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.

RENESAS

User's Manual

μPD703229Y, 70F3229Y

32-Bit Single-Chip Microcontrollers

Hardware

Document No. U16470EJ2V0UD00 (2nd edition)
Date Published December 2003 N CP(K)

© NEC Electronics Corporation 2003
Printed in Japan

[MEMO]

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

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

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

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

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

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

Purchase of NEC Electronics I^2C components conveys a license under the Philips I^2C Patent Rights to use these components in an I^2C system, provided that the system conforms to the I^2C Standard Specification as defined by Philips.

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

- **The information in this document is current as of December, 2003. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".
  The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.
  "Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots.
  "Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).
  "Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

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

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

M8E 02. 11-1

Regional Information

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

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

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

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

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

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

- **Sucursal en España**
  Madrid, Spain
  Tel: 091-504 27 87
- **Succursale Française**
  Vélizy-Villacoublay, France
  Tel: 01-30-67 58 00
- **Filiale Italiana**
  Milano, Italy
  Tel: 02-66 75 41
- **Branch The Netherlands**
  Eindhoven, The Netherlands
  Tel: 040-244 58 45
- **Tyskland Filial**
  Taeby, Sweden
  Tel: 08-63 80 820
- **United Kingdom Branch**
  Milton Keynes, UK
  Tel: 01908-691-133

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

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

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

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

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

J03.4

PREFACE

Readers

This manual is intended for users who wish to understand the functions of the μPD703229Y and 70F3229Y and design application systems using these products.

Purpose

This manual is intended to give users an understanding of the hardware functions of the μPD703229Y and 70F3229Y shown in the Organization below.

Organization

This manual is divided into two parts: Hardware (this manual) and Architecture (**V850ES Architecture User's Manual**).

Hardware	Architecture
• Pin functions	• Data types
• CPU function	• Register set
• On-chip peripheral functions	• Instruction format and instruction set
• Flash memory programming	• Interrupts and exceptions
• Electrical specifications	• Pipeline operation

How to Read This Manual

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

To find the details of a register where the name is known
→Refer to **APPENDIX A REGISTER INDEX**.

To understand the details of an instruction function
→Refer to the **V850ES Architecture User's Manual** available separately.

Register format
→The name of the bit whose number is in angle brackets (<>) in the figure of the register format of each register is defined as a reserved word in the device file.

To understand the overall functions of the μPD703229Y and 70F3229Y
→Read this manual according to the **CONTENTS**.

To know the electrical specifications of the μPD703229Y and 70F3229Y
→Refer to **CHAPTER 25 ELECTRICAL SPECIFICATIONS.**

The mark ★ shows major revised points.

Conventions	Data significance:	Higher digits on the left and lower digits on the right
	Active low representation:	$\overline{xxx}$ (overscore over pin or signal name)
	Memory map address:	Higher addresses on the top and lower addresses on the bottom
	Note:	Footnote for item marked with **Note** in the text
	Caution:	Information requiring particular attention
	Remark:	Supplementary information
	Numeric representation:	Binary ... xxxx or xxxxB
		Decimal ... xxxx
		Hexadecimal ... xxxxH
	Prefix indicating power of 2 (address space, memory capacity):	K (kilo): 2^{10} = 1,024
		M (mega): 2^{20} = $1{,}024^2$
		G (giga): 2^{30} = $1{,}024^3$

Related Documents

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

Documents related to μPD703229Y, 70F3229Y

Document Name	Document No.
V850ES Architecture User's Manual	U15943E
μPD703229Y, 70F3229Y Hardware User's Manual	This manual

Documents related to development tools (User's Manuals)

Document Name		Document No.
IE-V850ES-G1 (In-Circuit Emulator)		U16313E
IE-703229-G1-EM1 (In-Circuit Emulator Option Board)		To be prepared
IE-V850E1-CD-NW (PCMCIA Card Type On-Chip Debug Emulator)		U16647E
CA850 Ver. 2.50 C Compiler Package	Operation	U16053E
	C Language	U16054E
	Assembly Language	U16042E
PM plus Ver. 5.10		U16569E
ID850 Ver. 2.50 Integrated Debugger	Operation	U16217E
ID850NWC Ver. 2.51 Integrated Debugger	Operation	U16525E
SM850 Ver. 2.40 System Simulator	Operation	U15182E
SM850 Ver. 2.00 or Later System Simulator	External Part User Open Interface Specification	U14873E
RX850 Ver. 3.13 or Later Real-Time OS	Basics	U13430E
	Installation	U13410E
	Technical	U13431E
RX850 Pro Ver. 3.15 Real-Time OS	Basics	U13773E
	Installation	U13774E
	Technical	U13772E
RD850 Ver. 3.01 Task Debugger		U13737E
RD850 Pro Ver. 3.01 Task Debugger		U13916E
AZ850 Ver. 3.10 System Performance Analyzer		U14410E
PG-FP4 Flash Memory Programmer		U15260E

CONTENTS

CHAPTER 1 INTRODUCTION 18

1.1 General 18
1.2 Features 19
1.3 Applications 20
1.4 Ordering Information 20
1.5 Pin Configuration (Top View) 21
1.6 Function Block Configuration 23
1.6.1 Internal block diagram 23
1.6.2 Internal units 24

CHAPTER 2 PIN FUNCTIONS 27

2.1 List of Pin Functions 27
2.2 Pin States 34
2.3 Pin I/O Circuit Types, I/O Buffer Power Supplies and Handling of Unused Pins 35

CHAPTER 3 CPU FUNCTION 38

3.1 Features 38
3.2 CPU Register Set 39
3.2.1 Program register set 40
3.2.2 System register set 41
3.3 Operation Modes 47
★ 3.3.1 Specifying operation mode 47
3.4 Address Space 48
3.4.1 CPU address space 48
3.4.2 Wrap-around of CPU address space 49
3.4.3 Memory map 50
3.4.4 Areas 52
3.4.5 Recommended use of address space 54
3.4.6 Peripheral I/O registers 56
3.4.7 Special registers 65
3.4.8 Notes 69

CHAPTER 4 PORT FUNCTIONS 72

4.1 Features 72
4.2 Basic Port Configuration 72
4.3 Port Configuration 73
4.3.1 Port 0 78
4.3.2 Port 1 82
4.3.3 Port 3 84
4.3.4 Port 4 89
4.3.5 Port 5 92
4.3.6 Port 7 97

4.3.7 Port 9 99
4.3.8 Port CM 105
4.3.9 Port CS 108
4.3.10 Port CT 110
4.3.11 Port DH 113
4.3.12 Port DL 114
4.4 Block Diagrams 117
4.5 Port Register Settings When Alternate Function Is Used 144
4.6 Cautions 150
4.6.1 Cautions on setting port pins 150
4.6.2 Cautions on bit manipulation instruction for port n register (Pn) 151
4.6.3 Cautions on on-chip debug pins 152
4.6.4 Hysteresis characteristics 152

CHAPTER 5 BUS CONTROL FUNCTION 153

5.1 Features 153
5.2 Bus Control Pins 154
5.2.1 Pin status when internal ROM, internal RAM, or on-chip peripheral I/O is accessed 154
5.2.2 Pin status in each operation mode 154
5.3 Memory Block Function 155
5.4 Bus Access 156
5.4.1 Number of clocks for access 156
5.4.2 Bus size setting function 156
5.4.3 Access by bus size 157
5.5 Wait Function 164
5.5.1 Programmable wait function 164
5.5.2 External wait function 165
5.5.3 Relationship between programmable wait and external wait 165
5.5.4 Programmable address wait function 166
5.6 Idle State Insertion Function 167
5.7 Bus Hold Function 168
5.7.1 Functional outline 168
5.7.2 Bus hold procedure 169
5.7.3 Operation in power save mode 169
5.8 Bus Priority 170
5.9 Bus Timing 171

CHAPTER 6 CLOCK GENERATION FUNCTION 174

6.1 Overview 174
6.2 Configuration 175
6.3 Control Registers 177
6.4 Operation 182
6.4.1 Operation of each clock 182
6.4.2 Clock output function 182
6.5 PLL Function 183
6.5.1 Overview 183

6.5.2 Control registers 183
6.5.3 Usage 186

CHAPTER 7 16-BIT TIMER/EVENT COUNTER P (TMP) 187

7.1 Overview 187
7.2 Functions 187
7.3 Configuration 188
7.4 Registers 190
7.5 Operation 201
7.5.1 Interval timer mode (TPnMD2 to TPnMD0 = 000) 202
7.5.2 External event count mode (TPnMD2 to TPnMD0 = 001) 212
7.5.3 External trigger pulse output mode (TPnMD2 to TPnMD0 = 010) 220
7.5.4 One-shot pulse output mode (TPnMD2 to TPnMD0 = 011) 232
7.5.5 PWM output mode (TPnMD2 to TPnMD0 = 100) 239
7.5.6 Free-running timer mode (TPnMD2 to TPnMD0 = 101) 248
7.5.7 Pulse width measurement mode (TPnMD2 to TPnMD0 = 110) 265
7.5.8 Timer output operations 271
7.6 Eliminating Digital Noise on Timer Input Pin 272
7.7 Selector Function 275

CHAPTER 8 16-BIT TIMER/EVENT COUNTER Q (TMQ) 277

8.1 Overview 277
8.2 Functions 277
8.3 Configuration 278
8.4 Registers 281
8.5 Operation 296
8.5.1 Interval timer mode (TQ0MD2 to TQ0MD0 = 000) 297
8.5.2 External event count mode (TQ0MD2 to TQ0MD0 = 001) 306
8.5.3 External trigger pulse output mode (TQ0MD2 to TQ0MD0 = 010) 315
8.5.4 One-shot pulse output mode (TQ0MD2 to TQ0MD0 = 011) 328
8.5.5 PWM output mode (TQ0MD2 to TQ0MD0 = 100) 337
8.5.6 Free-running timer mode (TQ0MD2 to TQ0MD0 = 101) 348
8.5.7 Pulse width measurement mode (TQ0MD2 to TQ0MD0 = 110) 368
8.5.8 Timer output operations 374
8.6 Eliminating Digital Noise on Timer Input Pin 375

CHAPTER 9 16-BIT INTERVAL TIMER M (TMM) 377

9.1 Overview 377
9.2 Configuration 378
9.3 Control Register 379
9.4 Operation 380
9.4.1 Interval timer mode 380
9.4.2 Cautions 384

CHAPTER 10 WATCH TIMER FUNCTIONS **385**

10.1 Functions **385**
10.2 Configuration **386**
10.3 Registers **388**
10.4 Operation **392**
10.4.1 Operation as watch timer 392
10.4.2 Operation as interval timer 393
10.4.3 Cautions 394

CHAPTER 11 FUNCTIONS OF WATCHDOG TIMER 2 ..... **395**

11.1 Functions **395**
11.2 Configuration **396**
11.3 Control Registers **397**
11.4 Operation **400**

CHAPTER 12 A/D CONVERTER **401**

12.1 Overview **401**
12.2 Functions **401**
12.3 Configuration **402**
12.4 Registers **405**
12.5 Operation **414**
12.5.1 Basic operation 414
12.5.2 Trigger mode 416
12.5.3 Operation mode 418
12.5.4 Power-fail compare mode 420
12.6 Cautions **423**
12.7 How to Read A/D Converter Characteristics Table **426**

CHAPTER 13 ASYNCHRONOUS SERIAL INTERFACE A (UARTA) **430**

13.1 Features **430**
13.2 Configuration **431**
13.3 Control Registers **433**
13.4 Interrupt Request Signals **440**
13.5 Operation **441**
13.5.1 Data format 441
13.5.2 SBF transmission/reception format 443
13.5.3 SBF transmission 445
13.5.4 SBF reception 446
13.5.5 UART transmission 447
13.5.6 Continuous transmission procedure 448
13.5.7 UART reception 450
13.5.8 Reception errors 451
13.5.9 Parity types and operations 452
13.5.10 Receive data noise filter 453
13.6 Dedicated Baud Rate Generator **454**

13.7 Cautions **461**

CHAPTER 14 3-WIRE VARIABLE-LENGTH SERIAL I/O (CSIB) **462**

14.1 Mode Switching of CSIB and Other Serial Interfaces **462**
14.1.1 CSIB0 and I²C00 mode switching 462
14.2 Features **462**
14.3 Configuration **463**
14.4 Control Registers **465**
★ **14.5 Interrupt Request Signals** **471**
14.6 Operation **472**
14.6.1 Single transfer (master mode, transmission/reception mode) 472
14.6.2 Single transfer mode (master mode, reception mode) 473
14.6.3 Continuous mode (master mode, transmission/reception mode) 474
14.6.4 Continuous mode (master mode, reception mode) 475
14.6.5 Continuous reception mode (error) 476
14.6.6 Continuous mode (slave mode, transmission/reception mode) 477
14.6.7 Continuous mode (slave mode, reception mode) 478
14.6.8 Clock timing 479
14.7 Output Pins **481**
14.8 Operation Flow **482**
14.9 Baud Rate Generator **488**
14.9.1 Baud rate generation 489

CHAPTER 15 I²C BUS **490**

15.1 Configuration **491**
15.2 I²C Control Registers **495**
15.3 I²C Bus Mode Functions **509**
15.3.1 Pin configuration 509
15.4 I²C Bus Definitions and Control Methods **510**
15.4.1 Start condition 510
15.4.2 Address 511
15.4.3 Transfer direction specification 511
15.4.4 Acknowledge signal ($\overline{\text{ACK}}$) 512
15.4.5 Stop condition 513
15.4.6 Wait signal ($\overline{\text{WAIT}}$) 514
15.5 I²C Interrupt Request Signal (INTIIC0) **516**
15.5.1 Master device operation 516
15.5.2 Slave device operation (when receiving slave address data (matches with SVA0)) 519
15.5.3 Slave device operation (when receiving extension code) 523
15.5.4 Operation without communication 527
15.5.5 Arbitration loss operation (operation as slave after arbitration loss) 527
15.5.6 Operation when arbitration loss occurs (no communication after arbitration loss) 529
15.6 Interrupt Request Signal (INTIIC0) Generation Timing and Wait Control **534**
15.7 Address Match Detection Method **536**
15.8 Error Detection **536**
15.9 Extension Code **536**

15.10 Arbitration **537**
15.11 Wakeup Function **538**
15.12 Communication Reservation **539**
15.12.1 When communication reservation function is enabled (IICF0.IICRSV0 bit = 0) 539
15.12.2 When communication reservation function is disabled (IICF0.IICRSV0 bit = 1) 542
15.13 Cautions **543**
15.14 Communication Operations **544**
15.14.1 Master operation 1 544
15.14.2 Master operation 2 545
15.14.3 Slave operation 546
15.15 Timing of Data Communication **549**

CHAPTER 16 DMA FUNCTION (DMA CONTROLLER) **556**

16.1 Features **556**
16.2 Configuration **557**
16.3 Control Registers **558**
16.4 Transfer Targets **565**
16.5 Transfer Modes **565**
16.6 Transfer Types **566**
16.7 DMA Channel Priorities **567**
16.8 Time Related to DMA Transfer **567**
16.9 DMA Transfer Start Factors **568**
16.10 DMA Abort Factors **569**
16.11 End of DMA Transfer **569**
16.12 Operation Timing **569**
16.13 Cautions **574**

CHAPTER 17 INTERRUPT/EXCEPTION PROCESSING FUNCTION **579**

17.1 Features **579**
17.2 Non-Maskable Interrupts **583**
17.2.1 Operation 586
17.2.2 Restore 587
17.2.3 NP flag 588
17.3 Maskable Interrupts **589**
17.3.1 Operation 589
17.3.2 Restore 591
17.3.3 Priorities of maskable interrupts 592
17.3.4 Interrupt control register (xxICn) 596
17.3.5 Interrupt mask registers 0 to 3 (IMR0 to IMR3) 599
17.3.6 In-service priority register (ISPR) 600
17.3.7 ID flag 600
17.4 External Interrupt Request Input Pins (NMI and INTP0 to INTP7) **601**
17.4.1 Noise elimination 601
17.4.2 Edge detection 601
17.5 Software Exception **607**
17.5.1 Operation 607

17.5.2 Restore 608
17.5.3 EP flag 609
17.6 Exception Trap 610
17.6.1 Illegal opcode definition 610
17.6.2 Debug trap 612
17.7 Interrupt Response Time of CPU 614
17.8 Periods in Which Interrupts Are Not Acknowledged by CPU 615
★ **17.9 Cautions 615**

CHAPTER 18 KEY INTERRUPT FUNCTION 616

18.1 Function 616
18.2 Control Register 617

CHAPTER 19 STANDBY FUNCTION 618

19.1 Overview 618
19.2 Control Registers 621
19.3 HALT Mode 624
19.3.1 Setting and operation status 624
19.3.2 Releasing HALT mode 624
19.4 IDLE1 Mode 626
19.4.1 Setting and operation status 626
19.4.2 Releasing IDLE1 mode 626
19.5 IDLE2 Mode 628
19.5.1 Setting and operation status 628
19.5.2 Releasing IDLE2 mode 628
19.5.3 Securing setup time when releasing IDLE2 mode 630
19.6 STOP Mode 631
19.6.1 Setting and operation status 631
19.6.2 Releasing STOP mode 631
19.7 Securing Oscillation Stabilization Time 633
19.8 Subclock Operation Mode 634
19.8.1 Setting and operation status 634
19.8.2 Releasing subclock operation mode 634
19.9 Sub-IDLE Mode 636
19.9.1 Setting and operation status 636
19.9.2 Releasing sub-IDLE mode 636

CHAPTER 20 RESET FUNCTIONS 638

20.1 Overview 638
20.2 Registers to Check Reset Source 638
20.3 Operation 640
20.3.1 Reset operation by $\overline{\text{RESET}}$ pin 640
20.3.2 Reset operation by WDT2RES signal 642
20.3.3 Operation by low-voltage detector 644
20.3.4 Clock monitor 649

CHAPTER 21 REGULATOR **652**

21.1 Outline **652**
21.2 Operation **653**

CHAPTER 22 ROM CORRECTION FUNCTION **654**

22.1 Overview **654**
22.2 Control Registers **655**
22.3 ROM Correction Operation and Program Flow **656**

CHAPTER 23 FLASH MEMORY **658**

23.1 Features **658**
23.2 Memory Configuration **659**
23.3 Functional Outline **660**
23.4 Rewriting by Dedicated Flash Programmer **662**
23.4.1 Programming environment 662
23.4.2 Communication mode 663
23.4.3 Flash memory control 668
23.4.4 Selection of communication mode 669
23.4.5 Communication commands 670
23.4.6 Pin connection 671
23.5 Rewriting by Self Programming **676**
23.5.1 Overview 676
23.5.2 Features 677
23.5.3 Standard self programming flow 678
23.5.4 Flash functions 679
23.5.5 Pin processing 679
23.5.6 Internal resources used 680

★ **CHAPTER 24 ON-CHIP DEBUG FUNCTION** **681**

24.1 Features **681**
24.2 Connection Circuit Example **682**
24.3 Interface Signals **682**
24.4 Control Register **684**
24.5 Operation **686**
24.6 ROM Security Function **687**
24.6.1 Security ID 687
24.6.2 Setting 688
24.7 Cautions **689**

★ **CHAPTER 25 ELECTRICAL SPECIFICATIONS** **690**

★ **CHAPTER 26 PACKAGE DRAWING** **715**

★ APPENDIX A REGISTER INDEX 716

★ APPENDIX B INSTRUCTION SET LIST 724

B.1 Conventions 724
B.2 Instruction Set (in Alphabetical Order) 727

★ APPENDIX C REVISION HISTORY 734

C.1 Major Revisions in This Edition 734

CHAPTER 1 INTRODUCTION

The μPD703229Y and 70F3229Y are products in the NEC Electronics V850 Series of single-chip microcontrollers designed for low-power operation for real-time control applications.

1.1 General

The μPD703229Y and 70F3229Y are 32-bit single-chip microcontrollers that include the V850ES CPU core and peripheral functions such as ROM/RAM, a timer/counter, serial interfaces, and an A/D converter.

In addition to high real-time response characteristics and 1-clock-pitch basic instructions, the μPD703229Y and 70F3229Y feature multiply instructions, saturated operation instructions, bit manipulation instructions, etc., realized by a hardware multiplier, as optimum instructions for digital servo control applications. Moreover, as a real-time control system, the μPD703229Y and 70F3229Y enable extremely high cost-performance for applications that require a low power consumption, such as audio and car audio.

1.2 Features

- Minimum instruction execution time: 50 ns (operating with main clock (f_{XX}) of 20 MHz)
- General-purpose registers: 32 bits × 32 registers
- CPU features:
  - Signed multiplication (16 × 16 → 32): 1 to 2 clocks
  - Signed multiplication (32 × 32 → 64): 1 to 5 clocks
  - Saturated operations (overflow and underflow detection functions included)
  - 32-bit shift instruction: 1 clock
  - Bit manipulation instructions
  - Load/store instructions with long/short format
- Memory space:
  - 64 MB of linear address space (for programs and data)
  - External expansion: Up to 4 MB (including 1 MB used as internal ROM/RAM)
  - Memory block division function: 2 MB and 2 MB/total 2 blocks
  - • Internal memory
    - μPD703229Y (mask ROM: 384 KB/RAM: 32 KB)
    - μPD70F3229Y (flash memory: 384 KB/RAM: 32 KB)
  - • External bus interface
    - Multiplexed bus output
    - 8-/16-bit data bus sizing function
    - Wait function
      - • Programmable wait function
      - • External wait function
    - Idle state function
    - Bus hold function
- Interrupts and exceptions
  - Non-maskable interrupts: 2 sources
  - Maskable interrupts: 45 sources
    - 8 external sources, 37 internal sources
  - Software exceptions: 32 sources
  - Exception trap: 2 sources
- I/O lines
  - I/O ports: 84
- Timer function
  - 16-bit interval timer M (TMM): 1 channel
  - 16-bit timer/event counter P (TMP): 4 channels
  - 16-bit timer/event counter Q (TMQ): 1 channel
  - Watch timer: 1 channel
  - Watchdog timer: 1 channel
- Serial interface
  - Asynchronous serial interface A (UARTA)
  - 3-wire variable-length serial interface B (CSIB)
  - I^2C bus interface (I^2C)
    - UARTA: 3 channels
    - CSIB: 1 channel
    - CSIB/I^2C: 1 channel
- A/D converter
  - 10-bit resolution: 12 channels
- DMA controller:
  - 4 channels
- ROM correction:
  - 4 correction addresses specifiable
- Clock generator
  - During main system clock or subsystem clock operation
  - 7-level CPU clock (f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, f_{XT})
- Ring-OSC:
  - 200 kHz (TYP.)
- Power-save functions:
  - HALT/IDLE1/IDLE2/STOP mode, subclock/sub-IDLE mode
- Package
  - 100-pin plastic LQFP (fine pitch) (14 × 14)

1.3 Applications

Audio, car audio, consumer devices

1.4 Ordering Information

Part Number	Package	Internal ROM
μPD703229YGC-×××-8EA	100-pin plastic LQFP (fine pitch) (14 × 14)	Mask ROM (384 KB)
μPD70F3229YGC-8EA	100-pin plastic LQFP (fine pitch) (14 × 14)	Flash memory (384 KB)

Remark ××× indicates ROM code suffix.

1.5 Pin Configuration (Top View)

100-pin plastic LQFP (fine pitch) (14 × 14)

μPD703229YGC-×××-8EA

μPD70F3229YGC-8EA

Notes 1. IC (μPD703229Y): Connect to VSS.

FLMD0, FLMD1 (μPD70F3229Y): Connect to VSS in the normal operation mode.

★ **2.** Connect the REGC pin to VSS via a 4.7 μF capacitor.

3. μPD70F3229Y only

Pin names

A16, A17:	Address bus	PCM0 to PCM3:	Port CM
AD0 to AD15:	Address/data bus	PCS0, PCS1:	Port CS
ADTRG:	A/D trigger input	PCT0, PCT1,	
ANI0 to ANI11:	Analog input	PCT4, PCT6:	Port CT
ASCKA0:	Asynchronous serial clock	PDH0, PDH1:	Port DH
ASTB:	Address strobe	PDL0 to PDL15:	Port DL
AV_{REF0}:	Analog reference voltage	$\overline{RD}$:	Read strobe
AV_{SS}:	Analog V_{SS}	REGC:	Regulator control
BV_{DD}:	Power supply for bus interface	$\overline{RESET}$:	Reset
BV_{SS}:	Ground for bus interface	RXDA0 to RXDA2:	Receive data
CLKOUT:	Clock output	$\overline{SCKB0}$, $\overline{SCKB1}$:	Serial clock
$\overline{CS0}$, $\overline{CS1}$:	Chip select	SCL00:	Serial clock
DCK:	Debug clock	SDA00:	Serial data
DDI:	Debug data input	SIB0, SIB1:	Serial input
DDO:	Debug data output	SOB0, SOB1:	Serial output
DMS:	Debug mode select	TIP00, TIP01,	
$\overline{DRST}$:	Debug reset	TIP10, TIP11,	
EV_{DD}:	Power supply for port	TIP20, TIP21,	
EV_{SS}:	Ground for port	TIP30, TIP31,	
FLMD0, FLMD1:	Flash programming mode	TIQ00 to TIQ03:	Timer input
$\overline{HLDAK}$:	Hold acknowledge	TOP00, TOP01,	
$\overline{HLDRQ}$:	Hold request	TOP10, TOP11,	
IC:	Internally connected	TOP20, TOP21,	
INTP0 to INTP7:	External interrupt input	TOP30, TOP31,	
KR0 to KR7:	Key return	TOQ00 to TOQ03:	Timer output
NMI:	Non-maskable interrupt request	TXDA0 to TXDA2:	Transmit data
P00 to P06:	Port 0	V_{DD}:	Power supply
P10, P11:	Port 1	V_{SS}:	Ground
P30 to P39:	Port 3	$\overline{WAIT}$:	Wait
P40 to P42:	Port 4	$\overline{WR0}$:	Lower byte write strobe
P50 to P55:	Port 5	$\overline{WR1}$:	Upper byte write strobe
P70 to P711:	Port 7	X1, X2:	Crystal for main clock
P90 to P915:	Port 9	XT1, XT2:	Crystal for subclock

1.6 Function Block Configuration

1.6.1 Internal block diagram

- μPD703229Y, 70F3229Y

Notes 1. μPD703229Y: 384 KB (mask ROM)
μPD70F3229Y: 384 KB (flash memory)

2. IC: μPD703229Y only
FLMD0, FLMD1: μPD70F3229Y only

3. μPD70F3229Y only

1.6.2 Internal units

(1) CPU

The CPU uses five-stage pipeline control to enable single-clock execution of address calculations, arithmetic logic operations, data transfers, and almost all other instruction processing.

Other dedicated on-chip hardware, such as a multiplier (16 bits × 16 bits → 32 bits) and a barrel shifter (32 bits) contribute to faster complex processing.

(2) Bus control unit (BCU)

The BCU starts a required external bus cycle based on the physical address obtained by the CPU. When an instruction is fetched from external memory space and the CPU does not send a bus cycle start request, the BCU generates a prefetch address and prefetches the instruction code. The prefetched instruction code is stored in an instruction queue.

(3) ROM

This consists of a 384 KB mask ROM or flash memory mapped to addresses 0000000H to 005FFFFH. ROM can be accessed by the CPU in one clock during instruction fetch.

(4) RAM

This consists of a 32 KB RAM mapped to addresses 3FF7000H to 3FFEFFFH. RAM can be accessed by the CPU in one clock during data access.

(5) Interrupt controller (INTC)

This controller handles hardware interrupt requests (NMI, INTP0 to INTP7) from on-chip peripheral hardware and external hardware. Eight levels of interrupt priorities can be specified for these interrupt requests, and multiplexed servicing control can be performed.

(6) Clock generator (CG)

A main clock oscillator and subclock oscillator are provided and generate the main clock oscillation frequency (f_X) and subclock frequency (f_{XT}), respectively. There are two modes: In the clock-through mode, f_X is used as the main clock frequency (f_{XX}) as is. In the PLL mode, f_X is used multiplied by 4.

The CPU clock frequency (f_{CPU}) can be selected from among f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, and f_{XT}.

(7) Ring-OSC

A Ring-OSC oscillator is provided on chip. The oscillation frequency is 200 kHz (TYP.). This Ring-OSC oscillator supplies the clock for the watchdog timer 2 and timer M.

(8) Timer/counter

A four-channel 16-bit timer/event counter P (TMP), a one-channel 16-bit timer/event counter Q (TMQ), and a one-channel 16-bit interval timer M (TMM), are provided on chip.

(9) Watch timer

This timer counts the reference time period (0.5 s) for counting the clock (the 32.768 kHz subclock or the 32.768 kHz f_{BRG} from prescaler 3). The watch timer can also be used as an interval timer for the main clock.

(10) Watchdog timer 2

A watchdog timer is provided on chip to detect inadvertent program loops, system abnormalities, etc.
Either the Ring-OSC clock or the main clock can be selected as the source clock.
Watchdog timer 2 generates a non-maskable interrupt request signal (INTWDT2) or a system reset signal (WDT2RES) after an overflow occurs.

(11) Serial interface (SIO)

Three kinds of serial interfaces: asynchronous serial interface A (UARTA), 3-wire variable-length serial interface B (CSIB), and I^2C bus interface (I^2C) are provided on chip.
In the case of UARTA, data is transferred via the TXDA0 to TXDA2 pins and RXDA0 to RXDA2 pins.
In the case of CSIB, data is transferred via the SOB0 and SOB1 pins, SIB0 and SIB1 pins, and $\overline{\text{SCKB0}}$ and $\overline{\text{SCKB1}}$ pins.
In the case of I^2C, data is transferred via the SDA00 and SCL00 pins.

(12) A/D converter

This 10-bit A/D converter includes 12 analog input pins. Conversion is performed using the successive approximation method.

(13) DMA controller

A four-channel DMA controller is provided on chip. This controller transfers data between the internal RAM and on-chip peripheral I/O devices in response to interrupt requests sent by on-chip peripheral I/O.

(14) ROM correction

A ROM correction function that replaces part of a program in the mask ROM with a program in the internal RAM is provided. Up to four correction addresses can be specified.

(15) Key interrupt function

Key interrupt request signals (INTKR) can be generated by inputting a falling edge to key input pins (8 channels).

(16) On-chip debug function (μPD70F3229Y only)

An on-chip debug function via an N-Wire-type emulator that uses the JTAG (Joint Test Action Group) communication specifications is provided. Switching between the normal port function and on-chip debug function is done with the control pin input level and the on-chip debug mode setting register (OCDM).

(17) Port

There are general-purpose port functions and control pin functions, as listed below.

Port	I/O	Alternate Function
P0	7-bit I/O	NMI, external interrupt, A/D converter trigger, debug reset, timer I/O
P1	2-bit I/O	–
P3	10-bit I/O	External interrupt, serial interface, timer I/O
P4	3-bit I/O	Serial interface
P5	6-bit I/O	Timer I/O, key interrupt input, debug I/O
P7	12-bit I/O	A/D converter analog input
P9	16-bit I/O	Serial interface, key interrupt input, timer I/O, external interrupt
PCM	4-bit I/O	External control signal
PCS	2-bit I/O	Chip select output
PCT	4-bit I/O	External control signal
PDH	2-bit I/O	External address bus
PDL	16-bit I/O	External address/data bus

CHAPTER 2 PIN FUNCTIONS

2.1 List of Pin Functions

The names and functions of the pins of the μPD703229Y and 70F3229Y are described below.

There are three types of pin I/O buffer power supplies: AV_{REF0}, BV_{DD}, and EV_{DD}. The relationship between these power supplies and the pins is described below.

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

Power Supply	Corresponding Pin
AV_{REF0}	Port 7
BV_{DD}	Ports CM, CS, CT, DH, DL
EV_{DD}	$\overline{RESET}$, ports 0, 1, 3 to 5, 9

★ **(1) Port pins**

(1/3)

Pin Name	Pin No.	I/O	On-Chip Pull-up Resistor	Function	Alternate Function
P00	6	I/O	Provided	Port 0 7-bit I/O port Input/output can be specified in 1-bit units.	TIP31/TOP31
P01	7				TIP30/TOP30
P02	17				NMI
P03	18				INTP0/ADTRG
P04	19				INTP1
P05	20				INTP2/$\overline{\text{DRST}}$[Note]
P06	21				INTP3
P10	3	I/O	Provided	Port 1 2-bit I/O port Input/output can be specified in 1-bit units.	–
P11	4				–
P30	25	I/O	Provided	Port 3 10-bit I/O port Input/output can be specified in 1-bit units.	TXDA0
P31	26				RXDA0/INTP7
P32	27				ASCKA0/TIP00/TOP00
P33	28				TIP01/TOP01
P34	29				TIP10/TOP10
P35	30				TIP11/TOP11
P36	31				–
P37	32				–
P38	35				TXDA2
P39	36				RXDA2
P40	22	I/O	Provided	Port 4 3-bit I/O port Input/output can be specified in 1-bit units.	SIB0/SDA00
P41	23				SOB0/SCL00
P42	24				$\overline{\text{SCKB0}}$
P50	37	I/O	Provided	Port 5 6-bit I/O port Input/output can be specified in 1-bit units.	KR0/TIQ01/TOQ01
P51	38				KR1/TIQ02/TOQ02
P52	39				KR2/TIQ03/TOQ03/DDI[Note]
P53	40				KR3/TIQ00/TOQ00/DDO[Note]
P54	41				KR4/DCK[Note]
P55	42				KR5/DMS[Note]
P70	100	I/O	None	Port 7 12-bit I/O port Input/output can be specified in 1-bit units.	ANI0
P71	99				ANI1
P72	98				ANI2
P73	97				ANI3
P74	96				ANI4
P75	95				ANI5
P76	94				ANI6
P77	93				ANI7
P78	92				ANI8
P79	91				ANI9
P710	90				ANI10
P711	89				ANI11

Note μPD70F3229Y only

(2/3)

Pin Name	Pin No.	I/O	On-Chip Pull-up Resistor	Function	Alternate Function
P90	43	I/O	Provided	Port 9 16-bit I/O port Input/output can be specified in 1-bit units.	KR6/TXDA1
P91	44				KR7/RXDA1
P92	45				–
P93	46				–
P94	47				–
P95	48				–
P96	49				TIP21/TOP21
P97	50				SIB1/TIP20/TOP20
P98	51				SOB1
P99	52				$\overline{\text{SCKB1}}$
P910	53				–
P911	54				–
P912	55				–
P913	56				INTP4
P914	57				INTP5
P915	58				INTP6
PCM0	61	I/O	None	Port CM 4-bit I/O port Input/output can be specified in 1-bit units.	$\overline{\text{WAIT}}$
PCM1	62				CLKOUT
PCM2	63				$\overline{\text{HLDAK}}$
PCM3	64				$\overline{\text{HLDRQ}}$
PCS0	59	I/O	None	Port CS 2-bit I/O port Input/output can be specified in 1-bit units.	$\overline{\text{CS0}}$
PCS1	60				$\overline{\text{CS1}}$
PCT0	65	I/O	None	Port CT 4-bit I/O port Input/output can be specified in 1-bit units.	$\overline{\text{WR0}}$
PCT1	66				$\overline{\text{WR1}}$
PCT4	67				$\overline{\text{RD}}$
PCT6	68				ASTB
PDH0	87	I/O	None	Port DH 2-bit I/O port Input/output can be specified in 1-bit units.	A16
PDH1	88				A17

(3/3)

Pin Name	Pin No.	I/O	On-Chip Pull-up Resistor	Function	Alternate Function
PDL0	71	I/O	None	Port DL 16-bit I/O port Input/output can be specified in 1-bit units.	AD0
PDL1	72				AD1
PDL2	73				AD2
PDL3	74				AD3
PDL4	75				AD4
PDL5	76				AD5/FLMD1[Note]
PDL6	77				AD6
PDL7	78				AD7
PDL8	79				AD8
PDL9	80				AD9
PDL10	81				AD10
PDL11	82				AD11
PDL12	83				AD12
PDL13	84				AD13
PDL14	85				AD14
PDL15	86				AD15

Note μPD70F3229Y only

★ **(2) Non-port pins**

(1/3)

Pin Name	Pin No.	I/O	On-Chip Pull-up Resistor	Function	Alternate Function
A16	87	I/O	None	Address bus for external memory	PDH0
A17	88				PDH1
AD0	71	I/O	None	Address bus/data bus for external memory	PDL0
AD1	72				PDL1
AD2	73				PDL2
AD3	74				PDL3
AD4	75				PDL4
AD5	76				PDL5/FLMD1[Note]
AD6	77				PDL6
AD7	78				PDL7
AD8	79				PDL8
AD9	80				PDL9
AD10	81				PDL10
AD11	82				PDL11
AD12	83				PDL12
AD13	84				PDL13
AD14	85				PDL14
AD15	86				PDL15
ADTRG	18	Input	None	A/D converter external trigger input	P03/INTP0
ANI0	100	Input	None	Analog voltage input for A/D converter	P70
ANI1	99				P71
ANI2	98				P72
ANI3	97				P73
ANI4	96				P74
ANI5	95				P75
ANI6	94				P76
ANI7	93				P77
ANI8	92				P78
ANI9	91				P79
ANI10	90				P710
ANI11	89				P711
ASCKA0	27	Input	Provided	UARTA0 baud rate clock input	P32/TIP00/TOP00
ASTB	68	Output	None	Address strobe signal output for external memory	PCT6
AV_{REF0}	1	–	–	Reference voltage input for A/D converter/positive power supply for port 7	–
AV_{SS}	2	–	–	Ground potential for A/D converter (same potential as V_{SS})	–
BV_{DD}	70	–	–	Positive power supply for bus interface and alternate-function ports	–
BV_{SS}	69	–	–	Ground potential for bus interface and alternate-function ports	–
CLKOUT	62	Output	None	Internal system clock output	PCM1
$\overline{CS0}$	59	Output	None	Chip select output	PCS0
$\overline{CS1}$	60				PCS1

Note μPD70F3229Y only

(2/3)

Pin Name	Pin No.	I/O	On-Chip Pull-up Resistor	Function	Alternate Function
DCK[Note 1]	41	Input	Provided	Debug clock input	P54/KR4
DDI[Note 1]	39	Input	Provided	Debug data input	P52/KR2/TIQ03/TOQ03
DDO[Note 1]	40	Output	Provided	Debug data output	P53/KR3/TIQ00/TOQ00
DMS[Note 1]	42	Input	Provided	Debug mode select input	P55/KR5
$\overline{\text{DRST}}$[Note 1]	20	Input	Provided	Debug reset input	P05/INTP2
EV_{DD}	5, 34	–	–	Positive power supply for external (same potential as V_{DD})	–
EV_{SS}	33	–	–	Ground potential for external (same potential as V_{SS})	–
FLMD0[Note 1]	8	Input	None	Flash memory programming mode setting pin	–
FLDM1[Note 1]	76				PDL5/AD5
$\overline{\text{HLDAK}}$	63	Output	None	Bus hold acknowledge output	PCM2
$\overline{\text{HLDRQ}}$	64	Input	None	Bus hold request input	PCM3
IC[Note 2]	8	–	–	Internally connected	–
INTP0	18	Input	Provided	External interrupt request input (maskable, analog noise elimination). Analog noise elimination or digital noise elimination selectable for INTP3 pin.	P03/ADTRG
INTP1	19				P04
INTP2	20				P05/$\overline{\text{DRST}}$[Note 1]
INTP3	21				P06
INTP4	56				P913
INTP5	57				P914
INTP6	58				P915
INTP7	26				P31/RXDA0
KR0	37	Input	Provided	Key interrupt input	P50/TIQ01/TOQ01
KR1	38				P51/TIQ02/TOQ02
KR2	39				P52/TIQ03/TOQ03/DDI[Note 1]
KR3	40				P53/TIQ00/TOQ00/DDO[Note 1]
KR4	41				P54/DCK[Note 1]
KR5	42				P55/DMS[Note 1]
KR6	43				P90/TXDA1
KR7	44				P91/RXDA1
NMI	17	Input	Provided	External interrupt input (non-maskable, analog noise elimination)	P02
$\overline{\text{RD}}$	67	Output	None	Read strobe signal output for external memory	PCT4
REGC	10	–	–	Connection of regulator output stabilization capacitance	–
$\overline{\text{RESET}}$	14	Input	–	System reset input	–
RXDA0	26	Input	Provided	Serial receive data input (UARTA0, UARTA1, UARTA2)	P31/INTP7
RXDA1	44				P91/KR7
RXDA2	36				P39
$\overline{\text{SCKB0}}$	24	I/O	Provided	Serial clock I/O (CSIB0, CSIB1)	P42
$\overline{\text{SCKB1}}$	52				P99
SCL00	23	I/O	Provided	Serial clock I/O (I^2C00)	P41/SOB0
SDA00	22	I/O	Provided	Serial transmit/receive data I/O (I^2C00)	P40/SIB0
SIB0	22	Input	Provided	Serial receive data input (CSIB0, CSIB1)	P40/SDA00
SIB1	50				P97/TIP20/TOP20

Notes 1. μPD70F3229Y only

2. μPD703229Y only

(3/3)

Pin Name	Pin No.	I/O	On-Chip Pull-up Resistor	Function	Alternate Function
SOB0	23	Output	Provided	Serial transmit data output (CSIB0, CSIB1)	P41/SCL00
SOB1	51				P98
TIP00	27	Input	Provided	External event/clock input (TMP0)	P32/ASCKA0/TOP00
TIP01	28				P33/TOP01
TIP10	29			External event/clock input (TMP1)	P34/TOP10
TIP11	30				P35/TOP11
TIP20	50			External event/clock input (TMP2)	P97/SIB1/TOP20
TIP21	49				P96/TOP21
TIP30	7			External event/clock input (TMP3)	P01/TOP30
TIP31	6				P00/TOP31
TIQ00	40	Input	Provided	External event/clock input (TMQ0)	P53/KR3/TOQ00/DDO[Note]
TIQ01	37				P50/KR0/TOQ01
TIQ02	38				P51/KR1/TOQ02
TIQ03	39				P52/KR2/TOQ03/DDI[Note]
TOP00	27	Output	Provided	Timer output (TMP0)	P32/ASCKA0/TIP00
TOP01	28				P33/TIP01
TOP10	29			Timer output (TMP1)	P34/TIP10
TOP11	30				P35/TIP11
TOP20	50			Timer output (TMP2)	P97/SIB1/TIP20
TOP21	49				P96/TIP21
TOP30	7			Timer output (TMP3)	P01/TIP30
TOP31	6				P00/TIP31
TOQ00	40	Output	Provided	Timer output (TMQ0)	P53/KR3/TIQ00/DDO[Note]
TOQ01	37				P50/KR0/TIQ01
TOQ02	38				P51/KR1/TIQ02
TOQ03	39				P52/KR2/TIQ03/DDI[Note]
TXDA0	25	Output	Provided	Serial transmit data output (UARTA0, UARTA1, UARTA2)	P30
TXDA1	43				P90/KR6
TXDA2	35				P38
V_{DD}	9	–	–	Positive power supply for internal	–
V_{SS}	11	–	–	Ground potential for internal	–
$\overline{WAIT}$	61	Input	None	External wait input	PCM0
$\overline{WR0}$	65	Output	None	Write strobe for external memory (lower 8 bits)	PCT0
$\overline{WR1}$	66			Write strobe for external memory (higher 8 bits)	PCT1
X1	12	Input	None	Connection of resonator for main clock	–
X2	13	–	None		–
XT1	15	Input	None	Connection of resonator for subclock	–
XT2	16	–	None		–

Note μPD70F3229Y only

2.2 Pin States

The operation states of pins in each mode are described below.

★ **Table 2-2. Pin Operation States in Each Mode**

Bus Control Pin	Reset	HALT Mode[Note 1]	IDLE Mode STOP Mode	Idle State[Note 2]	Bus Hold
AD0 to AD15	Hi-Z[Note 3]	Undefined	Hi-Z	Held	Hi-Z
A16, A17					
$\overline{\text{WAIT}}$		–	–	–	–
CLKOUT		Operating	L	Operating	Operating
$\overline{\text{CS0}}$, $\overline{\text{CS1}}$		H	H	Held	Hi-Z
$\overline{\text{WR0}}$, $\overline{\text{WR1}}$		Inactive level (H)	H	H	Hi-Z
$\overline{\text{RD}}$		H			
ASTB					
$\overline{\text{HLDAK}}$		Operating			L
$\overline{\text{HLDRQ}}$			–	–	Operating

Notes 1. Even in HALT mode, pins operate during DMA operations.

2. The state of the pins in the idle state inserted following the T3 state is shown.

3. The bus control pins are shared with port pins, so they are initialized to the input mode (port mode).

Remark Hi-Z: High impedance
Held: The state during the immediately preceding external bus cycle is held.
L: Low-level output
H: High-level output
–: Input without sampling (not acknowledged)

2.3 Pin I/O Circuit Types, I/O Buffer Power Supplies and Handling of Unused Pins

(1/2)

Pin	Alternate Function	I/O Circuit Type	Recommended Connection
P00	TIP31/TOP31	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P01	TIP30/TOP30		
P02	NMI		
P03	INTP0/ADTRG		
P04	INTP1		
P05	INTP2/$\overline{\text{DRST}}$[Note]	5-AF	Independently connect to EV_{SS} via a resistor. Output: Leave open.
P06	INTP3	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P10, P11	–	5-A	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P30	TXDA0	5-A	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P31	RXDA0/INTP7	5-W	
P32	ASCKA0/TIP00/TOP00		
P33	TIP01/TOP01		
P34	TIP10/TOP10		
P35	TIP11/TOP11		
P36, P37	–	5-A	
P38	TXDA2		
P39	RXDA2	5-W	
P40	SIB0/SDA00	10-F	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P41	SOB0/SCL00		
P42	$\overline{\text{SCKB0}}$	5-W	
P50	KR0/TIQ01/TOQ01	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P51	KR1/TIQ02/TOQ02		
P52	KR2/TIQ03/TOQ03/DDI[Note]		
P53	KR3/TIQ00/TOQ00/DDO[Note]		
P54	KR4/DCK[Note]		
P55	KR5/DMS[Note]		
P70 to P711	ANI0 to ANI11	11-G	Independently connect to AV_{REF0} or AV_{SS} via a resistor. Output: Leave open.
P90	KR6/TXDA1	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P91	KR7/RXDA1		
P92 to P95	–	5-A	
P96	TIP21/TOP21	5-W	
P97	SIB1/TIP20/TOP20		
P98	SOB1	5-A	
P99	$\overline{\text{SCKB1}}$	5-W	
P910 to P912	–	5-A	
P913 to P915	INTP4 to INTP6	5-W	

Note μPD70F3229Y only

(2/2)

Pin	Alternate Function	I/O Circuit Type	Recommended Connection
PCM0	$\overline{\text{WAIT}}$	5	Input: Independently connect to BV_{DD} or BV_{SS} via a resistor. Output: Leave open.
PCM1	CLKOUT		
PCM2	$\overline{\text{HLDAK}}$		
PCM3	$\overline{\text{HLDRQ}}$		
PCS0, PCS1	$\overline{\text{CS0}}$, $\overline{\text{CS1}}$	5	Input: Independently connect to BV_{DD} or BV_{SS} via a resistor. Output: Leave open.
PCT0, PCT1	$\overline{\text{WR0}}$, $\overline{\text{WR1}}$	5	Input: Independently connect to BV_{DD} or BV_{SS} via a resistor. Output: Leave open.
PCT4	$\overline{\text{RD}}$		
PCT6	ASTB		
PDH0, PDH1	A16, A17	5	Input: Independently connect to BV_{DD} or BV_{SS} via a resistor. Output: Leave open.
PDL0 to PDL4	AD0 to AD4	5	Input: Independently connect to BV_{DD} or BV_{SS} via a resistor. Output: Leave open.
PDL5	AD5/FLMD1[Note 1]		Independently connect to BV_{SS} via a resistor. Output: Leave open.
PDL6 to PDL15	AD6 to AD15		Input: Independently connect to BV_{DD} or BV_{SS} via a resistor. Output: Leave open.
AV_{REF0}	–	–	Always connect to power supply (even in the standby mode).
AV_{SS}	–	–	Always connect directly to GND (even in the standby mode).
BV_{DD}	–		Always connect to power supply (even in the standby mode).
BV_{SS}	–		Always connect directly to GND (even in the standby mode).
EV_{DD}	–	–	Always connect to power supply (even in the standby mode).
EV_{SS}	–	–	Always connect directly to GND (even in the standby mode).
FLMD0[Note 1]	–	–	Directly connect to V_{SS} in a mode other than the flash memory programming mode.
IC[Note 2]	–	–	Directly connect to V_{SS}.
REGC	–	–	Connect regulator output stabilization capacitance.
$\overline{\text{RESET}}$	–	2	–
V_{DD}	–	–	Always connect to power supply (even in the standby mode).
V_{SS}	–	–	Always connect directly to GND (even in the standby mode).
X1	–	–	–
X2	–	–	–
XT1	–	16	Connect to V_{SS}.
XT2	–	16	Leave open.

Notes 1. μPD70F3229Y only

2. μPD703229Y only

Figure 2-1. Pin I/O Circuits

Note Hysteresis characteristics are not available in port mode.

CHAPTER 3 CPU FUNCTION

The CPU of the μPD703229Y and 70F3229Y is based on RISC architecture and executes almost all instructions in one clock by using a 5-stage pipeline.

3.1 Features

○ Minimum instruction execution time: 50 ns (at 20 MHz operation)
30.5 μs (with subclock (f_{XT} = 32.768 kHz operation))

○ Memory space Program (physical address) space: 64 MB linear
Data (logical address) space: 4 GB linear
Memory block division function: 2 MB and 2 MB/total 2 blocks

○ General-purpose registers: 32 bits × 32 registers

○ Internal 32-bit architecture

○ 5-stage pipeline control

○ Multiplication/division instruction

○ Saturation operation instruction

○ 32-bit shift instruction: 1 clock

○ Load/store instruction with long/short format

○ Four types of bit manipulation instructions
- SET1
- CLR1
- NOT1
- TST1

3.2 CPU Register Set

The registers of the μPD703229Y and 70F3229Y can be classified into two types: general-purpose program registers and dedicated system registers. All the registers are 32 bits wide.

For details, refer to the **V850ES Architecture User's Manual**.

(1) Program register set

31 — 0

r0	(Zero register)
r1	(Assembler-reserved register)
r2	
r3	(Stack pointer (SP))
r4	(Global pointer (GP))
r5	(Text pointer (TP))
r6	
r7	
r8	
r9	
r10	
r11	
r12	
r13	
r14	
r15	
r16	
r17	
r18	
r19	
r20	
r21	
r22	
r23	
r24	
r25	
r26	
r27	
r28	
r29	
r30	(Element pointer (EP))
r31	(Link pointer (LP))

31 — 0

PC	(Program counter)

(2) System register set

31 — 0

EIPC	(Interrupt status saving register)
EIPSW	(Interrupt status saving register)

FEPC	(NMI status saving register)
FEPSW	(NMI status saving register)

ECR	(Interrupt source register)

PSW	(Program status word)

CTPC	(CALLT execution status saving register)
CTPSW	(CALLT execution status saving register)

DBPC	(Exception/debug trap status saving register)
DBPSW	(Exception/debug trap status saving register)

CTBP	(CALLT base pointer)

3.2.1 Program register set

The program registers include general-purpose registers and a program counter.

(1) General-purpose registers (r0 to r31)

Thirty-two general-purpose registers, r0 to r31, are available. Any of these registers can be used to store a data variable or an address variable.

However, r0 and r30 are implicitly used by instructions and care must be exercised when these registers are used. r0 always holds 0 and is used for an operation that uses 0 or addressing of offset 0. r30 is used by the SLD and SST instructions as a base pointer when these instructions access the memory. r1, r3 to r5, and r31 are implicitly used by the assembler and C compiler. When using these registers, save their contents for protection, and then restore the contents after using the registers. r2 is sometimes used by the real-time OS. If the real-time OS does not use r2, it can be used as a register for variables.

Table 3-1. Program Registers

Name	Usage	Operation
r0	Zero register	Always holds 0.
r1	Assembler-reserved register	Used as working register to create 32-bit immediate data
r2	Register for address/data variable (if real-time OS does not use r2)	
r3	Stack pointer	Used to create a stack frame when a function is called
r4	Global pointer	Used to access a global variable in the data area
r5	Text pointer	Used as register that indicates the beginning of a text area (area where program codes are located)
r6 to r29	Register for address/data variable	
r30	Element pointer	Used as base pointer to access memory
r31	Link pointer	Used when the compiler calls a function
PC	Program counter	Holds the instruction address during program execution

(2) Program counter (PC)

The program counter holds the instruction address during program execution. The lower 26 bits of this register are valid. Bits 31 to 26 are fixed to 0. A carry from bit 25 to 26 is ignored even if it occurs.

Bit 0 is fixed to 0. This means that execution cannot branch to an odd address.

3.2.2 System register set

The system registers control the status of the CPU and hold interrupt information.

These registers can be read or written by using system register load/store instructions (LDSR and STSR), using the system register numbers listed below.

Table 3-2. System Register Numbers

System Register Number	System Register Name	Operand Specification	
		LDSR Instruction	STSR Instruction
0	Interrupt status saving register (EIPC)[Note 1]	√	√
1	Interrupt status saving register (EIPSW)[Note 1]	√	√
2	NMI status saving register (FEPC)[Note 1]	√	√
3	NMI status saving register (FEPSW)[Note 1]	√	√
4	Interrupt source register (ECR)	×	√
5	Program status word (PSW)	√	√
6 to 15	Reserved for future function expansion (operation is not guaranteed if these registers are accessed)	×	×
16	CALLT execution status saving register (CTPC)	√	√
17	CALLT execution status saving register (CTPSW)	√	√
18	Exception/debug trap status saving register (DBPC)	√[Note 2]	√
19	Exception/debug trap status saving register (DBPSW)	√[Note 2]	√
20	CALLT base pointer (CTBP)	√	√
21 to 31	Reserved for future function expansion (operation is not guaranteed if these registers are accessed)	×	×

Notes 1. Because only one set of these registers is available, the contents of these registers must be saved by program if multiple interrupts are enabled.

2. These registers can be accessed only after DBTRAP instruction execution and before DBRET instruction execution.

Caution **Even if EIPC or FEPC, or bit 0 of CTPC is set to 1 by the LDSR instruction, bit 0 is ignored when execution is returned to the main routine by the RETI instruction after interrupt servicing (this is because bit 0 of the PC is fixed to 0). Set an even value to EIPC, FEPC, and CTPC (bit 0 = 0).**

Remark √: Can be accessed
×: Access prohibited

(1) Interrupt status saving registers (EIPC and EIPSW)

EIPC and EIPSW are used to save the status when an interrupt occurs.

If a software exception or a maskable interrupt occurs, the contents of the program counter (PC) are saved to EIPC, and the contents of the program status word (PSW) are saved to EIPSW (these contents are saved to the NMI status saving registers (FEPC and FEPSW) if a non-maskable interrupt (NMI) occurs).

The address of the instruction next to the one of the instruction under execution, except some instructions (see **17.8 Periods in Which Interrupts Are Not Acknowledged by CPU**), is saved to EIPC when a software exception or a maskable interrupt occurs.

The current contents of the PSW are saved to EIPSW.

Because only one set of interrupt status saving registers is available, the contents of these registers must be saved by program when multiple interrupts are enabled.

Bits 31 to 26 of EIPC and bits 31 to 8 of EIPSW are reserved for future function expansion (these bits are always fixed to 0).

When the RETI instruction is executed, the values in EIPC and EIPSW are restored to the PC and PSW, respectively.

(2) NMI status saving registers (FEPC and FEPSW)

FEPC and FEPSW are used to save the status when a non-maskable interrupt (NMI) occurs.

If an NMI occurs, the contents of the program counter (PC) are saved to FEPC, and those of the program status word (PSW) are saved to FEPSW.

The address of the instruction next to the one of the instruction under execution, except some instructions, is saved to FEPC when an NMI occurs.

The current contents of the PSW are saved to FEPSW.

Because only one set of NMI status saving registers is available, the contents of these registers must be saved by program when multiple interrupts are enabled.

Bits 31 to 26 of FEPC and bits 31 to 8 of FEPSW are reserved for future function expansion (these bits are always fixed to 0).

When the RETI instruction is executed, the values in FEPC and FEPSW are restored to the PC and PSW, respectively.

(3) Interrupt source register (ECR)

The interrupt source register (ECR) holds the source of an exception or interrupt if an exception or interrupt occurs. This register holds the exception code of each interrupt source. Because this register is a read-only register, data cannot be written to this register using the LDSR instruction.

Bit position	Bit name	Meaning
31 to 16	FECC	Exception code of non-maskable interrupt (NMI)
15 to 0	EICC	Exception code of exception or maskable interrupt

(4) Program status word (PSW)

The program status word (PSW) is a collection of flags that indicate the status of the program (result of instruction execution) and the status of the CPU.

If the contents of a bit of this register are changed by using the LDSR instruction, the new contents are validated immediately after completion of LDSR instruction execution.

Interrupt request acknowledgment is held pending while the PSW write instruction is being executed by the LDSR instruction.

Bits 31 to 8 of this register are reserved for future function expansion (these bits are fixed to 0).

(1/2)

Bit position	Flag name	Meaning
31 to 8	RFU	Reserved field. Fixed to 0.
7	NP	Indicates that a non-maskable interrupt (NMI) is being serviced. This bit is set to 1 when an NMI request is acknowledged, disabling multiple interrupts. 0: NMI is not being serviced. 1: NMI is being serviced.
6	EP	Indicates that an exception is being processed. This bit is set to 1 when an exception occurs. Even if this bit is set, interrupt requests are acknowledged. 0: Exception is not being processed. 1: Exception is being processed.
5	ID	Indicates whether a maskable interrupt can be acknowledged. 0: Interrupt enabled 1: Interrupt disabled
4	SAT[Note]	Indicates that the result of a saturation operation has overflowed and is saturated. Because this is a cumulative flag, it is set to 1 when the result of a saturation operation instruction is saturated, and is not cleared to 0 even if the subsequent operation result is not saturated. Use the LDSR instruction to clear this bit. This flag is neither set to 1 nor cleared to 0 by execution of an arithmetic operation instruction. 0: Not saturated 1: Saturated
3	CY	Indicates whether a carry or a borrow occurs as a result of an operation. 0: Carry or borrow does not occur. 1: Carry or borrow occurs.
2	OV[Note]	Indicates whether an overflow occurs during operation. 0: Overflow does not occur. 1: Overflow occurs.
1	S[Note]	Indicates whether the result of an operation is negative. 0: The result is positive or 0. 1: The result is negative.
0	Z	Indicates whether the result of an operation is 0. 0: The result is not 0. 1: The result is 0.

Remark Also read **Note** on the next page.

(2/2)

Note The result of the operation that has performed saturation processing is determined by the contents of the OV and S flags. The SAT flag is set to 1 only when the OV flag is set to 1 when a saturation operation is performed.

Status of operation result	Flag status			Result of operation of saturation processing
	SAT	OV	S	
Maximum positive value is exceeded.	1	1	0	7FFFFFFFH
Maximum negative value is exceeded.	1	1	1	80000000H
Positive (maximum value is not exceeded)	Holds value before operation	0	0	Operation result itself
Negative (maximum value is not exceeded)			1	

(5) CALLT execution status saving registers (CTPC and CTPSW)

CTPC and CTPSW are CALLT execution status saving registers.

When the CALLT instruction is executed, the contents of the program counter (PC) are saved to CTPC, and those of the program status word (PSW) are saved to CTPSW.

The contents saved to CTPC are the address of the instruction next to CALLT.

The current contents of the PSW are saved to CTPSW.

Bits 31 to 26 of CTPC and bits 31 to 8 of CTPSW are reserved for future function expansion (fixed to 0).

(6) Exception/debug trap status saving registers (DBPC and DBPSW)

DBPC and DBPSW are exception/debug trap status saving registers.

If an exception trap or debug trap occurs, the contents of the program counter (PC) are saved to DBPC, and those of the program status word (PSW) are saved to DBPSW.

The contents to be saved to DBPC are the address of the instruction next to the one that is being executed when an exception trap or debug trap occurs.

The current contents of the PSW are saved to DBPSW.

Bits 31 to 26 of DBPC and bits 31 to 8 of DBPSW are reserved for future function expansion (fixed to 0).

When the DBRET instruction is executed, the values in DBPC and DBPSW are restored to the PC and PSW, respectively.

(7) CALLT base pointer (CTBP)

The CALLT base pointer (CTBP) is used to specify a table address or generate a target address (bit 0 is fixed to 0).

Bits 31 to 26 of this register are reserved for future function expansion (fixed to 0).

3.3 Operation Modes

The μPD703229Y and 70F3229Y have the following operation modes.

(1) Normal operation mode

In this mode, each pin related to the bus interface is set to the port mode after system reset has been released. Execution branches to the reset entry address of the internal ROM, and then instruction processing is started.

(2) Flash memory programming mode (μPD70F3229Y only)

In this mode, the internal flash memory can be programmed by using a flash programmer.

(3) On-chip debug mode (μPD70F3229Y only)

The μPD70F3229Y is provided with an on-chip debug function that employs the JTAG (Joint Test Action Group) communication specifications and that is executed via an N-Wire emulator.

For details, refer to **CHAPTER 24 ON-CHIP DEBUG FUNCTION**.

★ **3.3.1 Specifying operation mode**

Specify the operation mode by using the FLMD0 and FLMD1 pins.

In the normal mode, make sure that the FLMD0/IC pin goes low when reset is released.

In the flash memory programming mode, a high level is input to the FLMD0 pin from the flash programmer if a flash programmer is connected, but it must be input from an external circuit in the self-programming mode.

Operation When Reset Is Released		Operation Mode After Reset
FLMD0	FLMD1	
L	×	Normal operation mode
H	L	Flash memory programming mode
H	H	Setting prohibited

Remark L: Low-level input
H: High-level input
×: Don't care

3.4 Address Space

3.4.1 CPU address space

For addressing instruction addresses, up to 4 MB of external memory area, internal ROM area, and internal RAM area in an area of up to 64 MB of linear address space (program space) is supported. Up to 4 GB of linear address space (data space) is supported for operand addressing (data access). In the 4 GB address space, it seems that there are sixty-four 64 MB physical address spaces. This means that the same 64 MB physical address space is accessed, regardless of the values of bits 31 to 26.

Figure 3-1. Image on Address Space

3.4.2 Wrap-around of CPU address space

(1) Program space

Of the 32 bits of the PC (program counter), the higher 6 bits are fixed to 0 and only the lower 26 bits are valid. The higher 6 bits ignore a carry or borrow from bit 25 to 26 during branch address calculation.

Therefore, the lowest address of the program space, 00000000H, and the highest address, 03FFFFFFH, are contiguous addresses. That the lowest address and the highest address of the program space are contiguous in this way is called wrap-around.

Caution Because the 4 KB area of addresses 03FFF000H to 03FFFFFFH is a peripheral I/O area, instructions cannot be fetched from this area. Therefore, do not execute an operation in which the result of a branch address calculation affects this area.

(2) Data space

The result of an operand address calculation operation that exceeds 32 bits is ignored.

Therefore, the lowest address of the data space, 00000000H, and the highest address, FFFFFFFFH, are contiguous, and wrap-around occurs at the boundary of these addresses.

3.4.3 Memory map

The areas shown below are reserved in the μPD703229Y and 70F3229Y.

Figure 3-2. Data Memory Map (Physical Addresses)

Note This area is an external memory area in the case of a data write access.

Caution A write access to addresses 0000000H to 003FFFFH is the same operation as a write access to addresses 0100000H to 013FFFFH.

Figure 3-3. Program Memory Map

3.4.4 Areas

(1) Internal ROM area

1 MB of addresses 0000000H to 00FFFFFH is reserved as the internal ROM area.

384 KB are mapped to addresses 000000H to 005FFFFH as the physical internal ROM.

Figure 3-4. Internal ROM Area (384 KB)

(2) Internal RAM area

60 KB of addresses 3FF0000H to 3FFEFFFH are reserved as the internal RAM area.

32 KB of addresses 3FF7000H to 3FFEFFFH are mapped in the μPD703229Y and 70F3229Y as physical internal RAM.

Figure 3-5. Internal RAM Area (32 KB)

(3) On-chip peripheral I/O area

4 KB of addresses 3FFF000H to 3FFFFFFH are reserved as the on-chip peripheral I/O area.

Figure 3-6. On-Chip Peripheral I/O Area

Peripheral I/O registers that have functions to specify the operation mode for and monitor the status of the on-chip peripheral I/O are mapped to the on-chip peripheral I/O area. Program cannot be fetched from this area.

Cautions 1. When a register is accessed in word units, a word area is accessed twice in halfword units in the order of lower area and higher area, with the lower 2 bits of the address ignored.

2. If a register that can be accessed in byte units is accessed in halfword units, the higher 8 bits are undefined when the register is read, and data is written to the lower 8 bits.

3. Addresses not defined as registers are reserved for future expansion. The operation is undefined and not guaranteed when these addresses are accessed.

(4) External memory area

3 MB (0100000H to 03FFFFFH) are allocated as the external memory area. For details, refer to **CHAPTER 5 BUS CONTROL FUNCTION**.

3.4.5 Recommended use of address space

The architecture of the μPD703229Y and 70F3229Y requires that a register that serves as a pointer be secured for address generation when operand data in the data space is accessed. The address stored in this pointer ±32 KB can be directly accessed by an instruction for operand data. Because the number of general-purpose registers that can be used as a pointer is limited, however, by keeping the performance from dropping during address calculation when a pointer value is changed, as many general-purpose registers as possible can be secured for variables, and the program size can be reduced.

(1) Program space

Of the 32 bits of the PC (program counter), the higher 6 bits are fixed to 0, and only the lower 26 bits are valid. Regarding the program space, therefore, a 64 MB space of contiguous addresses starting from 00000000H unconditionally corresponds to the memory map.

To use the internal RAM area as the program space, access addresses 3FF7000H to 3FFEFFFH.

★ **Caution If a branch instruction is allocated at the upper boundary of the internal RAM area, a prefetch operation (invalid fetch) spanning the on-chip peripheral I/O area does not occur.**

(2) Data space

With the μPD703229Y and 70F3229Y, it seems that there are sixty-four 64 MB address spaces on the 4 GB CPU address space. Therefore, the least significant bit (bit 25) of a 26-bit address is sign-extended to 32 bits and allocated as an address.

(a) An application example of wrap-around is shown below.

If R = r0 (zero register) is specified for the LD/ST disp16 [R] instruction, a range of addresses 00000000H ±32 KB can be addressed by sign-extended disp16. All the resources, including the internal hardware, can be addressed by one pointer.

The zero register (r0) is a register fixed to 0 by hardware, and practically eliminates the need for registers dedicated to pointers.

Figure 3-7. Recommended Memory Map

Note Access to this area is prohibited. To access the on-chip peripheral I/O in this area, specify addresses FFFF000H to FFFFFFFH.

Remark ↕ indicates the recommended area.

3.4.6 Peripheral I/O registers

(1/9)

	Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
					1	8	16	
★	FFFFF004H	Port DL register	PDL	R/W			√	0000H[Note]
★	FFFFF004H	Port DL register L	PDLL		√	√		00H[Note]
★	FFFFF005H	Port DL register H	PDLH		√	√		00H[Note]
★	FFFFF006H	Port DH register	PDH		√	√		00H[Note]
★	FFFFF008H	Port CS register	PCS		√	√		00H[Note]
★	FFFFF00AH	Port CT register	PCT		√	√		00H[Note]
★	FFFFF00CH	Port CM register	PCM		√	√		00H[Note]
	FFFFF024H	Port DL mode register	PMDL				√	FFFFH
	FFFFF024H	Port DL mode register L	PMDLL		√	√		FFH
	FFFFF025H	Port DL mode register H	PMDLH		√	√		FFH
	FFFFF026H	Port DH mode register	PMDH		√	√		FFH
	FFFFF028H	Port CS mode register	PMCS		√	√		FFH
	FFFFF02AH	Port CT mode register	PMCT		√	√		FFH
	FFFFF02CH	Port CM mode register	PMCM		√	√		FFH
	FFFFF044H	Port DL mode control register	PMCDL				√	0000H
	FFFFF044H	Port DL mode control register L	PMCDLL		√	√		00H
	FFFFF045H	Port DL mode control register H	PMCDLH		√	√		00H
	FFFFF046H	Port DH mode control register H	PMCDH		√	√		00H
	FFFFF048H	Port CS mode control register	PMCCS		√	√		00H
	FFFFF04AH	Port CT mode control register	PMCCT		√	√		00H
	FFFFF04CH	Port CM mode control register	PMCCM		√	√		00H
	FFFFF066H	Bus size configuration register	BSC				√	5555H
	FFFFF06EH	System wait control register	VSWC			√		77H
	FFFFF080H	DMA source address register 0L	DSA0L				√	Undefined
	FFFFF082H	DMA source address register 0H	DSA0H				√	Undefined
	FFFFF084H	DMA destination address register 0L	DDA0L				√	Undefined
	FFFFF086H	DMA destination address register 0H	DDA0H				√	Undefined
	FFFFF088H	DMA source address register 1L	DSA1L				√	Undefined
	FFFFF08AH	DMA source address register 1H	DSA1H				√	Undefined
	FFFFF08CH	DMA destination address register 1L	DDA1L				√	Undefined
	FFFFF08EH	DMA destination address register 1H	DDA1H				√	Undefined
	FFFFF090H	DMA source address register 2L	DSA2L				√	Undefined
	FFFFF092H	DMA source address register 2H	DSA2H				√	Undefined
	FFFFF094H	DMA destination address register 2L	DDA2L				√	Undefined
	FFFFF096H	DMA destination address register 2H	DDA2H				√	Undefined
	FFFFF098H	DMA source address register 3L	DSA3L				√	Undefined
	FFFFF09AH	DMA source address register 3H	DSA3H				√	Undefined
	FFFFF09CH	DMA destination address register 3L	DDA3L				√	Undefined
	FFFFF09EH	DMA destination address register 3H	DDA3H				√	Undefined
	FFFFF0C0H	DMA transfer count register 0	DBC0				√	Undefined
	FFFFF0C2H	DMA transfer count register 1	DBC1				√	Undefined

★ **Note** The output latch is 00H or 0000H. When these registers are input, the pin statuses are read.

(2/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF0C4H	DMA transfer count register 2	DBC2	R/W			√	Undefined
FFFFF0C6H	DMA transfer count register 3	DBC3				√	Undefined
FFFFF0D0H	DMA addressing control register 0	DADC0				√	0000H
FFFFF0D2H	DMA addressing control register 1	DADC1				√	0000H
FFFFF0D4H	DMA addressing control register 2	DADC2				√	0000H
FFFFF0D6H	DMA addressing control register 3	DADC3				√	0000H
FFFFF0E0H	DMA channel control register 0	DCHC0		√	√		00H
FFFFF0E2H	DMA channel control register 1	DCHC1		√	√		00H
FFFFF0E4H	DMA channel control register 2	DCHC2		√	√		00H
FFFFF0E6H	DMA channel control register 3	DCHC3		√	√		00H
FFFFF100H	Interrupt mask register 0	IMR0				√	FFFFH
FFFFF100H	Interrupt mask register 0L	IMR0L		√	√		FFH
FFFFF101H	Interrupt mask register 0H	IMR0H		√	√		FFH
FFFFF102H	Interrupt mask register 1	IMR1				√	FFFFH
FFFFF102H	Interrupt mask register 1L	IMR1L		√	√		FFH
FFFFF103H	Interrupt mask register 1H	IMR1H		√	√		FFH
FFFFF104H	Interrupt mask register 2	IMR2				√	FFFFH
FFFFF104H	Interrupt mask register 2L	IMR2L		√	√		FFH
FFFFF105H	Interrupt mask register 2H	IMR2H		√	√		FFH
FFFFF106H	Interrupt mask register 3	IMR3				√	FFFFH
FFFFF106H	Interrupt mask register 3L	IMR3L		√	√		FFH
FFFFF107H	Interrupt mask register 3H	IMR3H		√	√		FFH
FFFFF110H	Interrupt control register	LVIIC		√	√		47H
FFFFF112H	Interrupt control register	PIC0		√	√		47H
FFFFF114H	Interrupt control register	PIC1		√	√		47H
FFFFF116H	Interrupt control register	PIC2		√	√		47H
FFFFF118H	Interrupt control register	PIC3		√	√		47H
FFFFF11AH	Interrupt control register	PIC4		√	√		47H
FFFFF11CH	Interrupt control register	PIC5		√	√		47H
FFFFF11EH	Interrupt control register	PIC6		√	√		47H
FFFFF120H	Interrupt control register	PIC7		√	√		47H
FFFFF122H	Interrupt control register	TQ0OVIC		√	√		47H
FFFFF124H	Interrupt control register	TQ0CCIC0		√	√		47H
FFFFF126H	Interrupt control register	TQ0CCIC1		√	√		47H
FFFFF128H	Interrupt control register	TQ0CCIC2		√	√		47H
FFFFF12AH	Interrupt control register	TQ0CCIC3		√	√		47H
FFFFF12CH	Interrupt control register	TP0OVIC		√	√		47H
FFFFF12EH	Interrupt control register	TP0CCIC0		√	√		47H
FFFFF130H	Interrupt control register	TP0CCIC1		√	√		47H
FFFFF132H	Interrupt control register	TP1OVIC		√	√		47H
FFFFF134H	Interrupt control register	TP1CCIC0		√	√		47H
FFFFF136H	Interrupt control register	TP1CCIC1		√	√		47H
FFFFF138H	Interrupt control register	TP2OVIC		√	√		47H

(3/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF13AH	Interrupt control register	TP2CCIC0	R/W	√	√		47H
FFFFF13CH	Interrupt control register	TP2CCIC1		√	√		47H
FFFFF13EH	Interrupt control register	TP3OVIC		√	√		47H
FFFFF140H	Interrupt control register	TP3CCIC0		√	√		47H
FFFFF142H	Interrupt control register	TP3CCIC1		√	√		47H
FFFFF144H	Interrupt control register	TM0EQIC0		√	√		47H
FFFFF146H	Interrupt control register	CB0RIC/IICIC0		√	√		47H
FFFFF148H	Interrupt control register	CB0TIC		√	√		47H
FFFFF14AH	Interrupt control register	CB1RIC		√	√		47H
FFFFF14CH	Interrupt control register	CB1TIC		√	√		47H
FFFFF14EH	Interrupt control register	UA0RIC		√	√		47H
FFFFF150H	Interrupt control register	UA0TIC		√	√		47H
FFFFF152H	Interrupt control register	UA1RIC		√	√		47H
FFFFF154H	Interrupt control register	UA1TIC		√	√		47H
FFFFF156H	Interrupt control register	ADIC		√	√		47H
FFFFF160H	Interrupt control register	KRIC		√	√		47H
FFFFF162H	Interrupt control register	WTIIC		√	√		47H
FFFFF164H	Interrupt control register	WTIC		√	√		47H
FFFFF176H	Interrupt control register	UA2RIC		√	√		47H
FFFFF178H	Interrupt control register	UA2TIC		√	√		47H
FFFFF182H	Interrupt control register	DMAIC0		√	√		47H
FFFFF184H	Interrupt control register	DMAIC1		√	√		47H
FFFFF186H	Interrupt control register	DMAIC2		√	√		47H
FFFFF188H	Interrupt control register	DMAIC3		√	√		47H
FFFFF1FAH	In-service priority register	ISPR	R	√	√		00H
FFFFF1FCH	Command register	PRCMD	W		√		Undefined
FFFFF1FEH	Power save control register	PSC	R/W	√	√		00H
FFFFF200H	A/D converter mode register 0	ADA0M0		√	√		00H
FFFFF201H	A/D converter mode register 1	ADA0M1		√	√		00H
FFFFF202H	A/D converter channel specification register	ADA0S		√	√		00H
FFFFF203H	A/D converter mode register 2	ADA0M2		√	√		00H
FFFFF204H	Power-fail compare mode register	ADA0PFM		√	√		00H
FFFFF205H	Power-fail compare threshold value register	ADA0PFT		√	√		00H
FFFFF210H	A/D conversion result register 0	ADA0CR0	R			√	Undefined
FFFFF211H	A/D conversion result register 0H	ADA0CR0H			√		Undefined
FFFFF212H	A/D conversion result register 1	ADA0CR1				√	Undefined
FFFFF213H	A/D conversion result register 1H	ADA0CR1H			√		Undefined
FFFFF214H	A/D conversion result register 2	ADA0CR2				√	Undefined
FFFFF215H	A/D conversion result register 2H	ADA0CR2H			√		Undefined
FFFFF216H	A/D conversion result register 3	ADA0CR3				√	Undefined
FFFFF217H	A/D conversion result register 3H	ADA0CR3H			√		Undefined
FFFFF218H	A/D conversion result register 4	ADA0CR4				√	Undefined
FFFFF219H	A/D conversion result register 4H	ADA0CR4H			√		Undefined

(4/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF21AH	A/D conversion result register 5	ADA0CR5	R			√	Undefined
FFFFF21BH	A/D conversion result register 5H	ADA0CR5H			√		Undefined
FFFFF21CH	A/D conversion result register 6	ADA0CR6				√	Undefined
FFFFF21DH	A/D conversion result register 6H	ADA0CR6H			√		Undefined
FFFFF21EH	A/D conversion result register 7	ADA0CR7				√	Undefined
FFFFF21FH	A/D conversion result register 7H	ADA0CR7H			√		Undefined
FFFFF220H	A/D conversion result register 8	ADA0CR8				√	Undefined
FFFFF221H	A/D conversion result register 8H	ADA0CR8H			√		Undefined
FFFFF222H	A/D conversion result register 9	ADA0CR9				√	Undefined
FFFFF223H	A/D conversion result register 9H	ADA0CR9H			√		Undefined
FFFFF224H	A/D conversion result register 10	ADA0CR10				√	Undefined
FFFFF225H	A/D conversion result register 10H	ADA0CR10H			√		Undefined
FFFFF226H	A/D conversion result register 11	ADA0CR11				√	Undefined
FFFFF227H	A/D conversion result register 11H	ADA0CR11H			√		Undefined
FFFFF300H	Key return mode register	KRM	R/W	√	√		00H
FFFFF308H	Selector operation control register 0	SELCNT0		√	√		00H
FFFFF318H	Noise elimination control register	NFC		√	√		00H
FFFFF340H	IIC division clock select register 0	OCKS0			√		00H
★ FFFFF400H	Port 0 register	P0		√	√		00H[Note]
★ FFFFF402H	Port 1 register	P1		√	√		00H[Note]
★ FFFFF406H	Port 3 register	P3				√	0000H[Note]
★ FFFFF406H	Port 3 register L	P3L		√	√		00H[Note]
★ FFFFF407H	Port 3 register H	P3H		√	√		00H[Note]
★ FFFFF408H	Port 4 register	P4		√	√		00H[Note]
★ FFFFF40AH	Port 5 register	P5		√	√		00H[Note]
★ FFFFF40EH	Port 7 register L	P7L		√	√		00H[Note]
★ FFFFF40FH	Port 7 register H	P7H		√	√		00H[Note]
★ FFFFF412H	Port 9 register	P9				√	0000H[Note]
★ FFFFF412H	Port 9 register L	P9L		√	√		00H[Note]
★ FFFFF413H	Port 9 register H	P9H		√	√		00H[Note]
FFFFF420H	Port 0 mode register	PM0		√	√		FFH
FFFFF422H	Port 1 mode register	PM1		√	√		FFH
FFFFF426H	Port 3 mode register	PM3				√	FFFFH
FFFFF426H	Port 3 mode register L	PM3L		√	√		FFH
FFFFF427H	Port 3 mode register H	PM3H		√	√		FFH
FFFFF428H	Port 4 mode register	PM4		√	√		FFH
FFFFF42AH	Port 5 mode register	PM5		√	√		FFH
FFFFF42EH	Port 7 mode register L	PM7L		√	√		FFH
FFFFF42FH	Port 7 mode register H	PM7H		√	√		FFH
FFFFF432H	Port 9 mode register	PM9				√	FFFFH
FFFFF432H	Port 9 mode register L	PM9L		√	√		FFH
FFFFF433H	Port 9 mode register H	PM9H		√	√		FFH

★ **Note** The output latch is 00H or 0000H. When these registers are input, the pin statuses are read.

(5/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF440H	Port 0 mode control register	PMC0	R/W	√	√		00H
FFFFF446H	Port 3 mode control register	PMC3				√	0000H
FFFFF446H	Port 3 mode control register L	PMC3L		√	√		00H
FFFFF447H	Port 3 mode control register H	PMC3H		√	√		00H
FFFFF448H	Port 4 mode control register	PMC4		√	√		00H
FFFFF44AH	Port 5 mode control register	PMC5		√	√		00H
FFFFF452H	Port 9 mode control register	PMC9				√	0000H
FFFFF452H	Port 9 mode control register L	PMC9L		√	√		00H
FFFFF453H	Port 9 mode control register H	PMC9H		√	√		00H
FFFFF460H	Port 0 function control register	PFC0		√	√		00H
FFFFF466H	Port 3 function control register L	PFC3L		√	√		00H
FFFFF468H	Port 4 function control register	PFC4		√	√		00H
FFFFF46AH	Port 5 function control register	PFC5		√	√		00H
FFFFF472H	Port 9 function control register	PFC9				√	0000H
FFFFF472H	Port 9 function control register L	PFC9L		√	√		00H
FFFFF473H	Port 9 function control register H	PFC9H		√	√		00H
FFFFF484H	Data wait control register 0	DWC0				√	7777H
FFFFF488H	Address wait control register	AWC				√	FFFFH
FFFFF48AH	Bus cycle control register	BCC				√	AAAAH
FFFFF540H	TMQ0 control register 0	TQ0CTL0		√	√		00H
FFFFF541H	TMQ0 control register 1	TQ0CTL1		√	√		00H
FFFFF542H	TMQ0 I/O control register 0	TQ0IOC0		√	√		00H
FFFFF543H	TMQ0 I/O control register 1	TQ0IOC1		√	√		00H
FFFFF544H	TMQ0 I/O control register 2	TQ0IOC2		√	√		00H
FFFFF545H	TMQ0 option register 0	TQ0OPT0		√	√		00H
FFFFF546H	TMQ0 capture/compare register 0	TQ0CCR0			√	√	0000H
FFFFF548H	TMQ0 capture/compare register 1	TQ0CCR1			√	√	0000H
FFFFF54AH	TMQ0 capture/compare register 2	TQ0CCR2			√	√	0000H
FFFFF54CH	TMQ0 capture/compare register 3	TQ0CCR3			√	√	0000H
FFFFF54EH	TMQ0 counter read buffer register	TQ0CNT	R			√	0000H
FFFFF590H	TMP0 control register 0	TP0CTL0	R/W	√	√		00H
FFFFF591H	TMP0 control register 1	TP0CTL1		√	√		00H
FFFFF592H	TMP0 I/O control register 0	TP0IOC0		√	√		00H
FFFFF593H	TMP0 I/O control register 1	TP0IOC1		√	√		00H
FFFFF594H	TMP0 I/O control register 2	TP0IOC2		√	√		00H
FFFFF595H	TMP0 option register 0	TP0OPT0		√	√		00H
FFFFF596H	TMP0 capture/compare register 0	TP0CCR0				√	0000H
FFFFF598H	TMP0 capture/compare register 1	TP0CCR1				√	0000H
FFFFF59AH	TMP0 counter read buffer register	TP0CNT	R			√	0000H
FFFFF5A0H	TMP1 control register 0	TP1CTL0	R/W	√	√		00H
FFFFF5A1H	TMP1 control register 1	TP1CTL1		√	√		00H
FFFFF5A2H	TMP1 I/O control register 0	TP1IOC0		√	√		00H
FFFFF5A3H	TMP1 I/O control register 1	TP1IOC1		√	√		00H

(6/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF5A4H	TMP1 I/O control register 2	TP1IOC2	R/W	√	√		00H
FFFFF5A5H	TMP1 option register 0	TP1OPT0		√	√		00H
FFFFF5A6H	TMP1 capture/compare register 0	TP1CCR0				√	0000H
FFFFF5A8H	TMP1 capture/compare register 1	TP1CCR1				√	0000H
FFFFF5AAH	TMP1 counter read buffer register	TP1CNT	R			√	0000H
FFFFF5B0H	TMP2 control register 0	TP2CTL0	R/W	√	√		00H
FFFFF5B1H	TMP2 control register 1	TP2CTL1		√	√		00H
FFFFF5B2H	TMP2 I/O control register 0	TP2IOC0		√	√		00H
FFFFF5B3H	TMP2 I/O control register 1	TP2IOC1		√	√		00H
FFFFF5B4H	TMP2 I/O control register 2	TP2IOC2		√	√		00H
FFFFF5B5H	TMP2 option register 0	TP2OPT0		√	√		00H
FFFFF5B6H	TMP2 capture/compare register 0	TP2CCR0				√	0000H
FFFFF5B8H	TMP2 capture/compare register 1	TP2CCR1				√	0000H
FFFFF5BAH	TMP2 counter read buffer register	TP2CNT	R			√	0000H
FFFFF5C0H	TMP3 control register 0	TP3CTL0	R/W	√	√		00H
FFFFF5C1H	TMP3 control register 1	TP3CTL1		√	√		00H
FFFFF5C2H	TMP3 I/O control register 0	TP3IOC0		√	√		00H
FFFFF5C3H	TMP3 I/O control register 1	TP3IOC1		√	√		00H
FFFFF5C4H	TMP3 I/O control register 2	TP3IOC2		√	√		00H
FFFFF5C5H	TMP3 option register 0	TP3OPT0		√	√		00H
FFFFF5C6H	TMP3 capture/compare register 0	TP3CCR0				√	0000H
FFFFF5C8H	TMP3 capture/compare register 1	TP3CCR1				√	0000H
FFFFF5CAH	TMP3 counter read buffer register	TP3CNT	R			√	0000H
FFFFF680H	Watch timer operation mode register	WTM	R/W	√	√		00H
FFFFF690H	TMM0 timer control register 0	TM0CTL0		√	√		00H
FFFFF694H	TMM0 compare register 0	TM0CMP0				√	0000H
FFFFF6C0H	Oscillation stabilization time select register	OSTS			√		06H
FFFFF6C1H	PLL lockup time specification register	PLLS			√		03H
FFFFF6D0H	Watchdog timer mode register 2	WDTM2		√	√		67H
FFFFF6D1H	Watchdog timer enable register	WDTE			√		9AH
FFFFF706H	Port 3 function control expansion register L	PFCE3L		√	√		00H
FFFFF70AH	Port 5 function control expansion register	PFCE5		√	√		00H
FFFFF712H	Port 9 function control expansion register L	PFCE9L		√	√		00H
FFFFF802H	System status register	SYS		√	√		00H
FFFFF80CH	Ring-OSC mode register	RCM		√	√		00H
FFFFF810H	DMA trigger factor register 0	DTFR0		√	√		00H
FFFFF812H	DMA trigger factor register 1	DTFR1		√	√		00H
FFFFF814H	DMA trigger factor register 2	DTFR2		√	√		00H
FFFFF816H	DMA trigger factor register 3	DTFR3		√	√		00H
FFFFF820H	Power save mode register	PSMR		√	√		00H
FFFFF824H	Lock register	LOCKR	R	√	√		00H
FFFFF828H	Processor clock control register	PCC	R/W	√	√		03H
FFFFF82CH	PLL control register	PLLCTL		√	√		01H

(7/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits				Default Value
				1	8	16	32	
FFFFF82EH	CPU operation clock status register	CCLS	R	√	√			00H
FFFFF840H	Correction address register 0	CORAD0	R/W				√	00000000H
FFFFF840H	Correction address register 0L	CORAD0L				√		0000H
FFFFF842H	Correction address register 0H	CORAD0H				√		0000H
FFFFF844H	Correction address register 1	CORAD1					√	00000000H
FFFFF844H	Correction address register 1L	CORAD1L				√		0000H
FFFFF846H	Correction address register 1H	CORAD1H				√		0000H
FFFFF848H	Correction address register 2	CORAD2					√	00000000H
FFFFF848H	Correction address register 2L	CORAD2L				√		0000H
FFFFF84AH	Correction address register 2H	CORAD2H				√		0000H
FFFFF84CH	Correction address register 3	CORAD3					√	00000000H
FFFFF84CH	Correction address register 3L	CORAD3L				√		0000H
FFFFF84EH	Correction address register 3H	CORAD3H				√		0000H
FFFFF870H	Clock monitor mode register	CLM		√	√			00H
FFFFF880H	Correction control register	CORCN		√	√			00H
FFFFF888H	Reset source flag register 0	RESF		√	√			00H
FFFFF890H	Low-voltage detection register 0	LVIM		√	√			00H
FFFFF891H	Low-voltage detection-level select register	LVIS			√			00H
FFFFF892H	Internal RAM data status register	RAMS		√	√			01H
FFFFF8B0H	Prescaler mode register 0	PRSM0		√	√			00H
FFFFF8B1H	Prescaler compare register 0	PRSCM0			√			00H
FFFFF9FCH	On-chip debug alternate-function pin setting register	OCDM		√	√			01H
FFFFFA00H	UARTA0 control register 0	UA0CTL0		√	√			10H
FFFFFA01H	UARTA0 control register 1	UA0CTL1			√			00H
FFFFFA02H	UARTA0 control register 2	UA0CTL2			√			FFH
FFFFFA03H	UARTA0 option control register 0	UA0OPT0		√	√			14H
FFFFFA04H	UARTA0 status register	UA0STR		√	√			00H
FFFFFA06H	UARTA0 receive data register	UA0RX	R		√			FFH
FFFFFA07H	UARTA0 transmit data register	UA0TX	R/W		√			FFH
FFFFFA10H	UARTA1 control register 0	UA1CTL0		√	√			10H
FFFFFA11H	UARTA1 control register 1	UA1CTL1			√			00H
FFFFFA12H	UARTA1 control register 2	UA1CTL2			√			FFH
FFFFFA13H	UARTA1 option control register 0	UA1OPT0		√	√			14H
FFFFFA14H	UARTA1 status register	UA1STR		√	√			00H
FFFFFA16H	UARTA1 receive data register	UA1RX	R		√			FFH
FFFFFA17H	UARTA1 transmit data register	UA1TX	R/W		√			FFH
FFFFFA20H	UARTA2 control register 0	UA2CTL0		√	√			10H
FFFFFA21H	UARTA2 control register 1	UA2CTL1			√			00H
FFFFFA22H	UARTA2 control register 2	UA2CTL2			√			FFH
FFFFFA23H	UARTA2 option control register 0	UA2OPT0		√	√			14H
FFFFFA24H	UARTA2 status register	UA2STR		√	√			00H
FFFFFA26H	UARTA2 receive data register	UA2RX	R		√			FFH
FFFFFA27H	UARTA2 transmit data register	UA2TX	R/W		√			FFH

(8/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFFB00H	TIP00 noise elimination control register	P00NFC	R/W	√	√		00H
FFFFFB04H	TIP01 noise elimination control register	P01NFC		√	√		00H
FFFFFB08H	TIP10 noise elimination control register	P10NFC		√	√		00H
FFFFFB0CH	TIP11 noise elimination control register	P11NFC		√	√		00H
FFFFFB10H	TIP20 noise elimination control register	P20NFC		√	√		00H
FFFFFB14H	TIP21 noise elimination control register	P21NFC		√	√		00H
FFFFFB18H	TIP30 noise elimination control register	P30NFC		√	√		00H
FFFFFB1CH	TIP31 noise elimination control register	P31NFC		√	√		00H
FFFFFB50H	TIQ00 noise elimination control register	Q00NFC		√	√		00H
FFFFFB54H	TIQ01 noise elimination control register	Q01NFC		√	√		00H
FFFFFB58H	TIQ02 noise elimination control register	Q02NFC		√	√		00H
FFFFFB5CH	TIQ03 noise elimination control register	Q03NFC		√	√		00H
FFFFFC00H	External interrupt falling edge specification register 0	INTF0		√	√		00H
FFFFFC06H	External interrupt falling edge specification register 3L	INTF3L		√	√		00H
FFFFFC13H	External interrupt falling edge specification register 9H	INTF9H		√	√		00H
FFFFFC20H	External interrupt rising edge specification register 0	INTR0		√	√		00H
FFFFFC26H	External interrupt rising edge specification register 3L	INTR3L		√	√		00H
FFFFFC33H	External interrupt rising edge specification register 9H	INTR9H		√	√		00H
FFFFFC40H	Pull-up resistor option register 0	PU0		√	√		00H
FFFFFC42H	Pull-up resistor option register 1	PU1		√	√		00H
FFFFFC46H	Pull-up resistor option register 3	PU3				√	0000H
FFFFFC46H	Pull-up resistor option register 3L	PU3L		√	√		00H
FFFFFC47H	Pull-up resistor option register 3H	PU3H		√	√		00H
FFFFFC48H	Pull-up resistor option register 4	PU4		√	√		00H
FFFFFC4AH	Pull-up resistor option register 5	PU5		√	√		00H
FFFFFC52H	Pull-up resistor option register 9	PU9				√	0000H
FFFFFC52H	Pull-up resistor option register 9L	PU9L		√	√		00H
FFFFFC53H	Pull-up resistor option register 9H	PU9H		√	√		00H
FFFFFC68H	Port 4 function control register	PF4		√	√		00H
FFFFFD00H	CSIB0 control register 0	CB0CTL0		√	√		01H
FFFFFD01H	CSIB0 control register 1	CB0CTL1		√	√		00H
FFFFFD02H	CSIB0 control register 2	CB0CTL2			√		00H
FFFFFD03H	CSIB0 status register	CB0STR		√	√		00H
FFFFFD04H	CSIB0 receive data register	CB0RX	R			√	0000H
FFFFFD04H	CSIB0 receive data register L	CB0RXL			√		00H
FFFFFD06H	CSIB0 transmit data register	CB0TX	R/W			√	0000H
FFFFFD06H	CSIB0 transmit data register L	CB0TXL			√		00H
FFFFFD10H	CSIB1 control register 0	CB1CTL0		√	√		01H
FFFFFD11H	CSIB1 control register 1	CB1CTL1		√	√		00H
FFFFFD12H	CSIB1 control register 2	CB1CTL2			√		00H
FFFFFD13H	CSIB1 status register	CB1STR		√	√		00H
FFFFFD14H	CSIB1 receive data register	CB1RX	R			√	0000H
FFFFFD14H	CSIB1 receive data register L	CB1RXL			√		00H

(9/9)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFFD16H	CSIB1 transmit data register	CB1TX	R/W			√	0000H
FFFFFD16H	CSIB1 transmit data register L	CB1TXL			√		00H
FFFFFD80H	IIC shift register 0	IIC0			√		00H
FFFFFD82H	IIC control register 0	IICC0		√	√		00H
FFFFFD83H	Slave address register 0	SVA0			√		00H
FFFFFD84H	IIC clock select register 0	IICCL0		√	√		00H
FFFFFD85H	IIC function expansion register 0	IICX0		√	√		00H
FFFFFD86H	IIC status register 0	IICS0	R	√	√		00H
FFFFFD8AH	IIC flag register 0	IICF0	R/W	√	√		00H

3.4.7 Special registers

Special registers are registers that are protected from being written with illegal data due to a program hang-up. The μPD703229Y and 70F3229Y have the following seven special registers.

- Power save control register (PSC)
- Processor clock control register (PCC)
- On-chip debug mode setting register (OCDM)
- Clock monitor mode register (CLM)
- Reset source flag register (RESF)
- Low-voltage detection register (LVIM)
- Internal RAM data status register (RAMS)

In addition, a command register (PRCDM) is provided to protect against a write access to the special registers so that the application system does not inadvertently stop due to a program hang-up. A write access to the special registers is made in a specific sequence, and an illegal store operation is reported to the system status register (SYS).

(1) Setting data to special registers

Set data to the special registers in the following sequence.

<1> Disable DMA operation.
<2> Prepare data to be set to the special register in a general-purpose register.
<3> Write the data prepared in <2> to the PRCMD register.
<4> Write the setting data to the special register (by using the following instructions).
- Store instruction (ST/SST instruction)
- Bit manipulation instruction (SET1/CLR1/NOT1 instruction)

(<5> to <9> Insert NOP instructions (5 instructions).)[Note]
<10> Enable DMA operation if necessary.

[Example] With PSC register (setting standby mode)

```
   ST.B r11, PSMR[r0]    ; Set PSMR register (setting IDLE1, IDLE2, and STOP modes).
<1>CLR1 0, DCHCn[r0]     ; Disable DMA operation.  n = 0 to 3
<2>MOV 0x02, r10
<3>ST.B r10, PRCMD[r0]   ; Write PRCMD register.
<4>ST.B r10, PSC[r0]     ; Set PSC register.
<5>NOP[Note]             ; Dummy instruction
<6>NOP[Note]             ; Dummy instruction
<7>NOP[Note]             ; Dummy instruction
<8>NOP[Note]             ; Dummy instruction
<9>NOP[Note]             ; Dummy instruction
<10>SET10, DCHCn[r0]     ; Enable DMA operation.  n = 0 to 3
(next instruction)
```

There is no special sequence to read a special register.

Note Five NOP instructions or more must be inserted immediately after setting the IDLE1 mode, IDLE2 mode, or STOP mode (by setting the PSC.STP bit to 1).

Cautions 1. When a store instruction is executed to store data in the command register, interrupts are not acknowledged. This is because it is assumed that steps <3> and <4> above are performed by successive store instructions. If another instruction is placed between <3> and <4>, and if an interrupt is acknowledged by that instruction, the above sequence may not be established, causing malfunction.

2. Although dummy data is written to the PRCMD register, use the same general-purpose register used to set the special register (<4> in Example) to write data to the PRCMD register (<3> in Example). The same applies when a general-purpose register is used for addressing.

(2) Command register (PRCMD)

The PRCMD register is an 8-bit register that protects the registers that may seriously affect the application system from being written, so that the system does not inadvertently stop due to a program hang-up. The first write access to a special register is valid after data has been written in advance to the PRCMD register. In this way, the value of the special register can be rewritten only in a specific sequence, so as to protect the register from an illegal write access.

The PRCMD register is write-only, in 8-bit units (undefined data is read when this register is read).

Reset input makes this register undefined.

After reset: Undefined W Address: FFFFF1FCH

	7	6	5	4	3	2	1	0
PRCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0

(3) System status register (SYS)

Status flags that indicate the operation status of the overall system are allocated to this register.
This register can be read or written in 8-bit or 1-bit units.
Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF802H

	7	6	5	4	3	2	1	<0>
SYS	0	0	0	0	0	0	0	PRERR

PRERR	Detects protection error
0	Protection error did not occur.
1	Protection error occurred.

The PRERR flag operates under the following conditions.

(a) Set condition (PRERR flag = 1)

(i) When data is written to a special register without writing anything to the PRCMD register (when <4> is executed without executing <3> in **3.4.7 (1) Setting data to special registers**)

(ii) When data is written to a peripheral I/O register other than a special register (including execution of a bit manipulation instruction) after writing data to the PRCMD register (if <4> in **3.4.7 (1) Setting data to special registers** is not the setting of a special register)

★ **Remark** Even if an on-chip peripheral I/O register is read (except by a bit manipulation instruction) between an operation to write the PRCMD register and an operation to write a special register, the PRERR flag is not set, and the set data can be written to the special register.

(b) Clear condition (PRERR flag = 0)

(i) When 0 is written to the PRERR flag

(ii) When the system is reset

Cautions 1. If 0 is written to the PRERR bit of the SYS register, which is not a special register, immediately after a write access to the PRCMD register, the PRERR bit is cleared to 0 (the write access takes precedence).

2. If data is written to the PRCMD register, which is not a special register, immediately after a write access to the PRCMD register, the PRERR bit is set to 1.

3.4.8 Notes

(1) Registers to be set first

Be sure to set the following registers first when using the μPD703229Y and 70F3229Y.

- System wait control register (VSWC)
- On-chip debug mode register (OCDM)

After setting the OCDM register, set the VSWC register and set the other registers as necessary.
When using the external bus, set each pin as the alternate-function bus control pin by using the port-related registers after setting the above registers.

(a) System wait control register (VSWC)

The VSWC register controls wait of bus access to the on-chip peripheral I/O registers.
Three clocks are required to access an on-chip peripheral I/O register (without a wait cycle). The μPD703229Y and 70F3229Y require wait cycles according to the operating frequency. Set the VSWC register to 01H to access the bus.
The VSWC register can be read or written in 8-bit units (address: FFFFF06EH, default value: 77H).

Operating Frequency (f_{CLK})	Set Value of VSWC	Number of Waits
32 kHz ≤ f_{CLK} < 16.6 MHz	00H	0 (no waits)
16.6 MHz ≤ f_{CLK} ≤ 20 MHz	01H	1

(b) On-chip debug mode register (OCDM)

For details, refer to **CHAPTER 24 ON-CHIP DEBUG FUNCTION**.

★ **(2) Access to special on-chip peripheral I/O registers**

This product has two types of internal system buses.

One type is a bus for the CPU and the other is a peripheral bus for interfacing with low-speed peripheral hardware.

Since the CPU bus clock and peripheral bus clock are asynchronous, if a conflict occurs during access between the CPU and peripheral hardware, illegal data may be passed unexpectedly. Therefore, when accessing peripheral hardware that may cause a conflict, the number of access cycles is changed so that the data is received/passed correctly in the CPU. As a result, the CPU does not shift to the next instruction processing and enters the wait status. When this wait status occurs, the number of execution clocks of the instruction is increased by the number of wait clocks as indicated below.

Note this with caution when performing real-time processing.

When accessing a special on-chip peripheral I/O register, further waits may be required in addition to the waits set by the VSWC register.

The access conditions at that time and the method used to calculate the number of waits to be inserted (number of CPU clocks) are shown below.

Peripheral Function	Register Name	Access	k
16-bit timer/event counter P (TMP) (n = 0 to 3)	TPnCNT	Read	1
	TPnCCR0	Read/write	1
	TPnCCR1	Read/write	1
16-bit timer/event counter Q (TMQ)	TQ0CNT	Read	1
	TQ0CCR0	Read/write	1
	TQ0CCR1	Read/write	1
	TQ0CCR2	Read/write	1
	TQ0CCR3	Read/write	1
Watchdog timer 2 (WDT2)	WDTM2	Write (during WDT2 operation)	3
I^2C00	IICS0	Read	1
A/D converter	ADA0M0	Read	1
	ADA0CR0 to ADA0CR11	Read	1
	ADA0CR0H to ADA0CR11H	Read	1

Number of clocks for access $= 3 + (k \times 2)$ [clocks] (when VSWC register = 00H)

$= 4 + (k \times 3)$ [clocks] (when VSWC register = 01H)

Caution When the CPU operates on the subclock and main clock oscillation is stopped, accessing a register in which a wait occurs is prohibited. If a wait occurs, it can only be released by a reset.

★ **(3) Cautions when using flash memory version (μPD70F3229Y)**

000007AH is the area reserved by the system. The data in 000007AH must be 00H.

Caution When the data in the flash memory has been deleted, all the bits are set to 1.

CHAPTER 4 PORT FUNCTIONS

4.1 Features

- I/O ports: 84
- Input/output specifiable in 1-bit units

4.2 Basic Port Configuration

The μPD703229Y and 70F3229Y feature a total of 84 I/O ports consisting of the ports 0, 1, 3 to 5, 7, 9, CM, CS, CT, DH, and DL. The port configuration is shown below.

Figure 4-1. Port Configuration Diagram

Table 4-1. I/O Buffer Power Supplies for Pins

Power Supply	Corresponding Pins
AV_{REF0}	Port 7
BV_{DD}	Ports CM, CS, CT, DH, DL
EV_{DD}	$\overline{RESET}$, ports 0, 1, 3 to 5, 9

★ 4.3 Port Configuration

Table 4-2. Port Configuration

Item	Configuration
Control registers	Port n register (Pn: n = 0, 1, 3 to 5, 7, 9, CM, CS, CT, DH, DL) Port n mode register (PMn: n = 0, 1, 3 to 5, 7, 9, CM, CS, CT, DH, DL) Port n mode control register (PMCn: n = 0, 3 to 5, 9, CM, CT, CS, DH, DL) Port n function control register (PFCn: n = 0, 3 to 5, 9) Port n function control expansion register (PFCEn: n = 3, 5, 9) Port n function register (PF4) Pull-up resistor option register n (PUn: n = 0, 1, 3 to 5, 9)
Ports	I/O: 84

(1) Port n register (Pn)

Data is input from or output to an external device by writing or reading the Pn register.

The Pn register consists of a port latch that holds output data, and a circuit that reads the status of pins.

Each bit of the Pn register corresponds to one pin of port n, and can be read or written in 1-bit units.

After reset: 00H (output latch) R/W

	7	6	5	4	3	2	1	0
Pn	Pn7	Pn6	Pn5	Pn4	Pn3	Pn2	Pn1	Pn0

Pnm	Control of output data (in output mode)
0	Output 0.
1	Output 1.

Data is written to or read from the Pn register as follows, regardless of the setting of the PMCn register.

Table 4-3. Writing/Reading Pn Register

Setting of PMn Register	Writing to Pn Register	Reading from Pn Register
Output mode (PMnm = 0)	Data is written to the output latch[Note]. In the port mode (PMCn = 0), the contents of the output latch are output from the pins.	The value of the output latch is read.
Input mode (PMnm = 1)	Data is written to the output latch. The pin status is not affected[Note].	The pin status is read.

Note The value written to the output latch is retained until a new value is written to the output latch.

(2) Port n mode register (PMn)

The PMn register specifies the input or output mode of the corresponding port pin.

Each bit of this register corresponds to one pin of port n, and the input or output mode can be specified in 1-bit units.

After reset: FFH R/W

	7	6	5	4	3	2	1	0
PMn	PMn7	PMn6	PMn5	PMn4	PMn3	PMn2	PMn1	PMn0

PMnm	Control of I/O mode
0	Output mode
1	Input mode

(3) Port n mode control register (PMCn)

The PMCn register specifies the port mode or alternate function.

Each bit of this register corresponds to one pin of port n, and the mode of the port can be specified in 1-bit units.

After reset: 00H R/W

	7	6	5	4	3	2	1	0
PMCn	PMCn7	PMCn6	PMCn5	PMCn4	PMCn3	PMCn2	PMCn1	PMCn0

PMCnm	Specification of operation mode
0	Port mode
1	Alternate function mode

(4) Port n function control register (PFCn)

The PFCn register specifies the alternate function of a port pin to be used if the pin has two alternate functions.

Each bit of this register corresponds to one pin of port n, and the alternate function of a port pin can be specified in 1-bit units.

After reset: 00H R/W

	7	6	5	4	3	2	1	0
PFCn	PFCn7	PFCn6	PFCn5	PFCn4	PFCn3	PFCn2	PFCn1	PFCn0

PFCnm	Specification of alternate function
0	Alternate function 1
1	Alternate function 2

(5) Port n function control expansion register (PFCEn)

The PFCEn register specifies the alternate function of a port pin to be used if the pin has three or more alternate functions.

Each bit of this register corresponds to one pin of port n, and the alternate function of a port pin can be specified in 1-bit units.

After reset: 00H R/W

	7	6	5	4	3	2	1	0
PFCEn	PFCEn7	PFCEn6	PFCEn5	PFCEn4	PFCEn3	PFCEn2	PFCEn1	PFCEn0

	7	6	5	4	3	2	1	0
PFCn	PFCn7	PFCn6	PFCn5	PFCn4	PFCn3	PFCn2	PFCn1	PFCn0

PFCEnm	PFCnm	Specification of alternate function
0	0	Alternate function 1
0	1	Alternate function 2
1	0	Alternate function 3
1	1	Alternate function 4

(6) Port n function register (PFn)

The PFn register specifies normal output or N-ch open-drain output.

Each bit of this register corresponds to one pin of port n, and the output mode of the port pin can be specified in 1-bit units.

After reset: 00H R/W

	7	6	5	4	3	2	1	0
PFn	PFn7	PFn6	PFn5	PFn4	PFn3	PFn2	PFn1	PFn0

PFnm[Note]	Control of normal output/N-ch open-drain output
0	Normal output (CMOS output)
1	N-ch open-drain output

Note The PFnm bit of the PFn register is valid only when the PMnm bit of the PMn register is 0 (when the output mode is specified) in port mode (PMCnm bit = 0). When the PMnm bit is 1 (when the input mode is specified), the set value of the PFn register is invalid.

(7) Pull-up resistor option register (PUn)

The PUn register specifies the connection of an on-chip pull-up resistor.

Each bit of the pull-up resistor option register corresponds to one pin of port n and can be specified in 1-bit units.

After reset: 00H R/W

	7	6	5	4	3	2	1	0
PUn	PUn7	PUn6	PUn5	PUn4	PUn3	PUn2	PUn1	PUn0

PUnm	Control of on-chip pull-up resistor connection
0	Not connected
1	Connected

(8) Port setting

Set a port as illustrated below.

Figure 4-2. Setting of Each Register and Pin Function

	PFCEnm	PFCnm
(a)	0	0
(b)	0	1
(c)	1	0
(d)	1	1

Remark Set the alternate functions in the following sequence.

<1> Set the PFCn and PFCEn registers.
<2> Set the PMCn register.
<3> Set the INTRn or INTFn register (to specify an external interrupt pin).

If the PMCn register is set first, an unintended function may be set while the PFCn and PFCEn registers are being set.

4.3.1 Port 0

Port 0 is a 7-bit port for which I/O settings can be controlled in 1-bit units.

Port 0 includes the following alternate-function pins.

Table 4-4. Port 0 Alternate-Function Pins

Pin Name	Pin No.	Alternate Function	I/O	PULL[Note 1]	Remark	Block Type
P00	6	TIP31/TOP31	I/O	Provided	Digital noise elimination[Note 2]	G-1
P01	7	TIP30/TOP30	I/O			G-1
P02	17	NMI[Note 3]	Input		Analog noise elimination	L-1
P03	18	INTP0/ADTRG	Input			N-1
P04	19	INTP1	Input			L-1
P05	20	INTP2/$\overline{\text{DRST}}$[Note 4]	Input			AA-1
P06	21	INTP3	Input		**Note 5**	L-2

Notes 1. Software pull-up function

2. For details of digital noise elimination, refer to **7.6 Eliminating Digital Noise on Timer Input Pin**.

3. The NMI pin function is enabled by setting the PMC0, INTF0, and INTR0 registers. Note that the NMI pin is not enabled if the PMC0, INTF0, and INTR0 registers are not set. How to enable the NMI pin is shown below.

<1> Set the PMC0.PMC02 bit to 1 and enable the alternate function (NMI).

<2> Set any value to the INTF0.INTF02 and INTR0.INTR02 bits and set the desired edge detection.

4. The $\overline{\text{DRST}}$ pin is the pin for on-chip debugging (μPD70F3229Y only).

When on-chip debugging is not used, fix the P05/INTP2/$\overline{\text{DRST}}$ pin state to low level after reset is released by the $\overline{\text{RESET}}$ pin until the OCDM.OCDM0 bit is cleared (0).

For details, refer to **4.6.3 Cautions on on-chip debug pins**.

5. For how to set the internal noise filter of the INTP3 pin (analog delay/digital delay), refer to **17.4.2 (4) Noise elimination control register**.

Caution Pins P00 to P06 have hysteresis characteristics in the input mode of the alternate function, but not in port mode.

(1) Port 0 register (P0)

After reset: 00H (output latch) R/W Address: FFFFF400H

	7	6	5	4	3	2	1	0
P0	0	P06	P05	P04	P03	P02	P01	P00

P0n	Control of output data (in output mode) (n = 0 to 6)
0	Outputs 0
1	Outputs 1

(2) Port 0 mode register (PM0)

After reset: FFH R/W Address: FFFFF420H

	7	6	5	4	3	2	1	0
PM0	1	PM06	PM05	PM04	PM03	PM02	PM01	PM00

PM0n	Control of I/O mode (n = 0 to 6)
0	Output mode
1	Input mode

(3) Port 0 mode control register (PMC0)

After reset: 00H R/W Address: FFFFF440H

	7	6	5	4	3	2	1	0
PMC0	0	PMC06	PMC05	PMC04	PMC03	PMC02	PMC01	PMC00

PMC06	Specification of P06 pin operation mode
0	I/O port
1	INTP3 input

PMC05	Specification of P05 pin operation mode
0	I/O port
1	INTP2 input

PMC04	Specification of P04 pin operation mode
0	I/O port
1	INTP1 input

PMC03	Specification of P03 pin operation mode
0	I/O port
1	INTP0/ADTRG input

PMC02	Specification of P02 pin operation mode
0	I/O port
1	NMI input

PMC01	Specification of P01 pin operation mode
0	I/O port
1	TIP30/TOP30 I/O

PMC00	Specification of P00 pin operation mode
0	I/O port
1	TIP31/TOP31 I/O

Caution The P05/INTP2/$\overline{\text{DRST}}$ pin becomes $\overline{\text{DRST}}$ regardless of the value of the PMC05 bit when the OCDM.OCDM0 bit = 1.

(4) Port 0 function control register (PFC0)

After reset: 00H R/W Address: FFFFF460H

	7	6	5	4	3	2	1	0
PFC0	0	0	0	0	PFC03	0	PFC01	PFC00

PFC03	Specification of P03 pin alternate function
0	INTP0 input
1	ADTRG input

PFC01	Specification of P01 pin alternate function
0	TIP30 input
1	TOP30 output

PFC00	Specification of P00 pin alternate function
0	TIP31 input
1	TOP31 output

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

After reset: 00H R/W Address: FFFFFC40H

	7	6	5	4	3	2	1	0
PU0	0	PU06	PU05	PU04	PU03	PU02	PU01	PU00

PU0n	Control of on-chip pull-up resistor connection (n = 0 to 6)
0	Not connected
1	Connected

4.3.2 Port 1

Port 1 is a 2-bit port for which I/O settings can be controlled in 1-bit units.

Port 1 includes the following alternate-function pins.

Table 4-5. Port 1 Alternate-Function Pins

Pin Name	Pin No.	Alternate Function	I/O	PULL[Note]	Remark	Block Type
P10	3	–	–	Provided	–	C-1
P11	4	–	–			C-1

Note Software pull-up function

(1) Port 1 register (P1)

After reset: 00H (output latch) R/W Address: FFFFF402H

	7	6	5	4	3	2	1	0
P1	0	0	0	0	0	0	P11	P10

P1n	Control of output data (in output mode) (n = 0, 1)
0	Outputs 0
1	Outputs 1

(2) Port 1 mode register (PM1)

After reset: FFH R/W Address: FFFFF422H

	7	6	5	4	3	2	1	0
PM1	1	1	1	1	1	1	PM11	PM10

PM1n	Control of I/O mode (n = 0, 1)
0	Output mode
1	Input mode

(3) Pull-up resistor option register 1 (PU1)

After reset: 00H R/W Address: FFFFFC42H

	7	6	5	4	3	2	1	0
PU1	0	0	0	0	0	0	PU11	PU10

PU1n	Control of on-chip pull-up resistor connection (n = 0, 1)
0	Not connected
1	Connected

4.3.3 Port 3

Port 3 is a 10-bit port for which I/O settings can be controlled in 1-bit units.

Port 3 includes the following alternate-function pins.

Table 4-6. Port 3 Alternate-Function Pins

Pin Name	Pin No.	Alternate Function	I/O	PULL[Note 1]	Remark	Block Type
P30	25	TXDA0	Output	Provided	–	E-2
P31	26	RXDA0	Input		–	L-2
		INTP7	Input		Analog noise elimination	L-2
P32	27	ASCKA0	Input		–	U-1
		TIP00/TOP00	I/O		Digital noise elimination[Note 2]	U-1
P33	28	TIP01/TOP01	I/O		Digital noise elimination[Note 2]	G-1
P34	29	TIP10/TOP10	I/O			G-1
P35	30	TIP11/TOP11	I/O			G-1
P36	31	–	–		–	C-1
P37	32	–	–			C-1
P38	35	TXDA2	Output			E-2
P39	36	RXDA2	Input			E-1

Notes 1. Software pull-up function

2. For details of digital noise elimination, refer to **7.6 Eliminating Digital Noise on Timer Input Pin**.

Caution Pins P31 to P35, and P39 have hysteresis characteristics in the input mode of the alternate function, but not in port mode.

(1) Port 3 register (P3)

After reset: 0000H (output latch) R/W Address: P3 FFFFF406H,
P3L FFFFF406H, P3H FFFFF407H

	15	14	13	12	11	10	9	8
P3 (P3H[Note])	0	0	0	0	0	0	P39	P38
	7	6	5	4	3	2	1	0
(P3L)	P37	P36	P35	P34	P33	P32	P31	P30

P3n	Control of output data (in output mode) (n = 0 to 9)
0	Outputs 0
1	Outputs 1

Note To read/write bits 8 to 15 of the P3 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the P3H register.

Remark The P3 register can be read or written in 16-bit units.
However, when using the higher 8 bits of the P3 register as the P3H register and the lower 8 bits as the P3L register, they can be read or written in 8-bit or 1-bit units.

(2) Port 3 mode register (PM3)

After reset: FFFFH R/W Address: PM3 FFFFF426H,
PM3L FFFFF426H, PM3H FFFFF427H

	15	14	13	12	11	10	9	8
PM3 (PM3H[Note])	1	1	1	1	1	1	PM39	PM38
	7	6	5	4	3	2	1	0
(PM3L)	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30

PM3n	Control of I/O mode (n = 0 to 9)
0	Output mode
1	Input mode

Note To read/write bits 8 to 15 of the PM3 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PM3H register.

Remark The PM3 register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PM3 register as the PM3H register and the lower 8 bits as the PM3L register, they can be read or written in 8-bit or 1-bit units.

(3) Port 3 mode control register (PMC3)

(1/2)

After reset: 0000H R/W Address: PMC3 FFFFF446H,
PMC3L FFFFF446H, PMC3H FFFFF447H

	15	14	13	12	11	10	9	8
PMC3 (PMC3H[Note])	0	0	0	0	0	0	PMC39	PMC38
	7	6	5	4	3	2	1	0
(PMC3L)	0	0	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

PMC39	Specification of P39 pin operation mode
0	I/O port
1	RXDA2 input

PMC38	Specification of P38 pin operation mode
0	I/O port
1	TXDA2 output

Note To read/write bits 8 to 15 of the PMC3 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PMC3H register.

Remark The PMC3 register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PMC3 register as the PMC3H register and the lower 8 bits as the PMC3L register, they can be read or written in 8-bit or 1-bit units.

(2/2)

PMC35	Specification of P35 pin operation mode
0	I/O port
1	TIP11/TOP11 I/O

PMC34	Specification of P34 pin operation mode
0	I/O port
1	TIP10/TOP10 I/O

PMC33	Specification of P33 pin operation mode
0	I/O port
1	TIP01/TOP01 I/O

PMC32	Specification of P32 pin operation mode
0	I/O port
1	ASCKA0/TIP00/TOP00 I/O

PMC31	Specification of P31 pin operation mode
0	I/O port
1	RXDA0/INTP7 input[Note]

PMC30	Specification of P30 pin operation mode
0	I/O port
1	TXDA0 output

Note The INTP7 pin and RXDA0 pin are alternate-function pins. When using the pin as the RXDA0 pin, disable edge detection for the INTP7 alternate-function pin (clear the INTF3.INTF31 bit and the INTR3.INTR31 bit to 0). When using the pin as the INTP7 pin, stop UARTA0 reception (clear the UA0CTL0.UA0RXE bit to 0).

Remark The PMC3 register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PMC3 register as the PMC3H register and the lower 8 bits as the PMC3L register, they can be read or written in 8-bit or 1-bit units.

(4) Port 3 function control register L (PFC3L)

After reset: 00H R/W Address: FFFFF466H

	7	6	5	4	3	2	1	0
PFC3L	0	0	PFC35	PFC34	PFC33	PFC32	0	0

Remark For details on alternate function specification, refer to **4.3.3 (6) Port 3 alternate function specification**.

(5) Port 3 function control expansion register L (PFCE3L)

After reset: 00H R/W Address: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3L	0	0	0	0	0	PFCE32	0	0

Remark For details on alternate function specification, refer to **4.3.3 (6) Port 3 alternate function specification**.

(6) Port 3 alternate function specification

PFC35	Specification of P35 Pin Alternate Function
0	TIP11 input
1	TOP11 output

PFC34	Specification of P34 Pin Alternate Function
0	TIP10 input
1	TOP10 output

PFC33	Specification of P33 Pin Alternate Function
0	TIP01 input
1	TOP01 output

PFCE32	PFC32	Specification of P32 Pin Alternate Function
0	0	ASCKA0 input
0	1	Setting prohibited
1	0	TIP00 input
1	1	TOP00 output

(7) Pull-up resistor option register 3 (PU3)

After reset: 0000H R/W Address: PU3 FFFFFC46H,
PU3L FFFFFC46H, PU3H FFFFFC47H

	15	14	13	12	11	10	9	8
PU3 (PU3H[Note])	0	0	0	0	0	0	PU39	PU38
	7	6	5	4	3	2	1	0
(PU3L)	PU37	PU36	PU35	PU34	PU33	PU32	PU31	PU30

PU3n	Control of on-chip pull-up resistor connection (n = 0 to 9)
0	Not connected
1	Connected

Note To read/write bits 8 to 15 of the PU3 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PU3H register.

Remark The PU3 register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PU3 register as the PU3H register and the lower 8 bits as the PU3L register, they can be read or written in 8-bit or 1-bit units.

4.3.4 Port 4

Port 4 is a 3-bit port for which I/O settings can be controlled in 1-bit units.

Port 4 includes the following alternate-function pins.

Table 4-7. Port 4 Alternate-Function Pins

Pin Name	Pin No.	Alternate Function	I/O	PULL[Note]	Remark	Block Type
P40	22	SIB0/SDA00	I/O	Provided	–	Q-1
P41	23	SOB0/SCL00	I/O			Q-2
P42	24	$\overline{\text{SCKB0}}$	I/O			E-3

Note Software pull-up function

Caution Pins P40 to P42 have hysteresis characteristics in the input mode of the alternate function, but not in port mode.

(1) Port 4 register (P4)

After reset: 00H (output latch) R/W Address: FFFFF408H

	7	6	5	4	3	2	1	0
P4	0	0	0	0	0	P42	P41	P40

P4n	Control of output data (in output mode) (n = 0 to 2)
0	Outputs 0
1	Outputs 1

(2) Port 4 mode register (PM4)

After reset: FFH R/W Address: FFFFF428H

	7	6	5	4	3	2	1	0
PM4	1	1	1	1	1	PM42	PM41	PM40

PM4n	Control of I/O mode (n = 0 to 2)
0	Output mode
1	Input mode

(3) Port 4 mode control register (PMC4)

After reset: 00H R/W Address: FFFFF448H

	7	6	5	4	3	2	1	0
PMC4	0	0	0	0	0	PMC42	PMC41	PMC40

PMC42	Specification of P42 pin operation mode
0	I/O port
1	$\overline{\text{SCKB0}}$ I/O

PMC41	Specification of P41 pin operation mode
0	I/O port
1	SOB0/SCL00 I/O

PMC40	Specification of P40 pin operation mode
0	I/O port
1	SIB0/SDA00 I/O

(4) Port 4 function control register (PFC4)

After reset: 00H R/W Address: FFFFF468H

	7	6	5	4	3	2	1	0
PFC4	0	0	0	0	0	0	PFC41	PFC40

PFC41	Specification of P41 pin alternate function
0	SOB0 output
1	SCL00 I/O

PFC40	Specification of P40 pin alternate function
0	SIB0 input
1	SDA00 I/O

(5) Port 4 function register (PF4)

After reset: 00H R/W Address: FFFFFC68H

	7	6	5	4	3	2	1	0
PF4	0	0	0	0	0	0	PF41	PF40

PF4n	Control of normal output/N-ch open-drain output (n = 0, 1)
0	Normal output (CMOS output)
1	N-ch open-drain output

(6) Pull-up resistor option register 4 (PU4)

After reset: 00H R/W Address: FFFFFC48H

	7	6	5	4	3	2	1	0
PU4	0	0	0	0	0	PU42	PU41	PU40

PU4n	Control of on-chip pull-up resistor connection (n = 0 to 2)
0	Not connected
1	Connected

4.3.5 Port 5

Port 5 is a 6-bit port for which I/O settings can be controlled in 1-bit units.

Port 5 includes the following alternate-function pins.

Table 4-8. Port 5 Alternate-Function Pins

Pin Name	Pin No.	Alternate Function	I/O	PULL[Note 1]	Remark	Block Type
P50	37	KR0	Input	Provided	Analog noise elimination	U-4
		TIQ01/TOQ01	I/O		Digital noise elimination[Note 2]	U-4
P51	38	KR1	Input		Analog noise elimination	U-4
		TIQ02/TOQ02	I/O		Digital noise elimination[Note 2]	U-4
P52	39	KR2	Input		Analog noise elimination	U-5
		TIQ03/TOQ03	I/O		Digital noise elimination[Note 2]	U-5
		DDI[Note 3]	Input		–	U-5
P53	40	KR3	Input		Analog noise elimination	U-6
		TIQ00/TOQ00	I/O		Digital noise elimination[Note 2]	U-6
		DDO[Note 3]	Output		–	U-6
P54	41	KR4	Input		Analog noise elimination	G-2
		DCK[Note 3]	Input		–	G-2
P55	42	KR5	Input		Analog noise elimination	G-2
		DMS[Note 3]	Input		–	G-2

Notes 1. Software pull-up function

2. For details of digital noise elimination, refer to **7.6 Eliminating Digital Noise on Timer Input Pin**.

3. The DDI, DDO, DCK, and DMS pins are the pins for on-chip debugging (μPD70F3229Y only).
When on-chip debugging is not used, fix the P05/INTP2/$\overline{\text{DRST}}$ pin state to low level after reset is released by the $\overline{\text{RESET}}$ pin until the OCDM.OCDM0 bit is cleared (0).
For details, refer to **4.6.3 Cautions on on-chip debug pins**.

Caution Pins P50 to P55 have hysteresis characteristics in the input mode of the alternate function, but not in port mode.

(1) Port 5 register (P5)

After reset: 00H (output latch) R/W Address: FFFFF40AH

	7	6	5	4	3	2	1	0
P5	0	0	P55	P54	P53	P52	P51	P50

P5n	Control of output data (in output mode) (n = 0 to 5)
0	Outputs 0
1	Outputs 1

(2) Port 5 mode register (PM5)

After reset: FFH R/W Address: FFFFF42AH

	7	6	5	4	3	2	1	0
PM5	1	1	PM55	PM54	PM53	PM52	PM51	PM50

PM5n	Control of I/O mode (n = 0 to 5)
0	Output mode
1	Input mode

(3) Port 5 mode control register (PMC5)

After reset: 00H R/W Address: FFFFF44AH

	7	6	5	4	3	2	1	0
PMC5	0	0	PMC55	PMC54	PMC53	PMC52	PMC51	PMC50

PMC55	Specification of P55 pin operation mode
0	I/O port
1	KR5 input

PMC54	Specification of P54 pin operation mode
0	I/O port
1	KR4 input

PMC53	Specification of P53 pin operation mode
0	I/O port
1	KR3/TIQ00/TOQ00 I/O

PMC52	Specification of P52 pin operation mode
0	I/O port
1	KR2/TIQ03/TOQ03 I/O

PMC51	Specification of P51 pin operation mode
0	I/O port
1	KR1/TIQ02/TOQ02 I/O

PMC50	Specification of P50 pin operation mode
0	I/O port
1	KR0/TIQ01/TOQ01 I/O

(4) Port 5 function control register (PFC5)

Caution If an alternate function is set using the PMC5.PMC5n bit with the PFC5n bit and the PFCE5.PFCE5m bit set to the initial value (0), this register's output is undefined. To set an alternate function, set the PFC5n and PFCE5m bits and then set the PMC5n bit to 1 (n = 0 to 5, m = 0 to 3).

After reset: 00H R/W Address: FFFFF46AH

	7	6	5	4	3	2	1	0
PFC5	0	0	PFC55	PFC54	PFC53	PFC52	PFC51	PFC50

Remark For details on alternate function specification, refer to **4.3.5 (6) Port 5 alternate function specification**.

(5) Port 5 function control expansion register (PFCE5)

Caution If an alternate function is set using the PMC5.PMC5n bit with the PFC5.PFC5n bit and the PFCE5m bit set to the initial value (0), this register's output is undefined. To set an alternate function, set the PFC5n and PFCE5m bits and then set the PMC5n bit to 1 (n = 0 to 5, m = 0 to 3).

After reset: 00H R/W Address: FFFFF70AH

	7	6	5	4	3	2	1	0
PFCE5	0	0	0	0	PFCE53	PFCE52	PFCE51	PFCE50

Remark For details on alternate function specification, refer to **4.3.5 (6) Port 5 alternate function specification**.

(6) Port 5 alternate function specification

PFC55	Specification of P55 Pin Alternate Function
0	Setting prohibited
1	KR5 input

PFC54	Specification of P54 Pin Alternate Function
0	Setting prohibited
1	KR4 input

PFCE53	PFC53	Specification of P53 Pin Alternate Function
0	0	Setting prohibited
0	1	TIQ00/KR3[Note] input
1	0	TOQ00 output
1	1	Setting prohibited

PFCE52	PFC52	Specification of P52 Pin Alternate Function
0	0	Setting prohibited
0	1	TIQ03/KR2[Note] input
1	0	TOQ03 output
1	1	Setting prohibited

PFCE51	PFC51	Specification of P51 Pin Alternate Function
0	0	Setting prohibited
0	1	TIQ02/KR1[Note] input
1	0	TOQ02 output
1	1	Setting prohibited

PFCE50	PFC50	Specification of P50 Pin Alternate Function
0	0	Setting prohibited
0	1	TIQ01/KR0[Note] input
1	0	TOQ01 output
1	1	Setting prohibited

Note The KRn pin and TIQ0m pin are alternate-function pins. When using the pin as the TIQ0m pin, disable KRn pin key return detection, which is the alternate function (clear the corresponding bit in the KRM register to 0). Also, when using the pin as the KRn pin, disable TIQ0m pin edge detection, which is the alternate function (n = 0 to 3, m = 0 to 3).

Pin Name	Use as TIQ0m Pin	Use as KRn Pin
KR0/TIQ01	KRM.KRM0 bit = 0	TQ0IOC1.TQ0TIG2, TQ0IOC1.TQ0TIG3 bits = 0
KR1/TIQ02	KRM.KRM1 bit = 0	TQ0IOC1.TQ0TIG4, TQ0IOC1.TQ0TIG5 bits = 0
KR2/TIQ03	KRM.KRM2 bit = 0	TQ0IOC1.TQ0TIG6, TQ0IOC1.TQ0TIG7 bits = 0
KR3/TIQ00	KRM.KRM3 bit = 0	TQ0IOC1.TQ0TIG0, TQ0IOC1.TQ0TIG1 bits = 0 TQ01OC2.TQ0EES0, TQ01OC2.TQ0EES1 bits = 0 TQ01OC2.TQ0ETS0, TQ01OC2.TQ0ETS1 bits = 0

(7) Pull-up resistor option register 5 (PU5)

After reset: 00H R/W Address: FFFFFC4AH

	7	6	5	4	3	2	1	0
PU5	0	0	PU55	PU54	PU53	PU52	PU51	PU50

PU5n	Control of on-chip pull-up resistor connection (n = 0 to 5)
0	Not connected
1	Connected

4.3.6 Port 7

Port 7 is a 12-bit port for which I/O settings can be controlled in 1-bit units.

Port 7 includes the following alternate-function pins.

Table 4-9. Port 7 Alternate-Function Pins

Pin Name	Pin No.	Alternate Function	I/O	PULL[Note]	Remark	Block Type
P70	100	ANI0	Input	None	–	A-1
P71	99	ANI1	Input			A-1
P72	98	ANI2	Input			A-1
P73	97	ANI3	Input			A-1
P74	96	ANI4	Input			A-1
P75	95	ANI5	Input			A-1
P76	94	ANI6	Input			A-1
P77	93	ANI7	Input			A-1
P78	92	ANI8	Input			A-1
P79	91	ANI9	Input			A-1
P710	90	ANI10	Input			A-1
P711	89	ANI11	Input			A-1

Note Software pull-up function

(1) Port 7 register H, port 7 register L (P7H, P7L)

After reset: Undefined R/W Address: P7H FFFFF40EH, P7L FFFFF40FH

	7	6	5	4	3	2	1	0
P7H	0	0	0	0	P711	P710	P79	P78

	7	6	5	4	3	2	1	0
P7L	P77	P76	P75	P74	P73	P72	P71	P70

P7n	Control of output data (in output mode) (n = 0 to 11)
0	Outputs 0
1	Outputs 1

Caution Do not read the P7H and P7L registers during A/D conversion.

(2) Port 7 mode register H, port 7 mode register L (PM7H, PM7L)

After reset: FFH R/W Address: PM7H FFFFF42EH, PM7L FFFFF42FH

	7	6	5	4	3	2	1	0
PM7H	1	1	1	1	PM711	PM710	PM79	PM78

	7	6	5	4	3	2	1	0
PM7L	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	Control of I/O mode (n = 0 to 11)
0	Output mode
1	Input mode

Caution To use the alternate function (ANIn pin) of the P7n pin, set the PM7n bit to 1.

4.3.7 Port 9

Port 9 is a 16-bit port for which I/O settings can be controlled in 1-bit units.

Port 9 includes the following alternate-function pins.

Table 4-10. Port 9 Alternate-Function Pins

Pin Name	Pin No.	Alternate Function	I/O	PULL[Note 1]	Remark	Block Type
P90	43	KR6	Input	Provided	Analog noise elimination	U-12
		TXDA1	Output		–	U-12
P91	44	KR7	Input		Analog noise elimination	U-7
		RXDA1	Input		–	U-7
P92	45	–	–			C-1
P93	46	–	–			C-1
P94	47	–	–			C-1
P95	48	–	–			C-1
P96	49	TIP21/TOP21	I/O		Digital noise elimination[Note 2]	U-9
P97	50	SIB1	Input		–	U-8
		TIP20/TOP20	I/O		Digital noise elimination[Note 2]	U-8
P98	51	SOB1	Output		–	G-3
P99	52	$\overline{\text{SCKB1}}$	I/O			G-5
P910	53	–	–			C-1
P911	54	–	–			C-1
P912	55	–	–			C-1
P913	56	INTP4	Input		Analog noise elimination	N-2
P914	57	INTP5	Input			N-2
P915	58	INTP6	Input			N-2

Notes 1. Software pull-up function

2. For details of digital noise elimination, refer to **7.6 Eliminating Digital Noise on Timer Input Pin**.

Caution Pins P90, P91, P96, P97, P99, P913 to P915 have hysteresis characteristics in the input mode of the alternate function, but not in port mode.

(1) Port 9 register (P9)

After reset: 0000H (output latch) R/W Address: P9 FFFFF412H,
P9L FFFFF412H, P9H FFFFF413H

	15	14	13	12	11	10	9	8
P9 (P9H[Note])	P915	P914	P913	P912	P911	P910	P99	P98

	7	6	5	4	3	2	1	0
(P9L)	P97	P96	P95	P94	P93	P92	P91	P90

P9n	Control of output data (in output mode) (n = 0 to 15)
0	Outputs 0
1	Outputs 1

Note To read/write bits 8 to 15 of the P9 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the P9H register.

Remark The P9 register can be read or written in 16-bit units.
However, when using the higher 8 bits of the P9 register as the P9H register and the lower 8 bits as the P9L register, they can be read or written in 8-bit or 1-bit units.

(2) Port 9 mode register (PM9)

After reset: FFFFH R/W Address: PM9 FFFFF432H,
PM9L FFFFF432H, PM9H FFFFF433H

	15	14	13	12	11	10	9	8
PM9 (PM9H[Note])	PM915	PM914	PM913	PM912	PM911	PM910	PM99	PM98

	7	6	5	4	3	2	1	0
(PM9L)	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90

PM9n	Control of I/O mode (n = 0 to 15)
0	Output mode
1	Input mode

Note To read/write bits 8 to 15 of the PM9 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PM9H register.

Remark The PM9 register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PM9 register as the PM9H register and the lower 8 bits as the PM9L register, they can be read or written in 8-bit or 1-bit units.

(3) Port 9 mode control register (PMC9)

After reset: 0000H R/W Address: PMC9 FFFFF452H,
PMC9L FFFFF452H, PMC9H FFFFF453H

	15	14	13	12	11	10	9	8
PMC9 (PMC9H[Note])	PMC915	PMC914	PMC913	0	0	0	PMC99	PMC98

	7	6	5	4	3	2	1	0
(PMC9L)	PMC97	PMC96	0	0	0	0	PMC91	PMC90

PMC915	Specification of P915 pin operation mode
0	I/O port
1	INTP6 input

PMC914	Specification of P914 pin operation mode
0	I/O port
1	INTP5 input

PMC913	Specification of P913 pin operation mode
0	I/O port
1	INTP4 input

PMC99	Specification of P99 pin operation mode
0	I/O port
1	$\overline{\text{SCKB1}}$ I/O

PMC98	Specification of P98 pin operation mode
0	I/O port
1	SOB1 output

PMC97	Specification of P97 pin operation mode
0	I/O port
1	SIB1/TIP20/TOP20 I/O

PMC96	Specification of P96 pin operation mode
0	I/O port
1	TIP21/TOP21 I/O

PMC91	Specification of P91 pin operation mode
0	I/O port
1	KR7/RXDA1 input

PMC90	Specification of P90 pin operation mode
0	I/O port
1	KR6/TXDA1 I/O

Note To read/write bits 8 to 15 of the PMC9 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PMC9H register.

Remark The PMC9 register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PMC9 register as the PMC9H register and the lower 8 bits as the PMC9L register, they can be read or written in 8-bit or 1-bit units.

(4) Port 9 function control register (PFC9)

After reset: 0000H R/W Address: PFC9 FFFFF472H,
PFC9L FFFFF472H, PFC9H FFFFF473H

	15	14	13	12	11	10	9	8
PFC9 (PFC9H[Note])	PFC915	PFC914	PFC913	0	0	0	PFC99	PFC98

	7	6	5	4	3	2	1	0
(PFC9L)	PFC97	PFC96	0	0	0	0	PFC91	PFC90

Note To read/write bits 8 to 15 of the PFC9 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PFC9H register.

Remarks 1. For details on alternate function specification, refer to **4.3.7 (6) Port 9 alternate function specification**.

2. The PFC9 register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PFC9 register as the PFC9H register and the lower 8 bits as the PFC9L register, they can be read or written in 8-bit or 1-bit units.

(5) Port 9 function control expansion register L (PFCE9L)

After reset: 00H R/W Address: FFFFF712H

	7	6	5	4	3	2	1	0
PFCE9L	PFCE97	PFCE96	0	0	0	0	PFCE91	PFCE90

Remark For details on alternate function specification, refer to **4.3.7 (6) Port 9 alternate function specification**.

(6) Port 9 alternate function specification

PFC915	Specification of P915 Pin Alternate Function
0	Setting prohibited
1	INTP6 input

PFC914	Specification of P914 Pin Alternate Function
0	Setting prohibited
1	INTP5 input

PFC913	Specification of P913 Pin Alternate Function
0	Setting prohibited
1	INTP4 input

PFC99	Specification of P99 Pin Alternate Function
0	Setting prohibited
1	$\overline{\text{SCKB1}}$ I/O

PFC98	Specification of P98 Pin Alternate Function
0	Setting prohibited
1	SOB1 output

PFCE97	PFC97	Specification of P97 Pin Alternate Function
0	0	Setting prohibited
0	1	SIB1 input
1	0	TIP20 input
1	1	TOP20 output

PFCE96	PFC96	Specification of P96 Pin Alternate Function
0	0	Setting prohibited
0	1	Setting prohibited
1	0	TIP21 input
1	1	TOP21 output

PFCE91	PFC91	Specification of P91 Pin Alternate Function
0	0	Setting prohibited
0	1	KR7 input
1	0	RXDA1 input/KR7 input
1	1	Setting prohibited

PFCE90	PFC90	Specification of P90 Pin Alternate Function
0	0	Setting prohibited
0	1	KR6 input
1	0	TXDA1 output
1	1	Setting prohibited

(7) Pull-up resistor option register 9 (PU9)

After reset: 0000H R/W Address: PU9 FFFFFC52H,
PU9L FFFFFC52H, PU9H FFFFFC53H

	15	14	13	12	11	10	9	8
PU9 (PU9H[Note])	PU915	PU914	PU913	PU912	PU911	PU910	PU99	PU98

	7	6	5	4	3	2	1	0
(PU9L)	PU97	PU96	PU95	PU94	PU93	PU92	PU91	PU90

PU9n	Control of on-chip pull-up resistor connection (n = 0 to 15)
0	Not connected
1	Connected

Note To read/write bits 8 to 15 of the PU9 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PU9H register.

Remark The PU9 register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PU9 register as the PU9H register and the lower 8 bits as the PU9L register, they can be read or written in 8-bit or 1-bit units.

4.3.8 Port CM

Port CM is a 4-bit port for which I/O settings can be controlled in 1-bit units.

Port CM includes the following alternate-function pins.

Table 4-11. Port CM Alternate-Function Pins

Pin Name	Pin No.	Alternate Function	I/O	PULL[Note]	Remark	Block Type
PCM0	61	$\overline{\text{WAIT}}$	Input	None	–	D-1
PCM1	62	CLKOUT	Output			D-2
PCM2	63	$\overline{\text{HLDAK}}$	Output			D-2
PCM3	64	$\overline{\text{HLDRQ}}$	Input			D-1

Note Software pull-up function

(1) Port CM register (PCM)

After reset: 00H (output latch) R/W Address: FFFFF00CH

	7	6	5	4	3	2	1	0
PCM	0	0	0	0	PCM3	PCM2	PCM1	PCM0

PCMn	Control of output data (in output mode) (n = 0 to 3)
0	Outputs 0
1	Outputs 1

(2) Port CM mode register (PMCM)

After reset: FFH R/W Address: FFFFF02CH

	7	6	5	4	3	2	1	0
PMCM	1	1	1	1	PMCM3	PMCM2	PMCM1	PMCM0

PMCMn	Control of I/O mode (n = 0 to 3)
0	Output mode
1	Input mode

(3) Port CM mode control register (PMCCM)

After reset: 00H R/W Address: FFFFF04CH

	7	6	5	4	3	2	1	0
PMCCM	0	0	0	0	PMCCM3	PMCCM2	PMCCM1	PMCCM0

PMCCM3	Specification of PCM3 pin operation mode
0	I/O port
1	$\overline{\text{HLDRQ}}$ input

PMCCM2	Specification of PCM2 pin operation mode
0	I/O port
1	$\overline{\text{HLDAK}}$ output

PMCCM1	Specification of PCM1 pin operation mode
0	I/O port
1	CLKOUT output

PMCCM0	Specification of PCM0 pin operation mode
0	I/O port
1	$\overline{\text{WAIT}}$ input

4.3.9 Port CS

Port CS is a 2-bit port for which I/O settings can be controlled in 1-bit units.

Port CS includes the following alternate-function pins.

Table 4-12. Port CS Alternate-Function Pins

Pin Name	Pin No.	Alternate Function	I/O	PULL[Note]	Remark	Block Type
PCS0	59	$\overline{CS0}$	Output	None	–	D-2
PCS1	60	$\overline{CS1}$	Output			D-2

Note Software pull-up function

(1) Port CS register (PCS)

After reset: 00H (output latch) R/W Address: FFFFF008H

	7	6	5	4	3	2	1	0
PCS	0	0	0	0	0	0	PCS1	PCS0

PCSn	Control of output data (in output mode) (n = 0, 1)
0	Outputs 0
1	Outputs 1

(2) Port CS mode register (PMCS)

After reset: FFH R/W Address: FFFFF028H

	7	6	5	4	3	2	1	0
PMCS	1	1	1	1	1	1	PMCS1	PMCS0

PMCSn	Control of I/O mode (n = 0, 1)
0	Output mode
1	Input mode

(3) Port CS mode control register (PMCCS)

After reset: 00H R/W Address: FFFFF048H

	7	6	5	4	3	2	1	0
PMCCS	0	0	0	0	0	0	PMCCS1	PMCCS0

PMCCSn	Specification of PCSn pin operation mode (n = 0, 1)
0	I/O port
1	$\overline{\text{CSn}}$ output

4.3.10 Port CT

Port CT is a 4-bit port for which I/O settings can be controlled in 1-bit units.

Port CT includes the following alternate-function pins.

Table 4-13. Port CT Alternate-Function Pins

Pin Name	Pin No.	Alternate Function	I/O	PULL[Note]	Remark	Block Type
PCT0	65	$\overline{WR0}$	Output	None	–	D-2
PCT1	66	$\overline{WR1}$	Output			D-2
PCT4	67	$\overline{RD}$	Output			D-2
PCT6	68	ASTB	Output			D-2

Note Software pull-up function

(1) Port CT register (PCT)

After reset: 00H (output latch) R/W Address: FFFFF00AH

	7	6	5	4	3	2	1	0
PCT	0	PCT6	0	PCT4	0	0	PCT1	PCT0

PCTn	Control of output data (in output mode) (n = 0, 1, 4, 6)
0	Outputs 0
1	Outputs 1

(2) Port CT mode register (PMCT)

After reset: FFH R/W Address: FFFFF02AH

	7	6	5	4	3	2	1	0
PMCT	1	PMCT6	1	PMCT4	1	1	PMCT1	PMCT0

PMCTn	Control of I/O mode (n = 0, 1, 4, 6)
0	Output mode
1	Input mode

(3) Port CT mode control register (PMCCT)

After reset: 00H R/W Address: FFFFF04AH

	7	6	5	4	3	2	1	0
PMCCT	0	PMCCT6	0	PMCCT4	0	0	PMCCT1	PMCCT0

PMCCT6	Specification of PCT6 pin operation mode
0	I/O port
1	ASTB output

PMCCT4	Specification of PCT4 pin operation mode
0	I/O port
1	$\overline{RD}$ output

PMCCT1	Specification of PCT1 pin operation mode
0	I/O port
1	$\overline{WR1}$ output

PMCCT0	Specification of PCT0 pin operation mode
0	I/O port
1	$\overline{WR0}$ output

4.3.11 Port DH

Port DH is a 2-bit port for which I/O settings can be controlled in 1-bit units.

Port DH includes the following alternate-function pins.

Table 4-14. Port DH Alternate-Function Pins

Pin Name	Pin No.	Alternate Function	I/O	PULL[Note]	Remark	Block Type
PDH0	87	A16	Output	None	–	D-2
PDH1	88	A17	Output			D-2

Note Software pull-up function

(1) Port DH register (PDH)

After reset: 00H (output latch) R/W Address: FFFFF006H

	7	6	5	4	3	2	1	0
PDH	0	0	0	0	0	0	PDH1	PDH0

PDHn	Control of output data (in output mode) (n = 0, 1)
0	Outputs 0
1	Outputs 1

(2) Port DH mode register (PMDH)

After reset: FFH R/W Address: FFFFF026H

	7	6	5	4	3	2	1	0
PMDH	1	1	1	1	1	1	PMDH1	PMDH0

PMDHn	Control of I/O mode (n = 0, 1)
0	Output mode
1	Input mode

(3) Port DH mode control register (PMCDH)

After reset: 00H R/W Address: FFFFF046H

	7	6	5	4	3	2	1	0
PMCDH	0	0	0	0	0	0	PMCDH1	PMCDH0

PMCDHn	Specification of PDHn pin operation mode (n = 0, 1)
0	I/O port
1	Am output (address bus output) (m = 16, 17)

4.3.12 Port DL

Port DL is a 16-bit port for which I/O settings can be controlled in 1-bit units.

Port DL includes the following alternate-function pins.

Table 4-15. Port DL Alternate-Function Pins

Pin Name	Pin No.	Alternate Function	I/O	PULL[Note 1]	Remark	Block Type
PDL0	71	AD0	I/O	None	–	D-3
PDL1	72	AD1	I/O			D-3
PDL2	73	AD2	I/O			D-3
PDL3	74	AD3	I/O			D-3
PDL4	75	AD4	I/O			D-3
PDL5	76	AD5/FLMD1[Note 2]	I/O			D-3
PDL6	77	AD6	I/O			D-3
PDL7	78	AD7	I/O			D-3
PDL8	79	AD8	I/O			D-3
PDL9	80	AD9	I/O			D-3
PDL10	81	AD10	I/O			D-3
PDL11	82	AD11	I/O			D-3
PDL12	83	AD12	I/O			D-3
PDL13	84	AD13	I/O			D-3
PDL14	85	AD14	I/O			D-3
PDL15	86	AD15	I/O			D-3

Notes 1. Software pull-up function

2. Since this pin is set in the flash memory programming mode, it need not be manipulated with the port control register. For details, refer to **CHAPTER 23 FLASH MEMORY**.

(1) Port DL register (PDL)

After reset: 0000H (output latch) R/W Address: PDL FFFFF004H,
PDLL FFFFF004H, PDLH FFFFF005H

	15	14	13	12	11	10	9	8
PDL (PDLH[Note])	PDL15	PDL14	PDL13	PDL12	PDL11	PDL10	PDL9	PDL8

	7	6	5	4	3	2	1	0
(PDLL)	PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0

PDLn	Control of output data (in output mode) (n = 0 to 15)
0	Outputs 0
1	Outputs 1

Note To read/write bits 8 to 15 of the PDL register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PDLH register.

Remark The PDL register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PDL register as the PDLH register and the lower 8 bits as the PDLL register, they can be read or written in 8-bit or 1-bit units.

(2) Port DL mode register (PMDL)

After reset: FFFFH R/W Address: PMDL FFFFF024H,
PMDLL FFFFF024H, PMDLH FFFFF025H

	15	14	13	12	11	10	9	8
PMDL (PMDLH[Note])	PMDL15	PMDL14	PMDL13	PMDL12	PMDL11	PMDL10	PMDL9	PMDL8

	7	6	5	4	3	2	1	0
(PMDLL)	PMDL7	PMDL6	PMDL5	PMDL4	PMDL3	PMDL2	PMDL1	PMDL0

PMDLn	Control of I/O mode (n = 0 to 15)
0	Output mode
1	Input mode

Note To read/write bits 8 to 15 of the PMDL register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PMDLH register.

Remark The PMDL register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PMDL register as the PMDLH register and the lower 8 bits as the PMDLL register, they can be read or written in 8-bit or 1-bit units.

(3) Port DL mode control register (PMCDL)

After reset: 0000H R/W Address: PMCDL FFFFF044H,
PMCDLL FFFFF044H, PMCDLH FFFFF045H

	15	14	13	12	11	10	9	8
PMCDL (PMCDLH[Note])	PMCDL15	PMCDL14	PMCDL13	PMCDL12	PMCDL11	PMCDL10	PMCDL9	PMCDL8

	7	6	5	4	3	2	1	0
(PMCDLL)	PMCDL7	PMCDL6	PMCDL5	PMCDL4	PMCDL3	PMCDL2	PMCDL1	PMCDL0

PMCDLn	Specification of PDLn pin operation mode (n = 0 to 15)
0	I/O port
1	ADn I/O (address/data bus I/O)

Note To read/write bits 8 to 15 of the PMCDL register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PMCDLH register.

Remark The PMCDL register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PMCDL register as the PMCDLH register and the lower 8 bits as the PMCDLL register, they can be read or written in 8-bit or 1-bit units.

★ 4.4 Block Diagrams

Figure 4-3. Block Diagram of Type A-1

Figure 4-4. Block Diagram of Type C-1

Figure 4-5. Block Diagram of Type D-1

Figure 4-6. Block Diagram of Type D-2

Figure 4-7. Block Diagram of Type D-3

Figure 4-8. Block Diagram of Type E-1

Note Hysteresis characteristics are not available in port mode.

Figure 4-9. Block Diagram of Type E-2

Figure 4-10. Block Diagram of Type E-3

Note Hysteresis characteristics are not available in port mode.

Figure 4-11. Block Diagram of Type G-1

Note Hysteresis characteristics are not available in port mode.

Figure 4-12. Block Diagram of Type G-2

Note Hysteresis characteristics are not available in port mode.

Figure 4-13. Block Diagram of Type G-3

Figure 4-14. Block Diagram of Type G-5

Note Hysteresis characteristics are not available in port mode.

Figure 4-15. Block Diagram of Type L-1

Notes 1. Refer to **17.4 External Interrupt Request Input Pins (NMI and INTP0 to INTP7)**.

2. Hysteresis characteristics are not available in port mode.

Figure 4-16. Block Diagram of Type L-2

Notes 1. Refer to **17.4 External Interrupt Request Input Pins (NMI and INTP0 to INTP7)**.

2. Hysteresis characteristics are not available in port mode.

Figure 4-17. Block Diagram of Type N-1

Notes 1. Refer to **17.4 External Interrupt Request Input Pins (NMI and INTP0 to INTP7)**.

2. Hysteresis characteristics are not available in port mode.

Figure 4-18. Block Diagram of Type N-2

Notes 1. Refer to **17.4 External Interrupt Request Input Pins (NMI and INTP0 to INTP7)**.

2. Hysteresis characteristics are not available in port mode.

Figure 4-19. Block Diagram of Type Q-1

Note Hysteresis characteristics are not available in port mode.

Figure 4-20. Block Diagram of Type Q-2

Note Hysteresis characteristics are not available in port mode.

Figure 4-21. Block Diagram of Type U-1

Note Hysteresis characteristics are not available in port mode.

Figure 4-22. Block Diagram of Type U-4

Note Hysteresis characteristics are not available in port mode.

Figure 4-23. Block Diagram of Type U-5

Note Hysteresis characteristics are not available in port mode.

Figure 4-24. Block Diagram of Type U-6

Note Hysteresis characteristics are not available in port mode.

Figure 4-25. Block Diagram of Type U-7

Note Hysteresis characteristics are not available in port mode.

Figure 4-26. Block Diagram of Type U-8

Note Hysteresis characteristics are not available in port mode.

Figure 4-27. Block Diagram of Type U-9

Note Hysteresis characteristics are not available in port mode.

Figure 4-28. Block Diagram of Type U-12

Note Hysteresis characteristics are not available in port mode.

Figure 4-29. Block Diagram of Type AA-1

Notes 1. Refer to **17.4 External Interrupt Request Input Pins (NMI and INTP0 to INTP7)**.

2. Hysteresis characteristics are not available in port mode.

4.5 Port Register Settings When Alternate Function Is Used

Table 4-16 shows the port register settings when each port is used for an alternate function. When using a port pin as an alternate-function pin, refer to the description of each pin.

Table 4-16. Using Port Pin as Alternate-Function Pin (1/5)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register	Other Bits (Registers)
	Name	I/O						
P00	TIP31	Input	P00 = Setting not required	PM00 = Setting not required	PMC00 = 1	–	PFC00 = 0	
	TOP31	Output	P00 = Setting not required	PM00 = Setting not required	PMC00 = 1	–	PFC00 = 1	
P01	TIP30	Input	P01 = Setting not required	PM01 = Setting not required	PMC01 = 1	–	PFC01 = 0	
	TOP30	Output	P01 = Setting not required	PM01 = Setting not required	PMC01 = 1	–	PFC01 = 1	
P02	NMI	Input	P02 = Setting not required	PM02 = Setting not required	PMC02 = 1	–	–	
P03	INTP0	Input	P03 = Setting not required	PM03 = Setting not required	PMC03 = 1	–	PFC03 = 0	
	ADTRG	Input	P03 = Setting not required	PM03 = Setting not required	PMC03 = 1	–	PFC03 = 1	
P04	INTP1	Input	P04 = Setting not required	PM04 = Setting not required	PMC04 = 1	–	–	
P05	INTP2	Input	P05 = Setting not required	PM05 = Setting not required	PMC05 = 1	–	–	
	$\overline{\text{DRST}}$ Note 1	Input	P05 = Setting not required	PM05 = Setting not required	PMC05 = 1	–	–	OCDM0 (OCDM) = 1
P06	INTP3	Input	P06 = Setting not required	PM06 = Setting not required	PMC06 = 1	–	–	
P30	TXDA0	Output	P30 = Setting not required	PM30 = Setting not required	PMC30 = 1	–	–	
P31	RXDA0	Input	P31 = Setting not required	PM31 = Setting not required	PMC31 = 1	–	–	**Note 2**
	INTP7	Input	P31 = Setting not required	PM31 = Setting not required	PMC31 = 1	–	–	**Note 2**
P32	ASCKA0	Input	P32 = Setting not required	PM32 = Setting not required	PMC32 = 1	PFCE32 = 0	PFC32 = 0	
	TIP00	Input	P32 = Setting not required	PM32 = Setting not required	PMC32 = 1	PFCE32 = 1	PFC32 = 0	
	TOP00	Output	P32 = Setting not required	PM32 = Setting not required	PMC32 = 1	PFCE32 = 1	PFC32 = 1	
P33	TIP01	Input	P33 = Setting not required	PM33 = Setting not required	PMC33 = 1	–	PFC33 = 0	
	TOP01	Output	P33 = Setting not required	PM33 = Setting not required	PMC33 = 1	–	PFC33 = 1	
P34	TIP10	Input	P34 = Setting not required	PM34 = Setting not required	PMC34 = 1	–	PFC34 = 0	
	TOP10	Output	P34 = Setting not required	PM34 = Setting not required	PMC34 = 1	–	PFC34 = 1	
P35	TIP11	Input	P35 = Setting not required	PM35 = Setting not required	PMC35 = 1	–	PFC35 = 0	
	TOP11	Output	P35 = Setting not required	PM35 = Setting not required	PMC35 = 1	–	PFC35 = 1	
P38	TXDA2	Output	P38 = Setting not required	PM38 = Setting not required	PMC38 = 1	–	–	
P39	RXDA2	Input	P39 = Setting not required	PM39 = Setting not required	PMC39 = 1	–	–	

Notes 1. μPD70F3229Y only

2. The INTP7 pin and RXDA0 pin are alternate-function pins. When using the pin as the RXDA0 pin, disable edge detection for the INTP7 alternate-function pin (clear the INTF3.INTF31 and INTR3.INTR31 bits to 0). When using the pin as the INTP7 pin, stop UARTA0 reception (clear the UA0CTL0.UA0RXE bit to 0).

Table 4-16. Using Port Pin as Alternate-Function Pin (2/5)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register	Other Bits (Registers)
	Name	I/O						
P40	SIB0	Input	P40 = Setting not required	PM40 = Setting not required	PMC40 = 1	–	PFC40 = 0	
	SDA00	I/O	P40 = Setting not required	PM40 = Setting not required	PMC40 = 1	–	PFC40 = 1	
P41	SOB0	Output	P41 = Setting not required	PM41 = Setting not required	PMC41 = 1	–	PFC41 = 0	
	SCL00	I/O	P41 = Setting not required	PM41 = Setting not required	PMC41 = 1	–	PFC41 = 1	
P42	$\overline{\text{SCKB0}}$	I/O	P42 = Setting not required	PM42 = Setting not required	PMC42 = 1	–	–	
P50	KR0	Input	P50 = Setting not required	PM50 = Setting not required	PMC50 = 1	PFCE50 = 0	PFC50 = 1	TQ0TIG2, TQ0TIG3 (TQ0IOC1) = 0
	TIQ01	Input	P50 = Setting not required	PM50 = Setting not required	PMC50 = 1	PFCE50 = 0	PFC50 = 1	KRM0 (KRM) = 0
	TOQ01	Output	P50 = Setting not required	PM50 = Setting not required	PMC50 = 1	PFCE50 = 1	PFC50 = 0	
P51	KR1	Input	P51 = Setting not required	PM51 = Setting not required	PMC51 = 1	PFCE51 = 0	PFC51 = 1	TQ0TIG4, TQ0TIG5 (TQ0IOC1) = 0
	TIQ02	Input	P51 = Setting not required	PM51 = Setting not required	PMC51 = 1	PFCE51 = 0	PFC51 = 1	KRM1 (KRM) = 0
	TOQ02	Output	P51 = Setting not required	PM51 = Setting not required	PMC51 = 1	PFCE51 = 1	PFC51 = 0	
P52	KR2	Input	P52 = Setting not required	PM52 = Setting not required	PMC52 = 1	PFCE52 = 0	PFC52 = 1	TQ0TIG6, TQ0TIG7 (TQ0IOC1) = 0
	TIQ03	Input	P52 = Setting not required	PM52 = Setting not required	PMC52 = 1	PFCE52 = 0	PFC52 = 1	KRM2 (KRM) = 0
	TOQ03	Output	P52 = Setting not required	PM52 = Setting not required	PMC52 = 1	PFCE52 = 1	PFC52 = 0	
	DDI[Note]	Input	P52 = Setting not required	PM52 = Setting not required	PMC52 = Setting not required	–	–	OCDM0 (OCDM) = 1
P53	KR3	Input	P53 = Setting not required	PM53 = Setting not required	PMC53 = 1	PFCE53 = 0	PFC53 = 1	TQ0TIG0, TQ0TIG1 (TQ0IOC1) = 0, TQ0EES0, TQ0EES1 (TQ0IOC2) = 0, TQ0ETS0, TQ0ETS1 (TQ0IOC2) = 0
	TIQ00	Input	P53 = Setting not required	PM53 = Setting not required	PMC53 = 1	PFCE53 = 0	PFC53 = 1	KRM3 (KRM) = 0
	TOQ00	Output	P53 = Setting not required	PM53 = Setting not required	PMC53 = 1	PFCE53 = 1	PFC53 = 0	KRM3 (KRM) = 0
	DDO[Note]	Output	P53 = Setting not required	PM53 = Setting not required	PMC53 = Setting not required	–	–	OCDM0 (OCDM) = 1
P54	KR4	Input	P54 = Setting not required	PM54 = Setting not required	PMC54 = 1	–	PFC54 = 1	
	DCK[Note]	Input	P54 = Setting not required	PM54 = Setting not required	PMC54 = Setting not required	–	–	OCDM0 (OCDM) = 1
P55	KR5	Input	P55 = Setting not required	PM55 = Setting not required	PMC55 = 1	–	PFC55 = 1	
	DMS[Note]	Input	P55 = Setting not required	PM55 = Setting not required	PMC55 = Setting not required	–	–	OCDM0 (OCDM) = 1

Note μPD70F3229Y only

Table 4-16. Using Port Pin as Alternate-Function Pin (3/5)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register	Other Bits (Registers)
	Name	I/O						
P70	ANI0	Input	P70 = Setting not required	PM70 = 1	–	–	–	
P71	ANI1	Input	P71 = Setting not required	PM71 = 1	–	–	–	
P72	ANI2	Input	P72 = Setting not required	PM72 = 1	–	–	–	
P73	ANI3	Input	P73 = Setting not required	PM73 = 1	–	–	–	
P74	ANI4	Input	P74 = Setting not required	PM74 = 1	–	–	–	
P75	ANI5	Input	P75 = Setting not required	PM75 = 1	–	–	–	
P76	ANI6	Input	P76 = Setting not required	PM76 = 1	–	–	–	
P77	ANI7	Input	P77 = Setting not required	PM77 = 1	–	–	–	
P78	ANI8	Input	P78 = Setting not required	PM78 = 1	–	–	–	
P79	ANI9	Input	P79 = Setting not required	PM79 = 1	–	–	–	
P710	ANI10	Input	P710 = Setting not required	PM710 = 1	–	–	–	
P711	ANI11	Input	P711 = Setting not required	PM711 = 1	–	–	–	
P90	KR6	Input	P90 = Setting not required	PM90 = Setting not required	PMC90 = 1	PFCE90 = 0	PFC90 = 1	
	TXDA1	Output	P90 = Setting not required	PM90 = Setting not required	PMC90 = 1	PFCE90 = 1	PFC90 = 0	
P91	KR7	Input	P91 = Setting not required	PM91 = Setting not required	PMC91 = 1	PFCE91 = 0	PFC91 = 1	
	RXDA1/KR7	Input	P91 = Setting not required	PM91 = Setting not required	PMC91 = 1	PFCE91 = 1	PFC91 = 0	
P96	TIP21	Input	P96 = Setting not required	PM96 = Setting not required	PMC96 = 1	PFCE96 = 1	PFC96 = 0	
	TOP21	Output	P96 = Setting not required	PM96 = Setting not required	PMC96 = 1	PFCE96 = 1	PFC96 = 1	
P97	SIB1	Input	P97 = Setting not required	PM97 = Setting not required	PMC97 = 1	PFCE97 = 0	PFC97 = 1	
	TIP20	Input	P97 = Setting not required	PM97 = Setting not required	PMC97 = 1	PFCE97 = 1	PFC97 = 0	
	TOP20	Output	P97 = Setting not required	PM97 = Setting not required	PMC97 = 1	PFCE97 = 1	PFC97 = 1	
P98	SOB1	Output	P98 = Setting not required	PM98 = Setting not required	PMC98 = 1	–	PFC98 = 1	
P99	$\overline{\text{SCKB1}}$	I/O	P99 = Setting not required	PM99 = Setting not required	PMC99 = 1	–	PFC99 = 1	
P913	INTP4	Input	P913 = Setting not required	PM913 = Setting not required	PMC913 = 1	–	PFC913 = 1	
P914	INTP5	Input	P914 = Setting not required	PM914 = Setting not required	PMC914 = 1	–	PFC914 = 1	
P915	INTP6	Input	P915 = Setting not required	PM915 = Setting not required	PMC915 = 1	–	PFC915 = 1	

Table 4-16. Using Port Pin as Alternate-Function Pin (4/5)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register	Other Bits (Registers)
	Name	I/O						
PCM0	$\overline{\text{WAIT}}$	Input	PCM0 = Setting not required	PMCM0 = Setting not required	PMCCM0 = 1	–	–	
PCM1	CLKOUT	Output	PCM1 = Setting not required	PMCM1 = Setting not required	PMCCM1 = 1	–	–	
PCM2	$\overline{\text{HLDAK}}$	Output	PCM2 = Setting not required	PMCM2 = Setting not required	PMCCM2 = 1	–	–	
PCM3	$\overline{\text{HLDRQ}}$	Input	PCM3 = Setting not required	PMCM3 = Setting not required	PMCCM3 = 1	–	–	
PCS0	$\overline{\text{CS0}}$	Output	PCS0 = Setting not required	PMCS0 = Setting not required	PMCCS0 = 1	–	–	
PCS1	$\overline{\text{CS1}}$	Output	PCS1 = Setting not required	PMCS1 = Setting not required	PMCCS1 = 1	–	–	
PCT0	$\overline{\text{WR0}}$	Output	PCT0 = Setting not required	PMCT0 = Setting not required	PMCCT0 = 1	–	–	
PCT1	$\overline{\text{WR1}}$	Output	PCT1 = Setting not required	PMCT1 = Setting not required	PMCCT1 = 1	–	–	
PCT4	$\overline{\text{RD}}$	Output	PCT4 = Setting not required	PMCT4 = Setting not required	PMCCT4 = 1	–	–	
PCT6	ASTB	Output	PCT6 = Setting not required	PMCT6 = Setting not required	PMCCT6 = 1	–	–	

Table 4-16. Using Port Pin as Alternate-Function Pin (5/5)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register	Other Bits (Registers)
	Name	I/O						
PDH0	A16	Output	PDH0 = Setting not required	PMDH0 = Setting not required	PMCDH0 = 1	–	–	
PDH1	A17	Output	PDH1 = Setting not required	PMDH1 = Setting not required	PMCDH1 = 1	–	–	
PDL0	AD0	I/O	PDL0 = Setting not required	PMDL0 = Setting not required	PMCDL0 = 1	–	–	
PDL1	AD1	I/O	PDL1 = Setting not required	PMDL1 = Setting not required	PMCDL1 = 1	–	–	
PDL2	AD2	I/O	PDL2 = Setting not required	PMDL2 = Setting not required	PMCDL2 = 1	–	–	
PDL3	AD3	I/O	PDL3 = Setting not required	PMDL3 = Setting not required	PMCDL3 = 1	–	–	
PDL4	AD4	I/O	PDL4 = Setting not required	PMDL4 = Setting not required	PMCDL4 = 1	–	–	
PDL5	AD5	I/O	PDL5 = Setting not required	PMDL5 = Setting not required	PMCDL5 = 1	–	–	
	FLMD1[Note]	Input	PDL5 = Setting not required	PMDL5 = Setting not required	Setting not required	–	–	
PDL6	AD6	I/O	PDL6 = Setting not required	PMDL6 = Setting not required	PMCDL6 = 1	–	–	
PDL7	AD7	I/O	PDL7 = Setting not required	PMDL7 = Setting not required	PMCDL7 = 1	–	–	
PDL8	AD8	I/O	PDL8 = Setting not required	PMDL8 = Setting not required	PMCDL8 = 1	–	–	
PDL9	AD9	I/O	PDL9 = Setting not required	PMDL9 = Setting not required	PMCDL9 = 1	–	–	
PDL10	AD10	I/O	PDL10 = Setting not required	PMDL10 = Setting not required	PMCDL10 = 1	–	–	
PDL11	AD11	I/O	PDL11 = Setting not required	PMDL11 = Setting not required	PMCDL11 = 1	–	–	
PDL12	AD12	I/O	PDL12 = Setting not required	PMDL12 = Setting not required	PMCDL12 = 1	–	–	
PDL13	AD13	I/O	PDL13 = Setting not required	PMDL13 = Setting not required	PMCDL13 = 1	–	–	
PDL14	AD14	I/O	PDL14 = Setting not required	PMDL14 = Setting not required	PMCDL14 = 1	–	–	
PDL15	AD15	I/O	PDL15 = Setting not required	PMDL15 = Setting not required	PMCDL15 = 1	–	–	

Note Since this pin is set in the flash memory programming mode, it need not be manipulated with the port control register. For details, refer to **CHAPTER 23 FLASH MEMORY.**

★ 4.6 Cautions

4.6.1 Cautions on setting port pins

(1) Set the registers of a port in the following sequence.

<1> Set PFCn and PFCEn registers.
<2> Set PMCn register.
<3> Set INTFn and INTRn registers.

If the PFCn and PFCEn registers are set after the PMCn register is set, an unexpected peripheral function may be selected while the PFCn and PFCEn registers are being set.

(2) In port mode, the PFn.PFnm bit is valid only in the output mode (PMn.PMnm bit = 0). In the input mode (PMnm bit = 1), the value of the PFnm bit is not reflected in the buffer.

(3) The internal pull-up resistors are connected only in the input mode (PMn register = 1). In the output mode (PMn register = 0), they are disconnected by hardware.

4.6.2 Cautions on bit manipulation instruction for port n register (Pn)

When a 1-bit manipulation instruction is executed on a port that provides both input and output functions, the value of the output latch 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 value of the output latch when switching a port from input mode to output mode.

<Example> When P90 pin is an output port, P91 to P97 pins are input ports (all pin statuses are high level), and the value of the port latch is 00H, if the output of P90 pin is changed from low level to high level via a bit manipulation instruction, the value of the port latch 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 bit manipulation instruction is executed in the following order in the μPD703229Y and 70F3229Y.

<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 value of the output latch (0) of P90 pin, which is an output port, is read, while the pin statuses of P91 to P97 pins, which are input ports, are read. If the pin statuses of P91 to P97 pins 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 4-30. Bit Manipulation Instruction (P90 Pin)

4.6.3 Cautions on on-chip debug pins

The $\overline{\text{DRST}}$, DCK, DMS, DDI, and DDO pins are on-chip debug pins (these pins are available only in the flash-memory versions).

After reset by the $\overline{\text{RESET}}$ pin, the P05/INTP2/$\overline{\text{DRST}}$ pin is initialized to function as an on-chip debug pin ($\overline{\text{DRST}}$). If a high level is input to the $\overline{\text{DRST}}$ pin at this time, the on-chip debug mode is set, and the DCK, DMS, DDI, and DDO pins can be used.

The following action must be taken if on-chip debugging is not used.

- Clear the OCDM0 bit of the OCDM register (special register) (0)

At this time, fix the P05/INTP2/$\overline{\text{DRST}}$ pin to low level from when reset by the $\overline{\text{RESET}}$ pin is released until the above action is taken.

If a high level is input to the $\overline{\text{DRST}}$ pin before the above action is taken, it may cause a malfunction (CPU deadlock). Handle the P05 pin with the utmost care.

Cautions 1. The P05/INTP2/$\overline{\text{DRST}}$ pin does not have to be fixed to low level by an external circuit because it has an internal pull-down resistor (30 kΩ TYP.). The pull-down resistor is disconnected when the OCDM0 bit is cleared (0).

2. After reset by the WDT2RES signal, clock monitor (CLM), or low-voltage detector (LVI), the P05/INTP2/$\overline{\text{DRST}}$ pin is not initialized to function as an on-chip debug pin ($\overline{\text{DRST}}$). The OCDM register holds the current value.

4.6.4 Hysteresis characteristics

In port mode, the following port pins do not have hysteresis characteristics.

P00 to P06
P31 to P35, P39
P40 to P42
P50 to P55
P90, P91, P96, P97, P99, P913 to P915

CHAPTER 5 BUS CONTROL FUNCTION

The μPD703229Y and 70F3229Y are provided with an external bus interface function by which external memories such as ROM and RAM, and I/O can be connected.

5.1 Features

○ Output is selectable from a multiplexed bus with a minimum of 3 bus cycles.
○ 8-bit/16-bit data bus selectable
○ Wait function
- Programmable wait function of up to 7 states
- External wait function using $\overline{\text{WAIT}}$ pin

○ Idle state function
○ Bus hold function
○ Up to 4 MB of physical memory connectable
○ The bus can be controlled using a different voltage from the operating voltage by setting $BV_{DD} \leq EV_{DD} = V_{DD}$.

5.2 Bus Control Pins

The pins used to connect an external device are listed in the table below.

Table 5-1. Bus Control Pins

Bus Control Pin	Alternate-Function Pin	I/O	Function
AD0 to AD15	PDL0 to PDL15	I/O	Address/data bus
A16, A17	PDH0, PDH1	Output	Address bus
$\overline{\text{WAIT}}$	PCM0	Input	External wait control
CLKOUT	PCM1	Output	Internal system clock
$\overline{\text{CS0}}$, $\overline{\text{CS1}}$	PCS0, PCS1	Output	Chip select
$\overline{\text{WR0}}$, $\overline{\text{WR1}}$	PCT0, PCT1	Output	Write strobe signal
$\overline{\text{RD}}$	PCT4	Output	Read strobe signal
ASTB	PCT6	Output	Address strobe signal
$\overline{\text{HLDRQ}}$	PCM3	Input	Bus hold control
$\overline{\text{HLDAK}}$	PCM2	Output	

★ 5.2.1 Pin status when internal ROM, internal RAM, or on-chip peripheral I/O is accessed

When the internal ROM, internal RAM, or on-chip peripheral I/O is accessed, the status of each pin is as follows.

Table 5-2. Pin Statuses When Internal ROM, Internal RAM, or On-Chip Peripheral I/O Is Accessed

Multiplexed Bus Mode	
Address bus (A17, A16)	Undefined
Address/data bus (AD15 to AD0)	Undefined
Control signal	Inactive

Caution When a write access is performed to the internal ROM area, address, data, and control signals are activated in the same way as access to the external memory area.

5.2.2 Pin status in each operation mode

For the pin status of the μPD703229Y and 70F3229Y in each operation mode, refer to **2.2 Pin States**.

5.3 Memory Block Function

The 4 MB external memory space is divided into chip select ($\overline{CSn}$) areas of lower 2 MB and 256 KB. The programmable wait function and bus cycle operation mode for each of these $\overline{CSn}$ areas can be independently controlled (n = 0, 1).

Figure 5-1. Data Memory Map: Physical Address

Note This area is an external memory area in the case of a data write access.

Caution A write access to addresses 0000000H to 003FFFFH is the same operation as a write access to addresses 0100000H to 013FFFFH.

5.4 Bus Access

5.4.1 Number of clocks for access

The following table shows the number of basic clocks required for accessing each resource.

Bus Cycle Type \ Area (Bus Width)	Internal ROM (32 Bits)	Internal RAM (32 Bits)	External Memory (16 Bits)
Instruction fetch (normal access)	1	1 Note	3 + n
Instruction fetch (branch)	2	2 Note	3 + n
Operand data access	3	1	3 + n

Note +1 if a conflict with a data access occurs.

Remark Unit: Clocks/access

5.4.2 Bus size setting function

The bus size of each external memory area selected by $\overline{CSn}$ can be set (to 8 bits or 16 bits) by using the BSC register.

The external memory area of the μPD703229Y and 70F3229Y is selected by $\overline{CS0}$ and $\overline{CS1}$.

(1) Bus size configuration register (BSC)

The BSC register can be read or written in 16-bit units.

Reset input sets this register to 5555H.

Caution Write to the BSC register after reset, and then do not change the set values. Also, do not access an external memory area until the initial settings of the BSC register are complete.

After reset: 5555H R/W Address: FFFFF066H

	15	14	13	12	11	10	9	8
BSC	0	1	0	1	0	1	0	1
	7	6	5	4	3	2	1	0
	0	1	0	1	0	BS10	0	BS00
						$\overline{CS1}$		$\overline{CS0}$

BSn0	Data bus width of CSn space (n = 0, 1)
0	8 bits
1	16 bits

Caution Be sure to set bits 14, 12, 10, 8, 6, and 4 to 1, and clear bits 15, 13, 11, 9, 7, 5, 3, and 1 to 0.

5.4.3 Access by bus size

The μPD703229Y and 70F3229Y access the peripheral I/O and external memory in 8-bit, 16-bit, or 32-bit units. The bus size is as follows.

- The bus size of the peripheral I/O is fixed to 16 bits.
- The bus size of the external memory is selectable from 8 bits or 16 bits (by using the BSC register).

The operation when each of the above is accessed is described below. All data is accessed starting from the lower side.

The μPD703229Y and 70F3229Y support only the little-endian format.

Figure 5-2. Little-Endian Address in Word

31 – 24	23 – 16	15 – 8	7 – 0
000BH	000AH	0009H	0008H
0007H	0006H	0005H	0004H
0003H	0002H	0001H	0000H

(1) Data space

The μPD703229Y and 70F3229Y have an address misalign function.

With this function, data can be placed at all addresses, regardless of the format of the data (word data or halfword data). However, if the word data or halfword data is not aligned at the boundary, a bus cycle is generated at least twice, causing the bus efficiency to drop.

(a) Halfword-length data access

A byte-length bus cycle is generated twice if the least significant bit of the address is 1.

(b) Word-length data access

(i) A byte-length bus cycle, halfword-length bus cycle, and byte-length bus cycle are generated in that order if the least significant bit of the address is 1.

(ii) A halfword-length bus cycle is generated twice if the lower 2 bits of the address are 10.

(2) Byte access (8 bits)

(a) 16-bit data bus width

<1> Access to even address (2n)

<2> Access to odd address (2n + 1)

(b) 8-bit data bus width

<1> Access to even address (2n)

<2> Access to odd address (2n + 1)

(3) Halfword access (16 bits)

(a) 16-bit data bus width

<1> Access to even address (2n)

<2> Access to odd address (2n + 1)

(b) 8-bit data bus width

<1> Access to even address (2n)

<2> Access to odd address (2n + 1)

(4) Word access (32 bits)

(a) 16-bit data bus width (1/2)

<1> Access to address (4n)

<2> Access to address (4n + 1)

(a) 16-bit data bus width (2/2)

<3> Access to address (4n + 2)

<4> Access to address (4n + 3)

(b) 8-bit data bus width (1/2)

<1> Access to address (4n)

<2> Access to address (4n + 1)

(b) 8-bit data bus width (2/2)

<3> Access to address (4n + 2)

<4> Access to address (4n + 3)

5.5 Wait Function

5.5.1 Programmable wait function

(1) Data wait control register 0 (DWC0)

To realize interfacing with a low-speed memory or I/O, up to seven data wait states can be inserted in the bus cycle that is executed for each $\overline{\text{CSn}}$ area.

The number of wait states can be programmed by using the DWC0 register. Immediately after system reset, 7 data wait states are inserted for all $\overline{\text{CSn}}$ areas.

The DWC0 register can be read or written in 16-bit units.

Reset input sets this register to 7777H.

Cautions 1. The internal ROM and internal RAM areas are not subject to programmable wait, and are always accessed without a wait state. The on-chip peripheral I/O area is also not subject to programmable wait, and only wait control from each peripheral function is performed.

2. Write to the DWC0 register after reset, and then do not change the set values. Also, do not access an external memory area until the initial settings of the DWC0 register are complete.

After reset: 7777H R/W Address: FFFFF484H

	15	14	13	12	11	10	9	8
DWC0	0	1	1	1	0	1	1	1

7	6	5	4	3	2	1	0
0	DW12	DW11	DW10	0	DW02	DW01	DW00
	$\overline{\text{CS1}}$				$\overline{\text{CS0}}$		

DWn2	DWn1	DWn0	Number of wait states inserted in CSn space (n = 0, 1)
0	0	0	None
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Caution Be sure to clear bits 15, 11, 7, and 3 to 0.
Be sure to set bits 14 to 12 and 10 to 8 to 1.

5.5.2 External wait function

To synchronize an extremely slow memory, I/O, or asynchronous system, any number of wait states can be inserted in the bus cycle by using the external wait pin ($\overline{\text{WAIT}}$).

Access to each area of the internal ROM, internal RAM, and on-chip peripheral I/O is not subject to control by the external wait function, in the same manner as the programmable wait function.

The $\overline{\text{WAIT}}$ signal can be input asynchronously to CLKOUT, and is sampled at the falling edge of the clock in the T2 and TW states of the bus cycle. If the setup/hold time of the sampling timing is not satisfied, a wait state is inserted in the next state, or not inserted at all.

5.5.3 Relationship between programmable wait and external wait

Wait cycles are inserted as the result of an OR operation between the wait cycles specified by the set value of the programmable wait and the wait cycles controlled by the $\overline{\text{WAIT}}$ pin.

For example, if the timing of the programmable wait and the $\overline{\text{WAIT}}$ pin signal is as illustrated below, three wait states will be inserted in the bus cycle.

Figure 5-3. Example of Inserting Wait States

Remark The circles indicate the sampling timing.

5.5.4 Programmable address wait function

Address-setup or address-hold waits to be inserted in each bus cycle can be set by using the AWC register. Address wait insertion is set for each chip select area ($\overline{CS0}$, $\overline{CS1}$).

If an address setup wait is inserted, it seems that the high-clock period of T1 state is extended by 1 clock. If an address hold wait is inserted, it seems that the low-clock period of T1 state is extended by 1 clock.

(1) Address wait control register (AWC)

The AWC register can be read or written in 16-bit units.

★ **Cautions 1. The internal ROM, internal RAM, and on-chip peripheral I/O areas are not subject to address setup wait/address hold wait insertion.**

2. Write to the AWC register after reset, and then do not change the set values. Also, do not access an external memory area until the initial settings of the AWC register are complete.

After reset: FFFFH R/W Address: FFFFF488H

	15	14	13	12	11	10	9	8
AWC	1	1	1	1	1	1	1	1
	7	6	5	4	3	2	1	0
	1	1	1	1	AHW1	ASW1	AHW0	ASW0
					$\overline{CS1}$		$\overline{CS0}$	

AHWn	Specifies insertion of address hold wait (n = 0, 1)
0	Not inserted
1	Inserted

ASWn	Specifies insertion of address setup wait (n = 0, 1)
0	Not inserted
1	Inserted

Caution Be sure to set bits 15 to 4 to 1.

5.6 Idle State Insertion Function

To facilitate interfacing with low-speed memories, one idle state (TI) can be inserted after the T3 state in the bus cycle that is executed for each space selected by the chip select function in the multiplexed address/data bus mode. By inserting an idle state, the data output float delay time of the memory can be secured during read access (an idle state cannot be inserted during write access).

Whether the idle state is to be inserted can be programmed by using the BCC register.

An idle state is inserted for all the areas immediately after system reset.

(1) Bus cycle control register (BCC)

The BCC register can be read or written in 16-bit units.

Reset input sets this register to AAAAH.

Cautions 1. The internal ROM, internal RAM, and on-chip peripheral I/O areas are not subject to idle state insertion.

2. Write to the BCC register after reset, and then do not change the set values. Also, do not access an external memory area until the initial settings of the BCC register are complete.

After reset: AAAAH R/W Address: FFFFF48AH

	15	14	13	12	11	10	9	8
BCC	1	0	1	0	1	0	1	0
	7	6	5	4	3	2	1	0
	1	0	1	0	BC11	0	BC01	0
					$\overline{CS1}$		$\overline{CS0}$	

BCn1	Specifies insertion of idle state (n = 0, 1)
0	Not inserted
1	Inserted

Caution Be sure to set bits 15, 13, 11, 9, 7, and 5 to 1, and clear bits 14, 12, 10, 8, 6, 4, 2, and 0 to 0.

5.7 Bus Hold Function

5.7.1 Functional outline

The $\overline{\text{HLDRQ}}$ and $\overline{\text{HLDAK}}$ functions are valid if the PCM2 and PCM3 pins are set in the control mode.

When the $\overline{\text{HLDRQ}}$ pin is asserted (low level), indicating that another bus master has requested bus mastership, the external address/data bus goes into a high-impedance state and is released (bus hold status). If the request for the bus mastership is cleared and the $\overline{\text{HLDRQ}}$ pin is deasserted (high level), driving these pins is started again.

During the bus hold period, execution of the program in the internal ROM and internal RAM is continued until an on-chip peripheral I/O register or the external memory is accessed.

The bus hold status is indicated by assertion of the $\overline{\text{HLDAK}}$ pin (low level). The bus hold function enables the configuration multi-processor type systems in which two or more bus masters exist.

Note that the bus hold request is not acknowledged during a multiple-access cycle initiated by the bus sizing function or a bit manipulation instruction.

Status	Data Bus Width	Access Type	Timing at Which Bus Hold Request Is Not Acknowledged
CPU bus lock	16 bits	Word access to even address	Between first and second access
		Word access to odd address	Between first and second access
			Between second and third access
		Halfword access to odd address	Between first and second access
	8 bits	Word access	Between first and second access
			Between second and third access
			Between third and fourth access
		Halfword access	Between first and second access
Read-modify-write access of bit manipulation instruction	–	–	Between read access and write access

5.7.2 Bus hold procedure

The bus hold status transition procedure is shown below.

5.7.3 Operation in power save mode

Because the internal system clock is stopped in the STOP, IDLE1, and IDLE2 modes, the bus hold status is not entered even if the HLDRQ pin is asserted.

In the HALT mode, the HLDAK pin is asserted as soon as the HLDRQ pin has been asserted, and the bus hold status is entered. When the HLDRQ pin is later deasserted, the HLDAK pin is also deasserted, and the bus hold status is cleared.

5.8 Bus Priority

Bus hold, instruction fetch (branch), instruction fetch (successive), and operand data accesses are executed in the external bus cycle.

Bus hold has the highest priority, followed by operand data access, instruction fetch (branch), and instruction fetch (successive).

An instruction fetch may be inserted between the read access and write access in a read-modify-write access.

If an instruction is executed for two or more accesses, an instruction fetch and bus hold are not inserted between accesses due to bus size limitations.

Table 5-3. Bus Priority

Priority	External Bus Cycle	Bus Master
High	Bus hold	External device
↕	DMA transfer	DMAC
	Operand data access	CPU
	Instruction fetch (branch)	CPU
Low	Instruction fetch (successive)	CPU

★ 5.9 Bus Timing

Figure 5-4. Multiplex Bus Read Timing (Bus Size: 16 Bits, 16-Bit Access)

8-bit access	Odd address	Even address
AD15 to AD8	Active	Hi-Z
AD7 to AD0	Hi-Z	Active

Remark The broken lines indicate high impedance.

Figure 5-5. Multiplex Bus Read Timing (Bus Size: 8 Bits)

Remark The broken lines indicate high impedance.

Figure 5-6. Multiplex Bus Write Timing (Bus Size: 16 Bits, 16-Bit Access)

8-bit access	Odd address	Even address
AD15 to AD8	Active	Undefined
AD7 to AD0	Undefined	Active
$\overline{WR1}$, $\overline{WR0}$	01	10

Figure 5-7. Multiplex Bus Write Timing (Bus Size: 8 Bits)

Figure 5-8. Multiplex Bus Hold Timing (Bus Size: 16 Bits, 16-Bit Access)

Note This idle state (TI) does not depend on the BCC register settings.

Remarks 1. Refer to **Table 2-2** for the pin statuses in the bus hold mode.

2. The broken lines indicate high impedance.

CHAPTER 6 CLOCK GENERATION FUNCTION

6.1 Overview

The following clock generation functions are available.

- ○ Main clock oscillator
  - In clock-through mode
    f_X = 4 to 5 MHz (internal f_{XX} = 4 to 5 MHz)
  - In PLL mode
    f_X = 4 to 5 MHz (internal f_{XX} = 16 to 20 MHz)
- ○ Subclock oscillator
  - 32.768 kHz
- ○ Multiply (×4) function by PLL (Phase Locked Loop)
  - Clock-through mode/PLL mode selectable
- ○ Ring OSC
  - f_R = 100 to 400 kHz
- ○ Internal system clock generation
  - 7 steps (f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, f_{XT})
- ○ Peripheral clock generation
- ○ Clock output function

Remark f_X: Main clock oscillation frequency
f_{XX}: Main clock frequency
f_R: Ring-OSC clock frequency

6.2 Configuration

Figure 6-1. Clock Generator

★

f_X: Main clock oscillation frequency
f_{XX}: Main clock frequency
f_{CLK}: Internal system clock frequency
f_{XT}: Subclock frequency
f_{CPU}: CPU clock frequency
f_{BRG}: Watch timer clock frequency

(1) Main resonator: Main clock oscillator

The main resonator oscillates the following frequencies (f_X).

- In clock-through mode
  f_X = 4 to 5 MHz
- In PLL mode
  f_X = 4 to 5 MHz

(2) Sub-resonator: Subclock oscillator

The sub-resonator oscillates a frequency of 32.768 kHz (f_{XT}).

(3) Main resonator stop control

This circuit generates a control signal that stops oscillation of the main resonator.
Oscillation of the main resonator is stopped in the STOP mode or when the PCC.MCK bit = 1 (valid only when the PCC.CLS bit = 1).

(4) Ring-OSC

Outputs a frequency (f_R) of 100 to 400 kHz.

(5) Prescaler 1

This prescaler generates the clock (f_{XX} to f_{XX}/1024) to be supplied to the following on-chip peripheral functions: TMP0 to TMP3, TMQ, TMM, CSIB1, UARTA0 to UARTA2, I^2C00, ADC, and WDT2

(6) Prescaler 2

This circuit divides the main clock (f_{XX}).
The clock generated by prescaler 2 (f_{XX} to f_{XX}/32) is supplied to the selector that generates the CPU clock (f_{CPU}) and internal system clock (f_{CLK}).
f_{CLK} is the clock supplied to the INTC, ROM correction, ROM, and RAM blocks, and can be output from the CLKOUT pin.

(7) Prescaler 3

This circuit divides the clock (f_X) generated by the main resonator to a specific frequency (32.768 kHz) and supplies that clock to the WT block and CSIB0.
For details, refer to **CHAPTER 10 WATCH TIMER FUNCTIONS** and **CHAPTER 14 3-WIRE VARIABLE-LENGTH SERIAL I/O (CSIB)**.

(8) Prescaler 4

This prescaler generates the clock (f_X to f_X/1024) to be supplied to on-chip peripheral functions.
This clock is only supplied to WDT2.

(9) PLL

This circuit multiplies the clock (f_X) generated by the main resonator by 4.
It operates in two modes: clock-through mode in which f_X is output as is, and PLL mode in which a multiplied clock is output. These modes can be selected by using the PLLCTL.SELPLL bit.
Operation of the PLL can be started or stopped by the PLLCTL.PLLON bit.

6.3 Control Registers

(1) Processor clock control register (PCC)

The PCC register is a special register. Data can be written to this register only in combination of specific sequences (refer to **3.4.7 Special registers**).

This register can be read or written in 8-bit or 1-bit units.

Reset input sets this register to 03H.

After reset: 03H R/W Address: FFFFF828H

	7	<6>	5	<4>	<3>	2	1	0
PCC	FRC	MCK	MFRC	CLS[Note]	CK3	CK2	CK1	CK0

FRC	Use of subclock on-chip feedback resistor
0	Used
1	Not used

MCK	Main clock oscillator control
0	Oscillation enabled
1	Oscillation stopped

- Even if the MCK bit is set to 1 while the system is operating with the main clock as the CPU clock, the operation of the main clock does not stop. It stops after the CPU clock has been changed to the subclock.
- When the main clock is stopped and the device is operating on the subclock, clear the MCK bit to 0 and wait until the oscillation stabilization time has been secured by the program before switching back to the main clock.

MFRC	Use of main clock on-chip feedback resistor
0	Used
1	Not used

CLS[Note]	Status of CPU clock (f_{CPU})
0	Main clock operation
1	Subclock operation

CK3	CK2	CK1	CK0	Clock selection (f_{CLK}/f_{CPU})
0	0	0	0	f_{XX}
0	0	0	1	$f_{XX}/2$
0	0	1	0	$f_{XX}/4$
0	0	1	1	$f_{XX}/8$
0	1	0	0	$f_{XX}/16$
0	1	0	1	$f_{XX}/32$
0	1	1	×	Setting prohibited
1	×	×	×	f_{XT}

Note The CLS bit is a read-only bit.

Cautions 1. Do not change the CPU clock (by using the CK3 to CK0 bits) while CLKOUT is being output.

2. Use a bit manipulation instruction to manipulate the CK3 bit. When using an 8-bit manipulation instruction, do not change the set values of the CK2 to CK0 bits.

Remark ×: don't care

★ **(a) Example of setting main clock operation → subclock operation**

<1> CK3 bit ← 1: Use of a bit manipulation instruction is recommended. Do not change the CK2 to CK0 bits.

<2> Subclock operation: Read the CLS bit to check if subclock operation has started. It takes the following time after the CK3 bit is set until subclock operation is started.

Max.: $1/f_{XT}$ (1/subclock frequency)

<3> MCK bit ← 1: Set the MCK bit to 1 only when stopping the main clock.

Cautions 1. When stopping the main clock, stop the PLL.

2. If the following conditions are not satisfied, change the CK2 to CK0 bits so that the conditions are satisfied, then change to the subclock operation mode.

Main clock (f_{XX}) > Subclock (f_{XT}: 32.768 kHz) × 4

[Description example]

```
     _DMA_DISABLE:
     clrl       0, DCHCn[r0]        -- DMA operation disabled.  n = 0 to 3
<1>  _SET_SUB_RUN :
     st.b       r0, PRCMD[r0]
     set1       3, PCC[r0]          -- CK3 bit ← 1
<2>  _CHECK_CLS :
     tst1       4, PCC[r0]          -- Wait until subclock operation starts.
     bz         _CHECK_CLS
<3>  _STOP_MAIN_CLOCK :
     st.b       r0, PRCMD[r0]
     set1       6, PCC[r0]          -- MCK bit ← 1, main clock is stopped.
     _DMA_ENABLE:
     setl       0, DCHCn[r0]        -- DMA operation enabled.  n = 0 to 3
```

Remark The description above is simply an example. Note that in <2> above, the CLS bit is read in a closed loop.

★ **(b) Example of setting subclock operation → main clock operation**

<1> MCK bit ← 0: Main clock starts oscillating

<2> Insert waits by the program and wait until the oscillation stabilization time of the main clock elapses.

<3> CK3 bit ← 0: Use of a bit manipulation instruction is recommended. Do not change the CK2 to CK0 bits.

<4> Main clock operation: It takes the following time after the CK3 bit is set until main clock operation is started.

Max.: $1/f_{XT}$ (1/subclock frequency)

Therefore, insert one NOP instruction immediately after setting the CK3 bit to 0 or read the CLS bit to check if main clock operation has started.

[Description example]

```
     _DMA_DISABLE:
     clrl       0, DCHCn[r0]              -- DMA operation disabled.  n = 0 to 3
<1>  _START_MAIN_OSC :
     st.b       r0, PRCMD[r0]             -- Release of protection of special registers
     clr1       6, PCC[r0]                -- Main clock starts oscillating.
<2>  movea      0x55, r0, r11             -- Wait for oscillation stabilization time.
     _WAIT_OST :
     nop
     nop
     nop
     addi       -1, r11, r11
     cmp        r0, r11
     bne                _WAIT_OST
<3>  st.b       r0, PRCMD[r0]
     clr1       3, PCC[r0]                -- CK3 ← 0
<4>  _CHECK_CLS :
     tst1       4, PCC[r0]                -- Wait until main clock operation starts.
     bnz        _CHECK_CLS
     _DMA_ENABLE:
     setl       0, DCHCn[r0]              -- DMA operation enabled.  n = 0 to 3
```

Remark The description above is simply an example. Note that in <4> above, the CLS bit is read in a closed loop.

(2) Ring-OSC mode register (RCM)

The RCM register is an 8-bit register that sets the operation mode of the Ring-OSC oscillator.
This register can be read or written in 8-bit or 1-bit units.
Reset input clears this register to 00H.

After reset: 00H R/W After reset: FFFFF80CH

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

RSTOP	Oscillation/stop of Ring-OSC
0	Ring-OSC oscillating
1	Ring-OSC stopped

(3) CPU operation clock status register (CCLS)

The CCLS register indicates the status of the CPU operation clock.
This register is read-only, in 8-bit or 1-bit units.
Reset input clears this register to 00H.

After reset: 00H R Address: FFFFF82EH

	7	6	5	4	3	2	1	0
CCLS	0	0	0	0	0	0	0	CCLSF

CCLSF	CPU operation clock status
0	Operating on main clock (f_X) or subclock (f_{XT})
1	Operating on Ring-OSC (f_R)

6.4 Operation

6.4.1 Operation of each clock

The following table shows the operation status of each clock.

Table 6-1. Operation Status of Each Clock

Register Setting and Operation Status / Target Clock	PCC Register								
	CLS Bit = 0, MCK Bit = 0					CLS Bit = 1, MCK Bit = 0		CLS Bit = 1, MCK Bit = 1	
	During Reset	During Oscillation Stabilization Time Count	HALT Mode	IDLE1, IDLE2 Mode	STOP Mode	Subclock Mode	Sub-IDLE Mode	Subclock Mode	Sub-IDLE Mode
Main resonator (f_X)	×	○	○	○	×	○	○	×	×
Sub-resonator (f_{XT})	○	○	○	○	○	○	○	○	○
CPU clock (f_{CPU})	×	×	×	×	×	○	×	○	×
Internal system clock (f_{CLK})	×	×	○	×	×	○	×	○	×
Peripheral clock (f_{XX} to f_{XX}/1,024)	×	×	○	×	×	○	×	×	×
WT clock (main)	×	×	○	○	×	○	○	×	×
WT clock (sub)	○	○	○	○	○	○	○	○	○
WDT2 clock (ring)	×	○	○	○	○	○	○	○	○
WDT2 clock (main)	×	×	○	○	×	○	○	×	×

Remark ○: Operable
×: Stopped

6.4.2 Clock output function

The clock output function is used to output the internal system clock (f_{CLK}) from the CLKOUT pin.

The internal system clock (f_{CLK}) is selected by using the PCC.CK3 to PCC.CK0 bits and CLS bit.

The CLKOUT pin functions alternately as the PCM1 pin and functions as a clock output pin if so specified by the control register of port CM.

The status of the CLKOUT pin is the same as the internal system clock in Table 6-1 and the pin can output the clock when it is in the operable status. It outputs a low level in the stopped status. However, after reset, the CLKOUT pin is in the port mode until the output of the CLKOUT pin is set (PCM1: input mode). Consequently, the CLKOUT pin goes into a high-impedance state.

6.5 PLL Function

6.5.1 Overview

The PLL function is used to output the operating clock of the CPU and peripheral macro at a frequency 4 times higher than the oscillation frequency, and select the clock-through mode.

When PLL function is used: Input clock = 4 to 5 MHz (output: 16 to 20 MHz)
Clock-through mode: Input clock = 4 to 5 MHz (output: 4 to 5 MHz)

6.5.2 Control registers

(1) PLL control register (PLLCTL)

The PLLCTL register is an 8-bit register that controls the PLL function.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 01H.

After reset: 01H R/W Address: FFFFF82CH

	7	6	5	4	3	2	<1>	<0>
PLLCTL	0	0	0	0	0	0	SELPLL	PLLON

PLLON	PLL operation stop register
0	PLL stopped
1	PLL operating (After PLL operation starts, a lockup time is required for frequency stabilization)

SELPLL	CPU operation clock selection register
0	Clock-through operation
1	PLL operation

Cautions 1. When the PLLON bit is cleared to 0, the SELPLL bit is automatically cleared to 0 (clock-through mode).

2. The SELPLL bit can be set to 1 only when the PLL clock frequency is stabilized. If not (unlocked), “0” is written to the SELPLL bit if data is written to it.

(2) Lock register (LOCKR)

Phase lock occurs at a given frequency following power application or immediately after the STOP mode is released, and the time required for stabilization is the lockup time (frequency stabilization time). This time until stabilization is called the lockup status, and the stabilized state is called the locked status.

The LOCKR register includes a LOCK bit that reflects the PLL frequency stabilization status.

This register is read-only in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R Address: FFFFF824H

	7	6	5	4	3	2	1	<0>
LOCKR	0	0	0	0	0	0	0	LOCK

LOCK	PLL lock status check
0	Locked status
1	Unlocked status

Caution The LOCK register does not reflect the PLL lock status in real time. The set/reset conditions are as follows.

[Set conditions]

- Upon system reset[Note]
- In IDLE2 or STOP mode[Note]
- Upon setting of PLL stop (setting of the PLLCTL.PLLON bit to 0)
- Upon stopping main clock and using CPU with subclock (setting of the PCC.CK3 bit to 1 and setting of the PCC.MCK bit to 1)

Note This register is set to 01H by reset and cleared to 00H after the reset has been released and the oscillation stabilization time has elapsed.

[Reset conditions]

- Upon overflow of oscillation stabilization time following reset release (OSTS register default time (refer to **19.2 (3) Oscillation stabilization time select register (OSTS)**))
- Upon oscillation stabilization timer overflow (time set by the OSTS register) following STOP mode release, when the STOP mode was set in the PLL operating status
- Upon PLL lockup time timer overflow (time set by the PLLS register) when the PLLON bit is changed from 0 to 1

(3) PLL lockup time specification register (PLLS)

The PLLS register is an 8-bit register used to select the PLL lockup time when the PLLCTL.PLLON bit is changed from 0 to 1.

This register can be read or written in 8-bit units.

Reset input sets this register to 03H.

After reset: 03H R/W Address: FFFFF6C1H

	7	6	5	4	3	2	1	0
PLLS	0	0	0	0	0	0	PLLS1	PLLS0

PLLS1	PLLS0	PLL lockup time selection
0	0	Setting prohibited
0	1	Setting prohibited
1	0	$2^{12}/f_X$
1	1	$2^{13}/f_X$ (default value)

Caution Set so that the lockup time is 800 μs or longer.

6.5.3 Usage

★ **(1) When PLL is used**

- After the reset signal has been released, the PLL operates (PLLCTL.PLLON bit = 1), but because the default mode is the clock-through mode (PLLCTL.SELPLL bit = 0), select the PLL mode (SELPLL bit = 1).
- To enable the PLL operation, first set the PLLON bit to 1, and then set the SELPLL bit to 1 after the LOCKR.LOCK bit = 0. To stop the PLL, first select the clock-through mode (SELPLL bit = 0), wait for 8 clocks or more, and then stop the PLL (PLLON bit = 0).
- The PLL stops during transition to IDLE2 or STOP mode regardless of the setting and is restored from IDLE2 or STOP mode to the status before transition. The time required for restoration is as follows.

  - From STOP mode: Oscillation stabilization time (1 ms (min.))
  - From IDLE2 mode: Setup time (350 μs (min.))

  When shifting to the IDLE2 or STOP mode while remaining in the PLL operation mode, set the OSTS register as follows.
  Setup time / Oscillation stabilization time > PLL lockup time
  When shifting to the IDLE1 mode, the PLL does not stop. Stop the PLL if necessary.

(2) When PLL is not used

- The clock-through mode (SELPLL bit = 0) is selected after the reset signal has been released, but the PLL is operating (PLLON bit = 1) and must therefore be stopped (PLLON bit = 0).

★ CHAPTER 7 16-BIT TIMER/EVENT COUNTER P (TMP)

Timer P (TMP) is a 16-bit timer/event counter.
The μPD703229Y and 70F3229Y have four timer/event counter channels, TMP0 to TMP3.

7.1 Overview

An outline of TMPn 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

Remark n = 0 to 3

7.2 Functions

TMPn 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

Remark n = 0 to 3

7.3 Configuration

TMPn includes the following hardware.

Table 7-1. 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	2 (TIPn0[Note], TIPn1 pins)
Timer outputs	2 (TOPn0, TOPn1 pins)
Control registers	TMPn control registers 0, 1 (TPnCTL0, TPnCTL1) TMPn I/O control registers 0 to 2 (TPnIOC0 to TPnIOC2) TMPn option registers 0, 1 (TPnOPT0, TPnOPT1)

Note The TIPn0 pin functions alternately as a capture trigger input signal, external event count input signal, and external trigger input signal.

Remark n = 0 to 3

Figure 7-1. Block Diagram of TMPn

Notes 1. TMP0, TMP2

2. TMP1, TMP3

Remark fXX: Main 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 input clears 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 TIPn0 and TIPn1 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 TOPn0 and TOPn1 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.

7.4 Registers

(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 input clears this register to 00H.

The same value can always be written to the TPnCTL0 register by software.

After reset: 00H R/W Address: TP0CTL0 FFFFF590H, TP1CTL0 FFFFF5A0H, TP2CTL0 FFFFF5B0H, TP3CTL0 FFFFF5C0H

	<7>	6	5	4	3	2	1	0
TPnCTL0 (n = 0 to 3)	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, 2	n = 1, 3
0	0	0	f_{XX}	
0	0	1	$f_{XX}/2$	
0	1	0	$f_{XX}/4$	
0	1	1	$f_{XX}/8$	
1	0	0	$f_{XX}/16$	
1	0	1	$f_{XX}/32$	
1	1	0	$f_{XX}/64$	
1	1	1	$f_{XX}/128$	f_{XT}

Note TPnOPT0.TPnOVF bit, 16-bit counter, timer output (TOPn0, TOPn1 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 set bits 3 to 6 to 0.

Remark f_{XX}: Main 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 input clears this register to 00H.

After reset: 00H R/W Address: TP0CTL1 FFFFF591H, TP1CTL1 FFFFF5A1H, TP2CTL1 FFFFF5B1H, TP3CTL1 FFFFF5C1H

	7	6	5	4	3	2	1	0
TPnCTL1 (n = 0 to 3)	0	TPnEST	TPnEEE	0	0	TPnMD2	TPnMD1	TPnMD0

TPnEST	Software trigger control
0	–
1	Generate 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
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.)
The TPnEEE bit selects whether counting is performed with the internal count clock or the valid edge of the external event count input.	

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 set 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 (TOPn0, TOPn1 pins).

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: TP0IOC0 FFFFF592H, TP1IOC0 FFFFF5A2H, TP2IOC0 FFFFF5B2H, TP3IOC0 FFFFF5C2H

	7	6	5	4	3	<2>	1	<0>
TPnIOC0	0	0	0	0	TPnOL1	TPnOE1	TPnOL0	TPnOE0

(n = 0 to 3)

TPnOL1	TOPn1 pin output level setting
0	TOPn1 pin output inversion disabled
1	TOPn1 pin output inversion enabled

TPnOE1	TOPn1 pin output setting
0	Timer output disabled • When TPnOL1 bit = 0: Low level is output from the TOPn1 pin • When TPnOL1 bit = 1: High level is output from the TOPn1 pin
1	Timer output enabled (a square wave is output from the TOPn1 pin).

TPnOL0	TOPn0 pin output level setting
0	TOPn0 pin output inversion disabled
1	TOPn0 pin output inversion enabled

TPnOE0	TOPn0 pin output setting
0	Timer output disabled • When TPnOL0 bit = 0: Low level is output from the TOPn0 pin • When TPnOL0 bit = 1: High level is output from the TOPn0 pin
1	Timer output enabled (a square wave is output from the TOPn0 pin).

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 TOPnm 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 (TIPn0, TIPn1 pins).

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: TP0IOC1 FFFFF593H, TP1IOC1 FFFFF5A3H, TP2IOC1 FFFFF5B3H, TP3IOC1 FFFFF5C3H

	7	6	5	4	3	2	1	0
TPnIOC1	0	0	0	0	TPnIS3	TPnIS2	TPnIS1	TPnIS0

(n = 0 to 3)

TPnIS3	TPnIS2	Capture trigger input signal (TIPn1 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 (TIPn0 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 (TIPn0 pin) and external trigger input signal (TIPn0 pin).

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: TP0IOC2 FFFFF594H, TP1IOC2 FFFFF5A4H, TP2IOC2 FFFFF5B4H, TP3IOC2 FFFFF5C4H

	7	6	5	4	3	2	1	0
TPnIOC2 (n = 0 to 3)	0	0	0	0	TPnEES1	TPnEES0	TPnETS1	TPnETS0

TPnEES1	TPnEES0	External event count input signal (TIPn0 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 (TIPn0 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.

(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 input clears this register to 00H.

After reset: 00H R/W Address: TP0OPT0 FFFFF595H, TP1OPT0 FFFFF5A5H, TP2OPT0 FFFFF5B5H, TP3OPT0 FFFFF5C5H

	7	6	5	4	3	2	1	<0>
TPnOPT0 (n = 0 to 3)	0	0	TPnCCS1	TPnCCS0	0	0	0	TPnOVF

TPnCCS1	TPnCCR1 register capture/compare selection
0	Compare register selected
1	Capture register selected
The TPnCCS1 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
The TPnCCS0 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

- The TPnOVF bit is set to 1 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.
- The TPnOVF bit is not cleared even when the TPnOVF bit or the TPnOPT0 register are read when the TPnOVF bit = 1.
- The TPnOVF bit can be both read and written, but the TPnOVF bit 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 set 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 input clears this register to 0000H.

Caution Accessing the TPnCCR0 register is disabled during subclock operation with the main clock stopped. For details, refer to 3.4.8 (2).

After reset: 0000H R/W Address: TP0CCR0 FFFFF596H, TP1CCR0 FFFFF5A6H, TP2CCR0 FFFFF5B6H, TP3CCR0 FFFFF5C6H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPnCCR0 (n = 0 to 3)																

(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 TOPn0 pin output is enabled at this time, the output of the TOPn0 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 (TIPn0 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 (TIPn0 pin) 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 7-2. 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 input clears this register to 0000H.

Caution Accessing the TPnCCR1 register is disabled during subclock operation with the main clock stopped. For details, refer to 3.4.8 (2).

After reset: 0000H R/W Address: TP0CCR1 FFFFF598H, TP1CCR1 FFFFF5A8H, TP2CCR1 FFFFF5B8H, TP3CCR1 FFFFF5C8H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPnCCR1 (n = 0 to 3)																

(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 TOPn1 pin output is enabled at this time, the output of the TOPn1 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 (TIPn1 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 (TIPn1 pin) 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 7-3. 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 disabled during subclock operation with the main clock stopped. For details, refer to 3.4.8 (2).

After reset: 0000H R Address: TP0CNT FFFFF59AH, TP1CNT FFFFF5AAH, TP2CNT FFFFF5BAH, TP3CNT FFFFF5CAH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPnCNT (n = 0 to 3)																

7.5 Operation

TMPn can perform the following operations.

Operation	TPnCTL1.TPnEST Bit (Software Trigger Bit)	TIPn0 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 TIPn0 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 3

7.5.1 Interval timer mode (TPnMD2 to TPnMD0 = 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 TOPn0 pin.

Usually, the TPnCCR1 register is not used in the interval timer mode.

Figure 7-2. Configuration in Interval Timer Mode

Remark n = 0 to 3

Figure 7-3. Basic Timing in Interval Timer Mode

Remark n = 0 to 3

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 TOPn0 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 TOPn0 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 3

Figure 7-4. Register Setting in Interval Timer Mode (1/2)

Figure 7-4. Register Setting in Interval Timer Mode (2/2)

(d) TMPn counter read buffer register (TPnCNT)

The count value of the 16-bit counter can be read by reading the TPnCNT register.

(e) TMPn capture/compare register 0 (TPnCCR0)

If the TPnCCR0 register is set to D_0, the interval is as follows.

Interval = $(D_0 + 1) \times$ 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 3

(1) Operation flow in interval timer mode

Figure 7-5. 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 3

(2) Operation timing in interval timer mode

(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, and the output of the TOPn0 pin is inverted.

The value of the 16-bit counter is always 0000H.

Remark n = 0 to 3

(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 TOPn0 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 3

(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): $(D_1 + 1) \times$ Count clock cycle

Interval time (NG): $(10000H + D_2 + 1) \times$ Count clock cycle

Interval time (2): $(D_2 + 1) \times$ Count clock cycle

2. n = 0 to 3

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 TOPn0 pin is inverted.

Therefore, the INTTPnCC0 signal may not be generated at the interval time "$(D_1 + 1) \times$ Count clock cycle" or "$(D_2 + 1) \times$ Count clock cycle" originally expected, but may be generated at an interval of "$(10000H + D_2 + 1) \times$ Count clock cycle".

(d) Operation of TPnCCR1 register

Figure 7-6. Configuration of TPnCCR1 Register

Remark n = 0 to 3

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 TOPn1 pin is inverted. The TOPn1 pin outputs a square wave with the same cycle as that output by the TOPn0 pin.

Figure 7-7. Timing Chart When $D_{01} \geq D_{11}$

Remark n = 0 to 3

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 TOPn1 pin changed.

Figure 7-8. Timing Chart When $D_{01} < D_{11}$

Remark n = 0 to 3

7.5.2 External event count mode (TPnMD2 to TPnMD0 = 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 TOPn0 pin cannot be used.

Usually, the TPnCCR1 register is not used in the external event count mode.

Figure 7-9. Configuration in External Event Count Mode

Remark n = 0 to 3

Figure 7-10. Basic Timing in External Event Count Mode

Caution **This figure shows the basic timing when the rising edge is specified as the valid edge of the external event count input.**

Remark n = 0 to 3

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 7-11. Register Setting in External Event Count Mode (1/2)

Figure 7-11. Register Setting 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).

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 3

(1) Operation flow in external event count mode

Figure 7-12. Software Processing Flow 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 3

(2) Operation timing in external event count mode

(a) Operation if TPnCCR0 register is set to 0000H

If the TPnCCR0 register is set to 0000H, the INTTPnCC0 signal is generated each time the valid edge of the external event count signal has been detected.

The 16-bit counter is always 0000H.

Remark n = 0 to 3

(b) 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.

Remark n = 0 to 3

(c) 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.

Remark n = 0 to 3

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".

(d) Operation of TPnCCR1 register

Figure 7-13. Configuration of TPnCCR1 Register

Remark n = 0 to 3

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. At the same time, the output signal of the TOPn1 pin is inverted.

Figure 7-14. Timing Chart When $D_{01} \geq D_{11}$

Remark n = 0 to 3

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. Nor is the output signal of the TOPn1 pin changed.

Figure 7-15. Timing Chart When $D_{01} < D_{11}$

Remark n = 0 to 3

7.5.3 External trigger pulse output mode (TPnMD2 to TPnMD0 = 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 TOPn1 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 TOPn0 pin.

Figure 7-16. Configuration in External Trigger Pulse Output Mode

Figure 7-17. 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 TOPn1 pin.

If the trigger is generated again while the counter is operating, the counter is cleared to 0000H and restarted.

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 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.

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 3, m = 0, 1

Figure 7-18. Register Setting 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[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	0/1	0/1	0	0	0	1	0

0, 1, 0:
External trigger pulse output mode

0: Operate on count clock selected by TPnCKS0 to TPnCKS2 bits
1: Count with external event input signal

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	0/1

0: Disable TOPn0 pin output
1: Enable TOPn0 pin output

Settings of output level while operation of TOPn0 pin is disabled
0: Low level
1: High level

0: Disable TOPn1 pin output
1: Enable TOPn1 pin output

Specifies active level of TOPn1 pin output
0: Active-high
1: Active-low

- When TPnOL1 bit = 0

- When TPnOL1 bit = 1

Figure 7-18. Register Setting 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	0/1	0/1	0/1	0/1

Select valid edge of external trigger input

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 width 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 external trigger pulse output mode.

2. n = 0 to 3

(1) Operation flow in external trigger pulse output mode

Figure 7-19. Software Processing Flow in External Trigger Pulse Output Mode (1/2)

Remark n = 0 to 3

Figure 7-19. Software Processing Flow in External Trigger Pulse Output Mode (2/2)

Remark n = 0 to 3
m = 0, 1

(2) Operation timing in external trigger pulse output mode

(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 3
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 3

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 3

(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 TOPn1 pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.

Remark n = 0 to 3

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 TOPn1 pin remains active. Consequently, the active period of the PWM waveform is extended.

Remark n = 0 to 3

(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 TOPn1 pin is extended by time from generation of the INTTPnCC0 signal to trigger detection.

Remark n = 0 to 3

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 TOPn1 pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.

Remark n = 0 to 3

(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 3

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 TOPn1 pin.

7.5.4 One-shot pulse output mode (TPnMD2 to TPnMD0 = 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 TOPn1 pin.

Instead of the external trigger, a software trigger can also be generated to output the pulse. When the software trigger is used, the TOPn0 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 7-20. Configuration in One-Shot Pulse Output Mode

Remark n = 0 to 3

Figure 7-21. 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 TOPn1 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 3
m = 0, 1

Figure 7-22. Register Setting 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

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	0/1	0/1	0	0	0	1	1

0, 1, 1:
One-shot pulse output mode

0: Operate on count clock selected by TPnCKS0 to TPnCKS2 bits
1: Count with external event count input signal

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	0/1

0: Disable TOPn0 pin output
1: Enable TOPn0 pin output

Setting of output level while operation of TOPn0 pin is disabled
0: Low level
1: High level

0: Disable TOPn1 pin output
1: Enable TOPn1 pin output

Specifies active level of TOPn1 pin output
0: Active-high
1: Active-low

• When TPnOL1 bit = 0

• When TPnOL1 bit = 1

Figure 7-22. Register Setting 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	0/1	0/1	0/1	0/1

Select valid edge of external trigger input

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 active level width and output delay period of the one-shot pulse are as follows.

Active level width = $(D_1 - D_0 + 1) \times$ Count clock cycle

Output delay period = $D_1 \times$ Count clock cycle

Remarks 1. TMPn I/O control register 1 (TPnIOC1) and TMPn option register 0 (TPnOPT0) are not used in the one-shot pulse output mode.

2. n = 0 to 3

(1) Operation flow in one-shot pulse output mode

Figure 7-23. Software Processing Flow in One-Shot Pulse Output 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). Trigger wait status

<2> Count operation stop flow

Count operation is stopped.

Remark n = 0 to 3

(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 TOPn1 pin. When the count value matches D_{01}, the counter generates the INTTPnCC0 signal, deasserts the TOPn1 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 3
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.

Remark n = 0 to 3

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 TOPn1 pin.

Remark n = 0 to 3

7.5.5 PWM output mode (TPnMD2 to TPnMD0 = 100)

In the PWM output mode, a PWM waveform is output from the TOPn1 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 TOPn0 pin.

Figure 7-24. Configuration in PWM Output Mode

Remark n = 0 to 3

Figure 7-25. 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 TOPn1 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 3, m = 0, 1

Figure 7-26. Register Setting 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	0	0	1	0	0

1, 0, 0:
PWM output mode

(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

0: Disable TOPn0 pin output
1: Enable TOPn0 pin output

Setting of output level while operation of TOPn0 pin is disabled
0: Low level
1: High level

0: Disable TOPn1 pin output
1: Enable TOPn1 pin output

Specifies active level of TOPn1 pin output
0: Active-high
1: Active-low

- When TPnOL1 bit = 0

- When TPnOL1 bit = 1

Figure 7-26. Register Setting in PWM Output Mode (2/2)

(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.

(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 width 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 3

(1) Operation flow in PWM output mode

Figure 7-27. Software Processing Flow in PWM Output Mode (1/2)

Remark n = 0 to 3
m = 0, 1

Figure 7-27. Software Processing Flow in PWM Output Mode (2/2)

Remark n = 0 to 3
m = 0, 1

(2) Operation timing in PWM output mode

(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 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 3
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 3

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 3

(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 3

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 TOPn1 pin.

7.5.6 Free-running timer mode (TPnMD2 to TPnMD0 = 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 7-28. Configuration in Free-Running Timer Mode

Remark n = 0 to 3
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 TOPn0 and TOPn1 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 TOPnm 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 7-29. Basic Timing in Free-Running Timer Mode (Compare Function)

Remark n = 0 to 3
m = 0, 1

When the TPnCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIPnm 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 7-30. Basic Timing in Free-Running Timer Mode (Capture Function)

Remark n = 0 to 3

Figure 7-31. 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

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	1

1, 0, 1:
Free-running mode

0: Operate on count clock selected by TPnCKS0 to TPnCKS2 bits
1: Count with 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

0: Disable TOPn0 pin output
1: Enable TOPn0 pin output

Setting of output level with operation of TOPn0 pin disabled
0: Low level
1: High level

0: Disable TOPn1 pin output
1: Enable TOPn1 pin output

Setting of output level with operation of TOPn1 pin disabled
0: Low level
1: High level

Figure 7-31. 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 TIPn0 pin input

Select valid edge of TIPn1 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 TIPnm 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 TOPnm pin is inverted.

Remark n = 0 to 3
m = 0, 1

(1) Operation flow in free-running timer mode

(a) When using capture/compare register as compare register

Figure 7-32. Software Processing Flow in Free-Running Timer Mode (Compare Function) (1/2)

Remark n = 0 to 3

Figure 7-32. Software Processing Flow in Free-Running Timer Mode (Compare Function) (2/2)

Remark n = 0 to 3

(b) When using capture/compare register as capture register

Figure 7-33. Software Processing Flow in Free-Running Timer Mode (Capture Function) (1/2)

Remark n = 0 to 3

Figure 7-33. Software Processing Flow in Free-Running Timer Mode (Capture Function) (2/2)

Remark n = 0 to 3

(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.

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 3

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.

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 3
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 TIPn0 pin input).
<2> Read the TPnCCR1 register (setting of the default value of the TIPn1 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 TIPn0 pin input).

<2> Read the TPnCCR1 register (setting of the default value of the TIPn1 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 TIPn0 pin input).

<2> Read the TPnCCR1 register (setting of the default value of the TIPn1 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 a long pulse width is measured in the free-running timer mode.

<1> Read the TPnCCRm register (setting of the default value of the TIPnm 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 below.

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 TIPnm 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 ($N \times 10000H + D_{m1} - D_{m0}$).

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.

Remark n = 0 to 3

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.

7.5.7 Pulse width measurement mode (TPnMD2 to TPnMD0 = 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 TIPnm 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 TIPn0 or TIPn1 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 TIPn1 pin because the external clock is fixed to the TIPn0 pin. At this time, clear the TPnIOC1.TPnIS1 and TPnIOC1.TPnIS0 bits to 00 (capture trigger input (TIPn0 pin): No edge detected).

Figure 7-34. Configuration in Pulse Width Measurement Mode

Remark n = 0 to 3
m = 0, 1

Figure 7-35. Basic Timing in Pulse Width Measurement Mode

Remark n = 0 to 3
m = 0, 1

When the TPnCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIPnm 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 = $(D_N + 1) \times$ Count clock cycle

If the valid edge is not input to the TIPnm 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 = $(D_N + 10001H) \times$ Count clock cycle

Remark n = 0 to 3
m = 0, 1

Figure 7-36. 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

Select count clock[Note]

0: Stop counting
1: Enable counting

Note Setting is invalid when the TPnEEE bit = 1.

(b) TMPn control register 1 (TPnCTL1)

		TPnEST	TPnEEE			TPnMD2	TPnMD1	TPnMD0
TPnCTL1	0	0	0/1	0	0	1	1	0

1, 1, 0:
Pulse width measurement mode

0: Operate on count clock selected by TPnCKS0 to TPnCKS2 bits
1: Count with external event count input signal

(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

Select valid edge of TIPn0 pin input

Select valid edge of TIPn1 pin input

(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 7-36. Register Setting in Pulse Width Measurement Mode (2/2)

(e) TMPn option register 0 (TPnOPT0)

			TPnCCS1	TPnCCS0				TPnOVF
TPnOPT0	0	0	0	0	0	0	0	0/1

Overflow flag

(f) TMPn counter read buffer register (TPnCNT)

The value of the 16-bit counter can be read by reading the TPnCNT register.

(g) 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 TIPnm pin is detected.

Remarks 1. TMPn I/O control register 0 (TPnIOC0) is not used in the pulse width measurement mode.

2. n = 0 to 3

m = 0, 1

(1) Operation flow in pulse width measurement mode

Figure 7-37. Software Processing Flow in Pulse Width Measurement 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 3

(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.

Remark n = 0 to 3

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.

7.5.8 Timer output operations

The following table shows the operations and output levels of the TOPn0 and TOPn1 pins.

Table 7-4. Timer Output Control in Each Mode

Operation Mode	TOPn1 Pin	TOPn0 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 3

Table 7-5. Truth Table of TOPn0 and TOPn1 Pins Under Control of Timer Output Control Bits

TPnIOC0.TPnOLm Bit	TPnIOC0.TPnOEm Bit	TPnCTL0.TPnCE Bit	Level of TOPnm 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 3
m = 0, 1

7.6 Eliminating Digital Noise on Timer Input Pin

The μPD703229Y and 70F3229Y have a digital noise eliminator for the timer input pins (TIP00, TIP01, TIP10, TIP11, TIP20, TIP21, TIP30, and TIP31).

The sampling clock for digital sampling can be selected from f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/16$, $f_{XX}/32$, and $f_{XX}/64$ for TMP0 and TMP2, and from f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, and f_{XT} for TMP1 and TMP3. Sampling is performed three times or twice.

(1) TIPn0 noise elimination register and TIPn1 noise elimination register (Pn0NFC and Pn1NFC)

The Pn0NFC and Pn1NFC registers are used to select the digital noise elimination clock for the TIP00, TIP01, TIP10, TIP11, TIP20, TIP21, TIP30, and TIP31 pins.

Digital noise is eliminated from the signal input from the timer input pin before the Pn0NFC and Pn1NFC registers are set, and the signal is always output.

After setting the sampling clock (NFC2 to NFC0 bits) and the number of times of sampling (NFSTS bit), therefore, enable the timer operation after the time of "(Sampling clock) × (Number of times of sampling)[Note]" has elapsed.

These registers can be read or written in 8-bit or 1-bit units.

Reset input clears these registers to 00H.

Note The time of "(Sampling clock) × (Number of times of sampling)" is required until the digital noise eliminator is initialized after digital noise elimination conditions have been set.

(1/2)

After reset: 00H R/W Address: P00NFC FFFFFB00H, P01NFC FFFFFB04H,
P10NFC FFFFFB08H, P11NFC FFFFFB0CH,
P20NFC FFFFFB10H, P21NFC FFFFFB14H,
P30NFC FFFFFB18H, P31NFC FFFFFB1CH

Pn0NFC, Pn1NFC (n = 0 to 3)

7	6	5	4	3	2	1	0
0	NFSTS	0	0	0	NFC2	NFC1	NFC0

NFSTS	Sets number of times of sampling for digital noise elimination
0	Number of times of sampling = 3
1	Number of times of sampling = 2

• For TIP00, TIP01, TIP20, and TIP21 pins of TMP0 and TMP2

NFSTS	NFC2	NFC1	NFC0	Sampling clock	Width of noise accurately eliminated[Note]
					f_{XX} = 20 MHz
0	0	0	0	f_{XX}	0.10 μs
0	0	0	1	$f_{XX}/2$	0.20 μs
0	0	1	0	$f_{XX}/4$	0.40 μs
0	0	1	1	$f_{XX}/16$	1.60 μs
0	1	0	0	$f_{XX}/32$	3.20 μs
0	1	0	1	$f_{XX}/64$	6.40 μs
1	0	0	0	f_{XX}	0.05 μs
1	0	0	1	$f_{XX}/2$	0.10 μs
1	0	1	0	$f_{XX}/4$	0.20 μs
1	0	1	1	$f_{XX}/16$	0.80 μs
1	1	0	0	$f_{XX}/32$	1.60 μs
1	1	0	1	$f_{XX}/64$	3.20 μs
Other than above				Setting prohibited	

Note The width of the noise that can be accurately eliminated is (Sampling clock) × (Number of times of sampling – 1). Even noise with a narrower width than this causes incorrect counting if it is input in synchronization with the sampling clock.

(2/2)

• For TIP10, TIP11, TIP30, and TIP31 pins of TMP1 and TMP3

NFSTS	NFC2	NFC1	NFC0	Sampling clock	Width of noise accurately eliminated[Note]
					f_{XX} = 20 MHz
0	0	0	0	f_{XX}	0.10 μs
0	0	0	1	$f_{XX}/2$	0.20 μs
0	0	1	0	$f_{XX}/4$	0.40 μs
0	0	1	1	$f_{XX}/8$	0.80 μs
0	1	0	0	$f_{XX}/16$	1.60 μs
0	1	0	1	f_{XT}	61.04 μs
1	0	0	0	f_{XX}	0.05 μs
1	0	0	1	$f_{XX}/2$	0.10 μs
1	0	1	0	$f_{XX}/4$	0.20 μs
1	0	1	1	$f_{XX}/8$	0.40 μs
1	1	0	0	$f_{XX}/16$	0.80 μs
1	1	0	1	f_{XT}	30.52 μs
Other than above				Setting prohibited	

Note The width of the noise that can be accurately eliminated is (Sampling clock) × (Number of times of sampling – 1). Even noise with a narrower width than this causes incorrect counting if it is input in synchronization with the sampling clock.

7.7 Selector Function

With the μPD703229Y and 70F3229Y, the TMP capture trigger input can be selected using the alternate-function pin of the port and peripheral I/O (TMP, UARTA, TMM) signals.

This function allows the following.

- Selection of the RXDA pin of UARTA
  → Used for baud rate calculation during LIN communication
- Selection of the INTTM0EQ0 signal of TMM0
  → Used for measuring the subclock (WT clock)/Ring-OSC (WDT2 clock) frequency

Cautions 1. When using the selector function, set the connected timer (TMP) to capture trigger input.
2. When setting the selector function, disable the connected peripheral I/O (TMP, UARTA, TMM) operations first.

Set the capture input via the selector function using the following register.

(1) Selector operation control register 0 (SELCNT0)

The SELCNT0 register is an 8-bit register that selects the capture trigger for TMPn.

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF308H

	7	6	5	4	3	2	1	0
SELCNT0	0	0	ISEL05	ISEL04	ISEL03	ISEL02	0	0

ISEL05	Selection of TIP30 input signal (TMP3)
0	TIP30 pin input
1	RXDA2 pin input

ISEL04	Selection of TIP11 input signal (TMP1)
0	TIP11 pin input
1	RXDA1 pin input

ISEL03	Selection of TIP10 input signal (TMP1)
0	TIP10 pin input
1	RXDA0 pin input

ISEL02	Selection of TIP01 input signal (TMP0)
0	TIP01 pin input
1	INTTM0EQ0 interrupt of TMM0[Note]

Note When using the INTTM0EQ0 signal as the TMP01 input, use it within the following range.

TMM operating clock cycle ≥ TMP operating clock cycle × 4

Cautions 1. To set the ISEL05 to ISEL02 bits to 1, set the corresponding pin in the capture input mode.

2. Be sure to set bits 0, 1, 6, and 7 to 0.

★ CHAPTER 8 16-BIT TIMER/EVENT COUNTER Q (TMQ)

Timer Q (TMQ) is a 16-bit timer/event counter.
The μPD703229Y and 70F3229Y incorporate TMQ0.

8.1 Overview

An outline of TMQ0 is shown below.

- Clock selection: 8 ways
- Capture trigger input pins: 4
- External event count input pins: 1
- External trigger input pins: 1
- Timer/counters: 1
- Capture/compare registers: 4
- Capture/compare match interrupt request signals: 4
- Timer output pins: 4

8.2 Functions

TMQ0 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

8.3 Configuration

TMQ0 includes the following hardware.

Table 8-1. Configuration of TMQ0

Item	Configuration
Timer register	16-bit counter
Registers	TMQ0 capture/compare registers 0 to 3 (TQ0CCR0 to TQ0CCR3) TMQ0 counter read buffer register (TQ0CNT) CCR0 to CCR3 buffer registers
Timer inputs	4 (TIQ00[Note] to TIQ03 pins)
Timer outputs	4 (TOQ00 to TOQ03 pins)
Control registers	TMQ0 control registers 0, 1 (TQ0CTL0, TQ0CTL1) TMQ0 I/O control registers 0 to 2 (TQ0IOC0 to TQ0IOC2) TMQ0 option register 0 (TQ0OPT0)

Note The TIQ00 pin functions alternately as a capture trigger input signal, external event count input signal, and external trigger input signal.

Figure 8-1. Block Diagram of TMQ0

Remark fxx: Main 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 TQ0CNT register.

When the TQ0CTL0.TQ0CE bit = 0, the value of the 16-bit counter is FFFFH. If the TQ0CNT register is read at this time, 0000H is read.

Reset input clears the TQ0CE 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 TQ0CCR0 register is used as a compare register, the value written to the TQ0CCR0 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 (INTTQ0CC0) is generated.

The CCR0 buffer register cannot be read or written directly.

The CCR0 buffer register is cleared to 0000H after reset, as the TQ0CCR0 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 TQ0CCR1 register is used as a compare register, the value written to the TQ0CCR1 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 (INTTQ0CC1) is generated.

The CCR1 buffer register cannot be read or written directly.

The CCR1 buffer register is cleared to 0000H after reset, as the TQ0CCR1 register is cleared to 0000H.

(4) CCR2 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TQ0CCR2 register is used as a compare register, the value written to the TQ0CCR2 register is transferred to the CCR2 buffer register. When the count value of the 16-bit counter matches the value of the CCR2 buffer register, a compare match interrupt request signal (INTTQ0CC2) is generated.

The CCR2 buffer register cannot be read or written directly.

The CCR2 buffer register is cleared to 0000H after reset, as the TQ0CCR2 register is cleared to 0000H.

(5) CCR3 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TQ0CCR3 register is used as a compare register, the value written to the TQ0CCR3 register is transferred to the CCR3 buffer register. When the count value of the 16-bit counter matches the value of the CCR3 buffer register, a compare match interrupt request signal (INTTQ0CC3) is generated.

The CCR3 buffer register cannot be read or written directly.

The CCR3 buffer register is cleared to 0000H after reset, as the TQ0CCR3 register is cleared to 0000H.

(6) Edge detector

This circuit detects the valid edges input to the TIQ00 and TIQ03 pins. No edge, rising edge, falling edge, or both the rising and falling edges can be selected as the valid edge by using the TQ0IOC1 and TQ0IOC2 registers.

(7) Output controller

This circuit controls the output of the TOQ00 to TOQ03 pins. The output controller is controlled by the TQ0IOC0 register.

(8) 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.

8.4 Registers

(1) TMQ0 control register 0 (TQ0CTL0)

The TQ0CTL0 register is an 8-bit register that controls the operation of TMQ0.

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

The same value can always be written to the TQ0CTL0 register by software.

After reset: 00H R/W Address: FFFFF540H

	<7>	6	5	4	3	2	1	0
TQ0CTL0	TQ0CE	0	0	0	0	TQ0CKS2	TQ0CKS1	TQ0CKS0

TQ0CE	TMQ0 operation control
0	TMQ0 operation disabled (TMQ0 reset asynchronously[Note]).
1	TMQ0 operation enabled. TMQ0 operation started.

TQ0CKS2	TQ0CKS1	TQ0CKS0	Internal count clock selection
0	0	0	f_{XX}
0	0	1	$f_{XX}/2$
0	1	0	$f_{XX}/4$
0	1	1	$f_{XX}/8$
1	0	0	$f_{XX}/16$
1	0	1	$f_{XX}/32$
1	1	0	$f_{XX}/64$
1	1	1	$f_{XX}/128$

Note TQ0OPT0.TQ0OVF bit, 16-bit counter, timer output (TOQ00 to TOQ03 pins)

Cautions 1. Set the TQ0CKS2 to TQ0CKS0 bits when the TQ0CE bit = 0. When the value of the TQ0CE bit is changed from 0 to 1, the TQ0CKS2 to TQ0CKS0 bits can be set simultaneously.

2. Be sure to set bits 3 to 6 to 0.

Remark f_{XX}: Main clock frequency

(2) TMQ0 control register 1 (TQ0CTL1)

The TQ0CTL1 register is an 8-bit register that controls the operation of TMQ0.

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF541H

	7	6	5	4	3	2	1	0
TQ0CTL1	0	TQ0EST	TQ0EEE	0	0	TQ0MD2	TQ0MD1	TQ0MD0

TQ0EST	Software trigger control
0	–
1	Generate a valid signal for external trigger input. • In one-shot pulse output mode: A one-shot pulse is output with writing 1 to the TQ0EST bit as the trigger. • In external trigger pulse output mode: A PWM waveform is output with writing 1 to the TQ0EST bit as the trigger.

TQ0EEE	Count clock selection
0	Disable operation with external event count input. (Perform counting with the count clock selected by the TQ0CTL0.TQ0CK0 to TQ0CK2 bits.)
1	Enable operation with external event count input. (Perform counting at the valid edge of the external event count input signal.)
The TQ0EEE bit selects whether counting is performed with the internal count clock or the valid edge of the external event count input.	

TQ0MD2	TQ0MD1	TQ0MD0	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 TQ0EST 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 TQ0EEE bit.

3. Set the TQ0EEE and TQ0MD2 to TQ0MD0 bits when the TQ0CTL0.TQ0CE bit = 0. (The same value can be written when the TQ0CE bit = 1.) The operation is not guaranteed when rewriting is performed with the TQ0CE bit = 1. If rewriting was mistakenly performed, clear the TQ0CE bit to 0 and then set the bits again.

4. Be sure to set bits 3, 4, and 7 to 0.

(3) TMQ0 I/O control register 0 (TQ0IOC0)

The TQ0IOC0 register is an 8-bit register that controls the timer output (TOQ00 to TOQ03 pins).

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF542H

	7	<6>	5	<4>	3	<2>	1	<0>
TQ0IOC0	TQ0OL3	TQ0OE3	TQ0OL2	TQ0OE2	TQ0OL1	TQ0OE1	TQ0OL0	TQ0OE0

TQ0OLm	TOQ0m pin output level setting (m = 0 to 3)
0	TOQ0m pin output inversion disabled
1	TOQ0m pin output inversion enabled

TQ0OEm	TOQ0m pin output setting (m = 0 to 3)
0	Timer output disabled • When TQ0OLm bit = 0: Low level is output from the TOQ0m pin • When TQ0OLm bit = 1: High level is output from the TOQ0m pin
1	Timer output enabled (A square wave is output from the TOQ0m pin).

Cautions 1. Rewrite the TQ0OLm and TQ0OEm bits when the TQ0CTL0.TQ0CE bit = 0. (The same value can be written when the TQ0CE bit = 1.) If rewriting was mistakenly performed, clear the TQ0CE bit to 0 and then set the bits again.

2. Even if the TQ0OLm bit is manipulated when the TQ0CE and TQ0OEm bits are 0, the TOQ0m pin output level varies.

Remark m = 0 to 3

(4) TMQ0 I/O control register 1 (TQ0IOC1)

The TQ0IOC1 register is an 8-bit register that controls the valid edge of the capture trigger input signals (TIQ00 to TIQ03 pins).

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF543H

	7	6	5	4	3	2	1	0
TQ0IOC1	TQ0IS7	TQ0IS6	TQ0IS5	TQ0IS4	TQ0IS3	TQ0IS2	TQ0IS1	TQ0IS0

TQ0IS7	TQ0IS6	Capture trigger input signal (TIQ03 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

TQ0IS5	TQ0IS4	Capture trigger input signal (TIQ02 pin) valid edge detection
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

TQ0IS3	TQ0IS2	Capture trigger input signal (TIQ01 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

TQ0IS1	TQ0IS0	Capture trigger input signal (TIQ00 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 TQ0IS7 to TQ0IS0 bits when the TQ0CTL0.TQ0CE bit = 0. (The same value can be written when the TQ0CE bit = 1.) If rewriting was mistakenly performed, clear the TQ0CE bit to 0 and then set the bits again.

2. The TQ0IS7 to TQ0IS0 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) TMQ0 I/O control register 2 (TQ0IOC2)

The TQ0IOC2 register is an 8-bit register that controls the valid edge of the external event count input signal (TIQ00 pin) and external trigger input signal (TIQ00 pin).

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF544H

	7	6	5	4	3	2	1	0
TQ0IOC2	0	0	0	0	TQ0EES1	TQ0EES0	TQ0ETS1	TQ0ETS0

TQ0EES1	TQ0EES0	External event count input signal (TIQ00 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

TQ0ETS1	TQ0ETS0	External trigger input signal (TIQ00 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 TQ0EES1, TQ0EES0, TQ0ETS1, and TQ0ETS0 bits when the TQ0CTL0.TQ0CE bit = 0. (The same value can be written when the TQ0CE bit = 1.) If rewriting was mistakenly performed, clear the TQ0CE bit to 0 and then set the bits again.

2. The TQ0EES1 and TQ0EES0 bits are valid only when the TQ0CTL1.TQ0EEE bit = 1 or when the external event count mode (TQ0CTL1.TQ0MD2 to TQ0CTL1.TQ0MD0 bits = 001) has been set.

(6) TMQ0 option register 0 (TQ0OPT0)

The TQ0OPT0 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 input clears this register to 00H.

After reset: 00H R/W Address: FFFFF545H

	7	6	5	4	3	2	1	<0>
TQ0OPT0	TQ0CCS3	TQ0CCS2	TQ0CCS1	TQ0CCS0	0	0	0	TQ0OVF

TQ0CCSm	TQ0CCRm register capture/compare selection
0	Compare register selected
1	Capture register selected
The TQ0CCSm bit setting is valid only in the free-running timer mode.	

TQ0OVF	TMQ0 overflow detection
Set (1)	Overflow occurred
Reset (0)	TQ0OVF bit 0 written or TQ0CTL0.TQ0CE bit = 0

- The TQ0OVF bit is set to 1 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 (INTTQ0OV) is generated at the same time that the TQ0OVF bit is set to 1. The INTTQ0OV signal is not generated in modes other than the free-running timer mode and the pulse width measurement mode.
- The TQ0OVF bit is not cleared even when the TQ0OVF bit or the TQ0OPT0 register are read when the TQ0OVF bit = 1.
- The TQ0OVF bit can be both read and written, but the TQ0OVF bit cannot be set to 1 by software. Writing 1 has no influence on the operation of TMQ0.

Cautions 1. Rewrite the TQ0CCS3 to TQ0CCS0 bits when the TQ0CE bit = 0. (The same value can be written when the TQ0CE bit = 1.) If rewriting was mistakenly performed, clear the TQ0CE bit to 0 and then set the bits again.

2. Be sure to set bits 1 to 3 to 0.

Remark m = 0 to 3

(7) TMQ0 capture/compare register 0 (TQ0CCR0)

The TQ0CCR0 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 TQ0OPT0.TQ0CCS0 bit. In the pulse width measurement mode, the TQ0CCR0 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TQ0CCR0 register can be read or written during operation.

This register can be read or written in 16-bit units.

Reset input clears this register to 0000H.

Caution Accessing the TQ0CCR0 register is disabled during subclock operation with the main clock stopped. For details, refer to 3.4.8 (2).

After reset: 0000H R/W Address: FFFFF546H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TQ0CCR0																

(a) Function as compare register

The TQ0CCR0 register can be rewritten even when the TQ0CTL0.TQ0CE bit = 1.

The set value of the TQ0CCR0 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 (INTTQ0CC0) is generated. If TOQ00 pin output is enabled at this time, the output of the TOQ00 pin is inverted.

When the TQ0CCR0 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 TQ0CCR0 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 TQ0CCR0 register if the valid edge of the capture trigger input pin (TIQ00 pin) is detected. In the pulse width measurement mode, the count value of the 16-bit counter is stored in the TQ0CCR0 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIQ00 pin) is detected.

Even if the capture operation and reading the TQ0CCR0 register conflict, the correct value of the TQ0CCR0 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 8-2. 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) TMQ0 capture/compare register 1 (TQ0CCR1)

The TQ0CCR1 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 TQ0OPT0.TQ0CCS1 bit. In the pulse width measurement mode, the TQ0CCR1 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TQ0CCR1 register can be read or written during operation.

This register can be read or written in 16-bit units.

Reset input clears this register to 0000H.

Caution Accessing the TQ0CCR1 register is disabled during subclock operation with the main clock stopped. For details, refer to 3.4.8 (2).

After reset: 0000H R/W Address: FFFFF548H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TQ0CCR1																

(a) Function as compare register

The TQ0CCR1 register can be rewritten even when the TQ0CTL0.TQ0CE bit = 1.

The set value of the TQ0CCR1 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 (INTTQ0CC1) is generated. If TOQ01 pin output is enabled at this time, the output of the TOQ01 pin is inverted.

(b) Function as capture register

When the TQ0CCR1 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 TQ0CCR1 register if the valid edge of the capture trigger input pin (TIQ01 pin) is detected. In the pulse width measurement mode, the count value of the 16-bit counter is stored in the TQ0CCR1 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIQ01 pin) is detected.

Even if the capture operation and reading the TQ0CCR1 register conflict, the correct value of the TQ0CCR1 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 8-3. 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) TMQ0 capture/compare register 2 (TQ0CCR2)

The TQ0CCR2 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 TQ0OPT0.TQ0CCS2 bit. In the pulse width measurement mode, the TQ0CCR2 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TQ0CCR2 register can be read or written during operation.

This register can be read or written in 16-bit units.

Reset input clears this register to 0000H.

Caution Accessing the TQ0CCR2 register is disabled during subclock operation with the main clock stopped. For details, refer to 3.4.8 (2).

After reset: 0000H R/W Address: FFFFF54AH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TQ0CCR2																

(a) Function as compare register

The TQ0CCR2 register can be rewritten even when the TQ0CTL0.TQ0CE bit = 1.

The set value of the TQ0CCR2 register is transferred to the CCR2 buffer register. When the value of the 16-bit counter matches the value of the CCR2 buffer register, a compare match interrupt request signal (INTTQ0CC2) is generated. If TOQ02 pin output is enabled at this time, the output of the TOQ02 pin is inverted.

(b) Function as capture register

When the TQ0CCR2 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 TQ0CCR2 register if the valid edge of the capture trigger input pin (TIQ02 pin) is detected. In the pulse width measurement mode, the count value of the 16-bit counter is stored in the TQ0CCR2 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIQ02 pin) is detected.

Even if the capture operation and reading the TQ0CCR2 register conflict, the correct value of the TQ0CCR2 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 8-4. 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	–

(10) TMQ0 capture/compare register 3 (TQ0CCR3)

The TQ0CCR3 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 TQ0OPT0.TQ0CCS3 bit. In the pulse width measurement mode, the TQ0CCR3 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TQ0CCR3 register can be read or written during operation.

This register can be read or written in 16-bit units.

Reset input clears this register to 0000H.

Caution Accessing the TQ0CCR3 register is disabled during subclock operation with the main clock stopped. For details, refer to 3.4.8 (2).

After reset: 0000H R/W Address: FFFFF54CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TQ0CCR3																

(a) Function as compare register

The TQ0CCR3 register can be rewritten even when the TQ0CTL0.TQ0CE bit = 1.

The set value of the TQ0CCR3 register is transferred to the CCR3 buffer register. When the value of the 16-bit counter matches the value of the CCR3 buffer register, a compare match interrupt request signal (INTTQ0CC3) is generated. If TOQ03 pin output is enabled at this time, the output of the TOQ03 pin is inverted.

(b) Function as capture register

When the TQ0CCR3 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 TQ0CCR3 register if the valid edge of the capture trigger input pin (TIQ03 pin) is detected. In the pulse width measurement mode, the count value of the 16-bit counter is stored in the TQ0CCR3 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIQ03 pin) is detected.

Even if the capture operation and reading the TQ0CCR3 register conflict, the correct value of the TQ0CCR3 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 8-5. 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	–

(11) TMQ0 counter read buffer register (TQ0CNT)

The TQ0CNT register is a read buffer register that can read the count value of the 16-bit counter.

If this register is read when the TQ0CTL0.TQ0CE 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 TQ0CNT register is cleared to 0000H when the TQ0CE bit = 0. If the TQ0CNT 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 TQ0CNT register is cleared to 0000H after reset, as the TQ0CE bit is cleared to 0.

Caution Accessing the TQ0CNT register is disabled during subclock operation with the main clock stopped. For details, refer to 3.4.8 (2).

After reset: 0000H R Address: FFFFF54EH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TQ0CNT																

8.5 Operation

TMQ0 can perform the following operations.

Operation	TQ0CTL1.TQ0EST Bit (Software Trigger Bit)	TIQ00 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 TIQ00 pin capture trigger input is not detected (by clearing the TQ0IOC1.TQ0IS1 and TQ0IOC1.TQ0IS0 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 TQ0CTL1.TQ0EEE bit to 0).

8.5.1 Interval timer mode (TQ0MD2 to TQ0MD0 = 000)

In the interval timer mode, an interrupt request signal (INTTQ0CC0) is generated at the specified interval if the TQ0CTL0.TQ0CE bit is set to 1. A square wave whose half cycle is equal to the interval can be output from the TOQ00 pin.

Usually, the TQ0CCR1 to TQ0CCR3 registers are not used in the interval timer mode.

Figure 8-2. Configuration in Interval Timer Mode

Figure 8-3. Basic Timing in Interval Timer Mode

When the TQ0CE 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 TOQ00 pin is inverted. Additionally, the set value of the TQ0CCR0 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 TOQ00 pin is inverted, and a compare match interrupt request signal (INTTQ0CC0) is generated.

The interval can be calculated by the following expression.

Interval = (Set value of TQ0CCR0 register + 1) × Count clock cycle

Figure 8-4. Register Setting in Interval Timer Mode (1/2)

Figure 8-4. Register Setting in Interval Timer Mode (2/2)

(c) TMQ0 I/O control register 0 (TQ0IOC0)

	TQ0OL3	TQ0OE3	TQ0OL2	TQ0OE2	TQ0OL1	TQ0OE1	TQ0OL0	TQ0OE0
TQ0IOC0	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

- TQ0OE0: 0: Disable TOQ00 pin output
1: Enable TOQ00 pin output
- TQ0OL0: Setting of output level with operation of TOQ00 pin disabled
0: Low level
1: High level
- TQ0OE1: 0: Disable TOQ01 pin output
1: Enable TOQ01 pin output
- TQ0OL1: Setting of output level with operation of TOQ01 pin disabled
0: Low level
1: High level
- TQ0OE2: 0: Disable TOQ02 pin output
1: Enable TOQ02 pin output
- TQ0OL2: Setting of output level with operation of TOQ02 pin disabled
0: Low level
1: High level
- TQ0OE3: 0: Disable TOQ03 pin output
1: Enable TOQ03 pin output
- TQ0OL3: Setting of output level with operation of TOQ03 pin disabled
0: Low level
1: High level

(d) TMQ0 counter read buffer register (TQ0CNT)

The count value of the 16-bit counter can be read by reading the TQ0CNT register.

(e) TMQ0 capture/compare register 0 (TQ0CCR0)

If the TQ0CCR0 register is set to D_0, the interval is as follows.

Interval = $(D_0 + 1) \times$ Count clock cycle

(f) TMQ0 capture/compare registers 1 to 3 (TQ0CCR1 to TQ0CCR3)

Usually, the TQ0CCR1 to TQ0CCR3 registers are not used in the interval timer mode. However, the set values of the TQ0CCR1 to TQ0CCR3 registers are transferred to the CCR1 to CCR3 buffer registers. The compare match interrupt request signals (INTTQ0CC1 to INTTQ0CC3) are generated when the count value of the 16-bit counter matches the value of the CCR1 to CCR3 buffer registers.

Therefore, mask the interrupt request by using the corresponding interrupt mask flags (TQ0CCMK1 to TQ0CCMK3).

Remark TMQ0 I/O control register 1 (TQ0IOC1), TMQ0 I/O control register 2 (TQ0IOC2), and TMQ0 option register 0 (TQ0OPT0) are not used in the interval timer mode.

(1) Operation flow in interval timer mode

Figure 8-5. Software Processing Flow in Interval Timer Mode

<1> Count operation start flow

Initial setting of these registers is performed before setting the TQ0CE bit to 1.

The TQ0CKS0 to TQ0CKS2 bits can be set at the same time when counting has been started (TQ0CE bit = 1).

<2> Count operation stop flow

The counter is initialized and counting is stopped by clearing the TQ0CE bit to 0.

(2) Operation timing in interval timer mode

(a) Operation if TQ0CCR0 register is set to 0000H

If the TQ0CCR0 register is set to 0000H, the INTTQ0CC0 signal is generated at each count clock, and the output of the TOQ00 pin is inverted.

The value of the 16-bit counter is always 0000H.

(b) Operation if TQ0CCR0 register is set to FFFFH

If the TQ0CCR0 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 INTTQ0CC0 signal is generated and the output of the TOQ00 pin is inverted. At this time, an overflow interrupt request signal (INTTQ0OV) is not generated, nor is the overflow flag (TQ0OPT0.TQ0OVF bit) set to 1.

(c) Notes on rewriting TQ0CCR0 register

To change the value of the TQ0CCR0 register to a smaller value, stop counting once and then change the set value.

If the value of the TQ0CCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow.

Remark Interval time (1): $(D_1 + 1) \times$ Count clock cycle
Interval time (NG): $(10000H + D_2 + 1) \times$ Count clock cycle
Interval time (2): $(D_2 + 1) \times$ Count clock cycle

If the value of the TQ0CCR0 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 TQ0CCR0 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 INTTQ0CC0 signal is generated and the output of the TOQ00 pin is inverted.

Therefore, the INTTQ0CC0 signal may not be generated at the interval time "$(D_1 + 1) \times$ Count clock cycle" or "$(D_2 + 1) \times$ Count clock cycle" originally expected, but may be generated at an interval of "$(10000H + D_2 + 1) \times$ Count clock cycle".

(d) Operation of TQ0CCR1 to TQ0CCR3 registers

Figure 8-6. Configuration of TQ0CCR1 to TQ0CCR3 Registers

If the set value of the TQ0CCRk register is less than the set value of the TQ0CCR0 register, the INTTQ0CCk signal is generated once per cycle. At the same time, the output of the TOQ0k pin is inverted. The TOQ0k pin outputs a square wave with the same cycle as that output by the TOQ00 pin.

Remark k = 1 to 3

Figure 8-7. Timing Chart When $D_{01} \geq D_{k1}$

If the set value of the TQ0CCRk register is greater than the set value of the TQ0CCR0 register, the count value of the 16-bit counter does not match the value of the TQ0CCRk register. Consequently, the INTTQ0CCk signal is not generated, nor is the output of the TOQ0k pin changed.

Remark k = 1 to 3

Figure 8-8. Timing Chart When $D_{01} < D_{k1}$

8.5.2 External event count mode (TQ0MD2 to TQ0MD0 = 001)

In the external event count mode, the valid edge of the external event count input is counted when the TQ0CTL0.TQ0CE bit is set to 1, and an interrupt request signal (INTTQ0CC0) is generated each time the specified number of edges have been counted. The TOQ00 pin cannot be used.

Usually, the TQ0CCR1 to TQ0CCR3 registers are not used in the external event count mode.

Figure 8-9. Configuration in External Event Count Mode

Figure 8-10. Basic Timing in External Event Count Mode

Caution This figure shows the basic timing when the rising edge is specified as the valid edge of the external event count input.

When the TQ0CE 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 TQ0CCR0 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 (INTTQ0CC0) is generated.

The INTTQ0CC0 signal is generated each time the valid edge of the external event count input has been detected (set value of TQ0CCR0 register + 1) times.

Figure 8-11. Register Setting in External Event Count Mode (1/2)

(a) TMQ0 control register 0 (TQ0CTL0)

	TQ0CE					TQ0CKS2	TQ0CKS1	TQ0CKS0
TQ0CTL0	0/1	0	0	0	0	0	0	0

0: Stop counting
1: Enable counting

(b) TMQ0 control register 1 (TQ0CTL1)

		TQ0EST	TQ0EEE			TQ0MD2	TQ0MD1	TQ0MD0
TQ0CTL1	0	0	0	0	0	0	0	1

0, 0, 1:
External event count mode

Figure 8-11. Register Setting in External Event Count Mode (2/2)

(c) TMQ0 I/O control register 0 (TQ0IOC0)

	TQ0OL3	TQ0OE3	TQ0OL2	TQ0OE2	TQ0OL1	TQ0OE1	TQ0OL0	TQ0OE0
TQ0IOC0	0/1	0/1	0/1	0/1	0/1	0/1	0	0

- TQ0OE0: 0: Disable TOQ00 pin output
- TQ0OE1: 0: Disable TOQ01 pin output; 1: Enable TOQ01 pin output
- TQ0OL1: Setting of output level with operation of TOQ01 pin disabled; 0: Low level; 1: High level
- TQ0OE2: 0: Disable TOQ02 pin output; 1: Enable TOQ02 pin output
- TQ0OL2: Setting of output level with operation of TOQ02 pin disabled; 0: Low level; 1: High level
- TQ0OE3: 0: Disable TOQ03 pin output; 1: Enable TOQ03 pin output
- TQ0OL3: Setting of output level with operation of TOQ03 pin disabled; 0: Low level; 1: High level

(d) TMQ0 I/O control register 2 (TQ0IOC2)

					TQ0EES1	TQ0EES0	TQ0ETS1	TQ0ETS0
TQ0IOC2	0	0	0	0	0/1	0/1	0	0

- TQ0EES1, TQ0EES0: Select valid edge of external event count input

(e) TMQ0 counter read buffer register (TQ0CNT)

The count value of the 16-bit counter can be read by reading the TQ0CNT register.

(f) TMQ0 capture/compare register 0 (TQ0CCR0)

If D_0 is set to the TQ0CCR0 register, the counter is cleared and a compare match interrupt request signal (INTTQ0CC0) is generated when the number of external event counts reaches (D_0 + 1).

(g) TMQ0 capture/compare registers 1 to 3 (TQ0CCR1 to TQ0CCR3)

Usually, the TQ0CCR1 to TQ0CCR3 registers are not used in the external event count mode. However, the set values of the TQ0CCR1 to TQ0CCR3 registers are transferred to the CCR1 to CCR3 buffer registers. When the count value of the 16-bit counter matches the value of the CCR1 to CCR3 buffer registers, compare match interrupt request signals (INTTQ0CC1 to INTTQ0CC3) are generated. Therefore, mask the interrupt signal by using the interrupt mask flags (TQ0CCMK1 to TQ0CCMK3).

Remark The TMQ0 I/O control register 1 (TQ0IOC1) and TMQ0 option register 0 (TQ0OPT0) are not used in the external event count mode.

(1) Operation flow in external event count mode

Figure 8-12. Software Processing Flow in External Event Count Mode

<1> Count operation start flow

Initial setting of these registers is performed before setting the TQ0CE bit to 1.

The TQ0CKS0 to TQ0CKS2 bits can be set at the same time when counting has been started (TQ0CE bit = 1).

<2> Count operation stop flow

The counter is initialized and counting is stopped by clearing the TQ0CE bit to 0.

(2) Operation timing in external event count mode

(a) Operation if TQ0CCR0 register is set to 0000H

If the TQ0CCR0 register is set to 0000H, the INTTQ0CC0 signal is generated each time the valid edge of the external event count signal has been detected.

The 16-bit counter is always 0000H.

(b) Operation if TQ0CCR0 register is set to FFFFH

If the TQ0CCR0 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 INTTQ0CC0 signal is generated. At this time, the TQ0OPT0.TQ0OVF bit is not set.

(c) Notes on rewriting the TQ0CCR0 register

To change the value of the TQ0CCR0 register to a smaller value, stop counting once and then change the set value.

If the value of the TQ0CCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow.

If the value of the TQ0CCR0 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 TQ0CCR0 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 INTTQ0CC0 signal is generated.

Therefore, the INTTQ0CC0 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”.

(d) Operation of TQ0CCR1 to TQ0CCR3 registers

Figure 8-13. Configuration of TQ0CCR1 to TQ0CCR3 Registers

If the set value of the TQ0CCRk register is smaller than the set value of the TQ0CCR0 register, the INTTQ0CCk signal is generated once per cycle. At the same time, the output signal of the TOQ0k pin is inverted.

Remark k = 1 to 3

Figure 8-14. Timing Chart When $D_{01} \geq D_{k1}$

If the set value of the TQ0CCRk register is greater than the set value of the TQ0CCR0 register, the INTTQ0CCk signal is not generated because the count value of the 16-bit counter and the value of the TQ0CCRk register do not match. Nor is the output signal of the TOQ0k pin changed.

Remark k = 1 to 3

Figure 8-15. Timing Chart When $D_{01} < D_{k1}$

8.5.3 External trigger pulse output mode (TQ0MD2 to TQ0MD0 = 010)

In the external trigger pulse output mode, 16-bit timer/event counter Q waits for a trigger when the TQ0CTL0.TQ0CE bit is set to 1. When the valid edge of an external trigger input signal is detected, 16-bit timer/event counter Q starts counting, and outputs a PWM waveform from the TOQ01 to TOQ03 pins.

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 TOQ00 pin.

Figure 8-16. Configuration in External Trigger Pulse Output Mode

Figure 8-17. Basic Timing in External Trigger Pulse Output Mode

16-bit timer/event counter Q waits for a trigger when the TQ0CE 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 TOQ0k pin.

If the trigger is generated again while the counter is operating, the counter is cleared to 0000H and restarted.

The active level width, cycle, and duty factor of the PWM waveform can be calculated as follows.

Active level width = (Set value of TQ0CCRk register) × Count clock cycle
Cycle = (Set value of TQ0CCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TQ0CCRk register)/(Set value of TQ0CCR0 register + 1)

The compare match interrupt request signal INTTQ0CC0 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 INTTQ0CCk is generated when the count value of the 16-bit counter matches the value of the CCRk buffer register.

The value set to the TQ0CCRm register is transferred to the CCRm buffer register 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 valid edge of an external trigger input signal, or setting the software trigger (TQ0CTL1.TQ0EST bit) to 1 is used as the trigger.

Remark k = 1 to 3, m = 0 to 3

Figure 8-18. Register Setting in External Trigger Pulse Output Mode (1/3)

(a) TMQ0 control register 0 (TQ0CTL0)

Note The setting is invalid when the TQ0CTL1.TQ0EEE bit = 1.

Figure 8-18. Register Setting in External Trigger Pulse Output Mode (2/3)

(b) TMQ0 control register 1 (TQ0CTL1)

	TQ0EST	TQ0EEE			TQ0MD2	TQ0MD1	TQ0MD0
TQ0CTL1	0	0/1	0/1	0	0	0	1	0

0, 1, 0:
External trigger pulse output mode

0: Operate on count clock selected by TQ0CKS0 to TQ0CKS2 bits
1: Count with external event input signal

Generate software trigger when 1 is written

(c) TMQ0 I/O control register 0 (TQ0IOC0)

	TQ0OL3	TQ0OE3	TQ0OL2	TQ0OE2	TQ0OL1	TQ0OE1	TQ0OL0	TQ0OE0
TQ0IOC0	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

0: Disable TOQ00 pin output
1: Enable TOQ00 pin output

Setting of output level while operation of TOQ00 pin is disabled
0: Low level
1: High level

0: Disable TOQ01 pin output
1: Enable TOQ01 pin output

Specification of active level of TOQ01 pin output
0: Active-high
1: Active-low

0: Disable TOQ02 pin output
1: Enable TOQ02 pin output

Specification of active level of TOQ02 pin output
0: Active-high
1: Active-low

0: Disable TOQ03 pin output
1: Enable TOQ03 pin output

Specification of active level of TOQ03 pin output
0: Active-high
1: Active-low

- When TQ0OLk bit = 0

- When TQ0OLk bit = 1

Figure 8-18. Register Setting in External Trigger Pulse Output Mode (3/3)

(d) TMQ0 I/O control register 2 (TQ0IOC2)

					TQ0EES1	TQ0EES0	TQ0ETS1	TQ0ETS0
TQ0IOC2	0	0	0	0	0/1	0/1	0/1	0/1

Select valid edge of external trigger input

Select valid edge of external event count input

(e) TMQ0 counter read buffer register (TQ0CNT)

The value of the 16-bit counter can be read by reading the TQ0CNT register.

(f) TMQ0 capture/compare registers 0 to 3 (TQ0CCR0 to TQ0CCR3)

If D_0 is set to the TQ0CCR0 register, D_1 to the TQ0CCR1 register, D_2 to the TQ0CCR2 register, and D_3 to the TQ0CCR3 register, the cycle and active level width of the PWM waveform are as follows.

Cycle = $(D_0 + 1) \times$ Count clock cycle

TOQ01 pin PWM waveform active level width = $D_1 \times$ Count clock cycle

TOQ02 pin PWM waveform active level width = $D_2 \times$ Count clock cycle

TOQ03 pin PWM waveform active level width = $D_3 \times$ Count clock cycle

Remark TMQ0 I/O control register 1 (TQ0IOC1) and TMQ0 option register 0 (TQ0OPT0) are not used in the external trigger pulse output mode.

(1) Operation flow in external trigger pulse output mode

Figure 8-19. Software Processing Flow in External Trigger Pulse Output Mode (1/2)

Figure 8-19. Software Processing Flow in External Trigger Pulse Output Mode (2/2)

Remark m = 0 to 3

(2) Operation timing in external trigger pulse output mode

(a) Note on changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TQ0CCR1 register last.

Rewrite the TQ0CCRk register after writing the TQ0CCR1 register after the INTTQ0CC0 signal is detected.

In order to transfer data from the TQ0CCRm register to the CCRm buffer register, the TQ0CCR1 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 TQ0CCR0 register, set the active level width to the TQ0CCR2 and TQ0CCR3 registers, and then set an active level width to the TQ0CCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TQ0CCR0 register, and then write the same value to the TQ0CCR1 register.

To change only the active level width (duty factor) of the PWM waveform, first set an active level width to the TQ0CCR2 and TQ0CCR3 registers and then set an active level width to the TQ0CCR1 register.

To change only the active level width (duty factor) of the PWM waveform output by the TOQ01 pin, only the TQ0CCR1 register has to be set.

To change only the active level width (duty factor) of the PWM waveform output by the TOQ02 and TOQ03 pins, first set an active level width to the TQ0CCR2 and TQ0CCR3 registers, and then write the same value to the TQ0CCR1 register.

After data is written to the TQ0CCR1 register, the value written to the TQ0CCRm 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 TQ0CCR0 to TQ0CCR3 registers again after writing the TQ0CCR1 register once, do so after the INTTQ0CC0 signal is generated. Otherwise, the value of the CCRm buffer register may become undefined because timing of transferring data from the TQ0CCRm register to the CCRm buffer register conflicts with writing the TQ0CCRm register.

Remark m = 0 to 3

(b) 0%/100% output of PWM waveform

To output a 0% waveform, set the TQ0CCRk register to 0000H. If the set value of the TQ0CCR0 register is FFFFH, the INTTQ0CCk signal is generated periodically.

Remark k = 1 to 3

To output a 100% waveform, set a value of (set value of TQ0CCR0 register + 1) to the TQ0CCRk register. If the set value of the TQ0CCR0 register is FFFFH, 100% output cannot be produced.

Remark k = 1 to 3

(c) Conflict between trigger detection and match with CCRk buffer register

If the trigger is detected immediately after the INTTQ0CCk signal is generated, the 16-bit counter is immediately cleared to 0000H, the output signal of the TOQ0k pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.

Remark k = 1 to 3

If the trigger is detected immediately before the INTTQ0CCk signal is generated, the INTTQ0CCk signal is not generated, and the 16-bit counter is cleared to 0000H and continues counting. The output signal of the TOQ0k pin remains active. Consequently, the active period of the PWM waveform is extended.

Remark k = 1 to 3

(d) Conflict between trigger detection and match with CCR0 buffer register

If the trigger is detected immediately after the INTTQ0CC0 signal is generated, the 16-bit counter is cleared to 0000H and continues counting up. Therefore, the active period of the TOQ0k pin is extended by time from generation of the INTTQ0CC0 signal to trigger detection.

Remark k = 1 to 3

If the trigger is detected immediately before the INTTQ0CC0 signal is generated, the INTTQ0CC0 signal is not generated. The 16-bit counter is cleared to 0000H, the TOQ0k pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.

Remark k = 1 to 3

(e) Generation timing of compare match interrupt request signal (INTTQ0CCk)

The timing of generation of the INTTQ0CCk signal in the external trigger pulse output mode differs from the timing of other INTTQ0CCk signals; the INTTQ0CCk signal is generated when the count value of the 16-bit counter matches the value of the CCRk buffer register.

Remark k = 1 to 3

Usually, the INTTQ0CCk signal is generated in synchronization with the next count up after the count value of the 16-bit counter matches the value of the CCRk buffer 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 TOQ0k pin.

8.5.4 One-shot pulse output mode (TQ0MD2 to TQ0MD0 = 011)

In the one-shot pulse output mode, 16-bit timer/event counter Q waits for a trigger when the TQ0CTL0.TQ0CE bit is set to 1. When the valid edge of an external trigger input is detected, 16-bit timer/event counter Q starts counting, and outputs a one-shot pulse from the TOQ01 to TOQ03 pins.

Instead of the external trigger, a software trigger can also be generated to output the pulse. When the software trigger is used, the TOQ00 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 8-20. Configuration in One-Shot Pulse Output Mode

Figure 8-21. Basic Timing in One-Shot Pulse Output Mode

When the TQ0CE bit is set to 1, 16-bit timer/event counter Q 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 TOQ0k 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 TQ0CCRk register) × Count clock cycle

Active level width = (Set value of TQ0CCR0 register − Set value of TQ0CCRk register + 1) × Count clock cycle

The compare match interrupt request signal INTTQ0CC0 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 INTTQ0CCk is generated when the count value of the 16-bit counter matches the value of the CCRk buffer register.

The valid edge of an external trigger input or setting the software trigger (TQ0CTL1.TQ0EST bit) to 1 is used as the trigger.

Remark k = 1 to 3

Figure 8-22. Register Setting in One-Shot Pulse Output Mode (1/3)

Figure 8-22. Register Setting in One-Shot Pulse Output Mode (2/3)

(c) TMQ0 I/O control register 0 (TQ0IOC0)

	TQ0OL3	TQ0OE3	TQ0OL2	TQ0OE2	TQ0OL1	TQ0OE1	TQ0OL0	TQ0OE0
TQ0IOC0	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

- TQ0OE0: 0: Disable TOQ00 pin output; 1: Enable TOQ00 pin output
- TQ0OL0: Setting of output level while operation of TOQ00 pin is disabled; 0: Low level; 1: High level
- TQ0OE1: 0: Disable TOQ01 pin output; 1: Enable TOQ01 pin output
- TQ0OL1: Specification of active level of TOQ01 pin output; 0: Active-high; 1: Active-low
- TQ0OE2: 0: Disable TOQ02 pin output; 1: Enable TOQ02 pin output
- TQ0OL2: Specification of active level of TOQ02 pin output; 0: Active-high; 1: Active-low
- TQ0OE3: 0: Disable TOQ03 pin output; 1: Enable TOQ03 pin output
- TQ0OL3: Specification of active level of TOQ03 pin output; 0: Active-high; 1: Active-low

• When TQ0OLk bit = 0

• When TQ0OLk bit = 1

(d) TMQ0 I/O control register 2 (TQ0IOC2)

					TQ0EES1	TQ0EES0	TQ0ETS1	TQ0ETS0
TQ0IOC2	0	0	0	0	0/1	0/1	0/1	0/1

- TQ0ETS1, TQ0ETS0: Select valid edge of external trigger input
- TQ0EES1, TQ0EES0: Select valid edge of external event count input

Figure 8-22. Register Setting in One-Shot Pulse Output Mode (3/3)

(e) TMQ0 counter read buffer register (TQ0CNT)

The value of the 16-bit counter can be read by reading the TQ0CNT register.

(f) TMQ0 capture/compare registers 0 to 3 (TQ0CCR0 to TQ0CCR3)

If D_0 is set to the TQ0CCR0 register and D_k to the TQ0CCRk register, the active level width and output delay period of the one-shot pulse are as follows.

Active level width = $(D_k - D_0 + 1) \times$ Count clock cycle

Output delay period = $D_k \times$ Count clock cycle

Remarks 1. TMQ0 I/O control register 1 (TQ0IOC1) and TMQ0 option register 0 (TQ0OPT0) are not used in the one-shot pulse output mode.

2. k = 1 to 3

(1) Operation flow in one-shot pulse output mode

Figure 8-23. Software Processing Flow in One-Shot Pulse Output Mode (1/2)

Figure 8-23. Software Processing Flow in One-Shot Pulse Output Mode (2/2)

(2) Operation timing in one-shot pulse output mode

(a) Note on rewriting TQ0CCRm register

To change the set value of the TQ0CCRm register to a smaller value, stop counting once, and then change the set value.

If the value of the TQ0CCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow.

When the TQ0CCR0 register is rewritten from D_{00} to D_{01} and the TQ0CCRk register from D_{k0} to D_{k1} where $D_{00} > D_{01}$ and $D_{k0} > D_{k1}$, if the TQ0CCRk register is rewritten when the count value of the 16-bit counter is greater than D_{k1} and less than D_{k0} and if the TQ0CCR0 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_{k1}, the counter generates the INTTQ0CCk signal and asserts the TOQ0k pin. When the count value matches D_{01}, the counter generates the INTTQ0CC0 signal, deasserts the TOQ0k 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 k = 1 to 3

(b) Generation timing of compare match interrupt request signal (INTTQ0CCk)

The generation timing of the INTTQ0CCk signal in the one-shot pulse output mode is different from other INTTQ0CCk signals; the INTTQ0CCk signal is generated when the count value of the 16-bit counter matches the value of the TQ0CCRk register.

Usually, the INTTQ0CCk signal is generated when the 16-bit counter counts up next time after its count value matches the value of the TQ0CCRk 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 TOQ0k pin.

Remark k = 1 to 3

8.5.5 PWM output mode (TQ0MD2 to TQ0MD0 = 100)

In the PWM output mode, a PWM waveform is output from the TOQ01 to TOQ03 pins when the TQ0CTL0.TQ0CE bit is set to 1.

In addition, a pulse with one cycle of the PWM waveform as half its cycle is output from the TOQ00 pin.

Figure 8-24. Configuration in PWM Output Mode

Figure 8-25. Basic Timing in PWM Output Mode

When the TQ0CE bit is set to 1, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs PWM waveform from the TOQ0k pin.

The active level width, cycle, and duty factor of the PWM waveform can be calculated as follows.

Active level width = (Set value of TQ0CCRk register) × Count clock cycle
Cycle = (Set value of TQ0CCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TQ0CCRk register)/(Set value of TQ0CCR0 register + 1)

The PWM waveform can be changed by rewriting the TQ0CCRm 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 INTTQ0CC0 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 INTTQ0CCk is generated when the count value of the 16-bit counter matches the value of the CCRk buffer register.

Remark k = 1 to 3, m = 0 to 3

Figure 8-26. Register Setting in PWM Output Mode (1/3)

(a) TMQ0 control register 0 (TQ0CTL0)

	TQ0CE					TQ0CKS2	TQ0CKS1	TQ0CKS0
TQ0CTL0	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 TQ0CTL1.TQ0EEE bit = 1.

(b) TMQ0 control register 1 (TQ0CTL1)

		TQ0EST	TQ0EEE			TQ0MD2	TQ0MD1	TQ0MD0
TQ0CTL1	0	0	0	0	0	1	0	0

1, 0, 0:
PWM output mode

Figure 8-26. Register Setting in PWM Output Mode (2/3)

(c) TMQ0 I/O control register 0 (TQ0IOC0)

	TQ0OL3	TQ0OE3	TQ0OL2	TQ0OE2	TQ0OL1	TQ0OE1	TQ0OL0	TQ0OE0
TQ0IOC0	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

- TQ0OE0: 0: Disable TOQ00 pin output; 1: Enable TOQ00 pin output
- TQ0OL0: Setting of output level while operation of TOQ00 pin is disabled; 0: Low level; 1: High level
- TQ0OE1: 0: Disable TOQ01 pin output; 1: Enable TOQ01 pin output
- TQ0OL1: Specification of active level of TOQ01 pin output; 0: Active-high; 1: Active-low
- TQ0OE2: 0: Disable TOQ02 pin output; 1: Enable TOQ02 pin output
- TQ0OL2: Specification of active level of TOQ02 pin output; 0: Active-high; 1: Active-low
- TQ0OE3: 0: Disable TOQ03 pin output; 1: Enable TOQ03 pin output
- TQ0OL3: Specification of active level of TOQ03 pin output; 0: Active-high; 1: Active-low

- When TQ0OLk bit = 0

- When TQ0OLk bit = 1

(d) TMQ0 I/O control register 2 (TQ0IOC2)

					TQ0EES1	TQ0EES0	TQ0ETS1	TQ0ETS0
TQ0IOC2	0	0	0	0	0/1	0/1	0	0

- TQ0EES1, TQ0EES0: Select valid edge of external event count input.

(e) TMQ0 counter read buffer register (TQ0CNT)

The value of the 16-bit counter can be read by reading the TQ0CNT register.

Figure 8-26. Register Setting in PWM Output Mode (3/3)

(f) TMQ0 capture/compare registers 0 to 3 (TQ0CCR0 to TQ0CCR3)

If D_0 is set to the TQ0CCR0 register and D_k to the TQ0CCRk register, the cycle and active level width of the PWM waveform are as follows.

Cycle = $(D_0 + 1) \times$ Count clock cycle

Active level width = $D_k \times$ Count clock cycle

Remark TMQ0 I/O control register 1 (TQ0IOC1) and TMQ0 option register 0 (TQ0OPT0) are not used in the PWM output mode.

(1) Operation flow in PWM output mode

Figure 8-27. Software Processing Flow in PWM Output Mode (1/2)

Figure 8-27. Software Processing Flow in PWM Output Mode (2/2)

Remark k = 1 to 3
m = 0 to 3

(2) Operation timing in PWM output mode

(a) Changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TQ0CCR1 register last.

Rewrite the TQ0CCRk register after writing the TQ0CCR1 register after the INTTQ0CC1 signal is detected.

To transfer data from the TQ0CCRm register to the CCRm buffer register, the TQ0CCR1 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 TQ0CCR0 register, set the active level width to the TQ0CCR2 and TQ0CCR3 registers, and then set an active level width to the TQ0CCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TQ0CCR0 register, and then write the same value to the TQ0CCR1 register.

To change only the active level width (duty factor) of the PWM waveform, first set the active level width to the TQ0CCR2 and TQ0CCR3 registers, and then set an active level width to the TQ0CCR1 register.

To change only the active level width (duty factor) of the PWM waveform output by the TOQ01 pin, only the TQ0CCR1 register has to be set.

To change only the active level width (duty factor) of the PWM waveform output by the TOQ02 and TOQ03 pins, first set an active level width to the TQ0CCR2 and TQ0CCR3 registers, and then write the same value to the TQ0CCR1 register.

After the TQ0CCR1 register is written, the value written to the TQ0CCRm register is transferred to the CCRm buffer register in synchronization with the timing of clearing the 16-bit counter, and is used as a value to be compared with the value of the 16-bit counter.

To write the TQ0CCR0 to TQ0CCR3 registers again after writing the TQ0CCR1 register once, do so after the INTTQ0CC0 signal is generated. Otherwise, the value of the CCRm buffer register may become undefined because the timing of transferring data from the TQ0CCRm register to the CCRm buffer register conflicts with writing the TQ0CCRm register.

Remark m = 0 to 3

(b) 0%/100% output of PWM waveform

To output a 0% waveform, set the TQ0CCRk register to 0000H. If the set value of the TQ0CCR0 register is FFFFH, the INTTQ0CCk signal is generated periodically.

Remark k = 1 to 3

To output a 100% waveform, set a value of (set value of TQ0CCR0 register + 1) to the TQ0CCRk register. If the set value of the TQ0CCR0 register is FFFFH, 100% output cannot be produced.

Remark k = 1 to 3

(c) Generation timing of compare match interrupt request signal (INTTQ0CCk)

The timing of generation of the INTTQ0CCk signal in the PWM output mode differs from the timing of other INTTQ0CCk signals; the INTTQ0CCk signal is generated when the count value of the 16-bit counter matches the value of the TQ0CCRk register.

Remark k = 1 to 3

Usually, the INTTQ0CCk signal is generated in synchronization with the next counting up after the count value of the 16-bit counter matches the value of the TQ0CCRk 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 TOQ0k pin.

8.5.6 Free-running timer mode (TQ0MD2 to TQ0MD0 = 101)

In the free-running timer mode, 16-bit timer/event counter Q starts counting when the TQ0CTL0.TQ0CE bit is set to 1. At this time, the TQ0CCRm register can be used as a compare register or a capture register, depending on the setting of the TQ0OPT0.TQ0CCS0 and TQ0OPT0.TQ0CCS1 bits.

Remark m = 0 to 3

Figure 8-28. Configuration in Free-Running Timer Mode

When the TQ0CE bit is set to 1, 16-bit timer/event counter Q starts counting, and the output signals of the TOQ00 to TOQ03 pins are inverted. When the count value of the 16-bit counter later matches the set value of the TQ0CCRm register, a compare match interrupt request signal (INTTQ0CCm) is generated, and the output signal of the TOQ0m 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 (INTTQ0OV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TQ0OPT0.TQ0OVF bit) is also set to 1. Clear the overflow flag to 0 by executing the CLR instruction by software.

The TQ0CCRm 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 8-29. Basic Timing in Free-Running Timer Mode (Compare Function)

When the TQ0CE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIQ0m pin is detected, the count value of the 16-bit counter is stored in the TQ0CCRm register, and a capture interrupt request signal (INTTQ0CCm) 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 (INTTQ0OV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TQ0OVF bit) is also set to 1. Clear the overflow flag to 0 by executing the CLR instruction by software.

Figure 8-30. Basic Timing in Free-Running Timer Mode (Capture Function)

Figure 8-31. Register Setting in Free-Running Timer Mode (1/3)

(a) TMQ0 control register 0 (TQ0CTL0)

	TQ0CE					TQ0CKS2	TQ0CKS1	TQ0CKS0
TQ0CTL0	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 TQ0CTL1.TQ0EEE bit = 1.

(b) TMQ0 control register 1 (TQ0CTL1)

		TQ0EST	TQ0EEE			TQ0MD2	TQ0MD1	TQ0MD0
TQ0CTL1	0	0	0/1	0	0	1	0	1

1, 0, 1:
Free-running mode

0: Operate on count clock selected by TQ0CKS0 to TQ0CKS2 bits
1: Count with external event count input signal

Figure 8-31. Register Setting in Free-Running Timer Mode (2/3)

(c) TMQ0 I/O control register 0 (TQ0IOC0)

	TQ0OL3	TQ0OE3	TQ0OL2	TQ0OE2	TQ0OL1	TQ0OE1	TQ0OL0	TQ0OE0
TQ0IOC0	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

- TQ0OE0: 0: Disable TOQ00 pin output
1: Enable TOQ00 pin output
- TQ0OL0: Setting of output level with operation of TOQ00 pin disabled
0: Low level
1: High level
- TQ0OE1: 0: Disable TOQ01 pin output
1: Enable TOQ01 pin output
- TQ0OL1: Setting of output level with operation of TOQ01 pin disabled
0: Low level
1: High level
- TQ0OE2: 0: Disable TOQ02 pin output
1: Enable TOQ02 pin output
- TQ0OL2: Setting of output level with operation of TOQ02 pin disabled
0: Low level
1: High level
- TQ0OE3: 0: Disable TOQ03 pin output
1: Enable TOQ03 pin output
- TQ0OL3: Setting of output level with operation of TOQ03 pin disabled
0: Low level
1: High level

(d) TMQ0 I/O control register 1 (TQ0IOC1)

	TQ0IS7	TQ0IS6	TQ0IS5	TQ0IS4	TQ0IS3	TQ0IS2	TQ0IS1	TQ0IS0
TQ0IOC1	0	0	0	0	0/1	0/1	0/1	0/1

- TQ0IS0, TQ0IS1: Select valid edge of TIQ00 pin input
- TQ0IS2, TQ0IS3: Select valid edge of TIQ01 pin input
- TQ0IS4, TQ0IS5: Select valid edge of TIQ02 pin input
- TQ0IS6, TQ0IS7: Select valid edge of TIQ03 pin input

Figure 8-31. Register Setting in Free-Running Timer Mode (3/3)

(e) TMQ0 I/O control register 2 (TQ0IOC2)

					TQ0EES1	TQ0EES0	TQ0ETS1	TQ0ETS0
TQ0IOC2	0	0	0	0	0/1	0/1	0	0

Select valid edge of external event count input

(f) TMQ0 option register 0 (TQ0OPT0)

	TQ0CCS3	TQ0CCS2	TQ0CCS1	TQ0CCS0				TQ0OVF
TQ0OPT0	0/1	0/1	0/1	0/1	0	0	0	0/1

- TQ0OVF: Overflow flag
- TQ0CCS0: Specifies if TQ0CCR0 register functions as capture or compare register
- TQ0CCS1: Specifies if TQ0CCR1 register functions as capture or compare register
- TQ0CCS2: Specifies if TQ0CCR2 register functions as capture or compare register
- TQ0CCS3: Specifies if TQ0CCR3 register functions as capture or compare register

(g) TMQ0 counter read buffer register (TQ0CNT)

The value of the 16-bit counter can be read by reading the TQ0CNT register.

(h) TMQ0 capture/compare registers 0 to 3 (TQ0CCR0 to TQ0CCR3)

These registers function as capture registers or compare registers depending on the setting of the TQ0OPT0.TQ0CCSm 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 TIQ0m pin is detected.

When the registers function as compare registers and when D_m is set to the TQ0CCRm register, the INTTQ0CCm signal is generated when the counter reaches (D_m + 1), and the output signal of the TOQ0m pin is inverted.

Remark m = 0 to 3

(1) Operation flow in free-running timer mode

(a) When using capture/compare register as compare register

Figure 8-32. Software Processing Flow in Free-Running Timer Mode (Compare Function) (1/2)

Figure 8-32. Software Processing Flow in Free-Running Timer Mode (Compare Function) (2/2)

(b) When using capture/compare register as capture register

Figure 8-33. Software Processing Flow in Free-Running Timer Mode (Capture Function) (1/2)

Figure 8-33. Software Processing Flow in Free-Running Timer Mode (Capture Function) (2/2)

(2) Operation timing in free-running timer mode

(a) Interval operation with compare register

When 16-bit timer/event counter Q is used as an interval timer with the TQ0CCRm 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 INTTQ0CCm signal has been detected.

When performing an interval operation in the free-running timer mode, four intervals can be set.
To perform the interval operation, the value of the corresponding TQ0CCRm register must be re-set in the interrupt servicing that is executed when the INTTQ0CCm signal is detected.
The set value for re-setting the TQ0CCRm 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 m = 0 to 3

(b) Pulse width measurement with capture register

When pulse width measurement is performed with the TQ0CCRm register used as a capture register, software processing is necessary for reading the capture register each time the INTTQ0CCm signal has been detected and for calculating an interval.

When executing pulse width measurement in the free-running timer mode, four pulse widths can be measured.

To measure a pulse width, the pulse width can be calculated by reading the value of the TQ0CCRm register in synchronization with the INTTQ0CCm signal, and calculating the difference between the read value and the previously read value.

Remark m = 0 to 3

(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 TQ0CCR0 register (setting of the default value of the TIQ00 pin input).
<2> Read the TQ0CCR1 register (setting of the default value of the TIQ01 pin input).
<3> Read the TQ0CCR0 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 TQ0CCR1 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 TQ0OVF0 and TQ0OVF1 flags are set on the internal RAM by software.

<1> Read the TQ0CCR0 register (setting of the default value of the TIQ00 pin input).

<2> Read the TQ0CCR1 register (setting of the default value of the TIQ01 pin input).

<3> An overflow occurs. Set the TQ0OVF0 and TQ0OVF1 flags to 1 in the overflow interrupt servicing, and clear the overflow flag to 0.

<4> Read the TQ0CCR0 register.
Read the TQ0OVF0 flag. If the TQ0OVF0 flag is 1, clear it to 0.
Because the TQ0OVF0 flag is 1, the pulse width can be calculated by (10000H + D_{01} – D_{00}).

<5> Read the TQ0CCR1 register.
Read the TQ0OVF1 flag. If the TQ0OVF1 flag is 1, clear it to 0 (the TQ0OVF0 flag is cleared in <4>, and the TQ0OVF1 flag remains 1).
Because the TQ0OVF1 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 TQ0OVF0 and TQ0OVF1 flags are set on the internal RAM by software.

<1> Read the TQ0CCR0 register (setting of the default value of the TIQ00 pin input).

<2> Read the TQ0CCR1 register (setting of the default value of the TIQ01 pin input).

<3> An overflow occurs. Nothing is done by software.

<4> Read the TQ0CCR0 register.

Read the overflow flag. If the overflow flag is 1, set only the TQ0OVF1 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 TQ0CCR1 register.

Read the overflow flag. Because the overflow flag is cleared in <4>, 0 is read.

Read the TQ0OVF1 flag. If the TQ0OVF1 flag is 1, clear it to 0.

Because the TQ0OVF1 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 a long pulse width is measured in the free-running timer mode.

<1> Read the TQ0CCRm register (setting of the default value of the TIQ0m 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 TQ0CCRm 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 below.

Example when capture trigger interval is long

Note The overflow counter is set arbitrarily by software on the internal RAM.

<1> Read the TQ0CCRm register (setting of the default value of the TIQ0m 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 TQ0CCRm register.

Read the overflow counter.

→ When the overflow counter is “N”, the pulse width can be calculated by ($N \times 10000H + D_{m1} - D_{m0}$).

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 TQ0OVF bit to 0 with the CLR instruction and by writing 8-bit data (bit 0 is 0) to the TQ0OPT0 register. To accurately detect an overflow, read the TQ0OVF bit when it is 1, and then clear the overflow flag by using a bit manipulation instruction.

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.

8.5.7 Pulse width measurement mode (TQ0MD2 to TQ0MD0 = 110)

In the pulse width measurement mode, 16-bit timer/event counter Q starts counting when the TQ0CTL0.TQ0CE bit is set to 1. Each time the valid edge input to the TIQ0m pin has been detected, the count value of the 16-bit counter is stored in the TQ0CCRm register, and the 16-bit counter is cleared to 0000H.

The interval of the valid edge can be measured by reading the TQ0CCRm register after a capture interrupt request signal (INTTQ0CCm) occurs.

Select either of the TIQ00 to TIQ03 pins as the capture trigger input pin. Specify “No edge detected” by using the TQ0IOC1 register for the unused pins.

When an external clock is used as the count clock, measure the pulse width of the TIQ0k pin because the external clock is fixed to the TIQ00 pin. At this time, clear the TQ0IOC1.TQ0IS1 and TQ0IOC1.TQ0IS0 bits to 00 (capture trigger input (TIQ00 pin): No edge detected).

Remark m = 0 to 3
k = 1 to 3

Figure 8-34. Configuration in Pulse Width Measurement Mode

Figure 8-35. Basic Timing in Pulse Width Measurement Mode

Remark m = 0 to 3

When the TQ0CE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIQ0m pin is later detected, the count value of the 16-bit counter is stored in the TQ0CCRm register, the 16-bit counter is cleared to 0000H, and a capture interrupt request signal (INTTQ0CCm) is generated.

The pulse width is calculated as follows.

Pulse width = $(D_N + 1) \times$ Count clock cycle

If the valid edge is not input to the TIQ0m pin even when the 16-bit counter counted up to FFFFH, an overflow interrupt request signal (INTTQ0OV) is generated at the next count clock, and the counter is cleared to 0000H and continues counting. At this time, the overflow flag (TQ0OPT0.TQ0OVF 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 = $(D_N + 10001H) \times$ Count clock cycle

Remark m = 0 to 3

Figure 8-36. Register Setting in Pulse Width Measurement Mode (1/2)

(a) TMQ0 control register 0 (TQ0CTL0)

	TQ0CE					TQ0CKS2	TQ0CKS1	TQ0CKS0
TQ0CTL0	0/1	0	0	0	0	0/1	0/1	0/1

Select count clock[Note]

0: Stop counting
1: Enable counting

Note Setting is invalid when the TQ0EEE bit = 1.

(b) TMQ0 control register 1 (TQ0CTL1)

		TQ0EST	TQ0EEE			TQ0MD2	TQ0MD1	TQ0MD0
TQ0CTL1	0	0	0/1	0	0	1	1	0

1, 1, 0:
Pulse width measurement mode

0: Operate on count clock selected by TQ0CKS0 to TQ0CKS2 bits
1: Count with external event count input signal

(c) TMQ0 I/O control register 1 (TQ0IOC1)

	TQ0IS7	TQ0IS6	TQ0IS5	TQ0IS4	TQ0IS3	TQ0IS2	TQ0IS1	TQ0IS0
TQ0IOC1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

Select valid edge of TIQ00 pin input

Select valid edge of TIQ01 pin input

Select valid edge of TIQ02 pin input

Select valid edge of TIQ03 pin input

(d) TMQ0 I/O control register 2 (TQ0IOC2)

					TQ0EES1	TQ0EES0	TQ0ETS1	TQ0ETS0
TQ0IOC2	0	0	0	0	0/1	0/1	0	0

Select valid edge of external event count input

Figure 8-36. Register Setting in Pulse Width Measurement Mode (2/2)

(e) TMQ0 option register 0 (TQ0OPT0)

	TQ0CCS3	TQ0CCS2	TQ0CCS1	TQ0CCS0				TQ0OVF
TQ0OPT0	0	0	0	0	0	0	0	0/1

Overflow flag

(f) TMQ0 counter read buffer register (TQ0CNT)

The value of the 16-bit counter can be read by reading the TQ0CNT register.

(g) TMQ0 capture/compare registers 0 to 3 (TQ0CCR0 to TQ0CCR3)

These registers store the count value of the 16-bit counter when the valid edge input to the TIQ0m pin is detected.

Remarks 1. TMQ0 I/O control register 0 (TQ0IOC0) is not used in the pulse width measurement mode.

2. m = 0 to 3

(1) Operation flow in pulse width measurement mode

Figure 8-37. Software Processing Flow in Pulse Width Measurement Mode

<1> Count operation start flow

Initial setting of these registers is performed before setting the TQ0CE bit to 1.

The TQ0CKS0 to TQ0CKS2 bits can be set at the same time when counting has been started (TQ0CE bit = 1).

<2> Count operation stop flow

The counter is initialized and counting is stopped by clearing the TQ0CE bit to 0.

(2) Operation timing in pulse width measurement mode

(a) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TQ0OVF bit to 0 with the CLR instruction and by writing 8-bit data (bit 0 is 0) to the TQ0OPT0 register. To accurately detect an overflow, read the TQ0OVF bit when it is 1, and then clear the overflow flag by using a bit manipulation instruction.

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.

8.5.8 Timer output operations

The following table shows the operations and output levels of the TOQ00 to TOQ03 pins.

Table 8-6. Timer Output Control in Each Mode

Operation Mode	TOQ00 Pin	TOQ01 Pin	TOQ02 Pin	TOQ03 Pin
Interval timer mode	Square wave output			
External event count mode	Square wave output	–		
External trigger pulse output mode	Square wave output	External trigger pulse output	External trigger pulse output	External trigger pulse output
One-shot pulse output mode		One-shot pulse output	One-shot pulse output	One-shot pulse output
PWM output mode		PWM output	PWM output	PWM output
Free-running timer mode	Square wave output (only when compare function is used)			
Pulse width measurement mode	–			

Table 8-7. Truth Table of TOQ00 to TOQ03 Pins Under Control of Timer Output Control Bits

TQ0IOC0.TQ0OLm Bit	TQ0IOC0.TQ0OEm Bit	TQ0CTL0.TQ0CE Bit	Level of TOQ0m 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 m = 0 to 3

8.6 Eliminating Digital Noise on Timer Input Pin

The μPD703229Y and 70F3229Y have a digital noise eliminator for the timer input pins (TIQ00 to TIQ03).

The sampling clock for digital sampling can be selected from f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/16$, $f_{XX}/32$, and $f_{XX}/64$. Sampling is performed three times or twice.

(1) TMQ0 noise elimination register (Q0nNFC)

The Q0nNFC register is used to select the digital noise elimination clock for the TIQ00 to TIQ03 pins.

Digital noise is eliminated from the signal input from the timer input pin before the Q0nNFC register is set, and the signal is always output.

After setting the sampling clock (NFC2 to NFC0 bits) and the number of times of sampling (NFSTS bit), therefore, enable the timer operation after the time of "(Sampling clock) × (Number of times of sampling)[Note]" has elapsed.

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

Note The time of "(Sampling clock) × (Number of times of sampling)" is required until the digital noise eliminator is initialized after digital noise elimination conditions have been set.

After reset: 00H R/W Address: Q00NFC FFFFFB50H, Q01NFC FFFFFB54H, Q02NFC FFFFFB58H, Q03NFC FFFFFB5CH

Q0nNFC (n = 0 to 3)

7	6	5	4	3	2	1	0
0	NFSTS	0	0	0	NFC2	NFC1	NFC0

NFSTS	Sets number of times of sampling for digital noise elimination
0	Number of times of sampling = 3
1	Number of times of sampling = 2

- For TIQ0n pin of TMQ0n

NFSTS	NFC2	NFC1	NFC0	Sampling clock	Width of noise accurately eliminated[Note]
					f_{XX} = 20 MHz
0	0	0	0	f_{XX}	0.10 μs
0	0	0	1	$f_{XX}/2$	0.20 μs
0	0	1	0	$f_{XX}/4$	0.40 μs
0	0	1	1	$f_{XX}/16$	1.60 μs
0	1	0	0	$f_{XX}/32$	3.20 μs
0	1	0	1	$f_{XX}/64$	6.40 μs
1	0	0	0	f_{XX}	0.05 μs
1	0	0	1	$f_{XX}/2$	0.10 μs
1	0	1	0	$f_{XX}/4$	0.20 μs
1	0	1	1	$f_{XX}/16$	0.80 μs
1	1	0	0	$f_{XX}/32$	1.60 μs
1	1	0	1	$f_{XX}/64$	3.20 μs
Other than above				Setting prohibited	

Note The width of the noise that can be accurately eliminated is (Sampling clock) × (Number of times of sampling – 1). Even noise with a narrower width than this causes incorrect counting if it is input in synchronization with the sampling clock.

CHAPTER 9 16-BIT INTERVAL TIMER M (TMM)

9.1 Overview

- Interval function
- 8 clocks selectable
- 16-bit counter × 1
  (The 16-bit counter cannot be read during timer count operation.)
- Compare register × 1
  (The compare register cannot be written during timer counter operation.)
- Compare match interrupt × 1

Timer M supports only the clear & start mode. The free-running timer mode is not supported.

9.2 Configuration

TMM0 includes the following hardware.

Table 9-1. Configuration of TMM0

Item	Configuration
Timer register	16-bit counter
Register	TMM0 compare register 0 (TM0CMP0)
Control register	TMM0 control register 0 (TM0CTL0)

Figure 9-1. Block Diagram of TMM0

Remark f_{XX}: Main clock frequency
f_R: Ring-OSC clock frequency
f_{XT}: Subclock frequency
INTWT: Watch timer interrupt request signal

(1) 16-bit counter

This is a 16-bit counter that counts the internal clock.

The 16-bit counter cannot be read or written.

(2) TMM0 compare register 0 (TM0CMP0)

The TM0CMP0 register is a 16-bit compare register.

This register can be read or written in 16-bit units.

Reset input clears this register to 0000H.

The same value can always be written to the TM0CMP0 register by software.

TM0CMP0 register rewrite is prohibited when the TM0CTL0.TM0CE bit = 1.

After reset: 0000H R/W Address: FFFFF694H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TM0CMP0																

9.3 Control Register

(1) TMM0 control register 0 (TM0CTL0)

The TM0CTL0 register is an 8-bit register that controls the TMM0 operation.

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

The same value can always be written to the TM0CTL0 register by software.

After reset: 00H R/W Address: FFFFF690H

	<7>	6	5	4	3	2	1	0
TM0CTL0	TM0CE	0	0	0	0	TM0CKS2	TM0CKS1	TM0CKS0

TM0CE	Internal clock operation enable/disable specification
0	TMM0 operation disabled (16-bit counter reset asynchronously). Operation clock supply stopped.
1	TMM0 operation enabled. Operation clock supply started. TMM0 operation started.
The internal clock control and internal circuit reset for TMM0 are performed asynchronously with the TM0CE bit. When the TM0CE bit is cleared to 0, the internal clock of TMM0 is stopped (fixed to low level) and 16-bit counter is reset asynchronously.	

TM0CKS2	TM0CKS1	TM0CKS0	Count clock selection
0	0	0	f_{XX}
0	0	1	$f_{XX}/2$
0	1	0	$f_{XX}/4$
0	1	1	$f_{XX}/64$
1	0	0	$f_{XX}/512$
1	0	1	INTWT
1	1	0	$f_R/8$
1	1	1	f_{XT}

Cautions 1. Set the TM0CKS2 to TM0CKS0 bits when TM0CE bit = 0.
When changing the value of the TM0CE bit from 0 to 1, it is also possible to set the value of the TM0CKS2 to TM0CKS0 bits simultaneously.
2. Be sure to set bits 3 to 6 to 0.

Remark f_{XX}: Main clock frequency
f_R: Ring-OSC clock frequency
f_{XT}: Subclock frequency

9.4 Operation

★ 9.4.1 Interval timer mode

In the interval timer mode, an interrupt request signal (INTTM0EQ0) is generated at the specified interval if the TM0CTL0.TM0CE bit is set to 1.

Figure 9-2. Configuration of Interval Timer

Figure 9-3. Basic Timing in Interval Timer Mode

When the TM0CE 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.

When the count value of the 16-bit counter matches the value of the TM0CMP0 register, the 16-bit counter is cleared to 0000H, and a compare match interrupt request signal (INTTM0EQ0) is generated.

The interval can be calculated by the following expression.

Interval = (Set value of TM0CMP0 register + 1) × Count clock cycle

Figure 9-4. Register Setting in Interval Timer Mode

(a) TMM0 control register 0 (TM0CTL0)

(b) TMM0 compare register 0 (TM0CMP0)

If the TM0CMP0 register is set to D_0, the interval is as follows.

Interval = (D_0 + 1) × Count clock cycle

(1) Operation flow in interval timer mode

Figure 9-5. Software Processing Flow in Interval Timer Mode

<1> Count operation start flow

Initial setting of these registers is performed before setting the TM0CE bit to 1.

The TM0CKS0 to TM0CKS2 bits can be set at the same time when counting has been started (TM0CE bit = 1).

<2> Count operation stop flow

The counter is initialized and counting is stopped by clearing the TM0CE bit to 0.

(2) Operation timing in interval timer mode

(a) Operation if TM0CMP0 register is set to 0000H

If the TM0CMP0 register is set to 0000H, the INTTM0EQ0 signal is generated at each count clock. The value of the 16-bit counter is always 0000H.

(b) Operation if TM0CMP0 register is set to FFFFH

If the TM0CMP0 register is set to FFFFH, the 16-bit counter counts up to FFFFH. The 16-bit counter is cleared to 0000H in synchronization with the next count-up timing and the INTTM0EQ0 signal is generated.

Note The INTTM0EQ0 signal is also generated immediately after the operation is enabled.

★ 9.4.2 Cautions

(1) It takes the 16-bit counter up to the following time to start counting after the TM0CTL0.TM0CE bit is set to 1, depending on the count clock selected.

Selected Count Clock	Maximum Time Before Counting Start
f_{XX}	$2/f_{XX}$
$f_{XX}/2$	$6/f_{XX}$
$f_{XX}/4$	$24/f_{XX}$
$f_{XX}/64$	$128/f_{XX}$
$f_{XX}/512$	$1024/f_{XX}$
INTWT	Second rising edge of INTWT signal
$f_R/8$	$16/f_R$
f_{XT}	$2/f_{XT}$

(2) Rewriting the TM0CMP0 and TM0CTL0 registers is prohibited while TMM0 is operating.
If these registers are rewritten while the TM0CE bit is 1, the operation cannot be guaranteed.
If they are rewritten by mistake, clear the TM0CTL0.TM0CE bit to 0, and re-set the registers.

(3) When the TM0CMP0 register is set to FFFFH, an interrupt is generated immediately after the operation is enabled.

CHAPTER 10 WATCH TIMER FUNCTIONS

10.1 Functions

The watch timer has the following functions.

- Watch timer: An interrupt request signal (INTWT) is generated at intervals of 0.5 or 0.25 seconds by using the main clock or subclock.
- Interval timer: An interrupt request signal (INTWTI) is generated at set intervals.

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

★ 10.2 Configuration

The block diagram of the watch timer is shown below.

Caution The prescaler of the watch timer functions alternately as the baud rate generator of CSIB0.

Figure 10-1. Block Diagram of Watch Timer

Remark f_X: Main clock oscillation frequency
f_{BGCS}: Watch timer source clock frequency
f_{BRG}: Watch timer count clock frequency
f_{XT}: Subclock frequency
f_W: Watch timer clock frequency
INTWT: Watch timer interrupt request signal
INTWTI: Interval timer interrupt request signal

(1) Clock control

This block controls supplying and stopping the operating clock (f_X) when the watch timer operates on the main clock.

(2) 3-bit prescaler

This prescaler divides f_X to generate $f_X/2$, $f_X/4$, or $f_X/8$.

(3) 8-bit counter

This 8-bit counter counts the source clock (f_{BGCS}).

(4) 11-bit prescaler

This prescaler divides f_W to generate a clock of $f_W/2^4$ to $f_W/2^{11}$.

(5) 5-bit counter

This counter counts f_W or $f_W/2^9$, and generates a watch timer interrupt request signal at intervals of $2^4/f_W$, $2^5/f_W$, $2^{13}/f_W$, or $2^{14}/f_W$.

(6) Selector

The watch timer has the following five selectors.

- Selector that selects one of f_X, $f_X/2$, $f_X/4$, or $f_X/8$ as the source clock of the watch timer
- Selector that selects the main clock (f_X) or subclock (f_{XT}) as the clock of the watch timer
- Selector that selects f_W or $f_W/2^9$ as the count clock frequency of the 5-bit counter
- Selector that selects $2^4/f_W$, $2^{13}/f_W$, $2^5/f_W$, or $2^{14}/f_W$ as the INTWT signal generation time interval
- Selector that selects $2^4/f_W$ to $2^{11}/f_W$ as the interval timer interrupt request signal (INTWTI) generation time interval

(7) PRSCM register

This is an 8-bit compare register that sets the interval time.

(8) PRSM register

This register controls clock supply to the watch timer.

(9) WTM register

This is an 8-bit register that controls the operation of the watch timer/interval timer, and sets the interrupt request signal generation interval.

10.3 Registers

The following registers are provided for the watch timer.

- Prescaler mode register 0 (PRSM0)
- Prescaler compare register 0 (PRSCM0)
- Watch timer operation mode register (WTM)

(1) Prescaler mode register 0 (PRSM0)

The PRSM0 register controls the generation of the watch timer count clock.
This register can be read or written in 8-bit or 1-bit units.
Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF8B0H

	7	6	5	<4>	3	2	1	0
PRSM0	0	0	0	BGCE0	0	0	BGCS01	BGCS00

BGCE0	Main clock operation enable
0	Disabled
1	Enabled

BGCS01	BGCS00	Selection of watch timer source clock (f_{BGCS})		
			5 MHz	4 MHz
0	0	f_X	50 ns	250 ns
0	1	$f_X/2$	100 ns	500 ns
1	0	$f_X/4$	200 ns	1 μs
1	1	$f_X/8$	400 ns	2 μs

Cautions 1. Do not change the values of the BGCS00 and BGCS01 bits during watch timer operation.
2. Set the PRSM0 register before setting the BGCE0 bit to 1.
3. Set the PRSM0 and PRSCM0 registers according to the main clock frequency that is used so as to obtain an f_{BRG} frequency of 32.768 kHz.

(2) Prescaler compare register 0 (PRSCM0)

The PRSCM0 register is an 8-bit compare register.

This register can be read or written in 8-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF8B1H

	7	6	5	4	3	2	1	0
PRSCM0	PRSCM07	PRSCM06	PRSCM05	PRSCM04	PRSCM03	PRSCM02	PRSCM01	PRSCM00

Cautions 1. Do not rewrite the PRSCM0 register during watch timer operation.

2. Set the PRSCM0 register before setting the PRSM0.BGCE0 bit to 1.

3. Set the PRSM0 and PRSCM0 registers according to the main clock frequency that is used so as to obtain an f_{BRG} frequency of 32.768 kHz.

(3) Watch timer operation mode register (WTM)

The WTM register enables or disables the count clock and operation of the watch timer, sets the interval time of the prescaler, controls the operation of the 5-bit counter, and sets the set time of the watch flag.

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

(1/2)

After reset: 00H R/W Address: FFFFF680H

	7	6	5	4	3	2	<1>	<0>
WTM	WTM7	WTM6	WTM5	WTM4	WTM3	WTM2	WTM1	WTM0

WTM7	WTM6	WTM5	WTM4	Selection of interval time of prescaler
0	0	0	0	$2^4/f_W$ (488 μs: $f_W = f_{XT}$)
0	0	0	1	$2^5/f_W$ (977 μs: $f_W = f_{XT}$)
0	0	1	0	$2^6/f_W$ (1.95 ms: $f_W = f_{XT}$)
0	0	1	1	$2^7/f_W$ (3.91 ms: $f_W = f_{XT}$)
0	1	0	0	$2^8/f_W$ (7.81 ms: $f_W = f_{XT}$)
0	1	0	1	$2^9/f_W$ (15.6 ms: $f_W = f_{XT}$)
0	1	1	0	$2^{10}/f_W$ (31.3 ms: $f_W = f_{XT}$)
0	1	1	1	$2^{11}/f_W$ (62.5 ms: $f_W = f_{XT}$)
1	0	0	0	$2^4/f_W$ (488 μs: $f_W = f_{BRG}$)
1	0	0	1	$2^5/f_W$ (977 μs: $f_W = f_{BRG}$)
1	0	1	0	$2^6/f_W$ (1.95 ms: $f_W = f_{BRG}$)
1	0	1	1	$2^7/f_W$ (3.91 ms: $f_W = f_{BRG}$)
1	1	0	0	$2^8/f_W$ (7.81 ms: $f_W = f_{BRG}$)
1	1	0	1	$2^9/f_W$ (15.6 ms: $f_W = f_{BRG}$)
1	1	1	0	$2^{10}/f_W$ (31.3 ms: $f_W = f_{BRG}$)
1	1	1	1	$2^{11}/f_W$ (62.5 ms: $f_W = f_{BRG}$)

(2/2)

WTM7	WTM3	WTM2	Selection of set time of watch flag
0	0	0	$2^{14}/f_W$ (0.5 s: $f_W = f_{XT}$)
0	0	1	$2^{13}/f_W$ (0.25 s: $f_W = f_{XT}$)
0	1	0	$2^{5}/f_W$ (977 μs: $f_W = f_{XT}$)
0	1	1	$2^{4}/f_W$ (488 μs: $f_W = f_{XT}$)
1	0	0	$2^{14}/f_W$ (0.5 s: $f_W = f_{BRG}$)
1	0	1	$2^{13}/f_W$ (0.25 s: $f_W = f_{BRG}$)
1	1	0	$2^{5}/f_W$ (977 μs: $f_W = f_{BRG}$)
1	1	1	$2^{4}/f_W$ (488 μs: $f_W = f_{BRG}$)

WTM1	Control of 5-bit counter operation
0	Clears after operation stops
1	Starts

WTM0	Watch timer operation enable
0	Stops operation (clears both prescaler and 5-bit counter)
1	Enables operation

Caution Rewrite the WTM2 to WTM7 bits while both the WTM0 and WTM1 bits are 0.

Remarks 1. f_W: Watch timer clock frequency

2. Values in parentheses apply to operation with f_W = 32.768 kHz.

10.4 Operation

★ 10.4.1 Operation as watch timer

The watch timer generates an interrupt request signal (INTWT) at fixed time intervals. The watch timer operates using time intervals of 0.25 or 0.5 seconds with the subclock (32.768 kHz) or main clock.

The count operation starts when the WTM.WTM1 and WTM.WTM0 bits are set to 11. When the WTM0 bit is cleared to 0, the 11-bit prescaler and 5-bit counter are cleared and the count operation stops.

The time of the watch timer can be adjusted by clearing the WTM1 bit to 0 and then the 5-bit counter when operating at the same time as the interval timer. At this time, an error of up to 15.6 ms may occur for the watch timer, but the interval timer is not affected.

If the main clock is used as the count clock of the watch timer, select the count clock using the PRSM0.BGCS01 and BGCS00 bits, the 8-bit comparison value using the PRSCM0 register, and set the count clock frequency (f_{BRG}) of the watch timer to 32.768 kHz.

When the PRSM0.BGCE0 bit is set to 1, f_{BRG} is supplied to the watch timer.

f_{BRG} can be calculated by the following expression.

$$f_{BRG} = f_X/(2^{m+1} \times N)$$

To set f_{BRG} to 32.768 kHz, perform the following calculation and set the BGCS01 and BGCS00 bits, and the PRSCM0 register.

<1> Set N = f_X/65,536 (round up the first decimal place). Set m = 0.
<2> Set N = N/2 and m = m + 1 where N is even.
<3> Repeat <2> until N is odd or m = 3.
<4> Set N to the PRSCM0 register and m to the BGCS01 and BGCS00 bits.

Example: When f_X = 4.00 MHz
<1> N = 4,000,000/65,536 = 61 (round up the first decimal place), m = 0
<2>, <3> Because N is odd, N = 61, m = 0.
<4> Set value of PRSCM0 register: 3DH (61), set value of BGCS01 and BGCS00 bits: 00

Remark m: Division value (set value of BGCS01 and BGCS00 bits) = 0 to 3
N: Set value of PRSCM0 register = 1 to 256 (N = 256 where set value of PRSCM0 register = 00H)
f_X: Main clock oscillation frequency

10.4.2 Operation as interval timer

The watch timer can also be used as an interval timer that repeatedly generates an interrupt request signal (INTWTI) at intervals specified by a preset count value.

The interval time can be selected by the WTM4 to WTM7 bits of the WTM register.

Table 10-1. Interval Time of Interval Timer

WTM7	WTM6	WTM5	WTM4	Interval Time	
0	0	0	0	$2^4 \times 1/fw$	488 μs (operating at $f_W = f_{XT}$ = 32.768 kHz)
0	0	0	1	$2^5 \times 1/fw$	977 μs (operating at $f_W = f_{XT}$ = 32.768 kHz)
0	0	1	0	$2^6 \times 1/fw$	1.95 ms (operating at $f_W = f_{XT}$ = 32.768 kHz)
0	0	1	1	$2^7 \times 1/fw$	3.91 ms (operating at $f_W = f_{XT}$ = 32.768 kHz)
0	1	0	0	$2^8 \times 1/fw$	7.81 ms (operating at $f_W = f_{XT}$ = 32.768 kHz)
0	1	0	1	$2^9 \times 1/fw$	15.6 ms (operating at $f_W = f_{XT}$ = 32.768 kHz)
0	1	1	0	$2^{10} \times 1/fw$	31.3 ms (operating at $f_W = f_{XT}$ = 32.768 kHz)
0	1	1	1	$2^{11} \times 1/fw$	62.5 ms (operating at $f_W = f_{XT}$ = 32.768 kHz)
1	0	0	0	$2^4 \times 1/fw$	488 μs (operating at $f_W = f_{BRG}$ = 32.768 kHz)
1	0	0	1	$2^5 \times 1/fw$	977 μs (operating at $f_W = f_{BRG}$ = 32.768 kHz)
1	0	1	0	$2^6 \times 1/fw$	1.95 ms (operating at $f_W = f_{BRG}$ = 32.768 kHz)
1	0	1	1	$2^7 \times 1/fw$	3.91 ms (operating at $f_W = f_{BRG}$ = 32.768 kHz)
1	1	0	0	$2^8 \times 1/fw$	7.81 ms (operating at $f_W = f_{BRG}$ = 32.768 kHz)
1	1	0	1	$2^9 \times 1/fw$	15.6 ms (operating at $f_W = f_{BRG}$ = 32.768 kHz)
1	1	1	0	$2^{10} \times 1/fw$	31.3 ms (operating at $f_W = f_{BRG}$ = 32.768 kHz)
1	1	1	1	$2^{11} \times 1/fw$	62.5 ms (operating at $f_W = f_{BRG}$ = 32.768 kHz)

Remark f_W: Watch timer clock frequency

Figure 10-2. Operation Timing of Watch Timer/Interval Timer

Remarks 1. When 0.5 seconds of the watch timer interrupt time is set.

2. fw: Watch timer clock frequency

Values in parentheses apply to operation with fw = 32.768 kHz.

n: Number of interval timer operations

10.4.3 Cautions

Some time is required before the first watch timer interrupt request signal (INTWT) is generated after operation is enabled (WTM.WTM1 and WTM.WTM0 bits = 1).

Figure 10-3. Example of Generation of Watch Timer Interrupt Request Signal (INTWT) (When Interrupt Cycle = 0.5 s)

It takes 0.515625 seconds (max.) for the first INTWT signal to be generated ($2^9 \times 1/32768 = 0.015625$ seconds longer (max.)). The INTWT signal is then generated every 0.5 seconds.

CHAPTER 11 FUNCTIONS OF WATCHDOG TIMER 2

11.1 Functions

Watchdog timer 2 has the following functions.

- Default-start watchdog timer[Note 1]
  → Reset mode: Reset operation upon overflow of watchdog timer 2 (generation of WDT2RES signal)
  → Non-maskable interrupt request mode: NMI operation upon overflow of watchdog timer 2 (generation of INTWDT2 signal)[Note 2]
- Input selectable from main clock and Ring-OSC clock as the source clock

Notes 1. Watchdog timer 2 automatically starts in the reset mode following reset release.
When watchdog timer 2 is not used, either stop its operation before reset is executed via this function, or clear watchdog timer 2 once and stop it within the next interval time.
Also, write to the WDTM2 register for verification purposes only once, even if the default settings (reset mode, interval time: $f_R/2^{19}$) do not need to be changed.

2. For the non-maskable interrupt servicing due to a non-maskable interrupt request signal (INTWDT2), refer to **17.9 Cautions**.

11.2 Configuration

The following shows the block diagram of watchdog timer 2.

Figure 11-1. Block Diagram of Watchdog Timer 2

Remark f_{XX}: Main clock frequency
f_R: Ring-OSC clock frequency
INTWDT2: Non-maskable interrupt request signal from watchdog timer 2
WDT2RES: Watchdog timer 2 reset signal

Watchdog timer 2 includes the following hardware.

Table 11-1. Configuration of Watchdog Timer 2

Item	Configuration
Control registers	Watchdog timer mode register 2 (WDTM2) Watchdog timer enable register (WDTE)

11.3 Control Registers

(1) Watchdog timer mode register 2 (WDTM2)

The WDTM2 register sets the overflow time and operation clock of watchdog timer 2.

This register can be read or written in 8-bit units. This register can be read any number of times, but it can be written only once following reset release.

Reset input sets this register to 67H.

★ **Caution When the subclock is operating and the main clock is stopped, accessing the WDTM2 register is disabled. For details, refer to 3.4.8 (2).**

After reset: 67H R/W Address: FFFFF6D0H

	7	6	5	4	3	2	1	0
WDTM2	0	WDM21	WDM20	WDCS24	WDCS23	WDCS22	WDCS21	WDCS20

WDM21	WDM20	Selection of operation mode of watchdog timer 2
0	0	Stops operation
0	1	Non-maskable interrupt request mode (generation of INTWDT2 signal)
1	–	Reset mode (generation of WDT2RES signal)

Cautions 1. For details of the WDCS24 to WDCS20 bits, see Table 11-2 Watchdog Timer 2 Clock Selection.

2. Although watchdog timer 2 can be stopped just by stopping the operation of Ring-OSC, set the WDTM2 register to 1FH to securely stop the timer (to avoid selection of the main clock due to an erroneous write operation).

3. If the WDTM2 register is rewritten twice or more after reset, an overflow signal is forcibly generated.

4. To stop the operation of watchdog timer 2, set the RCM.RSTP bit to 1 (to stop Ring-OSC) and write 1FH to the WDTM2 register.

Table 11-2. Watchdog Timer 2 Clock Selection

WDCS24	WDCS23	WDCS22	WDCS21	WDCS20	Selected Clock	100 kHz (MIN.)	200 kHz (TYP.)	400 kHz (MAX.)
0	0	0	0	0	$2^{12}/f_R$	41.0 ms	20.5 ms	10.2 ms
0	0	0	0	1	$2^{13}/f_R$	81.9 ms	41.0 ms	20.5 ms
0	0	0	1	0	$2^{14}/f_R$	163.8 ms	81.9 ms	41.0 ms
0	0	0	1	1	$2^{15}/f_R$	327.7 ms	163.8 ms	81.9 ms
0	0	1	0	0	$2^{16}/f_R$	655.4 ms	327.7 ms	163.8 ms
0	0	1	0	1	$2^{17}/f_R$	1,310.7 ms	655.4 ms	327.7 ms
0	0	1	1	0	$2^{18}/f_R$	2,621.4 ms	1,310.7 ms	655.4 ms
0	0	1	1	1	$2^{19}/f_R$	5,242.9 ms	2,621.47 ms	1,310.7 ms
						f_X = 5 MHz	f_X = 4 MHz	f_X = 2.5 MHz
0	1	0	0	0	$2^{16}/f_X$	13.1 ms	16.4 ms	26.2 ms
0	1	0	0	1	$2^{17}/f_X$	26.2 ms	32.8 ms	52.4 ms
0	1	0	1	0	$2^{18}/f_X$	52.4 ms	65.5 ms	104.9 ms
0	1	0	1	1	$2^{19}/f_X$	104.9 ms	131.1 ms	209.7 ms
0	1	1	0	0	$2^{20}/f_X$	209.7 ms	262.1 ms	419.4 ms
0	1	1	0	1	$2^{21}/f_X$	419.4 ms	524.3 ms	838.9 ms
0	1	1	1	0	$2^{22}/f_X$	838.9 ms	1,048.6 ms	1,677.7 ms
0	1	1	1	1	$2^{23}/f_X$	1,677.7 ms	2,097.2 ms	3,355.4 ms
1	×	×	×	×	–	Operation stops		

(2) Watchdog timer enable register (WDTE)

The counter of watchdog timer 2 is cleared and counting is restarted by writing “ACH” to the WDTE register.

This register can be read or written in 8-bit units.

Reset input sets this register to 9AH.

After reset: 9AH R/W Address: FFFFF6D1H

	7	6	5	4	3	2	1	0
WDTE								

Cautions 1. When a value other than “ACH” is written to the WDTE register, an overflow signal is forcibly output.

2. When a 1-bit memory manipulation instruction is executed for the WDTE register, an overflow signal is forcibly output.

3. The read value of the WDTE register is “9AH” (which differs from written value “ACH”).

★ 11.4 Operation

Watchdog timer 2 automatically starts in the reset mode following reset release.

The WDTM2 register can be written to only once following reset using byte access. To use watchdog timer 2, write the operation mode and the interval time to the WDTM2 register using an 8-bit memory manipulation instruction. After this, the operation of watchdog timer 2 cannot be stopped.

The WDTM2.WDCS24 to WDTM2.WDCS20 bits are used to select the watchdog timer 2 loop detection time interval.

Writing ACH to the WDTE register clears the counter of watchdog timer 2 and starts the count operation again. After the count operation has started, write ACH to the WDTE register within the loop detection time interval.

If the time interval expires without ACH being written to the WDTE register, a reset signal (WDT2RES) or a non-maskable interrupt request signal (INTWDT2) is generated, depending on the set values of the WDTM2.WDM21 and WDTM2.WDM20 bits.

To not use watchdog timer 2, write 1FH to the WDTM2 register.

For the non-maskable interrupt servicing while the non-maskable interrupt request mode is set, refer to **17.9 Cautions**.

★

CHAPTER 12 A/D CONVERTER

12.1 Overview

The A/D converter converts analog input signals into digital values, has a resolution of 10 bits, and can handle 12 analog input signal channels (ANI0 to ANI11).

The A/D converter has the following features.

- ○ 10-bit resolution
- ○ 12 channels
- ○ Successive approximation method
- ○ Operating voltage: AV_{REF0} = 4.0 to 5.5 V
- ○ Analog input voltage: 0 V to AV_{REF0}
- ○ The following functions are provided as operation modes.
  - Continuous select mode
  - Continuous scan mode
- ○ The following functions are provided as trigger modes.
  - Software trigger mode
  - External trigger mode (external, 1)
  - Timer trigger mode
- ○ Power-fail monitor function (conversion result compare function)

12.2 Functions

(1) 10-bit resolution A/D conversion

An analog input channel is selected from ANI0 to ANI11, and an A/D conversion operation is repeated at a resolution of 10 bits. Each time A/D conversion has been completed, an interrupt request signal (INTAD) is generated.

(2) Power-fail detection function

This function is used to detect a drop in the battery voltage. The result of A/D conversion (the value of the ADA0CRnH register) is compared with the value of the ADA0PFT register, and the INTAD signal is generated only when a specified comparison condition is satisfied.

12.3 Configuration

The block diagram of the A/D converter is shown below.

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

The A/D converter includes the following hardware.

Table 12-1. Configuration of A/D Converter

Item	Configuration
Analog inputs	12 channels (ANI0 to ANI11 pins)
Registers	Successive approximation register (SAR) A/D conversion result registers 0 to 11 (ADA0CR0 to ADA0CR11) A/D conversion result registers 0H to 11H (ADCR0H to ADCR11H): Only higher 8 bits can be read
Control registers	A/D converter mode registers 0 to 2 (ADA0M0 to ADA0M2) A/D converter channel specification register 0 (ADA0S) Power-fail compare mode register (ADA0PFM) Power-fail compare threshold value register (ADA0PFT)

(1) Successive approximation register (SAR)

The SAR register compares the voltage value of the analog input signal with the voltage tap (compare voltage) value from the series resistor string, and holds the comparison result starting from the most significant bit (MSB).

When the comparison result has been held down to the least significant bit (LSB) (i.e., when A/D conversion is complete), the contents of the SAR register are transferred to the ADA0CRn register.

Remark n = 0 to 11

(2) A/D conversion result register n (ADA0CRn), A/D conversion result register nH (ADA0CRnH)

The ADA0CRn register is a 16-bit register that stores the A/D conversion result. ADA0CRn consist of 12 registers and the A/D conversion result is stored in the higher 10 bits of the ADA0CRn register corresponding to analog input (the lower 6 bits are fixed to 0).

(3) A/D converter mode register 0 (ADA0M0)

This register specifies the operation mode and controls the conversion operation by the A/D converter.

(4) A/D converter mode register 1 (ADA0M1)

This register sets the conversion time of the analog input signal to be converted.

(5) A/D converter mode register 2 (ADA0M2)

This register sets the hardware trigger mode.

(6) A/D converter channel specification register (ADA0S)

This register sets the input port that inputs the analog voltage to be converted.

(7) Power-fail compare mode register (ADA0PFM)

This register sets the power-fail monitor mode.

(8) Power-fail compare threshold value register (ADA0PFT)

The ADA0PFT register sets a threshold value that is compared with the value of A/D conversion result register nH (ADA0CRnH). The 8-bit data set to the ADA0PFT register is compared with the higher 8 bits of the A/D conversion result register (ADA0CRnH).

(9) Controller

The controller compares the result of the A/D conversion (the value of the ADA0CRnH register) with the value of the ADA0PFT register when A/D conversion is completed or when the power-fail detection function is used, and generates the INTAD signal only when a specified comparison condition is satisfied.

(10) Sample & hold circuit

The sample & hold circuit samples each of the analog input signals selected by the input circuit and sends the sampled data to the voltage comparator. This circuit also holds the sampled analog input signal voltage during A/D conversion.

(11) Voltage comparator

The voltage comparator compares a voltage value that has been sampled and held with the output voltage value of the series resistor string.

(12) Series resistor string

This series resistor string is connected between AV_{REF0} and AV_{SS} and generates a voltage for comparison with the analog input signal.

(13) ANI0 to ANI11 pins

These are analog input pins for the 12 A/D converter channels and are used to input analog signals to be converted into digital signals. Pins other than the one selected as the analog input by the ADA0S register can be used as input port pins.

Caution Make sure that the voltages input to the ANI0 to ANI11 pins do not exceed the rated values. In particular if a voltage of AV_{REF0} or higher is input to a channel, the conversion value of that channel becomes undefined, and the conversion values of the other channels may also be affected.

(14) AV_{REF0} pin

This is the pin used to input the reference voltage of the A/D converter. Always make the potential at this pin the same as that at the V_{DD} pin even when the A/D converter is not used. The signals input to the ANI0 to ANI11 pins are converted to digital signals based on the voltage applied between the AV_{REF0} and AV_{SS} pins.

(15) AV_{SS} pin

This is the ground pin of the A/D converter. Always make the potential at this pin the same as that at the V_{SS} pin even when the A/D converter is not used.

12.4 Registers

The A/D converter is controlled by the following registers.

- A/D converter mode registers 0, 1, 2 (ADA0M0, ADA0M1, ADA0M2)
- A/D converter channel specification register 0 (ADA0S)
- Power-fail compare mode register (ADA0PFM)

The following registers are also used.

- A/D conversion result register n (ADA0CRn)
- A/D conversion result register nH (ADA0CRnH)
- Power-fail compare threshold value register (ADA0PFT)

(1) A/D converter mode register 0 (ADA0M0)

The ADA0M0 register is an 8-bit register that specifies the operation mode and controls conversion operations. This register can be read or written in 8-bit or 1-bit units. However, ADA0EF bit is read-only.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF200H

	<7>	6	5	4	3	2	1	<0>
ADA0M0	ADA0CE	0	ADA0MD1	ADA0MD0	ADA0ETS1	ADA0ETS0	ADA0TMD	ADA0EF

ADA0CE	A/D conversion control
0	Stops conversion
1	Enables conversion

ADA0MD1	ADA0MD0	Specification of A/D converter operation mode
0	0	Continuous select mode
0	1	Continuous scan mode
1	0	Setting prohibited
1	1	Setting prohibited

ADA0ETS1	ADA0ETS0	Specification of external trigger (ADTRG pin) input valid edge
0	0	No edge detection
0	1	Falling edge detection
1	0	Rising edge detection
1	1	Detection of both rising and falling edges

ADA0TMD	Trigger mode specification
0	Software trigger mode
1	External trigger mode/timer trigger mode

ADA0EF	A/D converter status display
0	A/D conversion stopped
1	A/D conversion in progress

Cautions 1. If bit 0 is written, this is ignored.

2. Changing the ADA0M1.ADA0FR2 to ADA0M1.ADA0FR0 bits during conversion (ADA0CE0 bit = 1) is prohibited.

3. When not using the A/D converter, stop the operation by setting the ADA0CE bit to 0 to reduce the power consumption.

4. When the subclock is operating and the main clock is stopped, accessing the ADA0M0 register is disabled. For details, refer to 3.4.8 (2).

(2) A/D converter mode register 1 (ADA0M1)

The ADA0M1 register is an 8-bit register used to control the conversion time.

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF201H

	7	6	5	4	3	2	1	0
ADA0M1	ADA0HS1	ADA0HS0	0	0	ADA0FR3	ADA0FR2	ADA0FR1	ADA0FR0

Remark For A/D conversion time setting examples, refer to **Table 12-2**.

Table 12-2. High-Speed Conversion Mode Setting Examples

ADA0HS		ADA0FR3	ADA0FR2	ADA0FR1	ADA0FR0	A/D Conversion Time[Note 1]	f_{XX} = 20 MHz	f_{XX} = 16 MHz	f_{XX} = 4 MHz	A/D Stabilization Time[Note 2]
1	0									
1	X	0	0	0	0	31/f_{XX}	Setting prohibited	Setting prohibited	7.75 μs	16/f_{XX}
		0	0	0	1	62/f_{XX}	3.10 μs	3.88 μs	15.50 μs	31/f_{XX}
		0	0	1	0	93/f_{XX}	4.65 μs	5.81 μs	Setting prohibited	47/f_{XX}
		0	0	1	1	124/f_{XX}	6.20 μs	7.75 μs	Setting prohibited	50/f_{XX}
		0	1	0	0	155/f_{XX}	7.75 μs	9.69 μs	Setting prohibited	50/f_{XX}
		0	1	0	1	186/f_{XX}	9.30 μs	11.63 μs	Setting prohibited	50/f_{XX}
		0	1	1	0	217/f_{XX}	10.85 μs	13.56 μs	Setting prohibited	50/f_{XX}
		0	1	1	1	248/f_{XX}	12.40 μs	15.50 μs	Setting prohibited	50/f_{XX}
		1	0	0	0	279/f_{XX}	13.95 μs	Setting prohibited	Setting prohibited	50/f_{XX}
		1	0	0	1	310/f_{XX}	15.50 μs	Setting prohibited	Setting prohibited	50/f_{XX}
		1	0	1	0	341/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited	50/f_{XX}
		1	0	1	1	372/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited	50/f_{XX}
		1	1	0	0	403/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited	50/f_{XX}
		1	1	0	1	434/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited	50/f_{XX}
		1	1	1	0	465/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited	50/f_{XX}
		1	1	1	1	496/f_{XX}	Setting prohibited	Setting prohibited	Setting prohibited	50/f_{XX}

Notes 1. Set the conversion time so that 2.91 μs ≤ conversion time ≤ 16 μs.
A/D conversion is not performed if f_{XX} is lower than 1.78 MHz.

2. When the ADA0M0.ADA0CE bit is changed from 0 to 1 to secure the A/D converter stabilization time (1 μs), prior to the first conversion only, A/D conversion starts after one of the above clock values is input.

Remark X: Don't care

(3) A/D converter mode register 2 (ADA0M2)

The ADA0M2 register specifies the hardware trigger mode.

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF203H

	7	6	5	4	3	2	1	0
ADA0M2	0	0	0	0	0	0	ADA0TMD1	ADA0TMD0

ADA0TMD1	ADA0TMD0	Specification of hardware trigger mode
0	0	External trigger mode (when ADTRG pin valid edge detected)
0	1	Timer trigger mode 0 (when INTTP2CC0 interrupt request generated)
1	0	Timer trigger mode 1 (when INTTP2CC1 interrupt request generated)
1	1	Setting prohibited

Caution Be sure to clear bits 7 to 2 to 0.

(4) A/D converter channel specification register 0 (ADA0S)

The ADA0S register specifies the pin that inputs the analog voltage to be converted into a digital signal.

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF202H

	7	6	5	4	3	2	1	0
ADA0S	0	0	0	0	ADA0S3	ADA0S2	ADA0S1	ADA0S0

ADA0S3	ADA0S2	ADA0S1	ADA0S0	Select mode	Scan mode
0	0	0	0	ANI0	ANI0
0	0	0	1	ANI1	ANI0, ANI1
0	0	1	0	ANI2	ANI0 to ANI2
0	0	1	1	ANI3	ANI0 to ANI3
0	1	0	0	ANI4	ANI0 to ANI4
0	1	0	1	ANI5	ANI0 to ANI5
0	1	1	0	ANI6	ANI0 to ANI6
0	1	1	1	ANI7	ANI0 to ANI7
1	0	0	0	ANI8	ANI0 to ANI8
1	0	0	1	ANI9	ANI0 to ANI9
1	0	1	0	ANI10	ANI0 to ANI10
1	0	1	1	ANI11	ANI0 to ANI11
1	1	0	0	Setting prohibited	Setting prohibited
1	1	0	1	Setting prohibited	Setting prohibited
1	1	1	0	Setting prohibited	Setting prohibited
1	1	1	1	Setting prohibited	Setting prohibited

Caution Be sure to clear bits 7 to 4 to 0.

(5) A/D conversion result registers n, nH (ADA0CRn, ADA0CRnH)

The ADA0CRn and ADA0CRnH registers store the A/D conversion results.

These registers are read-only, in 16-bit or 8-bit units. However, specify the ADA0CRn register for 16-bit access and the ADA0CRnH register for 8-bit access. The 10 bits of the conversion result are read from the higher 10 bits of the ADA0CRn register, and 0 is read from the lower 6 bits. The higher 8 bits of the conversion result are read from the ADA0CRnH register.

After reset: Undefined R Address: ADA0CR0 FFFFF210H, ADA0CR1 FFFFF212H, ADA0CR2 FFFFF214H, ADA0CR3 FFFFF216H, ADA0CR4 FFFFF218H, ADA0CR5 FFFFF21AH, ADA0CR6 FFFFF21CH, ADA0CR7 FFFFF21EH, ADA0CR8 FFFFF220H, ADA0CR9 FFFFF222H, ADA0CR10 FFFFF224H, ADA0CR11 FFFFF226H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADA0CRn (n = 0 to 11)	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	0	0	0	0	0	0

After reset: Undefined R Address: ADA0CR0H FFFFF211H, ADA0CR1H FFFFF213H, ADA0CR2H FFFFF215H, ADA0CR3H FFFFF217H, ADA0CR4H FFFFF219H, ADA0CR5H FFFFF21BH, ADA0CR6H FFFFF21DH, ADA0CR7H FFFFF21FH, ADA0CR8H FFFFF221H, ADA0CR9H FFFFF223H, ADA0CR10H FFFFF225H, ADA0CR11H FFFFF227H

	7	6	5	4	3	2	1	0
ADA0CRnH (n = 0 to 11)	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2

Cautions 1. A write operation to the ADA0M0 and ADA0S registers may cause the contents of the ADA0CRn register to become undefined. After the conversion, read the conversion result before writing to the ADA0M0 and ADA0S registers. Correct conversion results may not be read if a sequence other than the above is used.

2. When the subclock is operating and the main clock is stopped, accessing the ADA0CRn and ADA0CRnH registers is disabled. For details, refer to 3.4.8 (2).

The relationship between the analog voltage input to the analog input pins (ANI0 to ANI11) and the A/D conversion result (ADA0CRn register) is as follows.

$$SAR = INT\left(\frac{V_{IN}}{AV_{REF0}} \times 1{,}024 + 0.5\right)$$

$$ADA0CR^{Note} = SAR \times 64$$

Or,

$$(SAR - 0.5) \times \frac{AV_{REF0}}{1{,}024} \leq V_{IN} < (SAR + 0.5) \times \frac{AV_{REF0}}{1{,}024}$$

INT( ): Function that returns the integer of the value in ( )
V_{IN}: Analog input voltage
AV_{REF0}: AV_{REF0} pin voltage
ADA0CR: Value of ADA0CRn register

Note The lower 6 bits of the ADA0CRn register are fixed to 0.

The following shows the relationship between the analog input voltage and the A/D conversion results.

Figure 12-2. Relationship Between Analog Input Voltage and A/D Conversion Results

(6) Power-fail compare mode register (ADA0PFM)

The ADA0PFM register is an 8-bit register that sets the power-fail compare mode.

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF204H

	7	6	5	4	3	2	1	0
ADA0PFM	ADA0PFE	ADA0PFC	0	0	0	0	0	0

ADA0PFE	Selection of power-fail compare enable/disable
0	Power-fail compare disabled
1	Power-fail compare enabled

ADA0PFC	Selection of power-fail compare mode
0	Generates an interrupt request signal (INTAD) when ADA0CRnH ≥ ADA0PFT
1	Generates an interrupt request signal (INTAD) when ADA0CRnH < ADA0PFT

Cautions 1. In the select mode, the 8-bit data set to the ADA0PFT register is compared with the value of the ADA0CRnH register specified by the ADA0S register. If the result matches the condition specified by the ADA0PFC bit, the conversion result is stored in the ADA0CRn register and the INTAD signal is generated. If it does not match, however, the interrupt signal is not generated.

2. In the scan mode, the 8-bit data set to the ADA0PFT register is compared with the contents of the ADA0CR0H register. If the result matches the condition specified by the ADA0PFC bit, the conversion result is stored in the ADA0CR0 register and the INTAD signal is generated. If it does not match, however, the INTAD signal is not generated. Regardless of the comparison result, the scan operation is continued and the conversion result is stored in the ADA0CRn register until the scan operation is completed. However, the INTAD signal is not generated after the scan operation has been completed.

(7) Power-fail compare threshold value register (ADA0PFT)

The ADA0PFT register sets the compare value in the power-fail compare mode.

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF205H

	7	6	5	4	3	2	1	0
ADA0PFT								

12.5 Operation

12.5.1 Basic operation

<1> Set the operation mode, trigger mode, and conversion time for executing A/D conversion by using the ADA0M0, ADA0M1, ADA0M2, and ADA0S registers. When the ADA0CE bit of the ADA0M0 register is set, conversion is started in the software trigger mode and the A/D converter waits for a trigger in the external or timer trigger mode.

<2> When A/D conversion is started, the voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<3> When the sample & hold circuit samples the input channel for a specific time, it enters the hold status, and holds the input analog voltage until A/D conversion is complete.

<4> Set bit 9 of the successive approximation register (SAR). The tap selector selects (1/2) AV_{REF0} as the voltage tap of the series resistor string.

<5> The voltage difference between the voltage of the series resistor string and the analog input voltage is compared by the voltage comparator. If the analog input voltage is higher than (1/2) AV_{REF0}, the MSB of the SAR register remains set. If it is lower than (1/2) AV_{REF0}, the MSB is reset.

<6> Next, bit 8 of the SAR register is automatically set and the next comparison is started. Depending on the value of bit 9, to which a result has been already set, the voltage tap of the series resistor string is selected as follows.
- Bit 9 = 1: (3/4) AV_{REF0}
- Bit 9 = 0: (1/4) AV_{REF0}

This voltage tap and the analog input voltage are compared and, depending on the result, bit 8 is manipulated as follows.
Analog input voltage ≥ Voltage tap: Bit 8 = 1
Analog input voltage ≤ Voltage tap: Bit 8 = 0

<7> This comparison is continued to bit 0 of the SAR register.

<8> When comparison of the 10 bits is complete, the valid digital result is stored in the SAR register, which is then transferred to and stored in the ADA0CRn register. At the same time, an A/D conversion end interrupt request signal (INTAD) is generated.

Figure 12-3. A/D Converter Basic Operation

12.5.2 Trigger mode

The timing of starting the conversion operation is specified by setting a trigger mode. The trigger mode includes a software trigger mode and hardware trigger modes. The hardware trigger modes include timer trigger modes 0 and 1, and external trigger mode. The ADA0M0.ADA0TMD bit is used to set the trigger mode. The hardware trigger modes are set by the ADA0M2.ADA0TMD1 and ADA0M2.ADA0TMD0 bits.

(1) Software trigger mode

When the ADA0M0.ADA0CE bit is set to 1, the signal of the analog input pins (ANI0 to ANI11 pins) specified by the ADA0S register is converted. When conversion is complete, the result is stored in the ADA0CRn register. At the same time, the A/D conversion end interrupt request signal (INTAD) is generated.

If the operation mode specified by the ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 bits is the continuous select/continuous scan mode, the next conversion is started, unless the ADA0CE bit is cleared to 0 after completion of the first conversion.

When conversion is started, the ADA0M0.ADA0EF bit is set to 1 (indicating that conversion is in progress).

If the ADA0M0, ADA0M2, ADA0S, ADA0PFM, or ADA0PFT register is written during conversion, the conversion is aborted and started again from the beginning.

(2) External trigger mode

In this mode, converting the signal of the analog input pins (ANI0 to ANI11 pins) specified by the ADA0S register is started when an external trigger is input (to the ADTRG pin). Which edge of the external trigger is to be detected (i.e., the rising edge, falling edge, or both rising and falling edges) can be specified by using the ADA0M0.ADA0ETS1 and ADA0M0.ADA0ETS0 bits. When the ADA0CE bit is set to 1, the A/D converter waits for the trigger, and starts conversion after the external trigger has been input.

When conversion is completed, the result of conversion is stored in the ADA0CRn register, regardless of whether the continuous select or continuous scan mode is set as the operation mode by the ADA0MD1 and ADA0MD0 bits. At the same time, the INTAD signal is generated, and the A/D converter waits for the trigger again.

When conversion is started, the ADA0EF bit is set to 1 (indicating that conversion is in progress). While the A/D converter is waiting for the trigger, however, the ADA0EF bit is cleared to 0 (indicating that conversion is stopped). If the valid trigger is input during the conversion operation, the conversion is aborted and started again from the beginning.

If the ADA0M0, ADA0M2, ADA0S, ADA0PFM, or ADA0PFT register is written during conversion, the conversion is aborted, and the A/D converter waits for the trigger again.

(3) Timer trigger mode

In this mode, converting the signal of the analog input pins (ANI0 to ANI11 pins) specified by the ADA0S register is started by the compare match interrupt request signal (INTTP2CC0 or INTTP2CC1) of the capture/compare register connected to the timer. The INTTP2CC0 or INTTP2CC1 signal is selected by the ADA0TMD1 and ADA0TMD0 bits, and conversion is started at the rising edge of the specified compare match interrupt request signal. When the ADA0CE bit is set to 1, the A/D converter waits for a trigger, and starts conversion when the compare match interrupt request signal of the timer is input.

When conversion is completed, regardless of whether the continuous select or continuous scan mode is set as the operation mode by the ADA0MD1 and ADA0MD0 bits, the result of the conversion is stored in the ADA0CRn register. At the same time, the INTAD signal is generated, and the A/D converter waits for the trigger again.

When conversion is started, the ADA0EF bit is set to 1 (indicating that conversion is in progress). While the A/D converter is waiting for the trigger, however, the ADA0EF bit is cleared to 0 (indicating that conversion is stopped). If the valid trigger is input during the conversion operation, the conversion is aborted and started again from the beginning.

If the ADA0M0, ADA0M2, ADA0S, ADA0PFM, or ADA0PFT register is written during conversion, the conversion is aborted, and the A/D converter waits for the trigger again.

12.5.3 Operation mode

Two operation modes are available as the modes in which to set the ANI0 to ANI11 pins: continuous select mode and continuous scan mode.

The operation mode is selected by the ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 bits.

(1) Continuous select mode

In this mode, the voltage of one analog input pin selected by the ADA0S register is continuously converted into a digital value.

The conversion result is stored in the ADA0CRn register corresponding to the analog input pin. In this mode, an analog input pin corresponds to an ADA0CRn register on a one-to-one basis. Each time A/D conversion is completed, the A/D conversion end interrupt request signal (INTAD) is generated. After completion of conversion, the next conversion is started, unless the ADA0M0.ADA0CE bit is cleared to 0 (n = 0 to 11).

Figure 12-4. Timing Example of Continuous Select Mode Operation (ADA0S Register = 01H)

(2) Continuous scan mode

In this mode, analog input pins are sequentially selected, from the ANI0 pin to the pin specified by the ADA0S register, and their values are converted into digital values.

The result of each conversion is stored in the ADA0CRn register corresponding to the analog input pin. When conversion of the analog input pin specified by the ADA0S register is complete, the INTAD signal is generated, and A/D conversion is started again from the ANI0 pin, unless the ADA0CE bit is cleared to 0 (n = 0 to 11).

Figure 12-5. Timing Example of Continuous Scan Mode Operation (ADA0S Register = 03H)

(a) Timing example

(b) Block diagram

12.5.4 Power-fail compare mode

The A/D conversion end interrupt request signal (INTAD) can be controlled as follows by the ADA0PFM and ADA0PFT registers.

- When the ADA0PFM.ADA0PFE bit = 0, the INTAD signal is generated each time conversion is completed (normal use of the A/D converter).
- When the ADA0PFE bit = 1 and when the ADA0PFM.ADA0PFC bit = 0, the value of the ADA0CRnH register is compared with the value of the ADA0PFT register when conversion is completed, and the INTAD signal is generated only if ADA0CR0H ≥ ADA0PFT.
- When the ADA0PFE bit = 1 and when the ADA0PFC bit = 1, the value of the ADA0CRnH register is compared with the value of the ADA0PFT register when conversion is completed, and the INTAD signal is generated only if ADA0CR0H < ADA0PFT.

Remark n = 0 to 11

In the power-fail compare mode, two modes are available as modes in which to set the ANI0 to ANI11 pins: continuous select mode and continuous scan mode.

(1) Continuous select mode

In this mode, the result of converting the voltage of one analog input pin specified by the ADA0S register is compared with the set value of the ADA0PFT register. If the result of power-fail comparison matches the condition set by the ADA0PFC bit, the conversion result is stored in the ADA0CRn register, and the INTAD signal is generated. If it does not match, the conversion result is stored in the ADA0CRn register, and the INTAD signal is not generated. After completion of the first conversion, the next conversion is started, unless the ADA0M0.ADA0CE bit is cleared to 0 (n = 0 to 11).

Figure 12-6. Timing Example of Continuous Select Mode Operation (When Power-Fail Comparison Is Made: ADA0S Register = 01H)

(2) Continuous scan mode

In this mode, the results of converting the voltages of the analog input pins sequentially selected from the ANI0 pin to the pin specified by the ADA0S register are stored, and the set value of the ADA0CR0H register of channel 0 is compared with the value of the ADA0PFT register. If the result of power-fail comparison matches the condition set by the ADA0PFC bit, the conversion result is stored in the ADA0CR0 register, and the INTAD signal is generated. If it does not match, the conversion result is stored in the ADA0CR0 register, and the INTAD signal is not generated.

After the result of the first conversion has been stored in the ADA0CR0 register, the results of sequentially converting the voltages on the analog input pins up to the pin specified by the ADA0S register are continuously stored. After completion of conversion, the next conversion is started from the ANI0 pin again, unless the ADA0CE bit is cleared to 0.

Figure 12-7. Timing Example of Continuous Scan Mode Operation (When Power-Fail Comparison Is Made: ADA0S Register = 03H)

(a) Timing example

(b) Block diagram

12.6 Cautions

(1) When A/D converter is not used

When the A/D converter is not used, the power consumption can be reduced by clearing the ADA0M0.ADA0CE bit to 0.

(2) Input range of ANI0 to ANI11 pins

Input the voltage within the specified range to the ANI0 to ANI11 pins. If a voltage equal to or higher than AV_{REF0} or equal to or lower than AV_{SS} (even within the range of the absolute maximum ratings) is input to any of these pins, the conversion value of that channel is undefined, and the conversion value of the other channels may also be affected.

(3) Countermeasures against noise

To maintain the 10-bit resolution, the ANI0 to ANI11 pins must be effectively protected from noise. The influence of noise increases as the output impedance of the analog input source becomes higher. To lower the noise, connecting an external capacitor as shown in Figure 12-8 is recommended.

Figure 12-8. Processing of Analog Input Pin

(4) Alternate I/O

The analog input pins (ANI0 to ANI11) function alternately as port pins. When selecting one of the ANI0 to ANI11 pins to execute A/D conversion, do not execute an instruction to read an input port or write to an output port during conversion as the conversion resolution may drop.

If a digital pulse is applied to a pin adjacent to the pin whose input signal is being converted, the A/D conversion value may not be as expected due to the influence of coupling noise. Therefore, do not apply a pulse to a pin adjacent to the pin undergoing A/D conversion.

(5) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the contents of the ADA0S register are changed. If the analog input pin is changed during A/D conversion, therefore, the result of converting the previously selected analog input signal may be stored and the conversion end interrupt request flag may be set immediately before the ADA0S register is rewritten. If the ADIF flag is read immediately after the ADA0S register is rewritten, the ADIF flag may be set even though the A/D conversion of the newly selected analog input pin has not been completed. When A/D conversion is stopped, clear the ADIF flag before resuming conversion.

Figure 12-9. Generation Timing of A/D Conversion End Interrupt Request

Remark n = 0 to 11
m = 0 to 11

(6) AV_{REF0} pin

(a) The AV_{REF0} pin is used as the power supply pin of the A/D converter and also supplies power to the alternate-function ports. In an application where a backup power supply is used, be sure to supply the same voltage as V_{DD} to the AV_{REF0} pin as shown in Figure 12-10.

(b) The AV_{REF0} pin is also used as the reference voltage pin of the A/D converter. If the source supplying power to the AV_{REF0} pin has a high impedance or if the power supply has a low current supply capability, the reference voltage may fluctuate due to the current that flows during conversion (especially, immediately after the conversion operation enable bit ADA0CE has been set to 1). As a result, the conversion accuracy may drop. To avoid this, it is recommended to connect a capacitor across the AV_{REF0} and AV_{SS} pins as shown in Figure 12-10 to suppress the reference voltage fluctuation.

(c) If the source supplying power to the AV_{REF0} pin has a high DC resistance (for example, because of insertion of a diode), the voltage when conversion is enabled may be lower than the voltage when conversion is stopped, because of a voltage drop caused by the A/D conversion current.

Figure 12-10. AV_{REF0} Pin Processing Example

Note Parasitic inductance

(7) Reading ADA0CRn register

When the ADA0M0 to ADA0M2 or ADA0S register is written, the contents of the ADA0CRn register may be undefined. Read the conversion result after completion of conversion and before writing to the ADA0M0 to ADA0M2 and ADA0S registers. The correct conversion result may not be read at a timing different from the above.

(8) Standby mode

The A/D converter stops operation in the STOP mode. Consequently, its conversion result is invalid, and the power consumption can be reduced.

In the IDLE1, IDLE2, or subclock operation mode, however, the operation continues. To lower the power consumption, therefore, clear the ADA0M0.ADA0CE bit to 0.

12.7 How to Read A/D Converter Characteristics Table

This section describes the terms related to the A/D converter.

(1) Resolution

The minimum analog input voltage that can be recognized, i.e., the ratio of an analog input voltage to 1 bit of digital output is called 1 LSB (least significant bit). The ratio of 1 LSB to the full scale is expressed as %FSR (full-scale range). %FSR is the ratio of a range of convertible analog input voltages expressed as a percentage, and can be expressed as follows, independently of the resolution.

1%FSR = (Maximum value of convertible analog input voltage – Minimum value of convertible analog input voltage)/100

$= (AV_{REF0} - 0)/100$

$= AV_{REF0}/100$

When the resolution is 10 bits, 1 LSB is as follows:

1 LSB = $1/2^{10}$ = 1/1,024

= 0.098 %FSR

The accuracy is determined by the overall error, independently of the resolution.

(2) Overall error

This is the maximum value of the difference between an actually measured value and a theoretical value.
It is a total of zero-scale error, full-scale error, linearity error, and a combination of these errors.
The overall error in the characteristics table does not include the quantization error.

Figure 12-11. Overall Error

(3) Quantization error

This is an error of ±1/2 LSB that inevitably occurs when an analog value is converted into a digital value. Because the A/D converter converts analog input voltages in a range of ±1/2 LSB into the same digital codes, a quantization error is unavoidable.

This error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, or differential linearity error in the characteristics table.

Figure 12-12. Quantization Error

(4) Zero-scale error

This is the difference between the actually measured analog input voltage and its theoretical value when the digital output changes from 0…000 to 0…001 (1/2 LSB).

Figure 12-13. Zero-Scale Error

(5) Full-scale error

This is the difference between the actually measured analog input voltage and its theoretical value when the digital output changes from 1…110 to 1…111 (full scale – 3/2 LSB).

Figure 12-14. Full-Scale Error

(6) Differential linearity error

Ideally, the width to output a specific code is 1 LSB. This error indicates the difference between the actually measured value and its theoretical value when a specific code is output.

Figure 12-15. Differential Linearity Error

(7) Integral linearity error

This error indicates the extent to which the conversion characteristics differ from the ideal linear relationship. It indicates the maximum value of the difference between the actually measured value and its theoretical value where the zero-scale error and full-scale error are 0.

Figure 12-16. Integral Linearity Error

(8) Conversion time

This is the time required to obtain a digital output after each trigger has been output.

The conversion time in the characteristics table includes the sampling time.

(9) Sampling time

This is the time for which the analog switch is ON to load an analog voltage to the sample & hold circuit.

Figure 12-17. Sampling Time

CHAPTER 13 ASYNCHRONOUS SERIAL INTERFACE A (UARTA)

13.1 Features

- Transfer rate: 300 bps to 312.5 kbps (using internal system clock of 20 MHz and dedicated baud rate generator)
- Full-duplex communication: Internal UARTA receive data register n (UAnRX)
  Internal UARTA transmit data register n (UAnTX)
- 2-pin configuration: TXDAn: Transmit data output pin
  RXDAn: Receive data input pin
- Reception error detection function
  - Parity error
  - Framing error
  - Overrun error
- Interrupt sources: 2
  - Reception complete interrupt (INTUAnR): This interrupt occurs upon transfer of receive data from the shift register to receive buffer register n after serial transfer completion, in the reception enabled status.
  - Transmission enable interrupt (INTUAnT): This interrupt occurs upon transfer of transmit data from the transmit buffer register to the shift register in the transmission enabled status.
- Character length: 7, 8 bits
- Parity function: Odd, even, 0, none
- Transmission stop bit: 1, 2 bits
- On-chip dedicated baud rate generator
- MSB-/LSB-first transfer selectable
- Transmit/receive data inverted input/output possible
- 13 to 20 bits selectable for the SBF (Sync Break Field) in the LIN (Local Interconnect Network) communication format
  - Recognition of 11 bits or more possible for SBF reception in LIN communication format
  - SBF reception flag provided

Remark n = 0 to 2

13.2 Configuration

The block diagram of UARTAn is shown below.

Figure 13-1. Block Diagram of Asynchronous Serial Interface An

Note UARTA0 only

Remarks 1. n = 0 to 2

2. For the configuration of the baud rate generator, refer to **Figure 13-13**.

UARTAn includes the following hardware.

Table 13-1. Configuration of UARTAn

Item	Configuration
Registers	UARTAn control register 0 (UAnCTL0)
	UARTAn control register 1 (UAnCTL1)
	UARTAn control register 2 (UAnCTL2)
	UARTAn option control register 0 (UAnOPT0)
	UARTAn status register (UAnSTR)
	UARTAn receive shift register
	UARTAn receive data register (UAnRX)
	UARTAn transmit shift register
	UARTAn transmit data register (UAnTX)

(1) UARTAn control register 0 (UAnCTL0)

The UAnCTL0 register is an 8-bit register used to specify the UARTAn operation.

(2) UARTAn control register 1 (UAnCTL1)

The UAnCTL1 register is an 8-bit register used to select the input clock for UARTAn.

(3) UARTAn control register 2 (UAnCTL2)

The UAnCTL2 register is an 8-bit register used to control the baud rate for UARTAn.

(4) UARTAn option control register 0 (UAnOPT0)

The UAnOPT0 register is an 8-bit register used to control serial transfer for UARTAn.

(5) UARTAn status register (UAnSTR)

The UAnSTR register consists of flags indicating the error contents when a reception error occurs. Each one of the reception error flags is set (to 1) upon occurrence of a reception error and is reset (to 0) by reading the UAnSTR register.

(6) UARTAn receive shift register

This is a shift register used to convert the serial data input to the RXDAn pin into parallel data. Upon reception of 1 byte of data and detection of the stop bit, the receive data is transferred to the UAnRX register.

This register cannot be manipulated directly.

(7) UARTAn receive data register (UAnRX)

The UAnRX register is an 8-bit buffer register that holds receive data. When 7 characters are received, 0 is stored in the highest bit (when data is received LSB first).

In the reception enabled status, receive data is transferred from the UARTAn receive shift register to the UAnRX register in synchronization with the completion of shift-in processing of 1 frame.

Transfer to the UAnRX register also causes the reception complete interrupt request signal (INTUAnR) to be output.

(8) UARTAn transmit shift register

The transmit shift register is a shift register used to convert the parallel data transferred from the UAnTX register into serial data.

When 1 byte of data is transferred from the UAnTX register, the shift register data is output from the TXDAn pin.

This register cannot be manipulated directly.

(9) UARTAn transmit data register (UAnTX)

The UAnTX register is an 8-bit transmit data buffer. Transmission starts when transmit data is written to the UAnTX register. When data can be written to the UAnTX register (when data of one frame is transferred from the UAnTX register to the UARTAn transmit shift register), the transmission enable interrupt request signal (INTUAnT) is generated.

13.3 Control Registers

(1) UARTAn control register 0 (UAnCTL0)

The UAnCTL0 register is an 8-bit register that controls the UARTAn serial transfer operation.

This register can be read or written in 8-bit or 1-bit units.

Reset input sets this register to 10H.

(1/2)

After reset: 10H R/W Address: UA0CTL0 FFFFFA00H, UA1CTL0 FFFFFA10H, UA2CTL0 FFFFFA20H

	<7>	<6>	<5>	<4>	3	2	1	0
UAnCTL0 (n = 0 to 2)	UAnPWR	UAnTXE	UAnRXE	UAnDIR	UAnPS1	UAnPS0	UAnCL	UAnSL

UAnPWR	UARTAn operation control
0	Disable UARTAn operation (UARTAn reset asynchronously)
1	Enable UARTAn operation
The UARTAn operation is controlled by the UAnPWR bit. The TXDAn pin output is fixed to high level by clearing the UAnPWR bit to 0 (fixed to low level if UAnOPT0.UAnTDL bit = 1).	

UAnTXE	Transmission operation enable
0	Disable transmission operation
1	Enable transmission operation
• To start transmission, set the UAnPWR bit to 1 and then set the UAnTXE bit to 1. To stop, transmission clear the UAnTXE bit to 0 and then UAnPWR bit to 0. • To initialize the transmission unit, clear the UAnTXE bit to 0, wait for two cycles of the base clock, and then set the UAnTXE bit to 1 again. Otherwise, initialization may not be executed (for the base clock, refer to **13.6 (1) (a) Base clock**).	

★

UAnRXE	Reception operation enable
0	Disable reception operation
1	Enable reception operation
• To start reception, set the UAnPWR bit to 1 and then set the UAnRXE bit to 1. To stop reception, clear the UAnRXE bit to 0 and then UAnPWR bit to 0. • To initialize the reception unit, clear the UAnRXE bit to 0, wait for two periods of the base clock, and then set the UAnRXE bit to 1 again. Otherwise, initialization may not be executed (for the base clock, refer to **13.6 (1) (a) Base clock**).	

★

(2/2)

UAnDIR	Transfer direction selection
0	MSB-first transfer
1	LSB-first transfer
This register can be rewritten only when the UAnPWR bit = 0 or the UAnTXE bit = the UAnRXE bit = 0.	

UAnPS1	UAnPS0	Parity selection during transmission	Parity selection during reception
0	0	No parity output	Reception with no parity
0	1	0 parity output	Reception with 0 parity
1	0	Odd parity output	Odd parity check
1	1	Even parity output	Even parity check
• This register can be rewritten only when the UAnPWR bit = 0 or the UAnTXE bit = the UAnRXE bit = 0. • If “Reception with 0 parity” is selected during reception, a parity check is not performed. Therefore, since the UAnSTR.UAnPE bit is not set, no error interrupt is output. • When transmission and reception are performed in the LIN format, set the UAnPS1 and UAnPS0 bits to 00.			

UAnCL	Specification of data character length of 1 frame of transmit/receive data
0	7 bits
1	8 bits
This register can be rewritten only when the UAnPWR bit = 0 or the UAnTXE bit = the UAnRXE bit = 0.	

UAnSL	Specification of length of stop bit for transmit data
0	1 bit
1	2 bits
This register can be rewritten only when the UAnPWR bit = 0 or the UAnTXE bit = the UAnRXE bit = 0.	

Remark For details of parity, see **13.5.9 Parity types and operations**.

(2) UARTAn control register 1 (UAnCTL1)

For details, see **13.6 (2) UARTAn control register 1 (UAnCTL1)**.

(3) UARTAn control register 2 (UAnCTL2)

For details, see **13.6 (3) UARTAn control register 2 (UAnCTL2)**.

(4) UARTAn option control register 0 (UAnOPT0)

The UAnOPT0 register is an 8-bit register that controls the UARTAn serial transfer operation.

This register can be read or written in 8-bit or 1-bit units.

Reset input sets this register to 14H.

(1/2)

After reset: 14H R/W Address: UA0OPT0 FFFFFA03H, UA1OPT0 FFFFFA13H, UA2OPT0 FFFFFA23H

	<7>	6	5	4	3	2	1	0
UAnOPT0 (n = 0 to 2)	UAnSRF	UAnSRT	UAnSTT	UAnSLS2	UAnSLS1	UAnSLS0	UAnTDL	UAnRDL

UAnSRF	SBF reception flag
0	When the UAnCTL0.UAnPWR bit = UAnCTL0.UAnRXE bit = 0 are set. Also upon normal end of SBF reception.
1	During SBF reception

- SBF (Sync Brake Field) reception is judged during LIN communication.
- The UAnSRF bit is held at 1 when an SBF reception error occurs, and then SBF reception is started again.

UAnSRT	SBF reception trigger
0	–
1	SBF reception trigger

- This is the SBF reception trigger bit during LIN communication, and when read, “0” is always read. For SBF reception, set the UAnSRT bit (to 1) to enable SBF reception.
- Set the UAnSRT bit after setting the UAnPWR bit = UAnRXE bit = 1.

UAnSTT	SBF transmission trigger
0	–
1	SBF transmission trigger

- This is the SBF transmission trigger bit during LIN communication, and when read, “0” is always read.
- Set the UAnSTT bit after setting the UAnPWR bit = UAnTXE bit = 1.

(2/2)

UAnSLS2	UAnSLS1	UAnSLS0	SBF transmit length selection
1	0	1	13-bit output (reset value)
1	0	0	14-bit output
1	1	1	15-bit output
0	0	0	16-bit output
0	0	1	17-bit output
0	1	0	18-bit output
0	1	1	19-bit output
1	1	0	20-bit output
This register can be set when the UAnPWR bit = 0 or when the UAnTXE bit = 0.			

UAnTDL	Transmit data level bit
0	Normal output of transfer data
1	Inverted output of transfer data
• The output level of the TXDAn pin can be inverted using the UAnTDL bit. • This register can be set when the UAnPWR bit = 0 or when the UAnTXE bit = 0.	

UAnRDL	Receive data level bit
0	Normal input of transfer data
1	Inverted input of transfer data
• The output level of the RXDAn pin can be inverted using the UAnRDL bit. • This register can be set when the UAnPWR bit = 0 or the UAnRXE bit = 0.	

(5) UARTAn status register (UAnSTR)

The UAnSTR register is an 8-bit register that displays the UARTAn transfer status and reception error contents. This register can be read or written in 8-bit or 1-bit units, but the UAnTSF bit is a read-only bit, while the UAnPE, UAnFE, and UAnOVE bits can both be read and written. However, these bits can only be cleared by writing 0; they cannot be set by writing 1 (even if 1 is written to them, the value is retained).

The initialization conditions are shown below.

Register/Bit	Initialization Conditions
UAnSTR register	• Reset • UAnCTL0.UAnPWR bit = 0
UAnTSF bit	• UAnCTL0.UAnTXE bit = 0
UAnPE, UAnFE, UAnOVE bits	• 0 write • UAnCTL0.UAnRXE bit = 0

After reset: 00H R/W Address: UA0STR FFFFFA04H, UA1STR FFFFFA14H, UA2STR FFFFFA24H

	<7>	6	5	4	3	<2>	<1>	<0>
UAnSTR	UAnTSF	0	0	0	0	UAnPE	UAnFE	UAnOVE

(n = 0 to 2)

UAnTSF	Transfer status flag
0	• When the UAnPWR bit = 0 or the UAnTXE bit = 0 has been set • When, following transfer completion, there was no next data transfer from UAnTX register
1	Write to UAnTX register

The UAnTSF bit is always 1 when performing continuous transmission. When initializing the transmission unit, check that the UAnTSF bit = 0 before performing initialization. The transmit data is not guaranteed when initialization is performed while the UAnTSF bit = 1.

UAnPE	Parity error flag
0	• When the UAnPWR bit = 0 or the UAnRXE bit = 0 has been set • When 0 has been written
1	When parity of data and parity bit do not match during reception

- The operation of the UAnPE bit is controlled by the settings of the UAnCTL0.UAnPS1 and UAnCTL0.UAnPS0 bits.
- The UAnPE bit can be read and written, but it can only be cleared by writing 0 to it, and it cannot be set by writing 1 to it. When 1 is written to this bit, the value is retained.

UAnFE	Framing error flag
0	• When the UAnPWR bit = 0 or the UAnRXE bit = 0 has been set • When 0 has been written
1	When no stop bit is detected during reception

- Only the first bit of the receive data stop bits is checked, regardless of the value of the UAnCTL0.UAnSL bit.
- The UAnFE bit can be both read and written, but it can only be cleared by writing 0 to it, and it cannot be set by writing 1 to it. When 1 is written to this bit, the value is retained.

UAnOVE	Overrun error flag
0	• When the UAnPWR bit = 0 or the UAnRXE bit = 0 has been set • When 0 has been written
1	When receive data has been set to the UAnRX register and the next receive operation is completed before that receive data has been read

- When an overrun error occurs, the data is discarded without the next receive data being written to the receive buffer.
- The UAnOVE bit can be both read and written, but it can only be cleared by writing 0 to it, and it cannot be set by writing 1 to it. When 1 is written to this bit, the value is retained.

(6) UARTAn receive data register (UAnRX)

The UAnRX register is an 8-bit buffer register that stores parallel data converted by the receive shift register.

The data stored in the receive shift register is transferred to the UAnRX register upon completion of reception of 1 byte of data.

During LSB-first reception when the data length has been specified as 7 bits, the receive data is transferred to bits 6 to 0 of the UAnRX register and the MSB always becomes 0. During MSB-first reception, the receive data is transferred to bits 7 to 1 of the UAnRX register and the LSB always becomes 0.

When an overrun error (UAnOVE) occurs, the receive data at this time is not transferred to the UAnRX register and is discarded.

This register is read-only, in 8-bit units.

In addition to reset input, the UAnRX register can be set to FFH by clearing the UAnCTL0.UAnPWR bit to 0.

After reset: FFH R Address: UA0RX FFFFFA06H, UA1RX FFFFFA16H, UA2RX FFFFFA26H

	7	6	5	4	3	2	1	0
UAnRX (n = 0 to 2)								

(7) UARTAn transmit data register (UAnTX)

The UAnTX register is an 8-bit register used to set transmit data.

This register can be read or written in 8-bit units.

Reset input sets this register to FFH.

After reset: FFH R/W Address: UA0TX FFFFFA07H, UA1TX FFFFFA17H, UA2TX FFFFFA27H

	7	6	5	4	3	2	1	0
UAnTX (n = 0 to 2)								

13.4 Interrupt Request Signals

The following two interrupt request signals are generated from UARTAn.

- Reception complete interrupt request signal (INTUAnR)
- Transmission enable interrupt request signal (INTUAnT)

The default priority for these two interrupt request signals is reception complete interrupt request signal then transmission enable interrupt request signal.

Table 13-2. Interrupts and Their Default Priorities

Interrupt	Priority
Reception complete	High
Transmission enable	Low

(1) Reception complete interrupt request signal (INTUAnR)

A reception complete interrupt request signal is output when data is shifted into the receive shift register and transferred to the UAnRX register in the reception enabled status.

When a reception complete interrupt request signal is received and the data is read, read the UAnSTR register and check that the reception result is not an error.

No reception complete interrupt request signal is generated in the reception disabled status.

(2) Transmission enable interrupt request signal (INTUAnT)

If transmit data is transferred from the UAnTX register to the UARTAn transmit shift register with transmission enabled, the transmission enable interrupt request signal is generated.

13.5 Operation

13.5.1 Data format

Full-duplex serial data reception and transmission is performed.

As shown in Figure 13-2, one data frame of transmit/receive data consists of a start bit, character bits, parity bit, and stop bit(s).

Specification of the character bit length within 1 data frame, parity selection, specification of the stop bit length, and specification of MSB/LSB-first transfer are performed using the UAnCTL0 register.

Moreover, control of UART output/inverted output for the TXDAn pin is performed using the UAnOPT0.UAnTDL bit.

- Start bit ..... 1 bit
- Character bits..... 7 bits/8 bits
- Parity bit ..... Even parity/odd parity/0 parity/no parity
- Stop bit..... 1 bit/2 bits

Figure 13-2. UARTA Transmit/Receive Data Format

(a) 8-bit data length, LSB first, even parity, 1 stop bit, transfer data: 55H

(b) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H

(c) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H, TXDAn inversion

(d) 7-bit data length, LSB first, odd parity, 2 stop bits, transfer data: 36H

(e) 8-bit data length, LSB first, no parity, 1 stop bit, transfer data: 87H

13.5.2 SBF transmission/reception format

The μPD703229Y and 70F3229Y have an SBF (Sync Break Field) transmission/reception control function to enable use of the LIN function.

★ **Remark** LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol intended to aid the cost reduction of an automotive network.
LIN communication is single-master communication, and up to 15 slaves can be connected to one master.
The LIN slaves are used to control the switches, actuators, and sensors, and these are connected to the LIN master via the LIN network.
Normally, the LIN master is connected to a network such as CAN (Controller Area Network).
In addition, the LIN bus uses a single-wire method and is connected to the nodes via a transceiver that complies with ISO9141.
In the LIN protocol, the master transmits a frame with baud rate information and the slave receives it and corrects the baud rate error. Therefore, communication is possible when the baud rate error in the slave is ±15% or less.

Figures 13-3 and 13-4 outline the transmission and reception manipulations of LIN.

Figure 13-3. LIN Transmission Manipulation Outline

Notes 1. The interval between each field is controlled by software.

2. SBF output is performed by hardware. The output width is the bit length set by the UAnOPT0.UAnSLS2 to UAnOPT0.UAnSLS0 bits. If even finer output width adjustments are required, such adjustments can be performed using the UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits.

3. 80H transfer in the 8-bit mode is substituted for the wakeup signal frame.

4. A transmission enable interrupt request signal (INTUAnT) is output at the start of each transmission. The INTUAnT signal is also output at the start of each SBF transmission.

Figure 13-4. LIN Reception Manipulation Outline

Notes 1. The wakeup signal is sent by the pin edge detector, UARTAn is enabled, and the SBF reception mode is set.

2. The receive operation is performed until detection of the stop bit. Upon detection of SBF reception of 11 or more bits, normal SBF reception end is judged, and an interrupt signal is output. Upon detection of SBF reception of less than 11 bits, an SBF reception error is judged, no interrupt signal is output, and the mode returns to the SBF reception mode.

3. If SBF reception ends normally, an interrupt request signal is output. The timer is enabled by an SBF reception complete interrupt. Moreover, error detection for the UAnSTR.UAnOVE, UAnSTR.UAnPE, and UAnSTR.UAnFE bits is suppressed and UART communication error detection processing and UARTAn receive shift register and data transfer of the UAnRX register are not performed. The UARTAn receive shift register holds the initial value, FFH.

4. The RXDAn pin is connected to TI (capture input) of the timer, the transfer rate is calculated, and the baud rate error is calculated. The value of the UAnCTL2 register obtained by correcting the baud rate error after dropping UARTA enable is set again, causing the status to become the reception status.

5. Checksum field distinctions are made by software. UARTAn is initialized following CSF reception, and the processing for setting the SBF reception mode again is performed by software.

13.5.3 SBF transmission

When the UAnCTL0.UAnPWR bit = UAnCTL0.UAnTXE bit = 1, the transmission enabled status is entered, and SBF transmission is started by setting (to 1) the SBF transmission trigger (UAnOPT0.UAnSTT bit).

Thereafter, a low-level width of bits 13 to 20 specified by the UAnOPT0.UAnSLS2 to UAnOPT0.UAnSLS0 bits is output. A transmission enable interrupt request signal (INTUAnT) is generated upon SBF transmission start. Following the end of SBF transmission, the UAnSTT bit is automatically cleared. Thereafter, the UART transmission mode is restored.

Transmission is suspended until the data to be transmitted next is written to the UAnTX register, or until the SBF transmission trigger (UAnSTT bit) is set.

Figure 13-5. SBF Transmission

13.5.4 SBF reception

The reception enabled status is achieved by setting the UAnCTL0.UAnPWR bit to 1 and then setting the UAnCTL0.UAnRX bit to 1.

The SBF reception wait status is set by setting the SBF reception trigger (UAnOPT0.UAnSTR bit) to 1.

In the SBF reception wait status, similarly to the UART reception wait status, the RXDAn pin is monitored and start bit detection is performed.

Following detection of the start bit, reception is started and the internal counter counts up according to the set baud rate.

When a stop bit is received, if the SBF width is 11 or more bits, normal processing is judged and a reception complete interrupt request signal (INTUAnR) is output. The UAnOPT0.UAnSRF bit is automatically cleared and SBF reception ends. Error detection for the UAnSTR.UAnOVE, UAnSTR.UAnPE, and UAnSTR.UAnFE bits is suppressed and UART communication error detection processing is not performed. Moreover, data transfer of the UARTAn receive shift register and UAnRX register is not performed and FFH, the initial value, is held. If the SBF width is 10 or fewer bits, reception is terminated as error processing without outputting an interrupt, and the SBF reception mode is returned to. The UAnSRF bit is not cleared at this time.

Figure 13-6. SBF Reception

13.5.5 UART transmission

A high level is output to the TXDAn pin by setting the UAnCTL0.UAnPWR bit to 1.

Next, the transmission enabled status is set by setting the UAnCTL0.UAnTXE bit to 1, and transmission is started by writing transmit data to the UAnTX register. The start bit, parity bit, and stop bit are automatically added.

Since the CTS (transmit enable signal) input pin is not provided in UARTAn, use a port to check that reception is enabled at the transmit destination.

The data in the UAnTX register is transferred to the UARTAn transmit shift register upon the start of the transmit operation.

A transmission enable interrupt request signal (INTUAnT) is generated upon completion of transmission of the data of the UAnTX register to the UARTAn transmit shift register, and thereafter the contents of the UARTAn transmit shift register are output to the TXDAn pin.

Write of the next transmit data to the UAnTX register is enabled by generating the INTUAnT signal.

Figure 13-7. UART Transmission

Remark LSB first

13.5.6 Continuous transmission procedure

UARTAn can write the next transmit data to the UAnTX register when the UARTAn transmit shift register starts the shift operation. The transmit timing of the UARTAn transmit shift register can be judged from the transmission enable interrupt request signal (INTUAnT). An efficient communication rate is realized by writing the data to be transmitted next to the UAnTX register during transfer.

Caution During continuous transmission execution, perform initialization after checking that the UAnSTR.UAnTSF bit is 0. The transmit data cannot be guaranteed when initialization is performed while the UAnTSF bit is 1.

Figure 13-8. Continuous Transmission Processing Flow

Figure 13-9. Continuous Transmission Operation Timing

13.5.7 UART reception

The reception wait status is set by setting the UAnCTL0.UAnPWR bit to 1 and then setting the UAnCTL0.UAnRXE bit to 1. In the reception wait status, the RXDAn pin is monitored and start bit detection is performed.

Start bit detection is performed using a two-step detection routine.

First the falling edge of the RXDAn pin is detected and sampling is started at the falling edge. The start bit is recognized if the RXDAn pin is low level at the start bit sampling point. After a start bit has been recognized, the receive operation starts, and serial data is saved to the UARTAn receive shift register according to the set baud rate.

When the reception complete interrupt request signal (INTUAnR) is output upon reception of the stop bit, the data of the UARTAn receive shift register is written to the UAnRX register. However, if an overrun error (UAnSTR.UAnOVE bit) occurs, the receive data at this time is not written to the UAnRX register and is discarded.

Even if a parity error (UAnSTR.UAnPE bit) or a framing error (UAnSTR.UAnFE bit) occurs during reception, reception continues until the reception position of the first stop bit, and INTUAnR signal is output following reception completion.

Figure 13-10. UART Reception

Cautions 1. Be sure to read the UAnRX register even when a reception error occurs. If the UAnRX register is not read, an overrun error occurs during reception of the next data, and reception errors continue occurring indefinitely.

2. The operation during reception is performed assuming that there is only one stop bit. A second stop bit is ignored.

3. When reception is completed, read the UAnRX register after the reception complete interrupt request signal (INTUAnR) has been generated, and clear the UAnPWR or UAnRXE bit to 0. If the UAnPWR or UAnRXE bit is cleared to 0 before the INTUAnR signal is generated, the read value of the UAnRX register cannot be guaranteed.

4. If receive completion processing (INTUAnR signal generation) of UARTAn and the UAnPWR bit = 0 or UAnRXE bit = 0 conflict, the INTUAnR signal may be generated in spite of these being no data stored in the UAnRX register.

To complete reception without waiting INTUAnR signal generation, be sure to clear (0) the interrupt request flag (UAnRIF) of the UAnRIC register, after setting (1) the interrupt mask flag (UAnRMK) of the interrupt control register (UAnRIC) and then set the UAnPWR bit = 0 or UAnRXE bit = 0.

13.5.8 Reception errors

Errors during a receive operation are of three types: parity errors, framing errors, and overrun errors. Data reception result error flags are set in the UAnSTR register and a reception complete interrupt request signal (INTUAnR) is output when an error occurs.

It is possible to ascertain which error occurred during reception by reading the contents of the UAnSTR register.

Clear the reception error flag by writing 0 to it after reading it.

- Reception error causes

Error Flag	Reception Error	Cause
UAnPE	Parity error	Received parity bit does not match the setting
UAnFE	Framing error	Stop bit not detected
UAnOVE	Overrun error	Reception of next data completed before data was read from receive buffer

13.5.9 Parity types and operations

Caution When using the LIN function, fix the UAnCTL0.UAnPS1 and UAnCTL0.UAnPS0 bits to 00.

The parity bit is used to detect bit errors in the communication data. Normally the same parity is used on the transmission side and the reception side.

In the case of even parity and odd parity, it is possible to detect odd-count bit errors. In the case of 0 parity and no parity, errors cannot be detected.

(a) Even parity

(i) During transmission

The number of bits whose value is “1” among the transmit data, including the parity bit, is controlled so as to be an even number. The parity bit values are as follows.

- Odd number of bits whose value is “1” among transmit data: 1
- Even number of bits whose value is “1” among transmit data: 0

(ii) During reception

The number of bits whose value is “1” among the receive data, including the parity bit, is counted, and if it is an odd number, a parity error is output.

(b) Odd parity

(i) During transmission

Opposite to even parity, the number of bits whose value is “1” among the transmit data, including the parity bit, is controlled so that it is an odd number. The parity bit values are as follows.

- Odd number of bits whose value is “1” among transmit data: 0
- Even number of bits whose value is “1” among transmit data: 1

(ii) During reception

The number of bits whose value is “1” among the receive data, including the parity bit, is counted, and if it is an even number, a parity error is output.

(c) 0 parity

During transmission, the parity bit is always made 0, regardless of the transmit data.

During reception, parity bit check is not performed. Therefore, no parity error occurs, regardless of whether the parity bit is 0 or 1.

(d) No parity

No parity bit is added to the transmit data.

Reception is performed assuming that there is no parity bit. No parity error occurs since there is no parity bit.

13.5.10 Receive data noise filter

This filter samples the RXDAn pin using the base clock of the prescaler output.

When the same sampling value is read twice, the match detector output changes and the RXDAn signal is sampled as the input data. Therefore, data not exceeding 2 clock width is judged to be noise and is not delivered to the internal circuit (see **Figure 13-12**). Refer to **13.6 (1) (a) Base clock** regarding the base clock.

Moreover, since the circuit is as shown in Figure 13-11, the processing that goes on within the receive operation is delayed by 3 clocks in relation to the external signal status.

Figure 13-11. Noise Filter Circuit

★ **Figure 13-12. Timing of RXDAn Signal Judged as Noise**

13.6 Dedicated Baud Rate Generator

The dedicated baud rate generator consists of a source clock selector block and an 8-bit programmable counter, and generates a serial clock during transmission and reception with UARTAn. Regarding the serial clock, a dedicated baud rate generator output can be selected for each channel.

There is an 8-bit counter for transmission and another one for reception.

(1) Baud rate generator configuration

Figure 13-13. Configuration of Baud Rate Generator

Note Only UARTA0 is valid; setting UARTA1 and UARTA2 is prohibited.

Remarks 1. n = 0 to 2

2. fXX: Main clock frequency

(a) Base clock

When the UAnCTL0.UAnPWR bit is 1, the clock selected by the UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits is supplied to the 8-bit counter. This clock is called the base clock. When the UAnPWR bit = 0, fUCLK is fixed to the low level.

(b) Serial clock generation

A serial clock can be generated by setting the UAnCTL1 register and the UAnCTL2 register (n = 0 to 2).

The base clock is selected by UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits.

The frequency division value for the 8-bit counter can be set using the UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits.

(2) UARTAn control register 1 (UAnCTL1)

The UAnCTL1 register is an 8-bit register that selects the UARTAn base clock.

This register can be read or written in 8-bit units.

Reset input clears this register to 00H.

★ **Caution Clear the UAnCTL0.UAnPWR bit to 0 before rewriting the UAnCTL1 register.**

After reset: 00H R/W Address: UA0CTL1 FFFFFA01H, UA1CTL1 FFFFFA11H, UA2CTL1 FFFFFA21H

UAnCTL1 (n = 0 to 2)

7	6	5	4	3	2	1	0
0	0	0	0	UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0

UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0	Base clock (f_{UCLK}) selection
0	0	0	0	fxx
0	0	0	1	fxx/2
0	0	1	0	fxx/4
0	0	1	1	fxx/8
0	1	0	0	fxx/16
0	1	0	1	fxx/32
0	1	1	0	fxx/64
0	1	1	1	fxx/128
1	0	0	0	fxx/256
1	0	0	1	fxx/512
1	0	1	0	fxx/1,024
1	0	1	1	External clock[Note] (ASCKA0 pin)
Other than above				Setting prohibited

Note Only UARTA0 is valid; setting UARTA1 and UARTA2 is prohibited.

Remark fxx: Main clock frequency

(3) UARTAn control register 2 (UAnCTL2)

The UAnCTL2 register is an 8-bit register that selects the baud rate (serial transfer speed) clock of UARTAn.
This register can be read or written in 8-bit units.
Reset input sets this register to FFH.

★ **Caution Clear the UAnCTL0.UAnPWR bit to 0 or clear the UAnTXE and UAnRXE bits to 00 before rewriting the UAnCTL2 register.**

After reset: FFH R/W Address: UA0CTL2 FFFFFA02H, UA1CTL2 FFFFFA12H, UA2CTL2 FFFFFA22H

	7	6	5	4	3	2	1	0
UAnCTL2 (n = 0 to 2)	UAnBRS7	UAnBRS6	UAnBRS5	UAnBRS4	UAnBRS3	UAnBRS2	UAnBRS1	UAnBRS0

UAn BRS7	UAn BRS6	UAn BRS5	UAn BRS4	UAn BRS3	UAn BRS2	UAn BRS1	UAn BRS0	Default (k)	Serial clock
0	0	0	0	0	0	×	×	×	Setting prohibited
0	0	0	0	0	1	0	0	4	$f_{UCLK}/4$
0	0	0	0	0	1	0	1	5	$f_{UCLK}/5$
0	0	0	0	0	1	1	0	6	$f_{UCLK}/6$
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	0	0	252	$f_{UCLK}/252$
1	1	1	1	1	1	0	1	253	$f_{UCLK}/253$
1	1	1	1	1	1	1	0	254	$f_{UCLK}/254$
1	1	1	1	1	1	1	1	255	$f_{UCLK}/255$

Remark f_{UCLK}: Clock frequency selected by the UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits

(4) Baud rate

The baud rate is obtained by the following equation.

$$\text{Baud rate} = \frac{f_{UCLK}}{2 \times k} \text{ [bps]}$$

f_{UCLK} = Frequency of base clock selected by the UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits
k = Value set using the UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits (k = 4, 5, 6, ..., 255)

(5) Baud rate error

The baud rate error is obtained by the following equation.

$$\text{Error (\%)} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Target baud rate (correct baud rate)}} - 1 \right) \times 100 \text{ [\%]}$$

Cautions 1. The baud rate error during transmission must be within the error tolerance on the receiving side.

2. The baud rate error during reception must satisfy the range indicated in (7) Allowable baud rate range during reception.

Example Base clock frequency = 20 MHz = 20,000,000 Hz
Setting value of UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits = 01000001B (k = 65)
Target baud rate = 153,600 bps

$$\text{Baud rate} = 20{,}000{,}000/(2 \times 65) = 153{,}846 \text{ [bps]}$$

$$\text{Error} = (153{,}846/153{,}600 - 1) \times 100 = 0.160 \text{ [\%]}$$

(6) Baud rate setting example

Table 13-3. Baud Rate Generator Setting Data

Baud Rate	f_{XX} = 20 MHz			f_{XX} = 16 MHz			f_{XX} = 10 MHz		
(bps)	UAnCTL1	UAnCTL2	ERR (%)	UAnCTL1	UAnCTL2	ERR (%)	UAnCTL1	UAnCTL2	ERR (%)
300	09H	41H	0.16	0AH	1AH	0.16	08H	41H	0.16
600	08H	41H	0.16	0AH	0DH	0.16	07H	41H	0.16
1,200	07H	41H	0.16	09H	0DH	0.16	06H	41H	0.16
2,400	06H	41H	0.16	08H	0DH	0.16	05H	41H	0.16
4,800	05H	41H	0.16	07H	0DH	0.16	04H	41H	0.16
9,600	04H	41H	0.16	06H	0DH	0.16	03H	41H	0.16
19,200	03H	41H	0.16	05H	0DH	0.16	02H	41H	0.16
31,250	06H	05H	0.00	06H	04H	0.00	05H	05H	0.00
38,400	02H	41H	0.16	04H	0DH	0.16	01H	41H	0.16
76,800	01H	41H	0.16	03H	0DH	0.16	00H	41H	0.16
153,600	00H	41H	0.16	02H	0DH	0.16	00H	21H	−1.36
312,500	03H	04H	0.00	01H	0DH	−1.54	02H	04H	0.00

Remark f_{XX}: Main clock frequency
ERR: Baud rate error (%)

(7) Allowable baud rate range during reception

The baud rate error range at the destination that is allowable during reception is shown below.

Caution The baud rate error during reception must be set within the allowable error range using the following equation.

Figure 13-14. Allowable Baud Rate Range During Reception

Remark n = 0 to 2

As shown in Figure 13-14, the receive data latch timing is determined by the counter set using the UAnCTL2 register following start bit detection. The transmit data can be normally received if up to the last data (stop bit) can be received in time for this latch timing.

When this is applied to 11-bit reception, the following is the theoretical result.

$$FL = (Brate)^{-1}$$

Brate: UARTAn baud rate (n = 0 to 2)

k: Setting value of UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits (n = 0 to 2)

FL: 1-bit data length

Latch timing margin: 2 clocks

Minimum allowable transfer rate: $FLmin = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$

Therefore, the maximum baud rate that can be received by the destination is as follows.

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k + 2} Brate$$

Similarly, obtaining the following maximum allowable transfer rate yields the following.

$$\frac{10}{11} \times FLmax = 11 \times FL - \frac{k + 2}{2 \times k} \times FL = \frac{21k - 2}{2 \times k} FL$$

$$FLmax = \frac{21k - 2}{20\ k} FL \times 11$$

Therefore, the minimum baud rate that can be received by the destination is as follows.

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k - 2} Brate$$

Obtaining the allowable baud rate error for UARTAn and the destination from the above-described equations for obtaining the minimum and maximum baud rate values yields the following.

Table 13-4. Maximum/Minimum Allowable Baud Rate Error

Division Ratio (k)	Maximum Allowable Baud Rate Error	Minimum Allowable Baud Rate Error
4	+2.32%	–2.43%
8	+3.52%	–3.61%
20	+4.26%	–4.30%
50	+4.56%	–4.58%
100	+4.66%	–4.67%
255	+4.72%	–4.72%

Remarks 1. The reception accuracy depends on the bit count in 1 frame, the input clock frequency, and the division ratio (k). The higher the input clock frequency and the larger the division ratio (k), the higher the accuracy.

2. k: Setting value of UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits (n = 0 to 2)

(8) Transfer rate during continuous transmission

During continuous transmission, the transfer rate from the stop bit to the next start bit is usually 2 base clocks longer. However, timing initialization is performed via start bit detection by the receiving side, so this has no influence on the transfer result.

Figure 13-15. Transfer Rate During Continuous Transmission

Assuming 1 bit data length: FL; stop bit length: FLstp; and base clock frequency: f_{UCLK}, we obtain the following equation.

$$FLstp = FL + 2/f_{UCLK}$$

Therefore, the transfer rate during continuous transmission is as follows.

$$\text{Transfer rate} = 11 \times FL + (2/f_{UCLK})$$

★ 13.7 Cautions

When the clock supply to UARTAn is stopped (for example, in IDLE1, IDLE2, or STOP mode), the operation stops with each register retaining the value it had immediately before the clock supply was stopped. The TXDAn pin output also holds and outputs the value it had immediately before the clock supply was stopped. However, the operation is not guaranteed after the clock supply is resumed. Therefore, after the clock supply is resumed, the circuits should be initialized by setting the UAnCTL0.UAnPWR, UAnCTL0.UAnRXEn, and UAnCTL0.UAnTXEn bits to 000.

CHAPTER 14 3-WIRE VARIABLE-LENGTH SERIAL I/O (CSIB)

14.1 Mode Switching of CSIB and Other Serial Interfaces

14.1.1 CSIB0 and I²C00 mode switching

In the μPD703229Y and 70F3229Y, CSIB0 and I²C00 are alternate functions of the same pin and therefore cannot be used simultaneously. CSIB0 and I²C00 switching must be set in advance using the PMC4 and PFC4 registers.

Caution The transmit/receive operation of CSIB0 and I²C00 is not guaranteed if these functions are switched during transmission or reception. Be sure to disable the one that is not used.

Figure 14-1. CSIB0 and I²C00 Mode Switch Settings

After reset: 00H R/W Address: FFFFF448H

	7	6	5	4	3	2	1	0
PMC4	0	0	0	0	0	PMC42	PMC41	PMC40

After reset: 00H R/W Address: FFFFF468H

	7	6	5	4	3	2	1	0
PFC4	0	0	0	0	0	0	PFC41	PFC40

PMC4n	PFC4n	Operation mode
0	×	Port I/O mode
1	0	CSIB0 mode
1	1	I²C00 mode

Remarks 1. n = 0, 1
2. × = don't care

14.2 Features

- Transfer rate: 10 Mbps to 4.9 kbps (f_{XX} = 20 MHz, using internal clock)
- Master mode and slave mode selectable
- 8-bit to 16-bit transfer, 3-wire serial interface
- Interrupt request signals (INTCBnT, INTCBnR) × 2
- Serial clock and data phase switchable
- Transfer data length selectable in 1-bit units between 8 and 16 bits
- Transfer data MSB-first/LSB-first switchable
- 3-wire transfer SOBn: Serial data output
  SIBn: Serial data input
  $\overline{\text{SCKBn}}$: Serial clock I/O

  Transmission mode, reception mode, and transmission/reception mode specifiable

Remark n = 0, 1

14.3 Configuration

The following shows the block diagram of CSIBn.

Figure 14-2. Block Diagram of CSIBn

Note When n = 0, f_{BRG}
When n = 1, TOP01

Remark n = 0, 1

CSIBn includes the following hardware.

Table 14-1. Configuration of CSIBn

Item	Configuration
Registers	CSIBn receive data register (CBnRX) CSIBn transmit data register (CBnTX)
Control registers	CSIBn control register 0 (CBnCTL0) CSIBn control register 1 (CBnCTL1) CSIBn control register 2 (CBnCTL2) CSIBn status register (CBnSTR)

(1) CSIBn receive data register (CBnRX)

The CBnRX register is a 16-bit buffer register that holds receive data.

This register is read-only, in 16-bit units.

The receive operation is started by reading the CBnRX register in the reception enabled status.

If the transfer data length is 8 bits, the lower 8 bits of this register are read-only in 8-bit units as the CBnRXL register.

Reset input clears this register to 0000H.

In addition to reset input, the CBnRX register can be initialized by clearing (to 0) the CBnPWR bit of the CBnCTL0 register.

After reset: 0000H R Address: CB0RX FFFFFD04H, CB1RX FFFFFD14H

(2) CSIB transmit data register (CBnTX)

The CBnTX register is a 16-bit buffer register used to write the CSIBn transfer data.

This register can be read or written in 16-bit units.

The transmit operation is started by writing data to the CBnTX register in the transmission enabled status.

If the transfer data length is 8 bits, the lower 8 bits of this register can be read or written in 8-bit units as the CBnTXL register.

Reset input clears this register to 0000H.

After reset: 0000H R/W Address: CB0TX FFFFFD06H, CB1TX FFFFFD16H

★ **Remark** The communication start conditions are shown below.

Transmission mode (CBnTXE bit = 1, CBnRXE bit = 0):	Write to CBnTX register
Transmission/reception mode (CBnTXE bit = 1, CBnRXE bit = 1):	Write to CBnTX register
Reception mode (CBnTXE bit = 0, CBnRXE bit = 1):	Read from CBnRX register

14.4 Control Registers

The following registers are used to control CSIBn.

- CSIBn control register 0 (CBnCTL0)
- CSIBn control register 1 (CBnCTL1)
- CSIBn control register 2 (CBnCTL2)
- CSIBn status register (CBnSTR)

★ **(1) CSIBn control register 0 (CBnCTL0)**

CBnCTL0 is a register that controls the CSIBn serial transfer operation.
This register can be read or written in 8-bit or 1-bit units.
Reset input sets this register to 01H.

(1/2)

After reset: 01H R/W Address: CB0CTL0 FFFFFD00H, CB1CTL0 FFFFFD10H

	<7>	<6>	<5>	<4>	3	2	1	<0>
CBnCTL0 (n = 0, 1)	CBnPWR	CBnTXE[Note]	CBnRXE[Note]	CBnDIR[Note]	0	0	CBnTMS[Note]	CBnSCE

CBnPWR	Specification of CSIBn operation disable/enable
0	Disable CSIBn operation and reset the CBnSTR register
1	Enable CSIBn operation
• The CBnPWR bit controls the CSIBn operation and resets the internal circuit.	

CBnTXE[Note]	Specification of transmit operation disable/enable
0	Disable transmit operation
1	Enable transmit operation
• The SOBn output is low level when the CBnTXE bit is 0.	

CBnRXE[Note]	Specification of receive operation disable/enable
0	Disable receive operation
1	Enable receive operation
• Because the receive operation is disabled by clearing the CBnRXE bit to 0, no reception complete interrupt is output even when the prescribed data is transferred, and the receive data (CBnRX register) is not updated.	

Note These bits can only be rewritten when the CBnPWR bit = 0. However, the CBnPWR bit can be set to 1 at the same time as rewriting these bits.

(2/2)

CBnDIR[Note 1]	Specification of transfer direction mode (MSB/LSB)
0	MSB first
1	LSB first

CBnTMS[Note 1]	Transfer mode specification
0	Single transfer mode
1	Continuous transfer mode

CBnSCE	Specification of start transfer disable/enable
0	Communication start trigger invalid
1	Communication start trigger valid

- In master mode
  This bit enables or disables the communication start trigger.
  (a) In single transmission or transmission/reception mode, or continuous transmission or continuous transmission/reception mode
  A communication operation can be started only when the CBnSCE bit is 1.
  Set the CBnSCE bit to 1.
  (b) In single reception mode
  Clear the CBnSCE bit to 0 before reading the receive data (CBnRX register)[Note 2].
  (c) In continuous reception mode
  Clear the CBnSCE bit to 0 one communication clock before reception of the last data is completed[Note 3].
- In slave mode
  This bit enables or disables the communication start trigger.
  Set the CBnSCE bit to 1.

Notes 1. These bits can only be rewritten when the CBnPWR bit = 0. However, the CBnPWR bit can be set to 1 at the same time as rewriting these bits.

2. If the CBnSCE bit is read while it is 1, the next communication operation is started.

3. If the CBnSCE bit is not cleared to 0 one communication clock before the completion of the last data reception, the next communication operation is automatically started.

(2) CSIBn control register 1 (CBnCTL1)

CBnCTL1 is an 8-bit register that controls the CSIBn serial transfer operation.

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

★ **Caution The CBnCTL1 register can be rewritten only when the CBnCTL0.CBnPWR bit = 0.**

After reset: 00H R/W Address: CB0CTL1 FFFFFD01H, CB1CTL1 FFFFFD11H

	7	6	5	4	3	2	1	0
CBnCTL1 (n = 0, 1)	0	0	0	CBnCKP	CBnDAP	CBnCKS2	CBnCKS1	CBnCKS0

	CBnCKP	CBnDAP	Specification of data transmission/ reception timing in relation to $\overline{\text{SCKBn}}$
Communication type 1	0	0	
Communication type 2	0	1	
Communication type 3	1	0	
Communication type 4	1	1	

CBnCKS2	CBnCKS1	CBnCKS0	Communication clock	Mode
0	0	0	$f_{XX}/2$	Master mode
0	0	1	$f_{XX}/4$	Master mode
0	1	0	$f_{XX}/8$	Master mode
0	1	1	$f_{XX}/16$	Master mode
1	0	0	$f_{XX}/32$	Master mode
1	0	1	$f_{XX}/64$	Master mode
1	1	0	**Note**	Master mode
1	1	1	External clock ($\overline{\text{SCKBn}}$)	Slave mode

Note When n = 0, f_{BRG}

When n = 1, TOP01

For details of baud rate generator, see **14.9 Baud Rate Generator**.

(3) CSIBn control register 2 (CBnCTL2)

CBnCTL2 is an 8-bit register that controls the number of CSIBn serial transfer bits.

This register can be read or written in 8-bit units.

Reset input clears this register to 00H.

Caution The CBnCTL2 register can be rewritten only when the CBnCTL0.CBnPWR bit = 0 or when both the CBnTXE and CBnRXE bits = 0.

After reset: 00H R/W Address: CB0CTL2 FFFFFD02H, CB1CTL2 FFFFFD12H

	7	6	5	4	3	2	1	0
CBnCTL2 (n = 0, 1)	0	0	0	0	CBnCL3	CBnCL2	CBnCL1	CBnCL0

CBnCL3	CBnCL2	CBnCL1	CBnCL0	Serial register bit length
0	0	0	0	8 bits
0	0	0	1	9 bits
0	0	1	0	10 bits
0	0	1	1	11 bits
0	1	0	0	12 bits
0	1	0	1	13 bits
0	1	1	0	14 bits
0	1	1	1	15 bits
1	×	×	×	16 bits

Remark If the number of transfer bits is other than 8 or 16, prepare and use data stuffed from the LSB of the CBnTX and CBnRX registers.

(a) Transfer data length change function

The CSIBn transfer data length can be set in 1-bit units between 8 and 16 bits using the CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits.

When the transfer bit length is set to a value other than 16 bits, set the data to the CBnTX or CBnRX register starting from the LSB, regardless of whether the transfer start bit is the MSB or LSB. Any data can be set for the higher bits that are not used, but the receive data becomes 0 following serial transfer.

(i) Transfer bit length = 10 bits, MSB first

(ii) Transfer bit length = 12 bits, LSB first

(4) CSIBn status register (CBnSTR)

CBnSTR is an 8-bit register that displays the CSIBn status.

This register can be read or written in 8-bit or 1-bit units, but the CBnTSF flag is read-only.

Reset input clears this register to 00H.

In addition to reset input, the CBnSTR register can be initialized by clearing (0) the CBnCTL0.CBnPWR bit.

After reset: 00H R/W Address: CB0STR FFFFFD03H, CB1STR FFFFFD13H

	<7>	6	5	4	3	2	1	<0>
CBnSTR (n = 0, 1)	CBnTSF	0	0	0	0	0	0	CBnOVE

CBnTSF	Communication status flag
0	Communication stopped
1	Communicating

- During transmission, this register is set when data is prepared in the CBnTX register, and during reception, it is set when a dummy read of the CBnRX register is performed.
  When transfer ends, this flag is cleared to 0 at the last edge of the clock.

CBnOVE	Overrun error flag
0	No overrun
1	Overrun

- An overrun error occurs when the next reception starts without performing a CPU read of the value of the receive buffer, upon completion of the receive operation. The CBnOVE flag displays the overrun error occurrence status in this case.
- The CBnOVE flag is cleared by writing 0 to it. It cannot be set even by writing 1 to it.

★ 14.5 Interrupt Request Signals

CSIBn can generate the following two types of interrupt request signals.

- Reception complete interrupt request signal (INTCBnR)
- Transmission enable interrupt request signal (INTCBnT)

Of these two interrupt request signals, the reception complete interrupt request signal has the higher priority by default, and the priority of the transmission enable interrupt request signal is lower.

Table 14-2. Interrupts and Their Default Priority

Interrupt	Priority
Reception complete	High
Transmission enable	Low

(1) Reception complete interrupt request signal (INTCBnR)

When receive data is transferred to the CBnRX register while reception is enabled, the reception complete interrupt request signal is generated.

This interrupt request signal can also be generated if a reception error occurs, instead of a reception error interrupt.

When the reception complete interrupt request signal is acknowledged and the data is read, read the CBnSTR register to check that the result of reception is not an error.

The reception complete interrupt request signal is not generated while reception is disabled.

(2) Transmission enable interrupt request signal (INTCBnT)

The transmission enable interrupt request signal is generated when transmit data is transferred from the CBnTX register while transmission enabled.

★ 14.6 Operation

14.6.1 Single transfer (master mode, transmission/reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (refer to **14.4 (2) CSIBn control register 1 (CBnCTL1)**), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0, 0, 0, 0)

(1) Clear the CBnCTL0.CBnPWR bit to 0.
(2) Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
(3) Set the CBnCTL0.CBnTXE, CBnCTL0.CBnRXE, and CBnCTL0.CBnSCE bits to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the transmission/reception enabled status.
(4) Set the CBnPWR bit to 1 to enable the CSIBn operation.
(5) Write transfer data to the CBnTX register (transmission start).
(6) The reception complete interrupt request signal (INTCBnR) is output.
(7) Read the CBnRX register before clearing the CBnPWR bit to 0.
(8) Check that the CBnSTR.CBnTSF bit = 0 and clear the CBnPWR bit to 0 to stop the operation of CSIBn (end of transmission/reception).

To continue transfer, repeat steps (5) to (7) before (8).
In transmission mode or transmission/reception mode, communication is not started by reading the CBnRX register.

Note In single transmission or single transmission/reception mode, the INTCBnT signal is not generated. When communication is complete, the INTCBnR signal is generated.

Remark The processing of steps (3) and (4) can be set simultaneously.

14.6.2 Single transfer mode (master mode, reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (refer to **14.4 (2) CSIBn control register 1 (CBnCTL1)**), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0, 0, 0, 0)

(1) Clear the CBnCTL0.CBnPWR bit to 0.
(2) Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
(3) Set the CBnCTL0.CBnRXE and CBnCTL0.CBnSCE bits to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the reception enabled status.
(4) Set the CBnPWR bit to 1 to enable the CSIBn operation.
(5) Perform a dummy read of the CBnRX register (reception start trigger).
(6) The reception complete interrupt request signal (INTCBnR) is output.
(7) Clear the CBnSCE bit to 0 to set the final receive data status.
(8) Read the CBnRX register.
(9) Check that the CBnSTR.CBnTSF bit = 0 and clear the CBnPWR bit to 0 to stop the operation of CSIBn (end of reception).

To continue transfer, repeat steps (5) and (6) before (7) (at this time, (5) is not a dummy read, but a receive data read combined with the reception trigger).

Remark The processing of steps (3) and (4) can be set simultaneously.

14.6.3 Continuous mode (master mode, transmission/reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 3 (refer to **14.4 (2) CSIBn control register 1 (CBnCTL1)**), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0, 0, 0, 0)

(1) Clear the CBnCTL0.CBnPWR bit to 0.
(2) Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
(3) Set the CBnCTL0.CBnTXE, CBnCTL0.CBnRXE, and CBnCTL0.CBnSCE bits to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the transmission/reception enabled status.
(4) Set the CBnPWR bit to 1 to enable the CSIBn operation.
(5) Write transfer data to the CBnTX register (transmission start).
(6) The transmission enable interrupt request signal (INTCBnT) is received and transfer data is written to the CBnTX register.
(7) The reception complete interrupt request signal (INTCBnR) is output.
Read the CBnRX register before the next receive data arrives or before the CBnPWR bit is cleared to 0.
(8) Check that the CBnSTR.CBnTSF bit = 0 and clear the CBnPWR bit to 0 to stop the operation of CSIBn (end of transmission/reception).

To continue transfer, repeat steps (5) to (7) before (8).

In transmission mode or transmission/reception mode, the communication is not started by reading the CBnRX register.

14.6.4 Continuous mode (master mode, reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 2 (refer to **14.4 (2) CSIBn control register 1 (CBnCTL1)**), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0, 0, 0, 0)

(1) Clear the CBnCTL0.CBnPWR bit to 0.
(2) Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
(3) Set the CBnCTL0.CBnRXE bit to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the reception enabled status.
(4) Set the CBnPWR bit to 1 to enable the CSIBn operation.
(5) Perform a dummy read of the CBnRX register (reception start trigger).
(6) The reception complete interrupt request signal (INTCBnR) is output.
Read the CBnRX register before the next receive data arrives or before the CBnPWR bit is cleared to 0.
(7) Clear the CBnCTL0.CBnSCE bit to 0 while the last data being received to set the final receive data status.
(8) Check that the CBnSTR.CBnTSF bit = 0 and clear the CBnPWR bit to 0 to stop the operation of CSIBn (end of reception).

To continue transfer, repeat steps (5) and (6) before (7).

14.6.5 Continuous reception mode (error)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 2 (refer to **14.4 (2) CSIBn control register 1 (CBnCTL1)**), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0, 0, 0, 0)

(1) Clear the CBnCTL0.CBnPWR bit to 0.
(2) Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
(3) Set the CBnCTL0.CBnRXE bit to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the reception enabled status.
(4) Set the CBnPWR bit to 1 to enable the CSIBn operation.
(5) Perform a dummy read of the CBnRX register (reception start trigger).
(6) The reception complete interrupt request signal (INTCBnR) is output.
(7) If the data could not be read before the end of the next transfer, the CBnSTR.CBnOVE flag is set to 1 upon the end of reception and the INTCBnR signal is output.
(8) Overrun error processing is performed after checking that the CBnOVE bit = 1 in the INTCBnR interrupt servicing.
(9) Clear CBnOVE bit to 0.
(10) Check that the CBnSTR.CBnTSF bit = 0 and clear the CBnPWR bit to 0 to stop the operation of CSIBn (end of reception).

14.6.6 Continuous mode (slave mode, transmission/reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 2 (refer to **14.4 (2) CSIBn control register 1 (CBnCTL1)**), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0, 0, 0, 0)

(1) Clear the CBnCTL0.CBnPWR bit to 0.

(2) Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.

(3) Set the CBnCTL0.CBnTXE, CBnCTL0.CBnRXE and CBnCTL0.CBnSCE bits to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the transmission/reception enabled status.

(4) Set the CBnPWR bit to 1 to enable the CSIBn operation.

(5) Write the transfer data to the CBnTX register.

(6) The transmission enable interrupt request signal (INTCBnT) is received and the transfer data is written to the CBnTX register.

(7) The reception complete interrupt request signal (INTCBnR) is output.
Read the CBnRX register.

(8) Check that the CBnSTR.CBnTSF bit = 0 and clear the CBnPWR bit to 0 to stop the operation of CSIBn (end of transmission/reception).

To continue transfer, repeat steps (5) to (7) before (8).

14.6.7 Continuous mode (slave mode, reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (refer to **14.4 (2) CSIBn control register 1 (CBnCTL1)**), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0, 0, 0, 0)

(1) Clear the CBnCTL0.CBnPWR bit to 0.
(2) Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
(3) Set the CBnCTL0.CBnRXE and CBnCTL0.CBnSCE bits to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the reception enabled status.
(4) Set the CBnPWR bit to 1 to enable the CSIBn operation.
(5) Perform a dummy read of the CBnRX register (reception start trigger).
(6) The reception complete interrupt request signal (INTCBnR) is output.
Read the CBnRX register.
(7) Check that the CBnSTR.CBnTSF bit = 0 and clear the CBnPWR bit to 0 to stop the operation of CSIBn (end of reception).

To continue transfer, repeat steps (5) and (6) before (7).

14.6.8 Clock timing

(1/2)

(i) Communication type 1 (CBnCKP = 0, CBnDAP = 0)

(ii) Communication type 3 (CBnCKP = 1, CBnDAP = 0)

Notes 1. The INTCBnT interrupt is set when the data written to the transmit buffer is transferred to the data shift register in the continuous transmission or continuous transmission/reception mode. In the single transmission or single transmission/reception mode, the INTCBnT interrupt request signal is not generated, but the INTCBnR interrupt request signal is generated upon completion of communication.

2. The INTCBnR interrupt occurs if reception is correctly completed and receive data is ready in the CBnRX register while reception is enabled, and if an overrun error occurs. In the single mode, the INTCBnR interrupt request signal is generated even in the transmission mode, upon completion of communication.

(2/2)

(iii) Communication type 2 (CBnCKP = 0, CBnDAP = 1)

(iv) Communication type 4 (CBnCKP = 1, CBnDAP = 1)

Notes 1. The INTCBnT interrupt is set when the data written to the transmit buffer is transferred to the data shift register in the continuous transmission or continuous transmission/reception mode. In the single transmission or single transmission/reception mode, the INTCBnT interrupt request signal is not generated, but the INTCBnR interrupt request signal is generated upon completion of communication.

2. The INTCBnR interrupt occurs if reception is correctly completed and receive data is ready in the CBnRX register while reception is enabled, and if an overrun error occurs. In the single mode, the INTCBnR interrupt request signal is generated even in the transmission mode, upon completion of communication.

14.7 Output Pins

(1) $\overline{\text{SCKBn}}$ pin

When CSIBn operation is disabled (CBnCTL0.CBnPWR bit = 0), the $\overline{\text{SCKBn}}$ pin output status is as follows.

★

CBnCKP	CBnCKS2	CBnCKS1	CBnCKS0	$\overline{\text{SCKBn}}$ Pin Output
0	Don't care	Don't care	Don't care	Fixed to high level
1	1	1	1	High impedance
	Other than above			Fixed to low level

Remarks 1. The output level of the $\overline{\text{SCKBn}}$ pin changes if any of the CBnCTL1.CBnCKP and CBnCKS2 to CBnCKS0 bits is rewritten.

2. n = 0, 1

(2) SOBn pin

When CSIBn operation is disabled (CBnPWR bit = 0), the SOBn pin output status is as follows.

CBnTXE	CBnDAP	CBnDIR	SOBn Pin Output
0	×	×	Fixed to low level
1	0	×	SOBn latch value (low level)
	1	0	CBnTXn value (MSB)
		1	CBnTXn value (LSB)

Remarks 1. The output level of the SOBn pin changes if any of the CBnCTL0.CBnTXE, CBnCTL0.CBnDIR, and CBnCTL1.CBnDAP bits is rewritten.

2. n = 0, 1

3. ×: don't care

★ 14.8 Operation Flow

(1) Single transmission

Note Set the CBnSCE bit to 1 in the initial setting.

Caution **In the slave mode, data cannot be correctly transmitted if the next transfer clock is input earlier than the CBnTX register is written.**

Remark n = 0, 1

(2) Single reception

Note Set the CBnSCE bit to 1 in the initial setting.

Caution **In the single mode, data cannot be correctly received if the next transfer clock is input earlier than the CBnRX register is read.**

Remark n = 0, 1

(3) Single transmission/reception

Notes 1. Set the CBnSCE bit to 1 in the initial setting.

2. If the next transfer is reception only, dummy data is written to the CBnTX register.

Caution **Even in the single mode, the CBnSTR.CBnOVE flag is set to 1. If only transmission is used in the transmission/reception mode, therefore, programming without checking the CBnOVE flag is recommended.**

Remark n = 0, 1

(4) Continuous transmission

Note Set the CBnSCE bit to 1 in the initial setting.

Remark n = 0, 1

(5) Continuous reception

Note Set the CBnSCE bit to 1 in the initial setting.

Caution **In the master mode, the clock is output without limit when dummy data is read from the CBnRX register. To stop the clock, execute the flow marked ◆ in the above flowchart.**
In the slave mode, malfunction due to noise during communication can be prevented by executing the flow marked ◆ in the above flowchart.
Before resuming communication, set the CBnCTL0.CBnSCE bit to 1, and read dummy data from the CBnRX register.

Remark n = 0, 1

(6) Continuous transmission/reception

Note Set the CBnSCE bit to 1 in the initial setting.

Remark n = 0, 1

14.9 Baud Rate Generator

The BRG0 baud rate generator is connected to CSIB0 and the watch timer (WT).

(1) Prescaler mode register 0 (PRSM0)

The PRSM0 register controls generation of the baud rate signal for CSIB0.

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF8B0H

	7	6	5	<4>	3	2	1	0
PRSM0	0	0	0	BGCE0	0	0	BGCS01	BGCS00

BGCE0	Baud rate output
0	Disabled
1	Enabled

BGCS01	BGCS00	Input clock selection (f_{BGCS})	Setting value (k)
0	0	f_{XX}	0
0	1	$f_{XX}/2$	1
1	0	$f_{XX}/4$	2
1	1	$f_{XX}/8$	3

Cautions 1. Do not rewrite the PRSM0 register during operation.

2. Set the PRSM0 register before setting the BGCE0 bit to 1.

(2) Prescaler compare register 0 (PRSCM0)

The PRSCM0 register is an 8-bit compare register.

This register can be read or written in 8-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF8B1H

	7	6	5	4	3	2	1	0
PRSCM0	PRSCM07	PRSCM06	PRSCM05	PRSCM04	PRSCM03	PRSCM02	PRSCM01	PRSCM00

Cautions 1. Do not rewrite the PRSCM0 register during operation.

2. Set the PRSCM0 register before setting the PRSM0.BGCE0 bit to 1.

14.9.1 Baud rate generation

The transmission/reception clock is generated by dividing the main clock. The baud rate generated from the main clock is obtained by the following equation.

$$f_{BRG} = \frac{f_X}{2^k \times N \times 2}$$

Remark f_{BRG}: BRG count clock

f_X: Main clock oscillation frequency

k: PRSM0 register setting value ($0 \leq k \leq 3$)

N: PRSCM0 register setting value (1 to FFH)

In the case of PRSCM0 register setting values 01H to FFH and 00H, N = 256

CHAPTER 15 I²C BUS

To use the I²C bus function, set the P40/SDA00 and P41/SCL00 pins to N-ch open-drain output (alternate function).

I²C00 has the following two modes.

- Operation stopped mode
- I²C (Inter IC) bus mode (multimasters supported)

(1) Operation stopped mode

In this mode, serial transfers are not performed, thus enabling a reduction in power consumption.

(2) I²C bus mode (multimasters supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock line (SCL00) and a serial data bus line (SDA00).

This mode complies with the I²C bus format and the master device can output “start condition”, “data”, and “stop condition” data to the slave device via the serial data bus. The slave device automatically detects these received data by hardware. This function can simplify the part of an application program that controls the I²C bus.

Since SCL00 and SDA00 are used for open-drain outputs, I²C00 requires pull-up resistors for the serial clock line and the serial data bus line.

15.1 Configuration

The block diagram of I²C00 is shown below.

Figure 15-1. Block Diagram of I²C00

A serial bus configuration example is shown below.

Figure 15-2. Serial Bus Configuration Example Using I²C Bus

I²C00 includes the following hardware.

Table 15-1. Configuration of I²C00

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 select register 0 (IICCL0) IIC function expansion register 0 (IICX0) IIC division clock select register 0 (OCKS0)

(1) IIC shift register 0 (IIC0)

The IIC0 register is used to convert 8-bit serial data to 8-bit parallel data and to convert 8-bit parallel data to 8-bit serial data, and can be used for both transmission and reception.

Write and read operations to the IIC0 register are used to control the actual transmit and receive operations.

This register can be read or written in 8-bit units.

Reset input clears this register to 00H.

(2) Slave address register 0 (SVA0)

The SVA0 register sets local addresses when in slave mode.

This register can be read or written in 8-bit units.

Reset input clears this register to 00H.

(3) SO latch

The SO latch is used to hold the output level of the SDA00 pin.

(4) Wakeup controller

This circuit generates an interrupt request when the address received by this register matches the address value set to the SVA0 register 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 and the serial clocks that are 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 I²C interrupt is generated following either of two triggers.

- Falling edge of eighth or ninth clock of the serial clock (set by IICC0.WTIM0 bit)
- Interrupt request generated when a stop condition is detected (set by IICC0.SPIE0 bit)

(8) Serial clock controller

In master mode, this circuit generates the clock output via the SCL00 pin from a sampling clock.

(9) Serial clock wait controller

This circuit controls the wait timing.

(10) Acknowledge output circuit, stop condition detector, start condition detector, and acknowledge detector

These circuits are used to output and detect various control signals.

(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

A start condition is issued when the IICC0.STT0 bit is set.

However, in the communication reservation disabled status (IICF0.IICRSV0 bit = 1), this request is ignored and the IICF0.STCF0 bit is set if the bus is not released (IICF0.IICBSY0 bit = 1).

(13) Bus status detector

Whether the bus is released or not is ascertained by detecting a start condition and stop condition.

However, the bus status cannot be detected immediately after operation, so set the bus status detector to the initial status by using the IICF0.STCEN0 bit.

15.2 I²C Control Registers

I²C00 is controlled by the following registers.

- IIC control register 0 (IICC0)
- IIC status register 0 (IICS0)
- IIC flag register 0 (IICF0)
- IIC clock select register 0 (IICCL0)
- IIC function expansion register 0 (IICX0)
- IIC division clock select register 0 (OCKS0)

The following registers are also used.

- IIC shift register 0 (IIC0)
- Slave address register 0 (SVA0)

Remark For the alternate-function pin settings, refer to **Table 4-16 Using Port Pin as Alternate-Function Pin**.

(1) IIC control register 0 (IICC0)

The IICC0 register is used to enable/stop I²C00 operations, set the wait timing, and set other I²C operations.
This register can be read or written in 8-bit or 1-bit units.
Reset input clears this register to 00H.

(1/4)

After reset: 00H R/W Address: FFFFFD82H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICC0	IICE0	LREL0	WREL0	SPIE0	WTIM0	ACKE0	STT0	SPT0

IICE0	I²C00 operation enable/disable specification
0	Operation stopped. IICS0 register reset[Note 1]. Internal operation stopped.
1	Operation enabled.
Condition for clearing (IICE0 bit = 0)	Condition for setting (IICE0 bit = 1)
• Cleared by instruction • After reset	• Set by instruction

LREL0	Exit from communications
0	Normal operation
1	This exits from the current communication operation and sets standby mode. This setting is automatically cleared after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCL00 and SDA00 lines are set for high impedance. The STT0 and SPT0 bits and the IICS0.MSTS0, IICS0.EXC0, IICS0.COI0, IICS0.TRC0, IICS0.ACKD0, and IICS0.STD0 bits are cleared.
The standby mode following exit from communications remains in effect until the following communication 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 bit = 0)[Note 2]	Condition for setting (LREL0 bit = 1)
• Automatically cleared after execution • After reset	• Set by instruction

WREL0	Wait cancellation control
0	Wait not canceled
1	Wait canceled. This setting is automatically cleared after wait is canceled.
Condition for clearing (WREL0 bit = 0)[Note 2]	Condition for setting (WREL0 bit = 1)
• Automatically cleared after execution • After reset	• Set by instruction

SPIE0	Enable/disable generation of interrupt request when stop condition is detected
0	Disabled
1	Enabled
Condition for clearing (SPIE0 bit = 0)[Note 2]	Condition for setting (SPIE0 bit = 1)
• Cleared by instruction • After reset	• Set by instruction

★ **Notes 1.** The IICS0 register, IICF0.STCF0 and IICF0.IICBSY0 bits, and IICCL0.CLD0 and IICCL0.DAD0 bits are reset.

2. This flag's signal is invalid when the IICE0 bit = 0.

(2/4)

WTIM0	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.
During address transfer, an interrupt occurs at the falling edge of the ninth clock regardless of this bit setting. This bit setting becomes valid when the address transfer is completed. 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 $\overline{\text{ACK}}$ signal is issued. When the slave device has received an extension code, however, a wait is inserted at the falling edge of the eighth clock.	
Condition for clearing (WTIM0 bit = 0)[Note]	Condition for setting (WTIM0 bit = 1)
• Cleared by instruction • After reset	• Set by instruction

ACKE0	Acknowledgment control
0	Acknowledgment disabled.
1	Acknowledgment enabled. During the ninth clock period, the SDA00 line is set to low level. However, the $\overline{\text{ACK}}$ is invalid in other than extension mode during address transfers.
Condition for clearing (ACKE0 bit = 0)[Note]	Condition for setting (ACKE0 bit = 1)
• Cleared by instruction • After reset	• Set by instruction

Note This flag's signal is invalid when the IICE0 bit = 0.

(3/4)

STT0	Start condition trigger
0	Start condition is not generated.
1	When bus is released (in STOP mode): Generates a start condition (for starting as master). The SDA00 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, SCL00 is changed to low level. During communication with a third party: If the communication reservation function is enabled (IICF0.IICRSV0 bit = 0) • This trigger functions as a start condition reserve flag. When set, it releases the bus and then automatically generates a start condition. If the communication reservation function is disabled (IICRSV0 bit = 1) • The IICF0.STCF0 bit is set. This trigger does not generate a start condition. In the wait state (when master device): Generates a restart condition after releasing the wait.

Cautions concerning set timing

- For master reception: Cannot be set during transfer. Can be set only when the ACKE0 bit has been set to 0 and slave has been notified of final reception.
- For master transmission: A start condition cannot be generated normally during the ACK0 period. Set during the wait period.
- For slave: Even when the communication reservation function is disabled (IICRSV0 bit = 1), the communication reservation status is entered.
- Cannot be set at the same time as SPT0

Condition for clearing (STT0 bit = 0)[Note]	Condition for setting (STT0 bit = 1)
• Cleared by loss in arbitration • Cleared after start condition is generated by master device • When the LREL0 bit = 1 (communication save) • When the IICE0 bit = 0 (operation stop) • After reset	• Set by instruction

Note This flag's signal is invalid when the IICE0 bit = 0.

Remark The STT0 bit is 0 if it is read immediately after data setting.

(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 SDA00 line goes to low level, either set the SCL00 line to high level or wait until it goes to high level. Next, after the rated amount of time has elapsed, the SDA00 line is changed from low level to high level and a stop condition is generated.

Cautions concerning set timing

For master reception: Cannot be set during transfer.
Can be set only when the ACKE0 bit has been set to 0 and during the wait period after slave has been notified of final reception.

For master transmission: A stop condition cannot be generated normally during the ACK0 period. Set during the wait period.

- Cannot be set at the same time as the STT0 bit.
  The SPT0 bit can be set only when in master mode.[Note 1]
  When the WTIM0 bit has been set to 0, if the SPT0 bit is set 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.
  When a ninth clock must be output, the WTIM0 bit should be changed from 0 to 1 during the wait period following output of eight clocks, and the SPT0 bit should be set during the wait period that follows output of the ninth clock.

Condition for clearing (SPT0 bit = 0)[Note 2]	Condition for setting (SPT0 bit = 1)
• Cleared by loss in arbitration • Automatically cleared after stop condition is detected • When the LREL0 bit = 1 (communication save) • When the IICE0 bit = 0 (operation stop) • After reset	• Set by instruction

Notes 1. Set the SPT0 bit only in master mode. However, when the IICRSV0 bit is 0, the SPT0 bit must be set and a stop condition generated before the first stop condition is detected following the switch to the operation enabled status. For details, refer to **15.13 Cautions**.

★ **2.** This flag's signal is invalid when the IICE0 bit = 0.

Caution When the TRC0 bit is set to 1, the WREL0 bit is set during the ninth clock and wait is canceled, after which the TRC0 bit is cleared and the SDA00 line is set to high impedance.

Remark The SPT0 bit is 0 if it is read immediately after data setting.

(2) IIC status register 0 (IICS0)

The IICS0 register indicates the status of the I²C00 bus.

This register is read-only, in 8-bit or 1-bit units.

Reset input clears this register to 00H.

★ **Caution When the subclock is operating and the main clock is stopped, accessing the IICS0 register is disabled. For details, refer to 3.4.8 (2).**

(1/3)

After reset: 00H R Address: FFFFFD86H

	<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 bit = 0)	Condition for setting (MSTS0 bit = 1)
• When a stop condition is detected • When the ALD0 bit = 1 (arbitration loss) • Cleared by LREL0 bit = 1 (communication save) • When the IICC0.IICE0 bit changes from 1 to 0 (operation stop) • After 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”. The MSTS0 bit is cleared.

Condition for clearing (ALD0 bit = 0)	Condition for setting (ALD0 bit = 1)
• Automatically cleared after the IICS0 register is read[Note] • When the IICE0 bit changes from 1 to 0 (operation stop) • After 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 bit = 0)	Condition for setting (EXC0 bit = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0 bit = 1 (communication save) • When the IICE0 bit changes from 1 to 0 (operation stop) • After 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 bit manipulation instruction is executed for bits other than the IICS0 register.

(2/3)

COI0	Detection of matching addresses
0	Addresses do not match.
1	Addresses match.

Condition for clearing (COI0 bit = 0)	Condition for setting (COI0 bit = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0 bit = 1 (communication save) • When the IICE0 bit changes from 1 to 0 (operation stop) • After reset	• When the received address matches the local address (SVA0 register) (set at the rising edge of the eighth clock).

TRC0	Detection of transmit/receive status
0	Receive status (other than transmit status). The SDA00 line is set for high impedance.
1	Transmit status. The value in the SO latch is enabled for output to the SDA00 line (valid starting at the rising edge of the first byte's ninth clock).

Condition for clearing (TRC0 bit = 0)	Condition for setting (TRC0 bit = 1)
• When a stop condition is detected • Cleared by LREL0 bit = 1 (communication save) • When the IICE0 bit changes from 1 to 0 (operation stop) • Cleared by IICC0.WREL0 bit = 1[Note] • When the ALD0 bit changes from 0 to 1 (arbitration loss) • After reset Master • When "1" is output to the first byte's LSB (transfer direction specification bit) Slave • When a start condition is detected When not used for communication	Master • When a start condition is generated Slave • When "1" is input to the first byte's LSB (transfer direction specification bit)

ACKD0	Detection of ACK
0	ACK was not detected.
1	ACK was detected.

Condition for clearing (ACKD0 bit = 0)	Condition for setting (ACKD0 bit = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock • Cleared by LREL0 bit = 1 (communication save) • When the IICE0 bit changes from 1 to 0 (operation stop) • After reset	• After the SDA00 line is set to low level at the rising edge of the SCL00's ninth clock

Note The TRC0 bit is cleared and SDA00 line becomes high impedance when the WREL0 bit is set and the wait state is canceled at the ninth clock by TRC0 bit = 1.

(3/3)

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 bit = 0)	Condition for setting (STD0 bit = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock following address transfer • Cleared by LREL0 bit = 1 (communication save) • When the IICE0 bit changes from 1 to 0 (operation stop) • After 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 bit = 0)	Condition for setting (SPD0 bit = 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 the IICE0 bit changes from 1 to 0 (operation stop) • After reset	• When a stop condition is detected

(3) IIC flag register 0 (IICF0)

The IICF0 register sets the I²C00 operation mode and indicates the I²C bus status.

This register can be read or written in 8-bit or 1-bit units. However, the STCF0 and IICBSY0 bits are read-only.

The IICRSV0 bit enables/disables the communication reservation function (refer to **15.12 Communication Reservation**).

The initial value of the IICBSY0 bit is set by using the STCEN0 bit (refer to **15.13 Cautions**).

The IICRSV0 and STCEN0 bits can be written only when operation of I²C00 is disabled (IICC0.IICE0 bit = 0).

After operation is enabled, the IICF0 register can be read.

Reset input clears this register to 00H.

(1/2)

After reset: 00H R/W[Note] Address: FFFFFD8AH

	<7>	<6>	5	4	3	2	<1>	<0>
IICF0	STCF0	IICBSY0	0	0	0	0	STCEN0	IICRSV0

STCF0	STT0 bit clear
0	Start condition issued
1	Start condition cannot be issued, STT0 bit cleared

Condition for clearing (STCF0 bit = 0)	Condition for setting (STCF0 bit = 1)
• Cleared by IICC0.STT0 bit = 1 • After reset	• When start condition is not issued and STT0 flag is cleared during communication reservation is disabled (IICRSV0 bit = 1)

IICBSY0	I²C00 bus status
0	Bus released status
1	Bus communication status

Condition for clearing (IICBSY0 bit = 0)	Condition for setting (IICBSY0 bit = 1)
• When stop condition is detected • After reset	• When start condition is detected • By setting the IICC0.IICE0 bit when the STCEN0 bit = 0

Note Bits 6 and 7 are read-only bits.

(2/2)

STCEN0	Initial start enable trigger
0	Start conditions cannot be generated until a stop condition is detected following operation enable (IICE0 bit = 1).
1	Start conditions can be generated even if a stop condition is not detected following operation enable (IICE0 bit = 1).

Condition for clearing (STCEN0 bit = 0)	Condition for setting (STCEN0 bit = 1)
• When start condition is detected • After reset	• Set by instruction

IICRSV0	Communication reservation function disable bit
0	Communication reservation enabled
1	Communication reservation disabled

Condition for clearing (IICRSV0 bit = 0)	Condition for setting (IICRSV0 bit = 1)
• Cleared by instruction • After reset	• Set by instruction

Cautions 1. Write the STCEN0 and IICRSV0 bits only when operation is stopped (IICE0 bit = 0).

2. When the STCEN0 bit = 1, the bus released status (IICBSY0 bit = 0) is recognized regardless of the actual bus status immediately after the I²C00 bus operation is enabled. Therefore, to issue the first start condition (STT0 bit = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

(4) IIC clock select register 0 (IICCL0)

The IICCL0 register is used to set the transfer clock for the I²C00 bus.

This register can be read or written in 8-bit or 1-bit units. However, the CLD0 and DAD0 bits are read-only. The SMC0, CL01, and CL00 bits are set by the combination of the IICX0.CLX0 bit and the OCKS0.OCKSTH0, OCKS0.OCKS01, and OCKS0.OCKS00 bits (refer to **15.2 (6) I²C00 transfer clock setting method**).

Reset input clears this register to 00H.

After reset: 00H R/W[Note] Address: FFFFFD84H

	7	6	<5>	<4>	3	2	1	0
IICCL0	0	0	CLD0	DAD0	SMC0	DFC0	CL01	CL00

CLD0	Detection of SCL00 line level (valid only when IICC0.IICE0 bit = 1)
0	SCL00 line was detected at low level.
1	SCL00 line was detected at high level.

Condition for clearing (CLD0 bit = 0)	Condition for setting (CLD0 bit = 1)
• When the SCL00 line is at low level • When the IICE0 bit = 0 (operation stop) • After reset	• When the SCL00 line is at high level

DAD0	Detection of SDA00 line level (valid only when IICE0 bit = 1)
0	SDA00 line was detected at low level.
1	SDA00 line was detected at high level.

Condition for clearing (DAD0 bit = 0)	Condition for setting (DAD0 bit = 1)
• When the SDA00 line is at low level • When the IICE0 bit = 0 (operation stop) • After reset	• When the SDA00 line 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 the DFC0 bit switching (on/off). The digital filter is used to eliminate noise in high-speed mode.	

Note Bits 4 and 5 are read-only bits.

Caution Always clear bits 7 and 6 to 0.

(5) IIC function expansion register 0 (IICX0)

The IICX0 register is a register used to set I²C00 function expansion (valid only in the high-speed mode).

This register can be read or written in 8-bit or 1-bit units.

The CLX0 bit is set by the combination of the IICCL0.SMC0, IICCL0.CL01, and IICCL0.CL00 bits and the OCKS0.OCKSTH0, OCKS0.OCKS01, and OCKS0.OCKS00 bits (refer to **15.2 (6) I²C00 transfer clock setting method**).

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFFD85H

	7	6	5	4	3	2	1	<0>
IICX0	0	0	0	0	0	0	0	CLX0

(6) I²C00 transfer clock setting method

The I²C00 transfer clock frequency (f_{SCL}) is calculated using the following expression.

$$f_{SCL} = 1/(m \times T + t_R + t_F)$$

m = 12, 24, 36, 48, 60, 72, 86, 88, 96, 132, 172, 176, 220, 258, 344 (refer to **Table 15-2 Selection Clock Settings**).

T: $1/f_{XX}$

t_R: SCL00 rise time

t_F: SCL00 fall time

For example, the I²C00 transfer clock frequency (f_{SCL}) when f_{XX} = 20 MHz, m = 220, t_R = 200 ns, and t_F = 50 ns is calculated using following expression.

$$f_{SCL} = 1/(220 \times 50\ \text{ns} + 200\ \text{ns} + 50\ \text{ns}) \cong 88.9\ \text{kHz}$$

The clock to be selected can be set by the combination of the IICCL0.SMC0, IICCL0.CL01, and IICCL0.CL00 bits, the IICX0.CLX0 bit, and the OCKS0.OCKSTH0, OCKS0.OCKS01, and OCKS0.OCKS00 bits.

Table 15-2. Selection Clock Settings

IICX0	IICCL0			Selection Clock	Transfer Clock	Settable Main Clock Frequency (f_{XX}) Range	Operating Mode
Bit 0	Bit 3	Bit 1	Bit 0				
CLX0	SMC0	CL01	CL00				
0	0	0	0	f_{XX}/2 (when OCKS0 = 10H set)	f_{XX}/88	4.00 MHz ≤ f_{XX} ≤ 8.38 MHz	Standard mode (SMC0 = 0)
				f_{XX}/3 (when OCKS0 = 11H set)	f_{XX}/132	6.00 MHz ≤ f_{XX} ≤ 12.57 MHz	
				f_{XX}/4 (when OCKS0 = 12H set)	f_{XX}/176	8.00 MHz ≤ f_{XX} ≤ 16.76 MHz	
				f_{XX}/5 (when OCKS0 = 13H set)	f_{XX}/220	10.00 MHz ≤ f_{XX} ≤ 20.00 MHz	
0	0	0	1	f_{XX}/2 (when OCKS0 = 10H set)	f_{XX}/172	8.38 MHz ≤ f_{XX} ≤ 16.76 MHz	
				f_{XX}/3 (when OCKS0 = 11H set)	f_{XX}/258	12.57 MHz ≤ f_{XX} ≤ 20.00 MHz	
				f_{XX}/4 (when OCKS0 = 12H set)	f_{XX}/344	16.76 MHz ≤ f_{XX} ≤ 20.00 MHz	
0	0	1	0	f_{XX} Note	f_{XX}/86	4.19 MHz ≤ f_{XX} ≤ 8.38 MHz	
0	1	0	×	f_{XX}/2 (when OCKS0 = 10H set)	f_{XX}/48	8.00 MHz ≤ f_{XX} ≤ 16.76 MHz	High-speed mode (SMC0 = 1)
				f_{XX}/3 (when OCKS0 = 11H set)	f_{XX}/72	12.00 MHz ≤ f_{XX} ≤ 20.00 MHz	
				f_{XX}/4 (when OCKS0 = 12H set)	f_{XX}/96	16.00 MHz ≤ f_{XX} ≤ 20.00 MHz	
0	1	1	0	f_{XX} Note	f_{XX}/24	4.00 MHz ≤ f_{XX} ≤ 8.38 MHz	
1	1	0	×	f_{XX}/2 (when OCKS0 = 10H set)	f_{XX}/24	8.00 MHz ≤ f_{XX} ≤ 8.38 MHz	
				f_{XX}/3 (when OCKS0 = 11H set)	f_{XX}/36	12.00 MHz ≤ f_{XX} ≤ 12.57 MHz	
				f_{XX}/4 (when OCKS0 = 12H set)	f_{XX}/48	16.00 MHz ≤ f_{XX} ≤ 16.67 MHz	
				f_{XX}/5 (when OCKS0 = 13H set)	f_{XX}/60	20.00 MHz	
1	1	1	0	f_{XX} Note	f_{XX}/12	4.00 MHz ≤ f_{XX} ≤ 4.19 MHz	
Other than above				Setting prohibited	–	–	–

Note Since the selection clock is f_{XX} regardless of the value set to the OCKS0 register, set the OCKS0 register = 00H (OPS = stopped status).

★ **(7) IIC division clock select register 0 (OCKS0)**

The OCKS0 register is used to control the I²C00 transfer clock.

This register can be read or written in 8-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF340H

	7	6	5	4	3	2	1	0
OCKS0	0	0	0	OCKSEN0	OCKSTH0	0	OCKS01	OCKS00

OCKSEN0	Operation setting of I²C division clock
0	Disable I²C division clock operation
1	Enable I²C division clock operation

OCKSTH0	OCKS01	OCKS00	Selected clock
0	0	0	$f_{XX}/2$
0	0	1	$f_{XX}/3$
0	1	0	$f_{XX}/4$
0	1	1	$f_{XX}/5$
1	0	0	f_{XX}
Other than above			Setting prohibited

(8) IIC shift register 0 (IIC0)

The IIC0 register is used for serial transmission/reception (shift operations) synchronized with the serial clock.

This register can be read or written in 8-bit units, but data should not be written to IIC0 during a data transfer.

A wait state is released by writing the IIC0 register during the wait period, and data transfer is started.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFFD80H

	7	6	5	4	3	2	1	0
IIC0								

(9) Slave address register 0 (SVA0)

The SVA0 register holds the I²C bus's slave addresses.

This register can be read or written in 8-bit units, but bit 0 should be fixed to 0.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFFD83H

	7	6	5	4	3	2	1	0
SVA0								0

15.3 I²C Bus Mode Functions

15.3.1 Pin configuration

The serial clock pin (SCL00) and serial data bus pin (SDA00) are configured as follows.

SCL00 ..... 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.
SDA00 ..... 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 15-3. Pin Configuration Diagram

15.4 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. The transfer timing for the "start condition", "data", and "stop condition" output via the I²C bus's serial data bus is shown below.

Figure 15-4. I²C Bus Serial Data Transfer Timing

The master device outputs the start condition, slave address, and stop condition.

The acknowledge signal ($\overline{ACK}$) can be output by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCL00) is continuously output by the master device. However, in the slave device, the SCL00's low-level period can be extended and a wait can be inserted.

15.4.1 Start condition

A start condition is met when the SCL00 pin is high level and the SDA00 pin changes from high level to low level. The start conditions for the SCL00 pin and SDA00 pin are signals that the master device outputs to the slave device when starting a serial transfer. When used as the slave device, the start condition can be detected.

Figure 15-5. Start Condition

A start condition is output when the IICC0.STT0 bit is set to 1 after a stop condition has been detected (IICS0.SPD0 bit = 1). When a start condition is detected, the IICS0.STD0 bit is set to 1.

15.4.2 Address

The 7 bits of data that follow the start condition are defined as an address.

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 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 the SVA0 register. If the address data matches the values of the SVA0 register, the slave device is selected and communicates with the master device until the master device transmits a start condition or stop condition.

Figure 15-6. Address

Note The interrupt request signal (INTIIC0) is generated if 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 **15.4.3 Transfer direction specification** below, are written together to the IIC0 register and then output. Received addresses are written to the IIC0 register.

The slave address is assigned to the higher 7 bits of the IIC0 register.

15.4.3 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 15-7. Transfer Direction Specification

Note The interrupt request signal (INTIIC0) is generated if a local address or extension code is received during slave device operation.

15.4.4 Acknowledge signal ($\overline{ACK}$)

The acknowledge signal ($\overline{ACK}$) is used by the transmitting and receiving devices to confirm serial data reception.

The receiving device returns one $\overline{ACK}$ signal for each 8 bits of data it receives. The transmitting device normally receives an $\overline{ACK}$ signal after transmitting 8 bits of data. However, when the master device is the receiving device, it does not output an $\overline{ACK}$ signal after receiving the final data to be transmitted. The transmitting device detects whether or not an $\overline{ACK}$ signal is returned after it transmits 8 bits of data. When an $\overline{ACK}$ signal is returned, the reception is judged as normal and processing continues. If the slave device does not return an $\overline{ACK}$ signal, the master device outputs either a stop condition or a restart condition and then stops the current transmission. Failure to return an $\overline{ACK}$ signal may be caused by the following two factors.

(a) Reception was not performed normally.

(b) The final data was received.

When the receiving device sets the SDA00 line to low level during the ninth clock, the $\overline{ACK}$ signal becomes active (normal receive response).

When the IICC0.ACKE0 bit is set to 1, automatic $\overline{ACK}$ signal generation is enabled.

Transmission of the eighth bit following the 7 address data bits causes the IICS0.TRC0 bit to be set. When this TRC0 bit's value is 0, it indicates receive mode. Therefore, the ACKE0 bit should be set to 1.

When the slave device is receiving (when TRC0 bit = 0), if the slave device does not need to receive any more data after receiving several bytes, setting the ACKE0 bit to 0 will prevent the master device from starting transmission of the subsequent data.

Similarly, when the master device is receiving (when TRC0 bit = 0) and the subsequent data is not needed and when either a restart condition or a stop condition should therefore be output, setting the ACKE0 bit to 0 will prevent the $\overline{ACK}$ signal from being returned. This prevents the MSB data from being output via the SDA00 line (i.e., stops transmission) during transmission from the slave device.

Figure 15-8. $\overline{ACK}$ Signal

When the local address is received, an $\overline{ACK}$ signal is automatically output in synchronization with the falling edge of the SCL00's eighth clock regardless of the value of the ACKE0 bit. No $\overline{ACK}$ signal is output if the received address is not a local address.

The $\overline{ACK}$ signal output method during data reception is based on the wait timing setting, as described below.

- When 8-clock wait is selected: (IICC0.WTIM0 bit = 0) — $\overline{ACK}$ signal is output at the falling edge of the SCL00's eighth clock if the ACKE0 bit is set to 1 before wait cancellation.
- When 9-clock wait is selected: (WTIM0 bit = 1) — $\overline{ACK}$ signal is automatically output at the falling edge of the SCL00's eighth clock if the ACKE0 bit has already been set to 1.

15.4.5 Stop condition

When the SCL00 pin is high level, changing the SDA00 pin from low level to high level generates a stop condition.

A stop condition is a signal that the master device outputs to the slave device when serial transfer has been completed. When used as the slave device, the stop condition can be detected.

Figure 15-9. Stop Condition

A stop condition is generated when the IICC0.SPT0 bit is set to 1. When the stop condition is detected, the IICS0.SPD0 bit is set to 1 and the INTIIC0 signal is generated when the IICC0.SPIE0 bit is set to 1.

15.4.6 Wait signal ($\overline{\text{WAIT}}$)

The wait signal ($\overline{\text{WAIT}}$) 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 SCL00 pin to low level notifies the communication partner of the wait status. When the wait status has been canceled for both the master and slave devices, the next data transfer can begin.

Figure 15-10. Wait Signal (1/2)

Figure 15-10. Wait Signal (2/2)

A wait may be automatically generated depending on the setting of the IICC0.WTIM0 bit.

Normally, when the IICC0.WREL0 bit is set to 1 or when FFH is written to the IIC0 register on the receiving side, the wait status is canceled and the transmitting side writes data to the IIC0 register to cancel the wait status.

The master device can also cancel the wait status via either of the following methods.

- By setting the IICC0.STT0 bit to 1
- By setting the IICC0.SPT0 bit to 1

15.5 I²C Interrupt Request Signal (INTIIC0)

The following shows the value of the IICS0 register at the INTIIC0 interrupt request signal generation timing and at the INTIIC0 interrupt timing.

15.5.1 Master device operation

(1) Start ~ Address ~ Data ~ Data ~ Stop (normal transmission/reception)

<1> When WTIM0 bit = 0

▲1: IICS0 register = 10XXX110B

▲2: IICS0 register = 10XXX000B

▲3: IICS0 register = 10XXX000B (WTIM0 bit = 1)

▲4: IICS0 register = 10XXXX00B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1

▲1: IICS0 register = 10XXX110B

▲2: IICS0 register = 10XXX100B

▲3: IICS0 register = 10XXXX00B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(2) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

<1> When WTIM0 bit = 0

▲1: IICS0 register = 10XXX110B

▲2: IICS0 register = 10XXX000B (WTIM0 bit = 1)

▲3: IICS0 register = 10XXXX00B (WTIM0 bit = 0)

▲4: IICS0 register = 10XXX110B (WTIM0 bit = 0)

▲5: IICS0 register = 10XXX000B (WTIM0 bit = 1)

▲6: IICS0 register = 10XXXX00B

Δ 7: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1

▲1: IICS0 register = 10XXX110B

▲2: IICS0 register = 10XXXX00B

▲3: IICS0 register = 10XXX110B

▲4: IICS0 register = 10XXXX00B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(3) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

<1> When WTIM0 bit = 0

▲1: IICS0 register = 1010X110B

▲2: IICS0 register = 1010X000B

▲3: IICS0 register = 1010X000B (WTIM0 bit = 1)

▲4: IICS0 register = 1010XX00B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1

▲1: IICS0 register = 1010X110B

▲2: IICS0 register = 1010X100B

▲3: IICS0 register = 1010XX00B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

15.5.2 Slave device operation (when receiving slave address data (matches with SVA0))

(1) Start ~ Address ~ Data ~ Data ~ Stop

<1> When WTIM0 bit = 0

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
				▲1	▲2		▲3	Δ4

▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001X000B

▲3: IICS0 register = 0001X000B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
				▲1		▲2	▲3	Δ4

▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001X100B

▲3: IICS0 register = 0001XX00B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(2) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIM0 bit = 0 (after restart, match with SVA0)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
				▲1	▲2					▲3	▲4	Δ5

▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001X000B

▲3: IICS0 register = 0001X110B

▲4: IICS0 register = 0001X000B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1 (after restart, match with SVA0)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
				▲1		▲2				▲3	▲4	Δ5

▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001XX00B

▲3: IICS0 register = 0001X110B

▲4: IICS0 register = 0001XX00B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(3) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

<1> When WTIM0 bit = 0 (after restart, extension code reception)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
				▲1	▲2				▲3		▲4	Δ5

▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001X000B

▲3: IICS0 register = 0010X010B

▲4: IICS0 register = 0010X000B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1 (after restart, extension code reception)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
				▲1		▲2			▲3	▲4	▲5	Δ6

▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001XX00B

▲3: IICS0 register = 0010X010B

▲4: IICS0 register = 0010X110B

▲5: IICS0 register = 0010XX00B

Δ 6: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(4) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIM0 bit = 0 (after restart, mismatch with address (= not extension code))

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
				▲1	▲2					▲3		Δ4

▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001X000B

▲3: IICS0 register = 00000X10B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1 (after restart, mismatch with address (= not extension code))

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
				▲1		▲2				▲3		Δ4

▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001XX00B

▲3: IICS0 register = 00000X10B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

15.5.3 Slave device operation (when receiving extension code)

(1) Start ~ Code ~ Data ~ Data ~ Stop

<1> When WTIM0 bit = 0

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1		▲2		▲3	Δ4

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X000B

▲3: IICS0 register = 0010X000B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1	▲2		▲3	▲4	Δ5

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X110B

▲3: IICS0 register = 0010X100B

▲4: IICS0 register = 0010XX00B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(2) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIM0 bit = 0 (after restart, match with SVA0)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1		▲2					▲3	▲4	Δ5

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X000B

▲3: IICS0 register = 0001X110B

▲4: IICS0 register = 0001X000B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1 (after restart, match with SVA0)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1	▲2		▲3				▲4	▲5	Δ6

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X110B

▲3: IICS0 register = 0010XX00B

▲4: IICS0 register = 0001X110B

▲5: IICS0 register = 0001XX00B

Δ 6: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(3) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

<1> When WTIM0 bit = 0 (after restart, extension code reception)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1		▲2				▲3		▲4	Δ5

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X000B

▲3: IICS0 register = 0010X010B

▲4: IICS0 register = 0010X000B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1 (after restart, extension code reception)

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1	▲2		▲3			▲4	▲5	▲6	Δ7

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X110B

▲3: IICS0 register = 0010XX00B

▲4: IICS0 register = 0010X010B

▲5: IICS0 register = 0010X110B

▲6: IICS0 register = 0010XX00B

Δ 7: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(4) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIM0 bit = 0 (after restart, mismatch with address (= not extension code))

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1		▲2					▲3		Δ4

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X000B

▲3: IICS0 register = 00000X10B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1 (after restart, mismatch with address (= not extension code))

ST	AD6 to AD0	RW	AK	D7 to D0	AK	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
			▲1	▲2		▲3				▲4		Δ5

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X110B

▲3: IICS0 register = 0010XX00B

▲4: IICS0 register = 00000X10B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

15.5.4 Operation without communication

(1) Start ~ Code ~ Data ~ Data ~ Stop

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
								Δ1

Δ 1: IICS0 register = 00000001B

Remark Δ: Generated only when SPIE0 bit = 1

15.5.5 Arbitration loss operation (operation as slave after arbitration loss)

(1) When arbitration loss occurs during transmission of slave address data

<1> When WTIM0 bit = 0

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
				▲1	▲2		▲3	Δ4

▲1: IICS0 register = 0101X110B (Example: When ALD0 bit is read during interrupt servicing)

▲2: IICS0 register = 0001X000B

▲3: IICS0 register = 0001X000B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
				▲1		▲2	▲3	Δ4

▲1: IICS0 register = 0101X110B (Example: When ALD0 bit is read during interrupt servicing)

▲2: IICS0 register = 0001X100B

▲3: IICS0 register = 0001XX00B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(2) When arbitration loss occurs during transmission of extension code

<1> When WTIM0 bit = 0

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1		▲2		▲3	Δ4

▲1: IICS0 register = 0110X010B (Example: When ALD0 bit is read during interrupt servicing)

▲2: IICS0 register = 0010X000B

▲3: IICS0 register = 0010X000B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
			▲1	▲2		▲3	▲4	Δ5

▲1: IICS0 register = 0110X010B (Example: When ALD0 bit is read during interrupt servicing)

▲2: IICS0 register = 0010X110B

▲3: IICS0 register = 0010X100B

▲4: IICS0 register = 0010XX00B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

15.5.6 Operation when arbitration loss occurs (no communication after arbitration loss)

(1) When arbitration loss occurs during transmission of slave address data

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 Δ2

▲1: IICS0 register = 01000110B (Example: When ALD0 bit is read during interrupt servicing)

Δ 2: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1

(2) When arbitration loss occurs during transmission of extension code

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 Δ2

▲1: IICS0 register = 0110X010B (Example: When ALD0 bit is read during interrupt servicing)

IICC0.LREL0 bit is set to 1 by software

Δ 2: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(3) When arbitration loss occurs during data transfer

<1> When WTIM0 bit = 0

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
				▲1	▲2			Δ3

▲1: IICS0 register = 10001110B

▲2: IICS0 register = 01000000B (Example: When ALD0 bit is read during interrupt servicing)

Δ 3: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1

<2> When WTIM0 bit = 1

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	SP
				▲1		▲2		Δ3

▲1: IICS0 register = 10001110B

▲2: IICS0 register = 01000100B (Example: When ALD0 bit is read during interrupt servicing)

Δ 3: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1

(4) When arbitration loss occurs due to restart condition during data transfer

<1> Not extension code (Example: Mismatches with SVA0)

ST	AD6 to AD0	RW	AK	D7 to Dn	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
				▲1					▲2		Δ3

▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 01000110B (Example: When ALD0 bit is read during interrupt servicing)

Δ 3: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care
Dn = D6 to D0

<2> Extension code

ST	AD6 to AD0	RW	AK	D7 to Dn	ST	AD6 to AD0	RW	AK	D7 to D0	AK	SP
				▲1					▲2		Δ3

▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 0110X010B (Example: When ALD0 bit is read during interrupt servicing)

IICC0.LREL0 bit is set to 1 by software

Δ 3: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care
Dn = D6 to D0

(5) When arbitration loss occurs due to stop condition during data transfer

ST	AD6 to AD0	RW	AK	D7 to Dn	SP

▲1 Δ2

▲1: IICS0 register = 1000X110B

Δ 2: IICS0 register = 01000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care
Dn = D6 to D0

(6) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

When WTIM0 bit = 1

STT0 bit = 1
↓

ST	AD6 to AD0	RW	AK	D7 to D0	AK	D7 to D0	AK	D7 to D0	AK	SP

▲1 ▲2 ▲3 Δ4

▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000XX00B

▲3: IICS0 register = 01000100B (Example: When ALD0 bit is read during interrupt servicing)

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(7) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

When WTIM0 bit = 1

▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000XX00B

Δ 3: IICS0 register = 01000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(8) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

When WTIM0 bit = 1

▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000XX00B

▲3: IICS0 register = 01000000B (Example: When ALD0 bit is read during interrupt servicing)

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

15.6 Interrupt Request Signal (INTIIC0) Generation Timing and Wait Control

The setting of the IICC0.WTIM0 bit determines the timing at which the INTIIC0 signal is generated and the corresponding wait control, as shown below.

Table 15-3. 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 occur at the falling edge of the ninth clock only when there is a match with the address set to the SVA0 register.
At this point, the $\overline{ACK}$ signal is output regardless of the value set to the IICC0.ACKE0 bit. For a slave device that has received an extension code, the INTIIC0 signal occurs at the falling edge of the eighth clock.
When the address does not match after restart, the INTIIC0 signal occurs at the falling edge of the ninth clock, but no wait occurs.

2. If the received address does not match the contents of the SVA0 register and an extension code is not received, neither the INTIIC0 signal 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 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.

- By setting the IICC0.WREL0 bit to 1
- By writing to the IIC0 register
- By start condition setting (IICC0.STT0 bit = 1)
- By stop condition setting (IICC0.SPT0 bit = 1)

When an 8-clock wait has been selected (WTIM0 bit = 0), the output level of $\overline{\text{ACK}}$ must be determined prior to wait cancellation.

(5) Stop condition detection

The INTIIC0 signal is generated when a stop condition is detected.

15.7 Address Match Detection Method

In I²C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

Address match detection is performed automatically by hardware. The INTIIC0 signal occurs when a local address has been set to the SVA0 register and when the address set to the SVA0 register matches the slave address sent by the master device, or when an extension code has been received.

15.8 Error Detection

In I²C bus mode, the status of the serial data bus (SDA00) during data transmission is captured by the IIC0 register of the transmitting device, so the data of the IIC0 register 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.

15.9 Extension Code

(1) When the higher 4 bits of the receive address are either 0000 or 1111, the extension code reception flag (IICS0.EXC0 bit) is set for extension code reception and an interrupt request signal (INTIIC0) is issued at the falling edge of the eighth clock.
The local address stored in the SVA0 register is not affected.

(2) If 11110xx0 is set to the SVA0 register by a 10-bit address transfer and 11110xx0 is transferred from the master device, the results are as follows. Note that the INTIIC0 signal occurs at the falling edge of the eighth clock.

- Higher four bits of data match: EXC0 bit = 1
- Seven bits of data match: IICS0.COI0 bit = 1

(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.
For example, when operation as a slave is not desired after the extension code is received, set the IICC0.LREL0 bit to 1 and the CPU will enter the next communication wait state.

Table 15-4. Extension Code Bit Definitions

Slave Address	R/W Bit	Description
0000 000	0	General call address
0000 000	1	Start byte
0000 001	X	CBUS address
0000 010	X	Address that is reserved for different bus format
1111 0xx	X	10-bit slave address specification

15.10 Arbitration

When several master devices simultaneously output a start condition (when the IICC0.STT0 bit is set to 1 before the IICS0.STD0 bit is set to 1), communication among the master devices is performed while the number of clocks are being 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 (IICS0.ALD0 bit) is set to 1 via the timing by which the arbitration loss occurred, and the SCL00 and SDA00 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 signal (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALD0 bit = 1 setting that has been made by software.

For details of interrupt request timing, refer to **15.5 I²C Interrupt Request Signal (INTIIC0)**.

Figure 15-11. Arbitration Timing Example

Table 15-5. Status During Arbitration and Interrupt Request Signal Generation Timing

Status During Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer[Note 1]
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During $\overline{\text{ACK}}$ signal transfer period after data reception	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is output (when IICC0.SPIE0 bit = 1)[Note 2]
When data is at low level while attempting to output a restart condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When stop condition is detected while attempting to output a restart condition	When stop condition is output (when IICC0.SPIE0 bit = 1)[Note 2]
When data is at low level while attempting to output a stop condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When SCL00 is at low level while attempting to output a restart condition	

Notes 1. When the IICC0.WTIM0 bit = 1, an interrupt request occurs at the falling edge of the ninth clock. When the WTIM0 bit = 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 possibility that arbitration will occur, set the SPIE0 bit = 1 for master device operation.

15.11 Wakeup Function

The I²C 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 interrupt request signals 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 output a start condition) to a slave device.

However, when a stop condition is detected, the IICC0.SPIE0 bit is set regardless of the wakeup function, and this determines whether interrupt request signals are enabled or disabled.

15.12 Communication Reservation

15.12.1 When communication reservation function is enabled (IICF0.IICRSV0 bit = 0)

To start master device communications when not currently using the bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes in 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 ($\overline{ACK}$ is not returned and the bus was released when the IICC0.LREL0 bit was set to 1).

If the IICC0.STT0 bit is set to 1 while the bus is not used, a start condition is automatically generated and wait status is set after the bus is released (after a stop condition is detected).

When the bus release is detected (when a stop condition is detected), writing to the IIC0 register causes the master's address transfer to start. At this point, the IICC0.SPIE0 bit should be set to 1.

When the STT0 bit 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

To detect which operation mode has been determined for the STT0 bit , set the STT0 bit to 1, wait for the wait period, then check the IICS0.MSTS0 bit.

The wait periods, which should be set via software, are listed in Table 15-6. These wait periods can be set by the IICCL0.SMC0, IICCL0.CL01, and IICCL0.CL00 bits.

Table 15-6. Wait Periods

SMC0	CL01	CL00	Wait Period
0	0	0	26 clocks
0	0	1	46 clocks
0	1	0	92 clocks
0	1	1	37 clocks
1	0	0	16 clocks
1	0	1	
1	1	0	32 clocks
1	1	1	13 clocks

The communication reservation timing is shown below.

Figure 15-12. Communication Reservation Timing

Communication reservations are accepted via the following timing. After the STD0 bit is set to 1, a communication reservation can be made by setting the STT0 bit to 1 before a stop condition is detected.

Figure 15-13. Timing for Accepting Communication Reservations

The communication reservation flowchart is illustrated below.

Figure 15-14. Communication Reservation Flowchart

Note The communication reservation operation executes a write to the IIC0 register when a stop condition interrupt request occurs.

15.12.2 When communication reservation function is disabled (IICF0.IICRSV0 bit = 1)

When the IICC0.STT0 bit is set when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. There are two modes in 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 ($\overline{ACK}$ is not returned and the bus was released when the IICC0.LREL0 bit was set to 1).

To confirm whether the start condition was generated or request was rejected, check the IICF0.STCF0 flag. The time shown in Table 15-7 is required until the STCF0 flag is set after setting the STT0 bit to 1. Therefore, secure the time by software.

Table 15-7. Wait Periods

OCKSEN0	OCKS01	OCKS00	CL01	CL00	Wait Period
1	0	0	0	×	6 clocks
1	0	1	0	×	9 clocks
1	1	0	0	×	12 clocks
1	1	1	0	×	15 clocks
0	0	0	1	0	3 clocks

Remark ×: don't care

15.13 Cautions

(1) When IICF0.STCEN0 bit = 0

Immediately after the I²C operation is enabled, the bus communication status (IICF0.IICBSY0 bit = 1) is recognized regardless of the actual bus status. To execute master communication in the status where a stop condition has not been detected, generate a stop condition and then release the bus before starting the master communication.

Use the following sequence for generating a stop condition.

<1> Set the IICCL0 register.
<2> Set the IICC0.IICE0 bit.
<3> Set the IICC0.SPT0 bit.

(2) When IICF0.STCEN0 bit = 1

Immediately after I²C operation is enabled, the bus released status (IICBSY0 bit = 0) is recognized regardless of the actual bus status. To issue the first start condition (IICC0.STT0 bit = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

15.14 Communication Operations

15.14.1 Master operation 1

The following shows the flowchart for master communication when the communication reservation function is enabled (IICF0.IICRSV0 bit = 0) and the master operation is started after a stop condition is detected (IICF0.STCEN0 bit = 0).

Figure 15-15. Master Operation Flowchart (1)

15.14.2 Master operation 2

The following shows the flowchart for master communication when the communication reservation function is disabled (IICF0.IICRSV0 bit = 1) and the master operation is started without detecting a stop condition (IICF0.STCEN0 bit = 1).

Figure 15-16. Master Operation Flowchart (2)

★ 15.14.3 Slave operation

The following shows the processing procedure of the slave operation.

Basically, the operation of the slave device is event-driven. Therefore, processing by an INTIIC0 interrupt (processing requiring a significant change of the operation status, such as stop condition detection during communication) is necessary.

The following description assumes that data communication does not support extension codes. Also, it is assumed that the INTIIC0 interrupt servicing performs only status change processing and that the actual data communication is performed during the main processing.

Figure 15-17. Software Outline During Slave Operation

Therefore, the following three flags are prepared so that the data transfer processing can be performed by transmitting these flags to the main processing instead of INTIIC0 signal.

(1) Communication mode flag

This flag indicates the following communication statuses.

Clear mode: Data communication not in progress

Communication mode: Data communication in progress (valid address detection to stop condition detection, $\overline{\text{ACK}}$ signal from master not detected, address mismatch)

(2) Ready flag

This flag indicates that data communication is enabled. This is the same status as an INTIIC0 interrupt during normal data transfer. This flag is set in the interrupt processing block and cleared in the main processing block. The ready flag for the first data for transmission is not set in the interrupt processing block, so the first data is transmitted without clear processing (the address match is regarded as a request for the next data).

(3) Communication direction flag

This flag indicates the direction of communication and is the same as the value of IICS0.TRC0 bit.

The following shows the operation of the main processing block during slave operation.

Start I²C00 and wait for the communication enabled status. When communication is enabled, perform transfer using the communication mode flag and ready flag (the processing of the stop condition and start condition is performed by interrupts, conditions are confirmed by flags).

For transmission, repeat the transmission operation until the master device stops returning $\overline{\text{ACK}}$ signal. When the master device stops returning $\overline{\text{ACK}}$ signal, transfer is complete.

For reception, receive the required number of data and do not return $\overline{ACK}$ signal for the next data immediately after transfer is complete. After that, the master device generates the stop condition or restart condition. This causes exit from communications.

Figure 15-18. Slave Operation Flowchart (1)

The following shows an example of the processing of the slave device by an INTIIC0 interrupt (it is assumed that no extension codes are used here). During an INTIIC0 interrupt, the status is confirmed and the following steps are executed.

<1> When a stop condition is detected, communication is terminated.

<2> When a start condition is detected, the address is confirmed. If the address does not match, communication is terminated. If the address matches, the communication mode is set and wait is released, and operation returns from the interrupt (the ready flag is cleared).

<3> For data transmission/reception, when the ready flag is set, operation returns from the interrupt while the I²C00 bus remains in the wait status.

Remark <1> to <3> in the above correspond to <1> to <3> in **Figure 15-19 Slave Operation Flowchart (2)**.

Figure 15-19. Slave Operation Flowchart (2)

15.15 Timing of Data Communication

When using I²C 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 IICS0.TRC0 bit, which specifies the data transfer direction, and then starts serial communication with the slave device.

The shift operation of the IIC0 register is synchronized with the falling edge of the serial clock (SCL00). The transmit data is transferred to the SO latch and is output (MSB first) via the SDA00 pin.

Data input via the SDA00 pin is captured by the IIC0 register at the rising edge of SCL00.

The data communication timing is shown below.

Figure 15-20. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)

Note To cancel slave wait, write FFH to IIC0 or set WREL0.

Figure 15-20. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)

Note To cancel slave wait, write FFH to IIC0 or set WREL0.

Figure 15-20. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)

Note To cancel slave wait, write FFH to IIC0 or set WREL0.

Figure 15-21. Example of Slave to Master Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)

Note To cancel master wait, write FFH to IIC0 or set WREL0.

Figure 15-21. Example of Slave to Master Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)

Note To cancel master wait, write FFH to IIC0 or set WREL0.

Figure 15-21. Example of Slave to Master Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)

(c) Stop condition

Note To cancel master wait, write FFH to IIC0 or set WREL0.

CHAPTER 16 DMA FUNCTION (DMA CONTROLLER)

The μPD703229Y and 70F3229Y include a direct memory access (DMA) controller (DMAC) that executes and controls DMA transfer.

The DMAC controls data transfer between memory and I/O, between memories, or between I/Os based on DMA requests issued by the on-chip peripheral I/O (serial interface, timer/counter, and A/D converter), interrupts from external input pins, or software triggers (memory refers to internal RAM or external memory).

16.1 Features

- 4 independent DMA channels
- Transfer unit: 8/16 bits
- Maximum transfer count: 65,536 (2^{16})
- Transfer type: Two-cycle transfer
- Transfer mode: Single transfer mode
- Transfer requests
  - Request by interrupts from on-chip peripheral I/O (serial interface, timer/counter, A/D converter) or interrupts from external input pin
  - Requests by software trigger
- Transfer targets
  - Internal RAM ↔ Peripheral I/O
  - Peripheral I/O ↔ Peripheral I/O
  - Internal RAM ↔ External memory
  - External memory ↔ Peripheral I/O
  - External memory ↔ External memory

16.2 Configuration

Remark n = 0 to 3

★16.3 Control Registers

(1) DMA source address registers 0 to 3 (DSA0 to DSA3)

The DSA0 to DSA3 registers set the DMA source addresses (26 bits each) for DMA channel n (n = 0 to 3).
These registers are divided into two 16-bit registers, DSAnH and DSAnL.
These registers can be read or written in 16-bit units.

After reset: Undefined R/W Address: DSA0H FFFFF082H, DSA1H FFFFF08AH,
DSA2H FFFFF092H, DSA3H FFFFF09AH,
DSA0L FFFFF080H, DSA1L FFFFF088H,
DSA2L FFFFF090H, DSA3L FFFFF098H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSAnH (n = 0 to 3)	IR	0	0	0	0	0	SA25	SA24	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSAnL (n = 0 to 3)	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0

IR	Specification of DMA transfer source
0	External memory or on-chip peripheral I/O
1	Internal RAM

SA25 to SA16	Set the address (A25 to A16) of the DMA transfer source (default value is undefined). During DMA transfer, the next DMA transfer source address is held. When DMA transfer is completed, the DMA address set first is held.

SA15 to SA0	Set the address (A15 to A0) of the DMA transfer source (default value is undefined). During DMA transfer, the next DMA transfer source address is held. When DMA transfer is completed, the DMA address set first is held.

Cautions 1. Be sure to set bits 14 to 10 of the DSAnH register to 0.

2. Set the DSAnH and DSAnL registers at the following timing while DMA is not in progress.
- **Period from after reset to start of first DMA transfer**
- **Period from after channel initialization by DCHCn.INITn bit to start of DMA transfer**
- **Period from after completion of DMA transfer (DCHCn.TCn bit = 1) to start of the next DMA transfer**

3. When the value of the DSAn register is read, two 16-bit registers, DSAnH and DSAnL, are read. If reading and updating conflict, the value being updated may be read (refer to 16.13 Cautions).

(2) DMA destination address registers 0 to 3 (DDA0 to DDA3)

The DDA0 to DDA3 registers set the DMA destination address (26 bits each) for DMA channel n (n = 0 to 3). These registers are divided into two 16-bit registers, DDAnH and DDAnL.

These registers can be read or written in 16-bit units.

After reset: Undefined R/W Address: DDA0H FFFFF086H, DDA1H FFFFF08EH, DDA2H FFFFF096H, DDA3H FFFFF09EH, DDA0L FFFFF084H, DDA1L FFFFF08CH, DDA2L FFFFF094H, DDA3L FFFFF09CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDAnH (n = 0 to 3)	IR	0	0	0	0	0	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDAnL (n = 0 to 3)	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

IR	Specification of DMA transfer destination
0	External memory or on-chip peripheral I/O
1	Internal RAM

DA25 to DA16	Set an address (A25 to A16) of DMA transfer destination (default value is undefined). During DMA transfer, the next DMA transfer destination address is held. When DMA transfer is completed, the DMA transfer source address set first is held.

DA15 to DA0	Set an address (A15 to A0) of DMA transfer destination (default value is undefined). During DMA transfer, the next DMA transfer destination address is held. When DMA transfer is completed, the DMA transfer source address set first is held.

Cautions 1. Be sure to set bits 14 to 10 of the DDAnH register to 0.

2. Set the DDAnH and DDAnL registers at the following timing while DMA is not in progress.

- **Period from after reset to start of first DMA transfer**
- **Period from after channel initialization by DCHCn.INITn bit to start of DMA transfer**
- **Period from after completion of DMA transfer (DCHCn.TCn bit = 1) to start of the next DMA transfer**

3. When the value of the DDAn register is read, two 16-bit registers, DDAnH and DDAnL, are read. If reading and updating conflict, a value being updated may be read (refer to 16.13 Cautions).

(3) DMA byte count registers 0 to 3 (DBC0 to DBC3)

The DBC0 to DBC3 registers are 16-bit registers that set the byte transfer count for DMA channel n (n = 0 to 3). These registers hold the remaining transfer count during DMA transfer.

These registers are decremented by 1 per one transfer regardless of the transfer data unit (8/16 bits), and the transfer is terminated if a borrow occurs.

These registers can be read or written in 16-bit units.

After reset: Undefined R/W Address: DBC0 FFFFF0C0H, DBC1 FFFFF0C2H, DBC2 FFFFF0C4H, DBC3 FFFFF0C6H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBCn (n = 0 to 3)	BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0

BC15 to BC0	Byte transfer count setting or remaining byte transfer count during DMA transfer
0000H	Byte transfer count 1 or remaining byte transfer count
0001H	Byte transfer count 2 or remaining byte transfer count
:	:
FFFFH	Byte transfer count 65,536 (2^{16}) or remaining byte transfer count
The number of transfer data set first is held when DMA transfer is complete.	

Caution Set the DBCn register at the following timing while DMA is not in progress.

- **Period from after reset to start of first DMA transfer**
- **Period from after channel initialization by DCHCn.INITn bit to start of DMA transfer**
- **Period from after completion of DMA transfer (DCHCn.TCn bit = 1) to start of the next DMA transfer**

(4) DMA addressing control registers 0 to 3 (DADC0 to DADC3)

The DADC0 to DADC3 registers are 16-bit registers that control the DMA transfer mode for DMA channel n (n = 0 to 3).

These registers can be read or written in 16-bit units.

Reset input clears these registers to 0000H.

After reset: 0000H R/W Address: DADC0 FFFFF0D0H, DADC1 FFFFF0D2H, DADC2 FFFFF0D4H, DADC3 FFFFF0D6H

DADCn (n = 0 to 3)

15	14	13	12	11	10	9	8
0	DS0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
SAD1	SAD0	DAD1	DAD0	0	0	0	0

DS0	Setting of transfer data size
0	8 bits
1	16 bits

SAD1	SAD0	Setting of count direction of the transfer source address
0	0	Increment
0	1	Decrement
1	0	Fixed
1	1	Setting prohibited

DAD1	DAD0	Setting of count direction of the destination address
0	0	Increment
0	1	Decrement
1	0	Fixed
1	1	Setting prohibited

Cautions 1. Be sure to set bits 15, 13 to 8, and 3 to 0 of the DADCn register to 0.

2. Set the DADCn register at the following timing while DMA is not in progress.
- **Period from after reset to start of first DMA transfer**
- **Period from after channel initialization by DCHCn.INITn bit to start of DMA transfer**
- **Period from after completion of DMA transfer (DCHCn.TCn bit = 1) to start of the next DMA transfer**

3. The DS0 bit specifies the size of the transfer data, and does not control bus sizing. If 8-bit data (DS0 bit = 0) is set, therefore, the lower data bus is not always used.

4. If the transfer data size is set to 16 bits (DS0 bit = 1), transfer cannot be started from an odd address. Transfer is always started from an address with the first bit of the lower address aligned to 0.

5. If DMA transfer is executed on an on-chip peripheral I/O register (as the transfer source or destination), be sure to specify the same transfer size as the register size. For example, to execute DMA transfer on an 8-bit register, be sure to specify 8-bit transfer.

(5) DMA channel control registers 0 to 3 (DCHC0 to DCHC3)

The DCHC0 to DCHC3 registers are 8-bit registers that control the DMA transfer operating mode for DMA channel n.

These registers can be read or written in 8-bit or 1-bit units. (However, bit 7 is read-only and bits 1 and 2 are write-only. If bit 1 or 2 is read, the read value is always 0.)

Reset input clears these registers to 00H.

After reset: 00H R/W Address: DCHC0 FFFFF0E0H, DCHC1 FFFFF0E2H, DCHC2 FFFFF0E4H, DCHC3 FFFFF0E6H

	<7>	6	5	4	3	<2>	<1>	<0>
DCHCn	TCn[Note 1]	0	0	0	0	INITn[Note 2]	STGn[Note 2]	Enn

(n = 0 to 3)

TCn[Note 1]	Status flag indicates whether DMA transfer through DMA channel n has completed or not
0	DMA transfer had not completed.
1	DMA transfer had completed.
It is set to 1 on the last DMA transfer and cleared to 0 when it is read.	

INITn[Note 2]	If the INITn bit is set to 1 with DMA transfer disabled (Enn bit = 0), the DMA transfer status can be initialized. When re-setting the DMA transfer status (re-setting the DDAnH, DDAnL, DSAnH, DSAnL, DBCn, and DADCn registers) before DMA transfer is completed (before the TCn bit is set to 1), be sure to initialize the DMA channel. When initializing the DMA controller, however, be sure to observe the procedure described in **16.13 Cautions**.

STGn[Note 2]	This is a software startup trigger of DMA transfer. If this bit is set to 1 in the DMA transfer enable state (TCn bit = 0, Enn bit = 1), DMA transfer is started.

Enn	Setting of whether DMA transfer through DMA channel n is to be enabled or disabled
0	DMA transfer disabled
1	DMA transfer enabled
DMA transfer is enabled when the Enn bit is set to 1. When DMA transfer is completed (when a terminal count is generated), this bit is automatically cleared to 0. To abort DMA transfer, clear the Enn bit to 0 by software. To resume, set the Enn bit to 1 again. When aborting or resuming DMA transfer, however, be sure to observe the procedure described in **16.13 Cautions**.	

Notes 1. The TCn bit is read-only.

2. The INITn and STGn bits are write-only.

Cautions 1. Be sure to set bits 6 to 3 of the DCHCn register to 0.

2. When DMA transfer is completed (when a terminal count is generated), the Enn bit is cleared to 0 and then the TCn bit is set to 1. If the DCHCn register is read while its bits are being updated, a value indicating "transfer not completed and transfer is disabled" (TCn bit = 0 and Enn bit = 0) may be read.

(6) DMA trigger factor registers 0 to 3 (DTFR0 to DTFR3)

The DTFR0 to DTFR3 registers are 8-bit registers that control the DMA transfer start trigger via interrupt request signals from on-chip peripheral I/O.

The interrupt request signals set by these registers serve as DMA transfer start factors.

These registers can be read or written in 8-bit units. However, DFn bit can be read or written in 1-bit units.

Reset input clears these registers to 00H.

After reset: 00H R/W Address: DTFR0 FFFFF810H, DTFR1 FFFFF812H, DTFR2 FFFFF814H, DTFR3 FFFFF816H

	<7>	6	5	4	3	2	1	0
DTFRn	DFn	0	IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0

(n = 0 to 3)

DFn[Note]	DMA transfer request flag
0	No DMA transfer request
1	DMA transfer request

Note The DFn bit is a write-only bit. Write 0 to this bit to clear a DMA transfer request if an interrupt that is specified as the cause of starting DMA transfer occurs while DMA transfer is disabled.

Cautions 1. Set the DTFRn.IFCn5 to DTFRn.IFCn0 bits at the following timing while DMA is not in progress.

- **Period from after reset to start of first DMA transfer**
- **Period from after channel initialization by DCHCn.INITn bit to start of DMA transfer**
- **Period from after completion of DMA transfer (DCHCn.TCn bit = 1) to start of the next DMA transfer**

2. An interrupt request that is generated in the standby mode (IDLE1, IDLE2, STOP, or sub-IDLE mode) does not start the DMA transfer cycle (nor is the DFn bit set to 1).

3. If a DMA start factor is selected by the IFCn5 to IFCn0 bits, the DFn bit is set to 1 when an interrupt occurs from the selected on-chip peripheral I/O, regardless of whether the DMA operation is enabled or disabled. If DMA is enabled in this status, DMA transfer is immediately started.

Remark For the IFCn5 to IFCn0 bits, refer to **Table 16-1 DMA Start Factors**.

Table 16-1. DMA Start Factors

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	Interrupt Source
0	0	0	0	0	0	DMA request by interrupt disabled
0	0	0	0	0	1	INTLVI
0	0	0	0	1	0	INTP0
0	0	0	0	1	1	INTP1
0	0	0	1	0	0	INTP2
0	0	0	1	0	1	INTP3
0	0	0	1	1	0	INTP4
0	0	0	1	1	1	INTP5
0	0	1	0	0	0	INTP6
0	0	1	0	0	1	INTP7
0	0	1	0	1	0	INTTQ0OV
0	0	1	0	1	1	INTTQ0CC0
0	0	1	1	0	0	INTTQ0CC1
0	0	1	1	0	1	INTTQ0CC2
0	0	1	1	1	0	INTTQ0CC3
0	0	1	1	1	1	INTTP0OV
0	1	0	0	0	0	INTTP0CC0
0	1	0	0	0	1	INTTP0CC1
0	1	0	0	1	0	INTTP1OV
0	1	0	0	1	1	INTTP1CC0
0	1	0	1	0	0	INTTP1CC1
0	1	0	1	0	1	INTTP2OV
0	1	0	1	1	0	INTTP2CC0
0	1	0	1	1	1	INTTP2CC1
0	1	1	0	0	0	INTTP3OV
0	1	1	0	0	1	INTTP3CC0
0	1	1	0	1	0	INTTP3CC1
0	1	1	0	1	1	INTTM0EQ0
0	1	1	1	0	0	INTCB0R/INTIIC0
0	1	1	1	0	1	INTCB0T
0	1	1	1	1	0	INTCB1R
0	1	1	1	1	1	INTCB1T
1	0	0	0	0	0	INTUA0R
1	0	0	0	0	1	INTUA0T
1	0	0	0	1	0	INTUA1R
1	0	0	0	1	1	INTUA1T
1	0	0	1	0	0	INTAD
1	0	1	0	0	1	INTKR
1	0	1	1	1	1	INTUA2R
1	1	0	0	0	0	INTUA2T
Setting prohibited for other than above						–

Remark n = 0 to 3

★ 16.4 Transfer Targets

Table 16-2 shows the relationship between the transfer targets (√: Transfer enabled, ×: Transfer disabled).

Table 16-2. Relationship Between Transfer Targets

		Transfer Destination			
		Internal ROM	On-Chip Peripheral I/O	Internal RAM	External Memory
Source	On-chip peripheral I/O	×	√	√	√
	Internal RAM	×	√	×	√
	External memory	×	√	√	√
	Internal ROM	×	×	×	×

Caution The operation is not guaranteed for combinations of transfer destination and source marked with "×" in Table 16-2.

★ 16.5 Transfer Modes

Single transfer is supported as the transfer mode.

In single transfer mode, the bus is released at each byte/halfword transfer. If there is a subsequent DMA transfer request, transfer is performed again once. This operation continues until a terminal count occurs.

When the DMAC has released the bus, if another higher priority DMA transfer request is issued, the higher priority DMA request always takes precedence.

If a new transfer request of the same channel and a transfer request of another channel with a lower priority are generated in a transfer cycle, DMA transfer of the channel with the lower priority is executed after the bus is released to the CPU (the new transfer request of the same channel is ignored in the transfer cycle).

★ 16.6 Transfer Types

As a transfer type, the two-cycle transfer is supported.

In two-cycle transfer, data transfer is performed in two cycles, a read cycle and a write cycle.

In the read cycle, the transfer source address is output and reading is performed from the source to the DMAC. In the write cycle, the transfer destination address is output and writing is performed from the DMAC to the destination.

An idle cycle of one clock is always inserted between a read cycle and a write cycle. If the data bus width differs between the transfer source and destination for DMA transfer of two cycles, the operation is performed as follows.

<16-bit data transfer>

<1> Transfer from 32-bit bus → 16-bit bus
A read cycle (the higher 16 bits are in a high-impedance state) is generated, followed by generation of a write cycle (16 bits).

<2> Transfer from 16-/32-bit bus to 8-bit bus
A 16-bit read cycle is generated once, and then an 8-bit write cycle is generated twice.

<3> Transfer from 8-bit bus to 16-/32-bit bus
An 8-bit read cycle is generated twice, and then a 16-bit write cycle is generated once.

<4> Transfer between 16-bit bus and 32-bit bus
A 16-bit read cycle is generated once, and then a 16-bit write cycle is generated once.

For DMA transfer executed to an on-chip peripheral I/O register (transfer source/destination), be sure to specify the same transfer size as the register size. For example, for DMA transfer to an 8-bit register, be sure to specify byte (8-bit) transfer.

Remark The bus width of each transfer target (transfer source/destination) is as follows.

- On-chip peripheral I/O: 16-bit bus width
- Internal RAM: 32-bit bus width
- External memory: 8-bit or 16-bit bus width

★ 16.7 DMA Channel Priorities

The DMA channel priorities are fixed as follows.

DMA channel 0 > DMA channel 1 > DMA channel 2 > DMA channel 3

The priorities are checked for every transfer cycle.

★ 16.8 Time Related to DMA Transfer

The time required to respond to a DMA request, and the minimum number of clocks required for DMA transfer are shown below.

Single transfer: DMA response time (<1>) + Transfer source memory access (<2>) + 1[Note 1] + Transfer destination memory access (<2>)

DMA Cycle		Minimum Number of Execution Clocks
<1> DMA request response time		4 clocks (MIN.) + Noise elimination time[Note 2]
<2> Memory access	External memory access	Depends on connected memory.
	Internal RAM access	2 clocks[Note 3]
	Peripheral I/O register access	3 clocks + Number of wait cycles specified by VSWC register[Note 4]

Notes 1. One clock is always inserted between a read cycle and a write cycle in DMA transfer.

2. If an external interrupt (INTPn) is specified as the trigger to start DMA transfer, noise elimination time is added (n = 0 to 7).

3. Two clocks are required for a DMA cycle.

4. More wait cycles are necessary for accessing a special on-chip peripheral I/O register (for details, refer to **3.4.8 (2)**).

★16.9 DMA Transfer Start Factors

There are two types of DMA transfer start factors, as shown below.

(1) Request by software

If the STGn bit is set to 1 while the DCHCn.TCn bit = 0 and Enn bit = 1 (DMA transfer enabled), DMA transfer is started.

To request the next DMA transfer cycle immediately after that, confirm, by using the DBCn register, that the preceding DMA transfer cycle has been completed, and set the STGn bit to 1 again (n = 0 to 3).

TCn bit = 0, Enn bit = 1
↓
STGn bit = 1 … Starts the first DMA transfer.
↓
Confirm that the contents of the DBCn register have been updated.
STGn bit = 1 … Starts the second DMA transfer.
↓
:
↓
Generation of terminal count … Enn bit = 0, TCn bit = 1, and INTDMAn signal is generated.

(2) Request by on-chip peripheral I/O

If an interrupt request is generated from the on-chip peripheral I/O set by the DTFRn register when the DCHCn.TCn bit = 0 and Enn bit = 1 (DMA transfer enabled), DMA transfer is started.

Cautions 1. Two start factors (software trigger and hardware trigger) cannot be used for one DMA channel. If two start factors are simultaneously generated for one DMA channel, only one of them is valid. The start factor that is valid cannot be identified.

2. A new transfer request that is generated after the preceding DMA transfer request was generated or in the preceding DMA transfer cycle is ignored (cleared).

3. The transfer request interval of the same DMA channel varies depending on the setting of bus wait in the DMA transfer cycle, the start status of the other channels, or the external bus hold request. In particular, as described in Caution 2, a new transfer request that is generated for the same channel before the DMA transfer cycle or during the DMA transfer cycle is ignored. Therefore, the transfer request intervals for the same DMA channel must be sufficiently separated by the system. When the software trigger is used, completion of the DMA transfer cycle that was generated before can be checked by updating the DBCn register.

★ 16.10 DMA Abort Factors

DMA transfer is aborted if a bus hold occurs.

The same applies if transfer is executed between the internal memory/on-chip peripheral I/O and internal memory/on-chip peripheral I/O.

When the bus hold is cleared, DMA transfer is resumed.

★ 16.11 End of DMA Transfer

When DMA transfer has been completed the number of times set to the DBCn register and when the DCHCn.Enn bit is cleared to 0 and TCn bit is set to 1, a DMA transfer end interrupt request signal (INTDMAn) is generated for the interrupt controller (INTC) (n = 0 to 3).

The μPD703229Y and 70F3229Y do not output a terminal count signal to an external device. Therefore, confirm completion of DMA transfer by using the DMA transfer end interrupt or polling the TCn bit.

★ 16.12 Operation Timing

The DMA transfer operation timing is shown below.

Figure 16-1. Priority of DMA (1)

Remarks 1. Transfer in the order of DMA0 → DMA1 → DMA2

2. In the case of transfer between external memory spaces (multiplexed bus, no wait)

Figure 16-2. Priority of DMA (2)

Remarks 1. Transfer in the order of DMA0 → DMA1 → DMA0 (DMA2 is held pending.)

2. In the case of transfer between external memory spaces (multiplexed bus, no wait)

Figure 16-3. Period in Which DMA Transfer Request Is Ignored (1)

Notes 1. Interrupt from on-chip peripheral I/O, or software trigger (STGn bit)

2. New DMA request of the same channel is ignored between when the first request is generated and the end processing is complete.

Remark In the case of transfer between external memory spaces (multiplexed bus, no wait)

Figure 16-4. Period in Which DMA Transfer Request Is Ignored (2)

<1> DMA0 transfer request

<2> New DMA0 transfer request is generated during DMA0 transfer.

→ A DMA transfer request of the same channel is ignored during DMA transfer.

<3> Requests for DMA0 and DMA1 are generated at the same time.

→ DMA0 request is ignored (a DMA transfer request of the same channel during transfer is ignored).

→ DMA1 request is acknowledged.

<4> Requests for DMA0, DMA1, and DMA2 are generated at the same time.

→ DMA1 request is ignored (a DMA transfer request of the same channel during transfer is ignored).

→ DMA0 request is acknowledged according to priority. DMA2 request is held pending (transfer of DMA2 occurs next).

★16.13 Cautions

(1) Caution for VSWC register

When using the DMAC, be sure to set an appropriate value, in accordance with the operating frequency, to the VSWC register.

When the default value (77H) of the VSWC register is used, or if an inappropriate value is set to the VSWC register, the operation is not correctly performed (for details of the VSWC register, refer to **3.4.8 (1) (a) System wait control register (VSWC)**).

(2) Caution for DMA transfer executed on internal RAM

When executing the following instructions located in the internal RAM, do not execute a DMA transfer that transfers data to/from the internal RAM (transfer source/destination), because the CPU may not operate correctly afterward.

- Bit manipulation instruction located in internal RAM (SET1, CLR1, or NOT1)
- Data access instruction to misaligned address located in internal RAM

Conversely, when executing a DMA transfer to transfer data to/from the internal RAM (transfer source/destination), do not execute the above two instructions.

(3) Caution for reading DCHCn.TCn bit (n = 0 to 3)

The TCn bit is cleared to 0 when it is read, but it is not automatically cleared even if it is read at a specific timing. To accurately clear the TCn bit, add the following processing.

(a) When waiting for completion of DMA transfer by polling TCn bit

Confirm that the TCn bit has been set to 1 (after TCn bit = 1 is read), and then read the TCn bit three more times.

(b) When reading TCn bit in interrupt servicing routine

Execute reading the TCn bit three times.

(4) DMA transfer initialization procedure (setting DCHCn.INITn bit to 1)

Even if the INITn bit is set to 1 when the channel executing DMA transfer is to be initialized, the channel may not be initialized. To accurately initialize the channel, execute either of the following two procedures.

(a) Temporarily stop transfer of all DMA channels

Initialize the channel executing DMA transfer using the procedure in <1> to <7> below.

Note, however, that TCn bit is cleared to 0 when step <5> is executed. Make sure that the other processing programs do not expect that the TCn bit is 1.

<1> Disable interrupts (DI).

<2> Read the DCHCn.Enn bit of DMA channels other than the one to be forcibly terminated, and transfer the value to a general-purpose register.

<3> Clear the Enn bit of the DMA channels used (including the channel to be forcibly terminated) to 0. To clear the Enn bit of the last DMA channel, execute the clear instruction twice. If the target of DMA transfer (transfer source/destination) is the internal RAM, execute the instruction three times.

Example: Execute instructions in the following order if channels 0, 1, and 2 are used (if the target of transfer is not the internal RAM).

- Clear the DCHC0.E00 bit to 0.
- Clear the DCHC1.E11 bit to 0.
- Clear the DCHC2.E22 bit to 0.
- Clear the DCHC2.E22 bit to 0 again.

<4> Set the INITn bit of the channel to be forcibly terminated to 1.

<5> Read the TCn bit of each channel not to be forcibly terminated. If both the TCn bit and the Enn bit read in <2> are 1 (logical product (AND) is 1), clear the saved Enn bit to 0.

<6> After the operation in <5>, write the Enn bit value to the DCHCn register.

<7> Enable interrupts (EI).

Caution Be sure to execute step <5> above to prevent illegal setting of the Enn bit of the channels whose DMA transfer has been normally completed between <2> and <3>.

(b) Repeatedly execute setting INITn bit until transfer is forcibly terminated correctly

<1> Suppress a request from the DMA request source of the channel to be forcibly terminated (stop operation of the on-chip peripheral I/O).
<2> Check that the DMA transfer request of the channel to be forcibly terminated is not held pending, by using the DTFRn.DFn bit. If a DMA transfer request is held pending, wait until execution of the pending request is completed.
<3> When it has been confirmed that the DMA request of the channel to be forcibly terminated is not held pending, clear the Enn bit to 0.
<4> Again, clear the Enn bit of the channel to be forcibly terminated.
If the target of transfer for the channel to be forcibly terminated (transfer source/destination) is the internal RAM, execute this operation once more.
<5> Copy the initial number of transfers of the channel to be forcibly terminated to a general-purpose register.
<6> Set the INITn bit of the channel to be forcibly terminated to 1.
<7> Read the value of the DBCn register of the channel to be forcibly terminated, and compare it with the value copied in <5>. If the two values do not match, repeat operations <6> and <7>.

Remarks 1. When the value of the DBCn register is read in <7>, the initial number of transfers is read if forced termination has been correctly completed. If not, the remaining number of transfers is read.
2. Note that method (b) may take a long time if the application frequently uses DMA transfer for a channel other than the DMA channel to be forcibly terminated.

(5) Procedure of temporarily stopping DMA transfer (clearing Enn bit)

Stop and resume the DMA transfer under execution using the following procedure.

<1> Suppress a transfer request from the DMA request source (stop the operation of the on-chip peripheral I/O).
<2> Check that the DMA transfer request is not held pending, by using the DFn bit (check if the DFn bit = 0).
If a request is pending, wait until execution of the pending DMA transfer request is completed.
<3> If it has been confirmed that no DMA transfer request is held pending, clear the Enn bit to 0 (this operation stops DMA transfer).
<4> Set the Enn bit to 1 to resume DMA transfer.
<5> Resume the operation of the DMA request source that has been stopped (start the operation of the on-chip peripheral I/O).

(6) Memory boundary

The operation is not guaranteed if the address of the transfer source or destination exceeds the area of the DMA target (external memory, internal RAM, or on-chip peripheral I/O) during DMA transfer.

(7) Transferring misaligned data

DMA transfer of misaligned data with a 16-bit bus width is not supported.
If an odd address is specified as the transfer source or destination, the least significant bit of the address is forcibly assumed to be 0.

(8) Bus arbitration for CPU

Because the DMA controller has a higher priority bus mastership than the CPU, a CPU access that takes place during DMA transfer is held pending until the DMA transfer cycle is completed and the bus is released to the CPU.

However, the CPU can access the external memory, on-chip peripheral I/O, and internal RAM to/from which DMA transfer is not being executed.

- The CPU can access the internal RAM when DMA transfer is being executed between the external memory and on-chip peripheral I/O.
- The CPU can access the internal RAM and on-chip peripheral I/O when DMA transfer is being executed between the external memory and external memory.

(9) Registers/bits that must not be rewritten during DMA operation

Set the following registers at the following timing when a DMA operation is not under execution.

[Registers]

- DSAnH, DSAnL, DDAnH, DDAnL, DBCn, and DADCn registers
- DTFRn.IFCn5 to DTFRn.IFCn0 bits

[Timing of setting]

- Period from after reset to start of the first DMA transfer
- Time after channel initialization to start of DMA transfer
- Period from after completion of DMA transfer (TCn bit = 1) to start of the next DMA transfer

(10) Be sure to set the following register bits to 0.

- Bits 14 to 10 of DSAnH register
- Bits 14 to 10 of DDAnH register
- Bits 15, 13 to 8, and 3 to 0 of DADCn register
- Bits 6 to 3 of DCHCn register

(11) DMA start factor

Do not start two or more DMA channels with the same start factor. If two or more channels are started with the same factor, a DMA channel with a lower priority may be acknowledged earlier than a DMA channel with a higher priority.

(12) Read values of DSAn and DDAn registers

Values in the middle of updating may be read from the DSAn and DDAn registers during DMA transfer (n = 0 to 3).

For example, if the DSAnH register and then the DSAnL register are read when the DMA transfer source address (DSAn register) is 0000FFFFH and the count direction is incremental (DADCn.SAD1 and DADCn.SAD0 bits = 00), the value of the DSAnL register differs as follows, depending on whether DMA transfer is executed immediately after the DSAnH register is read.

(a) If DMA transfer does not occur while DSAn register is read

<1> Read value of DSAnH register: DSAnH = 0000H
<2> Read value of DSAnL register: DSAnL = FFFFH

(b) If DMA transfer occurs while DSAn register is read

<1> Read value of DSAnH register: DSAnH = 0000H
<2> Occurrence of DMA transfer
<3> Incrementing DSAn register: DSAn = 00100000H
<4> Read value of DSAnL register: DSAnL = 0000H

CHAPTER 17 INTERRUPT/EXCEPTION PROCESSING FUNCTION

The μPD703229Y and 70F3229Y are provided with a dedicated interrupt controller (INTC) for interrupt servicing and can process a total of 47 interrupt requests.

An interrupt is an event that occurs independently of program execution, and an exception is an event whose occurrence is dependent on program execution.

The μPD703229Y and 70F3229Y can process interrupt requests from the on-chip peripheral hardware and external sources. Moreover, exception processing can be started by the TRAP instruction (software exception) or by generation of an exception event (i.e. fetching of an illegal opcode) (exception trap).

17.1 Features

○ Interrupts
- Non-maskable interrupts: 2 sources
- Maskable interrupts: External: 8, Internal: 37 sources
- 8 levels of programmable priorities (maskable interrupts)
- Multiple interrupt control according to priority
- Masks can be specified for each maskable interrupt request
- Noise elimination, edge detection, and valid edge specification for external interrupt request signals

○ Exceptions
- Software exceptions: 32 sources
- Exception trap: 2 sources (illegal opcode exception, debug trap)

Interrupt/exception sources are listed in Table 17-1.

Table 17-1. Interrupt Source List (1/2)

Type	Classification	Default Priority	Name	Trigger	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Reset	Interrupt	–	RESET	$\overline{\text{RESET}}$ pin input	Pin	0000H	00000000H	Undefined	–
				Internal reset by WDT2	WDT2				
Non-maskable	Interrupt	–	NMI	NMI pin valid edge input	Pin	0010H	00000010H	nextPC	–
		–	INTWDT2	WDT2 overflow	WDT2	0020H	00000020H	**Note 1**	–
Software exception	Exception	–	TRAP0n[Note 2]	TRAP instruction	–	004nH[Note 2]	00000040H	nextPC	–
		–	TRAP1n[Note 2]	TRAP instruction	–	005nH[Note 2]	00000050H	nextPC	–
Exception trap	Exception	–	ILGOP/ DBG0	Illegal opcode/ DBTRAP instruction	–	0060H	00000060H	nextPC	–
Maskable	Interrupt	0	INTLVI	Low-voltage detection interrupt	POC/LVI	0080H	00000080H	nextPC	LVIIC
		1	INTP0	External interrupt pin input edge detection (INTP0)	Pin	0090H	00000090H	nextPC	PIC0
		2	INTP1	External interrupt pin input edge detection (INTP1)	Pin	00A0H	000000A0H	nextPC	PIC1
		3	INTP2	External interrupt pin input edge detection (INTP2)	Pin	00B0H	000000B0H	nextPC	PIC2
		4	INTP3	External interrupt pin input edge detection (INTP3)	Pin	00C0H	000000C0H	nextPC	PIC3
		5	INTP4	External interrupt pin input edge detection (INTP4)	Pin	00D0H	000000D0H	nextPC	PIC4
		6	INTP5	External interrupt pin input edge detection (INTP5)	Pin	00E0H	000000E0H	nextPC	PIC5
		7	INTP6	External interrupt pin input edge detection (INTP6)	Pin	00F0H	000000F0H	nextPC	PIC6
		8	INTP7	External interrupt pin input edge detection (INTP7)	Pin	0100H	00000100H	nextPC	PIC7
		9	INTTQ0OV	TMQ0 overflow	TMQ0	0110H	00000110H	nextPC	TQ0OVIC
		10	INTTQ0CC0	TMQ0 capture 0/ compare 0 match	TMQ0	0120H	00000120H	nextPC	TQ0CCIC0
		11	INTTQ0CC1	TMQ0 capture 1/ compare 1 match	TMQ0	0130H	00000130H	nextPC	TQ0CCIC1
		12	INTTQ0CC2	TMQ0 capture 2/ compare 2 match	TMQ0	0140H	00000140H	nextPC	TQ0CCIC2
		13	INTTQ0CC3	TMQ0 capture 3/ compare 3 match	TMQ0	0150H	00000150H	nextPC	TQ0CCIC3
		14	INTTP0OV	TMP0 overflow	TMP0	0160H	00000160H	nextPC	TP0OVIC
		15	INTTP0CC0	TMP0 capture 0/ compare 0 match	TMP0	0170H	00000170H	nextPC	TP0CCIC0
		16	INTTP0CC1	TMP0 capture 1/ compare 1 match	TMP0	0180H	00000180H	nextPC	TP0CCIC1

★ **Notes 1.** For restoring from INTWDT2, refer to **17. 9 Cautions**.

2. n = 0 to FH

Table 17-1. Interrupt Source List (2/2)

Type	Classification	Default Priority	Name	Trigger	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Maskable	Interrupt	17	INTTP1OV	TMP1 overflow	TMP1	0190H	00000190H	nextPC	TP1OVIC
		18	INTTP1CC0	TMP1 capture 0/ compare 0 match	TMP1	01A0H	000001A0H	nextPC	TP1CCIC0
		19	INTTP1CC1	TMP1 capture 1/ compare 1 match	TMP1	01B0H	000001B0H	nextPC	TP1CCIC1
		20	INTTP2OV	TMP2 overflow	TMP2	01C0H	000001C0H	nextPC	TP2OVIC
		21	INTTP2CC0	TMP2 capture 0/ compare 0 match	TMP2	01D0H	000001D0H	nextPC	TP2CCIC0
		22	INTTP2CC1	TMP2 capture 1/ compare 1 match	TMP2	01E0H	000001E0H	nextPC	TP2CCIC1
		23	INTTP3OV	TMP3 overflow	TMP3	01F0H	000001F0H	nextPC	TP3OVIC
		24	INTTP3CC0	TMP3 capture 0/ compare 0 match	TMP3	0200H	00000200H	nextPC	TP3CCIC0
		25	INTTP3CC1	TMP3 capture 1/ compare 1 match	TMP3	0210H	00000210H	nextPC	TP3CCIC1
		26	INTTM0EQ0	TMM0 compare match	TMM0	0220H	00000220H	nextPC	TM0EQIC0
		27	INTCB0R/ INTIIC0	CSIB0 reception completion/ CSIB0 reception error/ IIC0 transfer completion	CSIB0/ IIC00	0230H	00000230H	nextPC	CB0RIC/ IICIC0
		28	INTCB0T	CSIB0 consecutive transmission write enable	CSIB0	0240H	00000240H	nextPC	CB0TIC
		29	INTCB1R	CSIB1 reception completion/ CSIB1 reception error	CSIB1	0250H	00000250H	nextPC	CB1RIC
		30	INTCB1T	CSIB1 consecutive transmission write enable	CSIB1	0260H	00000260H	nextPC	CB1TIC
		31	INTUA0R	UARTA0 reception completion/ UARTA0 reception error	UARTA0	0270H	00000270H	nextPC	UA0RIC
		32	INTUA0T	UARTA0 consecutive transmission enable	UARTA0	0280H	00000280H	nextPC	UA0TIC
		33	INTUA1R	UARTA1 reception completion/ UARTA1 reception error	UARTA1	0290H	00000290H	nextPC	UA1RIC
		34	INTUA1T	UARTA1 consecutive transmission enable	UARTA1	02A0H	000002A0H	nextPC	UA1TIC
		35	INTAD	A/D conversion completion	A/D	02B0H	000002B0H	nextPC	ADIC
		36	INTKR	Key return interrupt	KR	0300H	00000300H	nextPC	KRIC
		37	INTWTI	Watch timer interval	WT	0310H	00000310H	nextPC	WTIIC
		38	INTWT	Watch timer reference time	WT	0320H	00000320H	nextPC	WTIC
		39	INTUA2R	UARTA2 reception completion/ UARTA2 reception error	UARTA2	03B0H	000003B0H	nextPC	UA2RIC
		40	INTUA2T	UARTA2 consecutive transmission enable	UARTA2	03C0H	000003C0H	nextPC	UA2TIC
		41	INTDMA0	DMA0 transfer completion	DMA	0410H	00000410H	nextPC	DMAIC0
		42	INTDMA1	DMA1 transfer completion	DMA	0420H	00000420H	nextPC	DMAIC1
		43	INTDMA2	DMA2 transfer completion	DMA	0430H	00000430H	nextPC	DMAIC2
		44	INTDMA3	DMA3 transfer completion	DMA	0440H	00000440H	nextPC	DMAIC3

Remarks 1. Default Priority: The priority order when two or more maskable interrupt request signals occur at the same time. The highest priority is 0.
The priority of the non-maskable interrupts is INTWDT2 > NMI.

Restored PC: The value of the program counter (PC) saved to EIPC or FEPC when interrupt servicing is started. Note, however, that the restored PC when a non-maskable or maskable interrupt is acknowledged while one of the following instructions is being executed does not become the nextPC (if an interrupt is acknowledged during instruction execution, execution stops, and then resumes after the interrupt servicing has finished).

- Load instructions (SLD.B, SLD.BU, SLD.H, SLD.HU, SLD.W)
- Division instructions (DIV, DIVH, DIVU, DIVHU)
- PREPARE, DISPOSE instructions (only if an interrupt is generated before the stack pointer is updated)

nextPC: The PC value that starts the processing following interrupt/exception processing.

2. The execution address of the illegal instruction when an illegal opcode exception occurs is calculated by (Restored PC – 4).

17.2 Non-Maskable Interrupts

A non-maskable interrupt request signal is acknowledged unconditionally, even when interrupts are in the interrupt disabled (DI) status. An NMI is not subject to priority control and takes precedence over all the other interrupt request signals.

This product has the following two non-maskable interrupt request signals.

- NMI pin input (NMI)
- Non-maskable interrupt request signal generated by overflow of watchdog timer (INTWDT2)

The valid edge of the NMI pin can be selected from four types: “rising edge”, “falling edge”, “both edges”, and “no edge detection”[Note].

★ **Note** The NMI pin function is enabled by setting the PMC0, INTF0, and INTR0 registers. Note that the NMI pin is not enabled if the PMC0, INTF0, and INTR0 registers are not set. How to enable the NMI pin is shown below.

<1> Set the PMC0.PMC02 bit to 1 and enable the alternate function (NMI).

<2> Set any value to the INTF0.INTF02 and INTR0.INTR02 bits and set the desired edge detection.

The non-maskable interrupt request signal generated by overflow of watchdog timer 2 (INTWDT2) functions when the WDTM2.WDM21 and WDTM2.WDM20 bits are set to “01”.

If two or more non-maskable interrupt request signals occur at the same time, the interrupt with the higher priority is serviced, as follows (the interrupt request signal with the lower priority is ignored).

INTWDT2 > NMI

If a new NMI or INTWDT2 request signal is issued while a NMI is being serviced, it is serviced as follows.

(1) If new NMI request signal is issued while NMI is being serviced

The new NMI request signal is held pending, regardless of the value of the PSW.NP bit. The pending NMI request signal is acknowledged after the NMI currently under execution has been serviced (after the RETI instruction has been executed).

(2) If new INTWDT2 request signal is issued while NMI is being serviced

The INTWDT2 request signal is held pending if the NP bit remains set (1) while the NMI is being serviced. The pending INTWDT2 request signal is acknowledged after the NMI currently under execution has been serviced (after the RETI instruction has been executed).

If the NP bit is cleared (0) while the NMI is being serviced, the newly generated INTWDT2 request signal is executed (the NMI servicing is stopped).

★ **Caution For the non-maskable interrupt servicing by the non-maskable interrupt request signal (INTWDT2), refer to 17.9 Cautions.**

Figure 17-1. Non-Maskable Interrupt Request Signal Acknowledgment Operation (1/2)

(a) NMI and INTWDT2 request signals generated at the same time

Figure 17-1. Non-Maskable Interrupt Request Signal Acknowledgment Operation (2/2)

(b) Non-maskable interrupt request signal generated during non-maskable interrupt servicing

Non-maskable interrupt being serviced	Non-maskable interrupt request signal generated during non-maskable interrupt servicing	
	NMI	INTWDT2
NMI	• NMI request generated during NMI servicing 	• INTWDT2 request generated during NMI servicing (NP = 1 retained before INTWDT2 request) • INTWDT2 request generated during NMI servicing (NP = 0 set before INTWDT2 request) • INTWDT2 request generated during NMI servicing (NP = 0 set after INTWDT2 request)
INTWDT2	• NMI request generated during INTWDT2 servicing 	• INTWDT2 request generated during INTWDT2 servicing

17.2.1 Operation

If a non-maskable interrupt request signal is generated by NMI input, the CPU performs the following processing, and transfers control to the handler routine.

<1> Saves the restored PC to FEPC.
<2> Saves the current PSW to FEPSW.
<3> Writes exception code 0010H to the higher halfword (FECC) of ECR.
<4> Sets the PSW.NP and PSW.ID bits to 1 and clears the EP bit to 0.
<5> Sets the handler address (00000010H, 00000020H) corresponding to the non-maskable interrupt to the PC, and transfers control.

The servicing configuration of a non-maskable interrupt is shown in Figure 17-2.

Figure 17-2. Servicing Configuration of Non-Maskable Interrupt

17.2.2 Restore

(1) NMI pin input

Execution is restored from the NMI servicing by the RETI instruction.

When the RETI instruction is executed, the CPU performs the following processing, and transfers control to the address of the restored PC.

<1> Loads the restored PC and PSW from FEPC and FEPSW, respectively, because the PSW.EP bit is 0 and the PSW.NP bit is 1.

<2> Transfers control back to the address of the restored PC and PSW.

Figure 17-3 illustrates how the RETI instruction is processed.

Figure 17-3. RETI Instruction Processing

Caution When the EP and NP bits are changed by the LDSR instruction during non-maskable interrupt servicing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set the EP bit back to 0 and NP bit back to 1 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

(2) INTWDT2 signal

★ For the non-maskable interrupt servicing by the non-maskable interrupt request signal (INTWDT2), refer to **17.9 Cautions**.

17.2.3 NP flag

The NP flag is a status flag that indicates that non-maskable interrupt servicing is under execution.

This flag is set when a non-maskable interrupt request signal has been acknowledged, and masks non-maskable interrupt requests to prohibit multiple interrupts from being acknowledged.

After reset: 00000020H

	31 – 8	7	6	5	4	3	2	1	0
PSW	0	NP	EP	ID	SAT	CY	OV	S	Z

NP	Non-maskable interrupt servicing status
0	No non-maskable interrupt servicing
1	Non-maskable interrupt currently being serviced

17.3 Maskable Interrupts

Maskable interrupt request signals can be masked by interrupt control registers. The μPD703229Y and 70F3229Y have 45 maskable interrupt sources.

If two or more maskable interrupt request signals are generated at the same time, they are acknowledged according to the default priority. In addition to the default priority, eight levels of priorities can be specified by using the interrupt control registers (programmable priority control).

When an interrupt request signal has been acknowledged, the acknowledgment of other maskable interrupt request signals is disabled and the interrupt disabled (DI) status is set.

When the EI instruction is executed in an interrupt service routine, the interrupt enabled (EI) status is set, which enables servicing of interrupts having a higher priority than the interrupt request signal in progress (specified by the interrupt control register). Note that only interrupts with a higher priority will have this capability; interrupts with the same priority level cannot be nested.

To enable multiple interrupts, however, save EIPC and EIPSW to memory or registers before executing the EI instruction, and execute the DI instruction before the RETI instruction to restore the original values of EIPC and EIPSW.

17.3.1 Operation

If a maskable interrupt occurs by INT input, the CPU performs the following processing, and transfers control to a handler routine.

<1> Saves the restored PC to EIPC.
<2> Saves the current PSW to EIPSW.
<3> Writes an exception code to the lower halfword of ECR (EICC).
<4> Sets the PSW.ID bit to 1 and clears the PSW.EP bit to 0.
<5> Sets the handler address corresponding to each interrupt to the PC, and transfers control.

The maskable interrupt request signal masked by INTC and the maskable interrupt request signal generated while another interrupt is being serviced (while PSW.NP bit = 1 or ID bit = 1) are held pending inside INTC. In this case, servicing a new maskable interrupt is started in accordance with the priority of the pending maskable interrupt request signal if either the maskable interrupt is unmasked or the NP and ID bits are cleared to 0 by using the RETI or LDSR instruction.

How maskable interrupts are serviced is illustrated below.

Figure 17-4. Maskable Interrupt Servicing

Note For the ISPR register, refer to **17.3.6 In-service priority register (ISPR)**.

17.3.2 Restore

Recovery from maskable interrupt servicing is carried out by the RETI instruction.

When the RETI instruction is executed, the CPU performs the following steps, and transfers control to the address of the restored PC.

<1> Loads the restored PC and PSW from EIPC and EIPSW because the PSW.EP bit is 0 and the PSW.NP bit is 0.

<2> Transfers control to the address of the restored PC and PSW.

Figure 17-5 illustrates the processing of the RETI instruction.

Figure 17-5. RETI Instruction Processing

Note For the ISPR register, refer to **17.3.6 In-service priority register (ISPR)**.

Caution **When the EP and NP bits are changed by the LDSR instruction during maskable interrupt servicing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set the EP bit back to 0 and the NP bit back to 0 using the LDSR instruction immediately before the RETI instruction.**

Remark The solid line shows the CPU processing flow.

17.3.3 Priorities of maskable interrupts

The INTC performs multiple interrupt servicing in which an interrupt is acknowledged while another interrupt is being serviced. Multiple interrupts can be controlled by priority levels.

There are two types of priority level control: control based on the default priority levels, and control based on the programmable priority levels that are specified by the interrupt priority level specification bit (xxPRn) of the interrupt control register (xxICn). When two or more interrupts having the same priority level specified by the xxPRn bit are generated at the same time, interrupt request signals are serviced in order depending on the priority level allocated to each interrupt request type (default priority level) beforehand. For more information, refer to **Table 17-1 Interrupt Source List**. The programmable priority control customizes interrupt request signals into eight levels by setting the priority level specification flag.

Note that when an interrupt request signal is acknowledged, the PSW.ID flag is automatically set to 1. Therefore, when multiple interrupts are to be used, clear the ID flag to 0 beforehand (for example, by placing the EI instruction in the interrupt service program) to set the interrupt enable mode.

Remark xx: Identification name of each peripheral unit (refer to **Table 17-2 Interrupt Control Registers (xxICn)**)

n: Peripheral unit number (refer to **Table 17-2 Interrupt Control Registers (xxICn)**)

Figure 17-6. Example of Processing in Which Another Interrupt Request Signal Is Issued While an Interrupt Is Being Serviced (1/2)

Caution To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Remarks 1. a to u in the figure are the temporary names of interrupt request signals shown for the sake of explanation.

2. The default priority in the figure indicates the relative priority between two interrupt request signals.

Figure 17-6. Example of Processing in Which Another Interrupt Request Signal Is Issued While an Interrupt Is Being Serviced (2/2)

Caution To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Figure 17-7. Example of Servicing Interrupt Request Signals Simultaneously Generated

Caution To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Remarks 1. a to c in the figure are the temporary names of interrupt request signals shown for the sake of explanation.

2. The default priority in the figure indicates the relative priority between two interrupt request signals.

17.3.4 Interrupt control register (xxICn)

An interrupt control register is assigned to each interrupt request signal (maskable interrupt) and sets the control conditions for each maskable interrupt request.

This register can be read or written in 8-bit or 1-bit units.

Reset input sets this register to 47H.

Caution Disable interrupts (DI) to read the xxIFn bit of the xxICn register. If the xxIFn bit is read while interrupts are enabled (EI), the correct value may not be read when acknowledging an interrupt and reading the bit conflict.

After reset: 47H R/W Address: FFFFF112H to FFFFF17CH

	<7>	<6>	5	4	3	2	1	0
xxICn	xxIFn	xxMKn	0	0	0	xxPRn2	xxPRn1	xxPRn0

xxIFn	Interrupt request flag[Note]
0	Interrupt request signal not issued
1	Interrupt request signal issued

xxMKn	Interrupt mask flag
0	Interrupt servicing enabled
1	Interrupt servicing disabled (pending)

xxPRn2	xxPRn1	xxPRn0	Interrupt priority specification bit
0	0	0	Specifies level 0 (highest).
0	0	1	Specifies level 1.
0	1	0	Specifies level 2.
0	1	1	Specifies level 3.
1	0	0	Specifies level 4.
1	0	1	Specifies level 5.
1	1	0	Specifies level 6.
1	1	1	Specifies level 7 (lowest).

Note The flag xxIFn is reset automatically by the hardware if an interrupt request signal is acknowledged.

Remark xx: Identification name of each peripheral unit (refer to **Table 17-2 Interrupt Control Registers (xxICn)**)

n: Peripheral unit number (refer to **Table 17-2 Interrupt Control Registers (xxICn)**)

The addresses and bits of the interrupt control registers are as follows.

Table 17-2. Interrupt Control Registers (xxICn) (1/2)

Address	Register	Bit							
		<7>	<6>	5	4	3	2	1	0
FFFFF110H	LVIIC	LVIIF	LVIMK	0	0	0	LVIPR2	LVIPR1	LVIPR0
FFFFF112H	PIC0	PIF0	PMK0	0	0	0	PPR02	PPR01	PPR00
FFFFF114H	PIC1	PIF1	PMK1	0	0	0	PPR12	PPR11	PPR10
FFFFF116H	PIC2	PIF2	PMK2	0	0	0	PPR22	PPR21	PPR20
FFFFF118H	PIC3	PIF3	PMK3	0	0	0	PPR32	PPR31	PPR30
FFFFF11AH	PIC4	PIF4	PMK4	0	0	0	PPR42	PPR41	PPR40
FFFFF11CH	PIC5	PIF5	PMK5	0	0	0	PPR52	PPR51	PPR50
FFFFF11EH	PIC6	PIF6	PMK6	0	0	0	PPR62	PPR61	PPR60
FFFFF120H	PIC7	PIF7	PMK7	0	0	0	PPR72	PPR71	PPR70
FFFFF122H	TQ0OVIC	TQ0OVIF	TQ0OVMK	0	0	0	TQ0OVPR02	TQ0OVPR01	TQ0OVPR00
FFFFF124H	TQ0CCIC0	TQ0CCIF0	TQ0CCMK0	0	0	0	TQ0CCPR02	TQ0CCPR01	TQ0CCPR00
FFFFF126H	TQ0CCIC1	TQ0CCIF1	TQ0CCMK1	0	0	0	TQ0CCPR12	TQ0CCPR11	TQ0CCPR10
FFFFF128H	TQ0CCIC2	TQ0CCIF2	TQ0CCMK2	0	0	0	TQ0CCPR22	TQ0CCPR21	TQ0CCPR20
FFFFF12AH	TQ0CCIC3	TQ0CCIF3	TQ0CCMK3	0	0	0	TQ0CCPR32	TQ0CCPR31	TQ0CCPR30
FFFFF12CH	TP0OVIC	TP0OVIF	TP0OVMK	0	0	0	TP0OVPR2	TP0OVPR1	TP0OVPR0
FFFFF12EH	TP0CCIC0	TP0CCIF0	TP0CCMK0	0	0	0	TP0CCPR02	TP0CCPR01	TP0CCPR00
FFFFF130H	TP0CCIC1	TP0CCIF1	TP0CCMK1	0	0	0	TP0CCPR12	TP0CCPR11	TP0CCPR10
FFFFF132H	TP1OVIC	TP1OVIF	TP1OVMK	0	0	0	TP1OVPR2	TP1OVPR1	TP1OVPR0
FFFFF134H	TP1CCIC0	TP1CCIF0	TP1CCMK0	0	0	0	TP1CCPR02	TP1CCPR01	TP1CCPR00
FFFFF136H	TP1CCIC1	TP1CCIF1	TP1CCMK1	0	0	0	TP1CCPR12	TP1CCPR11	TP1CCPR10
FFFFF138H	TP2OVIC	TP2OVIF	TP2OVMK	0	0	0	TP2OVPR2	TP2OVPR1	TP2OVPR0
FFFFF13AH	TP2CCIC0	TP2CCIF0	TP2CCMK0	0	0	0	TP2CCPR02	TP2CCPR01	TP2CCPR00
FFFFF13CH	TP2CCIC1	TP2CCIF1	TP2CCMK1	0	0	0	TP2CCPR12	TP2CCPR11	TP2CCPR10
FFFFF13EH	TP3OVIC	TP3OVIF	TP3OVMK	0	0	0	TP3OVPR2	TP3OVPR1	TP3OVPR0
FFFFF140H	TP3CCIC0	TP3CCIF0	TP3CCMK0	0	0	0	TP3CCPR02	TP3CCPR01	TP3CCPR00
FFFFF142H	TP3CCIC1	TP3CCIF1	TP3CCMK1	0	0	0	TP3CCPR12	TP3CCPR11	TP3CCPR10
FFFFF144H	TM0EQIC0	TM0EQIF0	TM0EQMK0	0	0	0	TM0EQPR02	TM0EQPR01	TM0EQPR00
FFFFF146H	CB0RIC/ IICIC0	CB0RIF/ IICIF0	CB0RMK/ IICMK0	0	0	0	CB0RPR2/ IICPR02	CB0RPR1/ IICPR01	CB0RPR0/ IICPR00
FFFFF148H	CB0TIC	CB0TIF	CB0TMK	0	0	0	CB0TPR2	CB0TPR1	CB0TPR0
FFFFF14AH	CB1RIC	CB1RIF	CB1RMK	0	0	0	CB1RPR2	CB1RPR1	CB1RPR0
FFFFF14CH	CB1TIC	CB1TIF	CB1TMK	0	0	0	CB1TPR2	CB1TPR1	CB1TPR0
FFFFF14EH	UA0RIC	UA0RIF	UA0RMK	0	0	0	UA0RPR2	UA0RPR1	UA0RPR0
FFFFF150H	UA0TIC	UA0TIF	UA0TMK	0	0	0	UA0TPR2	UA0TPR1	UA0TPR0
FFFFF152H	UA1RIC	UA1RIF	UA1RMK	0	0	0	UA1RPR2	UA1RPR1	UA1RPR0
FFFFF154H	UA1TIC	UA1TIF	UA1TMK	0	0	0	UA1TPR2	UA1TPR1	UA1TPR0
FFFFF156H	ADIC	ADIF	ADMK	0	0	0	ADPR2	ADPR11	ADPR0
FFFFF160H	KRIC	KRIF	KRMK	0	0	0	KRPR2	KRPR1	KRPR0
FFFFF162H	WTIIC	WTIIF	WTIMK	0	0	0	WTIPR2	WTIPR1	WTIPR0
FFFFF164H	WTIC	WTIF	WTMK	0	0	0	WTPR2	WTPR1	WTPR0
FFFFF176H	UA2RIC	UA2RIF	UA2RMK	0	0	0	UA2RPR2	UA2RPR1	UA2RPR0
FFFFF178H	UA2TIC	UA2TIF	UA2TMK	0	0	0	UA2TPR2	UA2TPR1	UA2TPR0

Table 17-2. Interrupt Control Registers (xxICn) (2/2)

Address	Register	Bit							
		<7>	<6>	5	4	3	2	1	0
FFFFF182H	DMAIC0	DMAIF0	DMAMK0	0	0	0	DMAPR02	DMAPR01	DMAPR00
FFFFF184H	DMAIC1	DMAIF1	DMAMK1	0	0	0	DMAPR12	DMAPR11	DMAPR10
FFFFF186H	DMAIC2	DMAIF2	DMAMK2	0	0	0	DMAPR22	DMAPR21	DMAPR20
FFFFF188H	DMAIC3	DMAIF3	DMAMK3	0	0	0	DMAPR32	DMAPR31	DMAPR30

17.3.5 Interrupt mask registers 0 to 3 (IMR0 to IMR3)

These registers set the interrupt mask state for the maskable interrupts. The xxMKn bit of the IMR0 to IMR3 registers is equivalent to the xxMKn bit of the xxICn register.

The IMRm register can be read or written in 16-bit units (m = 0 to 3).

If the higher 8 bits of the IMRm register are used as an IMRmH register and the lower 8 bits as an IMRmL register, these registers can be read or written in 8-bit or 1-bit units (m = 0 to 3).

Reset input sets these registers to FFFFH.

Caution The device file defines the xxMKn bit of the xxICn register as a reserved word. If a bit is manipulated using the name of xxMKn, the contents of the xxICn register, instead of the IMRm register, are rewritten (as a result, the contents of the IMRm register are also rewritten).

After reset: FFFFH R/W Address: IMR3 FFFFF106H, IMR3L FFFFF106H, IMR3H FFFFF107H

	15	14	13	12	11	10	9	8
IMR3 (IMR3H[Note])	1	1	1	DMAMK3	DMAMK2	DMAMK1	DMAMK0	1
	7	6	5	4	3	2	1	0
(IMR3L)	1	1	1	UA2TMK	UA2RMK	1	1	1

After reset: FFFFH R/W Address: IMR2 FFFFF104H, IMR2L FFFFF104H, IMR2H FFFFF105H

	15	14	13	12	11	10	9	8
IMR2 (IMR2H[Note])	1	1	1	1	1	WTMK	WTIMK	KRMK
	7	6	5	4	3	2	1	0
(IMR2L)	1	1	1	1	ADMK	UA1TMK	UA1RMK	UA0TMK

After reset: FFFFH R/W Address: IMR1 FFFFF102H, IMR1L FFFFF102H, IMR1H FFFFF103H

	15	14	13	12	11	10	9	8
IMR1 (IMR1H[Note])	UA0RMK	CB1TMK	CB1RMK	CB0TMK	CB0RMK/ IICMK0	TM0EQMK0	TP3CCMK1	TP3CCMK0
	7	6	5	4	3	2	1	0
(IMR1L)	TP3OVMK	TP2CCMK1	TP2CCMK0	TP2OVMK	TP1CCMK1	TP1CCMK0	TP1OVMK	TP0CCMK1

After reset: FFFFH R/W Address: IMR0 FFFFF100H, IMR0L FFFFF100H, IMR0H FFFFF101H

	15	14	13	12	11	10	9	8
IMR0 (IMR0H[Note])	TP0CCMK0	TP0OVMK	TQ0CCMK3	TQ0CCMK2	TQ0CCMK1	TQ0CCMK0	TQ0OVMK	PMK7
	7	6	5	4	3	2	1	0
(IMR0L)	PMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK

xxMKn	Interrupt mask flag setting
0	Interrupt servicing enabled
1	Interrupt servicing disabled

★ **Note** When bits 8 to 15 of the IMR0 to IMR3 registers are read/written in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the IMR0H to IMR3H registers.

★ **Caution Set bits 4 to 7 and 11 to 15 of the IMR2 register and bits 0 to 2, 5 to 8, and 13 to 15 of the IMR3 register to 1. If these settings are changed, operation is not guaranteed.**

Remark xx: Identification name of each peripheral unit (refer to **Table 17-2 Interrupt Control Registers (xxICn)**)

n: Peripheral unit number (refer to **Table 17-2 Interrupt Control Registers (xxICn)**)

17.3.6 In-service priority register (ISPR)

This register holds the priority level of the maskable interrupt currently acknowledged. When an interrupt request signal is acknowledged, the bit of this register corresponding to the priority level of that interrupt request signal is set to 1 and remains set while the interrupt is serviced.

When the RETI instruction is executed, the bit corresponding to the interrupt request signal having the highest priority is automatically reset to 0 by hardware. However, it is not reset to 0 when execution is returned from non-maskable interrupt servicing or exception processing.

This register is read-only in 8-bit or 1-bit units.

Reset input clears this register to 00H.

Caution If an interrupt is acknowledged while the ISPR register is being read in the interrupt enabled (EI) status, the value of the ISPR register after the bits of the register have been set by acknowledging the interrupt may be read. To accurately read the value of the ISPR register before an interrupt is acknowledged, read the register while interrupts are disabled (DI status).

After reset: 00H R Address: FFFFF1FAH

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
ISPR	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0

ISPRn	Priority of interrupt currently acknowledged
0	Interrupt request signal with priority n not acknowledged
1	Interrupt request signal with priority n acknowledged

Remark n = 0 to 7 (priority level)

17.3.7 ID flag

This flag controls the maskable interrupt's operating state, and stores control information regarding enabling or disabling of interrupt request signals. An interrupt disable flag (ID) is assigned to the PSW.

Reset input sets this flag to 00000020H.

After reset: 00000020H

	31	8	7	6	5	4	3	2	1	0
PSW	0		NP	EP	ID	SAT	CY	OV	S	Z

ID	Maskable interrupt servicing specification[Note]
0	Maskable interrupt request signal acknowledgment enabled
1	Maskable interrupt request signal acknowledgment disabled

Note Interrupt disable flag (ID) function

This bit is set to 1 by the DI instruction and cleared to 0 by the EI instruction. Its value is also modified by the RETI instruction or LDSR instruction when referencing the PSW.

Non-maskable interrupt request signals and exceptions are acknowledged regardless of this flag. When a maskable interrupt request signal is acknowledged, the ID flag is automatically set to 1 by hardware.

The interrupt request signal generated during the acknowledgment disabled period (ID flag = 1) is acknowledged when the xxICn.xxIFn bit is set to 1, and the ID flag is cleared to 0.

17.4 External Interrupt Request Input Pins (NMI and INTP0 to INTP7)

17.4.1 Noise elimination

(1) Eliminating noise on NMI pin

The NMI pin has an internal noise eliminator that uses analog delay. Therefore, the input level of the NMI pin is not detected as an edge unless it is maintained for a specific time or longer. Therefore, an edge is detected after specific time.

The NMI pin can be used to release the STOP mode. In the STOP mode, noise is not eliminated by using the system clock because the internal system clock is stopped.

(2) Eliminating noise on INTP0 to INTP7 pins

The INTP0 to INTP7 pins have an internal noise eliminator that uses analog delay. Therefore, the input level of the NMI pin is not detected as an edge unless it is maintained for a specific time or longer. Therefore, an edge is detected after specific time.

17.4.2 Edge detection

The valid edge of each of the NMI and INTP0 to INTP7 pins can be selected from the following four.

- Rising edge
- Falling edge
- Both rising and falling edges
- No edge detected

The edge of the NMI pin is not detected after reset. Therefore, the interrupt request signal is not acknowledged unless a valid edge is enabled by using the INTF0 and INTR0 registers (the NMI pin functions as a normal port pin).

To use the P00/NMI pin as an I/O port pin, specify the valid edge of the NMI pin to be “No edge detected”.

(1) External interrupt falling, rising edge specification register 0 (INTF0, INTR0)

The INTF0 and INTR0 registers are 8-bit registers that specify detection of the falling and rising edges of the NMI pin via bit 2 and the external interrupt pins (INTP0 to INTP3) via bits 3 to 6.

These registers can be read or written in 8-bit or 1-bit units.

Reset input clears these registers to 00H.

Caution When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear the INTF0n and INTR0n bits to 00, and then set the port mode.

After reset: 00H R/W Address: INTF0 FFFFFC00H, INTR0 FFFFFC20H

	7	6	5	4	3	2	1	0
INTF0	0	INTF06	INTF05	INTF04	INTF03	INTF02	0	0
		INTP3	INTP2	INTP1	INTP0	NMI		

	7	6	5	4	3	2	1	0
INTR0	0	INTR06	INTR05	INTR04	INTR03	INTR02	0	0
		INTP3	INTP2	INTP1	INTP0	NMI		

Remark For how to specify a valid edge, see **Table 17-3**.

Table 17-3. Valid Edge Specification

INTF0n	INTR0n	Valid Edge Specification (n = 2 to 6)
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Caution Be sure to clear the INTF0n and INTR0n bits to 00 when these registers are not used as the NMI or INTP0 to INTP3 pins.

Remark n = 2: Control of NMI pin
n = 3 to 6: Control of INTP0 to INTP3 pins

(2) External interrupt falling, rising edge specification register 3L (INTF3L, INTR3L)

The INTF3 and INTR3 registers are 8-bit registers that specify detection of the falling and rising edges of the external interrupt pin (INTP7).

These registers can be read or written in 8-bit or 1-bit units.

Reset input clears these registers to 00H.

Cautions 1. When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear the INTF31 and INTR31 bits to 00, and then set the port mode.

2. The INTP7 pin and RXDA0 pin are alternate-function pins. When using the pin as the RXDA0 pin, disable edge detection for the INTP7 alternate-function pin (clear the INTF3L.INTF31 bit and the INTR3L.INTR31 bit to 0). When using the pin as the INTP7 pin, stop UARTA0 reception (clear the UA0CTL0.UA0RXE bit to 0).

After reset: 00H R/W Address: INTF3L FFFFFC06H, INTR3L FFFFFC26H

	7	6	5	4	3	2	1	0
INTF3L	0	0	0	0	0	0	INTF31	0
							INTP7	

	7	6	5	4	3	2	1	0
INTR3L	0	0	0	0	0	0	INTR31	0
							INTP7	

Remark For how to specify a valid edge, see **Table 17-4**.

Table 17-4. Valid Edge Specification

INTF31	INTR31	Valid Edge Specification
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Caution Be sure to clear the INTF31 and INTR31 bits to 00 when these registers are not used as INTP7 pin.

(3) External interrupt falling, rising edge specification register 9H (INTF9H, INTR9H)

The INTF9H and INTR9H registers are 8-bit registers that specify detection of the falling and rising edges of the external interrupt pins (INTP4 to INTP6).

These registers can be read or written in 8-bit or 1-bit units.

Reset input clears these registers to 00H.

Caution When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear the INTF9n and INTR9n bits to 0, and then set the port mode.

After reset: 00H R/W Address: INTF9H FFFFFC13H, INTR9H FFFFFC33H

	15	14	13	12	11	10	9	8
INTF9H	INTF915	INTF914	INTF913	0	0	0	0	0
	INTP6	INTP5	INTP4					

	15	14	13	12	11	10	9	8
INTR9H	INTR915	INTR914	INTR913	0	0	0	0	0
	INTP6	INTP5	INTP4					

Remark For how to specify a valid edge, see **Table 17-5**.

Table 17-5. Valid Edge Specification

INTF9n	INTR9n	Valid Edge Specification (n = 13 to 15)
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Caution Be sure to clear the INTF9n and INTR9n bits to 00 when these registers are not used as the INTP4 to INTP6 pins.

Remark n = 13 to 15: Control of INTP4 to INTP6 pins

(4) Noise elimination control register

Digital noise elimination can be selected for the INTP3 pin. The noise elimination settings are performed with the NFC register.

When digital noise elimination is selected, the sampling clock for digital sampling can be selected from among $f_{XX}/64$, $f_{XX}/128$, $f_{XX}/256$, $f_{XX}/512$, $f_{XX}/1{,}024$, and f_{XT}. Sampling is performed 2 or 3 times.

Even when digital noise elimination is selected, using f_{XT} as the sampling clock makes it possible to use the INTP3 interrupt request signal to release the IDLE1, IDLE2, and STOP modes.

This register can be read or written in 8-bit or 1-bit units.

Caution After the sampling clock has been changed, time consisting of "sampling clock × number of times of sampling" is required until the digital noise eliminator is initialized. Therefore, an interrupt request signal may be generated if the valid edge of INTP3 is input during the time of "sampling clock × number of times of sampling" after the sampling clock has been changed. Therefore, be careful about the following points when using the interrupt and DMA functions.

- **When the interrupt function is used, clear the interrupt request flag (PIC3.PIF3 bit to 0) after the clock equivalent to the "sampling clock × number of times of sampling" has elapsed, and then enable interrupts.**
- **When the DMA function is used (started by INTP3), enable DMA after the clock equivalent to the "sampling clock × number of times of sampling" has elapsed.**

After reset: 00H R/W Address: FFFFF318H

	7	6	5	4	3	2	1	0
NFC	NFEN	NFSTS	0	0	0	NFC2	NFC1	NFC0

NFEN	INTP3 pin noise elimination settings
0	Analog noise elimination (60 ns (TYP.))
1	Digital noise elimination

NFSTS	Setting of number of times of sampling for eliminating digital noise
0	Number of times of sampling = 3 times
1	Number of times of sampling = 2 times

NFSTS	NFC2	NFC1	NFC0	Sampling clock	Width of noise accurately eliminated
					f_{XX} = 20 MHz
0	0	0	0	$f_{XX}/64$	6.40 μs
0	0	0	1	$f_{XX}/128$	12.80 μs
0	0	1	0	$f_{XX}/256$	25.60 μs
0	0	1	1	$f_{XX}/512$	51.20 μs
0	1	0	0	$f_{XX}/1,024$	102.40 μs
0	1	0	1	f_{XT}	61.04 μs
1	0	0	0	$f_{XX}/64$	3.20 μs
1	0	0	1	$f_{XX}/128$	6.40 μs
1	0	1	0	$f_{XX}/256$	12.80 μs
1	0	1	1	$f_{XX}/512$	25.60 μs
1	1	0	0	$f_{XX}/1,024$	51.20 μs
1	1	0	1	f_{XT}	30.52 μs
Other than above				Setting prohibited	

Remark In the case of noise with a width smaller than sampling clock × (number of times of sampling – 1), an interrupt request signal is generated if noise synchronized with the sampling clock is input.

17.5 Software Exception

A software exception is generated when the CPU executes the TRAP instruction, and can always be acknowledged.

17.5.1 Operation

If a software exception occurs, the CPU performs the following processing, and transfers control to the handler routine.

<1> Saves the restored PC to EIPC.
<2> Saves the current PSW to EIPSW.
<3> Writes an exception code to the lower 16 bits (EICC) of ECR (interrupt source).
<4> Sets the PSW.EP and PSW.ID bits to 1.
<5> Sets the handler address (00000040H or 00000050H) corresponding to the software exception to the PC, and transfers control.

Figure 17-8 illustrates the processing of a software exception.

Figure 17-8. Software Exception Processing

Note TRAP instruction format: TRAP vector (the vector is a value from 00H to 1FH.)

The handler address is determined by the TRAP instruction's operand (vector). If the vector is 00H to 0FH, it becomes 00000040H, and if the vector is 10H to 1FH, it becomes 00000050H.

17.5.2 Restore

Recovery from software exception processing is carried out by the RETI instruction.

By executing the RETI instruction, the CPU carries out the following processing and shifts control to the restored PC's address.

<1> Loads the restored PC and PSW from EIPC and EIPSW because the PSW.EP bit is 1.
<2> Transfers control to the address of the restored PC and PSW.

Figure 17-9 illustrates the processing of the RETI instruction.

Figure 17-9. RETI Instruction Processing

Caution When the EP and NP bits are changed by the LDSR instruction during software exception processing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set the EP bit back to 1 and the NP bit back to 0 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

17.5.3 EP flag

The EP flag is a status flag used to indicate that exception processing is in progress. It is set when an exception occurs.

After reset: 00000020H

	31	8	7	6	5	4	3	2	1	0
PSW	0		NP	**EP**	ID	SAT	CY	OV	S	Z

EP	Exception processing status
0	Exception processing not in progress.
1	Exception processing in progress.

17.6 Exception Trap

An exception trap is an interrupt that is requested when the illegal execution of an instruction takes place. In the μPD703229Y and 70F3229Y, an illegal opcode exception (ILGOP: Illegal Opcode Trap) is considered as an exception trap.

17.6.1 Illegal opcode definition

The illegal instruction has an opcode (bits 10 to 5) of 111111B, a sub-opcode (bits 26 to 23) of 0111B to 1111B, and a sub-opcode (bit 16) of 0B. An exception trap is generated when an instruction applicable to this illegal instruction is executed.

×: Arbitrary

Caution Since it is possible to assign this instruction to an illegal opcode in the future, it is recommended that it not be used.

(1) Operation

If an exception trap occurs, the CPU performs the following processing, and transfers control to the handler routine.

<1> Saves the restored PC to DBPC.
<2> Saves the current PSW to DBPSW.
<3> Sets the PSW.NP, PSW.EP, and PSW.ID bits to 1.
<4> Sets the handler address (00000060H) corresponding to the exception trap to the PC, and transfers control.

Figure 17-10 illustrates the processing of the exception trap.

Figure 17-10. Exception Trap Processing

(2) Restore

Recovery from an exception trap is carried out by the DBRET instruction. By executing the DBRET instruction, the CPU carries out the following processing and controls the address of the restored PC.

<1> Loads the restored PC and PSW from DBPC and DBPSW.
<2> Transfers control to the address indicated by the restored PC and PSW.

Figure 17-11 illustrates the restore processing from an exception trap.

Figure 17-11. Restore Processing from Exception Trap

17.6.2 Debug trap

A debug trap is an exception that is generated when the DBTRAP instruction is executed and is always acknowledged.

Upon occurrence of a debug trap, the CPU performs the following processing.

(1) Operation

<1> Saves restored PC to DBPC.
<2> Saves current PSW to DBPSW.
<3> Sets the PSW.NP, PSW.EP, and PSW.ID bits to 1.
<4> Sets handler address (00000060H) for debug trap to PC and transfers control.

Figure 17-12 shows the debug trap processing format.

Figure 17-12. Debug Trap Processing Format

(2) Restore

Recovery from a debug trap is carried out by the DBRET instruction.

By executing the DBRET instruction, the CPU performs the following steps and shifts control to the restored PC's address.

<1> The restored PC and PSW are read from DBPC and DBPSW.

<2> Control is transferred to the fetched address of the restored PC and PSW.

★ **Caution DBPC and DBPSW can be accessed after the DBTRAP instruction is executed and before the DBRET instruction is executed.**

Table 17-13 shows the processing for restoring from a debug trap.

Figure 17-13. Processing for Restoring from Debug Trap

17.7 Interrupt Response Time of CPU

Except the following cases, the interrupt response time of the CPU is 4 clocks minimum. To input interrupt request signals successively, input the next interrupt request signal at least 4 clocks after the preceding interrupt.

- In IDLE1/IDLE2/STOP mode
- When the external bus is accessed
- When interrupt request non-sample instructions are successively executed (refer to **17.8 Periods in Which Interrupts Are Not Acknowledged by CPU**.)
- When an interrupt control register is accessed

★ **Figure 17-14. Pipeline Operation at Interrupt Request Signal Acknowledgment (Outline)**

(1) Minimum interrupt response time

(2) Maximum interrupt response time

Remark INT1 to INT4: Interrupt acknowledgment processing
IFX: Invalid instruction fetch
IDX: Invalid instruction decode

Interrupt response time (internal system clock)			Condition
	Internal interrupt	External interrupt	
Minimum	4	4 + Analog delay time	The following cases are exceptions. • In IDLE1/IDLE2/STOP mode • External bus access • Two or more interrupt request non-sample instructions are executed in succession • Access to peripheral I/O register
Maximum	6	6 + Analog delay time	

17.8 Periods in Which Interrupts Are Not Acknowledged by CPU

An interrupt is acknowledged by the CPU while an instruction is being executed. However, no interrupt will be acknowledged between an interrupt request non-sample instruction and the next instruction (interrupt is held pending).

The interrupt request non-sample instructions are as follows.

- EI instruction
- DI instruction
- LDSR reg2, 0x5 instruction (for PSW)
- Store instruction for the PRCMD register, ISPR register, and PSC register
- Store or bit manipulation instructions for the following interrupt-related registers.
  Interrupt control registers (xxICn), interrupt mask registers 0 to 3 (IMR0 to IMR3)

★ 17.9 Cautions

Restoring from non-maskable interrupt servicing executed by the non-maskable interrupt request (INTWDT2) by using the RETI instruction is disabled. Execute the following software reset processing.

Figure 17-15. Software Reset Processing

CHAPTER 18 KEY INTERRUPT FUNCTION

18.1 Function

A key interrupt request signal (INTKR) can be generated by inputting a falling edge to the eight key input pins (KR0 to KR7) by setting the key return mode register (KRM).

★ **Caution If a low level is input to any of the KR0 to KR7 pins, the INTKR signal is not generated even if the falling edge of another pin is input.**

Table 18-1. Assignment of Key Return Detection Pins

Flag	Pin Description
KRM0	Controls KR0 signal in 1-bit units
KRM1	Controls KR1 signal in 1-bit units
KRM2	Controls KR2 signal in 1-bit units
KRM3	Controls KR3 signal in 1-bit units
KRM4	Controls KR4 signal in 1-bit units
KRM5	Controls KR5 signal in 1-bit units
KRM6	Controls KR6 signal in 1-bit units
KRM7	Controls KR7 signal in 1-bit units

Figure 18-1. Block Diagram of Key Return

18.2 Control Register

(1) Key return mode register (KRM)

The KRM register controls the KR0 to KR7 signals using the KRM0 to KRM7 bits.

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF300H

	7	6	5	4	3	2	1	0
KRM	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	Control of key return mode
0	Does not detect key return signal
1	Detects key return signal

Caution If the KRM register is changed, an interrupt request signal (INTKR) may be generated. To prevent this, change the KRM register after disabling interrupts (DI), and then enable interrupts (EI) after clearing the interrupt request flag (KRIC.KRIF bit) to 0.

Remark For the alternate-function pin settings, refer to **Table 4-16 Using Port Pin as Alternate-Function Pin**.

CHAPTER 19 STANDBY FUNCTION

19.1 Overview

The power consumption of the system can be effectively reduced by using the standby modes in combination and selecting the appropriate mode for the application. The available standby modes are listed in Table 19-1.

Table 19-1. Standby Modes

Mode	Functional Outline
HALT mode	Mode to stop only the operating clock of the CPU
IDLE1 mode	Mode to stop all the operations of the internal circuits except the oscillator, PLL operation[Note], and flash memory
IDLE2 mode	Mode to stop all the internal operations of the chip except the oscillator
STOP mode	Mode to stop all the internal operations of the chip except the subclock oscillator
Subclock operation mode	Mode to use the subclock as the internal system clock
Sub-IDLE mode	Mode to stop all the operations of the internal circuits except the oscillator, PLL operation[Note], and flash memory, in the subclock operation mode

Note The PLL retains the previous operation status.

Figure 19-1. Status Transition (1/2)

Notes 1. IDLE2 mode only

2. Non-maskable interrupt request signal (NMI signal or INTWDT2 signal), unmasked external interrupt request signal, or unmasked internal maskable interrupt request signal operable in IDLE1/IDLE2 mode

3. Non-maskable interrupt request signal, unmasked external interrupt request signal, or unmasked internal maskable interrupt request

4. Reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, low-voltage detector (LVI), or clock monitor (CLM)

5. Non-maskable interrupt request signal (NMI signal or INTWDT2 signal), unmasked external interrupt request signal, or unmasked internal maskable interrupt request signal operable in STOP mode

6. Reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, or low-voltage detector (LVI)

★ **Figure 19-1. Status Transition (2/2)**

Notes 1. Reset input by $\overline{\text{RESET}}$ pin input, WDT2RES signal, low-voltage detector (LVI), or clock monitor (CLM)

2. Non-maskable interrupt request signal (NMI pin input), unmasked external interrupt request signal (INTP0 to INTP6 pin input), or unmasked internal interrupt request signal from peripheral functions operable in sub-IDLE mode

19.2 Control Registers

(1) Power save control register (PSC)

The PSC register is an 8-bit register that controls the standby function. The STP bit of this register is used to specify the STOP mode. This register is a special register that can be written only by the special sequence combinations (see **3.4.7 Special registers**).

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF1FEH

	7	<6>	<5>	<4>	3	2	<1>	0
PSC	0	NMI1M	NMI0M	INTM	0	0	STP	0

NMI1M	Standby mode release control upon occurrence of INTWDT2 signal
0	Standby mode release by INTWDT2 signal enabled
1	Standby mode release by INTWDT2 signal disabled

NMI0M	Standby mode release control by NMI pin input
0	Standby mode release by NMI pin input enabled
1	Standby mode release by NMI pin input disabled

INTM	Standby mode release control via maskable interrupt request signal
0	Standby mode release by maskable interrupt request signal enabled
1	Standby mode release by maskable interrupt request signal disabled

STP	Standby mode setting
0	Normal mode
1	Standby mode

Cautions 1. Before setting the IDLE1, IDLE2, STOP, or sub-IDLE mode, set the PSMR.PSM1 and PSMR.PSM0 bits and then set the STP bit.

2. Settings of the NMI1M, NMI0M, and INTM bits are invalid when HALT mode is released.

(2) Power save mode register (PSMR)

The PSMR register is an 8-bit register that controls the operation status in the power save mode and the clock operation.

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF820H

	7	6	5	4	3	2	<1>	<0>
PSMR	0	0	0	0	0	0	PSM1	PSM0

PSM1	PSM0	Specification of operation in software standby mode
0	0	IDLE1 mode
0	1	STOP mode
1	0	IDLE2 mode
1	1	STOP mode

Cautions 1. Be sure to clear bits 2 to 7 to 0.

2. The PSM0 and PSM1 bits are valid only when the PSC.STP bit is 1.

Remark IDLE1: In this mode, all operations except the oscillator operation and some other circuits (flash memory and PLL) are stopped.
After the IDLE1 mode is released, the normal operation mode is restored without needing to secure the oscillation stabilization time, like the HALT mode.

IDLE2: In this mode, all operations except the oscillator operation are stopped.
After the IDLE2 mode is released, the normal operation mode is restored following the lapse of the setup time specified by the OSTS register (flash memory and PLL).

STOP: In this mode, all operations except the subclock oscillator operation are stopped.
After the STOP mode is released, the normal operation mode is restored following the lapse of the oscillation stabilization time specified by the OSTS register.

(3) Oscillation stabilization time select register (OSTS)

The wait time until the oscillation stabilizes after the STOP mode is released or the wait time until the on-chip flash memory stabilizes after the IDLE2 mode is released is controlled by the OSTS register.

The OSTS register can be read or written in 8-bit units.

Reset input sets this register to 06H.

After reset: 06H R/W Address: FFFFF6C0H

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

OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time/setup time[Note]	fx	
				4 MHz	5 MHz
0	0	0	$2^{10}/f_x$	0.256 ms	0.205 ms
0	0	1	$2^{11}/f_x$	0.512 ms	0.410 ms
0	1	0	$2^{12}/f_x$	1.024 ms	0.819 ms
0	1	1	$2^{13}/f_x$	2.048 ms	1.638 ms
1	0	0	$2^{14}/f_x$	4.096 ms	3.277 ms
1	0	1	$2^{15}/f_x$	8.192 ms	6.554 ms
1	1	0	$2^{16}/f_x$	16.38 ms	13.107 ms
1	1	1	Setting prohibited		

Note The oscillation stabilization time and setup time are required when the STOP mode and IDLE2 mode are released, respectively.

Cautions 1. The wait time following release of the STOP mode does not include the time until the clock oscillation starts ("a" in the figure below) following release of the STOP mode, regardless of whether the STOP mode is released by reset input or the occurrence of an interrupt request signal.

2. Be sure to clear bits 3 to 7 to 0.

3. The oscillation stabilization time following reset release is $2^{16}/f_x$ (because the initial value of the OSTS register = 06H).

Remark fx = Main clock oscillation frequency

19.3 HALT Mode

19.3.1 Setting and operation status

The HALT mode is set when a dedicated instruction (HALT) is executed in the normal operation mode.

In the HALT mode, the clock oscillator continues operating. Only clock supply to the CPU is stopped; clock supply to the other on-chip peripheral functions continues.

As a result, program execution is stopped, and the internal RAM retains the contents before the HALT mode was set. The on-chip peripheral functions that are independent of instruction processing by the CPU continue operating.

Table 19-3 shows the operation status in the HALT mode.

The average power consumption of the system can be reduced by using the HALT mode in combination with the normal operation mode for intermittent operation.

Cautions 1. Insert five or more NOP instructions after the HALT instruction.

2. When the HALT instruction is executed while an unmasked interrupt request signal is being held pending, the status shifts to HALT mode, but the HALT mode is then released immediately by the pending interrupt request.

19.3.2 Releasing HALT mode

The HALT mode is released by a non-maskable interrupt request signal (NMI pin input or INTWDT2 signal), unmasked external interrupt request signal (INTP0 to INTP7 pin input), unmasked internal interrupt request signal from a peripheral function operable in the HALT mode, or reset signal (reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, low-voltage detector (LVI), or clock monitor (CLM)).

After the HALT mode has been released, the normal operation mode is restored.

(1) Releasing HALT mode by non-maskable interrupt request signal or unmasked maskable interrupt request signal

The HALT mode is released by a non-maskable interrupt request signal or an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request signal. If the HALT mode is set in an interrupt servicing routine, however, an interrupt request signal that is issued later is serviced as follows.

(a) If an interrupt request signal with a priority lower than that of the interrupt request signal currently being serviced is issued, only the HALT mode is released, and that interrupt request signal is not acknowledged. The interrupt request signal itself is retained.

(b) If an interrupt request signal with a priority higher than that of the interrupt request signal currently being serviced is issued (including a non-maskable interrupt request signal), the HALT mode is released and that interrupt request signal is acknowledged.

Table 19-2. Operation After Releasing HALT Mode by Interrupt Request Signal

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request signal	Execution branches to the handler address.	
Maskable interrupt request signal	Execution branches to the handler address or the next instruction is executed.	The next instruction is executed.

(2) Releasing HALT mode by reset

The same operation as the normal reset operation is performed.

Table 19-3. Operation Status in HALT Mode

Setting of HALT Mode / Item		Operation Status	
		When Subclock Is Not Used	When Subclock Is Used
Main clock oscillator		Oscillation enabled	
Subclock oscillator		–	Oscillation enabled
Ring-OSC generator		Oscillation enabled	
PLL		Operable	
CPU		Stops operation	
DMA		Operable	
Interrupt controller		Operable	
ROM correction		Stops operation	
Timer P (TMP0 to TMP3)		Operable	
Timer Q (TMQ0)		Operable	
Timer M (TMM0)		Operable when a clock other than f_{XT} is selected as the count clock	Operable
Watch timer		Operable when f_X (divided BRG) is selected as the count clock	Operable
Watchdog timer 2		Operable	
Serial interface	CSIB0, CSIB1	Operable	
	I^2C00	Operable	
	UARTA0 to UARTA2	Operable	
A/D converter		Operable	
Key interrupt function (KR)		Operable	
External bus interface		Refer to **CHAPTER 5 BUS CONTROL FUNCTION**.	
Port function		Retains status before HALT mode was set.	
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the HALT mode was set.	

19.4 IDLE1 Mode

19.4.1 Setting and operation status

The IDLE1 mode is set by clearing the PSMR.PSM1 and PSMR.PSM0 bits to 00 and setting the PSC.STP bit to 1 in the normal operation mode.

In the IDLE1 mode, the clock oscillator, PLL operation, and flash memory continue operating but clock supply to the CPU and other on-chip peripheral functions stops.

As a result, program execution stops and the contents of the internal RAM before the IDLE1 mode was set are retained. The CPU and other on-chip peripheral functions stop operating. However, the on-chip peripheral functions that can operate with the subclock or an external clock continue operating.

Table 19-5 shows the operation status in the IDLE1 mode.

The IDLE1 mode can reduce the power consumption more than the HALT mode because it stops the operation of the on-chip peripheral functions. The main clock oscillator, PLL operation, and flash memory do not stop, so the normal operation mode can be restored without waiting for the oscillation stabilization time after the IDLE1 mode has been released, in the same manner as when the HALT mode is released.

Caution Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the IDLE1 mode.

19.4.2 Releasing IDLE1 mode

The IDLE1 mode is released by a non-maskable interrupt request signal (NMI pin input or INTWDT2 signal), unmasked external interrupt request signal (INTP0 to INTP7 pin input), unmasked internal interrupt request signal from a peripheral function operable in the IDLE1 mode, or reset signal (reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, low-voltage detector (LVI), or clock monitor (CLM)).

After the IDLE1 mode has been released, the normal operation mode is restored.

(1) Releasing IDLE1 mode by non-maskable interrupt request signal or unmasked maskable interrupt request signal

The IDLE1 mode is released by a non-maskable interrupt request signal or an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request signal. If the IDLE1 mode is set in an interrupt servicing routine, however, an interrupt request signal that is issued later is processed as follows.

Caution An interrupt request signal that is disabled by setting the PSC.NMI1M, PSC.NMI0M, and PSC.INTM bits to 1 becomes invalid and IDLE1 mode is not released.

(a) If an interrupt request signal with a priority lower than that of the interrupt request signal currently being serviced is issued, only the IDLE1 mode is released, and that interrupt request signal is not acknowledged. The interrupt request signal itself is retained.

(b) If an interrupt request signal with a priority higher than that of the interrupt request signal currently being serviced is issued (including a non-maskable interrupt request signal), the IDLE1 mode is released and that interrupt request signal is acknowledged.

Table 19-4. Operation After Releasing IDLE1 Mode by Interrupt Request Signal

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request signal	Execution branches to the handler address.	
Maskable interrupt request signal	Execution branches to the handler address or the next instruction is executed.	The next instruction is executed.

(2) Releasing IDLE1 mode by reset

The same operation as the normal reset operation is performed.

Table 19-5. Operation Status in IDLE1 Mode

Setting of IDLE1 Mode / Item		Operation Status	
		When Subclock Is Not Used	When Subclock Is Used
Main clock oscillator		Oscillation enabled	
Subclock oscillator		–	Oscillation enabled
Ring-OSC generator		Oscillation enabled	
PLL		Operable	
CPU		Stops operation	
DMA		Stops operation	
Interrupt controller		Stops operation (but standby mode release is enabled)	
ROM correction		Stops operation	
Timer P (TMP0 to TMP3)		Stops operation	Operable when f_{XT} is selected as the count clock
Timer Q (TMQ0)		Stops operation	
Timer M (TMM0)		Operable when $f_R/8$ is selected as the count clock	Operable when $f_R/8$ or f_{XT} is selected as the count clock
Watch timer		Operable when f_X (divided BRG) is selected as the count clock	Operable
Watchdog timer 2		Operable	
Serial interface	CSIB0, CSIB1	Operable when $\overline{\text{SCKBn}}$ input clock is selected as the count clock (n = 0, 1)	
	I^2C00	Stops operation	
	UARTA0 to UARTA2	Stops operation (but UARTA0 is operable when the ASCKA0 input clock is selected)	
A/D converter		Retains operation (conversion result retained)[Note]	
Key interrupt function (KR)		Operable	
External bus interface		Refer to **CHAPTER 5 BUS CONTROL FUNCTION**.	
Port function		Retains status before IDLE1 mode was set.	
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the IDLE1 mode was set.	

★ **Note** To realize low power consumption, stop the A/D converter before shifting to the IDLE1 mode.

19.5 IDLE2 Mode

19.5.1 Setting and operation status

The IDLE2 mode is set by setting the PSMR.PSM1 and PSMR.PSM0 bits to 10 and setting the PSC.STP bit to 1 in the normal operation mode.

In the IDLE2 mode, the clock oscillator continues operation but clock supply to the CPU, PLL, flash memory, and other on-chip peripheral functions stops.

As a result, program execution stops and the contents of the internal RAM before the IDLE2 mode was set are retained. The CPU, PLL, and other on-chip peripheral functions stop operating. However, the on-chip peripheral functions that can operate with the subclock or an external clock continue operating.

Table 19-7 shows the operation status in the IDLE2 mode.

The IDLE2 mode can reduce the power consumption more than the IDLE1 mode because it stops the operations of the on-chip peripheral functions, PLL, and flash memory. However, because the PLL and flash memory are stopped, a setup time for the PLL and flash memory is required when IDLE2 mode is released.

Caution Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the IDLE2 mode.

19.5.2 Releasing IDLE2 mode

The IDLE2 mode is released by a non-maskable interrupt request signal (NMI pin input or INTWDT2 signal), unmasked external interrupt request signal (INTP0 to INTP7 pin input), unmasked internal interrupt request signal from the peripheral functions operable in the IDLE2 mode, or reset signal (reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, low-voltage detector (LVI), or clock monitor (CLM)). The PLL returns to the operation status before the IDLE2 mode was set.

After the IDLE2 mode has been released, the normal operation mode is restored.

(1) Releasing IDLE2 mode by non-maskable interrupt request signal or unmasked maskable interrupt request signal

The IDLE2 mode is released by a non-maskable interrupt request signal or an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request signal. If the IDLE2 mode is set in an interrupt servicing routine, however, an interrupt request signal that is issued later is processed as follows.

Caution The interrupt request signal that is disabled by setting the PSC.NMI1M, PSC.NMI0M, and PSC.INTM bits to 1 becomes invalid and IDLE2 mode is not released.

(a) If an interrupt request signal with a priority lower than that of the interrupt request signal currently being serviced is issued, only the IDLE2 mode is released, and that interrupt request signal is not acknowledged. The interrupt request signal itself is retained.

(b) If an interrupt request signal with a priority higher than that of the interrupt request signal currently being serviced is issued (including a non-maskable interrupt request signal), the IDLE2 mode is released and that interrupt request signal is acknowledged.

Table 19-6. Operation After Releasing IDLE2 Mode by Interrupt Request Signal

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request signal	Execution branches to the handler address after securing the prescribed setup time.	
Maskable interrupt request signal	Execution branches to the handler address or the next instruction is executed after securing the prescribed setup time.	The next instruction is executed after securing the prescribed setup time.

(2) Releasing IDLE2 mode by reset

The same operation as the normal reset operation is performed.

Table 19-7. Operation Status in IDLE2 Mode

Setting of IDLE2 Mode / Item		Operation Status	
		When Subclock Is Not Used	When Subclock Is Used
Main clock oscillator		Oscillation enabled	
Subclock oscillator		–	Oscillation enabled
Ring-OSC generator		Oscillation enabled	
PLL		Stops operation	
CPU		Stops operation	
DMA		Stops operation	
Interrupt controller		Stops operation (but standby mode release is enabled)	
ROM correction		Stops operation	
Timer P (TMP0 to TMP3)		Stops operation	Operable when f_{XT} is selected as the count clock
Timer Q (TMQ0)		Stops operation	
Timer M (TMM0)		Operable when $f_R/8$ is selected as the count clock	Operable when $f_R/8$ or f_{XT} is selected as the count clock
Watch timer		Operable when f_X (divided BRG) is selected as the count clock	Operable
Watchdog timer 2		Operable	
Serial interface	CSIB0, CSIB1	Operable when $\overline{\text{SCKBn}}$ input clock is selected as the count clock (n = 0, 1)	
	I^2C00	Stops operation	
	UARTA0 to UARTA2	Stops operation (but UARTA0 is operable when the ASCKA0 input clock is selected)	
A/D converter		Retains operation (conversion result retained)[Note]	
Key interrupt function (KR)		Operable	
External bus interface		Refer to **CHAPTER 5 BUS CONTROL FUNCTION**.	
Port function		Retains status before IDLE2 mode was set.	
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the IDLE2 mode was set.	

★ **Note** To realize low power consumption, stop the A/D converter before shifting to the IDLE2 mode.

19.5.3 Securing setup time when releasing IDLE2 mode

Secure the setup time for the ROM (flash memory) after releasing the IDLE2 mode because the operation of the blocks other than the main clock oscillator stops after the IDLE2 mode is set.

(1) Releasing IDLE2 mode by non-maskable interrupt request signal or unmasked maskable interrupt request signal

Secure the specified setup time by setting the OSTS register.

When the releasing source of IDLE2 mode is generated, the dedicated internal timer starts counting according to the OSTS register setting. When it overflows, the normal operation mode is restored.

(2) Releasing IDLE2 mode by reset ($\overline{\text{RESET}}$ pin input, WDT2RES generation)

The same operation as the normal reset operation is performed.

The oscillation stabilization time is the initial value of the OSTS register, 2^{16}/fx.

19.6 STOP Mode

19.6.1 Setting and operation status

The STOP mode is set by setting the PSMR.PSM1 and PSMR.PSM0 bits to 01 and setting the PSC.STP bit to 1 in the normal operation mode.

In the STOP mode, the subclock oscillator continues operation but the main clock oscillator stops. Clock supply to the CPU and the on-chip peripheral functions is stopped.

As a result, program execution stops, and the contents of the internal RAM before the STOP mode was set are retained. The CPU and other on-chip peripheral functions stop operating. However, the on-chip peripheral functions that can operate with the subclock oscillator or an external clock continue operating.

Table 19-9 shows the operation status in the STOP mode.

Because the STOP mode stops operation of the main clock oscillator, it reduces the power consumption to a level lower than the IDLE2 mode. If the subclock oscillator and external clock are not used, the power consumption can be minimized with only leakage current flowing.

Caution Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the STOP mode.

19.6.2 Releasing STOP mode

The STOP mode is released by a non-maskable interrupt request signal (NMI pin input or INTWDT2 signal), unmasked external interrupt request signal (INTP0 to INTP7 pin input), unmasked internal interrupt request signal from the peripheral functions operable in the STOP mode, or reset signal (reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, or low-voltage detector (LVI)).

After the STOP mode has been released, the normal operation mode is restored after the oscillation stabilization time has been secured.

Caution The interrupt request signal that is disabled by setting the PSC.NMI1M, PSC.NMI0M, and PSC.INTM bits to 1 becomes invalid and STOP mode is not released.

(1) Releasing STOP mode by non-maskable interrupt request signal or unmasked maskable interrupt request signal

The STOP mode is released by a non-maskable interrupt request signal or an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request signal. If the STOP mode is set in an interrupt servicing routine, however, an interrupt request signal that is issued later is serviced as follows.

(a) If an interrupt request signal with a priority lower than that of the interrupt request signal currently being serviced is issued, only the STOP mode is released, and that interrupt request signal is not acknowledged. The interrupt request signal itself is retained.

(b) If an interrupt request signal with a priority higher than that of the interrupt request signal currently being serviced is issued (including a non-maskable interrupt request signal), the STOP mode is released and that interrupt request signal is acknowledged.

Table 19-8. Operation After Releasing STOP Mode by Interrupt Request Signal

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request signal	Execution branches to the handler address after securing the oscillation stabilization time.	
Maskable interrupt request signal	Execution branches to the handler address or the next instruction is executed after securing the oscillation stabilization time.	The next instruction is executed after securing the oscillation stabilization time.

(2) Releasing STOP mode by reset

The same operation as the normal reset operation is performed.

Table 19-9. Operation Status in STOP Mode

Setting of STOP Mode / Item		Operation Status	
		When Subclock Is Not Used	When Subclock Is Used
Main clock oscillator		Stops oscillation	
Subclock oscillator		–	Oscillation enabled
Ring-OSC generator		Oscillation enabled	
PLL		Stops operation	
CPU		Stops operation	
DMA		Stops operation	
Interrupt controller		Stops operation (but standby mode release is enabled)	
ROM correction		Stops operation	
Timer P (TMP0 to TMP3)		Stops operation	Operable when f_{XT} is selected as the count clock
Timer Q (TMQ0)		Stops operation	
Timer M (TMM0)		Operable when $f_R/8$ is selected as the count clock	Operable when $f_R/8$ or f_{XT} is selected as the count clock
Watch timer		Stops operation	Operable when f_{XT} is selected as the count clock
Watchdog timer 2		Operable when f_R is selected as the count clock	
Serial interface	CSIB0, CSIB1	Operable when $\overline{\text{SCKBn}}$ input clock is selected as the count clock (n = 0, 1)	
	I^2C00	Stops operation	
	UARTA0 to UARTA2	Stops operation (but UARTA0 is operable when the ASCKA0 input clock is selected)	
A/D converter		Stops operation (conversion result undefined)[Notes 1, 2]	
Key interrupt function (KR)		Operable	
External bus interface		Refer to **CHAPTER 5 BUS CONTROL FUNCTION**.	
Port function		Retains status before STOP mode was set.	
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the STOP mode was set.	

★ **Notes 1.** If the STOP mode is set while the A/D converter is operating, the A/D converter is automatically stopped and starts operating again after the STOP mode is released. However, in that case, the A/D conversion results up to the second conversion after the STOP mode is released are invalid (the third or later conversion results are valid). All the A/D conversion results before the STOP mode is set are invalid.

★ **2.** Even if the STOP mode is set while the A/D converter is operating, the power consumption is reduced equivalently to when the A/D converter is stopped before the STOP mode is set.

19.7 Securing Oscillation Stabilization Time

When the STOP mode is released, only the oscillation stabilization time set by the OSTS register elapses. If the STOP mode has been released by $\overline{\text{RESET}}$ pin input, however, the reset value of the OSTS register, 2^{16}/fx elapses.

The timer for counting the oscillation stabilization time is shared with watchdog timer 2, so the oscillation stabilization time equal to the overflow time of watchdog timer 2 elapses.

Figure 19-2 shows the operation performed when the STOP mode is released by an interrupt request signal.

Figure 19-2. Oscillation Stabilization Time

19.8 Subclock Operation Mode

19.8.1 Setting and operation status

The subclock operation mode is set by setting the PCC.CK3 bit to 1 in the normal operation mode.

When the subclock operation mode is set, the internal system clock is changed from the main clock to the subclock. Check whether the clock has been switched by using the PCC.CLS bit.

When the PCC.MCK bit is set to 1, the operation of the main clock oscillator is stopped. As a result, the system operates only with the subclock.

In the subclock operation mode, the power consumption can be reduced to a level lower than in the normal operation mode because the subclock is used as the internal system clock. In addition, the power consumption can be further reduced to the level of the STOP mode by stopping the operation of the main clock oscillator.

Table 19-10 shows the operation status in subclock operation mode.

Cautions 1. When manipulating the CK3 bit, do not change the set values of the PCC.CK2 to PCC.CK0 bits (using a bit manipulation instruction to manipulate the bit is recommended). For details of the PCC register, refer to 6.3 (1) Processor clock control register (PCC).

★ **2. If the following conditions are not satisfied, change the CK2 to CK0 bits so that the conditions are satisfied and set the subclock operation mode.**

Main clock (f_{XX}) > Subclock (f_{XT} = 32.768 kHz) × 4

19.8.2 Releasing subclock operation mode

The subclock operation mode is released by clearing the CK3 bit to 0 or by a reset signal (reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, low-voltage detector (LVI), or clock monitor (CLM)).

If the main clock is stopped (MCK bit = 1), set the MCK bit to 1, secure the oscillation stabilization time of the main clock by software, and clear the CK3 bit to 0.

The normal operation mode is restored when the subclock operation mode is released.

Caution When manipulating the CK3 bit, do not change the set values of the CK2 to CK0 bits (using a bit manipulation instruction to manipulate the bit is recommended).
For details of the PCC register, refer to 6.3 (1) Processor clock control register (PCC).

Table 19-10. Operation Status in Subclock Operation Mode

Setting of Subclock Operation Mode / Item		Operation Status	
		When Main Clock Is Oscillating	When Main Clock Is Stopped
Subclock oscillator		Oscillation enabled	
Ring-OSC generator		Oscillation enabled	
PLL		Operable	Stops operation
CPU		Operable	
DMA		Operable	
Interrupt controller		Operable	
ROM correction		Operable	
Timer P (TMP0 to TMP3)		Operable	Operable when f_{XT} is selected as the count clock
Timer Q (TMQ0)		Operable	Stops operation
Timer M (TMM0)		Operable	Operable when $f_R/8$ or f_{XT} is selected as the count clock
Watch timer		Operable	Operable when f_{XT} is selected as the count clock
Watchdog timer 2		Operable	Operable when f_R is selected as the count clock
Serial interface	CSIB0, CSIB1	Operable	Operable when $\overline{\text{SCKBn}}$ input clock is selected as the count clock (n = 0, 1)
	I^2C00	Operable	Stops operation
	UARTA0 to UARTA2	Operable	Stops operation (but UARTA0 is operable when the ASCKA0 input clock is selected)
A/D converter		Operable	Stops operation
Key interrupt function (KR)		Operable	
External bus interface		Refer to **CHAPTER 5 BUS CONTROL FUNCTION**.	
Port function		Settable	
Internal data		Settable	

★ **Caution When the CPU is operating on the subclock and main clock oscillation is stopped, accessing a register in which a wait occurs is disabled. If a wait is generated, it can be released only by reset (refer to 3.4.8 (2)).**

19.9 Sub-IDLE Mode

19.9.1 Setting and operation status

The sub-IDLE mode is set by setting the PSMR.PSM1 and PSMR.PSM0 bits to 10 and setting the PSC.STP bit to 1 in the subclock operation mode.

In this mode, the clock oscillator continues operation but clock supply to the CPU, flash memory, and the other on-chip peripheral functions is stopped.

As a result, program execution stops and the contents of the internal RAM before the sub-IDLE mode was set are retained. The CPU and the other on-chip peripheral functions are stopped. However, the on-chip peripheral functions that can operate with the subclock or an external clock continue operating.

Because the sub-IDLE mode stops operation of the CPU, flash memory, and other on-chip peripheral functions, it can reduce the power consumption more than the subclock operation mode.

If the sub-IDLE mode is set after the main clock has been stopped, the power consumption can be reduced to a level as low as that in the STOP mode.

Table 19-12 shows the operation status in the sub-IDLE mode.

19.9.2 Releasing sub-IDLE mode

The sub-IDLE mode is released by a non-maskable interrupt request signal (NMI pin input or INTWDT2 signal), unmasked external interrupt request signal (INTP0 to INTP7 pin input), unmasked internal interrupt request signal from the peripheral functions operable in the sub-IDLE mode, or reset signal (reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, low-voltage detector (LVI), or clock monitor (CLM)). The PLL returns to the operation status before the IDLE mode was set.

When the sub-IDLE mode is released by an interrupt request signal, the subclock operation mode is set. If it is released by reset input, the normal operation mode is restored.

(1) Releasing sub-IDLE mode by non-maskable interrupt request signal or unmasked maskable interrupt request signal

The sub-IDLE mode is released by a non-maskable interrupt request signal or an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request signal.

If the sub-IDLE mode is set in an interrupt servicing routine, however, an interrupt request signal that is issued later is serviced as follows.

Cautions 1. An interrupt request signal that is disabled by setting the PSC.NMI1M, PSC.NMI0M, and PSC.INTM bits to 1 becomes invalid and sub-IDLE mode is not released.

(a) If an interrupt request signal with a priority lower than that of the interrupt request signal currently being serviced is issued, only the sub-IDLE mode is released, and that interrupt request signal is not acknowledged. The interrupt request signal itself is retained.

(b) If an interrupt request signal with a priority higher than that of the interrupt request signal currently being serviced is issued (including a non-maskable interrupt request signal), the sub-IDLE mode is released and that interrupt request signal is acknowledged.

★ **2. When returning from the sub-IDLE mode, 12 cycles of the subclock (approximately 366 μs) are inserted between when the interrupt request signal that releases sub-IDLE mode is generated and when sub-IDLE mode is actually released.**

Table 19-11. Operation After Releasing Sub-IDLE Mode by Interrupt Request Signal

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request signal	Execution branches to the handler address.	
Maskable interrupt request signal	Execution branches to the handler address or the next instruction is executed.	The next instruction is executed.

(2) Releasing sub-IDLE mode by reset

The same operation as the normal reset operation is performed.

Table 19-12. Operation Status in Sub-IDLE Mode

Setting of Sub-IDLE Mode / Item		Operation Status	
		When Main Clock Is Oscillating	When Main Clock Is Stopped
Subclock oscillator		Oscillation enabled	
Ring-OSC generator		Oscillation enabled	
PLL		Operable	Stops operation[Note 1]
CPU		Stops operation	
DMA		Stops operation	
Interrupt controller		Stops operation (but standby mode release is enabled)	
ROM correction		Stops operation	
Timer P (TMP0 to TMP3)		Operable when f_{XT} is selected as the count clock	
Timer Q (TMQ0)		Stops operation	
Timer M (TMM0)		Operable when $f_R/8$ or f_{XT} is selected as the count clock	
Watch timer		Operable	Operable when f_{XT} is selected as the count clock
Watchdog timer 2		Operable	Operable when f_R is selected as the count clock
Serial interface	CSIB0, CSIB1	Operable when $\overline{\text{SCKBn}}$ input clock is selected as the count clock (n = 0, 1)	
	I^2C00	Stops operation	
	UARTA0 to UARTA2	Stops operation (but UARTA0 is operable when the ASCKA0 input clock is selected)	
A/D converter		Retains operation (conversion result retained)[Note 2]	
Key interrupt function (KR)		Operable	
External bus interface		Refer to **CHAPTER 5 BUS CONTROL FUNCTION** (same operation status as IDLE mode).	
Port function		Retains status before sub-IDLE mode was set.	
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the sub-IDLE mode was set.	

Notes 1. Be sure to stop the PLL (PLLON bit = 0) before stopping the main clock.

★ **2.** To realize low power consumption, stop the A/D converter before shifting to the sub-IDLE mode.

CHAPTER 20 RESET FUNCTIONS

20.1 Overview

The following reset functions are available.

- Reset by $\overline{\text{RESET}}$ pin input
- Reset by WDT2 overflow (WDT2RES)
- System reset by low-voltage detector (LVI)
- System reset by clock monitor (CLM)

20.2 Registers to Check Reset Source

★ **(1) Reset source flag register (RESF)**

The RESF register is a special register that can be written only by a combination of specific sequences (refer to **3.4.7 Special registers**).

The RESF register indicates the source from which a reset request is generated.

This register can be read or written in 8-bit or 1-bit units.

$\overline{\text{RESET}}$ pin input clears this register to 00H. The default value differs if the source of reset is other than the $\overline{\text{RESET}}$ pin signal.

After reset: 00H[Note] R/W Address: FFFFF888H

	7	6	5	4	3	2	1	0
RESF	0	0	0	WDT2RF	0	0	CLMRF	LVIRF

WDT2RF	Reset signal from WDT2
0	Not generated
1	Generated

CLMRF	Reset signal from CLM
0	Not generated
1	Generated

LVIRF	Reset signal from LVI
0	Not generated
1	Generated

Note When reset by the $\overline{\text{RESET}}$ pin, the value is 00H, when reset by the WDT2RES signal, the value is 10H, when reset by the low-voltage detector (LVI), the value is 01H, and when reset by the clock monitor (CLM), the value is 02H.

★ **Caution Only "0" can be written to each bit of this register. If writing "0" conflicts with setting the flag (occurrence of reset), setting the flag takes precedence.**

After reset is released by the RESF and CCLS registers, the type of internal reset and the CPU operating clock (Ring-OSC clock or main clock) can be identified in the processing routine.

The operations after reset is released are illustrated below.

Figure 20-1. Operations After Release of Reset

20.3 Operation

20.3.1 Reset operation by $\overline{\text{RESET}}$ pin

When a low level is input to the $\overline{\text{RESET}}$ pin, the system is reset, and each hardware unit is initialized.

When the level of the $\overline{\text{RESET}}$ pin is changed from low to high, the reset status is released.

If the reset status is released by $\overline{\text{RESET}}$ pin input, the oscillation stabilization time elapses (reset value of OSTS register: $2^{16}/f_X$) and then the CPU starts program execution.

Table 20-1. Hardware Status on $\overline{\text{RESET}}$ Pin Input

Item	During Reset	After Reset
Main clock oscillator (f_X)	Oscillation stops	Oscillation starts
Subclock oscillator (f_{XT})	Oscillation continues	
Ring-OSC generator	Oscillation stops	Oscillation starts
Peripheral clock (f_X to f_X/1,024)	Operation stops	Operation starts after securing oscillation stabilization time
Internal system clock (f_{CLK}), CPU clock (f_{CPU})	Operation stops	Operation starts after securing oscillation stabilization time (initialization to f_{XX}/8)
CPU	Initialization	Operation starts after securing oscillation stabilization time
WDT2	Operation stops	Operation starts
Internal RAM	Undefined if power-on reset or writing data to RAM (by CPU) and reset input conflict (data is damaged). Otherwise value immediately before reset input is retained[Note].	
I/O lines (port/alternate-function pin)	High impedance	
On-chip peripheral I/O registers	Initialized to specified status, OCDM register is set (01H).	
Other on-chip peripheral functions	Operation stops	Operation can be started after securing oscillation stabilization time

Note The firmware of the μPD703229Y and 70F3229Y uses part of the internal RAM after the internal system reset operation has been released because it supports a boot selection function. Therefore, the contents of some RAM areas (3FF7000H to 3FF7095H) are not retained on power-on reset.

Caution The μPD70F3229Y may enter on-chip debug mode after the reset status has been released, depending on the pin status. For details, refer to CHAPTER 4 PORT FUNCTIONS.

Figure 20-2. Timing of Reset Operation by $\overline{\text{RESET}}$ Pin Input

Figure 20-3. Timing of Power-on Reset Operation

20.3.2 Reset operation by WDT2RES signal

When watchdog timer 2 is set to the reset operation mode due to overflow, upon watchdog timer 2 overflow (WDT2RES signal occurrence), a system reset is performed and the hardware is initialized to the initial status.

Following watchdog timer 2 overflow, the reset status is entered and lasts the predetermined time (analog delay), and the reset status is then automatically released. Following reset release, the CPU starts program execution after securing the oscillation stabilization time (initial value of OSTS register: $2^{16}/f_X$) of the main clock oscillator.

The main clock oscillator is stopped during the reset period.

Table 20-2. Hardware Statuses During WDT2RES Signal Input

Item	During Reset	After Reset
Main clock oscillator (f_X)	Oscillation stops	Oscillation starts
Subclock oscillator (f_{XT})	Oscillation continues	
Ring-OSC generator	Oscillation stops	Oscillation starts
Peripheral clock (f_X to f_X/1,024)	Operation stops	Operation starts after securing oscillation stabilization time
Internal system clock (f_{CLK}), CPU clock (f_{CPU})	Operation stops	Operation starts after securing oscillation stabilization time (initialization to f_{XX}/8)
CPU	Initialization	Operation starts after securing oscillation stabilization time
WDT2	Operation stops	Operation starts
Internal RAM	Undefined if power-on reset or writing data to RAM (by CPU) and reset input conflict (data is damaged). Otherwise value immediately before reset input is retained[Note].	
I/O lines (port/alternate-function pin)	High impedance	
On-chip peripheral I/O register	Initialized to specified status, OCDM register retains its value.	
On-chip peripheral functions other than above	Operation stops	Operation can be started after securing oscillation stabilization time.

Note The firmware of the μPD703229Y and 70F3229Y uses part of the internal RAM after the internal system reset operation has been released because it supports a boot selection function. Therefore, the contents of some RAM areas (3FF7000H to 3FF7095H) are not retained on power-on reset.

Figure 20-4. Timing of Reset Operation by WDT2RES Pin Input

20.3.3 Operation by low-voltage detector

If the supply voltage falls below the voltage detected by the low-voltage detector when LVI operation is enabled, a system reset is performed (when the LVIM.LVIMD bit is set to 1), and the hardware is initialized to the initial status.

The reset status lasts from when a supply voltage drop has been detected until the supply voltage rises above the voltage detected by the low-voltage detector. Following reset release, the CPU starts program execution after securing the oscillation stabilization time (initial value of OSTS register: $2^{16}/f_X$) of the main clock oscillator.

The main clock oscillator is stopped during the reset period.

When the LVIMD bit = 0, an interrupt request signal (INTLVI) is generated if a low voltage is detected.

Table 20-3. Hardware Statuses During Reset Operation by Low-Voltage Detector

Item	During Reset	After Reset
Main clock oscillator (f_X)	Oscillation stops	Oscillation starts
Subclock oscillator (f_{XT})	Oscillation continues	
Ring-OSC generator	Oscillation stops	Oscillation starts
Peripheral clock (f_X to f_X/1,024)	Operation stops	Operation starts after securing oscillation stabilization time
Internal system clock (f_{CLK}), CPU clock (f_{CPU})	Operation stops	Operation starts after securing oscillation stabilization time (initialization to f_{XX}/8)
CPU	Initialization	Operation starts after securing oscillation stabilization time
WDT2	Operation stops	Operation starts
Internal RAM	Undefined if power-on reset or writing data to RAM (by CPU) and reset input conflict (data is damaged). Otherwise value immediately before reset input is retained[Note].	
I/O lines (port/alternate-function pin)	High impedance	
On-chip peripheral I/O register	Initialized to specified status, OCDM register retains its value.	
LVI	Operation stops	
On-chip peripheral functions other than above	Operation stops	Operation can be started after securing oscillation stabilization time.

Note The firmware of the μPD703229Y and 70F3229Y uses part of the internal RAM after the internal system reset operation has been released because it supports a boot selection function. Therefore, the contents of some RAM areas (3FF7000H to 3FF7095H) are not retained on power-on reset.

The operation setting method is shown below.

(1) Internal reset operation

<To start operation>

<1> Mask the INTLVI interrupt (LVIIC.LVIMK bit = 1).

<2> Set the detection voltage (V_{LVI}) by using the LVIS.LVIS0 bit.

<3> Set the LVIM.LVION bit to 1 (low-voltage detector operation enabled).

<4> Insert a wait cycle of 0.2 ms or more by software.

<5> Check that the LVIM.LVIF bit is 0 (supply voltage (V_{DD}) > detection voltage (V_{LVI})).
If the LVIF bit is 1, insert a wait cycle by software until the LVIF bit becomes 0.

<6> Set the LVIM.LVIMD bit to 1 (an internal reset signal ($\overline{\text{LVIRES}}$) is generated when supply voltage (V_{DD}) < detection voltage (V_{LVI})).

Cautions 1. When the LVIMD bit is set to 1, the LVIM and LVIS registers cannot be changed until a reset request other than the $\overline{\text{LVIRES}}$ signal is generated.

2. If setting with the LVIMK bit = 0 in <1>, an interrupt request signal (INTLVI) may be generated at the point of <3>.

<To stop operation>

The low-voltage detection operation cannot be stopped until a reset request other than the $\overline{\text{LVIRES}}$ signal is generated.

(2) Interrupt operation (INTLVI)

<To start operation>

<1> Mask the INTLVI interrupt (LVIMK bit = 1).

<2> Set the detection voltage (V_{LVI}) by using the LVIS0 bit.

<3> Set the LVION bit to 1 (low-voltage detector operation enabled).

<4> Insert a wait cycle of 0.2 ms or more by software.

<5> Check that the LVIF bit is 0 (supply voltage (V_{DD}) > detection voltage (V_{LVI})).
If the LVIF bit is 1, insert a wait cycle by software until the LVIF bit becomes 0.

<6> Clear the INTLVI interrupt request flag (LVIIC.LVIIF bit) to 0.

<7> Release the mask status of the INTLVI interrupt (LVIMK bit = 0).

Caution If setting with the LVIMK bit = 0 in <1>, the INTLVI signal may be generated at the point of <3>.

<To stop operation>

Clear the LVION bit to 0 (low-voltage detector operation disabled).

Figure 20-5. Timing of Reset Operation by Low-Voltage Detector

(3) Low-voltage detection register (LVIM)

The LVIM register is a special register that can be written only by the special sequence combinations (refer to **3.4.7 Special registers**).

The LVIM register is used to enable or disable low-voltage detection and select the operation mode.

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF890H

	<7>	6	5	4	3	2	<1>	<0>
LVIM	LVION	0	0	0	0	0	LVIMD	LVIF

LVION	Enables/disables low-voltage detection operation
0	Operation disabled.
1	Operation enabled.

LVIMD	Selects operation mode of low-voltage detection
0	Generate interrupt signal INTLVI when supply voltage < detected voltage.
1	Generate internal reset signal LVIRES when supply voltage < detected voltage.

LVIF	Low-voltage detection flag
0	When supply voltage > detected voltage or when operation is disabled
1	Supply voltage < detected voltage

Cautions 1. After setting the LVION bit to 1, wait for 0.1 ms (TYP.) before checking the voltage by the LVIF bit.

2. The value of the LVIF flag is output as the INTLVI signal when the LVION bit = 1 and LVIMD bit = 0.

3. Be sure to set bits 2 to 6 to 0.

(4) Low-voltage detection level select register (LVIS)

The LVIS register is used to select the low voltage level to be detected.

This register can be read or written in 8-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF891H

	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	0	0	0	LVIS0

LVIS0	Detected level
0	4.4 V ±0.2 V
1	4.2 V ±0.2 V

Caution Be sure to set bits 1 to 7 to 0.

(5) Internal RAM data status register (RAMS)

The RAMS register is a special register that can be written only by the special sequence combinations (refer to **3.4.7 Special registers**).

The RAMS register is used to make the data of the internal RAM valid or invalid.

This register can be read or written in 8-bit or 1-bit units.

The value of this register is retained by reset[Note] input.

★ **Note** The special sequence after reset is as follows.

Set conditions: Voltage detection lower than the detection level
Write to 1 in special sequence
Reset generation by WDT2
Reset generation during RAM access

Clear conditions: Write to 0 in special sequence

After reset: Retained R/W Address: FFFFF892H

	7	6	5	4	3	2	1	0
RAMS	0	0	0	0	0	0	0	RAMF

RAMF	Internal RAM data valid/invalid
0	Valid
1	Invalid

20.3.4 Clock monitor

(1) Function of clock monitor

The clock monitor samples the main clock using the internal Ring-OSC oscillator and generates a reset request signal when oscillation of the main clock is stopped.

After reset is released, the CPU operates on Ring-OSC.

Once the operation of the clock monitor has been enabled by the CLM.CLME bit, it can be stopped only by reset.

The clock monitor is automatically stopped under the following conditions.

- When the oscillation stabilization time is counted after the STOP mode has been released
- When the main clock is stopped (PCC.MCK bit = 1 when subclock operates and PCC.CLS bit = 0 when main clock operates)
- When the sampling clock is stopped (Ring-OSC)
- When the CPU operates on Ring-OSC

★ **Table 20-4. Operation Status of Clock Monitor (CLME Bit = 1, with Ring-OSC Operating)**

CPU Operation Clock	Operation Mode	Main Clock Status	Ring-OSC Clock Status	Clock Monitor Status
Main clock	Normal operation/ HALT mode	Oscillates	Oscillates[Note 1]	Operates[Note 2]
	IDLE1/IDLE2 mode	Oscillates	Oscillates[Note 1]	Operates[Note 2]
	STOP mode	Stops	Oscillates[Note 1]	Stops
Subclock (MCK bit = 0)	IDLE mode	Oscillates	Oscillates[Note 1]	Operates[Note 2]
Subclock (MCK bit = 1)	IDLE mode	Stops	Oscillates[Note 1]	Stops
Ring-OSC clock	–	Stops	Stops[Note 1]	Stops
During reset	–	Stops	Stops	Stops

Notes 1. Ring-OSC can be stopped by setting the RCM.RSTOP bit to 1 (μPD703229Y only).

2. The clock monitor is stopped when Ring-OSC is stopped.

(2) Clock monitor mode register (CLM)

The CLM register is a special register that can be written only by the special sequence combinations (refer to **3.4.7 Special registers**).

The CLM register is used to select the operation mode of the clock monitor.

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF870H

	7	6	5	4	3	2	1	<0>
CLM	0	0	0	0	0	0	0	CLME

CLME	Enables/disables operation of clock monitor
0	Disabled operation of clock monitor.
1	Enabled operation of clock monitor.

Caution Once the CLME bit has been set to 1, it can be cleared to 0 only by reset.

(3) Operation of clock monitor

(a) Operation when main clock oscillation is stopped (CLM.CLME bit = 1)

If oscillation of the main clock is stopped when the CLME bit = 1, an internal reset signal is generated at the following timing.

Figure 20-6. When Oscillation of Main Clock Is Stopped

(b) Operation in STOP mode and after STOP mode is released

If the STOP mode is set when the CLME bit = 1, the monitor operation is stopped in the STOP mode and while the oscillation stabilization time is being counted. The monitor operation is automatically started after the oscillation stabilization time has elapsed.

Figure 20-7. Operation in STOP Mode and After STOP Mode Is Released

(c) Operation when main clock is stopped

If the main clock is stopped by setting the PCC.MCK bit to 1 while the subclock is operating (PCC.CLS bit = 1), the monitor operation is stopped until the main clock operates (CLS bit = 0). The monitor operation is automatically started when the main clock starts operating.

Figure 20-8. Operation When Main Clock Is Stopped

(d) Operation when CPU operates on Ring-OSC clock (CCLS.CCLSF bit = 1)

The monitor operation is not started even if the CLME bit is set to 1 when the CCLSF bit signal is 1.

CHAPTER 21 REGULATOR

21.1 Outline

The μPD703229Y and 70F3229Y include a regulator to reduce the power consumption and noise.

This regulator supplies a stepped-down V_{DD} power supply voltage to the oscillator block and internal logic circuits (except the A/D converter and output buffer). The regulator output voltage is set to 2.5 V (TYP.).

Figure 21-1. Regulator

Remark $3.5\ V \leq V_{DD} \leq EV_{DD} \leq 5.5\ V$

$3.0\ V \leq BV_{DD} \leq 5.5\ V$

$BV_{DD} \leq V_{DD} \leq EV_{DD}$

21.2 Operation

The regulator of this product always operates in any mode (normal operation mode, HALT mode, IDLE1 mode, IDLE2 mode, STOP mode, or during reset).

Be sure to connect a capacitor (4.7 μF (recommended value)) to the REGC pin to stabilize the regulator output.

A diagram of the regulator pin connection method is shown below.

Figure 21-2. REGC Pin Connection

CHAPTER 22 ROM CORRECTION FUNCTION

22.1 Overview

The ROM correction function is used to replace part of the program in the mask ROM with the program of an external memory or the internal RAM.

By using this function, program bugs found in the mask ROM can be corrected.

Up to four addresses can be specified for correction.

Figure 22-1. Block Diagram of ROM Correction

Remark n = 0 to 3

22.2 Control Registers

(1) Correction address registers 0 to 3 (CORAD0 to CORAD3)

The CORAD0 to CORAD3 registers set the first address of the program to be corrected in the ROM.

The program can be corrected at up to four places because four CORADn registers are provided (n = 0 to 3).

Set the correction addresses within the range of 0000000H to 005FFFFH.

The CORADn register can be read or written in 32-bit units.

If the higher 16 bits of the CORADn register are used as the CORADnH register, and the lower 16 bits as the CORADnL register, these registers can be read or written in 16-bit units.

Reset input clears these registers to 00000000H.

After reset: 00000000H R/W Address: CORAD0 FFFFF840H,
CORAD0L FFFFF840H, CORAD0H FFFFF842H,
CORAD1 FFFFF844H,
CORAD1L FFFFF844H, CORAD1H FFFFF846H,
CORAD2 FFFFF848H,
CORAD2L FFFFF848H, CORAD2H FFFFF84AH,
CORAD3 FFFFF84CH,
CORAD3L FFFFF84CH, CORAD3H FFFFF84EH

CORADn (n = 0 to 3)

31	19	18	1	0
Fixed to 0	Note	Correction address		0

★ **Note** Set to 0.

(2) Correction control register (CORCN)

The CORCN register disables or enables the correction operation at the addresses set in the CORADn register (n = 0 to 3).

Each channel can be enabled or disabled by this register.

This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF880H

	7	6	5	4	3	2	1	0
CORCN	0	0	0	0	CORENn3	CORENn2	CORENn1	CORENn0

CORENn	Enables/disables correction operation
0	Disabled
1	Enabled

Remark n = 0 to 3

Table 22-1. Correspondence Between CORCN Register Bits and CORADn Registers

CORCN Register Bit	Corresponding CORADn Register
CORENn3	CORADn3
CORENn2	CORADn2
CORENn1	CORADn1
CORENn0	CORADn0

22.3 ROM Correction Operation and Program Flow

<1> If the address to be corrected and the fetch address of the internal ROM match, the fetch code is replaced by the DBTRAP instruction.

<2> When the DBTRAP instruction is executed, execution branches to address 00000060H.

<3> Software processing after branching causes the result of ROM correction to be judged (the fetch address and ROM correction operation are confirmed) and execution to branch to the correction software.

<4> After the correction software has been executed, the return address is set, and return processing is started by the DBRET instruction.

Cautions 1. The software that performs <3> and <4> must be executed in the internal ROM/RAM.

2. When setting an address to be corrected to the CORADn register, clear the higher bits to 0 in accordance with the capacity of the internal ROM.

3. The ROM correction function cannot be used to correct the data of the internal ROM. It can only be used to correct instruction codes. If ROM correction is used to correct data, that data is replaced with the DBTRAP instruction code.

Figure 22-2. ROM Correction Operation and Program Flow

Remarks 1. : Processing by user program (software)

: Processing by ROM correction (hardware)

2. n = 0 to 3

★ CHAPTER 23 FLASH MEMORY

The following product is the flash memory version of the μPD703229Y.

Caution There are differences in the amount of noise tolerance and noise radiation between flash memory versions and mask ROM versions. When considering changing from a flash memory version to a mask ROM version during the process from experimental manufacturing to mass production, make sure to sufficiently evaluate commercial samples (CS) (not engineering samples (ES)) of the mask ROM versions.
For the electrical specifications related to the flash memory rewriting, refer to CHAPTER 25 ELECTRICAL SPECIFICATIONS.

μPD70F3229Y: 384 KB on-chip flash memory

The flash memory version is commonly used in the following development environments and mass production applications.

- For altering software after the μPD70F3229Y is soldered onto the target system.
- For data adjustment when starting mass production.
- For differentiating software according to the specifications in small scale production of various models.
- For facilitating inventory management.
- For updating software after shipment.

23.1 Features

- 4-byte/1-clock access (when instruction is fetched)
- Capacity: 384 KB
- Rewrite voltage: Erase/write with a single power supply
- Rewriting method
  - Rewriting by communication with dedicated flash programmer via serial interface (on-board/off-board programming)
  - Rewriting flash memory by user program (self programming)
- Flash memory rewrite prohibit function supported (security function)
- Safe rewriting of entire flash memory area by self programming using boot swap function
- Interrupts can be acknowledged during self programming.

23.2 Memory Configuration

The 384 KB internal flash memory area is divided into 12 blocks and can be programmed/erased in block units. All the blocks can also be erased at once.

When the boot swap function is used, the physical memory located at the addresses of blocks 0 and 1 is replaced by the physical memory located at the addresses of blocks 2 and 3. For details of the boot swap function, refer to **23.5 Rewriting by Self Programming**.

Figure 23-1. Flash Memory Mapping

Note Blocks 0 and 1: Boot area
Blocks 2 and 3: Area used to replace boot area via boot swap function

23.3 Functional Outline

The internal flash memory of the μPD70F3229Y can be rewritten by using the rewrite function of the dedicated flash programmer, regardless of whether the μPD70F3229Y has already been mounted on the target system or not (on-board/off-board programming).

In addition, a security function that prohibits rewriting the user program written to the internal flash memory is also supported, so that the program cannot be changed by an unauthorized person.

The rewrite function using the user program (self programming) is ideal for an application where it is assumed that the program is changed after production/shipment of the target system. A boot swap function that rewrites the entire flash memory area safely is also supported. In addition, interrupt servicing is supported during self programming, so that the flash memory can be rewritten under various conditions, such as while communicating with an external device.

Table 23-1. Rewrite Method

Rewrite Method	Functional Outline	Operation Mode
On-board programming	Flash memory can be rewritten after the device is mounted on the target system, by using a dedicated flash programmer.	Flash memory programming mode
Off-board programming	Flash memory can be rewritten before the device is mounted on the target system, by using a dedicated flash programmer and a dedicated program adapter board (FA series).	
Self programming	Flash memory can be rewritten by executing a user program that has been written to the flash memory in advance by means of on-board/off-board programming. (During self-programming, instructions cannot be fetched from or data access cannot be made to the internal flash memory area. Therefore, the rewrite program must be transferred to the internal RAM or external memory in advance).	Normal operation mode

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

Table 23-2. Basic Functions

Function	Functional Outline	Support (○: Supported, ×: Not Supported)	
		On-Board/Off-Board Programming	Self Programming
Block erasure	The contents of specified memory blocks are erased.	○	○
Chip erasure	The contents of the entire memory area are erased all at once.	○	×
Write	Writing to specified addresses, and a verify check to see if write level is secured are performed.	○	○
Verify/checksum	Data read from the flash memory is compared with data transferred from the flash programmer.	○	× (Can be read by user program)
Blank check	The erasure status of the entire memory is checked.	○	○
Security setting	Use of the block erase command, chip erase command, and program command can be prohibited.	○	× (Only values set by on-board/off-board programming can be retained)

The following table lists the security functions. The block erase command prohibit, chip erase command prohibit, and program command prohibit functions are enabled by default after shipment, and security can be set by rewriting via on-board/off-board programming. Each security function can be used in combination with the others at the same time.

Table 23-3. Security Functions

Function	Functional Outline	Rewriting Operation When Prohibited (○: Executable, ×: Not Executable)	
		On-Board/Off-Board Programming	Self Programming
Block erase command prohibit	Execution of a block erase command on all blocks is prohibited. Setting of prohibition can be initialized by execution of a chip erase command.	Block erase command: × Chip erase command: ○ Program command: ○	Can always be rewritten regardless of setting of prohibition
Chip erase command prohibit	Execution of block erase and chip erase commands on all the blocks is prohibited. Once prohibition is set, setting of prohibition cannot be initialized because the chip erase command cannot be executed.	Block erase command: × Chip erase command: × Program command: ○	
Program command prohibit	Write and block erase commands on all the blocks are prohibited. Setting of prohibition can be initialized by execution of the chip erase command.	Block erase command: × Chip erase command: ○ Program command: ×	

23.4 Rewriting by Dedicated Flash Programmer

The flash memory can be rewritten by using a dedicated flash programmer after the μPD70F3229Y is mounted on the target system (on-board programming). The flash memory can also be rewritten before the device is mounted on the target system (off-board programming) by using a dedicated program adapter (FA series).

23.4.1 Programming environment

The following shows the environment required for writing programs to the flash memory of the μPD70F3229Y.

Figure 23-2. Environment Required for Writing Programs to Flash Memory

A host machine is required for controlling the dedicated flash programmer.

UARTA0 or CSIB0 is used for the interface between the dedicated flash programmer and the μPD70F3229Y to perform writing, erasing, etc. A dedicated program adapter (FA series) is required for off-board programming.

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

23.4.2 Communication mode

Communication between the dedicated flash programmer and the μPD70F3229Y is performed by serial communication using the UARTA0 or CSIB0 interface of the μPD70F3229Y.

(1) UARTA0

Transfer rate: 9,600 to 153,600 bps

Figure 23-3. Communication with Dedicated Flash Programmer (UARTA0)

(2) CSIB0

Serial clock: 2.4 kHz to 2.5 MHz (MSB first)

Figure 23-4. Communication with Dedicated Flash Programmer (CSIB0)

(3) CSIB0 + HS

Serial clock: 2.4 kHz to 2.5 MHz (MSB first)

Figure 23-5. Communication with Dedicated Flash Programmer (CSIB0 + HS)

The dedicated flash programmer outputs the transfer clock, and the μPD70F3229Y operates as a slave.

When the PG-FP4 is used as the dedicated flash programmer, it generates the following signals for the μPD70F3229Y. For details, refer to the **PG-FP4 User's Manual (U15260E)**.

Table 23-4. Signal Connections of Dedicated Flash Programmer (PG-FP4)

PG-FP4			μPD70F3229Y	Processing for Connection		
Signal Name	I/O	Pin Function	Pin Name	UARTA0	CSIB0	CSIB0 + HS
FLMD0	Output	Write enable/disable	FLMD0	◎	◎	◎
FLMD1	Output	Write enable/disable	FLMD1	◎ Note 1	◎ Note 1	◎ Note 1
VDD	–	V_{DD} voltage generation/voltage monitor	V_{DD}	× Note 2	× Note 2	× Note 2
GND	–	Ground	V_{SS}	◎	◎	◎
CLK	Output	Clock output to μPD70F3229Y	X1, X2	× Note 3	× Note 3	× Note 3
$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$	◎	◎	◎
SI/RxD	Input	Receive signal	SOB0/TXDA0	◎	◎	◎
SO/TxD	Output	Transmit signal	SIB0/RXDA0	◎	◎	◎
SCK	Output	Transfer clock	$\overline{\text{SCKB0}}$	×	◎	◎
HS	Input	Handshake signal for CSIB0 + HS communication	PCM0	×	×	◎

Notes 1. Wire these pins as shown in Figure 23-6, or connect then to GND via a pull-down resistor on board.

2. Connect these pins to supply power from the PG-FP4.

3. Connect these pins to supply a clock from the PG-FP4 (wire as shown in Figure 23-6, or create an oscillator on board and supply the clock).

Remark ◎: Must be connected.

×: Does not have to be connected.

Table 23-5. Wiring of μPD70F3229Y Flash Writing Adapters (FA-100GC-8EA)

Flash Programmer (PG-FP4) Connection Pin			Name of FA Board Pin	CSIB0 + HS Used		CSIB0 Used		UARTA0 Used	
Signal Name	I/O	Pin Function		Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
SI/RxD	Input	Receive signal	SI	P41/SOB0/ SCL00	23	P41/SOB0/ SCL00	23	P30/TXDA0	25
SO/TxD	Output	Transmit signal	SO	P40/SIB0/ SDA00	22	P40/SIB0/ SDA00	22	P31/RXDA0/ INTP7	26
SCK	Output	Transfer clock	SCK	P42/$\overline{\text{SCKB0}}$	24	P42/$\overline{\text{SCKB0}}$	24	Not needed	–
CLK	Output	Clock to μPD70F3229Y	X1	X1	12	X1	12	X1	12
			X2	X2	13	X2	13	X2	13
/RESET	Output	Reset signal	/RESET	$\overline{\text{RESET}}$	14	$\overline{\text{RESET}}$	14	$\overline{\text{RESET}}$	14
FLMD0	Input	Write voltage	FLMD0	FLMD0	8	FLMD0	8	FLMD0	8
FLMD1	Input	Write voltage	FLMD1	PLD5/AD5/ FLMD1	76	PLD5/AD5/ FLMD1	76	PLD5/AD5/ FLMD1	76
HS	Input	Handshake signal for CSI0 + HS communication	RESERVE/ HS	PCM0/$\overline{\text{WAIT}}$	61	Not needed	–	Not needed	–
VDD	–	VDD voltage generation/ voltage monitor	VDD	V_{DD}	9	V_{DD}	9	V_{DD}	9
				BV_{DD}	70	BV_{DD}	70	BV_{DD}	70
				EV_{DD}	5, 34	EV_{DD}	5, 34	EV_{DD}	5, 34
				AV_{REF0}	1	AV_{REF0}	1	AV_{REF0}	1
GND	–	Ground	GND	V_{SS}	11	V_{SS}	11	V_{SS}	11
				AV_{SS}	2	AV_{SS}	2	AV_{SS}	2
				BV_{SS}	69	BV_{SS}	69	BV_{SS}	69
				EV_{SS}	33	EV_{SS}	33	EV_{SS}	33

Cautions 1. Be sure to connect the REGC pin to GND via a 4.7 μF capacitor.

2. A clock cannot be supplied from the CLK pin of the flash programmer. Create an oscillator on the board to supply the clock.

Figure 23-6. Wiring Example of μPD70F3229Y Flash Writing Adapter (FA-100GC-8EA) (1/2)

Figure 23-6. Wiring Example of μPD70F3229Y Flash Writing Adapter (FA-100GC-8EA) (2/2)

Notes 1. Wire the FLMD1 pin as shown below, or connect it to GND on board via a pull-down resistor.

2. Create an oscillator on the flash writing adapter (shown in broken lines) to supply a clock. Here is an example of the oscillator.

Example:

3. Corresponding pins when UARTA0 is used.

Caution Do not input a high level to the $\overline{\text{DRST}}$ pin.

Remarks 1. Handle the pins not shown in accordance with the handling of unused pins (refer to **2.3 Pin I/O Circuit Types, I/O Buffer Power Supplies and Handling of Unused Pins**). It is recommended to use a resistor of 1 kΩ to 10 kΩ.

2. This adapter is for a 100-pin plastic LQFP package.

3. This figure shows the wiring when CSIB0 supporting handshaking is used.

23.4.3 Flash memory control

The following shows the procedure for manipulating the flash memory.

Figure 23-7. Procedure for Manipulating Flash Memory

23.4.4 Selection of communication mode

In the μPD70F3229Y, the communication mode is selected by inputting pulses (11 pulses max.) to the FLMD0 pin after switching to the flash memory programming mode. The FLMD0 pulse is generated by the dedicated flash programmer.

The following shows the relationship between the number of pulses and the communication mode.

Figure 23-8. Selection of Communication Mode

Note The number of clocks is as follows depending on the communication mode.

FLMD0 Pulse	Communication Mode	Remarks
0	UARTA0	Communication rate: 9,600 bps (after reset), LSB first
8	CSIB0	μPD70F3229Y performs slave operation, MSB first
11	CSIB0 + HS	μPD70F3229Y performs slave operation, MSB first
Other	RFU	Setting prohibited

Caution When UARTA0 is selected, the receive clock is calculated based on the reset command sent from the dedicated flash programmer after receiving the FLMD0 pulse.

23.4.5 Communication commands

The μPD70F3229Y communicates with the dedicated flash programmer by means of commands. The signals sent from the dedicated flash programmer to the μPD70F3229Y are called “commands”. The response signals sent from the μPD70F3229Y to the dedicated flash programmer are called “response commands”.

Figure 23-9. Communication Commands

The following shows the commands for flash memory control in the μPD70F3229Y. All of these commands are issued from the dedicated flash programmer, and the μPD70F3229Y performs the processing corresponding to the commands.

Table 23-6. Flash Memory Control Commands

Classification	Command Name	Support			Function
		CSIB0	CSIB0 + HS	UARTA0	
Blank check	Block blank check command	○	○	○	Checks if the contents of the memory in the specified block have been correctly erased.
Erase	Chip erase command	○	○	○	Erases the contents of the entire memory.
	Block erase command	○	○	○	Erases the contents of the memory of the specified block.
Write	Write command	○	○	○	Writes the specified address range, and executes a contents verify check.
Verify	Verify command	○	○	○	Compares the contents of memory in the specified address range with data transferred from the flash programmer.
	Checksum command	○	○	○	Reads the checksum in the specified address range.
System setting, control	Silicon signature command	○	○	○	Reads silicon signature information.
	Security setting command	○	○	○	Disables the chip erase command, enables the block erase command, and disables the write command.

23.4.6 Pin connection

When performing on-board programming, mount a connector on the target system to connect to the dedicated flash programmer. Also, incorporate a function on-board to switch from the normal operation mode to the flash memory programming mode.

In the flash memory programming mode, all the pins not used for flash memory programming become the same status as that immediately after reset. Therefore, pin handling is required when the external device does not acknowledge the status immediately after a reset.

(1) FLMD0 pin

In the normal operation mode, input a voltage of V_{SS} level to the FLMD0 pin. In the flash memory programming mode, supply a write voltage of V_{DD} level to the FLMD0 pin.

Because the FLMD0 pin serves as a write protection pin in the self programming mode, a voltage of V_{DD} level must be supplied to the FLMD0 pin via port control, etc., before writing to the flash memory. For details, refer to **23.5.5 (1) FLMD0 pin**.

Figure 23-10. FLMD0 Pin Connection Example

(2) FLMD1 pin

When 0 V is input to the FLMD0 pin, the FLMD1 pin does not function. When V_{DD} is supplied to the FLMD0 pin, the flash memory programming mode is entered, so 0 V must be input to the FLMD1 pin. The following shows an example of the connection of the FLMD1 pin.

Figure 23-11. FLMD1 Pin Connection Example

Caution If the V_{DD} signal is input to the FLMD1 pin from another device during on-board programming and immediately after reset, isolate this signal.

Table 23-7. Relationship Between FLMD0 and FLMD1 Pins and Operation Mode When Reset Is Released

FLMD0	FLMD1	Operation Mode
0	Don't care	Normal operation mode
V_{DD}	0	Flash memory programming mode
V_{DD}	V_{DD}	Setting prohibited

(3) Serial interface pin

The following shows the pins used by each serial interface.

Table 23-8. Pins Used by Serial Interfaces

Serial Interface	Pins Used
UARTA0	TXDA0, RXDA0
CSIB0	SOB0, SIB0, $\overline{\text{SCKB0}}$
CSIB0 + HS	SOB0, SIB0, $\overline{\text{SCKB0}}$, PCM0

When connecting a dedicated flash programmer to a serial interface pin that is connected to another device on-board, care should be taken to avoid conflict of signals and malfunction of the other device.

(a) Conflict of signals

When the dedicated flash programmer (output) is connected to a serial interface pin (input) that is connected to another device (output), a conflict of signals occurs. To avoid the conflict of signals, isolate the connection to the other device or set the other device to the output high-impedance status.

Figure 23-12. Conflict of Signals (Serial Interface Input Pin)

(b) Malfunction of other device

When the dedicated flash programmer (output or input) is connected to a serial interface pin (input or output) that is connected to another device (input), the signal is output to the other device, causing the device to malfunction. To avoid this, isolate the connection to the other device.

Figure 23-13. Malfunction of Other Device

(4) $\overline{\text{RESET}}$ pin

When the reset signals of the dedicated flash programmer are connected to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generator on-board, a conflict of signals occurs. To avoid the conflict of signals, isolate the connection to the reset signal generator.

When a reset signal is input from the user system in the flash memory programming mode, the programming operation will not be performed correctly. Therefore, do not input signals other than the reset signals from the dedicated flash programmer.

Figure 23-14. Conflict of Signals ($\overline{\text{RESET}}$ Pin)

(5) Port pins (including NMI)

When the system shifts to the flash memory programming mode, all the pins that are not used for flash memory programming are in the same status as that immediately after reset. If the external device connected to each port does not recognize the status of the port immediately after reset, pins require appropriate processing, such as connecting to V_{DD} via a resistor or connecting to V_{SS} via a resistor.

(6) Other signal pins

Connect X1, X2, XT1, XT2, and REGC in the same status as that in the normal operation mode.

During flash memory programming, input a low level to the $\overline{\text{DRST}}$ pin or leave it open. Do not input a high level.

(7) Power supply

Supply the same power (V_{DD}, V_{SS}, EV_{DD}, EV_{SS}, AV_{SS}, BV_{DD}, BV_{SS}, AV_{REF0}) as in normal operation mode.

23.5 Rewriting by Self Programming

23.5.1 Overview

The μPD70F3229Y supports a flash macro service that allows the user program to rewrite the internal flash memory by itself. By using this interface and a self programming library that is used to rewrite the flash memory with a user application program, the flash memory can be rewritten by a user application transferred in advance to the internal RAM or external memory. Consequently, the user program can be upgraded and constant data can be rewritten in the field.

Figure 23-15. Concept of Self Programming

23.5.2 Features

(1) Secure self programming (boot swap function)

The μPD70F3229Y supports a boot swap function that can exchange the physical memory of blocks 0 and 1 with the physical memory of blocks 2 and 3. By writing the start program to be rewritten to blocks 2 and 3 in advance and then swapping the physical memory, the entire area can be safely rewritten even if a power failure occurs during rewriting because the correct user program always exists in blocks 0 and 1.

Figure 23-16. Rewriting Entire Memory Area (Boot Swap)

(2) Interrupt support

Instructions cannot be fetched from the flash memory during self programming. Conventionally, therefore, a user handler written to the flash memory could not be used even if an interrupt occurred. With the μPD70F3229Y, a user handler can be registered to an entry RAM area by using a library function, so that interrupt servicing can be performed by internal RAM or external memory execution.

23.5.3 Standard self programming flow

The entire processing to rewrite the flash memory by flash self programming is illustrated below.

Figure 23-17. Standard Self Programming Flow

Notes 1. If a security setting is not performed, flash information setting processing does not have to be executed.

2. If boot swap is not used, flash information setting processing and boot area swap processing do not have to be executed.

23.5.4 Flash functions

Table 23-9. Flash Function List

Function Name	Outline	Support
FlashEnv	Initialization of flash control macro	√
FlashBlockErase	Erasure of only specified one block	√
FlashWordWrite	Writing from specified address	√
FlashBlockIVerify	Internal verification of specified one block	√
FlashBlockBlankCheck	Blank check of specified one block	√
FlashFLMDCheck	Check of FLMD pin	√
FlashStatusCheck	Status check of operation specified immediately before	√
FlashGetInfo	Reading of flash information	√
FlashSetInfo	Setting of flash information	√
FlashBootSwap	Swapping of boot area	√
FlashSetUserHandler	User interrupt handler registration function	√

23.5.5 Pin processing

(1) FLMD0 pin

The FLMD0 pin is used to set the operation mode when reset is released and to protect the flash memory from being written during self rewriting. It is therefore necessary to keep the voltage applied to the FLMD0 pin at 0 V when reset is released and a normal operation is executed. It is also necessary to apply a voltage of V_{DD} level to the FLMD0 pin during the self programming mode period via port control before the memory is rewritten.

When self programming has been completed, the voltage on the FLMD0 pin must be returned to 0 V.

Caution When noise exceeding the noise elimination width is input to the $\overline{\text{RESET}}$ pin during self programming, if a capacitor is connected to the FLMD0 pin, the mode switches to on-board programming, depending on the timing of releasing the capacitor charge. Therefore, do not connect a capacitor to the FLMD0 pin.

Figure 23-18. Mode Change Timing

Caution Make sure that the FLMD0 pin is at 0 V when reset is released.

23.5.6 Internal resources used

The following table lists the internal resources used for self programming. These internal resources can also be used freely for purposes other than self programming.

Table 23-10. Internal Resources Used

Resource Name	Description
Entry RAM area (internal RAM/external RAM size[Note])	Routines and parameters used for the flash macro service are located in this area. The entry program and default parameters are copied by calling a library initialization function.
Stack area (stack size[Note])	An extension of the stack used by the user is used by the library (can be used in both the internal RAM and external RAM).
Library code (code size[Note])	Program entity of library (can be used anywhere other than the flash memory block to be manipulated).
Application program	Executed as user application. Calls flash functions.
Maskable interrupt	Can be used in user application execution status or self programming status. To use this interrupt in the self programming status, the interrupt servicing start address must be registered in advance by a registration function.
NMI interrupt	Can be used in user application execution status or self programming status. To use this interrupt in the self programming status, the interrupt servicing start address must be registered in advance by a registration function.

Note For the capacity to be used, refer to the **V850 Series Flash Memory Self Programming (Single Power Supply Flash Memory) User's Manual** (under preparation).

★ CHAPTER 24 ON-CHIP DEBUG FUNCTION

The μPD70F3229 has an on-chip debug function that uses the JTAG (Joint Test Action Group) interface ($\overline{\text{DRST}}$, DCK, DMS, DDI, and DDO pins) and that can be used via an N-Wire emulator (IE-V850E1-CD-NW).

24.1 Features

- ○ Hardware break function: 2 points
- ○ Software break function: 4 points
- ○ Real-time RAM monitor function: Memory contents can be read during program execution.
- ○ Dynamic memory modification function (DMM function): RAM contents can be rewritten during program execution.
- ○ Mask function: $\overline{\text{RESET}}$, NMI, $\overline{\text{HLDRQ}}$, $\overline{\text{WAIT}}$
- ○ ROM security function: 10-byte ID code authentication

Caution The following functions are not supported.

- **Trace function**
- **Event function**
- **Debug interrupt interface function (DBINT)**

24.2 Connection Circuit Example

Notes 1. Example of pin processing when N-Wire emulator is not connected

2. A pull-down resistor is provided on chip.

3. For flash memory rewriting

24.3 Interface Signals

The interface signals are described below.

(1) $\overline{\text{DRST}}$

This is a reset input signal for the on-chip debug unit. It is a negative-logic signal that asynchronously initializes the debug control unit.

The IE-V850E1-CD-NW raises the $\overline{\text{DRST}}$ signal when it detects V_{DD} of the target system after the integrated debugger is started, and starts the on-chip debug unit of the device.

When the $\overline{\text{DRST}}$ signal goes high, a reset signal is also generated in the CPU.

When starting debugging by starting the integrated debugger, a CPU reset is always generated.

(2) DCK

This is a clock input signal. It supplies a 20 MHz clock from the IE-V850E1-CD-NW. In the on-chip debug unit, the DMS and DDI signals are sampled at the rising edge of the DCK signal, and the data DDO is output at its falling edge.

(3) DMS

This is a transfer mode select signal. The transfer status in the debug unit changes depending on the level of the DMS signal.

(4) DDI

This is a data input signal. It is sampled in the on-chip debug unit at the rising edge of DCK.

(5) DDO

This is a data output signal. It is output from the on-chip debug unit at the falling edge of the DCK signal.

(6) EV_{DD}

This signal is used to detect VDD of the target system. If VDD from the target system is not detected, the signals output from the IE-V850E1-CD-NW ($\overline{DRST}$, DCK, DMS, DDI, FLMD0, and $\overline{RESET}$) go into a high-impedance state.

(7) FLMD0

The flash self programming function is used for the function to download data to the flash memory via the integrated debugger. During flash self programming, the FLMD0 pin must be kept high. In addition, connect a pull-down resistor to the FLMD0 pin.

The FLMD0 pin can be controlled in either of the following two ways.

<1> To control from IE-V850E1-CD-NW

Connect the FLMD0 signal of the IE-V850E1-CD-NW to the FLMD0 pin.

In the normal mode, nothing is driven by the IE-V850E1-CD-NW (high impedance).

During a break, the IE-V850E1-CD-NW raises the FLMD0 pin to the high level when the download function of the integrated debugger is executed.

<2> To control from port

Connect any port of the device to the FLMD0 pin.

The same port as the one used by the user program to realize the flash self programming function may be used.

On the console of the integrated debugger, make a setting to raise the port pin to high level before executing the download function, or lower the port pin to low level after executing the download function.

For details, refer to the **ID850NWC Ver. 2.51 Integrated Debugger Operation User's Manual (U16525E)**.

(8) $\overline{RESET}$

This is a system reset input signal. If the $\overline{DRST}$ pin is made invalid by the value of the OCDM0 bit of the OCDM register set by the user program, on-chip debugging cannot be executed. Therefore, reset is effected by the IE-V850E1-CD-NW, using the $\overline{RESET}$ pin, to make the $\overline{DRST}$ pin valid (initialization).

24.4 Control Register

(1) On-chip debug mode register (OCDM)

The OCDM register is used to select the normal operation mode or on-chip debug mode. This register is a special register and can be written only in a combination of specific sequences (refer to **3.4.7 Special registers**).

This register is also used to specify whether a pin provided with an on-chip debug function is used as an on-chip debug pin or as an ordinary port/peripheral function pin. It also is used to disconnect the internal pull-down resistor of the P05/INTP2/$\overline{\text{DRST}}$ pin.

The OCDM register can be written only while a low level is input to the $\overline{\text{DRST}}$ pin.

This register can be read or written in 8-bit or 1-bit units.

After reset: 01H[Note] R/W Address: FFFFF9FCH

	7	6	5	4	3	2	1	<0>
OCDM	0	0	0	0	0	0	0	OCDM0

OCDM0	Operation mode
0	Selects normal operation mode (in which a pin that functions alternately as on-chip debug function pin is used as a port/peripheral function pin) and disconnects the on-chip pull-down resistor of the P05/INTP2/$\overline{\text{DRST}}$ pin.
1	When $\overline{\text{DRST}}$ pin is low: Normal operation mode (in which a pin that functions alternately as an on-chip debug function pin is used as a port/peripheral function pin) When $\overline{\text{DRST}}$ pin is high: On-chip debug mode (in which a pin that functions alternately as an on-chip debug function pin is used as an on-chip debug mode pin)

Note $\overline{\text{RESET}}$ input sets this register to 01H. After reset by the WDT2RES signal, clock monitor (CLM), or low-voltage detector (LVI), however, the value of the OCDM register is retained.

Cautions 1. When using the DDI, DDO, DCK, and DMS pins not as on-chip debug pins but as port pins after external reset, any of the following actions must be taken.

- **Input a low level to the P05/INTP2/$\overline{\text{DRST}}$ pin.**
- **Set the ODCM0 bit. In this case, take the following actions.**
  <1> Clear the OCDM0 bit to 0.
  <2> Fix the P05/INTP2/$\overline{\text{DRST}}$ pin to the low level until <1> is completed.

2. The $\overline{\text{DRST}}$ pin has an on-chip pull-down resistor. This resistor is disconnected when the OCDM0 flag is cleared to 0. The mask ROM version does not have an on-chip debug function but it has the above pull-down resistor. With the mask ROM version also, therefore, the on-chip pull-down resistor must be disconnected by clearing the OCDM0 bit to 0.

24.5 Operation

The on-chip debug function is made valid under the conditions shown in the table below.
When this function is not used, keep the $\overline{\text{DRST}}$ pin low until the OCDM.OCDM0 flag is cleared to 0.

OCDM0 Flag / $\overline{\text{DRST}}$ Pin	0	1
L	Invalid	Invalid
H	Invalid	Valid

Remark L: Low-level input
H: High-level input

Figure 24-1. Timing When On-Chip Debug Function Is Not Used

24.6 ROM Security Function

24.6.1 Security ID

The flash memory versions of the μPD70F3229Y perform authentication using a 10-byte ID code to prevent the contents of the flash memory from being read by an unauthorized person during on-chip debugging by the N-Wire emulator.

Set the ID code in the 10-byte on-chip flash memory area from 0000070H to 0000079H to allow the debugger perform ID authentication.

If the IDs match, the security is released and reading flash memory and using the N-Wire emulator are enabled.

- Set the 10-byte ID code to 0000070H to 0000079H.
- Bit 7 of 0000079H is the N-Wire emulator enable flag.
  (0: Disable, 1: Enable)
- When the N-Wire emulator is started, the debugger requests ID input. When the ID code input on the debugger and the ID code set in 0000070H to 0000079H match, the debugger starts.
- Debugging cannot be performed if the N-Wire emulator enable flag is 0, even if the ID codes match.

Caution When the data in the flash memory has been deleted, all the bits are set to 1.

24.6.2 Setting

When the ID code is “112233445566778899AAH”

Address	Value	
0x70	0xAA	
0x71	0x99	
0x72	0x88	
0x73	0x77	
0x74	0x66	
0x75	0x55	
0x76	0x44	
0x77	0x33	
0x78	0x22	
0x79	0x11	
0x7A	0x00	← Reserved code (Refer to **3.4.8 (3)**.)

When the CA850 is used, continue using the handler at address 0x60 (ILGOP) to input the security code and the data of the system reserved area (00H).

Caution Description of a link directive is not necessary because the handler name at address 0x60 is used.

[Program example]

```
#-------------------------------------
#       ILGOP  handler
#-------------------------------------
        .section        "ILGOP"    --Interrupt handler address 0x60
                                   --Input ILGOP handler code
        .org            0x10       --Skip handler address to 0x70
#-------------------------------------
#       SECURITYID (continue ILGOP  handler)
#-------------------------------------
        .word                      0x778899aa   --0-3 byte code
        .word                      0x33445566   --4-7 byte code
        .hword                     0x1122       --8-9 byte code
        .byte                      0x00         --Reserve code
```

24.7 Cautions

- If a reset signal is input (from the target system or a reset signal from an internal reset source) during RUN (program execution), the break function may malfunction.

- Even if the reset signal is masked by the mask function, the I/O buffer (port pin) may be reset if a reset signal is input from a pin.

- Because a software breakpoint set in the internal flash memory is realized by the ROM correction function, it is made temporarily invalid by target reset or internal reset generated by watchdog timer 2. The breakpoint becomes valid again when a hardware break or forced break occurs, but a software break does not occur until then.

- Pin reset during a break is masked and the CPU and peripheral I/O are not reset. If pin reset or internal reset is generated as soon as the flash memory is rewritten by DMA or read by the RAM monitor function while the user program is being executed, the CPU and peripheral I/O may not be correctly reset.

- Emulation of ROM correction cannot be executed.

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}	$BV_{DD} \leq V_{DD} = EV_{DD}$	–0.5 to +6.5	V
	EV_{DD}	$BV_{DD} \leq V_{DD} = EV_{DD}$	–0.5 to +6.5	V
	BV_{DD}	$BV_{DD} \leq V_{DD} = EV_{DD}$	–0.5 to +6.5	V
	AV_{REF0}		–0.5 to +6.5	V
	V_{SS}	$V_{SS} = EV_{SS} = BV_{SS} = AV_{SS}$	–0.5 to +0.5	V
	EV_{SS}	$V_{SS} = EV_{SS} = BV_{SS} = AV_{SS}$	–0.5 to +0.5	V
	BV_{SS}	$V_{SS} = EV_{SS} = BV_{SS} = AV_{SS}$	–0.5 to +0.5	V
	AV_{SS}	$V_{SS} = EV_{SS} = BV_{SS} = AV_{SS}$	–0.5 to +0.5	V
Input voltage	V_{I1}	P00 to P06, P10, P11, P30 to P39, P40 to P42, P50 to P55, P90 to P915, $\overline{\text{RESET}}$, FLMD0	–0.5 to EV_{DD} + 0.5 Note 1	V
	V_{I2}	PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 to PDL15, PDH0, PDH1	–0.5 to BV_{DD} + 0.5 Note 1	V
	V_{I3}	X1, X2, XT1, XT2	–0.5 to V_{RO} Note 2 + 0.5 Note 1	V
Analog input voltage	V_{IAN}	P70 to P711	–0.5 to AV_{REF0} + 0.5 Note 1	V

Notes 1. Be sure not to exceed the absolute maximum ratings (MAX. value) of each supply voltage.

2. On-chip regulator output voltage (2.5 V (TYP.))

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
Output current, low	I_{OL}	P00 to P06, P10, P11, P30 to P39, P40 to P42, P50 to P55, P90 to P915	Per pin	4	mA
			Total of all pins	50	mA
		P70 to P711	Per pin	4	mA
			Total of all pins	20	mA
		PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 to PDL15, PDH0, PDH1	Per pin	4	mA
			Total of all pins	50	mA
Output current, high	I_{OH}	P00 to P06, P10, P11, P30 to P39, P40 to P42, P50 to P55, P90 to P915	Per pin	−4	mA
			Total of all pins	−50	mA
		P70 to P711	Per pin	−4	mA
			Total of all pins	−20	mA
		PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 to PDL15, PDH0, PDH1	Per pin	−4	mA
			Total of all pins	−50	mA
Operating ambient temperature	T_A			–40 to +85	°C
Storage temperature	T_{stg}	μPD703229Y		−65 to +150	°C
		μPD70F3229Y		−40 to +125	°C

Cautions 1. Do not directly connect the output (or I/O) pins of IC products to each other, or to V_{DD}, V_{CC}, and GND. Open-drain pins or open-collector pins, however, can be directly connected to each other. Direct connection of the output pins between an IC product and an external circuit is possible, if the output pins can be set to the high-impedance state and the output timing of the external circuit is designed to avoid output conflict.

2. 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.

The ratings and conditions indicated for DC characteristics and AC characteristics represent the quality assurance range during normal operation.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Capacitance (T_A = 25°C, V_{DD} = EV_{DD} = BV_{DD} = AV_{REF0} = V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
I/O capacitance	C_{IO}	f_X = 1 MHz Unmeasured pins returned to 0 V			10	pF

Operating Conditions

(T_A = –40 to +85°C, $BV_{DD} \leq V_{DD}$ = EV_{DD}, AV_{REF0} = 4.0 to 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Internal System Clock Frequency	Conditions	Supply Voltage		Unit
		V_{DD}	BV_{DD}	
f_{XX} = 4 to 20 MHz	C = 4.7 μF	3.5 to 5.5	3.0 to 5.5	V
f_{XT} = 32.768 kHz	C = 4.7 μF	3.5 to 5.5	3.0 to 5.5	V

PLL Characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, BV_{DD} = 3.0 to 5.5 V, $BV_{DD} \leq V_{DD}$, AV_{REF0} = 4.0 to 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input frequency	f_X		4		5	MHz
Output frequency	f_{XX}		16		20	MHz
Lock time	t_{PLL}	After V_{DD} reaches 3.5 V (MIN.)			800	μs

Ring-OSC Characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, BV_{DD} = 3.0 to 5.5 V, $BV_{DD} \leq V_{DD}$, AV_{REF0} = 4.0 to 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Output frequency	f_R		100	200	400	kHz

Regulator Characteristics

(T_A = –40 to +85°C, $BV_{DD} \leq V_{DD}$ = EV_{DD}, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage	V_{DD}		3.5		5.5	V
Output voltage	V_{RO}			2.5		V
Regulator output stabilization time	t_{RG}	After V_{DD} reaches 3.5 V (MIN.), Stabilization capacitance C = 4.7 μF connected to REGC pin			1	ms

Main Clock Oscillator Characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, BV_{DD} = 3.0 to 5.5 V, $BV_{DD} \leq V_{DD}$, AV_{REF0} = 4.0 to 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Resonator	Circuit Example	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator/ crystal resonator		Oscillation frequency (fx)[Note 1]		4		5	MHz
		Oscillation stabilization time[Note 2]	After reset is released		2^{16}/fx		s
			After STOP mode is released	1[Note 4]	**Note 3**		ms
			After IDLE2 mode is released	350[Note 4]	**Note 3**		μs

Notes 1. The oscillation frequency shown above indicates only oscillator characteristics. Use the μPD703229Y and 70F3229Y so that the internal operation conditions do not exceed the ratings shown in AC characteristics and DC characteristics.

2. Time required from when V_{DD} reaches the oscillation voltage range (3.5 V (MIN.)) to when the crystal resonator stabilizes.

3. The value varies depending on the setting of the OSTS register.

4. Time required to set up the flash memory. Secure the setup time using the OSTS register.

Cautions 1. When using the main clock oscillator, wire as follows in the area enclosed by the broken lines in the above figure to avoid an adverse effect from wiring capacitance.

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

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

3. For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

Subclock Oscillator Characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, BV_{DD} = 3.0 to 5.5 V, $BV_{DD} \leq V_{DD}$, AV_{REF0} = 4.0 to 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Resonator	Circuit Example	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		Oscillation frequency (f_{XT})[Note 1]		32	32.768	35	kHz
		Oscillation stabilization time[Note 2]				10	s

Notes 1. The oscillation frequency shown above indicates only oscillator characteristics. Use the μPD703229Y and 70F3229Y so that the internal operation conditions do not exceed the ratings shown in AC characteristics and DC characteristics.

2. Time required from when V_{DD} reaches the oscillation voltage range (3.5 V (MIN.)) to when the crystal resonator stabilizes.

Cautions 1. When using the subclock oscillator, wire as follows in the area enclosed by the broken lines in the above figure to avoid an adverse effect from wiring capacitance.

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

2. The subclock oscillator is designed as a low-amplitude circuit for reducing current consumption, and is more prone to malfunction due to noise than the main clock oscillator.
Particular care is therefore required with the wiring method when the subclock is used.

3. For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

DC Characteristics

(T_A = –40 to 85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, BV_{DD} = 3.0 to 5.5 V, $BV_{DD} \leq V_{DD}$, AV_{REF0} = 4.0 to 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V) (1/3)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P30, P36 to P38, P40, P41, P92 to P95, P98, P910 to P912	$0.7EV_{DD}$		EV_{DD}	V
	V_{IH2}	P00 to P06, P10, P11, P31 to P35, P39, P42, P50 to P55, P90, P91, P96, P97, P99, P913 to P915	$0.8EV_{DD}$		EV_{DD}	V
	V_{IH3}	PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 to PDL15, PDH0, PDH1	$0.7BV_{DD}$		BV_{DD}	V
	V_{IH4}	P70 to P711	$0.7AV_{REF0}$		AV_{REF0}	V
	V_{IH5}	$\overline{RESET}$	$0.8EV_{DD}$		EV_{DD}	V
Input voltage, low	V_{IL1}	P30, P36 to P38, P40, P41, P92 to P95, P98, P910 to P912	EV_{SS}		$0.3EV_{DD}$	V
	V_{IL2}	P00 to P06, P10, P11, P31 to P35, P39, P42, P50 to P55, P90, P91, P96, P97, P99, P913 to P915	EV_{SS}		$0.2EV_{DD}$	V
	V_{IL3}	PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 to PDL15, PDH0, PDH1	BV_{SS}		$0.3BV_{DD}$	V
	V_{IL4}	P70 to P711	AV_{SS}		$0.3AV_{REF0}$	V
	V_{IL5}	$\overline{RESET}$	EV_{SS}		$0.2EV_{DD}$	V
Input leakage current, high	I_{LIH}	$V_I = V_{DD}$			5	μA
Input leakage current, low	I_{LIL}	V_I = 0 V			–5	μA
Output leakage current, high	I_{LOH}	$V_O = V_{DD}$			5	μA
Output leakage current, low	I_{LOL}	V_O = 0 V			–5	μA

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics

(T_A = –40 to 85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, BV_{DD} = 3.0 to 5.5 V, $BV_{DD} \leq V_{DD}$, AV_{REF0} = 4.0 to 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V) (2/3)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Output voltage, high[Note 1]	V_{OH1}	**Note 1**	Per pin I_{OH} = –1.0 mA	Total of all pins –20 mA	EV_{DD} – 1.0		EV_{DD}	V
			Per pin I_{OH} = –100 μA	Total of all pins –4.4 mA	EV_{DD} – 0.5		EV_{DD}	V
	V_{OH2}	**Note 2**	Per pin I_{OH} = –1.0 mA	Total of all pins –20 mA	BV_{DD} – 1.0		BV_{DD}	V
			Per pin I_{OH} = –100 μA	Total of all pins –2.8 mA	BV_{DD} – 0.5		BV_{DD}	V
	V_{OH3}	P70 to P711	Per pin I_{OH} = –1.0 mA	Total of all pins –12 mA	A_{REF0} – 1.0		AV_{REF0}	V
			Per pin I_{OH} = –100 μA	Total of all pins –1.2 mA	A_{REF0} – 0.5		AV_{REF0}	V
Output voltage, low[Note 1]	V_{OL1}	**Note 3**	Per pin I_{OL} = 1.0 mA	Total of all pins 20 mA	0		0.4	V
	V_{OL4}	P40, P41	Per pin I_{OL} = 3.0 mA		0		0.4	V
	V_{OL2}	**Note 2**	Per pin I_{OL} = 1.0 mA	Total of all pins 20 mA	0		0.4	V
	V_{OL3}	P70 to P711	Per pin I_{OL} = 1.0 mA	Total of all pins 12 mA	0		0.4	V
Software pull-up resistance	R_1	V_I = 0 V			10	30	100	kΩ
Software pull-down resistance[Note 2]	R_2	V_I = V_{DD}			10	30	100	kΩ

Notes 1. P00 to P06, P10, P11, P30 to P39, P40 to P42, P50 to P55, P90 to P915

2. PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 to PDL15, PDH0, PDH1

3. P00 to P06, P10, P11, P30 to P39, P42, P50 to P55, P90 to P915

4. $\overline{\text{DRST}}$ pin only

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

2. When the I_{OH} and I_{OL} conditions are not satisfied for a pin but the total value of all pins is satisfied, only that pin does not satisfy the DC characteristics.

DC Characteristics

(T_A = –40 to 85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, BV_{DD} = 3.0 to 5.5 V, $BV_{DD} \leq V_{DD}$, AV_{REF0} = 4.0 to 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V) (3/3)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Supply current[Note] (μPD70F3229Y)	I_{DD1}	Normal operation	f_{XX} = 20 MHz (f_X = 5 MHz) PLL mode		29	55	mA
	I_{DD2}	HALT mode	f_{XX} = 20 MHz (f_X = 5 MHz) PLL mode		19	35	mA
	I_{DD3}	IDLE1 mode	f_{XX} = 5 MHz (f_X = 5 MHz) PLL off		0.6	2.0	mA
	I_{DD4}	IDLE2 mode	f_{XX} = 5 MHz (f_X = 5 MHz) PLL off		0.3	1.0	mA
	I_{DD5}	Subclock operating mode	f_{XT} = 32.768 kHz, main clock stopped, Ring-OSC operating		200	500	μA
	I_{DD6}	Sub-IDLE mode	f_{XT} = 32.768 kHz main clock stopped, Ring-OSC operating		30	130	μA
	I_{DD7}	STOP mode	main clock, subclock, Ring-OSC stopped		10	50	μA
			main clock, subclock stopped, Ring-OSC operating		20	70	μA
Supply current[Note] (μPD703229Y)	I_{DD1}	Normal operation	f_{XX} = 20 MHz (f_X = 5 MHz) PLL mode		20	35	mA
	I_{DD2}	HALT mode	f_{XX} = 20 MHz (f_X = 5 MHz) PLL mode		12	20	mA
	I_{DD3}	IDLE1 mode	f_{XX} = 5 MHz (f_X = 5 MHz) PLL off		0.4	1.1	mA
	I_{DD4}	IDLE2 mode	f_{XX} = 5 MHz (f_X = 5 MHz) PLL off		0.3	1.0	mA
	I_{DD5}	Subclock operating mode	f_{XT} = 32.768 kHz, main clock stopped, Ring-OSC operating		50	350	μA
	I_{DD6}	Sub-IDLE mode	f_{XT} = 32.768 kHz main clock stopped, Ring-OSC operating		30	120	μA
	I_{DD7}	STOP mode	main clock, subclock, Ring-OSC stopped		10	50	μA
			main clock, subclock stopped, Ring-OSC operating		20	70	μA

Note Current flowing through the output buffer and A/D converter is not included.

Data Retention Characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 2.0 to 5.5 V, $BV_{DD} \leq V_{DD}$, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention voltage	V_{DDDR}	STOP mode	**Note**		5.5	V
Data retention current	I_{DDDR}	V_{DDDR} = 2.0 V		10	40	μA
Supply voltage rise time	t_{RVD}		1			μs
Supply voltage fall time	t_{FVD}		1			μs
Supply voltage retention time	t_{HVD}	After STOP mode released	0			ms
STOP release signal input time	t_{DREL}	After V_{DD} reaches 3.5 V (MIN.)	0			ms
Data retention input voltage, high	V_{IHDR}	All input pins	$0.9V_{DDDR}$		V_{DDDR}	V
Data retention input voltage, low	V_{ILDR}	All input pins	0		$0.1V_{DDDR}$	V

Note Voltage (V_{RAMH}) detected using the RAMS.RAMF bit

Caution Shifting to STOP mode and restoring from STOP mode must be performed within the rated operating range.

AC Characteristics

AC Test Input Measurement Points (V_{DD}, AV_{REF0}, EV_{DD}, BV_{DD})

AC Test Output Measurement Points

Load Conditions

Caution If the load capacitance exceeds 50 pF due to the circuit configuration, bring the load capacitance of the device to 50 pF or less by inserting a buffer or by some other means.

CLKOUT Output Timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, BV_{DD} = 3.0 to 5.5 V, $BV_{DD} \leq V_{DD}$, AV_{REF0} = 4.0 to 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Output cycle	t_{CYK}	<1>		50 ns	80 μs	
High-level width	t_{WKH}	<2>		t_{CYK}/2 – 20		ns
Low-level width	t_{WKL}	<3>		t_{CYK}/2 – 20		ns
Rise time	t_{KR}	<4>			20	ns
Fall time	t_{KF}	<5>			20	ns

Clock Timing

Bus Timing

(1) In multiplexed bus mode

(a) CLKOUT asynchronous: In multiplexed bus mode

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, BV_{DD} = 3.0 to 5.5 V, $BV_{DD} \leq V_{DD}$, AV_{REF0} = 4.0 to 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Address setup time (to ASTB↓)	t_{SAST}	<6>		0.5T – 20		ns
Address hold time (from ASTB↓)	t_{HSTA}	<7>		0.5T – 15		ns
Delay time from $\overline{RD}$↓ to address float	t_{FRDA}	<8>			16	ns
Data input setup time from address	t_{SAID}	<9>			(2 + n)T – 45	ns
Data input setup time from $\overline{RD}$↓	t_{SRID}	<10>			(1 + n)T – 35	ns
Delay time from ASTB↓ to $\overline{RD}$, $\overline{WRm}$↓	$t_{DSTRDWR}$	<11>		0.5T – 15		ns
Data input hold time (from $\overline{RD}$↑)	t_{HRDID}	<12>		0		ns
Address output time from $\overline{RD}$↑	t_{DRDA}	<13>		(1 + i)T – 15		ns
Delay time from $\overline{RD}$, $\overline{WRm}$↑ to ASTB↑	$t_{DRDWRST}$	<14>		0.5T – 15		ns
Delay time from $\overline{RD}$↑ to ASTB↓	t_{DRDST}	<15>		(1.5 + i)T – 15		ns
$\overline{RD}$, $\overline{WRm}$ low-level width	t_{WRDWRL}	<16>		(1 + n)T – 15		ns
ASTB high-level width	t_{WSTH}	<17>		T – 15		ns
Data output time from $\overline{WRm}$↓	t_{DWROD}	<18>			20	ns
Data output setup time (to $\overline{WRm}$↑)	t_{SODWR}	<19>		(1 + n)T – 30		ns
Data output hold time (from $\overline{WRm}$↑)	t_{HWROD}	<20>		T – 15		ns
$\overline{WAIT}$ setup time (to address)	t_{SAWT1}	<21>	n ≥ 1		1.5T – 55	ns
	t_{SAWT2}	<22>			(1.5 + n)T – 55	ns
$\overline{WAIT}$ hold time (from address)	t_{HAWT1}	<23>	n ≥ 1	(0.5 + n)T		ns
	t_{HAWT2}	<24>		(1.5 + n)T		ns
$\overline{WAIT}$ setup time (to ASTB↓)	t_{SSTWT1}	<25>	n ≥ 1		T – 45	ns
	t_{SSTWT2}	<26>			(1 + n)T – 45	ns
$\overline{WAIT}$ hold time (from ASTB↓)	t_{HSTWT1}	<27>	n ≥ 1	nT		ns
	t_{HSTWT2}	<28>		(1 + n)T		ns
$\overline{HLDRQ}$ high-level width	t_{WHQH}	<29>		T + 10		ns
$\overline{HLDAK}$ low-level width	t_{WHAL}	<30>		T – 25		ns
Delay time from $\overline{HLDAK}$↑ to bus output	t_{DHAC}	<31>		–3		ns
Delay time from $\overline{HLDRQ}$↓ to $\overline{HLDAK}$↓	t_{DHQHA1}	<32>			(2n + 7.5)T + 25	ns
Delay time from $\overline{HLDRQ}$↑ to $\overline{HLDAK}$↑	t_{DHQHA2}	<33>		0.5T	1.5T + 45	ns

Remarks 1. T = 1/f_{CPU} (f_{CPU}: CPU operating clock frequency)

2. n: Number of wait clocks inserted in the bus cycle
The sampling timing changes when a programmable wait is inserted.

3. m = 0, 1

4. i: Number of idle states inserted after a read cycle (0 or 1)

5. The values in the above specifications are values for when clocks with a 1:1 duty ratio are input from X1.

(b) CLKOUT synchronous: In multiplexed bus mode

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, BV_{DD} = 3.0 to 5.5 V, $BV_{DD} \leq V_{DD}$, AV_{REF0} = 4.0 to 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from CLKOUT↑ to address	t_{DKA}	<34>		0	29	ns
Delay time from CLKOUT↑ to address float	t_{FKA}	<35>		0	29	ns
Delay time from CLKOUT↓ to ASTB	t_{DKST}	<36>		–12	15	ns
Delay time from CLKOUT↑ to $\overline{RD}$, $\overline{WRm}$	t_{DKRDWR}	<37>		–5	14	ns
Data input setup time (to CLKOUT↑)	t_{SIDK}	<38>		25		ns
Data input hold time (from CLKOUT↑)	t_{HKID}	<39>		5		ns
Delay time from CLKOUT↑ to data output	t_{DKOD}	<40>			27	ns
$\overline{WAIT}$ setup time (to CLKOUT↓)	t_{SWTK}	<41>		40		ns
$\overline{WAIT}$ hold time (from CLKOUT↓)	t_{HKWT}	<42>		5		ns
$\overline{HLDRQ}$ setup time (to CLKOUT↓)	t_{SHQK}	<43>		40		ns
$\overline{HLDRQ}$ hold time (from CLKOUT↓)	t_{HKHQ}	<44>		5		ns
Delay time from CLKOUT↑ to bus float	t_{DKF}	<45>			29	ns
Delay time from CLKOUT↑ to $\overline{HLDAK}$	t_{DKHA}	<46>			29	ns

Remarks 1. m = 0, 1

2. The values in the above specifications are values for when clocks with a 1:1 duty ratio are input from X1.

Read Cycle (CLKOUT Synchronous/Asynchronous, 1 Wait): In Multiplexed Bus Mode

Remark $\overline{\text{WR0}}$ and $\overline{\text{WR1}}$ are high level.

Write Cycle (CLKOUT Synchronous/Asynchronous, 1 Wait): In Multiplexed Bus Mode

Remark $\overline{\text{RD}}$ is high level.

Bus Hold: In Multiplexed Bus Mode

Reset, Interrupt, FLMD0 Pin Timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, BV_{DD} = 3.0 to 5.5 V, $BV_{DD} \leq V_{DD}$, AV_{REF0} = 4.0 to 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
$\overline{\text{RESET}}$ low-level width	t_{WRSL1}		500		ns
NMI high-level width	t_{WNIH}	Analog noise elimination	500		ns
NMI low-level width	t_{WNIL}	Analog noise elimination	500		ns
INTPn high-level width	t_{WITH}	n = 0 to 7 (Analog noise elimination)	500		ns
		n = 3 (Digital noise elimination)	$2T_{smp}$ + 20/ $3T_{smp}$ + 20		ns
INTPn low-level width	t_{WITL}	n = 0 to 7 (Analog noise elimination)	500		ns
		n = 3 (Digital noise elimination)	$2T_{smp}$ + 20/ $3T_{smp}$ + 20		ns
FLMD0 high-level width	t_{WMDH}		500		ns
FLMD0 low-level width	t_{WMDL}		500		ns

Remark T_{smp}: Noise elimination sampling clock cycle

Key Return Timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, BV_{DD} = 3.0 to 5.5 V, $BV_{DD} \leq V_{DD}$, AV_{REF0} = 4.0 to 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
KRn high-level width	t_{WKRH}	Analog noise elimination (n = 0 to 7)	500		ns
KRn low-level width	t_{WKRL}	Analog noise elimination (n = 0 to 7)	500		ns

Timer Timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, BV_{DD} = 3.0 to 5.5 V, $BV_{DD} \leq V_{DD}$, AV_{REF0} = 4.0 to 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
TI high-level width	t_{TIH}	TIP00, TIP01, TIP10, TIP11, TIP20, TIP21, TIP30, TIP31, TIQ00 to TIQ03	$2T_{smp}$ + 20/ $3T_{smp}$ + 20		ns
TI low-level width	t_{TIL}	TIP00, TIP01, TIP10, TIP11, TIP20, TIP21, TIP30, TIP31, TIQ00 to TIQ03	$2T_{smp}$ + 20/ $3T_{smp}$ + 20		ns

Remark T_{smp}: Noise elimination sampling clock cycle

UART Timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, BV_{DD} = 3.0 to 5.5 V, $BV_{DD} \leq V_{DD}$, AV_{REF0} = 4.0 to 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Transmit rate				312.5	kbps
ASCK0 cycle time				10	MHz

CSIB Timing

(1) Master mode

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, BV_{DD} = 3.0 to 5.5 V, $BV_{DD} \leq V_{DD}$, AV_{REF0} = 4.0 to 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{SCKBn}}$ cycle time	t_{KCY1}	<47>		125		ns
$\overline{\text{SCKBn}}$ high-/low-level width	t_{KH1}, t_{KL1}	<48>		$t_{KCY1}/2 - 15$		ns
SIBn setup time (to $\overline{\text{SCKBn}}\uparrow$)	t_{SIK1}	<49>		30		ns
SIBn hold time (from $\overline{\text{SCKBn}}\uparrow$)	t_{KSI1}	<50>		25		ns
Delay time from $\overline{\text{SCKBn}}\downarrow$ to SOBn output	t_{KSO1}	<51>			25	ns

Remark n = 0, 1

(2) Slave mode

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 4.0 to 5.5 V, BV_{DD} = 4.0 to 5.5 V, $BV_{DD} \leq V_{DD}$, AV_{REF0} = 4.0 to 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{SCKBn}}$ cycle time	t_{KCY2}	<47>		125		ns
$\overline{\text{SCKBn}}$ high-/low-level width	t_{KH2}, t_{KL2}	<48>		57.5		ns
SIBn setup time (to $\overline{\text{SCKBn}}\uparrow$)	t_{SIK2}	<49>		30		ns
SIBn hold time (from $\overline{\text{SCKBn}}\uparrow$)	t_{KSI2}	<50>		30		ns
Delay time from $\overline{\text{SCKBn}}\downarrow$ to SOBn output	t_{KSO2}	<51>			30	ns

Remark n = 0, 1

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, BV_{DD} = 3.0 to 5.5 V, $BV_{DD} \leq V_{DD}$, AV_{REF0} = 4.0 to 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{SCKBn}}$ cycle time	t_{KCY2}	<47>		200		ns
$\overline{\text{SCKBn}}$ high-/low-level width	t_{KH2}, t_{KL2}	<48>		90		ns
SIBn setup time (to $\overline{\text{SCKBn}}\uparrow$)	t_{SIK2}	<49>		50		ns
SIBn hold time (from $\overline{\text{SCKBn}}\uparrow$)	t_{KSI2}	<50>		50		ns
Delay time from $\overline{\text{SCKBn}}\downarrow$ to SOBn output	t_{KSO2}	<51>			50	ns

Remark n = 0, 1

CSIB Timing

Remark n = 0, 1

I^2C Bus Mode

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, BV_{DD} = 3.0 to 5.5 V, $BV_{DD} \leq V_{DD}$, AV_{REF0} = 4.0 to 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter		Symbol		Standard Mode		High-Speed Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCL00 clock frequency		f_{CLK}		0	100	0	400	kHz
Bus free time (between start and stop conditions)		t_{BUF}	<52>	4.7	–	1.3	–	μs
Hold time[Note 1]		$t_{HD: STA}$	<53>	4.0	–	0.6	–	μs
SCL00 clock low-level width		t_{LOW}	<54>	4.7	–	1.3	–	μs
SCL00 clock high-level width		t_{HIGH}	<55>	4.0	–	0.6	–	μs
Setup time for start/restart conditions		$t_{SU: STA}$	<56>	4.7	–	0.6	–	μs
Data hold time	CBUS compatible master	$t_{HD: DAT}$	<57>	5.0	–	–	–	μs
	I^2C mode			0[Note 2]	–	0[Note 2]	0.9[Note 3]	μs
Data setup time		$t_{SU: DAT}$	<58>	250	–	100[Note 4]	–	ns
SDA00 and SCL00 signal rise time		t_R	<59>	–	1000	20 + 0.1Cb[Note 5]	300	ns
SDA00 and SCL00 signal fall time		t_F	<60>	–	300	20 + 0.1Cb[Note 5]	300	ns
Stop condition setup time		$t_{SU: STO}$	<61>	4.0	–	0.6	–	μs
Pulse width of spike suppressed by input filter		t_{SP}	<62>	–	–	0	50	ns
Capacitance load of each bus line		Cb		–	400	–	400	pF

Notes 1. At the start condition, the first clock pulse is generated after the hold time.

2. The system requires a minimum of 300 ns hold time internally for the SDA00 signal (at $V_{IHmin.}$ of SCL00 signal) in order to occupy the undefined area at the falling edge of SCL00.

3. If the system does not extend the SCL00 signal low hold time (t_{LOW}), only the maximum data hold time ($t_{HD:DAT}$) needs to be satisfied.

4. The high-speed mode I^2C bus can be used in the standard-mode I^2C bus system. In this case, set the high-speed mode I^2C bus so that it meets the following conditions.

- If the system does not extend the SCL00 signal's low state hold time: $t_{SU:DAT} \geq$ 250 ns
- If the system extends the SCL00 signal's low state hold time:
  Transmit the following data bit to the SDA00 line prior to the SCL00 line release ($t_{Rmax.}$ + $t_{SU:DAT}$ = 1,000 + 250 = 1,250 ns: Standard mode I^2C bus specification).

5. Cb: Total capacitance of one bus line (unit: pF)

I²C Bus Mode

SCL0 (I/O)
SDA0 (I/O)
<54>
<55>
<60>
<59>
<57>
<58>
<56>
<53>
<62>
<61>
<52>
<59>
<60>
Stop condition
Start condition
Restart condition
Stop condition

A/D Converter

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, BV_{DD} = 3.0 to 5.5 V, $BV_{DD} \leq V_{DD}$, AV_{REF0} = 4.0 to 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution					10	bit
Overall error[Note]		$4.0 \leq AV_{REF0} \leq 5.5$ V			±0.35	%FSR
Conversion time	t_{CONV}		2.91		16	μs
Analog input voltage	V_{IAN}		AV_{SS}		AV_{REF0}	V
AV_{REF0} current	AI_{REF0}	High-speed conversion mode		5	10	mA
		When A/D converter unused		1	10	μA

Note Excluding quantization error (±0.05 %FSR).

Caution Do not set (read/write) alternate-function ports during A/D conversion; otherwise the conversion resolution may be degraded.

Remark FSR: Full Scale Range

LVI Circuit Characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, BV_{DD} = 3.0 to 5.5 V, $BV_{DD} \leq V_{DD}$, AV_{REF0} = 4.0 to 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{LVI0}		4.2	4.4	4.6	V
	V_{LVI1}		4.0	4.2	4.2	V
Response time[Note]	t_{LD}	After V_{DD} reaches V_{LVI0}/V_{LVI1} (MAX.), or after V_{DD} has dropped to V_{LVI0}/V_{LVI1} (MIN.)		0.2	2.0	ms
Minimum pulse width	t_{LW}		0.2			ms
Reference voltage stabilization wait time	t_{LWAIT}	After V_{DD} reaches 3.5 V (MIN.)		0.1	0.2	ms

Note Time required to detect the detection voltage and output an interrupt or $\overline{\text{RESET}}$ signal.

RAM Retention Detection

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 1.9 to 5.5 V, AV_{REF0} = 4.0 to 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{RAMH}		1.9	2.0	2.1	V
Supply voltage rise time	t_{RAMHTH}	V_{DD} = 0 V → 3.5 V	0.002		1800	ms
Response time[Note]	t_{RAMHD}	After V_{DD} reaches 2.1 V (MIN.)		0.2	2.0	ms
Minimum pulse width	t_{RAMHW}		0.2			ms

Note Time required to detect the detection voltage and set the RAMS.RAMF bit.

Flash Memory Programming Characteristics

($BV_{DD} \leq V_{DD} = EV_{DD}$, AV_{REF0} = 4.0 to 5.5 V, $V_{SS} = EV_{SS} = BV_{SS} = AV_{SS}$ = 0 V, C_L = 50 pF)

(1) Basic characteristics

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Operating frequency	f_{CPU}		4		20	MHz
Supply voltage	V_{DD}		3.5		5.5	V
Number of rewrites	C_{WRT}				100	Times
Input voltage, high	V_{IH}	FLMD0	$0.8EV_{DD}$		EV_{DD}	V
Input voltage, low	V_{IL}	FLMD0	EV_{SS}		$0.2EV_{SS}$	V
Write time + erase time	t_{IWRT} + t_{ERASE}				T. B. D	s
Programming temperature	t_{PRG}		–40		+85	°C

(2) Serial write operation characteristics

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
FLMD0 count start time from RESET↑	t_{RFCF}		$2^{16}/f_X + \alpha$			s
Count execution time	t_{COUNT}				T. B. D	ms
FLMD0 counter high-level width/ low-level width	t_{CH}/t_{CL}		1			μs
FLMD0 counter rise time/fall time	t_R/t_F				T. B. D	μs

Flash write mode set timing

100-PIN PLASTIC LQFP (FINE PITCH) (14x14)

NOTE

Each lead centerline is located within 0.08 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	16.00±0.20
B	14.00±0.20
C	14.00±0.20
D	16.00±0.20
F	1.00
G	1.00
H	$0.22^{+0.05}_{-0.04}$
I	0.08
J	0.50 (T.P.)
K	1.00±0.20
L	0.50±0.20
M	$0.17^{+0.03}_{-0.07}$
N	0.08
P	1.40±0.05
Q	0.10±0.05
R	$3^{\circ}{}^{+7^{\circ}}_{-3^{\circ}}$
S	1.60 MAX.

S100GC-50-8EU, 8EA-2

★

APPENDIX A REGISTER INDEX

(1/8)

Symbol	Name	Unit	Page
ADA0CR0	A/D conversion result register 0	ADC	411
ADA0CR0H	A/D conversion result register 0H	ADC	411
ADA0CR1	A/D conversion result register 1	ADC	411
ADA0CR1H	A/D conversion result register 1H	ADC	411
ADA0CR2	A/D conversion result register 2	ADC	411
ADA0CR2H	A/D conversion result register 2H	ADC	411
ADA0CR3	A/D conversion result register 3	ADC	411
ADA0CR3H	A/D conversion result register 3H	ADC	411
ADA0CR4	A/D conversion result register 4	ADC	411
ADA0CR4H	A/D conversion result register 4H	ADC	411
ADA0CR5	A/D conversion result register 5	ADC	411
ADA0CR5H	A/D conversion result register 5H	ADC	411
ADA0CR6	A/D conversion result register 6	ADC	411
ADA0CR6H	A/D conversion result register 6H	ADC	411
ADA0CR7	A/D conversion result register 7	ADC	411
ADA0CR7H	A/D conversion result register 7H	ADC	411
ADA0CR8	A/D conversion result register 8	ADC	411
ADA0CR8H	A/D conversion result register 8H	ADC	411
ADA0CR9	A/D conversion result register 9	ADC	411
ADA0CR9H	A/D conversion result register 9H	ADC	411
ADA0CR10	A/D conversion result register 10	ADC	411
ADA0CR10H	A/D conversion result register 10H	ADC	411
ADA0CR11	A/D conversion result register 11	ADC	411
ADA0CR11H	A/D conversion result register 11H	ADC	411
ADA0M0	A/D converter mode register 0	ADC	406
ADA0M1	A/D converter mode register 1	ADC	407
ADA0M2	A/D converter mode register 2	ADC	409
ADA0PFM	Power-fail compare mode register	ADC	413
ADA0PFT	Power-fail compare threshold value register	ADC	413
ADA0S	A/D converter channel specification register 0	ADC	410
ADIC	Interrupt control register	INTC	596
AWC	Address wait control register	BCU	166
BCC	Bus cycle control register	BCU	167
BSC	Bus size configuration register	BCU	156
CB0CTL0	CSIB0 control register 0	CSI	465
CB0CTL1	CSIB0 control register 1	CSI	467
CB0CTL2	CSIB0 control register 2	CSI	468
CB0RIC	Interrupt control register	INTC	596
CB0RX	CSIB0 receive data register	CSI	464
CB0RXL	CSIB0 receive data register L	CSI	464
CB0STR	CSIB0 status register	CSI	470
CB0TIC	Interrupt control register	INTC	596

(2/8)

Symbol	Name	Unit	Page
CB0TX	CSIB0 transmit data register	CSI	464
CB0TXL	CSIB0 transmit data register L	CSI	464
CB1CTL0	CSIB1 control register 0	CSI	465
CB1CTL1	CSIB1 control register 1	CSI	467
CB1CTL2	CSIB1 control register 2	CSI	468
CB1RIC	Interrupt control register	INTC	596
CB1RX	CSIB1 receive data register	CSI	464
CB1RXL	CSIB1 receive data register L	CSI	464
CB1STR	CSIB1 status register	CSI	470
CB1TIC	Interrupt control register	INTC	596
CB1TX	CSIB1 transmit data register	CSI	464
CB1TXL	CSIB1 transmit data register L	CSI	464
CCLS	CPU operation clock status register	CG	181
CLM	Clock monitor mode register	CLM	650
CORAD0	Correction address register 0	ROMC	655
CORAD0H	Correction address register 0H	ROMC	655
CORAD0L	Correction address register 0L	ROMC	655
CORAD1	Correction address register 1	ROMC	655
CORAD1H	Correction address register 1H	ROMC	655
CORAD1L	Correction address register 1L	ROMC	655
CORAD2	Correction address register 2	ROMC	655
CORAD2H	Correction address register 2H	ROMC	655
CORAD2L	Correction address register 2L	ROMC	655
CORAD3	Correction address register 3	ROMC	655
CORAD3H	Correction address register 3H	ROMC	655
CORAD3L	Correction address register 3L	ROMC	655
CORCN	Correction control register	ROMC	656
DADC0	DMA addressing control register 0	DMA	561
DADC1	DMA addressing control register 1	DMA	561
DADC2	DMA addressing control register 2	DMA	561
DADC3	DMA addressing control register 3	DMA	561
DBC0	DMA byte count register 0	DMA	560
DBC1	DMA byte count register 1	DMA	560
DBC2	DMA byte count register 2	DMA	560
DBC3	DMA byte count register 3	DMA	560
DCHC0	DMA channel control register 0	DMA	562
DCHC1	DMA channel control register 1	DMA	562
DCHC2	DMA channel control register 2	DMA	562
DCHC3	DMA channel control register 3	DMA	562
DDA0H	DMA destination address register 0H	DMA	559
DDA0L	DMA destination address register 0L	DMA	559
DDA1H	DMA destination address register 1H	DMA	559
DDA1L	DMA destination address register 1L	DMA	559
DDA2H	DMA destination address register 2H	DMA	559
DDA2L	DMA destination address register 2L	DMA	559

(3/8)

Symbol	Name	Unit	Page
DDA3H	DMA destination address register 3H	DMA	559
DDA3L	DMA destination address register 3L	DMA	559
DMAIC0	Interrupt control register	INTC	596
DMAIC1	Interrupt control register	INTC	596
DMAIC2	Interrupt control register	INTC	596
DMAIC3	Interrupt control register	INTC	596
DSA0H	DMA source address register 0H	DMA	558
DSA0L	DMA source address register 0L	DMA	558
DSA1H	DMA source address register 1H	DMA	558
DSA1L	DMA source address register 1L	DMA	558
DSA2H	DMA source address register 2H	DMA	558
DSA2L	DMA source address register 2L	DMA	558
DSA3H	DMA source address register 3H	DMA	558
DSA3L	DMA source address register 3L	DMA	558
DTFR0	DMA trigger factor register 0	DMA	563
DTFR1	DMA trigger factor register 1	DMA	563
DTFR2	DMA trigger factor register 2	DMA	563
DTFR3	DMA trigger factor register 3	DMA	563
DWC0	Data wait control register 0	BCU	164
IIC0	IIC shift register 0	I^2C	508
IICC0	IIC control register 0	I^2C	495
IICCL0	IIC clock select register 0	I^2C	505
IICF0	IIC flag register 0	I^2C	503
IICIC0	Interrupt control register	INTC	596
IICS0	IIC status register 0	I^2C	500
IICX0	IIC function expansion register 0	I^2C	506
IMR0	Interrupt mask register 0	INTC	599
IMR0H	Interrupt mask register 0H	INTC	599
IMR0L	Interrupt mask register 0L	INTC	599
IMR1	Interrupt mask register 1	INTC	599
IMR1H	Interrupt mask register 1H	INTC	599
IMR1L	Interrupt mask register 1L	INTC	599
IMR2	Interrupt mask register 2	INTC	599
IMR2H	Interrupt mask register 2H	INTC	599
IMR2L	Interrupt mask register 2L	INTC	599
IMR3	Interrupt mask register 3	INTC	599
IMR3H	Interrupt mask register 3H	INTC	599
IMR3L	Interrupt mask register 3L	INTC	599
INTF0	External interrupt falling edge specification register 0	INTC	602
INTF3L	External interrupt falling edge specification register 3L	INTC	603
INTF9H	External interrupt falling edge specification register 9H	INTC	604
INTR0	External interrupt rising edge specification register 0	INTC	602
INTR3L	External interrupt rising edge specification register 3L	INTC	603
INTR9H	External interrupt rising edge specification register 9H	INTC	604
ISPR	In-service priority register	INTC	600

(4/8)

Symbol	Name	Unit	Page
KRIC	Interrupt control register	INTC	596
KRM	Key return mode register	KR	617
LOCKR	Lock register	CG	184
LVIIC	Interrupt control register	INTC	596
LVIM	Low-voltage detection register	LVI	647
LVIS	Low-voltage detection level select register	LVI	648
NFC	Noise elimination control register	INTC	605
OCDM	On-chip debug mode register	Debug	684
OCKS0	IIC division clock select register 0	I^2C	508
OSTS	Oscillation stabilization time select register	Standby	623
P0	Port 0 register	Port	79
P00NFC	TIP00 noise elimination control register	Timer	272
P01NFC	TIP01 noise elimination control register	Timer	272
P1	Port 1 register	Port	83
P10NFC	TIP10 noise elimination control register	Timer	272
P11NFC	TIP11 noise elimination control register	Timer	272
P20NFC	TIP20 noise elimination control register	Timer	272
P21NFC	TIP21 noise elimination control register	Timer	272
P3	Port 3 register	Port	84
P30NFC	TIP30 noise elimination control register	Timer	272
P31NFC	TIP31 noise elimination control register	Timer	272
P3H	Port 3 register H	Port	84
P3L	Port 3 register L	Port	84
P4	Port 4 register	Port	89
P5	Port 5 register	Port	92
P7H	Port 7 register H	Port	98
P7L	Port 7 register L	Port	98
P9	Port 9 register	Port	100
P9H	Port 9 register H	Port	100
P9L	Port 9 register L	Port	100
PCC	Processor clock control register	CG	177
PCM	Port CM register	Port	106
PCS	Port CS register	Port	109
PCT	Port CT register	Port	111
PDH	Port DH register	Port	113
PDL	Port DL register	Port	115
PDLH	Port DL register H	Port	115
PDLL	Port DL register L	Port	115
PF4	Port 4 function register	Port	91
PFC0	Port 0 function control register	Port	81
PFC3L	Port 3 function control register L	Port	87
PFC4	Port 4 function control register	Port	90
PFC5	Port 5 function control register	Port	94
PFC9	Port 9 function control register	Port	102

(5/8)

Symbol	Name	Unit	Page
PFC9H	Port 9 function control register H	Port	102
PFC9L	Port 9 function control register L	Port	102
PFCE3L	Port 3 function control expansion register L	Port	87
PFCE5	Port 5 function control expansion register	Port	94
PFCE9L	Port 9 function control expansion register L	Port	102
PIC0	Interrupt control register	INTC	596
PIC1	Interrupt control register	INTC	596
PIC2	Interrupt control register	INTC	596
PIC3	Interrupt control register	INTC	596
PIC4	Interrupt control register	INTC	596
PIC5	Interrupt control register	INTC	596
PIC6	Interrupt control register	INTC	596
PIC7	Interrupt control register	INTC	596
PLLCTL	PLL control register	CG	183
PLLS	PLL lockup time specification register	CG	185
PM0	Port 0 mode register	Port	79
PM1	Port 1 mode register	Port	83
PM3	Port 3 mode register	Port	85
PM3H	Port 3 mode register H	Port	85
PM3L	Port 3 mode register L	Port	85
PM4	Port 4 mode register	Port	89
PM5	Port 5 mode register	Port	93
PM7H	Port 7 mode register H	Port	98
PM7L	Port 7 mode register L	Port	98
PM9	Port 9 mode register	Port	100
PM9H	Port 9 mode register H	Port	100
PM9L	Port 9 mode register L	Port	100
PMC0	Port 0 mode control register	Port	80
PMC3	Port 3 mode control register	Port	85
PMC3H	Port 3 mode control register H	Port	85
PMC3L	Port 3 mode control register L	Port	85
PMC4	Port 4 mode control register	Port	90
PMC5	Port 5 mode control register	Port	93
PMC9	Port 9 mode control register	Port	101
PMC9H	Port 9 mode control register H	Port	101
PMC9L	Port 9 mode control register L	Port	101
PMCCM	Port CM mode control register	Port	107
PMCCS	Port CS mode control register	Port	109
PMCCT	Port CT mode control register	Port	112
PMCDH	Port DH mode control register	Port	113
PMCDL	Port DL mode control register	Port	116
PMCDLH	Port DL mode control register H	Port	116
PMCDLL	Port DL mode control register L	Port	116
PMCM	Port CM mode register	Port	106

(6/8)

Symbol	Name	Unit	Page
PMCS	Port CS mode register	Port	109
PMCT	Port CT mode register	Port	111
PMDH	Port DH mode register	Port	113
PMDL	Port DL mode register	Port	116
PMDLH	Port DL mode register H	Port	116
PMDLL	Port DL mode register L	Port	116
PRCMD	Command register	CPU	67
PRSCM0	Prescaler compare register 0	WT	389, 489
PRSM0	Prescaler mode register 0	WT	388, 488
PSC	Power save control register	Standby	621
PSMR	Power save mode register	Standby	622
PU0	Pull-up resistor option register 0	Port	81
PU1	Pull-up resistor option register 1	Port	83
PU3	Pull-up resistor option register 3	Port	88
PU3H	Pull-up resistor option register 3H	Port	88
PU3L	Pull-up resistor option register 3L	Port	88
PU4	Pull-up resistor option register 4	Port	91
PU5	Pull-up resistor option register 5	Port	96
PU9	Pull-up resistor option register 9	Port	104
PU9H	Pull-up resistor option register 9H	Port	104
PU9L	Pull-up resistor option register 9L	Port	104
Q00NFC	TIQ00 noise elimination control register	Timer	375
Q01NFC	TIQ01 noise elimination control register	Timer	375
Q02NFC	TIQ02 noise elimination control register	Timer	375
Q03NFC	TIQ03 noise elimination control register	Timer	375
RAMS	Internal RAM data status register	Reset	648
RCM	Ring-OSC mode register	CG	181
RESF	Reset source flag register	Reset	638
SELCNT0	Selector operation control register 0	Timer	276
SVA0	Slave address register 0	I^2C	508
SYS	System status register	CPU	68
TM0CMP0	TMM0 compare register 0	Timer	378
TM0CTL0	TMM0 timer control register 0	Timer	379
TM0EQIC0	Interrupt control register	INTC	596
TP0CCIC0	Interrupt control register	INTC	596
TP0CCIC1	Interrupt control register	INTC	596
TP0CCR0	TMP0 capture/compare register 0	Timer	196
TP0CCR1	TMP0 capture/compare register 1	Timer	198
TP0CNT	TMP0 counter read buffer register	Timer	200
TP0CTL0	TMP0 control register 0	Timer	190
TP0CTL1	TMP0 control register 1	Timer	191
TP0IOC0	TMP0 I/O control register 0	Timer	192
TP0IOC1	TMP0 I/O control register 1	Timer	193
TP0IOC2	TMP0 I/O control register 2	Timer	194

(7/8)

Symbol	Name	Unit	Page
TP0OPT0	TMP0 option register 0	Timer	195
TP0OVIC	Interrupt control register	INTC	596
TP1CCIC0	Interrupt control register	INTC	596
TP1CCIC1	Interrupt control register	INTC	596
TP1CCR0	TMP1 capture/compare register 0	Timer	196
TP1CCR1	TMP1 capture/compare register 1	Timer	198
TP1CNT	TMP1 counter read buffer register	Timer	200
TP1CTL0	TMP1 control register 0	Timer	190
TP1CTL1	TMP1 control register 1	Timer	191
TP1IOC0	TMP1 I/O control register 0	Timer	192
TP1IOC1	TMP1 I/O control register 1	Timer	193
TP1IOC2	TMP1 I/O control register 2	Timer	194
TP1OPT0	TMP1 option register 0	Timer	195
TP1OVIC	Interrupt control register	INTC	596
TP2CCIC0	Interrupt control register	INTC	596
TP2CCIC1	Interrupt control register	INTC	596
TP2CCR0	TMP2 capture/compare register 0	Timer	196
TP2CCR1	TMP2 capture/compare register 1	Timer	198
TP2CNT	TMP2 counter read buffer register	Timer	200
TP2CTL0	TMP2 control register 0	Timer	190
TP2CTL1	TMP2 control register 1	Timer	191
TP2IOC0	TMP2 I/O control register 0	Timer	192
TP2IOC1	TMP2 I/O control register 1	Timer	193
TP2IOC2	TMP2 I/O control register 2	Timer	194
TP2OPT0	TMP2 option register 0	Timer	195
TP2OVIC	Interrupt control register	INTC	596
TP3CCIC0	Interrupt control register	INTC	596
TP3CCIC1	Interrupt control register	INTC	596
TP3CCR0	TMP3 capture/compare register 0	Timer	196
TP3CCR1	TMP3 capture/compare register 1	Timer	198
TP3CNT	TMP3 counter read buffer register	Timer	200
TP3CTL0	TMP3 control register 0	Timer	190
TP3CTL1	TMP3 control register 1	Timer	191
TP3IOC0	TMP3 I/O control register 0	Timer	192
TP3IOC1	TMP3 I/O control register 1	Timer	193
TP3IOC2	TMP3 I/O control register 2	Timer	194
TP3OPT0	TMP3 option register 0	Timer	195
TP3OVIC	Interrupt control register	INTC	596
TQ0CCIC0	Interrupt control register	INTC	596
TQ0CCIC1	Interrupt control register	INTC	596
TQ0CCIC2	Interrupt control register	INTC	596
TQ0CCIC3	Interrupt control register	INTC	596
TQ0CCR0	TMQ0 capture/compare register 0	Timer	287
TQ0CCR1	TMQ0 capture/compare register 1	Timer	289

(8/8)

Symbol	Name	Unit	Page
TQ0CCR2	TMQ0 capture/compare register 2	Timer	291
TQ0CCR3	TMQ0 capture/compare register 3	Timer	293
TQ0CNT	TMQ0 counter read buffer register	Timer	295
TQ0CTL0	TMQ0 control register 0	Timer	281
TQ0CTL1	TMQ0 control register 1	Timer	282
TQ0IOC0	TMQ0 I/O control register 0	Timer	283
TQ0IOC1	TMQ0 I/O control register 1	Timer	284
TQ0IOC2	TMQ0 I/O control register 2	Timer	285
TQ0OPT0	TMQ0 option register 0	Timer	286
TQ0OVIC	Interrupt control register	INTC	596
UA0CTL0	UARTA0 control register 0	UART	433
UA0CTL1	UARTA0 control register 1	UART	455
UA0CTL2	UARTA0 control register 2	UART	456
UA0OPT0	UARTA0 option control register 0	UART	435
UA0RIC	Interrupt control register	INTC	596
UA0RX	UARTA0 receive data register	UART	439
UA0STR	UARTA0 status register	UART	437
UA0TIC	Interrupt control register	INTC	596
UA0TX	UARTA0 transmit data register	UART	439
UA1CTL0	UARTA1 control register 0	UART	433
UA1CTL1	UARTA1 control register 1	UART	455
UA1CTL2	UARTA1 control register 2	UART	456
UA1OPT0	UARTA1 option control register 0	UART	435
UA1RIC	Interrupt control register	INTC	596
UA1RX	UARTA1 receive data register	UART	439
UA1STR	UARTA1 status register	UART	437
UA1TIC	Interrupt control register	INTC	596
UA1TX	UARTA1 transmit data register	UART	439
UA2CTL0	UARTA2 control register 0	UART	433
UA2CTL1	UARTA2 control register 1	UART	455
UA2CTL2	UARTA2 control register 2	UART	456
UA2OPT0	UARTA2 option control register 0	UART	435
UA2RIC	Interrupt control register	INTC	596
UA2RX	UARTA2 receive data register	UART	439
UA2STR	UARTA2 status register	UART	437
UA2TIC	Interrupt control register	INTC	596
UA2TX	UARTA2 transmit data register	UART	439
VSWC	System wait control register	CPU	69
WDTE	Watchdog timer enable register	WDT	399
WDTM2	Watchdog timer mode register 2	WDT	397
WTIC	Interrupt control register	INTC	596
WTIIC	Interrupt control register	INTC	596
WTM	Watch timer operation mode register	WT	390

★ APPENDIX B INSTRUCTION SET LIST

B.1 Conventions

(1) Register symbols used to describe operands

Register Symbol	Explanation
reg1	General-purpose register: Used as source register.
reg2	General-purpose register: Used mainly as destination register. Also used as source register in some instructions.
reg3	General-purpose register: Used mainly to store the remainder of division result and the higher 32 bits of multiplication result.
bit#3	3-bit data for specifying the bit number
immX	X bit immediate data
dispX	X bit displacement data
regID	System register number
vector	5-bit data that specifies the trap vector (00H to 1FH)
cccc	4-bit data that shows the condition codes
sp	Stack pointer (r3)
ep	Element pointer (r30)
listX	X item register list

(2) Register symbols used to describe opcodes

Register Symbol	Explanation
R	1-bit data of a code that specifies reg1 or regID
r	1-bit data of the code that specifies reg2
w	1-bit data of the code that specifies reg3
d	1-bit displacement data
I	1-bit immediate data (indicates the higher bits of immediate data)
i	1-bit immediate data
cccc	4-bit data that shows the condition codes
CCCC	4-bit data that shows the condition codes of Bcond instruction
bbb	3-bit data for specifying the bit number
L	1-bit data that specifies a program register in the register list

(3) Register symbols used in operations

Register Symbol	Explanation
←	Input for
GR [ ]	General-purpose register
SR [ ]	System register
zero-extend (n)	Expand n with zeros until word length.
sign-extend (n)	Expand n with signs until word length.
load-memory (a, b)	Read size b data from address a.
store-memory (a, b, c)	Write data b into address a in size c.
load-memory-bit (a, b)	Read bit b of address a.
store-memory-bit (a, b, c)	Write c to bit b of address a.
saturated (n)	Execute saturated processing of n (n is a 2's complement). If, as a result of calculations, n ≥ 7FFFFFFFH, let it be 7FFFFFFFH. n ≤ 80000000H, let it be 80000000H.
result	Reflects the results in a flag.
Byte	Byte (8 bits)
Halfword	Halfword (16 bits)
Word	Word (32 bits)
+	Addition
–	Subtraction
ll	Bit concatenation
×	Multiplication
÷	Division
%	Remainder from division results
AND	Logical product
OR	Logical sum
XOR	Exclusive OR
NOT	Logical negation
logically shift left by	Logical shift left
logically shift right by	Logical shift right
arithmetically shift right by	Arithmetic shift right

(4) Register symbols used in execution clock

Register Symbol	Explanation
i	If executing another instruction immediately after executing the first instruction (issue).
r	If repeating execution of the same instruction immediately after executing the first instruction (repeat).
l	If using the results of instruction execution in the instruction immediately after the execution (latency).

(5) Register symbols used in flag operations

Register Symbol	Explanation
(Blank)	No change
0	Clear to 0
×	Set or cleared in accordance with the results.
R	Previously saved values are restored.

(6) Condition codes

Condition Name (cond)	Condition Code (cccc)	Condition Formula	Explanation
V	0 0 0 0	OV = 1	Overflow
NV	1 0 0 0	OV = 0	No overflow
C/L	0 0 0 1	CY = 1	Carry Lower (Less than)
NC/NL	1 0 0 1	CY = 0	No carry Not lower (Greater than or equal)
Z	0 0 1 0	Z = 1	Zero
NZ	1 0 1 0	Z = 0	Not zero
NH	0 0 1 1	(CY or Z) = 1	Not higher (Less than or equal)
H	1 0 1 1	(CY or Z) = 0	Higher (Greater than)
S/N	0 1 0 0	S = 1	Negative
NS/P	1 1 0 0	S = 0	Positive
T	0 1 0 1	–	Always (Unconditional)
SA	1 1 0 1	SAT = 1	Saturated
LT	0 1 1 0	(S xor OV) = 1	Less than signed
GE	1 1 1 0	(S xor OV) = 0	Greater than or equal signed
LE	0 1 1 1	((S xor OV) or Z) = 1	Less than or equal signed
GT	1 1 1 1	((S xor OV) or Z) = 0	Greater than signed

B.2 Instruction Set (in Alphabetical Order)

(1/6)

Mnemonic	Operand	Opcode	Operation		Execution Clock			Flags				
					i	r	l	CY	OV	S	Z	SAT
ADD	reg1,reg2	rrrrr001110RRRRR	GR[reg2]←GR[reg2]+GR[reg1]		1	1	1	×	×	×	×	
	imm5,reg2	r r r r r 0 1 0 0 1 0 i i i i i	GR[reg2]←GR[reg2]+sign-extend(imm5)		1	1	1	×	×	×	×	
ADDI	imm16,reg1,reg2	rrrrr110000RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1]+sign-extend(imm16)		1	1	1	×	×	×	×	
AND	reg1,reg2	rrrrr001010RRRRR	GR[reg2]←GR[reg2]AND GR[reg1]		1	1	1		0	×	×	
ANDI	imm16,reg1,reg2	rrrrr110110RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1]AND zero-extend(imm16)		1	1	1		0	×	×	
Bcond	disp9	ddddd1011dddcccc **Note 1**	if conditions are satisfied then PC←PC+sign-extend(disp9)	When conditions are satisfied	2 Note 2	2 Note 2	2 Note 2					
				When conditions are not satisfied	1	1	1					
BSH	reg2,reg3	rrrrr11111100000 wwwww01101000010	GR[reg3]←GR[reg2] (23 : 16) ll GR[reg2] (31 : 24) ll GR[reg2] (7 : 0) ll GR[reg2] (15 : 8)		1	1	1	×	0	×	×	
BSW	reg2,reg3	rrrrr11111100000 wwwww01101000000	GR[reg3]←GR[reg2] (7 : 0) ll GR[reg2] (15 : 8) ll GR [reg2] (23 : 16) ll GR[reg2] (31 : 24)		1	1	1	×	0	×	×	
CALLT	imm6	0 0 0 0 0 0 1 0 0 0 i i i i i i	CTPC←PC+2(return PC) CTPSW←PSW adr←CTBP+zero-extend(imm6 logically shift left by 1) PC←CTBP+zero-extend(Load-memory(adr,Halfword))		4	4	4					
CLR1	bit#3,disp16[reg1]	10bbb111110RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,0)		3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100100	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,0)		3 Note 3	3 Note 3	3 Note 3				×	
CMOV	cccc,imm5,reg2,reg3	r r r r r 1 1 1 1 1 i i i i i wwwww011000cccc0	if conditions are satisfied then GR[reg3]←sign-extend(imm5) else GR[reg3]←GR[reg2]		1	1	1					
	cccc,reg1,reg2,reg3	rrrrr111111RRRR wwwww011001cccc0	if conditions are satisfied then GR[reg3]←GR[reg1] else GR[reg3]←GR[reg2]		1	1	1					
CMP	reg1,reg2	rrrrr001111RRRRR	result←GR[reg2]–GR[reg1]		1	1	1	×	×	×	×	
	imm5,reg2	r r r r r 0 1 0 0 1 1 i i i i i	result←GR[reg2]–sign-extend(imm5)		1	1	1	×	×	×	×	
CTRET		0000011111100000 0000000101000100	PC←CTPC PSW←CTPSW		3	3	3	R	R	R	R	R
DBRET		0000011111100000 0000000101000110	PC←DBPC PSW←DBPSW		3	3	3	R	R	R	R	R

(2/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
DBTRAP		1111100001000000	DBPC←PC+2 (restored PC) DBPSW←PSW PSW.NP←1 PSW.EP←1 PSW.ID←1 PC←00000060H	3	3	3					
DI		0000011111100000 0000000101100000	PSW.ID←1	1	1	1					
DISPOSE	imm5,list12	0 0 0 0 0 1 1 0 0 1 i i i i i L L L L L L L L L L L L 0 0 0 0 0	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded	n+1 Note 4	n+1 Note 4	n+1 Note 4					
	imm5,list12[reg1]	0 0 0 0 0 1 1 0 0 1 i i i i i L L L L L L L L L L L L R R R R R **Note 5**	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded PC←GR[reg1]	n+3 Note 4	n+3 Note 4	n+3 Note 4					
DIV	reg1,reg2,reg3	r r r r r 1 1 1 1 1 1 R R R R R w w w w w 0 1 0 1 1 0 0 0 0 0	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVH	reg1,reg2	r r r r r 0 0 0 0 1 0 R R R R R	GR[reg2]←GR[reg2]÷GR[reg1]Note 6	35	35	35		×	×	×	
	reg1,reg2,reg3	r r r r r 1 1 1 1 1 1 R R R R R w w w w w 0 1 0 1 0 0 0 0 0 0	GR[reg2]←GR[reg2]÷GR[reg1]Note 6 GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVHU	reg1,reg2,reg3	r r r r r 1 1 1 1 1 1 R R R R R w w w w w 0 1 0 1 0 0 0 0 1 0	GR[reg2]←GR[reg2]÷GR[reg1]Note 6 GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
DIVU	reg1,reg2,reg3	r r r r r 1 1 1 1 1 1 R R R R R w w w w w 0 1 0 1 1 0 0 0 1 0	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
EI		1000011111100000 0000000101100000	PSW.ID←0	1	1	1					
HALT		0000011111100000 0000000100100000	Stop	1	1	1					
HSW	reg2,reg3	r r r r r 1 1 1 1 1 1 0 0 0 0 0 w w w w w 0 1 1 0 1 0 0 0 1 0 0	GR[reg3]←GR[reg2](15 : 0) ll GR[reg2] (31 : 16)	1	1	1	×	0	×	×	
JARL	disp22,reg2	r r r r r 1 1 1 1 0 d d d d d d d d d d d d d d d d d d d d d 0 **Note 7**	GR[reg2]←PC+4 PC←PC+sign-extend(disp22)	2	2	2					
JMP	[reg1]	00000000011RRRRR	PC←GR[reg1]	3	3	3					
JR	disp22	0000011110dddddd dddddddddddddd0 **Note 7**	PC←PC+sign-extend(disp22)	2	2	2					
LD.B	disp16[reg1],reg2	r r r r r 1 1 1 0 0 0 R R R R R d d d d d d d d d d d d d d d d	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	Note 11					
LD.BU	disp16[reg1],reg2	r r r r r 1 1 1 1 0 b R R R R R d d d d d d d d d d d d d d d 1 **Notes 8, 10**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	Note 11					

(3/6)

Mnemonic	Operand	Opcode	Operation		Execution Clock			Flags				
					i	r	l	CY	OV	S	Z	SAT
LD.H	disp16[reg1],reg2	rrrrr111001RRRRR ddddddddddddddd0 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Halfword))		1	1	Note 11					
LDSR	reg2,regID	rrrrr111111RRRRR 0000000000100000 **Note 12**	SR[regID]←GR[reg2]	Other than regID = PSW	1	1	1					
				regID = PSW	1	1	1	×	×	×	×	×
LD.HU	disp16[reg1],reg2	rrrrr111111RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Halfword)		1	1	Note 11					
LD.W	disp16[reg1],reg2	rrrrr111001RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←Load-memory(adr,Word)		1	1	Note 11					
MOV	reg1,reg2	rrrrr000000RRRRR	GR[reg2]←GR[reg1]		1	1	1					
	imm5,reg2	r r r r r 0 1 0 0 0 0 i i i i i	GR[reg2]←sign-extend(imm5)		1	1	1					
	imm32,reg1	00000110001RRRRR i i i i i i i i i i i i i i i i l l l l l l l l l l l l l l l l	GR[reg1]←imm32		2	2	2					
MOVEA	imm16,reg1,reg2	rrrrr110001RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1]+sign-extend(imm16)		1	1	1					
MOVHI	imm16,reg1,reg2	rrrrr110010RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1]+(imm16 ll 0^{16})		1	1	1					
MUL	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100000	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1] **Note 14**		1	4	5					
	imm9,reg2,reg3	r r r r r 1 1 1 1 1 1 i i i i i w w w w w 0 1 0 0 1 l l l l 0 0 **Note 13**	GR[reg3] ll GR[reg2]←GR[reg2]xsign-extend(imm9)		1	4	5					
MULH	reg1,reg2	rrrrr000111RRRRR	GR[reg2]←GR[reg2]$^{Note\ 6}$xGR[reg1]$^{Note\ 6}$		1	1	2					
	imm5,reg2	r r r r r 0 1 0 1 1 1 i i i i i	GR[reg2]←GR[reg2]$^{Note\ 6}$xsign-extend(imm5)		1	1	2					
MULHI	imm16,reg1,reg2	rrrrr110111RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1]$^{Note\ 6}$ximm16		1	1	2					
MULU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100010	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1] **Note 14**		1	4	5					
	imm9,reg2,reg3	r r r r r 1 1 1 1 1 1 i i i i i w w w w w 0 1 0 0 1 l l l l 1 0 **Note 13**	GR[reg3] ll GR[reg2]←GR[reg2]xzero-extend(imm9)		1	4	5					
NOP		0000000000000000	Pass at least one clock cycle doing nothing.		1	1	1					
NOT	reg1,reg2	rrrrr000001RRRRR	GR[reg2]←NOT(GR[reg1])		1	1	1		0	×	×	
NOT1	bit#3,disp16[reg1]	01bbb111110RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,Z flag)		3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100010	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,Z flag)		3 Note 3	3 Note 3	3 Note 3				×	

(4/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
OR	reg1,reg2	rrrrr001000RRRRR	GR[reg2]←GR[reg2]OR GR[reg1]	1	1	1		0	×	×	
ORI	imm16,reg1,reg2	rrrrr110100RRRRR iiiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]OR zero-extend(imm16)	1	1	1		0	×	×	
PREPARE	list12,imm5	0000011110iiiiiL LLLLLLLLLLL00001	Store-memory(sp–4,GR[reg in list12],Word) sp←sp–4 repeat 1 step above until all regs in list12 is stored sp←sp–zero-extend(imm5)	n+1 Note 4	n+1 Note 4	n+1 Note 4					
	list12,imm5, sp/imm[Note 15]	0000011110iiiiiL LLLLLLLLLLLff011 imm16/imm32 **Note 16**	Store-memory(sp–4,GR[reg in list12],Word) sp←sp–4 repeat 1 step above until all regs in list12 is stored sp←sp–zero-extend (imm5) ep←sp/imm	n+2 Note 4 Note 17	n+2 Note 4 Note 17	n+2 Note 4 Note 17					
RETI		0000011111100000 0000000101000000	if PSW.EP=1 then PC ←EIPC PSW ←EIPSW else if PSW.NP=1 then PC ←FEPC PSW ←FEPSW else PC ←EIPC PSW ←EIPSW	3	3	3	R	R	R	R	R
SAR	reg1,reg2	rrrrr111111RRRRR 0000000010100000	GR[reg2]←GR[reg2]arithmetically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010101iiiii	GR[reg2]←GR[reg2]arithmetically shift right by zero-extend (imm5)	1	1	1	×	0	×	×	
SASF	cccc,reg2	rrrrr1111110cccc 0000001000000000	if conditions are satisfied then GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000001H else GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000000H	1	1	1					
SATADD	reg1,reg2	rrrrr000110RRRRR	GR[reg2]←saturated(GR[reg2]+GR[reg1])	1	1	1	×	×	×	×	×
	imm5,reg2	rrrrr010001iiiii	GR[reg2]←saturated(GR[reg2]+sign-extend(imm5))	1	1	1	×	×	×	×	×
SATSUB	reg1,reg2	rrrrr000101RRRRR	GR[reg2]←saturated(GR[reg2]–GR[reg1])	1	1	1	×	×	×	×	×
SATSUBI	imm16,reg1,reg2	rrrrr110011RRRRR iiiiiiiiiiiiiiiii	GR[reg2]←saturated(GR[reg1]–sign-extend(imm16))	1	1	1	×	×	×	×	×
SATSUBR	reg1,reg2	rrrrr000100RRRRR	GR[reg2]←saturated(GR[reg1]–GR[reg2])	1	1	1	×	×	×	×	×
SETF	cccc,reg2	rrrrr1111110cccc 0000000000000000	If conditions are satisfied then GR[reg2]←00000001H else GR[reg2]←00000000H	1	1	1					

(5/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
SET1	bit#3,disp16[reg1]	00bbb111110RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,1)	3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100000	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,1)	3 Note 3	3 Note 3	3 Note 3				×	
SHL	reg1,reg2	rrrrr111111RRRRR 0000000011000000	GR[reg2]←GR[reg2] logically shift left by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	r r r r r 0 1 0 1 1 0 i i i i i	GR[reg2]←GR[reg2] logically shift left by zero-extend(imm5)	1	1	1	×	0	×	×	
SHR	reg1,reg2	rrrrr111111RRRRR 0000000010000000	GR[reg2]←GR[reg2] logically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	r r r r r 0 1 0 1 0 0 i i i i i	GR[reg2]←GR[reg2] logically shift right by zero-extend(imm5)	1	1	1	×	0	×	×	
SLD.B	disp7[ep],reg2	rrrrr0110ddddddd	adr←ep+zero-extend(disp7) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	Note 9					
SLD.BU	disp4[ep],reg2	rrrrr0000110dddd **Note 18**	adr←ep+zero-extend(disp4) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	Note 9					
SLD.H	disp8[ep],reg2	rrrrr1000ddddddd **Note 19**	adr←ep+zero-extend(disp8) GR[reg2]←sign-extend(Load-memory(adr,Halfword))	1	1	Note 9					
SLD.HU	disp5[ep],reg2	rrrrr0000111dddd **Notes 18, 20**	adr←ep+zero-extend(disp5) GR[reg2]←zero-extend(Load-memory(adr,Halfword))	1	1	Note 9					
SLD.W	disp8[ep],reg2	rrrrr1010ddddddd0 **Note 21**	adr←ep+zero-extend(disp8) GR[reg2]←Load-memory(adr,Word)	1	1	Note 9					
SST.B	reg2,disp7[ep]	rrrrr0111ddddddd	adr←ep+zero-extend(disp7) Store-memory(adr,GR[reg2],Byte)	1	1	1					
SST.H	reg2,disp8[ep]	rrrrr1001ddddddd **Note 19**	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Halfword)	1	1	1					
SST.W	reg2,disp8[ep]	rrrrr1010ddddddd1 **Note 21**	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Word)	1	1	1					
ST.B	reg2,disp16[reg1]	rrrrr111010RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr,GR[reg2],Byte)	1	1	1					
ST.H	reg2,disp16[reg1]	rrrrr111011RRRRR dddddddddddddd0 **Note 8**	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr,GR[reg2], Halfword)	1	1	1					
ST.W	reg2,disp16[reg1]	rrrrr111011RRRRR dddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr,GR[reg2], Word)	1	1	1					
STSR	regID,reg2	rrrrr111111RRRRR 0000000001000000	GR[reg2]←SR[regID]	1	1	1					

(6/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
SUB	reg1,reg2	rrrrr001101RRRRR	GR[reg2]←GR[reg2]–GR[reg1]	1	1	1	×	×	×	×	
SUBR	reg1,reg2	rrrrr001100RRRRR	GR[reg2]←GR[reg1]–GR[reg2]	1	1	1	×	×	×	×	
SWITCH	reg1	00000000010RRRRR	adr←(PC+2) + (GR [reg1] logically shift left by 1) PC←(PC+2) + (sign-extend (Load-memory (adr,Halfword)) logically shift left by 1	5	5	5					
SXB	reg1	00000000101RRRRR	GR[reg1]←sign-extend (GR[reg1] (7 : 0))	1	1	1					
SXH	reg1	00000000111RRRRR	GR[reg1]←sign-extend (GR[reg1] (15 : 0))	1	1	1					
TRAP	vector	0 0 0 0 0 1 1 1 1 1 1 i i i i i 0000000100000000	EIPC ←PC+4 (Restored PC) EIPSW ←PSW ECR.EICC ←Interrupt code PSW.EP ←1 PSW.ID ←1 PC ←00000040H (when vector is 00H to 0FH) 00000050H (when vector is 10H to 1FH)	3	3	3					
TST	reg1,reg2	rrrrr001011RRRRR	result←GR[reg2] AND GR[reg1]	1	1	1		0	×	×	
TST1	bit#3,disp16[reg1]	11bbb111110RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit (adr,bit#3))	3 Note 3	3 Note 3	3 Note 3				×	
	reg2, [reg1]	rrrrr111111RRRRR 0000000011100110	adr←GR[reg1] Z flag←Not (Load-memory-bit (adr,reg2))	3 Note 3	3 Note 3	3 Note 3				×	
XOR	reg1,reg2	rrrrr001001RRRRR	GR[reg2]←GR[reg2] XOR GR[reg1]	1	1	1		0	×	×	
XORI	imm16,reg1,reg2	rrrrr110101RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1] XOR zero-extend (imm16)	1	1	1		0	×	×	
ZXB	reg1	00000000100RRRRR	GR[reg1]←zero-extend (GR[reg1] (7 : 0))	1	1	1					
ZXH	reg1	00000000110RRRRR	GR[reg1]←zero-extend (GR[reg1] (15 : 0))	1	1	1					

Notes 1. ddddddd: Higher 8 bits of disp9.

2. 3 if there is an instruction that rewrites the contents of the PSW immediately before.

3. If there is no wait state (3 + the number of read access wait states).

4. n is the total number of list12 load registers. (According to the number of wait states. Also, if there are no wait states, n is the total number of list12 registers. If n = 0, same operation as when n = 1)

5. RRRRR: other than 00000.

6. The lower halfword data only are valid.

7. dddddddddddddddddddddd: The higher 21 bits of disp22.

8. ddddddddddddddd: The higher 15 bits of disp16.

9. According to the number of wait states (1 if there are no wait states).

10. b: bit 0 of disp16.

11. According to the number of wait states (2 if there are no wait states).

Notes 12. In this instruction, for convenience of mnemonic description, the source register is made reg2, but the reg1 field is used in the opcode. Therefore, the meaning of register specification in the mnemonic description and in the opcode differs from other instructions.

rrrrr = regID specification

RRRRR = reg2 specification

13. iiiii: Lower 5 bits of imm9.

IIII: Higher 4 bits of imm9.

14. Do not specify the same register for general-purpose registers reg1 and reg3.

15. sp/imm: specified by bits 19 and 20 of the sub-opcode.

16. ff = 00: Load sp in ep.

01: Load sign expanded 16-bit immediate data (bits 47 to 32) in ep.

10: Load 16-bit logically left shifted 16-bit immediate data (bits 47 to 32) in ep.

11: Load 32-bit immediate data (bits 63 to 32) in ep.

17. If imm = imm32, n + 3 clocks.

18. rrrrr: Other than 00000.

19. ddddddd: Higher 7 bits of disp8.

20. dddd: Higher 4 bits of disp5.

21. dddddd: Higher 6 bits of disp8.

★

APPENDIX C REVISION HISTORY

C.1 Major Revisions in This Edition

(1/3)

Page	Description
p. 21	Addition of **Note 2** to **1.5 Pin Configuration (Top View)**
pp. 28 to 30	Modification of **2.1 (1) Port pins**
pp. 31 to 33	Modification of **2.1 (2) Non-port pins**
p. 34	Modification of **Table 2-2 Pin Operation States in Various Modes**
p. 47	Addition of **3.3.1 Specifying operation mode**
p. 54	Addition of **Caution** to **3.4.5 Recommended use of address space**
pp. 56, 59	Addition of **Note** to port n register in **3.4.6 Peripheral I/O registers**
p. 68	Addition of **Remark** to **3.4.7 (3) (a) Set condition (PRERR flag = 1)**
p. 70	Addition of **3.4.8 (2) Access to special on-chip peripheral I/O registers**
p. 71	Addition of **3.4.8 (3) Cautions when using flash memory version (μPD70F3229Y)**
p. 73	Modification of **4.3 Port Configuration**
p. 117	Addition of **4.4 Block Diagrams**
p. 150	Addition of **4.6 Cautions**
p. 154	Modification of **5.2.1 Pin status when internal ROM, internal RAM, or on-chip peripheral I/O is accessed**
p. 166	Addition of **Cautions** to **5.5.4 Programmable address wait function**
p. 171	Modification of **5.9 Bus Timing**
p. 175	Modification of **Figure 6-1 Clock Generator**
p. 179	Modification of **6.3 (1) (a) Example of setting main clock operation → subclock operation**
p. 180	Modification of **6.3 (1) (b) Example of setting subclock operation → main clock operation**
p. 185	Modification of description on PLLS1 and PLLS0 bits and addition of **Caution** in **6.5.2 (3) PLL lockup time specification register (PLLS)**
p. 186	Modification of **6.5.3 (1) When PLL is used**
p. 187	Modification of **CHAPTER 7 16-BIT TIMER/EVENT COUNTER P (TMP)**
p. 277	Modification of **CHAPTER 8 16-BIT TIMER/EVENT COUNTER Q (TMQ)**
p. 380	Modification of **9.4.1 Interval timer mode**
p. 384	Addition of **9.4.2 Cautions**
p. 386	Modification of **10.2 Configuration**
p. 392	Modification of **10.4.1 Operation as watch timer**
p. 397	Addition of **Caution** to **11.3 (1) Watchdog timer mode register 2 (WDTM2)**
p. 400	Addition of **11.4 Operation**
p. 401	Modification of **CHAPTER 12 A/D CONVERTER**
p. 433	Modification of **13.3 (1) UARTAn control register 0 (UAnCTL0)**
p. 443	Addition of **Remark** to **13.5.2 SBF transmission/reception format**
p. 453	Addition of **Figure 13-12 Timing of RXDAn Signal Judged as Noise**
p. 455	Addition of **Caution** to **13.6 (2) UARTAn control register 1 (UAnCTL1)**
p. 456	Addition of **Caution** to **13.6 (3) UARTAn control register 2 (UAnCTL2)**
p. 461	Addition of **13.7 Cautions**
p. 464	Addition of **Remark** to **14.3 Configuration**

(2/3)

Page	Description
p. 465	Modification of **14.4 (1) CSIBn control register 0 (CBnCTL0)**
p. 467	Modification of **Caution** in **14.4 (2) CSIBn control register 1 (CBnCTL1)**
p. 471	Addition of **14.5 Interrupt Request Signals**
p. 472	Modification of **14.6 Operation**
p. 481	Modification of **14.7 (1) $\overline{\text{SCKBn}}$ pin**
p. 482	Modification of **14.8 Operation Flow**
pp. 496, 499	Addition of **Notes** to **15.2 (1) IIC control register 0 (IICC0)**
p. 500	Addition of **Caution** to **15.2 (2) IIC status register 0 (IICS0)**
p. 508	Modification of **15.2 (7) IIC division clock select register 0 (OCKS0)**
p. 546	Modification of **15.14.3 Slave operation**
p. 558	Modification of **16.3 Control Registers**
p. 565	Modification of **16.4 Transfer Targets**
p. 565	Modification of **16.5 Transfer Modes**
p. 566	Modification of **16.6 Transfer Types**
p. 567	Modification of **16.7 DMA Channel Priorities**
p. 567	Addition of **16.8 Time Related to DMA Transfer**
p. 568	Modification of **16.9 DMA Transfer Start Factors**
p. 569	Modification of **16.10 DMA Abort Factors**
p. 569	Modification of **16.11 End of DMA Transfer**
p. 569	Addition of **16.12 Operation Timing**
p. 574	Modification of **16.13 Cautions**
p. 580	Addition of **Note 1** to **Table 17-1 Interrupt Source List**
p. 583	Addition of **Note** and modification of **Caution** in **17.2 Non-Maskable Interrupts**
p. 587	Modification of **17.2.2 (2) INTWDT2 signal**
p. 599	Addition of **Note** and **Caution** to **17.3.5 Interrupt mask registers 0 to 3 (IMR0 to IMR3)**
p. 613	Addition of **Caution** to **17.6.2 (2) Restore**
p. 614	Modification of **Figure 17-14 Pipeline Operation at Interrupt Request Signal Acknowledgment (Outline)**
p. 615	Addition of **17.9 Cautions**
p. 616	Addition of **Caution** to **18.1 Function**
p. 620	Modification of **Figure 19-1 Status Transition**
p. 627	Addition of **Note** to **Table 19-5 Operation Status in IDLE1 Mode**
p. 629	Addition of **Note** to **Table 19-7 Operation Status in IDLE2 Mode**
p. 632	Addition of **Notes 1** and **2** to **Table 19-9 Operation Status in STOP Mode**
p. 634	Addition of **Caution 2** to **19.8.1 Setting and operation status**
p. 635	Addition of **Caution** to **Table 19-10 Operation Status in Subclock Operation Mode**
p. 636	Addition of **Caution 2** to **19.9.2 Releasing sub-IDLE mode**
p. 637	Addition of **Note 2** to **Table 19-12 Operation Status in Sub-IDLE Mode**
p. 638	Modification of **20.2 (1) Reset source flag register (RESF)**
p. 648	Addition of **Note** in **20.3.3 (5) Internal RAM data status register (RAMS)**
p. 649	Modification of **Table 20-4 Operation Status of Clock Monitor (CLME Bit = 1, with Ring-OSC Operating)**
p. 655	Addition of **Note** to **22.2 (1) Correction address registers 0 to 3 (CORAD0 to CORAD3)**

(3/3)

Page	Description
p. 658	Modification of **CHAPTER 23 FLASH MEMORY**
p. 681	Addition of **CHAPTER 24 ON-CHIP DEBUG FUNCTION**
p. 690	Addition of **CHAPTER 25 ELECTRICAL SPECIFICATIONS**
p. 715	Addition of **CHAPTER 26 PACKAGE DRAWING**
p. 716	Addition of **APPENDIX A REGISTER INDEX**
p. 724	Addition of **APPENDIX B INSTRUCTION SET LIST**
p. 734	Addition of **APPENDIX C REVISION HISTORY**