4)

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
SEG26	Output	LCD controller/driver segment signal outputs	Ext.: PD Int.: HZ	Input	P76/SO11[Note1] P82[Note2]
SEG27					P77/$\overline{\text{SSI11}}$[Note1]/<TIOP20> P83[Note2]
SEG28 to SEG31[Note3]					P84 to P87
SEG32 to SEG39[Note4]					P90 to P97
SGO	Output	Sound generator output	HZ	Input	P73/SGOF/BUZ
SGOA		Sound generator amplitude PWM output			P72/PCL
SGOF		Sound generator frequency output			P73/SGO/BUZ
SI10	Input	Serial data input to serial interface	HZ	Input	P11/RxD61/INTPR61
SI11			Ext.: PD Int.: HZ		P75/ SEG25[Note1]
SM11[Note5]	Output	Stepper motor 1 output sin +	Ext.: PD Int.: HZ	Input	P80
SM12[Note5]		Stepper motor 1 output sin –			P81
SM13[Note5]		Stepper motor 1 output cos +			P82
SM14[Note5]		Stepper motor 1 output cos –			P83/ZPD14
SM21[Note5]		Stepper motor 2 output sin +			P84
SM22[Note5]		Stepper motor 2 output sin –			P85
SM23[Note5]		Stepper motor 2 output cos +			P86
SM24[Note5]		Stepper motor 2 output cos –			P87/ZPD24
SM31[Note6]		Stepper motor 3 output sin +			P90
SM32[Note6]		Stepper motor 3 output sin –			P91
SM33[Note6]		Stepper motor 3 output cos +			P92
SM34[Note6]		Stepper motor 3 output cos –			P93/ZPD34
SM41[Note6]		Stepper motor 4 output sin +			P94
SM42[Note6]		Stepper motor 4 output sin –			P95
SM43[Note6]		Stepper motor 4 output cos +			P96
SM44[Note6]		Stepper motor 4 output cos –			P97/ZPD44
SMV$_{DD}$	–	Stepper motor controller/driver supply voltage	–	–	–
SMV$_{SS}$	–	Stepper motor controller/driver ground potential	–	–	–
SO10	Output	Serial data output from serial interface	HZ	Input	P12/INTP2
SO11			Ext.: PD Int.: HZ		P76/ SEG26[Note1]

Notes 1. For μPD78F0842, 78F0843, 78F0848, and 78F0849.

2. For μPD78F0838, 78F0839, 78F0840, 78F0841, 78F0846, and 78F0847.

3. μPD78F0838, 78F0839, 78F0840, 78F0841, 78F0846, and 78F0847 only.

4. μPD78F0838 and 78F0839 only.

5. μPD78F0842, 78F0843, 78F0848, and 78F0849 only.

6. μPD78F0840, 78F0841, 78F0842, 78F0843, 78F0846, 78F0847, 78F0848, and 78F0849 only.

Remarks 1. Ext. (external reset): POC reset or pin reset, Int. (internal reset): WDT reset or LVI reset, PD: Pull down, HZ: High impedance

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

Table 2-5. Non-port Pins for 78K0/DF2 (4/4)

Pin Name	I/O	Function	During Reset	After Reset	Alternate Function
$\overline{SSI11}$	Input	Serial interface chip select input	Ext.: PD Int.: HZ	Input	P77/<TIOP20>/SEG27[Note1]
TIO50	I/O	External count clock input/timer output (TM50)	Ext.: PD Int.: HZ	Input	P02/SEG14
TIO51		External count clock input/timer output (TM51)			P03/SEG15
TIOP00	I/O	External event count input/capture trigger input/external trigger input/timer output (TMP0)	Ext.: PD Int.: HZ	Input	P16/SEG20
TIOP01		Capture trigger input/timer output (TMP0)			P04/SEG16
TIOP10		External event count input/capture trigger input/external trigger input/timer output (TMP1)			P15/SEG21
TIOP11		Capture trigger input/timer output (TMP1)			P05/SEG17
TIOP20		External event count input/capture trigger input/external trigger input/timer output (TMP2)			P14/SEG22/RxD60/INTPR60
<TIOP20>					P77/$\overline{SSI11}$/SEG27[Note1]
TIOP21		Capture trigger input/timer output (TMP2)			P06/SEG18
TIOP30		External event count input/capture trigger input/external trigger input/timer output (TMP3)			P13/SEG23/TxD60
<TIOP30>			HZ		P17/INTP0
TIOP31		Capture trigger input/timer output (TMP3)	Ext.: PD Int.: HZ		P07/SEG19
TIOP40		External event count input/capture trigger input/external trigger input/timer output (TMP4)			P00/SEG12
TIOP41		Capture trigger input/timer output (TMP4)			P01/SEG13
TxD60	Output	Serial data output from asynchronous serial interface	Ext.: PD Int.: HZ	Input	P13/SEG23/TIOP30
<TxD60>			HZ		P71/CTxD[Note2]
TxD61					P10/$\overline{SCK10}$/INTP4
V_{DD}/EV_{DD}	–	Positive power supply (except for ports)/positive power supply for ports	–	–	–
V_{SS}/EV_{SS}	–	Ground potential (except for ports)/ground potential for ports	–	–	–
X1	Input	Connecting resonator for high-speed system clock	HZ	Input	P121/OCD0A
X2	–				P122/EXCLK/OCD0B
XT1	Input	Connecting resonator for subsystem clock	HZ	Input	P123
XT2	–				P124/EXCLKS
ZPD14[Note3]	Input	Zero point detection input	Ext.: PD Int.: HZ	Input	P83/SM14
ZPD24[Note3]					P87/SM24
ZPD34[Note4]					P93/SM34
ZPD44[Note4]					P97/SM44

Notes 1. For μPD78F0842, 78F0843, 78F0848, and 78F0849.

2. μPD78F0846, 78F0847, 78F0848, and 78F0849 only.

3. μPD78F0842, 78F0843, 78F0848, and 78F0849 only.

4. μPD78F0840, 78F0841, 78F0842, 78F0843, 78F0846, 78F0847, 78F0848, and 78F0849 only.

Remarks 1. Ext. (external reset): POC reset or pin reset, Int. (internal reset): WDT reset or LVI reset, PD: Pull down, HZ: High impedance

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

2.2 Description of Pin Functions

2.2.1 P00 to P07 (port 0)

P00 to P07 function as an 8-bit I/O port. These pins also function as timer I/O and segment signal outputs for the LCD controller/driver.

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

(1) Port mode

P00 to P07 function as 8-bit I/O port. P00 to P07 can be set to input or output in 1-bit units using port mode register 0 (PM0). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 0 (PU0).

(2) Control mode

P00 to P07 function as timer I/O and segment signal outputs for the LCD controller/driver.

(a) TIO50, TIO51

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

(b) TIOP01, TIOP11, TIOP21, TIOP31, TIOP41

These are the pins for capture trigger inputs and timer outputs (TMP0 to TMP4).

(c) TIOP40

This is the pin for external event count input, capture trigger input, external trigger input, and timer output (TMP4).

(d) SEG12 to SEG19

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

2.2.2 P10 to P17 (port 1)

P10 to P17 function as an 8-bit I/O port. These pins also function as pins for serial interface data I/O, clock I/O, timer I/O, external interrupt request input, and segment signal outputs for the LCD controller/driver.

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 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 serial interface data I/O, clock I/O, timer I/O, and segment signal outputs for the LCD controller/driver.

(a) SI10

This is a serial interface serial data input pin.

(b) SO10

This is a serial interface serial data output pin.

(c) $\overline{\textbf{SCK10}}$

This is a serial interface serial clock I/O pin.

(d) RxD60, RxD61[Note]

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

Note RxD61 is for 78K0/DF2 only.

(e) TxD60, TxD61[Note]

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

Note TxD61 is for 78K0/DF2 only.

(f) TIOP00, TIOP10, TIOP20, TIOP30

These are the pins for external event count inputs, capture trigger inputs, external trigger inputs, and timer outputs (TMP0 to TMP3).

(g) INTP0, INTP2, INTP4, INTPR60, INTPR61[Note]

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

Note INTPR61 is for 78K0/DF2 only.

(h) SEG20 to SEG23

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

2.2.3 P20 to P27 (port 2)

P20 to P27 function as an 8-bit I/O port in 78K0/DF2 and P20 to P23 function as a 4-bit I/O port in 78K0/DE2. These pins also function as pins for A/D converter analog input.

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 in 78K0/DF2 and P20 to P23 function as a 4-bit I/O port in 78K0/DE2. These ports can be set to input or output in 1-bit units using port mode register 2 (PM2).

(2) Control mode

P20 to P27 function as A/D converter analog input pins (ANI0 to ANI7 in 78K0/DF2, ANI0 to ANI3 in 78K0/DE2). When using these pins as analog input pins, see **11.6 (5) P20/ANI0 to P27/ANI7**.

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

2.2.4 P30 to P37 (port 3)

P30 to P37 function as an 8-bit I/O port. These pins also function as pins for segment signal outputs for the LCD controller/driver.

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

(1) Port mode

P30 to P37 function as an 8-bit I/O port. P30 to P37 can be set to input or output 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 P37 function as segment signal outputs for the LCD controller/driver.

(a) SEG4 to SEG11

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

<R> **Cautions 1. Be sure to pull the P31/SEG5/OCD1A pin down before a reset release, to prevent malfunction.**

2. Set P31 and P32 as follows when these pins are used for on-chip debug. In this case, P31/SEG5/OCD1A and P32/SEG6/OCD1B pins are used only as OCD1A or OCD1B functions.
- P31/SEG5/OCD1A: Set to input mode (PM31 = 1) and port mode (PF31 = 0).
- P32/SEG6/OCD1B: Set to port mode (PF32 = 0).
When P31/SEG5 and P32/SEG6 functions need to be evaluated with on-chip debug, use OCD0A and OCD0B.

Remark P31/SEG5 and P32/SEG6 can be used as on-chip debug mode setting pins (OCD1A, OCD1B) when the on-chip debug function is used. For how to connect an in-circuit emulator supporting on-chip debugging (QB-78K0MINI or QB-MINI2), see **CHAPTER 27 ON-CHIP DEBUG FUNCTION**.

2.2.5 P60, P61 (port 6)

P60 and P61 function as a 2-bit I/O port. P60 and P61 can be set to input port or output port in 1-bit units using port mode register 6 (PM6). P60 and P61 use of an on-chip pull-up resistor can be specified by pull-up resistor option register 6 (PU6).

(1) Port mode

P60 and P61 function as a 2-bit I/O port. P60 and P61 can be set to input or output in 1-bit units using port mode register 6 (PM6). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 6 (PU6).

(2) Control mode

P60 and P61 function as serial clock I/O and data I/O for IIC0.

(a) INTP1, INTP3

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) SCL0

This is a serial clock I/O pin for serial interface IIC0.

(c) SDA0

This is a serial data I/O pin for serial interface IIC0.

2.2.6 P70 to P77 (port 7)

P70 to P73 function as a 4-bit I/O port in 78K0/DE2 and P70 to P77 function as an 8-bit I/O port in 78K0/DF2. These pins also function as output pins for the sound generator, clock output pins, buzzer output pins, CAN I/F I/O, serial interface data I/O and clock I/O, timer I/O, and segment output pins for the LCD controller/driver.

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

(1) Port mode

P70 to P73 function as a 4-bit I/O port in 78K0/DE2 and P70 to P77 function as an 8-bit I/O port in 78K0/DF2. P70 to P77 can be set to input or output in 1-bit units using port mode register 7 (PM7). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 7 (PU7).

(2) Control mode

These pins function as output pins for the sound generator, clock output pins, buzzer output pins, CAN I/F I/O, serial interface data I/O and clock I/O, timer I/O, and segment output pins for the LCD controller/driver.

(a) SGOA

This is the amplitude PWM output pin for the sound generator.

(b) SGO

This is the output pin for the sound generator.

(c) SGOF

This is the frequency output pin for the sound generator.

(d) PCL

This is a clock output pin.

(e) BUZ

This is a buzzer output pin.

(f) RxD60

This is the serial data input pin of the asynchronous serial interface.

(g) TxD60

This is the serial data output pin of the asynchronous serial interface.

(h) INTPR60

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.

(i) CRxD (μPD78F0844, 78F0845, 78F0846, 78F0847, 78F0848, and 78F0849 only)

This is the CAN serial receive data input pin.

(j) CTxD (μPD78F0844, 78F0845, 78F0846, 78F0847, 78F0848, and 78F0849 only)

This is the CAN serial transmit data output pin.

(k) $\overline{\text{SCK11}}$ (78K0/DF2 only)

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

(l) SI11 (78K0/DF2 only)

This is a serial interface serial data input pin.

(m) SO11 (78K0/DF2 only)

This is a serial interface serial data output pin.

(n) $\overline{\text{SSI11}}$ (78K0/DF2 only)

This is the serial interface chip select input pin.

(o) TIOP20 (78K0/DF2 only)

This is the pin for external event count input, capture trigger input, external trigger input, and timer output (TMP2).

(p) SEG24 to SEG27 (μPD78F0842, 78F0843, 78F0848, and 78F0849 only)

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

2.2.7 P80 to P87 (port 8)

P80 to P87 function as an 8-bit I/O port. These pins also function as stepper motor controller/driver outputs/inputs or LCD segment outputs.

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

(1) Port mode

P80 to P87 function as an 8-bit I/O port. P80 to P87 can be set to input or output in 1-bit units using port mode register 8 (PM8).

(2) Control mode

P80 to P87 function as stepper motor controller/driver outputs/inputs or LCD segment outputs.

(a) SEG24 to SEG31 (μPD78F0838, 78F0839, 78F0840, 78F0841, 78F0846, and 78F0847 only)

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

(b) SM11 to SM14, SM21 to SM24 (μPD78F0836, 78F0837, 78F0842, 78F0843, 78F0844, 78F0845, 78F0848, and 78F0849 only)

These are the output pins for the stepper motor controller/driver.

(c) ZPD14, ZPD24 (μPD78F0836, 78F0837, 78F0842, 78F0843, 78F0844, 78F0845, 78F0848, and 78F0849 only)

These are the Zero Point Detection (ZPD) input pins for the stepper motor controller/driver.

2.2.8 P90 to P97 (port 9) (78K0/DF2 only)

P90 to P97 function as an 8-bit I/O port. These pins also function as stepper motor controller/driver outputs/inputs or LCD segment outputs.

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

(1) Port mode

P90 to P97 function as an 8-bit I/O port. P90 to P97 can be set to input or output in 1-bit units using port mode register 9 (PM9).

(2) Control mode

P80 to P87 function as stepper motor controller/driver outputs/inputs or LCD segment outputs.

(a) SEG32 to SEG39 (μPD78F0838, 78F0839 only)

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

(b) SM31 to SM34, SM41 to SM44 (μPD78F0840, 78F0841, 78F0842, 78F0843, 78F0846, 78F0847, 78F0848, and 78F0849 only)

These are the output pins for the stepper motor controller/driver.

(c) ZPD34, ZPD44 (μPD78F0840, 78F0841, 78F0842, 78F0843, 78F0846, 78F0847, 78F0848, and 78F0849 only)

These are the Zero Point Detection (ZPD) input pins for the stepper motor controller/driver.

2.2.9 P120 to P124 (port 12)

P120 to P124 function as a 5-bit I/O port. These pins also function as external clock input for main system clock, external clock input for subsystem clock and potential input for external low-voltage detection. The following operation modes can be specified in 1-bit units.

(1) Port mode

P120 to P124 function as a 5-bit I/O port. P120 to P124 can be set to input or output 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 potential input for external low-voltage detection, resonator connection for main system clock, resonator connection for subsystem clock, external clock input for main system clock and external clock input for subsystem clock.

(a) EXLVI

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

(b) X1, X2

These are the pins for connecting a resonator for high-speed system clock.
When supplying an external clock, input a signal to the X1 pin and input the inverse signal to the X2 pin.

Caution Connect P121/X1 as follows when writing the flash memory with a flash programmer.
- P121/X1: When using this pin as a port, connect it to V_{SS} via a resistor (10 kΩ: recommended) (in the input mode) or leave it open (in the output mode).
The above connection is not necessary when writing the flash memory by means of self programming.

Remark The P121/X1 and P122/X2/EXCLK pins can be used as on-chip debug mode setting pins (OCD0A, OCD0B) when the on-chip debug function is used. For how to connect an in-circuit emulator supporting on-chip debugging (QB-78K0MINI or QB-MINI2), see **CHAPTER 27 ON-CHIP DEBUG FUNCTION**.

(c) EXCLK

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

(d) XT1, XT2

These are the pins for connecting a resonator for subsystem clock.
When supplying an external clock, input a signal to the XT1 pin and input the inverse signal to the XT2 pin.

(e) EXCLKS

This is an external clock input pin for subsystem clock.

2.2.10 AV$_{REF}$

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

When the A/D converter is not used, connect this pin directly to V$_{DD}$/EV$_{DD}$.

2.2.11 AV$_{SS}$

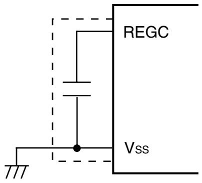
This is the A/D converter ground potential pin. Even when the A/D converter is not used, always use this pin with the same potential as the V$_{SS}$/EV$_{SS}$ pin.

2.2.12 $\overline{\text{RESET}}$

This is the active-low system reset input pin.

2.2.13 REGC

This is the pin for connecting regulator output (2.5 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 for the broken-line part in the above figure.

2.2.14 V$_{DD}$/EV$_{DD}$

V$_{DD}$ is the positive power supply pin for other than ports.

EV$_{DD}$ is the positive power supply pin for ports.

2.2.15 V$_{SS}$/EV$_{SS}$

V$_{SS}$ is the ground potential pin for other than ports.

EV$_{SS}$ is the ground potential pin for ports.

2.2.16 FLMD0

This is a pin for setting flash memory programming mode.

Connect to V$_{SS}$/EV$_{SS}$ in the normal operation mode. In flash memory programming mode, be sure to connect this pin to the flash memory programmer.

2.2.17 SEG0 to SEG3

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

2.2.18 COM0 to COM3

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

2.2.19 SMV$_{DD}$

This is the power supply voltage pin for the stepper motor controller/driver.

2.2.20 SMV$_{SS}$

This is the ground pin for the stepper motor controller/driver.

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

The following table shows the types of pin I/O circuits and the recommended connections of unused pins. Refer to **Figure 2-1 Pin I/O Circuit List** for the configuration of the I/O circuit of each type.

Table 2-6. Pin I/O Circuit Types (1/8)

(a) 78K0/DE2 (μPD78F0836, 78F0837, 78F0844, 78F0845) (1/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/SEG12/TIOP40	17-W	I/O	Input: Independently connect to V_{SS} via a resistor. Output: Leave open.
P01/SEG13/TIOP41			
P02/SEG14/TIO50			
P03/SEG15/TIO51			
P04/SEG16/TIOP01			
P05/SEG17/TIOP11			
P06/SEG18/TIOP21			
P07/SEG19/TIOP31			
P10/$\overline{\text{SCK10}}$/INTP4	5-AW		
P11/SI10			
P12/SO10/INTP2			
P13/SEG23/TIOP30/TxD60	17-W		
P14/SEG22/TIOP20/RxD60/INTPR60			
P15/SEG21/TIOP10			
P16/SEG20/TIOP00			
P17/INTP0/<TIOP30>	5-AW		
P20/ANI0 to P23/ANI3[Note 1]	11-G	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P30/SEG4	17-V	I/O	Input: Independently connect to V_{SS} via a resistor. Output: Leave open.
P31/SEG5/OCD1A[Note 2, 3]	17-W		
P32/SEG6/OCD1B[Note 2]			
P33/SEG7 to P35/SEG9			
P36/SEG10	17-V		
P37/SEG11	17-W		

Notes 1. P20/ANI0 to P27/ANI3 are set in the analog input mode after release of reset.

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

<R> **3.** Set P31 and P32 as follows when these pins are used for on-chip debug. In this case, P31/SEG5/OCD1A and P32/SEG6/OCD1B pins are used only as OCD1A or OCD1B functions.

- P31/SEG5/OCD1A: Set to input mode (PM31 = 1) and port mode (PF31 = 0).
- P32/SEG6/OCD1B: Set to port mode (PF32 = 0).

When P31/SEG5 and P32/SEG6 functions need to be evaluated with on-chip debug, use OCD0A and OCD0B.

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

Table 2-6. Pin I/O Circuit Types (2/8)
(a) 78K0/DE2 (μPD78F0836, 78F0837, 78F0844, 78F0845) (2/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P60/SCL0/INTP1	5-AH	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
<R> P61/SDA0/INTP3			
P70/CRxD[Note 1]/<RXD60/INTPR60>	5-AH	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P71/CTxD[Note 1]/<TxD60>	5-AG		
P72/SGOA/PCL			
P73/SGO/SGOF/BUZ			
P80/SM11 to P82/SM13	5-AU	I/O	Input: Independently connect to SMV_{SS} via a resistor. Output: Leave open.
P83/SM14/ZPD14	5-AX		
P84/SM21 to P86/SM23	5-AU		
P87/SM24/ZPD24	5-AX		
P120/EXLVI	5-AH	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P121/X1/OCD0A[Note 2, 3]	37-D		
P122/X2/EXCLK/OCD0B[Note 2]			
P123/XT1[Note 2]			
P124/XT2/EXCLKS[Note 2]			
SEG0 to SEG3	17-U	Output	Leave open.
COM0 to COM3	18-G		
$\overline{\text{RESET}}$	2	Input	–
AV_{REF}	–	–	Connect directly to V_{DD}.
AV_{SS}	–	–	Connect directly to V_{SS}.
SMV_{DD}	–	–	Connect directly to V_{DD}.
SMV_{SS}	–	–	Connect directly to V_{SS}.
<R> FLMD0	38	Input	Connect directly to V_{SS}.

Notes 1. μPD78F0844 and 78F0845 only.

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

3. Connect P121/X1 as follows when writing the flash memory with a flash memory programmer.

- P121/X1: When using this pin as a port, connect it to EV_{SS} via a resistor (10 kΩ: recommended) (in the input mode) or leave it open (in the output mode).

The above connection is not necessary when writing the flash memory by means of self programming.

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

Table 2-6. Pin I/O Circuit Types (3/8)

(b) 78K0/DF2 (μPD78F0838, 78F0839) (1/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/SEG12/TIOP40	17-W	I/O	Input: Independently connect to V_{SS} via a resistor. Output: Leave open.
P01/SEG13/TIOP41			
P02/SEG14/TIO50			
P03/SEG15/TIO51			
P04/SEG16/TIOP01			
P05/SEG17/TIOP11			
P06/SEG18/TIOP21			
P07/SEG19/TIOP31			
P10/$\overline{\text{SCK10}}$/TxD61/INTP4	5-AH		
P11/SI10/RxD61/INTPR61			
P12/SO10/INTP2			
P13/SEG23/TIOP30/TxD60	17-W		
P14/SEG22/TIOP20/RxD60/INTPR60			
P15/SEG21/TIOP10			
P16/SEG20/TIOP00			
P17/INTP0/<TIOP30>	5-AH		
P20/ANI0 to P27/ANI7[Note 1]	11-G	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P30/SEG4	17-V	I/O	Input: Independently connect to V_{SS} via a resistor. Output: Leave open.
P31/SEG5/OCD1A[Note 2, 3]	17-W		
P32/SEG6/OCD1B[Note 2]			
P33/SEG7 to P35/SEG9			
P36/SEG10	17-V		
P37/SEG11	17-W		
P60/SCL0/INTP1	5-AH	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
<R> P61/SDA0/INTP3			

Notes 1. P20/ANI0 to P27/ANI7 are set in the analog input mode after release of reset.

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

<R> **3.** Set P31 and P32 as follows when these pins are used for on-chip debug. In this case, P31/SEG5/OCD1A and P32/SEG6/OCD1B pins are used only as OCD1A or OCD1B functions.
- P31/SEG5/OCD1A: Set to input mode (PM31 = 1) and port mode (PF31 = 0).
- P32/SEG6/OCD1B: Set to port mode (PF32 = 0).

When P31/SEG5 and P32/SEG6 functions need to be evaluated with on-chip debug, use OCD0A and OCD0B.

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

Table 2-6. Pin I/O Circuit Types (4/8)
(b) 78K0/DF2 (μPD78F0838, 78F0839) (2/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P70/<RXD60/INTPR60>	5-AH	I/O	Input: Independently connect to V_{SS} via a resistor. Output: Leave open.
P71/<TxD60>	5-AG		
P72/SGOA/PCL			
P73/SGO/SGOF/BUZ			
P74/$\overline{\text{SCK11}}$	5-AW		
P75/SI11			
P76/SO11	5-AU		
P77/$\overline{\text{SSI11}}$/<TIOP20>	5-AW		
P80/SEG24 to P87/SEG31	17-X	I/O	Input: Independently connect to SMV_{SS} via a resistor. Output: Leave open.
P90/SEG32 to P93/SEG35	17-W		
P94/SEG36 to P97/SEG39	17-X		
P120/EXLVI	5-AH	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P121/X1/OCD0A[Note 1, 2]	37-D		
P122/X2/EXCLK/OCD0B[Note 1]			
P123/XT1[Note 1]			
P124/XT2/EXCLKS[Note 1]			
SEG0 to SEG3	17-U	Output	Leave open.
COM0 to COM3	18-G		
$\overline{\text{RESET}}$	2	Input	–
AV_{REF}	–	–	Connect directly to V_{DD}.
AV_{SS}	–	–	Connect directly to V_{SS}.
SMV_{DD}	–	–	Connect directly to V_{DD}.
SMV_{SS}	–	–	Connect directly to V_{SS}.
<R> FLMD0	38	Input	Connect directly to V_{SS}.

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

2. Connect P121/X1 as follows when writing the flash memory with a flash memory programmer.

- P121/X1: When using this pin as a port, connect it to V_{SS} via a resistor (10 kΩ: recommended) (in the input mode) or leave it open (in the output mode).

The above connection is not necessary when writing the flash memory by means of self programming.

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

Table 2-6. Pin I/O Circuit Types (5/8)
(c) 78K0/DF2 (μPD78F0840, 78F0841, 78F0846, 78F0847) (1/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/SEG12/TIOP40	17-W	I/O	Input: Independently connect to V_{SS} via a resistor. Output: Leave open.
P01/SEG13/TIOP41			
P02/SEG14/TIO50			
P03/SEG15/TIO51			
P04/SEG16/TIOP01			
P05/SEG17/TIOP11			
P06/SEG18/TIOP21			
P07/SEG19/TIOP31			
P10/$\overline{\text{SCK10}}$/TxD61/INTP4	5-AH		
P11/SI10/RxD61/INTPR61			
P12/SO10/INTP2			
P13/SEG23/TIOP30/TxD60	17-W		
P14/SEG22/TIOP20/RxD60/INTPR60			
P15/SEG21/TIOP10			
P16/SEG20/TIOP00			
P17/INTP0/<TIOP30>	5-AH		
P20/ANI0 to P27/ANI7[Note 1]	11-G	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P30/SEG4	17-V	I/O	Input: Independently connect to V_{SS} via a resistor. Output: Leave open.
P31/SEG5/OCD1A[Note 2, 3]	17-W		
P32/SEG6/OCD1B[Note 2]			
P33/SEG7 to P35/SEG9			
P36/SEG10	17-V		
P37/SEG11	17-W		
P60/SCL0/INTP1	5-AH	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave these pins open after clearing the output latch of the port to "0".
P61/SDA0/INTP3			

Notes 1. P20/ANI0 to P27/ANI7 are set in the analog input mode after release of reset.

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

<R> **3.** Set P31 and P32 as follows when these pins are used for on-chip debug. In this case, P31/SEG5/OCD1A and P32/SEG6/OCD1B pins are used only as OCD1A or OCD1B functions.

- P31/SEG5/OCD1A: Set to input mode (PM31 = 1) and port mode (PF31 = 0).
- P32/SEG6/OCD1B: Set to port mode (PF32 = 0).

When P31/SEG5 and P32/SEG6 functions need to be evaluated with on-chip debug, use OCD0A and OCD0B.

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

Table 2-6. Pin I/O Circuit Types (6/8)

(c) 78K0/DF2 (μPD78F0840, 78F0841, 78F0846, 78F0847) (2/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P70/CRxD[Note 1]/<RXD60/INTPR60>	5-AH	I/O	Input: Independently connect to V_{SS} via a resistor. Output: Leave open.
P71/CTxD[Note 1]/<TxD60>	5-AG		
P72/SGOA/PCL			
P73/SGO/SGOF/BUZ			
P74/$\overline{\text{SCK11}}$	5-AW		
P75/SI11			
P76/SO11	5-AU		
P77/$\overline{\text{SSI11}}$/<TIOP20>	5-AW		
P80/SEG24 to P87/SEG31	17-X	I/O	Input: Independently connect to SMV_{SS} via a resistor. Output: Leave open.
P90/SM31 to P92/SM33	5-AW		
P93/SM34/ZPD34	5-AV		
P94/SM41 to P96/SM43	5-AU		
P97/SM44/ZPD44	5-AX		
P120/EXLVI	5-AH	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P121/X1/OCD0A[Note 2, 3]	37-D		
P122/X2/EXCLK/OCD0B[Note 2]			
P123/XT1[Note 2]			
P124/XT2/EXCLKS[Note 2]			
SEG0 to SEG3	17-U	Output	Leave open.
COM0 to COM3	18-G		
$\overline{\text{RESET}}$	2	Input	–
AV_{REF}	–	–	Connect directly to V_{DD}.
AV_{SS}	–	–	Connect directly to V_{SS}.
SMV_{DD}	–	–	Connect directly to V_{DD}.
SMV_{SS}	–	–	Connect directly to V_{SS}.
<R> FLMD0	38	Input	Connect directly to V_{SS}.

Notes 1. μPD78F0846 and 78F0847 only.

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

3. Connect P121/X1 as follows when writing the flash memory with a flash memory programmer.

- P121/X1: When using this pin as a port, connect it to V_{SS} via a resistor (10 kΩ: recommended) (in the input mode) or leave it open (in the output mode).

The above connection is not necessary when writing the flash memory by means of self programming.

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

Table 2-6. Pin I/O Circuit Types (7/8)
(d) 78K0/DF2 (μPD78F0842, 78F0843, 78F0848, 78F0849) (1/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/SEG12/TIOP40	17-W	I/O	Input: Independently connect to V_{SS} via a resistor. Output: Leave open.
P01/SEG13/TIOP41			
P02/SEG14/TIO50			
P03/SEG15/TIO51			
P04/SEG16/TIOP01			
P05/SEG17/TIOP11			
P06/SEG18/TIOP21			
P07/SEG19/TIOP31			
P10/$\overline{\text{SCK10}}$/TxD61/INTP4	5-AH		
P11/SI10/RxD61/INTPR61			
P12/SO10/INTP2			
P13/SEG23/TIOP30/TxD60	17-W		
P14/SEG22/TIOP20/RxD60/INTPR60			
P15/SEG21/TIOP10			
P16/SEG20/TIOP00			
P17/INTP0/<TIOP30>	5-AH		
P20/ANI0 to P27/ANI7[Note 1]	11-G	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P30/SEG4	17-V	I/O	Input: Independently connect to V_{SS} via a resistor. Output: Leave open.
P31/SEG5/OCD1A[Note 2, 3]	17-W		
P32/SEG6/OCD1B[Note 2]			
P33/SEG7 to P35/SEG9			
P36/SEG10	17-V		
P37/SEG11	17-W		
P60/SCL0/INTP1	5-AH	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave these pins open after clearing the output latch of the port to "0".
P61/SDA0/INTP3			

Notes 1. P20/ANI0 to P27/ANI7 are set in the analog input mode after release of reset.

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

<R> 3. Set P31 and P32 as follows when these pins are used for on-chip debug. In this case, P31/SEG5/OCD1A and P32/SEG6/OCD1B pins are used only as OCD1A or OCD1B functions.
- P31/SEG5/OCD1A: Set to input mode (PM31 = 1) and port mode (PF31 = 0).
- P32/SEG6/OCD1B: Set to port mode (PF32 = 0).
When P31/SEG5 and P32/SEG6 functions need to be evaluated with on-chip debug, use OCD0A and OCD0B.

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

Table 2-6. Pin I/O Circuit Types (8/8)

(d) 78K0/DF2 (μPD78F0842, 78F0843, 78F0848, 78F0849) (2/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P70/CRxD[Note 1]/<RXD60/INTPR60>	5-AH	I/O	Input: Independently connect to V_{SS} via a resistor. Output: Leave open.
P71/CTxD[Note 1]/<TxD60>	5-AG		
P72/SGOA/PCL			
P73/SGO/SGOF/BUZ			
P74/SEG24/$\overline{\text{SCK11}}$	17-W		
P75/SEG25/SI11			
P76/SEG26/SO11	17-V		
P77/SEG27/$\overline{\text{SSI11}}$/<TIOP20>	17-W		
P80/SM11 to P82/SM13	5-AU	I/O	Input: Independently connect to SMV_{SS} via a resistor. Output: Leave open.
P83/SM14/ZPD14	5-AX		
P84/SM21 to P86/SM23	5-AU		
P87/SM24/ZPD24	5-AX		
P90/SM31 to P92/SM33	5-AW		
P93/SM34/ZPD34	5-AV		
P94/SM41 to P96/SM43	5-AU		
P97/SM44/ZPD44	5-AX		
P120/EXLVI	5-AH	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P121/X1/OCD0A[Note 2, 3]	37-D		
P122/X2/EXCLK/OCD0B[Note 2]			
P123/XT1[Note 2]			
P124/XT2/EXCLKS[Note 2]			
SEG0 to SEG3	17-U	Output	Leave open.
COM0 to COM3	18-G		
$\overline{\text{RESET}}$	2	Input	–
AV_{REF}	–	–	Connect directly to V_{DD}.
AV_{SS}	–	–	Connect directly to V_{SS}.
SMV_{DD}	–	–	Connect directly to V_{DD}.
SMV_{SS}	–	–	Connect directly to V_{SS}.
<R> FLMD0	38	Input	Connect directly to V_{SS}.

Notes 1. μPD78F0848, and 78F0849 only.

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

3. Connect P121/X1 as follows when writing the flash memory with a flash memory programmer.

- P121/X1: When using this pin as a port, connect it to V_{SS} via a resistor (10 kΩ: recommended) (in the input mode) or leave it open (in the output mode).

The above connection is not necessary when writing the flash memory by means of self programming.

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

Figure 2-1. Pin I/O Circuit List (1/5)

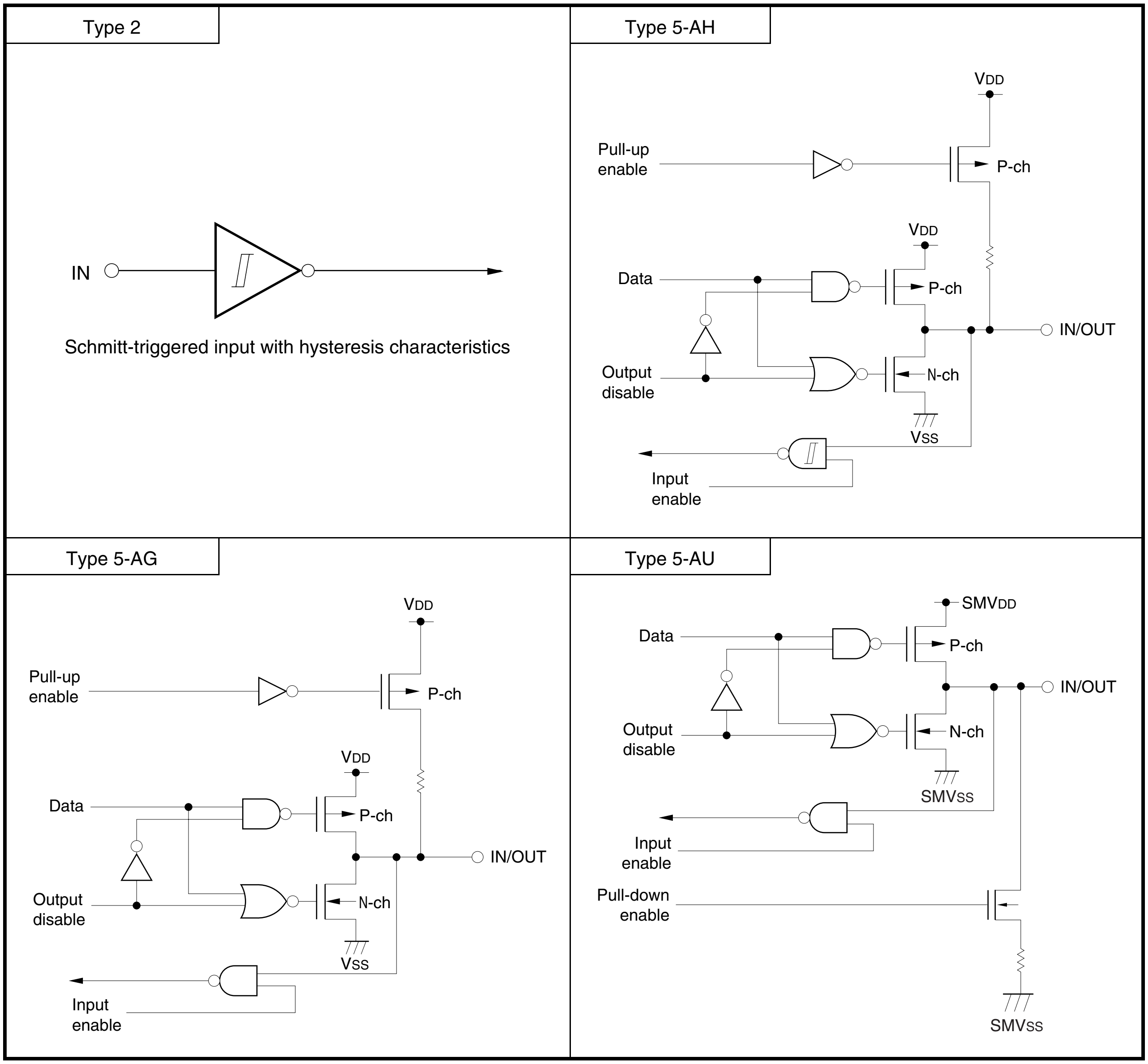


Figure 2-1. Pin I/O Circuit List (2/5)

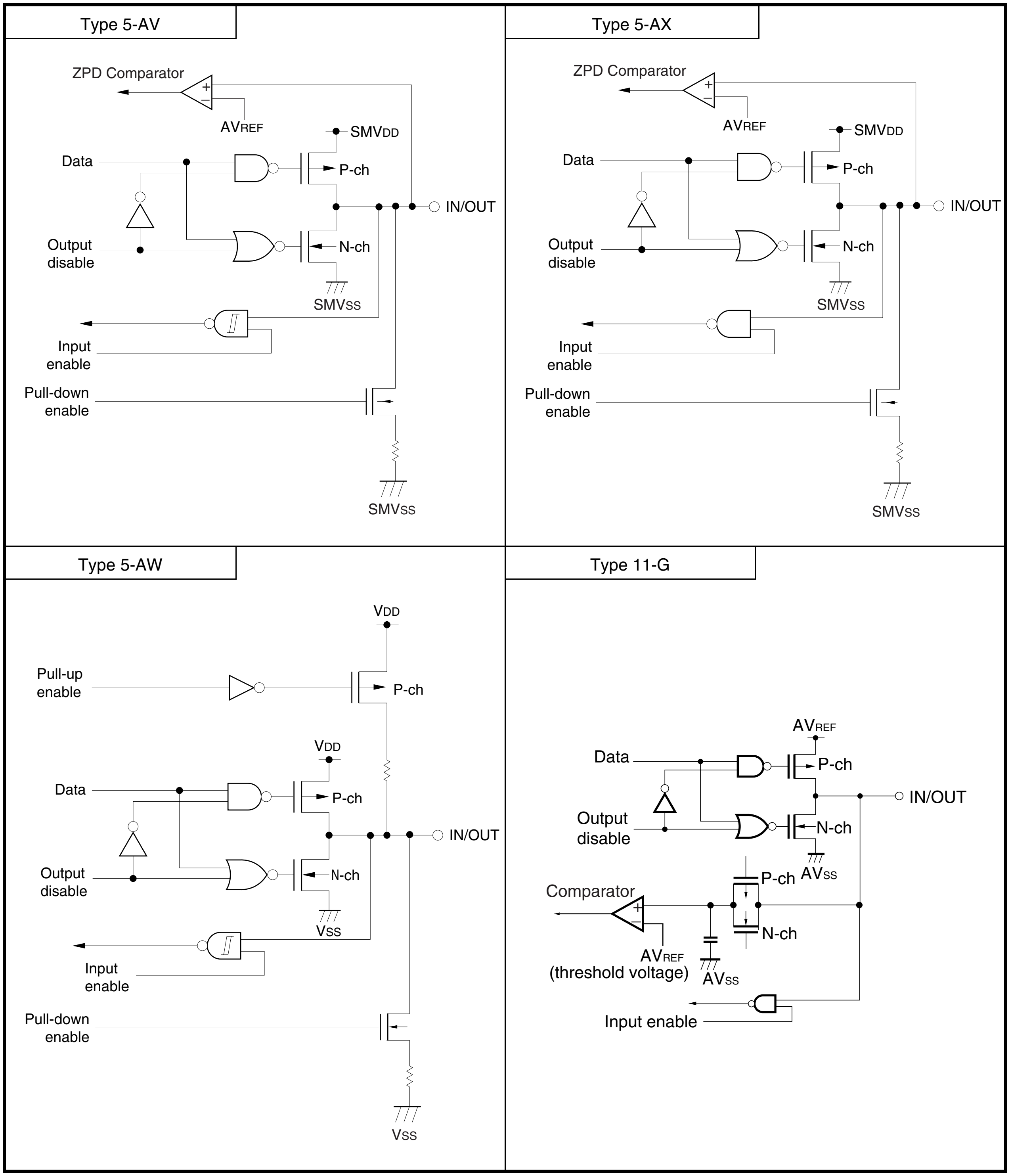


Figure 2-1. Pin I/O Circuit List (3/5)

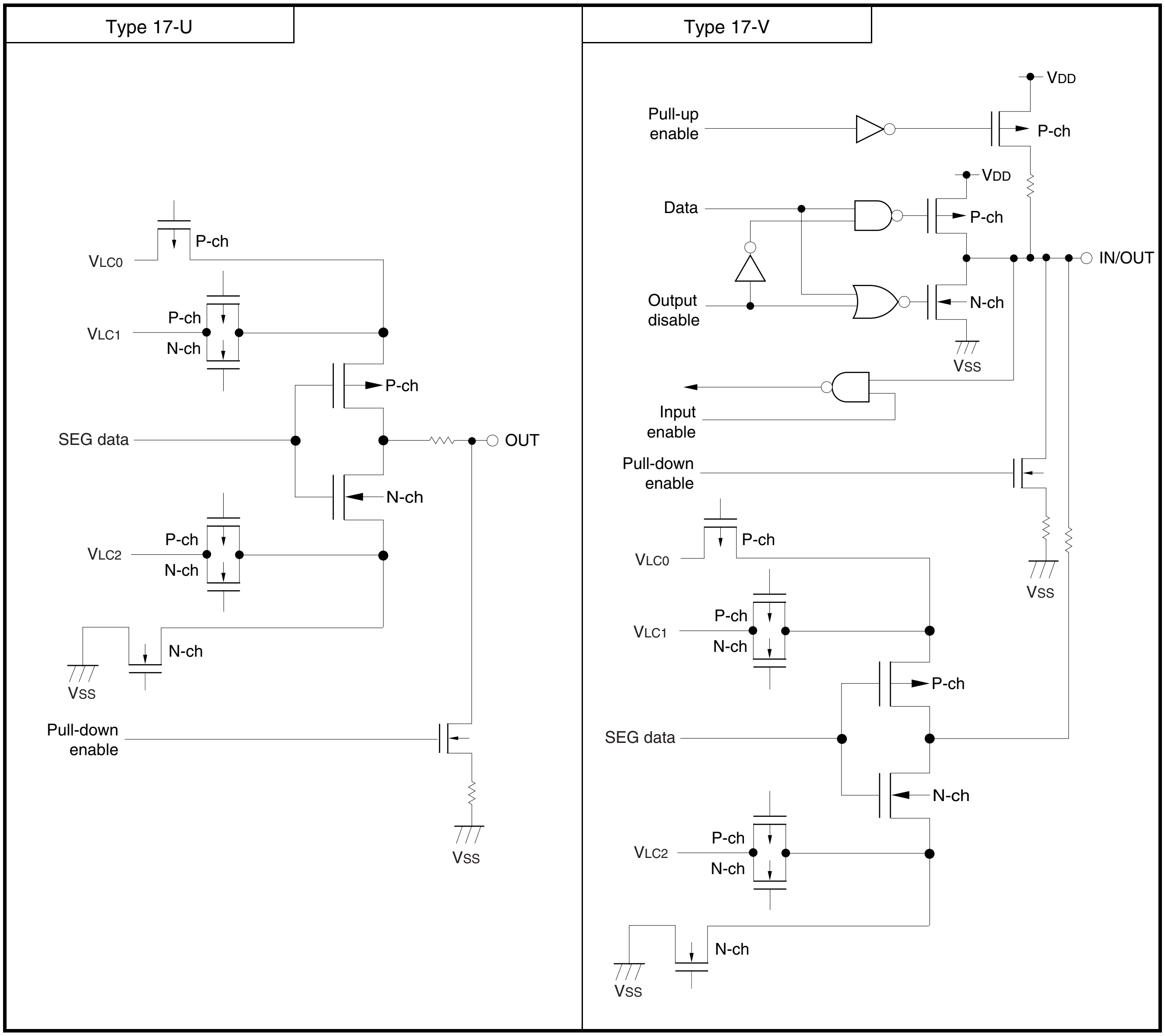


Figure 2-1. Pin I/O Circuit List (4/5)

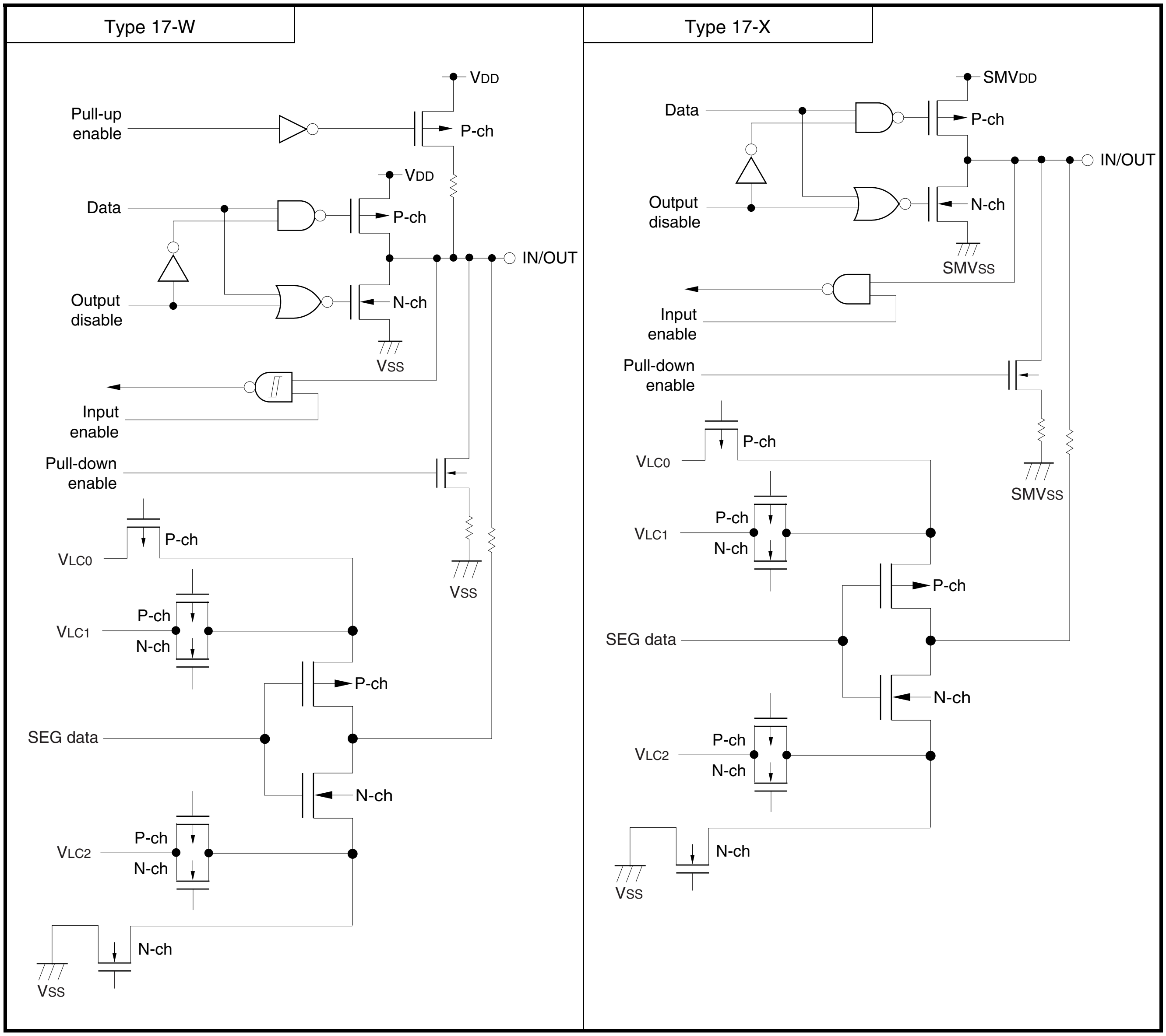
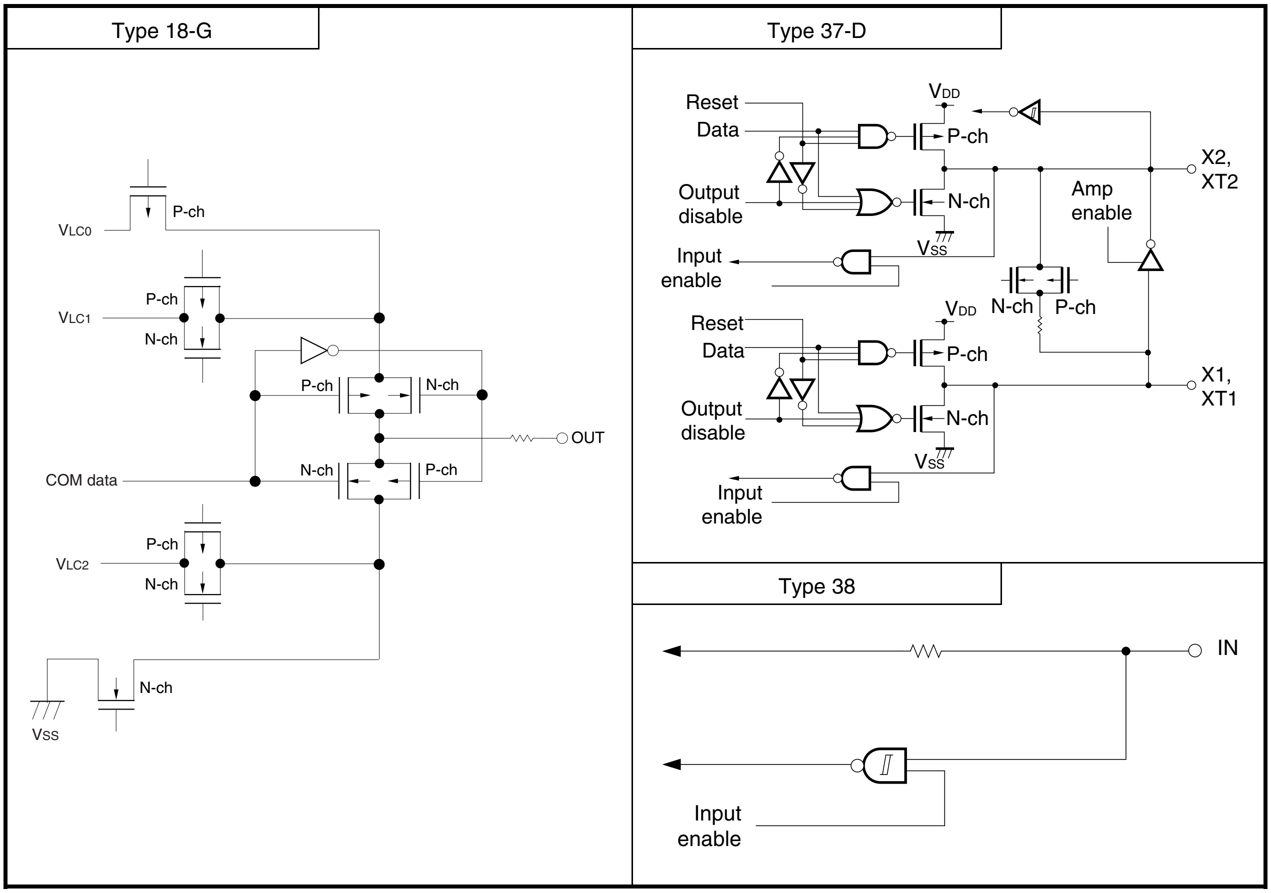


Figure 2-1. Pin I/O Circuit List (5/5)



Type 18-G

Type 37-D

Type 38

CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

Products in the 78K0/Dx2 can each access a 24, 32, 48, or 64 KB memory space. **Figures 3-1** to **3-4** show the memory maps.

Cautions 1. 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 the 78K0/Dx2 is fixed (IMS = CFH, IXS = 0CH). Therefore, set the value corresponding to each product as indicated below.

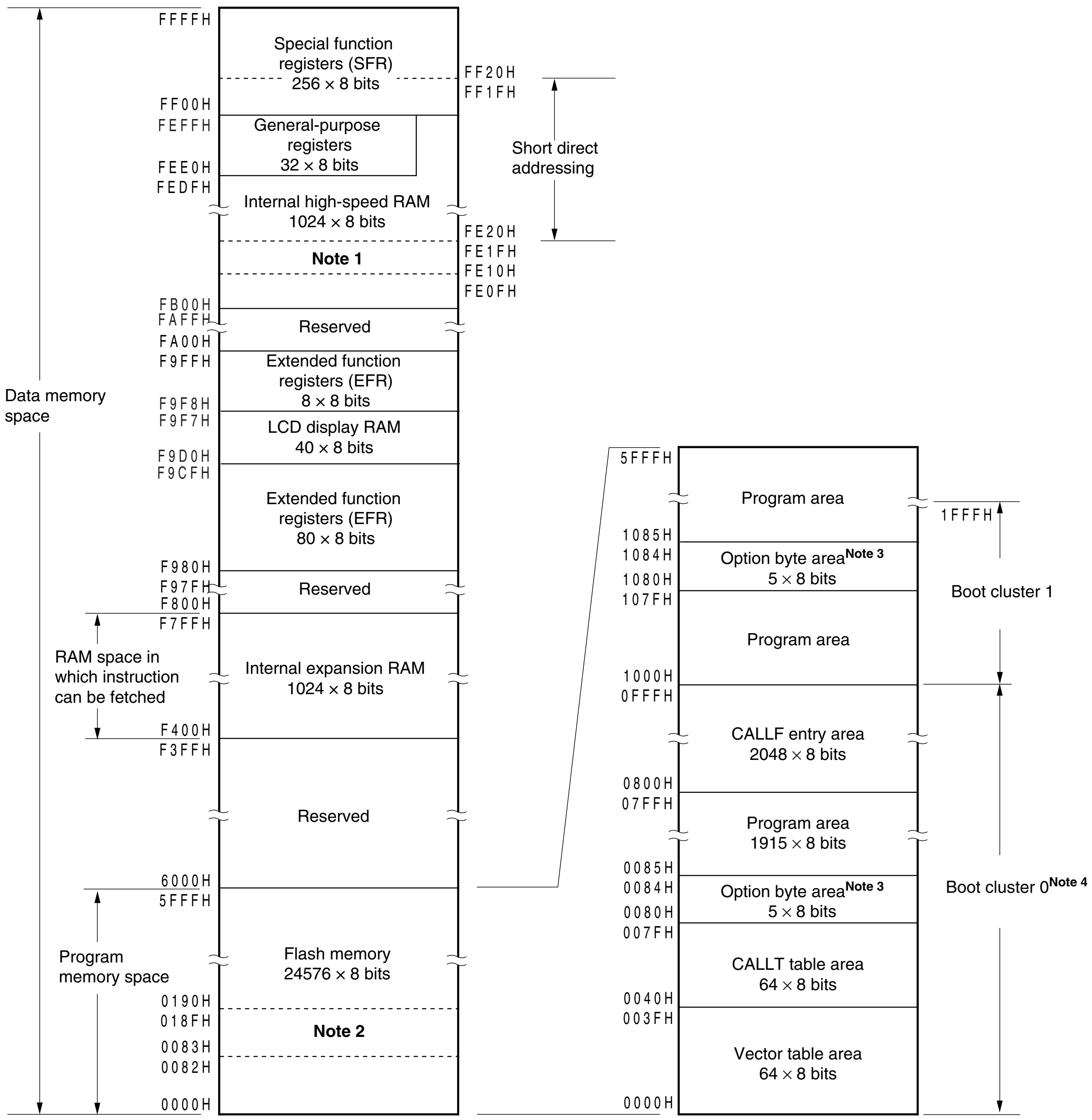
2. To set the memory size, set IMS and then IXS. Set the memory size so that the internal ROM and internal expansion RAM areas do not overlap.

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

Flash Memory Version		IMS	IXS
78K0/DE2	78K0/DF2		
μPD78F0836	μPD78F0838, 78F0840, 78F0842	C6H	0AH
μPD78F0844	μPD78F0846, 78F0848	C8H	
μPD78F0837	μPD78F0839, 78F0841, 78F0843	CCH	08H
μPD78F0845	μPD78F0847, 78F0849	CFH	

Figure 3-1. Memory Map (μPD78F0836, 78F0838, 78F0840, 78F0842)

<R>


FFFFH
Special function registers (SFR) 256 × 8 bits
FF20H
FF1FH
FF00H
FEFFH
General-purpose registers 32 × 8 bits
FEE0H
FEDFH
Short direct addressing
Internal high-speed RAM 1024 × 8 bits
FE20H
FE1FH
FE10H
FE0FH
Note 1
FB00H
FAFFH
Reserved
FA00H
F9FFH
Extended function registers (EFR) 8 × 8 bits
F9F8H
F9F7H
LCD display RAM 40 × 8 bits
F9D0H
F9CFH
Extended function registers (EFR) 80 × 8 bits
F980H
F97FH
Reserved
F800H
F7FFH
Data memory space
RAM space in which instruction can be fetched
Internal expansion RAM 1024 × 8 bits
F400H
F3FFH
Reserved
6000H
5FFFH
Program memory space
Flash memory 24576 × 8 bits
0190H
018FH
Note 2
0083H
0082H
0000H
5FFFH
Program area
1FFFH
1085H
1084H
Option byte areaNote 3 5 × 8 bits
1080H
107FH
Boot cluster 1
Program area
1000H
0FFFH
CALLF entry area 2048 × 8 bits
0800H
07FFH
Program area 1915 × 8 bits
0085H
0084H
Option byte areaNote 3 5 × 8 bits
Boot cluster 0Note 4
0080H
007FH
CALLT table area 64 × 8 bits
0040H
003FH
Vector table area 64 × 8 bits
0000H

Notes 1. During on-chip debugging, use of this area is disabled since it is used as the user data backup area for communication.

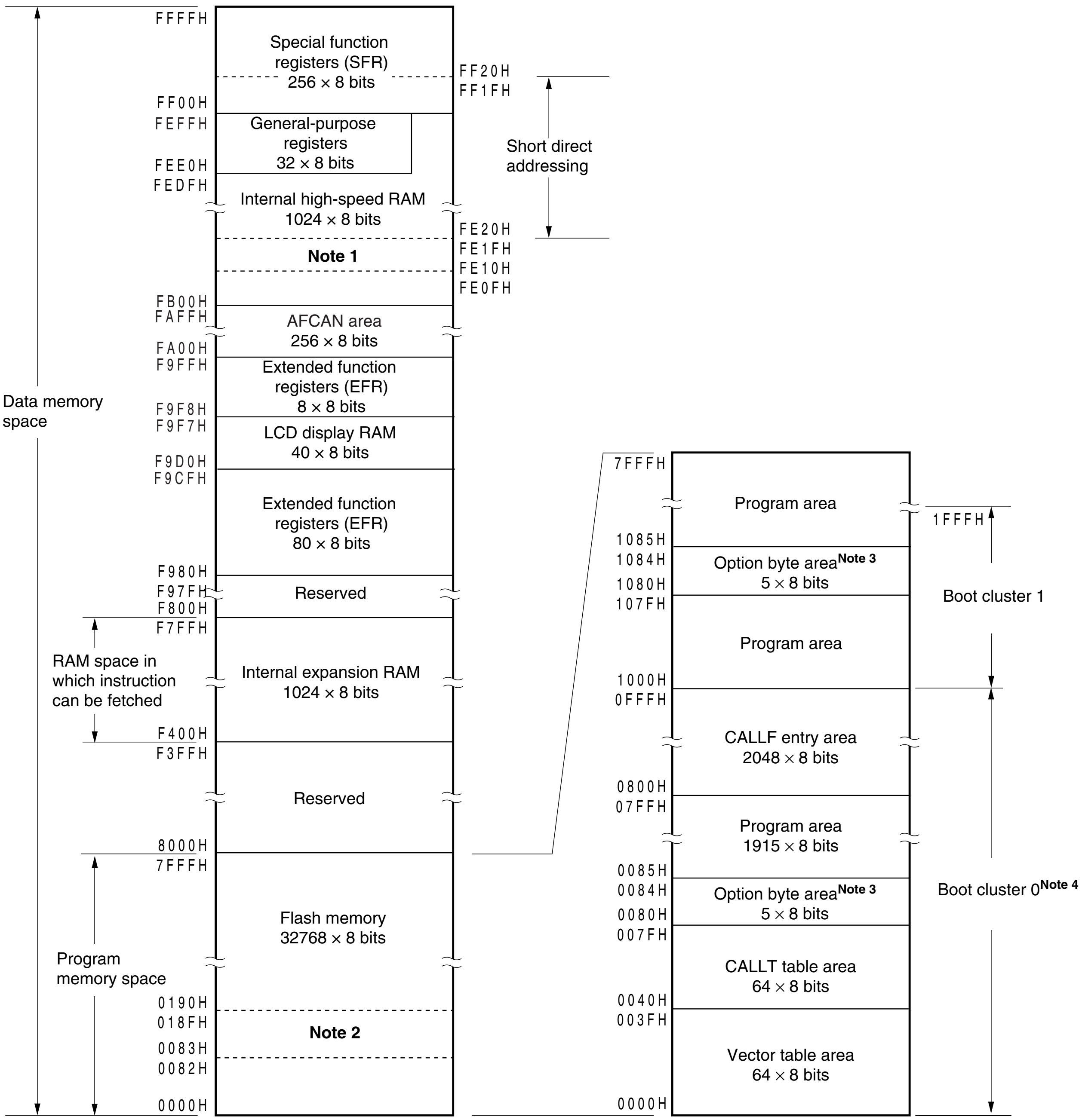
2. During on-chip debugging, use of this area is disabled since it is used as the communication command area (269 bytes).

3. When boot swap is not used: Set the option bytes to 0080H to 0084H.
When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H.

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **26.8 Security Settings**).

Figure 3-2. Memory Map (μPD78F0844, 78F0846, 78F0848)

<R>


FFFFH
Special function registers (SFR) 256 × 8 bits
FF20H
FF1FH
FF00H
FEFFH
General-purpose registers 32 × 8 bits
FEE0H
FEDFH
Short direct addressing
Internal high-speed RAM 1024 × 8 bits
FE20H
FE1FH
Note 1
FE10H
FE0FH
FB00H
FAFFH
AFCAN area 256 × 8 bits
FA00H
F9FFH
Extended function registers (EFR) 8 × 8 bits
Data memory space
F9F8H
F9F7H
LCD display RAM 40 × 8 bits
F9D0H
F9CFH
Extended function registers (EFR) 80 × 8 bits
F980H
F97FH
Reserved
F800H
F7FFH
RAM space in which instruction can be fetched
Internal expansion RAM 1024 × 8 bits
F400H
F3FFH
Reserved
8000H
7FFFH
Flash memory 32768 × 8 bits
Program memory space
0190H
018FH
Note 2
0083H
0082H
0000H
7FFFH
Program area
1FFFH
1085H
1084H
Option byte area Note 3 5 × 8 bits
1080H
107FH
Boot cluster 1
Program area
1000H
0FFFH
CALLF entry area 2048 × 8 bits
0800H
07FFH
Program area 1915 × 8 bits
0085H
0084H
Option byte area Note 3 5 × 8 bits
Boot cluster 0 Note 4
0080H
007FH
CALLT table area 64 × 8 bits
0040H
003FH
Vector table area 64 × 8 bits
0000H

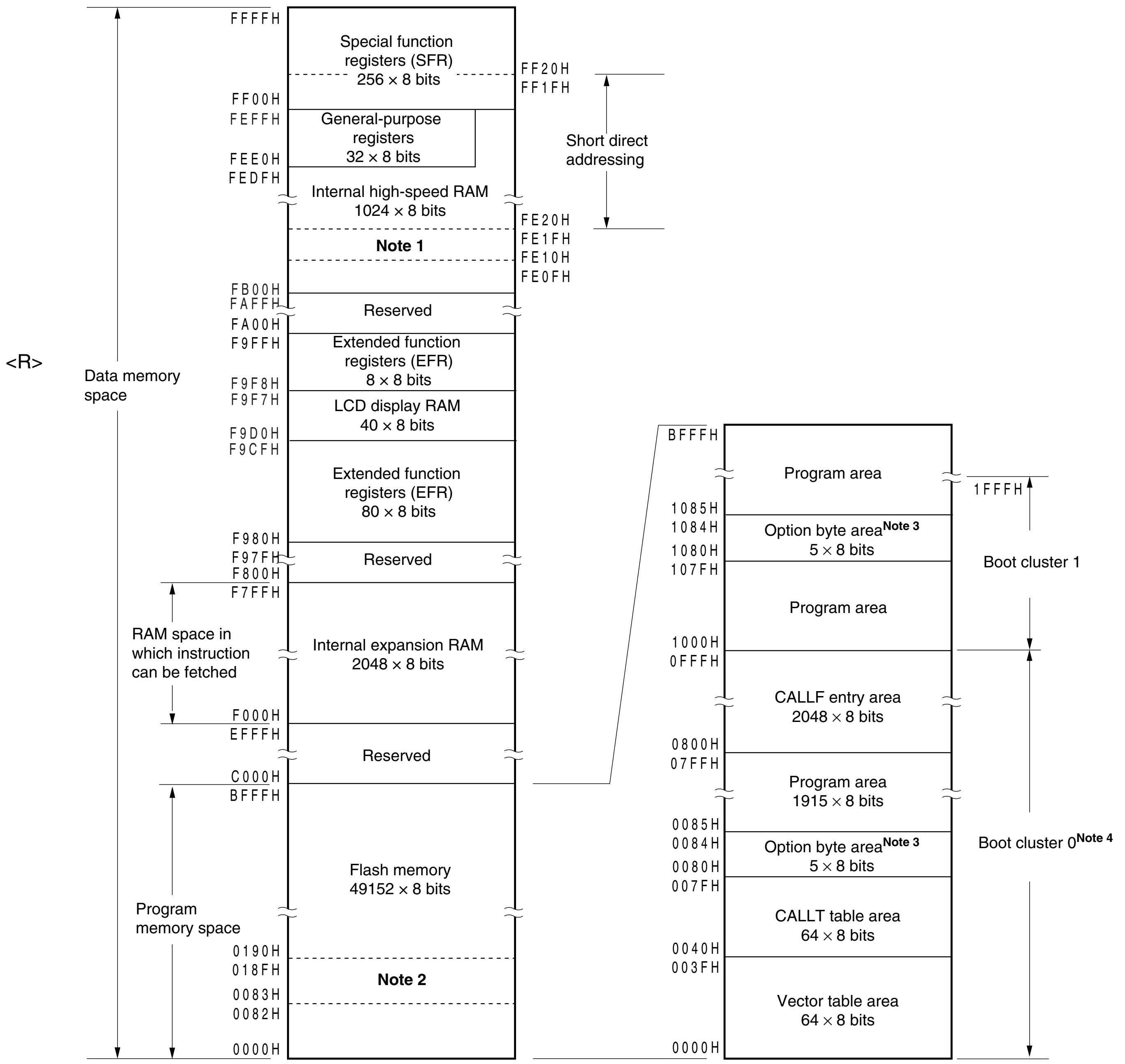
Notes 1. During on-chip debugging, use of this area is disabled since it is used as the user data backup area for communication.

2. During on-chip debugging, use of this area is disabled since it is used as the communication command area (269 bytes).

3. When boot swap is not used: Set the option bytes to 0080H to 0084H.
When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H.

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **26.8 Security Settings**).

Figure 3-3. Memory Map (μPD78F0837, 78F0839, 78F0841, 78F0843)


<R>
Data memory space
RAM space in which instruction can be fetched
Program memory space
FFFFH
Special function registers (SFR) 256 × 8 bits
FF20H
FF1FH
FF00H
FEFFH
General-purpose registers 32 × 8 bits
FEE0H
FEDFH
Internal high-speed RAM 1024 × 8 bits
Short direct addressing
FE20H
FE1FH
Note 1
FE10H
FE0FH
FB00H
FAFFH
Reserved
FA00H
F9FFH
Extended function registers (EFR) 8 × 8 bits
F9F8H
F9F7H
LCD display RAM 40 × 8 bits
F9D0H
F9CFH
Extended function registers (EFR) 80 × 8 bits
F980H
F97FH
Reserved
F800H
F7FFH
Internal expansion RAM 2048 × 8 bits
F000H
EFFFH
Reserved
C000H
BFFFH
Flash memory 49152 × 8 bits
0190H
018FH
Note 2
0083H
0082H
0000H
BFFFH
Program area
1FFFH
1085H
1084H
Option byte areaNote 3 5 × 8 bits
1080H
107FH
Boot cluster 1
Program area
1000H
0FFFH
CALLF entry area 2048 × 8 bits
0800H
07FFH
Program area 1915 × 8 bits
0085H
0084H
Option byte areaNote 3 5 × 8 bits
Boot cluster 0Note 4
0080H
007FH
CALLT table area 64 × 8 bits
0040H
003FH
Vector table area 64 × 8 bits
0000H

Notes 1. During on-chip debugging, use of this area is disabled since it is used as the user data backup area for communication.

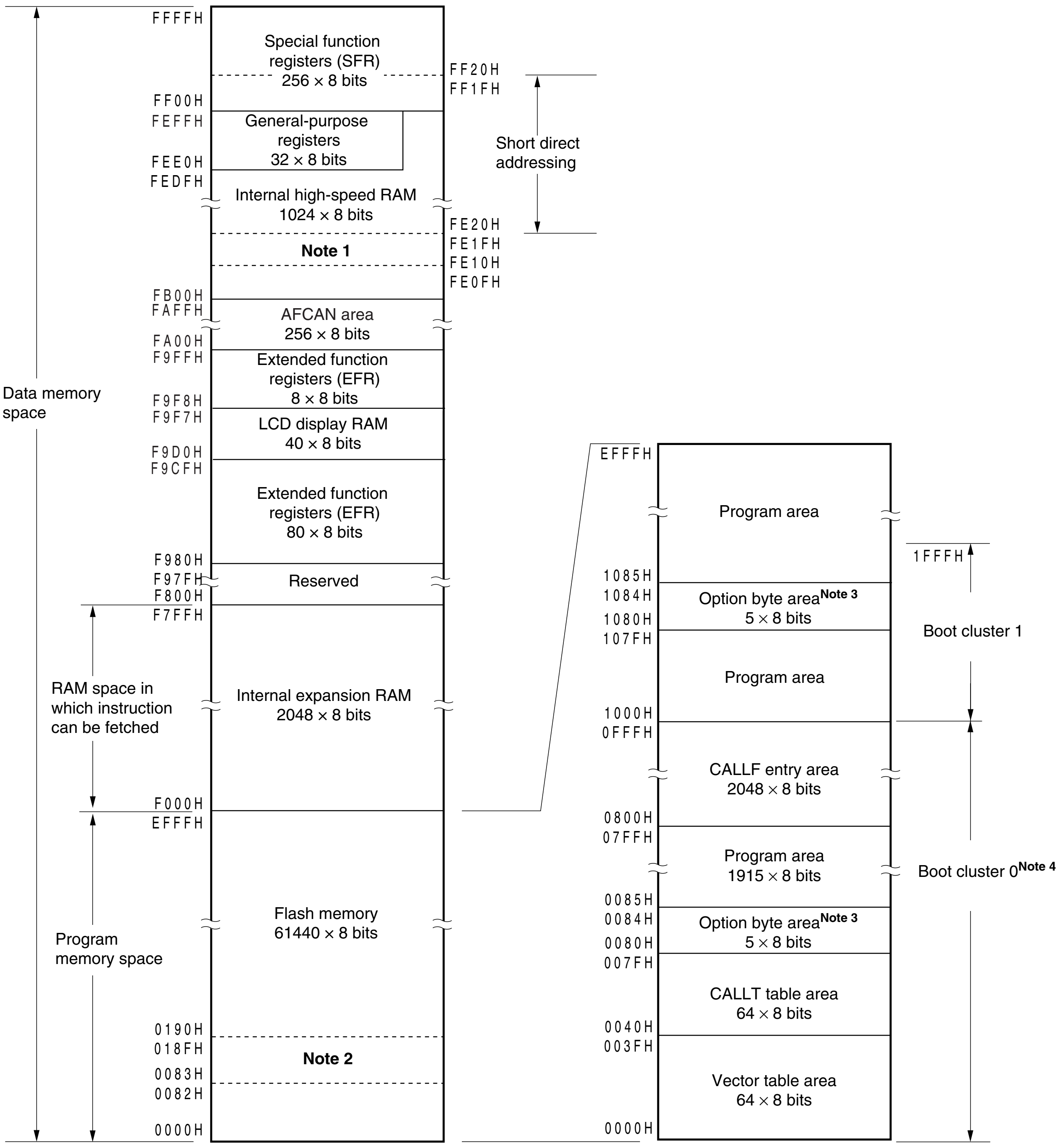
2. During on-chip debugging, use of this area is disabled since it is used as the communication command area (269 bytes).

3. When boot swap is not used: Set the option bytes to 0080H to 0084H.
When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H.

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **26.8 Security Settings**).

Figure 3-4. Memory Map (μPD78F0845, 78F0847, 78F0849)

<R>


FFFFH
Special function registers (SFR) 256 × 8 bits
FF20H
FF1FH
FF00H
FEFFH
General-purpose registers 32 × 8 bits
FEE0H
FEDFH
Short direct addressing
Internal high-speed RAM 1024 × 8 bits
FE20H
FE1FH
Note 1
FE10H
FE0FH
FB00H
FAFFH
AFCAN area 256 × 8 bits
FA00H
F9FFH
Extended function registers (EFR) 8 × 8 bits
Data memory space
F9F8H
F9F7H
LCD display RAM 40 × 8 bits
F9D0H
F9CFH
Extended function registers (EFR) 80 × 8 bits
F980H
F97FH
Reserved
F800H
F7FFH
RAM space in which instruction can be fetched
Internal expansion RAM 2048 × 8 bits
F000H
EFFFH
Program memory space
Flash memory 61440 × 8 bits
0190H
018FH
Note 2
0083H
0082H
0000H
EFFFH
Program area
1FFFH
1085H
1084H
Option byte area Note 3 5 × 8 bits
1080H
107FH
Boot cluster 1
Program area
1000H
0FFFH
CALLF entry area 2048 × 8 bits
0800H
07FFH
Program area 1915 × 8 bits
Boot cluster 0 Note 4
0085H
0084H
Option byte area Note 3 5 × 8 bits
0080H
007FH
CALLT table area 64 × 8 bits
0040H
003FH
Vector table area 64 × 8 bits
0000H

Notes 1. During on-chip debugging, use of this area is disabled since it is used as the user data backup area for communication.

2. During on-chip debugging, use of this area is disabled since it is used as the communication command area (269 bytes).

3. When boot swap is not used: Set the option bytes to 0080H to 0084H.
When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H.

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **26.8 Security Settings**).

Remark The flash memory is divided into blocks (one block = 1 KB). For correspondence between the address values and block numbers in the flash memory, see the following table.

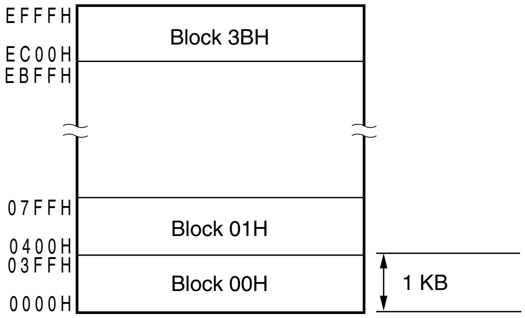


Table 3-2. Correspondence between Address Values and Block Numbers in Flash Memory

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		

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/Dx2 products incorporate internal ROM (flash memory), as shown below.

Table 3-3. Internal ROM Capacity

Part Number		Internal ROM	
78K0/DE2	78K0/DF2	Structure	Capacity
μPD78F0836	μPD78F0838, 78F0840, 78F0842	Flash memory	24576 × 8 bits (0000H to 5FFFH)
μPD78F0837	μPD78F0839, 78F0841, 78F0843		49152 × 8 bits (0000H to BFFFH)
μPD78F0844	μPD78F0846, 78F0848		32768 × 8 bits (0000H to 7FFFH)
μPD78F0845	μPD78F0847, 78F0849		61440 × 8 bits (0000H to EFFFH)

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

(1) Vector code area

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

Vector Code Address	Interrupt Source	Vector Code Address	Interrupt Source
0000H	$\overline{\text{RESET}}$ input, POC, LVI, WDT	0020H	INTP2/INTCSI10
0002H	INTCK2	0022H[Note2]	INTPR61
0004H	INTLVI	0024H[Note2]	INTSR61
0006H	INTP0	0026H[Note2]	INTST61
0008H	INTP1/INTIIC0	0028H[Note2]	INTCSI11
000AH	INTP3/INTP4	002AH	INTTM50/INTTP2OV
000CH	INTTP0CC0	002CH	INTTP2CC0
000EH	INTTP0CC1	002EH	INTTP2CC1
0010H	INTTP1CC0	0030H	INTAD
0012H	INTTP1CC1	0032H	INTWT/INTWTI
0014H[Note1]	INTC0ERR/INTC0WUP	0034H	INTTM51/INTTP3OV
0016H[Note1]	INTC0REC	0036H	INTTP3CC0
0018H[Note1]	INTC0TRX	0038H	INTTP3CC1
001AH	INTPR60	003AH	INTTP4CC0
001CH	INTSR60	003CH	INTTP4CC1
001EH	INTST60	003EH	BRK

Notes 1. μPD78F0844, 78F0845, 78F0846, 78F0847, 78F0848, 78F0849 only.

2. 78K0/DF2 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

The option byte area is assigned to the 1-byte area of 0080H. Refer to **CHAPTER 25 OPTION BYTE** for details.

(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 27 ON-CHIP DEBUG FUNCTION**.

3.1.2 Internal data memory space

78K0/Dx2 products incorporate the following RAM.

(1) Internal high-speed RAM

Table 3-5. Internal High-Speed RAM Capacity

Part Number		Internal High-Speed RAM
78K0/DE2	78K0/DF2	
μPD78F0836	μPD78F0838, 78F0840, 78F0842	1024 × 8 bits (FB00H to FEFFH)
μPD78F0837	μPD78F0839, 78F0841, 78F0843	
μPD78F0844	μPD78F0846, 78F0848	
μPD78F0845	μPD78F0847, 78F0849	

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

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
78K0/DE2	78K0/DF2	
μPD78F0836, 78F0844	μPD78F0838, 78F0840, 78F0842, 78F0846, 78F0848	1024 × 8 bits (F400H to F7FFH)
μPD78F0837, 78F0845	μPD78F0839, 78F0841, 78F0843, 78F0847, 78F0849	2048 × 8 bits (F000H to F7FFH)

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.

3.1.3 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area FF00H to FFFFH (refer to **Table 3-7. Special Function Register List**).

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

<R>

3.1.4 Extended function register (EFR) area

On-chip peripheral hardware extended function registers (EFRs) are allocated in the area F980H to F9CFH and F9F8H to F9FFH (refer to **Table 3-8. Extended Function Register List**).

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

Remark The following methods are available to specify the EFRs to undergo manipulation during instruction execution.

1. Direct addressing
2. Register indirect addressing
3. Based addressing
4. Based indexed addressing

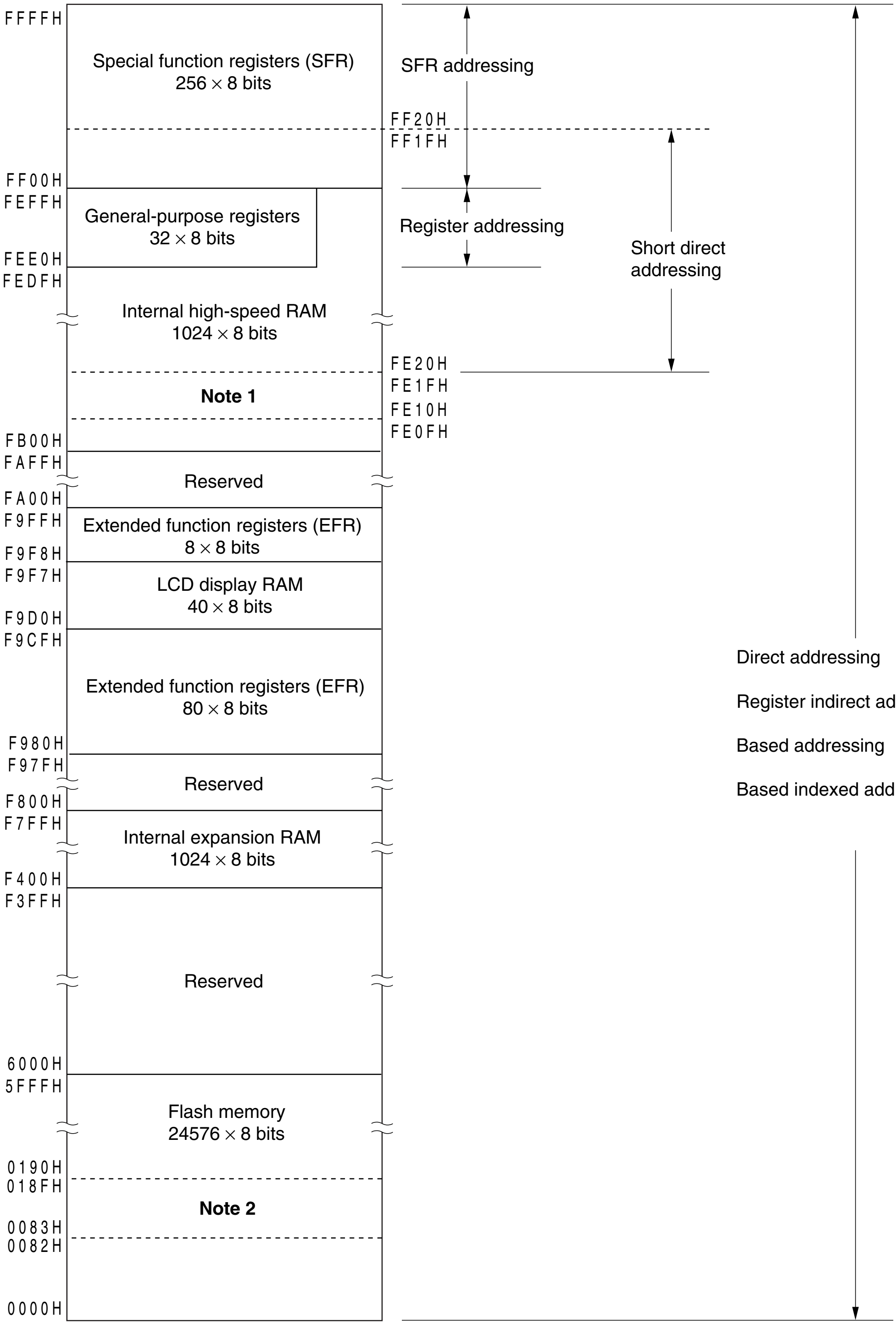
3.1.5 Data memory addressing

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/Dx2, 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. **Figure 3-5** to **3-8** show correspondence between data memory and addressing. For details of each addressing mode, refer to **3.4 Operand Address Addressing**.

Figure 3-5. Correspondence between Data Memory and Addressing (μPD78F0836, 78F0838, 78F0840, 78F0842)

<R>

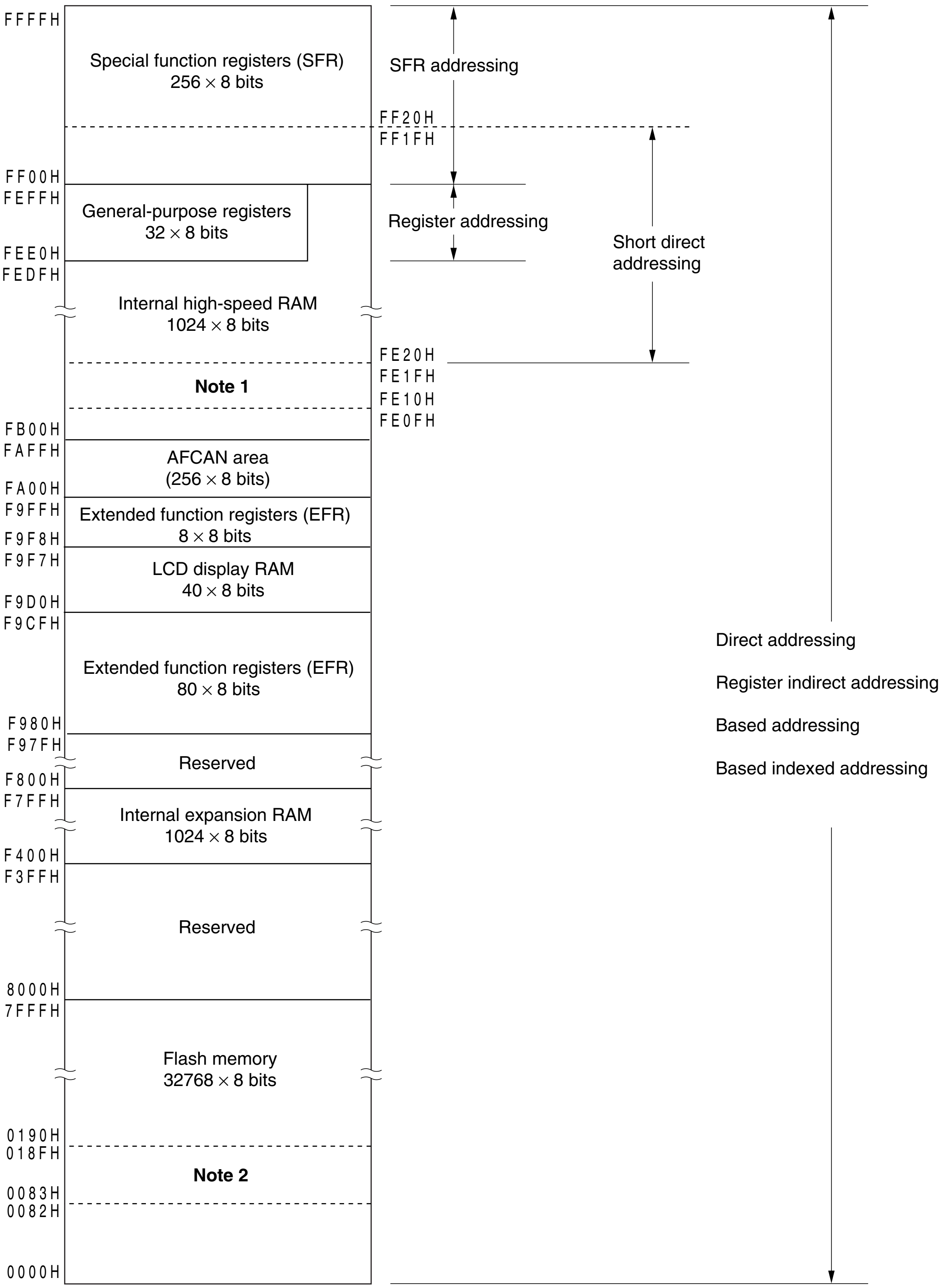


Notes 1. During on-chip debugging, use of this area is disabled since it is used as the user data backup area for communication.

2. During on-chip debugging, use of this area is disabled since it is used as the communication command area (269 bytes).

Figure 3-6. Correspondence between Data Memory and Addressing (μPD78F0844, 78F0846, 78F0848)

<R>

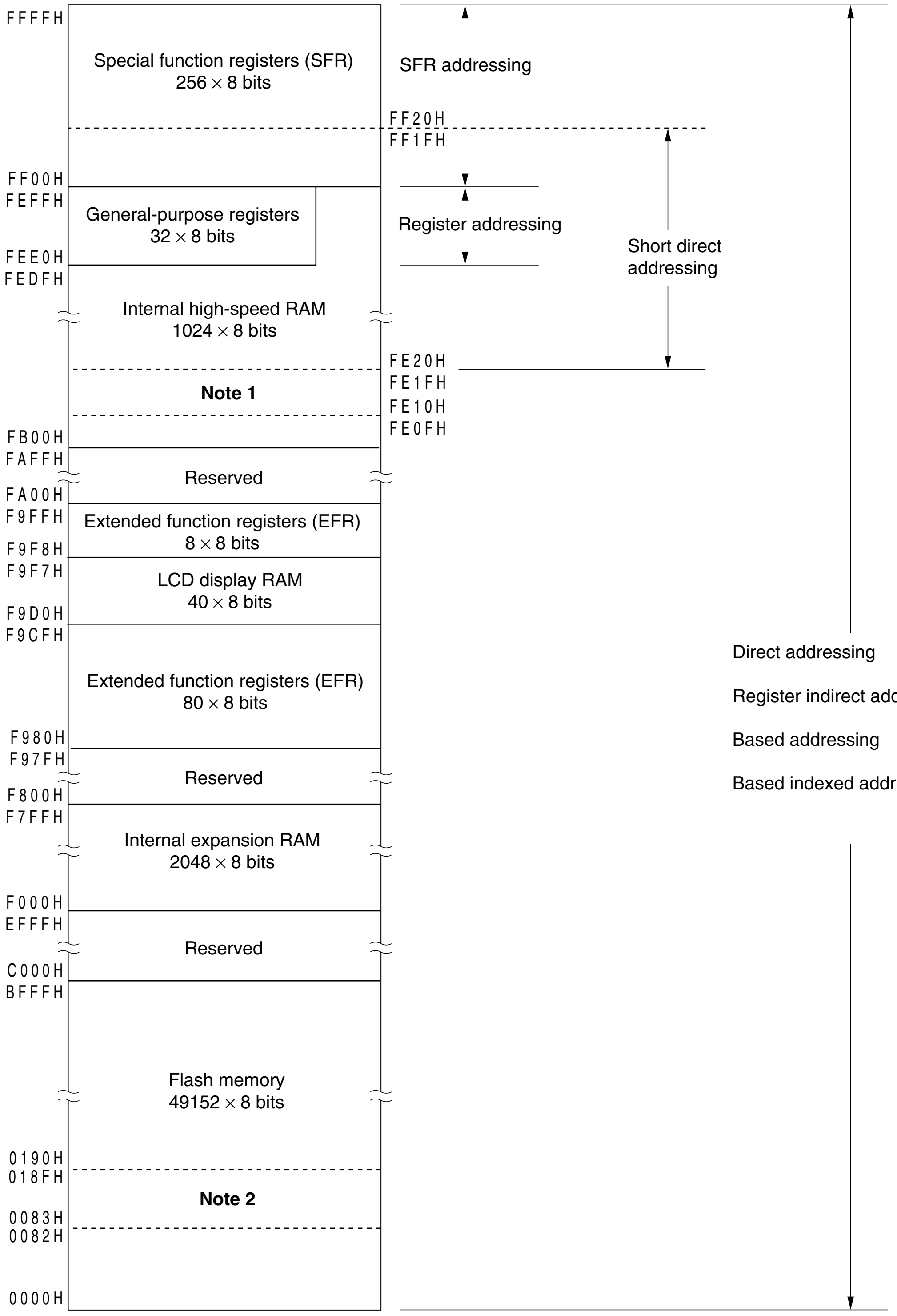


Notes 1. During on-chip debugging, use of this area is disabled since it is used as the user data backup area for communication.

2. During on-chip debugging, use of this area is disabled since it is used as the communication command area (269 bytes).

Figure 3-7. Correspondence between Data Memory and Addressing
(μPD78F0837, 78F0839, 78F0841, 78F0843)

<R>

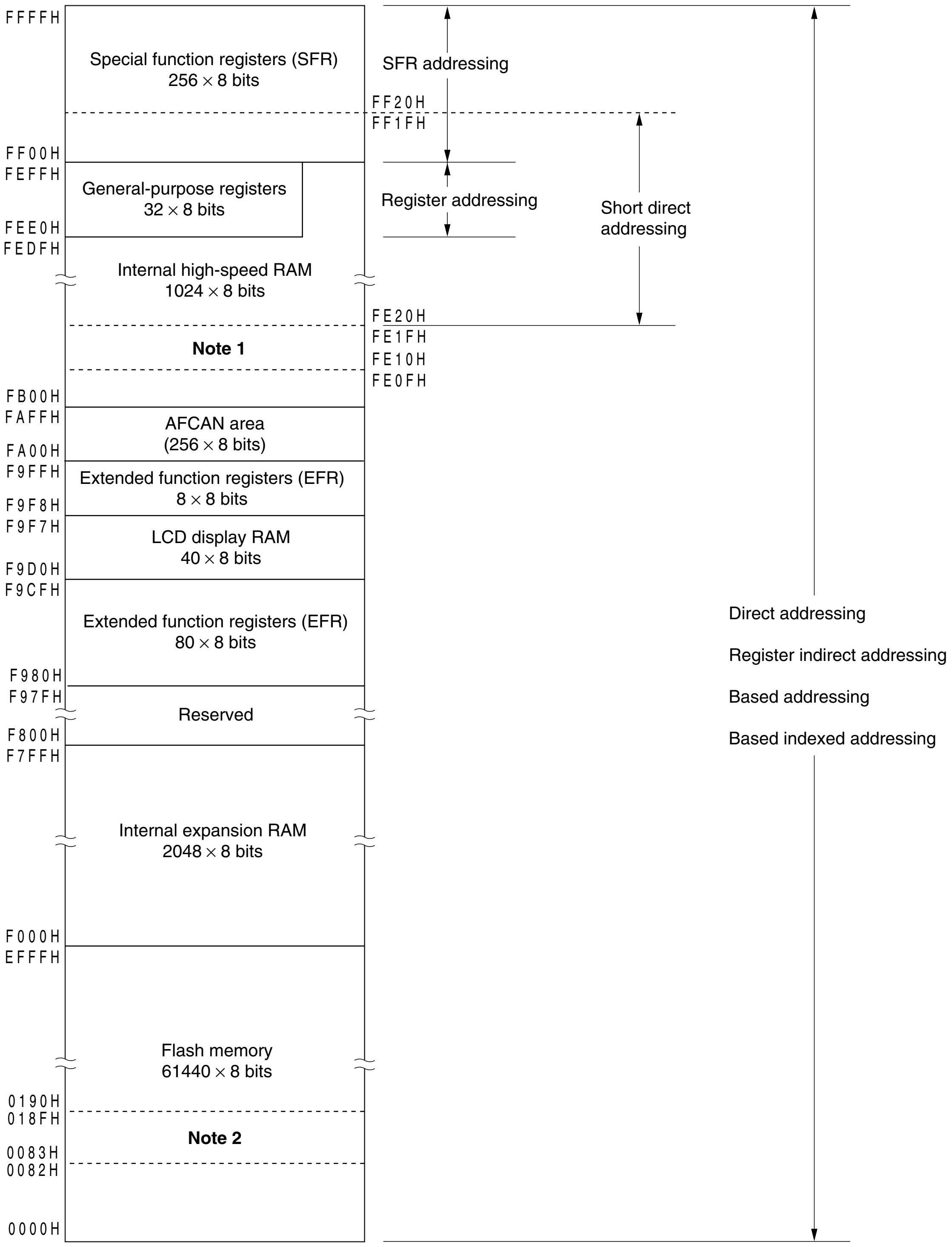


Notes 1. During on-chip debugging, use of this area is disabled since it is used as the user data backup area for communication.

2. During on-chip debugging, use of this area is disabled since it is used as the communication command area (269 bytes).

Figure 3-8. Correspondence between Data Memory and Addressing (μPD78F0845, 78F0847, 78F0849)

<R>



Notes 1. During on-chip debugging, use of this area is disabled since it is used as the user data backup area for communication.

2. During on-chip debugging, use of this area is disabled since it is used as the communication command area (269 bytes).

3.2 Processor Registers

78K0/Dx2 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, the PC is automatically incremented according to the number of bytes of the instruction to be fetched. When a branch instruction is executed, immediate data and register contents are set.

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

Figure 3-9. 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 automatically stacked upon interrupt request generation or PUSH PSW instruction execution and are restored upon execution of the RETB, RETI and POP PSW instructions.

Reset signal sets the PSW to 02H.

Figure 3-10. 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. Other interrupt requests are all disabled.

When 1, the IE flag is set to the interrupt enabled (EI) state and interrupt request acknowledgement 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 acknowledgement 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) (refer to **19.3 (3) Priority specification flag registers (PR0L, PR0H, PR1L, PR1H)**) can not be acknowledged. Actual request acknowledgement 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 3-11. Format of Stack Pointer

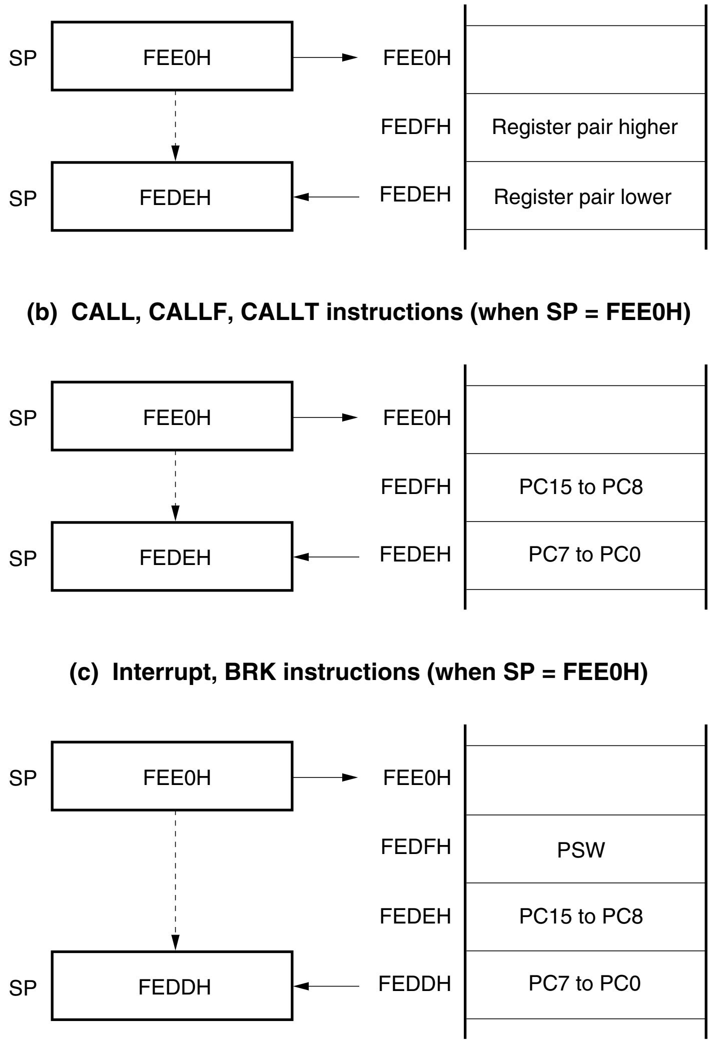
	15															0
SP	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0

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 3-12** and **3-13**.

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

Figure 3-12. Data to Be Saved to Stack Memory

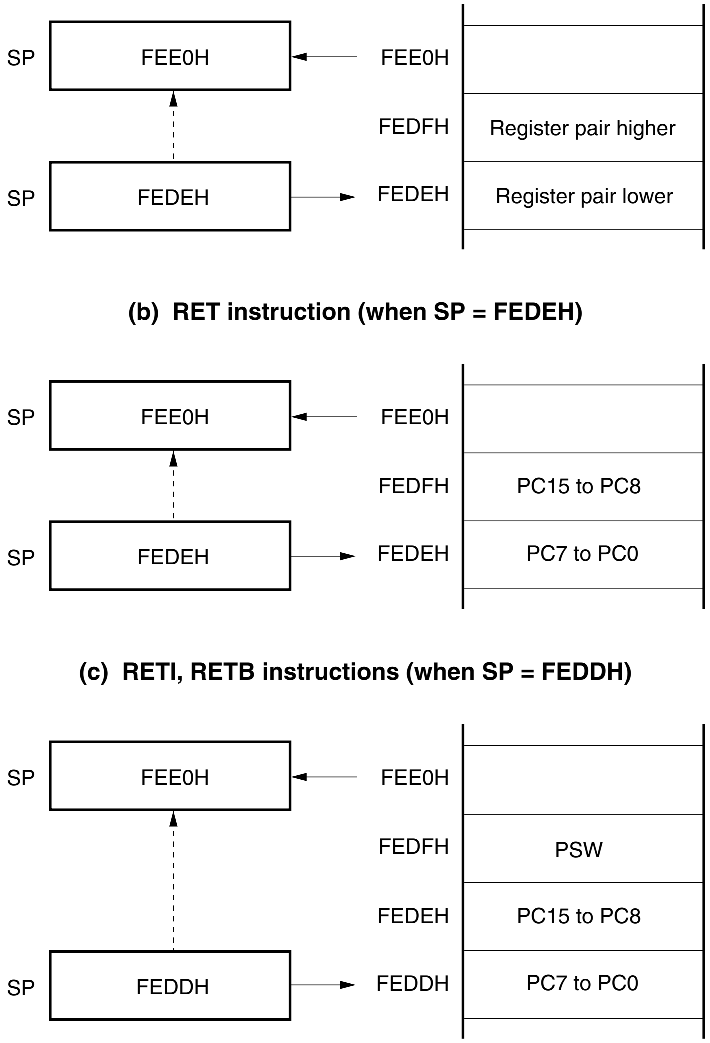


(a) PUSH rp instruction (when SP = FEE0H)

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

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

Figure 3-13. 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).

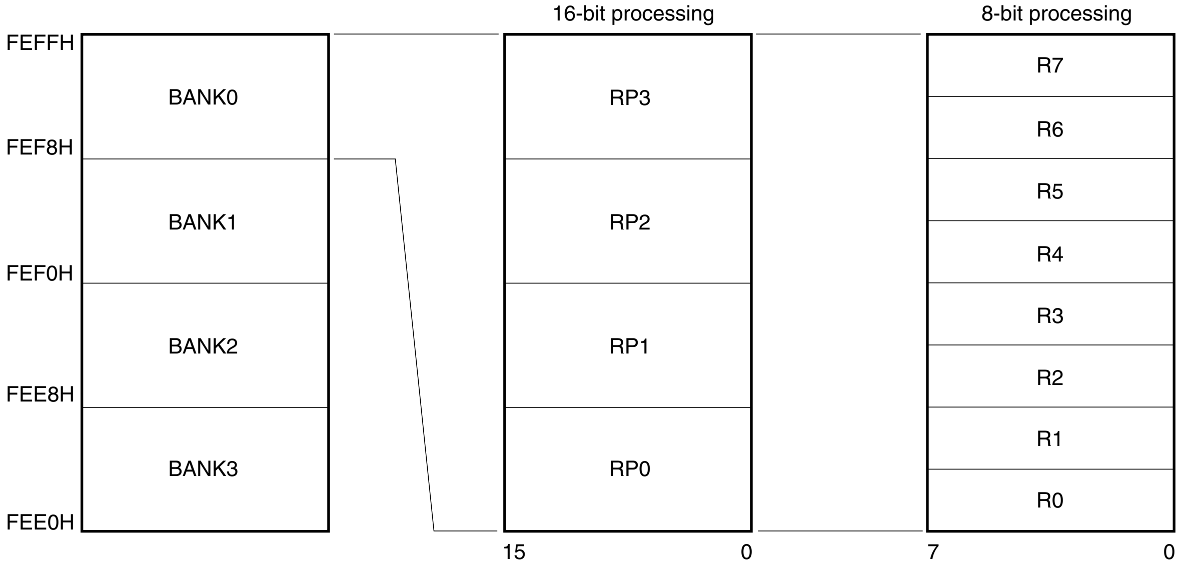
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

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).

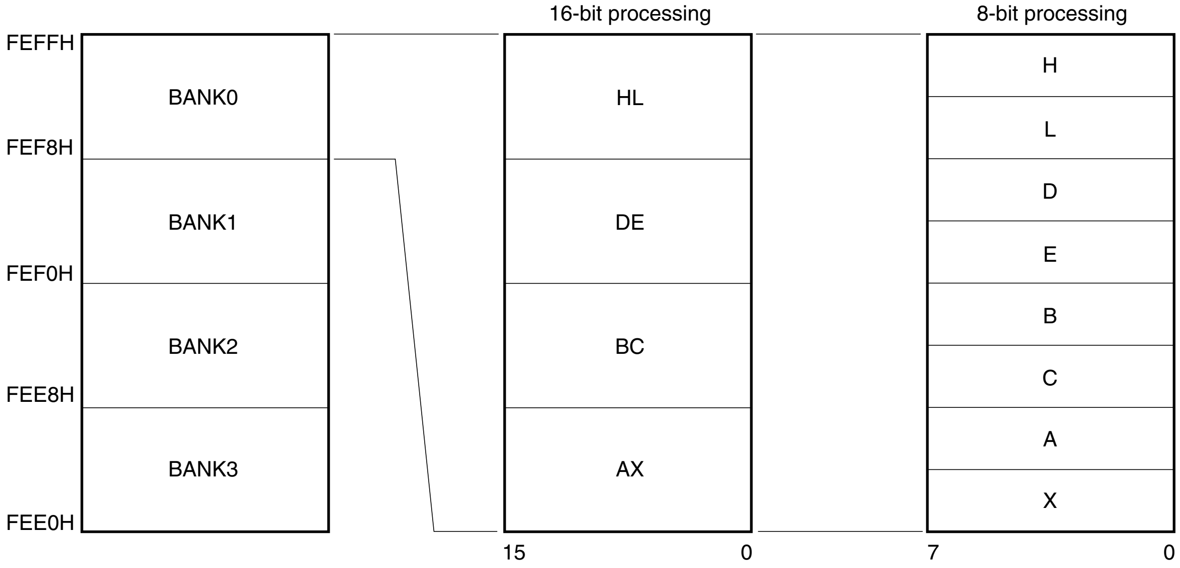
Register banks to be used for instruction execution are set by the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupts for each bank.

Figure 3-14. Configuration of General-Purpose Registers

(a) Absolute name



(b) Function 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 area.

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 3-7. Special Function Register List 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 by the header file “sfrbit.h” in the CC78K0. When using the RA78K0, ID78K0-NS, ID78K0, or SM78K0, symbols can be written as an instruction operand.
- R/W

  Indicates whether the corresponding special function register can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulatable bit units

  Indicates the manipulatable bit unit (1, 8, or 16). “–” indicates a bit unit for which manipulation is not possible.
- After reset

  Indicates each register status upon reset signal generation.
- Target

  Indicates available products for a special function register.
  “–”: for all products.
  CAN: μPD78F0844, 78F0845, 78F0846, 78F0847, 78F0848, and 78F0849 only.
  <R> DF2: 78K0/DF2 (μPD78F0838, 78F0839, 78F0840, 78F0841, 78F0842, 78F0843, 78F0846, 78F0847, 78F0848, and 78F0849) only.

Table 3-7. Special Function Register List (1/6)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset	Target
				1 Bit	8 Bits	16 Bits		
FF00H	Port register 0	P0	R/W	√	√	–	00H	–
FF01H	Port register 1	P1	R/W	√	√	–	00H	–
FF02H	Port register 2	P2	R/W	√	√	–	00H	–
FF03H	Port register 3	P3	R/W	√	√	–	00H	–
FF06H	Port register 6	P6	R/W	√	√	–	00H	–
FF07H	Port register 7	P7	R/W	√	√	–	00H	–
FF08H	Port register 8	P8	R/W	√	√	–	00H	–
FF09H	Port register 9	P9	R/W	√	√	–	00H	DF2
FF0AH	Receive buffer register 60	RXB60	R	–	√	–	FFH	–
FF0BH	Transmit buffer register 60	TXB60	R/W	–	√	–	FFH	–
FF0CH	Port register 12	P12	R/W	√	√	–	00H	–
FF0DH	LCD port function register 3	LCDPF3	R/W	√	√	–	00H	–
FF0EH	Port output mode control register 6	POM6	R/W	√	√	–	00H	–
FF0FH	Serial I/O shift register 10	SIO10	R	–	√	–	00H	–
FF10H	16-bit timer P4 control register 0	TP4CTL0	R/W	√	√	–	00H	–
FF11H	16-bit timer P4 control register 1	TP4CTL1	R/W	√	√	–	00H	–
FF12H	16-bit timer P4 I/O control register 0	TP4IOC0	R/W	√	√	–	00H	–
FF13H	16-bit timer P4 I/O control register 1	TP4IOC1	R/W	√	√	–	00H	–
FF14H	16-bit timer P4 I/O control register 2	TP4IOC2	R/W	√	√	–	00H	–
FF15H	16-bit timer P4 option register 0	TP4OPT0	R/W	√	√	–	00H	–
FF16H	8-bit timer counter 50	TM50	R	–	√	–	00H	–
FF17H	8-bit timer compare register 50	CR50	R/W	–	√	–	00H	–
FF18H	10- bit A/D conversion result register	ADCR	R	–	–	√	0000H	–
FF19H	8-bit A/D conversion result register	ADCRH	R	–	√	–	00H	–
FF1AH	LCD port function register ALL	LCDPFALL	R/W	√	√	–	00H	–
FF1BH	LCD port function register 0	LCDPF0	R/W	√	√	–	00H	–
FF1FH	8-bit timer counter 51	TM51	R	–	√	–	00H	–
FF20H	Port mode register 0	PM0	R/W	√	√	–	FFH	–
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	–
FF26H	Port mode register 6	PM6	R/W	√	√	–	FFH	–
FF27H	Port mode register 7	PM7	R/W	√	√	–	FFH	–
FF28H	Port mode register 8	PM8	R/W	√	√	–	FFH	–
FF29H	Port mode register 9	PM9	R/W	√	√	–	FFH	DF2
FF2AH	A/D converter mode register	ADM	R/W	√	√	–	00H	–
FF2BH	Analog input channel specification register	ADS	R/W	√	√	–	00H	–

Table 3-7. Special Function Register List (2/6)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset	Target
				1 Bit	8 Bits	16 Bits		
FF2CH	Port mode register 12	PM12	R/W	√	√	–	FFH	–
FF2DH	A/D port configuration register	ADPC	R/W	√	√	–	00H	–
FF2EH	Asynchronous serial interface operation mode register 61	ASIM61	R/W	√	√	–	01H	DF2
FF2FH	Asynchronous serial interface reception error status register 61	ASIS61	R	–	√	–	00H	DF2
FF30H	Pull-up resistor option register 0	PU0	R/W	√	√	–	00H	–
FF31H	Pull-up resistor option register 1	PU1	R/W	√	√	–	00H	–
FF33H	Pull-up resistor option register 3	PU3	R/W	√	√	–	00H	–
FF36H	Pull-up resistor option register 6	PU6	R/W	√	√	–	00H	–
FF37H	Pull-up resistor option register 7	PU7	R/W	√	√	–	00H	–
FF38H	Asynchronous serial interface transmission status register 61	ASIF61	R	–	√	–	00H	DF2
FF39H	Clock selection register 61	CKSR61	R/W	–	√	–	00H	DF2
FF3AH	Receive buffer register 61	RXB61	R	–	√	–	FFH	DF2
FF3BH	Transmit buffer register 61	TXB61	R/W	–	√	–	FFH	DF2
FF3CH	Pull-up resistor option register 12	PU12	R/W	√	√	–	00H	–
FF3DH	SM port mode control register	SMPC	R/W	√	√	–	00H	–
FF3EH	Baud rate generator control register 61	BRGC61	R/W	–	√	–	FFH	DF2
FF3FH	Asynchronous serial interface control register 61	ASICL61	R/W	√	√	–	16H	DF2
FF40H	Clock output selection register	CKS	R/W	√	√	–	00H	–
FF41H	8-bit timer compare register 51	CR51	R/W	–	√	–	00H	–
FF42H	CAN module last out-pointer register	C0LOPT	R	–	√	–	Undefined	CAN
FF43H	8-bit timer mode control register 51	TMC51	R/W	√	√	–	00H	–
FF44H	CAN module receive history list register	C0RGPT	R/W	–	–	√	xx02H	CAN
FF45H								
FF47H	Serial I/O shift register 11	SIO11	R	–	√	–	00H	DF2
FF48H	External interrupt rising edge enable register	EGP	R/W	√	√	–	00H	–
FF49H	External interrupt falling edge enable register	EGN	R/W	√	√	–	00H	–
FF4AH	CAN module transmit history list register	C0TGPT	R/W	–	–	√	xx02H	CAN
FF4BH								
FF4CH	CAN global control register	C0GMCTRL	R/W	–	–	√	0000H	CAN
FF4DH								
FF4EH	Transmit buffer register 11	SOTB11	R/W	–	√	–	00H	DF2
FF4FH	Input switch control register	ISC	R/W	√	√	–	00H	–

Table 3-7. Special Function Register List (3/6)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset	Target
					1 Bit	8 Bits	16 Bits		
FF50H	Asynchronous serial interface operation mode register 60	ASIM60		R/W	√	√	–	01H	–
FF53H	Asynchronous serial interface reception error status register 60	ASIS60		R	–	√	–	00H	–
FF54H	Input noise filter control register 0 for 16-bit timer P0, P1	TIPNF0		R/W	√	√	–	00H	–
FF55H	Asynchronous serial interface transmission status register 60	ASIF60		R	–	√	–	00H	–
FF56H	Clock selection register 60	CKSR60		R/W	–	√	–	00H	–
FF57H	Baud rate generator control register 60	BRGC60		R/W	–	√	–	FFH	–
FF58H	Asynchronous serial interface control register 60	ASICL60		R/W	√	√	–	16H	–
FF59H	Input noise filter control register 1 for 16-bit timer P2, P3	TIPNF1		R/W	√	√	–	00H	–
FF5AH	LCD mode register	LCDMD		R/W	√	√	–	00H	–
FF5BH	LCD display mode register	LCDM		R/W	√	√	–	00H	–
FF5CH	LCD clock control register 0	LCDC0		R/W	√	√	–	00H	–
FF60H	DMU remainder data register 0	SDR0	SDR0L	R	–	√	√	00H	–
FF61H			SDR0H	R	–	√		00H	–
FF62H	DMU multiplication/division data register A0L	MDA0L	MDA0LL	R	–	√	√	00H	–
FF63H			MDA0LH	R	–	√		00H	–
FF64H	DMU multiplication/division data register A0H	MDA0H	MDA0HL	R/W	–	√	√	00H	–
FF65H			MDA0HH	R/W	–	√		00H	–
FF66H	DMU multiplication/division data register B0	MDB0	MDB0L	R/W	–	√	√	00H	–
FF67H			MDB0H	R/W	–	√		00H	–
FF68H	DMU multiplier/divider control register 0	DMUC0		R/W	√	√	–	00H	–
FF69H	Input noise filter control register 2 for 16-bit timer P4	TIPNF2		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	16-bit timer P4 capture/compare register 0	TP4CCR0		R/W	–	–	√	0000H	–
FF6DH									
FF6EH	CAN global clock selection register	C0GMCS		R/W	–	√	–	0FH	CAN
FF6FH	CAN global automatic block transmission delay setting register	C0GMABTD		R/W	–	√	–	00H	CAN
FF70H	CAN module mask 1 register L	C0MASK1L		R/W	–	–	√	Undefined	CAN
FF71H									
FF72H	CAN module mask 1 register H	C0MASK1H		R/W	–	–	√	Undefined	CAN
FF73H									

Table 3-7. Special Function Register List (4/6)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset	Target
				1 Bit	8 Bits	16 Bits		
FF74H	CAN module mask 2 register L	C0MASK2L	R/W	–	–	√	Undefined	CAN
FF75H								
FF76H	CAN module mask 2 register H	C0MASK2H	R/W	–	–	√	Undefined	CAN
FF77H								
FF78H	CAN module mask 3 register L	C0MASK3L	R/W	–	–	√	Undefined	CAN
FF79H								
FF7AH	CAN module mask 3 register H	C0MASK3H	R/W	–	–	√	Undefined	CAN
FF7BH								
FF7CH	CAN module mask 4 register L	C0MASK4L	R/W	–	–	√	Undefined	CAN
FF7DH								
FF7EH	CAN module mask 4 register H	C0MASK4H	R/W	–	–	√	Undefined	CAN
FF7FH								
FF80H	Serial operation mode register 10	CSIM10	R/W	√	√	–	00H	–
FF81H	Serial clock selection register 10	CSIC10	R/W	√	√	–	00H	–
FF84H	Transmit buffer register 10	SOTB10	R/W	–	√	–	00H	–
FF88H	Serial operation mode register 11	CSIM11	R/W	√	√	–	00H	DF2
FF89H	Serial clock selection register 11	CSIC11	R/W	√	√	–	00H	DF2
FF8AH	CAN module time stamp register	C0TS	R/W	–	–	√	0000H	CAN
FF8BH								
FF8CH	Timer clock selection register 51	TCL51	R/W	√	√	–	00H	–
FF8FH	Watch timer operation mode register	WTM	R/W	√	√	–	00H	–
FF90H	CAN module control register	C0CTRL	R/W	–	–	√	0000H	CAN
FF91H								
FF92H	CAN module last error code register	C0LEC	R/W	–	√	–	00H	CAN
FF93H	CAN module information register	C0INFO	R	–	√	–	00H	CAN
FF94H	CAN module error counter register	C0ERC	R	–	–	√	0000H	CAN
FF95H								
FF96H	CAN module interrupt enable register	C0IE	R/W	–	–	√	0000H	CAN
FF97H								
FF98H	CAN module interrupt status register	C0INTS	R/W	–	–	√	0000H	CAN
FF99H								
FF9BH	Watchdog timer enable register	WDTE	R/W	–	√	–	1AH/9AH[Note]	–
FF9CH	CAN module bit rate register	C0BTR	R/W	–	–	√	370FH	CAN
FF9DH								
FF9EH	CAN module bit rate prescaler register	C0BRP	R/W	–	√	–	FFH	CAN
FF9FH	CAN module last in-pointer register	C0LIPT	R	–	√	–	Undefined	CAN

Note The reset value of WDTE is determined by setting of option byte.

Table 3-7. Special Function Register List (5/6)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset	Target
				1 Bit	8 Bits	16 Bits		
FFA0H	Internal oscillator mode register	RCM	R/W	√	√	–	00H[Note1]	–
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	SG0 control register	SG0CTL	R/W	√	√	–	00H	–
FFA6H	SG0 amplitude register	SG0PWM	R/W	–	–	√	0000H	–
FFA7H								
FFA8H	SG0 frequency low register	SG0FL	R/W	–	–	√	0000H	–
FFA9H								
FFAAH	SG0 frequency high register	SG0FH	R/W	–	–	√	0000H	–
FFABH								
FFACH	Reset control flag register	RESF	R	–	√	–	00H[Note 2]	–
FFAEH	CAN global automatic block transmission control register	C0GMABT	R/W	–	–	√	0000H	CAN
FFAFH								
FFB0H	I^{2}C shift register 0	IIC0	R/W	–	√	–	00H	–
FFB1H	I^{2}C control register 0	IICC0	R/W	√	√	–	00H	–
FFB2H	Slave address register 0	SVA0	R/W	–	√	–	00H	–
FFB3H	I^{2}C clock selection register 0	IICCL0	R/W	√	√	–	00H	–
FFB4H	I^{2}C function expansion register 0	IICX0	R/W	√	√	–	00H	–
FFB5H	I^{2}C status register 0	IICS0	R	–	√	–	00H	–
FFB7H	I^{2}C flag register 0	IICF0	R/W	√	√	–	00H	–
FFBAH	16-bit timer P4 capture/compare register 1	TP4CCR1	R/W	–	–	√	0000H	–
FFBBH								
FFBCH	16-bit timer P4 counter read buffer register	TP4CNT	R/W	–	–	√	0000H	–
FFBDH								
FFBEH	Low-voltage detection register	LVIM	R/W	√	√	–	00H	–
FFBFH	Low-voltage detection level selection register	LVIS	R/W	√	√	–	00H	–

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. This value varies depending on the source.

Table 3-7. Special Function Register List (6/6)

<R>

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset	Target
					1 Bit	8 Bits	16 Bits		
FFE0H	Interrupt request flag register 0L	IF0	IF0L	R/W	√	√	√	00H	–
FFE1H	Interrupt request flag register 0H		IF0H	R/W	√	√		00H	–
FFE2H	Interrupt request flag register 1L	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	√	√		DFH	–
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	–
FFEFH	Clock operation mode select register	OSCCTL		R/W	√	√	–	00H	–
FFF0H	Internal memory size switching register[Note]	IMS		R/W	–	√	–	CFH	–
FFF4H	Internal expansion RAM size switching register[Note]	IXS		R/W	–	√	–	0CH	–
FFFBH	Processor clock control register	PCC		R/W	√	√	–	01H	–

Note 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 the 78K0/Dx2 is fixed (IMS = CFH, IXS = 0CH). Therefore, set the value corresponding to each product as indicated below.

Flash Memory Version		IMS	IXS
78K0/DE2	78K0/DF2		
μPD78F0836	μPD78F0838, 78F0840, 78F0842	C6H	0AH
μPD78F0844	μPD78F0846, 78F0848	C8H	
μPD78F0837	μPD78F0839, 78F0841, 78F0843	CCH	08H
μPD78F0845	μPD78F0847, 78F0849	CFH	

<R>

Table 3-8. Extended Function Register List (1/2)

Address	Extended Function Register (EFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset	Target
				1 Bit	8 Bits	16 Bits		
F980H	16-bit timer P0 control register 0	TP0CTL0	R/W	√	√	–	00H	–
F981H	16-bit timer P0 control register 1	TP0CTL1	R/W	√	√	–	00H	–
F982H	16-bit timer P0 I/O control register 0	TP0IOC0	R/W	√	√	–	00H	–
F983H	16-bit timer P0 I/O control register 1	TP0IOC1	R/W	√	√	–	00H	–
F984H	16-bit timer P0 I/O control register 2	TP0IOC2	R/W	√	√	–	00H	–
F985H	16-bit timer P0 option register 0	TP0OPT0	R/W	√	√	–	00H	–
F986H	16-bit timer P0 capture/compare register 0	TP0CCR0	R/W	–	–	√	0000H	–
F987H								
F988H	16-bit timer P0 capture/compare register 1	TP0CCR1	R/W	–	–	√	0000H	–
F989H								
F98AH	16-bit timer P0 counter read buffer register	TP0CNT	R	–	–	√	0000H	–
F98BH								
F990H	16-bit timer P1 control register 0	TP1CTL0	R/W	√	√	–	00H	–
F991H	16-bit timer P1 control register 1	TP1CTL1	R/W	√	√	–	00H	–
F992H	16-bit timer P1 I/O control register 0	TP1IOC0	R/W	√	√	–	00H	–
F993H	16-bit timer P1 I/O control register 1	TP1IOC1	R/W	√	√	–	00H	–
F994H	16-bit timer P1 I/O control register 2	TP1IOC2	R	√	√	–	00H	–
F995H	16-bit timer P1 option register 0	TP1OPT0	R/W	√	√	–	00H	–
F996H	16-bit timer P1 capture/compare register 0	TP1CCR0	R/W	–	–	√	0000H	–
F997H								
F998H	16-bit timer P1 capture/compare register 1	TP1CCR1	R/W	–	–	√	0000H	–
F999H								
F99AH	16-bit timer P1 counter read buffer register	TP1CNT	R	–	–	√	0000H	–
F99BH								
F9A0H	16-bit timer P2 control register 0	TP2CTL0	R/W	√	√	–	00H	–
F9A1H	16-bit timer P2 control register 1	TP2CTL1	R/W	√	√	–	00H	–
F9A2H	16-bit timer P2 I/O control register 0	TP2IOC0	R/W	√	√	–	00H	–
F9A3H	16-bit timer P2 I/O control register 1	TP2IOC1	R/W	√	√	–	00H	–
F9A4H	16-bit timer P2 I/O control register 2	TP2IOC2	R	√	√	–	00H	–
F9A5H	16-bit timer P2 option register 0	TP2OPT0	R/W	√	√	–	00H	–
F9A6H	16-bit timer P2 capture/compare register 0	TP2CCR0	R/W	–	–	√	0000H	–
F9A7H								
F9A8H	16-bit timer P2 capture/compare register 1	TP2CCR1	R/W	–	–	√	0000H	–
F9A9H								
F9AAH	16-bit timer P2 counter read buffer register	TP2CNT	R	–	–	√	0000H	–
F9ABH								

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

Address	Extended Function Register (EFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset	Target
					1 Bit	8 Bits	16 Bits		
F9B0H	16-bit timer P3 control register 0	TP3CTL0		R/W	√	√	–	00H	–
F9B1H	16-bit timer P3 control register 1	TP3CTL1		R/W	√	√	–	00H	–
F9B2H	16-bit timer P3 I/O control register 0	TP3IOC0		R/W	√	√	–	00H	–
F9B3H	16-bit timer P3 I/O control register 1	TP3IOC1		R/W	√	√	–	00H	–
F9B4H	16-bit timer P3 I/O control register 2	TP3IOC2		R	√	√	–	00H	–
F9B5H	16-bit timer P3 option register 0	TP3OPT0		R/W	√	√	–	00H	–
F9B6H	16-bit timer P3 capture/compare register 0	TP3CCR0		R/W	–	–	√	0000H	–
F9B7H									
F9B8H	16-bit timer P3 capture/compare register 1	TP3CCR1		R/W	–	–	√	0000H	–
F9B9H									
F9BAH	16-bit timer P3 counter read buffer register	TP3CNT		R	–	–	√	0000H	–
F9BBH									
F9C0H	MTRC timer mode control register 0	MCNTC0		R/W	√	√	–	00H	–
<R> F9C2H	MTRC compare register 10	MCMP1HW	MCMP10	R/W	–	√	√	00H	**Note1**
F9C3H	MTRC compare register 11		MCMP11	R/W	–	√		00H	**Note1**
F9C4H	MTRC compare register 20	MCMP2HW	MCMP20	R/W	–	√	√	00H	**Note1**
F9C5H	MTRC compare register 21		MCMP21	R/W	–	√		00H	**Note1**
F9C6H	MTRC compare register 30	MCMP3HW	MCMP30	R/W	–	√	√	00H	**Note2**
F9C7H	MTRC compare register 31		MCMP31	R/W	–	√		00H	**Note2**
F9C8H	MTRC compare register 40	MCMP4HW	MCMP40	R/W	–	√	√	00H	**Note2**
F9C9H	MTRC compare register 41		MCMP41	R/W	–	√		00H	**Note2**
F9CAH	MTRC compare control register 1	MCMPC1		R/W	√	√	–	00H	**Note1**
F9CCH	MTRC compare control register 2	MCMPC2		R/W	√	√	–	00H	**Note1**
F9CEH	MTRC compare control register 3	MCMPC3		R/W	√	√	–	00H	**Note2**
F9F8H	MTRC compare control register 4	MCMPC4		R/W	√	√	–	00H	**Note2**
F9FCH	MTRC ZPD detection voltage setting register 0	ZPDS0		R/W	√	√	–	00H	**Note1**
F9FDH	MTRC ZPD detection voltage setting register 1	ZPDS1		R/W	√	√	–	00H	**Note2**
F9FEH	MTRC ZPD flag detection clock setting register	CMPCTL		R/W	√	√	–	00H	–
F9FFH	MTRC ZPD operational control register	ZPDEN		R/W	√	√	–	00H	–

Note 1. μPD78F0836, 78F0837, 78F0842, 78F0843, 78F0844, 78F0845, 78F0848, and 78F0849 only.

2. μPD78F0840, 78F0841, 78F0842, 78F0843, 78F0846, 78F0847, 78F0848, and 78F0849 only.

3.3 Instruction Address Addressing

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

3.3.1 Relative addressing

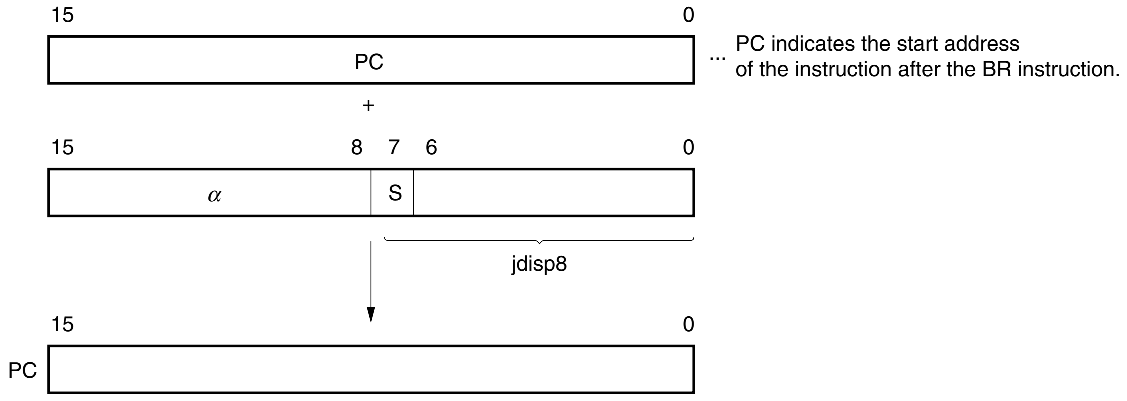
[Function]

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

In other words, relative addressing consists 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]


15
0
PC
... PC indicates the start address of the instruction after the BR instruction.
+
15
8
7
6
0
α
S
jdisp8
15
0
PC

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

3.3.2 Immediate addressing

[Function]

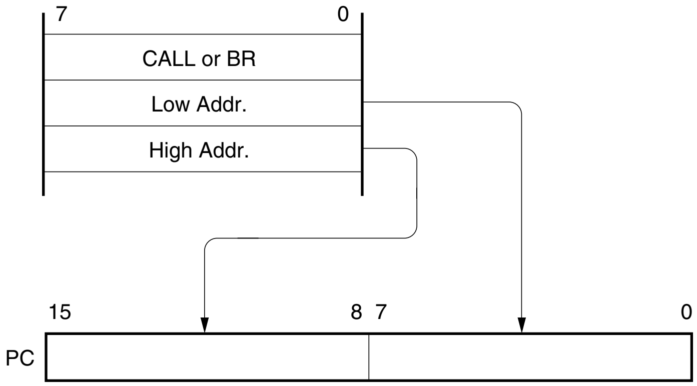
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

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

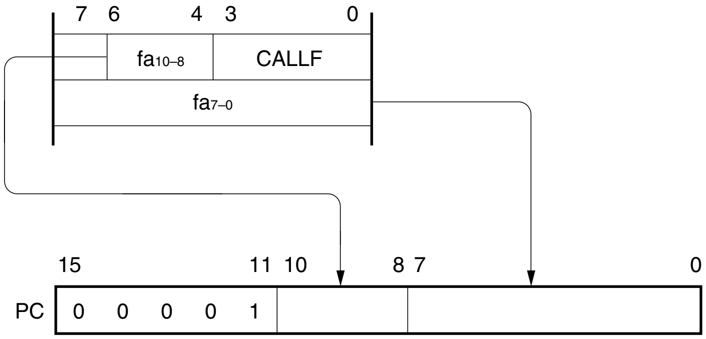
CALL !addr16 and BR !addr16 instructions can be branched to the entire memory space. The CALLF !addr11 instruction is branched to the 0800H to 0FFFH area.

[Illustration]

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



In the case of CALLF !addr11 instruction



3.3.3 Table indirect addressing

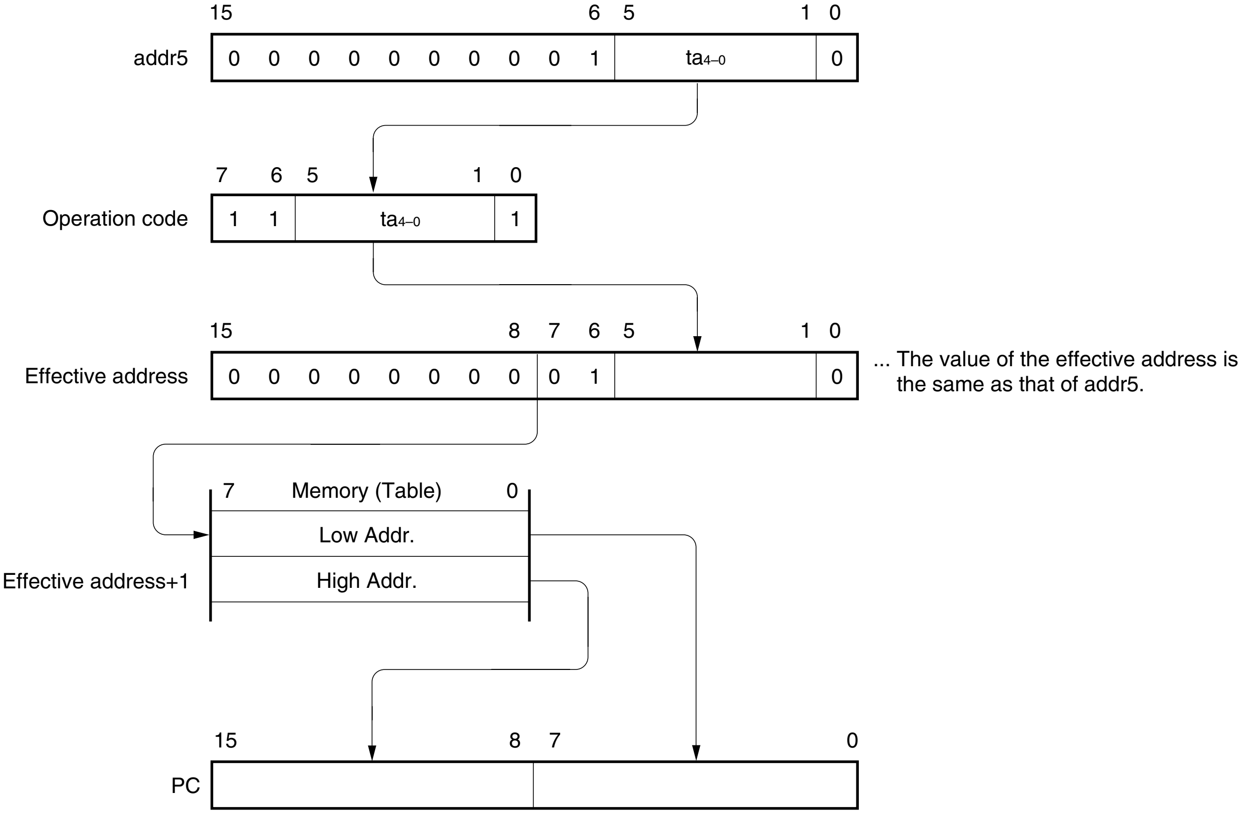
[Function]

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 that is indicated by addr5 and is stored in the memory table from 0040H to 007FH, and allows branching to the entire memory space.

[Illustration]



... The value of the effective address is the same as that of addr5.

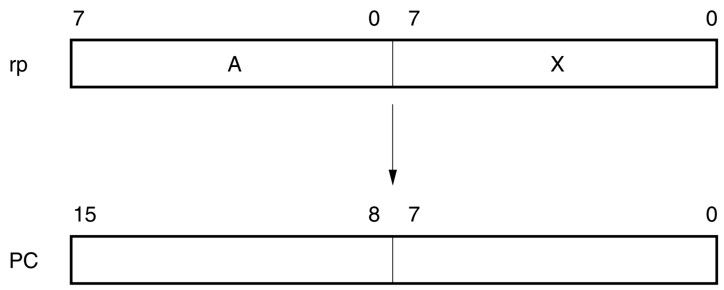
3.3.4 Register addressing

[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/Dx2 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 employed with an instruction, no particular operand format is necessary.

[Description example]

In the case of MULU X

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

3.4.2 Register addressing

[Function]

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

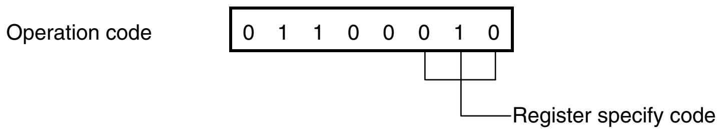
[Operand format]

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

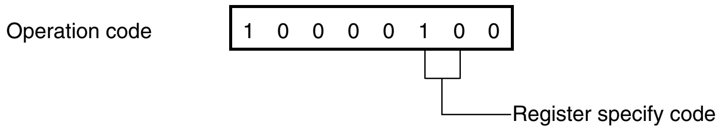
‘r’ and ‘rp’ can be described by absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[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]

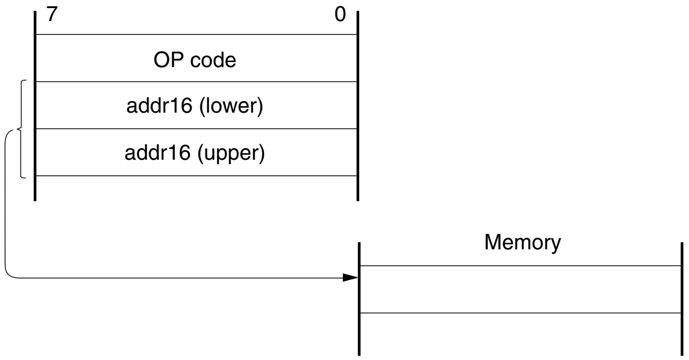
Identifier	Description
addr16	Label or 16-bit immediate data

[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 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. Refer to the **[Illustration]** shown below.

[Operand format]

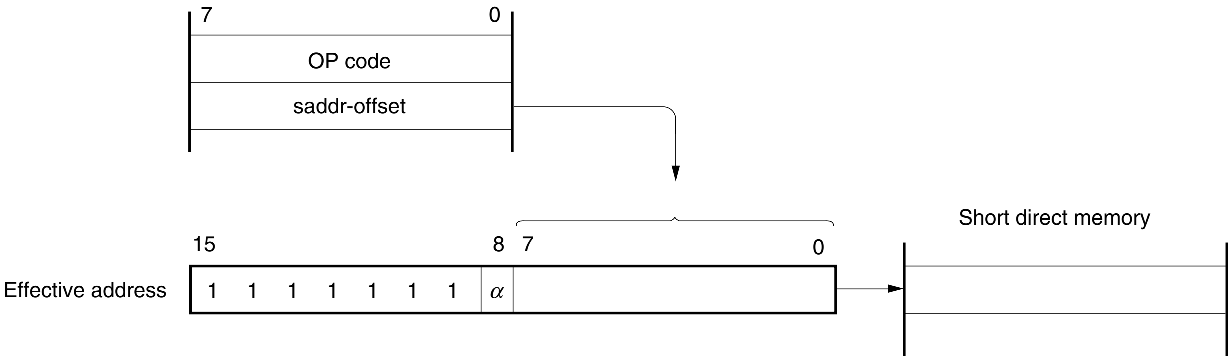
Identifier	Description
saddr	Immediate data that indicate label or FE20H to FF1FH
saddrp	Immediate data that indicate label or FE20H to FF1FH (even address only)

[Description example]

MOV 0FE30H, A; when transferring value of A register to 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]

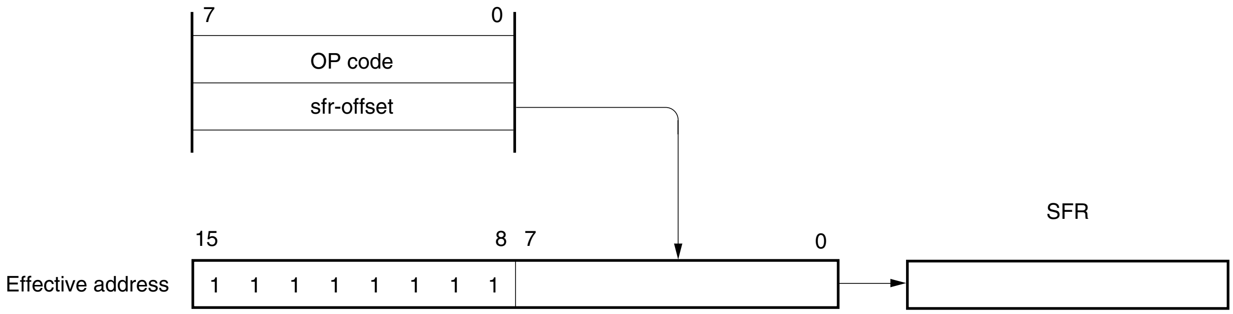
Identifier	Description
sfr	Special function register name
sfrp	16-bit manipulatable special function register name (even address only)

[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 the memory spaces.

[Operand format]

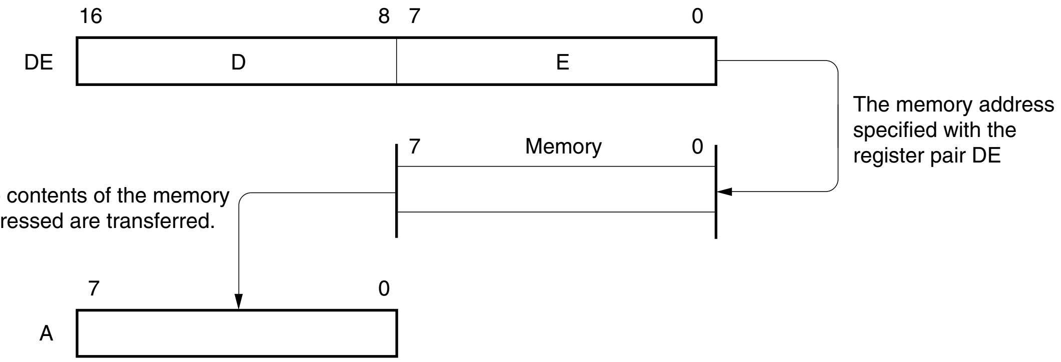
Identifier	Description
–	[DE], [HL]

[Description example]

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

Operation code								
	1	0	0	0	0	1	0	1

[Illustration]


16
8
7
0
DE
D
E
The memory address specified with the register pair DE
7
Memory
0
The contents of the memory addressed are transferred.
7
0
A

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 the memory spaces.

[Operand format]

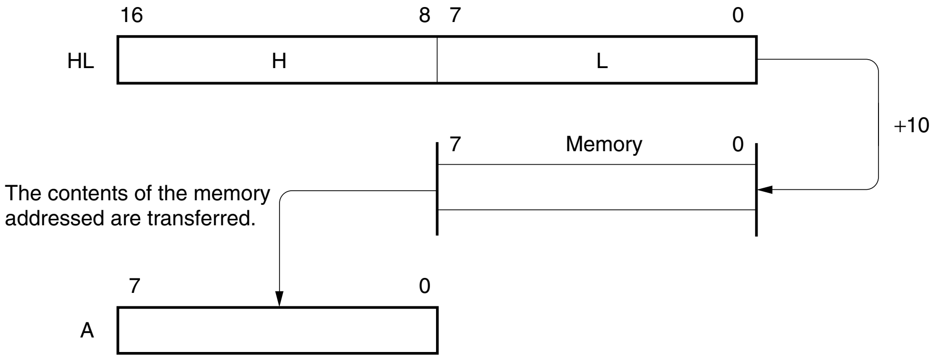
Identifier	Description
–	[HL + byte]

[Description example]

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

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

[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 the memory spaces.

[Operand format]

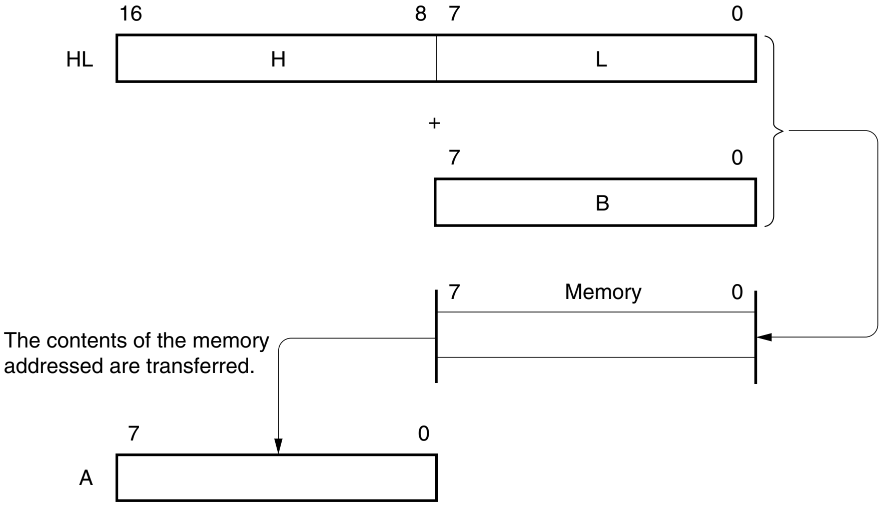
Identifier	Description
–	[HL + B], [HL + C]

[Description example]

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

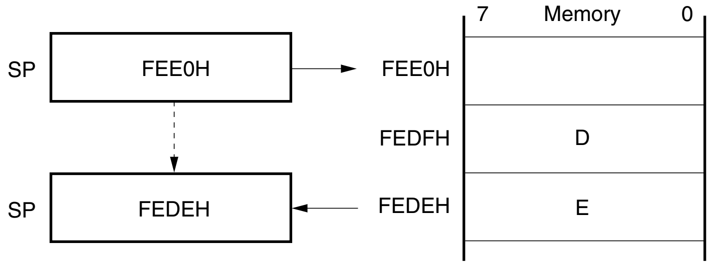
With stack addressing, only the internal high-speed RAM area can be accessed.

[Description example]

In the case of PUSH DE; (saving DE register)

Operation code 1 0 1 1 0 1 0 1

[Illustration]



CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

There are four types of pin I/O buffer power supplies: AV_{REF}, EV_{DD}, SMV_{DD}, 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}	P20 to P23, P24 to P27[Note]
EV_{DD}	Port pins other than P20 to P27, P80 to P87, P90 to P97, and P121 to P124
SMV_{DD}	• P80 to P87 • P90 to P97[Note]
V_{DD}	• P121 to P124 • Non-port pins

Note 78K0/DF2 only.

78K0/Dx2 products are provided with the ports shown in **Figure 4-1**, which enable variety of control operations.

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

The 78K0/DE2 has a total of 47 I/O ports, ports 0 to 3, 6 to 8, and 12. Also, the 78K0/DF2 has a total of 63 I/O ports, ports 0 to 3, 6 to 9, and 12. The port configuration is shown below.

Figure 4-1. Port Types (1/2)

(a) 78K0/DE2

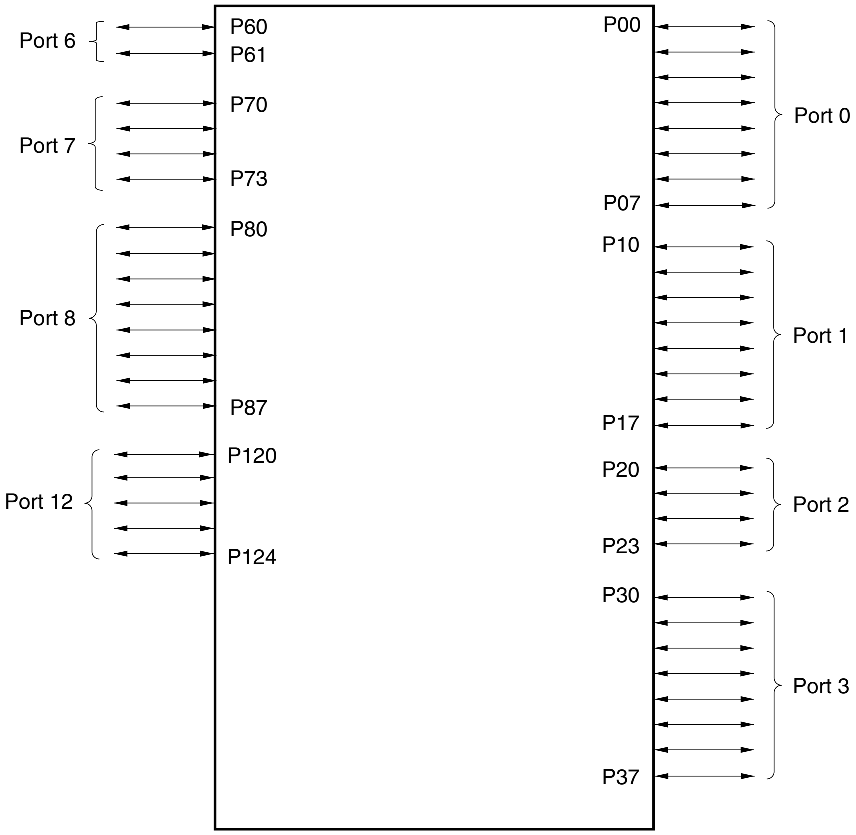
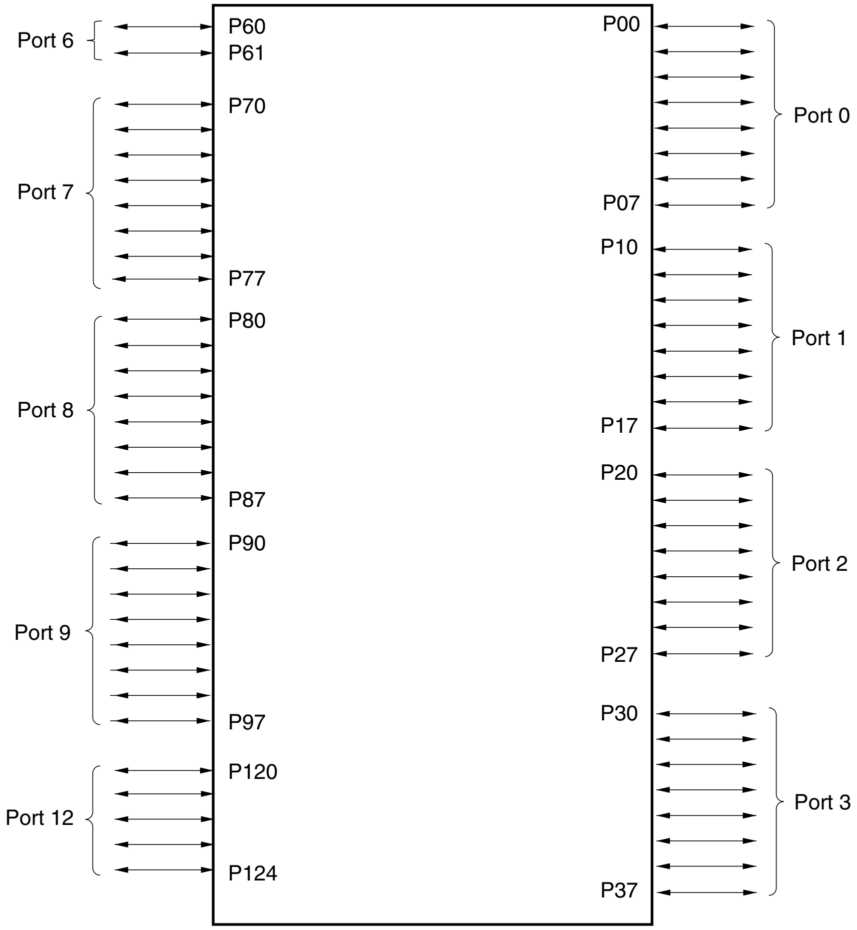


Figure 4-1. Port Types (2/2)

(b) 78K0/DF2



4.2 Port Configuration

Ports include the following hardware.

Table 4-2. Port Configuration

Item		Configuration	
		78K0/DE2	78K0/DF2
Control register	Port mode register	PM0 to PM3, PM6 to PM8, PM12	PM0 to PM3, PM6 to PM9, PM12
	Port register	P0 to P3, P6 to P8, P12	P0 to P3, P6 to P9, P12
	Pull-up resistor option register	PU0, PU1, PU3, PU6, PU7, PU12	
Port	Total	47 (CMOS I/O: 45, N-ch open drain I/O[Note]: 2)	63 (CMOS I/O: 61, N-ch open drain I/O[Note]: 2)
Pull-up resistor	Total	31	41

Note 5 V tolerant / N-ch open-drain output selectable.

4.2.1 Port 0

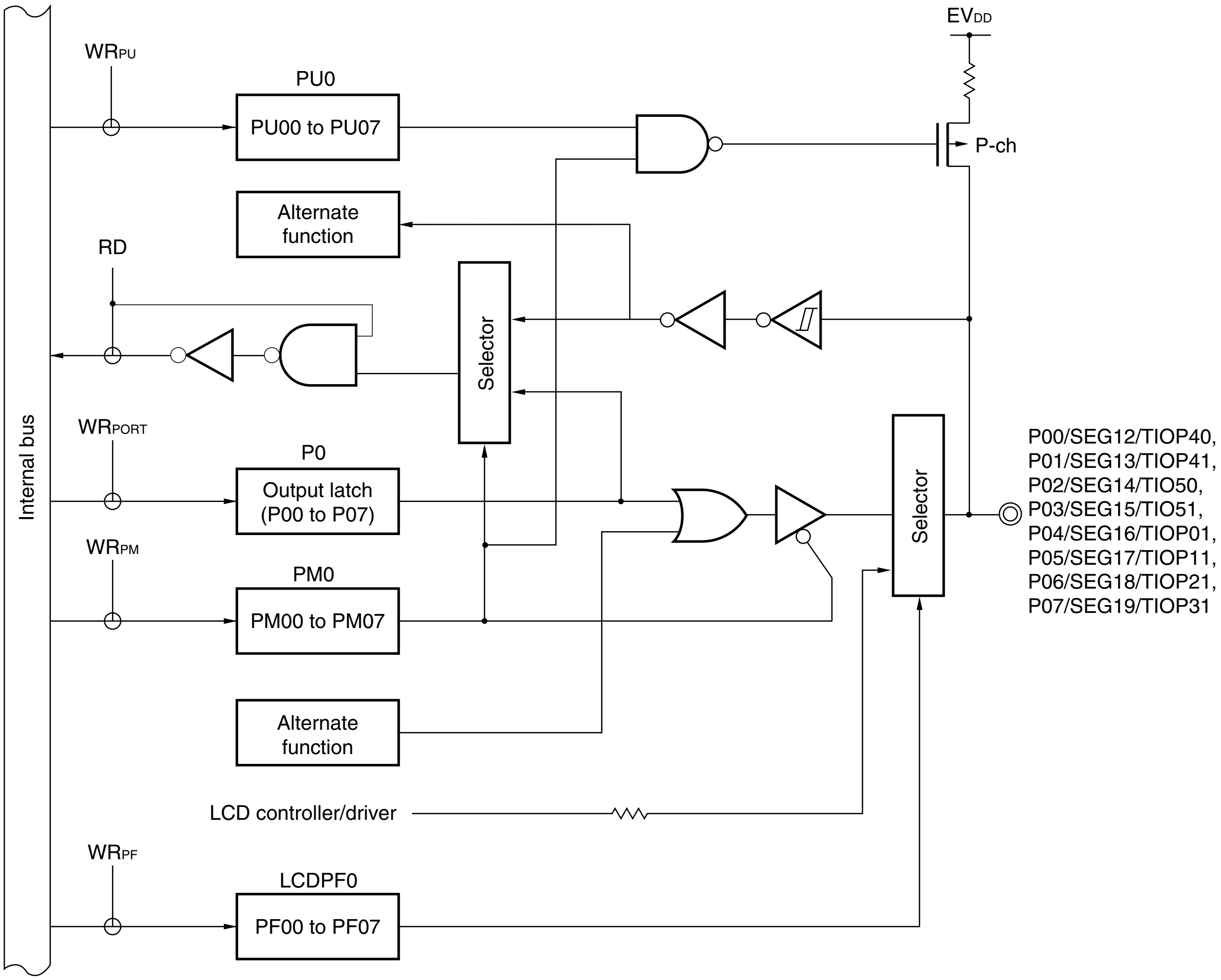
Port 0 is an 8-bit I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 to P07 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register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

This port can also be used for timer I/O and segment signal outputs for the LCD controller/driver.

Reset signal generation sets port 0 to input mode.

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

Figure 4-2. Block Diagram of P00 and P07



P0:	Port register 0
PU0:	Pull-up resistor option register 0
PM0:	Port mode register 0
LCDPF0:	LCD port function register 0
RD:	Read signal
WR××:	Write signal

4.2.2 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 interface data I/O, clock I/O, timer I/O, external interrupt request input, and segment signal outputs for the LCD controller/driver.

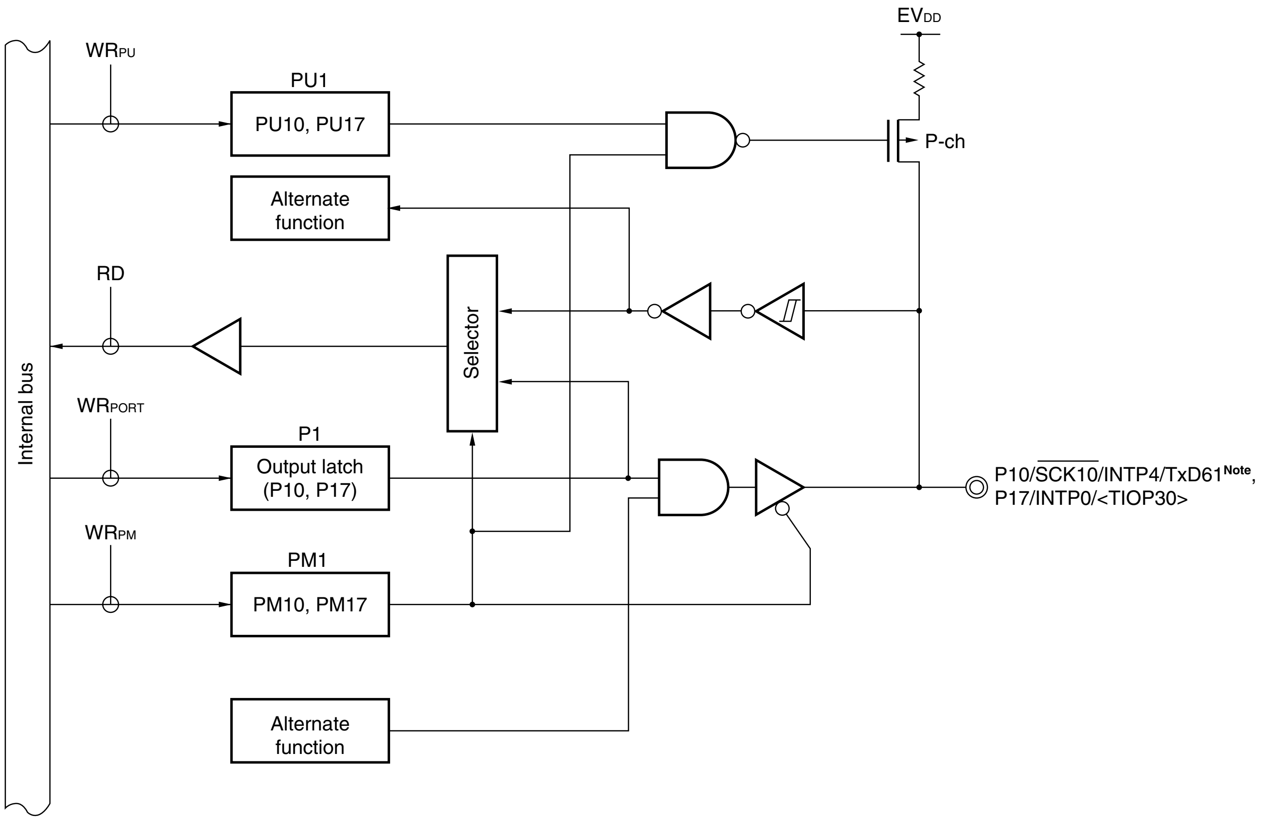
Reset signal generation sets port 1 to input mode.

Figures 4-3 to **4-8** show block diagrams of port 1.

Cautions 1. To use P10/$\overline{\text{SCK10}}$/INTP4/TxD61 and P12/SO10/INTP2 as general-purpose ports, set serial operation mode register 10 (CSIM10) and serial clock selection register 10 (CSIC10) to the default status (00H).

2. To use P10/$\overline{\text{SCK10}}$/INTP4/TxD61 and P13/SEG23/TIOP30/TxD60 as general-purpose port, clear bit 0 (TXDLV60, TXDLV61) of asynchronous serial interface control registers 60, 61 (ASICL60, ASICL61) to 0 (normal output of TxD60, TxD61).

Figure 4-3. Block Diagram of P10 and P17

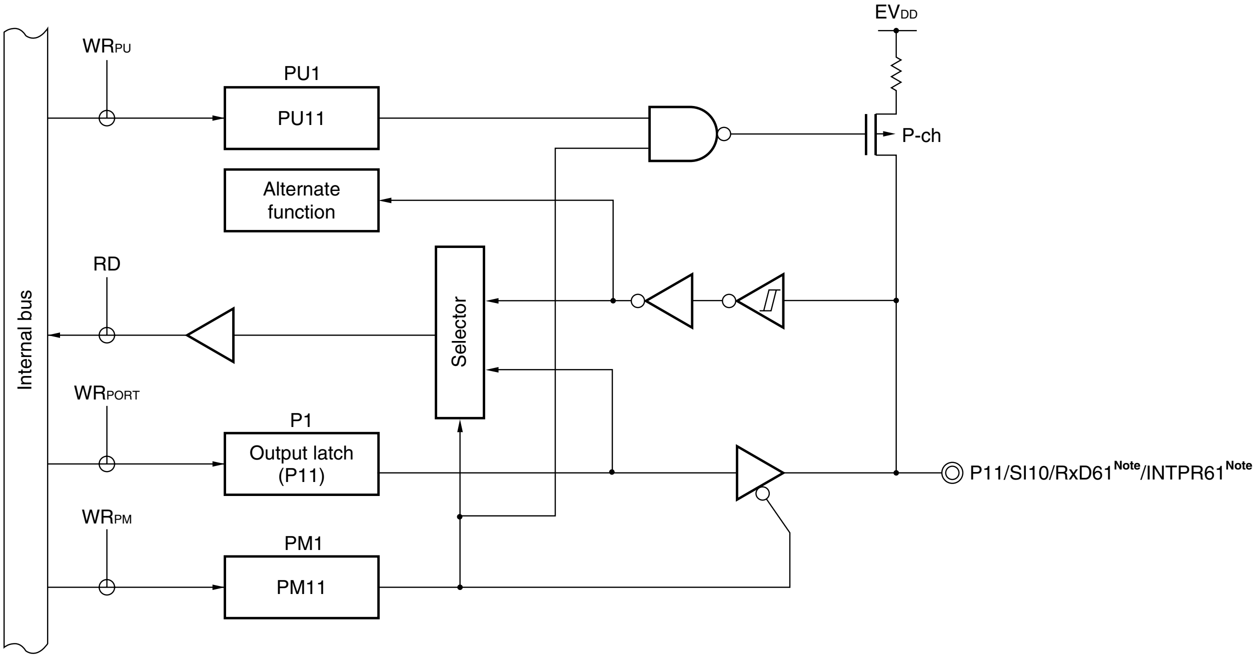


Note 78K0/DF2 only.

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

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

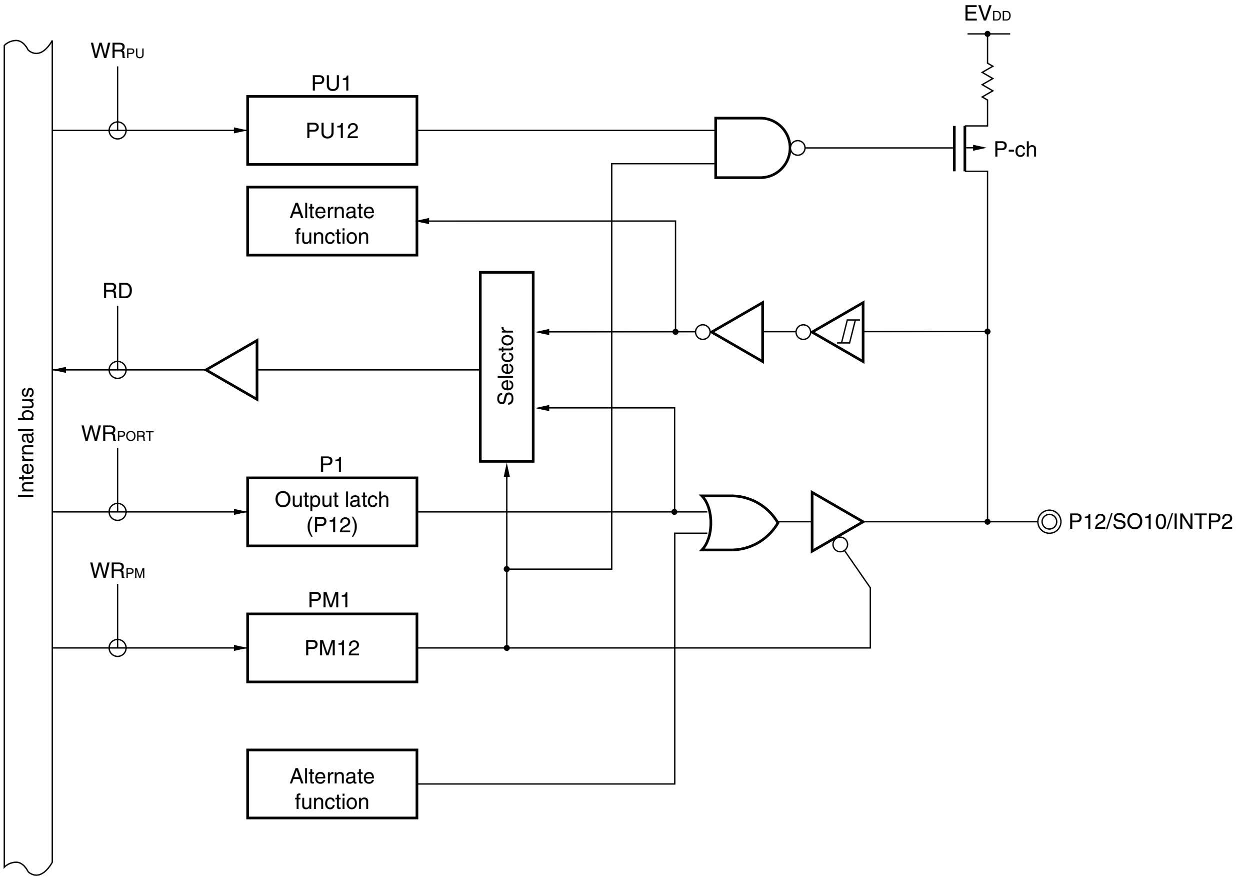
Figure 4-4. Block Diagram of P11



Note 78K0/DF2 only.

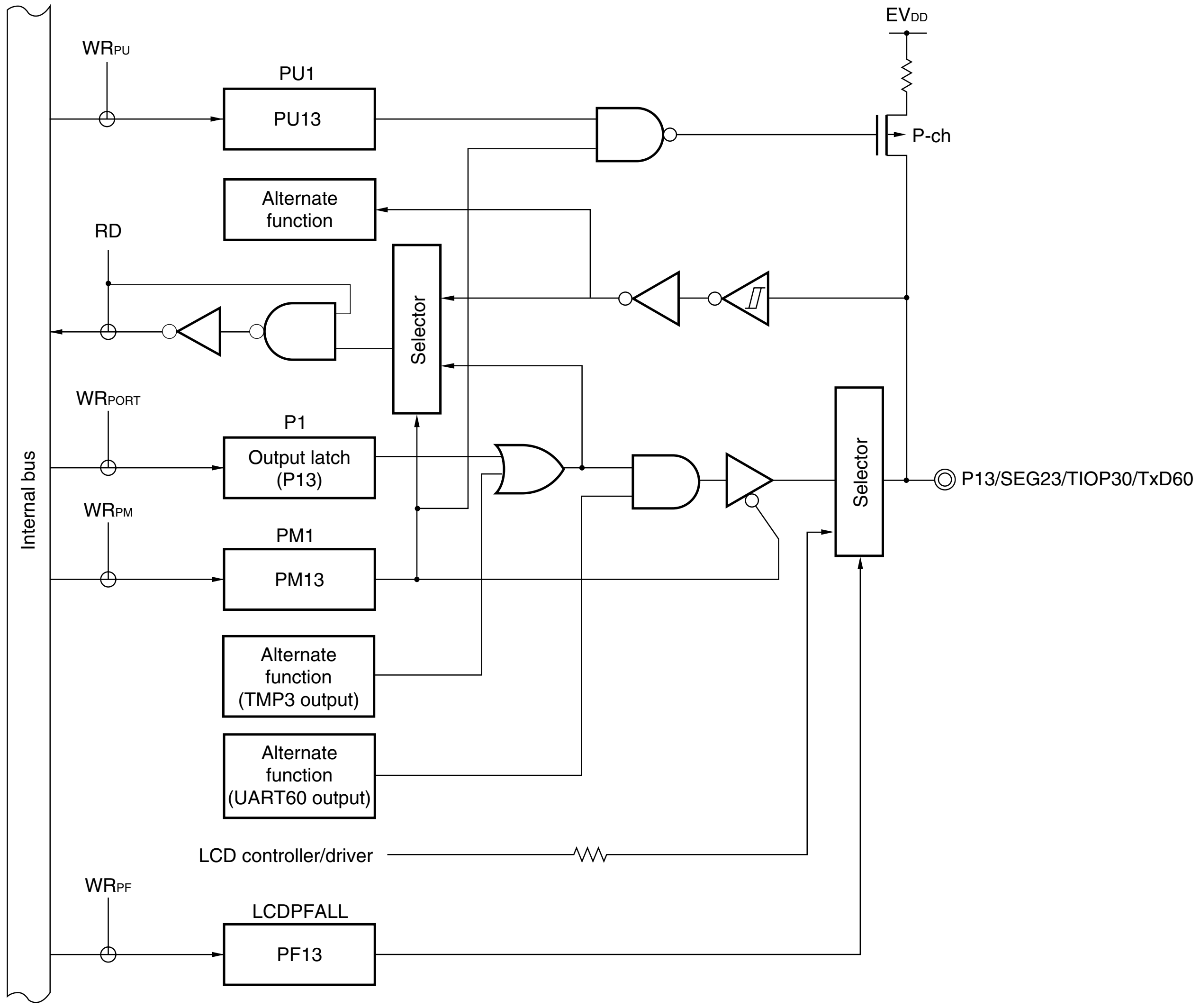
P1:	Port register 1
PU1:	Pull-up resistor option register 1
PM1:	Port mode register 1
RD:	Read signal
WR××:	Write signal

Figure 4-5. Block Diagram of P12



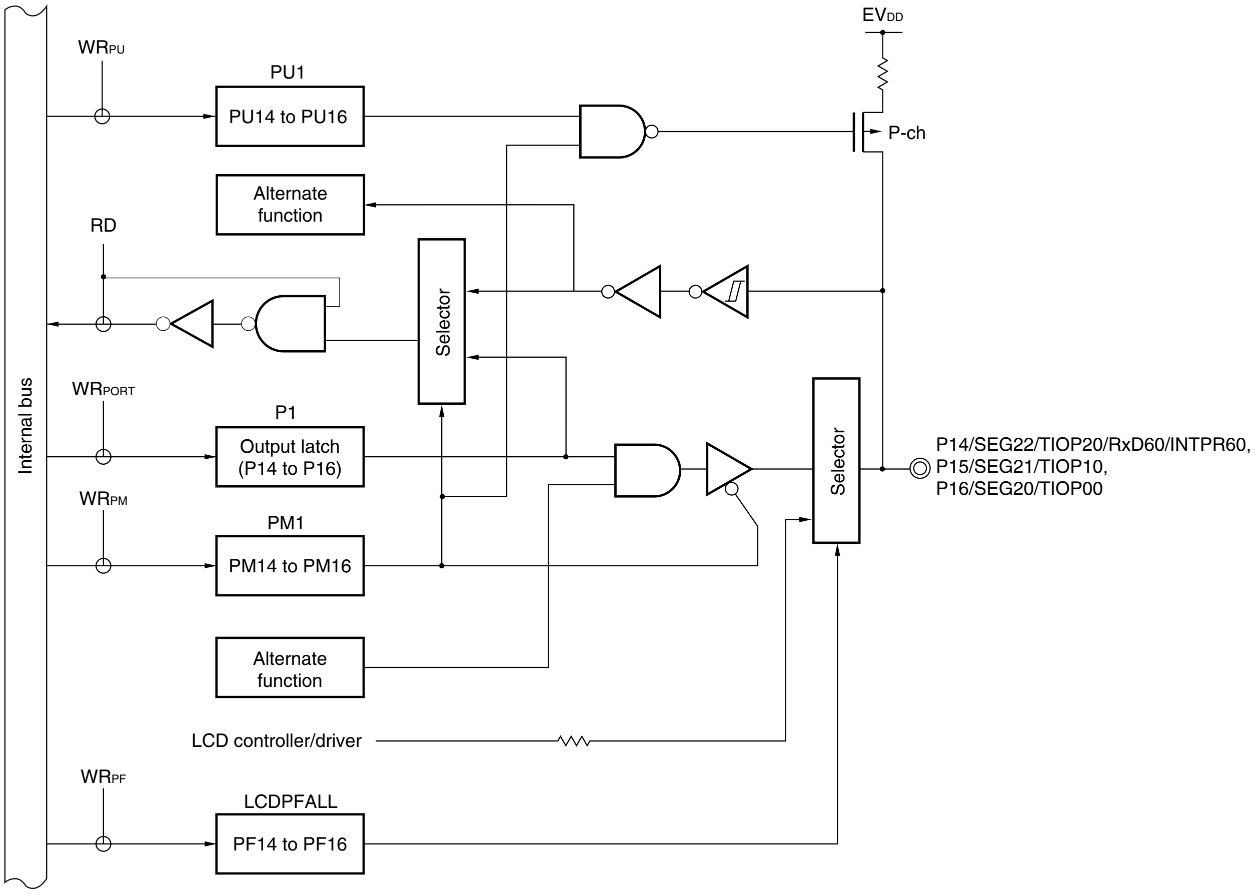
P1:	Port register 1
PU1:	Pull-up resistor option register 1
PM1:	Port mode register 1
RD:	Read signal
WR××:	Write signal

Figure 4-6. Block Diagram of P13



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

Figure 4-7. Block Diagram of P14 to P16



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

4.2.3 Port 2

Port 2 is a 4-bit I/O port in 78K0/DE2 and an 8-bit I/O port in 78K0/DF2 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 A/D converter analog input.

To use P20/ANI0 to P27/ANI7 (P20/ANI0 to P23/ANI3 in 78K0/DE2, P20/ANI0 to P27/ANI7 in 78K0/DF2) as digital input pins, set them in the digital I/O mode by using the A/D port configuration register (ADPC) and in the input mode by using PM2 (for details, see **11.3 (5) A/D port configuration register (ADPC)**). Use these pins starting from the lower bit.

Table 4-3. Setting Functions of P20/ANI0 to P27/ANI7 Pins

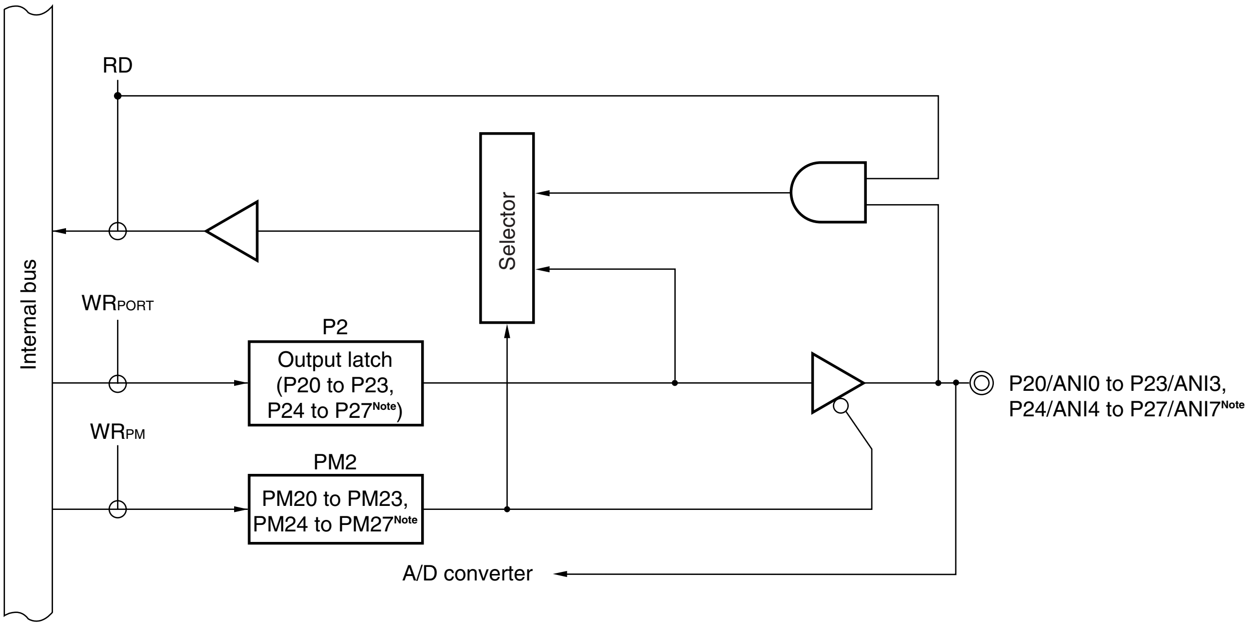
ADPC	PM2	ADS	P20/ANI0 to P27/ANI7 Pin
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

All P20/ANI0 to P27/ANI7 are set in the analog input mode when the reset signal is generated.

Figure 4-8 shows a block diagram of port 2.

Caution Make the AV_{REF} pin the same potential as the V_{DD} pin when port 2 is used as a digital port.

Figure 4-8. Block Diagram of P20 to P27



Note 78K0/DF2 only

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

4.2.4 Port 3

Port 3 is an 8-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 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 segment signal outputs for the LCD controller/driver.

Reset signal generation sets port 3 to input mode.

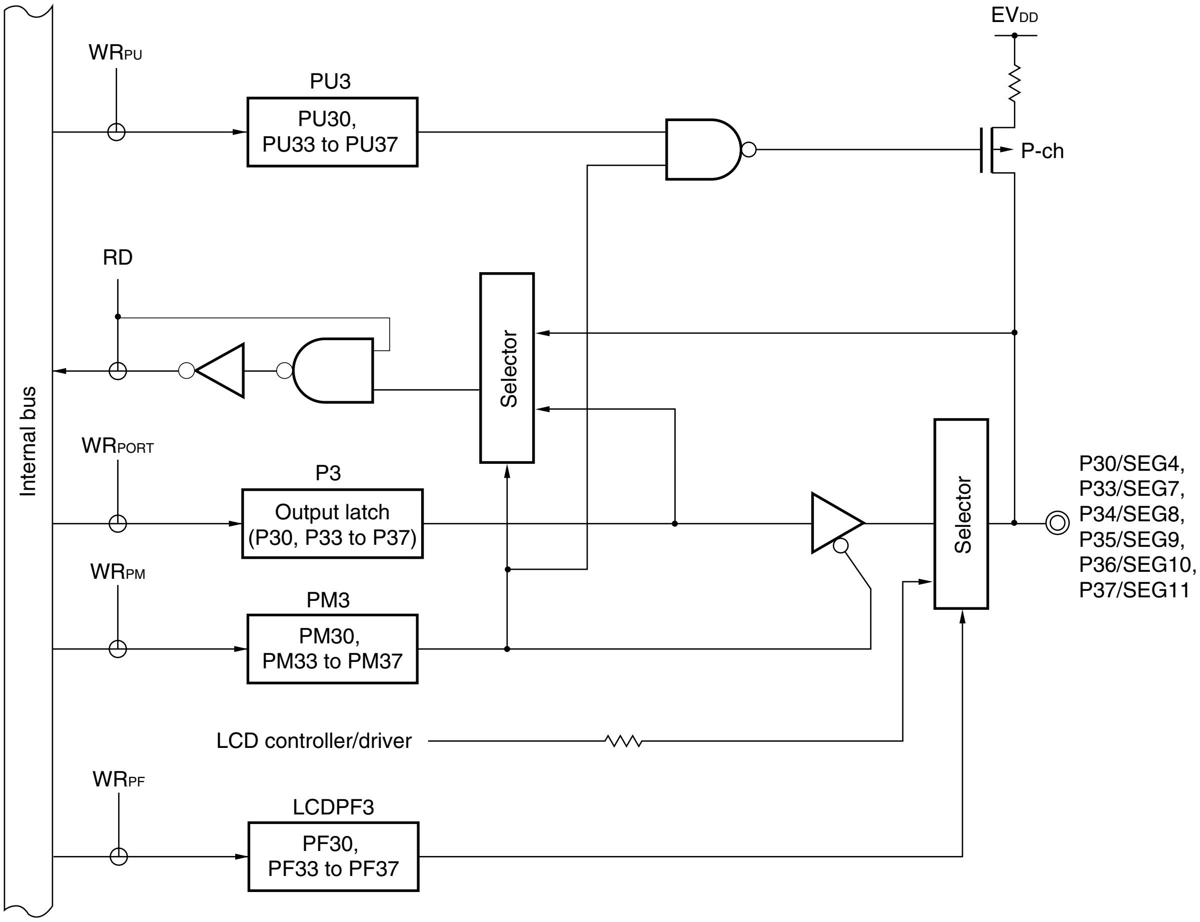
Figures 4-9 and **4-13** show block diagrams of port 3.

Cautions 1. Be sure to pull the P31/SEG5/OCD1A pin down before a reset release, to prevent malfunction.

<R> **2. Set P31 and P32 as follows when these pins are used for on-chip debug. In this case, P31/SEG5/OCD1A and P32/SEG6/OCD1B pins are used only as OCD1A or OCD1B functions.**
- P31/SEG5/OCD1A: Set to input mode (PM31 = 1) and port mode (PF31 = 0).
- P32/SEG6/OCD1B: Set to port mode (PF32 = 0).
When P31/SEG5 and P32/SEG6 functions need to be evaluated with on-chip debug, use OCD0A and OCD0B.

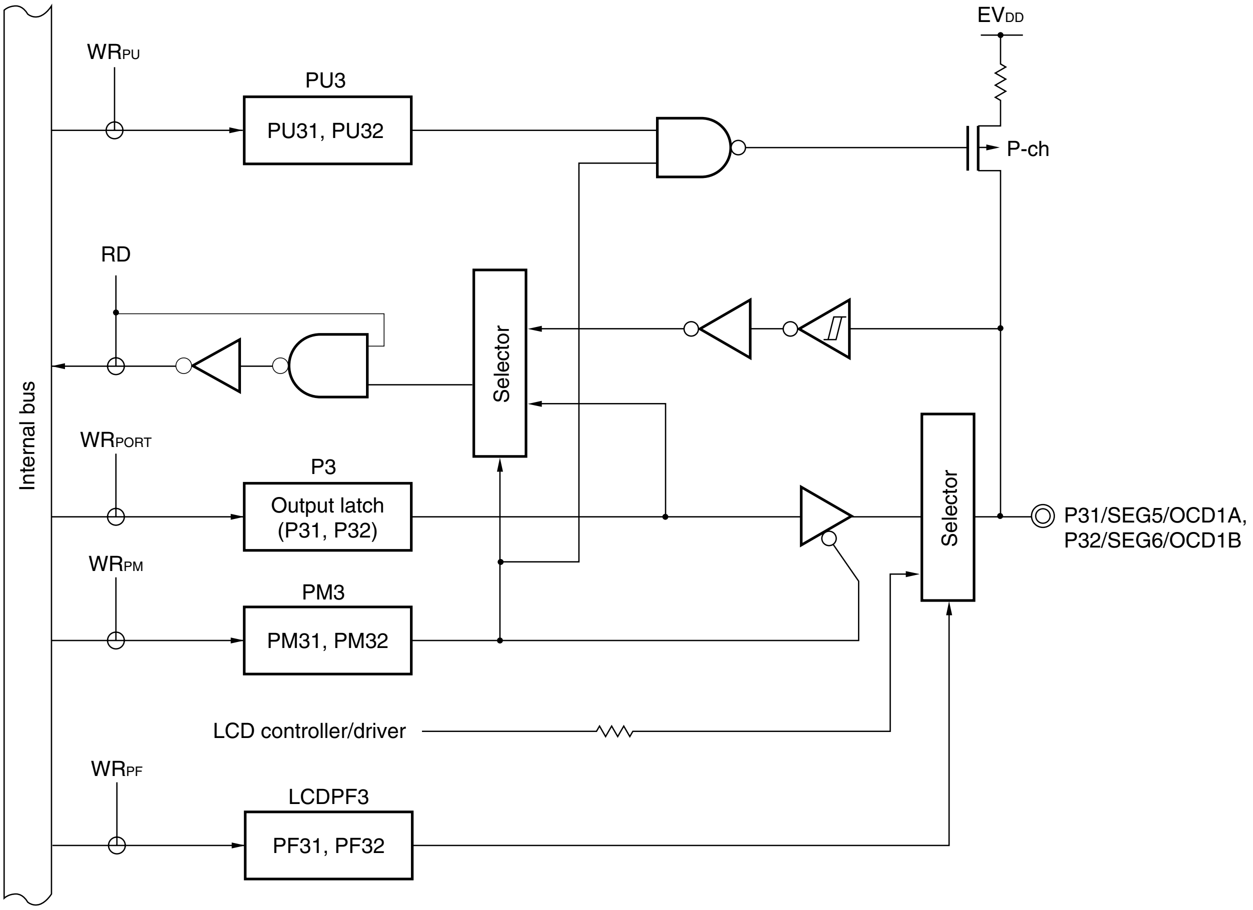
Remark The P31/SEG5 and P32/SEG6 pins can be used for on-chip debug mode setting (OCD1A, OCD1B) when the on-chip debug function is used. For how to connect an in-circuit emulator supporting on-chip debugging (QB-78K0MINI or QB-MINI2), see **CHAPTER 27 ON-CHIP DEBUG FUNCTION**.

Figure 4-9. Block Diagram of P30 and P33 to 37



P3:	Port register 3
PU3:	Pull-up resistor option register 3
PM3:	Port mode register 3
LCDPF3:	LCD port function register 3
RD:	Read signal
WR××:	Write signal

Figure 4-10. Block Diagram of P31 and P32



P3:	Port register 3
PU3:	Pull-up resistor option register 3
PM3:	Port mode register 3
LCDPF3:	LCD port function register 3
RD:	Read signal
WR××:	Write signal

4.2.5 Port 6

Port 6 is a 2-bit I/O port with an output latch. Port 6 can be set to the input mode or output mode in 1-bit units using port mode register 6 (PM6). P60 and P61 use of an on-chip pull-up resistor can be specified in 1-bit units using pull-up resistor option register 6 (PU6).

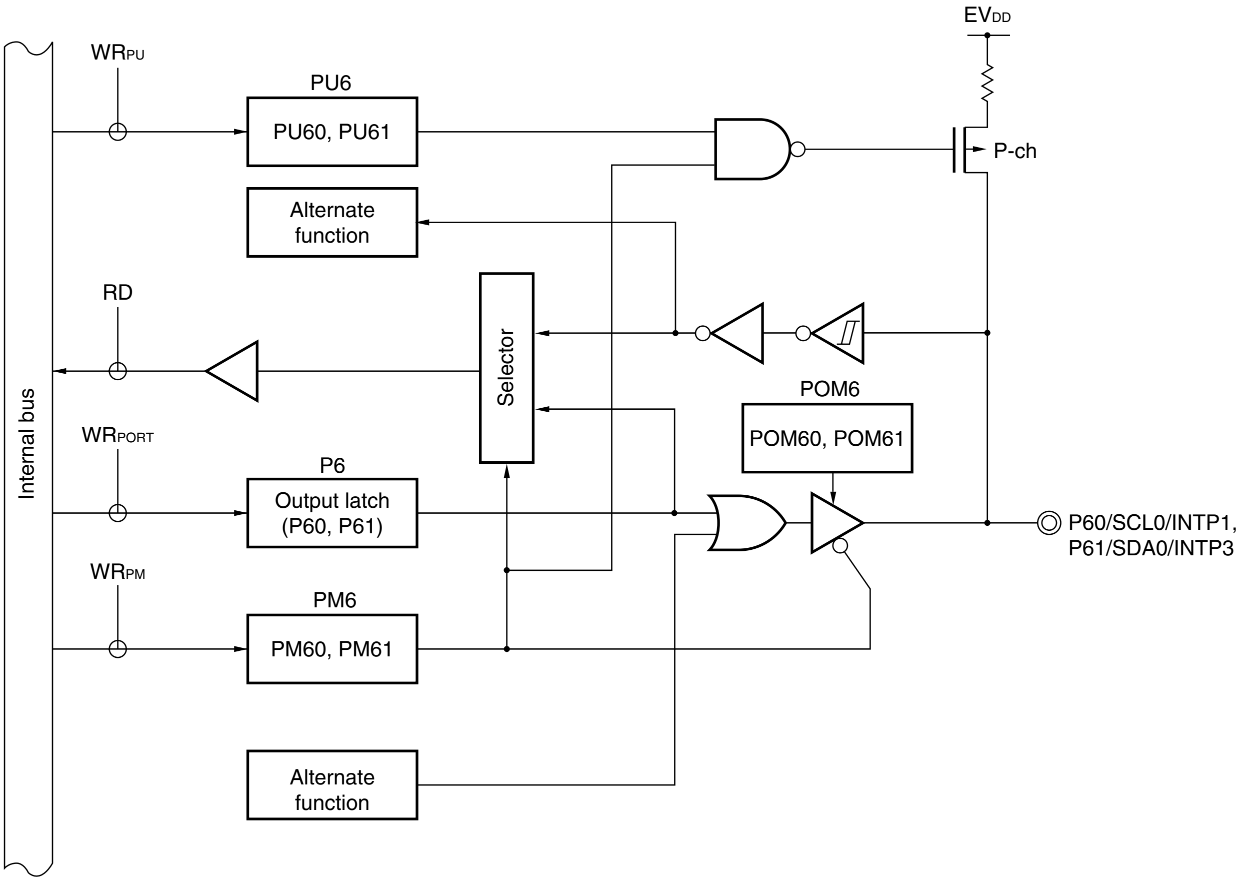
Output from the P60 and P61 pins can be specified as normal CMOS output or N-ch open-drain output (5 V-torelant) in 1-bit units, using port output mode register 6 (POM6).

This port can also be used for external interrupt request input and serial clock I/O and data I/O for IIC0.

Reset signal generation sets port 6 to input mode.

Figure 4-11 shows a block diagram of port 6.

Figure 4-11. Block Diagram of P60 and P61



P6:	Port register 6
PU6:	Pull-up resistor option register 6
PM6:	Port mode register 6
POM6:	Port output control register 6
RD:	Read signal
WR××:	Write signal

4.2.6 Port 7

Port 7 is a 4-bit I/O port in 78K0/DE2 and an 8-bit I/O port in 78K0/DF2 with an output latch. Port 7 can be set to the input mode or output mode in 1-bit units using port mode register 7 (PM7). When the P70 to P77 (P70 to P73 in 78K0/DE2, P70 to P77 in 78K0/DF2) 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 7 (PU7).

This port can also be used for output pins for the sound generator, clock output pins, buzzer output pins, CAN I/F I/O, serial interface data I/O and clock I/O, timer I/O, and segment output pins for the LCD controller/driver.

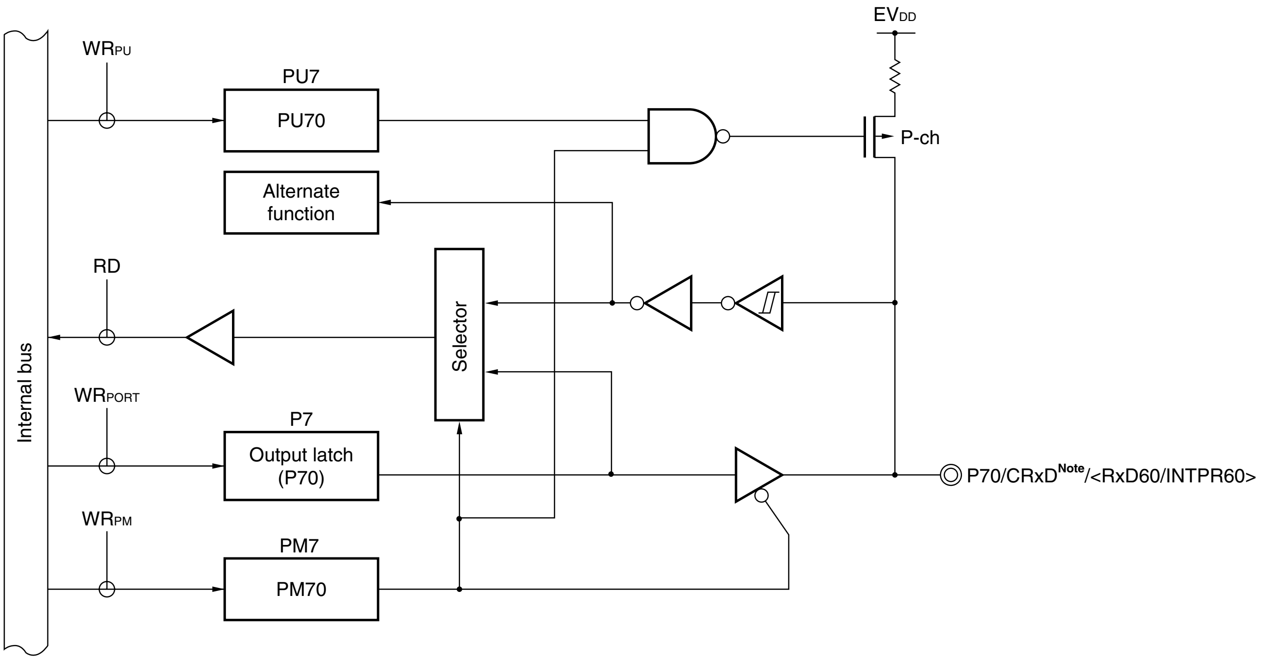
Reset signal generation sets port 7 to input mode.

Figures 4-15 to **4-21** show block diagrams of port 7.

Cautions 1. To use P74/$\overline{\text{SCK11}}$ and P76/SO11 as general-purpose ports, set serial operation mode register 10 (CSIM10) and serial clock selection resister 10 (CSIC10) to the default status (00H).

2. To use P77/$\overline{\text{SSI11}}$/SEG27 as general-purpose ports, set serial operation mode register 11 (CSIM11) to the default status (00H).

Figure 4-12. Block Diagram of P70

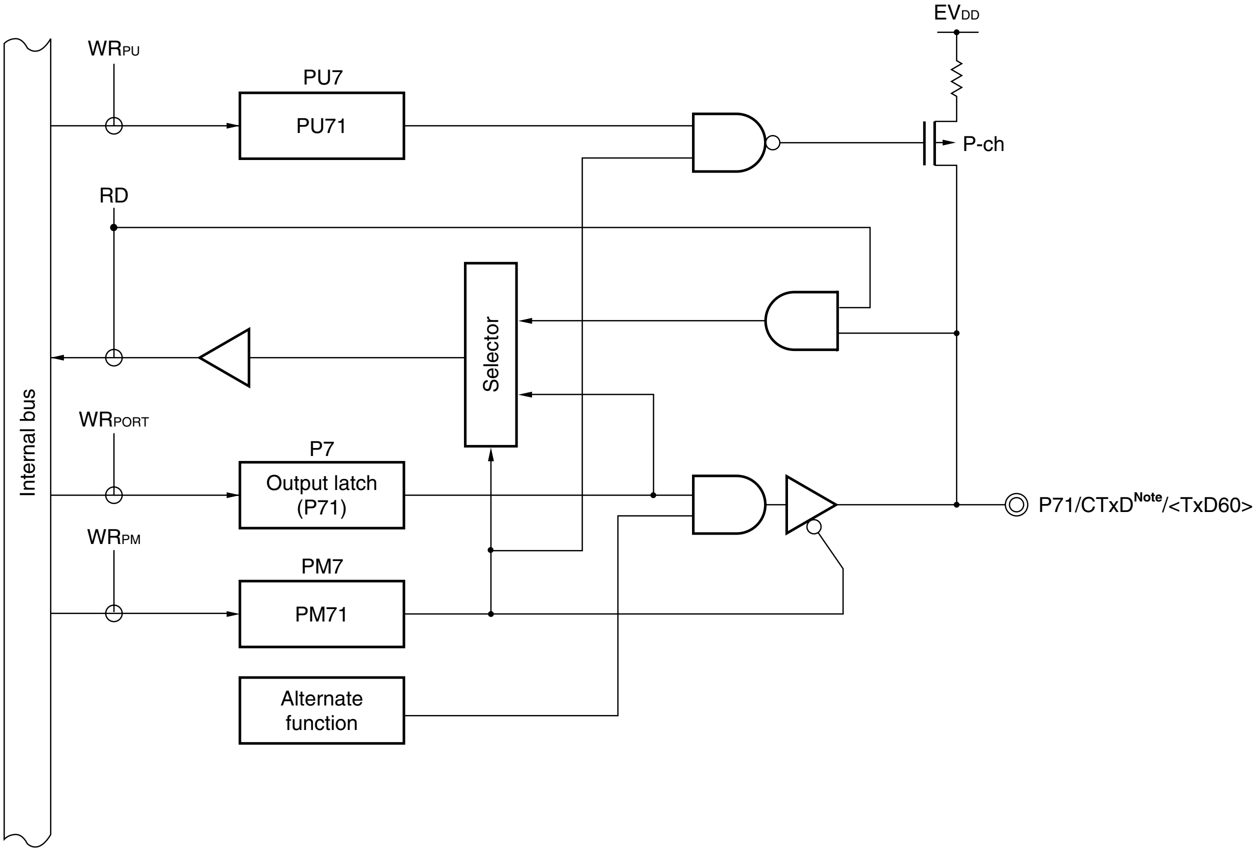


Note μPD78F0844, 78F0845, 78F0846, 78F0847, 78F0848, and 78F0849 only.

P7:	Port register 7
PU7:	Pull-up resistor option register 7
PM7:	Port mode register 7
RD:	Read signal
WR××:	Write signal

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

Figure 4-13. Block Diagram of P71

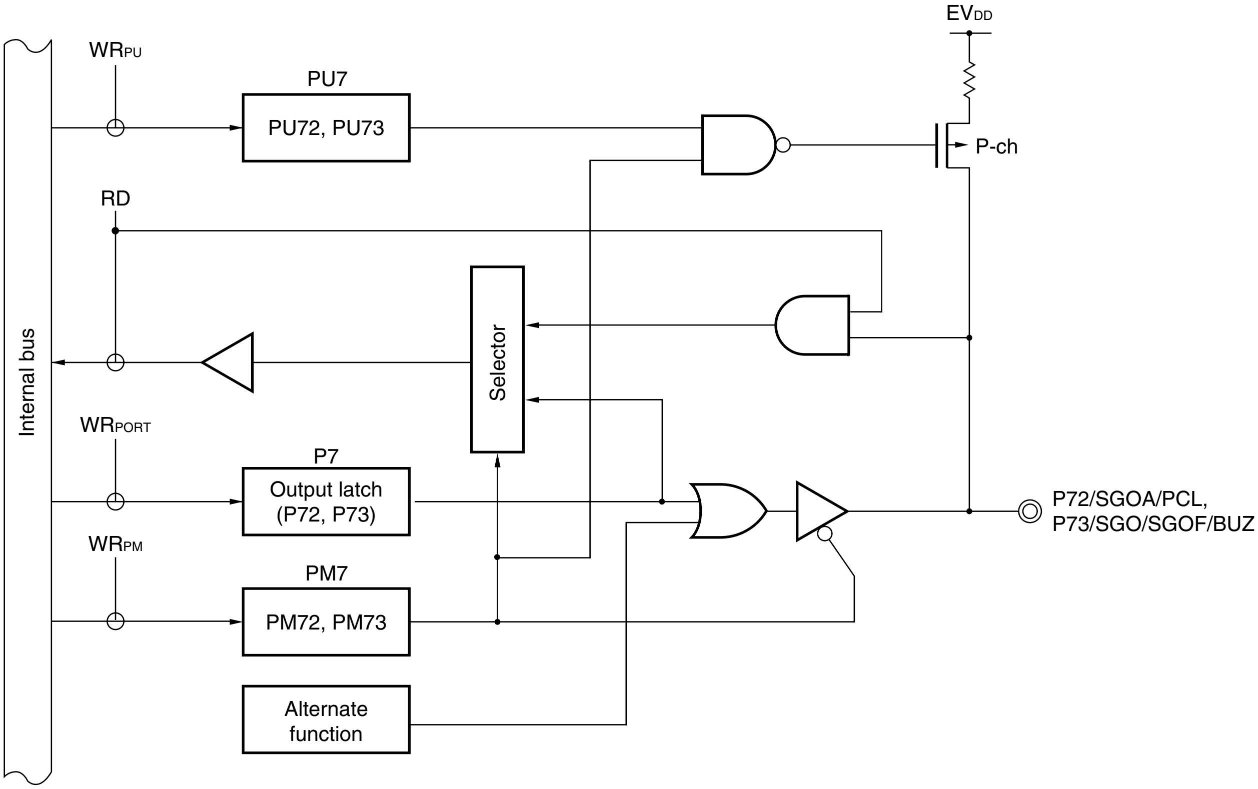


Note µPD78F0844, 78F0845, 78F0846, 78F0847, 78F0848, and 78F0849 only.

P7:	Port register 7
PU7:	Pull-up resistor option register 7
PM7:	Port mode register 7
RD:	Read signal
WR××:	Write signal

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

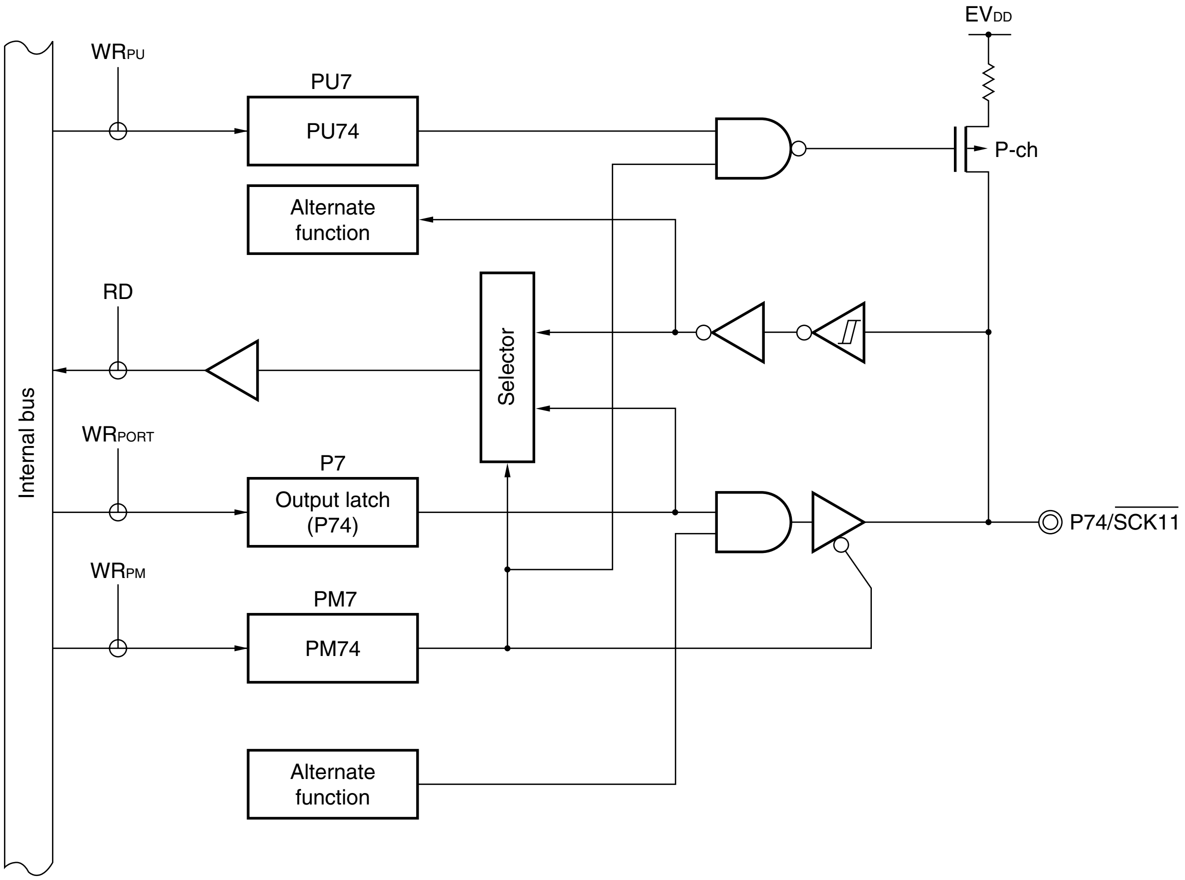
Figure 4-14. Block Diagram of P72 and P73



P7:	Port register 7
PU7:	Pull-up resistor option register 7
PM7:	Port mode register 7
RD:	Read signal
WR××:	Write signal

Figure 4-15. Block Diagram of P74 (1/2)

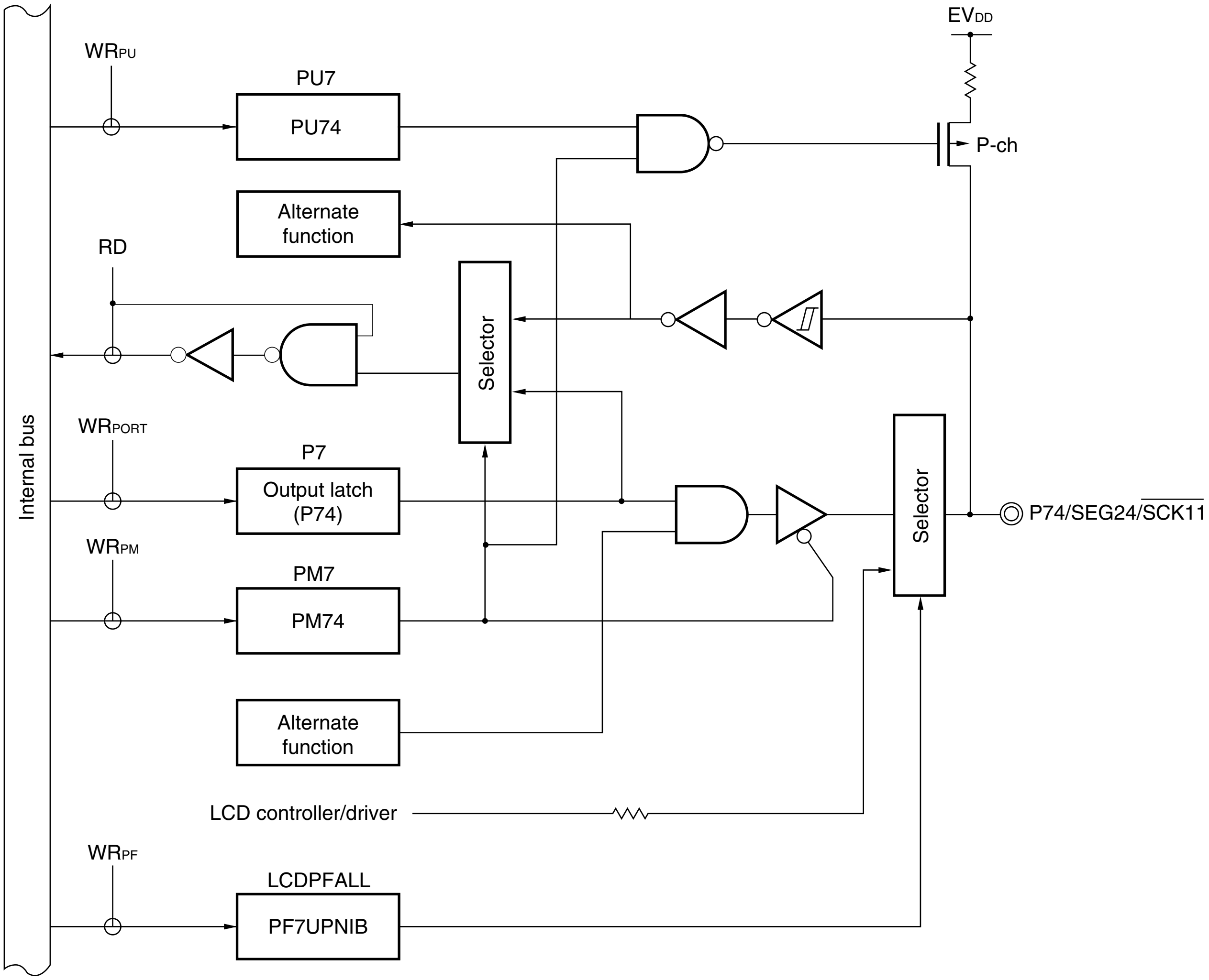
(a) μ PD78F038, 78F039, 78F040, 78F041, 78F036, and 78F047



P7:	Port register 7
PU7:	Pull-up resistor option register 7
PM7:	Port mode register 7
RD:	Read signal
WR××:	Write signal

Figure 4-15. Block Diagram of P74 (2/2)

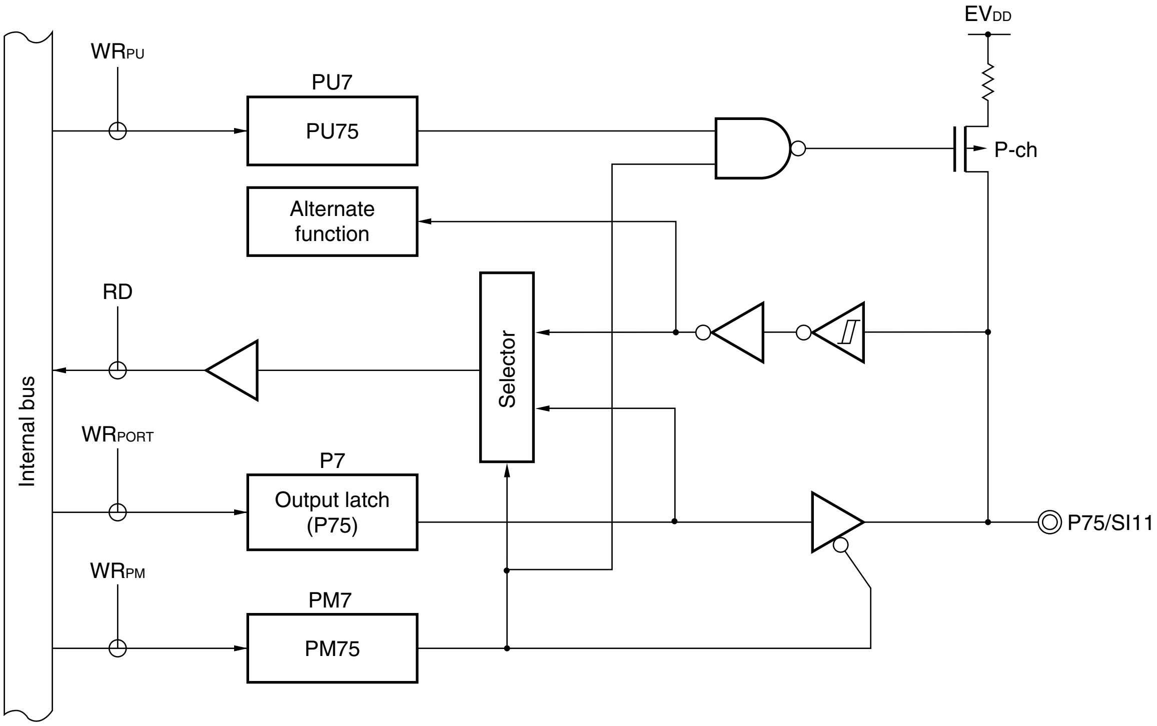
(b) μ PD78F042, 78F043, 78F048, and 78F049



P7:	Port register 7
PU7:	Pull-up resistor option register 7
PM7:	Port mode register 7
LCDPFALL:	LCD port function register ALL
RD:	Read signal
WR××:	Write signal

Figure 4-16. Block Diagram of P75 (1/2)

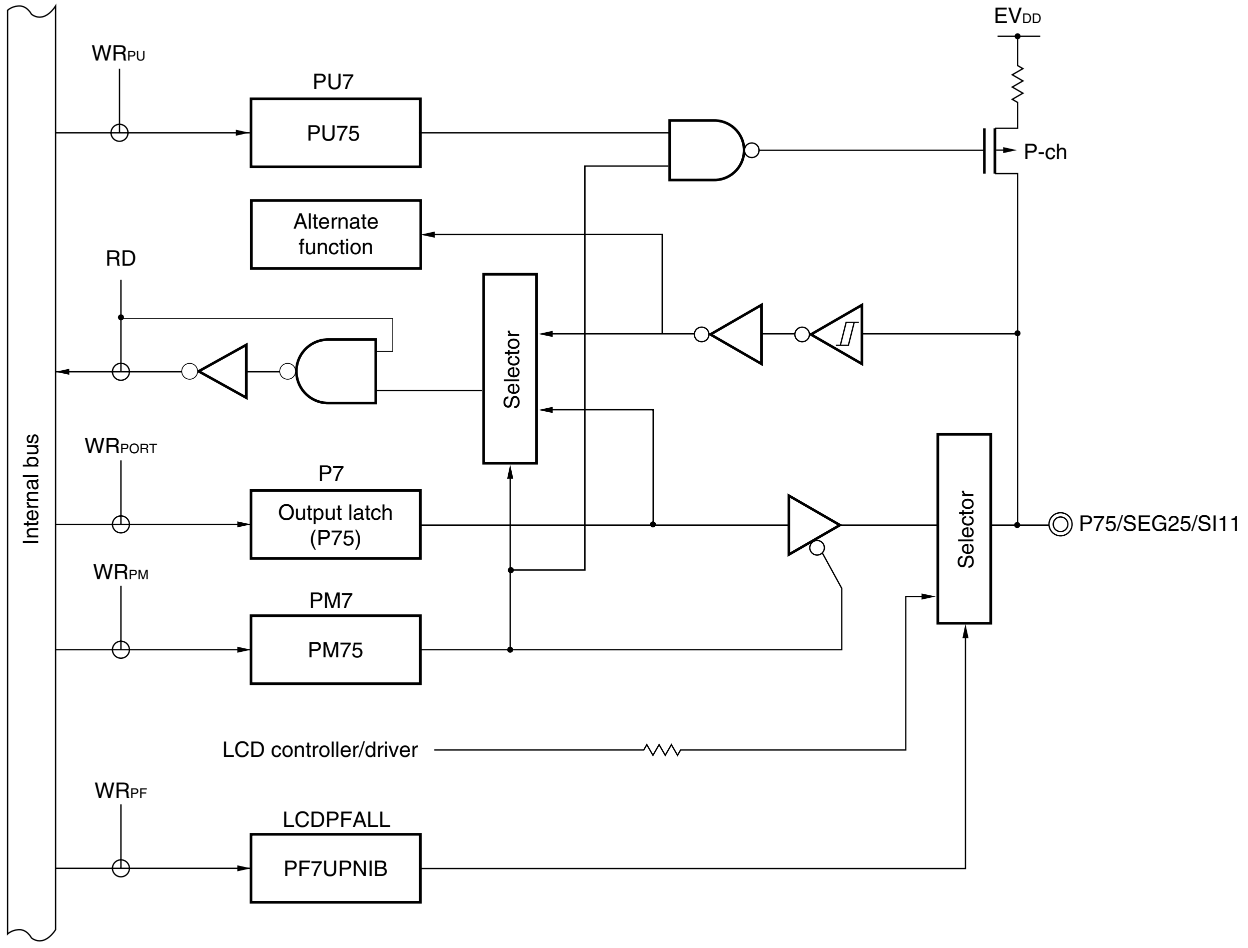
(a) μ PD78F038, 78F039, 78F040, 78F041, 78F036, and 78F047



P7:	Port register 7
PU7:	Pull-up resistor option register 7
PM7:	Port mode register 7
RD:	Read signal
WR××:	Write signal

Figure 4-16. Block Diagram of P75 (2/2)

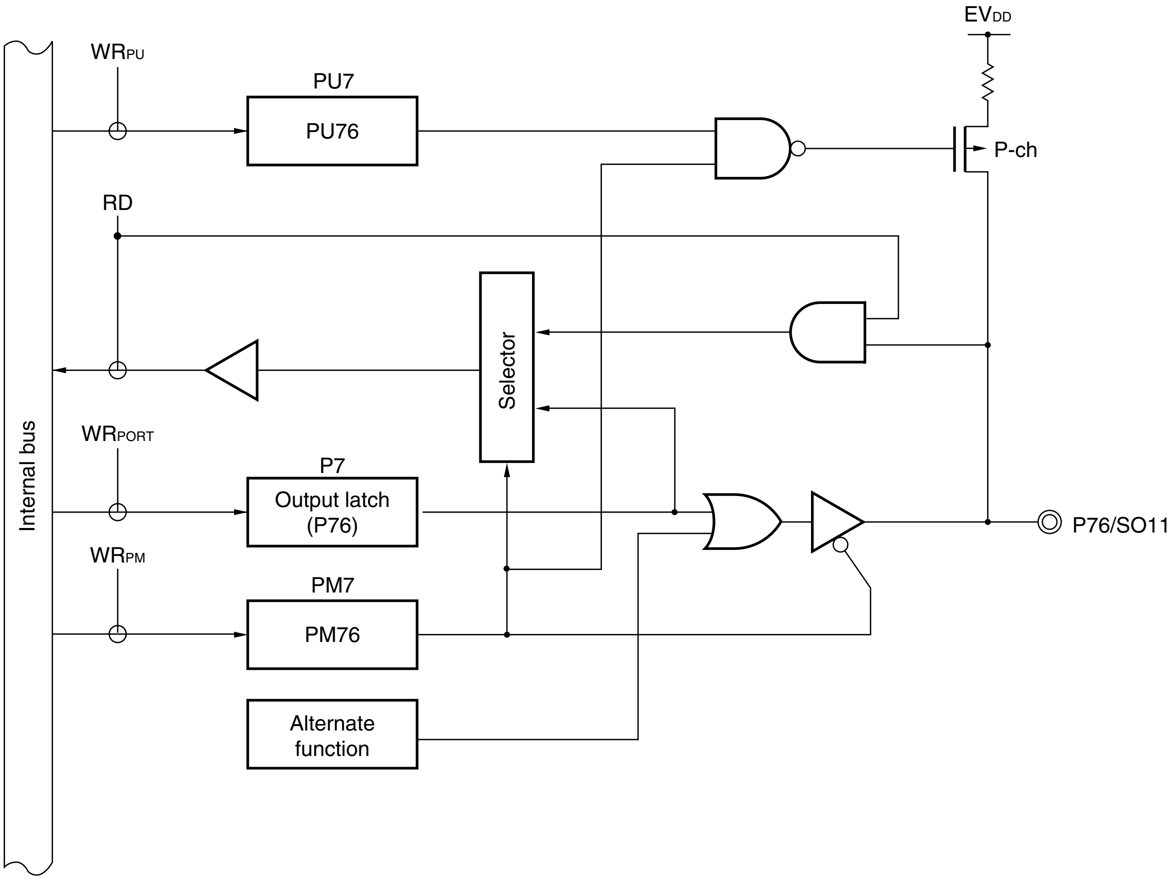
(b) μ PD78F042, 78F043, 78F048, and 78F049



P7:	Port register 7
PU7:	Pull-up resistor option register 7
PM7:	Port mode register 7
LCDPFALL:	LCD port function register ALL
RD:	Read signal
WR××:	Write signal

Figure 4-17. Block Diagram of P76 (1/2)

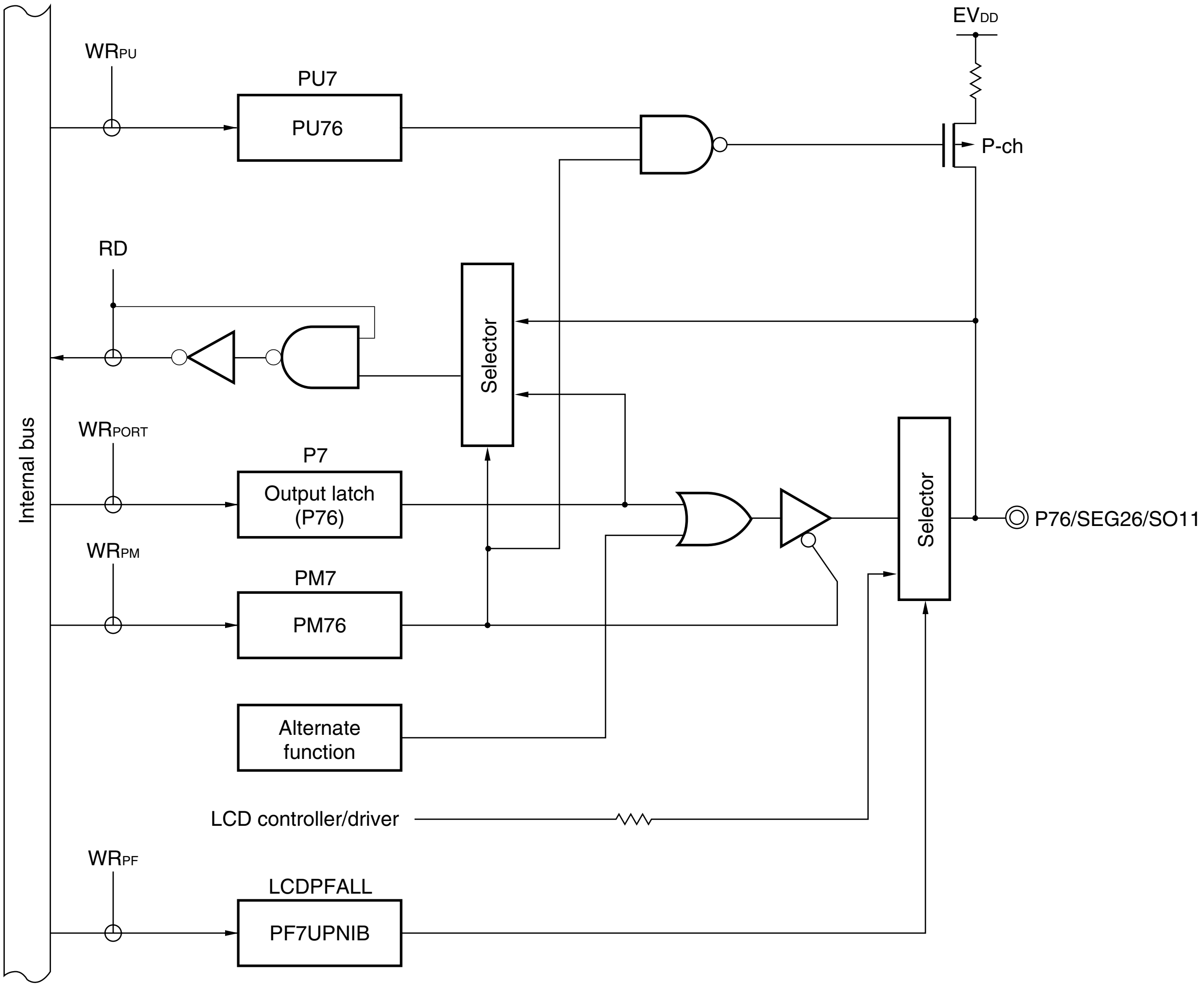
(a) μPD78F038, 78F039, 78F040, 78F041, 78F036, and 78F047



P7:	Port register 7
PU7:	Pull-up resistor option register 7
PM7:	Port mode register 7
RD:	Read signal
WR××:	Write signal

Figure 4-17. Block Diagram of P76 (2/2)

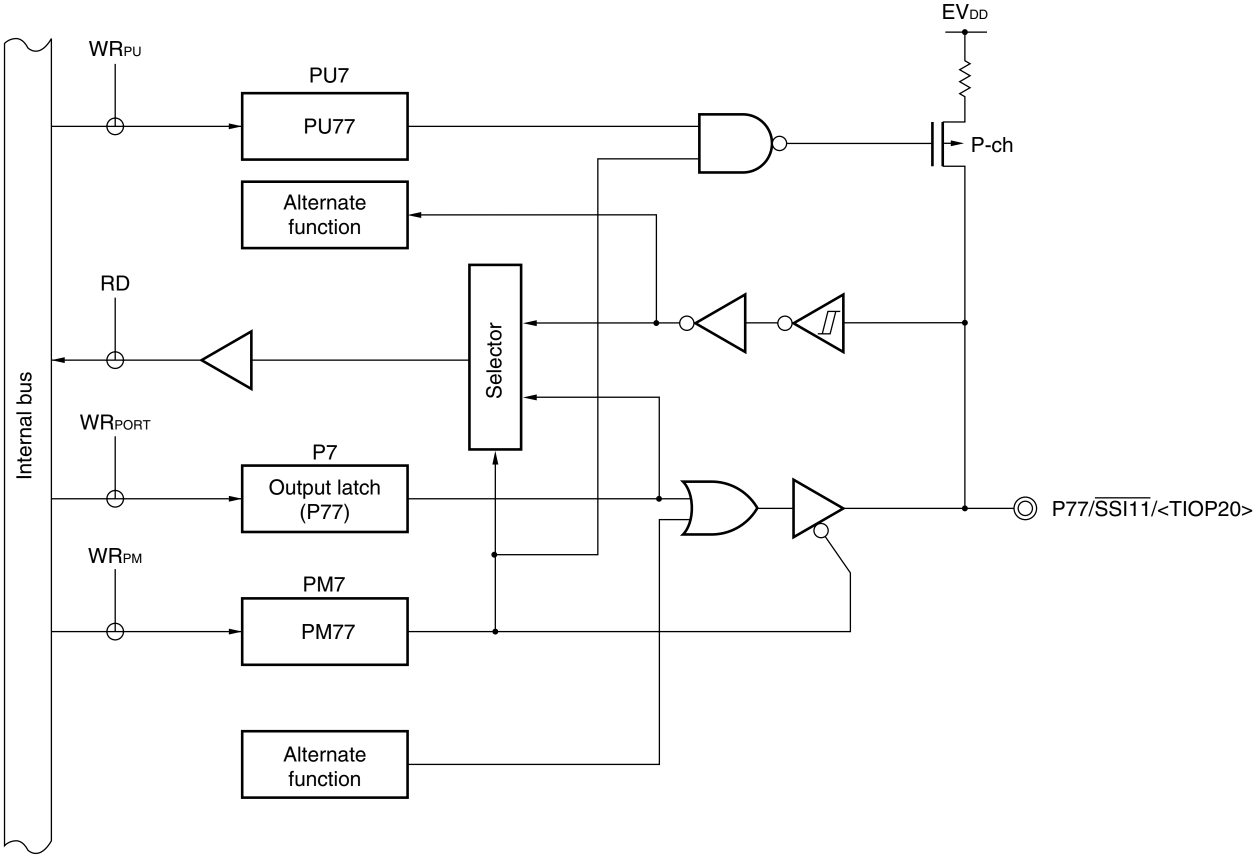
(b) μ PD78F042, 78F043, 78F048, and 78F049



P7:	Port register 7
PU7:	Pull-up resistor option register 7
PM7:	Port mode register 7
LCDPFALL:	LCD port function register ALL
RD:	Read signal
WR××:	Write signal

Figure 4-18. Block Diagram of P77 (1/2)

(a) μ PD78F038, 78F039, 78F040, 78F041, 78F036, and 78F047

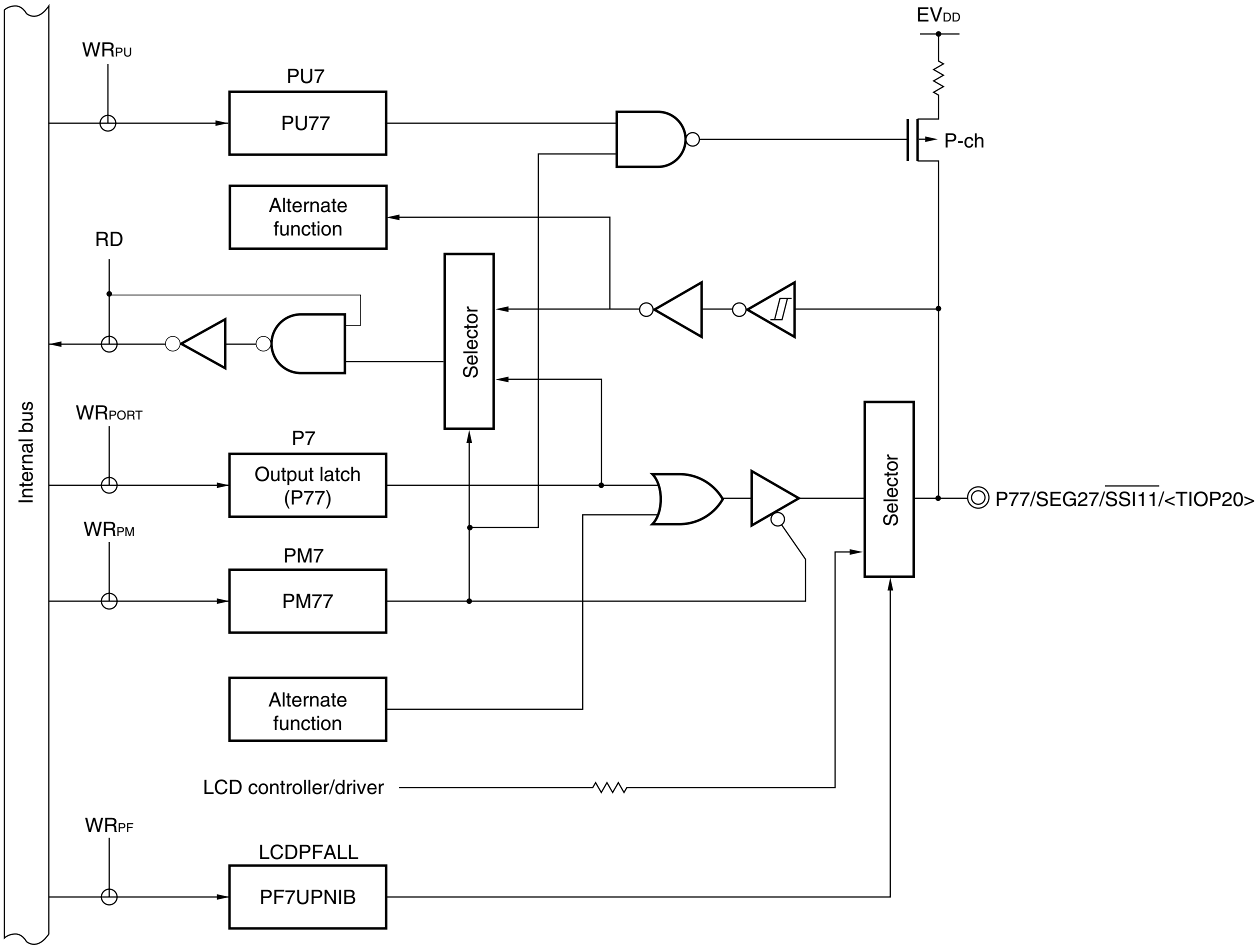


P7:	Port register 7
PU7:	Pull-up resistor option register 7
PM7:	Port mode register 7
RD:	Read signal
WR××:	Write signal

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

Figure 4-18. Block Diagram of P77 (2/2)

(b) μ PD78F042, 78F043, 78F048, and 78F049



P7:	Port register 7
PU7:	Pull-up resistor option register 7
PM7:	Port mode register 7
LCDPFALL:	LCD port function register ALL
RD:	Read signal
WR××:	Write signal

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

4.2.7 Port 8

Port 8 is an 8-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).

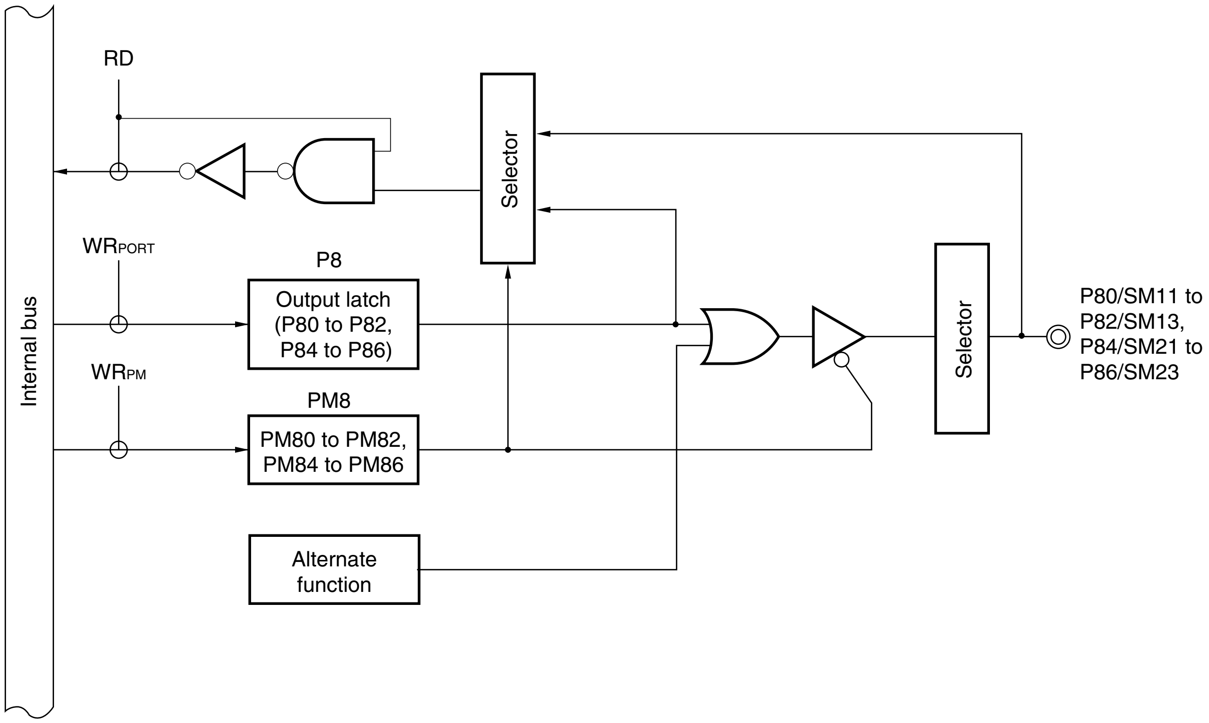
This port can also be used for stepper motor controller/driver outputs/inputs in μ PD78F0836, 78F0837, 78F0842, 78F0843, 78F0844, 78F0845, 78F0848, and 78F0849 or LCD segment outputs in μ PD78F0838, 78F0839, 78F0840, 78F0841, 78F0846, and 78F0847.

Reset signal generation sets port 8 to input mode.

Figures 4-22 and **4-23** show block diagrams of port 8.

Figure 4-19. Block Diagram of P80 to P82 and P84 to P86 (1/2)

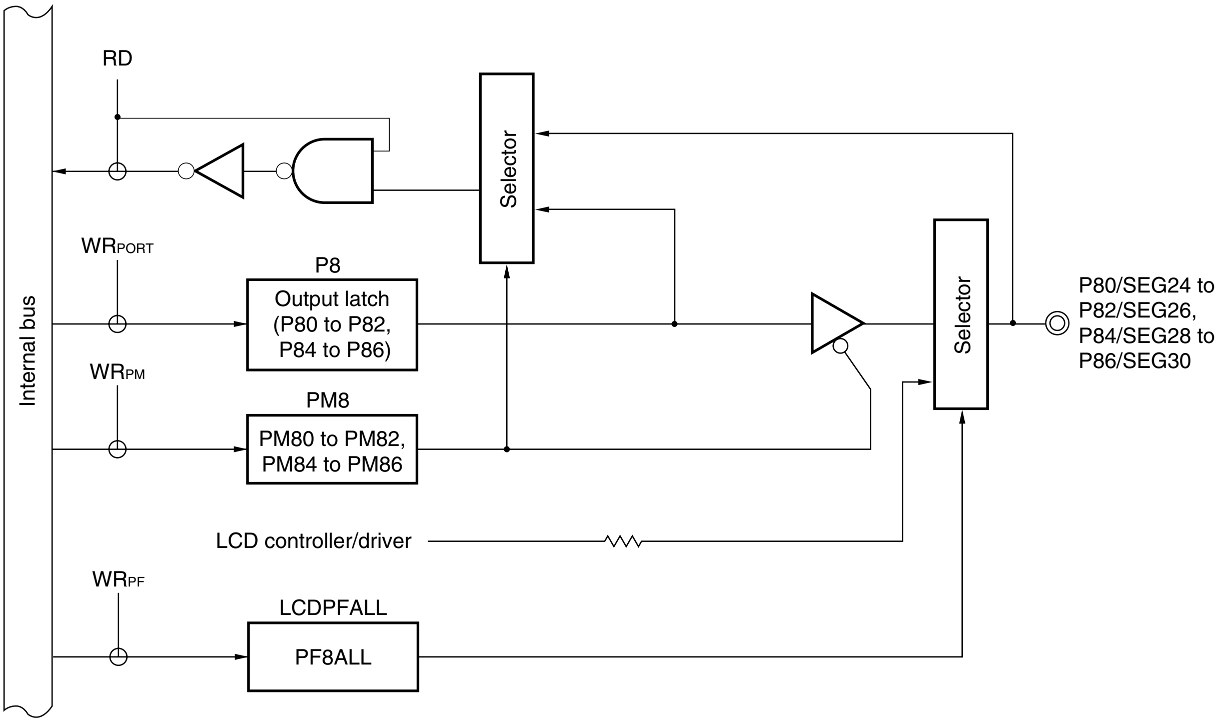
(a) μ PD78F0836, 78F0837, 78F0842, 78F0843, 78F0844, 78F0845, 78F0848, and 78F0849


RD
Selector
WRPORT
P8
Output latch (P80 to P82, P84 to P86)
Internal bus
WRPM
PM8
PM80 to PM82, PM84 to PM86
Alternate function
Selector
P80/SM11 to P82/SM13, P84/SM21 to P86/SM23

P8:	Port register 8
PM8:	Port mode register 8
RD:	Read signal
WR××:	Write signal

Figure 4-19. Block Diagram of P80 to P82 and P84 to P86 (2/2)

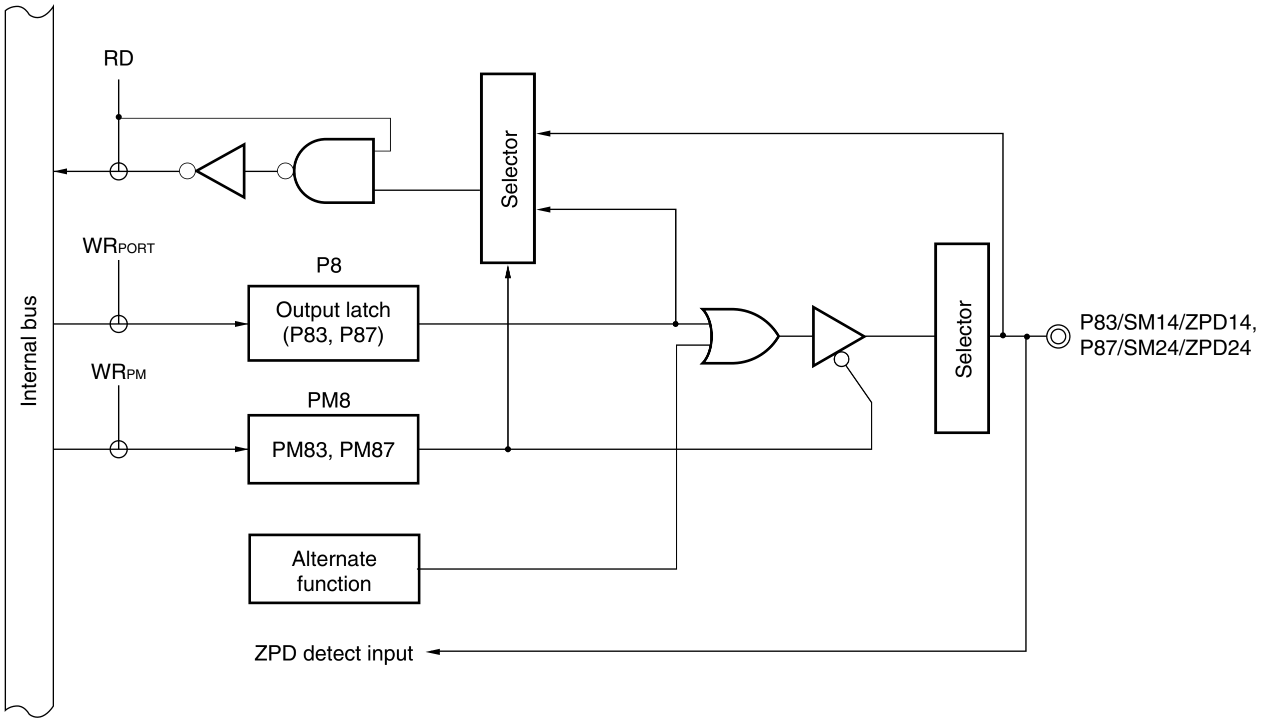
(b) μ PD78F0838, 78F0839, 78F0840, 78F0841, 78F0846, and 78F0847


RD
Selector
WRPORT
P8
Output latch (P80 to P82, P84 to P86)
Internal bus
WRPM
PM8
PM80 to PM82, PM84 to PM86
Selector
P80/SEG24 to P82/SEG26, P84/SEG28 to P86/SEG30
LCD controller/driver
WRPF
LCDPFALL
PF8ALL

P8:	Port register 8
PM8:	Port mode register 8
LCDPFALL:	LCD port function register ALL
RD:	Read signal
WR××:	Write signal

Figure 4-20. Block Diagram of P83 and P87 (1/2)

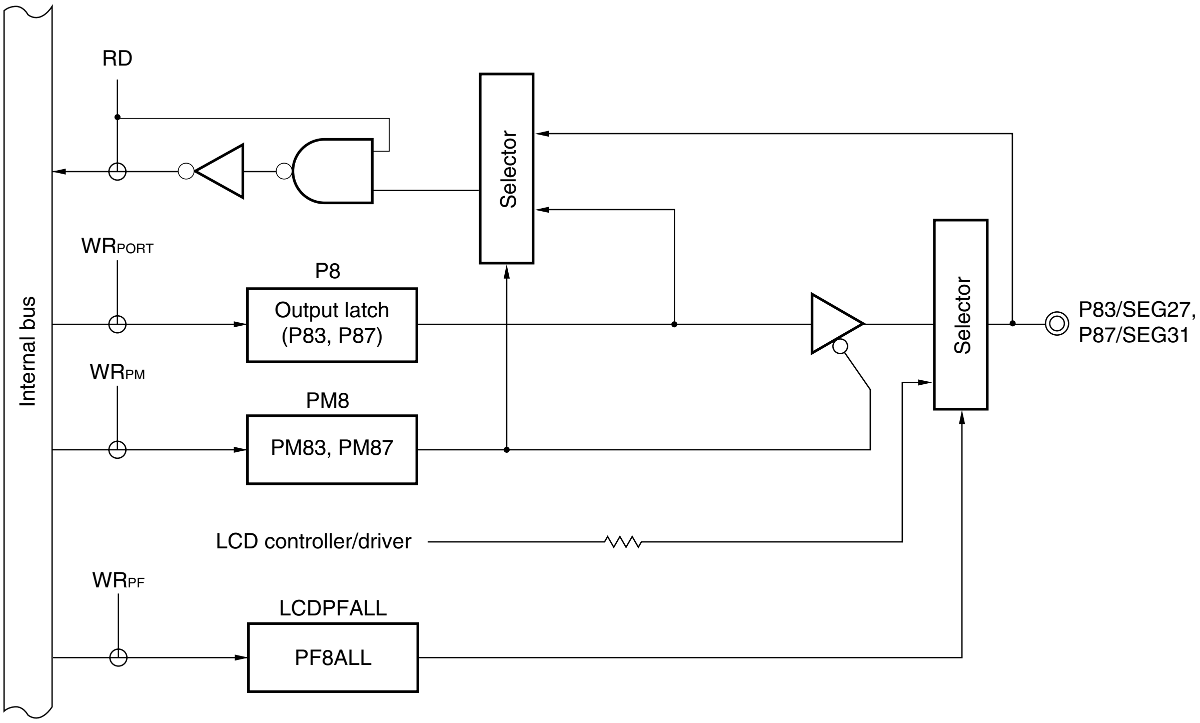
(a) μ PD78F0836, 78F0837, 78F0842, 78F0843, 78F0844, 78F0845, 78F0848, and 78F0849



P8:	Port register 8
PM8:	Port mode register 8
RD:	Read signal
WR××:	Write signal

Figure 4-20. Block Diagram of P83 and P87 (2/2)

(b) μ PD78F0838, 78F0839, 78F0840, 78F0841, 78F0846, and 78F0847



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

4.2.8 Port 9 (78K0/DF2 only)

Port 9 is an 8-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).

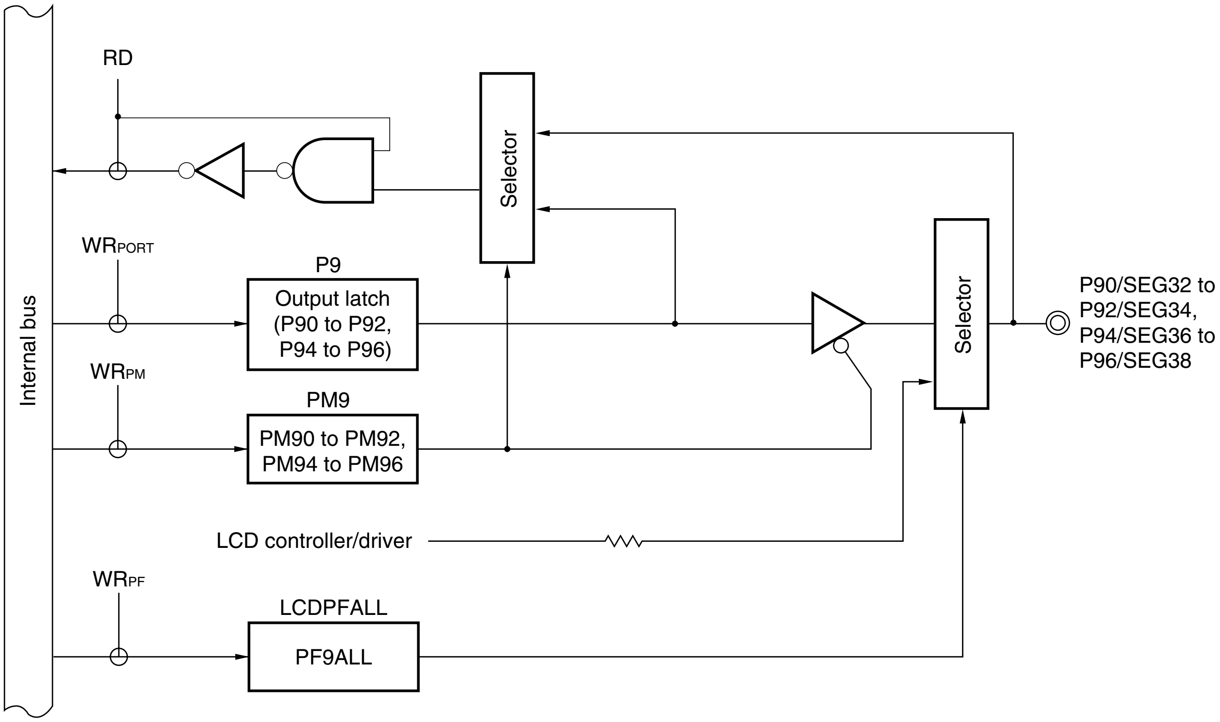
This port can also be used for LCD segment outputs in μ PD78F0838 and 78F0839 or stepper motor controller/driver outputs/inputs in μ PD78F0840, 78F0841, 78F0842, 78F0843, 78F0846, 78F0847, 78F0848, and 78F0849.

Reset signal generation sets port 9 to input mode.

Figures 4-24 and **4-25** shows block diagrams of port 9.

Figure 4-21. Block Diagram of P90 to P92 and P94 to P96 (1/2)

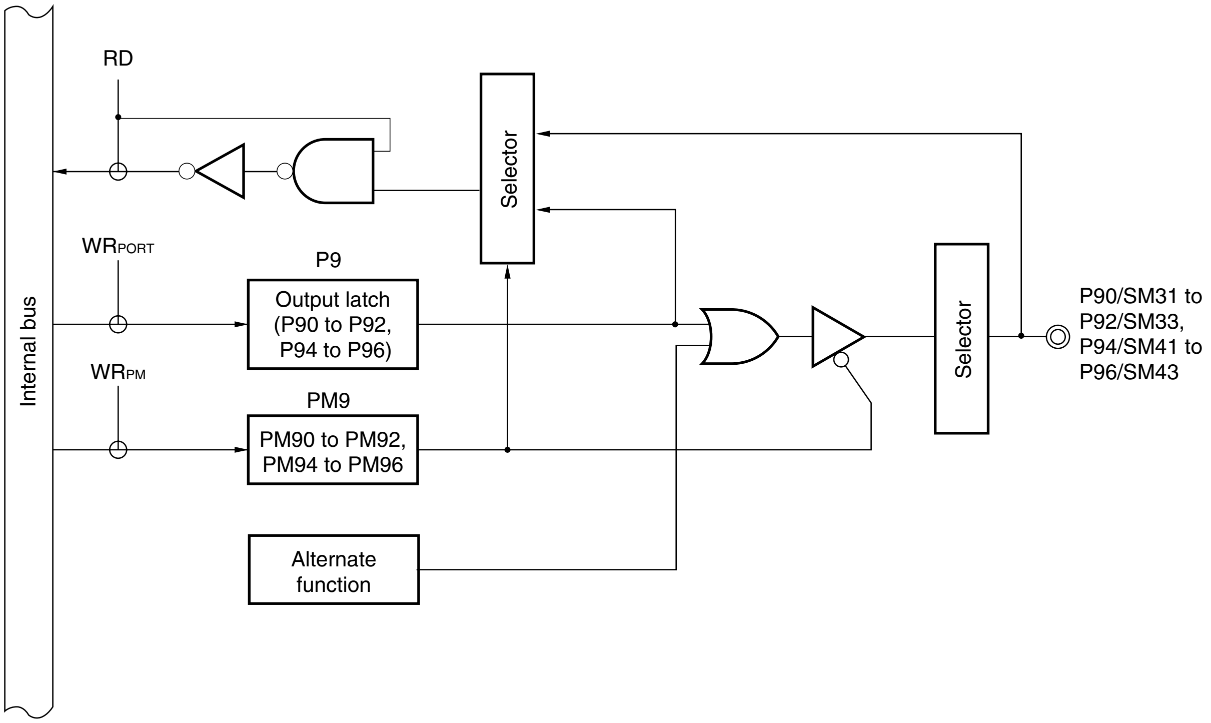
(a) μ PD78F0838 and 78F0839



P9:	Port register 9
PM9:	Port mode register 9
LCDPFALL:	LCD port function register ALL
RD:	Read signal
WR××:	Write signal

Figure 4-21. Block Diagram of P90 to P92 and P94 to P96 (2/2)

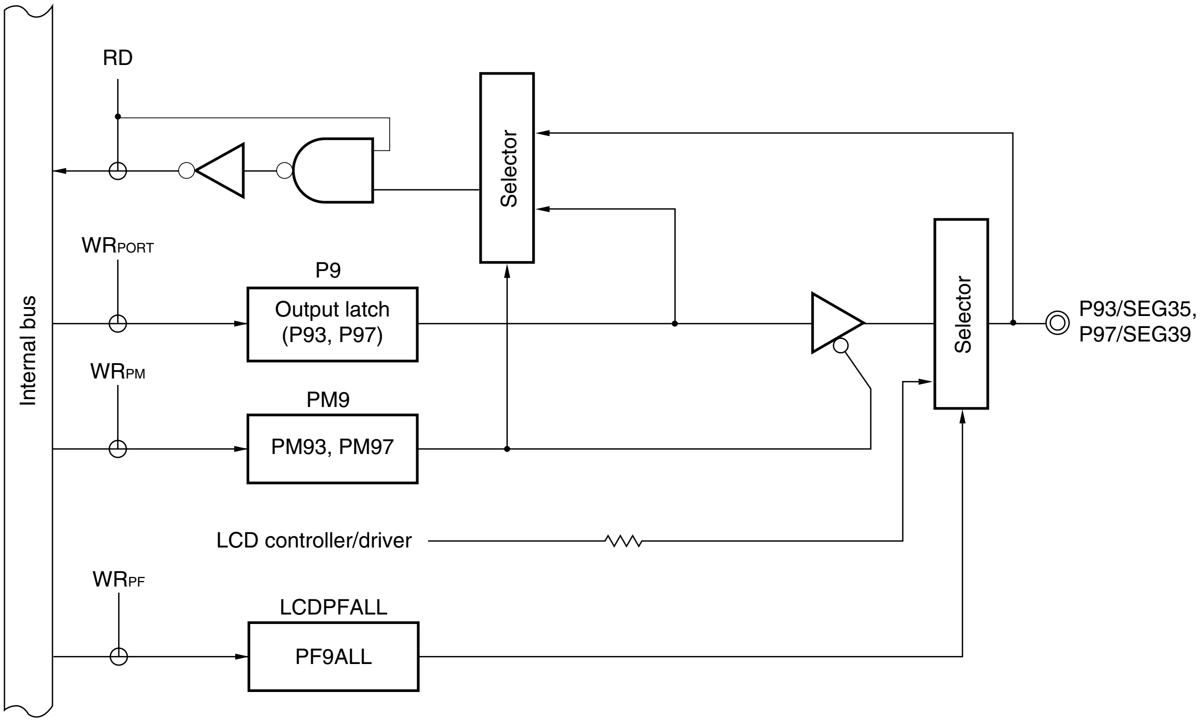
(b) μ PD78F0840, 78F0841, 78F0842, 78F0843, 78F0846, 78F0847, 78F0848, and 78F0849


RD
Internal bus
Selector
WRPORT
P9
Output latch (P90 to P92, P94 to P96)
WRPM
PM9
PM90 to PM92, PM94 to PM96
Alternate function
Selector
P90/SM31 to P92/SM33, P94/SM41 to P96/SM43

P9:	Port register 9
PM9:	Port mode register 9
RD:	Read signal
WR××:	Write signal

Figure 4-22. Block Diagram of P93 and P97 (1/2)

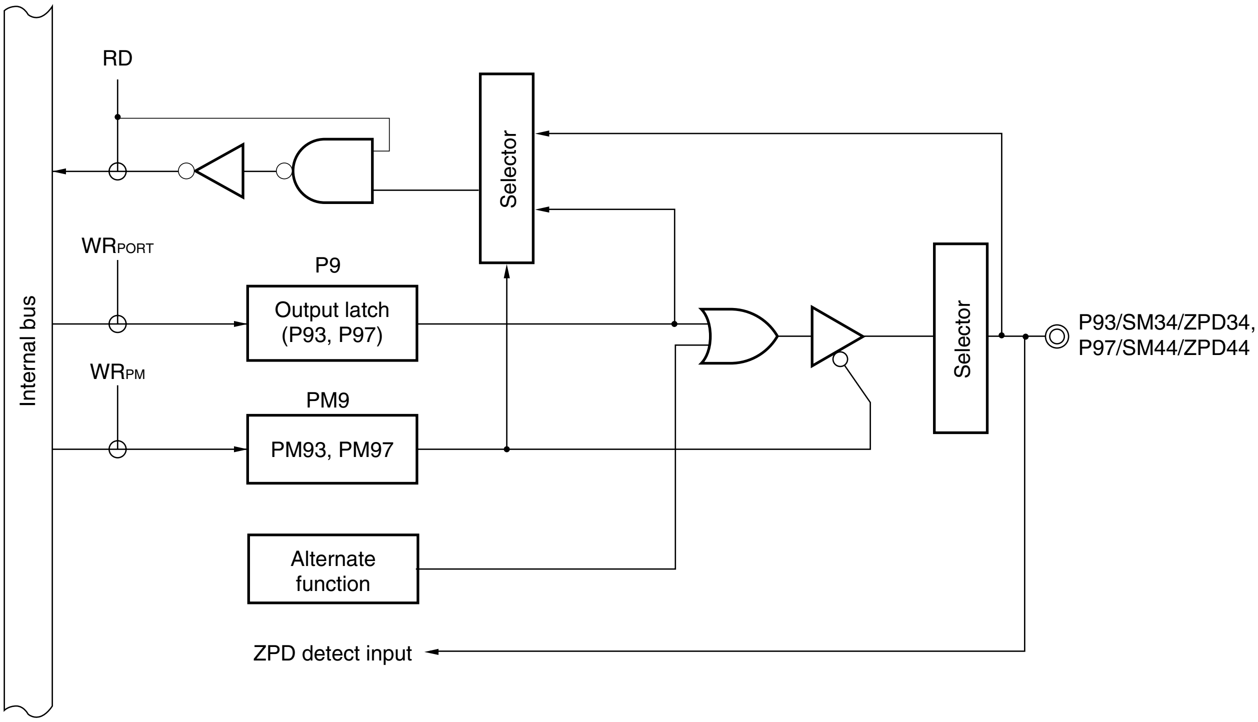
(a) μ PD78F0838 and 78F0839



P9:	Port register 9
PM9:	Port mode register 9
LCDPFALL:	LCD port function register ALL
RD:	Read signal
WR××:	Write signal

Figure 4-22. Block Diagram of P93 and P97 (2/2)

(b) μ PD78F0840, 78F0841, 78F0842, 78F0843, 78F0846, 78F0847, 78F0848, and 78F0849



P9:	Port register 9
PM9:	Port mode register 9
RD:	Read signal
WR××:	Write signal

4.2.9 Port 12

Port 12 is a 5-bit I/O port with an output latch. Port 12 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 for external interrupt input, potential input for external low-voltage detector, connecting resonator for main system clock, connecting resonator for subsystem clock, external clock input for main system clock, external clock input for subsystem clock.

Reset signal generation sets port 12 to input mode.

Figures 4-26 and **4-27** show block diagrams of port 12.

Cautions 1. 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) or subsystem clock (EXCLKS), 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 detail, see 5.3 (5) Clock operation mode select register (OSCCTL)). The reset value of OSCCTL is 00H (all of the P121 to P124 pins are I/O port pins). At this time, setting of the PM121 to PM124 and P121 to P124 pins is not necessary.

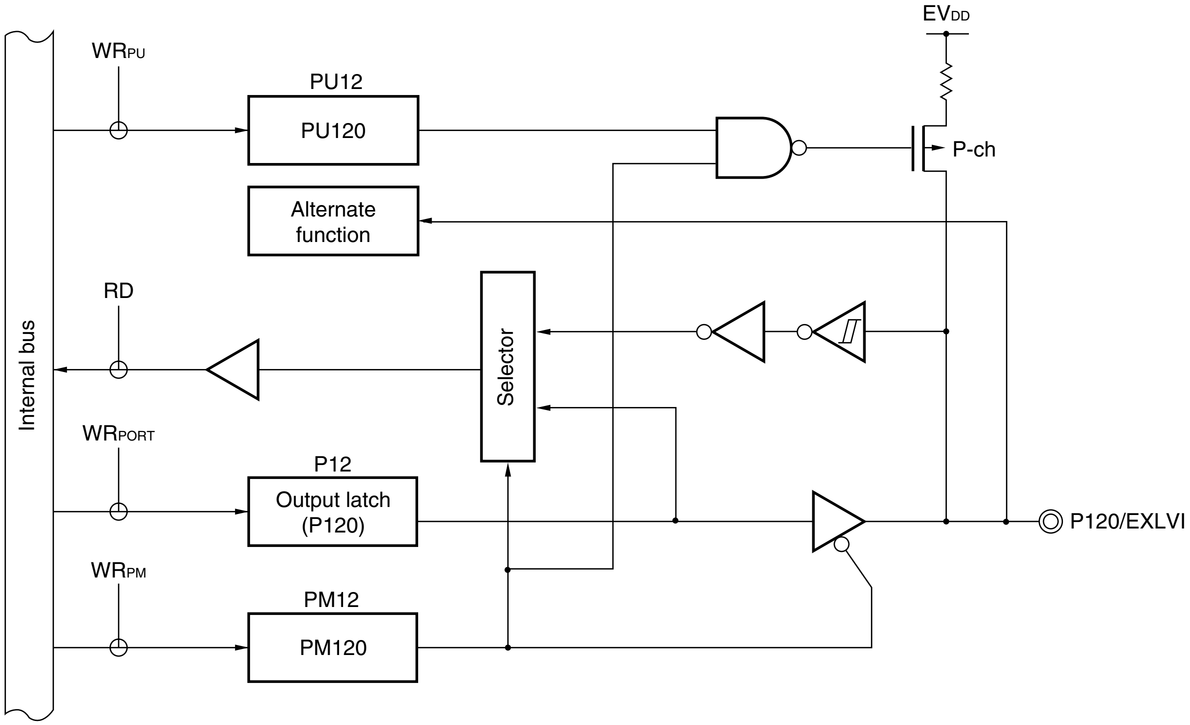
2. Connect P121/X1 as follows when writing the flash memory with a flash programmer.

- P121/X1: When using this pin as a port, connect it to V_{SS} via a resistor (10 kΩ: recommended) (in the input mode) or leave it open (in the output mode).

The above connection is not necessary when writing the flash memory by means of self programming.

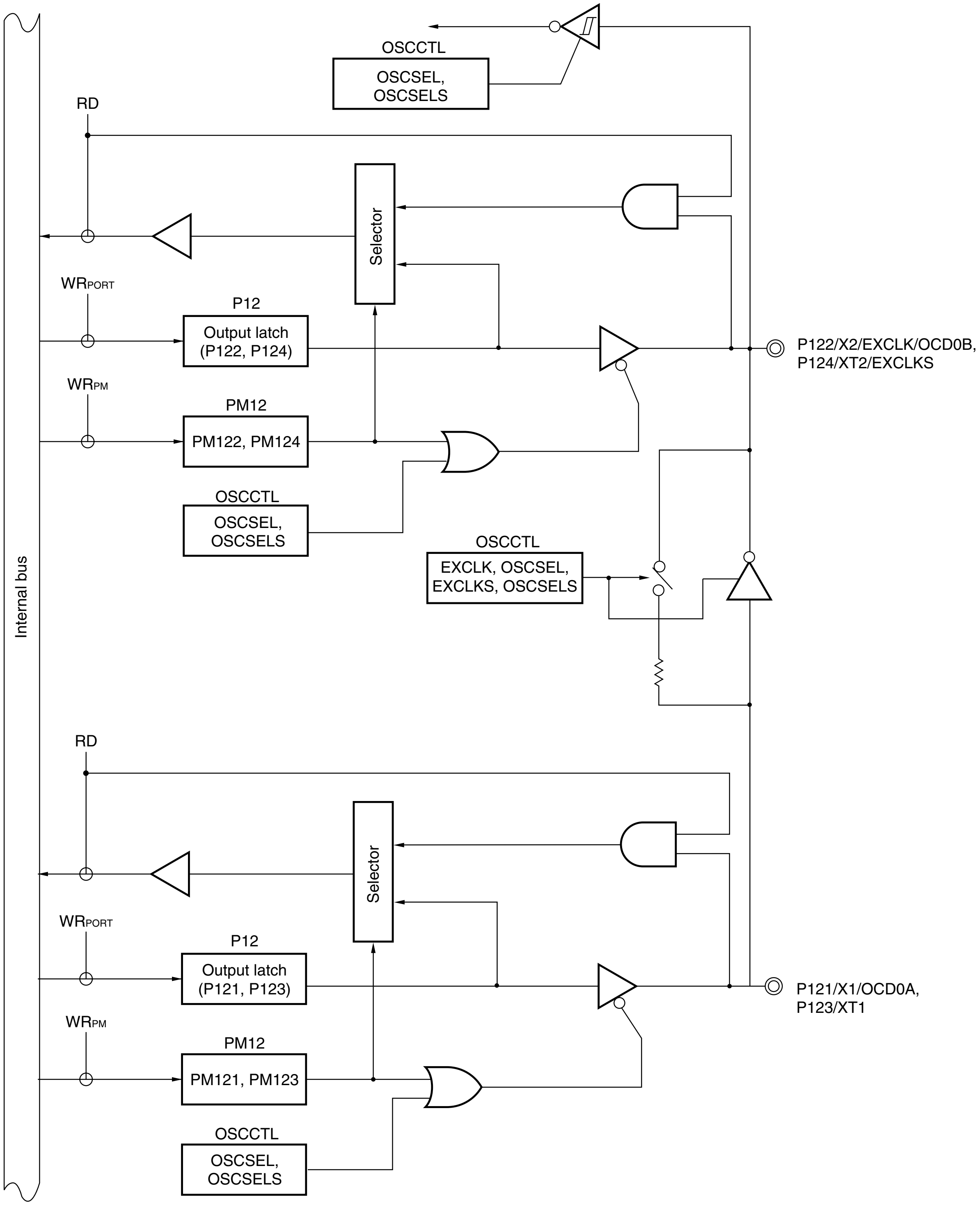
Remark P121/X1 and P122/X2/EXCLK can be used as on-chip debug mode setting pins (OCD0A, OCD0B) when the on-chip debug function is used. For how to connect an in-circuit emulator supporting on-chip debugging (QB-78K0MINI or QB-MINI2), see **CHAPTER 27 ON-CHIP DEBUG FUNCTION**.

Figure 4-23. 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 4-24. Block Diagram of P121 to P124



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

4.3 Registers Controlling Port Function

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

- Port mode registers (PM0 to PM3, PM3 to PM8, PM9[Note], PM12)
- Port registers (P0 to P3, P3 to P8, P9[Note], P12)
- Pull-up resistor option registers (PU0, PU1, PU3, PU6, PU7, PU12)
- Port output mode control register 6 (POM6)
- LCD port function register 0 (LCDPF0)
- LCD port function register 3 (LCDPF3)
- LCD port function register ALL (LCDPFALL)
- Stepper motor port mode control register (SMPC)
- A/D port configuration register (ADPC)

Note 78K0/DF2 only

(1) Port mode registers (PM0 to PM3, PM6 to PM9, PM12)

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 **4.4 Settings of LCDPFALL, LCDPF0, LCDPF3, ISC, Port Mode Register, and Output Latch When Using Alternate Function**.

Figure 4-25. Format of Port Mode Register (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FF20H	FFH	R/W

	7	6	5	4	3	2	1	0			
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W

For 78K0/DE2

	7	6	5	4	3	2	1	0			
PM2	1	1	1	1	PM23	PM22	PM21	PM20	FF22H	FFH	R/W

For 78K0/DF2

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

	7	6	5	4	3	2	1	0			
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FF23H	FFH	R/W

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

For 78K0/DE2

	7	6	5	4	3	2	1	0			
PM7	1	1	1	1	PM73	PM72	PM71	PM70	FF27H	FFH	R/W

For 78K0/DF2

	7	6	5	4	3	2	1	0			
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	FF27H	FFH	R/W

	7	6	5	4	3	2	1	0			
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	PM80	FF28H	FFH	R/W

Figure 4-25. Format of Port Mode Register (2/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM9[Note]	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90	FF29H	FFH	R/W
	7	6	5	4	3	2	1	0			
PM12	1	1	1	PM124	PM123	PM122	PM121	PM120	FF2CH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 3, 6 to 8, 9[Note], 12; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note 78K0/DF2 only

(2) Port registers (P0 to P3, P6 to P9, P12)

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 value of the output latch 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 4-26. Format of Port Register (1/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	P07	P06	P05	P04	P03	P02	P01	P00	FF00H	00H (output latch)	R/W

	7	6	5	4	3	2	1	0			
P1	P17	P16	P15	P14	P13	P12	P11	P10	FF01H	00H (output latch)	R/W

For 78K0/DE2

	7	6	5	4	3	2	1	0			
P2	0	0	0	0	P23	P22	P21	P20	FF02H	00H (output latch)	R/W

For 78K0/DF2

	7	6	5	4	3	2	1	0			
P2	P27	P26	P25	P24	P23	P22	P21	P20	FF02H	00H (output latch)	R/W

	7	6	5	4	3	2	1	0			
P3	P37	P36	P35	P34	P33	P32	P31	P30	FF03H	00H (output latch)	R/W

	7	6	5	4	3	2	1	0			
P6	0	0	0	0	0	0	P61	P60	FF06H	00H (output latch)	R/W

For 78K0/DE2

	7	6	5	4	3	2	1	0			
P7	0	0	0	0	P73	P72	P71	P70	FF07H	00H (output latch)	R/W

For 78K0/DF2

	7	6	5	4	3	2	1	0			
P7	P77	P76	P75	P74	P73	P72	P71	P70	FF07H	00H (output latch)	R/W

	7	6	5	4	3	2	1	0			
P8	P87	P86	P85	P84	P83	P82	P81	P80	FF08H	00H (output latch)	R/W

Figure 4-26. Format of Port Register (2/2)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P9[Note]	P97	P96	P95	P94	P93	P92	P91	P90	FF09H	00H (output latch)	R/W

	7	6	5	4	3	2	1	0			
P12	0	0	0	P124	P123	P122	P121	P120	FF0CH	00H (output latch)	R/W

Remarks 1. An undefined value (pin input level) is read for the value after reset when P0 is read in the input mode. When P2 is read in the output mode, 00H (output latch value) is output.

2. P121 to P124 always indicate 00H (output latch value) in the input mode.

Pmn	m = 0 to 3, 6 to 8, 9[Note], 12; 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

Note 78K0/DF2 only

(3) Pull-up resistor option registers (PU0, PU1, PU3, PU6, PU7, PU12)

These registers specify whether the on-chip pull-up resistors of P00 to P07, P10 to P17, P30 to P37, P60, P61, P70 to P77, and P120 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 PU0, PU1, PU3, PU6, PU7, and PU12. 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 PU0, PU1, PU3, PU6, PU7, and PU12.

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

Reset signal generation clears these registers to 00H.

Figure 4-27. Format of Pull-up Resistor Option Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	PU07	PU06	PU05	PU04	PU03	PU02	PU01	PU00	FF30H	00H	R/W

	7	6	5	4	3	2	1	0			
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	FF31H	00H	R/W

	7	6	5	4	3	2	1	0			
PU3	PU37	PU36	PU35	PU34	PU33	PU32	PU31	PU30	FF33H	00H	R/W

	7	6	5	4	3	2	1	0			
PU6	0	0	0	0	0	0	PU61	PU60	FF36H	00H	R/W

For 78K0/DE2

	7	6	5	4	3	2	1	0			
PU7	0	0	0	0	PU73	PU72	PU71	PU70	FF37H	00H	R/W

For 78K0/DF2

	7	6	5	4	3	2	1	0			
PU7	PU77	PU76	PU75	PU74	PU73	PU72	PU71	PU70	FF37H	00H	R/W

	7	6	5	4	3	2	1	0			
PU12	0	0	0	0	0	0	0	PU120	FF3CH	00H	R/W

PUmn	PUmn pin on-chip pull-up resistor selection (m = 0, 1, 3, 6, 7, 12; n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

(4) Port output mode control register 6 (POM6)

This register sets the output mode of P60 and P61 in 1-bit units. During I^{2}C communication, set P60/SCL0/INTP1 and P61/SDA0/INTP3 to N-ch open-drain output (5 V-torelant) mode.

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

Reset signal generation clears this register to 00H.

Figure 4-28. Format of Port Output Mode Control Register 6 (POM6)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM6	0	0	0	0	0	0	POM61	POM60	FF0EH	00H	R/W

POM6n	P6n pin output selection (n = 0, 1)
0	CMOS output
1	N-ch open-drain output (5 V-torelant)

(5) LCD port function register 0 (LCDPF0)

This register sets whether to use pins P00 to P07 as port pins (other than segment output pins) or segment output pins.

LCDPF0 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets LCDPF0 to 00H.

Figure 4-29. Format of LCD Port Function Register 0 (LCDPF0)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
LCDPF0	PF07	PF06	PF05	PF04	PF03	PF02	PF01	PF00	FF1BH	00H	R/W

PF0n	P0n pin port/segment output specification (n = 0 to 7)
0	Used as port (other than segment output)
1	Used as segment output

(6) LCD port function register 3 (LCDPF3)

This register sets whether to use pins P30 to P37 as port pins (other than segment output pins) or segment output pins.

LCDPF3 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets LCDPF3 to 00H.

Figure 4-30. Format of LCD Port Function Register 3 (LCDPF3)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
LCDPF3	PF37	PF36	PF35	PF34	PF33	PF32	PF31	PF30	FF0DH	00H	R/W

PF3n	P3n pin port/segment output specification (n = 0 to 7)
0	Used as port (other than segment output)
1	Used as segment output

(7) LCD port function register ALL (LCDPFALL)

This register sets whether to use pins P13 to P16, P74 to P77, P80 to P87, and P90 to P97 as port pins (other than segment output pins) or segment output pins.

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

Reset signal generation sets LCDPFALL to 00H.

Figure 4-31. Format of LCD Port Function Register ALL (LCDPFALL)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
LCDPFALL	PF7UPNIB	PF16	PF15	PF14	PF13	PF9ALL	PF8ALL	0	FF1AH	00H	R/W

PF1n	P1n pin port/segment output specification (n = 3 to 6)
0	Used as port (other than segment output)
1	Used as segment output

PF7UPNIB	P7n pin port/segment output specification (n = 4 to 7)
0	Used as port (other than segment output)
1	Used as segment output

PFnALL	Pnm pin port/segment output specification (n = 8, 9; m = 0 to 7)
0	Used as port (other than segment output)
1	Used as segment output

(8) Stepper motor port mode control register (SMPC)

This register sets the output mode of stepper motor controller/driver.

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

Reset signal generation clears this register to 00H.

Figure 4-32. Format of Stepper Motor Port Mode Control Register (SMPC)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
SMPC	MOD4	MOD3	MOD2	MOD1	EN4	EN3	EN2	EN1	FF3DH	00H	R/W

ENn	MODn	Port mode selection (n = 0 to 4)
0	–	Port mode All SMnm (n = 1 to 4, m = 1 to 4) are set to port function.
1	0	PWM full bridge mode SMnm (n = 1 to 4, m = 1 to 4) are set to full bridge output control mode.
1	1	PWM half bridge mode Depending on the DIRnk (n = 1 to 4, k = 0, 1) bit of the compare control registers (MCMPCn; n = 1 to 4), SMnm (n = 1 to 4, m = 1 to 4) are set to PWM output control mode or port mode.

<R>

An example of settings when n = 1 is as follows:

EN1	MOD1	DIR11	DIR10	PWM Output Pin Control				Output Mode
				SM11 (sin+)	SM12 (sin–)	SM13 (cos+)	SM14 (cos–)	
0	–	–	–	port	port	port	port	Port mode
1	0	0	0	PWM	0	PWM	0	PWM full bridge mode
1	0	0	1	PWM	0	0	PWM	
1	0	1	0	0	PWM	0	PWM	
1	0	1	1	0	PWM	PWM	0	
1	1	0	0	PWM	port	PWM	port	PWM half bridge mode
1	1	0	1	PWM	port	port	PWM	
1	1	1	0	port	PWM	port	PWM	
1	1	1	1	port	PWM	PWM	port	

Caution Set port registers (Pn) and port mode registers (PMn) whose pins are not in the PWM mode to 00H in the PWM full bridge mode.

(9) A/D port configuration register (ADPC)

This register switches the P20/ANI0 to P27/ANI7 (P20/ANI0 to P23/ANI3 in 78K0/DE2, P20/ANI0 to P27/ANI7 in 78K0/DF2) pins to analog input of A/D converter or digital I/O of port.

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

Reset signal generation clears this register to 00H.

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

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ADPC	0	0	0	0	ADPC3	ADPC2	ADPC1	ADPC0	FF22H	00H	R/W

ADPC3	ADPC2	ADPC1	ADPC0	Analog input (A)/ digital input (D) switching							
				P27/ANI7	P26/ANI6	P25/ANI5	P24/ANI4	P23/ANI3	P22/ANI2	P21/ANI1	P20/ANI0
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							

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 ADPC as digital I/O by analog input channel specification register (ADS).

3. If data is written to ADPC, a wait cycle is generated. Do not write data to ADPC when the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped. For details, see CHAPTER 32 CAUTIONS FOR WAIT.

4.4 Settings of LCDPFALL, LCDPF0, LCDPF3, SMPC, ISC, Port Mode Register, and Output Latch When Using Alternate Function

To use the alternate function of a port pin, set the LCDPFALL, LCDPF0, LCDPF3, SMPC, ISC, port mode register, and output latch as shown in the following tables.

The descriptions of the column titles and the value in the tables are as follows:

×:	Don’t care
–:	Don’t apply
LCDPFALL:	LCD port function register ALL
LCDPF0:	LCD port function register 0
LCDPF3:	LCD port function register 3
SMPC:	Stepper motor port mode control register
ISC:	Input switch control register
PM××:	Port mode register
P××:	Port output latch

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

<R>

Table 4-4. Settings of LCDPFALL, LCDPF0, LCDPF3, SMPC, ISC, Port Mode Register, and Output Latch When Using Alternate Function (1/7)

(a) 78K0/DE2

(1/3)

Pin Name	Alternate Function		LCDPFALL	LCDPF0	LCDPF3	SMPC	ISC	PM××	P××
	Function Name	I/O							
P00	SEG12	Output	–	PF00 = 1	–	–	–	×	×
<R>	TIOP40	Input	–	PF00 = 0	–	–	ISC0 = 0	1	×
		Output						0	0
P01	SEG13	Output	–	PF01 = 1	–	–	–	×	×
	TIOP41	Input	–	PF01 = 0	–	–	–	1	×
		Output						0	0
P02	SEG14	Output	–	PF02 = 1	–	–	–	×	×
	TIO50	Input	–	PF02 = 0	–	–	–	1	×
		Output						0	0
P03	SEG15	Output	–	PF03 = 1	–	–	–	×	×
	TIO51	Input	–	PF03 = 0	–	–	–	1	×
		Output						0	0
P04	SEG16	Output	–	PF04 = 1	–	–	–	×	×
	TIOP01	Input	–	PF04 = 0	–	–	–	1	×
		Output						0	0
P05	SEG17	Output	–	PF05 = 1	–	–	–	×	×
	TIOP11	Input	–	PF05 = 0	–	–	–	1	×
		Output						0	0
P06	SEG18	Output	–	PF06 = 1	–	–	–	×	×
<R>	TIOP21	Input	–	PF06 = 0	–	–	ISC3 = 0	1	×
		Output						0	0
P07	SEG19	Output	–	PF07 = 1	–	–	–	×	×
	TIOP31	Input	–	PF07 = 0	–	–	–	1	×
		Output						0	0
P10	INTP4	Input	–	–	–	–	–	1	×
	$\overline{\text{SCK10}}$	Input	–	–	–	–	–	1	×
		Output	–	–	–			0	1
P11	SI10	Input	–	–	–	–	–	1	×
P12	INTP2	Input	–	–	–	–	–	1	×
	SO10	Output	–	–	–	–	–	0	0
P13	SEG23	Output	PF13 = 1	–	–	–	–	×	×
	TIOP30	Input	PF13 = 0	–	–	–	ISC6 = 0	1	×
		Output						0	0
	TxD60	Output	PF13 = 0	–	–	–	ISC7 = 0	0	1

Table 4-4. Settings of LCDPFALL, LCDPF0, LCDPF3, SMPC, ISC, Port Mode Register, and Output Latch When Using Alternate Function (2/7)

(a) 78K0/DE2

(2/3)

<R>

Pin Name	Alternate Function		LCDPFALL	LCDPF0	LCDPF3	SMPC	ISC	PM××	P××
	Function Name	I/O							
P14	INTPR60	Input	PF14 = 0	–	–	–	ISC7 = 0	1	×
	RxD60	Input	PF14 = 0	–	–	–		1	×
	SEG22	Output	PF14 = 1	–	–	–	–	×	×
	TIOP20	Input	PF14 = 0	–	–	–	ISC1 = 0	1	×
		Output						0	0
P15	SEG21	Output	PF15 = 1	–	–	–	–	×	×
	TIOP10	Input	PF15 = 0	–	–	–	–	1	×
		Output						0	0
P16	SEG20	Output	PF16 = 1	–	–	–	–	×	×
	TIOP00	Input	PF16 = 0	–	–	–	–	1	×
		Output						0	0
P17	INTP0	Input	–	–	–	–	–	1	×
	<TIOP30>	Input	–	–	–	–	ISC6 = 1	1	×
		Output						0	0
P20 to P23	ANI0 to ANI3	Input	–	–	–	–	–	1	×
P30	SEG4	Output	–	–	PF30 = 1	–	–	×	×
P31	SEG5	Output	–	–	PF31 = 1	–	–	×	×
	OCD1A	I/O	–	–	PF31 = 0	–	–	1	×
P32	SEG6	Output	–	–	PF32 = 1	–	–	×	×
	OCD1B	I/O	–	–	PF32 = 0	–	–	1	×
P33	SEG7	Output	–	–	PF33 = 1	–	–	×	×
P34	SEG8	Output	–	–	PF34 = 1	–	–	×	×
P35	SEG9	Output	–	–	PF35 = 1	–	–	×	×
P36	SEG10	Output	–	–	PF36 = 1	–	–	×	×
P37	SEG11	Output	–	–	PF37 = 1	–	–	×	×
P60	INTP1	Input	–	–	–	–	–	1	×
	SCL0	I/O	–	–	–	–	–	0	0
P61	INTP3	Input	–	–	–	–	–	1	×
	SDA0	I/O	–	–	–	–	–	0	0
P70	CRxD[Note]	Input	–	–	–	–	–	1	×
	<RxD60/INTPR60>	Input	–	–	–	–	ISC7 = 1	1	×
P71	CTxD[Note]	Output	–	–	–	–	–	0	1
	<TxD60>	Output	–	–	–	–	ISC7 = 1	0	1

Note μPD78F0844 and 78F0845 only.

Table 4-4. Settings of LCDPFALL, LCDPF0, LCDPF3, SMPC, ISC, Port Mode Register, and Output Latch When Using Alternate Function (3/7)

(a) 78K0/DE2

(3/3)

Pin Name	Alternate Function		LCDPFALL	LCDPF0	LCDPF3	SMPC	ISC	PM××	P××
	Function Name	I/O							
P72	PCL	Output	–	–	–	–	–	0	0
	SGOA	Output	–	–	–	–	–	0	0
P73	BUZ	Output	–	–	–	–	–	0	0
	SGO	Output	–	–	–	–	–	0	0
	SGOF	Output	–	–	–	–	–	0	0
P80 to P82	SM11 to SM13	Output	–	–	–	EN1 = 1	–	0	0
P83	SM14	Output	–	–	–	EN1 = 1	–	0	0
	ZPD14	Input	–	–	–	–	–	1	×
P84 to P86	SM21 to SM23	Output	–	–	–	EN2 = 1	–	0	0
P87	SM24	Output	–	–	–	EN2 = 1	–	0	0
	ZPD24	Input	–	–	–	–	–	1	×
P120	EXLVI	Input	–	–	–	–	–	1	×
P121	X1	Input	–	–	–	–	–	1	×
	OCD0A	I/O	–	–	–	–	–	1	×
P122	X2	Input	–	–	–	–	–	1	×
	EXCLK	Input	–	–	–	–	–	1	×
	OCD0B	I/O	–	–	–	–	–	1	×
P123	XT1	Input	–	–	–	–	–	1	×
P124	XT2	Input	–	–	–	–	–	1	×
	EXCLKS	Input	–	–	–	–	–	1	×

Table 4-4. Settings of LCDPFALL, LCDPF0, LCDPF3, SMPC, ISC, Port Mode Register, and Output Latch When Using Alternate Function (4/7)

(b) 78K0/DF2

(1/4)

	Pin Name	Alternate Function		LCDPFALL	LCDPF0	LCDPF3	SMPC	ISC	PM××	P××
		Function Name	I/O							
	P00	SEG12	Output	–	PF00 = 1	–	–	–	×	×
<R>		TIOP40	Input	–	PF00 = 0	–	–	ISC0 = 0	1	×
			Output						0	0
	P01	SEG13	Output	–	PF01 = 1	–	–	–	×	×
		TIOP41	Input	–	PF01 = 0	–	–	–	1	×
			Output						0	0
	P02	SEG14	Output	–	PF02 = 1	–	–	–	×	×
		TIO50	Input	–	PF02 = 0	–	–	–	1	×
			Output						0	0
	P03	SEG15	Output	–	PF03 = 1	–	–	–	×	×
		TIO51	Input	–	PF03 = 0	–	–	–	1	×
			Output						0	0
	P04	SEG16	Output	–	PF04 = 1	–	–	–	×	×
		TIOP01	Input	–	PF04 = 0	–	–	–	1	×
			Output						0	0
	P05	SEG17	Output	–	PF05 = 1	–	–	–	×	×
		TIOP11	Input	–	PF05 = 0	–	–	–	1	×
			Output						0	0
	P06	SEG18	Output	–	PF06 = 1	–	–	–	×	×
<R>		TIOP21	Input	–	PF06 = 0	–	–	ISC3 = 0	1	×
			Output						0	0
	P07	SEG19	Output	–	PF07 = 1	–	–	–	×	×
		TIOP31	Input	–	PF07 = 0	–	–	–	1	×
			Output						0	0
	P10	INTP4	Input	–	–	–	–	–	1	×
		$\overline{\text{SCK10}}$	Input	–	–	–	–	–	1	×
			Output	–	–	–	–	–	0	1
		TxD61	Output	–	–	–	–	–	0	1
	P11	INTPR61	Input	–	–	–	–	–	1	×
		SI10	Input	–	–	–	–	–	1	×
		RxD61	Input	–	–	–	–	–	1	×
	P12	INTP2	Input	–	–	–	–	–	1	×
		SO10	Output	–	–	–	–	–	0	0
	P13	SEG23	Output	PF13 = 1	–	–	–	–	×	×
<R>		TIOP30	Input	PF13 = 0	–	–	–	ISC2 = 0 & ISC6 = 0	1	×
			Output						0	0
		TxD60	Output	PF13 = 0	–	–	–	ISC7 = 0	0	1

Table 4-4. Settings of LCDPFALL, LCDPF0, LCDPF3, SMPC, ISC, Port Mode Register, and Output Latch When Using Alternate Function (5/7)

(b) 78K0/DF2

(2/4)

	Pin Name	Alternate Function		LCDPFALL	LCDPF0	LCDPF3	SMPC	ISC	PM××	P××
		Function Name	I/O							
	P14	INTPR60	Input	PF14 = 0	–	–	–	ISC7 = 0	1	×
		RxD60	Input	PF14 = 0	–	–	–		1	×
		SEG22	Output	PF14 = 1	–	–	–	–	×	×
<R>		TIOP20	Input	PF14 = 0	–	–	–	ISC1 = 0 & ISC5 = 0	1	×
			Output						0	0
	P15	SEG21	Output	PF15 = 1	–	–	–	–	×	×
		TIOP10	Input	PF15 = 0	–	–	–	–	1	×
			Output						0	0
	P16	SEG20	Output	PF16 = 1	–	–	–	–	×	×
		TIOP00	Input	PF16 = 0	–	–	–	–	1	×
			Output						0	0
	P17	INTP0	Input	–	–	–	–	–	1	×
<R>		<TIOP30>	Input	–	–	–	–	ISC2 = 0 & ISC6 = 1	1	×
			Output						0	0
	P20 to P27	ANI0 to ANI7	Input	–	–	–	–	–	1	×
	P30	SEG4	Output	–	–	PF30 = 1	–	–	×	×
	P31	SEG5	Output	–	–	PF31 = 1	–	–	×	×
		OCD1A	I/O	–	–	PF31 = 0	–	–	1	×
	P32	SEG6	Output	–	–	PF32 = 1	–	–	×	×
		OCD1B	I/O	–	–	PF32 = 0	–	–	1	×
	P33 to P37	SEG7 to SEG11	Output	–	–	PF33 to PF37 = 1	–	–	×	×
	P60	INTP1	Input	–	–	–	–	–	1	×
		SCL0	I/O	–	–	–	–	–	0	0
	P61	INTP3	Input	–	–	–	–	–	1	×
		SDA0	I/O	–	–	–	–	–	0	0
	P70	CRxD[Note]	Input	–	–	–	–	–	1	×
		<RxD60/INTPR60>	Input	–	–	–	–	ISC7 = 1	1	×
	P71	CTxD[Note]	Output	–	–	–	–	–	0	1
		<TxD60>	Output	–	–	–	–	ISC7 = 1	0	1
	P72	PCL	Output	–	–	–	–	–	0	0
		SGOA	Output	–	–	–	–	–	0	0
	P73	BUZ	Output	–	–	–	–	–	0	0
		SGO	Output	–	–	–	–	–	0	0
		SGOF	Output	–	–	–	–	–	0	0

Note μ PD78F0846, 78F0847, 78F0848, and 78F0849 only.

Table 4-4. Settings of LCDPFALL, LCDPF0, LCDPF3, SMPC, ISC, Port Mode Register, and Output Latch When Using Alternate Function (6/7)

(b) 78K0/DF2

(3/4)

Pin Name	Alternate Function		LCDPFALL	LCDPF0	LCDPF3	SMPC	ISC	PM××	P××
	Function Name	I/O							
P74	SCK11	Input	PF7UPNIB = 0	–	–	–	–	1	×
		Output	PF7UPNIB = 0	–	–	–	–	0	1
	SEG24$^{\text{Note1}}$	Output	PF7UPNIB = 1	–	–	–	–	×	×
P75	SI11	Input	PF7UPNIB = 0	–	–	–	–	1	×
	SEG25$^{\text{Note1}}$	Output	PF7UPNIB = 1	–	–	–	–	×	×
P76	SO11	Output	PF7UPNIB = 0	–	–	–	–	0	0
	SEG26$^{\text{Note1}}$	Output	PF7UPNIB = 1	–	–	–	–	×	×
P77	SEG27$^{\text{Note1}}$	Output	PF7UPNIB = 1	–	–	–	–	×	×
	SSI11	Input	PF7UPNIB = 0	–	–	–	–	1	×
	<TIOP20>	Input	PF7UPNIB = 0	–	–	–	ISC5 = 1	1	×
		Output						0	0
P80 to P82	SEG24 to SEG26$^{\text{Note2}}$	Output	PF8ALL = 1	–	–	–	–	×	×
	SM11 to SM13$^{\text{Note3}}$	Output	PF8ALL = 0	–	–	EN1 = 1	–	0	0
P83	SEG27$^{\text{Note2}}$	Output	PF8ALL = 1	–	–	–	–	×	×
	SM14$^{\text{Note3}}$	Output	PF8ALL = 0	–	–	EN1 = 1	–	0	0
	ZPD14$^{\text{Note3}}$	Input	PF8ALL = 0	–	–	–	–	1	×
P84 to P86	SEG28 to SEG30$^{\text{Note2}}$	Output	PF8ALL = 1	–	–	–	–	×	×
	SM21 to SM23$^{\text{Note3}}$	Output	PF8ALL = 0	–	–	EN2 = 1	–	0	0
P87	SEG31$^{\text{Note2}}$	Output	PF8ALL = 1	–	–	–	–	×	×
	SM24$^{\text{Note3}}$	Output	PF8ALL = 0	–	–	EN2 = 1	–	0	0
	ZPD24$^{\text{Note3}}$	Input	PF8ALL = 0	–	–	–	–	1	×
P90 to P92	SEG32 to SEG34$^{\text{Note4}}$	Output	PF9ALL = 1	–	–	–	–	×	×
	SM31 to SM33$^{\text{Note5}}$	Output	PF9ALL = 0	–	–	EN3 = 1	–	0	0
P93	SEG35$^{\text{Note4}}$	Output	PF9ALL = 1	–	–	–	–	×	×
	SM34$^{\text{Note5}}$	Output	PF9ALL = 0	–	–	EN3 = 1	–	0	0
	ZPD34$^{\text{Note5}}$	Input	PF9ALL = 0	–	–	–	–	1	×

Notes 1. μPD78F0842, 78F0843, 78F0848, and 78F0849 only.

2. For μPD78F0838, 78F0839, 78F0840, 78F0841, 78F0846, and 78F0847.

3. For μPD78F0842, 78F0843, 78F0848, and 78F0849.

4. For μPD78F0838 and 78F0839.

5. For μPD78F0840, 78F0841, 78F0842, 78F0843, 78F0846, 78F0847, 78F0848, and 78F0849.

Table 4-4. Settings of LCDPFALL, LCDPF0, LCDPF3, SMPC, ISC, Port Mode Register, and Output Latch When Using Alternate Function (7/7)

(b) 78K0/DF2

(4/4)

Pin Name	Alternate Function		LCDPFALL	LCDPF0	LCDPF3	SMPC	ISC	PM××	P××
	Function Name	I/O							
P94 to P96	SEG36 to SEG38[Note1]	Output	PF9ALL = 1	–	–	–	–	×	×
	SM41 to SM43[Note2]	Output	PF9ALL = 0	–	–	EN4 = 1	–	0	0
P97	SEG39[Note1]	Output	PF9ALL = 1	–	–	–	–	×	×
	SM44[Note2]	Output	PF9ALL = 0	–	–	EN4 = 1	–	0	0
	ZPD44[Note2]	Input	PF9ALL = 0	–	–	–	–	1	×
P120	EXLVI	Input	–	–	–	–	–	1	×
P121	X1	Input	–	–	–	–	–	1	×
	OCD0A	I/O	–	–	–	–	–	1	×
P122	X2	Input	–	–	–	–	–	1	×
	EXCLK	Input	–	–	–	–	–	1	×
	OCD0B	I/O	–	–	–	–	–	1	×
P123	XT1	Input	–	–	–	–	–	1	×
P124	XT2	Input	–	–	–	–	–	1	×
	EXCLKS	Input	–	–	–	–	–	1	×

Notes 1. For μ PD78F0838 and 78F0839.

2. For μ PD78F0840, 78F0841, 78F0842, 78F0843, 78F0846, 78F0847, 78F0848, and 78F0849.

Remarks 1. The function of the ANI0/P20 to ANI7/P27 pins can be selected by using the A/D port configuration register (ADPC), the analog input channel specification register (ADS), and PM2.

ADPC	PM2	ADS	P20/ANI0 to P27/ANI7 Pin
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

2. 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) or subsystem clock (EXCLKS), 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 detail, see **5.3 (5) Clock operation mode select register (OSCCTL)**). The reset value of OSCCTL is 00H (all of the P121 to P124 pins are I/O port pins). At this time, setting of the PM121 to PM124 and P121 to P124 pins is not necessary.

3. P31/SEG5, P32/SEG6, P121/X1, and P122/X2/EXCLK can be used as on-chip debug mode setting pins (OCD1A, OCD1B, OCD0A, OCD0B) when the on-chip debug function is used. For how to connect an in-circuit emulator supporting on-chip debugging (QB-78K0MINI or QB-MINI2), see **CHAPTER 27 ON-CHIP DEBUG FUNCTION**.

4.5 Port Function Operations

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

4.5.1 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 by reset.

(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.

4.5.2 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.

4.5.3 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 by reset.

(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.

4.6 Cautions on 1-bit Manipulation Instruction for Port Register n (Pn)

When a 1-bit manipulation instruction is executed on a port that provides both input and output functions, the output latch value of an input port that is not subject to manipulation may be written in addition to the targeted bit.

Therefore, it is recommended to rewrite the output latch when switching a port from input mode to output mode.

<Example> When P10 is an output port, P11 to P17 are input ports (all pin statuses are high level), and the port latch value of port 1 is 00H, if the output of output port P10 is changed from low level to high level via a 1-bit manipulation instruction, the output latch value of port 1 is FFH.

Explanation: The targets of writing to and reading from the Pn register of a port whose PMnm bit is 1 are the output latch and pin status, respectively.

A 1-bit manipulation instruction is executed in the following order in the 78K0/Dx2.

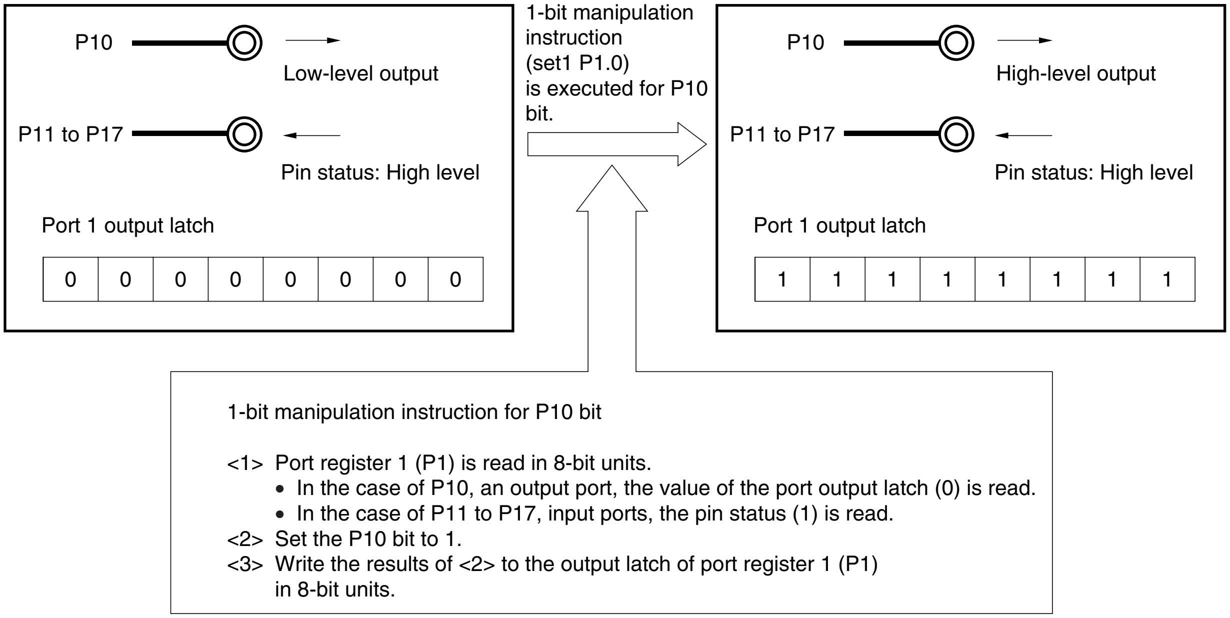
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

In step <1>, the output latch value (0) of P10, which is an output port, is read, while the pin statuses of P11 to P17, which are input ports, are read. If the pin statuses of P11 to P17 are high level at this time, the read value is FEH.

The value is changed to FFH by the manipulation in <2>.

FFH is written to the output latch by the manipulation in <3>.

Figure 4-34. Bit Manipulation Instruction (P10)



1-bit manipulation instruction for P10 bit

<1> Port register 1 (P1) is read in 8-bit units.
- In the case of P10, an output port, the value of the port output latch (0) is read.
- In the case of P11 to P17, input ports, the pin status (1) is read.

<2> Set the P10 bit to 1.
<3> Write the results of <2> to the output latch of port register 1 (P1) in 8-bit units.

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 system clocks and clock oscillators are selectable.

(1) Main system clock

<1> X1 oscillator

This circuit oscillates a clock of f_X = 4 to 20 MHz. 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_{OSC8} = 8 MHz (TYP.). After a $\overline{\text{RESET}}$ 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 oscillator mode register (RCM).

An external main system clock (f_{EXT} = 4 to 20 MHz) can also be supplied from the EXCLK pin. 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).

An external subsystem clock (f_{EXTS} = 32.768 kHz) can also be supplied from the EXCLKS pin.

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

- **Internal low-speed oscillator**

This circuit oscillates a clock of f_{OSC} = 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 TM50 (when f_{OSC}, $f_{OSC}/2^7$, or $f_{OSC}/2^9$ is selected)

Remarks 1. f_X: X1 clock oscillation frequency
2. f_{OSC8}: Internal high-speed oscillation clock frequency
3. f_{EXT}: External main system clock frequency
4. f_{XT}: XT1 clock oscillation frequency
5. f_{EXTS}: External subsystem clock frequency
6. f_{OSC}: Internal low-speed oscillation clock frequency

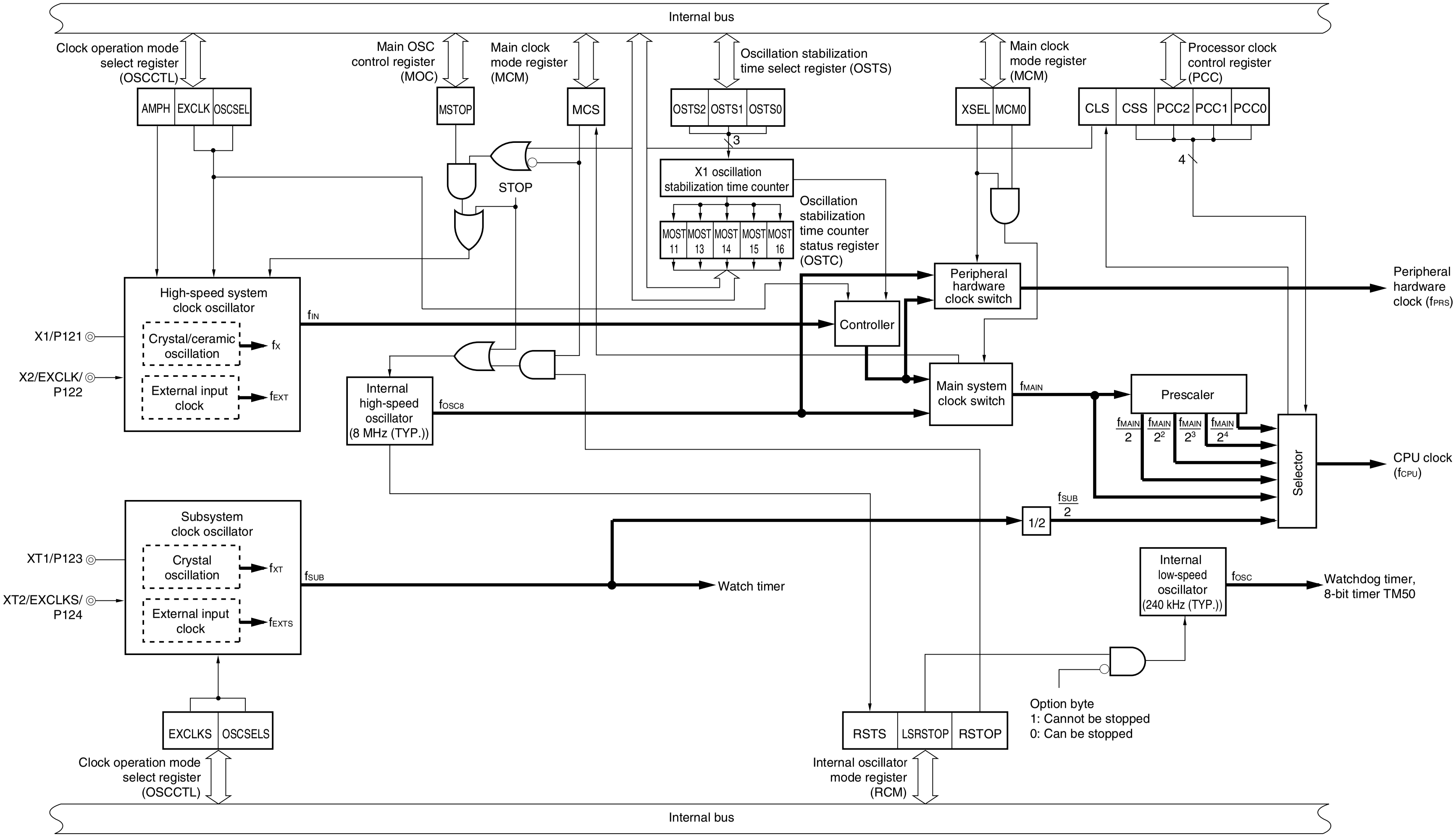
5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

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

Figure 5-1. Block Diagram of Clock Generator



Remark	f_X:	X1 clock oscillation frequency
	f_{OSC8}:	Internal high-speed oscillation clock frequency
	f_{EXT}:	External main system clock frequency
	f_{IN}:	High-speed system clock oscillation frequency
	f_{MAIN}:	Main system clock oscillation frequency
	f_{PRS}:	Peripheral hardware clock frequency
	f_{CPU}:	CPU clock oscillation frequency
	f_{XT}:	XT1 clock oscillation frequency
	f_{EXTS}:	External subsystem clock frequency
	f_{SUB}:	Subsystem clock frequency
	f_{OSC}:	Internal low-speed oscillation clock frequency

5.3 Registers Controlling Clock Generator

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

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

(1) Processor clock control register (PCC)

This register is used to select the CPU clock and the division ratio.

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

Reset signal generation sets PCC to 01H.

Figure 5-2. 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[Note 2]	PCC2	PCC1	PCC0	CPU clock (f_{CPU}) selection
0	0	0	0	f_{MAIN}
	0	0	1	$f_{MAIN}/2$ (default)
	0	1	0	$f_{MAIN}/2^2$
	0	1	1	$f_{MAIN}/2^3$
	1	0	0	$f_{MAIN}/2^4$
1	0	0	0	$f_{SUB}/2$
	0	0	1	
	0	1	0	
	0	1	1	
	1	0	0	
Other than above				Setting prohibited

Notes 1. Bit 5 is read-only.

2. Be sure to switch CSS from 1 to 0 when bits 1 (MCS) and 0 (MCM0) of the main clock mode register (MCM) are 1.

Caution Be sure to clear bits 3 and 6 to 0.

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

2. f_{SUB}: Subsystem clock frequency

The fastest instruction can be executed in 2 clocks of the CPU clock in the 78K0/Dx2. 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}			
	High-Speed System Clock[Note]		Internal high-speed oscillation clock[Note]	Subsystem Clock
	At 10 MHz Operation	At 20 MHz Operation	At 8 MHz (TYP.) Operation	At 32.768 kHz Operation
f_{MAIN}	0.2 μs	0.1 μs	0.25 μs (TYP.)	–
$f_{MAIN}/2$	0.4 μs	0.2 μs	0.5 μs (TYP.)	–
$f_{MAIN}/2^2$	0.8 μs	0.4 μs	1.0 μs (TYP.)	–
$f_{MAIN}/2^3$	1.6 μs	0.8 μs	2.0 μs (TYP.)	–
$f_{MAIN}/2^4$	3.2 μs	1.6 μs	4.0 μs (TYP.)	–
$f_{SUB}/2$	–		–	122.1 μs

Note The main clock mode register (MCM) is used to set the CPU clock (high-speed system clock/internal high-speed oscillation clock) (see **Figure 5-4**).

(2) Internal oscillator 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 5-3. Format of Internal Oscillator 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 oscillation
0	Waiting for stabilization of internal high-speed oscillator oscillation in high-accuracy mode (internal high-speed oscillator operation in low-accuracy mode)
1	Internal high-speed oscillator operation in high-accuracy mode

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 oscillation 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 RSTOP to 1 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)**

(3) 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 5-4. 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_{MAIN})	Peripheral hardware clock (f_{PRS})
0	0	Internal high-speed oscillation clock (f_{OSC8})	Internal high-speed oscillation clock (f_{OSC8})
0	1		
1	0		High-speed system clock (f_{IN})
1	1	High-speed system clock (f_{IN})	

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. 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.

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

- **Watchdog timer**
- **When "f_{OSC}, $f_{OSC}/2^7$, $f_{OSC}/2^9$" is selected as the count clock for 8-bit timer TM50**
- **Peripheral hardware selects the external clock as the clock source**

4. It takes one clock to change the CPU clock.

(4) 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 MSTOP to 1 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.

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.

(5) Clock operation mode select register (OSCCTL)

This register selects the operation modes of the high-speed system and subsystem clocks.

OSCCTL 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 Clock Operation Mode Select Register (OSCCTL)

Address: FFEFH After reset: 00H R/W

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

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

EXCLKS	OSCSELS	Subsystem clock operation mode	P123/XT1 pin	P124/XT2/EXCLKS pin
0	0	I/O port mode	I/O port	
0	1	XT1 oscillation mode	Crystal resonator connection	
1	0	I/O port mode	I/O port	
1	1	External clock input mode	I/O port	External clock input

AMPH	Operating frequency control
0	$4\text{ MHz} \le f_{IN} \le 10\text{ MHz}$
1	$10\text{ MHz} < f_{IN} \le 20\text{ MHz}$

Cautions 1. Be sure to set AMPH to 1 if the high-speed system clock oscillation frequency exceeds 10 MHz.

2. Set AMPH before setting the main clock mode register (MCM).

3. Set AMPH before setting the peripheral functions after a reset release. The value of AMPH can be changed only once after a reset release. When the high-speed system clock (X1 oscillation) is selected as the CPU clock, supply of the CPU clock is stopped for 4.06 to 16.12 µs after AMPH is set to 1. When the high-speed system clock (external clock input) is selected as the CPU clock, supply of the CPU clock is stopped for the duration of 160 external clocks after AMPH is set to 1.

4. If the STOP instruction is executed when AMPH = 1, supply of the CPU clock is stopped for 4.06 to 16.12 µs after the STOP mode is released when the internal high-speed oscillation clock is selected as the CPU clock, or for the duration of 160 external clocks when the high-speed system clock (external clock input) is selected as the CPU clock. When the high-speed system clock (X1 oscillation) is selected as the CPU clock, the oscillation stabilization time is counted after the STOP mode is released.

5. AMPH can be changed only once after a reset release.

Cautions 6. 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).

7. To change the value of EXCLKS and OSCSELS, confirm that bit 5 (CLS) of the processor clock control register (PCC) is 0 (the CPU is operating with the high-speed system clock).

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

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

This is the status register of the X1 clock oscillation stabilization time counter. If the internal high-speed oscillation clock or subsystem clock is 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

MOST11	MOST13	MOST14	MOST15	MOST16	Oscillation stabilization time status [min.]		
						fx = 10 MHz	fx = 20 MHz
1	0	0	0	0	2^{11}/fx	204.8 μs	102.4 μs
1	1	0	0	0	2^{13}/fx	819.2 μs	409.6 μs
1	1	1	0	0	2^{14}/fx	1.64 ms	819.2 μs
1	1	1	1	0	2^{15}/fx	3.27 ms	1.64 ms
1	1	1	1	1	2^{16}/fx	6.55 ms	3.27 ms

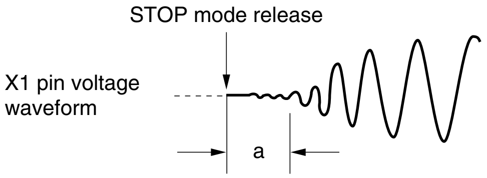
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST11 and remain 1.

2. If the STOP mode is entered and then released while the internal high-speed oscillation clock or subsystem 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**

The oscillation stabilization time counter counts up to the 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

(7) 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. The wait time set by OSTS is valid only after the STOP mode is released with the X1 clock selected as the CPU clock. After the STOP mode is released with the internal high-speed oscillation clock or subsystem clock selected as the CPU clock, the oscillation stabilization time must be confirmed by 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 = 10 MHz	fx = 20 MHz
0	0	1	2^{11}/fx	204.8 μs	102.4 μs
0	1	0	2^{13}/fx	819.2 μs	409.6 μs
0	1	1	2^{14}/fx	1.64 ms	819.2 μs
1	0	0	2^{15}/fx	3.27 ms	1.64 ms
1	0	1	2^{16}/fx	6.55 ms	3.27 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.

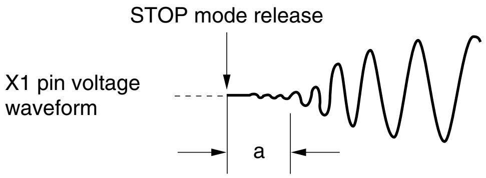
2. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

3. If the STOP mode is entered and then released while the internal high-speed oscillation clock or subsystem 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**

The oscillation stabilization time counter counts up to the 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

5.4 System Clock Oscillator

5.4.1 X1 oscillator

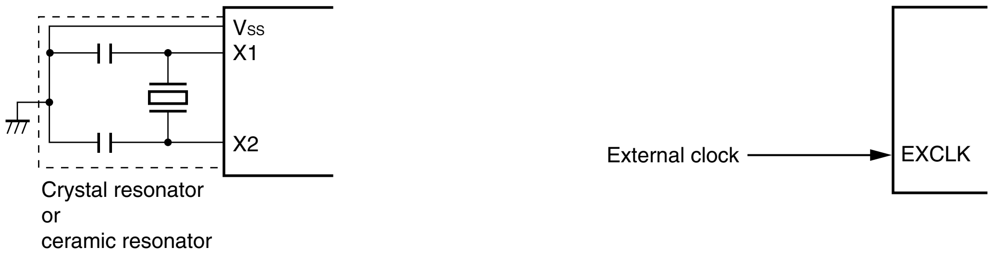
The X1 oscillator oscillates with a crystal resonator or ceramic resonator (4 to 20 MHz) connected to the X1 and X2 pins.

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

Figure 5-9 shows an example of the external circuit of the X1 oscillator.

Figure 5-9. Example of External Circuit of X1 Oscillator

(a) Crystal or ceramic oscillation **(b) External clock**



Cautions are listed on the next page.

5.4.2 XT1 oscillator

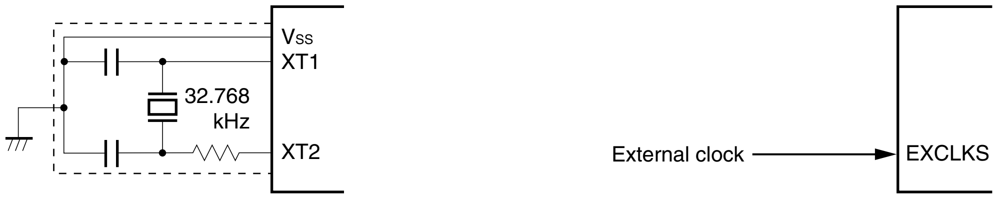
The XT1 oscillator oscillates with a crystal resonator (standard: 32.768 kHz) connected to the XT1 and XT2 pins.

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

Figure 5-10 shows an example of the external circuit of the XT1 oscillator.

Figure 5-10. Example of External Circuit of XT1 Oscillator

(a) Crystal oscillation **(b) External clock**



Cautions are listed on the next page.

Caution When using the X1 oscillator and XT1 oscillator, wire as follows in the area enclosed by the broken lines in the Figures 5-9 and 5-10 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.**

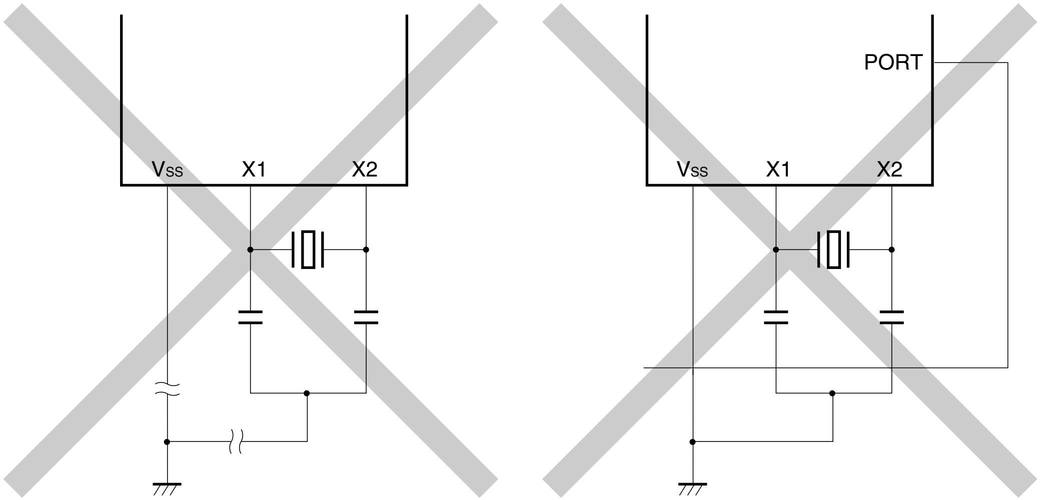
Note that the XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption.

Figure 5-11 shows examples of incorrect resonator connection.

Figure 5-11. Examples of Incorrect Resonator Connection (1/2)

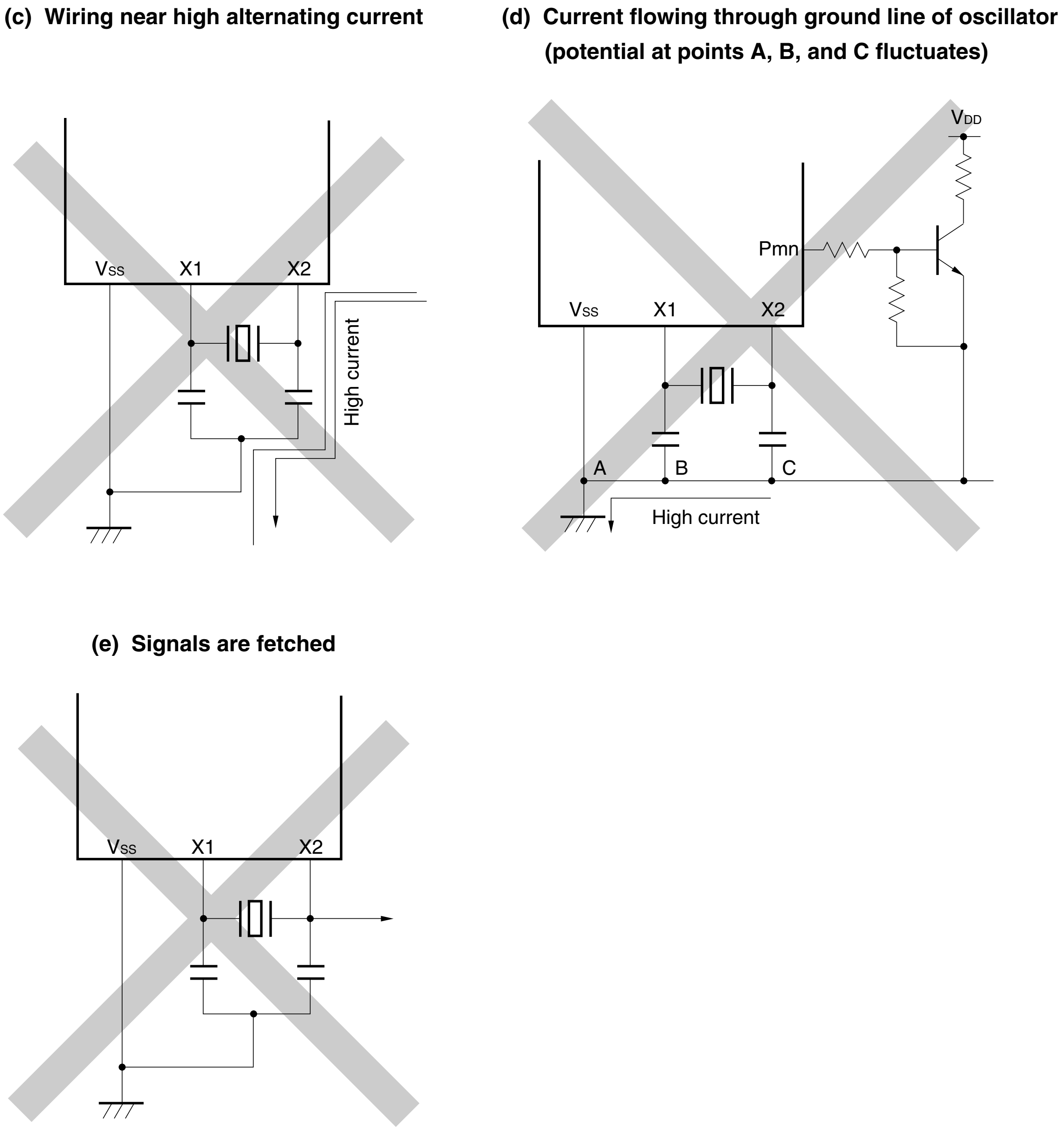
(a) Too long wiring

(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 5-11. 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 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 I/O mode (OSCSELS = 0) and connect them as follows.

Input (PM123/PM124 = 1): Independently connect to V_{DD} or V_{SS} via a resistor.
Output (PM123/PM124 = 0): Leave open.

Remark OSCSELS: Bit 4 of clock operation mode select register (OSCCTL)
PM123, PM124: Bits 3 and 4 of port mode register 12 (PM12)

5.4.4 Internal high-speed oscillator

The internal high-speed oscillator is incorporated in the 78K0/Dx2. Oscillation can be controlled by the internal oscillator mode register (RCM).

After a $\overline{\text{RESET}}$ release, the internal high-speed oscillation clock starts oscillation (8 MHz (TYP.)).

5.4.5 Internal low-speed oscillator

The internal low-speed oscillator is incorporated in the 78K0/Dx2.

The internal low-speed oscillation clock is only used as the watchdog timer and the clock of 8-bit timer H1. 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 oscillator mode register (RCM).

After a $\overline{\text{RESET}}$ release, the internal low-speed oscillation clock starts oscillation and the watchdog timer is operated (240 kHz (TYP.)).

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.

- Main system clock f_{MAIN}
  - High-speed system clock f_{IN}
    - X1 clock f_X
    - External main system clock f_{EXT}
  - Internal high-speed oscillation clock f_{OSC8}
- Subsystem clock f_{SUB}
  - XT1 clock f_{XT}
  - External subsystem clock f_{EXTS}
- Internal low-speed oscillation clock f_{OSC}
- CPU clock f_{CPU}
- Peripheral hardware clock f_{PRS}

The CPU starts operation when the on-chip internal high-speed oscillator starts outputting after a reset release in the 78K0/Dx2, thus enabling the following.

(1) Enhancement of security function

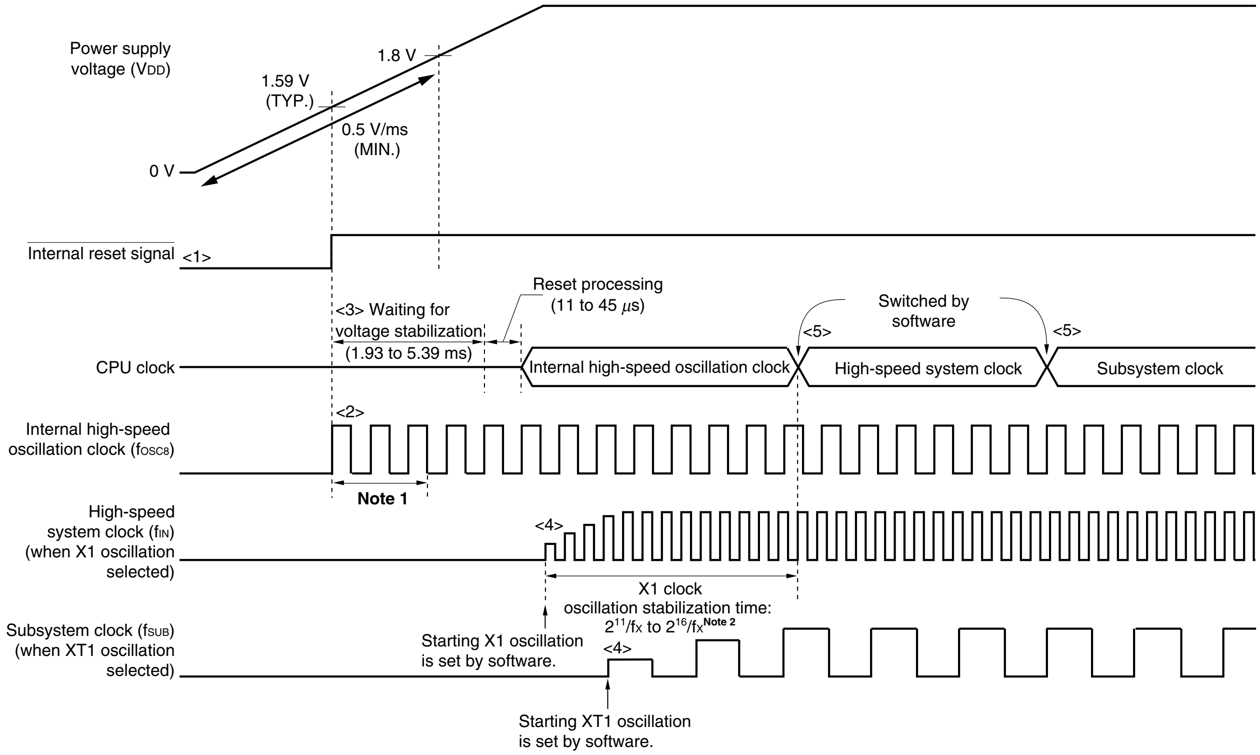
When the X1 clock is set as the CPU clock by the default setting, the device cannot operate if the X1 clock is damaged or badly connected and therefore does not operate after reset is released. However, the start clock of the CPU is the on-chip internal high-speed oscillation clock, so the device can 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.

A timing diagram of the CPU default start using the internal high-speed oscillation clock is shown in **Figures 5-12** and **5-13**.

Figure 5-12 Operation of the Clock Generating Circuit When Power Supply Voltage Injection (When 1.59 V POC mode setup (option byte: LVISTART = 0))


Power supply voltage (VDD)
1.8 V
1.59 V (TYP.)
0.5 V/ms (MIN.)
0 V
Internal reset signal
<1>
Reset processing (11 to 45 μs)
<3> Waiting for voltage stabilization (1.93 to 5.39 ms)
Switched by software
<5>
<5>
CPU clock
Internal high-speed oscillation clock
High-speed system clock
Subsystem clock
<2>
Internal high-speed oscillation clock (fOSC8)
Note 1
High-speed system clock (fIN) (when X1 oscillation selected)
<4>
X1 clock oscillation stabilization time: 2^{11}/fx to 2^{16}/fx Note 2
Subsystem clock (fSUB) (when XT1 oscillation selected)
Starting X1 oscillation is set by software.
<4>
Starting XT1 oscillation is set by software.

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).

<1> The internal reset signal by the power-on clear (POC) circuit is generated after a power supply injection.

<2> If power supply voltage exceeds 1.59 V (TYP.), reset will be released and the oscillation start of the high-speed oscillator will be carried out automatically.

<3> If power supply voltage is rose by inclination of 0.5 V/ms (MIN.), after the voltage stable waiting time of a power supply/regulator passed after reset release and reset processing will be performed, CPU carries out a start of operation with high-speed oscillation clock.

<4> One clock or XT1 clock should set up an oscillation start by software (see **(1)** in **5.6.1 Controlling high-speed system clock** and **(1)** in **5.6.3 Example of controlling subsystem clock**).

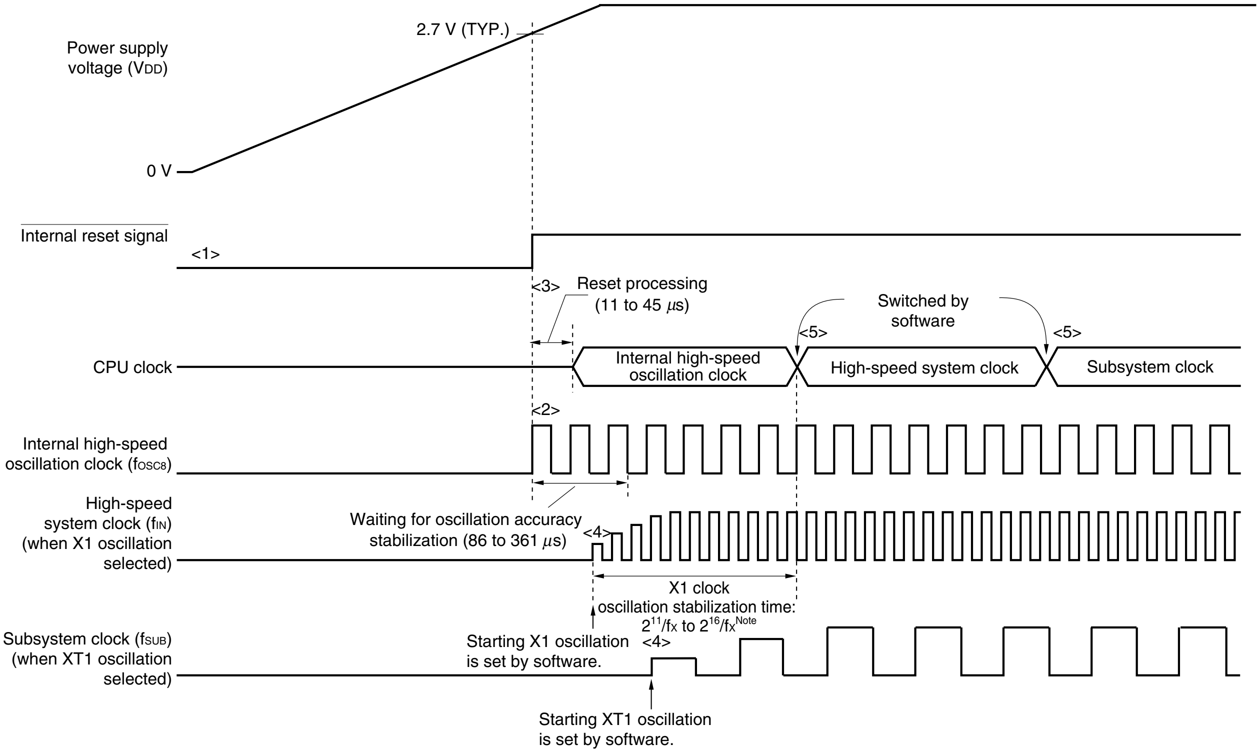
<5> When you change CPU to X1 clock or XT1 clock, set up a change by software after the oscillation stability waiting of a clock (see **(3)** in **5.6.1 Controlling high-speed system clock** and **(3)** in **5.6.3 Example of controlling subsystem clock**).

Cautions 1. When the standup of voltage until it reaches 1.8 V from the time of a power supply injection is looser than 0.5 V/ms (MIN.), input a low level into RESET pin, or set up 2.7 V/1.59 V POC mode (LVISTART = 1) from an option byte until it reaches 1.8 V from the time of a power supply injection (refer to Figure 5-13). When a low level is inputted into RESET pin until it reaches 1.8 V, after the reset release by RESET pin operates to the same timing as <2> of Figure 5-12 or subsequent ones.

2. When using the external clock input from EXCLK pin and EXCLKS pin, oscillation stable waiting time is unnecessary.

Remark The clock which is not used as a CPU clock can be suspended by setup of software during microcomputer operation. Moreover, high-speed oscillation clock and a high-speed system clock can suspend a clock by execution of a STOP command (see **(4)** in **5.6.1 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 5-13 Operation of the Clock Generating Circuit When Power Supply Voltage Injection (When 2.7 V/1.59V POC mode setup (option byte: LVISTART = 1))



Power supply voltage (VDD)
2.7 V (TYP.)
0 V
Internal reset signal
<1>
<3>
Reset processing (11 to 45 μs)
Switched by software
<5>
<5>
CPU clock
Internal high-speed oscillation clock
High-speed system clock
Subsystem clock
<2>
Internal high-speed oscillation clock (fOSC8)
High-speed system clock (fIN) (when X1 oscillation selected)
Waiting for oscillation accuracy stabilization (86 to 361 μs)
<4>
X1 clock oscillation stabilization time: 2^11/fx to 2^16/fx Note
Starting X1 oscillation is set by software.
<4>
Subsystem clock (fSUB) (when XT1 oscillation selected)
Starting XT1 oscillation is set by software.

Note Check the oscillation stable time of X1 clock with an oscillation stable time counter status register (OSTC) when STOP mode release in case the time of reset release (above figure) and a CPU clock are high-speed oscillation clocks . Moreover, when a CPU clock is a high-speed system clock (X1 oscillation), set up the oscillation stable time at the time of STOP mode release by the oscillation stable time selection register (OSTS).

<1> The internal reset signal by the power-on clear (POC) circuit is generated after a power supply injection.

<2> If power supply voltage exceeds 1.59 V (TYP.), reset will be canceled and the oscillation start of the high-speed oscillator will be carried out automatically.

<3> After reset release, after reset processing is performed, CPU carries out a start of operation with high-speed oscillation clock.

<4> X1 clock or XT1 clock should set up an oscillation start by software (see **(1)** in **5.6.1 Controlling high-speed system clock** and **(1)** in **5.6.3 Example of controlling subsystem clock**).

<5> When you change CPU to X1 clock or XT1 clock, set up a change by software after the oscillation stability waiting of a clock (see **(3)** in **5.6.1 Controlling high-speed system clock** and **(3)** in **5.6.3 Example of controlling subsystem clock**).

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 supply voltage rises from 1.59 V (TYP.) to 2.7 V (TYP.) within 1.93 ms, the power supply oscillation stabilization time of 0 to 5.39 ms is automatically generated before reset processing.

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

Remark The clock which is not used as a CPU clock can be suspended by setup of software during microcomputer operation. Moreover, high-speed oscillation clock and a high-speed system clock can suspend a clock by execution of a STOP command (see **(4)** in **5.6.1 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 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 X1/P121 and X2/EXCLK/P122 pins can be used as I/O port pins.

Caution The X1/P121 and 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 frequency (OSCCTL register)

Using AMPH, set the gain of the on-chip oscillator according to the frequency to be used.

AMPH[Note]	Operating Frequency Control
0	4 MHz ≤ f_{IN} ≤ 10 MHz
1	10 MHz < f_{IN} ≤ 20 MHz

Note Set AMPH before setting the peripheral functions after a reset release. The value of AMPH can be changed only once after a reset release. When AMPH is set to 1, the clock supply to the CPU is stopped for 4.06 to 16.12 μs.

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

<2> 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	

<3> Controlling oscillation of X1 clock (MOC register)

If MSTOP is cleared to 0, the X1 oscillator starts oscillating.

<4> 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 29 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)).

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

<1> Setting frequency (OSCCTL register)

Using AMPH, set the frequency to be used.

AMPH[Note]	Operating Frequency Control
0	4 MHz ≤ f_{IN} ≤ 10 MHz
1	10 MHz < f_{IN} ≤ 20 MHz

Note Set AMPH before setting the peripheral functions after a reset release. The value of AMPH can be changed only once after a reset release. The clock supply to the CPU is stopped for the duration of 160 external clocks after AMPH is set to 1.

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

<2> 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

<3> 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 29 ELECTRICAL SPECIFICATIONS ((A) GRADE 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_{MAIN})	Peripheral Hardware Clock (f_{PRS})
1	1	High-speed system clock (f_{IN})	High-speed system clock (f_{IN})

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_{MAIN}
	0	0	1	$f_{MAIN}/2$ (default)
	0	1	0	$f_{MAIN}/2^2$
	0	1	1	$f_{MAIN}/2^3$
	1	0	0	$f_{MAIN}/2^4$
	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 and stopping the X1 oscillation (disabling clock input if the external clock is used)
- 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 20 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_{MAIN})	Peripheral Hardware Clock (f_{PRS})
0	0	Internal high-speed oscillation clock (f_{OSC8})	Internal high-speed oscillation clock (f_{OSC8})
0	1		
1	0		High-speed system clock (f_{IN})

<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_{MAIN}
	0	0	1	$f_{MAIN}/2$ (default)
	0	1	0	$f_{MAIN}/2^2$
	0	1	1	$f_{MAIN}/2^3$
	1	0	0	$f_{MAIN}/2^4$
	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 20 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.
- External subsystem clock: External clock is input to the EXCLKS pin.

When the subsystem clock is not used, the XT1/P123 and XT2/EXCLKS/P124 pins can be used as I/O port pins.

Caution The XT1/P123 and XT2/EXCLKS/P124 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 XT1 clock
(2) When using external subsystem clock
(3) When using subsystem clock as CPU clock
(4) 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 EXCLKS and OSCSELS are set as any of the following, the mode is switched from port mode to XT1 oscillation mode.

EXCLKS	OSCSELS	Operation Mode of Subsystem Clock Pin	P123/XT1 Pin	P124/XT2/ EXCLKS Pin
0	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 EXCLKS and OSCSELS while the subsystem clock is operating.

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

<1> Setting XT1 and XT2 pins, selecting XT1 clock/external clock and controlling oscillation (PCC and OSCCTL registers)

When EXCLKS and OSCSELS are set to 1, the mode is switched from port mode to external clock input mode. In this case, input the external clock to the EXCLKS/XT2/P124 pins.

EXCLKS	OSCSELS	Operation Mode of Subsystem Clock Pin	P123/XT1 Pin	P124/XT2/ EXCLKS Pin
1	1	External clock input mode	I/O port	External clock input

Caution Do not change the value of EXCLKS and OSCSELS while the subsystem clock is operating.

(3) 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** and **(2) Example of setting procedure when using the external subsystem 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

(4) 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 (the input of the external clock is disabled).

Cautions 1. Be sure to confirm that CLS = 0 when clearing OSCSELS to 0. In addition, stop the watch timer if it is operating on the subsystem clock.

2. The subsystem clock oscillation cannot be stopped using the STOP instruction.

5.6.4 Controlling internal low-speed oscillation clock

The internal low-speed oscillation clock is a clock for the watchdog timer. It cannot be used as the CPU clock. With this clock, only the following peripheral hardware can operate.

- Watchdog timer
- 8-bit timer H1 (if fOSC is selected as the count clock)

In addition, the following operation modes can be selected by the option byte.

- Internal low-speed oscillation clock oscillation cannot be stopped
- Internal low-speed oscillation clock oscillation 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) To stop the internal low-speed oscillation clock (example of setting method)

<1> Setting LSRSTOP to 1 (RCM register)
If LSRSTOP is set to 1, the internal low-speed oscillator oscillation is stopped.

(2) To oscillate the internal low-speed oscillation clock (example of setting method)

<1> Clearing LSRSTOP to 0 (RCM register)
If LSRSTOP is cleared to 0, the internal low-speed oscillation clock is oscillated.

Caution If "Internal low-speed oscillation clock oscillation 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 5-3. Clocks Supplied to CPU and Peripheral Hardware, and Register Setting

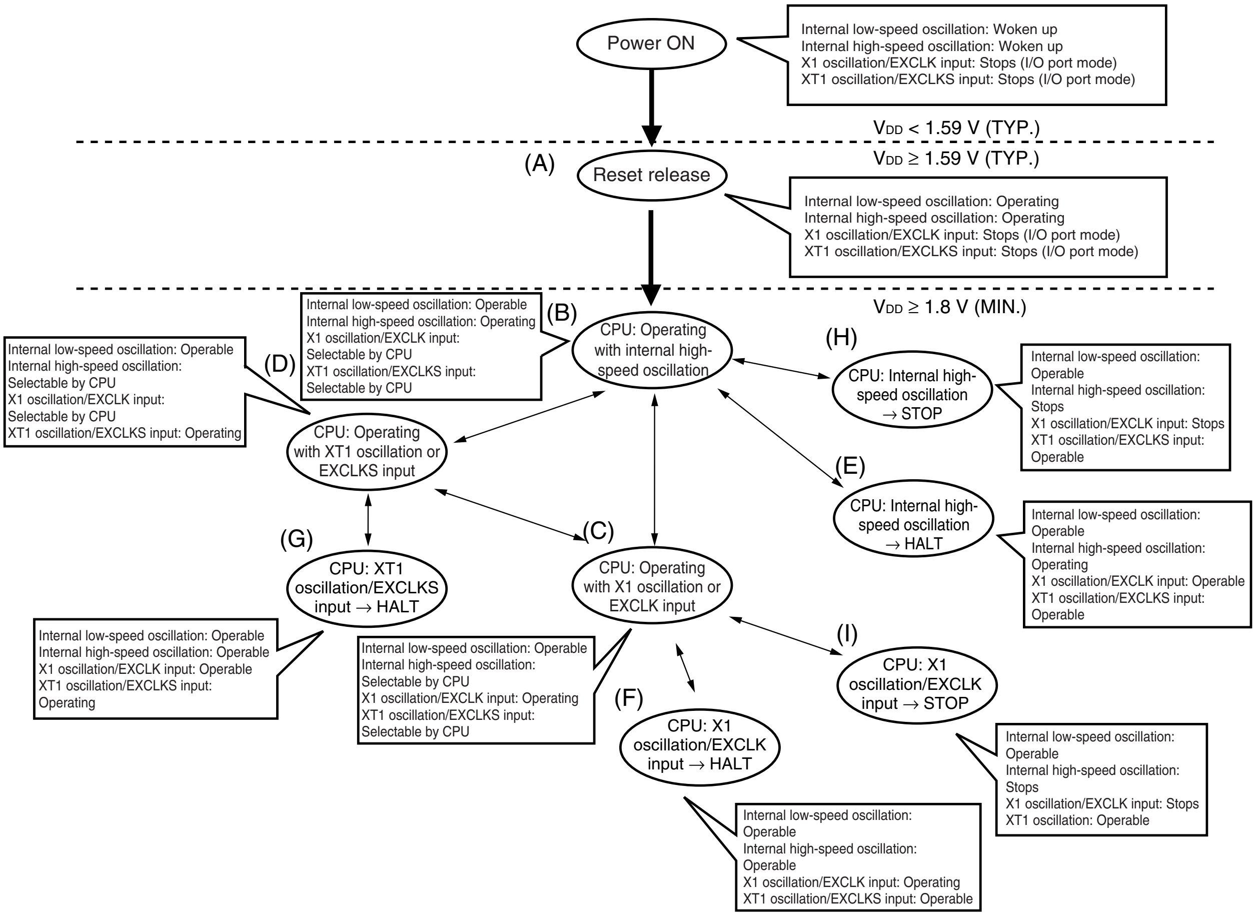
XSEL	CSS	MCM0	EXCLK	Supplied Clock	
				Clock Supplied to CPU	Clock Supplied to Peripheral Hardware
0	0	×	×	Internal high-speed oscillation clock	
0	1	×	×	Subsystem clock	Internal high-speed oscillation clock
1	0	0	0	Internal high-speed oscillation clock	X1 clock
1	0	0	1		External main system clock
1	0	1	0	X1 clock	
1	0	1	1	External main system clock	
1	1	0	0	Subsystem clock	X1 clock
1	1	0	1		External main system clock
1	1	1	0		X1 clock
1	1	1	1		External main system clock

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)

5.6.6 CPU clock status transition diagram

Figure 5-14 shows the CPU clock status transition diagram of this product.

Figure 5-14. CPU Clock Status Transition Diagram
(When 1.59 V POC Mode is Set (Option Byte: LVISTART = 0))


Power ON
Internal low-speed oscillation: Woken up
Internal high-speed oscillation: Woken up
X1 oscillation/EXCLK input: Stops (I/O port mode)
XT1 oscillation/EXCLKS input: Stops (I/O port mode)
VDD < 1.59 V (TYP.)
VDD ≥ 1.59 V (TYP.)
(A)
Reset release
Internal low-speed oscillation: Operating
Internal high-speed oscillation: Operating
X1 oscillation/EXCLK input: Stops (I/O port mode)
XT1 oscillation/EXCLKS input: Stops (I/O port mode)
VDD ≥ 1.8 V (MIN.)
Internal low-speed oscillation: Operable
Internal high-speed oscillation: Operating
X1 oscillation/EXCLK input: Selectable by CPU
XT1 oscillation/EXCLKS input: Selectable by CPU
(B)
CPU: Operating with internal high-speed oscillation
(H)
CPU: Internal high-speed oscillation → STOP
Internal low-speed oscillation: Operable
Internal high-speed oscillation: Stops
X1 oscillation/EXCLK input: Stops
XT1 oscillation/EXCLKS input: Operable
Internal low-speed oscillation: Operable
Internal high-speed oscillation: Selectable by CPU
X1 oscillation/EXCLK input: Selectable by CPU
XT1 oscillation/EXCLKS input: Operating
(D)
CPU: Operating with XT1 oscillation or EXCLKS input
(E)
CPU: Internal high-speed oscillation → HALT
Internal low-speed oscillation: Operable
Internal high-speed oscillation: Operating
X1 oscillation/EXCLK input: Operable
XT1 oscillation/EXCLKS input: Operable
(G)
CPU: XT1 oscillation/EXCLKS input → HALT
(C)
CPU: Operating with X1 oscillation or EXCLK input
Internal low-speed oscillation: Operable
Internal high-speed oscillation: Operable
X1 oscillation/EXCLK input: Operable
XT1 oscillation/EXCLKS input: Operating
Internal low-speed oscillation: Operable
Internal high-speed oscillation: Selectable by CPU
X1 oscillation/EXCLK input: Operating
XT1 oscillation/EXCLKS input: Selectable by CPU
(I)
CPU: X1 oscillation/EXCLK input → STOP
(F)
CPU: X1 oscillation/EXCLK input → HALT
Internal low-speed oscillation: Operable
Internal high-speed oscillation: Stops
X1 oscillation/EXCLK input: Stops
XT1 oscillation: Operable
Internal low-speed oscillation: Operable
Internal high-speed oscillation: Operable
X1 oscillation/EXCLK input: Operating
XT1 oscillation/EXCLKS input: Operable

Remark In the 2.7 V/1.59 V POC mode (option byte: LVISTART = 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 45 μs).

Table 5-4 shows transition of the CPU clock and examples of setting the SFR registers.

Table 5-4. CPU Clock Transition and SFR Register Setting Examples (1/5)

(1) 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	AMPH	EXCLK	OSCSEL	MSTOP	OSTC Register	XSEL	MCM0
(A) → (B) → (C) (X1 clock: less than 10 MHz)	0	0	1	0	Must be checked	1	1
(A) → (B) → (C) (external main clock: less than 10 MHz)	0	1	1	0	Must not be checked	1	1
(A) → (B) → (C) (X1 clock: 10 MHz or more)	1	0	1	0	Must be checked	1	1
(A) → (B) → (C) (external main clock: 10 MHz or more)	1	1	1	0	Must not be checked	1	1

(2) 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).

(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	EXCLKS	OSCSELS	Waiting for Oscillation Stabilization	CSS
(A) → (B) → (D) (XT1 clock)	0	1	Necessary	1
(A) → (B) → (D) (external subsystem clock)	1	1	Unnecessary	1

Remarks 1. (A) to (I) in **Table 5-4** correspond to (A) to (I) in **Figure 5-14**.

2. EXCLK, OSCSEL, EXCLKS, OSCSELS, AMPH:
Bits 7 to 4 and 0 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 5-4. CPU Clock Transition and SFR Register Setting Examples (2/5)

(4) CPU clock changing from internal high-speed oscillation clock (B) to high-speed system clock (C)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register Status Transition	AMPH	EXCLK	OSCSEL	MSTOP	OSTC Register	XSEL	MCM0
(B) → (C) (X1 clock: less than 10 MHz)	0	0	1	0	Must be checked	1	1
(B) → (C) (external main clock: less than 10 MHz)	0	1	1	0	Must not be checked	1	1
(B) → (C) (X1 clock: 10 MHz or more)	1	0	1	0	Must be checked	1	1
(B) → (C) (external main clock: 10 MHz or more)	1	1	1	0	Must not be checked	1	1

Unnecessary if these registers are already set

Unnecessary if the CPU is operating with the high-speed system clock

↑ Unnecessary if this register is already set

(5) CPU clock changing from internal high-speed oscillation clock (B) to subsystem clock (D)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register Status Transition	EXCLKS	OSCSELS	Waiting for Oscillation Stabilization	CSS
(B) → (D) (XT1 clock)	0	1	Necessary	1
(B) → (D) (external subsystem clock)	1	1	Unnecessary	1

Unnecessary if the CPU is operating with the subsystem clock

Remarks 1. (A) to (I) in **Table 5-4** correspond to (A) to (I) in **Figure 5-14**.

2. EXCLK, OSCSEL, EXCLKS, OSCSELS, AMPH:
Bits 7 to 4 and 0 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 5-4. CPU Clock Transition and SFR Register Setting Examples (3/5)

(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

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	EXCLKS	OSCSELS	Waiting for Oscillation Stabilization	CSS
(C) → (D) (XT1 clock)	0	1	Necessary	1
(C) → (D) (external subsystem clock)	1	1	Unnecessary	1

Unnecessary if the CPU is operating
with the subsystem clock

Remarks 1. (A) to (I) in **Table 5-4** correspond to (A) to (I) in **Figure 5-14**.

2. EXCLKS, OSCSELS:
Bits 5and 4 of the clock operation mode select register (OSCCTL)
MCM0: Bit 0 of the main clock mode register (MCM)
CSS: Bit 4 of the processor clock control register (PCC)
RSTS, RSTOP: Bits 7 and 0 of the internal oscillator mode register (RCM)

Table 5-4. CPU Clock Transition and SFR Register Setting Examples (4/5)

(8) CPU clock changing from subsystem clock (D) to high-speed system clock (C)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register Status Transition	AMPH	EXCLK	OSCSEL	MSTOP	OSTC Register	XSEL	MCM0	CSS
(D) → (C) (X1 clock: less than 10 MHz)	0	0	1	0	Must be checked	1	1	0
(D) → (C) (external main clock: less than 10 MHz)	0	1	1	0	Must not be checked	1	1	0
(D) → (C) (X1 clock: 10 MHz or more)	1	0	1	0	Must be checked	1	1	0
(D) → (C) (external main clock: 10 MHz or more)	1	1	1	0	Must not be checked	1	1	0

AMPH, EXCLK, OSCSEL: Unnecessary if these registers are already set

MSTOP, OSTC Register: Unnecessary if the CPU is operating with the high-speed system clock

XSEL ↑ Unnecessary if this register is already set

(9) 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 5-4** correspond to (A) to (I) in **Figure 5-14**.

2. EXCLK, OSCSEL, AMPH:
Bits 7, 6, and 0 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)
RSTS, RSTOP: Bits 7 and 0 of the internal oscillator mode register (RCM)

Table 5-4. CPU Clock Transition and SFR Register Setting Examples (5/5)

(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

Remark (A) to (I) in **Table 5-4** correspond to (A) to (I) in **Figure 5-14**.

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 5-5. 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). • Clock supply to CPU is stopped for 4.06 to 16.12 μs after AMPH has been set to 1.
	External main system clock	Enabling input of external clock from EXCLK pin • MSTOP = 0, OSCSEL = 1, EXCLK = 1	• Internal high-speed oscillator can be stopped (RSTOP = 1). • Clock supply to CPU is stopped for the duration of 160 external clocks from the EXCLK pin after AMPH has been set to 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 • EXCLKS = 0, 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).
Internal high-speed oscillation clock	External subsystem clock	Enabling input of external clock from EXCLKS pin • EXCLKS = 1, OSCSELS = 1	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, external subsystem 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 or external subsystem clock input can be disabled (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	• XT1 oscillation can be stopped or external subsystem clock input can be disabled (OSCSELS = 0). • Clock supply to CPU is stopped for 4.06 to 16.12 μs after AMPH has been set to 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	• XT1 oscillation can be stopped or external subsystem clock input can be disabled (OSCSELS = 0). • Clock supply to CPU is stopped for the duration of 160 external clocks from the EXCLK pin after AMPH has been set to 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 5-6**).

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 5-6. Time Required for Switchover of CPU Clock and Main System Clock Cycle Division Factor

Set Value Before Switchover				Set Value After Switchover [clock(s)]					
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	16	16	16	$2f_{MAIN}/f_{SUB}$
	0	0	1	8		8	8	8	f_{MAIN}/f_{SUB}
	0	1	0	4	4		4	4	$f_{MAIN}/2f_{SUB}$
	0	1	1	2	2	2		2	$f_{MAIN}/4f_{SUB}$
	1	0	0	1	1	1	1		$f_{MAIN}/8f_{SUB}$
1	×	×	×	2	2	2	2	2	

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 5-6** 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 $f_{MAIN}/2$ to f_{SUB} /2 (@ oscillation with f_{MAIN} = 10 MHz, f_{SUB} = 32.768 kHz)

$f_{MAIN}/f_{SUB} = 10000/32.768 \cong 305.1 \rightarrow 306$ 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 5-7**).

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 5-7. Maximum Time Required for Main System Clock Switchover

Set Value Before Switchover	Set Value After Switchover [clocks]	
MCM0	MCM0	
	0	1
0		$1 + 2f_{OSC8}/f_{IN}$
1	$1 + 2f_{IN}/f_{OSC8}$	

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 5-7** is the number of main system clocks before switchover.

2. Calculate the number of clocks in **Table 5-7** 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_{OSC8} = 8 MHz, f_{IN} = 10 MHz)

$1 + 2f_{OSC8}/f_{IN} = 1 + 2 \times 8/10 = 1 + 2 \times 0.8 = 1 + 1.6 = 2.6 \rightarrow$ 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 5-8. 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
External subsystem clock		

CHAPTER 6 16-BIT TIMER/EVENT COUNTER P (TMP)

Timer P (TMP) is a 16-bit timer/event counter.
The 78K0/Dx2 has five timer/event counter channels, TMP0 to TMP4.

6.1 Overview

An outline of TMPn (n = 0 to 4) is shown below.

- Clock selection: 8 ways
- Capture/trigger input pins: 2
- External event count input pins: 1
- External trigger input pins: 1
- Timer/counters: 1
- Capture/compare registers: 2
- Capture/compare match interrupt request signals: 2
- Timer output pins: 2

6.2 Functions

TMPn (n = 0 to 4) has the following functions.

- Interval timer
- External event counter
- External trigger pulse output
- One-shot pulse output
- PWM output
- Free-running timer
- Pulse width measurement
- Timer conjunction function
- Timer synchronous operation function

6.3 Configuration

TMPn includes the following hardware.

Table 6-1. Configuration of TMPn

Item	Configuration
Timer register	16-bit counter
Registers	TMPn capture/compare registers 0, 1 (TPnCCR0, TPnCCR1) TMPn counter read buffer register (TPnCNT) CCR0, CCR1 buffer registers
Timer inputs/outputs	2 (TIOPn0[Note 1], TIOPn1 pins)
Control registers[Note 2]	TMPn control registers 0, 1 (TPnCTL0, TPnCTL1) TMPn I/O control registers 0 to 2 (TPnIOC0 to TPnIOC2) TMPn option register 0 (TPnOPT0) Input switch control register (ISC) TMPn input noise filter control register 0 to 2 (TIPNF0 to TIPNF2)

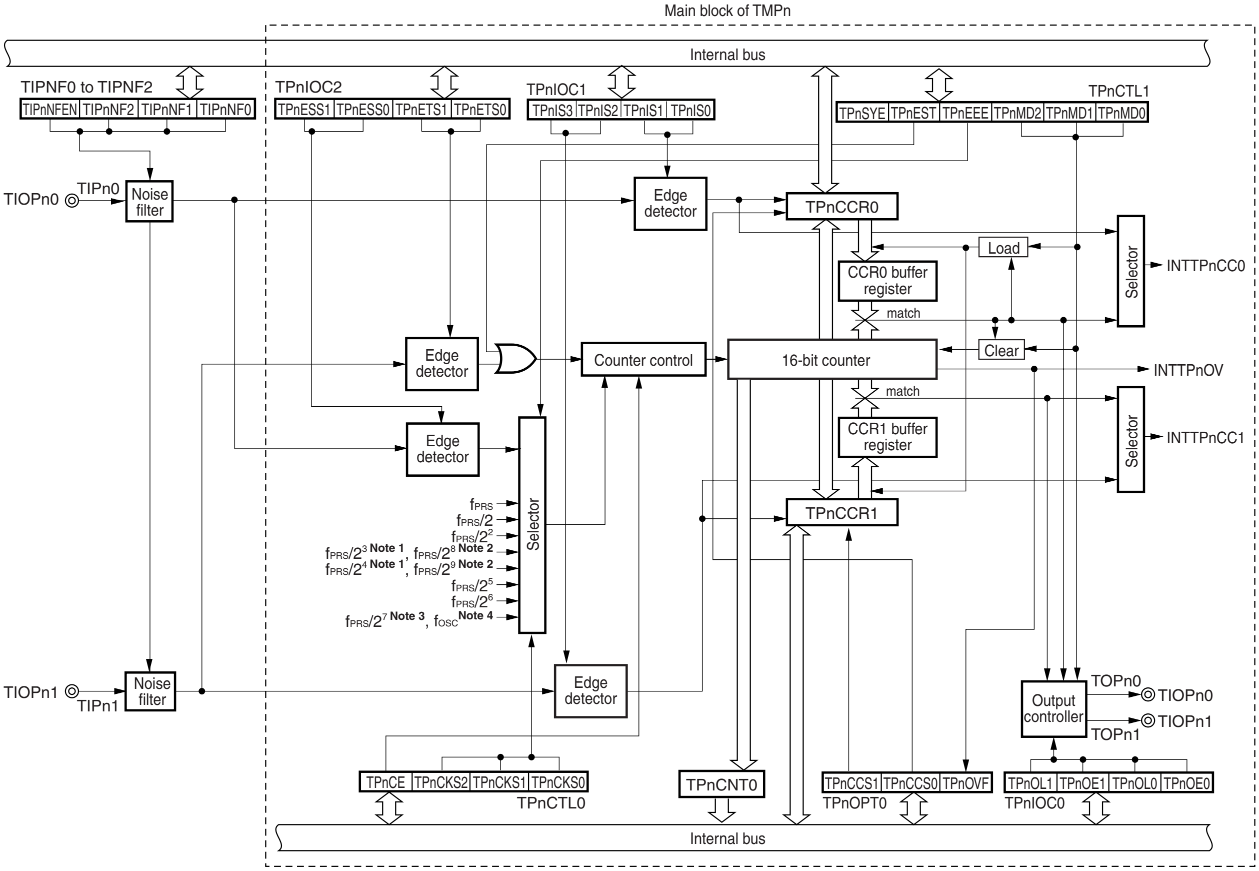
Notes 1. The TIOPn0 pin functions alternately as a capture trigger input signal, external event count input signal, and external trigger input signal.

2. When using the functions of the TIOPn0 and TIOPn1 pins, see **4.4 Settings of LCDPFALL, LCDPF0, LCDPF3, ISC, Port Mode Register, and Output Latch When Using Alternate Function**.

Remark n = 0 to 4

<R>

Figure 6-1. Block Diagram of TMPn



Notes 1. TMP0, TMP1, TMP3, and TMP4 only.

2. TMP2 only.

3. TMP0 to TMP2, TMP4 only.

4. TMP3 only.

Remark fPRS: Peripheral hardware clock frequency

fOSC: Internal low-speed oscillation clock frequency

(1) 16-bit counter

This 16-bit counter can count internal clocks or external events.

The count value of this counter can be read by using the TPnCNT register.

When the TPnCTL0.TPnCE bit = 0, the value of the 16-bit counter is FFFFH. If the TPnCNT register is read at this time, 0000H is read.

Reset sets the TPnCE bit to 0. Therefore, the 16-bit counter is set to FFFFH.

(2) CCR0 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TPnCCR0 register is used as a compare register, the value written to the TPnCCR0 register is transferred to the CCR0 buffer register. When the count value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTPnCC0) is generated.

The CCR0 buffer register cannot be read or written directly.
The CCR0 buffer register is cleared to 0000H after reset, as the TPnCCR0 register is cleared to 0000H.

(3) CCR1 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.
When the TPnCCR1 register is used as a compare register, the value written to the TPnCCR1 register is transferred to the CCR1 buffer register. When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTPnCC1) is generated.
The CCR1 buffer register cannot be read or written directly.
The CCR1 buffer register is cleared to 0000H after reset, as the TPnCCR1 register is cleared to 0000H.

(4) Edge detector

This circuit detects the valid edges input to the TIOPn0 and TIOPn1 pins. No edge, rising edge, falling edge, or both the rising and falling edges can be selected as the valid edge by using the TPnIOC1 and TPnIOC2 registers.

(5) Output controller

This circuit controls the output of the TIOPn0 and TIOPn1 pins. The output controller is controlled by the TPnIOC0 register.

(6) Selector

This selector selects the count clock for the 16-bit counter. Eight types of internal clocks or an external event can be selected as the count clock.

(7) Digital noise filter

This circuit rejects the noise of digital input. The filters are controlled with the input noise filter control registers (TIPNF0 to TIPNF2).

(8) TMP2 and TMP3 conjunction

TMP3 output of TIOP30 is connected to TMP2 input of TIOP21. This function is controlled with the bit3 of the input switch control register (ISC).

6.4 Registers

The registers that control TMPn are as follows.

- TMPn control register 0 (TPnCTL0)
- TMPn control register 1 (TPnCTL1)
- TMPn I/O control register 0 (TPnIOC0)
- TMPn I/O control register 1 (TPnIOC1)
- TMPn I/O control register 2 (TPnIOC2)
- TMPn option register 0 (TPnOPT0)
- TMPn capture/compare register 0 (TPnCCR0)
- TMPn capture/compare register 1 (TPnCCR1)
- TMPn counter read buffer register (TPnCNT)
- Input switch control register (ISC)
- TMP0, TMP1 input noise filter control register 0 (TIPNF0)
- TMP2, TMP3 input noise filter control register 1 (TIPNF1)
- TMP4 input noise filter control register 2 (TIPNF2)

Remarks 1. When using the functions of the TIOPn0 and TIOPn1 pins, see **4.4 Settings of LCDPFALL, LCDPF0, LCDPF3, ISC, Port Mode Register, and Output Latch When Using Alternate Function**.

2. n = 0 to 4

(1) TMPn control register 0 (TPnCTL0)

The TPnCTL0 register is an 8-bit register that controls the operation of TMPn.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

The same value can always be written to the TPnCTL0 register by software.

After reset: 00H R/W Address: TP0CTL0 FFFFF980H, TP1CTL0 FFFFF990H
TP2CTL0 FFFFF9A0H, TP3CTL0 FFFFF9B0H
TP4CTL0 FFFFFF10H

	<7>	6	5	4	3	2	1	0
TPnCTL0 (n = 0 to 4)	TPnCE	0	0	0	0	TPnCKS2	TPnCKS1	TPnCKS0

TPnCE	TMPn operation control
0	TMPn operation disabled (TMPn reset asynchronously[Note]).
1	TMPn operation enabled. TMPn operation started.

TPnCKS2	TPnCKS1	TPnCKS0	Internal count clock selection		
			n = 0, 1, 4 (TMP0, TMP1, TMP4)	n = 2 (TMP2)	n = 3 (TMP3)
0	0	0	f_{PRS}		
0	0	1	$f_{PRS}/2$		
0	1	0	$f_{PRS}/2^2$		
0	1	1	$f_{PRS}/2^3$	$f_{PRS}/2^8$	$f_{PRS}/2^3$
1	0	0	$f_{PRS}/2^4$	$f_{PRS}/2^9$	$f_{PRS}/2^4$
1	0	1	$f_{PRS}/2^5$		
1	1	0	$f_{PRS}/2^6$		
1	1	1	$f_{PRS}/2^7$		f_{OSC}

Note TPnOPT0.TPnOVF bit, 16-bit counter, timer output (TIOPn0, TIOPn1 pins)

Cautions 1. Set the TPnCKS2 to TPnCKS0 bits when the TPnCE bit = 0. When the value of the TPnCE bit is changed from 0 to 1, the TPnCKS2 to TPnCKS0 bits can be set simultaneously.

2. Be sure to clear bits 3 to 6 to “0”.

Remark f_{PRS}: Peripheral hardware clock frequency
f_{OSC}: Internal low-speed oscillation clock frequency

(2) TMPn control register 1 (TPnCTL1)

The TPnCTL1 register is an 8-bit register that controls the operation of TMPn.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: TP0CTL1 FFFFF981H, TP1CTL1 FFFFF991H
TP2CTL1 FFFFF9A1H, TP3CTL1 FFFFF9B1H
TP4CTL1 FFFFFF11H

	7	<6>	<5>	4	3	2	1	0
TPnCTL1	TPnSYE	TPnEST	TPnEEE	0	0	TPnMD2	TPnMD1	TPnMD0

(n = 0 to 4)

<R>

TPnSYE	Tuned operation mode enable control
0	Independent operation mode (asynchronous operation mode)
1	Tuned operation mode (specification of slave operation) In this mode, timer P can operate in synchronization with a master timer. For TMP1, TMP2, TMP3, and TMP4 is specified as a slave timer. For the tuned operation mode, refer to **6.6 Timer Synchronous Operation Function**.

TPnEST	Software trigger control
0	–
1	Generates a valid signal for external trigger input. • In one-shot pulse output mode: A one-shot pulse is output with writing 1 to the TPnEST bit as the trigger. • In external trigger pulse output mode: A PWM waveform is output with writing 1 to the TPnEST bit as the trigger.

TPnEEE	Count clock selection Selects whether counting is performed with the internal count clock or the valid edge of the external event count input.
0	Disable operation with external event count input. (Perform counting with the count clock selected by the TPnCTL0.TPnCK0 to TPnCK2 bits.)
1	Enable operation with external event count input. (Perform counting at the valid edge of the external event count input signal.)

(2/2)

TPnMD2	TPnMD1	TPnMD0	Timer mode selection
0	0	0	Interval timer mode
0	0	1	External event count mode
0	1	0	External trigger pulse output mode
0	1	1	One-shot pulse output mode
1	0	0	PWM output mode
1	0	1	Free-running timer mode
1	1	0	Pulse width measurement mode
1	1	1	Setting prohibited

Cautions 1. The TPnEST bit is valid only in the external trigger pulse output mode or one-shot pulse output mode. In any other mode, writing 1 to this bit is ignored.

2. External event count input is selected in the external event count mode regardless of the value of the TPnEEE bit.

3. Set the TPnEEE and TPnMD2 to TPnMD0 bits when the TPnCTL0.TPnCE bit = 0. (The same value can be written when the TPnCE bit = 1.) The operation is not guaranteed when rewriting is performed with the TPnCE bit = 1. If rewriting was mistakenly performed, clear the TPnCE bit to 0 and then set the bits again.

4. Be sure to clear bits 3, 4, and 7 to “0”.

(3) TMPn I/O control register 0 (TPnIOC0)

The TPnIOC0 register is an 8-bit register that controls the timer output (TIOPn0, TIOPn1 pins).

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: TP0IOC0 FFFFF982H, TP1IOC0 FFFFF992H
TP2IOC0 FFFFF9A2H, TP3IOC0 FFFFF9B2H
TP4IOC0 FFFFFF12H

	7	6	5	4	3	<2>	1	<0>
TPnIOC0	0	0	0	0	TPnOL1	TPnOE1	TPnOL0	TPnOE0

(n = 0 to 4)

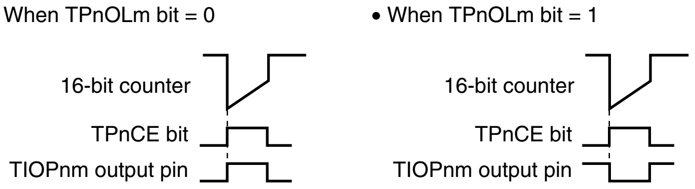
TPnOL1	TIOPn1 pin output level setting[Note]
0	TIOPn1 pin output starts at high level
1	TIOPn1 pin output starts at low level

TPnOE1	TIOPn1 pin output setting
0	Timer output disabled • When TPnOL1 bit = 0: Low level is output from the TIOPn1 pin • When TPnOL1 bit = 1: High level is output from the TIOPn1 pin
1	Timer output enabled (a square wave is output from the TIOPn1 pin).

TPnOL0	TIOPn0 pin output level setting[Note]
0	TIOPn0 pin output starts at high level
1	TIOPn0 pin output starts at low level

TPnOE0	TIOPn0 pin output setting
0	Timer output disabled • When TPnOL0 bit = 0: Low level is output from the TIOPn0 pin • When TPnOL0 bit = 1: High level is output from the TIOPn0 pin
1	Timer output enabled (a square wave is output from the TIOPn0 pin).

Note The output level of the timer output pin (TIOPnm) specified by the TPnOLm bit is shown below (m = 0, 1).



(2/2)

Cautions 1. Rewrite the TPnOL1, TPnOE1, TPnOL0, and TPnOE0 bits when the TPnCTL0.TPnCE bit = 0. (The same value can be written when the TPnCE bit = 1.) If rewriting was mistakenly performed, clear the TPnCE bit to 0 and then set the bits again.

2. Even if the TPnOLm bit is manipulated when the TPnCE and TPnOEm bits are 0, the TIOPnm pin output level varies (m = 0, 1).

(4) TMPn I/O control register 1 (TPnIOC1)

The TPnIOC1 register is an 8-bit register that controls the valid edge of the capture trigger input signals (TIOPn0, TIOPn1 pins).

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: TP0IOC1 FFFFF983H, TP1IOC1 FFFFF993H
TP2IOC1 FFFFF9A3H, TP3IOC1 FFFFF9B3H
TP4IOC1 FFFFFF13H

	7	6	5	4	3	2	1	0
TPnIOC1 (n = 0 to 4)	0	0	0	0	TPnIS3	TPnIS2	TPnIS1	TPnIS0

TPnIS3	TPnIS2	Capture trigger input signal (TIOPn1 pin) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TPnIS1	TPnIS0	Capture trigger input signal (TIOPn0 pin) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Cautions 1. Rewrite the TPnIS3 to TPnIS0 bits when the TPnCTL0.TPnCE bit = 0. (The same value can be written when the TPnCE bit = 1.) If rewriting was mistakenly performed, clear the TPnCE bit to 0 and then set the bits again.

2. The TPnIS3 to TPnIS0 bits are valid only in the free-running timer mode and the pulse width measurement mode. In all other modes, a capture operation is not possible.

(5) TMPn I/O control register 2 (TPnIOC2)

The TPnIOC2 register is an 8-bit register that controls the valid edge of the external event count input signal (TIOPn0 pin) and external trigger input signal (TIOPn0 pin).

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: TP0IOC2 FFFFF984H, TP1IOC2 FFFFF994H
TP2IOC2 FFFFF9A4H, TP3IOC2 FFFFF9B4H
TP4IOC2 FFFFFF14H

	7	6	5	4	3	2	1	0
TPnIOC2	0	0	0	0	TPnEES1	TPnEES0	TPnETS1	TPnETS0

(n = 0 to 4)

TPnEES1	TPnEES0	External event count input signal (TIOPn0 pin) valid edge setting
0	0	No edge detection (external event count invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TPnETS1	TPnETS0	External trigger input signal (TIOPn0 pin) valid edge setting
0	0	No edge detection (external trigger invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Cautions 1. Rewrite the TPnEES1, TPnEES0, TPnETS1, and TPnETS0 bits when the TPnCTL0.TPnCE bit = 0. (The same value can be written when the TPnCE bit = 1.) If rewriting was mistakenly performed, clear the TPnCE bit to 0 and then set the bits again.

2. The TPnEES1 and TPnEES0 bits are valid only when the TPnCTL1.TPnEEE bit = 1 or when the external event count mode (TPnCTL1.TPnMD2 to TPnCTL1.TPnMD0 bits = 001) has been set.

3. The TPnETS1 and TPnETS0 bits are valid only when the external trigger pulse output mode (TPnCTL1.TPnMD2 to TPnCTL1.TPnMD0 bits = 010) or the one-shot pulse output mode (TPnCTL1.TPnMD2 to TPnCTL1.TPnMD0 = 011) is set.

(6) TMPn option register 0 (TPnOPT0)

The TPnOPT0 register is an 8-bit register used to set the capture/compare operation and detect an overflow.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: TP0OPT0 FFFFF985H, TP1OPT0 FFFFF995H
TP2OPT0 FFFFF9A5H, TP3OPT0 FFFFF9B5H
TP4OPT0 FFFFFF15H

	7	6	5	4	3	2	1	<0>
TPnOPT0 (n = 0 to 4)	0	0	TPnCCS1	TPnCCS0	0	0	0	TPnOVF

TPnCCS1	TPnCCR1 register capture/compare selection
0	Compare register selected
1	Capture register selected
• This bit setting is valid only in the free-running timer mode.	

TPnCCS0	TPnCCR0 register capture/compare selection
0	Compare register selected
1	Capture register selected
• This bit setting is valid only in the free-running timer mode.	

TPnOVF	TMPn overflow detection flag
Set (1)	Overflow occurred
Reset (0)	TPnOVF bit 0 written or TPnCTL0.TPnCE bit = 0
• This bit is set when the 16-bit counter count value overflows from FFFFH to 0000H in the free-running timer mode or the pulse width measurement mode. • An interrupt request signal (INTTPnOV) is generated at the same time that the TPnOVF bit is set to 1. The INTTPnOV signal is not generated in modes other than the free-running timer mode and the pulse width measurement mode. • This bit is not cleared even when the TPnOVF bit or the TPnOPT0 register are read when the TPnOVF bit = 1. • This bit can be both read and written, but cannot be set to 1 by software. Writing 1 has no influence on the operation of TMPn.	

Cautions 1. Rewrite the TPnCCS1 and TPnCCS0 bits when the TPnCE bit = 0. (The same value can be written when the TPnCE bit = 1.) If rewriting was mistakenly performed, clear the TPnCE bit to 0 and then set the bits again.

2. Be sure to clear bits 1 to 3, 6, and 7 to "0".

(7) TMPn capture/compare register 0 (TPnCCR0)

The TPnCCR0 register can be used as a capture register or a compare register depending on the mode.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TPnOPT0.TPnCCS0 bit. In the pulse width measurement mode, the TPnCCR0 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TPnCCR0 register can be read or written during operation.

This register can be read or written in 16-bit units.

Reset sets this register to 0000H.

Caution Accessing the TPnCCR0 register is prohibited in the following statuses.

- **When the CPU operates with the subclock and the main clock oscillation is stopped**
- **When the CPU operates with the internal oscillation clock**

After reset: 0000H R/W Address: TP0CCR0 FFFFF986H, TP1CCR0 FFFFF996H
TP2CCR0 FFFFF9A6H, TP3CCR0 FFFFF9B6H
TP4CCR0 FFFFFF6CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPnCCR0																
(n = 0 to 4)																

(a) Function as compare register

The TPnCCR0 register can be rewritten even when the TPnCTL0.TPnCE bit = 1.

The set value of the TPnCCR0 register is transferred to the CCR0 buffer register. When the value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTPnCC0) is generated. If TIOPn0 pin output is enabled at this time, the output of the TIOPn0 pin is inverted.

When the TPnCCR0 register is used as a cycle register in the interval timer mode, external event count mode, external trigger pulse output mode, one-shot pulse output mode, or PWM output mode, the value of the 16-bit counter is cleared (0000H) if its count value matches the value of the CCR0 buffer register.

(b) Function as capture register

When the TPnCCR0 register is used as a capture register in the free-running timer mode, the count value of the 16-bit counter is stored in the TPnCCR0 register if the valid edge of the capture trigger input pin (TIOPn0 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TPnCCR0 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIOPn0) is detected.

Even if the capture operation and reading the TPnCCR0 register conflict, the correct value of the TPnCCR0 register can be read.

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 6-2. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write
One-shot pulse output	Compare register	Anytime write
PWM output	Compare register	Batch write
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement	Capture register	–

(8) TMPn capture/compare register 1 (TPnCCR1)

The TPnCCR1 register can be used as a capture register or a compare register depending on the mode.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TPnOPT0.TPnCCS1 bit. In the pulse width measurement mode, the TPnCCR1 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TPnCCR1 register can be read or written during operation.

This register can be read or written in 16-bit units.

Reset sets this register to 0000H.

Caution Accessing the TPnCCR1 register is prohibited in the following statuses.

- **When the CPU operates with the subclock and the main clock oscillation is stopped**
- **When the CPU operates with the internal oscillation clock**

After reset: 0000H R/W Address: TP0CCR1 FFFFF988H, TP1CCR1 FFFFF998H
TP2CCR1 FFFFF9A8H, TP3CCR1 FFFFF9B8H
TP4CCR1 FFFFFBAH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPnCCR1																
(n = 0 to 4)																

(a) Function as compare register

The TPnCCR1 register can be rewritten even when the TPnCTL0.TPnCE bit = 1.

The set value of the TPnCCR1 register is transferred to the CCR1 buffer register. When the value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTPnCC1) is generated. If TIOPn1 pin output is enabled at this time, the output of the TIOPn1 pin is inverted.

(b) Function as capture register

When the TPnCCR1 register is used as a capture register in the free-running timer mode, the count value of the 16-bit counter is stored in the TPnCCR1 register if the valid edge of the capture trigger input pin (TIOPn1 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TPnCCR1 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIOPn1) is detected.

Even if the capture operation and reading the TPnCCR1 register conflict, the correct value of the TPnCCR1 register can be read.

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 6-3. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write
One-shot pulse output	Compare register	Anytime write
PWM output	Compare register	Batch write
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement	Capture register	–

(9) TMPn counter read buffer register (TPnCNT)

The TPnCNT register is a read buffer register that can read the count value of the 16-bit counter.

If this register is read when the TPnCTL0.TPnCE bit = 1, the count value of the 16-bit timer can be read.

This register is read-only, in 16-bit units.

The value of the TPnCNT register is cleared to 0000H when the TPnCE bit = 0. If the TPnCNT register is read at this time, the value of the 16-bit counter (FFFFH) is not read, but 0000H is read.

The value of the TPnCNT register is cleared to 0000H after reset, as the TPnCE bit is cleared to 0.

Caution Accessing the TPnCNT register is prohibited in the following statuses.

- **When the CPU operates with the subclock and the main clock oscillation is stopped**
- **When the CPU operates with the internal oscillation clock**

After reset: 0000H R Address: TP0CNT FFFFF98AH, TP1CNT FFFFF99AH
TP2CNT FFFFF9AAH, TP3CNT FFFFF9BAH
TP4CNT FFFFFFBCH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPnCNT																

(n = 0 to 4)

(10) Input switch control register (ISC)

This register selects the pins for timer I/O and UART6 I/O.

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

Reset signal generation sets this register to 00H.

(1/2)

After reset: 00H R/W Address: FFFFFF4FH

- **78K0/DE2**

	7	6	5	4	3	2	1	0
ISC	ISC7	ISC6	0	0	ISC3	0	ISC1	ISC0[Note]

Note μPD78F0844 and 78F0845 only.

<R>

ISC7	UART60 pin selection control	
	TxD60	RxD60/INTPR60
0	TxD60 (P13)	RxD60/INTPR60 (P14)
1	<TxD60> (P71)	<RxD60/INTPR60> (P70)

ISC6	TIOP30 pin selection control
0	TIOP30 (P13)
1	<TIOP30> (P17)

ISC3	TMP2 input source (TIP21) selection control [For timer conjunction function of TMP]
0	TIOP21 (P06)
1	TMP3 output signal (TOP30)

ISC1	TMP2 input source (TIP20) selection control [For LIN reception operation of UART60]
0	TIOP20 (P14)
1	RxD60[Note]

Note Selected with ISC7.

ISC0	TMP4 input source (TIP40) selection control [For time stamp function of CAN]
0	TIOP40 (P00)
1	TSOUT

(2/2)

- **78K0/DF2**

	7	6	5	4	3	2	1	0
ISC	ISC7	ISC6	ISC5	0	ISC3	ISC2	ISC1	ISC0[Note]

Note μPD78F0846, 78F0847, 78F0848, and 78F0849 only.

<R>

ISC7	UART60 pin selection control	
	TxD60	RxD60/INTPR60
0	TxD60 (P13)	RxD60/INTPR60 (P14)
1	<TxD60> (P71)	<RxD60/INTPR60> (P70)

ISC6	TIOP30 pin selection control
0	TIOP30 (P13)
1	<TIOP30> (P17)

ISC5	TIOP20 pin selection control
0	TIOP20 (P14)
1	<TIOP20> (P77)

ISC3	TMP2 input source (TIP21) selection control [For timer conjunction function of TMP]
0	TIOP21 (P06)
1	TMP3 output signal (TOP30)

ISC2	TMP3 input source (TIP30) selection control [For LIN reception operation of UART61]
0	TIOP30[Note]
1	RxD61 (P11)

Note Selected with ISC6.

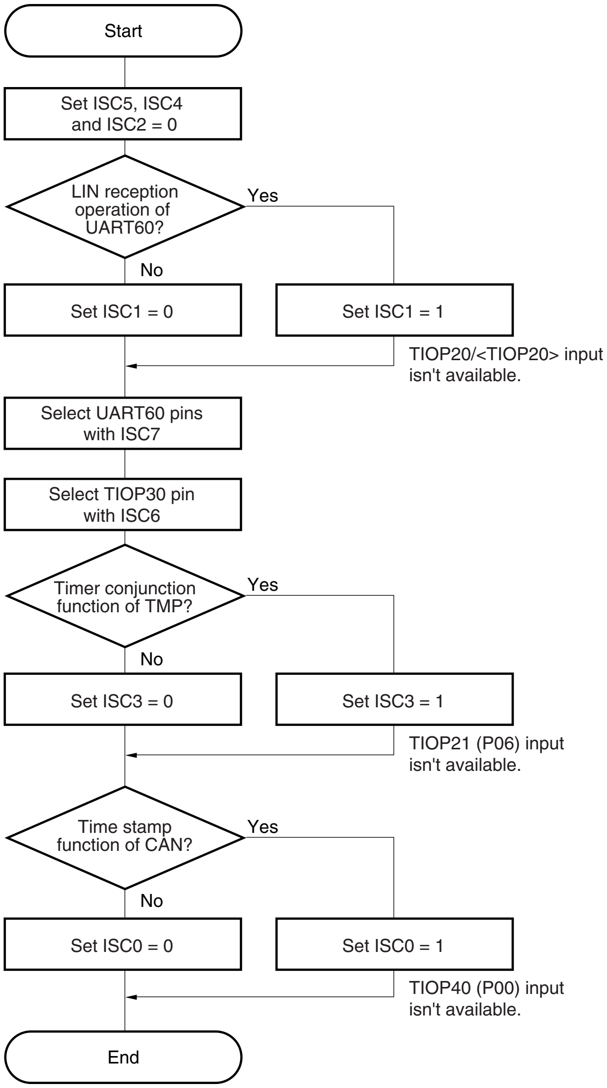
ISC1	TMP2 input source (TIP20) selection control [For LIN reception operation of UART60]
0	TIOP20 (P14)
1	RxD60[Note]

Note Selected with ISC7.

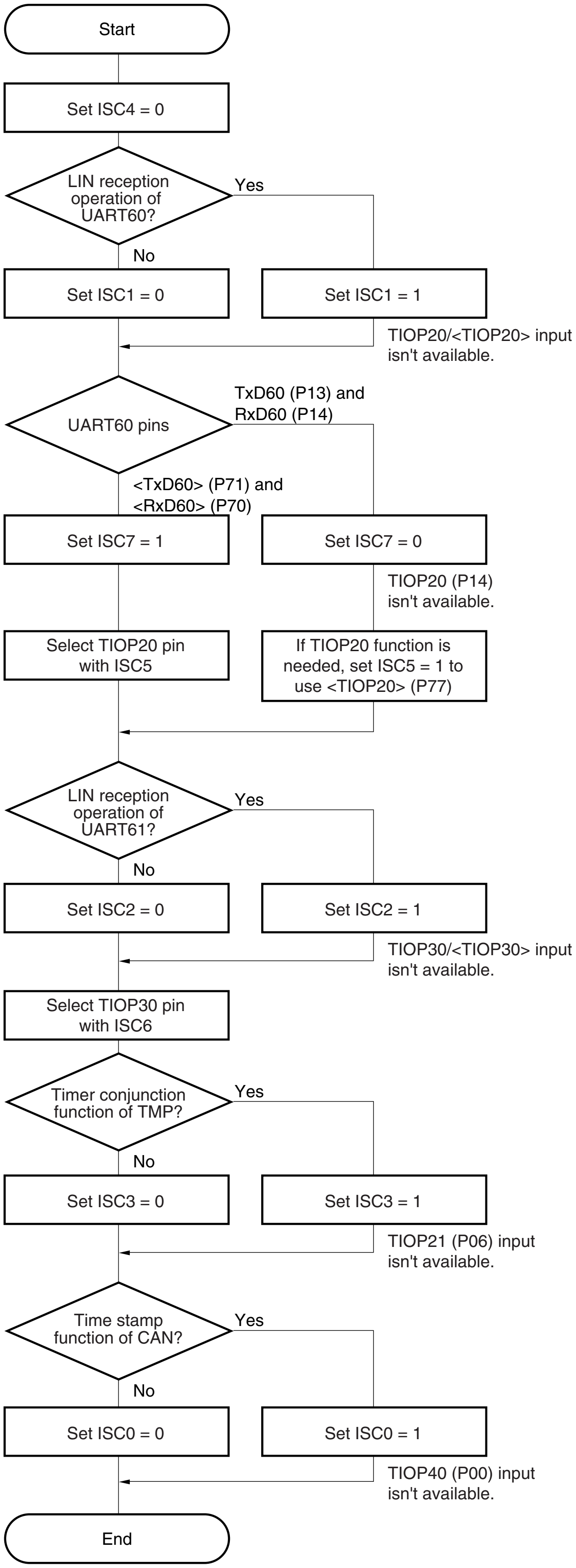
ISC0	TMP4 input source (TIP40) selection control [For time stamp function of CAN]
0	TIOP40 (P00)
1	TSOUT

<R> The following figures show how to select ISC register value.

78K0/DE2


Start
Set ISC5, ISC4 and ISC2 = 0
LIN reception operation of UART60?
Yes
No
Set ISC1 = 0
Set ISC1 = 1
TIOP20/<TIOP20> input isn't available.
Select UART60 pins with ISC7
Select TIOP30 pin with ISC6
Timer conjunction function of TMP?
Yes
No
Set ISC3 = 0
Set ISC3 = 1
TIOP21 (P06) input isn't available.
Time stamp function of CAN?
Yes
No
Set ISC0 = 0
Set ISC0 = 1
TIOP40 (P00) input isn't available.
End

78K0/DF2


Start
Set ISC4 = 0
LIN reception operation of UART60?
Yes
No
Set ISC1 = 0
Set ISC1 = 1
TIOP20/<TIOP20> input isn't available.
UART60 pins
TxD60 (P13) and RxD60 (P14)
<TxD60> (P71) and <RxD60> (P70)
Set ISC7 = 1
Set ISC7 = 0
TIOP20 (P14) isn't available.
Select TIOP20 pin with ISC5
If TIOP20 function is needed, set ISC5 = 1 to use <TIOP20> (P77)
LIN reception operation of UART61?
Yes
No
Set ISC2 = 0
Set ISC2 = 1
TIOP30/<TIOP30> input isn't available.
Select TIOP30 pin with ISC6
Timer conjunction function of TMP?
Yes
No
Set ISC3 = 0
Set ISC3 = 1
TIOP21 (P06) input isn't available.
Time stamp function of CAN?
Yes
No
Set ISC0 = 0
Set ISC0 = 1
TIOP40 (P00) input isn't available.
End

(11) TMPn input noise filter control registers 0 to 2 (TIPNF0 to TIPNF2)

The digital noise rejection settings are performed using the TIPNFn registers.

When digital noise rejection is selected, the sampling clock for digital sampling can be selected from among f_{PRS}, $f_{PRS}/2$, $f_{PRS}/2^2$, $f_{PRS}/2^3$ (or $f_{PRS}/2^8$), $f_{PRS}/2^4$ (or $f_{PRS}/2^9$), $f_{PRS}/2^5$, $f_{PRS}/2^6$, and $f_{PRS}/2^7$ (or f_{OSC}). Sampling is performed two times.

These registers can be read or written in 1-bit or 8-bit units.

Reset sets these registers to 00H.

Cautions 1. To use the noise filter, proceed as follows:

<1> Enable the noise filter by setting TIPNFn.

<2> Wait for 3 sampling clocks.

<3> Enable the timer operation by setting TPnCTL0.

2. To stop the operation TMPn and noise filter, proceed as follows:

<1> Disable the timer operation by setting TPnCTL0.

<2> Disable the noise filter by setting TIPNFn.

3. After the sampling clock has been changed, it takes 3 sampling clocks to initialize the digital noise rejection. If one of the external event counter mode, external trigger plus output mode, and capture trigger function is used, the operation of TMPn should be enabled after over 3 sampling clocks.

(1/2)

After reset: 00H R/W Address: FFFFFF54H

	7	6	5	4	3	2	1	0
TIPNF0	TIP1NFEN	TIP1NF2	TIP1NF1	TIP1NF0	TIP0NFEN	TIP0NF2	TIP0NF1	TIP0NF0

After reset: 00H R/W Address: FFFFFF59H

	7	6	5	4	3	2	1	0
TIPNF1	TIP3NFEN	TIP3NF2	TIP3NF1	TIP3NF0	TIP2NFEN	TIP2NF2	TIP2NF1	TIP2NF0

After reset: 00H R/W Address: FFFFFF69H

	7	6	5	4	3	2	1	0
TIPNF2	0	0	0	0	TIP4NFEN	TIP4NF2	TIP4NF1	TIP4NF0

TIPnNFEN	Digital noise filter operation (n = 0 to 4)
0	TMPn noise filter disabled
1	TMPn noise filter enabled

(2/2)

TPnNF2	TPnNF1	TPnNF0	Sampling check for TIPn0 and TIPn1		
			n = 0, 1, 4 (TMP0, TMP1, TMP4)	n = 2 (TMP2)	n = 3 (TMP3)
0	0	0	f_{PRS}		
0	0	1	$f_{PRS}/2$		
0	1	0	$f_{PRS}/2^{2}$		
0	1	1	$f_{PRS}/2^{3}$	$f_{PRS}/2^{8}$	$f_{PRS}/2^{3}$
1	0	0	$f_{PRS}/2^{4}$	$f_{PRS}/2^{9}$	$f_{PRS}/2^{4}$
1	0	1	$f_{PRS}/2^{5}$		
1	1	0	$f_{PRS}/2^{6}$		
1	1	1	$f_{PRS}/2^{7}$		f_{OSC}

Caution Select the same sample clock as selected in TPnCKS2 to TPnCKS0.

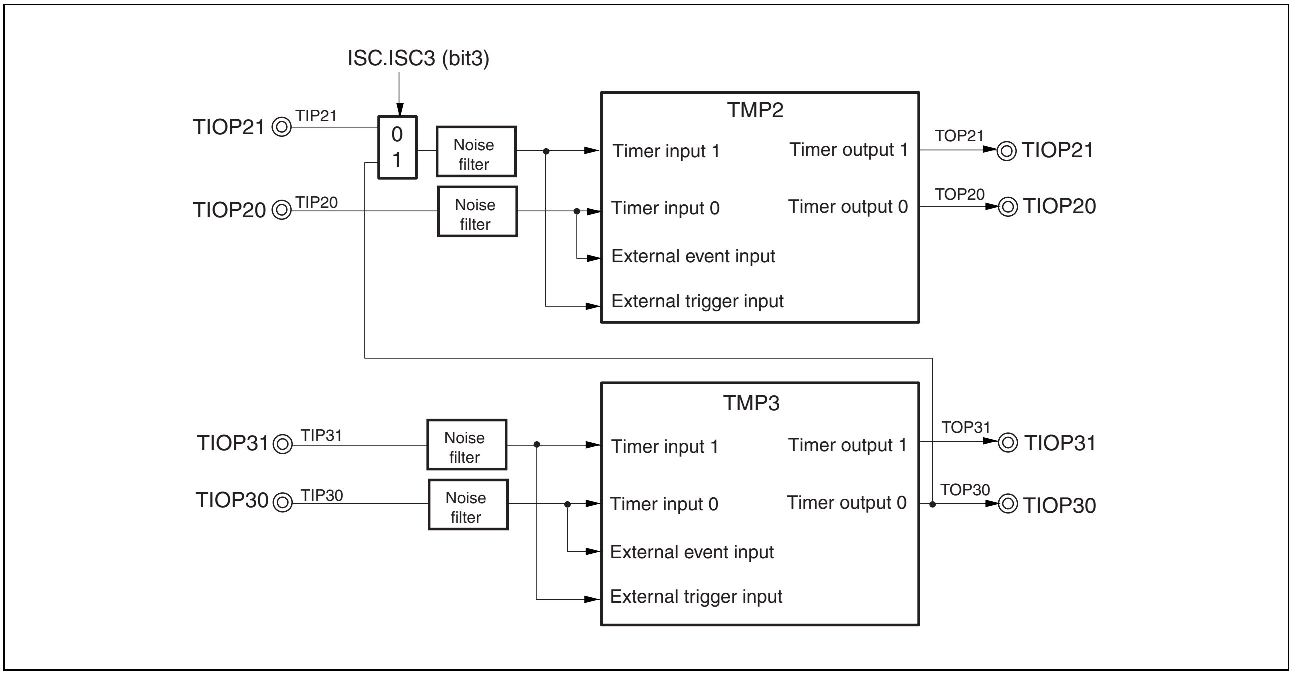
Remark f_{PRS}: Peripheral hardware clock frequency

f_{OSC}: Internal low-speed oscillation clock frequency

6.5 Timer Conjunction Function

TMP3 output of TIOP30 is connected to TMP2 input of TIOP21. This function is controlled with the bit3 of the input switch control register (ISC).

Figure 6-2. Timer Conjunction Function



Caution **In the external event counter mode, the pulse must not be input before the TMP operation enabled.**

Remark For low frequency input pulse, selecting the capture function is recommended. For high frequency input pulse, setting the TPnEEE bit to 1 is recommended.

6.6 Timer Synchronous Operation Function

Timer P has a timer synchronized operation function. TMP0 as a master timer and TMP1 to TMP4 as slave timers can be synchronized.

Cautions 1. The tuned operation mode is enabled or disabled by the TPnSYE bit of the TPnCTL1 register. For TMP0, TMP1 to TMP4 can be specified as slaves.

2. Set the tuned operation mode using the following procedure:

<1> Set the TPnSYE bit of the TPnCTL1 register of the slave timer to enable the tuned operation.
Set the TPnMD2 to TPnMD0 bits of the TPnCTL1 register of the slave timer to the free-running mode.

<2> Set the timer mode by using the TPnMD2 to TPnMD0 bits of the TPnCTL1 register. At this time, do not set the TPnSYE bit of the TPnCTL1 register of the master timer.

<3> Set the compare register value of the master and slave timers.

<4> Set the TPnCE bit of the TPnCTL0 register of the slave timer to enable operation on the internal operating clock.

<5> Set the TPnCE bit of the TPnCTL0 register of the master timer to enable operation on the internal operating clock.

The free-running mode and PWM mode can be used for TMP0 as a master timer in the tuned operation mode as follows:

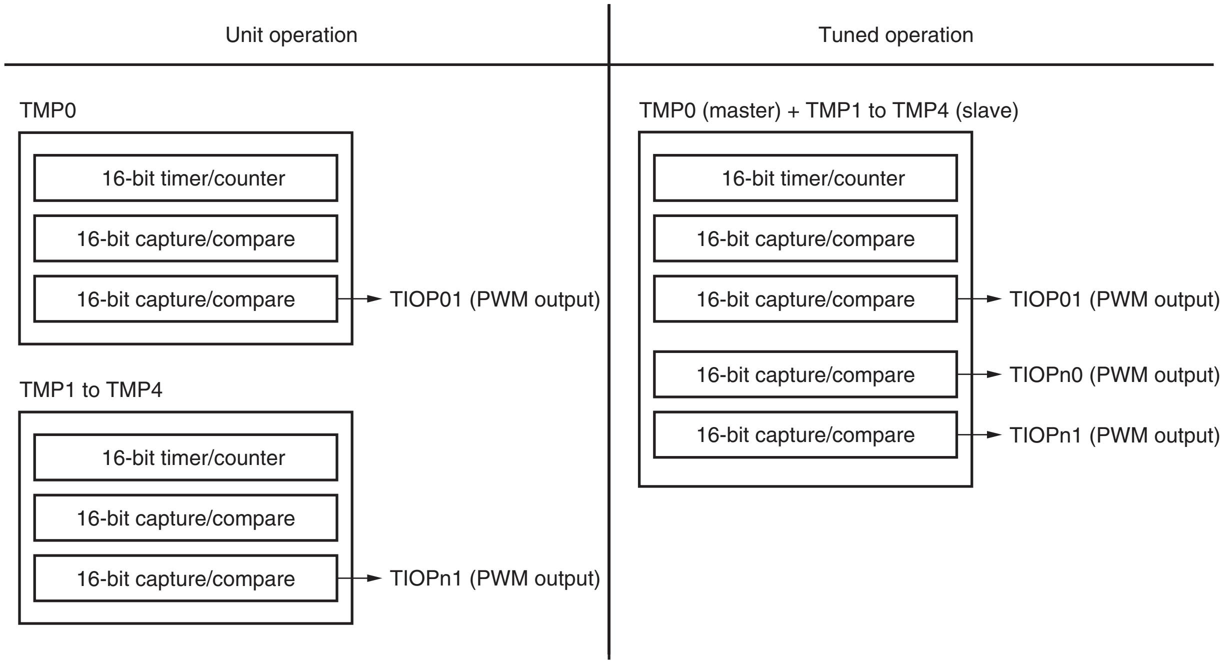
Table 6-4. Timer Output Functions

Tuned Channel	Timer	Pin	Free-running Mode		PWM Mode	
			Tuning OFF	Tuning ON	Tuning OFF	Tuning ON
Ch0	TMP0 (master)	TIOP00	PPG	←	Toggle	←
		TIOP01			PWM	←
	TMP1 (slave)	TIOP10			Toggle	PWM
		TIOP11			PWM	←
	TMP2 (slave)	TIOP20			Toggle	PWM
		TIOP21			PWM	←
	TMP3 (slave)	TIOP30			Toggle	PWM
		TIOP31			PWM	←
	TMP4 (slave)	TIOP40			Toggle	PWM
		TIOP41			PWM	←

Remark The timing of transmitting data from the compare register of the master timer to the compare register of the slave timer is as follows.

PPG:	CPU write timing
Toggle, PWM:	Timing at which timer counter and compare register match TIOPn0 (n = 0 to 4)

Figure 6-3. Tuned Operation Image (TMP0 to TMP4)

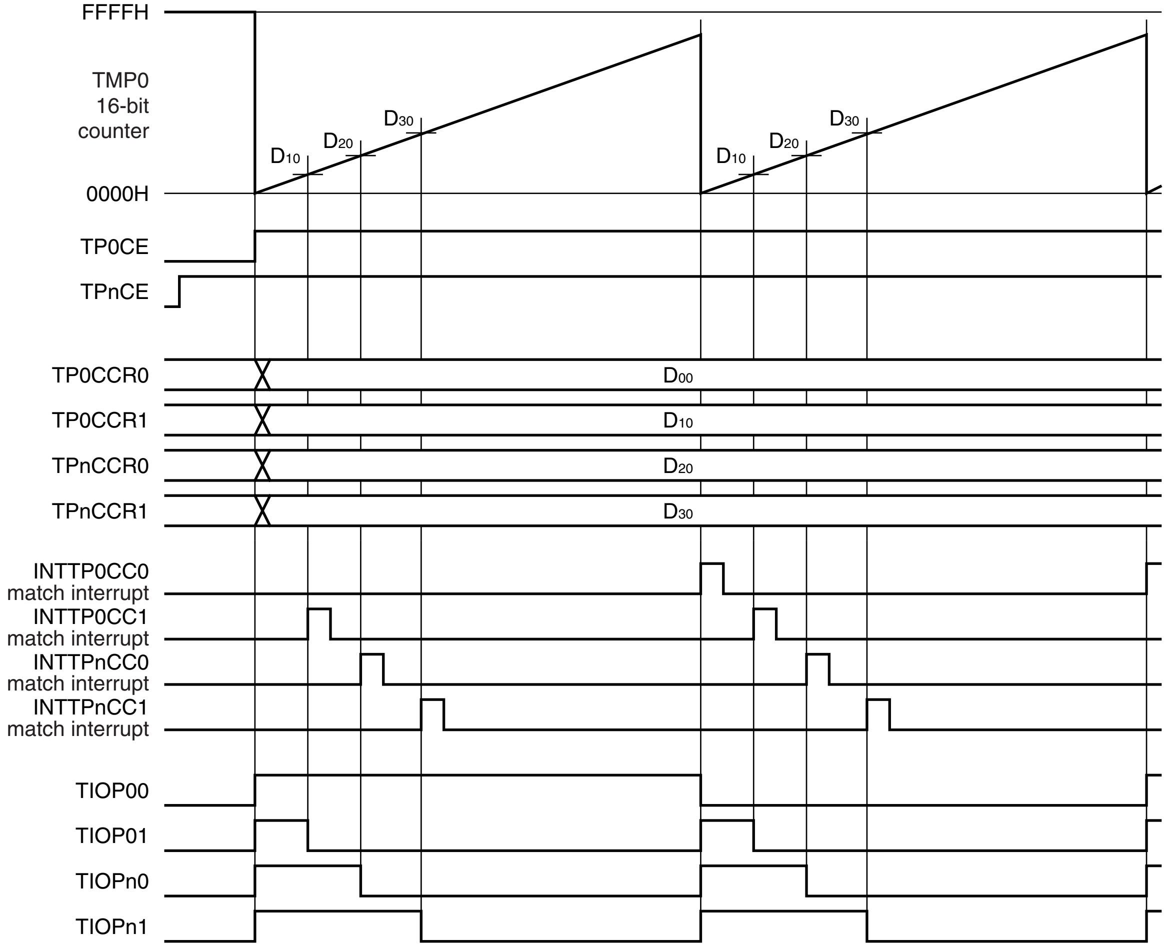


Five PWM outputs are available
when PWM is operated as a single unit.

Nine PWM outputs are available
when PWM is operated in tuned operation mode.

Remark n = 1 to 4

Figure 6-4. Basic Operation Timing of Tuned PWM Function (TMP0 to TMP4)



Remark n = 1 to 4

6.7 Operation

TMPn can perform the following operations.

Operation	TPnCTL1.TPnEST Bit (Software Trigger Bit)	TIOPn0 Pin (External Trigger Input)	Capture/Compare Register Setting	Compare Register Write
Interval timer mode	Invalid	Invalid	Compare only	Anytime write
External event count mode[Note 1]	Invalid	Invalid	Compare only	Anytime write
External trigger pulse output mode[Note 2]	Valid	Valid	Compare only	Batch write
One-shot pulse output mode[Note 2]	Valid	Valid	Compare only	Anytime write
PWM output mode	Invalid	Invalid	Compare only	Batch write
Free-running timer mode	Invalid	Invalid	Switching enabled	Anytime write
Pulse width measurement mode[Note 2]	Invalid	Invalid	Capture only	Not applicable

Notes 1. To use the external event count mode, specify that the valid edge of the TIOPn0 pin capture trigger input is not detected (by clearing the TPnIOC1.TPnIS1 and TPnIOC1.TPnIS0 bits to “00”).

2. When using the external trigger pulse output mode, one-shot pulse output mode, and pulse width measurement mode, select the internal clock as the count clock (by clearing the TPnCTL1.TPnEEE bit to 0).

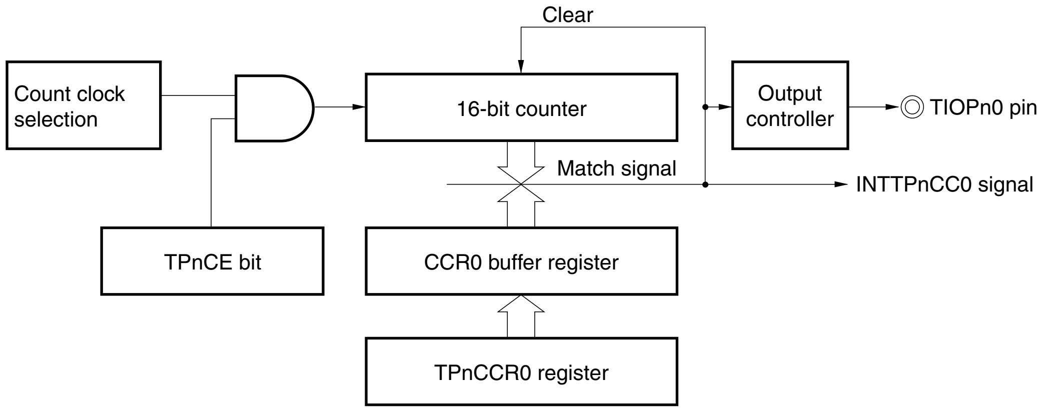
Remark n = 0 to 4

6.7.1 Interval timer mode (TPnMD2 to TPnMD0 bits = 000)

In the interval timer mode, an interrupt request signal (INTTPnCC0) is generated at the specified interval if the TPnCTL0.TPnCE bit is set to 1. A square wave whose half cycle is equal to the interval can be output from the TIOPn0 pin.

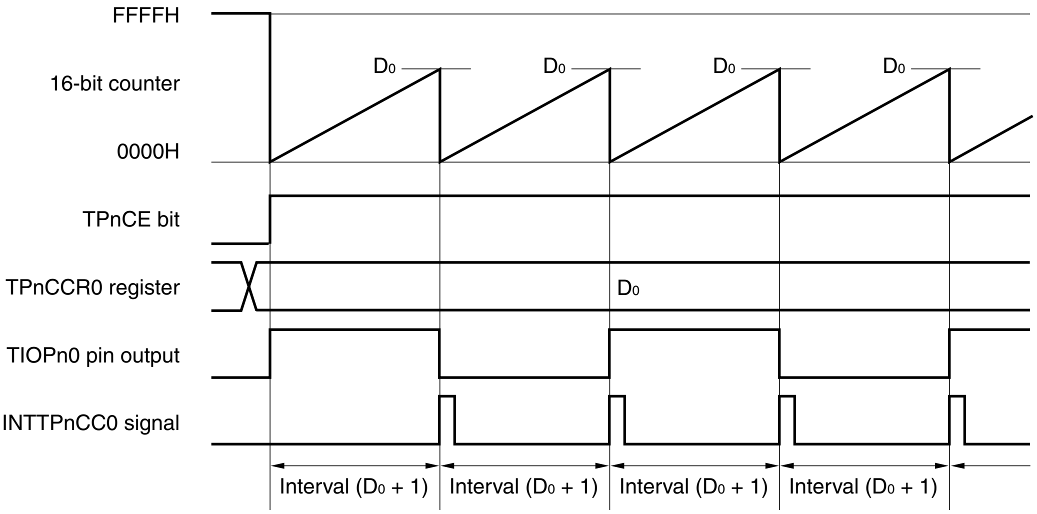
Usually, the TPnCCR1 register is not used in the interval timer mode.

Figure 6-5. Configuration of Interval Timer



Remark n = 0 to 4

Figure 6-6. Basic Timing of Operation in Interval Timer Mode



Remark n = 0 to 4

When the TPnCE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H in synchronization with the count clock, and the counter starts counting. At this time, the output of the TIOPn0 pin is inverted. Additionally, the set value of the TPnCCR0 register is transferred to the CCR0 buffer register.

When the count value of the 16-bit counter matches the value of the CCR0 buffer register, the 16-bit counter is cleared to 0000H, the output of the TIOPn0 pin is inverted, and a compare match interrupt request signal (INTTPnCC0) is generated.

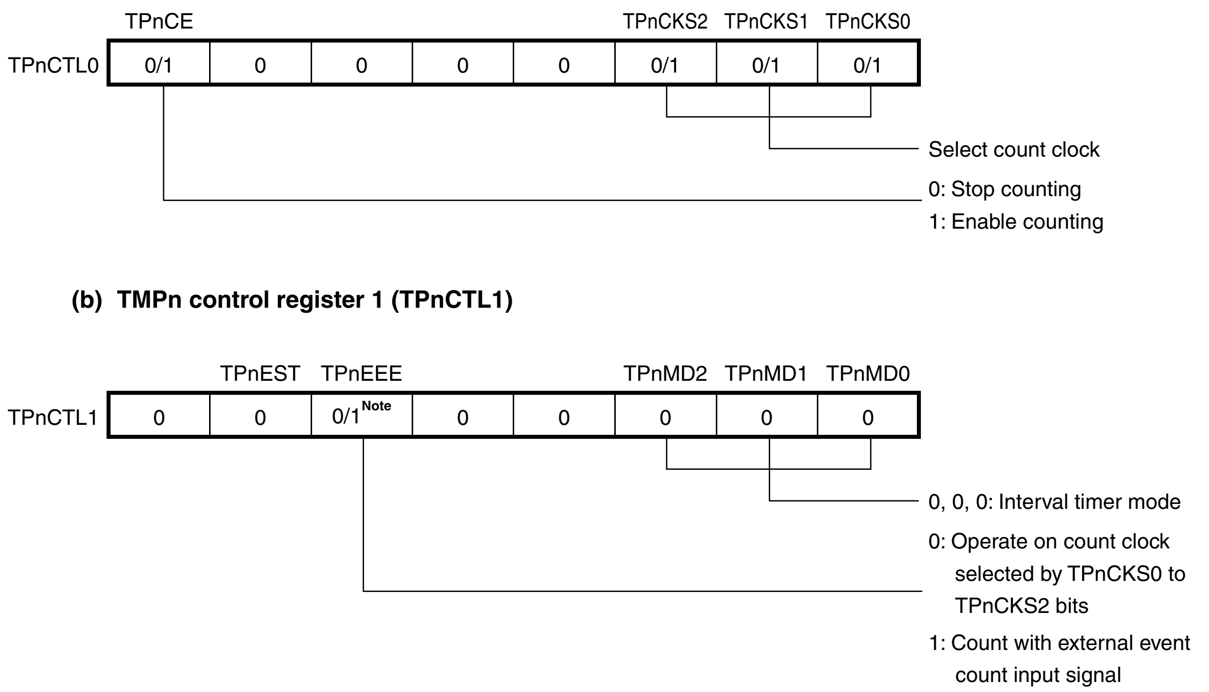
The interval can be calculated by the following expression.

Interval = (Set value of TPnCCR0 register + 1) × Count clock cycle

Remark n = 0 to 4

Figure 6-7. Register Setting for Interval Timer Mode Operation (1/2)

(a) TMPn control register 0 (TPnCTL0)



(b) TMPn control register 1 (TPnCTL1)

TPnEST
TPnEEE
TPnMD2
TPnMD1
TPnMD0
TPnCTL1
0
0
0/1Note
0
0
0
0
0
0, 0, 0: Interval timer mode
0: Operate on count clock selected by TPnCKS0 to TPnCKS2 bits
1: Count with external event count input signal

Note This bit can be set to 1 only when the interrupt request signals (INTTPnCC0 and INTTPnCC1) are masked by the interrupt mask flags (TPnCCMK0 and TPnCCMK1) and timer output (TIOPn1) is performed at the same time. However, set the TPnCCR0 and TPnCCR1 registers to the same value (see **6.7.1 (2) (d) Operation of TPnCCR1 register**).

Figure 6-7. Register Setting for Interval Timer Mode Operation (2/2)

(c) TMPn I/O control register 0 (TPnIOC0)

					TPnOL1	TPnOE1	TPnOL0	TPnOE0
TPnIOC0	0	0	0	0	0/1	0/1	0/1	0/1

TPnOE0:
0: Disable TIOPn0 pin output
1: Enable TIOPn0 pin output

TPnOL0:
Setting of output level with operation of TIOPn0 pin disabled
0: Low level
1: High level

TPnOE1:
0: Disable TIOPn1 pin output
1: Enable TIOPn1 pin output

TPnOL1:
Setting of output level with operation of TIOPn1 pin disabled
0: Low level
1: High level

(d) TMPn counter read buffer register (TPnCNT)

By reading the TPnCNT register, the count value of the 16-bit counter can be read.

(e) TMPn capture/compare register 0 (TPnCCR0)

If the TPnCCR0 register is set to D_0, the interval is as follows.

Interval = (D_0 + 1) × Count clock cycle

(f) TMPn capture/compare register 1 (TPnCCR1)

Usually, the TPnCCR1 register is not used in the interval timer mode. However, the set value of the TPnCCR1 register is transferred to the CCR1 buffer register. A compare match interrupt request signal (INTTPnCC1) is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

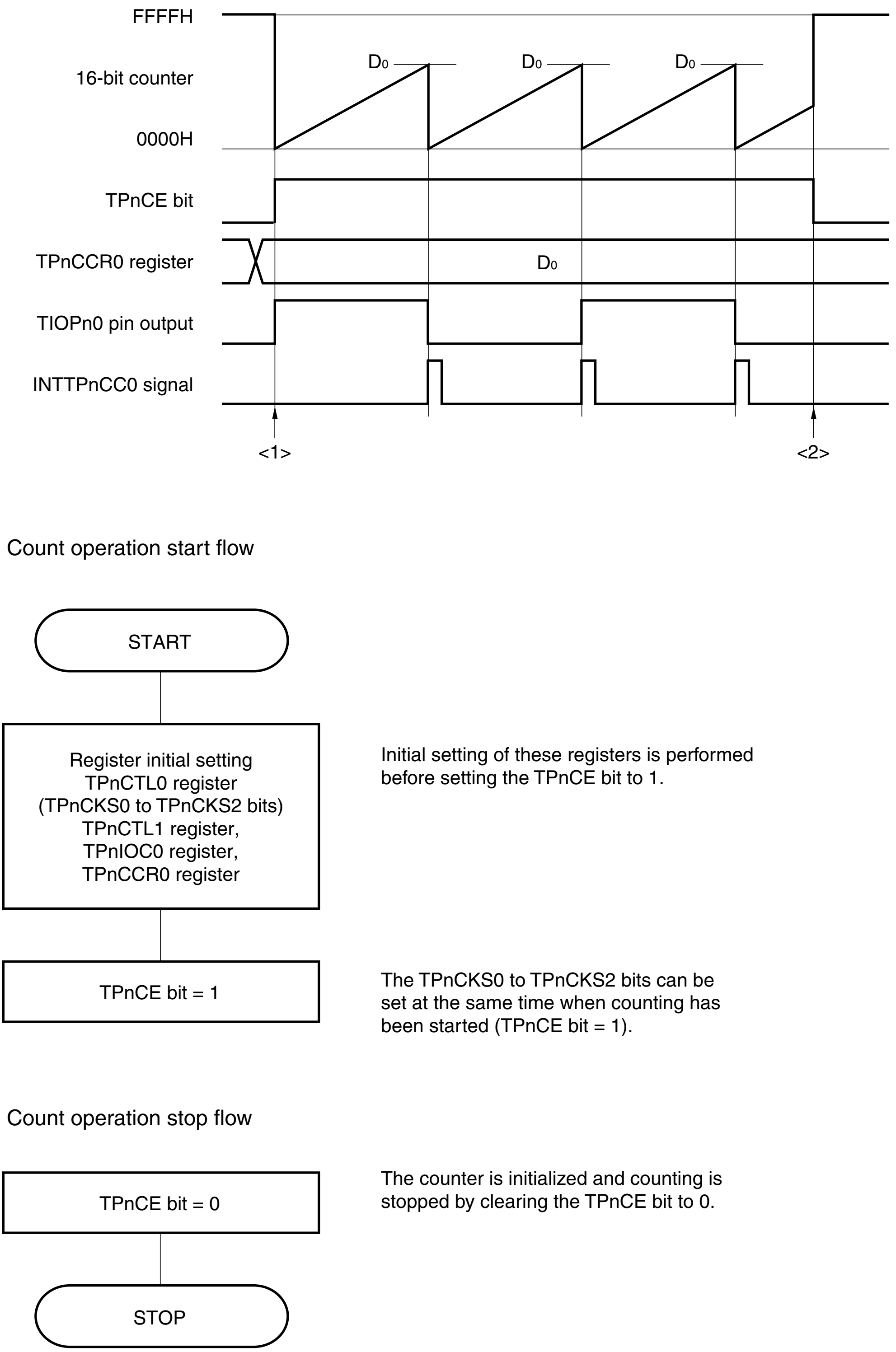
Therefore, mask the interrupt request by using the corresponding interrupt mask flag (TPnCCMK1).

Remarks 1. TMPn I/O control register 1 (TPnIOC1), TMPn I/O control register 2 (TPnIOC2), and TMPn option register 0 (TPnOPT0) are not used in the interval timer mode.

2. n = 0 to 4

(1) Interval timer mode operation flow

Figure 6-8. Software Processing Flow in Interval Timer Mode



<1> Count operation start flow

Initial setting of these registers is performed before setting the TPnCE bit to 1.

The TPnCKS0 to TPnCKS2 bits can be set at the same time when counting has been started (TPnCE bit = 1).

<2> Count operation stop flow

The counter is initialized and counting is stopped by clearing the TPnCE bit to 0.

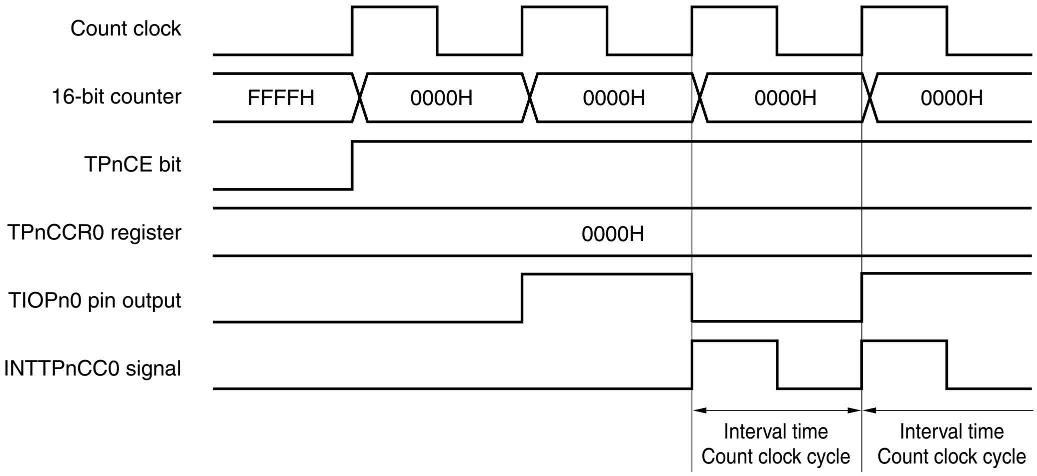
Remark n = 0 to 4

(2) Interval timer mode operation timing

(a) Operation if TPnCCR0 register is set to 0000H

If the TPnCCR0 register is set to 0000H, the INTTPnCC0 signal is generated at each count clock subsequent to the first count clock, and the output of the TIOPn0 pin is inverted.

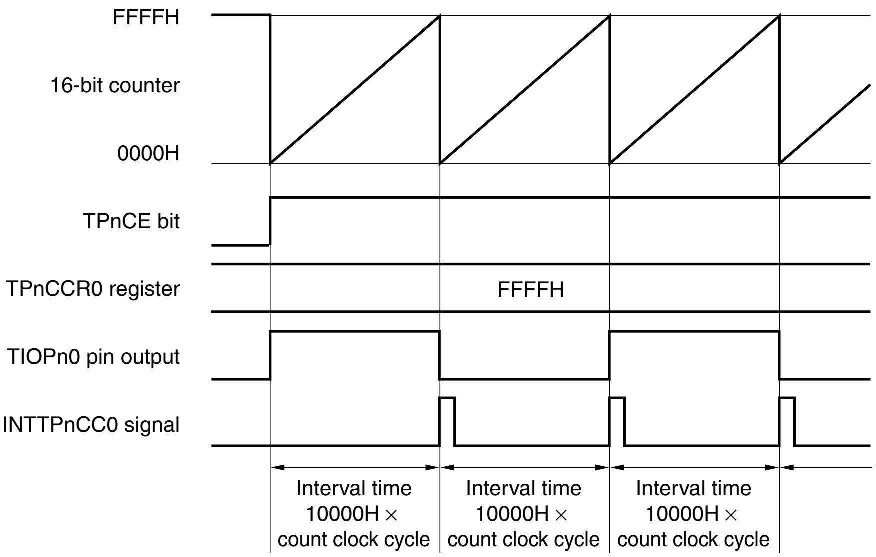
The value of the 16-bit counter is always 0000H.



Remark n = 0 to 4

(b) Operation if TPnCCR0 register is set to FFFFH

If the TPnCCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH. The counter is cleared to 0000H in synchronization with the next count-up timing. The INTTPnCC0 signal is generated and the output of the TIOPn0 pin is inverted. At this time, an overflow interrupt request signal (INTTPnOV) is not generated, nor is the overflow flag (TPnOPT0.TPnOVF bit) set to 1.

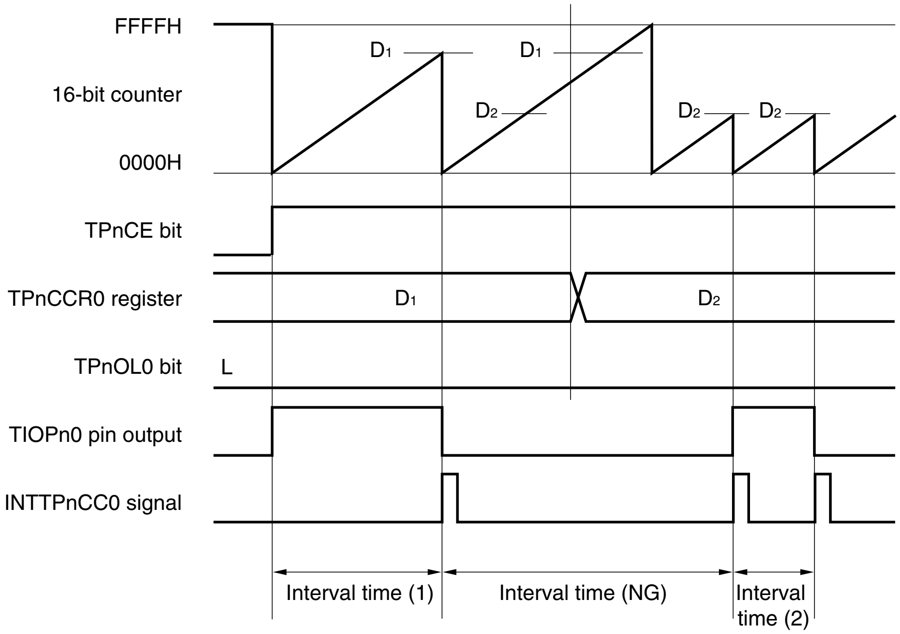


Remark n = 0 to 4

(c) Notes on rewriting TPnCCR0 register

To change the value of the TPnCCR0 register to a smaller value, stop counting once and then change the set value.

If the value of the TPnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow.



Remarks 1. Interval time (1): (D_1 + 1) × Count clock cycle
Interval time (NG): (10000H + D_2 + 1) × Count clock cycle
Interval time (2): (D_2 + 1) × Count clock cycle

2. n = 0 to 4

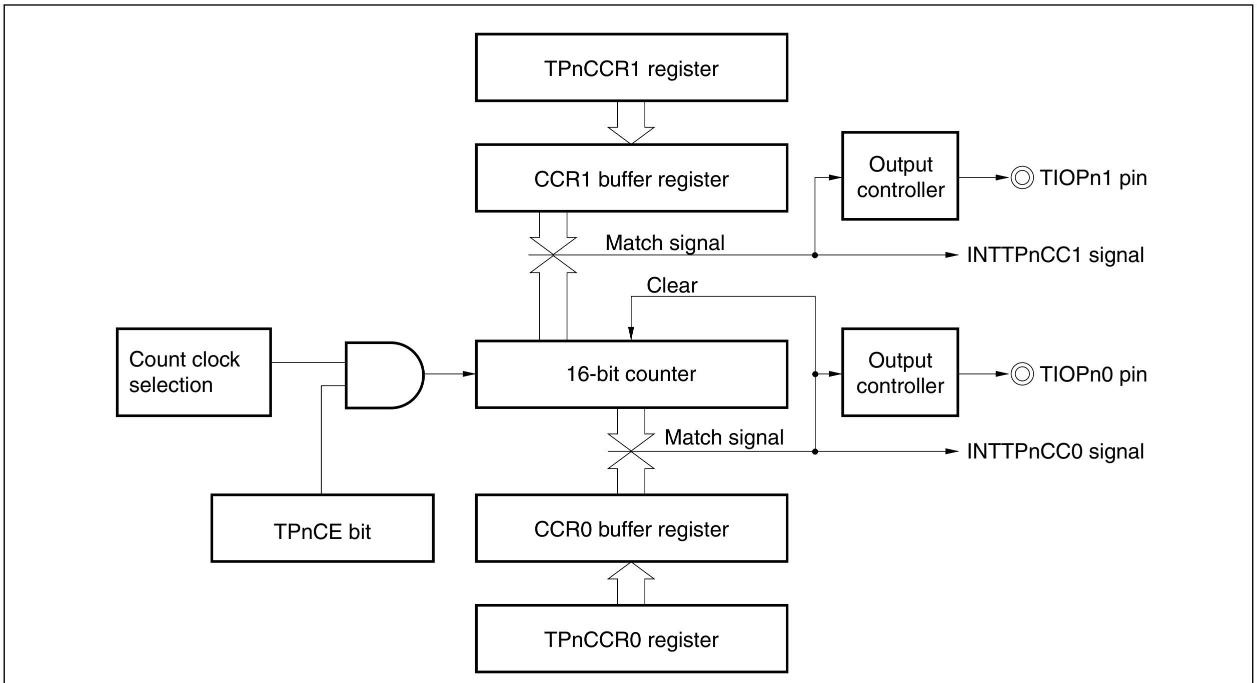
If the value of the TPnCCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TPnCCR0 register has been rewritten. Consequently, the value of the 16-bit counter that is compared is D_2.

Because the count value has already exceeded D_2, however, the 16-bit counter counts up to FFFFH, overflows, and then counts up again from 0000H. When the count value matches D_2, the INTTPnCC0 signal is generated and the output of the TIOPn0 pin is inverted.

Therefore, the INTTPnCC0 signal may not be generated at the interval time "(D_1 + 1) × Count clock cycle" or "(D_2 + 1) × Count clock cycle" originally expected, but may be generated at an interval of "(10000H + D_2 + 1) × Count clock period".

(d) Operation of TPnCCR1 register

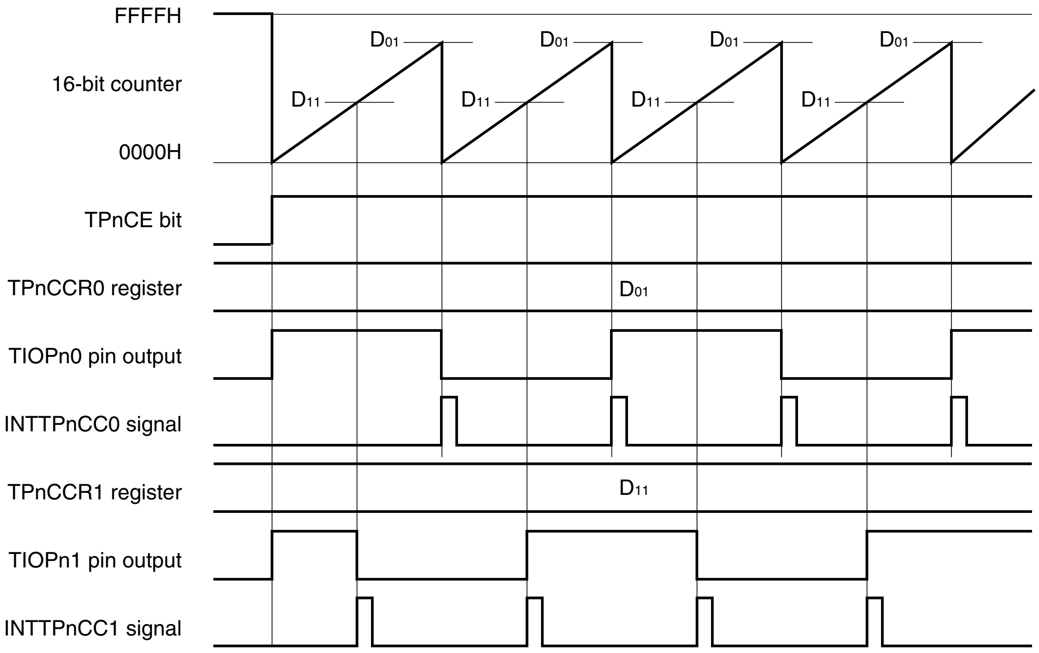
Figure 6-9. Configuration of TPnCCR1 Register



Remark n = 0 to 4

If the set value of the TPnCCR1 register is less than the set value of the TPnCCR0 register, the INTTPnCC1 signal is generated once per cycle. At the same time, the output of the TIOPn1 pin is inverted. The TIOPn1 pin outputs a square wave with the same cycle as that output by the TIOPn0 pin.

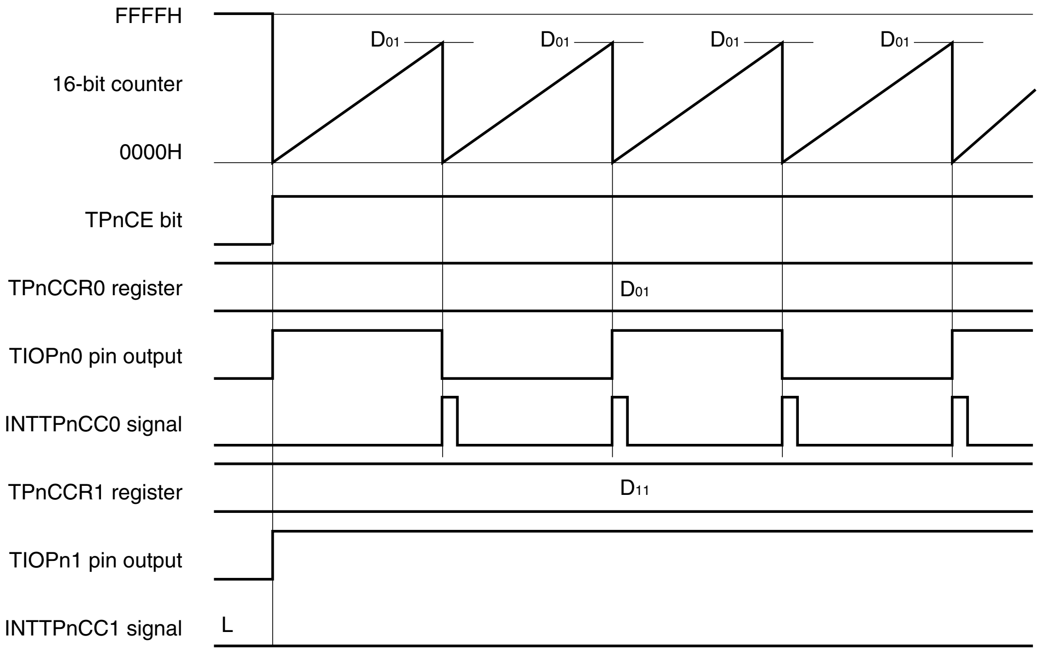
Figure 6-10. Timing Chart When $D_{01} \geq D_{11}$



Remark n = 0 to 4

If the set value of the TPnCCR1 register is greater than the set value of the TPnCCR0 register, the count value of the 16-bit counter does not match the value of the TPnCCR1 register. Consequently, the INTTPnCC1 signal is not generated, nor is the output of the TIOPn1 pin changed.

Figure 6-11. Timing Chart When D_{01} < D_{11}



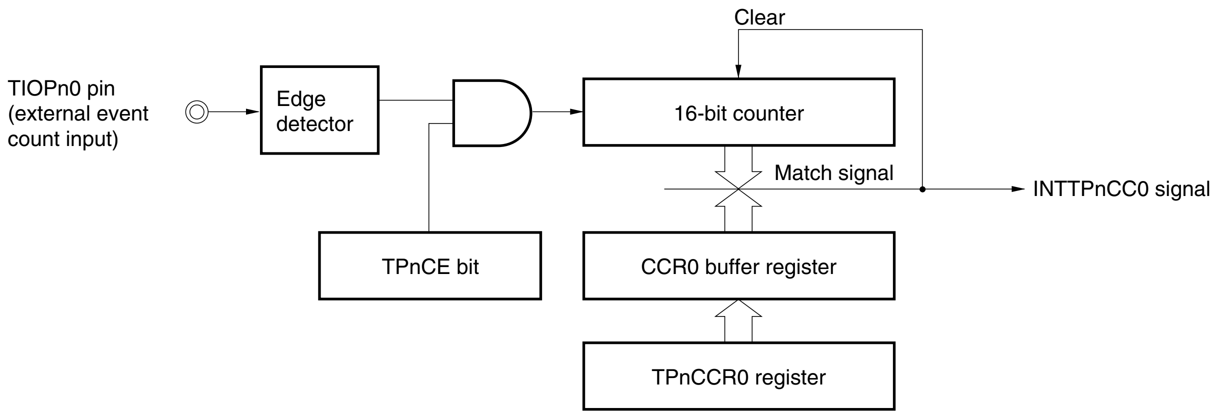
Remark n = 0 to 4

6.7.2 External event count mode (TPnMD2 to TPnMD0 bits = 001)

In the external event count mode, the valid edge of the external event count input is counted when the TPnCTL0.TPnCE bit is set to 1, and an interrupt request signal (INTTPnCC0) is generated each time the specified number of edges have been counted. The TIOPn0 pin cannot be used.

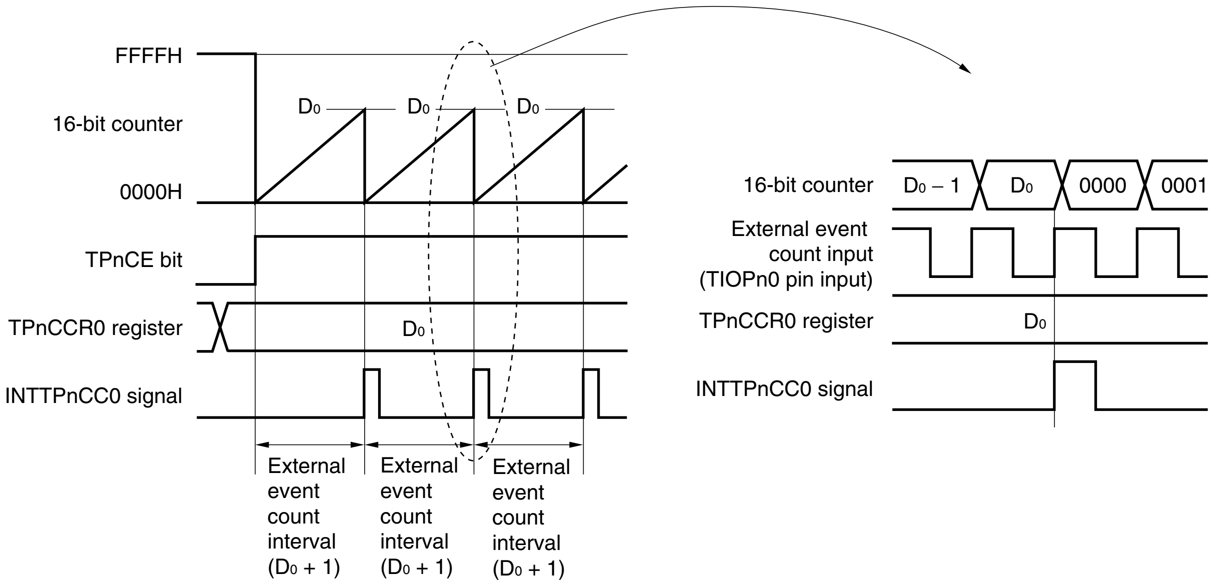
Usually, the TPnCCR1 register is not used in the external event count mode.

Figure 6-12. Configuration in External Event Count Mode



Remark n = 0 to 4

Figure 6-13. Basic Timing in External Event Count Mode



Remarks 1. This figure shows the basic timing when the rising edge is specified as the valid edge of the external event count input.

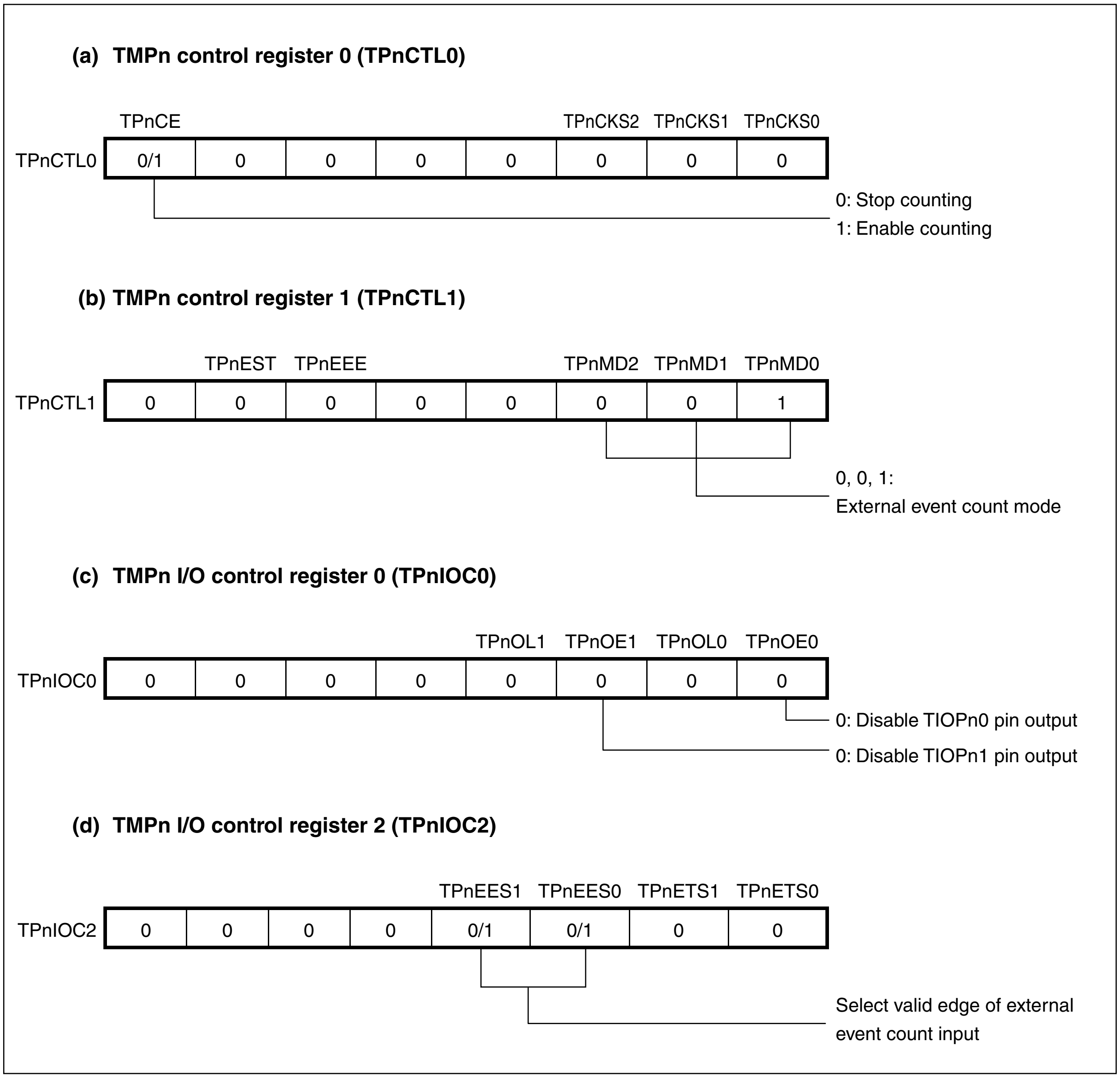
2. n = 0 to 4

When the TPnCE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H. The counter counts each time the valid edge of external event count input is detected. Additionally, the set value of the TPnCCR0 register is transferred to the CCR0 buffer register.

When the count value of the 16-bit counter matches the value of the CCR0 buffer register, the 16-bit counter is cleared to 0000H, and a compare match interrupt request signal (INTTPnCC0) is generated.

The INTTPnCC0 signal is generated each time the valid edge of the external event count input has been detected (set value of TPnCCR0 register + 1) times.

Figure 6-14. Register Setting for Operation in External Event Count Mode (1/2)



(a) TMPn control register 0 (TPnCTL0)

	TPnCE					TPnCKS2	TPnCKS1	TPnCKS0
TPnCTL0	0/1	0	0	0	0	0	0	0

0: Stop counting
1: Enable counting

(b) TMPn control register 1 (TPnCTL1)

		TPnEST	TPnEEE			TPnMD2	TPnMD1	TPnMD0
TPnCTL1	0	0	0	0	0	0	0	1

0, 0, 1:
External event count mode

(c) TMPn I/O control register 0 (TPnIOC0)

					TPnOL1	TPnOE1	TPnOL0	TPnOE0
TPnIOC0	0	0	0	0	0	0	0	0

0: Disable TIOPn0 pin output
0: Disable TIOPn1 pin output

(d) TMPn I/O control register 2 (TPnIOC2)

					TPnEES1	TPnEES0	TPnETS1	TPnETS0
TPnIOC2	0	0	0	0	0/1	0/1	0	0

Select valid edge of external
event count input

Figure 6-14. Register Setting for Operation in External Event Count Mode (2/2)

(e) TMPn counter read buffer register (TPnCNT)

The count value of the 16-bit counter can be read by reading the TPnCNT register.

(f) TMPn capture/compare register 0 (TPnCCR0)

If D_0 is set to the TPnCCR0 register, the counter is cleared and a compare match interrupt request signal (INTTPnCC0) is generated when the number of external event counts reaches (D_0 + 1).

(g) TMPn capture/compare register 1 (TPnCCR1)

Usually, the TPnCCR1 register is not used in the external event count mode. However, the set value of the TPnCCR1 register is transferred to the CCR1 buffer register. When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTPnCC1) is generated.

Therefore, mask the interrupt signal by using the interrupt mask flag (TPnCCMK1).

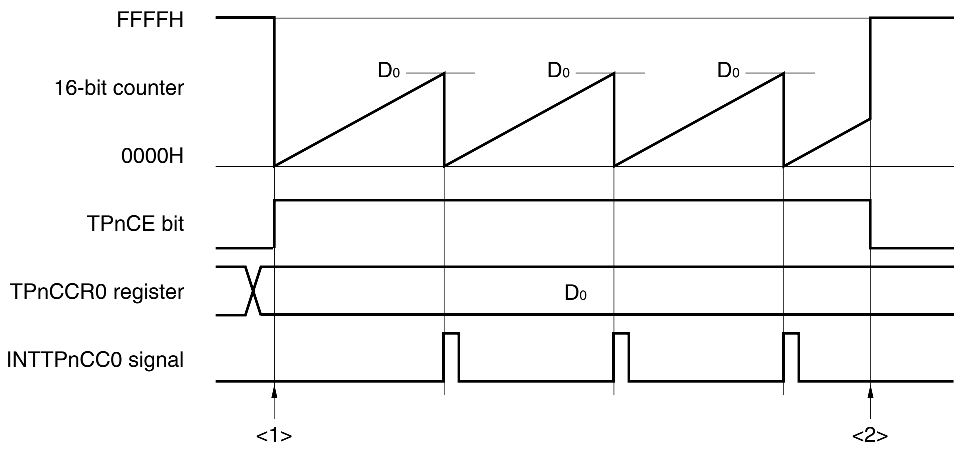
Caution When an external clock is used as the count clock, the external clock can be input only from the TIOPn0 pin. At this time, set the TPnIOC1.TPnIS1 and TPnIOC1.TPnIS0 bits to 00 (capture trigger input (TIOPn0 pin): no edge detection).

Remarks 1. TMPn I/O control register 1 (TPnIOC1) and TMPn option register 0 (TPnOPT0) are not used in the external event count mode.

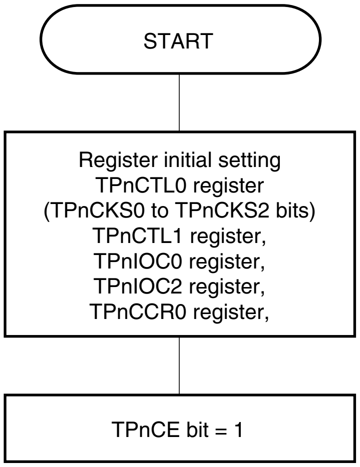
2. n = 0 to 4

(1) External event count mode operation flow

Figure 6-15. Flow of Software Processing in External Event Count Mode



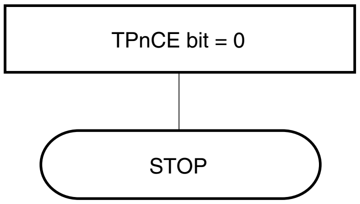
<1> Count operation start flow



Initial setting of these registers is performed before setting the TPnCE bit to 1.

The TPnCKS0 to TPnCKS2 bits can be set at the same time when counting has been started (TPnCE bit = 1).

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TPnCE bit to 0.

Remark n = 0 to 4

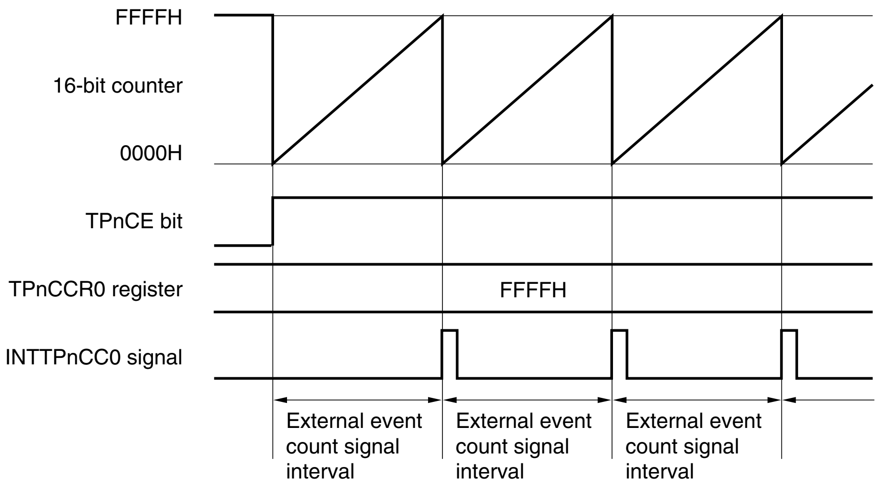
(2) Operation timing in external event count mode

Cautions 1. In the external event count mode, do not set the TPnCCR0 register to 0000H.

2. In the external event count mode, use of the timer output is disabled. If performing timer output using external event count input, set the interval timer mode, and select the operation enabled by the external event count input for the count clock (TPnCTL1.TPnMD2 to TPnCTL1.TPnMD0 bits = 000, TPnCTL1.TPnEEE bit = 1).

(a) Operation if TPnCCR0 register is set to FFFFH

If the TPnCCR0 register is set to FFFFH, the 16-bit counter counts to FFFFH each time the valid edge of the external event count signal has been detected. The 16-bit counter is cleared to 0000H in synchronization with the next count-up timing, and the INTTPnCC0 signal is generated. At this time, the TPnOPT0.TPnOVF bit is not set.

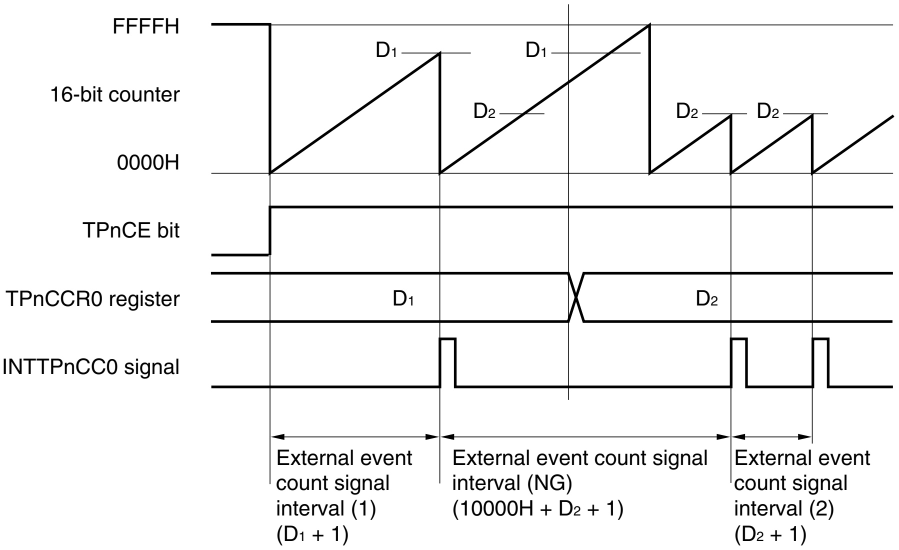

FFFFH
16-bit counter
0000H
TPnCE bit
TPnCCR0 register
FFFFH
INTTPnCC0 signal
External event count signal interval
External event count signal interval
External event count signal interval

Remark n = 0 to 4

(b) Notes on rewriting the TPnCCR0 register

To change the value of the TPnCCR0 register to a smaller value, stop counting once and then change the set value.

If the value of the TPnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow.


FFFFH
16-bit counter
0000H
D1
D1
D2
D2
D2
TPnCE bit
TPnCCR0 register
D1
D2
INTTPnCC0 signal
External event count signal interval (1) (D1 + 1)
External event count signal interval (NG) (10000H + D2 + 1)
External event count signal interval (2) (D2 + 1)

Remark n = 0 to 4

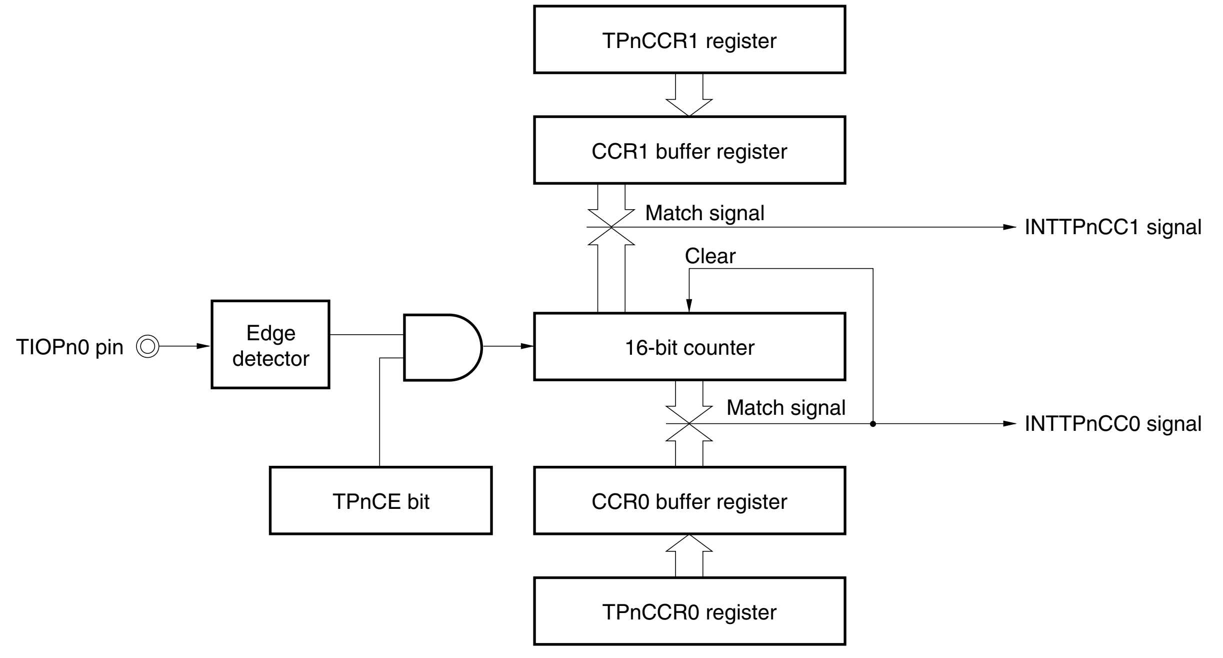
If the value of the TPnCCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TPnCCR0 register has been rewritten. Consequently, the value that is compared with the 16-bit counter is D_2.

Because the count value has already exceeded D_2, however, the 16-bit counter counts up to FFFFH, overflows, and then counts up again from 0000H. When the count value matches D_2, the INTTPnCC0 signal is generated.

Therefore, the INTTPnCC0 signal may not be generated at the valid edge count of “(D_1 + 1) times” or “(D_2 + 1) times” originally expected, but may be generated at the valid edge count of “(10000H + D_2 + 1) times”.

(c) Operation of TPnCCR1 register

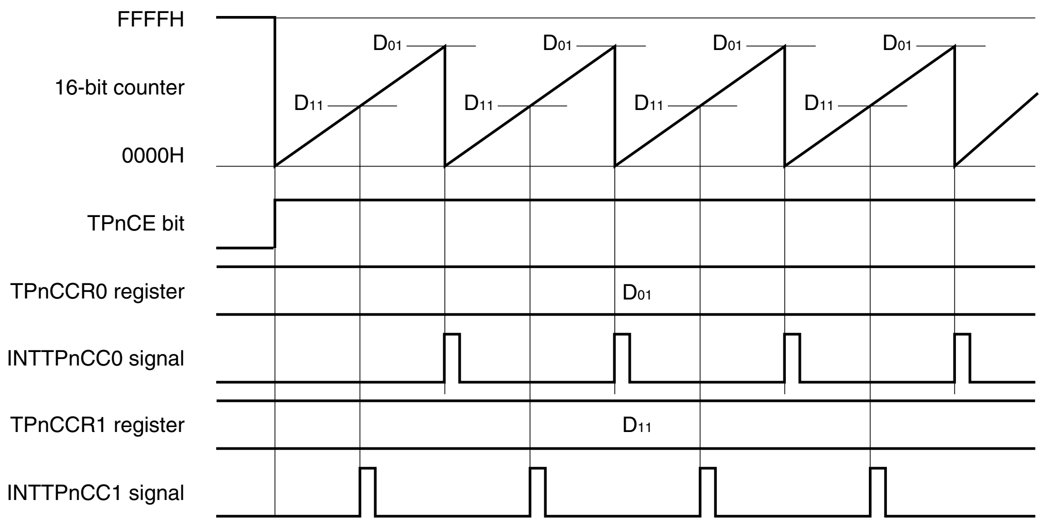
Figure 6-16. Configuration of TPnCCR1 Register



Remark n = 0 to 4

If the set value of the TPnCCR1 register is smaller than the set value of the TPnCCR0 register, the INTTPnCC1 signal is generated once per cycle.

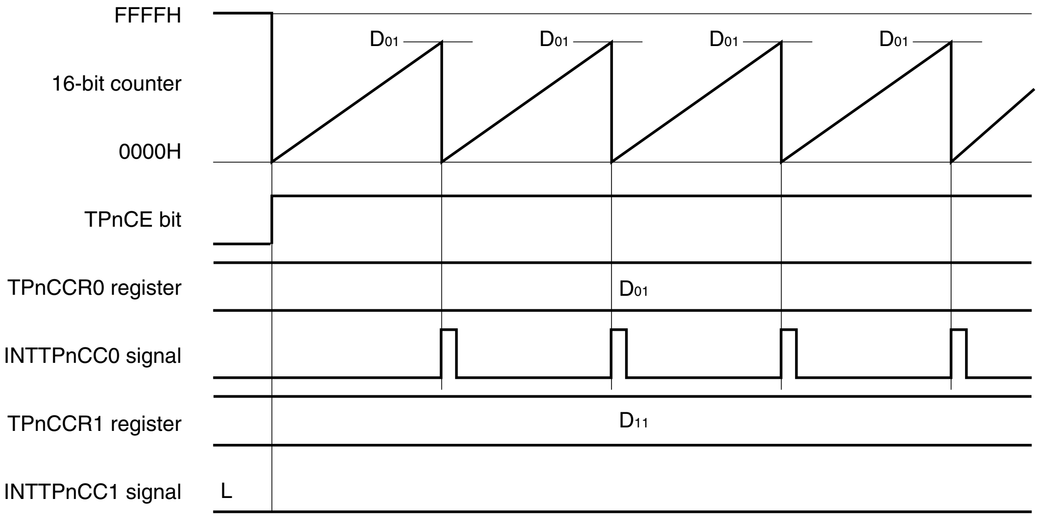
Figure 6-17. Timing Chart When $D_{01} \geq D_{11}$



Remark n = 0 to 4

If the set value of the TPnCCR1 register is greater than the set value of the TPnCCR0 register, the INTTPnCC1 signal is not generated because the count value of the 16-bit counter and the value of the TPnCCR1 register do not match.

Figure 6-18. Timing Chart When $D_{01} < D_{11}$



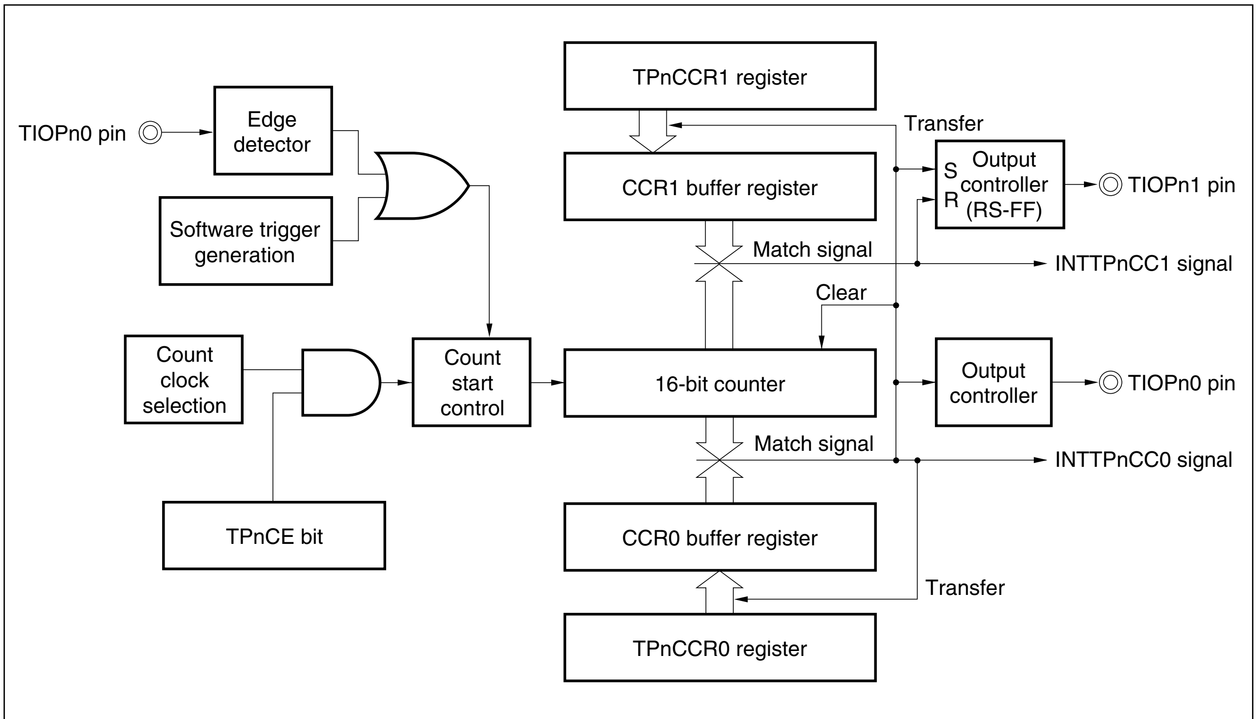
Remark n = 0 to 4

6.7.3 External trigger pulse output mode (TPnMD2 to TPnMD0 bits = 010)

In the external trigger pulse output mode, 16-bit timer/event counter P waits for a trigger when the TPnCTL0.TPnCE bit is set to 1. When the valid edge of an external trigger input signal is detected, 16-bit timer/event counter P starts counting, and outputs a PWM waveform from the TIOPn1 pin.

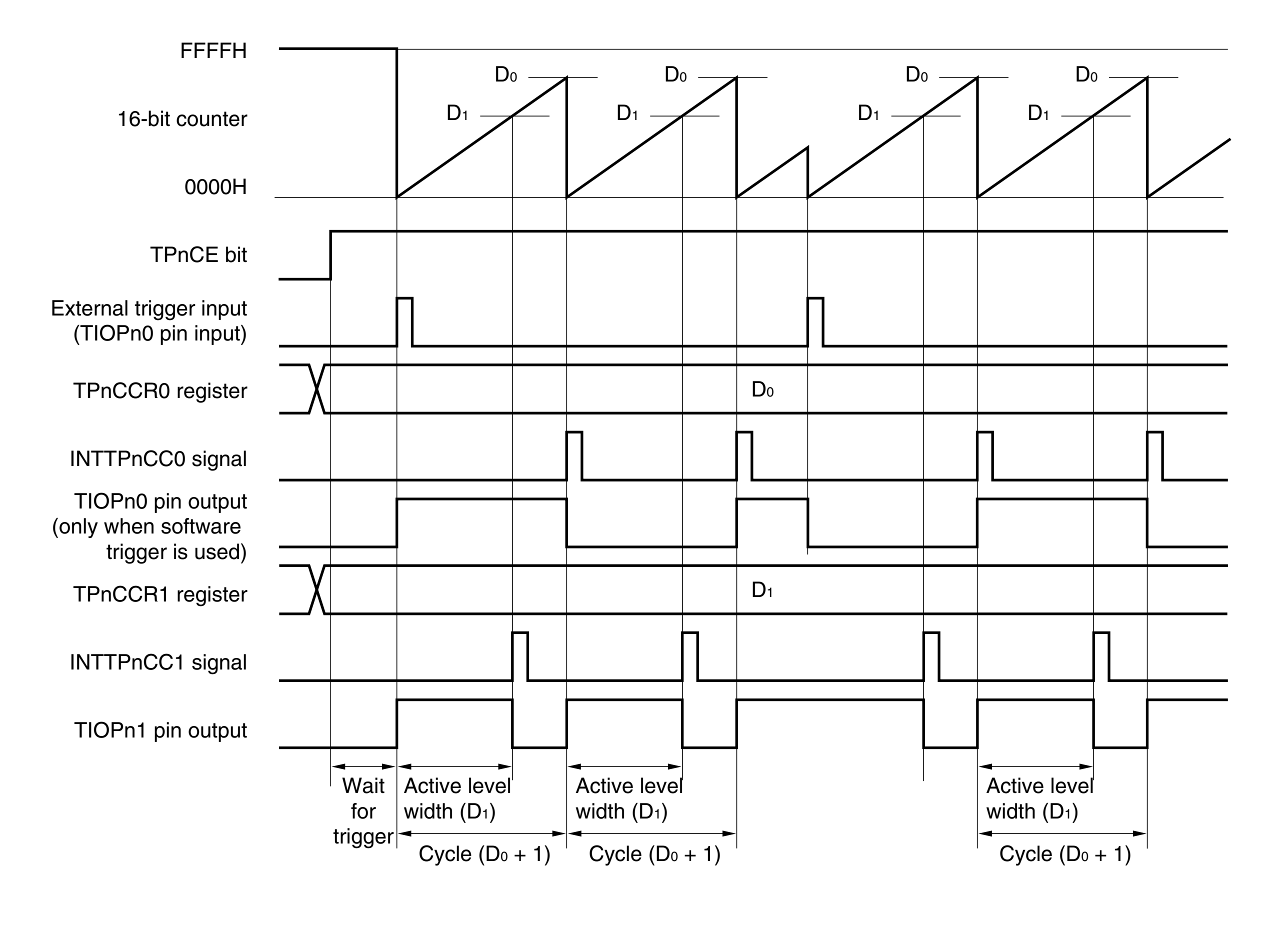
Pulses can also be output by generating a software trigger instead of using the external trigger. When using a software trigger, a square wave that has one cycle of the PWM waveform as half its cycle can also be output from the TIOPn0 pin.

Figure 6-19. Configuration in External Trigger Pulse Output Mode



Remark n = 0 to 4

Figure 6-20. Basic Timing in External Trigger Pulse Output Mode



16-bit timer/event counter P waits for a trigger when the TPnCE bit is set to 1. When the trigger is generated, the 16-bit counter is cleared from FFFFH to 0000H, starts counting at the same time, and outputs a PWM waveform from the TIOPn1 pin. If the trigger is generated again while the counter is operating, the counter is cleared to 0000H and restarted. (The output of the TIOPn0 pin is inverted. The TIOPn1 pin outputs a high-level regardless of the status (high/low) when a trigger occurs.)

The active level width, cycle, and duty factor of the PWM waveform can be calculated as follows.

Active level width = (Set value of TPnCCR1 register) × Count clock cycle
Cycle = (Set value of TPnCCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TPnCCR1 register)/(Set value of TPnCCR0 register + 1)

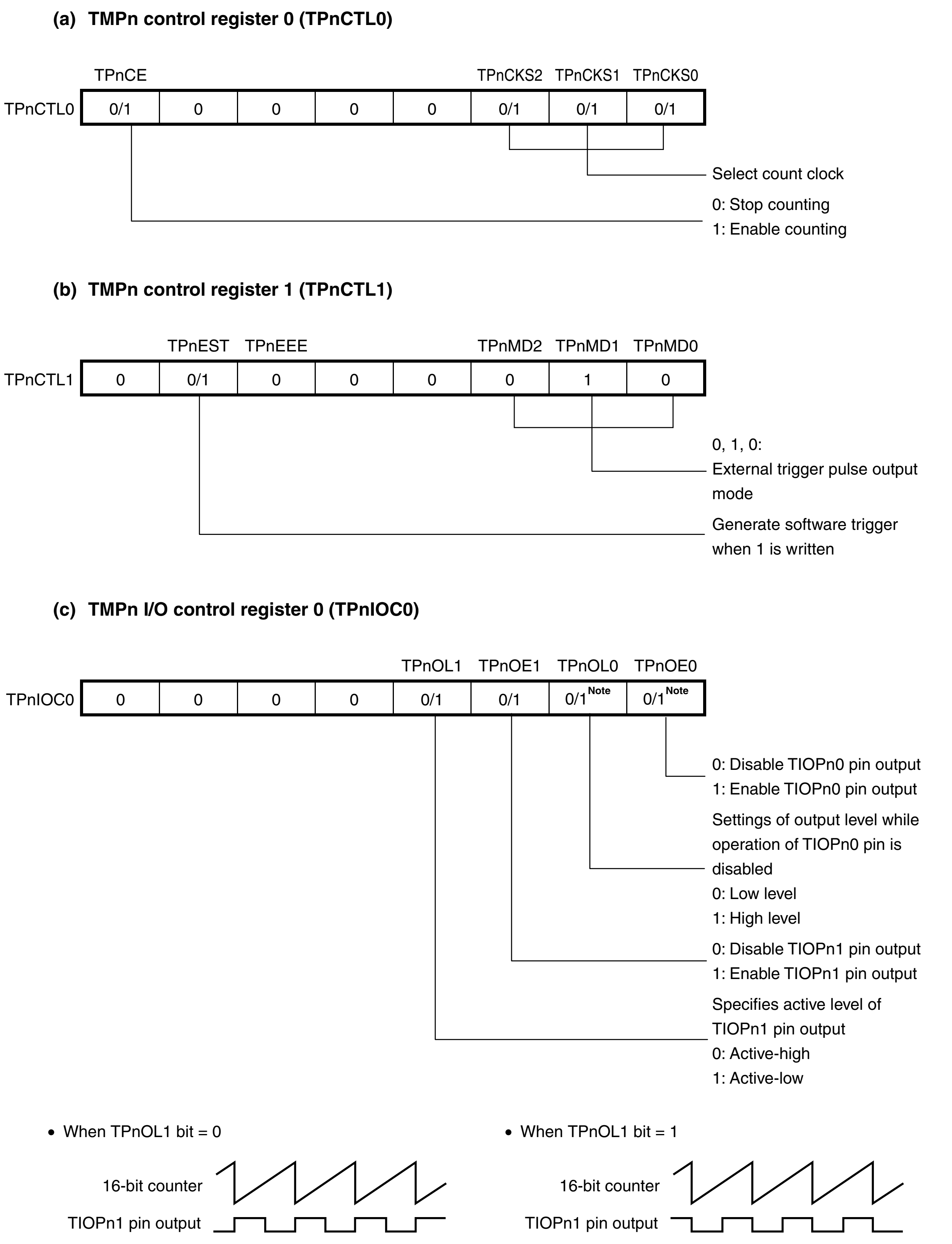
The compare match request signal INTTPnCC0 is generated when the 16-bit counter counts next time after its count value matches the value of the CCR0 buffer register, and the 16-bit counter is cleared to 0000H. The compare match interrupt request signal INTTPnCC1 is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

The value set to the TPnCCRm register is transferred to the CCRm buffer register when the count value of the 16-bit counter matches the value of the CCRm buffer register and the 16-bit counter is cleared to 0000H.

The valid edge of an external trigger input signal, or setting the software trigger (TPnCTL1.TPnEST bit) to 1 is used as the trigger.

Remark n = 0 to 4, m = 0, 1

Figure 6-21. Register Setting for Operation in External Trigger Pulse Output Mode (1/2)


(a) TMPn control register 0 (TPnCTL0)
TPnCE
TPnCKS2
TPnCKS1
TPnCKS0
TPnCTL0
0/1
0
0
0
0
0/1
0/1
0/1
Select count clock
0: Stop counting
1: Enable counting
(b) TMPn control register 1 (TPnCTL1)
TPnEST
TPnEEE
TPnMD2
TPnMD1
TPnMD0
TPnCTL1
0
0/1
0
0
0
0
1
0
0, 1, 0:
External trigger pulse output mode
Generate software trigger when 1 is written
(c) TMPn I/O control register 0 (TPnIOC0)
TPnOL1
TPnOE1
TPnOL0
TPnOE0
TPnIOC0
0
0
0
0
0/1
0/1
0/1 Note
0/1 Note
0: Disable TIOPn0 pin output
1: Enable TIOPn0 pin output
Settings of output level while operation of TIOPn0 pin is disabled
0: Low level
1: High level
0: Disable TIOPn1 pin output
1: Enable TIOPn1 pin output
Specifies active level of TIOPn1 pin output
0: Active-high
1: Active-low
• When TPnOL1 bit = 0
16-bit counter
TIOPn1 pin output
• When TPnOL1 bit = 1
16-bit counter
TIOPn1 pin output

Note Clear this bit to 0 when the TIOPn0 pin is not used in the external trigger pulse output mode.

Figure 6-21. Register Setting for Operation in External Trigger Pulse Output Mode (2/2)

(d) TMPn I/O control register 2 (TPnIOC2)

					TPnEES1	TPnEES0	TPnETS1	TPnETS0
TPnIOC2	0	0	0	0	0	0	0/1	0/1

TPnETS1, TPnETS0: Select valid edge of external event trigger input

(e) TMPn counter read buffer register (TPnCNT)

The value of the 16-bit counter can be read by reading the TPnCNT register.

(f) TMPn capture/compare registers 0 and 1 (TPnCCR0 and TPnCCR1)

If D_0 is set to the TPnCCR0 register and D_1 to the TPnCCR1 register, the cycle and active level of the PWM waveform are as follows.

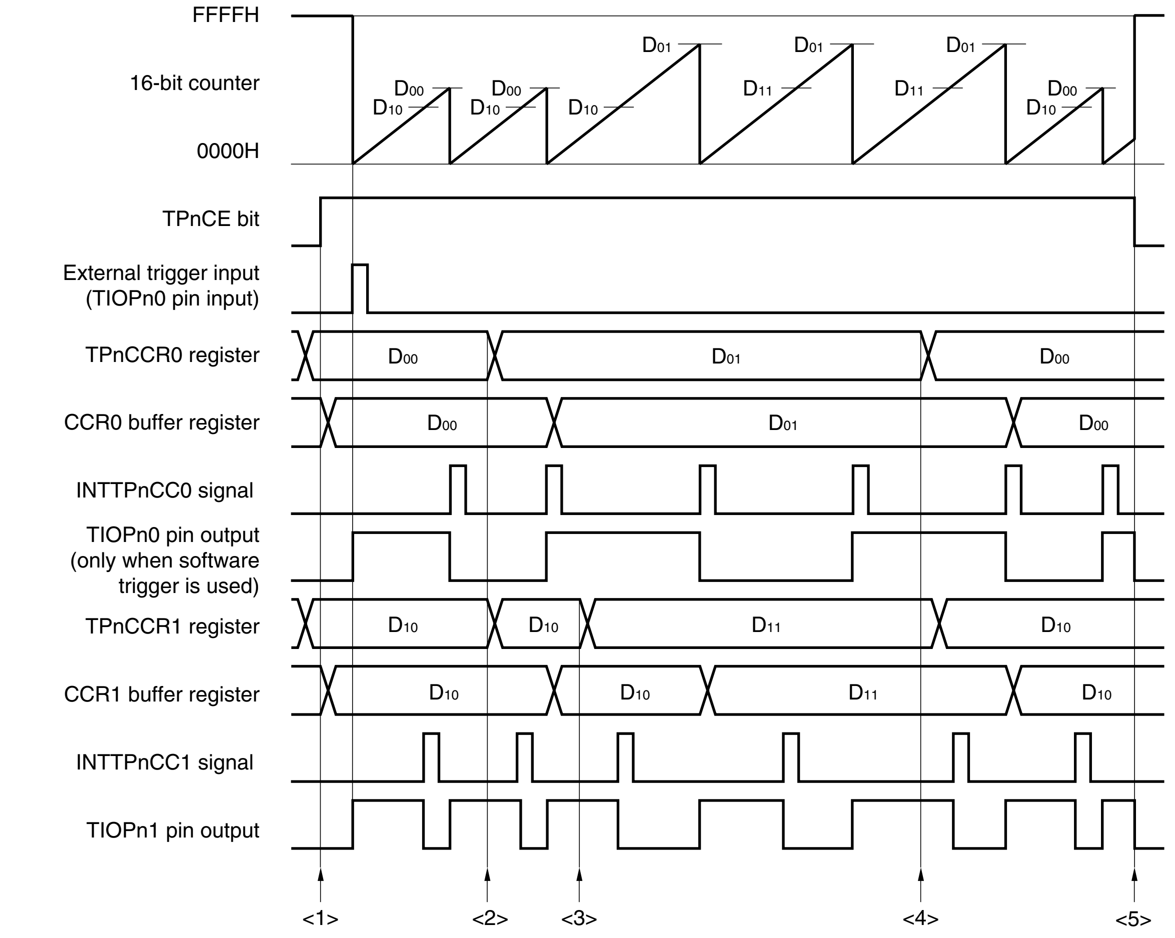
Cycle = (D_0 + 1) × Count clock cycle
Active level width = D_1 × Count clock cycle

Remarks 1. TMPn I/O control register 1 (TPnIOC1) and TMPn option register 0 (TPnOPT0) are not used in the external trigger pulse output mode.

2. n = 0 to 4

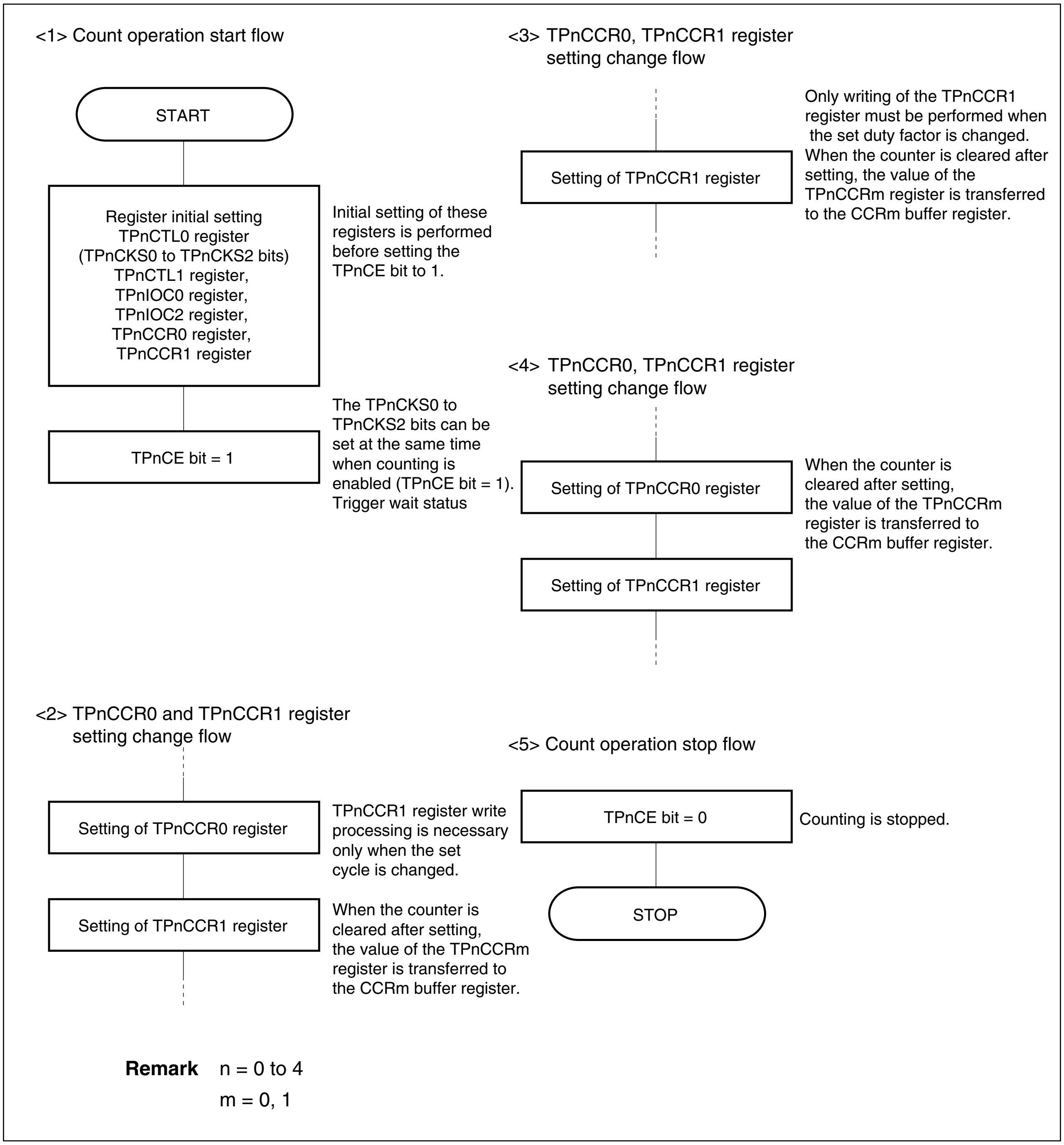
(1) Operation flow in external trigger pulse output mode

Figure 6-22. Software Processing Flow in External Trigger Pulse Output Mode (1/2)



Remark n = 0 to 4

Figure 6-22. Software Processing Flow in External Trigger Pulse Output Mode (2/2)

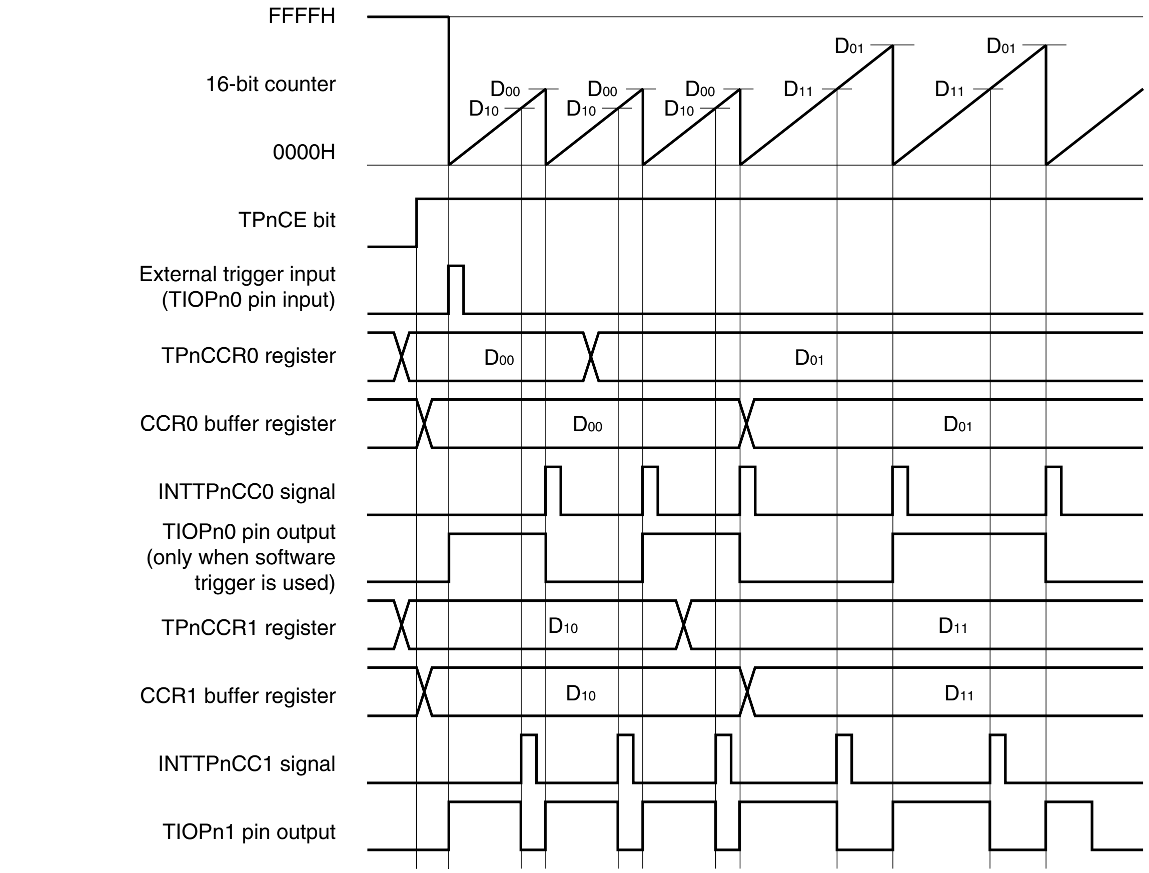

<1> Count operation start flow
START
Register initial setting
TPnCTL0 register
(TPnCKS0 to TPnCKS2 bits)
TPnCTL1 register,
TPnIOC0 register,
TPnIOC2 register,
TPnCCR0 register,
TPnCCR1 register
Initial setting of these registers is performed before setting the TPnCE bit to 1.
TPnCE bit = 1
The TPnCKS0 to TPnCKS2 bits can be set at the same time when counting is enabled (TPnCE bit = 1).
Trigger wait status
<2> TPnCCR0 and TPnCCR1 register setting change flow
Setting of TPnCCR0 register
TPnCCR1 register write processing is necessary only when the set cycle is changed.
Setting of TPnCCR1 register
When the counter is cleared after setting, the value of the TPnCCRm register is transferred to the CCRm buffer register.
<3> TPnCCR0, TPnCCR1 register setting change flow
Setting of TPnCCR1 register
Only writing of the TPnCCR1 register must be performed when the set duty factor is changed.
When the counter is cleared after setting, the value of the TPnCCRm register is transferred to the CCRm buffer register.
<4> TPnCCR0, TPnCCR1 register setting change flow
Setting of TPnCCR0 register
When the counter is cleared after setting, the value of the TPnCCRm register is transferred to the CCRm buffer register.
Setting of TPnCCR1 register
<5> Count operation stop flow
TPnCE bit = 0
Counting is stopped.
STOP
Remark n = 0 to 4
m = 0, 1

(2) External trigger pulse output mode operation timing

(a) Note on changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TPnCCR1 register last.

Rewrite the TPnCCRm register after writing the TPnCCR1 register after the INTTPnCC0 signal is detected.



In order to transfer data from the TPnCCRm register to the CCRm buffer register, the TPnCCR1 register must be written.

To change both the cycle and active level width of the PWM waveform at this time, first set the cycle to the TPnCCR0 register and then set the active level width to the TPnCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TPnCCR0 register, and then write the same value to the TPnCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, only the TPnCCR1 register has to be set.

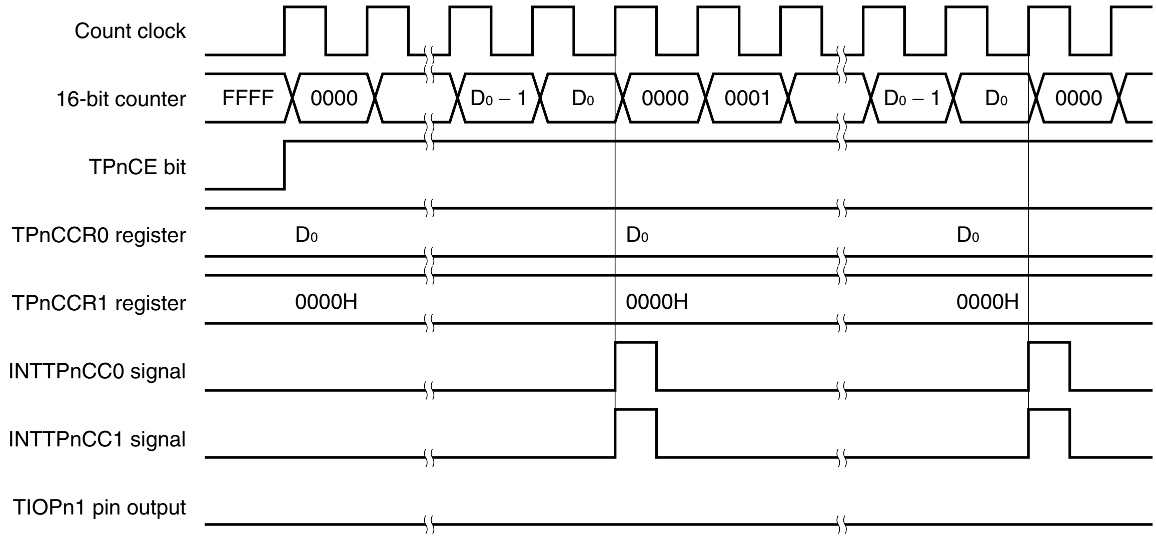
After data is written to the TPnCCR1 register, the value written to the TPnCCRm register is transferred to the CCRm buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TPnCCR0 or TPnCCR1 register again after writing the TPnCCR1 register once, do so after the INTTPnCC0 signal is generated. Otherwise, the value of the CCRm buffer register may become undefined because the timing of transferring data from the TPnCCRm register to the CCRm buffer register conflicts with writing the TPnCCRm register.

Remark n = 0 to 4
m = 0, 1

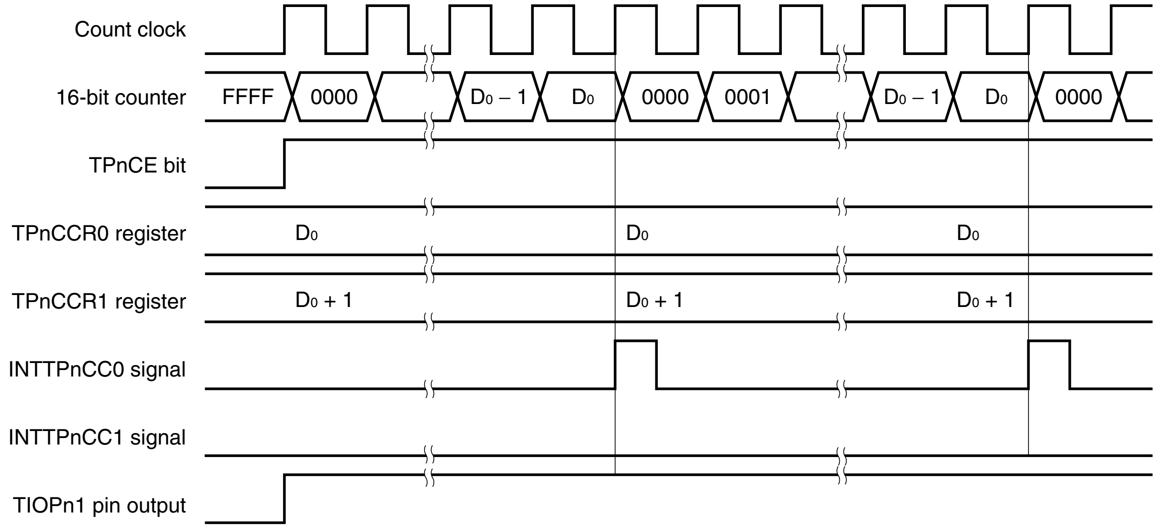
(b) 0%/100% output of PWM waveform

To output a 0% waveform, set the TPnCCR1 register to 0000H. If the set value of the TPnCCR0 register is FFFFH, the INTTPnCC1 signal is generated periodically.



Remark n = 0 to 4

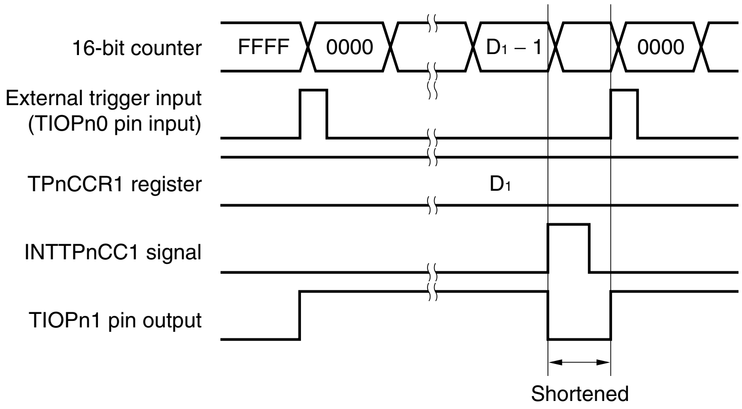
To output a 100% waveform, set a value of (set value of TPnCCR0 register + 1) to the TPnCCR1 register. If the set value of the TPnCCR0 register is FFFFH, 100% output cannot be produced.



Remark n = 0 to 4

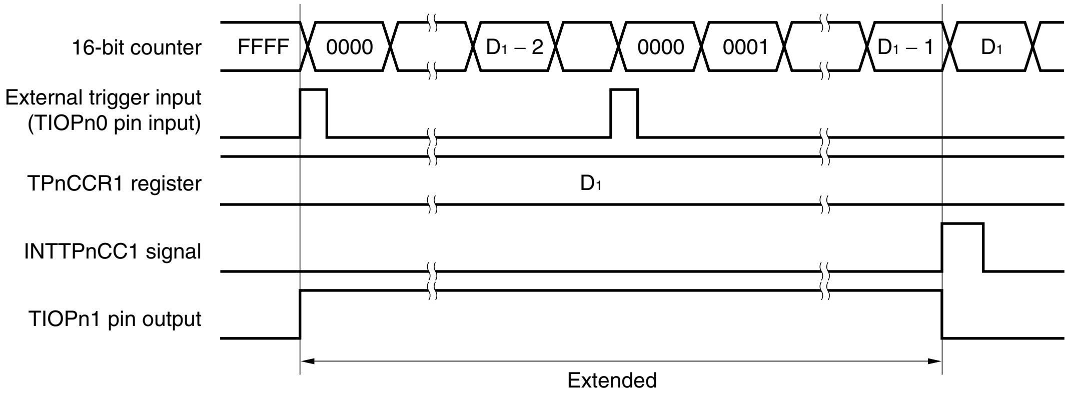
(c) Conflict between trigger detection and match with TPnCCR1 register

If the trigger is detected immediately after the INTTPnCC1 signal is generated, the 16-bit counter is immediately cleared to 0000H, the output signal of the TIOPn1 pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



Remark n = 0 to 4

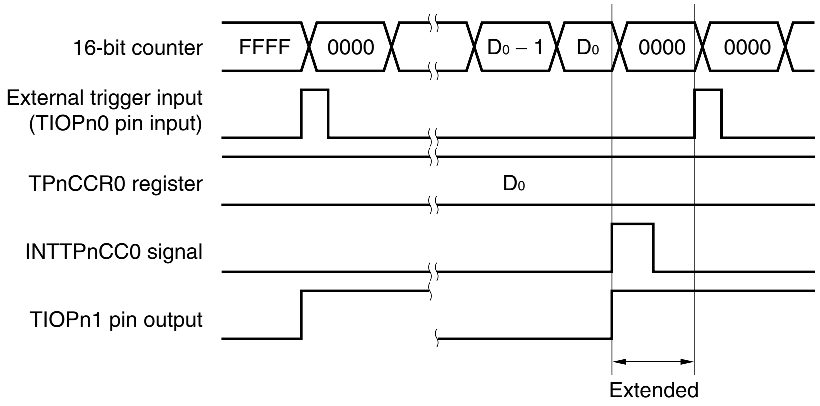
If the trigger is detected immediately before the INTTPnCC1 signal is generated, the INTTPnCC1 signal is not generated, and the 16-bit counter is cleared to 0000H and continues counting. The output signal of the TIOPn1 pin remains active. Consequently, the active period of the PWM waveform is extended.



Remark n = 0 to 4

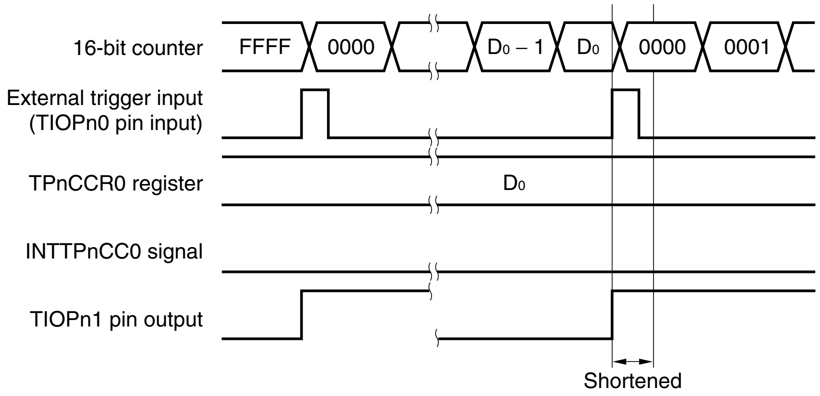
(d) Conflict between trigger detection and match with TPnCCR0 register

If the trigger is detected immediately after the INTTPnCC0 signal is generated, the 16-bit counter is cleared to 0000H and continues counting up. Therefore, the active period of the TIOPn1 pin is extended by time from generation of the INTTPnCC0 signal to trigger detection.



Remark n = 0 to 4

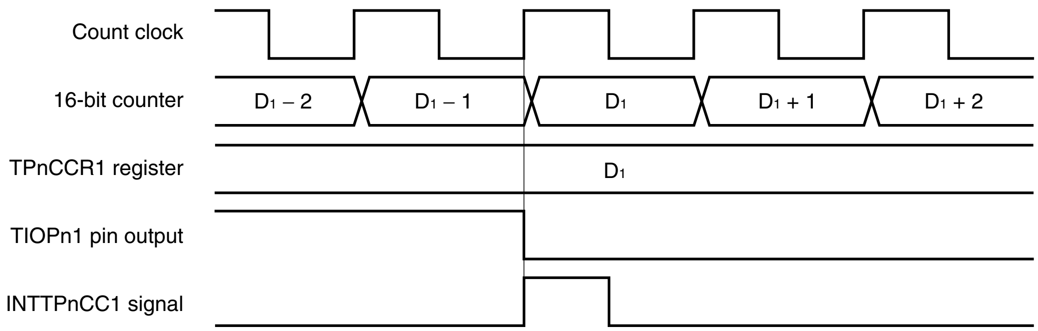
If the trigger is detected immediately before the INTTPnCC0 signal is generated, the INTTPnCC0 signal is not generated. The 16-bit counter is cleared to 0000H, the TIOPn1 pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



Remark n = 0 to 4

(e) Generation timing of compare match interrupt request signal (INTTPnCC1)

The timing of generation of the INTTPnCC1 signal in the external trigger pulse output mode differs from the timing of other INTTPnCC1 signals; the INTTPnCC1 signal is generated when the count value of the 16-bit counter matches the value of the TPnCCR1 register.



Remark n = 0 to 4

Usually, the INTTPnCC1 signal is generated in synchronization with the next count up, after the count value of the 16-bit counter matches the value of the TPnCCR1 register.

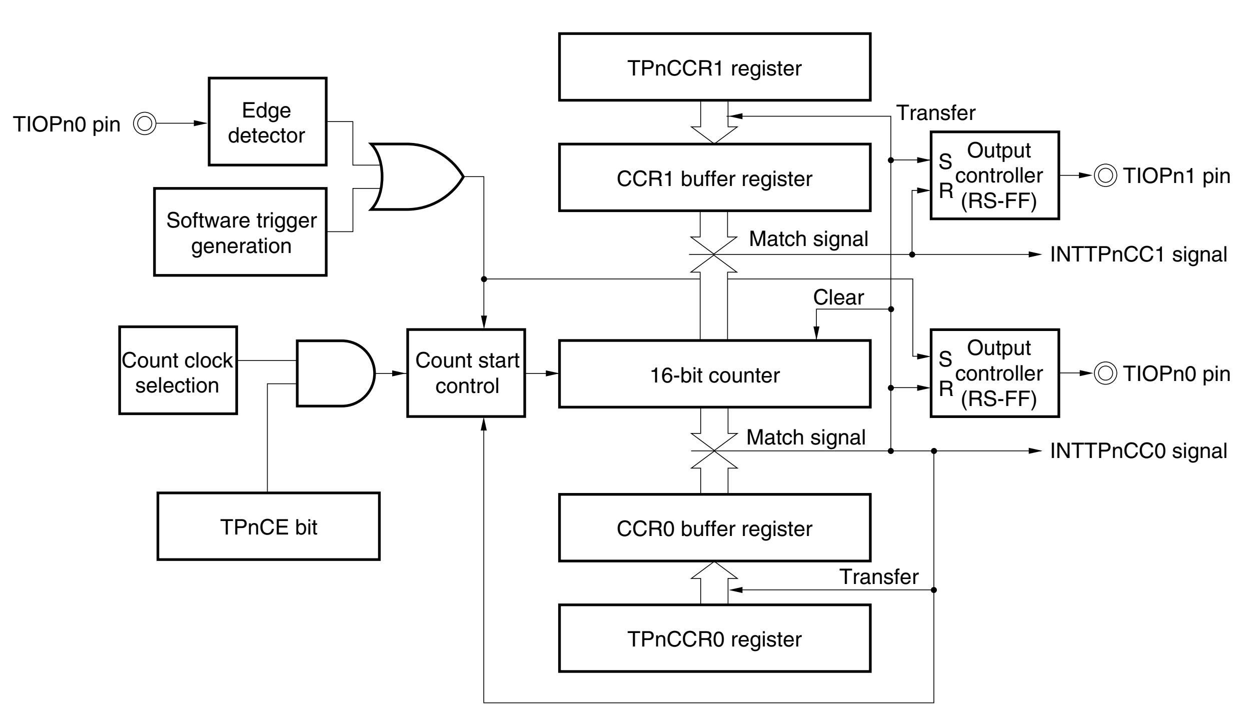
In the external trigger pulse output mode, however, it is generated one clock earlier. This is because the timing is changed to match the timing of changing the output signal of the TIOPn1 pin.

6.7.4 One-shot pulse output mode (TPnMD2 to TPnMD0 bits = 011)

In the one-shot pulse output mode, 16-bit timer/event counter P waits for a trigger when the TPnCTL0.TPnCE bit is set to 1. When the valid edge of an external trigger input is detected, 16-bit timer/event counter P starts counting, and outputs a one-shot pulse from the TIOPn1 pin.

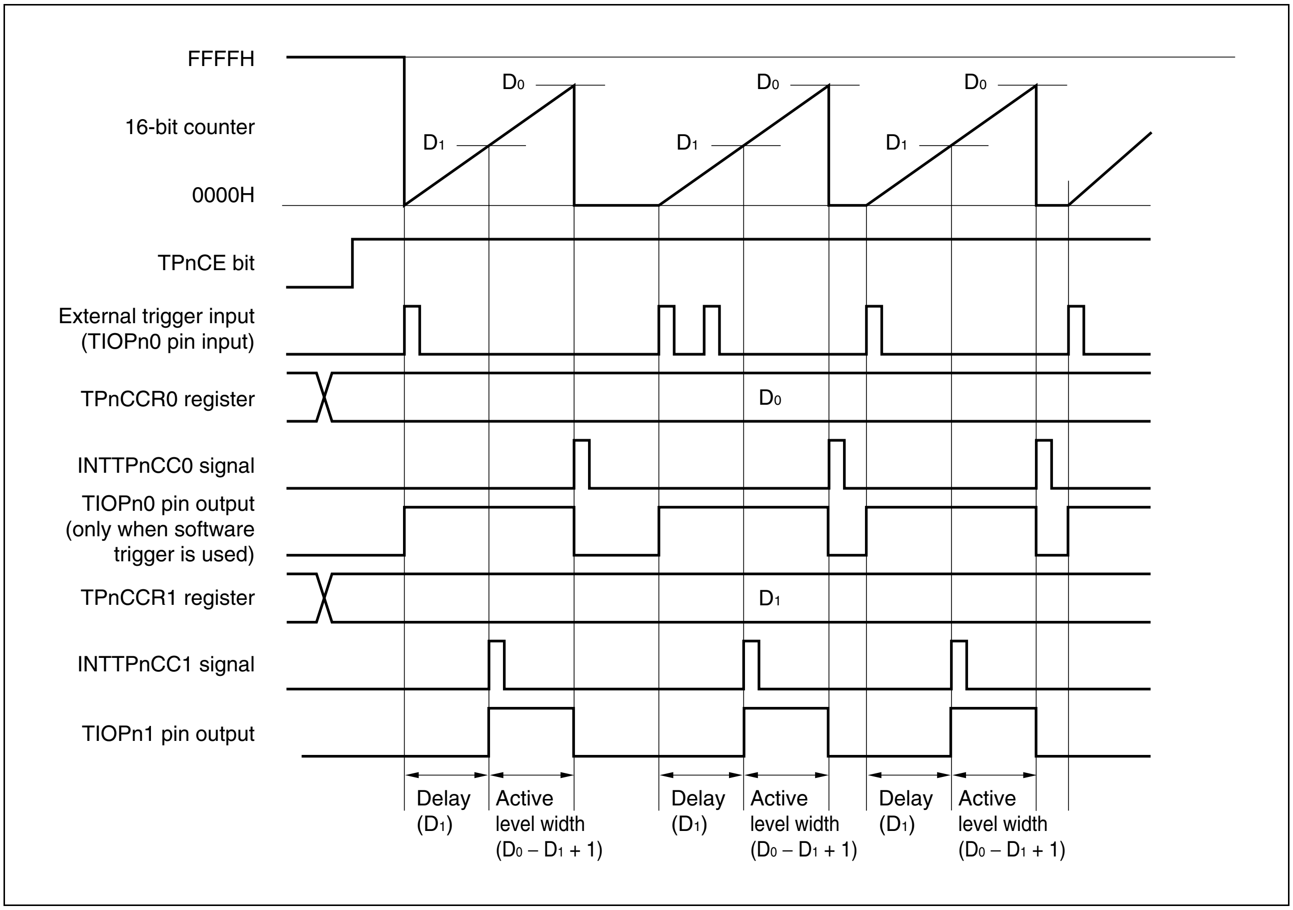
Instead of the external trigger, a software trigger can also be generated to output the pulse. When the software trigger is used, the TIOPn0 pin outputs the active level while the 16-bit counter is counting, and the inactive level when the counter is stopped (waiting for a trigger).

Figure 6-23. Configuration in One-shot Pulse Output Mode



Remark n = 0 to 4

Figure 6-24. Basic Timing in One-shot Pulse Output Mode



When the TPnCE bit is set to 1, 16-bit timer/event counter P waits for a trigger. When the trigger is generated, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs a one-shot pulse from the TIOPn1 pin. After the one-shot pulse is output, the 16-bit counter is set to FFFFH, stops counting, and waits for a trigger. If a trigger is generated again while the one-shot pulse is being output, it is ignored.

The output delay period and active level width of the one-shot pulse can be calculated as follows.

Output delay period = (Set value of TPnCCR1 register) × Count clock cycle

Active level width = (Set value of TPnCCR0 register – Set value of TPnCCR1 register + 1) × Count clock cycle

The compare match interrupt request signal INTTPnCC0 is generated when the 16-bit counter counts after its count value matches the value of the CCR0 buffer register. The compare match interrupt request signal INTTPnCC1 is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

The valid edge of an external trigger input or setting the software trigger (TPnCTL1.TPnEST bit) to 1 is used as the trigger.

Remark n = 0 to 4
m = 0, 1

Figure 6-25. Register Setting for Operation in One-shot Pulse Output Mode (1/2)

(a) TMPn control register 0 (TPnCTL0)

	TPnCE					TPnCKS2	TPnCKS1	TPnCKS0
TPnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

TPnCKS2 to TPnCKS0: Select count clock

TPnCE:
0: Stop counting
1: Enable counting

(b) TMPn control register 1 (TPnCTL1)

		TPnEST	TPnEEE			TPnMD2	TPnMD1	TPnMD0
TPnCTL1	0	0/1	0	0	0	0	1	1

TPnMD2 to TPnMD0:
0, 1, 1:
One-shot pulse output mode

TPnEST: Generate software trigger when 1 is written

(c) TMPn I/O control register 0 (TPnIOC0)

					TPnOL1	TPnOE1	TPnOL0	TPnOE0
TPnIOC0	0	0	0	0	0/1	0/1	0/1 Note	0/1 Note

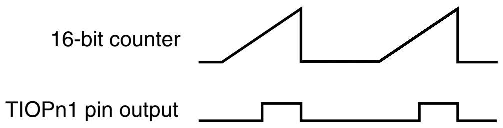
TPnOE0:
0: Disable TIOPn0 pin output
1: Enable TIOPn0 pin output

TPnOL0: Setting of output level while operation of TIOPn0 pin is disabled
0: Low level
1: High level

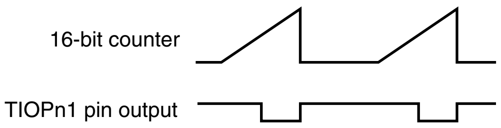
TPnOE1:
0: Disable TIOPn1 pin output
1: Enable TIOPn1 pin output

TPnOL1: Specifies active level of TIOPn1 pin output
0: Active-high
1: Active-low

- When TPnOL1 bit = 0



- When TPnOL1 bit = 1



Note Clear this bit to 0 when the TIOPn0 pin is not used in the one-shot pulse output mode.

Figure 6-25. Register Setting for Operation in One-shot Pulse Output Mode (2/2)

(d) **TMPn I/O control register 2 (TPnIOC2)**

					TPnEES1	TPnEES0	TPnETS1	TPnETS0
TPnIOC2	0	0	0	0	0	0	0/1	0/1

Select valid edge of external trigger input

(e) **TMPn counter read buffer register (TPnCNT)**

The value of the 16-bit counter can be read by reading the TPnCNT register.

(f) **TMPn capture/compare registers 0 and 1 (TPnCCR0 and TPnCCR1)**

If D_0 is set to the TPnCCR0 register and D_1 to the TPnCCR1 register, the active level width and output delay period of the one-shot pulse are as follows.

Active level width = $(D_0 - D_1 + 1) \times$ Count clock cycle

Output delay period = $(D_1) \times$ Count clock cycle

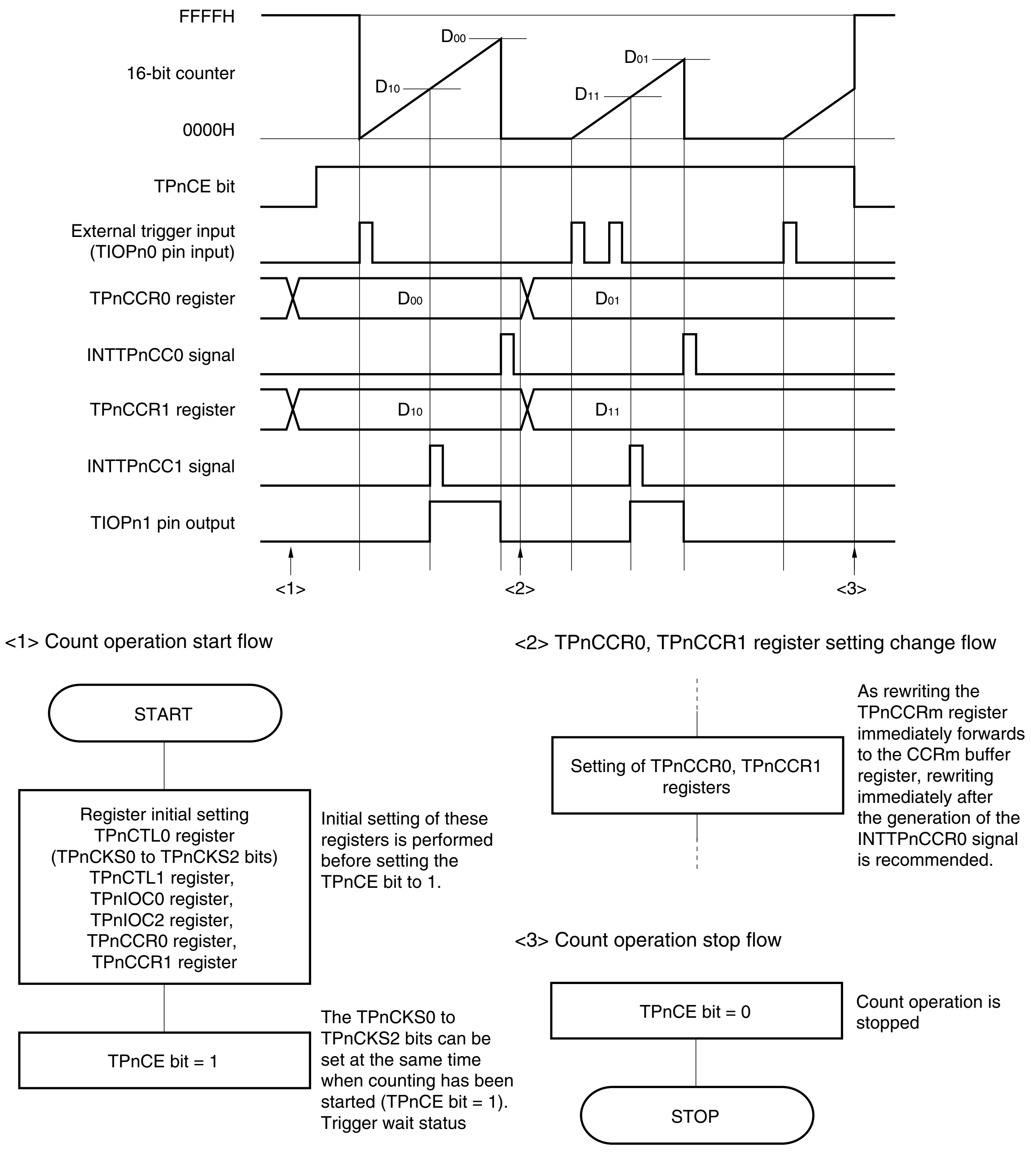
Caution One-shot pulses are not output even in the one-shot pulse output mode, if the value set in the TPnCCR1 register is greater than that set in the TPnCCR0 register.

Remarks 1. TMPn I/O control register 1 (TPnIOC1) and TMPn option register 0 (TPnOPT0) are not used in the one-shot pulse output mode.

2. n = 0 to 4

(1) Operation flow in one-shot pulse output mode

Figure 6-26. Software Processing Flow in One-shot Pulse Output Mode


FFFFH
16-bit counter
0000H
D00
D10
D01
D11
TPnCE bit
External trigger input (TIOPn0 pin input)
TPnCCR0 register
D00
D01
INTTPnCC0 signal
TPnCCR1 register
D10
D11
INTTPnCC1 signal
TIOPn1 pin output
<1>
<2>
<3>
<1> Count operation start flow
START
Register initial setting TPnCTL0 register (TPnCKS0 to TPnCKS2 bits) TPnCTL1 register, TPnIOC0 register, TPnIOC2 register, TPnCCR0 register, TPnCCR1 register
TPnCE bit = 1
<2> TPnCCR0, TPnCCR1 register setting change flow
Setting of TPnCCR0, TPnCCR1 registers
<3> Count operation stop flow
TPnCE bit = 0
STOP

Initial setting of these registers is performed before setting the TPnCE bit to 1.

The TPnCKS0 to TPnCKS2 bits can be set at the same time when counting has been started (TPnCE bit = 1). Trigger wait status

As rewriting the TPnCCRm register immediately forwards to the CCRm buffer register, rewriting immediately after the generation of the INTTPnCCR0 signal is recommended.

Count operation is stopped

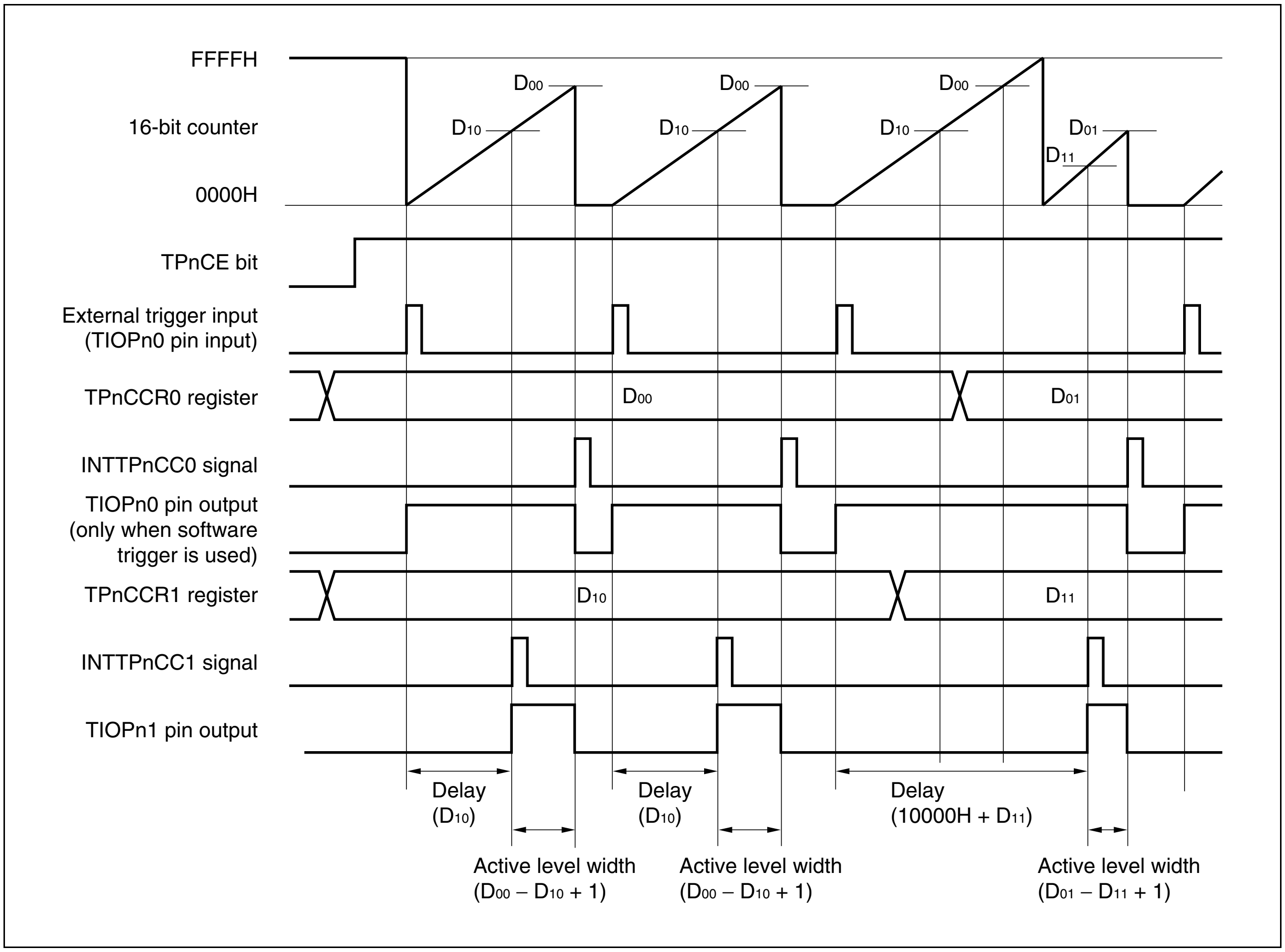
Remark n = 0 to 4
m = 0, 1

(2) Operation timing in one-shot pulse output mode

(a) Note on rewriting TPnCCRm register

To change the set value of the TPnCCRm register to a smaller value, stop counting once, and then change the set value.

If the value of the TPnCCRm register is rewritten to a smaller value during counting, the 16-bit counter may overflow.



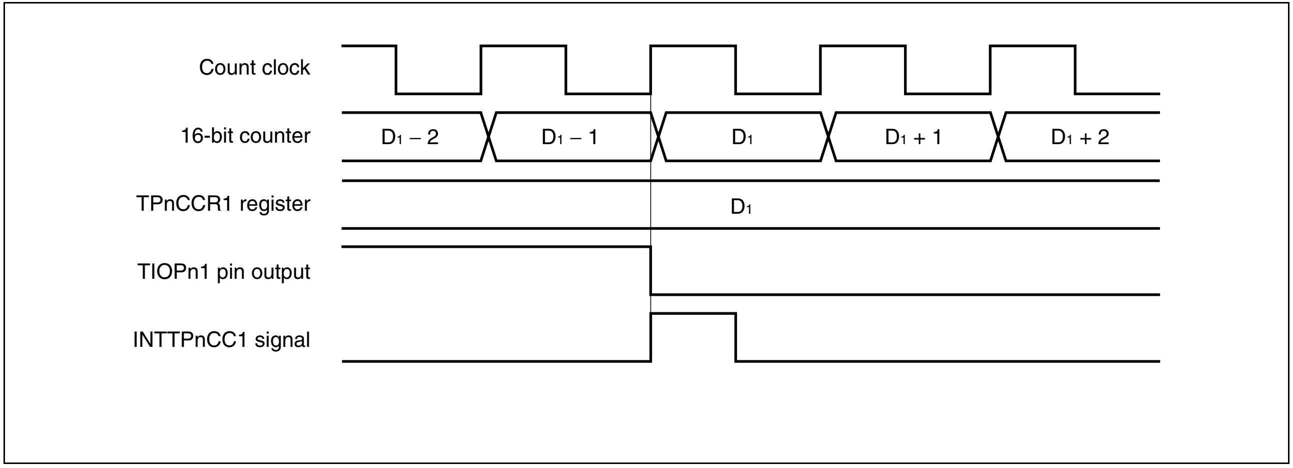
When the TPnCCR0 register is rewritten from D_{00} to D_{01} and the TPnCCR1 register from D_{10} to D_{11} where $D_{00} > D_{01}$ and $D_{10} > D_{11}$, if the TPnCCR1 register is rewritten when the count value of the 16-bit counter is greater than D_{11} and less than D_{10} and if the TPnCCR0 register is rewritten when the count value is greater than D_{01} and less than D_{00}, each set value is reflected as soon as the register has been rewritten and compared with the count value. The counter counts up to FFFFH and then counts up again from 0000H. When the count value matches D_{11}, the counter generates the INTTPnCC1 signal and asserts the TIOPn1 pin. When the count value matches D_{01}, the counter generates the INTTPnCC0 signal, deasserts the TIOPn1 pin, and stops counting.

Therefore, the counter may output a pulse with a delay period or active period different from that of the one-shot pulse that is originally expected.

Remark n = 0 to 4
m = 0, 1

(b) Generation timing of compare match interrupt request signal (INTTPnCC1)

The generation timing of the INTTPnCC1 signal in the one-shot pulse output mode is different from other INTTPnCC1 signals; the INTTPnCC1 signal is generated when the count value of the 16-bit counter matches the value of the TPnCCR1 register.



Usually, the INTTPnCC1 signal is generated when the 16-bit counter counts up next time after its count value matches the value of the TPnCCR1 register.

In the one-shot pulse output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the TIOPn1 pin.

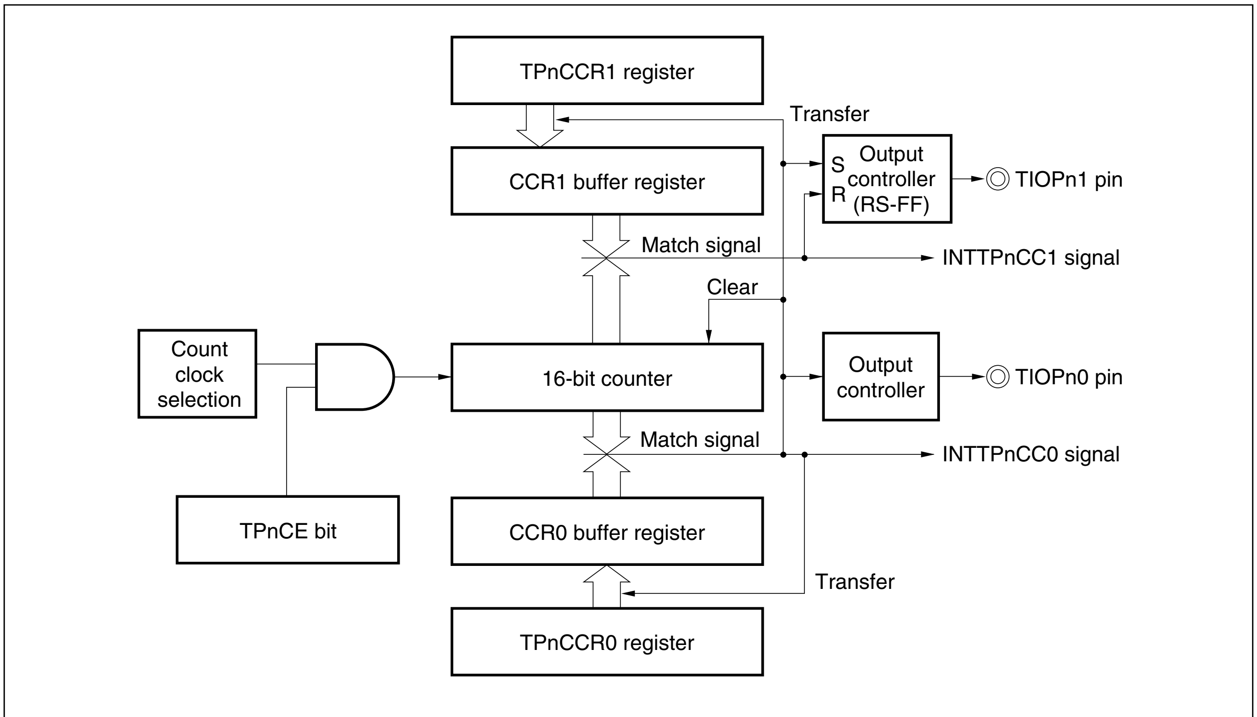
Remark n = 0 to 4

6.7.5 PWM output mode (TPnMD2 to TPnMD0 bits = 100)

In the PWM output mode, a PWM waveform is output from the TIOPn1 pin when the TPnCTL0.TPnCE bit is set to 1.

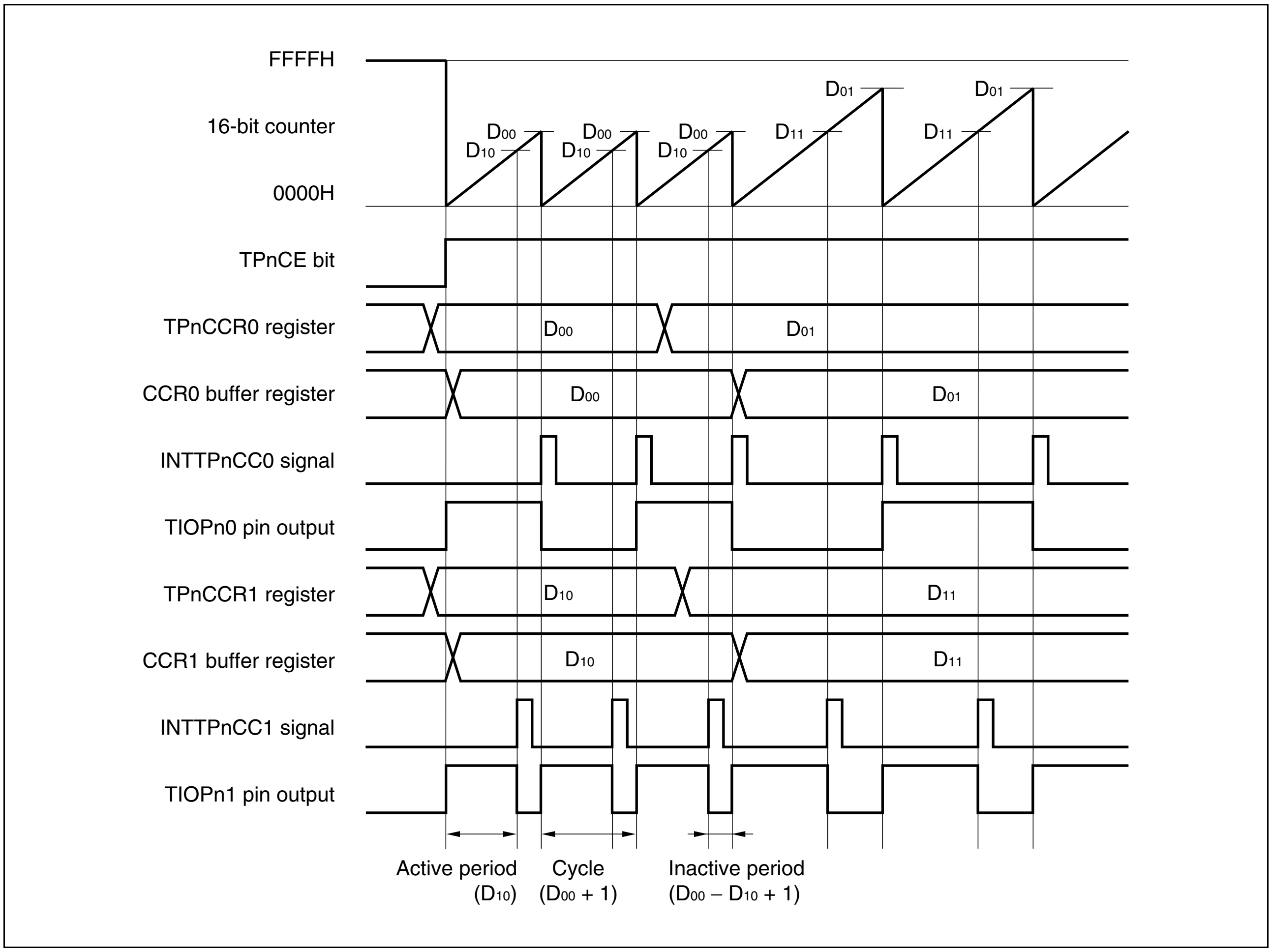
In addition, a pulse with one cycle of the PWM waveform as half its cycle is output from the TIOPn0 pin.

Figure 6-27. Configuration in PWM Output Mode



Remark n = 0 to 4

Figure 6-28. Basic Timing in PWM Output Mode



When the TPnCE bit is set to 1, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs a PWM waveform from the TIOPn1 pin.

The active level width, cycle, and duty factor of the PWM waveform can be calculated as follows.

Active level width = (Set value of TPnCCR1 register) × Count clock cycle
Cycle = (Set value of TPnCCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TPnCCR1 register)/(Set value of TPnCCR0 register + 1)

The PWM waveform can be changed by rewriting the TPnCCRm register while the counter is operating. The newly written value is reflected when the count value of the 16-bit counter matches the value of the CCR0 buffer register and the 16-bit counter is cleared to 0000H.

The compare match interrupt request signal INTTPnCC0 is generated when the 16-bit counter counts next time after its count value matches the value of the CCR0 buffer register, and the 16-bit counter is cleared to 0000H. The compare match interrupt request signal INTTPnCC1 is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

The value set to the TPnCCRm register is transferred to the CCRm buffer register when the count value of the 16-bit counter matches the value of the CCRm buffer register and the 16-bit counter is cleared to 0000H.

Remark n = 0 to 4, m = 0, 1

Figure 6-29. Register Setting for Operation in PWM Output Mode (1/2)

(a) TMPn control register 0 (TPnCTL0)

	TPnCE					TPnCKS2	TPnCKS1	TPnCKS0
TPnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

Select count clock[Note]

0: Stop counting

1: Enable counting

Note The setting is invalid when the TPnCTL1.TPnEEE bit = 1.

(b) TMPn control register 1 (TPnCTL1)

		TPnEST	TPnEEE			TPnMD2	TPnMD1	TPnMD0
TPnCTL1	0	0	0/1	0	0	1	0	0

1, 0, 0: PWM output mode

0: Operate on count clock selected by TPnCKS0 to TPnCKS2 bits

1: Count external event input signal

Figure 6-29. Register Setting for Operation in PWM Output Mode (2/2)

(c) TMPn I/O control register 0 (TPnIOC0)

					TPnOL1	TPnOE1	TPnOL0	TPnOE0
TPnIOC0	0	0	0	0	0/1	0/1	0/1[Note]	0/1[Note]

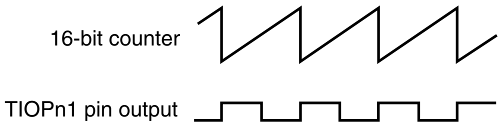
TPnOE0:
0: Disable TIOPn0 pin output
1: Enable TIOPn0 pin output

TPnOL0:
Setting of output level while operation of TIOPn0 pin is disabled
0: Low level
1: High level

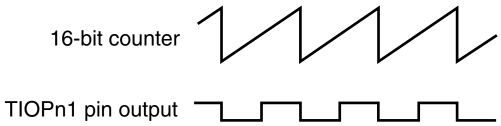
TPnOE1:
0: Disable TIOPn1 pin output
1: Enable TIOPn1 pin output

TPnOL1:
Setting of active level TIOPn1 pin output
0: Active-high
1: Active-low

- When TPnOL1 bit = 0



- When TPnOL1 bit = 1



Note Clear this bit to 0 when the TIOPn0 pin is not used in the PWM output mode.

(d) TMPn I/O control register 2 (TPnIOC2)

					TPnEES1	TPnEES0	TPnETS1	TPnETS0
TPnIOC2	0	0	0	0	0/1	0/1	0	0

TPnEES1, TPnEES0:
Select valid edge of external event count input

(e) TMPn counter read buffer register (TPnCNT)

The value of the 16-bit counter can be read by reading the TPnCNT register.

(f) TMPn capture/compare registers 0 and 1 (TPnCCR0 and TPnCCR1)

If D_0 is set to the TPnCCR0 register and D_1 to the TPnCCR1 register, the cycle and active level of the PWM waveform are as follows.

Cycle = $(D_0 + 1) \times$ Count clock cycle

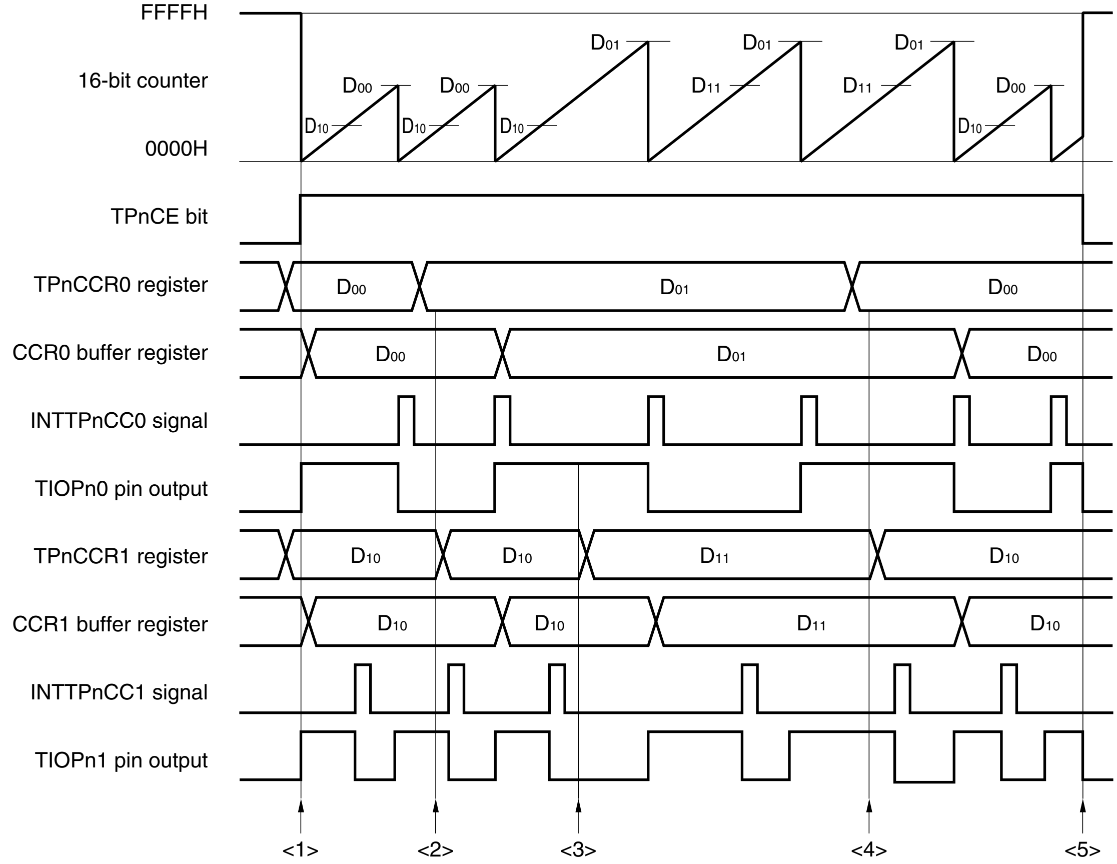
Active level width = $D_1 \times$ Count clock cycle

Remarks 1. TMPn I/O control register 1 (TPnIOC1) and TMPn option register 0 (TPnOPT0) are not used in the PWM output mode.

2. n = 0 to 4

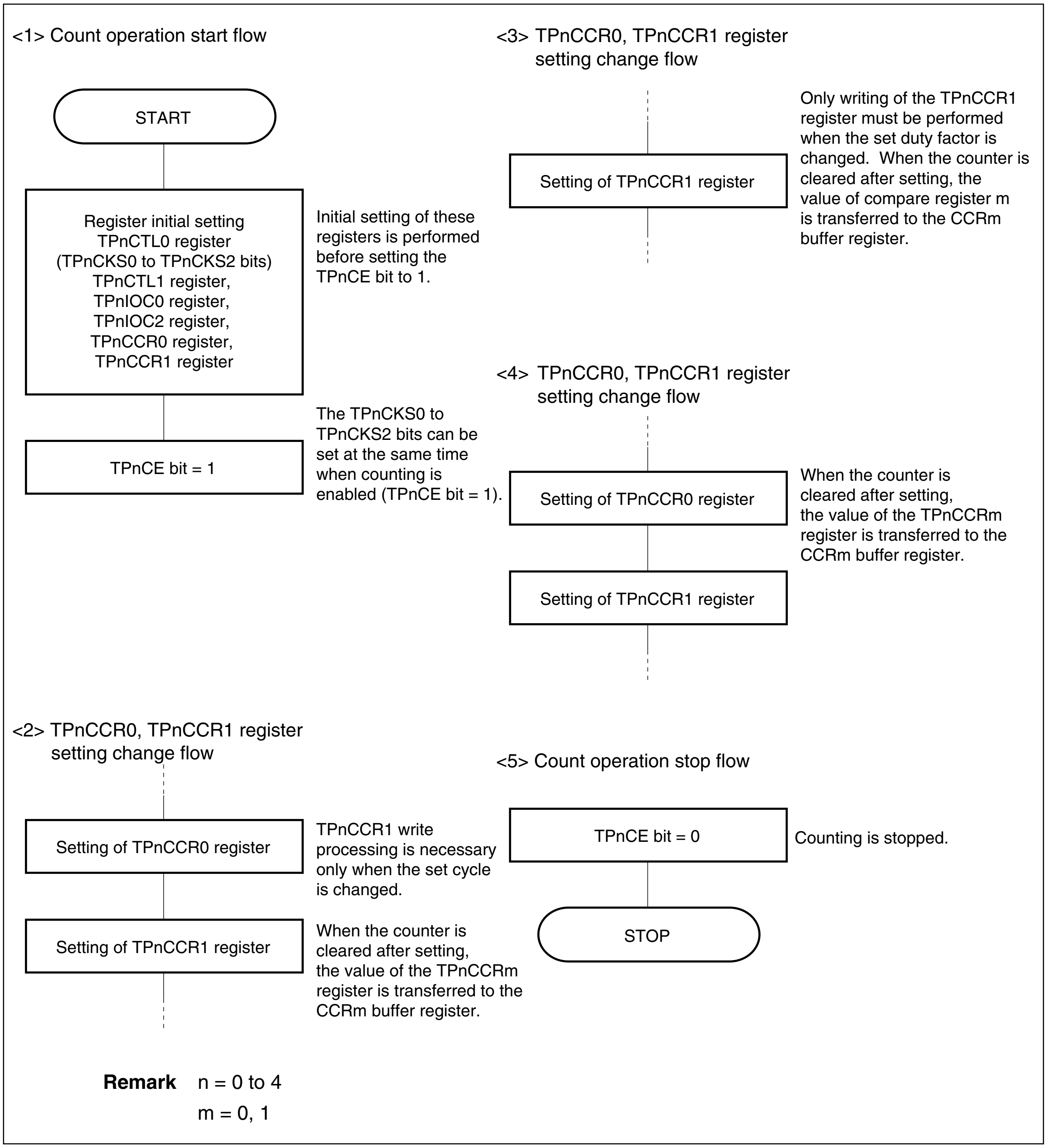
(1) Operation flow in PWM output mode

Figure 6-30. Software Processing Flow in PWM Output Mode (1/2)



Remark n = 0 to 4
m = 0, 1

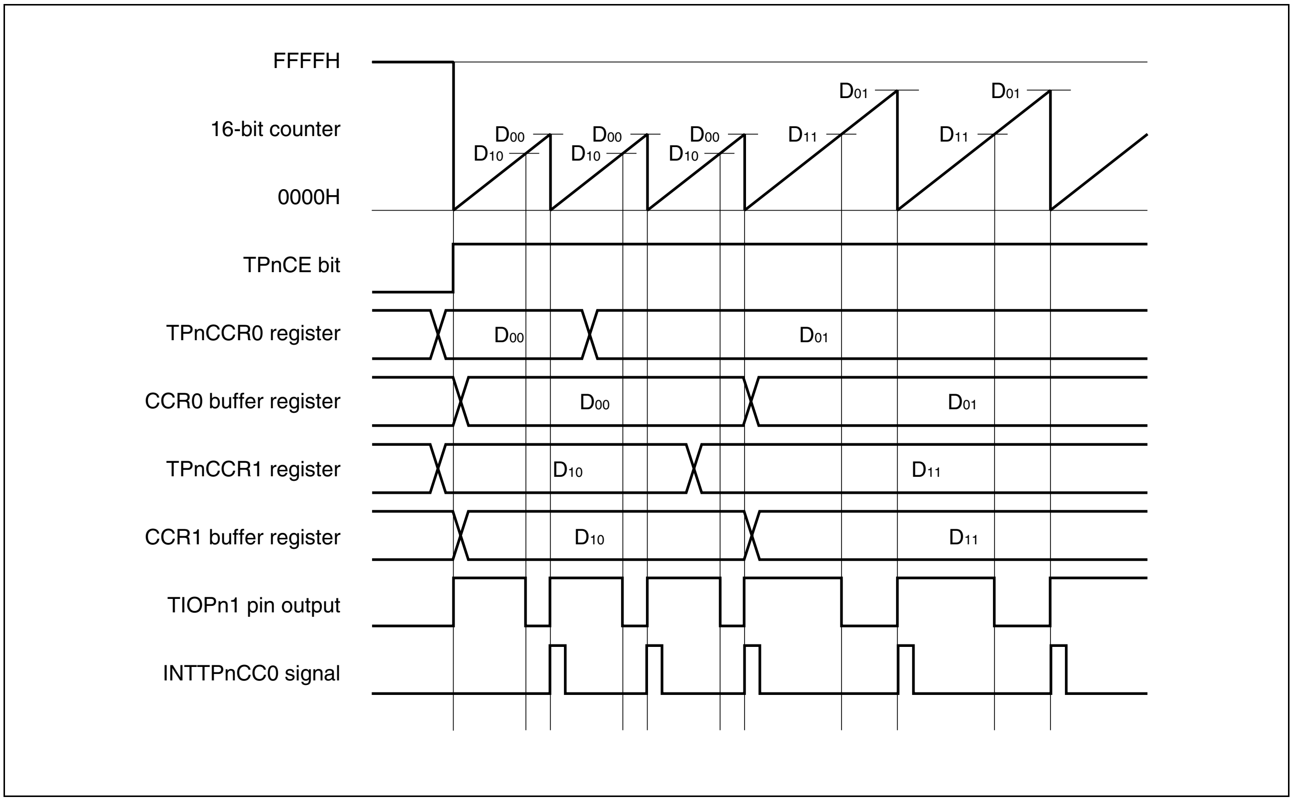
Figure 6-30. Software Processing Flow in PWM Output Mode (2/2)


<1> Count operation start flow
START
Register initial setting
TPnCTL0 register
(TPnCKS0 to TPnCKS2 bits)
TPnCTL1 register,
TPnIOC0 register,
TPnIOC2 register,
TPnCCR0 register,
TPnCCR1 register
Initial setting of these registers is performed before setting the TPnCE bit to 1.
TPnCE bit = 1
The TPnCKS0 to TPnCKS2 bits can be set at the same time when counting is enabled (TPnCE bit = 1).
<2> TPnCCR0, TPnCCR1 register setting change flow
Setting of TPnCCR0 register
TPnCCR1 write processing is necessary only when the set cycle is changed.
Setting of TPnCCR1 register
When the counter is cleared after setting, the value of the TPnCCRm register is transferred to the CCRm buffer register.
<3> TPnCCR0, TPnCCR1 register setting change flow
Setting of TPnCCR1 register
Only writing of the TPnCCR1 register must be performed when the set duty factor is changed. When the counter is cleared after setting, the value of compare register m is transferred to the CCRm buffer register.
<4> TPnCCR0, TPnCCR1 register setting change flow
Setting of TPnCCR0 register
When the counter is cleared after setting, the value of the TPnCCRm register is transferred to the CCRm buffer register.
Setting of TPnCCR1 register
<5> Count operation stop flow
TPnCE bit = 0
Counting is stopped.
STOP
Remark n = 0 to 4
m = 0, 1

(2) PWM output mode operation timing

(a) Changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TPnCCR1 register last.
Rewrite the TPnCCRm register after writing the TPnCCR1 register after the INTTPnCC1 signal is detected.



To transfer data from the TPnCCRm register to the CCRm buffer register, the TPnCCR1 register must be written.

To change both the cycle and active level of the PWM waveform at this time, first set the cycle to the TPnCCR0 register and then set the active level to the TPnCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TPnCCR0 register, and then write the same value to the TPnCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, only the TPnCCR1 register has to be set.

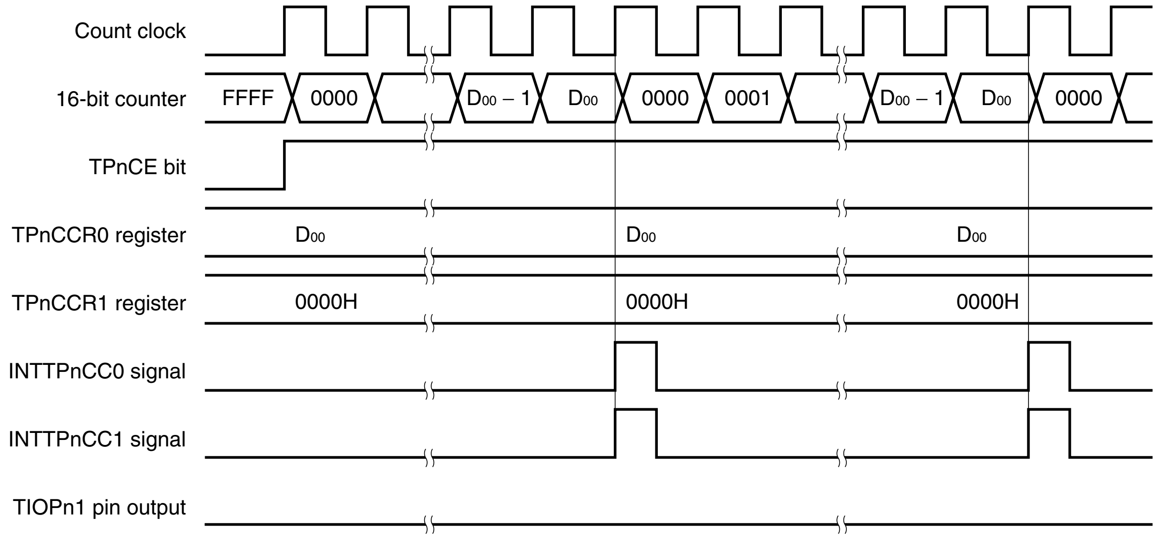
After data is written to the TPnCCR1 register, the value written to the TPnCCRm register is transferred to the CCRm buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TPnCCR0 or TPnCCR1 register again after writing the TPnCCR1 register once, do so after the INTTPnCC0 signal is generated. Otherwise, the value of the CCRm buffer register may become undefined because the timing of transferring data from the TPnCCRm register to the CCRm buffer register conflicts with writing the TPnCCRm register.

Remark n = 0 to 4,
m = 0, 1

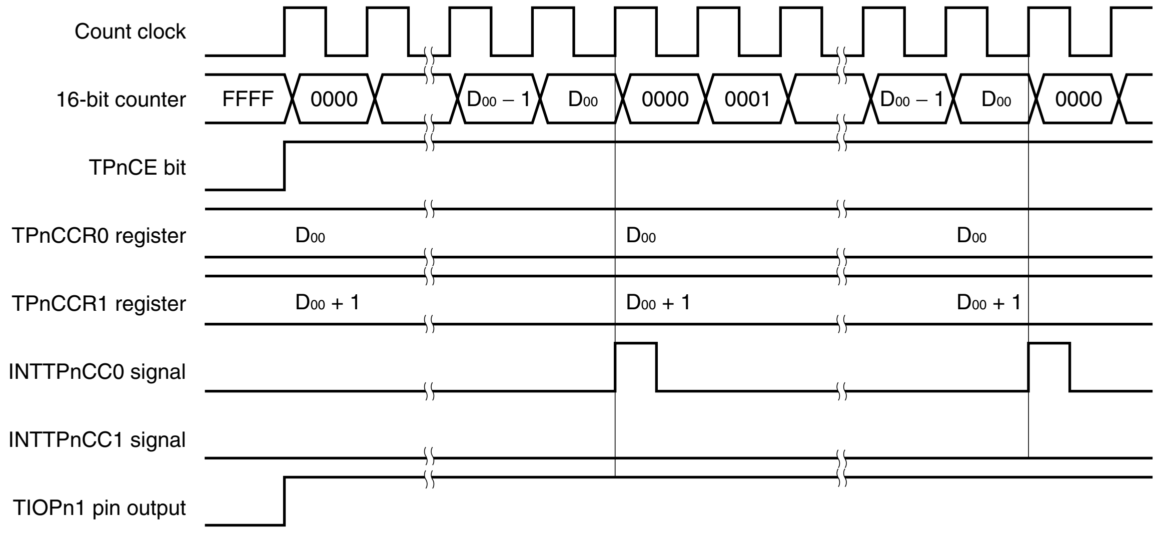
(b) 0%/100% output of PWM waveform

To output a 0% waveform, set the TPnCCR1 register to 0000H. If the set value of the TPnCCR0 register is FFFFH, the INTTPnCC1 signal is generated periodically.



Remark n = 0 to 4

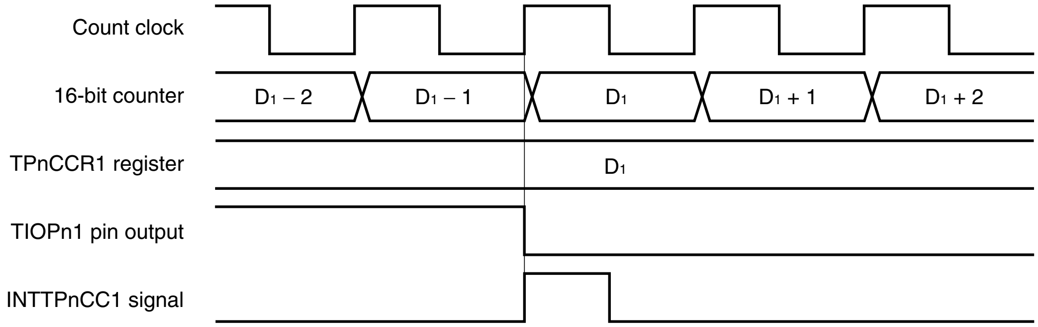
To output a 100% waveform, set a value of (set value of TPnCCR0 register + 1) to the TPnCCR1 register. If the set value of the TPnCCR0 register is FFFFH, 100% output cannot be produced.



Remark n = 0 to 4

(c) Generation timing of compare match interrupt request signal (INTTPnCC1)

The timing of generation of the INTTPnCC1 signal in the PWM output mode differs from the timing of other INTTPnCC1 signals; the INTTPnCC1 signal is generated when the count value of the 16-bit counter matches the value of the TPnCCR1 register.



Remark n = 0 to 4

Usually, the INTTPnCC1 signal is generated in synchronization with the next counting up after the count value of the 16-bit counter matches the value of the TPnCCR1 register.

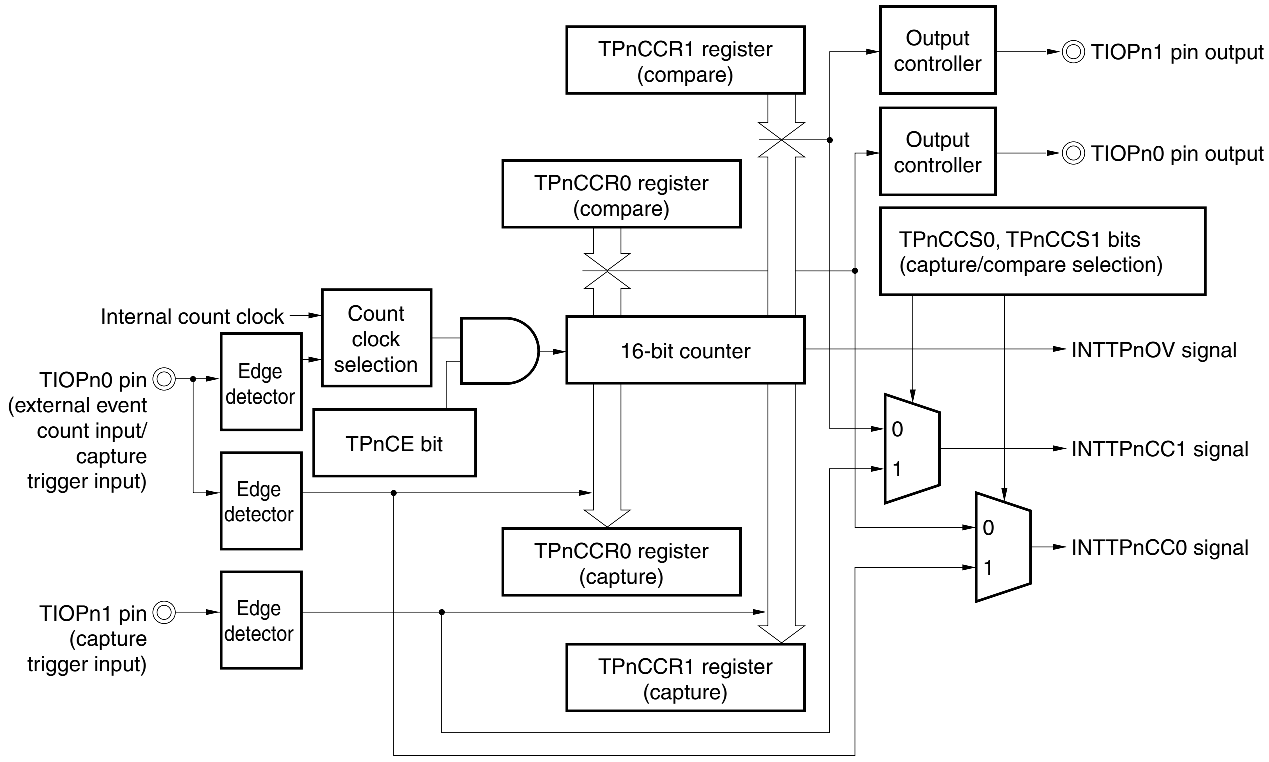
In the PWM output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the output signal of the TIOPn1 pin.

6.7.6 Free-running timer mode (TPnMD2 to TPnMD0 bits = 101)

In the free-running timer mode, 16-bit timer/event counter P starts counting when the TPnCTL0.TPnCE bit is set to 1. At this time, the TPnCCRm register can be used as a compare register or a capture register, depending on the setting of the TPnOPT0.TPnCCS0 and TPnOPT0.TPnCCS1 bits.

Figure 6-31. Configuration in Free-running Timer Mode

<R>



<R>

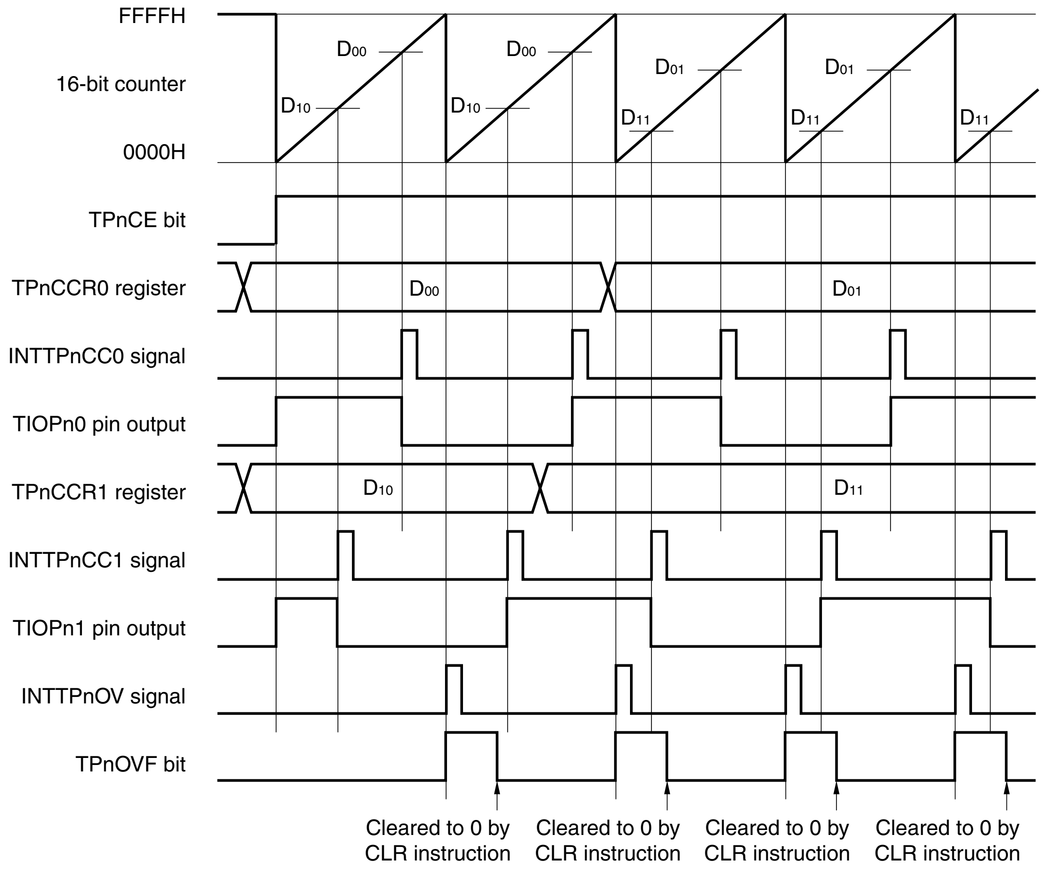
Remark n = 0 to 4
m = 0, 1

When the TPnCE bit is set to 1, 16-bit timer/event counter P starts counting, and the output signals of the TIOPn0 and TIOPn1 pins are inverted. When the count value of the 16-bit counter later matches the set value of the TPnCCRm register, a compare match interrupt request signal (INTTPnCCm) is generated, and the output signal of the TIOPnm pin is inverted.

The 16-bit counter continues counting in synchronization with the count clock. When it counts up to FFFFH, it generates an overflow interrupt request signal (INTTPnOV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TPnOPT0.TPnOVF bit) is also set to 1. Clear the overflow flag to 0 by executing the CLR instruction by software.

The TPnCCRm register can be rewritten while the counter is operating. If it is rewritten, the new value is reflected at that time, and compared with the count value.

Figure 6-32. Basic Timing in Free-running Timer Mode (Compare Function)

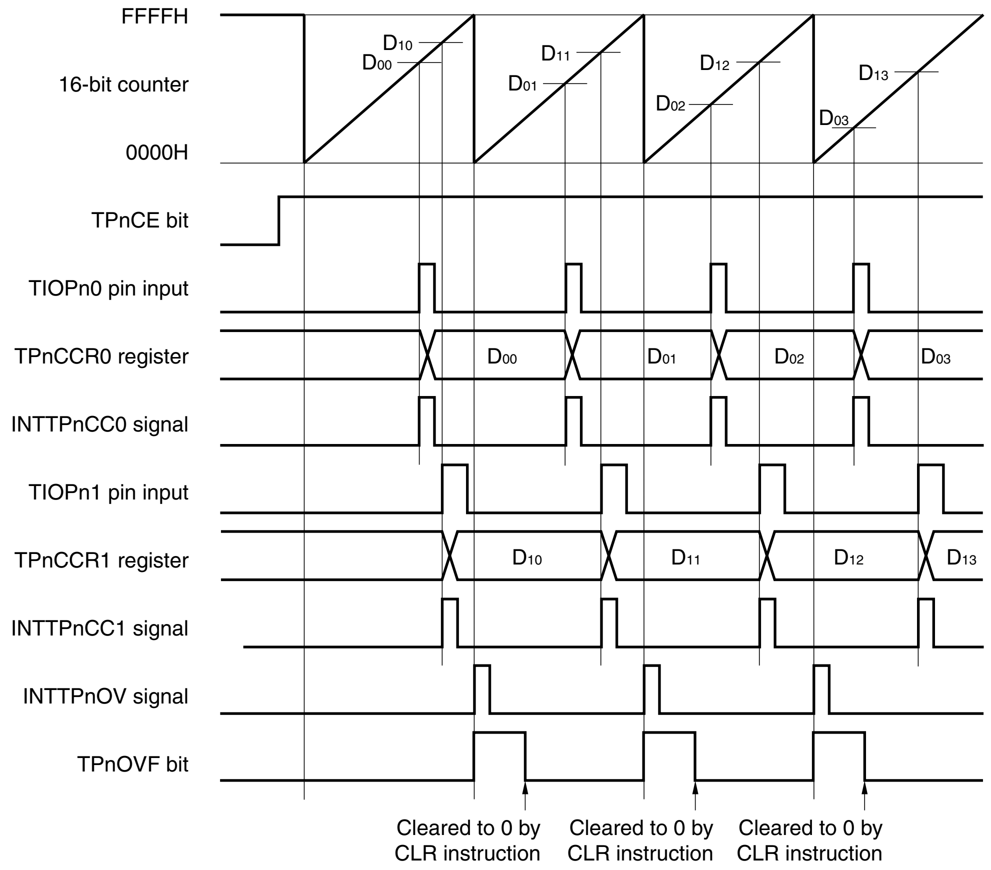


Remark n = 0 to 4
m = 0, 1

When the TPnCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIOPnm pin is detected, the count value of the 16-bit counter is stored in the TPnCCRm register, and a capture interrupt request signal (INTTPnCCm) is generated.

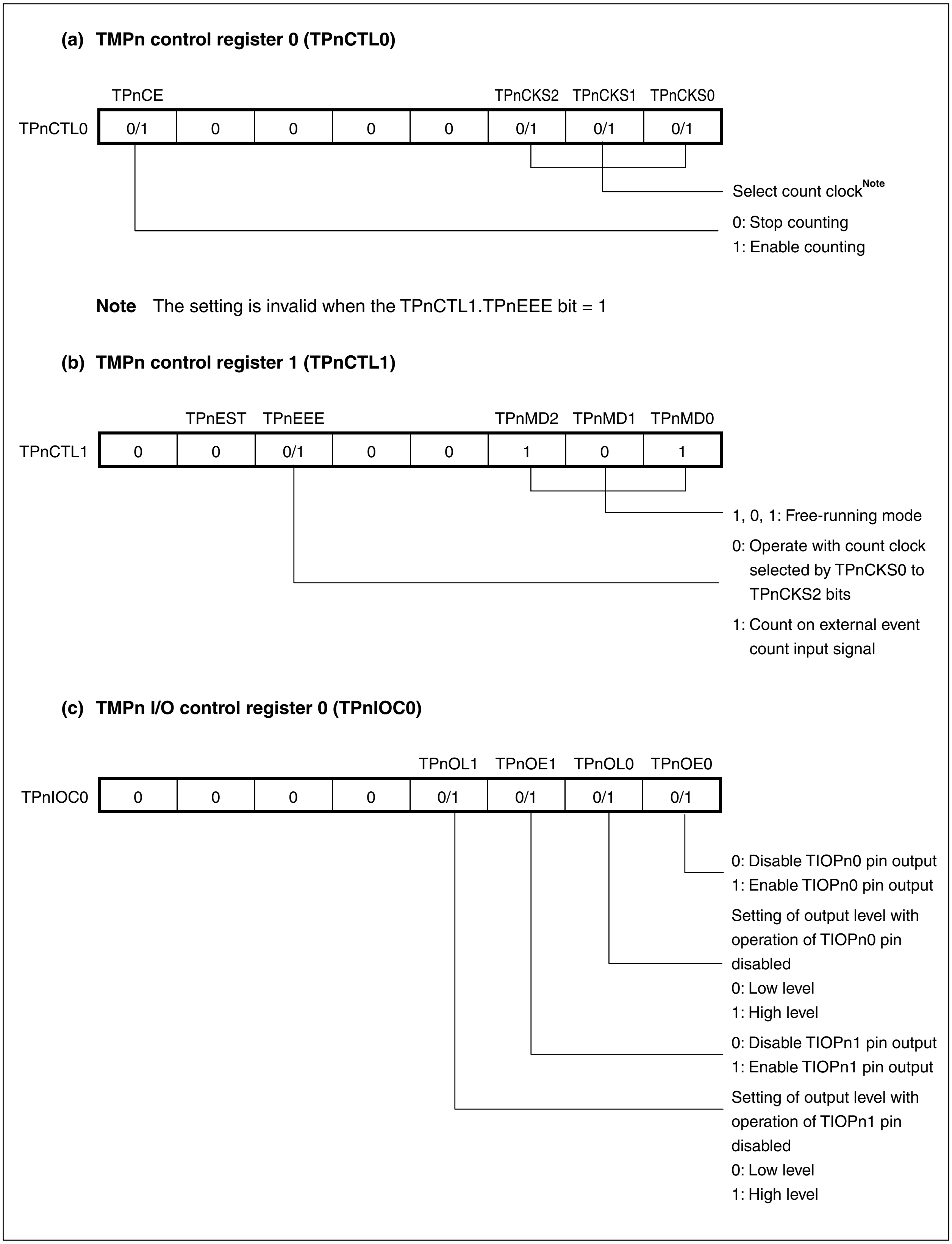
The 16-bit counter continues counting in synchronization with the count clock. When it counts up to FFFFH, it generates an overflow interrupt request signal (INTTPnOV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TPnOPT0.TPnOVF bit) is also set to 1. Clear the overflow flag to 0 by executing the CLR instruction by software.

Figure 6-33. Basic Timing in Free-running Timer Mode (Capture Function)



Remark n = 0 to 4

Figure 6-34. Register Setting in Free-running Timer Mode (1/2)



(a) TMPn control register 0 (TPnCTL0)

	TPnCE					TPnCKS2	TPnCKS1	TPnCKS0
TPnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

TPnCKS2 to TPnCKS0: Select count clock[Note]

TPnCE:
0: Stop counting
1: Enable counting

Note The setting is invalid when the TPnCTL1.TPnEEE bit = 1

(b) TMPn control register 1 (TPnCTL1)

		TPnEST	TPnEEE			TPnMD2	TPnMD1	TPnMD0
TPnCTL1	0	0	0/1	0	0	1	0	1

TPnMD2 to TPnMD0: 1, 0, 1: Free-running mode

TPnEEE:
0: Operate with count clock selected by TPnCKS0 to TPnCKS2 bits
1: Count on external event count input signal

(c) TMPn I/O control register 0 (TPnIOC0)

					TPnOL1	TPnOE1	TPnOL0	TPnOE0
TPnIOC0	0	0	0	0	0/1	0/1	0/1	0/1

TPnOE0:
0: Disable TIOPn0 pin output
1: Enable TIOPn0 pin output

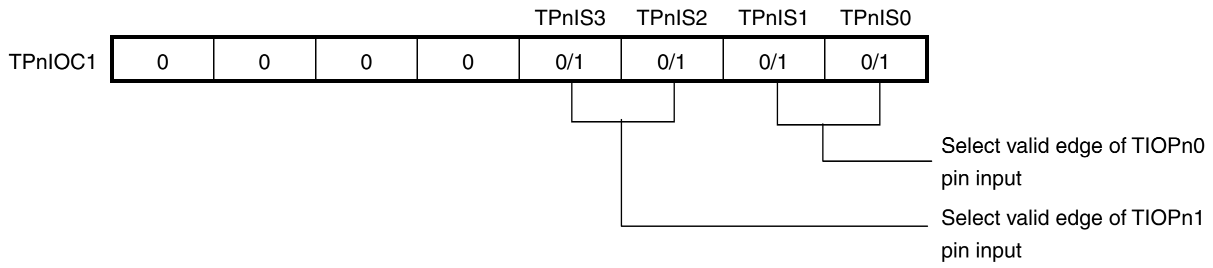
TPnOL0:
Setting of output level with operation of TIOPn0 pin disabled
0: Low level
1: High level

TPnOE1:
0: Disable TIOPn1 pin output
1: Enable TIOPn1 pin output

TPnOL1:
Setting of output level with operation of TIOPn1 pin disabled
0: Low level
1: High level

Figure 6-34. Register Setting in Free-running Timer Mode (2/2)

(d) TMPn I/O control register 1 (TPnIOC1)

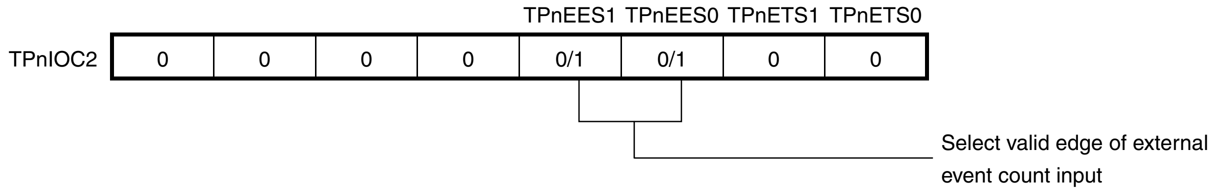


					TPnIS3	TPnIS2	TPnIS1	TPnIS0
TPnIOC1	0	0	0	0	0/1	0/1	0/1	0/1

Select valid edge of TIOPn0 pin input

Select valid edge of TIOPn1 pin input

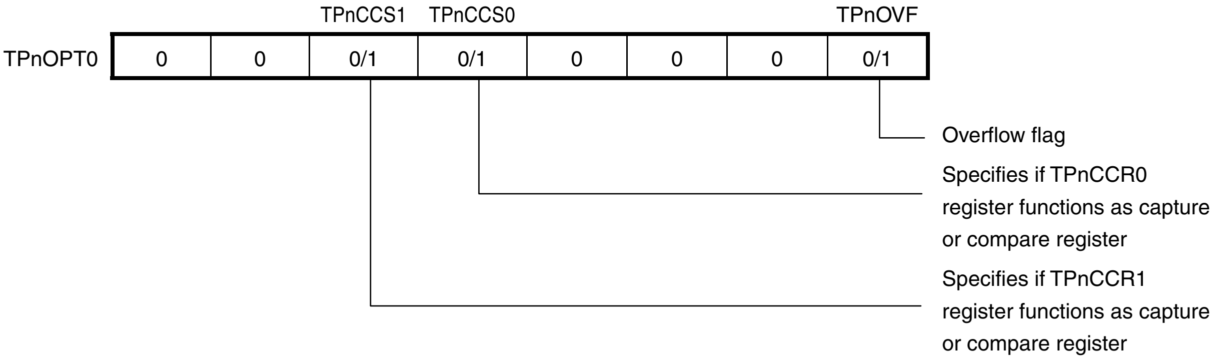
(e) TMPn I/O control register 2 (TPnIOC2)



					TPnEES1	TPnEES0	TPnETS1	TPnETS0
TPnIOC2	0	0	0	0	0/1	0/1	0	0

Select valid edge of external event count input

(f) TMPn option register 0 (TPnOPT0)



			TPnCCS1	TPnCCS0				TPnOVF
TPnOPT0	0	0	0/1	0/1	0	0	0	0/1

Overflow flag

Specifies if TPnCCR0 register functions as capture or compare register

Specifies if TPnCCR1 register functions as capture or compare register

(g) TMPn counter read buffer register (TPnCNT)

The value of the 16-bit counter can be read by reading the TPnCNT register.

(h) TMPn capture/compare registers 0 and 1 (TPnCCR0 and TPnCCR1)

These registers function as capture registers or compare registers depending on the setting of the TPnOPT0.TPnCCSm bit.

When the registers function as capture registers, they store the count value of the 16-bit counter when the valid edge input to the TIOPnm pin is detected.

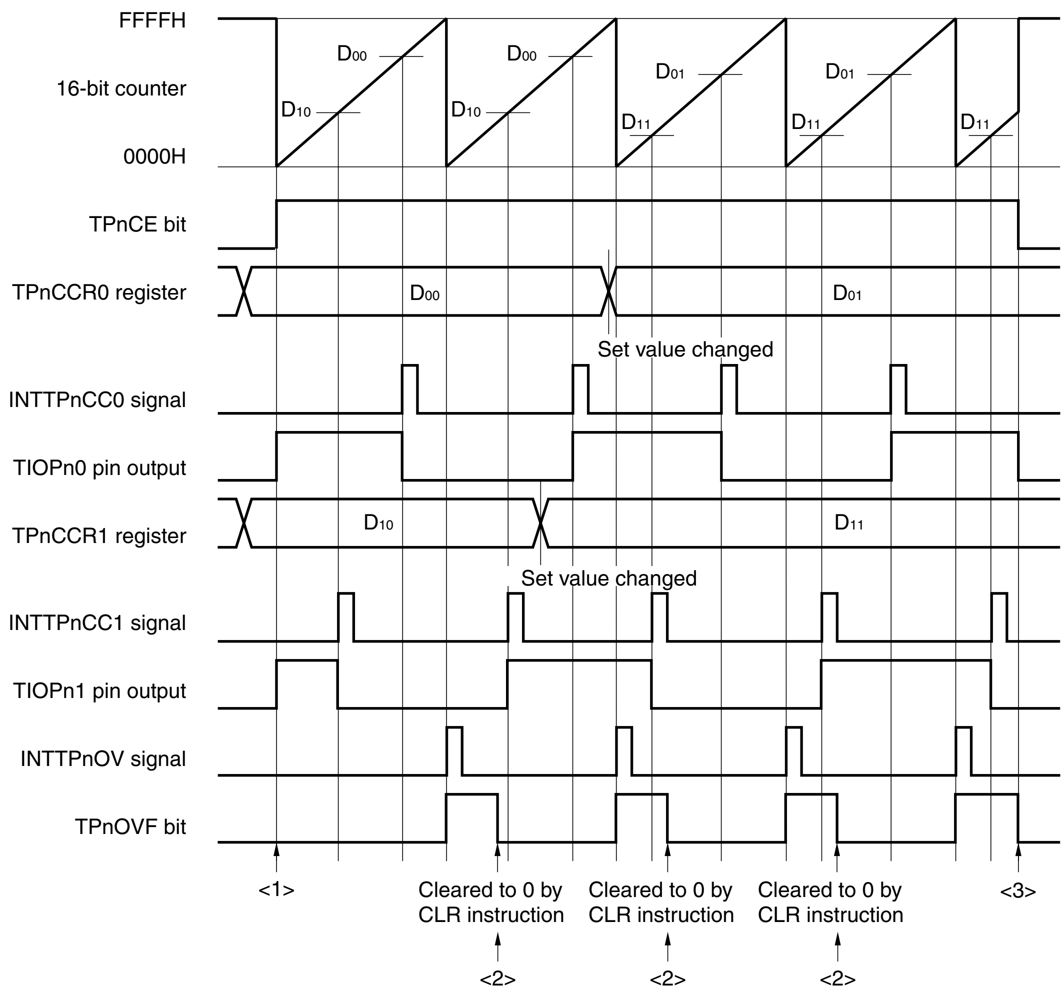
When the registers function as compare registers and when D_m is set to the TPnCCRm register, the INTTPnCCm signal is generated when the counter reaches (D_m + 1), and the output signal of the TIOPnm pin is inverted.

Remark n = 0 to 4
m = 0, 1

(1) Operation flow in free-running timer mode

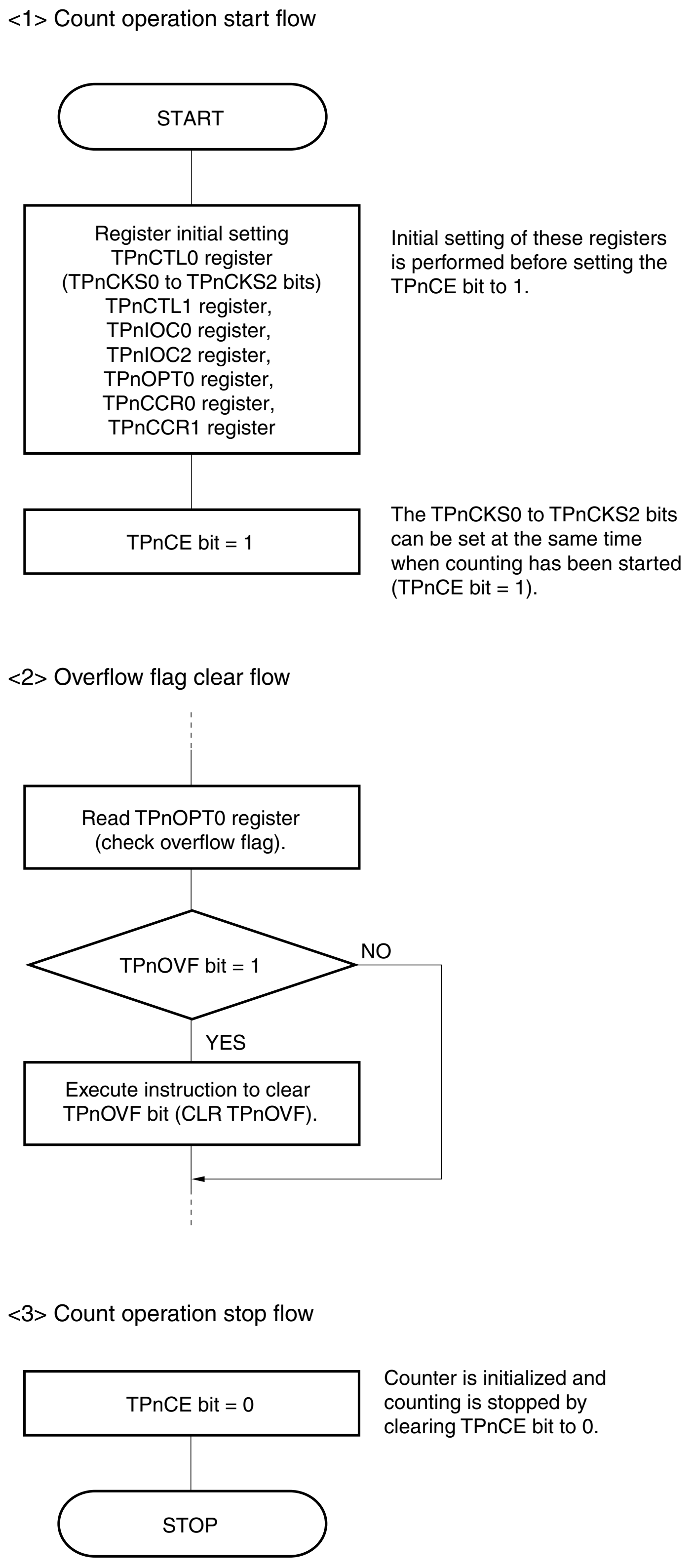
(a) When using capture/compare register as compare register

Figure 6-35. Software Processing Flow in Free-running Timer Mode (Compare Function) (1/2)



Remark n = 0 to 4

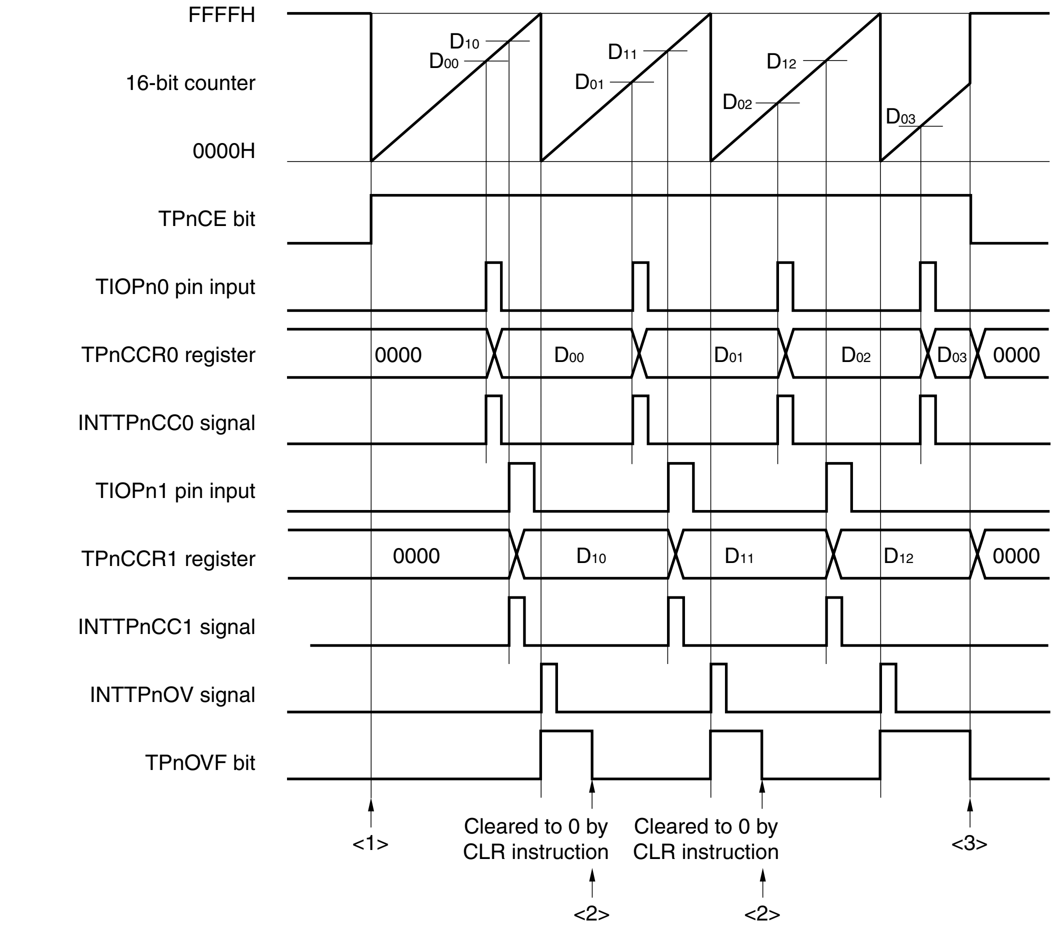
Figure 6-35. Software Processing Flow in Free-running Timer Mode (Compare Function) (2/2)


<1> Count operation start flow
START
Register initial setting
TPnCTL0 register
(TPnCKS0 to TPnCKS2 bits)
TPnCTL1 register,
TPnIOC0 register,
TPnIOC2 register,
TPnOPT0 register,
TPnCCR0 register,
TPnCCR1 register
Initial setting of these registers is performed before setting the TPnCE bit to 1.
TPnCE bit = 1
The TPnCKS0 to TPnCKS2 bits can be set at the same time when counting has been started (TPnCE bit = 1).
<2> Overflow flag clear flow
Read TPnOPT0 register (check overflow flag).
TPnOVF bit = 1
NO
YES
Execute instruction to clear TPnOVF bit (CLR TPnOVF).
<3> Count operation stop flow
TPnCE bit = 0
Counter is initialized and counting is stopped by clearing TPnCE bit to 0.
STOP

Remark n = 0 to 4

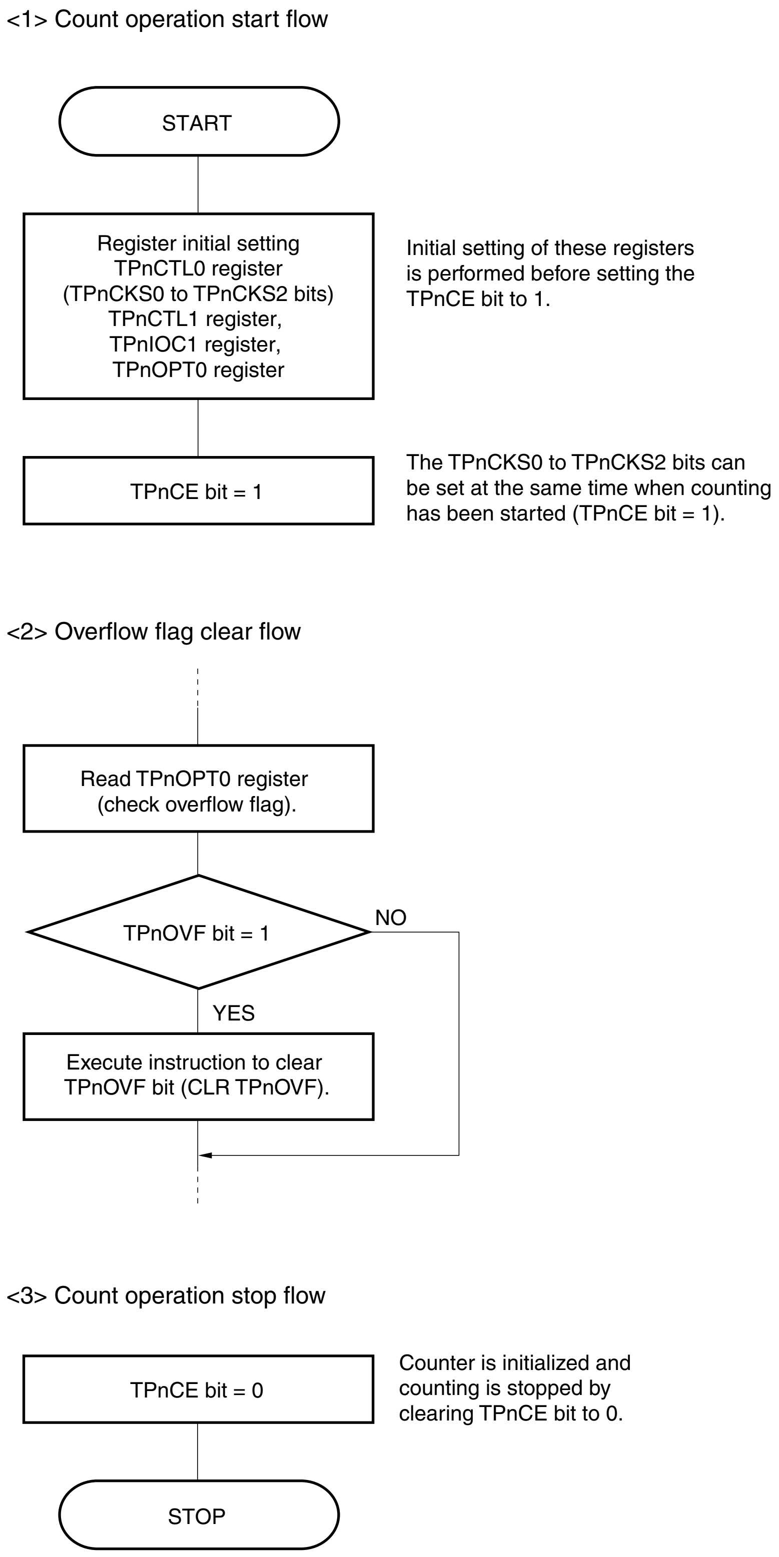
(b) When using capture/compare register as capture register

Figure 6-36. Software Processing Flow in Free-running Timer Mode (Capture Function) (1/2)



Remark n = 0 to 4

Figure 6-36. Software Processing Flow in Free-running Timer Mode (Capture Function) (2/2)



<1> Count operation start flow

Initial setting of these registers is performed before setting the TPnCE bit to 1.

The TPnCKS0 to TPnCKS2 bits can be set at the same time when counting has been started (TPnCE bit = 1).

<2> Overflow flag clear flow

<3> Count operation stop flow

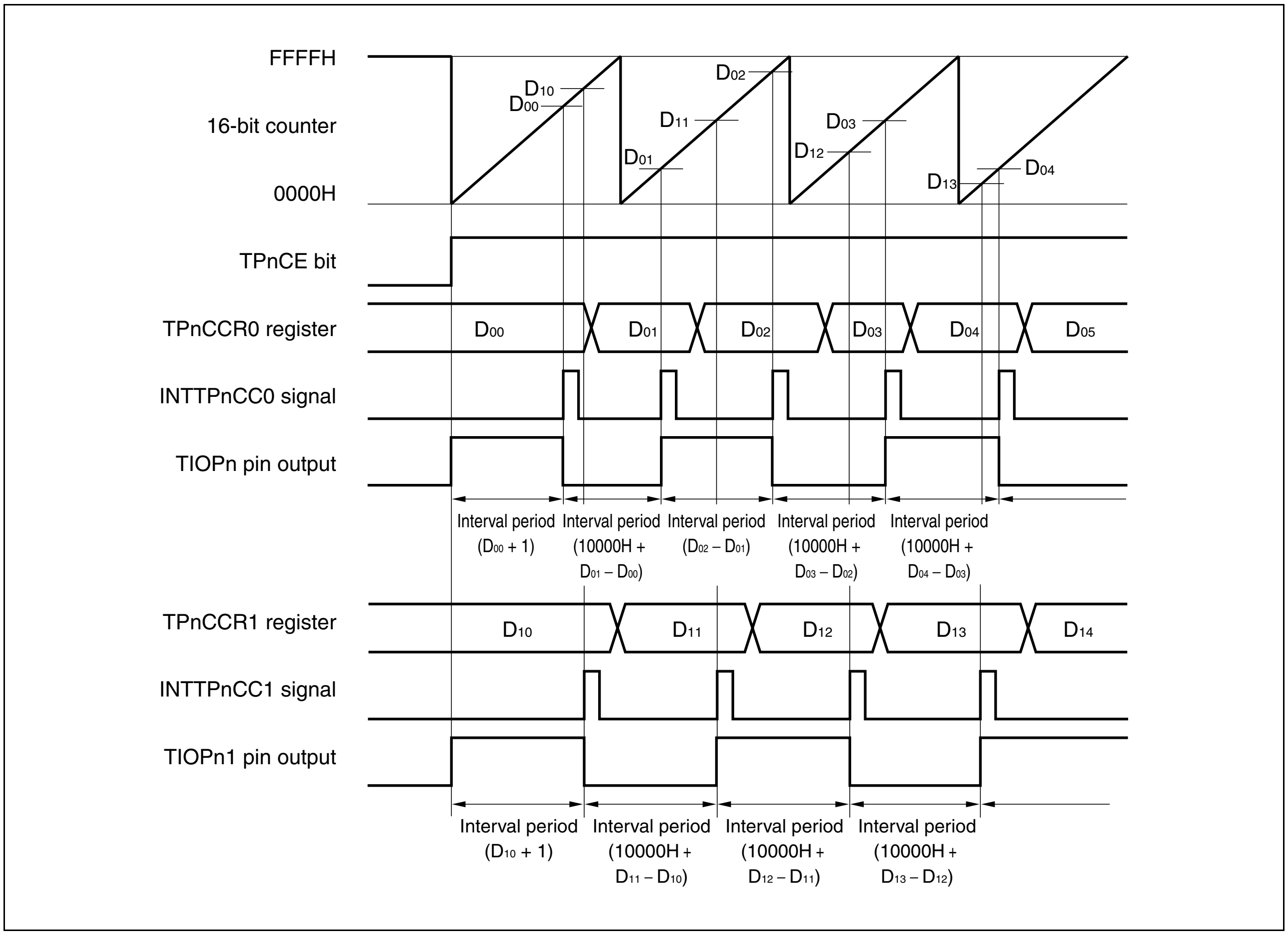
Counter is initialized and counting is stopped by clearing TPnCE bit to 0.

Remark n = 0 to 4

(2) Operation timing in free-running timer mode

(a) Interval operation with compare register

When 16-bit timer/event counter P is used as an interval timer with the TPnCCRm register used as a compare register, software processing is necessary for setting a comparison value to generate the next interrupt request signal each time the INTTPnCCm signal has been detected.


FFFFH
16-bit counter
0000H
D10
D00
D01
D11
D02
D12
D03
D13
D04
TPnCE bit
TPnCCR0 register
D00
D01
D02
D03
D04
D05
INTTPnCC0 signal
TIOPn pin output
Interval period (D00 + 1)
Interval period (10000H + D01 – D00)
Interval period (D02 – D01)
Interval period (10000H + D03 – D02)
Interval period (10000H + D04 – D03)
TPnCCR1 register
D10
D11
D12
D13
D14
INTTPnCC1 signal
TIOPn1 pin output
Interval period (D10 + 1)
Interval period (10000H + D11 – D10)
Interval period (10000H + D12 – D11)
Interval period (10000H + D13 – D12)

When performing an interval operation in the free-running timer mode, two intervals can be set with one channel.

To perform the interval operation, the value of the corresponding TPnCCRm register must be re-set in the interrupt servicing that is executed when the INTTPnCCm signal is detected.

The set value for re-setting the TPnCCRm register can be calculated by the following expression, where "D_m" is the interval period.

Compare register default value: D_m – 1

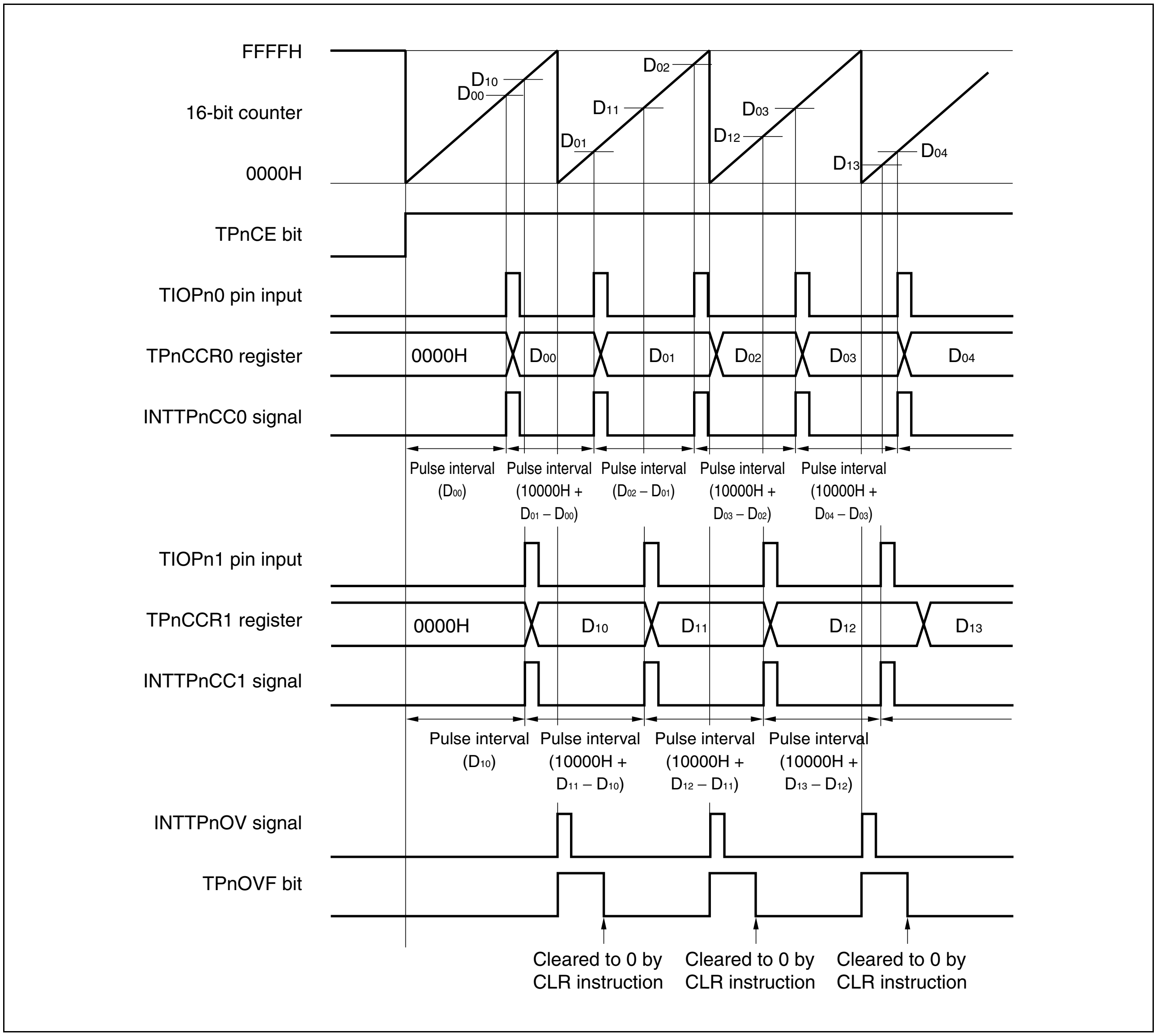
Value set to compare register second and subsequent time: Previous set value + D_m

(If the calculation result is greater than FFFFH, subtract 10000H from the result and set this value to the register.)

Remark n = 0 to 4
m = 0, 1

(b) Pulse width measurement with capture register

When pulse width measurement is performed with the TPnCCRm register used as a capture register, software processing is necessary for reading the capture register each time the INTTPnCCm signal has been detected and for calculating an interval.


FFFFH
16-bit counter
0000H
D10
D00
D02
D11
D01
D03
D12
D13
D04
TPnCE bit
TIOPn0 pin input
TPnCCR0 register
0000H
D00
D01
D02
D03
D04
INTTPnCC0 signal
Pulse interval (D00)
Pulse interval (10000H + D01 – D00)
Pulse interval (D02 – D01)
Pulse interval (10000H + D03 – D02)
Pulse interval (10000H + D04 – D03)
TIOPn1 pin input
TPnCCR1 register
0000H
D10
D11
D12
D13
INTTPnCC1 signal
Pulse interval (D10)
Pulse interval (10000H + D11 – D10)
Pulse interval (10000H + D12 – D11)
Pulse interval (10000H + D13 – D12)
INTTPnOV signal
TPnOVF bit
Cleared to 0 by CLR instruction
Cleared to 0 by CLR instruction
Cleared to 0 by CLR instruction

When executing pulse width measurement in the free-running timer mode, two pulse widths can be measured with one channel.

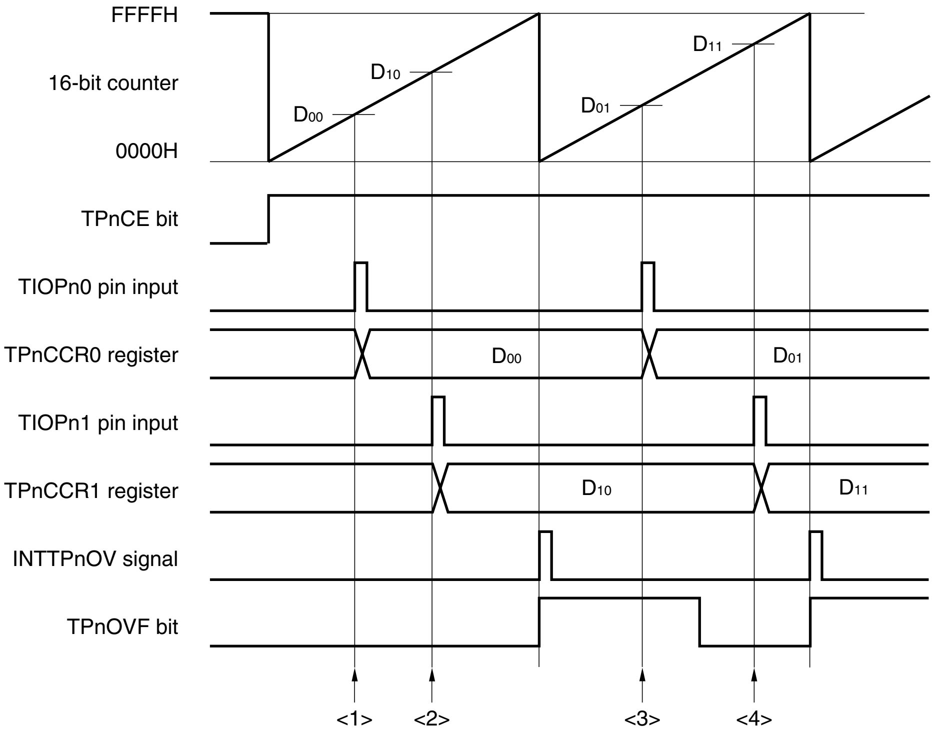
To measure a pulse width, the pulse width can be calculated by reading the value of the TPnCCRm register in synchronization with the INTTPnCCm signal, and calculating the difference between the read value and the previously read value.

Remark n = 0 to 4
m = 0, 1

(c) Processing of overflow when two capture registers are used

Care must be exercised in processing the overflow flag when two capture registers are used. First, an example of incorrect processing is shown below.

Example of incorrect processing when two capture registers are used



The following problem may occur when two pulse widths are measured in the free-running timer mode.

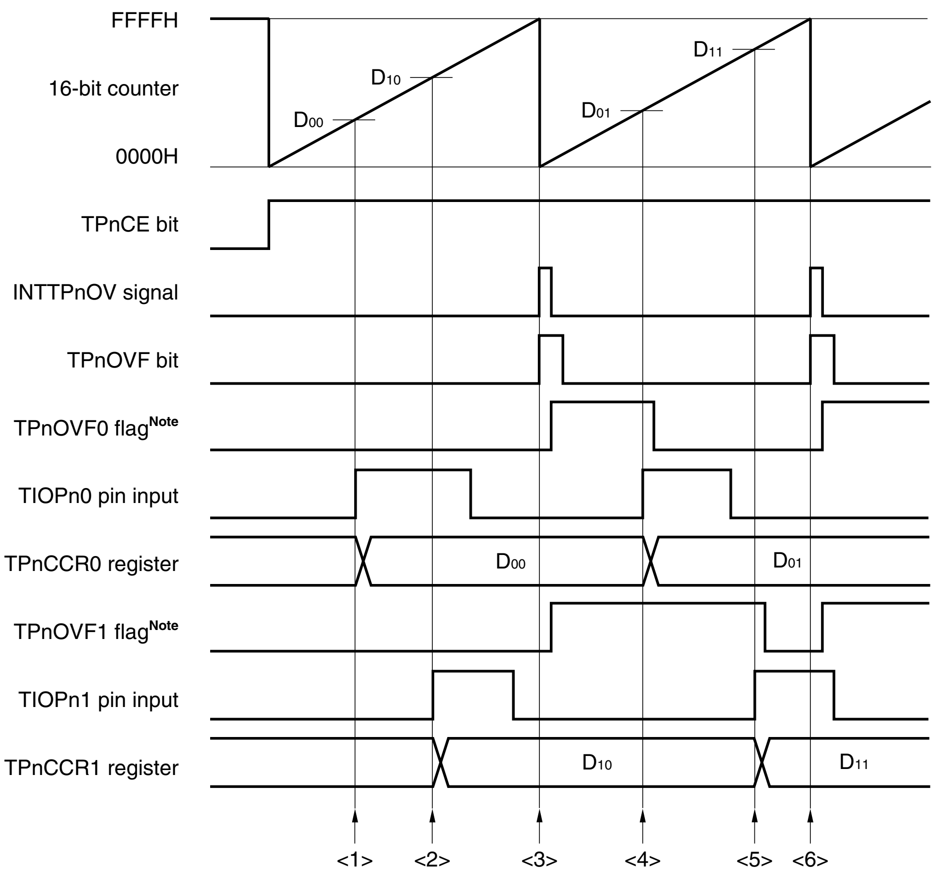
<1> Read the TPnCCR0 register (setting of the default value of the TIOPn0 pin input).
<2> Read the TPnCCR1 register (setting of the default value of the TIOPn1 pin input).
<3> Read the TPnCCR0 register.
Read the overflow flag. If the overflow flag is 1, clear it to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{01} – D_{00}).
<4> Read the TPnCCR1 register.
Read the overflow flag. Because the flag is cleared in <3>, 0 is read.
Because the overflow flag is 0, the pulse width can be calculated by (D_{11} – D_{10}) (incorrect).

When two capture registers are used, and if the overflow flag is cleared to 0 by one capture register, the other capture register may not obtain the correct pulse width.

Use software when using two capture registers. An example of how to use software is shown below.

(1/2)

Example when two capture registers are used (using overflow interrupt)



Note The TPnOVF0 and TPnOVF1 flags are set on the internal RAM by software.

<1> Read the TPnCCR0 register (setting of the default value of the TIOPn0 pin input).

<2> Read the TPnCCR1 register (setting of the default value of the TIOPn1 pin input).

<3> An overflow occurs. Set the TPnOVF0 and TPnOVF1 flags to 1 in the overflow interrupt servicing, and clear the overflow flag to 0.

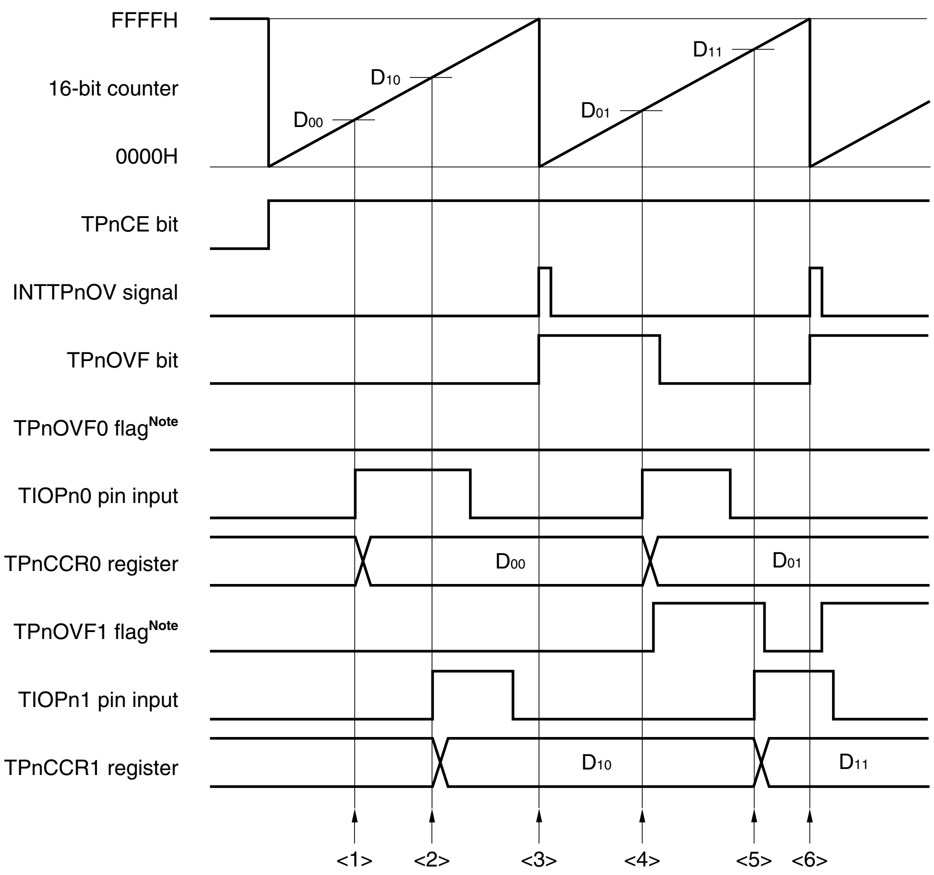
<4> Read the TPnCCR0 register.
Read the TPnOVF0 flag. If the TPnOVF0 flag is 1, clear it to 0.
Because the TPnOVF0 flag is 1, the pulse width can be calculated by (10000H + D_{01} – D_{00}).

<5> Read the TPnCCR1 register.
Read the TPnOVF1 flag. If the TPnOVF1 flag is 1, clear it to 0 (the TPnOVF0 flag is cleared in <4>, and the TPnOVF1 flag remains 1).
Because the TPnOVF1 flag is 1, the pulse width can be calculated by (10000H + D_{11} – D_{10}) (correct).

<6> Same as <3>

(2/2)

Example when two capture registers are used (without using overflow interrupt)



Note The TPnOVF0 and TPnOVF1 flags are set on the internal RAM by software.

<1> Read the TPnCCR0 register (setting of the default value of the TIOPn0 pin input).

<2> Read the TPnCCR1 register (setting of the default value of the TIOPn1 pin input).

<3> An overflow occurs. Nothing is done by software.

<4> Read the TPnCCR0 register.
Read the overflow flag. If the overflow flag is 1, set only the TPnOVF1 flag to 1, and clear the overflow flag to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{01} – D_{00}).

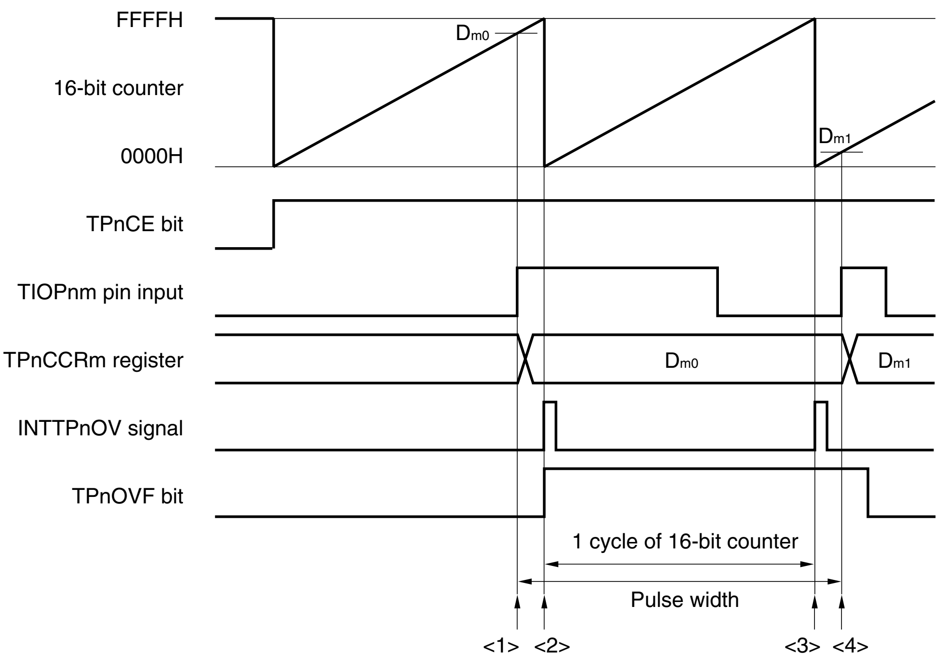
<5> Read the TPnCCR1 register.
Read the overflow flag. Because the overflow flag is cleared in <4>, 0 is read.
Read the TPnOVF1 flag. If the TPnOVF1 flag is 1, clear it to 0.
Because the TPnOVF1 flag is 1, the pulse width can be calculated by (10000H + D_{11} – D_{10}) (correct).

<6> Same as <3>

(d) Processing of overflow if capture trigger interval is long

If the pulse width is greater than one cycle of the 16-bit counter, care must be exercised because an overflow may occur more than once from the first capture trigger to the next. First, an example of incorrect processing is shown below.

Example of incorrect processing when capture trigger interval is long



The following problem may occur when long pulse width is measured in the free-running timer mode.

<1> Read the TPnCCRm register (setting of the default value of the TIOPnm pin input).

<2> An overflow occurs. Nothing is done by software.

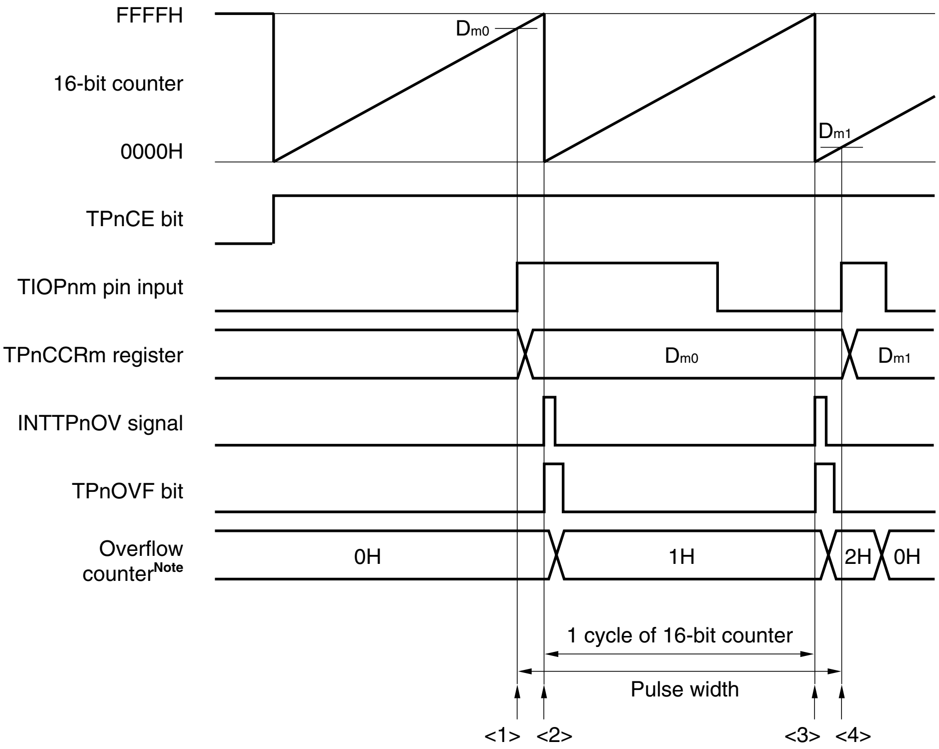
<3> An overflow occurs a second time. Nothing is done by software.

<4> Read the TPnCCRm register.
Read the overflow flag. If the overflow flag is 1, clear it to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{m1} – D_{m0}) (incorrect).
Actually, the pulse width must be (20000H + D_{m1} – D_{m0}) because an overflow occurs twice.

If an overflow occurs twice or more when the capture trigger interval is long, the correct pulse width may not be obtained.

If the capture trigger interval is long, slow the count clock to lengthen one cycle of the 16-bit counter, or use software. An example of how to use software is shown next.

Example when capture trigger interval is long



Note The overflow counter is set arbitrarily by software on the internal RAM.

<1> Read the TPnCCRm register (setting of the default value of the TIOPnm pin input).

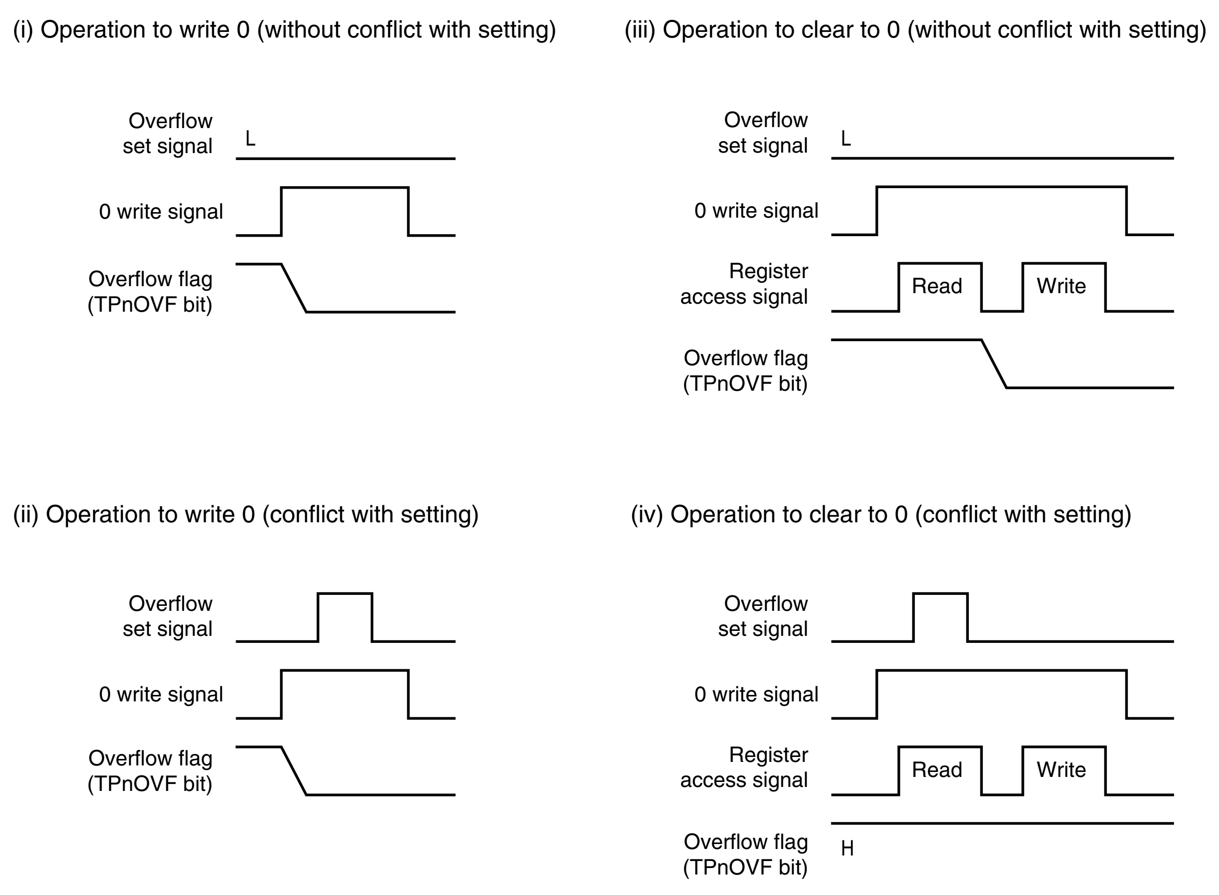
<2> An overflow occurs. Increment the overflow counter and clear the overflow flag to 0 in the overflow interrupt servicing.

<3> An overflow occurs a second time. Increment (+1) the overflow counter and clear the overflow flag to 0 in the overflow interrupt servicing.

<4> Read the TPnCCRm register.
Read the overflow counter.
→ When the overflow counter is “N”, the pulse width can be calculated by (N × 10000H + D_{m1} – D_{m0}).
In this example, the pulse width is (20000H + D_{m1} – D_{m0}) because an overflow occurs twice.
Clear the overflow counter (0H).

(e) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TPnOVF bit to 0 with the CLR instruction and by writing 8-bit data (bit 0 is 0) to the TPnOPT0 register. To accurately detect an overflow, read the TPnOVF bit when it is 1, and then clear the overflow flag by using a bit manipulation instruction.


(i) Operation to write 0 (without conflict with setting)
Overflow set signal
L
0 write signal
Overflow flag (TPnOVF bit)
(iii) Operation to clear to 0 (without conflict with setting)
Overflow set signal
L
0 write signal
Register access signal
Read
Write
Overflow flag (TPnOVF bit)
(ii) Operation to write 0 (conflict with setting)
Overflow set signal
0 write signal
Overflow flag (TPnOVF bit)
(iv) Operation to clear to 0 (conflict with setting)
Overflow set signal
0 write signal
Register access signal
Read
Write
Overflow flag (TPnOVF bit)
H

Remark n = 0 to 4

To clear the overflow flag to 0, read the overflow flag to check if it is set to 1, and clear it with the CLR instruction. If 0 is written to the overflow flag without checking if the flag is 1, the set information of overflow may be erased by writing 0 ((ii) in the above chart). Therefore, software may judge that no overflow has occurred even when an overflow actually has occurred.

If execution of the CLR instruction conflicts with occurrence of an overflow when the overflow flag is cleared to 0 with the CLR instruction, the overflow flag remains set even after execution of the clear instruction.

6.7.7 Pulse width measurement mode (TPnMD2 to TPnMD0 bits = 110)

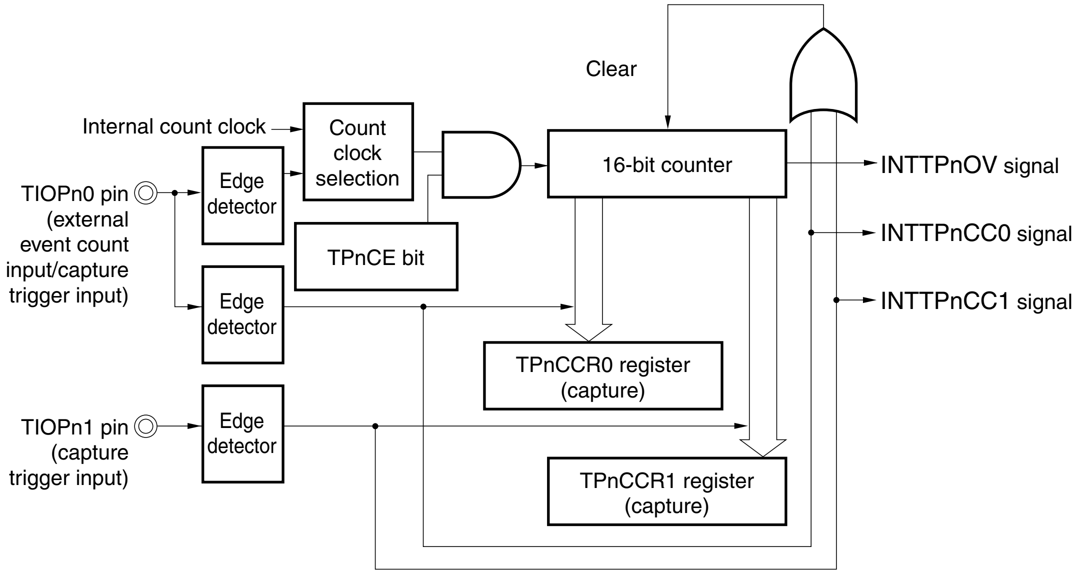
In the pulse width measurement mode, 16-bit timer/event counter P starts counting when the TPnCTL0.TPnCE bit is set to 1. Each time the valid edge input to the TIOPnm pin has been detected, the count value of the 16-bit counter is stored in the TPnCCRm register, and the 16-bit counter is cleared to 0000H.

The interval of the valid edge can be measured by reading the TPnCCRm register after a capture interrupt request signal (INTTPnCCm) occurs.

Select either the TIOPn0 or TIOPn1 pin as the capture trigger input pin. Specify “No edge detected” by using the TPnIOC1 register for the unused pins.

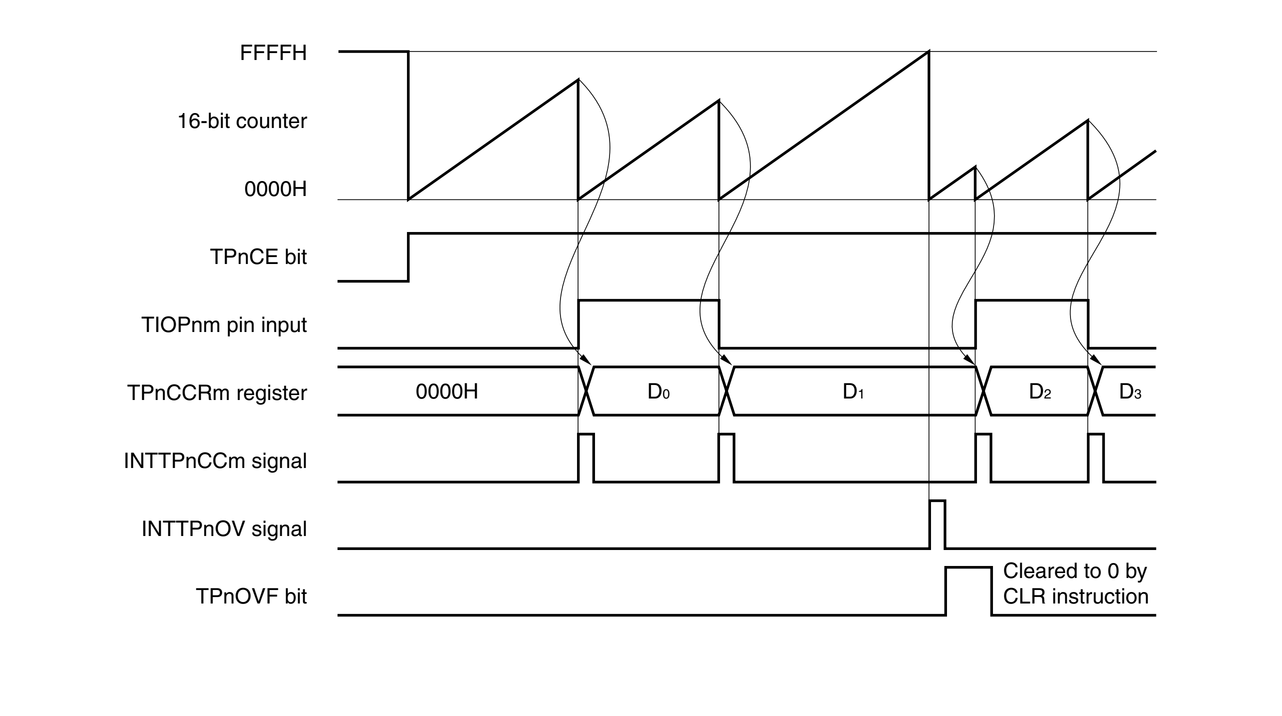
When an external clock is used as the count clock, measure the pulse width of the TIOPn1 pin because the external clock is fixed to the TIOPn0 pin. At this time, clear the TPnIOC1.TPnIS1 and TPnIOC1.TPnIS0 bits to 00 (capture trigger input (TIOPn0 pin): No edge detected).

Figure 6-37. Configuration in Pulse Width Measurement Mode



Remark n = 0 to 4
m = 0, 1

Figure 6-38. Basic Timing in Pulse Width Measurement Mode



When the TPnCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIOPnm pin is later detected, the count value of the 16-bit counter is stored in the TPnCCRm register, the 16-bit counter is cleared to 0000H, and a capture interrupt request signal (INTTPnCCm) is generated.

The pulse width is calculated as follows.

Pulse width = Captured value × Count clock cycle

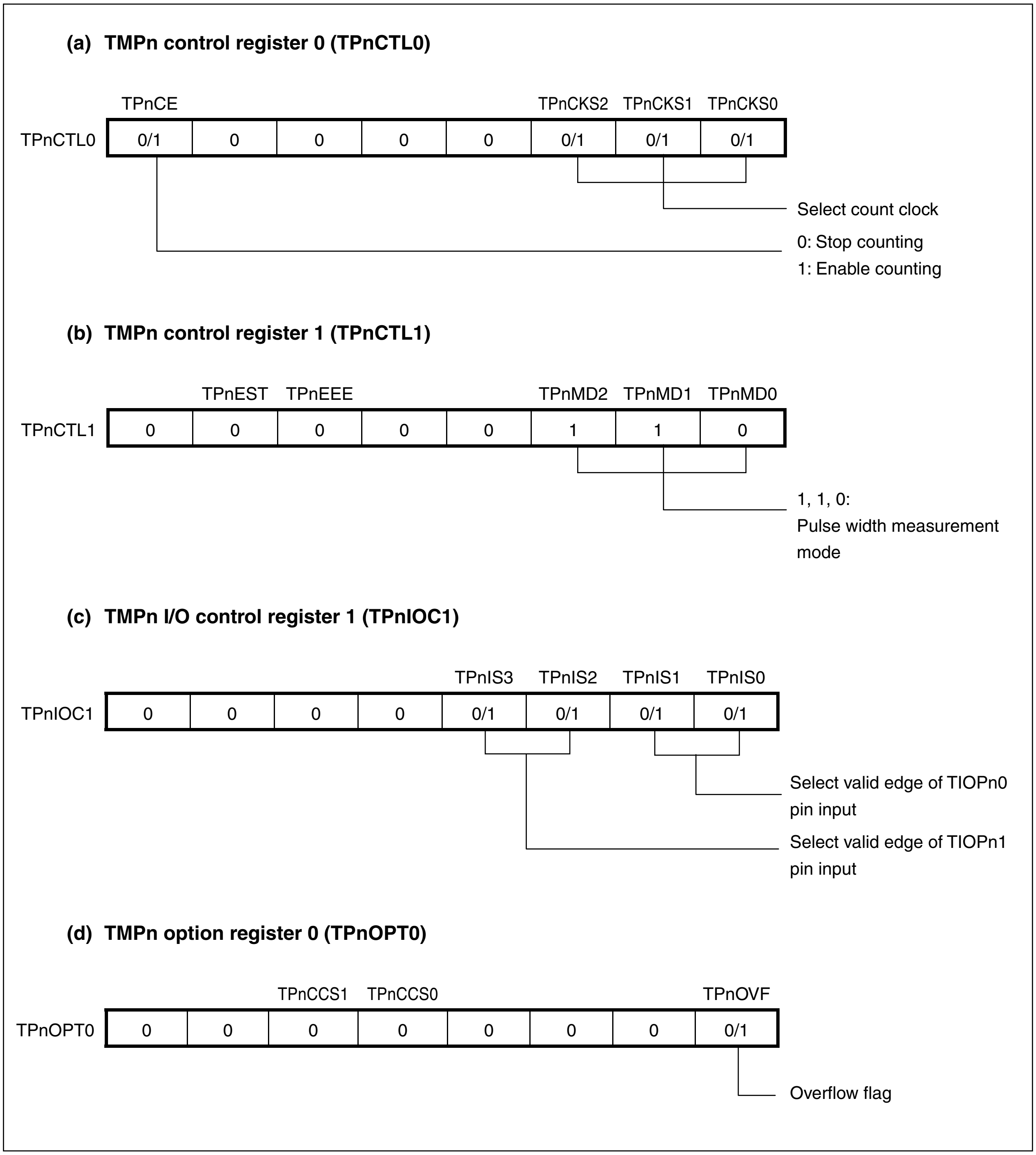
If the valid edge is not input to the TIOPnm pin even when the 16-bit counter counted up to FFFFH, an overflow interrupt request signal (INTTPnOV) is generated at the next count clock, and the counter is cleared to 0000H and continues counting. At this time, the overflow flag (TPnOPT0.TPnOVF bit) is also set to 1. Clear the overflow flag to 0 by executing the CLR instruction via software.

If the overflow flag is set to 1, the pulse width can be calculated as follows.

Pulse width = (10000H × TPnOVF bit set (1) count + Captured value) × Count clock cycle

Remark n = 0 to 4
m = 0, 1

Figure 6-39. Register Setting in Pulse Width Measurement Mode (1/2)



(a) TMPn control register 0 (TPnCTL0)

	TPnCE					TPnCKS2	TPnCKS1	TPnCKS0
TPnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

TPnCKS2 to TPnCKS0: Select count clock

TPnCE:
0: Stop counting
1: Enable counting

(b) TMPn control register 1 (TPnCTL1)

		TPnEST	TPnEEE			TPnMD2	TPnMD1	TPnMD0
TPnCTL1	0	0	0	0	0	1	1	0

1, 1, 0:
Pulse width measurement mode

(c) TMPn I/O control register 1 (TPnIOC1)

					TPnIS3	TPnIS2	TPnIS1	TPnIS0
TPnIOC1	0	0	0	0	0/1	0/1	0/1	0/1

TPnIS1, TPnIS0: Select valid edge of TIOPn0 pin input

TPnIS3, TPnIS2: Select valid edge of TIOPn1 pin input

(d) TMPn option register 0 (TPnOPT0)

			TPnCCS1	TPnCCS0				TPnOVF
TPnOPT0	0	0	0	0	0	0	0	0/1

TPnOVF: Overflow flag

Figure 6-39. Register Setting in Pulse Width Measurement Mode (2/2)

(e) TMPn counter read buffer register (TPnCNT)

The value of the 16-bit counter can be read by reading the TPnCNT register.

(f) TMPn capture/compare registers 0 and 1 (TPnCCR0 and TPnCCR1)

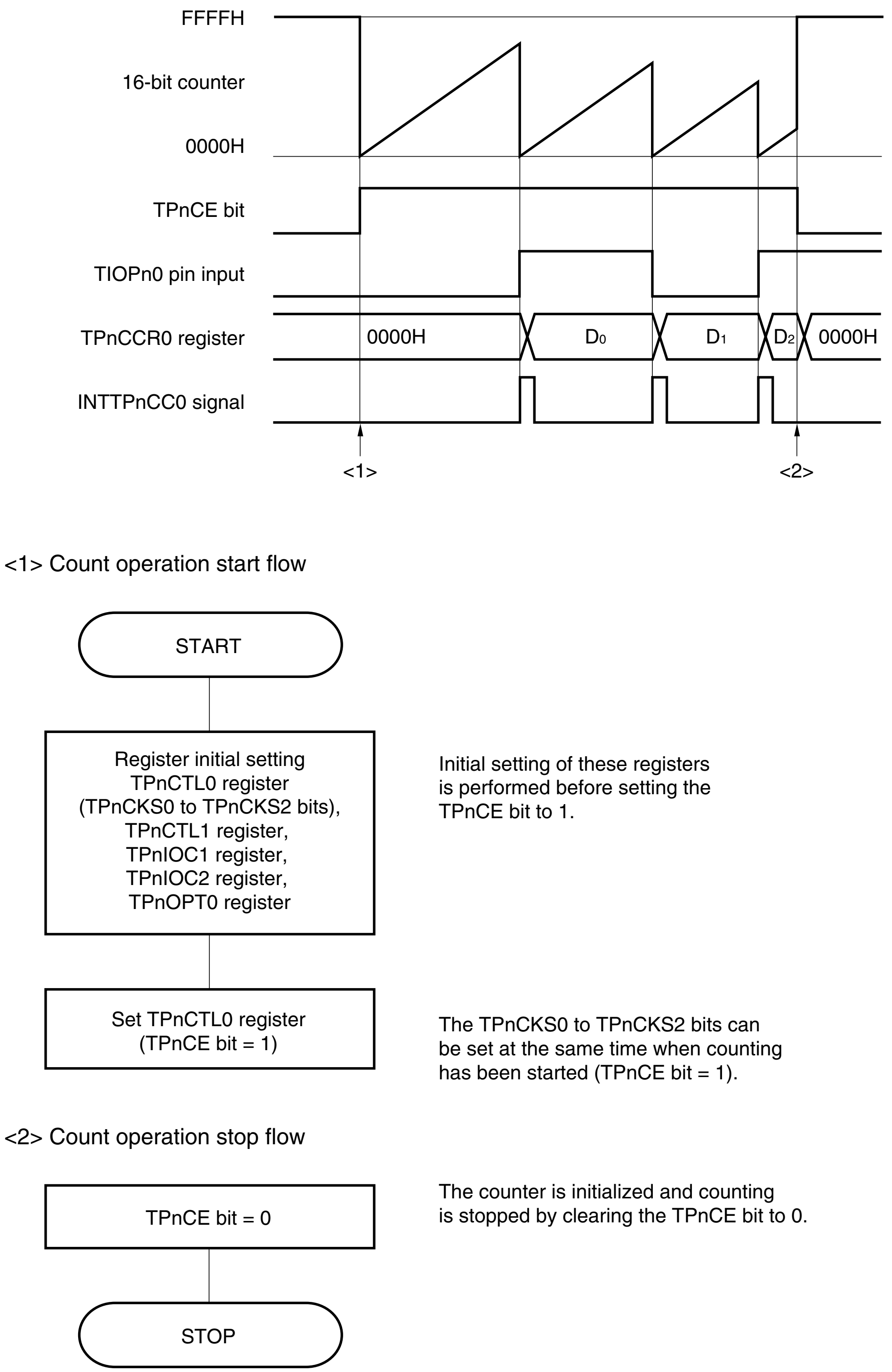
These registers store the count value of the 16-bit counter when the valid edge input to the TIOPnm pin is detected.

Remarks 1. TMPn I/O control register 0 (TPnIOC0) and TMPn I/O control register 2 (TPnIOC2) are not used in the pulse width measurement mode.

2. n = 0 to 4
m = 0, 1

(1) Operation flow in pulse width measurement mode

Figure 6-40. Software Processing Flow in Pulse Width Measurement Mode



Initial setting of these registers is performed before setting the TPnCE bit to 1.

The TPnCKS0 to TPnCKS2 bits can be set at the same time when counting has been started (TPnCE bit = 1).

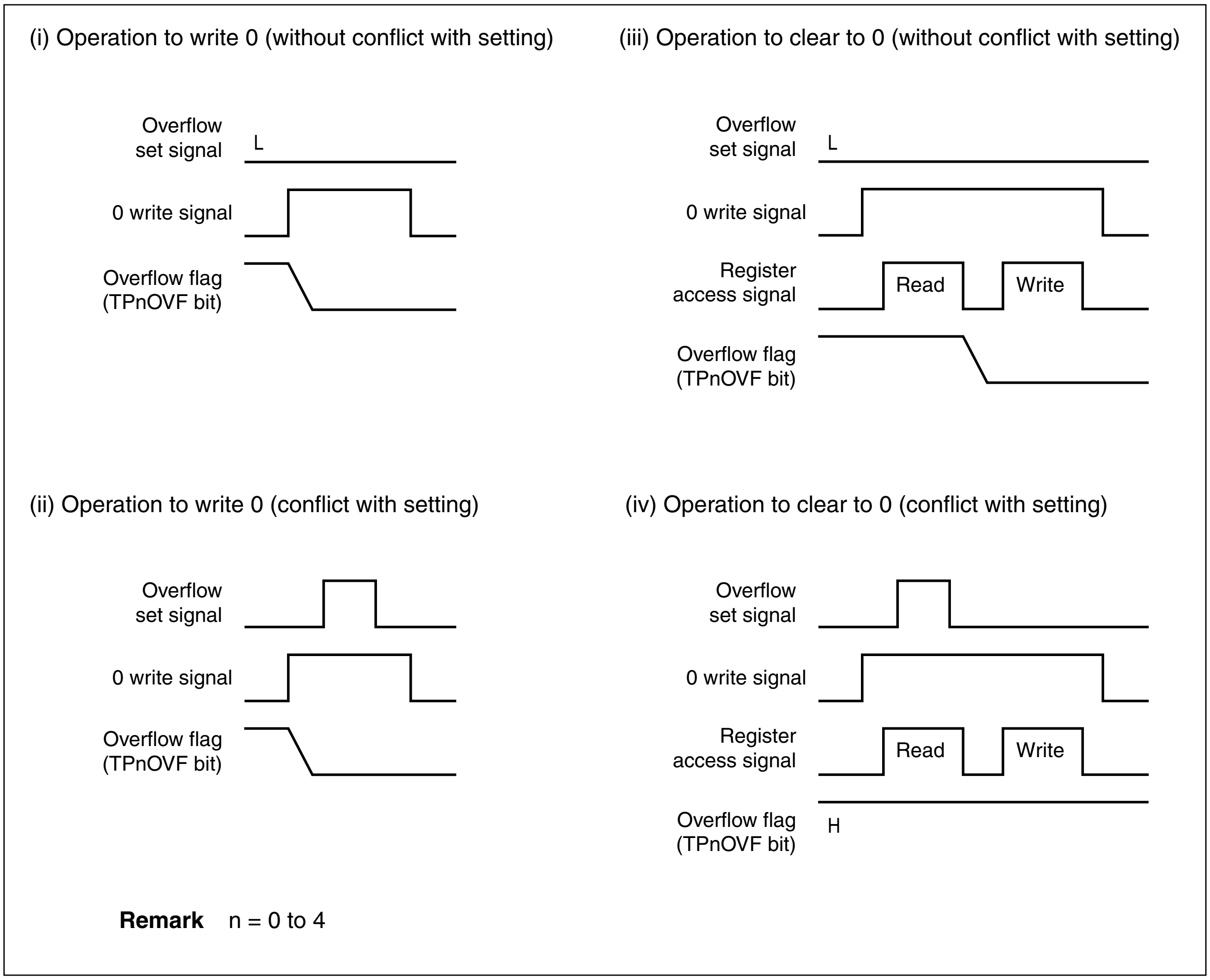
The counter is initialized and counting is stopped by clearing the TPnCE bit to 0.

Remark n = 0 to 4

(2) Operation timing in pulse width measurement mode

(a) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TPnOVF bit to 0 with the CLR instruction and by writing 8-bit data (bit 0 is 0) to the TPnOPT0 register. To accurately detect an overflow, read the TPnOVF bit when it is 1, and then clear the overflow flag by using a bit manipulation instruction.


(i) Operation to write 0 (without conflict with setting)

(iii) Operation to clear to 0 (without conflict with setting)

(ii) Operation to write 0 (conflict with setting)

(iv) Operation to clear to 0 (conflict with setting)

Remark n = 0 to 4

To clear the overflow flag to 0, read the overflow flag to check if it is set to 1, and clear it with the CLR instruction. If 0 is written to the overflow flag without checking if the flag is 1, the set information of overflow may be erased by writing 0 ((ii) in the above chart). Therefore, software may judge that no overflow has occurred even when an overflow actually has occurred.

If execution of the CLR instruction conflicts with occurrence of an overflow when the overflow flag is cleared to 0 with the CLR instruction, the overflow flag remains set even after execution of the clear instruction.

6.7.8 Timer output operations

The following table shows the operations and output levels of the TIOPn0 and TIOPn1 pins.

Table 6-5. Timer Output Control in Each Mode

Operation Mode	TIOPn1 Pin	TIOPn0 Pin
Interval timer mode	Square wave output	
External event count mode	Square wave output	–
External trigger pulse output mode	External trigger pulse output	Square wave output
One-shot pulse output mode	One-shot pulse output	
PWM output mode	PWM output	
Free-running timer mode	Square wave output (only when compare function is used)	
Pulse width measurement mode	–	

Remark n = 0 to 4

Table 6-6. Truth Table of TIOPn0 and TIOPn1 Pins Under Control of Timer Output Control Bits

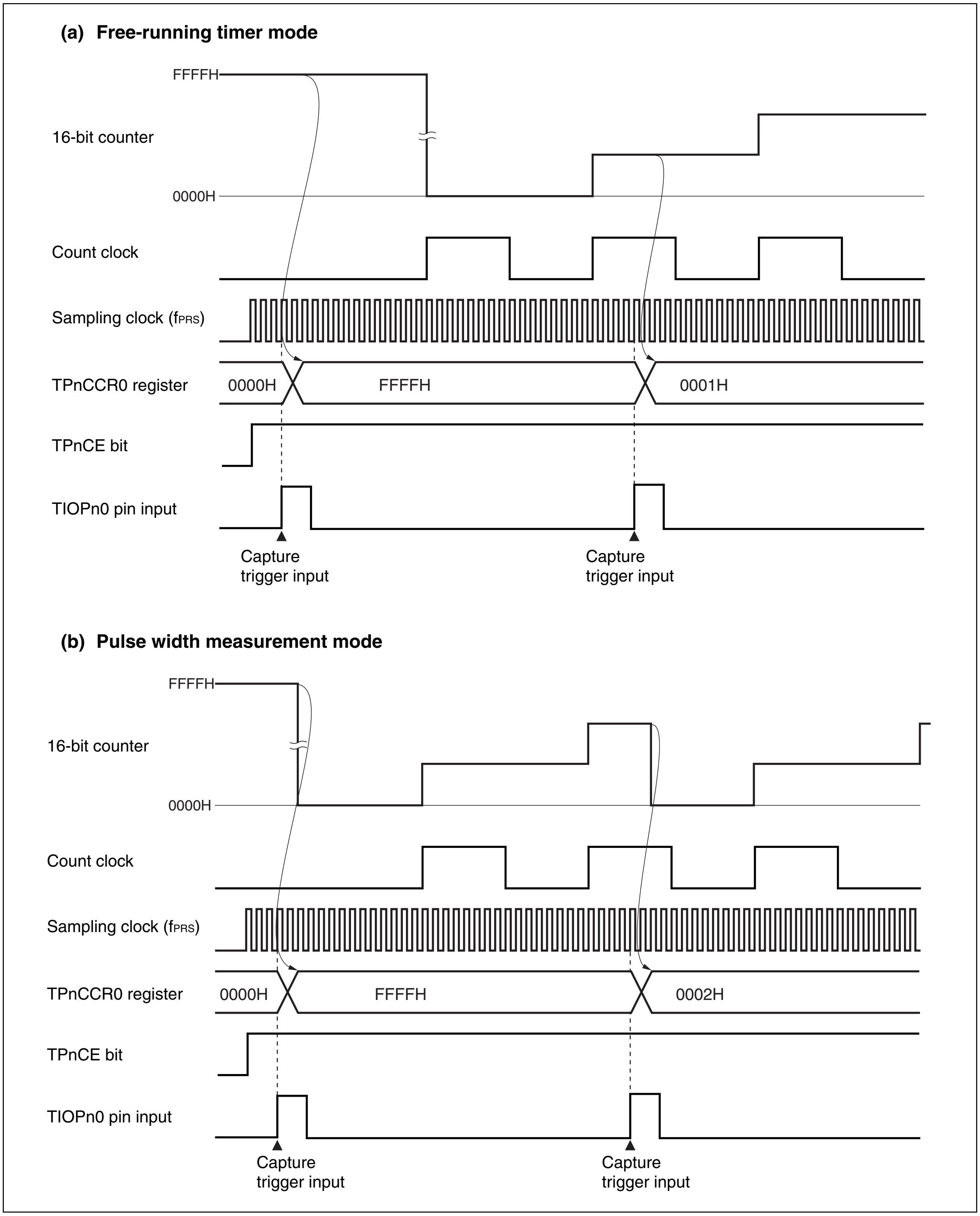
TPnIOC0.TPnOLm Bit	TPnIOC0.TPnOEm Bit	TPnCTL0.TPnCE Bit	Level of TIOPnm Pin
0	0	×	Low-level output
	1	0	Low-level output
		1	Low level immediately before counting, high level after counting is started
1	0	×	High-level output
	1	0	High-level output
		1	High level immediately before counting, low level after counting is started

Remark n = 0 to 4
m = 0, 1

6.8 Cautions

(1) Capture operation

When the capture operation is used and a slow clock is selected as the count clock, FFFFH, not 0000H, may be captured in the TPnCCR0 and TPnCCR1 registers if the capture trigger is input immediately after the TPnCE bit is set to 1.



(a) Free-running timer mode

(b) Pulse width measurement mode

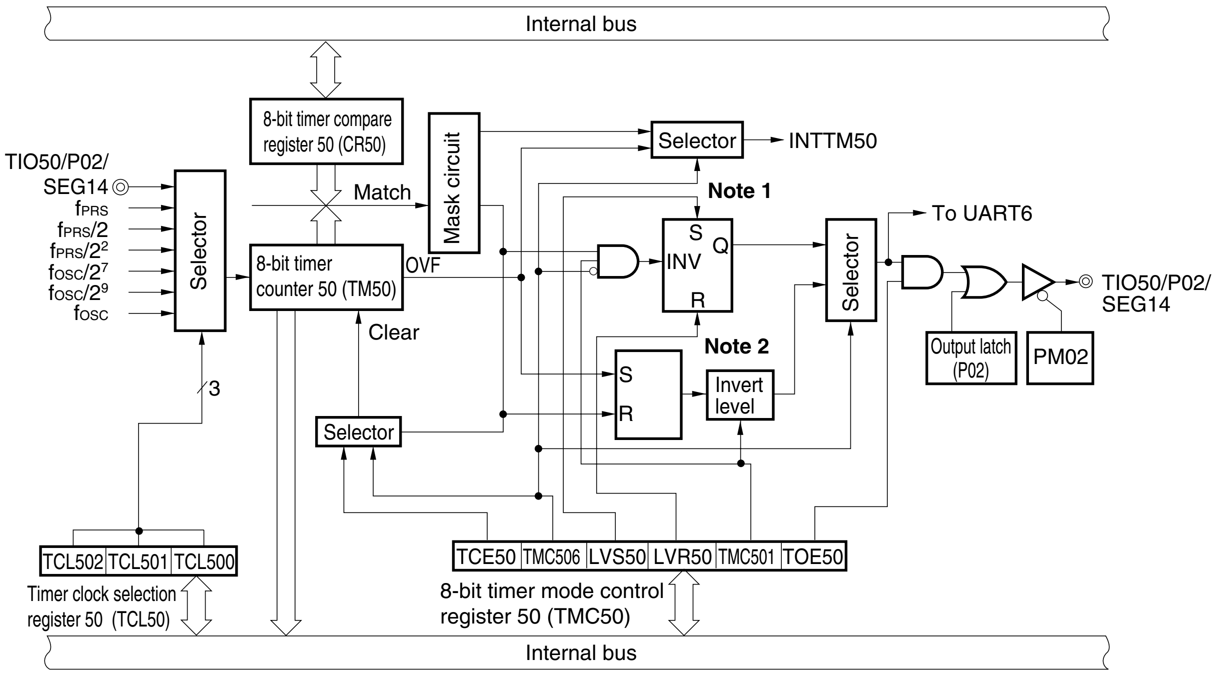
7.1 Functions of 8-Bit Timer/Event Counters 50 and 51

8-bit timer/event counters 50 and 51 have the following functions.

- Interval timer
- External event counter
- Square-wave output
- PWM output

Figures 7-1 and **7-2** show the block diagrams of 8-bit timer/event counters 50 and 51.

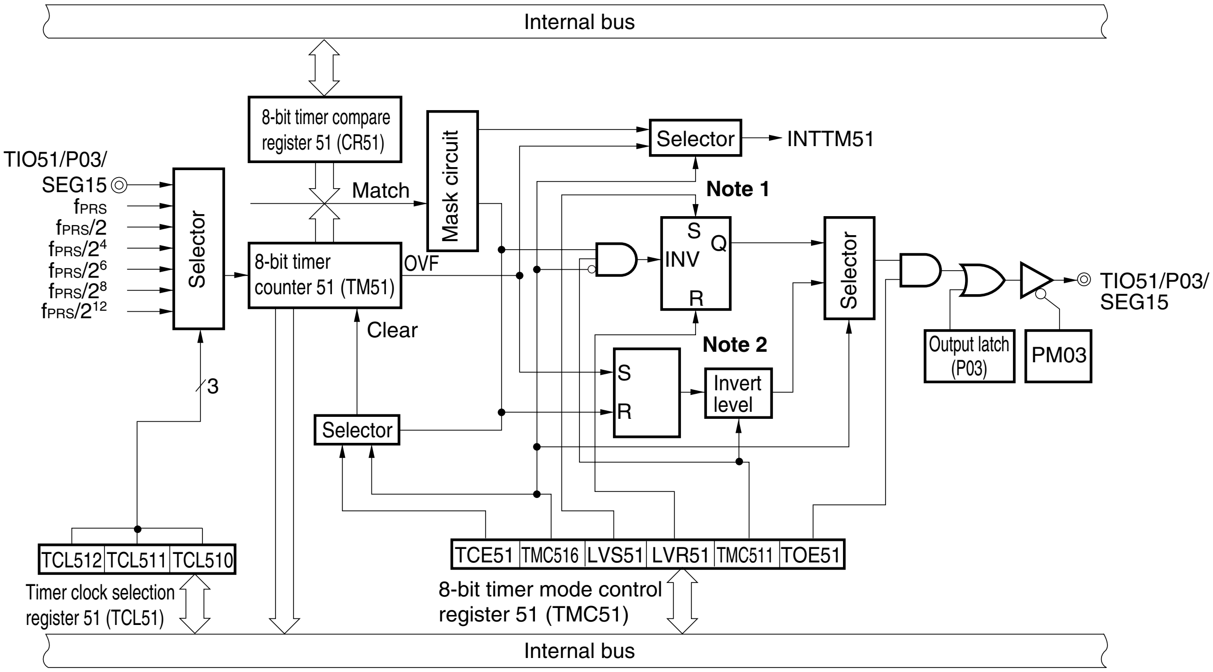
Figure 7-1. Block Diagram of 8-Bit Timer/Event Counter 50



Notes 1. Timer output F/F

2. PWM output F/F

Figure 7-2. Block Diagram of 8-Bit Timer/Event Counter 51



Notes 1. Timer output F/F

2. PWM output F/F

7.2 Configuration of 8-Bit Timer/Event Counters 50 and 51

8-bit timer/event counters 50 and 51 include the following hardware.

Table 7-1. Configuration of 8-Bit Timer/Event Counters 50 and 51

Item	Configuration
Timer register	8-bit timer counter 5n (TM5n)
Register	8-bit timer compare register 5n (CR5n)
Timer input	TIO5n
Timer output	TIO5n
Control registers	Timer clock selection register 5n (TCL5n) 8-bit timer mode control register 5n (TMC5n) Port mode register 0 (PM0) Port register 0 (P0)

(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 7-3. Format of 8-Bit Timer Counter 5n (TM5n)

Address: FF16H (TM50), FF1FH (TM51)   After reset: 00H   R

Symbol	7	6	5	4	3	2	1	0
TM5n (n = 0, 1)								

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.

Remark n = 0, 1

(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 PWM mode, when the TIO5n pin becomes active due to a TM5n overflow and the values of TM5n and CR5n match, the TIO5n pin becomes inactive.

The value of CR5n can be set within 00H to FFH.

Reset signal generation clears CR5n to 00H.

Figure 7-4. Format of 8-Bit Timer Compare Register 5n (CR5n)

Address: FF17H (CR50), FF41H (CR51) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CR5n (n = 0, 1)								

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, 1

7.3 Registers Controlling 8-Bit Timer/Event Counters 50 and 51

The following four registers are used to control 8-bit timer/event counters 50 and 51.

- Timer clock selection register 5n (TCL5n)
- 8-bit timer mode control register 5n (TMC5n)
- Port mode register 0 (P0)
- Port register 0 (P0)

(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 TIO5n pin input.
TCL5n can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears TCL5n to 00H.

Remark n = 0, 1

Figure 7-5. 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	f_{PRS} = 20 MHz
0	0	0	TIO50 pin falling edge[Note 2]				
0	0	1	TIO50 pin rising edge[Note 3]				
0	1	0	f_{PRS}	2 MHz	5 MHz	10 MHz	20 MHz
0	1	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
1	0	0	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
1	0	1	$f_{OSC}/2^7$	Typ. 1.88 kHz			
1	1	0	$f_{OSC}/2^9$	Typ. 0.47 kHz			
1	1	1	f_{OSC}	Typ. 240 kHz			

Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{IN}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 4.0 to 5.5 V: $f_{PRS} \leq 20$ MHz
- V_{DD} = 2.7 to 4.0 V: $f_{PRS} \leq 10$ MHz

2. In the on-board mode, the FLMD0 pin falling edge is selected.

3. In the on-board mode, the FLMD0 pin rising edge is selected.

Cautions 1. When rewriting TCL50 to other data, stop the timer operation beforehand.

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

Remark f_{PRS}: Peripheral hardware clock frequency

Figure 7-6. 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]				
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	TIO51 pin falling edge				
0	0	1	TIO51 pin rising edge				
0	1	0	f_{PRS}	2 MHz	5 MHz	10 MHz	20 MHz
0	1	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
1	0	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz
1	0	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz
1	1	0	$f_{PRS}/2^8$	7.81 kHz	19.53 kHz	39.06 kHz	78.13 kHz
1	1	1	$f_{PRS}/2^{12}$	0.49 kHz	1.22 kHz	2.44 kHz	4.88 kHz

Note If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{IN}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 4.0 to 5.5 V: $f_{PRS} \leq 20$ MHz
- V_{DD} = 2.7 to 4.0 V: $f_{PRS} \leq 10$ MHz

Cautions 1. When rewriting TCL51 to other data, stop the timer operation beforehand.

2. Be sure to set 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
<3> Timer output F/F (flip flop) status setting
<4> Active level selection in timer F/F control or PWM (free-running) mode.
<5> Timer output control

TMC5n can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Remark n = 0, 1

Figure 7-7. 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 TIO50 output: low level)
1	0	Timer output F/F set (1) (default value of TIO50 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 (TM50 output is low level)
1	Output enabled

Note Bits 2 and 3 are write-only.

(Refer to **Cautions** and **Remarks** on the next page.)

Figure 7-8. 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 TIO51 output: low)
1	0	Timer output F/F set (1) (default value of TIO51 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 (TM51 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. Stop operation before rewriting TMC5n6.

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 TIO5n pin regardless of the value of TCE5n.

4. n = 0, 1

(3) Port mode register 0 (PM0)

This register sets port 0 input/output in 1-bit units.

When using the P02/TIO50/SEG14 and P03/TIO51/SEG15 pins for timer output, clear PM02 and PM03 and the output latches of P02 and P03 to 0.

When using the P02/TIO50/SEG14 and P03/TIO51/SEG15 pins for timer input, set PM02 and PM03 to 1. The output latches of P02 and P03 at this time may be 0 or 1.

PM0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 7-9. Format of Port Mode Register 0 (PM0)

Address: FF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00

PM0n	P0n 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 and 51

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.

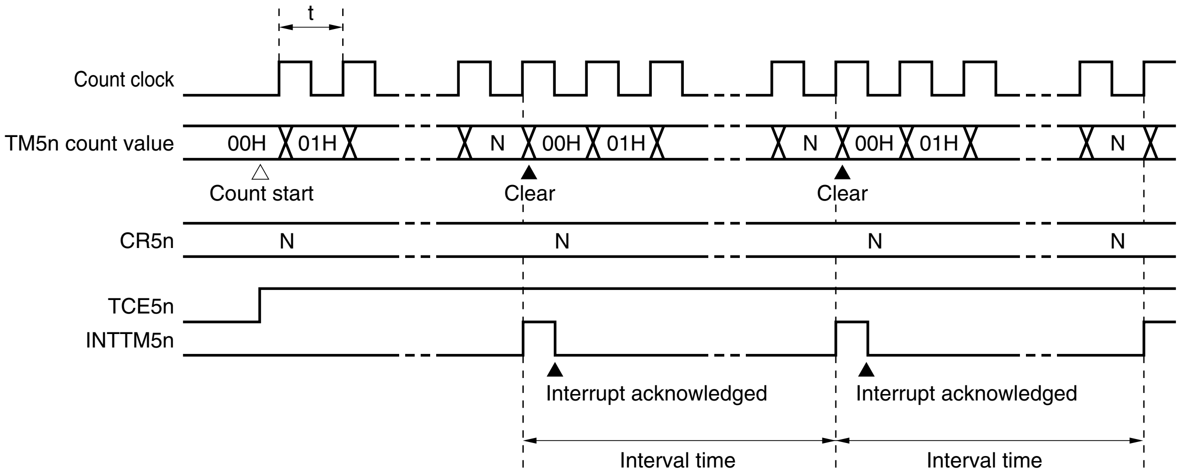
<3> If the values of TM5n and CR5n match, INTTM5n is generated (TM5n is cleared to 00H).

<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.

Figure 7-10. Interval Timer Operation Timing (1/2)

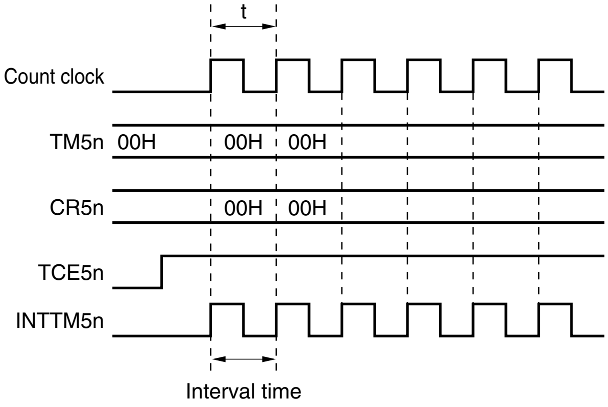
(a) Basic operation



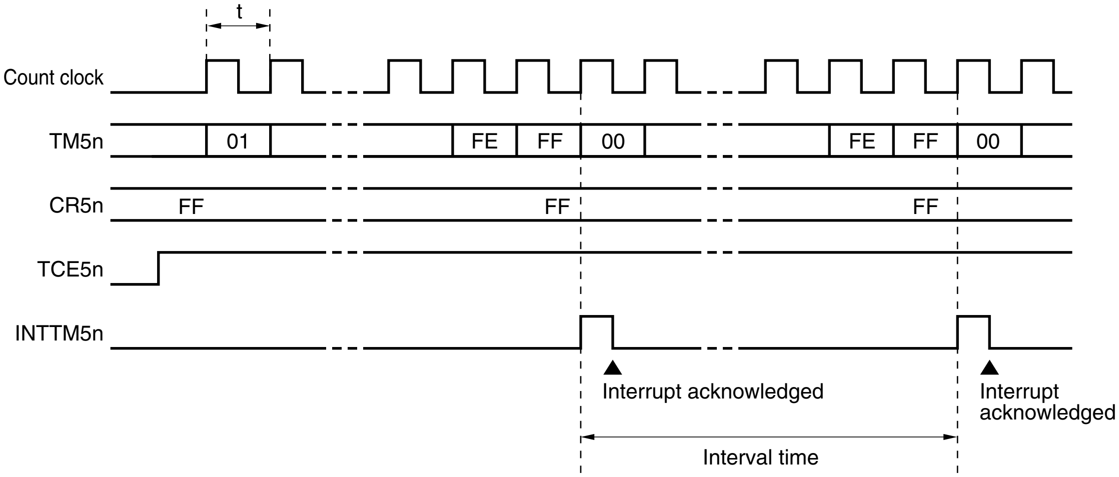
Remark Interval time = (N + 1) × t
N = 00H to FFH
n = 0, 1

Figure 7-10. Interval Timer Operation Timing (2/2)

(b) When CR5n = 00H



(c) When CR5n = FFH



Remark n = 0, 1

7.4.2 Operation as external event counter

The external event counter counts the number of external clock pulses to be input to the TIO5n 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 (PM02 or PM03)Note to 1.
- TCL5n: Select TIO5n pin input edge.
  TIO5n pin falling edge → TCL5n = 00H
  TIO5n 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)

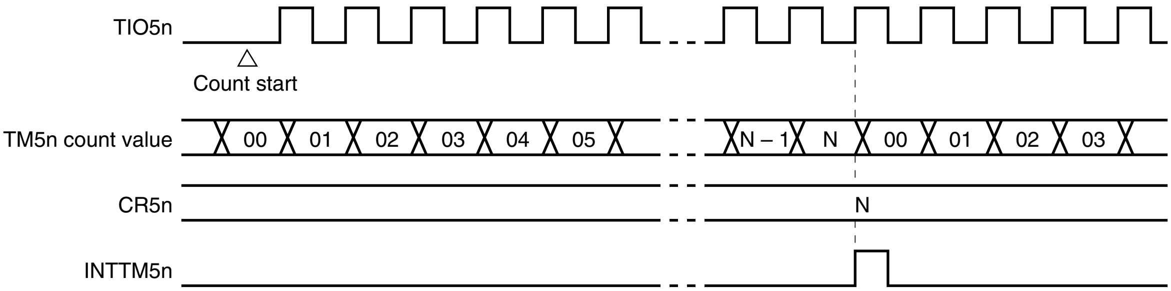
<2> When TCE5n = 1 is set, the number of pulses input from the TIO5n pin is counted.

<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: PM02
8-bit timer/event counter 51: PM03

Figure 7-11. External Event Counter Operation Timing (with Rising Edge Specified)



Remark N = 00H to FFH
n = 0, 1

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 TIO5n 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 (P02 or P03)[Note] and port mode register (PM02 or PM03)[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	High-level output
0	1	Low-level output

Timer output F/F inversion enabled
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 TIO5n.

The frequency is as follows.

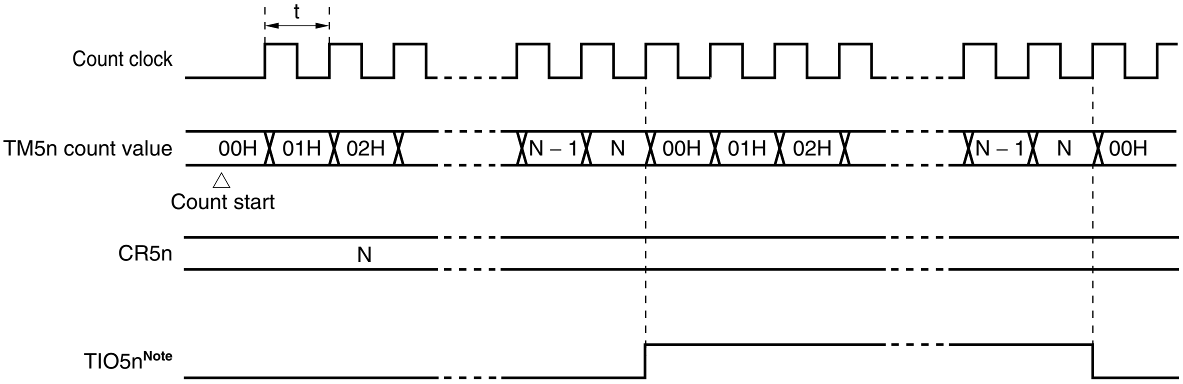
Frequency = 1/2t (N + 1)
(N: 00H to FFH)

Note 8-bit timer/event counter 50: P02, PM02
8-bit timer/event counter 51: P03, PM03

Caution Do not write other values to CR5n during operation.

Remark n = 0, 1

Figure 7-12. Square-Wave Output Operation Timing



Note The initial value of TIO5n 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 TIO5n.

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 (P02 or P03)[Note] and port mode register (PM02 or PM03)[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: P02, PM02
8-bit timer/event counter 51: P03, PM03

PWM output operation

<1> PWM output (output from TIO5n) 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.

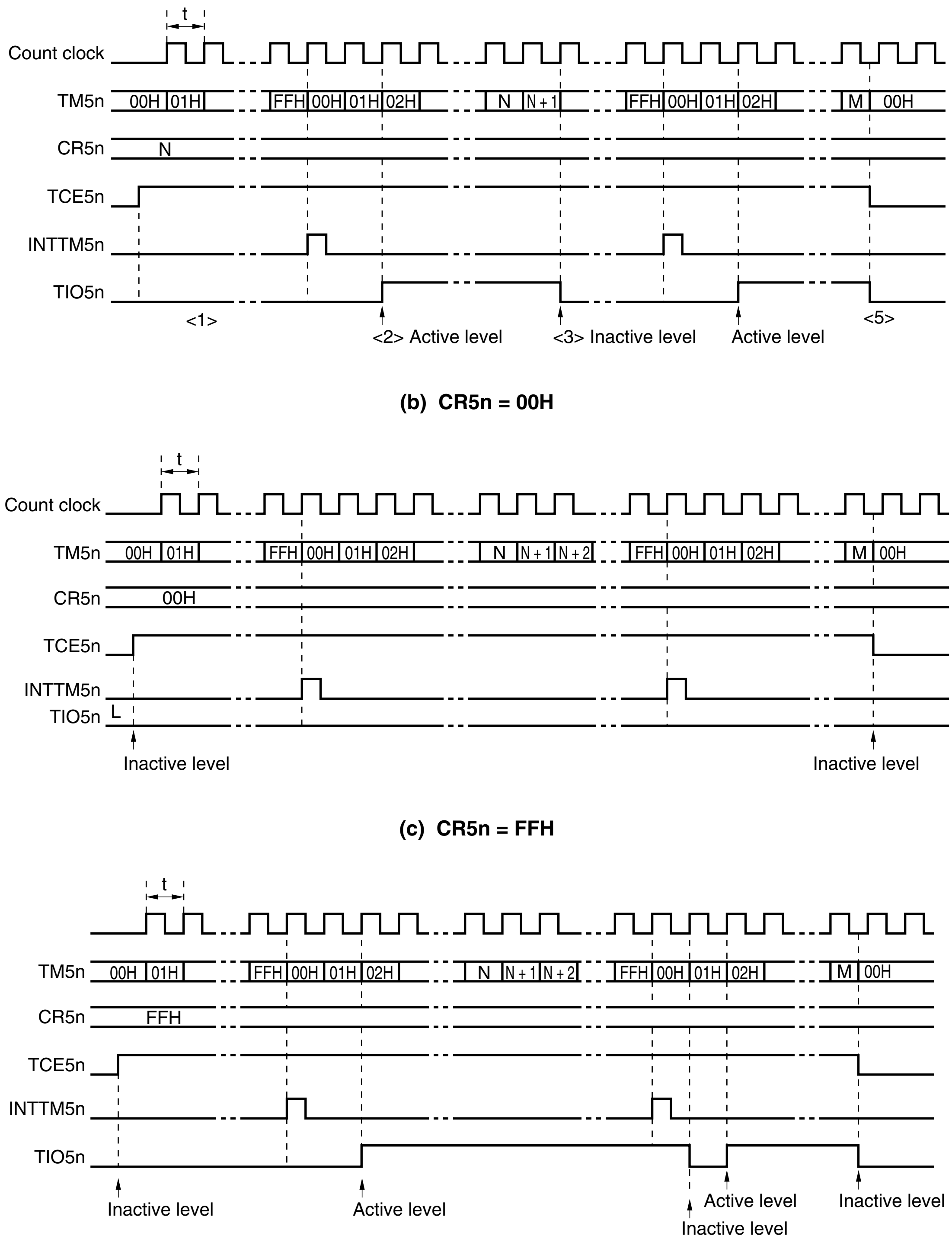
<4> Operations <2> and <3> are repeated until the count operation stops.

<5> When the count operation is stopped with TCE5n = 0, PWM output becomes inactive.
For details of timing, see **Figures 7-13** and **7-14**.
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 7-13. PWM Output Operation Timing


(a) Basic operation (active level = H)

(b) CR5n = 00H

(c) CR5n = FFH

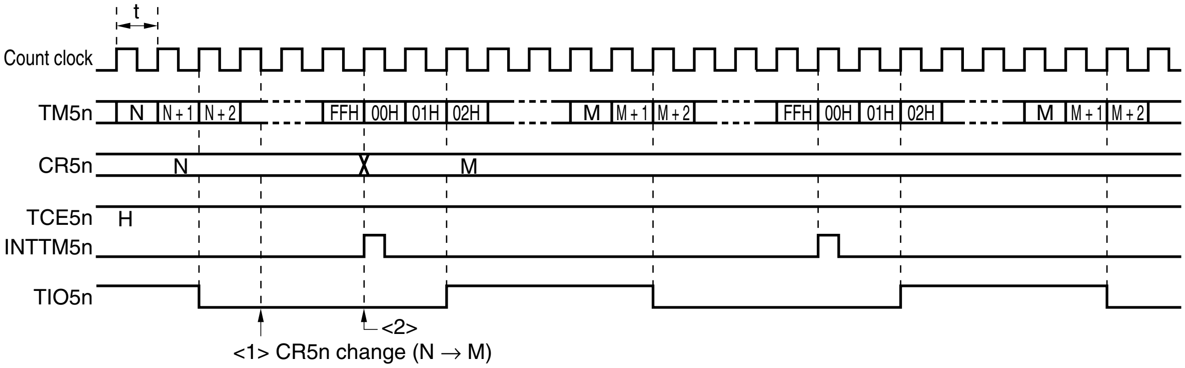
Remarks 1. <1> to <3> and <5> in **Figure 7-13** (a) correspond to <1> to <3> and <5> in PWM output operation in **7. 4. 4 (1) PWM output basic operation**.

2. n = 0, 1

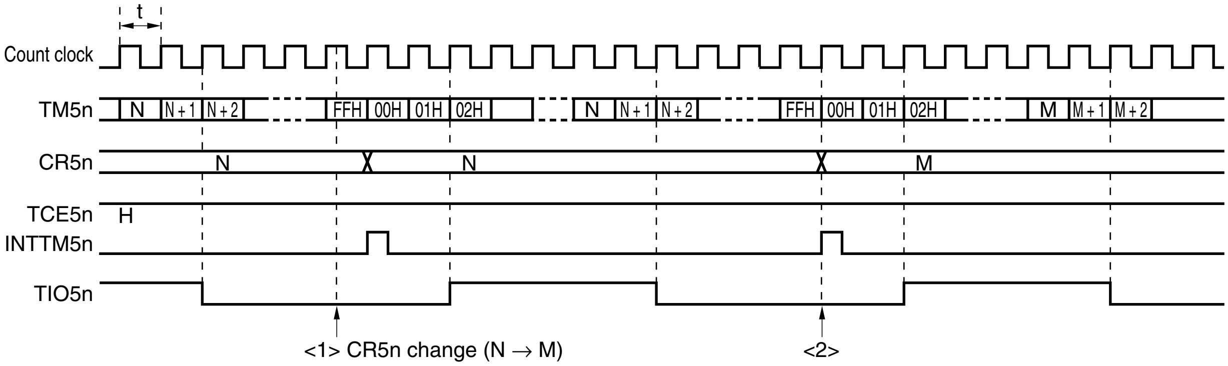
(2) Operation with CR5n changed

Figure 7-14. 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.



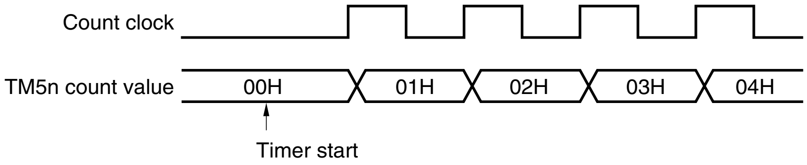
Caution When reading from CR5n between <1> and <2> in Figure 7-14, the value read differs from the actual value (read value: M, actual value of CR5n: N).

7.5 Cautions for 8-Bit Timer/Event Counters 50 and 51

(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 and 51 (TM50, TM51) are started asynchronously to the count clock.

Figure 7-15. 8-Bit Timer Counter 5n Start Timing



Remark n = 0, 1

CHAPTER 8 WATCH TIMER

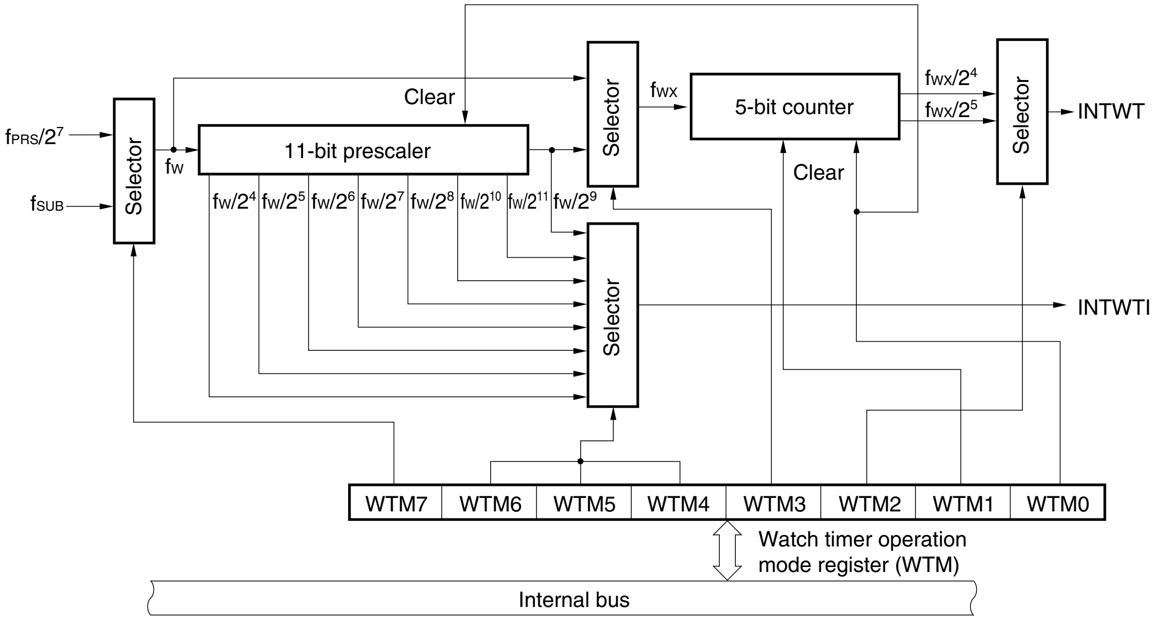
8.1 Functions of Watch Timer

The watch timer has the following functions.

- Watch timer
- Interval timer

The watch timer and the interval timer can be used simultaneously.
Figure 8-1 shows the watch timer block diagram.

Figure 8-1. Block Diagram of Watch Timer



Remark fPRS: Peripheral hardware clock frequency

fSUB: Subsystem clock frequency

fW: Watch timer clock frequency (fPRS/2^7 or fSUB)

fWX: fW or fW/2^9

(1) Watch timer

When the high-speed system clock or subsystem clock is used, interrupt requests (INTWT) are generated at preset intervals.

<R>

Table 8-1. Watch Timer Interrupt Time

Interrupt Time	When Operated at fSUB = 32.768 kHz	When Operated at fPRS = 2 MHz	When Operated at fPRS = 5 MHz	When Operated at fPRS = 10 MHz	When Operated at fPRS = 20 MHz
2^4/fw	488 μs	1.02 ms	410 μs	205 μs	102 μs
2^5/fw	977 μs	2.05 ms	819 μs	410 μs	205 μs
2^{13}/fw	0.25 s	0.52 s	0.210 s	0.105 s	52.5 ms
2^{14}/fw	0.5 s	1.05 s	0.419 s	0.210 s	0.105 s

Remark fPRS: Peripheral hardware clock frequency

fSUB: Subsystem clock frequency

fw: Watch timer clock frequency (fPRS/2^7 or fSUB)

(2) Interval timer

Interrupt requests (INTWTI) are generated at preset time intervals.

<R>

Table 8-2. Interval Timer Interval Time

Interrupt Time	When Operated at fSUB = 32.768 kHz	When Operated at fPRS = 2 MHz	When Operated at fPRS = 5 MHz	When Operated at fPRS = 10 MHz	When Operated at fPRS = 20 MHz
2^4/fw	488 μs	1.02 ms	410 μs	205 μs	102 μs
2^5/fw	977 μs	2.05 ms	820 μs	410 μs	205 μs
2^6/fw	1.95 ms	4.10 ms	1.64 ms	820 μs	410 μs
2^7/fw	3.91 ms	8.20 ms	3.28 ms	1.64 ms	820 μs
2^8/fw	7.81 ms	16.4 ms	6.55 ms	3.28 ms	1.64 ms
2^9/fw	15.6 ms	32.8 ms	13.1 ms	6.55 ms	3.28 ms
2^{10}/fw	31.3 ms	65.5 ms	26.2 ms	13.1 ms	6.55 ms
2^{11}/fw	62.5 ms	131.1 ms	52.4 ms	26.2 ms	13.1 ms

Remark fPRS: Peripheral hardware clock frequency

fSUB: Subsystem clock frequency

fw: Watch timer clock frequency (fPRS/2^7 or fSUB)

8.2 Configuration of Watch Timer

The watch timer includes the following hardware.

Table 8-3. Watch Timer Configuration

Item	Configuration
Counter	5 bits × 1
Prescaler	11 bits × 1
Control register	Watch timer operation mode register (WTM)

8.3 Register Controlling Watch Timer

The watch timer is controlled by the watch timer operation mode register (WTM).

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

  This register sets the watch timer count clock, enables/disables operation, prescaler interval time, and 5-bit counter operation control.

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

  Reset signal generation clears WTM to 00H.

Figure 8-2. Format of Watch Timer Operation Mode Register (WTM)

Address: FF8FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
WTM	WTM7	WTM6	WTM5	WTM4	WTM3	WTM2	WTM1	WTM0

WTM7	Watch timer count clock selection (f_W) [Note]					
		f_{SUB} = 32.768 kHz	f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	$f_{PRS}/2^7$	–	15.625 kHz	39.625 kHz	78.125 kHz	156.25 kHz
1	f_{SUB}	32.768 kHz	–			

<R>

WTM6	WTM5	WTM4	Prescaler interval time selection
0	0	0	$2^4/f_W$
0	0	1	$2^5/f_W$
0	1	0	$2^6/f_W$
0	1	1	$2^7/f_W$
1	0	0	$2^8/f_W$
1	0	1	$2^9/f_W$
1	1	0	$2^{10}/f_W$
1	1	1	$2^{11}/f_W$

WTM3	WTM2	Interrupt time selection
0	0	$2^{14}/f_W$
0	1	$2^{13}/f_W$
1	0	$2^5/f_W$
1	1	$2^4/f_W$

WTM1	5-bit counter operation control
0	Clear after operation stop
1	Start

WTM0	Watch timer operation enable
0	Operation stop (clear both prescaler and 5-bit counter)
1	Operation enable

Note If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{IN}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 4.0 to 5.5 V: $f_{PRS} \leq$ 20 MHz
- V_{DD} = 2.7 to 4.0 V: $f_{PRS} \leq$ 10 MHz

Caution Do not change the count clock and interval time (by setting bits 4 to 7 (WTM4 to WTM7) of WTM) during watch timer operation.

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

2. f_{PRS}: Peripheral hardware clock frequency

3. f_{SUB}: Subsystem clock frequency

8.4 Watch Timer Operations

8.4.1 Watch timer operation

The watch timer generates an interrupt request (INTWT) at a specific time interval by using the peripheral hardware clock or subsystem clock.

When bit 0 (WTM0) and bit 1 (WTM1) of the watch timer operation mode register (WTM) are set to 1, the count operation starts. When these bits are cleared to 0, the 5-bit counter is cleared and the count operation stops.

When the interval timer is simultaneously operated, zero-second start can be achieved only for the watch timer by clearing WTM1 to 0. In this case, however, the 11-bit prescaler is not cleared. Therefore, an error up to $2^9 \times 1/f_W$ seconds occurs in the first overflow (INTWT) after zero-second start.

The interrupt request is generated at the following time intervals.

<R>

Table 8-4. Watch Timer Interrupt Time

WTM3	WTM2	Interrupt Time Selection	When Operated at f_{SUB} = 32.768 kHz (WTM7 = 1)	When Operated at f_{PRS} = 2 MHz (WTM7 = 0)	When Operated at f_{PRS} = 5 MHz (WTM7 = 0)	When Operated at f_{PRS} = 10 MHz (WTM7 = 0)	When Operated at f_{PRS} = 20 MHz (WTM7 = 0)
0	0	$2^{14}/f_W$	0.5 s	1.05 s	0.419 s	0.210 s	0.105 s
0	1	$2^{13}/f_W$	0.25 s	0.52 s	0.210 s	0.105 s	52.5 ms
1	0	$2^5/f_W$	977 μs	2.05 ms	819 μs	410 μs	205 μs
1	1	$2^4/f_W$	488 μs	1.02 ms	410 μs	205 μs	102 μs

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

2. f_{PRS}: Peripheral hardware clock frequency

3. f_{SUB}: Subsystem clock frequency

8.4.2 Interval timer operation

The watch timer operates as interval timer which generates interrupt requests (INTWTI) repeatedly at an interval of the preset count value.

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

When bit 0 (WTM0) of the WTM is set to 1, the count operation starts. When this bit is set to 0, the count operation stops.

<R>

Table 8-5. Interval Timer Interval Time

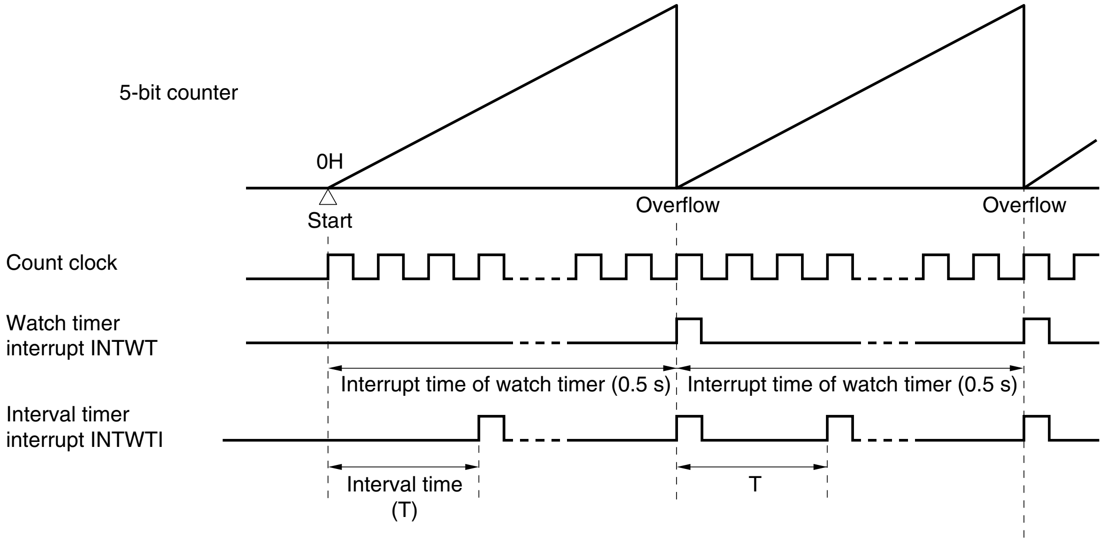
WTM6	WTM5	WTM4	Interval Time	When Operated at f_{SUB} = 32.768 kHz (WTM7 = 1)	When Operated at f_{PRS} = 2 MHz (WTM7 = 0)	When Operated at f_{PRS} = 5 MHz (WTM7 = 0)	When Operated at f_{PRS} = 10 MHz (WTM7 = 0)	When Operated at f_{PRS} = 20 MHz (WTM7 = 0)
0	0	0	$2^4/f_W$	488 μs	1.02 ms	410 μs	205 μs	102 μs
0	0	1	$2^5/f_W$	977 μs	2.05 ms	820 μs	410 μs	205 μs
0	1	0	$2^6/f_W$	1.95 ms	4.10 ms	1.64 ms	820 μs	410 μs
0	1	1	$2^7/f_W$	3.91 ms	8.20 ms	3.28 ms	1.64 ms	820 μs
1	0	0	$2^8/f_W$	7.81 ms	16.4 ms	6.55 ms	3.28 ms	1.64 ms
1	0	1	$2^9/f_W$	15.6 ms	32.8 ms	13.1 ms	6.55 ms	3.28 ms
1	1	0	$2^{10}/f_W$	31.3 ms	65.5 ms	26.2 ms	13.1 ms	6.55 ms
1	1	1	$2^{11}/f_W$	62.5 ms	131.1 ms	52.4 ms	26.2 ms	13.1 ms

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

2. f_{PRS}: Peripheral hardware clock frequency

3. f_{SUB}: Subsystem clock frequency

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



Remark f_W: Watch timer clock frequency

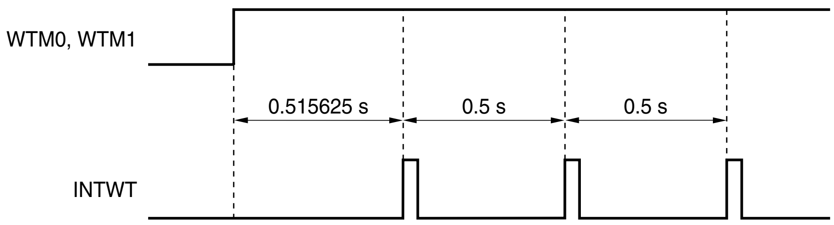
Figures in parentheses are for operation with f_W = 32.768 kHz (WTM7 = 1, WTM3, WTM2 = 0, 0)

8.5 Cautions for Watch Timer

When operation of the watch timer and 5-bit counter is enabled by the watch timer mode control register (WTM) (by setting bits 0 (WTM0) and 1 (WTM1) of WTM to 1), the interval until the first interrupt request (INTWT) is generated after the register is set does not exactly match the specification made with bits 2 and 3 (WTM2, WTM3) of WTM. Subsequently, however, the INTWT signal is generated at the specified intervals.

Figure 8-4. Example of Generation of Watch Timer Interrupt Request (INTWT) (When Interrupt Period = 0.5 s)

It takes 0.515625 seconds for the first INTWT to be generated (2^9 × 1/32768 = 0.015625 s longer). INTWT is then generated every 0.5 seconds.



9.1 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 FFCFH and FFE0H 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 21 RESET FUNCTION**.

9.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 9-1. Configuration of Watchdog Timer

Item	Configuration
Control register	Watchdog timer enable register (WDTE)

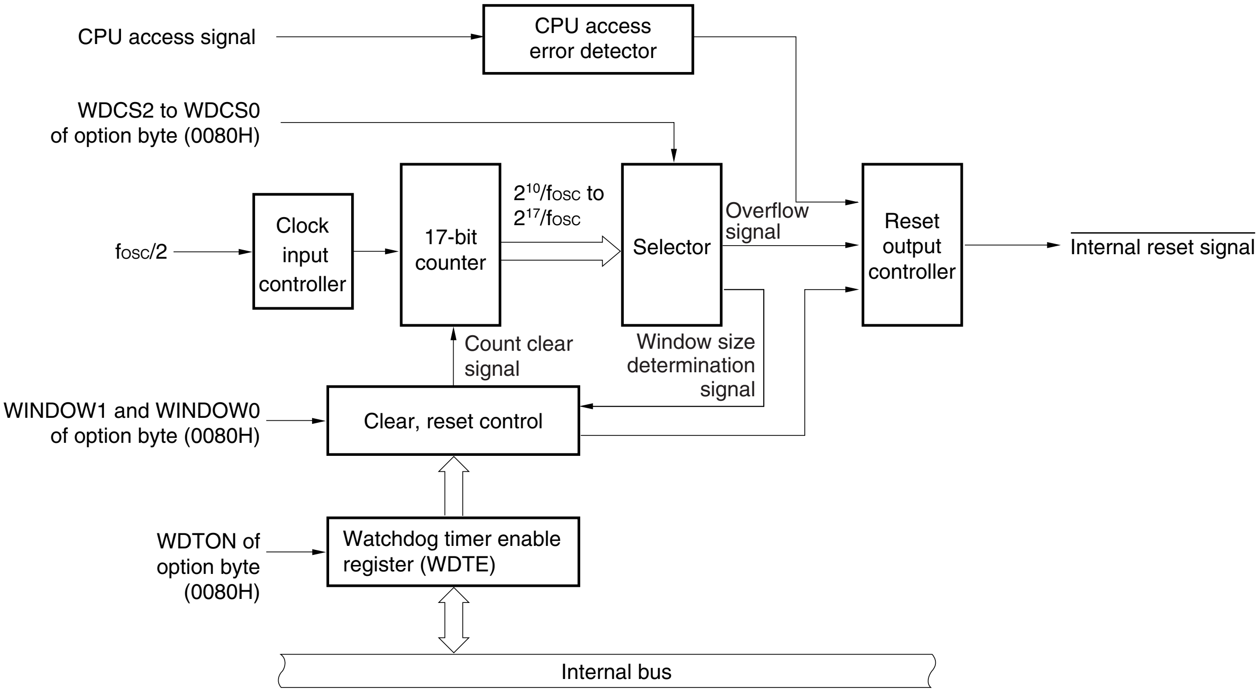
How the counter operation is controlled, overflow time, and window open period are set by the option byte.

Table 9-2. 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 25 OPTION BYTE**.

Figure 9-1. Block Diagram of Watchdog Timer


CPU access signal
CPU access error detector
WDCS2 to WDCS0 of option byte (0080H)
fOSC/2
Clock input controller
17-bit counter
2^{10}/fOSC to 2^{17}/fOSC
Selector
Overflow signal
Reset output controller
Internal reset signal
Count clear signal
Window size determination signal
WINDOW1 and WINDOW0 of option byte (0080H)
Clear, reset control
WDTON of option byte (0080H)
Watchdog timer enable register (WDTE)
Internal bus

9.3 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 9-2. Format of Watchdog Timer Enable Register (WDTE)

Address: FF9BH 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)).**

9.4 Operation of Watchdog Timer

9.4.1 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 25 OPTION BYTE**).

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 **9.4.2 Setting overflow time of watchdog timer** and **CHAPTER 25 OPTION BYTE**).
   - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (0080H) (for details, see **9.4.3 Setting window open period of watchdog timer** and **CHAPTER 25 OPTION BYTE**).

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 period other than the window open period, an internal reset signal is generated.
5. If the overflow time expires without “ACH” written to WDTE, an internal reset signal is generated. An 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 FFCFH and FFE0H 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_{OSC} 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.

9.4.2 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 9-3. Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer
0	0	0	$2^{10}/f_{OSC}$ (4.27 ms)
0	0	1	$2^{11}/f_{OSC}$ (8.53 ms)
0	1	0	$2^{12}/f_{OSC}$ (17.07 ms)
0	1	1	$2^{13}/f_{OSC}$ (34.13 ms)
1	0	0	$2^{14}/f_{OSC}$ (68.27 ms)
1	0	1	$2^{15}/f_{OSC}$ (136.53 ms)
1	1	0	$2^{16}/f_{OSC}$ (273.07 ms)
1	1	1	$2^{17}/f_{OSC}$ (546.13 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_{OSC}: Internal low-speed oscillation clock frequency

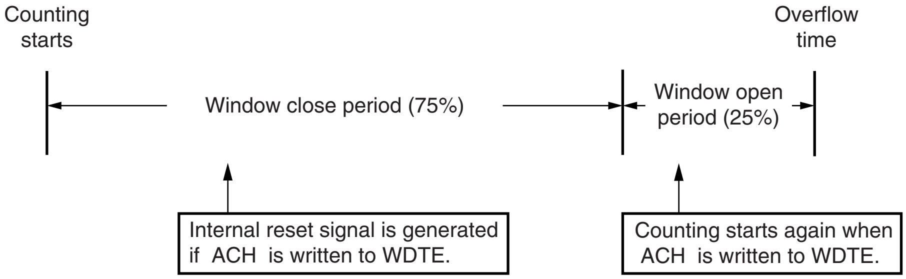
2. ( ): f_{OSC} = 264 kHz (MAX.)

9.4.3 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%


Counting starts
Overflow time
Window close period (75%)
Window open period (25%)
Internal reset signal is generated if ACH is written to WDTE.
Counting starts again when ACH is written to WDTE.

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 9-4. 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.

Remark If the overflow time is set to $2^{10}/f_{OSC}$, the window close time and open time are as follows.

(when 2.7 V ≤ V_{DD} ≤ 5.5 V)

	Setting of Window Open Period			
	25%	50%	75%	100%
Window close time	0 to 3.2 ms	0 to 2.13 ms	0 to 1.07 ms	None
Window open time	3.2 to 4.27 ms	2.13 to 4.27 ms	1.07 to 4.27 ms	0 to 4.27 ms

<When window open period is 25%>

- Overflow time:

  $2^{10}/f_{OSC}$ (MAX.) = 2^{10}/264 kHz (MAX.) = 4.27 ms

- Window close time:

  0 to $2^{10}/f_{OSC}$ (MIN.) × (1 – 0.25) = 0 to 2^{10}/216 kHz (MIN.) × 0.75 = 0 to 3.2 ms

- Window open time:

  $2^{10}/f_{OSC}$ (MIN.) × (1 – 0.25) to $2^{10}/f_{OSC}$ (MAX.) = 2^{10}/216 kHz (MIN.) × 0.75 to 2^{10}/264 kHz (MAX.) = 3.2 to 4.27 ms

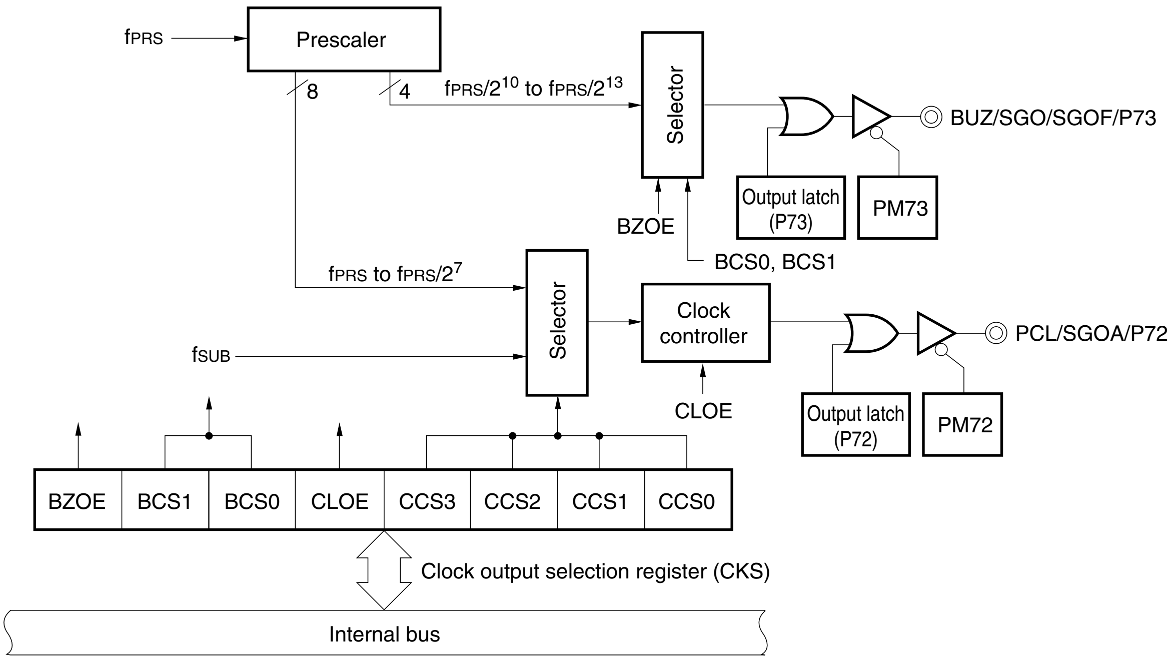
10.1 Functions of Clock Output/Buzzer Output Controller

The clock output controller is intended for carrier output during remote controlled transmission and clock output for supply to peripheral LSIs. 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 10-1 shows the block diagram of clock output/buzzer output controller.

Figure 10-1. Block Diagram of Clock Output/Buzzer Output Controller



10.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 10-1. Clock Output/Buzzer Output Controller Configuration

Item	Configuration
Control registers	Clock output selection register (CKS) Port mode register 7 (PM7) Port register 7 (P7)

10.3 Register 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 7 (PM7)

(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 10-2. 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} = 10 MHz	f_{PRS} = 20 MHz
0	0	$f_{PRS}/2^{10}$	9.77 kHz	19.54 kHz
0	1	$f_{PRS}/2^{11}$	4.88 kHz	9.77 kHz
1	0	$f_{PRS}/2^{12}$	2.44 kHz	4.88 kHz
1	1	$f_{PRS}/2^{13}$	1.22 kHz	2.44 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} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	0	f_{PRS}	–	10 MHz	Setting prohibited Note 2
0	0	0	1	$f_{PRS}/2$		5 MHz	10 MHz
0	0	1	0	$f_{PRS}/2^2$		2.5 MHz	5 MHz
0	0	1	1	$f_{PRS}/2^3$		1.25 MHz	2.5 MHz
0	1	0	0	$f_{PRS}/2^4$		625 kHz	1.25 MHz
0	1	0	1	$f_{PRS}/2^5$		312.5 kHz	625 kHz
0	1	1	0	$f_{PRS}/2^6$		156.25 kHz	312.5 kHz
0	1	1	1	$f_{PRS}/2^7$		78.125 kHz	156.25 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_{IN}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 4.0 to 5.5 V: $f_{PRS} \leq 20$ MHz
- V_{DD} = 2.7 to 4.0 V: $f_{PRS} \leq 10$ MHz

2. The PCL output clock prohibits settings if they exceed 10 MHz.

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 7 (PM7)

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

When using the P72/SGOA/PCL pin for clock output and the P73/SGO/SGOF/BUZ pin for buzzer output, set PM72, PM73 and the output latch of P72, P73 to 0.

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

Reset signal generation sets PM7 to FFH.

Figure 10-3. Format of Port Mode Register 7 (PM7)

Address: FF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	1	PM76	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	P7n pin I/O mode selection (n = 0 to 6)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

10.4 Clock Output/Buzzer Output Controller Operations

10.4.1 Clock output operation

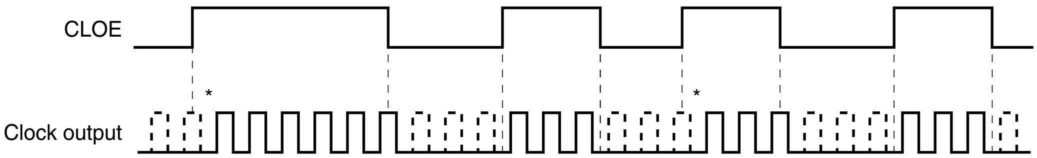
The clock pulse is output as the following procedure.

<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 10-4**, 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 10-4. Remote Control Output Application Example



10.4.2 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 11 A/D CONVERTER

11.1 Function of A/D Converter

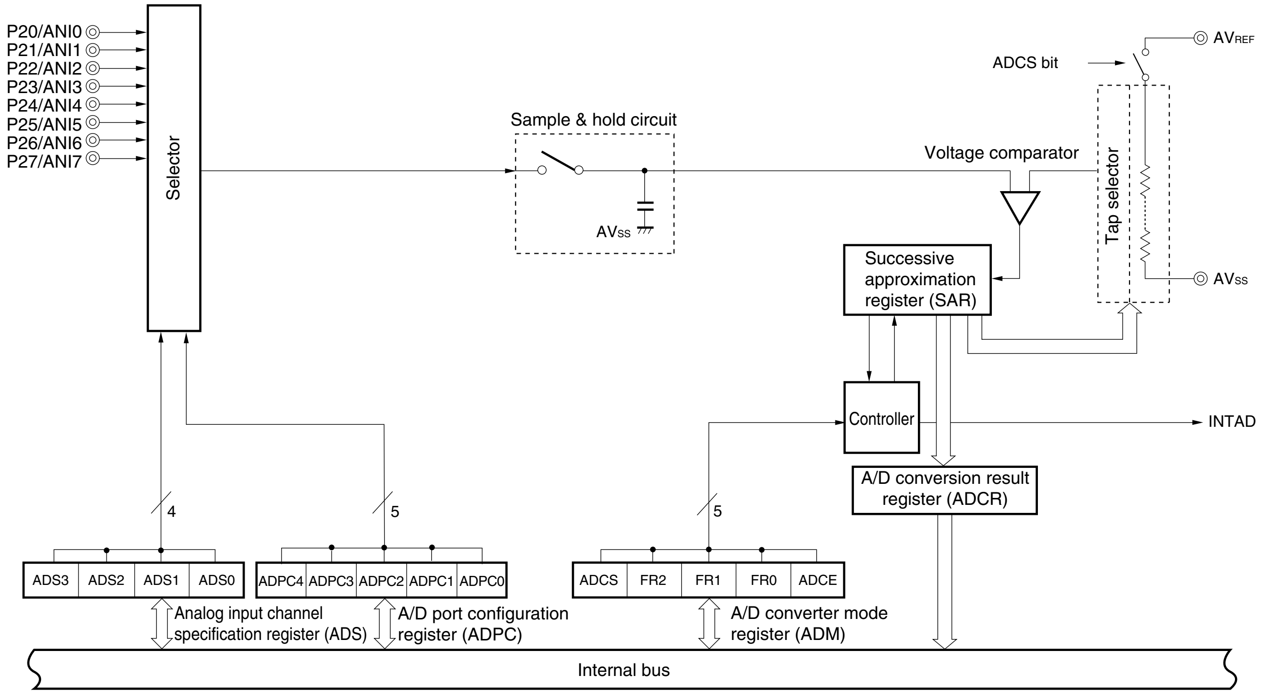
The A/D converter converts an analog input signal into a digital value, and consists of up to eight channels (ANI0 to ANI7) with a resolution of 10 bits.

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 channel selected from analog inputs ANI0 to ANI7. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

Figure 11-1. Block Diagram of A/D Converter



11.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI0 to ANI7 pins

These are the analog input pins of the 16-channel 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.

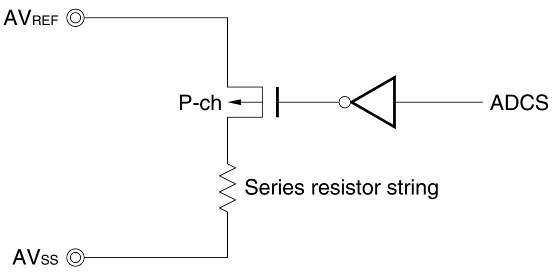
(2) Sample & hold circuit

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 11-2. 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 32 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. Make this pin the same potential as the V_{DD} pin when port 2 is used as a digital port.

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 (ADPC)

This register switches the P20/ANI0 to P27/ANI7 pins to analog input of 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 P20/ANI0 to P27/ANI7 pins to input or output.

11.3 Registers Used in A/D Converter

The A/D converter uses the following seven registers.

- A/D converter mode register (ADM)
- A/D port configuration register (ADPC)
- Analog input channel specification register (ADS)
- 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 11-3. Format of A/D Converter Mode Register (ADM)

Address: FF2AH After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM	ADCS	0	FR2[Note 1]	FR1[Note 1]	FR0[Note 1]	0	0	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 FR2 to FR0, and A/D conversion, see **Table 11-2 A/D Conversion Time Selection**.

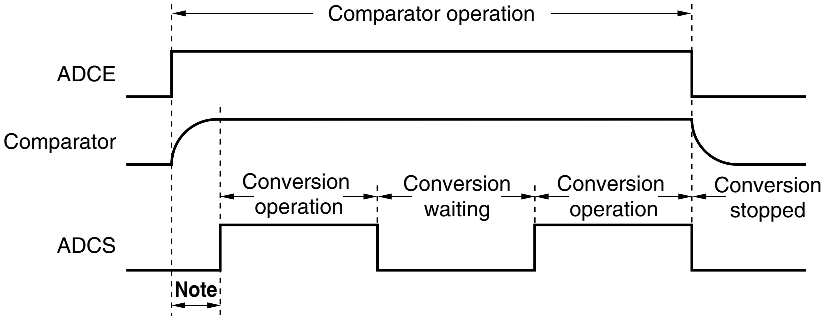
2. The operation of the comparator is controlled by ADCS and ADCE, and it takes 1 μs from operation start to operation stabilization. Therefore, when ADCS is set to 1 after 1 μs or more has elapsed from the time ADCE is set to 1, the conversion result at that time has priority over the first conversion result. Otherwise, ignore data of the first conversion.

Table 11-1. 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 because it is not guaranteed range.

Figure 11-4. Timing Chart When Comparator Is Used



Note To stabilize the internal circuit, the time from the rising of the ADCE bit to the rising of the ADCS bit must be 1 μs or longer.

Cautions 1. A/D conversion must be stopped before rewriting bits FR0 to FR2 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 32 CAUTIONS FOR WAIT.

Table 11-2. A/D Conversion Time Selection

A/D Converter Mode Register (ADM)			Conversion Time Selection				Conversion Clock (f_{AD})
FR2	FR1	FR0		f_{PRS} = 2 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz	
0	0	0	264/f_{PRS}	Setting prohibited	26.4 μs	13.2 μs	f_{PRS}/12
0	0	1	176/f_{PRS}		17.6 μs	8.8 μs[Note]	f_{PRS}/8
0	1	0	132/f_{PRS}		13.2 μs	6.6 μs[Note]	f_{PRS}/6
0	1	1	88/f_{PRS}		8.8 μs[Note]	Setting prohibited	f_{PRS}/4
1	0	0	66/f_{PRS}	33.0 μs	6.6 μs[Note]		f_{PRS}/3
1	0	1	44/f_{PRS}	22.0 μs	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.

Cautions 1. Set the conversion times with the following conditions.

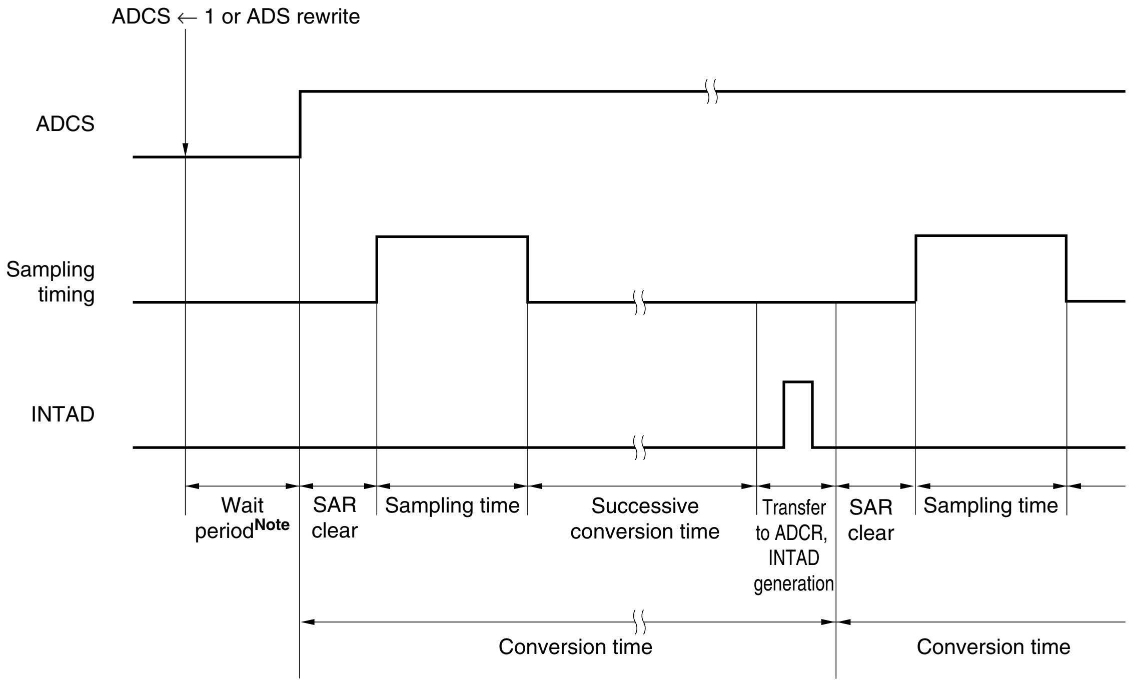
- **4.0 V ≤ AV_{REF} ≤ 5.5 V: f_{AD} = 0.6 to 3.6 MHz**
- **2.7 V ≤ AV_{REF} < 4.0 V: f_{AD} = 0.6 to 1.8 MHz**

2. When rewriting FR2 to FR0 to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

3. 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 11-5. A/D Converter Sampling and A/D Conversion Timing



Note For details of wait period, see **CHAPTER 32 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, and is stored in ADCR in order starting from bit 7 of FF19H. FF19H indicates the higher 8 bits of the conversion result, and FF18H indicates the lower 2 bits of the conversion result.

ADCR can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 11-6. Format of 10-Bit A/D Conversion Result Register (ADCR)

Address: FF18H, FF19H After reset: 0000H R

Symbol	FF19H								FF18H							
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 (ADPC), the contents of ADCR may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC. 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 32 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 11-7. Format of 8-Bit A/D Conversion Result Register (ADCRH)

Address: FF19H 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 (ADPC), the contents of ADCRH may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC. 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 32 CAUTIONS FOR WAIT.

(4) Analog input channel specification register (ADS)

This register specifies the input port 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 11-8. Format of Analog Input Channel Specification Register (ADS)

Address: FF2BH 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. Because ADS and ADPC do not control input and output, set the channel used for A/D conversion in the input mode by using port mode register 2 (PM2). If the channel is set in the output mode, selection of ADPC is disabled.

3. Do not set a pin to be used as a digital input pin with ADPC with ADS.

4. 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 32 CAUTIONS FOR WAIT.

(5) A/D port configuration register (ADPC)

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

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

Reset signal generation clears this register to 00H.

Figure 11-9. Format of A/D Port Configuration Register (ADPC)

Address: FF2DH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC	0	0	0	0	ADPC3	ADPC2	ADPC1	ADPC0

ADPC3	ADPC2	ADPC1	ADPC0	Analog input (A)/ digital I/O (D) switching							
				P27/ANI7	P26/ANI6	P25/ANI5	P24/ANI4	P23/ANI3	P22/ANI2	P21/ANI1	P20/ANI0
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							

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

2. If data is written to ADPC, a wait cycle is generated. Do not write data to ADPC when the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped. For details, see CHAPTER 32 CAUTIONS FOR WAIT.

(6) Port mode register 2 (PM2)

When using the P20/ANI0 to P27/ANI7 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 11-10. 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 to P27/ANI7 pins are as shown below depending on the settings of ADPC, ADS, PM2.

Table 11-3. Setting Functions of P20/ANI0 to P27/ANI7 Pins

ADPC	PM2	ADS	P20/ANI0 to P27/ANI7 Pins
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output

11.4 A/D Converter Operations

11.4.1 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 (ADPC) and set to input mode by using port mode register 2 (PM2).

<3> Set A/D conversion time by using bits 5 to 3 (FR2 to FR0) 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.

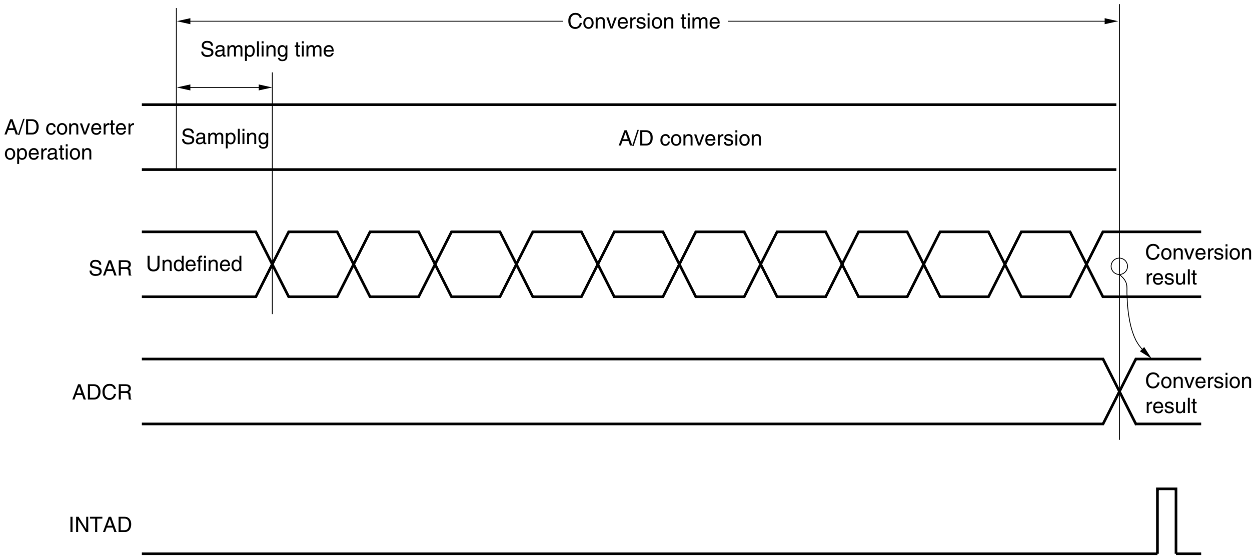
<13> Repeat steps <6> to <12>, until ADCS is cleared to 0.
To stop the A/D converter, clear ADCS to 0.
To restart A/D conversion from the status of ADCE = 1, start from <5>. To start A/D conversion again when ADCE = 0, set ADCE to 1, wait for 1 μs or longer, and start <5>. To change a channel of A/D 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 11-11. 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.

11.4.2 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.

$$\text{SAR} = \text{INT}\left(\frac{V_{\text{AIN}}}{AV_{\text{REF}}} \times 1024 + 0.5\right)$$

$$\text{ADCR} = \text{SAR} \times 64$$

or

$$(\text{ADCR} - 0.5) \times \frac{AV_{\text{REF}}}{1024} \le V_{\text{AIN}} < (\text{ADCR} + 0.5) \times \frac{AV_{\text{REF}}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses

V_{AIN}: Analog input voltage

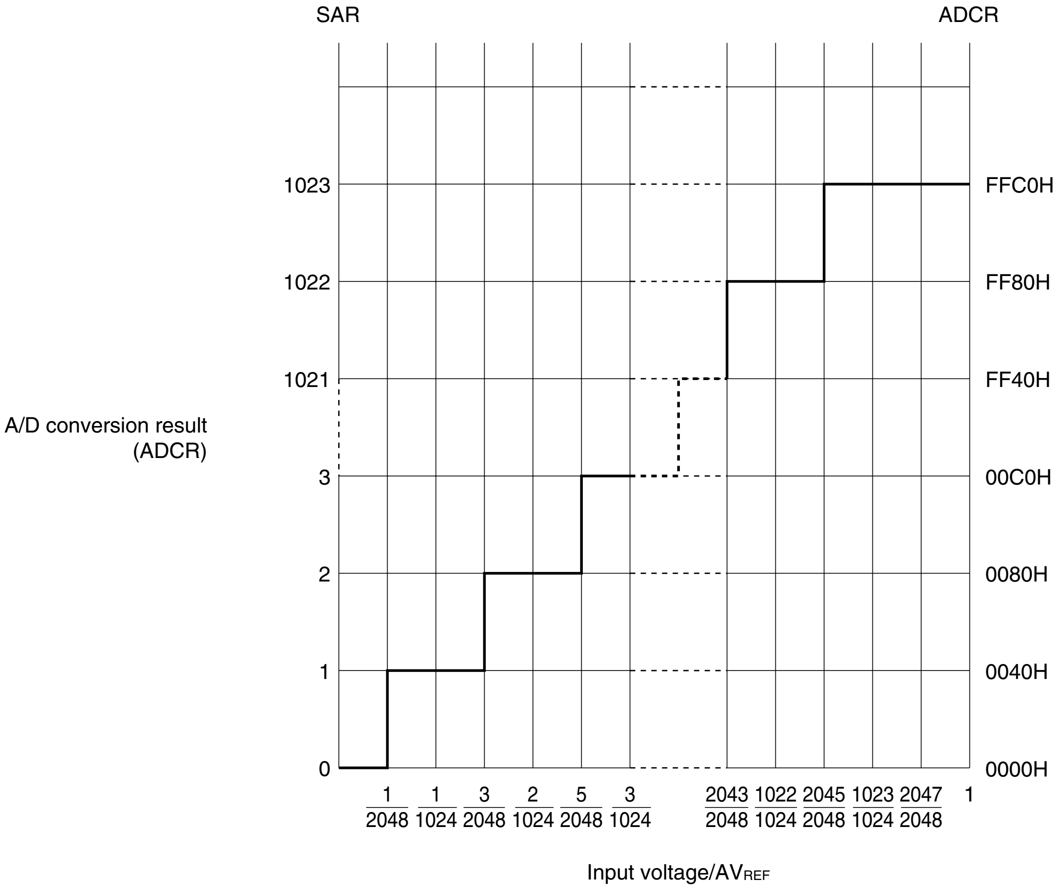
AV_{REF}: AV_{REF} pin voltage

ADCR: A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 11-12 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 11-12. Relationship Between Analog Input Voltage and A/D Conversion Result



11.4.3 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

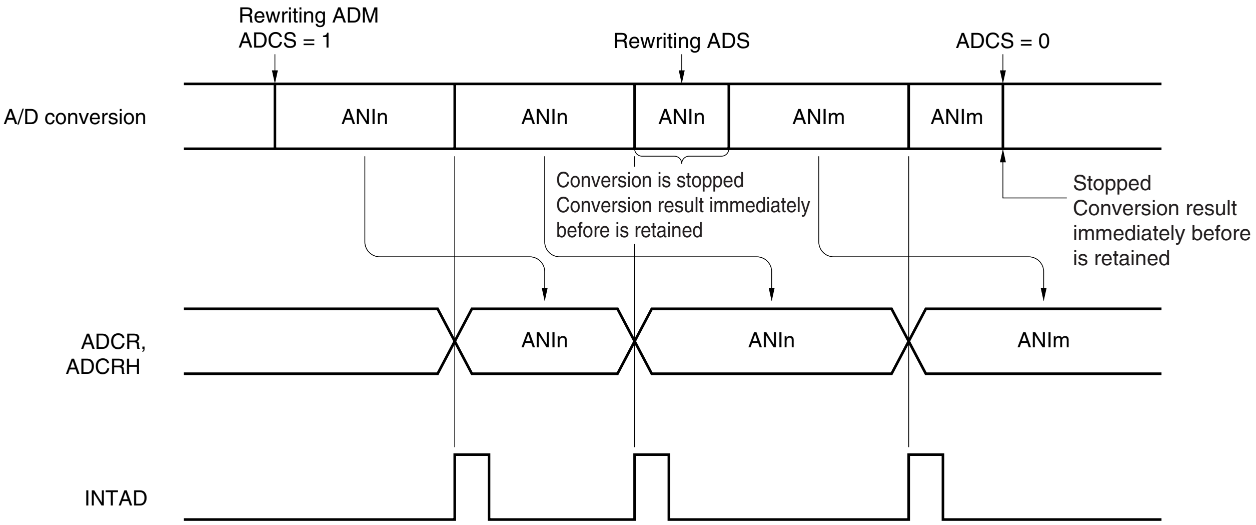
By setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 1, the A/D conversion operation of the voltage, which is applied to the analog input pin specified by the analog input channel specification register (ADS), is started.

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 11-13. 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 4 to 0 (ADPC4 to ADPC0) of the A/D port configuration register (ADPC) and bits 7 to 0 (PM27 to PM20) of port mode register 2 (PM2).

<3> Select conversion time by using bits 5 to 3 (FR2 to FR0) of ADM.

<4> Select a channel to be used by using bits 3 to 0 (ADS3 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 3 to 0 (ADS3 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 5 to 3 (FR2 to FR0) of ADM. The period from <8> to <9> is the conversion time set using FR2 to FR0.

11.5 How to Read A/D Converter Characteristics Table

Here, special terms unique to the 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 11-14. Overall Error

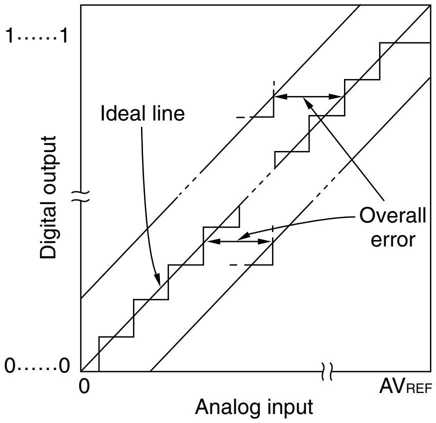
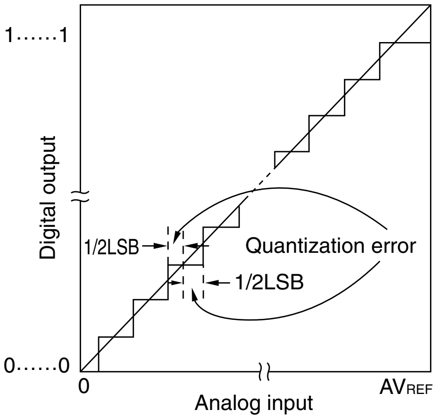


Figure 11-15. Quantization Error



(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 11-16. Zero-Scale Error

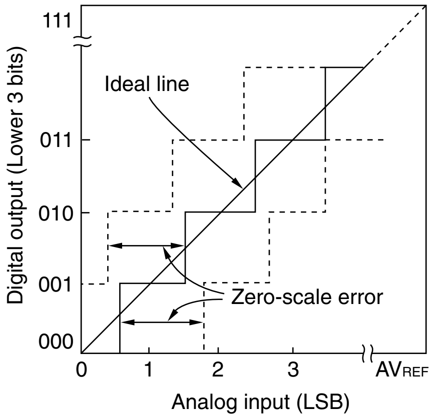


Figure 11-17. Full-Scale Error

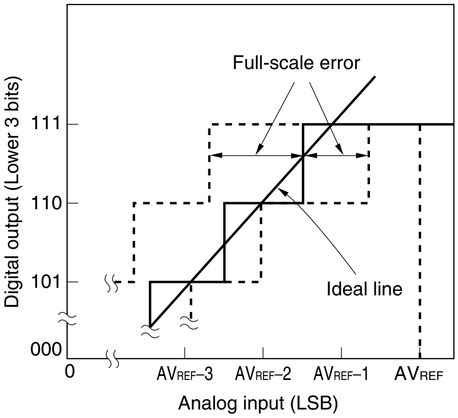


Figure 11-18. Integral Linearity Error

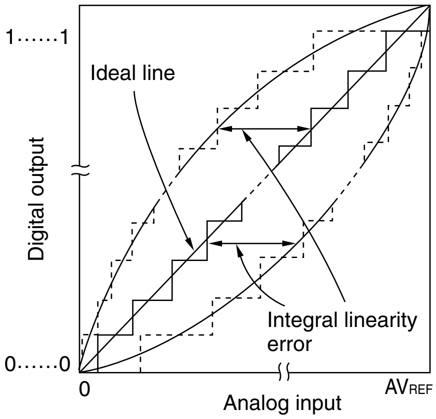
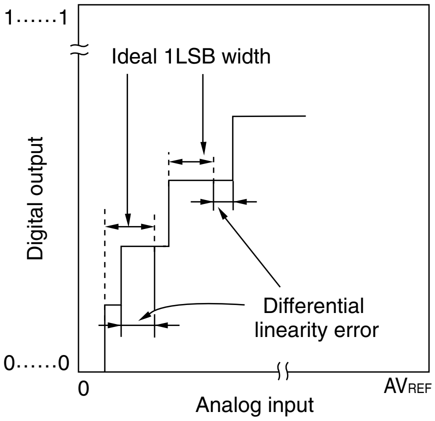


Figure 11-19. Differential Linearity Error



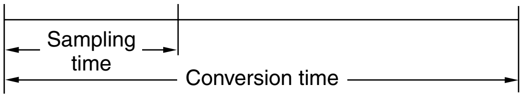
(8) Conversion time

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.



11.6 Cautions for 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 6 (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 (ADPC) write upon the end of conversion

ADM, ADS, or ADPC 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.

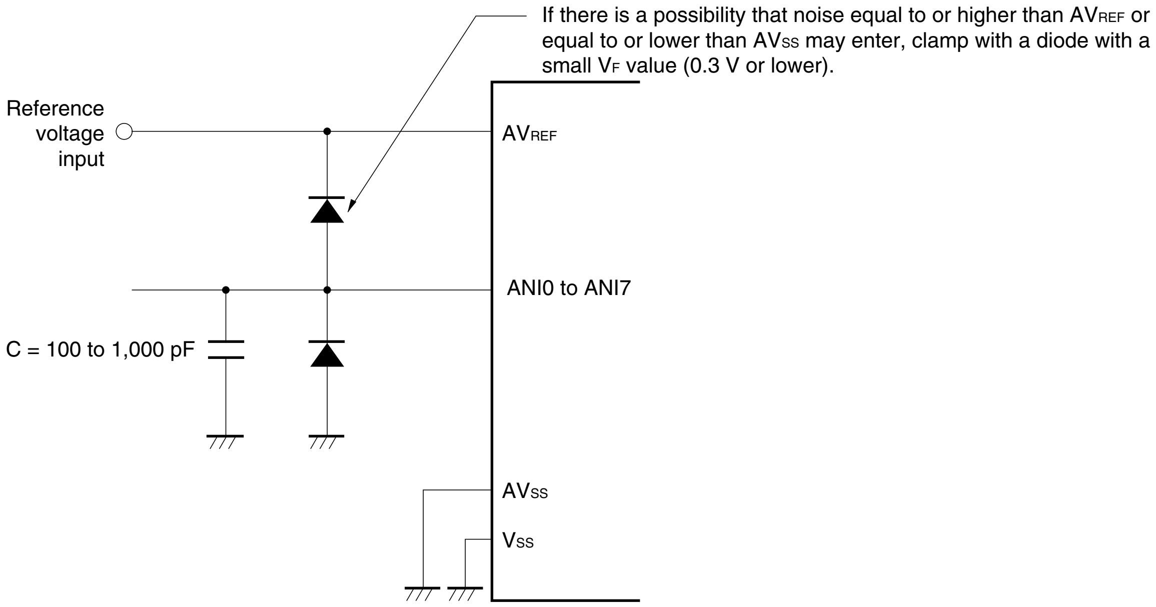
<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

<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 11-20** 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 11-20. Analog Input Pin Connection


If there is a possibility that noise equal to or higher than AVREF or equal to or lower than AVSS may enter, clamp with a diode with a small VF value (0.3 V or lower).
Reference voltage input
AVREF
ANI0 to ANI7
C = 100 to 1,000 pF
AVSS
VSS

(5) P20/ANI0 to P27/ANI7

<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 select pins used as P20 to P27 starting with the P20/ANI0 that is the furthest from AV_{REF}.

<2> If a digital pulse is applied 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 apply a pulse 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 11-20. Analog Input Pin Connection**).

(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.

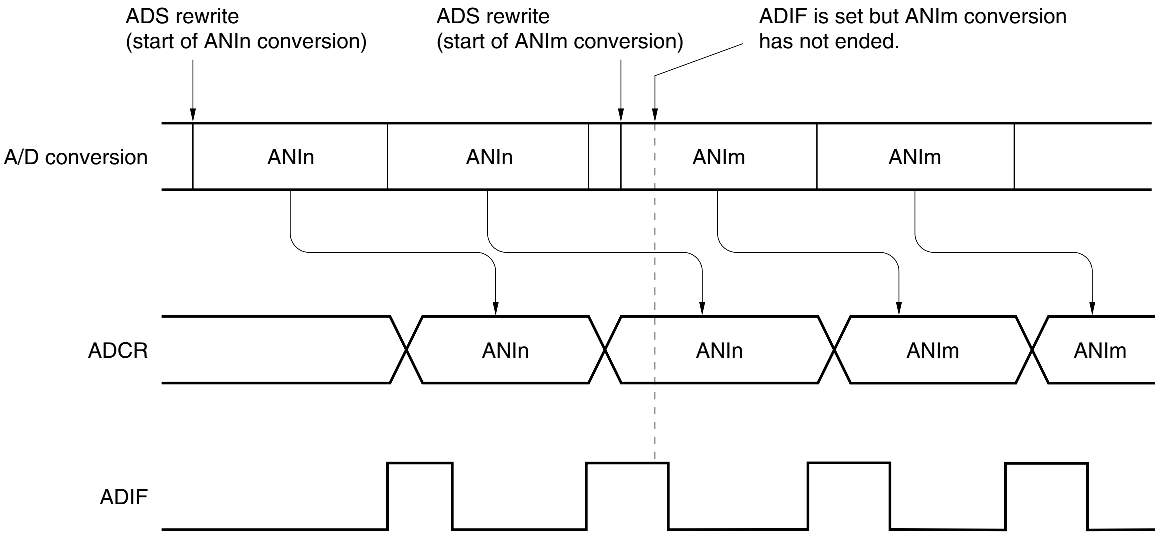
(8) Interrupt request flag (ADIF)

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 11-21. 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 (ADPC), the contents of ADCR and ADCRH may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC. Using a timing other than the above may cause an incorrect conversion result to be read.

(11) Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 11-22. Internal Equivalent Circuit of ANIn Pin

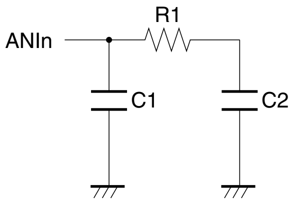


Table 11-4. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

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

Remarks 1. The resistance and capacitance values shown in **Table 11-4** are not guaranteed values.

2. n = 0 to 7

CHAPTER 12 SERIAL INTERFACES UART60 AND UART61

The 78K0/Dx2 incorporate serial interfaces UART60 and UART61.

12.1 Functions of Serial Interfaces UART60 and UART61

Caution Serial interface UART61 is for 78K0/DF2 only.

Serial interfaces UART60 and UART61 have 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 **12.4.1 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 **12.4.2 Asynchronous serial interface (UART) mode** and **12.4.3 Dedicated baud rate generator**.

- Maximum transfer rate: 625 kbps
- Two-pin configuration TxD6n: Transmit data output pin
  RxD6n: Receive data input pin
- 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).
- Twelve operating clock inputs selectable
- 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 TXD6n 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 interfaces UART60 and UART61 are not stopped (e.g., in the HALT mode), normal operation continues. If clock supply to serial interfaces UART60 and UART61 are stopped (e.g., in the STOP mode), each register stops operating, and holds the value immediately before clock supply was stopped. The TXD6n pins 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 by setting POWER6n = 0, RXE6n = 0, and TXE6n = 0.

3. Set POWER6n = 1 and then set TXE6n = 1 (transmission) or RXE6n = 1 (reception) to start communication.

4. TXE6n and RXE6n are synchronized by the base clock (f_{XCLK6}) set by CKSR6n. To enable transmission or reception again, set TXE6n or RXE6n to 1 at least two clocks of the base clock after TXE6n or RXEn6 has been cleared to 0. If TXE6n or RXE6n is set within two clocks of the base clock, the transmission circuit or reception circuit may not be initialized.

Cautions 5. Set transmit data to TXB6n at least one base clock (f_{XCLK6}) after setting TXE6n = 1.

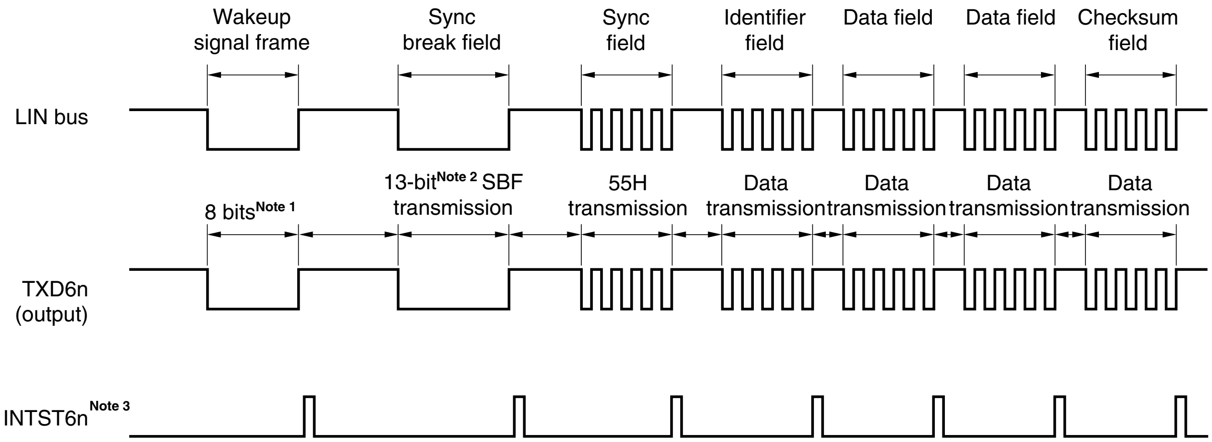
6. If data is continuously transmitted, the communication timing from the stop bit to the next start bit is extended two operating clocks of the macro. However, this does not affect the result of communication because the reception side initializes the timing when it has detected a start bit. Do not use the continuous transmission function if the interface is used in LIN communication operation.

Remarks 1. 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.
n 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.

2. n = 0, 1

Figures 12-1 and **12-2** outline the transmission and reception operations of LIN.

Figure 12-1. LIN Transmission Operation



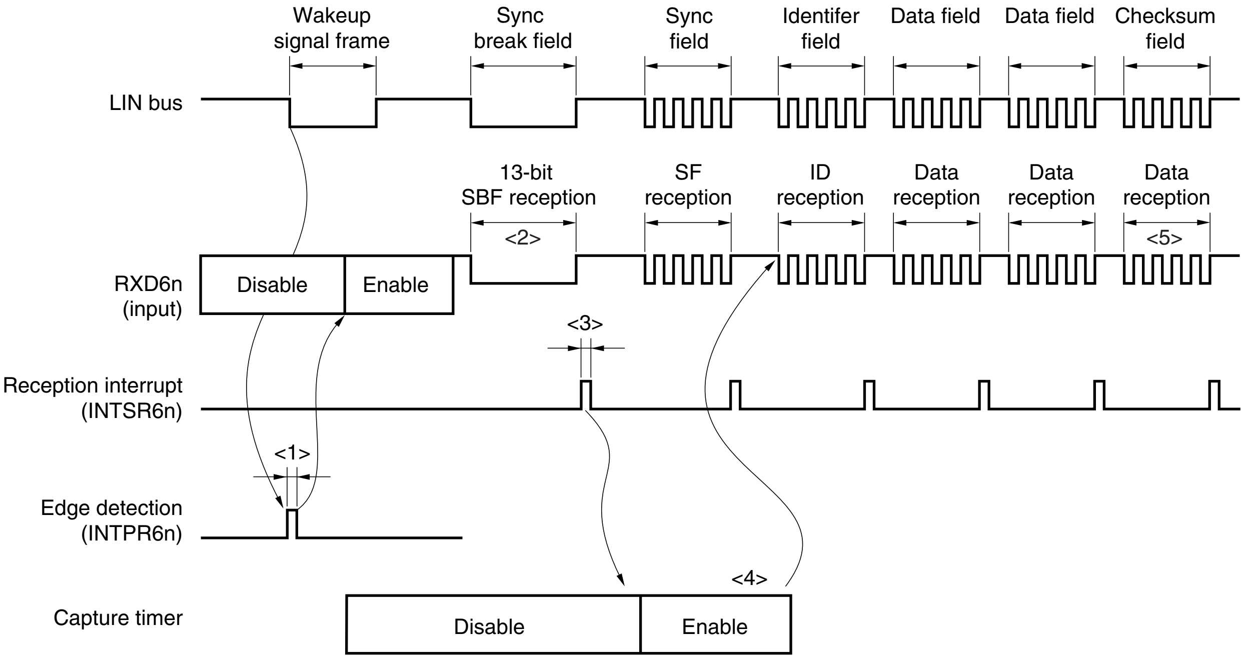
Notes 1. The wakeup signal frame is substituted by 80H transmission in the 8-bit mode.

2. The sync break field is output by hardware. The output width is the bit length set by bits 4 to 2 (SBL62n to SBL60n) of asynchronous serial interface control register 6n (ASICL6n). If more precise output width adjustment is necessary, use baud rate generator control register 6n (BRGC6n) (see **12.4.2 (2) (h) SBF transmission**).

3. INTST6n 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.
n = 0, 1

Figure 12-2. LIN Reception Operation



Reception processing is as follows.

<1> The wakeup signal is detected at the edge of the pin, and enables UART6n 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 **6.5.7 Pulse width measurement mode (TPnMD2 to TPnMD0 bits = 110)**). Detection of errors OVE6n, PE6n, and FE6n is suppressed, and error detection processing of UART communication and data transfer of the shift register and RXB6n is not performed. The shift register holds the reset value FFH.

<4> Calculate the baud rate error from the bit length of the sync field, disable UART6n after SF reception, and then re-set baud rate generator control register 6n (BRGC6n).

<5> Distinguish the checksum field by software. Also perform processing by software to initialize UART6n after reception of the checksum field and to set the SBF reception mode again.

Remark n = 0, 1

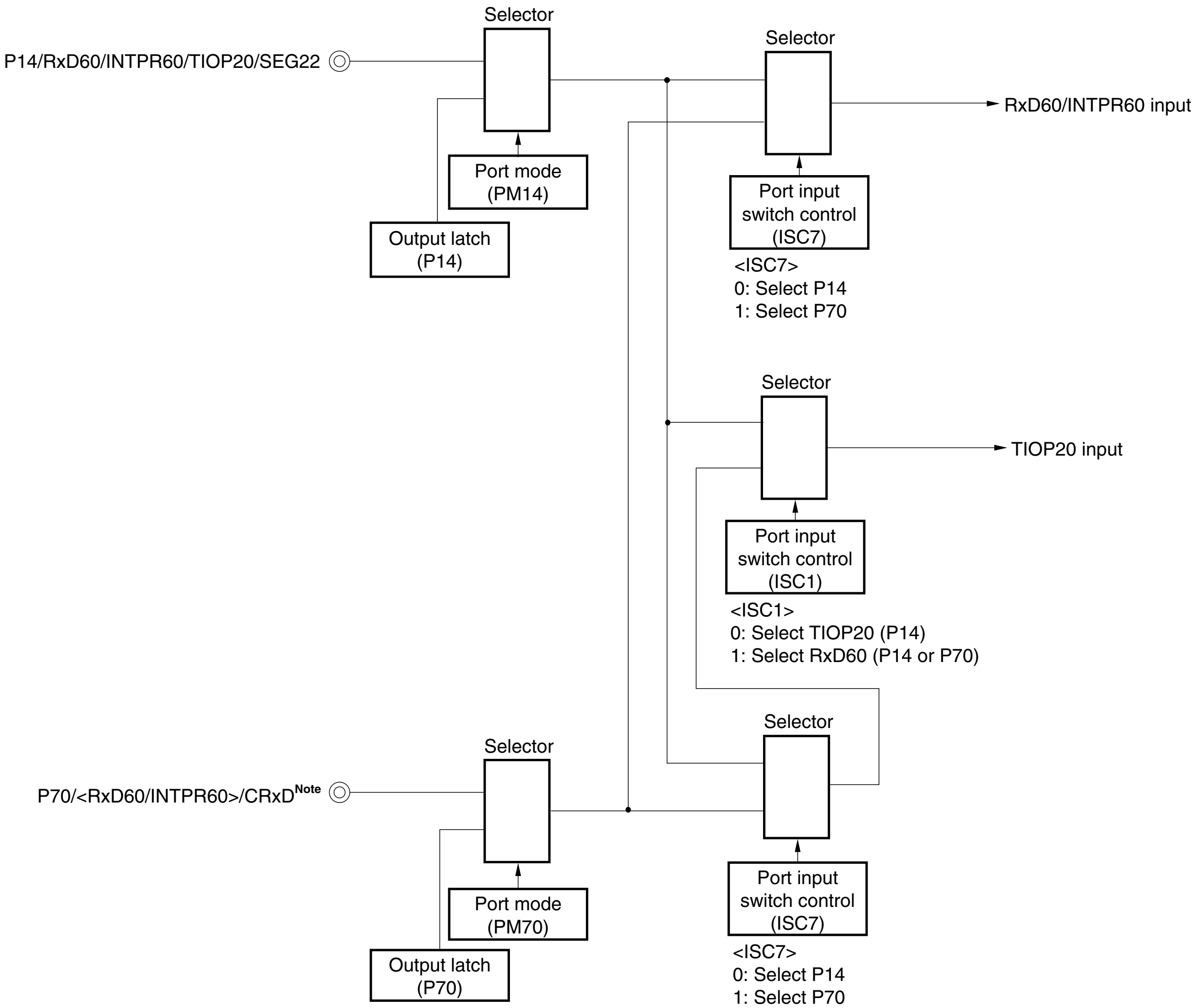
Figures 12-3 and **12-4** show the port configuration for LIN reception operation.

The wakeup signal transmitted from the LIN master is received by detecting the edge of the external interrupt (INTPR60 and INTPR61). 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 P2, P3, and the baud rate error can be calculated.

The input source of the reception port input (RxD60 and RxD61) can be input to 16-bit timer/event counter P2, P3 by port input switch control (ISC), without connecting RxD60/INTPR60, RxD61/INTPR61, TIOP20 and TIOP30 externally.

Figure 12-3. Port Configuration for LIN Reception Operation (UART60) (1/2)

(a) 78K0/DE2

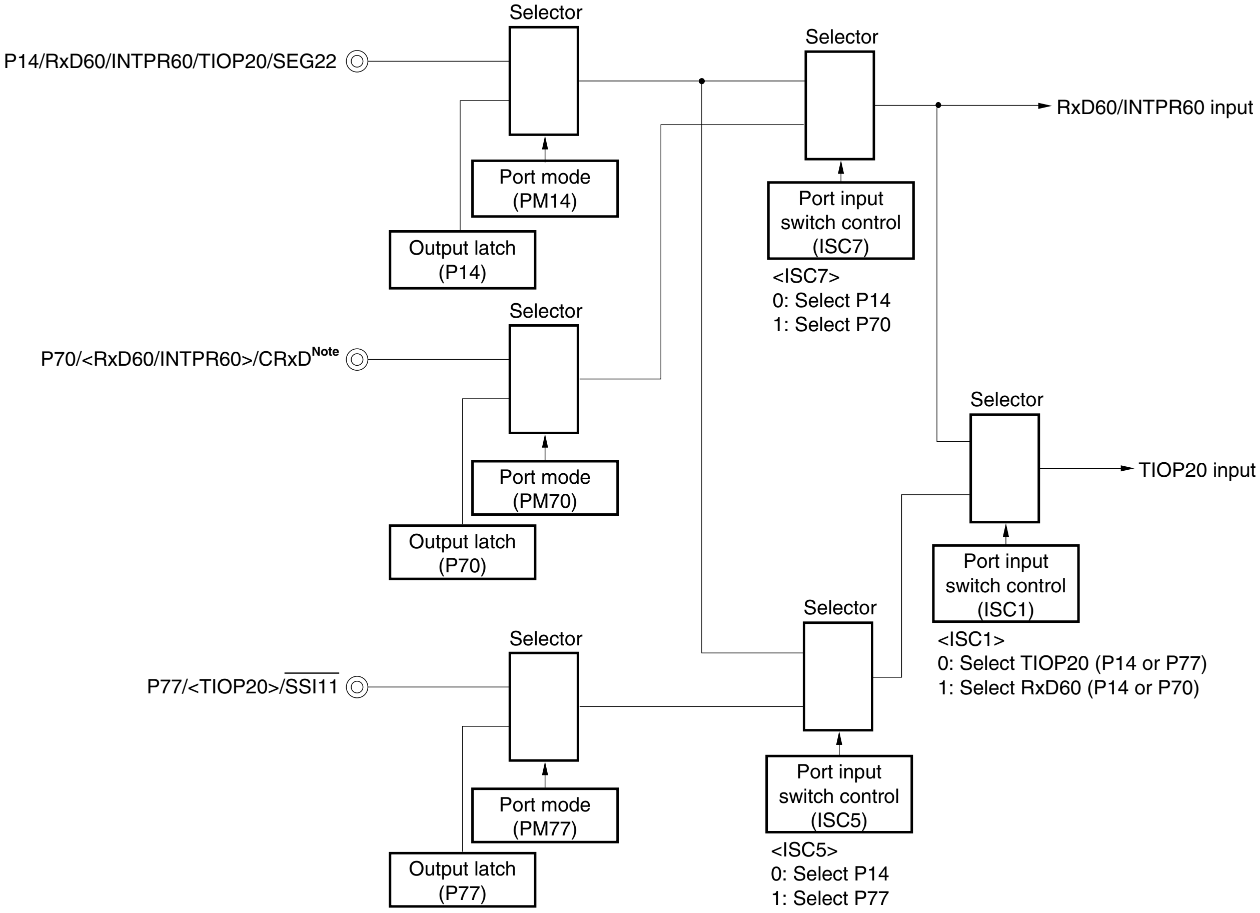


Note μPD78F0844 and 78F0845 only.

Remark ISC1, ISC7: Bits 1 and 7 of the input switch control register (ISC) (see **Figure 12-19. Format of Input Switch Control Register (ISC)**.)

Figure 12-3. Port Configuration for LIN Reception Operation (UART60) (2/2)

(b) 78K0/DF2



Note μPD78F0846, 78F0847, 78F0848, and 78F0849 only.

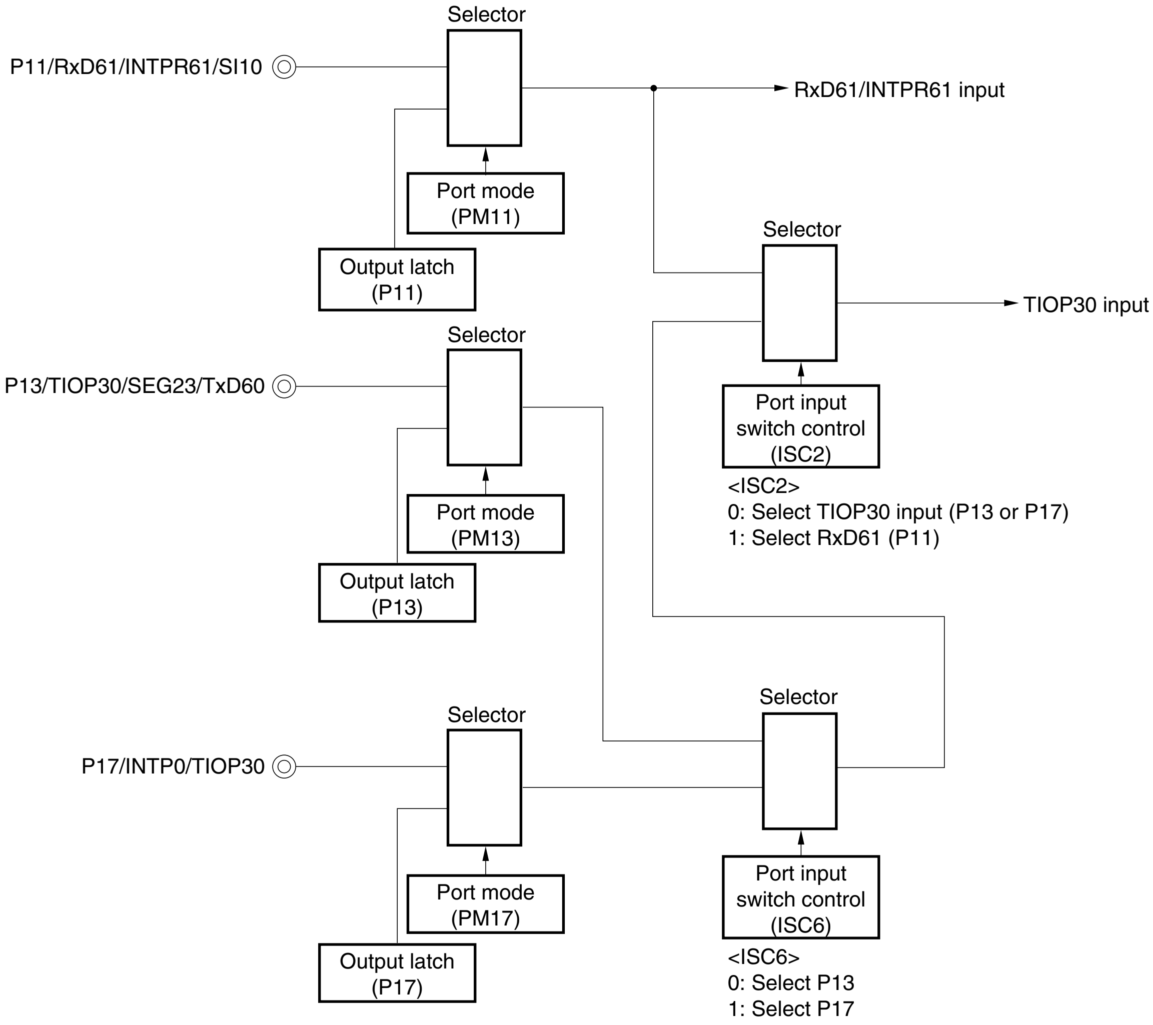
Remark ISC1, ISC5, ISC7: Bits 1, 5, and 7 of the input switch control register (ISC) (see **Figure 12-19. Format of Input Switch Control Register (ISC)**.)

The peripheral functions used in the LIN communication operation are shown below.

<Peripheral functions used>

- External interrupt (INTPR60); wakeup signal detection
  Use: Detects the wakeup signal edges and detects start of communication.
- 16-bit timer/event counter P2 (TMP2); baud rate error detection
  Use: Detects the baud rate error (measures the TMP2 input edge interval in the capture mode) by detecting the sync field (SF) length and divides it by the number of bits.
- Serial interface UART60.

Figure 12-4. Port Configuration for LIN Reception Operation (UART61)



Remark ISC2, ISC6: Bits 2 and 6 of the input switch control register (ISC) (see **Figure 12-19. Format of Input Switch Control Register (ISC)**.)

The peripheral functions used in the LIN communication operation are shown below.

<Peripheral functions used>

- External interrupt (INTPR61); wakeup signal detection

  Use: Detects the wakeup signal edges and detects start of communication.
- 16-bit timer/event counter P3 (TMP3); baud rate error detection

  Use: Detects the baud rate error (measures the TMP3 input edge interval in the capture mode) by detecting the sync field (SF) length and divides it by the number of bits.
- Serial interface UART61.

12.2 Configurations of Serial Interface UART60 and UART61

Serial interfaces UART60 and UART61 include the following hardware.

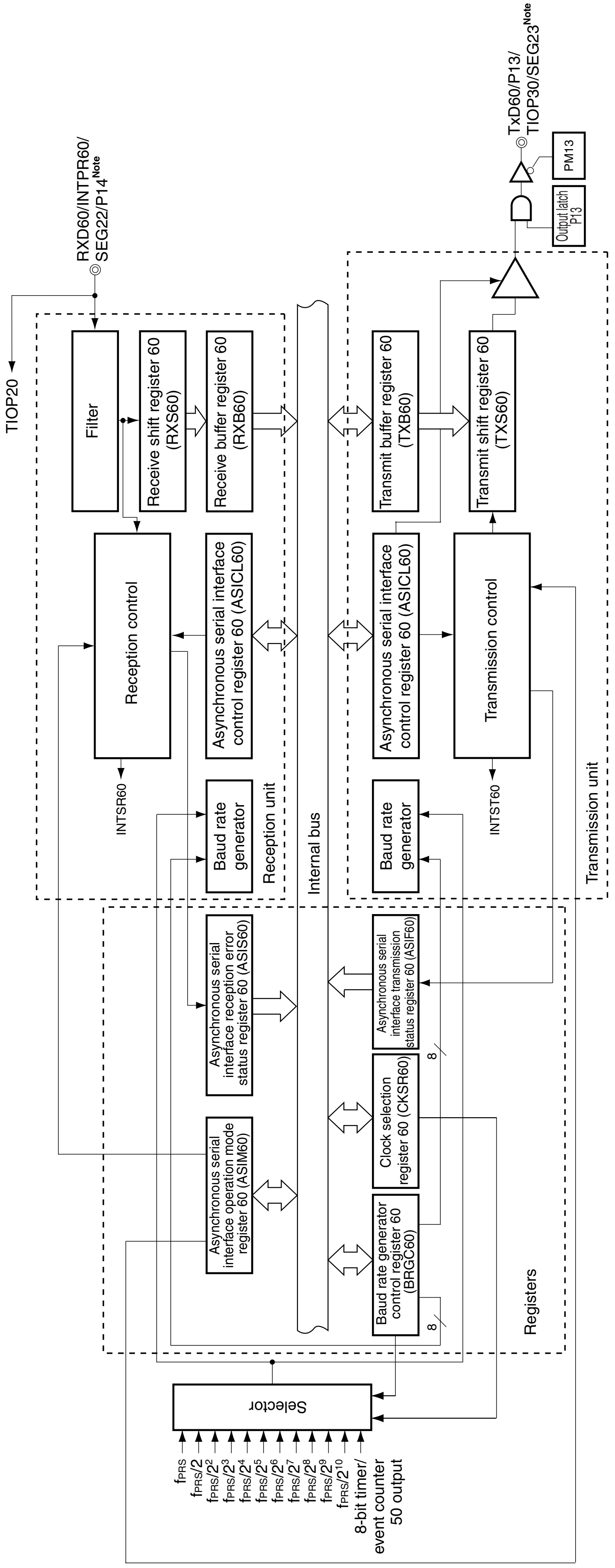
Table 12-1. Configurations of Serial Interface UART60 and UART61

Item	Configuration
Registers	Receive buffer register 6n (RXB6n) Receive shift register 6n (RXS6n) Transmit buffer register 6n (TXB6n) Transmit shift register 6n (TXS6n)
Control registers	Asynchronous serial interface operation mode register 6n (ASIM6n) Asynchronous serial interface reception error status register 6n (ASIS6n) Asynchronous serial interface transmission status register 6n (ASIF6n) Clock selection register 6n (CKSR6n) Baud rate generator control register 6n (BRGC6n) Asynchronous serial interface control register 6n (ASICL6n) Input switch control register (ISC) Port mode registers 1 and 7 (PM1, PM7) Port registers 1 and 7 (P1, P7)

Remark n = 0, 1

<R>

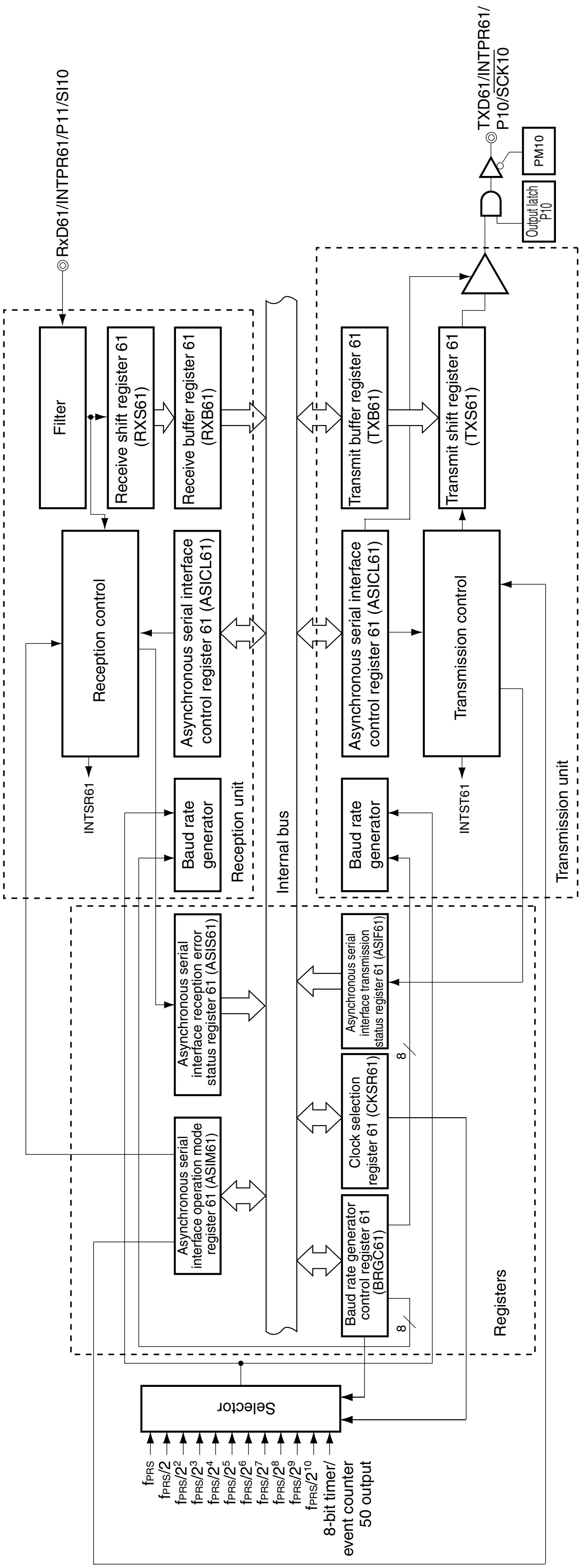
Figure 12-5. Block Diagram of Serial Interface UART60


RXD60/INTPR60/
SEG22/P14Note
TIOP20
Filter
Receive shift register 60 (RXS60)
Receive buffer register 60 (RXB60)
Reception control
Asynchronous serial interface control register 60 (ASICL60)
INTSR60
Baud rate generator
Reception unit
Internal bus
Transmit buffer register 60 (TXB60)
Transmit shift register 60 (TXS60)
Asynchronous serial interface control register 60 (ASICL60)
Transmission control
INTST60
Baud rate generator
Transmission unit
TxD60/P13/
TIOP30/SEG23Note
Output latch P13
PM13
Asynchronous serial interface reception error status register 60 (ASIS60)
Asynchronous serial interface operation mode register 60 (ASIM60)
Asynchronous serial interface transmission status register 60 (ASIF60)
Clock selection register 60 (CKSR60)
Baud rate generator control register 60 (BRGC60)
8
8
Registers
Selector
fPRS
fPRS/2
fPRS/2^2
fPRS/2^3
fPRS/2^4
fPRS/2^5
fPRS/2^6
fPRS/2^7
fPRS/2^8
fPRS/2^9
fPRS/2^10
8-bit timer/
event counter
50 output

Note Selectable with input switch control register (ISC).

<R>

Figure 12-6. Block Diagram of Serial Interface UART61



(1) Receive buffer register 6n (RXB6n)

This 8-bit register stores parallel data converted by receive shift register 6n (RXS6n).

Each time 1 byte of data has been received, new receive data is transferred to this register from RXS6n. 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 RXB6n and the MSB of RXB6n is always 0.
- In MSB-first reception, the receive data is transferred to bits 1 to 7 of RXB6n and the LSB of RXB6n is always 0.

If an overrun error (OVE6n) occurs, the receive data is not transferred to RXB6n.

RXB6n 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 6n (RXS6n)

This register converts the serial data input to the RxD6n pins into parallel data.

RXS6n cannot be directly manipulated by a program.

(3) Transmit buffer register 6n (TXB6n)

This buffer register is used to set transmit data. Transmission is started when data is written to TXB6n.

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 TXB6n when bit 1 (TXBF6n) of asynchronous serial interface transmission status register 6n (ASIF6n) is 1.

2. Do not refresh (write the same value to) TXB6n by software during a communication operation (when bits 7 and 6 (POWER6n, TXE6n) of asynchronous serial interface operation mode register 6n (ASIM6n) are 1 or when bits 7 and 5 (POWER6n, RXE6n) of ASIM6n are 1).

3. Set transmit data to TXB6n at least one base clock (f_{XCLK6}) after setting TXE6n = 1.

(4) Transmit shift register 6n (TXS6n)

This register transmits the data transferred from TXB6n from the TxD6n pins as serial data. Data is transferred from TXB6n immediately after TXB6n is written for the first transmission, or immediately before INTST6n occurs after one frame was transmitted for continuous transmission. Data is transferred from TXB6n and transmitted from the TxD6n pins at the falling edge of the base clock.

TXS6n cannot be directly manipulated by a program.

Remark n = 0, 1

12.3 Registers Controlling Serial Interfaces UART60 and UART61

Serial interfaces UART60 and UART61 are controlled by the following nine registers.

- Asynchronous serial interface operation mode register 6n (ASIM6n)
- Asynchronous serial interface reception error status register 6n (ASIS6n)
- Asynchronous serial interface transmission status register 6n (ASIF6n)
- Clock selection register 6n (CKSR6n)
- Baud rate generator control register 6n (BRGC6n)
- Asynchronous serial interface control register 6n (ASICL6n)
- Input switch control register (ISC)
- Port mode registers 1 and 7 (PM1, PM7)
- Port registers 1 and 7 (P1, P7)

Remark n = 0, 1 (the registers given with n = 1 are for 78K0/DF2 only).

(1) Asynchronous serial interface operation mode register 6n (ASIM6n)

This 8-bit register controls the serial communication operations of serial interface UART60 and UART61.

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

Reset signal generation sets this register to 01H.

Remarks 1. ASIM6n can be refreshed (the same value is written) by software during a communication operation (when bits 7 and 6 (POWER6n, TXE6n) of ASIM6n = 1 or bits 7 and 5 (POWER6n, RXE6n) of ASIM6n = 1).

2. n = 0, 1

Figure 12-7. Format of Asynchronous Serial Interface Operation Mode Register 60 (ASIM60) (1/2)

Address: FF50H After reset: 01H R/W

<R>

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM60	POWER60	TXE60	RXE60	PS610	PS600	CL60	SL60	1

POWER60	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

TXE60	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).
1	Enables transmission

RXE60	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).
1	Enables reception

Notes 1. The output of the TxD60 pins goes high level and the input from the RxD60 pins is fixed to the high level when POWER60 = 0 during transmission.

2. Asynchronous serial interface reception error status register 60 (ASIS60), asynchronous serial interface transmission status register 60 (ASIF60), bit 7 (SBRF60) and bit 6 (SBRT60) of asynchronous serial interface control register 60 (ASICL60), and receive buffer register 60 (RXB60) are reset.

Figure 12-7. Format of Asynchronous Serial Interface Operation Mode Register 60 (ASIM60) (2/2)

PS610	PS600	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.

CL60	Specifies character length of transmit/receive data
0	Character length of data = 7 bits
1	Character length of data = 8 bits

SL60	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 (PE60) of asynchronous serial interface reception error status register 60 (ASIS60) is not set and the error interrupt does not occur.

<R> **Cautions 1. Be sure to set bit 0 to 1.**

2. To start the transmission, set POWER60 to 1 and then set TXE60 to 1. To stop the transmission, clear TXE60 to 0, and then clear POWER60 to 0.

3. To start the reception, set POWER60 to 1 and then set RXE60 to 1. To stop the reception, clear RXE60 to 0, and then clear POWER60 to 0.

4. Set POWER60 to 1 and then set RXE60 to 1 while a high level is input to the RxD60 pins. If POWER60 is set to 1 and RXE60 is set to 1 while a low level is input, reception is started.

5. TXE60 and RXE60 are synchronized by the base clock (f_{XCLK6}) set by CKSR60. To enable transmission or reception again, set TXE60 or RXE60 to 1 at least two clocks of the base clock after TXE60 or RXE60 has been cleared to 0. If TXE60 or RXE60 is set within two clocks of the base clock, the transmission circuit or reception circuit may not be initialized.

6. Set transmit data to TXB60 at least one base clock (f_{XCLK6}) after setting TXE60 = 1.

7. Clear the TXE60 and RXE60 bits to 0 before rewriting the PS610, PS600, and CL60 bits.

8. Fix the PS610 and PS600 bits to 0 when used in LIN communication operation.

9. Clear TXE60 to 0 before rewriting the SL60 bit. Reception is always performed with “the number of stop bits = 1”, and therefore, is not affected by the set value of the SL60 bit.

<R>

Figure 12-8. Format of Asynchronous Serial Interface Operation Mode Register 61 (ASIM61) (1/2)

Address: FF2EH After reset: 01H R/W

<R>

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM61	POWER61	TXE61	RXE61	PS611	PS601	CL61	SL61	1

POWER61	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

TXE61	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).
1	Enables transmission

RXE61	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).
1	Enables reception

Notes 1. The output of the TXD61 pins goes high level and the input from the RXD61 pins is fixed to the high level when POWER61 = 0 during transmission.

2. Asynchronous serial interface reception error status register 61 (ASIS61), asynchronous serial interface transmission status register 61 (ASIF61), bit 7 (SBRF61) and bit 6 (SBRT61) of asynchronous serial interface control register 61 (ASICL61), and receive buffer register 61 (RXB61) are reset.

Figure 12-8. Format of Asynchronous Serial Interface Operation Mode Register 61 (ASIM61) (2/2)

PS611	PS601	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.

CL61	Specifies character length of transmit/receive data
0	Character length of data = 7 bits
1	Character length of data = 8 bits

SL61	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 (PE61) of asynchronous serial interface reception error status register 61 (ASIS61) is not set and the error interrupt does not occur.

<R> **Cautions 1. Be sure to set bit 0 to 1.**

2. To start the transmission, set POWER61 to 1 and then set TXE61 to 1. To stop the transmission, clear TXE61 to 0, and then clear POWER61 to 0.

3. To start the reception, set POWER61 to 1 and then set RXE61 to 1. To stop the reception, clear RXE61 to 0, and then clear POWER61 to 0.

4. Set POWER61 to 1 and then set RXE61 to 1 while a high level is input to the RxD61 pins. If POWER61 is set to 1 and RXE61 is set to 1 while a low level is input, reception is started.

5. TXE61 and RXE61 are synchronized by the base clock (f_{XCLK6}) set by CKSR61. To enable transmission or reception again, set TXE61 or RXE61 to 1 at least two clocks of the base clock after TXE61 or RXE61 has been cleared to 0. If TXE61 or RXE61 is set within two clocks of the base clock, the transmission circuit or reception circuit may not be initialized.

6. Set transmit data to TXB61 at least one base clock (f_{XCLK6}) after setting TXE61 = 1.

7. Clear the TXE61 and RXE61 bits to 0 before rewriting the PS611, PS601, and CL61 bits.

8. Fix the PS611 and PS601 bits to 0 when used in LIN communication operation.

9. Clear TXE61 to 0 before rewriting the SL61 bit. Reception is always performed with “the number of stop bits = 1”, and therefore, is not affected by the set value of the SL61 bit.

<R>

(2) Asynchronous serial interface reception error status register 6n (ASIS6n)

This register indicates an error status on completion of reception by serial interfaces UART60 and UART61. It includes three error flag bits (PE6n, FE6n, OVE6n).

This register is read-only by an 8-bit memory manipulation instruction.

Reset signal generation, or clearing bit 7 (POWER6n) or bit 5 (RXE6n) of ASIM6n to 0 clears this register to 00H. 00H is read when this register is read. If a reception error occurs, read ASIS6n and then read receive buffer register 6n (RXB6n) to clear the error flag.

Remark n = 0, 1

Figure 12-9. Format of Asynchronous Serial Interface Reception Error Status Register 60 (ASIS60)

Address: FF53H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS60	0	0	0	0	0	PE60	FE60	OVE60

PE60	Status flag indicating parity error
0	If POWER60 = 0 and RXE60 = 0, or if ASIS60 register is read
1	If the parity of transmit data does not match the parity bit on completion of reception

FE60	Status flag indicating framing error
0	If POWER60 = 0 and RXE60 = 0, or if ASIS60 register is read
1	If the stop bit is not detected on completion of reception

OVE60	Status flag indicating overrun error
0	If POWER60 = 0 and RXE60 = 0, or if ASIS60 register is read
1	If receive data is set to the RXB60 register and the next reception operation is completed before the data is read.

Cautions 1. The operation of the PE60 bit differs depending on the set values of the PS610 and PS600 bits of asynchronous serial interface operation mode register 60 (ASIM60).

2. 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 60 (RXB60) but discarded.

4. If data is read from ASIS60, a wait cycle is generated. Do not read data from ASIS60 when the CPU is operating on the subsystem clock and the high-speed system clock is stopped. For details, see CHAPTER 32 CAUTIONS FOR WAIT.

Figure 12-10. Format of Asynchronous Serial Interface Reception Error Status Register 61 (ASIS61)

Address: FF2FH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS61	0	0	0	0	0	PE61	FE61	OVE61

PE61	Status flag indicating parity error
0	If POWER61 = 0 and RXE61 = 0, or if ASIS61 register is read
1	If the parity of transmit data does not match the parity bit on completion of reception

FE61	Status flag indicating framing error
0	If POWER61 = 0 and RXE61 = 0, or if ASIS61 register is read
1	If the stop bit is not detected on completion of reception

OVE61	Status flag indicating overrun error
0	If POWER61 = 0 and RXE61 = 0, or if ASIS61 register is read
1	If receive data is set to the RXB61 register and the next reception operation is completed before the data is read.

Cautions 1. The operation of the PE61 bit differs depending on the set values of the PS611 and PS601 bits of asynchronous serial interface operation mode register 61 (ASIM61).

2. 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 61 (RXB61) but discarded.

4. If data is read from ASIS61, a wait cycle is generated. Do not read data from ASIS6 when the CPU is operating on the subsystem clock and the high-speed system clock is stopped. For details, see CHAPTER 32 CAUTIONS FOR WAIT.

(3) Asynchronous serial interface transmission status register 6n (ASIF6n)

This register indicates the status of transmission by serial interfaces UART60 and UART61. It includes two status flag bits (TXBF6n and TXSF6n).

Transmission can be continued without disruption even during an interrupt period, by writing the next data to the TXB6n register after data has been transferred from the TXB6n register to the TXS6n register.

This register is read-only by an 8-bit memory manipulation instruction.

Reset signal generation, or clearing bit 7 (POWER6n) or bit 6 (TXE6n) of ASIM6n to 0 clears this register to 00H.

Remark n = 0, 1

Figure 12-11. Format of Asynchronous Serial Interface Transmission Status Register 60 (ASIF60)

Address: FF55H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIF60	0	0	0	0	0	0	TXBF60	TXSF60

TXBF60	Transmit buffer data flag
0	If POWER60 = 0 or TXE60 = 0, or if data is transferred to transmit shift register 60 (TXS60)
1	If data is written to transmit buffer register 60 (TXB60) (if data exists in TXB60)

TXSF60	Transmit shift register data flag
0	If POWER60 = 0 or TXE60 = 0, or if the next data is not transferred from transmit buffer register 60 (TXB60) after completion of transfer
1	If data is transferred from transmit buffer register 60 (TXB60) (if data transmission is in progress)

Cautions 1. To transmit data continuously, write the first transmit data (first byte) to the TXB60 register. Be sure to check that the TXBF60 flag is “0”. If so, write the next transmit data (second byte) to the TXB60 register. If data is written to the TXB60 register while the TXBF60 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 TXSF60 flag is “0” after generation of the transmission completion interrupt, and then execute initialization. If initialization is executed while the TXSF60 flag is “1”, the transmit data cannot be guaranteed.

Figure 12-12. Format of Asynchronous Serial Interface Transmission Status Register 61 (ASIF61)

Address: FF38H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIF61	0	0	0	0	0	0	TXBF61	TXSF61

TXBF61	Transmit buffer data flag
0	If POWER61 = 0 or TXE61 = 0, or if data is transferred to transmit shift register 61 (TXS61)
1	If data is written to transmit buffer register 61 (TXB61) (if data exists in TXB61)

TXSF61	Transmit shift register data flag
0	If POWER61 = 0 or TXE61 = 0, or if the next data is not transferred from transmit buffer register 61 (TXB61) after completion of transfer
1	If data is transferred from transmit buffer register 61 (TXB61) (if data transmission is in progress)

Cautions 1. To transmit data continuously, write the first transmit data (first byte) to the TXB61 register. Be sure to check that the TXBF61 flag is “0”. If so, write the next transmit data (second byte) to the TXB61 register. If data is written to the TXB61 register while the TXBF61 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 TXSF61 flag is “0” after generation of the transmission completion interrupt, and then execute initialization. If initialization is executed while the TXSF61 flag is “1”, the transmit data cannot be guaranteed.

(4) Clock selection register 6n (CKSR6n)

This register selects the base clocks of serial interface UART60 and UART61.

CKSR6n can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Remarks1. CKSR6n can be refreshed (the same value is written) by software during a communication operation (when bits 7 and 6 (POWER6n, TXE6n) of ASIM6n = 1 or bits 7 and 5 (POWER6n, RXE6n) of ASIM6n = 1).

2. n = 0, 1

Figure 12-13. Format of Clock Selection Register 60 (CKSR60)

Address: FF56H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CKSR60	0	0	0	0	TPS630	TPS620	TPS610	TPS600

TPS630	TPS620	TPS610	TPS600	Base clock (f_{XCLK6}) selection[Note 1]				
					f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	0	f_{PRS}	2 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{PRS}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	312.5 kHz	625 kHz
0	1	1	0	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz
0	1	1	1	$f_{PRS}/2^7$	15.625 kHz	39.06 kHz	78.13 kHz	156.25 kHz
1	0	0	0	$f_{PRS}/2^8$	7.813 kHz	19.53 kHz	39.06 kHz	78.13 kHz
1	0	0	1	$f_{PRS}/2^9$	3.906 kHz	9.77 kHz	19.53 kHz	39.06 kHz
1	0	1	0	$f_{PRS}/2^{10}$	1.953 kHz	4.88 kHz	9.77 kHz	19.53 kHz
1	0	1	1	TM50 output[Note 2]				
Other than above				Setting prohibited				

Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{IN}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 4.0 to 5.5 V: $f_{PRS} \leq 20$ MHz
- V_{DD} = 2.7 to 4.0 V: $f_{PRS} \leq 10$ 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 the TO50 pin as a timer output pin in any mode.

Caution Make sure POWER60 = 0 when rewriting TPS630 to TPS600.

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 12-14. Format of Clock Selection Register 61 (CKSR61)

Address: FF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CKSR61	0	0	0	0	TPS631	TPS621	TPS611	TPS601

TPS631	TPS621	TPS611	TPS601	Base clock (f_{XCLK6}) selection[Note 1]				
					f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz
0	0	0	0	f_{PRS}	2 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{PRS}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	312.5 kHz	625 kHz
0	1	1	0	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz
0	1	1	1	$f_{PRS}/2^7$	15.625 kHz	39.06 kHz	78.13 kHz	156.25 kHz
1	0	0	0	$f_{PRS}/2^8$	7.813 kHz	19.53 kHz	39.06 kHz	78.13 kHz
1	0	0	1	$f_{PRS}/2^9$	3.906 kHz	9.77 kHz	19.53 kHz	39.06 kHz
1	0	1	0	$f_{PRS}/2^{10}$	1.953 kHz	4.88 kHz	9.77 kHz	19.53 kHz
1	0	1	1	TM50 output[Note 2]				
Other than above				Setting prohibited				

Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{IN}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 4.0 to 5.5 V: $f_{PRS} \leq 20$ MHz
- V_{DD} = 2.7 to 4.0 V: $f_{PRS} \leq 10$ 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 the TO50 pin as a timer output pin in any mode.

Caution Make sure POWER61 = 0 when rewriting TPS631 to TPS601.

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 6n (BRGC6n)

This register sets the division value of the 8-bit counters of serial interface UART60 and UART61.

BRGC6n can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Remarks1. BRGC6n can be refreshed (the same value is written) by software during a communication operation (when bits 7 and 6 (POWER6n, TXE6n) of ASIM6n = 1 or bits 7 and 5 (POWER6n, RXE6n) of ASIM6n = 1).

2. n = 0, 1

Figure 12-15. Format of Baud Rate Generator Control Register 60 (BRGC60)

Address: FF57H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
BRGC60	MDL670	MDL660	MDL650	MDL640	MDL630	MDL620	MDL610	MDL600

MDL670	MDL660	MDL650	MDL640	MDL630	MDL620	MDL610	MDL600	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 (TXE60) and bit 5 (RXE60) of the ASIM6n register = 0 when rewriting the MDL670 to MDL600 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 TPS630 to TPS600 bits of CKSR60 register

2. k: Value set by MDL670 to MDL600 bits (k = 4, 5, 6, ..., 255)

3. ×: Don’t care

Figure 12-16. Format of Baud Rate Generator Control Register 61 (BRGC61)

Address: FF3EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
BRGC61	MDL671	MDL661	MDL651	MDL641	MDL631	MDL621	MDL611	MDL601

MDL671	MDL661	MDL651	MDL641	MDL631	MDL621	MDL611	MDL601	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 (TXE61) and bit 5 (RXE61) of the ASIM61 register = 0 when rewriting the MDL671 to MDL601 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 TPS631 to TPS601 bits of CKSR61 register

2. k: Value set by MDL671 to MDL601 bits (k = 4, 5, 6, ..., 255)

3. ×: Don’t care

(6) Asynchronous serial interface control register 6n (ASICL6n)

This register controls the serial communication operations of serial interface UART60 and UART61.

ASICL6n can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 16H.

Caution ASICL6n can be refreshed (the same value is written) by software during a communication operation (when bits 7 and 6 (POWER6n, TXE6n) of ASIM6n = 1 or bits 7 and 5 (POWER6n, RXE6n) of ASIM6n = 1). However, do not set both SBRT6n and SBTT6n to 1 by a refresh operation during SBF reception (SBRT6n = 1) or SBF transmission (until INTST6n occurs since SBTT6n has been set (1)), because it may re-trigger SBF reception or SBF transmission.

Remark n = 0, 1

Figure 12-17. Format of Asynchronous Serial Interface Control Register 60 (ASICL60) (1/2)

Address: FF58H After reset: 16H R/W[Note]

Symbol	<7>	<6>	5	4	3	2	1	0
ASICL60	SBRF60	SBRT60	SBTT60	SBL620	SBL610	SBL600	DIR60	TXDLV60

SBRF60	SBF reception status flag
0	If POWER60 = 0 and RXE60 = 0 or if SBF reception has been completed correctly
1	SBF reception in progress

SBRT60	SBF reception trigger
0	–
1	SBF reception trigger

SBTT60	SBF transmission trigger
0	–
1	SBF transmission trigger

Note Bit 7 is read-only.

Figure 12-17. Format of Asynchronous Serial Interface Control Register 60 (ASICL60) (2/2)

SBL620	SBL610	SBL600	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.

DIR60	First-bit specification
0	MSB
1	LSB

TXDLV60	Enables/disables inverting TxD6n output
0	Normal output of TxD60
1	Inverted output of TxD60

Cautions 1. In the case of an SBF reception error, the mode returns to the SBF reception mode. The status of the SBRF60 flag is held (1).

2. Before setting the SBRT60 bit, make sure that bit 7 (POWER60) and bit 5 (RXE60) of ASIM60 = 1. After setting the SBRT60 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 SBRT60 bit is always 0. SBRT60 is automatically cleared to 0 after SBF reception has been correctly completed.

4. Before setting the SBTT60 bit to 1, make sure that bit 7 (POWER60) and bit 6 (TXE60) of ASIM60 = 1. After setting the SBTT60 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 SBTT60 bit is always 0. SBTT60 is automatically cleared to 0 at the end of SBF transmission.

6. Do not set the SBRT60 bit to 1 during reception, and do not set the SBTT60 bit to 1 during transmission.

7. Before rewriting the DIR60 and TXDLV60 bits, clear the TXE60 and RXE60 bits to 0.

8. When the TXDLV60 bit is set to 1 (inverted TxD60 output), the TxD60/P13/SEG23/TIOP30[Note] pin cannot be used as a general-purpose port, regardless of the settings of POWER60 and TXE60. When using the TxD60/P13/SEG23/TIOP30[Note] pin as a general-purpose port, clear the TXDLV60 bit to 0 (normal TxD60 output).

Note The TxD60/P13/SEG23/TIOP30 pin is selectable with input switch control (ISC) register. See **Figure 12-19. Format of Input Switch Control Register (ISC)** for details.

Figure 12-18. Format of Asynchronous Serial Interface Control Register 61 (ASICL61) (1/2)

Address: FF3FH After reset: 16H R/W[Note]

Symbol	<7>	<6>	5	4	3	2	1	0
ASICL61	SBRF61	SBRT61	SBTT61	SBL621	SBL611	SBL601	DIR61	TXDLV61

SBRF61	SBF reception status flag
0	If POWER61 = 0 and RXE61 = 0 or if SBF reception has been completed correctly
1	SBF reception in progress

SBRT61	SBF reception trigger
0	–
1	SBF reception trigger

SBTT61	SBF transmission trigger
0	–
1	SBF transmission trigger

Note Bit 7 is read-only.

Figure 12-18. Format of Asynchronous Serial Interface Control Register 61 (ASICL61) (2/2)

SBL621	SBL611	SBL601	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.

DIR61	First-bit specification
0	MSB
1	LSB

TXDLV61	Enables/disables inverting TxD6n output
0	Normal output of TxD6n
1	Inverted output of TxD6n

Cautions 1. In the case of an SBF reception error, the mode returns to the SBF reception mode. The status of the SBRF61 flag is held (1).

2. Before setting the SBRT61 bit, make sure that bit 7 (POWER61) and bit 5 (RXE61) of ASIM61 = 1. After setting the SBRT61 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 SBRT61 bit is always 0. SBRT61 is automatically cleared to 0 after SBF reception has been correctly completed.

4. Before setting the SBTT61 bit to 1, make sure that bit 7 (POWER61) and bit 6 (TXE61) of ASIM61 = 1. After setting the SBTT61 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 SBTT61 bit is always 0. SBTT61 is automatically cleared to 0 at the end of SBF transmission.

6. Do not set the SBRT61 bit to 1 during reception, and do not set the SBTT61 bit to 1 during transmission.

7. Before rewriting the DIR61 and TXDLV61 bits, clear the TXE61 and RXE61 bits to 0.

8. When the TXDLV61 bit is set to 1 (inverted TxD61 output), the TxD61/$\overline{\text{SCK10}}$/P10/INTP4 pin cannot be used as a general-purpose port, regardless of the settings of POWER61 and TXE61. When using the TxD61/$\overline{\text{SCK10}}$/P10/INTP4 pin as a general-purpose port, clear the TXDLV61 bit to 0 (normal TxD61 output).

(7) Input switch control register (ISC)

The input switch control register (ISC) is used to receive a status signal transmitted from the master during LIN (Local Interconnect Network) reception. The input source is switched by setting ISC.

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

Reset signal generation clears this register to 00H.

Figure 12-19. Format of Input Switch Control Register (ISC) (1/2)

After reset: 00H R/W Address: FFFFFF4FH

- **78K0/DE2**

Symbol	7	6	5	4	3	2	1	0
ISC	ISC7	ISC6	0	0	ISC3	0	ISC1	ISC0[Note]

Note μPD78F0844 and 78F0845 only.

<R>

ISC7	UART60 pin selection control	
	TxD60	RxD60/INTPR60
0	TxD60 (P13)	RxD60/INTPR60 (P14)
1	<TxD60> (P71)	<RxD60/INTPR60> (P70)

ISC6	TIOP30 pin selection control
0	TIOP30 (P13)
1	<TIOP30> (P17)

ISC3	TMP2 input source (TIP21) selection control [For timer conjunction function of TMP]
0	TIOP21 (P06)
1	TMP3 output signal (TOP30)

ISC1	TMP2 input source (TIP20) selection control [For LIN reception operation of UART60]
0	TIOP20 (P14)
1	RxD60[Note]

Note Selected with ISC7.

ISC0	TMP4 input source (TIP40) selection control [For time stamp function of CAN]
0	TIOP40 (P00)
1	TSOUT

Figure 12-19. Format of Input Switch Control Register (ISC) (2/2)

- **78K0/DF2**

Symbol	7	6	5	4	3	2	1	0
ISC	ISC7	ISC6	ISC5	0	ISC3	ISC2	ISC1	ISC0[Note]

Note μ PD78F0846, 78F0847, 78F0848, and 78F0849 only.

<R>

ISC7	UART60 pin selection control	
	TxD60	RxD60/INTPR60
0	TxD60 (P13)	RxD60/INTPR60 (P14)
1	<TxD60> (P71)	<RxD60/INTPR60> (P70)

ISC6	TIOP30 pin selection control
0	TIOP30 (P13)
1	<TIOP30> (P17)

ISC5	TIOP20 pin selection control
0	TIOP20 (P14)
1	<TIOP20> (P77)

ISC3	TMP2 input source (TIP21) selection control [For timer conjunction function of TMP]
0	TIOP21 (P06)
1	TMP3 output signal (TOP30)

ISC2	TMP3 input source (TIP30) selection control [For LIN reception operation of UART61]
0	TIOP30[Note]
1	RxD61 (P11)

Note Selected with ISC6.

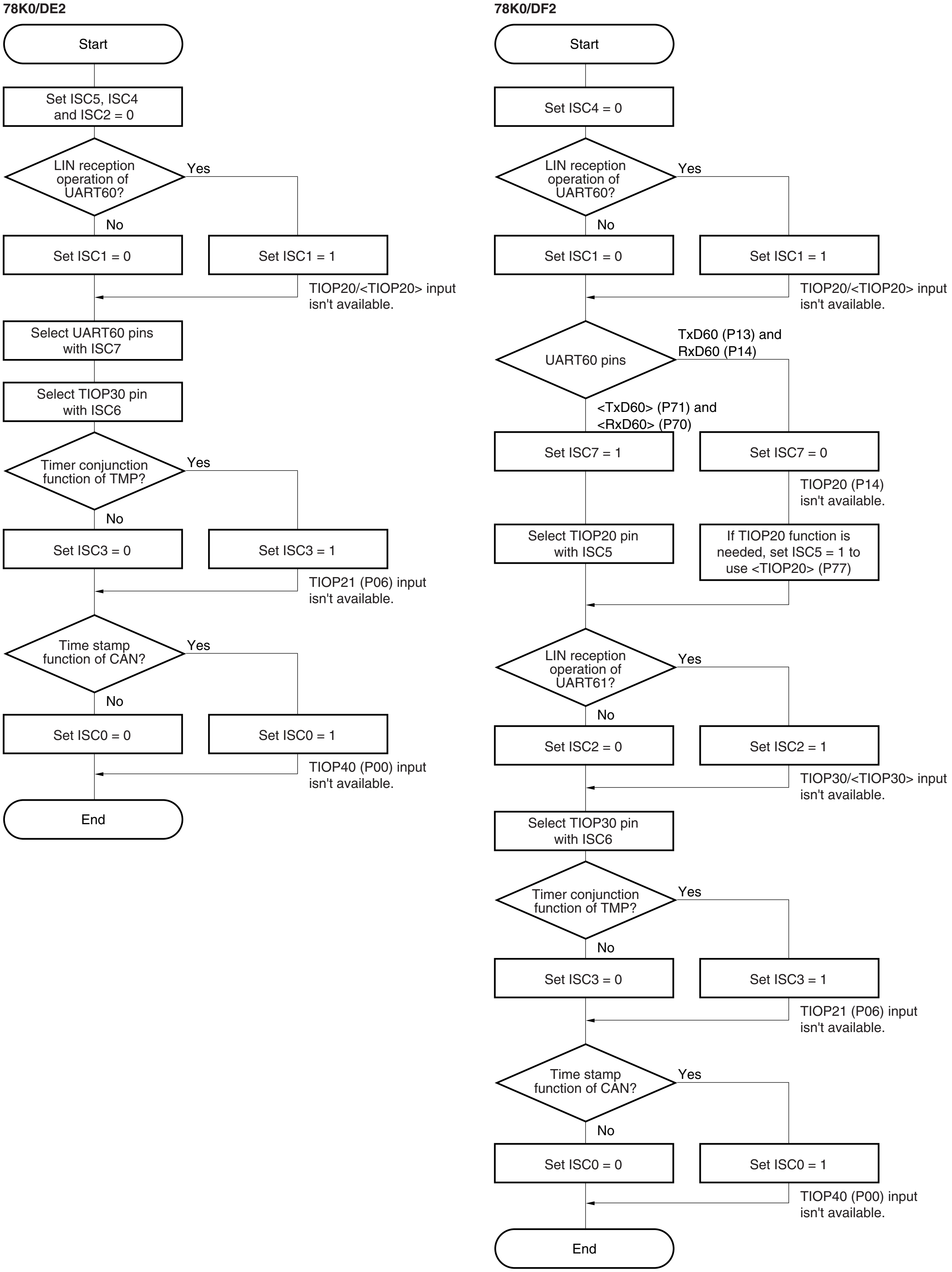
ISC1	TMP2 input source (TIP20) selection control [For LIN reception operation of UART60]
0	TIOP20 (P14)
1	RxD60[Note]

Note Selected with ISC7.

ISC0	TMP4 input source (TIP40) selection control [For time stamp function of CAN]
0	TIOP40 (P00)
1	TSOUT

<R>

Figure 12-20. ISC Register Value Selection



78K0/DE2
Start
Set ISC5, ISC4 and ISC2 = 0
LIN reception operation of UART60?
Yes
No
Set ISC1 = 0
Set ISC1 = 1
TIOP20/<TIOP20> input isn't available.
Select UART60 pins with ISC7
Select TIOP30 pin with ISC6
Timer conjunction function of TMP?
Yes
No
Set ISC3 = 0
Set ISC3 = 1
TIOP21 (P06) input isn't available.
Time stamp function of CAN?
Yes
No
Set ISC0 = 0
Set ISC0 = 1
TIOP40 (P00) input isn't available.
End
78K0/DF2
Start
Set ISC4 = 0
LIN reception operation of UART60?
Yes
No
Set ISC1 = 0
Set ISC1 = 1
TIOP20/<TIOP20> input isn't available.
UART60 pins
TxD60 (P13) and RxD60 (P14)
<TxD60> (P71) and <RxD60> (P70)
Set ISC7 = 1
Set ISC7 = 0
TIOP20 (P14) isn't available.
Select TIOP20 pin with ISC5
If TIOP20 function is needed, set ISC5 = 1 to use <TIOP20> (P77)
LIN reception operation of UART61?
Yes
No
Set ISC2 = 0
Set ISC2 = 1
TIOP30/<TIOP30> input isn't available.
Select TIOP30 pin with ISC6
Timer conjunction function of TMP?
Yes
No
Set ISC3 = 0
Set ISC3 = 1
TIOP21 (P06) input isn't available.
Time stamp function of CAN?
Yes
No
Set ISC0 = 0
Set ISC0 = 1
TIOP40 (P00) input isn't available.
End

(8) Port mode registers 1 and 7 (PM1, PM7)

PM1 sets port 1 input/output in 1-bit units. PM7 sets port 7 input/output in 1-bit units.

When using the P13/TxD60/SEG23/TIOP30 and P10/$\overline{\text{SCK10}}$/TxD61/INTP4 pins for serial interface data output, clear PM13 and PM10 to 0 and set the output latch of P13 and P10 to 1.

When using the P14/RxD60/INTPR60/SEG22/TIOP20 and P11/SI10/RxD61/INTPR61 pins for serial interface data input, set PM14 and PM11 to 1. The output latch of P14 and P11 at this time may be 0 or 1.

When the ISC7 bit of input switch control (ISC) register is set to 1, the P70/RxD60/INTPR60/CRxD[Note] pin is used for serial interface data input and the P71/TxD60/CTxD[Note] pin is used for serial interface output instead of the P13/TxD60/SEG23/TIOP30 and P14/RxD60/INTPR60/SEG22/TIOP20 pins. In this case, clear PM71 to 0 and set the output latch of P71 to 1 to use the P71/TxD60/CTxD[Note] pin. Also, set PM70 to 1 to use the P70/RxD60/INTPR60/CRxD[Note] pin. The output latch of P70 at this time may be 0 or 1.

PM1 and PM7 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Note μPD78F0844, 78F0845, 78F0846, 78F0847, 78F0848 and 78F0849 only.

Figure 12-21. Format of Port Mode Registers 1 and 7 (PM1, PM7)

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

Address: FF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	PM77[Note]	PM76[Note]	PM75[Note]	PM74[Note]	PM73	PM72	PM71	PM70

PMnm	Pnm pin I/O mode selection (n = 1, 7; m = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note These bits are set to 1 in 78K0/DE2.

12.4 Operations of Serial Interface UART60 and UART61

Serial interfaces UART60 and UART61 have the following two modes.

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

12.4.1 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 (POWER6n, TXE6n, and RXE6n) of ASIM6n to 0.

(1) Register used

The operation stop mode is set by asynchronous serial interface operation mode register 6n (ASIM6n).
ASIM6n 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
ASIM6n	POWER6n	TXE6n	RXE6n	PS61n	PS60n	CL6n	SL6n	ISRM6n

POWER6n	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].

TXE6n	Enables/disables transmission
0	Disables transmission operation (synchronously resets the transmission circuit).

RXE6n	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).

Notes 1. The output of the TxD6n pins goes high and the input from the RxD6n pins is fixed to high level when POWER6n = 0.

2. Asynchronous serial interface reception error status register 6n (ASIS6n), asynchronous serial interface transmission status register 6n (ASIF6n), bit 7 (SBRF6n) and bit 6 (SBRT6n) of asynchronous serial interface control register 6n (ASICL6n), and receive buffer register 6n (RXB6n) are reset.

Caution Clear POWER6n to 0 after clearing TXE6n and RXE6n to 0 to stop the operation.
To start the communication, set POWER6n to 1, and then set TXE6n and RXE6n to 1.

Remarks 1. To use the RxD60/INTPR60/P14/SEG22/TIOP20, RxD61/INTPR61/P11/SI10, TxD60/P13/SEG23/TIOP30, and TxD61/P10/$\overline{\text{SCK10}}$/INTP4 pins as general-purpose port pins, see **CHAPTER 4 PORT FUNCTIONS** (the RxD60/INTPR60/P14/SEG22/TIOP20 and TxD60/P13/SEG23/TIOP30 pins are selectable with input switch control (ISC) register).

2. n = 0, 1

12.4.2 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 6n (ASIM6n)
- Asynchronous serial interface reception error status register 6n (ASIS6n)
- Asynchronous serial interface transmission status register 6n (ASIF6n)
- Clock selection register 6n (CKSR6n)
- Baud rate generator control register 6n (BRGC6n)
- Asynchronous serial interface control register 6n (ASICL6n)
- Input switch control register (ISC)
- 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 CKSR6n register (see **Figures 12-13** and **12-14**).
<2> Set the BRGC6n register (see **Figures 12-15** and **12-16**).
<3> Set bits 0 to 4 (ISRM6n, SL6n, CL6n, PS60n, PS61n) of the ASIM6n register (see **Figures 12-7** and **12-8**).
<4> Set bits 0 and 1 (TXDLV6n, DIR6n) of the ASICL6n register (see **Figures 12-17** and **12-18**).
<5> Set bit 7 (POWER6n) of the ASIM6n register to 1.
<6> Set bit 6 (TXE6n) of the ASIM6n register to 1. → Transmission is enabled.
Set bit 5 (RXE6n) of the ASIM6n register to 1. → Reception is enabled.
<7> Write data to transmit buffer register 6n (TXB6n). → Data transmission is started.

Caution Take relationship with the other party of communication when setting the port mode register and port register.

Remark n = 0, 1

The relationship between the register settings and pins is shown below.

Table 12-2. Relationship between Register Settings and Pins

(a) UART60

POWER6n	TXE6n	RXE6n	PM13[Note 1]	P13[Note 1]	PM14[Note 1]	P14[Note 1]	UART60 Operation	Pin Function	
								TxD60/TIOP30/ SEG23/P13[Note 1]	RxD60/INTPR60/ TIOP20/SEG22/ P14[Note 1]
0	0	0	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	Stop	P13	P14
1	0	1	×[Note 2]	×[Note 2]	1	×	Reception	P13	RxD60
	1	0	0	1	×[Note 2]	×[Note 2]	Transmission	TxD60	P14
	1	1	0	1	1	×	Transmission/ reception	TxD60	RxD60

(b) UART61

POWER6n	TXE6n	RXE6n	PM10	P10	PM11	P11	UART61 Operation	Pin Function	
								TxD61/P10/ $\overline{\text{SCK10}}$/INTP4	RxD61/INTPR61/ P11/SI10
0	0	0	×[Note 2]	×[Note 2]	×[Note 2]	×[Note 2]	Stop	P10	P11
1	0	1	×[Note 2]	×[Note 2]	1	×	Reception	P10	RxD61
	1	0	0	1	×[Note 2]	×[Note 2]	Transmission	TxD61	P11
	1	1	0	1	1	×	Transmission/ reception	TxD61	RxD61

Notes 1. The P13/TxD60/TIOP30/SEG23 and P14/RxD60/INTPR60/TIOP20/SEG22 pins are selectable with input switch control (ISC) register. See **Figure 12-19. Format of Input Switch Control Register (ISC)** for details.

2. Can be set as port function.

Remarks 1. ×: don't care

POWER6n: Bit 7 of asynchronous serial interface operation mode register 6n (ASIM6n)

TXE6n: Bit 6 of ASIM6n

RXE6n: Bit 5 of ASIM6n

PM1×: Port mode register

P1×: Port output latch

2. n = 0, 1

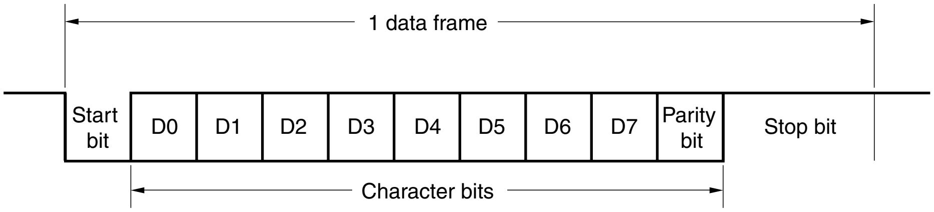
(2) Communication operation

(a) Format and waveform example of normal transmit/receive data

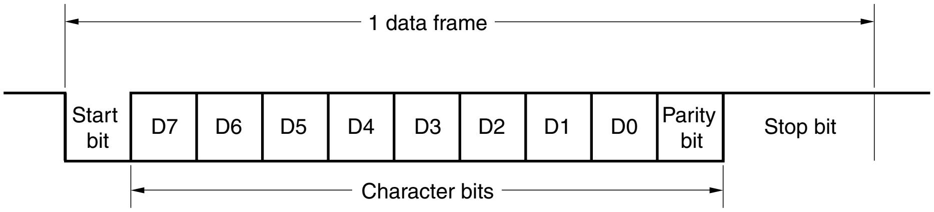
Figures 12-22 and **12-23** show the format and waveform example of the normal transmit/receive data.

Figure 12-22. 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 6n (ASIM6n).

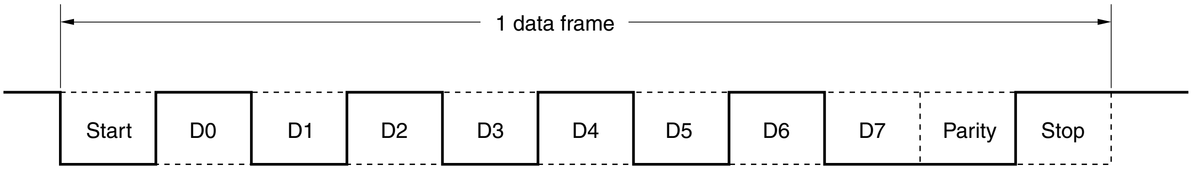
Whether data is communicated with the LSB or MSB first is specified by bit 1 (DIR6n) of asynchronous serial interface control register 6n (ASICL6n).

Whether the TxD6n pins outputs normal or inverted data is specified by bit 0 (TXDLV6n) of ASICL6n.

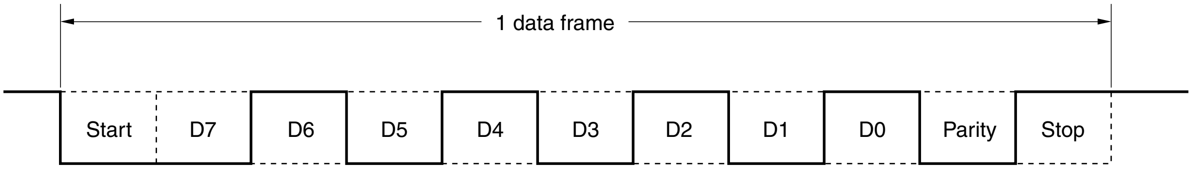
Remark n = 0, 1

Figure 12-23. Example of Normal UART Transmit/Receive Data Waveform

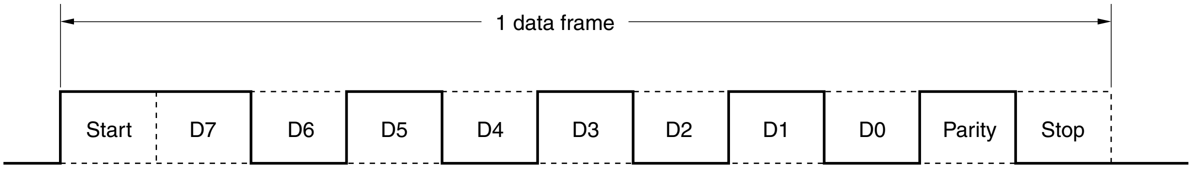
1. **Data length: 8 bits, LSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H**



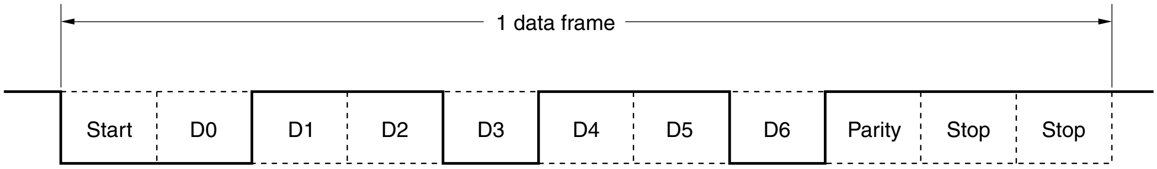
2. **Data length: 8 bits, MSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H**



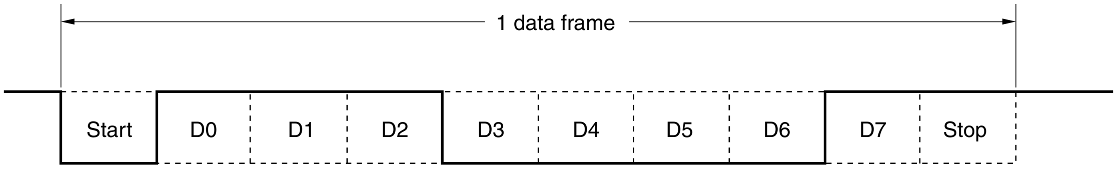
3. **Data length: 8 bits, MSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H, TxD6n pin inverted output**



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**



Remark n = 0, 1

(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 PS61n and PS60n 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.

Remark n = 0, 1

(c) Normal transmission

When bit 7 (POWER6n) of asynchronous serial interface operation mode register 6n (ASIM6n) is set to 1 and bit 6 (TXE6n) of ASIM6n is then set to 1, transmission is enabled. Transmission can be started by writing transmit data to transmit buffer register 6n (TXB6n). The start bit, parity bit, and stop bit are automatically appended to the data.

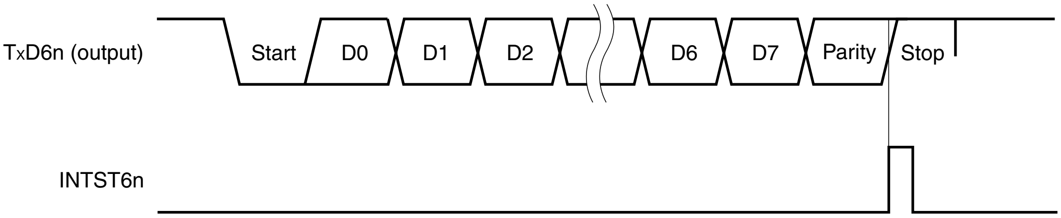
When transmission is started, the data in TXB6n is transferred to transmit shift register 6n (TXS6n). After that, the transmit data is sequentially output from TXS6n to the TxD6n pins. When transmission is completed, the parity and stop bits set by ASIM6n are appended and a transmission completion interrupt request (INTST6n) is generated.

Transmission is stopped until the data to be transmitted next is written to TXB6n.

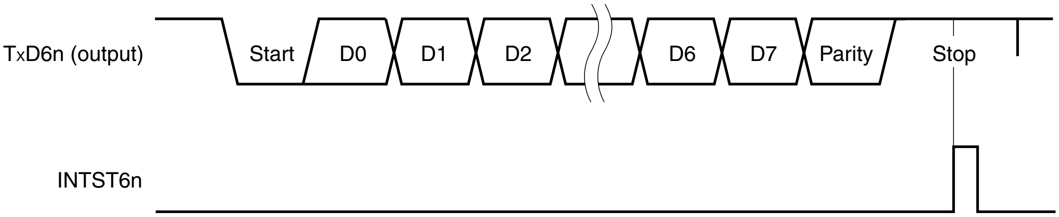
Figure 12-23 shows the timing of the transmission completion interrupt request (INTST6n). This interrupt occurs as soon as the last stop bit has been output.

Figure 12-24. Normal Transmission Completion Interrupt Request Timing

1. Stop bit length: 1



2. Stop bit length: 2



Remark n = 0, 1

(d) Continuous transmission

The next transmit data can be written to transmit buffer register 6n (TXB6n) as soon as transmit shift register 6 (TXS6n) has started its shift operation. Consequently, even while the INTST6n 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 TXB6n register can be efficiently written twice (2 bytes) without having to wait for the transmission time of one data frame, by reading bit 0 (TXSF6n) of asynchronous serial interface transmission status register 6n (ASIF6n) when the transmission completion interrupt has occurred.

To transmit data continuously, be sure to reference the ASIF6n register to check the transmission status and whether the TXB6n register can be written, and then write the data.

Cautions 1. The TXBF6n and TXSF6n flags of the ASIF6n register change from “10” to “11”, and to “01” during continuous transmission. To check the status, therefore, do not use a combination of the TXBF6n and TXSF6n flags for judgment. Read only the TXBF6n flag when executing continuous transmission.

2. When the device is used in LIN communication operation, the continuous transmission function cannot be used. Make sure that asynchronous serial interface transmission status register 6n (ASIF6n) is 00H before writing transmit data to transmit buffer register 6n (TXB6n).

TXBF6n	Writing to TXB6 Register
0	Writing enabled
1	Writing disabled

Caution To transmit data continuously, write the first transmit data (first byte) to the TXB6n register. Be sure to check that the TXBF6n flag is “0”. If so, write the next transmit data (second byte) to the TXB6n register. If data is written to the TXB6n register while the TXBF6n flag is “1”, the transmit data cannot be guaranteed.

The communication status can be checked using the TXSF6n flag.

TXSF6n	Transmission Status
0	Transmission is completed.
1	Transmission is in progress.

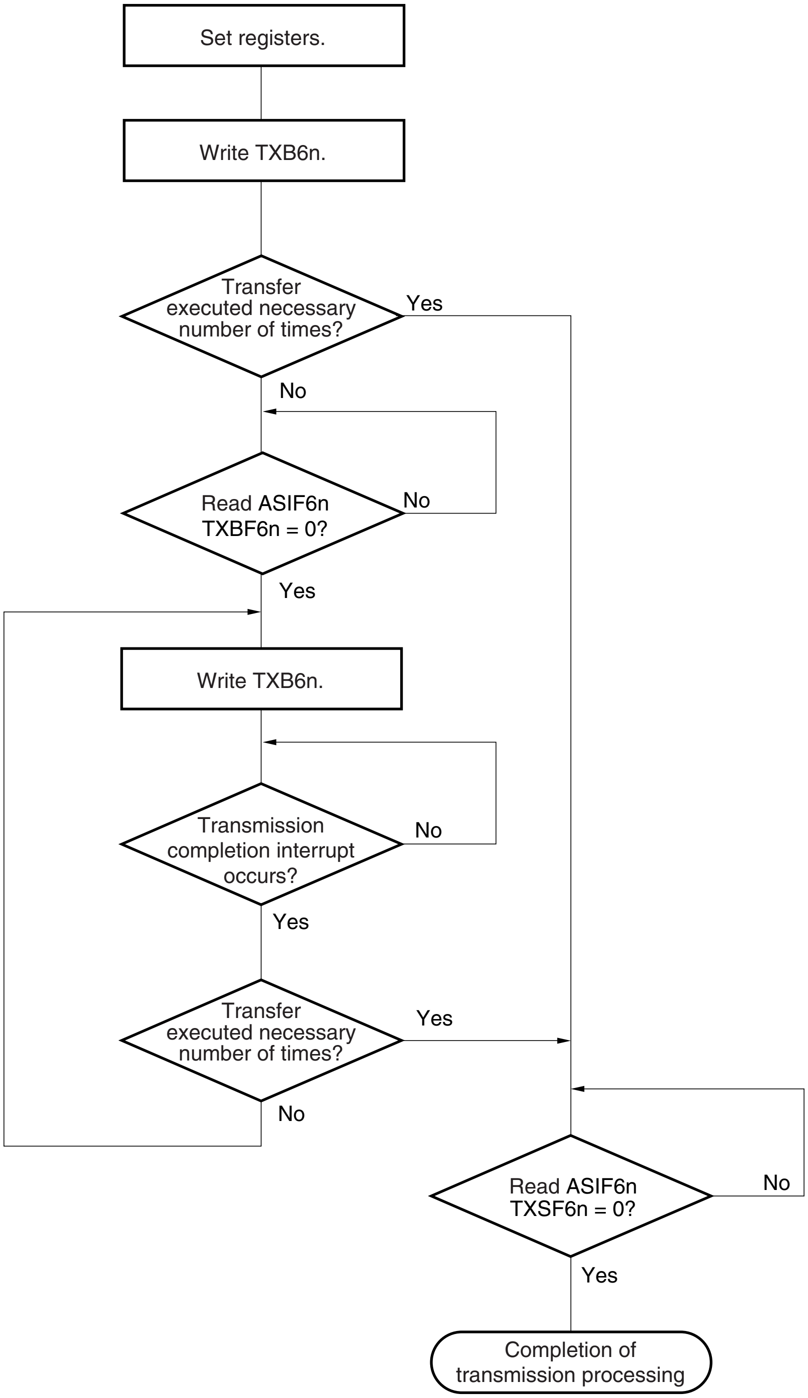
Cautions 1. To initialize the transmission unit upon completion of continuous transmission, be sure to check that the TXSF6n flag is “0” after generation of the transmission completion interrupt, and then execute initialization. If initialization is executed while the TXSF6n flag is “1”, the transmit data cannot be guaranteed.

2. During continuous transmission, an overrun error may occur, which means that the next transmission was completed before execution of INTST6n interrupt servicing after transmission of one data frame. An overrun error can be detected by developing a program that can count the number of transmit data and by referencing the TXSF6n flag.

Remark n = 0, 1

Figure 12-25 shows an example of the continuous transmission processing flow.

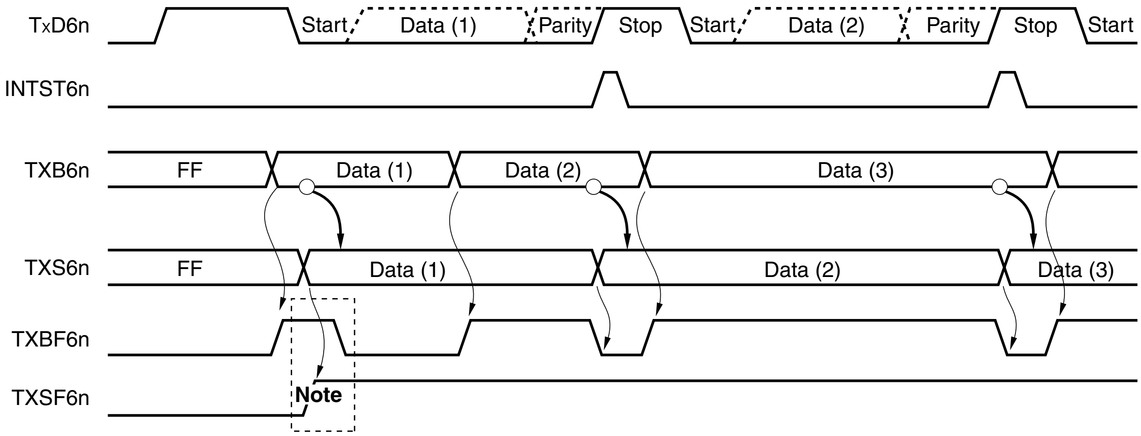
Figure 12-25. Example of Continuous Transmission Processing Flow


Set registers.
Write TXB6n.
Transfer executed necessary number of times?
Yes
No
Read ASIF6n TXBF6n = 0?
No
Yes
Write TXB6n.
Transmission completion interrupt occurs?
No
Yes
Transfer executed necessary number of times?
Yes
No
Read ASIF6n TXSF6n = 0?
No
Yes
Completion of transmission processing

Remark	TXB6n:	Transmit buffer register 6n
	ASIF6n:	Asynchronous serial interface transmission status register 6n
	TXBF6n:	Bit 1 of ASIF6n (transmit buffer data flag)
	TXSF6n:	Bit 0 of ASIF6n (transmit shift register data flag)
	n = 0, 1	

Figure 12-26 shows the timing of starting continuous transmission, and **Figure 12-27** shows the timing of ending continuous transmission.

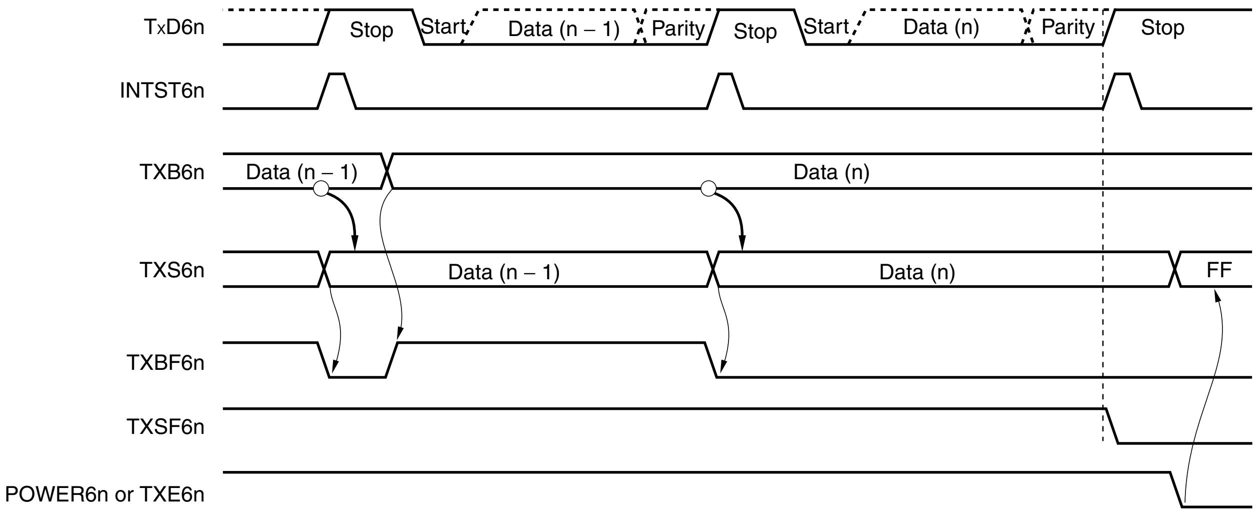
Figure 12-26. Timing of Starting Continuous Transmission



Note When ASIF6n is read, there is a period in which TXBF6n and TXSF6n = 1, 1. Therefore, judge whether writing is enabled using only the TXBF6n bit.

Remark		
	TxD6n:	TxD6n pins (output)
	INTST6n:	Interrupt request signal
	TXB6n:	Transmit buffer register 6n
	TXS6n:	Transmit shift register 6n
	ASIF6n:	Asynchronous serial interface transmission status register 6n
	TXBF6n:	Bit 1 of ASIF6n
	TXSF6n:	Bit 0 of ASIF6n
	n = 0, 1	

Figure 12-27. Timing of Ending Continuous Transmission



Remark	TxD6n:	TxD6n pins (output)
	INTST6n:	Interrupt request signal
	TXB6n:	Transmit buffer register 6n
	TXS6n:	Transmit shift register 6n
	ASIF6n:	Asynchronous serial interface transmission status register 6n
	TXBF6n:	Bit 1 of ASIF6n
	TXSF6n:	Bit 0 of ASIF6n
	POWER6n:	Bit 7 of asynchronous serial interface operation mode register (ASIM6n)
	TXE6n:	Bit 6 of asynchronous serial interface operation mode register (ASIM6n)
	n = 0, 1	

(e) Normal reception

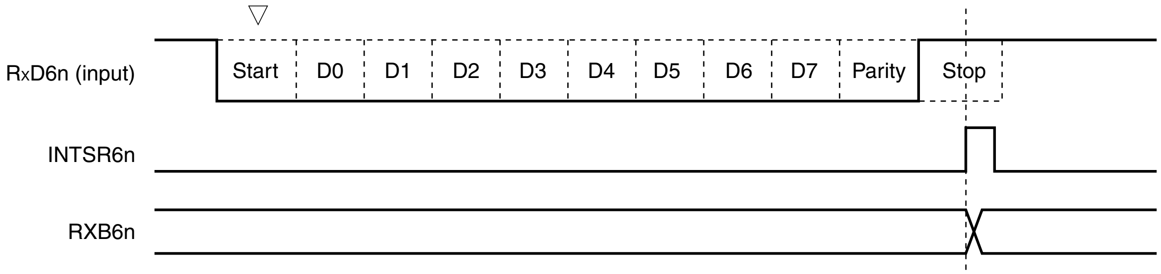
Reception is enabled and the RxD6n pins input is sampled when bit 7 (POWER6n) of asynchronous serial interface operation mode register 6n (ASIM6n) is set to 1 and then bit 5 (RXE6n) of ASIM6n is set to 1.

The 8-bit counter of the baud rate generator starts counting when the falling edge of the RxD6n pins input is detected. When the set value of baud rate generator control register 6n (BRGC6n) has been counted, the RxD6n pins input is sampled again (▽ in **Figure 12-28**). If the RxD6n pins are 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 6n (RXS6n) at the set baud rate. When the stop bit has been received, the reception completion interrupt (INTSR6n) is generated and the data of RXS6n is written to receive buffer register 6n (RXB6n). If an overrun error (OVE6n) occurs, however, the receive data is not written to RXB6n.

Even if a parity error (PE6n) occurs while reception is in progress, reception continues to the reception position of the stop bit, and a reception error interrupt (INTSR6n) is generated on completion of reception. <R>

Figure 12-28. Reception Completion Interrupt Request Timing



Cautions 1. If a reception error occurs, read ASIS6n and then RXB6n 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 6n (ASIS6n) before reading RXB6n.

Remark n = 0, 1

(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 6n (ASIS6n) is set as a result of data <R> reception, a reception error interrupt request (INTSR6n) is generated.

Which error has occurred during reception can be identified by reading the contents of ASIS6n in the reception error interrupt (INTSR6n) servicing (see **Figure 12-9. Format of Asynchronous Serial Interface Reception Error Status Register 60 (ASIS60)** and **Figure 12-10. Format of Asynchronous Serial Interface Reception Error Status Register 61 (ASIS61)**).

The contents of ASIS6n are cleared to 0 when ASIS6n is read.

Table 12-3. 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 6n (RXB6n).

<R>

Remark n = 0, 1

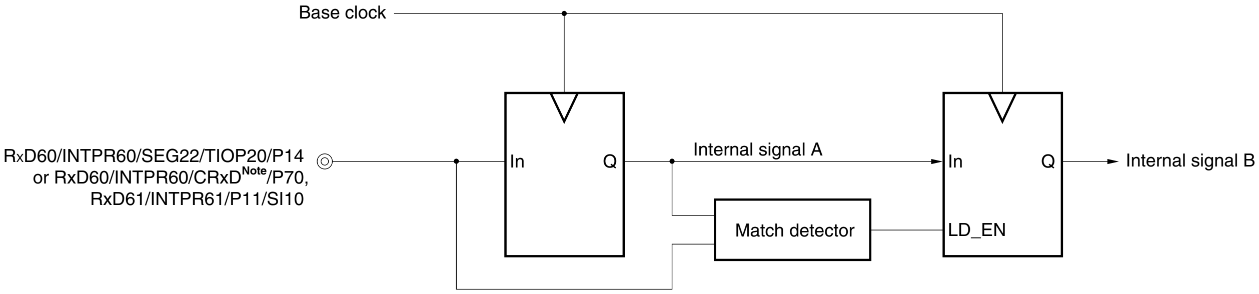
(g) Noise filter of receive data

The RXD6n (n = 0, 1) signal's 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 the following figure, the internal processing of the reception operation is delayed by two clocks from the external signal status.

Figure 12-29. Noise Filter Circuit



Note μPD78F0844, 78F0845, 78F0846, 78F0847, 78F0848, and 78F0849 only.

(h) SBF transmission

When the device is used in LIN communication operation, the SBF (Synch Break Field) transmission control function is used for transmission. For the transmission operation of LIN, see **Figure 12-1 LIN Transmission Operation**.

When bit 7 (POWER6n) of asynchronous serial interface operation mode register 6n (ASIM6n) is set to 1 and bit 6 (TXE6n) of ASIM6n is then set to 1, transmission is enabled.

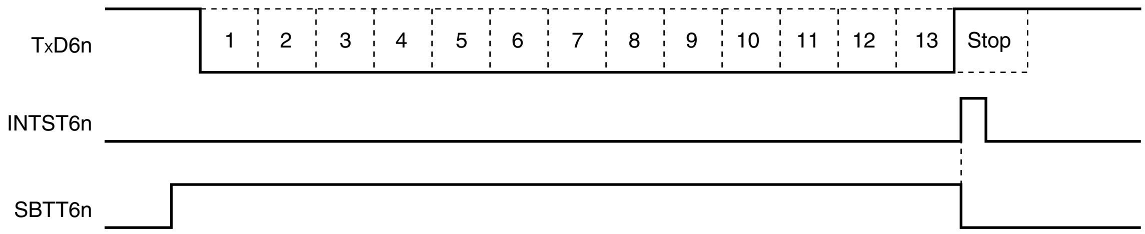
SBF transmission can be started by setting bit 5 (SBTT6n) of asynchronous serial interface control register 6n (ASICL6n) to 1.

Thereafter, a low level of bits 13 to 20 (set by bits 4 to 2 (SBL62n to SBL60n) of ASICL6n) is output. Following the end of SBF transmission, the transmission completion interrupt request (INTST6n) is generated and SBTT6n 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 6n (TXB6n), or until SBTT6n is set to 1.

Remark n = 0, 1

Figure 12-30. SBF Transmission



Remark TxD6n: TxD6n pins (output)
INTST6n: Transmission completion interrupt request
SBTT6n: Bit 5 of asynchronous serial interface control register 6n (ASICL6n)
n = 0, 1

(i) SBF reception

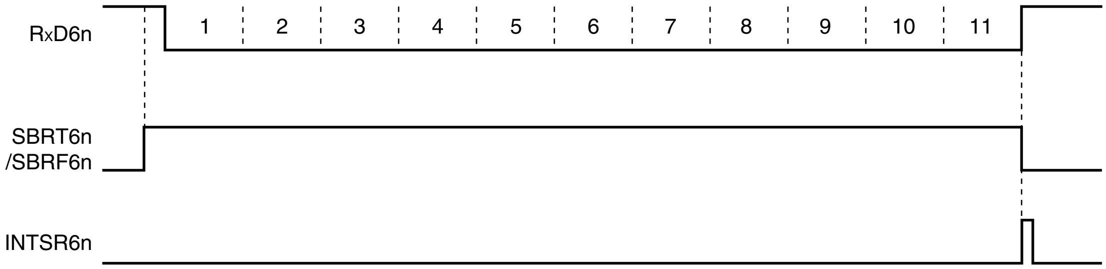
When the device is used in LIN communication operation, the SBF (Synch Break Field) reception control function is used for reception. For the reception operation of LIN, see **Figure 12-2 LIN Reception Operation**.

Reception is enabled when bit 7 (POWER6n) of asynchronous serial interface operation mode register 6n (ASIM6n) is set to 1 and then bit 5 (RXE6n) of ASIM6n is set to 1. SBF reception is enabled when bit 6 (SBRT6n) of asynchronous serial interface control register 6n (ASICL6n) is set to 1. In the SBF reception enabled status, the RxD6n pins are sampled and the start bit is detected in the same manner as the normal reception enable status.

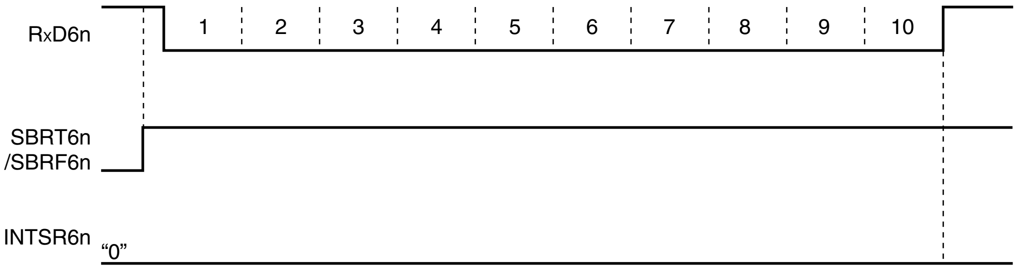
When the start bit has been detected, reception is started, and serial data is sequentially stored in the receive shift register 6n (RXS6n) at the set baud rate. When the stop bit is received and if the width of SBF is 11 bits or more, a reception completion interrupt request (INTSR6n) is generated as normal processing. At this time, the SBRF6n and SBRT6n bits are automatically cleared, and SBF reception ends. Detection of errors, such as OVE6n, PE6n, and FE6n (bits 0 to 2 of asynchronous serial interface reception error status register 6n (ASIS6n)) is suppressed, and error detection processing of UART communication is not performed. In addition, data transfer between receive shift register 6n (RXS6n) and receive buffer register 6n (RXB6n) 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 SBRF6n and SBRT6n bits are not cleared.

Figure 12-31. 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	RxD6n:	RxD6n pins (input)
	SBRT6n:	Bit 6 of asynchronous serial interface control register 6n (ASICL6n)
	SBRF6n:	Bit 7 of ASICL6n
	INTSR6n:	Reception completion interrupt request
	n = 0, 1	

12.4.3 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 UART60 and UART61.

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 (TPS63n to TPS60n) of clock selection register 6n (CKSR6n) is supplied to each module when bit 7 (POWER6n) of asynchronous serial interface operation mode register 6n (ASIM6n) 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 POWER6n = 0.

- Transmission counter

  This counter stops operation, cleared to 0, when bit 7 (POWER6n) or bit 6 (TXE6n) of asynchronous serial interface operation mode register 6n (ASIM6n) is 0.

  It starts counting when POWER6n = 1 and TXE6n = 1.

  The counter is cleared to 0 when the first data transmitted is written to transmit buffer register 6n (TXB6n).

  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 POWER6n or TXE6n is cleared to 0.

- Reception counter

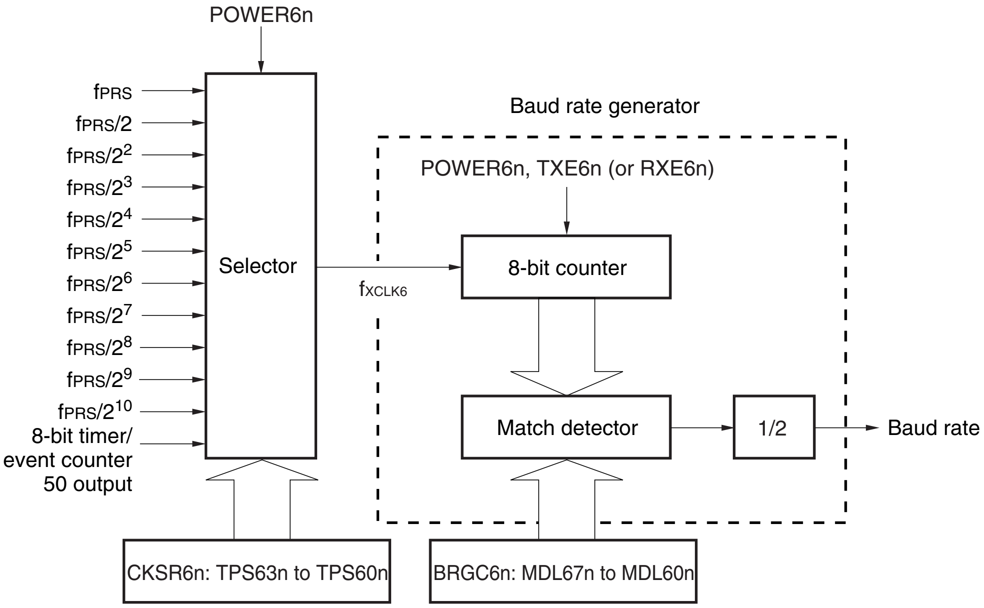
  This counter stops operation, cleared to 0, when bit 7 (POWER6n) or bit 5 (RXE6n) of asynchronous serial interface operation mode register 6n (ASIM6n) is 0.

  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.

Remark n = 0, 1

Figure 12-32. Configuration of Baud Rate Generator



Remark POWER6n: Bit 7 of asynchronous serial interface operation mode register 6n (ASIM6n)

TXE6n: Bit 6 of ASIM6n

RXE6n: Bit 5 of ASIM6n

CKSR6n: Clock selection register 6n

BRGC6n: Baud rate generator control register 6n

n = 0, 1

(2) Generation of serial clock

A serial clock can be generated by using clock selection register 6n (CKSR6n) and baud rate generator control register 6n (BRGC6n).

Select the clock to be input to the 8-bit counter by using bits 3 to 0 (TPS63n to TPS60n) of CKSR6n.

Bits 7 to 0 (MDL67n to MDL60n) of BRGC6n can be used to select the division value of the 8-bit counter.

(a) Baud rate

The baud rate can be calculated by the following expression.

- Baud rate = $\dfrac{f_{XCLK6}}{2 \times k}$ [bps]

f_{XCLK6}: Frequency of base clock selected by TPS63n to TPS60n bits of CKSR6n register

k: Value set by MDL67n to MDL60n bits of BRGC6n register (k = 4, 5, 6, ..., 255)

(b) Error of baud rate

The baud rate error can be calculated by the following expression.

- 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 MDL67n to MDL60n 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 12-4. Set Data of Baud Rate Generator

Baud Rate [bps]	f_{PRS} = 5.0 MHz				f_{PRS} = 10.0 MHz				f_{PRS} = 20.0 MHz			
	TPS63n, TPS60n	k	Calculated Value	ERR [%]	TPS63n, TPS60n	k	Calculated Value	ERR [%]	TPS63n, TPS60n	k	Calculated Value	ERR [%]
300	7H	65	301	0.16	8H	65	301	0.16	9H	65	301	0.16
600	6H	65	601	0.16	7H	65	601	0.16	8H	65	601	0.16
1200	5H	65	1202	0.16	6H	65	1202	0.16	7H	65	1202	0.16
2400	4H	65	2404	0.16	5H	65	2404	0.16	6H	65	2404	0.16
4800	3H	65	4808	0.16	4H	65	4808	0.16	5H	65	4808	0.16
9600	2H	65	9615	0.16	3H	65	9615	0.16	4H	65	9615	0.16
19200	1H	65	19231	0.16	2H	65	19231	0.16	3H	65	19231	0.16
24000	3H	13	24038	0.16	4H	13	24038	0.16	5H	13	24038	0.16
31250	4H	5	31250	0	5H	5	31250	0	6H	5	31250	0
38400	0H	65	38462	0.16	1H	65	38462	0.16	2H	65	38462	0.16
48000	2H	13	48077	0.16	3H	13	48077	0.16	4H	13	48077	0.16
76800	0H	33	75758	–1.36	0H	65	76923	0.16	1H	65	76923	0.16
115200	1H	11	113636	–1.36	0H	43	116279	0.94	0H	87	114943	–0.22
153600	1H	8	156250	1.73	0H	33	151515	–1.36	1H	33	151515	–1.36
312500	0H	8	312500	0	1H	8	312500	0	2H	8	312500	0
625000	0H	4	625000	0	1H	4	625000	0	2H	4	625000	0

Remark TPS63n to TPS60n: Bits 3 to 0 of clock selection register 6n (CKSR6n) (setting of base clock (f_{XCLK6}))

k: Value set by MDL67n to MDL60n bits of baud rate generator control register 6n (BRGC6n) (k = 4, 5, 6, ..., 255)

f_{PRS}: Peripheral hardware clock frequency

ERR: Baud rate error

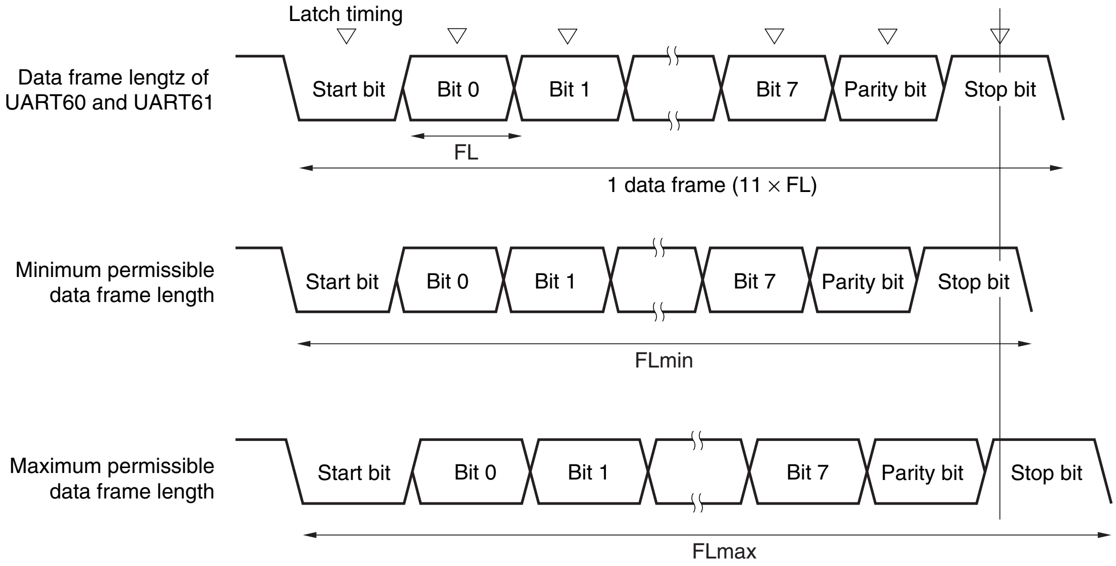
n = 0, 1

(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 12-33. Permissible Baud Rate Range During Reception



As shown in **Figure 12-33**, the latch timing of the receive data is determined by the counter set by baud rate generator control register 6n (BRGC6n) 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 UART60 and UART61

k: Set value of BRGC6n

FL: 1-bit data length

Margin of latch timing: 2 clocks

Remark n = 0, 1

Minimum permissible data frame length: $\text{FLmin} = 11 \times \text{FL} - \frac{k-2}{2k} \times \text{FL} = \frac{21k+2}{2k}\ \text{FL}$

Therefore, the maximum receivable baud rate at the transmission destination is as follows.

$$\text{BRmax} = (\text{FLmin}/11)^{-1} = \frac{22k}{21k+2}\ \text{Brate}$$

Similarly, the maximum permissible data frame length can be calculated as follows.

$$\frac{10}{11} \times \text{FLmax} = 11 \times \text{FL} - \frac{k+2}{2 \times k} \times \text{FL} = \frac{21k-2}{2 \times k}\ \text{FL}$$

$$\text{FLmax} = \frac{21k-2}{20k}\ \text{FL} \times 11$$

Therefore, the minimum receivable baud rate at the transmission destination is as follows.

$$\text{BRmin} = (\text{FLmax}/11)^{-1} = \frac{20k}{21k-2}\ \text{Brate}$$

The permissible baud rate error between UART60 and UART61 and the transmission destination can be calculated from the above minimum and maximum baud rate expressions, as follows.

Table 12-5. 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%

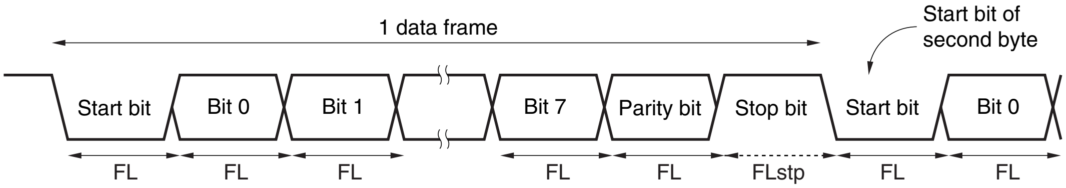
Remarks 1. The permissible error of reception depends on the number of bits in one frame, input clock frequency, and division ratio (k). The higher the input clock frequency and the higher the division ratio (k), the higher the permissible error.

2. k: Set value of BRGC6n (n = 0, 1)

(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 12-34. 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 13 SERIAL INTERFACES CSI10 AND CSI11

The 78K0/Dx2 incorporate serial interfaces CSI10 and CSI11.

13.1 Functions of Serial Interfaces CSI10 and CSI11

Caution Serial interface CSI11 is for 78K0/DF2 only.

Serial interfaces CSI10 and CSI11 have the following two modes.

- Operation stop mode
- 3-wire serial I/O mode

(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 **13.4.1 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 ($\overline{\text{SCK1n}}$) and two serial data lines (SI1n and SO1n).

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 **13.4.2 3-wire serial I/O mode**.

Remark n = 0, 1

13.2 Configuration of Serial Interfaces CSI10 and CSI11

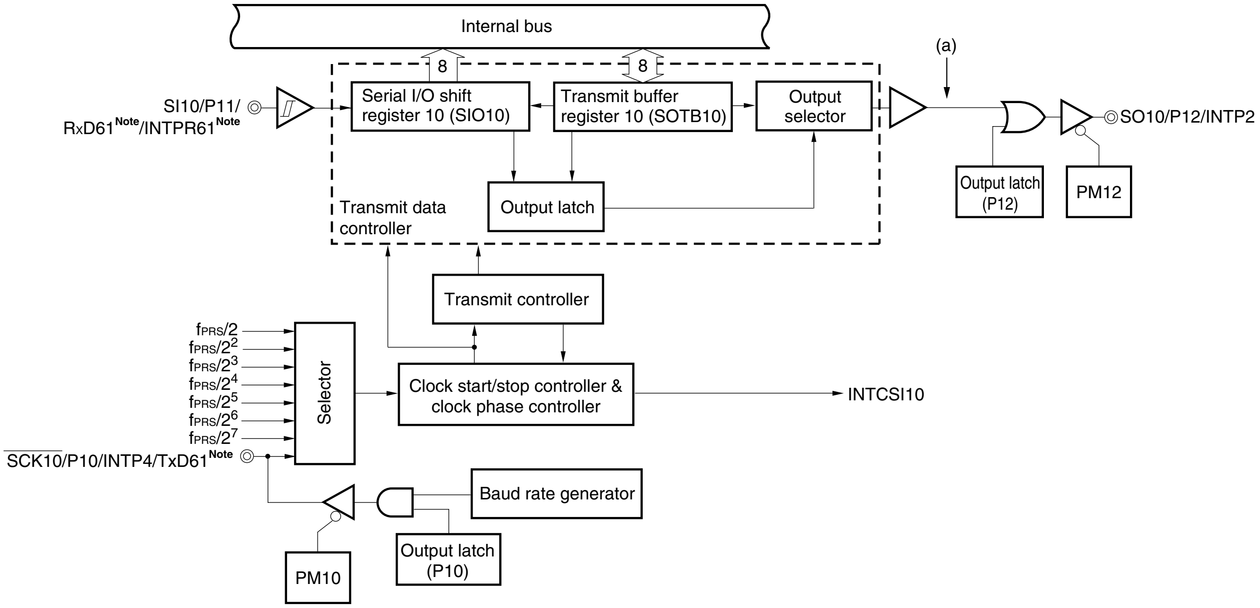
Serial interfaces CSI10 and CSI11 include the following hardware.

Table 13-1. Configuration of Serial Interfaces CSI10 and CSI11

Item	Configuration	
	78K0/DE2	78K0/DF2
Controller	Transmit controller Clock start/stop controller & clock phase controller	
Registers	Transmit buffer register 1n (SOTB1n) Serial I/O shift register 1n (SIO1n)	
Control registers	Serial operation mode register 1n (CSIM1n) Serial clock selection register 1n (CSIC1n) Port mode register 1 (PM1) Port register 1 (P1)	
	–	Port mode register 7 (PM7) Port register 7 (P7)

Remark n = 0, 1 (the registers given with n = 1 are for 78K0/DF2 only).

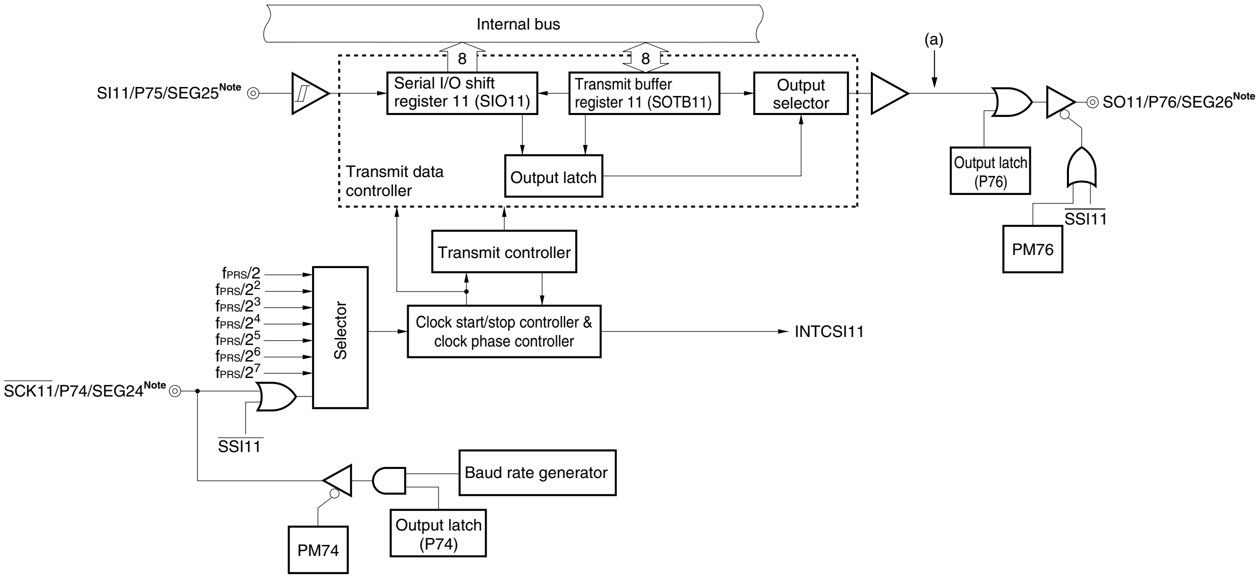
Figure 13-1. Block Diagram of Serial Interface CSI10



Note 78K0/DF2 only.

Remark (a): SO10 output

Figure 13-2. Block Diagram of Serial Interface CSI11



Note μ PD78F0842, 78F0843, 78F0848, and 78F0849 only.

Remark (a): SO11 output

(1) Transmit buffer register 1n (SOTB1n)

This register sets the transmit data.

Transmission/reception is started by writing data to SOTB1n when bit 7 (CSIE1n) and bit 6 (TRMD1n) of serial operation mode register 1n (CSIM1n) is 1.

The data written to SOTB1n is converted from parallel data into serial data by serial I/O shift register 1n, and output to the serial output pin (SO1n).

SOTB1n can be written or read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Cautions 1. Do not access SOTB1n when CSOT1n = 1 (during serial communication).

2. In the slave mode, transmission/reception is started when data is written to SOTB11 with a low level input to the $\overline{\text{SSI11}}$ pin. For details of the transmission/reception operation, see 13.4.2 (2) Communication operation.

(2) Serial I/O shift register 1n (SIO1n)

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 SIO1n if bit 6 (TRMD1n) of serial operation mode register 1n (CSIM1n) is 0.

During reception, the data is read from the serial input pin (SI1n) to SIO1n.

Reset signal generation clears this register to 00H.

Cautions 1. Do not access SIO1n when CSOT1n = 1 (during serial communication).

2. In the slave mode, reception is started when data is read from SIO11 with a low level input to the $\overline{\text{SSI11}}$ pin. For details of the reception operation, see 13.4.2 (2) Communication operation.

Remark n = 0, 1

13.3 Registers Controlling Serial Interfaces CSI10 and CSI11

Serial interfaces CSI10 and CSI11 are controlled by the following four registers.

- Serial operation mode register 1n (CSIM1n)
- Serial clock selection register 1n (CSIC1n)
- Port mode registers 1, 7 (PM1, PM7)
- Port registers 1, 7 (P1, P7)

Remark CSIM11, CSIC11, PM7, and P7 are for 78K0/DF2 only.

(1) Serial operation mode register 1n (CSIM1n)

CSIM1n is used to select the operation mode and enable or disable operation.

CSIM1n can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Remark n = 0, 1

Figure 13-3. 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	Operation control in 3-wire serial I/O mode
0	Disables operation[Note 2] 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 P10/$\overline{\text{SCK10}}$/INTP4/TxD61, and P12/SO10/INTP2 as general-purpose ports, set CSIM10 in the default status (00H). (The TxD61 function is for 78K0/DF2 only.)

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 **(a)** in **Figure 13-1. Block Diagram of Serial Interface CSI10**) 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).

Caution Be sure to clear bit 5 to 0.

Figure 13-4. Format of Serial Operation Mode Register 11 (CSIM11)

Address: FF88H After reset: 00H R/W[Note 1]

Symbol	<7>	6	5	4	3	2	1	0
CSIM11	CSIE11	TRMD11	SSE11	DIR11	0	0	0	CSOT11

CSIE11	Operation control in 3-wire serial I/O mode
0	Disables operation[Note 2] and asynchronously resets the internal circuit[Note 3].
1	Enables operation

TRMD11[Note 4]	Transmit/receive mode control
0[Note 5]	Receive mode (transmission disabled).
1	Transmit/receive mode

SSE11[Notes 6, 7]	$\overline{\text{SSI11}}$ pin use selection
0	$\overline{\text{SSI11}}$ pin is not used
1	$\overline{\text{SSI11}}$ pin is used

DIR11[Note 8]	First bit specification
0	MSB
1	LSB

CSOT11	Communication status flag
0	Communication is stopped.
1	Communication is in progress.

Notes 1. Bit 0 is a read-only bit.

2. To use P74/$\overline{\text{SCK11}}$/SEG24, P76/SO11/SEG26, and P77/$\overline{\text{SSI11}}$/TIOP20/SEG27 as general-purpose ports, set CSIM11 in the default status (00H). (The SEG24, SEG26, and SEG27 functions are for μPD78F0842, 78F0843, 78F0848, and 78F0849 only.)

3. Bit 0 (CSOT11) of CSIM11 and serial I/O shift register 11 (SIO11) are reset.

4. Do not rewrite TRMD11 when CSOT11 = 1 (during serial communication).

5. The SO11 output (see **(a)** in **Figure 13-2. Block Diagram of Serial Interface CSI11**) is fixed to the low level when TRMD11 is 0. Reception is started when data is read from SIO11.

6. Do not rewrite SSE11 when CSOT11 = 1 (during serial communication).

7. Before setting this bit to 1, fix the $\overline{\text{SSI11}}$ pin input level to 0 or 1.

8. Do not rewrite DIR11 when CSOT11 = 1 (during serial communication).

(2) Serial clock selection register 1n (CSIC1n)

This register specifies the timing of the data transmission/reception and sets the serial clock.

CSIC1n can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Remark n = 0, 1

Figure 13-5. Format of Serial Clock Selection Register 10 (CSIC10) (1/2)

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

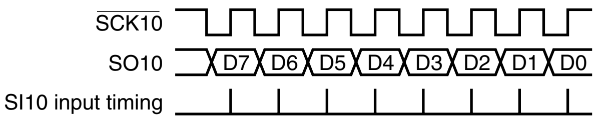
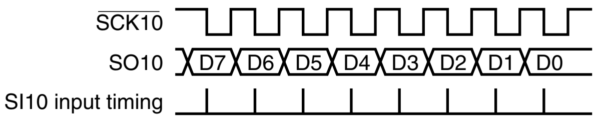
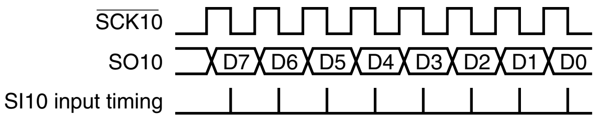
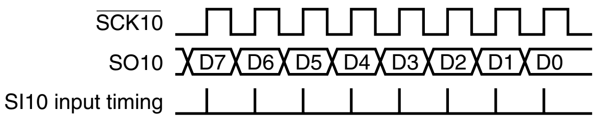
CKP10	DAP10	Specification of data transmission/reception timing	Type
0	0	 $\overline{\text{SCK10}}$ SO10 SI10 input timing D7 D6 D5 D4 D3 D2 D1 D0	1
0	1	 $\overline{\text{SCK10}}$ SO10 SI10 input timing D7 D6 D5 D4 D3 D2 D1 D0	2
1	0	 $\overline{\text{SCK10}}$ SO10 SI10 input timing D7 D6 D5 D4 D3 D2 D1 D0	3
1	1	 $\overline{\text{SCK10}}$ SO10 SI10 input timing D7 D6 D5 D4 D3 D2 D1 D0	4

Figure 13-5. Format of Serial Clock Selection Register 10 (CSIC10) (2/2)

CKS102	CKS101	CKS100	CSI10 serial clock selection[Notes 1, 2]					Mode
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz	
0	0	0	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	Master mode
0	0	1	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	
0	1	0	$f_{PRS}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	
0	1	1	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz	
1	0	0	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	312.5 kHz	625 kHz	
1	0	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz	
1	1	0	$f_{PRS}/2^7$	15.63 kHz	39.06 kHz	78.13 kHz	156.25 kHz	
1	1	1	External clock input to $\overline{\text{SCK10}}$					Slave mode

Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{IN}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 4.0 to 5.5 V: $f_{PRS} \leq$ 20 MHz
- V_{DD} = 2.7 to 4.0 V: $f_{PRS} \leq$ 10 MHz

2. Set the serial clock to satisfy the following conditions.

Supply Voltage	(A) Grade Products	(A2) Grade Products
V_{DD} = 4.0 to 5.5 V	Serial clock ≤ 5 MHz	Serial clock ≤ 5 MHz
V_{DD} = 2.7 to 4.0 V	Serial clock ≤ 2.5 MHz	Serial clock ≤ 2.5 MHz

Cautions 1. Do not write to CSIC10 while CSIE10 = 1 (operation enabled).

2. To use P10/$\overline{\text{SCK10}}$/INTP4/TxD61 and P12/SO10/INTP2 as general-purpose ports, set CSIC10 in the default status (00H). (The TxD61 function is for 78K0/DF2 only.)

3. The phase type of the data clock is type 1 after reset.

Remark f_{PRS}: Peripheral hardware clock frequency

Figure 13-6. Format of Serial Clock Selection Register 11 (CSIC11) (1/2)

Address: FF89H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIC11	0	0	0	CKP11	DAP11	CKS112	CKS111	CKS110

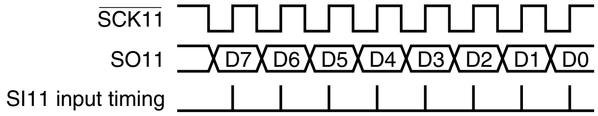
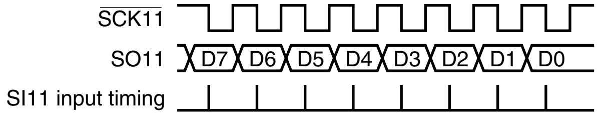
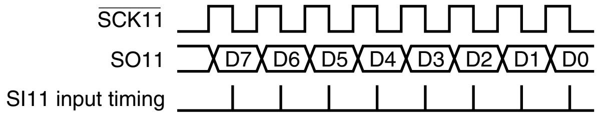
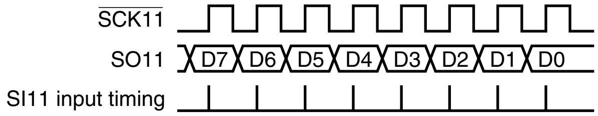
CKP11	DAP11	Specification of data transmission/reception timing	Type
0	0	 $\overline{\text{SCK11}}$ SO11 D7 D6 D5 D4 D3 D2 D1 D0 SI11 input timing	1
0	1	 $\overline{\text{SCK11}}$ SO11 D7 D6 D5 D4 D3 D2 D1 D0 SI11 input timing	2
1	0	 $\overline{\text{SCK11}}$ SO11 D7 D6 D5 D4 D3 D2 D1 D0 SI11 input timing	3
1	1	 $\overline{\text{SCK11}}$ SO11 D7 D6 D5 D4 D3 D2 D1 D0 SI11 input timing	4

Figure 13-6. Format of Serial Clock Selection Register 11 (CSIC11) (1/2)

CKS112	CKS111	CKS110	CSI11 serial clock selection[Note 1, 2]					Mode
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz	f_{PRS} = 20 MHz	
0	0	0	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	Master mode
0	0	1	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	
0	1	0	$f_{PRS}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	
0	1	1	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz	
1	0	0	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	312.5 kHz	625 kHz	
1	0	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz	
1	1	0	$f_{PRS}/2^7$	15.63 kHz	39.06 kHz	78.13 kHz	156.25 kHz	
1	1	1	External clock input to SCK11					Slave mode

Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{IN}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 4.0 to 5.5 V: $f_{PRS} \leq$ 20 MHz
- V_{DD} = 2.7 to 4.0 V: $f_{PRS} \leq$ 10 MHz

2. Set the serial clock to satisfy the following conditions.

Supply Voltage	(A) Grade Products	(A2) Grade Products
V_{DD} = 4.0 to 5.5 V	Serial clock ≤ 5 MHz	Serial clock ≤ 5 MHz
V_{DD} = 2.7 to 4.0 V	Serial clock ≤ 2.5 MHz	Serial clock ≤ 2.5 MHz

Cautions 1. Do not write to CSIC11 while CSIE11 = 1 (operation enabled).

2. To use P74/$\overline{\text{SCK11}}$/SEG24 and P76/SO11/SEG26 as general-purpose ports, set CSIC11 in the default status (00H). (The SEG24 and SEG26 functions are for μ PD78F0842, 78F0843, 78F0848, and 78F0849 only.)

3. The phase type of the data clock is type 1 after reset.

Remark f_{PRS}: Peripheral hardware clock frequency

(3) Port mode registers 1, 7 (PM1, PM7)

These registers set port 1 and 7 input/output in 1-bit units.

When using P10/$\overline{\text{SCK10}}$/INTP4/TxD61[Note 1] and P74/$\overline{\text{SCK11}}$/SEG24[Note 2] as the clock output pins of the serial interface, clear PM10 and PM76, and the output latches of P10 and P74 to 1.

When using P12/SO10/INTP2 and P76/SO11/SEG26[Note 2] as the data output pins of the serial interface, clear PM12, PM76, P12 and P76 to 0.

When using P10/$\overline{\text{SCK10}}$/INTP4/TxD61[Note 1] and P74/$\overline{\text{SCK11}}$/SEG24[Note 2] as the clock input pins of the serial interface, P11/SI10/RxD61[Note 1]/INTPR61[Note 1] and P75/SI11/SEG25[Note 2] as the data input pins, and P77/$\overline{\text{SSI11}}$/TIOP20/SEG27[Note 2] as the chip select input pin, set PM10, PM74, PM11, PM75, and PM77 to 1. At this time, the output latches of P10, P74, P11, P75, and P77 may be 0 or 1.

PM1 and PM7 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Notes 1. 78K0/DF2 only.

2. μPD78F0842, 78F0843, 78F0848, and 78F0849 only.

Figure 13-7. 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)

Figure 13-8. Format of Port Mode Register 7 (PM7)

Address: FF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	PM77[Note]	PM76[Note]	PM75[Note]	PM74[Note]	PM73	PM72	PM71	PM70

PM7n	P7n pin I/O mode selection (n = 0 to 7)
0	Output mode (Output buffer on)
1	Input mode (Output buffer off)

Note These bits are set to 1 in 78K0/DE2.

13.4 Operation of Serial Interfaces CSI10 and CSI11

Serial interfaces CSI10 and CSI11 can be used in the following two modes.

- Operation stop mode
- 3-wire serial I/O mode

Remark Serial interface CSI11 is for 78K0/DF2 only.

13.4.1 Operation stop mode

Serial communication is not executed in this mode. Therefore, the power consumption can be reduced. In addition, the P10/SCK10/INTP4/TxD61, P11/SI10/RxD61/INTPR61, P12/SO10/INTP2, P74/SCK11/SEG24, P75/SI11/SEG25, and P76/SO11/SEG26 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 1n (CSIM1n).

To set the operation stop mode, clear bit 7 (CSIE1n) of CSIM1n to 0.

(a) Serial operation mode register 1n (CSIM1n)

CSIM1n can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears CSIM1n to 00H.

Remark n = 0, 1

- Serial operation mode register 10 (CSIM10)

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 P10/SCK10/INTP4/TxD61 and P12/SO10/INTP2 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.

- Serial operation mode register 11 (CSIM11)

Address: FF88H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CSIM11	CSIE11	TRMD11	SSE11	DIR11	0	0	0	CSOT11

CSIE11	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 P74/SCK11/SEG24, P77/SSI11/TIOP20/SEG27, and P76/SO11/SEG26 as general-purpose ports, set CSIM11 in the default status (00H).

2. Bit 0 (CSOT11) of CSIM11 and serial I/O shift register 11 (SIO11) are reset.

Remarks 1. The TxD61, RxD61, and INTPR61 functions are for 78K0/DF2 only.

2. The SEG24 to SEG27 functions are for μPD78F0842, 78F0843, 78F0848, and 78F0849 only.

13.4.2 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 ($\overline{\text{SCK1n}}$), serial output (SO1n), and serial input (SI1n) lines.

(1) Registers used

- Serial operation mode register 1n (CSIM1n)
- Serial clock selection register 1n (CSIC1n)
- Port mode register 1 (PM1) or port mode register 7 (PM7)
- Port register 1 (P1) or port register 7 (P7)

The basic procedure of setting an operation in the 3-wire serial I/O mode is as follows.

<1> Set the CSIC1n register (see **Figures 13-5** and **13-6**).

<2> Set bits 0 and 4 to 6 (CSOT1n, DIR1n, SSE11 (serial interface CSI11 only), and TRMD1n) of the CSIM1n register (see **Figures 13-3** and **13-4**).

<3> Set bit 7 (CSIE1n) of the CSIM1n register to 1. → Transmission/reception is enabled.

<4> Write data to transmit buffer register 1n (SOTB1n). → Data transmission/reception is started.
Read data from serial I/O shift register 1n (SIO1n). → Data reception is started.

Caution Take relationship with the other party of communication when setting the port mode register and port register.

Remark n = 0, 1

The relationship between the register settings and pins is shown below.

Table 13-2. Relationship between Register Settings and Pins (1/2)

(a) Serial interface CSI10

CSIE10	TRMD10	PM11	P11	PM12	P12	PM10	P10	CSI10 Operation	Pin Function		
									SI10/RxD61/ INTPR61/ P11	SO10/ INTP2/ P12	$\overline{\text{SCK10}}$/ INTP4/ TxD61/P10
0	×	× Note 1	× Note 1	× Note 1	× Note 1	× Note 1	× Note 1	Stop	RxD61/ INTPR61/ P11	INTP2/P12	TxD61/ INTP4/ P10[Note 2]
1	0	1	×	× Note 1	× Note 1	1	×	Slave reception[Note 3]	SI10	INTP2/P12	$\overline{\text{SCK10}}$ (input)[Note 3]
1	1	× Note 1	× Note 1	0	0	1	×	Slave transmission[Note 3]	RxD61/ INTPR61/ P11	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	INTP2/P12	$\overline{\text{SCK10}}$ (output)
1	1	× Note 1	× Note 1	0	0	0	1	Master transmission	RxD61/ INTPR61/ P11	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 P10/$\overline{\text{SCK10}}$/INTP4/TxD61 as port pins, clear CKP10 to 0.

3. To use the slave mode, set CKS102, CKS101, and CKS100 to 1, 1, 1.

Remarks 1.

×:	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. The RxD61, INTPR61, and TxD61 functions are for 78K0/DF2 only.

Table 13-2. Relationship between Register Settings and Pins (2/2)

(b) Serial interface CSI11

CSIE11	TRMD11	SSE11	PM75	P75	PM76	P76	PM74	P74	PM77	P77	CSI11 Operation	Pin Function			
												SI11/ SEG25/ P75	SO11/ SEG26/ P76	$\overline{\text{SCK11}}$/ SEG24/ P74	$\overline{\text{SSI11}}$/ TIOP20/ SEG27/ P77
0	×	×	× Note 1	× Note 1	× Note 1	× Note 1	× Note 1	× Note 1	× Note 1	× Note 1	Stop	SEG25 /P75	SEG26 /P76	SEG24/ P74 Note 2	TIOP20 /SEG27 /P77
1	0	0	1	×	× Note 1	× Note 1	1	×	× Note 1	× Note 1	Slave reception Note 3	SI11	SEG26 /P76	$\overline{\text{SCK11}}$ (input) Note 3	TIOP20 /SEG27 /P77
		1							1	×					$\overline{\text{SSI11}}$
1	1	0	× Note 1	× Note 1	0	0	1	×	× Note 1	× Note 1	Slave transmission Note 3	SEG25 /P75	SO11	$\overline{\text{SCK11}}$ (input) Note 3	TIOP20 /SEG27 /P77
		1							1	×					$\overline{\text{SSI11}}$
1	1	0	1	×	0	0	1	×	× Note 1	× Note 1	Slave transmission /reception Note 3	SI11	SO11	$\overline{\text{SCK11}}$ (input) Note 3	TIOP20 /SEG27 /P77
		1							1	×					$\overline{\text{SSI11}}$
1	0	0	1	×	× Note 1	× Note 1	0	1	× Note 1	× Note 1	Master reception	SI11	SEG26 /P76	$\overline{\text{SCK11}}$ (output)	TIOP20 /SEG27 /P77
1	1	0	× Note 1	× Note 1	0	0	0	1	× Note 1	× Note 1	Master transmission	SEG25 /P75	SO11	$\overline{\text{SCK11}}$ (output)	TIOP20 /SEG27 /P77
1	1	0	1	×	0	0	0	1	× Note 1	× Note 1	Master transmission /reception	SI11	SO11	$\overline{\text{SCK11}}$ (output)	TIOP20 /SEG27 /P77

Notes 1. Can be set as port function.

2. To use P74/$\overline{\text{SCK11}}$/SEG24 as port pins, clear CKP11 to 0.

3. To use the slave mode, set CKS112, CKS111, and CKS110 to 1, 1, 1.

Remarks 1.

×:	don't care
CSIE11:	Bit 7 of serial operation mode register 11 (CSIM11)
TRMD11:	Bit 6 of CSIM11
CKP11:	Bit 4 of serial clock selection register 11 (CSIC11)
CKS112, CKS111, CKS110:	Bits 2 to 0 of CSIC11
PM7×:	Port mode register 7×
P7×:	Port 7× output latch

2. The SEG24 to SEG27 functions are for μ PD78F0842, 78F0843, 78F0848, and 78F0849 only.

(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 (TRMD1n) of serial operation mode register 1n (CSIM1n) is 1. Transmission/reception is started when a value is written to transmit buffer register 1n (SOTB1n). In addition, data can be received when bit 6 (TRMD1n) of serial operation mode register 1n (CSIM1n) is 0.

Reception is started when data is read from serial I/O shift register 1n (SIO1n).

However, communication is performed as follows if bit 5 (SSE11) of CSIM11 is 1 when serial interface CSI11 is in the slave mode.

<1> Low level input to the $\overline{\text{SSI11}}$ pin
→ Transmission/reception is started when SOTB11 is written, or reception is started when SIO11 is read.

<2> High level input to the $\overline{\text{SSI11}}$ pin
→ Transmission/reception or reception is held, therefore, even if SOTB11 is written or SIO11 is read, transmission/reception or reception will not be started.

<3> Data is written to SOTB11 or data is read from SIO11 while a high level is input to the $\overline{\text{SSI11}}$ pin, then a low level is input to the $\overline{\text{SSI11}}$ pin
→ Transmission/reception or reception is started.

<4> A high level is input to the $\overline{\text{SSI11}}$ pin during transmission/reception or reception
→ Transmission/reception or reception is suspended.

After communication has been started, bit 0 (CSOT1n) of CSIM1n is set to 1. When communication of 8-bit data has been completed, a communication completion interrupt request flag (CSIIF1n) is set, and CSOT1n is cleared to 0. Then the next communication is enabled.

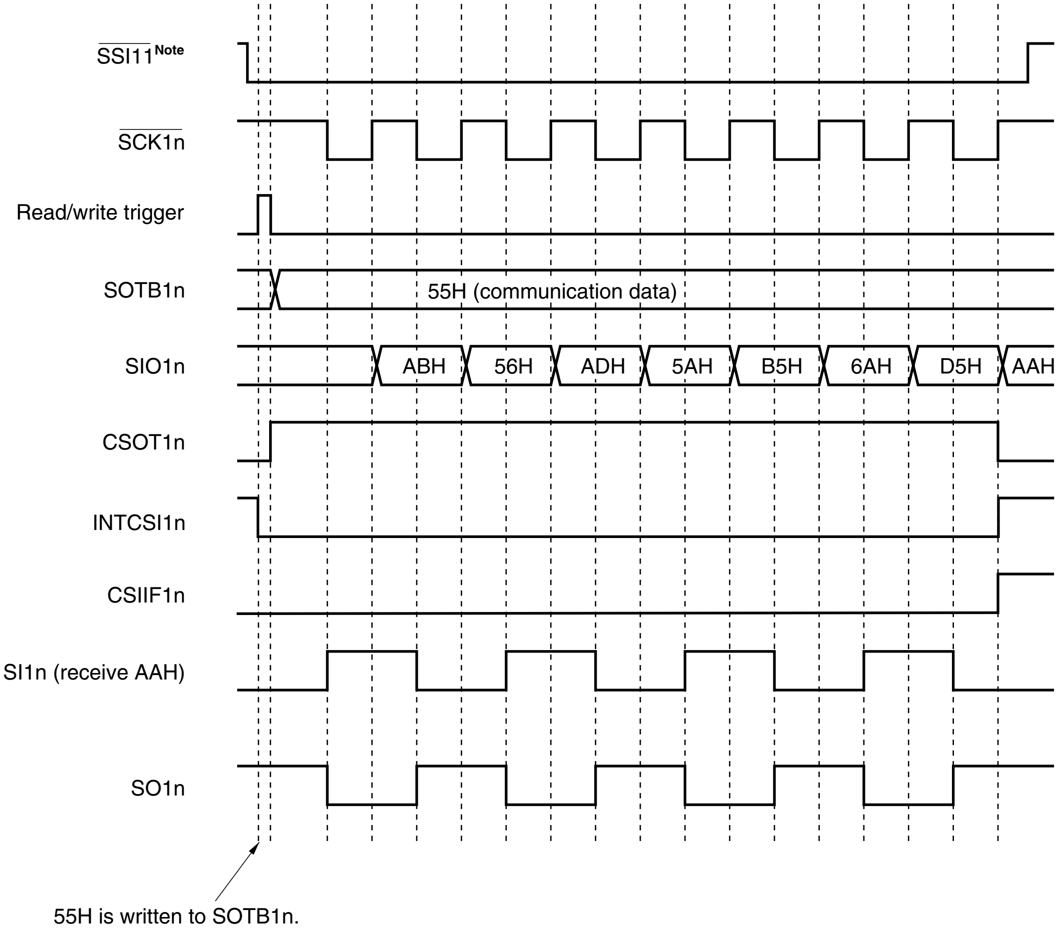
Cautions 1. Do not access the control register and data register when CSOT1n = 1 (during serial communication).

2. When using serial interface CSI11, wait for the duration of at least one clock before the clock operation is started to change the level of the $\overline{\text{SSI11}}$ pin in the slave mode; otherwise, malfunctioning may occur.

Remark n = 0, 1

Figure 13-9. Timing in 3-wire Serial I/O Mode (1/2)

(1) Transmission/reception timing (Type 1; TRMD1n = 1, DIR1n = 0, CKP1n = 0, DAP1n = 0, SSE11 = 1[Note])

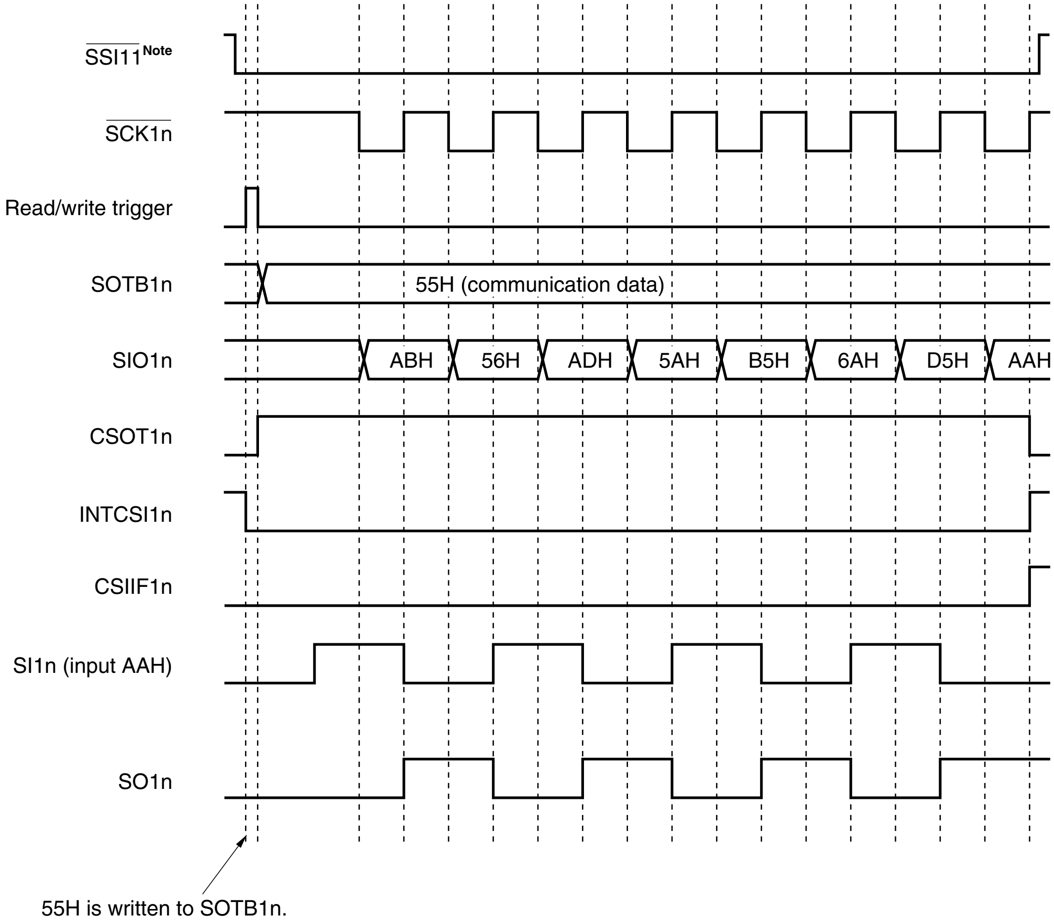


Note The SSE11 flag and $\overline{\text{SSI11}}$ pin are available only for serial interface CSI11, and are used in the slave mode.

Remark n = 0, 1

Figure 13-9. Timing in 3-wire Serial I/O Mode (2/2)

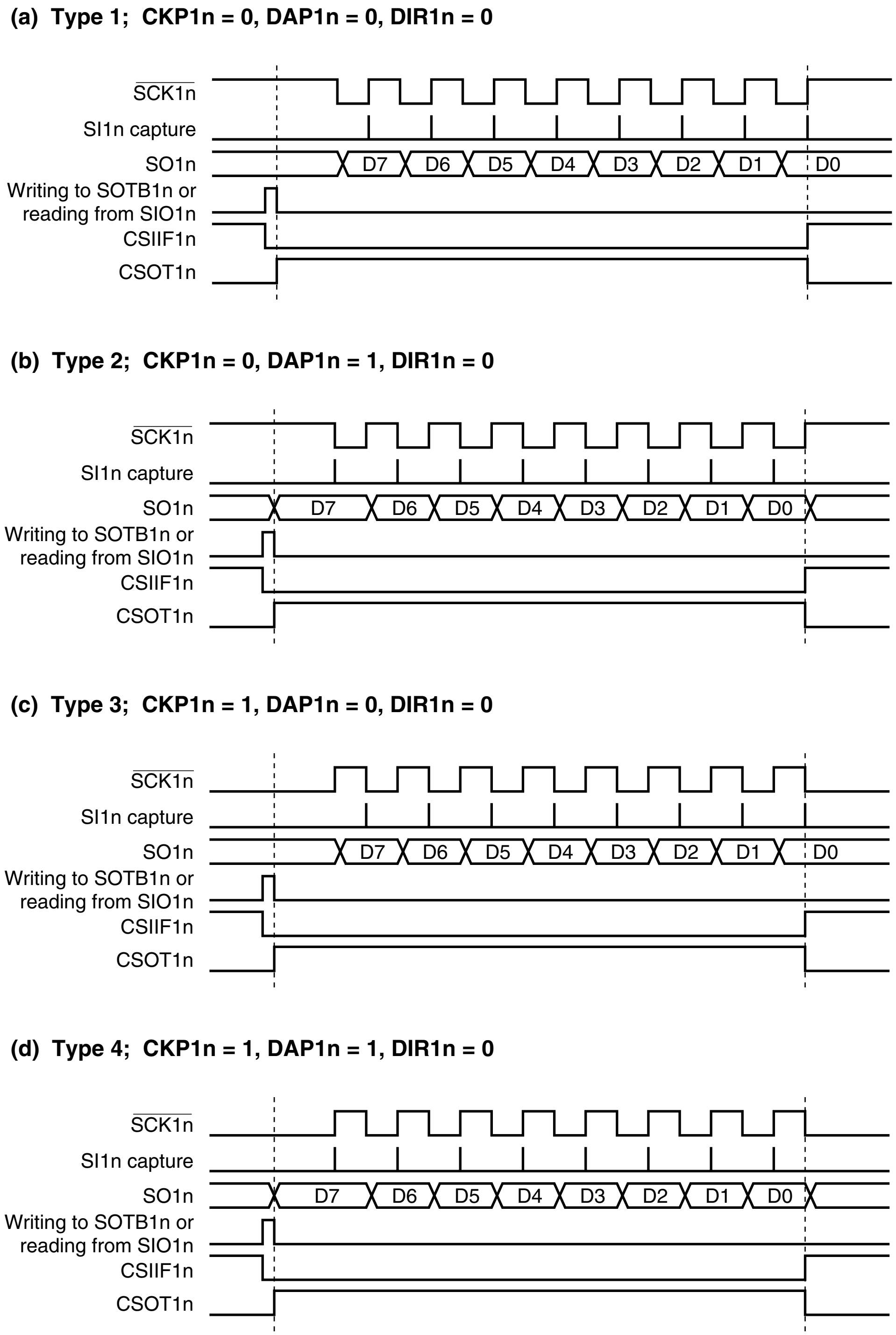
(2) Transmission/reception timing (Type 2; TRMD1n = 1, DIR1n = 0, CKP1n = 0, DAP1n = 1, SSE11 = 1[Note]**)**



Note The SSE11 flag and $\overline{\text{SSI11}}$ pin are available only for serial interface CSI11, and are used in the slave mode.

Remark n = 0, 1

Figure 13-10. Timing of Clock/data Phase



(a) Type 1; CKP1n = 0, DAP1n = 0, DIR1n = 0

(b) Type 2; CKP1n = 0, DAP1n = 1, DIR1n = 0

(c) Type 3; CKP1n = 1, DAP1n = 0, DIR1n = 0

(d) Type 4; CKP1n = 1, DAP1n = 1, DIR1n = 0

Remarks 1. The above figure illustrates a communication operation where data is transmitted with the MSB first.

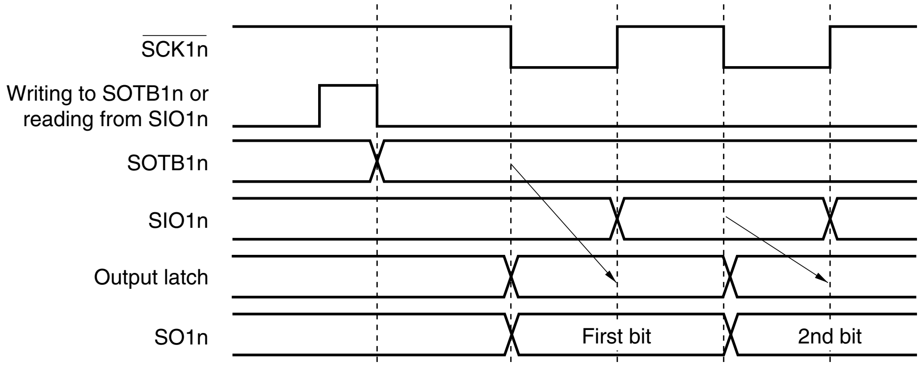
2. n = 0, 1

(3) Timing of output to SO1n pin (first bit)

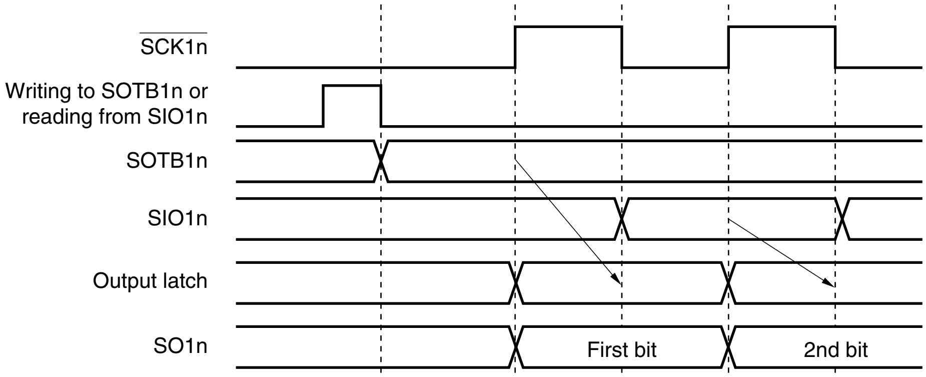
When communication is started, the value of transmit buffer register 1n (SOTB1n) is output from the SO1n pin. The output operation of the first bit at this time is described below.

Figure 13-11. Output Operation of First Bit (1/2)

(a) Type 1: CKP1n = 0, DAP1n = 0


$\overline{\text{SCK1n}}$
Writing to SOTB1n or reading from SIO1n
SOTB1n
SIO1n
Output latch
SO1n
First bit
2nd bit

(b) Type 3: CKP1n = 1, DAP1n = 0


$\overline{\text{SCK1n}}$
Writing to SOTB1n or reading from SIO1n
SOTB1n
SIO1n
Output latch
SO1n
First bit
2nd bit

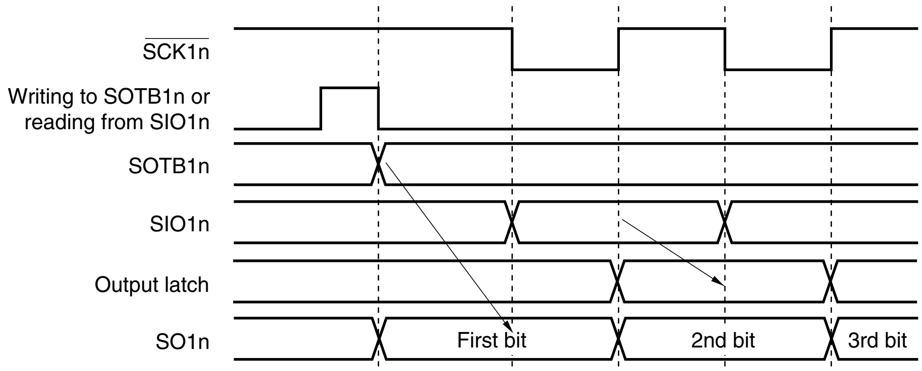
The first bit is directly latched by the SOTB1n register to the output latch at the falling (or rising) edge of $\overline{\text{SCK1n}}$, and output from the SO1n pin via an output selector. Then, the value of the SOTB1n register is transferred to the SIO1n register at the next rising (or falling) edge of $\overline{\text{SCK1n}}$, and shifted one bit. At the same time, the first bit of the receive data is stored in the SIO1n register via the SI1n pin.

The second and subsequent bits are latched by the SIO1n register to the output latch at the next falling (or rising) edge of $\overline{\text{SCK1n}}$, and the data is output from the SO1n pin.

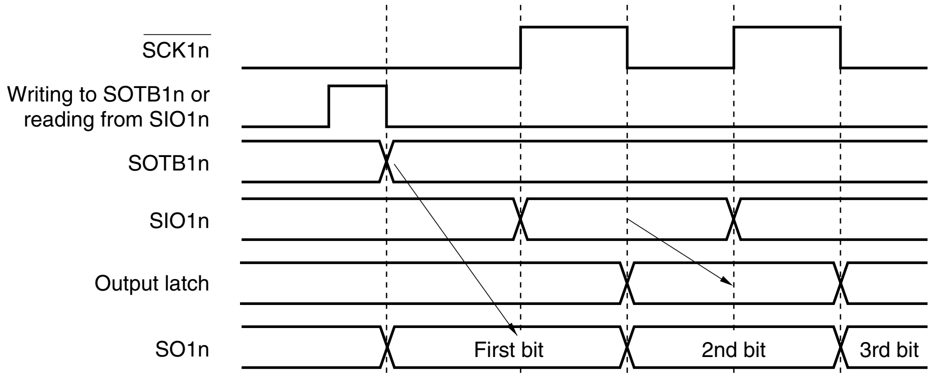
Remark n = 0, 1

Figure 13-11. Output Operation of First Bit (2/2)

(c) Type 2: CKP1n = 0, DAP1n = 1


$\overline{\text{SCK1n}}$
Writing to SOTB1n or reading from SIO1n
SOTB1n
SIO1n
Output latch
SO1n
First bit
2nd bit
3rd bit

(d) Type 4: CKP1n = 1, DAP1n = 1


$\overline{\text{SCK1n}}$
Writing to SOTB1n or reading from SIO1n
SOTB1n
SIO1n
Output latch
SO1n
First bit
2nd bit
3rd bit

The first bit is directly latched by the SOTB1n register at the falling edge of the write signal of the SOTB1n register or the read signal of the SIO1n register, and output from the SO1n pin via an output selector. Then, the value of the SOTB1n register is transferred to the SIO1n register at the next falling (or rising) edge of $\overline{\text{SCK1n}}$, and shifted one bit. At the same time, the first bit of the receive data is stored in the SIO1n register via the SI1n pin. The second and subsequent bits are latched by the SIO1n register to the output latch at the next rising (or falling) edge of $\overline{\text{SCK1n}}$, and the data is output from the SO1n pin.

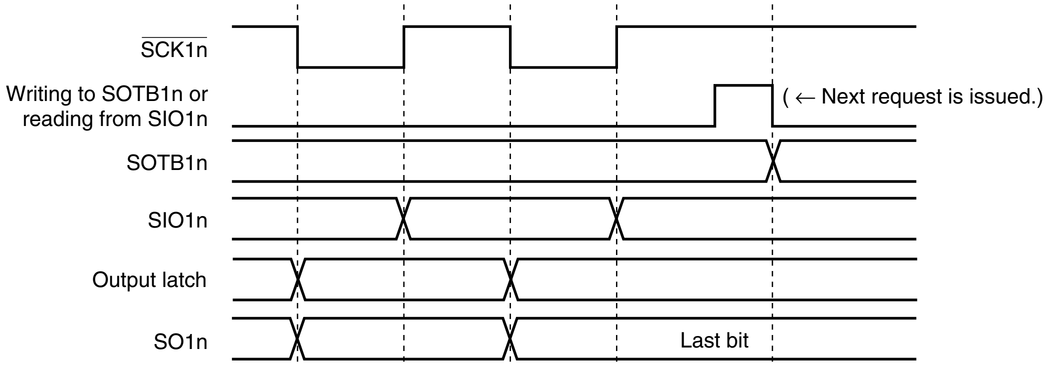
Remark n = 0, 1

(4) Output value of SO1n pin (last bit)

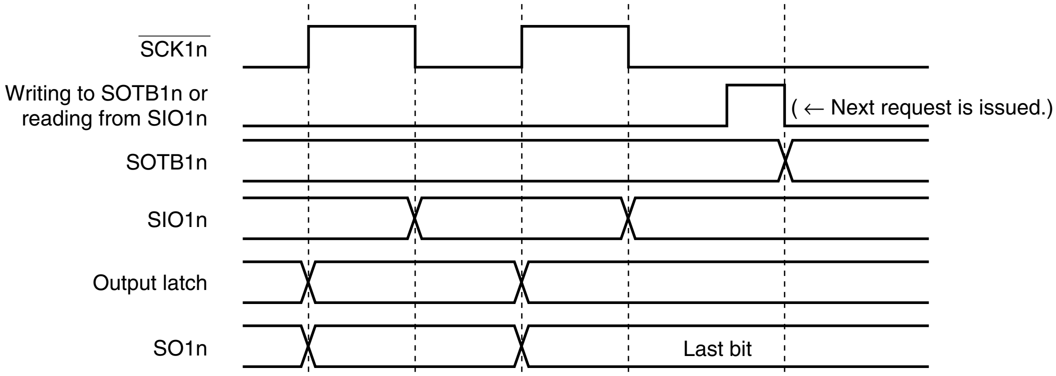
After communication has been completed, the SO1n pin holds the output value of the last bit.

Figure 13-12. Output Value of SO1n Pin (Last Bit) (1/2)

(a) Type 1: CKP1n = 0, DAP1n = 0


$\overline{\text{SCK1n}}$
Writing to SOTB1n or reading from SIO1n
(← Next request is issued.)
SOTB1n
SIO1n
Output latch
SO1n
Last bit

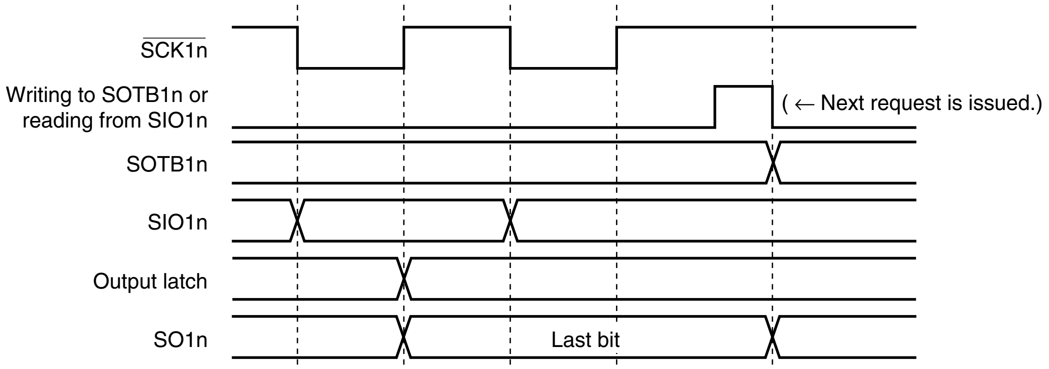
(b) Type 3: CKP1n = 1, DAP1n = 0


$\overline{\text{SCK1n}}$
Writing to SOTB1n or reading from SIO1n
(← Next request is issued.)
SOTB1n
SIO1n
Output latch
SO1n
Last bit

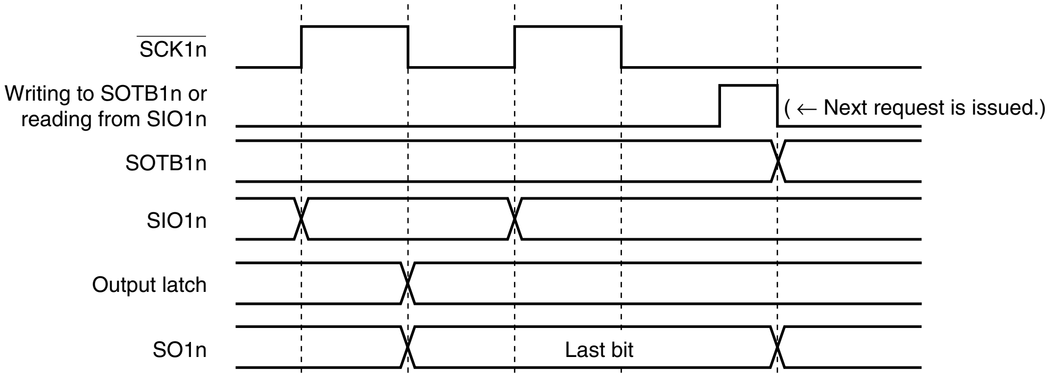
Remark n = 0, 1

Figure 13-12. Output Value of SO1n Pin (Last Bit) (2/2)

(c) Type 2: CKP1n = 0, DAP1n = 1


$\overline{\text{SCK1n}}$
Writing to SOTB1n or reading from SIO1n
(← Next request is issued.)
SOTB1n
SIO1n
Output latch
SO1n
Last bit

(d) Type 4: CKP1n = 1, DAP1n = 1


$\overline{\text{SCK1n}}$
Writing to SOTB1n or reading from SIO1n
(← Next request is issued.)
SOTB1n
SIO1n
Output latch
SO1n
Last bit

Remark n = 0, 1

(5) SO1n output (see (a) in Figures 13-1 and 13-2)

The status of the SO1n output is as follows if bit 7 (CSIE1n) of serial operation mode register 1n (CSIM1n) is cleared to 0.

Table 13-3. SO1n Output Status

TRMD1n	DAP1n	DIR1n	SO1n Output[Note 1]
TRMD1n = 0[Note 2]	–	–	Outputs low level[Note 2]
TRMD1n = 1	DAP1n = 0	–	Value of SO1n latch (low-level output)
	DAP1n = 1	DIR1n = 0	Value of bit 7 of SOTB1n
		DIR1n = 1	Value of bit 0 of SOTB1n

Notes 1. The actual output of the SO10/INTP2/P12 or SO11/P76 pin is determined according to PM12 and P12 or PM76 and P76, as well as the SO1n output.

2. Status after reset

Caution If a value is written to TRMD1n, DAP1n, and DIR1n, the output value of SO1n changes.

Remark n = 0, 1

CHAPTER 14 SERIAL INTERFACE IIC0

Caution Do not use serial interface IIC0 and the multiplier/divider simultaneously, because various flags corresponding to interrupt request sources are shared among serial interface IIC0 and the multiplier/divider.

14.1 Functions of Serial Interface IIC0

Serial interface IIC0 are mounted onto all 78K0/Dx2 microcontroller products.
Serial interface IIC0 has the following two modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

(2) I^{2}C bus mode (multimaster supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCL0) line and a serial data bus (SDA0) line.

This mode complies with the I^{2}C bus format and the master device can generated “start condition”, “address”, “transfer direction specification”, “data”, and “stop condition” data to the slave device, via the serial data bus. The slave device automatically detects these received status and data by hardware. This function can simplify the part of application program that controls the I^{2}C bus.

Since the SCL0 and SDA0 pins are used for open drain outputs, IIC0 requires pull-up resistors for the serial clock line and the serial data bus line.

Figure 14-1 shows a block diagram of serial interface IIC0.

Figure 14-1. Block Diagram of Serial Interface IIC0

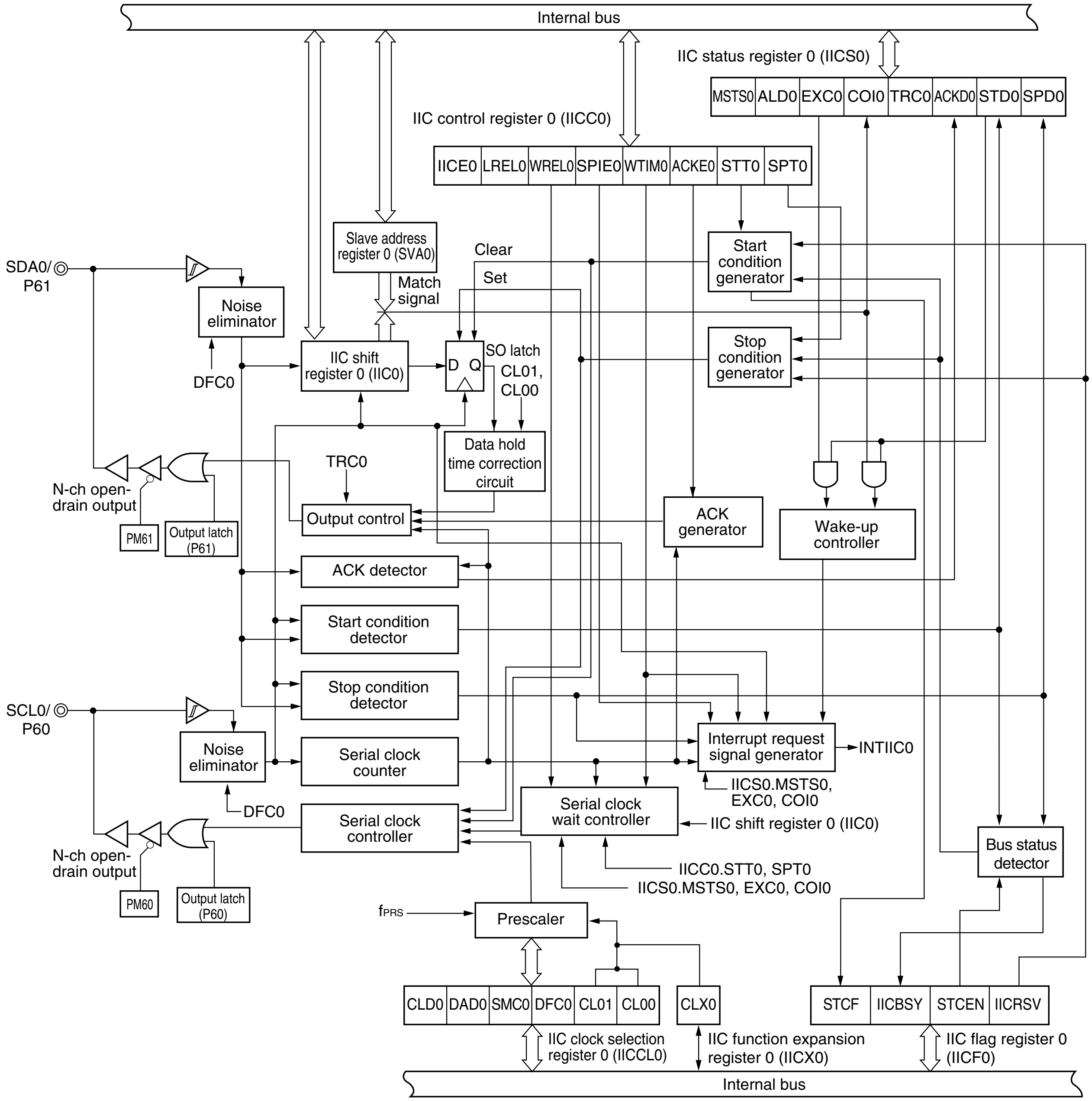
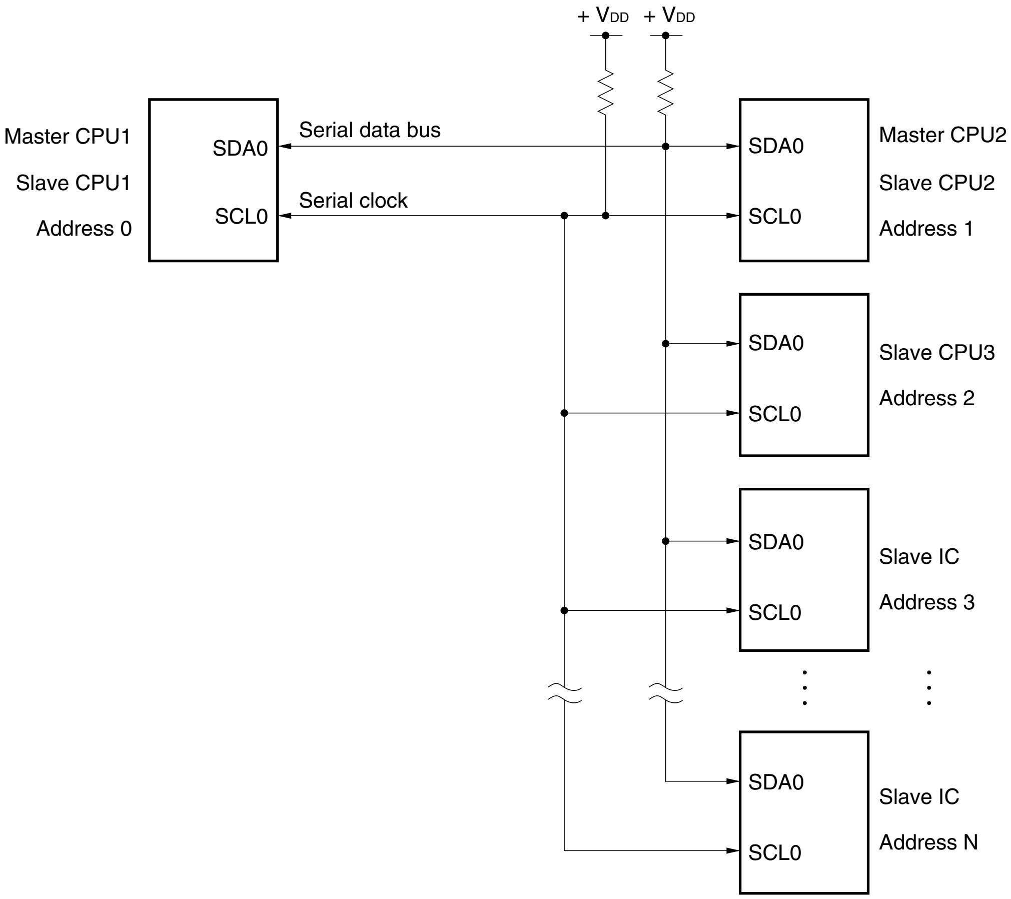


Figure 14-2 shows a serial bus configuration example.

Figure 14-2. Serial Bus Configuration Example Using I^{2}C Bus



14.2 Configuration of Serial Interface IIC0

Serial interface IIC0 includes the following hardware.

Table 14-1. Configuration of Serial Interface IIC0

Item	Configuration
Registers	IIC shift register 0 (IIC0) Slave address register 0 (SVA0)
Control registers	IIC control register 0 (IICC0) IIC status register 0 (IICS0) IIC flag register 0 (IICF0) IIC clock selection register 0 (IICCL0) IIC function expansion register 0 (IICX0) Port mode register 6 (PM6) Port register 6 (P6)

(1) IIC shift register 0 (IIC0)

IIC0 is used to convert 8-bit serial data to 8-bit parallel data and vice versa in synchronization with the serial clock. IIC0 can be used for both transmission and reception.

The actual transmit and receive operations can be controlled by writing and reading operations to IIC0.

Cancel the wait state and start data transfer by writing data to IIC0 during the wait period.

IIC0 is set by an 8-bit memory manipulation instruction.

Reset signal generation clears IIC0 to 00H.

Figure 14-3. Format of IIC Shift Register 0 (IIC0)

Address: FFB0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IIC0								

Cautions 1. Do not write data to IIC0 during data transfer.

2. Write or read IIC0 only during the wait period. Accessing IIC0 in a communication state other than during the wait period is prohibited. When the device serves as the master, however, IIC0 can be written only once after the communication trigger bit (STT0) is set to 1.

(2) Slave address register 0 (SVA0)

This register stores local addresses when in slave mode.

SVA0 is set by an 8-bit memory manipulation instruction.

However, rewriting to this register is prohibited while STD0 = 1 (while the start condition is detected).

Reset signal generation clears SVA0 to 00H.

Figure 14-4. Format of Slave Address Register 0 (SVA0)

Address: FFB2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SVA0								0[Note]

Note Bit 0 is fixed to 0.

(3) SO latch

The SO latch is used to retain the SDA0 pin's output level.

(4) Wake-up controller

This circuit generates an interrupt request (INTIIC0) when the address received by this register matches the address value set to slave address register 0 (SVA0) or when an extension code is received.

(5) Prescaler

This selects the sampling clock to be used.

(6) Serial clock counter

This counter counts the serial clocks that are output or input during transmit/receive operations and is used to verify that 8-bit data was transmitted or received.

(7) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (INTIIC0).

An I^{2}C interrupt request is generated by the following two triggers.

- Falling edge of eighth or ninth clock of the serial clock (set by WTIM0 bit)
- Interrupt request generated when a stop condition is detected (set by SPIE0 bit)

Remark WTIM0 bit: Bit 3 of IIC control register 0 (IICC0)
SPIE0 bit: Bit 4 of IIC control register 0 (IICC0)

(8) Serial clock controller

In master mode, this circuit generates the clock output via the SCL0 pin from a sampling clock.

(9) Serial clock wait controller

This circuit controls the wait timing.

(10) ACK generator, stop condition detector, start condition detector, and ACK detector

These circuits generate and detect each status.

(11) Data hold time correction circuit

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

(12) Start condition generator

This circuit generates a start condition when the STT0 bit is set to 1.

However, in the communication reservation disabled status (IICRSV bit = 1), when the bus is not released (IICBSY bit = 1), start condition requests are ignored and the STCF bit is set to 1.

(13) Stop condition generator

This circuit generates a stop condition when the SPT0 bit is set to 1.

(14) Bus status detector

This circuit detects whether or not the bus is released by detecting start conditions and stop conditions.

However, as the bus status cannot be detected immediately following operation, the initial status is set by the STCEN bit.

Remark		
	STT0 bit:	Bit 1 of IIC control register 0 (IICC0)
	SPT0 bit:	Bit 0 of IIC control register 0 (IICC0)
	IICRSV bit:	Bit 0 of IIC flag register 0 (IICF0)
	IICBSY bit:	Bit 6 of IIC flag register 0 (IICF0)
	STCF bit:	Bit 7 of IIC flag register 0 (IICF0)
	STCEN bit:	Bit 1 of IIC flag register 0 (IICF0)

14.3 Registers to Control Serial Interface IIC0

Serial interface IIC0 is controlled by the following seven registers.

- IIC control register 0 (IICC0)
- IIC flag register 0 (IICF0)
- IIC status register 0 (IICS0)
- IIC clock selection register 0 (IICCL0)
- IIC function expansion register 0 (IICX0)
- Port mode register 6 (PM6)
- Port register 6 (P6)

(1) IIC control register 0 (IICC0)

This register is used to enable/stop I^{2}C operations, set wait timing, and set other I^{2}C operations.

IICC0 is set by a 1-bit or 8-bit memory manipulation instruction. However, set the SPIE0, WTIM0, and ACKE0 bits while IICE0 bit = 0 or during the wait period. These bits can be set at the same time when the IICE0 bit is set from “0” to “1”.

Reset signal generation clears IICC0 to 00H.

Figure 14-5. Format of IIC Control Register 0 (IICC0) (1/4)

Address: FFB1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICC0	IICE0	LREL0	WREL0	SPIE0	WTIM0	ACKE0	STT0	SPT0

IICE0	I^2C operation enable	
0	Stop operation. Reset IIC status register 0 (IICS0)[Note 1]. Stop internal operation.	
1	Enable operation.	
Be sure to set this bit (1) while the SCL0 and SDA0 lines are at high level.		
Condition for clearing (IICE0 = 0)		Condition for setting (IICE0 = 1)
• Cleared by instruction • Reset		• Set by instruction

LREL0[Note 2]	Exit from communications	
0	Normal operation	
1	This exits from the current communications and sets standby mode. This setting is automatically cleared to 0 after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCL0 and SDA0 lines are set to high impedance. The following flags of IIC control register 0 (IICC0) and IIC status register 0 (IICS0) are cleared to 0. • STT0 • SPT0 • MSTS0 • EXC0 • COI0 • TRC0 • ACKD0 • STD0	
The standby mode following exit from communications remains in effect until the following communications entry conditions are met. • After a stop condition is detected, restart is in master mode. • An address match or extension code reception occurs after the start condition.		
Condition for clearing (LREL0 = 0)		Condition for setting (LREL0 = 1)
• Automatically cleared after execution • Reset		• Set by instruction

WREL0[Note 2]	Wait cancellation	
0	Do not cancel wait	
1	Cancel wait. This setting is automatically cleared after wait is canceled.	
When WREL0 is set (wait canceled) during the wait period at the ninth clock pulse in the transmission status (TRC0 = 1), the SDA0 line goes into the high impedance state (TRC0 = 0).		
Condition for clearing (WREL0 = 0)		Condition for setting (WREL0 = 1)
• Automatically cleared after execution • Reset		• Set by instruction

Notes 1. The IICS0 register, the STCF0 and IICBSY bits of the IICF0 register, and the CLD0 and DAD0 bits of the IICCL0 register are reset.

2. This flag's signal is invalid when IICE0 = 0.

Caution The start condition is detected immediately after I^2C is enabled to operate (IICE0 = 1) while the SCL0 line is at high level and the SDA0 line is at low level. Immediately after enabling I^2C to operate (IICE0 = 1), set LREL0 (1) by using a 1-bit memory manipulation instruction.

Figure 14-5. Format of IIC Control Register 0 (IICC0) (2/4)

SPIE0[Note 1]	Enable/disable generation of interrupt request when stop condition is detected
0	Disable
1	Enable
Condition for clearing (SPIE0 = 0)	Condition for setting (SPIE0 = 1)
• Cleared by instruction • Reset	• Set by instruction

WTIM0[Note 1]	Control of wait and interrupt request generation
0	Interrupt request is generated at the eighth clock's falling edge. Master mode: After output of eight clocks, clock output is set to low level and wait is set. Slave mode: After input of eight clocks, the clock is set to low level and wait is set for master device.
1	Interrupt request is generated at the ninth clock's falling edge. Master mode: After output of nine clocks, clock output is set to low level and wait is set. Slave mode: After input of nine clocks, the clock is set to low level and wait is set for master device.
An interrupt is generated at the falling edge of the ninth clock during address transfer independently of the setting of this bit. The setting of this bit is valid when the address transfer is completed. When in master mode, a wait is inserted at the falling edge of the ninth clock during address transfers. For a slave device that has received a local address, a wait is inserted at the falling edge of the ninth clock after an acknowledge ($\overline{\text{ACK}}$) is issued. However, when the slave device has received an extension code, a wait is inserted at the falling edge of the eighth clock.	
Condition for clearing (WTIM0 = 0)	Condition for setting (WTIM0 = 1)
• Cleared by instruction • Reset	• Set by instruction

ACKE0[Notes 1, 2]	Acknowledgment control
0	Disable acknowledgment.
1	Enable acknowledgment. During the ninth clock period, the SDA0 line is set to low level.
Condition for clearing (ACKE0 = 0)	Condition for setting (ACKE0 = 1)
• Cleared by instruction • Reset	• Set by instruction

Notes 1. This flag's signal is invalid when IICE0 = 0.

2. The set value is invalid during address transfer and if the code is not an extension code.
When the device serves as a slave and the addresses match, an acknowledge is generated regardless of the set value.

Figure 14-5. Format of IIC Control Register 0 (IICC0) (3/4)

<table>
<tr><th>STT0[Note]</th><th colspan="2">Start condition trigger</th></tr>
<tr><td>0</td><td colspan="2">Do not generate a start condition.</td></tr>
<tr><td>1</td><td colspan="2">When bus is released (in STOP mode):
Generate a start condition (for starting as master). When the SCL0 line is high level, the SDA0 line is changed from high level to low level and then the start condition is generated. Next, after the rated amount of time has elapsed, SCL0 is changed to low level (wait state).
When a third party is communicating:
• When communication reservation function is enabled (IICRSV = 0)
Functions as the start condition reservation flag. When set to 1, automatically generates a start condition after the bus is released.
• When communication reservation function is disabled (IICRSV = 1)
STCF is set to 1 and information that is set (1) to STT0 is cleared. No start condition is generated.
In the wait state (when master device):
Generates a restart condition after releasing the wait.</td></tr>
<tr><td colspan="3">Cautions concerning set timing
• For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in the waiting period when ACKE0 has been cleared to 0 and slave has been notified of final reception.
• For master transmission: A start condition cannot be generated normally during the acknowledge period. Set to 1 during the wait period that follows output of the ninth clock.
• Cannot be set to 1 at the same time as SPT0.
• Setting STT0 to 1 and then setting it again before it is cleared to 0 is prohibited.</td></tr>
<tr><td colspan="2">Condition for clearing (STT0 = 0)</td><td>Condition for setting (STT0 = 1)</td></tr>
<tr><td colspan="2">• Cleared by setting SST0 to 1 while communication reservation is prohibited.
• Cleared by loss in arbitration
• Cleared after start condition is generated by master device
• Cleared by LREL0 = 1 (exit from communications)
• When IICE0 = 0 (operation stop)
• Reset</td><td>• Set by instruction</td></tr>
</table>

Note This flag's signal is invalid when IICE0 = 0.

Remarks 1. Bit 1 (STT0) becomes 0 when it is read after data setting.

2. IICRSV: Bit 0 of IIC flag register (IICF0)
STCF: Bit 7 of IIC flag register (IICF0)

Figure 14-5. Format of IIC Control Register 0 (IICC0) (4/4)

SPT0	Stop condition trigger
0	Stop condition is not generated.
1	Stop condition is generated (termination of master device's transfer). After the SDA0 line goes to low level, either set the SCL0 line to high level or wait until it goes to high level. Next, after the rated amount of time has elapsed, the SDA0 line changes from low level to high level and a stop condition is generated.

Cautions concerning set timing

- For master reception: Cannot be set to 1 during transfer.
  Can be set to 1 only in the waiting period when ACKE0 has been cleared to 0 and slave has been notified of final reception.
- For master transmission: A stop condition cannot be generated normally during the acknowledge period. Therefore, set it during the wait period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as STT0.
- SPT0 can be set to 1 only when in master mode[Note].
- When WTIM0 has been cleared to 0, if SPT0 is set to 1 during the wait period that follows output of eight clocks, note that a stop condition will be generated during the high-level period of the ninth clock. WTIM0 should be changed from 0 to 1 during the wait period following the output of eight clocks, and SPT0 should be set to 1 during the wait period that follows the output of the ninth clock.
- Setting SPT0 to 1 and then setting it again before it is cleared to 0 is prohibited.

Condition for clearing (SPT0 = 0)	Condition for setting (SPT0 = 1)
• Cleared by loss in arbitration • Automatically cleared after stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When IICE0 = 0 (operation stop) • Reset	• Set by instruction

Note Set SPT0 to 1 only in master mode. However, SPT0 must be set to 1 and a stop condition generated before the first stop condition is detected following the switch to the operation enabled status. For details, see **14.5.15 Cautions**.

Caution When bit 3 (TRC0) of IIC status register 0 (IICS0) is set to 1, WREL0 is set to 1 during the ninth clock and wait is canceled, after which TRC0 is cleared and the SDA0 line is set to high impedance.

Remark Bit 0 (SPT0) becomes 0 when it is read after data setting.

(2) IIC status register 0 (IICS0)

This register indicates the status of I^2C.

IICS0 is read by a 1-bit or 8-bit memory manipulation instruction only when STT0 = 1 and during the wait period.

Reset signal generation clears IICS0 to 00H.

Caution If data is read from IICS0, a wait cycle is generated. Do not read data from IICS0 when the peripheral hardware clock (f_{PRS}) is stopped. For details, see CHAPTER 32 CAUTIONS FOR WAIT.

Figure 14-6. Format of IIC Status Register 0 (IICS0) (1/3)

Address: FFB5H After reset: 00H R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICS0	MSTS0	ALD0	EXC0	COI0	TRC0	ACKD0	STD0	SPD0

MSTS0	Master device status
0	Slave device status or communication standby status
1	Master device communication status
Condition for clearing (MSTS0 = 0)	Condition for setting (MSTS0 = 1)
• When a stop condition is detected • When ALD0 = 1 (arbitration loss) • Cleared by LREL0 = 1 (exit from communications) • When IICE0 changes from 1 to 0 (operation stop) • Reset	• When a start condition is generated

ALD0	Detection of arbitration loss
0	This status means either that there was no arbitration or that the arbitration result was a “win”.
1	This status indicates the arbitration result was a “loss”. MSTS0 is cleared.
Condition for clearing (ALD0 = 0)	Condition for setting (ALD0 = 1)
• Automatically cleared after IICS0 is read[Note] • When IICE0 changes from 1 to 0 (operation stop) • Reset	• When the arbitration result is a “loss”.

EXC0	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received.
Condition for clearing (EXC0 = 0)	Condition for setting (EXC0 = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When IICE0 changes from 1 to 0 (operation stop) • Reset	• When the higher four bits of the received address data is either “0000” or “1111” (set at the rising edge of the eighth clock).

Note This register is also cleared when a 1-bit memory manipulation instruction is executed for bits other than IICS0. Therefore, when using the ALD0 bit, read the data of this bit before the data of the other bits.

Remark LREL0: Bit 6 of IIC control register 0 (IICC0)
IICE0: Bit 7 of IIC control register 0 (IICC0)

Figure 14-6. Format of IIC Status Register 0 (IICS0) (2/3)

COI0	Detection of matching addresses
0	Addresses do not match.
1	Addresses match.
Condition for clearing (COI0 = 0)	Condition for setting (COI0 = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When IICE0 changes from 1 to 0 (operation stop) • Reset	• When the received address matches the local address (slave address register 0 (SVA0)) (set at the rising edge of the eighth clock).

TRC0	Detection of transmit/receive status
0	Receive status (other than transmit status). The SDA0 line is set for high impedance.
1	Transmit status. The value in the SO0 latch is enabled for output to the SDA0 line (valid starting at the falling edge of the first byte's ninth clock).
Condition for clearing (TRC0 = 0)	Condition for setting (TRC0 = 1)
<Both master and slave> • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When IICE0 changes from 1 to 0 (operation stop) • Cleared by WREL0 = 1[Note] (wait cancel) • When ALD0 changes from 0 to 1 (arbitration loss) • Reset <Master> • When "1" is output to the first byte's LSB (transfer direction specification bit) <Slave> • When a start condition is detected • When "0" is input to the first byte's LSB (transfer direction specification bit) <When not used for communication>	<Master> • When a start condition is generated • When "0" is output to the first byte's LSB (transfer direction specification bit) <Slave> • When "1" is input to the first byte's LSB (transfer direction specification bit)

Note If the wait state is canceled by setting bit 5 (WREL0) of IIC control register 0 (IICC0) to 1 at the ninth clock when bit 3 (TRC0) of IIC status register 0 (IICS0) is 1, TRC0 is cleared, and the SDA0 line goes into a high-impedance state.

Remark LREL0: Bit 6 of IIC control register 0 (IICC0)
IICE0: Bit 7 of IIC control register 0 (IICC0)

Figure 14-6. Format of IIC Status Register 0 (IICS0) (3/3)

ACKD0	Detection of acknowledge ($\overline{\text{ACK}}$)
0	Acknowledge was not detected.
1	Acknowledge was detected.
Condition for clearing (ACKD0 = 0)	Condition for setting (ACKD0 = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock • Cleared by LREL0 = 1 (exit from communications) • When IICE0 changes from 1 to 0 (operation stop) • Reset	• After the SDA0 line is set to low level at the rising edge of SCL0's ninth clock

STD0	Detection of start condition
0	Start condition was not detected.
1	Start condition was detected. This indicates that the address transfer period is in effect.
Condition for clearing (STD0 = 0)	Condition for setting (STD0 = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock following address transfer • Cleared by LREL0 = 1 (exit from communications) • When IICE0 changes from 1 to 0 (operation stop) • Reset	• When a start condition is detected

SPD0	Detection of stop condition
0	Stop condition was not detected.
1	Stop condition was detected. The master device's communication is terminated and the bus is released.
Condition for clearing (SPD0 = 0)	Condition for setting (SPD0 = 1)
• At the rising edge of the address transfer byte's first clock following setting of this bit and detection of a start condition • When IICE0 changes from 1 to 0 (operation stop) • Reset	• When a stop condition is detected

Remark LREL0: Bit 6 of IIC control register 0 (IICC0)
IICE0: Bit 7 of IIC control register 0 (IICC0)

(3) IIC flag register 0 (IICF0)

This register sets the operation mode of I^{2}C and indicates the status of the I^{2}C bus.

IICF0 is set by a 1-bit or 8-bit memory manipulation instruction. However, the STCF and IICBSY bits are read-only.

The IICRSV bit can be used to enable/disable the communication reservation function (see **14.5.14 Communication reservation**).

STCEN can be used to set the initial value of the IICBSY bit (see **14.5.15 Cautions**).

IICRSV and STCEN can be written only when the operation of I^{2}C is disabled (bit 7 (IICE0) of IIC control register 0 (IICC0) = 0). When operation is enabled, the IICF0 register can be read.

Reset signal generation clears IICF0 to 00H.

Figure 14-7. Format of IIC Flag Register 0 (IICF0)

Address: FFB7H After reset: 00H R/W[Note]

Symbol	<7>	<6>	5	4	3	2	<1>	<0>
IICF0	STCF	IICBSY	0	0	0	0	STCEN	IICRSV

STCF	STT0 clear flag
0	Generate start condition
1	Start condition generation unsuccessful: clear STT0 flag
Condition for clearing (STCF = 0)	Condition for setting (STCF = 1)
• Cleared by STT0 = 1 • When IICE0 = 0 (operation stop) • Reset	• Generating start condition unsuccessful and STT0 cleared to 0 when communication reservation is disabled (IICRSV = 1).

IICBSY	I^{2}C bus status flag
0	Bus release status (communication initial status when STCEN = 1)
1	Bus communication status (communication initial status when STCEN = 0)
Condition for clearing (IICBSY = 0)	Condition for setting (IICBSY = 1)
• Detection of stop condition • When IICE0 = 0 (operation stop) • Reset	• Detection of start condition • Setting of IICE0 when STCEN = 0

STCEN	Initial start enable trigger
0	After operation is enabled (IICE0 = 1), enable generation of a start condition upon detection of a stop condition.
1	After operation is enabled (IICE0 = 1), enable generation of a start condition without detecting a stop condition.
Condition for clearing (STCEN = 0)	Condition for setting (STCEN = 1)
• Detection of start condition • Reset	• Set by instruction

IICRSV	Communication reservation function disable bit
0	Enable communication reservation
1	Disable communication reservation
Condition for clearing (IICRSV = 0)	Condition for setting (IICRSV = 1)
• Cleared by instruction • Reset	• Set by instruction

Note Bits 6 and 7 are read-only.

Cautions 1. Write to STCEN only when the operation is stopped (IICE0 = 0).

2. As the bus release status (IICBSY = 0) is recognized regardless of the actual bus status when STCEN = 1, when generating the first start condition (STT0 = 1), it is necessary to verify that no third party communications are in progress in order to prevent such communications from being destroyed.

3. Write to IICRSV only when the operation is stopped (IICE0 = 0).

Remark STT0: Bit 1 of IIC control register 0 (IICC0)
IICE0: Bit 7 of IIC control register 0 (IICC0)

(4) IIC clock selection register 0 (IICCL0)

This register is used to set the transfer clock for the I^2C bus.

IICCL0 is set by a 1-bit or 8-bit memory manipulation instruction. However, the CLD0 and DAD0 bits are read-only. The SMC0, CL01, and CL00 bits are set in combination with bit 0 (CLX0) of IIC function expansion register 0 (IICX0) (see **(6) I^2C transfer clock setting method**).

Set IICCL0 while bit 7 (IICE0) of IIC control register 0 (IICC0) is 0.

Reset signal generation clears IICCL0 to 00H.

Figure 14-8. Format of IIC Clock Selection Register 0 (IICCL0)

Address: FFB3H After reset: 00H R/W[Note]

Symbol	7	6	<5>	<4>	<3>	<2>	1	0
IICCL0	0	0	CLD0	DAD0	SMC0	DFC0	CL01	CL00

CLD0	Detection of SCL0 pin level (valid only when IICE0 = 1)
0	The SCL0 pin was detected at low level.
1	The SCL0 pin was detected at high level.
Condition for clearing (CLD0 = 0)	Condition for setting (CLD0 = 1)
• When the SCL0 pin is at low level • When IICE0 = 0 (operation stop) • Reset	• When the SCL0 pin is at high level

DAD0	Detection of SDA0 pin level (valid only when IICE0 = 1)
0	The SDA0 pin was detected at low level.
1	The SDA0 pin was detected at high level.
Condition for clearing (DAD0 = 0)	Condition for setting (DAD0 = 1)
• When the SDA0 pin is at low level • When IICE0 = 0 (operation stop) • Reset	• When the SDA0 pin is at high level

SMC0	Operation mode switching
0	Operates in standard mode.
1	Operates in high-speed mode.

DFC0	Digital filter operation control
0	Digital filter off.
1	Digital filter on.
Digital filter can be used only in high-speed mode. In high-speed mode, the transfer clock does not vary regardless of DFC0 bit set (1)/clear (0). The digital filter is used for noise elimination in high-speed mode.	

Note Bits 4 and 5 are read-only.

Remark IICE0: Bit 7 of IIC control register 0 (IICC0)

(5) IIC function expansion register 0 (IICX0)

This register sets the function expansion of I^{2}C.

IICX0 is set by a 1-bit or 8-bit memory manipulation instruction. The CLX0 bit is set in combination with bits 3, 1, and 0 (SMC0, CL01, and CL00) of IIC clock selection register 0 (IICCL0) (see **(6) I^{2}C transfer clock setting method**).

Set IICX0 while bit 7 (IICE0) of IIC control register 0 (IICC0) is 0.

Reset signal generation clears IICX0 to 00H.

Figure 14-9. Format of IIC Function Expansion Register 0 (IICX0)

Address: FFB4H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
IICX0	0	0	0	0	0	0	0	CLX0

(6) I^{2}C transfer clock setting method

The I^{2}C transfer clock frequency (f_{SCL}) is calculated using the following expression.

$$f_{SCL} = 1/(m \times T + t_R + t_F)$$

m = 12, 18, 24, 44, 66, 86 (see **Table 14-2. Selection Clock Setting**)

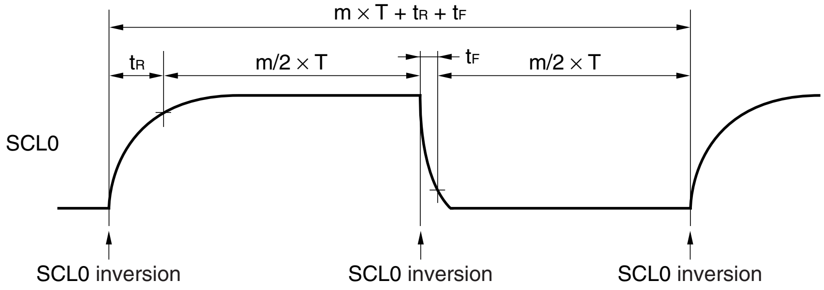
T: 1/f_W

t_R: SCL0 rise time

t_F: SCL0 fall time

For example, the I^{2}C transfer clock frequency (f_{SCL}) when $f_W = f_{PRS}/2 = 4.19$ MHz, m = 86, t_R = 200 ns, and t_F = 50 ns is calculated using following expression.

$$f_{SCL} = 1/(88 \times 238.7 \text{ ns} + 200 \text{ ns} + 50 \text{ ns}) \cong 48.1 \text{ kHz}$$



The selection clock is set using a combination of bits 3, 1, and 0 (SMC0, CL01, and CL00) of IIC clock selection register 0 (IICCL0) and bit 0 (CLX0) of IIC function expansion register 0 (IICX0).

Table 14-2. Selection Clock Setting

IICX0	IICCL0			Selection Clock (f_W)[Note]	Transfer Clock (f_W/m)	Settable Selection Clock (f_W) Range	Operation Mode
Bit 0	Bit 3	Bit 1	Bit 0				
CLX0	SMC0	CL01	CL00				
0	0	0	0	f_{PRS}/2	f_W/44	2.00 to 4.19 MHz	Normal mode (SMC0 bit = 0)
0	0	0	1	f_{PRS}/2	f_W/86	4.19 to 8.38 MHz	
0	0	1	0	f_{PRS}/4	f_W/86		
0	0	1	1	Setting prohibited			
0	1	0	×	f_{PRS}/2	f_W/24	4.00 to 8.38 MHz	High-speed mode (SMC0 bit = 1)
0	1	1	0	f_{PRS}/4	f_W/24		
0	1	1	1	Setting prohibited			
1	0	×	×				
1	1	0	×	f_{PRS}/2	f_W/12	4.00 to 4.19 MHz	High-speed mode (SMC0 bit = 1)
1	1	1	0	f_{PRS}/4	f_W/12		
1	1	1	1	Setting prohibited			

Note If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{IN}) (XSEL = 0), set CLX0, SMC0, CL01 and CL00 as follows.

IICX0	IICCL0			Selection Clock (f_W)	Transfer Clock (f_W/m)	Settable Selection Clock (f_W) Range	Operation Mode
Bit 0	Bit 3	Bit 1	Bit 0				
CLX0	SMC0	CL01	CL00				
0	0	0	0	f_{PRS}/2	f_W/44	3.8 MHz to 4.2 MHz	Normal mode (SMC0 bit = 0)
0	1	0	×	f_{PRS}/2	f_W/24		High-speed mode (SMC0 bit = 1)

Caution Determine the transfer clock frequency of I^{2}C by using CLX0, SMC0, CL01, and CL00 before enabling the operation (by setting bit 7 (IICE0) of IIC control register 0 (IICC0) to 1). To change the transfer clock frequency, clear IICE0 once to 0.

Remarks 1. ×: don't care

2. f_{PRS}: Peripheral hardware clock frequency

(7) Port mode register 6 (PM6)

This register sets the input/output of port 6 in 1-bit units.

When using the P60/SCL0 pin as clock I/O and the P61/SDA0 pin as serial data I/O, clear PM60 and PM61, and the output latches of P60 and P61 to 0.

Set IICE0 (bit 7 of IIC control register 0 (IICC0)) to 1 before setting the output mode because the P60/SCL0 and P61/SDA0 pins output a low level (fixed) when IICE0 is 0.

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

Reset signal generation sets PM6 to FFH.

Figure 14-10. Format of Port Mode Register 6 (PM6)

Address: FF26H After reset: FFH R/W

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

PM6n	P6n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

14.4 I^{2}C Bus Mode Functions

14.4.1 Pin configuration

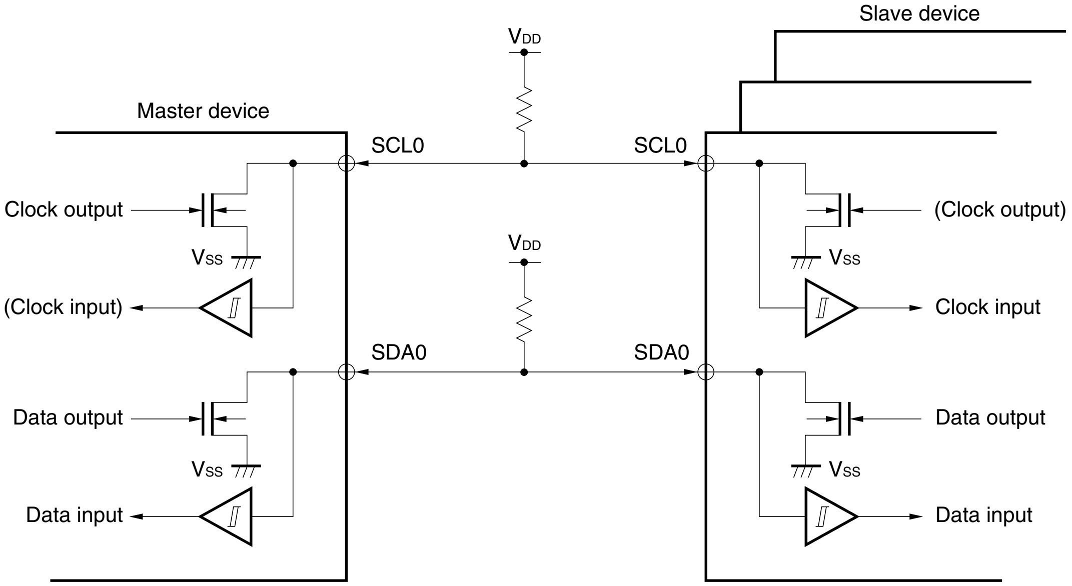
The serial clock pin (SCL0) and serial data bus pin (SDA0) are configured as follows.

(1) SCL0....... This pin is used for serial clock input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

(2) SDA0 ...... This pin is used for serial data input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required.

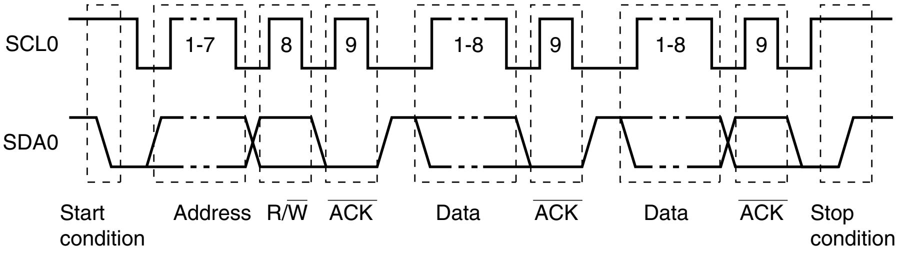
Figure 14-11. Pin Configuration Diagram



14.5 I²C Bus Definitions and Control Methods

The following section describes the I²C bus's serial data communication format and the signals used by the I²C bus. **Figure 14-12** shows the transfer timing for the "start condition", "address", "data", and "stop condition" output via the I²C bus's serial data bus.

Figure 14-12. I²C Bus Serial Data Transfer Timing



The master device generates the start condition, slave address, and stop condition.

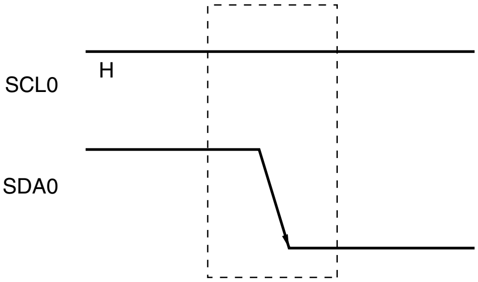
The acknowledge ($\overline{ACK}$) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCL0) is continuously output by the master device. However, in the slave device, the SCL0's low level period can be extended and a wait can be inserted.

14.5.1 Start conditions

A start condition is met when the SCL0 pin is at high level and the SDA0 pin changes from high level to low level. The start conditions for the SCL0 pin and SDA0 pin are signals that the master device generates to the slave device when starting a serial transfer. When the device is used as a slave, start conditions can be detected.

Figure 14-13. Start Conditions



A start condition is output when bit 1 (STT0) of IIC control register 0 (IICC0) is set (to 1) after a stop condition has been detected (SPD0: Bit 0 = 1 in IIC status register 0 (IICS0)). When a start condition is detected, bit 1 (STD0) of IICS0 is set (to 1).

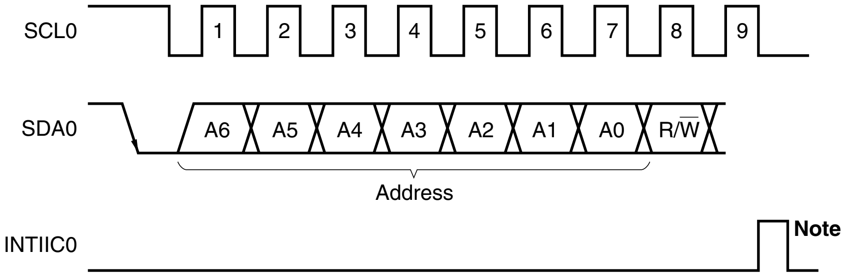
14.5.2 Addresses

The address is defined by the 7 bits of data that follow the start condition.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via the bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in slave address register 0 (SVA0). If the address data matches the SVA0 values, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

Figure 14-14. Address



Note INTIIC0 is not issued if data other than a local address or extension code is received during slave device operation.

The slave address and the eighth bit, which specifies the transfer direction as described in **14.5.3 Transfer direction specification** below, are together written to IIC shift register 0 (IIC0) and are then output. Received addresses are written to IIC0.

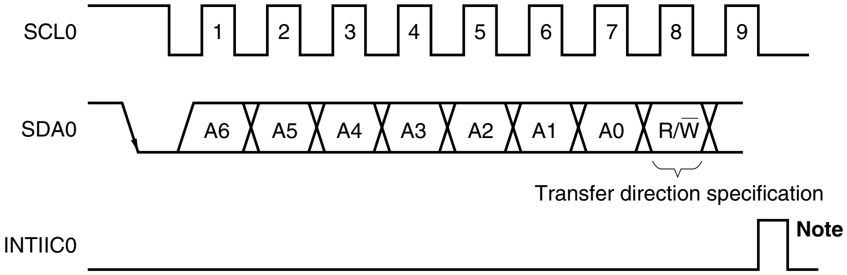
The slave address is assigned to the higher 7 bits of IIC0.

14.5.3 Transfer direction specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction.

When this transfer direction specification bit has a value of “0”, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of “1”, it indicates that the master device is receiving data from a slave device.

Figure 14-15. Transfer Direction Specification



Note INTIIC0 is not issued if data other than a local address or extension code is received during slave device operation.

14.5.4 Acknowledge ($\overline{\text{ACK}}$)

$\overline{\text{ACK}}$ is used to check the status of serial data at the transmission and reception sides.

The reception side returns $\overline{\text{ACK}}$ each time it has received 8-bit data.

The transmission side usually receives $\overline{\text{ACK}}$ after transmitting 8-bit data. When $\overline{\text{ACK}}$ is returned from the reception side, it is assumed that reception has been correctly performed and processing is continued. Whether $\overline{\text{ACK}}$ has been detected can be checked by using bit 2 (ACKD0) of IIC status register 0 (IICS0).

When the master receives the last data item, it does not return $\overline{\text{ACK}}$ and instead generates a stop condition. If a slave does not return $\overline{\text{ACK}}$ after receiving data, the master outputs a stop condition or restart condition and stops transmission. If $\overline{\text{ACK}}$ is not returned, the possible causes are as follows.

<1> Reception was not performed normally.

<2> The final data item was received.

<3> The reception side specified by the address does not exist.

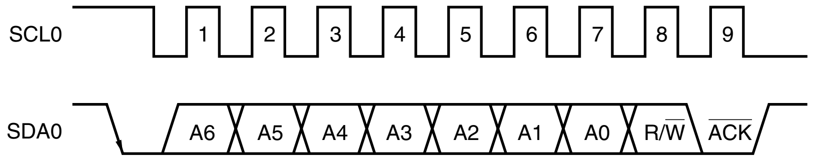
To generate $\overline{\text{ACK}}$, the reception side makes the SDA0 line low at the ninth clock (indicating normal reception).

Automatic generation of $\overline{\text{ACK}}$ is enabled by setting bit 2 (ACKE0) of IIC control register 0 (IICC0) to 1. Bit 3 (TRC0) of the IICS0 register is set by the data of the eighth bit that follows 7-bit address information. Usually, set ACKE0 to 1 for reception (TRC0 = 0).

If a slave can receive no more data during reception (TRC0 = 0) or does not require the next data item, then the slave must inform the master, by clearing ACKE0 to 0, that it will not receive any more data.

When the master does not require the next data item during reception (TRC0 = 0), it must clear ACKE0 to 0 so that $\overline{\text{ACK}}$ is not generated. In this way, the master informs a slave at the transmission side that it does not require any more data (transmission will be stopped).

Figure 14-16. $\overline{\text{ACK}}$



When the local address is received, $\overline{\text{ACK}}$ is automatically generated, regardless of the value of ACKE0. When an address other than that of the local address is received, $\overline{\text{ACK}}$ is not generated (NACK).

When an extension code is received, $\overline{\text{ACK}}$ is generated if ACKE0 is set to 1 in advance.

How $\overline{\text{ACK}}$ is generated when data is received differs as follows depending on the setting of the wait timing.

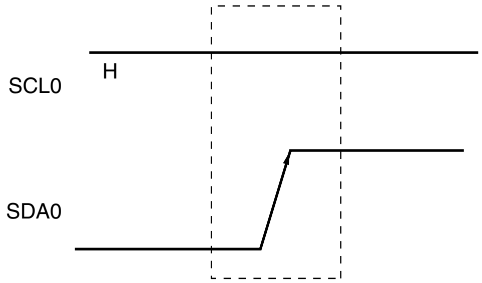
- When 8-clock wait state is selected (bit 3 (WTIM0) of IICC0 register = 0):
  By setting ACKE0 to 1 before releasing the wait state, $\overline{\text{ACK}}$ is generated at the falling edge of the eighth clock of the SCL0 pin.
- When 9-clock wait state is selected (bit 3 (WTIM0) of IICC0 register = 1):
  $\overline{\text{ACK}}$ is generated by setting ACKE0 to 1 in advance.

14.5.5 Stop condition

When the SCL0 pin is at high level, changing the SDA0 pin from low level to high level generates a stop condition.

A stop condition is a signal that the master device generates to the slave device when serial transfer has been completed. When the device is used as a slave, stop conditions can be detected.

Figure 14-17. Stop Condition



A stop condition is generated when bit 0 (SPT0) of IIC control register 0 (IICC0) is set to 1. When the stop condition is detected, bit 0 (SPD0) of IIC status register 0 (IICS0) is set to 1 and INTIIC0 is generated when bit 4 (SPIE0) of IICC0 is set to 1.

14.5.6 Wait

The wait is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a wait state).

Setting the SCL0 pin to low level notifies the communication partner of the wait state. When wait state has been canceled for both the master and slave devices, the next data transfer can begin.

Figure 14-18. Wait (1/2)

(1) When master device has a nine-clock wait and slave device has an eight-clock wait (master transmits, slave receives, and ACKE0 = 1)

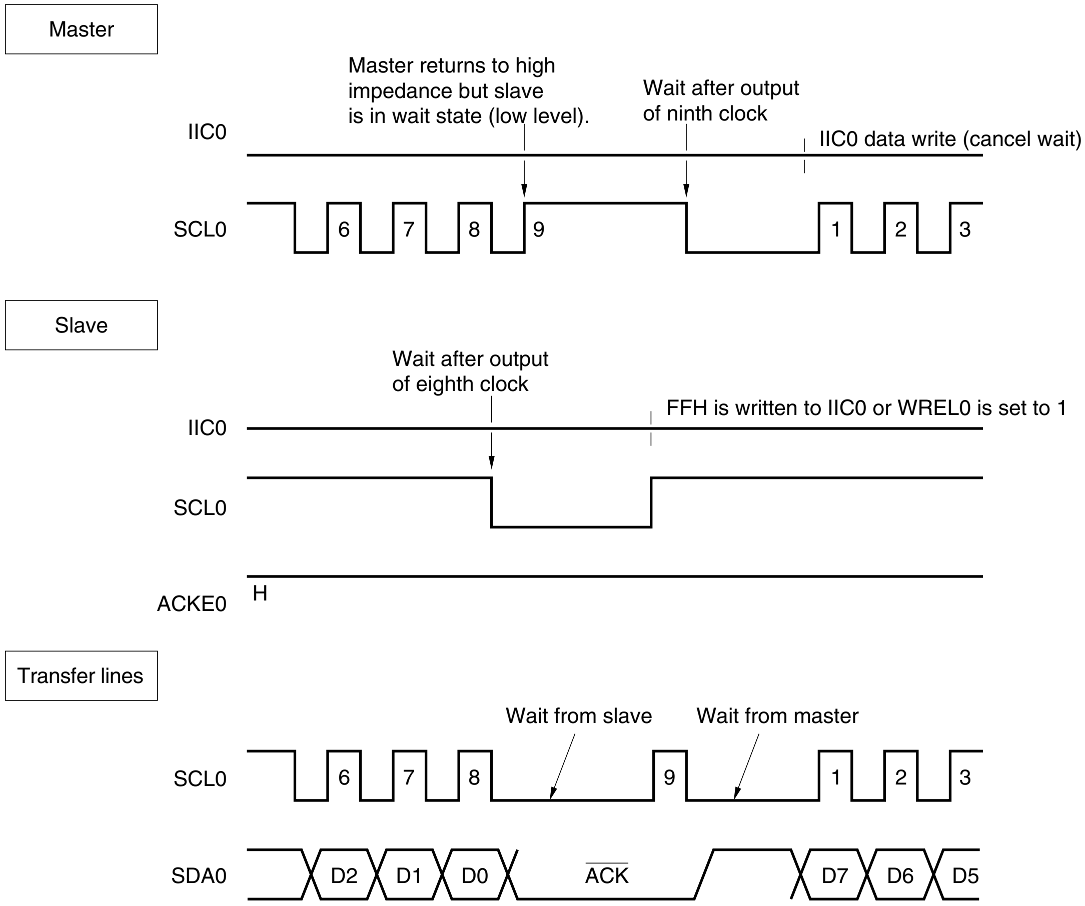
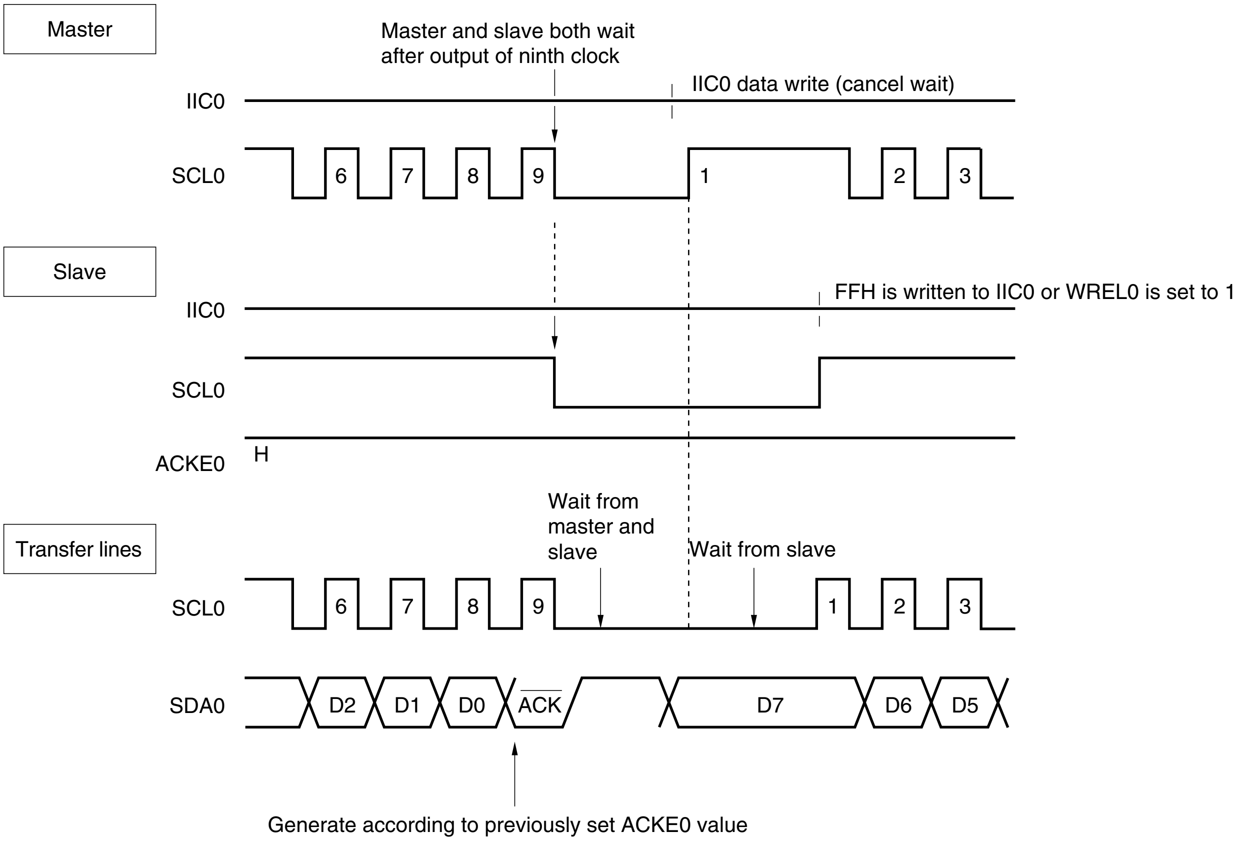

Master
Master returns to high impedance but slave is in wait state (low level).
Wait after output of ninth clock
IIC0
IIC0 data write (cancel wait)
SCL0
6
7
8
9
1
2
3
Slave
Wait after output of eighth clock
IIC0
FFH is written to IIC0 or WREL0 is set to 1
SCL0
ACKE0
H
Transfer lines
Wait from slave
Wait from master
SCL0
6
7
8
9
1
2
3
SDA0
D2
D1
D0
$\overline{\text{ACK}}$
D7
D6
D5

Figure 14-18. Wait (2/2)

(2) When master and slave devices both have a nine-clock wait (master transmits, slave receives, and ACKE0 = 1)


Master
Master and slave both wait after output of ninth clock
IIC0
IIC0 data write (cancel wait)
SCL0
6
7
8
9
1
2
3
Slave
IIC0
FFH is written to IIC0 or WREL0 is set to 1
SCL0
ACKE0
H
Transfer lines
Wait from master and slave
Wait from slave
SCL0
6
7
8
9
1
2
3
SDA0
D2
D1
D0
$\overline{\text{ACK}}$
D7
D6
D5
Generate according to previously set ACKE0 value

Remark ACKE0: Bit 2 of IIC control register 0 (IICC0)
WREL0: Bit 5 of IIC control register 0 (IICC0)

A wait may be automatically generated depending on the setting of bit 3 (WTIM0) of IIC control register 0 (IICC0).

Normally, the receiving side cancels the wait state when bit 5 (WREL0) of IICC0 is set to 1 or when FFH is written to IIC shift register 0 (IIC0), and the transmitting side cancels the wait state when data is written to IIC0.

The master device can also cancel the wait state via either of the following methods.

- By setting bit 1 (STT0) of IICC0 to 1
- By setting bit 0 (SPT0) of IICC0 to 1

14.5.7 Canceling wait

The I^{2}C usually cancels a wait state by the following processing.

- Writing data to IIC shift register 0 (IIC0)
- Setting bit 5 (WREL0) of IIC control register 0 (IICC0) (canceling wait)
- Setting bit 1 (STT0) of IIC0 register (generating start condition)[Note]
- Setting bit 0 (SPT0) of IIC0 register (generating stop condition)[Note]

Note Master only

When the above wait canceling processing is executed, the I^{2}C cancels the wait state and communication is resumed.

To cancel a wait state and transmit data (including addresses), write the data to IIC0.

To receive data after canceling a wait state, or to complete data transmission, set bit 5 (WREL0) of the IIC0 control register 0 (IICC0) to 1.

To generate a restart condition after canceling a wait state, set bit 1 (STT0) of IICC0 to 1.

To generate a stop condition after canceling a wait state, set bit 0 (SPT0) of IICC0 to 1.

Execute the canceling processing only once for one wait state.

If, for example, data is written to IIC0 after canceling a wait state by setting WREL0 to 1, an incorrect value may be output to SDA0 because the timing for changing the SDA0 line conflicts with the timing for writing IIC0.

In addition to the above, communication is stopped if IICE0 is cleared to 0 when communication has been aborted, so that the wait state can be canceled.

If the I^{2}C bus has deadlocked due to noise, processing is saved from communication by setting bit 6 (LREL0) of IICC0, so that the wait state can be canceled.

14.5.8 Interrupt request (INTIIC0) generation timing and wait control

The setting of bit 3 (WTIM0) of IIC control register 0 (IICC0) determines the timing by which INTIIC0 is generated and the corresponding wait control, as shown in **Table 14-3**.

Table 14-3. INTIIC0 Generation Timing and Wait Control

WTIM0	During Slave Device Operation			During Master Device Operation		
	Address	Data Reception	Data Transmission	Address	Data Reception	Data Transmission
0	9[Notes 1, 2]	8[Note 2]	8[Note 2]	9	8	8
1	9[Notes 1, 2]	9[Note 2]	9[Note 2]	9	9	9

Notes 1. The slave device's INTIIC0 signal and wait period occurs at the falling edge of the ninth clock only when there is a match with the address set to slave address register 0 (SVA0).

At this point, $\overline{\text{ACK}}$ is generated regardless of the value set to IICC0's bit 2 (ACKE0). For a slave device that has received an extension code, INTIIC0 occurs at the falling edge of the eighth clock.

However, if the address does not match after restart, INTIIC0 is generated at the falling edge of the 9th clock, but wait does not occur.

2. If the received address does not match the contents of slave address register 0 (SVA0) and extension code is not received, neither INTIIC0 nor a wait occurs.

Remark The numbers in the table indicate the number of the serial clock's clock signals. Interrupt requests and wait control are both synchronized with the falling edge of these clock signals.

(1) During address transmission/reception

- Slave device operation: Interrupt and wait timing are determined depending on the conditions described in Notes 1 and 2 above, regardless of the WTIM0 bit.
- Master device operation: Interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIM0 bit.

(2) During data reception

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

(3) During data transmission

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

(4) Wait cancellation method

The four wait cancellation methods are as follows.

- Writing data to IIC shift register 0 (IIC0)
- Setting bit 5 (WREL0) of IIC control register 0 (IICC0) (canceling wait)
- Setting bit 1 (STT0) of IIC0 register (generating start condition)[Note]
- Setting bit 0 (SPT0) of IIC0 register (generating stop condition)[Note]

Note Master only.

When an 8-clock wait has been selected (WTIM0 = 0), the presence/absence of $\overline{\mathrm{ACK}}$ generation must be determined prior to wait cancellation.

(5) Stop condition detection

INTIIC0 is generated when a stop condition is detected (only when SPIE0 = 1).

14.5.9 Address match detection method

In I^{2}C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

Address match can be detected automatically by hardware. An interrupt request (INTIIC0) occurs when a local address has been set to slave address register 0 (SVA0) and when the address set to SVA0 matches the slave address sent by the master device, or when an extension code has been received.

14.5.10 Error detection

In I^{2}C bus mode, the status of the serial data bus (SDA0) during data transmission is captured by IIC shift register 0 (IIC0) of the transmitting device, so the IIC0 data prior to transmission can be compared with the transmitted IIC0 data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

14.5.11 Extension code

(1) When the higher 4 bits of the receive address are either “0000” or “1111”, the extension code reception flag (EXC0) is set to 1 for extension code reception and an interrupt request (INTIIC0) is issued at the falling edge of the eighth clock. The local address stored in slave address register 0 (SVA0) is not affected.

(2) If “11110××0” is set to SVA0 by a 10-bit address transfer and “11110××0” is transferred from the master device, the results are as follows. Note that INTIIC0 occurs at the falling edge of the eighth clock.

- Higher four bits of data match: EXC0 = 1
- Seven bits of data match: COI0 = 1

Remark EXC0: Bit 5 of IIC status register 0 (IICS0)
COI0: Bit 4 of IIC status register 0 (IICS0)

(3) Since the processing after the interrupt request occurs differs according to the data that follows the extension code, such processing is performed by software.
If the extension code is received while a slave device is operating, then the slave device is participating in communication even if its address does not match.
For example, after the extension code is received, if you do not wish to operate the target device as a slave device, set bit 6 (LREL0) of the IIC control register 0 (IICC0) to 1 to set the standby mode for the next communication operation.

Table 14-4. Bit Definitions of Main Extension Code

Slave Address	R/W Bit	Description
0 0 0 0 0 0 0	0	General call address
1 1 1 1 0 x x	0	10-bit slave address specification (for address authentication)
1 1 1 1 0 x x	1	10-bit slave address specification (for read command issuance after address match)

Remark For extension codes other than the above, refer to THE I^{2}C-BUS SPECIFICATION published by NXP.

14.5.12 Arbitration

When several master devices simultaneously generate a start condition (when STT0 is set to 1 before STD0 is set to 1), communication among the master devices is performed as the number of clocks are adjusted until the data differs. This kind of operation is called arbitration.

When one of the master devices loses in arbitration, an arbitration loss flag (ALD0) in IIC status register 0 (IICS0) is set (1) via the timing by which the arbitration loss occurred, and the SCL0 and SDA0 lines are both set to high impedance, which releases the bus.

The arbitration loss is detected based on the timing of the next interrupt request (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALD0 = 1 setting that has been made by software.

For details of interrupt request timing, see **14.5.17 Timing of I²C interrupt request (INTIIC0) occurrence**.

Remark STD0: Bit 1 of IIC status register 0 (IICS0)
STT0: Bit 1 of IIC control register 0 (IICC0)

Figure 14-19. Arbitration Timing Example

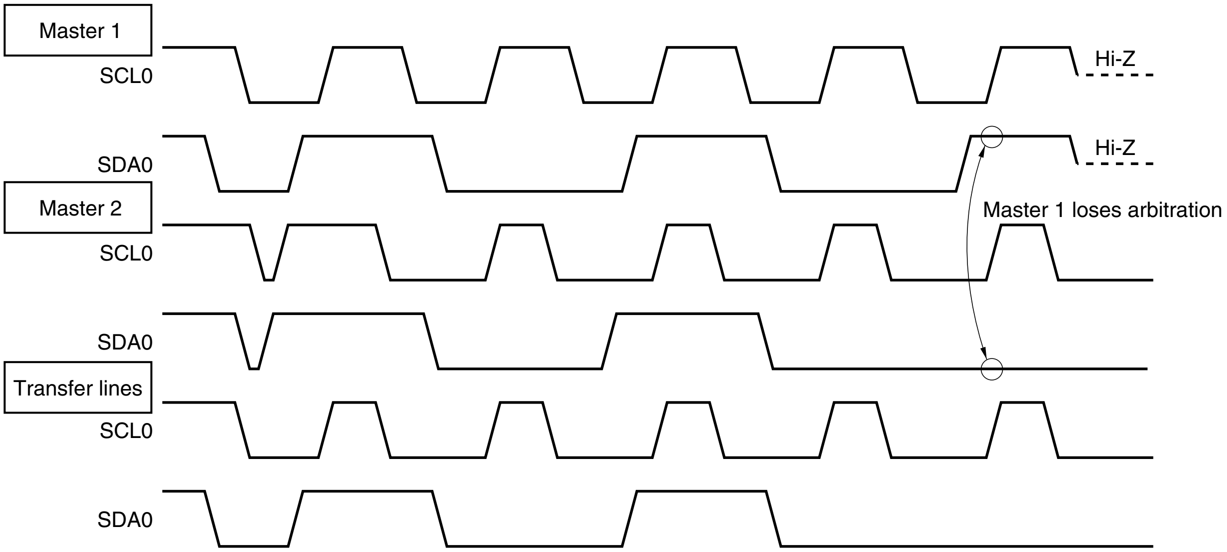


Table 14-5. Status during Arbitration and Interrupt Request Generation Timing

Status During Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer$^{\text{Note 1}}$
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During $\overline{\text{ACK}}$ transfer period after data transmission	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is generated (when SPIE0 = 1)$^{\text{Note 2}}$
When data is at low level while attempting to generate a restart condition	At falling edge of eighth or ninth clock following byte transfer$^{\text{Note 1}}$
When stop condition is detected while attempting to generate a restart condition	When stop condition is generated (when SPIE0 = 1)$^{\text{Note 2}}$
When data is at low level while attempting to generate a stop condition	At falling edge of eighth or ninth clock following byte transfer$^{\text{Note 1}}$
When SCL0 is at low level while attempting to generate a restart condition	

Notes 1. When WTIM0 (bit 3 of IIC control register 0 (IICC0)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIM0 = 0 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock.

2. When there is a chance that arbitration will occur, set SPIE0 = 1 for master device operation.

Remark SPIE0: Bit 4 of IIC control register 0 (IICC0)

14.5.13 Wakeup function

The I^{2}C bus slave function is a function that generates an interrupt request signal (INTIIC0) when a local address and extension code have been received.

This function makes processing more efficient by preventing unnecessary INTIIC0 signal from occurring when addresses do not match.

When a start condition is detected, wakeup standby mode is set. This wakeup standby mode is in effect while addresses are transmitted due to the possibility that an arbitration loss may change the master device (which has generated a start condition) to a slave device.

However, when a stop condition is detected, bit 4 (SPIE0) of IIC control register 0 (IICC0) is set regardless of the wakeup function, and this determines whether interrupt requests are enabled or disabled.

14.5.14 Communication reservation

(1) When communication reservation function is enabled (bit 0 (IICRSV) of IIC flag register 0 (IICF0) = 0)

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{\text{ACK}}$ is not returned and the bus was released when bit 6 (LREL0) of IIC control register 0 (IICC0) was set to 1).

If bit 1 (STT0) of IICC0 is set to 1 while the bus is not used (after a stop condition is detected), a start condition is automatically generated and wait state is set.

If an address is written to IIC shift register 0 (IIC0) after bit 4 (SPIE0) of IICC0 was set to 1, and it was detected by generation of an interrupt request signal (INTIIC0) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to IIC0 before the stop condition is detected is invalid.

When STT0 has been set to 1, the operation mode (as start condition or as communication reservation) is determined according to the bus status.

- If the bus has been released .... a start condition is generated
- If the bus has not been released (standby mode) .... communication reservation

Check whether the communication reservation operates or not by using MSTS0 (bit 7 of IIC status register 0 (IICS0)) after STT0 is set to 1 and the wait time elapses.

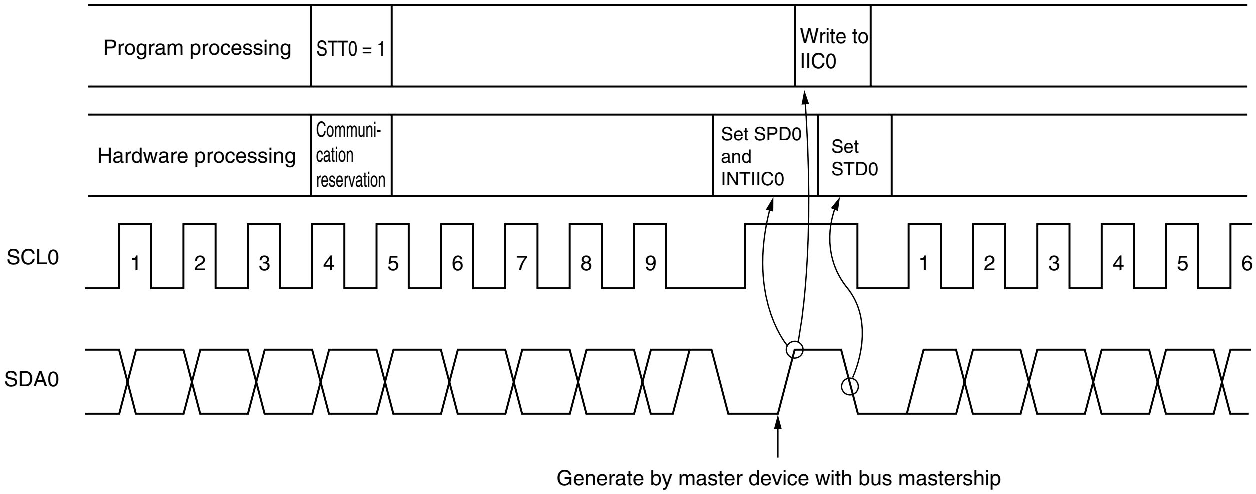
The wait periods, which should be set via software, are listed in **Table 14-6**.

Table 14-6. Wait Periods

CLX0	SMC0	CL01	CL00	Wait Period
0	0	0	0	46 clocks
0	0	0	1	86 clocks
0	0	1	0	172 clocks
0	0	1	1	34 clocks
0	1	0	0	30 clocks
0	1	0	1	
0	1	1	0	60 clocks
0	1	1	1	12 clocks
1	1	0	0	18 clocks
1	1	0	1	
1	1	1	0	36 clocks

Figure 14-20 shows the communication reservation timing.

Figure 14-20. Communication Reservation Timing



Remark IIC0: IIC shift register 0
STT0: Bit 1 of IIC control register 0 (IICC0)
STD0: Bit 1 of IIC status register 0 (IICS0)
SPD0: Bit 0 of IIC status register 0 (IICS0)

Communication reservations are accepted via the following timing. After bit 1 (STD0) of IIC status register 0 (IICS0) is set to 1, a communication reservation can be made by setting bit 1 (STT0) of IIC control register 0 (IICC0) to 1 before a stop condition is detected.

Figure 14-21. Timing for Accepting Communication Reservations

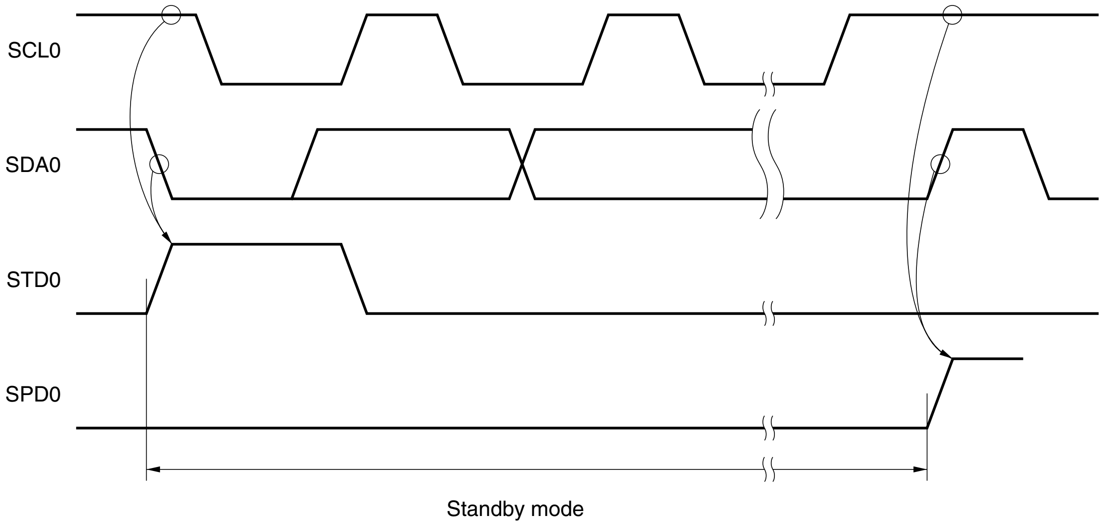
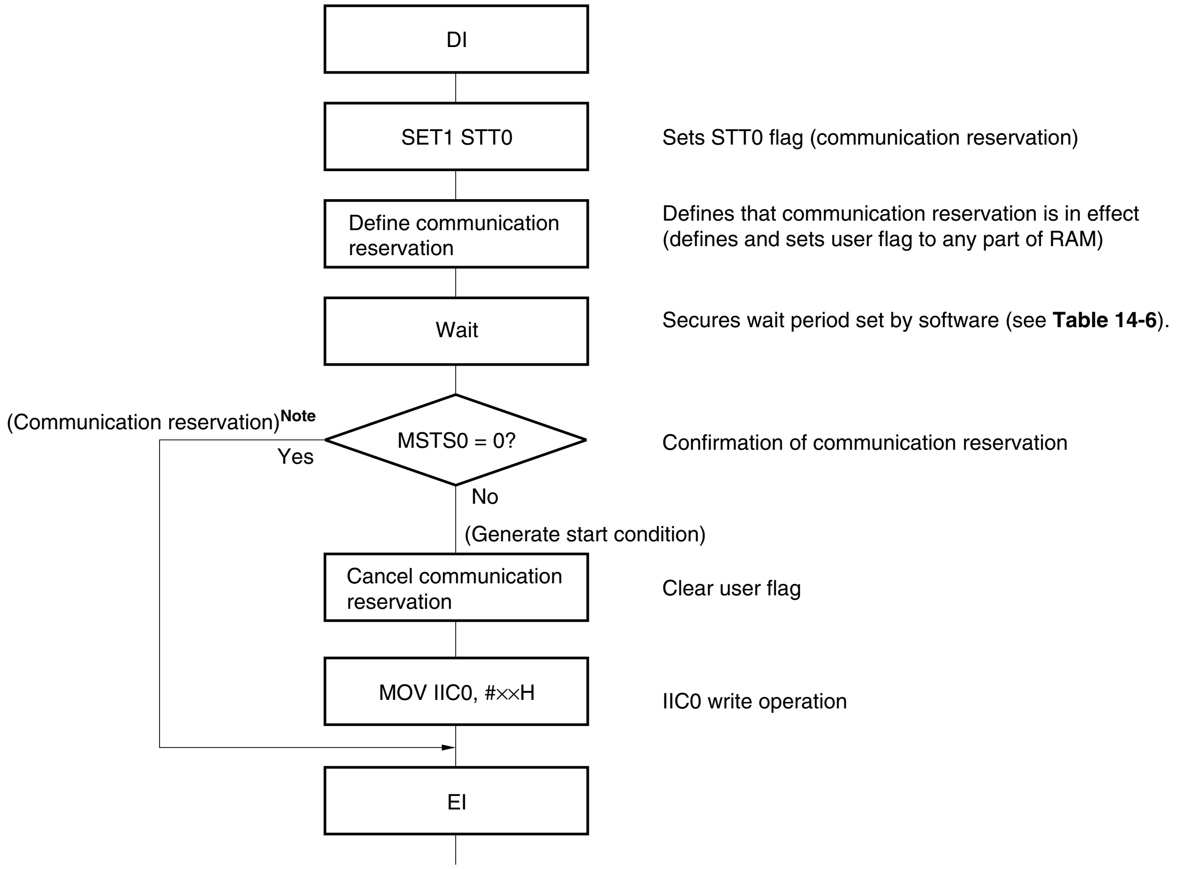


Figure 14-22 shows the communication reservation protocol.

Figure 14-22. Communication Reservation Protocol


DI
SET1 STT0
Sets STT0 flag (communication reservation)
Define communication reservation
Defines that communication reservation is in effect (defines and sets user flag to any part of RAM)
Wait
Secures wait period set by software (see **Table 14-6**).
(Communication reservation)Note
Yes
MSTS0 = 0?
Confirmation of communication reservation
No
(Generate start condition)
Cancel communication reservation
Clear user flag
MOV IIC0, #××H
IIC0 write operation
EI

Note The communication reservation operation executes a write to IIC shift register 0 (IIC0) when a stop condition interrupt request occurs.

Remark		
	STT0:	Bit 1 of IIC control register 0 (IICC0)
	MSTS0:	Bit 7 of IIC status register 0 (IICS0)
	IIC0:	IIC shift register 0

(2) When communication reservation function is disabled (bit 0 (IICRSV) of IIC flag register 0 (IICF0) = 1)

When bit 1 (STT0) of IIC control register 0 (IICC0) is set to 1 when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. The following two statuses are included in the status where bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{\text{ACK}}$ is not returned and the bus was released when bit 6 (LREL0) of IICC0 was set to 1)

To confirm whether the start condition was generated or request was rejected, check STCF (bit 7 of IICF0). The time shown in **Table 14-7** is required until STCF is set to 1 after setting STT0 = 1. Therefore, secure the time by software.

Table 14-7. Wait Periods

CL01	CL00	Wait Period
0	0	6 clocks
0	1	6 clocks
1	0	12 clocks
1	1	3 clocks

14.5.15 Cautions

(1) When STCEN (bit 1 of IIC flag register 0 (IICF0)) = 0

Immediately after I^{2}C operation is enabled (IICE0 = 1), the bus communication status (IICBSY (bit 6 of IICF0) = 1) is recognized regardless of the actual bus status. When changing from a mode in which no stop condition has been detected to a master device communication mode, first generate a stop condition to release the bus, then perform master device communication.

When using multiple masters, it is not possible to perform master device communication when the bus has not been released (when a stop condition has not been detected).

Use the following sequence for generating a stop condition.

<1> Set IIC clock selection register 0 (IICCL0).
<2> Set bit 7 (IICE0) of IIC control register 0 (IICC0) to 1.
<3> Set bit 0 (SPT0) of IICC0 to 1.

(2) When STCEN = 1

Immediately after I^{2}C operation is enabled (IICE0 = 1), the bus released status (IICBSY = 0) is recognized regardless of the actual bus status. To generate the first start condition (STT0 (bit 1 of IIC control register 0 (IICC0)) = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

(3) If other I^{2}C communications are already in progress

If I^{2}C operation is enabled and the device participates in communication already in progress when the SDA0 pin is low and the SCL0 pin is high, the macro of I^{2}C recognizes that the SDA0 pin has gone low (detects a start condition). If the value on the bus at this time can be recognized as an extension code, $\overline{\text{ACK}}$ is returned, but this interferes with other I^{2}C communications. To avoid this, start I^{2}C in the following sequence.

<1> Clear bit 4 (SPIE0) of IICC0 to 0 to disable generation of an interrupt request signal (INTIIC0) when the stop condition is detected.
<2> Set bit 7 (IICE0) of IICC0 to 1 to enable the operation of I^{2}C.
<3> Wait for detection of the start condition.
<4> Set bit 6 (LREL0) of IICC0 to 1 before $\overline{\text{ACK}}$ is returned (4 to 80 clocks after setting IICE0 to 1), to forcibly disable detection.

(4) Determine the transfer clock frequency by using SMC0, CL01, CL00 (bits 3, 1, and 0 of IICL0), and CLX0 (bit 0 of IICX0) before enabling the operation (IICE0 = 1). To change the transfer clock frequency, clear IICE0 to 0 once.

(5) Setting STT0 and SPT0 (bits 1 and 0 of IICC0) again after they are set and before they are cleared to 0 is prohibited.

(6) When transmission is reserved, set SPIE0 (bit 4 of IICL0) to 1 so that an interrupt request is generated when the stop condition is detected. Transfer is started when communication data is written to IIC0 after the interrupt request is generated. Unless the interrupt is generated when the stop condition is detected, the device stops in the wait state because the interrupt request is not generated when communication is started. However, it is not necessary to set SPIE0 to 1 when MSTS0 (bit 7 of IICS0) is detected by software.

14.5.16 Communication operations

The following shows three operation procedures with the flowchart.

(1) Master operation in single master system

The flowchart when using the 78K0/Dx2 microcontrollers as the master in a single master system is shown below.

This flowchart is broadly divided into the initial settings and communication processing. Execute the initial settings at startup. If communication with the slave is required, prepare the communication and then execute communication processing.

(2) Master operation in multimaster system

In the I^{2}C bus multimaster system, whether the bus is released or used cannot be judged by the I^{2}C bus specifications when the bus takes part in a communication. Here, when data and clock are at a high level for a certain period (1 frame), the 78K0/Dx2 microcontrollers takes part in a communication with bus released state.

This flowchart is broadly divided into the initial settings, communication waiting, and communication processing. The processing when the 78K0/Dx2 microcontrollers looses in arbitration and is specified as the slave is omitted here, and only the processing as the master is shown. Execute the initial settings at startup to take part in a communication. Then, wait for the communication request as the master or wait for the specification as the slave. The actual communication is performed in the communication processing, and it supports the transmission/reception with the slave and the arbitration with other masters.

(3) Slave operation

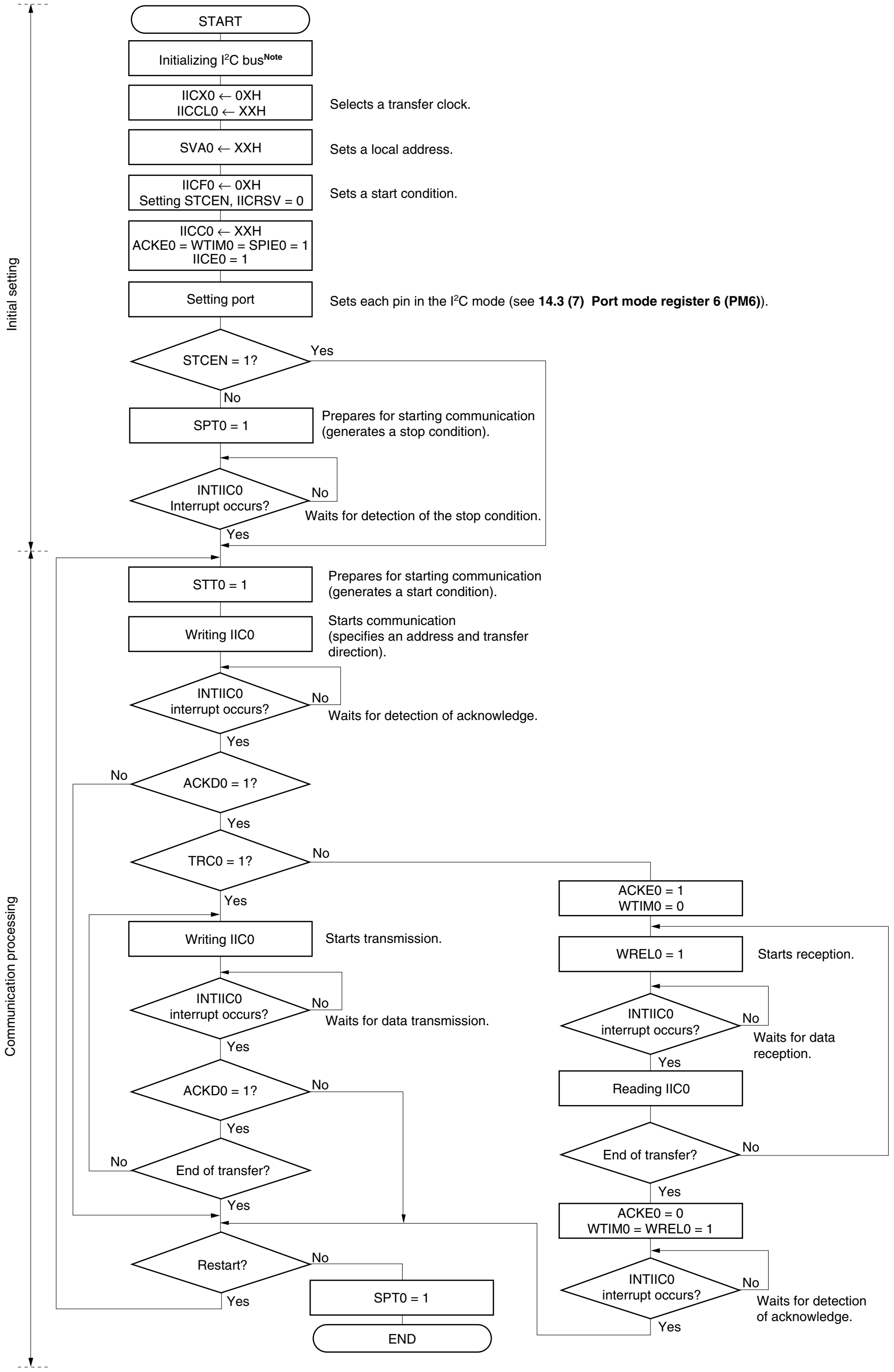
An example of when the 78K0/Dx2 microcontrollers is used as the I^{2}C bus slave is shown below.

When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIIC0 interrupt occurrence (communication waiting). When an INTIIC0 interrupt occurs, the communication status is judged and its result is passed as a flag over to the main processing.

By checking the flags, necessary communication processing is performed.

(1) Master operation in single-master system

Figure 14-23. Master Operation in Single-Master System

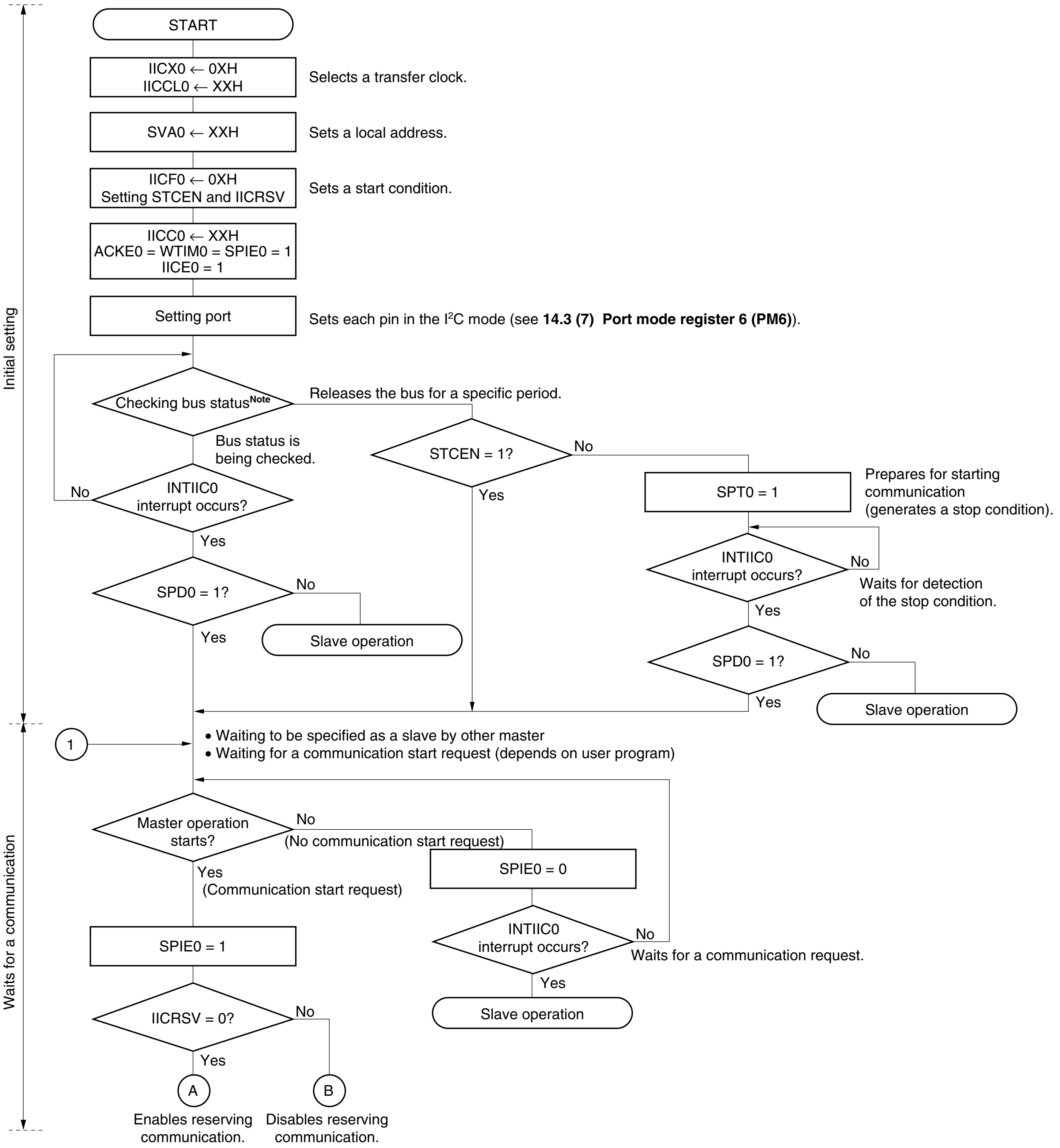


Note Release (SCL0 and SDA0 pins = high level) the I^{2}C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDA0 pin, for example, set the SCL0 pin in the output port mode, and output a clock pulse from the output port until the SDA0 pin is constantly at high level.

Remark Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

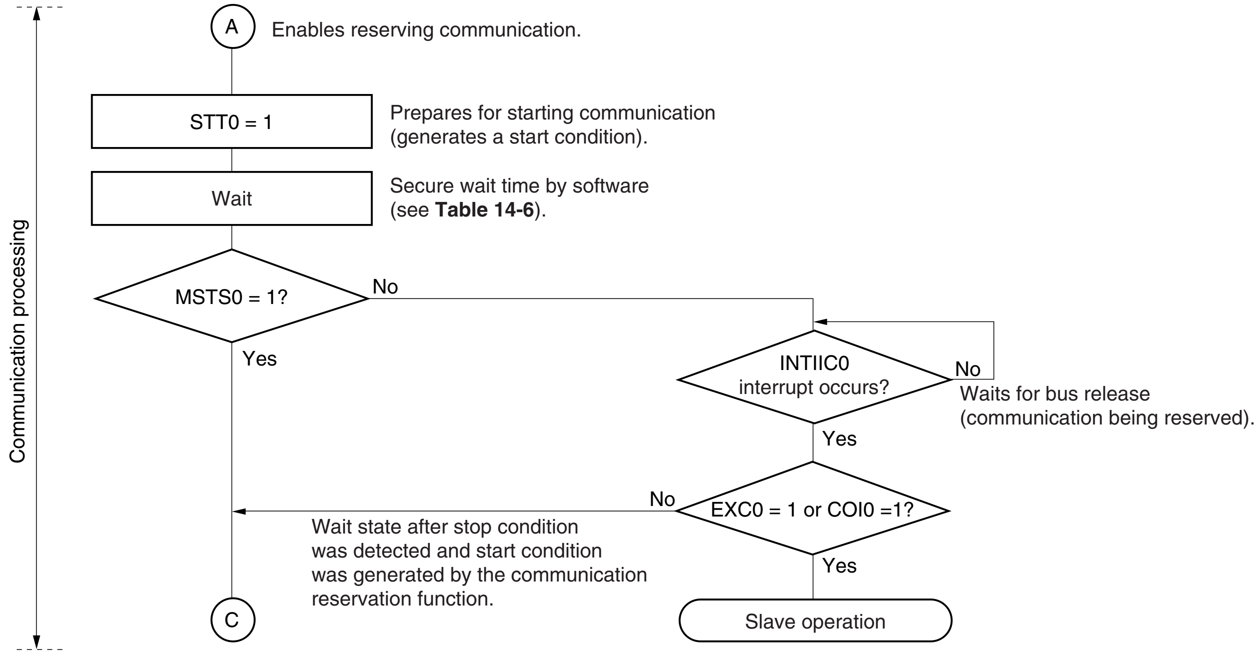
(2) Master operation in multi-master system

Figure 14-24. Master Operation in Multi-Master System (1/3)


Initial setting
START
IICX0 ← 0XH
IICCL0 ← XXH
Selects a transfer clock.
SVA0 ← XXH
Sets a local address.
IICF0 ← 0XH
Setting STCEN and IICRSV
Sets a start condition.
IICC0 ← XXH
ACKE0 = WTIM0 = SPIE0 = 1
IICE0 = 1
Setting port
Sets each pin in the I²C mode (see 14.3 (7) Port mode register 6 (PM6)).
Checking bus status Note
Releases the bus for a specific period.
Bus status is being checked.
STCEN = 1?
No
Yes
SPT0 = 1
Prepares for starting communication (generates a stop condition).
INTIIC0 interrupt occurs?
No
Yes
Waits for detection of the stop condition.
SPD0 = 1?
No
Yes
Slave operation
INTIIC0 interrupt occurs?
No
Yes
SPD0 = 1?
No
Yes
Slave operation
Waits for a communication
1
• Waiting to be specified as a slave by other master
• Waiting for a communication start request (depends on user program)
Master operation starts?
No
(No communication start request)
SPIE0 = 0
Yes
(Communication start request)
SPIE0 = 1
INTIIC0 interrupt occurs?
No
Waits for a communication request.
Yes
Slave operation
IICRSV = 0?
No
Yes
A
B
Enables reserving communication.
Disables reserving communication.

Note Confirm that the bus is released (CLD0 bit = 1, DAD0 bit = 1) for a specific period (for example, for a period of one frame). If the SDA0 pin is constantly at low level, decide whether to release the I²C bus (SCL0 and SDA0 pins = high level) in conformance with the specifications of the product that is communicating.

Figure 14-24. Master Operation in Multi-Master System (2/3)


Communication processing
A
Enables reserving communication.
STT0 = 1
Prepares for starting communication
(generates a start condition).
Wait
Secure wait time by software
(see Table 14-6).
MSTS0 = 1?
No
Yes
INTIIC0
interrupt occurs?
No
Waits for bus release
(communication being reserved).
Yes
No
EXC0 = 1 or COI0 =1?
Wait state after stop condition
was detected and start condition
was generated by the communication
reservation function.
Yes
C
Slave operation

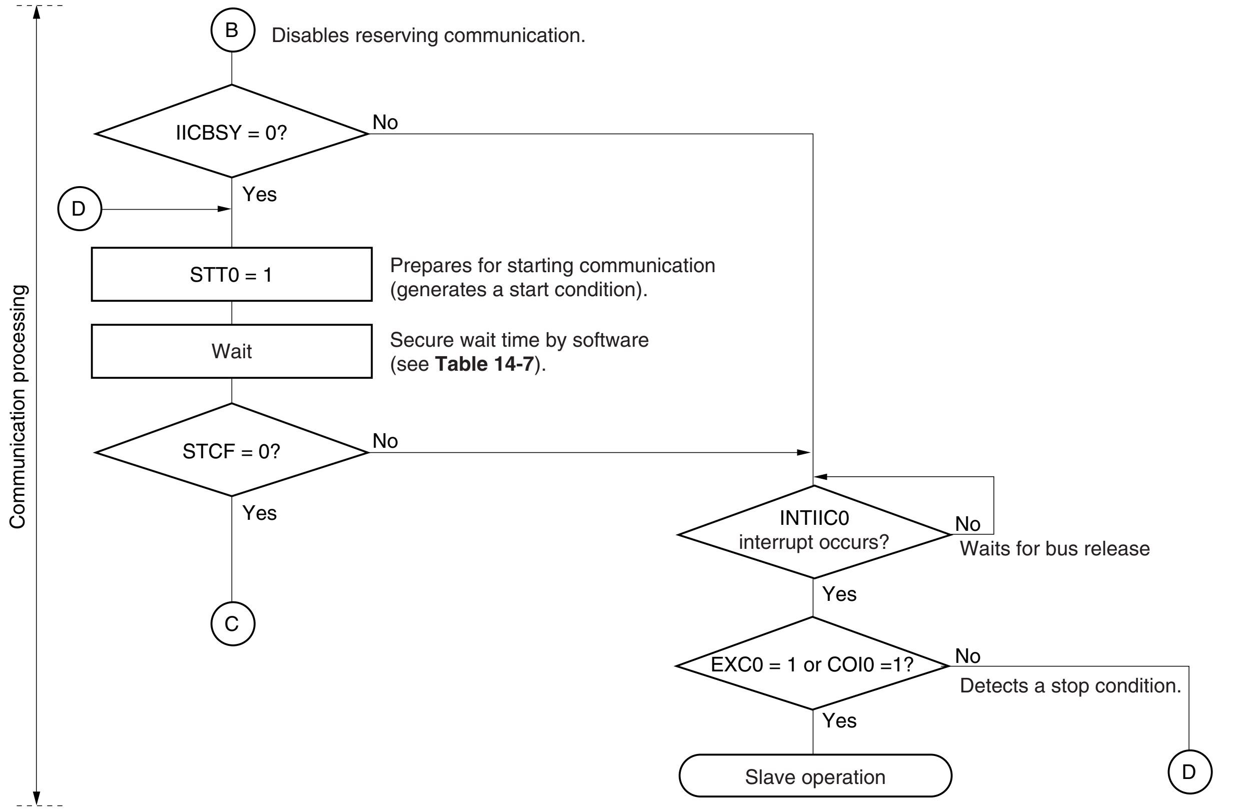
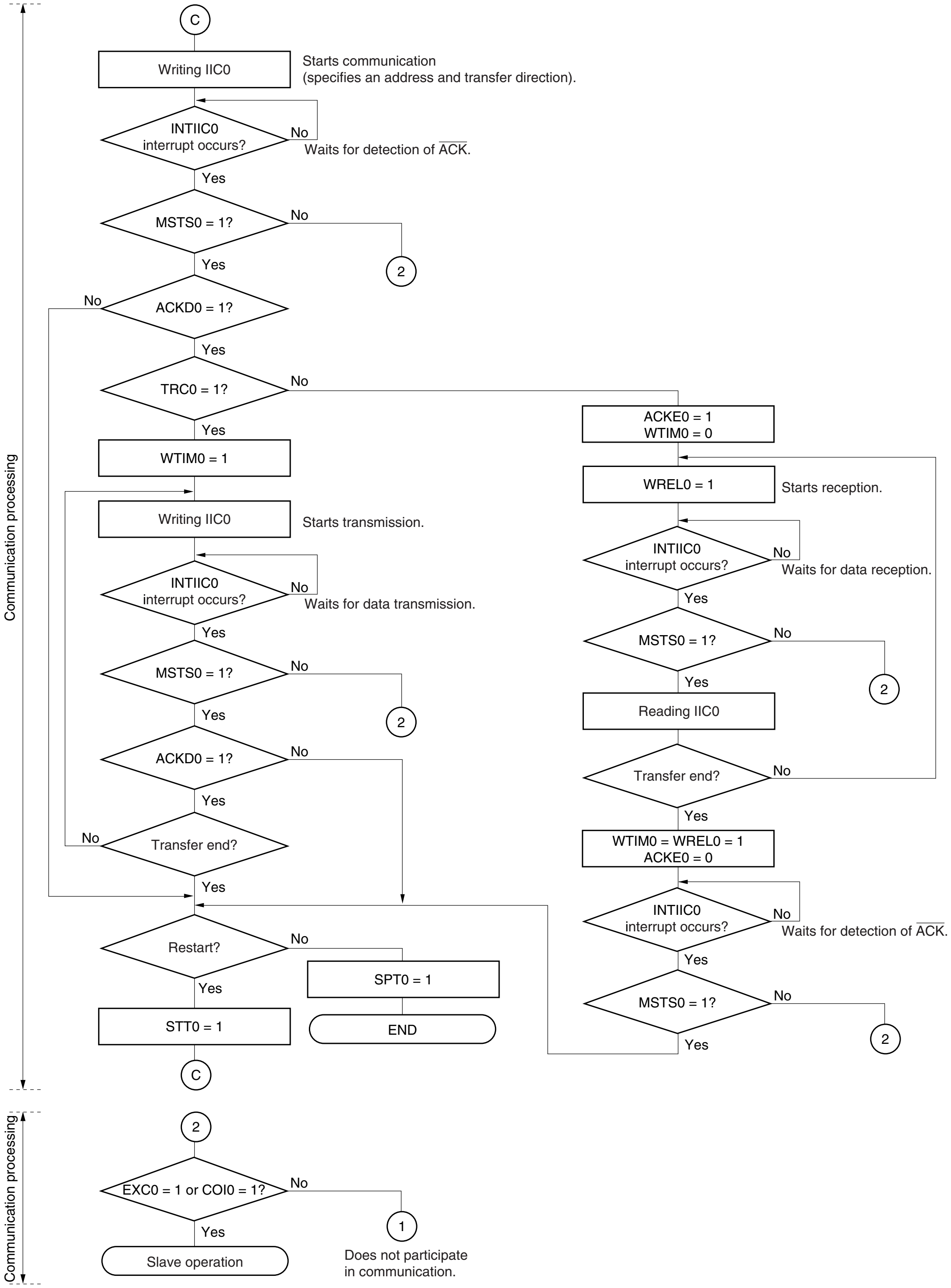


Figure 14-24. Master Operation in Multi-Master System (3/3)



Remarks 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

2. To use the device as a master in a multi-master system, read the MSTS0 bit each time interrupt INTIIC0 has occurred to check the arbitration result.

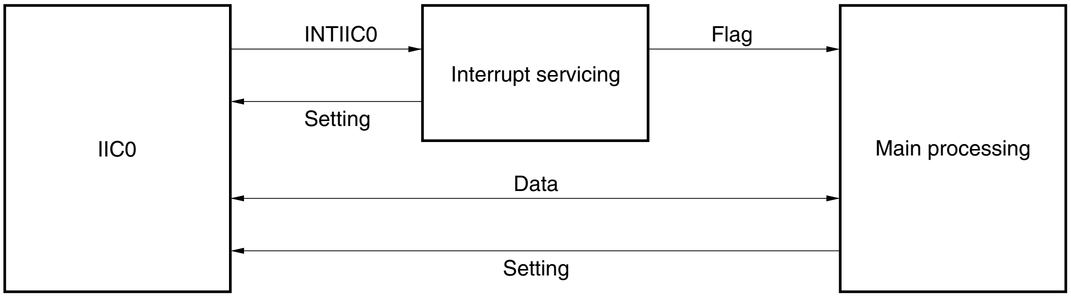
3. To use the device as a slave in a multi-master system, check the status by using the IICS0 and IICF0 registers each time interrupt INTIIC0 has occurred, and determine the processing to be performed next.

(3) Slave operation

The processing procedure of the slave operation is as follows.

Basically, the slave operation is event-driven. Therefore, processing by the INTIIC0 interrupt (processing that must substantially change the operation status such as detection of a stop condition during communication) is necessary.

In the following explanation, it is assumed that the extension code is not supported for data communication. It is also assumed that the INTIIC0 interrupt servicing only performs status transition processing, and that actual data communication is performed by the main processing.



Therefore, data communication processing is performed by preparing the following three flags and passing them to the main processing instead of INTIIC0.

<1> Communication mode flag

This flag indicates the following two communication statuses.

- Clear mode: Status in which data communication is not performed
- Communication mode: Status in which data communication is performed (from valid address detection to stop condition detection, no detection of $\overline{\text{ACK}}$ from master, address mismatch)

<2> Ready flag

This flag indicates that data communication is enabled. Its function is the same as the INTIIC0 interrupt for ordinary data communication. This flag is set by interrupt servicing and cleared by the main processing. Clear this flag by interrupt servicing when communication is started. However, the ready flag is not set by interrupt servicing when the first data is transmitted. Therefore, the first data is transmitted without the flag being cleared (an address match is interpreted as a request for the next data).

<3> Communication direction flag

This flag indicates the direction of communication. Its value is the same as TRC0.

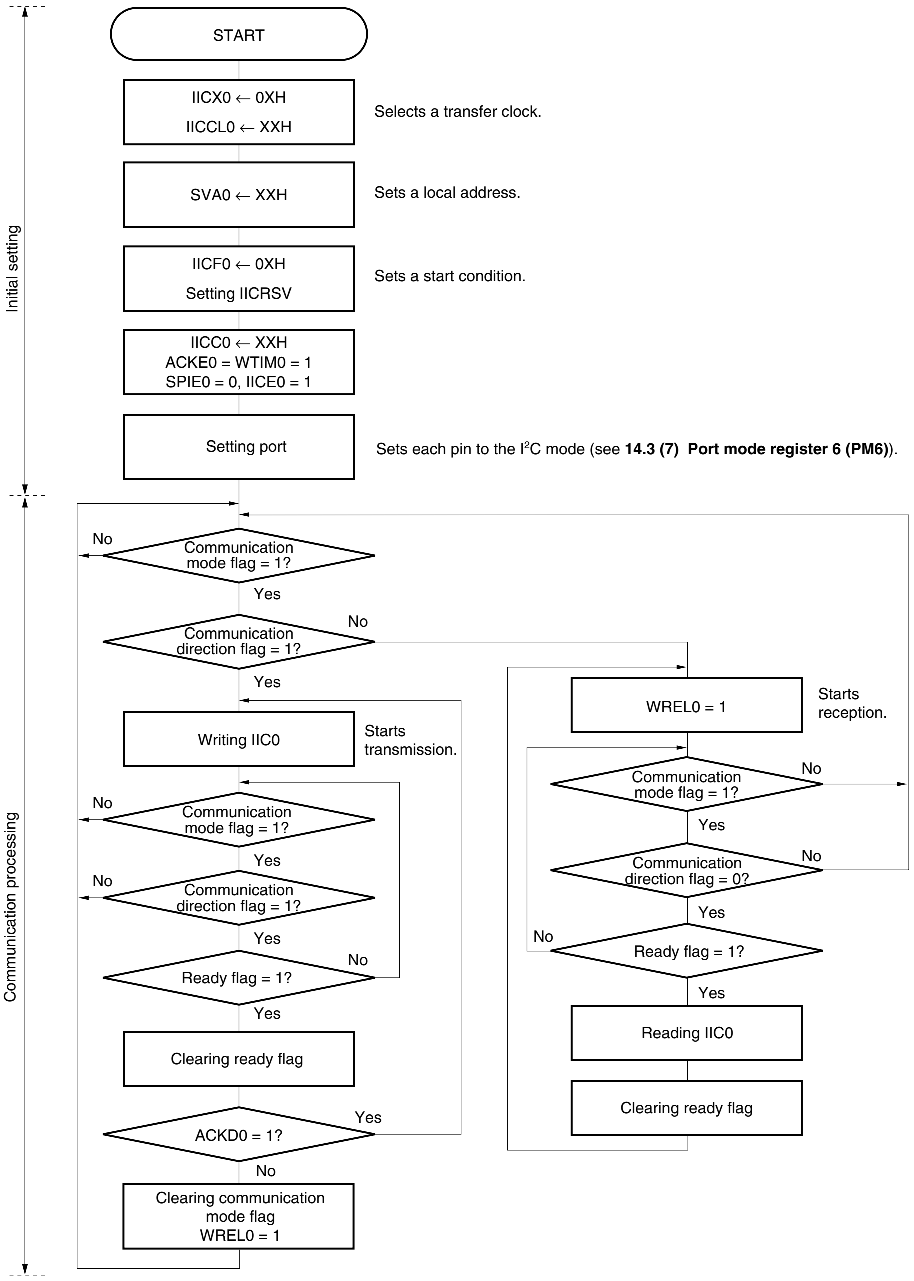
The main processing of the slave operation is explained next.

Start serial interface IIC0 and wait until communication is enabled. When communication is enabled, execute communication by using the communication mode flag and ready flag (processing of the stop condition and start condition is performed by an interrupt. Here, check the status by using the flags).

The transmission operation is repeated until the master no longer returns ACK. If $\overline{\text{ACK}}$ is $\overline{\text{not}}$ returned from the master, communication is completed.

For reception, the necessary amount of data is received. When communication is completed, ACK is $\overline{\text{not}}$ returned as the next data. After that, the master generates a stop condition or restart condition. Exit from the communication status occurs in this way.

Figure 14-25. Slave Operation Flowchart (1)



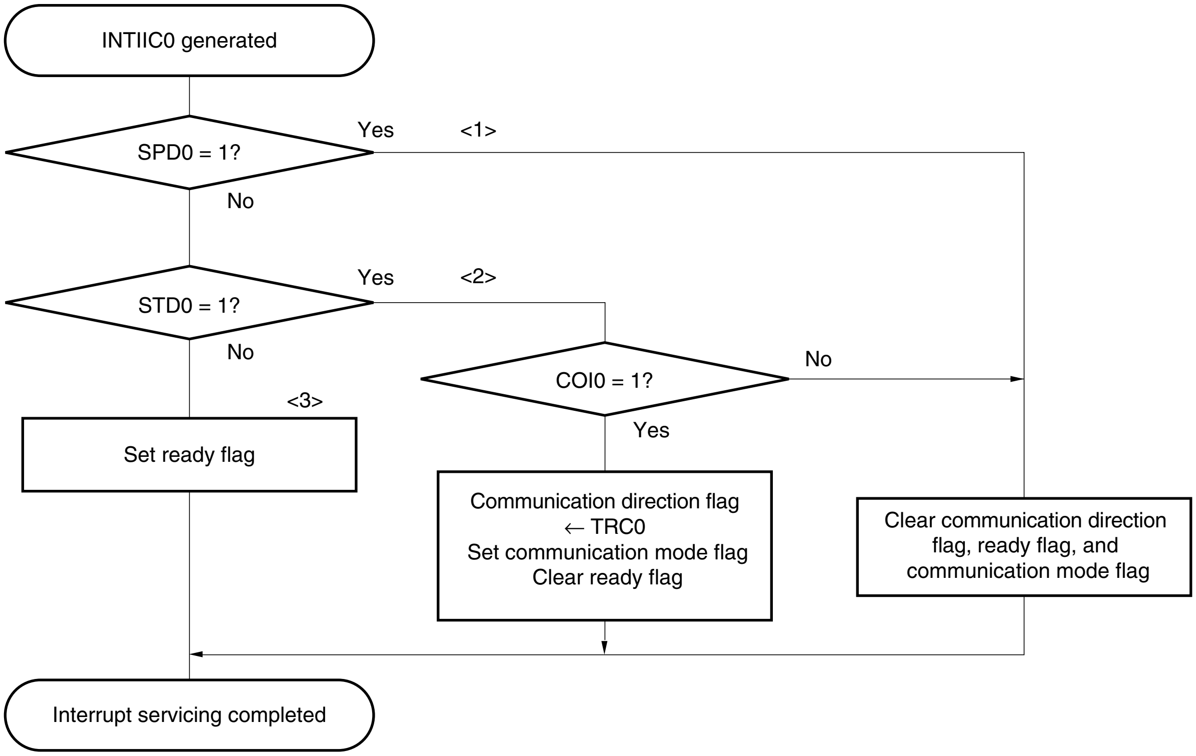
Remark Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.

An example of the processing procedure of the slave with the INTIIC0 interrupt is explained below (processing is performed assuming that no extension code is used). The INTIIC0 interrupt checks the status, and the following operations are performed.

<1> Communication is stopped if the stop condition is issued.
<2> If the start condition is issued, the address is checked and communication is completed if the address does not match. If the address matches, the communication mode is set, wait is cancelled, and processing returns from the interrupt (the ready flag is cleared).
<3> For data transmit/receive, only the ready flag is set. Processing returns from the interrupt with the I^{2}C bus remaining in the wait state.

Remark <1> to <3> above correspond to <1> to <3> in **Figure 14-26 Slave Operation Flowchart (2)**.

Figure 14-26. Slave Operation Flowchart (2)



14.5.17 Timing of I^{2}C interrupt request (INTIIC0) occurrence

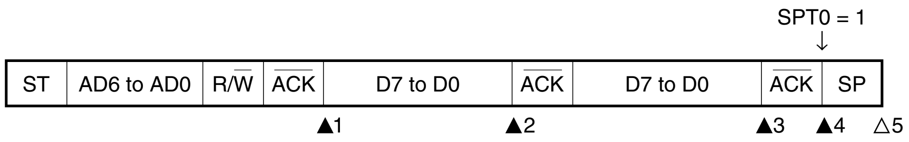
The timing of transmitting or receiving data and generation of interrupt request signal INTIIC0, and the value of the IICS0 register when the INTIIC0 signal is generated are shown below.

Remark		
	ST:	Start condition
	AD6 to AD0:	Address
	R/$\overline{\text{W}}$:	Transfer direction specification
	$\overline{\text{ACK}}$:	Acknowledge
	D7 to D0:	Data
	SP:	Stop condition

(1) Master device operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (transmission/reception)

(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B

▲2: IICS0 = 1000×000B

▲3: IICS0 = 1000×000B (Sets WTIM0 to 1[Note])

▲4: IICS0 = 1000××00B (Sets SPT0 to 1)

△5: IICS0 = 00000001B

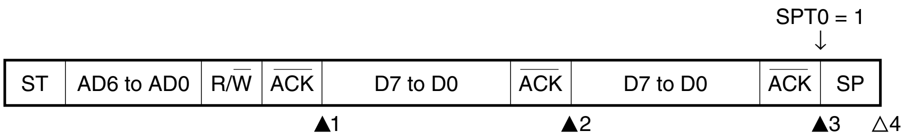
Note To generate a stop condition, set WTIM0 to 1 and change the timing for generating the INTIIC0 interrupt request signal.

Remark ▲: Always generated

△: Generated only when SPIE0 = 1

×: Don't care

(ii) When WTIM0 = 1



▲1: IICS0 = 1000×110B

▲2: IICS0 = 1000×100B

▲3: IICS0 = 1000××00B (Sets SPT0 to 1)

△4: IICS0 = 00000001B

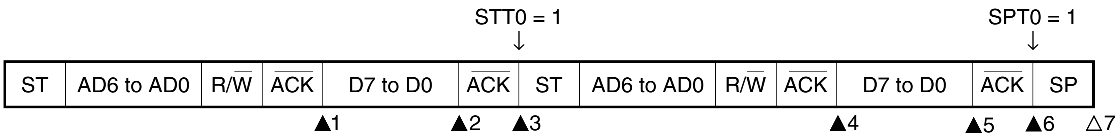
Remark ▲: Always generated

△: Generated only when SPIE0 = 1

×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B

▲2: IICS0 = 1000×000B (Sets WTIM0 to 1[Note 1])

▲3: IICS0 = 1000××00B (Clears WTIM0 to 0[Note 2], sets STT0 to 1)

▲4: IICS0 = 1000×110B

▲5: IICS0 = 1000×000B (Sets WTIM0 to 1[Note 3])

▲6: IICS0 = 1000××00B (Sets SPT0 to 1)

△7: IICS0 = 00000001B

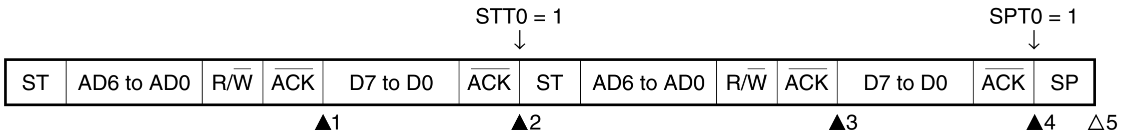
Notes 1. To generate a start condition, set WTIM0 to 1 and change the timing for generating the INTIIC0 interrupt request signal.

2. Clear WTIM0 to 0 to restore the original setting.

3. To generate a stop condition, set WTIM0 to 1 and change the timing for generating the INTIIC0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



▲1: IICS0 = 1000×110B

▲2: IICS0 = 1000××00B (Sets STT0 to 1)

▲3: IICS0 = 1000×110B

▲4: IICS0 = 1000××00B (Sets SPT0 to 1)

△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

(i) When WTIM0 = 0

								SPT0 = 1 ↓	
ST	AD6 to AD0	$R/\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP	
			▲1	▲2		▲3	▲4		△5

▲1: IICS0 = 1010×110B

▲2: IICS0 = 1010×000B

▲3: IICS0 = 1010×000B (Sets WTIM0 to 1[Note])

▲4: IICS0 = 1010××00B (Sets SPT0 to 1)

△5: IICS0 = 00000001B

Note To generate a stop condition, set WTIM0 to 1 and change the timing for generating the INTIIC0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(ii) When WTIM0 = 1

								SPT0 = 1 ↓	
ST	AD6 to AD0	$R/\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP	
			▲1		▲2		▲3		△4

▲1: IICS0 = 1010×110B

▲2: IICS0 = 1010×100B

▲3: IICS0 = 1010××00B (Sets SPT0 to 1)

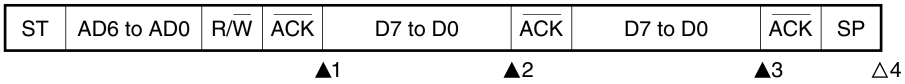
△4: IICS0 = 00001001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(2) Slave device operation (slave address data reception)

(a) Start ~ Address ~ Data ~ Data ~ Stop

(i) When WTIM0 = 0



▲1: IICS0 = 0001×110B

▲2: IICS0 = 0001×000B

▲3: IICS0 = 0001×000B

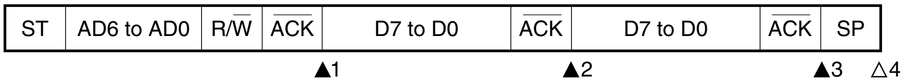
△4: IICS0 = 00000001B

Remark ▲: Always generated

△: Generated only when SPIE0 = 1

×: Don't care

(ii) When WTIM0 = 1



▲1: IICS0 = 0001×110B

▲2: IICS0 = 0001×100B

▲3: IICS0 = 0001××00B

△4: IICS0 = 00000001B

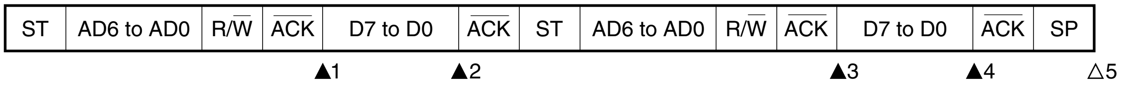
Remark ▲: Always generated

△: Generated only when SPIE0 = 1

×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, matches with SVA0)



▲1: IICS0 = 0001×110B

▲2: IICS0 = 0001×000B

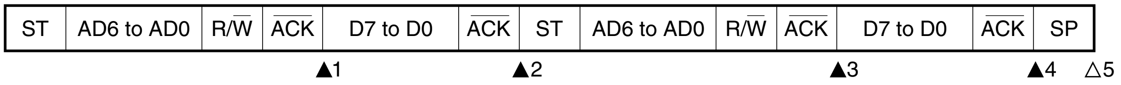
▲3: IICS0 = 0001×110B

▲4: IICS0 = 0001×000B

△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(ii) When WTIM0 = 1 (after restart, matches with SVA0)



▲1: IICS0 = 0001×110B

▲2: IICS0 = 0001××00B

▲3: IICS0 = 0001×110B

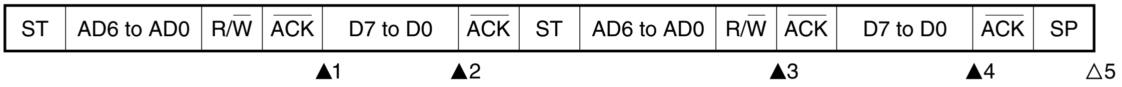
▲4: IICS0 = 0001××00B

△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, does not match address (= extension code))



▲1: IICS0 = 0001×110B

▲2: IICS0 = 0001×000B

▲3: IICS0 = 0010×010B

▲4: IICS0 = 0010×000B

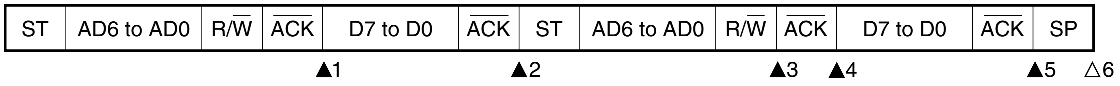
△5: IICS0 = 00000001B

Remark ▲: Always generated

△: Generated only when SPIE0 = 1

×: Don't care

(ii) When WTIM0 = 1 (after restart, does not match address (= extension code))



▲1: IICS0 = 0001×110B

▲2: IICS0 = 0001××00B

▲3: IICS0 = 0010×010B

▲4: IICS0 = 0010×110B

▲5: IICS0 = 0010××00B

△6: IICS0 = 00000001B

Remark ▲: Always generated

△: Generated only when SPIE0 = 1

×: Don't care

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, does not match address (= not extension code))

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
			▲1	▲2					▲3			△4

▲1: IICS0 = 0001×110B

▲2: IICS0 = 0001×000B

▲3: IICS0 = 00000110B

△4: IICS0 = 00000001B

Remark ▲: Always generated

△: Generated only when SPIE0 = 1

×: Don’t care

(ii) When WTIM0 = 1 (after restart, does not match address (= not extension code))

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
			▲1		▲2				▲3			△4

▲1: IICS0 = 0001×110B

▲2: IICS0 = 0001××00B

▲3: IICS0 = 00000110B

△4: IICS0 = 00000001B

Remark ▲: Always generated

△: Generated only when SPIE0 = 1

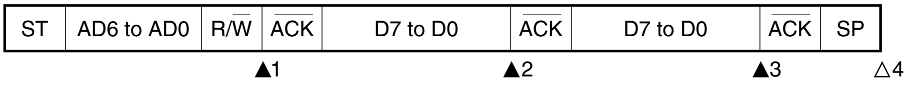
×: Don’t care

(3) Slave device operation (when receiving extension code)

The device is always participating in communication when it receives an extension code.

(a) Start ~ Code ~ Data ~ Data ~ Stop

(i) When WTIM0 = 0



▲1: IICS0 = 0010×010B

▲2: IICS0 = 0010×000B

▲3: IICS0 = 0010×000B

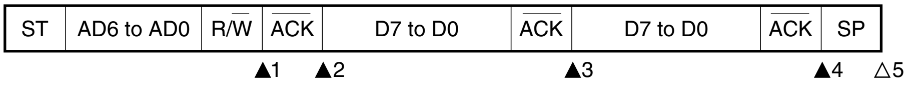
△4: IICS0 = 00000001B

Remark ▲: Always generated

△: Generated only when SPIE0 = 1

×: Don’t care

(ii) When WTIM0 = 1



▲1: IICS0 = 0010×010B

▲2: IICS0 = 0010×110B

▲3: IICS0 = 0010×100B

▲4: IICS0 = 0010××00B

△5: IICS0 = 00000001B

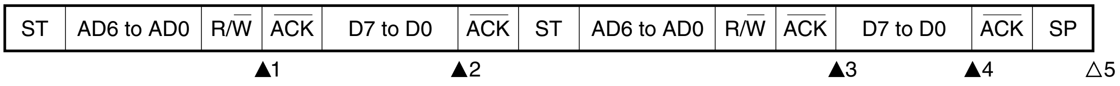
Remark ▲: Always generated

△: Generated only when SPIE0 = 1

×: Don’t care

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, matches SVA0)



▲1: IICS0 = 0010×010B

▲2: IICS0 = 0010×000B

▲3: IICS0 = 0001×110B

▲4: IICS0 = 0001×000B

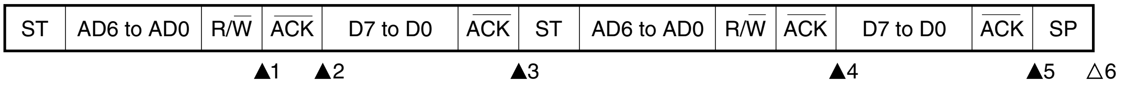
△5: IICS0 = 00000001B

Remark ▲: Always generated

△: Generated only when SPIE0 = 1

×: Don't care

(ii) When WTIM0 = 1 (after restart, matches SVA0)



▲1: IICS0 = 0010×010B

▲2: IICS0 = 0010×110B

▲3: IICS0 = 0010××00B

▲4: IICS0 = 0001×110B

▲5: IICS0 = 0001××00B

△6: IICS0 = 00000001B

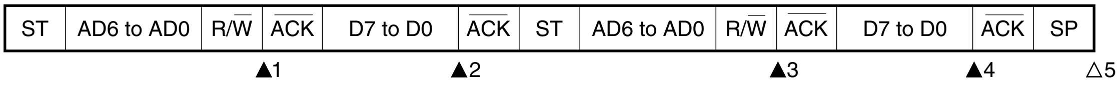
Remark ▲: Always generated

△: Generated only when SPIE0 = 1

×: Don't care

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

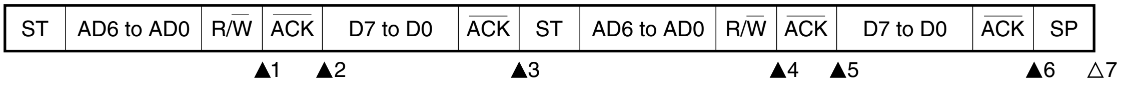
(i) When WTIM0 = 0 (after restart, extension code reception)



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, extension code reception)



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 0010×010B
▲5: IICS0 = 0010×110B
▲6: IICS0 = 0010××00B
△7: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, does not match address (= not extension code))

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP	
		▲1		▲2					▲3				△4

▲1: IICS0 = 00100010B

▲2: IICS0 = 00100000B

▲3: IICS0 = 00000110B

△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(ii) When WTIM0 = 1 (after restart, does not match address (= not extension code))

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP	
		▲1	▲2		▲3				▲4				△5

▲1: IICS0 = 00100010B

▲2: IICS0 = 00100110B

▲3: IICS0 = 00100×00B

▲4: IICS0 = 00000110B

△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
								△1

△1: IICS0 = 00000001B

Remark △: Generated only when SPIE0 = 1

(5) Arbitration loss operation (operation as slave after arbitration loss)

When the device is used as a master in a multi-master system, read the MSTS0 bit each time interrupt request signal INTIIC0 has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data

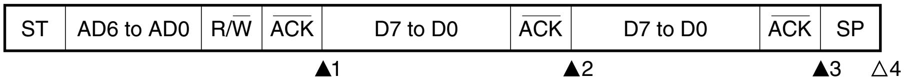
(i) When WTIM0 = 0

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
			▲1	▲2		▲3		△4

▲1: IICS0 = 0101×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(ii) When WTIM0 = 1

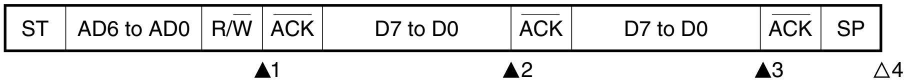


▲1: IICS0 = 0101×110B
▲2: IICS0 = 0001×100B
▲3: IICS0 = 0001××00B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(b) When arbitration loss occurs during transmission of extension code

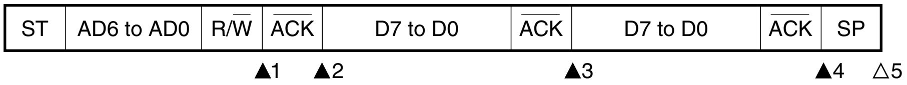
(i) When WTIM0 = 0



▲1: IICS0 = 0110×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(ii) When WTIM0 = 1



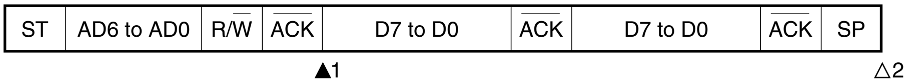
▲1: IICS0 = 0110×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010×100B
▲4: IICS0 = 0010××00B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(6) Operation when arbitration loss occurs (no communication after arbitration loss)

When the device is used as a master in a multi-master system, read the MSTS0 bit each time interrupt request signal INTIIC0 has occurred to check the arbitration result.

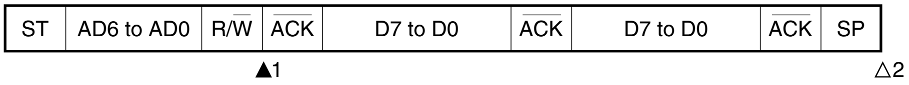
(a) When arbitration loss occurs during transmission of slave address data (when WTIM0 = 1)



▲1: IICS0 = 01000110B
△2: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(b) When arbitration loss occurs during transmission of extension code

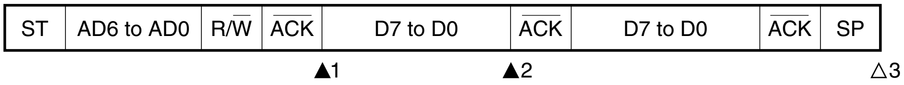


▲1: IICS0 = 0110×010B
Sets LREL0 = 1 by software
△2: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) When arbitration loss occurs during transmission of data

(i) When WTIM0 = 0



▲1: IICS0 = 10001110B
▲2: IICS0 = 01000000B
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(ii) When WTIM0 = 1

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
			▲1		▲2			△3

▲1: IICS0 = 10001110B
▲2: IICS0 = 01000100B
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(d) When loss occurs due to restart condition during data transfer

(i) Not extension code (Example: unmatches with SVA0)

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to Dn	ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP
			▲1					▲2			△3

▲1: IICS0 = 1000×110B
▲2: IICS0 = 01000110B
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care
n = 6 to 0

(ii) Extension code

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to Dn	ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP	
			▲1					▲2				△3

▲1: IICS0 = 1000×110B

▲2: IICS0 = 01100010B

Sets LREL0 = 1 by software

△3: IICS0 = 00000001B

Remark ▲: Always generated

△: Generated only when SPIE0 = 1

×: Don't care

n = 6 to 0

(e) When loss occurs due to stop condition during data transfer

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to Dn	SP	
			▲1			△2

▲1: IICS0 = 10000110B

△2: IICS0 = 01000001B

Remark ▲: Always generated

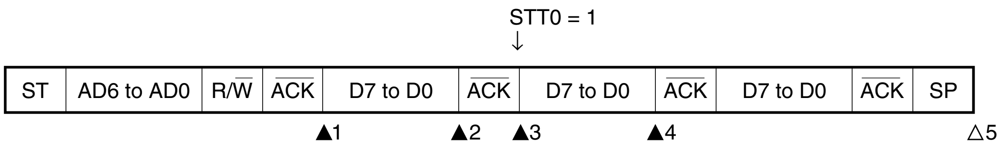
△: Generated only when SPIE0 = 1

×: Don't care

n = 6 to 0

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

(i) When WTIM0 = 0


STT0 = 1
↓
ST
AD6 to AD0
R/$\overline{\mathrm{W}}$
$\overline{\mathrm{ACK}}$
D7 to D0
$\overline{\mathrm{ACK}}$
D7 to D0
$\overline{\mathrm{ACK}}$
D7 to D0
$\overline{\mathrm{ACK}}$
SP
▲1
▲2
▲3
▲4
△5

▲1: IICS0 = 1000×110B

▲2: IICS0 = 1000×000B (Sets WTIM0 to 1)

▲3: IICS0 = 1000×100B (Clears WTIM0 to 0)

▲4: IICS0 = 01000000B

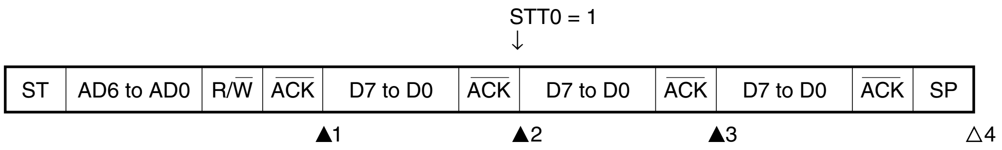
△5: IICS0 = 00000001B

Remark ▲: Always generated

△: Generated only when SPIE0 = 1

×: Don’t care

(ii) When WTIM0 = 1


STT0 = 1
↓
ST
AD6 to AD0
R/$\overline{\mathrm{W}}$
$\overline{\mathrm{ACK}}$
D7 to D0
$\overline{\mathrm{ACK}}$
D7 to D0
$\overline{\mathrm{ACK}}$
D7 to D0
$\overline{\mathrm{ACK}}$
SP
▲1
▲2
▲3
△4

▲1: IICS0 = 1000×110B

▲2: IICS0 = 1000×100B (Sets STT0 to 1)

▲3: IICS0 = 01000100B

△4: IICS0 = 00000001B

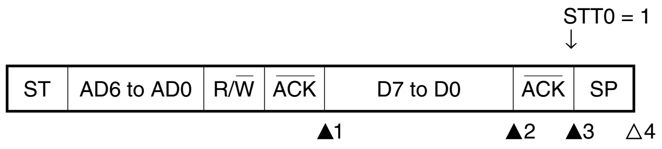
Remark ▲: Always generated

△: Generated only when SPIE0 = 1

×: Don’t care

(g) **When arbitration loss occurs due to a stop condition when attempting to generate a restart condition**

(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B

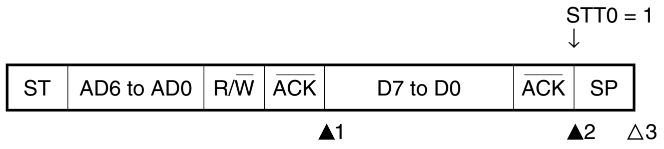
▲2: IICS0 = 1000×000B (Sets WTIM0 to 1)

▲3: IICS0 = 1000××00B (Sets STT0 to 1)

△4: IICS0 = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



▲1: IICS0 = 1000×110B

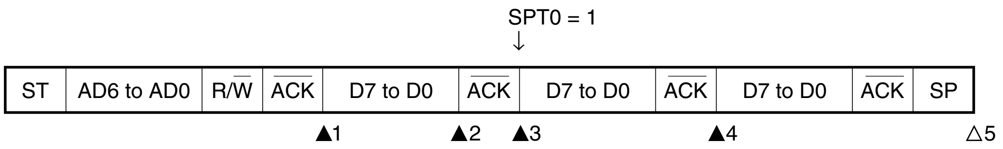
▲2: IICS0 = 1000××00B (Sets STT0 to 1)

△3: IICS0 = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B

▲2: IICS0 = 1000×000B (Sets WTIM0 to 1)

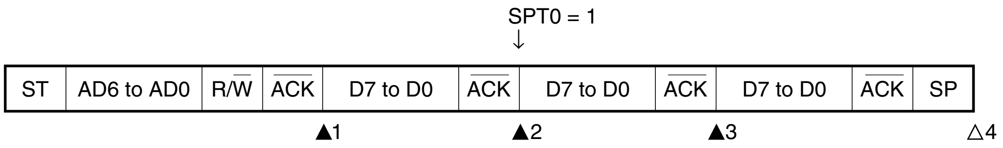
▲3: IICS0 = 1000×100B (Clears WTIM0 to 0)

▲4: IICS0 = 01000100B

△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(ii) When WTIM0 = 1



▲1: IICS0 = 1000×110B

▲2: IICS0 = 1000×100B (Sets SPT0 to 1)

▲3: IICS0 = 01000100B

△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

14.6 Timing Charts

When using the I^2C bus mode, the master device outputs an address via the serial bus to select one of several slave devices as its communication partner.

After outputting the slave address, the master device transmits the TRC0 bit (bit 3 of IIC status register 0 (IICS0)), which specifies the data transfer direction, and then starts serial communication with the slave device.

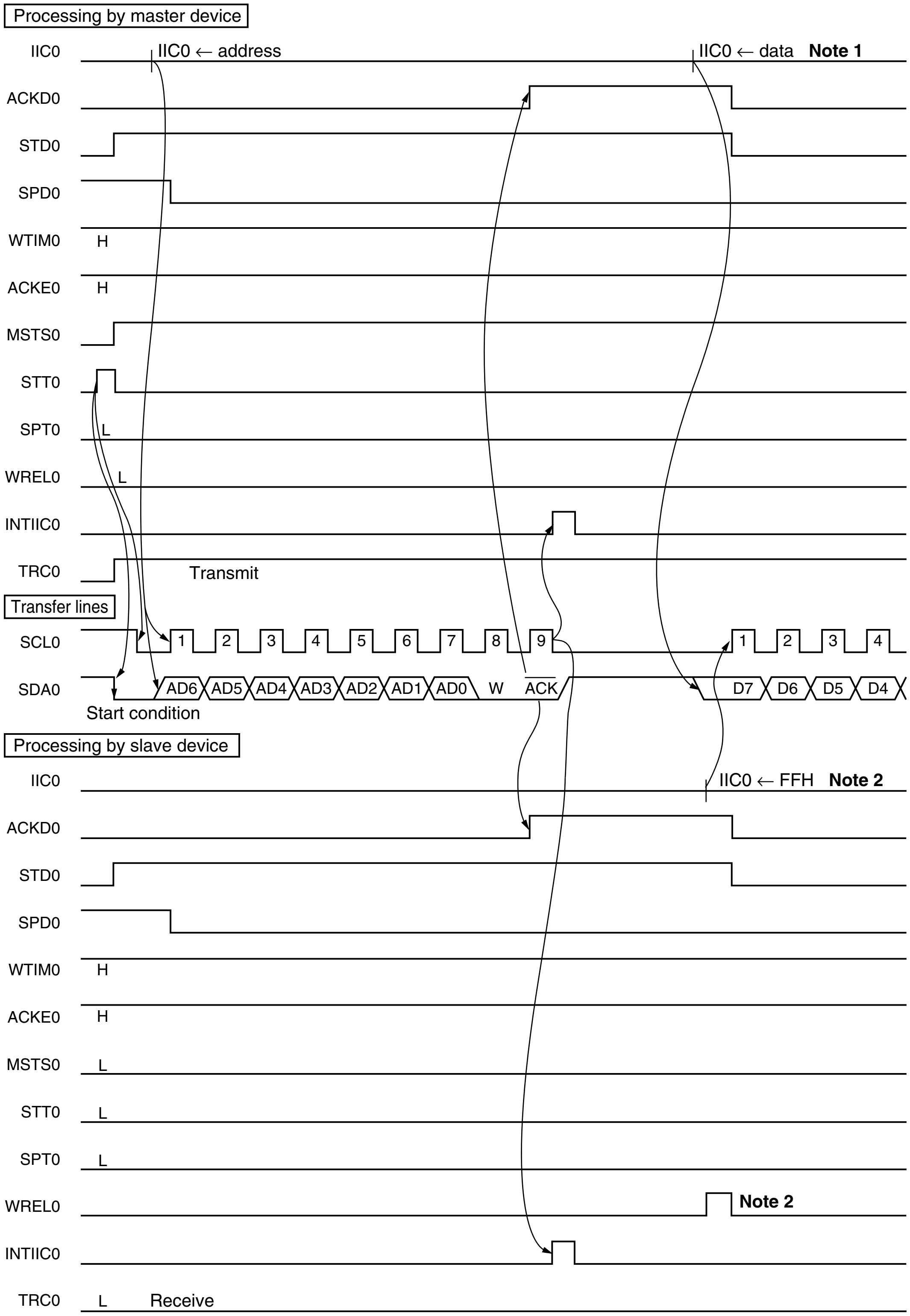
Figures 14-27 and **14-28** show timing charts of the data communication.

IIC shift register 0 (IIC0)’s shift operation is synchronized with the falling edge of the serial clock (SCL0). The transmit data is transferred to the SO0 latch and is output (MSB first) via the SDA0 pin.

Data input via the SDA0 pin is captured into IIC0 at the rising edge of SCL0.

Figure 14-27. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)

(1) Start condition ~ address

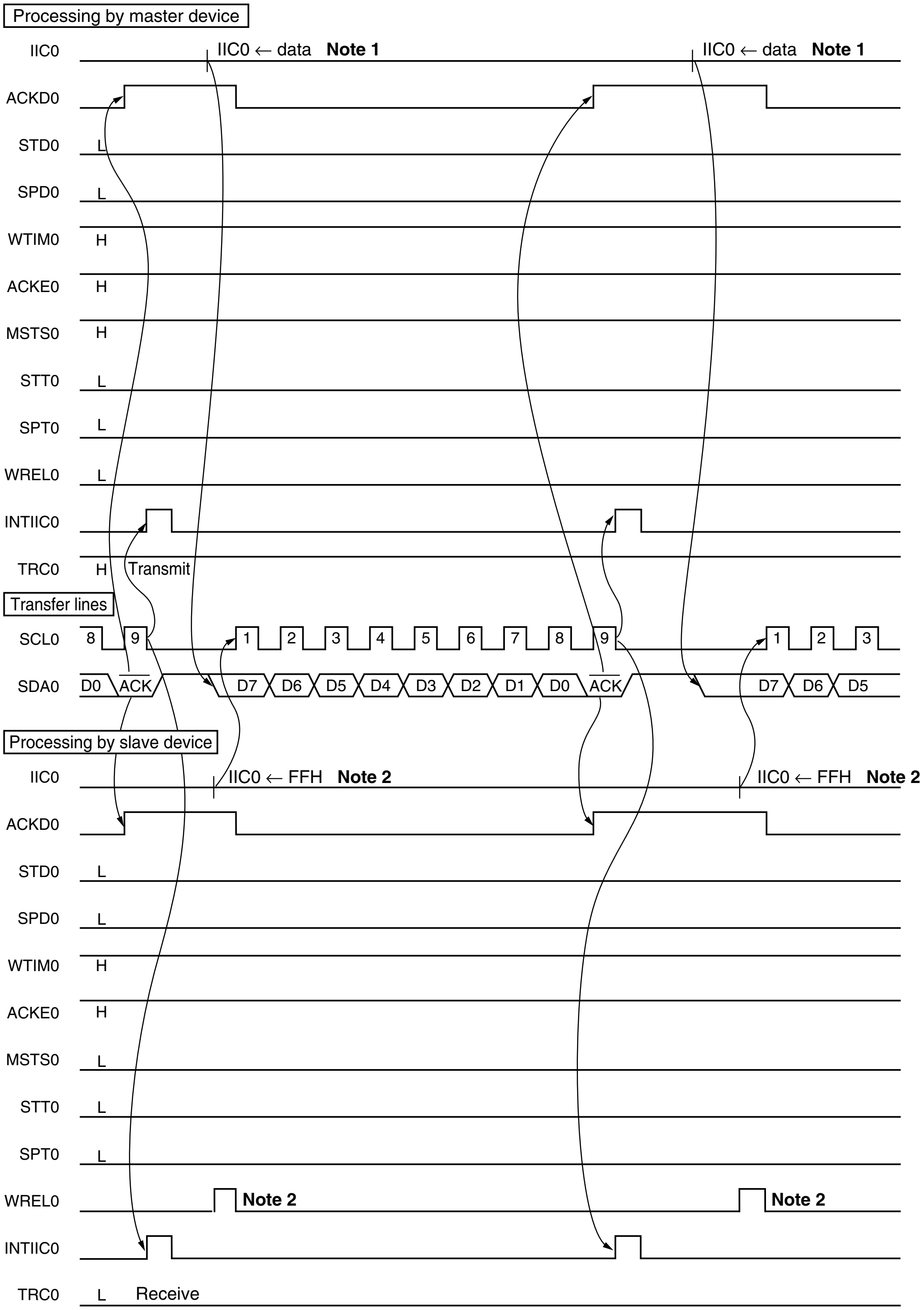


Notes 1. Write data to IIC0, not setting WREL0, in order to cancel a wait state during master transmission.

2. To cancel slave wait, write “FFH” to IIC0 or set WREL0.

Figure 14-27. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)

(2) Data

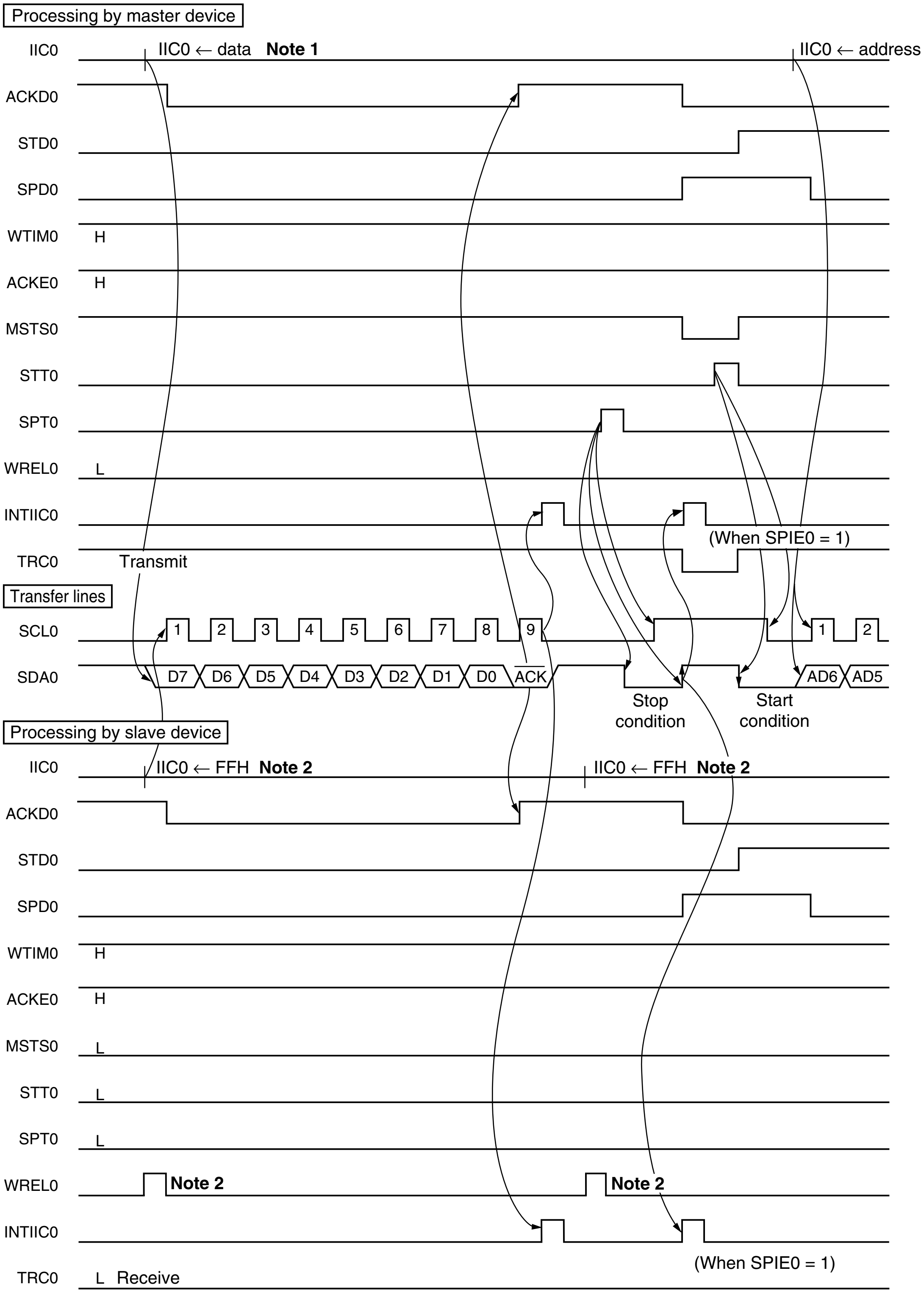


Notes 1. Write data to IIC0, not setting WREL0, in order to cancel a wait state during master transmission.

2. To cancel slave wait, write “FFH” to IIC0 or set WREL0.

Figure 14-27. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)

(3) Stop condition

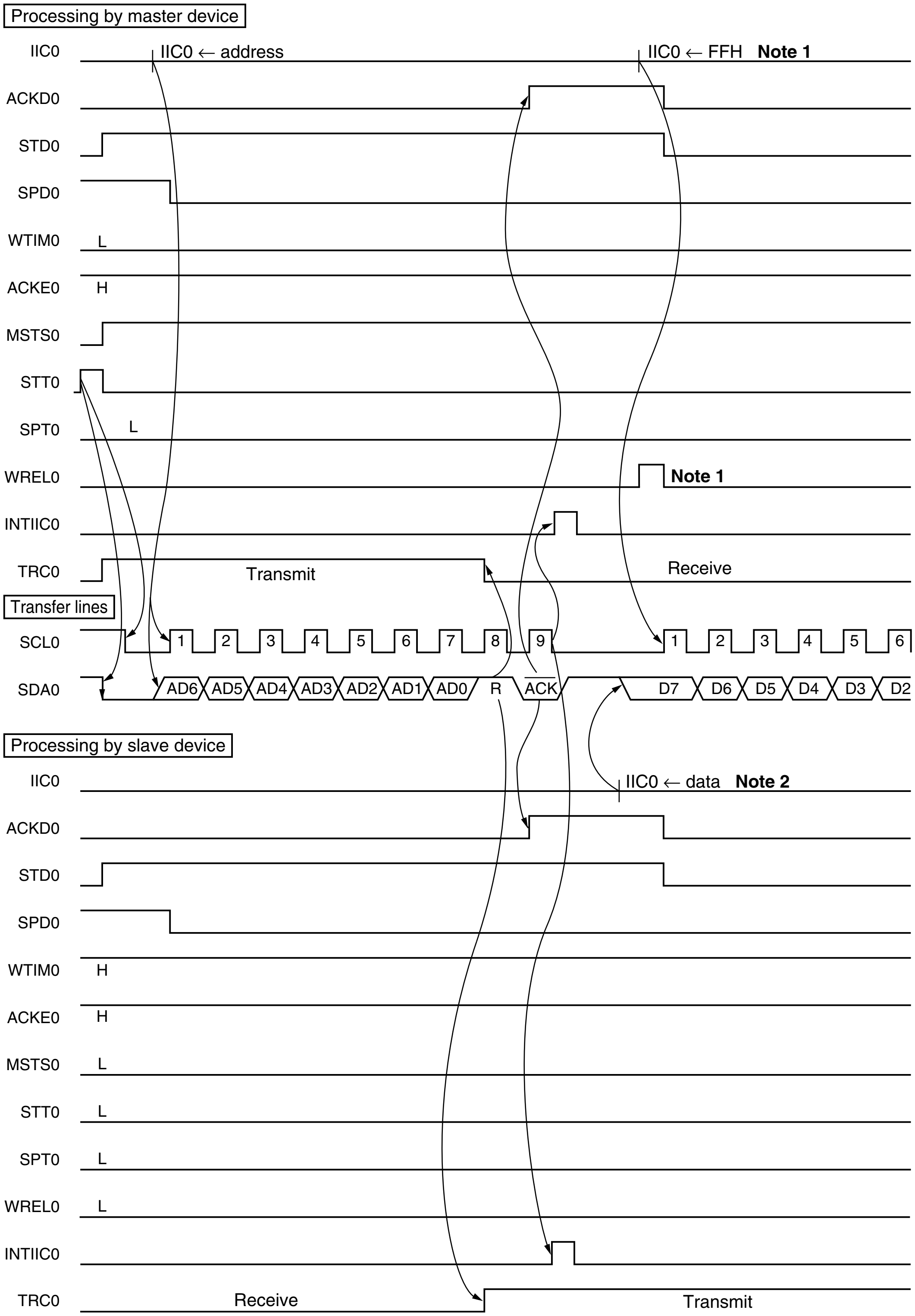


Notes 1. Write data to IIC0, not setting WREL0, in order to cancel a wait state during master transmission.

2. To cancel slave wait, write “FFH” to IIC0 or set WREL0.

Figure 14-28. Example of Slave to Master Communication (When 8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (1/3)

(1) Start condition ~ address

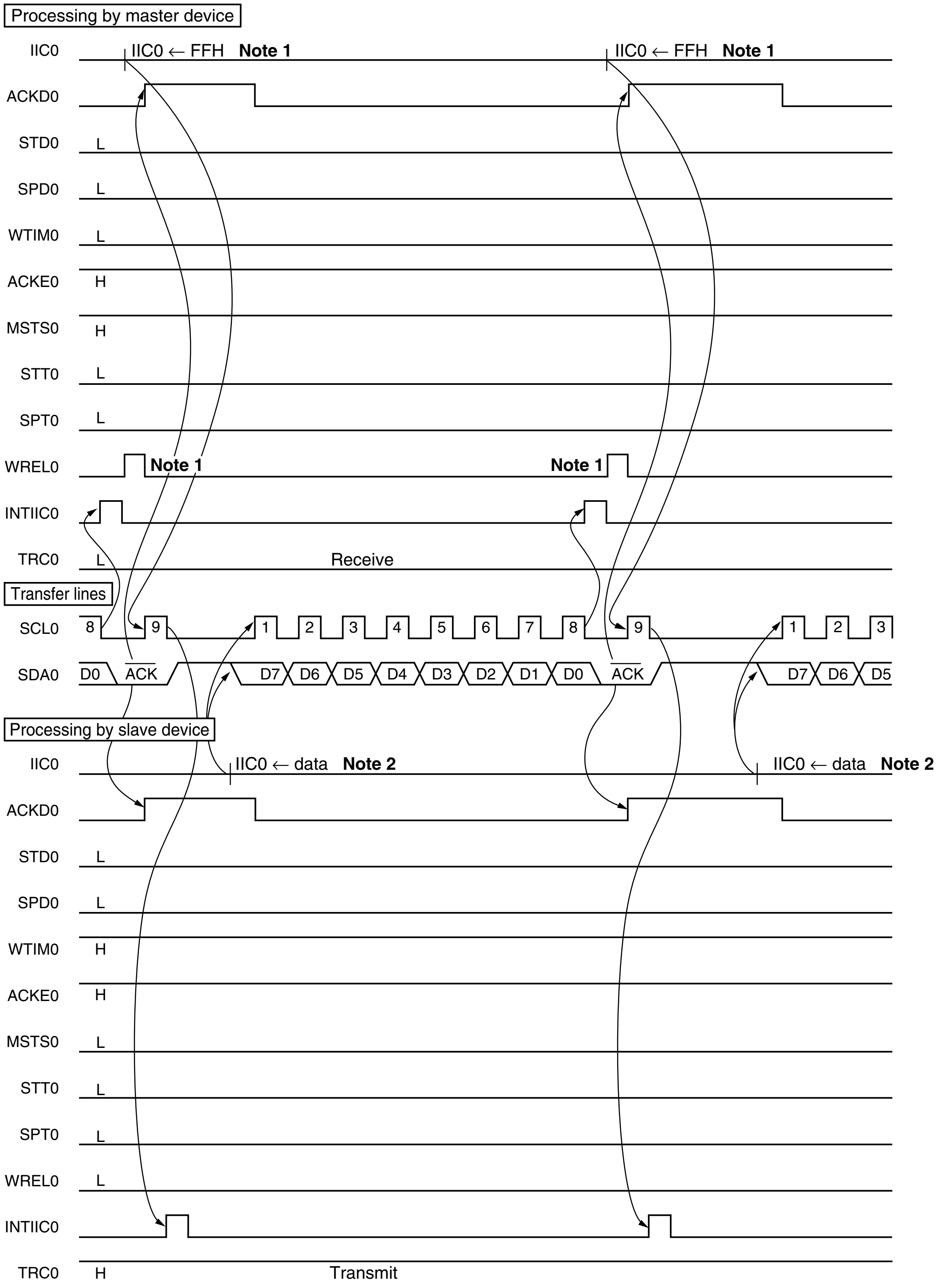


Notes 1. To cancel master wait, write "FFH" to IIC0 or set WREL0.

2. Write data to IIC0, not setting WREL0, in order to cancel a wait state during slave transmission.

Figure 14-28. Example of Slave to Master Communication (When 8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (2/3)

(2) Data

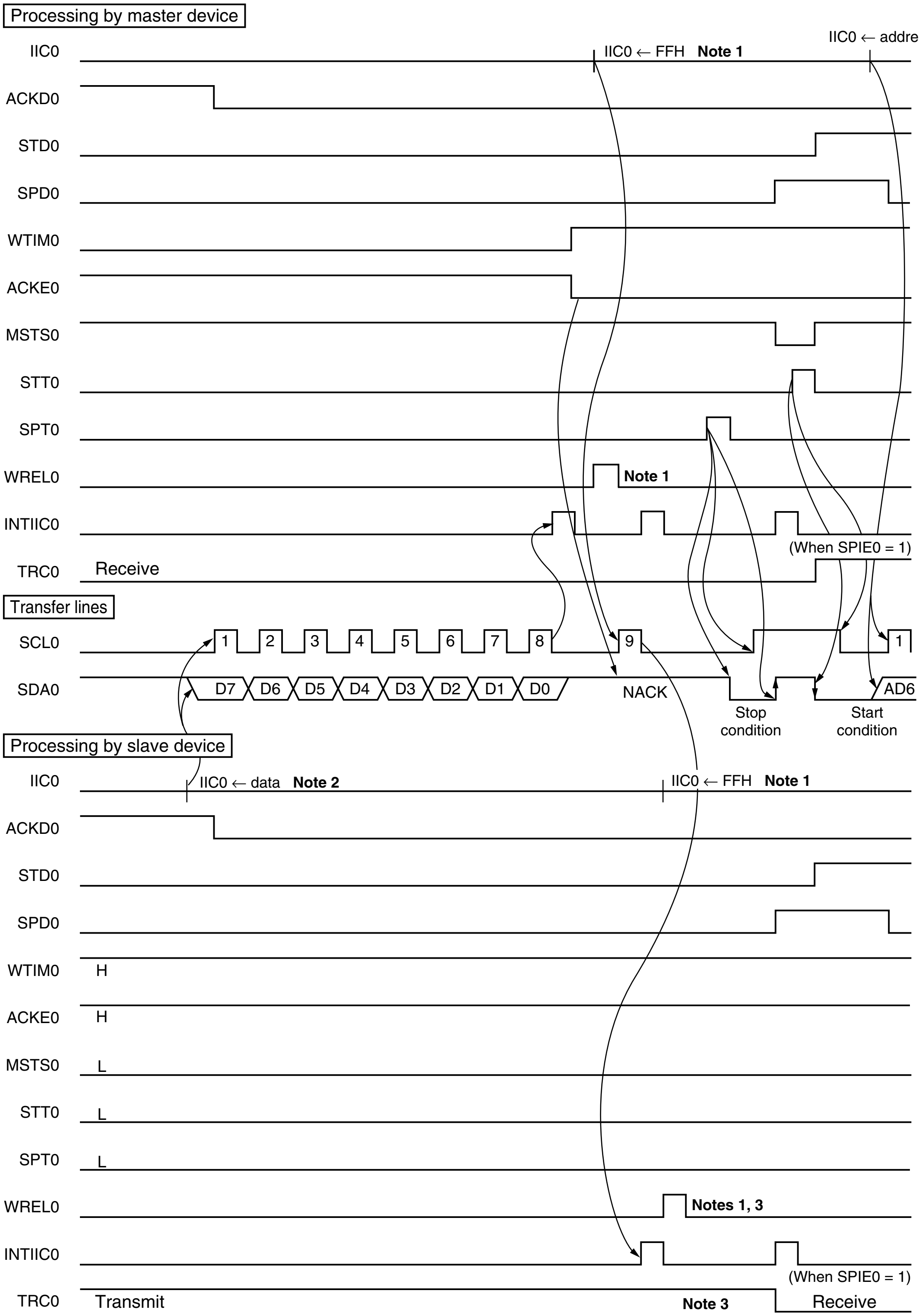


Notes 1. To cancel master wait, write “FFH” to IIC0 or set WREL0.

2. Write data to IIC0, not setting WREL0, in order to cancel a wait state during slave transmission.

Figure 14-28. Example of Slave to Master Communication (When 8-Clock and 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (3/3)

(3) Stop condition



Notes 1. To cancel wait, write “FFH” to IIC0 or set WREL0.

2. Write data to IIC0, not setting WREL0, in order to cancel a wait state during slave transmission.

3. If a wait state during slave transmission is canceled by setting WREL0, TRC0 will be cleared.

CHAPTER 15 CAN CONTROLLER

15.1 Outline Description

This product features an on-chip 1-channel CAN (Controller Area Network) controller that complies with CAN protocol as standardized in ISO 11898.

15.1.1 Features

- Compliant with ISO 11898 and tested according to ISO/DIS 16845 (CAN conformance test)
- Standard frame and extended frame transmission/reception enabled
- Transfer rate: 1 Mbps max. (CAN clock input ≥ 8 MHz)
- 16 message buffers/1 channel
- Receive/transmit history list function
- Automatic block transmission function
- Multi-buffer receive block function
- Mask setting of four patterns is possible for each channel

15.1.2 Overview of functions

Table 15-1 presents an overview of the CAN controller functions.

Table 15-1. Overview of Functions

Function	Details
Protocol	CAN protocol ISO 11898 (standard and extended frame transmission/reception)
Baud rate	Maximum 1 Mbps (CAN clock input ≥ 8 MHz)
Data storage	Storing messages in the CAN RAM
Number of messages	- 16 message buffers/1 channel - Each message buffer can be set to be either a transmit message buffer or a receive message buffer.
Message reception	- Unique ID can be set to each message buffer. - Mask setting of four patterns is possible for each channel. - A receive completion interrupt is generated each time a message is received and stored in a message buffer. - Two or more receive message buffers can be used as a FIFO receive buffer (multi-buffer receive block function). - Receive history list function
Message transmission	- Unique ID can be set to each message buffer. - Transmit completion interrupt for each message buffer - Message buffer number 0 to 7 specified as the transmit message buffer can be used for automatic block transfer. Message transmission interval is programmable (automatic block transmission function (hereafter referred to as “ABT”)). - Transmission history list function
Remote frame processing	Remote frame processing by transmit message buffer
Time stamp function	- The time stamp function can be set for a message reception when a 16-bit timer is used in combination. Time stamp capture trigger can be selected (SOF or EOF in a CAN message frame can be detected.).
Diagnostic function	- Readable error counters - “Valid protocol operation flag” for verification of bus connections - Receive-only mode - Single-shot mode - CAN protocol error type decoding - Self-test mode
Forced release from bus-off state	- Forced release from bus-off (by ignoring timing constraint) possible by software. - No automatic release from bus-off (software must re-enable).
Power save mode	- CAN sleep mode (can be woken up by CAN bus) - CAN stop mode (cannot be woken up by CAN bus)

Caution To use the CAN controller, set P70 to 1.

15.1.3 Configuration

The CAN controller is composed of the following four blocks.

(1) NPB interface

This functional block provides an NPB (NEC peripheral I/O bus) interface and means of transmitting and receiving signals between the CAN module and the host CPU.

(2) MCM (Message Control Module)

This functional block controls access to the CAN protocol layer and to the CAN RAM within the CAN module.

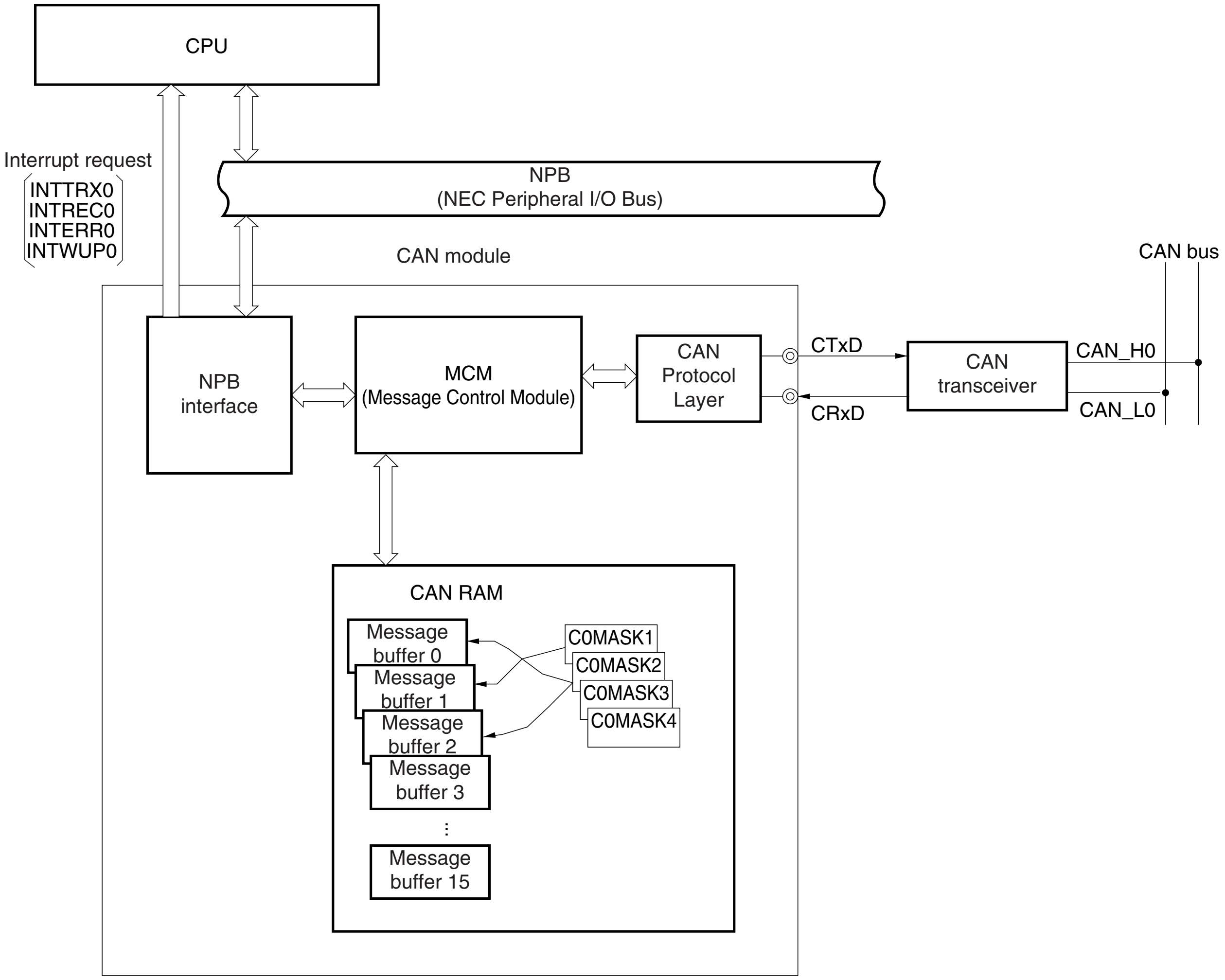
(3) CAN protocol layer

This functional block is involved in the operation of the CAN protocol and its related settings.

(4) CAN RAM

This is the CAN memory functional block, which is used to store message IDs, message data, etc.

Figure 15-1. Block Diagram of CAN Module



15.2 CAN Protocol

CAN (Controller Area Network) is a high-speed multiplex communication protocol for real-time communication in automotive applications (class C). CAN is prescribed by ISO 11898. For details, refer to the ISO 11898 specifications.

The CAN specification is generally divided into two layers: a physical layer and a data link layer. In turn, the data link layer includes logical link and medium access control. The composition of these layers is illustrated below.

Figure 15-2. Composition of Layers

Higher	Data link layer[Note]	• Logical link control (LLC)	• Acceptance filtering • Overload report • Recovery management
		• Medium access control (MAC)	• Data capsuled/not capsuled • Frame coding (stuffing/not stuffing) • Medium access management • Error detection • Error report • Acknowledgement • Seriated/not seriated
Lower	Physical layer		Prescription of signal level and bit description

Note CAN controller specification

15.2.1 Frame format

(1) Standard format frame

- The standard format frame uses 11-bit identifiers, which means that it can handle up to 2048 messages.

(2) Extended format frame

- The extended format frame uses 29-bit (11 bits + 18 bits) identifiers which increase the number of messages that can be handled to 2048 x 218 messages.
- Extended format frame is set when “recessive level” (CMOS level equals “1”) is set for both the SRR and IDE bits in the arbitration field.

15.2.2 Frame types

The following four types of frames are used in the CAN protocol.

Table 15-2. Frame Types

Frame Type	Description
Data frame	Frame used to transmit data
Remote frame	Frame used to request a data frame
Error frame	Frame used to report error detection
Overload frame	Frame used to delay the next data frame or remote frame

(1) Bus value

The bus values are divided into dominant and recessive.

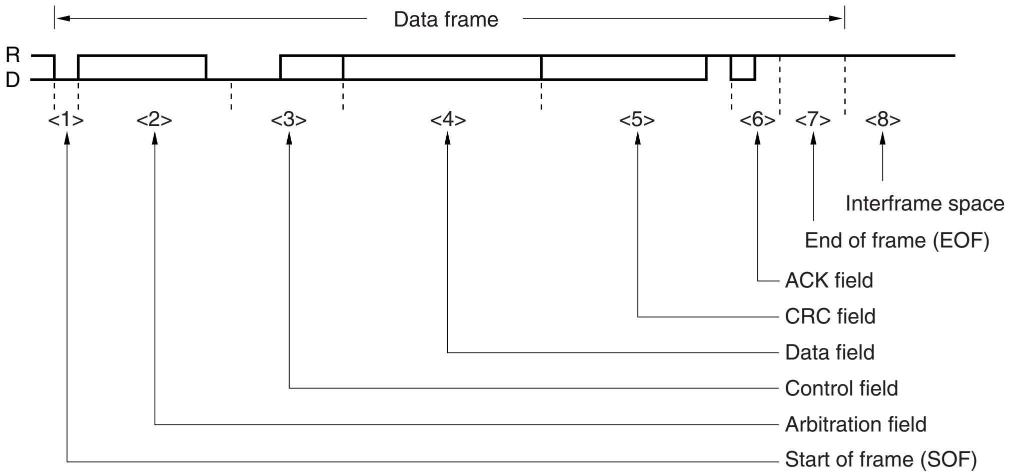
- Dominant level is indicated by logical 0.
- Recessive level is indicated by logical 1.
- When a dominant level and a recessive level are transmitted simultaneously, the bus value becomes dominant level.

15.2.3 Data frame and remote frame

(1) Data frame

A data frame is composed of seven fields.

Figure 15-3. Data Frame

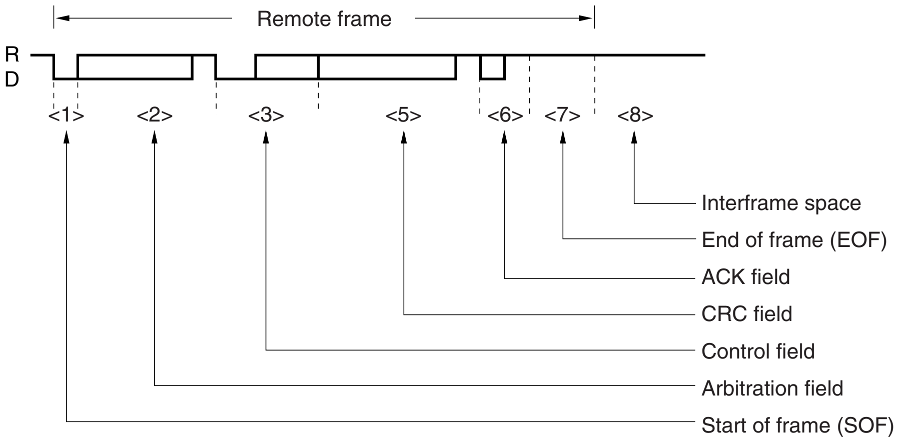


Remark D: Dominant = 0
R: Recessive = 1

(2) Remote frame

A remote frame is composed of six fields.

Figure 15-4. Remote Frame



Remarks 1. The data field is not transferred even if the control field's data length code is not "0000B".

2. D: Dominant = 0

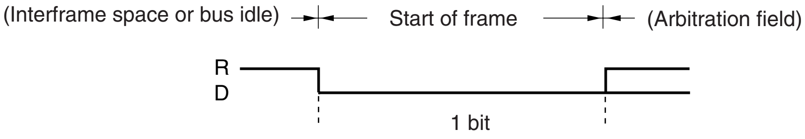
R: Recessive = 1

(3) Description of fields

<1> Start of frame (SOF)

The start of frame field is located at the start of a data frame or remote frame.

Figure 15-5. Start of Frame (SOF)



Remark D: Dominant = 0

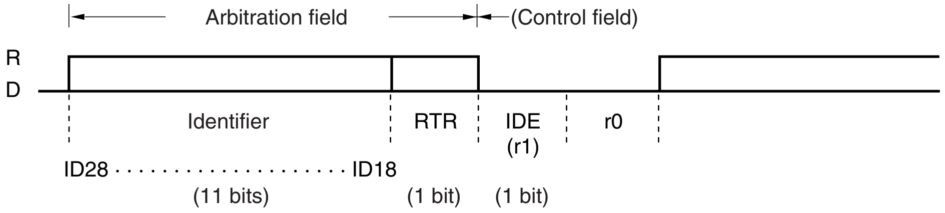
R: Recessive = 1

- If dominant level is detected in the bus idle state, a hard-synchronization is performed (the current TQ is assigned to be the SYNC segment).
- If dominant level is sampled at the sample point following such a hard-synchronization, the bit is assigned to be a SOF. If recessive level is detected, the protocol layer returns to the bus idle state and regards the preceding dominant pulse as a disturbance only. No error frame is generated in such case.

<2> Arbitration field

The arbitration field is used to set the priority, data frame/remote frame, and frame format.

Figure 15-6. Arbitration Field (in Standard Format Mode)



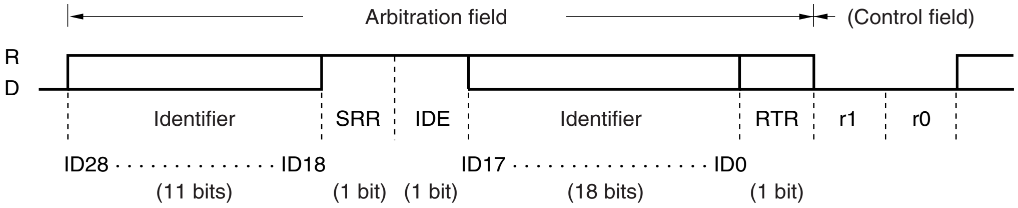
Cautions 1. ID28 to ID18 are identifiers.

2. An identifier is transmitted MSB first.

Remark D: Dominant = 0

R: Recessive = 1

Figure 15-7. Arbitration Field (in Extended Format Mode)



Cautions 1. ID28 to ID18 are identifiers.

2. An identifier is transmitted MSB first.

Remark D: Dominant = 0

R: Recessive = 1

Table 15-3. RTR Frame Settings

Frame Type	RTR Bit
Data frame	0 (D)
Remote frame	1 (R)

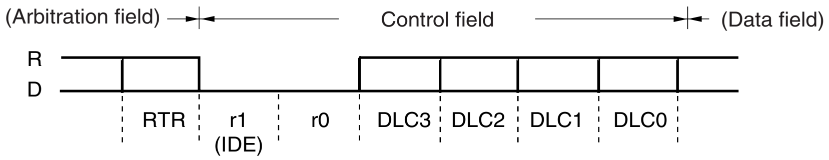
Table 15-4. Frame Format Setting (IDE Bit) and Number of Identifier (ID) Bits

Frame Format	SRR Bit	IDE Bit	Number. of Bits
Standard format mode	None	0 (D)	11 bits
Extended format mode	1 (R)	1 (R)	29 bits

<3> Control field

The control field sets “N” as the number of data bytes in the data field (N = 0 to 8).

Figure 15-8. Control Field



Remark D: Dominant = 0
R: Recessive = 1

In a standard format frame, the control field’s IDE bit is the same as the r1 bit.

Table 15-5. Data Length Setting

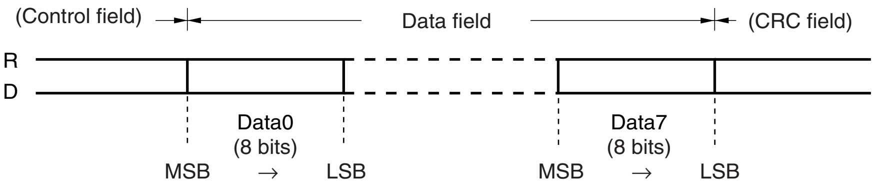
Data Length Code				Data Byte Count
DLC3	DLC2	DLC1	DLC0	
0	0	0	0	0 bytes
0	0	0	1	1 byte
0	0	1	0	2 bytes
0	0	1	1	3 bytes
0	1	0	0	4 bytes
0	1	0	1	5 bytes
0	1	1	0	6 bytes
0	1	1	1	7 bytes
1	0	0	0	8 bytes
Other than above				8 bytes regardless of the value of DLC3 to DLC0

Caution In the remote frame, there is no data field even if the data length code is not 0000B.

<4> Data field

The data field contains the amount of data (byte units) set by the control field. Up to 8 units of data can be set.

Figure 15-9. Data Field

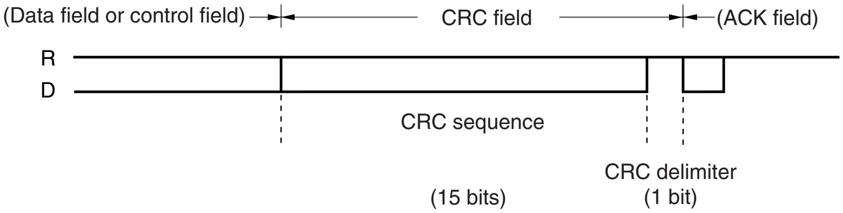


Remark D: Dominant = 0
R: Recessive = 1

<5> CRC field

The CRC field is a 16-bit field that is used to check for errors in transmit data.

Figure 15-10. CRC Field



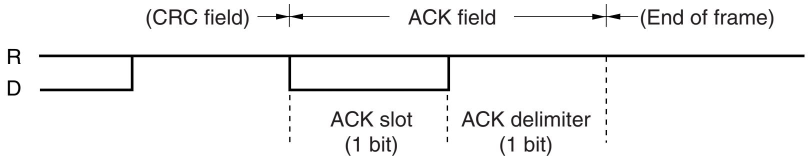
Remark D: Dominant = 0
R: Recessive = 1

- The polynomial P(X) used to generate the 15-bit CRC sequence is expressed as follows.
  $P(X) = X^{15} + X^{14} + X^{10} + X^{8} + X^{7} + X^{4} + X^{3} + 1$
- Transmitting node: Transmits the CRC sequence calculated from the data (before bit stuffing) in the start of frame, arbitration field, control field, and data field.
- Receiving node: Compares the CRC sequence calculated using data bits that exclude the stuffing bits in the receive data with the CRC sequence in the CRC field. If the two CRC sequences do not match, the node issues an error frame.

<6> ACK field

The ACK field is used to acknowledge normal reception.

Figure 15-11. ACK Field



Remark D: Dominant = 0

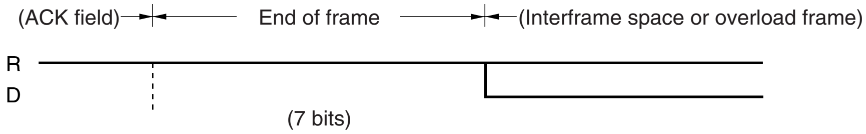
R: Recessive = 1

- If no CRC error is detected, the receiving node sets the ACK slot to the dominant level.
- The transmitting node outputs two recessive-level bits.

<7> End of frame (EOF)

The end of frame field indicates the end of data frame/remote frame.

Figure 15-12. End of Frame (EOF)



Remark D: Dominant = 0

R: Recessive = 1

<8> Interframe space

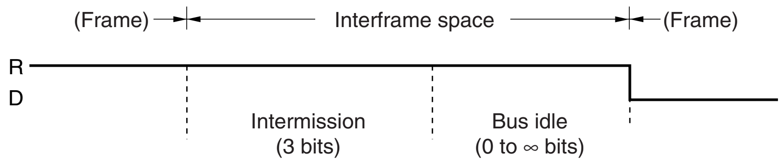
The interframe space is inserted after a data frame, remote frame, error frame, or overload frame to separate one frame from the next.

- The bus state differs depending on the error status.

(a) Error active node

The interframe space consists of a 3-bit intermission field and a bus idle field.

Figure 15-13. Interframe Space (Error Active Node)



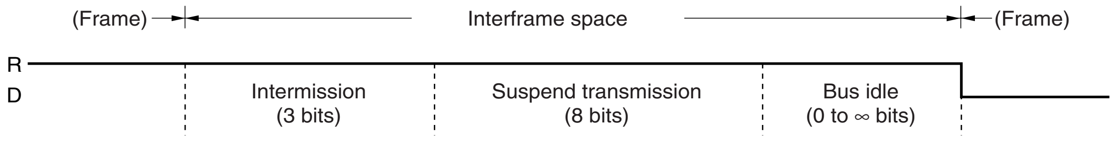
Remarks 1. Bus idle: State in which the bus is not used by any node.

2. D: Dominant = 0
R: Recessive = 1

(b) Error passive node

The interframe space consists of an intermission field, a suspend transmission field, and a bus idle field.

Figure 15-14. Interframe Space (Error Passive Node)



Remarks 1. Bus idle: State in which the bus is not used by any node.
Suspend transmission: Sequence of 8 recessive-level bits transmitted from the node in the error passive status.

2. D: Dominant = 0
R: Recessive = 1

Usually, the intermission field is 3 bits. If the transmitting node detects a dominant level at the third bit of the intermission field, however, it executes transmission.

- Operation in error status

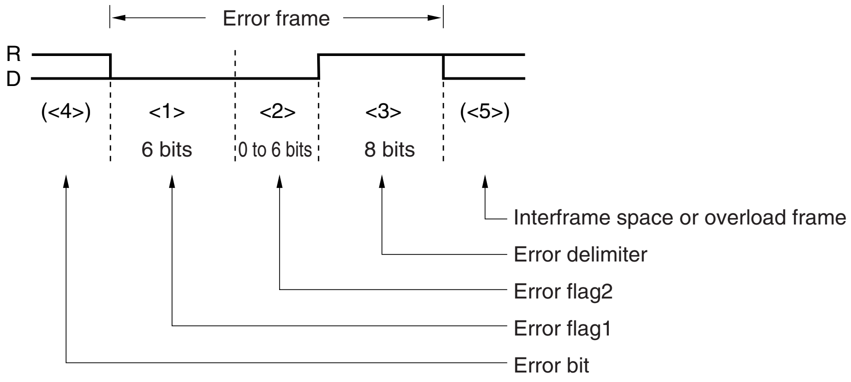
Table 15-6. Operation in Error Status

Error Status	Operation
Error active	A node in this status can transmit immediately after a 3-bit intermission.
Error passive	A node in this status can transmit 8 bits after the intermission.

15.2.4 Error frame

An error frame is output by a node that has detected an error.

Figure 15-15. Error Frame



Remark D: Dominant = 0
R: Recessive = 1

Table 15-7. Definition Error Frame Fields

No.	Name	Bit Count	Definition
<1>	Error flag1	6	Error active node: Outputs 6 dominant-level bits consecutively. Error passive node: Outputs 6 recessive-level bits consecutively. If another node outputs a dominant level while one node is outputting a passive error flag, the passive error flag is not cleared until the same level is detected 6 bits in a row.
<2>	Error flag2	0 to 6	Nodes receiving error flag 1 detect bit stuff errors and issues this error flag.
<3>	Error delimiter	8	Outputs 8 recessive-level bits consecutively. If a dominant level is detected at the 8th bit, an overload frame is transmitted from the next bit.
<4>	Error bit	–	The bit at which the error was detected. The error flag is output from the bit next to the error bit. In the case of a CRC error, this bit is output following the ACK delimiter.
<5>	Interframe space/overload frame	–	An interframe space or overload frame starts from here.5

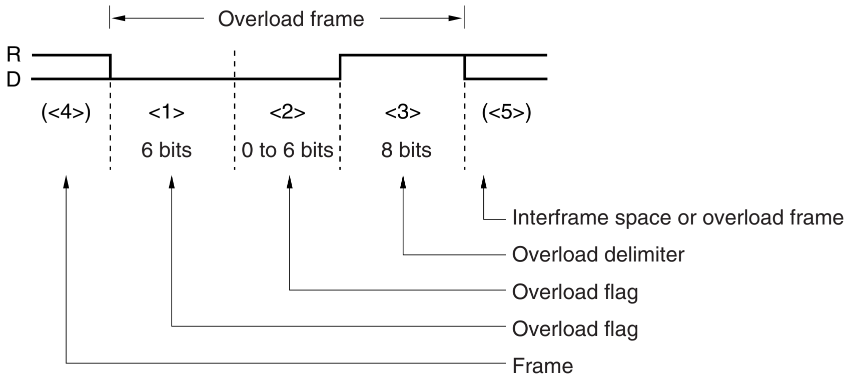
15.2.5 Overload frame

An overload frame is transmitted under the following conditions.

- When the receiving node has not completed the reception operation[Note]
- If a dominant level is detected at the first two bits during intermission
- If a dominant level is detected at the last bit (7th bit) of the end of frame or at the last bit (8th bit) of the error delimiter/overload delimiter

Note The CAN is internally fast enough to process all received frames not generating overload frames.

Figure 15-16. Overload Frame



Remark D: Dominant = 0
R: Recessive = 1

Table 15-8. Definition of Overload Frame Fields

No	Name	Bit Count	Definition
<1>	Overload flag	6	Outputs 6 dominant-level bits consecutively.
<2>	Overload flag from other node	0 to 6	The node that received an overload flag in the interframe space outputs an overload flag.
<3>	Overload delimiter	8	Outputs 8 recessive-level bits consecutively. If a dominant level is detected at the 8th bit, an overload frame is transmitted from the next bit.
<4>	Frame	–	Output following an end of frame, error delimiter, or overload delimiter.
<5>	Interframe space/overload frame	–	An interframe space or overload frame starts from here.

15.3 Functions

15.3.1 Determining bus priority

(1) When a node starts transmission:

- During bus idle, the node that output data first transmits the data.

(2) When more than one node starts transmission:

- The node that outputs the dominant level for the longest consecutively from the first bit of the arbitration field acquires the bus priority (if a dominant level and a recessive level are simultaneously transmitted, the dominant level is taken as the bus value).
- The transmitting node compares its output arbitration field and the data level on the bus.

Table 15-9. Determining Bus Priority

Level match	Continuous transmission
Level mismatch	Stops transmission at the bit where mismatch is detected and starts reception at the following bit

(3) Priority of data frame and remote frame

- When a data frame and a remote frame are on the bus, the data frame has priority because its RTR bit, the last bit in the arbitration field, carries a dominant level.

Caution If the extended-format data frame and the standard-format remote frame conflict on the bus (if ID28 to ID18 of both of them are the same), the standard-format remote frames takes priority.

15.3.2 Bit stuffing

Bit stuffing is used to establish synchronization by appending 1-bit inverted data if the same level continues for 5 bits, in order to prevent a burst error.

Table 15-10. Bit Stuffing

Transmission	During the transmission of a data frame or remote frame, when the same level continues for 5 bits in the data between the start of frame and the ACK field, 1 inverted-level bit of data is inserted before the following bit.
Reception	During the reception of a data frame or remote frame, when the same level continues for 5 bits in the data between the start of frame and the ACK field, reception is continued after deleting the next bit.

15.3.3 Multi masters

As the bus priority (a node acquiring transmit functions) is determined by the identifier, any node can be the bus master.

15.3.4 Multi cast

Although there is one transmitting node, two or more nodes can receive the same data at the same time because the same identifier can be set to two or more nodes.

15.3.5 CAN sleep mode/CAN stop mode function

The CAN sleep mode/CAN stop mode function puts the CAN controller in waiting mode to achieve low power consumption.

The controller is woken up from the CAN sleep mode by bus operation but it is not woken up from the CAN stop mode by bus operation (the CAN stop mode is controlled by CPU access).

15.3.6 Error control function

(1) Error types

Table 15-11. Error Types

Type	Description of Error		Detection State	
	Detection Method	Detection Condition	Transmission/ Reception	Field/Frame
Bit error	Comparison of output level and level on the bus	Mismatch of levels	Transmitting/ receiving node	Bit that outputting data on the bus at the start of frame to end of frame, error frame and overload frame.
Stuff error	Check the receive data at the stuff bit	6 consecutive bits of the same output level	Receiving node	Start of frame to CRC sequence
CRC error	Comparison of the CRC sequence generated from the receive data and the received CRC sequence	Mismatch of CRC	Receiving node	CRC field
Form error	Field/frame check of the fixed format	Detection of fixed format violation	Receiving node	CRC delimiter ACK field End of frame Error frame Overload frame
ACK error	Check of the ACK slot by the transmitting node	Detection of recessive level in ACK slot	Transmitting node	ACK slot

(2) Output timing of error frame

Table 15-12. Output Timing of Error Frame

Type	Output Timing
Bit error, stuff error, form error, ACK error	Error frame output is started at the timing of the bit following the detected error.
CRC error	Error frame output is started at the timing of the bit following the ACK delimiter.

(3) Processing in case of error

The transmission node re-transmits the data frame or remote frame after the error frame (However, it does not re-transmit the frame in the single-shot mode).

(4) Error state

(a) Types of error states

The following three types of error states are defined by the CAN specification.

- Error active
- Error passive
- Bus-off

These types of error states are classified by the values of the TEC7 to TEC0 bits (transmission error counter bits) and the REC6 to REC0 bits (reception error counter bits) of the CAN error counter register (C0ERC) as shown in **Table 15-13**.

The present error state is indicated by the CAN module information register (C0INFO).

When each error counter value becomes equal to or greater than the error warning level (96), the TECS0 or RECS0 bit of the C0INFO register is set to 1. In this case, the bus state must be tested because it is considered that the bus has a serious fault. An error counter value of 128 or more indicates an error passive state and the TECS1 or RECS1 bit of the C0INFO register is set to 1.

- If the value of the transmission error counter is greater than or equal to 256 (actually, the transmission error counter does not indicate a value greater than or equal to 256), the bus-off state is reached and the BOFF bit of the C0INFO register is set to 1.
- If only one node is active on the bus at startup (i.e., a particular case such as when the bus is connected only to the local station), ACK is not returned even if data is transmitted. Consequently, re-transmission of the error frame and data is repeated. In the error passive state, however, the transmission error counter is not incremented and the bus-off state is not reached.

Table 15-13. Types of Error States

Type	Operation	Value of Error Counter	Indication of C0INFO Register	Operation specific to Given Error State
Error active	Transmission	0-95	TECS1, TECS0 = 00	- Outputs an active error flag (6 consecutive dominant-level bits) on detection of the error.
	Reception	0-95	RECS1, RECS0 = 00	
	Transmission	96-127	TECS1, TECS0 = 01	
	Reception	96-127	RECS1, RECS0 = 01	
Error passive	Transmission	128-255	TECS1, TECS0 = 11	- Outputs a passive error flag (6 consecutive recessive-level bits) on detection of the error. - Transmits 8 recessive-level bits, in between transmissions, following an intermission (suspend transmission).
	Reception	128 or more	RECS1, RECS0 = 11	
Bus-off	Transmission	256 or more (not indicated)[Note]	BOFF = 1, TECS1, TECS0 = 11	- Communication is not possible. Messages are not stored when receiving frames, however, the following operations of <1>, <2>, and <3> are done. <1> TSOUT toggles. <2> REC is incremented/decremented. <3> VALID bit is set. - If the CAN module is entered to the initialization mode and then transition request to any operation mode is made, and when 11 consecutive recessive-level bits are detected 128 times, the error counter is reset to 0 and the error active state can be restored.

Note The value of the transmission error counter (TEC) is invalid when the BOFF bit is set to 1. If an error that increments the value of the transmission error counter by +8 while the counter value is in a range of 248 to 255, the counter is not incremented and the bus-off state is assumed.

(b) Error counter

The error counter counts up when an error has occurred, and counts down upon successful transmission and reception. The error counter is updated immediately after error detection.

Table 15-14. Error Counter

State	Transmission Error Counter (TEC7 to TEC0)	Reception Error Counter (REC6 to REC0)
Receiving node detects an error (except bit error in the active error flag or overload flag).	No change	+1 (when REPS bit = 0)
Receiving node detects dominant level following error flag of error frame.	No change	+8 (when REPS bit = 0)
Transmitting node transmits an error flag. [As exceptions, the error counter does not change in the following cases.] <1> ACK error is detected in error passive state and dominant level is not detected while the passive error flag is being output. <2> A stuff error is detected in an arbitration field that transmitted a recessive level as a stuff bit, but a dominant level is detected.	+8	No change
Bit error detection while active error flag or overload flag is being output (error-active transmitting node)	+8	No change
Bit error detection while active error flag or overload flag is being output (error-active receiving node)	No change	+8 (when REPS bit = 0)
When the node detects 14 consecutive dominant-level bits from the beginning of the active error flag or overload flag, and then subsequently detects 8 consecutive dominant-level bits. When the node detects 8 consecutive dominant levels after a passive error flag	+8 (during transmission)	+8 (during reception, when REPS bit = 0)
When the transmitting node has completed transmission without error (±0 if error counter = 0)	–1	No change
When the receiving node has completed reception without error	No change	- –1 (1 ≤ REC6 to REC0 ≤ 127, when REPS bit = 0) - ±0 (REC6 to REC0 = 0, when REPS bit = 0) - Value of 119 to 255 is set (when REPS bit = 1)

(c) Occurrence of bit error in intermission

An overload frame is generated.

Caution If an error occurs, the error flag output (active or passive) is controlled according to the contents of the transmission error counter and reception error counter before the error occurred. The value of the error counter is incremented after the error flag has been output.

(5) Recovery from bus-off state

When the CAN module is in the bus-off state, the CAN module permanently sets its output signals (CTxD) to recessive level.

The CAN module recovers from the bus-off state in the following bus-off recovery sequence.

<1> A request to enter the CAN initialization mode

<2> A request to enter a CAN operation mode

(a) Recovery operation through normal recovery sequence

(b) Forced recovery operation that skips recovery sequence

(a) Recovery operation from bus-off state through normal recovery sequence

The CAN module first issues a request to enter the initialization mode (refer to timing <1> in **Figure 15-17**). This request will be immediately acknowledged, and the OPMODE bits of the C0CTRL register are cleared to 000B. Processing such as analyzing the fault that has caused the bus-off state, re-defining the CAN module and message buffer using application software, or stopping the operation of the CAN module can be performed by clearing the GOM bit to 0.

Next, the user requests to change the mode from the initialization mode to an operation mode (refer to timing <2> in **Figure 15-17**). This starts an operation to recover the CAN module from the bus-off state. The conditions under which the module can recover from the bus-off state are defined by the CAN protocol ISO 11898, and it is necessary to detect 11 consecutive recessive-level bits 128 times. At this time, the request to change the mode to an operation mode is held pending until the recovery conditions are satisfied. When the recovery conditions are satisfied (refer to timing <3> in **Figure 15-17**), the CAN module can enter the operation mode it has requested. Until the CAN module enters this operation mode, it stays in the initialization mode. Completion to be requested operation mode can be confirmed by reading the OPMODE bits of the C0CTRL register.

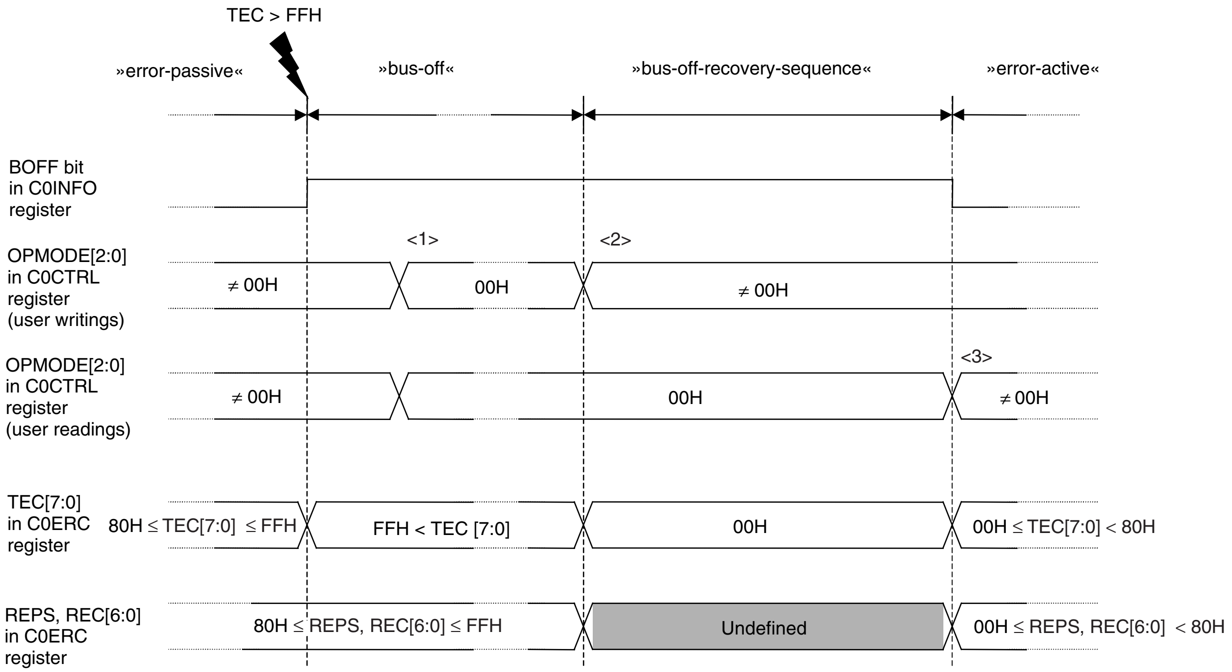
During the bus-off period and bus-off recovery sequence, the BOFF bit of the C0INFO register stays set (to 1). In the bus-off recovery sequence, the reception error counter (REC[6:0]) counts the number of times 11 consecutive recessive-level bits have been detected on the bus. Therefore, the recovery state can be checked by reading REC[6:0].

Cautions 1. If the Bus-off Recovery Sequence is interrupted by entering Initialization Mode and re-entering any Operation Mode, the Bus-off Recovery Sequence will restart from the beginning, and the waiting phase will be again 128 times 11 recessive-level bits, counted from this point.

2. In the bus-off recovery sequence, REC [6:0] counts up (+1) each time 11 consecutive recessive-level bits have been detected. Even during the bus-off period, the CAN module can enter the CAN sleep mode or CAN stop mode. To start the bus-off recovery sequence, it is necessary to transit to the initialization mode once.

However, when the CAN module is in either CAN sleep mode or CAN stop mode, transition request to the initialization mode is not accepted, thus you have to release the CAN sleep mode first. In this case, as soon as the CAN sleep mode is released, the bus-off recovery sequence starts and no transition to initialization mode is necessary. If the CAN module detects a dominant edge on the CAN bus while in sleep mode even during bus-off, the sleep mode will be left and the bus-off recovery sequence will start (in the state that the CAN clock is supplied, it is necessary to clear the PSMODE by software after dominant edge detection).

Figure 15-17. Recovery Operation from Bus-off State through Normal Recovery Sequence



(b) Forced recovery operation that skips bus-off recovery sequence

The CAN module can be forcibly released from the bus-off state, regardless of the bus state, by skipping the bus-off recovery sequence. Here is the procedure.

First, the CAN module requests to enter the initialization mode. For the operation and points to be noted at this time, refer to (a) Recovery operation from bus-off state through normal recovery sequence.

Next, the module requests to enter an operation mode. At the same time, the CCERC bit of the C0CTRL register must be set to 1.

As a result, the bus-off recovery sequence defined by the CAN protocol ISO 11898 is skipped, and the module immediately enters the operation mode. In this case, the module is connected to the CAN bus after it has monitored 11 consecutive recessive-level bits. For details, refer to the processing in **Figure 15-56. Bus-Off Recovery (Expect Normal Operation Mode with ABT)**.

Caution This function is not defined by the CAN protocol ISO 11898. When using this function, thoroughly evaluate its effect on the network system.

(6) Initializing CAN module error counter register (C0ERC) in initialization mode

If it is necessary to initialize the CAN module error counter register (C0ERC) and CAN module information register (C0INFO) for debugging or evaluating a program, they can be initialized to the default value by setting the CCERC bit of the C0CTRL register in the initialization mode. When initialization has been completed, the CCERC bit is automatically cleared to 0.

Cautions 1. This function is enabled only in the initialization mode. Even if the CCERC bit is set to 1 in a CAN operation mode, the C0ERC and C0INFO registers are not initialized.

2. The CCERC bit can be set at the same time as the request to enter a CAN operation mode.

15.3.7 Baud rate control function

(1) Prescaler

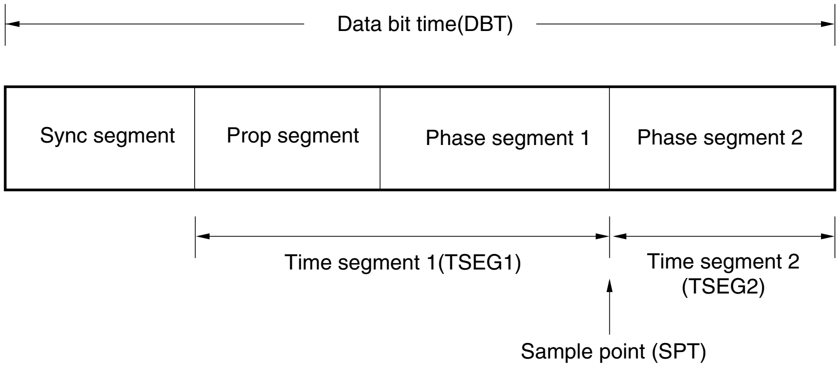
The CAN controller has a prescaler that divides the clock (f_{CAN}) supplied to CAN. This prescaler generates a CAN protocol layer basic clock (f_{TQ}) derived from the CAN module system clock (f_{CANMOD}), and divided by 1 to 256 (refer to **15.7 (12) CAN Bit Rate Prescaler Register (C0BRP)**).

(2) Data bit time (8-25 time quanta)

One data bit time is defined as shown in **Figure 15-18**.

The CAN controller sets time segment 1, time segment 2, and reSynchronization Jump Width (SJW) as the parameter of data bit time, as shown in **Figure 15-18**. Time segment 1 is equivalent to the total of the propagation (prop) segment and phase segment 1 that are defined by the CAN protocol specification. Time segment 2 is equivalent to phase segment 2.

Figure 15-18. Segment Setting

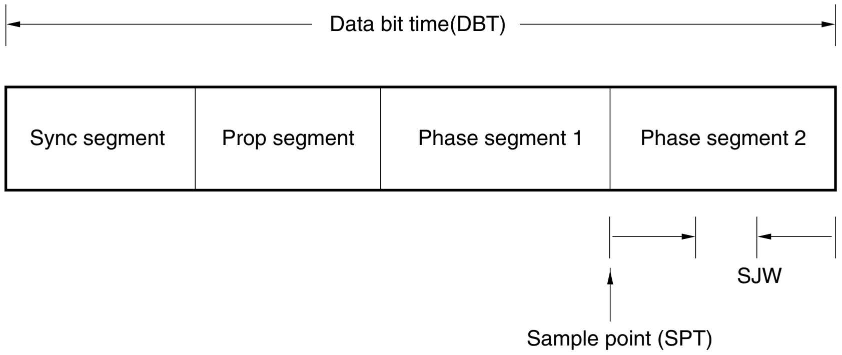


Segment Name	Settable Range	Notes on Setting to Confirm to CAN Specification
Time Segment 1 (TSEG1)	2TQ-16TQ	—
Time Segment 2 (TSEG2)	1TQ-8TQ	IPT of the CAN controller is 0TQ. To conform to the CAN protocol specification, therefore, a length equal to phase segment 1 must be set here. This means that the length of time segment 1 minus 1TQ is the settable upper limit of time segment 2.
Resynchronization jump width(SJW)	1TQ-4TQ	The length of time segment 1 minus 1TQ or 4 TQ, whichever is smaller.

Remark IPT : Information Processing Time
TQ : Time Quanta

Reference: The CAN standard ISO 11898 specification defines the segments constituting the data bit time as shown in **Figure 15-19**.

Figure 15-19. Reference: Configuration of Data Bit Time Defined by CAN Specification



Segment Name	Segment Length	Description
Sync Segment (Synchronization Segment)	1	This segment starts at the edge where the level changes from recessive to dominant when hard-synchronization is established.
Prop Segment	Programmable to 1 to 8 or more	This segment absorbs the delay of the output buffer, CAN bus, and input buffer. The length of this segment is set so that ACK is returned before the start of phase segment 1. Time of prop segment ≥ (Delay of output buffer) + 2 x (Delay of CAN bus) + (Delay of input buffer)
Phase Segment 1	Programmable to 1 to 8	This segment compensates for an error of data bit time. The longer this segment, the wider the permissible range but the slower the communication speed.
Phase Segment 2	Phase Segment 1 or IPT, whichever greater	
SJW	Programmable from 1TQ to length of segment 1 or 4TQ, whichever is smaller	This width sets the upper limit of expansion or contraction of the phase segment during resynchronization.

Remark IPT : Information Processing Time
TQ : Time Quanta

(3) Synchronizing data bit

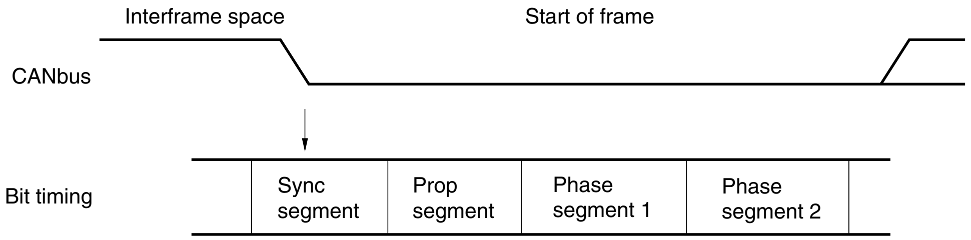
- The receiving node establishes synchronization by a level change on the bus because it does not have a sync signal.
- The transmitting node transmits data in synchronization with the bit timing of the transmitting node.

(a) Hard-synchronization

This synchronization is established when the receiving node detects the start of frame in the interframe space.

- When a falling edge is detected on the bus, that TQ means the sync segment and the next segment is the prop segment. In this case, synchronization is established regardless of SJW.

Figure 15-20. Hard-synchronization at Recognition of Dominant Level during Bus Idle

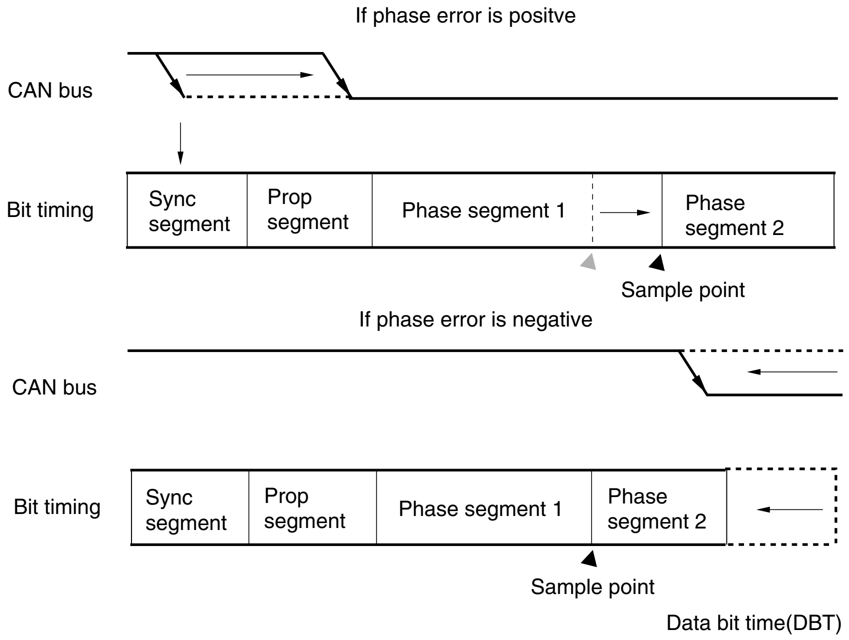


(b) Resynchronization

Synchronization is established again if a level change is detected on the bus during reception (only if a recessive level was sampled previously).

- The phase error of the edge is given by the relative position of the detected edge and sync segment.
  <Sign of phase error>
    0: If the edge is within the sync segment
    Positive: If the edge is before the sample point (phase error)
    Negative: If the edge is after the sample point (phase error)
    If phase error is positive: Phase segment 1 is longer by specified SJW.
    If phase error is negative: Phase segment 2 is shorter by specified SJW.
- The sample point of the data of the receiving node moves relatively due to the “discrepancy” in baud rate between the transmitting node and receiving node.

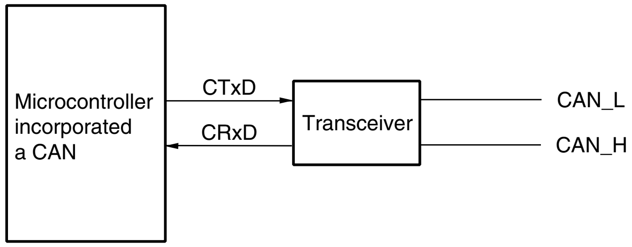
Figure 15-21. Resynchronization



15.4 Connection with Target System

The microcontroller incorporated a CAN has to be connected to the CAN bus using an external transceiver.

Figure 15-22. Connection to CAN Bus



15.5 Internal Registers of CAN Controller

15.5.1 CAN controller configuration

Table 15-15. List of CAN Controller Registers

Item	Register Name
CAN global registers	CAN global control register (C0GMCTRL)
	CAN global clock selection register (C0GMCS)
	CAN global automatic block transmission control register (C0GMABT)
	CAN global automatic block transmission delay setting register (C0GMABTD)
CAN module registers	CAN module mask 1 register (C0MASK1L, C0MASK1H)
	CAN module mask 2 register (C0MASK2L, C0MASK2H)
	CAN module mask3 register (C0MASK3L, C0MASK3H)
	CAN module mask 4 registers (C0MASK4L, C0MASK4H)
	CAN module control register (C0CTRL)
	CAN module last error code register (C0LEC)
	CAN module information register (C0INFO)
	CAN module error counter register (C0ERC)
	CAN module interrupt enable register (C0IE)
	CAN module interrupt status register (C0INTS)
	CAN module bit rate prescaler register (C0BRP)
	CAN module bit rate register (C0BTR)
	CAN module last in-pointer register (C0LIPT)
	CAN module receive history list register (C0RGPT)
	CAN module last out-pointer register (C0LOPT)
	CAN module transmit history list register (C0TGPT)
	CAN module time stamp register (C0TS)
Message buffer registers	CAN message data byte 01 register m (C0MDATA01m)
	CAN message data byte 0 register m (C0MDATA0m)
	CAN message data byte 1 register m (C0MDATA1m)
	CAN message data byte 23 register m (C0MDATA23m)
	CAN message data byte 2 register m (C0MDATA2m)
	CAN message data byte 3 Register m (C0MDATA3m)
	CAN message data byte 45 Register m (C0MDATA45m)
	CAN message data byte 4 Register m (C0MDATA4m)
	CAN message data byte 5 Register m (C0MDATA5m)
	CAN message data byte 67 Register m (C0MDATA67m)
	CAN message data byte 6 register m (C0MDATA6m)
	CAN message data byte 7 register m (C0MDATA7m)
	CAN message data length register m (C0MDLCm)
	CAN message configuration register m (C0MCONFm)
	CAN message ID register m (C0MIDLm, C0MIDHm)
	CAN message control register m (C0MCTRLm)

Remarks 1. CAN global registers are identified by C0GM<register function>.
CAN module registers are identified by C0<register function>.
Message buffer registers are identified by C0M<register function>.

2. m = 0 to 15

15.5.2 Register access type

Table 15-16. Register Access Types (1/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
FA00H	CAN0 message data byte 01 register 00	C0MDATA0100	R/W			√	Undefined
FA00H	CAN0 message data byte 0 register 00	C0MDATA000			√		Undefined
FA01H	CAN0 message data byte 1 register 00	C0MDATA100			√		Undefined
FA02H	CAN0 message data byte 23 register 00	C0MDATA2300				√	Undefined
FA02H	CAN0 message data byte 2 register 00	C0MDATA200			√		Undefined
FA03H	CAN0 message data byte 3 register 00	C0MDATA300			√		Undefined
FA04H	CAN0 message data byte 45 register 00	C0MDATA4500				√	Undefined
FA04H	CAN0 message data byte 4 register 00	C0MDATA400			√		Undefined
FA05H	CAN0 message data byte 5 register 00	C0MDATA500			√		Undefined
FA06H	CAN0 message data byte 67 register 00	C0MDATA6700				√	Undefined
FA06H	CAN0 message data byte 6 register 00	C0MDATA600			√		Undefined
FA07H	CAN0 message data byte 7 register 00	C0MDATA700			√		Undefined
FA08H	CAN0 message data length code register 00	C0MDLC00			√		0000xxxxB
FA09H	CAN0 message configuration register 00	C0MCONF00			√		Undefined
FA0AH	CAN0 message ID register 00	C0MIDL00				√	Undefined
FA0CH		C0MIDH00				√	Undefined
FA0EH	CAN0 message control register 00	C0MCTRL00				√	00x00000 000xx000B
FA10H	CAN0 message data byte 01 register 01	C0MDATA0101				√	Undefined
FA10H	CAN0 message data byte 0 register 01	C0MDATA001			√		Undefined
FA11H	CAN0 message data byte 1 register 01	C0MDATA101			√		Undefined
FA12H	CAN0 message data byte 23 register 01	C0MDATA2301				√	Undefined
FA12H	CAN0 message data byte 2 register 01	C0MDATA201			√		Undefined
FA13H	CAN0 message data byte 3 register 01	C0MDATA301			√		Undefined
FA14H	CAN0 message data byte 45 register 01	C0MDATA4501				√	Undefined
FA14H	CAN0 message data byte 4 register 01	C0MDATA401			√		Undefined
FA15H	CAN0 message data byte 5 register 01	C0MDATA501			√		Undefined
FA16H	CAN0 message data byte 67 register 01	C0MDATA6701				√	Undefined
FA16H	CAN0 message data byte 6 register 01	C0MDATA601			√		Undefined
FA17H	CAN0 message data byte 7 register 01	C0MDATA701			√		Undefined
FA18H	CAN0 message data length code register 01	C0MDLC01			√		0000xxxxB
FA19H	CAN0 message configuration register 01	C0MCONF01			√		Undefined
FA1AH	CAN0 message ID register 01	C0MIDL01				√	Undefined
FA1CH		C0MIDH01				√	Undefined
FA1EH	CAN0 message control register 01	C0MCTRL01				√	00x00000 000xx000B

Table 15-16. Register Access Types (2/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
FA20H	CAN0 message data byte 01 register 02	C0MDATA0102	R/W			√	Undefined
FA20H	CAN0 message data byte 0 register 02	C0MDATA002			√		Undefined
FA21H	CAN0 message data byte 1 register 02	C0MDATA102			√		Undefined
FA22H	CAN0 message data byte 23 register 02	C0MDATA2302				√	Undefined
FA22H	CAN0 message data byte 2 register 02	C0MDATA202			√		Undefined
FA23H	CAN0 message data byte 3 register 02	C0MDATA302			√		Undefined
FA24H	CAN0 message data byte 45 register 02	C0MDATA4502				√	Undefined
FA24H	CAN0 message data byte 4 register 02	C0MDATA402			√		Undefined
FA25H	CAN0 message data byte 5 register 02	C0MDATA502			√		Undefined
FA26H	CAN0 message data byte 67 register 02	C0MDATA6702				√	Undefined
FA26H	CAN0 message data byte 6 register 02	C0MDATA602			√		Undefined
FA27H	CAN0 message data byte 7 register 02	C0MDATA702			√		Undefined
FA28H	CAN0 message data length code register 02	C0MDLC02			√		0000xxxxB
FA29H	CAN0 message configuration register 02	C0MCONF02			√		Undefined
FA2AH	CAN0 message ID register 02	C0MIDL02				√	Undefined
FA2CH		C0MIDH02				√	Undefined
FA2EH	CAN0 message control register 02	C0MCTRL02				√	00x00000000xx000B
FA30H	CAN0 message data byte 01 register 03	C0MDATA0103				√	Undefined
FA30H	CAN0 message data byte 0 register 03	C0MDATA003			√		Undefined
FA31H	CAN0 message data byte 1 register 03	C0MDATA103			√		Undefined
FA32H	CAN0 message data byte 23 register 03	C0MDATA2303				√	Undefined
FA32H	CAN0 message data byte 2 register 03	C0MDATA203			√		Undefined
FA33H	CAN0 message data byte 3 register 03	C0MDATA303			√		Undefined
FA34H	CAN0 message data byte 45 register 03	C0MDATA4503				√	Undefined
FA34H	CAN0 message data byte 4 register 03	C0MDATA403			√		Undefined
FA35H	CAN0 message data byte 5 register 03	C0MDATA503			√		Undefined
FA36H	CAN0 message data byte 67 register 03	C0MDATA6703				√	Undefined
FA36H	CAN0 message data byte 6 register 03	C0MDATA603			√		Undefined
FA37H	CAN0 message data byte 7 register 03	C0MDATA703			√		Undefined
FA38H	CAN0 message data length code register 03	C0MDLC03			√		0000xxxxB
FA39H	CAN0 message configuration register 03	C0MCONF03			√		Undefined
FA3AH	CAN0 message ID register 03	C0MIDL03				√	Undefined
FA3CH		C0MIDH03				√	Undefined
FA3EH	CAN0 message control register 03	C0MCTRL03				√	00x00000000xx000B

Table 15-16. Register Access Types (3/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
FA40H	CAN0 message data byte 01 register 04	C0MDATA0104	R/W			√	Undefined
FA40H	CAN0 message data byte 0 register 04	C0MDATA004			√		Undefined
FA41H	CAN0 message data byte 1 register 04	C0MDATA104			√		Undefined
FA42H	CAN0 message data byte 23 register 04	C0MDATA2304				√	Undefined
FA42H	CAN0 message data byte 2 register 04	C0MDATA204			√		Undefined
FA43H	CAN0 message data byte 3 register 04	C0MDATA304			√		Undefined
FA44H	CAN0 message data byte 45 register 04	C0MDATA4504				√	Undefined
FA44H	CAN0 message data byte 4 register 04	C0MDATA404			√		Undefined
FA45H	CAN0 message data byte 5 register 04	C0MDATA504			√		Undefined
FA46H	CAN0 message data byte 67 register 04	C0MDATA6704				√	Undefined
FA46H	CAN0 message data byte 6 register 04	C0MDATA604			√		Undefined
FA47H	CAN0 message data byte 7 register 04	C0MDATA704			√		Undefined
FA48H	CAN0 message data length code register 04	C0MDLC04			√		0000xxxxB
FA49H	CAN0 message configuration register 04	C0MCONF04			√		Undefined
FA4AH	CAN0 message ID register 04	C0MIDL04				√	Undefined
FA4CH		C0MIDH04				√	Undefined
FA4EH	CAN0 message control register 04	C0MCTRL04				√	00x00000 000xx000B
FA50H	CAN0 message data byte 01 register 05	C0MDATA0105				√	Undefined
FA50H	CAN0 message data byte 0 register 05	C0MDATA005			√		Undefined
FA51H	CAN0 message data byte 1 register 05	C0MDATA105			√		Undefined
FA52H	CAN0 message data byte 23 register 05	C0MDATA2305				√	Undefined
FA52H	CAN0 message data byte 2 register 05	C0MDATA205			√		Undefined
FA53H	CAN0 message data byte 3 register 05	C0MDATA305			√		Undefined
FA54H	CAN0 message data byte 45 register 05	C0MDATA4505				√	Undefined
FA54H	CAN0 message data byte 4 register 05	C0MDATA405			√		Undefined
FA55H	CAN0 message data byte 5 register 05	C0MDATA505			√		Undefined
FA56H	CAN0 message data byte 67 register 05	C0MDATA6705				√	Undefined
FA56H	CAN0 message data byte 6 register 05	C0MDATA605			√		Undefined
FA57H	CAN0 message data byte 7 register 05	C0MDATA705			√		Undefined
FA58H	CAN0 message data length code register 05	C0MDLC05			√		0000xxxxB
FA59H	CAN0 message configuration register 05	C0MCONF05			√		Undefined
FA5AH	CAN0 message ID register 05	C0MIDL05				√	Undefined
FA5CH		C0MIDH05				√	Undefined
FA5EH	CAN0 message configuration register 05	C0MCTRL05				√	00x00000 000xx000B

Table 15-16. Register Access Types (4/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
FA60H	CAN0 message data byte 01 register 06	C0MDATA0106	R/W			√	Undefined
FA60H	CAN0 message data byte 0 register 06	C0MDATA006			√		Undefined
FA61H	CAN0 message data byte 1 register 06	C0MDATA106			√		Undefined
FA62H	CAN0 message data byte 23 register 06	C0MDATA2306				√	Undefined
FA62H	CAN0 message data byte 2 register 06	C0MDATA206			√		Undefined
FA63H	CAN0 message data byte 3 register 06	C0MDATA306			√		Undefined
FA64H	CAN0 message data byte 45 register 06	C0MDATA4506				√	Undefined
FA64H	CAN0 message data byte 4 register 06	C0MDATA406			√		Undefined
FA65H	CAN0 message data byte 5 register 06	C0MDATA506			√		Undefined
FA66H	CAN0 message data byte 67 register 06	C0MDATA6706				√	Undefined
FA66H	CAN0 message data byte 6 register 06	C0MDATA606			√		Undefined
FA67H	CAN0 message data byte 7 register 06	C0MDATA706			√		Undefined
FA68H	CAN0 message data length code register 06	C0MDLC06			√		0000xxxxB
FA69H	CAN0 message configuration register 06	C0MCONF06			√		Undefined
FA6AH	CAN0 message ID register 06	C0MIDL06				√	Undefined
FA6CH		C0MIDH06				√	Undefined
FA6EH	CAN0 message control register 06	C0MCTRL06				√	00x00000 000xx000B
FA70H	CAN0 message data byte 01 register 07	C0MDATA0107				√	Undefined
FA70H	CAN0 message data byte 0 register 07	C0MDATA007			√		Undefined
FA71H	CAN0 message data byte 1 register 07	C0MDATA107			√		Undefined
FA72H	CAN0 message data byte 23 register 07	C0MDATA2307				√	Undefined
FA72H	CAN0 message data byte 2 register 07	C0MDATA207			√		Undefined
FA73H	CAN0 message data byte 3 register 07	C0MDATA307			√		Undefined
FA74H	CAN0 message data byte 45 register 07	C0MDATA4507				√	Undefined
FA74H	CAN0 message data byte 4 register 07	C0MDATA407			√		Undefined
FA75H	CAN0 message data byte 5 register 07	C0MDATA507			√		Undefined
FA76H	CAN0 message data byte 67 register 07	C0MDATA6707				√	Undefined
FA76H	CAN0 message data byte 6 register 07	C0MDATA607			√		Undefined
FA77H	CAN0 message data byte 7 register 07	C0MDATA707			√		Undefined
FA78H	CAN0 message data length code register 07	C0MDLC07			√		0000xxxxB
FA79H	CAN0 message configuration register 07	C0MCONF07			√		Undefined
FA7AH	CAN0 message ID register 07	C0MIDL07				√	Undefined
FA7CH		C0MIDH07				√	Undefined
FA7EH	CAN0 message control register 07	C0MCTRL07				√	00x00000 000xx000B

Table 15-16. Register Access Types (5/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
FA80H	CAN0 message data byte 01 register 08	C0MDATA0108	R/W			√	Undefined
FA80H	CAN0 message data byte 0 register 08	C0MDATA008			√		Undefined
FA81H	CAN0 message data byte 1 register 08	C0MDATA108			√		Undefined
FA82H	CAN0 message data byte 23 register 08	C0MDATA2308				√	Undefined
FA82H	CAN0 message data byte 2 register 08	C0MDATA208			√		Undefined
FA83H	CAN0 message data byte 3 register 08	C0MDATA308			√		Undefined
FA84H	CAN0 message data byte 45 register 08	C0MDATA4508				√	Undefined
FA84H	CAN0 message data byte 4 register 08	C0MDATA408			√		Undefined
FA85H	CAN0 message data byte 5 register 08	C0MDATA508			√		Undefined
FA86H	CAN0 message data byte 67 register 08	C0MDATA6708				√	Undefined
FA86H	CAN0 message data byte 6 register 08	C0MDATA608			√		Undefined
FA87H	CAN0 message data byte 7 register 08	C0MDATA708			√		Undefined
FA88H	CAN0 message data length code register 08	C0MDLC08			√		0000xxxxB
FA89H	CAN0 message configuration register 08	C0MCONF08			√		Undefined
FA8AH	CAN0 message ID register 08	C0MIDL08				√	Undefined
FA8CH		C0MIDH08				√	Undefined
FA8EH	CAN0 message control register 08	C0MCTRL08				√	00x00000 000xx000B
FA90H	CAN0 message data byte 01 register 09	C0MDATA0109				√	Undefined
FA90H	CAN0 message data byte 0 register 09	C0MDATA009			√		Undefined
FA91H	CAN0 message data byte 1 register 09	C0MDATA109			√		Undefined
FA92H	CAN0 message data byte 23 register 09	C0MDATA2309				√	Undefined
FA92H	CAN0 message data byte 2 register 09	C0MDATA209			√		Undefined
FA93H	CAN0 message data byte 3 register 09	C0MDATA309			√		Undefined
FA94H	CAN0 message data byte 45 register 09	C0MDATA4509				√	Undefined
FA94H	CAN0 message data byte 4 register 09	C0MDATA409			√		Undefined
FA95H	CAN0 message data byte 5 register 09	C0MDATA509			√		Undefined
FA96H	CAN0 message data byte 67 register 09	C0MDATA6709				√	Undefined
FA96H	CAN0 message data byte 6 register 09	C0MDATA609			√		Undefined
FA97H	CAN0 message data byte 7 register 09	C0MDATA709			√		Undefined
FA98H	CAN0 message data length code register 09	C0MDLC09			√		0000xxxxB
FA99H	CAN0 message configuration register 09	C0MCONF09			√		Undefined
FA9AH	CAN0 message ID register 09	C0MIDL09				√	Undefined
FA9CH		C0MIDH09				√	Undefined
FA9EH	CAN0 message control register 09	C0MCTRL09				√	00x00000 000xx000B

Table 15-16. Register Access Types (6/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
FAA0H	CAN0 message data byte 01 register 10	C0MDATA0110	R/W			√	Undefined
FAA0H	CAN0 message data byte 0 register 10	C0MDATA010			√		Undefined
FAA1H	CAN0 message data byte 1 register 10	C0MDATA110			√		Undefined
FAA2H	CAN0 message data byte 23 register 10	C0MDATA2310				√	Undefined
FAA2H	CAN0 message data byte 2 register 10	C0MDATA210			√		Undefined
FAA3H	CAN0 message data byte 3 register 10	C0MDATA310			√		Undefined
FAA4H	CAN0 message data byte 45 register 10	C0MDATA4510				√	Undefined
FAA4H	CAN0 message data byte 4 register 10	C0MDATA410			√		Undefined
FAA5H	CAN0 message data byte 5 register 10	C0MDATA510			√		Undefined
FAA6H	CAN0 message data byte 67 register 10	C0MDATA6710				√	Undefined
FAA6H	CAN0 message data byte 6 register 10	C0MDATA610			√		Undefined
FAA7H	CAN0 message data byte 7 register 10	C0MDATA710			√		Undefined
FAA8H	CAN0 message data length code register 10	C0MDLC10			√		0000xxxxB
FAA9H	CAN0 message configuration register 10	C0MCONF10			√		Undefined
FAAAH	CAN0 message ID register 10	C0MIDL10				√	Undefined
FAACH		C0MIDH10				√	Undefined
FAAEH	CAN0 message control register 10	C0MCTRL10				√	00x00000 000xx000B
FAB0H	CAN0 message data byte 01 register 11	C0MDATA0111				√	Undefined
FAB0H	CAN0 message data byte 0 register 11	C0MDATA011			√		Undefined
FAB1H	CAN0 message data byte 1 register 11	C0MDATA111			√		Undefined
FAB2H	CAN0 message data byte 23 register 11	C0MDATA2311				√	Undefined
FAB2H	CAN0 message data byte 2 register 11	C0MDATA211			√		Undefined
FAB3H	CAN0 message data byte 3 register 11	C0MDATA311			√		Undefined
FAB4H	CAN0 message data byte 45 register 11	C0MDATA4511				√	Undefined
FAB4H	CAN0 message data byte 4 register 11	C0MDATA411			√		Undefined
FAB5H	CAN0 message data byte 51 register 11	C0MDATA511			√		Undefined
FAB6H	CAN0 message data byte 67 register 11	C0MDATA6711				√	Undefined
FAB6H	CAN0 message data byte 6 register 11	C0MDATA611			√		Undefined
FAB7H	CAN0 message data byte 71 register 11	C0MDATA711			√		Undefined
FAB8H	CAN0 message data length code register 11	C0MDLC11			√		0000xxxxB
FAB9H	CAN0 message configuration register 11	C0MCONF11			√		Undefined
FABAH	CAN0 message ID register 11	C0MIDL11				√	Undefined
FABCH		C0MIDH11				√	Undefined
FABEH	CAN0 message control register 11	C0MCTRL11				√	00x00000 000xx000B

Table 15-16. Register Access Types (7/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
FAC0H	CAN0 message data byte 01 register 12	C0MDATA0112	R/W			√	Undefined
FAC0H	CAN0 message data byte 0 register 12	C0MDATA012			√		Undefined
FAC1H	CAN0 message data byte 1 register 12	C0MDATA112			√		Undefined
FAC2H	CAN0 message data byte 23 register 12	C0MDATA2312				√	Undefined
FAC2H	CAN0 message data byte 2 register 12	C0MDATA212			√		Undefined
FAC3H	CAN0 message data byte 3 register 12	C0MDATA312			√		Undefined
FAC4H	CAN0 message data byte 45 register 12	C0MDATA4512				√	Undefined
FAC4H	CAN0 message data byte 4 register 12	C0MDATA412			√		Undefined
FAC5H	CAN0 message data byte 5 register 12	C0MDATA512			√		Undefined
FAC6H	CAN0 message data byte 67 register 12	C0MDATA6712				√	Undefined
FAC6H	CAN0 message data byte 6 register 12	C0MDATA612			√		Undefined
FAC7H	CAN0 message data byte 7 register 12	C0MDATA712			√		Undefined
FAC8H	CAN0 message data length code register 12	C0MDLC12			√		0000xxxxB
FAC9H	CAN0 message configuration register 12	C0MCONF12			√		Undefined
FACAH	CAN0 message ID register 12	C0MIDL12				√	Undefined
FACCH		C0MIDH12				√	Undefined
FACEH	CAN0 message control register 12	C0MCTRL12				√	00x00000 000xx000B
FAD0H	CAN0 message data byte 01 register 13	C0MDATA0113				√	Undefined
FAD0H	CAN0 message data byte 0 register 13	C0MDATA013			√		Undefined
FAD1H	CAN0 message data byte 1 register 13	C0MDATA113			√		Undefined
FAD2H	CAN0 message data byte 23 register 13	C0MDATA2313				√	Undefined
FAD2H	CAN0 message data byte 2 register 13	C0MDATA213			√		Undefined
FAD3H	CAN0 message data byte 3 register 13	C0MDATA313			√		Undefined
FAD4H	CAN0 message data byte 45 register 13	C0MDATA4513				√	Undefined
FAD4H	CAN0 message data byte 4 register 13	C0MDATA413			√		Undefined
FAD5H	CAN0 message data byte 5 register 13	C0MDATA513			√		Undefined
FAD6H	CAN0 message data byte 67 register 13	C0MDATA6713				√	Undefined
FAD6H	CAN0 message data byte 6 register 13	C0MDATA613			√		Undefined
FAD7H	CAN0 message data byte 7 register 13	C0MDATA713			√		Undefined
FAD8H	CAN0 message data length code register 13	C0MDLC13			√		0000xxxxB
FAD9H	CAN0 message configuration register 13	C0MCONF13			√		Undefined
FADAH	CAN0 message ID register 13	C0MIDL13				√	Undefined
FADCH		C0MIDH13				√	Undefined
FADEH	CAN0 message control register 13	C0MCTRL13				√	00x00000 000xx000B

Table 15-16. Register Access Types (8/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
FAE0H	CAN0 message data byte 01 register 14	C0MDATA0114	R/W			√	Undefined
FAE0H	CAN0 message data byte 0 register 14	C0MDATA014			√		Undefined
FAE1H	CAN0 message data byte 1 register 14	C0MDATA114			√		Undefined
FAE2H	CAN0 message data byte 23 register 14	C0MDATA2314				√	Undefined
FAE2H	CAN0 message data byte 2 register 14	C0MDATA214			√		Undefined
FAE3H	CAN0 message data byte 3 register 14	C0MDATA314			√		Undefined
FAE4H	CAN0 message data byte 45 register 14	C0MDATA4514				√	Undefined
FAE4H	CAN0 message data byte 4 register 14	C0MDATA414			√		Undefined
FAE5H	CAN0 message data byte 5 register 14	C0MDATA514			√		Undefined
FAE6H	CAN0 message data byte 67 register 14	C0MDATA6714				√	Undefined
FAE6H	CAN0 message data byte 6 register 14	C0MDATA614			√		Undefined
FAE7H	CAN0 message data byte 7 register 14	C0MDATA714			√		Undefined
FAE8H	CAN0 message data length code register 14	C0MDLC14			√		0000xxxxB
FAE9H	CAN0 message configuration register 14	C0MCONF14			√		Undefined
FAEAH	CAN0 message ID register 14	C0MIDL14				√	Undefined
FAECH		C0MIDH14				√	Undefined
FAEEH	CAN0 message control register 14	C0MCTRL14				√	00x00000 000xx000B
FAF0H	CAN0 message data byte 01 register 15	C0MDATA0115				√	Undefined
FAF0H	CAN0 message data byte 0 register 15	C0MDATA015			√		Undefined
FAF1H	CAN0 message data byte 1 register 15	C0MDATA115			√		Undefined
FAF2H	CAN0 message data byte 23 register 15	C0MDATA2315				√	Undefined
FAF2H	CAN0 message data byte 2 register 15	C0MDATA215			√		Undefined
FAF3H	CAN0 message data byte 3 register 15	C0MDATA315			√		Undefined
FAF4H	CAN0 message data byte 45 register 15	C0MDATA4515				√	Undefined
FAF4H	CAN0 message data byte 4 register 15	C0MDATA415			√		Undefined
FAF5H	CAN0 message data byte 5 register 15	C0MDATA515			√		Undefined
FAF6H	CAN0 message data byte 67 register 15	C0MDATA6715				√	Undefined
FAF6H	CAN0 message data byte 6 register 15	C0MDATA615			√		Undefined
FAF7H	CAN0 message data byte 7 register 15	C0MDATA715			√		Undefined
FAF8H	CAN0 message data length code register 15	C0MDLC15			√		0000xxxxB
FAF9H	CAN0 message configuration register 15	C0MCONF15			√		Undefined
FAFAH	CAN0 message ID register 15	C0MIDL15				√	Undefined
FAFCH		C0MIDH15				√	Undefined
FAFEH	CAN0 message control register 15	C0MCTRL15				√	00x00000 000xx000B

Table 15-16. Register Access Types (9/9)

Address	Register Name	Symbol	R/W	Bit Manipulation Units			Default Value
				1	8	16	
FF60H	CAN0 module receive history list register	C0RGPT	R/W	–	–	√	xx02H
FF62H	CAN0 module transmit history list register	C0TGPT	R/W	–	–	√	xx02H
FF64H	CAN0 global control register	C0GMCTRL	R/W	–	–	√	0000H
FF66H	CAN0 global automatic block transmission control register	C0GMABT	R/W	–	–	√	0000H
FF68H	CAN0 module last out-pointer register	C0LOPT	R	–	√	–	Undefined
FF6EH	CAN0 global clock select register	C0GMCS	R/W	–	√	–	0FH
FF6FH	CAN0 global automatic block transmission delay setting register	C0GMABTD	R/W	–	√	–	00H
FF70H	CAN0 module mask 1 register	C0MASK1L	R/W	–	–	√	Undefined
FF72H		C0MASK1H					
FF74H	CAN0 module mask 2 register	C0MASK2L	R/W	–	–	√	Undefined
FF76H		C0MASK2H					
FF78H	CAN0 module mask 3 register	C0MASK3L	R/W	–	–	√	Undefined
FF7AH		C0MASK3H					
FF7CH	CAN0 module mask 4 register	C0MASK4L	R/W	–	–	√	Undefined
FF7EH		C0MASK4H					
FF8AH	CAN0 module time stamp register	C0TS	R/W	–	–	√	0000H
FF90H	CAN0 module control register	C0CTRL	R/W	–	–	√	0000H
FF92H	CAN0 module last error information register	C0LEC	R/W	–	√	–	00H
FF93H	CAN0 module information register	C0INFO	R	–	√	–	00H
FF94H	CAN0 module error counter register	C0ERC	R	–	–	√	0000H
FF96H	CAN0 module interrupt enable register	C0IE	R/W	–	–	√	0000H
FF98H	CAN0 module interrupt status register	C0INTS	R/W	–	–	√	0000H
FF9CH	CAN0 module bit rate register	C0BTR	R/W	–	–	√	370FH
FF9EH	CAN0 module bit rate prescaler register	C0BRP	R/W	–	√	–	FFH
FF9FH	CAN0 module last in-pointer register	C0LIPT	R	–	√	–	Undefined

15.5.3 Register bit configuration

Table 15-17. Bit Configuration of CAN Global Registers

Address	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
FF64H	C0GMCTRL(W)	0	0	0	0	0	0	0	Clear GOM
FF65H		0	0	0	0	0	0	Set EFSD	Set GOM
FF64H	C0GMCTRL(R)	0	0	0	0	0	0	EFSD	GOM
FF65H		MBON	0	0	0	0	0	0	0
FF66H	C0GMABT(W)	0	0	0	0	0	0	0	Clear ABTTRG
FF67H		0	0	0	0	0	0	Set ABTCLR	Set ABTTRG
FF66H	C0GMABT(R)	0	0	0	0	0	0	ABTCLR	ABTTRG
FF67H		0	0	0	0	0	0	0	0
FF6EH	C0GMCS	0	0	0	0	CCP3	CCP2	CCP1	CCP0
FF6FH	C0GMABTD	0	0	0	0	ABTD3	ABTD2	ABTD1	ABTD0

Caution The actual register address is calculated as follows:
Register Address = Global Register Area Offset (CH dependent) + Offset Address as listed in the table above

Remark (R) When read
(W) When write

Table 15-18. Bit Configuration of CAN Module Registers (1/2)

<table>
<tr><th>Address</th><th>Symbol</th><th>Bit 7/15</th><th>Bit 6/14</th><th>Bit 5/13</th><th>Bit 4/12</th><th>Bit 3/11</th><th>Bit 2/10</th><th>Bit 1/9</th><th>Bit 0/8</th></tr>
<tr><td>FF60H</td><td rowspan="2">C0RGPT(W)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Clear
ROVF</td></tr>
<tr><td>FF61H</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>FF60H</td><td rowspan="2">C0RGPT(R)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>RHPM</td><td>ROVF</td></tr>
<tr><td>FF61H</td><td colspan="8">RGPT [7:0]</td></tr>
<tr><td>FF62H</td><td>C0LOPT</td><td colspan="8">LOPT [7:0]</td></tr>
<tr><td>FF64H</td><td rowspan="2">C0TGPT(W)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Clear
TOVF</td></tr>
<tr><td>FF65H</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>FF64H</td><td rowspan="2">C0TGPT(R)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>THPM</td><td>TOVF</td></tr>
<tr><td>FF65H</td><td colspan="8">TGPT [7:0]</td></tr>
<tr><td>FF70H</td><td rowspan="2">C0MASK1L</td><td colspan="8">CM1ID [7:0]</td></tr>
<tr><td>FF71H</td><td colspan="8">CM1ID [15:8]</td></tr>
<tr><td>FF72H</td><td rowspan="2">C0MASK1H</td><td colspan="8">CM1ID [23:16]</td></tr>
<tr><td>FF73H</td><td>0</td><td>0</td><td>0</td><td colspan="5">CM1ID [28:24]</td></tr>
<tr><td>FF74H</td><td rowspan="2">C0MASK2L</td><td colspan="8">CM2ID [7:0]</td></tr>
<tr><td>FF75H</td><td colspan="8">CM2ID [15:8]</td></tr>
<tr><td>FF76H</td><td rowspan="2">C0MASK2H</td><td colspan="8">CM2ID [23:16]</td></tr>
<tr><td>FF77H</td><td>0</td><td>0</td><td>0</td><td colspan="5">CM2ID [28:24]</td></tr>
<tr><td>FF78H</td><td rowspan="2">C0MASK3L</td><td colspan="8">CM3ID [7:0]</td></tr>
<tr><td>FF79H</td><td colspan="8">CM3ID [15:8]</td></tr>
<tr><td>FF7AH</td><td rowspan="2">C0MASK3H</td><td colspan="8">CM3ID [23:16]</td></tr>
<tr><td>FF7BH</td><td>0</td><td>0</td><td>0</td><td colspan="5">CM3ID [28:24]</td></tr>
<tr><td>FF7CH</td><td rowspan="2">C0MASK4L</td><td colspan="8">CM4ID [7:0]</td></tr>
<tr><td>FF7DH</td><td colspan="8">CM4ID [15:8]</td></tr>
<tr><td>FF7EH</td><td rowspan="2">C0MASK4H</td><td colspan="8">CM4ID [23:16]</td></tr>
<tr><td>FF7FH</td><td>0</td><td>0</td><td>0</td><td colspan="5">CM4ID [28:24]</td></tr>
<tr><td>FF8AH</td><td rowspan="2">C0TS(W)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Clear
TSLOCK</td><td>Clear
TSSEL</td><td>Clear
TSEN</td></tr>
<tr><td>FF8BH</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Set
TSLOCK</td><td>Set
TSSEL</td><td>Set
TSEN</td></tr>
<tr><td>FF8AH</td><td rowspan="2">C0TS(R)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>TSLOCK</td><td>TSSEL</td><td>TSEN</td></tr>
<tr><td>FF8BH</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

Caution The actual register address is calculated as follows:
Register Address = Global Register Area Offset (CH dependent) + Offset Address as listed in the table above

Remark (R) When read
(W) When write

Table 15-18. Bit Configuration of CAN Module Registers (2/2)

Address	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
FF90H	C0CTRL(W)	Clear CCERC	Clear AL	Clear VALID	Clear PSMODE1	Clear PSMODE0	Clear OPMODE2	Clear OPMODE1	Clear OPMODE0
FF91H		Set CCERC	Set AL	0	Set PSMODE1	Set PSMODE0	Set OPMODE2	Set OPMODE1	Set OPMODE0
FF90H	C0CTRL(R)	CCERC	AL	VALID	PSMODE1	PSMODE0	OPMODE2	OPMODE1	OPMODE0
FF91H		0	0	0	0	0	0	RSTAT	TSTAT
FF92H	C0LEC(W)	0	0	0	0	0	0	0	0
FF92H	C0LEC(R)	0	0	0	0	0	LEC2	LEC1	LEC0
FF93H	C0INFO	0	0	0	BOFF	TECS1	TECS0	RECS1	RECS0
FF94H	C0ERC	TEC [7:0]							
FF95H		REC [7:0]							
FF96H	C0IE(W)	0	0	Clear CIE5	Clear CIE4	Clear CIE3	Clear CIE2	Clear CIE1	Clear CIE0
FF97H		0	0	Set CIE5	Set CIE4	Set CIE3	Set CIE2	Set CIE1	Set CIE0
FF96H	C0IE(R)	0	0	CIE5	CIE4	CIE3	CIE2	CIE1	CIE0
FF97H		0	0	0	0	0	0	0	0
FF98H	C0INTS(W)	0	0	Clear CINTS5	Clear CINTS4	Clear CINTS3	Clear CINTS2	Clear CINTS1	Clear CINTS0
FF99H		0	0	0	0	0	0	0	0
FF98H	C0INTS(R)	0	0	CINTS5	CINTS4	CINTS3	CINTS2	CINTS1	CINTS0
FF99H		0	0	0	0	0	0	0	0
FF9CH	C0BTR	0	0	0	0	TSEG1 [3:0]			
FF9DH		0	0	SJW [1:0]		0	TSEG2 [2:0]		
FF9EH	C0BRP	TQPRS [7:0]							
FF9FH	C0LIPT	LIPT [7:0]							

Caution The actual register address is calculated as follows:

Register Address = Global Register Area Offset (CH dependent) + Offset Address as listed in the table above

Remark (R) When read

(W) When write

Table 15-19. Bit Configuration of Message Buffer Registers

Address	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
FAx0H	C0MDATA01m	Message data (byte 0)							
FAx1H		Message data (byte 1)							
FAx0H	C0MDATA0m	Message data (byte 0)							
FAx1H	C0MDATA1m	Message data (byte 1)							
FAx2H	C0MDATA23m	Message data (byte 2)							
FAx3H		Message data (byte 3)							
FAx2H	C0MDATA2m	Message data (byte 2)							
FAx3H	C0MDATA3m	Message data (byte 3)							
FAx4H	C0MDATA45m	Message data (byte 4)							
FAx5H		Message data (byte 5)							
FAx4H	C0MDATA4m	Message data (byte 4)							
FAx5H	C0MDATA5m	Message data (byte 5)							
FAx6H	C0MDATA67m	Message data (byte 6)							
FAx7H		Message data (byte 7)							
FAx6H	C0MDATA6m	Message data (byte 6)							
FAx7H	C0MDATA7m	Message data (byte 7)							
FAx8H	C0MDLCm	0	0	0	0	MDLC3	MDLC2	MDLC1	MDLC0
FAx9H	C0MCONFm	OWS	RTR	MT2	MT1	MT0	0	0	MA0
FAxAH	C0MIDLm	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
FAxBH		ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
FAxCH	C0MIDHm	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16
FAxDH		IDE	0	0	ID28	ID27	ID26	ID25	ID24
FAxEH	C0MCTRLm (W)	0	0	0	Clear MOW	Clear IE	Clear DN	Clear TRQ	Clear RDY
FAxFH		0	0	0	0	Set IE	0	Set TRQ	Set RDY
FAxEH	C0MCTRLm (R)	0	0	0	MOW	IE	DN	TRQ	RDY
FAxFH		0	0	MUC	0	0	0	0	0

Caution The actual register address is calculated as follows:
Register Address = Global Register Area Offset (CH dependent) + Offset Address as listed in the table above

Remarks 1. (R) When read
(W) When write

2. m = 0 to 15

15.6 Bit Set/Clear Function

The CAN control registers include registers whose bits can be set or cleared via the CPU and via the CAN interface. An operation error occurs if the following registers are written directly. Do not write any values directly via bit manipulation, read/modify/write, or direct writing of target values.

- CAN global control register (C0GMCTRL)
- CAN global automatic block transmission control register (C0GMABT)
- CAN module control register (C0CTRL)
- CAN module interrupt enable register (C0IE)
- CAN module interrupt status register (C0INTS)
- CAN module receive history list register (C0RGPT)
- CAN module transmit history list register (C0TGPT)
- CAN module time stamp register (C0TS)
- CAN message control register (C0MCTRLm)

Remark m = 0 to 15

All the 16 bits in the above registers can be read via the usual method. Use the procedure described in **Figure 15-23** below to set or clear the lower 8 bits in these registers.

Setting or clearing of lower 8 bits in the above registers is performed in combination with the higher 8 bits (refer to the 16-bit data after a write operation in **Figure 15-24**). **Figure 15-23** shows how the values of set bits or clear bits relate to set/clear/no change operations in the corresponding register.

Figure 15-23. Example of Bit Setting/Clearing Operations

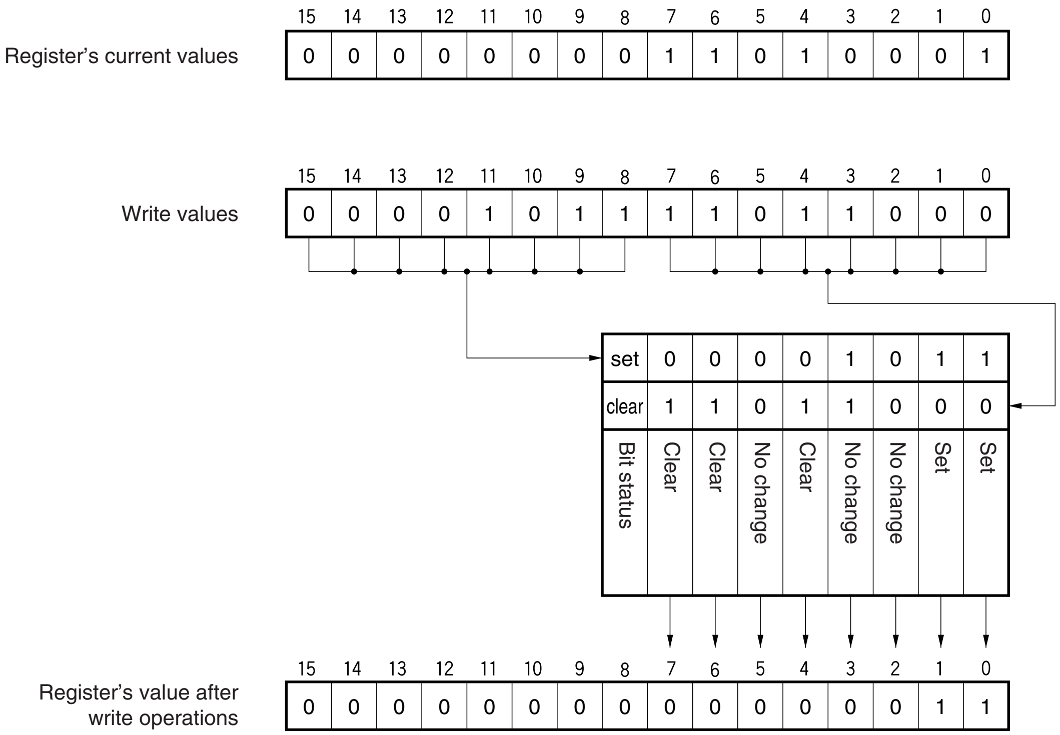


Figure 15-24. 16-Bit Data during Write Operation

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
set 7	set 6	set 5	set 4	set 3	set 2	set 1	set 0	clear 7	clear 6	clear 5	clear 4	clear 3	clear 2	clear 1	clear 0

set n	clear n	Status of bit n after bit set/clear operation
0	0	No change
0	1	0
1	0	1
1	1	No change

Remark n = 0 to 7

15.7 Control Registers

Remark m = 0 to 15

(1) CAN global control register (C0GMCTRL)

The C0GMCTRL register is used to control the operation of the CAN module.

After reset: 0000H R/W Address: FF4CH, FF4DH

(a) Read

	15	14	13	12	11	10	9	8
C0GMCTRL	MBON	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	EFSD	GOM

(b) Write

	15	14	13	12	11	10	9	8
C0GMCTRL	0	0	0	0	0	0	Set EFSD	Set GOM
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Clear GOM

(a) Read

MBON	Bit Enabling Access to Message Buffer Register, Transmit/Receive History List Registers
0	Write access and read access to the message buffer register and the transmit/receive history list registers is disabled.
1	Write access and read access to the message buffer register and the transmit/receive history list registers is enabled.

Cautions 1. While the MBON bit is cleared (to 0), software access to the message buffers (C0MDATA0m, C0MDATA1m, C0MDATA01m, C0MDATA2m, C0MDATA3m, C0MDATA23m, C0MDATA4m, C0MDATA5m, C0MDATA45m, C0MDATA6m, C0MDATA7m, C0MDATA67m, C0MDLCm, C0MCONFm, C0MIDLm, C0MIDHm, and C0MCTRLm), or registers related to transmit history or receive history (C0LOPT, C0TGPT, C0LIPT, and C0RGPT) is disabled.

2. This bit is read-only. Even if 1 is written to MBON while it is 0, the value of MBON does not change, and access to the message buffer registers, or registers related to transmit history or receive history remains disabled.

Remark MBON bit is cleared (to 0) when the CAN module enters CAN sleep mode/CAN stop mode or GOM bit is cleared (to 0).

MBON bit is set (to 1) when the CAN sleep mode/the CAN stop mode is released or GOM bit is set (to 1).

EFSD	Bit Enabling Forced Shut Down
0	Forced shut down by GOM = 0 disabled.
1	Forced shut down by GOM = 0 enabled.

Caution To request forced shutdown, the GOM bit must be cleared to 0 in a subsequent, immediately following write access after the EFSD bit has been set to 1. If access to another register (including reading the C0GMCTRL register) is executed without clearing the GOM bit immediately after the EFSD bit has been set to 1, the EFSD bit is forcibly cleared to 0, and the forced shutdown request is invalid.

GOM	Global Operation Mode Bit
0	CAN module is disabled from operating.
1	CAN module is enabled to operate.

Caution The GOM bit can be cleared only in the initialization mode or immediately after EFSD bit is set (to 1).

(b) Write

Set EFSD	EFSD Bit Setting
0	No change in ESFD bit .
1	EFSD bit set to 1.

Set GOM	Clear GOM	GOM Bit Setting
0	1	GOM bit cleared to 0.
1	0	GOM bit set to 1.
Other than above		No change in GOM bit.

Caution Set GOM bit and ESFD bit always separately.

(2) CAN global clock selection register (C0GMCS)

The C0GMCS register is used to select the CAN module system clock.

After reset: 0FH R/W Address: FF6EH

	7	6	5	4	3	2	1	0
C0GMCS	0	0	0	0	CCP3	CCP2	CCP1	CCP0

CCP3	CCP2	CCP1	CCP1	CAN Module System Clock (f_{CANMOD})
0	0	0	0	$f_{\text{CAN}}/1$
0	0	0	1	$f_{\text{CAN}}/2$
0	0	1	0	$f_{\text{CAN}}/3$
0	0	1	1	$f_{\text{CAN}}/4$
0	1	0	0	$f_{\text{CAN}}/5$
0	1	0	1	$f_{\text{CAN}}/6$
0	1	1	0	$f_{\text{CAN}}/7$
0	1	1	1	$f_{\text{CAN}}/8$
1	0	0	0	$f_{\text{CAN}}/9$
1	0	0	1	$f_{\text{CAN}}/10$
1	0	1	0	$f_{\text{CAN}}/11$
1	0	1	1	$f_{\text{CAN}}/12$
1	1	0	0	$f_{\text{CAN}}/13$
1	1	0	1	$f_{\text{CAN}}/14$
1	1	1	0	$f_{\text{CAN}}/15$
1	1	1	1	$f_{\text{CAN}}/16$ (Default value)

Remark f_{CAN} = Clock supplied to CAN

(3) CAN global automatic block transmission control register (C0GMABT)

The C0GMABT register is used to control the automatic block transmission (ABT) operation.

After reset: 0000H R/W Address: FFAEH, FFAFH

(a) Read

	15	14	13	12	11	10	9	8
C0GMABT	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	ABTCLR	ABTTRG

(b) Write

	15	14	13	12	11	10	9	8
C0GMABT	0	0	0	0	0	0	Set ABTCLR	Set ABTTRG
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Clear ABTTRG

Caution Before changing the normal operation mode with ABT to the initialization mode, be sure to set the C0GMABT register to the default value (0000H) and confirm the C0GMABT register is surely initialized to the default value (0000H).

(a) Read

ABTCLR	Automatic Block Transmission Engine Clear Status Bit
0	Clearing the automatic transmission engine is completed.
1	The automatic transmission engine is being cleared.

Remarks 1. Set the ABTCLR bit to 1 while the ABTTRG bit is cleared (0).
The operation is not guaranteed if the ABTCLR bit is set to 1 while the ABTTRG bit is set to 1.

2. When the automatic block transmission engine is cleared by setting the ABTCLR bit to 1, the ABTCLR bit is automatically cleared to 0 as soon as the requested clearing processing is complete.

ABTTRG	Automatic Block Transmission Status Bit
0	Automatic block transmission is stopped.
1	Automatic block transmission is under execution.

Caution Do not set the ABTTRG bit (ABTTRG = 1) in the initialization mode. If the ABTTRG bit is set in the initialization mode, the operation is not guaranteed after the CAN module has entered the normal operation mode with ABT.
Do not set the ABTTRG bit (1) while the C0CTRL.TSTAT bit is set (1). Confirm TSTAT = 0 directly in advance before setting ABTTRG bit.

(b) Write

Set ABTCLR	Automatic Block Transmission Engine Clear Request Bit
0	The automatic block transmission engine is in idle state or under operation.
1	Request to clear the automatic block transmission engine. After the automatic block transmission engine has been cleared, automatic block transmission is started from message buffer 0 by setting the ABTTRG bit to 1.

Set ABTTRG	Clear ABTTRG	Automatic Block Transmission Start Bit
0	1	Request to stop automatic block transmission.
1	0	Request to start automatic block transmission.
Other than above		No change in ABTTRG bit.

Caution While receiving a message from another node or transmitting the messages other than the ABT messages (message buffer 8 to 15), there is a possibility not to begin immediately the transmission even if the ABTTRG bit is set to 1. Transmission is not aborted even if the ABTTRG bit is cleared to 0, until the transmission of the ABT message, which is currently being transmitted is completed (successfully or not). After that, the transmission is aborted.

(4) CAN global automatic block transmission delay setting register (C0GMABTD)

The C0GMABTD register is used to set the interval at which the data of the message buffer assigned to ABT is to be transmitted in the normal operation mode with ABT.

After reset: 00H R/W Address: FF6FH

	7	6	5	4	3	2	1	0
C0GMABTD	0	0	0	0	ABTD3	ABTD2	ABTD1	ABTD0

ABTD3	ABTD2	ABTD1	ABTD0	Data frame interval during automatic block transmission (unit: Data bit time (DBT))
0	0	0	0	0 DBT (default value)
0	0	0	1	2^5 DBT
0	0	1	0	2^6 DBT
0	0	1	1	2^7 DBT
0	1	0	0	2^8 DBT
0	1	0	1	2^9 DBT
0	1	1	0	2^{10} DBT
0	1	1	1	2^{11} DBT
1	0	0	0	2^{12} DBT
Other than above				Setting prohibited

Cautions 1. Do not change the contents of the C0GMABTD register while the ABTTRG bit is set to 1.

2. The timing at which the ABT message is actually transmitted onto the CAN bus differs depending on the status of transmission from the other station or how a request to transmit a message other than an ABT message (message buffers 8 to 15) is made.

(5) CAN module mask control register (C0MASKaL, C0MASKaH) (a = 1, 2, 3, or 4)

The C0MASKaL and C0MASKaH registers are used to extend the number of receivable messages into the same message buffer by masking part of the ID comparison of a message and invalidating the ID of the masked part.

- CAN Module Mask 1 Register (C0MASK1L, C0MASK1H)

After reset: Undefined   R/W   Address: C0MASK1L FF70H, FF71H
C0MASK1H FF72H, FF73H

	15	14	13	12	11	10	9	8
C0MASK1L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0
	15	14	13	12	11	10	9	8
C0MASK1H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

- CAN Module Mask 2 Register (C0MASK2L, C0MASK2H)

After reset: Undefined   R/W   Address: C0MASK2L FF74H, FF75H
C0MASK2H FF76H, FF77H

	15	14	13	12	11	10	9	8
C0MASK2L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0
	15	14	13	12	11	10	9	8
C0MASK2H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

- CAN Module Mask 3 Register (C0MASK3L, C0MASK3H)

After reset: Undefined R/W Address: C0MASK3L FF78H, FF79H
C0MASK3H FF7AH, FF7BH

	15	14	13	12	11	10	9	8
C0MASK3L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0
	15	14	13	12	11	10	9	8
C0MASK3H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

- CAN Module Mask 4 Register (C0MASK4L, C0MASK4H)

After reset: Undefined R/W Address: C0MASK4L FF7CH, FF7DH
C0MASK4H FF7EH, FF7FH

	15	14	13	12	11	10	9	8
C0MASK4L	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
	7	6	5	4	3	2	1	0
	CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0
	15	14	13	12	11	10	9	8
C0MASK4H	0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
	7	6	5	4	3	2	1	0
	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

CMID28-CMID0	Sets Mask Pattern of ID Bit.
0	The ID bits of the message buffer set by the CMID28 to CMID0 bits are compared with the ID bits of the received message frame.
1	The ID bits of the message buffer set by the CMID28 to CMID0 bits are not compared with the ID bits of the received message frame (they are masked).

Remark Masking is always defined by an ID length of 29 bits. If a mask is assigned to a message with a standard ID, CMID17 to CMID0 are ignored. Therefore, only CMID28 to CMID18 of the received ID are masked. The same mask can be used for both the standard and extended IDs.

(6) CAN module control register (C0CTRL)

The C0CTRL register is used to control the operation mode of the CAN module.

After reset: 0000H R/W Address: FF90H, FF91H

(a) Read

	15	14	13	12	11	10	9	8
C0CTRL	0	0	0	0	0	0	RSTAT	TSTAT
	7	6	5	4	3	2	1	0
	CCERC	AL	VALID	PSMODE1	PSMODE0	OPMODE2	OPMODE1	OPMODE0

(b) Write

	15	14	13	12	11	10	9	8
C0CTRL	Set CCERC	Set AL	0	Set PSMODE1	Set PSMODE0	Set OPMODE2	Set OPMODE1	Set OPMODE0
	7	6	5	4	3	2	1	0
	Clear CCERC	Clear AL	Clear VALID	Clear PSMODE1	Clear PSMODE0	Clear OPMODE2	Clear OPMODE1	Clear OPMODE0

(a) Read

RSTAT	Reception Status Bit
0	Reception is stopped.
1	Reception is in progress.

Remark - The RSTAT bit is set to 1 under the following conditions (timing).

- The SOF bit of a receive frame is detected
- On occurrence of arbitration loss during a transmit frame

- The RSTAT bit is cleared to 0 under the following conditions (timing)

- When a recessive level is detected at the second bit of the interframe space
- On transition to the initialization mode at the first bit of the interframe space

TSTAT	Transmission Status Bit
0	Transmission is stopped.
1	Transmission is in progress.

Remark - The TSTAT bit is set to 1 under the following conditions (timing).

- The SOF bit of a transmit frame is detected

- The TSTAT bit is cleared to 0 under the following conditions (timing).

- During transition to bus-off state
- On occurrence of arbitration loss in transmit frame
- On detection of recessive level at the second bit of the interframe space
- On transition to the initialization mode at the first bit of the interframe space

CCERC	Error Counter Clear Bit
0	The C0ERC and C0INFO registers are not cleared in the initialization mode.
1	The C0ERC and C0INFO registers are cleared in the initialization mode.

Remarks 1. The CCERC bit is used to clear the C0ERC and C0INFO registers for re-initialization or forced recovery from the bus-off state. This bit can be set to 1 only in the initialization mode.

2. When the C0ERC and C0INFO registers have been cleared, the CCERC bit is also cleared to 0 automatically.

3. The CCERC bit can be set to 1 at the same time as a request to change the initialization mode to an operation mode is made.

4. The receive data may be corrupted in case of setting the CCERC bit to (1) immediately after entering the INIT mode from self-test mode.

AL	Bit to Set Operation in Case of Arbitration Loss
0	Re-transmission is not executed in case of an arbitration loss in the single-shot mode.
1	Re-transmission is executed in case of an arbitration loss in the single-shot mode.

Remark The AL bit is valid only in the single-shot mode.

VALID	Valid Receive Message Frame Detection Bit
0	A valid message frame has not been received since the VALID bit was last cleared to 0.
1	A valid message frame has been received since the VALID bit was last cleared to 0.

Remarks 1. Detection of a valid receive message frame is not dependent upon storage in the receive message buffer (data frame) or transmit message buffer (remote frame).

2. Clear the VALID bit (0) before changing the initialization mode to an operation mode.

3. If only two CAN nodes are connected to the CAN bus with one transmitting a message frame in the normal operation mode and the other in the receive-only mode, the VALID bit is not set to 1 before the transmitting node enters the error passive state, because in receive-only mode no acknowledge is generated.

4. In order to clear the VALID bit, set the Clear VALID bit to 1 first and confirm that the VALID bit is cleared. If it is not cleared, perform clearing processing again.

PSMODE1	PSMODE0	Power Save Mode
0	0	No power save mode is selected.
0	1	CAN sleep mode
1	0	Setting prohibited
1	1	CAN stop mode

Cautions1. Transition to and from the CAN stop mode must be made via CAN sleep mode. A request for direct transition to and from the CAN stop mode is ignored.

2. The MBON flag of C0GMCTRL must be checked after releasing a power save mode, prior to access the message buffers again.

3. CAN Sleep mode requests are kept pending, until cancelled by software or entered on appropriate bus condition (bus idle). Software can check the actual status by reading PSMODE.

OPMODE2	OPMODE1	OPMODE0	Operation Mode
0	0	0	No operation mode is selected (CAN module is in the initialization mode).
0	0	1	Normal operation mode
0	1	0	Normal operation mode with automatic block transmission function (normal operation mode with ABT)
0	1	1	Receive-only mode
1	0	0	Single-shot mode
1	0	1	Self-test mode
Other than above			Setting prohibited

Caution Transit to initialization mode or power saving modes may take some time. Be sure to verify the success of mode change by reading the values, before proceeding.

Remark The OPMODE[2:0] bits are read-only in the CAN sleep mode or CAN stop mode.

(b) Write

Set CCERC	Clear CCERC	Setting of CCERC Bit
1	1	CCERC bit is set to 1.
Other than above	0	CCERC bit is not changed.

Set AL	Clear AL	Setting of AL Bit
0	1	AL bit is cleared to 0.
1	0	AL bit is set to 1.
Other than above		AL bit is not changed.

Clear VALID	Setting of VALID Bit
0	VALID bit is not changed.
1	VALID bit is cleared to 0.

Set PSMODE0	Clear PSMODE0	Setting of PSMODE0 Bit
0	1	PSMODE0 bit is cleared to 0.
1	0	PSMODE bit is set to 1.
Other than above		PSMODE0 bit is not changed.

Set PSMODE1	Clear PSMODE1	Setting of PSMODE1 Bit
0	1	PSMODE1 bit is cleared to 0.
1	0	PSMODE1 bit is set to 1.
Other than above		PSMODE1 bit is not changed.

Set OPMODE0	Clear OPMODE0	Setting of OPMODE0 Bit
0	1	OPMODE0 bit is cleared to 0.
1	0	OPMODE0 bit is set to 1.
Other than above		OPMODE0 bit is not changed.

Set OPMODE1	Clear OPMODE1	Setting of OPMODE1 Bit
0	1	OPMODE1 bit is cleared to 0.
1	0	OPMODE1 bit is set to 1.
Other than above		OPMODE1 bit is not changed.

Set OPMODE2	Clear OPMODE2	Setting of OPMODE2 Bit
0	1	OPMODE2 bit is cleared to 0.
1	0	OPMODE2 bit is set to 1.
Other than above		OPMODE2 bit is not changed.

(7) CAN module last error code register (C0LEC)

The C0LEC register provides the error information of the CAN protocol.

After reset: 00H R/W Address: FF92H

	7	6	5	4	3	2	1	0
C0LEC	0	0	0	0	0	LEC2	LEC1	LEC0

Remarks 1. The contents of the C0LEC register are not cleared when the CAN module changes from an operation mode to the initialization mode.

2. If an attempt is made to write a value other than 00H to the C0LEC register by software, the access is ignored.

LEC2	LEC1	LEC0	Last CAN Protocol Error Information
0	0	0	No error
0	0	1	Stuff error
0	1	0	Form error
0	1	1	ACK error
1	0	0	Bit error (The CAN module tried to transmit a recessive-level bit as part of a transmit message (except the arbitration field), but the value on the CAN bus is a dominant-level bit.)
1	0	1	Bit error (The CAN module tried to transmit a dominant-level bit as part of a transmit message, ACK bit, error frame, or overload frame, but the value on the CAN bus is a recessive-level bit.)
1	1	0	CRC error
1	1	1	Undefined

(8) CAN module information register (C0INFO)

The C0INFO register indicates the status of the CAN module.

After reset: 00H R Address: FF93H

	7	6	5	4	3	2	1	0
C0INFO	0	0	0	BOFF	TECS1	TECS0	RECS1	RECS0

BOFF	Bus-off State Bit
0	Not bus-off state (transmit error counter ≤ 255) (The value of the transmit counter is less than 256.)
1	Bus-off state (transmit error counter > 255) (The value of the transmit counter is 256 or more.)

TECS1	TECS0	Transmission Error Counter Status Bit
0	0	The value of the transmission error counter is less than that of the warning level (<96).
0	1	The value of the transmission error counter is in the range of the warning level (96 to 127).
1	0	Undefined
1	1	The value of the transmission error counter is in the range of the error passive or bus-off state (≥ 128).

RECS1	RECS0	Reception Error Counter Status Bit
0	0	The value of the reception error counter is less than that of the warning level (<96).
0	1	The value of the reception error counter is in the range of the warning level (96 to 127).
1	0	Undefined
1	1	The value of the reception error counter is in the error passive range (≥ 128).

(9) CAN module error counter register (C0ERC)

The C0ERC register indicates the count value of the transmission/reception error counter.

After reset: 0000H R Address: FF94H, FF95H

	15	14	13	12	11	10	9	8
C0ERC	REPS	REC6	REC5	REC4	REC3	REC2	REC1	REC0
	7	6	5	4	3	2	1	0
	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0

REPS	Reception error passive status bit
0	Reception error counter is not error passive (<128)
1	Reception error counter is error passive range (≥128)

REC6-REC0	Reception Error Counter Bit
0-127	Number of reception errors. These bits reflect the status of the reception error counter. The number of errors is defined by the CAN protocol.

Remark REC6 to REC0 of the reception error counter are invalid in the reception error passive state (RECS [1:0] = 11B).

TEC7-TEC0	Transmission Error Counter Bit
0-255	Number of transmission errors. These bits reflect the status of the transmission error counter. The number of errors is defined by the CAN protocol.

Remark TEC7 to TEC0 of the transmission error counter are invalid in the bus-off state (BOFF = 1).

(10) CAN module interrupt enable register (C0IE)

The C0IE register is used to enable or disable the interrupts of the CAN module.

After reset: 0000H R/W Address: FF96H, FF97H

(a) Read

	15	14	13	12	11	10	9	8
C0IE	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	CIE5	CIE4	CIE3	CIE2	CIE1	CIE0

(b) Write

	15	14	13	12	11	10	9	8
C0IE	0	0	Set CIE5	Set CIE4	Set CIE3	Set CIE2	Set CIE1	Set CIE0
	7	6	5	4	3	2	1	0
	0	0	Clear CIE5	Clear CIE4	Clear CIE3	Clear CIE2	Clear CIE1	Clear CIE0

(a) Read

CIE5-CIE0	CAN Module Interrupt Enable Bit
0	Output of the interrupt corresponding to interrupt status register CINTS5 to CINTS0 bits is disabled.
1	Output of the interrupt corresponding to interrupt status register CINTS5 to CINTS0 bits is enabled.

(b) Write

Set CIE5	Clear CIE5	Setting of CIE5 Bit
0	1	CIE5 bit is cleared to 0.
1	0	CIE5 bit is set to 1.
Other than above		CIE5 bit is not changed.

Set CIE4	Clear CIE4	Setting of CIE4 Bit
0	1	CIE4 bit is cleared to 0.
1	0	CIE4 bit is set to 1.
Other than above		CIE4 bit is not changed.

Set CIE3	Clear CIE3	Setting of CIE Bit
0	1	CIE3 bit is cleared to 0.
1	0	CIE3 bit is set to 1.
Other than above		CIE3 bit is not changed.

Set CIE2	Clear CIE2	Setting of CIE2 Bit
0	1	CIE2 bit is cleared to 0.
1	0	CIE2 bit is set to 1.
Other than above		CIE2 bit is not changed.

Set CIE1	Clear CIE1	Setting of CIE1 Bit
0	1	CIE1 bit is cleared to 0.
1	0	CIE1 bit is set to 1.
Other than above		CIE1 bit is not changed.

Set CIE0	Clear CIE0	Setting of CIE0 Bit
0	1	CIE0 bit is cleared to 0.
1	0	CIE0 bit is set to 1.
Other than above		CIE0 bit is not changed.

(11) CAN module interrupt status register (C0INTS)

The C0INTS register indicates the interrupt status of the CAN module.

After reset: 0000H R/W Address: FF98H, FF99H

(a) Read

	15	14	13	12	11	10	9	8
C0INTS	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	CINTS5	CINTS4	CINTS3	CINTS2	CINTS1	CINTS0

(b) Write

	15	14	13	12	11	10	9	8
C0INTS	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	Clear CINTS5	Clear CINTS4	Clear CINTS3	Clear CINTS2	Clear CINTS1	Clear CINTS0

(a) Read

CINTS5-CINTS0	CAN Interrupt Status Bit
0	No related interrupt source event is pending.
1	A related interrupt source event is pending.

Interrupt Status Bit	Related Interrupt Source Event
CINTS5	Wakeup interrupt from CAN sleep mode[Note]
CINTS4	Arbitration loss interrupt
CINTS3	CAN protocol error interrupt
CINTS2	CAN error status interrupt
CINTS1	Interrupt on completion of reception of valid message frame to message buffer m
CINTS0	Interrupt on normal completion of transmission of message frame from message buffer m

Note The CINTS5 bit is set only when the CAN module is woken up from the CAN sleep mode by a CAN bus operation. The CINTS5 bit is not set when the CAN sleep mode has been released by software.

(b) Write

Clear CINTS5-CINTS0	Setting of CINTS5 to CINTS0 Bits
0	CINTS5 to CINTS0 bits are not changed.
1	CINTS5 to CINTS0 bits are cleared to 0.

Caution Please clear the status bit of this register with software when the confirmation of each status is necessary in the interrupt processing, because these bits are not cleared automatically.

(12) CAN module bit rate prescaler register (C0BRP)

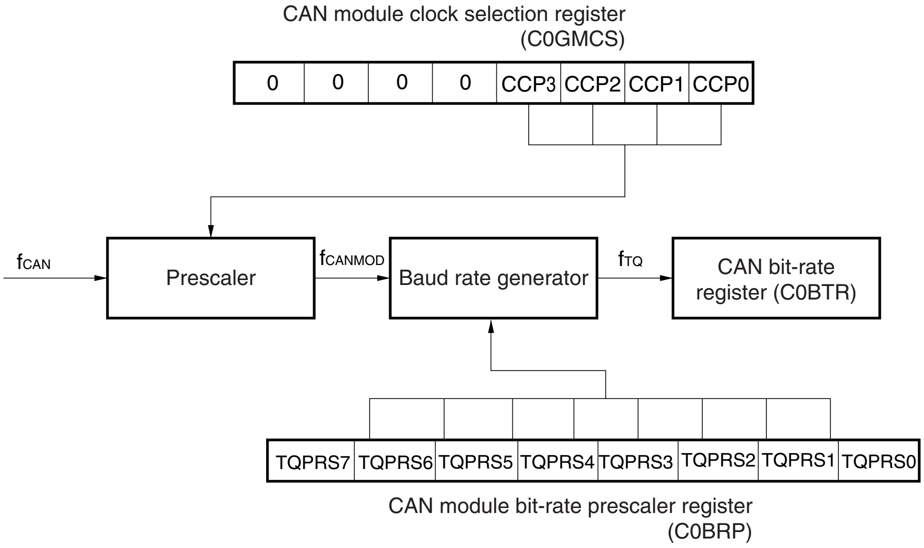
The C0BRP register is used to select the CAN protocol layer basic clock (f_{TQ}). The communication baud rate is set to the C0BTR register.

After reset: FFH R/W Address: FF9EH

	7	6	5	4	3	2	1	0
C0BRP	TQPRS7	TQPRS6	TQPRS5	TQPRS4	TQPRS3	TQPRS2	TQPRS1	TQPRS0

TQPRS7-TQPRS0	CAN Protocol Layer Basic System Clock (f_{TQ})
0	f_{CANMOD}/1
1	f_{CANMOD}/2
:	:
n	f_{CANMOD}/(n+1)
:	:
255	f_{CANMOD}/256 (default value)

Figure 15-25. CAN Module Clock



Caution The C0BRP register can be write-accessed only in the initialization mode.

Remark f_{CAN}: Clock supplied to CAN (f_{PRS})

f_{CANMOD}: CAN module system clock

f_{TQ}: CAN protocol layer basic system clock

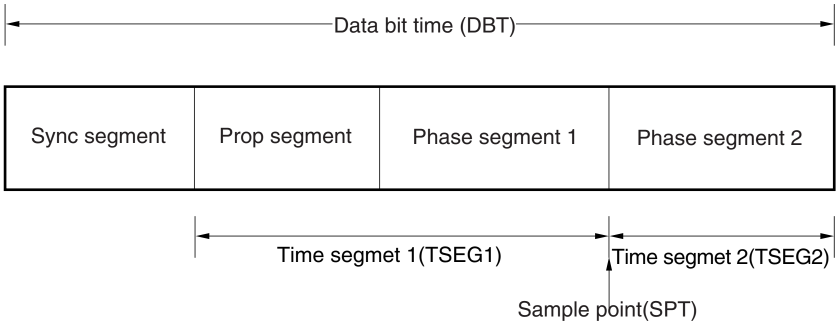
(13) CAN module bit rate register (C0BTR)

The C0BTR register is used to control the data bit time of the communication baud rate.

After reset: 370FH R/W Address: FF9CH, FF9DH

	15	14	13	12	11	10	9	8
C0BTR	0	0	SJW1	SJW0	0	TSEG22	TSEG21	TSEG20
	7	6	5	4	3	2	1	0
	0	0	0	0	TSEG13	TSEG12	TSEG11	TSEG10

Figure 15-26. Data Bit Time



SJW1	SJW0	Length of Synchronization jump width
0	0	1TQ
0	1	2TQ
1	0	3TQ
1	1	4TQ (default value)

TSEG22	TSEG21	TSEG20	Length of time segment 2
0	0	0	1TQ
0	0	1	2TQ
0	1	0	3TQ
0	1	1	4TQ
1	0	0	5TQ
1	0	1	6TQ
1	1	0	7TQ
1	1	1	8TQ (default value)

TSEG13	TSEG12	TSEG11	TSEG10	Length of time segment 1
0	0	0	0	Setting prohibited
0	0	0	1	2TQ[Note]
0	0	1	0	3TQ[Note]
0	0	1	1	4TQ
0	1	0	0	5TQ
0	1	0	1	6TQ
0	1	1	0	7TQ
0	1	1	1	8TQ
1	0	0	0	9TQ
1	0	0	1	10TQ
1	0	1	0	11TQ
1	0	1	1	12TQ
1	1	0	0	13TQ
1	1	0	1	14TQ
1	1	1	0	15TQ
1	1	1	1	16TQ (default value)

Note This setting must not be made when the C0BRP register = 00H.

Remark TQ = 1/f_{TQ} (f_{TQ}: CAN protocol layer basic system clock)

(14) CAN module last in-pointer register (C0LIPT)

The C0LIPT register indicates the number of the message buffer in which a data frame or a remote frame was last stored.

After reset: Undefined R Address: FF9FH

	7	6	5	4	3	2	1	0
C0LIPT	LIPT7	LIPT6	LIPT5	LIPT4	LIPT3	LIPT2	LIPT1	LIPT0

LIPT7-LIPT0	Last In-Pointer Register (C0LIPT)
0 to 15	When the C0LIPT register is read, the contents of the element indexed by the last in-pointer (LIPT) of the receive history list are read. These contents indicate the number of the message buffer in which a data frame or a remote frame was last stored.

Remark The read value of the C0LIPT register is undefined if a data frame or a remote frame has never been stored in the message buffer. If the RHPM bit of the C0RGPT register is set to 1 after the CAN module has changed from the initialization mode to an operation mode, therefore, the read value of the C0LIPT register is undefined.

(15) CAN module receive history list register (C0RGPT)

The C0RGPT register is used to read the receive history list.

After reset: xx02H R/W Address: FF44H, FF45H

(a) Read

C0RGPT

15	14	13	12	11	10	9	8
RGPT7	RGPT6	RGPT5	RGPT4	RGPT3	RGPT2	RGPT1	RGPT0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	RHPM	ROVF

(b) Write

C0RGPT

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	Clear ROVF

(a) Read

RGPT7-RGPT0	Receive History List Get Pointer
0 to 15	When the C0RGPT register is read, the contents of the element indexed by the receive history list get pointer (RGPT) of the receive history list are read. These contents indicate the number of the message buffer in which a data frame or a remote frame has been stored.

RHPM[Note]	Receive History List Pointer Match
0	The receive history list has at least one message buffer number that has not been read.
1	The receive history list has no message buffer numbers that has not been read.

Note The read value of RGPT0 to RGPT7 is invalid when RHPM = 1.

ROVF	Receive History List Overflow Bit
0	All the message buffer numbers that have not been read are preserved. All the numbers of the message buffer in which a new data frame or remote frame has been received and stored are recorded to the receive history list (the receive history list has a vacant element).
1	At least 23 entries have been stored since the host processor has serviced the RHL last time (i.e. read C0RGPT). The first 22 entries are sequentially stored while the last entry can have been overwritten whenever newly received message is stored because all buffer numbers are stored at position LIPT-1 when ROVF bit is set. Thus the sequence of receptions can not be recovered completely now.

Note If ROVF is set, RHPM is no longer cleared on message storage, but RHPM is still set, if all entries of C0RGPT are read by software.

(b) Write

Clear ROVF	Setting of ROVF Bit
0	ROVF bit is not changed.
1	ROVF bit is cleared to 0.

(16) CAN module last out-pointer register (C0LOPT)

The C0LOPT register indicates the number of the message buffer to which a data frame or a remote frame was transmitted last.

After reset: Undefined R Address: FF42H

	7	6	5	4	3	2	1	0
C0LOPT	LOPT7	LOPT6	LOPT5	LOPT4	LOPT3	LOPT2	LOPT1	LOPT0

LOPT7-LOPT0	Last Out-Pointer of Transmit History List (LOPT)
0 to 15	When the C0LOPT register is read, the contents of the element indexed by the last out-pointer (LOPT) of the receive history list are read. These contents indicate the number of the message buffer to which a data frame or a remote frame was transmitted last.

Remark The value read from the C0LOPT register is undefined if a data frame or remote frame has never been transmitted from a message buffer. If the THPM bit is set to 1 after the CAN module has changed from the initialization mode to an operation mode, therefore, the read value of the C0LOPT register is undefined.

(17) CAN module transmit history list register (C0TGPT)

The C0TGPT register is used to read the transmit history list.

After reset: xx02H R/W Address: FF4AH, FF4BH

(a) Read

	15	14	13	12	11	10	9	8
C0TGPT	TGPT7	TGPT6	TGPT5	TGPT4	TGPT3	TGPT2	TGPT1	TGPT0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	THPM	TOVF

(b) Write

	15	14	13	12	11	10	9	8
C0TGPT	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	Clear TOVF

(a) Read

TGPT7-TGPT0	Transmit History List Read Pointer
0 to 15	When the C0TGPT register is read, the contents of the element indexed by the read pointer (TGPT) of the transmit history list are read. These contents indicate the number of the message buffer to which a data frame or a remote frame was transmitted last.

THPM [Note]	Transmit History Pointer Match
0	The transmit history list has at least one message buffer number that has not been read.
1	The transmit history list has no message buffer number that has not been read.

Note The read value of TGPT0 to TGPT7 is invalid when THPM = 1.

TOVF	Transmit History List Overflow Bit
0	All the message buffer numbers that have not been read are preserved. All the numbers of the message buffers to which a new data frame or remote frame has been transmitted are recorded to the transmit history list (the transmit history list has a vacant element).
1	At least 7 entries have been stored since the host processor has serviced the THL last time (i.e. read C0TGPT). The first 6 entries are sequentially stored while the last entry can have been overwritten whenever a message is newly transmitted because all buffer numbers are stored at position LOPT-1 when TOVF bit is set. Thus the sequence of transmissions can not be recovered completely now.

Note If TOVF is set, THPM is no longer cleared on message transmission, but THPM is still set, if all entries of C0TGPT are read by software.

Remark Transmission from message buffer 0 to 7 is not recorded to the transmit history list in the normal operation mode with ABT.

(b) Write

Clear TOVF	Setting of TOVF Bit
0	TOVF bit is not changed.
1	TOVF bit is cleared to 0.

(18) CAN module time stamp register (C0TS)

The C0TS register is used to control the time stamp function.

After reset: 0000H R/W Address: FF8AH, FF8BH

(a) Read

	15	14	13	12	11	10	9	8
C0TS	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	TSLOCK	TSSEL	TSEN

(b) Write

	15	14	13	12	11	10	9	8
C0TS	0	0	0	0	0	Set TSLOCK	Set TSSEL	Set TSEN
	7	6	5	4	3	2	1	0
	0	0	0	0	0	Clear TSLOCK	Clear TSSEL	Clear TSEN

Remark The lock function of the time stamp function must not be used when the CAN module is in the normal operation mode with ABT.

(a) Read

TSLOCK	Time Stamp Lock Function Enable Bit
0	Time stamp lock function stopped. The TSOUT signal is toggled each time the selected time stamp capture event occurs.
1	Time stamp lock function enabled. The TSOUT signal is toggled each time the selected time stamp capture event occurs. However, the TSOUT output signal is locked when a data frame has been correctly received to message buffer 0[Note].

Note The TSEN bit is automatically cleared to 0.

TSSEL	Time Stamp Capture Event Selection Bit
0	The time stamp capture event is SOF.
1	The time stamp capture event is the last bit of EOF.

TSEN	TSOUT Signal Operation Setting Bit
0	Disable TSOUT signal toggle operation.
1	Enable TSOUT signal toggle operation.

Remark The signal TSOUT is output from the CAN macro to a timer resource, depending on implementation. Refer to **Figure 12-19 Format of Input Switch Control Register (ISC)**.

(b) Write

Set TSLOCK	Clear TSLOCK	Setting of TSLOCK Bit
0	1	TSLOCK bit is cleared to 0.
1	0	TSLOCK bit is set to 1.
Other than above		TSLOCK bit is not changed.

Set TSSEL	Clear TSSEL	Setting of TSSEL Bit
0	1	TSSEL bit is cleared to 0.
1	0	TSSEL bit is set to 1.
Other than above		TSSEL bit is not changed.

Set TSEN	Clear TSEN	Setting of TSEN Bit
0	1	TSEN bit is cleared to 0.
1	0	TSEN bit is set to 1.
Other than above		TSEN bit is not changed.

(19) CAN message data byte register (C0MDATAxm)(x = 0 to 7), (C0MDATAzm) (z = 01, 23, 45, 67)

The C0MDATAxm, C0MDATAzm registers are used to store the data of a transmit/receive message. The C0MDATAxm registers can access in 8-bit units. The C0MDATAzm registers can access the C0MDATAxm registers in 16-bit units.

After reset: Undefined R/W Address: See **Table 15-16 Register Access Types**.

- C0MDATAxm Register

	7	6	5	4	3	2	1	0
C0MDATA0m	MDATA07	MDATA06	MDATA05	MDATA04	MDATA03	MDATA02	MDATA01	MDATA00
C0MDATA1m	MDATA17	MDATA16	MDATA15	MDATA14	MDATA13	MDATA12	MDATA11	MDATA10
C0MDATA2m	MDATA27	MDATA26	MDATA25	MDATA24	MDATA23	MDATA22	MDATA21	MDATA20
C0MDATA3m	MDATA37	MDATA36	MDATA35	MDATA34	MDATA33	MDATA32	MDATA31	MDATA30
C0MDATA4m	MDATA47	MDATA46	MDATA45	MDATA44	MDATA43	MDATA42	MDATA41	MDATA40
C0MDATA5m	MDATA57	MDATA56	MDATA55	MDATA54	MDATA53	MDATA52	MDATA51	MDATA50
C0MDATA6m	MDATA67	MDATA66	MDATA65	MDATA64	MDATA63	MDATA62	MDATA61	MDATA60
C0MDATA7m	MDATA77	MDATA76	MDATA75	MDATA74	MDATA73	MDATA72	MDATA71	MDATA70

- C0MDATAzm Register

	15	14	13	12	11	10	9	8
C0MDATA01m	MDATA0115	MDATA0114	MDATA0113	MDATA0112	MDATA0111	MDATA0110	MDATA019	MDATA018
	7	6	5	4	3	2	1	0
	MDATA017	MDATA016	MDATA015	MDATA014	MDATA013	MDATA012	MDATA011	MDATA010

	15	14	13	12	11	10	9	8
C0MDATA23m	MDATA2315	MDATA2314	MDATA2313	MDATA2312	MDATA2311	MDATA2310	MDATA239	MDATA238
	7	6	5	4	3	2	1	0
	MDATA237	MDATA236	MDATA235	MDATA234	MDATA233	MDATA232	MDATA231	MDATA230

	15	14	13	12	11	10	9	8
C0MDATA45m	MDATA4515	MDATA4514	MDATA4513	MDATA4512	MDATA4511	MDATA4510	MDATA459	MDATA458
	7	6	5	4	3	2	1	0
	MDATA457	MDATA456	MDATA455	MDATA454	MDATA453	MDATA452	MDATA451	MDATA450

	15	14	13	12	11	10	9	8
C0MDATA67m	MDATA6715	MDATA6714	MDATA6713	MDATA6712	MDATA6711	MDATA6710	MDATA679	MDATA678
	7	6	5	4	3	2	1	0
	MDATA677	MDATA676	MDATA675	MDATA674	MDATA673	MDATA672	MDATA671	MDATA670

(20) CAN message data length register m (C0MDLCm)

The C0MDLCm register is used to set the number of bytes of the data field of a message buffer.

After reset: 0000xxxxB R/W Address: See **Table 15-16 Register Access Types**.

	7	6	5	4	3	2	1	0
C0MDLCm	0	0	0	0	MDLC3	MDLC2	MDLC1	MDLC0

MDLC3	MDLC2	MDLC1	MDLC0	Data Length Of Transmit/Receive Message
0	0	0	0	0 bytes
0	0	0	1	1 byte
0	0	1	0	2 bytes
0	0	1	1	3 bytes
0	1	0	0	4 bytes
0	1	0	1	5 bytes
0	1	1	0	6 bytes
0	1	1	1	7 bytes
1	0	0	0	8 bytes
1	0	0	1	Setting prohibited (If these bits are set during transmission, 8-byte data is transmitted regardless of the set DLC value when a data frame is transmitted. However, the DLC actually transmitted to the CAN bus is the DLC value set to this register.)^{Note}
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Note The data and DLC value actually transmitted to CAN bus are as follows.

Type of Transmit Frame	Length of Transmit Data	DLC Transmitted
Data frame	Number of bytes specified by DLC (However, 8 bytes if DLC ≥ 8)	MDLC[3:0]
Remote frame	0 bytes	

Cautions 1. Be sure to set bits 7 to 4 0000B.

2. Receive data is stored in as many C0MDATAxm as the number of bytes (however, the upper limit is 8) corresponding to DLC of the received frame. C0MDATAxm in which no data is stored is undefined.

(21) CAN message configuration register (C0MCONFm)

The C0MCONFm register is used to specify the type of the message buffer and to set a mask.

After reset: Undefined R/W Address: See **Table 15-16 Register Access Types**.

	7	6	5	4	3	2	1	0
C0MCONFm	OWS	RTR	MT2	MT1	MT0	0	0	MA0

OWS	Overwrite Control Bit
0	The message buffer that has already received a data frame[Note] is not overwritten by a newly received data frame. The newly received data frame is discarded.
1	The message buffer that has already received a data frame[Note] is overwritten by a newly received data frame.

Note The “message buffer that has already received a data frame” is a receive message buffer whose DN bit has been set to 1.

Remark A remote frame is received and stored, regardless of the setting of OWS bit and DN bit. A remote frame that satisfies the other conditions (ID matches, RTR = 0, TRQ = 0) is always received and stored in the corresponding message buffer (interrupt generated, DN flag set, MDLC [3:0] bits updated, and recorded to the receive history list).

RTR	Remote Frame Request Bit[Note]
0	Transmit a data frame.
1	Transmit a remote frame.

Note The RTR bit specifies the type of message frame that is transmitted from a message buffer defined as a transmit message buffer. Even if a valid remote frame has been received, RTR of the transmit message buffer that has received the frame remains cleared to 0. Even if a remote frame whose ID matches has been received from the CAN bus with the RTR bit of the transmit message buffer set to 1 to transmit a remote frame, that remote frame is not received or stored (interrupt generated, DN flag set, MDLC [3:0] bits updated, and recorded to the receive history list).

MT2	MT1	MT0	Message Buffer Type Setting Bit
0	0	0	Transmit message buffer
0	0	1	Receive message buffer (no mask setting)
0	1	0	Receive message buffer (mask 1 set)
0	1	1	Receive message buffer (mask 2 set)
1	0	0	Receive message buffer (mask 3 set)
1	0	1	Receive message buffer (mask 4 set)
Other than above			Setting prohibited

MA0	Message Buffer Assignment Bit
0	Message buffer not used.
1	Message buffer used.

Caution Be sure to write 0 to bits 2 and 1.

(22) CAN message id register m (C0MIDLm, C0MIDHm)

The C0MIDLm and C0MIDHm registers are used to set an identifier (ID).

After reset: Undefined R/W Address: See **Table 15-16 Register Access Types**.

	15	14	13	12	11	10	9	8
C0MIDLm	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
	7	6	5	4	3	2	1	0
	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0

	15	14	13	12	11	10	9	8
C0MIDHm	IDE	0	0	ID28	ID27	ID26	ID25	ID24
	7	6	5	4	3	2	1	0
	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16

IDE	Format Mode Specification Bit
0	Standard format mode (ID28 to ID18: 11 bits)[Note]
1	Extended format mode (ID28 to ID0: 29 bits)

Note The ID17 to ID0 bits are not used.

ID28 to ID0	Message ID
ID28 to ID18	Standard ID value of 11 bits (when IDE = 0)
ID28 to ID0	Extended ID value of 29 bits (when IDE = 1)

Cautions 1. Be sure to write 0 to bits 14 and 13 of the C0MIDHm register.

2. Be sure to align the ID value according to the given bit positions into this registers. Note that for standard ID, the ID value must be shifted to fit into ID28 to ID11 bit positions.

(23) CAN message control register m (C0MCTRLm)

The C0MCTRLm register is used to control the operation of the message buffer.

After reset: 00x000000
00000000B   R/W   Address: See **Table 15-16 Register Access Types**.

(a) Read

	15	14	13	12	11	10	9	8
C0MCTRLm	0	0	MUC	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	MOW	IE	DN	TRQ	RDY

(b) Write

	15	14	13	12	11	10	9	8
C0MCTRLm	0	0	0	0	Set IE	0	Set TRQ	Set RDY
	7	6	5	4	3	2	1	0
	0	0	0	Clear MOW	Clear IE	Clear DN	Clear TRQ	Clear RDY

(a) Read

MUC[Note]	Message Buffer Data Updating Bit
0	The CAN module is not updating the message buffer (reception and storage).
1	The CAN module is updating the message buffer (reception and storage).

Note The MUC bit is undefined until the first reception and storage is performed.

MOW	Message Buffer Overwrite Status Bit
0	The message buffer is not overwritten by a newly received data frame.
1	The message buffer is overwritten by a newly received data frame.

Remark MOW bit is not set to 1 even if a remote frame is received and stored in the transmit message buffer with DN = 1.

IE	Message Buffer Interrupt Request Enable Bit
0	Receive message buffer: Valid message reception completion interrupt disabled. Transmit message buffer: Normal message transmission completion interrupt disabled.
1	Receive message buffer: Valid message reception completion interrupt enabled. Transmit message buffer: Normal message transmission completion interrupt enabled.

DN	Message Buffer Data Updating Bit
0	A data frame or remote frame is not stored in the message buffer.
1	A data frame or remote frame is stored in the message buffer.

TRQ	Message Buffer Transmission Request Bit
0	No message frame transmitting request that is pending or being transmitted is in the message buffer.
1	The message buffer is holding transmission of a message frame pending or is transmitting a message frame.

Caution Do not set the TRQ bit and the RDY bit (1) at the same time. Set the RDY bit (1) before setting the TRQ bit.

RDY	Message Buffer Ready Bit
0	The message buffer can be written by software. The CAN module cannot write to the message buffer.
1	Writing the message buffer by software is ignored (except a write access to the RDY, TRQ, DN, and MOW bits). The CAN module can write to the message buffer.

Cautions1. Do not clear the RDY bit (0) during message transmission.
Follow the transmission abort process about clearing the RDY bit (0) for redefinition of the message buffer.

2. Clear again when RDY bit is not cleared even if this bit is cleared.

3. Be sure that RDY is cleared before writing to the message buffer registers. Perform this confirmation by reading back the RDY bit. However, setting the TRQ bit, clearing the DN bit, setting the RDY bit or clearing the MOW bit of the C0MCTRLm register need not be confirmed.

(b) Write

Clear MOW	Setting of MOW Bit
0	MOW bit is not changed.
1	MOW bit is cleared to 0.

Set IE	Clear IE	Setting of IE Bit
0	1	IE bit is cleared to 0.
1	0	IE bit is set to 1.
Other than above		IE bit is not changed.

Caution Set IE bit and RDY bit always separately.

Clear DN	Setting of DN Bit
0	DN bit is not changed.
1	DN bit is cleared to 0.

Caution Do not set the DN bit to 1 by software. Be sure to write 0 to bit 10.

Set TRQ	Clear TRQ	Setting of TRQ Bit
0	1	TRQ bit is cleared to 0.
1	0	TRQ bit is set to 1.
Other than above		TRQ bit is not changed.

Caution While receiving a message from another node or transmitting the messages, there is a possibility of not to begin immediately the transmission even if the TRQ bit is set to 1.

The transmission is not aborted even if the TRQ bit is cleared to 0. The transmission is continued if a message is currently being transmitted and until the transmission is completed (successfully or not).

Set RDY	Clear RDY	Setting of RDY Bit
0	1	RDY bit is cleared to 0.
1	0	RDY bit is set to 1.
Other than above		RDY bit is not changed.

Caution Set IE bit and RDY bit always separately.

15.8 CAN Controller Initialization

15.8.1 Initialization of CAN module

Before the CAN module operation is enabled, the CAN module system clock needs to be determined by setting the CCP[3:0] bits of the C0GMCS register by software. Do not change the setting of the CAN module system clock after CAN module operation is enabled.

The CAN module is enabled by setting the GOM bit of the C0GMCTRL register.

For the procedure of initializing the CAN module, refer to **15.16 Operation Of CAN Controller.**

15.8.2 Initialization of message buffer

After the CAN module is enabled, the message buffers contain undefined values. A minimum initialization for all the message buffers, even for those not used in the application, is necessary before switching the CAN module from the initialization mode to one of the operation modes.

- Clear the RDY, TRQ, and DN bits of the C0MCTRLm register to 0.
- Clear the MA0 bit of the C0MCONFm register to 0.

Remark m = 0 to 15

15.8.3 Redefinition of message buffer

Redefining a message buffer means changing the ID and control information of the message buffer while a message is being received or transmitted, without affecting other transmission/reception operations.

(1) To redefine message buffer in initialization mode

Place the CAN module in the initialization mode once and then change the ID and control information of the message buffer in the initialization mode. After changing the ID and control information, set the CAN module in an operation mode.

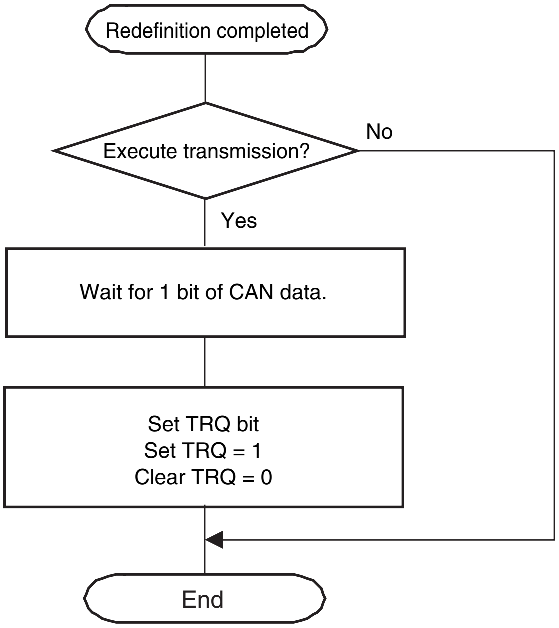
(2) To redefine message buffer during reception

Perform redefinition as shown in **Figure 15-40. Message Buffer Redefinition**.

(3) To redefine message buffer during transmission

To rewrite the contents of a transmit message buffer to which a transmission request has been set, perform transmission abort processing (refer to **15.10.4 (1) Transmission abort process except for in normal operation mode with automatic block transmission (ABT)** and **15.10.4 (2) Transmission abort process except for ABT transmission in normal operation mode with automatic block transmission (ABT)**. Confirm that transmission has been aborted or completed, and then redefine the message buffer. After redefining the transmit message buffer, set a transmission request using the procedure described below. When setting a transmission request to a message buffer that has been redefined without aborting the transmission in progress, however, the 1-bit wait time is not necessary.

Figure 15-27. Setting Transmission Request (TRQ) to Transmit Message Buffer After Redefining



Cautions 1. When a message is received, reception filtering is performed in accordance with the ID and mask set to each receive message buffer. If the procedure in Figure 15-40. Message Buffer Redefinition is not observed, the contents of the message buffer after it has been redefined may contradict the result of reception (result of reception filtering). If this happens, check that the ID and IDE received first and stored in the message buffer following redefinition are those stored after the message buffer has been redefined. If no ID and IDE are stored after redefinition, redefine the message buffer again.

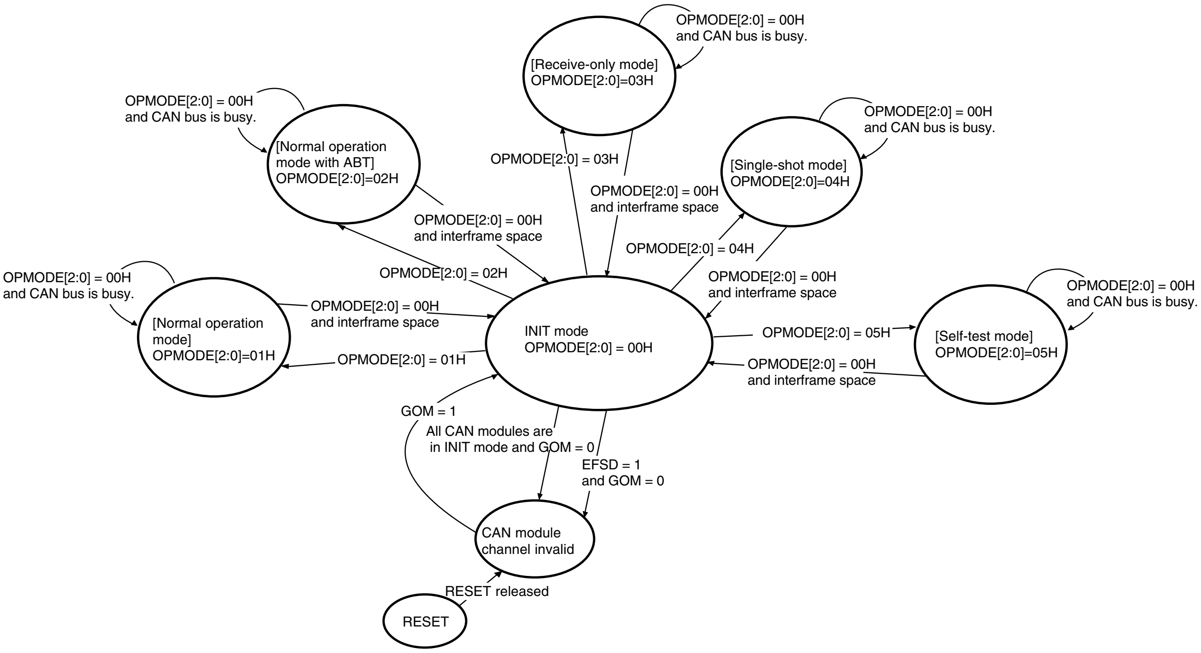
2. When a message is transmitted, the transmission priority is checked in accordance with the ID, IDE, and RTR bits set to each transmit message buffer to which a transmission request was set. The transmit message buffer having the highest priority is selected for transmission. If the procedure in Figure 15-41. Message Buffer Redefinition during Transmission is not observed, a message with an ID not having the highest priority may be transmitted after redefinition.

15.8.4 Transition from initialization mode to operation mode

The CAN module can be switched to the following operation modes.

- Normal operation mode
- Normal operation mode with ABT
- Receive-only mode
- Single-shot mode
- Self-test mode

Figure 15-28. Transition to Operation Modes



The transition from the initialization mode to an operation mode is controlled by the bit string OPMODE [2:0] in the C0CTRL register.

Changing from one operation mode into another requires shifting to the initialization mode in between. Do not change one operation mode to another directly; otherwise the operation will not be guaranteed.

Requests for transition from the operation mode to the initialization mode are held pending when the CAN bus is not in the interframe space (i.e., frame reception or transmission is in progress), and the CAN module enters the initialization mode at the first bit in the interframe space (the value of OPMODE [2:0] are changed to 00H). After issuing a request to change the mode to the initialization mode, read the OPMODE [2:0] bits until their value becomes 000B to confirm that the module has entered the initialization mode (refer to **Figure 15-37. Initialization**).

15.8.5 Resetting error counter C0ERC of CAN module

If it is necessary to reset the CAN module error counter register C0ERC and the CAN module information register C0INFO when re-initialization or forced recovery from the bus-off state is made, set the CCERC bit of the C0CTRL register to 1 in the initialization mode. When this bit is set to 1, the CAN module error counter register C0ERC and the CAN module information register C0INFO are cleared to their default values.

15.9 Message Reception

15.9.1 Message reception

In all the operation modes, the complete message buffer area is analyzed to find a suitable buffer to store a newly received message. All message buffers satisfying the following conditions are included in that evaluation (RX-search process).

- Used as a message buffer
  (MA0 bit of C0MCONFm register set to 1B.)
- Set as a receive message buffer
  (MT [2:0] bits of C0MCONFm register set to 001B, 010B, 011B, 100B, or 101B.)
- Ready for reception
  (RDY bit of C0MCTRLm register set to 1.)

When two or more message buffers of the CAN module receive a message, the message is stored according to the priority explained below. The message is always stored in the message buffer with the highest priority, not in a message buffer with a low priority. For example, when an unmasked receive message buffer and a receive message buffer linked to mask 1 have the same ID, the received message is not stored in the message buffer linked to mask 1, even if that message buffer has not received a message and a message has already been received in the unmasked receive message buffer. In other words, when a condition has been set to store a message in two or more message buffers with different priorities, the message buffer with the highest priority always stores the message; the message is not stored in message buffers with a lower priority. This also applies when the message buffer with the highest priority is unable to receive and store a message (i.e., when DN = 1 indicating that a message has already been received, but rewriting is disabled because OWS = 0). In this case, the message is not actually received and stored in the candidate message buffer with the highest priority, but neither is it stored in a message buffer with a lower priority.

Priority	Storing Condition If Same ID is Set	
1 (high)	Unmasked message buffer	DN = 0
		DN = 1 and OWS = 1
2	Message buffer linked to mask 1	DN = 0
		DN = 1 and OWS = 1
3	Message buffer linked to mask 2	DN = 0
		DN = 1 and OWS = 1
4	Message buffer linked to mask 3	DN = 0
		DN = 1 and OWS = 1
5(low)	Message buffer linked to mask 4	DN = 0
		DN = 1 and OWS = 1

Remark m = 0 to 15

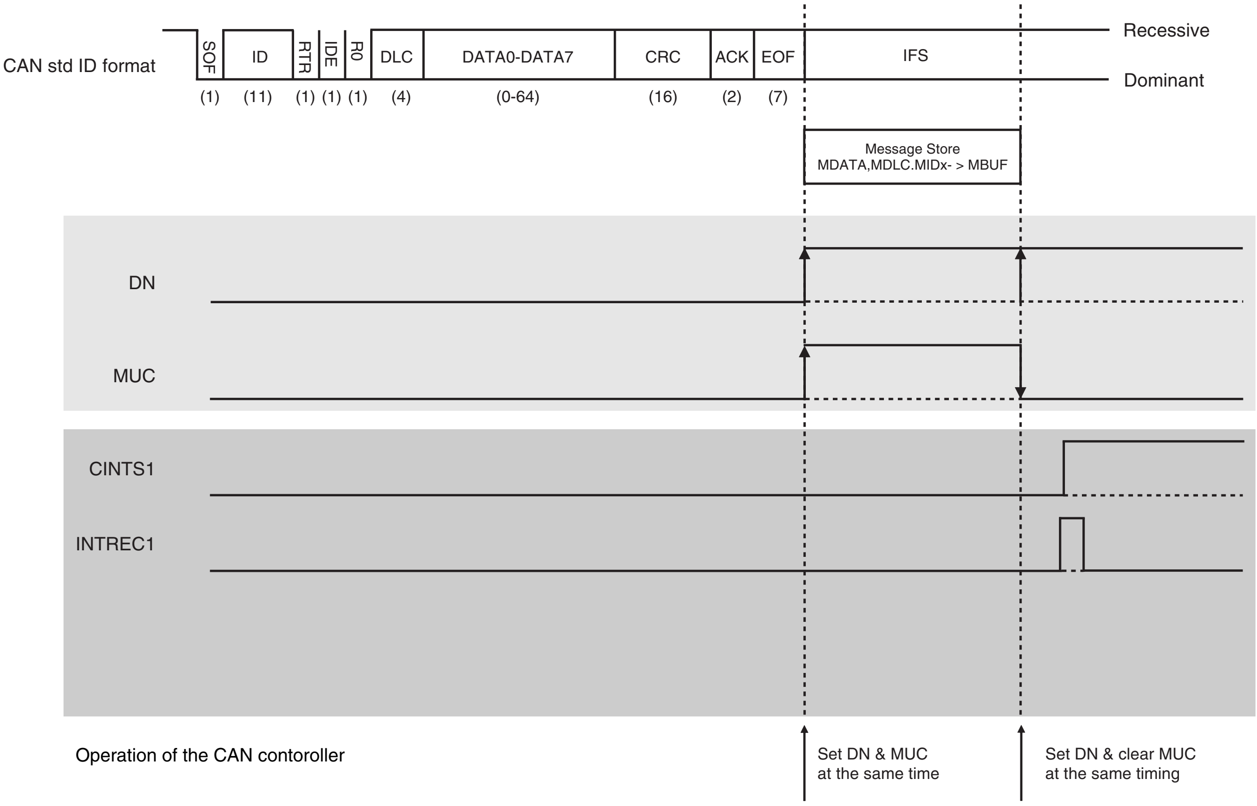
15.9.2 Receive Data Read

To keep data consistency when reading CAN message buffers, perform the data reading according to **Figures 15-51** to **15-53**.

During message reception, the CAN module sets DN of the C0MCTRLm register two times: at the beginning of the storage process of data to the message buffer, and again at the end of this storage process. During this storage process, the MUC bit of the C0MCTRLm register of the message buffer is set (Refer to **Figure 15-29**).

The receive history list is also updated just before the storage process. In addition, during storage process (MUC = 1), the RDY bit of the C0MCTRL register of the message buffer is locked to avoid the coincidental data WR by CPU. Note the storage process may be disturbed (delayed) when the CPU accesses the message buffer.

Figure 15-29. DN and MUC Bit Setting Period (for Standard ID Format)



Remark m = 0 to 15

15.9.3 Receive history list function

The receive history list (RHL) function records in the receive history list the number of the receive message buffer in which each data frame or remote frame was received and stored. The RHL consists of storage elements equivalent to up to 23 messages, the last in-message pointer (LIPT) with the corresponding C0LIPT register and the receive history list get pointer (RGPT) with the corresponding C0RGPT register.

The RHL is undefined immediately after the transition of the CAN module from the initialization mode to one of the operation modes.

The C0LIPT register holds the contents of the RHL element indicated by the value of the LIPT pointer minus 1. By reading the C0LIPT register, therefore, the number of the message buffer that received and stored a data frame or remote frame first can be checked. The LIPT pointer is utilized as a write pointer that indicates to what part of the RHL a message buffer number is recorded. Any time a data frame or remote frame is received and stored, the corresponding message buffer number is recorded to the RHL element indicated by the LIPT pointer. Each time recording to the RHL has been completed, the LIPT pointer is automatically incremented. In this way, the number of the message buffer that has received and stored a frame will be recorded chronologically.

The RGPT pointer is utilized as a read pointer that reads a recorded message buffer number from the RHL. This pointer indicates the first RHL element that the CPU has not read yet. By reading the C0RGPT register by software, the number of a message buffer that has received and stored a data frame or remote frame can be read. Each time a message buffer number is read from the C0RGPT register, the RGPT pointer is automatically incremented.

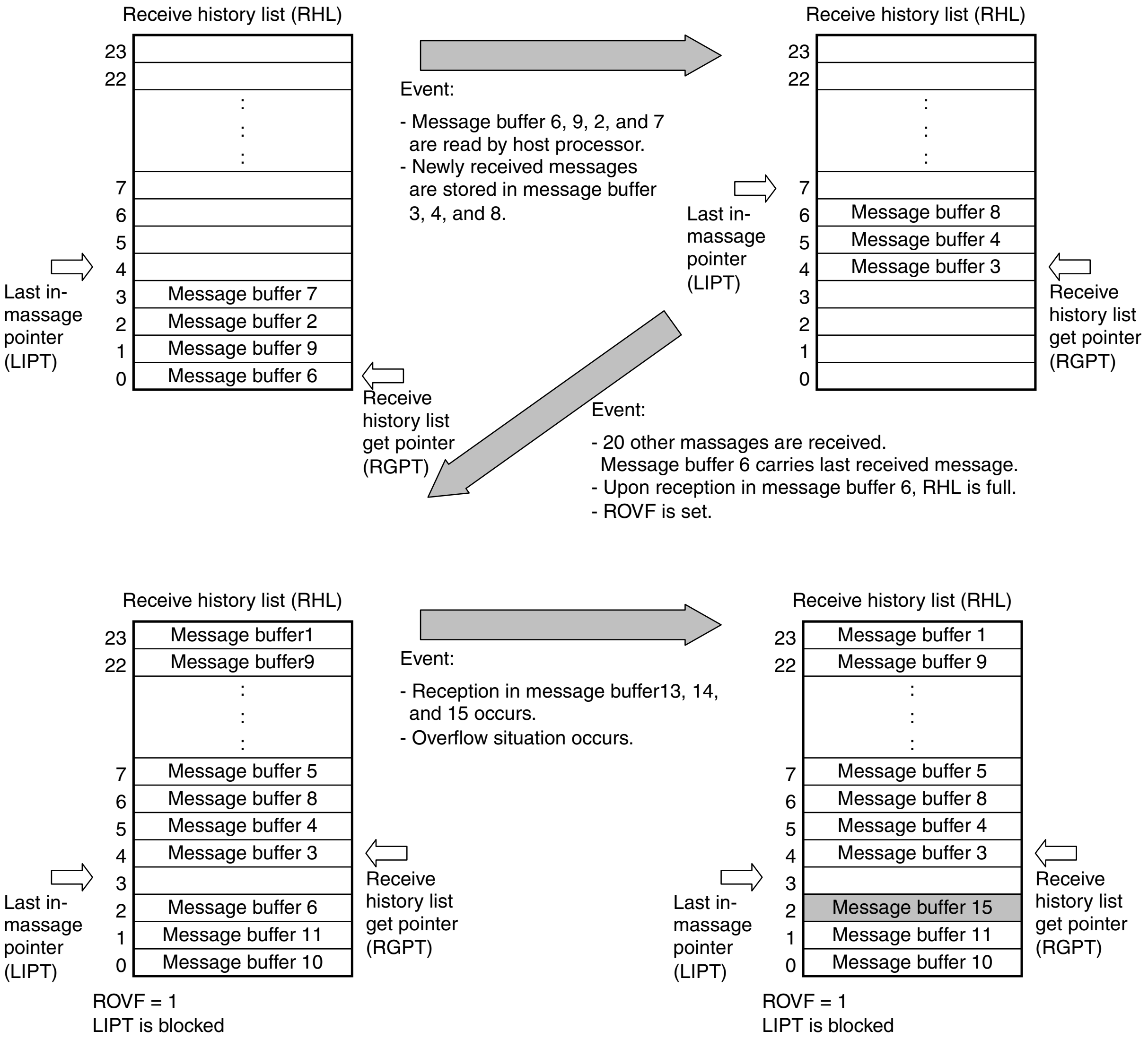
If the value of the RGPT pointer matches the value of the LIPT pointer, the RHPM bit (receive history list pointer match) of the C0RGPT register is set to 1. This indicates that no message buffer number that has not been read remains in the RHL. If a new message buffer number is recorded, the LIPT pointer is incremented and because its value no longer matches the value of the RGPT pointer, the RHPM bit is cleared. In other words, the numbers of the unread message buffers exist in the RHL.

If the LIPT pointer is incremented and matches the value of the RGPT pointer minus 1, the ROVF bit (receive history list overflow) of the C0RGPT register is set to 1. This indicates that the RHL is full of numbers of message buffers that have not been read. When further message reception and storing occur, the last recorded message buffer number is overwritten by the number of the message buffer that received and stored the new message. In this case, after the ROVF bit has been set (1), the recorded message buffer numbers in the RHL do not completely reflect the chronological order. However messages itself are not lost and can be located by CPU search in message buffer memory with the help of the DN bit.

Caution If the history list is in the overflow condition (ROVF is set), reading the history list contents is still possible, until the history list is empty (indicated by RHPM flag set). Nevertheless, the history list remains in the overflow condition, until ROVF is cleared by software. If ROVF is not cleared, the RHPM flag will also not be updated (cleared) upon a message storage of newly received frame. This may lead to the situation, that RHPM indicates an empty history list, although a reception has taken place, while the history list is in the overflow state (ROVF and RHPM are set).

As long as the RHL contains 23 or less entries the sequence of occurrence is maintained. If more receptions occur without reading the RHL by the host processor, complete sequence of receptions can not be recovered.

Figure 15-30. Receive History List



Event:

- Message buffer 6, 9, 2, and 7 are read by host processor.
- Newly received messages are stored in message buffer 3, 4, and 8.

Event:

- 20 other massages are received. Message buffer 6 carries last received message.
- Upon reception in message buffer 6, RHL is full.
- ROVF is set.

Event:

- Reception in message buffer13, 14, and 15 occurs.
- Overflow situation occurs.

ROVF = 1
LIPT is blocked

ROVF = 1
LIPT is blocked

ROVF = 1 denotes that LIPT equals RGPT–1 while message buffer number stored to element indicated by LIPT–1.

15.9.4 Mask function

For any message buffer, which is used for reception, the assignment to one of four global reception masks (or no mask) can be selected.

By using the mask function, the message ID comparison can be reduced by masked bits, herewith allowing the reception of several different IDs into one buffer.

While the mask function is in effect, an identifier bit that is defined to be "1" by a mask in the received message is not compared with the corresponding identifier bit in the message buffer.

However, this comparison is performed for any bit whose value is defined as "0" by the mask.

For example, let us assume that all messages that have a standard-format ID, in which bits ID27 to ID25 are "0" and bits ID24 and ID22 are "1", are to be stored in message buffer 14. The procedure for this example is shown below.

<1> Identifier to be stored in message buffer

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
x	0	0	0	1	x	1	x	x	x	x

x = don’t care

<2> Identifier to be configured in message buffer 14 (example)
(using CANn message ID registers L14 and H14 (C0MIDL14 and C0MIDH14))

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
x	0	0	0	1	x	1	x	x	x	x

ID17	ID16	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
x	x	x	x	x	x	x	x	x	x	x

ID6	ID5	ID4	ID3	ID2	ID1	ID0
x	x	x	x	x	x	x

ID with ID27 to ID25 cleared to "0" and ID24 and ID22 set to "1" is registered (initialized) to message buffer 14.

Remark Message buffer 14 is set as a standard format identifier that is linked to mask 1 (MT [2:0] of C0MCONF14 register are set to 010B).

<3> Mask setting for CAN module 1 (mask 1) (Example)
(Using CAN1 address mask 1 registers L and H (C1MASK1L and C1MASK1H))

CMID28	CMID27	CMID26	CMID25	CMID24	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18
1	0	0	0	0	1	0	1	1	1	1

CMID17	CMID16	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8	CMID7
1	1	1	1	1	1	1	1	1	1	1

CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0
1	1	1	1	1	1	1

1: Not compared (masked)
0: Compared

The CMID27 to CMID24 and CMID22 bits are cleared to "0", and CMID28, CMID23, and CMID21 to CMID0 bits are set to "1".

15.9.5 Multi buffer receive block function

The multi buffer receive block (MBRB) function is used to store a block of data in two or more message buffers sequentially with no CPU interaction, by setting the same ID to two or more message buffers with the same message buffer type.

Suppose, for example, the same message buffer type is set to 5 message buffers, message buffers 10 to 14, and the same ID is set to each message buffer. If the first message whose ID matches the ID of the message buffers is received, it is stored in message buffer 10. At this point, the DN bit of message buffer 10 is set, prohibiting overwriting the message buffer when subsequent messages are received.

If the next message with a matching ID is received, it is received and stored in message buffer 11. Each time a message with a matching ID is received, it is sequentially (in the ascending order) stored in message buffers 12, 13, and 14. Even when a data block consisting of multiple messages is received, the messages can be stored and received without overwriting the previously received matching-ID data.

Whether a data block has been received and stored can be checked by setting the IE bit of the C0MCTRLm register of each message buffer. For example, if a data block consists of k messages, k message buffers are initialized for reception of the data block. The IE bit in message buffers 0 to (k–2) is cleared to 0 (interrupts disabled), and the IE bit in message buffer k–1 is set to 1 (interrupts enabled). In this case, a reception completion interrupt occurs when a message has been received and stored in message buffer k–1, indicating that MBRB has become full. Alternatively, by clearing the IE bit of message buffers 0 to (k–3) and setting the IE bit of message buffer k–2, a warning that MBRB is about to overflow can be issued.

The basic conditions of storing receive data in each message buffer for the MBRB are the same as the conditions of storing data in a single message buffer.

Cautions 1. MBRB can be configured for each of the same message buffer types. Therefore, even if a message buffer of another MBRB whose ID matches but whose message buffer type is different has a vacancy, the received message is not stored in that message buffer, but instead discarded.

2. MBRB does not have a ring buffer structure. Therefore, after a message is stored in the message buffer having the highest number in the MBRB configuration, a newly received message will not be stored in the message buffer having the lowest message buffer number.

3. MBRB operates based on the reception and storage conditions; there are no settings dedicated to MBRB, such as function enable bits. By setting the same message buffer type and ID to two or more message buffers, MBRB is automatically configured.

4. With MBRB, "matching ID" means "matching ID after mask". Even if the ID set to each message buffer is not the same, if the ID that is masked by the mask register matches, it is considered a matching ID and the buffer that has this ID is treated as the storage destination of a message.

5. The priority between MBRBs is mentioned in 15.9.1 Message reception.

Remark m = 0 to 15

15.9.6 Remote frame reception

In all the operation modes, when a remote frame is received, the message buffer that is to store the remote frame is searched from all the message buffers satisfying the following conditions.

- Used as a message buffer
  (MA0 bit of C0MCONFm register set to 1B.)
- Set as a transmit message buffer
  (MT[2:0] bits in C0MCONFm register set to 000B)
- Ready for reception
  (RDY bit of C0MCTRLm register set to 1.)
- Set to transmit message
  (RTR bit of C0MCONFm register is cleared to 0.)
- Transmission request is not set.
  (TRQ bit of C0MCTRLm register is cleared to 0.)

Upon acceptance of a remote frame, the following actions are executed if the ID of the received remote frame matches the ID of a message buffer that satisfies the above conditions.

- The MDLC [3:0] bit string in the C0MDLCm register stores the received DLC value.
- C0MDATA0m to C0MDATA7m in the data area are not updated (data before reception is saved).
- The DN bit of the C0MCTRLm register is set to 1.
- The CINTS1 bit of the C0INTS register is set to 1 (if the IE bit in the C0MCTRLm register of the message buffer that receives and stores the frame is set to 1).
- The reception completion interrupt (INTC0REC) is output (if the IE bit in the C0MCTRLm register of the message buffer that receives and stores the frame is set to 1 and if the CIE1 bit of the C0IE register is set to 1).
- The message buffer number is recorded to the receive history list.

Caution When a message buffer is searched for receiving and storing a remote frame, overwrite control by the OWS bit of the C0MCONFm register of the message buffer and the DN bit of the C0MCTRLm register are not affected. The setting of OWS is ignored, and DN is set in any case. If more than one transmit message buffer has the same ID and the ID of the received remote frame matches that ID, the remote frame is stored in the transmit message buffer with the lowest message buffer number.

Remark m = 0 to 15

15.10 Message Transmission

15.10.1 Message transmission

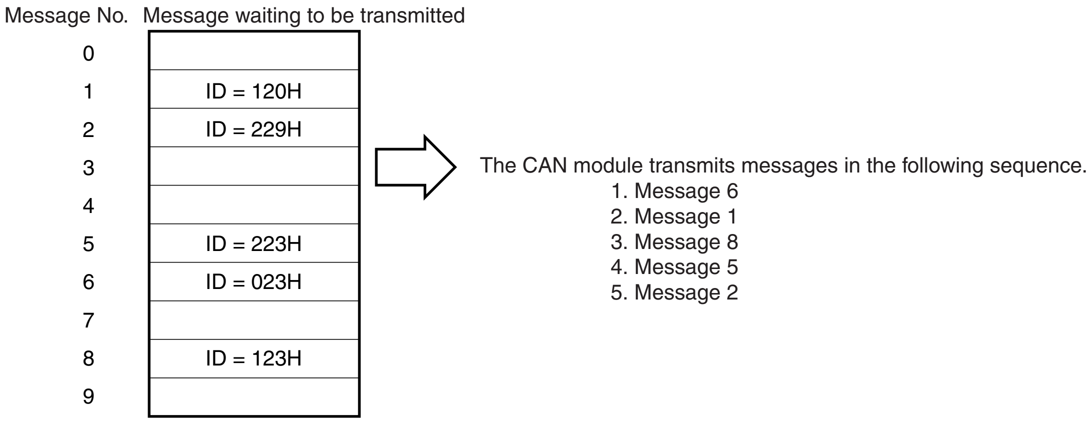
In all the operation modes, if the TRQ bit is set to 1 in a message buffer that satisfies the following conditions, the message buffer that is to transmit a message is searched.

- Used as a message buffer
  (MA0 bit of C0MCONFm register set to 1B.)
- Set as a transmit message buffer
  (MT [2:0] bits of C0MCONFm register set to 000B.)
- Ready for transmission
  (RDY bit of C0MCTRLm register set to 1.)

The CAN system is a multi-master communication system. In a system like this, the priority of message transmission is determined based on message identifiers (IDs). To facilitate transmission processing by software when there are several messages awaiting transmission, the CAN module uses hardware to check the ID of the message with the highest priority and automatically identifies that message. This eliminates the need for software-based priority control.

Transmission priority is controlled by the identifier (ID).

Figure 15-31. Message Processing Example



Message No.	Message waiting to be transmitted
0	
1	ID = 120H
2	ID = 229H
3	
4	
5	ID = 223H
6	ID = 023H
7	
8	ID = 123H
9	

The CAN module transmits messages in the following sequence.
1. Message 6
2. Message 1
3. Message 8
4. Message 5
5. Message 2

After the transmit message search, the transmit message with the highest priority of the transmit message buffers that have a pending transmission request (message buffers with the TRQ bit set to 1 in advance) is transmitted.

If a new transmission request is set, the transmit message buffer with the new transmission request is compared with the transmit message buffer with a pending transmission request. If the new transmission request has a higher priority, it is transmitted, unless transmission of a message with a low priority has already started. If transmission of a message with a low priority has already started, however, the new transmission request is transmitted later. To solve this priority inversion effect, the software can perform a transmission abort request for the lower priority message. The highest priority is determined according to the following rules.

Priority	Conditions	Description
1(high)	Value of first 11 bits of ID [ID28 to ID18]:	The message frame with the lowest value represented by the first 11 bits of the ID is transmitted first. If the value of an 11-bit standard ID is equal to or smaller than the first 11 bits of a 29-bit extended ID, the 11-bit standard ID has a higher priority than message frame with the 29-bit extended ID.
2	Frame type	A data frame with an 11-bit standard ID (RTR bit is cleared to 0) has a higher priority than a remote frame with a standard ID and a message frame with an extended ID.
3	ID type	A message frame with a standard ID (IDE bit is cleared to 0) has a higher priority than a message frame with an extended ID.
4	Value of lower 18 bits of ID [ID17 to ID0]:	If more than one transmission-pending extended ID message frame have equal values in the first 11 bits of the ID and the same frame type (equal RTR bit values), the message frame with the lowest value in the lower 18 bits of its extended ID is transmitted first.
5(low)	Message buffer number	If two or more message buffers request transmission of message frames with the same ID, the message from the message buffer with the lowest message buffer number is transmitted first.

Remarks 1. If automatic block transmission request bit ABTTRG is set to 1 in the normal operation mode with ABT, the TRQ bit is set to 1 only for one message buffer in the ABT message buffer group.

If the ABT mode was triggered by ABTTRG bit, one TRQ bit is set to 1 in the ABT area (buffer 0 through 7). Beyond this TRQ bit, the application can request transmissions (set TRQ to 1) for other TX-message buffers that do not belong to the ABT area. In that case an interval arbitration process (TX-search) evaluates all TX-message buffers with TRQ bit set to 1 and chooses the message buffer that contains the highest prioritized identifier for the next transmission. If there are 2 or more identifiers that have the highest priority (i.e. identical identifiers), the message located at the lowest message buffer number is transmitted at first.

Upon successful transmission of a message frame, the following operations are performed.

- The TRQ flag of the corresponding transmit message buffer is automatically cleared to 0.
- The transmission completion status bit CINTS0 of the C0INTS register is set to 1 (if the interrupt enable bit (IE) of the corresponding transmit message buffer is set to 1).
- An interrupt request signal INTC0TRX output (if the CIE0 bit of the C0IE register is set to 1 and if the interrupt enable bit (IE) of the corresponding transmit message buffer is set to 1).

2. When changing the contents of a transmit buffer, the RDY flag of this buffer must be cleared before updating the buffer contents. As during internal transfer actions, the RDY flag may be locked temporarily, the status of RDY must be checked by software, after changing it.

3. m = 0 to 15

15.10.2 Transmit history list function

The transmit history list (THL) function records in the transmit history list the number of the transmit message buffer from which data or remote frames have been were sent. The THL consists of storage elements equivalent to up to seven messages, the last out-message pointer (LOPT) with the corresponding C0LOPT register, and the transmit history list get pointer (TGPT) with the corresponding C0TGPT register.

The THL is undefined immediately after the transition of the CAN module from the initialization mode to one of the operation modes.

The C0LOPT register holds the contents of the THL element indicated by the value of the LOPT pointer minus 1. By reading the C0LOPT register, therefore, the number of the message buffer that transmitted a data frame or remote frame first can be checked. The LOPT pointer is utilized as a write pointer that indicates to what part of the THL a message buffer number is recorded. Any time a data frame or remote frame is transmitted, the corresponding message buffer number is recorded to the THL element indicated by the LOPT pointer. Each time recording to the THL has been completed, the LOPT pointer is automatically incremented. In this way, the number of the message buffer that has received and stored a frame will be recorded chronologically.

The TGPT pointer is utilized as a read pointer that reads a recorded message buffer number from the THL. This pointer indicates the first THL element that the CPU has not yet read. By reading the C0TGPT register by software, the number of a message buffer that has completed transmission can be read. Each time a message buffer number is read from the C0TGPT register, the TGPT pointer is automatically incremented.

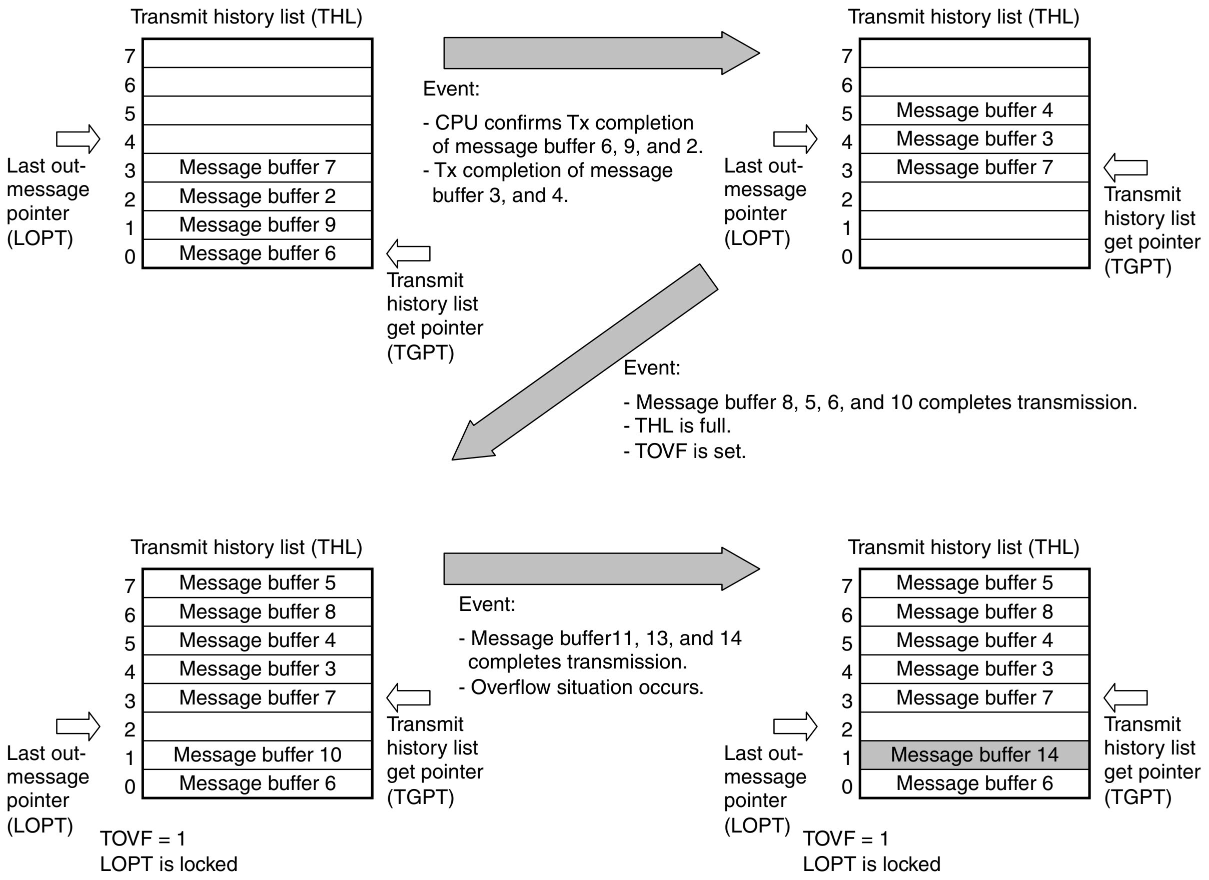
If the value of the TGPT pointer matches the value of the LOPT pointer, the THPM bit (transmit history list pointer match) of the C0TGPT register is set to 1. This indicates that no message buffer numbers that have not been read remain in the THL. If a new message buffer number is recorded, the LOPT pointer is incremented and because its value no longer matches the value of the TGPT pointer, the THPM bit is cleared. In other words, the numbers of the unread message buffers exist in the THL.

If the LOPT pointer is incremented and matches the value of the TGPT pointer minus 1, the TOVF bit (transmit history list overflow) of the C0TGPT register is set to 1. This indicates that the THL is full of message buffer numbers that have not been read. If a new message is received and stored, the message buffer number recorded last is overwritten by the number of the message buffer that transmitted its message afterwards. After the TOVF bit has been set (1), therefore, the recorded message buffer numbers in the THL do not completely reflect the chronological order. However the other transmitted messages can be found by a CPU search applied to all transmit message buffers unless the CPU has not overwritten a transmit object in one of these buffers beforehand. In total up to six transmission completions can occur without overflowing the THL.

Caution If the history list is in the overflow condition (TOVF is set), reading the history list contents is still possible, until the history list is empty (indicated by THPM flag set). Nevertheless, the history list remains in the overflow condition, until TOVF is cleared by software. If TOVF is not cleared, the THPM flag will also not be updated (cleared) upon successful transmission of a new message. This may lead to the situation, that THPM indicates an empty history list, although a successful transmission has taken place, while the history list is in the overflow state (TOVF and THPM are set).

Remark m = 0 to 15

Figure 15-32. Transmit History List


Transmit history list (THL)
7
6
5
4
3 Message buffer 7
2 Message buffer 2
1 Message buffer 9
0 Message buffer 6
Last out-message pointer (LOPT)
Transmit history list get pointer (TGPT)
Event:
- CPU confirms Tx completion of message buffer 6, 9, and 2.
- Tx completion of message buffer 3, and 4.
Transmit history list (THL)
7
6
5 Message buffer 4
4 Message buffer 3
3 Message buffer 7
2
1
0
Last out-message pointer (LOPT)
Transmit history list get pointer (TGPT)
Event:
- Message buffer 8, 5, 6, and 10 completes transmission.
- THL is full.
- TOVF is set.
Transmit history list (THL)
7 Message buffer 5
6 Message buffer 8
5 Message buffer 4
4 Message buffer 3
3 Message buffer 7
2
1 Message buffer 10
0 Message buffer 6
Last out-message pointer (LOPT)
Transmit history list get pointer (TGPT)
TOVF = 1
LOPT is locked
Event:
- Message buffer11, 13, and 14 completes transmission.
- Overflow situation occurs.
Transmit history list (THL)
7 Message buffer 5
6 Message buffer 8
5 Message buffer 4
4 Message buffer 3
3 Message buffer 7
2
1 Message buffer 14
0 Message buffer 6
Last out-message pointer (LOPT)
Transmit history list get pointer (TGPT)
TOVF = 1
LOPT is locked

TOVF = 1 denotes that LOPT equals TGPT−1 while message buffer number stored to element indicated by LOPT−1.

15.10.3 Automatic block transmission (ABT)

The automatic block transmission (ABT) function is used to transmit two or more data frames successively with no CPU interaction. The maximum number of transmit message buffers assigned to the ABT function is eight (message buffer numbers 0 to 7).

By setting OPMODE [2:0] bits of the C0CTRL register to 010B, "normal operation mode with automatic block transmission function" (hereafter referred to as ABT mode) can be selected.

To issue an ABT transmission request, define the message buffers by software first. Set the MA0 bit (1) in all the message buffers used for ABT, and define all the buffers as transmit message buffers by setting MT [2:0] bits to 000B. Be sure to set the ID for each message buffer for ABT even when the same ID is being used for all the message buffers. To use two or more IDs, set the ID of each message buffer by using the C0MIDLm and C0MIDHm registers. Set the C0MDLCm and C0MDATA0m to C0MDATA7m registers before issuing a transmission request for the ABT function.

After initialization of message buffers for ABT is finished, the RDY bit needs to be set (1). In the ABT mode, the TRQ bit does not have to be manipulated by software.

After the data for the ABT message buffers has been prepared, set the ABTTRG bit to 1. Automatic block transmission is then started. When ABT is started, the TRQ bit in the first message buffer (message buffer 0) is automatically set to 1. After transmission of the data of message buffer 0 has finished, TRQ bit of the next message buffer, message buffer 1, is set automatically. In this way, transmission is executed successively.

A delay time can be inserted by program in the interval in which the transmission request (TRQ) is automatically set while successive transmission is being executed. The delay time to be inserted is defined by the C0GMABTD register. The unit of the delay time is DBT (data bit time). DBT depends on the setting of the C0BRP and C0BTR registers.

Among transmit objects within the ABT-area, the priority of the transmission ID is not evaluated. The data of message buffers 0 to 7 are sequentially transmitted. When transmission of the data frame from message buffer 7 has been completed, the ABTTRG bit is automatically cleared to 0 and the ABT operation is finished.

If the RDY bit of an ABT message buffer is cleared during ABT, no data frame is transmitted from that buffer, ABT is stopped, and the ABTTRG bit is cleared. After that, transmission can be resumed from the message buffer where ABT stopped, by setting the RDY and ABTTRG bits to 1 by software. To not resume transmission from the message buffer where ABT stopped, the internal ABT engine can be reset by setting the ABTCLR bit to 1 while ABT mode is stopped and ABTTRG bit is cleared to 0. In this case, transmission is started from message buffer 0 if the ABTCLR bit is cleared to 0 and then the ABTTRG bit is set to 1.

An interrupt can be used to check if data frames have been transmitted from all the message buffers for ABT. To do so, the IE bit of the C0MCTRLm register of each message buffer except the last message buffer needs to be cleared (0).

If a transmit message buffer other than those used by the ABT function (message buffer 8 to 15) is assigned to a transmit message buffer, the message to be transmitted next is determined by the priority of the transmission ID of the ABT message buffer whose transmission is currently held pending and the transmission ID of the message buffers other than those used by the ABT function.

Transmission of a data frame from an ABT message buffer is not recorded in the transmit history list (THL).

Cautions 1. Set the ABTCLR bit to 1 while the ABTTRG bit is cleared to 0 in order to resume ABT operation at buffer No.0. If the ABTCLR bit is set to 1 while the ABTTRG bit is set to 1, the subsequent operation is not guaranteed.

2. If the automatic block transmission engine is cleared by setting the ABTCLR bit to 1, the ABTCLR bit is automatically cleared immediately after the processing of the clearing request is completed.

3. Do not set the ABTTRG bit in the initialization mode. If the ABTTRG bit is set in the initialization mode, the proper operation is not guaranteed after the mode is changed from the initialization mode to the ABT mode.

4. Do not set TRQ bit of the ABT message buffers to 1 by software in the normal operation mode with ABT. Otherwise, the operation is not guaranteed.

5. The C0GMABTD register is used to set the delay time that is inserted in the period from completion of the preceding ABT message to setting of the TRQ bit for the next ABT message when the transmission requests are set in the order of message numbers for each message for ABT that is successively transmitted in the ABT mode. The timing at which the messages are actually transmitted onto the CAN bus varies depending on the status of transmission from other stations and the status of the setting of the transmission request for messages other than the ABT messages (message buffer 8 to 15).

6. If a transmission request is made for a message other than an ABT message and if no delay time is inserted in the interval in which transmission requests for ABT are automatically set (C0GMABTD = 00H), messages other than ABT messages may be transmitted not depending on the priority of the ABT message.

7. Do not clear the RDY bit to 0 when ABTTRG = 1.

8. If a message is received from another node while normal operation mode with ABT is active, the TX-message from the ABT-area may be transmitted with delay of one frame although C0GMABTD register was set up with 00H.

Remark m = 0 to 15

15.10.4 Transmission abort process

(1) Transmission abort process except for in normal operation mode with automatic block transmission (ABT)

The user can clear the TRQ bit of the C0MCTRLm register to 0 to abort a transmission request. The TRQ bit will be cleared immediately if the abort was successful. Whether the transmission was successfully aborted or not can be checked using the TSTAT bit of the C0CTRL register and the C0TGPT register, which indicate the transmission status on the CAN bus (for details, refer to the processing in **Figure 15-47. Transmission Abort Processing (Except Normal Operation Mode with ABT)**).

(2) Transmission abort process except for ABT transmission in normal operation mode with automatic block transmission (ABT)

The user can clear the ABTTRG bit of the C0GMABT register to 0 to abort a transmission request. After checking the ABTTRG bit of the C0GMABT register = 0, clear the TRQ bit of the C0MCTRLm register to 0. The TRQ bit will be cleared immediately if the abort was successful. Whether the transmission was successfully aborted or not can be checked using the TSTAT bit of the C0CTRL register and the C0TGPT register, which indicate the transmission status on the CAN bus (for details, refer to the processing in **Figure 15-48. Transmission Abort Processing Except for ABT Transmission (Normal Operation Mode with ABT)**).

(3) Transmission abort process for ABT transmission in normal operation mode with automatic block transmission (ABT)

To abort ABT that is already started, clear the ABTTRG bit of the C0GMABT register to 0. In this case, the ABTTRG bit remains 1 if an ABT message is currently being transmitted and until the transmission is completed (successfully or not), and is cleared to 0 as soon as transmission is finished. This aborts ABT.

If the last transmission (before ABT) was successful, the normal operation mode with ABT is left with the internal ABT pointer pointing to the next message buffer to be transmitted.

In the case of an erroneous transmission, the position of the internal ABT pointer depends on the status of the TRQ bit in the last transmitted message buffer. If the TRQ bit is set to 1 when clearing the ABTTRG bit is requested, the internal ABT pointer points to the last transmitted message buffer (for details, refer to the process in **Figure 15-49. ABT Transmission Abort Processing (Normal Operation Mode with ABT)**). If the TRQ bit is cleared to 0 when clearing the ABTTRG bit is requested, the internal ABT pointer is incremented (+1) and points to the next message buffer in the ABT area (for details, refer to the process in **Figure 15-50. ABT Transmission Request Abort Processing (Normal Operation Mode with ABT)**).

Caution Be sure to abort ABT by clearing ABTTRG to 0. The operation is not guaranteed if aborting transmission is requested by clearing RDY bit.

When the normal operation mode with ABT is resumed after ABT has been aborted and ABTTRG bit is set to 1, the next ABT message buffer to be transmitted can be determined from the following table.

Status of TRQ of ABT Message Buffer	Abort After Successful Transmission	Abort after erroneous transmission
Set (1)	Next message buffer in the ABT area[Note]	Same message buffer in the ABT area
Cleared (0)	Next message buffer in the ABT area[Note]	Next message buffer in the ABT area[Note]

Note The above resumption operation can be performed only if a message buffer ready for ABT exists in the ABT area. For example, an abort request that is issued while ABT of message buffer 7 is in progress is regarded as completion of ABT, rather than abort, if transmission of message buffer 7 has been successfully completed, even if ABTTRG is cleared to 0. If the RDY bit in the next message buffer in the ABT area is cleared to 0, the internal ABT pointer is retained, but the resumption operation is not performed even if ABTTRG is set to 1, and ABT ends immediately.

Remark m = 0 to 15

15.10.5 Remote frame transmission

Remote frames can be transmitted only from transmit message buffers. Set whether a data frame or remote frame is transmitted via the RTR bit of the C0MCONFm register. Setting (1) the RTR bit sets remote frame transmission.

Remark m = 0 to 15

15.11 Power Save Modes

15.11.1 CAN sleep mode

The CAN sleep mode can be used to set the CAN controller to standby mode in order to reduce power consumption. The CAN module can enter the CAN sleep mode from all operation modes. Release of the CAN sleep mode returns the CAN module to exactly the same operation mode from which the CAN sleep mode was entered.

In the CAN sleep mode, the CAN module does not transmit messages, even when transmission requests are issued or pending.

(1) Entering CAN sleep mode

The CPU issues a CAN sleep mode transition request by writing 01B to the PSMODE [1:0] bits of the C0CTRL register.

This transition request is only acknowledged only under the following conditions.

- The CAN module is already in one of the following operation modes
  - Normal operation mode
  - Normal operation mode with ABT
  - Receive-only mode
  - Single-shot mode
  - Self-test mode
  - CAN stop mode in all the above operation modes
- The CAN bus state is bus idle (the 4th bit in the interframe space is recessive) [Note]
- No transmission request is pending

Note If the CAN bus is fixed to dominant, the request for transition to the CAN sleep mode is held pending. Also the transition from CAN stop mode to CAN sleep mode is independent of the CAN bus state.

Remark If a sleep mode request is pending, and at the same time a message is received in a message box, the sleep mode request is not cancelled, but is executed right after message storage has been finished. This may result in AFCAN being in sleep mode, while the CPU would execute the RX interrupt routine. Therefore, the interrupt routine must check the access to the message buffers as well as reception history list registers by using the MBON flag, if sleep mode is used.

If any one of the conditions mentioned above is not met, the CAN module will operate as follows.

- If the CAN sleep mode is requested from the initialization mode, the CAN sleep mode transition request is ignored and the CAN module remains in the initialization mode.
- If the CAN bus state is not bus idle (i.e., the CAN bus state is either transmitting or receiving) when the CAN sleep mode is requested in one of the operation modes, immediate transition to the CAN sleep mode is not possible. In this case, the CAN sleep mode transition request is held pending until the CAN bus state becomes bus idle (the 4th bit in the interframe space is recessive). In the time from the CAN sleep mode request to successful transition, the PSMODE [1:0] bits remain 00B. When the module has entered the CAN sleep mode, PSMODE [1:0] bits are set to 01B.
- If a request for transition to the initialization mode and a request for transition to the CAN sleep are made at the same time while the CAN module is in one of the operation modes, the request for the initialization mode is enabled. The CAN module enters the initialization mode at a predetermined timing. At this time, the CAN sleep mode request is not held pending and is ignored.

- Even when initialization mode and sleep mode are not requested simultaneously (i.e the first request has not been granted while the second request is made), the request for initialization has priority over the sleep mode request. The sleep mode request is cancelled when the initialization mode is requested. When a pending request for initialization mode is present, a subsequent request for Sleep mode request is cancelled right at the point in time where it was submitted.

(2) Status in CAN sleep mode

The CAN module is in one of the following states after it enters the CAN sleep mode.

- The internal operating clock is stopped and the power consumption is minimized.
- The function to detect the falling edge of the CAN reception pin (CRxD) remains in effect to wake up the CAN module from the CAN bus.
- To wake up the CAN module from the CPU, data can be written to PSMODE [1:0] of the CAN module control register (C0CTRL), but nothing can be written to other CAN module registers or bits.
- The CAN module registers can be read, except for C0LIPT, C0RGPT, C0LOPT, and C0TGPT.
- The CAN message buffer registers cannot be written or read.
- MBON bit of the CAN global control register (C0GMCTRL) is cleared.
- A request for transition to the initialization mode is not acknowledged and is ignored.

(3) Releasing CAN sleep mode

The CAN sleep mode is released by the following events.

- When the CPU writes 00B to the PSMODE [1:0] bits of the C0CTRL register
- A falling edge at the CAN reception pin (CRxD) (i.e. the CAN bus level shifts from recessive to dominant)

Cautions 1. Even if the falling edge belongs to the SOF of a receive message, this message will not be received and stored. If the CPU has turned off the clock to the CAN while the CAN was in sleep mode, even subsequently the CAN sleep mode will not be released and PSMODE [1:0] will continue to be 01B unless the clock to the CAN is supplied again. In addition to this, the receive message will not be received after that.

2. If the falling edge on the CAN reception pin (CRxD) is detected in the state that the CAN clock is supplied, it is necessary to clear the PSMODE0 bit by software (for details, refer to the processing in Figure 15-54. Setting CAN Sleep Mode/Stop Mode).

After releasing the sleep mode, the CAN module returns to the operation mode from which the CAN sleep mode was requested and the PSMODE [1:0] bits of the C0CTRL register are reset to 00B. If the CAN sleep mode is released by a change in the CAN bus state, the CINTS5 bit of the C0INTS register is set to 1, regardless of the CIE bit of the C0IE register. After the CAN module is released from the CAN sleep mode, it participates in the CAN bus again by automatically detecting 11 consecutive recessive-level bits on the CAN bus. The user application has to wait until MBON = 1, before accessing message buffers again.
When a request for transition to the initialization mode is made while the CAN module is in the CAN sleep mode, that request is ignored; the CPU has to be released from sleep mode by software first before entering the initialization mode.

Caution Be aware that the release of CAN sleep mode by CAN bus event, and thus the wake up interrupt may happen at any time, even right after requesting sleep mode, if a CAN bus event occurs.

Remark m = 0 to 15

15.11.2 CAN stop mode

The CAN stop mode can be used to set the CAN controller to standby mode to reduce power consumption. The CAN module can enter the CAN stop mode only from the CAN sleep mode. Release of the CAN stop mode puts the CAN module in the CAN sleep mode.

The CAN stop mode can only be released (entering CAN sleep mode) by writing 01B to the PSMODE [1:0] bits of the C0CTRL register and not by a change in the CAN bus state. No message is transmitted even when transmission requests are issued or pending.

(1) Entering CAN stop mode

A CAN stop mode transition request is issued by writing 11B to the PSMODE [1:0] bits of the C0CTRL register.

A CAN stop mode request is only acknowledged when the CAN module is in the CAN sleep mode. In all other modes, the request is ignored.

Caution To set the CAN module to the CAN stop mode, the module must be in the CAN sleep mode. To confirm that the module is in the sleep mode, check that PSMODE [1:0] = 01B, and then request the CAN stop mode. If a bus change occurs at the CAN reception pin (CRxD) while this process is being performed, the CAN sleep mode is automatically released. In this case, the CAN stop mode transition request cannot be acknowledged (However, in the state that the CAN clock is supplied, it is necessary to clear the PSMODE0 bit by software after a bus change occurs at the CAN reception pin (CRxD)).

(2) Status in CAN stop mode

The CAN module is in one of the following states after it enters the CAN stop mode.

- The internal operating clock is stopped and the power consumption is minimized.
- To wake up the CAN module from the CPU, data can be written to PSMODE [1:0] of the CAN module control register (C0CTRL), but nothing can be written to other CAN module registers or bits.
- The CAN module registers can be read, except for C0LIPT, C0RGPT, C0LOPT, and C0TGPT.
- The CAN message buffer registers cannot be written or read.
- MBON bit of the CAN global control register (C0GMCTRL) is cleared.
- An initialization mode transition request is not acknowledged and is ignored.

(3) Releasing CAN stop mode

The CAN stop mode can only be released by writing 01B to the PSMODE [1:0] bits of the C0CTRL register. After releasing the CAN stop mode, the CAN module enters the CAN sleep mode.

When the initialization mode is requested while the CAN module is in the CAN stop mode, that request is ignored; the CPU has to release the stop mode and subsequently CAN sleep mode before entering the initialization mode. It is impossible to enter the other operation mode directly from the CAN stop mode not entering the CAN sleep mode, that request is ignored.

Remark m = 0 to 15

15.11.3 Example of using power saving modes

In some application systems, it may be necessary to place the CPU in a power saving mode to reduce the power consumption. By using the power saving mode specific to the CAN module and the power saving mode specific to the CPU in combination, the CPU can be woken up from the power saving status by the CAN bus.

Here is an example of using the power saving modes.

First, put the CAN module in the CAN sleep mode (PSMODE = 01B). Next, put the CPU in the power saving mode. If an edge transition from recessive to dominant is detected at the CAN reception pin (CRxD) in this status, the CINTS5 bit in the CAN module is set to 1. If the CIE5 bit of the C0CTRL register is set to 1, a wakeup interrupt (INTC0WUP) is generated. The CAN module is automatically released from the CAN sleep mode (PSMODE = 00B) and returns to the normal operation mode (However, in the state that the CAN clock is supplied, it is necessary to clear the PSMODE0 bit by software after a bus change is detected at the CAN reception pin (CRxD)). The CPU, in response to INTC0WUP, can release its own power saving mode and return to the normal operation mode.

To further reduce the power consumption of the CPU, the internal clocks, including that of the CAN module, may be stopped. In this case, the operating clock supplied to the CAN module is stopped after the CAN module is put in the CAN sleep mode. Then the CPU enters a power saving mode in which the clock supplied to the CPU is stopped. If an edge transition from recessive to dominant is detected at the CAN reception pin (CRxD) in this status, the CAN module can set the CINTS5 bit to 1 and generate the wakeup interrupt (INTC0WUP) even if it is not supplied with the clock. The other functions, however, do not operate because clock supply to the CAN module is stopped, and the module remains in the CAN sleep mode. The CPU, in response to INTC0WUP, releases its power saving mode, resumes supply of the internal clocks, including the clock to the CAN module, after the oscillation stabilization time has elapsed, and starts instruction execution. The CAN module is immediately released from the CAN sleep mode when clock supply is resumed, and returns to the normal operation mode (PSMODE = 00B).

15.12 Interrupt Function

The CAN module provides 6 different interrupt sources.

The occurrence of these interrupt sources is stored in interrupt status registers. Four separate interrupt request signals are generated from the six interrupt sources. When an interrupt request signal that corresponds to two or more interrupt sources is generated, the interrupt sources can be identified by using an interrupt status register. After an interrupt source has occurred, the corresponding interrupt status bit must be cleared to 0 by software.

Table 15-20. List of CAN Module Interrupt Sources

No.	Interrupt Status Bit		Interrupt Enable Bit		Interrupt Request Signal	Interrupt Source Description
	Name	Register	Name	Register		
1	CINTS0 Note 1	C0INTS	CIE0 Note 1	C0IE	INTC0TRX	Message frame successfully transmitted from message buffer m
2	CINTS1 Note 1	C0INTS	CIE1 Note 1	C0IE	INTC0REC	Valid message frame reception in message buffer m
3	CINTS2	C0INTS	CIE2	C0IE	INTC0ERR	CAN module error state interrupt Note 2
4	CINTS3	C0INTS	CIE3	C0IE		CAN module protocol error interrupt Note 3
5	CINTS4	C0INTS	CIE4	C0IE		CAN module arbitration loss interrupt
6	CINTS5	C0INTS	CIE5	C0IE	INTC0WUP	CAN module wakeup interrupt from CAN sleep mode Note 4

Notes 1. The IE bit (message buffer interrupt enable bit) in the C0MCTRL register of the corresponding message buffer has to be set to 1 for that message buffer to participate in the interrupt generation process.

2. This interrupt is generated when the transmission/reception error counter is at the warning level, or in the error passive or bus-off state.

3. This interrupt is generated when a stuff error, form error, ACK error, bit error, or CRC error occurs.

4. This interrupt is generated when the CAN module is woken up from the CAN sleep mode because a falling edge is detected at the CAN reception pin (CAN bus transition from recessive to dominant).

Remark m = 0 to 15

15.13 Diagnosis Functions and Special Operational Modes

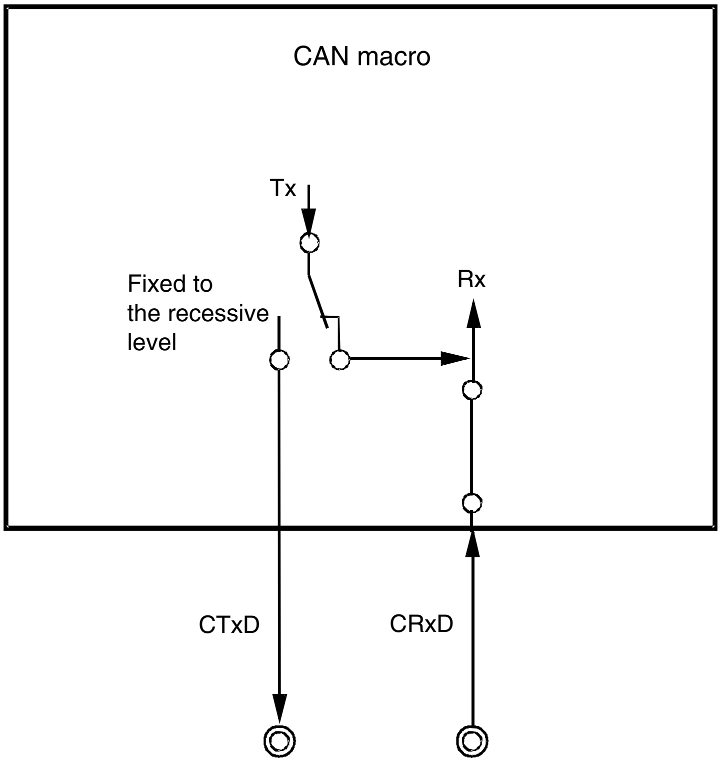
The CAN module provides a receive-only mode, single-shot mode, and self-test mode to support CAN bus diagnosis functions or the operation of specific CAN communication methods.

15.13.1 Receive-only mode

The receive-only mode is used to monitor receive messages without causing any interference on the CAN bus and can be used for CAN bus analysis nodes.

For example, this mode can be used for automatic baud-rate detection. The baud rate in the CAN module is changed until "valid reception" is detected, so that the baud rates in the module match ("valid reception" means a message frame has been received in the CAN protocol layer without occurrence of an error and with an appropriate ACK between nodes connected to the CAN bus). A valid reception does not require message frames to be stored in a receive message buffer (data frames) or transmit message buffer (remote frames). The event of valid reception is indicated by setting the VALID bit of the C0CTRL register (1).

Figure 15-33. CAN Module Terminal Connection in Receive-Only Mode



In the receive-only mode, no message frames can be transmitted from the CAN module to the CAN bus. Transmit requests issued for message buffers defined as transmit message buffers are held pending.

In the receive-only mode, the CAN transmission pin (CTxD) in the CAN module is fixed to the recessive level. Therefore, no active error flag can be transmitted from the CAN module to the CAN bus even when a CAN bus error is detected while receiving a message frame. Since no transmission can be issued from the CAN module, the transmission error counter TEC is never updated. Therefore, a CAN module in the receive-only mode does not enter the bus-off state.

Furthermore, ACK is not returned to the CAN bus in this mode upon the valid reception of a message frame. Internally, the local node recognizes that it has transmitted ACK. An overload frame cannot be transmitted to the CAN bus.

Caution If only two CAN nodes are connected to the CAN bus and one of them is operating in the receive-only mode, there is no ACK on the CAN bus. Due to the missing ACK, the transmitting node will transmit an active error flag, and repeat transmitting a message frame. The transmitting node becomes error passive after transmitting the message frame 16 times (assuming that the error counter was 0 in the beginning and no other errors have occurred). After the message frame for the 17th time is transmitted, the transmitting node generates a passive error flag. The receiving node in the receive-only mode detects the first valid message frame at this point, and the VALID bit is set to 1 for the first time.

15.13.2 Single-shot mode

In the single-shot mode, automatic re-transmission as defined in the CAN protocol is switched off (According to the CAN protocol, a message frame transmission that has been aborted by either arbitration loss or error occurrence has to be repeated without control by software.). All other behavior of single shot mode is identical to normal operation mode. Features of single shot mode can not be used in combination with normal mode with ABT.

The single-shot mode disables the re-transmission of an aborted message frame transmission according to the setting of the AL bit of the C0CTRL register. When the AL bit is cleared to 0, re-transmission upon arbitration loss and upon error occurrence is disabled. If the AL bit is set to 1, re-transmission upon error occurrence is disabled, but re-transmission upon arbitration loss is enabled. As a consequence, the TRQ bit in a message buffer defined as a transmit message buffer is cleared to 0 by the following events.

- Successful transmission of the message frame
- Arbitration loss while sending the message frame
- Error occurrence while sending the message frame

The events arbitration loss and error occurrence can be distinguished by checking the CINTS4 and CINTS3 bits of the C0INTS register respectively, and the type of the error can be identified by reading the LEC[2:0] bits of the C0LEC register.

Upon successful transmission of the message frame, the transmit completion interrupt bit CINTS0 of the C0INTS register is set to 1. If the CIE0 bit of the C0IE register is set to 1 at this time, an interrupt request signal is output.

The single-shot mode can be used when emulating time-triggered communication methods (e.g. TTCAN level 1).

Caution The AL bit is only valid in Single-shot mode. It does not influence the operation of re-transmission upon arbitration loss in the other operation modes.

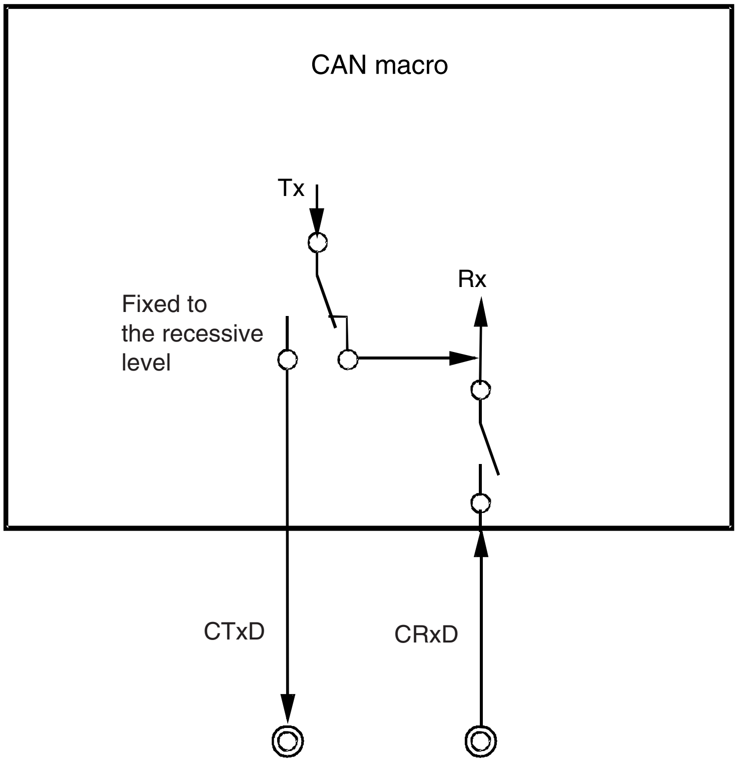
15.13.3 Self-test mode

In the self-test mode, message frame transmission and message frame reception can be tested without connecting the CAN node to the CAN bus or without affecting the CAN bus.

In the self-test mode, the CAN module is completely disconnected from the CAN bus, but transmission and reception are internally looped back. The CAN transmission pin (CTxD) is fixed to the recessive level.

If the falling edge on the CAN reception pin (CRxD) is detected after the CAN module has entered the CAN sleep mode from the self-test mode, however, the module is released from the CAN sleep mode in the same manner as the other operation modes (However, to release the CAN sleep mode in the state that the CAN clock is supplied, it is necessary to clear the PSMODE0 bit by software after the falling edge on the CAN reception pin (CRxD) is detected). To keep the module in the CAN sleep mode, use the CAN reception pin (CRxD) as a port pin.

Figure 15-34. CAN Module Terminal Connection in Self-test Mode



15.13.4 Receive/transmit operation in each operation mode

Table 15-21 shows outline of the receive/transmit operation in each operation mode.

Table 15-21. Outline of the Receive/Transmit in Each Operation Mode

Operation Mode	Transmission of data/ remote frame	Transmission of ACK	Transmission of error/ overload frame	Transmission retry	Automatic Block Transmission (ABT)	Set of VALID bit	Store Data to message buffer
Initialization Mode	No	No	No	No	No	No	No
Normal Operation Mode	Yes	Yes	Yes	Yes	No	Yes	Yes
Normal Operation Mode with ABT	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Receive-only mode	No	No	No	No	No	Yes	Yes
Single-shot Mode	Yes	Yes	Yes	No Note 1	No	Yes	Yes
Self-test Mode	Yes Note 2	Yes Note 2	Yes Note 2	Yes Note 2	No	Yes Note 2	Yes Note 2

Notes 1. When the arbitration lost occurs, control of re-transmission is possible by the AL bit of C0CTRL register.

2. Each signals are not generated to outside, but generated into the CAN module.

15.14 Time Stamp Function

CAN is an asynchronous, serial protocol. All nodes connected to the CAN bus have a local, autonomous clock. As a consequence, the clocks of the nodes have no relation (i.e., the clocks are asynchronous and may have different frequencies).

In some applications, however, a common time base over the network (= global time base) is needed. In order to build up a global time base, a time stamp function is used. The essential mechanism of a time stamp function is the capture of timer values triggered by signals on the CAN bus.

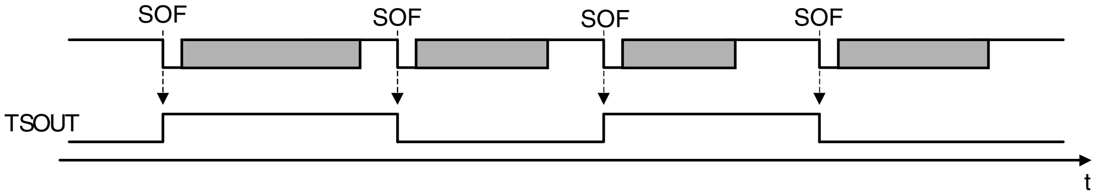
15.14.1 Time stamp function

The CAN controller supports the capturing of timer values triggered by a specific frame. An on-chip 16-bit capture timer unit in a microcontroller system is used in addition to the CAN controller. The 16-bit capture timer unit captures the timer value according to a trigger signal (TSOUT) for capturing that is output when a data frame is received from the CAN controller. The CPU can retrieve the time of occurrence of the capture event, i.e., the time stamp of the message received from the CAN bus, by reading the captured value. TSOUT signal can be selected from the following two event sources and is specified by the TSSEL bit of the C0TS register.

- SOF event (start of frame)	(TSSEL = 0)
- EOF event (last bit of end of frame)	(TSSEL = 1)

The TSOUT signal is enabled by setting the TSEN bit of the C0TS register to 1.

Figure 15-35. Timing Diagram of Capture Signal TSOUT



TSOUT signal toggles its level upon occurrence of the selected event during data frame reception (in the above timing diagram, the SOF is used as the trigger event source). To capture a timer value by using TSOUT signal, the capture timer unit must detect the capture signal at both the rising edge and falling edge.

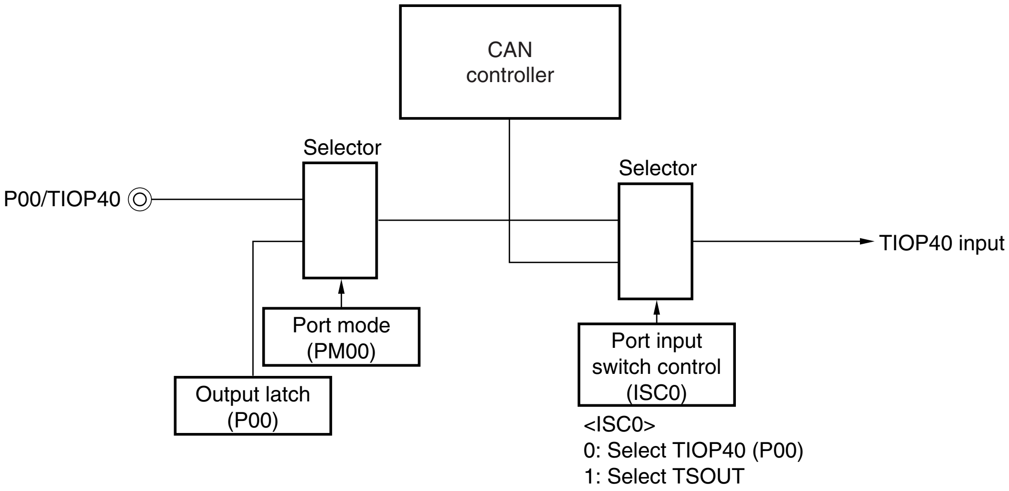
This time stamp function is controlled by the TSLOCK bit of the C0TS register. When TSLOCK is cleared to 0, TSOUT bit toggles upon occurrence of the selected event. If TSLOCK bit is set to 1, TSOUT toggles upon occurrence of the selected event, but the toggle is stopped as the TSEN bit is automatically cleared to 0 as soon as the message storing to the message buffer 0 starts. This suppresses the subsequent toggle occurrence by TSOUT, so that the time stamp value toggled last (= captured last) can be saved as the time stamp value of the time at which the data frame was received in message buffer 0.

Caution The time stamp function using TSLOCK bit is to stop toggle of TSOUT bit by receiving a data frame in message buffer 0. Therefore, message buffer 0 must be set as a receive message buffer. Since a receive message buffer cannot receive a remote frame, toggle of TSOUT bit cannot be stopped by reception of a remote frame. Toggle of TSOUT bit does not stop when a data frame is received in a message buffer other than message buffer 0.

For these reasons, a data frame cannot be received in message buffer 0 when the CAN module is in the normal operation mode with ABT, because message buffer 0 must be set as a transmit message buffer. In this operation mode, therefore, the function to stop toggle of TSOUT bit by TSLOCK bit cannot be used.

The input source of the timer value according to a trigger signal (TSOUT) can be input to the 16-bit timer/event counter 00 by port input switch control (ISC0), without connecting TIOP40, externally.

Figure 15-36. Port Input Switch Control



Remark ISC0: Bit 0 of the input switch control register (ISC) (see **Figure 12-19 Format of Input Switch Control Register (ISC)**)

15.15 Baud Rate Settings

15.15.1 Baud rate settings

Make sure that the settings are within the range of limit values for ensuring correct operation of the CAN controller, as follows.

(a) 5TQ ≤ SPT (sampling point) ≤ 17TQ
SPT = TSEG1 + 1TQ

(b) 8TQ ≤ DBT (data bit time) ≤ 25TQ
DBT = TSEG1 + TSEG2 + 1TQ = TSEG2 + SPT

(c) 1TQ ≤ SJW (synchronization jump width) ≤ 4TQ
SJW ≤ DBT – SPT

(d) 4TQ ≤ TSEG1 ≤ 16TQ [3 (Setting value of TSEG1 [3:0] ≤ 15]

(e) 1TQ ≤ TSEG2 ≤ 8TQ [0 (Setting value of TSEG2 [2:0] ≤ 7]

Remark TQ = 1/f_{TQ} (f_{TQ}: CAN protocol layer basic system clock)
TSEG1 [3:0]: Bits 3 to 0 of CAN0 bit rate register (C0BTR)
TSEG2 [2:0]: Bits 10 to 8 of CAN0 bit rate register (C0BTR)

Table 15-22 shows the combinations of bit rates that satisfy the above conditions.

Table 15-22. Settable Bit Rate Combinations (1/3)

Valid Bit Rate Setting					C0BTR Register Setting Value		Sampling Point (Unit %)
DBT Length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1[3:0]	TSEG2[2:0]	
25	1	8	8	8	1111	111	68.0
24	1	7	8	8	1110	111	66.7
24	1	9	7	7	1111	110	70.8
23	1	6	8	8	1101	111	65.2
23	1	8	7	7	1110	110	69.6
23	1	10	6	6	1111	101	73.9
22	1	5	8	8	1100	111	63.6
22	1	7	7	7	1101	110	68.2
22	1	9	6	6	1110	101	72.7
22	1	11	5	5	1111	100	77.3
21	1	4	8	8	1011	111	61.9
21	1	6	7	7	1100	110	66.7
21	1	8	6	6	1101	101	71.4
21	1	10	5	5	1110	100	76.2
21	1	12	4	4	1111	011	81.0
20	1	3	8	8	1010	111	60.0
20	1	5	7	7	1011	110	65.0
20	1	7	6	6	1100	101	70.0
20	1	9	5	5	1101	100	75.0
20	1	11	4	4	1110	011	80.0
20	1	13	3	3	1111	010	85.0
19	1	2	8	8	1001	111	57.9
19	1	4	7	7	1010	110	63.2
19	1	6	6	6	1011	101	68.4
19	1	8	5	5	1100	100	73.7
19	1	10	4	4	1101	011	78.9
19	1	12	3	3	1110	010	84.2
19	1	14	2	2	1111	001	89.5
18	1	1	8	8	1000	111	55.6
18	1	3	7	7	1001	110	61.1
18	1	5	6	6	1010	101	66.7
18	1	7	5	5	1011	100	72.2
18	1	9	4	4	1100	011	77.8
18	1	11	3	3	1101	010	83.3
18	1	13	2	2	1110	001	88.9
18	1	15	1	1	1111	000	94.4

Table 15-22. Settable Bit Rate Combinations (2/3)

Valid Bit Rate Setting					C0BTR Register Setting Value		Sampling Point (Unit %)
DBT Length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1[3:0]	TSEG2[2:0]	
17	1	2	7	7	1000	110	58.8
17	1	4	6	6	1001	101	64.7
17	1	6	5	5	1010	100	70.6
17	1	8	4	4	1011	011	76.5
17	1	10	3	3	1100	010	82.4
17	1	12	2	2	1101	001	88.2
17	1	14	1	1	1110	000	94.1
16	1	1	7	7	0111	110	56.3
16	1	3	6	6	1000	101	62.5
16	1	5	5	5	1001	100	68.8
16	1	7	4	4	1010	011	75.0
16	1	9	3	3	1011	010	81.3
16	1	11	2	2	1100	001	87.5
16	1	13	1	1	1101	000	93.8
15	1	2	6	6	0111	101	60.0
15	1	4	5	5	1000	100	66.7
15	1	6	4	4	1001	011	73.3
15	1	8	3	3	1010	010	80.0
15	1	10	2	2	1011	001	86.7
15	1	12	1	1	1100	000	93.3
14	1	1	6	6	0110	101	57.1
14	1	3	5	5	0111	100	64.3
14	1	5	4	4	1000	011	71.4
14	1	7	3	3	1001	010	78.6
14	1	9	2	2	1010	001	85.7
14	1	11	1	1	1011	000	92.9
13	1	2	5	5	0110	100	61.5
13	1	4	4	4	0111	011	69.2
13	1	6	3	3	1000	010	76.9
13	1	8	2	2	1001	001	84.6
13	1	10	1	1	1010	000	92.3
12	1	1	5	5	0101	100	58.3
12	1	3	4	4	0110	011	66.7
12	1	5	3	3	0111	010	75.0
12	1	7	2	2	1000	001	83.3
12	1	9	1	1	1001	000	91.7

Table 15-22. Settable Bit Rate Combinations (3/3)

Valid Bit Rate Setting					C0BTR Register Setting Value		Sampling Point (Unit %)
DBT Length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1[3:0]	TSEG2[2:0]	
11	1	2	4	4	0101	011	63.6
11	1	4	3	3	0110	010	72.7
11	1	6	2	2	0111	001	81.8
11	1	8	1	1	1000	000	90.9
10	1	1	4	4	0100	011	60.0
10	1	3	3	3	0101	010	70.0
10	1	5	2	2	0110	001	80.0
10	1	7	1	1	0111	000	90.0
9	1	2	3	3	0100	010	66.7
9	1	4	2	2	0101	001	77.8
9	1	6	1	1	0110	000	88.9
8	1	1	3	3	0011	010	62.5
8	1	3	2	2	0100	001	75.0
8	1	5	1	1	0101	000	87.5
7[Note]	1	2	2	2	0011	001	71.4
7[Note]	1	4	1	1	0100	000	85.7
6[Note]	1	1	2	2	0010	001	66.7
6[Note]	1	3	1	1	0011	000	83.3
5[Note]	1	2	1	1	0010	000	80.0
4[Note]	1	1	1	1	0001	000	75.0

Note Setting with a DBT value of 7 or less is valid only when the value of the C0BRP register is other than 00H.

Caution The values in Table 15-22 do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

15.15.2 Representative examples of baud rate settings

Tables 16-23 and **16-24** show representative examples of baud rate setting.

Table 15-23. Representative Examples of Baud Rate Settings (f_{CANMOD} = 8 MHz) (1/2)

Set Baud Rate Value (Unit: kbps)	Division Ratio of C0BRP	C0BRP Register Set Value	Valid Bit Rate Setting (Unit: kbps)					C0BTR Register Setting Value		Sampling point (Unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	TSEG1 [3:0]	TSEG2 [2:0]	
1000	1	00000000	8	1	1	3	3	0011	010	62.5
1000	1	00000000	8	1	3	2	2	0100	001	75.0
1000	1	00000000	8	1	5	1	1	0101	000	87.5
500	1	00000000	16	1	1	7	7	0111	110	56.3
500	1	00000000	16	1	3	6	6	1000	101	62.5
500	1	00000000	16	1	5	5	5	1001	100	68.8
500	1	00000000	16	1	7	4	4	1010	011	75.0
500	1	00000000	16	1	9	3	3	1011	010	81.3
500	1	00000000	16	1	11	2	2	1100	001	87.5
500	1	00000000	16	1	13	1	1	1101	000	93.8
500	2	00000001	8	1	1	3	3	0011	010	62.5
500	2	00000001	8	1	3	2	2	0100	001	75.0
500	2	00000001	8	1	5	1	1	0101	000	87.5
250	2	00000001	16	1	1	7	7	0111	110	56.3
250	2	00000001	16	1	3	6	6	1000	101	62.5
250	2	00000001	16	1	5	5	5	1001	100	68.8
250	2	00000001	16	1	7	4	4	1010	011	75.0
250	2	00000001	16	1	9	3	3	1011	010	81.3
250	2	00000001	16	1	11	2	2	1100	001	87.5
250	2	00000001	16	1	13	1	1	1101	000	93.8
250	4	00000011	8	1	3	2	2	0100	001	75.0
250	4	00000011	8	1	5	1	1	0101	000	87.5
125	4	00000011	16	1	1	7	7	0111	110	56.3
125	4	00000011	16	1	3	6	6	1000	101	62.5
125	4	00000011	16	1	5	5	5	1001	100	68.8
125	4	00000011	16	1	7	4	4	1010	011	75.0
125	4	00000011	16	1	9	3	3	1011	010	81.3
125	4	00000011	16	1	11	2	2	1100	001	87.5
125	4	00000011	16	1	13	1	1	1101	000	93.8
125	8	00000111	8	1	3	2	2	0100	001	75.0
125	8	00000111	8	1	5	1	1	0101	000	87.5

Caution The values in Table 15-23 do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

Table 15-23. Representative Examples of Baud Rate Settings (f_{CANMOD} = 8 MHz) (2/2)

Set Baud Rate Value (Unit: kbps)	Division Ratio of C0BRP	C0BRP Register Set Value	Valid Bit Rate Setting (Unit: kbps)					C0BTR Register Setting Value		Sampling point (Unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	TSEG1 [3:0]	TSEG2 [2:0]	
100	4	00000011	20	1	7	6	6	1100	101	70.0
100	4	00000011	20	1	9	5	5	1101	100	75.0
100	5	00000100	16	1	7	4	4	1010	011	75.0
100	5	00000100	16	1	9	3	3	1011	010	81.3
100	8	00000111	10	1	3	3	3	0101	010	70.0
100	8	00000111	10	1	5	2	2	0110	001	80.0
100	10	00001001	8	1	3	2	2	0100	001	75.0
100	10	00001001	8	1	5	1	1	0101	000	87.5
83.3	4	00000011	24	1	7	8	8	1110	111	66.7
83.3	4	00000011	24	1	9	7	7	1111	110	70.8
83.3	6	00000101	16	1	5	5	5	1001	100	68.8
83.3	6	00000101	16	1	7	4	4	1010	011	75.0
83.3	6	00000101	16	1	9	3	3	1011	010	81.3
83.3	6	00000101	16	1	11	2	2	1100	001	87.5
83.3	8	00000111	12	1	5	3	3	0111	010	75.0
83.3	8	00000111	12	1	7	2	2	1000	001	83.3
83.3	12	00001011	8	1	3	2	2	0100	001	75.0
83.3	12	00001011	8	1	5	1	1	0101	000	87.5
33.3	10	00001001	24	1	7	8	8	1110	111	66.7
33.3	10	00001001	24	1	9	7	7	1111	110	70.8
33.3	12	00001011	20	1	7	6	6	1100	101	70.0
33.3	12	00001011	20	1	9	5	5	1101	100	75.0
33.3	15	00001110	16	1	7	4	4	1010	011	75.0
33.3	15	00001110	16	1	9	3	3	1011	010	81.3
33.3	16	00001111	15	1	6	4	4	1001	011	73.3
33.3	16	00001111	15	1	8	3	3	1010	010	80.0
33.3	20	00010011	12	1	5	3	3	0111	010	75.0
33.3	20	00010011	12	1	7	2	2	1000	001	83.3
33.3	24	00010111	10	1	3	3	3	0101	010	70.0
33.3	24	00010111	10	1	5	2	2	0110	001	80.0
33.3	30	00011101	8	1	3	2	2	0100	001	75.0
33.3	30	00011101	8	1	5	1	1	0101	000	87.5

Caution **The values in Table 15-23 do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.**

Table 15-24. Representative Examples of Baud Rate Settings (f_{CANMOD} = 16 MHz) (1/2)

Set Baud Rate Value (Unit: kbps)	Division Ratio of C0BRP	C0BRP Register Set Value	Valid Bit Rate Setting (Unit: kbps)					C0BTR Register Setting Value		Sampling point (Unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	TSEG1 [3:0]	TSEG2 [2:0]	
1000	1	00000000	16	1	1	7	7	0111	110	56.3
1000	1	00000000	16	1	3	6	6	1000	101	62.5
1000	1	00000000	16	1	5	5	5	1001	100	68.8
1000	1	00000000	16	1	7	4	4	1010	011	75.0
1000	1	00000000	16	1	9	3	3	1011	010	81.3
1000	1	00000000	16	1	11	2	2	1100	001	87.5
1000	1	00000000	16	1	13	1	1	1101	000	93.8
1000	2	00000001	8	1	3	2	2	0100	001	75.0
1000	2	00000001	8	1	5	1	1	0101	000	87.5
500	2	00000001	16	1	1	7	7	0111	110	56.3
500	2	00000001	16	1	3	6	6	1000	101	62.5
500	2	00000001	16	1	5	5	5	1001	100	68.8
500	2	00000001	16	1	7	4	4	1010	011	75.0
500	2	00000001	16	1	9	3	3	1011	010	81.3
500	2	00000001	16	1	11	2	2	1100	001	87.5
500	2	00000001	16	1	13	1	1	1101	000	93.8
500	4	00000011	8	1	3	2	2	0100	001	75.0
500	4	00000011	8	1	5	1	1	0101	000	87.5
250	4	00000011	16	1	3	6	6	1000	101	62.5
250	4	00000011	16	1	5	5	5	1001	100	68.8
250	4	00000011	16	1	7	4	4	1010	011	75.0
250	4	00000011	16	1	9	3	3	1011	010	81.3
250	4	00000011	16	1	11	2	2	1100	001	87.5
250	8	00000111	8	1	3	2	2	0100	001	75.0
250	8	00000111	8	1	5	1	1	0101	000	87.5
125	8	00000111	16	1	3	6	6	1000	101	62.5
125	8	00000111	16	1	7	4	4	1010	011	75.0
125	8	00000111	16	1	9	3	3	1011	010	81.3
125	8	00000111	16	1	11	2	2	1100	001	87.5
125	16	00001111	8	1	3	2	2	0100	001	75.0
125	16	00001111	8	1	5	1	1	0101	000	87.5

Caution The values in Table 15-24 do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

Table 15-24. Representative Examples of Baud Rate Settings (f_{CANMOD} = 16 MHz) (2/2)

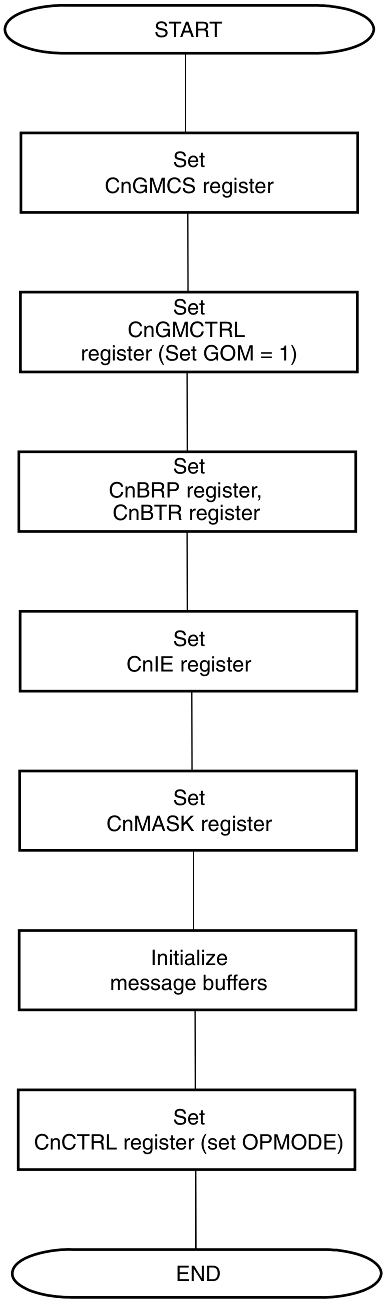
Set Baud Rate Value (Unit: kbps)	Division Ratio of C0BRP	C0BRP Register Set Value	Valid Bit Rate Setting (Unit: kbps)					C0BTR Register Setting Value		Sampling point (Unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	TSEG1 [3:0]	TSEG2 [2:0]	
100	8	00000111	20	1	9	5	5	1101	100	75.0
100	8	00000111	20	1	11	4	4	1110	011	80.0
100	10	00001001	16	1	7	4	4	1010	011	75.0
100	10	00001001	16	1	9	3	3	1011	010	81.3
100	16	00001111	10	1	3	3	3	0101	010	70.0
100	16	00001111	10	1	5	2	2	0110	001	80.0
100	20	00010011	8	1	3	2	2	0100	001	75.0
83.3	8	00000111	24	1	7	8	8	1110	111	66.7
83.3	8	00000111	24	1	9	7	7	1111	110	70.8
83.3	12	00001011	16	1	7	4	4	1010	011	75.0
83.3	12	00001011	16	1	9	3	3	1011	010	81.3
83.3	12	00001011	16	1	11	2	2	1100	001	87.5
83.3	16	00001111	12	1	5	3	3	0111	010	75.0
83.3	16	00001111	12	1	7	2	2	1000	001	83.3
83.3	24	00010111	8	1	3	2	2	0100	001	75.0
83.3	24	00010111	8	1	5	1	1	0101	000	87.5
33.3	30	00011101	24	1	7	8	8	1110	111	66.7
33.3	30	00011101	24	1	9	7	7	1111	110	70.8
33.3	24	00010111	20	1	9	5	5	1101	100	75.0
33.3	24	00010111	20	1	11	4	4	1110	011	80.0
33.3	30	00011101	16	1	7	4	4	1010	011	75.0
33.3	30	00011101	16	1	9	3	3	1011	010	81.3
33.3	32	00011111	15	1	8	3	3	1010	010	80.0
33.3	32	00011111	15	1	10	2	2	1011	001	86.7
33.3	37	00100100	13	1	6	3	3	1000	010	76.9
33.3	37	00100100	13	1	8	2	2	1001	001	84.6
33.3	40	00100111	12	1	5	3	3	0111	010	75.0
33.3	40	00100111	12	1	7	2	2	1000	001	83.3
33.3	48	00101111	10	1	3	3	3	0101	010	70.0
33.3	48	00101111	10	1	5	2	2	0110	001	80.0
33.3	60	00111011	8	1	3	2	2	0100	001	75.0
33.3	60	00111011	8	1	5	1	1	0101	000	87.5

Caution The values in Table 15-24 do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

15.16 Operation of CAN Controller

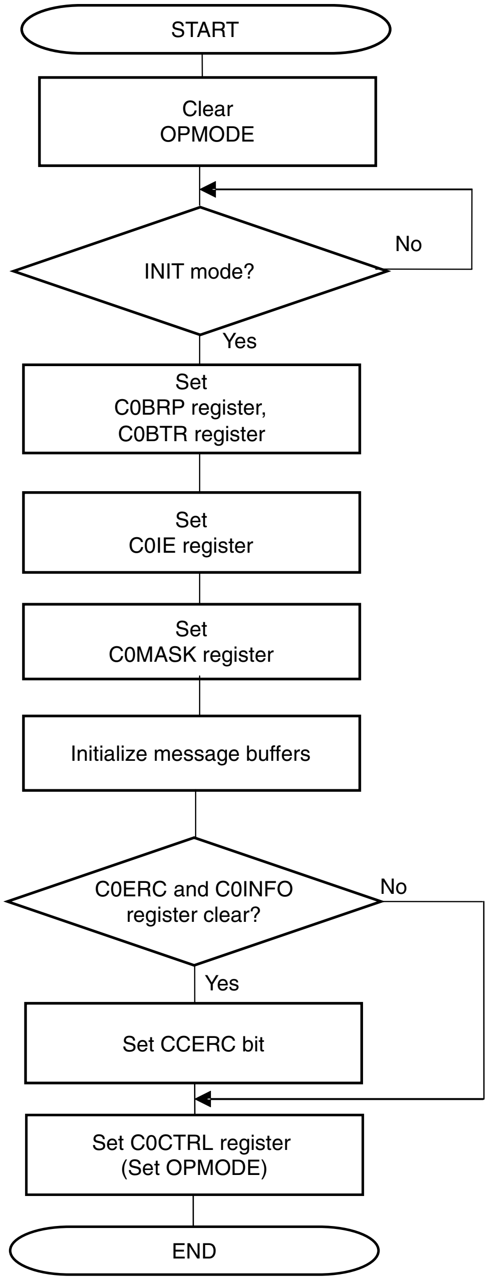
Remark m = 0 to 15

Figure 15-37. Initialization



Remark OPMODE: Normal operation mode, normal operation mode with ABT, receive-only mode, single-shot mode, self-test mode

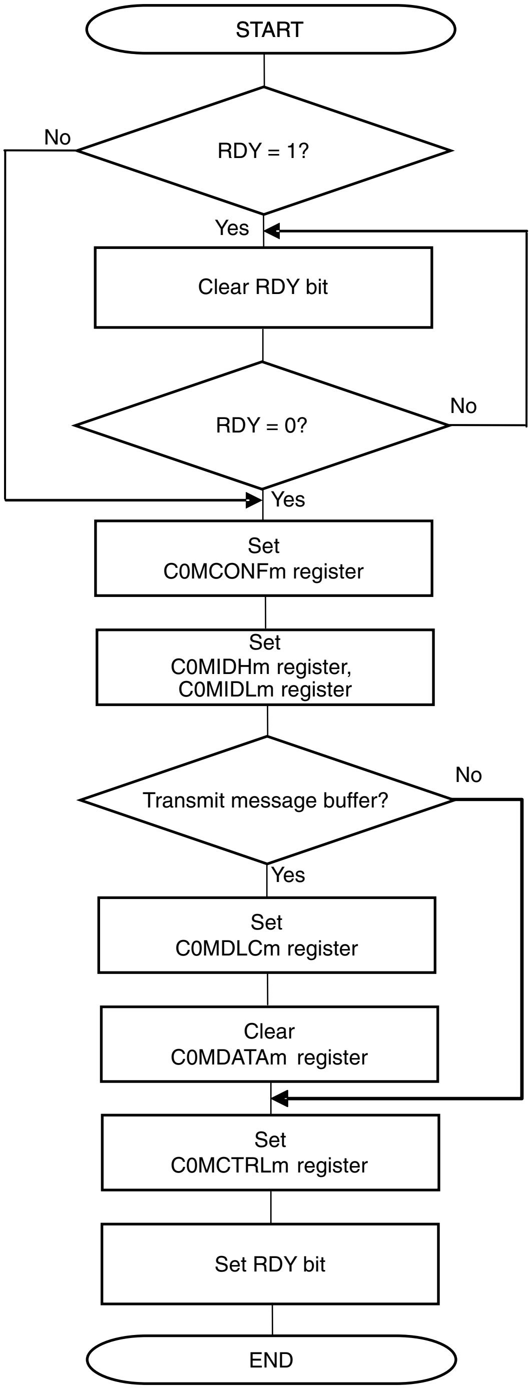
Figure 15-38. Re-initialization



Caution After setting the CAN module to the initialization mode, avoid setting the module to another operation mode immediately after. If it is necessary to immediately set the module to another operation mode, be sure to access registers other than the C0CTRL and C0GMCTRL registers (e.g. set a message buffer).

Remark OPMODE: Normal operation mode, normal operation mode with ABT, receive-only mode, single-shot mode, self-test mode

Figure 15-39. Message Buffer Initialization



Cautions 1. Before a message buffer is initialized, the RDY bit must be cleared.

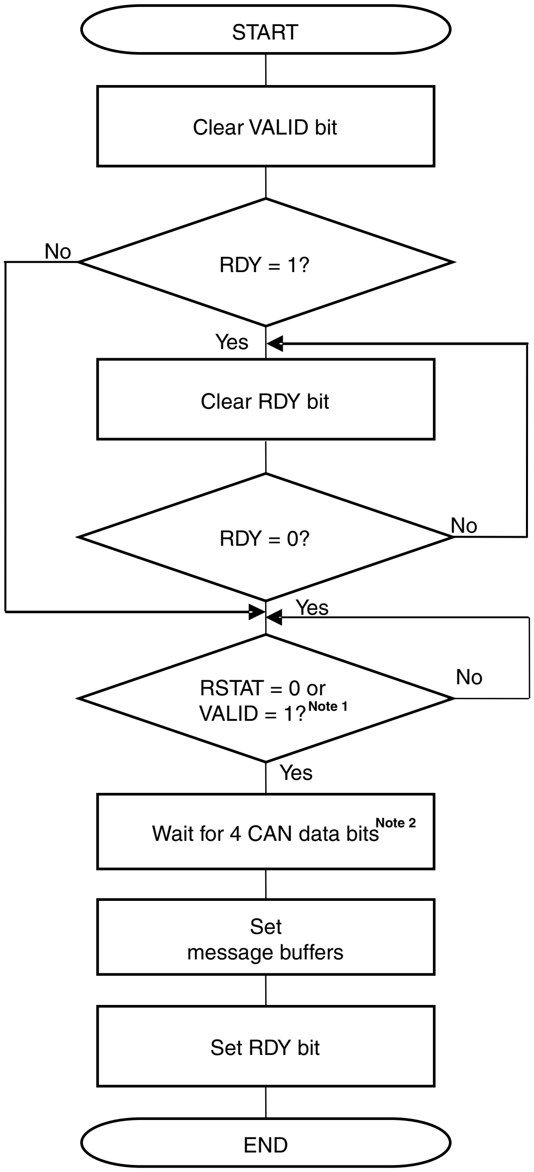
2. Make the following settings for message buffers not used by the application.

- Clear the RDY, TRQ, and DN bits of the C0MCTRLm register to 0.

- Clear the MA0 bit of the C0MCONFm register to 0.

Figure 15-40 shows the processing for a receive message buffer (MT [2:0] bits of C0MCONFm register = 001B to 101B).

Figure 15-40. Message Buffer Redefinition



Notes 1. Confirm that a message is being received because RDY bit must be set after a message is completely received.

2. Avoid message buffer redefinition during store operation of message reception by waiting additional 4 CAN data bits.

Figure 15-41 shows the processing for a transmit message buffer during transmission (MT [2:0] bits of C0MCONFm register = 000B).

Figure 15-41. Message Buffer Redefinition during Transmission

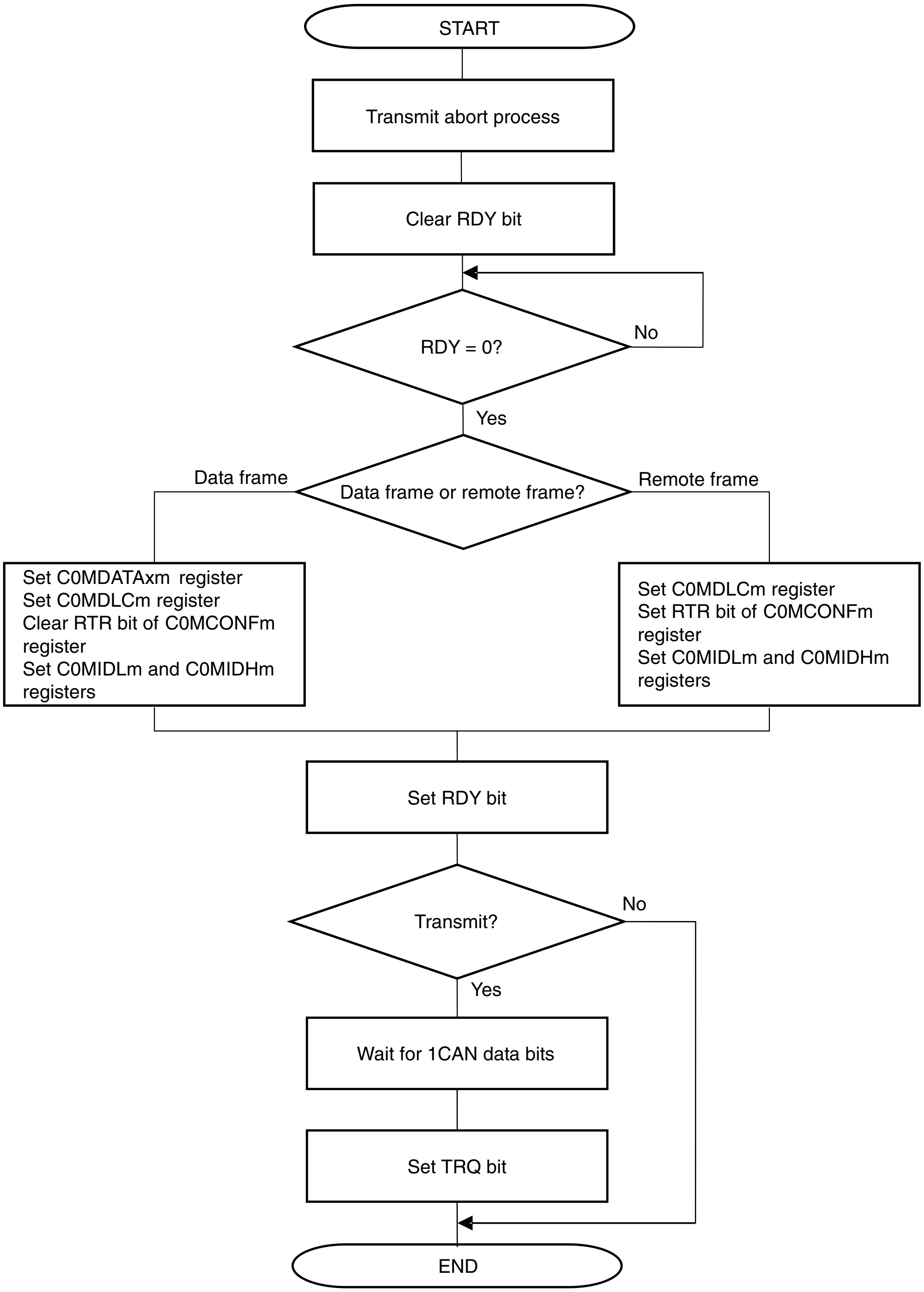
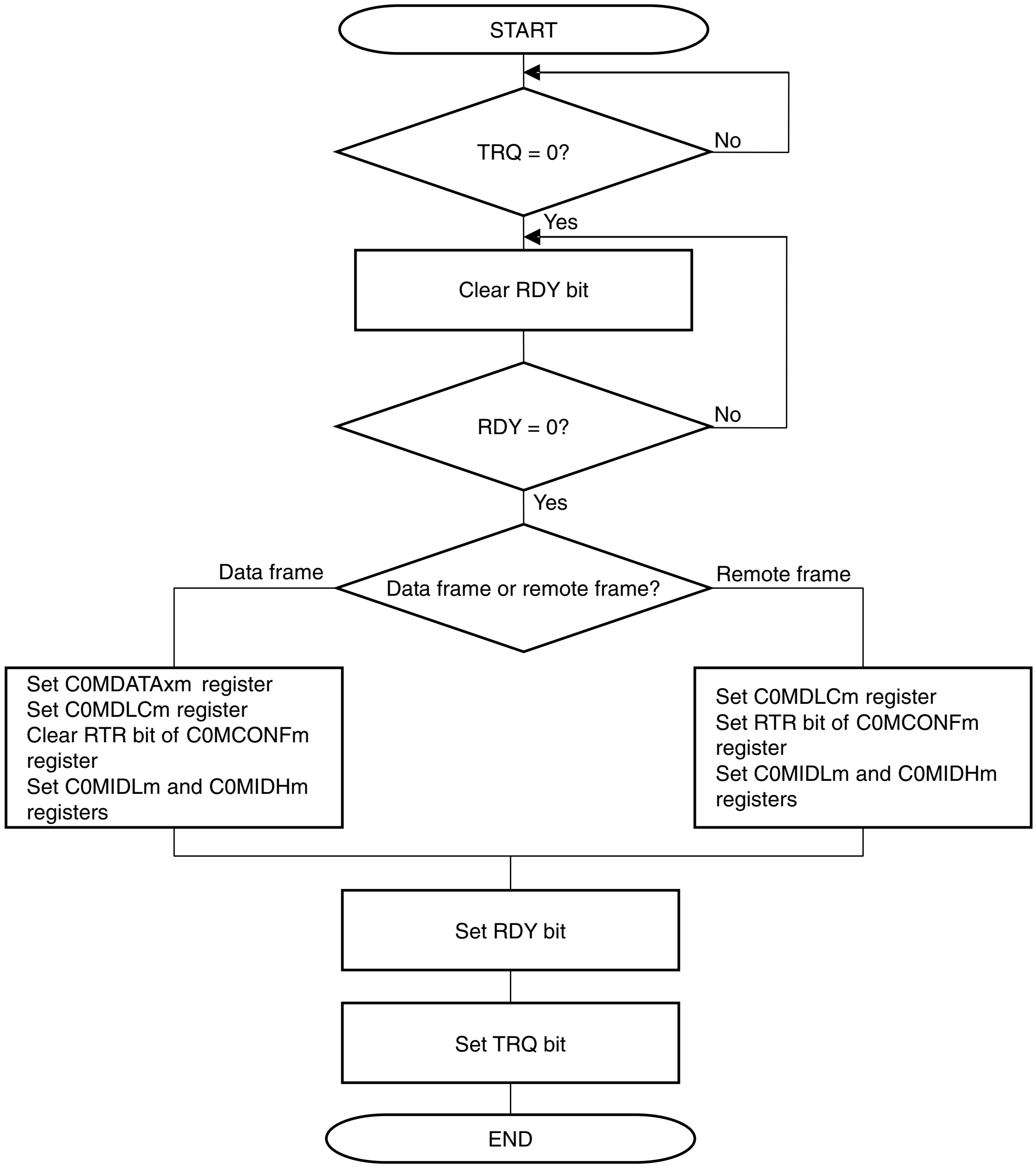


Figure 15-42 shows the processing for a transmit message buffer (MT [2:0] bits of C0MCONFm register = 000B).

Figure 15-42. Message Transmit Processing

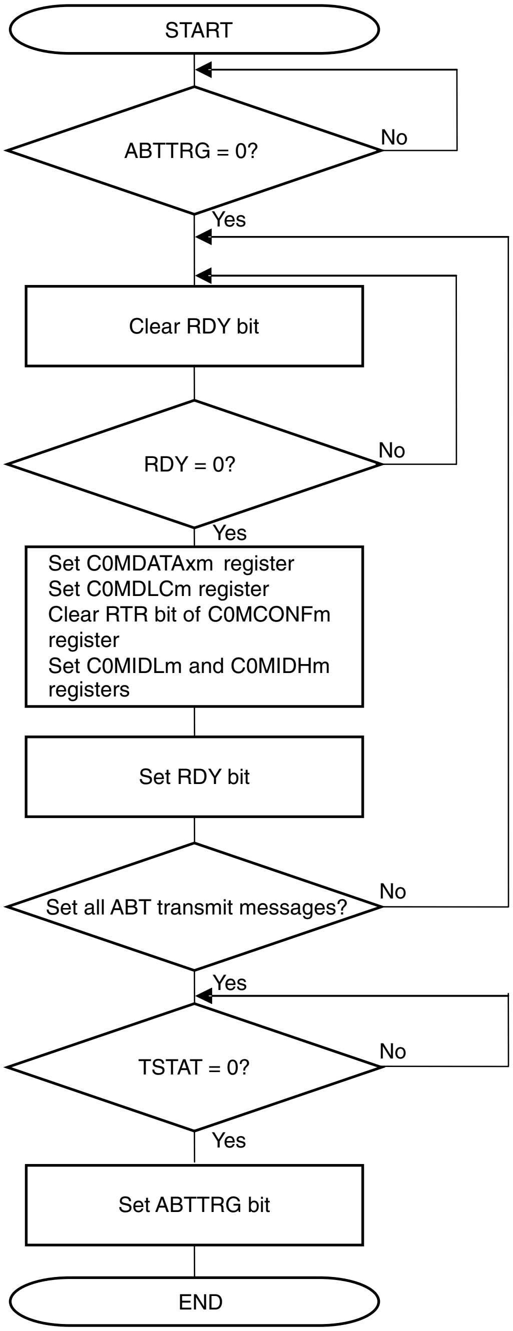

START
TRQ = 0?
No
Yes
Clear RDY bit
RDY = 0?
No
Yes
Data frame
Data frame or remote frame?
Remote frame
Set C0MDATAxm register
Set C0MDLCm register
Clear RTR bit of C0MCONFm register
Set C0MIDLm and C0MIDHm registers
Set C0MDLCm register
Set RTR bit of C0MCONFm register
Set C0MIDLm and C0MIDHm registers
Set RDY bit
Set TRQ bit
END

Cautions 1. The TRQ bit should be set after the RDY bit is set.

2. The RDY bit and TRQ bit should not be set at the same time.

Figure 15-43 shows the processing for a transmit message buffer (MT [2:0] bits of C0MCONFm register = 000B).

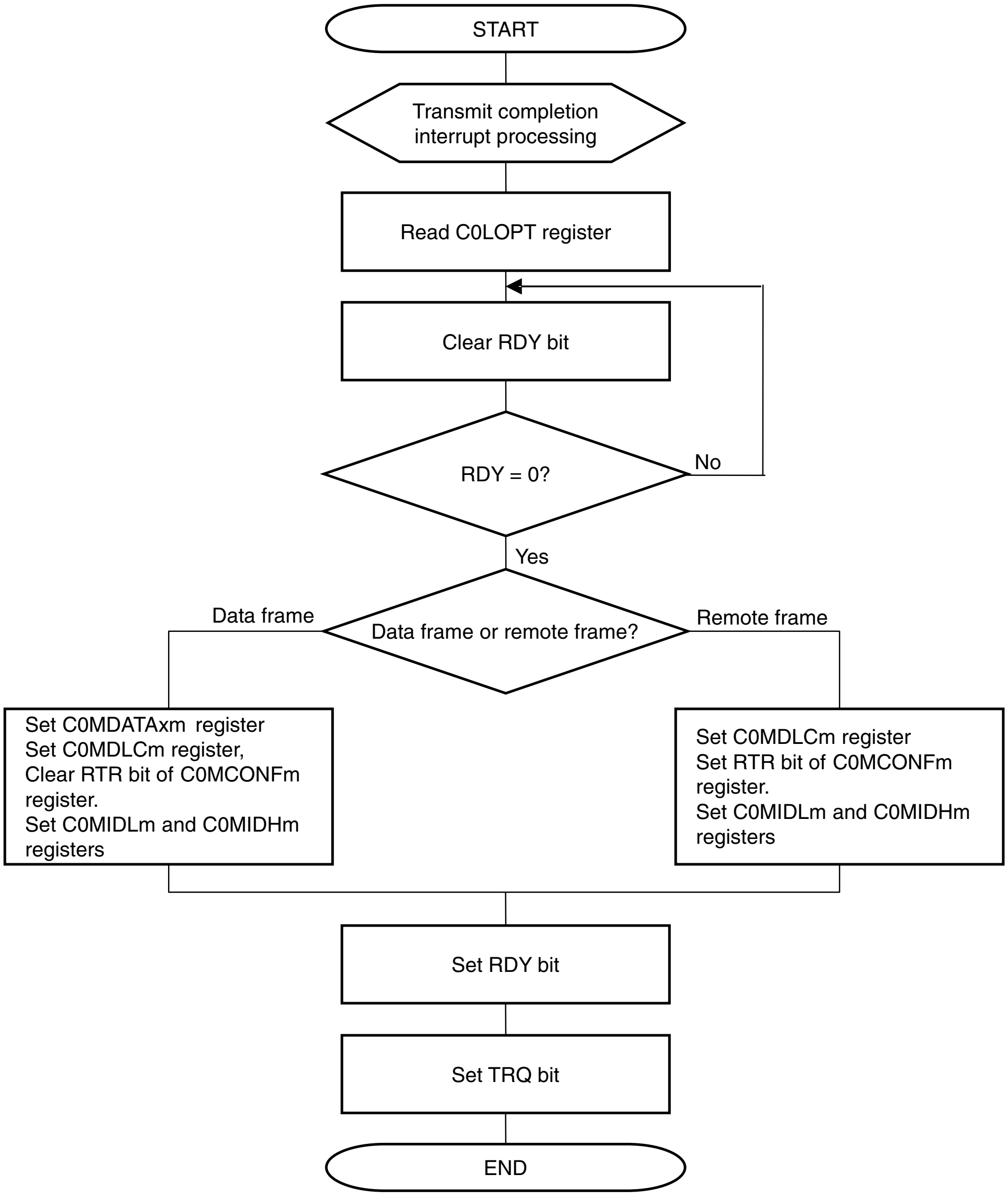
Figure 15-43. ABT Message Transmit Processing



Caution **The ABTTRG bit should be set to 1 after the TSTAT bit is cleared to 0. Checking the TSTAT bit and setting the ABTTRG bit to 1 must be processed continuously.**

Remark This processing (normal operation mode with ABS) can only be applied to message buffers 0 to 7. For message buffers other than the ABT message buffers, refer to **Figure 15-42. Message Transmit Processing**.

Figure 15-44. Transmission via Interrupt (Using C0LOPT register)

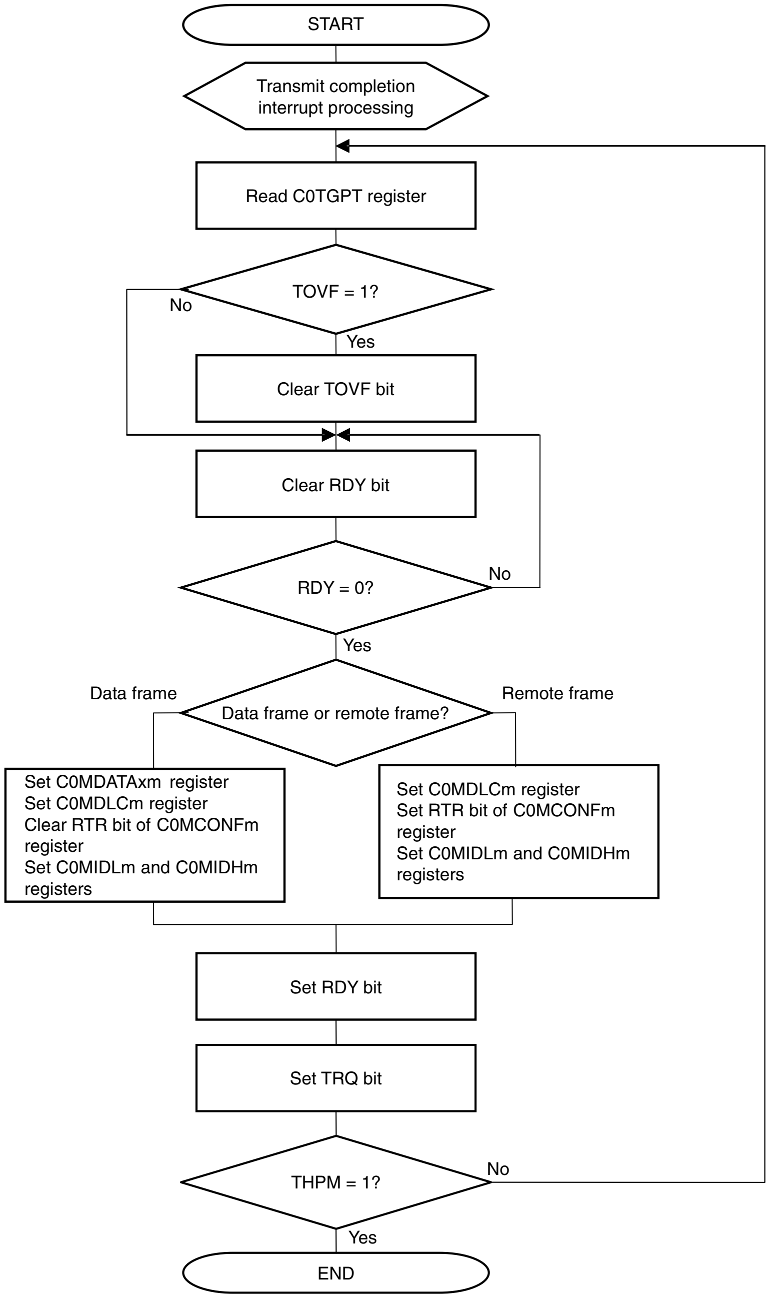

START
Transmit completion interrupt processing
Read C0LOPT register
Clear RDY bit
RDY = 0?
No
Yes
Data frame
Data frame or remote frame?
Remote frame
Set C0MDATAxm register
Set C0MDLCm register,
Clear RTR bit of C0MCONFm register.
Set C0MIDLm and C0MIDHm registers
Set C0MDLCm register
Set RTR bit of C0MCONFm register.
Set C0MIDLm and C0MIDHm registers
Set RDY bit
Set TRQ bit
END

Cautions 1. The TRQ bit should be set after the RDY bit is set.

2. The RDY bit and TRQ bit should not be set at the same time.

Remark Also check the MBON flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as TX history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.
It is recommended to cancel any sleep mode requests, before processing TX interrupts.

Figure 15-45. Transmit via Interrupt (Using C0TGPT register)



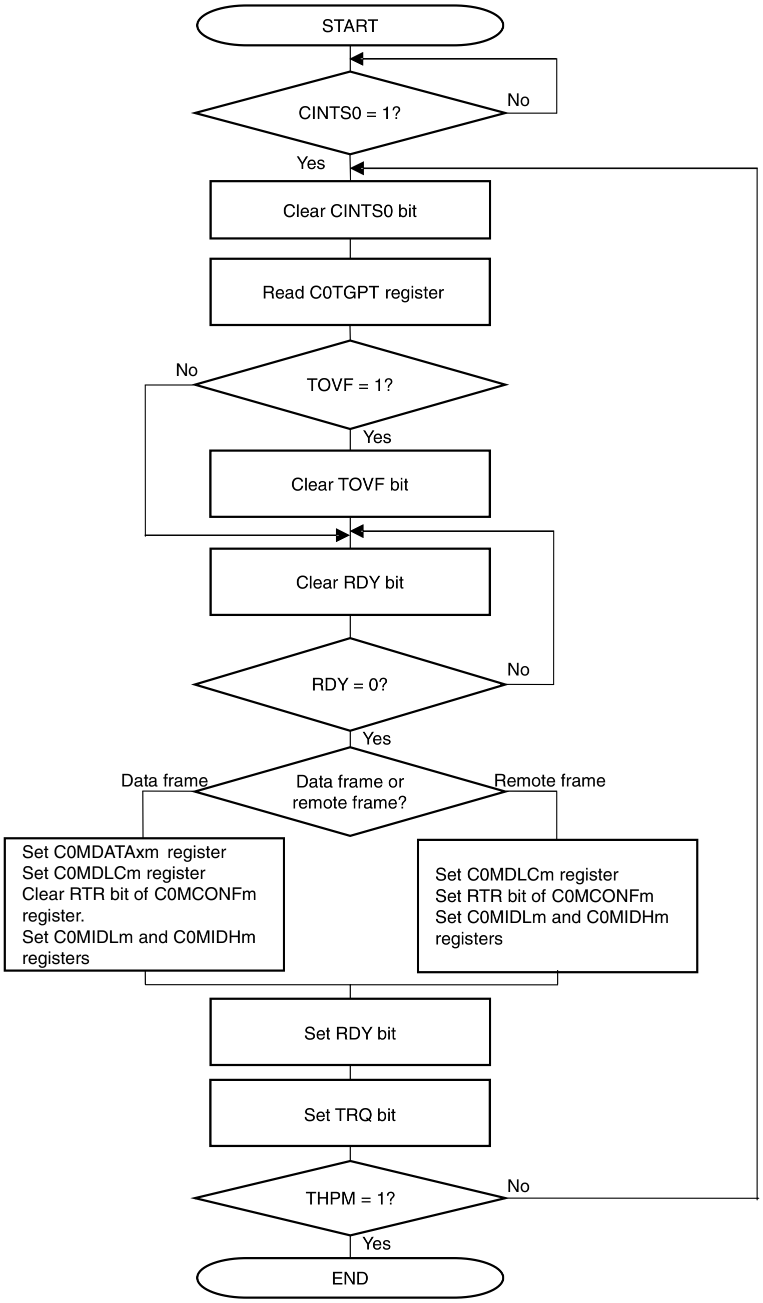
Cautions 1. The TRQ bit should be set after the RDY bit is set.

2. The RDY bit and TRQ bit should not be set at the same time.

Remarks 1. Also check the MBON flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as TX history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again. It is recommended to cancel any sleep mode requests, before processing TX interrupts.

2. If TOVF was set once, the transmit history list is inconsistent. Consider to scan all configured transmit buffers for completed transmissions.

Figure 15-46. Transmission via Software Polling



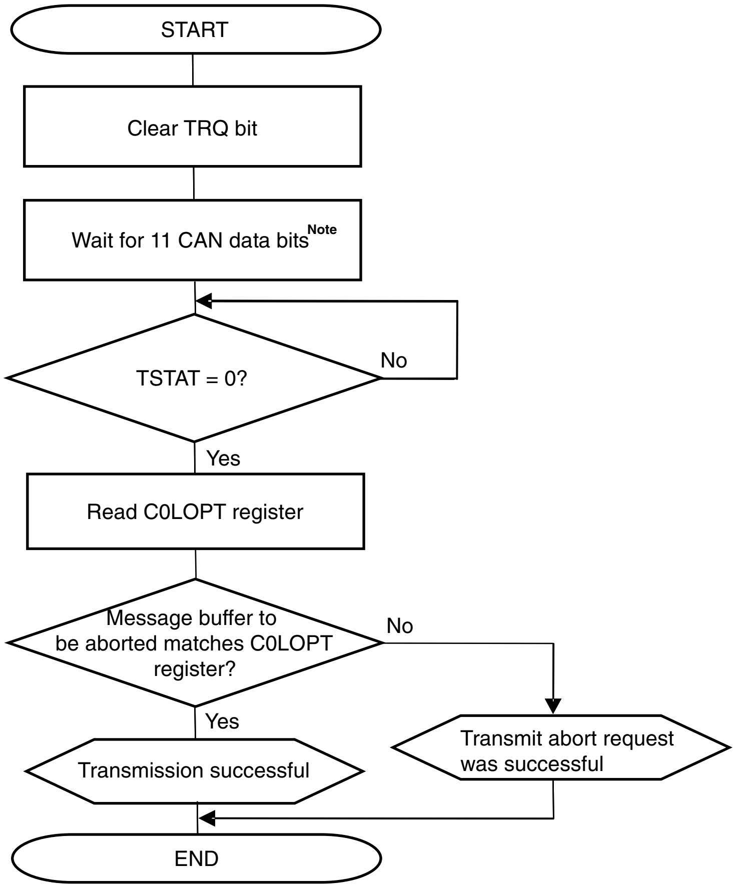
Cautions 1. The TRQ bit should be set after the RDY bit is set.

2. The RDY bit and TRQ bit should not be set at the same time.

Remarks 1. Also check the MBON flag at the beginning and at the end of the polling routine, in order to check the access to the message buffers as well as TX history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.

2. If TOVF was set once, the transmit history list is inconsistent. Consider to scan all configured transmit buffers for completed transmissions.

Figure 15-47. Transmission Abort Processing (Except Normal Operation Mode with ABT)


START
Clear TRQ bit
Wait for 11 CAN data bitsNote
TSTAT = 0?
No
Yes
Read C0LOPT register
Message buffer to be aborted matches C0LOPT register?
No
Yes
Transmission successful
Transmit abort request was successful
END

Note There is a possibility of starting the transmission without being aborted even if TRQ bit is cleared, because the transmission request to protocol layer might already been accepted between 11 bits, total of interframe space (3 bits) and suspend transmission (8 bits).

Cautions 1. Execute transmission request abort processing by clearing the TRQ bit, not the RDY bit.

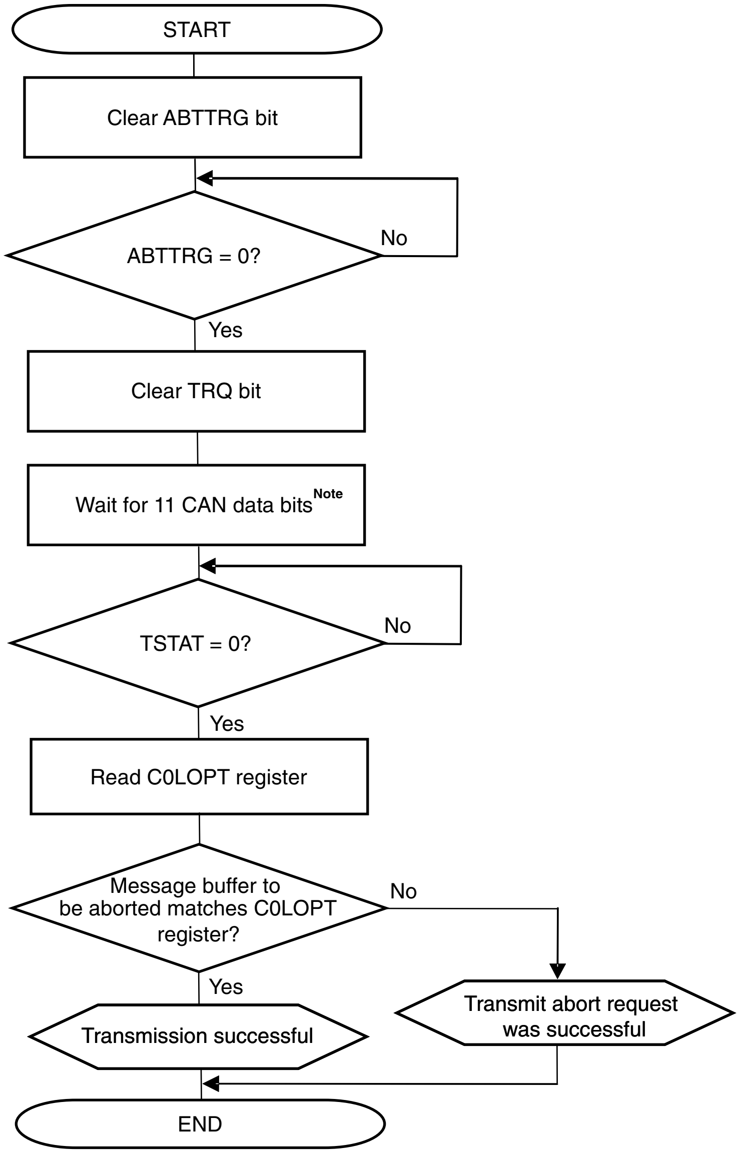
2. Before making a sleep mode transition request, confirm that there is no transmission request left using this processing.

3. The TSTAT bit can be periodically checked by a user application or can be checked after the transmit completion interrupt.

4. Do not execute the new transmission request including in the other message buffers while transmission abort processing is in progress.

5. There is a possibility that contradiction is caused in the judgment whether the transmission abort request was successful when the transmission from the same message buffer is consecutive or only one message buffer is used. In that case, judge it by using the history information etc. that the C0TGPT register indicates.

Figure 15-48. Transmission Abort Processing Except for ABT Transmission (Normal Operation Mode with ABT)


START
Clear ABTTRG bit
ABTTRG = 0?
No
Yes
Clear TRQ bit
Wait for 11 CAN data bits Note
TSTAT = 0?
No
Yes
Read C0LOPT register
Message buffer to be aborted matches C0LOPT register?
No
Yes
Transmission successful
Transmit abort request was successful
END

Note There is a possibility of starting the transmission without being aborted even if TRQ bit is cleared, because the transmission request to protocol layer might already been accepted between 11 bits, total of interframe space (3 bits) and suspend transmission (8 bits).

Cautions 1. Execute transmission request abort processing by clearing the TRQ bit, not the RDY bit.

2. Before making a sleep mode transition request, confirm that there is no transmission request left using this processing.

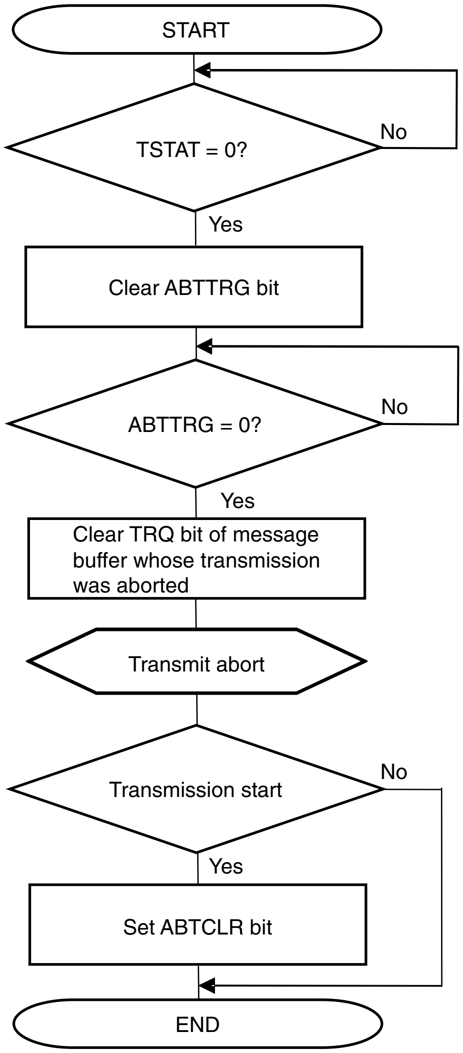
3. The TSTAT bit can be periodically checked by a user application or can be checked after the transmit completion interrupt.

4. Do not execute the new transmission request including in the other message buffers while transmission abort processing is in progress.

Cautions 5 There is a possibility that contradiction is caused in the judgment whether the transmission abort request was successful when the transmission from the same message buffer is consecutive or only one message buffer is used. In that case, judge it by using the history information etc. that the CnTGPT register indicates.

Figure 15-49 shows the processing not to skip resumption of transmitting a message that was stopped when transmission of an ABT message buffer was aborted.

Figure 15-49. ABT Transmission Abort Processing (Normal Operation Mode with ABT)

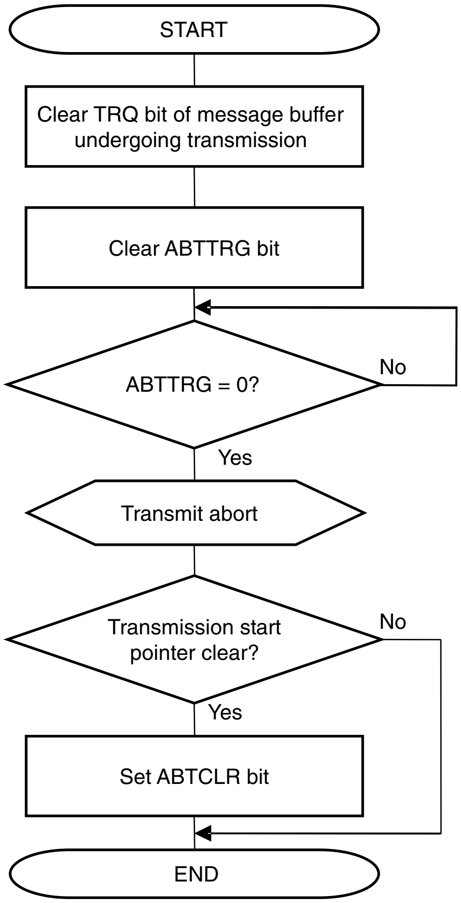


Cautions 1. Do not set any transmission requests while ABT transmission abort processing is in progress.

2. Make a CAN sleep mode/CAN stop mode transition request after ABTTRG bit is cleared (after ABT mode is aborted) following the procedure shown in Figure 15-49 or 15-50. When clearing a transmission request in an area other than the ABT area, follow the procedure shown in Figure 15-47. Transmission Abort Processing (Except Normal Operation Mode with ABT).

Figure 15-50 shows the processing to skip resumption of transmitting a message that was stopped when transmission of an ABT message buffer was aborted.

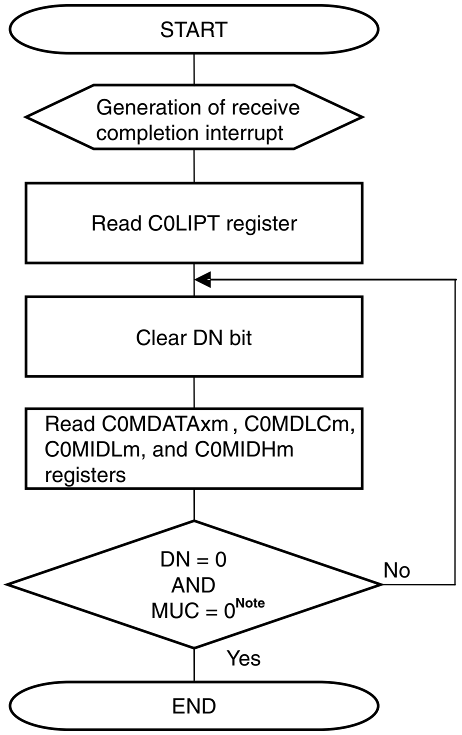
Figure 15-50. ABT Transmission Request Abort Processing (Normal Operation Mode with ABT)


START
Clear TRQ bit of message buffer undergoing transmission
Clear ABTTRG bit
ABTTRG = 0?
No
Yes
Transmit abort
Transmission start pointer clear?
No
Yes
Set ABTCLR bit
END

Cautions 1. Do not set any transmission requests while ABT transmission abort processing is in progress.

2. Make a CAN sleep mode/CAN stop mode request after ABTTRG is cleared (after ABT mode is stopped) following the procedure shown in Figure 15-49 or 15-50. When clearing a transmission request in an area other than the ABT area, follow the procedure shown in Figure 15-47. Transmission Abort Processing (Except Normal Operation Mode with ABT).

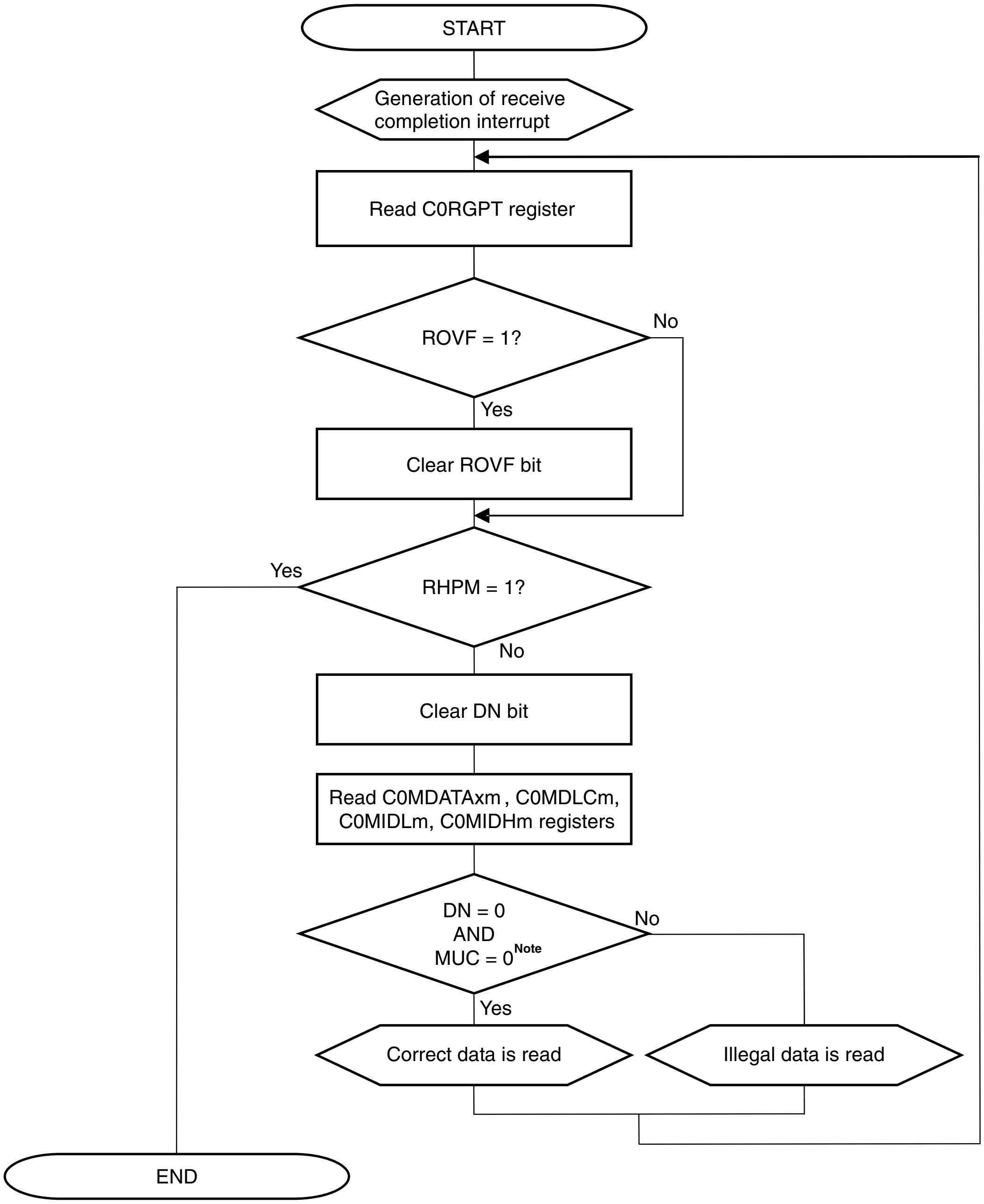
Figure 15-51. Reception via Interrupt (Using C0LIPT Register)


START
Generation of receive completion interrupt
Read C0LIPT register
Clear DN bit
Read C0MDATAxm , C0MDLCm, C0MIDLm, and C0MIDHm registers
DN = 0
AND
MUC = 0Note
No
Yes
END

Note Check the MUC and DN bits using one read access.

Remark Also check the MBON flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as reception history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.
It is recommended to cancel any sleep mode requests, before processing RX interrupts.

Figure 15-52. Reception via Interrupt (Using C0RGPT Register)



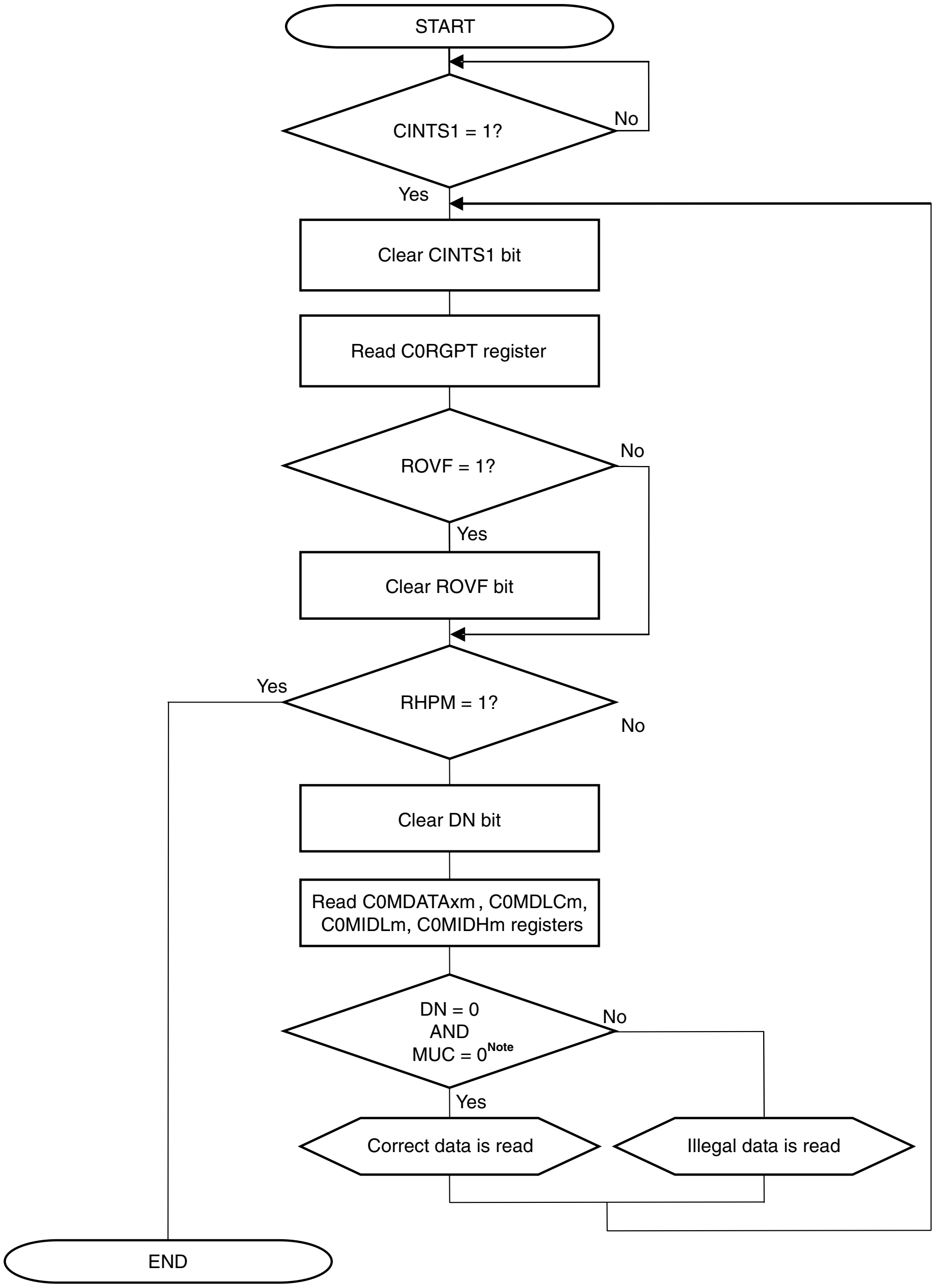
Note Check the MUC and DN bits using one read access.

Remarks 1. Also check the MBON flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as reception history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.

It is recommended to cancel any sleep mode requests, before processing RX interrupts.

2. If ROVF was set once, the receive history list is inconsistent. Consider to scan all configured receive buffers for receptions.

Figure 15-53. Reception via Software Polling



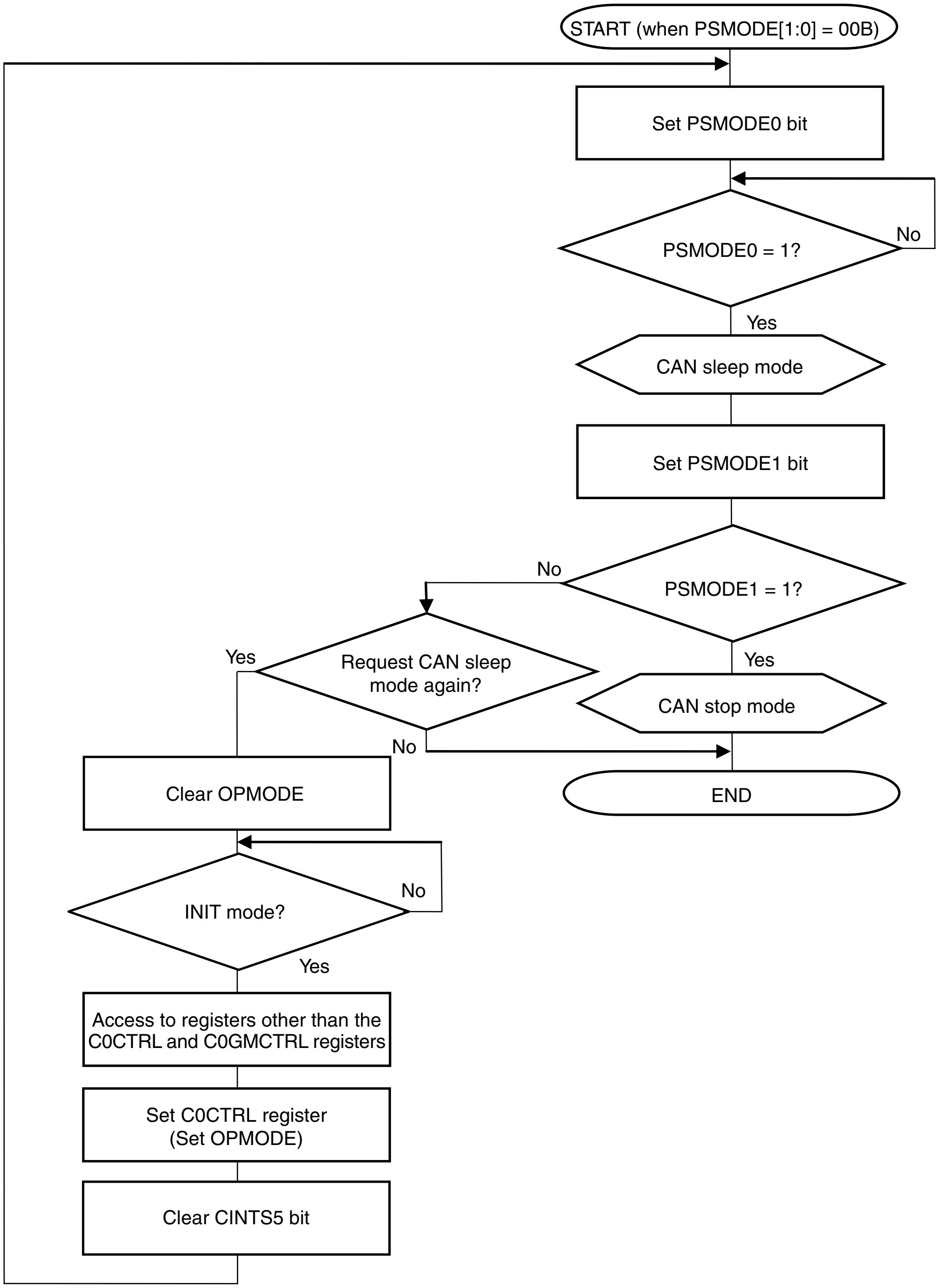
Note Check the MUC and DN bits using one read access.

Remarks 1. Also check the MBON flag at the beginning and at the end of the polling routine, in order to check the access to the message buffers as well as reception history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.

It is recommended to cancel any sleep mode requests, before processing RX interrupts.

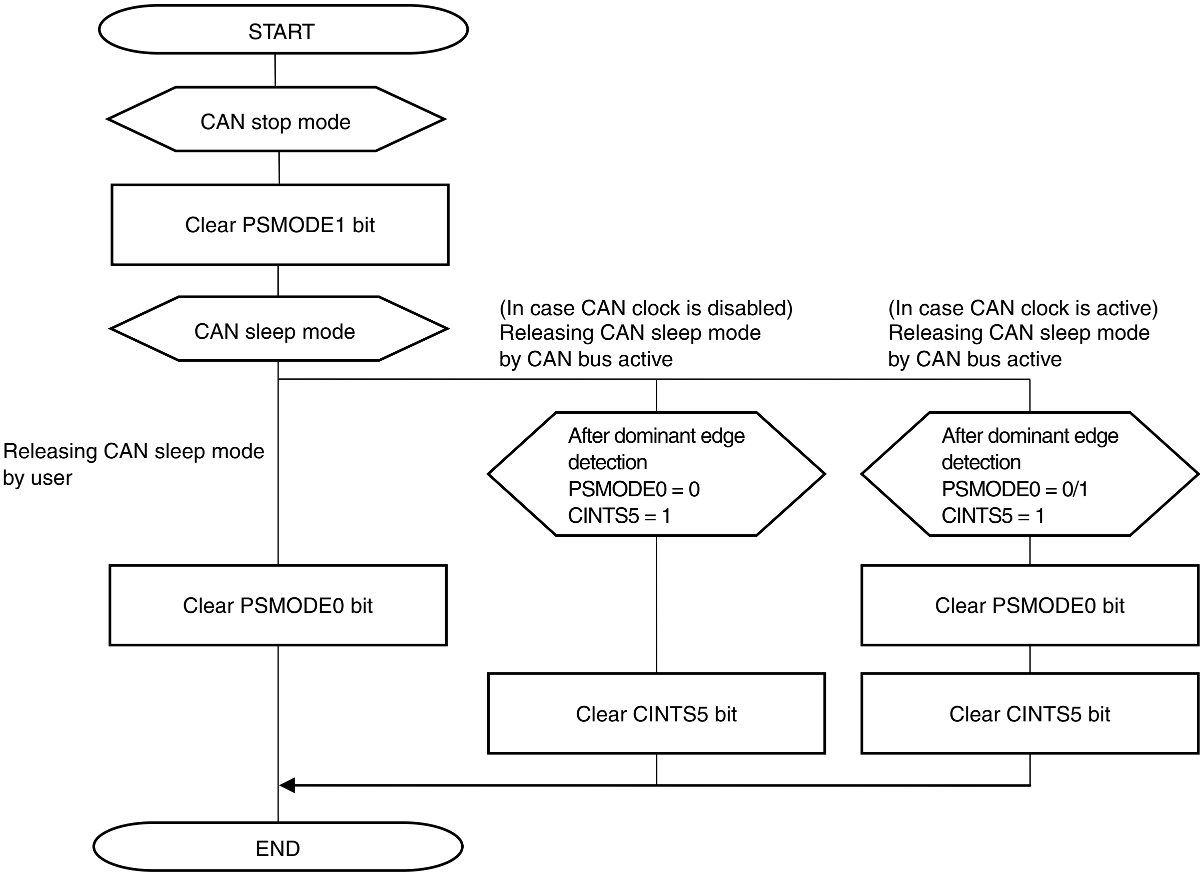
2. If ROVF was set once, the receive history list is inconsistent. Consider to scan all configured receive buffers for receptions.

Figure 15-54. Setting CAN Sleep Mode/Stop Mode


START (when PSMODE[1:0] = 00B)
Set PSMODE0 bit
PSMODE0 = 1?
No
Yes
CAN sleep mode
Set PSMODE1 bit
PSMODE1 = 1?
No
Yes
Request CAN sleep mode again?
Yes
No
CAN stop mode
END
Clear OPMODE
INIT mode?
No
Yes
Access to registers other than the C0CTRL and C0GMCTRL registers
Set C0CTRL register (Set OPMODE)
Clear CINTS5 bit

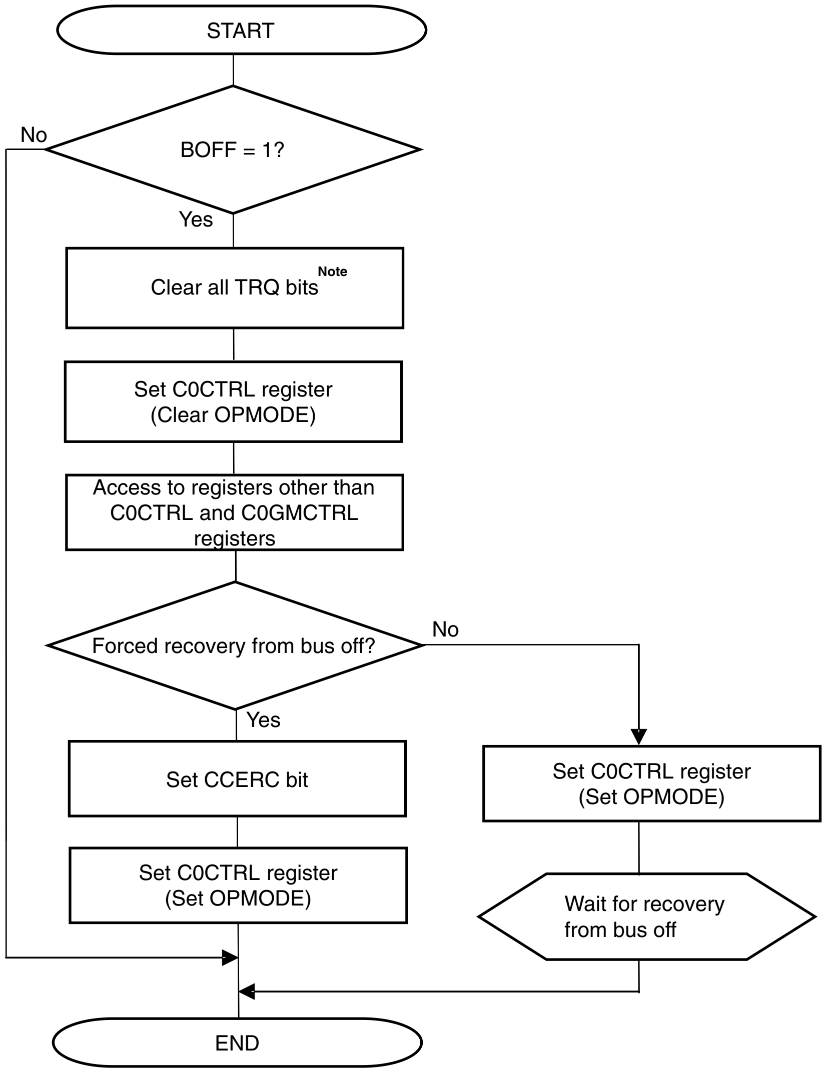
Caution **To abort transmission before making a request for the CAN sleep mode, perform processing according to Figures 15-47 to 15-50.**

Figure 15-55. Clear CAN Sleep/Stop Mode



Remark “In case CAN clock is active”: By means of the CPU standby mode, the CAN module clock has been switched off, and the CAN module is in sleep mode.

Figure 15-56. Bus-Off Recovery (Expect Normal Operation Mode with ABT)



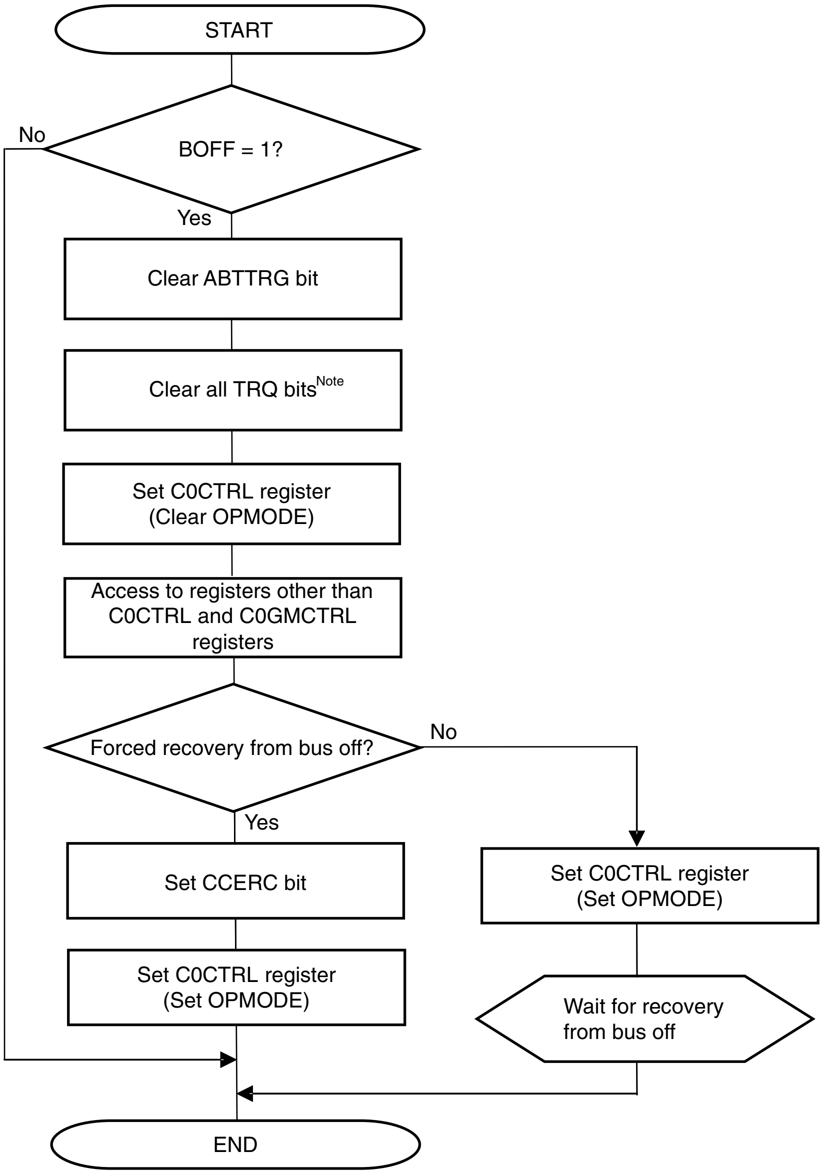
Note Clear all TRQ bits when re-initialization of message buffer is executed by clearing RDY bit before bus-off recovery sequence is started.

Caution When the transmission from the initialization mode to any operation modes is requested to execute bus-off recovery sequence again in the bus-off recovery sequence, reception error counter is cleared.

Therefore it is necessary to detect 11 consecutive recessive-level bits 128 times on the bus again.

Remark OPMODE: Normal operation mode, normal operation mode with ABT, receive-only mode, single-shot mode, self-test mode

Figure 15-57. Bus-Off Recovery (Normal Operation Mode with ABT)


START
BOFF = 1?
No
Yes
Clear ABTTRG bit
Clear all TRQ bitsNote
Set C0CTRL register (Clear OPMODE)
Access to registers other than C0CTRL and C0GMCTRL registers
Forced recovery from bus off?
No
Yes
Set CCERC bit
Set C0CTRL register (Set OPMODE)
Set C0CTRL register (Set OPMODE)
Wait for recovery from bus off
END

Note Clear all TRQ bits when re-initialization of message buffer is executed by clearing RDY bit before bus-off recovery sequence is started.

Caution When the transmission from the initialization mode to any operation modes is requested to execute bus-off recovery sequence again in the bus-off recovery sequence, reception error counter is cleared.

Therefore it is necessary to detect 11 consecutive recessive-level bits 128 times on the bus again.

Remark OPMODE: Normal operation mode, normal operation mode with ABT, receive-only mode, single-shot mode, self-test mode

Figure 15-58. Normal Shutdown Process

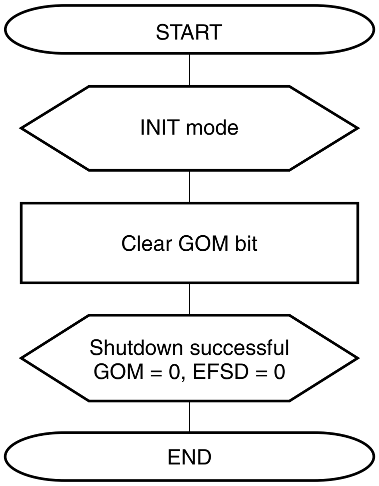
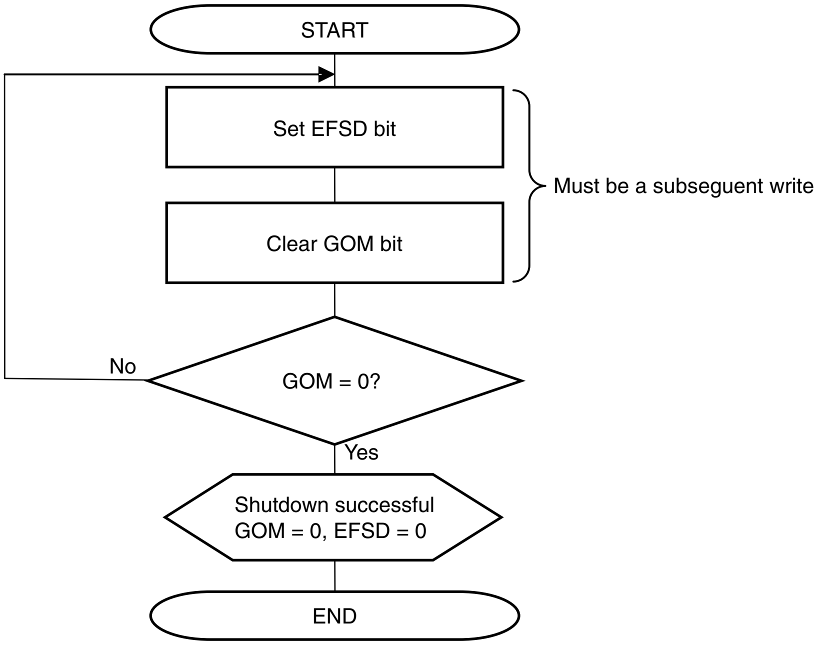


Figure 15-59. Forced Shutdown Process



Caution Do not read- or write-access any registers by software between setting the EFSD bit and clearing the GOM bit.

Figure 15-60. Error Handling

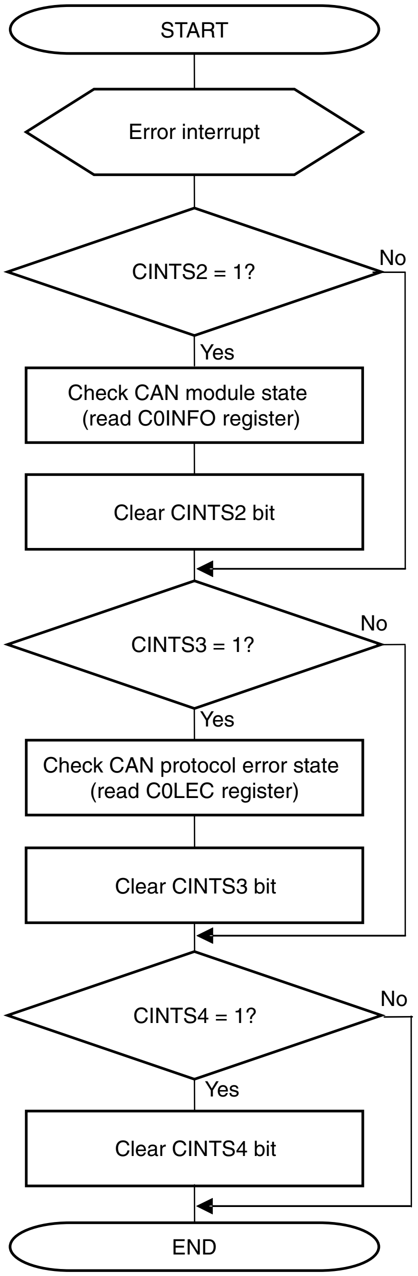
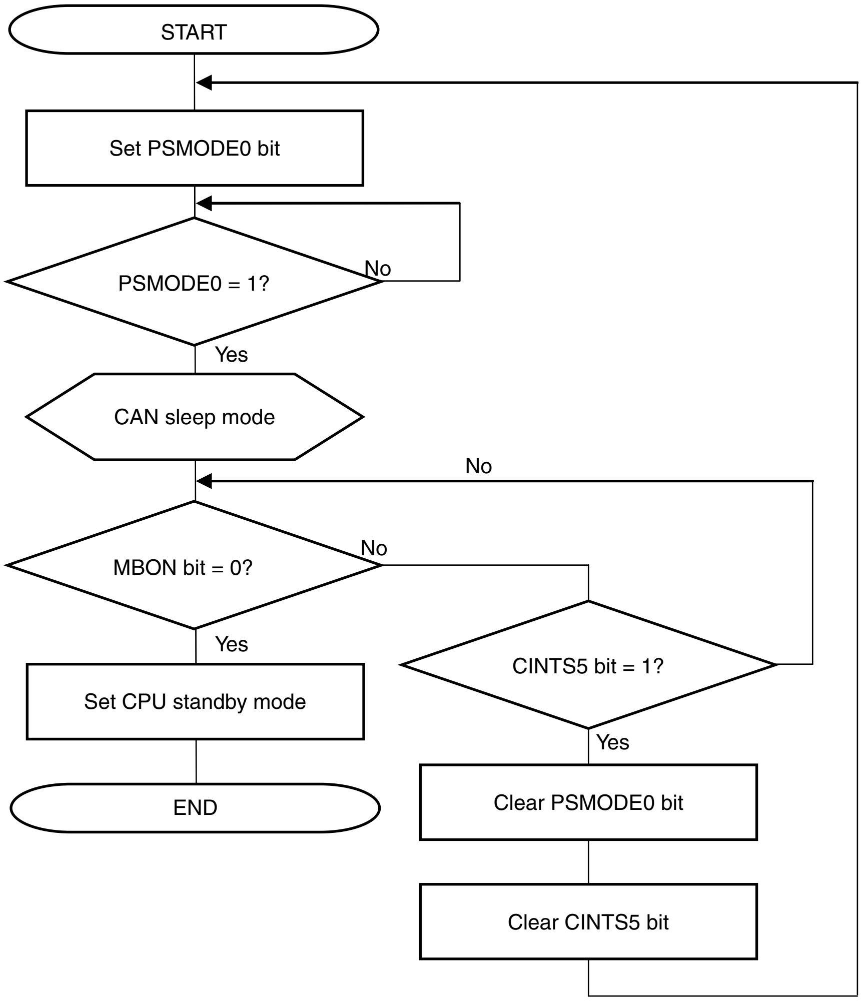
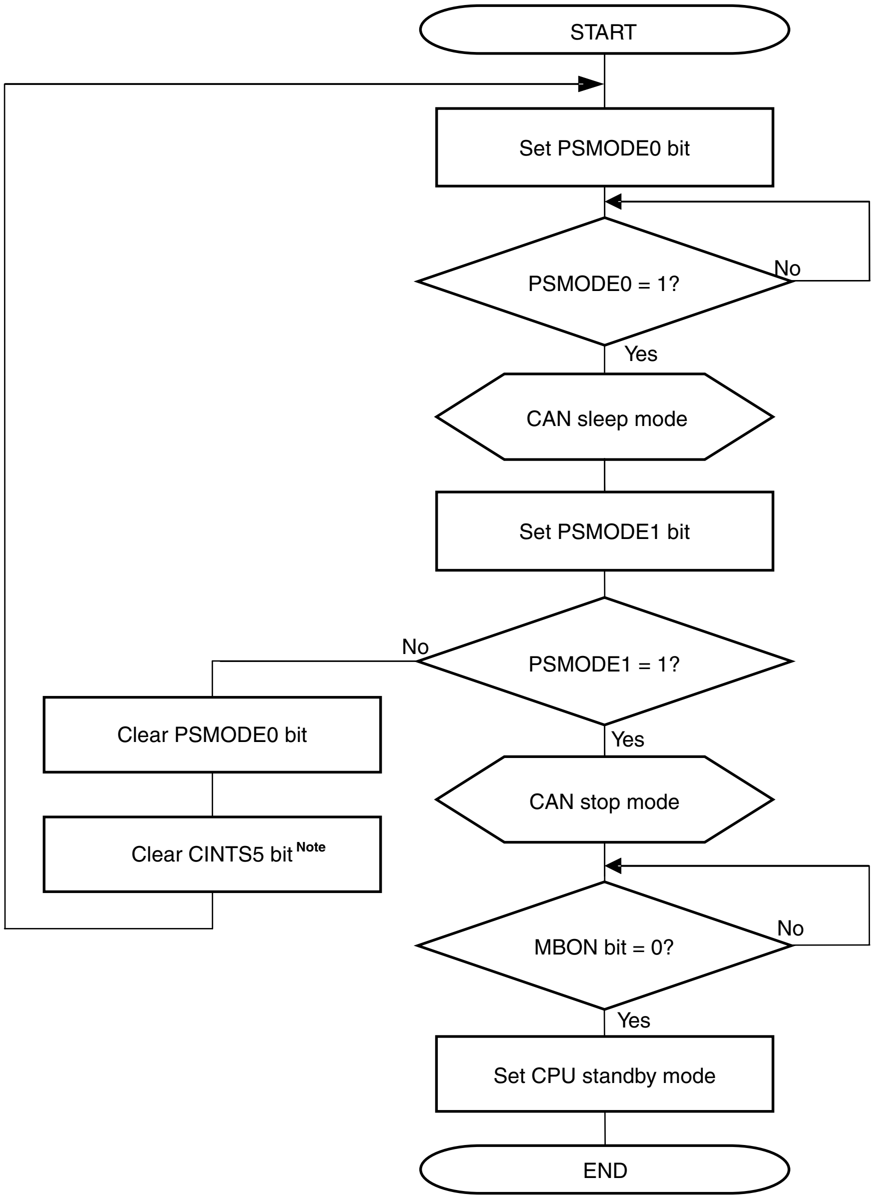


Figure 15-61. Setting CPU Standby (from CAN Sleep Mode)



Caution Before the CPU is set in the CPU standby mode, please check the CAN sleep mode or not. However, after check of the CAN sleep mode, until the CPU is set in the CPU standby mode, the CAN sleep mode may be cancelled by wakeup from CAN bus.

Figure 15-62. Setting CPU Standby (from CAN Stop Mode)



Note During wakeup interrupts

Caution The CAN stop mode can only be released by writing 01B to the PSMODE[1:0] bit of the C0CTRL register and not by a change in the CAN bus state.

CHAPTER 16 STEPPER MOTOR CONTROLLER/DRIVER

The Stepper Motor Controller/Driver module is comprised of four drivers (k = 1 to 4) for external 360° type meters or for bipolar and unipolar stepper motors.

16.1 Overview

The Stepper Motor Controller/Driver module generates pulse width modulated (PWM) output signals. Each driver generates up to four output signals.

Features summary

The generated output signals have the following features:

- Pulse width of 8 bits precision
- 1-bit addition function enables an average pulse width precision of 1/2 bit, resulting in a pseudo 9-bit precision
- PWM frequency up to 20 kHz
- Automatic PWM phase shift for reducing fluctuation on power supply and for reducing the susceptibility to electromagnetic interference
- Zero Point Detection (ZPD) function

16.1.1 Driver overview

A stepper motor is driven by PWM signals. The PWM signals are generated by comparing the contents of compare registers with the actual value of a free running up counter. The Stepper Motor Controller/Driver module contains one counter and assigned compare registers and control registers.

Figure 16-1 shows the main components of the Stepper Motor Controller/Driver 0. The Stepper Motor Controller/Driver 0 includes a free running up counter (CNT0). The counter is controlled by a timer mode control register (MCNTC0). Each of the four drivers consists of two compare registers, MCMPk0 and MCMPk1, respectively. Their contents define the pulse widths for the sine and the cosine side of the meters. The MCMPk0/MCMPk1 registers comprise a master-slave register combination. This allows to re-write the master register while the slave register is currently used for comparison with the counter CNT0.

The compare control register MCMPCk defines whether or not enhanced pulse width precision by one-bit addition is enabled, and it routes the output signals to the corresponding output pins (SMk1 to SMk4).

16.1.2 ZPD introduction

<R> Zero Point Detection (ZPD) enables calibration of meter needle without additional pins.

When needle reaches to the zero point of meter, an induced voltage would be generated due to back EMF effect. ZPD function has the circuit to detect the induced voltage which its reference voltage could be configured in advance. Each of the four drivers has one ZPD circuit.

For reliable results of the ZPD circuit, digital noise removal is applied.

Details on the internal reference voltage can be found in the Electrical Target Specification.

16.1.3 ZPD input pins

When Zero Point detection is enabled and the MCMPCk.TWIN bit is set, every driver can be used for calibration. Dedicated pins of each driver are then used as input pins to allow for calibration of the connected meter.

- At Stepper Motor Controller/Driver, the SMk4 pin (k = 1 to 4) can be used as ZPD input.

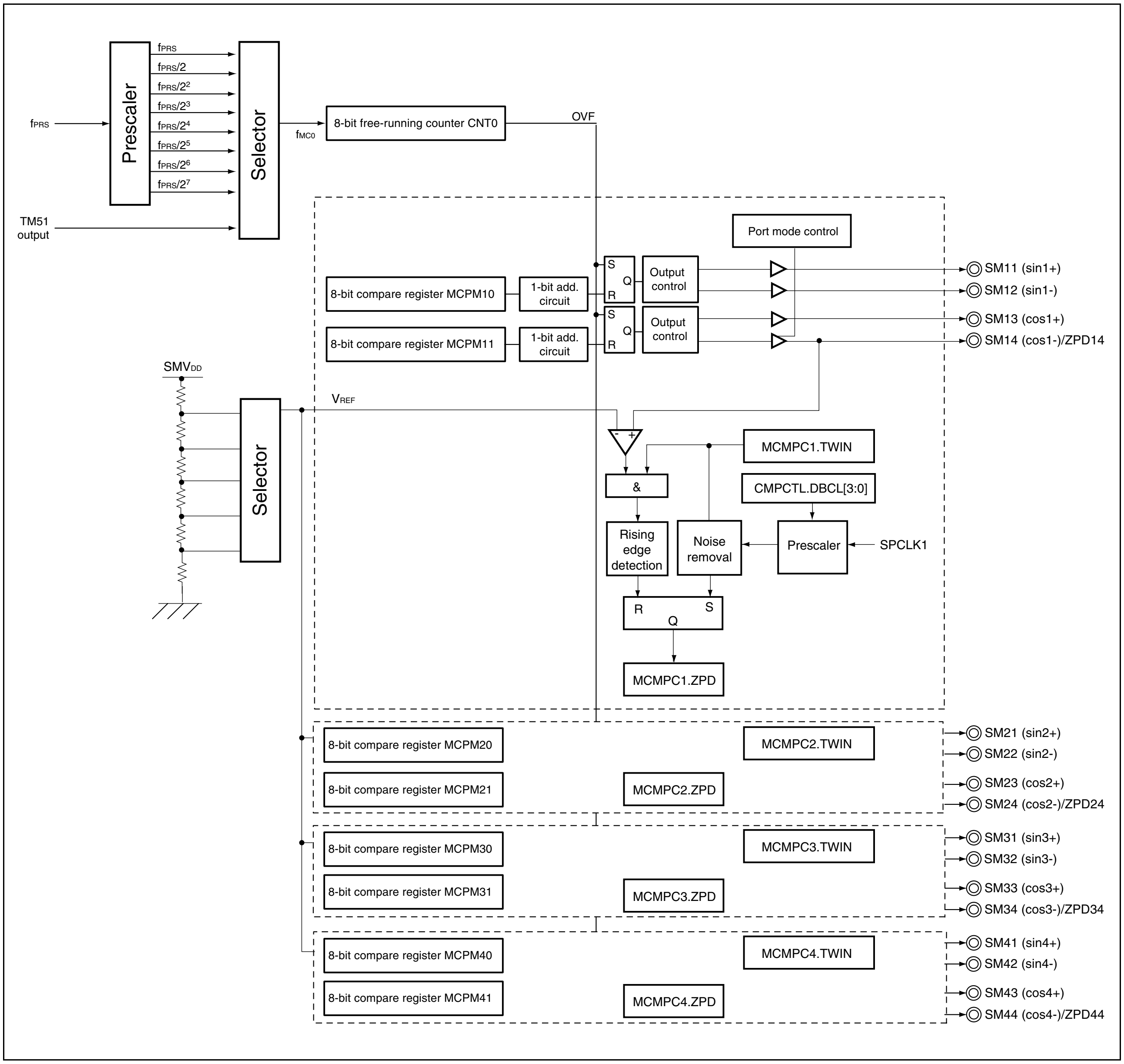
Note that these pins have to be configured as input pins.

The result of a voltage comparison is reflected in the MCMPCk.ZPD bit.
Properties of the ZPD can be set in the ZPD flag detection clock setting register CMPCTL.
Figure 16-1 shows the ZPD circuit of the Stepper Motor Controller/Driver.

Figure 16-1. Stepper Motor Controller/Driver Block Diagram

<R>

<R>



The external signals are listed in the following table.

Table 16-1. Stepper Motor Controller/Driver External Connections

Signal name	I/O	Active level	Reset level	Pins	Function
SM[1:4]1	O	–	L	SM11 to SM41	Driver signal, sine side (+)
SM[1:4]2	O	–	L	SM12 to SM42	Driver signal, sine side (–)
SM[1:4]3	O	–	L	SM13 to SM43	Driver signal, cosine side (+)
SM[1:4]4	O	–	L	SM14 to SM44	Driver signal, cosine side (–)

16.2 Stepper Motor Controller/Driver Registers

The Stepper Motor Controller/Driver is controlled and operated by means of the following registers:

Table 16-2. Stepper Motor Controller/Driver Registers Overview

Register name	Shortcut
Timer mode control registers	MCNTC0
Compare registers	MCMPk0 (k = 1 to 4)
	MCMPk1 (k = 1 to 4)
	MCMPkHW (k = 1 to 4)
Compare control registers	MCMPCk (k = 1 to 4)
Stepper motor port mode control register	SMPC
ZPD detection voltage setting registers	ZPDS0, ZPDS1
ZPD flag detection clock setting register	CMPCTL
ZPD operational control register	ZPDEN

(1) Timer mode control register (MCNTC0)

The 8-bit MCNTC0 register controls the operation of the free running up counters CNT0.

These registers can be read/written in 8-bit or 1-bit units.

This register is cleared by any reset.

Figure 16-2. Format of Timer Mode Control Register (MCNTC0) (1/2)

Address: F9C0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MCNTC0	CAE	0	FULL	PCE	PCS	SMCL2	SMCL1	SMCL0

CAE	Stepper Motor Controller/Driver control
0	Stepper Motor Controller/Driver operation is disabled.
1	Stepper Motor Controller/Driver operation is enabled.

FULL	Count range of the timer counter
0	Count range from 01H to FFH
1	Count range from 00H to FFH
The initial start value is 00H in both cases. For the impact of this bit on duty factor and PWM cycle time, see also **16.3.1 (3) Duty factor**.	

PCE	Timer operation control
0	Timer counter is stopped.
1	Timer counter is enabled.

PCS	Timer count clock
0	Count clock specified by SMCL2 to SMCL0
1	Rising edge of TM51 output as external clock

Figure 16-2. Format of Timer Mode Control Register (MCNTC0) (2/2)

SMCL2	SMCL1	SMCL0	Selected timer count clock
0	0	0	f_{PRS}
0	0	1	$f_{PRS}/2$
0	1	0	$f_{PRS}/2^2$
0	1	1	$f_{PRS}/2^3$
1	0	0	$f_{PRS}/2^4$
1	0	1	$f_{PRS}/2^5$
1	1	0	$f_{PRS}/2^6$
1	1	1	$f_{PRS}/2^7$
Sets the timer count clock for the timer counter			

Caution Bit 6 must be 0.

Power save mode preparation

Before entering any power save mode the Stepper-C/D must be shut down in advance in order to minimize power consumption.

Apply following sequence to shut down the Stepper-C/D:

1. Stop the counter CNT0 by setting MCNTC0.PCE = 0.
2. Disable the Stepper-C/D operation by setting MCNTC0.CAE = 0.

Remark Note that the MCNTC0.PCE and MCNTC0.CAE bits must not be cleared to 0 by a single write instruction. Perform two write instructions as shown above.

(2) Compare registers for sine side (MCMPk0) (k = 1 to 4)

The 8-bit MCMPk0 registers hold the values that define the PWM pulse width for the sine side of the connected meters.

The contents of the registers are continuously compared to the timer counter value:

- Registers MCMP10 to MCMP40 are compared to CNT0.

When the register contents match the timer counter contents, a match signal is generated. Thus a PWM pulse with a pulse width corresponding to the MCMPk0 register contents is output to the sine side of the connected meter.

These registers can be read/written in 8-bit units.

This register is cleared by any reset.

Address: F9C2H, F9C4H, F9C6H, F9C8H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MCMPk0	sine DATA							

Remarks 1. New data must only be written to registers MCMPk0 if the corresponding bit MCMPCk.TEN = 0.

2. Don't write to the compare register MCMPk0, until the corresponding bit MCMPCk.TEN has been reset to 0 automatically.

3. To enable master-to-slave register copy upon next CNTm overflow set MCMPCk.TEN = 1.

(3) Compare registers for cosine side (MCMPk1) (k = 1 to 4)

The 8-bit MCMPk1 registers hold the values that define the PWM pulse width for the cosine side of the connected meters.

The contents of the registers are continuously compared to the timer counter value:

- Registers MCMP11 to MCMP41 are compared to CNT0.

<R> When the register contents match the timer counter contents, a match signal is generated. Thus a PWM pulse with a pulse width corresponding to the MCMPk1 register contents is output to the cosine side of the connected meter.

These registers can be read/written in 8-bit units.

This register is cleared by any reset.

Address: F9C3H, F9C5H, F9C7H, F9C9H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MCMPk1	cosine DATA							

Remarks 1. New data must only be written to registers MCMPk1 if the corresponding bit MCMPCk.TEN = 0.

2. Don't write to the compare register MCMPk1, until the corresponding bit MCMPCk.TEN has been reset to 0 automatically.

3. To enable master-to-slave register copy upon next CNTm overflow set MCMPCk.TEN = 1.

(4) Combined compare registers (MCMPkHW) (k = 1 to 4)

The 16-bit MCMPkHW registers combine the sine and cosine registers MCMPk0 and MCMPk1. Via these registers it is possible to read or write the contents of MCMPk0 and MCMPk1 in a single instruction.

These registers can be read/written in 16-bit units.

This register is cleared by any reset.

Address: F9C2H, F9C4H, F9C6H, F9C8H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCMPkHW	cosine DATA								sine DATA							

<R> **Remarks 1.** New data must only be written to registers MCMPkHW if the corresponding bit MCMPCk.TEN = 0.

2. Don't write to the compare register MCMPkHW, until the corresponding bit MCMPCk.TEN has been reset to 0 automatically.

3. To enable master-to-slave register copy upon next CNTm overflow set MCMPCk.TEN = 1.

(5) Compare control registers (MCMPCk) (k = 1 to 4)

The 8-bit MCMPCk registers control the operation of the corresponding compare registers and the output direction of the PWM pin.

These registers can be read/written in 8-bit or 1-bit units.

This register is cleared by any reset.

Figure 16-3. Format of Compare Control Registers (MCMPCk) (1/2)

Address: F9CAH, F9CCH, F9CEH, F9F8H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MCMPCk	AOUT	TWIN	ZPD[Note]	TEN	ADB1	ADB0	DIR1	DIR0

Note This bit may be written, but writing is ignored.

AOUT	Output pins for sine and cosine signals
0	The PWM signals for sine and cosine side are output to those pins that are selected by bits DIR0 and DIR1. At all other pins, the output signal is 0 (SMV_{SS} level).
1	The PWM signal for the sine side is output to pins SMk1 and SMk2. The PWM signal for the cosine side is output to pins SMk3 and SMk4.

TWIN	0-point detection timing window
0	Disable writing to ZPD bit from the comparator (No 0-point detection)
1	Enable writing to ZPD bit from the comparator (0-point detection)

ZPD	Induced voltage detection bit for 0-point detection (Read only)
0	No induced voltage detection (0-point detection)
1	Induced voltage detection (No 0-point detection)

TEN	Transfer enable control bit
0	MCMPk0/MCMPk1 master-to-slave register copy is disabled. New data can be written to compare registers MCMPk0 or MCMPk1.
1	MCMPk0/MCMPk1 master-to-slave register copy is enabled. The copy process will take place when CNT0 overflows. Don't write to compare registers MCMPk0 or MCMPk1 while MCMPCk.TEN = 1.
Remark This bit functions as a control bit and status flag. It is automatically reset to zero upon the next timer counter overflow.	

ADB1	1-bit addition function for cosine side
0	No 1-bit addition to PWM signal
1	1-bit addition to PWM signal

ADB0	1-bit addition function for sine side
0	No 1-bit addition to PWM signal
1	1-bit addition to PWM signal

Figure 16-3. Format of Compare Control Registers (MCMPCk) (2/2)

DIR1	DIR0	Selected output pins
0	0	Quadrant 1: SMk1 (sin +), SMk3 (cos +)
0	1	Quadrant 2: SMk1 (sin +), SMk4 (cos –)
1	0	Quadrant 3: SMk2 (sin –), SMk4 (cos –)
1	1	Quadrant 4: SMk2 (sin –), SMk3 (cos +)

Selects the output pins for the PWM signals.

Bits DIR1 and DIR0 address the quadrant to be activated by sine and cosine. The PWM signal is routed to the specific pin with respect to the sin/cos of each quadrant.

At the other output pins, the output level is SMV_{SS}.

Remark These bits are only considered if bit AOUT is set to 0.

(6) Stepper motor port mode control register (SMPC)

The 8-bit SMPC register controls output mode of SMnm pins (n = 1 to 4, m 1 to 4).

These registers can be read/written in 8-bit or 1-bit units.

This register is cleared by any reset.

Figure 16-4. Format of Stepper Motor Port Mode Control Register (SMPC)

Address: FF3DH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SMPC	MOD4	MOD3	MOD2	MOD1	EN4	EN3	EN2	EN1

ENn	MODn	Port mode selection (n = 0 to 4)
0	–	Port mode All SMnm (n = 1 to 4, m = 1 to 4) are set to port function.
1	0	PWM full bridge mode SMnm (n = 1 to 4, m = 1 to 4) are set to full bridge output control mode.
1	1	PWM half bridge mode Depending on the DIRnk (n = 1 to 4, k = 0, 1) bit of the compare control registers (MCMPCn; n = 1 to 4), SMnm (n = 1 to 4, m = 1 to 4) are set to PWM output control mode or port mode.

<R>

An example of settings when n = 1 is as follows:

EN1	MOD1	DIR11	DIR10	PWM Output Pin Control				Output Mode
				SM11 (sin+)	SM12 (sin–)	SM13 (cos+)	SM14 (cos–)	
0	–	–	–	port	port	port	port	Port mode
1	0	0	0	PWM	0	PWM	0	PWM full bridge mode
1	0	0	1	PWM	0	0	PWM	
1	0	1	0	0	PWM	0	PWM	
1	0	1	1	0	PWM	PWM	0	
1	1	0	0	PWM	port	PWM	port	PWM half bridge mode
1	1	0	1	PWM	port	port	PWM	
1	1	1	0	port	PWM	port	PWM	
1	1	1	1	port	PWM	PWM	port	

Caution Set port registers (Pn) and port mode registers (PMn) whose pins are not in the PWM mode to 00H in the PWM full bridge mode.

(7) ZPD detection voltage setting registers (ZPDS0, ZPDS1)

The 8-bit ZPDS0, ZPDS1 registers set ZPD detection voltage and control ZPD analog input.

These registers can be read/written in 8-bit or 1-bit units.

This register is cleared by any reset.

Figure 16-5. Format of ZPD detection voltage setting registers (ZPDS0, ZPDS1)

Address: F9FCH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ZPDS0	ZPD2PC	ZPD2S2	ZPD2S1	ZPD2S0	ZPD1PC	ZPD1S2	ZPD1S1	ZPD1S0

Address: F9FDH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
<R> ZPDS1[Note]	ZPD4PC	ZPD4S2	ZPD4S1	ZPD4S0	ZPD3PC	ZPD3S2	ZPD3S1	ZPD3S0

Note 78K0/DF2 only.

ZPDnS2	ZPDnS1	ZPDnS0	0-point detection voltage setting for ZPDn (n = 1 to 4)
0	0	0	SMV_{DD}×3/100 = 0.15 V
0	0	1	SMV_{DD}×5/100 = 0.25 V
0	1	0	SMV_{DD}×7/100 = 0.35 V
0	1	1	SMV_{DD}×9/100 = 0.45 V
1	0	0	SMV_{DD}×11/100 = 0.55 V
Other than above			Setting prohibited

ZPDnPC	Analog input/digital port selection
0	Digital port/SM pin
1	ZPD analog input

(8) ZPD flag detection clock setting register (CMPCTL)

The 8-bit CMPCTL register controls the clock for noise elimination.

This register can be read/written in 8-bit or 1-bit units.

This register is cleared by any reset.

Figure 16-6. Format of ZPD Flag Detection Clock Setting Register (CMPCTL)

Address: F9FEH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMPCTL	0	0	0	0	DBCL3	DBCL2	DBCL1	DBCL0

<R>

DBCL3	DBCL2	DBCL1	DBCL0	Selected clock
0	0	0	0	f_{PRS}
0	0	0	1	$f_{PRS}/2$
0	0	1	0	$f_{PRS}/2^2$
0	0	1	1	$f_{PRS}/2^3$
0	1	0	0	$f_{PRS}/2^4$
0	1	0	1	$f_{PRS}/2^5$
0	1	1	0	$f_{PRS}/2^6$
0	1	1	1	$f_{PRS}/2^7$
1	0	0	0	$f_{PRS}/2^8$
1	0	0	1	$f_{PRS}/2^9$
Other than above				Setting prohibited

(9) ZPD operational control register (ZPDEN)

The 8-bit ZPDEN register controls ZPD operation.

These registers can be read/written in 8-bit or 1-bit units.

This register is cleared by any reset.

Figure 16-7. Format of ZPD Operational Control Register (ZPDEN)

Address: F9FFH After reset: 00H R/W

<R>

Symbol	7	6	5	4	3	2	1	0
ZPDEN	0	0	0	0	ZPD4EN	ZPD3EN	ZPD2EN	ZPD1EN

ZPDnEN	ZPDn comparator operation (n = 1 to 4)
0	Disables operation.
1	Enables operation.

16.3 Operation

In the following, the operation of the Stepper Motor Controller/Driver module as a driver for external meters is described.

16.3.1 Stepper motor controller/driver operation

This section describes the generation of PWM signals of the driver k for driving external meters. Further, the achievable duty factor is explained and how advanced precision can be gained by 1-bit addition.

<R> **Remark** k = 1 to 4

(1) Driving meters

<R> External meters can be driven both in full bridge configuration and in half bridge configuration:

- Driving meters in full bridge configuration

  Deflection of the needle of a meter in full bridge configuration is determined by the sine and cosine value of its desired angle. Since the PWM signals do not inherit a sign, separate signals for positive and negative sine and cosine values are generated.

  The four signals at pins SMk1 to SMk4 of the driver k are:

  – sine side, positive (sin +)

  – sine side, negative (sin –)

  – cosine side, positive (cos +)

  – cosine side, negative (cos –)

  Two output control circuits select which signal (sign) for sine side and cosine side is output (bits MCMPCk.DIR[1:0]). At the remaining two output pins, the signal is set to low level.

  To drive meter k in full bridge mode, set bit MCMPCk.AOUT to 0.

- Driving meters in half bridge configuration

  In this mode, the same signal is sent to both sine pins (SMk1 and SMk2) and both cosine pins (SMk3 and SMk4), respectively. The setting of output control bits MCMPCk.DIR[1:0] is neglected.

  To drive meter k in half bridge mode, set bit MCMPCk.AOUT to 1.

(2) Generation of PWM signals

Bit data corresponding to the length of the PWM pulses has to be written to the compare registers MCMPk0 (sine side) and MCMPk1 (cosine side).

A timer counter is counting up. The rising edge of the PWM pulse is initiated at the overflow of the counter. The falling edge of the PWM pulse is initiated when the counter value equals the contents of the compare register.

The absolute pulse length in seconds is defined by the timer count clock (f_{MC0}). Various cycle times can be set via the timer mode control register MCNTC0.

Instruction

When writing data to compare registers, proceed as follows:

1. Confirm that MCMPCk.TEN = 0.
2. Write 8-bit PWM data to MCMPk0 and MCMPk1.
3. Set MCMPCk.ADB0 and MCMPCk.ADB1 as desired.
4. Set MCMPCk.TEN = 1 to start the counting operation.

   The data in MCMPk0/MCMPk1 will automatically be copied to the compare slave register when the counter overflows. The new pulse width is valid immediately.

   Bit MCMPCk.TEN is automatically cleared to 0 by hardware.

(3) Duty factor

The minimum pulse width that can be generated is zero (output signal is low) and the maximum pulse width is 255 clock cycles (maximum value of 8-bit compare registers).

The count range of the timer counter defines the duty factor. It can be set by bit MCNTC0.FULL:

- count range 01H to FFH (MCNTC0.FULL = 0)

  Formula for the duty cycle:

  PWM duty = MCMPki / 255 with k = 1 to 4 and i = 0, 1

  One count cycle is comprised of 255 clock cycles. A PWM signal with maximum pulse length is a steady high level signal. The duty factor is 100%.

- count range 00H to FFH (MCNTC0.FULL = 1)

  Formula for the duty cycle:

  PWM duty = MCMPki / 256 with k = 1 to 4 and i = 0, 1

  One count cycle is comprised of 256 clock cycles. A PWM signal with maximum pulse length is comprised of 255 clock cycles at high level and one clock cycle at low level. The duty factor is 255/256 *100% = 99.6%.

(4) Advanced precision by 1-bit addition

The precision of the angle of a needle is implicitly defined by the number of bits of the compare registers MCMPk0 and MCMPk1 (8 bit).

If the 1-bit addition circuit is enabled, every second pulse of the PWM signal is extended by one bit (one clock cycle). In average, a pulse width precision of 1/2 bit (1/2 clock) can be achieved.

The following figures show the timing of PWM output signals with 1-bit addition disabled and enabled.

Remarks 1. The PWM pulse is not generated until the first overflow occurs after the counting operation has been started.

2. The PWM signal is two cycle counts delayed compared to the overflow signal and the match signal. This is not depicted in the figures.

(5) Detecting zero points

For the detection of zero points, proceed as follows:

1. Set ZPDn pin to analog input by setting ZPDnEN = 0, and select reference level using ZPDnS2 to ZPDnS0.
2. <R> Enable ZPD comparator operation by setting ZPDnEN = 1, and wait for comparator stabilization time.
3. Enable ZPDn flag operation by setting TWIN = 1.
4. Apply input signal to ZPDn pin, and start detection operation.

(6) Digital noise filter

The noise removal circuit suppresses short pulses/spikes of the comparator output to gain stable comparison results.

The minimum voltage comparator output pulse width to be validated is configurable by selecting the sampling clock for the digital noise removal, refer to CMPCTL.DBCL[3:0]. Spikes shorter than 2 sampling cycles are suppressed. Pulses longer than 3 sampling cycles are recognized as valid pulses. For pulses between 2 and 3 sampling cycles, the behavior is not defined.

Figure 16-8. Output Timing without 1-bit Addition

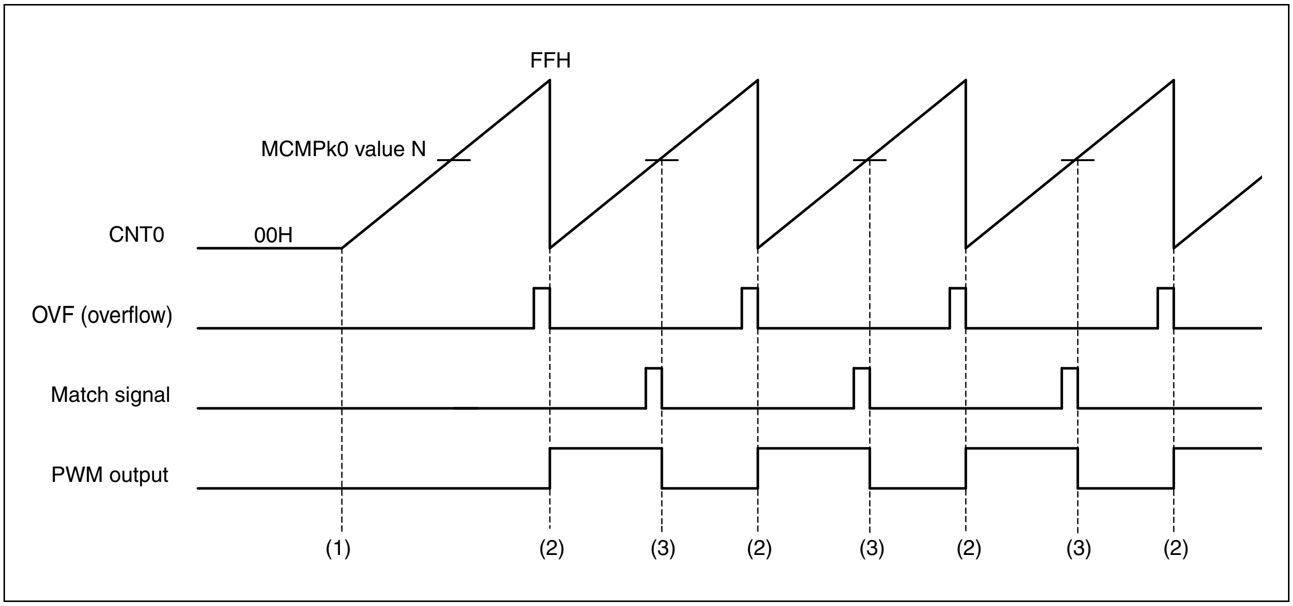
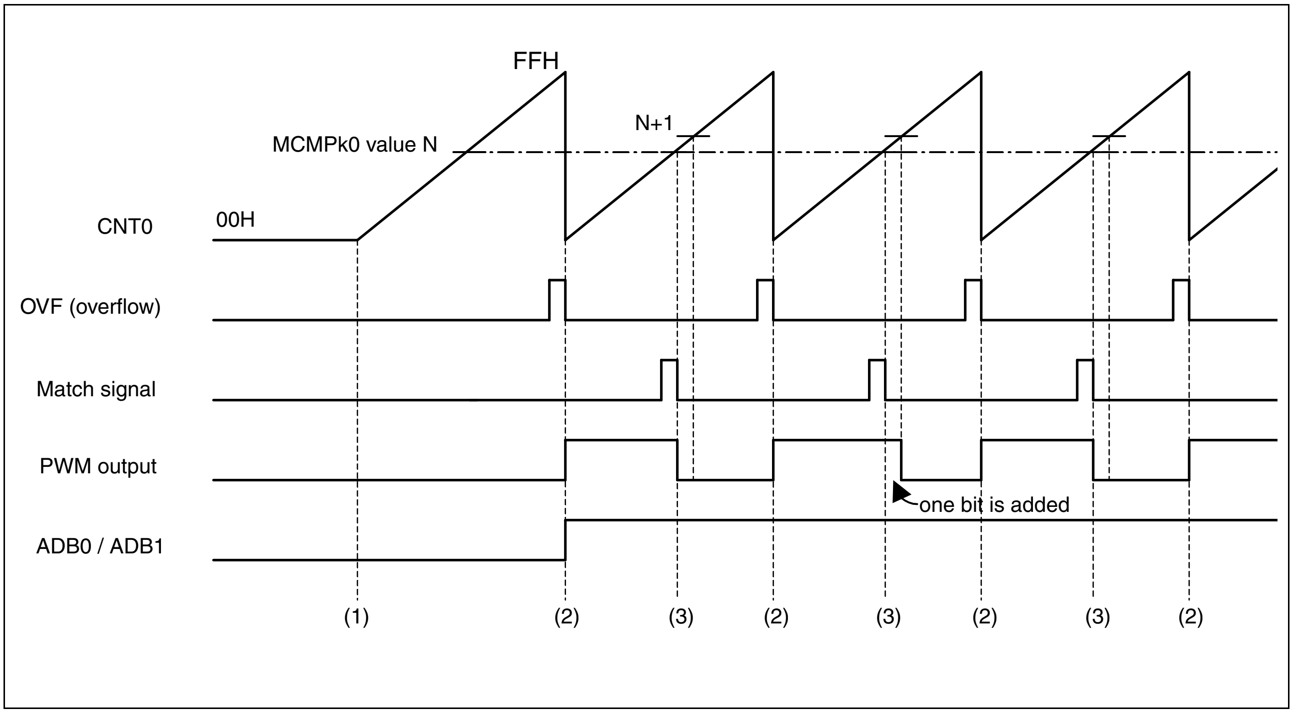


Figure 16-9. Output Timing with 1-bit Addition



Sequence

1. Start of counting (MCNTC0.PCE is set to 1)
2. Generation of overflow signal (start of PWM pulse)
3. Generation of match signal (timer counter CNT0 matches compare register, end of PWM pulse)

16.4 Timing

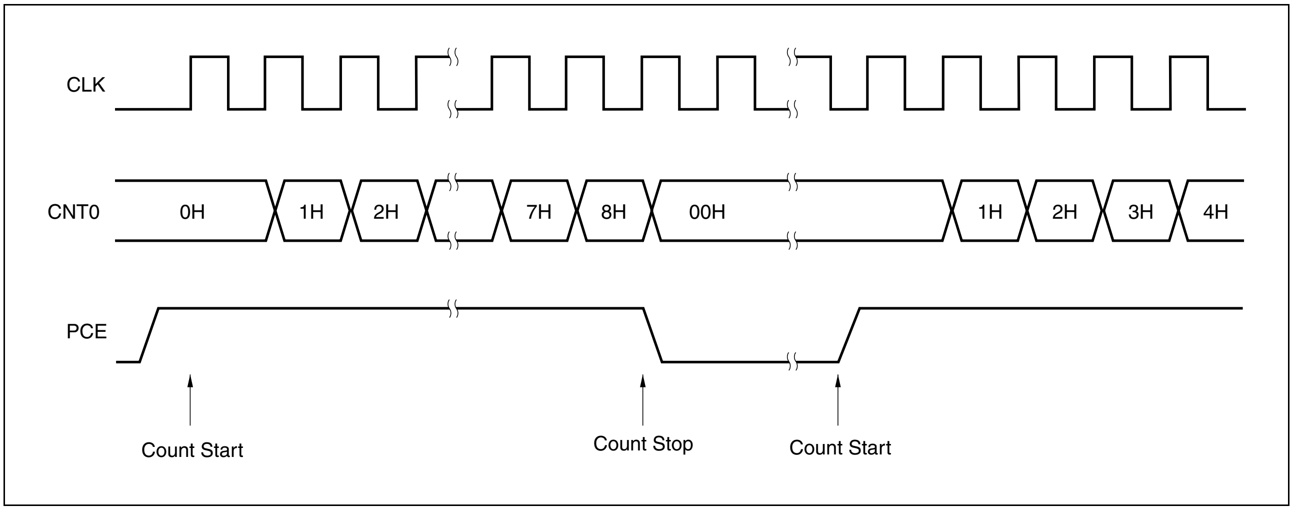
This section starts with the timing of the timer counter and general output timing behaviour. Then, examples of output signal generation with and without 1-bit addition are presented.

16.4.1 Timer counter

The free running up counter is clocked by the timer count clock selected in register MCNTC0.

The counting operation is enabled or disabled by the MCNTC0.PCE bit.

Figure 16-10. Restart Timing after Count Stop (Count Start—Count Stop—Count Start)



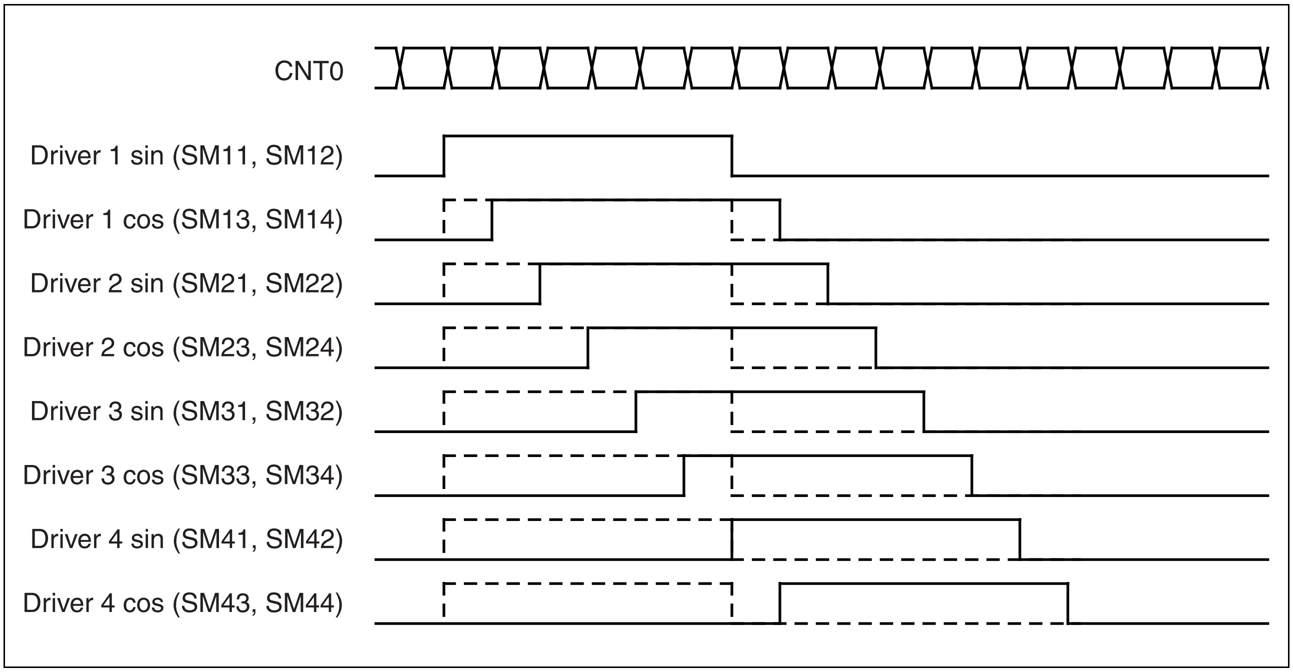
Sequence

- Count Start:
  - Enable counting operation (MCNTC0.PCE = 1)
  - Timer counter starts with value 00H. Depending on bit MCNTC0.FULL, all following counter cycles start with 00H or 01H, respectively.
- Count Stop:
  - Disable counting operation (MCNTC0.PCE = 0)
  - Counting is stopped and timer counter is set to 00H.

16.4.2 Automatic PWM phase shift

Simultaneous switching of sine and cosine output could lead to a fluctuation of the power supply and increase the susceptibility to electromagnetic interference. To prevent this for drivers 1 to 4, the output signals are automatically shifted by one timer count clock cycle defined in MCNTC0.

Figure 16-11. Output Timing of Signals SM11 to SM44


CNT0
Driver 1 sin (SM11, SM12)
Driver 1 cos (SM13, SM14)
Driver 2 sin (SM21, SM22)
Driver 2 cos (SM23, SM24)
Driver 3 sin (SM31, SM32)
Driver 3 cos (SM33, SM34)
Driver 4 sin (SM41, SM42)
Driver 4 cos (SM43, SM44)

17.1 Functions of LCD Controller/Driver

The functions of the LCD controller/driver in the 78K0/Dx2 are as follows.

(1) The LCD driver voltage generator uses internal resistance division method.

(2) Automatic output of segment and common signals based on automatic display data memory read

(3) Three different display modes:

- Static
- 1/3 duty (1/3 bias)
- 1/4 duty (1/3 bias)

(4) Six different frame frequencies, selectable in each display mode

(5)

μPD78F0838, 78F0839:	Segment signal outputs: 40 (SEG0 to SEG39), Common signal outputs: 4 (COM0 to COM3)
μPD78F0840, 78F0841, 78F0846, 78F0847:	Segment signal outputs: 32 (SEG0 to SEG31), Common signal outputs: 4 (COM0 to COM3)
μPD78F0842, 78F0843, 78F0848, 78F0849:	Segment signal outputs: 28 (SEG0 to SEG27), Common signal outputs: 4 (COM0 to COM3)
μPD78F0836, 78F0837, 78F0844, 78F0845:	Segment signal outputs: 24 (SEG0 to SEG23), Common signal outputs: 4 (COM0 to COM3)

Table 17-1 lists the maximum number of pixels that can be displayed in each display mode.

Table 17-1. Maximum Number of Pixels

(a) μPD78F0838, 78F0839

LCD Driver Voltage Generator	Bias Mode	Number of Time Slices	Common Signals Used	Number of Segments	Maximum Number of Pixels
• Internal resistance division	–	Static	COM0 (COM1 to COM3)	40	40 (40 segment signals, 1 common signal)[Note 1]
	1/3	3	COM0 to COM2		120 (40 segment signals, 3 common signals)[Note 2]
		4	COM0 to COM3		160 (40 segment signals, 4 common signals)[Note 3]

Notes 1. 5-digit LCD panel, each digit having an 8-segment 8 configuration.

2. 15-digit LCD panel, each digit having a 3-segment 8 configuration.

3. 20-digit LCD panel, each digit having a 2-segment 8 configuration.

(b) μPD78F0840, 78F0841, 78F0846, 78F0847

LCD Driver Voltage Generator	Bias Mode	Number of Time Slices	Common Signals Used	Number of Segments	Maximum Number of Pixels
• Internal resistance division	–	Static	COM0 (COM1 to COM3)	32	32 (32 segment signals, 1 common signal)[Note 1]
	1/3	3	COM0 to COM2		96 (32 segment signals, 3 common signals)[Note 2]
		4	COM0 to COM3		128 (32 segment signals, 4 common signals)[Note 3]

Notes 1. 4-digit LCD panel, each digit having an 8-segment 8 configuration.

2. 12-digit LCD panel, each digit having a 3-segment 8 configuration.

3. 16-digit LCD panel, each digit having a 2-segment 8 configuration.

(c) μPD78F0842, 78F0843, 78F0848, 78F0849

LCD Driver Voltage Generator	Bias Mode	Number of Time Slices	Common Signals Used	Number of Segments	Maximum Number of Pixels
• Internal resistance division	–	Static	COM0 (COM1 to COM3)	28	28 (28 segment signals, 1 common signal)[Note 1]
	1/3	3	COM0 to COM2		84 (28 segment signals, 3 common signals)[Note 2]
		4	COM0 to COM3		112 (28 segment signals, 4 common signals)[Note 3]

Notes 1. 4-digit LCD panel, each digit having an 8-segment 8 configuration.

2. 12-digit LCD panel, each digit having a 3-segment 8 configuration.

3. 16-digit LCD panel, each digit having a 2-segment 8 configuration.

(d) μPD78F0836, 78F0837, 78F0844, 78F0845

LCD Driver Voltage Generator	Bias Mode	Number of Time Slices	Common Signals Used	Number of Segments	Maximum Number of Pixels
• Internal resistance division	–	Static	COM0 (COM1 to COM3)	24	24 (24 segment signals, 1 common signal)[Note 1]
	1/3	3	COM0 to COM2		72 (24 segment signals, 3 common signals)[Note 2]
		4	COM0 to COM3		96 (24 segment signals, 4 common signals)[Note 3]

Notes 1. 3-digit LCD panel, each digit having an 8-segment 8. configuration.

2. 9-digit LCD panel, each digit having a 3-segment 8. configuration.

3. 12-digit LCD panel, each digit having a 2-segment 8. configuration.

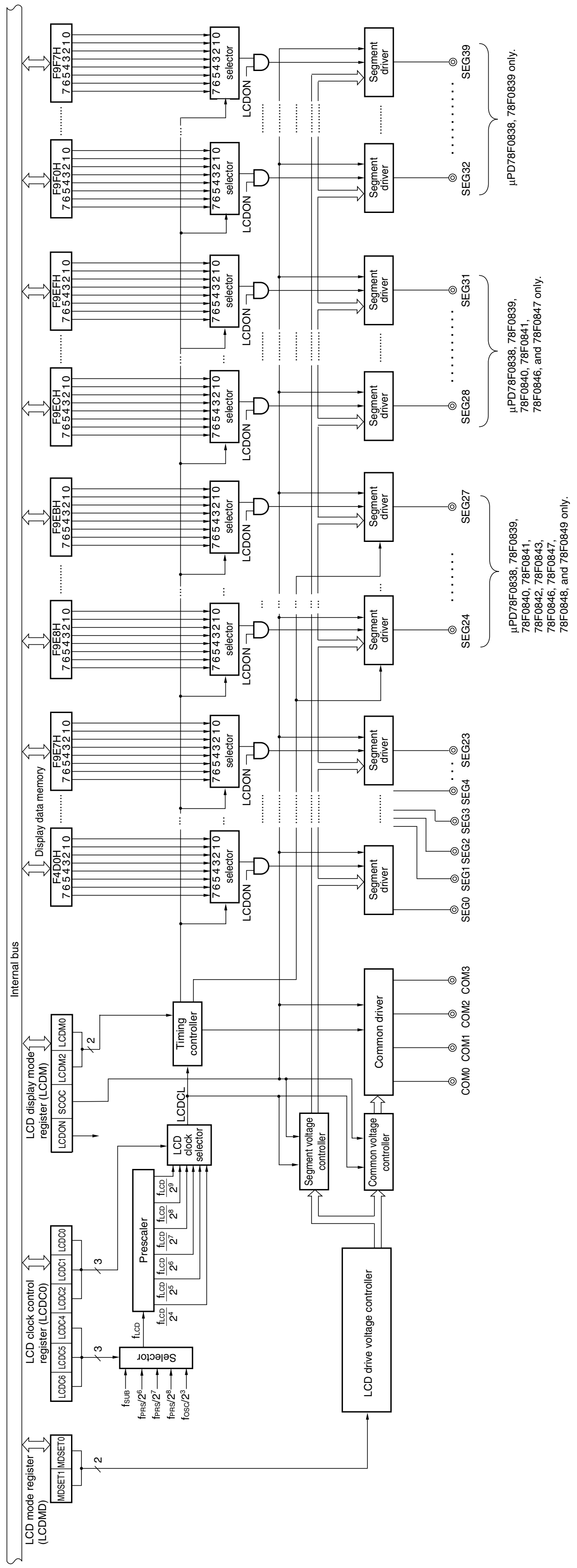
17.2 Configuration of LCD Controller/Driver

The LCD controller/driver consists of the following hardware.

Table 17-2. Configuration of LCD Controller/Driver

Item	Configuration	
Display outputs	μPD78F0838, 78F0839	40 segment signals (SEG0 to SEG39), 4 common signals (COM0 to COM3)
	μPD78F0840, 78F0841, 78F0846, 78F0847	32 segment signals (SEG0 to SEG31), 4 common signals (COM0 to COM3)
	μPD78F0842, 78F0843, 78F0848, 78F0849	28 segment signals (SEG0 to SEG27), 4 common signals (COM0 to COM3)
	μPD78F0836, 78F0837, 78F0844, 78F0845	24 segment signals (SEG0 to SEG23), 4 common signals (COM0 to COM3)
Control registers	LCD mode register (LCDMD) LCD display mode register (LCDM) LCD clock control register (LCDC0) LCD port function register 0 (LCDPF0) LCD port function register 3 (LCDPF3) LCD port function register ALL (LCDPFALL)	

Figure 17-1. Block Diagram of LCD Controller/Driver



17.3 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)
- LCD port function register 0 (LCDPF0)
- LCD port function register 3 (LCDPF3)
- LCD port function register ALL (LCDPFALL)

(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 17-2. Format of LCD Mode Register

Address: FF5AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDMD	0	0	MDSET1	MDSET0	0	0	0	0

MDSET1	MDSET0	LCD drive voltage generator selection
0	0	No internal resistor connection (power save mode).
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

Caution Bits 0 to 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 17-3. Format of LCD Display Mode Register

Address: FF5BH After reset: 00H R/W

Symbol	<7>	<6>	5	4	3	2	1	0
LCDM	LCDON	SCOC	0	0	0	LCDM2	0	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]
0	Output ground level to segment/common pin
1	Output deselect level to segment pin and LCD waveform to common pin

LCDM2	LCDM0	LCD controller/driver display mode selection	
		Resistance division method	
		Number of time slices	Bias mode
0	0	4	1/3
0	1	3	1/3
1	0	Static	
Other than above		Setting prohibited	

Note When LCD display is not to be performed or not required, power consumption can be reduced by using the following settings.

<1> Set SCOC (bit 6 of the LCD display mode register (LCDM)) to 0.

<2> Set MDSET0 and MDSET1 (bits 4 and 5 of the LCD mode register (LCDMD)) to 0.
(The current flowing to the internal resistors can be reduced.)

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

2. When displaying in a mode with a large number of COMs, such as 4 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 17-4. Format of LCD Clock Control Register

Address: FF5CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDC0	0	LCDC6	LCDC5	LCDC4	0	LCDC2	LCDC1	LCDC0

<R>

LCDC6	LCDC5	LCDC4	LCD source clock (f_{LCD}) selection
0	0	0	f_{SUB} (32.768 kHz)
0	0	1	$f_{PRS}/2^7$
0	1	0	$f_{PRS}/2^8$
0	1	1	$f_{PRS}/2^9$
1	0	0	$f_{OSC}/2^3$
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_{SUB}: XT1 clock oscillation frequency

2. f_{PRS}: Peripheral hardware clock frequency

3. f_{OSC}: Internal low-speed oscillation clock frequency

(4) LCD port function register 0 (LCDPF0)

This register sets whether to use pins P00 to P07 as port pins (other than segment output pins) or segment output pins.

LCDPF0 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets LCDPF0 to 00H.

Figure 17-5. Format of LCD Port Function Register 0

Address: FF1BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDPF0	PF07	PF06	PF05	PF04	PF03	PF02	PF01	PF00

PF0n	Port/segment output specification
0	Used as port (other than segment output)
1	Used as segment output

Remark n = 0 to 7

(5) LCD port function register 3 (LCDPF3)

This register sets whether to use pins P30 to P37 as port pins (other than segment output pins) or segment output pins.

LCDPF3 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets LCDPF3 to 00H.

Figure 17-6. Format of LCD Port Function Register 3

Address: FF0DH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDPF3	PF37	PF36	PF35	PF34	PF33	PF32	PF31	PF30

PF3n	Port/segment output specification
0	Used as port (other than segment output)
1	Used as segment output

Remark n = 0 to 7

(6) LCD port function register ALL (LCDPFALL)

This register sets whether to use pins P74 to P77, P13 to P16, P8 and P9 as port pins (other than segment output pins) or segment output pins.

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

Reset signal generation sets LCDPFALL to 00H.

Figure 17-7. Format of LCD Port Function Register ALL

Address: FF1AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDPF ALL	PF7UPNIB	PF16	PF15	PF14	PF13	PF9ALL	PF8ALL	0

PF7UPNIB	Port/segment output specification
0	Used as port (other than segment output)
1	Used as segment output

PF1n	Port/segment output specification
0	Used as port (other than segment output)
1	Used as segment output

Remark n = 3 to 6

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

Remark n = 8, 9

17.4 Setting LCD Controller/Driver

Set the LCD controller/driver using the following procedure:

<1> Set the LCD drive method via MDSET0 and MDSET1 (bits 4 and 5 of the LCD mode register (LCDMD)).
<2> Set the pins to be used as segment outputs to the port function registers (LCDPF0, LCDPF3, LCDALL).
<3> Set an initial value to the RAM for LCD display.
<4> Set the number of time slices via LCDM0 to LCDM2 (bits 0 to 2 of the LCD display mode register (LCDM)).
<5> Set the LCD source clock and LCD clock via LCD clock control register (LCDC0).
<6> Set SCOC (bit 6 of the LCD display mode register (LCDM)) to 1.
<7> Start output corresponding to each data memory by setting LCDON (bit 7 of the LCD display mode register (LCDM)) to 1.

Subsequent to this procedure, set the data to be displayed in the data memory.

Remark 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 to no internal resistor connection (power save mode). <R>

Caution When displaying in a mode with a large number of COMs, such as 4 COM, VLC0 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.

17.5 LCD Display Data Memory

The LCD display data memory is mapped at addresses F9D0H to F9F7H. Data in the LCD display data memory can be displayed on the LCD panel using the LCD controller/driver.

Figure 17-8 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 17-8. Relationship between LCD Display Data Memory Contents and Segment/Common Outputs (1/2)

(a) μPD78F0838, 78F0839

	b7	b6	b5	b4	b3	b2	b1	b0	
F9F7H	0	0	0	0					SEG39
F9F6H	0	0	0	0					SEG38
F9F5H	0	0	0	0					SEG37

F9D5H	0	0	0	0					SEG5
F9D4H	0	0	0	0					SEG4
F9D3H	0	0	0	0					SEG3
F9D2H	0	0	0	0					SEG2
F9D1H	0	0	0	0					SEG1
F9D0H	0	0	0	0					SEG0
					COM3	COM2	COM1	COM0	

(b) μPD78F0840, 78F0841, 78F0846, 78F0847

	b7	b6	b5	b4	b3	b2	b1	b0	
F9EFH	0	0	0	0					SEG31
F9EEH	0	0	0	0					SEG30
F9ECH	0	0	0	0					SEG29

F9D5H	0	0	0	0					SEG5
F9D4H	0	0	0	0					SEG4
F9D3H	0	0	0	0					SEG3
F9D2H	0	0	0	0					SEG2
F9D1H	0	0	0	0					SEG1
F9D0H	0	0	0	0					SEG0
					COM3	COM2	COM1	COM0	

Caution No memory is allocated to the higher 4 bits. Be sure to set there bits to 0.

Figure 17-8. Relationship between LCD Display Data Memory Contents and Segment/Common Outputs (2/2)

(c) μPD78F0842, 78F0843, 78F0848, 78F0849

	b7	b6	b5	b4	b3	b2	b1	b0	
F9EBH	0	0	0	0					SEG27
F9EAH	0	0	0	0					SEG26
F9E9H	0	0	0	0					SEG25

F9D5H	0	0	0	0					SEG5
F9D4H	0	0	0	0					SEG4
F9D3H	0	0	0	0					SEG3
F9D2H	0	0	0	0					SEG2
F9D1H	0	0	0	0					SEG1
F9D0H	0	0	0	0					SEG0
					COM3	COM2	COM1	COM0	

(d) μPD78F0836, 78F0837, 78F0844, 78F0845

	b7	b6	b5	b4	b3	b2	b1	b0	
F9E7H	0	0	0	0					SEG23
F9E6H	0	0	0	0					SEG22
F9E5H	0	0	0	0					SEG21

F9D5H	0	0	0	0					SEG5
F9D4H	0	0	0	0					SEG4
F9D3H	0	0	0	0					SEG3
F9D2H	0	0	0	0					SEG2
F9D1H	0	0	0	0					SEG1
F9D0H	0	0	0	0					SEG0
					COM3	COM2	COM1	COM0	

Caution No memory is allocated to the higher 4 bits. Be sure to set there bits to 0.

17.6 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}.

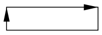
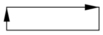
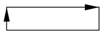
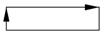
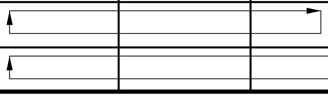
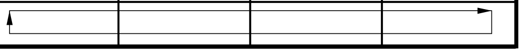
Applying DC voltage to the common and segment signals of an LCD panel causes deterioration. To avoid this problem, this LCD panel is driven by AC voltage.

(1) Common signals

Each common signal is selected sequentially according to a specified number of time slices at the timing listed in **Table 17-3**. In the static display mode, the same signal is output to COM0 to COM3.

In the three-time-slice mode, leave the COM3 pin open.

Table 17-3. COM Signals

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

(2) Segment signals

(a) μPD78F0838, 78F0839

The segment signals correspond to 40 bytes of the LCD display data memory (F9D0H to F9F7H) 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).

(b) μPD78F0840, 78F0841, 78F0846, 78F0847

The segment signals correspond to 32 bytes of the LCD display data memory (F9D0H to F9EFH) 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).

(c) μPD78F0842, 78F0843, 78F0848, 78F0849

The segment signals correspond to 28 bytes of the LCD display data memory (F9D0H to F9EBH) 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 SEG27).

(d) μPD78F0836, 78F0837, 78F0844, 78F0845

The segment signals correspond to 24 bytes of the LCD display data memory (F9D0H to F9E7H) 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 SEG23).

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, and bit 3 are not used for LCD display in the static display, three-time slot modes, respectively. So these bits can be used for purposes other than display.

The higher 4 bits of “F9D0H to F9D3H” are fixed to 0.

(3) Output waveforms of common signals and segment signals during LCD display signal output period

The voltages shown in **Table 17-4** 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 ±V_{LCD} is obtained. The other combinations of the signals correspond to the display off-voltage.

Table 17-4. LCD Drive Voltage

(a) Static display mode (during LCD display signal output period)

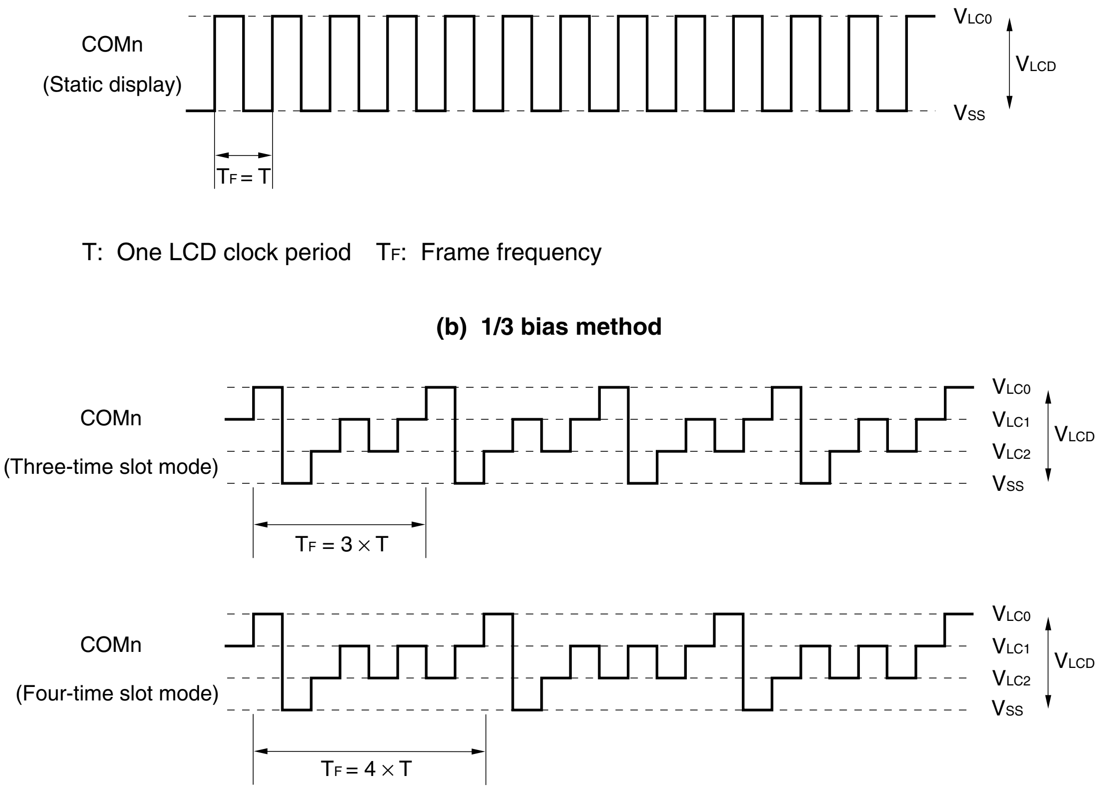
Segment Signal Common Signal	Select Signal Level	Deselect Signal Level
	V_{SS}/V_{LC0}	V_{LC0}/V_{SS}
V_{LC0}/V_{SS}	$-V_{LCD}/+V_{LCD}$	0 V/0 V

(b) 1/3 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}	$-\frac{1}{3}V_{LCD}/+\frac{1}{3}V_{LCD}$	$+\frac{1}{3}V_{LCD}/-\frac{1}{3}V_{LCD}$

Figure 17-9 shows the common signal waveforms, and **Figure 17-10** shows the voltages and phases of the common and segment signals.

Figure 17-9. Common Signal Waveforms



(a) Static display mode

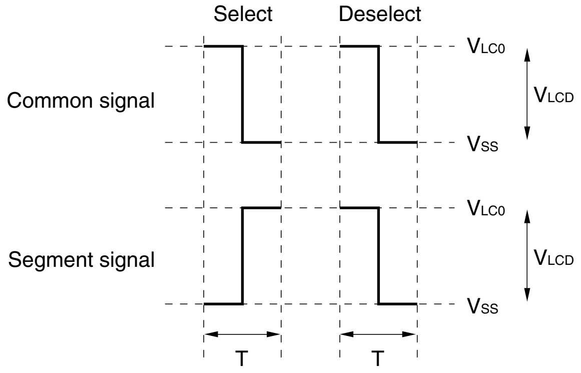
T: One LCD clock period T_F: Frame frequency

(b) 1/3 bias method

T: One LCD clock period T_F: Frame frequency

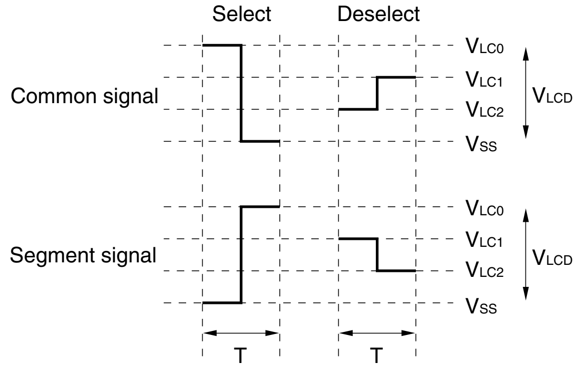
Figure 17-10. Voltages and Phases of Common and Segment Signals

(a) Static display mode



T: One LCD clock period

(b) 1/3 bias method



T: One LCD clock period

17.7 Display Modes

17.7.1 Static display example

Figure 17-12 shows how the three-digit LCD panel having the display pattern shown in **Figure 17-11** is connected to the segment signals (SEG0 to SEG23) and the common signal (COM0) of the 78K0/Dx2 chip. This example displays data "12.3" in the LCD panel. The contents of the display data memory (F9D0H to F9E7H) 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 17-5** at the timing of the common signal COM0; see **Figure 17-11** for the relationship between the segment signals and LCD segments.

Table 17-5. Select and Deselect Voltages (COM0)

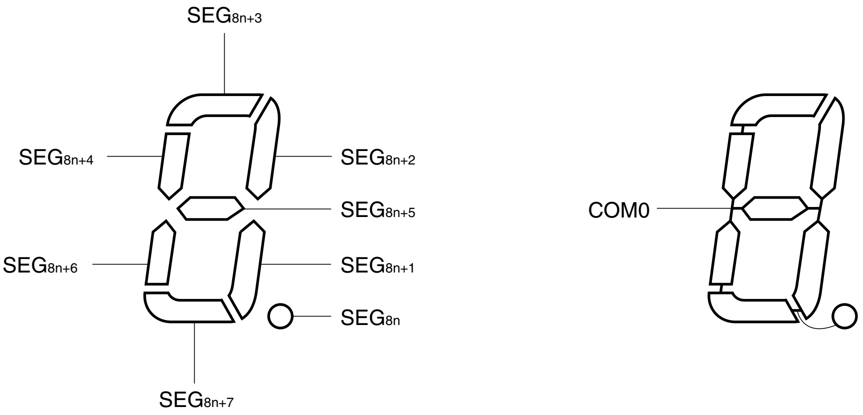
Segment Common	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15
COM0	Select	Deselect	Select	Select	Deselect	Select	Select	Select

According to **Table 17-5**, it is determined that the bit-0 pattern of the display data memory locations (F9D8H to F9DFH) must be 10110111.

Figure 17-13 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 17-11. Static LCD Display Pattern and Electrode Connections



Remark n = 0 to 2

Figure 17-12. Example of Connecting Static LCD Panel

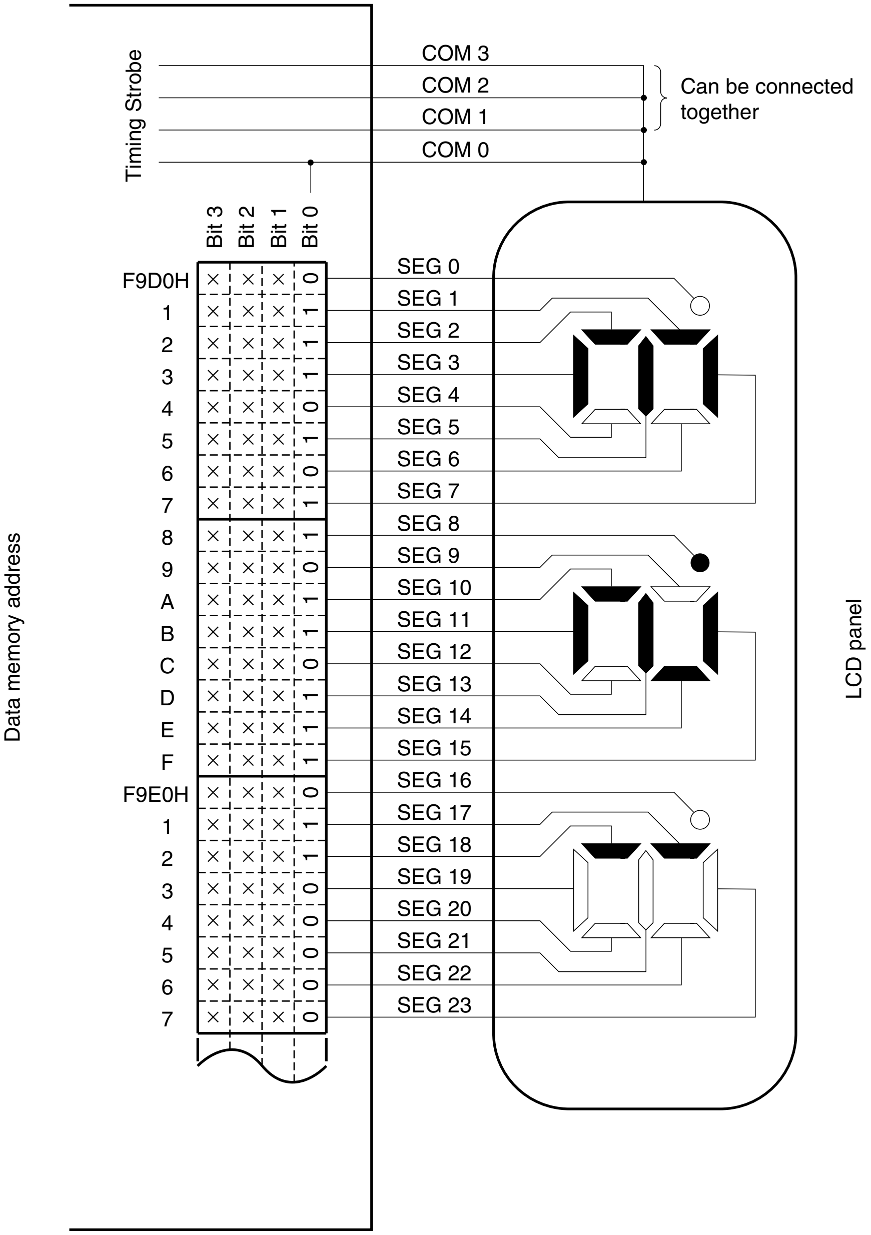
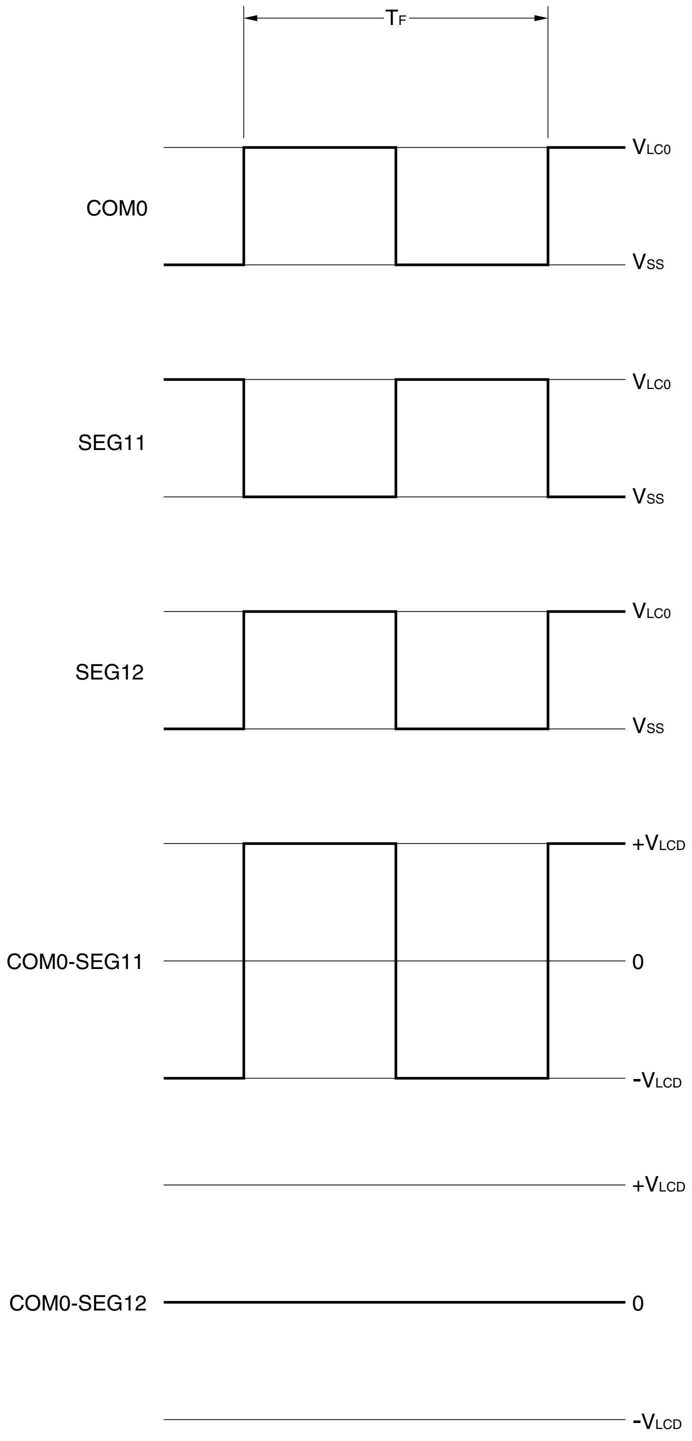


Figure 17-13. Static LCD Drive Waveform Examples



17.7.2 Three-time-slice display example

Figure 17-15 shows how the 8-digit LCD panel having the display pattern shown in **Figure 17-14** is connected to the segment signals (SEG0 to SEG23) and the common signals (COM0 to COM2) of the 78K0/Dx2 chip. This example displays data "123456.78" in the LCD panel. The contents of the display data memory (addresses F9D0H to F9E7H) 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 17-6** at the timing of the common signals COM0 to COM2; see **Figure 17-14** for the relationship between the segment signals and LCD segments.

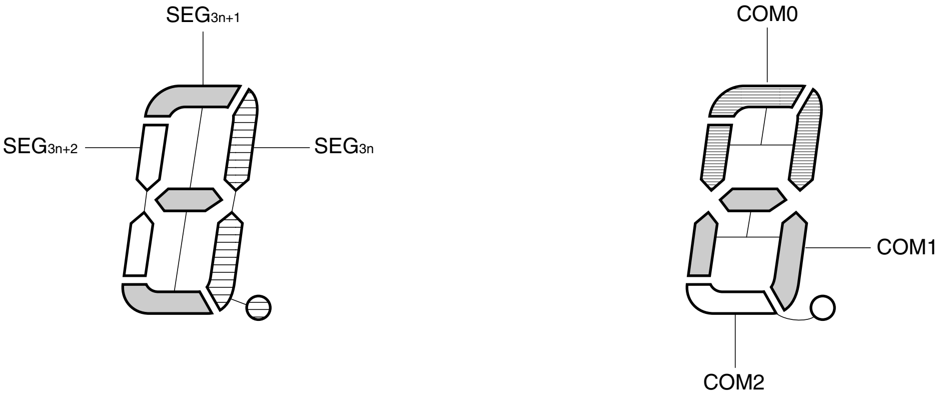
Table 17-6. Select and Deselect Voltages (COM0 to COM2)

Segment Common	SEG6	SEG7	SEG8
COM0	Deselect	Select	Select
COM1	Select	Select	Select
COM2	Select	Select	–

According to **Table 17-6**, it is determined that the display data memory location (F9D6H) that corresponds to SEG6 must contain x110.

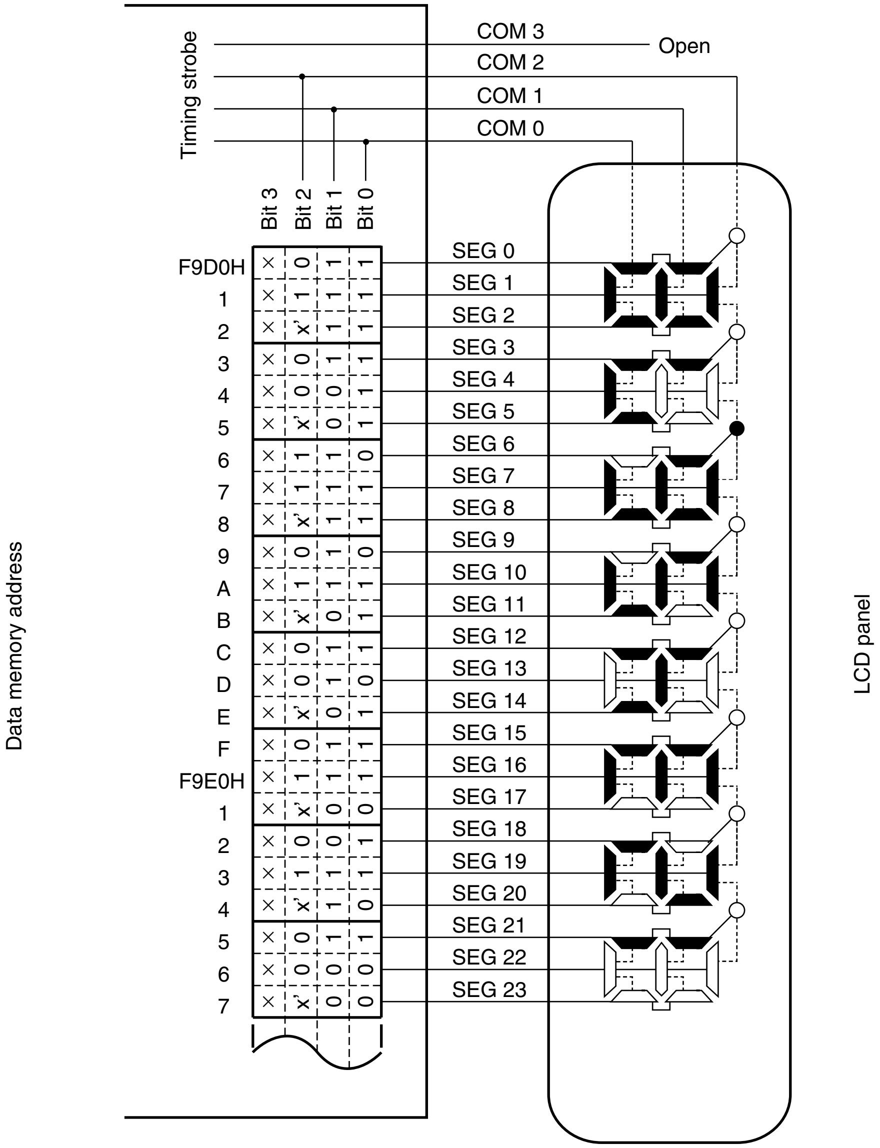
Figure 17-16 shows an example of LCD drive waveforms between the SEG6 signal and each common signal in the 1/3 bias method. When the select voltage is applied to SEG6 at the timing of COM1 or COM2, an alternate rectangle waveform, +V_{LCD}/−V_{LCD}, is generated to turn on the corresponding LCD segment.

Figure 17-14. Three-Time-Slice LCD Display Pattern and Electrode Connections



Remark n = 0 to 7

Figure 17-15. Example of Connecting Three-Time-Slice LCD Panel

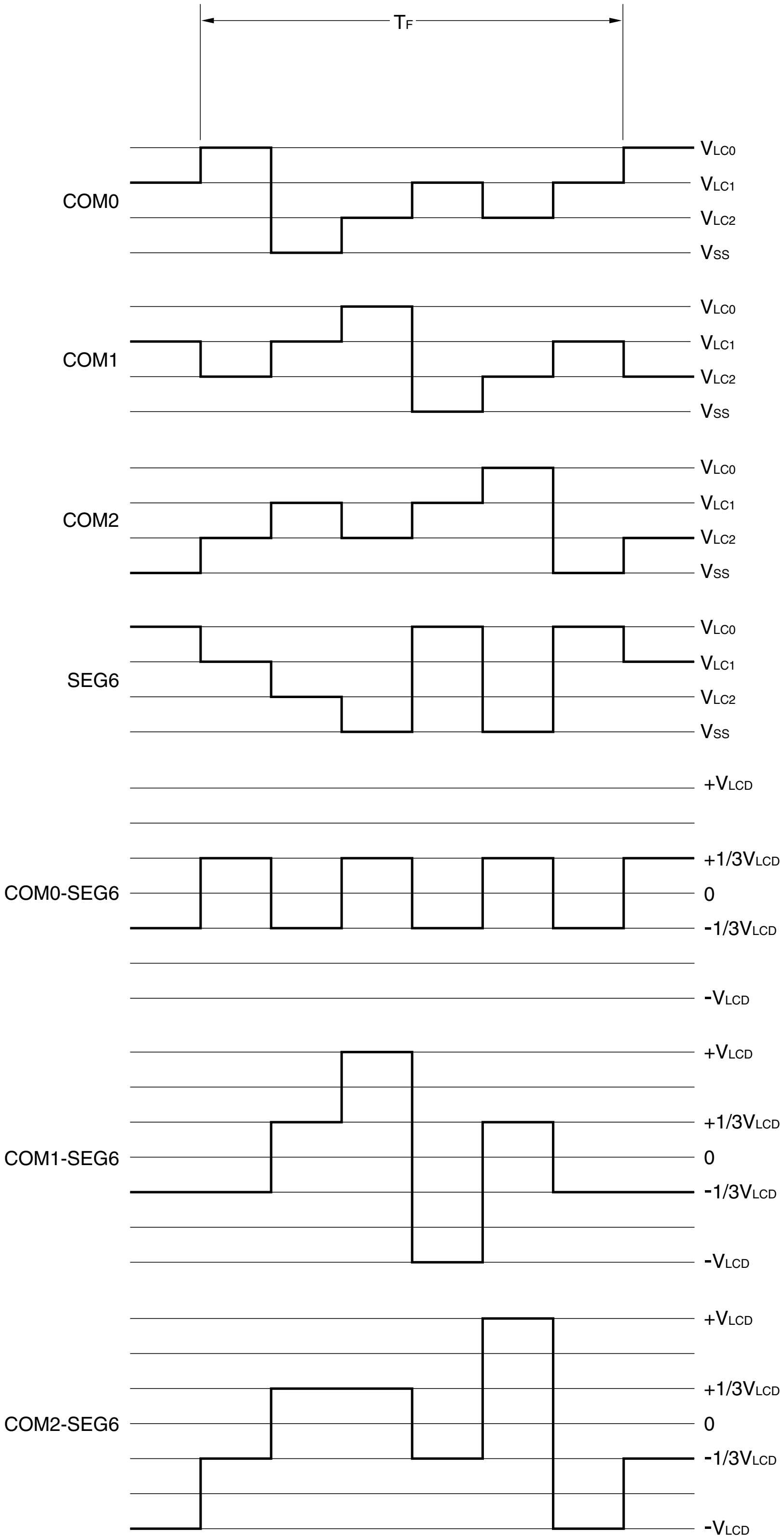


Data memory address	Bit 3	Bit 2	Bit 1	Bit 0
F9D0H	×	0	1	1
1	×	1	1	1
2	×	×'	1	1
3	×	0	1	1
4	×	0	0	1
5	×	×'	0	1
6	×	1	1	0
7	×	1	1	1
8	×	×'	1	1
9	×	0	1	0
A	×	1	1	1
B	×	×'	0	1
C	×	0	1	1
D	×	0	1	0
E	×	×'	0	1
F	×	0	1	1
F9E0H	×	1	1	1
1	×	×'	0	0
2	×	0	0	1
3	×	1	1	1
4	×	×'	1	0
5	×	0	1	1
6	×	0	0	0
7	×	×'	0	0

×': 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 17-16. Three-Time-Slice LCD Drive Waveform Examples (1/3 Bias Method)



17.7.3 Four-time-slice display example

Figure 17-18 shows how the 12-digit LCD panel having the display pattern shown in **Figure 17-17** is connected to the segment signals (SEG0 to SEG23) and the common signals (COM0 to COM3) of the 78K0/Dx2 chip. This example displays data "123456.789012" in the LCD panel. The contents of the display data memory (addresses F9D0H to F9E7H) 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 17-7** at the timing of the common signals COM0 to COM3; see **Figure 17-17** for the relationship between the segment signals and LCD segments.

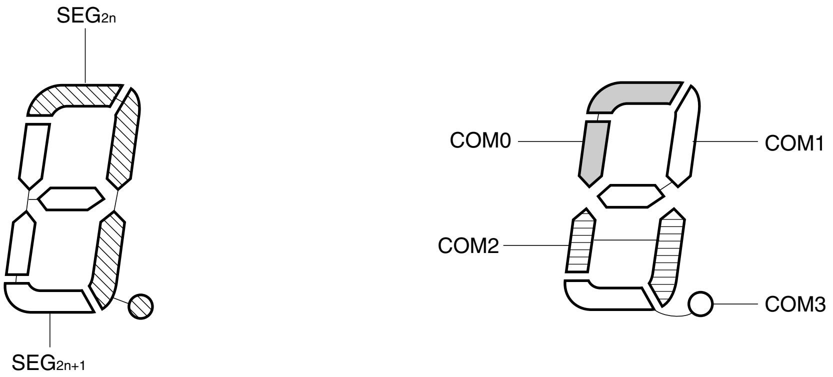
Table 17-7. Select and Deselect Voltages (COM0 to COM3)

Segment Common	SEG12	SEG13
COM0	Select	Select
COM1	Deselect	Select
COM2	Select	Select
COM3	Select	Select

According to **Table 17-7**, it is determined that the display data memory location (F9DCH) that corresponds to SEG12 must contain 1101.

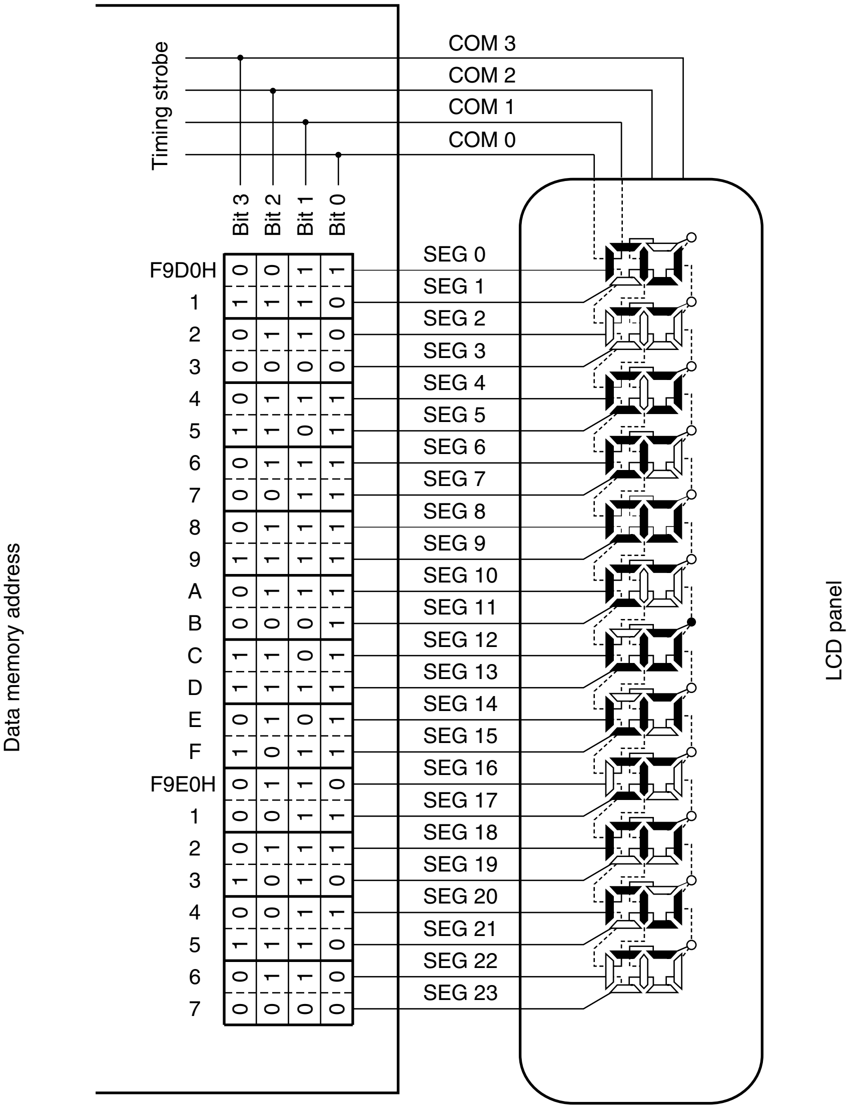
Figure 17-19 shows examples of LCD drive waveforms between the SEG12 signal and each common signal. When the select voltage is applied to SEG12 at the timing of COM0, an alternate rectangle waveform, $+V_{LCD}$/$-V_{LCD}$, is generated to turn on the corresponding LCD segment.

Figure 17-17. Four-Time-Slice LCD Display Pattern and Electrode Connections



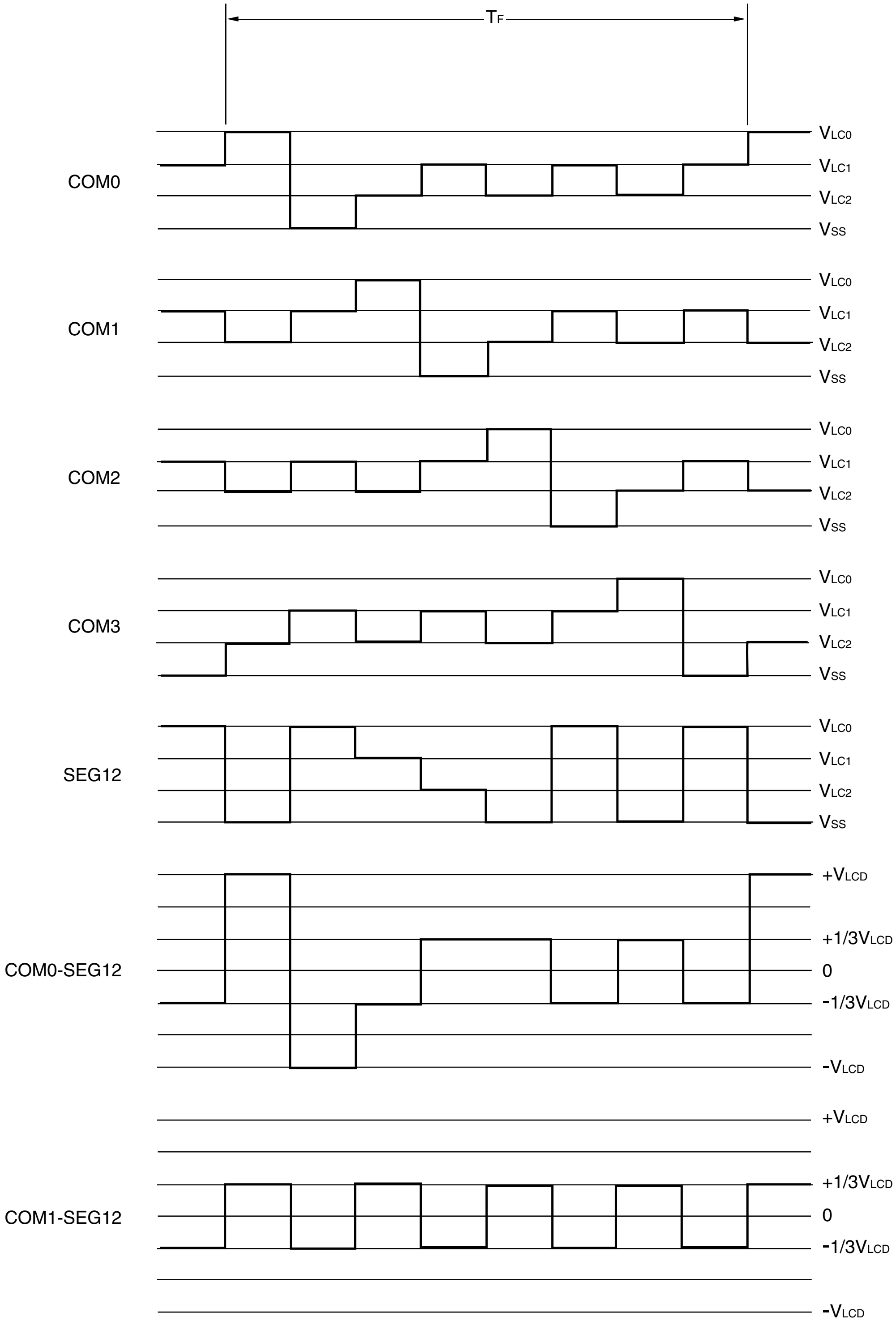
Remark n = 0 to 11

Figure 17-18. Example of Connecting Four-Time-Slice LCD Panel



	Bit 3	Bit 2	Bit 1	Bit 0
F9D0H	0	0	1	1
1	1	1	1	0
2	0	1	1	0
3	0	0	0	0
4	0	1	1	1
5	1	1	0	1
6	0	1	1	1
7	0	0	1	1
8	0	1	1	1
9	1	1	1	1
A	0	1	1	1
B	0	0	0	1
C	1	1	0	1
D	1	1	1	1
E	0	1	0	1
F	1	0	1	1
F9E0H	0	1	1	0
1	0	0	1	1
2	0	1	1	1
3	1	0	1	0
4	0	0	1	1
5	1	1	1	0
6	0	1	1	0
7	0	0	0	0

Figure 17-19. Four-Time-Slice LCD Drive Waveform Examples (1/3 Bias Method)



Remark The waveforms for COM2 to SEG12 and COM3 to SEG12 are omitted.

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

With the 78K0/Dx2, a LCD drive power supply is generated using internal resistance division method.

The 78K0/Dx2 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 17-8** can be generated, without using external voltage divider resistors.

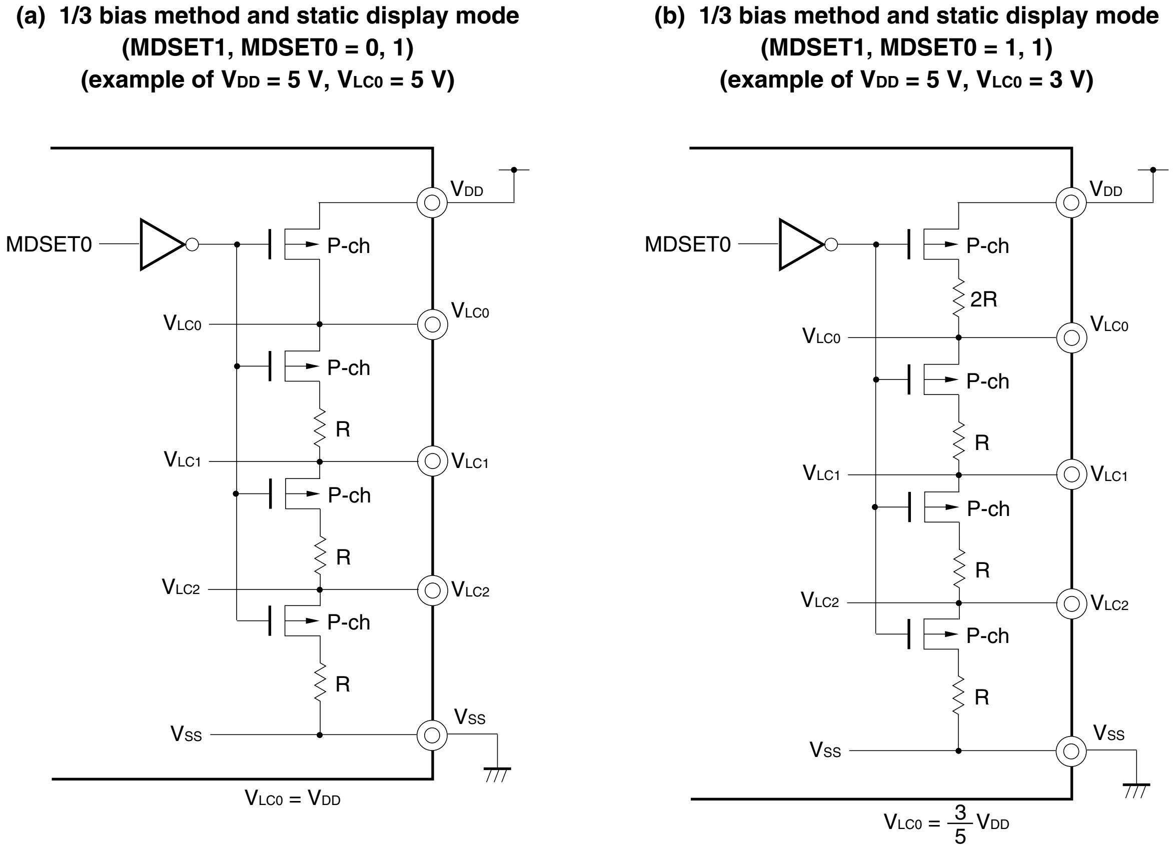
Table 17-8. LCD Drive Voltages (with On-Chip Voltage Divider Resistors)

Bias Method LCD Drive Voltage Pin	No Bias (Static)	1/3 Bias Method
V_{LC0}	V_{LCD}	V_{LCD}
V_{LC1}	$\frac{2}{3}V_{LCD}$	$\frac{2}{3}V_{LCD}$
V_{LC2}	$\frac{1}{3}V_{LCD}$	$\frac{1}{3}V_{LCD}$

<R>

Figure 17-20 shows examples of generating LCD drive voltages internally according to **Table 17-8**.

Figure 17-20. Examples of LCD Drive Power Connections



(a) 1/3 bias method and static display mode (MDSET1, MDSET0 = 0, 1) (example of V_{DD} = 5 V, V_{LC0} = 5 V)

(b) 1/3 bias method and static display mode (MDSET1, MDSET0 = 1, 1) (example of V_{DD} = 5 V, V_{LC0} = 3 V)

CHAPTER 18 SOUND GENERATOR

The Sound Generator generates an audio-frequency tone signal and a high-frequency pulse-width modulated (PWM) signal. The duty cycle of the PWM signal defines the volume.

By default, the two signal components are routed to separate pins. But both signals can also be combined to generate a composite signal that can be used to drive a loudspeaker circuit.

18.1 Overview

The Sound Generator consists of a programmable square wave tone generator and a programmable pulse-width modulator.

Features summary

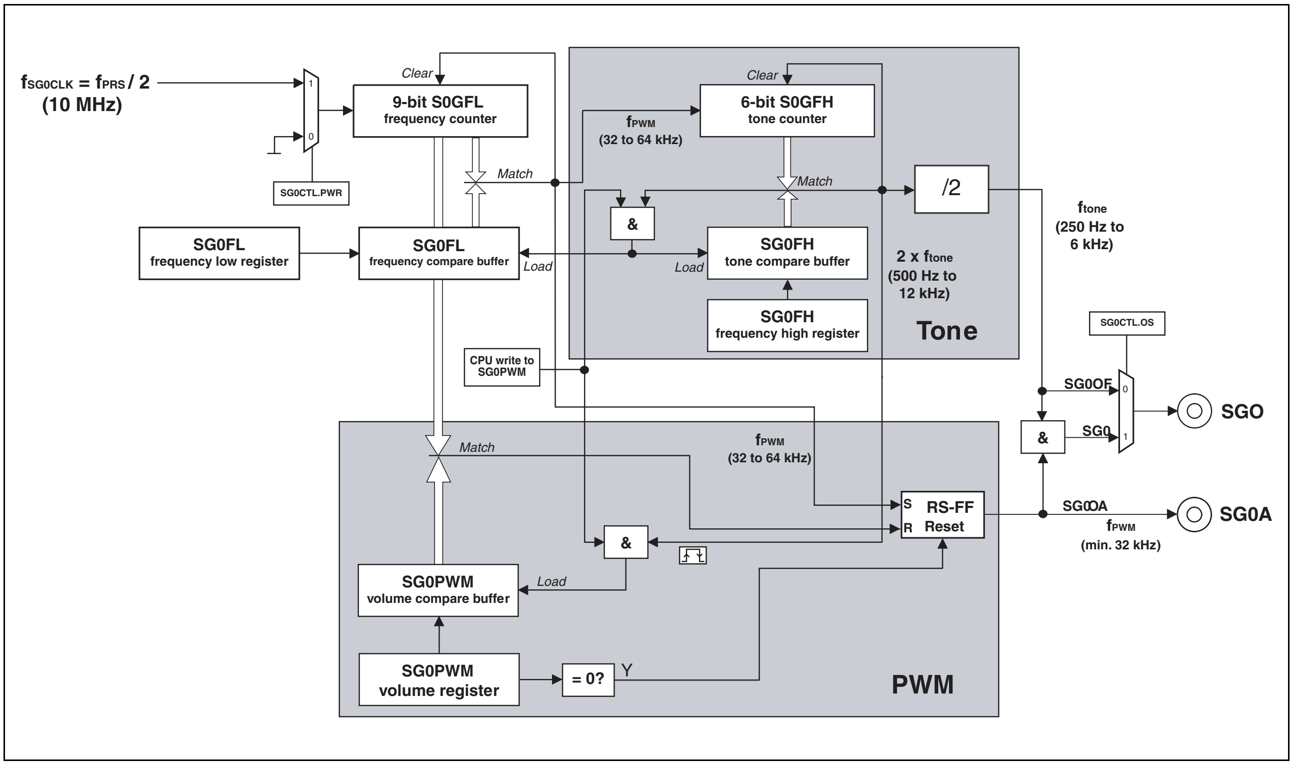
Special features of the Sound Generator are:

- Programmable tone frequency (250 Hz to 6 kHz with a minimum step size of 20 Hz)
- Programmable volume level (9 bit resolution)
- Wide range of PWM signal frequency (32 kHz to 64 kHz)
- Sound can be stopped or retriggered
- Composite or separated frequency/volume output for external circuitry variation
- Hardware-optimized update of frequency and volume to avoid audible artifacts

18.1.1 Description

The following figure provides a functional block diagram of the Sound Generator.

Figure 18-1. Sound Generator Block Diagram



The Sound Generator's input clock frequency f_{SG0CLK} is the 10 MHz clock which is divided the peripheral hardware clock oscillation frequency f_{PRS} in two.

Tone generator

The tone generator consists of two up-counters with compare registers. The values written to the frequency registers are automatically copied to compare buffers. The counters are reset to zero when their values match the contents of the associated compare buffers.

The 9-bit counter SG0FL generates a clock with a frequency between 32 kHz and 64 kHz. This clock constitutes the PWM frequency.

It is also the input of the second 6-bit counter SG0FH. The resulting tone signal behind the by-two-divider has a frequency between 250 Hz and 6 kHz and a 50 % duty cycle.

PWM

The PWM modulates the duty cycle according to the desired volume. It is controlled by the volume register SG0PWM. The value written to this register is automatically copied to the associated volume compare buffer.

The PWM continually compares the value of the counter SG0FL with the contents of its volume compare buffer.

The RS flipflop of the PWM is set by the pulses generated by the counter SG0FL. It is reset when the SG0FL counter value matches the contents of the volume buffer. Thus, the PWM output signal can have a duty cycle between 0 % (null volume) and 100 % (maximum volume).

The PWM frequency is above 32 kHz and hence outside the audible range.

Outputs

The Sound Generator is connected to the pins SGO and SGOA. By default, pin SGO provides the tone signal SG0OF and pin SGOA the PWM signal SG0OA that holds the volume (“amplitude”) information.

If bit SG0CTL.OS is set, pin SGO provides the composite signal SG0O that can directly control a speaker circuit.

18.1.2 Principle of operation

The software-controlled registers SG0FL, SG0FH, and SG0PWM are equipped with hardware buffers. The Sound Generator operates on these buffers.

This approach eliminates audible artifacts, because the buffers are only updated in synchronization with the generated tone waveform.

Remark This section provides an overview. For details please refer to **18.3 Sound Generator Operation**.

(1) Generation of the tone frequency

The tone frequency is determined by two counters and their associated compare register values. Two counters are necessary to keep the tone pulse and the PWM signal synchronized.

The first counter (SG0FL) provides the input to the second (SG0FH) and also to the PWM. It is used to keep the PWM frequency outside the audio range (above 30 kHz) and within the signal bandwidth of the external sound system (usually below 64 kHz). Its match value defines also the 100 % volume level.

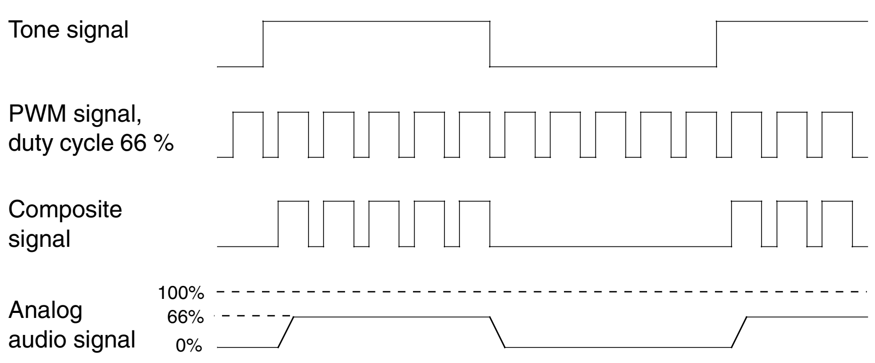
The second counter (SG0FH) generates the tone frequency (250 Hz to 6 kHz).

Remark If the target values of the counters SG0FL/SG0FH are changed to generate a different tone frequency, the volume register SG0PWM has to be adjusted to keep the same volume.

(2) Generation of the volume information

The volume information (the “amplitude” of the audible signal) is provided as a high-frequency PWM signal. In composite mode, the PWM signal is ANDed with the tone signal, as illustrated in the following figure.

Figure 18-2. Generation of the Composite Output Signal



After low-pass filtering, the analog signal amplitude corresponds to the duty cycle of the PWM signal. Low-pass filtering (averaging) is an inherent characteristic of a loudspeaker system.

The duty cycle can vary between 0 % and 100 %. Its generation is controlled by the counter register SG0FL and the volume register SG0PWM.

When the volume register SG0PWM is cleared, the sound stops immediately.

18.2 Sound Generator Registers

The Sound Generator is controlled by means of the following registers:

Table 18-1. Sound Generator Registers Overview

Register name	Shortcut	Address
SG0 control register	SG0CTL	FFA5H
SG0 frequency low register	SG0FL	FFA8H
SG0 frequency high register	SG0FH	FFAAH
SG0 volume register	SG0PWM	FFA6H

(1) SG0 control register (SG0CTL)

The 8-bit SG0CTL register controls the operation of the Sound Generator.

This register can be read/written in 8-bit or 1-bit units.

This register is cleared by any reset.

Figure 18-3. Format of SG0 Control Register (SG0CTL)

Address: FFA5H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SG0CTL	0	0	0	PWR	0	0	OS	0

PWR	Power save mode selection
0	Clock input switched off (the Sound Generator is disabled and does not operate).
1	Clock input switched on (the Sound Generator is enabled and ready to use).

OS	SG0 output mode selection
0	Selects SGOF and SGOA outputs (frequency and amplitude separated).
1	Selects SGO output (frequency and amplitude mixed).

Cautions 1. Bit 0 must be set to 0.

2. Change the contents of this register only when the sound is stopped (register SG0PWM cleared).

(2) SG0 frequency low register (SG0FL)

The 16-bit SG0FL register is used to specify the target value for the PWM frequency. It holds the target value for the 9-bit counter SG0FL.

This register can be read/written in 16-bit units. It cannot be written if bit SG0CTL.PWR = 0.

This register is cleared by any reset.

Address: FFA8H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SG0FL	0	0	0	0	0	0	0	Counter SG0FL target value								

For the calculation of the resulting PWM frequency refer to **18.3.2 (2) PWM calculations**.

The value written to SG0FL defines also the reference value for the maximum sound amplitude (100% PWM duty cycle). A 100 % duty cycle (continually high) will be generated if the SG0PWM value is higher than the SG0FL value. For details see **18.3.2 (2) PWM calculations**.

Remarks 1. The bits SG0FL[15:9] are not used.

2. The maximum value to be written is 510 (01FEH). This yields a PWM frequency of 19.7 kHz in case of the sound generator input clock SG0CLK 10 MHz. The minimum value to be written depends on the capability of the external circuit. A value of 255 (00FFH) would yield a PWM frequency of 39.1 kHz in case of the sound generator input clock SG0CLK 10 MHz.

3. The value read from this register does not necessarily reflect the current PWM frequency, because this frequency is determined by the frequency compare buffer value. The buffer might not be updated yet.

For details see **18.3.1 (1) Updating the frequency buffer values**.

(3) SG0 frequency high register (SG0FH)

The 16-bit SG0FH register is used to specify the final tone frequency. It holds the target value for the 6-bit counter SG0FH.

This register can be read/written in 16-bit units. It cannot be written if bit SG0CTL.PWR = 0.

This register is cleared by any reset.

Address: FFAAH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SG0FH	0	0	0	0	0	0	0	0	0	0	Counter SG0FH target value					

For the calculation of the resulting tone frequency refer to **18.3.1 (2) Tone frequency calculation**.

Remarks 1. The bits SG0FH[15:6] are not used.

2. Legal values depend on the contents of register SG0FL which defines the frequency of the input pulse. For example: If the counter SG0FL generates a frequency of 32.4 kHz, a value of 63 would generate a tone frequency of 253 Hz.

3. The value read from this register does not necessarily reflect the current tone frequency, because this frequency is determined by the frequency compare buffer value. The buffer might not be updated yet.

For details see **18.3.1 (1) Updating the frequency buffer values**.

(4) SG0 volume register (SG0PWM)

The 16-bit register SG0PWM is used to specify the sound volume. It holds the target value for the sound amplitude that is given by the duty cycle of the PWM signal.

This register can be read/written in 16-bit units. It cannot be written if bit SG0CTL.PWR = 0.

This register is cleared by any reset.

Address: FFA6H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SG0PWM	0	0	0	0	0	0	0	Sound volume target value								

The value written to this register must be considered in conjunction with the contents of register SG0FL. The register SG0FL specifies the maximum value of the counter SG0FL.

For the calculation of the resulting duty cycle refer to **18.3.2 (2) PWM calculations**.

The setting takes effect after the SG0PWM buffer has been updated (see **18.3.2 (1) Updating the volume buffer value**).

Remarks 1. The bits SG0PWM[15:9] are not used.

2. The value read from this register does not necessarily reflect the current volume, because the value of counter SG0FL is compared with the contents of the volume buffer. The buffer might not be updated yet.

3. The sound stops immediately when this register is cleared.

18.3 Sound Generator Operation

This section explains the details of the Sound Generator.

18.3.1 Generating the tone

The tone signal is generated by the compare match signal of the SG0FH counter value with the value of the SG0FH buffer, followed by a by-two-divider. At each compare match, the counter is reset to zero.

Remember that the SG0FH counter is clocked by the output of the SG0FL counter.

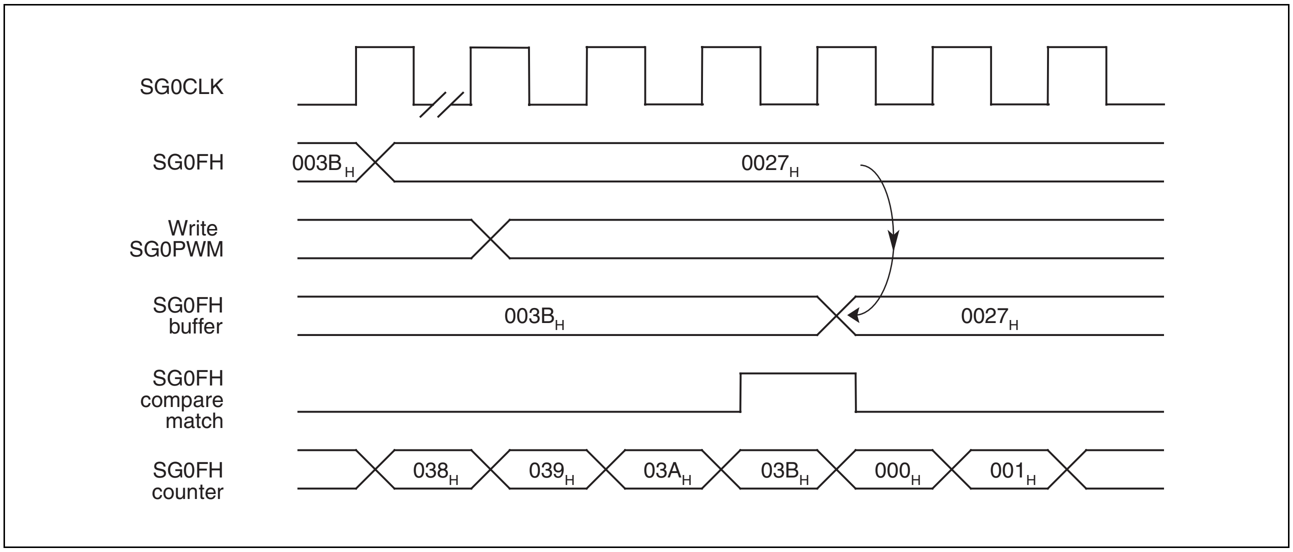
(1) Updating the frequency buffer values

The values of the frequency buffers can be changed by writing to the associated frequency registers SG0FL and SG0FH.

Changing the value of the SG0FL (equivalent to SG0F[15:0]) register would also yield a change of the PWM frequency, i.e. the sound volume. Therefore it is obligatory to write the correct PWM value to SG0PWM before a new SG0FL value is copied to the frequency buffers.

The following figure shows an example (not to scale).

Figure 18-4. Update Timing of the Frequency Buffers



If SG0FL is set to 01AEH and a 193 Hz tone is generated, as in the above example, the time span between writing to the SG0PWM register and updating the buffer can be up to 5.17 ms.

(2) Tone frequency calculation

The tone frequency can be calculated as:

$f_{tone} = f_{SG0CLK}$ / (([SG0FL buffer] + 1) × ([SG0FH buffer] + 1) × 2)

where:

f_{SG0CLK} : Frequency of Sound Generator's input clock

$f_{SG0CLK} = f_{PRS} / 2$

[SG0FL buffer] : Contents of the SG0FL buffer

[SG0FH buffer] : Contents of the SG0FH buffer

Example

If:

- f_{PRS} = 20 MHz
- $f_{SG0CLK} = f_{PRS} / 2$ = 10 MHz
- [SG0FL buffer] = 255 (00FFH) (this yields a PWM frequency of 39.01 kHz)
- [SG0FH buffer] = 32 (0020H)

then:

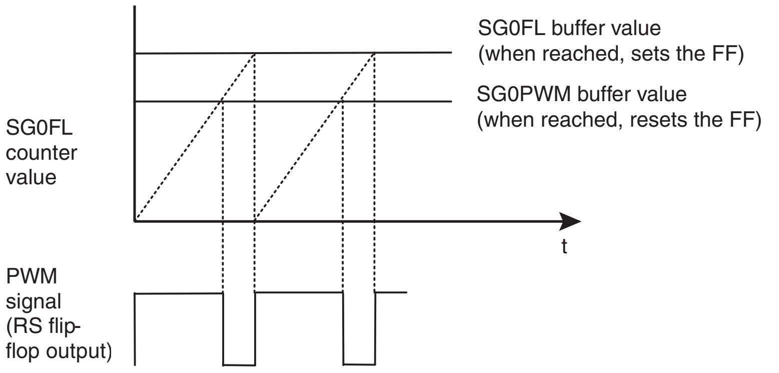
- f_{tone} = 592 Hz

Remark Note that the buffer contents can differ from the contents of the associated register until the next compare match.

18.3.2 Generating the volume information

The sound volume information is generated by comparing the SG0FL counter value with the contents of the SG0PWM volume buffer. An RS flipflop is set when the counter matches the SG0FL buffer and reset when the counter reaches the value of the volume buffer SG0PWM.

Figure 18-5. PWM Signal Generation


SG0FL buffer value
(when reached, sets the FF)
SG0PWM buffer value
(when reached, resets the FF)
SG0FL counter value
t
PWM signal (RS flip-flop output)

The duty cycle of the PWM signal is determined by the difference between the contents of the SG0FL counter buffer and the contents of the SG0PWM volume buffer. The larger the difference, the smaller the duty cycle.

The PWM signal is continually high when the value of the volume buffer is higher than the value of the frequency compare buffer.

Remark To achieve 100 % duty cycle for all PWM frequencies, SGOFL must not be set to a value above 1FEH.

The PWM signal is continually low when the value of the volume buffer is zero—the sound has stopped.

(1) Updating the volume buffer value

The value of the volume compare buffer can be changed by writing to the volume register SG0PWM.

- If the register is cleared by writing 0000H, the register value is copied to the volume compare buffer with the next rising edge of SG0CLK.
- As a result, the sound stops at the latest after one period of SG0CLK.
- If a non-zero value is written to the register, the buffer is updated with the next falling or rising edge of the tone frequency (match between SG0FH counter value and SG0FH buffer value).

(2) PWM calculations

PWM frequency

The PWM frequency is generated by the counter SG0FL. It can be calculated as:

$f_{PWM} = f_{SG0CLK}$ / (([SG0FL buffer] + 1)

where:

f_{SG0CLK} : Frequency of Sound Generator's input clock

$f_{SG0CLK} = f_{PRS} / 2$

[SG0FL buffer] : Contents of the SG0FL buffer

Duty cycle

The duty cycle of the PWM signal is calculated as follows:

- If [SG0PWM buffer] > [SG0FL buffer]:
  Duty cycle = 100 %
- If 0 ≤ [SG0PWM buffer] ≤ [SG0FL buffer]:
  Duty cycle = [SG0PWM buffer] / ([SG0FL buffer] + 1)

where:

[SG0PWM buffer] : Contents of SG0PWM buffer

[SG0FL buffer] : Contents of SG0FL buffer

Example

If [SG0FL] is set to 240 (00F0H), the following table applies:

Table 18-2. Duty Cycle Calculation Example

[SG0PWM]	Calculation	Duty cycle [%]
01FFH		100
...		100
00F1H	241 / 241	100
00F0H	240 / 241	99.6
00EFH	239 / 241	99.2
...	...	...
0001H	1 / 241	0.41
0000H	0 / 241	0

The table shows, how the contents of register SG0FL affects the achievable volume resolution.

18.4 Sound Generator Application Hints

This section provides supplementary programming information.

18.4.1 Initialization

To enable the Sound Generator, set SG0CTL.PWR to 1. This connects the SG0 to the clock SG0CLK.

Check bit SG0CTL.OS.

When SG0CTL.OS is 0, the signal at pin SGO is a symmetrical square waveform with the frequency f_{tone}. When SG0CTL.OS is 1, the signal at pin SGO is composed of the tone signal and PWM pulses.

The frequency data registers SG0FL and SG0FH provide the buffer values for the counters. The combined value represents the frequency of the tone.

18.4.2 Start and stop sound

The sound is started by writing a non-zero value to the volume register SG0PWM.

Before starting the sound, all other register settings must be made.

The sound is stopped by writing 0000H to the volume register SG0PWM. The sound is stopped regardless of the current value of amplitude output or frequency output. Thus, the sound can be stopped quickly, even if a very low sound frequency is chosen.

18.4.3 Change sound volume

The sound volume is changed by writing a new value to register SG0PWM.

The new volume takes effect with the next edge of the tone pulse (rising or falling).

18.4.4 Generate special sounds

To generate special sounds (like blinker clicks etc.), frequency and volume can be changed simultaneously.

To change the frequency of a sound that has already started:

1. Write to the frequency registers SG0FL and SG0FH separately in 16-bit mode.
2. Write to the volume register SG0PWM.

CHAPTER 19 INTERRUPT FUNCTIONS

19.1 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 19-1**.

A standby release signal is generated and STOP and HALT modes are released.

8 external interrupt requests and 27 internal interrupt requests are provided as maskable interrupts.

(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.

19.2 Interrupt Sources and Configuration

A total of 36 interrupt sources exist for maskable and software interrupts. In addition, they also have up to four reset sources (see **Table 19-1**).

Table 19-1. Interrupt Source List (1/3)

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	Pin input edge detection	External	0008H	(B)
		INTIIC0	End of IIC0 communication	Internal		(A)
	3	INTP3	Pin input edge detection	External	000AH	(B)
		INTP4	Pin input edge detection			
	4	INTTP0CC0	Match between TMP0 and CCR0 (when compare register is specified), TIOP00 pin valid edge detection (when capture register is specified)	Internal	000CH	(A)
	5	INTTP0CC1	Match between TMP0 and CCR1 (when compare register is specified), TIOP01 pin valid edge detection (when capture register is specified)		000EH	
	6	INTTP1CC0	Match between TMP1 and CCR0 (when compare register is specified), TIOP10 pin valid edge detection (when capture register is specified)		0010H	
	7	INTTP1CC1	Match between TMP1 and CCR1 (when compare register is specified), TIOP11 pin valid edge detection (when capture register is specified)		0012H	
	8	INTC0ERR	AFCAN0 error occurrence		0014H	
		INTC0WUP	AFCAN0 wakeup			
	9	INTC0REC	AFCAN0 reception completion		0016H	
	10	INTC0TRX	AFCAN0 transmission completion		0018H	
	11	INTPR60	UART60 pin input edge detection		001AH	
	12	INTSR60	End of UART60 reception/ UART60 reception error		001CH	
	13	INTST60	End of UART60 transmission		001EH	
	14	INTP2	Pin input edge detection	External	0020H	(B)
		INTCSI10	End of CSI10 transmission	Internal		(A)

Notes 1. The default priority is the priority applicable when two or more maskable interrupt are generated simultaneously. 0 is the highest priority, and 28 is the lowest.

2. Basic configuration types (A) to (C) correspond to (A) to (C) in **Figure 19-1**.

3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is set to 0.

Table 19-1. Interrupt Source List (2/3)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Maskable	15	INTPR61	UART61 pin input edge detection	Internal	0022H	(A)
	16	INTSR61	End of UART61 reception/ UART61 reception error		0024H	
	17	INTST61	End of UART61 transmission		0026H	
	18	INTCSI11	End of CSI11 communication		0028H	
	19	INTTM50	Match between TM50 and CR50 (when compare register is specified)		002AH	
		INTTP2OV	TMP2 overflow			
	20	INTTP2CC0	Match between TMP2 and CCR0 (when compare register is specified), TIOP20 pin valid edge detection (when capture register is specified)		002CH	
	21	INTTP2CC1	Match between TMP2 and CCR1 (when compare register is specified), TIOP21 pin valid edge detection (when capture register is specified)		002EH	
	22	INTAD	End of A/D conversion		0030H	
	23	INTWTI	Watch timer reference time interval signal		0032H	
		INTWT	Watch timer overflow			
<R>	24	INTTM51	Match between TM51 and CR51 (when compare register is specified)		0034H	
		INTTP3OV	TMP3 overflow			
	25	INTTP3CC0	Match between TMP3 and CCR0 (when compare register is specified), TIOP30 pin valid edge detection (when capture register is specified)		0036H	
	26	INTTP3CC1	Match between TMP3 and CCR1 (when compare register is specified), TIOP31 pin valid edge detection (when capture register is specified)		0038H	
	27	INTTP4CC0	Match between TMP4 and CCR0 (when compare register is specified), TIOP40 pin valid edge detection (when capture register is specified)		003AH	
	28	INTTP4CC1	Match between TMP4 and CCR1 (when compare register is specified), TIOP41 pin valid edge detection (when capture register is specified)		003CH	

Notes 1. The default priority is the priority applicable when two or more maskable interrupt are generated simultaneously. 0 is the highest priority, and 28 is the lowest.

2. Basic configuration types (A) to (C) correspond to (A) to (C) in **Figure 19-1**.

Table 19-1. Interrupt Source List (3/3)

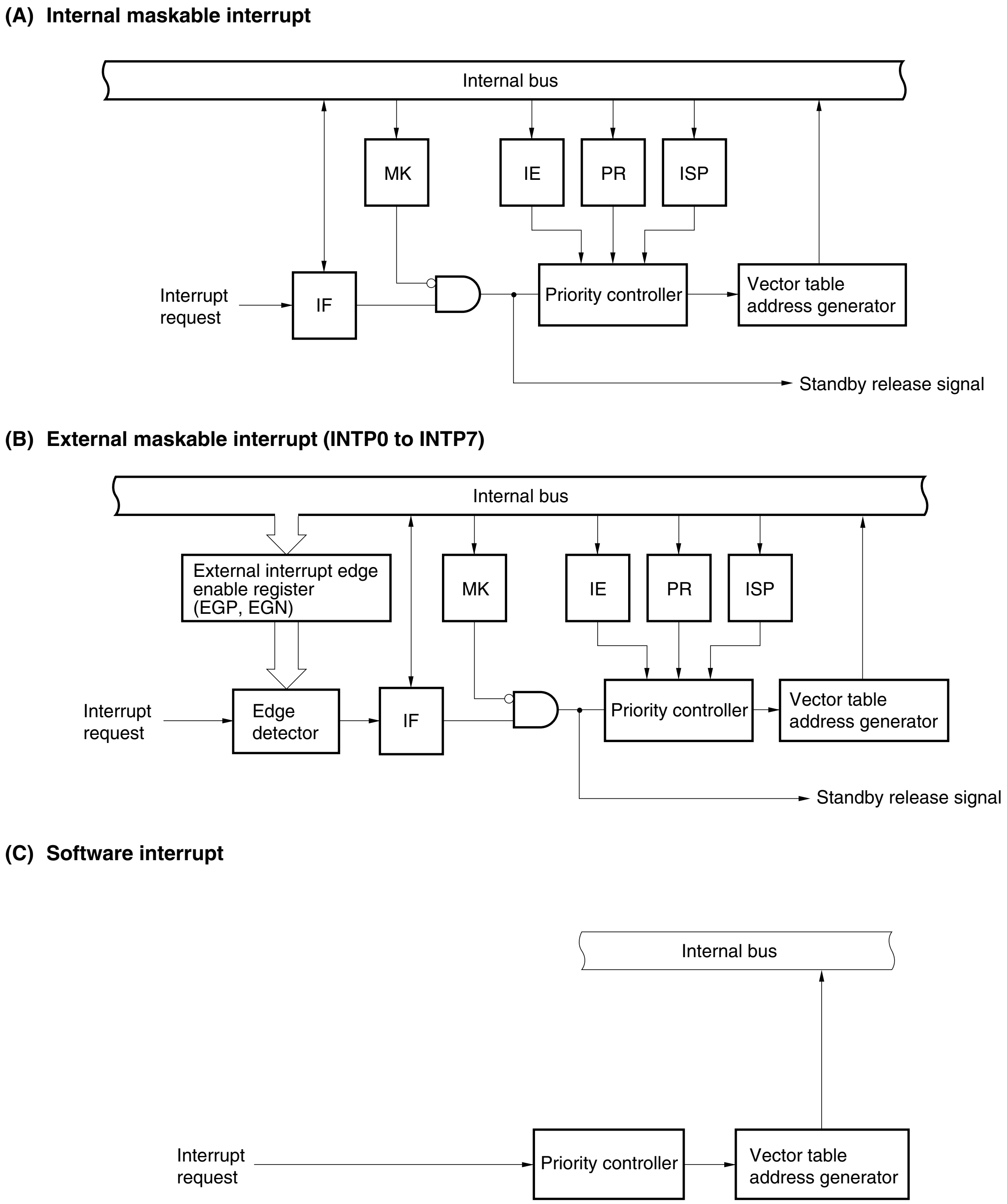
Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Software	–	BRK	BRK instruction execution	–	003EH	(C)
Non-maskable	–	INTCK2	Address break generation before OCD execution	–	0002H	–
Reset	–	RESET	Reset input	–	0000H	–
		POC	Power-on clear			
		LVI	Low-voltage detection[Note 3]			
		WDT	WDT overflow			

Notes 1. The default priority is the priority applicable when two or more maskable interrupt are generated simultaneously. 0 is the highest priority, and 28 is the lowest.

2. Basic configuration types (A) to (C) correspond to (A) to (C) in **Figure 19-1**.

3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is set to 1.

Figure 19-1. Basic Configuration of Interrupt Function



(A) Internal maskable interrupt

(B) External maskable interrupt (INTP0 to INTP7)

(C) Software interrupt

IF:	Interrupt request flag
IE:	Interrupt enable flag
ISP:	In-service priority flag
MK:	Interrupt mask flag
PR:	Priority specification flag

19.3 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 19-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 19-2. Flags Corresponding to Interrupt Request Sources

Interrupt Request	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	DUALIF0 Note 1		PMK1	DUALMK0 Note 2		PPR1	DUALPR0 Note 2	
INTIIC0	IICIF0			IICMK0			IICPR0		
INTP3	PIF3	DUALIF1 Note 1		PMK3	DUALMK1 Note 2		PPR3	DUALPR1 Note 2	
INTP4	PIF4			PMK4			PPR4		
INTTP0CC0	TP0CC0IF			TP0CC0MK			TP0CC0PR		
INTTP0CC1	TP0CC1IF			TP0CC1MK			TP0CC1PR		
INTTP1CC0	TP1CC0IF			TP1CC0MK			TP1CC0PR		
INTTP1CC1	TP1CC1IF			TP1CC1MK			TP1CC1PR		
INTC0ERR	C0ERRIF	DUALIF2 Note 1	IF0H	C0ERRMK	DUALMK2 Note 2	MK0H	C0ERRPR	DUALPR2 Note 2	PR0H
INTC0WUP	C0WUPIF			C0WUPMK			C0WUPPR		
INTC0REC	C0RECIF			C0RECMK			C0RECPR		
INTC0TRX	C0TRXIF			C0TRXMK			C0TRXPR		
INTPR60	PIF60			PMK60			PPR60		
INTSR60	SRIF60			SRMK60			SRPR60		
INTST60	STIF60			STMK60			STPR60		
INTP2	PIF2	DUALIF3 Note 1		PMK2	DUALMK3 Note 2		PPR2	DUALPR3 Note 2	
INTCSI10	CSIIF10			CSIMK10			CSIPR10		
INTPR61	PIF61			PMK61			PPR61		
INTSR61	SRIF61		IF1L	SRMK61		MK1L	SRPR61		PR1L
INTST61	STIF61			STMK61			STPR61		
INTCSI11	CSIIF11			CSIMK11			CSIPR11		
INTTM50	TMIF50	DUALIF4 Note 1		TMMK50	DUALMK4 Note 2		TMPR50	DUALPR4 Note 2	
INTTP2OV	TP2OVIF			TP2OVMK			TP2OVPR		
INTTP2CC0	TP2CC0IF			TP2CC0MK			TP2CC0PR		
INTTP2CC1	TP2CC1IF			TP2CC1MK			TP2CC1PR		
INTAD	ADIF			ADMK			ADPR		
INTWT	WTIF	DUALIF5 Note 1		WTMK	DUALMK5 Note 2		WTPR	DUALPR5 Note 2	
INTWTI	WTIIF			WTIMK			WTIPR		
<R> INTTM51	TMIF51	DUALIF6 Note 1	IF1H	TMMK51	DUALMK6 Note 2	MK1H	TMPR51	DUALPR6 Note 2	PR1H
INTTP3OV	TP3OVIF			TP3OVMK			TP3OVPR		
INTTP3CC0	TP3CC0IF			TP3CC0MK			TP3CC0PR		
INTTP3CC1	TP3CC1IF			TP3CC1MK			TP3CC1PR		
INTTP4CC0	TP4CC0IF			TP4CC0MK			TP4CC0PR		
INTTP4CC1	TP4CC1IF			TP4CC1MK			TP4CC1PR		

Notes 1. If either of the two types of interrupt sources is generated, these flags are set (1).

2. Both types of interrupt sources are supported.

(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 read with a 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 19-2. 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	TP1CC1IF	TP1CC0IF	TP0CC1IF	TP0CC0IF	DUALIF1 PIF3 PIF4	DUALIF0 PIF1 IICIF0	PIF0	LVIIF

Address: FFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	PIF61	DUALIF3 PIF2 CSIIF10	STIF60	SRIF60	PIF60	C0TRXIF	C0RECIF	DUALIF2 C0ERRIF C0WUPIF

Address: FFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	DUALIF5 WTIF WTIIF	ADIF	TP2CC1IF	TP2CC0IF	DUALIF4 TMIF50 TP2OVIF	CSIIF11	STIF61	SRIF61

Address: FFE3H After reset: 00H R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
IF1H	0	0	0	TP4CC1IF	TP4CC0IF	TP3CC1IF	TP3CC0IF	DUALIF6 TMIF51 TP3OVIF

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Cautions 1. Be sure to set 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 with a 16-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 19-3. 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	TP1CC1MK	TP1CC0MK	TP0CC1MK	TP0CC0MK	DUALMK1 PMK3 PMK4	DUALMK0 PMK1 IICMK0	PMK0	LVIMK

Address: FFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	PMK61	DUALMK3 PMK2 CSIMK10	STMK60	SRMK60	PMK60	C0TRXMK	C0RECMK	DUALMK2 C0ERRMK C0WUPMK

Address: FFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	DUALMK5 WTMK WTIMK	ADMK	TP2CC1MK	TP2CC0MK	DUALMK4 TMMK50 TP2OVMK	CSIMK11	STMK61	SRMK61

Address: FFE7H After reset: FFH R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
MK1H	1	1	1	TP4CC1MK	TP4CC0MK	TP3CC1MK	TP3CC0MK	DUALMK6 TMMK51 TP3OVMK

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

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 with a 16-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 19-4. 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	TP1CC1PR	TP1CC0PR	TP0CC1PR	TP0CC0PR	DUALPR1 PPR3 PPR4	DUALPR0 PPR1 IICPR0	PPR0	LVIPR

Address: FFE9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR0H	PPR61	DUALPR3 PPR2 CSIPR10	STPR60	SRPR60	PPR60	C0TRXPR	C0RECPR	DUALPR2 C0ERRPR C0WUPPR

Address: FFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR1L	DUALPR5 WTPR WTIPR	ADPR	TP2CC1PR	TP2CC0PR	DUALPR4 TMPR50 TP2OVPR	CSIPR11	STPR61	SRPR61

Address: FFEBH After reset: FFH R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
PR1H	1	1	1	TP4CC1PR	TP4CC0PR	TP3CC1PR	TP3CC0PR	DUALPR6 TMPR51 TP3OVPR

XXPRX	Priority level selection
0	High priority level
1	Low priority level

Caution Be sure to set bit 5 to 7 of PR1H to 1.

(4) External interrupt rising edge enable register (EGP), external interrupt falling edge enable register (EGN)

<R> These registers specify the valid edge for INTP0 to INTP4, INTP6, INTP7.
EGP and EGN are set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears these registers to 00H.

Figure 19-5. 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
<R> EGP	EGP7	EPG6	0	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
<R> EGN	EGN7	EGN6	0	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 7)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

<R> **Caution Be sure to set bit 5 of EGP and bit 5 of EGN to 0.**

Table 19-3 shows the ports corresponding to EGPn and EGNn.

<R> **Table 19-3. Ports Corresponding to EGPn and EGNn**

Detection Enable Register		Edge Detection Port	External Request Signal
EGP0	EGN0	P17	INTP0
EGP1	EGN1	P60	INTP1
EGP2	EGN2	P12	INTP2
EGP3	EGN3	P61	INTP3
EGP4	EGN4	P10	INTP4
EGP6	EGN6	P14 (ISC7 = 0) P70 (ISC7 = 1)	INTPR60
EGP7	EGN7	P11	INTPR61

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.

<R> **Remark** n = 0 to 4, 6, 7

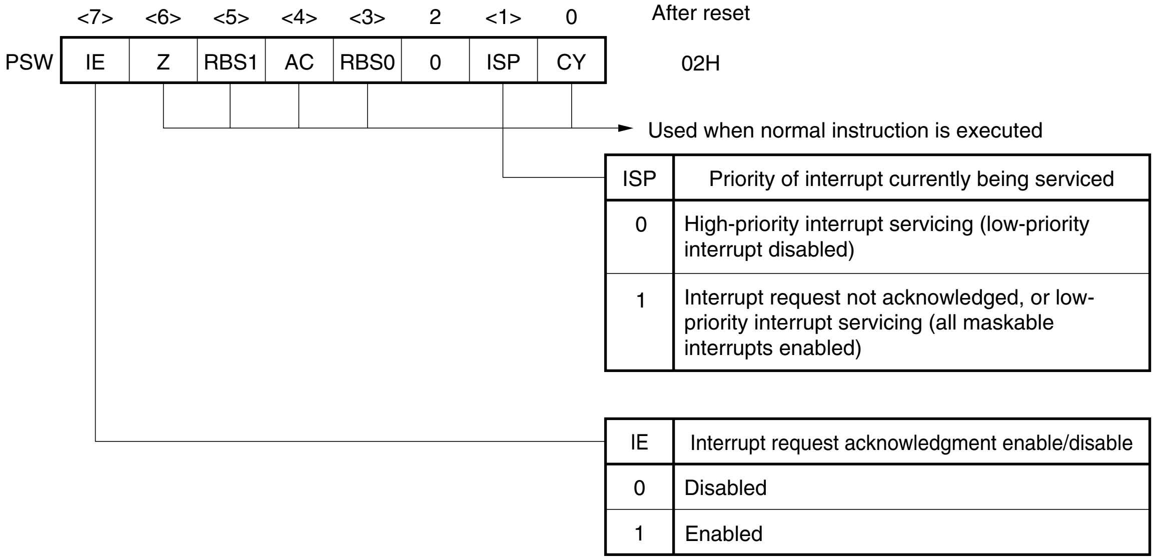
(5) Program status word (PSW)

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 19-6. 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

19.4 Interrupt Servicing Operations

19.4.1 Maskable interrupt acknowledgement

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 interrupt servicing is performed are listed in **Table 19-4** below.

For the interrupt request acknowledgement timing, see **Figures 19-8** and **19-9**.

Table 19-4. 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.

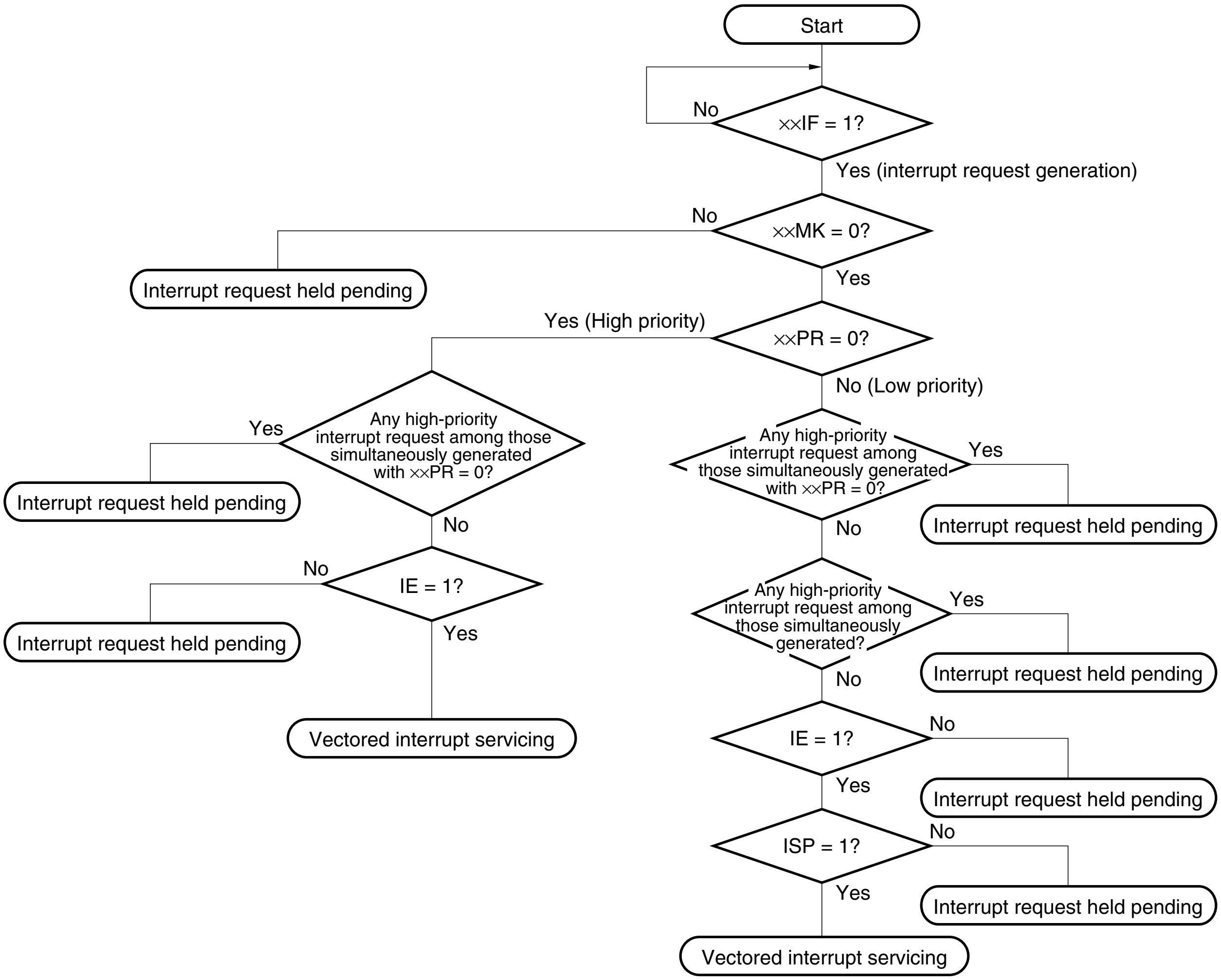
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 19-7 shows the interrupt request acknowledgement 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 19-7. Interrupt Request Acknowledgement Processing Algorithm


Start
××IF = 1?
No
Yes (interrupt request generation)
××MK = 0?
No
Interrupt request held pending
Yes
××PR = 0?
Yes (High priority)
No (Low priority)
Any high-priority interrupt request among those simultaneously generated with ××PR = 0?
Yes
Interrupt request held pending
No
IE = 1?
No
Interrupt request held pending
Yes
Vectored interrupt servicing
Any high-priority interrupt request among those simultaneously generated with ××PR = 0?
Yes
Interrupt request held pending
No
Any high-priority interrupt request among those simultaneously generated?
Yes
Interrupt request held pending
No
IE = 1?
No
Interrupt request held pending
Yes
ISP = 1?
No
Interrupt request held pending
Yes
Vectored interrupt servicing

××IF: Interrupt request flag

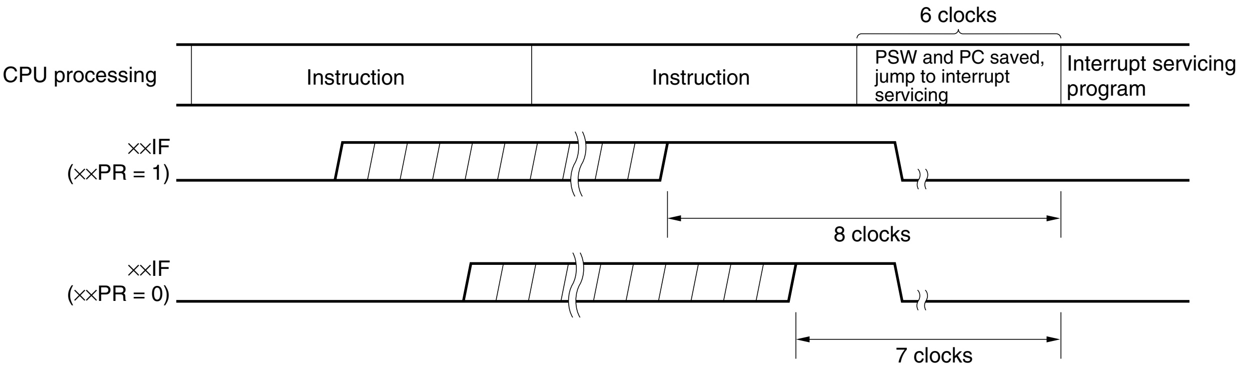
××MK: Interrupt mask flag

××PR: Priority specification flag

IE: Flag that controls acknowledgement of maskable interrupt request (1 = Enable, 0 = Disable)

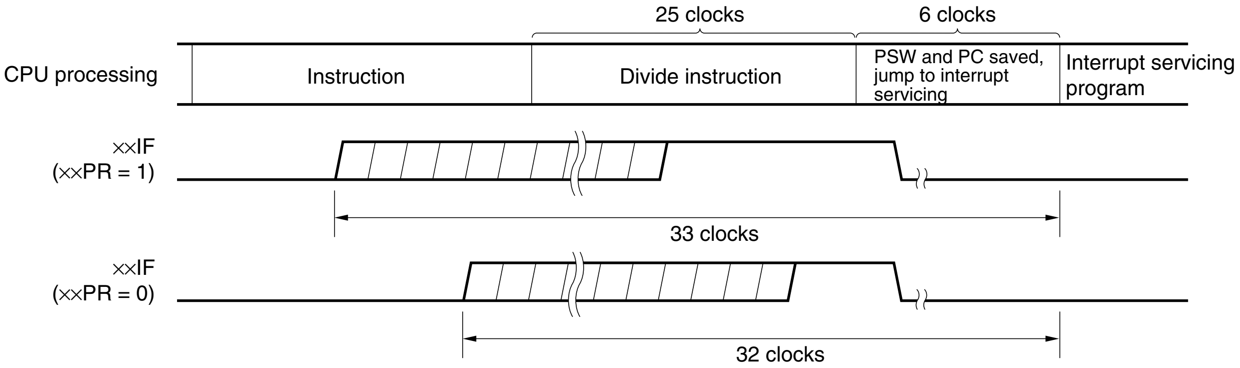
ISP: Flag that indicates the priority level of the interrupt currently being serviced (0 = high-priority interrupt servicing, 1 = No interrupt request acknowledged, or low-priority interrupt servicing)

Figure 19-8. Interrupt Request Acknowledgement Timing (Minimum Time)



Remark 1 clock: 1/fCPU (fCPU: CPU clock)

Figure 19-9. Interrupt Request Acknowledgement Timing (Maximum Time)



Remark 1 clock: 1/fCPU (fCPU: CPU clock)

19.4.2 Software interrupt request acknowledgement

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.

19.4.3 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 acknowledgement enabled state is selected (IE = 1). When an interrupt request is acknowledged, interrupt request acknowledgement 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 acknowledgement.

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 19-5 shows relationship between interrupt requests enabled for multiple interrupt servicing and **Figure 19-10** shows multiple interrupt servicing examples.

Table 19-5. 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. : Mu○ltiple 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 acknowledgement is disabled.

IE = 1: Interrupt request acknowledgement is enabled.

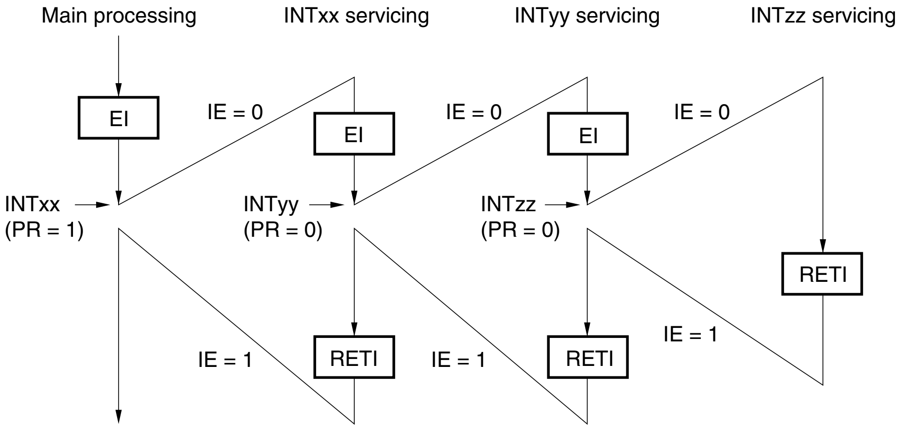
4. PR is a flag contained in PR0L, PR0H, PR1L, and PR1H.

PR = 0: Higher priority level

PR = 1: Lower priority level

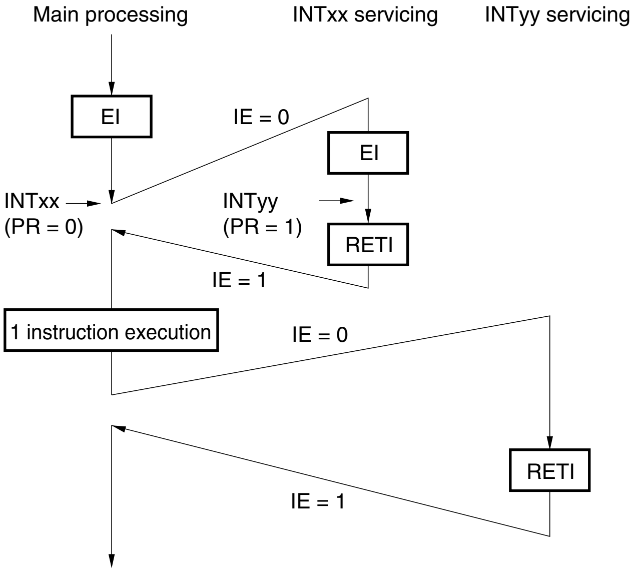
Figure 19-10. Examples of Multiple Interrupt Servicing (1/2)

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

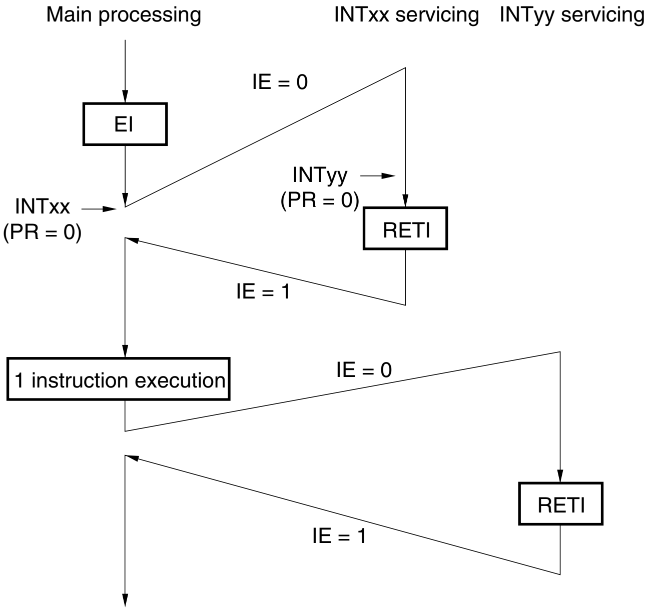


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level
PR = 1: Lower priority level
IE = 0: Interrupt request acknowledgment disabled

Figure 19-10. 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 acknowledgement disabled

19.4.4 Interrupt request hold

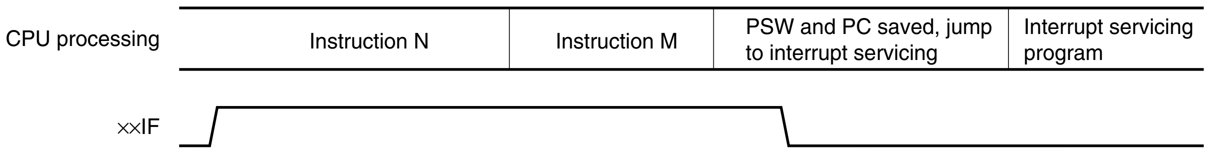
There are instructions where, even if an interrupt request is issued for them while another instruction is being executed, request acknowledgement is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- 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 19-11 shows the timing at which interrupt requests are held pending.

Figure 19-11. Interrupt Request Hold


CPU processing
Instruction N
Instruction M
PSW and PC saved, jump to interrupt servicing
Interrupt servicing program
××IF

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 (instruction request).

CHAPTER 20 STANDBY FUNCTION

20.1 Standby Function and Configuration

20.1.1 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.

(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, 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 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.

20.1.2 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 20-1. 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

MOST 11	MOST 13	MOST 14	MOST 15	MOST 16	Oscillation stabilization time status				
						fx = 2 MHz	fx = 5 MHz	fx = 10 MHz	fx = 20 MHz
0	0	0	0	0	Less than 2^{11}/fx	Less than 1.02 ms	Less than 409.6 µs	Less than 204.8 µs	Less than 102.4 µs
1	0	0	0	0	2^{11}/fx min.	1.02 ms min.	409.6 µs min.	204.8 µs min.	102.4 µs min.
1	1	0	0	0	2^{13}/fx min.	4.10 ms min.	1.64 ms min.	819.2 µs min.	409.6 µs min.
1	1	1	0	0	2^{14}/fx min.	8.19 ms min.	3.27 ms min.	1.64 ms min.	819.2 µs min.
1	1	1	1	0	2^{15}/fx min.	16.38 ms min.	6.55 ms min.	3.27 ms min.	1.64 ms min.
1	1	1	1	1	2^{16}/fx min.	32.77 ms min.	13.11 ms min.	6.55 ms min.	3.27 ms min.

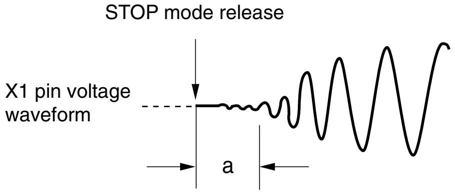
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST11 and remain 1.

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 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 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 20-2. 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 = 2 MHz	fx = 5 MHz	fx = 10 MHz	fx = 20 MHz
0	0	1	2^{11}/fx	1.02 ms	409.6 μs	204.8 μs	102.4 μs
0	1	0	2^{13}/fx	4.10 ms	1.64 ms	819.2 μs	409.6 μs
0	1	1	2^{14}/fx	8.19 ms	3.27 ms	1.64 ms	819.2 μs
1	0	0	2^{15}/fx	16.38 ms	6.55 ms	3.27 ms	1.64 ms
1	0	1	2^{16}/fx	32.77 ms	13.11 ms	6.55 ms	3.27 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.

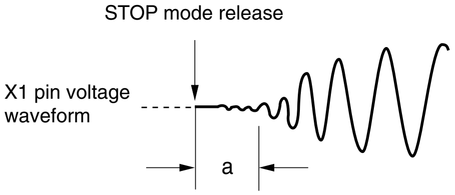
2. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

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 time does not include the time until clock oscillation starts (“a” below).



Remark fx: X1 clock oscillation frequency

20.2 Standby Function Operation

20.2.1 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 20-1. 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_{OSC8})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EXT})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{OSC8}	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_{EXT}	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_{EXTS}	Status before HALT mode was set is retained		
	f_{OSC}		Status before HALT mode was set is retained. Operation continues (cannot be stopped) when “internal low-speed oscillator cannot be stopped by software” is set by option byte.		
CPU			Operation stopped		
Flash memory			Operation stopped		
RAM			Status before HALT mode was set is retained		
Regulator			Operates in normal mode		
Port (latch)			Status before HALT mode was set is retained		
16-bit timer/event counter	TMP0		Operable		
	TMP1				
	TMP2				
	TMP3				
	TMP4				
8-bit timer/event counter	TM50				
	TM51				
Watch timer					
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					
A/D converter					
Serial interface	UART60				
	UART61				
	CSI10				
	CSI11				
	IIC0				
FCAN controller					
Multiplier/divider					
Stepper motor C/D (with ZPD)					
LCD C/D					
Sound generator					
Power-on-clear function					
Low-voltage detection function					
External interrupt					

Table 20-1. Operating Statuses in HALT Mode (2/2)

HALT Mode Setting Item			When HALT Instruction Is Executed While CPU Is Operating on Subsystem Clock	
			When CPU Is Operating on XT1 Clock (f_{XT})	When CPU Is Operating on External Subsystem Clock (f_{EXTS})
System clock			Clock supply to the CPU is stopped	
	Main system clock	f_{OSC8}	Status before HALT mode was set is retained	
		f_X		
		f_{EXT}	Operates or stops by external clock input	
	Subsystem clock	f_{XT}	Operation continues (cannot be stopped)	Status before HALT mode was set is retained
		f_{EXTS}	Operates or stops by external clock input	Operation continues (cannot be stopped)
	f_{OSC}		Status before HALT mode was set is retained. Operation continues (cannot be stopped) when “internal low-speed oscillator cannot be stopped by software” is set by option byte.	
CPU			Operation stopped	
Flash memory			Operation stopped	
RAM			Status before HALT mode was set is retained	
Regulator			Operates in low power consumption mode when f_{OSC8} stops Operates in normal mode when f_{OSC8} operates	
Port (latch)			Status before HALT mode was set is retained	
16-bit timer/event counter	TMP0[Note]		Operable	
	TMP1[Note]			
	TMP2[Note]			
	TMP3[Note]			
	TMP4[Note]			
8-bit timer/event counter	TM50[Note]			
	TM51[Note]			
Watch timer				
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	
A/D converter				
Serial interface	UART60		Operable	
	UART61			
	CSI10[Note]			
	CSI11[Note]			
	IIC0			
FCAN controller				
Multiplier/divider				
Stepper motor C/D (with ZPD)				
LCD C/D				
Sound generator				
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_{OSC8}:	Internal high-speed oscillation clock
f_X:	X1 clock
f_{EXT}:	External main system clock
f_{XT}:	XT1 clock
f_{EXTS}:	External subsystem clock
f_{OSC}:	Internal low-speed oscillation clock

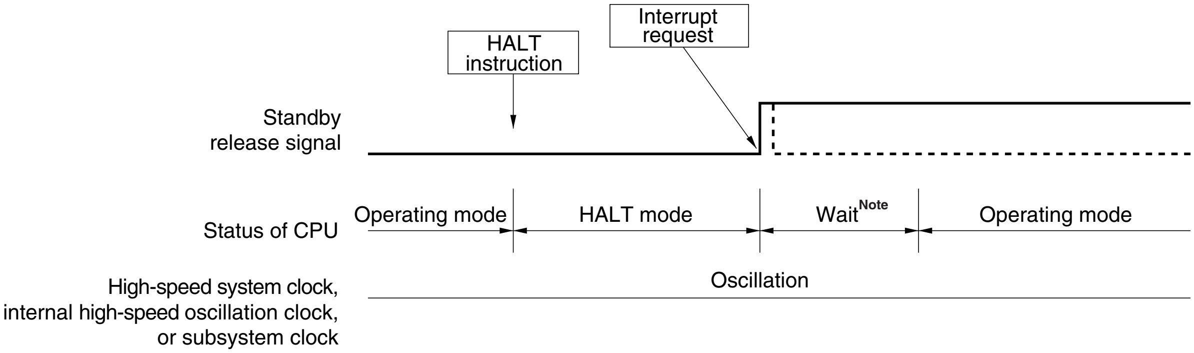
(2) HALT mode release

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 acknowledgement is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgement is disabled, the next address instruction is executed.

Figure 20-3. HALT Mode Release by Interrupt Request Generation



Note The wait time is as follows:

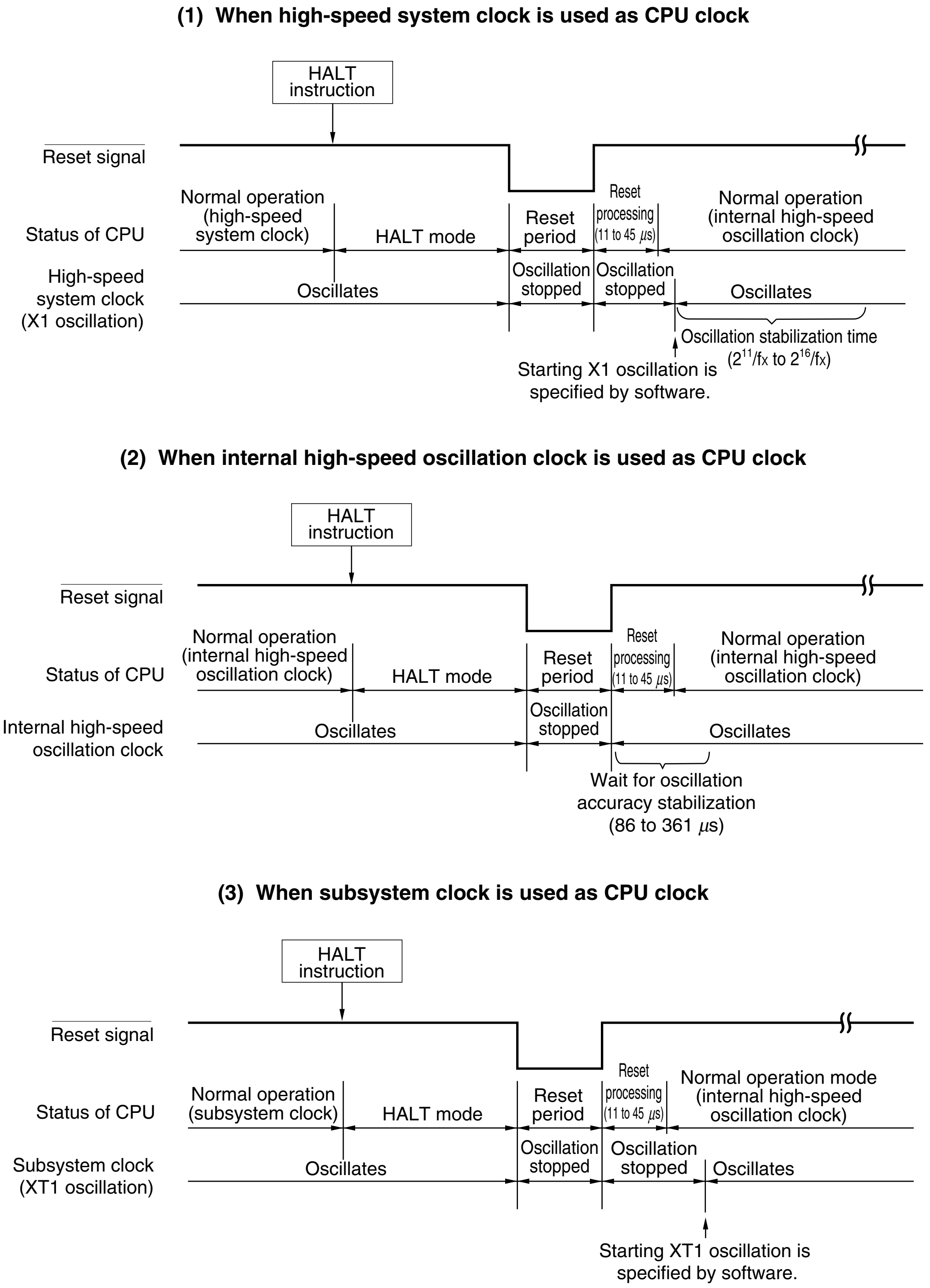
- When vectored interrupt servicing is carried out: 8 or 9 clocks
- When vectored interrupt servicing is not carried out: 2 or 3 clocks

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.

Figure 20-4. HALT Mode Release by Reset



(1) When high-speed system clock is used as CPU clock

HALT instruction
Reset signal
Status of CPU
Normal operation (high-speed system clock)
HALT mode
Reset period
Reset processing (11 to 45 μs)
Normal operation (internal high-speed oscillation clock)
High-speed system clock (X1 oscillation)
Oscillates
Oscillation stopped
Oscillation stopped
Oscillates
Oscillation stabilization time $(2^{11}/fx$ to $2^{16}/fx)$
Starting X1 oscillation is specified by software.

(2) When internal high-speed oscillation clock is used as CPU clock

HALT instruction
Reset signal
Status of CPU
Normal operation (internal high-speed oscillation clock)
HALT mode
Reset period
Reset processing (11 to 45 μs)
Normal operation (internal high-speed oscillation clock)
Internal high-speed oscillation clock
Oscillates
Oscillation stopped
Oscillates
Wait for oscillation accuracy stabilization (86 to 361 μs)

(3) When subsystem clock is used as CPU clock

HALT instruction
Reset signal
Status of CPU
Normal operation (subsystem clock)
HALT mode
Reset period
Reset processing (11 to 45 μs)
Normal operation mode (internal high-speed oscillation clock)
Subsystem clock (XT1 oscillation)
Oscillates
Oscillation stopped
Oscillation stopped
Oscillates
Starting XT1 oscillation is specified by software.

Remark fx: X1 clock oscillation frequency

Table 20-2. 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 signal input	−	−	×	×	Reset processing

×: don’t care

20.2.2 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 20-3. Operating Statuses in STOP Mode

STOP Mode Setting			When STOP Instruction Is Executed While CPU Is Operating on Main System Clock		
Item			When CPU Is Operating on Internal High-Speed Oscillation Clock (f_{OSC8})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EXT})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{OSC8}	Stopped		
		f_X			
		f_{EXT}	Input invalid		
	Subsystem clock	f_{XT}	Status before STOP mode was set is retained		
		f_{EXTS}	Status before STOP mode was set is retained		
	f_{OSC}		Status before STOP mode was set is retained. Operation continues (cannot be stopped) when “internal low-speed oscillator cannot be stopped by software” is set by option byte.		
CPU			Operation stopped		
Flash memory			Operation stopped		
RAM			Status before STOP mode was set is retained		
Regulator			Operates in low power consumption mode		
Port (latch)			Status before STOP mode was set is retained		
16-bit timer/event counter	TMP0[Note]		Operation stopped		
	TMP1[Note]				
	TMP2[Note]				
	TMP3[Note]				
	TMP4				
8-bit timer/event counter	TM50[Note]		Operable only when TI50, f_{OSC} base is selected as the count clock		
	TM51[Note]		Operable only when TI51 is selected as the count clock		
Watch timer			Operable only when f_{SUB} 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 f_{SUB} is selected as the count clock		
Buzzer output			Operation stopped		
A/D converter					
Serial interface	UART60		Operable only when TO50 is selected as the serial clock during 8-bit timer/event counter TM50 operation		
	UART61				
	CSI10[Note]		Operable only when external clock is selected as the serial clock		
	CSI11[Note]				
	IIC0				
FCAN controller			Operable. Can be woken up from sleep mode.		
Stepper motor C/D (with ZPD)			Operation stopped		
LCD C/D			Operable only when f_{SUB}, f_{OSC} is selected as the operation clock		
Sound Generator			Operation stopped		
Power-on-clear function			Operable		
Low-voltage detection function					
External interrupt					

Note Do not start operation of these functions on the external clock input from peripheral hardware pins in the stop mode.

Remark

f_{OSC8}:	Internal high-speed oscillation clock
f_X:	X1 clock
f_{EXT}:	External main system clock
f_{XT}:	XT1 clock
f_{EXTS}:	External subsystem clock
f_{OSC}:	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.

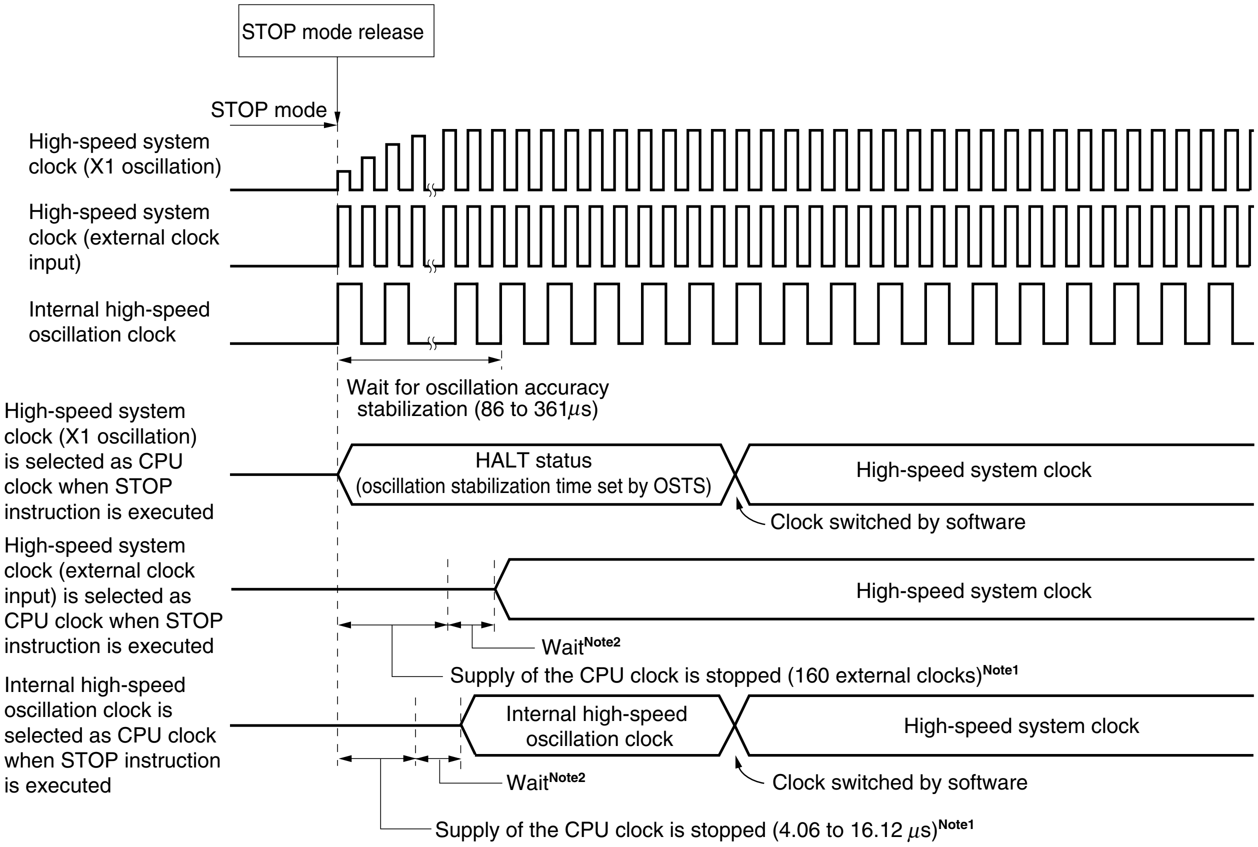
2. Even if “internal low-speed oscillator can be stopped by software” is selected by the option byte, the internal low-speed oscillator continues in the STOP mode in the status before the STOP mode is set. To stop the internal low-speed oscillator in the STOP mode, stop it by software and then execute the STOP instruction.

3. To shorten oscillation stabilization time after the STOP mode is released when the CPU operates with the high-speed system clock (X1 oscillation), temporarily switch the CPU clock to the internal high-speed oscillator internal oscillation Clock before the next execution of the STOP instruction. Before changing the CPU clock from the internal high-speed oscillator 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).

4. If the STOP instruction is executed when AMPH = 1, supply of the CPU clock is stopped for 4.06 to 16.12 μs after the STOP mode is released when the internal high-speed oscillation clock is selected as the CPU clock, or for the duration of 160 external clocks when the high-speed system clock (external clock input) is selected as the CPU clock.

(2) STOP mode release

Figure 20-5. Operation Timing When STOP Mode Is Released
(When Unmasked Interrupt Request Is Generated)


STOP mode release
STOP mode
High-speed system clock (X1 oscillation)
High-speed system clock (external clock input)
Internal high-speed oscillation clock
Wait for oscillation accuracy stabilization (86 to 361μs)
High-speed system clock (X1 oscillation) is selected as CPU clock when STOP instruction is executed
HALT status (oscillation stabilization time set by OSTS)
High-speed system clock
Clock switched by software
High-speed system clock (external clock input) is selected as CPU clock when STOP instruction is executed
High-speed system clock
Wait$^{\text{Note2}}$
Supply of the CPU clock is stopped (160 external clocks)$^{\text{Note1}}$
Internal high-speed oscillation clock is selected as CPU clock when STOP instruction is executed
Internal high-speed oscillation clock
High-speed system clock
Wait$^{\text{Note2}}$
Clock switched by software
Supply of the CPU clock is stopped (4.06 to 16.12 μs)$^{\text{Note1}}$

Notes 1. When AMPH = 1

2. 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

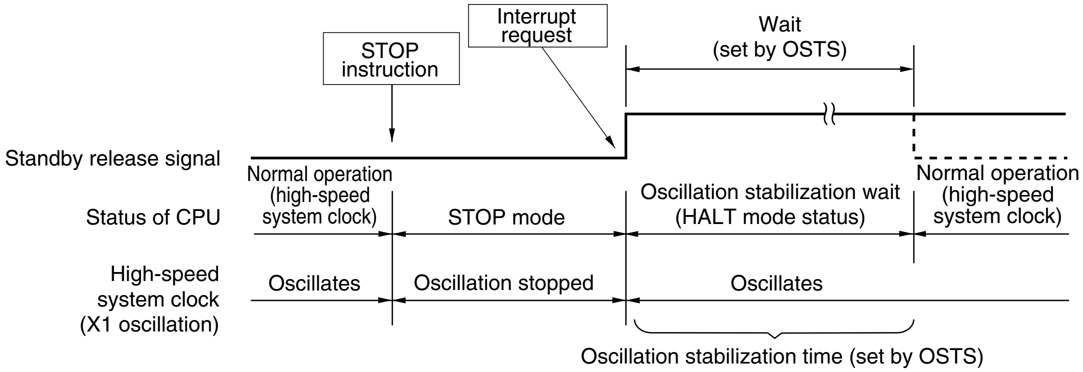
The STOP mode can be released by the following two sources.

(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.

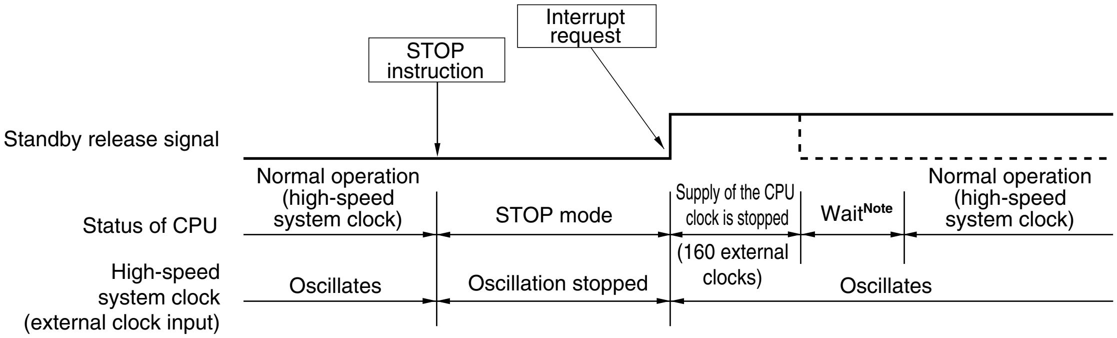
Figure 20-6. STOP Mode Release by Interrupt Request Generation (1/2)

(1) When high-speed system clock (X1 oscillation) is used as CPU clock


Interrupt request
STOP instruction
Wait (set by OSTS)
Standby release signal
Normal operation (high-speed system clock)
Status of CPU
STOP mode
Oscillation stabilization wait (HALT mode status)
Normal operation (high-speed system clock)
High-speed system clock (X1 oscillation)
Oscillates
Oscillation stopped
Oscillates
Oscillation stabilization time (set by OSTS)

(2) When high-speed system clock (external clock input) is used as CPU clock (1/2)

- When AMPH = 1


Interrupt request
STOP instruction
Standby release signal
Normal operation (high-speed system clock)
Status of CPU
STOP mode
Supply of the CPU clock is stopped
(160 external clocks)
Wait^Note
Normal operation (high-speed system clock)
High-speed system clock (external clock input)
Oscillates
Oscillation stopped
Oscillates

Note The wait time is as follows:

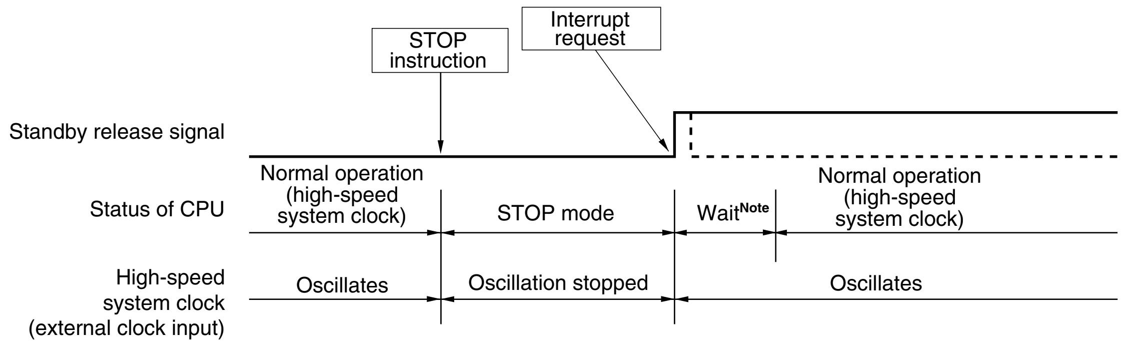
• When vectored interrupt servicing is carried out:	8 or 9 clocks
• When vectored interrupt servicing is not carried out:	2 or 3 clocks

Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

Figure 20-6. STOP Mode Release by Interrupt Request Generation (2/2)

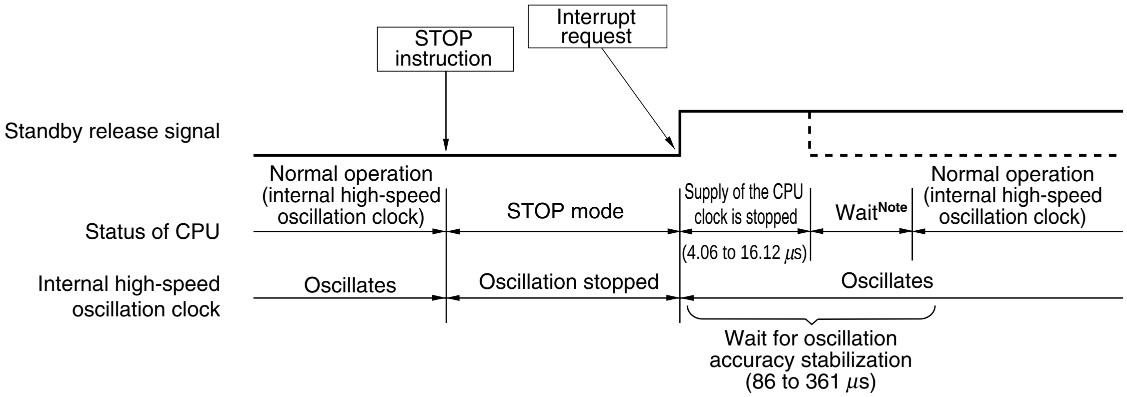
(2) When high-speed system clock (external clock input) is used as CPU clock (2/2)

- When AMPH = 0

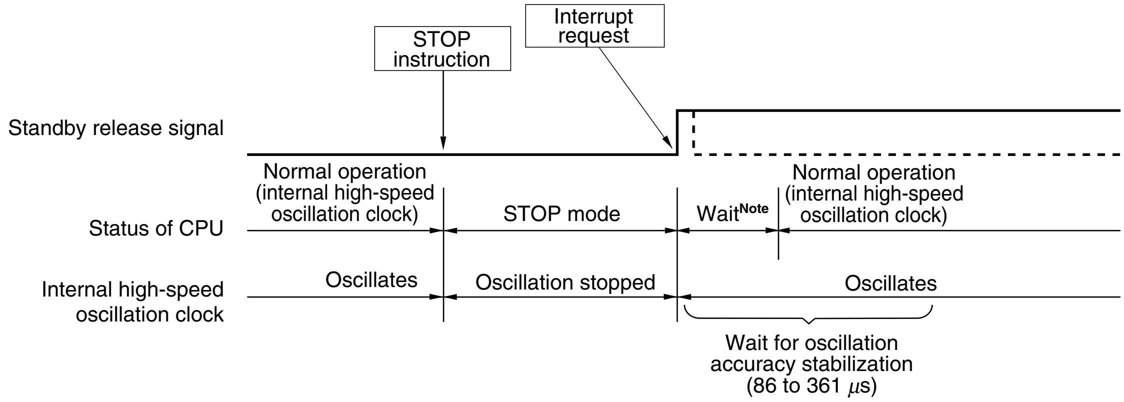

STOP instruction
Interrupt request
Standby release signal
Status of CPU
Normal operation (high-speed system clock)
STOP mode
Wait Note
Normal operation (high-speed system clock)
High-speed system clock (external clock input)
Oscillates
Oscillation stopped
Oscillates

(3) When internal high-speed oscillation clock is used as CPU clock

- When AMPH = 1


STOP instruction
Interrupt request
Standby release signal
Status of CPU
Normal operation (internal high-speed oscillation clock)
STOP mode
Supply of the CPU clock is stopped
(4.06 to 16.12 µs)
Wait Note
Normal operation (internal high-speed oscillation clock)
Internal high-speed oscillation clock
Oscillates
Oscillation stopped
Oscillates
Wait for oscillation accuracy stabilization (86 to 361 µs)

- When AMPH = 0


STOP instruction
Interrupt request
Standby release signal
Status of CPU
Normal operation (internal high-speed oscillation clock)
STOP mode
Wait Note
Normal operation (internal high-speed oscillation clock)
Internal high-speed oscillation clock
Oscillates
Oscillation stopped
Oscillates
Wait for oscillation accuracy stabilization (86 to 361 µs)

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

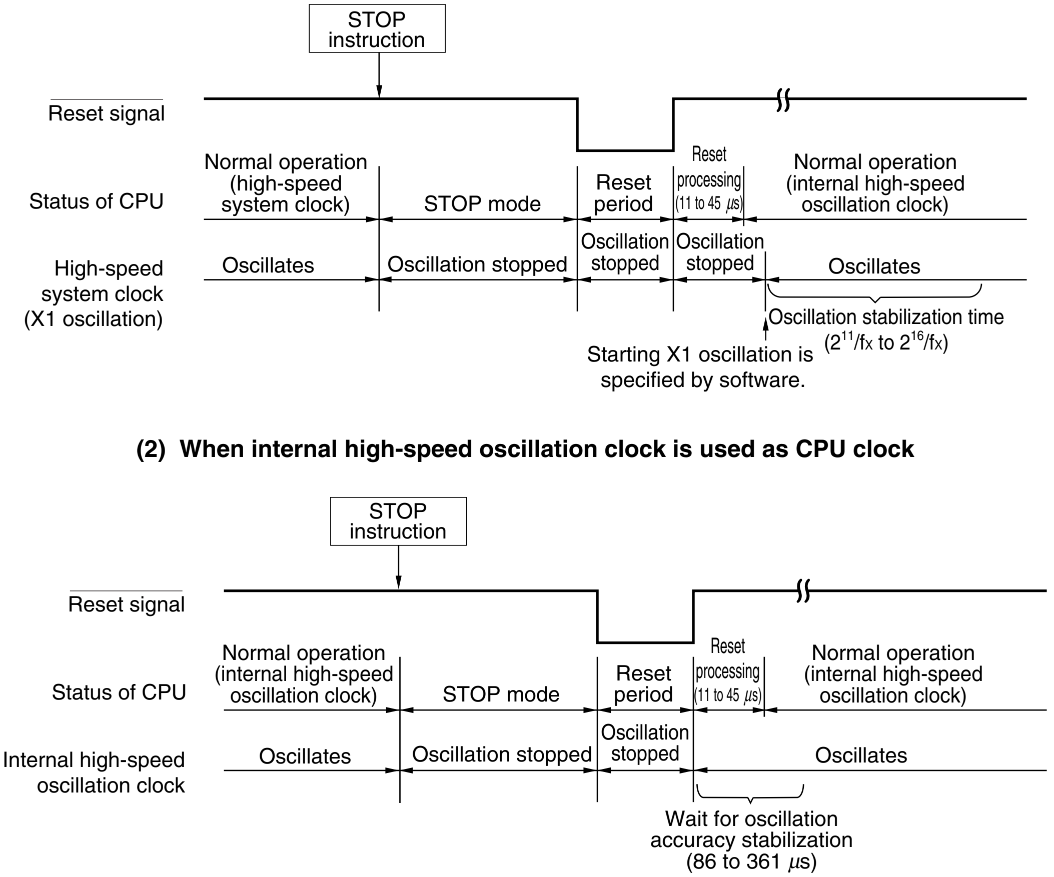
Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(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.

Figure 20-7. 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

STOP instruction
Reset signal
Status of CPU
Normal operation (high-speed system clock)
STOP mode
Reset period
Reset processing (11 to 45 μs)
Normal operation (internal high-speed oscillation clock)
High-speed system clock (X1 oscillation)
Oscillates
Oscillation stopped
Oscillation stopped
Oscillation stopped
Oscillates
Oscillation stabilization time (2^11/fx to 2^16/fx)
Starting X1 oscillation is specified by software.
STOP instruction
Reset signal
Status of CPU
Normal operation (internal high-speed oscillation clock)
STOP mode
Reset period
Reset processing (11 to 45 μs)
Normal operation (internal high-speed oscillation clock)
Internal high-speed oscillation clock
Oscillates
Oscillation stopped
Oscillation stopped
Oscillates
Wait for oscillation accuracy stabilization (86 to 361 μs)

Remark fx: X1 clock oscillation frequency

Table 20-4. 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 signal input	–	–	×	×	Reset processing

×: don’t care

CHAPTER 21 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 21-1** and **21-2**. Each pin is high impedance during reset signal generation or during the oscillation stabilization time just after a reset release.

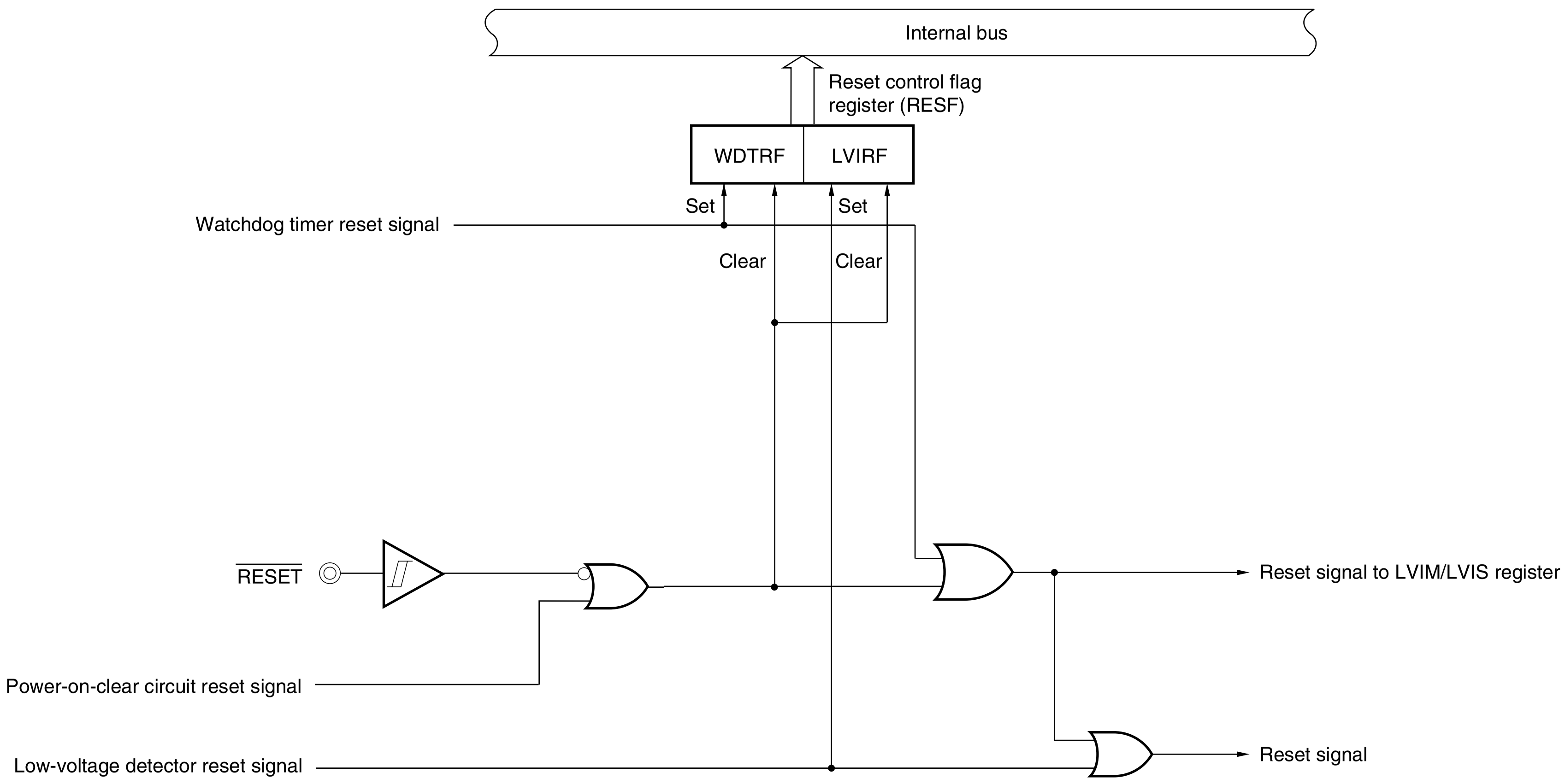
When a low level is input to the $\overline{\text{RESET}}$ pin, the device is reset. It is released from the reset status when a high level is input to the $\overline{\text{RESET}}$ pin and program execution is started with the internal high-speed oscillation clock after reset processing. A reset by the watchdog timer is automatically released, and program execution starts using the internal high-speed oscillation clock (see **Figures 21-2** to **21-4**) 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 23 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 24 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 and external subsystem clock input become 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 21-1. Block Diagram of Reset Function


Internal bus
Reset control flag
register (RESF)
WDTRF
LVIRF
Set
Set
Clear
Clear
Watchdog timer reset signal
RESET
Power-on-clear circuit reset signal
Low-voltage detector reset signal
Reset signal to LVIM/LVIS register
Reset signal

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

Figure 21-2. Timing of Reset by $\overline{\text{RESET}}$ Input

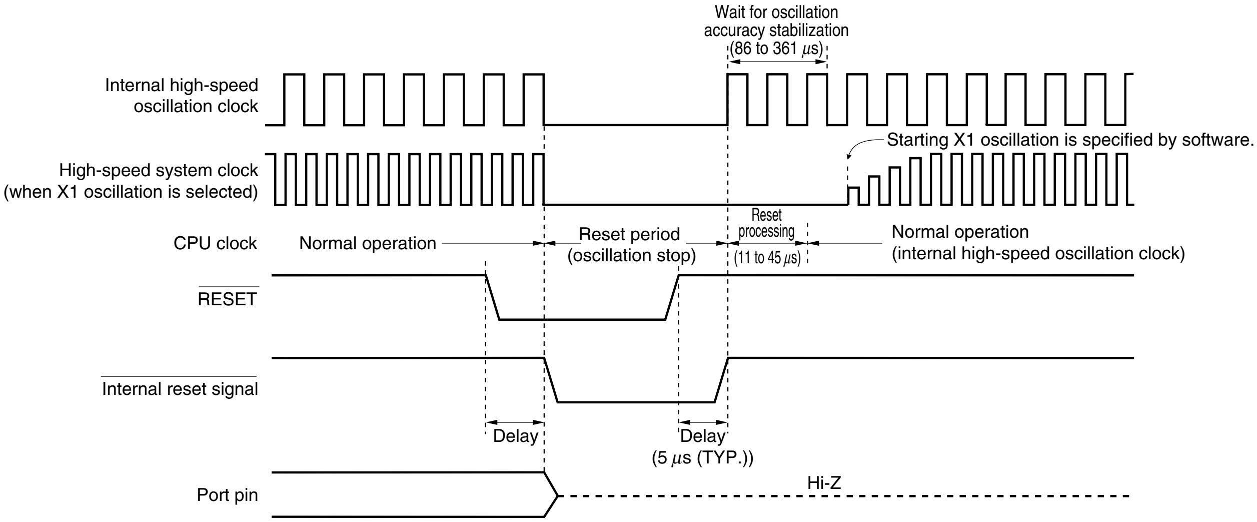
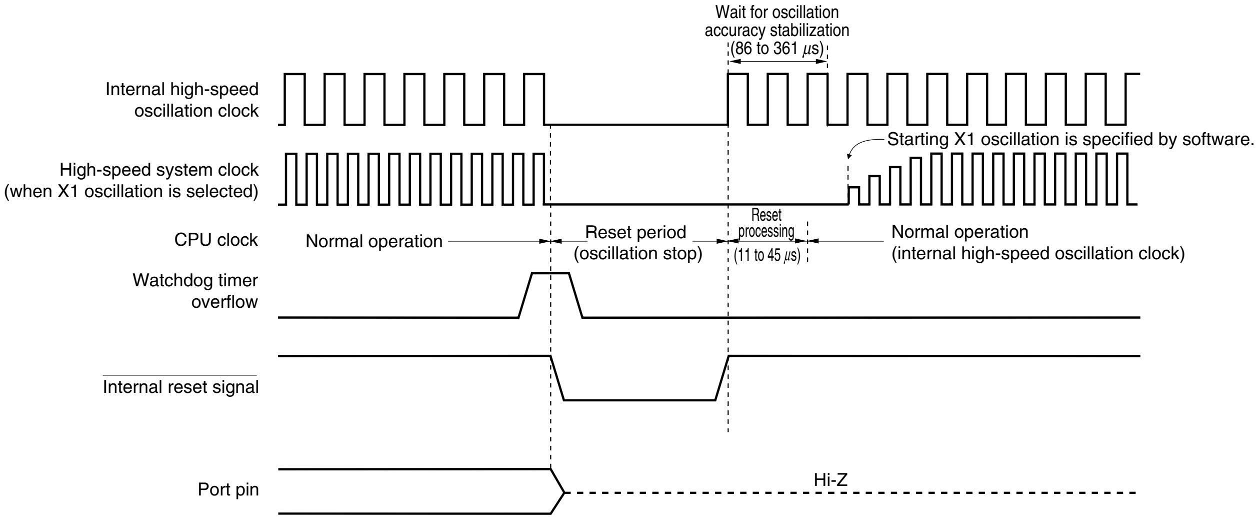
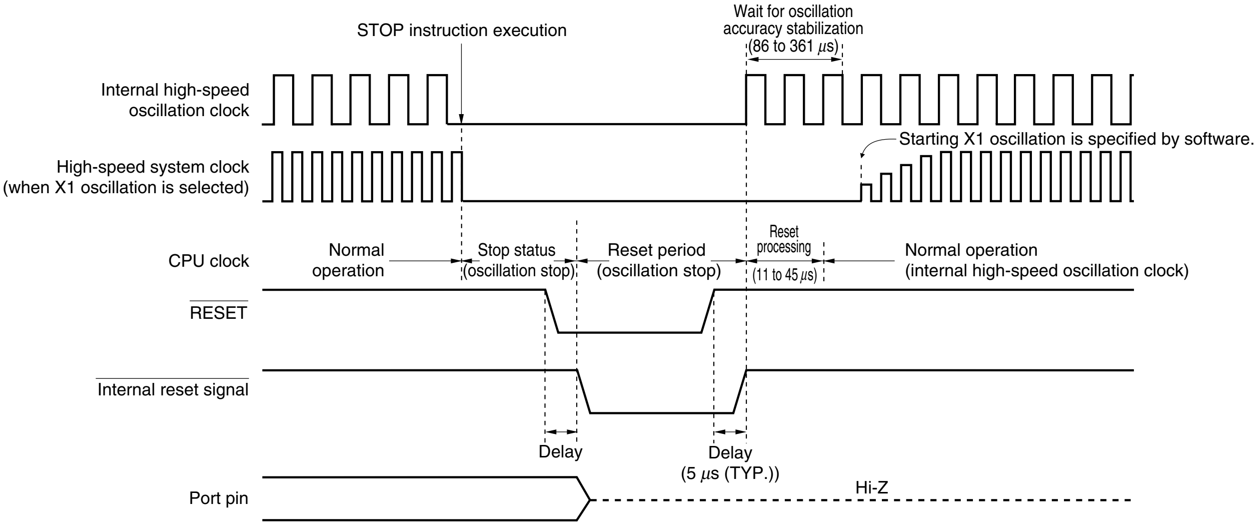


Figure 21-3. Timing of Reset Due to Watchdog Timer Overflow



Caution A watchdog timer internal reset resets the watchdog timer.

Figure 21-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input


STOP instruction execution
Wait for oscillation accuracy stabilization (86 to 361 μs)
Internal high-speed oscillation clock
Starting X1 oscillation is specified by software.
High-speed system clock (when X1 oscillation is selected)
CPU clock
Normal operation
Stop status (oscillation stop)
Reset period (oscillation stop)
Reset processing (11 to 45 μs)
Normal operation (internal high-speed oscillation clock)
RESET
Internal reset signal
Delay
Delay (5 μs (TYP.))
Port pin
Hi-Z

Remark For the reset timing of the power-on-clear circuit and low-voltage detector, see **CHAPTER 23 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 24 LOW-VOLTAGE DETECTOR**.

Table 21-1. Operation Statuses during Reset Period (1/2)

Item			Status during Reset Period
System clock			Clock supply to the CPU is stopped.
	Main system clock	f_{OSC8}	Operation stopped
		f_X	Operation stopped (pin is I/O port mode)
		f_{EXT}	Clock input invalid (pin is I/O port mode)
	Subsystem clock	f_{XT}	Operation stopped (pin is I/O port mode)
		f_{EXTS}	Clock input invalid (pin is I/O port mode)
	f_{OSC}		Operation stopped
CPU			
Flash memory			
RAM			
Regulator			Operable
Port (latch)			Operation stopped
16-bit timer/event counter P	00		
	01		
	02		
	03		
	04		
8-bit timer/event counter	50		
	51		
Watch timer			
Watchdog timer			
Clock output			
Buzzer output			
A/D converter			
Serial interface	UART60		
	UART61		
	CSI10		
	CSI11		
<R>	IIC0		
CAN controller			
Multiplier/divider			
<R> Stepper motor C/D (with ZPD)			
LCD controller/driver			
Sound generator			

Remark		
	f_{OSC8}:	Internal high-speed oscillation clock
	f_X:	X1 oscillation clock
	f_{EXT}:	External main system clock
	f_{XT}:	XT1 oscillation clock
	f_{EXTS}:	External subsystem clock
	f_{OSC}:	Internal low-speed oscillation clock

Table 21-1. Operation Statuses during Reset Period (2/2)

Item	Status during Reset Period
Power-on-clear function	Operable
Low-voltage detection function	Operation stopped
External interrupt	

Table 21-2. Hardware Statuses after Reset Acknowledgment (1/3)

Hardware		Status 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 (P0 to P3, P6 to P9, P12) (output latches)		00H
Port mode registers	PM0 to PM3, PM6 to PM9, PM12	FFH
Pull-up resistor option registers (PU0, PU1, PU3, PU6, PU7, PU12)		00H
Internal expansion RAM size switching register (IXS)		0CH[Note 3]
Internal memory size switching register (IMS)		CFH[Note 3]
Processor clock control register (PCC)		01H
Clock operation mode select register (OSCCTL)		00H
Internal oscillator mode register (RCM)		00H[Note 4]
Main clock mode register (MCM)		00H
Main OSC control register (MOC)		80H
Oscillation stabilization time select register (OSTS)		05H
Oscillation stabilization time counter status register (OSTC)		00H
16-bit timer/event counters P0-P4	Control registers 0, 1 (TP0CTL0-TP4CTL0, TP0CTL1-TP4CTL1)	00H
	I/O control registers 0-2 (TP0CTL0-TP4CTL0, TP0CTL1-TP4CTL1, TP0CTL2-TP4CTL2)	00H
	Option registers 0 (TP0OPT0-TP4OPT0)	00H
	Capture/compare registers 0, 1 (TP0CCR0-TP4CCR0, TP0CCR1-TP4CCR1)	0000H
	Counter read buffer registers (TP0CNT-TP4CNT)	0000H

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/Dx2 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		IMS	IXS
78K0/DE2	78K0/DF2		
μPD78F0836	μPD78F0838, 78F0840, 78F0842	C6H	0AH
μPD78F0844	μPD78F0846, 78F0848	C8H	
μPD78F0837	μPD78F0839, 78F0841, 78F0843	CCH	08H
μPD78F0845	μPD78F0847, 78F0849	CFH	

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

Table 21-2. Hardware Statuses after Reset Acknowledgment (2/3)

Hardware		Status after Reset Acknowledgment[Note 1]
8-bit timer/event counters 50, 51	Timer counters 50, 51 (TM50, TM51)	00H
	Compare registers 50, 51 (CR50, CR51)	00H
	Timer clock selection registers 50, 51 (TCL50, TCL51)	00H
	Mode control registers 50, 51 (TMC50, TMC51)	00H
Watch timer	Operation mode register (WTM)	00H
Clock output/buzzer output controller	Clock output selection register (CKS)	00H
Watchdog timer	Enable register (WDTE)	1AH/9AH[Note 2]
A/D converter	10-bit A/D conversion result register (ADCR)	0000H
	8-bit A/D conversion result register (ADCRH)	00H
	Mode register (ADM)	00H
	Analog input channel specification register (ADS)	00H
	A/D port configuration register (ADPC)	00H
Serial interfaces UART60, UART61	Receive buffer registers 60, 61 (RXB60, RXB61)	FFH
	Transmit buffer registers 60, 61 (TXB60, TXB61)	FFH
	Asynchronous serial interface operation mode registers 60, 61 (ASIM60, ASIM61)	01H
	Asynchronous serial interface reception error status registers 60, 61 (ASIS60, ASIS61)	00H
	Asynchronous serial interface transmission status registers 60, 61 (ASIF60, ASIF61)	00H
	Clock selection registers 60, 61 (CKSR60, CKSR61)	00H
	Baud rate generator control registers 60, 61 (BRGC60, BRGC61)	FFH
	Asynchronous serial interface control registers 60, 61 (ASICL60, ASICL61)	16H
	Input switch control register (ISC)	00H
Serial interfaces CSI10, CSI11	Transmit buffer registers 10, 11 (SOTB10, SOTB11)	00H
	Serial I/O shift registers 10, 11 (SIO10, SIO11)	00H
	Serial operation mode registers 10, 11 (CSIM10, CSIM11)	00H
	Serial clock selection registers 10, 11 (CSIC10, CSIC11)	00H
<R> Serial interface IIC0	IIC shift register 0 (IIC0)	00H
	Slave address register 0 (SVA0)	00H
	IIC control register 0 (IICC0)	00H
	IIC flag register 0 (IICF0)	00H
	IIC status register 0 (IICS0)	00H
	IIC clock selection register 0 (IICCL0)	00H
	IIC function expansion register 0 (IICX0)	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. The reset value of WDTE is determined by the option byte setting.

Table 21-2. Hardware Statuses after Reset Acknowledgment (3/3)

Hardware		Status after Reset Acknowledgment[Note 1]
Multiplier/divider	Remainder data register 0 (SDR0)	0000H
	Multiplication/division data registers A0 (MDA0H, MDA0L)	0000H
	Multiplication/division data register B0 (MDB0)	0000H
	Multiplier/divider control register 0 (DMUC0)	00H
<R> Stepper motor C/D (with ZPD)	Timer mode control registers (MCNTC0)	00H
	Compare registers (MCMP10, MCMP11, MCMP20, MCMP21, MCMP30, MCMP31, MCMP40, MCMP41)	00H
	Combined compare registers (MCMP1HW, MCMP2HW, MCMP3HW, MCMP4HW)	0000H
	Compare control registers (MCMPC1, MCMPC2, MCMPC3, MCMPC4)	00H
	Stepper motor port mode control register (SMPC)	00H
	ZPD detection voltage setting registers (ZPDS0, ZPDS1)	00H
	ZPD flag detection clock setting register (CMPCTL)	00H
	ZPD operational control register (ZPDEN)	00H
LCD controller/driver	LCD mode register (LCDMD)	00H
	LCD display mode register (LCDM)	00H
	LCD clock control register (LCDC0)	00H
	LCD port function registers 0, 3, ALL (LCDPF0, LCDPF3, LCDPFALL)	00H
Sound generator	SG0 control register (SG0CTL)	00H
	SG0 frequency low register (SG0FL)	0000H
	SG0 frequency high register (SG0FH)	0000H
	SG0 volume register (SG0PWM)	0000H
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]
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

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.

Reset Source Register		$\overline{\text{RESET}}$ 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					

21.1 Register for Confirming Reset Source

Many internal reset generation sources exist in the 78K0/Dx2. 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 input by power-on-clear (POC) circuit, and reading RESF clear RESF to 00H.

Figure 21-5. 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 21-3**.

Table 21-3. RESF Status When Reset Request Is Generated

Reset Source Flag	$\overline{\text{RESET}}$ Input	Reset by POC	Reset by WDT	Reset by LVI
WDTRF	Cleared (0)	Cleared (0)	Set (1)	Held
LVIRF			Held	Set (1)

CHAPTER 22 MULTIPLIER/DIVIDER

22.1 Functions of Multiplier/Divider

The multiplier/divider has the following functions.

- 16 bits × 16 bits = 32 bits (multiplication)
- 32 bits ÷ 16 bits = 32 bits, 16-bit remainder (division)

22.2 Configuration of Multiplier/Divider

The multiplier/divider includes the following hardware.

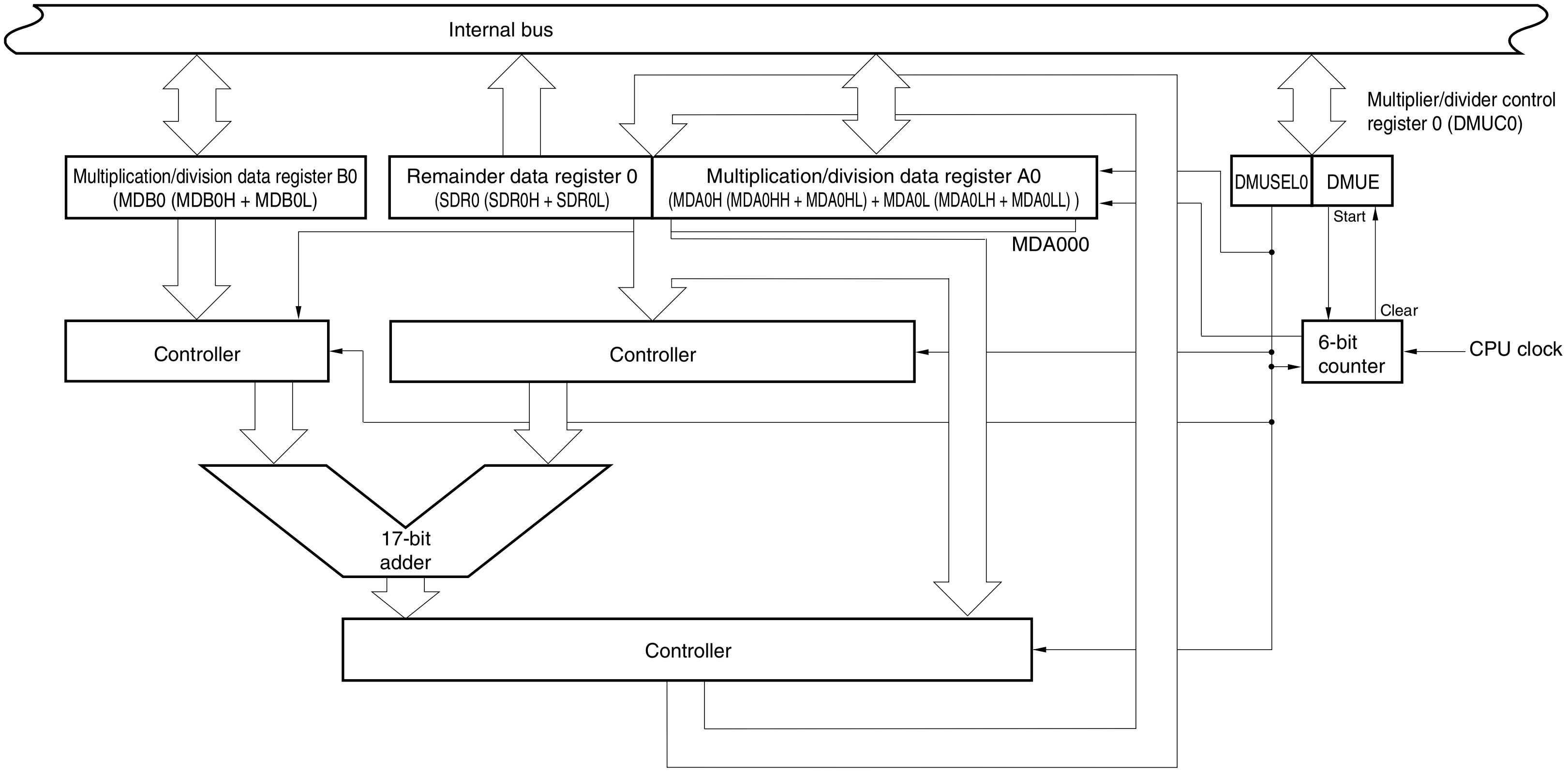
Table 22-1. Configuration of Multiplier/Divider

Item	Configuration
Registers	Remainder data register 0 (SDR0) Multiplication/division data registers A0 (MDA0H, MDA0L) Multiplication/division data register B0 (MDB0)
Control register	Multiplier/divider control register 0 (DMUC0)

Figure 22-1 shows the block diagram of the multiplier/divider.

<R>

Figure 22-1. Block Diagram of Multiplier/Divider


Internal bus
Multiplier/divider control register 0 (DMUC0)
Multiplication/division data register B0 (MDB0 (MDB0H + MDB0L)
Remainder data register 0 (SDR0 (SDR0H + SDR0L)
Multiplication/division data register A0 (MDA0H (MDA0HH + MDA0HL) + MDA0L (MDA0LH + MDA0LL))
MDA000
DMUSEL0
DMUE
Start
Clear
6-bit counter
CPU clock
Controller
Controller
17-bit adder
Controller

(1) Remainder data register 0 (SDR0)

SDR0 is a 16-bit register that stores a remainder. This register stores 0 in the multiplication mode and the remainder of an operation result in the division mode.

SDR0 can be read by an 8-bit or 16-bit memory manipulation instruction.

Reset signal generation clears SDR0 to 0000H.

Figure 22-2. Format of Remainder Data Register 0 (SDR0)

Address: FF60H, FF61H After reset: 0000H R

Symbol	FF61H (SDR0H)								FF60H (SDR0L)							
SDR0	SDR 015	SDR 014	SDR 013	SDR 012	SDR 011	SDR 010	SDR 009	SDR 008	SDR 007	SDR 006	SDR 005	SDR 004	SDR 003	SDR 002	SDR 001	SDR 000

Cautions 1. The value read from SDR0 during operation processing (while bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is 1) is not guaranteed.

2. SDR0 is reset when the operation is started (when DMUE is set to 1).

(2) Multiplication/division data register A0 (MDA0H, MDA0L)

MDA0 is a 32-bit register that sets a 16-bit multiplier A in the multiplication mode and a 32-bit dividend in the division mode, and stores the 32-bit result of the operation (higher 16 bits: MDA0H, lower 16 bits: MDA0L).

Figure 22-3. Format of Multiplication/Division Data Register A0 (MDA0H, MDA0L)

Address: FF62H, FF63H, FF64H, FF65H After reset: 0000H, 0000H R/W

Symbol	FF65H (MDA0HH)								FF64H (MDA0HL)							
MDA0H	MDA 031	MDA 030	MDA 029	MDA 028	MDA 027	MDA 026	MDA 025	MDA 024	MDA 023	MDA 022	MDA 021	MDA 020	MDA 019	MDA 018	MDA 017	MDA 016

Symbol	FF63H (MDA0LH)								FF62H (MDA0LL)							
MDA0L	MDA 015	MDA 014	MDA 013	MDA 012	MDA 011	MDA 010	MDA 009	MDA 008	MDA 007	MDA 006	MDA 005	MDA 004	MDA 003	MDA 002	MDA 001	MDA 000

Cautions 1. MDA0H is cleared to 0 when an operation is started in the multiplication mode (when multiplier/divider control register 0 (DMUC0) is set to 81H).

2. Do not change the value of MDA0 during operation processing (while bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is 1). Even in this case, the operation is executed, but the result is undefined.

3. The value read from MDA0 during operation processing (while DMUE is 1) is not guaranteed.

The functions of MDA0 when an operation is executed are shown in the table below.

Table 22-2. Functions of MDA0 During Operation Execution

DMUSEL0	Operation Mode	Setting	Operation Result
0	Division mode	Dividend	Division result (quotient)
1	Multiplication mode	Higher 16 bits: 0, Lower 16 bits: Multiplier A	Multiplication result (product)

The register configuration differs between when multiplication is executed and when division is executed, as follows.

- Register configuration during multiplication

  <Multiplier A> <Multiplier B> <Product>

  MDA0 (bits 15 to 0) × MDB0 (bits 15 to 0) = MDA0 (bits 31 to 0)

- Register configuration during division

  <Dividend> <Divisor> <Quotient> <Remainder>

  MDA0 (bits 31 to 0) ÷ MDB0 (bits 15 to 0) = MDA0 (bits 31 to 0) … SDR0 (bits 15 to 0)

MDA0 fetches the calculation result as soon as the clock is input, when bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is set to 1.
MDA0H and MDA0L can be set by an 8-bit or 16-bit memory manipulation instruction.
Reset signal generation clears MDA0H and MDA0L to 0000H.

(3) Multiplication/division data register B0 (MDB0)

MDB0 is a register that stores a 16-bit multiplier B in the multiplication mode and a 16-bit divisor in the division mode.
MDB0 can be set by an 8-bit or 16-bit memory manipulation instruction.
Reset signal generation clears MDB0 to 0000H.

Figure 22-4. Format of Multiplication/Division Data Register B0 (MDB0)

Address: FF66H, FF67H After reset: 0000H R/W

Symbol	FF67H (MDB0H)								FF66H (MDB0L)							
MDB0	MDB 015	MDB 014	MDB 013	MDB 012	MDB 011	MDB 010	MDB 009	MDB 008	MDB 007	MDB 006	MDB 005	MDB 004	MDB 003	MDB 002	MDB 001	MDB 000

Cautions 1. Do not change the value of MDB0 during operation processing (while bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is 1). Even in this case, the operation is executed, but the result is undefined.

2. Do not clear MDB0 to 0000H in the division mode. If set, undefined operation results are stored in MDA0 and SDR0.

22.3 Register Controlling Multiplier/Divider

The multiplier/divider is controlled by multiplier/divider control register 0 (DMUC0).

(1) Multiplier/divider control register 0 (DMUC0)

DMUC0 is an 8-bit register that controls the operation of the multiplier/divider.
DMUC0 can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears DMUC0 to 00H.

Figure 22-5. Format of Multiplier/Divider Control Register 0 (DMUC0)

Address: FF68H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
DMUC0	DMUE	0	0	0	0	0	0	DMUSEL0

DMUE[Note]	Operation start/stop
0	Stops operation
1	Starts operation

DMUSEL0	Operation mode (multiplication/division) selection
0	Division mode
1	Multiplication mode

Note When DMUE is set to 1, the operation is started. DMUE is automatically cleared to 0 after the operation is complete.

Cautions 1. If DMUE is cleared to 0 during operation processing (when DMUE is 1), the operation result is not guaranteed. If the operation is completed while the clearing instruction is being executed, the operation result is guaranteed, provided that the interrupt flag is set.

2. Do not change the value of DMUSEL0 during operation processing (while DMUE is 1). If it is changed, undefined operation results are stored in multiplication/division data register A0 (MDA0) and remainder data register 0 (SDR0).

3. If DMUE is cleared to 0 during operation processing (while DMUE is 1), the operation processing is stopped. To execute the operation again, set multiplication/division data register A0 (MDA0), multiplication/division data register B0 (MDB0), and multiplier/divider control register 0 (DMUC0), and start the operation (by setting DMUE to 1).

22.4 Operations of Multiplier/Divider

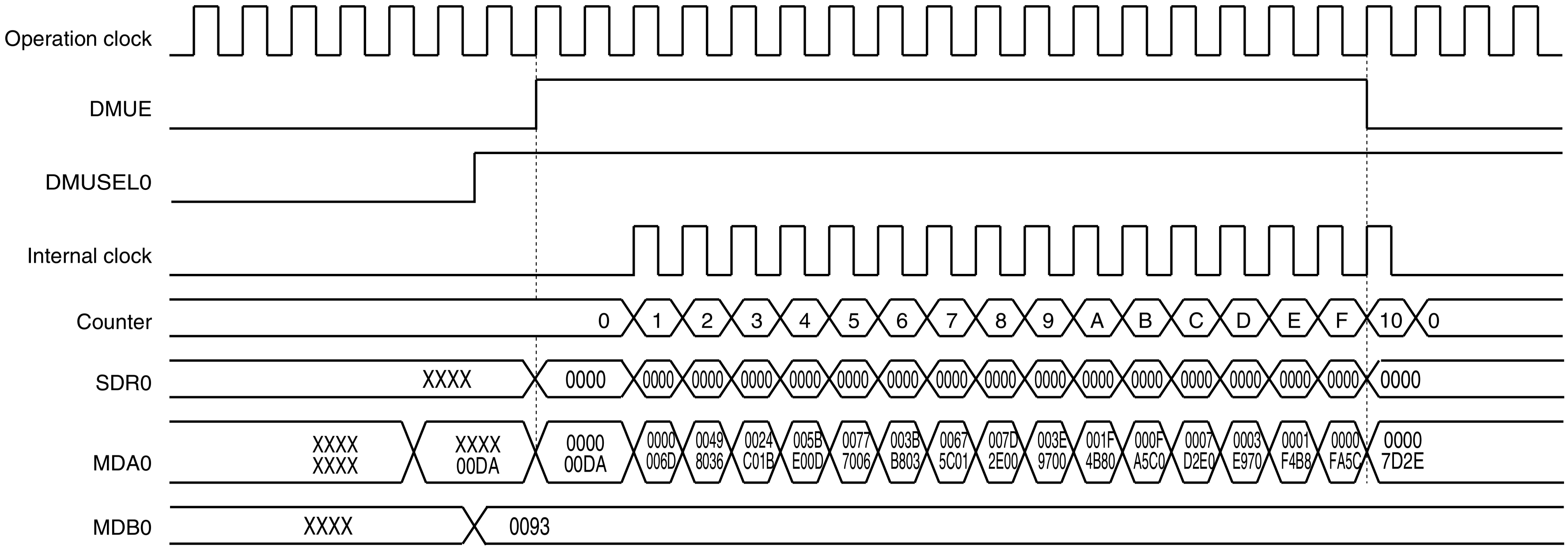
22.4.1 Multiplication operation

- Initial setting
  1. Set operation data to multiplication/division data register A0L (MDA0L) and multiplication/division data register B0 (MDB0).
  2. Set bits 0 (DMUSEL0) and 7 (DMUE) of multiplier/divider control register 0 (DMUC0) to 1. Operation will start.
- During operation
  3. The operation will be completed when 16 peripheral hardware clocks (f_{PRS}) have been issued after the start of the operation (intermediate data is stored in the MDA0L and MDA0H registers during operation, and therefore the read values of these registers are not guaranteed).
- End of operation
  4. The operation result data is stored in the MDA0L and MDA0H registers.
  5. DMUE is cleared to 0 (end of operation).

<R>
- Next operation
  6. To execute multiplication next, start from the initial setting in **22.4.1 Multiplication operation**.
  7. To execute division next, start from the initial setting in **22.4.2 Division operation**.

<R>

Figure 22-6. Timing Chart of Multiplication Operation (00DAH × 0093H)



22.4.2 Division operation

- Initial setting

1. Set operation data to multiplication/division data register A0 (MDA0L and MDA0H) and multiplication/division data register B0 (MDB0).
2. Set bits 0 (DMUSEL0) and 7 (DMUE) of multiplier/divider control register 0 (DMUC0) to 0 and 1, respectively. Operation will start.

- During operation

3. The operation will be completed when 32 peripheral hardware clocks (f_{PRS}) have been issued after the start of the operation (intermediate data is stored in the MDA0L and MDA0H registers and remainder data register 0 (SDR0) during operation, and therefore the read values of these registers are not guaranteed).

- End of operation

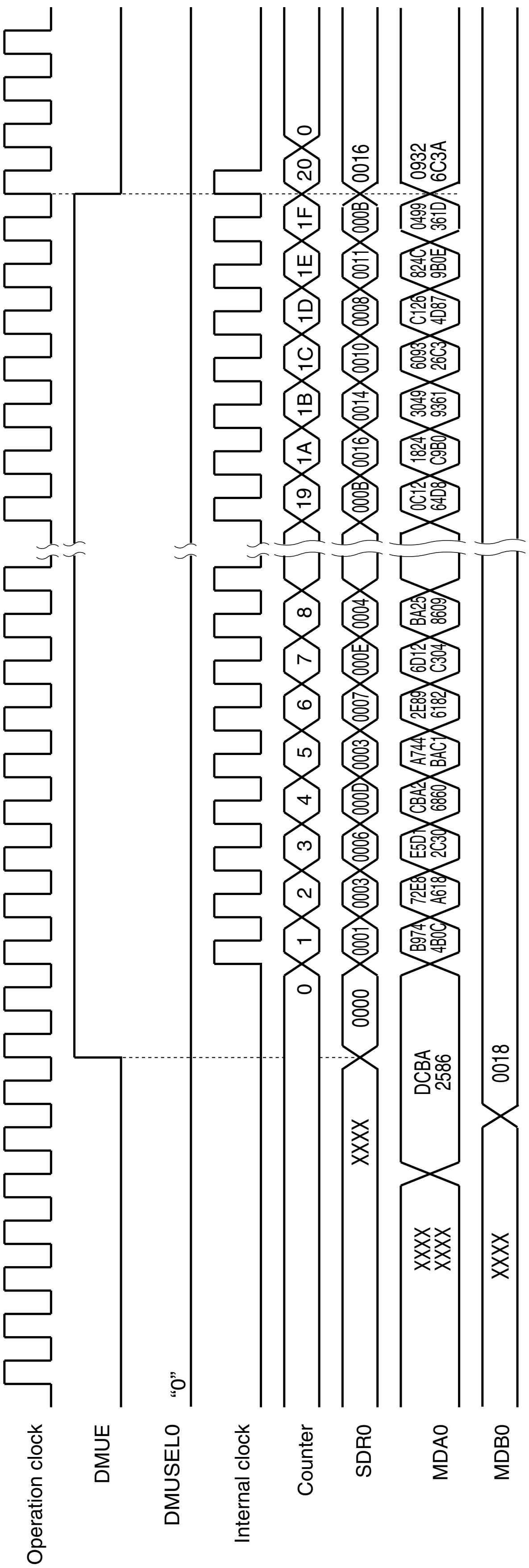
4. The result data is stored in the MDA0L, MDA0H, and SDR0 registers.
5. DMUE is cleared to 0 (end of operation).

<R> - Next operation

6. To execute multiplication next, start from the initial setting in **22.4.1 Multiplication operation**.
7. To execute division next, start from the initial setting in **22.4.2 Division operation**.

<R>

Figure 22-7. Timing Chart of Division Operation (DCBA2586H ÷ 0018H)



CHAPTER 23 POWER-ON-CLEAR CIRCUIT

23.1 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: LVISTART = 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: LVISTART = 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 $V_{DD} < V_{POC}$.

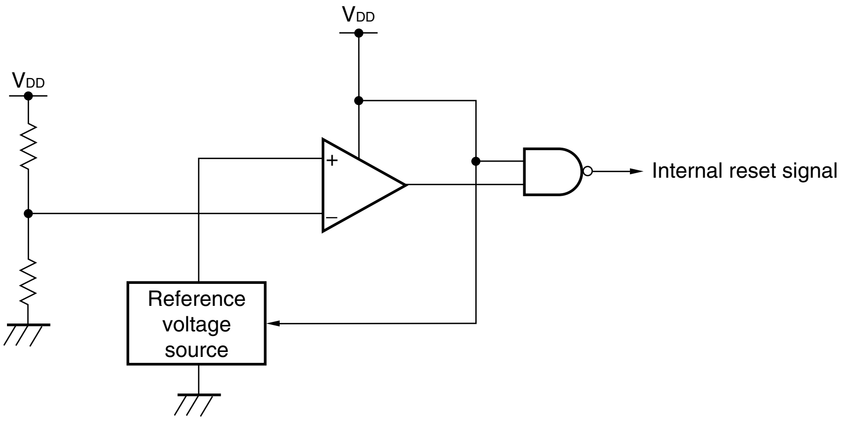
Caution If an internal reset signal is generated in the POC circuit, the reset control flag register (RESF) is cleared to 00H.

Remark The 78K0/Dx2 incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset cause 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 21 RESET FUNCTION**.

23.2 Configuration of Power-on-Clear Circuit

The block diagram of the power-on-clear circuit is shown in **Figure 23-1**.

Figure 23-1. Block Diagram of Power-on-Clear Circuit



23.3 Operation of Power-on-Clear Circuit

(1) In 1.59 V POC mode (option byte: LVISTART = 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 $V_{DD} \geq V_{POC}$.

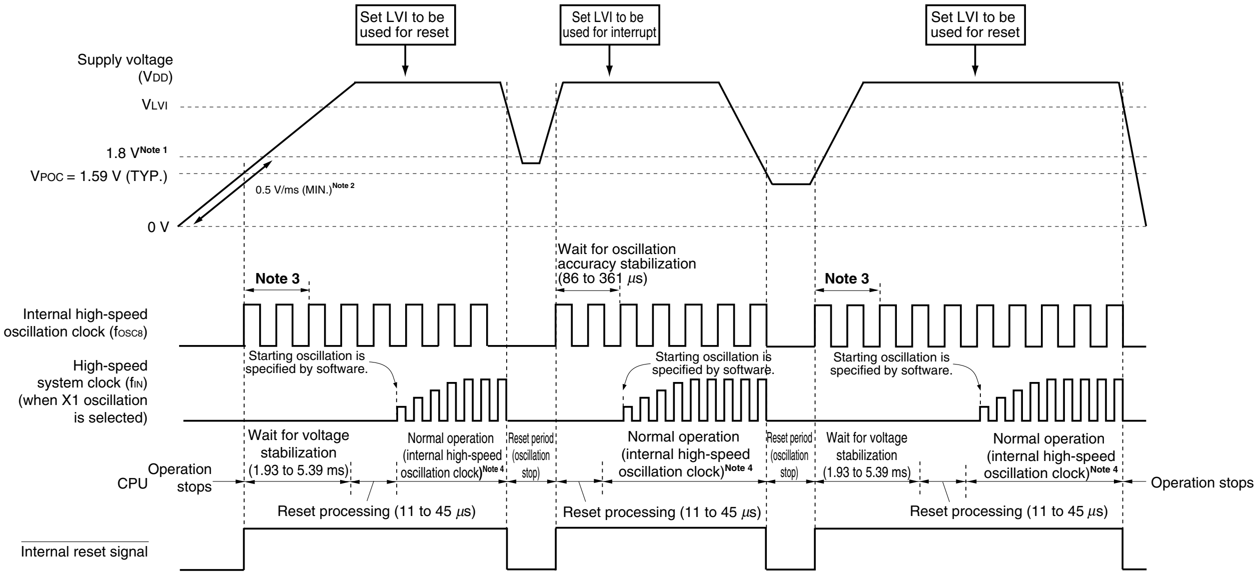
(2) In 2.7 V/1.59 V POC mode (option byte: LVISTART = 1)

- 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.

Figure 23-2. 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: LVISTART = 0)


Set LVI to be used for reset
Set LVI to be used for interrupt
Set LVI to be used for reset
Supply voltage (VDD)
VLVI
1.8 VNote 1
VPOC = 1.59 V (TYP.)
0.5 V/ms (MIN.)Note 2
0 V
Note 3
Wait for oscillation accuracy stabilization (86 to 361 µs)
Note 3
Internal high-speed oscillation clock (fOSC8)
High-speed system clock (fIN) (when X1 oscillation is selected)
Starting oscillation is specified by software.
Starting oscillation is specified by software.
Starting oscillation is specified by software.
CPU
Operation stops
Wait for voltage stabilization (1.93 to 5.39 ms)
Normal operation (internal high-speed oscillation clock)Note 4
Reset period (oscillation stop)
Normal operation (internal high-speed oscillation clock)Note 4
Reset period (oscillation stop)
Wait for voltage stabilization (1.93 to 5.39 ms)
Normal operation (internal high-speed oscillation clock)Note 4
Operation stops
Reset processing (11 to 45 µs)
Reset processing (11 to 45 µs)
Reset processing (11 to 45 µs)
Internal reset signal

Notes 1. The guaranteed operation range for the (A) grade products is 1.8 V ≤ V_{DD} ≤ 5.5 V, and 2.7 V ≤ V_{DD} ≤ 5.5 V for the (A2) grade products. To set the voltage range below the guaranteed operation range to the reset state when the supply voltage falls, use the reset function of the low-voltage detector, or input a low level to the $\overline{\text{RESET}}$ pin.

2. With the (A) grade products, 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 (LVISTART = 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.

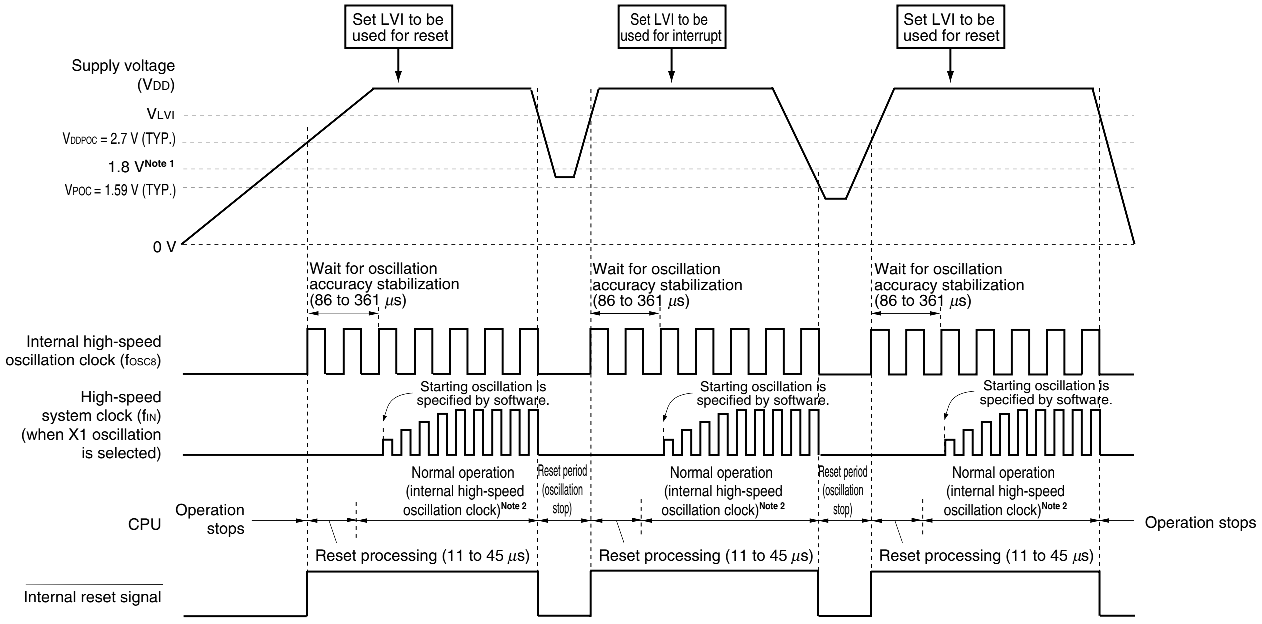
Caution Set the low-voltage detector by software after the reset status is released (see CHAPTER 24 LOW-VOLTAGE DETECTOR).

Remark V_{LVI} : LVI detection voltage

V_{POC} : POC detection voltage

Figure 23-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (2/2)

(2) In 2.7 V / 1.59V POC mode (option byte: LVISTART = 1)


Set LVI to be used for reset
Set LVI to be used for interrupt
Set LVI to be used for reset
Supply voltage (VDD)
VLVI
VDDPOC = 2.7 V (TYP.)
1.8 V Note 1
VPOC = 1.59 V (TYP.)
0 V
Wait for oscillation accuracy stabilization (86 to 361 μs)
Wait for oscillation accuracy stabilization (86 to 361 μs)
Wait for oscillation accuracy stabilization (86 to 361 μs)
Internal high-speed oscillation clock (fOSC8)
High-speed system clock (fIN) (when X1 oscillation is selected)
Starting oscillation is specified by software.
Starting oscillation is specified by software.
Starting oscillation is specified by software.
CPU
Operation stops
Normal operation (internal high-speed oscillation clock) Note 2
Reset period (oscillation stop)
Normal operation (internal high-speed oscillation clock) Note 2
Reset period (oscillation stop)
Normal operation (internal high-speed oscillation clock) Note 2
Operation stops
Reset processing (11 to 45 μs)
Reset processing (11 to 45 μs)
Reset processing (11 to 45 μs)
Internal reset signal

Notes 1. The guaranteed operation range for the (A) grade products is 1.8 V ≤ V_{DD} ≤ 5.5 V, and 2.7 V ≤ V_{DD} ≤ 5.5 V for the (A2) grade products. To set the voltage range below the guaranteed operation range to the reset state when the supply voltage falls, use the reset function of the low-voltage detector, or input a 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 24 LOW-VOLTAGE DETECTOR).

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 supply voltage rises from 1.59 V (TYP.) to 2.7 V (TYP.) within 1.93 ms, the power supply oscillation stabilization time of 0 to 5.39 ms is automatically generated before reset processing.

Remark V_{LVI} : LVI detection voltage
V_{POC} : POC detection voltage

23.4 Cautions for Power-on-Clear Circuit

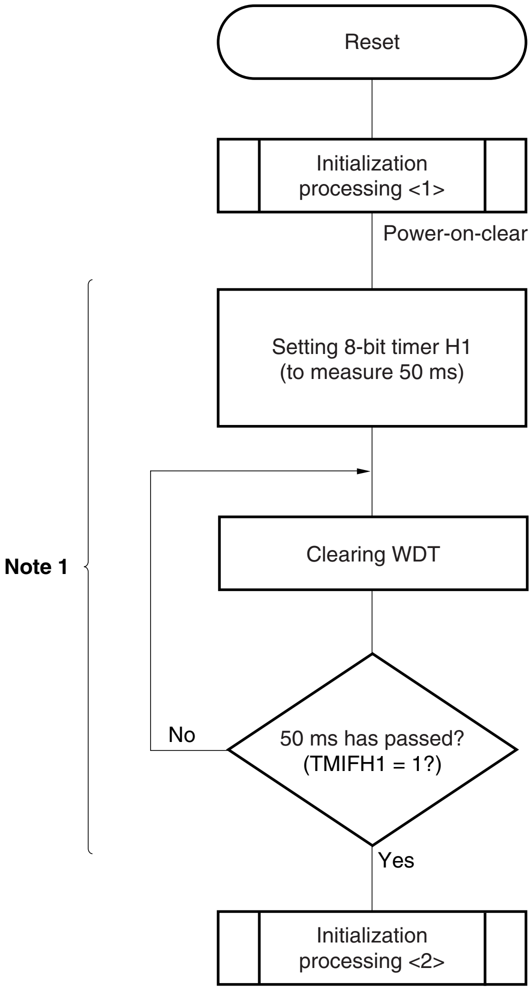
In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the POC detection voltage (V_{POC}), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller 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 23-3. Example of Software Processing After Reset Release (1/2)

- If supply voltage fluctuation is 50 ms or less in vicinity of POC detection voltage


Reset
Initialization processing <1>
Power-on-clear
Setting 8-bit timer H1 (to measure 50 ms)
Clearing WDT
No
50 ms has passed? (TMIFH1 = 1?)
Yes
Initialization processing <2>
Note 1

; Check the reset source[Note 2]
Initialize the port.

; f_{PRS} = Internal high-speed oscillation clock (8.4 MHz (MAX.)) (default)
Source: f_{PRS} (8.4 MHz (MAX.))/2^{12},
where comparison value = 102: ≅ 50 ms
Timer starts (TMHE1 = 1).

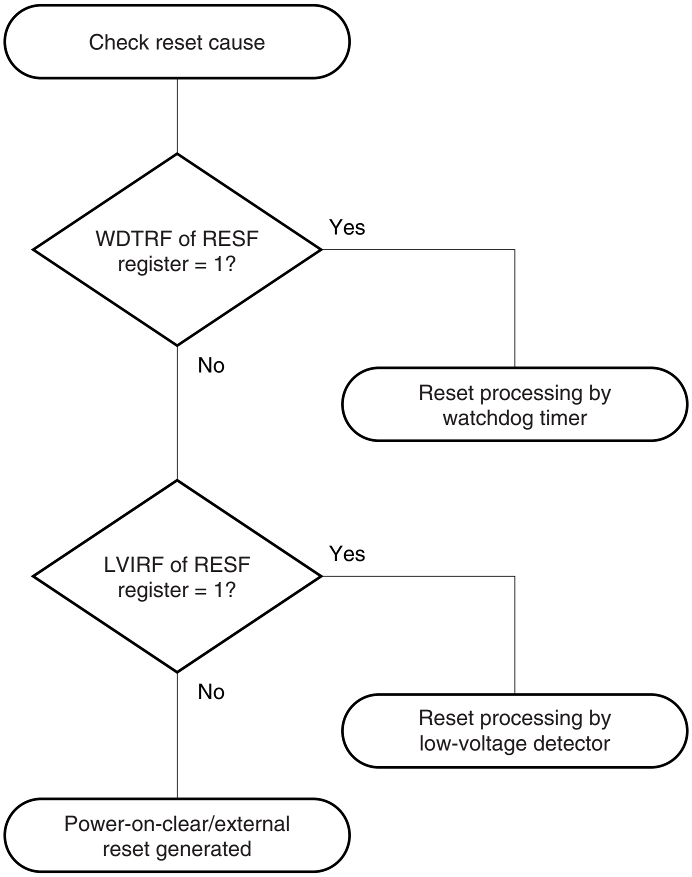
; Setting of division ratio of system clock,
such as setting of timer or A/D converter

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 23-3. Example of Software Processing After Release of Reset (2/2)

- Checking reset cause


Check reset cause
WDTRF of RESF register = 1?
Yes
No
Reset processing by watchdog timer
LVIRF of RESF register = 1?
Yes
No
Reset processing by low-voltage detector
Power-on-clear/external reset generated

CHAPTER 24 LOW-VOLTAGE DETECTOR

24.1 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

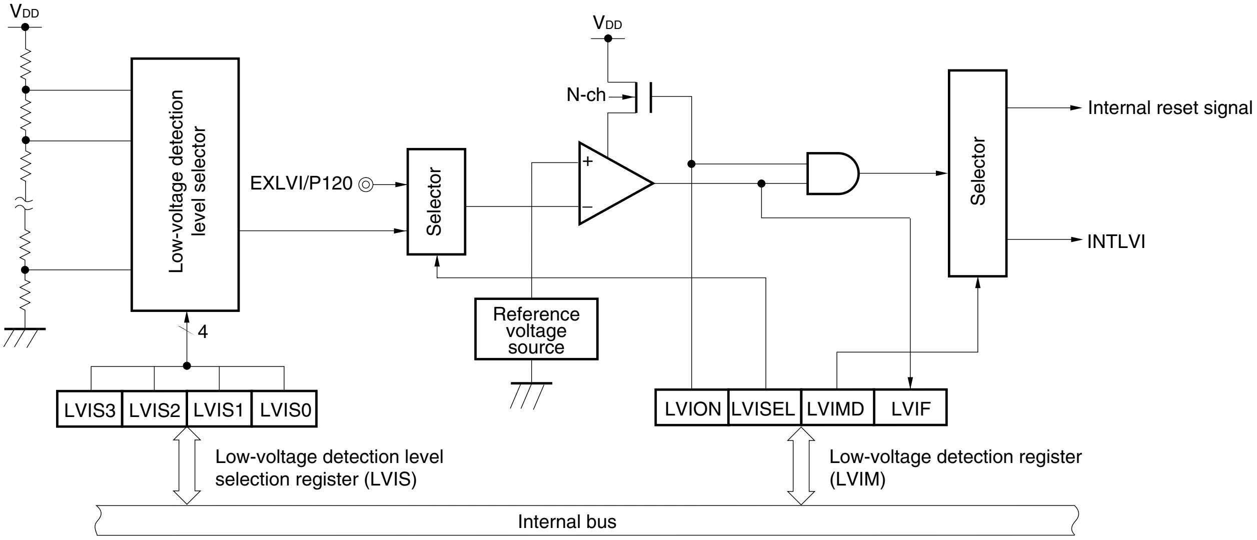
While the low-voltage detector is operating, whether the supply voltage or the input voltage from an external input pin is more than or less than the detection level can be checked by reading the low-voltage detection flag (LVIF: bit 0 of LVIM).

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 21 RESET FUNCTION**.

24.2 Configuration of Low-Voltage Detector

The block diagram of the low-voltage detector is shown in **Figure 24-1**.

Figure 24-1. Block Diagram of Low-Voltage Detector



24.3 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 24-2. Format of Low-Voltage Detection Register (LVIM)

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

Symbol	<7>	6	5	4	3	<2>	<1>	<0>
LVIM	LVION	0	0	0	0	LVISEL	LVIMD	LVIF

LVION[Notes 2, 3]	Enables low-voltage detection operation
0	Disables operation
1	Enables operation

LVISEL[Note 2]	Voltage detection selection
0	Detects level of supply voltage (V_{DD})
1	Detects level of input voltage from external input pin (EXLVI)

LVIMD[Note 2]	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 ≥ 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 ≥ V_{EXLVI}.

LVIF[Note 4]	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. Bit 0 is read-only.

2. 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.

3. 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 (MIN.)) and minimum pulse width 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).

4. The value of LVIF is output as the interrupt request signal INTLVI when LVION = 1 and LVIMD = 0.

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}.

3. After an LVI reset has been generated, do not write values to LVIS and LVIM when LVION = 1.

4. 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 24-3. Format of Low-Voltage Detection Level Selection Register (LVIS)

Address: FFBFH After reset: 00H [Note 1] 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) [Note 2]

Notes 1. The value of LVIS is not reset but retained as is, upon a reset by LVI. It is cleared to 00H upon other resets.

2. Do not set V_{LVI10} to V_{LVI15} for (A2) grade products.

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.

4. After an LVI reset has been generated, do not write values to LVIS and LVIM when LVION = 1.

(3) Port mode register 12 (PM12)

When using the P120/EXLVI 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 24-4. 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	PM124	PM123	PM122	PM121	PM120

PM12n	P12n pin I/O mode selection (n = 0 to 4)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

24.4 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

24.4.1 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 (MIN.)) and minimum pulse width.

<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 24-5 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

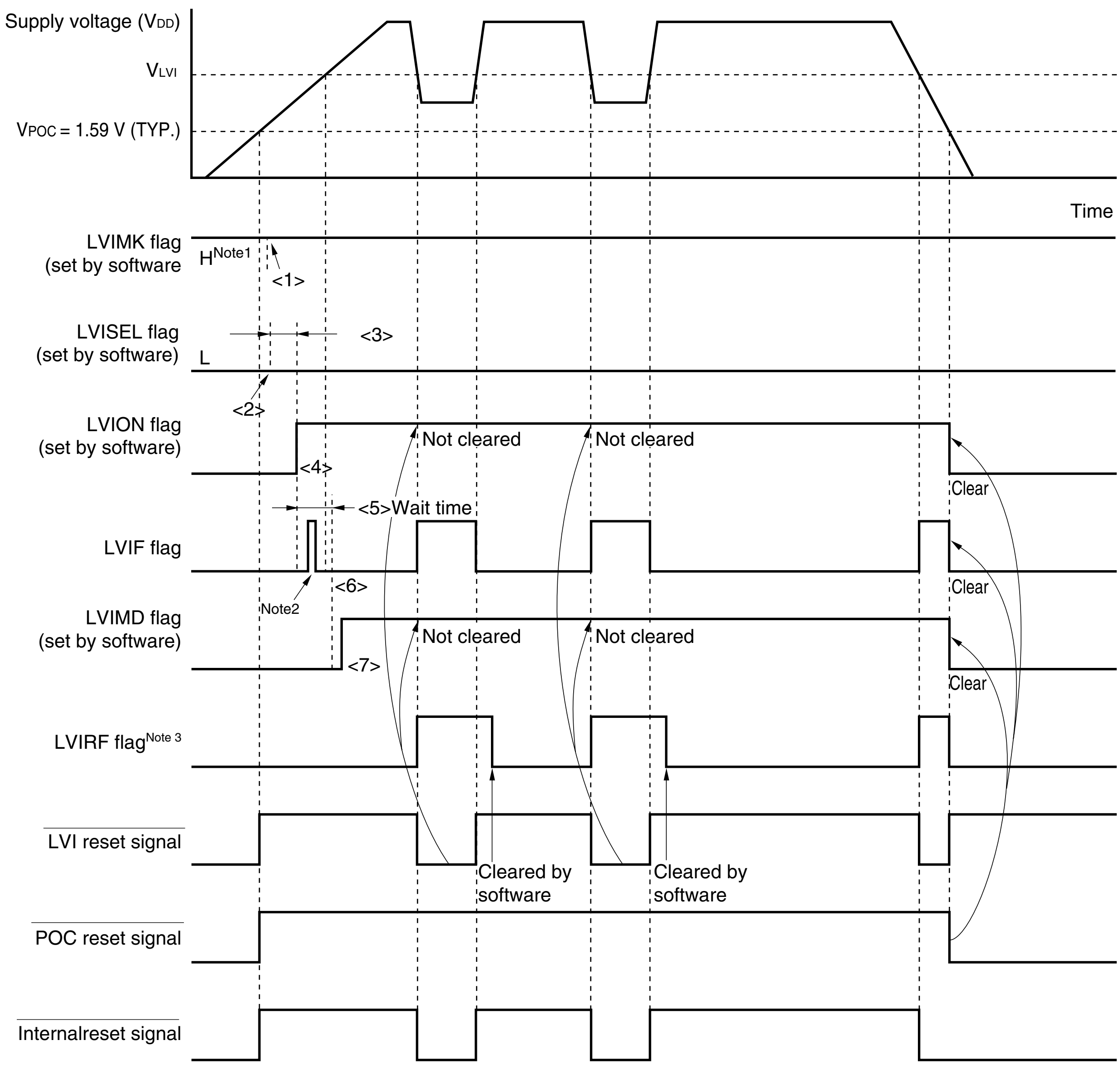
Either of the following procedures must be executed.

- 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 24-5. 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 setup (option byte: LVISTART = 0)



Notes 1. The LVIMK flag is set to “1” by reset signal generation.

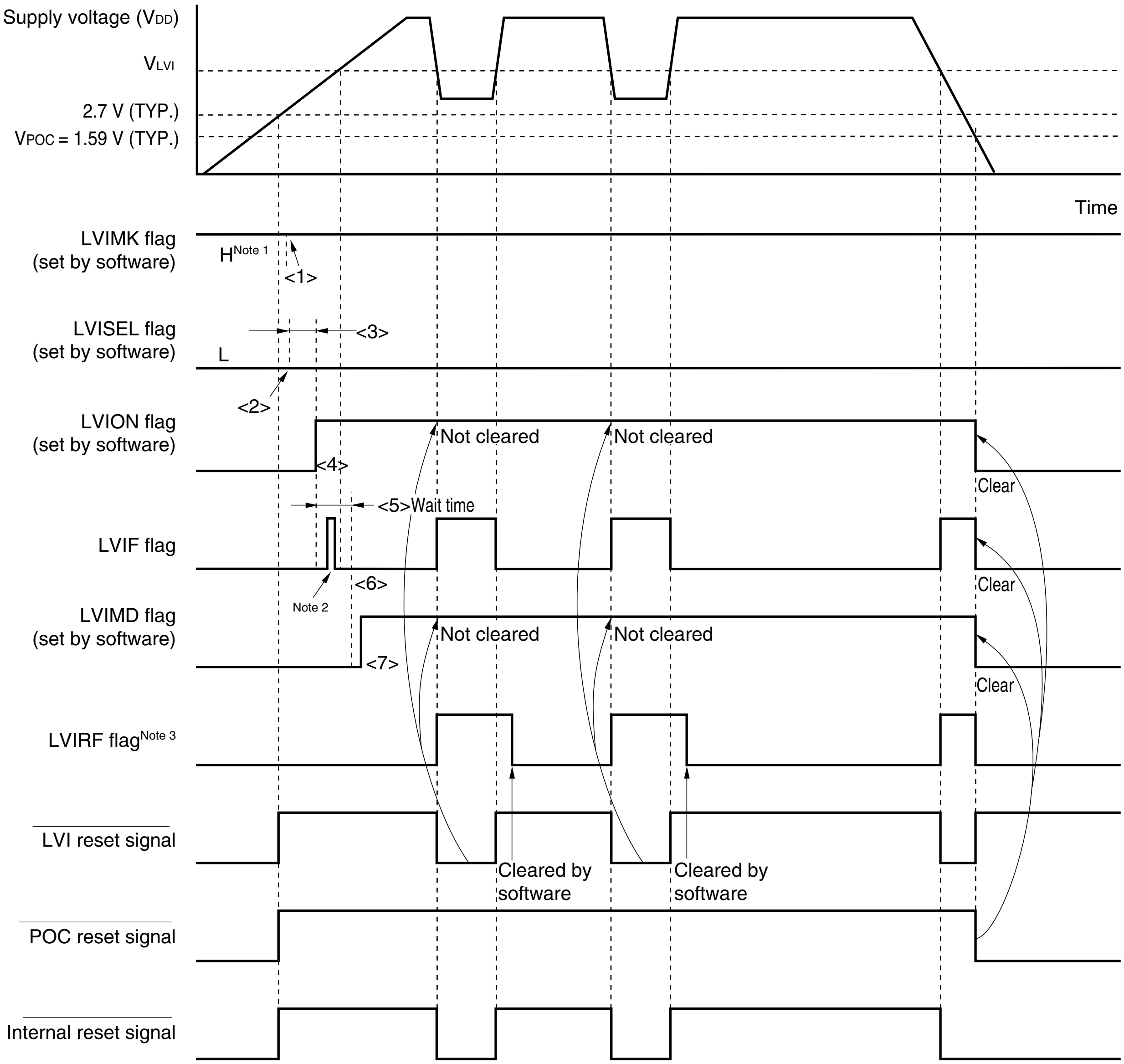
2. The LVIF flag may be set (1).

3. LVIRF is bit 0 of the reset control flag register (RESF). For details of RESF, see **CHAPTER 21 RESET FUNCTION**.

Remark <1> to <7> in **Figure 24-5** above correspond to <1> to <7> in the description of “When starting operation” in **24.4.1 (1) When detecting level of supply voltage (V_{DD})**.

Figure 24-5. Timing of Low-Voltage Detector Internal Reset Signal Generation (Detects Level of Supply Voltage (V_{DD})) (2/2)

(2) In 2.7/1.59 V POC mode setup (option byte: LVISTART = 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 21 RESET FUNCTION**.

Remark <1> to <7> in **Figure 24-5** above correspond to <1> to <7> in the description of “When starting operation” in **24.4.1 (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 (MIN.)) and minimum pulse width.

<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 24-6 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}.

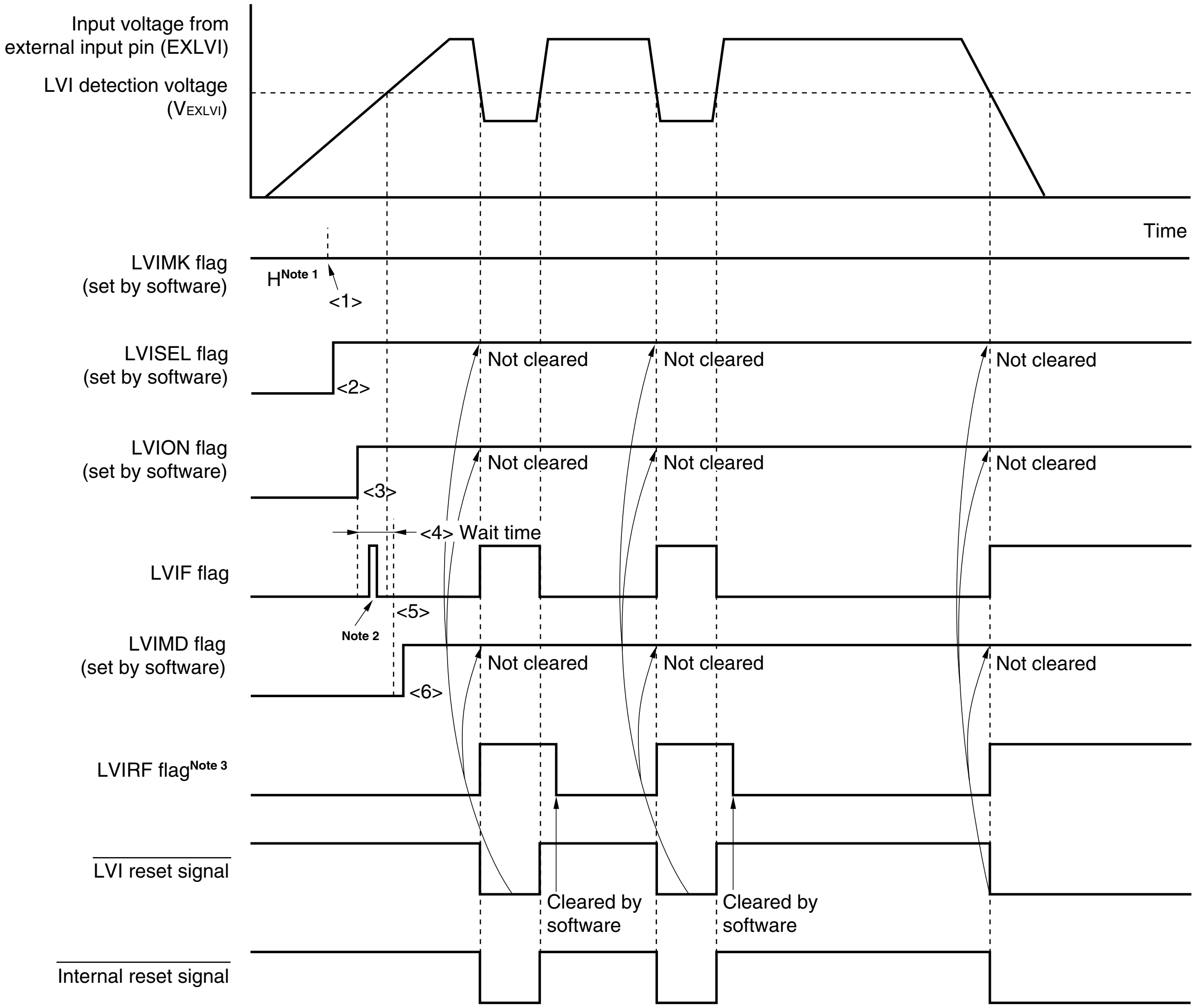
- When stopping operation

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 24-6. 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 21 RESET FUNCTION**.

Remark <1> to <6> in **Figure 24-6** above correspond to <1> to <6> in the description of “When starting operation” in **24.4.1 (2) When detecting level of input voltage from external input pin (EXLVI)**.

24.4.2 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 (MIN.)) and minimum pulse width.

<6> 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}, 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 24-7 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

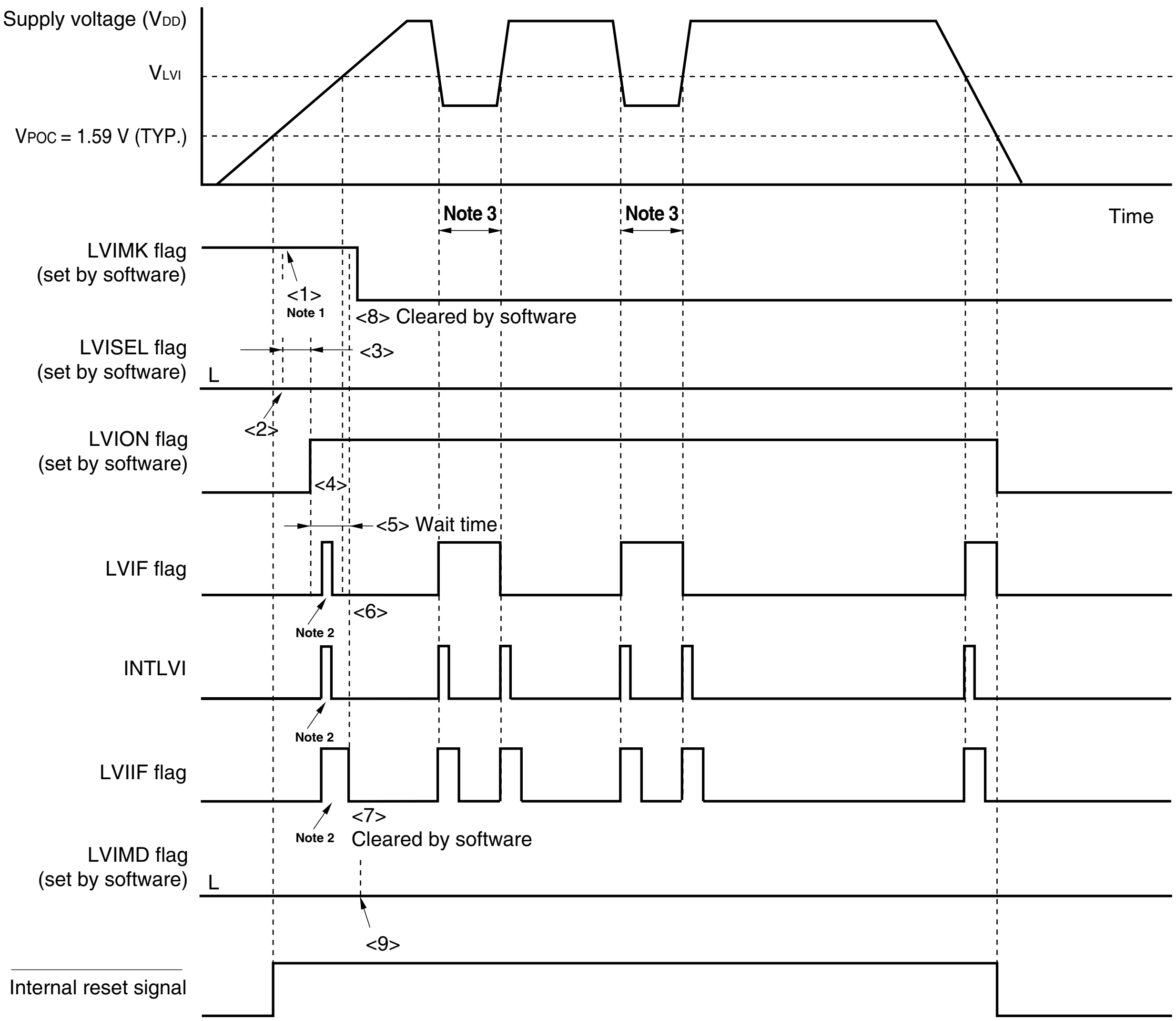
Either of the following procedures must be executed.

- When using 8-bit memory manipulation instruction:
  Write 00H to LVIM.

- When using 1-bit memory manipulation instruction:
  Clear LVION to 0.

Figure 24-7. Timing of Low-Voltage Detector Interrupt Signal Generation (Detects Level of Supply Voltage (VDD)) (1/2)

(1) In 1.59 V POC mode setup (option byte: LVISTART = 0)



Notes 1. The LVIMK flag is set to “1” by reset signal generation.

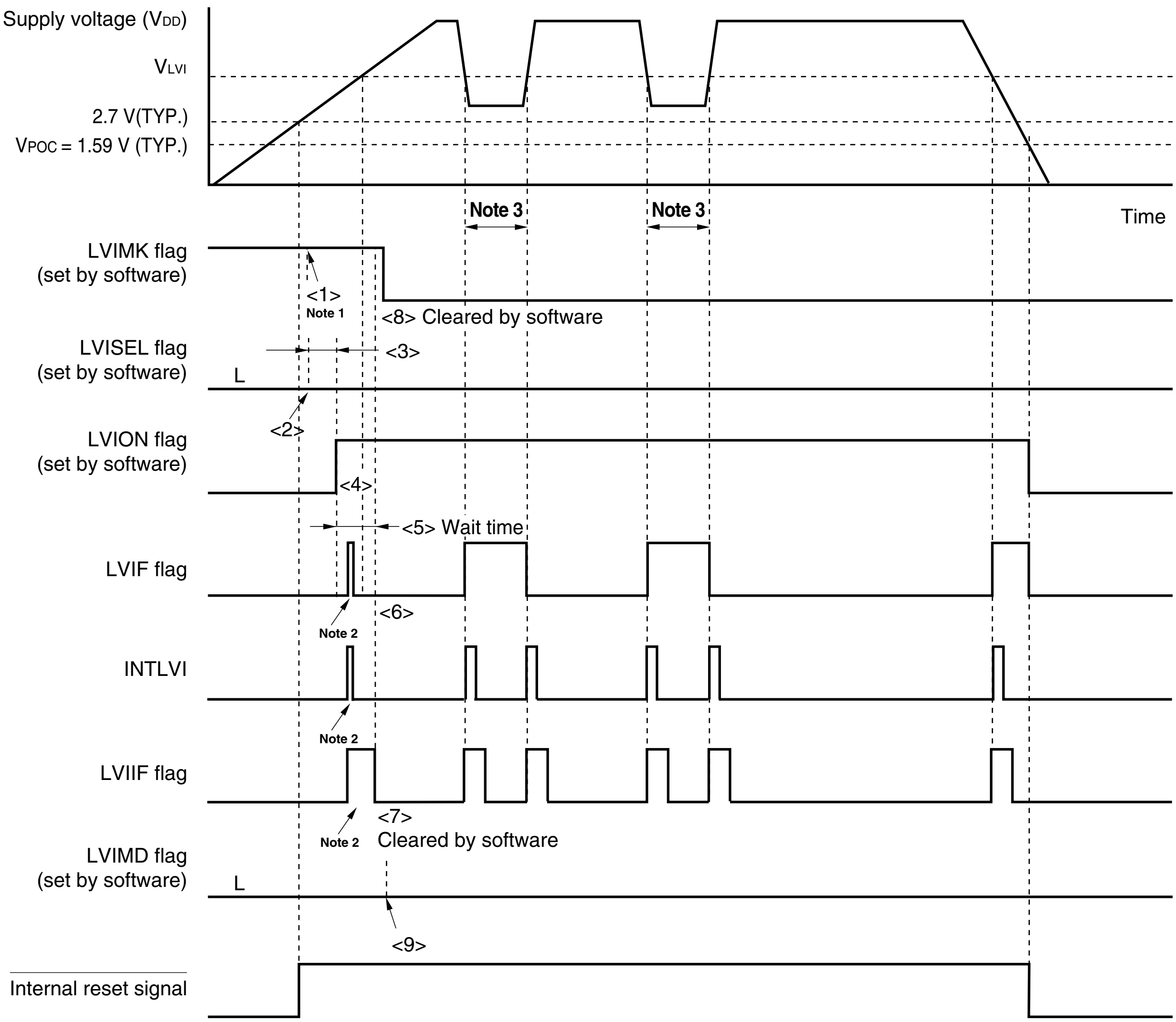
2. The interrupt request signal (INTLVI) is generated and the LVIF and LVIIF flags may be set (1).

3. If LVION is cleared (0) in state below the LVI detection voltage, an INTLVI signal is generated and LVIIF becomes 1.

Remark <1> to <9> in **Figure 24-7** above correspond to <1> to <9> in the description of “When starting operation” in **24.4.2 (1) When detecting level of supply voltage (VDD)**.

Figure 24-7. Timing of Low-Voltage Detector Interrupt Signal Generation (Detects Level of Supply Voltage (V_{DD})) (2/2)

(2) In 2.7/1.59 V POC mode setup (option byte: LVISTART = 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 state below the LVI detection voltage, an INTLVI signal is generated and LVIIF becomes 1.

Remark <1> to <9> in **Figure 24-7** above correspond to <1> to <9> in the description of “When starting operation” in **24.4.2 (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 (MIN.)) and minimum pulse width.

<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 24-8 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}.

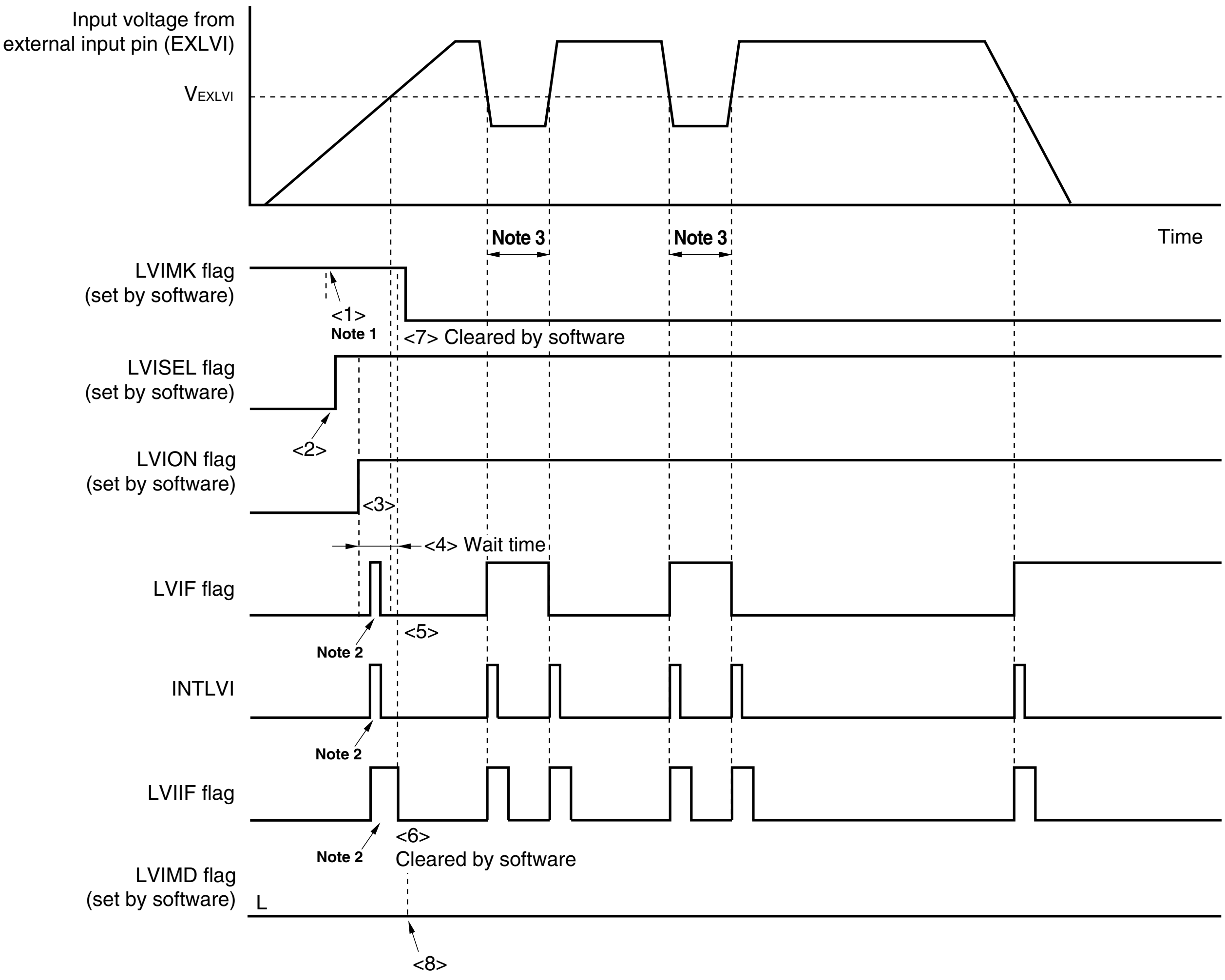
- When stopping operation

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 24-8. Timing of Low-Voltage Detector Interrupt Signal Generation (Detects Level of Input Voltage from External Input Pin (EXLVI))


Input voltage from external input pin (EXLVI)
VEXLVI
Time
Note 3
Note 3
LVIMK flag (set by software)
<1>
Note 1
<7> Cleared by software
LVISEL flag (set by software)
<2>
LVION flag (set by software)
<3>
<4> Wait time
LVIF flag
Note 2
<5>
INTLVI
Note 2
LVIIF flag
Note 2
<6>
Cleared by software
LVIMD flag (set by software)
L
<8>

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 state below the LVI detection voltage, an INTLVI signal is generated and LVIIF becomes 1.

Remark <1> to <8> in **Figure 24-8** above correspond to <1> to <8> in the description of “When starting operation” in **24.4.2 (2) When detecting level of input voltage from external input pin (EXLVI)**.

24.5 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 24-9**).

(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.

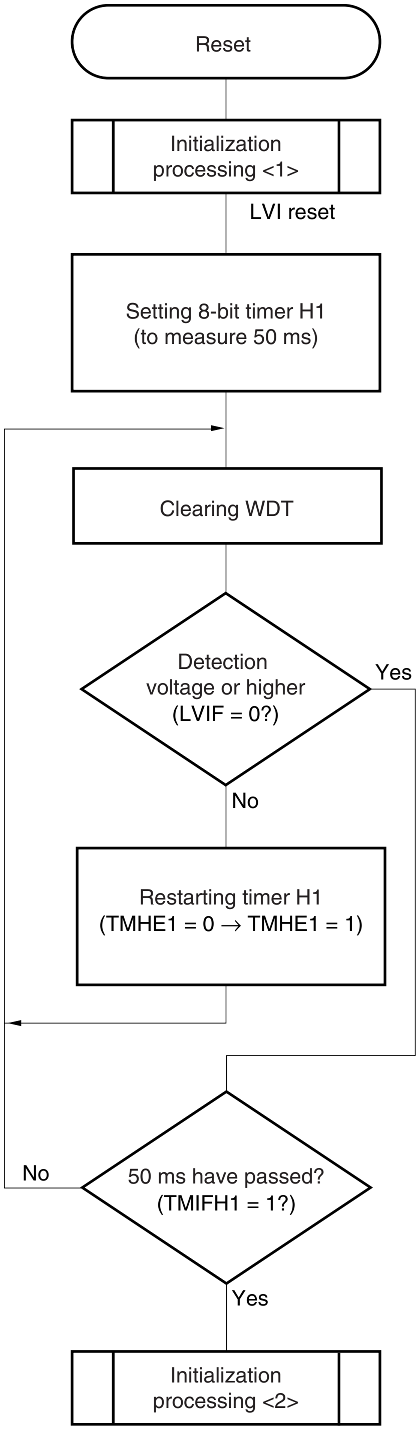
(b) In a system where the supply voltage fluctuation period is long in the vicinity of the LVI detection voltage, wait for the supply voltage fluctuation period, confirm that “supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})” when detecting the falling edge of V_{DD}, or “supply voltage (V_{DD}) < 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 24-9. Example of Software Processing After Reset Release (1/2)

- If supply voltage fluctuation is 50 ms or less in vicinity of LVI detection voltage


Reset
Initialization processing <1>
LVI reset
Setting 8-bit timer H1 (to measure 50 ms)
Clearing WDT
Detection voltage or higher (LVIF = 0?)
Yes
No
Restarting timer H1 (TMHE1 = 0 → TMHE1 = 1)
No
50 ms have passed? (TMIFH1 = 1?)
Yes
Initialization processing <2>

; Check the reset source[Note]
Initialize the port.

; f_{PRS} = Internal high-speed oscillation clock (8.4 MHz (MAX.)) (default)
Source: f_{PRS} (8.4 MHz (MAX.))/2^{12},
Where comparison value = 102: ≅ 50 ms
Timer starts (TMHE1 = 1).

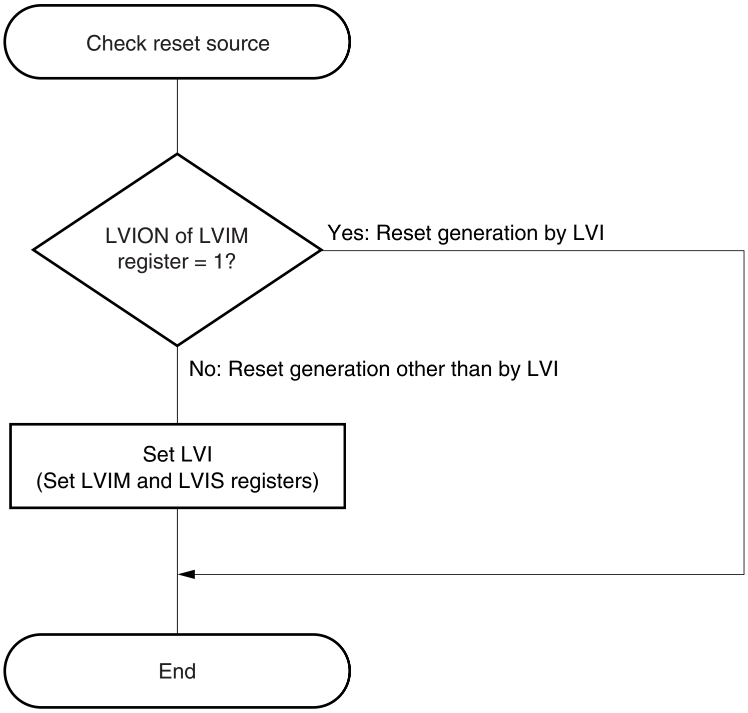
; The timer counter is cleared and the timer is started.

; Setting of division ratio of system clock,
such as setting of timer or A/D converter

Note A flowchart is shown on the next page.

Figure 24-9. Example of Software Processing After Reset Release (2/2)

- Checking reset cause


Check reset source
LVION of LVIM register = 1?
Yes: Reset generation by LVI
No: Reset generation other than by LVI
Set LVI
(Set LVIM and LVIS registers)
End

CHAPTER 25 OPTION BYTE

25.1 Functions of Option Bytes

The flash memory at 0080H to 0084H of the 78K0/Dx2 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 POC mode operation (LVISTART = 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 detected at 2.7 V (TYP.), in the same manner as on power application.
- During LVI operation stop mode (LVISTART = 0)

  The LVI operation stops.

Caution LVISTART 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, LVI operation stop 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.

25.2 Format of Option Byte

The format of the option byte is shown below.

Figure 25-1. 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}/fosc (4.27 ms)
0	0	1	2^{11}/fosc (8.53 ms)
0	1	0	2^{12}/fosc (17.07 ms)
0	1	1	2^{13}/fosc (34.13 ms)
1	0	0	2^{14}/fosc (68.27 ms)
1	0	1	2^{15}/fosc (136.53 ms)
1	1	0	2^{16}/fosc (273.07 ms)
1	1	1	2^{17}/fosc (546.13 ms)

LSROSC	Internal low-speed oscillator operation
0	Can be stopped by software (stopped when 1 is written to bit 0 (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 0 (LSRSTOP) of the internal oscillator mode register (RCM).**
4. **Be sure to clear bit 7 to 0.**

Remarks
1. fosc: Internal low-speed oscillation clock frequency
2. ( ): fosc = 264 kHz (MAX.)

Figure 25-1. 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	LVISTART

LVISTART	POC mode selection
0	LVI operation stop mode (default)
1	2.7 V POC mode

Notes 1. LVISTART 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, LVI operation stop 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 27 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: $2^{10}/f_{OSC}$,
                                ; Internal low-speed oscillator can be stopped by software.
          DB      00H           ; LVI operation stop 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 21 RESET FUNCTION**.

CHAPTER 26 FLASH MEMORY

The 78K0/Dx2 incorporates the flash memory to which a program can be written, erased, and overwritten while mounted on the board.

26.1 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 26-1 after a reset release.

Figure 26-1. 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
1	1	0	1024 bytes
Other than above			Setting prohibited

ROM3	ROM2	ROM1	ROM0	Internal ROM capacity selection
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

Caution To set the memory size, set IMS and then IXS. Set the memory size so that the internal ROM and internal expansion RAM areas do not overlap.

Table 26-1. Internal Memory Size Switching Register Settings

Flash Memory Version		IMS Setting
78K0/DE2	78K0/DF2	
μPD78F0836	μPD78F0838, 78F0840, 78F0842	C6H
μPD78F0844	μPD78F0846, 78F0848	C8H
μPD78F0837	μPD78F0839, 78F0841, 78F0843	CCH
μPD78F0845	μPD78F0847, 78F0849	CFH

26.2 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 26-2 after a reset release.

Figure 26-2. 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	0	1	0	1024 bytes
0	1	0	0	0	2048 bytes
Other than above					Setting prohibited

Caution To set memory size, set IMS and then IXS. Set memory size so that the internal ROM area and internal expansion RAM area do not overlap.

Table 26-2. Internal Expansion RAM Size Switching Register Settings

Flash Memory Version		IXS Setting
78K0/DE2	78K0/DF2	
μPD78F0836	μPD78F0838, 78F0840, 78F0842	0AH
μPD78F0844	μPD78F0846, 78F0848	
μPD78F0837	μPD78F0839, 78F0841, 78F0843	08H
μPD78F0845	μPD78F0847, 78F0849	

26.3 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/Dx2 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/Dx2 is mounted on the target system.

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

Table 26-3. Wiring between 78K0/Dx2 and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer			With CSI10			With UART60		
Signal Name	I/O	Pin Function	Pin Name	Pin No.		Pin Name	Pin No.	
				DE2	DF2		DE2	DF2
SI/RxD	Input	Receive signal	SO10/P12/INTP2	62	25	TxD60/P13/SEG23/TIOP30	19	31
SO/TxD	Output	Transmit signal	SI10/RxD61[Note 1]/P11/INTPR61[Note 1]	63	24	RxD60/P14/SEG22/TIOP20/INTPR60	20	32
SCK	Output	Transfer clock	$\overline{\text{SCK10}}$/TxD61[Note 1]/P10/INTP4	64	23	–	–	
CLK	Output	Clock to 78K0/Dx2	–[Note 2]	–		**Note 3**	**Note 3**	
<R> /RESET	Output	Reset signal	$\overline{\text{RESET}}$	8	12	$\overline{\text{RESET}}$	8	12
FLMD0	Output	Mode signal	FLMD0	11	15	FLMD0	11	15
V_{DD}	I/O	V_{DD} voltage generation/power monitoring	V_{DD}/EV_{DD}	16	20	V_{DD}/EV_{DD}	16	20
			AV_{REF}	47	59	AV_{REF}	47	59
GND	–	Ground	V_{SS}/EV_{SS}	15	19	V_{SS}/EV_{SS}	15	19
			AV_{SS}	48	60	AV_{SS}	48	60

Notes 1. 78K0/DF2 only.

2. Only the internal high-speed oscillation clock (f_{OSC8}) can be used when CSI10 is used.

3. Only the X1 clock (f_X) or external main system clock (f_{EXT}) can be used when UART60 is used. When using the clock output of the dedicated flash memory programmer, pin connection varies depending on the type of the dedicated flash memory programmer used.

- PG-FP5, FL-PR5, PG-FP4, FL-PR4: Connect CLK of the programmer to EXCLK/X2/P122/OCD0B.

Examples of the recommended connection when using the adapter for flash memory writing are shown below.

Figure 26-3. Example of Wiring Adapter for Flash Memory Writing in 3-Wire Serial I/O (CSI10) Mode

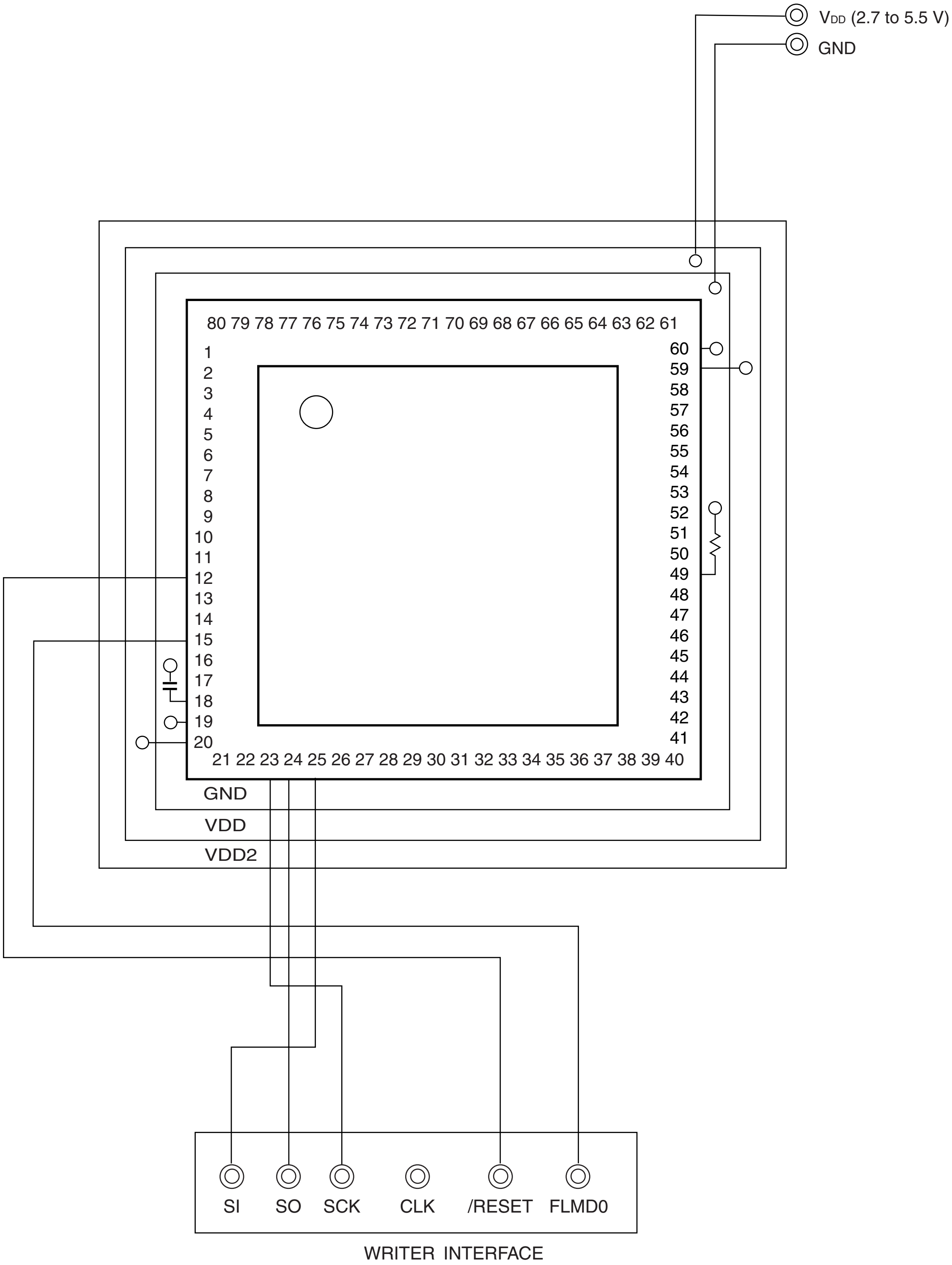
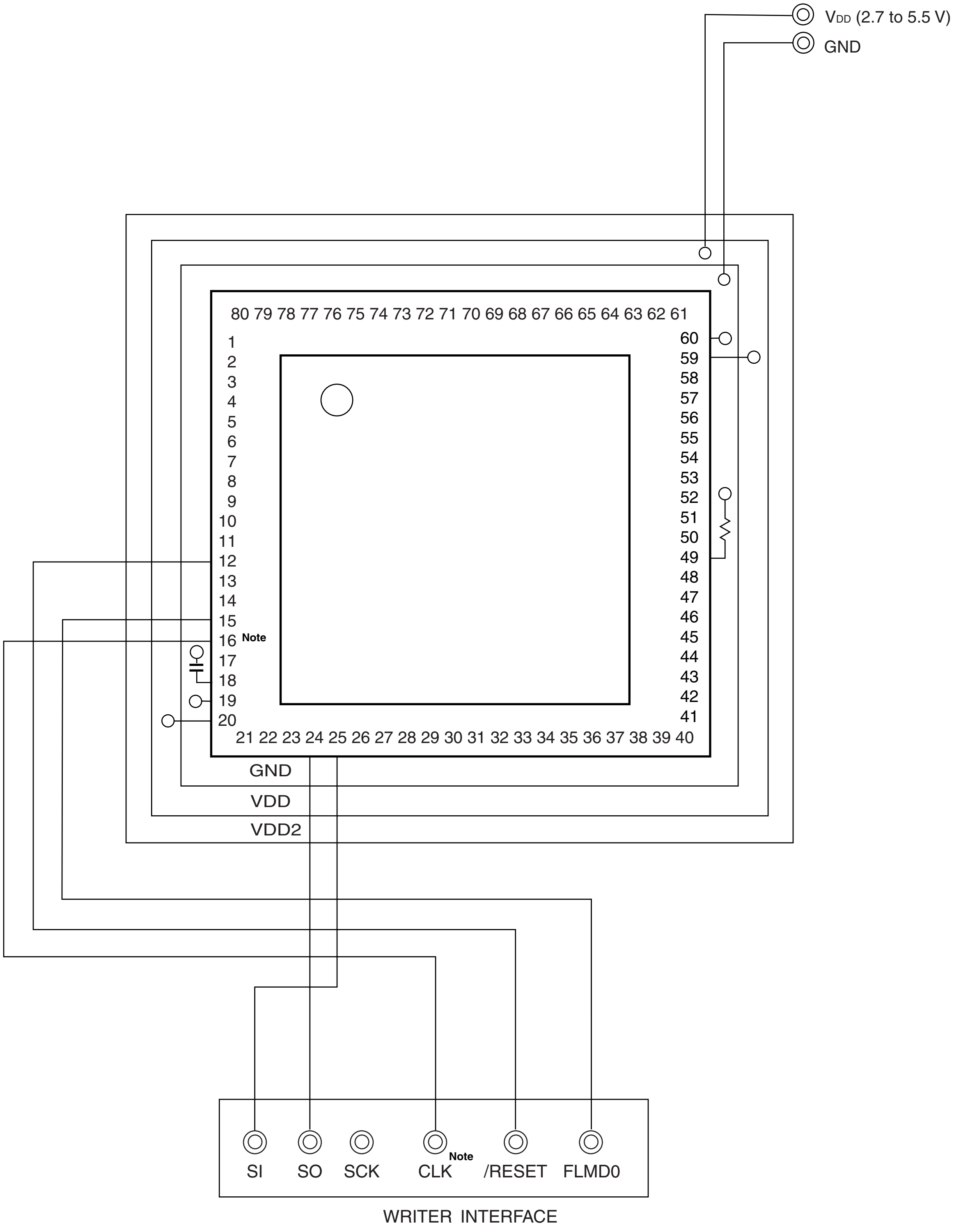


Figure 26-4. Example of Wiring Adapter for Flash Memory Writing in UART (UART60) Mode

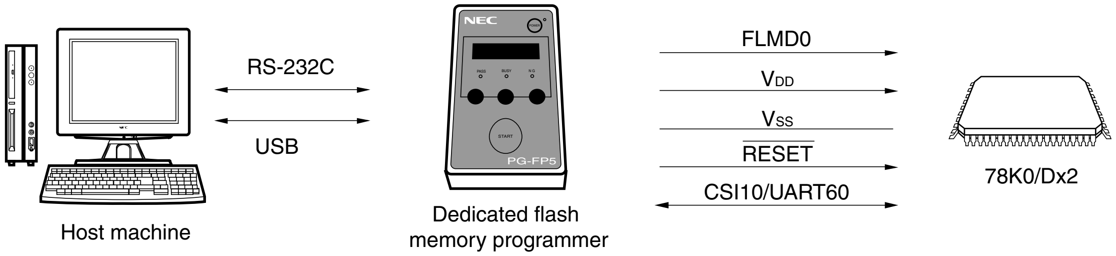


Note The above figure illustrates an example of wiring when using the clock output from the PG-FP5, FL-PR5, PG-FP4 or FL-PR4.

26.4 Programming Environment

The environment required for writing a program to the flash memory of the 78K0/Dx2 is illustrated below.

Figure 26-5. Environment for Writing Program to Flash Memory



A host machine that controls the dedicated flash memory programmer is necessary.

<R> Either CSI10 or a UART60 interface can optionally be used for the communication between the dedicated flash programmer and the microcontroller. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

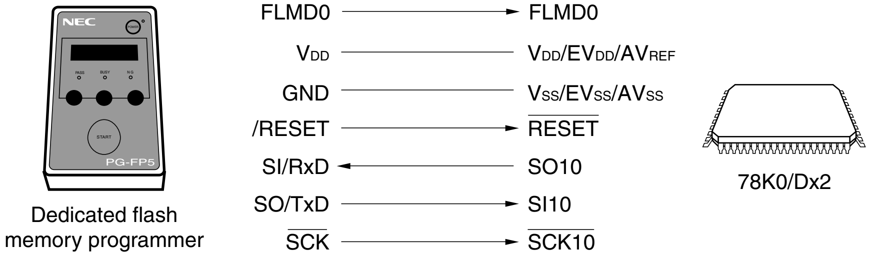
26.5 Communication Mode

Communication between the dedicated flash memory programmer and the 78K0/Dx2 is established by serial communication via CSI10 or UART60 of the 78K0/Dx2.

(1) CSI10

Transfer rate: 2.4 kHz to 2.5 MHz

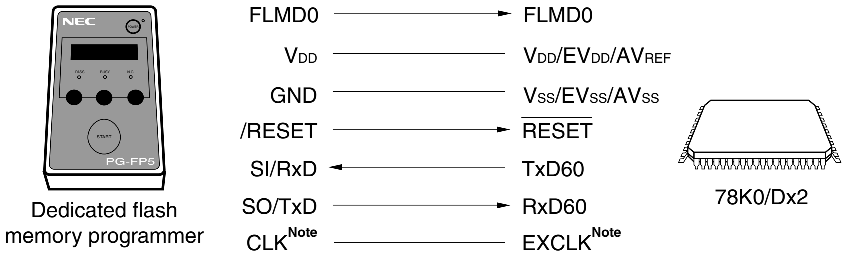
Figure 26-6. Communication with Dedicated Flash Memory Programmer (CSI10)



(2) UART60

Transfer rate: 115200 bps

Figure 26-7. Communication with Dedicated Flash Memory Programmer (UART60)



Note The above figure illustrates an example of wiring when using the clock output from the PG-FP5, FL-PR5, PG-FP4 or FL-PR4.

The dedicated flash memory programmer generates the following signals for the 78K0/Dx2. For details, refer to the user's manual for the PG-FP5, FL-PR5, PG-FP4, or FL-PR4.

Table 26-4. Pin Connection

Dedicated Flash memory programmer			78K0/Dx2	Connection	
Signal Name	I/O	Pin Function	Pin Name	CSI10	UART60
FLMD0	Output	Mode signal	FLMD0	◎	◎
V_{DD}	I/O	V_{DD} voltage generation/power monitoring	V_{DD}, EV_{DD}, AV_{REF}	◎	◎
GND	–	Ground	V_{SS}, EV_{SS}, AV_{SS}	◎	◎
CLK	Output	Clock output to 78K0/Dx2	**Note 1**	×[Note 2]	○[Note 1]
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	◎	◎
SI/RxD	Input	Receive signal	SO10/TxD60	◎	◎
SO/TxD	Output	Transmit signal	SI10/RxD60	◎	◎
SCK	Output	Transfer clock	$\overline{\text{SCK10}}$	◎	×

Notes 1. Only the X1 clock (f_X) or external main system clock (f_{EXT}) can be used when UART60 is used. When using the clock output of the dedicated flash memory programmer, pin connection varies depending on the type of the dedicated flash memory programmer used.

- PG-FP5, FL-PR5, PG-FP4, FL-PR4: Connect CLK of the programmer to EXCLK/X2/P122/OCD0B.

2. Only the internal high-speed oscillation clock (f_{OSC8}) can be used when CSI10 is used.

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.

26.6 Connection of Pins on Board

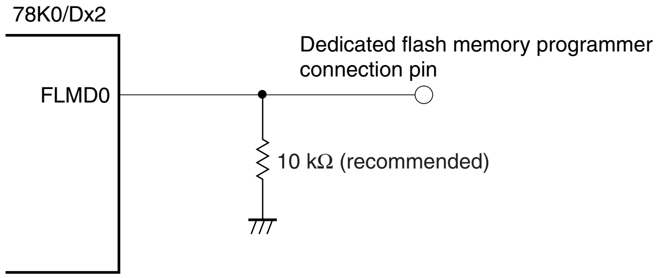
To write the flash memory on-board, connectors that connect the dedicated flash memory programmer must be provided on the target system. First provide a function that selects the normal operation mode or flash memory programming mode on the board.

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.

26.6.1 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 26-8. FLMD0 Pin Connection Example



26.6.2 Serial interface pins

The pins used by each serial interface are listed below.

Table 26-5. Pins Used by Each Serial Interface

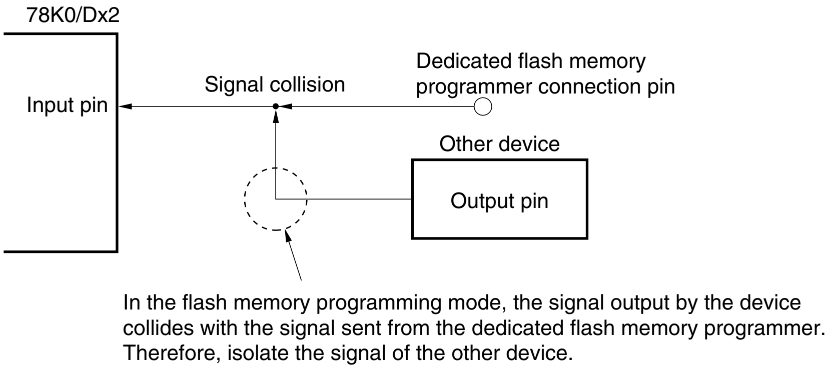
Serial Interface	Pins Used
CSI10	SO10, SI10, $\overline{\text{SCK10}}$
UART60	TxD60, RxD60

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 26-9. Signal Collision (Input Pin of Serial Interface)

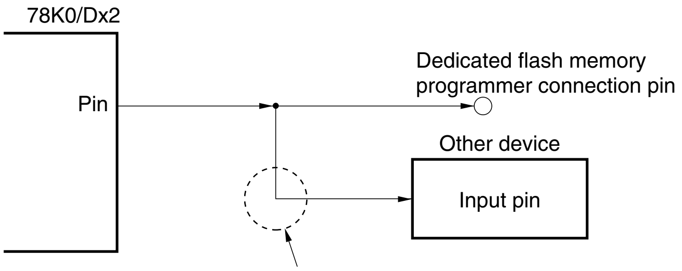


In the flash memory programming mode, the signal output by the device collides with the signal sent from the dedicated flash memory programmer. Therefore, isolate the signal of the other device.

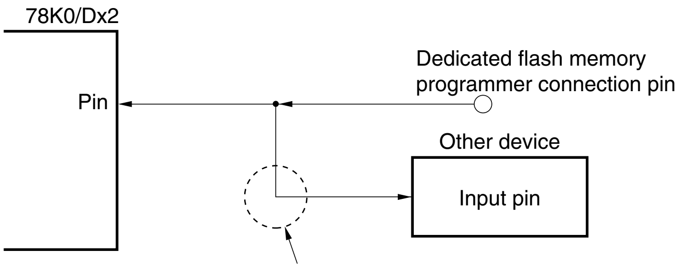
(2) Malfunction of other device

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 26-10. Malfunction of Other Device



If the signal output by the 78K0/Dx2 in the flash memory programming mode affects the other device, isolate the signal of the other device.



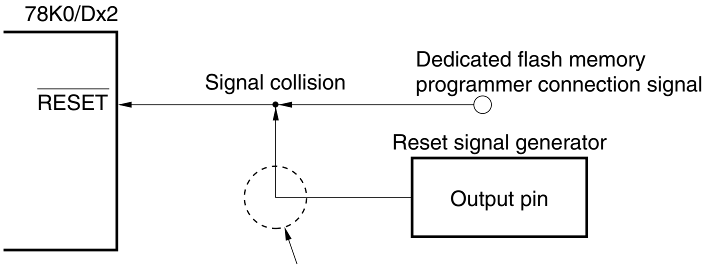
If the signal output by the dedicated flash memory programmer in the flash memory programming mode affects the other device, isolate the signal of the other device.

26.6.3 $\overline{\text{RESET}}$ 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 26-11. Signal Collision ($\overline{\text{RESET}}$ Pin)



In the flash memory programming mode, the signal output by the reset signal generator collides with the signal output by the dedicated flash memory programmer. Therefore, isolate the signal of the reset signal generator.

26.6.4 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.

26.6.5 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.

26.6.6 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 as follows.

- PG-FP5, FL-PR5, PG-FP4, FL-PR4: Connect CLK of the programmer to EXCLK/X2/P122/OCD0B.

Cautions 1. Only the internal high-speed oscillation clock (f_{OSC8}) can be used when CSI10 is used.

2. Only the X1 clock (f_X) or external main system clock (f_{EXT}) can be used when UART60 is used.

<R> **3. Connect P121/X1/OCD0A as follows when writing the flash memory with a flash memory programmer.**

- **P121/X1/OCD0A: When using this pin as a port, connect it to V_{SS} via a resistor (10 kΩ: recommended) (in the input mode) or leave it open (in the output mode).**

The above connection is not necessary when writing the flash memory by means of self programming.

<R> **4. Set P31 and P32 as follows when these pins are used for on-chip debug. In this case, P31/SEG5/OCD1A and P32/SEG6/OCD1B pins are used only as OCD1A or OCD1B functions.**

- P31/SEG5/OCD1A: Set to input mode (PM31 = 1) and port mode (PF31 = 0).

- P32/SEG6/OCD1B: Set to port mode (PF32 = 0).

When P31/SEG5 and P32/SEG6 functions need to be evaluated with on-chip debug, use OCD0A and OCD0B.

26.6.7 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.

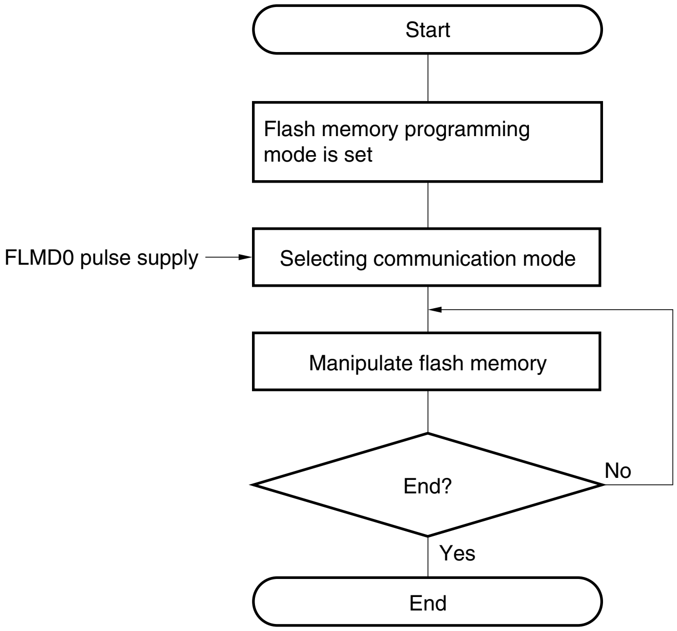
Supply the same other power supplies (EV_{DD}, EV_{SS}, AV_{REF}, and AV_{SS}) as those in the normal operation mode.

26.7 Programming Method

26.7.1 Controlling flash memory

The following figure illustrates the procedure to manipulate the flash memory.

Figure 26-12. Flash Memory Manipulation Procedure


Start
Flash memory programming mode is set
FLMD0 pulse supply
Selecting communication mode
Manipulate flash memory
End?
No
Yes
End

26.7.2 Flash memory programming mode

To rewrite the contents of the flash memory by using the dedicated flash memory programmer, set the 78K0/Dx2 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 26-13. Flash Memory Programming Mode

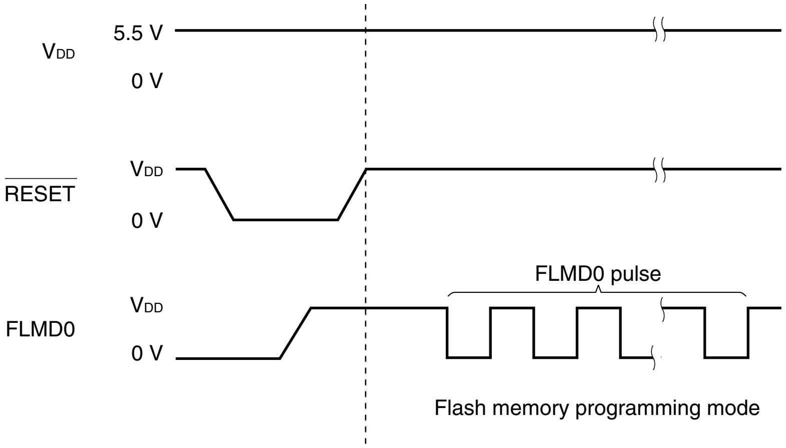


Table 26-6. Relationship between FLMD0 Pin and Operation Mode after Reset Release

FLMD0	Operation Mode
0	Normal operation mode
V_{DD}	Flash memory programming mode

26.7.3 Selecting communication mode

In the 78K0/Dx2, a communication mode is selected by inputting pulses (up to 8 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 26-7. Communication Modes

Communication Mode	Standard Setting[Note 1]				Pins Used	Peripheral Clock	Number of FLMD0 Pulses
	Port	Speed	Frequency	Multiply Rate			
UART (UART60)	UART-Ext-Osc	115200 bps[Note 2]	2 to 20 MHz[Note 3]	1.0	TxD60, RxD60	f_X	0
	UART-Ext-FP4CK					f_{EXT}	3
3-wire serial I/O (CSI10)	CSI-Internal-Osc	2.4 kHz to 2.5 MHz	–		SO10, SI10, $\overline{\text{SCK10}}$	f_{OSC8}	8

Notes 1. Selection items for Standard settings on GUI of the flash memory programmer.

2. 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.

3. The possible setting range differs depending on the voltage. For details, refer to the chapter of electrical specifications.

Caution When UART60 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

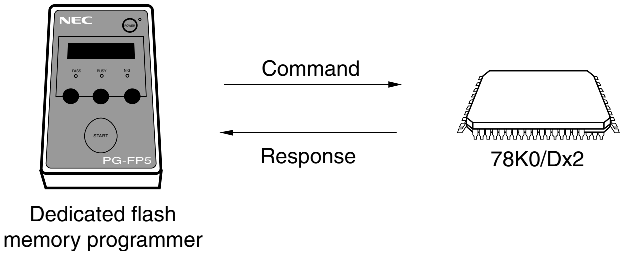
f_{EXT}: External main system clock

f_{OSC8}: Internal high-speed oscillation clock

26.7.4 Communication commands

The 78K0/Dx2 communicates with the dedicated flash memory programmer by using commands. The signals sent from the flash memory programmer to the 78K0/Dx2 are called commands, and the signals sent from the 78K0/Dx2 to the dedicated flash memory programmer are called response.

Figure 26-14. Communication Commands



The flash memory control commands of the 78K0/Dx2 are listed in the table below. All these commands are issued from the programmer and the 78K0/Dx2 perform processing corresponding to the respective commands.

Table 26-8. 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/Dx2 information (such as the part number and flash memory configuration).
	Version Get	Gets the 78K0/Dx2 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/Dx2 return a response for the command issued by the dedicated flash memory programmer. The response names sent from the 78K0/Dx2 are listed below.

Table 26-9. Response Names

Command Name	Function
ACK	Acknowledges command/data.
NAK	Acknowledges illegal command/data.

26.8 Security Settings

The 78K0/Dx2 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 26-10 shows the relationship between the erase and write commands when the 78K0/Dx2 security function is enabled.

Table 26-10. 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 26-11 shows how to perform security settings in each programming mode.

Table 26-11. 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.

26.9 Processing Time for Each Command When PG-FP4 or PG-FP5 Is Used (Reference)

The following table shows the processing time for each command (reference) when the PG-FP4 or PG-FP5 is used as a dedicated flash memory programmer.

Table 26-12. Processing Time for Each Command When PG-FP4 or PG-FP5 Is Used (Reference)

- **μPD78F0849 (internal ROM capacity: 60 KB)**

Command of PG-FP4 or PG-FP5	Port: CSI-Internal-OSC (Internal high-speed oscillation clock (f_{OSC8})), Speed: 2.5 MHz	Port: UART-Ext-FP4CK (External main system clock (f_{EXT})), Speed: 115,200 bps	
		Frequency: 2.0 MHz	Frequency: 20 MHz
Signature	0.5 s (TYP.)	0.5 s (TYP.)	0.5 s (TYP.)
Blankcheck	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)
Erase	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)
Program	5 s (TYP.)	9 s (TYP.)	9 s (TYP.)
Verify	2 s (TYP.)	6.5 s (TYP.)	6.5 s (TYP.)
E.P.V	6 s (TYP.)	10.5 s (TYP.)	10.5 s (TYP.)
Checksum	0.5 s (TYP.)	1 s (TYP.)	1 s (TYP.)
Security	0.5 s (TYP.)	0.5 s (TYP.)	0.5 s (TYP.)

Caution When executing boot swapping, do not use the E.P.V. command with the dedicated flash memory programmer.

26.10 Flash Memory Programming by Self-Programming

The 78K0/Dx2 supports 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 78K0/Dx2 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.

Remark For details of the self-programming function and the 78K0/Dx2 self-programming library, refer to a separate document to be published (document name: 78K0/Dx2 Application Note, release schedule: Pending).

Cautions 1. The self-programming function cannot be used when the CPU operates with the subsystem clock.

2. Input a high level to the FLMD0 pin during self-programming.

3. 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.

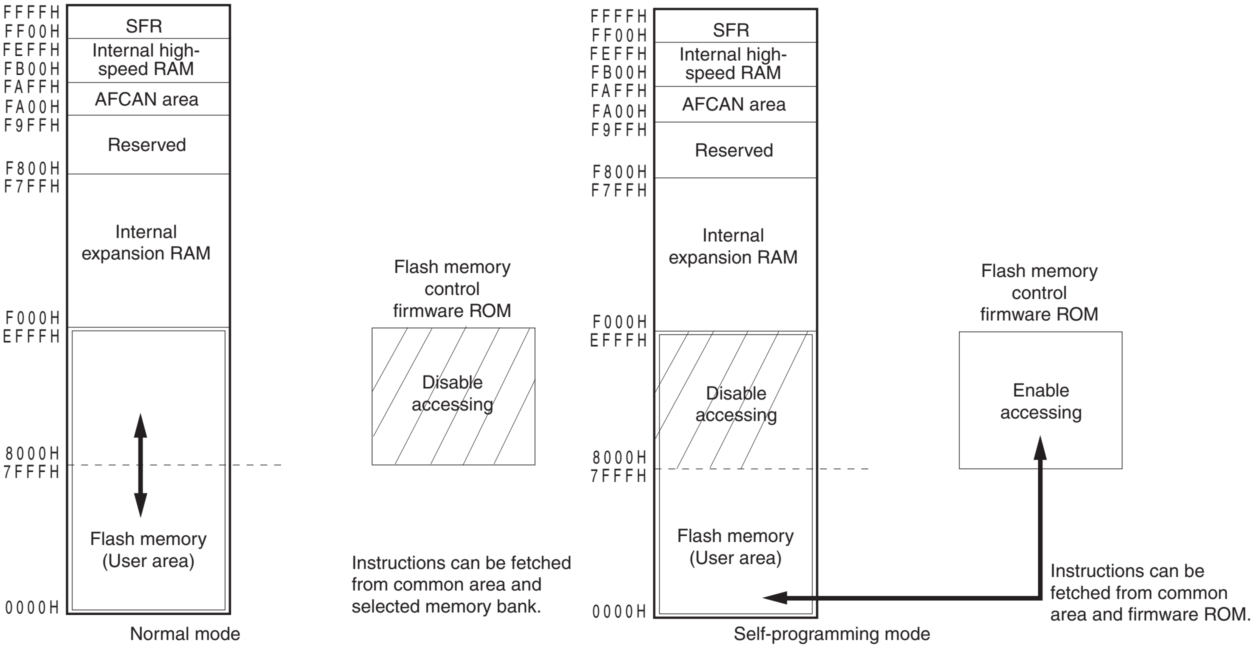
4. Self-programming is also stopped by an interrupt request that is not masked even in the DI status. To prevent this, mask the interrupt by using the interrupt mask flag registers (MK0L, MK0H, MK1L, and MK1H).

5. Self-programming is executed with the internal high-speed oscillation clock. If the CPU operates with the X1 clock or external main system clock, the oscillation stabilization wait time of the internal high-speed oscillation clock elapses during self-programming.

(Caution 6 is listed on the next page.)

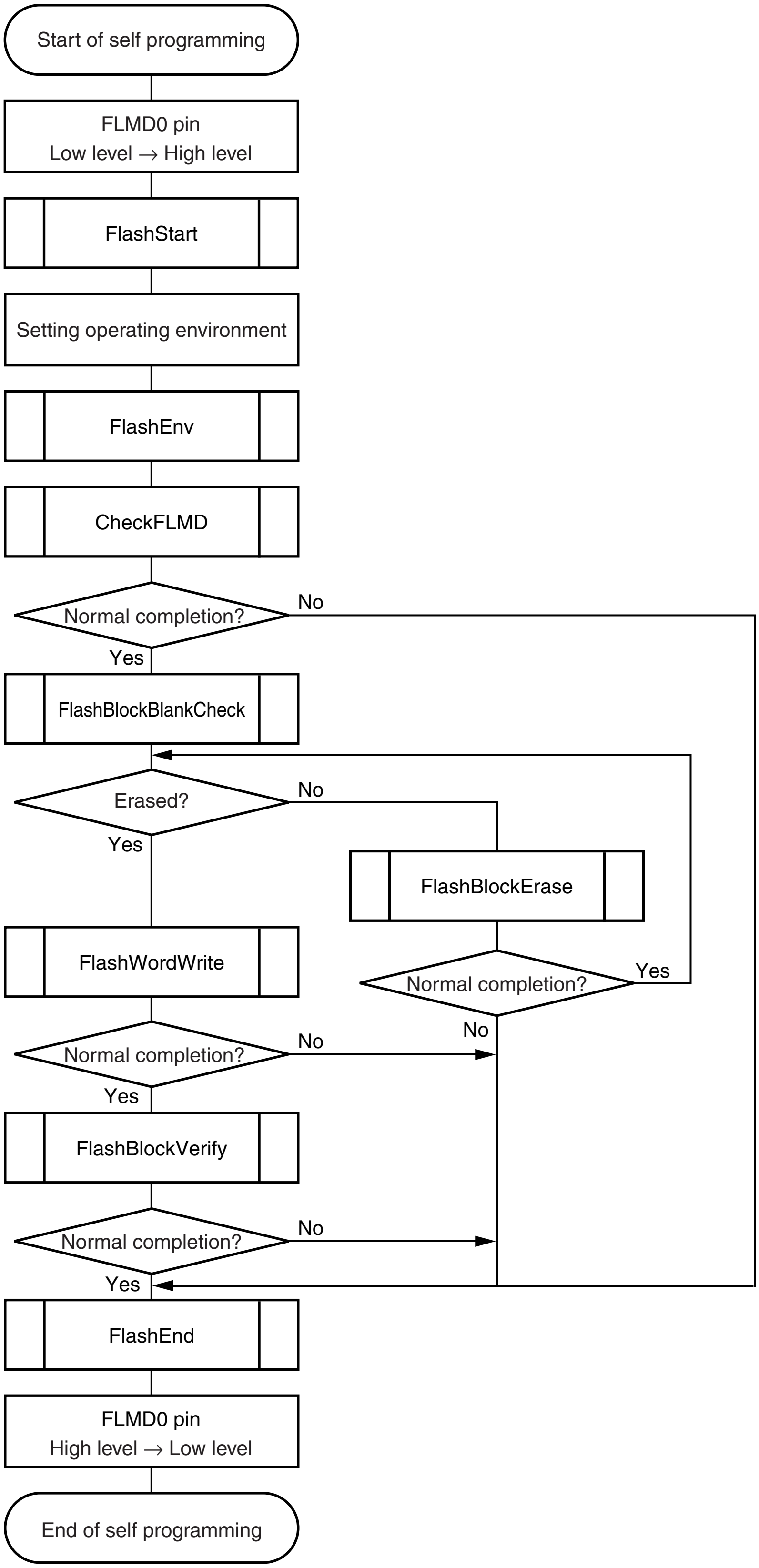
Cautions 6. Allocate the entry program for self-programming in the common area of 0000H to 7FFFH.

Figure 26-15. Operation Mode and Memory Map for Self-Programming (μPD78F0849)



The following figure illustrates a flow of rewriting the flash memory by using a self programming library.

Figure 26-16. Flow of Self-Programming (Rewriting Flash Memory)



The following table shows the processing time and interrupt response time for the self programming library.

Table 26-13. Processing Time for Self Programming Library (1/3)

(1) When internal high-speed oscillation clock is used and entry RAM is located outside short direct addressing range

Library Name		Processing Time (μs)			
		Normal Model of C Compiler		Static Model of C Compiler/Assembler	
		Min.	Max.	Min.	Max.
Self programming start library		4.25			
Initialize library		977.75			
Mode check library		753.875		753.125	
Block blank check library		12770.875		12765.875	
Block erase library		36909.5	356318	36904.5	356296.25
Word write library		1214 (1214.375)	2409 (2409.375)	1207 (1207.375)	2402 (2402.375)
Block verify library		25618.875		25613.875	
Self programming end library		4.25			
Get information library	Option value: 03H	871.25 (871.375)		866 (866.125)	
	Option value: 04H	863.375 (863.5)		858.125 (858.25)	
	Option value: 05H	1024.75 (1043.625)		1037.5 (1038.375)	
Set information library		105524.75	790809.375	105523.75	790808.375
EEPROM write library		1496.5 (1496.875)	2691.5 (2691.875)	1489.5 (1489.875)	2684.5 (2684.875)

(2) When internal high-speed oscillation clock is used and entry RAM is located in short direct addressing range

Library Name		Processing Time (μs)			
		Normal Model of C Compiler		Static Model of C Compiler/Assembler	
		Min.	Max.	Min.	Max.
Self programming start library		4.25			
Initialize library		443.5			
Mode check library		219.625		218.875	
Block blank check library		12236.625		12231.625	
Block erase library		36363.25	355771.75	36358.25	355750
Word write library		679.75 (680.125)	1874.75 (1875.125)	672.75 (673.125)	1867.75 (1868.125)
Block verify library		25072.625		25067.625	
Self programming end library		4.25			
Get information library	Option value: 03H	337 (337.125)		331.75 (331.875)	
	Option value: 04H	329.125 (239.25)		323.875 (324)	
	Option value: 05H	502.25 (503.125)		497 (497.875)	
Set information library		104978.5	541143.125	104977.5	541142.125
EEPROM write library		962.25 (962.625)	2157.25 (2157.625)	955.25 (955.625)	2150.25 (2150.625)

Remarks 1. Values in parentheses indicate values when a write start address structure is located other than in the internal high-speed RAM.

2. The above processing times are those during stabilized operation of the internal high-speed oscillator (RSTS = 1).

3. RSTS: Bit 7 of the internal oscillation mode register (RCM)

Table 26-13. Processing Time for Self Programming Library (2/3)

(3) When high-speed system clock (X1 oscillation or external clock input) is used and entry RAM is located outside short direct addressing range

Library Name		Processing Time (μs)			
		Normal Model of C Compiler		Static Model of C Compiler/Assembler	
		Min.	Max.	Min.	Max.
Self programming start library		34/fCPU			
Initialize library		49/fCPU + 485.8125			
Mode check library		35/fCPU + 374.75		29/fCPU + 374.75	
Block blank check library		174/fCPU + 6382.0625		134/fCPU + 6382.0625	
Block erase library		174/fCPU + 31093.875	174/fCPU + 298948.125	134/fCPU + 31093.875	134/fCPU + 298948.125
Word write library		318 (321)/fCPU + 644.125	318 (321)/fCPU + 1491.625	262 (265)/fCPU + 644.125	262 (265)/fCPU + 1491.625
Block verify library		174/fCPU + 13448.5625		134/fCPU + 13448.5625	
Self programming end library		34/fCPU			
Get information library	Option value: 03H	171 (172)/fCPU + 432.4375		129 (130)/fCPU + 432.4375	
	Option value: 04H	181 (182)/fCPU + 427.875		139 (140)/fCPU + 427.875	
	Option value: 05H	404 (411)/fCPU + 496.125		362 (369)/fCPU + 496.125	
Set information library		75/fCPU + 79157.6875	75/fCPU + 652400	67fCPU + 79157.6875	67fCPU + 652400
EEPROM write library		318 (321)/fCPU + 799.875	318 (321)/fCPU + 1647.375	262 (265)/fCPU + 799.875	262 (265)/fCPU + 1647.375

Remarks 1. Values in parentheses indicate values when a write start address structure is located other than in the internal high-speed RAM.

2. The above processing times are those during stabilized operation of the internal high-speed oscillator (RSTS = 1).

3. fCPU: CPU operation clock frequency

4. RSTS: Bit 7 of the internal oscillation mode register (RCM)

Table 26-13. Processing Time for Self Programming Library (3/3)

(4) When high-speed system clock (X1 oscillation or external clock input) is used and entry RAM is located in short direct addressing range

Library Name		Processing Time (μs)			
		Normal Model of C Compiler		Static Model of C Compiler/Assembler	
		Min.	Max.	Min.	Max.
Self programming start library		34/f_{CPU}			
Initialize library		49/f_{CPU} + 224.6875			
Mode check library		35/f_{CPU} + 113.625		29/f_{CPU} + 113.625	
Block blank check library		174/f_{CPU} + 6120.9375		134/f_{CPU} + 6120.9375	
Block erase library		174/f_{CPU} + 30820.75	174/f_{CPU} + 298675	134/f_{CPU} + 30820.75	134/f_{CPU} + 298675
Word write library		318 (321)/f_{CPU} + 383	318 (321)/f_{CPU} + 1230.5	262 (265)/f_{CPU} + 383	262 (265)/f_{CPU} + 1230.5
Block verify library		174/f_{CPU} + 13175.4375		134/f_{CPU} + 13175.4375	
Self programming end library		34/f_{CPU}			
Get information library	Option value: 03H	171 (172)/f_{CPU} + 171.3125		129 (130)/f_{CPU} + 171.3125	
	Option value: 04H	181 (182)/f_{CPU} + 166.75		139 (140)/f_{CPU} + 166.75	
	Option value: 05H	404 (411)/f_{CPU} + 231.875		362 (369)/f_{CPU} + 231.875	
Set information library		75/f_{CPU} + 78884.5625	75/f_{CPU} + 527566.875	67f_{CPU} + 78884.5625	67f_{CPU} + 527566.875
EEPROM write library		318 (321)/f_{CPU} + 538.75	318 (321)/f_{CPU} + 1386.25	262 (265)/f_{CPU} + 538.75	262 (265)/f_{CPU} + 1386.25

Remarks 1. Values in parentheses indicate values when a write start address structure is located other than in the internal high-speed RAM.

2. The above processing times are those during stabilized operation of the internal high-speed oscillator (RSTS = 1).

3. f_{CPU}: CPU operation clock frequency

4. RSTS: Bit 7 of the internal oscillation mode register (RCM)

Table 26-14. Interrupt Response Time for Self Programming Library (1/2)

(1) When internal high-speed oscillation clock is used

Library Name	Interrupt Response Time (μs (Max.))			
	Normal Model of C Compiler		Static Model of C Compiler/Assembler	
	Entry RAM location is outside short direct addressing range	Entry RAM location is in short direct addressing range	Entry RAM location is outside short direct addressing range	Entry RAM location is in short direct addressing range
Block blank check library	933.6	668.6	927.9	662.9
Block erase library	1026.6	763.6	1020.9	757.9
Word write library	2505.8	1942.8	2497.8	1934.8
Block verify library	958.6	693.6	952.9	687.9
Set information library	476.5	211.5	475.5	210.5
EEPROM write library	2760.8	2168.8	2759.5	2167.5

Remarks 1. The above interrupt response times are those during stabilized operation of the internal high-speed oscillator (RSTS = 1).

2. RSTS: Bit 7 of the internal oscillation mode register (RCM)

(2) When high-speed system clock is used (normal model of C compiler)

Library Name	Interrupt Response Time (μs (Max.))			
	RSTOP = 0, RSTS = 1		RSTOP = 1	
	Entry RAM location is outside short direct addressing range	Entry RAM location is in short direct addressing range	Entry RAM location is outside short direct addressing range	Entry RAM location is in short direct addressing range
Block blank check library	$179/f_{CPU} + 507$	$179/f_{CPU} + 407$	$179/f_{CPU} + 1650$	$179/f_{CPU} + 714$
Block erase library	$179/f_{CPU} + 559$	$179/f_{CPU} + 460$	$179/f_{CPU} + 1702$	$179/f_{CPU} + 767$
Word write library	$333/f_{CPU} + 1589$	$333/f_{CPU} + 1298$	$333/f_{CPU} + 2732$	$333/f_{CPU} + 1605$
Block verify library	$179/f_{CPU} + 518$	$179/f_{CPU} + 418$	$179/f_{CPU} + 1661$	$179/f_{CPU} + 725$
Set information library	$80/f_{CPU} + 370$	$80/f_{CPU} + 165$	$80/f_{CPU} + 1513$	$80/f_{CPU} + 472$
EEPROM write library[Note]	$29/f_{CPU} + 1759$	$29/f_{CPU} + 1468$	$29/f_{CPU} + 1759$	$29/f_{CPU} + 1468$
	$333/f_{CPU} + 834$	$333/f_{CPU} + 512$	$333/f_{CPU} + 2061$	$333/f_{CPU} + 873$

Note The longer value of the EEPROM write library interrupt response time becomes the Max. value, depending on the value of f_{CPU}.

Remarks 1. f_{CPU}: CPU operation clock frequency

2. RSTOP: Bit 0 of the internal oscillation mode register (RCM)

3. RSTS: Bit 7 of the internal oscillation mode register (RCM)

Table 26-14. Interrupt Response Time for Self Programming Library (2/2)

(3) When high-speed system clock is used (static model of C compiler/assembler)

Library Name	Interrupt Response Time (μs (Max.))			
	RSTOP = 0, RSTS = 1		RSTOP = 1	
	Entry RAM location is outside short direct addressing range	Entry RAM location is in short direct addressing range	Entry RAM location is outside short direct addressing range	Entry RAM location is in short direct addressing range
Block blank check library	$136/f_{CPU} + 507$	$136/f_{CPU} + 407$	$136/f_{CPU} + 1650$	$136/f_{CPU} + 714$
Block erase library	$136/f_{CPU} + 559$	$136/f_{CPU} + 460$	$136/f_{CPU} + 1702$	$136/f_{CPU} + 767$
Word write library	$272/f_{CPU} + 1589$	$272/f_{CPU} + 1298$	$272/f_{CPU} + 2732$	$272/f_{CPU} + 1605$
Block verify library	$136/f_{CPU} + 518$	$136/f_{CPU} + 418$	$136/f_{CPU} + 1661$	$136/f_{CPU} + 725$
Set information library	$72/f_{CPU} + 370$	$72/f_{CPU} + 165$	$72/f_{CPU} + 1513$	$72/f_{CPU} + 472$
EEPROM write library[Note]	$19/f_{CPU} + 1759$ $268/f_{CPU} + 834$	$19/f_{CPU} + 1468$ $268/f_{CPU} + 512$	$19/f_{CPU} + 1759$ $268/f_{CPU} + 2061$	$19/f_{CPU} + 1468$ $268/f_{CPU} + 873$

Note The longer value of the EEPROM write library interrupt response time becomes the Max. value, depending on the value of f_{CPU}.

Remarks 1. f_{CPU}: CPU operation clock frequency

2. RSTOP: Bit 0 of the internal oscillation mode register (RCM)

3. RSTS: Bit 7 of the internal oscillation mode register (RCM)

<R>

26.11 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/Dx2, 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.

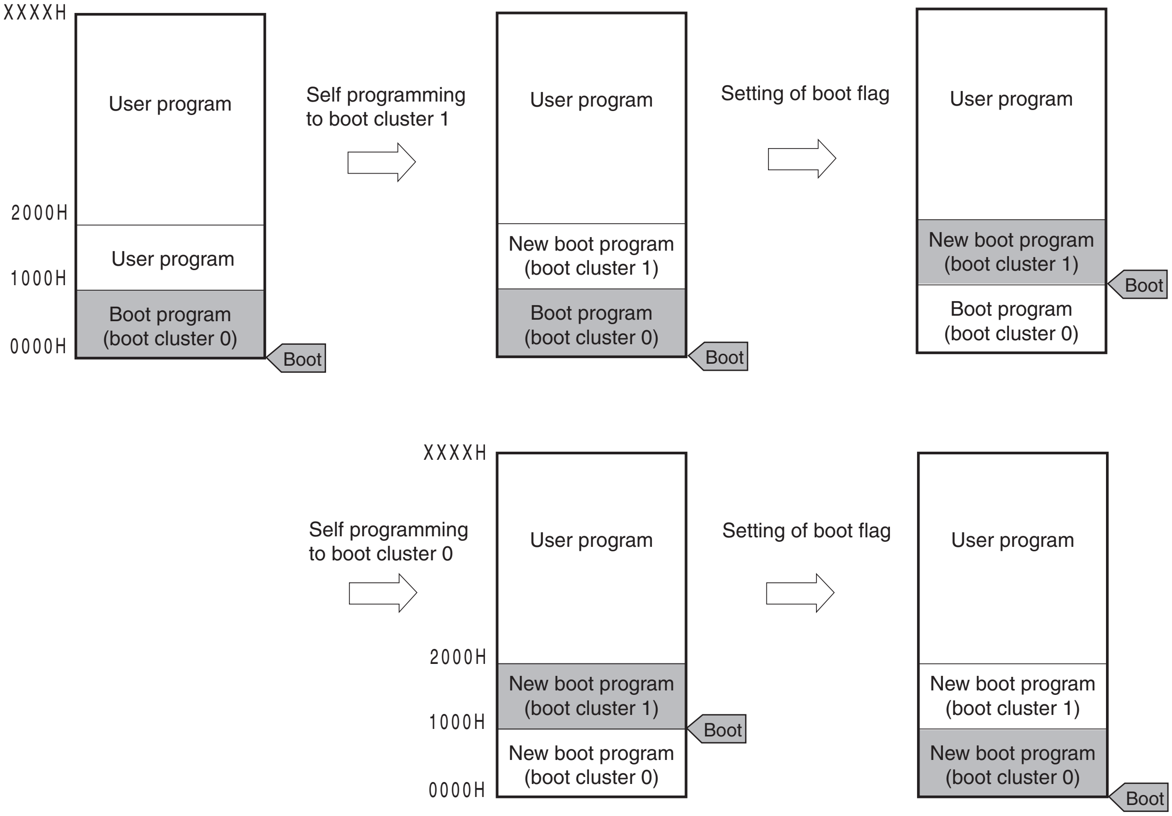
If the program has been correctly written to boot cluster 0, restore the original boot area by using the set information function of the firmware of the 78K0/Dx2.

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

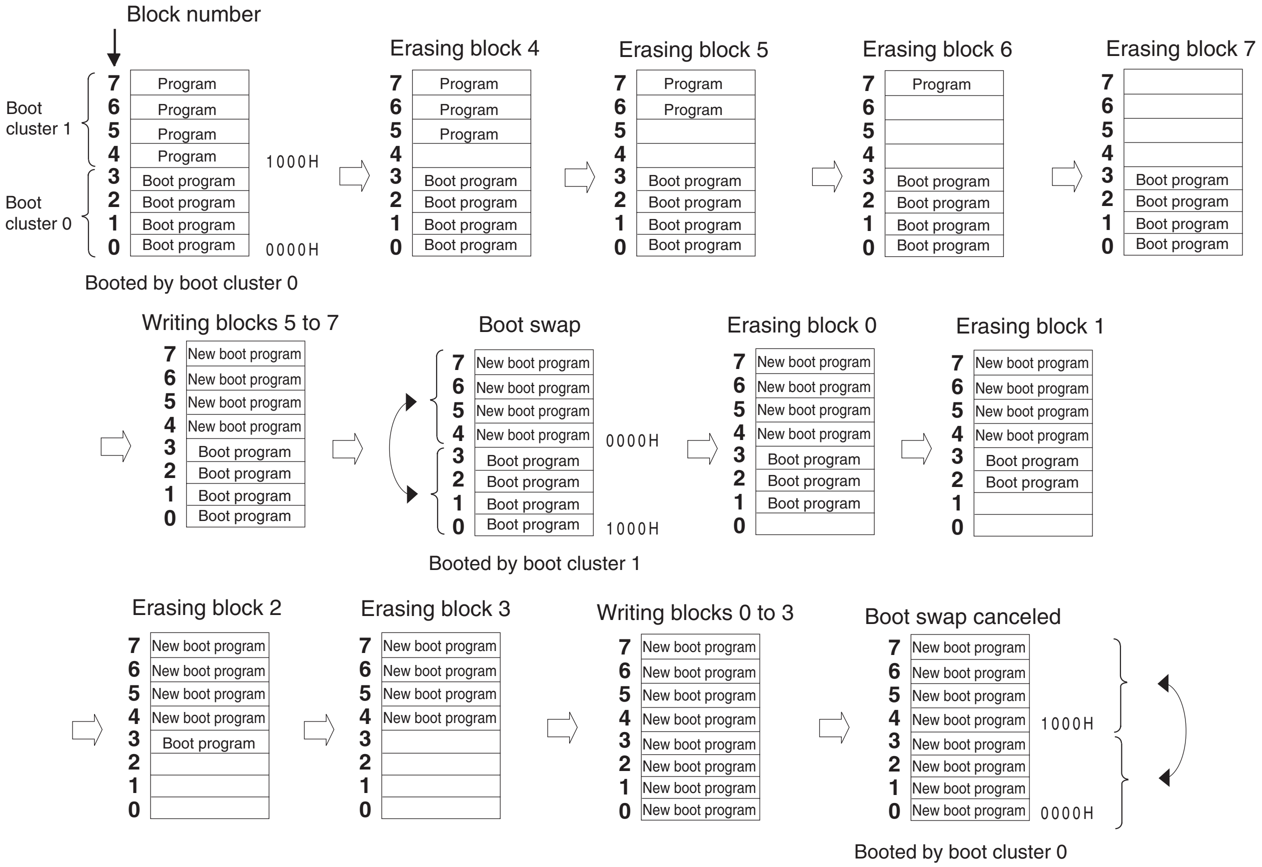
Caution When executing boot swapping, do not use the E.P.V command with the dedicated flash memory programmer.

Figure 26-17. Boot Swap Function



Remark Boot cluster 1 becomes 0000H to 0FFFH when a reset is generated after the boot flag has been set.

Figure 26-18. Example of Executing Boot Swapping



<R> 26.12 Creating ROM Code to Place Order for Programmed Flash Product

Before placing an order with NEC Electronics for a previously written product, the ROM code for the order must be created.

To create the ROM code, use the Hex Consolidation Utility (hereafter abbreviated to HCU) on the finished programs (hex files) and optional data (such as security settings for flash memory programs).

<R> The HCU is a software to generate a ROM code that is necessary for ordering the flash product programmed by NEC Electronics.

The HCU can be downloaded at the NEC Electronics website.

(1) Website

<R> http://www.necel.com/micro/en/ods → Click **Version-up Service**.

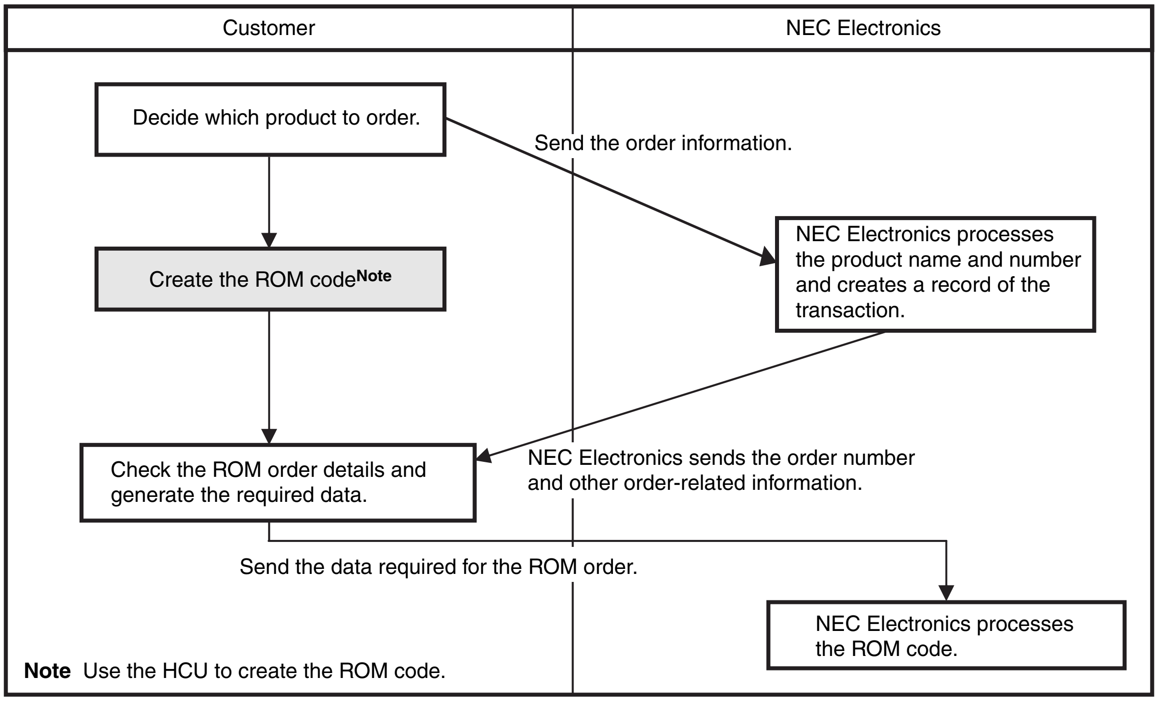
(2) Downloading the HCU

<R> To download the HCU software, click **Programmed_Flash_Products_Software** and then **HCU_GUI**.

Remark For details about how to install and use the HCU, see the materials (the user's manual) that comes with the HCU at the above website.

<R> 26.12.1 Procedures for ROM code ordering process using HCU

Use the HCU to create the ROM code by following the procedure below, and then place your order with NEC Electronics. For details, see the ROM Code Ordering Method Information (C10302J).


Customer
NEC Electronics
Decide which product to order.
Send the order information.
NEC Electronics processes the product name and number and creates a record of the transaction.
Create the ROM codeNote
Check the ROM order details and generate the required data.
NEC Electronics sends the order number and other order-related information.
Send the data required for the ROM order.
NEC Electronics processes the ROM code.

<R> **Note** Use the HCU to create the ROM code.

CHAPTER 27 ON-CHIP DEBUG FUNCTION

27.1 Outline of Functions

The 78K0/Dx2 uses the V_{DD}, FLMD0, $\overline{\text{RESET}}$, X1 (or P31), X2 (or P32), and V_{SS} pins to communicate with the host machine via an on-chip debug emulator (QB-78K0MINI or QB-MINI2). Whether X1 and P31, or X2 and P32 are used can be selected.

Caution The 78K0/Dx2 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.

27.2 Connection with MINICUBE or QB-MINI2

In order to connect QB-78K0MINI or QB-MINI2, it is necessary to mount the connector for emulator connection, and the circuit for connection on a target system.

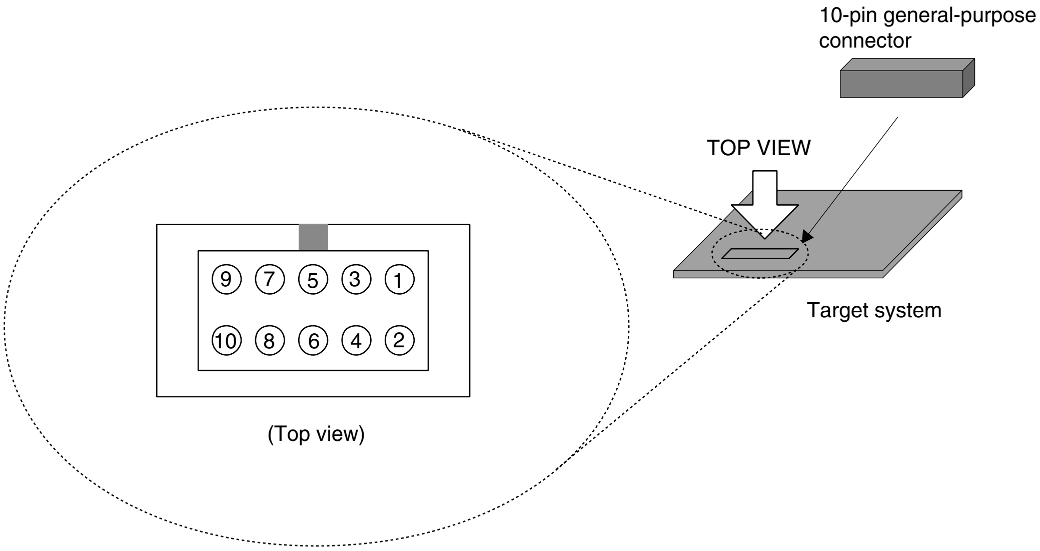
The connector for OCD (a two-row 2.54 pitch type connector, with reverse-insertion blocker) is described below.

- Recommended connectors: (straight) HIF3FC-10PA-2.54DSA (manufactured by Hirose Electric Co., Ltd.)
  (right angle) HIF3FC-10PA-2.54DS (manufactured by Hirose Electric Co., Ltd.))

Pin No.	Name	IN/OUT	Remark
1	RESET_IN	IN	Target reset input signal
2	$\overline{\text{RESET\_OUT}}$	OUT	Reset signal output to target device
3	FLMD0	OUT	Output signal[Note] used to control on-chip debugging functions
4	V_{DD}_IN	IN	This signal is used to generate an interface output signal when the target system's V_{DD} is detected.
5	X2	IN/OUT	Bidirectional signal used for data communications
6	GND	–	Connected to GND.
7	X1	OUT	Output signal used for clock supply
8	GND	–	Connected to GND.
9	RESERVED	–	Open
10	RESERVED	–	Open

Note FLMD0 is at high level during on-chip debugging.

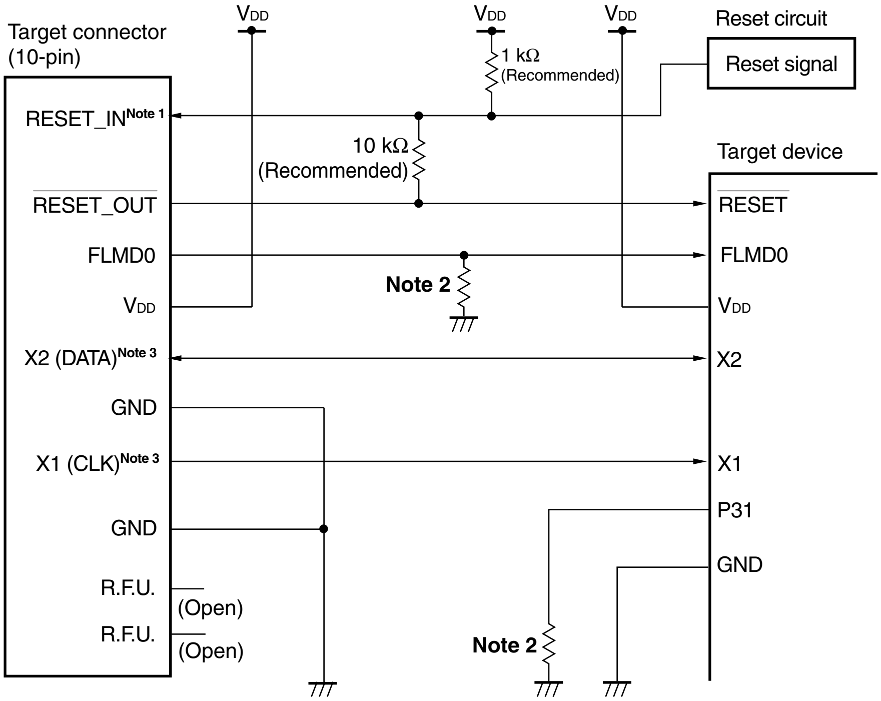
Figure 27-1. Connector Pin Layout



27.3 Connection Circuit Examples

The following are examples of circuits required when connecting the QB-78K0MINI or QB-MINI2 to the target system.

Figure 27-2. Connection Example of QB-78K0MINI or QB-MINI2 and 78K0/Dx2 (When X1 and X2 Are Used)



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-78K0MINI User's Manual (U17029E)** or **QB-MINI2 User's Manual (U18371E)**.

2. Make pull-down resistor 470 Ω or more (10 kΩ: recommended).

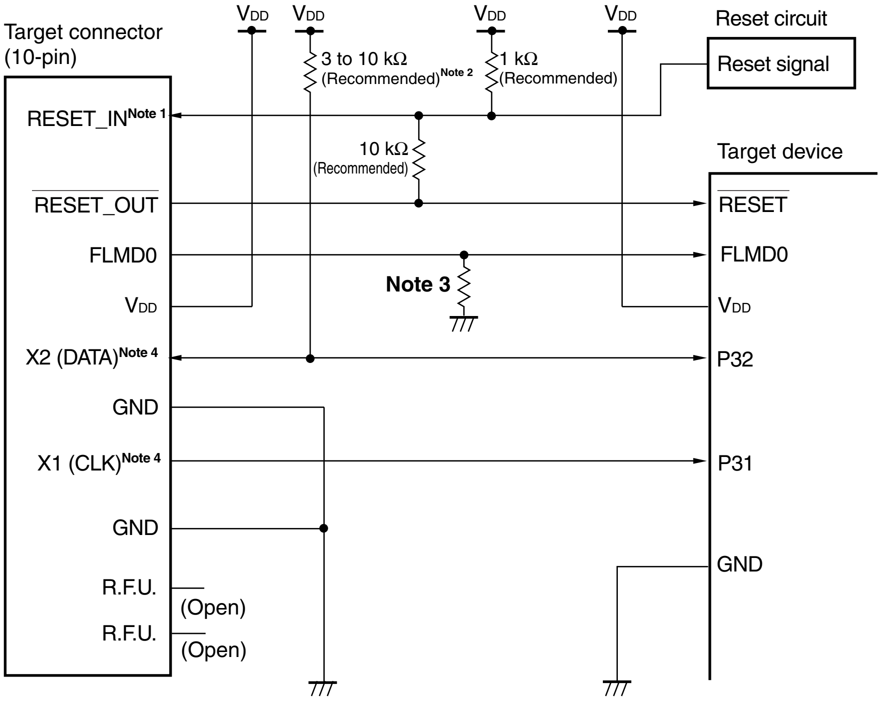
3. Characters without parentheses represent the QB-78K0MINI name, and those within parenthesis the QB-MINI2 name.

Cautions 1. Input the clock from the X1 pin during on-chip debugging.

2. Control the X1 and X2 pins by externally pulling down the P31 pin.

Figure 27-3. Connection Example of QB-78K0MINI or QB-MINI2 and 78K0/Dx2 (When P31 and P32 Are Used)

<R>



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-78K0MINI User’s Manual (U17029E)** or **QB-MINI2 User’s Manual (U18371E)**.

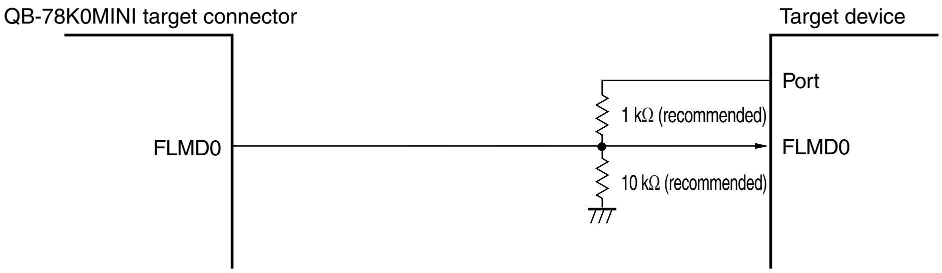
2. This is the processing of the pin when P32 is set as the input port (to prevent the pin from being left opened when not connected to QB-78K0MINI or QB-MINI2).

3. Make pull-down resistor 470 Ω or more (10 kΩ: recommended).

4. Characters without parentheses represent the QB-78K0MINI name, and those within parenthesis the QB-MINI2 name.

Connect the FLMD0 pin as follows when performing self programming by means of on-chip debugging.

Figure 27-4. Connection of FLMD0 Pin for Self Programming by Means of On-Chip Debugging



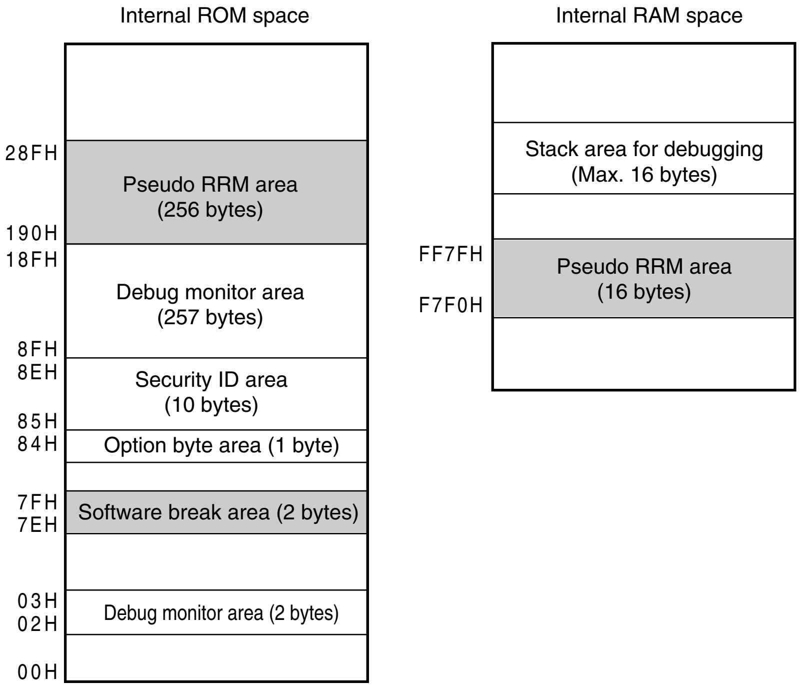
27.4 Reserved Area Used by QB-78K0MINI and QB-MINI2

QB-78K0MINI and QB-MINI2 use the reserved areas shown in **Figure 27-5** below to implement communication with the 78K0/Dx2, 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-78K0MINI User's Manual (U17029E)** or **QB-MINI2 User's Manual (U18371E)**.

Figure 27-5. Reserved Area Used by QB-78K0MINI and QB-MINI2



Remark Shaded reserved areas: Area used for the respective debug functions to be used
Other reserved areas: Areas always used for debugging

27.5 On-Chip Debug Security ID

The 78K0/Dx2 has an on-chip debug operation control flag in the flash memory at 0084H (see **CHAPTER 25 OPTION BYTE**) and an on-chip debug security ID setting area at 0085H to 008EH.

When the boot swap function is used, also set a value that is the same as that of 1084H and 1085H to 108EH in advance, because 0084H, 0085H to 008EH and 1084H, and 1085H to 108EH are switched.

For details on the on-chip debug security ID, refer to the **QB-78K0MINI User's Manual (U17029E)** or **QB-MINI2 User's Manual (U18731E)**.

Table 25-1. On-Chip Debug Security ID

Address	On-Chip Debug Security ID
0085H to 008EH	Any ID code of 10 bytes
1085H to 108EH	

27.6 Restrictions and Cautions on On-Chip Debug Function

When setting to on-chip debugging mode via the normal port, without using pins X1 and X2, two of the user ports will be unavailable for use.

In order to realize on-chip debug function, use the following user resource.

(a) Flash memory area

○Addresses 0x02 and 0x03
○Addresses 0x7E and 0x7F (when using a software break)
○Address 0x84
○Addresses 0x85 to 0x8E
○Addresses 0x8F to 0x18F: Standard value of program
(+256 bytes when using pseudo real-time RAM monitor function)
(when using a device with 10 or more SFRs the can be accessed in 16-bit units: +n (the number of exceeding registers x 6 bytes))

(b) Internal extended RAM area

○Addresses 0xF7F0 to 0xF7FF
(when using pseudo real-time RAM monitor function)

(c) Internal high-speed RAM area

○7 bytes as stack area: Standard value of stack
(+2 bytes when using software breaks)
(+7 bytes when using pseudo real-time RAM monitor function)

<R> In addition, set P31 and P32 as follows when these pins are used for on-chip debug. In this case, P31/SEG5/OCD1A and P32/SEG6/OCD1B pins are used only as OCD1A or OCD1B functions.

- P31/SEG5/OCD1A: Set to input mode (PM31 = 1) and port mode (PF31 = 0).
- P32/SEG6/OCD1B: Set to port mode (PF32 = 0).

When P31/SEG5 and P32/SEG6 functions need to be evaluated with on-chip debug, use OCD0A and OCD0B.

For details, refer to the **QB-78K0MINI User's Manual (U17029E)** or **QB-MINI2 User's Manual (U18731E)**.

CHAPTER 28 INSTRUCTION SET

This chapter lists each instruction set of 78K0/Dx2 in table form. For details of each operation and operation code, refer to the separate document **78K/0 Series Instructions User's Manual (U12326E)**.

28.1 Conventions Used in Operation List

28.1.1 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. Upper case 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 28-1. 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 3-7. Special Function Register List**.

28.1.2 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
∧:	Logical product (AND)
∨:	Logical sum (OR)
⊻:	Exclusive logical sum (exclusive OR)
‾:	Inverted data
addr16:	16-bit immediate data or label
jdisp8:	Signed 8-bit data (displacement value)

28.1.3 Description of flag operation column

(Blank):	Not affected
0:	Cleared to 0
1:	Set to 1
×:	Set/cleared according to the result
R:	Previously saved value is restored

28.2 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.

28.3 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

Caution The 78K0/Dx2 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.

29.1 Absolute Maximum Ratings

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}		–0.5 to +6.5	V
	EV_{DD}		–0.5 to +6.5	V
	V_{SS}		–0.5 to +0.3	V
	EV_{SS}		–0.5 to +0.3	V
	SMV_{DD}		–0.5 to +6.5	V
	SMV_{SS}		–0.5 to +0.3	V
	AV_{REF}		–0.5 to V_{DD} +0.3 [Note]	V
	AV_{SS}		–0.5 to +0.3	V
REGC pin Input voltage	V_{REGC}		–0.5 to +3.6 and ≤ V_{DD}	V
<R> Input voltage	V_{I1}	P00 to P07, P13 to P16, P30 to P37, P60, P61, P70 to P77, P120 to P124, (80-pin) P10 to P12, P17, X1, X2, XT1, XT2, $\overline{\text{RESET}}$, FLMD0	–0.3 to V_{DD} +0.3 [Note]	V
	V_{I2}	P20 to P27	–0.3 to AV_{REF} +0.3 [Note] and –0.3 to V_{DD} +0.3 [Note]	V
	V_{I3}	P80 to P87, P90 to P97, (64-pin) P10 to P12, P17	–0.3 to SMV_{DD} +0.3 [Note] and –0.3 to V_{DD} +0.3 [Note]	V
Output voltage	V_{O1}	P00 to P07, P13 to P16, P30 to P37, P60, P61, P70 to P77, P120 to P124, (80-pin) P10 to P12, P17	–0.3 to V_{DD} +0.3 [Note]	V
	V_{O2}	P20 to P27	–0.3 to AV_{REF} +0.3 [Note]	V
	V_{O3}	P80 to P87, P90 to P97, (64-pin) P10 to P12, P17	–0.3 to SMV_{DD} +0.3 [Note]	V
Analog input voltage	V_{AN}	ANI0 to ANI7	–0.3 to AV_{REF} +0.3 [Note] and –0.3 to V_{DD} +0.3 [Note]	V

Note Must be 6.5 V or lower.

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Absolute Maximum Ratings (T_A = 25°C) (2/2)

	Parameter	Symbol	Conditions		Ratings	Unit
	ZPD detection input voltage	V_{ZPD}	ZPD14, ZPD24, ZPD34, ZPD44		–0.3 to SMV_{DD} +0.3 [Note] and –0.3 to V_{DD} +0.3 [Note]	V
	Output current, high	I_{OH1}	Per pin	P00 to P07, P13 to P16, P30 to P37, P70 to P72 , P74 to P77, P120, (80-pin) P10 to P12, P17	–10	mA
			Total of all pins		–80	
		I_{OH2}	Per pin	P20 to P27	–0.5	mA
			Total of all pins		–2	
		I_{OH3}	Per pin	P121 to P124	–1	mA
			Total of all pins		–4	
<R>		I_{OH4}	Per pin	P80 to P87, P90 to P97, (64-pin) P10 to P12, P17	–45	mA
			Total of all pins	P80 to P87	–80	
			Total of all pins	P90 to P97	–80	
		I_{OH5}	Per pin	P73	–20	mA
	Output current, low	I_{OL1}	Per pin	P00 to P07, P13 to P16, P30 to P37, P70 to P72 , P74 to P77, P120, (80-pin) P10 to P12, P17	30	mA
			Total of all pins		200	
		I_{OL2}	Per pin	P20 to P27	1	mA
			Total of all pins		5	
		I_{OL3}	Per pin	P121 to P124	4	mA
			Total of all pins		10	
<R>		I_{OL4}	Per pin	P80 to P87, P90 to P97, (64-pin) P10 to P12, P17	45	mA
			Total of all pins	P80 to P87	80	
			Total of all pins	P90 to P97	80	
		I_{OL5}	Per pin	P73	20	mA
	Operating ambient temperature	T_A	In normal operation mode		–40 to +85	°C
			In flash memory programming mode		–40 to +85	
	Storage temperature	T_{stg}			–65 to +150	°C

Note Must be 6.5 V or lower.

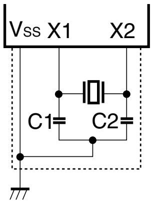
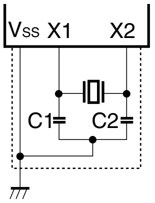
Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

29.2 Oscillator Characteristics

(1) Main System Clock (Crystal/Ceramic) Oscillator Characteristics

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ 5.5 V, 2.7 V ≤ SMV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = SMV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		X1 clock oscillation frequency (f_X) Note	4.0 V ≤ V_{DD} ≤ 5.5 V	4.0		20	MHz
			2.7 V ≤ V_{DD} < 4.0 V	4.0		10	
Crystal resonator		X1 clock oscillation frequency (f_X) Note	4.0 V ≤ V_{DD} ≤ 5.5 V	4.0		20	MHz
			2.7 V ≤ V_{DD} < 4.0 V	4.0		10	

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 8 MHz internal oscillator after reset, check the oscillation stabilization time of the main system clock using the oscillation stabilization time counter status register (OSTC). 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.

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.

(2) On-chip Internal Oscillator Characteristics

(T_A = −40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ 5.5 V, 2.7 V ≤ SMV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = SMV_{SS} = 0 V)

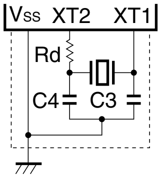
Resonator	Parameter	Conditions		MIN.	TYP.	MAX.	Unit
8 MHz internal oscillator	Internal high-speed oscillation clock frequency (f_{OSC8}) Note	RSTS = 1	2.7 V ≤ V_{DD} ≤ 5.5 V	7.6	8	8.4	MHz
		RSTS = 0		2.48	5	9.86	
240 kHz internal oscillator	Internal low-speed oscillation clock frequency (f_{OSC})	2.7 V ≤ V_{DD} ≤ 5.5 V		216	240	264	kHz

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Remark RSTS: Bit 7 of the internal oscillation mode register (RCM)

(3) Subsystem Clock Oscillator Characteristics

(T_A = −40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ 5.5 V, 2.7 V ≤ SMV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = SMV_{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 figures to avoid an adverse effect from wiring capacitance.

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

2. The subsystem clock oscillator is designed as a low-amplitude circuit for reducing power consumption, and is more prone to malfunction due to noise than the high-speed system clock oscillator. Particular care is therefore required with the wiring method when the subsystem clock is used.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

29.3 DC Characteristics

DC Characteristics (1/7)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ 5.5 V, 2.7 V ≤ SMV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = SMV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high [Note 1]	I_{OH1}	Per pin for P00 to P07, P13 to P16, P30 to P37, P60, 61, P70 to P72, P74 to P77, P120, (80-pin) P10 to P12, P17	4.0 V ≤ V_{DD} ≤ 5.5 V			–3.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–2.5	
		Total of pins [Note 2]	4.0 V ≤ V_{DD} ≤ 5.5 V			–23	
			2.7 V ≤ V_{DD} < 4.0 V			–18	
	I_{OH2}	Per pin for P20 to P27	AV_{REF} = V_{DD}			–100	μA
		Per pin for P121 to P124					
<R>	I_{OH3}	Per pin for P80 to P87, P90 to P97, (64-pin) P10 to P12, P17	4.5 V ≤ SMV_{DD} ≤ 5.5 V			–27	mA
			2.7 V ≤ SMV_{DD} < 4.5 V			–19	
		Total of pins [Note 2] P80 to P87	4.5 V ≤ SMV_{DD} ≤ 5.5 V			–70	
			2.7 V ≤ SMV_{DD} < 4.5 V			–50	
		Total of pins [Note 2] P90 to P97	4.5 V ≤ SMV_{DD} ≤ 5.5 V			–70	
			2.7 V ≤ SMV_{DD} < 4.5 V			–50	
	I_{OH4}	P73	4.5 V ≤ V_{DD} ≤ 5.5 V			–15.0	mA
			2.7 V ≤ V_{DD} < 4.5 V			–9	

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. 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.

Caution The pins mounted depend on the product.

DC Characteristics (2/7)

(T_A = −40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ 5.5 V, 2.7 V ≤ SMV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = SMV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low[Note 1]	I_{OL1}	Per pin for P00 to P07, P13 to P16, P30 to P37, P60, 61, P70 to P72, P74 to P77, P120, (80-pin) P10 to P12, P17	4.0 V ≤ V_{DD} ≤ 5.5 V			8.5	mA
			2.7 V ≤ V_{DD} < 4.0 V			5.0	
		Total of pins[Note 2]	4.0 V ≤ V_{DD} ≤ 5.5 V			65	
			2.7 V ≤ V_{DD} < 4.0 V			50	
	I_{OL2}	Per pin for P20 to P27	AV_{REF} = V_{DD}			400	μA
		Per pin for P121 to P124					
<R>	I_{OL3}	Per pin for P80 to P87, P90 to P97, (64-pin) P10 to P12, P17	4.5 V ≤ SMV_{DD} ≤ 5.5 V			27	mA
			2.7 V ≤ SMV_{DD} < 4.5 V			15	
		Total of pins[Note 2] P80 to P87	4.5 V ≤ SMV_{DD} ≤ 5.5 V			70	
			2.7 V ≤ SMV_{DD} < 4.5 V			50	
		Total of pins[Note 2] P90 to P97	4.5 V ≤ SMV_{DD} ≤ 5.5 V			70	
			2.7 V ≤ SMV_{DD} < 4.5 V			50	
	I_{OL4}	P73	4.5 V ≤ V_{DD} ≤ 5.5 V			15.0	mA
			2.7 V ≤ V_{DD} < 4.5 V			8.5	

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to GND.

2. 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.

Caution The pins mounted depend on the product.

DC Characteristics (3/7)

(T_A = −40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ 5.5 V, 2.7 V ≤ SMV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = SMV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P71 to P73, P76, P121 to P124, P30, P36			$0.7V_{DD}$		V_{DD}	V
	V_{IH2}	P00 to P07, P13 to P16, P31 to P35, P37, P60, P61, P70, P74, P75, P77, P120, $\overline{\text{RESET}}$, EXCLK, EXCLKS, (80-pin) P10 to P12, P17			$0.8V_{DD}$		V_{DD}	V
	V_{IH3}	P20 to P27		$AV_{REF} = V_{DD}$	$0.7AV_{REF}$		AV_{REF}	V
	V_{IH4}	P80 to P87, P90 to P97		$SMV_{DD} = V_{DD}$	$0.7SMV_{DD}$		SMV_{DD}	V
	V_{IH5}	(64-pin) P10 to P12, P17		$SMV_{DD} = V_{DD}$	$0.8SMV_{DD}$		SMV_{DD}	V
Input voltage, low	V_{IL1}	P71 to P73, P76, P121 to P124, P30, P36			0		$0.3V_{DD}$	V
	V_{IL2}	P00 to P07, P13 to P16, P31 to P35, P37, P60, P61, P70, P74, P75, P77, P120, $\overline{\text{RESET}}$, EXCLK, EXCLKS, (80-pin) P10 to P12, P17			0		$0.2V_{DD}$	V
	V_{IL3}	P20 to P27		$AV_{REF} = V_{DD}$	0		$0.3AV_{REF}$	V
	V_{IL4}	P80 to P87, P90 to P97		$SMV_{DD} = V_{DD}$	0		$0.3SMV_{DD}$	V
	V_{IL5}	(64-pin) P10 to P12, P17		$SMV_{DD} = V_{DD}$	0		$0.2SMV_{DD}$	V
Output voltage, high	V_{OH1}	I_{OH} = −2.5 mA	P00 to P07, P13 to P16, P30 to P37, P60, P61, P70 to P72, P74 to P77, P120, (80-pin) P10 to P12, P17	4.0 V ≤ V_{DD} ≤ 5.5 V	$V_{DD} - 0.7$			V
		I_{OH} = −2.0 mA		2.7 V ≤ V_{DD} < 5.5 V	$V_{DD} - 0.5$			
	V_{OH2}	I_{OH} = −100 μA	P20 to P27	$AV_{REF} = V_{DD}$	$V_{DD} - 0.5$			V
			P121 to P124					
	V_{OH3}	I_{OH} = −27.0 mA (T_A = 85°C)	P80 to P87, P90 to P97, (64-pin) P10 to P12, P17	4.5 V ≤ SMV_{DD} ≤ 5.5 V	$SMV_{DD} - 0.5$			V
		I_{OH} = −30.0 mA (T_A = 25°C)						
		I_{OH} = −40.0 mA (T_A = 40°C)						
<R>		I_{OH} = −19.0 mA (T_A = 85°C)		2.7 V ≤ SMV_{DD} ≤ 5.5 V	$SMV_{DD} - 0.5$			
		I_{OH} = −23.0 mA (T_A = 25°C)						
		I_{OH} = −26.0 mA (T_A = 40°C)						
	V_{OH4}	I_{OH} = −15.0 mA	P73	4.5 V ≤ V_{DD} ≤ 5.5 V	$V_{DD} - 0.7$			V
		I_{OH} = −9.0 mA		2.7 V ≤ SMV_{DD} ≤ 5.5 V	$V_{DD} - 0.7$			

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Caution The pins mounted depend on the product.

DC Characteristics (4/7)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ 5.5 V, 2.7 V ≤ SMV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = SMV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Output voltage, low	V_{OL1}	I_{OL} = 5.0 mA	P00 to P07, P13 to P16, P30 to P37, P60, P61, P70 to P72, P74 to P77, P120, (80-pin) P10 to P12, P17	4.0 V ≤ V_{DD} ≤ 5.5 V			0.7	V
		I_{OL} = 3.0 mA		2.7 V ≤ V_{DD} < 5.5 V			0.7	V
	V_{OL2}	I_{OL} = 400 μA	P20 to P27	AV_{REF} = V_{DD}			0.4	V
			P121 to P124					
	V_{OL3}	I_{OL} = 27.0 mA (T_A = 85°C)	P80 to P87, P90 to P97, (64-pin) P10 to P12, P17	4.5 V ≤ SMV_{DD} ≤ 5.5 V			0.5	V
		I_{OL} = 30.0 mA (T_A = 25°C)						
		I_{OL} = 40.0 mA (T_A = 40°C)						
		I_{OL} = 15.0 mA (T_A = 85°C)		2.7 V ≤ SMV_{DD} ≤ 5.5 V			0.5	V
<R>		I_{OL} = 19.0 mA (T_A = 25°C)						
<R>		I_{OL} = 23.0 mA (T_A = 40°C)						
	V_{OL4}	I_{OL} = 15.0 mA	P73	4.5 V ≤ V_{DD} ≤ 5.5 V			0.7	V
		I_{OL} = 8.5 mA		2.7 V ≤ SMV_{DD} ≤ 5.5 V			0.7	

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Caution The pins mounted depend on the product.

DC Characteristics (5/7)

(T_A = −40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ 5.5 V, 2.7 V ≤ SMV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = SMV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	$V_I = V_{DD}$	P71 to P73, P60, P61, P120, $\overline{\text{RESET}}$, (80-pin) P10 to P12, P17, FLMD0				1	μA
	I_{LIH2}	$V_I = V_{DD}$	P74 to P77, P13 to P16, P00 to P07, P30 to P37				10	μA
	I_{LIH3}	$V_I = AV_{REF}$	P20 to P27	$AV_{REF} = V_{DD}$			1	μA
	I_{LIH4}	$V_I = V_{DD}$	P121 to P124 (X1, X2, XT1, XT2)	I/O port mode			1	μA
				OSC mode			20	
	I_{LIH5}	$V_I = SMV_{DD}$	P80 to P87, P90 to P97, (64-pin) P10 to P12, P17				10	μA
Input leakage current, low	I_{LIL1}	$V_I = V_{SS}$	P71 to P73, P60, P61, P120, $\overline{\text{RESET}}$, (80-pin) P10 to P12, P17, FLMD0				−1	μA
	I_{LIL2}		P74 to P77, P13 to P16, P00 to P07, P30 to P37				−10	μA
	I_{LIL3}		P20 to P27	$AV_{REF} = V_{DD}$			−1	μA
	I_{LIL4}		P121 to P124 (X1, X2, XT1, XT2)	I/O port mode			−1	μA
				OSC mode			−20	
	I_{LIL5}		P80 to P87, P90 to P97, (64-pin) P10 to P12, P17				−10	μA
Pull-up resistor	R_{U1}	$V_I = V_{SS}$	P00 to P07, P13 to P16, P30 to P37, P60, P61, P70 to P77, P120, (80-pin) P10 to P12, P17		10	20	100	kΩ
	R_{U2}	$V_I = SMV_{SS}$	(64-pin) P10 to P12, P17		10	20	100	kΩ
Pull-down resistor	R_{D1}	$V_I = SMV_{DD}$	P80 to P87, P90 to P97, (64-pin) P10 to P12, P17		100			kΩ
	R_{D2}	$V_I = V_{DD}$	P74 to P77, P13 to P16, P00 to P07, P30 to P37, SEG0 to SEG3		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.

Caution The pins mounted depend on the product.

DC Characteristics (6/7)

(T_A = −40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ 5.5 V, 2.7 V ≤ SMV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = SMV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}	Operating mode	f_{IN} = 20 MHz[Note 2], V_{DD} = 5.0 V	Square wave input		3.8	7.6	mA
				Resonator connection		5.1	8.9	
			f_{IN} = 10 MHz[Notes 2, 3], V_{DD} = 5.0 V	Square wave input		2.2	4.4	
				Resonator connection		4.9	5.5	
			f_{IN} = 10 MHz[Notes 2, 3], V_{DD} = 3.0 V	Square wave input		2.1	4.3	
				Resonator connection		2.8	4.7	
			f_{IN} = 5 MHz[Notes 2, 3], V_{DD} = 3.0 V	Square wave input		1.2	2.4	
				Resonator connection		1.6	2.7	
			f_{OSC8} = 8 MHz [Note 4], V_{DD} = 5.0 V			1.7	3.1	
			f_{SUB} = 32.768 kHz[Note 5], V_{DD} = 5.0 V	Square wave input		6.1	98	μA
				Resonator connection		15.4	102	
	I_{DD2}	HALT mode	f_{IN} = 20 MHz[Note 2], V_{DD} = 5.0 V	Square wave input		1.2	4.7	mA
				Resonator connection		2.4	6.2	
			f_{IN} = 10 MHz[Notes 2, 3], V_{DD} = 5.0 V	Square wave input		0.8	2.7	
				Resonator connection		1.4	3.6	
			f_{IN} = 5 MHz[Notes 2, 3], V_{DD} = 3.0 V	Square wave input		0.4	1.3	
				Resonator connection		0.7	1.8	
			f_{OSC8} = 8 MHz [Note 4], V_{DD} = 5.0 V			0.6	1.7	
			f_{SUB} = 32.768 kHz[Note 5], V_{DD} = 5.0 V	Square wave input		3.1	92	μA
				Resonator connection		12.4	97	

Notes 1. Total current flowing into the internal power supply (V_{DD}, EV_{DD}, SMV_{DD}), including the peripheral operation current and the input leakage current flowing when the level of the input pin are fixed to V_{DD} or V_{SS}. However, the current flowing into the pull-up resistors and the output current of the port is not included.

2. Not including the operating current of the 8 MHz internal oscillator, XT1 oscillation, 240 kHz internal oscillator and the current flowing into the A/D converter, watchdog timer, LVI circuit, LCD controller/driver and ZPD.

3. When AMPH (bit 0 of clock operation mode select register (OSCCTL)) = 0.

4. Not including the operating current of the X1 oscillation, XT1 oscillation and 240 kHz internal oscillator. Not including the current flowing into the A/D converter, watchdog timer, LVI circuit, LCD controller/driver and ZPD.

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 ZPD.

Remarks 1. f_{IN}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{OSC8}: Internal high-speed oscillation clock frequency

3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency or external subsystem clock frequency)

DC Characteristics (7/7)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ 5.5 V, 2.7 V ≤ SMV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = SMV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current [Note 1]	I_{DD3} [Note 2]	STOP mode	V_{DD} = 5.0 V (CREG+POC)	T_A = –40 to +70°C		1	10	μA
				T_A = –40 to +85°C		1	20	
Watchdog timer operating current	I_{WDT} [Note 3]	During 240 kHz internal low-speed oscillation clock operation				5	10	μA
LVI operating current	I_{LVI} [Note 4]					9	18	μA
A/D converter operating current	I_{ADC} [Note 5]	ADCE = 1	2.3 V ≤ AV_{REF} ≤ V_{DD}			0.86	1.9	mA
LCD controller/driver operating current	I_{LCD1} [Note 6]	f_{PRS} = 20 MHz LCD source clock (f_{LCD}) = $f_{PRS}/2^7$ LCD clock = $f_{LCD}/2^4$ Four-time-slice display	LCD non-display waveform output	V_{DD} = 5.0 V		3.0	8.0	μA
				V_{DD} = 3.0 V		2.0	5.0	
	I_{LCD2} [Note 6]		LCD display waveform output	V_{DD} = 5.0 V		3.0	8.0	μA
				V_{DD} = 3.0 V		2.0	5.0	
<R> ZPD operating current	I_{ZPD}	One ZPD circuit operating		V_{DD} = 5.0 V		0.15	0.6	mA
				V_{DD} = 3.0 V		0.1	0.5	
		Four ZPD circuits operating		V_{DD} = 5.0 V		0.5	2.0	
				V_{DD} = 3.0 V		0.4	1.6	

Notes 1. Total current flowing into the internal power supply (V_{DD}, EV_{DD}, SMV_{DD}), including the peripheral operation current and the input leakage current flowing when the level of the input pin are fixed to V_{DD} or V_{SS}. However, the current flowing into the pull-up resistors and the output current of the port is not included.

2. Not including 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 ZPD.

3. Current flowing only to the watchdog timer (V_{DD}-pin) (including the operating current of the 240 kHz internal oscillator). The current value of the 78K0/Dx2 is the sum of I_{DD2} or I_{DD3} and IWDT when the watchdog timer operates in the HALT or STOP mode.

4. Current flowing only to the LVI circuit (V_{DD}-pin). The current value of the 78K0/Dx2 is the sum of I_{DD2} or I_{DD3} and ILVI when the LVI circuit operates in the HALT or STOP mode.

5. Current flowing only to the A/D converter (AV_{REF}-pin). The current value of the 78K0/Dx2 is the sum of I_{DD1} or I_{DD2} and I_{ADC} when the A/D converter operates in an operation mode or the HALT mode.

6. Not including the operating current flowing in the internal division resistor.

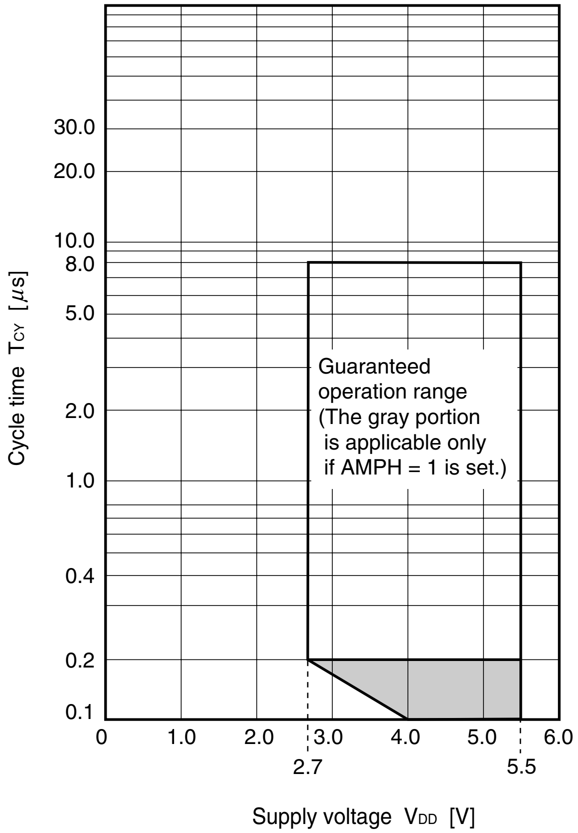
29.4 AC Characteristics

(1) Basic operation

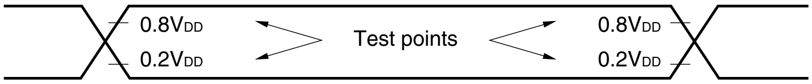
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = SMV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	4.0 V ≤ V_{DD} ≤ 5.5 V	0.1		8	μs
			2.7 V ≤ V_{DD} < 4.0 V	0.2		8	
		Subsystem clock (f_{SUB}) operation		114	122	125	
Peripheral hardware clock frequency	f_{PRS}	XSEL = 1	4.0 V ≤ V_{DD} ≤ 5.5 V			20	MHz
			2.7 V ≤ V_{DD} < 4.0 V			10	
		XSEL = 0	2.7 V ≤ V_{DD} < 4.0 V	7.6		8.4	
External main system clock frequency	f_{EXT}	4.0 V ≤ V_{DD} ≤ 5.5 V		4.0		20	MHz
		2.7 V ≤ V_{DD} < 4.0 V		4.0		10	
External clock input high level width, low level width	f_{EXTH}, f_{EXTL}			(1/f_{EXT}×1/2)–1			ns
External subsystem clock frequency	f_{EXTS}			32	32.768	35	kHz
External sub clock input high level width, low level width	f_{EXTSH}, f_{EXTSL}			(1/f_{EXTS}×1/2)–5			ns
TIOP00, 01, 10, 11, 20, 21, 30, 31, 40, 41, TIO50, 51 input frequency	f_{TI5}	4.0 V ≤ V_{DD} ≤ 5.5 V				10	MHz
		2.7 V ≤ V_{DD} < 4.0 V				10	
TIOP00, 01, 10, 11, 20, 21, 30, 31, 40, 41, TIO50, 51 input high-level width, low-level width	t_{TIH5}, t_{TIL5}	4.0 V ≤ V_{DD} ≤ 5.5 V		50			ns
		2.7 V ≤ V_{DD} < 4.0 V		50			
Interrupt input high-level width, low-level width	t_{INIH}, t_{INIL}			1			μs
$\overline{\text{RESET}}$ low-level width	t_{RSL}			10			μs

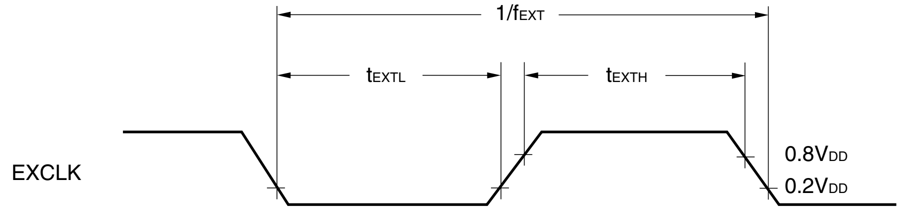
T_{CY} vs. V_{DD} (Main System Clock Operation)

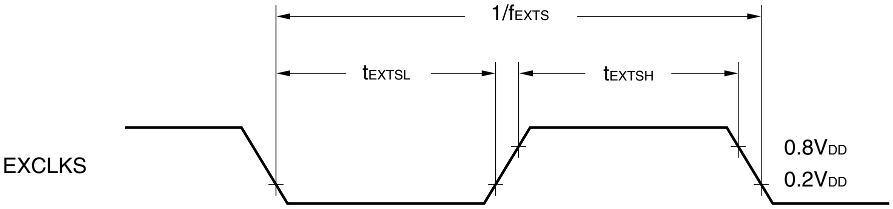


AC Timing Test Points (Excluding X1, XT1)

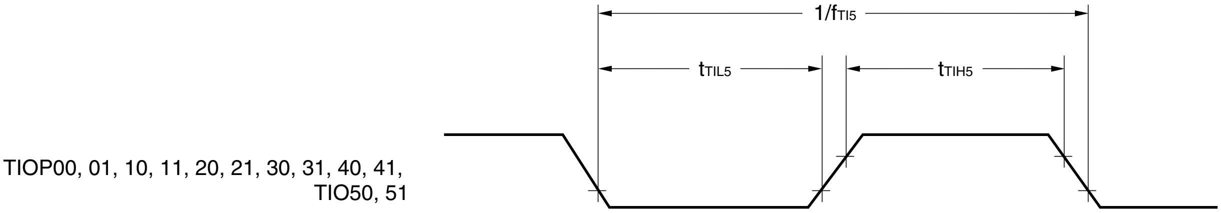


External clock input timing

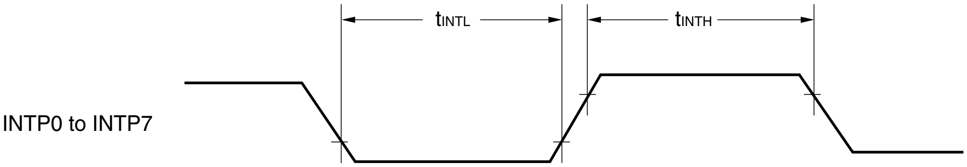




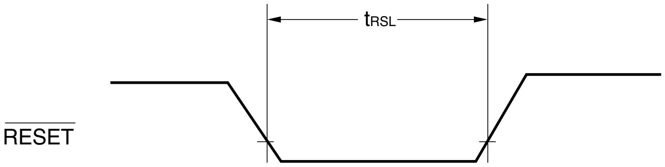
TI Timing



Interrupt Request Input Timing



$\overline{\text{RESET}}$ Input Timing



(2) Serial interface

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ 5.5 V, 2.7 V ≤ SMV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = SMV_{SS} = 0 V)

(a) UART6n (Dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					625	kbps

(b) IIC0

Parameter	Symbol	Conditions		Standard Mode		High-Speed Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCL0 clock frequency	f_{CLK}			0	100	0	400	kHz
Setup time of start/restart condition	$t_{SU:STA}$			4.7		0.6		µs
Hold time[Note 1]	$t_{HD:STA}$			4.0		0.6		µs
Hold time when SCL0 = “L”	t_{LOW}			4.7		1.3		µs
Hold time when SCL0 = “H”	t_{HIGH}			4.0		0.6		µs
Data setup time (reception)	$t_{SU:DAT}$			250		100		ns
Data hold time (transmission)[Note 2]	$t_{HD:DAT}$	$f_W = f_{IN}/2^N$ selected[Note 3]	DFC0 = 0	0	3.45	0	0.9[Note 4]	µs
							1.0[Note 5]	
			DFC0 = 1			0	0.9[Note 6]	
							1.125[Note 7]	
		$f_W = f_{OSC8}/2^N$ selected[Note 3]	DFC0 = 0	0	3.45	0	1.05	
			DFC0 = 1			0	1.184	
Setup time of stop condition	$t_{SU:STO}$			4.0		0.6		µs
Bus free time between stop and start conditions	t_{BUF}			4.7		1.3		µs

Notes

1. The first clock pulse is generated after this period when the start/restart condition is detected.
2. The maximum value (MAX.) of $t_{HD:DAT}$ is during normal transfer and a wait state is inserted in the ACK (acknowledge) timing.
3. fw indicates the IIC0 transfer clock selected by the IICCL and IICX0 registers.
4. When fw ≥ 4 MHz is selected
5. When fw < 4 MHz is selected
6. When fw ≥ 5 MHz is selected
7. When fw < 5 MHz is selected

Remark n = 0, 1

(c) CSI1n (Master mode, $\overline{\text{SCK1n}}$... internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK1n}}$ cycle time	t_{KCY1}	4.0 V ≤ V_{DD} ≤ 5.5 V	200			ns
		2.7 V ≤ V_{DD} < 4.0 V	400			ns
$\overline{\text{SCK1n}}$ high-/low-level width Note 1	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V	$t_{KCY1}/2 - 20$			ns
		2.7 V ≤ V_{DD} < 4.0 V	$t_{KCY1}/2 - 30$			ns
SI1n setup time (to $\overline{\text{SCK1n}}$↑)	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V	70			ns
		2.7 V ≤ V_{DD} < 4.0 V	100			ns
SI1n hold time (from $\overline{\text{SCK1n}}$↑)	t_{KSI1}		30			ns
Delay time from $\overline{\text{SCK1n}}$↓ to SO1n output	t_{KSO1}	C = 50 pF Note 2			40	ns

Notes 1. It is value at the time of f_X use. Keep in mind that spec different at the time of f_{OSC8} use.

2. C is the load capacitance of the $\overline{\text{SCK1n}}$ and SO1n output lines.

(d) CSI1n (Slave mode, $\overline{\text{SCK1n}}$... external clock input)

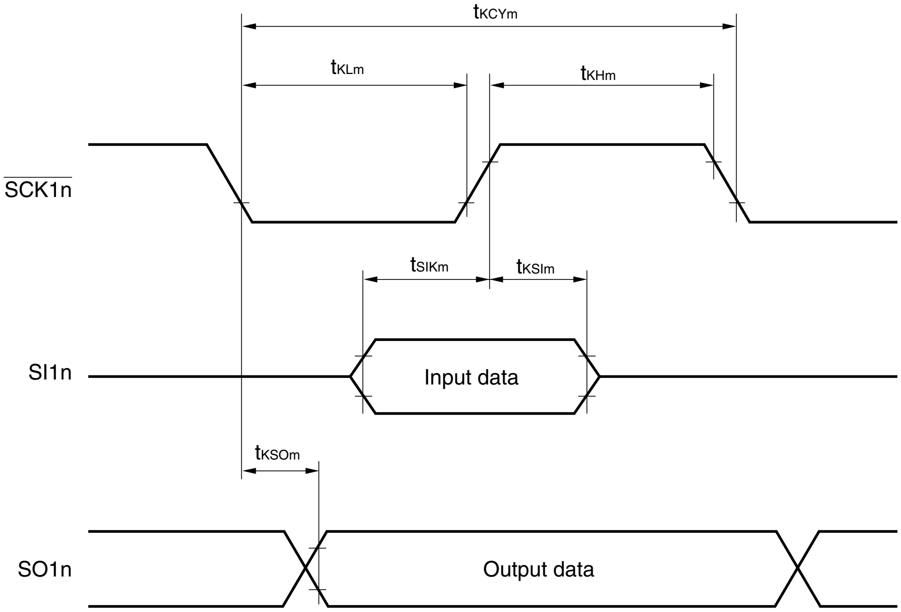
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK1n}}$ cycle time	t_{KCY2}		400			ns
$\overline{\text{SCK1n}}$ high-/low-level width	t_{KH2}, t_{KL2}		$t_{KCY2}/2$			ns
SI1n setup time (to $\overline{\text{SCK1n}}$↑)	t_{SIK2}		80			ns
SI1n hold time (from $\overline{\text{SCK1n}}$↑)	t_{KSI2}		50			ns
Delay time from $\overline{\text{SCK1n}}$↓ to SO1n output	t_{KSO2}	C = 50 pF Note			120	ns

Note C is the load capacitance of the SO1n output line.

Remark n = 0, 1

Serial Transfer Timing

CSI1n:

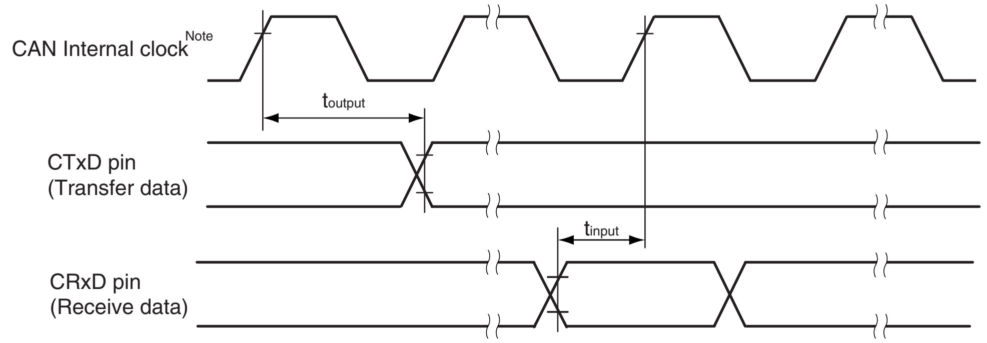


Remark m = 1, 2
n = 0, 1

(3) CAN controller

(T_A = −40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ 5.5 V, 2.7 V ≤ SMV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = SMV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					1	Mbps
Internal delay time	t_{NODE}				100	ns



Internal delay time (t_{NODE}) = Internal Transfer Delay (t_{output}) + Internal Receive Delay (t_{input})

Note CAN Internal clock (f_{CAN}): CAN baud rate clock

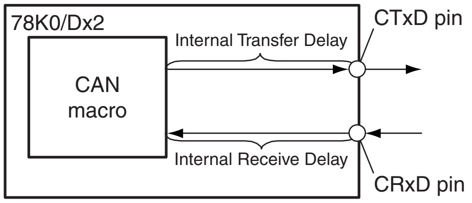


Image figure of internal delay

(4) A/D Converter Characteristics

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ 5.5 V, 2.7 V ≤ SMV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = SMV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	RES				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} < 5.5 V			±0.6	
Conversion time	t_{CONV}	4.0 V ≤ AV_{REF} ≤ 5.5 V	6.1		36.7	μs
		2.7 V ≤ AV_{REF} < 5.5 V	6.1		36.7	
Zero-scale error[Notes 1, 2]	E_{ZS}	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{REF} < 5.5 V			±0.6	
Full-scale error[Notes 1, 2]	E_{FS}	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{REF} < 5.5 V			±0.6	
Integral non-linearity error[Note 1]	I_{LE}	4.0 V ≤ AV_{REF} ≤ 5.5 V			±2.5	LSB
		2.7 V ≤ AV_{REF} < 5.5 V			±4.5	
Differential non-linearity error[Note 1]	D_{LE}	4.0 V ≤ AV_{REF} ≤ 5.5 V			±1.5	LSB
		2.7 V ≤ AV_{REF} < 5.5 V			±2.0	
Analog input voltage	V_{AIN}		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.

(5) Meter controller/driver and ZPD characteristics

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ 5.5 V, 2.7 V ≤ SMV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = SMV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Meter controller/driver input frequency	f_{MC} Note 1	V_{DD} = 4.5 V to 5.5 V				20	MHz
						10	
PWM output rise time	t_R	C = 50 pF Note 2 (10% - 90%)	4.5 V ≤ SMV_{DD} ≤ 5.5 V	12	25	70	ns
			2.7 V ≤ SMV_{DD} ≤ 5.5 V	12	25	200	
PWM output fall time	t_F	C = 50 pF Note 2 (10% - 90%)	4.5 V ≤ SMV_{DD} ≤ 5.5 V	12	25	70	ns
			2.7 V ≤ SMV_{DD} ≤ 5.5 V	12	25	200	
<R> Peak cross current Note 3	I_{CROSS}					50	mA
Output pulse width Note 4	t_{MO}			125			ns
Output pulse length deviation Note 5	t_{SMDEV}			–40		+10	ns
Symmetry performance Note 6	ΔHSPmn	I_{OH} = –27 mA ΔHSPmn = |V_{OH} [(SMmn)max–(SMmn)min]|				50	mV
	ΔHSPmn	I_{OL} = 27 mA ΔHSPmn = |V_{OL} [(SMmn)max–(SMmn)min]|				50	mV

Notes 1. Source clock of the free-running counter.

2. C is the load capacitance of the PWM output line.

3. The slew rate control generates a cross current in the output stage to control the energy of the external inductive load. The cross current flows only during the output transition time t_R, t_F. It flows in addition to the output current. The cross current is not tested, but derived from simulation.

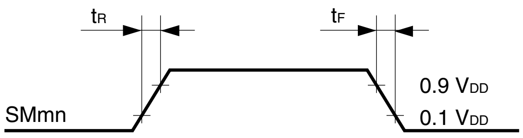
4. The output buffer can not generate high or low pulses shorter than this time, because of its slew rate control system. This value is not tested, but derived from simulation.

5. The slew rate control function causes a deviation of output pulse time compared to the ideal selected output pulse setting. This value is not tested, but derived from simulation.

6. Indicates the dispersion of up to 16 PWM output voltages. Not tested in production, specified by design.

Remark m = 1 to 4, n = 1 to 4

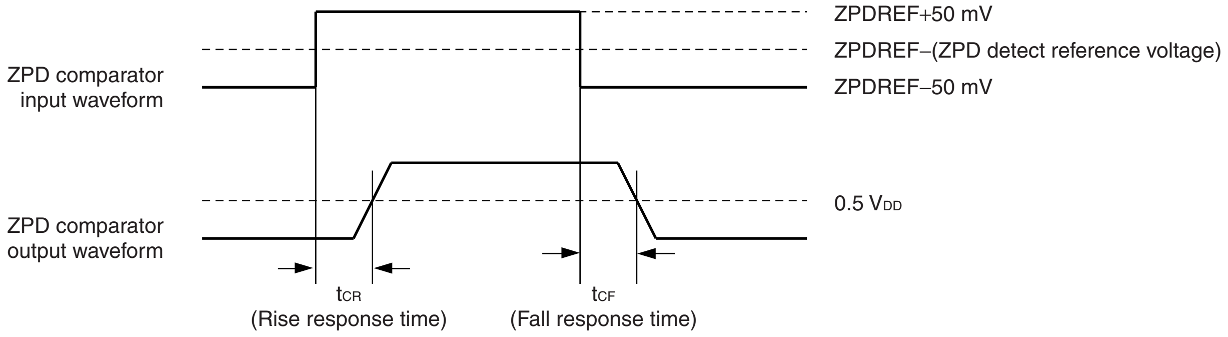
Meter Controller/Driver Output Timing



Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Threshold voltage	V_{ZPD}	0-point detection voltage setting = 000		3/100×SMV_{DD}±70 mV			V
		0-point detection voltage setting = 001		5/100×SMV_{DD}±70 mV			
		0-point detection voltage setting = 010		7/100×SMV_{DD}±70 mV			
		0-point detection voltage setting = 011		9/100×SMV_{DD}±70 mV			
		0-point detection voltage setting = 100		11/100×SMV_{DD}±70 mV			
Detection delay	t_{ZPDD}	100 mV step, 50 mV overdrive (see the figure below)	SMV_{DD} = 4.75 V to 5.25 V			100 Note	ns
			SMV_{DD} = 2.7 V to 5.5 V			100 Note	
Operation stabilization wait time	t_{ZPDW}	Reference voltage stabilization + ZPD comparator stabilization				1+5 = 6	μs

Note Not tested in production, specified by design.

ZPD Timing

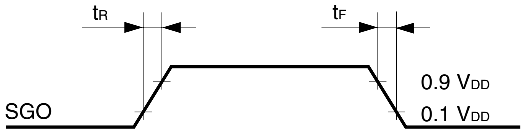

ZPD comparator input waveform
ZPD comparator output waveform
ZPDREF+50 mV
ZPDREF–(ZPD detect reference voltage)
ZPDREF–50 mV
0.5 V_{DD}
t_{CR}
(Rise response time)
t_{CF}
(Fall response time)

(6) Sound generator characteristics

(T_A = −40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ 5.5 V, 2.7 V ≤ SMV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = SMV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
<R> Sound generator input frequency	f_{SG0}	V_{DD} = 4.5 V to 5.5 V				10	MHz
						5	MHz
<R> SGO output rise time	t_R	C = 100 pF	4.5 V ≤ SMV_{DD} ≤ 5.5 V			200	ns
			2.7 V ≤ SMV_{DD} ≤ 5.5 V			200	ns
SGO output fall time	t_F	C = 100 pF	4.5 V ≤ SMV_{DD} ≤ 5.5 V			200	ns
			2.7 V ≤ SMV_{DD} ≤ 5.5 V			200	ns

Sound Generator Output Timing



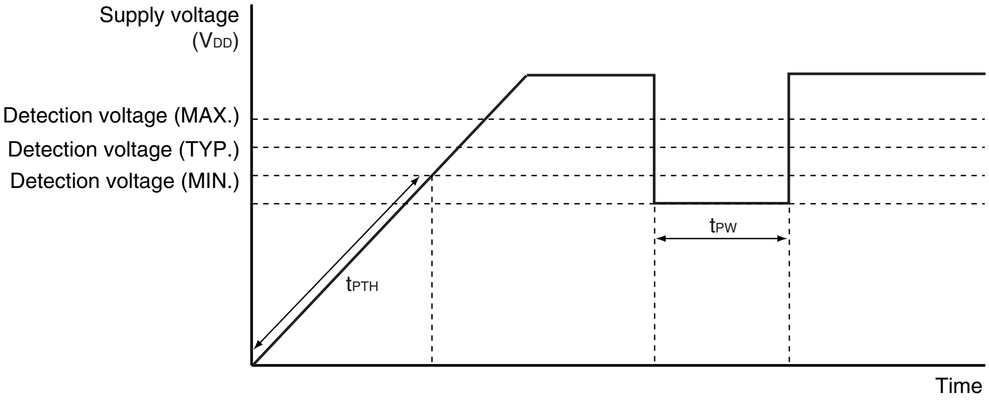
(7) POC circuit characteristics

(T_A = –40 to +85°C, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POC0}		1.44	1.59	1.74	V
Power supply rise time	t_{PTH}	V_{DD}: 0 V → V_{POC0}			0.5	V/ms
Minimum pulse width [Note]	t_{PW}	When the voltage drops	200			μs

Note When the voltage drops in shorter width than the minimum pulse width, the operation of POC detection isn't guaranteed.

POC Circuit Timing



(8) LVI circuit characteristics

(T_A = −40 to +85°C, $V_{POC} \leq V_{DD} = EV_{DD} \leq 5.5$ V, $V_{SS} = EV_{SS} = 0$ V)

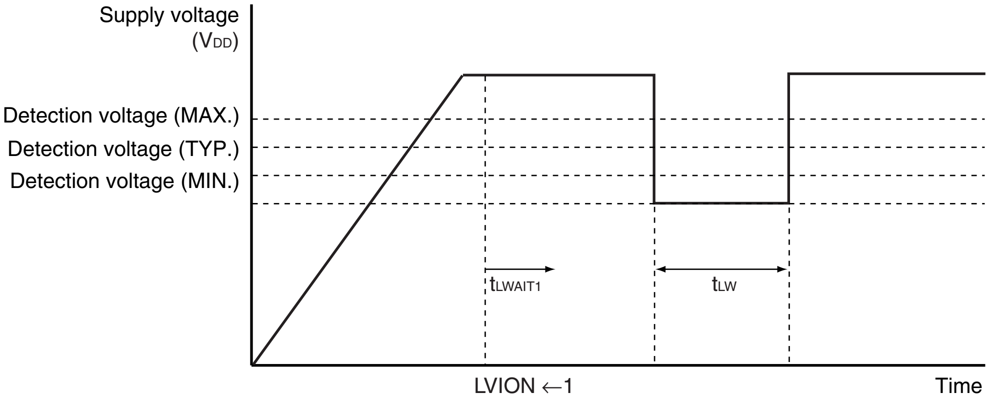
Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVI0}		4.14	4.24	4.34	V
		V_{LVI1}		3.99	4.09	4.19	V
		V_{LVI2}		3.83	3.93	4.03	V
		V_{LVI3}		3.68	3.78	3.88	V
		V_{LVI4}		3.52	3.62	3.72	V
		V_{LVI5}		3.37	3.47	3.57	V
		V_{LVI6}		3.22	3.32	3.42	V
		V_{LVI7}		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
	External input pin[Note 1]	EX_{LVI}	$EX_{LVI} < V_{DD}$, $2.7\ V \leq V_{DD} \leq 5.5\ V$	1.11	1.21	1.31	V
	Detection voltage on application of supply voltage	VDD_{LVI}	LVISTART (option byte) = 1	2.50	2.70	2.90	V
Minimum pulse width		t_{LW}		200			μs
Operation stabilization wait time[Note 2]		t_{LWAIT1}				10	μs

Notes 1. External input pin is alternate P120 pin.

2. Time required from setting LVION to 1 to operation stabilization.

Remark $V_{LVI\,(n-1)} > V_{LVIn}$ (n = 1 to 10)

LVI Circuit Timing

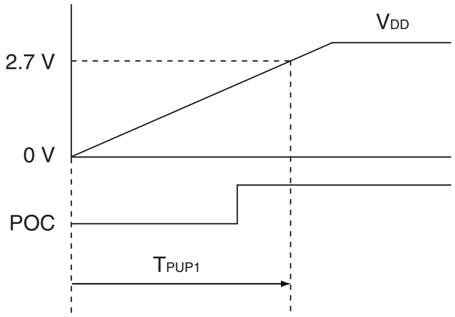


(9) Power supply starting time

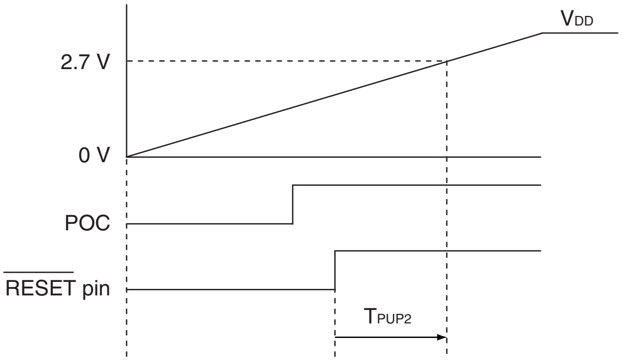
(T_A = –40 to +85°C, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Starting maximum time to V_{DD} min (2.7 V)[Note] (V_{DD}: 0 V→2.7 V)	t_{PUP1}	LVI starting option invalid When pin RESET intact			3.6	ms
Starting maximum time to V_{DD} min (2.7 V)[Note] (pin RESET release→V_{DD}: 2.7 V)	t_{PUP2}	LVI starting option invalid When pin RESET use			1.9	ms

Note Start a power supply in time shorter than this when LVI staring option invalid.



Pin RESET intact



Pin RESET use

(10) LCD circuit characteristics

(a) Static display mode

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD drive voltage	V_{LCD}				V_{DD}	V
LCD divider resistor [Note 1]	R_{LCD}		5	15	45	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, 2.7 V ≤ V_{DD} ≤ EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD drive voltage	V_{LCD}				V_{DD}	V
LCD divider resistor [Note 1]	R_{LCD}		5	15	45	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.

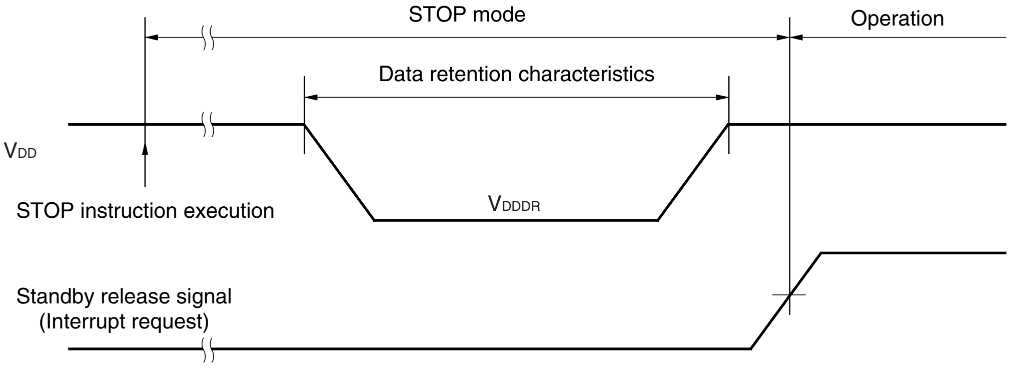
29.5 Data Retention Characteristics

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}		1.44Note		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.

Data Retention Timing



29.6 Flash EEPROM Programming Characteristics

(1) Basic characteristics

(T_A = −40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
V_{DD} supply current	I_{DD}					4.5	11.0	mA
Number of rewrites per chip [Note]	C_{erwr}	1 erase + 1 write after erase = 1 rewrite	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: 5 years	10000			Times

Note When a product is first written after shipment, “erase → write” and “write only” are both taken as one rewrite.

(2) Serial write operation characteristics

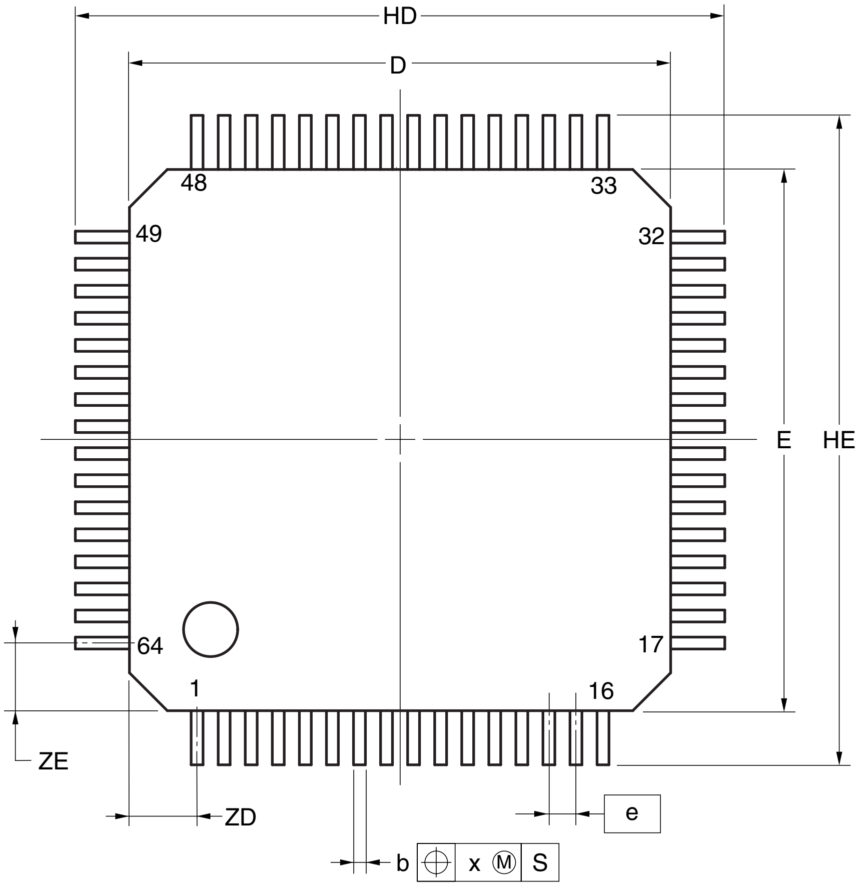
Refer to "Single Power Supply Flash Standard System Specification for 78K0/MF2 parameter (IDS-4006)".

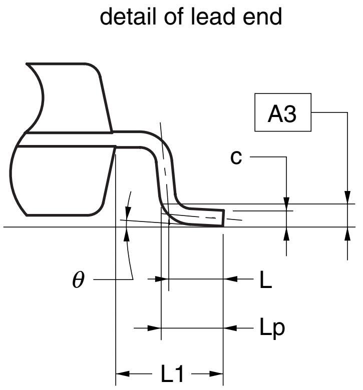
(3) Self write operation characteristics

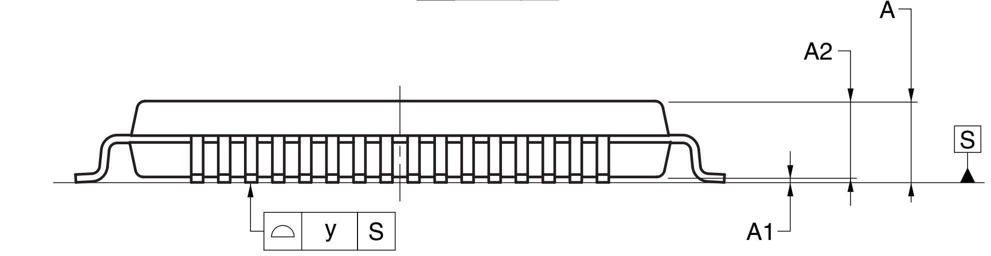
Refer to "Single Power Supply Flash Standard System Specification for 78K0/MF2 parameter (IDS-4006)".

<R> • *μ*PD78F0836GBA-GAH-AX, 78F0836GBA2-GAH-AX, 78F0837GBA-GAH-AX, 78F0837GBA2-GAH-AX, 78F0844GBA-GAH-AX, 78F0844GBA2-GAH-AX, 78F0845GBA-GAH-AX, 78F0845GBA2-GAH-AX

64-PIN PLASTIC LQFP(FINE PITCH)(10x10)







(UNIT:mm)

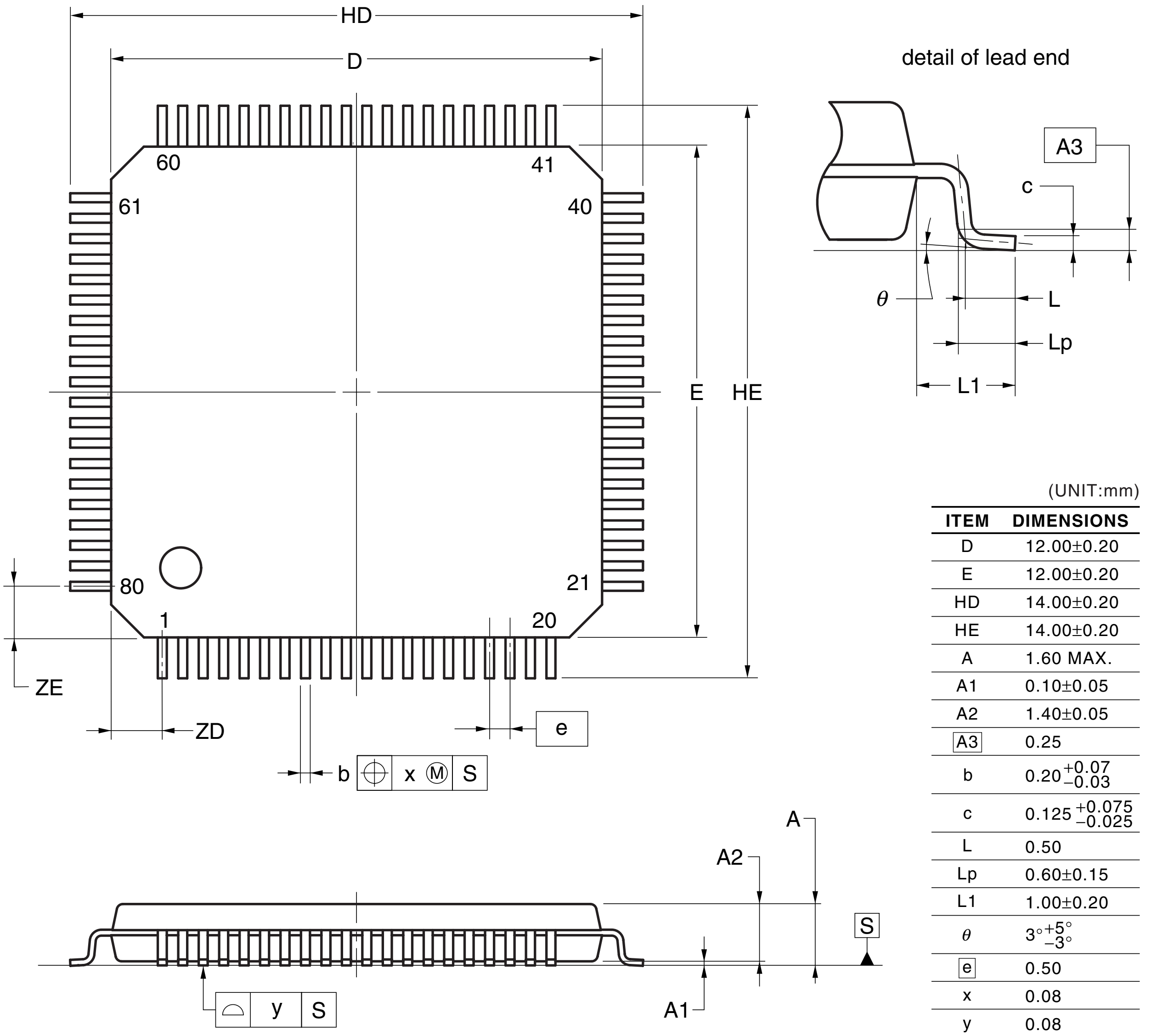
ITEM	DIMENSIONS
D	10.00±0.20
E	10.00±0.20
HD	12.00±0.20
HE	12.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
$\boxed{\text{A3}}$	0.25
b	$0.20^{+0.07}_{-0.03}$
c	$0.125^{+0.075}_{-0.025}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
$\boxed{\text{e}}$	0.50
x	0.08
y	0.08
ZD	1.25
ZE	1.25

P64GB-50-GAH

NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

<R> • **μPD78F0838GKA-GAK-AX, 78F0838GKA2-GAK-AX, 78F0839GKA-GAK-AX, 78F0839GKA2-GAK-AX, 78F0840GKA-GAK-AX, 78F0840GKA2-GAK-AX, 78F0841GKA-GAK-AX, 78F0841GKA2-GAK-AX, 78F0842GKA-GAK-AX, 78F0842GKA2-GAK-AX, 78F0843GKA-GAK-AX, 78F0843GKA2-GAK-AX, 78F0846GKA-GAK-AX, 78F0846GKA2-GAK-AX, 78F0847GKA-GAK-AX, 78F0847GKA2-GAK-AX, 78F0848GKA-GAK-AX, 78F0848GKA2-GAK-AX, 78F0849GKA-GAK-AX, 78F0849GKA2-GAK-AX**

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
$\boxed{\text{A3}}$	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	$3°^{+5°}_{-3°}$
$\boxed{\text{e}}$	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.

CHAPTER 31 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 31-1. Surface Mounting Type Soldering Conditions

<R>

- **64-pin plastic LQFP (10 × 10)**
  μPD78F0836GBA-GAH-AX, 78F0836GBA2-GAH-AX, 78F0837GBA-GAH-AX, 78F0837GBA2-GAH-AX, 78F0844GBA-GAH-AX, 78F0844GBA2-GAH-AX, 78F0845GBA-GAH-AX, 78F0845GBA2-GAH-AX
- **80-pin plastic LQFP (12 × 12)**
  μPD78F0838GKA-GAK-AX, 78F0838GKA2-GAK-AX, 78F0839GKA-GAK-AX, 78F0839GKA2-GAK-AX, 78F0840GKA-GAK-AX, 78F0840GKA2-GAK-AX, 78F0841GKA-GAK-AX, 78F0841GKA2-GAK-AX, 78F0842GKA-GAK-AX, 78F0842GKA2-GAK-AX, 78F0843GKA-GAK-AX, 78F0843GKA2-GAK-AX, 78F0846GKA-GAK-AX, 78F0846GKA2-GAK-AX, 78F0847GKA-GAK-AX, 78F0847GKA2-GAK-AX, 78F0848GKA-GAK-AX, 78F0848GKA2-GAK-AX, 78F0849GKA-GAK-AX, 78F0849GKA2-GAK-AX

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	T. B. D.	T. B. D.
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

Caution Do not use different soldering methods together (except for partial heating).

CHAPTER 32 CAUTIONS FOR WAIT

32.1 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 **Table 32-1**). This must be noted when real-time processing is performed.

32.2 Peripheral Hardware That Generates Wait

Table 32-1 lists the registers that issue a wait request when accessed by the CPU, and the number of CPU wait clocks.

Table 32-1. Registers That Generate Wait and Number of CPU Wait Clocks (1/2)

Peripheral Hardware	Register	Access	Number of Wait Clocks
16-bit timer/event counter TMPn (n = 0 to 4)	TPnCNT	Read	1 clock (fixed)
	TPnCCR0, TPnCCR1 Note 1	Read	1 clock (fixed)
	TPnCCR0, TPnCCR1 Note 2	Write	0
Serial interface UART60	ASIS60	Read	1 clock (fixed)
Serial interface UART61	ASIS61	Read	1 clock (fixed)
Serial interface IIC0	IICS0	Read	1 clock (fixed)
A/D converter	ADM	Write	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)
	ADS	Write	
	ADPC	Write	
	ADCR	Read	
	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_{CPU} to $f_{CPU}/16$) f_{PRS}: Peripheral hardware clock frequency <Conditions for maximum/minimum number of wait clocks> • Maximum number of times: Maximum speed of CPU (f_{CPU}), lowest speed of A/D conversion clock ($f_{PRS}/12$) • Minimum number of times: Minimum speed of CPU ($f_{CPU}/16$), highest speed of A/D conversion clock ($f_{PRS}/2$)		

Notes 1. During capture operation.

2. During compare operation.

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 32-1. Registers That Generate Wait and Number of CPU Wait Clocks (2/2)

Peripheral Hardware	Register	Access	Number of Wait Clocks
Stepper motor controller/driver (with ZPD)	MCNTC0	Write	1 to 3 clocks (when f_{MC0} = f_{PRS} is selected) 1 to 5 clocks (when f_{MC0} = $f_{PRS}/2$ is selected) 1 to 9 clocks (when f_{MC0} = $f_{PRS}/2^2$ is selected) 2 to 17 clocks (when f_{MC0} = $f_{PRS}/2^3$ is selected) 3 to 33 clocks (when f_{MC0} = $f_{PRS}/2^4$ is selected) 5 to 65 clocks (when f_{MC0} = $f_{PRS}/2^5$ is selected) 9 to 129 clocks (when f_{MC0} = $f_{PRS}/2^6$ is selected) 17 to 266 clocks (when f_{MC0} = $f_{PRS}/2^7$ is selected)
	MCMPCn (n = 1 to 4)	Write	
	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 =2 f_{CPU}/f_{MC0} + 1 * Fraction is truncated. f_{MC0}: Stepping motor controller/driver clock frequency (f_{PRS} to $f_{PRS}/2^7$) f_{CPU}: CPU clock frequency (f_{CPU} to $f_{CPU}/16$) f_{PRS}: Peripheral hardware clock frequency <Conditions for maximum/minimum number of wait clocks> • Maximum number of times: Maximum speed of CPU (f_{CPU}), lowest speed of Stepper motor controller/driver clock ($f_{PRS}/2^7$) • Minimum number of times: Minimum speed of CPU ($f_{CPU}/16$), highest speed of Stepper motor controller/driver clock (f_{PRS})		
Sound generator	SG0FL	Write	1 to 5 clocks (when f_{SG0} = $f_{PRS}/2$ is selected)
	SG0FH		
	SG0PWM		
	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 =2 f_{CPU}/f_{SG0} + 1 * Fraction is truncated. f_{SG0}: Sound Generator clock frequency ($f_{PRS}/2$) f_{CPU}: CPU clock frequency (f_{CPU} to $f_{CPU}/16$) f_{PRS}: Peripheral hardware clock frequency <Conditions for maximum/minimum number of wait clocks> • Maximum number of times: Maximum speed of CPU (f_{CPU}), lowest speed of Sound Generator clock ($f_{PRS}/2$) • Minimum number of times: Minimum speed of CPU ($f_{CPU}/16$), highest speed of Sound Generator clock ($f_{PRS}/2$)		

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 32-2. RAM Access That Generate Wait and Number of CPU Wait Clocks

Peripheral Hardware	Register	Access	Number of Wait Clocks		Cause
			MIN.	MAX.	
CAN	Global Reg.	Read/Write	1	1	Synchronizaition of NPB signals with VPCLK <Calculating number of wait clocks> MIN. ROUNDUP[(1/FVPCLK) × 1/(1/FVPSTB)] MAX. ROUNDUP[(1/FVPCLK) × 2/(1/FVPSTB)]
	CANmodule Reg.				
	C0RGPT	Read	2	14	Synchronization of NPB signals with VPCLK RAM access delay (1 RAM – RD access) <Calculating number of wait clocks> MIN. ROUNDUP[(1/FCANCLK) × 3/(1/FVPSTB)] MAX. ROUNDUP[(1/FCANCLK) × 4/(1/FVPSTB)]
	C0LIPT				
	C0TGPT				
	C0LOPT				
	Message Buf.				
	Message Buf.	Write(8 bit)	2	17	Synchronization of NPB signals with VPCLK RAM access delay (1RAM – RD + 1RAM – WR access) <Calculating number of wait clocks> MIN. ROUNDUP[(1/FCANCLK) × 4/(1/FVPSTB)] MAX. ROUNDUP[(1/FCANCLK) × 5/(1/FVPSTB)]
	Message Buf.	Write(16 bit)	1	11	Synchronization of NPB signals with VPCLK RAM access delay (1 RAM – WR access) <Calculating number of wait clocks> MIN. ROUNDUP[(1/FCANCLK) × 2/(1/FVPSTB)] MAX. ROUNDUP[(1/FCANCLK) × 3/(1/FVPSTB)]

Caution When Value is ΦCANMOD(CAN module system clock) ≥ 2 MHz.

Remark	FVPCLK:	VPCLK frequency
	FVPSTB:	VPSTB frequency
	FCANCLK:	AFCAN macro frequency

32.3 Example of Wait Occurrence

- Serial interface UART61

  <On execution of MOV A, ASIS61>

  Number of execution clocks: 6

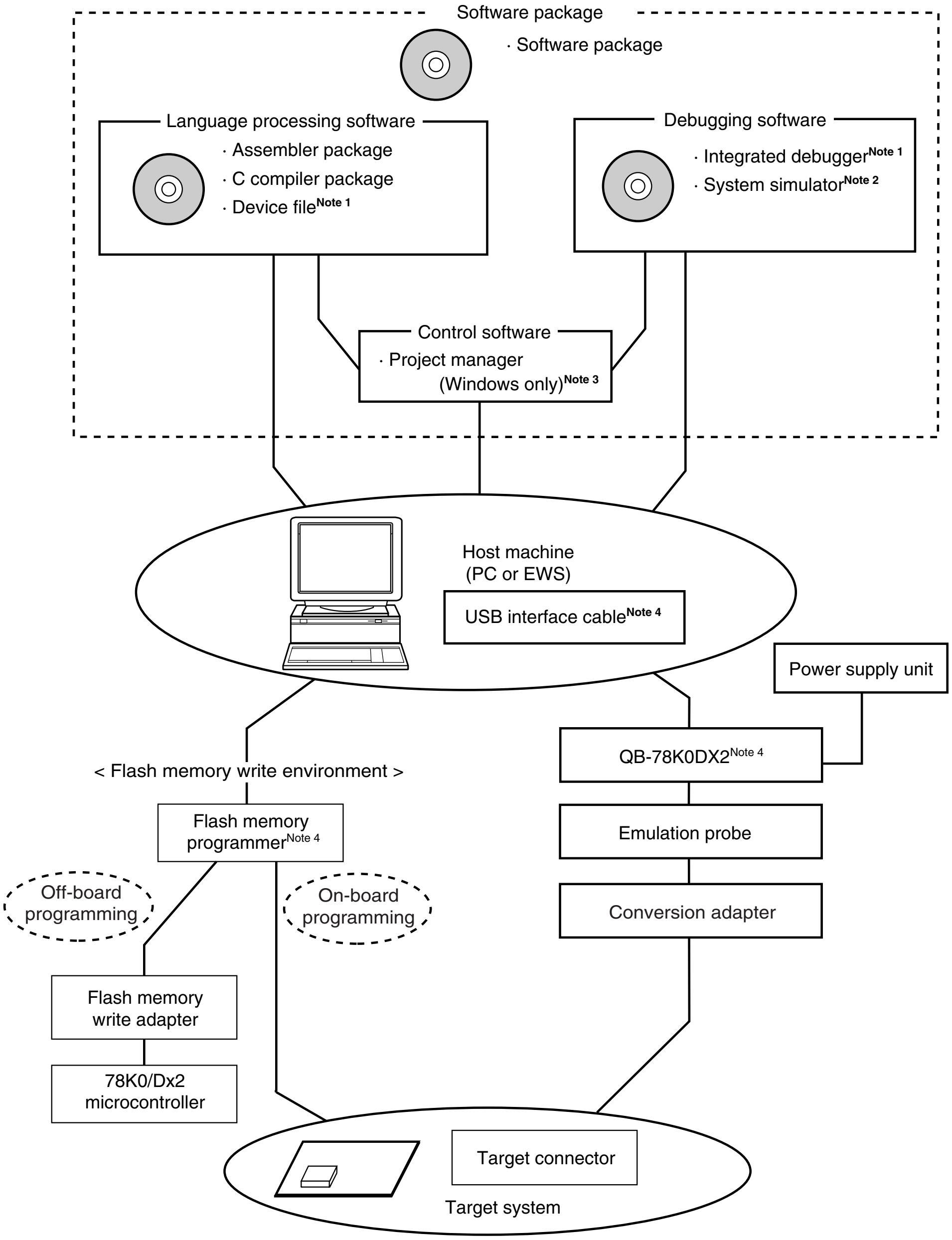
  (5 clocks when data is read from a register that does not issue a wait (MOV A, sfr).)

APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for the development of systems that employ the 78K0/Dx2.
Figure A-1 shows the development tool configuration.

Figure A-1. Development Tool Configuration (1/3)

(1) When using the in-circuit emulator QB-78K0DX2



Notes 1. Download the device file for 78K0/Dx2 microcontrollers (DF780849) and the integrated debugger ID78K0-QB from the download site for development tools (http://www.necel.com/micro/en/ods/index.html).

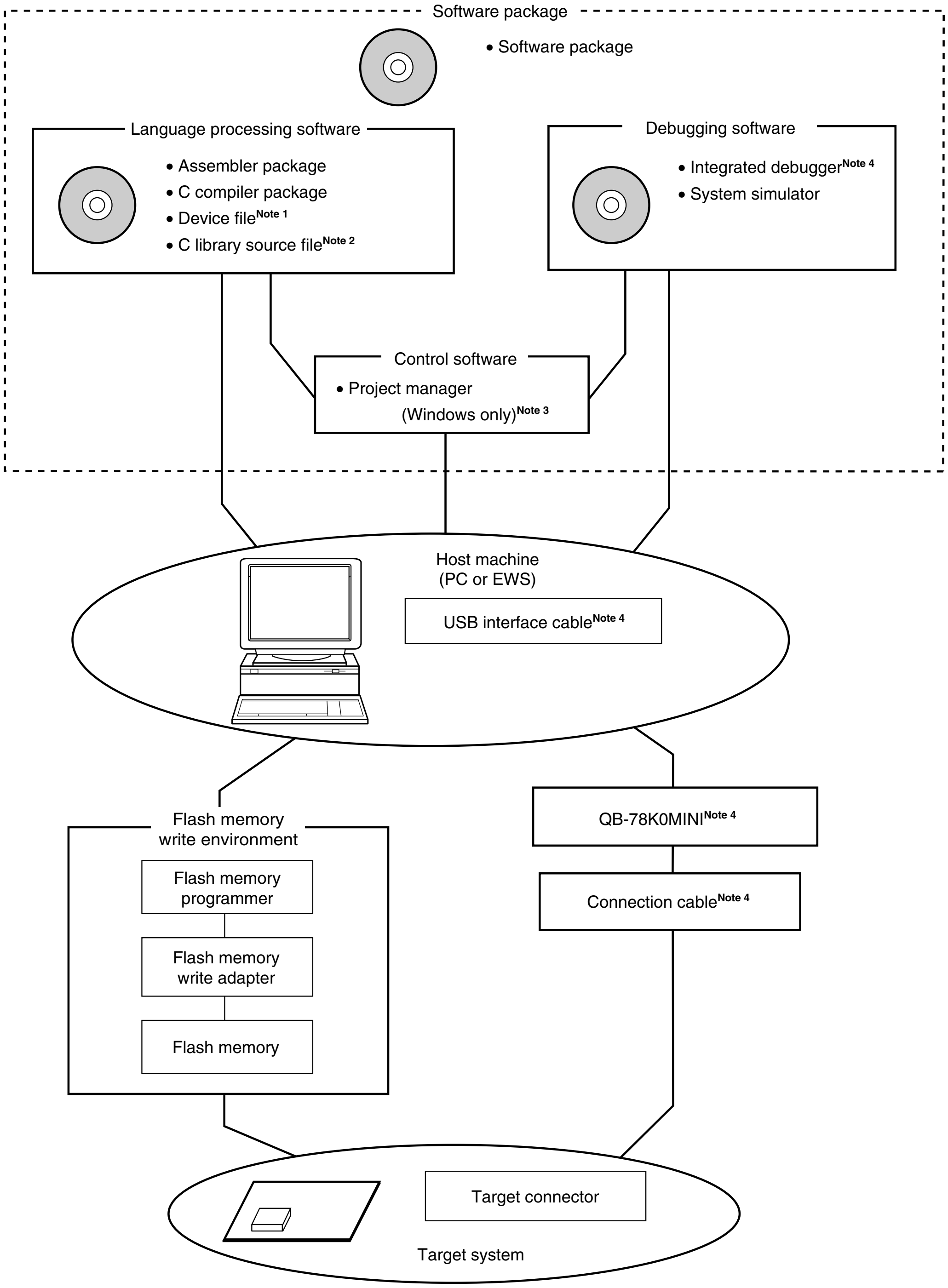
2. SM+ for 78K0 (instruction simulation version) is included in the software package. SM+ for 78K0/Dx2 (instruction + peripheral simulation version) is not included.

3. The project manager PM+ is included in the assembler package.
PM+ cannot be used other than with Windows™.

4. QB-78K0DX2 is supplied with the integrated debugger ID78K0-QB, a USB interface cable, the on-chip debug emulator with programming function QB-MINI2, connection cables (10-pin and 16-pin cables), and the 78K0-OCD board. Any other products are sold separately.

Figure A-1. Development Tool Configuration (2/3)

(2) When using the on-chip debug emulator QB-78K0MINI



Notes 1. Download the device file for 78K0/Dx2 (DF780849) and the integrated debugger ID78K0-QB from the download site for development tools (http://www.necel.com/micro/en/ods/index.html).

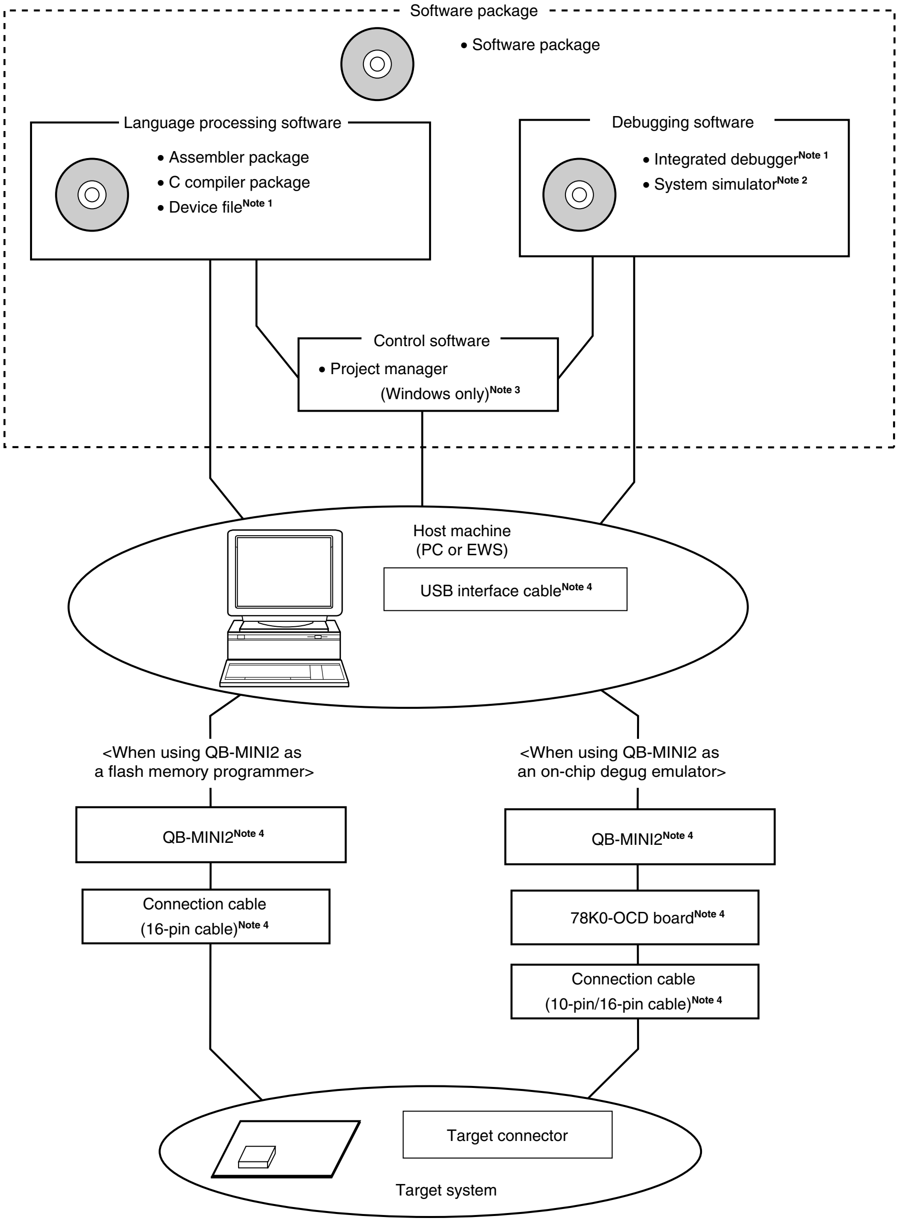
2. The C library source file is not included in the software package.

3. The project manager PM+ is included in the assembler package.
PM+ is only used for Windows.

4. On-chip debug emulator QB-78K0MINI is supplied with integrated debugger ID78K0-QB, USB interface cable, and connection cable. Any other products are sold separately.

Figure A-1. Development Tool Configuration (3/3)

(3) When using the on-chip debug emulator with programming function QB-MINI2



Notes 1. Download the device file for 78K0/Dx2 microcontrollers (DF780849) and the integrated debugger ID78K0-QB from the download site for development tools (http://www.necel.com/micro/en/ods/index.html).

2. SM+ for 78K0 (instruction simulation version) is included in the software package. SM+ for 78K0/Dx2 (instruction + peripheral simulation version) is not included.

3. The project manager PM+ is included in the assembler package.
PM+ cannot be used other than with Windows.

4. QB-MINI2 is supplied with USB interface cable, connection cables (10-pin cable and 16-pin cable), and 78K0-OCD board. Any other products are sold separately. In addition, download the software for operating the QB-MINI2 from the download site for development tools (http://www.necel.com/micro/en/ods/index.html).

A.1 Software Package

SP78K0 78K0 microcontroller software package	Development tools (software) common to the 78K0 microcontrollers are combined in this package.

A.2 Language Processing Software

RA78K0 [Note 1] 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 (DF780849). **<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 (PM+) on Windows. PM+ is included in assembler package.
CC78K0 [Note 1] 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. **<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 (PM+) on Windows. PM+ is included in assembler package.
DF780849 [Note 2] Device file	This file contains information peculiar to the device. This device file should be used in combination with a tool (RA78K0, CC78K0, ID78K0-QB, and the system simulator). The corresponding OS and host machine differ depending on the tool to be used.

Notes 1. If the versions of RA78K0 and CC78K0 are Ver.4.00 or later, different versions of RA78K0 and CC78K0 can be installed on the same machine.

2. The DF780849 can be used in common with the RA78K0, CC78K0, ID78K0-QB, and the system simulator. Download the DF780849 from the download site for development tools (http://www.necel.com/micro/en/ods/index.html).

A.3 Flash Memory Programming Tools

A.3.1 When using flash memory programmer PG-FP5, FL-PR5, PG-FP4, and FL-PR4

<R>

PG-FP5, FL-PR5, PG-FP4 Note 1, FL-PR4 Flash memory programmer	Flash memory programmer dedicated to microcontrollers with on-chip flash memory.
FA-64GB-GAH-B, FA-78F0849GB-GAH-RX, FA-78F0849GB-UEU-RX, FA-80GK-GAK-B, FA-78F0849GK-GAK-RX Flash memory programming adapter	Flash memory programming adapter used connected to the flash memory programmer for use. • FA-64GB-GAH-B, FA-78F0849GB-GAH-RX, FA-78F0849GB-UEU-RX: 64-pin plastic LQFP (GB-GAH and GB-UEU types) • FA-80GK-GAK-B, FA-78F0849GK-GAK-RX: 80-pin plastic LQFP (GK-GAK types)

Note Phase-out

Remarks 1. FL-PR5, FL-PR4, and FA-64GB-GAH-B, FA-78F0849GB-GAH-RX, FA-78F0849GB-UEU-RX, FA-80GK-GAK-B, FA-78F0849GK-GAK-RX are products of Naito Densei Machida Mfg. Co., Ltd (http://www.ndk-m.co.jp/, TEL: +81-42-750-4172).

2. Use the latest version of the flash memory programming adapter.

A.3.2 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/Dx2. 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 and connection cables (10-pin cable and 16-pin cable), and the 78K0-OCD board. A 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/en/ods/index.html).

A.4 Debugging Tools (Hardware)

A.4.1 When using in-circuit emulator QB-78K0DX2

QB-78K0DX2 In-circuit emulator	The in-circuit emulator serves to debug hardware and software when developing application systems using the 78K0/Dx2. It supports 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 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-64GB-EA-04T QB-80GK-EA-01T Exchange adapter	This adapter is used to perform the pin conversion from the in-circuit emulator to the target connector. • QB-64GB-EA-04T: For 64-pin plastic LQFP (GB-GAH, GB-UEU type) • QB-80GK-EA-01T: For 80-pin plastic LQFP (GK-8EU, GK-GAK type)
QB-64GB-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-64GB-YS-01T: For 64-pin plastic LQFP (GB-GAH, GB-UEU type) • QB-80GK-YS-01T: For 80-pin plastic LQFP (GK-8EU, GK-GAK type)
QB-64GB-YQ-01T QB-80GK-YQ-01T YQ connector	This YQ connector is used to connect the target connector and exchange adapter. • QB-64GB-YQ-01T: For 64-pin plastic LQFP (GB-GAH, GB-UEU type) • QB-80GK-YQ-01T: For 80-pin plastic LQFP (GK-8EU, GK-GAK type)
QB-64GB-HQ-01T QB-80GK-HQ-01T Mount adapter	This mount adapter is used to mount the target device with socket. • QB-64GB-HQ-01T: For 64-pin plastic LQFP (GB-GAH, GB-UEU type) • QB-80GK-HQ-01T: For 80-pin plastic LQFP (GK-8EU, GK-GAK type)
QB-64GB-NQ-01T QB-80GK-NQ-01T Target connector	This target connector is used to mount on the target system. • QB-64GB-NQ-01T: For 64-pin plastic LQFP (GB-GAH, GB-UEU type) • QB-80GK-NQ-01T: For 80-pin plastic LQFP (GK-8EU, GK-GAK type)

Remarks 1. The QB-78K0DX2 is supplied with the integrated debugger ID78K0-QB, a USB interface cable, 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/en/ods/index.html) when using the QB-MINI2.

2. The packed contents differ depending on the part number, as follows.

Package Contents Part Number	In-Circuit Emulator	Emulation Probe	Exchange Adapter	YQ Connector	Target Connector
QB-78K0DX2-ZZZ (-EE)	QB-78K0DX2	Not included			
QB-78K0DX2-T64GB		QB-80-EP-01T	QB-64GB-EA-04T	QB-64GB-YQ-01T	QB-64GB-NQ-01T
QB-78K0DX2-T80GK			QB-80GK-EA-01T	QB-80GK-YQ-01T	QB-80GK-NQ-01T

A.4.2 When using on-chip debug emulator QB-78K0MINI

QB-78K0MINI [Notes 1, 2] On-chip debug emulator	This on-chip debug emulator serves to debug hardware and software when developing application systems using the 78K0/Dx2. It supports the integrated debugger (ID78K0-QB). This emulator should be used in combination with connection cable and a USB interface cable that is used to connect the host machine.
Target connector specifications	10-pin general-purpose connector (2.54 mm pitch)

Notes 1. The QB-78K0MINI is supplied with a USB interface cable and a connection cable. As control software, the integrated debugger ID78K0-QB is supplied.

2. Phase-out

A.4.3 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/Dx2. 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 and connection cables (10-pin cable and 16-pin cable), and the 78K0-OCD board. A 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/en/ods/index.html).

A.5 Debugging Tools (Software)

ID78K0-QB[Note] Integrated debugger	This debugger supports the in-circuit emulators for the 78K0 microcontrollers. The ID78K0-QB is Windows-based software. It has improved C-compatible debugging functions and can display the results of tracing with the source program using an integrating window function that associates the source program, disassemble display, and memory display with the trace result. It should be used in combination with the device file (DF780849).
SM+ for 78K0 SM+ for 78K0/Dx2 System simulator	System simulator is Windows-based software. It is used to perform debugging at the C source level or assembler level while simulating the operation of the target system on a host machine. Use of system simulator allows the execution of application logical testing and performance testing on an independent basis from hardware development, thereby providing higher development efficiency and software quality. System simulator should be used in combination with the device file (DF780849). The following two types of system simulators supporting the 78K0/Dx2 microcontrollers are available. • SM+ for 78K0 (instruction simulation version) This can only simulate a CPU. It is included in the software package. • SM+ for 78K0/Dx2 (instruction + peripheral simulation version) This can simulate a CPU and peripheral hardware (ports, timers, serial interfaces, etc.). It is sold separately from the software package.

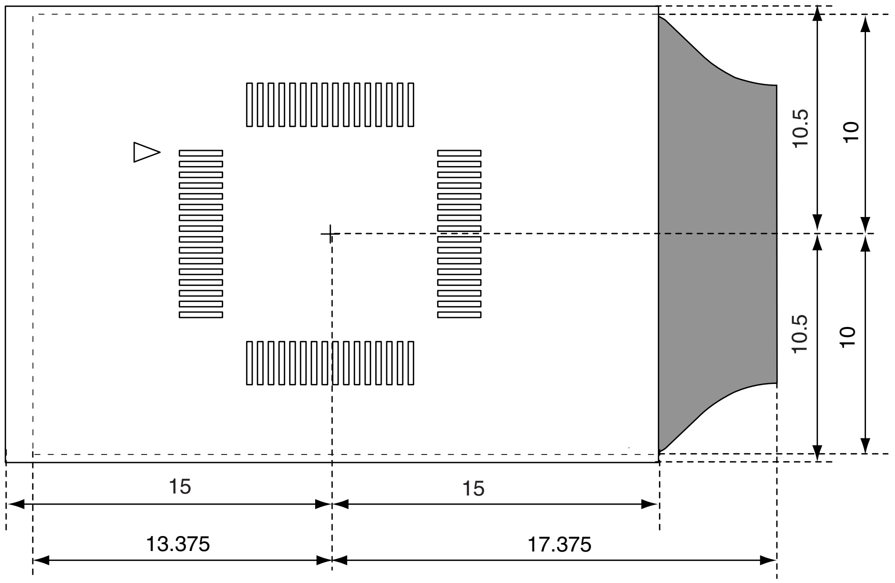
Note Download the ID78K0-QB from the download site for development tools (http://www.necel.com/micro/en/ods/index.html).

APPENDIX B NOTES ON TARGET SYSTEM DESIGN

This chapter shows areas on the target system where component mounting is prohibited and areas where there are component mounting height restrictions when the QB-78K0DX2 is used.

(a) Case of 64-pin GB package

Figure B-1. The Restriction Domain on a Target System (Case of 64-pin GB Package)



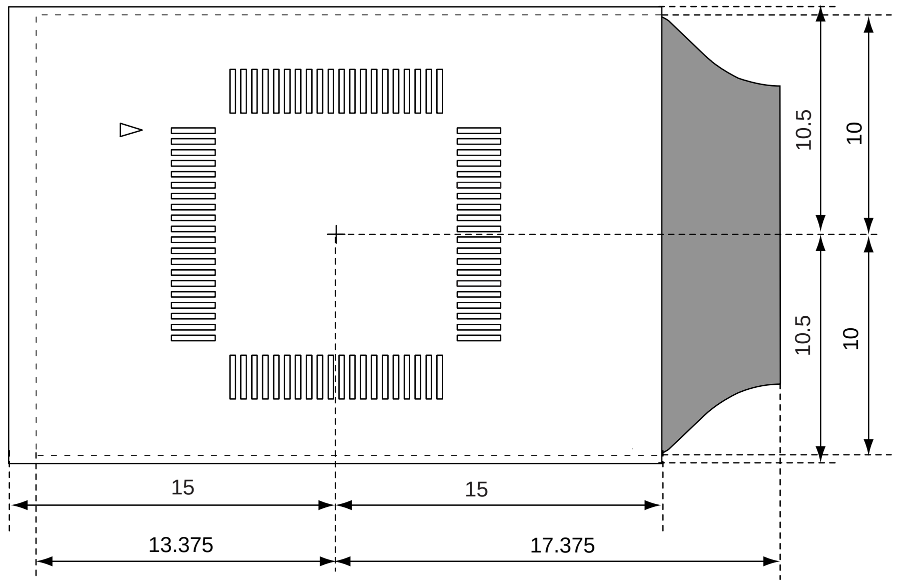
□ : Exchange adapter area: Components up to 17.45 mm in height can be mounted[Note]

▨ : Emulation probe tip area: Components up to 24.45 mm in height can be mounted[Note]

Note Height can be adjusted by using space adapters (each adds 2.4 mm)

(b) Case of 80-pin GK package

Figure B-2. The Restriction Domain on a Target System (Case of 80-pin GK Package)



□ : Exchange adapter area: Components up to 17.45 mm in height can be mounted[Note]

▨ : Emulation probe tip area: Components up to 24.45 mm in height can be mounted[Note]

Note Height can be adjusted by using space adapters (each adds 2.4 mm)

APPENDIX C REGISTER INDEX

C.1 Register Index (In Alphabetical Order with Respect to Register Names)

C.2 Register Index (In Alphabetical Order with Respect to Register Symbol)

APPENDIX D REVISION HISTORY

D.1 Main Revisions in this Edition

(1/4)

Page	Description	Edition
p. 6	Change of part number in **INTRODUCTION**.	Ver. 1.1
p. 7	Change of “Document No.” in **INTRODUCTION**.	Ver. 1.1
p. 19	Change of “CAN [channel]” in **1.1 Features**.	Ver. 1.1
p. 20	Change of Operating ambient temperature **1.1 Features**.	Ver. 1.1
p. 21	Change of part number in **1.3 Ordering Information**.	Ver. 1.1
pp. 27 to 30	Deletion of “Bank” and change of input/output for TIOP40/P00 in **1.5 Block Diagram**.	Ver. 1.1
pp. 32, 34, 36, 38	Change of “External” in **1.6 Outline of Functions**.	Ver. 1.1
pp. 31, 33, 35, 37	Change of “Timer outputs” in **1.6 Outline of Functions**.	Ver. 1.1
p. 39	Change of Outline of timers in **1.6 Outline of Functions**.	Ver. 1.1
pp. 41 to 52	Addition of “During Reset” in **Table 2-2. Port Pins for 78K0/DE2** to **Table 2-5. Non-port Pins for 78K0/DF2**.	Ver. 1.1
p. 55	Change of Cautions 1, 2 in **2.2.4 (2) (a) SEG4 to SEG11**.	Ver. 1.1
pp. 61, 63, 65, 67	Change of Note 3 in **Table 2-6. Pin I/O Circuit Types**.	Ver. 1.1
pp. 62, 63	Change of “Recommended Connection of Unused Pins” for P60/SCL0/INTP1, P61/SDA0/INTP3 in **Table 2-6. Pin I/O Circuit Types**.	Ver. 1.1
pp. 62, 64, 66, 68	Change of “I/O” for FLMD0 in **Table 2-6. Pin I/O Circuit Types**.	Ver. 1.1
pp. 75 to 78	Change of figures below: • **Figure 3-1. Memory Map (μPD78F0836, 78F0838, 78F0840, 78F0842)** • **Figure 3-2. Memory Map (μPD78F0844, 78F0846, 78F0848)** • **Figure 3-3. Memory Map (μPD78F0837, 78F0839, 78F0841, 78F0843)** • **Figure 3-4. Memory Map (μPD78F0845, 78F0847, 78F0849)**	Ver. 1.1
p. 82	Addition of **3.1.4 Extended function register (EFR) area**.	Ver. 1.1
pp. 83 to 86	Change of figures below: • **Figure 3-5. Correspondence between Data Memory and Addressing (μPD78F0836, 78F0838, 78F0840, 78F0842)** • **Figure 3-6. Correspondence between Data Memory and Addressing (μPD78F0844, 78F0846, 78F0848)** • **Figure 3-7. Correspondence between Data Memory and Addressing (μPD78F0837, 78F0839, 78F0841, 78F0843)** • **Figure 3-8. Correspondence between Data Memory and Addressing (μPD78F0845, 78F0847, 78F0849)**	Ver. 1.1
p. 92	Change of “Target” in **3.2.3 Special Function Registers (SFRs)**.	Ver. 1.1
p. 98	Deletion of PFCMD, PFS, and FLPMC in **Table 3-7. Special Function Register List**.	Ver. 1.1
p. 99	Change of title “**Table 3-8. Extended Function Register List**”.	Ver. 1.1
p. 100	Addition of Note 1 and Note 2 to MCMP10, MCMP11, MCMP20, MCMP21, MCMP30, MCMP31, MCMP40, MCMP41, and MCMPC1 to MCMPC4 in **Table 3-8. Extended Function Register List**.	Ver. 1.1

(2/4)

Page	Description	Edition
p. 126	Change of Caution 2 in **4.2.4 Port 3**.	Ver. 1.1
p. 160	Change of “Port mode selection (n = 0 to 4)” in **Figure 4-32. Format of Stepper Motor Port Mode Control Register (SMPC)**.	Ver. 1.1
pp. 163 to 169	Addition of “SMPC” and change of “ISC” in **Table 4-4. Settings of LCDPFALL, LCDPF0, LCDPF3, SMPC, ISC, Port Mode Register, and Output Latch When Using Alternate Function**.	Ver. 1.1
p. 215	Change of **Figure 6-1. Block Diagram of TMPn**.	Ver. 1.1
p. 219	Change of description for TPnSYE in **6.4 (2) TMPn control register 1 (TPnCTL1)**.	Ver. 1.1
pp. 230 to 232	Change of description and addition of figures in **6.4 (10) Input switch control register (ISC)**.	Ver. 1.1
p. 286	Change of **Figure 6-31. Configuration in Free-running Timer Mode**.	Ver. 1.1
pp. 330, 333, 334	Change of the tables below: • **Table 8-1. Watch Timer Interrupt Time** • **Table 8-2. Interval Timer Interval Time** • **Table 8-4. Watch Timer Interrupt Time** • **Table 8-5. Interval Timer Interval Time**	Ver. 1.1
p. 332	Change of description for WTM6 to WTM4 in **Figure 8-2. Format of Watch Timer Operation Mode Register (WTM)**.	Ver. 1.1
pp. 378, 379	Change of **Figure 12-5. Block Diagram of Serial Interface UART60** and **Figure 12-6. Block Diagram of Serial Interface UART61**.	Ver. 1.1
p. 382, 383	Change of bit 0, addition of Caution 1, and deletion of Caution 9 in **Figure 12-7. Format of Asynchronous Serial Interface Operation Mode Register 60 (ASIM60)**.	Ver. 1.1
p. 384, 385	Change of bit 0, addition of Caution 1, and deletion of Caution 9 in **Figure 12-8. Format of Asynchronous Serial Interface Operation Mode Register 61 (ASIM61)**.	Ver. 1.1
pp. 399 to 401	Change of description in **Figure 12-19. Format of Input Switch Control Register (ISC)** and addition of **Figure 12-20. ISC Register Value Selection**.	Ver. 1.1
p. 414	Change of description in **12.4.2 (2) (e) Normal reception**.	Ver. 1.1
p. 415	Change of description in **12.4.2 (2) (f) Reception error** and deletion of **Figure 12-28. Reception Error Interrupt**.	Ver. 1.1
p. 665	Change of description in **16.1.2 ZPD introduction**.	Ver. 1.1
p. 667	Addition of “TM51 output” and change of “Noise removal”, “Rising edge detection” in **Figure 16-1. Stepper Motor Controller/Driver Block Diagram**.	Ver. 1.1
p. 671	Change of description in **16.2 (3) Compare registers for cosine side (MCMPk1) (k = 1 to 4)**.	Ver. 1.1
p. 672	Change of Remarks 1, 2 in **16.2 (4) Combined compare registers (MCMPkHW) (k = 1 to 4)**.	Ver. 1.1
p. 675	Change of “Port mode selection (n = 0 to 4)” in **Figure 16-4. Format of Stepper Motor Port Mode Control Register (SMPC)**.	Ver. 1.1
p. 676	Addition of Note and change of ZPDS1 register bit names in **Figure 16-5. Format of ZPD detection voltage setting registers (ZPDS0, ZPDS1).**	Ver. 1.1
p. 677	Change of “Selected clock” in **Figure 16-6. Format of ZPD Flag Detection Clock Setting Register (CMPCTL).**	Ver. 1.1
p. 677	Deletion of ZPDRVEN bit and change of bit names in **Figure 16-7. Format of ZPD Operational Control Register (ZPDEN).**	Ver. 1.1

(3/4)

Page	Description	Edition
p. 678	Change of Remark in **16.3.1 Stepper motor controller/driver operation** and change of description in **16.3.1 (1) Driving meters**.	Ver. 1.1
p. 679	Deletion of 2. in **16.3.1 (5) Detecting zero points**.	Ver. 1.1
p. 690	Change of “LCD source clock (f_{LCD}) selection” in **Figure 17-4. Format of LCD Clock Control Register**.	Ver. 1.1
p. 693	Change of Remark in **17.4 Setting LCD Controller/Driver**.	Ver. 1.1
p. 710	Deletion of V_{LC3} in **Table 17-8. LCD Drive Voltages (with On-Chip Voltage Divider Resistors)**.	Ver. 1.1
p. 723	Deletion of Note 3 in **Table 19-1. Interrupt Source List**.	Ver. 1.1
p. 727	Deletion of Note 3 in **Table 19-2. Flags Corresponding to Interrupt Request Sources**.	Ver. 1.1
p. 732	Change of description in **(4) External interrupt rising edge enable register (EGP), external interrupt falling edge enable register (EGN)**, change of bit 5 of EGP, EGN and addition of Caution in **Figure 19-5. Format of External Interrupt Rising Edge Enable Register (EGP) and External Interrupt Falling Edge Enable Register (EGN)**, change of “Edge Detection Port” and change of Remark in **Table 19-3. Ports Corresponding to EGPn and EGNn**.	Ver. 1.1
p. 761	Addition of IIC0, stepper motor C/D (with ZPD), LCD controller/driver, and sound generator in **Table 21-1. Operation Statuses during Reset Period**.	Ver. 1.1
pp. 764, 765	Addition of IIC0, stepper motor C/D (with ZPD), LCD controller/driver, and sound generator in **Table 21-2. Hardware Statuses after Reset Acknowledgment**.	Ver. 1.1
p. 768	Change of **Figure 22-1. Block Diagram of Multiplier/Divider**.	Ver. 1.1
p. 772	Deletion of 6. in **22.4.1 Multiplication operation**.	Ver. 1.1
p. 773	Change of **Figure 22-6. Timing Chart of Multiplication Operation (00DAH × 0093H)**.	Ver. 1.1
p. 774	Deletion of 6. in **22.4.2 Division operation**.	Ver. 1.1
p. 775	Change of **Figure 22-7. Timing Chart of Division Operation (DCBA2586H ÷ 0018H)**.	Ver. 1.1
p. 807	Change of DF2 pin no. with UART60 for /RESET in **Table 26-3. Wiring between 78K0/Dx2 and Dedicated Flash Memory Programmer**.	Ver. 1.1
p. 810	Change of description in **26.4 Programming Environment**.	Ver. 1.1
p. 814	Change of Caution 3 and addition of Caution 4 in **26.6.6 Other signal pins**.	Ver. 1.1
p. 829	Deletion of **26.10.1 Registers used for self-programming function**.	Ver. 1.1
p. 832	Change of title “**26.12 Creating ROM Code to Place Order for Programmed Flash Product**”.	Ver. 1.1
p. 832	Change of description in **26.12 Creating ROM Code to Place Order for Programmed Flash Product**, **(1) Website**, and **(2) Downloading the HCU**.	Ver. 1.1
p. 832	Change of title “**26.12.1 Procedures for ROM code ordering process using HCU**”.	Ver. 1.1
p. 832	Change of Note in **26.12.1 Procedures for ROM code ordering process using HCU**.	Ver. 1.1
p. 836	Change of **Figure 27-3. Connection Example of QB-78K0MINI or QB-MINI2 and 78K0/Dx2 (When P31 and P32 Are Used)**.	Ver. 1.1
p. 838	Addition of description in **27.6 Restrictions and Cautions on On-Chip Debug Function**.	Ver. 1.1
p. 852	Change of ratings for V_{I1} in **29.1 Absolute Maximum Ratings**.	Ver. 1.1
p. 853	Change of ratings for I_{OH4}, I_{OL4} in **29.1 Absolute Maximum Ratings**.	Ver. 1.1
p. 856	Change of value for I_{OH3} in **29.3 DC Characteristics**.	Ver. 1.1
p. 857	Change of value for I_{OL3} in **29.3 DC Characteristics**.	Ver. 1.1

(4/4)

Page	Description	Edition
p. 858	Change of conditions for V_{OH3} in **29.3 DC Characteristics**.	Ver. 1.1
p. 859	Change of conditions for V_{OL3} in **29.3 DC Characteristics**.	Ver. 1.1
p. 862	Change of I_{ZPD} in **29.3 DC Characteristics**.	Ver. 1.1
p. 871	Addition of I_{CROSS}, t_{MO}, and Note 5 and change of Note 6 in **29.4 (5) Meter controller/driver and ZPD characteristics**.	Ver. 1.1
p. 873	Change of conditions for t_R, t_F and change of symbol for “Sound generator input timing” in **29.4 (6) Sound generator characteristics**.	Ver. 1.1
pp. 880, 881	Change of part number in **CHAPTER 30 PACKAGE DRAWINGS**.	Ver. 1.1
p. 882	Change of part number in **Table 31-1. Surface Mounting Type Soldering Conditions**.	Ver. 1.1
p. 893	Change of flash memory programmer name in **A.3.1 When using flash memory programmer PG-FP5, FL-PR5, PG-FP4, and FL-PR4**.	Ver. 1.1

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