

To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
 - “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
 - “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
 - “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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



User's Manual

V850ES/HF2

32-bit Single-Chip Microcontrollers

Hardware

μ PD70F3702

μ PD70F3703

μ PD70F3704

Document No. U17719EJ2V0UD00 (2nd edition)
Date Published March 2007 N CP(K)

© NEC Electronics Corporation 2005
Printed in Japan

[MEMO]

NOTES FOR CMOS DEVICES

① VOLTAGE APPLICATION WAVEFORM AT INPUT PIN

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

② HANDLING OF UNUSED INPUT PINS

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

③ PRECAUTION AGAINST ESD

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

④ STATUS BEFORE INITIALIZATION

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

⑤ POWER ON/OFF SEQUENCE

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

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

⑥ INPUT OF SIGNAL DURING POWER OFF STATE

Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

IECUBE is a registered trademark of NEC Electronics Corporation in Japan and Germany.

MINICUBE is a registered trademark of NEC Electronics Corporation in Japan and Germany or a trademark in the United States of America.

Applilet is a registered trademark of NEC Electronics in Japan, Germany, Hong Kong, China, the Republic of Korea, the United Kingdom, and the United States of America.

Windows and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

PC/AT is a trademark of International Business Machines Corporation.

SPARCstation is a trademark of SPARC International, Inc.

Solaris and SunOS are trademarks of Sun Microsystems, Inc.

TRON is an abbreviation of The Real-Time Operating system Nucleus.

ITRON is an abbreviation of Industrial TRON.

• **The information in this document is current as of September, 2006. 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

PREFACE

Readers This manual is intended for users who wish to understand the functions of the V850ES/HF2 and design application systems using the V850ES/HF2.

Purpose This manual is intended to give users an understanding of the hardware functions of the V850ES/HF2 shown in the **Organization** below.

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

| |
|----------|
| Hardware |
|----------|

- Pin functions
- CPU function
- On-chip peripheral functions
- Flash memory programming
- Electrical specifications

| |
|--------------|
| Architecture |
|--------------|

- Data types
- Register set
- Instruction format and instruction set
- Interrupts and exceptions
- 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 understand the overall functions of the V850ES/HF2

→ Read this manual according to the **CONTENTS**.

To find the details of a register where the name is known

→ Use **APPENDIX B REGISTER INDEX**.

To understand the details of an instruction function

→ Refer to the **V850ES Architecture User's Manual** available separately.

To know the electrical specifications of the V850ES/HF2

→ See **CHAPTER 25 ELECTRICAL SPECIFICATIONS**.

The “yyy bit of the xxx register” is described as the “xxx.yyy bit” in this manual. Note with caution that if “xxx.yyy” is described as is in a program, however, the compiler/assembler cannot recognize it correctly.

The mark <R> shows major revised points. The revised points can be easily searched by copying an “<R>” in the PDF file and specifying it in the “Find what:” field.

Conventions

| | |
|--|--|
| Data significance: | Higher digits on the left and lower digits on the right |
| Active low representation: | $\overline{\text{xxx}}$ (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 V850ES/HF2

| Document Name | Document No. |
|-----------------------------------|--------------|
| V850ES Architecture User's Manual | U15943E |
| V850ES/HF2 Hardware User's Manual | This manual |

Documents related to development tools

| Document Name | Document No. | |
|---|---|---------|
| QB-V850MINI On-Chip Debug Emulator | U17638E | |
| QB-MINI2 On-Chip Debug Emulator with Programming Function | U18371E | |
| CA850 Ver. 3.00 C Compiler Package | Operation | U17293E |
| | C Language | U17291E |
| | Assembly Language | U17292E |
| | Link Directives | U17294E |
| PM+ Ver. 6.20 Project Manager | U17990E | |
| ID850QB Ver. 3.20 Integrated Debugger | Operation | U17964E |
| SM850 Ver. 2.50 System Simulator | Operation | U16218E |
| SM850 Ver. 2.00 or Later System Simulator | External Part User Open Interface Specification | U14873E |
| SM+ System Simulator | Operation | U17246E |
| | User Open Interface Specification | U17247E |
| RX850 Ver. 3.20 Real-Time OS | Basics | U13430E |
| | Installation | U17419E |
| | Technical | U13431E |
| | Task Debugger | U17420E |
| RX850 Pro Ver. 3.20 Real-Time OS | Basics | U13773E |
| | Installation | U17421E |
| | Technical | U13772E |
| | Task Debugger | U17422E |
| AZ850 Ver. 3.30 System Performance Analyzer | U17423E | |
| PG-FP4 Flash Memory Programmer | U15260E | |

CONTENTS

| | |
|--|-----------|
| CHAPTER 1 INTRODUCTION | 17 |
| 1.1 General | 17 |
| 1.2 Features..... | 18 |
| 1.3 Application Fields | 18 |
| 1.4 Ordering Information | 19 |
| 1.5 Pin Configuration (Top View)..... | 20 |
| 1.6 Function Block Configuration..... | 22 |
| 1.6.1 Internal block diagram | 22 |
| 1.6.2 Internal units..... | 23 |
| CHAPTER 2 PIN FUNCTIONS..... | 25 |
| 2.1 Pin Function List | 25 |
| 2.2 Description of Pin Functions | 31 |
| 2.3 Pin I/O Circuit Types and Recommended Connection of Unused Pins | 37 |
| 2.4 Pin I/O Circuits..... | 39 |
| 2.5 Caution | 40 |
| CHAPTER 3 CPU FUNCTION..... | 41 |
| 3.1 Features..... | 41 |
| 3.2 CPU Register Set..... | 42 |
| 3.2.1 Program register set | 43 |
| 3.2.2 System register set..... | 44 |
| 3.3 Operation Modes | 50 |
| 3.3.1 Specifying operation mode | 50 |
| 3.4 Address Space | 51 |
| 3.4.1 CPU address space..... | 51 |
| 3.4.2 Wraparound of CPU address space | 52 |
| 3.4.3 Memory map..... | 53 |
| 3.4.4 Areas | 55 |
| 3.4.5 Recommended use of address space | 58 |
| 3.4.6 Peripheral I/O registers..... | 61 |
| 3.4.7 Special registers | 68 |
| 3.4.8 Cautions | 72 |
| CHAPTER 4 PORT FUNCTIONS..... | 75 |
| 4.1 Features..... | 75 |
| 4.2 Basic Configuration of Ports | 75 |
| 4.3 Port Functions | 77 |
| 4.3.1 Operation of port function | 77 |
| 4.3.2 Notes on setting port pins | 78 |
| 4.3.3 Port 0..... | 79 |
| 4.3.4 Port 3..... | 83 |
| 4.3.5 Port 4..... | 88 |
| 4.3.6 Port 5..... | 91 |
| 4.3.7 Port 7..... | 97 |

<R>

| | | |
|---|--|------------|
| 4.3.8 | Port 9 | 99 |
| 4.3.9 | Port CM | 107 |
| 4.3.10 | Port CS | 109 |
| 4.3.11 | Port CT | 111 |
| 4.3.12 | Port DL | 113 |
| 4.3.13 | Port pins that function alternately as on-chip debug function | 115 |
| 4.3.14 | Register settings to use port pins as alternate-function pins | 116 |
| 4.4 | Block Diagrams of Port | 120 |
| 4.5 | Cautions | 145 |
| 4.5.1 | Cautions on setting port pins | 145 |
| CHAPTER 5 CLOCK GENERATION FUNCTION | | 146 |
| 5.1 | Overview | 146 |
| 5.2 | Configuration | 147 |
| 5.3 | Registers | 149 |
| 5.4 | Operation | 154 |
| 5.4.1 | Operation of each clock | 154 |
| 5.4.2 | Clock output function | 154 |
| 5.5 | PLL Function | 155 |
| 5.5.1 | Overview | 155 |
| 5.5.2 | Registers | 155 |
| 5.5.3 | Usage | 159 |
| CHAPTER 6 16-BIT TIMER/EVENT COUNTER P (TMP) | | 160 |
| 6.1 | Overview | 160 |
| 6.2 | Functions | 160 |
| 6.3 | Configuration | 161 |
| 6.4 | Registers | 163 |
| 6.5 | Operation | 177 |
| 6.5.1 | Interval timer mode (TPnMD2 to TPnMD0 bits = 000) | 178 |
| 6.5.2 | External event count mode (TPnMD2 to TPnMD0 bits = 001) | 188 |
| 6.5.3 | External trigger pulse output mode (TPnMD2 to TPnMD0 bits = 010) | 196 |
| 6.5.4 | One-shot pulse output mode (TPnMD2 to TPnMD0 bits = 011) | 208 |
| 6.5.5 | PWM output mode (TPnMD2 to TPnMD0 bits = 100) | 215 |
| 6.5.6 | Free-running timer mode (TPnMD2 to TPnMD0 bits = 101) | 224 |
| 6.5.7 | Pulse width measurement mode (TPnMD2 to TPnMD0 bits = 110) | 241 |
| 6.5.8 | Timer output operations | 247 |
| 6.6 | Timer Tuned Operation Function | 248 |
| 6.7 | Selector Function | 252 |
| 6.8 | Cautions | 254 |
| CHAPTER 7 16-BIT TIMER/EVENT COUNTER Q (TMQ) | | 255 |
| 7.1 | Overview | 255 |
| 7.2 | Functions | 255 |
| 7.3 | Configuration | 256 |
| 7.4 | Registers | 259 |
| 7.5 | Operation | 277 |
| 7.5.1 | Interval timer mode (TQ0MD2 to TQ0MD0 bits = 000) | 278 |

| | | |
|--|--|------------|
| 7.5.2 | External event count mode (TQ0MD2 to TQ0MD0 bits = 001) | 287 |
| 7.5.3 | External trigger pulse output mode (TQ0MD2 to TQ0MD0 bits = 010) | 296 |
| 7.5.4 | One-shot pulse output mode (TQ0MD2 to TQ0MD0 bits = 011) | 309 |
| 7.5.5 | PWM output mode (TQ0MD2 to TQ0MD0 bits = 100) | 318 |
| 7.5.6 | Free-running timer mode (TQ0MD2 to TQ0MD0 bits = 101) | 329 |
| 7.5.7 | Pulse width measurement mode (TQ0MD2 to TQ0MD0 bits = 110) | 349 |
| 7.5.8 | Triangular wave PWM mode (TQ0MD2 to TQ0MD0 = 111) | 355 |
| 7.5.9 | Timer output operations | 356 |
| 7.6 | Timer Tuned Operation Function | 357 |
| 7.7 | Cautions | 361 |
| CHAPTER 8 16-BIT INTERVAL TIMER M (TMM)..... | | 362 |
| 8.1 | Overview..... | 362 |
| 8.2 | Configuration..... | 363 |
| 8.3 | Register | 364 |
| 8.4 | Operation..... | 365 |
| 8.4.1 | Interval timer mode..... | 365 |
| 8.4.2 | Cautions | 369 |
| CHAPTER 9 WATCH TIMER FUNCTIONS..... | | 370 |
| 9.1 | Functions | 370 |
| 9.2 | Configuration..... | 371 |
| 9.3 | Registers | 373 |
| 9.4 | Operation..... | 377 |
| 9.4.1 | Operation as watch timer..... | 377 |
| 9.4.2 | Operation as interval timer..... | 378 |
| 9.4.3 | Cautions | 379 |
| CHAPTER 10 FUNCTIONS OF WATCHDOG TIMER 2..... | | 380 |
| 10.1 | Functions | 380 |
| 10.2 | Configuration..... | 381 |
| 10.3 | Registers | 382 |
| 10.4 | Operation..... | 385 |
| CHAPTER 11 A/D CONVERTER..... | | 386 |
| 11.1 | Overview..... | 386 |
| 11.2 | Functions | 386 |
| 11.3 | Configuration..... | 387 |
| 11.4 | Registers | 390 |
| 11.5 | Operation..... | 398 |
| 11.5.1 | Basic operation..... | 398 |
| 11.5.2 | Trigger mode | 399 |
| 11.5.3 | Operation mode..... | 401 |
| 11.5.4 | Power-fail compare mode..... | 405 |
| 11.6 | Cautions | 410 |
| 11.7 | How to Read A/D Converter Characteristics Table..... | 414 |
| CHAPTER 12 ASYNCHRONOUS SERIAL INTERFACE A (UARTA)..... | | 418 |

| | | |
|-------------------|--|------------|
| 12.1 | Features | 418 |
| 12.2 | Configuration | 419 |
| 12.3 | Registers | 421 |
| 12.4 | Interrupt Request Signals | 427 |
| 12.5 | Operation | 428 |
| 12.5.1 | Data format..... | 428 |
| 12.5.2 | SBF transmission/reception format..... | 430 |
| 12.5.3 | SBF transmission..... | 432 |
| 12.5.4 | SBF reception..... | 433 |
| 12.5.5 | UART transmission..... | 434 |
| 12.5.6 | Continuous transmission procedure..... | 435 |
| 12.5.7 | UART reception | 437 |
| 12.5.8 | Reception errors | 438 |
| 12.5.9 | Parity types and operations | 440 |
| 12.5.10 | Receive data noise filter..... | 441 |
| 12.6 | Dedicated Baud Rate Generator | 442 |
| 12.7 | Cautions | 450 |
| CHAPTER 13 | 3-WIRE VARIABLE-LENGTH SERIAL I/O (CSIB) | 451 |
| 13.1 | Features | 451 |
| 13.2 | Configuration | 452 |
| 13.3 | Registers | 454 |
| 13.4 | Interrupt Request Signals | 461 |
| 13.5 | Operation | 462 |
| 13.5.1 | Single transfer mode (master mode, transmission mode)..... | 462 |
| 13.5.2 | Single transfer mode (master mode, reception mode)..... | 464 |
| 13.5.3 | Single transfer mode (master mode, transmission/reception mode)..... | 466 |
| 13.5.4 | Single transfer mode (slave mode, transmission mode)..... | 468 |
| 13.5.5 | Single transfer mode (slave mode, reception mode)..... | 470 |
| 13.5.6 | Single transfer mode (slave mode, transmission/reception mode)..... | 472 |
| 13.5.7 | Continuous transfer mode (master mode, transmission mode) | 474 |
| 13.5.8 | Continuous transfer mode (master mode, reception mode)..... | 476 |
| 13.5.9 | Continuous transfer mode (master mode, transmission/reception mode)..... | 479 |
| 13.5.10 | Continuous transfer mode (slave mode, transmission mode)..... | 483 |
| 13.5.11 | Continuous transfer mode (slave mode, reception mode) | 485 |
| 13.5.12 | Continuous transfer mode (slave mode, transmission/reception mode) | 488 |
| 13.5.13 | Reception error | 492 |
| 13.5.14 | Clock timing | 493 |
| 13.6 | Output Pin Status with Operation Disabled | 495 |
| 13.7 | Baud Rate Generator | 496 |
| 13.7.1 | Baud rate generation | 497 |
| 13.8 | Cautions | 498 |
| CHAPTER 14 | INTERRUPT/EXCEPTION PROCESSING FUNCTION | 499 |
| 14.1 | Features | 499 |
| 14.2 | Non-Maskable Interrupts | 502 |
| 14.2.1 | Operation..... | 504 |
| 14.2.2 | Restore | 505 |

| | | |
|-------------------|---|------------|
| 14.2.3 | NP flag..... | 506 |
| 14.3 | Maskable Interrupts | 507 |
| 14.3.1 | Operation..... | 507 |
| 14.3.2 | Restore | 509 |
| 14.3.3 | Priorities of maskable interrupts | 510 |
| 14.3.4 | Interrupt control register (xxICn)..... | 514 |
| 14.3.5 | Interrupt mask registers 0 to 2 (IMR0 to IMR2)..... | 516 |
| 14.3.6 | In-service priority register (ISPR)..... | 517 |
| 14.3.7 | ID flag..... | 518 |
| 14.3.8 | Watchdog timer mode register 2 (WDTM2) | 518 |
| 14.4 | Software Exception | 519 |
| 14.4.1 | Operation..... | 519 |
| 14.4.2 | Restore | 520 |
| 14.4.3 | EP flag..... | 521 |
| 14.5 | Exception Trap | 522 |
| 14.5.1 | Illegal opcode definition | 522 |
| 14.5.2 | Debug trap..... | 524 |
| 14.6 | External Interrupt Request Input Pins (NMI and INTP0 to INTP7) | 526 |
| 14.6.1 | Noise elimination | 526 |
| 14.6.2 | Edge detection..... | 526 |
| 14.7 | Interrupt Acknowledge Time of CPU..... | 532 |
| 14.8 | Periods in Which Interrupts Are Not Acknowledged by CPU..... | 533 |
| 14.9 | Cautions | 533 |
| CHAPTER 15 | KEY INTERRUPT FUNCTION | 534 |
| 15.1 | Function | 534 |
| 15.2 | Register | 535 |
| 15.3 | Cautions | 535 |
| CHAPTER 16 | STANDBY FUNCTION..... | 536 |
| 16.1 | Overview..... | 536 |
| 16.2 | Registers | 538 |
| 16.3 | HALT Mode..... | 541 |
| 16.3.1 | Setting and operation status | 541 |
| 16.3.2 | Releasing HALT mode | 541 |
| 16.4 | IDLE1 Mode..... | 543 |
| 16.4.1 | Setting and operation status | 543 |
| 16.4.2 | Releasing IDLE1 mode..... | 543 |
| 16.5 | IDLE2 Mode..... | 545 |
| 16.5.1 | Setting and operation status | 545 |
| 16.5.2 | Releasing IDLE2 mode..... | 545 |
| 16.5.3 | Securing setup time when releasing IDLE2 mode | 547 |
| 16.6 | STOP Mode | 548 |
| 16.6.1 | Setting and operation status | 548 |
| 16.6.2 | Releasing STOP mode..... | 548 |
| 16.6.3 | Securing oscillation stabilization time when releasing STOP mode..... | 550 |
| 16.7 | Subclock Operation Mode..... | 551 |
| 16.7.1 | Setting and operation status | 551 |

| | | | |
|-------------------|--|---|------------|
| | 16.7.2 | Releasing subclock operation mode | 551 |
| 16.8 | Sub-IDLE Mode | | 553 |
| | 16.8.1 | Setting and operation status | 553 |
| | 16.8.2 | Releasing sub-IDLE mode | 554 |
| CHAPTER 17 | RESET FUNCTIONS | | 556 |
| 17.1 | Overview | | 556 |
| 17.2 | Registers to Check Reset Source | | 557 |
| 17.3 | Operation | | 558 |
| | 17.3.1 | Reset operation via $\overline{\text{RESET}}$ pin | 558 |
| | 17.3.2 | Reset operation by watchdog timer 2 | 560 |
| | 17.3.3 | Reset operation by power-on-clear circuit | 561 |
| | 17.3.4 | Reset operation by low-voltage detector | 561 |
| | 17.3.5 | Reset operation by clock monitor | 561 |
| <R> | 17.4 | Operation After Reset Release | 562 |
| CHAPTER 18 | CLOCK MONITOR | | 564 |
| 18.1 | Functions | | 564 |
| 18.2 | Configuration | | 564 |
| 18.3 | Register | | 565 |
| 18.4 | Operation | | 566 |
| CHAPTER 19 | POWER-ON-CLEAR CIRCUIT | | 569 |
| 19.1 | Function | | 569 |
| 19.2 | Configuration | | 569 |
| 19.3 | Operation | | 570 |
| CHAPTER 20 | LOW-VOLTAGE DETECTOR | | 571 |
| 20.1 | Functions | | 571 |
| 20.2 | Configuration | | 571 |
| 20.3 | Registers | | 572 |
| 20.4 | Operation | | 574 |
| | 20.4.1 | To use for internal reset signal | 574 |
| | 20.4.2 | To use for interrupt | 576 |
| 20.5 | RAM Retention Voltage Detection Operation | | 577 |
| 20.6 | Emulation Function | | 578 |
| CHAPTER 21 | REGULATOR | | 579 |
| 21.1 | Overview | | 579 |
| 21.2 | Operation | | 580 |
| CHAPTER 22 | FLASH MEMORY | | 581 |
| 22.1 | Features | | 581 |
| <R> | 22.2 | Memory Configuration | 582 |
| <R> | 22.3 | Functional Outline | 583 |
| 22.4 | Rewriting by Dedicated Flash Programmer | | 586 |
| | 22.4.1 | Programming environment | 586 |
| | 22.4.2 | Communication mode | 587 |

| | | |
|---|--|------------|
| 22.4.3 | Flash memory control | 592 |
| 22.4.4 | Selection of communication mode | 593 |
| 22.4.5 | Communication commands | 594 |
| 22.4.6 | Pin connection | 595 |
| 22.5 | Rewriting by Self Programming | 599 |
| 22.5.1 | Overview | 599 |
| 22.5.2 | Features | 600 |
| 22.5.3 | Standard self programming flow | 601 |
| 22.5.4 | Flash functions | 602 |
| 22.5.5 | Pin processing | 602 |
| 22.5.6 | Internal resources used | 603 |
| CHAPTER 23 OPTION BYTE FUNCTION | | 604 |
| <R> | CHAPTER 24 ON-CHIP DEBUG FUNCTION | 606 |
| 24.1 | Debugging with DCU | 607 |
| 24.1.1 | Connection circuit example | 607 |
| 24.1.2 | Interface signals | 607 |
| 24.1.3 | Maskable functions | 609 |
| 24.1.4 | Register | 609 |
| 24.1.5 | Operation | 611 |
| 24.1.6 | Cautions | 612 |
| 24.2 | Debugging Without Using DCU | 613 |
| 24.2.1 | Circuit connection examples | 613 |
| 24.2.2 | Maskable functions | 614 |
| 24.2.3 | Securement of user resources | 615 |
| 24.2.4 | Cautions | 621 |
| 24.3 | ROM Security Function | 622 |
| 24.3.1 | Security ID | 622 |
| 24.3.2 | Setting | 623 |
| CHAPTER 25 ELECTRICAL SPECIFICATIONS | | 625 |
| 25.1 | Absolute Maximum Ratings | 625 |
| 25.2 | Capacitance | 627 |
| 25.3 | Operating Conditions | 627 |
| 25.4 | Oscillator Characteristics | 628 |
| 25.4.1 | Main clock oscillator characteristics | 628 |
| 25.4.2 | Subclock oscillator characteristics | 629 |
| 25.4.3 | PLL characteristics | 630 |
| 25.4.4 | Internal oscillator characteristics | 630 |
| 25.5 | Voltage Regulator Characteristics | 630 |
| 25.6 | DC Characteristics | 631 |
| 25.6.1 | I/O level | 631 |
| 25.6.2 | Pin leakage current | 632 |
| 25.6.3 | Supply current | 633 |
| 25.7 | Data Retention Characteristics | 634 |
| 25.8 | AC Characteristics | 635 |
| 25.8.1 | CLKOUT output timing | 636 |

| | | |
|----------------|---|-----|
| 25.9 | Basic Operation | 637 |
| 25.10 | Flash Memory Programming Characteristics..... | 644 |
| CHAPTER 26 | PACKAGE DRAWING | 645 |
| <R> CHAPTER 27 | RECOMMENDED SOLDERING CONDITIONS..... | 646 |
| <R> APPENDIX A | DEVELOPMENT TOOLS..... | 647 |
| A.1 | Software Package..... | 649 |
| A.2 | Language Processing Software..... | 649 |
| A.3 | Control Software..... | 649 |
| A.4 | Debugging Tools (Hardware) | 650 |
| A.4.1 | When using IECUBE QB-V850ESFX2..... | 650 |
| A.4.2 | When using MINICUBE QB-V850MINI | 652 |
| A.4.3 | When using MINICUBE2 QB-MINI2 | 653 |
| A.5 | Debugging Tools (Software)..... | 654 |
| A.6 | Embedded Software | 655 |
| A.7 | Flash Memory Writing Tools | 656 |
| APPENDIX B | REGISTER INDEX | 657 |
| APPENDIX C | INSTRUCTION SET LIST | 664 |
| C.1 | Conventions | 664 |
| C.2 | Instruction Set (in Alphabetical Order) | 667 |
| <R> APPENDIX D | LIST OF CAUTIONS..... | 674 |
| <R> APPENDIX E | REVISION HISTORY..... | 703 |
| E.1 | Major Revisions in This Edition | 703 |

CHAPTER 1 INTRODUCTION

The V850ES/HF2 is one of the products in the NEC Electronics V850 single-chip microcontrollers designed for low-power operation for real-time control applications.

1.1 General

The V850ES/HF2 is a 32-bit single-chip microcontroller that includes 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 V850ES/HF2 features multiply instructions, saturated operation instructions, bit manipulation instructions, etc., realized by a hardware multiplier, as optimum instructions for digital servo control applications.

Table 1-1 lists the products of the V850ES/HF2.

Table 1-1. V850ES/HF2 Product List

| Part Number | | μ PD70F3702 | μ PD70F3703 | μ PD70F3704 |
|------------------------------------|---------------|---|-----------------|-----------------|
| Internal memory | Flash memory | 64 KB | 128 KB | 256 KB |
| | RAM | 12 KB | | |
| Memory space | Logical space | 64 MB | | |
| General-purpose register | | 32 bits \times 32 registers | | |
| Main clock (oscillation frequency) | | Ceramic/crystal/external clock <ul style="list-style-type: none"> • In PLL mode: $f_x = 4$ to 5 MHz • In clock through mode: $f_x = 4$ to 5 MHz | | |
| Subclock (oscillation frequency) | | Crystal/external clock: $f_{XT} = 32.768$ kHz RC oscillation: 20 kHz | | |
| Internal oscillator | | $f_R = 200$ kHz (TYP.) | | |
| Minimum instruction execution time | | 50 ns (main clock (f_{xx}) = 20 MHz operation) | | |
| DSP function | | $32 \times 32 = 64$: 200 to 250 ns (at 20 MHz) $32 \times 32 + 32 = 32$: 300 ns (at 20 MHz) $16 \times 16 = 32$: 50 to 100 ns (at 20 MHz) $16 \times 16 + 32 = 32$: 150 ns (at 20 MHz) | | |
| I/O port | | I/O: 67 | | |
| Timer | | 16-bit timer/event counter P: 4 channels 16-bit timer/event counter Q: 1 channel 16-bit interval timer M: 1 channel Watchdog timer 2: 1 channel Watch timer: 1 channel | | |
| A/D converter | | 10-bit resolution \times 12 channels | | |
| Serial interface | | CSIB: 2 channels UARTA (for LIN): 2 channels | | |
| Interrupt source | | External: 9 (9) ^{Note} , internal: 32 | | |
| Power save function | | HALT/IDLE1/IDLE2/STOP/subclock/sub-IDLE mode | | |
| Reset | | RESET pin input, watchdog timer 2 (WDT2), clock monitor (CLM), POC circuit, low-voltage detector (LVI) | | |
| DCU | | Provided (RUN/break) | | |
| Operating power supply voltage | | 3.5 to 5.5 V (A/D converter: 4.0 to 5.5 V) | | |
| Operating ambient temperature | | -40 to +85°C | | |
| Package | | 80-pin plastic TQFP (fine pitch) (12 \times 12 mm) | | |

Note The figure in parentheses indicates the number of external interrupts that can release STOP mode.

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)
 - Internal memory:
 - RAM: 12 KB
 - Flash memory: 64 KB/128 KB/256 KB (see **Table 1-1**)
- Interrupts and exceptions:
 - Non-maskable interrupts: 2 sources
 - Maskable interrupts: 39 sources
 - Software exceptions: 32 sources
 - Exception trap: 2 sources
- I/O lines: I/O ports: 67
- 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 2: 1 channel
- Serial interface:
 - Asynchronous serial interface A (UARTA)
 - 3-wire variable-length serial interface B (CSIB)
 - UARTA (supporting LIN): 2 channels
 - CSIB: 2 channels
- A/D converter: 10-bit resolution: 12 channels
- DCU (debug control unit): JTAG interface
- Clock generator:
 - During main clock or subclock operation
 - 7-level CPU clock (f_{xx}, f_{xx}/2, f_{xx}/4, f_{xx}/8, f_{xx}/16, f_{xx}/32, f_{XT})
 - Clock-through mode/PLL mode selectable
- Internal oscillation clock: 200 kHz (TYP.)
- Power-save functions: HALT/IDLE1/IDLE2/STOP/subclock/sub-IDLE mode
- Package: 80-pin plastic TQFP (fine pitch) (12 × 12)

1.3 Application Fields

Consumer devices

1.4 Ordering Information

| Part Number | Package | On-Chip Flash Memory |
|-------------------------|--|----------------------|
| μ PD70F3702GK-9EU-A | 80-pin plastic TQFP (fine pitch) (12 × 12) | 64 KB |
| μ PD70F3703GK-9EU-A | 80-pin plastic TQFP (fine pitch) (12 × 12) | 128 KB |
| μ PD70F3704GK-9EU-A | 80-pin plastic TQFP (fine pitch) (12 × 12) | 256 KB |

Remark Products with -A at the end of the part number are lead-free products.

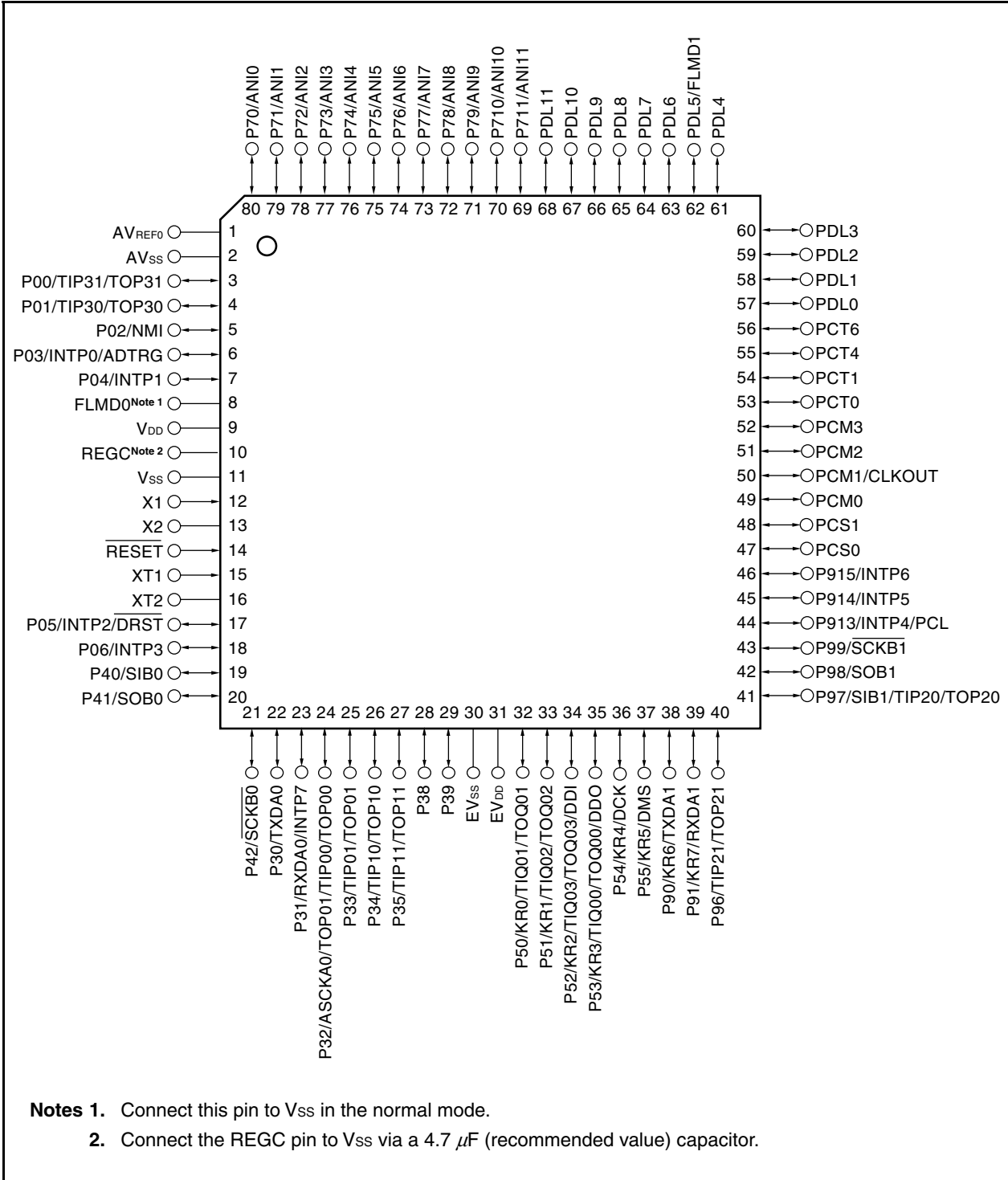
1.5 Pin Configuration (Top View)

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

μPD70F3702GK-9EU-A

μPD70F3704GK-9EU-A

μPD70F3703GK-9EU-A

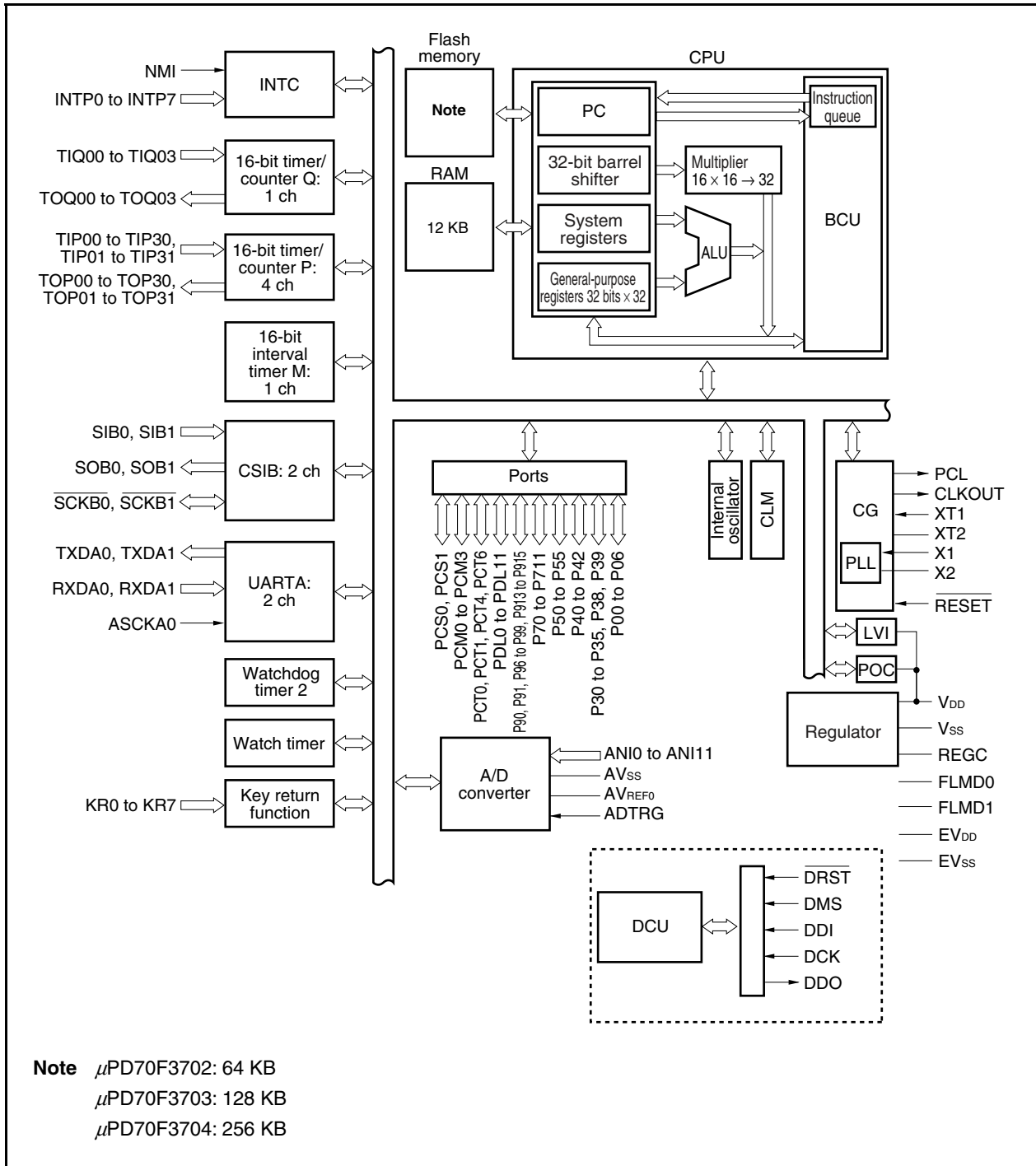


Pin identification

| | | | |
|----------------------|--------------------------------|-------------------|---------------------------|
| ADTRG: | A/D trigger input | PCL: | Programmable clock output |
| ANI0 to ANI11: | Analog input | PCM0 to PCM3: | Port CM |
| ASCKA0: | Asynchronous serial clock | PCS0, PCS1: | Port CS |
| AV _{REF0} : | Analog reference voltage | PCT0, PCT1, | |
| AV _{SS} : | Analog V _{SS} | PCT4, PCT6: | Port CT |
| CLKOUT: | Clock output | PDL0 to PDL11: | Port DL |
| DCK: | Debug clock | REGC: | Regulator control |
| DDI: | Debug data input | RESET: | Reset |
| DDO: | Debug data output | RXDA0, RXDA1: | Receive data |
| DMS: | Debug mode select | SCKB0, SCKB1: | Serial clock |
| DRST: | Debug reset | SIB0, SIB1: | Serial input |
| EV _{DD} : | Power supply for port | SOB0, SOB1: | Serial output |
| EV _{SS} : | Ground for port | TIP00, TIP01, | |
| FLMD0, FLMD1: | Flash programming mode | TIP10, TIP11, | |
| INTP0 to INTP7: | External interrupt request | TIP20, TIP21, | |
| KR0 to KR7: | Key return | TIP30, TIP31, | |
| NMI: | Non-maskable interrupt request | TIQ00 to TIQ03: | Timer input |
| P00 to P06: | Port 0 | TOP00, TOP01, | |
| P30 to P35, | | TOP10, TOP11, | |
| P38, P39: | Port 3 | TOP20, TOP21, | |
| P40 to P42: | Port 4 | TOP30, TOP31, | |
| P50 to P55: | Port 5 | TOQ00 to TOQ03: | Timer output |
| P70 to P711: | Port 7 | TXDA0, TXDA1: | Transmit data |
| P90, P91, | | V _{DD} : | Power supply |
| P96 to P99, | | V _{SS} : | Ground |
| P913 to P915: | Port 9 | X1, X2: | Crystal for main clock |
| | | XT1, XT2: | Crystal for subclock |

1.6 Function Block Configuration

1.6.1 Internal block diagram



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 \times 16 bits \rightarrow 32 bits) and a barrel shifter (32 bits) contribute to faster complex processing.

(2) Bus control unit (BCU)

The BCU controls the internal buses.

(3) ROM

This is a 256 KB/128 KB/64 KB flash memory mapped to addresses 0000000H to 003FFFFH/0000000H to 001FFFFH/0000000H to 000FFFFH. It can be accessed from the CPU in one clock during instruction fetch.

(4) RAM

This is a 12 KB RAM mapped to addresses 3FFC000H to 3FFEFFFH. It can be accessed from 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 multiple servicing control can be performed.

(6) Clock generator (CG)

A main clock oscillator that generates the main clock oscillation frequency (f_x) and a subclock oscillator that generates the subclock oscillation frequency (f_{XT}) are available. As the main clock frequency (f_{XX}), f_x is used as is in the clock-through mode and is multiplied by four in the PLL mode.

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

(7) Internal oscillator

An internal oscillator is provided on chip. The oscillation frequency is 200 kHz (TYP.). An internal oscillator supplies the clock for watchdog timer 2 and timer M.

(8) Timer/counter

Four-channel 16-bit timer/event counter P (TMP), one-channel 16-bit timer/event counter Q (TMQ), and 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 from the 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 internal oscillation 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

The V850ES/HF2 includes three kinds of serial interfaces: asynchronous serial interface A (UARTA) and 3-wire variable-length serial interface B (CSIB).

In the case of UARTA, data is transferred via the TXDA0, TXDA1, RXDA0, and RXDA1 pins.

In the case of CSIB, data is transferred via the SOB0, SOB1, SIB0, SIB1, $\overline{\text{SCKB0}}$, and $\overline{\text{SCKB1}}$ 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) Key interrupt function

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

(14) DCU (debug control unit)

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

(15) Ports

The general-purpose port functions and control pin functions are provided. For details, see **CHAPTER 4 PORT FUNCTIONS**.

CHAPTER 2 PIN FUNCTIONS

This section explains the names and functions of the pins of the V850ES/HF2.

2.1 Pin Function List

Two I/O buffer power supplies, AV_{REF0} and EV_{DD} , are available. The relationship between the power supplies and the pins is shown below.

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

| Power Supply | Corresponding Pin |
|--------------|--|
| AV_{REF0} | Port 7 |
| EV_{DD} | Ports 0, 3 to 5, 9, CM, CS, CT, DL, \overline{RESET} |

(1) Port pins

Table 2-2. List of Pins (Port Pins) (1/3)

| Pin Name | Pin No. | I/O | Function | Alternate Function |
|----------|---------|-----|---|---------------------------|
| P00 | 3 | I/O | Port 0 7-bit I/O port Input/output can be specified in 1-bit units. | TIP31/TOP31 |
| P01 | 4 | | | TIP30/TOP30 |
| P02 | 5 | | | NMI |
| P03 | 6 | | | INTP0/ \overline{ADTRG} |
| P04 | 7 | | | INTP1 |
| P05 | 17 | | | INTP2/ \overline{DRST} |
| P06 | 18 | | | INTP3 |
| P30 | 22 | I/O | Port 3 8-bit I/O port Input/output can be specified in 1-bit units. | TXDA0 |
| P31 | 23 | | | RXDA0/INTP7 |
| P32 | 24 | | | ASCKA0/TIP00/TOP00/TOP01 |
| P33 | 25 | | | TIP01/TOP01 |
| P34 | 26 | | | TIP10/TOP10 |
| P35 | 27 | | | TIP11/TOP11 |
| P38 | 28 | | | - |
| P39 | 29 | | | - |
| P40 | 19 | I/O | Port 4 3-bit I/O port Input/output can be specified in 1-bit units. | SIB0 |
| P41 | 20 | | | $\overline{SOB0}$ |
| P42 | 21 | | | $\overline{SCKB0}$ |

Table 2-2. List of Pins (Port Pins) (2/3)

| Pin Name | Pin No. | I/O | Function | Alternate Function |
|----------|---------|-------|--|---------------------|
| P50 | 32 | I/O | Port 5 6-bit I/O port Input/output can be specified in 1-bit units. | KR0/TIQ01/TOQ01 |
| P51 | 33 | | | KR1/TIQ02/TOQ02 |
| P52 | 34 | | | KR2/TIQ03/TOQ03/DDI |
| P53 | 35 | | | KR3/TIQ00/TOQ00/DDO |
| P54 | 36 | | | KR4/DCK |
| P55 | 37 | | | KR5/DMS |
| P70 | 80 | I/O | Port 7 12-bit I/O port Input/output can be specified in 1-bit units. | ANI0 |
| P71 | 79 | | | ANI1 |
| P72 | 78 | | | ANI2 |
| P73 | 77 | | | ANI3 |
| P74 | 76 | | | ANI4 |
| P75 | 75 | | | ANI5 |
| P76 | 74 | | | ANI6 |
| P77 | 73 | | | ANI7 |
| P78 | 72 | | | ANI8 |
| P79 | 71 | | | ANI9 |
| P710 | 70 | | | ANI10 |
| P711 | 69 | ANI11 | | |
| P90 | 38 | I/O | Port 9 9-bit I/O port Input/output can be specified in 1-bit units. | KR6/TXDA1 |
| P91 | 39 | | | KR7/RXDA1 |
| P96 | 40 | | | TIP21/TOP21 |
| P97 | 41 | | | SIB1/TIP20/TOP20 |
| P98 | 42 | | | SOB1 |
| P99 | 43 | | | SCKB1 |
| P913 | 44 | | | INTP4/PCL |
| P914 | 45 | | | INTP5 |
| P915 | 46 | INTP6 | | |
| PCM0 | 49 | I/O | Port CM 4-bit I/O port Input/output can be specified in 1-bit units. | – |
| PCM1 | 50 | | | CLKOUT |
| PCM2 | 51 | | | – |
| PCM3 | 52 | | | – |
| PCS0 | 47 | I/O | Port CS 2-bit I/O port Input/output can be specified in 1-bit units. | – |
| PCS1 | 48 | | | – |
| PCT0 | 53 | I/O | Port CT 4-bit I/O port Input/output can be specified in 1-bit units. | – |
| PCT1 | 54 | | | – |
| PCT4 | 55 | | | – |
| PCT6 | 56 | | | – |

Table 2-2. List of Pins (Port Pins) (3/3)

| Pin Name | Pin No. | I/O | Function | Alternate Function |
|----------|---------|-----|---|--------------------|
| PDL0 | 57 | I/O | Port DL 12-bit I/O port Input/output can be specified in 1-bit units. | – |
| PDL1 | 58 | | | – |
| PDL2 | 59 | | | – |
| PDL3 | 60 | | | – |
| PDL4 | 61 | | | – |
| PDL5 | 62 | | | FLMD1 |
| PDL6 | 63 | | | – |
| PDL7 | 64 | | | – |
| PDL8 | 65 | | | – |
| PDL9 | 66 | | | – |
| PDL10 | 67 | | | – |
| PDL11 | 68 | | | – |

(2) Non-port pins

Table 2-3. List of Pins (Non-Port Pins) (1/3)

| Pin Name | Pin No. | I/O | Function | Alternate Function |
|---------------------|---------|----------------------|---|-------------------------------|
| NMI ^{Note} | 5 | Input | External interrupt input (non-maskable, with analog noise eliminated) | P02 |
| INTP0 | 6 | Input | External interrupt request input (maskable, with analog noise eliminated) | P03/ADTRG |
| INTP1 | 7 | | | P04 |
| INTP2 | 17 | | | P05/ $\overline{\text{DRST}}$ |
| INTP3 | 18 | | | P06 |
| INTP4 | 44 | | | P913/PCL |
| INTP5 | 45 | | | P914 |
| INTP6 | 46 | | | P915 |
| INTP7 | 23 | | | P31/RXDA0 |
| TIP00 | 24 | Input | External event/clock input (TMP00) | P32/ASCKA0/TOP00/TOP01 |
| TIP01 | 25 | | External event input (TMP01) | P33/TOP01 |
| TIP10 | 26 | | External event/clock input (TMP10) | P34/TOP10 |
| TIP11 | 27 | | External event input (TMP11) | P35/TOP11 |
| TIP20 | 41 | | External event/clock input (TMP20) | P97/SIB1/TOP20 |
| TIP21 | 40 | | External event input (TMP21) | P96/TOP21 |
| TIP30 | 4 | | External event/clock input (TMP30) | P01/TOP30 |
| TIP31 | 3 | | External event input (TMP31) | P00/TOP31 |
| TOP00 | 24 | Output | Timer output (TMP00) | P32/ASCKA0/TIP00/TOP01 |
| TOP01 | 24 | | Timer output (TMP01) | P32/ASCKA0/TIP00/TOP00 |
| | 25 | | | P33/TIP01 |
| TOP10 | 26 | | Timer output (TMP10) | P34/TIP10 |
| TOP11 | 27 | | Timer output (TMP11) | P35/TIP11 |
| TOP20 | 41 | | Timer output (TMP20) | P97/SIB1/TIP20 |
| TOP21 | 40 | | Timer output (TMP21) | P96/TIP21 |
| TOP30 | 4 | | Timer output (TMP30) | P01/TIP30 |
| TOP31 | 3 | Timer output (TMP31) | P00/TIP31 | |
| TIQ00 | 35 | Input | External event/clock input (TMQ00) | P53/KR3/TOQ00/DDO |
| TIQ01 | 32 | | External event input (TMQ01) | P50/KR0/TOQ01 |
| TIQ02 | 33 | | External event input (TMQ02) | P51/KR1/TOQ02 |
| TIQ03 | 34 | | External event input (TMQ03) | P52/KR2/TOQ03/DDI |
| TOQ00 | 35 | Output | Timer output (TMQ00) | P53/KR3/TOQ00/DDO |
| TOQ01 | 32 | | Timer output (TMQ01) | P50/KR0/TOQ01 |
| TOQ02 | 33 | | Timer output (TMQ02) | P51/KR1/TOQ02 |
| TOQ03 | 34 | | Timer output (TMQ03) | P52/KR2/TOQ03/DDI |

Note The NMI pin alternately functions as the P02 pin. It functions as the P02 pin after reset. To enable the NMI pin, set the PMC0.PMC02 bit to 1. The initial setting of the NMI pin is "No edge detected". Select the NMI pin valid edge using INTF0 and INTRO registers.

Table 2-3. List of Pins (Non-Port Pins) (2/3)

| Pin Name | Pin No. | I/O | Function | Alternate Function |
|---------------------------|---------|--------|---|-----------------------|
| SIB0 | 19 | Input | Serial receive data input (CSIB0) | P40 |
| SIB1 | 41 | | Serial receive data input (CSIB1) | P97/TIP20/TOP20 |
| SOB0 | 20 | Output | Serial transmit data output (CSIB0) | P41 |
| SOB1 | 42 | | Serial transmit data output (CSIB1) | P98 |
| $\overline{\text{SCKB0}}$ | 21 | I/O | Serial clock I/O (CSIB0) | P42 |
| $\overline{\text{SCKB1}}$ | 43 | | Serial clock I/O (CSIB1) | P99 |
| RXDA0 | 23 | Input | Serial receive data input (UARTA0) | P31/INTP7 |
| RXDA1 | 39 | | Serial receive data input (UARTA1) | P91/KR7 |
| TXDA0 | 22 | Output | Serial transmit data output (UARTA0) | P30 |
| TXDA1 | 38 | | Serial transmit data output (UARTA1) | P90/KR6 |
| ASCKA0 | 24 | Input | Baud rate clock input to UARTA0 | P32/TIP00/TOP00/TOP01 |
| ANI0 | 80 | Input | Analog voltage input to A/D converter | P70 |
| ANI1 | 79 | | | P71 |
| ANI2 | 78 | | | P72 |
| ANI3 | 77 | | | P73 |
| ANI4 | 76 | | | P74 |
| ANI5 | 75 | | | P75 |
| ANI6 | 74 | | | P76 |
| ANI7 | 73 | | | P77 |
| ANI8 | 72 | | | P78 |
| ANI9 | 71 | | | P79 |
| ANI10 | 70 | | | P710 |
| ANI11 | 69 | | | P711 |
| AV _{REF0} | 1 | – | Reference voltage input to A/D converter, positive power supply for alternate-function port 7 | – |
| AV _{SS} | 2 | – | Ground potential for A/D converter (same potential as V _{SS}) | – |
| ADTRG | 6 | Input | A/D converter external trigger input | P03/INTP0 |
| KR0 | 32 | Input | Key interrupt input | P50/TIQ01/TOQ01 |
| KR1 | 33 | | | P51/TIQ02/TOQ02 |
| KR2 | 34 | | | P52/TIQ03/TOQ03/DDI |
| KR3 | 35 | | | P53/TIQ00/TOQ00/DDO |
| KR4 | 36 | | | P54/DCK |
| KR5 | 37 | | | P55/DMS |
| KR6 | 38 | | | P90/TXDA1 |
| KR7 | 39 | | | P91/RXDA1 |
| DMS | 37 | Input | Debug mode select | P55/KR5 |
| DDI | 34 | Input | Debug data input | P52/KR2/TIQ03/TOQ03 |
| DDO | 35 | Output | Debug data output | P53/KR3/TIQ00/TOQ00 |
| DCK | 36 | Input | Debug clock input | P54/KR4 |
| $\overline{\text{DRST}}$ | 17 | Input | Debug reset input | P05/INTP2 |

Table 2-3. List of Pins (Non-Port Pins) (3/3)

| Pin Name | Pin No. | I/O | Function | Alternate Function |
|---------------------------|---------|--------|---|--------------------|
| FLMD0 | 8 | Input | Flash programming mode setting pins | – |
| FLMD1 | 62 | | | PDL5 |
| CLKOUT | 50 | Output | Internal system clock output | PCM1 |
| PCL | 44 | Output | Clock output (timing output of X1 input clock and subclock) | P913/INTP4 |
| REGC | 10 | – | Regulator output stabilizing capacitor connection | – |
| $\overline{\text{RESET}}$ | 14 | Input | System reset input | – |
| X1 | 12 | Input | Main clock resonator connection | – |
| X2 | 13 | – | | – |
| XT1 | 15 | Input | Subclock resonator connection | – |
| XT2 | 16 | – | | – |
| V _{DD} | 9 | – | Positive power supply pin for internal circuitry | – |
| V _{SS} | 11 | – | Ground potential for internal circuitry | – |
| EV _{DD} | 31 | – | Positive power supply pin for external circuitry (same potential as V _{DD}) | – |
| EV _{SS} | 30 | – | Ground potential for external circuitry (same potential as V _{SS}) | – |

2.2 Description of Pin Functions

(1) P00 to P06 (port 0) ... 3-state I/O

P00 to P06 function as a 7-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as NMI input, external interrupt request signal input, timer/counter I/O, external trigger of the A/D converter, and debug reset input.

This port can be set in the port mode or control mode in 1-bit units. The valid edge of each pin is specified by the INTR0 and INTF0 registers.

An on-chip pull-up resistor can be connected to P00 to P06 by using pull-up resistor option register 0 (PU0).

(a) Port mode

P00 to P06 can be set in the input or output mode in 1-bit units, by using port mode register 0 (PM0).

(b) Control mode

(i) NMI (Non-maskable interrupt request) ... input

This pin inputs a non-maskable interrupt request signal.

(ii) INTP0 to INTP3 (External interrupt request) ... input

These pins input external interrupt request signals.

(iii) TIP30, TIP31 (Timer input) ... input

These pins input an external count clock to timer P3 (TMP3).

(iv) TOP30, TOP31 (Timer output) ... output

These pins output a pulse signal from timer P3 (TMP3).

(v) ADTRG (A/D trigger input) ... input

This pin inputs an external trigger to the A/D converter. It is controlled by using A/D converter mode register 0 (ADA0M0).

(vi) $\overline{\text{DRST}}$ (Debug reset) ... input

This pin inputs a debug reset signal, a negative-logic signal that asynchronously initializes the debug control unit (DCU). To deassert this signal, reset or invalidate the DCU. Deassert this signal when the debug function is not used.

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

(2) P30 to P35, P38, P39 (port 3) ... 3-state I/O

P30 to P35, P38, and P39 function as an 8-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, P30 to P35 operate as external interrupt request signal input, serial interface I/O, and timer/counter I/O. This port can be set in the port mode or control mode in 1-bit units. The valid edge of each pin is specified by the INTR3 and INTF3 registers.

An on-chip pull-up resistor can be connected to P30 to P35, P38, and P39 by using pull-up resistor option register 3 (PU3).

(a) Port mode

P30 to P35, P38, and P39 can be set in the input or output mode in 1-bit units, by using port mode register 3 (PM3).

(b) Control mode**(i) RXDA0 (Receive data) ... input**

This pin inputs the serial receive data of UARTA0.

(ii) TXDA0 (Transmit data) ... output

This pin outputs the serial transmit data of UARTA0.

(iii) ASCKA0 (Asynchronous serial clock) ... input

This is an input pin for UARTA0.

(iv) INTTP7 (External interrupt request) ... input

This pin inputs an external interrupt request signal.

(v) TIP00, TIP01, TIP10, TIP11 (Timer input) ... input

These are input pins for timers P0 and P1 (TMP0 and TMP1).

(vi) TOP00, TOP01, TOP10, TOP11 (Timer output) ... output

These are output pins for timers P0 and P1 (TMP0 and TMP1).

(3) P40 to P42 (port 4) ... 3-state I/O

P40 to P42 function as a 3-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as serial interface I/O. This port can be set in the port mode or control mode in 1-bit units.

An on-chip pull-up resistor can be connected to P40 to P42 by using pull-up resistor option register 4 (PU4).

(a) Port mode

P40 to P42 can be set in the input or output mode in 1-bit units, by using port mode register 4 (PM4).

(b) Control mode**(i) SIB0 (Serial input) ... input**

This pin inputs the serial receive data of CSIB0.

(ii) SOB0 (Serial output) ... output

This pin outputs the serial transmit data of CSIB0.

(iii) $\overline{\text{SCKB0}}$ (serial clock) ... 3-state I/O

This pin inputs/outputs the serial clock of CSIB0.

(4) P50 to P55 (Port 5) ... 3-state I/O

P50 to P55 function as a 6-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as timer/counter I/O, debug function I/O, and key interrupt input. This port can be set in the port mode or control mode in 1-bit units.

An on-chip pull-up resistor can be connected to P50 to P55 by using pull-up resistor option register 5 (PU5).

(a) Port mode

P50 to P55 can be set in the input or output mode in 1-bit units, by using port mode register 5 (PM5).

(b) Control mode**(i) KR0 to KR5 (Key return) ... input**

These pins input a key interrupt. Their operation is specified by using the key return mode register (KRM) in the input port mode.

(ii) TIQ00, TIQ01, TIQ02, TIQ03 (Timer input) ... input

These are input pins for timer Q0 (TMQ0).

(iii) TOQ00, TOQ01, TOQ02, TOQ03 (Timer output) ... output

These are output pins for timer Q0 (TMQ0).

(iv) DDI (Debug data input) ... input

This pin inputs debug data to the debug control unit (DCU).

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

(v) DDO (Debug data output) ... output

This pin outputs debug data from the DCU.

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

(vi) DCK (Debug clock input) ... input

This pin inputs a debug clock to the DCU.

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

(vii) DMS (Debug mode select) ... input

This pin selects the debug mode of the DCU.

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

(5) P70 to P711 (port 7) ... 3-state I/O

P70 to P711 function as a 12-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as analog input to the A/D converter in the control mode.

When using the analog input pins, however, set this port in the input mode. At this time, do not read the port.

(a) Port mode

P70 to P711 can be set in the input or output mode in 1-bit units, by using port mode registers 7L and 7H (PM7L and PM7H).

(b) Control mode

P70 to P711 function alternately as the ANI0 to ANI11 pins.

(i) ANI0 to ANI11 (Analog input 0 to 11) ... input

These pins input an analog signal to the A/D converter.

(6) P90, P91, P96 to P99, P913 to P915 (port 9) ... 3-state I/O

P90, P91, P96 to P99, and P913 to P915 function as a 9-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as serial interface I/O, timer/counter I/O, clock output, external interrupt request signal input, and key interrupt input. This port can be set in the port mode or control mode in 1-bit units. The valid edge of P913 to P915 is specified by INTR9H and INTF9H registers.

An on-chip pull-up resistor can be connected to P90, P91, P96 to P99, and P913 to P915 by using pull-up resistor option register 9 (PU9).

(a) Port mode

P90, P91, P96 to P99, and P913 to P915 can be set in the input or output mode in 1-bit units, by using port mode register 9 (PM9).

(b) Control mode**(i) SIB1 (Serial input) ... input**

This pin inputs the serial receive data of CSIB1.

(ii) SOB1 (Serial output) ... output

This pin outputs the serial transmit data of CSIB1.

(iii) $\overline{\text{SCKB1}}$ (Serial clock) ... 3-state I/O

This pin inputs/outputs the serial clock of CSIB1.

(iv) RXDA1 (Receive data) ... input

This pin inputs the serial receive data of UARTA1.

(v) TXDA1 (Transmit data) ... output

This pin outputs the serial transmit data of UARTA1.

(vi) TIP20, TIP21 (Timer input) ... input

These are input pins for timer P2 (TMP2).

(vii) TOP20, TOP21 (Timer output) ... output

These are output pins for timer P2 (TMP2).

(viii) PCL (Clock output) ... output

This pin outputs a clock.

(ix) INTP4 to INTP6 (External interrupt request) ... input

These pins input an external interrupt request signal.

(x) KR6, KR7 (Key return) ... input

These pins input a key interrupt. Their operation is specified by the key return mode register (KRM) in the input port mode.

(7) PCM0 to PCM3 (port CM) ... 3-state I/O

PCM0 to PCM3 function as a 4-bit I/O port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, these pins operate as bus clock output in the control mode.

(a) Port mode

PCM0 to PCM3 can be set in the input or output mode in 1-bit units, by using port mode register CM (PMCM).

(b) Control mode**(i) CLKOUT (Clock output) ... output**

This pin outputs an internally generated bus clock.

(8) PCS0, PCS1 (port CS) ... 3-state I/O

PCS0 and PCS1 function as a 2-bit I/O port that can be set to input or output in 1-bit units.

(a) Port mode

PCS0 and PCS1 can be set in the input or output mode in 1-bit units, by using port mode register CS (PMCS).

(9) PCT0, PCT1, PCT4, PCT6 (port CT) ... 3-state I/O

PCT0, PCT1, PCT4, and PCT6 function as a 4-bit I/O port that can be set to input or output in 1-bit units.

(a) Port mode

PCT0, PCT1, PCT4, and PCT6 can be set in the input or output mode in 1-bit units, by using port mode register CT (PMCT).

(10) PDL0 to PDL11 (port DL) ... 3-state I/O

PDL0 to PDL11 function as a 12-bit I/O port that can be set to input or output in 1-bit units.

PDL5 also functions as the FLMD1 pin when the flash memory is programmed (when a high level is input to FLMD0). At this time, be sure to input a low level to the FLMD1 pin.

(a) Port mode

PDL0 to PDL11 can be set in the input or output mode in 1-bit units, by using port mode register DL (PMDL).

(11) $\overline{\text{RESET}}$ (Reset) ... input

$\overline{\text{RESET}}$ input is asynchronous input. When a signal with a fixed low level width is input to the $\overline{\text{RESET}}$ pin regardless of the operating clock, the system is reset, taking precedence over all the other operations.

This pin is used to release the standby mode (HALT, IDLE, or STOP), as well as for normal initialization/start.

(12) X1, X2 (Crystal for main clock)

These pins are used to connect the resonator that generates the system clock.

(13) XT1, XT2 (Crystal for subclock)

These pins are used to connect the resonator that generates the subclock.

(14) AV_{SS} (Ground for analog)

This is a ground pin for the A/D converter and alternate-function ports.

(15) AV_{REF0} (Analog reference voltage) ... input

This pin supplies positive analog power to the A/D converter and alternate-function ports. It also supplies a reference voltage to the A/D converter.

(16) EV_{DD} (Power supply for port)

This pin supplies positive power to the I/O ports and alternate-function pins.

(17) EV_{SS} (Ground for port)

This is a ground pin for the I/O ports and alternate-function pins.

(18) V_{DD} (Power supply)

This pin supplies positive power. Connect all the V_{DD} pins to a positive power supply.

(19) V_{SS} (Ground)

This is a ground pin. Connect all the V_{SS} pins to ground.

(20) FLMD0 (Flash programming mode) ... input

This is a signal input pin for flash memory programming mode. Connect this pin to V_{SS} in the normal operation mode.

(21) REGC (Regulator control) ... input

This pin connects a capacitor for the regulator.

2.3 Pin I/O Circuit Types and Recommended Connection of Unused Pins

(1/2)

| Pin | Pin No. | I/O Circuit Type | Recommended Connection | |
|-----------------------------------|-------------|------------------|---|---|
| P00/TIP31/TOP31 | 3 | 5-W | Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open | |
| P01/TIP30/TOP30 | 4 | | | |
| P02/NMI | 5 | | | |
| P03/INTP0/ADTRG | 6 | | | |
| P04/INTP1 | 7 | | | |
| P05/INTP2/ \overline{DRST} | 17 | 5-AF | Input: Independently connect to EV_{SS} Output: Leave open | |
| P06/INTP3 | 18 | 5-W | Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open | |
| P30/TXDA0 | 22 | 5-A | Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open | |
| P31/RXDA0/INTP7 | 23 | 5-W | | |
| P32/ASCKA0/TIP00/ TOP00/ TOP01 | 24 | | | |
| P33/TIP01/TOP01 | 25 | | | |
| P34/TIP10/TOP10 | 26 | | | |
| P35/TIP11/TOP11 | 27 | | | |
| P38 | 28 | | | 5-A |
| P39 | 29 | | | |
| P40/SIB0 | 19 | 5-W | | Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open |
| P41/SOB0 | 20 | 5-A | | |
| P42/ $\overline{SCKB0}$ | 21 | 5-W | | |
| P50/KR0/TIQ01/TOQ01 | 32 | 5-W | Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open | |
| P51/KR1/TIQ02/TOQ02 | 33 | | | |
| P52/KR2/TIQ03/TOQ03/DDI | 34 | | | |
| P53/KR3/TIQ00/TOQ00/ DDO | 35 | | | |
| P54/KR4/DCK | 36 | | | |
| P55/KR5/DMS | 37 | | | |
| P70/ANI0 to P711/ANI11 | 80 to 69 | 11-G | Input: Independently connect to AV_{REF0} or AV_{SS} via a resistor Output: Leave open | |
| P90/KR6/TXDA1 | 38 | 5-W | Input: Independently connect to EV_{DD} or EV_{SS} via a resistor Output: Leave open | |
| P91/KR7/RXDA1 | 39 | | | |
| P96/TIP21/TOP21 | 40 | | | |
| P97/SIB1/TIP20/TOP20 | 41 | | | |
| P98/SOB1 | 42 | 5-A | | |
| P99/ $\overline{SCKB1}$ | 43 | 5-W | | |
| P913/INTP4/PCL | 44 | | | |
| P914/INTP5 | 45 | | | |
| P915/INTP6 | 46 | | | |

| Pin | Pin No. | I/O Circuit Type | Recommended Connection |
|------------------------|----------|------------------|---|
| PCM0 | 49 | 5 | Input: Independently connect to EV _{DD} or EV _{SS} via a resistor Output: Leave open |
| PCM1/CLKOUT | 50 | | |
| PCM2, PCM3 | 51, 52 | | |
| PCS0, PCS1 | 47, 48 | 5 | Input: Independently connect to EV _{DD} or EV _{SS} via a resistor Output: Leave open |
| PCT0, PCT1, PCT4, PCT6 | 53 to 56 | 5 | Input: Independently connect to EV _{DD} or EV _{SS} via a resistor Output: Leave open |
| PDL0 to PDL4 | 57 to 61 | 5 | Input: Independently connect to EV _{DD} or EV _{SS} via a resistor Output: Leave open |
| PDL5/FLMD1 | 62 | | |
| PDL6 to PDL11 | 63 to 68 | | |
| AV _{REF0} | 1 | – | Directly connect to V _{DD} |
| AV _{SS} | 2 | – | – |
| FLMD0 ^{Note} | 8 | – | Directly connect to V _{SS} |
| REGC | 10 | – | – |
| RESET | 14 | 2 | – |
| X1 | 12 | – | – |
| X2 | 13 | – | – |
| XT1 | 15 | 16 | Connect to V _{SS} via a resistor |
| XT2 | 16 | 16 | Leave open |
| V _{DD} | 9 | – | – |
| V _{SS} | 11 | – | – |
| EV _{DD} | 31 | – | – |
| EV _{SS} | 30 | – | – |

Note If noise that exceeds the noise elimination width is input to the $\overline{\text{RESET}}$ pin during self programming, the flash on-board mode may be entered depending on the capacitance charge end timing when a capacitor is connected to the FLMD0 pin. Therefore, do not connect a capacitor to the FLMD0 pin.

2.4 Pin I/O Circuits

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

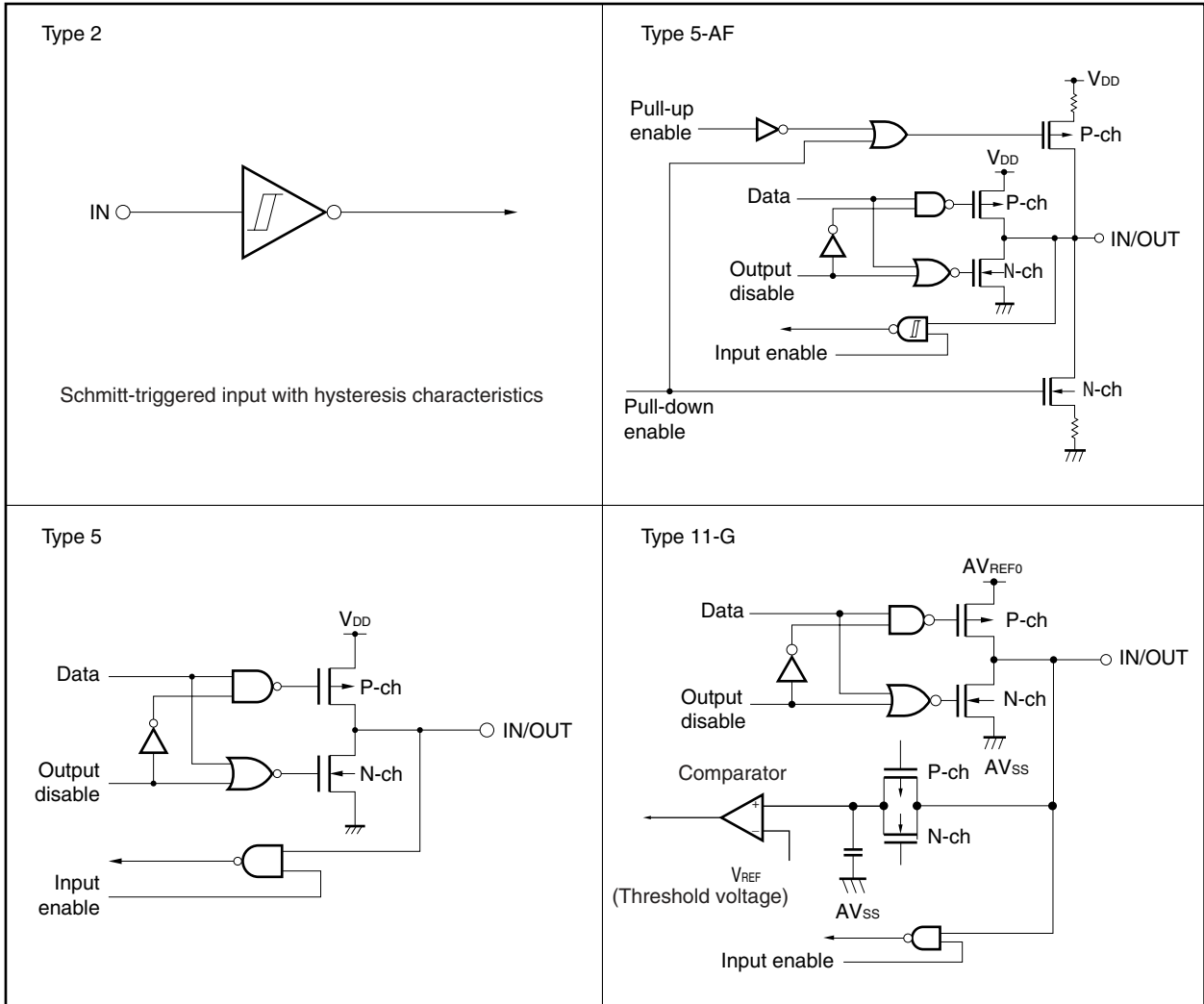
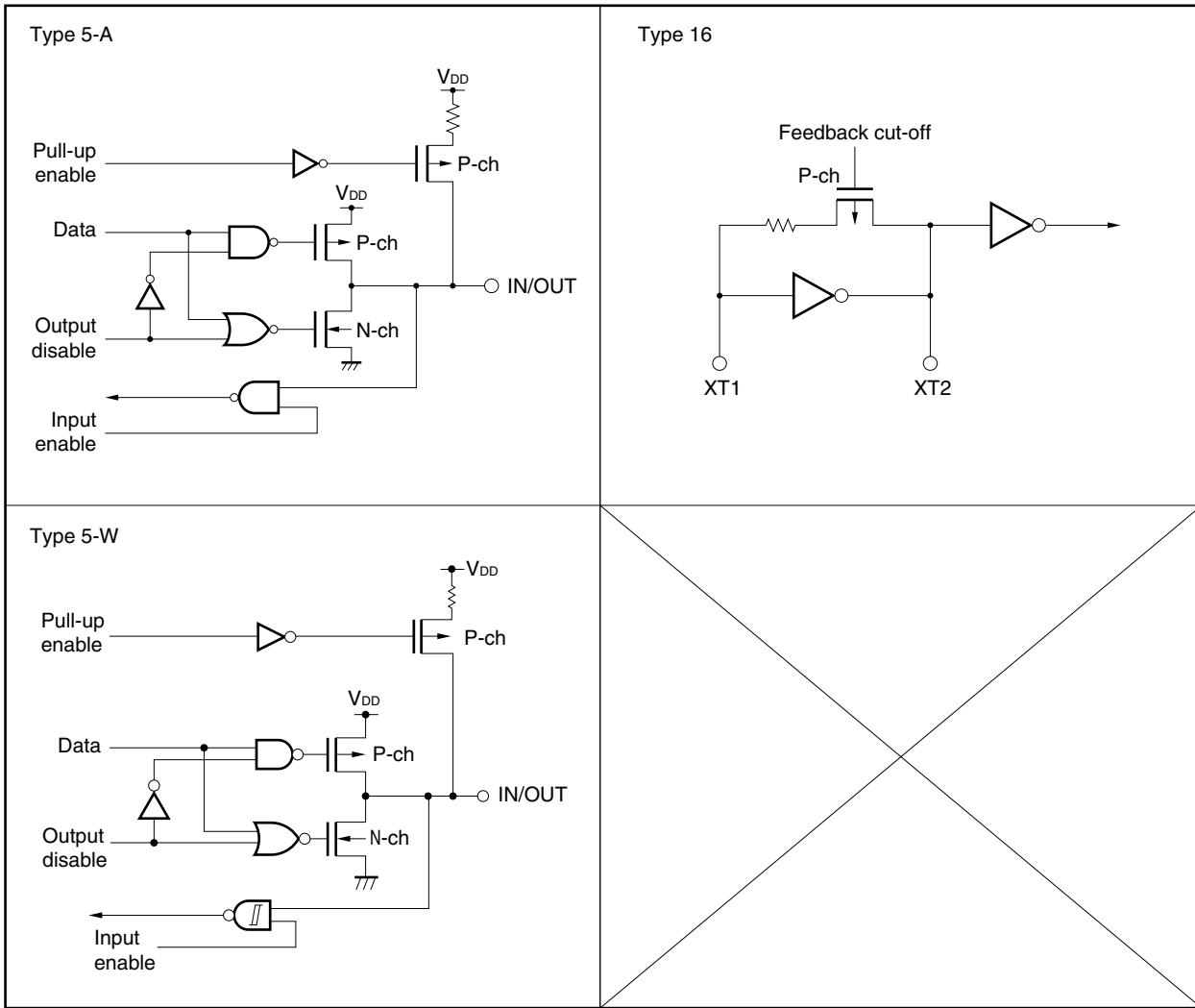


Figure 2-1. Pin I/O Circuit Types (2/2)



Remark Read V_{DD} as EV_{DD} . Also, read V_{SS} as EV_{SS} .

<R> **2.5 Caution**

Note that the following pin may temporarily output an undefined level, even during reset upon power application.
P53/KR3/TIQ00/TOQ00/DDO pin

CHAPTER 3 CPU FUNCTION

The CPU of the V850ES/HF2 is based on RISC architecture and executes almost all instructions with one clock by using a 5-stage pipeline.

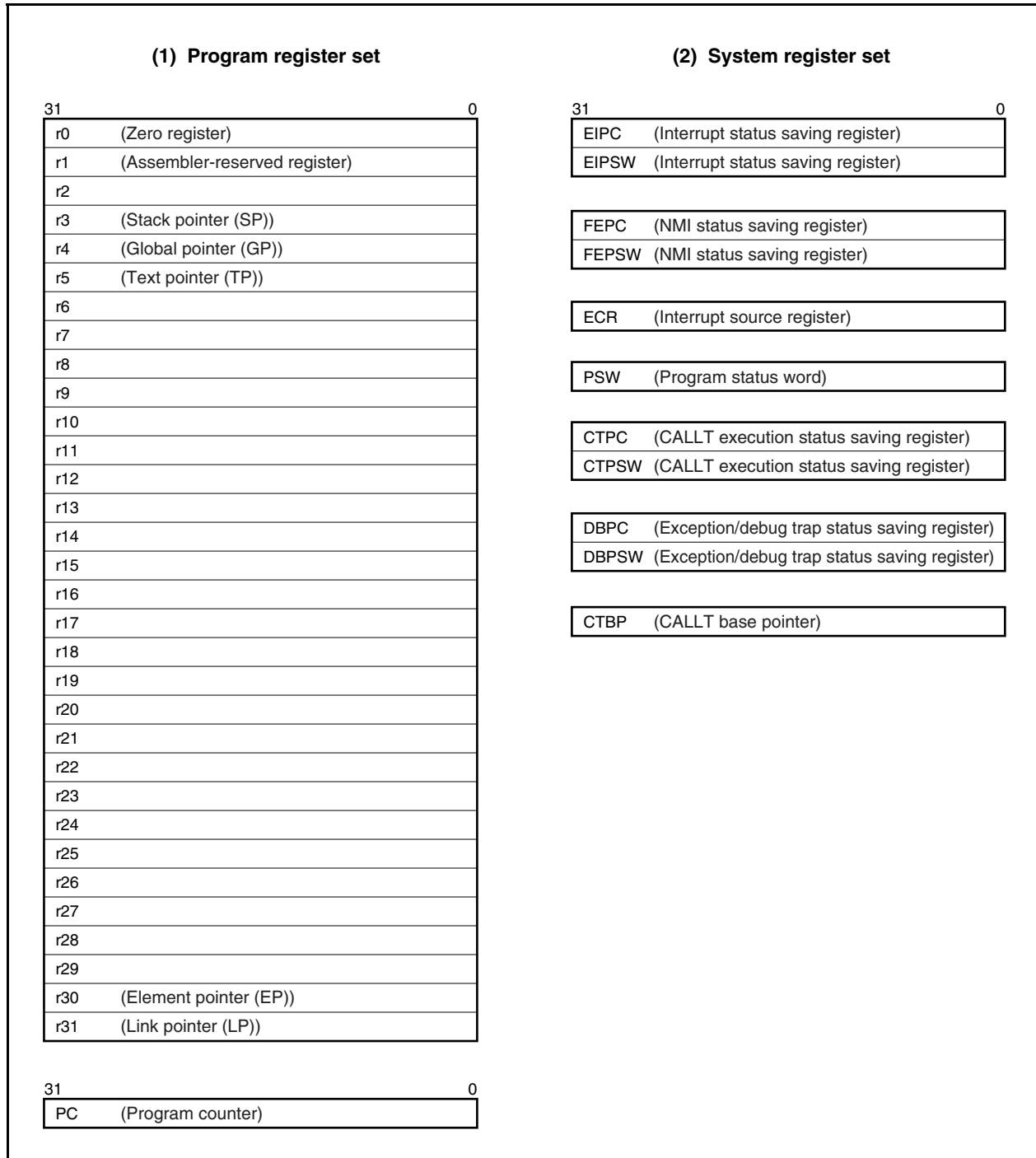
3.1 Features

- Minimum instruction execution time: 50 ns (at 20 MHz operation)
- Memory space Program (physical address) space: 64 MB linear
 Data (logical address) space: 4 GB linear
- 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 V850ES/HF2 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**.



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.

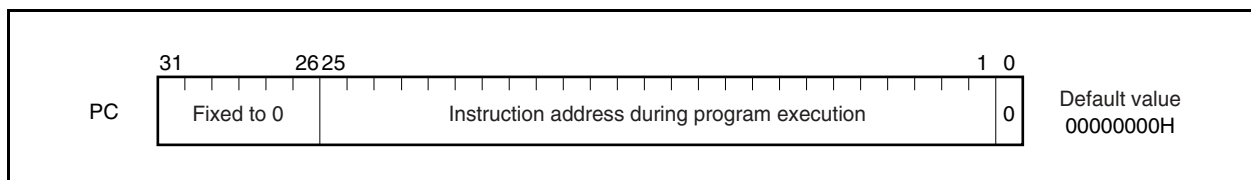
Table 3-1. Program Registers

| 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 |

Remark For further details on the r1, r3 to r5, and r31 that are used in the assembler and C compiler, refer to the **CA850 (C Compiler Package) Assembly Language User's Manual**.

(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} | √ ^{Note 2} |
| 19 | Exception/debug trap status saving register (DBPSW) | √ ^{Note 2} | √ ^{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 during the interval between the execution of the DBTRAP instruction or illegal opcode and the DBRET instruction.

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 occurs).

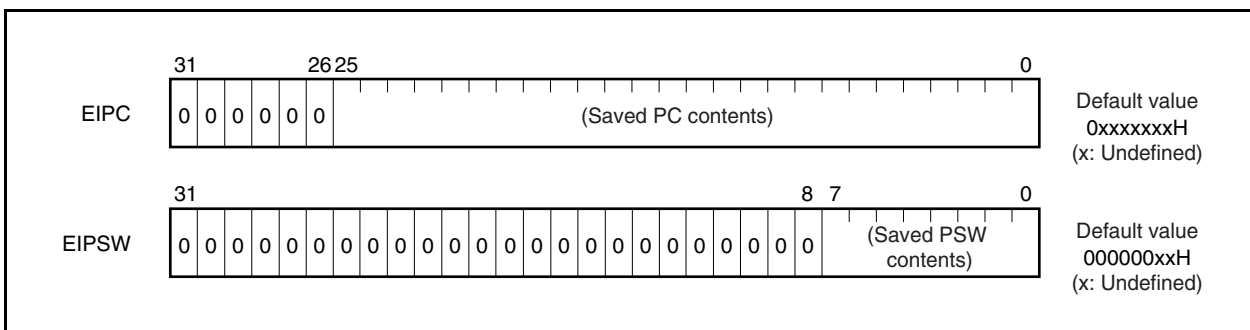
The address of the instruction next to the instruction under execution, except some instructions (see **14.8 Periods in Which Interrupts Are Not Acknowledged by CPU**), is saved to EIPC when a software exception or a maskable interrupt occurs.

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).

The value of EIPC is restored to the PC and the value of EIPSW to the PSW by the RETI instruction.



(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.

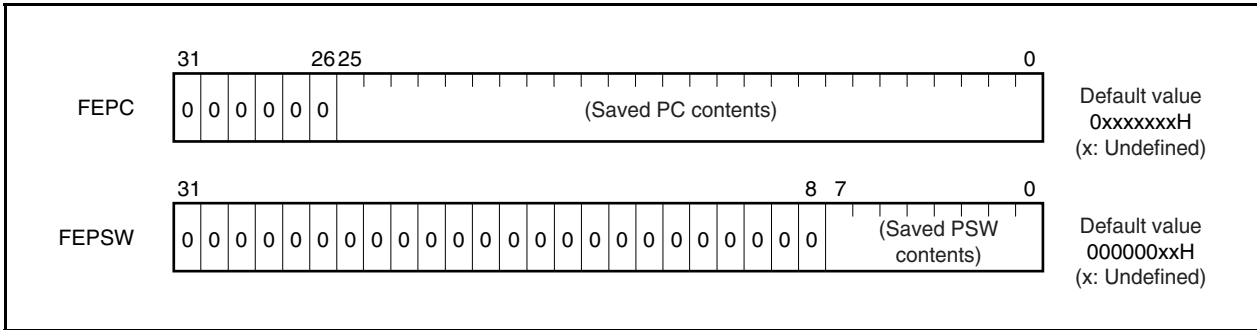
The address of the instruction next to the one of the instruction under execution, except some instructions, is saved to FEPC when an NMI occurs.

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.

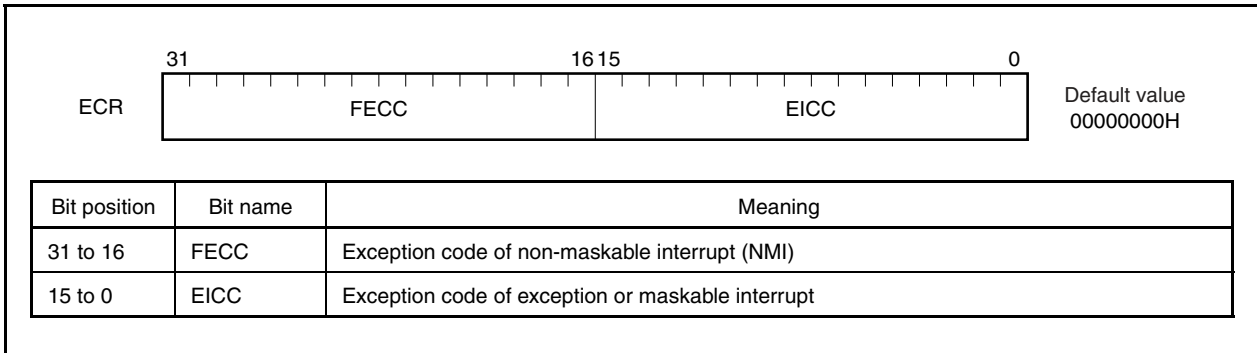
Bits 31 to 26 of FEPC and bits 31 to 8 of FEPSW are reserved for future function expansion (these bits are always fixed to 0).

The value of FEPC is restored to the PC and the value of FEPSW to the PSW by the RETI instruction.



(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.



(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. However if the ID flag is set to 1, interrupt requests will not be acknowledged while the LDSR instruction is being executed.

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

(1/2)

| PSW | <div style="display: flex; justify-content: space-between; align-items: center;"> 31 8 7 6 5 4 3 2 1 0 </div> <div style="border: 1px solid black; padding: 5px; display: flex; justify-content: space-between; align-items: center;"> RFU <table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td style="width: 15px;">NP</td> <td style="width: 15px;">EP</td> <td style="width: 15px;">ID</td> <td style="width: 15px;">SAT</td> <td style="width: 15px;">CY</td> <td style="width: 15px;">OV</td> <td style="width: 15px;">S</td> <td style="width: 15px;">Z</td> </tr> </table> </div> | NP | EP | ID | SAT | CY | OV | S | Z | Default value 00000020H |
|---|---|--|-----|----|-----|----|----|---|---|----------------------------|
| NP | EP | ID | SAT | CY | OV | S | Z | | | |
| 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. | | | | | | | | | | |

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)

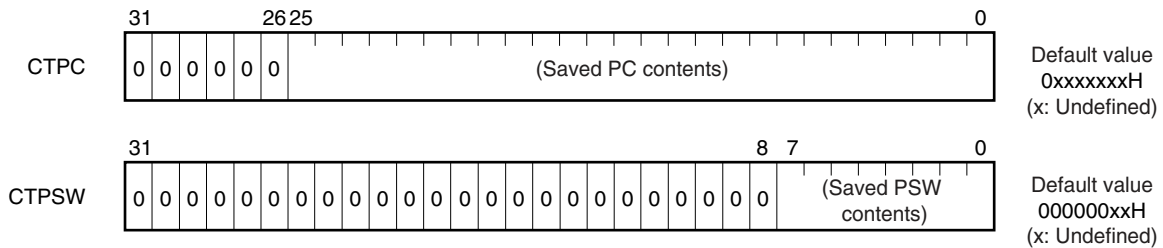
CTPC and CTPSW are CALLT execution status saving registers.

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 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.

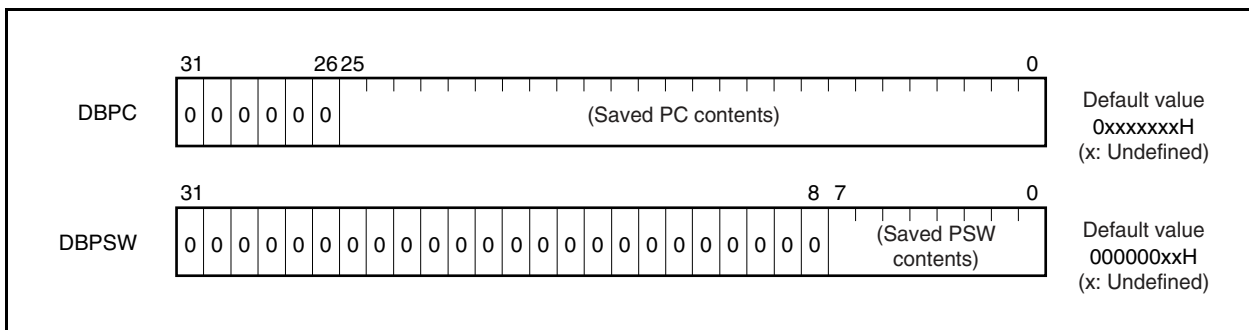
The contents to be saved to DBPC are the address of the instruction next to the one that is being executed when an exception trap or debug trap occurs.

The current contents of the PSW are saved to DBPSW.

This register can be read or written only during the interval between the execution of the DBTRAP instruction or illegal opcode and the DBRET instruction.

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

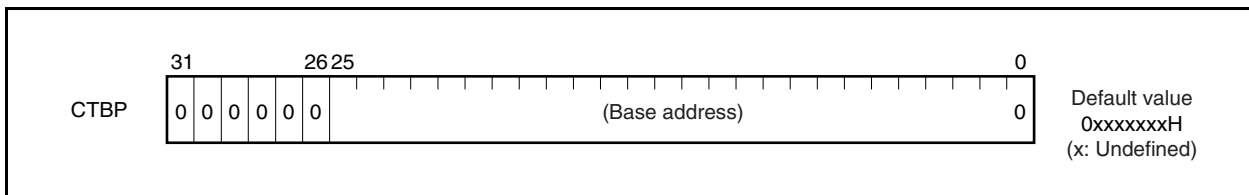
The value of DBPC is restored to the PC and the value of DBPSW to the PSW by the DBRET instruction.



(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 V850ES/HF2 has the following operation modes.

(1) Normal operation mode

In this mode, execution branches to the reset entry address of the internal ROM after system reset has been released, and then instruction processing is started.

(2) Flash memory programming mode

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

(3) On-chip debug mode

The V850ES/HF2 is provided with an on-chip debug function that employs the JTAG (Joint Test Action Group) communication specifications.

For details, see **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, input a low level to the FLMD0 pin 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 |

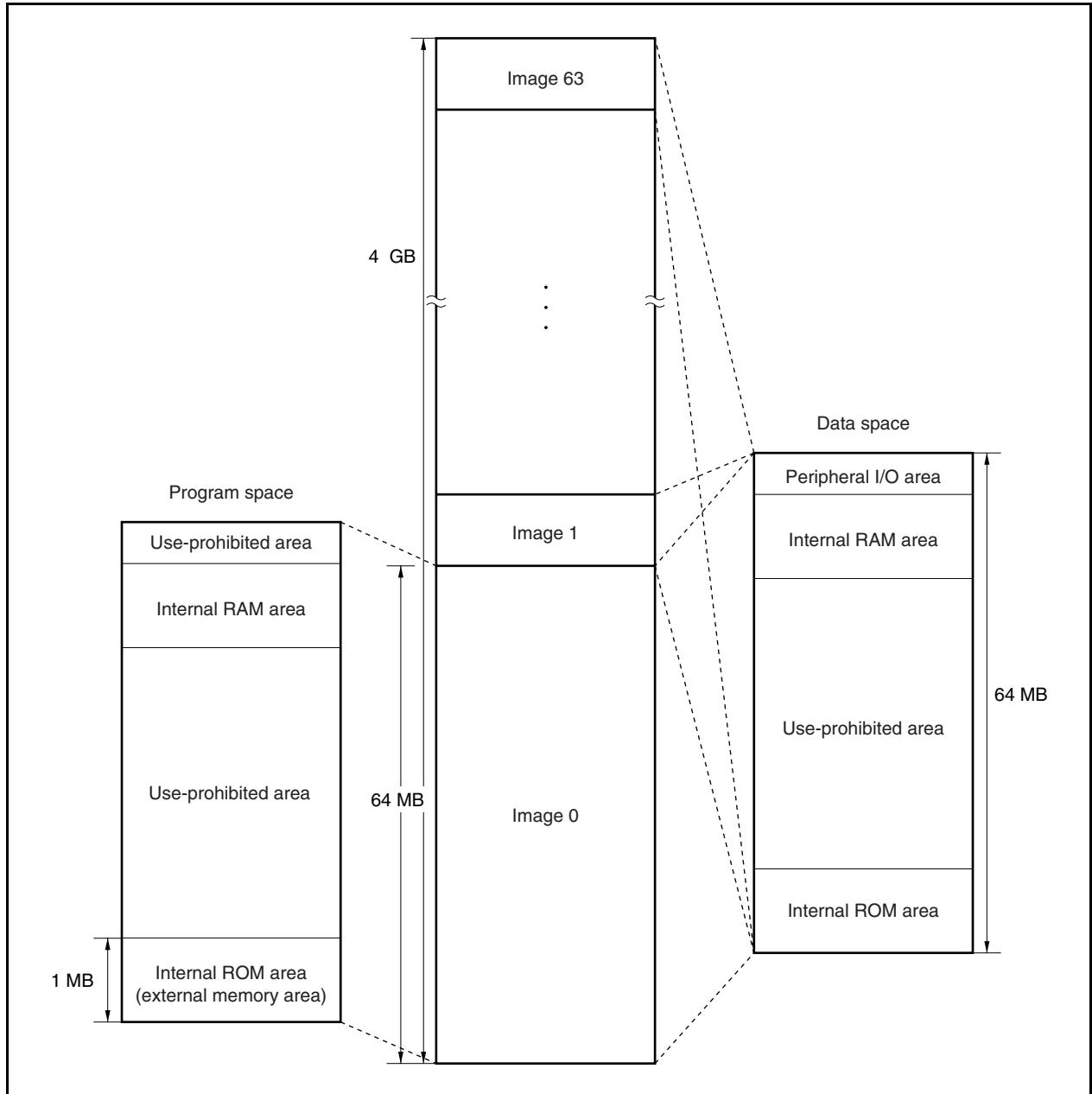
Remark L: Low-level input
H: High-level input
×: Don't care

3.4 Address Space

3.4.1 CPU address space

For instruction addressing, an internal ROM area of up to 1 MB, and an internal RAM area are supported in a linear address space (program space) of up to 64 MB. For operand addressing (data access), up to 4 GB of a linear address space (data space) is supported. The 4 GB address space, however, is viewed as 64 images of a 64 MB physical address space. This means that the same 64 MB physical address space is accessed regardless of the value of bits 31 to 26.

Figure 3-1. Image on Address Space



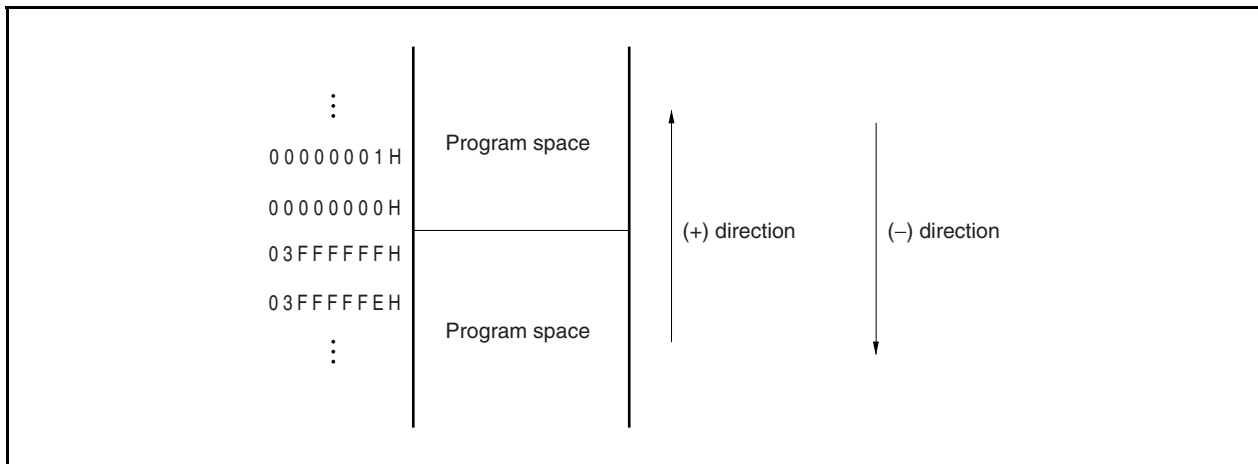
3.4.2 Wraparound of CPU address space

(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 highest address of the program space, 03FFFFFFH, and the lowest address, 00000000H, are contiguous addresses. That the highest address and the lowest address of the program space are contiguous in this way is called wraparound.

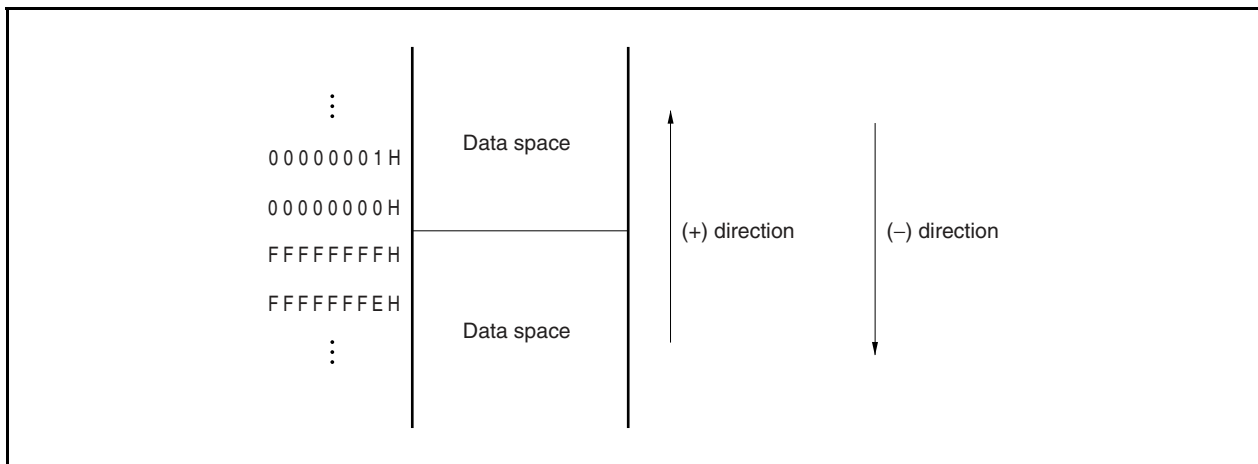
Caution Because the 4 KB area of addresses 03FFF000H to 03FFFFFFH is an on-chip peripheral I/O area, instructions cannot be fetched from this area. Therefore, do not execute an operation in which the result of a branch address calculation affects this area.



(2) Data space

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

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



3.4.3 Memory map

The areas shown below are reserved in the V850ES/HF2.

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

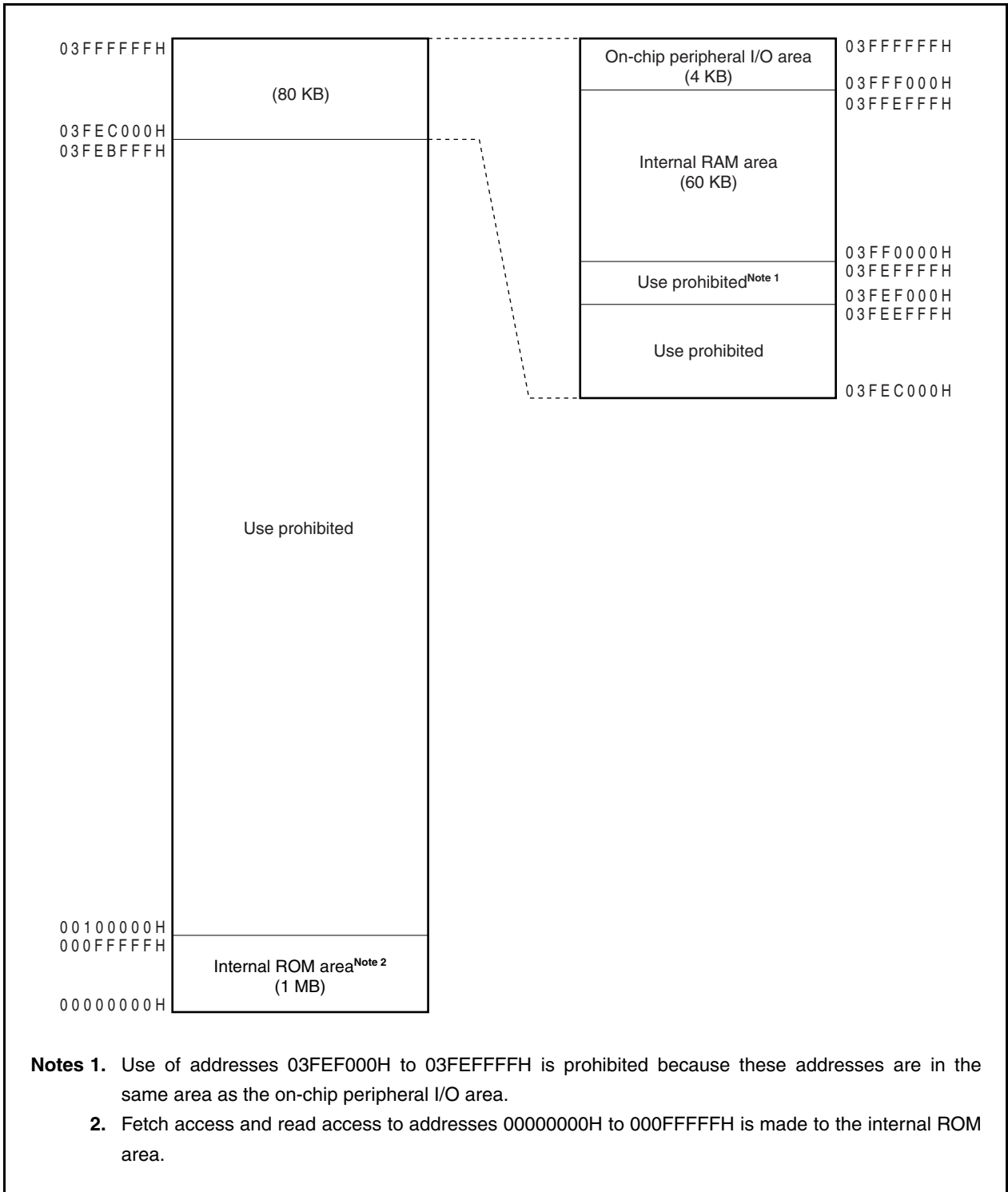
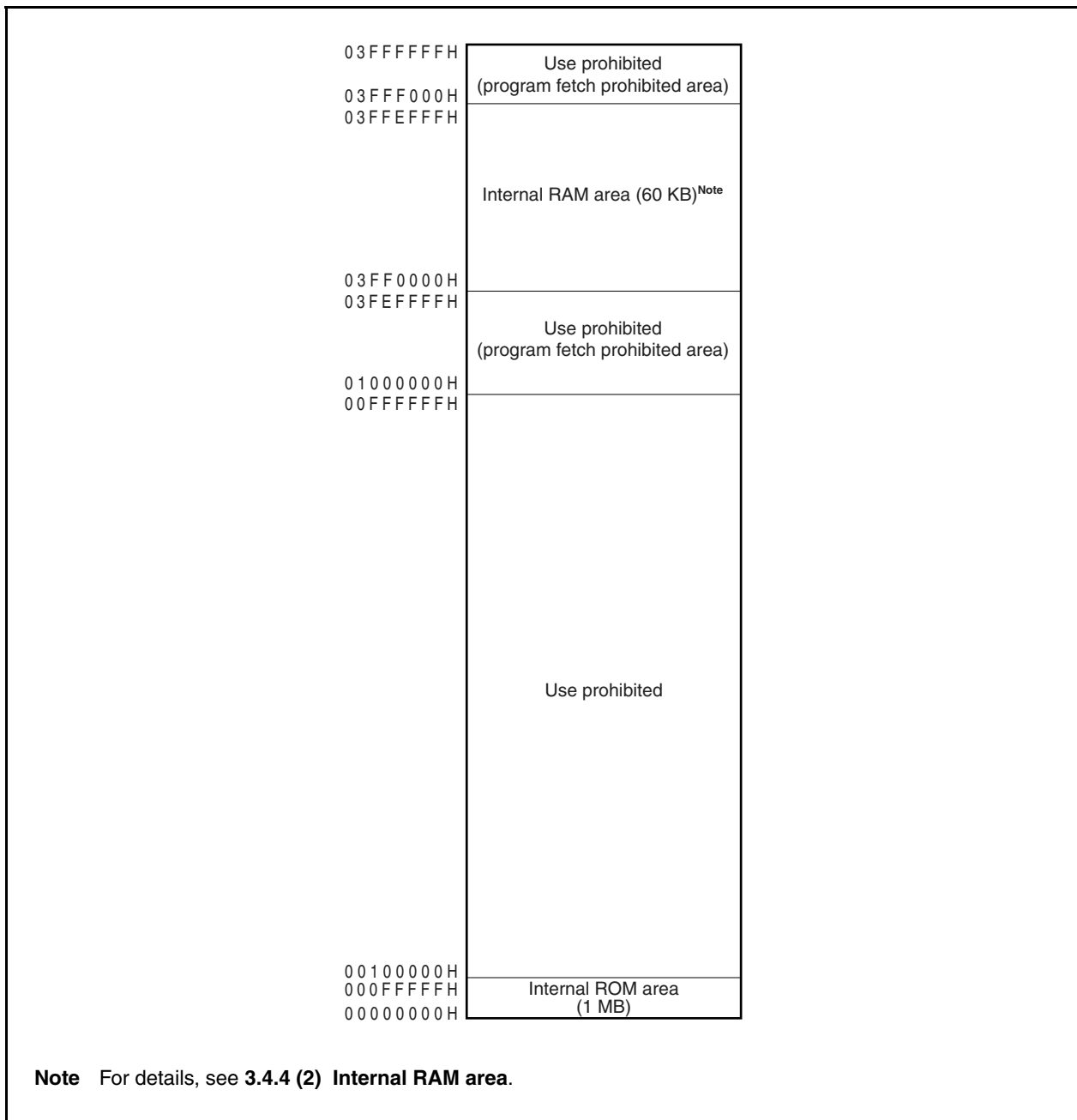


Figure 3-3. Program Memory Map



3.4.4 Areas

(1) Internal ROM area

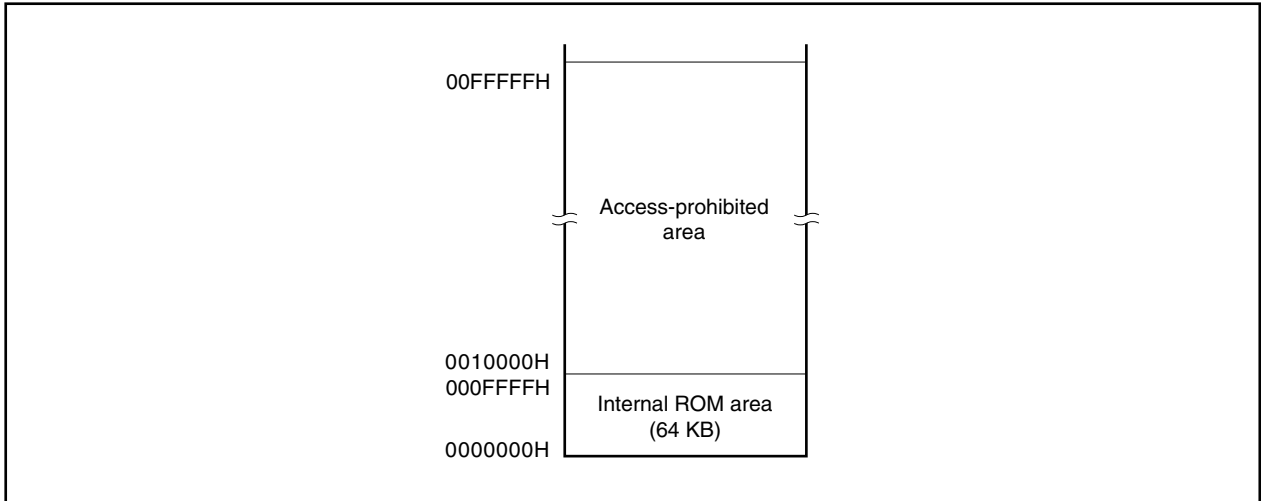
Up to 1 MB is reserved as an internal ROM area.

(a) Internal ROM (64 KB)

A 64 KB area from 0000000H to 00FFFFFFH is provided in the μ PD70F3702.

Addresses 0010000H to 00FFFFFFH are an access-prohibited area.

Figure 3-4. Internal ROM Area (64 KB)

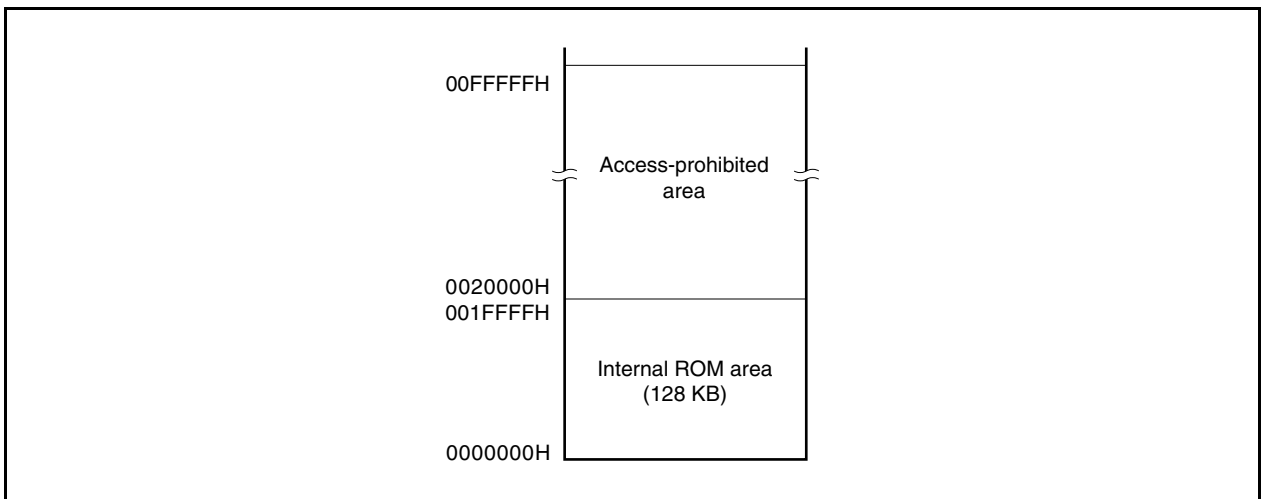


(b) Internal ROM (128 KB)

128 KB are allocated to addresses 0000000H to 001FFFFFH in the μ PD70F3703.

Accessing addresses 0020000H to 00FFFFFFH is prohibited.

Figure 3-5. Internal ROM Area (128 KB)

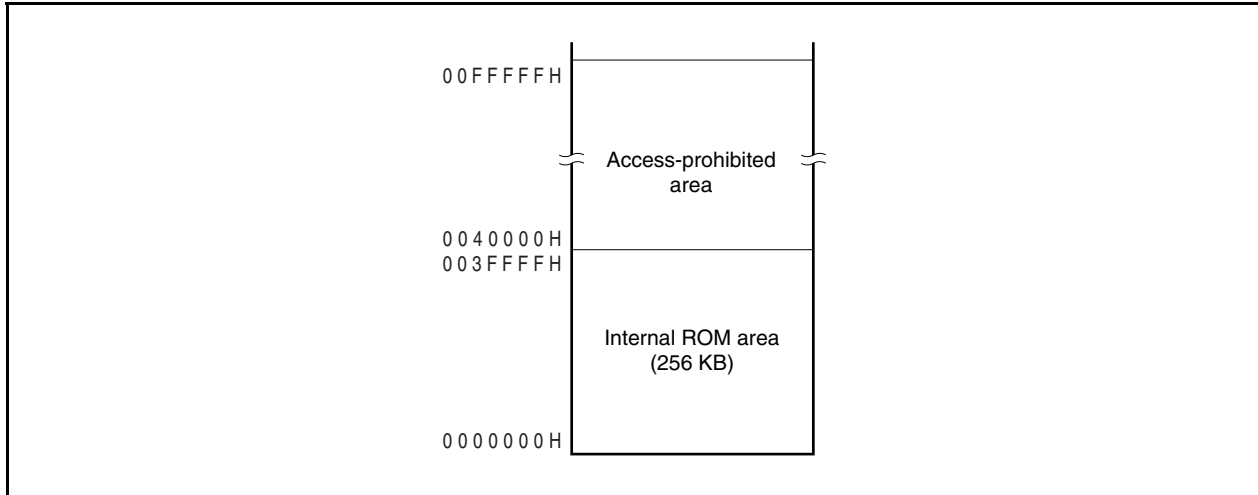


(c) Internal ROM (256 KB)

256 KB are allocated to addresses 0000000H to 003FFFFH in the μ PD70F3704.

Accessing addresses 0040000H to 00FFFFFFH is prohibited.

Figure 3-6. Internal ROM Area (256 KB)



(2) Internal RAM area

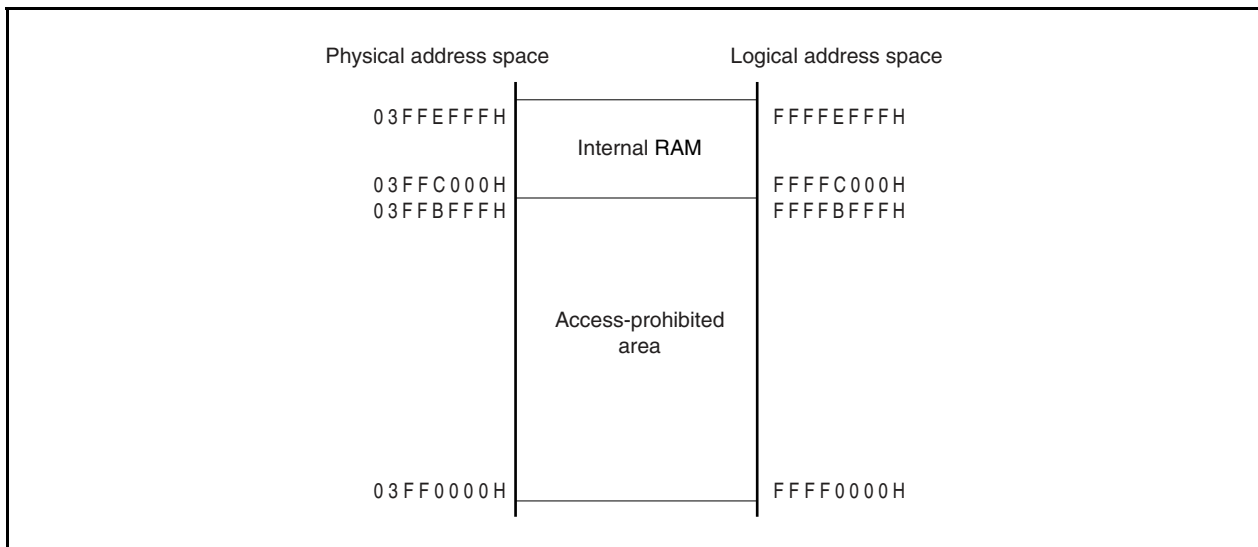
Up to 60 KB are reserved as the internal RAM area.

(a) Internal RAM (12 KB)

12 KB are allocated to addresses 03FFC000H to 03FFEFFFH in the V850ES/HF2.

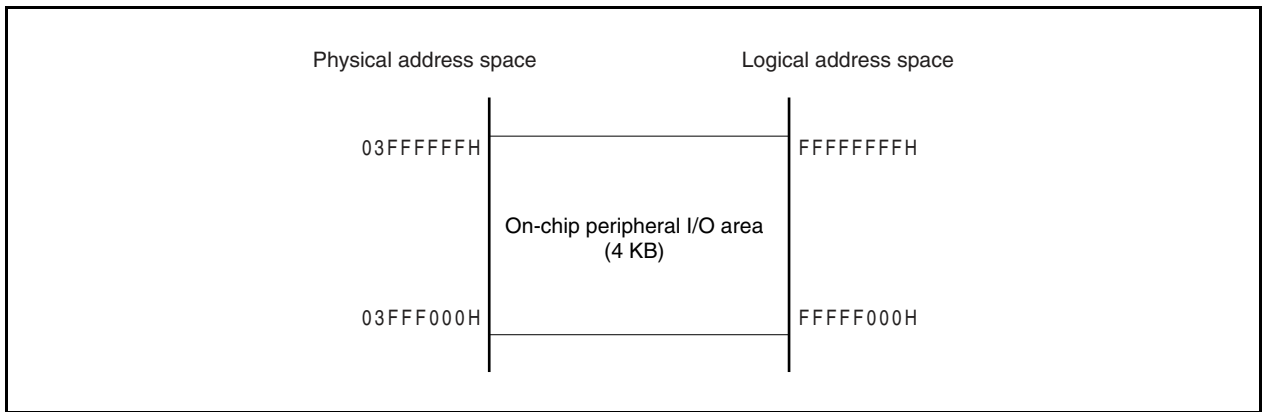
Accessing addresses 03FF0000H to 03FFBFFFH is prohibited.

Figure 3-7. Internal RAM Area (12 KB)



(3) On-chip peripheral I/O area

4 KB of addresses 03FFF000H to 03FFFFFFH are reserved as the on-chip peripheral I/O area.

Figure 3-8. 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.

3.4.5 Recommended use of address space

The architecture of the V850ES/HF2 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 03FFC000H to 03FFEFFFH.

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

(2) Data space

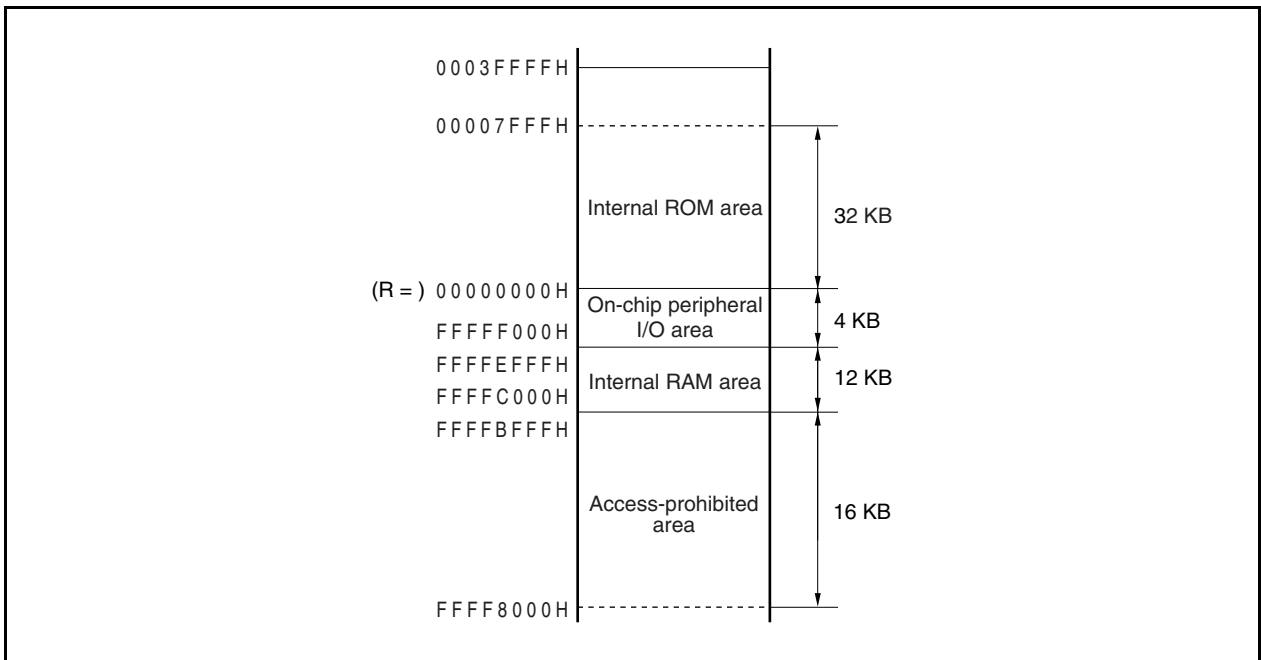
With the V850ES/HF2, it seems that there are sixty-four 64 MB address spaces on the 4 GB CPU address space. Therefore, the least significant bit (bit 25) of a 26-bit address is sign-extended to 32 bits and allocated as an address.

(a) Application example of wraparound

If R = r0 (zero register) is specified for the LD/ST disp16 [R] instruction, a range of addresses 00000000H \pm 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 3-9. Wraparound (μ PD70F3704)



3.4.6 Peripheral I/O registers

(1/7)

| Address | Function Register Name | Symbol | R/W | Manipulatable Bits | | | Default Value |
|-----------|-------------------------------|----------|-----|--------------------|---|----|---------------|
| | | | | 1 | 8 | 16 | |
| FFFFF004H | Port DL | PDL | R/W | | | √ | Undefined |
| FFFFF004H | Port DLL | PDLL | | √ | √ | | Undefined |
| FFFFF005H | Port DLH | PDLH | | √ | √ | | Undefined |
| FFFFF008H | Port CS | PCS | | √ | √ | | Undefined |
| FFFFF00AH | Port CT | PCT | | √ | √ | | Undefined |
| FFFFF00CH | Port CM | PCM | | √ | √ | | Undefined |
| FFFFF024H | Port mode register DL | PMDL | | | | √ | FFFFH |
| FFFFF024H | Port mode register DLL | PMDLL | | √ | √ | | FFH |
| FFFFF025H | Port mode register DLH | PMDLH | | √ | √ | | FFH |
| FFFFF028H | Port mode register CS | PMCS | | √ | √ | | FFH |
| FFFFF02AH | Port mode register CT | PMCT | | √ | √ | | FFH |
| FFFFF02CH | Port mode register CM | PMCM | | √ | √ | | FFH |
| FFFFF04CH | Port mode control register CM | PMCCM | | √ | √ | | 00H |
| FFFFF06EH | System wait control register | VSWC | | | √ | | 77H |
| 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 |
| 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 |

| Address | Function Register Name | Symbol | R/W | Manipulatable Bits | | | Default Value | |
|------------|--|----------|-----|--------------------|---|-----|---------------|-----------|
| | | | | 1 | 8 | 16 | | |
| FFFFFF12EH | Interrupt control register | TP0CCIC0 | R/W | √ | √ | | 47H | |
| FFFFFF130H | Interrupt control register | TP0CCIC1 | | √ | √ | | 47H | |
| FFFFFF132H | Interrupt control register | TP1OVIC | | √ | √ | | 47H | |
| FFFFFF134H | Interrupt control register | TP1CCIC0 | | √ | √ | | 47H | |
| FFFFFF136H | Interrupt control register | TP1CCIC1 | | √ | √ | | 47H | |
| FFFFFF138H | Interrupt control register | TP2OVIC | | √ | √ | | 47H | |
| FFFFFF13AH | Interrupt control register | TP2CCIC0 | | √ | √ | | 47H | |
| FFFFFF13CH | Interrupt control register | TP2CCIC1 | | √ | √ | | 47H | |
| FFFFFF13EH | Interrupt control register | TP3OVIC | | √ | √ | | 47H | |
| FFFFFF140H | Interrupt control register | TP3CCIC0 | | √ | √ | | 47H | |
| FFFFFF142H | Interrupt control register | TP3CCIC1 | | √ | √ | | 47H | |
| FFFFFF144H | Interrupt control register | TM0EQIC0 | | √ | √ | | 47H | |
| FFFFFF146H | Interrupt control register | CB0RIC | | √ | √ | | 47H | |
| FFFFFF148H | Interrupt control register | CB0TIC | | √ | √ | | 47H | |
| FFFFFF14AH | Interrupt control register | CB1RIC | | √ | √ | | 47H | |
| FFFFFF14CH | Interrupt control register | CB1TIC | | √ | √ | | 47H | |
| FFFFFF14EH | Interrupt control register | UA0RIC | | √ | √ | | 47H | |
| FFFFFF150H | Interrupt control register | UA0TIC | | √ | √ | | 47H | |
| FFFFFF152H | Interrupt control register | UA1RIC | | √ | √ | | 47H | |
| FFFFFF154H | Interrupt control register | UA1TIC | | √ | √ | | 47H | |
| FFFFFF156H | Interrupt control register | ADIC | √ | √ | | 47H | | |
| FFFFFF160H | Interrupt control register | KRIC | √ | √ | | 47H | | |
| FFFFFF162H | Interrupt control register | WTIIC | √ | √ | | 47H | | |
| FFFFFF164H | Interrupt control register | WTIC | √ | √ | | 47H | | |
| FFFFFF1FAH | In-service priority register | ISPR | R | √ | √ | | 00H | |
| FFFFFF1FCH | Command register | PRCMD | W | | √ | | Undefined | |
| FFFFFF1FEH | Power save control register | PSC | R/W | √ | √ | | 00H | |
| FFFFFF200H | A/D converter mode register 0 | ADA0M0 | | √ | √ | | 00H | |
| FFFFFF201H | A/D converter mode register 1 | ADA0M1 | | √ | √ | | 00H | |
| FFFFFF202H | A/D converter channel specification register 0 | ADA0S | | √ | √ | | 00H | |
| FFFFFF203H | A/D converter mode register 2 | ADA0M2 | | √ | √ | | 00H | |
| FFFFFF204H | Power-fail compare mode register | ADA0PFM | | √ | √ | | 00H | |
| FFFFFF205H | Power-fail compare threshold value register | ADA0PFT | | √ | √ | | 00H | |
| FFFFFF210H | A/D conversion result register 0 | ADA0CR0 | | R | | | √ | Undefined |
| FFFFFF211H | A/D conversion result register 0H | ADA0CR0H | | | | | √ | |
| FFFFFF212H | A/D conversion result register 1 | ADA0CR1 | | | | | √ | Undefined |
| FFFFFF213H | A/D conversion result register 1H | ADA0CR1H | | | | √ | | Undefined |
| FFFFFF214H | A/D conversion result register 2 | ADA0CR2 | | | | | √ | Undefined |
| FFFFFF215H | A/D conversion result register 2H | ADA0CR2H | | | | √ | | Undefined |

| Address | Function Register Name | Symbol | R/W | Manipulatable Bits | | | Default Value |
|-----------|---------------------------------------|-----------|-----|--------------------|---|----|---------------|
| | | | | 1 | 8 | 16 | |
| FFFFF216H | A/D conversion result register 3 | ADA0CR3 | R | | | √ | Undefined |
| FFFFF217H | A/D conversion result register 3H | ADA0CR3H | | | √ | | |
| FFFFF218H | A/D conversion result register 4 | ADA0CR4 | R/W | | | √ | Undefined |
| FFFFF219H | A/D conversion result register 4H | ADA0CR4H | | | √ | | |
| FFFFF21AH | A/D conversion result register 5 | ADA0CR5 | R/W | | | √ | Undefined |
| FFFFF21BH | A/D conversion result register 5H | ADA0CR5H | | | √ | | |
| FFFFF21CH | A/D conversion result register 6 | ADA0CR6 | R/W | | | √ | Undefined |
| FFFFF21DH | A/D conversion result register 6H | ADA0CR6H | | | √ | | |
| FFFFF21EH | A/D conversion result register 7 | ADA0CR7 | R/W | | | √ | Undefined |
| FFFFF21FH | A/D conversion result register 7H | ADA0CR7H | | | √ | | |
| FFFFF220H | A/D conversion result register 8 | ADA0CR8 | R/W | | | √ | Undefined |
| FFFFF221H | A/D conversion result register 8H | ADA0CR8H | | | √ | | |
| FFFFF222H | A/D conversion result register 9 | ADA0CR9 | R/W | | | √ | Undefined |
| FFFFF223H | A/D conversion result register 9H | ADA0CR9H | | | √ | | |
| FFFFF224H | A/D conversion result register 10 | ADA0CR10 | R/W | | | √ | Undefined |
| FFFFF225H | A/D conversion result register 10H | ADA0CR10H | | | √ | | |
| FFFFF226H | A/D conversion result register 11 | ADA0CR11 | R/W | | | √ | Undefined |
| FFFFF227H | A/D conversion result register 11H | ADA0CR11H | | | √ | | |
| FFFFF300H | Key return mode register | KRM | R/W | √ | √ | | 00H |
| FFFFF308H | Selector operation control register 0 | SELCNT0 | R/W | √ | √ | | 00H |
| FFFFF318H | Noise elimination control register | NFC | R/W | √ | √ | | 00H |
| FFFFF400H | Port 0 | P0 | R/W | √ | √ | | Undefined |
| FFFFF406H | Port 3 | P3 | R/W | | | √ | Undefined |
| FFFFF406H | Port 3L | P3L | R/W | √ | √ | | Undefined |
| FFFFF407H | Port 3H | P3H | R/W | √ | √ | | Undefined |
| FFFFF408H | Port 4 | P4 | R/W | √ | √ | | Undefined |
| FFFFF40AH | Port 5 | P5 | R/W | √ | √ | | Undefined |
| FFFFF40EH | Port 7L | P7L | R/W | √ | √ | | Undefined |
| FFFFF40FH | Port 7H | P7H | R/W | √ | √ | | Undefined |
| FFFFF412H | Port 9 | P9 | R/W | | | √ | Undefined |
| FFFFF412H | Port 9L | P9L | R/W | √ | √ | | Undefined |
| FFFFF413H | Port 9H | P9H | R/W | √ | √ | | Undefined |
| FFFFF420H | Port mode register 0 | PM0 | R/W | √ | √ | | FFH |
| FFFFF426H | Port mode register 3 | PM3 | R/W | | | √ | FFFFH |
| FFFFF426H | Port mode register 3L | PM3L | R/W | √ | √ | | FFH |
| FFFFF427H | Port mode register 3H | PM3H | R/W | √ | √ | | FFH |
| FFFFF428H | Port mode register 4 | PM4 | R/W | √ | √ | | FFH |
| FFFFF42AH | Port mode register 5 | PM5 | R/W | √ | √ | | FFH |
| FFFFF42EH | Port mode register 7L | PM7L | R/W | √ | √ | | FFH |
| FFFFF42FH | Port mode register 7H | PM7H | R/W | √ | √ | | FFH |

| Address | Function Register Name | Symbol | R/W | Manipulatable Bits | | | Default Value |
|-----------|-----------------------------------|---------|-----|--------------------|---|----|---------------|
| | | | | 1 | 8 | 16 | |
| FFFFF432H | Port mode register 9 | PM9 | R/W | | | √ | FFFFH |
| FFFFF432H | Port mode register 9L | PM9L | | √ | √ | | FFH |
| FFFFF433H | Port mode register 9H | PM9H | | √ | √ | | FFH |
| FFFFF440H | Port mode control register 0 | PMC0 | | √ | √ | | 00H |
| FFFFF446H | Port mode control register 3L | PMC3L | | √ | √ | | 00H |
| FFFFF448H | Port mode control register 4 | PMC4 | | √ | √ | | 00H |
| FFFFF44AH | Port mode control register 5 | PMC5 | | √ | √ | | 00H |
| FFFFF452H | Port mode control register 9 | PMC9 | | | | √ | 0000H |
| FFFFF452H | Port mode control register 9L | PMC9L | | √ | √ | | 00H |
| FFFFF453H | Port mode control register 9H | PMC9H | | √ | √ | | 00H |
| FFFFF460H | Port function control register 0 | PFC0 | | √ | √ | | 00H |
| FFFFF466H | Port function control register 3L | PFC3L | | √ | √ | | 00H |
| FFFFF46AH | Port function control register 5 | PFC5 | | √ | √ | | 00H |
| FFFFF472H | Port function control register 9 | PFC9 | | | | √ | 0000H |
| FFFFF472H | Port function control register 9L | PFC9L | | √ | √ | | 00H |
| FFFFF473H | Port function control register 9H | PFC9H | | √ | √ | | 00H |
| 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 |

| 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 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 function control expansion register 3L | PFCE3L | | √ | √ | | 00H | |
| FFFFF70AH | Port function control expansion register 5 | PFCE5 | | √ | √ | | 00H | |
| FFFFF712H | Port function control expansion register 9 | PFCE9 | | | | √ | 0000H | |
| FFFFF712H | Port function control expansion register 9L | PFCE9L | | √ | √ | | 00H | |
| FFFFF713H | Port function control expansion register 9H | PFCE9H | | √ | √ | | 00H | |
| FFFFF802H | System status register | SYS | | √ | √ | | 00H | |
| FFFFF80CH | Internal oscillation mode register | RCM | | √ | √ | | 00H | |
| FFFFF820H | Power save mode register | PSMR | | √ | √ | | 00H | |
| FFFFF824H | Lock register | LOCKR | | R | √ | √ | | 00H |

| Address | Function Register Name | Symbol | R/W | Manipulatable Bits | | | Default Value |
|-----------|--|---------|-----|--------------------|---|----|---------------|
| | | | | 1 | 8 | 16 | |
| FFFFF828H | Processor clock control register | PCC | R/W | √ | √ | | 03H |
| FFFFF82CH | PLL control register | PLLCTL | | √ | √ | | 01H |
| FFFFF82EH | CPU operating clock status register | CCLS | R | √ | √ | | 00H |
| FFFFF82FH | Programmable clock mode register | PCLM | R/W | √ | √ | | 00H |
| FFFFF870H | Clock monitor mode register | CLM | | √ | √ | | 00H |
| FFFFF888H | Reset source flag register | RESF | | √ | √ | | 00H |
| FFFFF890H | Low-voltage detection register | 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 mode register | OCDM | | √ | √ | | 01H |
| FFFFF9FEH | Peripheral emulation register 1 | PEMU1 | | √ | √ | | 00H |
| 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 |
| FFFFFB00H | TIP00 pin noise elimination control register | P00NFC | | √ | √ | | 00H |
| FFFFFB04H | TIP01 pin noise elimination control register | P01NFC | | √ | √ | | 00H |
| FFFFFB08H | TIP10 pin noise elimination control register | P10NFC | | √ | √ | | 00H |
| FFFFFB0CH | TIP11 pin noise elimination control register | P11NFC | | √ | √ | | 00H |
| FFFFFB10H | TIP20 pin noise elimination control register | P20NFC | | √ | √ | | 00H |
| FFFFFB14H | TIP21 pin noise elimination control register | P21NFC | | √ | √ | | 00H |
| FFFFFB18H | TIP30 pin noise elimination control register | P30NFC | | √ | √ | | 00H |
| FFFFFB1CH | TIP31 pin noise elimination control register | P31NFC | | √ | √ | | 00H |
| FFFFFB50H | TIQ00 pin noise elimination control register | Q00NFC | | √ | √ | | 00H |
| FFFFFB54H | TIQ01 pin noise elimination control register | Q01NFC | | √ | √ | | 00H |
| FFFFFB58H | TIQ02 pin noise elimination control register | Q02NFC | | √ | √ | | 00H |
| FFFFFB5CH | TIQ03 pin noise elimination control register | Q03NFC | | √ | √ | | 00H |

Caution For details of the OCDM register, see CHAPTER 24 ON-CHIP DEBUG FUNCTION.

| Address | Function Register Name | Symbol | R/W | Manipulatable Bits | | | Default Value |
|-----------|---|---------|-----|--------------------|---|-------|---------------|
| | | | | 1 | 8 | 16 | |
| FFFFFC00H | External interrupt falling edge specification register 0 | INTF0 | R/W | √ | √ | | 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 |
| 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 |
| 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 |
| FFFFFD16H | CSIB1 transmit data register | CB1TX | R/W | | √ | 0000H | |
| FFFFFD16H | CSIB1 transmit data register L | CB1TXL | | | √ | | 00H |

3.4.7 Special registers

Special registers are registers that are protected from being written with illegal data due to an inadvertent program loop. The V850ES/HF2 has the following seven special registers.

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

In addition, the PRCDM register is provided to protect against a write access to the special registers so that the application system does not inadvertently stop due to an inadvertent program loop. A write access to the special registers is made in a specific sequence, and an illegal store operation is reported to the SYS register (reported even when the read operation of the option data (address: 007AH) is illegal because of noise, instantaneous voltage drop, etc.).

(1) Setting data to special registers

Setting data to special registers is done in the following sequence.

- <1> Prepare the data to be set to the special register in a general-purpose register.
- <2> Write the data prepared in step <1> to the PRCMD register.
- <3> Write the setting data to the special register (using following instructions).
 - Store instruction (ST/SST instruction)
 - Bit manipulation instruction (SET1/CLR1/NOT1 instruction)
- <4> to <8> Insert NOP instructions (5 instructions)^{Note}.

[Description Example] When using PSC register (standby mode setting)

```

    ST.B r11,PSMR[r0]    ; PSMR register setting (IDLE, STOP mode setting)
<1> MOV 0x02,r10
<2> ST.B r10,PRCMD[r0]  ; PRCMD register write
<3> ST.B r10,PSC[r0]    ; PSC register setting
<4> NOPNote             ; Dummy instruction
<5> NOPNote             ; Dummy instruction
<6> NOPNote             ; Dummy instruction
<7> NOPNote             ; Dummy instruction
<8> NOPNote             ; Dummy instruction
(next instruction)

```

No special sequence is required to read special registers.

Note When switching to the IDLE1, IDLE2, STOP, or sub-IDLE mode (PSC.STP bit = 1), five NOP instructions must be inserted immediately after switching is performed.

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 <2> and <3> above are performed by successive store instructions. If another instruction is placed between <2> and <3>, and if an interrupt is acknowledged by that instruction, the above sequence may not be established, causing malfunction.

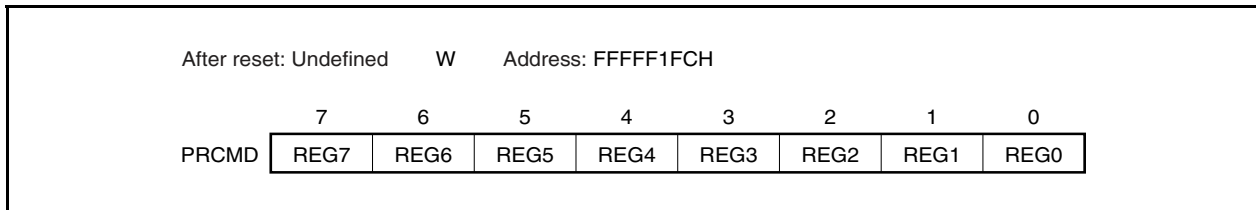
2. Although dummy data is written to the PRCMD register, use the same general-purpose register used to set the special register (<3> in Example) to write data to the PRCMD register (<2> in Example). The same applies when a general-purpose register is used for addressing.

(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 an inadvertent program loop. 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 makes this register undefined.



(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 sets 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 <3> is executed without executing <2> in **3.4.7 (1) Setting data to special registers**)
- (ii) When data is written to an on-chip peripheral I/O register other than a special register (including execution of a bit manipulation instruction) after writing data to the PRCMD register (if <3> in **3.4.7 (1) Setting data to special registers** is not the setting of a special register)

Remark Between an operation to write the PRCMD register and an operation to write a special register, even if the internal RAM is accessed, such as reading when an on-chip peripheral I/O register (except reading by a bit manipulation instruction), 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 Cautions

(1) Registers to be set first

Be sure to set the following registers first when using the V850ES/HF2.

- System wait control register (VSWC)
- On-chip debug mode register (OCDM)
- Watchdog timer mode register 2 (WDTM2)

After setting the VSWC, OCDM, and WDTM2 registers, set the other registers as necessary.

When using the external bus, set each pin to the alternate-function bus control pin mode 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 V850ES/HF2 requires wait cycles according to the operating frequency. Set the following value to the VSWC register in accordance with the frequency used.

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 |
|---|-------------------|-----------------|
| $32 \text{ kHz} \leq f_{CLK} < 16.6 \text{ MHz}$ | 00H | 0 (no waits) |
| $16.6 \text{ MHz} \leq f_{CLK} \leq 20 \text{ MHz}$ | 01H | 1 |

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

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

(c) Watchdog timer mode register 2 (WDTM2)

The WDTM2 register sets the overflow time and the operation clock of watchdog timer 2.

Watchdog timer 2 automatically starts in the reset mode after reset is released. Write the WDTM2 register to activate this operation.

For details, see **CHAPTER 10 FUNCTIONS OF WATCHDOG TIMER 2**.

(2) Accessing specific on-chip peripheral I/O registers

This product has two types of internal system buses.

One is a CPU bus and the other is a peripheral bus that interfaces with low-speed peripheral hardware.

The clock of the CPU bus and the clock of the peripheral bus are asynchronous. If an access to the CPU and an access to the peripheral hardware conflict, therefore, unexpected illegal data may be transferred. If there is a possibility of a conflict, the number of cycles for accessing the CPU changes when the peripheral hardware is accessed, so that correct data is transferred. As a result, the CPU does not start processing of the next instruction but enters the wait state. If this wait state occurs, the number of clocks required to execute an instruction increases by the number of wait clocks shown below.

This must be taken into consideration if real-time processing is required.

When specific on-chip peripheral I/O registers are accessed, more wait states may be required in addition to the wait states set by the VSWC register.

The access conditions and how to calculate the number of wait states to be inserted (number of CPU clocks) at this time are shown below.

| Peripheral Function | Register Name | Access | k |
|--|-----------------------|--------------------------------|---|
| 16-bit timer/event counter P (TMP) (n = 0 to 3) | TPnCNT | Read | 1 or 2 |
| | TPnCCR0, TPnCCR1 | Write | <ul style="list-style-type: none"> • 1st access: No wait • Continuous write: 3 or 4 |
| | | Read | 1 or 2 |
| 16-bit timer/event counter Q (TMQ) | TQ0CNT | Read | 1 or 2 |
| | TQ0CCR0 to TQ0CCR3 | Write | <ul style="list-style-type: none"> • 1st access: No wait • Continuous write: 3 or 4 |
| | | Read | 1 or 2 |
| Watchdog timer 2 (WDT2) | WDTM2 | Write (when WDT2 operating) | 3 |
| A/D converter | ADA0M0 | Read | 1 or 2 |
| | ADA0CR0 to ADA0CR11 | Read | 1 or 2 |
| | ADA0CR0H to ADA0CR11H | Read | 1 or 2 |

Number of clocks necessary for access = $3 + i + j + (2 + j) \times k$

Caution Accessing the above registers is prohibited in the following statuses. If a wait cycle is generated, it can only be cleared by a reset.

- When the CPU operates with the subclock and the main clock oscillation is stopped
- When the CPU operates with the internal oscillation clock

Remark i: Values (0 or 1) of higher 4 bits of VSWC register

j: Values (0 or 1) of lower 4 bits of VSWC register

(3) Restriction on conflict between sld instruction and interrupt request**(a) Description**

If a conflict occurs between the decode operation of an instruction in <2> immediately before the sld instruction following an instruction in <1> and an interrupt request before the instruction in <1> is complete, the execution result of the instruction in <1> may not be stored in a register.

Instruction <1>

- ld instruction: ld.b, ld.h, ld.w, ld.bu, ld.hu
- sld instruction: sld.b, sld.h, sld.w, sld.bu, sld.hu
- Multiplication instruction: mul, mulh, mulhi, mulu

Instruction <2>

| | | | |
|-------------------|-------------------|--------------------|-------------------|
| mov reg1, reg2 | not reg1, reg2 | satsubr reg1, reg2 | satsub reg1, reg2 |
| satadd reg1, reg2 | satadd imm5, reg2 | or reg1, reg2 | xor reg1, reg2 |
| and reg1, reg2 | tst reg1, reg2 | subr reg1, reg2 | sub reg1, reg2 |
| add reg1, reg2 | add imm5, reg2 | cmp reg1, reg2 | cmp imm5, reg2 |
| mulh reg1, reg2 | shr imm5, reg2 | sar imm5, reg2 | shl imm5, reg2 |

<Example>

<i> ld.w [r11], r10 If the decode operation of the mov instruction <ii> immediately before the sld instruction <iii> and an interrupt request conflict before execution of the ld instruction <i> is complete, the execution result of instruction <i> may not be stored in a register.

<ii> mov r10, r28

<iii> sld.w 0x28, r10

(b) Countermeasure

<1> When compiler (CA850) is used

Use CA850 Ver. 2.61 or later because generation of the corresponding instruction sequence can be automatically suppressed.

<2> Countermeasure by assembler

When executing the sld instruction immediately after instruction <ii>, avoid the above operation using either of the following methods.

- Insert a nop instruction immediately before the sld instruction.
- Do not use the same register as the sld instruction destination register in the above instruction <ii> executed immediately before the sld instruction.

CHAPTER 4 PORT FUNCTIONS

4.1 Features

- I/O ports: 67
- Port pins function alternately as other peripheral-function I/O pins
- Can be set in input or output mode in 1-bit units.

4.2 Basic Configuration of Ports

The V850ES/HF2 has a total of 67 input/output ports, ports 0, 3 to 5, 7, 9, CM, CS, CT, and DL. The port configuration is shown below.

Figure 4-1. Port Configuration

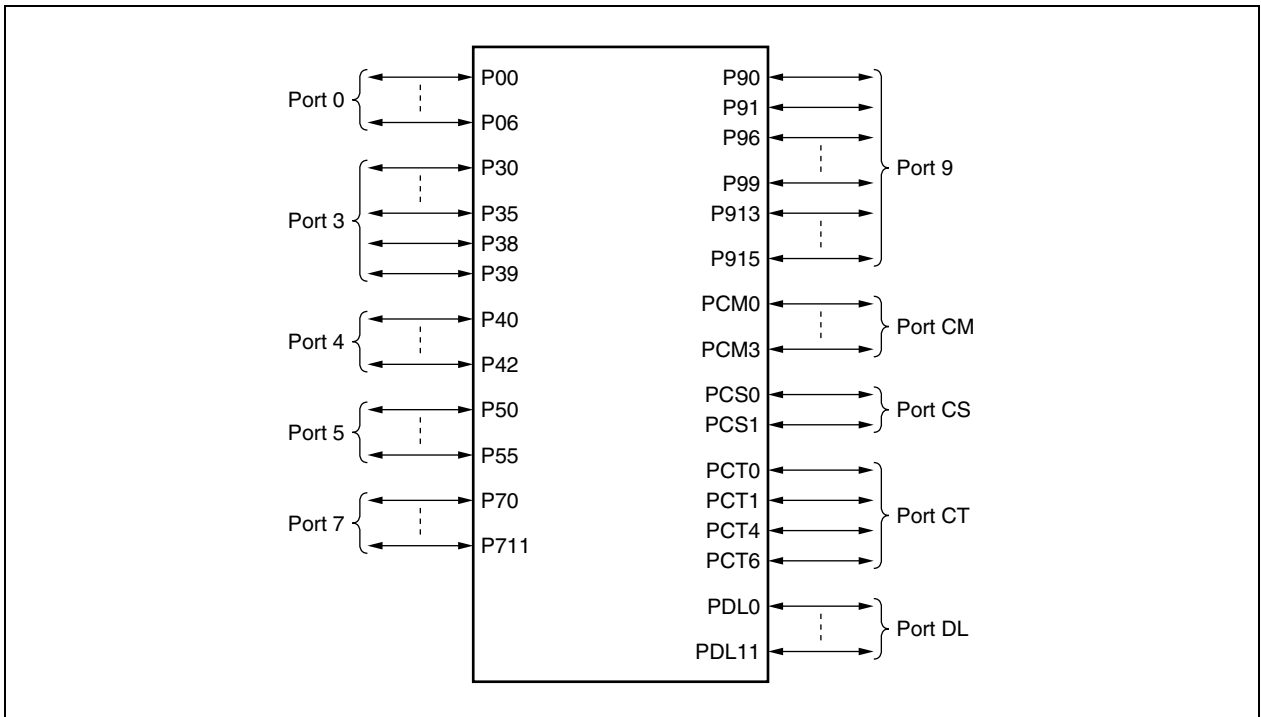


Table 4-1. Configuration of Ports

| Item | Configuration |
|-------------------|---|
| Control registers | Port mode register (PMn: n = 0, 3 to 5, 7L, 7H, 9, CM, CS, CT, or DL) |
| | Port mode control register (PMCn: n = 0, 3L, 4, 5, 9, or CM) |
| | Port function control register (PFCn: n = 0, 3L, 5, or 9) |
| | Port function control expansion register (PFCEn: n = 3L, 5, or 9) |
| | Pull-up resistor option register (PUn: n = 0, 3 to 5, or 9) |
| Ports | 67 |

Table 4-2. Pin I/O Buffer Power Supplies

| Power Supply | Corresponding Pin |
|--------------------|--|
| AV _{REF0} | Port 7 |
| EV _{DD} | Ports 0, 3 to 5, 9, CM, CS, CT, DL, <u>RESET</u> |

4.3 Port Functions

4.3.1 Operation of port function

The operation of a port differs depending on setting of the input or output mode, as follows.

(1) Writing to I/O port

(a) In output mode

A value can be written to the output latch by using a transfer instruction. The contents of the output latch are output from the pin. Once data has been written to the output latch, it is retained until new data is written to the output latch.

(b) In input mode

A value can be written to the output latch by using a transfer instruction. Because the output buffer is off, however, the status of the pin remains unchanged.

Once data has been written to the output latch, it is retained until new data is written to the output latch.

Caution Although a 1-bit memory manipulation instruction manipulates 1 bit, it accesses a port in 8-bit units. If a port has a mixture of input and output pins, therefore, the contents of the output latch of a pin set in the input mode become undefined, even if the pin is not subject to manipulation.

(2) Reading from I/O port

(a) In output mode

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

(b) In input mode

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

(3) Operation of I/O port

(a) In output mode

An operation is performed on the contents of the output latch and the result is written to the output latch. The contents of the output latch are output from the pin.

Once data has been written to the output latch, it is retained until new data is written to the output latch.

(b) In input mode

The contents of the output latch become undefined. Because the output buffer is off, however, the status of the pin remains unchanged.

Caution Although a 1-bit memory manipulation instruction manipulates 1 bit, it accesses a port in 8-bit units. If a port has a mixture of input and output pins, therefore, the contents of the output latch of a pin set in the input mode become undefined, even if the pin is not subject to manipulation.

4.3.2 Notes on setting port pins

(1) The number of ports and alternate functions differs depending on the product. Set the registers related to the unavailable ports and alternate functions to the value after reset.

(2) Set the registers of the ports using the following procedure.

<1> Set port function control register n (PFCn) and port function control expansion register n (PFCEn).

<2> Set port mode control register n (PMcN).

<3> Set external interrupt falling edge specification register n (INTFn) and external interrupt rising edge specification register n (INTRn).

If the PFCn and PFCEn registers are set after the PMcN register was set, an unexpected peripheral function pin may be set while the PFCn and PFCEn registers are being set.

(3) The PUnm bit (which connects an on-chip pull-up resistor) of the PUn register is valid only in the input mode (PMnm bit of PMn register = 1). In the output mode (PMnm bit of PMn register = 0), the on-chip pull-up register is disconnected by hardware.

(4) Reading the pin level and port latch is controlled by the port mode register (PMn). The same applies when an alternate function is used.

(5) The Schmitt (SHMT)-trigger input buffer does not operate as an SHMT buffer when it is read in the port mode.

4.3.3 Port 0

Port 0 is a 7-bit port (P00 to P06) for which I/O settings can be controlled in 1-bit units.

(1) Functions of port 0

- The input/output data of the port can be specified in 1-bit units.
Specified by port register 0 (P0)
- The input/output mode of the port can be specified in 1-bit units.
Specified by port mode register 0 (PM0)
- Port mode or control mode (alternate function) can be specified in 1-bit units.
Specified by port mode control register 0 (PMC0)
- Control mode 1 or control mode 2 can be specified in 1-bit units.
Specified by port function control register 0 (PFC0)
- An on-chip pull-up resistor can be connected in 1-bit units.
Specified by pull-up resistor option register 0 (PU0)

Port 0 functions alternately as the following pins.

Table 4-3. Alternate-Function Pins of Port 0

| Pin Name | Pin No. | Alternate-Function Pin Name | I/O | Remark | Block Type |
|----------|---------|---|-----|--------|------------|
| P00 | 3 | TP31/TOP31 | I/O | - | G-1 |
| P01 | 4 | TP30/TOP30 | | | G-1 |
| P02 | 5 | NMI ^{Note 1} | | | L-1 |
| P03 | 6 | INTP0/ADTRG | | | N-1 |
| P04 | 7 | INTP1 | | | L-1 |
| P05 | 17 | INTP2/ $\overline{\text{DRST}}$ ^{Note 2} | | | AA-1 |
| P06 | 18 | INTP3 | | | L-2 |

Notes 1. The NMI pin alternately functions as the P02 pin. It functions as the P02 pin after reset.

To enable the NMI pin, set the PMC0.PMC02 bit to 1. The initial setting of the NMI pin is “No edge detected”. Select the NMI pin valid edge using INTF0 and INTR0 registers.

2. The alternate function of the P05 pin is the on-chip debug function. After external reset, the P05/INTP2/ $\overline{\text{DRST}}$ pin is initialized as the on-chip debug pin ($\overline{\text{DRST}}$). To use the P05 pin as a port pin, not as an on-chip debug pin, the following actions must be taken.

<1> Clear the OCDM.OCDM0 bit (special register) to 0.

<2> Fix the P05/INTP2/ $\overline{\text{DRST}}$ pin to the low level until the above action has been taken.

When the on-chip debug function is not used, inputting a high level to the $\overline{\text{DRST}}$ pin before the above actions are taken may cause a malfunction (CPU deadlock). Exercise utmost care in handling the P05 pin.

When a high level is not input to the P05/INTP2/ $\overline{\text{DRST}}$ pin (when this pin is fixed to low level), it is not necessary to manipulate the OCDM.OCDM0 bit.

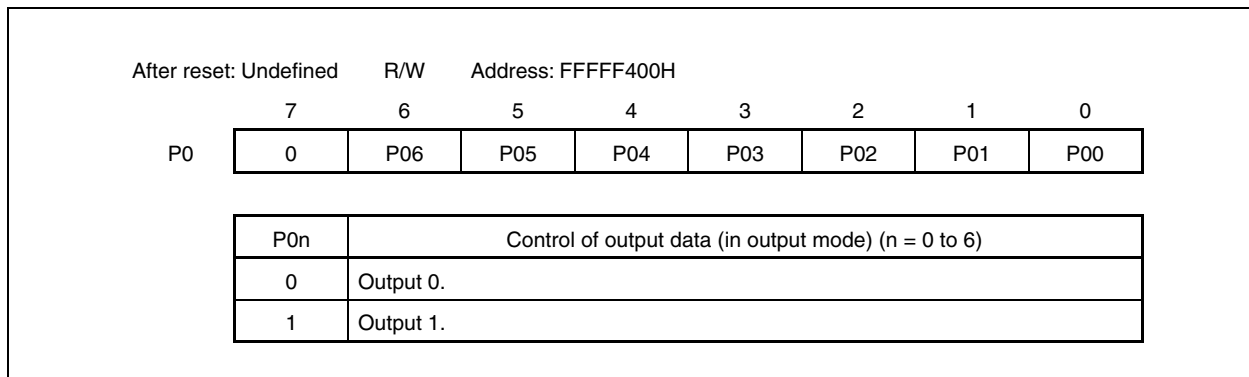
Because a pull-down resistor (30 k Ω TYP.) is connected to the buffer of the P05/INTP2/ $\overline{\text{DRST}}$ pin, the pin does not have to be fixed to the low level by an external source. The pull-down resistor is disconnected by clearing the OCDM0 bit to 0.

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

(2) Registers

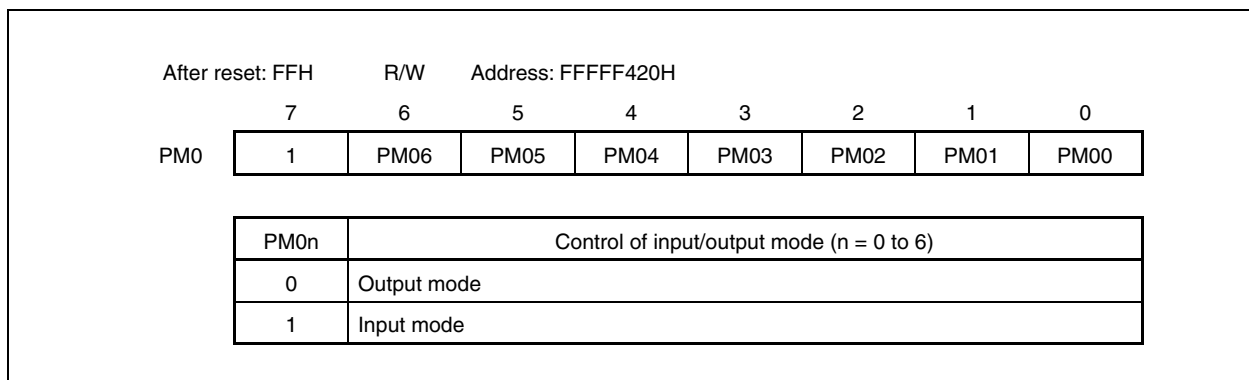
(a) Port register 0 (P0)

Port register 0 (P0) is an 8-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.



(b) Port mode register 0 (PM0)

This is an 8-bit register that specifies the input or output mode. It can be read or written in 8-bit or 1-bit units.



(c) Port mode control register 0 (PMC0)

This is an 8-bit register that specifies the port mode or control mode. It can be read or written in 8-bit or 1-bit units.

| | | | | | | | | |
|------------------|--|---------------------------------------|--------------------|-------|-------|-------|-------|-------|
| After reset: 00H | | R/W | Address: FFFFF440H | | | | | |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PMC0 | 0 | PMC06 | PMC05 | PMC04 | PMC03 | PMC02 | PMC01 | PMC00 |
| | Specification of operation mode of P06 pin | | | | | | | |
| | 0 | I/O port | | | | | | |
| | 1 | INTP3 input | | | | | | |
| | Specification of operation mode of P05 pin | | | | | | | |
| | 0 | I/O port | | | | | | |
| | 1 | INTP2/ $\overline{\text{DRST}}$ input | | | | | | |
| | Specification of operation mode of P04 pin | | | | | | | |
| | 0 | I/O port | | | | | | |
| | 1 | INTP1 input | | | | | | |
| | Specification of operation mode of P03 pin | | | | | | | |
| | 0 | I/O port | | | | | | |
| | 1 | INTP0/ADTRG input | | | | | | |
| | Specification of operation mode of P02 pin | | | | | | | |
| | 0 | I/O port | | | | | | |
| | 1 | NMI input | | | | | | |
| | Specification of operation mode of P01 pin | | | | | | | |
| | 0 | I/O port | | | | | | |
| | 1 | TIP30/TOP30 I/O | | | | | | |
| | Specification of operation mode of P00 pin | | | | | | | |
| | 0 | I/O port | | | | | | |
| | 1 | TIP31/TOP31 I/O | | | | | | |

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

(d) Port function control register 0 (PFC0)

This is an 8-bit register that specifies control mode 1 or control mode 2. It can be read or written in 8-bit or 1-bit units.

| | | | | | | | | | |
|------------------|-----|---|--------------|---|---|-------|---|-------|-------|
| 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 operation mode when P03 pin is in control mode | | | | | | | |
| | | 0 | INTP0 input | | | | | | |
| | | 1 | ADTRG input | | | | | | |
| PFC01 | | Specification of operation mode when P01 pin is in control mode | | | | | | | |
| | | 0 | TIP30 input | | | | | | |
| | | 1 | TOP30 output | | | | | | |
| PFC00 | | Specification of operation mode when P00 pin is in control mode | | | | | | | |
| | | 0 | TIP31 input | | | | | | |
| | | 1 | TOP31 output | | | | | | |

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

This is an 8-bit register that specifies connection of an on-chip pull-up resistor. It can be read or written in 8-bit or 1-bit units.

| | | | | | | | | | |
|------------------|-----|---|---------------|------|------|------|------|------|------|
| 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 | | | | | | |

4.3.4 Port 3

Port 3 is an 8-bit port (P30 to P35, P38, P39) for which I/O settings can be controlled in 1-bit units.

(1) Function of port 3

- The input/output data of the port can be specified in 1-bit units.
Specified by port register 3 (P3)
- The input/output mode of the port can be specified in 1-bit units.
Specified by port mode register 3 (PM3)
- Port mode or control mode (alternate function) can be specified in 1-bit units.
Specified by port mode control register 3L (PMC3L)
- Control mode can be specified in 1-bit units.
Specified by port function control register 3L (PFC3L) and port function control expansion register 3L (PFCE3L)
- An on-chip pull-up resistor can be connected in 1-bit units.
Specified by pull-up resistor option register 3 (PU3)

Port 3 functions alternately as the following pins.

Table 4-4. Alternate-Function Pins of Port 3

| Pin Name | Pin No. | Alternate-Function Pin Name | I/O | Remark | Block Type |
|----------|---------|-----------------------------|-----|--------|------------|
| P30 | 22 | TXDA0 | I/O | - | E-2 |
| P31 | 23 | RXDA0/INTP7 | | | L-2 |
| P32 | 24 | ASCKA0/TIP00/TOP00/TOP01 | | | U-13 |
| P33 | 25 | TIP01/TOP01 | | | G-1 |
| P34 | 26 | TIP10/TOP10 | | | G-1 |
| P35 | 27 | TIP11/TOP11 | | | G-1 |
| P38 | 28 | - | | | C-1 |
| P39 | 29 | - | | | C-1 |

Caution The P31 to P35 pins have hysteresis characteristics in the input mode of the alternate function, but do not have hysteresis characteristics in the port mode.

(2) Registers

(a) Port register 3 (P3)

Port register 3 (P3) is a 16-bit register that controls reading the pin level and writing the output level. This register can be read or written in 16-bit units.

If the higher 8 bits of the P3 register are used as the P3H register, and the lower 8 bits as the P3L register, however, these registers can be read or written in 8-bit or 1-bit units.

| | | | | | | | | | | |
|---------------------------|--|-----|-------------------------------|-----|-----|-----|-----|-----|--|--|
| After reset: Undefined | | R/W | Address: FFFFF406H, 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) | 0 | 0 | P35 | P34 | P33 | P32 | P31 | P30 | | |
| P3n | Control of output data (in output mode) (n = 0 to 5, 8, 9) | | | | | | | | | |
| 0 | Output 0. | | | | | | | | | |
| 1 | Output 1. | | | | | | | | | |

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

(b) Port mode register 3 (PM3)

This is a 16-bit register that specifies the input or output mode. It can be read or written in 16-bit units.

If the higher 8 bits of the PM3 register are used as the PM3H register, and the lower 8 bits as the PM3L register, however, these registers can be read or written in 8-bit or 1-bit units.

| | | | | | | | | | | |
|-----------------------------|--|-----|-------------------------------|------|------|------|------|------|--|--|
| After reset: FFFFH | | R/W | Address: FFFFF426H, 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) | 1 | 1 | PM35 | PM34 | PM33 | PM32 | PM31 | PM30 | | |
| PM3n | Control of I/O mode (n = 0 to 5, 8, 9) | | | | | | | | | |
| 0 | Output mode | | | | | | | | | |
| 1 | Input mode | | | | | | | | | |

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

(c) Port mode control register 3L (PMC3L)

This is an 8-bit register that specifies the port mode or control mode. It can be read or written in 8-bit or 1-bit units.

| | | | | | | | | | |
|------------------|-----|--------------------|---|-------|-------|-------|-------|-------|-------|
| After reset: 00H | R/W | Address: FFFFF446H | | | | | | | |
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PMC3L | | 0 | 0 | PMC35 | PMC34 | PMC33 | PMC32 | PMC31 | PMC30 |

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

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

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

| | |
|-------|--|
| PMC32 | Specification of operation mode of P32 pin |
| 0 | I/O port |
| 1 | ASCKA0/TIP00/TOP00/TOP01 I/O |

| | |
|-------|--|
| PMC31 | Specification of operation mode of P31 pin |
| 0 | I/O port |
| 1 | RXDA0/INTP7 input ^{Note} |

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

Note The INTP7 pin functions alternately as the RXDA0 pin. To use as the RXDA0 pin, invalidate the edge detection function of the alternate-function INTP7 pin (by fixing the INTF3.INTF31 and INTR3.INTR31 bits to 0). To use as the INTP7 pin, stop the reception operation of UARTA0 (by clearing the UA0CTL0.UA0RXE bit to 0).

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

This is an 8-bit register that specifies control mode 1, 2, 3, or 4. It can be read or written in 8-bit or 1-bit units.

| | | | | | | | | | |
|------------------|-----|--------------------|---|-------|-------|-------|-------|---|---|
| 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 how to specify a control mode, see 4.3.4 (2) (f) **Setting of control mode of P3 pin.**

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

This is an 8-bit register that specifies control mode 1, 2, 3, or 4. It can be read or written in 8-bit or 1-bit units.

| | | | | | | | | | |
|------------------|-----|--------------------|---|---|---|---|--------|---|---|
| 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 how to specify a control mode, see 4.3.4 (2) (f) **Setting of control mode of P3 pin.**

(f) Setting of control mode of P3 pin

| | | |
|-------|--|--|
| PFC35 | Specification of control mode of P35 pin | |
| 0 | TIP11 input | |
| 1 | TOP11 output | |

| | | |
|-------|--|--|
| PFC34 | Specification of control mode of P34 pin | |
| 0 | TIP10 input | |
| 1 | TOP10 output | |

| | | |
|-------|--|--|
| PFC33 | Specification of control mode of P33 pin | |
| 0 | TIP01 input | |
| 1 | TOP01 output | |

| | | |
|--------|-------|--|
| PFCE32 | PFC32 | Specification of control mode of P32 pin |
| 0 | 0 | ASCKA0 input |
| 0 | 1 | TOP01 output |
| 1 | 0 | TIP00 input |
| 1 | 1 | TOP00 output |

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

This is a 16-bit register that specifies connection of an on-chip pull-up resistor. It can be read or written in 16-bit units.

If the higher 8 bits of the PU3 register are used as the PU3H register, and the lower 8 bits as the PU3L register, however, these registers can be read or written in 8-bit or 1-bit units.

| | | |
|--------------------|-----|-----------------------------|
| After reset: 0000H | R/W | Address: FFFFC46H, FFFFC47H |
|--------------------|-----|-----------------------------|

| | | | | | | | | |
|-----------------------------|----|----|----|----|----|----|------|------|
| | 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) | 0 | 0 | PU35 | PU34 | PU33 | PU32 | PU31 | PU30 |

| PU3n | Control of on-chip pull-up resistor connection (n = 0 to 5, 8, 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 these bits as bits 0 to 7 of the PU3H register.

4.3.5 Port 4

Port 4 is a 3-bit port (P40 to P42) for which I/O settings can be controlled in 1-bit units.

(1) Functions of port 4

- The input/output data of the port can be specified in 1-bit units.
Specified by port register 4 (P4)
- The input/output mode of the port can be specified in 1-bit units.
Specified by port mode register 4 (PM4)
- Port mode or control mode (alternate function) can be specified in 1-bit units.
Specified by port mode control register 4 (PMC4)
- An on-chip pull-up resistor can be connected in 1-bit units.
Specified by pull-up resistor option register 4 (PU4)

Port 4 functions alternately as the following pins.

Table 4-5. Alternate-Function Pins of Port 4

| Pin Name | Pin No. | Alternate-Function Pin Name | I/O | Remark | Block Type |
|----------|---------|-----------------------------|-----|--------|------------|
| P40 | 19 | SIB0 | I/O | - | E-1 |
| P41 | 20 | SOB0 | | | E-2 |
| P42 | 21 | SCKB0 | | | E-3 |

Caution The P40 and P42 pins have hysteresis characteristics in the input mode of the alternate function, but do not have hysteresis characteristics in the port mode.

(2) Registers

(a) Port register 4 (P4)

Port register 4 (P4) is an 8-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.

| | | | | | | | | |
|------------------------|--|-----|--------------------|---|---|-----|-----|-----|
| After reset: Undefined | | 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 | Output 0. | | | | | | | |
| 1 | Output 1. | | | | | | | |

(b) Port mode register 4 (PM4)

This is an 8-bit register that specifies the input or output mode. It can be read or written in 8-bit or 1-bit units.

| | | | | | | | | |
|------------------|---|-----|--------------------|---|---|------|------|------|
| 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 input/output mode (n = 0 to 2) | | | | | | | |
| 0 | Output mode | | | | | | | |
| 1 | Input mode | | | | | | | |

(c) Port mode control register 4 (PMC4)

This is an 8-bit register that specifies the port mode or control mode. It can be read or written in 8-bit or 1-bit units.

| | | | | | | | | |
|------------------|---|-----|--------------------|---|---|-------|-------|-------|
| 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 operation mode of P42 pin | |
| 0 | I/O port | |
| 1 | SCKB0 I/O | |

| | | |
|-------|--|--|
| PMC41 | Specification of operation mode of P41 pin | |
| 0 | I/O port | |
| 1 | SOB0 output | |

| | | |
|-------|--|--|
| PMC40 | Specification of operation mode of P40 pin | |
| 0 | I/O port | |
| 1 | SIB0 input | |

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

This is an 8-bit register that specifies connection of an on-chip pull-up resistor. It can be read or written in 8-bit or 1-bit units.

| | | | | | | | | |
|------------------|---|-----|-------------------|---|---|------|------|------|
| After reset: 00H | | R/W | Address: FFFFC48H | | | | | |
| | 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 | |

4.3.6 Port 5

Port 5 is a 6-bit port (P50 to P55) for which I/O settings can be controlled in 1-bit units.

(1) Functions of port 5

- The input/output data of the port can be specified in 1-bit units.
Specified by port register 5 (P5)
- The input/output mode of the port can be specified in 1-bit units.
Specified by port mode register 5 (PM5)
- Port mode or control mode (alternate function) can be specified in 1-bit units.
Specified by port mode control register 5 (PMC5)
- Control mode can be specified in 1-bit units.
Specified by port function control register 5 (PFC5) or port function control expansion register 5 (PFCE5)
- An on-chip pull-up resistor can be connected in 1-bit units.
Specified by pull-up resistor option register 5 (PU5)

Port 5 functions alternately as the following pins.

Table 4-6. Alternate-Function Pins of Port 5

| Pin Name | Pin No. | Alternate-Function Pin Name | I/O | Remark | Block Type |
|----------|---------|-------------------------------------|-----|--------|------------|
| P50 | 32 | KR0/TIQ01/TOQ01 | I/O | - | U-4 |
| P51 | 33 | KR1/TIQ02/TOQ02 | | | U-4 |
| P52 | 34 | KR2/TIQ03/TOQ03/DDI ^{Note} | | | U-5 |
| P53 | 35 | KR3/TIQ00/TOQ00/DDO ^{Note} | | | U-6 |
| P54 | 36 | KR4/DCK ^{Note} | | | G-2 |
| P55 | 37 | KR5/DMS ^{Note} | | | G-2 |

Note The DDI, DDO, DCK, and DMS pins are for the on-chip debug function. To use the DDI, DDO, DCK, and DMS pins as port pins, not as on-chip debug pins, the following actions must be taken.

- <1> Clear the OCDM0 bit of the OCDM register (special register) to 0.
- <2> Fix the P05/INTP2/ $\overline{\text{DRST}}$ pin to the low level until the above action has been taken.

When the on-chip debug function is not used, inputting a high level to the $\overline{\text{DRST}}$ pin before the above actions are taken may cause a malfunction (CPU deadlock). Exercise utmost care in handling the P05 pin.

When a high level is not input to the P05/INTP2/ $\overline{\text{DRST}}$ pin (when this pin is fixed to low level), it is not necessary to manipulate the OCDM.OCDM0 bit.

Because a pull-down resistor (30 kΩ TYP.) is connected to the buffer of the P05/INTP2/ $\overline{\text{DRST}}$ pin, the pin does not have to be fixed to the low level by an external source. The pull-down resistor is disconnected by clearing the OCDM0 bit to 0.

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

(2) Registers

(a) Port register 5 (P5)

Port register 5 (P5) is an 8-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.

| | | | | | | | | |
|------------------------|--|-----|--------------------|-----|-----|-----|-----|-----|
| After reset: Undefined | | 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 | Output 0. | | | | | | | |
| 1 | Output 1. | | | | | | | |

(b) Port mode register 5 (PM5)

This is an 8-bit register that specifies the input or output mode. It can be read or written in 8-bit or 1-bit units.

| | | | | | | | | |
|------------------|----------------------------------|-----|--------------------|------|------|------|------|------|
| 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 | | | | | | | |

(c) Port mode control register 5 (PMC5)

This is an 8-bit register that specifies the port mode or control mode. It can be read or written in 8-bit or 1-bit units.

Caution If the control mode is specified by using the PMC5 register when the PFC5.PFC5n and PFCE5.PFCE5n bits are the default values (0), the output becomes undefined. For this reason, first set the PFC5.PFC5n and PFCE5.PFCE5n bits, and then set the PMC5n bit to 1 to set the control mode.

| | | | | | | | | |
|------------------|---|-----|--------------------|-------|-------|-------|-------|-------|
| 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 operation mode of P55 pin |
| 0 | I/O port |
| 1 | KR5 input |

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

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

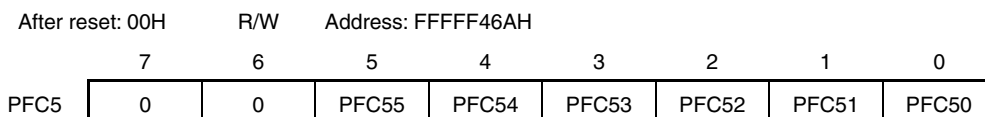
| | |
|-------|--|
| PMC52 | Specification of operation mode of P52 pin |
| 0 | I/O port |
| 1 | KR2/TIQ03/TOQ03 I/O |

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

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

(d) Port function control register 5 (PFC5)

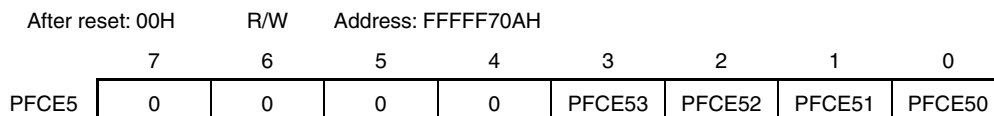
This is an 8-bit register that specifies control mode 1, 2, 3, or 4. It can be read or written in 8-bit or 1-bit units.



Remark For how to specify a control mode, see 4.3.6 (2) (f) **Setting of control mode of P5 pin.**

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

This is an 8-bit register that specifies control mode 1, 2, 3, or 4. It can be read or written in 8-bit or 1-bit units.



Remark For how to specify a control mode, see 4.3.6 (2) (f) **Setting of control mode of P5 pin.**

(f) Setting of control mode of P5 pin

Caution If the control mode is specified by using the PMC5 register when the PFC5.PFC5n and PFCE5.PFCE5n bits are the default values (0), the output becomes undefined.

For this reason, first set the PFC5.PFC5n and PFCE5.PFCE5n bits, and then set the PMC5n bit to 1 to set the control mode.

| PFC55 | Specification of control mode of P55 pin |
|-------|--|
| 0 | Setting prohibited |
| 1 | KR5 input |

| PFC54 | Specification of control mode of P54 pin |
|-------|--|
| 0 | Setting prohibited |
| 1 | KR4 input |

| PFCE53 | PFC53 | Specification of control mode of P53 pin |
|--------|-------|--|
| 0 | 0 | Setting prohibited |
| 0 | 1 | TIQ00/KR3 ^{Note} input |
| 1 | 0 | TOQ00 output |
| 1 | 1 | Setting prohibited |

| PFCE52 | PFC52 | Specification of control mode of P52 pin |
|--------|-------|--|
| 0 | 0 | Setting prohibited |
| 0 | 1 | TIQ03/KR2 ^{Note} input |
| 1 | 0 | TOQ03 output |
| 1 | 1 | Setting prohibited |

| PFCE51 | PFC51 | Specification of control mode of P51 pin |
|--------|-------|--|
| 0 | 0 | Setting prohibited |
| 0 | 1 | TIQ02/KR1 ^{Note} input |
| 1 | 0 | TOQ02 output |
| 1 | 1 | Setting prohibited |

| PFCE50 | PFC50 | Specification of control mode of P50 pin |
|--------|-------|--|
| 0 | 0 | Setting prohibited |
| 0 | 1 | TIQ01/KR0 ^{Note} input |
| 1 | 0 | TOQ01 output |
| 1 | 1 | Setting prohibited |

Note The KRn pin functions alternately as the TIQ0m pin. To use this pin as the TIQ0m pin, invalidate the key return detection function of the alternate-function KRn pin (by clearing the KRM.KRMn bit to 0). To use this pin as the KRn pin, invalidate the edge detection function of the alternate-function TIQ0m pin (n = 0 to 3, m = 0 to 3).

| Pin Name | Use as TIQ0m Pin | Use as KRn Pin |
|-----------|------------------------------|---|
| KR0/TIQ01 | KRM0 bit of KRM register = 0 | TQ0TIG2, TQ0TIG3 bits of TQ0IOC1 register = 0 |
| KR1/TIQ02 | KRM1 bit of KRM register = 0 | TQ0TIG4, TQ0TIG5 bits of TQ0IOC1 register = 0 |
| KR2/TIQ03 | KRM2 bit of KRM register = 0 | TQ0TIG6, TQ0TIG7 bits of TQ0IOC1 register = 0 |
| KR3/TIQ00 | KRM3 bit of KRM register = 0 | TQ0TIG0, TQ0TIG1 bits of TQ0IOC1 register = 0 TQ0EES0, TQ0EES1 bits of TQ0IOC2 register = 0 TQ0ETS0, TQ0ETS1 bits of TQ0IOC2 register = 0 |

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

This is an 8-bit register that specifies connection of an on-chip pull-up resistor. It can be read or written in 8-bit or 1-bit units.

| | | | | | | | | | |
|------------------|-----|----------------------|---|------|------|------|------|------|------|
| After reset: 00H | R/W | Address: FFFFFFFC4AH | | | | | | | |
| | | 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 |

4.3.7 Port 7

Port 7 is a 12-bit port (P70 to P711) for which I/O settings can be controlled in 1-bit units.

(1) Functions of port 7

- The input/output data of the port can be specified in 1-bit units.
Specified by port registers 7H, 7L (P7H, P7L)
- The input/output mode of the port can be specified in 1-bit units.
Specified by port mode registers 7H, 7L (PM7H, PM7L)

Port 7 functions alternately as the following pins.

Table 4-7. Alternate-Function Pins of Port 7

| Pin Name | Pin No. | Alternate-Function Pin Name | I/O | Remark | Block Type |
|----------|---------|-----------------------------|-----|--------|------------|
| P70 | 80 | ANI0 | I/O | - | A-1 |
| P71 | 79 | ANI1 | | | A-1 |
| P72 | 78 | ANI2 | | | A-1 |
| P73 | 77 | ANI3 | | | A-1 |
| P74 | 76 | ANI4 | | | A-1 |
| P75 | 75 | ANI5 | | | A-1 |
| P76 | 74 | ANI6 | | | A-1 |
| P77 | 73 | ANI7 | | | A-1 |
| P78 | 72 | ANI8 | | | A-1 |
| P79 | 71 | ANI9 | | | A-1 |
| P710 | 70 | ANI10 | | | A-1 |
| P711 | 69 | ANI11 | | | A-1 |

(2) Registers

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

Port registers 7H and 7L (P7H and P7L) are 8-bit registers that control reading the pin level and writing the output level. These registers can be read or written in 8-bit or 1-bit units. They cannot be accessed in 16-bit units.

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

| | | | | | | | | |
|-----|---|---|---|---|------|------|-----|-----|
| | 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 | Output 0. |
| 1 | Output 1. |

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

(b) Port mode registers 7H, 7L (PM7H, PM7L)

These are 8-bit registers that specify an input or output mode. They can be read or written in 8-bit or 1-bit units.

These registers cannot be accessed in 16-bit units.

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

| | | | | | | | | |
|------|---|---|---|---|-------|-------|------|------|
| | 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 of P7n (ANIn), set PM7n to 1.

4.3.8 Port 9

Port 9 is a 9-bit port (P90, P91, P96 to P99, P913 to P915) for which I/O settings can be controlled in 1-bit units.

(1) Functions of port 9

- The input/output data of the port can be specified in 1-bit units.
Specified by port register 9 (P9)
- The input/output mode of the port can be specified in 1-bit units.
Specified by port mode register 9 (PM9)
- Port mode or control mode (alternate function) can be specified in 1-bit units.
Specified by port mode control register 9 (PMC9)
- Control mode can be specified in 1-bit units.
Specified by port function control register 9 (PFC9) and port function control expansion register 9 (PFCE9)
- An on-chip pull-up resistor can be connected in 1-bit units.
Specified by pull-up resistor option register 9 (PU9)

Port 9 functions alternately as the following pins.

Table 4-8. Alternate-Function Pins of Port 9

| Pin Name | Pin No. | Alternate-Function Pin Name | I/O | Remark | Block Type |
|----------|---------|-----------------------------|-----|--------|------------|
| P90 | 38 | KR6/TXDA1 | I/O | - | U-12 |
| P91 | 39 | KR7/RXDA1 | | | U-7 |
| P96 | 40 | TIP21/TOP21 | | | U-9 |
| P97 | 41 | SIB1/TIP20/TOP20 | | | U-8 |
| P98 | 42 | SOB1 | | | G-3 |
| P99 | 43 | $\overline{\text{SCKB1}}$ | | | G-5 |
| P913 | 44 | INTP4/PCL | | | W-1 |
| P914 | 45 | INTP5 | | | N-2 |
| P915 | 46 | INTP6 | | | N-2 |

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

(2) Registers

(a) Port register 9 (P9)

Port register 9 (P9) is a 16-bit register that controls reading the pin level and writing the output level. This register can be read or written in 16-bit units.

If the higher 8 bits of the P9 register are used as the P9H register, and the lower 8 bits as the P9L register, however, these registers can be read or written in 8-bit or 1-bit units.

After reset: Undefined R/W Address: FFFFF412H, FFFFF413H

| | | | | | | | | |
|---------------------------|------|------|------|----|----|----|-----|-----|
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| P9 (P9H ^{Note}) | P915 | P914 | P913 | 0 | 0 | 0 | P99 | P98 |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| (P9L) | P97 | P96 | 0 | 0 | 0 | 0 | P91 | P90 |

| | |
|-----|--|
| P9n | Control of output data (in output mode) (n = 0, 1, 6 to 9, 13 to 15) |
| 0 | Output 0. |
| 1 | Output 1. |

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

(b) Port mode register 9 (PM9)

This is a 16-bit register that specifies the input or output mode. It can be read or written in 16-bit units.

If the higher 8 bits of the PM9 register are used as the PM9H register, and the lower 8 bits as the PM9L register, however, these registers can be read or written in 8-bit or 1-bit units.

After reset: FFFFH R/W Address: FFFFF432H, FFFFF433H

| | | | | | | | | |
|-----------------------------|-------|-------|-------|----|----|----|------|------|
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PM9 (PM9H ^{Note}) | PM915 | PM914 | PM913 | 1 | 1 | 1 | PM99 | PM98 |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| (PM9L) | PM97 | PM96 | 1 | 1 | 1 | 1 | PM91 | PM90 |

| | |
|------|--|
| PM9n | Control of I/O mode (n = 0, 1, 6 to 9, 13 to 15) |
| 0 | Output mode |
| 1 | Input mode |

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

(c) Port mode control register 9 (PMC9)

This is a 16-bit register that specifies the port mode or control mode. It can be read or written in 16-bit units.

If the higher 8 bits of the PMC9 register are used as the PMC9H register, and the lower 8 bits as the PMC9L register, however, these registers can be read or written in 8-bit or 1-bit units.

Caution If the control mode is specified by using the PMC9 register when the PFC9.PFC9n bit and the PFCE9.PFCE9n bit are the default values (0), the output becomes undefined. For this reason, first set the PFC9.PFC9n bit and the PFCE9.PFCE9n bit to 1, and then set the PMC9n bit to 1 to set the control mode.

(1/2)

| | | | | | | | | |
|--|--------|---|-------------------------------|----|----|----|-------|-------|
| After reset: 0000H | | R/W | Address: FFFFF452H, 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 operation mode of P915 pin | | | | | | |
| | 0 | I/O port | | | | | | |
| | 1 | INTP6 input | | | | | | |
| | PMC914 | Specification of operation mode of P914 pin | | | | | | |
| | 0 | I/O port | | | | | | |
| | 1 | INTP5 input | | | | | | |
| | PMC913 | Specification of operation mode of P913 pin | | | | | | |
| | 0 | I/O port | | | | | | |
| | 1 | INTP4/PCL I/O | | | | | | |
| Note To read or write bits 8 to 15 of the PMC9 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PMC9H register. | | | | | | | | |

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

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

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

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

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

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

(d) Port function control register 9 (PFC9)

This is a 16-bit register that specifies control mode 1, 2, 3, or 4. It can be read or written in 16-bit units. If the higher 8 bits of the PFC9 register are used as the PFC9H register, and the lower 8 bits as the PFC9L register, however, these registers can be read or written in 8-bit or 1-bit units.

| | | | | | | | | | |
|-------------------------------|-----|-------------------------------|--------|--------|----|----|----|-------|-------|
| After reset: 0000H | R/W | Address: FFFFF472H, 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 or write bits 8 to 15 of the PFC9 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PFC9H register.

Remark For how to specify a control mode, see 4.3.8 (2) (f) **Setting of control mode of P9 pin.**

(e) Port function control expansion register 9 (PFCE9)

This is a 16-bit register that specifies control mode 1, 2, 3, or 4. It can be read or written in 16-bit units. If the higher 8 bits of the PFC9 register are used as the PFC9H register, and the lower 8 bits as the PFC9L register, however, these registers can be read or written in 8-bit or 1-bit units.

| | | | | | | | | |
|---------------------------------|--------|--------|-------------------------------|----|----|----|--------|--------|
| After reset: 0000H | | R/W | Address: FFFFF712H, FFFFF713H | | | | | |
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PFCE9 (PFCE9H ^{Note}) | 0 | 0 | PFCE913 | 0 | 0 | 0 | 0 | 0 |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| (PFCE9L) | PFCE97 | PFCE96 | 0 | 0 | 0 | 0 | PFCE91 | PFCE90 |

Note To read or write bits 8 to 15 of the PFCE9 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PFCE9H register.

Remark For how to specify a control mode, see **4.3.8 (2) (f) Setting of control mode of P9 pin.**

(f) Setting of control mode of P9 pin

Caution If the control mode is specified by using the PMC9 register when the PFC9.PFC9n and PFCE9.PFCE9n bits are the default values (0), the output becomes undefined.
For this reason, first set the PFC9.PFC9n and PFCE9.PFCE9n bits, and then set the PMC9n bit to 1 to set the control mode.

| PFC915 | Specification of control mode of P915 pin |
|--------|---|
| 0 | Setting prohibited |
| 1 | INTP6 input |

| PFC914 | Specification of control mode of P914 pin |
|--------|---|
| 0 | Setting prohibited |
| 1 | INTP5 input |

| PFCE913 | PFC913 | Specification of control mode of P913 pin |
|---------|--------|---|
| 0 | 0 | Setting prohibited |
| 0 | 1 | INTP4 input |
| 1 | 0 | PCL output |
| 1 | 1 | Setting prohibited |

| PFC99 | Specification of control mode of P99 pin |
|-------|--|
| 0 | Setting prohibited |
| 1 | $\overline{\text{SCKB1}}$ I/O |

| PFC98 | Specification of control mode of P98 pin |
|-------|--|
| 0 | Setting prohibited |
| 1 | SOB1 output |

| PFCE97 | PFC97 | Specification of control mode of P97 pin |
|--------|-------|--|
| 0 | 0 | Setting prohibited |
| 0 | 1 | SIB1 input |
| 1 | 0 | TIP20 input |
| 1 | 1 | TOP20 output |

| PFCE96 | PFC96 | Specification of control mode of P96 pin |
|--------|-------|--|
| 0 | 0 | Setting prohibited |
| 0 | 1 | Setting prohibited |
| 1 | 0 | TIP21 input |
| 1 | 1 | TOP21 output |

| PFCE91 | PFC91 | Specification of control mode of P91 pin |
|--------|-------|--|
| 0 | 0 | Setting prohibited |
| 0 | 1 | KR7 input |
| 1 | 0 | KR7/RXDA1 input ^{Note} |
| 1 | 1 | Setting prohibited |

| PFCE90 | PFC90 | Specification of control mode of P90 pin |
|--------|-------|--|
| 0 | 0 | Setting prohibited |
| 0 | 1 | KR6 input |
| 1 | 0 | TXDA1 output |
| 1 | 1 | Setting prohibited |

Note The KR7 pin and RXDA1 pin are alternate-function pins.
 When using the pin as the RXDA1 pin, disable KR7 pin key return detection. (Clear the KRM7 bit of the KRM register to 0.) Also, when using the pin as the KR7 pin, it is recommended to set the PFC91 bit to 1 and clear the PFCE91 bit to 0.

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

This is a 16-bit register that specifies connection of an on-chip pull-up resistor. It can be read or written in 16-bit units.

If the higher 8 bits of the PU9 register are used as the PU9H register, and the lower 8 bits as the PU9L register, however, these registers can be read or written in 8-bit or 1-bit units.

After reset: 0000H R/W Address: FFFFC52H, FFFFC53H

| | | | | | | | | |
|-----------------------------|-------|-------|-------|----|----|----|------|------|
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PU9 (PU9H ^{Note}) | PU915 | PU914 | PU913 | 0 | 0 | 0 | PU99 | PU98 |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| (PU9L) | PU97 | PU96 | 0 | 0 | 0 | 0 | PU91 | PU90 |

| | |
|------|---|
| PU9n | Control of on-chip pull-up resistor connection (n = 0, 1, 6 to 9, 13 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 these bits as bits 0 to 7 of the PU9H register.

4.3.9 Port CM

Port CM is a 4-bit port (PCM0 to PCM3) for which I/O settings can be controlled in 1-bit units.

(1) Functions of port CM

- The input/output data of the port can be specified in 1-bit units.
Specified by port register CM (PCM)
- The input/output mode of the port can be specified in 1-bit units.
Specified by port mode register CM (PMCM)
- Port mode or control mode (alternate function) can be specified in 1-bit units.
Specified by port mode control register CM (PMCCM)

Port CM functions alternately as the following pins.

Table 4-9. Alternate-Function Pins of Port CM

| Pin Name | Pin No. | Alternate-Function Pin Name | I/O | Remark | Block Type |
|----------|---------|-----------------------------|-----|--------|------------|
| PCM0 | 49 | – | I/O | – | B-1 |
| PCM1 | 50 | CLKOUT | | | D-2 |
| PCM2 | 51 | – | | | B-1 |
| PCM3 | 52 | – | | | B-1 |

(2) Registers

(a) Port register CM (PCM)

Port register CM (PCM) is an 8-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.

| | | | | | | | | |
|------------------------|--|-----|--------------------|---|------|------|------|------|
| After reset: Undefined | | 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 | Output 0. | | | | | | | |
| 1 | Output 1. | | | | | | | |

(b) Port mode register CM (PMCM)

This is an 8-bit register that specifies the input or output mode. It can be read or written in 8-bit or 1-bit units.

| | | | | | | | | |
|------------------|----------------------------------|-----|--------------------|---|-------|-------|-------|-------|
| 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 | | | | | | | |

(c) Port mode control register CM (PMCCM)

This is an 8-bit register that specifies the port mode or control mode. It can be read or written in 8-bit or 1-bit units.

| | | | | | | | | |
|------------------|---|-----|--------------------|---|---|---|--------|---|
| After reset: 00H | | R/W | Address: FFFFF04CH | | | | | |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PMCCM | 0 | 0 | 0 | 0 | 0 | 0 | PMCCM1 | 0 |
| PMCCM1 | Specification of operation mode of PCM1 pin | | | | | | | |
| 0 | I/O port | | | | | | | |
| 1 | CLKOUT output | | | | | | | |

Caution Be sure to set bits 7 to 2 and 0 to “0”.

4.3.10 Port CS

Port CS is a 2-bit port (PCS0, PCS1) for which I/O settings can be controlled in 1-bit units.

(1) Functions of port CS

- The input/output data of the port can be specified in 1-bit units.
Specified by port register CS (PCS)
- The input/output mode of the port can be specified in 1-bit units.
Specified by port mode register CS (PMCS)

Port CS functions alternately as the following pins.

Table 4-10. Alternate-Function Pins of Port CS

| Pin Name | Pin No. | Alternate-Function Pin Name | I/O | Remark | Block Type |
|----------|---------|-----------------------------|-----|--------|------------|
| PCS0 | 47 | – | I/O | – | B-1 |
| PCS1 | 48 | – | | | B-1 |

(2) Registers

(a) Port register CS (PCS)

Port register CS (PCS) is an 8-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.

| | | | | | | | | | |
|------------------------|------------------|--|---|---|---|---|---|------|------|
| After reset: Undefined | R/W | Address: FFFFF008H | | | | | | | |
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PCS | | 0 | 0 | 0 | 0 | 0 | 0 | PCS1 | PCS0 |
| | PCS _n | Control of output data (in output mode) (n = 0, 1) | | | | | | | |
| | 0 | Output 0. | | | | | | | |
| | 1 | Output 1. | | | | | | | |

(b) Port mode register CS (PMCS)

This is an 8-bit register that specifies the input or output mode. It can be read or written in 8-bit or 1-bit units.

| | | | | | | | | | |
|------------------|-------------------|--------------------------------|---|---|---|---|---|-------|-------|
| After reset: FFH | R/W | Address: FFFFF028H | | | | | | | |
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PMCS | | 0 | 0 | 0 | 0 | 0 | 0 | PMCS1 | PMCS0 |
| | PMCS _n | Control of I/O mode (n = 0, 1) | | | | | | | |
| | 0 | Output mode | | | | | | | |
| | 1 | Input mode | | | | | | | |

4.3.11 Port CT

Port CT is a 4-bit port (PCT0, PCT1, PCT4, PCT6) for which I/O settings can be controlled in 1-bit units.

(1) Functions of port CT

- The input/output data of the port can be specified in 1-bit units.
Specified by port register CT (PCT)
- The input/output mode of the port can be specified in 1-bit units.
Specified by port mode register CT (PMCT)

Port CT functions alternately as the following pins.

Table 4-11. Alternate-Function Pins of Port CT

| Pin Name | Pin No. | Alternate-Function Pin Name | I/O | Remark | Block Type |
|----------|---------|-----------------------------|-----|--------|------------|
| PCT0 | 53 | – | I/O | – | B-1 |
| PCT1 | 54 | – | | | B-1 |
| PCT4 | 55 | – | | | B-1 |
| PCT6 | 56 | – | | | B-1 |

(2) Registers

(a) Port register CT (PCT)

Port register CT (PCT) is an 8-bit register that controls reading the pin level and writing the output level. This register can be read or written in 8-bit or 1-bit units.

| | | | | | | | | | |
|------------------------|------|--|------|---|------|---|---|------|------|
| After reset: Undefined | 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 | Output 0. | | | | | | | |
| | 1 | Output 1. | | | | | | | |

(b) Port mode register CT (PMCT)

This is an 8-bit register that specifies the input or output mode. It can be read or written in 8-bit or 1-bit units.

| | | | | | | | | | |
|------------------|-------|--------------------------------------|-------|---|-------|---|---|-------|-------|
| 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 | | | | | | | |

4.3.12 Port DL

Port DL is a 12-bit port (PDL0 to PDL11) for which I/O settings can be controlled in 1-bit units.

(1) Function of port DL

- The input/output data of the port can be specified in 1-bit units.
Specified by port register DL (PDL)
- The input/output mode of the port can be specified in 1-bit units.
Specified by port mode register DL (PMDL)

Port DL functions alternately as the following pins.

Table 4-12. Alternate-Function Pins of Port DL

| Pin Name | Pin No. | Alternate-Function Pin Name | I/O | Remark | Block Type |
|----------|---------|-----------------------------|-----|--------|------------|
| PDL0 | 57 | – | I/O | – | B-1 |
| PDL1 | 58 | – | | | B-1 |
| PDL2 | 59 | – | | | B-1 |
| PDL3 | 60 | – | | | B-1 |
| PDL4 | 61 | – | | | B-1 |
| PDL5 | 62 | FLMD1 ^{Note} | | | B-1 |
| PDL6 | 63 | – | | | B-1 |
| PDL7 | 64 | – | | | B-1 |
| PDL8 | 65 | – | | | B-1 |
| PDL9 | 66 | – | | | B-1 |
| PDL10 | 67 | – | | | B-1 |
| PDL11 | 68 | – | | | B-1 |

Note Because the FLMD1 pin is used in the flash programming mode, it does not have to be manipulated by using a port control register. For details, see **CHAPTER 22 FLASH MEMORY**.

(2) Registers

(a) Port register DL (PDL)

Port register DL (PDL) is a 16-bit register that controls reading the pin level and writing the output level. This register can be read or written in 16-bit units.

If the higher 8 bits of the PDL register are used as the PDLH register, and the lower 8 bits as the PDLL register, however, these registers can be read or written in 8-bit or 1-bit units.

After reset: Undefined R/W Address: FFFFF004H, FFFFF005H

| | | | | | | | | |
|-----------------------------|------|------|------|------|-------|-------|------|------|
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PDL (PDLH ^{Note}) | 0 | 0 | 0 | 0 | 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 11) |
| 0 | Output 0. |
| 1 | Output 1. |

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

(b) Port mode register DL (PMDL)

This is a 16-bit register that specifies the input or output mode. It can be read or written in 16-bit units.

If the higher 8 bits of the PMDL register are used as the PMDLH register, and the lower 8 bits as the PMDLL register, however, these registers can be read or written in 8-bit or 1-bit units.

After reset: FFFFH R/W Address: FFFFF024H, FFFFF025H

| | | | | | | | | |
|-------------------------------|-------|-------|-------|-------|--------|--------|-------|-------|
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PMDL (PMDLH ^{Note}) | 1 | 1 | 1 | 1 | 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 11) |
| 0 | Output mode |
| 1 | Input mode |

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

4.3.13 Port pins that function alternately as on-chip debug function

The pins shown in Table 4-13 function alternately as on-chip debug pins. After an external reset, these pins are initialized as on-chip debug pins ($\overline{\text{DRST}}$, DDI, DDO, DCK, and DMS).

Table 4-13. On-Chip Debug Pins

| Pin Name | Alternate Function Pin |
|----------|--------------------------------|
| P05 | $\overline{\text{INTP2/DRST}}$ |
| P52 | KR2/TIQ03/TOQ03/DDI |
| P53 | KR3/TIQ00/TOQ00/DDO |
| P54 | KR4/DCK |
| P55 | KR5/DMS |

To use these pins as port pins, not as on-chip debug pins, the following actions must be taken after an external reset.

- <1> Clear the OCDM0 bit of the OCDM register (special register) to 0.
- <2> Fix the P05/ $\overline{\text{INTP2/DRST}}$ pin to the low level until the above action has been taken.

When the on-chip debug function is not used, inputting a high level to the $\overline{\text{DRST}}$ pin before the above actions are taken may cause a malfunction (CPU deadlock). Exercise utmost care in handling the P05 pin.

When a high level is not input to the P05/ $\overline{\text{INTP2/DRST}}$ pin (when this pin is fixed to low level), it is not necessary to manipulate the OCDM.OCDM0 bit.

Because a pull-down resistor (30 k Ω TYP.) is connected to the buffer of the P05/ $\overline{\text{INTP2/DRST}}$ pin, the pin does not have to be fixed to the low level by an external source. The pull-down resistor is disconnected by clearing the OCDM0 bit to 0.

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

4.3.14 Register settings to use port pins as alternate-function pins

Table 4-14. Using Port Pin as Alternate-Function Pin (1/4)

| Pin Name | Alternate-Function Pin | | PMn Register | PMCn Register | PFCm Register | PFCEm Register | Other Bits (Register) |
|-----------------------|---------------------------|--------|----------------------|----------------------|---------------|----------------|-------------------------------|
| | Name | I/O | | | | | |
| P00 | TIP31 | Input | Setting not required | PMC00 = 1 | PFC00 = 0 | – | |
| | TOP31 | Output | Setting not required | PMC00 = 1 | PFC00 = 1 | – | |
| P01 | TIP30 | Input | Setting not required | PMC01 = 1 | PFC01 = 0 | – | |
| | TOP30 | Output | Setting not required | PMC01 = 1 | PFC01 = 1 | – | |
| P02 | NMI | Input | Setting not required | PMC02 = 1 | – | – | |
| P03 | INTP0 | Input | Setting not required | PMC03 = 1 | PFC03 = 0 | – | INTx03 (INTx0) |
| | ADTRG | Output | Setting not required | PMC03 = 1 | PFC03 = 1 | – | |
| P04 | INTP1 | Input | Setting not required | PMC04 = 1 | – | – | INTx04 (INTx0) |
| P05 ^{Note 1} | INTP2 | Input | Setting not required | PMC05 = 1 | – | – | INTx05 (INTx0) |
| | $\overline{\text{DRST}}$ | Input | Setting not required | Setting not required | – | – | OCDM0 (OCDM) = 1 |
| P06 | INTP3 | Input | Setting not required | PMC06 = 1 | – | – | INTx06 (INTx0) |
| P30 | TXDA0 | Output | Setting not required | PMC30 = 1 | – | – | |
| P31 | RXDA0 | Input | Setting not required | PMC31 = 1 | – | – | Note 2 |
| | INTP7 | Input | Setting not required | PMC31 = 1 | – | – | Note 2, INTx31 (INTx3) |
| P32 | ASCKA0 | Input | Setting not required | PMC32 = 1 | PFC32 = 0 | PFCE32 = 0 | |
| | TOP01 | Output | Setting not required | PMC32 = 1 | PFC32 = 1 | PFCE32 = 0 | |
| | TIP00 | Input | Setting not required | PMC32 = 1 | PFC32 = 0 | PFCE32 = 1 | |
| | TOP00 | Output | Setting not required | PMC32 = 1 | PFC32 = 1 | PFCE32 = 1 | |
| P33 | TIP01 | Input | Setting not required | PMC33 = 1 | PFC33 = 0 | – | |
| | TOP01 | Output | Setting not required | PMC33 = 1 | PFC33 = 1 | – | |
| P34 | TIP10 | Input | Setting not required | PMC34 = 1 | PFC34 = 0 | – | |
| | TOP10 | Output | Setting not required | PMC34 = 1 | PFC34 = 1 | – | |
| P35 | TIP11 | Input | Setting not required | PMC35 = 1 | PFC35 = 0 | – | |
| | TOP11 | Output | Setting not required | PMC35 = 1 | PFC35 = 1 | – | |
| P40 | SIB0 | Input | Setting not required | PMC40 = 1 | – | – | |
| P41 | SOB0 | Output | Setting not required | PMC41 = 1 | – | – | |
| P42 | $\overline{\text{SCKB0}}$ | I/O | Setting not required | PMC42 = 1 | – | – | |

Notes 1. After an external reset, the P05/INTP2/ $\overline{\text{DRST}}$ pin is initialized as an on-chip debug pin ($\overline{\text{DRST}}$). To not use the P05/INTP2/ $\overline{\text{DRST}}$ pin as an on-chip debug pin, see **CHAPTER 24 ON-CHIP DEBUG FUNCTION**.

- 2.** The INTP7 pin functions alternately as the RXDA0 pin. To use this pin as the RXDA0 pin, invalidate the edge detection function of the alternate-function INTP7 pin (by clearing the INTF3.INTF31 bit to 0 and the INTR3.INTR31 bit to 0). To use this pin as the INTP7 pin, stop the reception operation of UARTA0 (by clearing the UA0CTL0.UA0RXE bit to 0).

Remarks 1. The port register (Pn) does not have to be set when the alternate function is used.

- 2.** INTxn = INTFn, INTRn

Table 4-14. Using Port Pin as Alternate-Function Pin (2/4)

| Pin Name | Alternate-Function Pin | | PMn Register | PMCn Register | PFCm Register | PFCEm Register | Other Bits (Register) |
|----------|------------------------|--------|----------------------|----------------------|----------------------|----------------------|-----------------------|
| | Name | I/O | | | | | |
| P50 | KR0 | Input | Setting not required | PMC50 = 1 | PFC50 = 1 | PFCE50 = 0 | Note 1 |
| | TIQ01 | Input | Setting not required | PMC50 = 1 | PFC50 = 1 | PFCE50 = 0 | Note 1 |
| | TOQ01 | Output | Setting not required | PMC50 = 1 | PFC50 = 0 | PFCE50 = 1 | |
| P51 | KR1 | Input | Setting not required | PMC51 = 1 | PFC51 = 1 | PFCE54 = 0 | Note 1 |
| | TIQ02 | Input | Setting not required | PMC51 = 1 | PFC51 = 1 | PFCE51 = 0 | Note 1 |
| | TOQ02 | Output | Setting not required | PMC51 = 1 | PFC51 = 0 | PFCE51 = 1 | |
| P52 | KR2 | Input | Setting not required | PMC52 = 1 | PFC52 = 1 | PFCE52 = 0 | Note 1 |
| | TIQ03 | Input | Setting not required | PMC52 = 1 | PFC52 = 1 | PFCE52 = 0 | Note 1 |
| | TOQ03 | Output | Setting not required | PMC52 = 1 | PFC52 = 0 | PFCE52 = 1 | |
| | DDI ^{Note 2} | Input | Setting not required | Setting not required | Setting not required | Setting not required | OCDM0 (OCDM) = 1 |
| P53 | KR3 | Input | Setting not required | PMC53 = 1 | PFC53 = 1 | PFCE53 = 0 | Note 1 |
| | TIQ00 | Input | Setting not required | PMC53 = 1 | PFC53 = 1 | PFCE53 = 0 | Note 1 |
| | TOQ00 | Output | Setting not required | PMC53 = 1 | PFC53 = 0 | PFCE53 = 1 | |
| | DDO ^{Note 2} | Output | Setting not required | Setting not required | Setting not required | Setting not required | OCDM0 (OCDM) = 1 |
| P54 | KR4 | Input | Setting not required | PMC54 = 1 | PFC54 = 1 | – | |
| | DCK ^{Note 2} | Output | Setting not required | Setting not required | Setting not required | – | OCDM0 (OCDM) = 1 |
| P55 | KR5 | Input | Setting not required | PMC55 = 1 | PFC55 = 1 | – | |
| | DMS ^{Note 2} | Output | Setting not required | Setting not required | Setting not required | – | OCDM0 (OCDM) = 1 |

Notes 1. The KRn pin functions alternately as the TIQ0m pin. To use this pin as the TIQ0m pin, invalidate the key return detection function of the alternate-function KRn pin (by clearing the KRM.KRMn bit to 0). To use this pin as the KRn pin, invalidate the edge detection function of the alternate-function TIQ0m pin (n = 0 to 3, m = 0 to 3).

| Pin Name | When Used as TIQ0m Pin | When Used as KRn Pin |
|-----------|------------------------------|---|
| KR0/TIQ01 | KRM0 bit of KRM register = 0 | TQ0TIG2, TQ0TIG3 bits of TQ0IOC1 register = 0 |
| KR1/TIQ02 | KRM1 bit of KRM register = 0 | TQ0TIG4, TQ0TIG5 bits of TQ0IOC1 register = 0 |
| KR2/TIQ03 | KRM2 bit of KRM register = 0 | TQ0TIG6, TQ0TIG7 bits of TQ0IOC1 register = 0 |
| KR3/TIQ00 | KRM3 bit of KRM register = 0 | TQ0TIG0, TQ0TIG1 bits of TQ0IOC1 register = 0 TQ0EES0, TQ0EES1 bits of TQ0IOC2 register = 0 TQ0ETS0, TQ0ETS1 bits of TQ0IOC2 register = 0 |

2. The DDI, DDO, DCK, and DMS pins are on-chip debug pins. To not use these pins as on-chip debug pins after an external reset, see **CHAPTER 24 ON-CHIP DEBUG FUNCTION**.

Caution If the control mode is specified by using the PMC5 register when the PFC5.PFC5n bit and the PFCE5.PFCE5n bit are the default values (0), the output becomes undefined. For this reason, first set the PFC5.PFC5n bit and the PFCE5.PFCE5n bit, and then set the PMC5n bit to 1 to set the control mode.

Remarks 1. The port register (Pn) does not have to be set when the alternate function is used.
2. INTxn = INTFn, INTRn

Table 4-14. Using Port Pin as Alternate-Function Pin (3/4)

| Pin Name | Alternate-Function Pin | | PMn Register | PMCn Register | PFCm Register | PFCEm Register | Other Bits (Register) |
|----------|------------------------|--------|-----------------------------|---------------|---------------|----------------|-----------------------|
| | Name | I/O | | | | | |
| P70 | ANI0 | Input | PM70 = 1 ^{Note 1} | – | – | – | |
| P71 | ANI1 | Input | PM71 = 1 ^{Note 1} | – | – | – | |
| P72 | ANI2 | Input | PM72 = 1 ^{Note 1} | – | – | – | |
| P73 | ANI3 | Input | PM73 = 1 ^{Note 1} | – | – | – | |
| P74 | ANI4 | Input | PM74 = 1 ^{Note 1} | – | – | – | |
| P75 | ANI5 | Input | PM75 = 1 ^{Note 1} | – | – | – | |
| P76 | ANI6 | Input | PM76 = 1 ^{Note 1} | – | – | – | |
| P77 | ANI7 | Input | PM77 = 1 ^{Note 1} | – | – | – | |
| P78 | ANI8 | Input | PM78 = 1 ^{Note 1} | – | – | – | |
| P79 | ANI9 | Input | PM79 = 1 ^{Note 1} | – | – | – | |
| P710 | ANI10 | Input | PM710 = 1 ^{Note 1} | – | – | – | |
| P711 | ANI11 | Input | PM711 = 1 ^{Note 1} | – | – | – | |
| P90 | KR6 | Input | Setting not required | PMC90 = 1 | PFC90 = 1 | PFCE90 = 0 | |
| | TXDA1 | Output | Setting not required | PMC90 = 1 | PFC90 = 0 | PFCE90 = 1 | |
| P91 | KR7 ^{Note 2} | Input | Setting not required | PMC91 = 1 | PFC91 = 1 | PFCE91 = 0 | |
| | | | | | PFC91 = 0 | PFCE91 = 1 | |
| | RXDA1 | Input | Setting not required | PMC91 = 1 | PFC91 = 0 | PFCE91 = 1 | |
| P96 | TIP21 | Input | Setting not required | PMC96 = 1 | PFC96 = 0 | PFCE96 = 1 | |
| | TOP21 | Output | Setting not required | PMC96 = 1 | PFC96 = 1 | PFCE96 = 1 | |
| P97 | SIB1 | Input | Setting not required | PMC97 = 1 | PFC97 = 1 | PFCE97 = 0 | |
| | TIP20 | Input | Setting not required | PMC97 = 1 | PFC97 = 0 | PFCE97 = 1 | |
| | TOP20 | Output | Setting not required | PMC97 = 1 | PFC97 = 1 | PFCE97 = 1 | |
| P98 | SOB1 | Output | Setting not required | PMC98 = 1 | PFC98 = 1 | – | |
| P99 | SCKB1 | I/O | Setting not required | PMC99 = 1 | PFC99 = 1 | – | |
| P913 | INTP4 | Input | Setting not required | PMC913 = 1 | PFC913 = 1 | PFCE913 = 0 | INTx913 (INTx9H) |
| | PCL | Output | Setting not required | PMC913 = 1 | PFC913 = 0 | PFCE913 = 1 | |
| P914 | INTP5 | Input | Setting not required | PMC914 = 1 | PFC914 = 1 | – | INTx914 (INTx9H) |
| P915 | INTP6 | Input | Setting not required | PMC915 = 1 | PFC915 = 1 | – | INTx915 (INTx9H) |

Notes 1. Set PM7n to 1 to use the alternate function of P7n (ANIn).

2. The KR7 pin and RXDA1 pin are alternate-function pins.

When using the pin as the RXDA1 pin, disable KR7 pin key return detection. (Clear the KRM.KRM7 bit to 0.)

Also, when using the pin as the KR7 pin, it is recommended to set the PFC91 bit to 1 and clear the PFCE91 bit to 0.

Caution If the control mode is specified by using the PMC9 register when the PFC9.PFC9n bit and the PFCE9.PFCE9n bit are the default values (0), the output becomes undefined.

For this reason, first set the PFC9.PFC9n bit and the PFCE9.PFCE9n bit, and then set the PMC9n bit to 1 to set the control mode.

Remarks 1. The port register (Pn) does not have to be set when the alternate function is used.

2. INTxn = INTFn, INTRn

Table 4-14. Using Port Pin as Alternate-Function Pin (4/4)

| Pin Name | Alternate-Function Pin | | PMn Register | PMCn Register | PFCm Register | PFCEm Register | Other Bits (Register) |
|----------|------------------------|--------|----------------------|----------------------|---------------|----------------|-----------------------|
| | Name | I/O | | | | | |
| PCM1 | CLKOUT | Output | Setting not required | PMCCM1 = 1 | – | – | |
| PDL5 | FLMD1 | Input | Setting not required | Setting not required | – | – | Note |

Note The FLMD1 pin does not have to be manipulated by using a port control register because it is used in the flash programming mode. For details, see **CHAPTER 22 FLASH MEMORY**.

Remark The port register (Pn) does not have to be set when the alternate function is used.

4.4 Block Diagrams of Port

Figure 4-2. Block Diagram of Type A-1

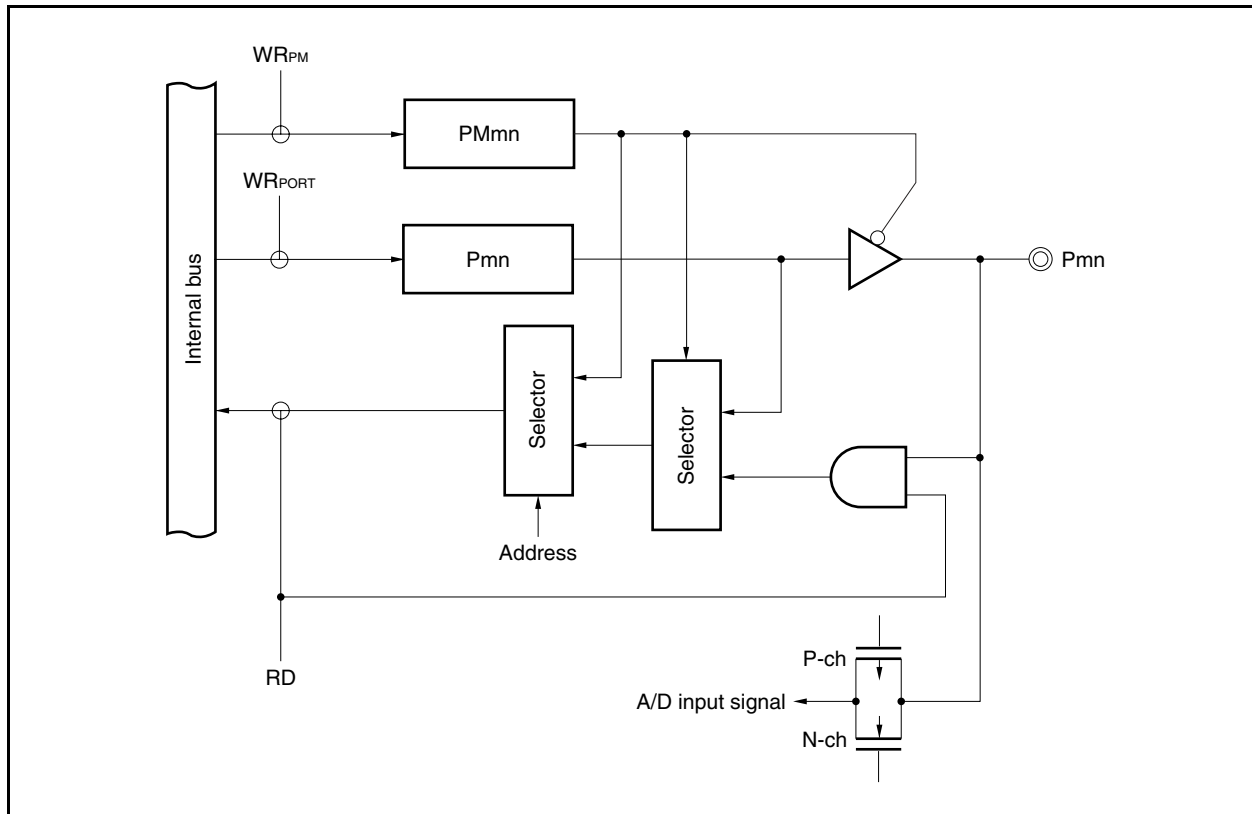


Figure 4-3. Block Diagram of Type B-1

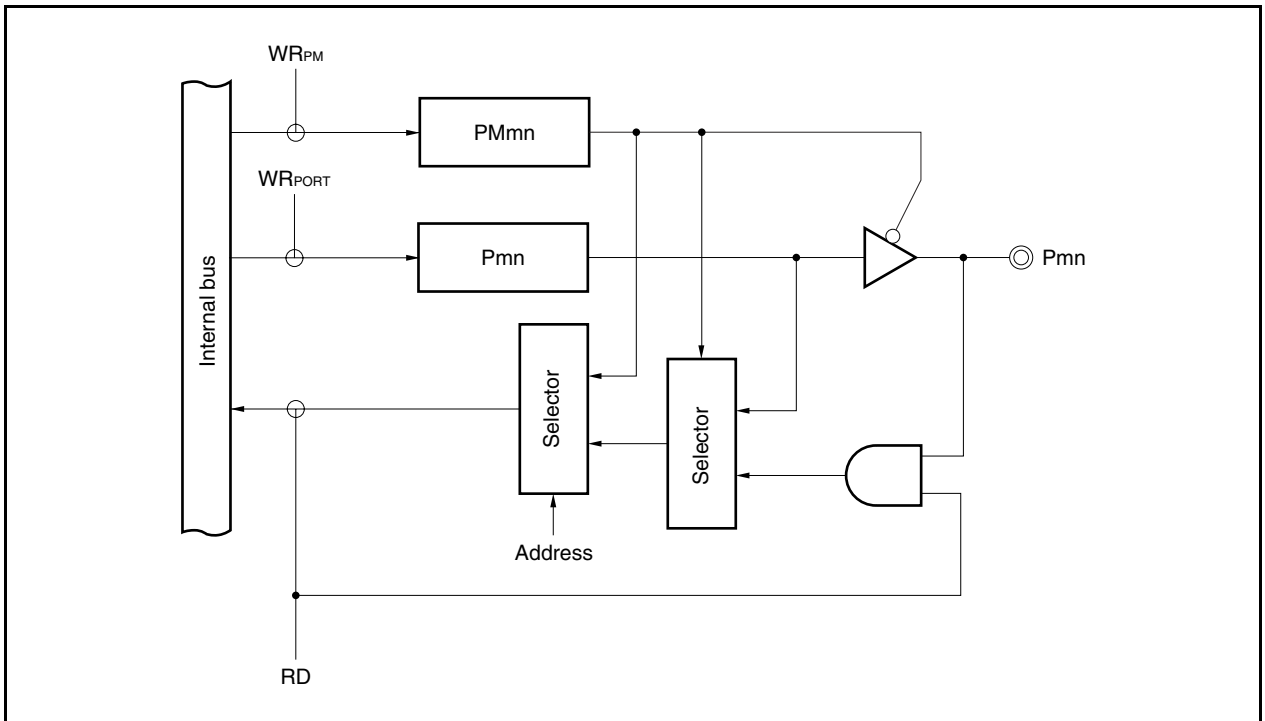


Figure 4-4. Block Diagram of Type C-1

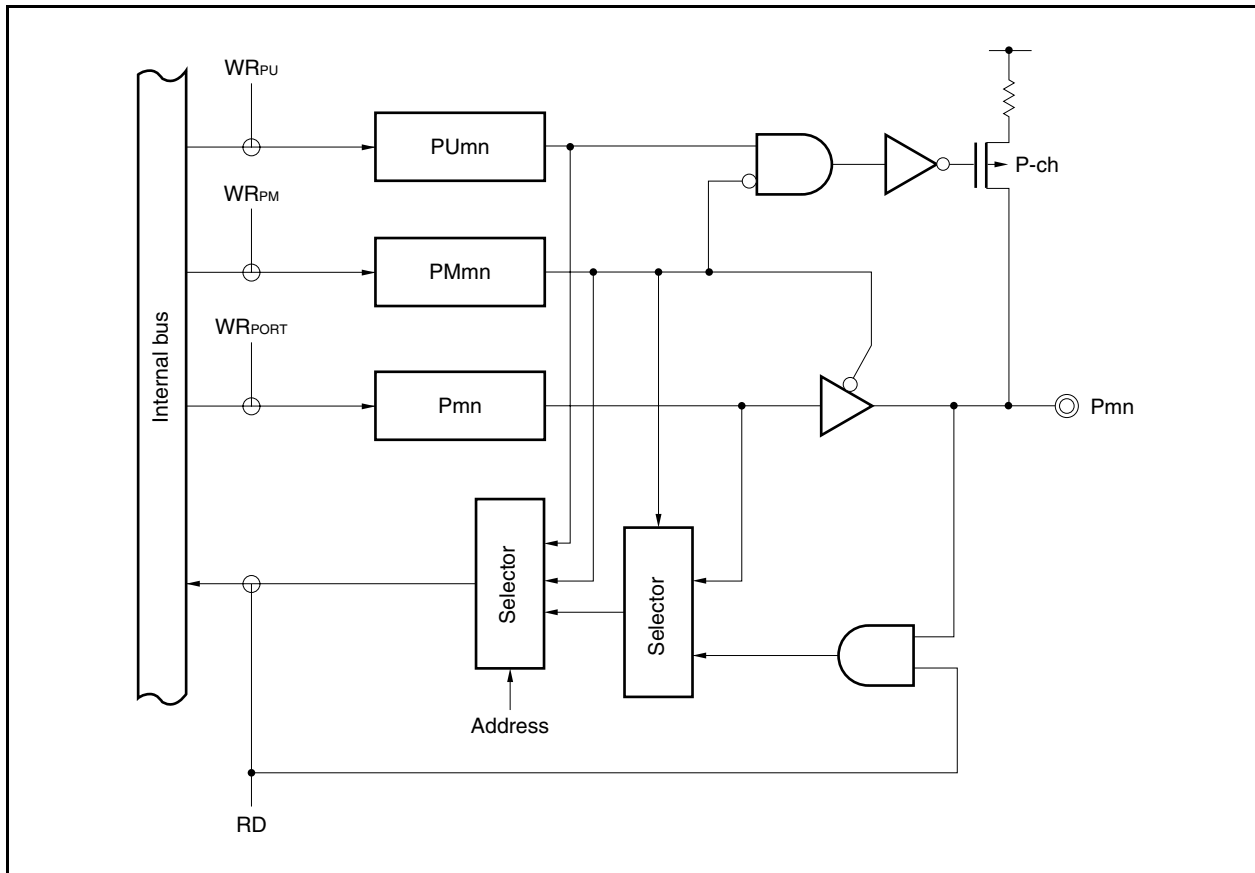


Figure 4-5. Block Diagram of Type D-2

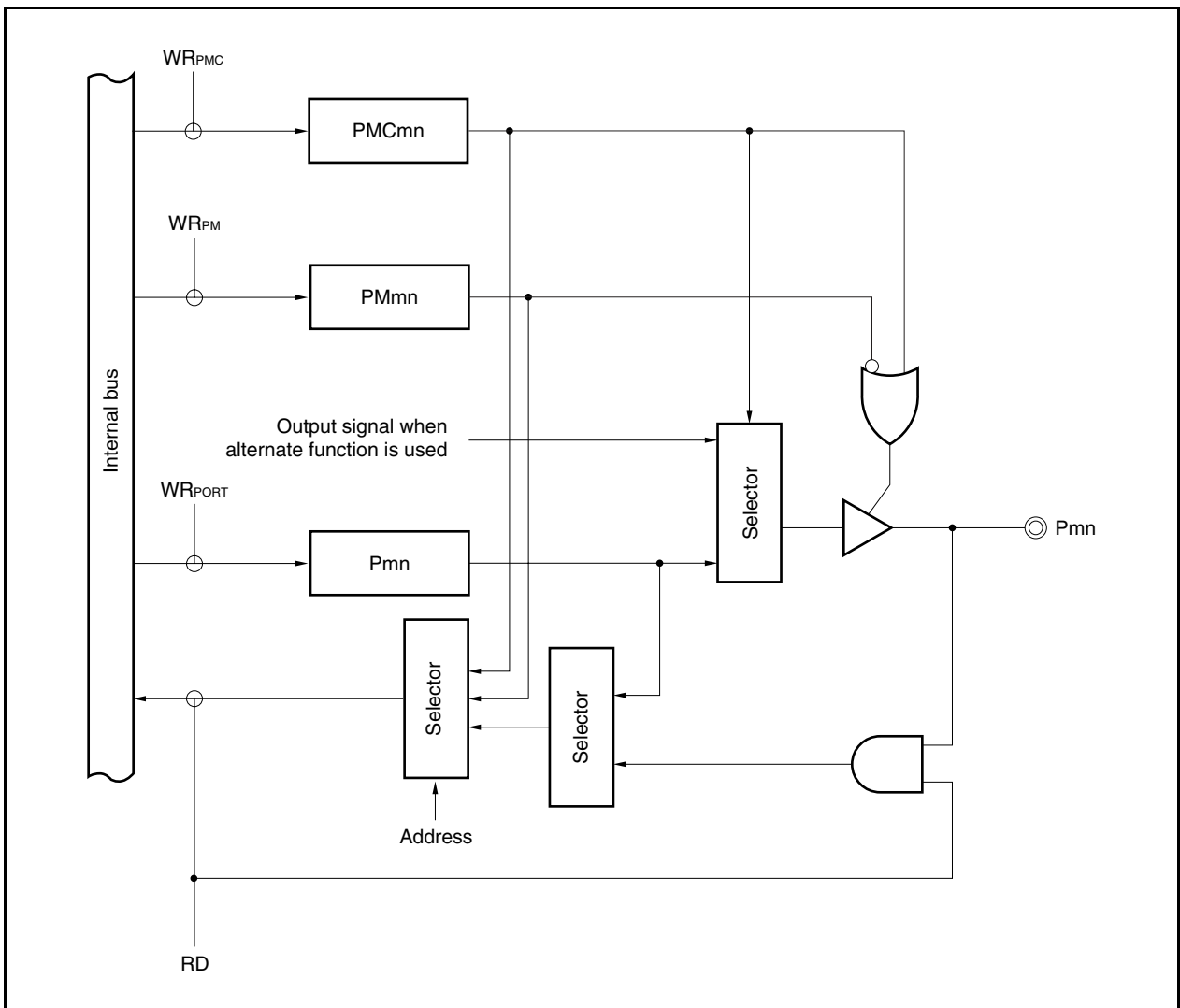
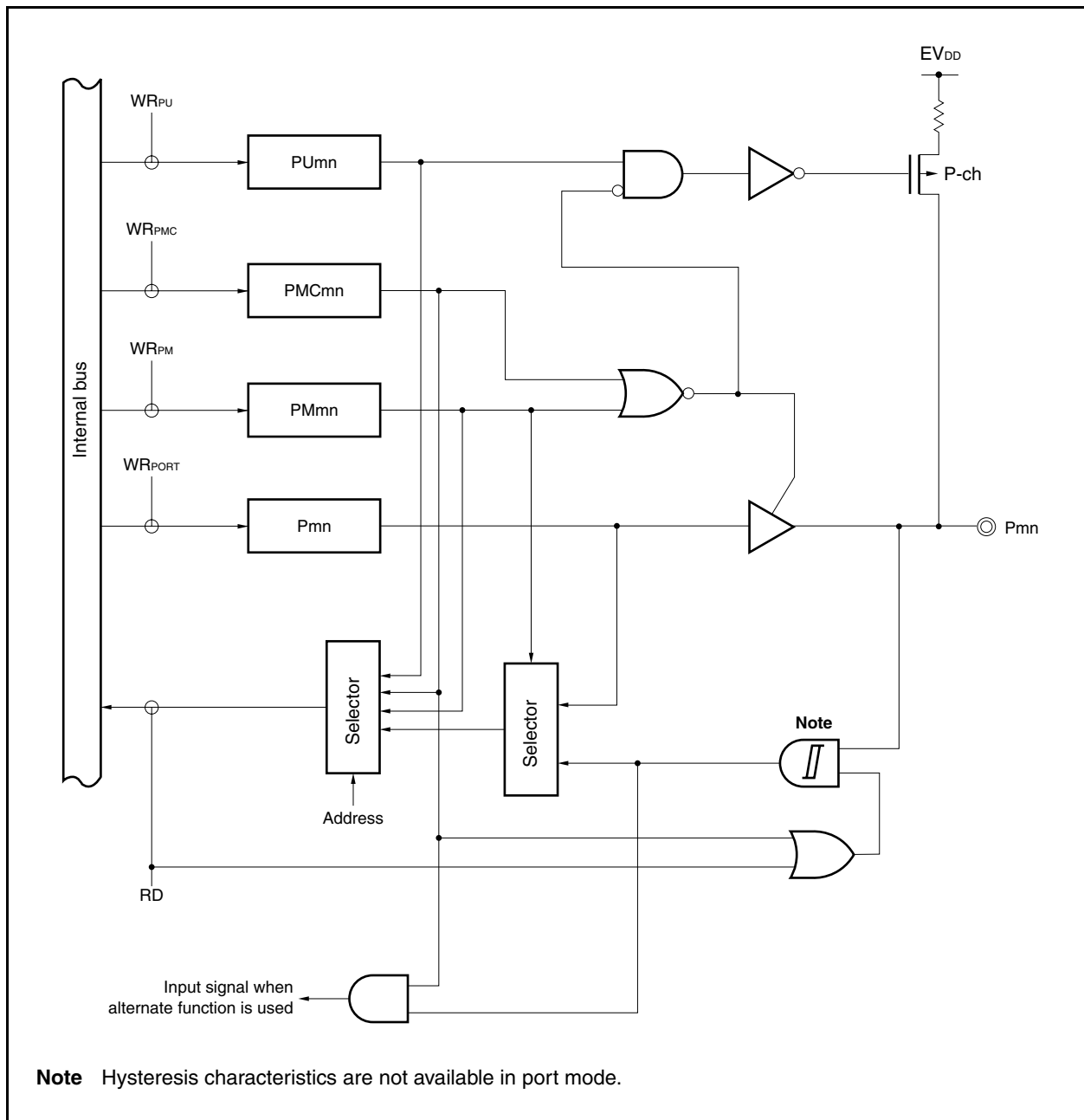


Figure 4-6. Block Diagram of Type E-1



Note Hysteresis characteristics are not available in port mode.

Figure 4-7. Block Diagram of Type E-2

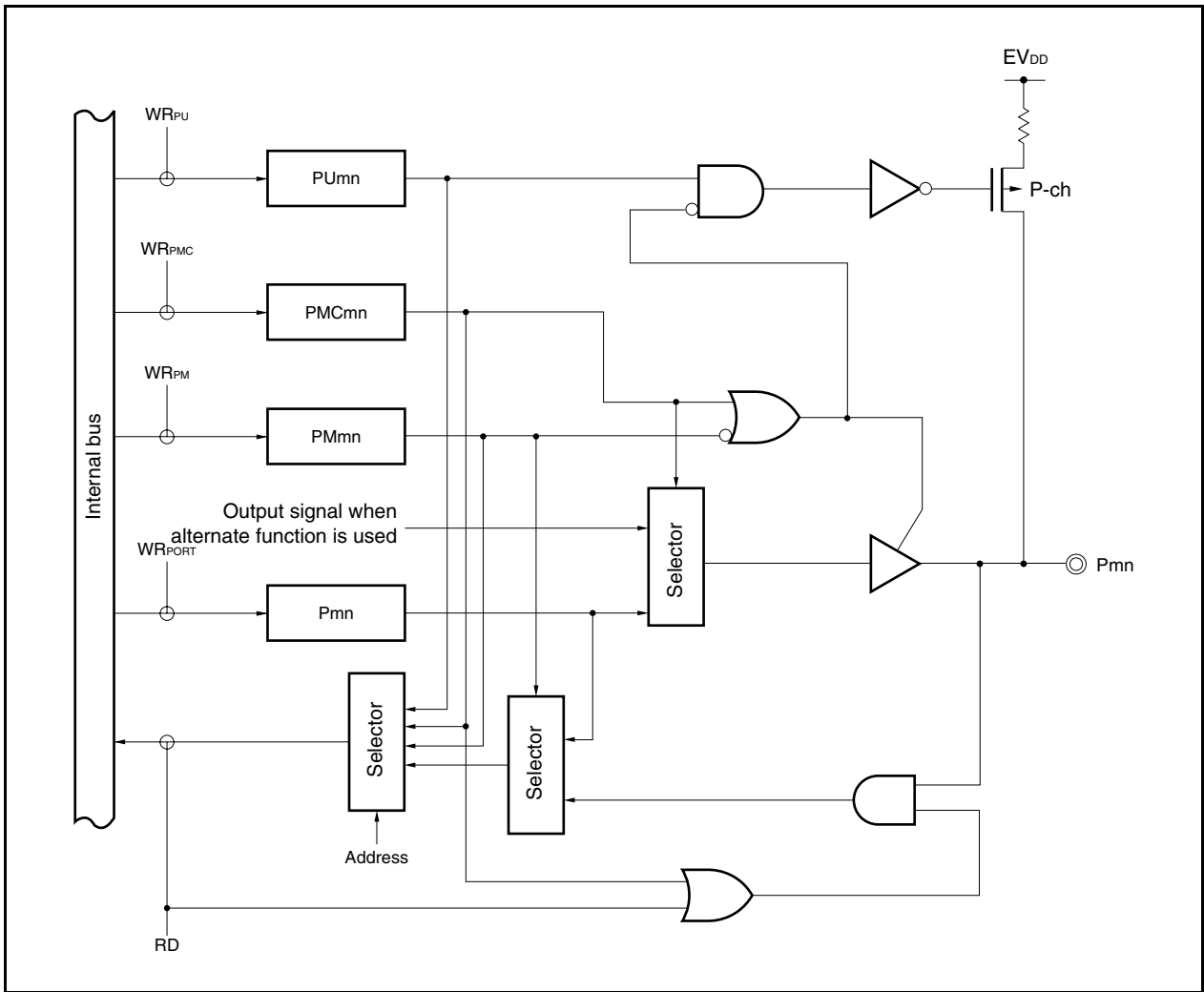


Figure 4-8. Block Diagram of Type E-3

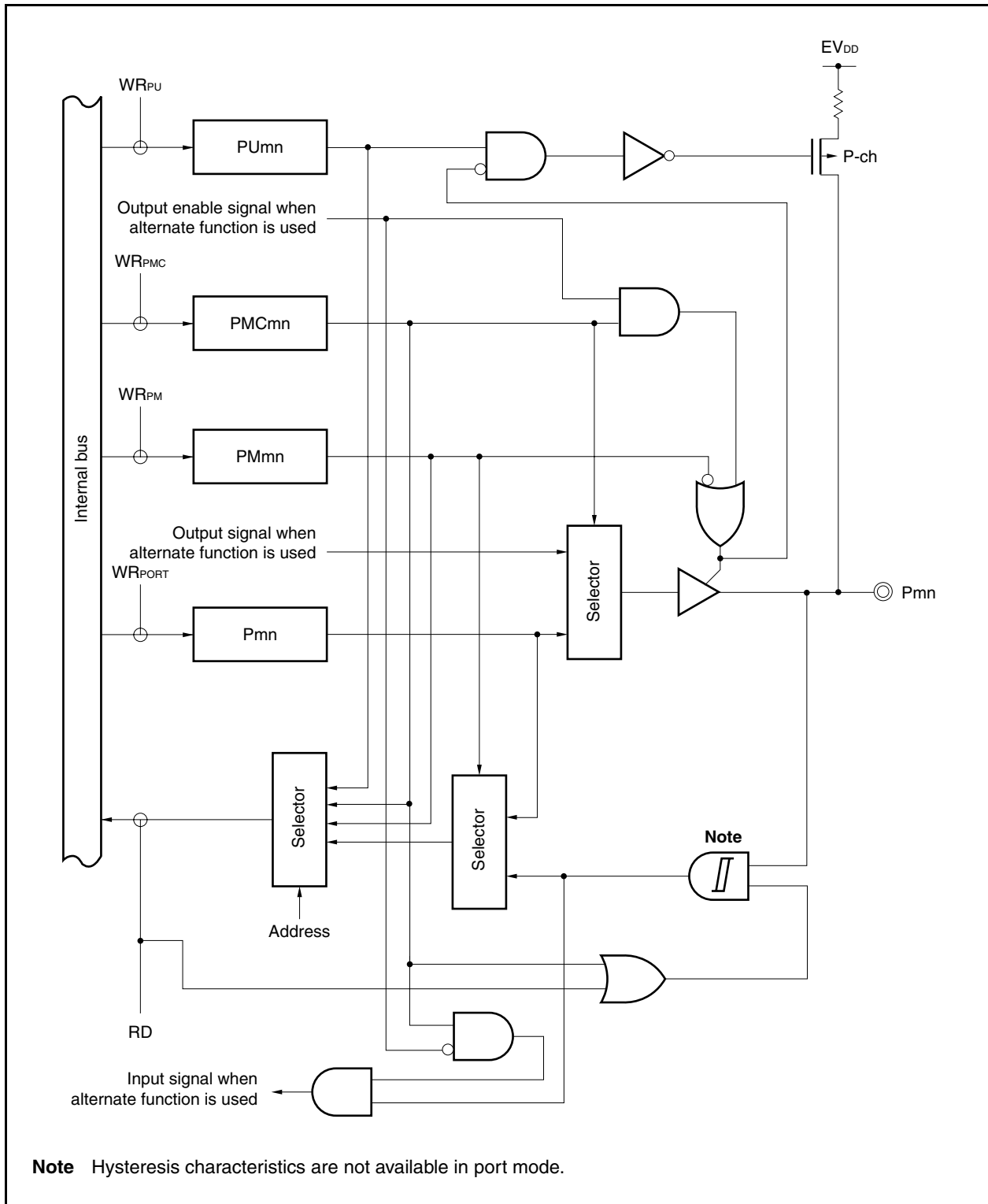


Figure 4-9. Block Diagram of Type G-1

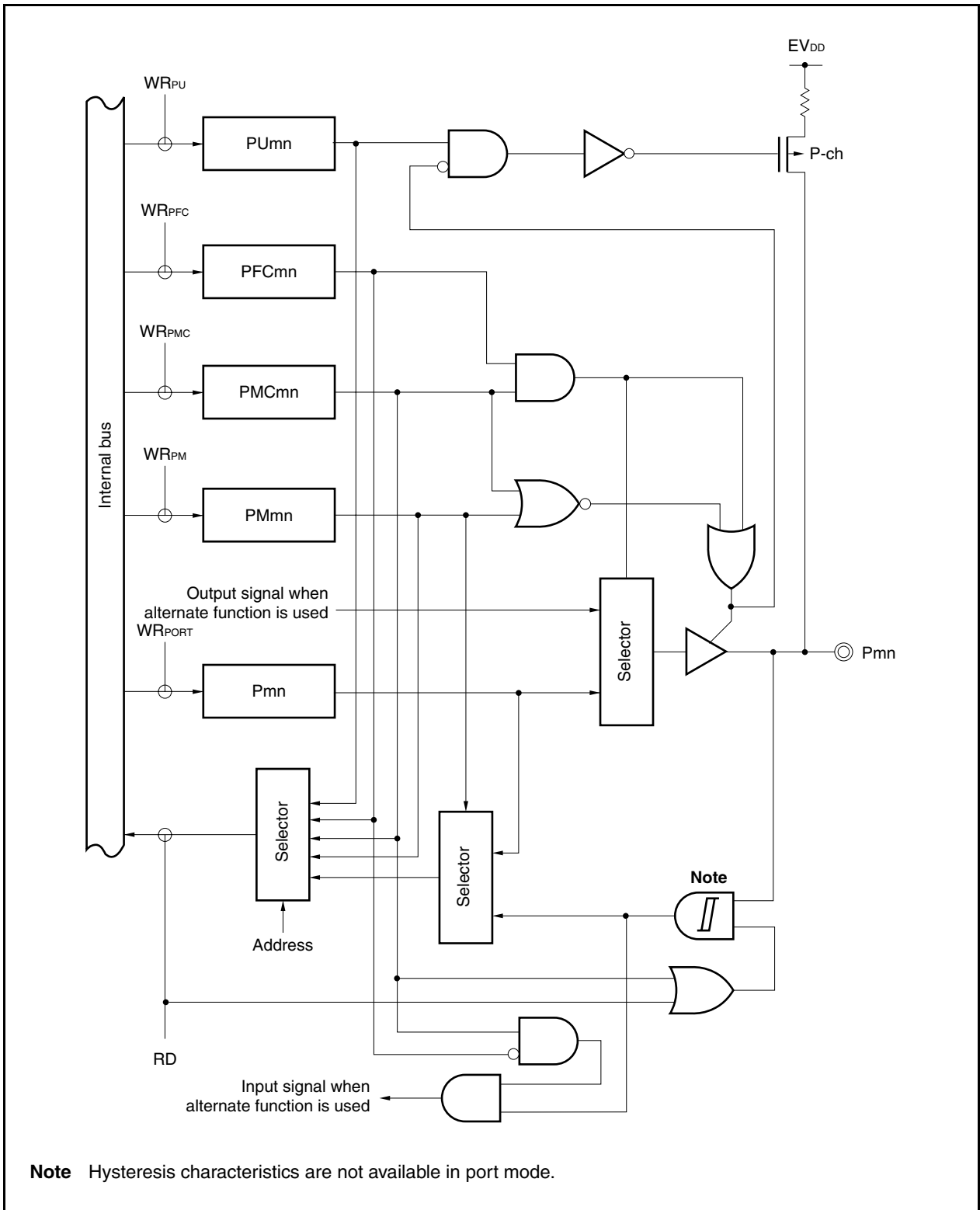
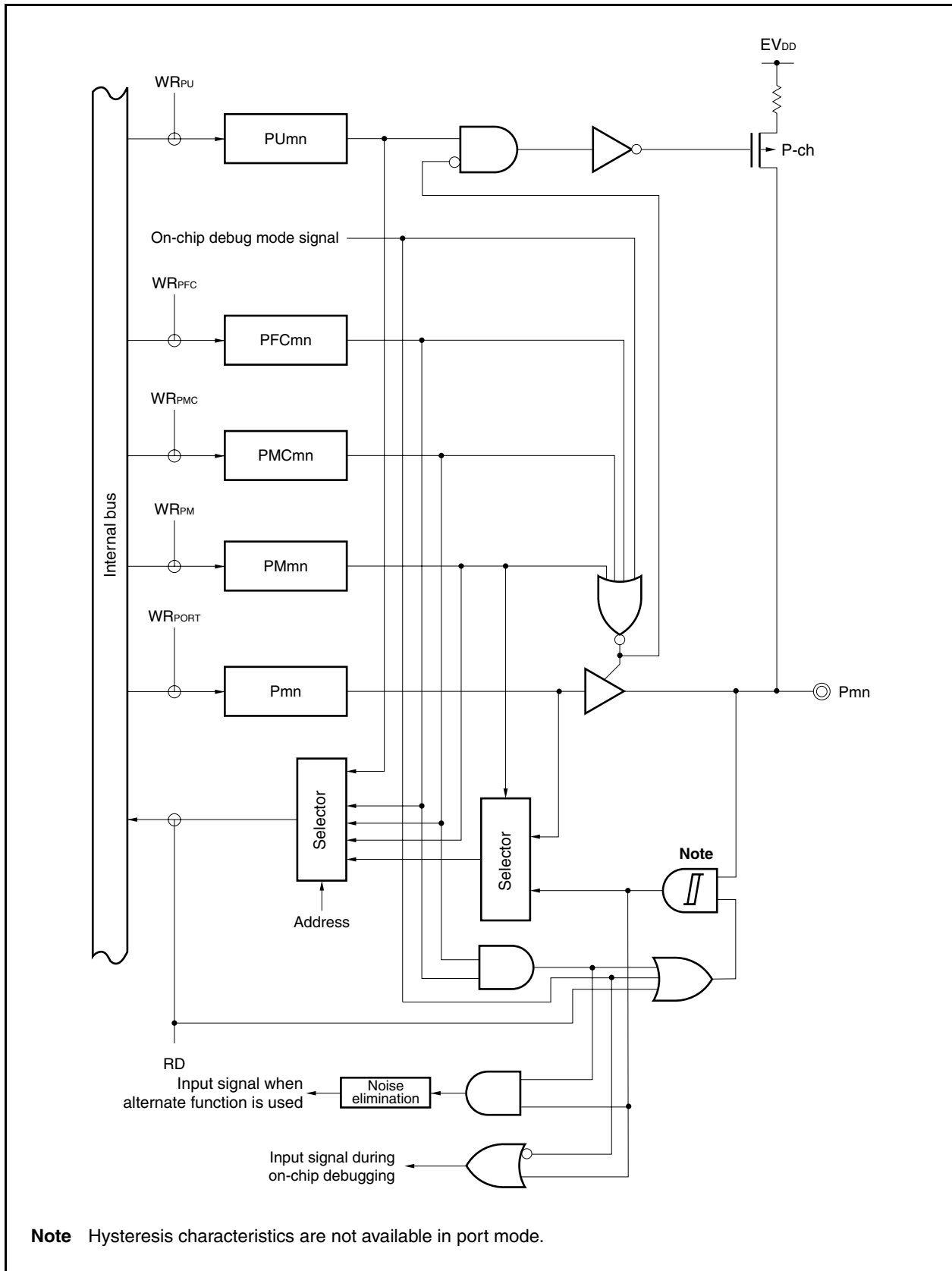


Figure 4-10. Block Diagram of Type G-2



Note Hysteresis characteristics are not available in port mode.

Figure 4-11. Block Diagram of Type G-3

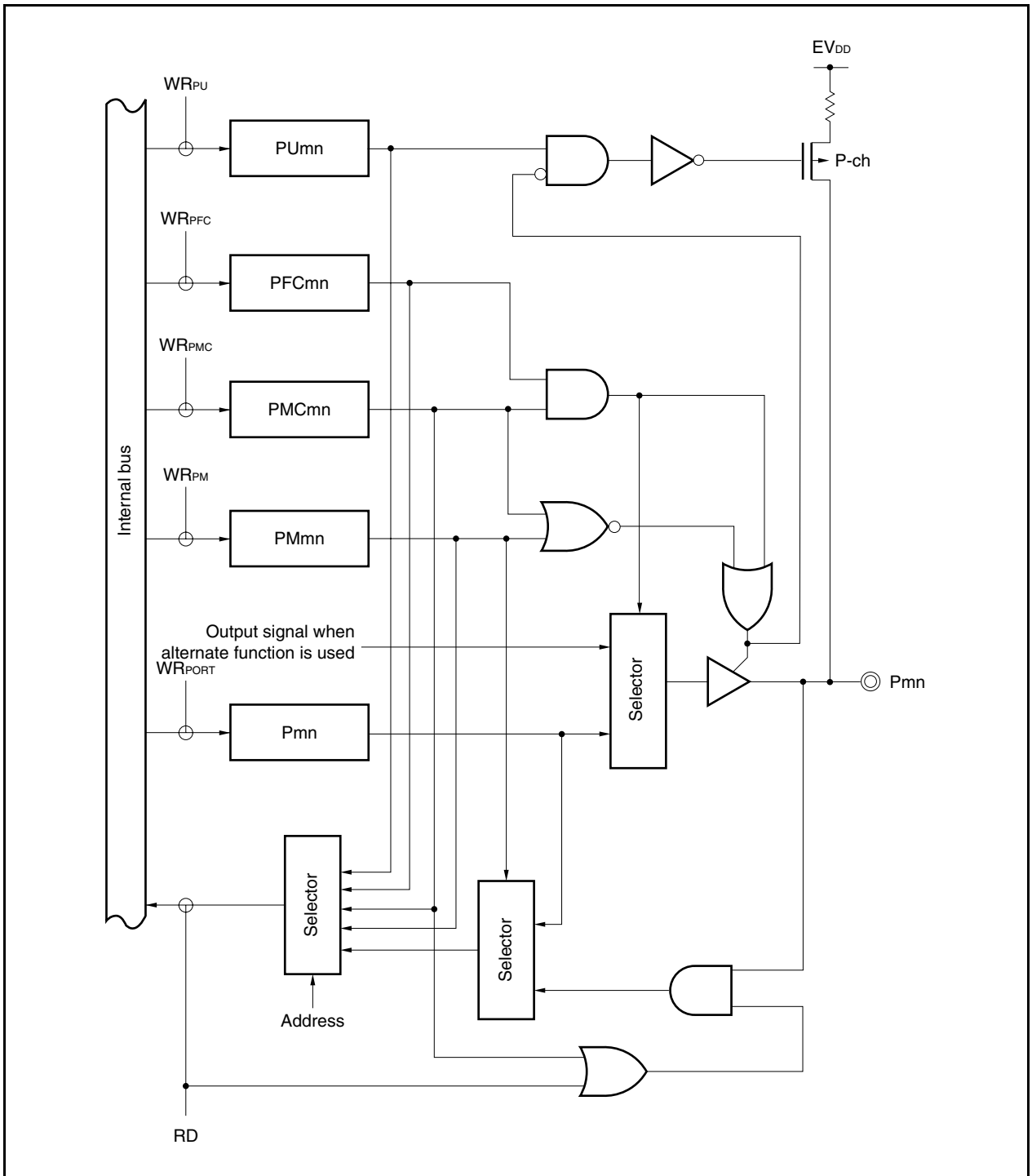


Figure 4-12. Block Diagram of Type G-5

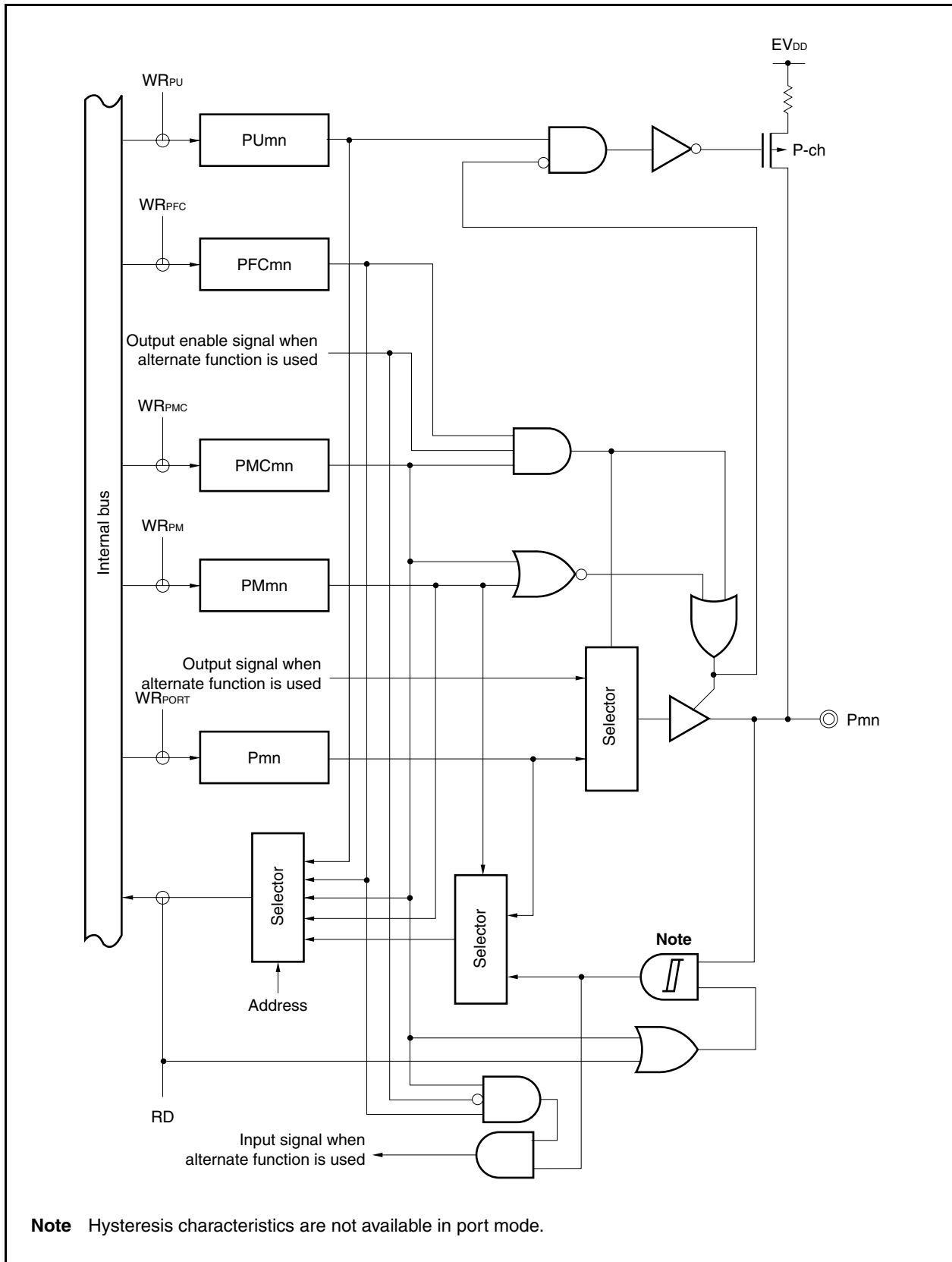
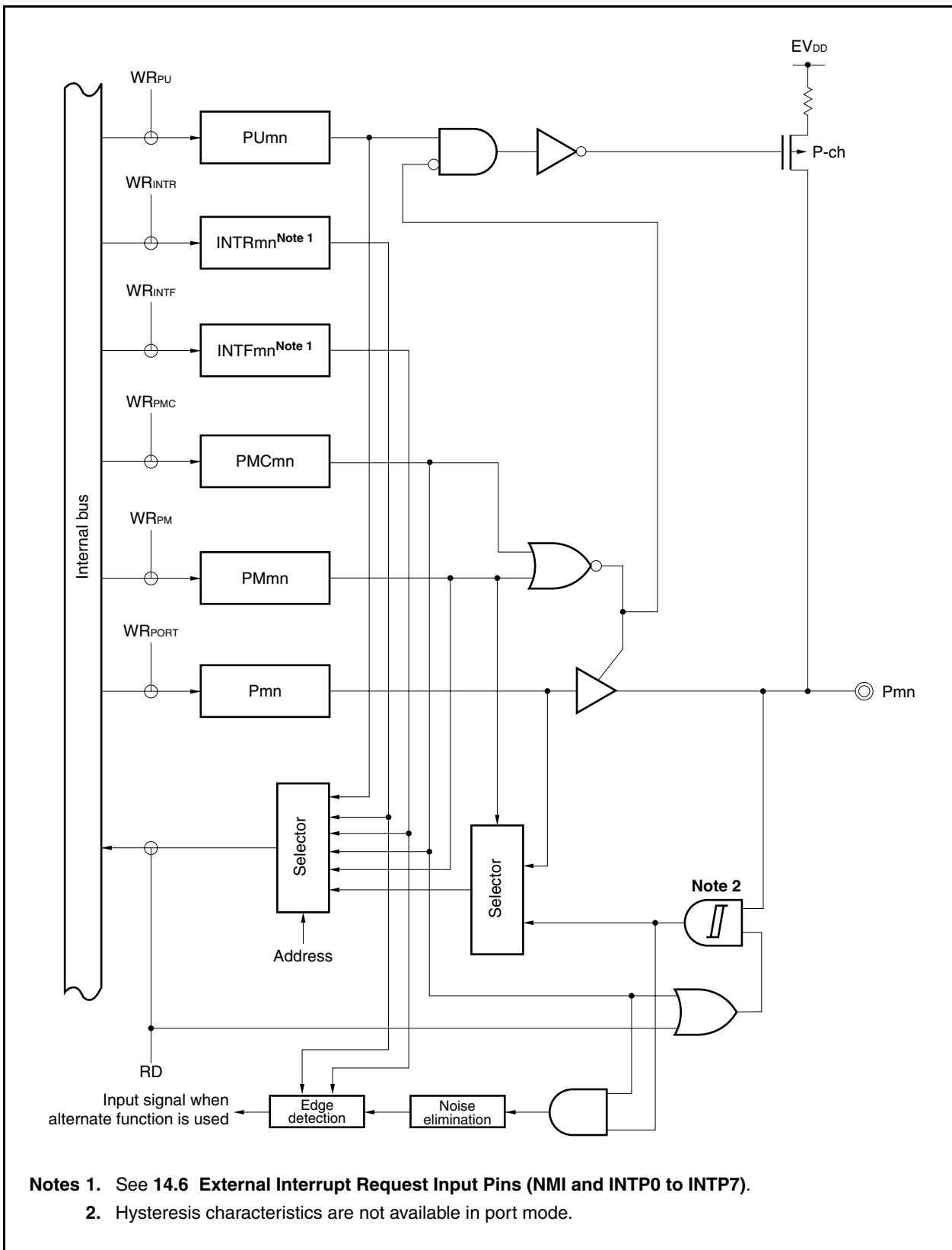


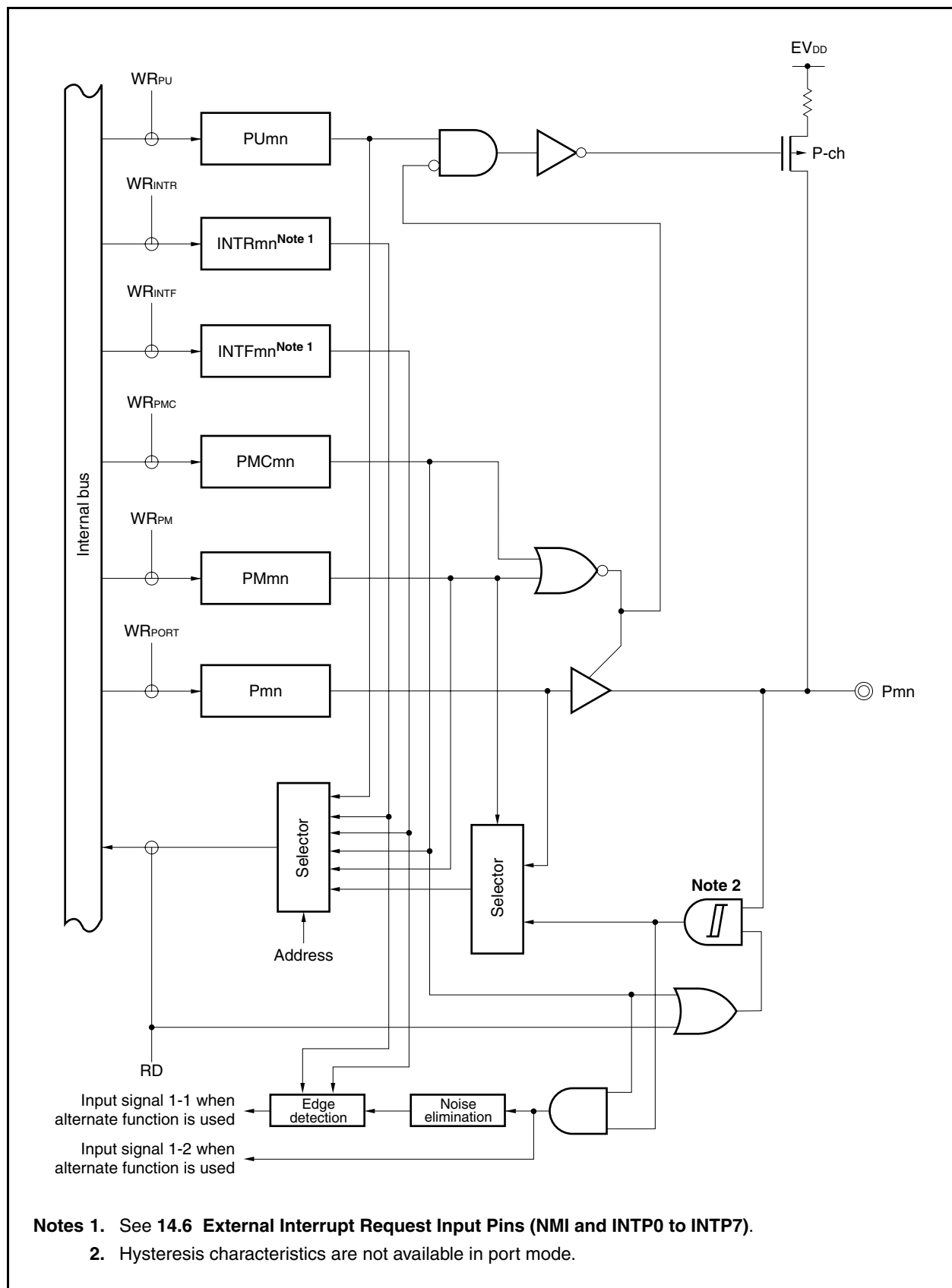
Figure 4-13. Block Diagram of Type L-1



Notes 1. See 14.6 External Interrupt Request Input Pins (NMI and INTP0 to INTP7).

2. Hysteresis characteristics are not available in port mode.

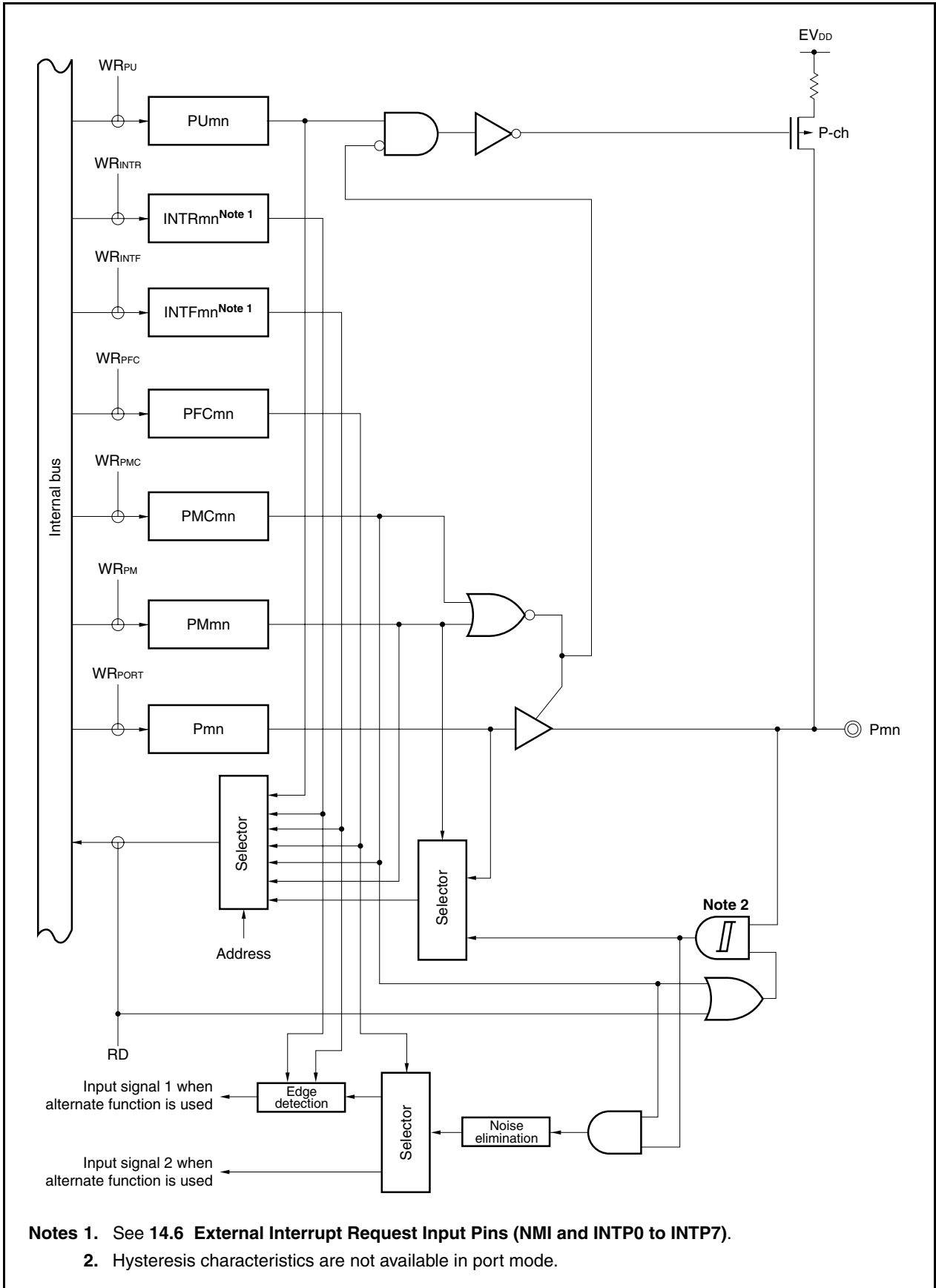
Figure 4-14. Block Diagram of Type L-2



Notes 1. See 14.6 External Interrupt Request Input Pins (NMI and INTP0 to INTP7).

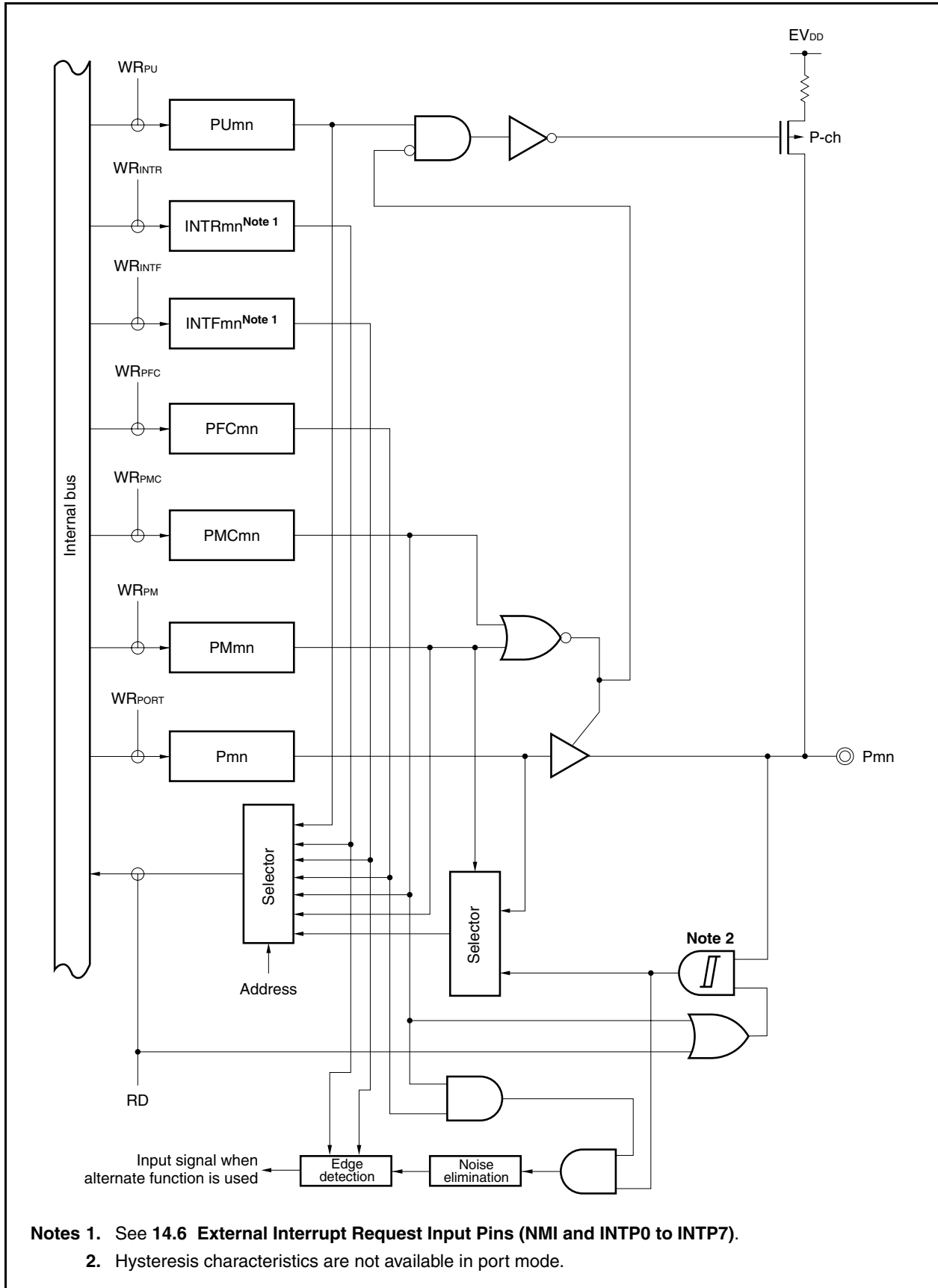
2. Hysteresis characteristics are not available in port mode.

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



- Notes 1.** See 14.6 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 N-2



- Notes 1.** See 14.6 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 U-4

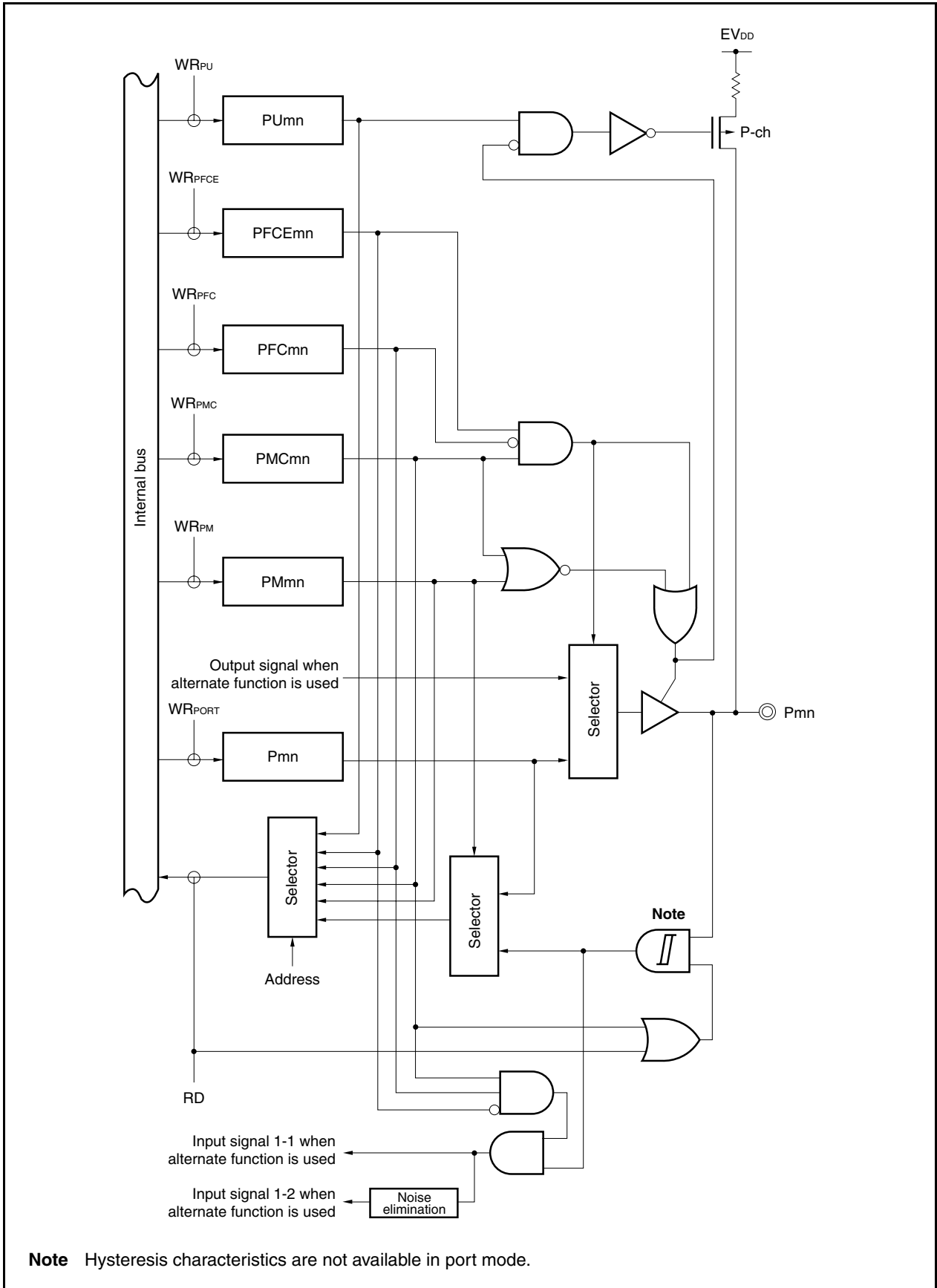


Figure 4-18. Block Diagram of Type U-5

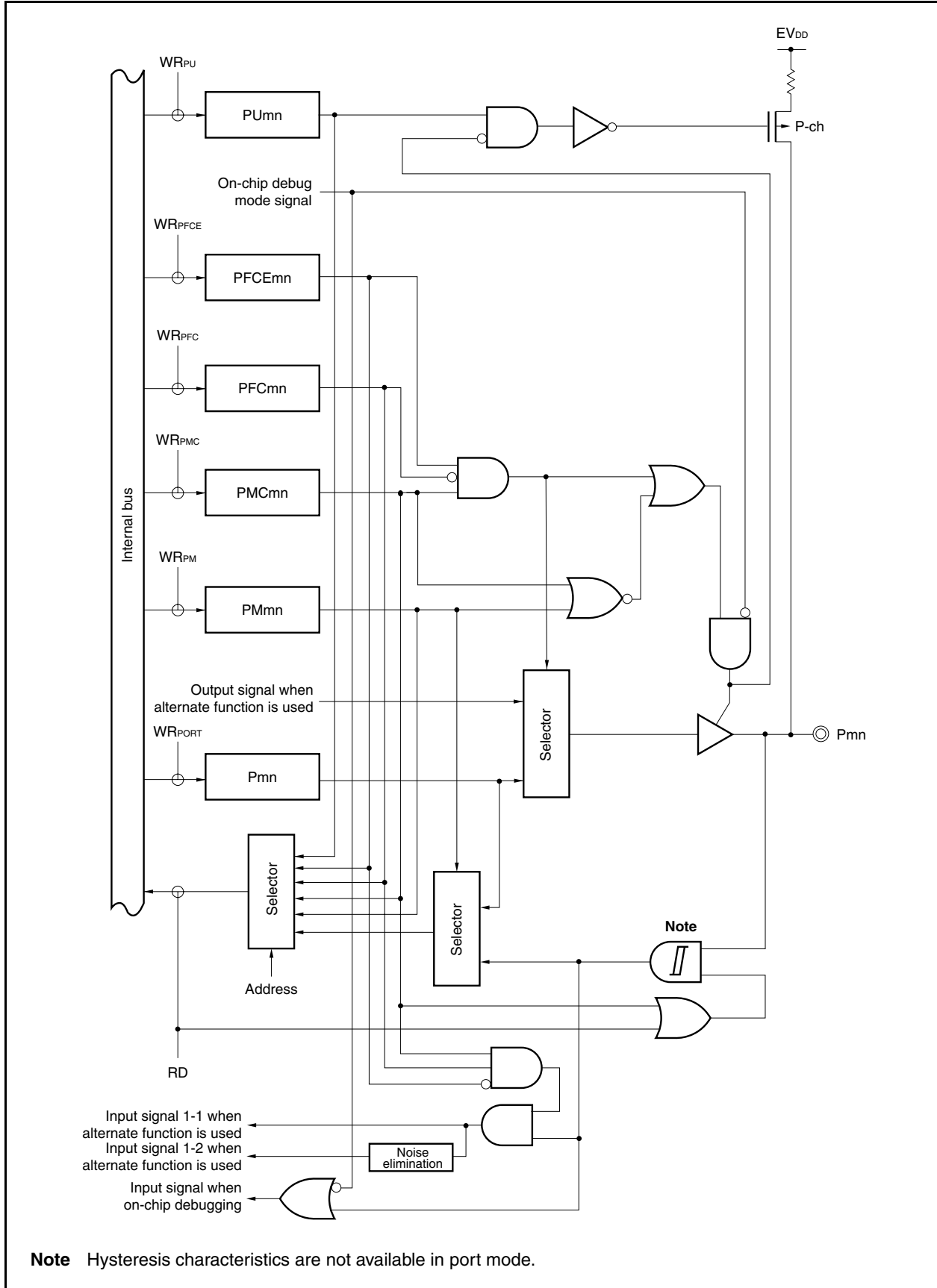


Figure 4-20. Block Diagram of Type U-7

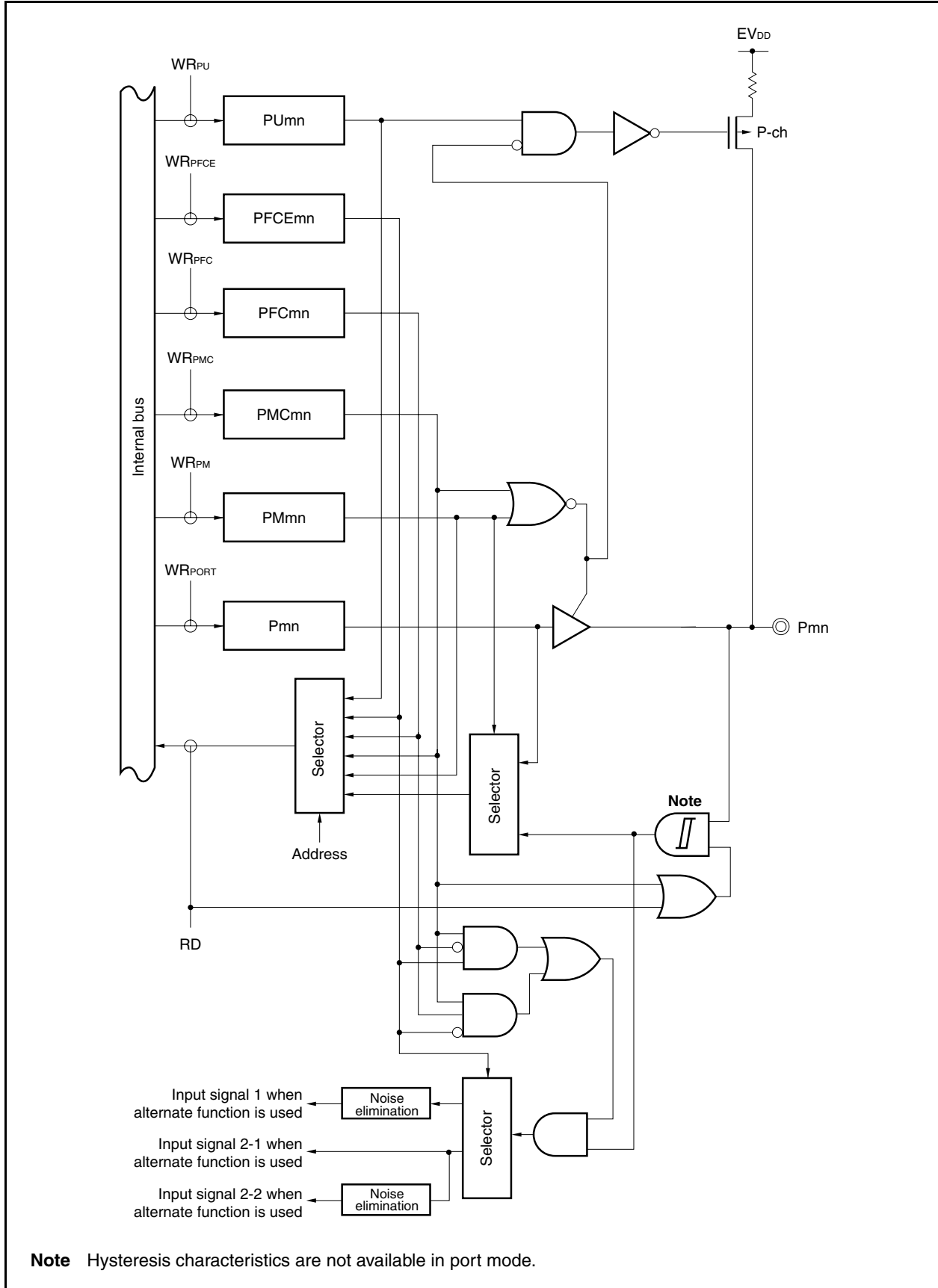
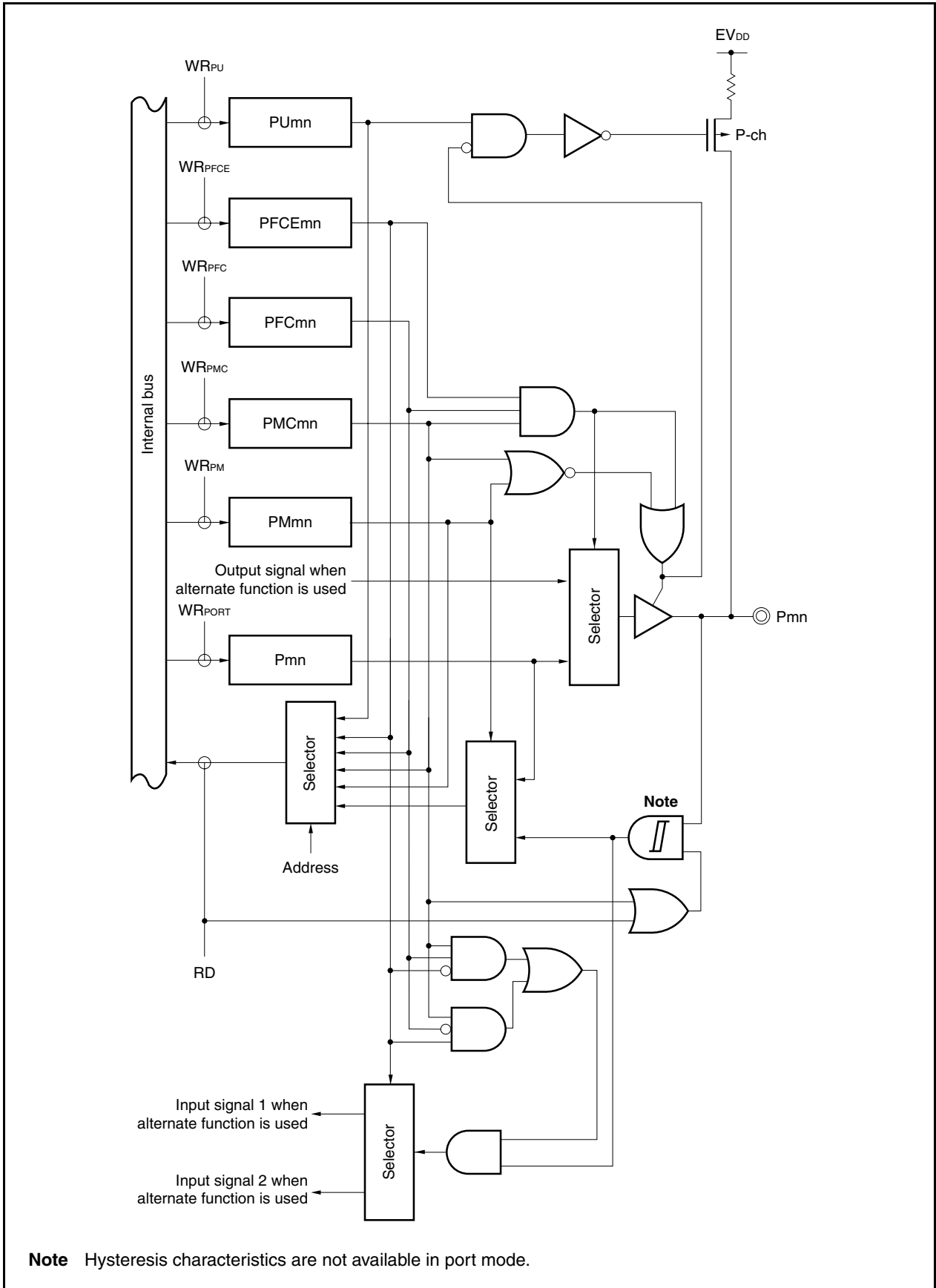
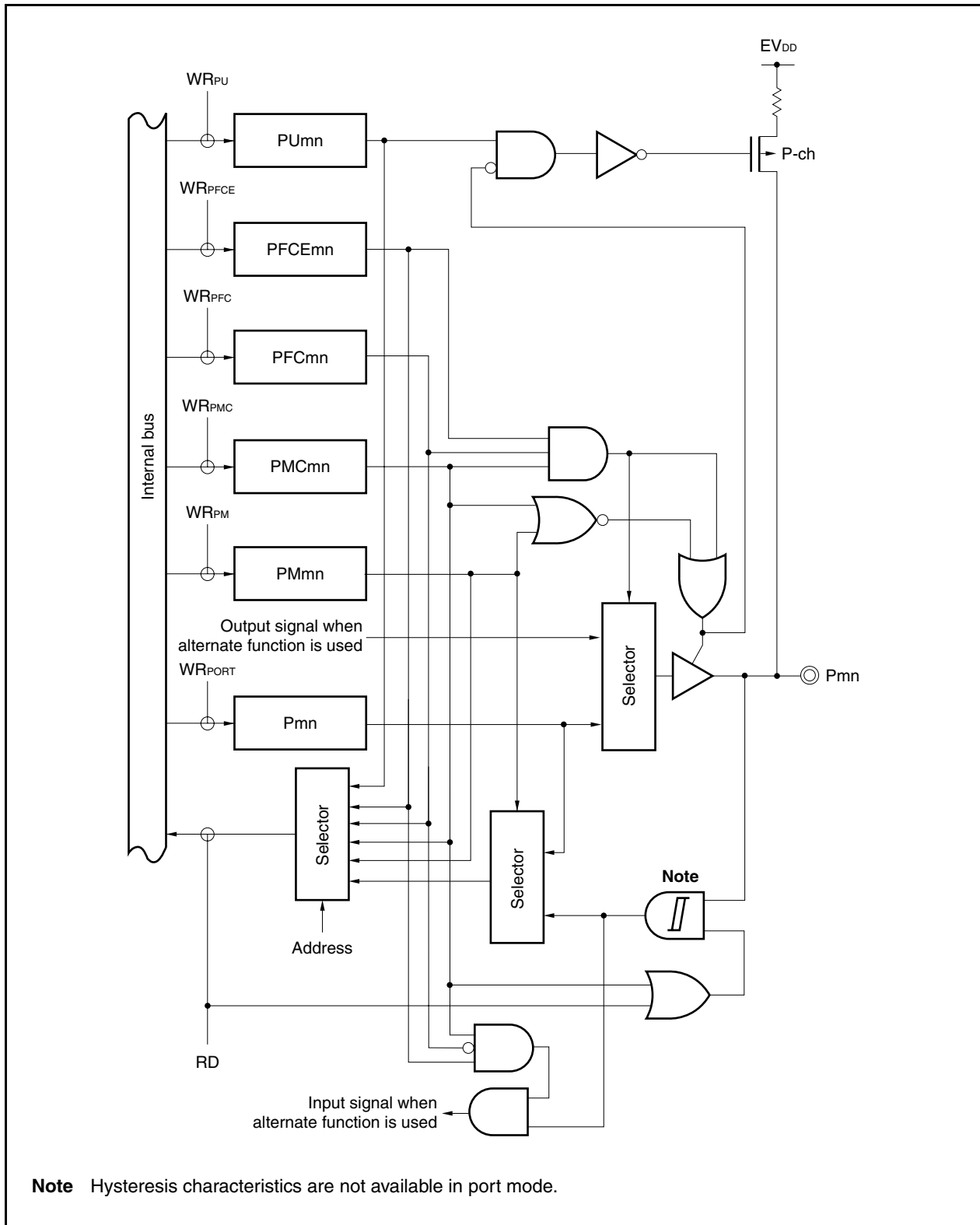


Figure 4-21. Block Diagram of Type U-8



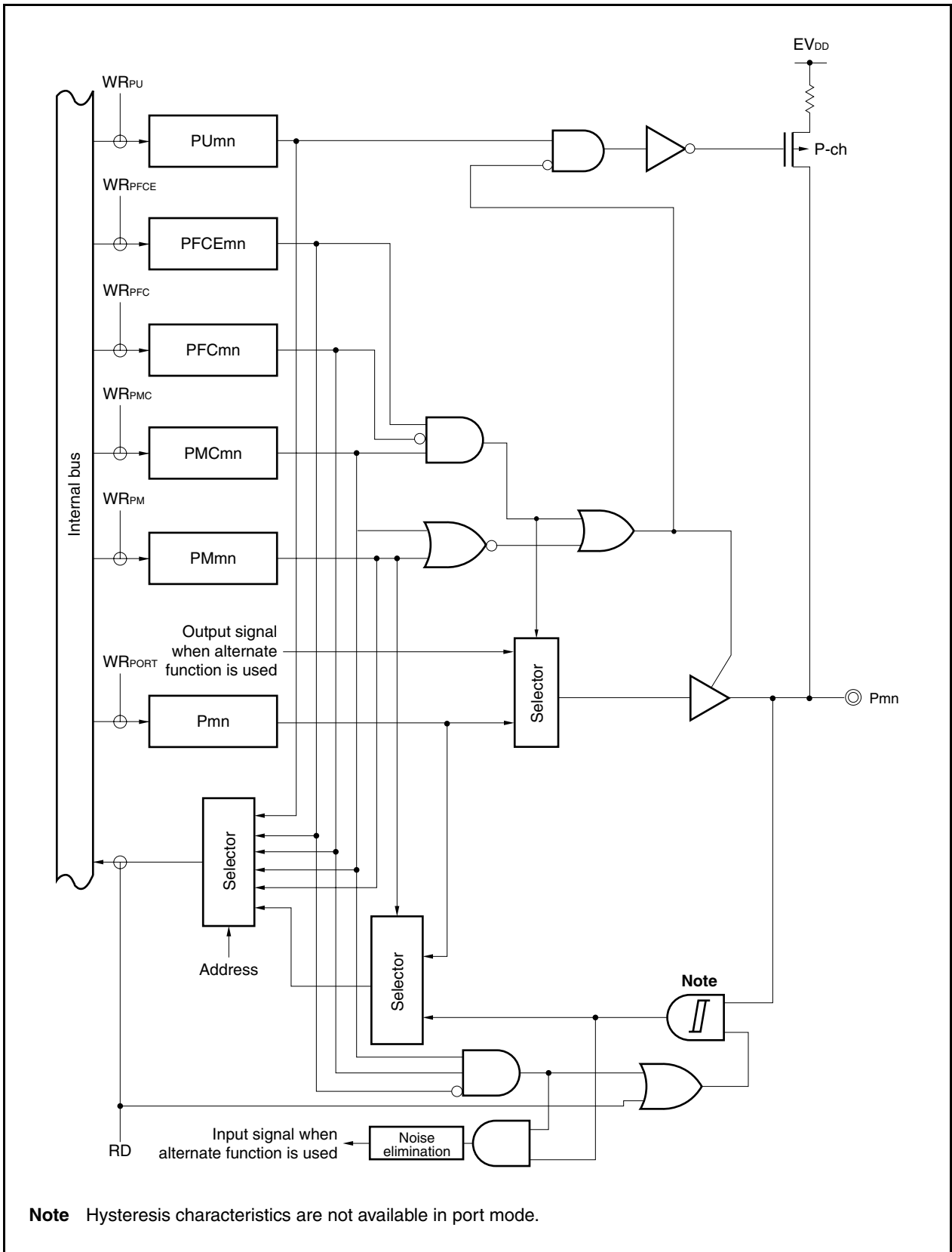
Note Hysteresis characteristics are not available in port mode.

Figure 4-22. Block Diagram of Type U-9



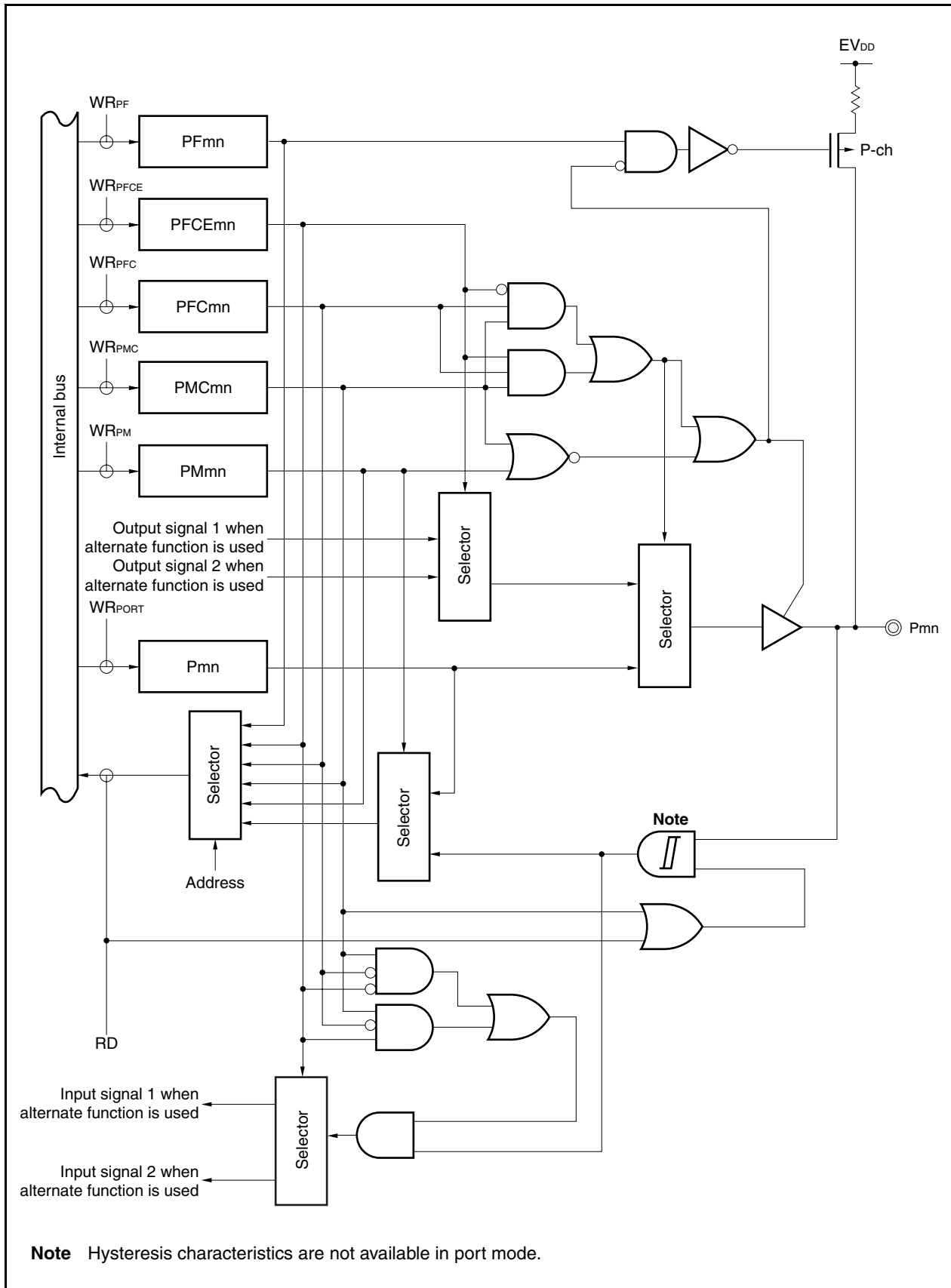
Note Hysteresis characteristics are not available in port mode.

Figure 4-23. Block Diagram of Type U-12



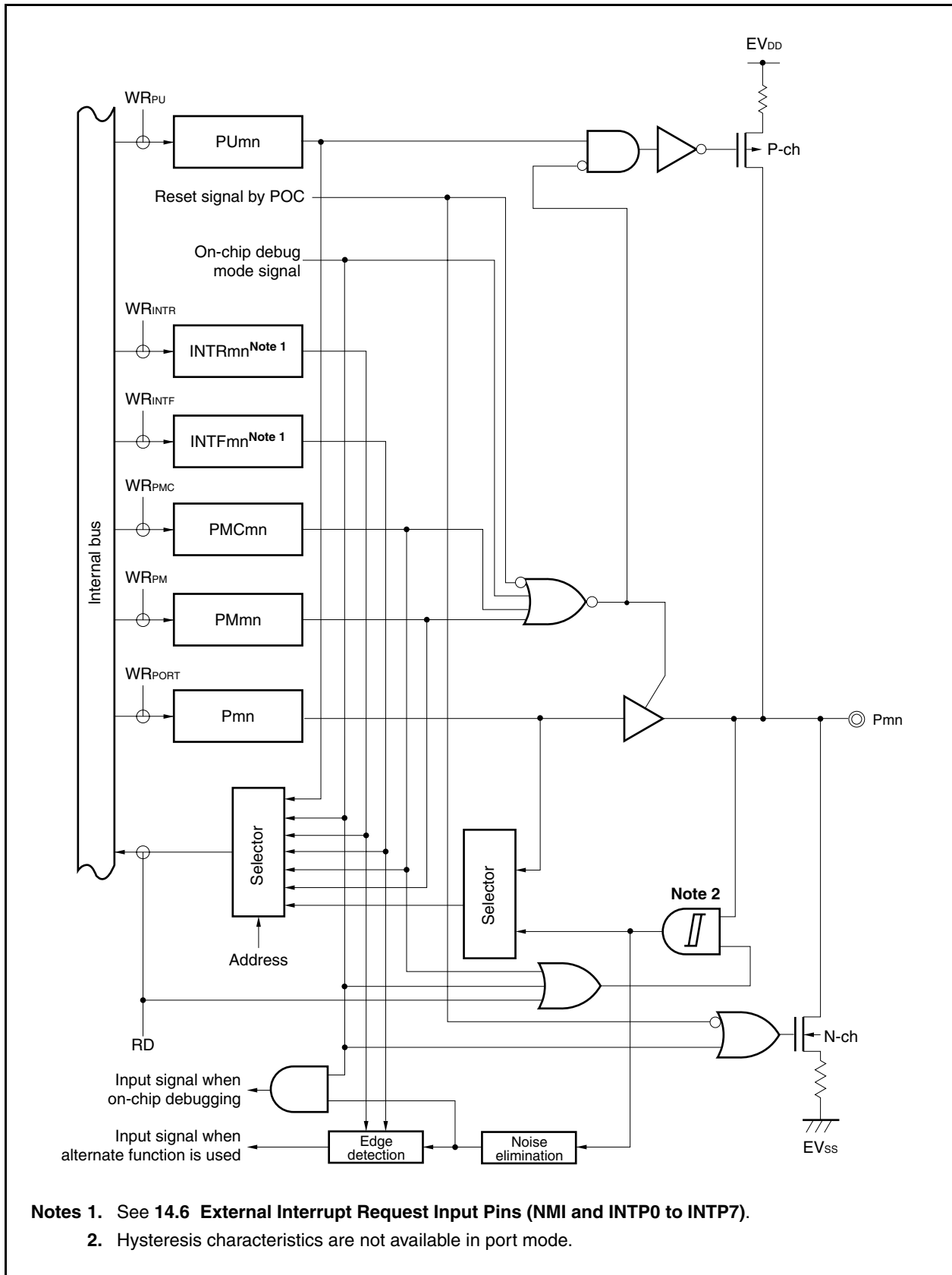
Note Hysteresis characteristics are not available in port mode.

Figure 4-24. Block Diagram of Type U-13



Note Hysteresis characteristics are not available in port mode.

Figure 4-26. Block Diagram of Type AA-1



4.5 Cautions

4.5.1 Cautions on setting port pins

(1) In the V850ES/HF2, the general-purpose port function and several peripheral function I/O pin share a pin. To switch between the general-purpose port (port mode) and the peripheral function I/O pin (alternate-function mode), set by the PMCn register. In regards to this register setting sequence, note with caution the following.

(a) Cautions on switching from port mode to alternate-function mode

To switch from the port mode to alternate-function mode in the following order.

<1> Set the PFCn and PFCEn registers: Alternate-function selection

<2> Set the corresponding bit of the PMCn register to 1: Switch to alternate-function mode

If the PMCn register is set first, note with caution that, at that moment or depending on the change of the pin states in accordance with the setting of the PFCn and PFCEn registers, unexpected operations may occur.

Caution Regardless of the port mode/alternate-function mode, the Pn register is read and written as follows.

- Pn register read: Read the port output latch value (when PMn.PMnm bit = 0), or read the pin states (PMn.PMnm bit = 1).
- Pn register write: Write to the port output latch

<R>

(b) Cautions on alternate-function mode (input)

The input signal to the alternate-function block is low level when the PMCn.PMCnm bit is 0 due to the AND output of the PMCn register set value and the pin level. Thus, depending on the port setting and alternate-function operation enable timing, unexpected operations may occur. Therefore, switch between the port mode and alternate-function mode in the following sequence.

- To switch from port mode to alternate-function mode (input)
Set the pins to the alternate-function mode using the PMCn register and then enable the alternate-function operation.
- To switch from alternate-function mode (input) to port mode
Stop the alternate-function operation and then switch the pins to the port mode.

CHAPTER 5 CLOCK GENERATION FUNCTION

5.1 Overview

The following clock generation functions are available.

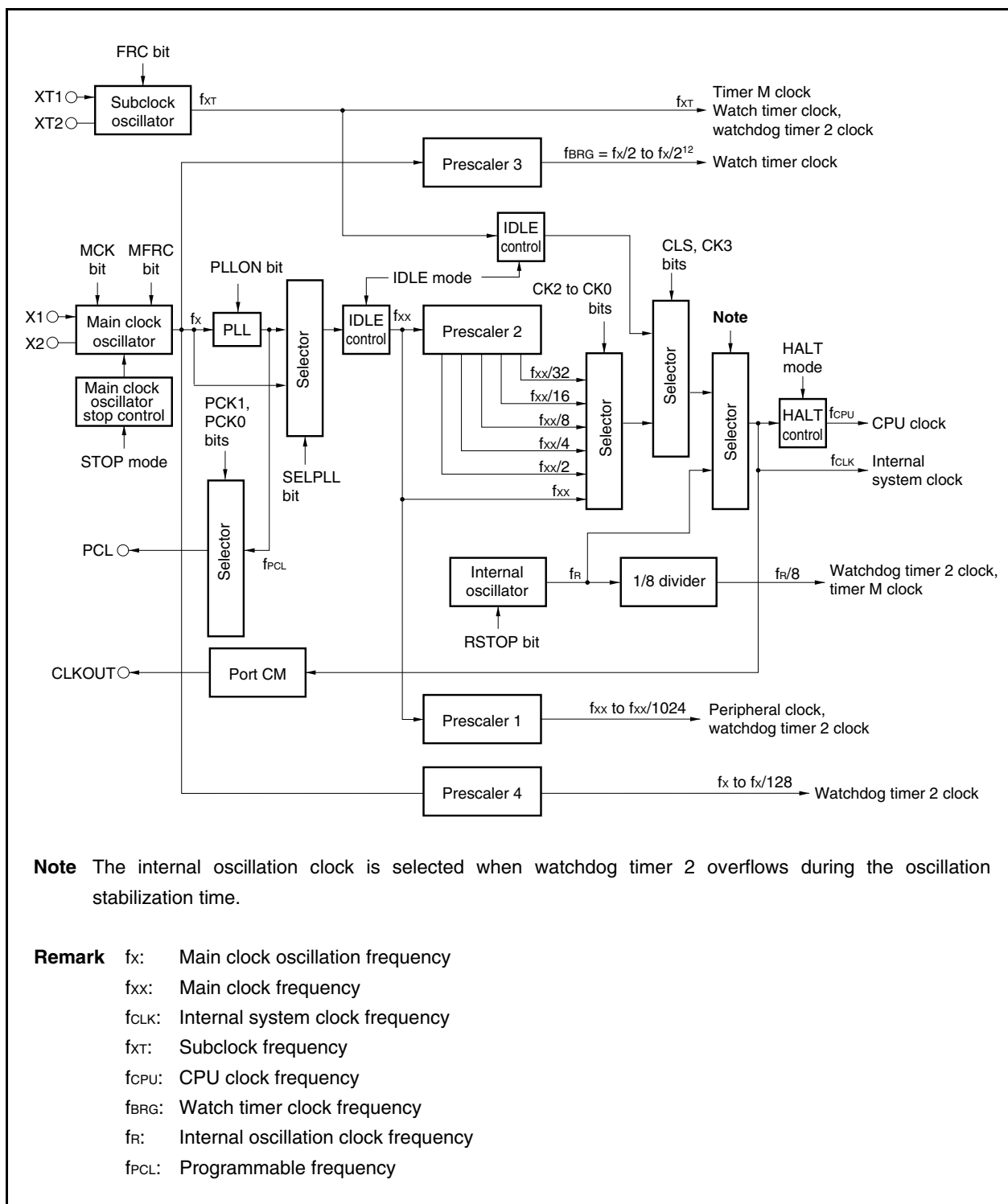
- Main clock oscillator
 - In clock-through mode
 $f_x = 4$ to 5 MHz ($f_{xx} = 4$ to 5 MHz)
 - In PLL mode
 $f_x = 4$ to 5 MHz ($f_{xx} = 16$ to 20 MHz)
- Subclock oscillator (crystal oscillation or RC oscillation selectable by option byte function)
 - $f_{XT} = 32.768$ kHz (crystal resonator)
 - $f_{XT} = 20$ kHz (RC oscillator)
- Multiply ($\times 4$) function by PLL (Phase Locked Loop)
 - Clock-through mode/PLL mode selectable
- Internal oscillator
 - $f_R = 200$ kHz (TYP.)
- 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
- Programmable clock (PCL) output function

Remark f_x : Main clock oscillation frequency
 f_{xx} : Main clock frequency
 f_R : Internal oscillation clock frequency
 f_{XT} : Subclock frequency

5.2 Configuration

<R>

Figure 5-1. Clock Generator



(1) 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 ($f_{xx} = 16$ to 20 MHz)

(2) Subclock oscillator

The sub-resonator oscillates a frequency (f_{XT}) of 32.768 kHz or 20 kHz.

(3) Main clock oscillator stop control

This circuit generates a control signal that stops oscillation of the main clock oscillator.

Oscillation of the main clock oscillator is stopped in the STOP mode or when the PCC.MCK bit = 1 (valid only when the PCC.CLS bit = 1).

(4) Internal oscillator

Oscillates a frequency (f_R) of 200 kHz (TYP.).

(5) Prescaler 1

This circuit generates the clock (f_{xx} to $f_{xx}/1,024$) to be supplied to the following on-chip peripheral functions: TMP0 to TMP3, TMQ0, TMM0, CSIB0, CSIB1, UARTA0, UARTA1, 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, and RAM blocks, and can be output from the CLKOUT pin.

(7) Prescaler 3

This circuit divides the clock generated by the main clock oscillator (f_x) to a specific frequency (32.768 kHz) and supplies that clock to the watch timer block.

For details, see **CHAPTER 9 WATCH TIMER FUNCTIONS**.

(8) Prescaler 4

<R> This circuit generates the clock (f_x to $f_x/128$) to be supplied to on-chip peripheral function.

The block to be supplied is WDT2 only.

(9) PLL

This circuit multiplies the clock generated by the main clock oscillator (f_x) 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.

5.3 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 (see **3.4.7 Special registers**).

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

Reset sets this register to 03H.

After reset: 03H R/W Address: FFFF828H

| | | | | | | | | |
|-----|-----|-----|------|---------------------|-----|-----|-----|-----|
| | 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 |
| <ul style="list-style-type: none"> • Even if the MCK bit is set (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. • Before setting the MCK bit from 0 to 1, stop the on-chip peripheral functions operating with the main clock. • When the main clock is stopped and the device is operating with the subclock, clear (0) the MCK bit and secure the oscillation stabilization time by software before switching the CPU clock to the main clock or operating the on-chip peripheral functions. | |

| | |
|------|---|
| 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. Also stop the operations of the on-chip peripheral functions operating with the main clock.

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.

$$\text{Internal system clock (f}_{CLK}\text{)} > \text{Subclock (f}_{XT}\text{)} \times 4$$

Remark Internal system clock (f_{CLK}): Clock generated from the main clock (f_{XX}) by setting bits CK2 to CK0

[Description example]

```
<1> _SET_SUB_RUN :
    st.b      r0, PRCMD[r0]
    setl     3, PCC[r0]          -- CK3 bit ← 1

<2> _CHECK_CLS :
    tstl     4, PCC[r0]          -- Wait until subclock operation starts.
    bz      _CHECK_CLS

<3> _STOP_MAIN_CLOCK :
    st.b      r0, PRCMD[r0]
    setl     6, PCC[r0]          -- MCK bit ← 1, main clock is stopped
```

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

(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.

Caution Enable operation of the on-chip peripheral functions operating with the main clock only after the oscillation of the main clock stabilizes. If their operations are enabled before the lapse of the oscillation stabilization time, a malfunction may occur.

[Description example]

```

<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
    mp       r0, r11
    bne       _PROGRAM_WAIT
<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
  
```

Remark The above description is an example. Note with caution that the CLS bit is read in a closed loop in <4>.

(2) Internal oscillation mode register (RCM)

The RCM register is an 8-bit register that sets the operation mode of the internal oscillator.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF80CH

| | | | | | | | | |
|-----|---|---|---|---|---|---|---|-------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RCM | 0 | 0 | 0 | 0 | 0 | 0 | 0 | RSTOP |

| | |
|-------|---|
| RSTOP | Oscillation/stop of internal oscillator |
| 0 | Internal oscillator oscillation |
| 1 | Internal oscillator stopped |

Cautions 1. The settings of the RCM register are valid by setting the option byte.

For details, see CHAPTER 23 OPTION BYTE FUNCTION.

2. The internal oscillator cannot be stopped while the CPU is operating on the internal oscillation clock (CCLS.CCLSF bit = 1). Do not set the RSTOP bit to 1.
3. The internal oscillator oscillates if the CCLS.CCLSF bit is set to 1 (when WDT overflow occurs during oscillation stabilization) even when the RSTOP bit is set to 1. At this time, the RSTOP bit remains being set to 1.

(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 sets this register to 00H.

After reset: 00H^{Note} 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 internal oscillation clock (f _R). |

Note If WDT overflow occurs during oscillation stabilization after a reset is released, the CCLSF bit is set to 1 and the reset value is 01H.

5.4 Operation

5.4.1 Operation of each clock

The following table shows the operation status of each clock.

Table 5-1. Operation Status of Each Clock

| Register Setting and Operation Status Target Clock | PCC Register | | | | | | | | |
|--|--------------------------|--|--------------|-------------------------|--------------|-----------------------------|------------------|-----------------------------|------------------|
| | CLK 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 clock oscillator (fx) | × | ○ | ○ | ○ | × | ○ | ○ | × | × |
| <R> Main system clock (fxx) | × | × | ○ | × | × | ○ | × | × | × |
| Subclock oscillator (fxT) | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| CPU clock (fCPU) | × | × | × | × | × | ○ | × | ○ | × |
| Internal system clock (fCLK) | × | × | ○ | × | × | ○ | × | ○ | × |
| Main clock (in PLL mode, fxx) | × | Note 1 | ○ | Note 2 | × | ○ | ○ | × | × |
| Peripheral clock (fxx to fxx/1,024) | × | × | ○ | × | × | ○ | × | × | × |
| WT clock (main) | × | × | ○ | ○ | × | ○ | ○ | × | × |
| WT clock (sub) | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| WDT2 clock (internal oscillation) | × | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| WDT2 clock (main) | × | × | ○ | × | × | ○ | × | × | × |

Notes 1. Oscillation starts after time 1/2 of the oscillation stabilization time, and the stable clock is supplied after lockup time.

2. Operable in the IDLE1 mode. Stopped in the IDLE2 mode.

Remark ○: Operable

×: Stopped

5.4.2 Clock output function

The clock output function is used to output the internal system clock (fCLK) from the CLKOUT pin.

The internal system clock (fCLK) is selected by using the PCC.CK3 to PCC.CK0 bits.

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 5-1 and the pin can output the clock when it is in the operable status. It outputs a low level in the stopped status. However, the CLKOUT pin is in the port mode (PCM1 pin: input mode) after reset and until it is set in the output mode. Therefore, the status of the pin is Hi-Z.

5.5 PLL Function

5.5.1 Overview

In the V850ES/HF2, an operating clock that is 4 times higher than the oscillation frequency output by the PLL function or the clock-through mode can be selected as the operating clock of the CPU and on-chip peripheral functions.

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)

5.5.2 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 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 mode | | | | | | | |
| 1 | PLL mode | | | | | | | |

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 state 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 sets this register to 00H.

After reset: 00H R Address: FFFF824H

| | | | | | | | | |
|-------|---|---|---|---|---|---|---|------|
| | 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 lock status of the PLL in real time. The set/clear conditions are as follows.

[Set conditions]

- Upon system reset^{Note}
- In IDLE2 or STOP mode
- Upon setting of PLL stop (clearing of PLLCTL.PLLON bit to 0)
- Upon stopping main clock and using CPU with subclock (setting of PCC.CK3 bit to 1 and setting of 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.

[Clear conditions]

- Upon overflow of oscillation stabilization time following reset release (OSTS register default time (see **16.2 (3) Oscillation stabilization time select register (OSTS)**))
- Upon oscillation stabilization timer overflow (time set by 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 PLLS register) when the PLLCTL.PLLON bit is changed from 0 to 1
- After the setup time inserted upon release of the IDLE2 mode is released (time set by the OSTS register) when the IDLE2 mode is set during PLL operation.

(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 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 | Selection of PLL lockup time |
|-------|-------|------------------------------|
| 0 | 0 | $2^{10}/f_x$ |
| 0 | 1 | $2^{11}f_x$ |
| 1 | 0 | $2^{12}/f_x$ |
| 1 | 1 | $2^{13}/f_x$ (default value) |

- Cautions**
1. Set so that the lockup time is 800 μ s or longer.
 2. Do not change the PLLS register setting during the lockup period.

Remark fxx: Main clock oscillation frequency

(4) Programmable clock mode register (PCLM)

The PCLM register is an 8-bit register used to control the PCL output.

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

| | | | | | | | | |
|------------------|---|-----|--------------------|------|---|---|------|------|
| After reset: 00H | | R/W | Address: FFFFF82FH | | | | | |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PCLM | 0 | 0 | 0 | PCLE | 0 | 0 | PCK1 | PCK0 |

| | |
|------|---|
| PCLE | Selection of PCL pin output operation |
| 0 | PCL pin output disabled (PCL pin is fixed to low level) |
| 1 | PCL pin output enabled |

Caution Set the port-related control registers (PM, PMC, PFC, and PFCE registers, etc.) first, and then set the PCLE bit to 1.

| | | |
|------|------|-------------------------------|
| PCK1 | PCK0 | Selection of PLL output clock |
| 0 | 0 | $f_{PCL}/2$ |
| 0 | 1 | $f_{PCL}/4$ |
| 1 | 0 | $f_{PCL}/8$ |
| 1 | 1 | $f_{PCL}/16$ |

Caution Set the PCLE bit to 1 only during PLL operation. To stop the PLL, clear the PCLE bit to 0.

Remark f_{PCL} : Programmable frequency

<R>

5.5.3 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 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.
 - (a) When transiting to IDLE2 or STOP mode from the clock through mode
 - STOP mode: Set the OSTS register so that the oscillation stabilization time is 1 ms (min.) or longer.
 - IDLE2 mode: Set the OSTS register so that the setup time is 350 μ s (min.) or longer.
 - (b) When shifting to the IDLE 2 or STOP mode while remaining in the PLL operation mode
 - STOP mode: Set the OSTS register so that the oscillation stabilization time is 1 ms (min.) or longer.
 - IDLE2 mode: Set the OSTS register so that the setup time is 800 μ s (min.) or longer.

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 6 16-BIT TIMER/EVENT COUNTER P (TMP)

Timer P (TMP) is a 16-bit timer/event counter.

The V850ES/HF2 has four timer/event counter channels, TMP0 to TMP3.

6.1 Overview

An outline of TMP_n 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

6.2 Functions

TMP_n 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

6.3 Configuration

TMPn includes the following hardware.

Table 6-1. Configuration of TMPn

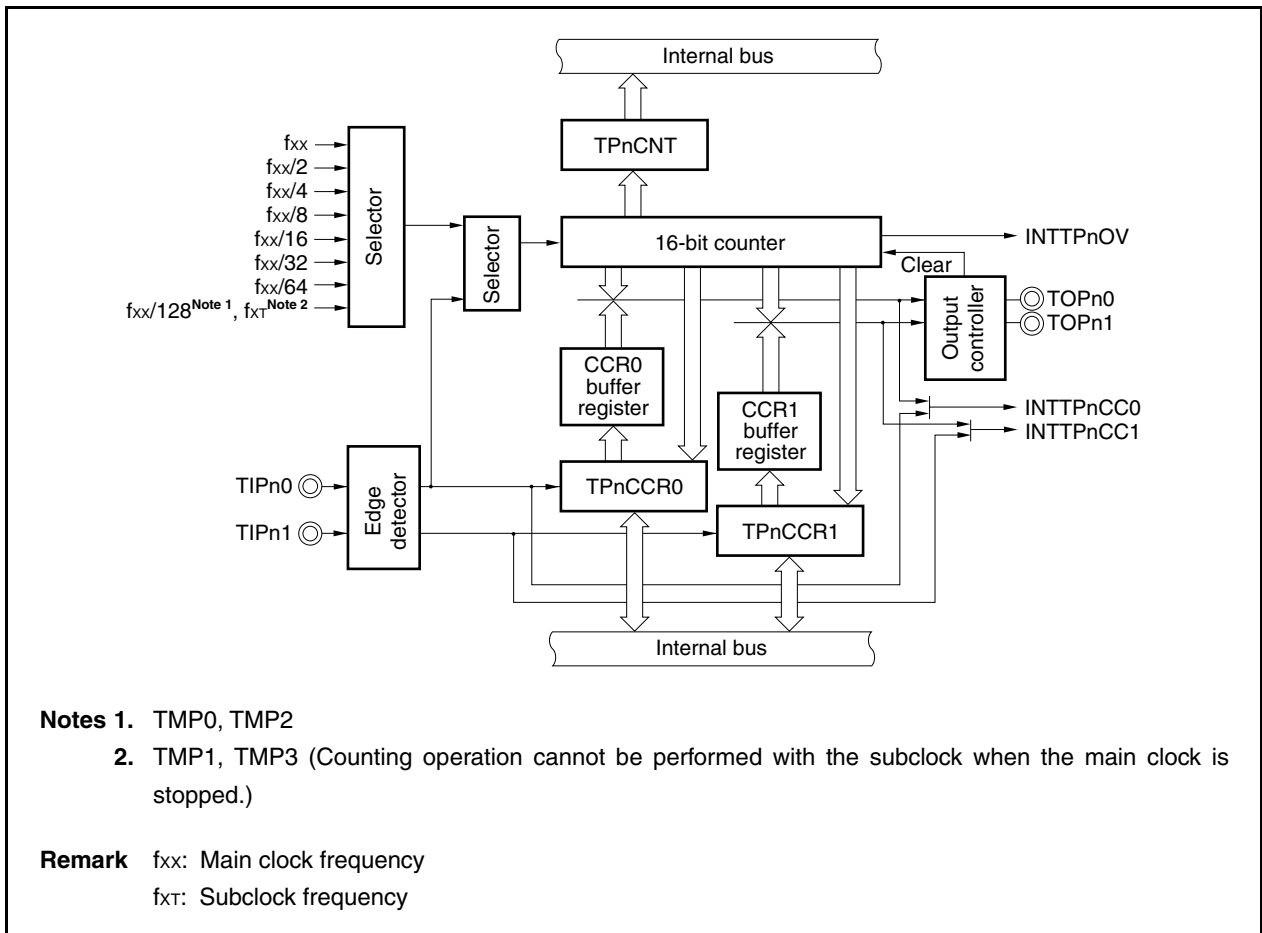
| Item | Configuration |
|-------------------------------------|--|
| Timer register | 16-bit counter |
| Registers | TMPn capture/compare registers 0, 1 (TPnCCR0, TPnCCR1) TMPn counter read buffer register (TPnCNT) CCR0, CCR1 buffer registers |
| Timer inputs | 2 (TIPn0 ^{Note 1} , TIPn1 pins) |
| Timer outputs | 2 (TOPn0, TOPn1 pins) |
| Control registers ^{Note 2} | TMPn control registers 0, 1 (TPnCTL0, TPnCTL1) TMPn I/O control registers 0 to 2 (TPnIOC0 to TPnIOC2) TMPn option register 0 (TPnOPT0) |

Notes 1. The TIPn0 pin functions alternately as a capture trigger input signal, external event count input signal, and external trigger input signal.

2. When using the functions of the TIPn0, TIPn1, TOPn0, and TOPn1 pins, see **Table 4-14 Using Port Pin as Alternate-Function Pin.**

Remark n = 0 to 3

Figure 6-1. Block Diagram of TMPn



Notes 1. TMP0, TMP2

2. TMP1, TMP3 (Counting operation cannot be performed with the subclock when the main clock is stopped.)

Remark fxx: Main clock frequency
fxt: Subclock frequency

(1) 16-bit counter

This 16-bit counter can count internal clocks or external events.

The count value of this counter can be read by using the TPnCNT register.

When the TPnCTL0.TPnCE bit = 0, the value of the 16-bit counter is FFFFH. If the TPnCNT register is read at this time, 0000H is read.

Reset sets the TPnCE bit to 0. Therefore, the 16-bit counter is set to FFFFH.

(2) CCR0 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TPnCCR0 register is used as a compare register, the value written to the TPnCCR0 register is transferred to the CCR0 buffer register. When the count value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTPnCC0) is generated.

The CCR0 buffer register cannot be read or written directly.

The CCR0 buffer register is cleared to 0000H after reset, as the TPnCCR0 register is cleared to 0000H.

(3) CCR1 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TPnCCR1 register is used as a compare register, the value written to the TPnCCR1 register is transferred to the CCR1 buffer register. When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTPnCC1) is generated.

The CCR1 buffer register cannot be read or written directly.

The CCR1 buffer register is cleared to 0000H after reset, as the TPnCCR1 register is cleared to 0000H.

(4) Edge detector

This circuit detects the valid edges input to the 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.

6.4 Registers

The registers that control TMPn are as follows.

- TMPn control register 0 (TPnCTL0)
- TMPn control register 1 (TPnCTL1)
- TMPn I/O control register 0 (TPnIOC0)
- TMPn I/O control register 1 (TPnIOC1)
- TMPn I/O control register 2 (TPnIOC2)
- TMPn option register 0 (TPnOPT0)
- TMPn capture/compare register 0 (TPnCCR0)
- TMPn capture/compare register 1 (TPnCCR1)
- TMPn counter read buffer register (TPnCNT)

Remarks 1. When using the functions of the TIPn0, TIPn1, TOPn0, and TOPn1 pins, see **Table 4-14 Using Port Pin as Alternate-Function Pin.**

2. n = 0 to 3

(1) TMPn control register 0 (TPnCTL0)

The TPnCTL0 register is an 8-bit register that controls the operation of TMPn.

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

Reset sets this register to 00H.

The same value can always be written to the TPnCTL0 register by software.

After reset: 00H R/W Address: TP0CTL0 FFFFF590H, TP1CTL0 FFFFF5A0H,
TP2CTL0 FFFFF5B0H, TP3CTL0 FFFFF5C0H

| | | | | | | | | |
|---------|-------|---|---|---|---|---------|---------|---------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TPnCTL0 | TPnCE | 0 | 0 | 0 | 0 | TPnCKS2 | TPnCKS1 | TPnCKS0 |

(n = 0 to 3)

| | |
|-------|---|
| TPnCE | TMPn operation control |
| 0 | TMPn operation disabled (TMPn reset asynchronously ^{Note 1}). |
| 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 2} |

Notes 1. TPn0PT0.TPnOVF bit, 16-bit counter, timer output (TOPn0, TOPn1 pins)

2. Counting operation cannot be performed with the subclock when the main clock is stopped.

Cautions 1. Set the TPnCKS2 to TPnCKS0 bits when the TPnCE bit = 0.

When the value of the TPnCE bit is changed from 0 to 1, the TPnCKS2 to TPnCKS0 bits can be set simultaneously.

2. Be sure to clear bits 3 to 6 to "0".

Remark f_{xx}: Main clock frequency

f_{xt}: Subclock frequency

(2) TMPn control register 1 (TPnCTL1)

The TPnCTL1 register is an 8-bit register that controls the operation of TMPn.

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

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: TP0CTL1 FFFFF591H, TP1CTL1 FFFFF5A1H,
TP2CTL1 FFFFF5B1H, TP3CTL1 FFFFF5C1H

| | | | | | | | | |
|-------------------------|--------|--------|--------|---|---|--------|--------|--------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TPnCTL1 (n = 0 to 3) | TPnSYE | TPnEST | TPnEEE | 0 | 0 | TPnMD2 | TPnMD1 | TPnMD0 |

| | | | | | | | | | | | |
|--------------|--|------|--------------|-------------|--|------|------|---|------|------|------|
| TPnSYE | Tuned operation mode enable control | | | | | | | | | | |
| 0 | Independent operation mode (asynchronous operation mode) | | | | | | | | | | |
| 1 | Tuned operation mode (specification of slave operation) In this mode, timer P can operate in synchronization with a master timer. | | | | | | | | | | |
| | <table border="1"> <tr> <td>Master timer</td> <td colspan="2">Slave timer</td> </tr> <tr> <td>TMP0</td> <td>TMP1</td> <td>–</td> </tr> <tr> <td>TMP2</td> <td>TMP3</td> <td>TMQ0</td> </tr> </table> | | Master timer | Slave timer | | TMP0 | TMP1 | – | TMP2 | TMP3 | TMQ0 |
| Master timer | Slave timer | | | | | | | | | | |
| TMP0 | TMP1 | – | | | | | | | | | |
| TMP2 | TMP3 | TMQ0 | | | | | | | | | |
| | For the tuned operation mode, see 6.6 Timer Tuned Operation Function . | | | | | | | | | | |
| | Caution Be sure to clear the TP0SYE and TP2SYE bits to 0. | | | | | | | | | | |

| | |
|--------|--|
| TPnEST | Software trigger control |
| 0 | – |
| 1 | Generate a valid signal for external trigger input. <ul style="list-style-type: none"> • 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. |

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. Be sure to clear bits 3 and 4 to “0”.

| 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**
- External event count input is selected in the external event count mode regardless of the value of the TPnEEE bit.
 - 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.

<R>

(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 sets 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 (n = 0 to 3) | 0 | 0 | 0 | 0 | TPnOL1 | TPnOE1 | TPnOL0 | TPnOE0 |

| | |
|--------|--|
| TPnOL1 | TOPn1 pin output level setting ^{Note} |
| 0 | TOPn1 pin output starts at high level |
| 1 | TOPn1 pin output starts at low level |

| | |
|--------|--|
| 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). |

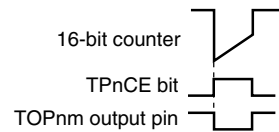
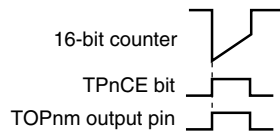
| | |
|--------|--|
| TPnOL0 | TOPn0 pin output level setting ^{Note} |
| 0 | TOPn0 pin output starts at high level |
| 1 | TOPn0 pin output starts at low level |

| | |
|--------|--|
| 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). |

Note The output level of the timer output pin (TOPnm) specified by the TPnOLm bit is shown below.

• When TPnOLm bit = 0

• When TPnOLm bit = 1



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.

Remark n = 0 to 3, 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 sets 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 (n = 0 to 3) | 0 | 0 | 0 | 0 | TPnIS3 | TPnIS2 | TPnIS1 | TPnIS0 |

| 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 sets 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.
 3. The TPnETS1 and TPnETS0 bits are valid only when the external trigger pulse output mode (TPnCTL1.TPnMD2 to TPnCTL1.TPnMD0 bits = 010) or the one-shot pulse output mode (TPnCTL1.TPnMD2 to TPnCTL1.TPnMD0 = 011) is set.

(6) TMPn option register 0 (TPnOPT0)

The TPnOPT0 register is an 8-bit register used to set the capture/compare operation and detect an overflow.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TP0OPT0 FFFF595H, TP1OPT0 FFFF5A5H,
TP2OPT0 FFFF5B5H, TP3OPT0 FFFF5C5H

| | | | | | | | | |
|-------------------------|---|---|---------|---------|---|---|---|--------|
| | 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 |
| <ul style="list-style-type: none"> 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 clear bits 1 to 3, 6, and 7 to “0”.**

(7) TMPn capture/compare register 0 (TPnCCR0)

The TPnCCR0 register can be used as a capture register or a compare register depending on the mode.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TPnOPT0.TPnCCS0 bit. In the pulse width measurement mode, the TPnCCR0 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TPnCCR0 register can be read or written during operation.

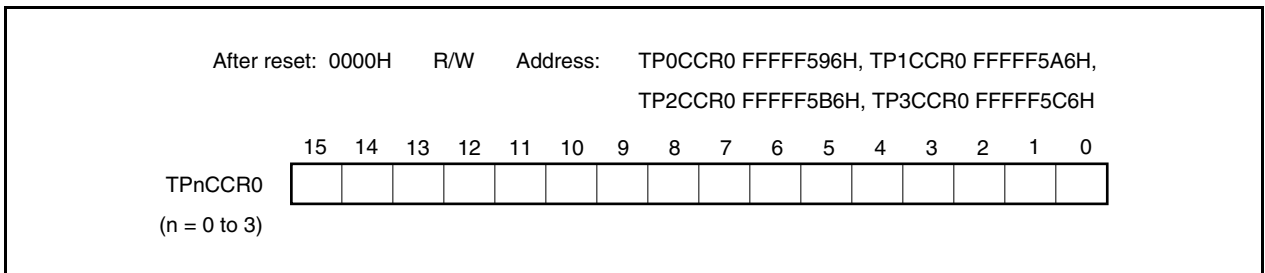
This register can be read or written in 16-bit units.

Reset sets this register to 0000H.

Caution Accessing the TPnCCR0 register is prohibited in the following statuses. For details, see 3.4.8

(2) Accessing specific on-chip peripheral I/O registers.

- When the CPU operates with the subclock and the main clock oscillation is stopped
- When the CPU operates with the internal oscillation clock



(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) is detected.

Even if the capture operation and reading the TPnCCR0 register conflict, the correct value of the TPnCCR0 register can be read.

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 6-2. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

| Operation Mode | Capture/Compare Register | How to Write Compare Register |
|-------------------------------|--------------------------|-------------------------------|
| Interval timer | Compare register | Anytime write |
| External event counter | Compare register | Anytime write |
| External trigger pulse output | Compare register | Batch write |
| One-shot pulse output | Compare register | Anytime write |
| PWM output | Compare register | Batch write |
| Free-running timer | Capture/compare register | Anytime write |
| Pulse width measurement | Capture register | – |

(8) TMPn capture/compare register 1 (TPnCCR1)

The TPnCCR1 register can be used as a capture register or a compare register depending on the mode.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TPnOPT0.TPnCCS1 bit. In the pulse width measurement mode, the TPnCCR1 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TPnCCR1 register can be read or written during operation.

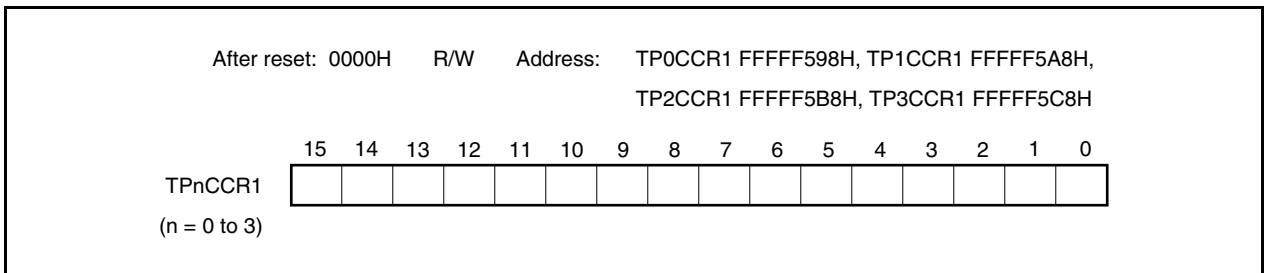
This register can be read or written in 16-bit units.

Reset sets this register to 0000H.

Caution Accessing the TPnCCR1 register is prohibited in the following statuses. For details, see 3.4.8

(2) Accessing specific on-chip peripheral I/O registers.

- When the CPU operates with the subclock and the main clock oscillation is stopped
- When the CPU operates with the internal oscillation clock



(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) is detected.

Even if the capture operation and reading the TPnCCR1 register conflict, the correct value of the TPnCCR1 register can be read.

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 6-3. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

| Operation Mode | Capture/Compare Register | How to Write Compare Register |
|-------------------------------|--------------------------|-------------------------------|
| Interval timer | Compare register | Anytime write |
| External event counter | Compare register | Anytime write |
| External trigger pulse output | Compare register | Batch write |
| One-shot pulse output | Compare register | Anytime write |
| PWM output | Compare register | Batch write |
| Free-running timer | Capture/compare register | Anytime write |
| Pulse width measurement | Capture register | – |

(9) TMPn counter read buffer register (TPnCNT)

The TPnCNT register is a read buffer register that can read the count value of the 16-bit counter.

If this register is read when the TPnCTL0.TPnCE bit = 1, the count value of the 16-bit timer can be read.

This register is read-only, in 16-bit units.

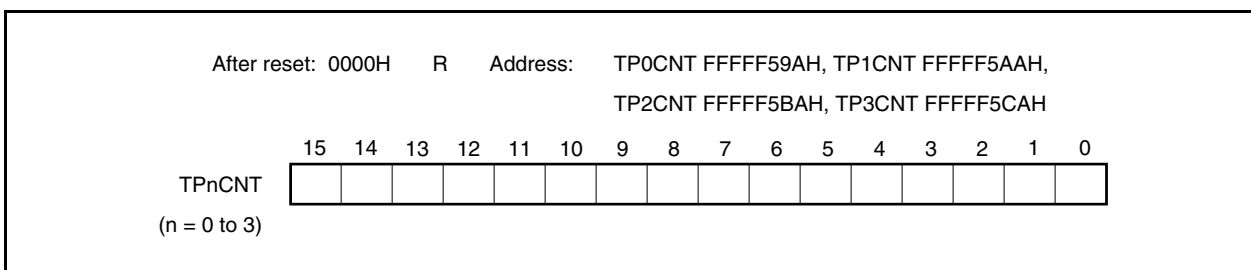
The value of the TPnCNT register is cleared to 0000H when the TPnCE bit = 0. If the TPnCNT register is read at this time, the value of the 16-bit counter (FFFFH) is not read, but 0000H is read.

The value of the TPnCNT register is cleared to 0000H after reset, as the TPnCE bit is cleared to 0.

Caution Accessing the TPnCNT register is prohibited in the following statuses. For details, see 3.4.8

(2) Accessing specific on-chip peripheral I/O registers.

- When the CPU operates with the subclock and the main clock oscillation is stopped
- When the CPU operates with the internal oscillation clock



(10) TIP_nm pin noise elimination control register (PnmNFC)

The PnmNFC register is an 8-bit register that sets the digital noise filter of the timer P input pin for noise elimination.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: P00NFC FFFFFFFB00H (TIP00 pin)
 P01NFC FFFFFFFB04H (TIP01 pin)
 P10NFC FFFFFFFB08H (TIP10 pin)
 P11NFC FFFFFFFB0CH (TIP11 pin)
 P20NFC FFFFFFFB10H (TIP20 pin)
 P21NFC FFFFFFFB14H (TIP21 pin)
 P30NFC FFFFFFFB18H (TIP30 pin)
 P31NFC FFFFFFFB1CH (TIP31 pin)

| | | | | | | | | |
|--------|---|-------|---|---|---|------|------|------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PnmNFC | 0 | NFSTS | 0 | 0 | 0 | NFC2 | NFC1 | NFC0 |

(n = 0 to 3, m = 0, 1)

| NFSTS | Setting of number of times of sampling by digital noise filter |
|-------|--|
| 0 | 3 times |
| 1 | 2 times |

| NFC2 | NFC1 | NFC0 | Sampling clock | |
|------------------|------|------|---------------------|---------------------|
| | | | 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} /16 | f _{xx} /8 |
| 1 | 0 | 0 | f _{xx} /32 | f _{xx} /16 |
| 1 | 0 | 1 | f _{xx} /64 | f _τ |
| Other than above | | | Setting prohibited | |

- Cautions**
1. Be sure to clear bits 3 to 5 and 7 to “0”.
 2. A signal input to the timer input pin (TIP_nm) before the PnmNFC register is set is output with digital noise eliminated.
 Therefore, set the sampling clock (NFC2 to NFC0) and the number of times of sampling (NFSTS) by using the PnmNFC register, wait for initialization time = (Sampling clock) × (Number of times of sampling), and enable the timer operation.

Remark The width of the noise that can be accurately eliminated is (Sampling clock) × (Number of times of sampling – 1). Even noise with a width narrower than this may cause a miscount if it is synchronized with the sampling clock.

6.5 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

6.5.1 Interval timer mode (TPnMD2 to TPnMD0 bits = 000)

In the interval timer mode, an interrupt request signal (INTTPnCC0) is generated at the specified interval if the TPnCTL0.TPnCE bit is set to 1. A square wave whose half cycle is equal to the interval can be output from the TOPn0 pin.

Usually, the TPnCCR1 register is not used in the interval timer mode.

Figure 6-2. Configuration of Interval Timer

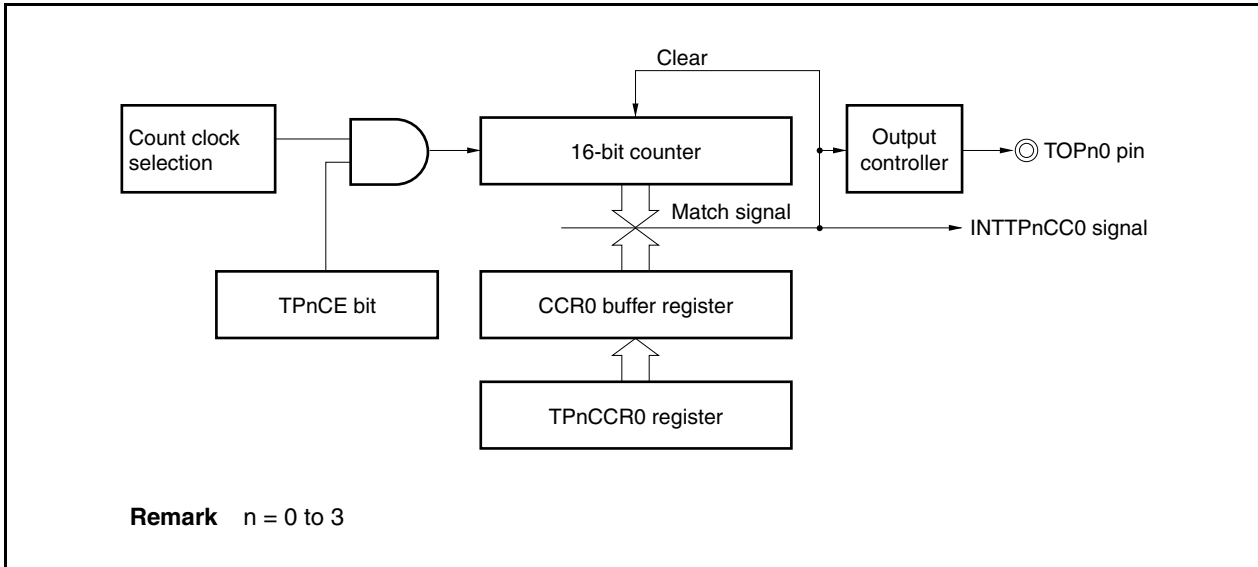
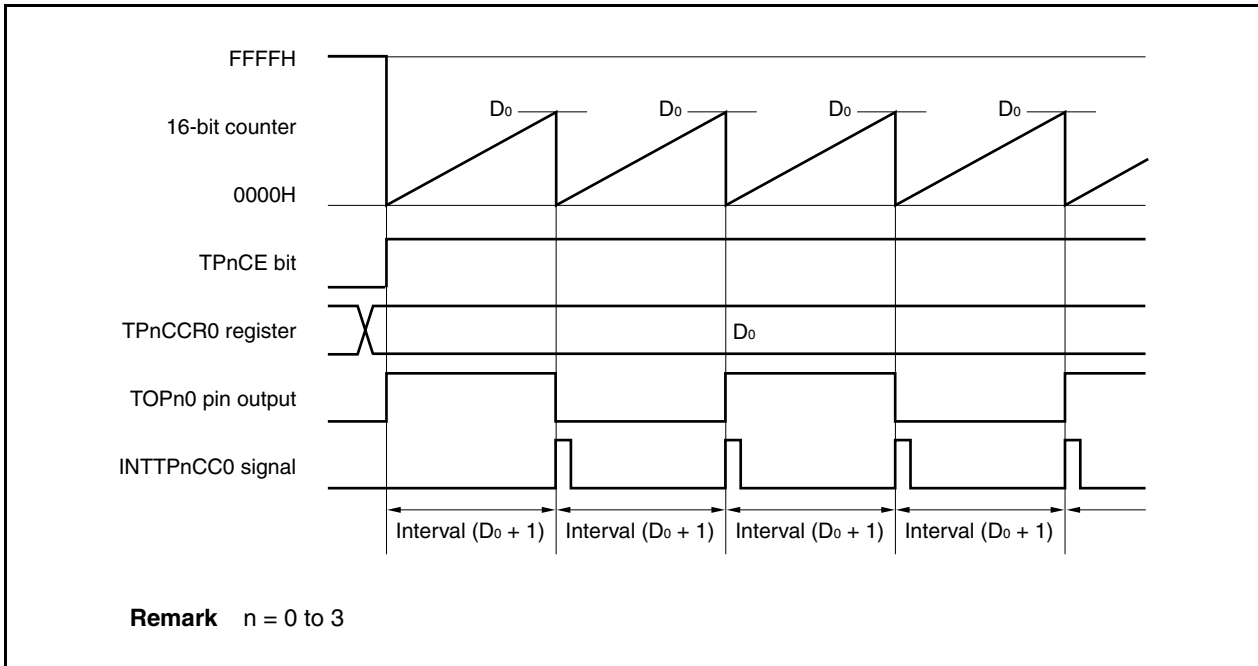


Figure 6-3. Basic Timing of Operation in Interval Timer Mode



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.

$$\text{Interval} = (\text{Set value of TPnCCR0 register} + 1) \times \text{Count clock cycle}$$

Remark n = 0 to 3

Figure 6-4. Register Setting for Interval Timer Mode Operation (1/2)

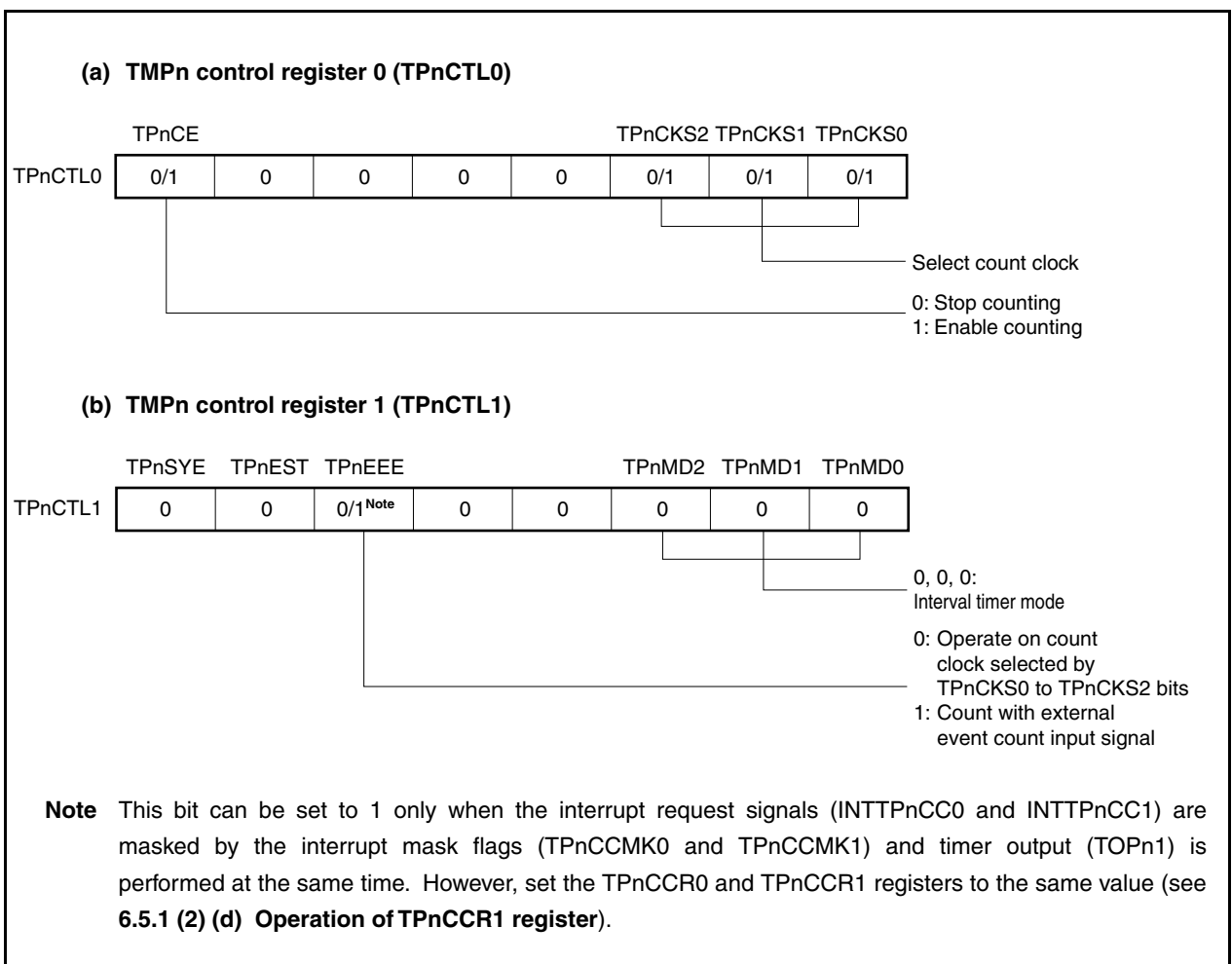
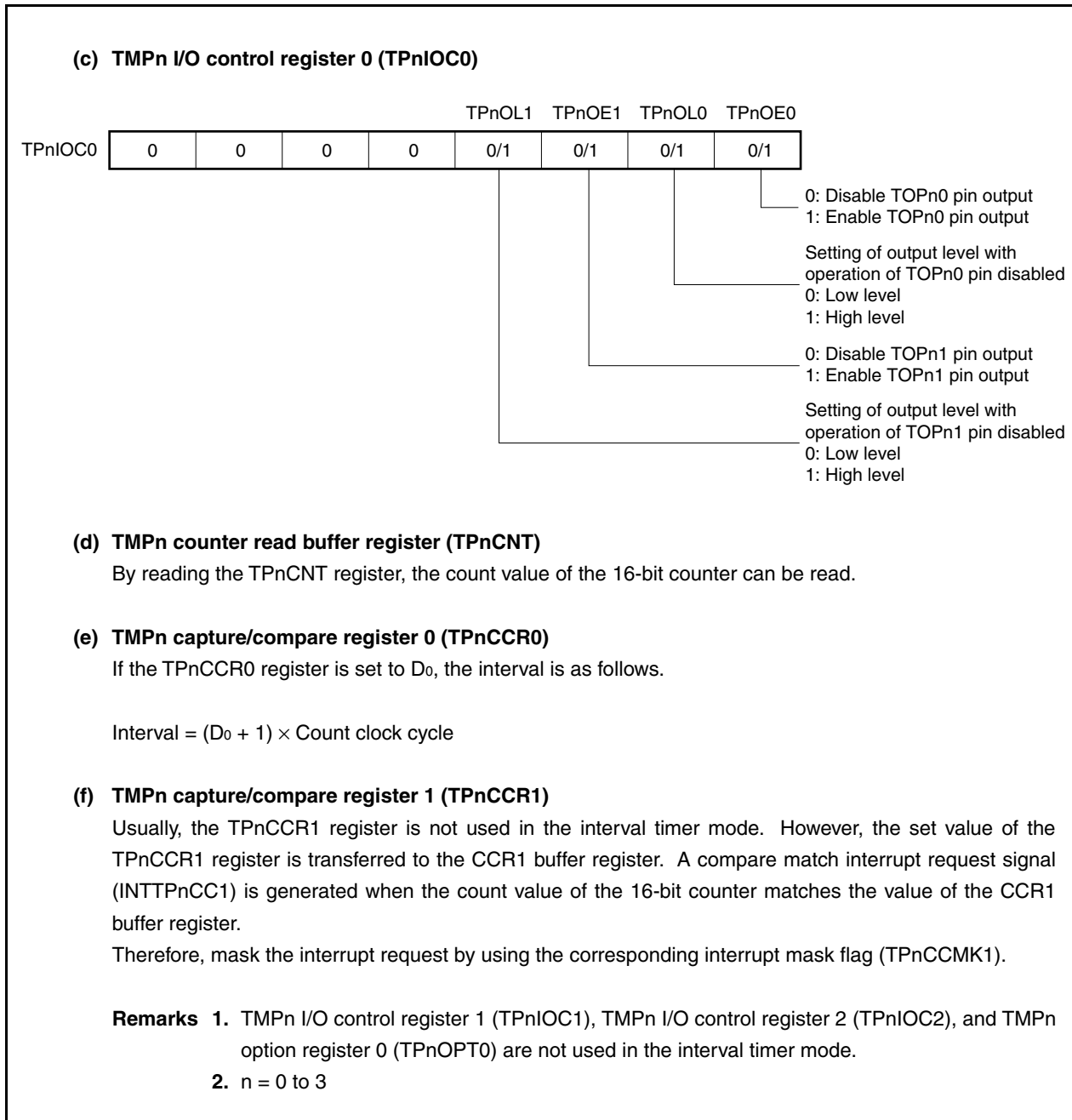
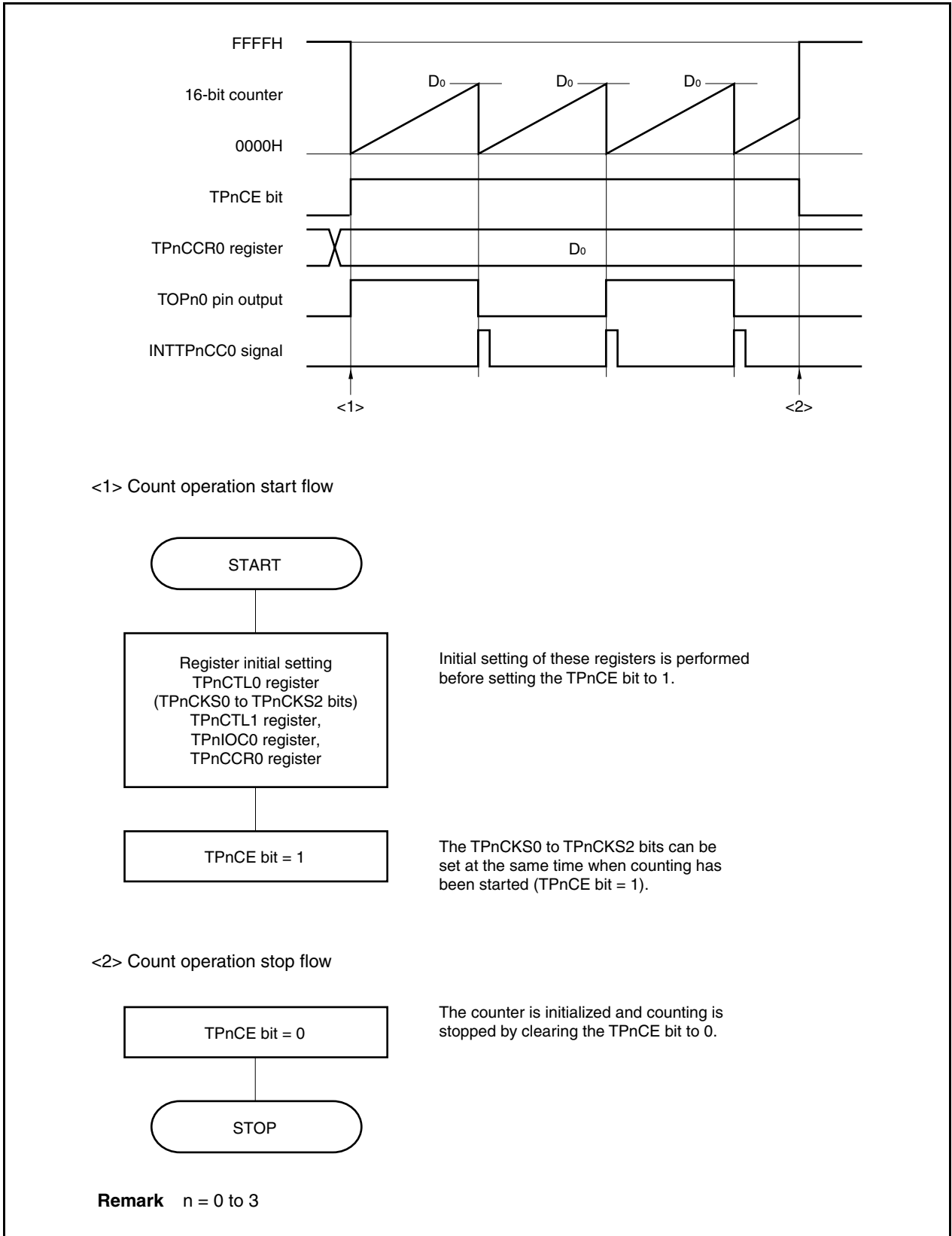


Figure 6-4. Register Setting for Interval Timer Mode Operation (2/2)



(1) Interval timer mode operation flow

Figure 6-5. Software Processing Flow in Interval Timer Mode

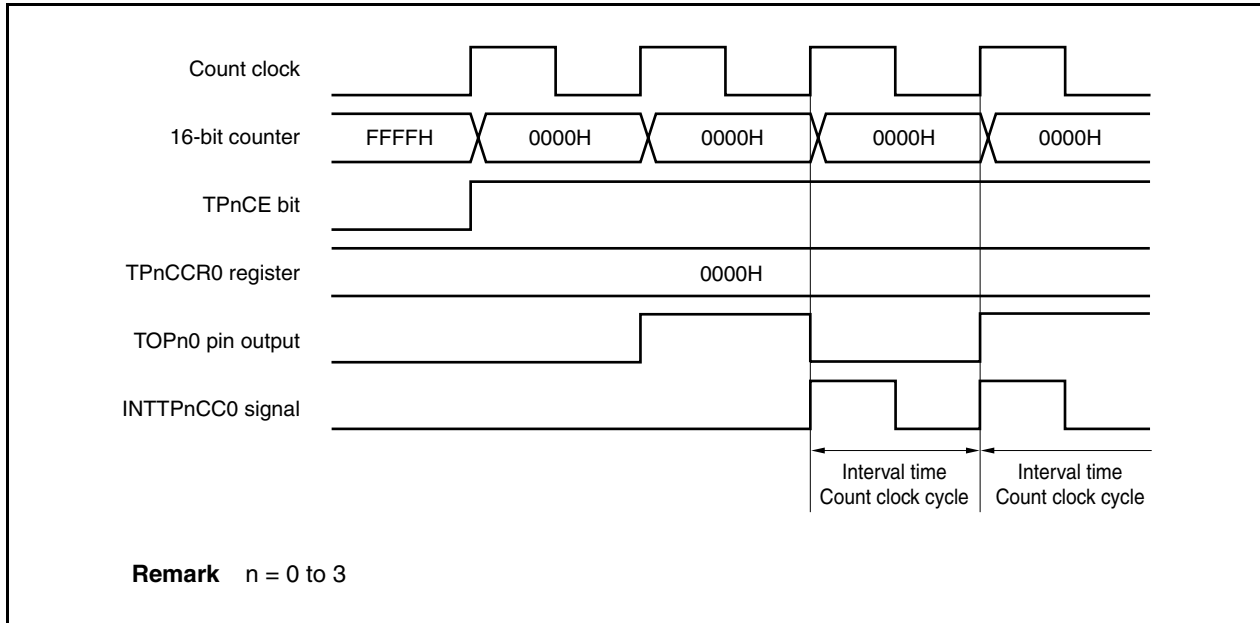


(2) Interval timer mode operation timing

(a) Operation if TPnCCR0 register is set to 0000H

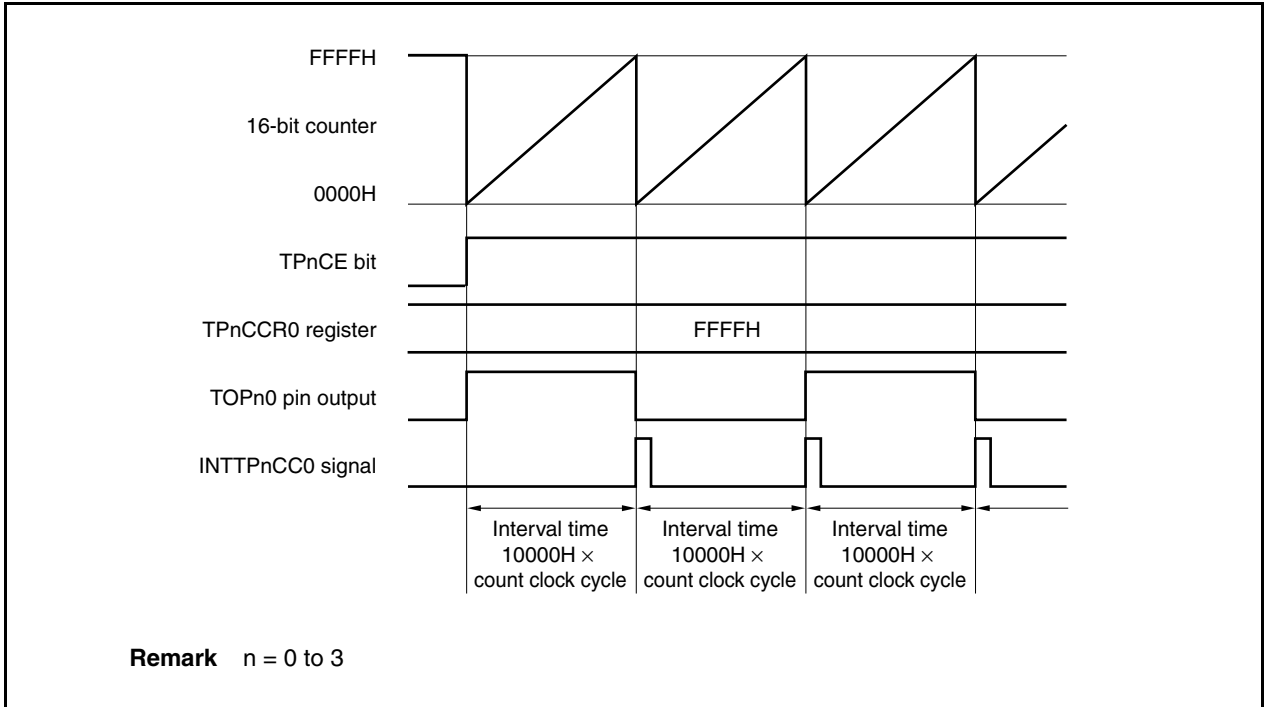
If the TPnCCR0 register is set to 0000H, the INTTPnCC0 signal is generated at each count clock subsequent to the first count clock, and the output of the TOPn0 pin is inverted.

The value of the 16-bit counter is always 0000H.



(b) Operation if TPnCCR0 register is set to FFFFH

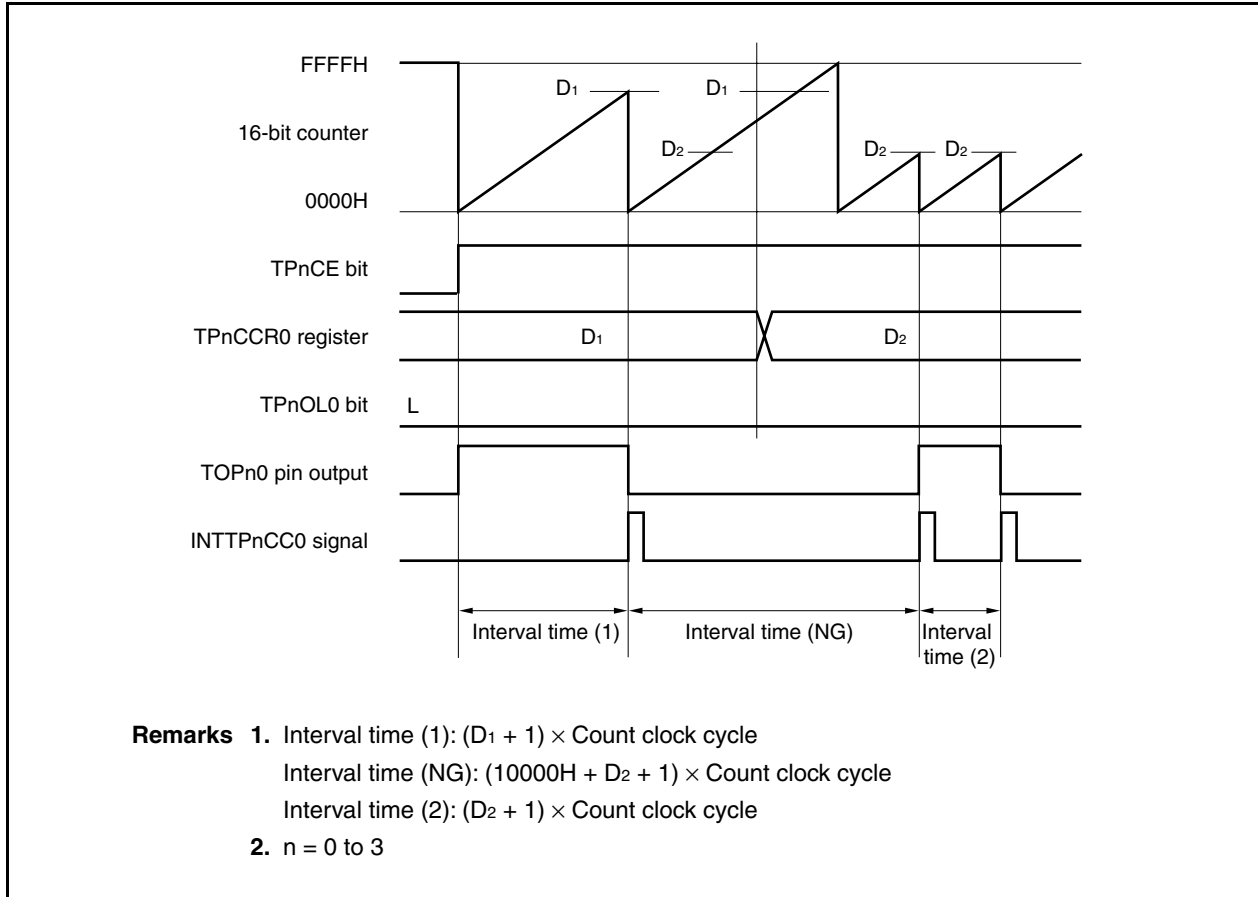
If the TPnCCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH. The counter is cleared to 0000H in synchronization with the next count-up timing. The INTTPnCC0 signal is generated and the output of the TOPn0 pin is inverted. At this time, an overflow interrupt request signal (INTTPnOV) is not generated, nor is the overflow flag (TPnOPT0.TPnOVF bit) set to 1.



(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.



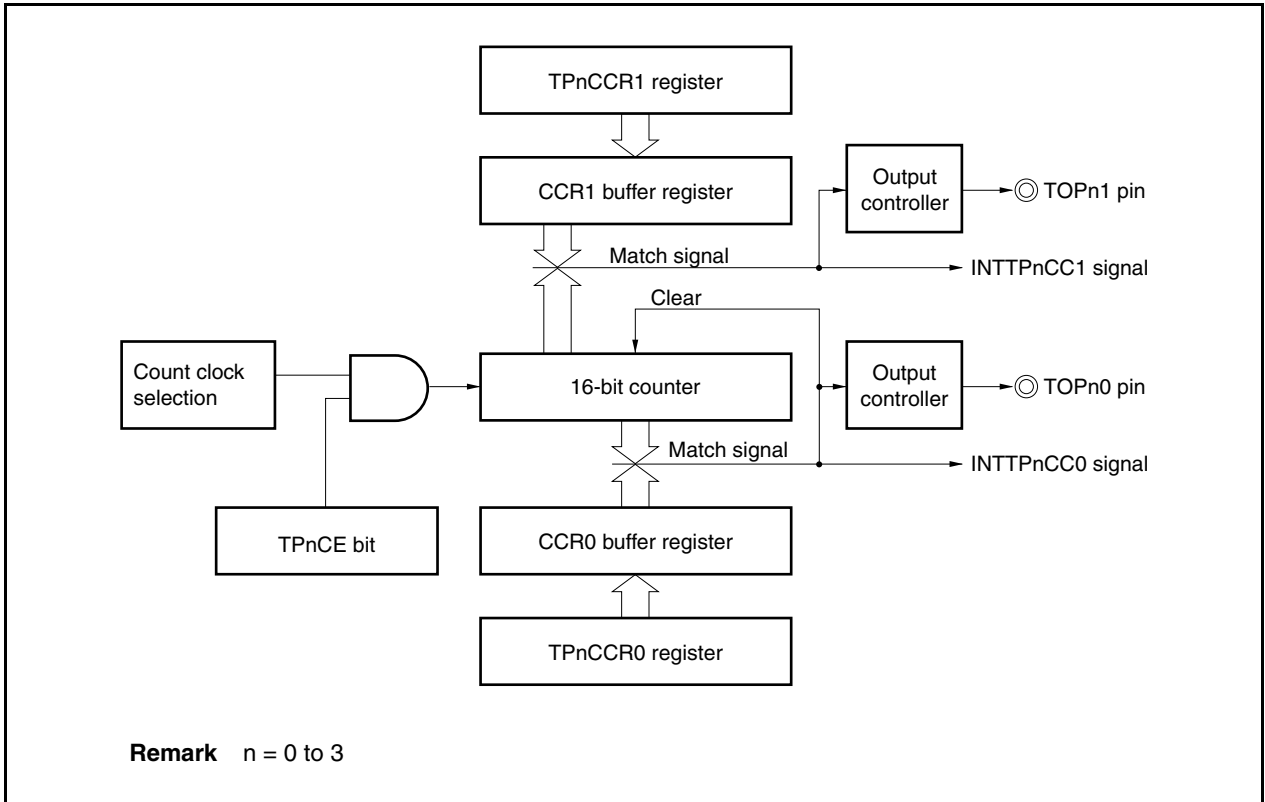
If the value of the TPnCCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1 , the count value is transferred to the CCR0 buffer register as soon as the TPnCCR0 register has been rewritten. Consequently, the value of the 16-bit counter that is compared is 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 " $(D_1 + 1) \times \text{Count clock cycle}$ " or " $(D_2 + 1) \times \text{Count clock cycle}$ " originally expected, but may be generated at an interval of " $(10000H + D_2 + 1) \times \text{Count clock period}$ ".

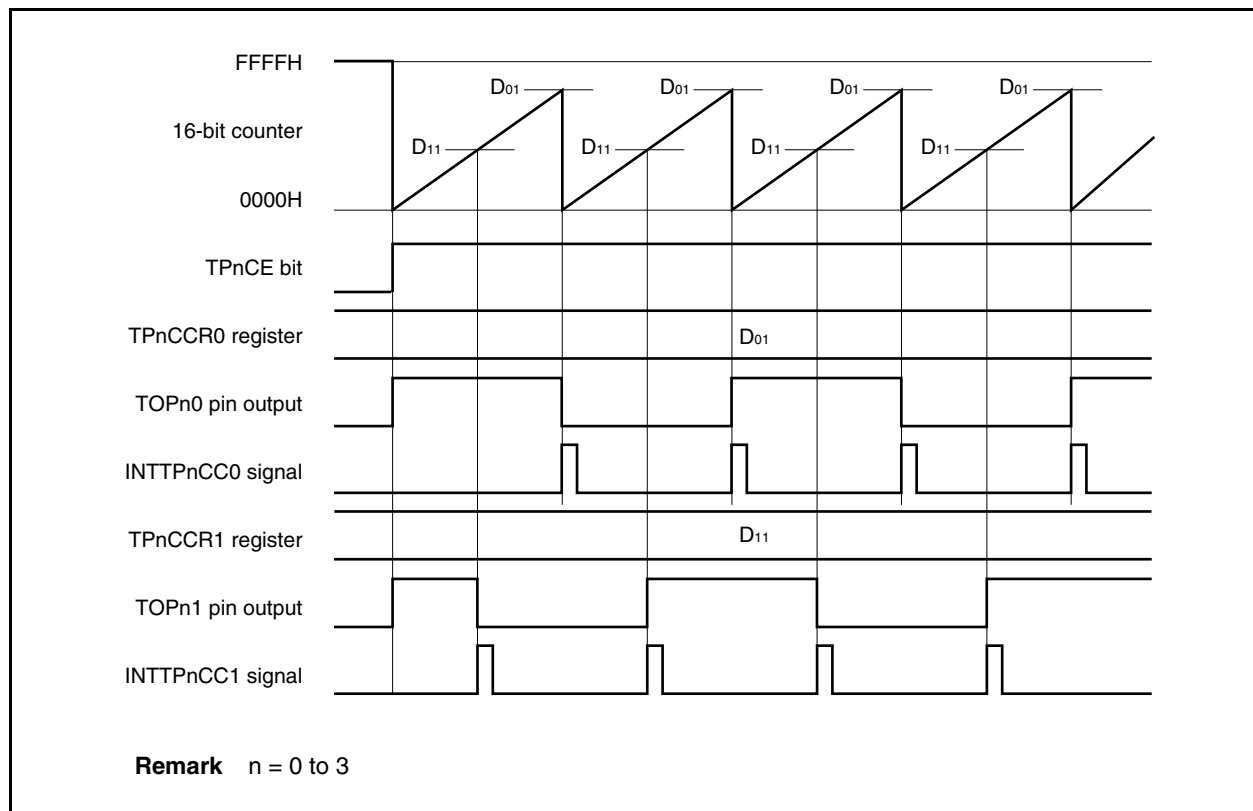
(d) Operation of TPnCCR1 register

Figure 6-6. Configuration of TPnCCR1 Register



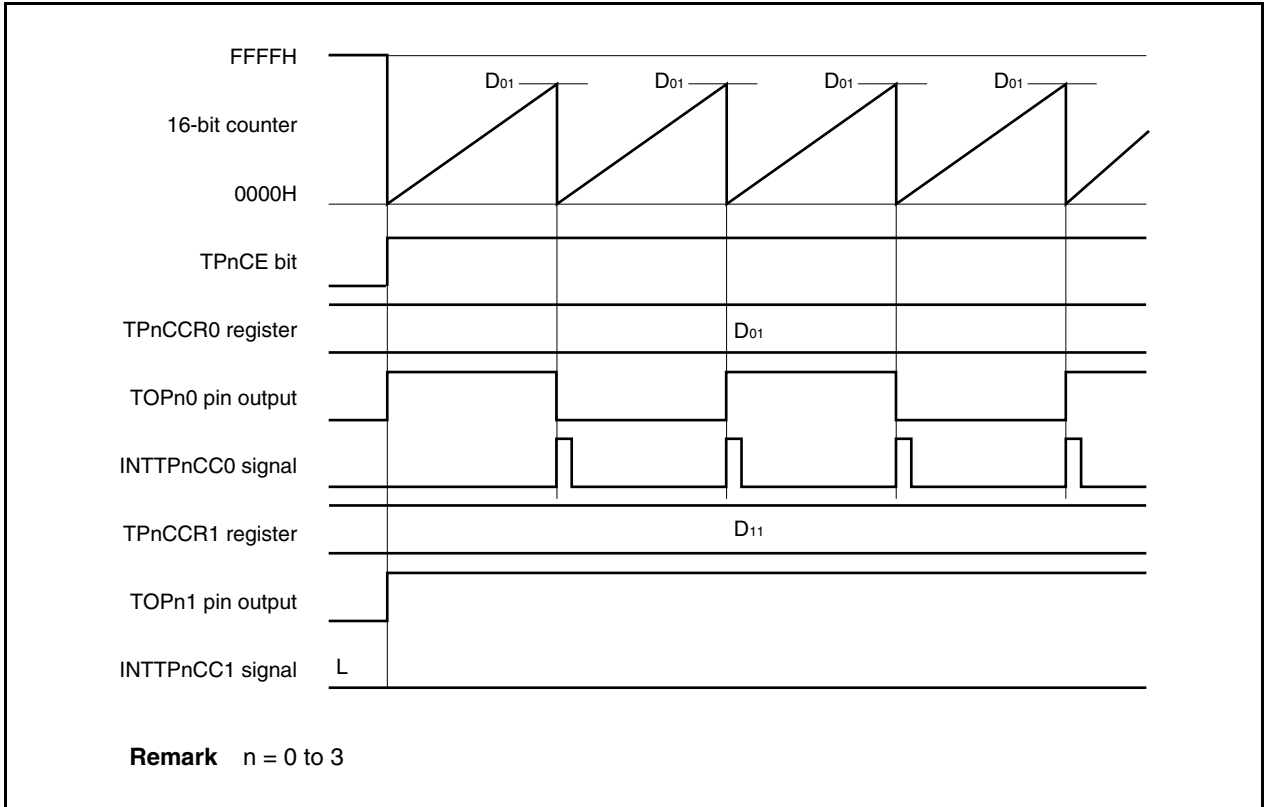
If the set value of the TPnCCR1 register is less than the set value of the TPnCCR0 register, the INTTPnCC1 signal is generated once per cycle. At the same time, the output of the TOPn1 pin is inverted. The TOPn1 pin outputs a square wave with the same cycle as that output by the TOPn0 pin.

Figure 6-7. Timing Chart When $D_{01} \geq D_{11}$



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 6-8. Timing Chart When $D_{01} < D_{11}$



6.5.2 External event count mode (TPnMD2 to TPnMD0 bits = 001)

In the external event count mode, the valid edge of the external event count input is counted when the TPnCTL0.TPnCE bit is set to 1, and an interrupt request signal (INTTPnCC0) is generated each time the specified number of edges have been counted. The TOPn0 pin cannot be used.

Usually, the TPnCCR1 register is not used in the external event count mode.

Figure 6-9. Configuration in External Event Count Mode

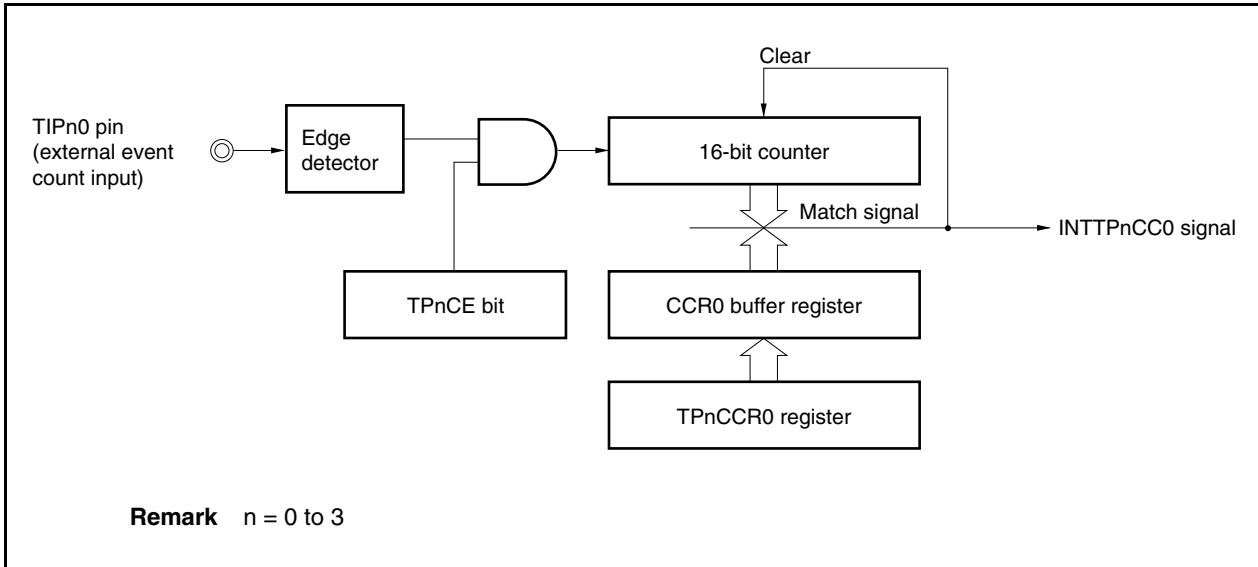
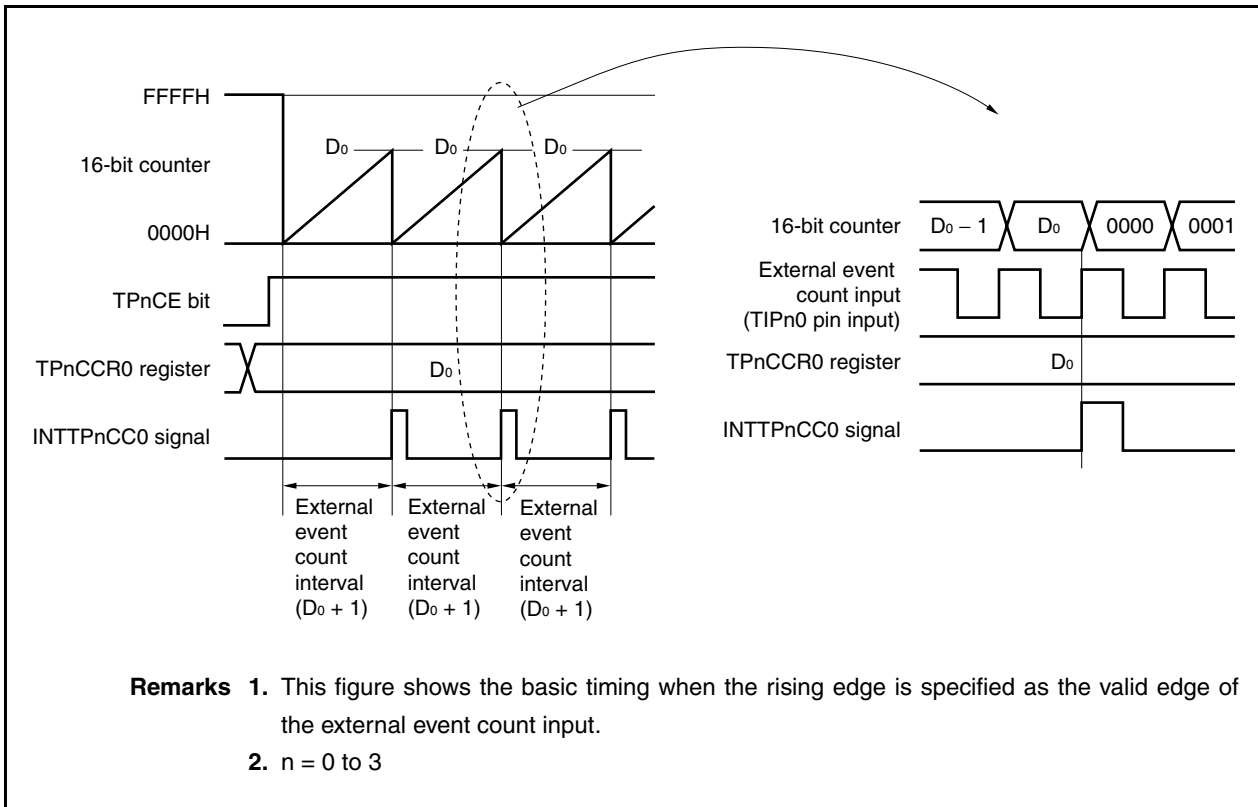


Figure 6-10. Basic Timing in External Event Count Mode



When the TPnCE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H. The counter counts each time the valid edge of external event count input is detected. Additionally, the set value of the TPnCCR0 register is transferred to the CCR0 buffer register.

When the count value of the 16-bit counter matches the value of the CCR0 buffer register, the 16-bit counter is cleared to 0000H, and a compare match interrupt request signal (INTTPnCC0) is generated.

The INTTPnCC0 signal is generated each time the valid edge of the external event count input has been detected (set value of TPnCCR0 register + 1) times.

Figure 6-11. Register Setting for Operation in External Event Count Mode (1/2)

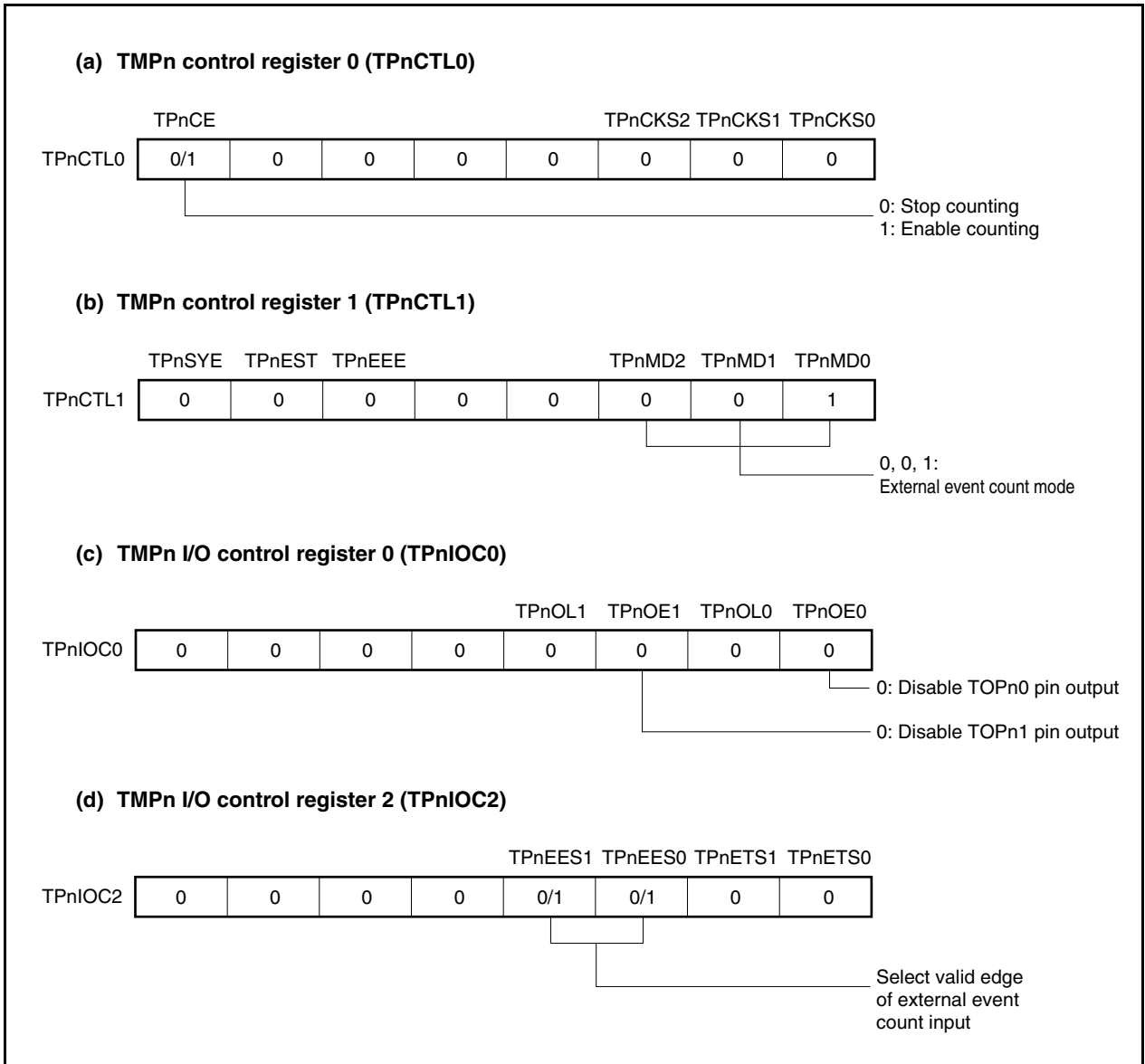


Figure 6-11. Register Setting for Operation in External Event Count Mode (2/2)

(e) TMPn counter read buffer register (TPnCNT)

The count value of the 16-bit counter can be read by reading the TPnCNT register.

(f) TMPn capture/compare register 0 (TPnCCR0)

If D_0 is set to the TPnCCR0 register, the counter is cleared and a compare match interrupt request signal (INTTPnCC0) is generated when the number of external event counts reaches ($D_0 + 1$).

(g) TMPn capture/compare register 1 (TPnCCR1)

Usually, the TPnCCR1 register is not used in the external event count mode. However, the set value of the TPnCCR1 register is transferred to the CCR1 buffer register. When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTPnCC1) is generated.

Therefore, mask the interrupt signal by using the interrupt mask flag (TPnCCMK1).

Caution When an external clock is used as the count clock, the external clock can be input only from the TIPn0 pin. At this time, set the TPnIOC1.TPnIS1 and TPnIOC1.TPnIS0 bits to 00 (capture trigger input (TIPn0 pin): no edge detection).

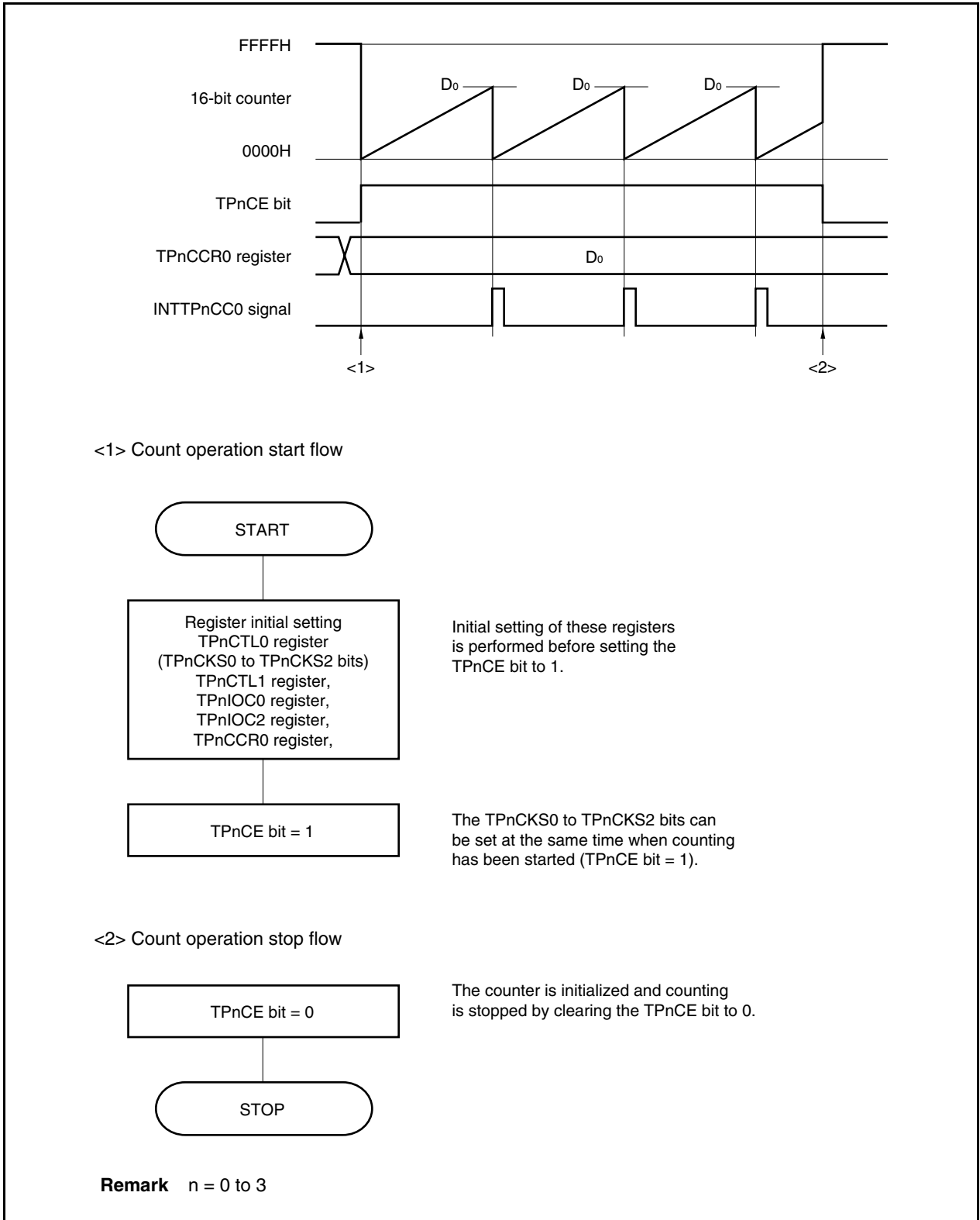
Remarks 1. TMPn I/O control register 1 (TPnIOC1) and TMPn option register 0 (TPnOPT0) are not used in the external event count mode.

2. $n = 0$ to 3

<R>

(1) External event count mode operation flow

Figure 6-12. Flow of Software Processing in External Event Count Mode

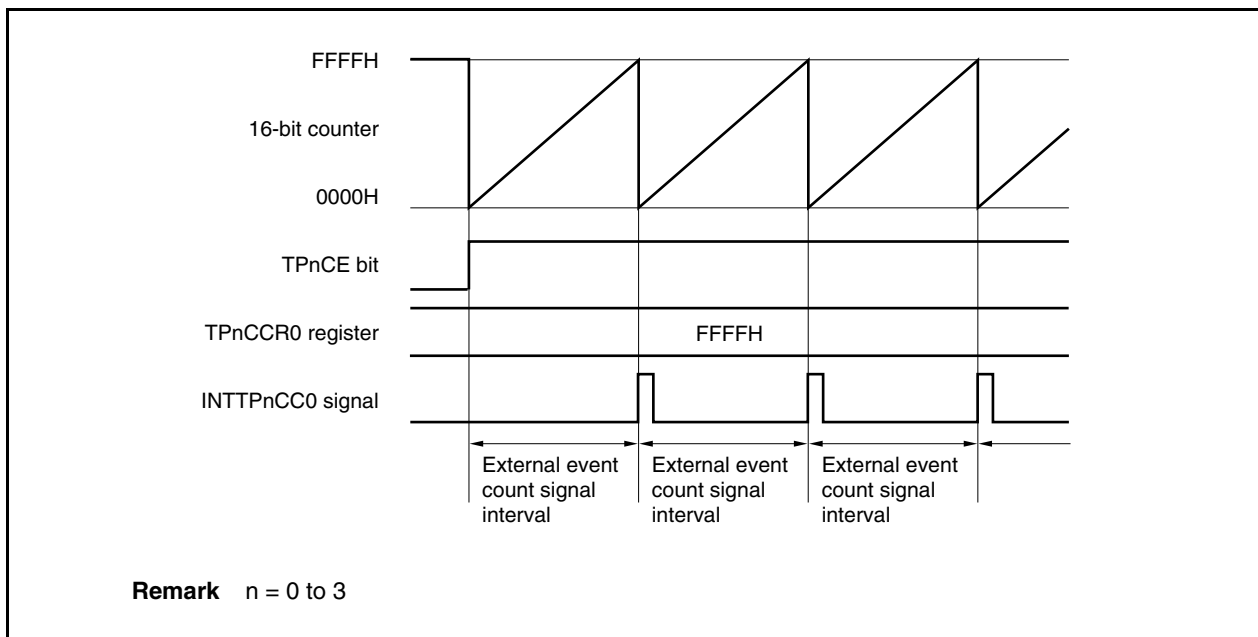


(2) Operation timing in external event count mode

- Cautions**
1. In the external event count mode, do not set the TPnCCR0 register to 0000H.
 2. In the external event count mode, use of the timer output is disabled. If performing timer output using external event count input, set the interval timer mode, and select the operation enabled by the external event count input for the count clock (TPnCTL1.TPnMD2 to TPnCTL1.TPnMD0 bits = 000, TPnCTL1.TPnEEE bit = 1).

(a) Operation if TPnCCR0 register is set to FFFFH

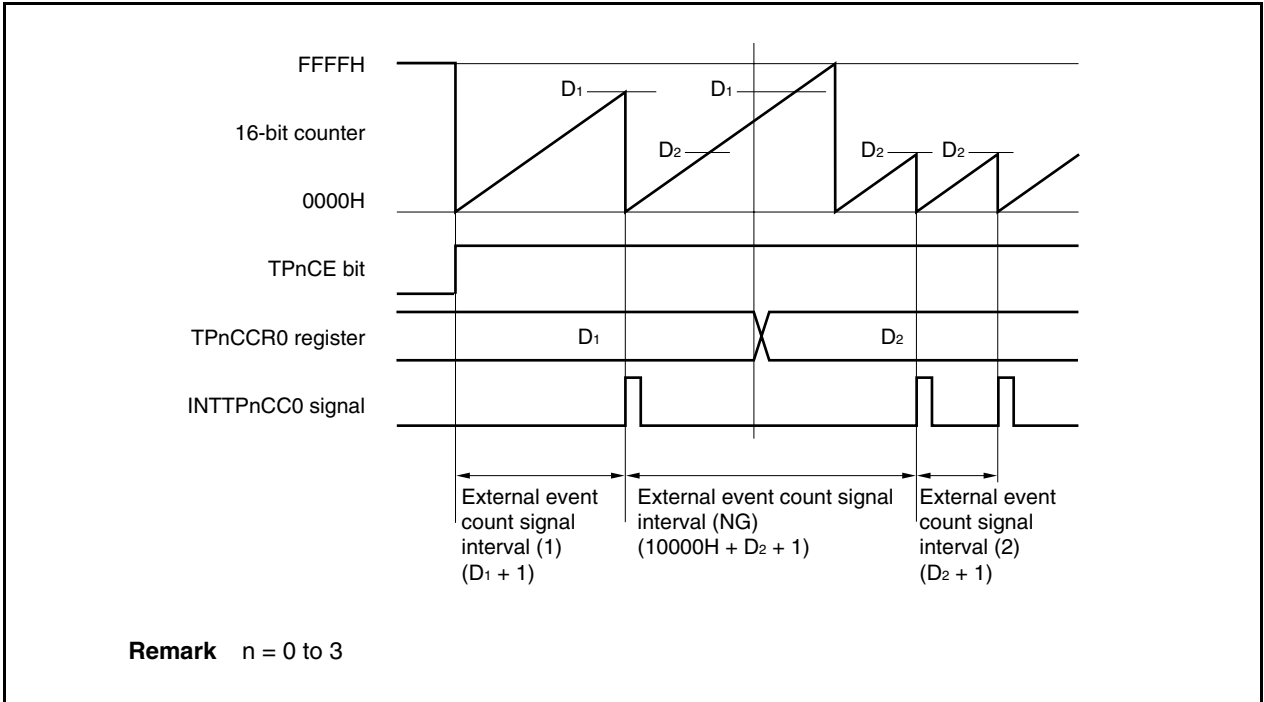
If the TPnCCR0 register is set to FFFFH, the 16-bit counter counts to FFFFH each time the valid edge of the external event count signal has been detected. The 16-bit counter is cleared to 0000H in synchronization with the next count-up timing, and the INTTPnCC0 signal is generated. At this time, the TPnOPT0.TPnOVF bit is not set.



(b) Notes on rewriting the TPnCCR0 register

To change the value of the TPnCCR0 register to a smaller value, stop counting once and then change the set value.

If the value of the TPnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow.



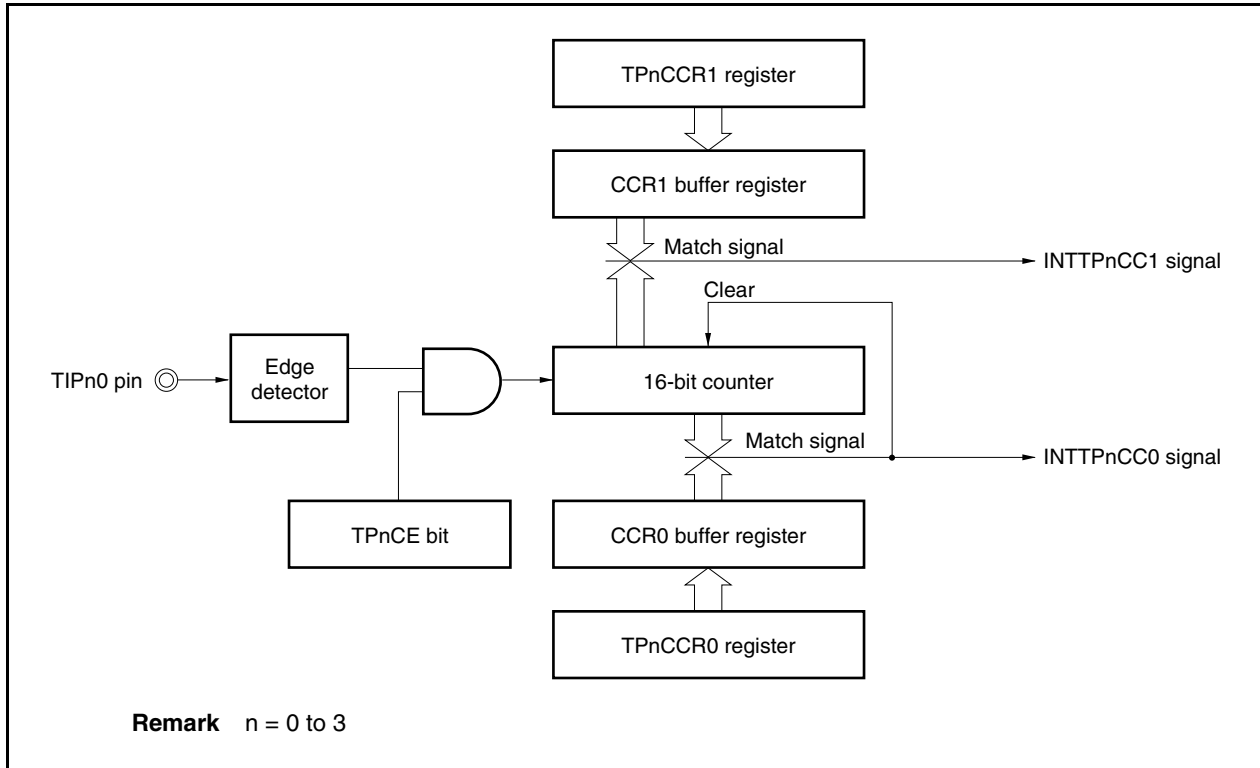
If the value of the TPnCCR0 register is changed from D₁ to D₂ while the count value is greater than D₂ but less than D₁, the count value is transferred to the CCR0 buffer register as soon as the TPnCCR0 register has been rewritten. Consequently, the value that is compared with the 16-bit counter is D₂.

Because the count value has already exceeded D₂, however, the 16-bit counter counts up to FFFFH, overflows, and then counts up again from 0000H. When the count value matches D₂, the INTTPnCC0 signal is generated.

Therefore, the INTTPnCC0 signal may not be generated at the valid edge count of “(D₁ + 1) times” or “(D₂ + 1) times” originally expected, but may be generated at the valid edge count of “(10000H + D₂ + 1) times”.

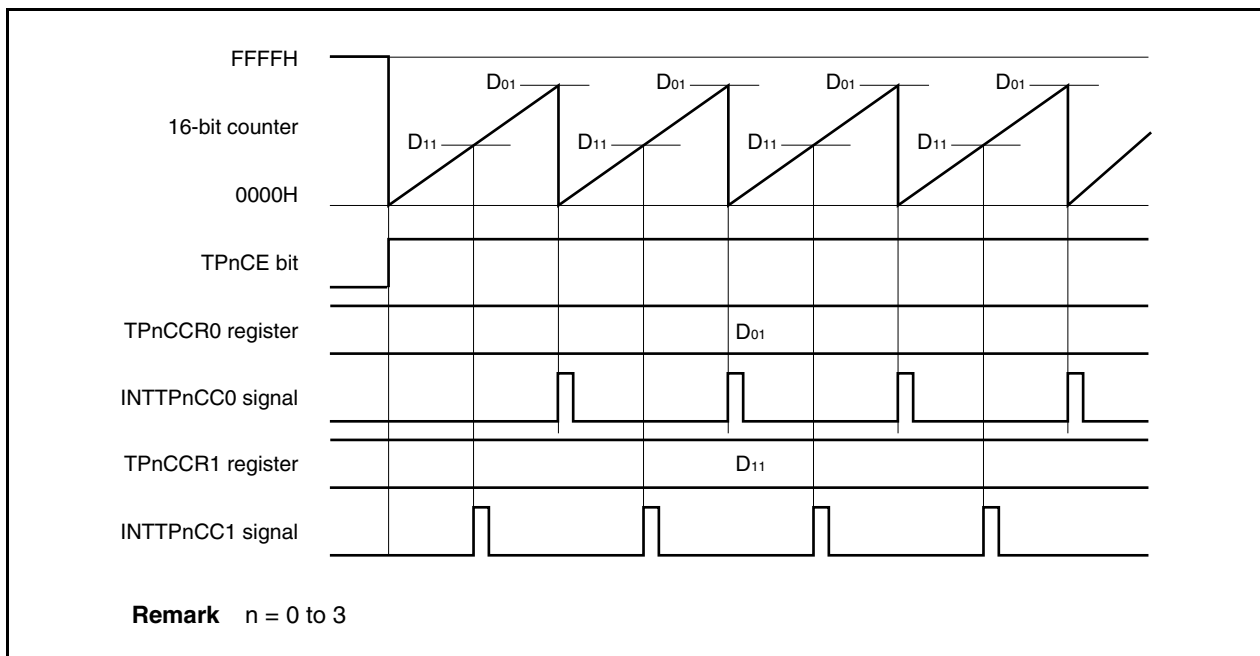
(c) Operation of TPnCCR1 register

Figure 6-13. Configuration of TPnCCR1 Register



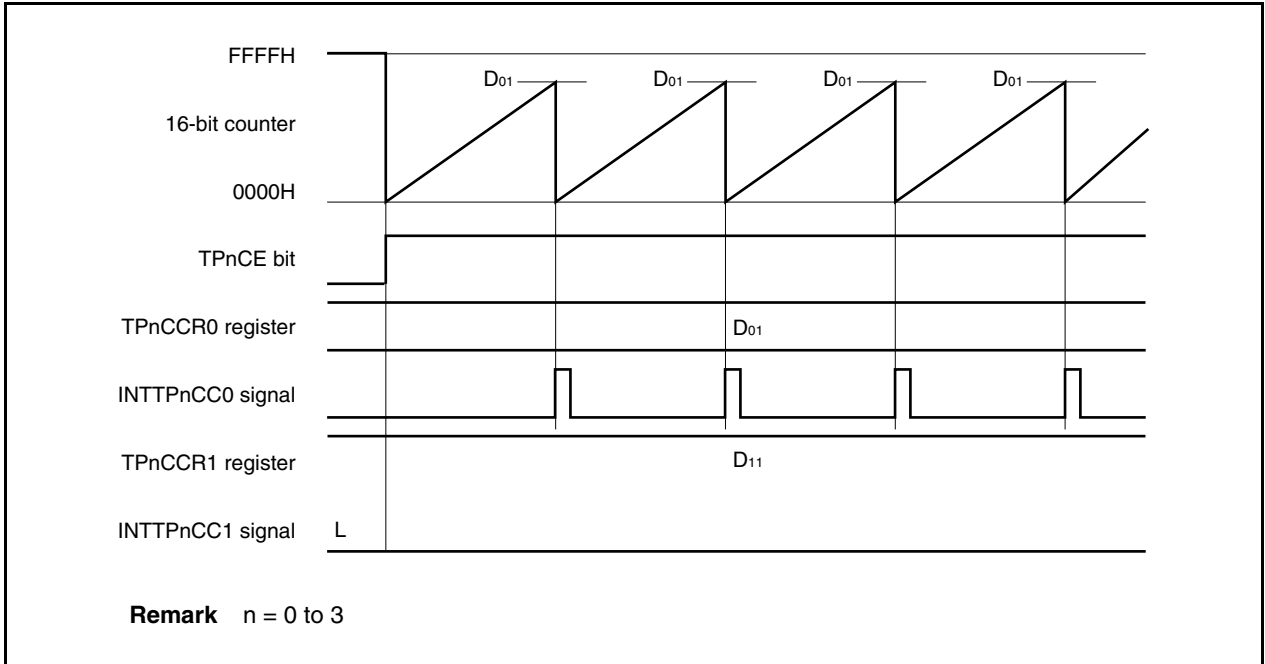
If the set value of the TPnCCR1 register is smaller than the set value of the TPnCCR0 register, the INTTPnCC1 signal is generated once per cycle.

Figure 6-14. Timing Chart When $D_{01} \geq D_{11}$



If the set value of the TPnCCR1 register is greater than the set value of the TPnCCR0 register, the INTTPnCC1 signal is not generated because the count value of the 16-bit counter and the value of the TPnCCR1 register do not match.

Figure 6-15. Timing Chart When $D_{01} < D_{11}$



6.5.3 External trigger pulse output mode (TPnMD2 to TPnMD0 bits = 010)

In the external trigger pulse output mode, 16-bit timer/event counter P waits for a trigger when the TPnCTL0.TPnCE bit is set to 1. When the valid edge of an external trigger input signal is detected, 16-bit timer/event counter P starts counting, and outputs a PWM waveform from the 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 6-16. Configuration in External Trigger Pulse Output Mode

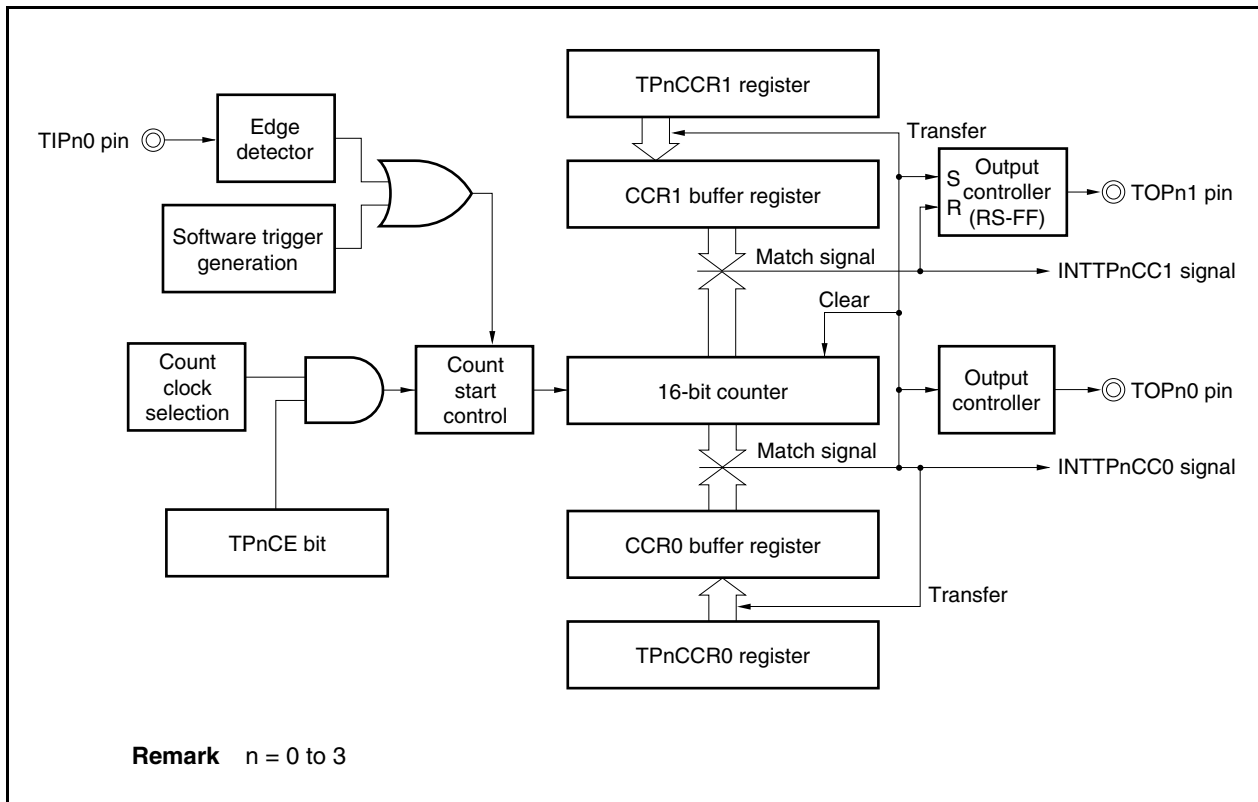
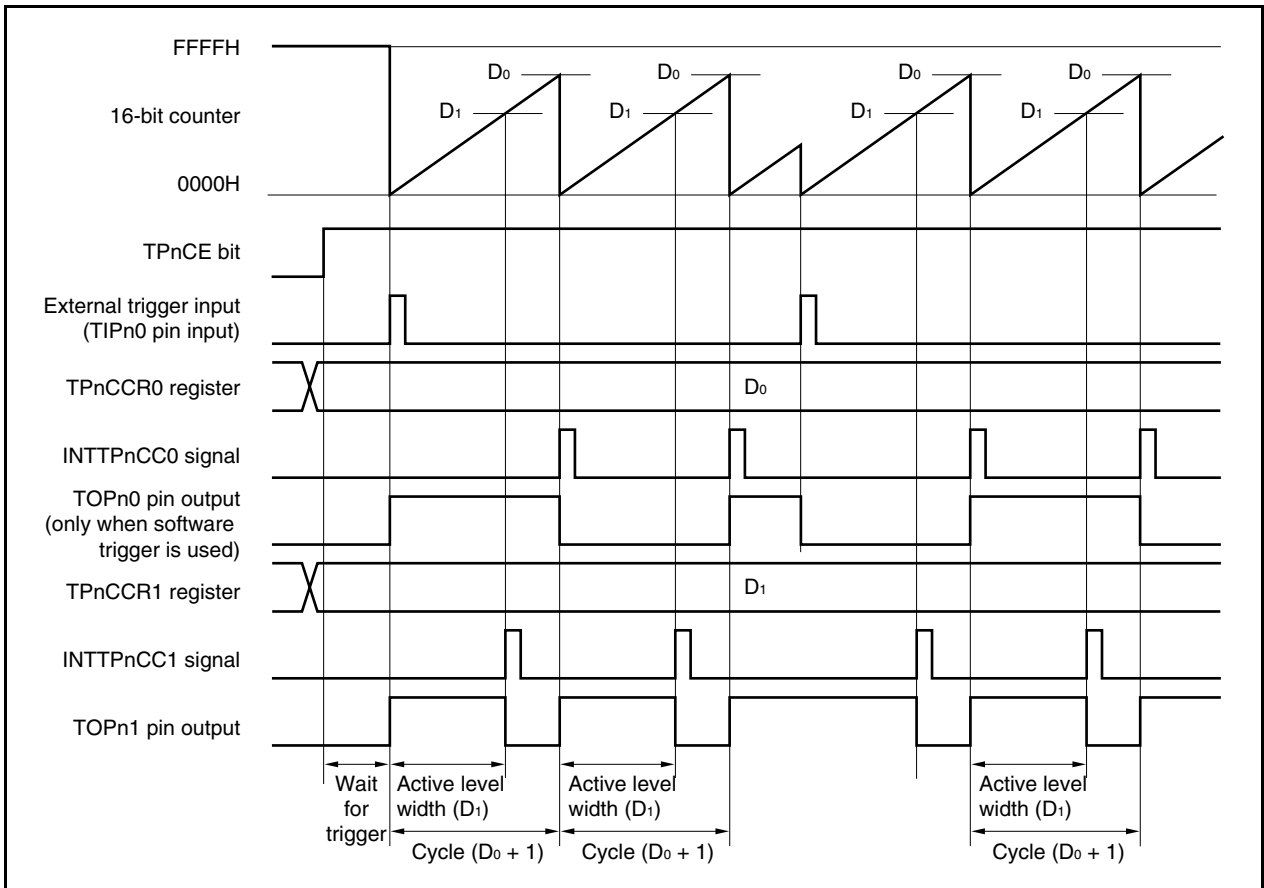


Figure 6-17. Basic Timing in External Trigger Pulse Output Mode



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 output of the TOPn0 pin is inverted. The TOPn1 pin outputs a high level regardless of the status (high/low) when a trigger occurs.)

The active level width, cycle, and duty factor of the PWM waveform can be calculated as follows.

$$\text{Active level width} = (\text{Set value of TPnCCR1 register}) \times \text{Count clock cycle}$$

$$\text{Cycle} = (\text{Set value of TPnCCR0 register} + 1) \times \text{Count clock cycle}$$

$$\text{Duty factor} = (\text{Set value of TPnCCR1 register}) / (\text{Set value of TPnCCR0 register} + 1)$$

The compare match request signal INTTPnCC0 is generated when the 16-bit counter counts next time after its count value matches the value of the CCR0 buffer register, and the 16-bit counter is cleared to 0000H. The compare match interrupt request signal INTTPnCC1 is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

The value set to the TPnCCRm register is transferred to the CCRm buffer register when the count value of the 16-bit counter matches the value of the CCRm buffer register and the 16-bit counter is cleared to 0000H.

The valid edge of an external trigger input signal, or setting the software trigger (TPnCTL1.TPnEST bit) to 1 is used as the trigger.

Remark n = 0 to 3, m = 0, 1

Figure 6-18. Setting of Registers in External Trigger Pulse Output Mode (1/2)

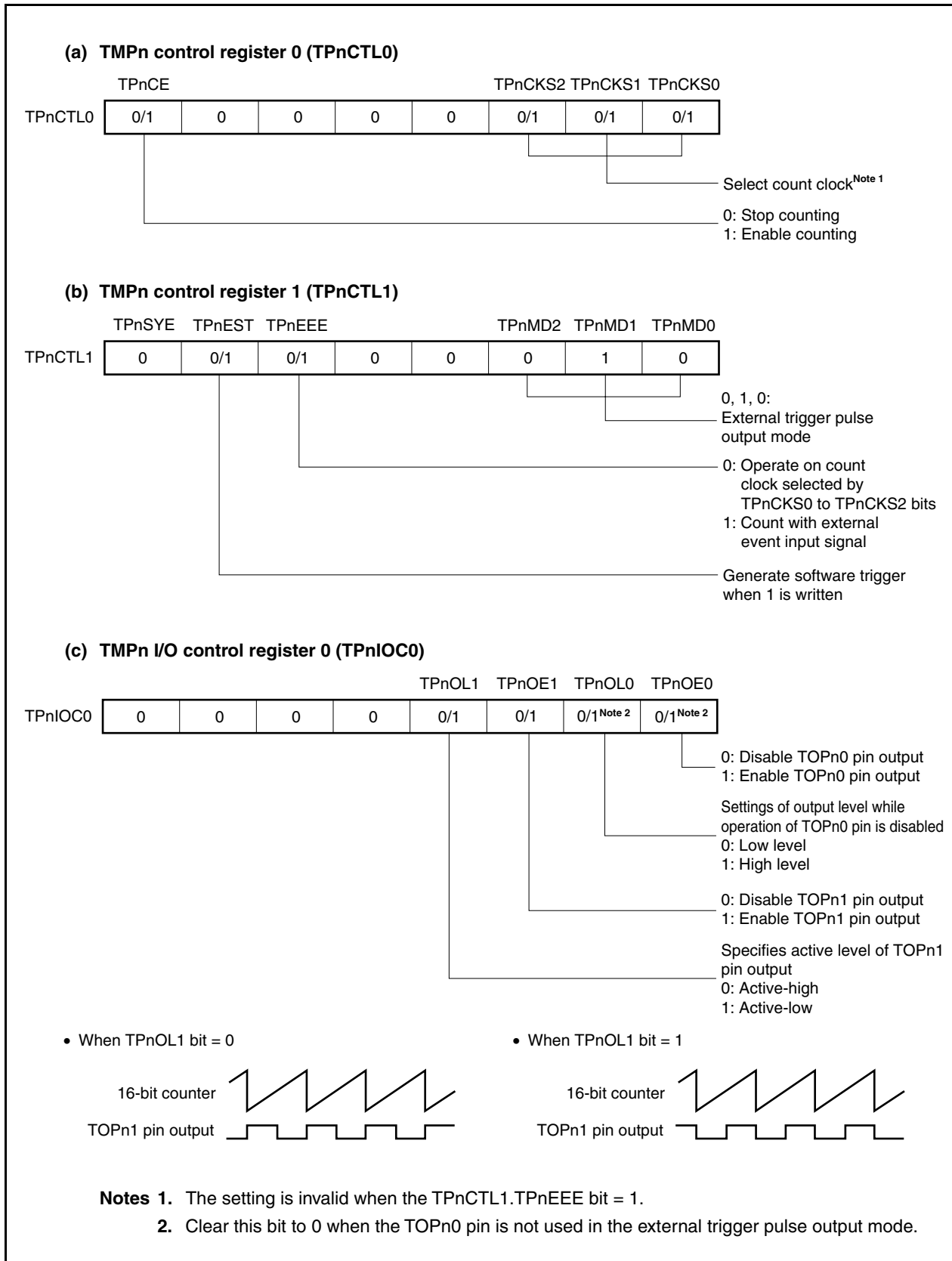
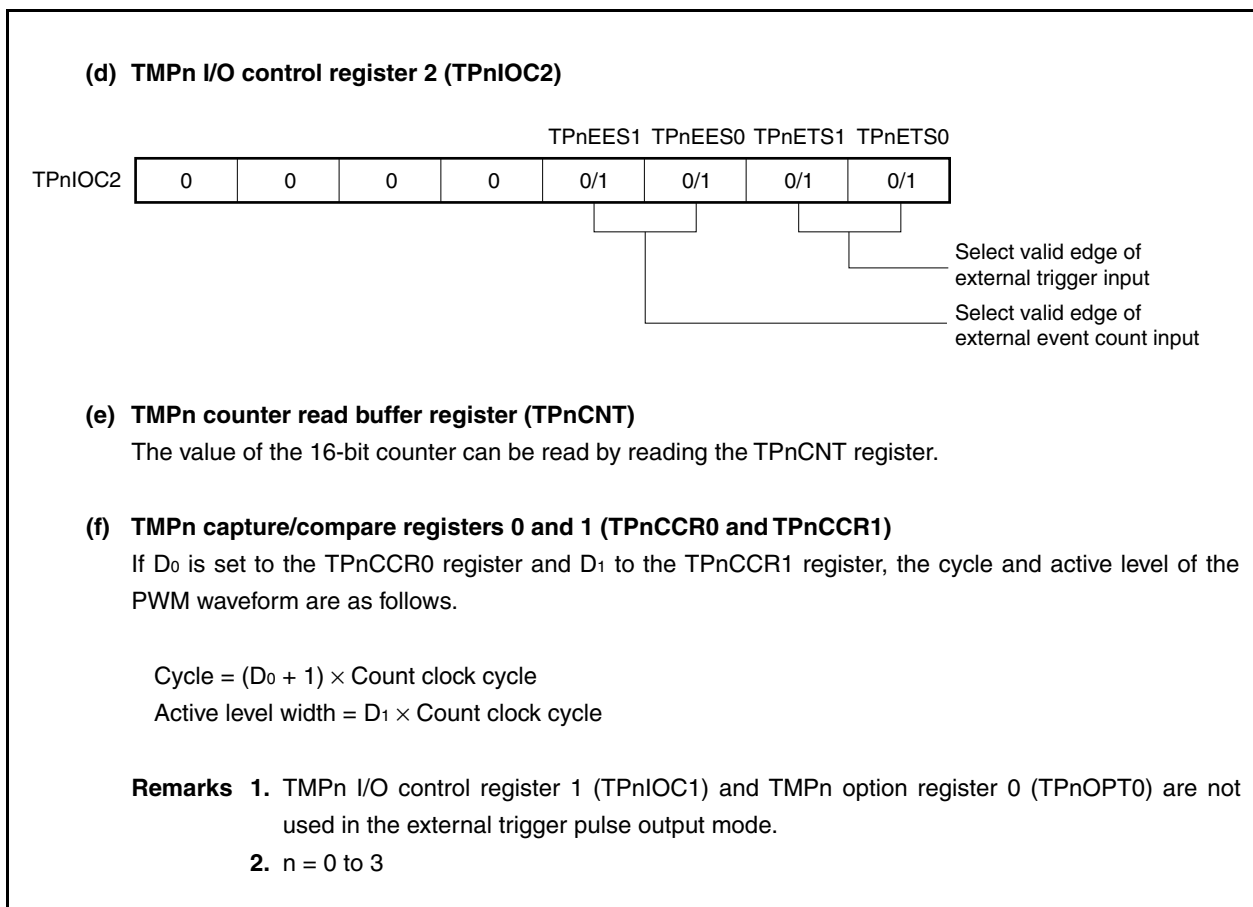


Figure 6-18. Setting of Registers in External Trigger Pulse Output Mode (2/2)



(1) Operation flow in external trigger pulse output mode

Figure 6-19. Software Processing Flow in External Trigger Pulse Output Mode (1/2)

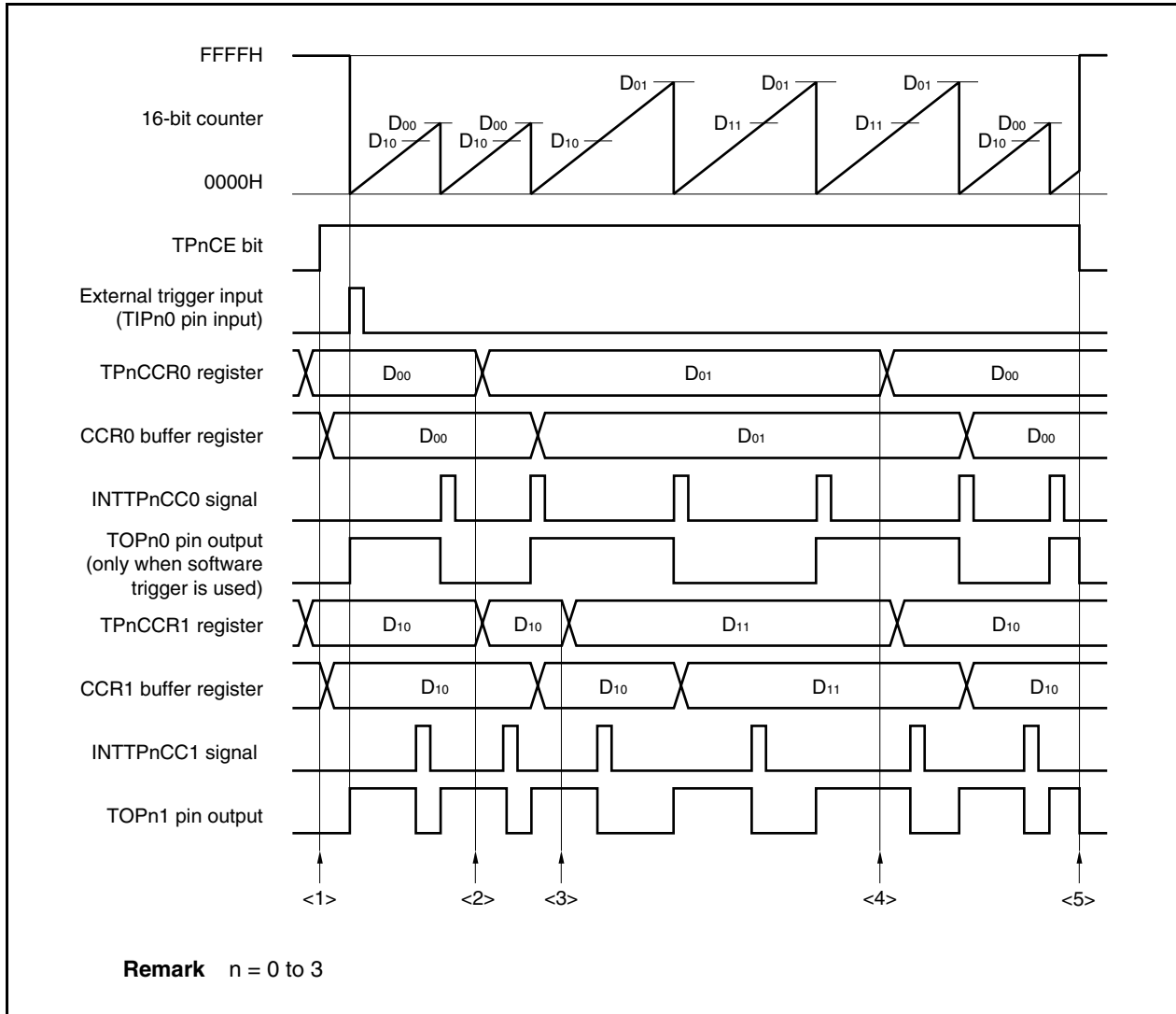
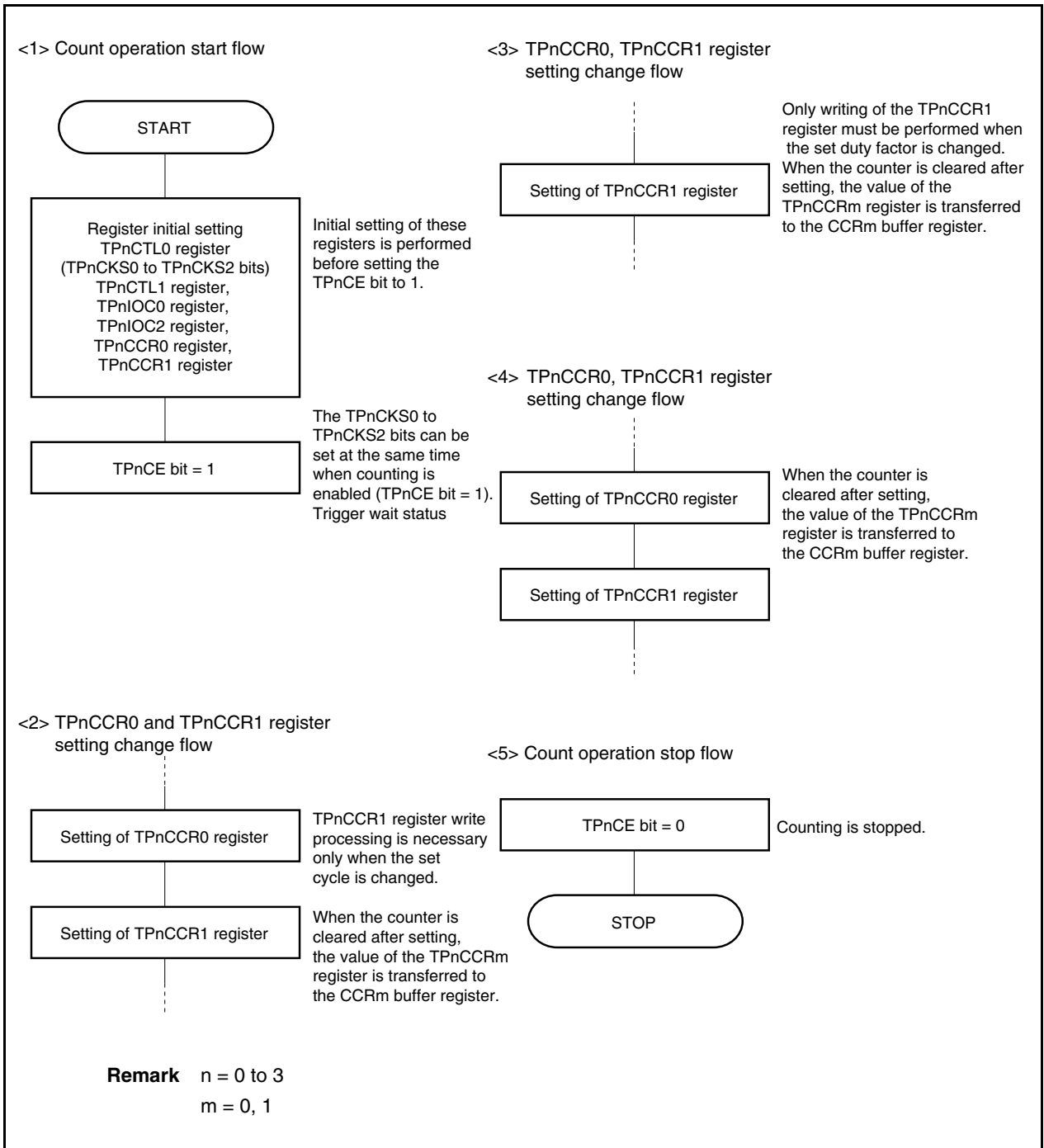


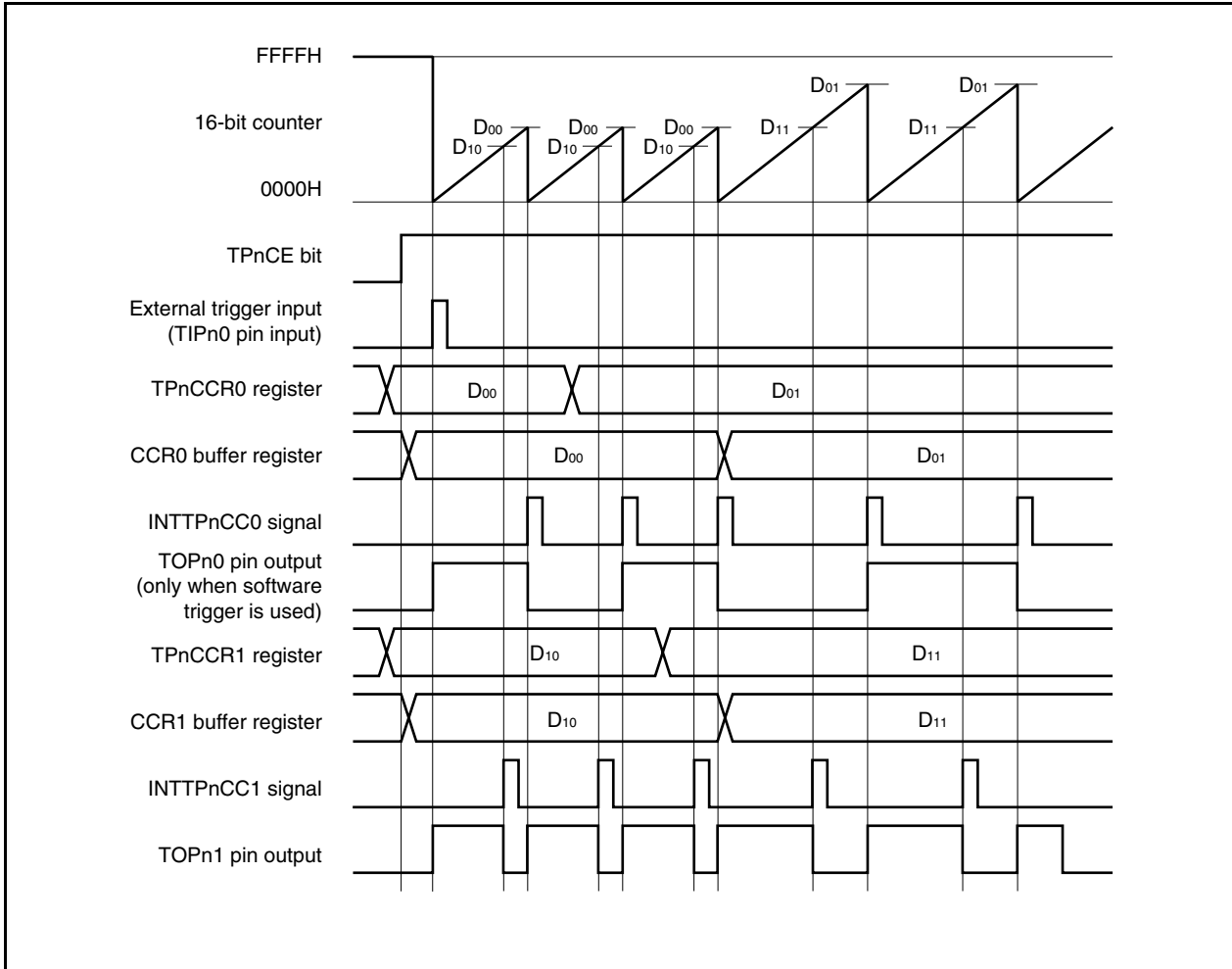
Figure 6-19. Software Processing Flow in External Trigger Pulse Output Mode (2/2)



(2) External trigger pulse output mode operation timing

(a) Note on changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TPnCCR1 register last.
 Rewrite the TPnCCRm register after writing the TPnCCR1 register after the INTTPnCC0 signal is detected.



In order to transfer data from the TPnCCRm register to the CCRm buffer register, the TPnCCR1 register must be written.

To change both the cycle and active level width of the PWM waveform at this time, first set the cycle to the TPnCCR0 register and then set the active level width to the TPnCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TPnCCR0 register, and then write the same value to the TPnCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, only the TPnCCR1 register has to be set.

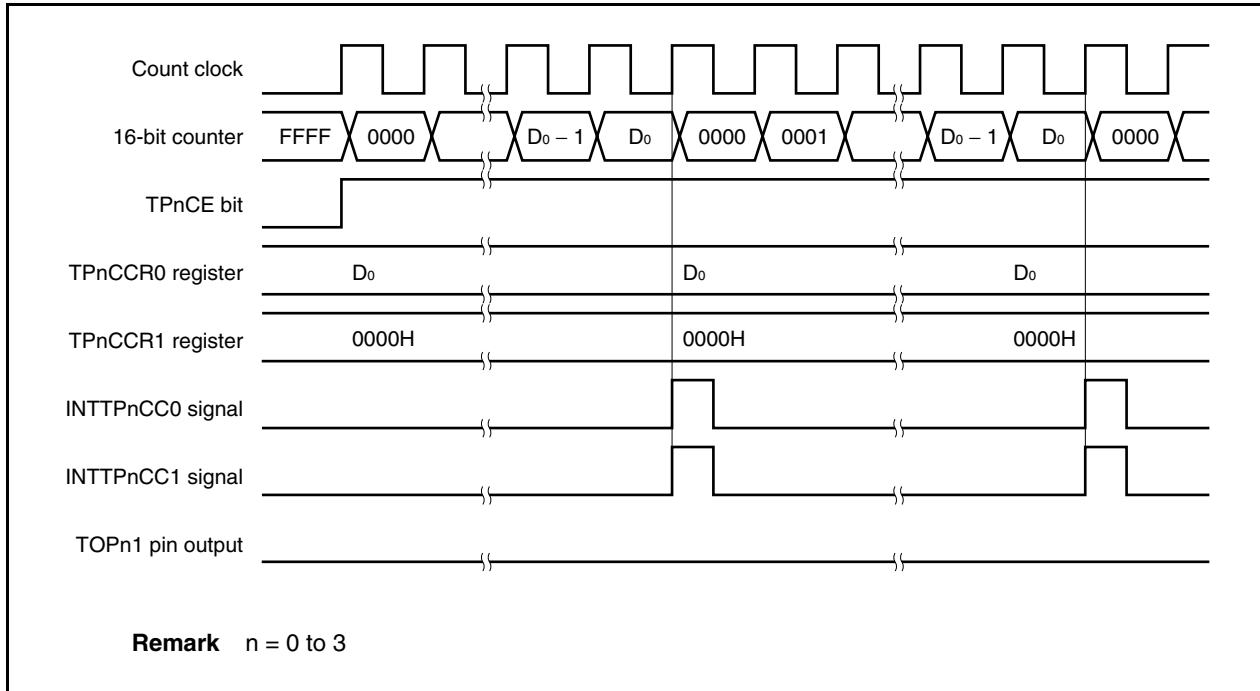
After data is written to the TPnCCR1 register, the value written to the TPnCCRm register is transferred to the CCRm buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TPnCCR0 or TPnCCR1 register again after writing the TPnCCR1 register once, do so after the INTTPnCC0 signal is generated. Otherwise, the value of the CCRm buffer register may become undefined because the timing of transferring data from the TPnCCRm register to the CCRm buffer register conflicts with writing the TPnCCRm register.

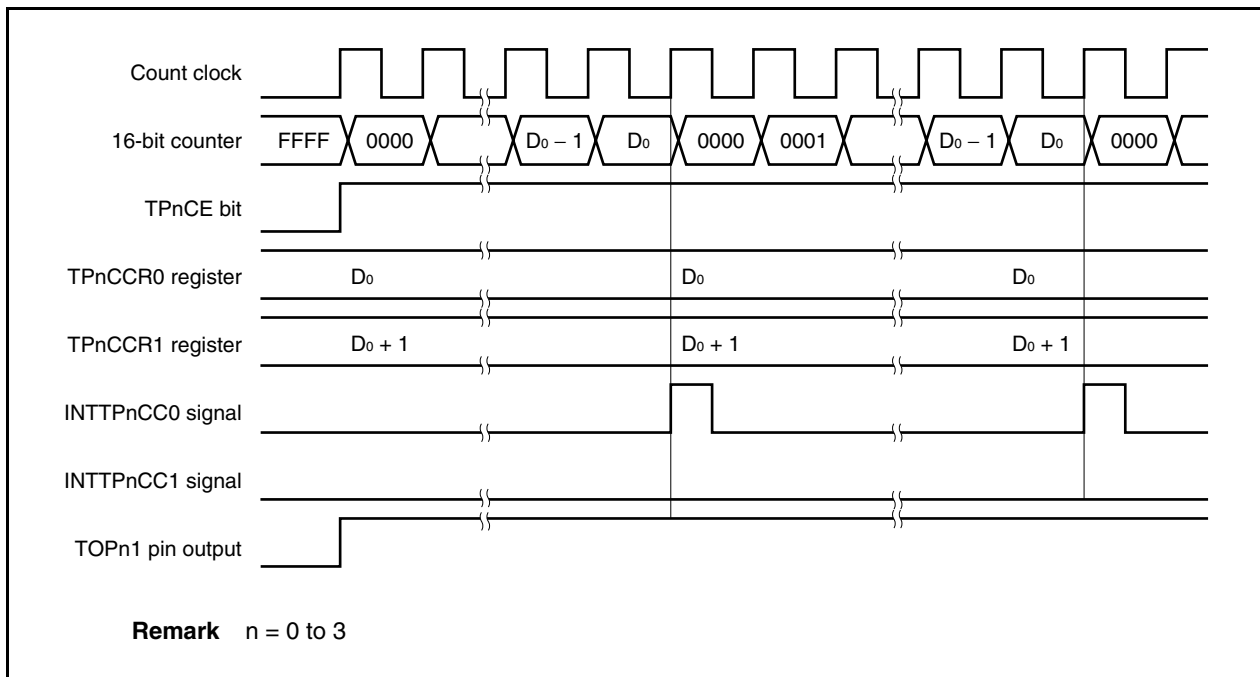
Remark n = 0 to 3
m = 0, 1

(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.

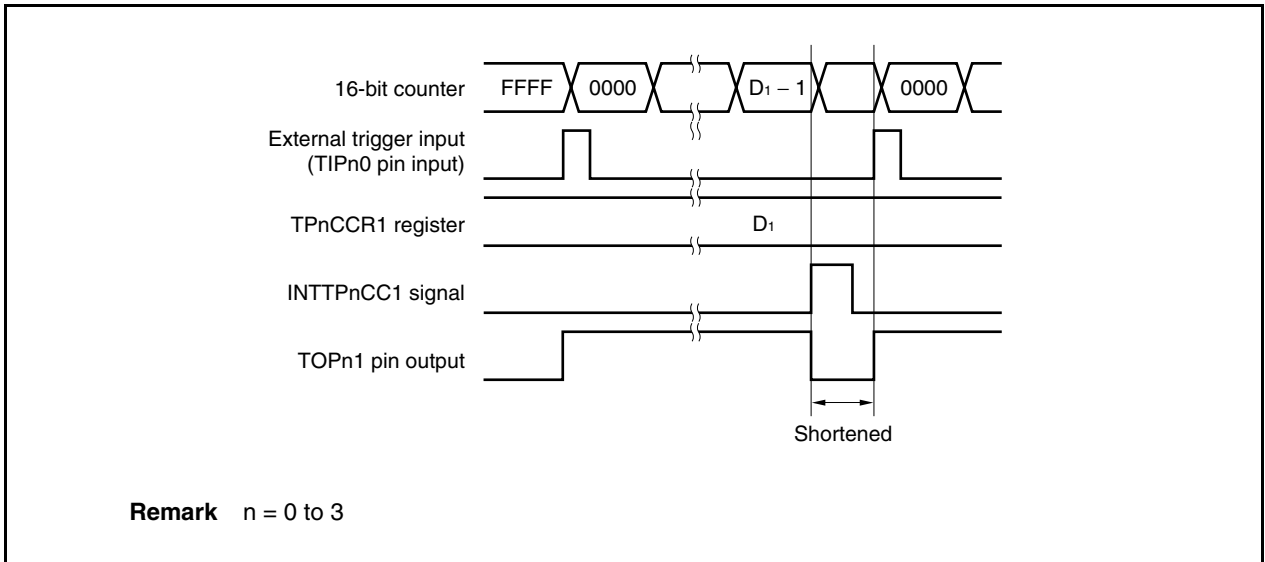


To output a 100% waveform, set a value of (set value of TPnCCR0 register + 1) to the TPnCCR1 register. If the set value of the TPnCCR0 register is FFFFH, 100% output cannot be produced.

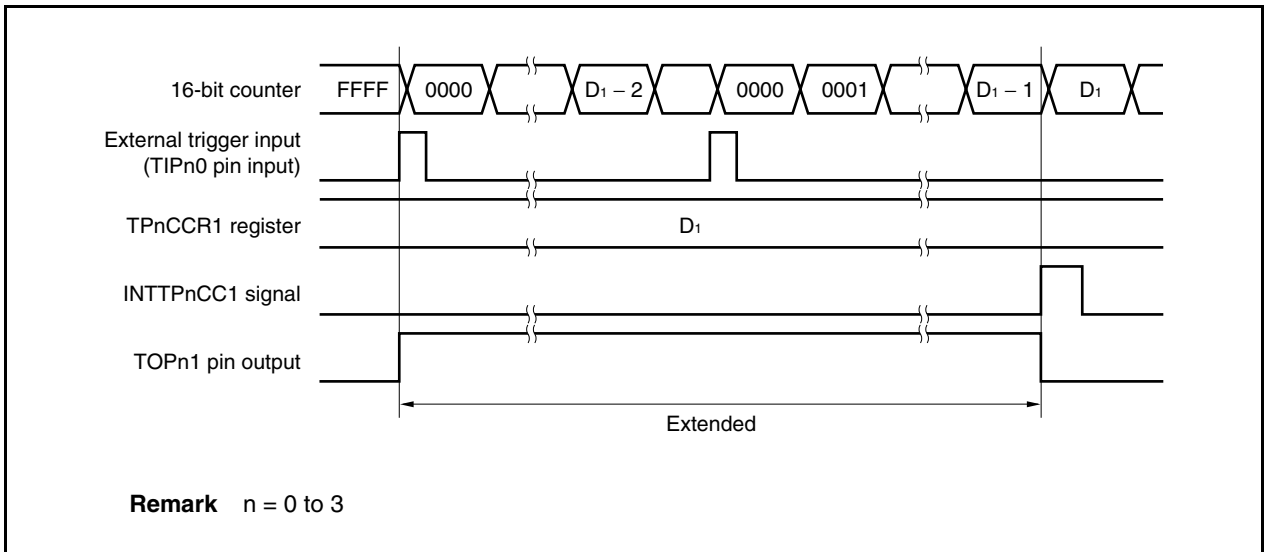


(c) Conflict between trigger detection and match with TPnCCR1 register

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.

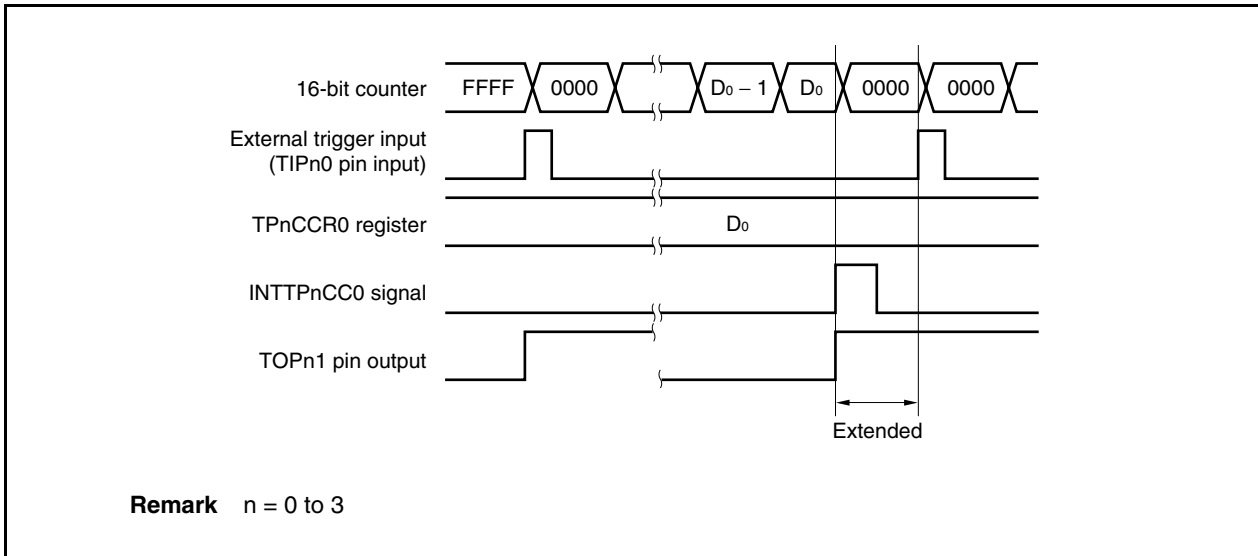


If the trigger is detected immediately before the INTTPnCC1 signal is generated, the INTTPnCC1 signal is not generated, and the 16-bit counter is cleared to 0000H and continues counting. The output signal of the TOPn1 pin remains active. Consequently, the active period of the PWM waveform is extended.

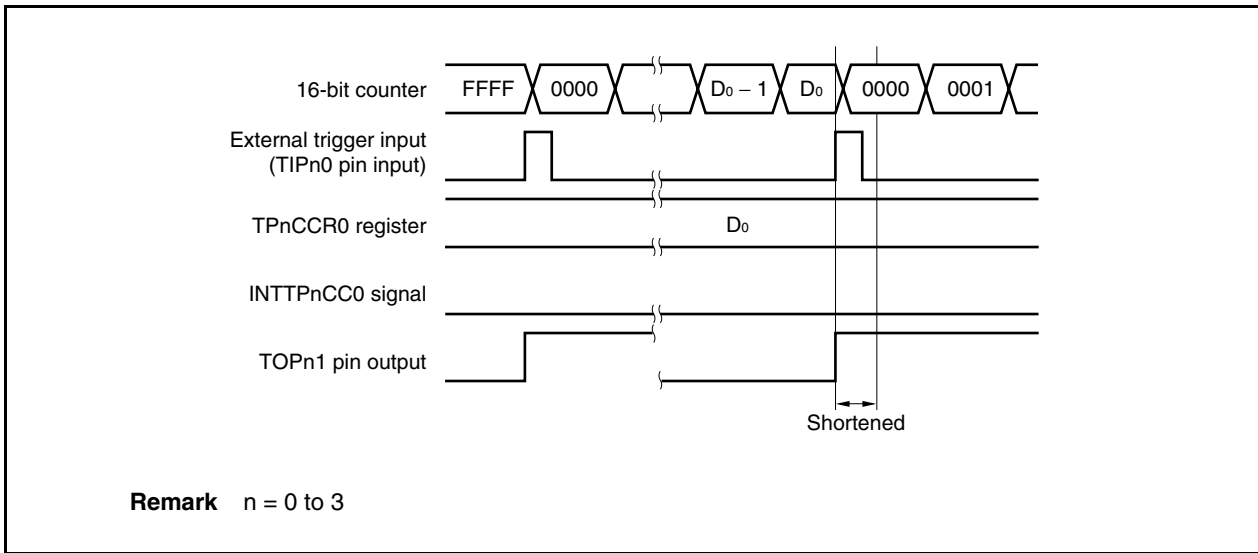


(d) Conflict between trigger detection and match with TPnCCR0 register

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.

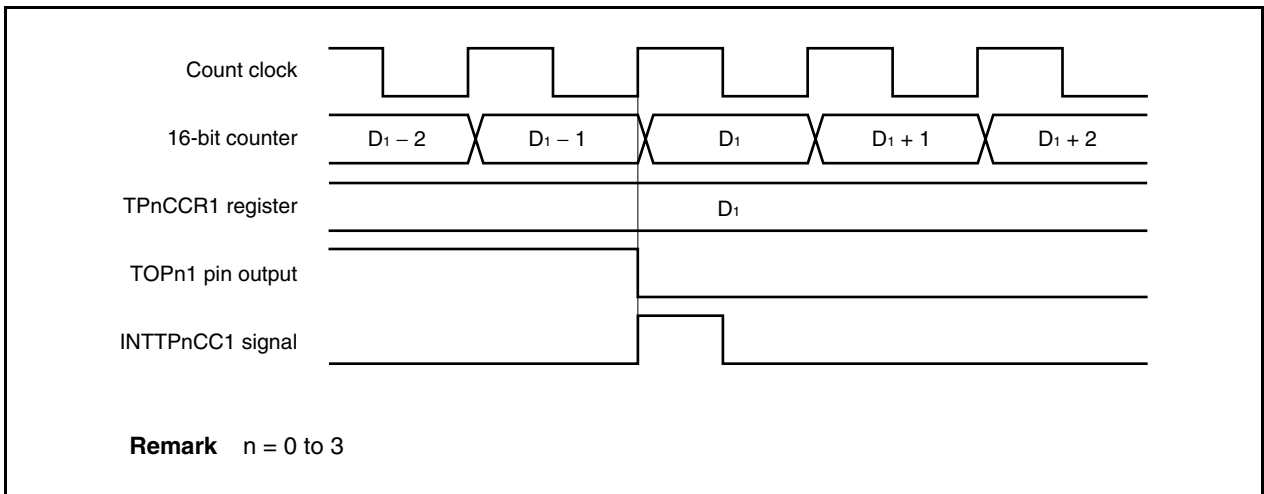


If the trigger is detected immediately before the INTTPnCC0 signal is generated, the INTTPnCC0 signal is not generated. The 16-bit counter is cleared to 0000H, the TOPn1 pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



(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.



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.

6.5.4 One-shot pulse output mode (TPnMD2 to TPnMD0 bits = 011)

In the one-shot pulse output mode, 16-bit timer/event counter P waits for a trigger when the TPnCTL0.TPnCE bit is set to 1. When the valid edge of an external trigger input is detected, 16-bit timer/event counter P starts counting, and outputs a one-shot pulse from the 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 6-20. Configuration in One-Shot Pulse Output Mode

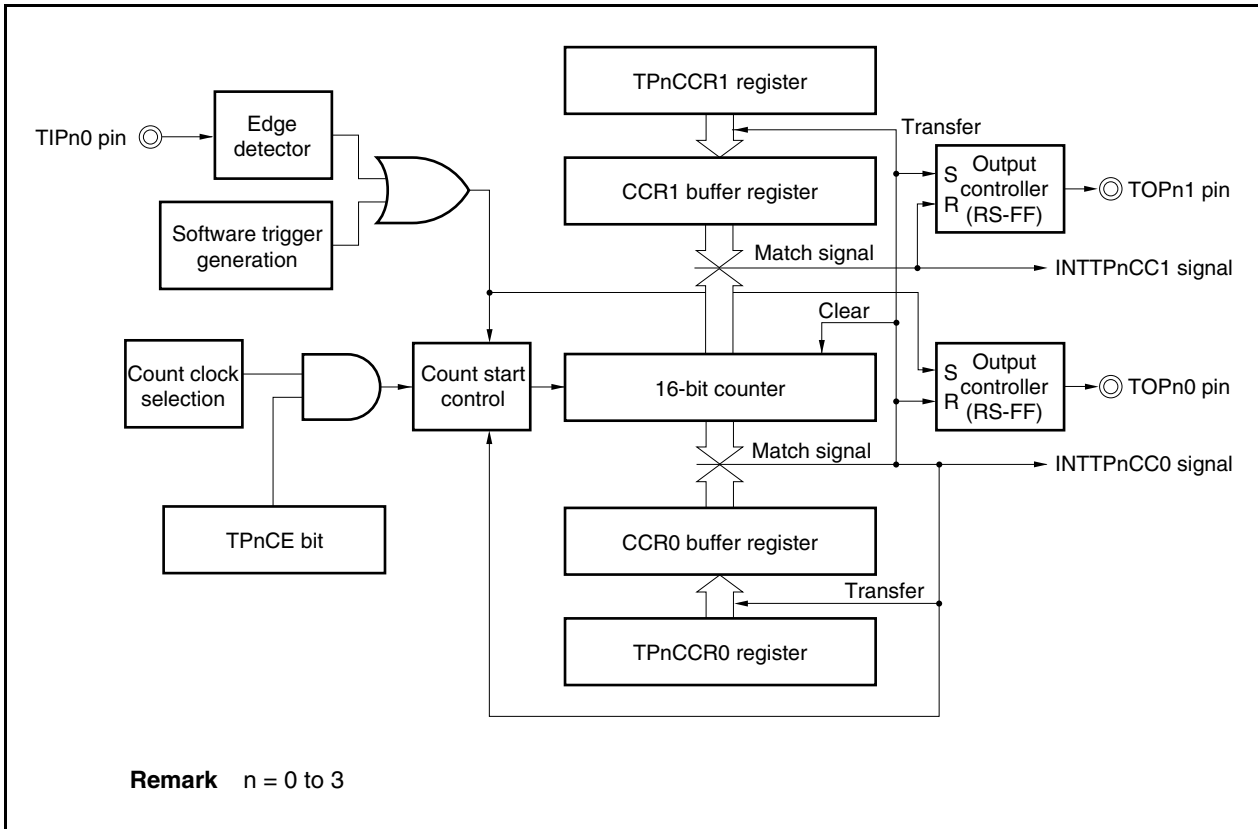
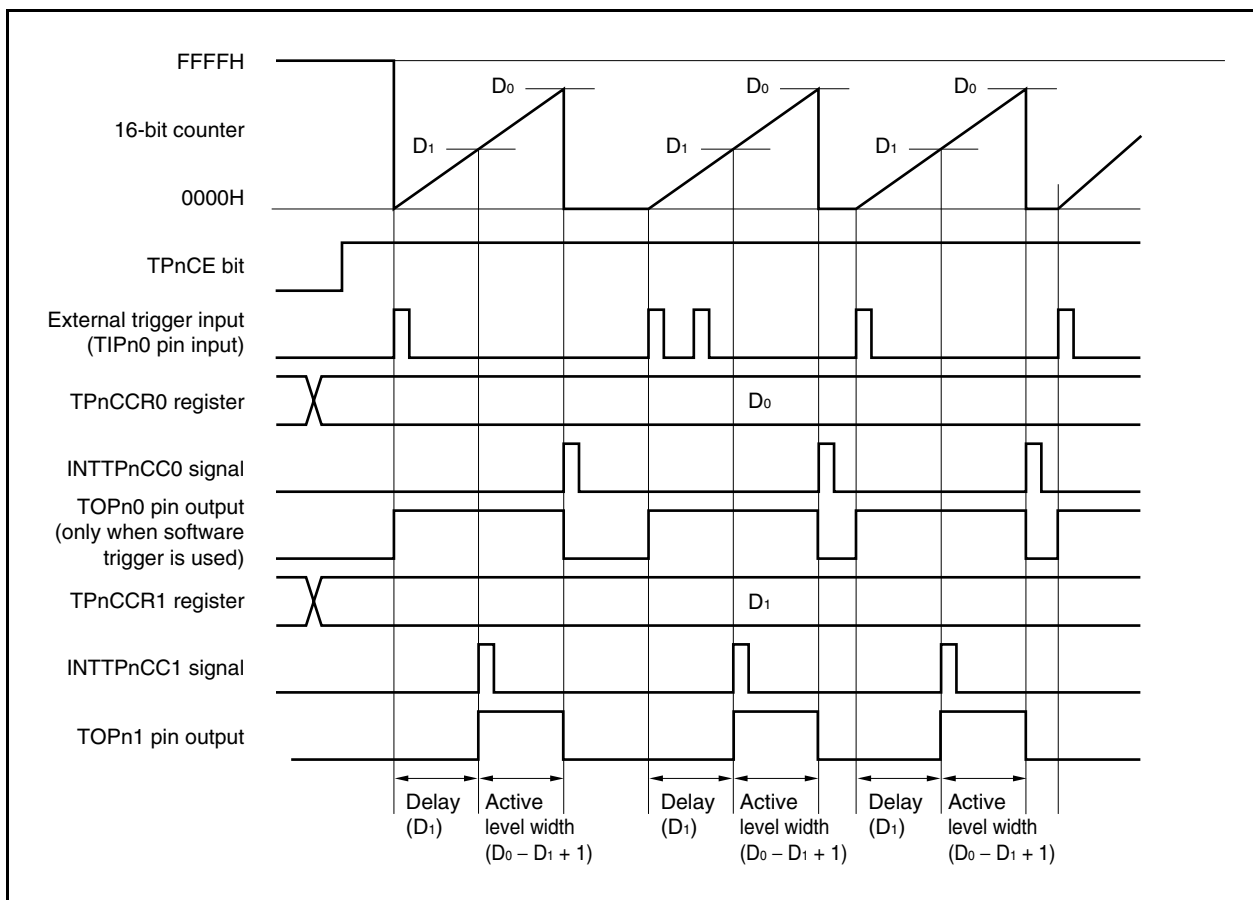


Figure 6-21. Basic Timing in One-Shot Pulse Output Mode



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.

$$\text{Output delay period} = (\text{Set value of TPnCCR1 register}) \times \text{Count clock cycle}$$

$$\text{Active level width} = (\text{Set value of TPnCCR0 register} - \text{Set value of TPnCCR1 register} + 1) \times \text{Count clock cycle}$$

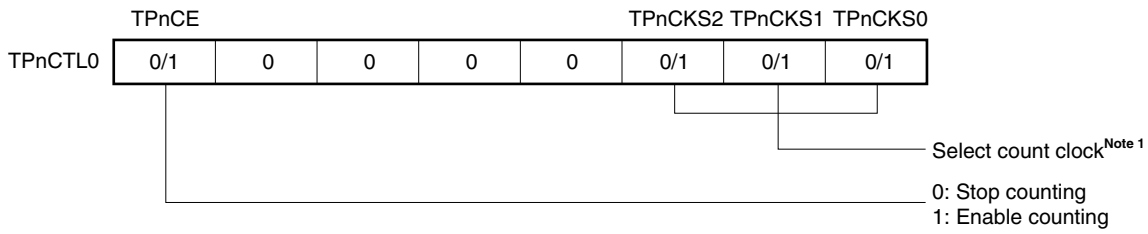
The compare match interrupt request signal INTTPnCC0 is generated when the 16-bit counter counts after its count value matches the value of the CCR0 buffer register. The compare match interrupt request signal INTTPnCC1 is generated when the count value of the 16-bit counter matches the value of the CCR1 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.

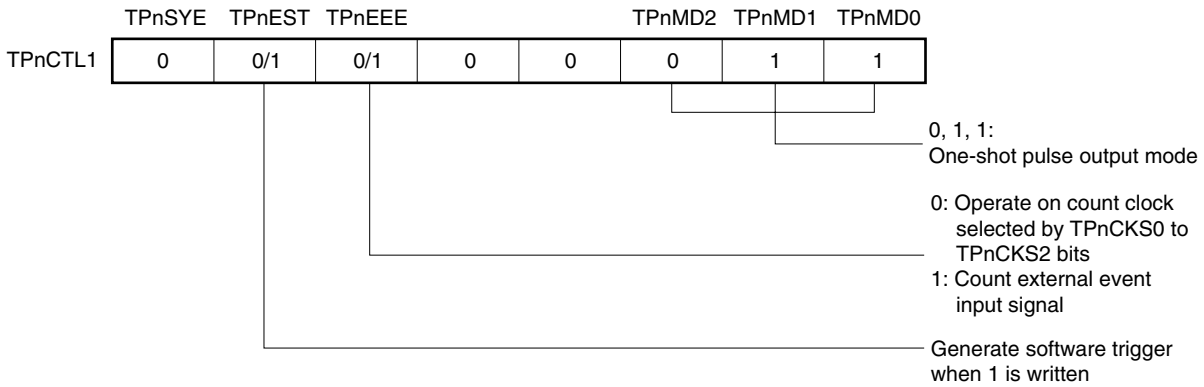
Remark n = 0 to 3
m = 0, 1

Figure 6-22. Setting of Registers in One-Shot Pulse Output Mode (1/2)

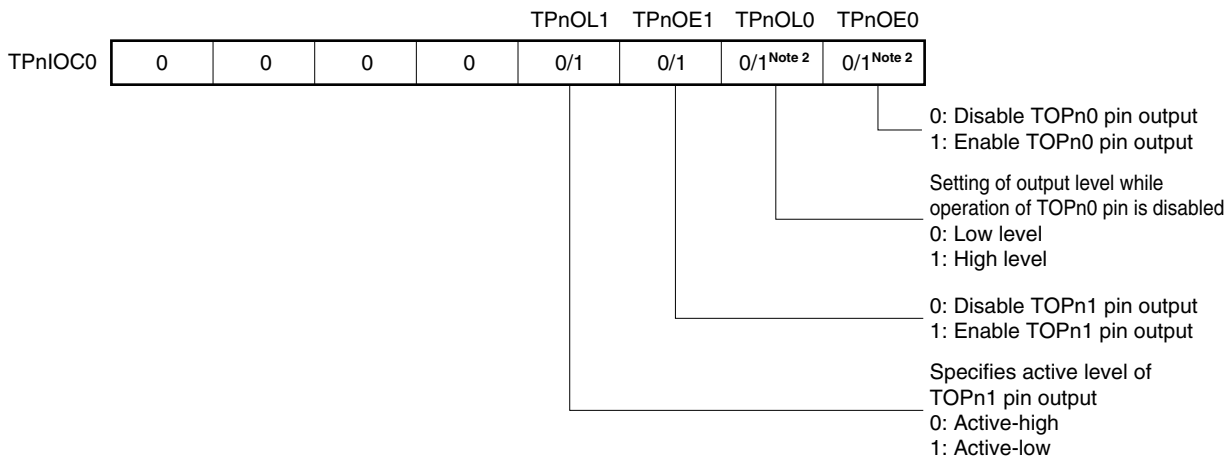
(a) TMPn control register 0 (TPnCTL0)



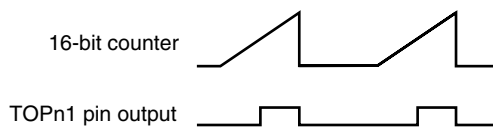
(b) TMPn control register 1 (TPnCTL1)



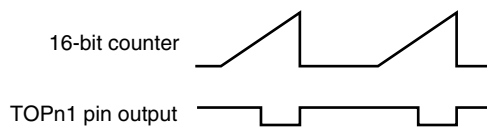
(c) TMPn I/O control register 0 (TPnIOC0)



• When TPnOL1 bit = 0



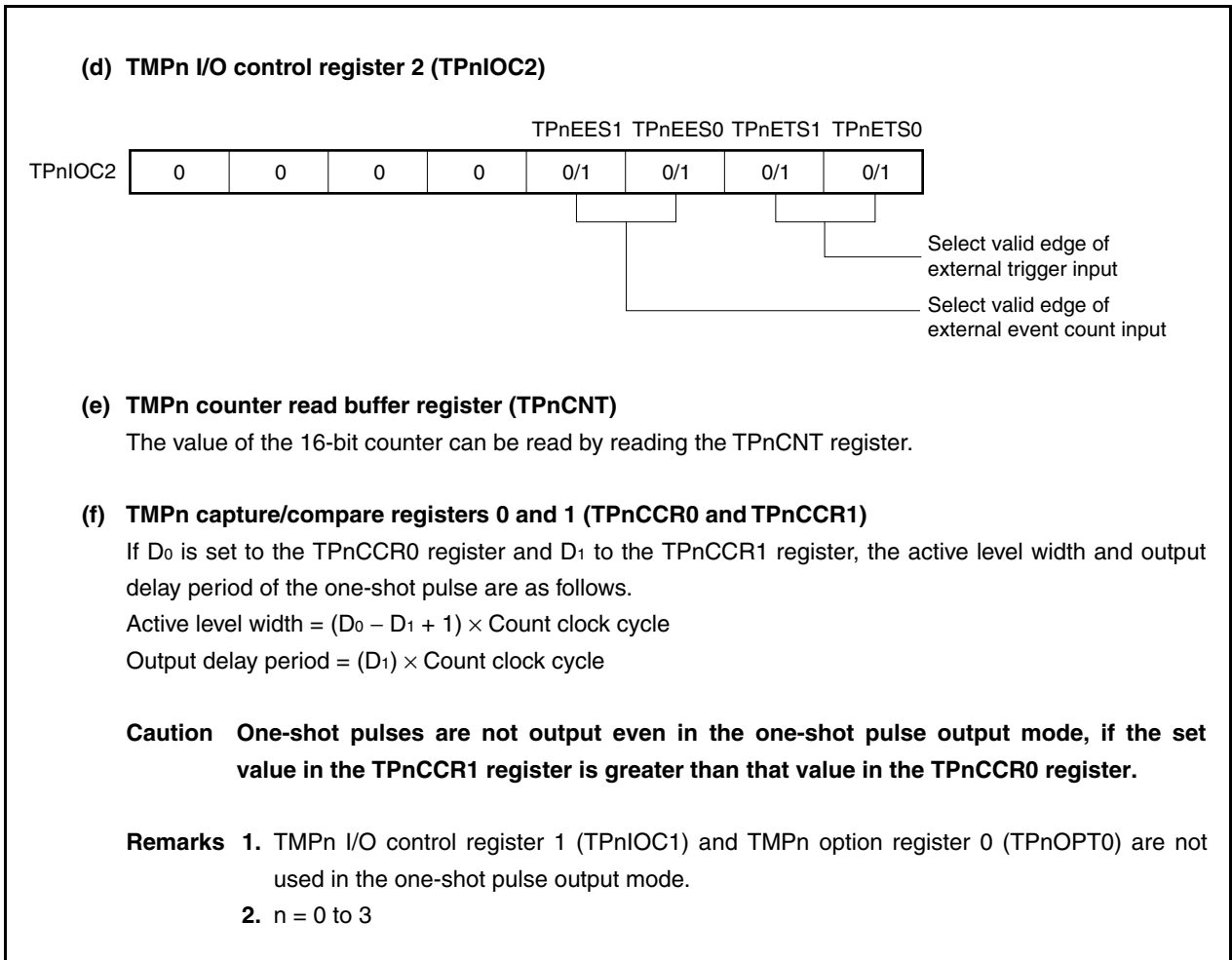
• When TPnOL1 bit = 1



Notes 1. The setting is invalid when the TPnCTL1.TPnEEE bit = 1.

2. Clear this bit to 0 when the TOPn0 pin is not used in the one-shot pulse output mode.

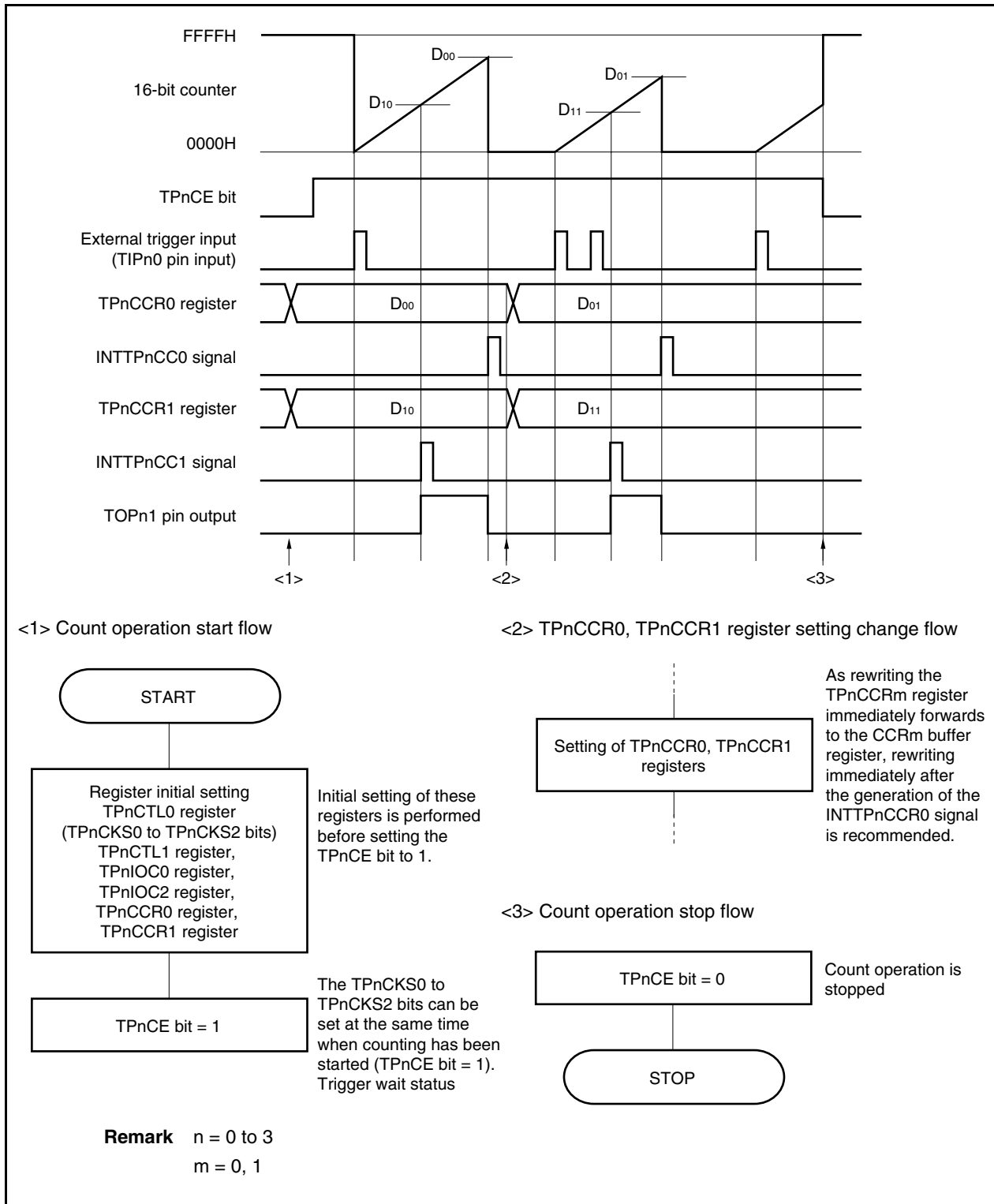
Figure 6-22. Setting of Registers in One-Shot Pulse Output Mode (2/2)



<R>

(1) Operation flow in one-shot pulse output mode

Figure 6-23. Software Processing Flow in One-Shot Pulse Output Mode

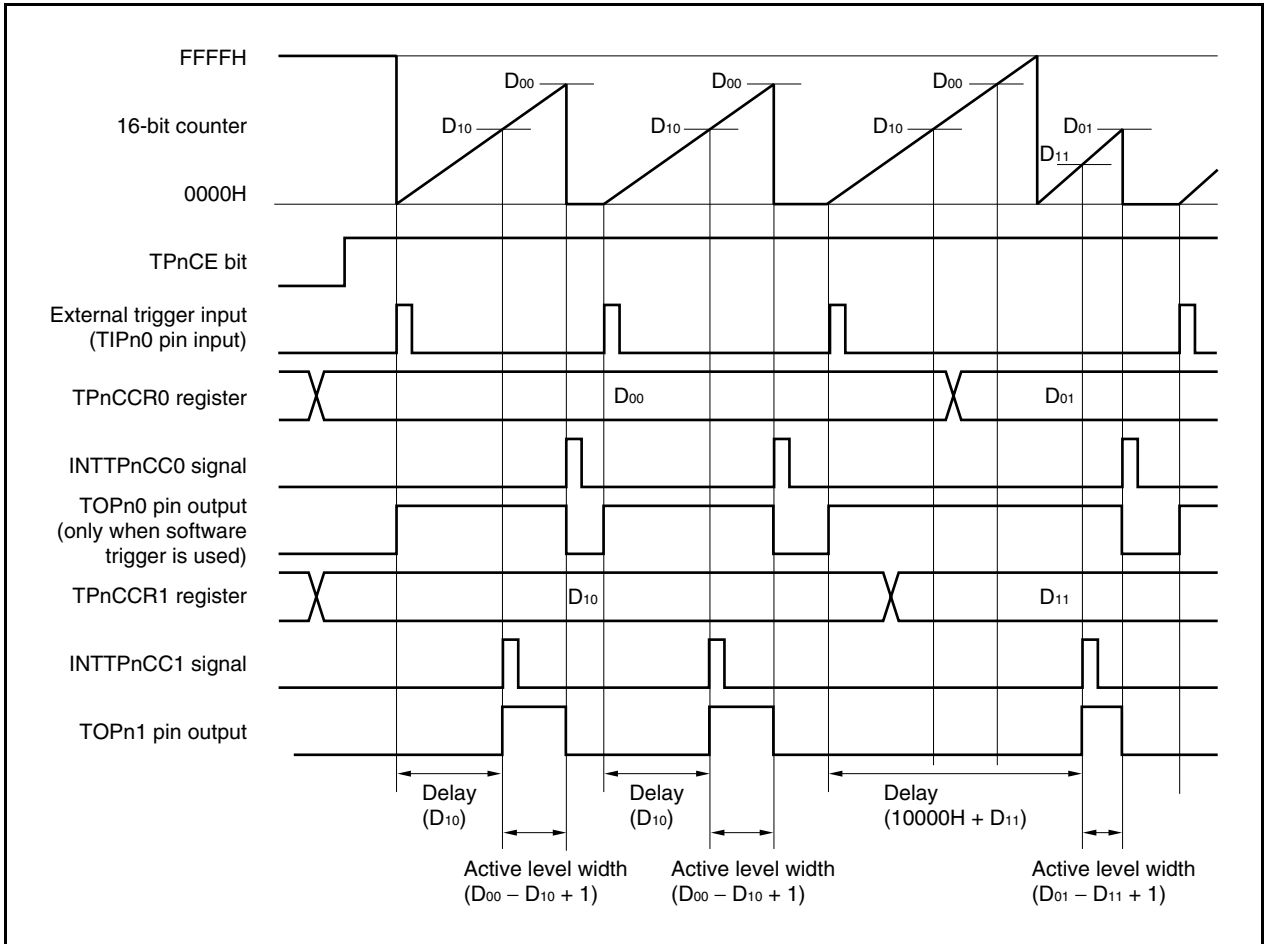


(2) Operation timing in one-shot pulse output mode

(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.



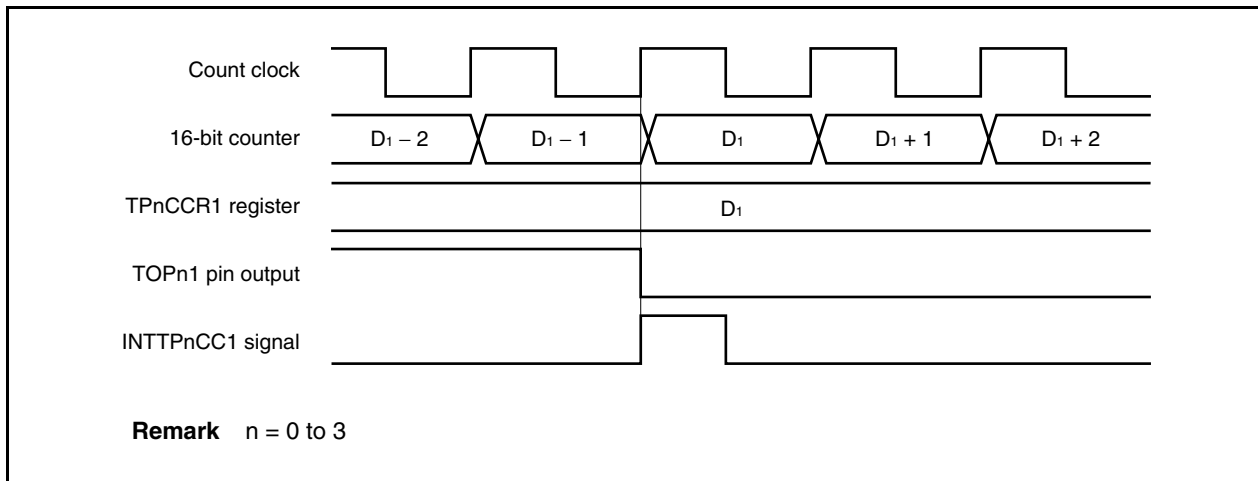
When the TPnCCR0 register is rewritten from D₀₀ to D₀₁ and the TPnCCR1 register from D₁₀ to D₁₁ where D₀₀ > D₀₁ and D₁₀ > D₁₁, if the TPnCCR1 register is rewritten when the count value of the 16-bit counter is greater than D₁₁ and less than D₁₀ and if the TPnCCR0 register is rewritten when the count value is greater than D₀₁ and less than D₀₀, 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 D₁₁, the counter generates the INTTPnCC1 signal and asserts the TOPn1 pin. When the count value matches D₀₁, 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.



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

6.5.5 PWM output mode (TPnMD2 to TPnMD0 bits = 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 6-24. Configuration in PWM Output Mode

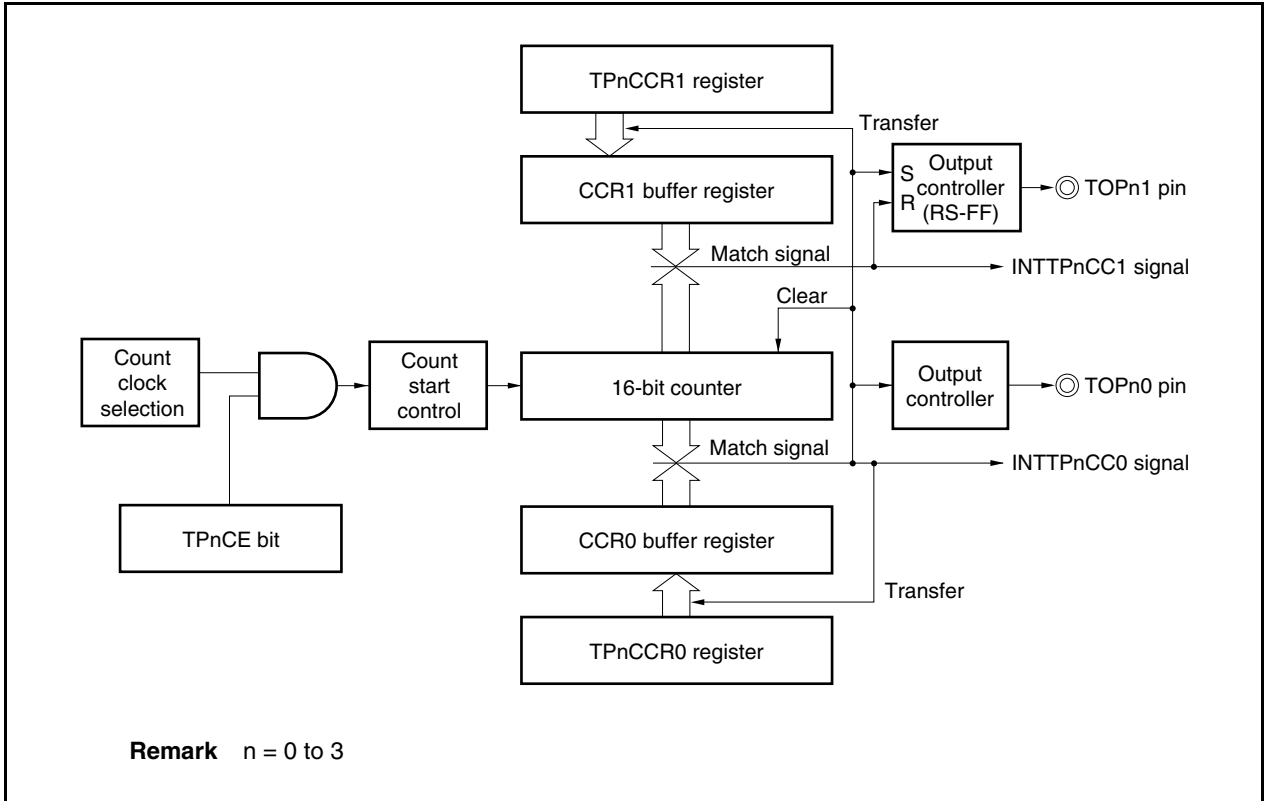
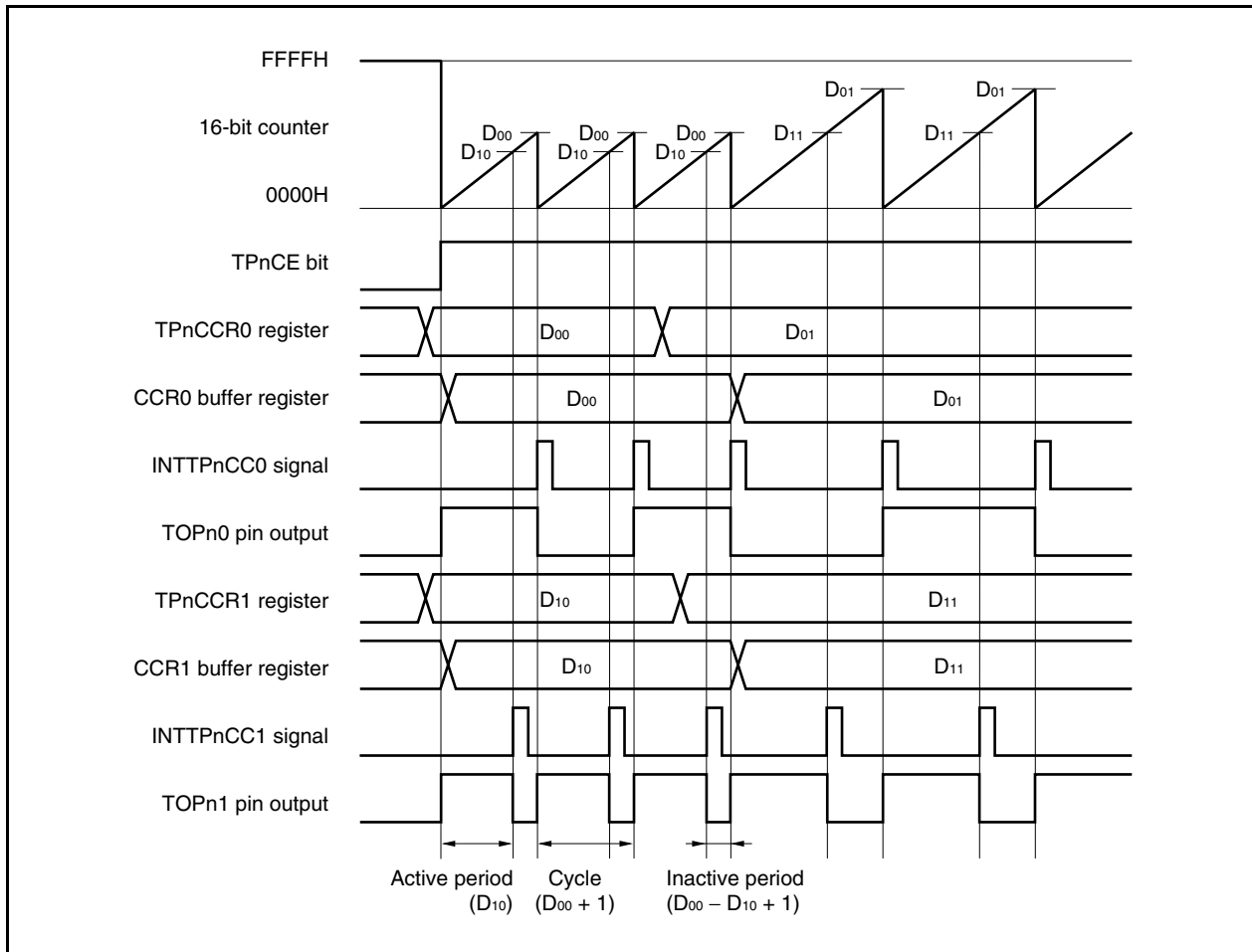


Figure 6-25. Basic Timing in PWM Output Mode



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.

$$\text{Active level width} = (\text{Set value of TPnCCR1 register}) \times \text{Count clock cycle}$$

$$\text{Cycle} = (\text{Set value of TPnCCR0 register} + 1) \times \text{Count clock cycle}$$

$$\text{Duty factor} = (\text{Set value of TPnCCR1 register}) / (\text{Set value of TPnCCR0 register} + 1)$$

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 6-26. Setting of Registers in PWM Output Mode (1/2)

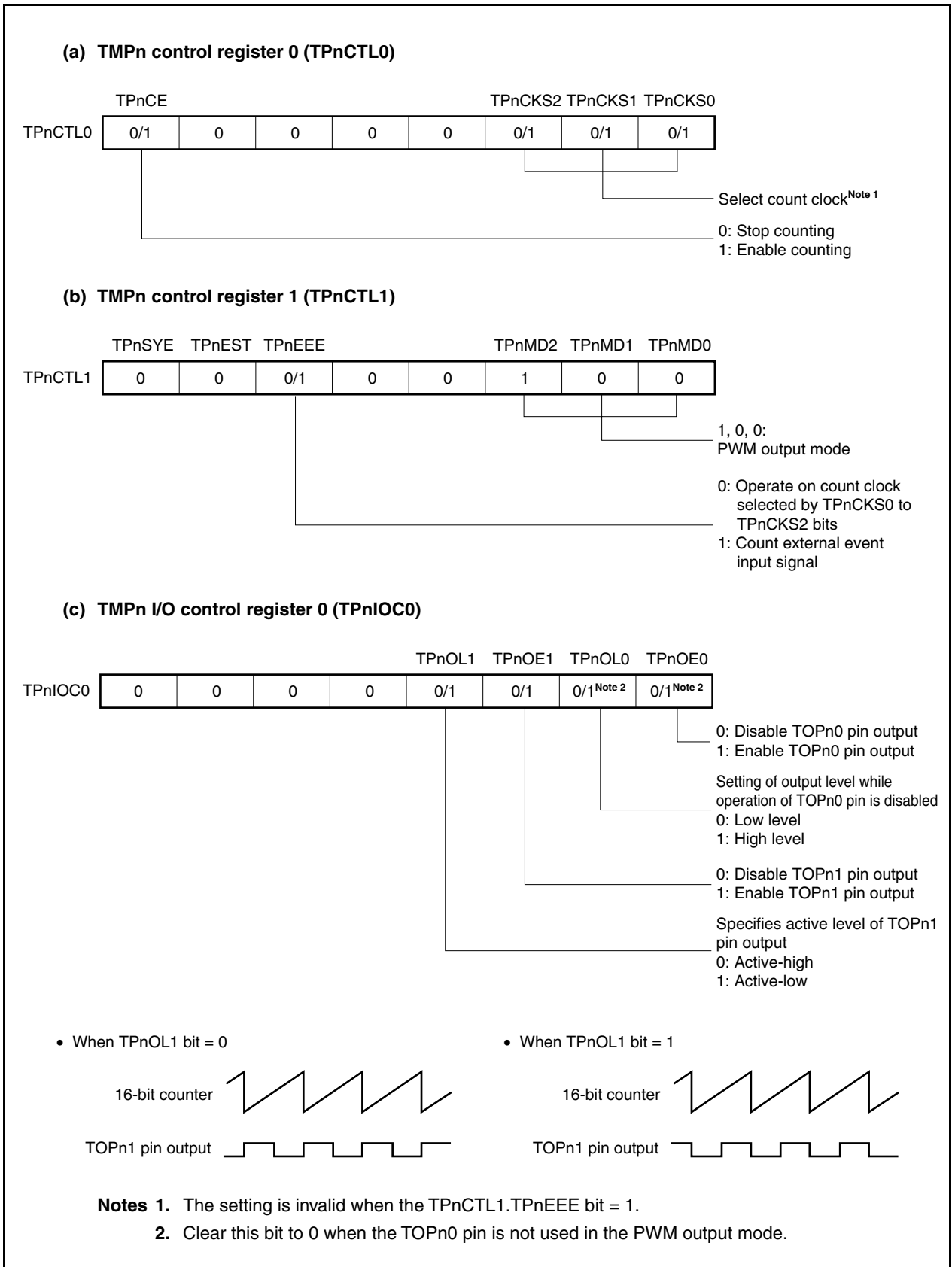
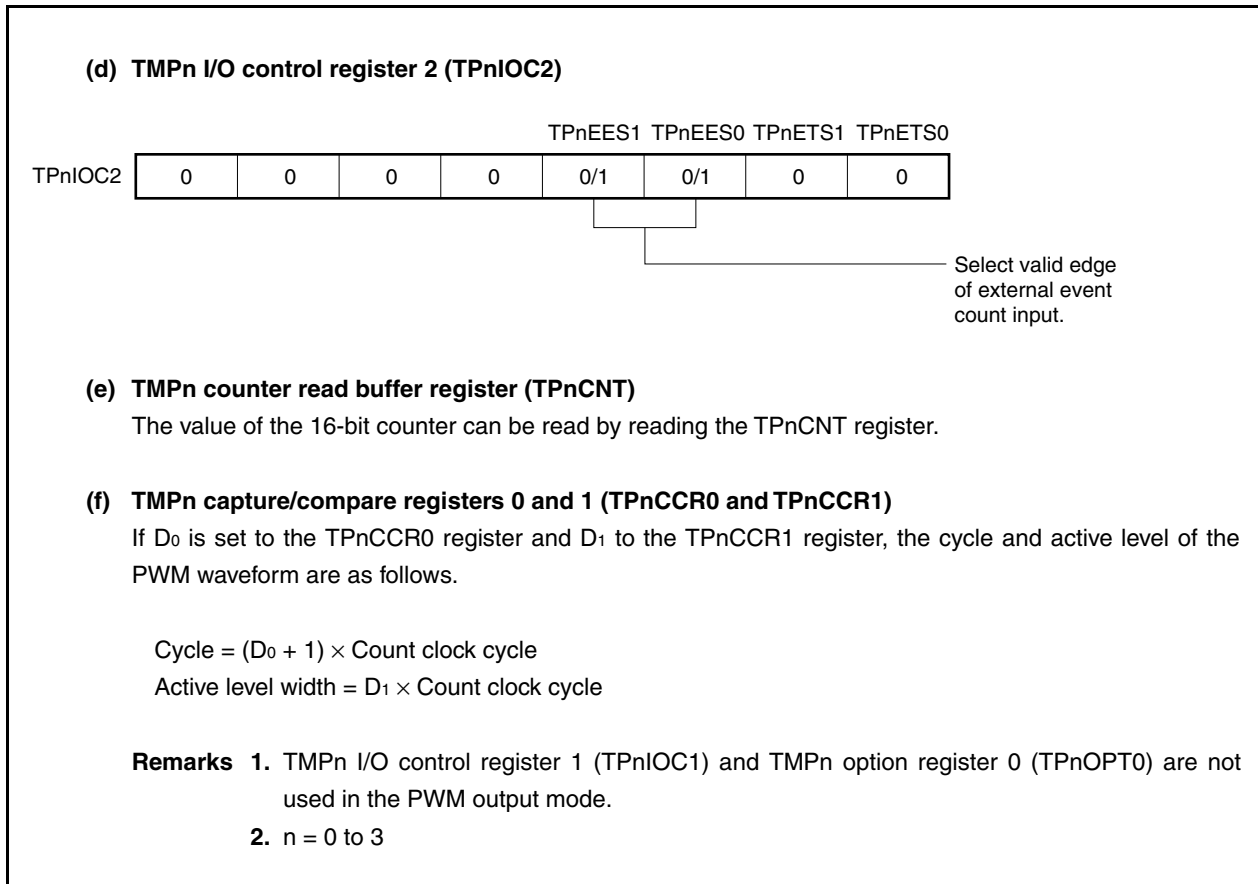


Figure 6-26. Register Setting in PWM Output Mode (2/2)



(1) Operation flow in PWM output mode

Figure 6-27. Software Processing Flow in PWM Output Mode (1/2)

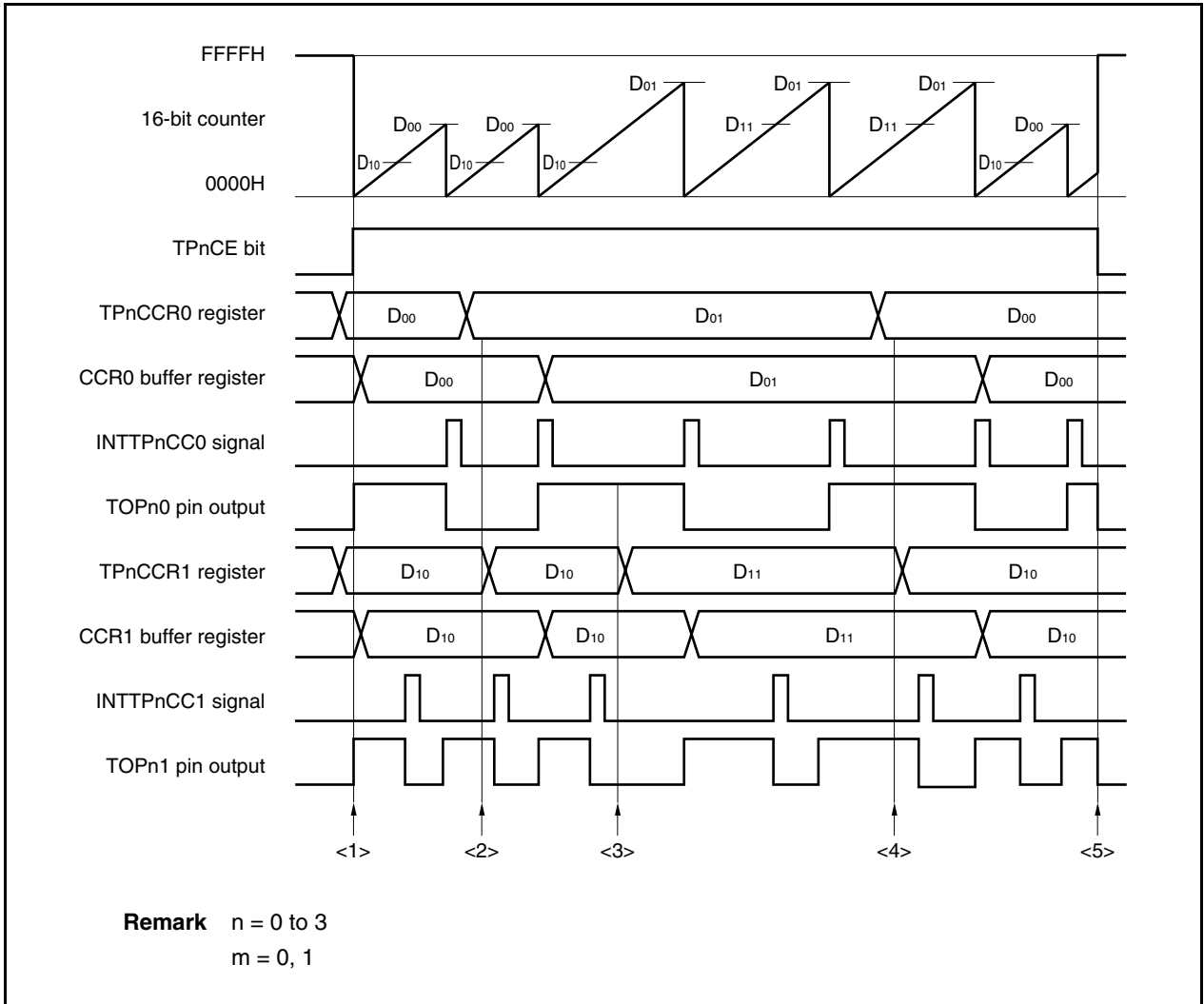
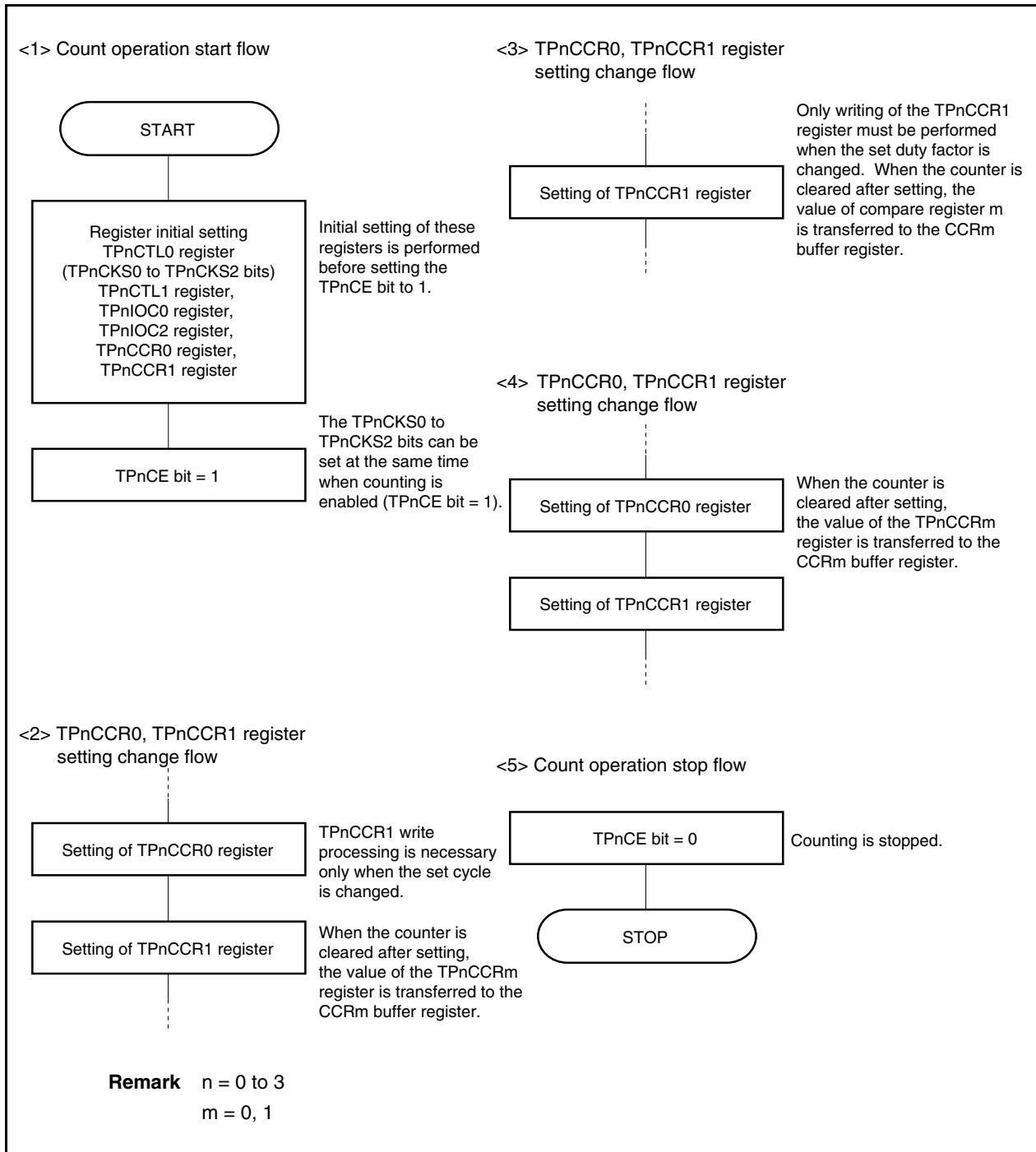


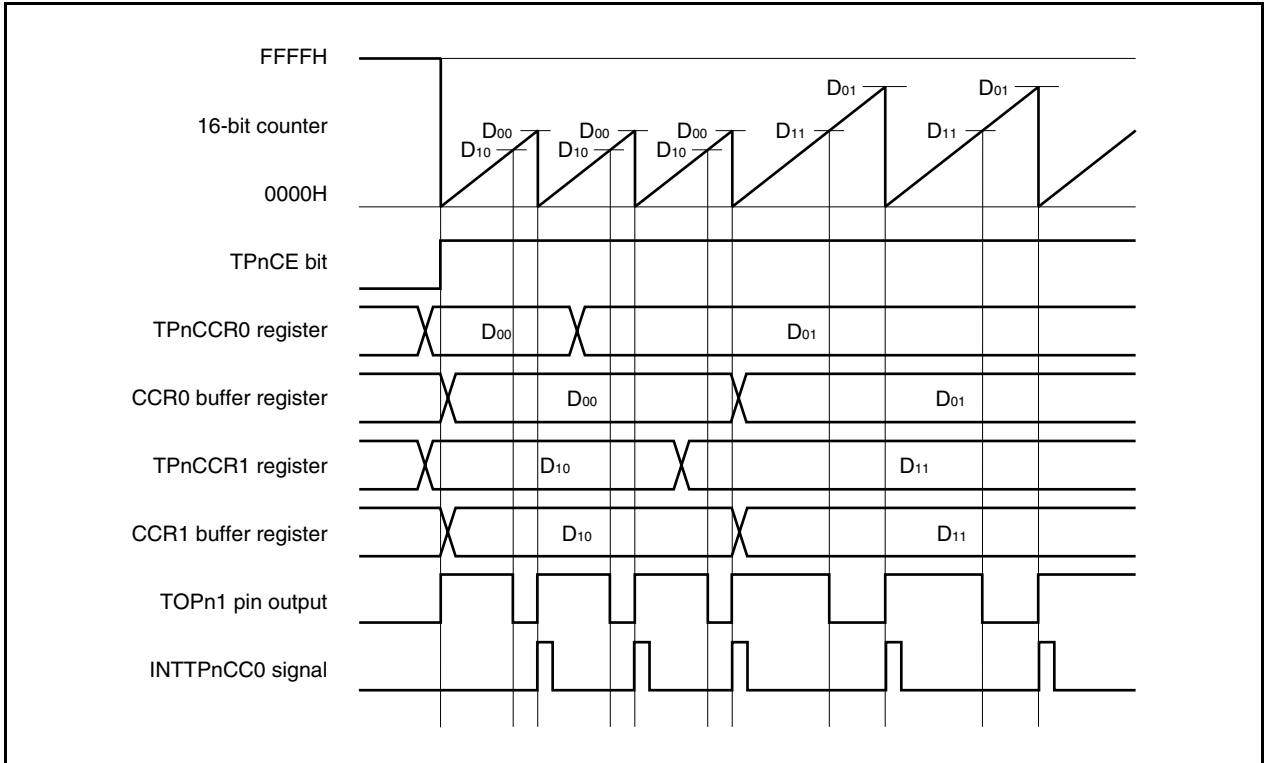
Figure 6-27. Software Processing Flow in PWM Output Mode (2/2)



(2) PWM output mode operation timing**(a) Changing pulse width during operation**

To change the PWM waveform while the counter is operating, write the TPnCCR1 register last.

Rewrite the TPnCCRm register after writing the TPnCCR1 register after the INTTPnCC1 signal is detected.



To transfer data from the TPnCCRm register to the CCRm buffer register, the TPnCCR1 register must be written.

To change both the cycle and active level of the PWM waveform at this time, first set the cycle to the TPnCCR0 register and then set the active level to the TPnCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TPnCCR0 register, and then write the same value to the TPnCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, only the TPnCCR1 register has to be set.

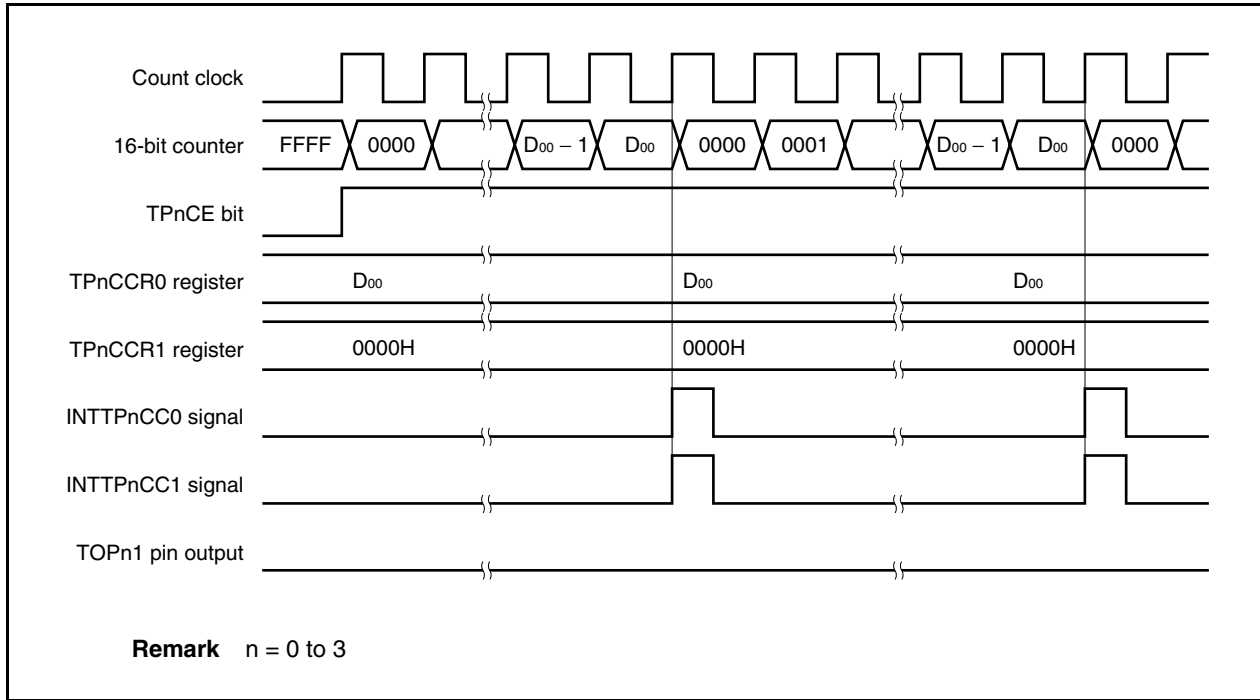
After data is written to the TPnCCR1 register, the value written to the TPnCCRm register is transferred to the CCRm buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TPnCCR0 or TPnCCR1 register again after writing the TPnCCR1 register once, do so after the INTTPnCC0 signal is generated. Otherwise, the value of the CCRm buffer register may become undefined because the timing of transferring data from the TPnCCRm register to the CCRm buffer register conflicts with writing the TPnCCRm register.

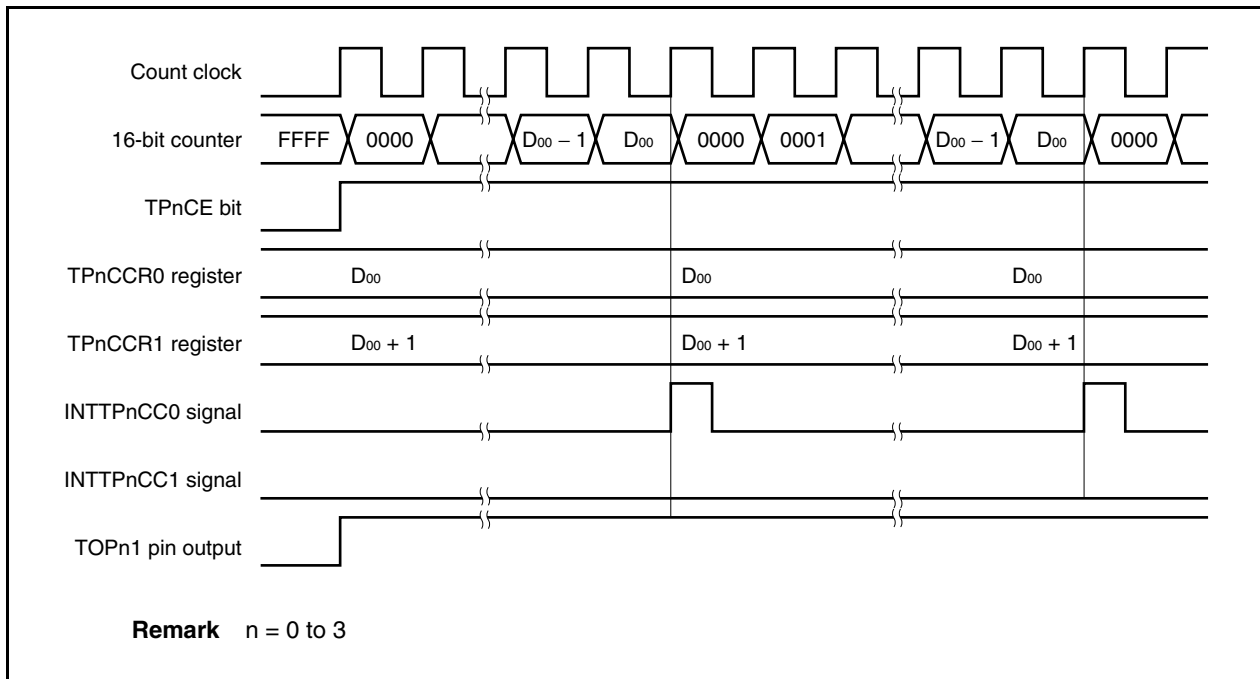
Remark n = 0 to 3, m = 0, 1

(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.

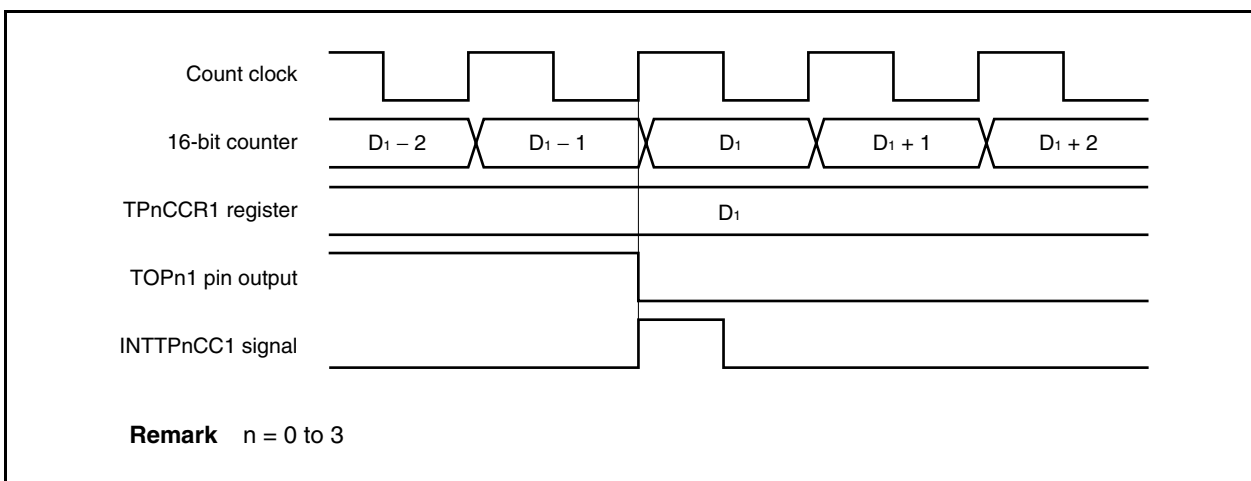


To output a 100% waveform, set a value of (set value of TPnCCR0 register + 1) to the TPnCCR1 register. If the set value of the TPnCCR0 register is FFFFH, 100% output cannot be produced.



(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.



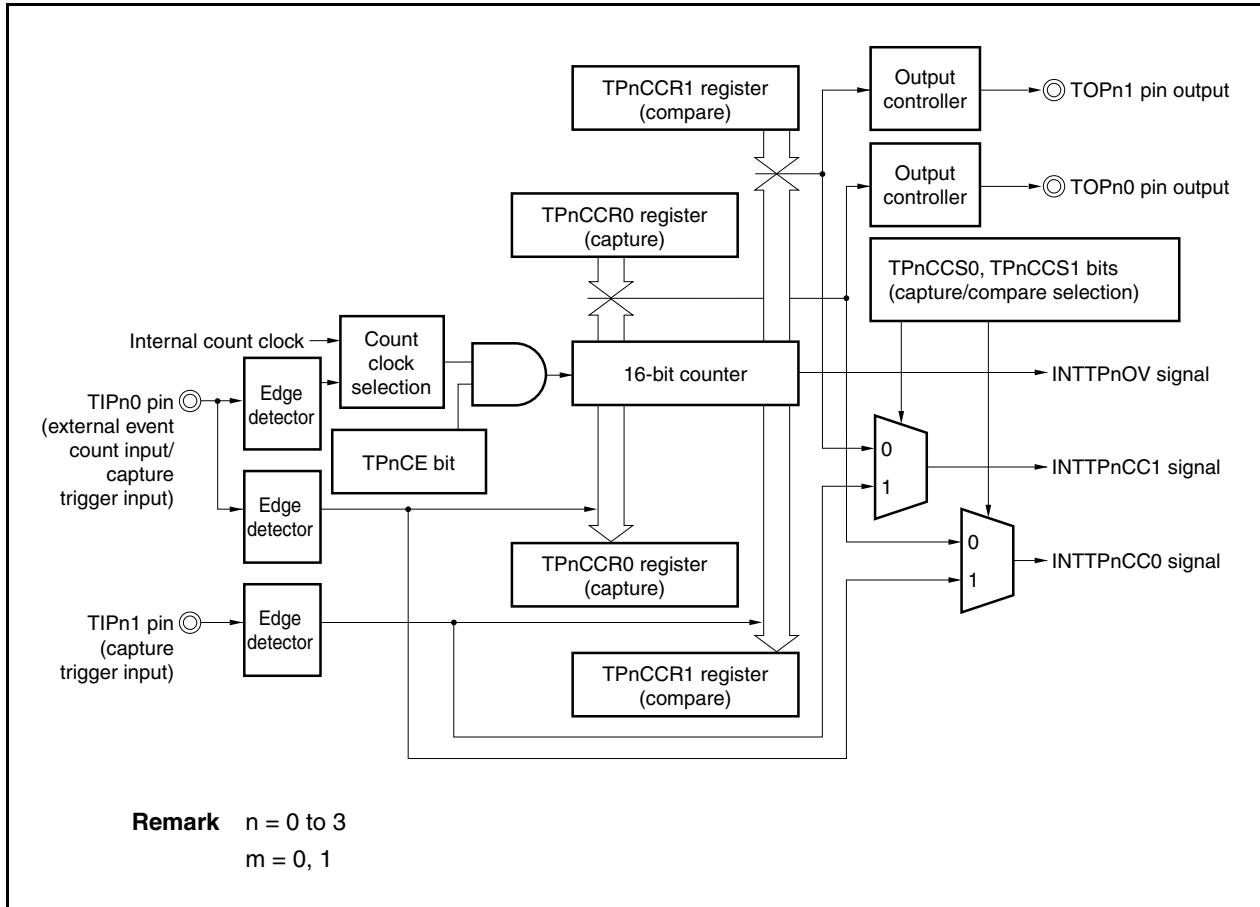
Usually, the INTTPnCC1 signal is generated in synchronization with the next counting up after the count value of the 16-bit counter matches the value of the TPnCCR1 register.

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.

6.5.6 Free-running timer mode (TPnMD2 to TPnMD0 bits = 101)

In the free-running timer mode, 16-bit timer/event counter P starts counting when the TPnCTL0.TPnCE bit is set to 1. At this time, the TPnCCRm register can be used as a compare register or a capture register, depending on the setting of the TPnOPT0.TPnCCS0 and TPnOPT0.TPnCCS1 bits.

Figure 6-28. Configuration in Free-Running Timer Mode

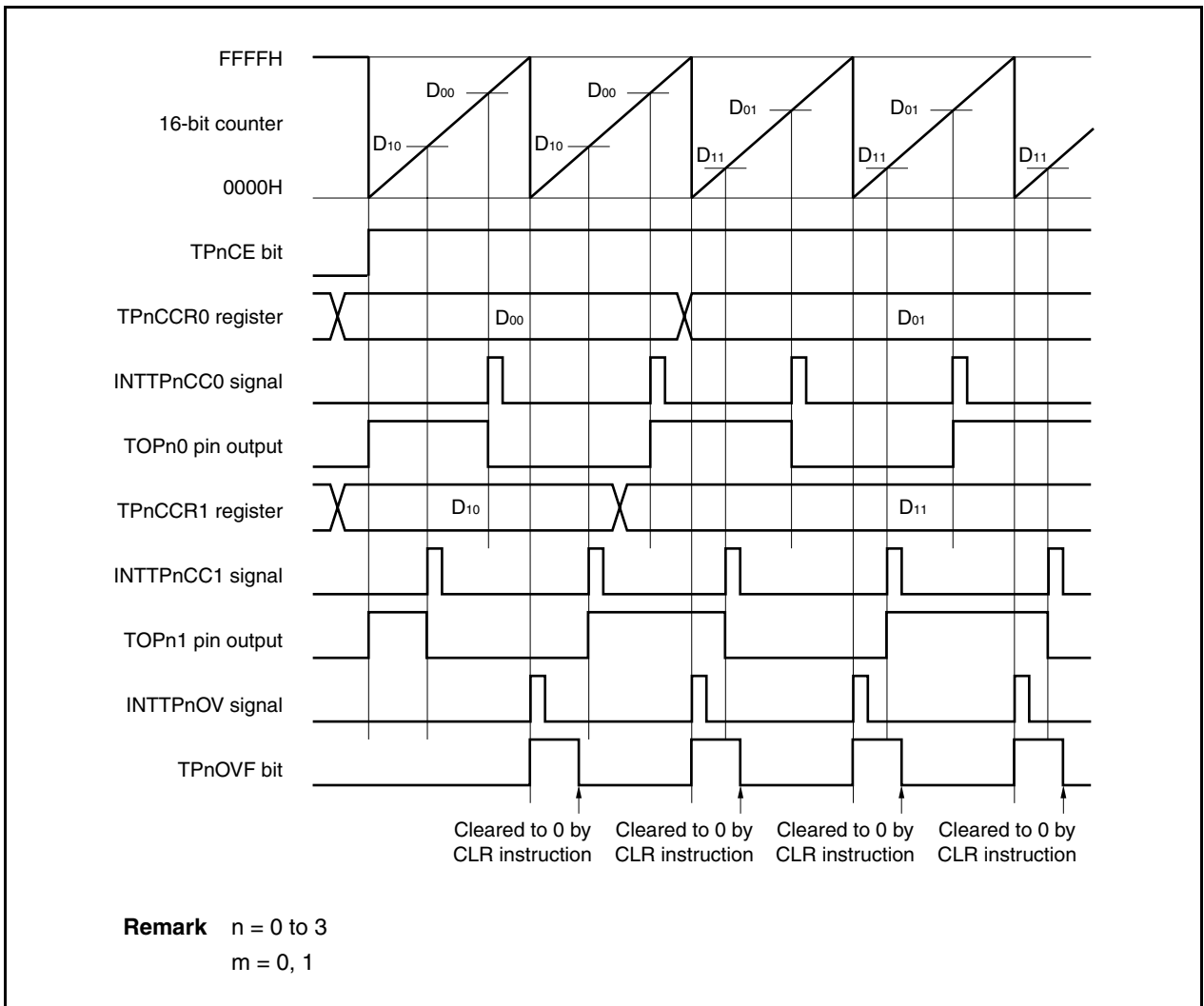


When the TPnCE bit is set to 1, 16-bit timer/event counter P starts counting, and the output signals of the TOPn0 and TOPn1 pins are inverted. When the count value of the 16-bit counter later matches the set value of the TPnCCRm register, a compare match interrupt request signal (INTTPnCCm) is generated, 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 6-29. Basic Timing in Free-Running Timer Mode (Compare Function)



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 6-30. Basic Timing in Free-Running Timer Mode (Capture Function)

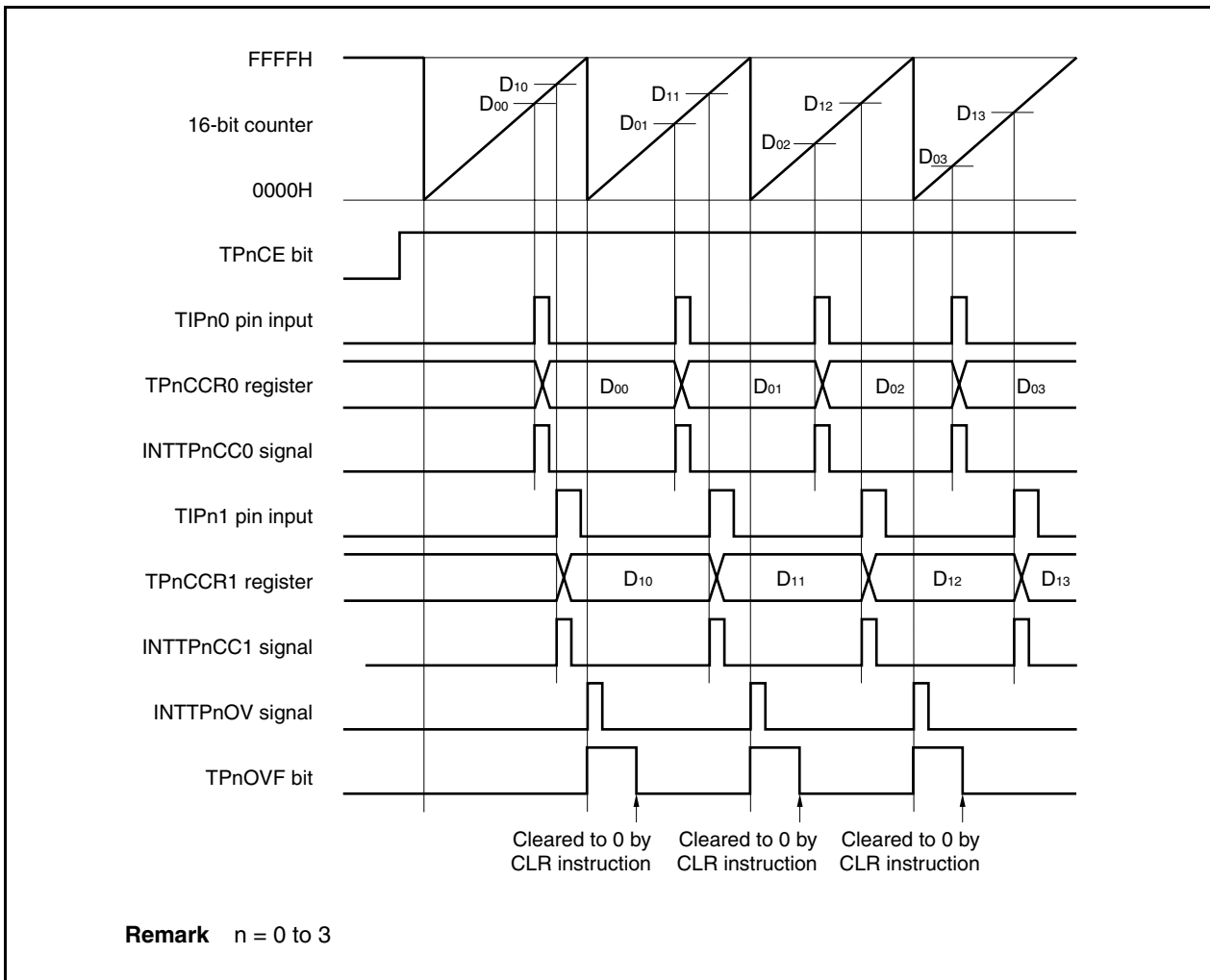


Figure 6-31. Register Setting in Free-Running Timer Mode (1/2)

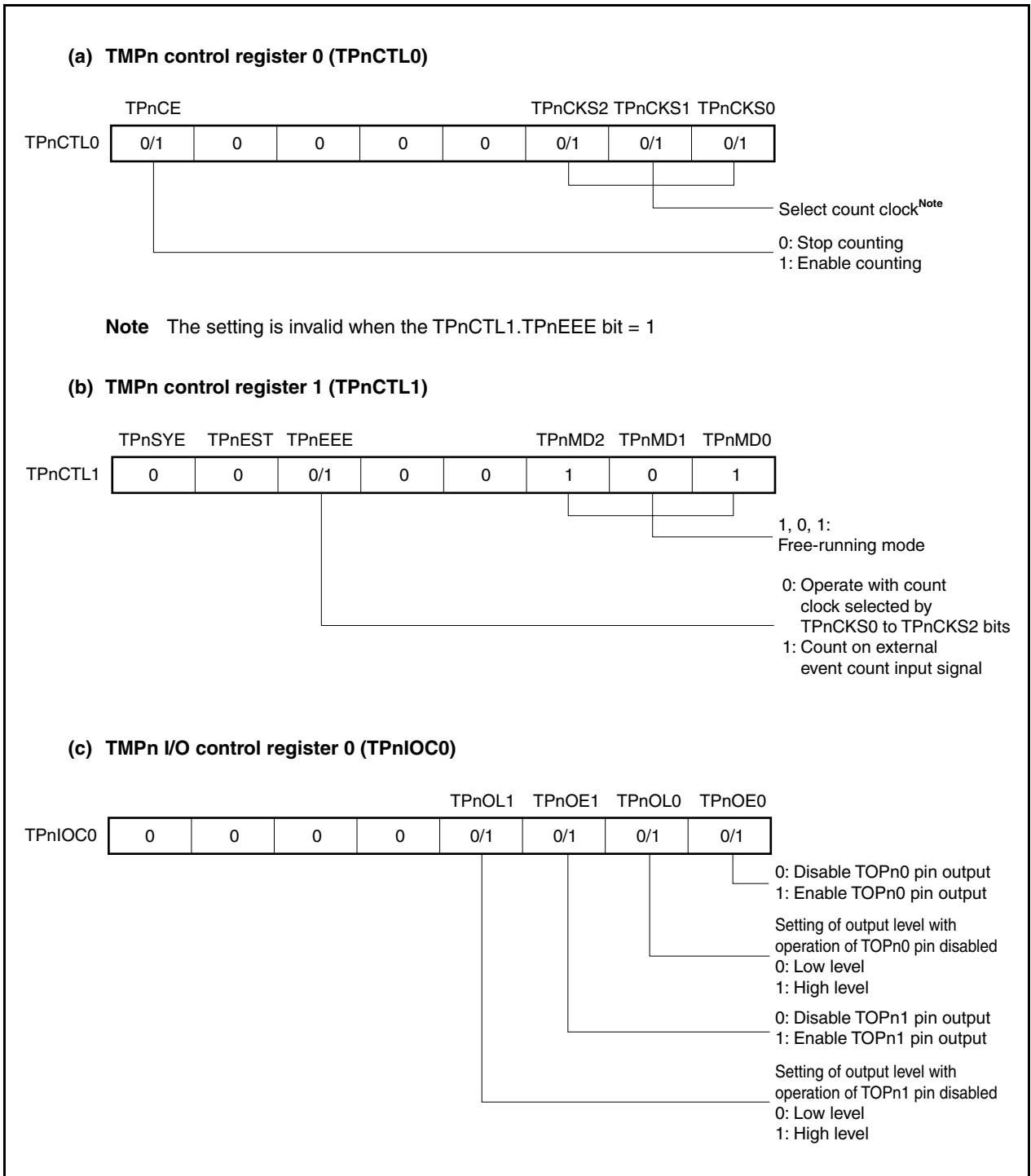
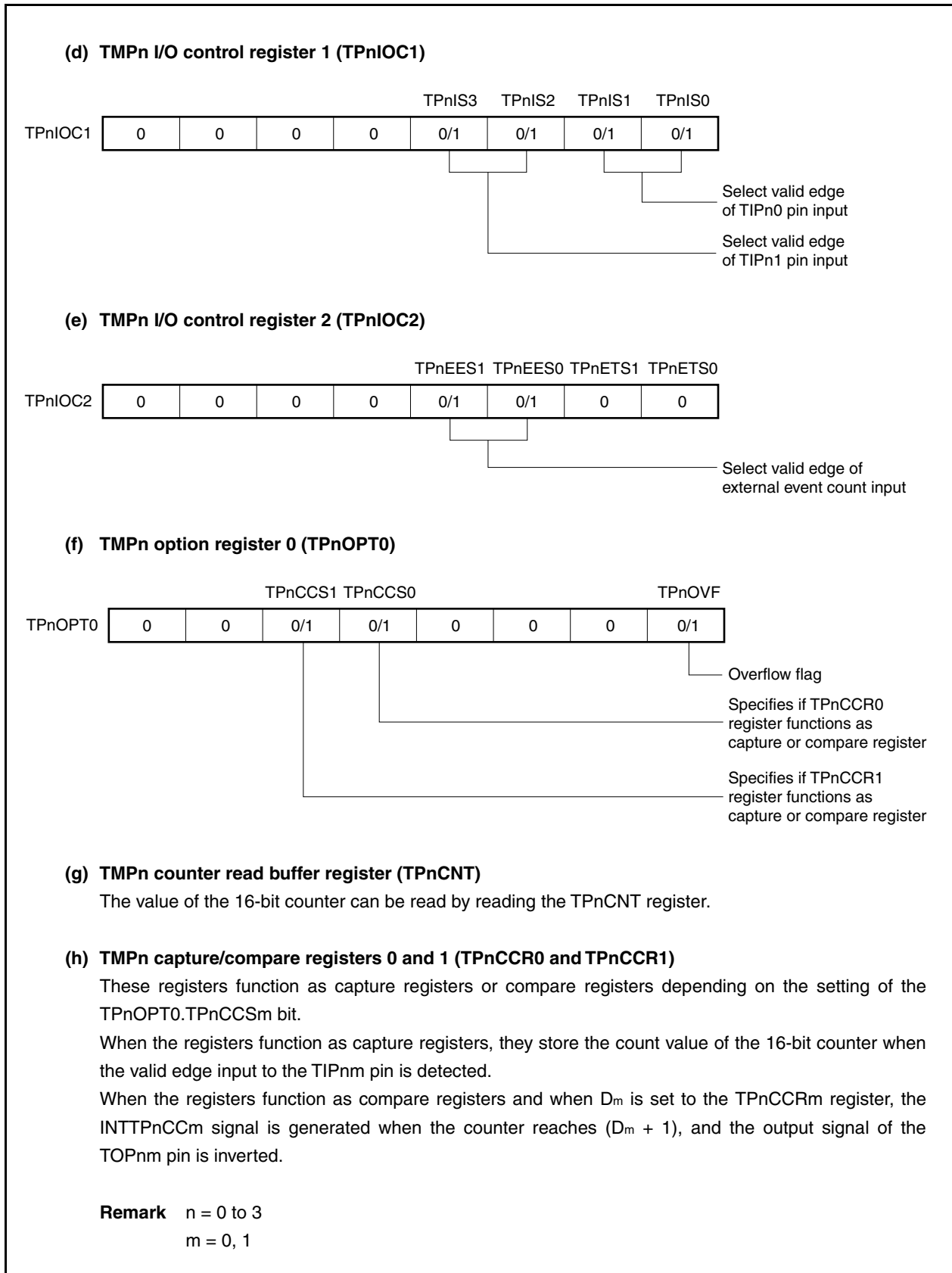


Figure 6-31. Register Setting in Free-Running Timer Mode (2/2)



(1) Operation flow in free-running timer mode

(a) When using capture/compare register as compare register

Figure 6-32. Software Processing Flow in Free-Running Timer Mode (Compare Function) (1/2)

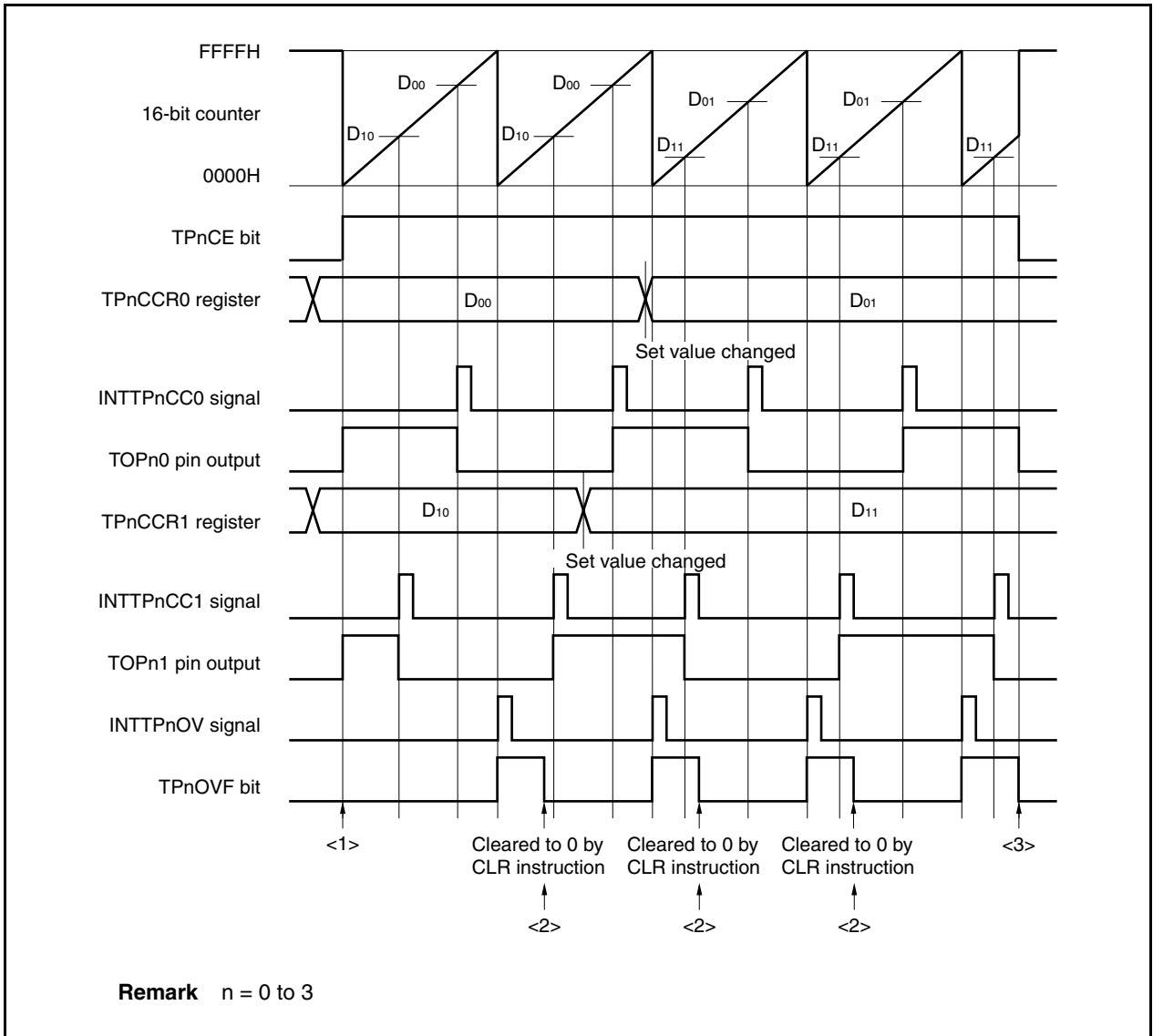
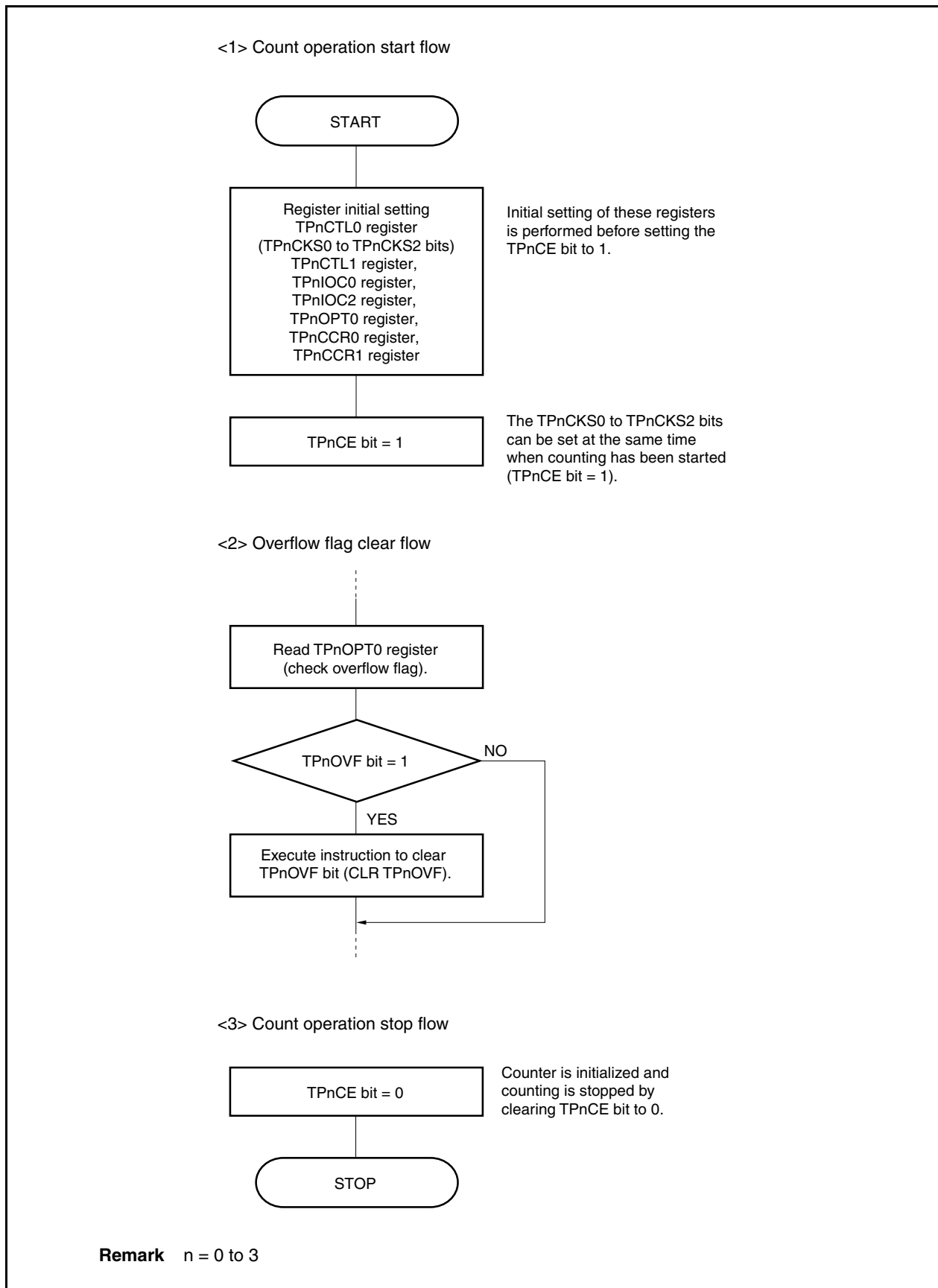


Figure 6-32. Software Processing Flow in Free-Running Timer Mode (Compare Function) (2/2)



(b) When using capture/compare register as capture register

Figure 6-33. Software Processing Flow in Free-Running Timer Mode (Capture Function) (1/2)

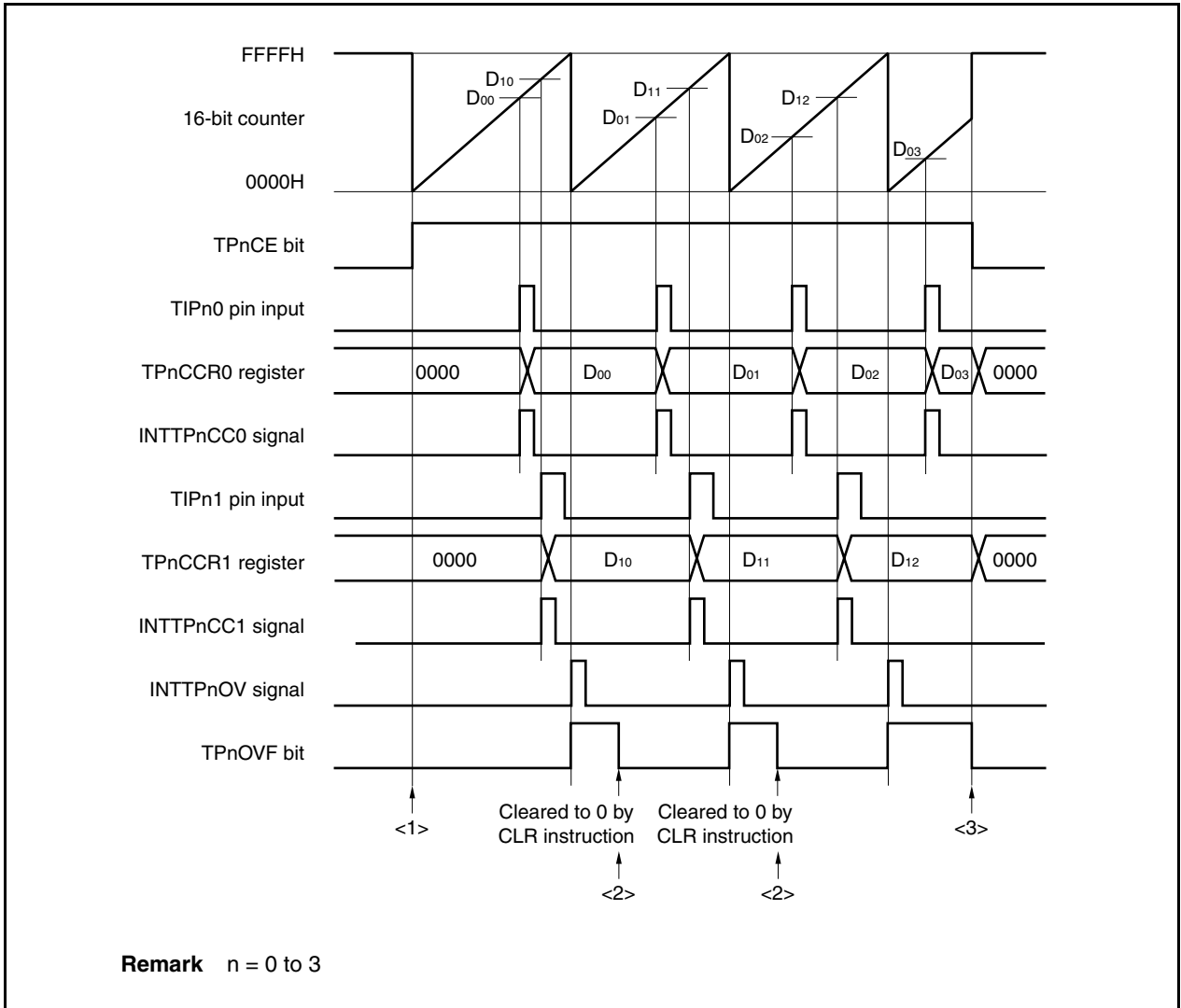
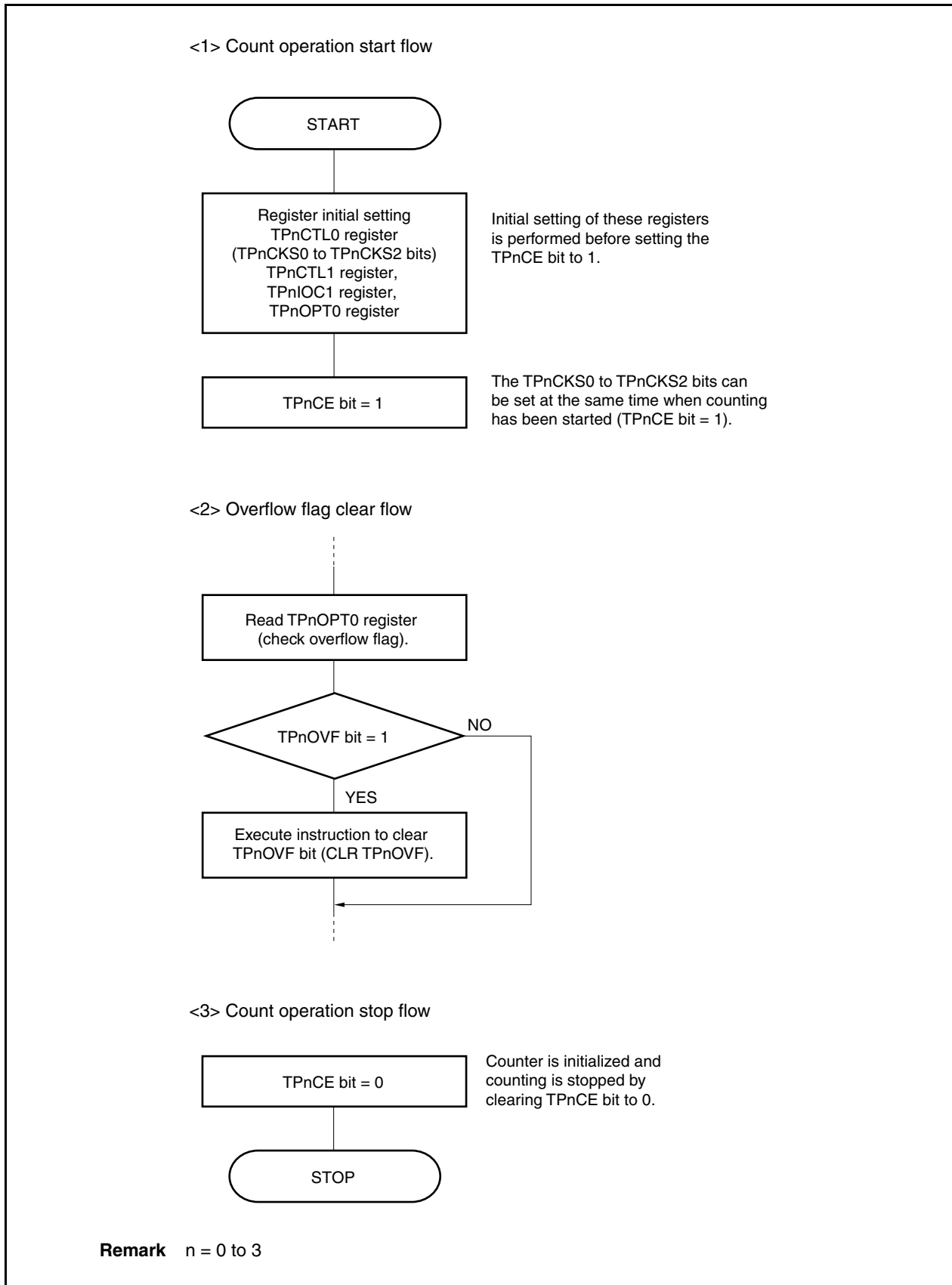


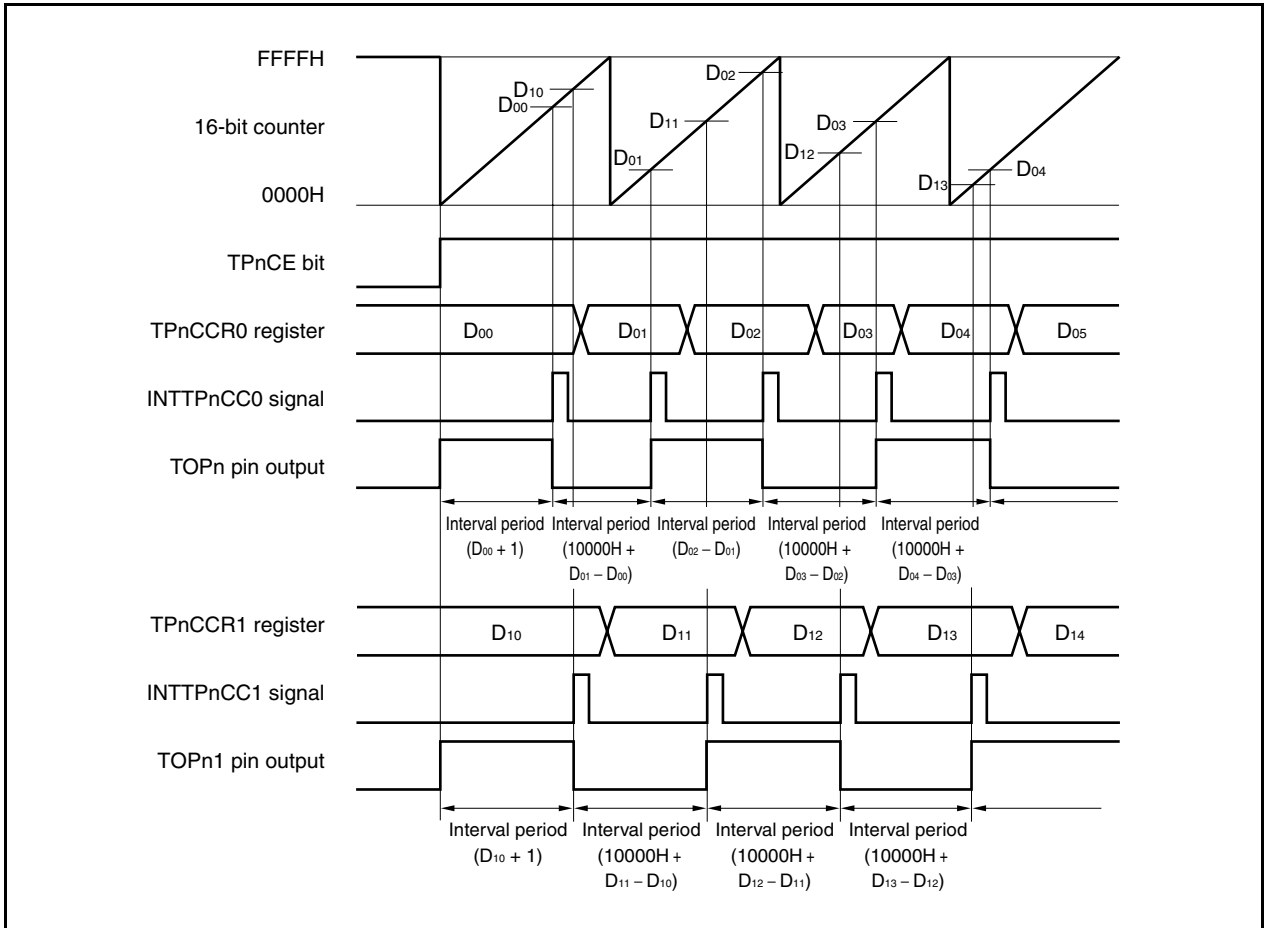
Figure 6-33. Software Processing Flow in Free-Running Timer Mode (Capture Function) (2/2)



(2) Operation timing in free-running timer mode

(a) Interval operation with compare register

When 16-bit timer/event counter 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

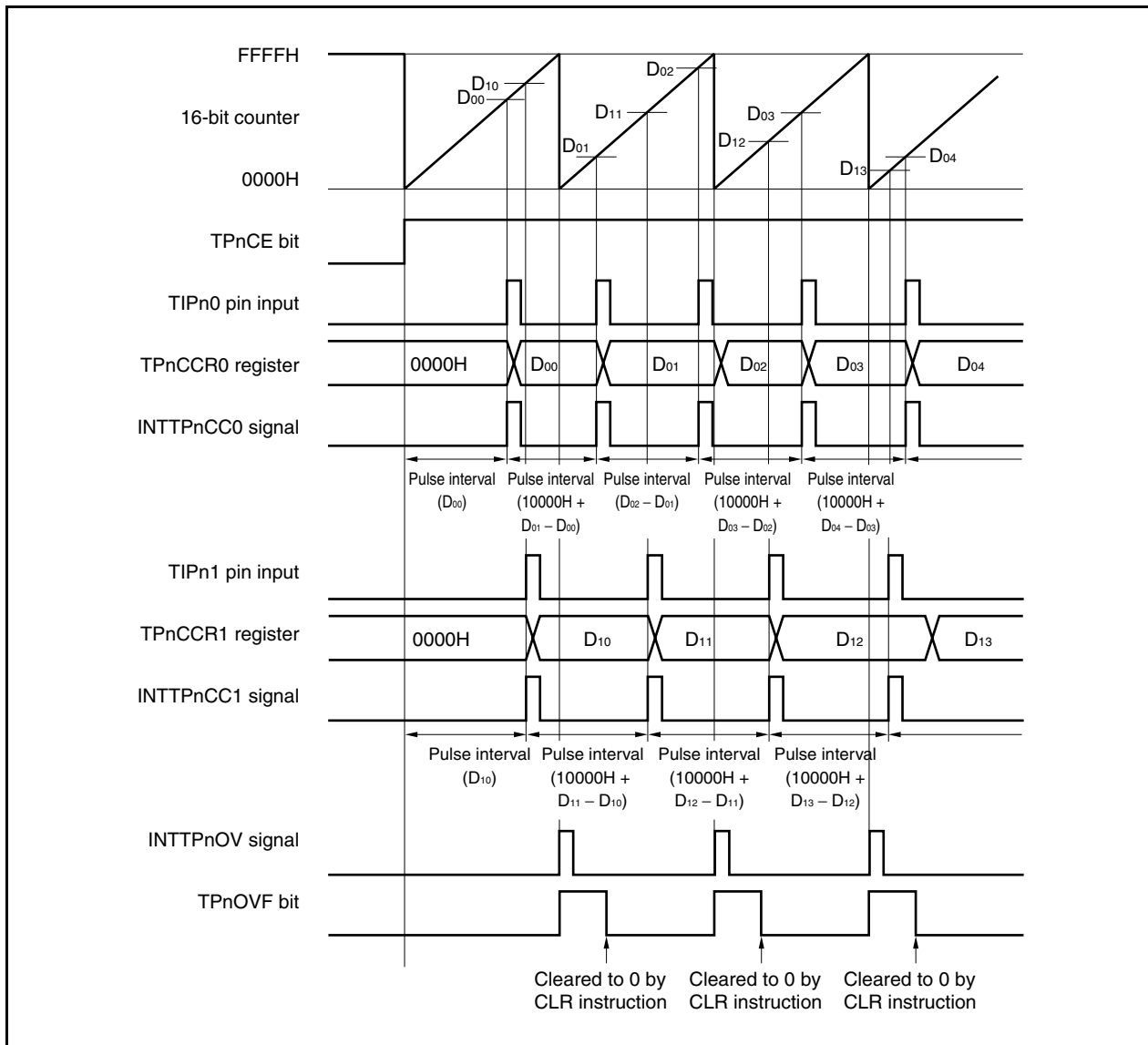
(If the calculation result is greater than FFFFH, subtract 10000H from the result and set this value to the register.)

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.



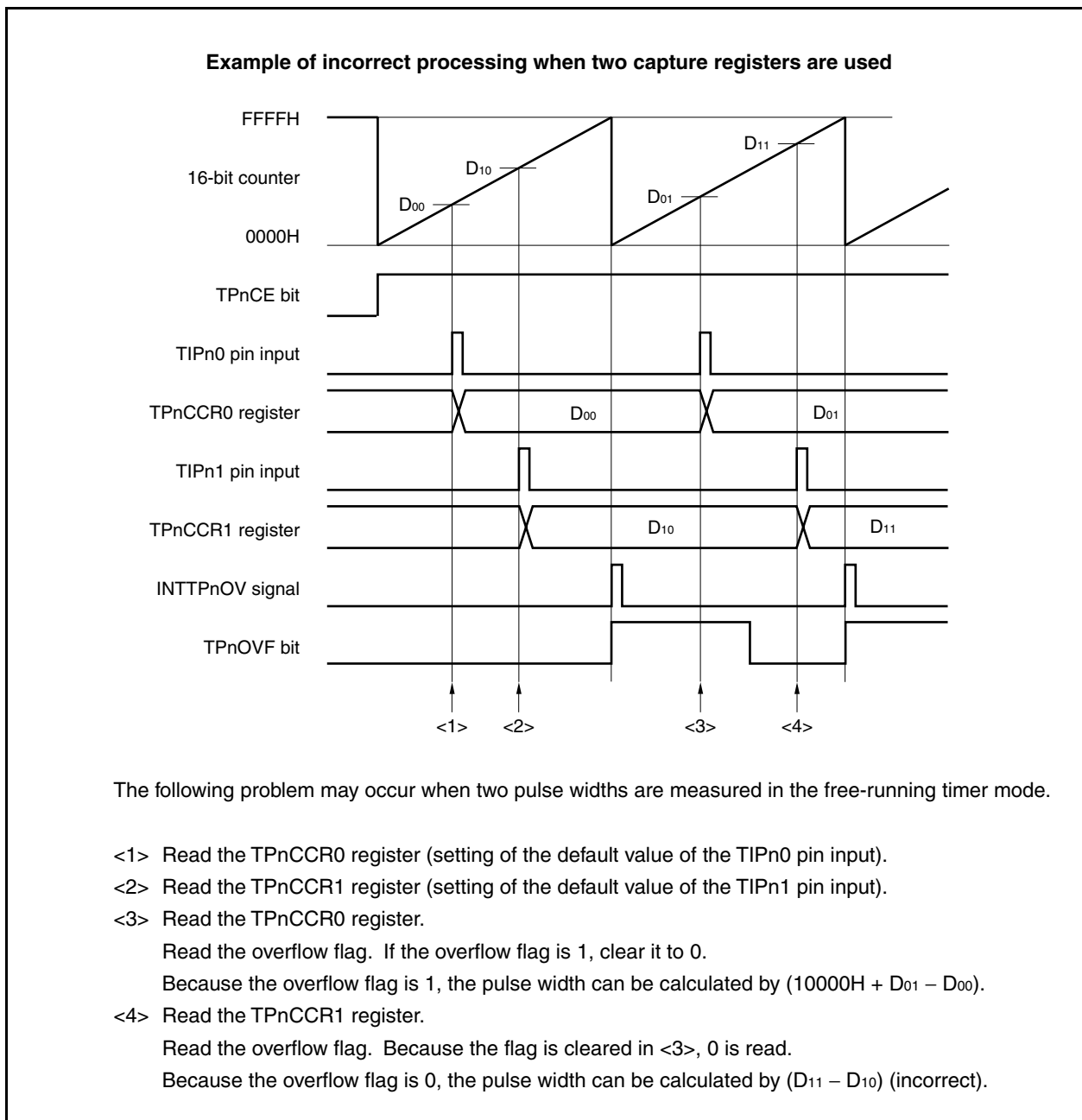
When executing pulse width measurement in the free-running timer mode, two pulse widths can be measured with one channel.

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

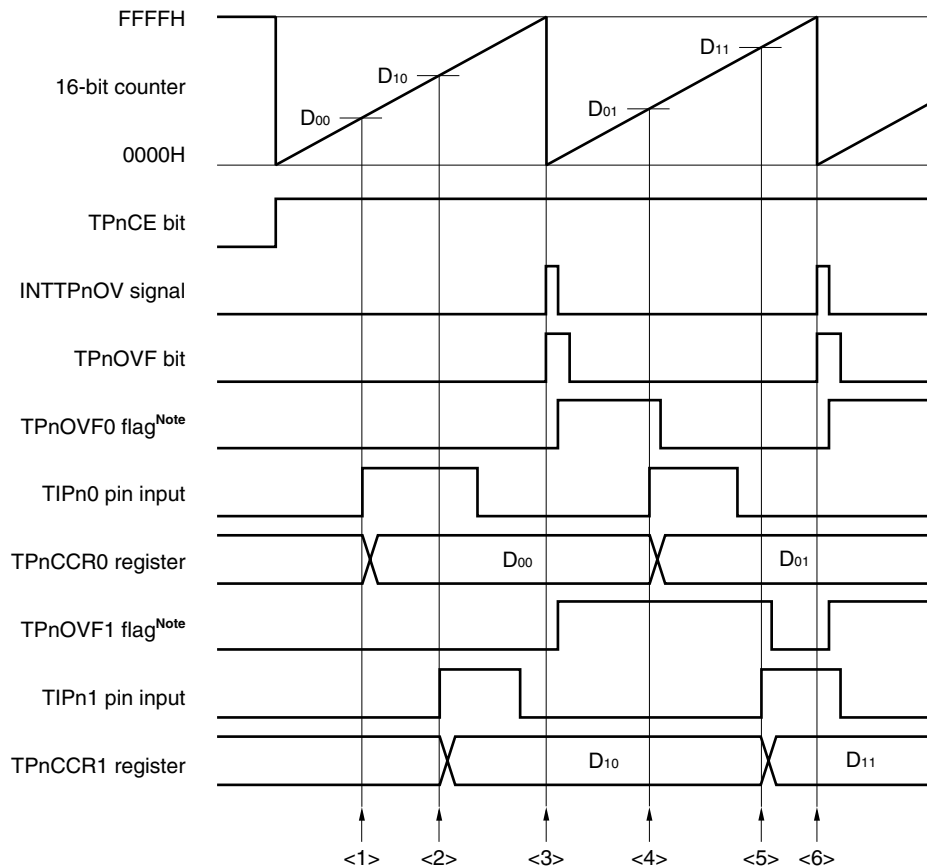
Care must be exercised in processing the overflow flag when two capture registers are used. First, an example of incorrect processing is shown below.



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.

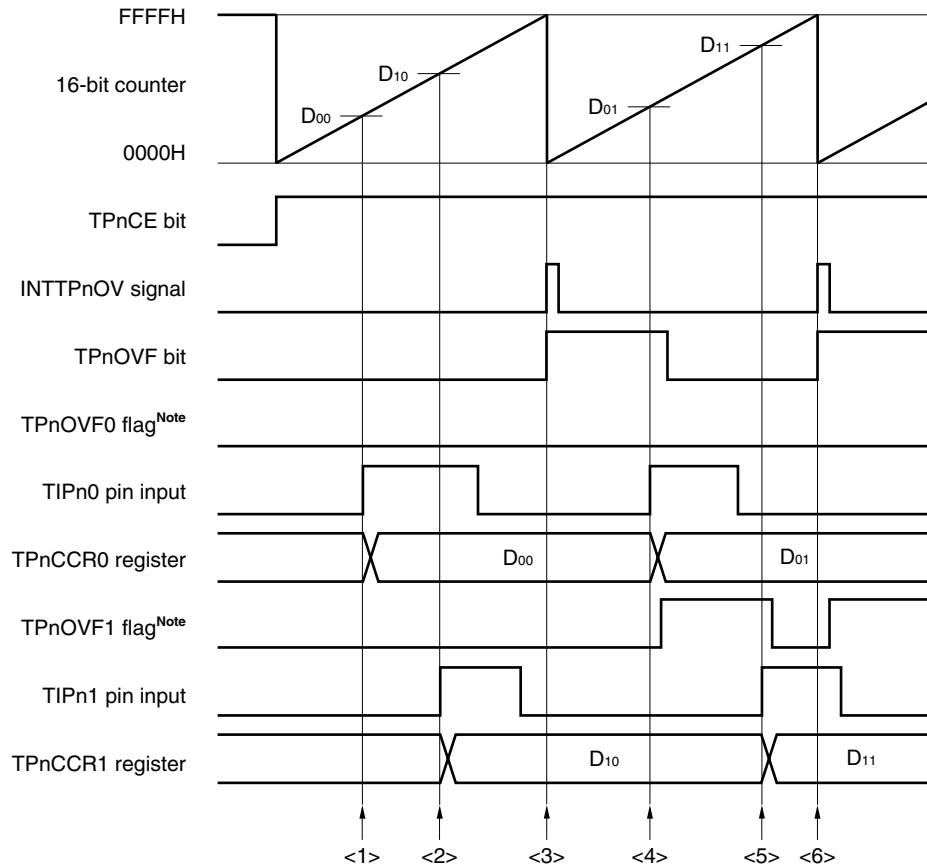
Use software when using two capture registers. An example of how to use software is shown below.

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>

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.

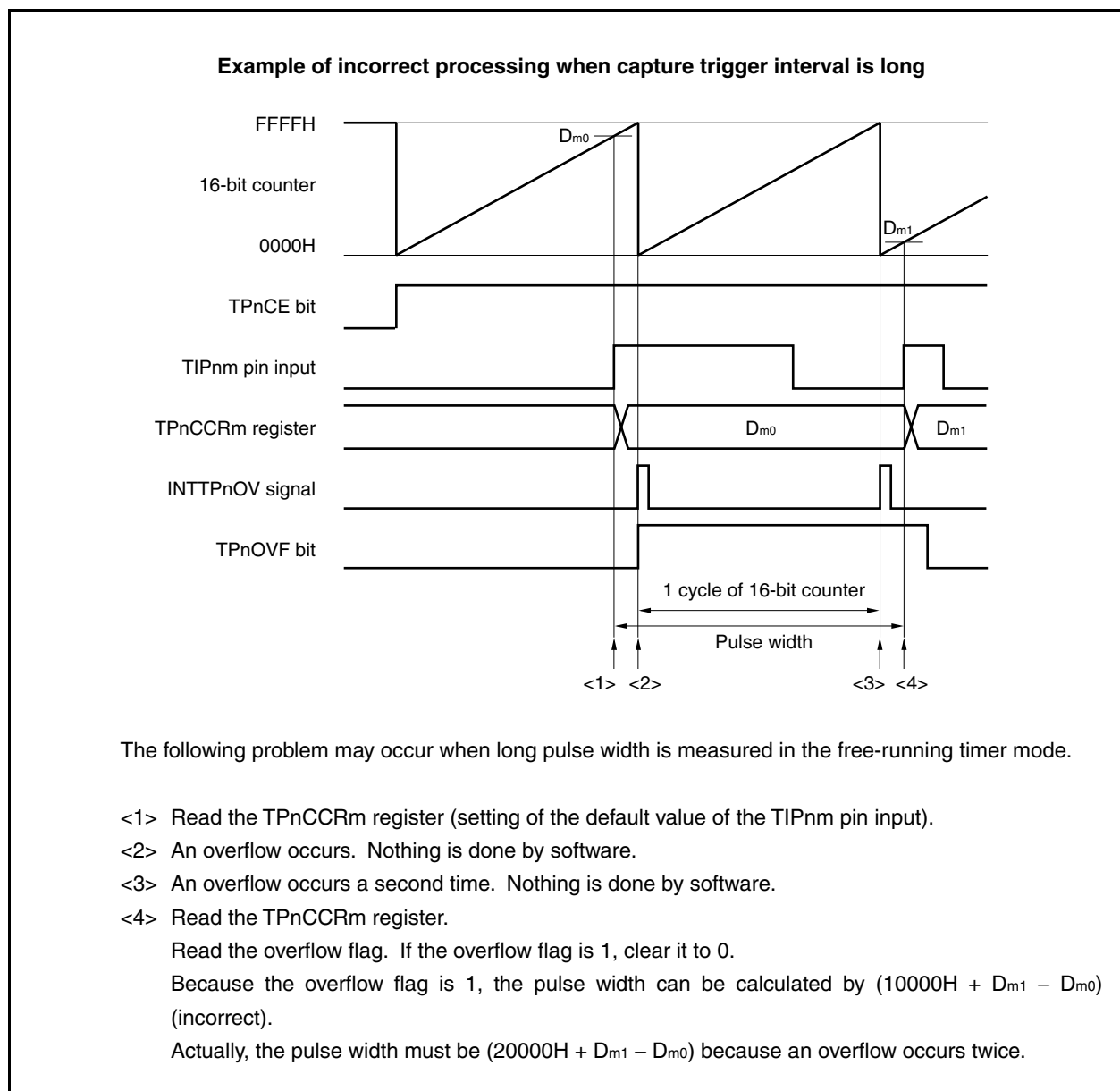
Read the TPnOVF1 flag. If the TPnOVF1 flag is 1, clear it to 0.

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

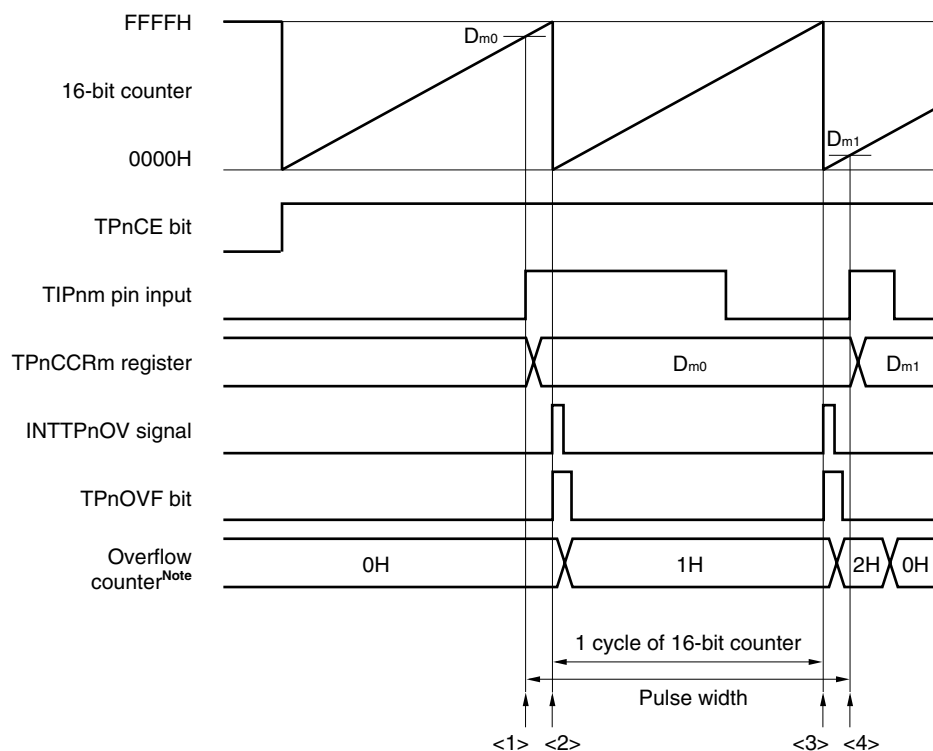
If the pulse width is greater than one cycle of the 16-bit counter, care must be exercised because an overflow may occur more than once from the first capture trigger to the next. First, an example of incorrect processing is shown below.



If an overflow occurs twice or more when the capture trigger interval is long, the correct pulse width may not be obtained.

If the capture trigger interval is long, slow the count clock to lengthen one cycle of the 16-bit counter, or use software. An example of how to use software is shown next.

Example when capture trigger interval is long

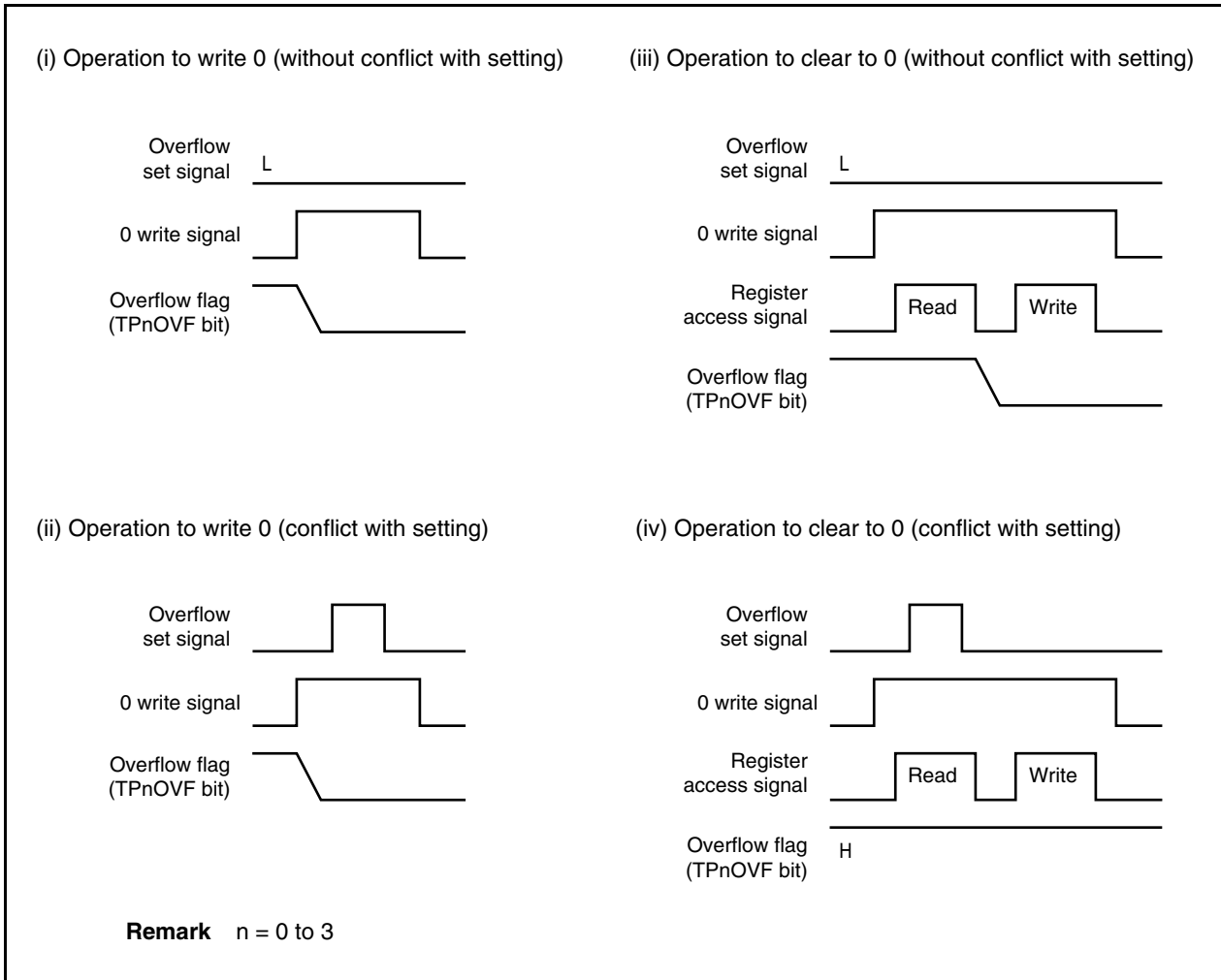


Note The overflow counter is set arbitrarily by software on the internal RAM.

- <1> Read the TPnCCRm register (setting of the default value of the 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.



To clear the overflow flag to 0, read the overflow flag to check if it is set to 1, and clear it with the CLR instruction. If 0 is written to the overflow flag without checking if the flag is 1, the set information of overflow may be erased by writing 0 ((ii) in the above chart). Therefore, software may judge that no overflow has occurred even when an overflow actually has occurred.

If execution of the CLR instruction conflicts with occurrence of an overflow when the overflow flag is cleared to 0 with the CLR instruction, the overflow flag remains set even after execution of the clear instruction.

6.5.7 Pulse width measurement mode (TPnMD2 to TPnMD0 bits = 110)

In the pulse width measurement mode, 16-bit timer/event counter P starts counting when the TPnCTL0.TPnCE bit is set to 1. Each time the valid edge input to the 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 6-34. Configuration in Pulse Width Measurement Mode

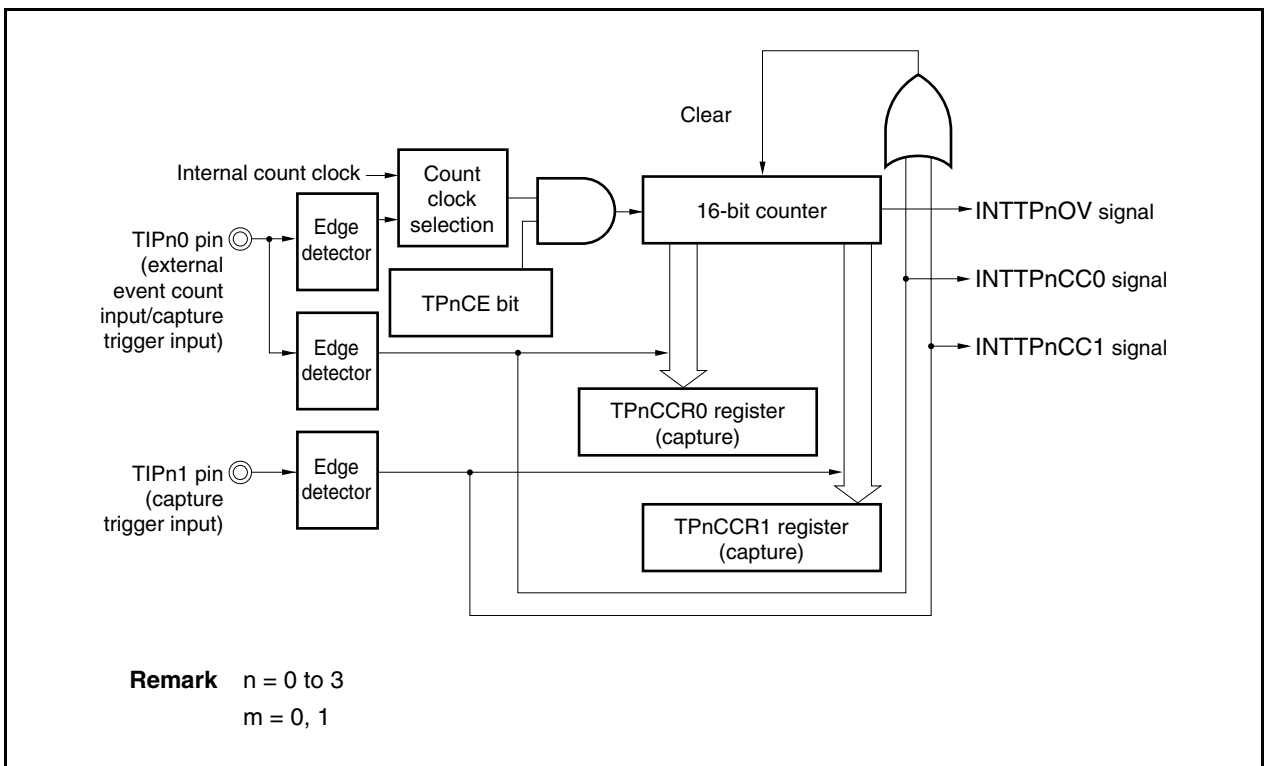
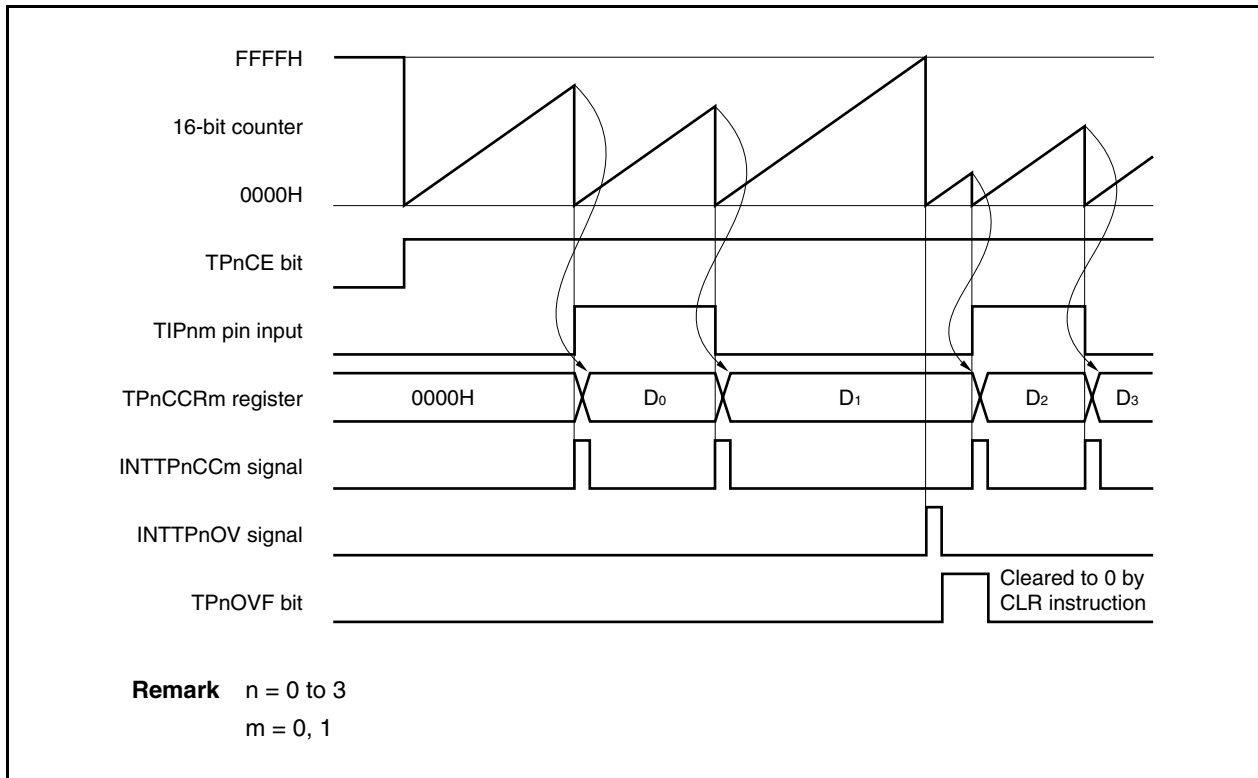


Figure 6-35. Basic Timing in Pulse Width Measurement Mode



When the TPnCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the 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.

$$\text{Pulse width} = \text{Captured value} \times \text{Count clock cycle}$$

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.

$$\text{Pulse width} = (10000\text{H} \times \text{TPnOVF bit set (1) count} + \text{Captured value}) \times \text{Count clock cycle}$$

Remark $n = 0$ to 3
 $m = 0, 1$

Figure 6-36. Register Setting in Pulse Width Measurement Mode (1/2)

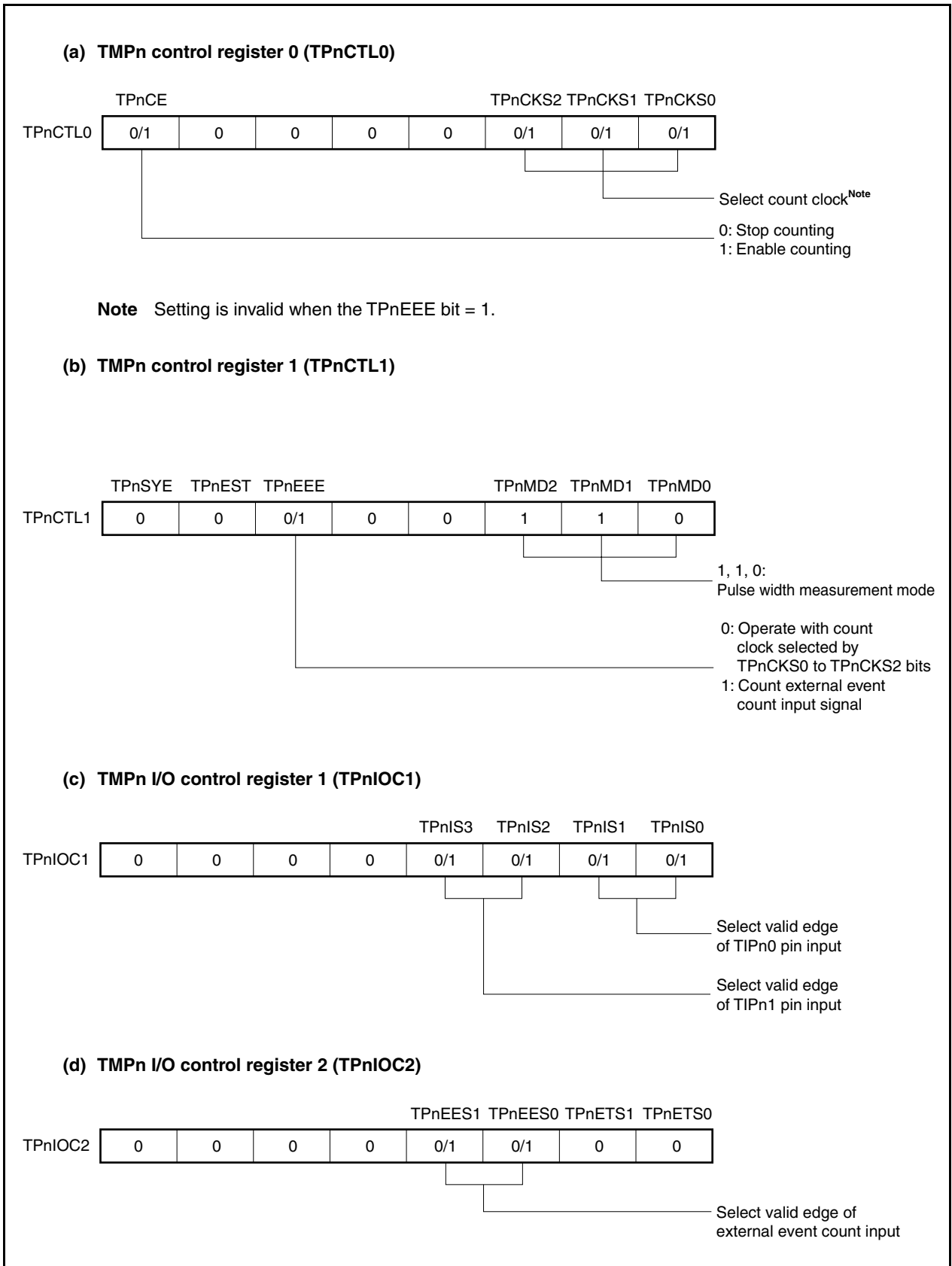
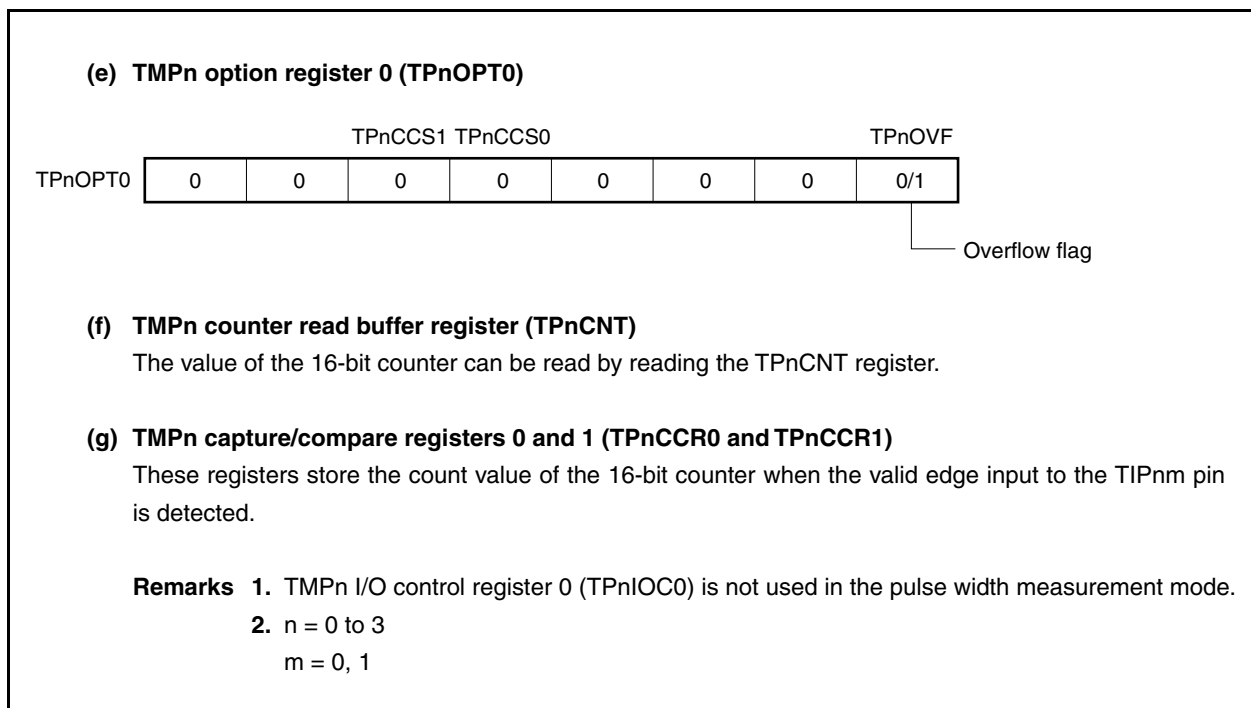
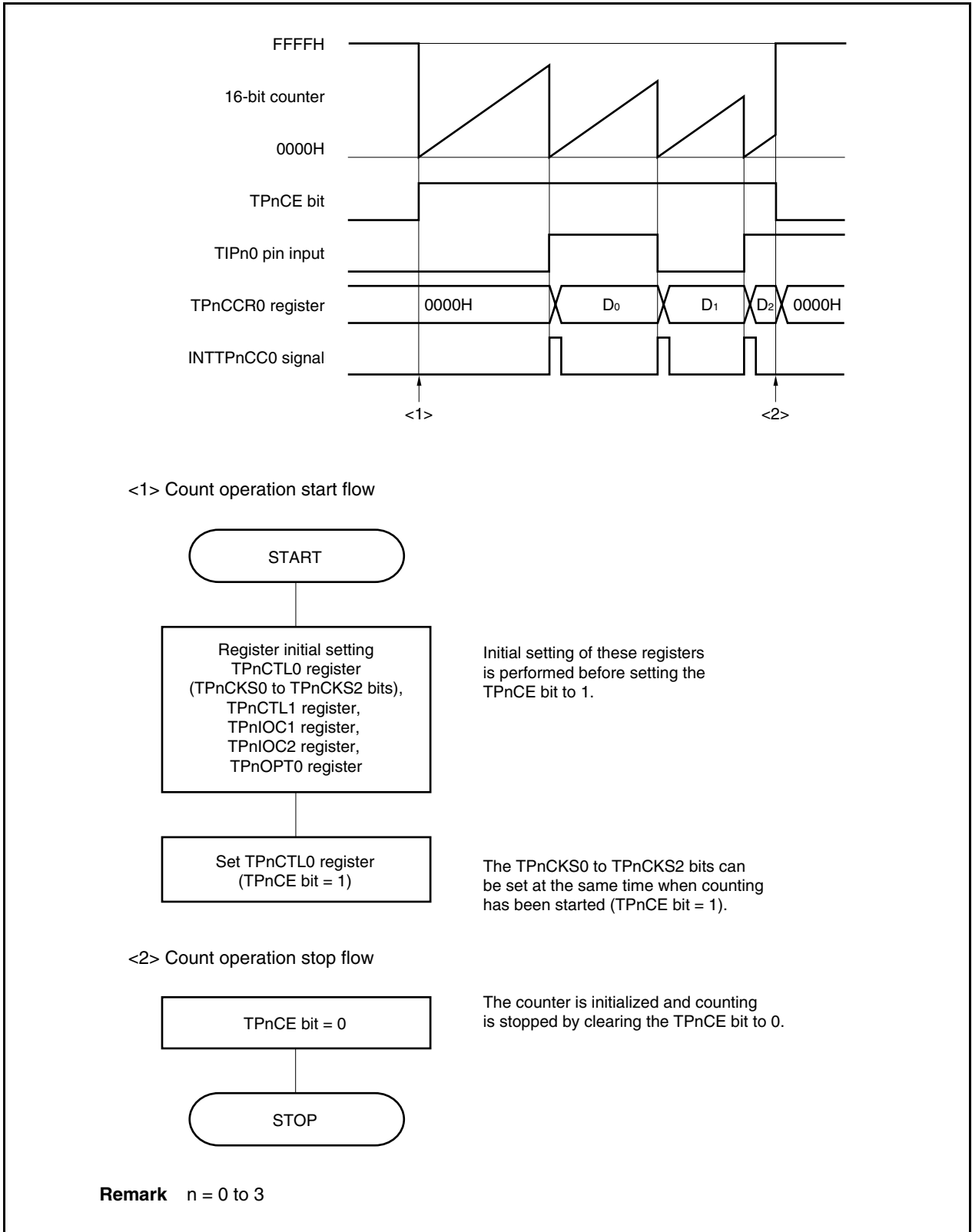


Figure 6-36. Register Setting in Pulse Width Measurement Mode (2/2)



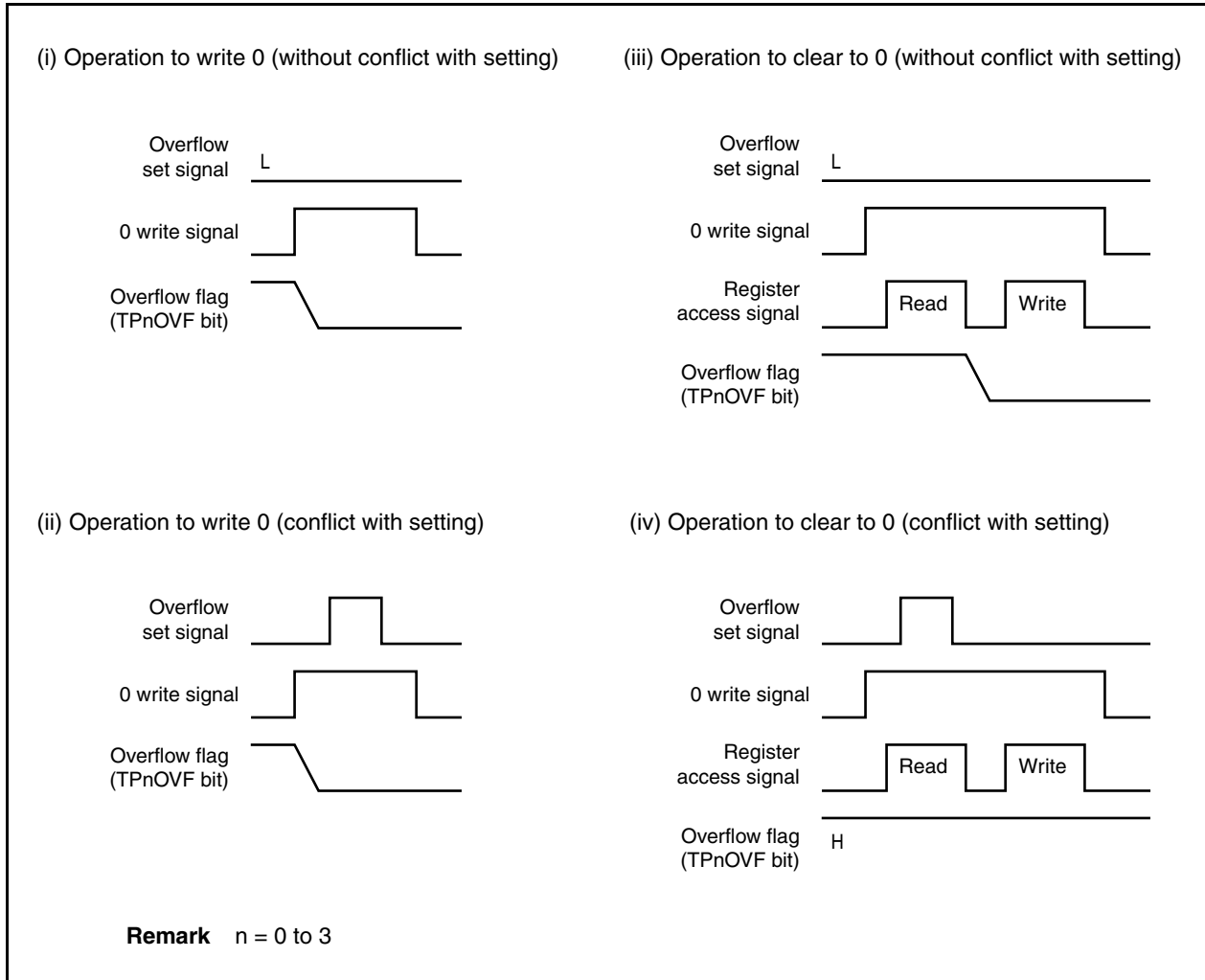
(1) Operation flow in pulse width measurement mode

Figure 6-37. Software Processing Flow in Pulse Width Measurement Mode



(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.



To clear the overflow flag to 0, read the overflow flag to check if it is set to 1, and clear it with the CLR instruction. If 0 is written to the overflow flag without checking if the flag is 1, the set information of overflow may be erased by writing 0 ((ii) in the above chart). Therefore, software may judge that no overflow has occurred even when an overflow actually has occurred.

If execution of the CLR instruction conflicts with occurrence of an overflow when the overflow flag is cleared to 0 with the CLR instruction, the overflow flag remains set even after execution of the clear instruction.

6.5.8 Timer output operations

The following table shows the operations and output levels of the TOPn0 and TOPn1 pins.

Table 6-4. 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 6-5. 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

6.6 Timer Tuned Operation Function

Timer P and timer Q have a timer tuned operation function.

The timers that can be synchronized are listed in Table 6-6.

Table 6-6. Tuned Operation Mode of Timers

| Master Timer | Slave Timer | |
|--------------|-------------|------|
| TMP0 | TMP1 | – |
| TMP2 | TMP3 | TMQ0 |

Cautions 1. The tuned operation mode is enabled or disabled by the TPmCTL1.TPmSYE and TQ0CTL1.TQ0SYE bits. For TMP2, either or both TMP3 and TMQ0 can be specified as slaves.

2. Set the tuned operation mode using the following procedure.

<1> Set the TPmCTL1.TPmSYE and TQ0CTL1.TQ0SYE bits of the slave timer to enable the tuned operation.

Set the TPmCTL1.TPmMD2 to TPmCTL1.TPmMD0 and TQ0CTL1.TQ0MD2 to TQ0CTL1.TQ0MD0 bits of the slave timer to the free-running mode.

<2> Set the timer mode by using the TPnCTL1.TPnMD2 to TPnCTL1.TPnMD0 bits.

At this time, do not set the TPnCTL1.TPnSYE bit of the master timer.

<3> Set the compare register value of the master and slave timers.

<4> Set the TPmCTL0.TPmCE and TQ0CTL0.TQ0CE bits of the slave timer to enable operation on the internal operating clock.

<5> Set the TPnCTL0.TPnCE bit of the master timer to enable operation on the internal operating clock.

Remark m = 1, 3
n = 0, 2

Tables 6-7 and 6-8 show the timer modes that can be used in the tuned operation mode (√: Settable, ×: Not settable).

Table 6-7. Timer Modes Usable in Tuned Operation Mode

| Master Timer | Free-Running Mode | PWM Mode | Triangular Wave PWM Mode |
|--------------|-------------------|----------|--------------------------|
| TMP0 | √ | √ | × |
| TMP2 | √ | √ | × |

Table 6-8. Timer Output Functions

| Tuned Channel | Timer | Pin | Free-Running Mode | | PWM Mode | | Triangular Wave PWM Mode | |
|---------------|------------------|----------------|-------------------|-----------|------------|-----------|--------------------------|-----------|
| | | | Tuning OFF | Tuning ON | Tuning OFF | Tuning ON | Tuning OFF | Tuning ON |
| Ch0 | TMP0 (master) | TOP00 | PPG | ← | Toggle | ← | N/A | ← |
| | | TOP01 | PPG | ← | PWM | ← | N/A | ← |
| | TMP1 (slave) | TOP10 | PGP | ← | Toggle | PWM | N/A | ← |
| | | TOP11 | PPG | ← | PWM | ← | N/A | ← |
| Ch1 | TMP2 (master) | TOP20 | PPG | ← | Toggle | ← | N/A | ← |
| | | TOP21 | PPG | ← | PWM | ← | N/A | ← |
| | TMP3 (slave) | TOP30 | PPG | ← | Toggle | PWM | N/A | ← |
| | | TOP31 | PPG | ← | PWM | ← | N/A | ← |
| | TMQ0 (slave) | TOQ00 | PPG | ← | Toggle | PWM | Toggle | N/A |
| | | TOQ01 to TOQ03 | PPG | ← | PWM | ← | Triangular wave PWM | N/A |

Remark The timing of transmitting data from the compare register of the master timer to the compare register of the slave timer is as follows.

PPG: CPU write timing

Toggle, PWM, triangular wave PWM: Timing at which timer counter and compare register match TOPn0 and TOQ00 (n = 0 to 3)

Figure 6-38. Tuned Operation Image (TMP2, TMP3, TMQ0)

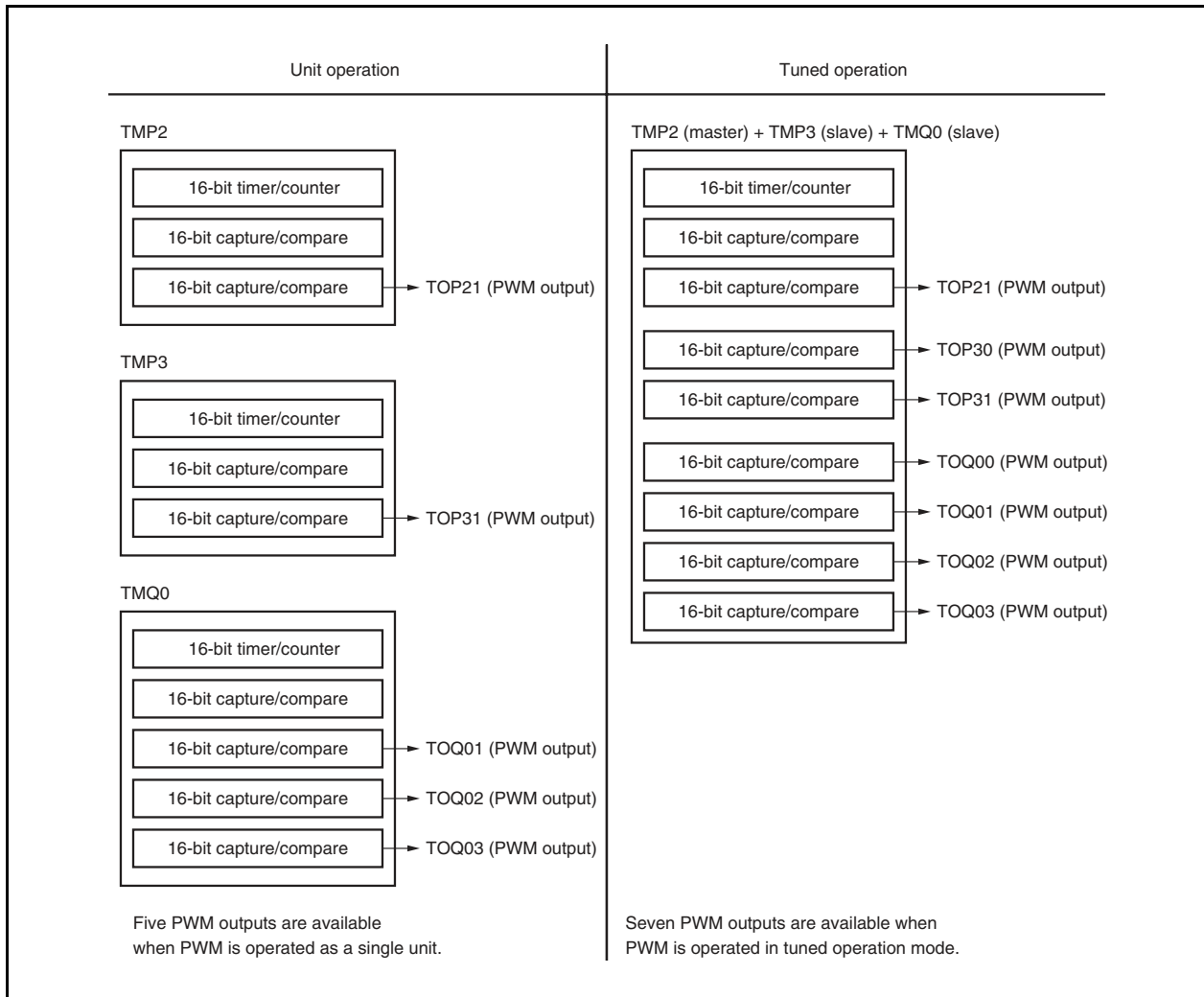
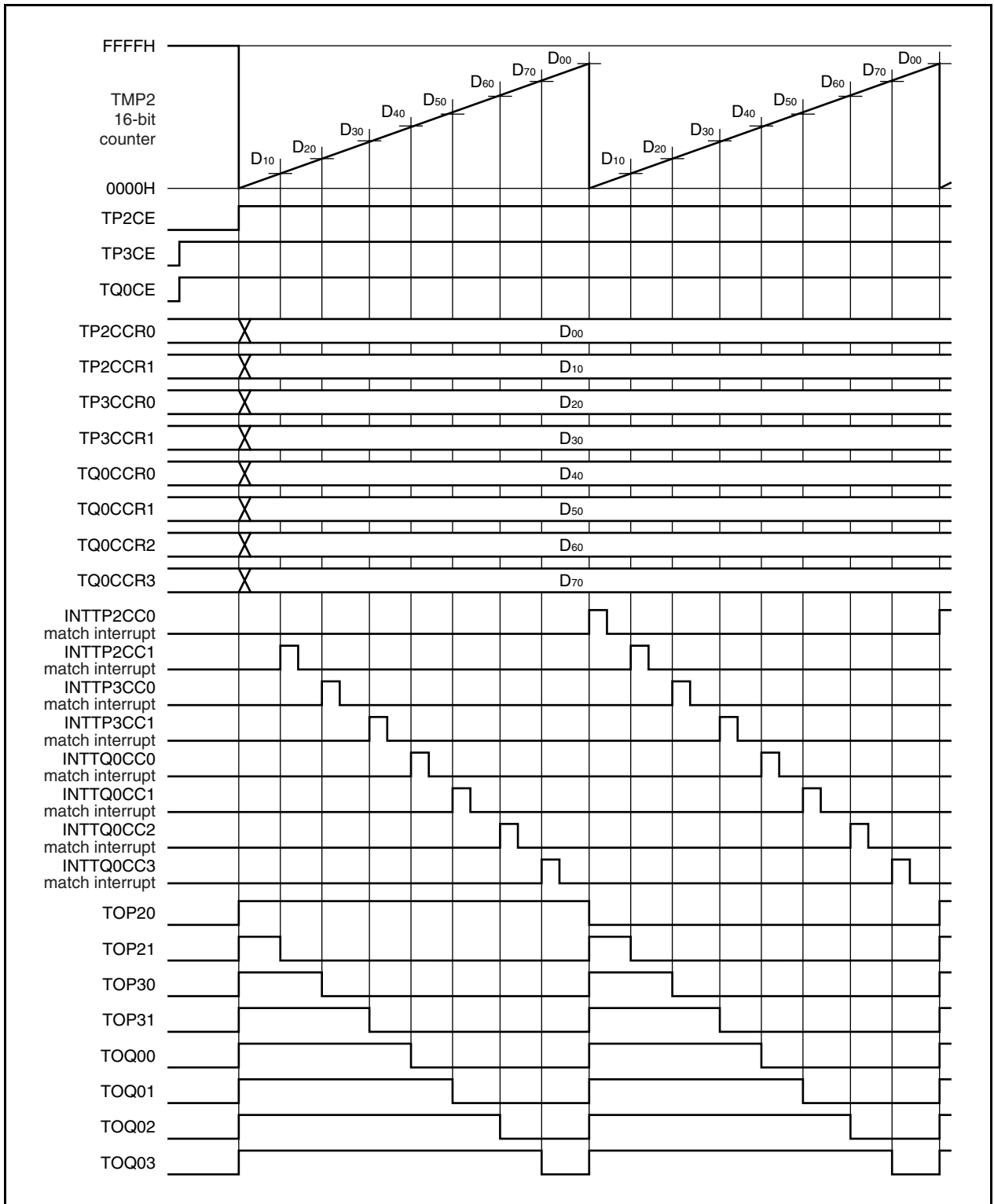


Figure 6-39. Basic Operation Timing of Tuned PWM Function (TMP2, TMP3, TMQ0)



6.7 Selector Function

In the V850ES/HF2, the alternate-function pins of port and peripheral I/O (TMP, TMM0, or UARTA) can be used to select the capture trigger input of TMP.

By using this function, the following is possible.

- The TIP10 and TIP11 input signals of TMP1 can be selected from the port/timer alternate-function pins (TIP10 and TIP11 pins) and the UARTA reception alternate-function pins (RXDA0 and RXDA1).
→ When the RXDA0 or RXDA1 signal of UARTA0 or UARTA1 is selected, the baud rate error of the UARTA LIN reception transfer can be calculated.
- The TIP01 input signal of TMP0 can be selected from the port/timer alternate-function pin (TIP01 pin) and the INTTMM0EQ0 signal of TMM0.

- Cautions**
1. **When using the selector function, set the capture trigger input of TMP before connecting the timer.**
 2. **When setting the selector function, first disable the peripheral I/O to be connected (TMP, TMM0, or UARTA).**

The capture input for the selector function is specified by the following register.

(1) Selector operation control register 0 (SELCNT0)

The SELCNT0 register is an 8-bit register that selects the capture trigger for TMP0 and TMP1.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF308H

| | | | | | | | | |
|---------|---|---|---|--------|--------|--------|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SELCNT0 | 0 | 0 | 0 | ISEL04 | ISEL03 | ISEL02 | 0 | 0 |

| | |
|---------------|--|
| 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^{Note} | Selection of TIP01 input signal (TMP0) |
| 0 | TIP01 pin input |
| 1 | INTTM0EQ0 interrupt of TMM0 |

Note Use the INTTM0EQ0 interrupt signal as the TIP01 input signal under the following condition.

TMM0 operation clock \geq TMP0 operation clock \times 4

Cautions 1. To set the ISEL02 to ISEL04 bits to 1, set the corresponding pin in the capture input mode.

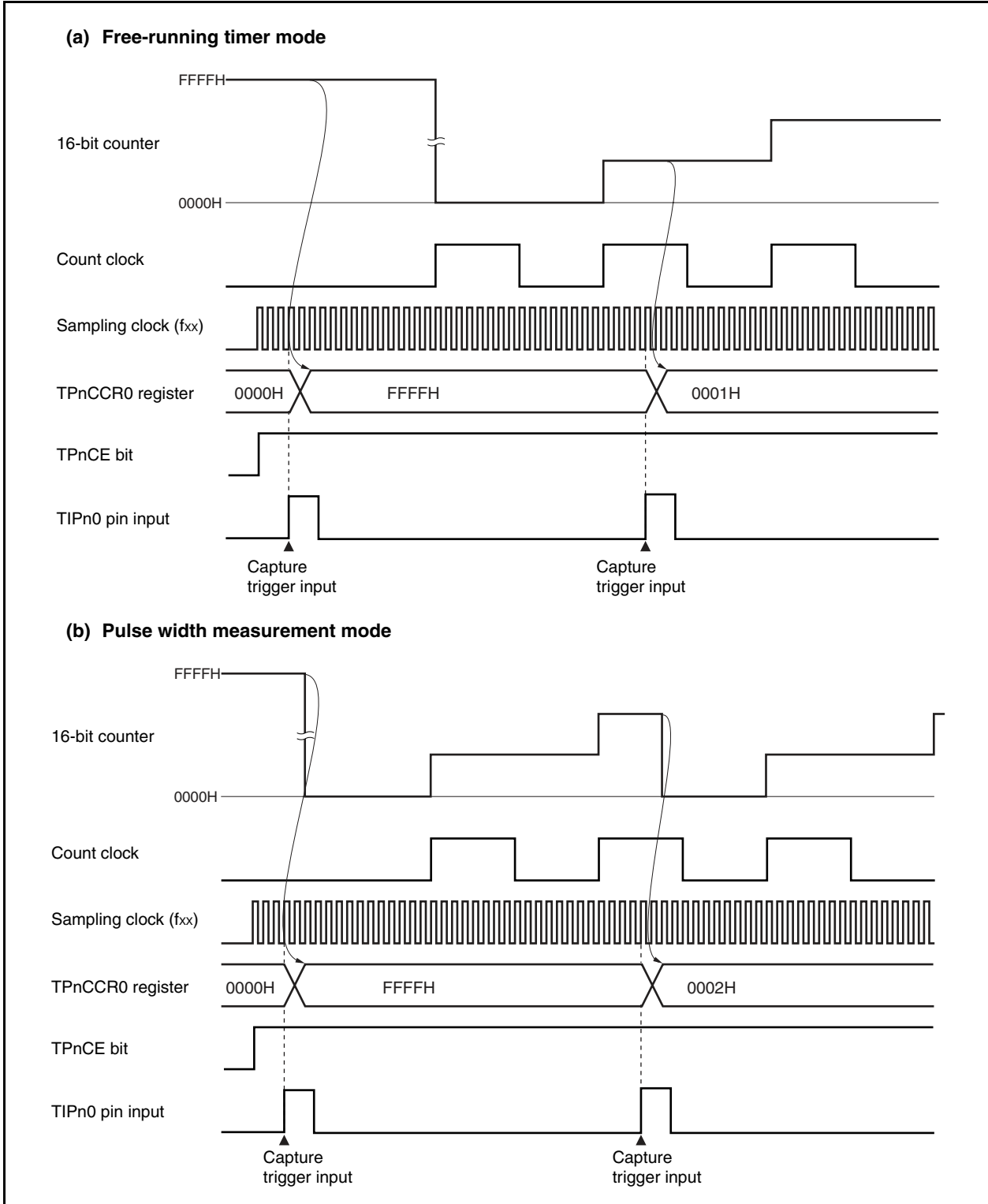
2. Set the ISEL02 to ISEL06 bits when operation of the target (TMP0, TMP1, TMM0, UARTA0, or UARTA1) is stopped.

<R>

6.8 Cautions

(1) Capture operation

When the capture operation is used and a slow clock is selected as the count clock, FFFFH, not 0000H, may be captured in the TPnCCR0 and TPnCCR1 registers if the capture trigger is input immediately after the TPnCE bit is set to 1.



CHAPTER 7 16-BIT TIMER/EVENT COUNTER Q (TMQ)

Timer Q (TMQ) is a 16-bit timer/event counter.
The V850ES/HF2 incorporates TMQ0.

7.1 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

7.2 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
- Triangular wave PWM output
- Timer tuned operation function

7.3 Configuration

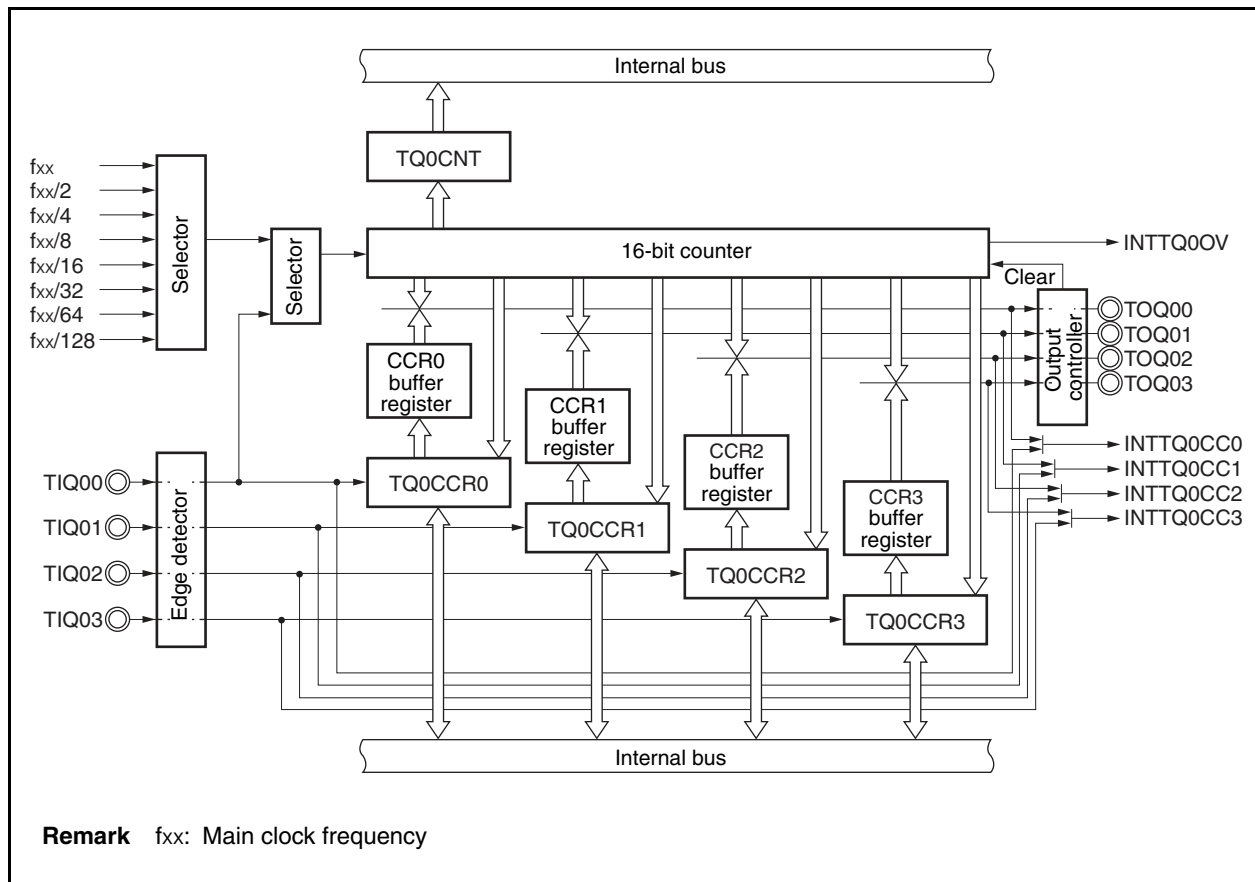
TMQ0 includes the following hardware.

Table 7-1. 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 1} to TIQ03 pins) |
| Timer outputs | 4 (TOQ00 to TOQ03 pins) |
| Control registers ^{Note 2} | TMQ0 control registers 0, 1 (TQ0CTL0, TQ0CTL1) TMQ0 I/O control registers 0 to 2 (TQ0IOC0 to TQ0IOC2) TMQ0 option register 0 (TQ0OPT0) |

- Notes 1.** The TIQ00 pin functions alternately as a capture trigger input signal, external event count input signal, and external trigger input signal.
- 2.** When using the functions of the TIQ00 to TIQ03 and TOQ00 to TOQ03 pins, refer to **Table 4-14 Using Port Pin as Alternate-Function Pin.**

Figure 7-1. Block Diagram of TMQ0



(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 sets 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.

7.4 Registers

The registers that control TMQ0 are as follows.

- TMQ0 control register 0 (TQ0CTL0)
- TMQ0 control register 1 (TQ0CTL1)
- TMQ0 I/O control register 0 (TQ0IOC0)
- TMQ0 I/O control register 1 (TQ0IOC1)
- TMQ0 I/O control register 2 (TQ0IOC2)
- TMQ0 option register 0 (TQ0OPT0)
- TMQ0 capture/compare register 0 (TQ0CCR0)
- TMQ0 capture/compare register 1 (TQ0CCR1)
- TMQ0 capture/compare register 2 (TQ0CCR2)
- TMQ0 capture/compare register 3 (TQ0CCR3)
- TMQ0 counter read buffer register (TQ0CNT)

Remark When using the functions of the TIQ00 to TIQ03 and TOQ00 to TOQ03 pins, refer to **Table 4-14 Using Port Pin as Alternate-Function Pin**.

(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 sets 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 clear 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 sets this register to 00H.

(1/2)

After reset: 00H R/W Address: FFFFF541H

| | | | | | | | |
|--------|--------|--------|---|---|--------|--------|--------|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TQ0SYE | TQ0EST | TQ0EEE | 0 | 0 | TQ0MD2 | TQ0MD1 | TQ0MD0 |

| | | | | | | | |
|---|--|--------------|-------------|--|------|------|------|
| TQ0SYE | Tuned operation mode enable control | | | | | | |
| 0 | Independent operation mode (asynchronous operation mode) | | | | | | |
| 1 | Tuned operation mode (specification of slave operation) In this mode, timer P can operate in synchronization with a master timer. | | | | | | |
| <table border="1" style="margin-left: 40px;"> <tr> <td style="text-align: center;">Master timer</td> <td colspan="2" style="text-align: center;">Slave timer</td> </tr> <tr> <td style="text-align: center;">TMP2</td> <td style="text-align: center;">TMP3</td> <td style="text-align: center;">TMQ0</td> </tr> </table> | | Master timer | Slave timer | | TMP2 | TMP3 | TMQ0 |
| Master timer | Slave timer | | | | | | |
| TMP2 | TMP3 | TMQ0 | | | | | |
| For the tuned operation mode, see 7.6 Timer Tuned Operation Function . | | | | | | | |

| | |
|--------|--|
| TQ0EST | Software trigger control |
| 0 | - |
| 1 | Generate a valid signal for external trigger input. <ul style="list-style-type: none"> • 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. |

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. Be sure to clear bits 3 and 4 to "0".

| 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 | Triangular wave PWM mode |

- Cautions**
- External event count input is selected in the external event count mode regardless of the value of the TQ0EEE bit.
 - 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.

<R>

(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 sets 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 |

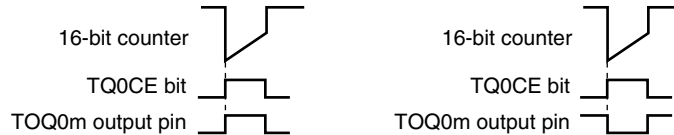
| | |
|--------|---|
| TQ0OLm | TOQ0m pin output level setting (m = 0 to 3) ^{Note} |
| 0 | TOQ0m pin high level start |
| 1 | TOQ0m pin low level start |

| | |
|--------|--|
| TQ0OEm | TOQ0m pin output setting (m = 0 to 3) |
| 0 | Timer output disabled <ul style="list-style-type: none"> • 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). |

Note The output level of the timer output pin (TOQ0m) specified by the TQ0OLm bit is shown below.

• When TQ0OLm bit = 0

• When TQ0OLm bit = 1



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 sets 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 sets 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.
 3. The TQ0ETS1 and TQ0ETS0 bits are valid only when the external trigger pulse output mode (TQ0CTL1.TQ0MD2 to TQ0CTL1.TQ0MD0 bits = 010) or the one-shot pulse output mode (TQ0CTL1.TQ0MD2 to TQ0CTL1.TQ0MD0 = 011) is 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 sets 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 |
| <ul style="list-style-type: none"> • 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 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. Be sure to clear 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.

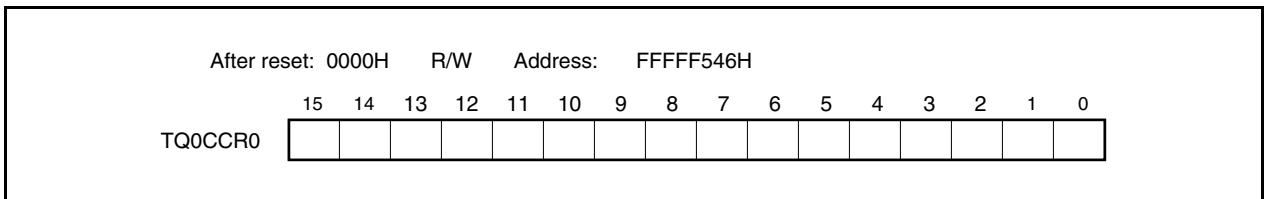
The TQ0CCR0 register can be read or written during operation.

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

Reset sets this register to 0000H.

Caution Accessing the TQ0CCR0 register is prohibited in the following statuses. For details, refer to 3.4.8 (2) Accessing specific on-chip peripheral I/O registers.

- When the CPU operates with the subclock and the main clock oscillation is stopped
- When the CPU operates with the internal oscillation clock



(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, PWM output mode, or triangular wave PWM 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 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 | — |
| Triangular wave PWM mode | Compare register | Batch write |

(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.

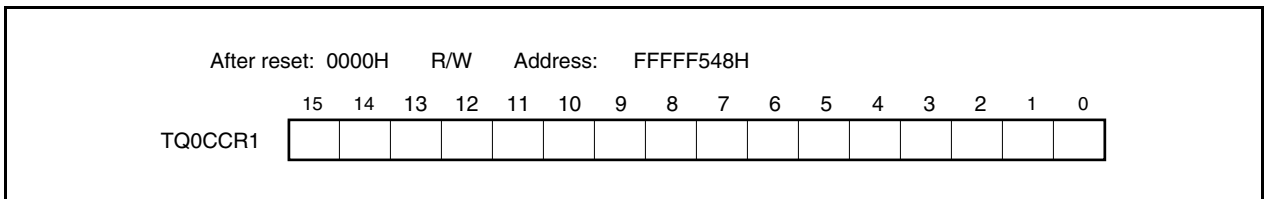
The TQ0CCR1 register can be read or written during operation.

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

Reset sets this register to 0000H.

Caution Accessing the TQ0CCR1 register is prohibited in the following statuses. For details, refer to 3.4.8 (2) Accessing specific on-chip peripheral I/O registers.

- When the CPU operates with the subclock and the main clock oscillation is stopped
- When the CPU operates with the internal oscillation clock



(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 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 | – |
| Triangular wave PWM mode | Compare register | Batch write |

(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.

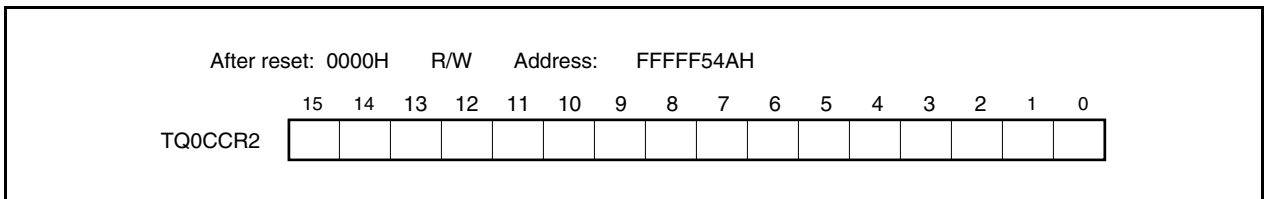
The TQ0CCR2 register can be read or written during operation.

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

Reset sets this register to 0000H.

Caution Accessing the TQ0CCR2 register is prohibited in the following statuses. For details, refer to 3.4.8 (2) Accessing specific on-chip peripheral I/O registers.

- When the CPU operates with the subclock and the main clock oscillation is stopped
- When the CPU operates with the internal oscillation clock



(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 7-4. 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 | – |
| Triangular wave PWM mode | Compare register | Batch write |

(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.

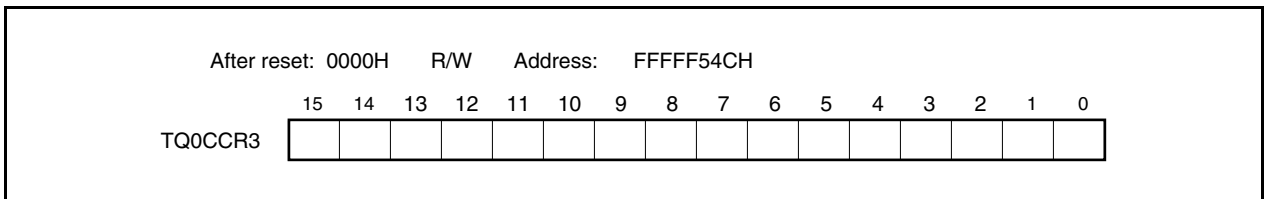
The TQ0CCR3 register can be read or written during operation.

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

Reset sets this register to 0000H.

Caution Accessing the TQ0CCR3 register is prohibited in the following statuses. For details, refer to 3.4.8 (2) Accessing specific on-chip peripheral I/O registers.

- When the CPU operates with the subclock and the main clock oscillation is stopped
- When the CPU operates with the internal oscillation clock



(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 7-5. 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 | – |
| Triangular wave PWM mode | Compare register | Batch write |

(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.

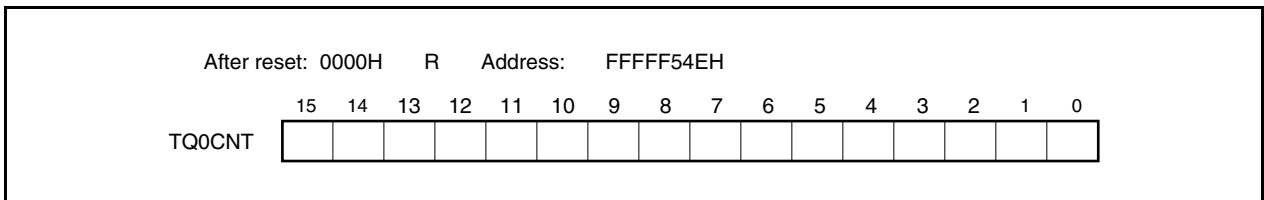
This register is read-only, in 16-bit units.

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 prohibited in the following statuses. For details, refer to 3.4.8 (2) Accessing specific on-chip peripheral I/O registers.

- When the CPU operates with the subclock and the main clock oscillation is stopped
- When the CPU operates with the internal oscillation clock



(12) TIQ0m pin noise elimination control register (Q0mNFC)

The Q0mNFC register is an 8-bit register that sets the digital noise filter of the timer Q input pin for noise elimination.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: Q00NFC: FFFF50H (TIQ00 pin)
 Q01NFC: FFFF54H (TIQ01 pin)
 Q02NFC: FFFF58H (TIQ02 pin)
 Q03NFC: FFFF5CH (TIQ03 pin)

| | | | | | | | | |
|--------|---|-------|---|---|---|------|------|------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Q0mNFC | 0 | NFSTS | 0 | 0 | 0 | NFC2 | NFC1 | NFC0 |

(m = 0 to 3)

| NFSTS | Setting of number of times of sampling by digital noise filter |
|-------|--|
| 0 | 3 times |
| 1 | 2 times |

| NFC2 | NFC1 | NFC0 | Sampling clock |
|------------------|------|------|--------------------|
| 0 | 0 | 0 | f_{xx} |
| 0 | 0 | 1 | $f_{xx}/2$ |
| 0 | 1 | 0 | $f_{xx}/4$ |
| 0 | 1 | 1 | $f_{xx}/16$ |
| 1 | 0 | 0 | $f_{xx}/32$ |
| 1 | 0 | 1 | $f_{xx}/64$ |
| Other than above | | | Setting prohibited |

Cautions 1. Be sure to clear bits 3 to 5 and 7 to "0".

2. A signal input to the timer input pin (TIQ0m) before the Q0mNFC register is set is output with digital noise eliminated.

Therefore, set the sampling clock (NFC2 to NFC0) and the number of times of sampling (NFSTS) by using the Q0mNFC register, wait for initialization time = (Sampling clock) × (Number of times of sampling), and enable the timer operation.

Remark The width of the noise that can be accurately eliminated is (Sampling clock) × (Number of times of sampling – 1). Even noise with a width narrower than this may cause a miscount if it is synchronized with the sampling clock.

7.5 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 |
| Triangular wave PWM mode | Invalid | Invalid | Compare only | Batch write |

- 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).

7.5.1 Interval timer mode (TQ0MD2 to TQ0MD0 bits = 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 7-2. Configuration of Interval Timer

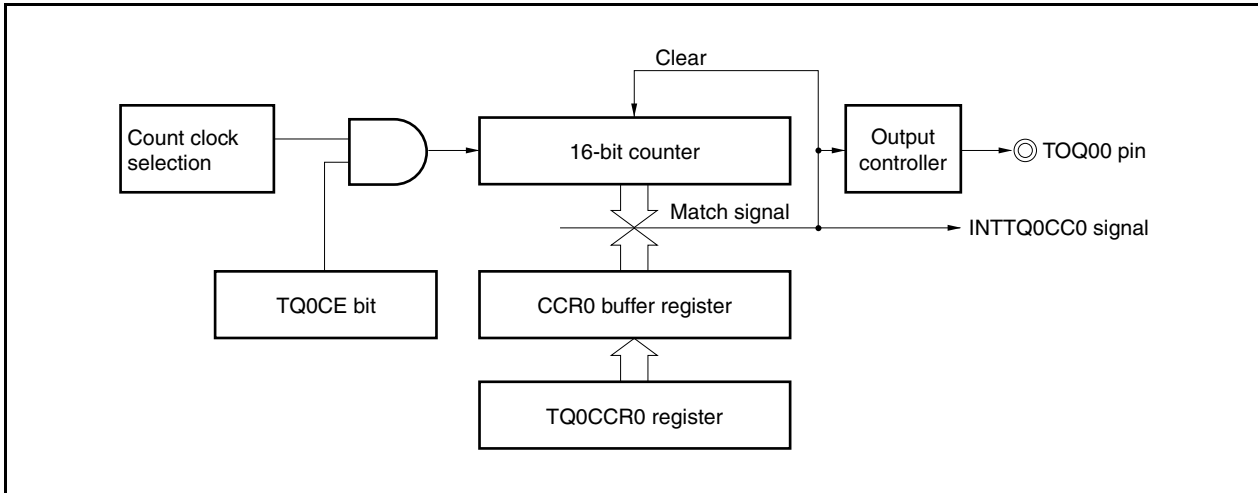
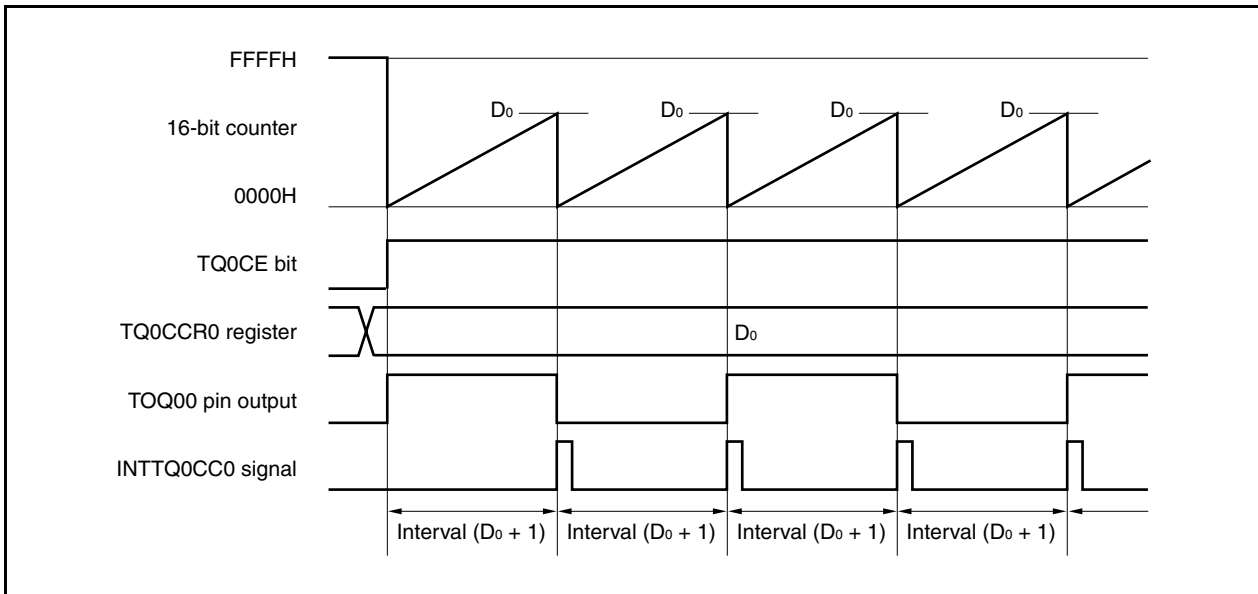


Figure 7-3. Basic Timing of Operation in Interval Timer Mode



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.

$$\text{Interval} = (\text{Set value of TQ0CCR0 register} + 1) \times \text{Count clock cycle}$$

Figure 7-4. Register Setting for Interval Timer Mode Operation (1/2)

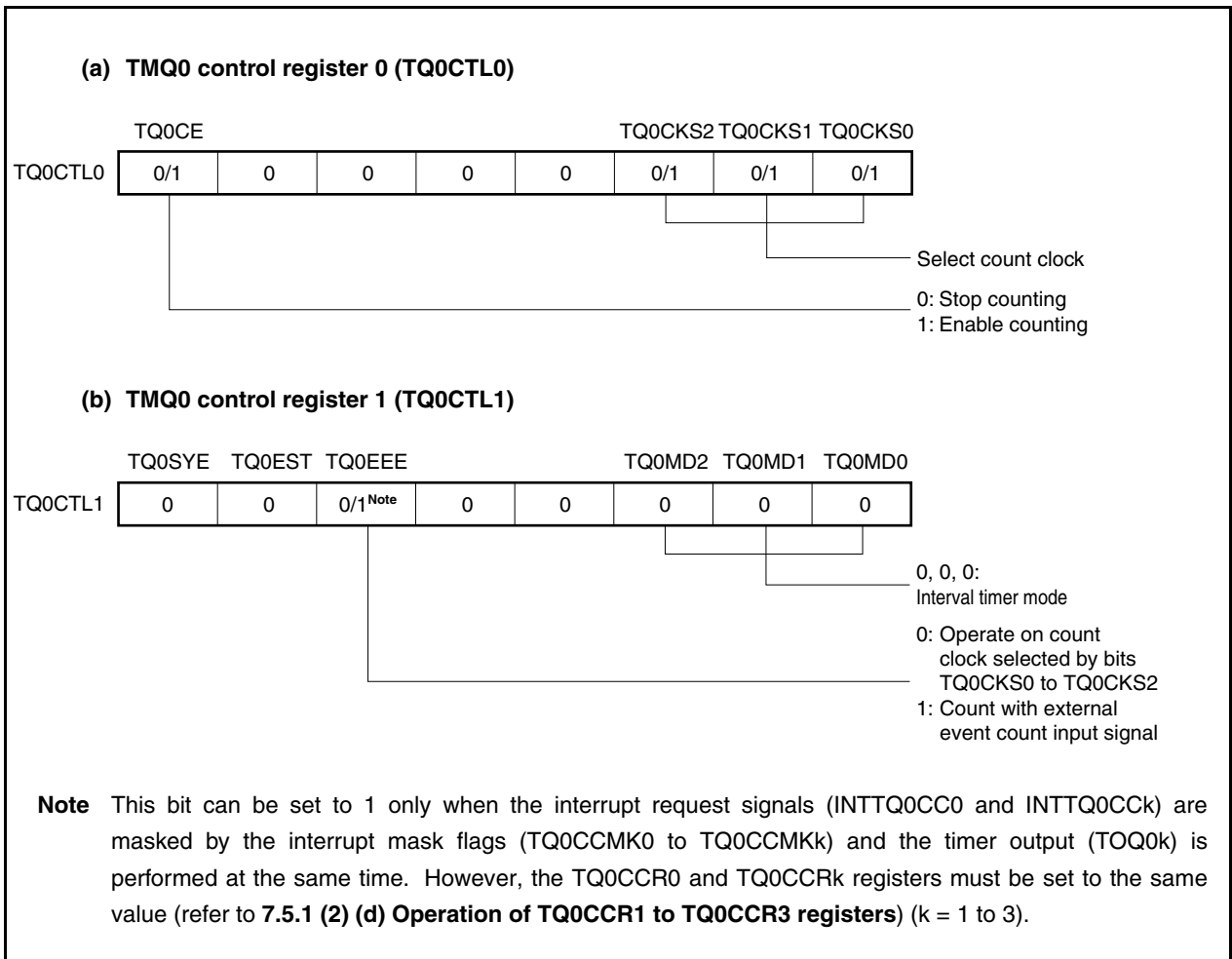
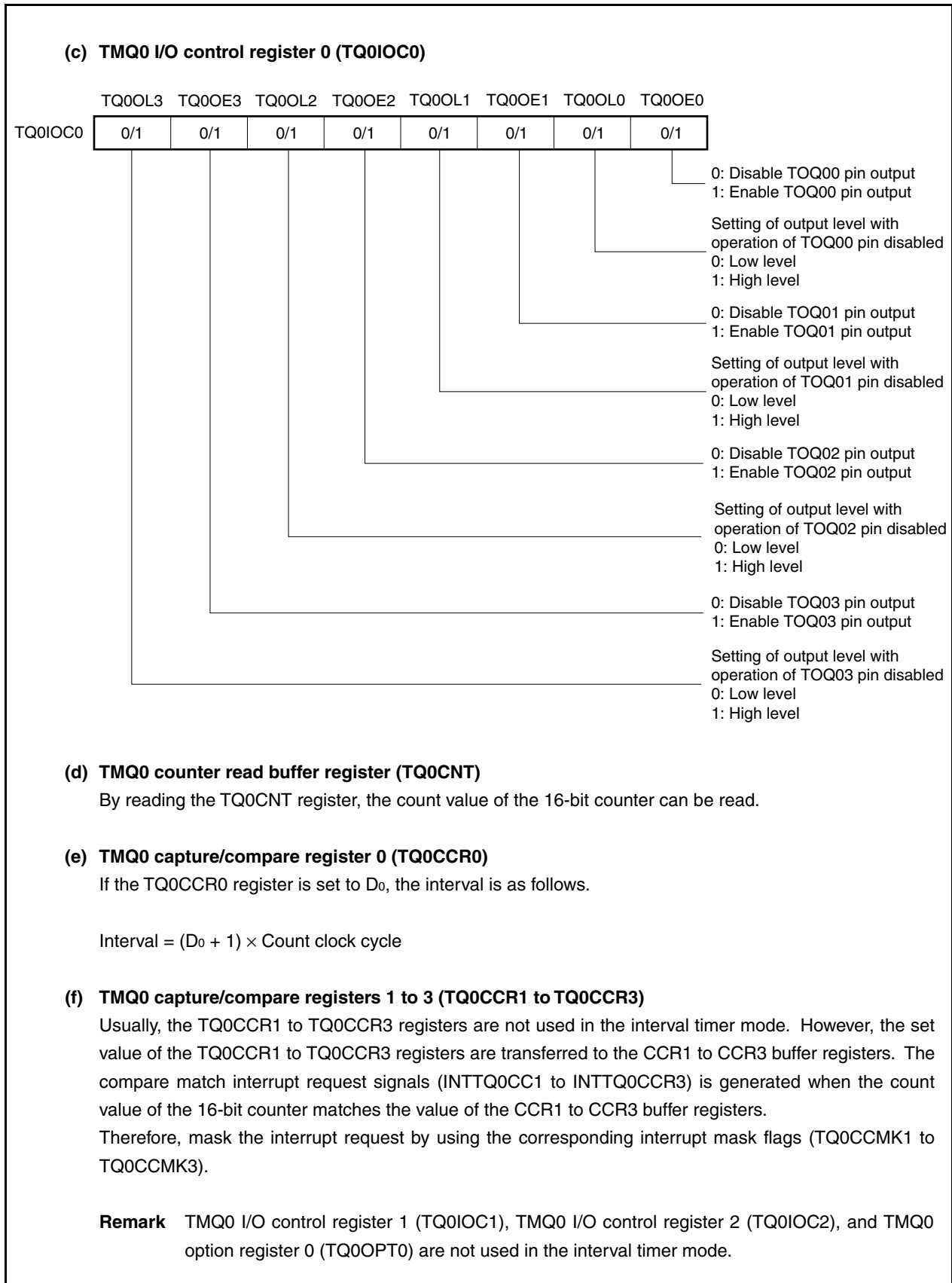
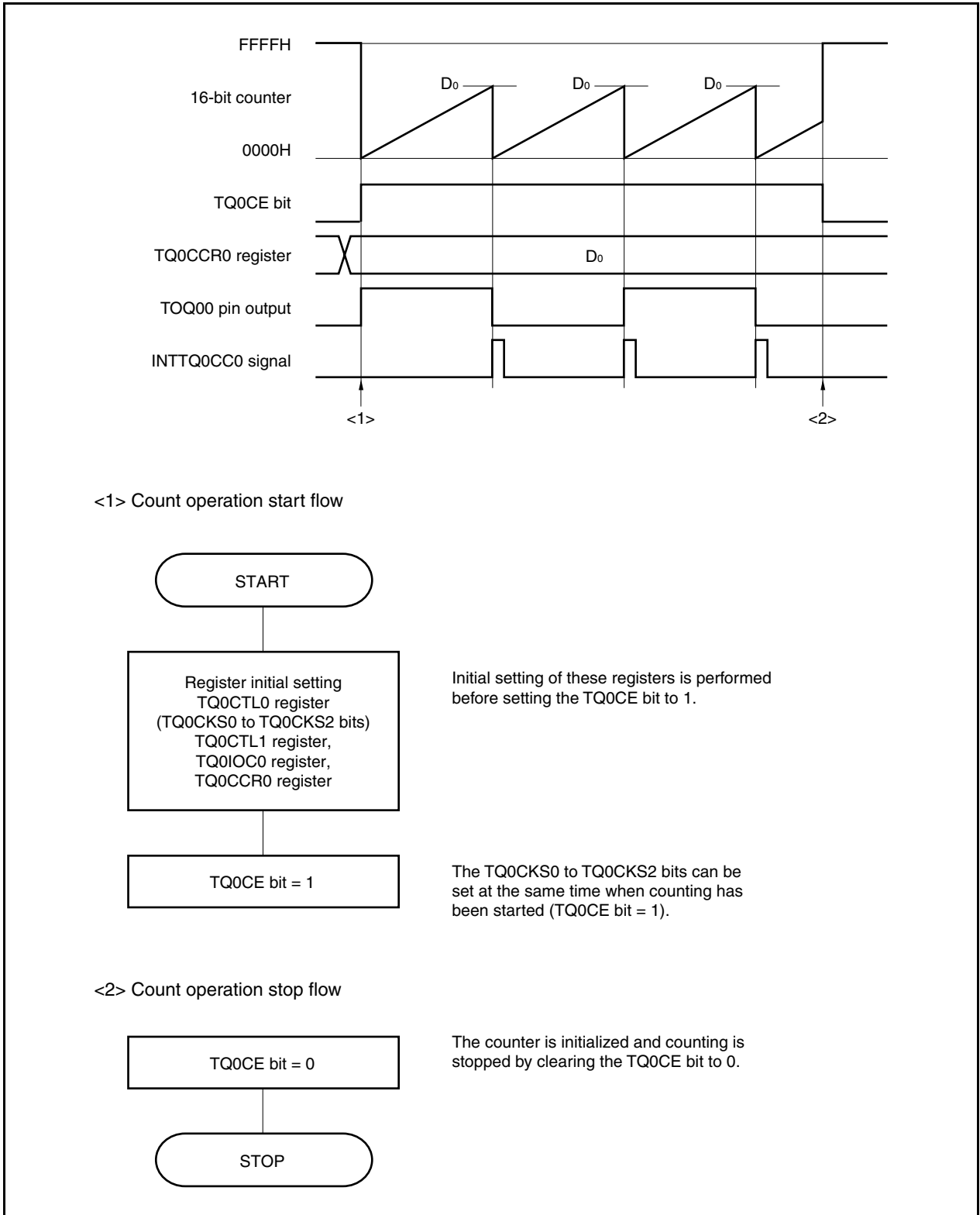


Figure 7-4. Register Setting for Interval Timer Mode Operation (2/2)



(1) Interval timer mode operation flow

Figure 7-5. Software Processing Flow in Interval Timer Mode

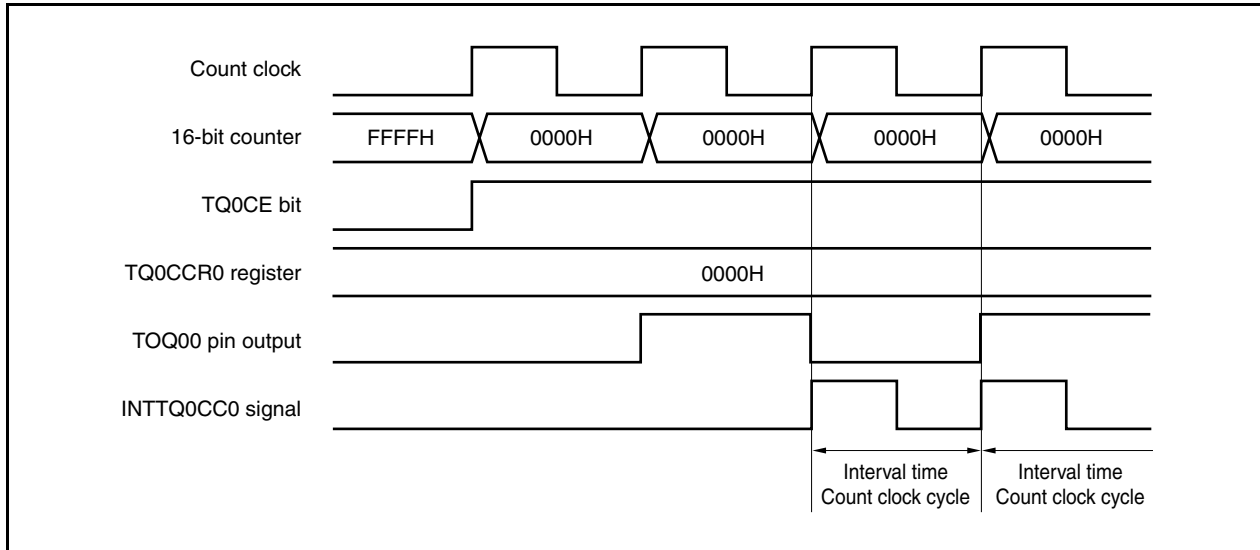


(2) Interval timer mode operation timing

(a) Operation if TQ0CCR0 register is set to 0000H

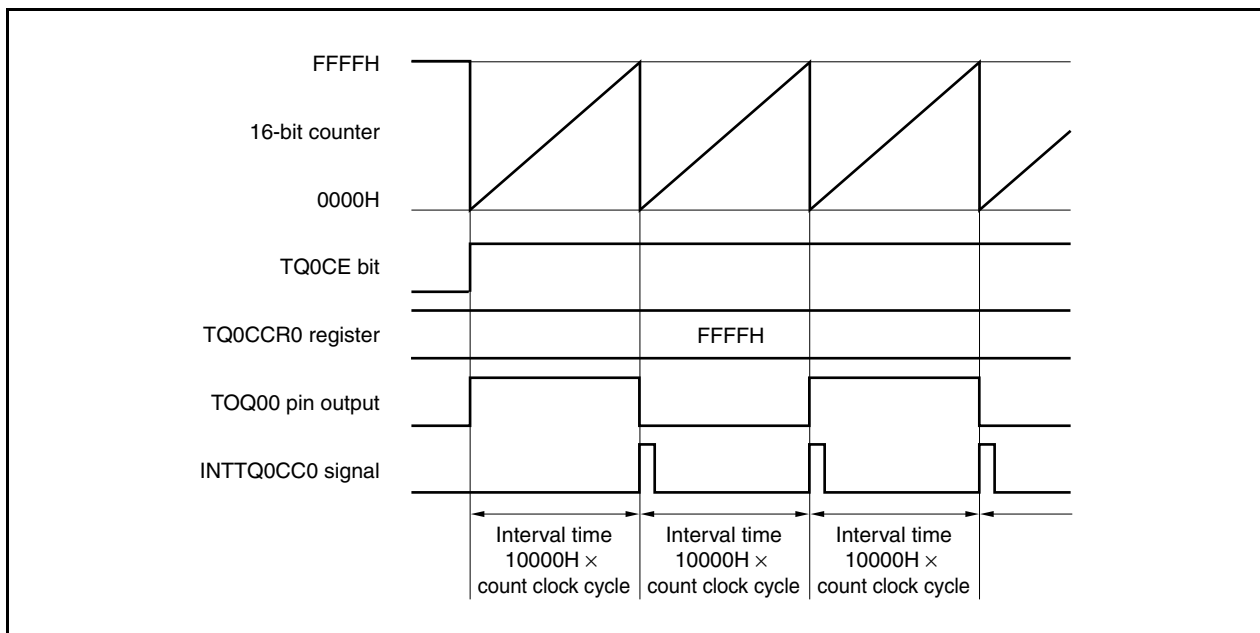
If the TQ0CCR0 register is set to 0000H, the INTTQ0CC0 signal is generated at each count clock subsequent to the first count clock, and the output of the TOQ00 pin is inverted.

The value of the 16-bit counter is always 0000H.



(b) Operation if TQ0CCR0 register is set to FFFFH

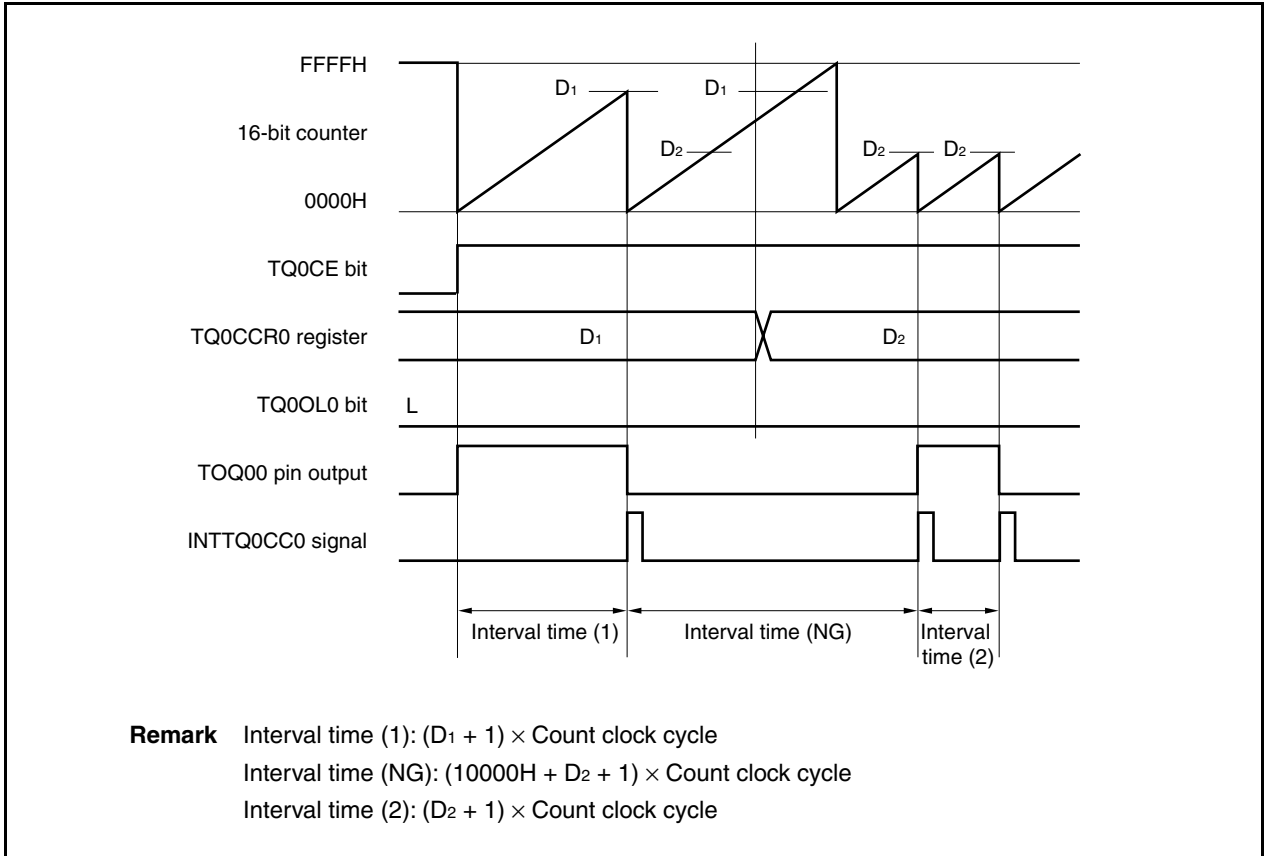
If the TQ0CCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH. The counter is cleared to 0000H in synchronization with the next count-up timing. The INTTQ0CC0 signal is generated and the output of the TOQ00 pin is inverted. At this time, an overflow interrupt request signal (INTTQ0OV) 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.



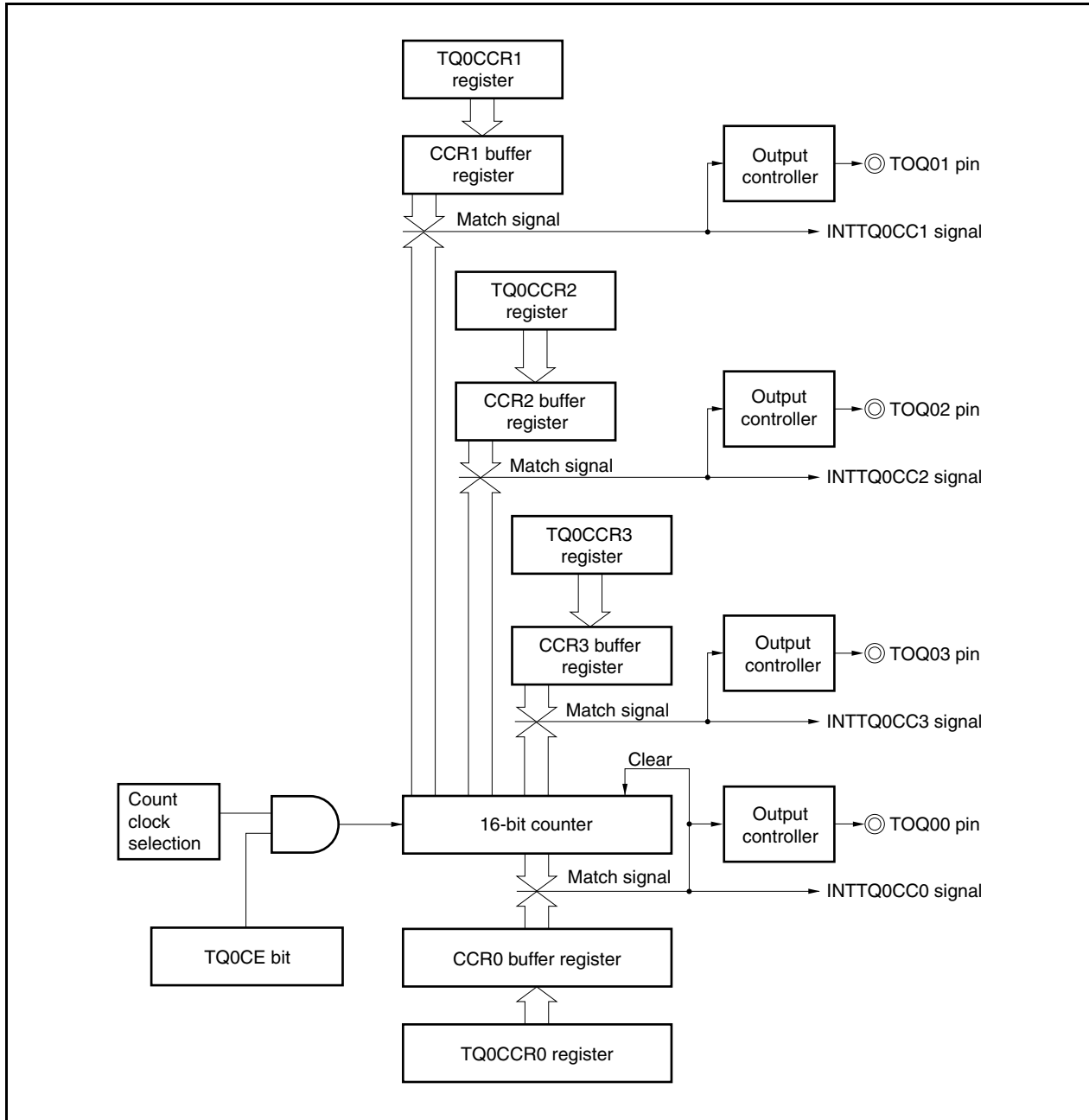
If the value of the TQ0CCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1 , the count value is transferred to the CCR0 buffer register as soon as the TQ0CCR0 register has been rewritten. Consequently, the value of the 16-bit counter that is compared is 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 " $(D_1 + 1) \times \text{Count clock cycle}$ " or " $(D_2 + 1) \times \text{Count clock cycle}$ " originally expected, but may be generated at an interval of " $(10000H + D_2 + 1) \times \text{Count clock period}$ ".

(d) Operation of TQ0CCR1 to TQ0CCR3 registers

Figure 7-6. Configuration of TQ0CCR1 to TQ0CCR3 Registers

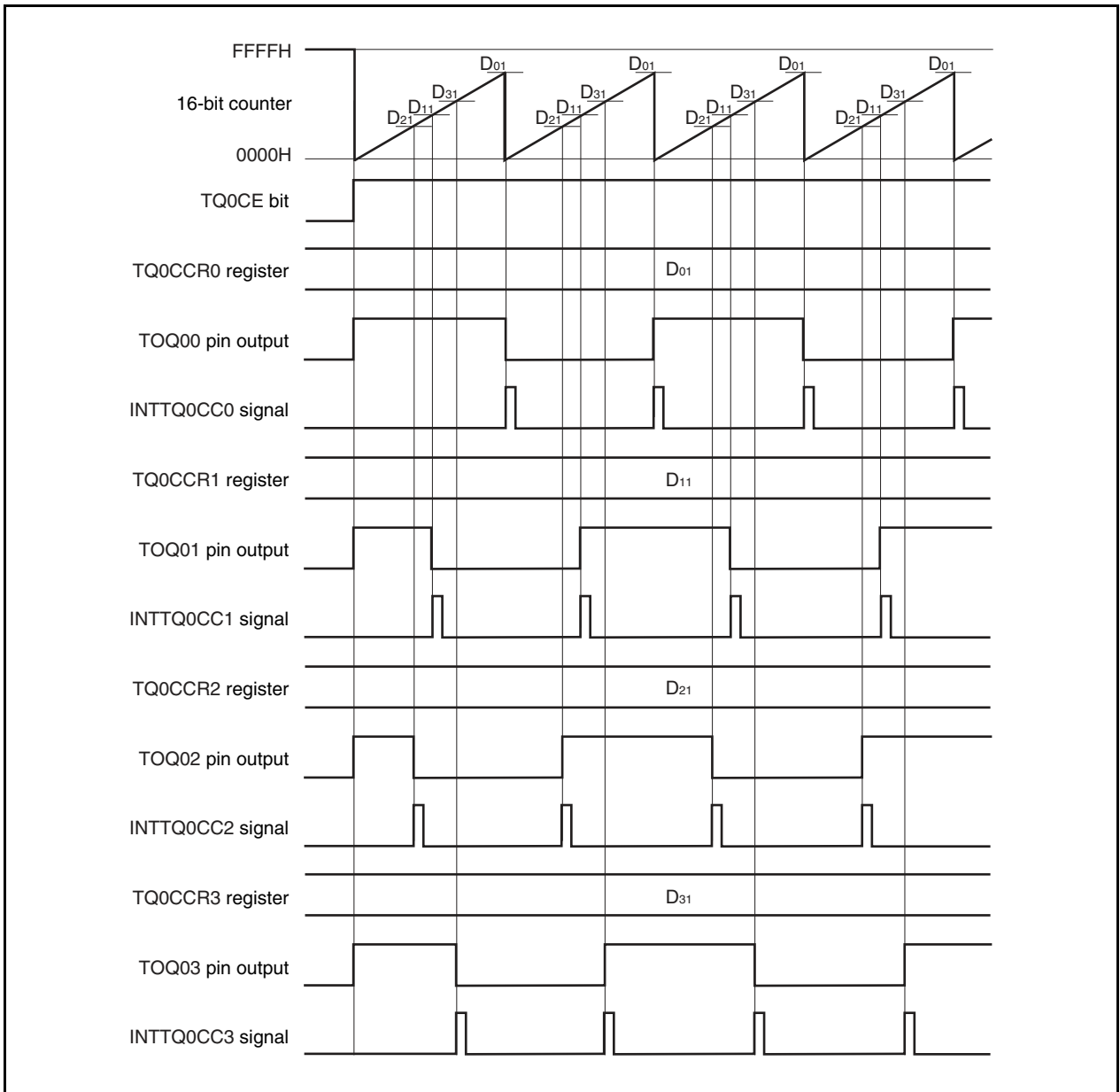


If the set value of the TQ0CCRk register is less than the set value of the TQ0CCR0 register, the INTTQ0CCk signal is generated once per cycle. At the same time, the output of the TOPQ0k pin is inverted.

The TOQ0k pin outputs a square wave with the same cycle as that output by the TOQ00 pin.

Remark k = 1 to 3

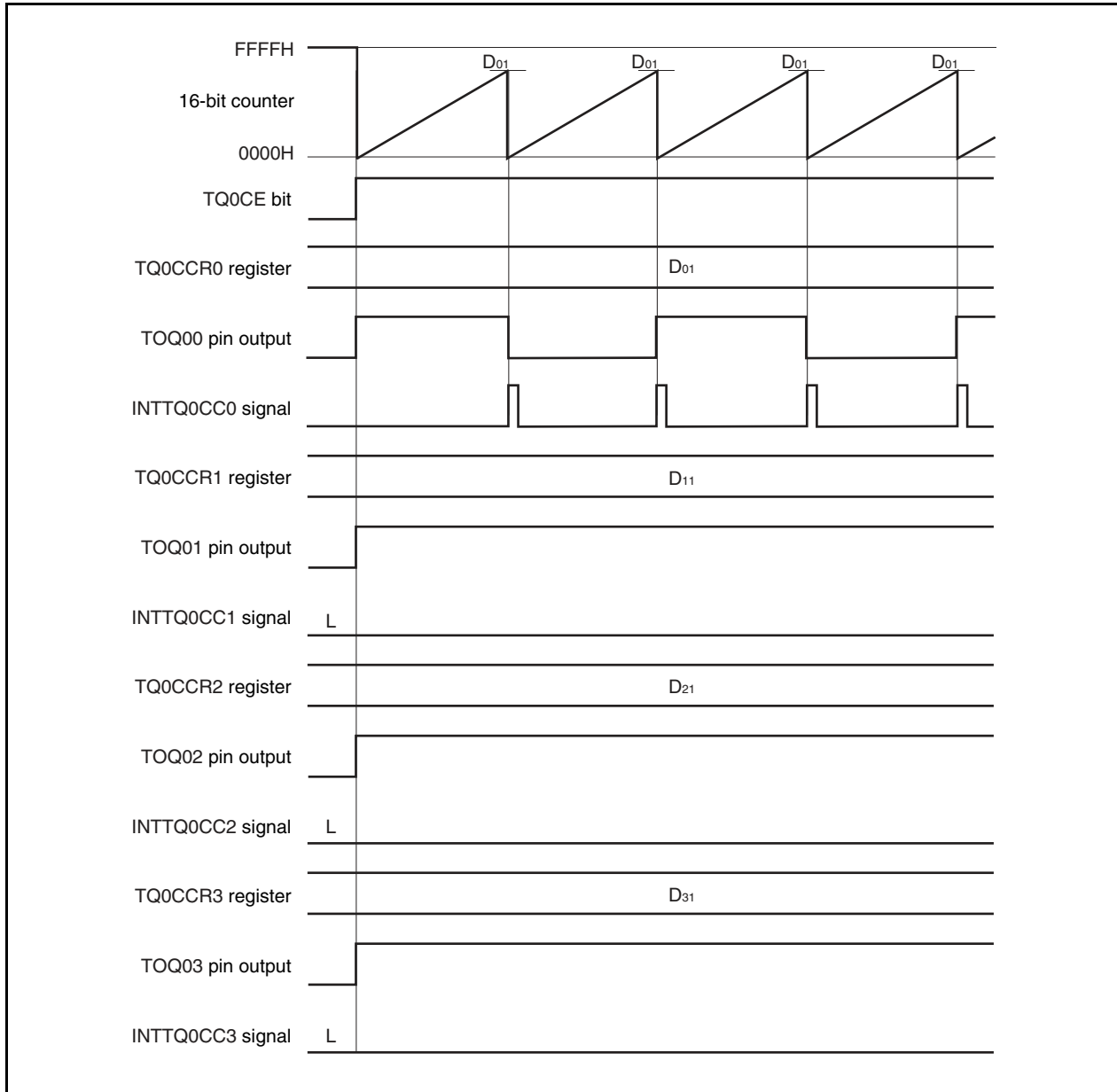
Figure 7-7. Timing Chart When $D_{01} \geq D_{k1}$



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 7-8. Timing Chart When $D_{01} < D_{k1}$



7.5.2 External event count mode (TQ0MD2 to TQ0MD0 bits = 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 TQ0Q0 pin cannot be used.

Usually, the TQ0CCR1 to TQ0CCR3 registers are not used in the external event count mode.

Figure 7-9. Configuration in External Event Count Mode

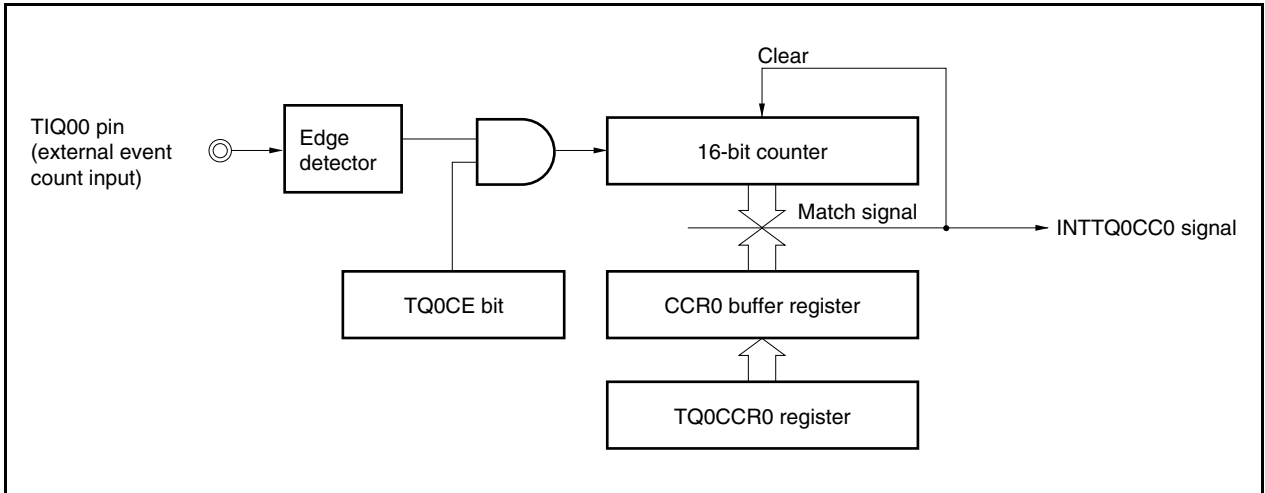
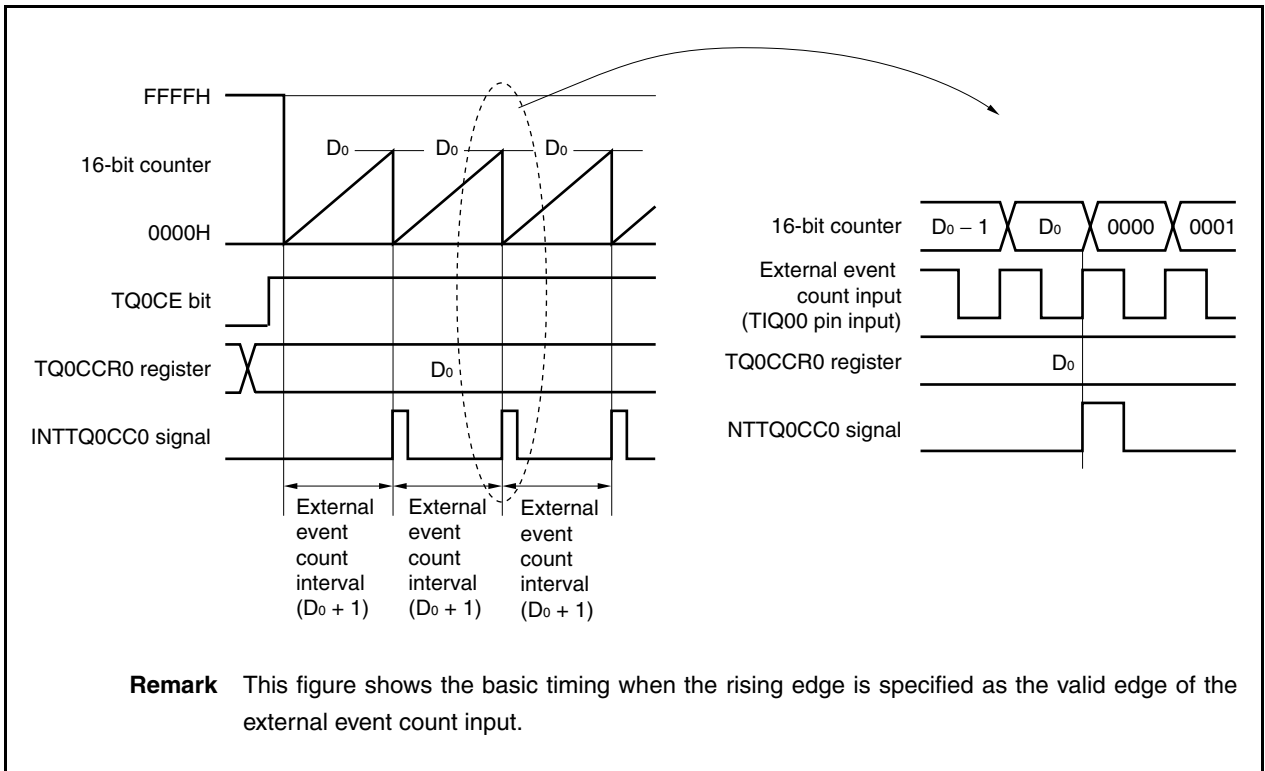


Figure 7-10. Basic Timing in External Event Count Mode



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 7-11. Register Setting for Operation in External Event Count Mode (1/2)

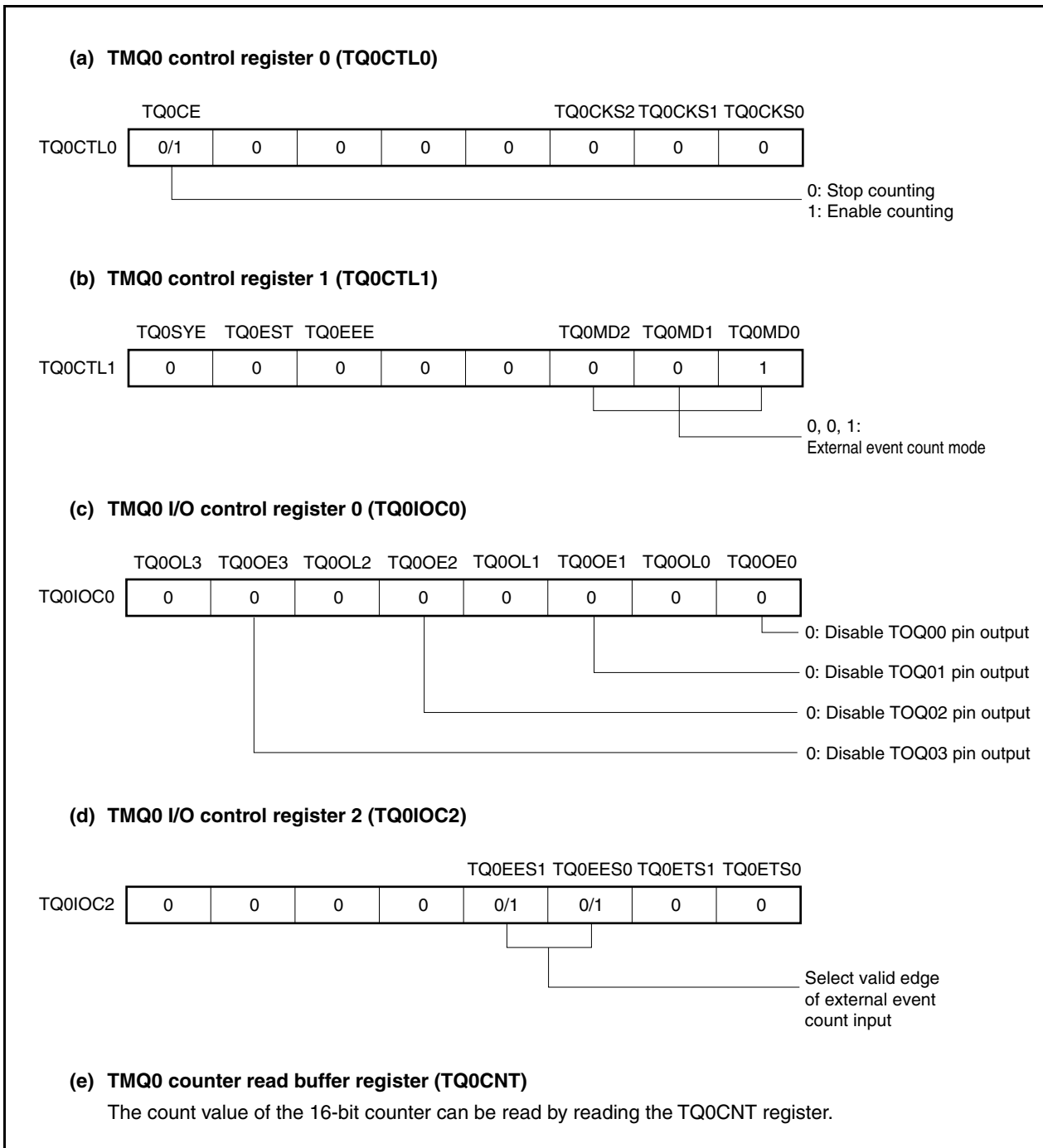


Figure 7-11. Register Setting for Operation in External Event Count Mode (2/2)

(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 value 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).

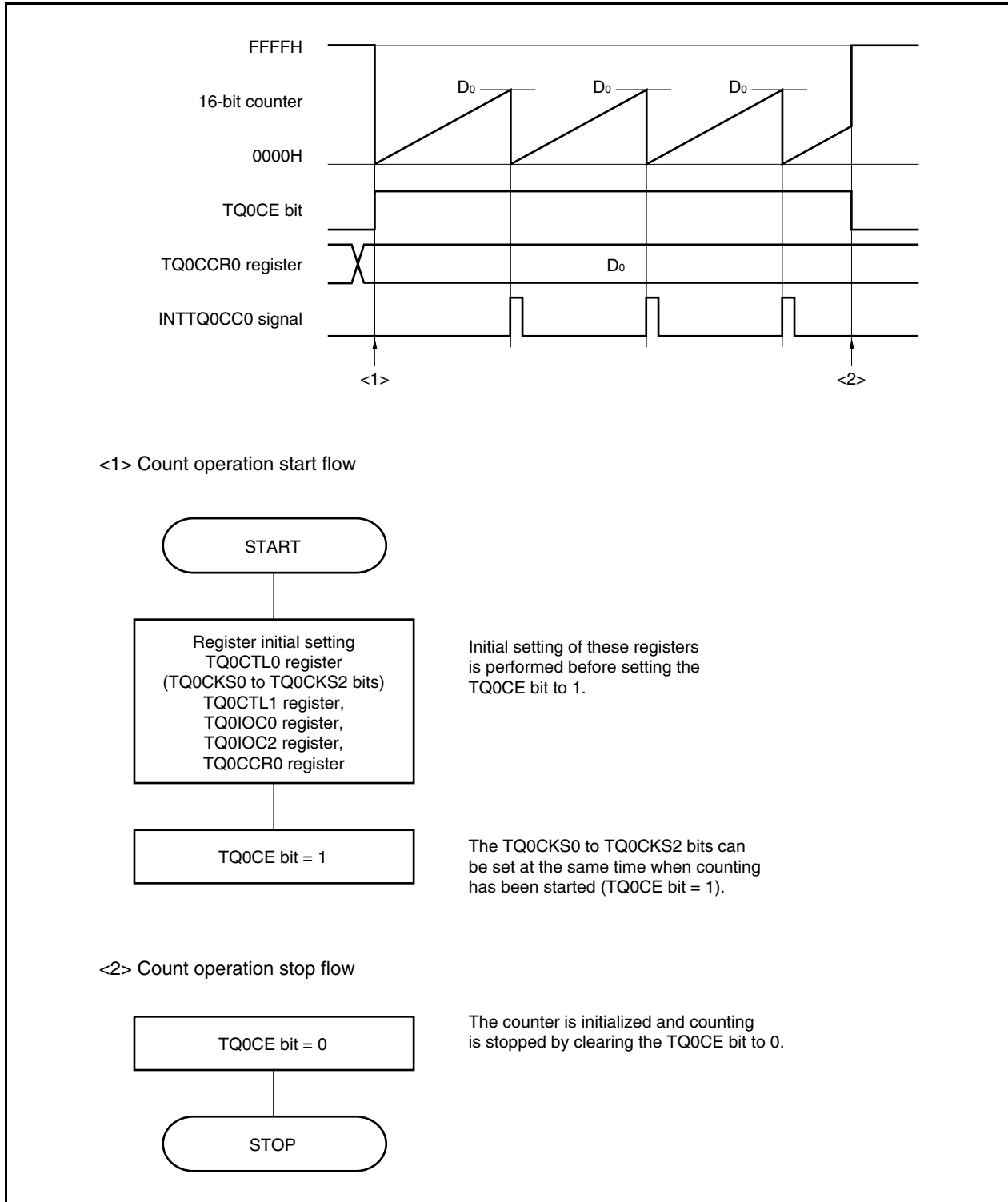
Caution When an external clock is used as the count clock, the external clock can be input only from the TIQ00 pin. At this time, set the TQ0IOC1.TQ0IS1 and TQ0IOC1.TQ0IS0 bits to 00 (capture trigger input (TIQ00 pin): no edge detection).

Remark The TMQ0 I/O control register 1 (TQ0IOC1) and TMQ0 option register 0 (TQ0OPT0) are not used in the external event count mode.

<R>

(1) External event count mode operation flow

Figure 7-12. Flow of Software Processing in External Event Count Mode



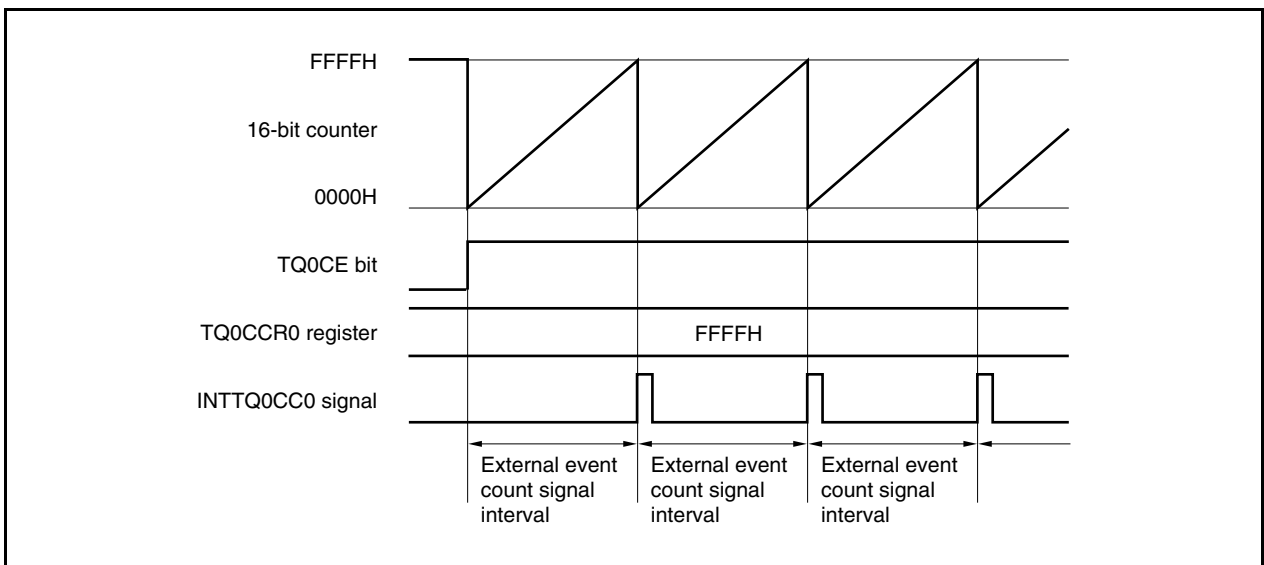
(2) Operation timing in external event count mode

Cautions 1. In the external event count mode, do not set the TQ0CCR0 register to 0000H.

2. In the external event count mode, use of the timer output is disabled. If performing timer output using external event count input, set the interval timer mode, and select the operation enabled by the external event count input for the count clock (TQ0CTL1.TQ0MD2 to TQ0CTL1.TQ0MD0 bits = 000, TQ0CTL1.TQ0EEE bit = 1).

(a) Operation if TQ0CCR0 register is set to FFFFH

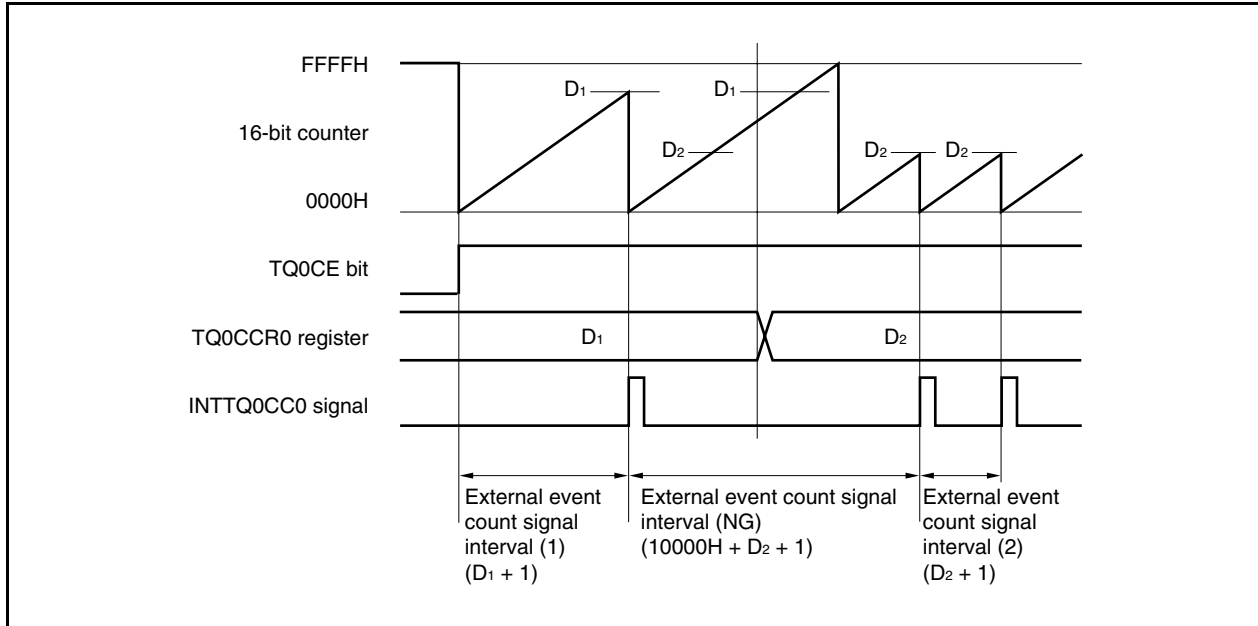
If the TQ0CCR0 register is set to FFFFH, the 16-bit counter counts to FFFFH each time the valid edge of the external event count signal has been detected. The 16-bit counter is cleared to 0000H in synchronization with the next count-up timing, and the INTTQ0CC0 signal is generated. At this time, the TQ0OPT0.TQ0OVF bit is not set.



(b) 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.



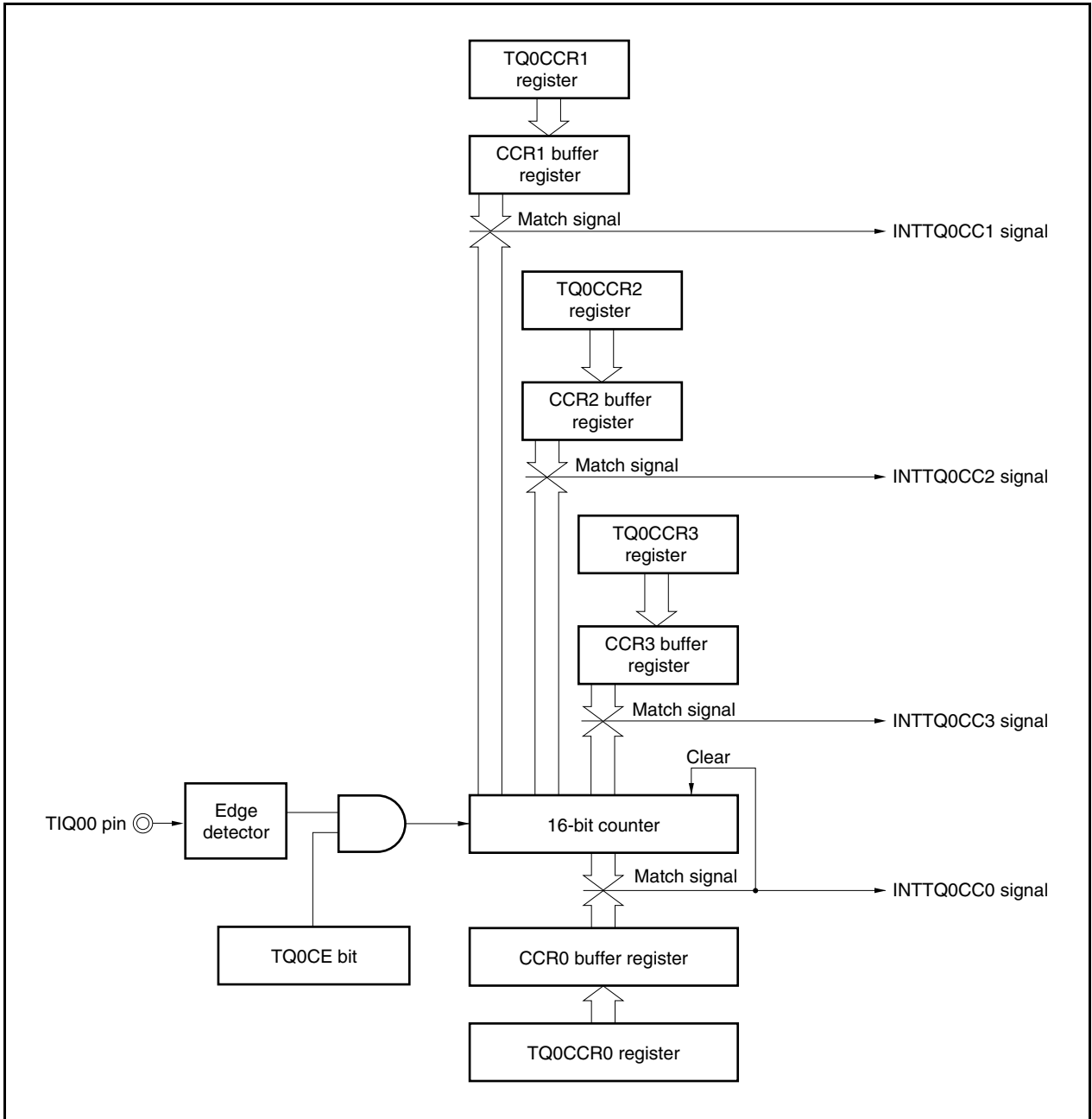
If the value of the TQ0CCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1 , the count value is transferred to the CCR0 buffer register as soon as the TQ0CCR0 register has been rewritten. Consequently, the value that is compared with the 16-bit counter 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”.

(c) Operation of TQ0CCR1 to TQ0CCR3 registers

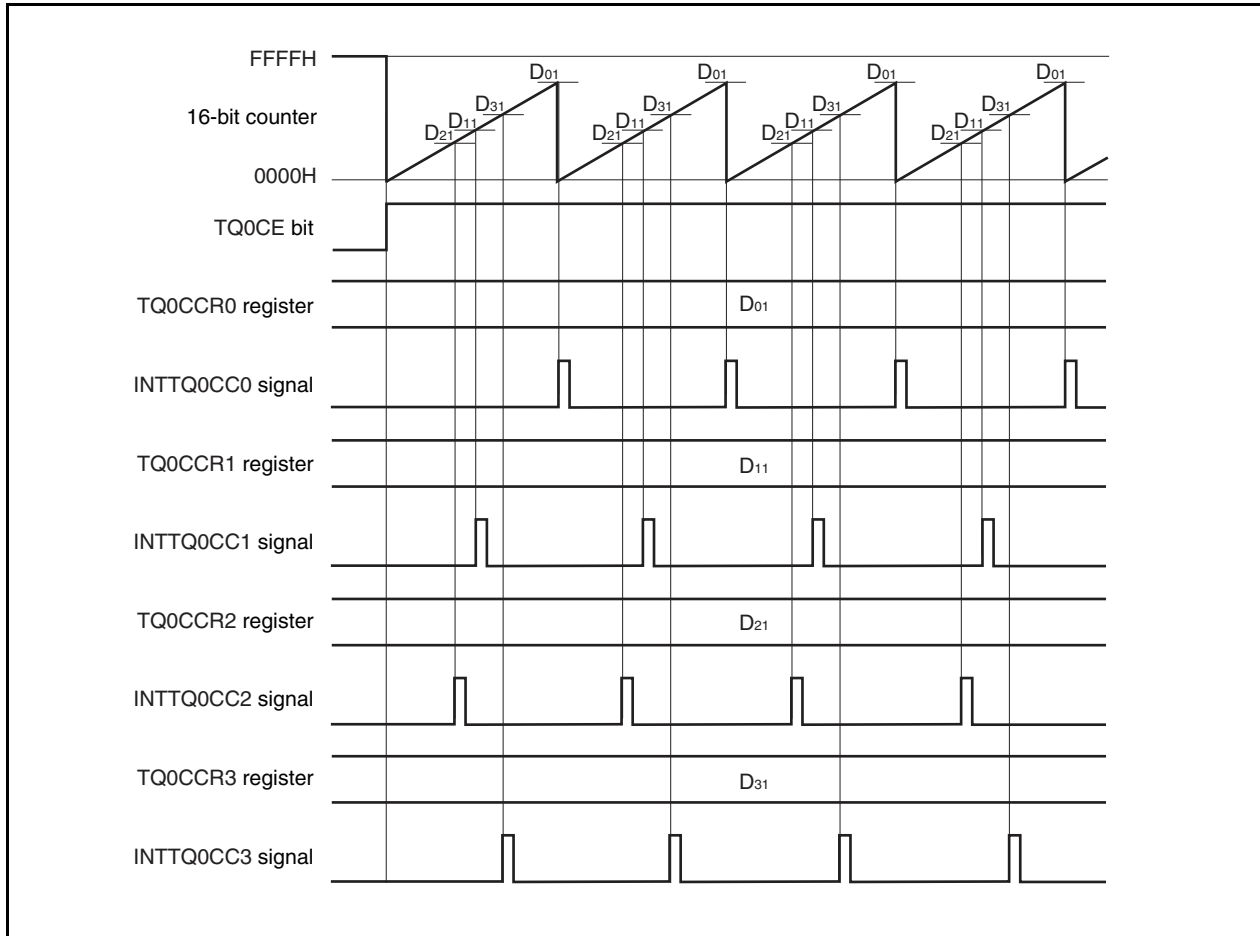
Figure 7-13. Configuration of TQ0CCR1 to TQ0CCR3 Registers



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.

Remark k = 1 to 3

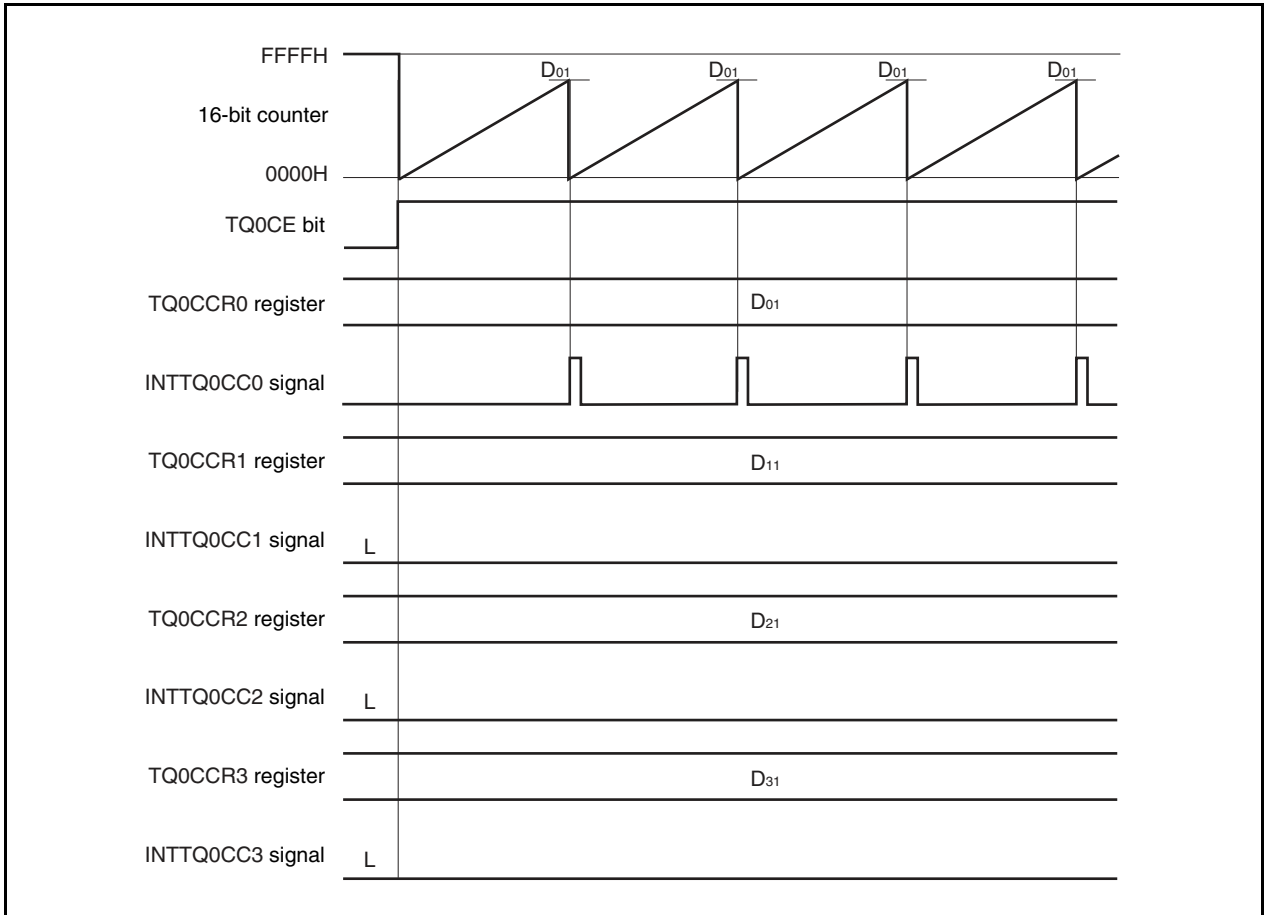
Figure 7-14. Timing Chart When $D_{01} \geq D_{k1}$



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.

Remark k = 1 to 3

Figure 7-15. Timing Chart When $D_{01} < D_{k1}$



7.5.3 External trigger pulse output mode (TQ0MD2 to TQ0MD0 bits = 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 7-16. Configuration in External Trigger Pulse Output Mode

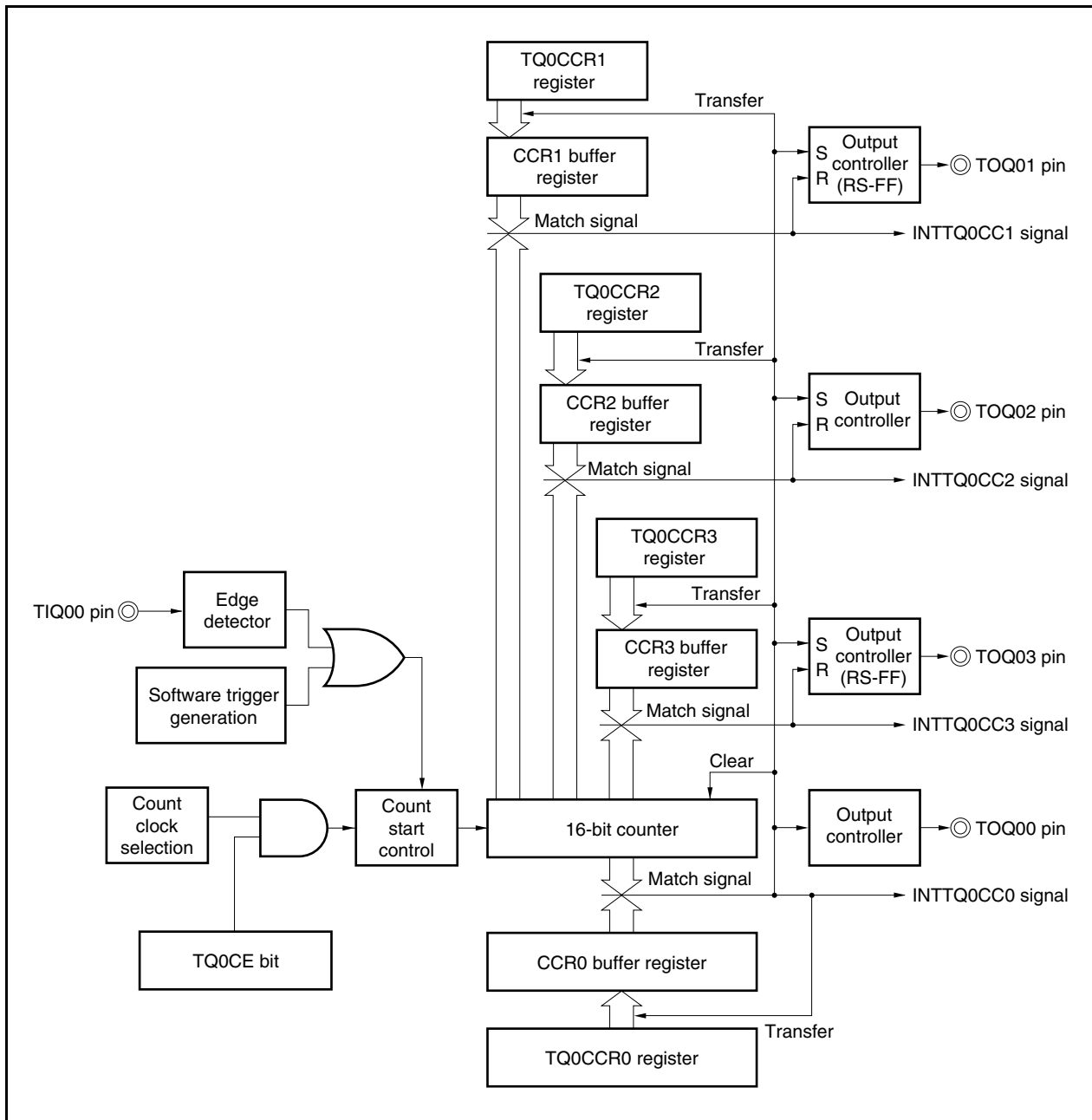
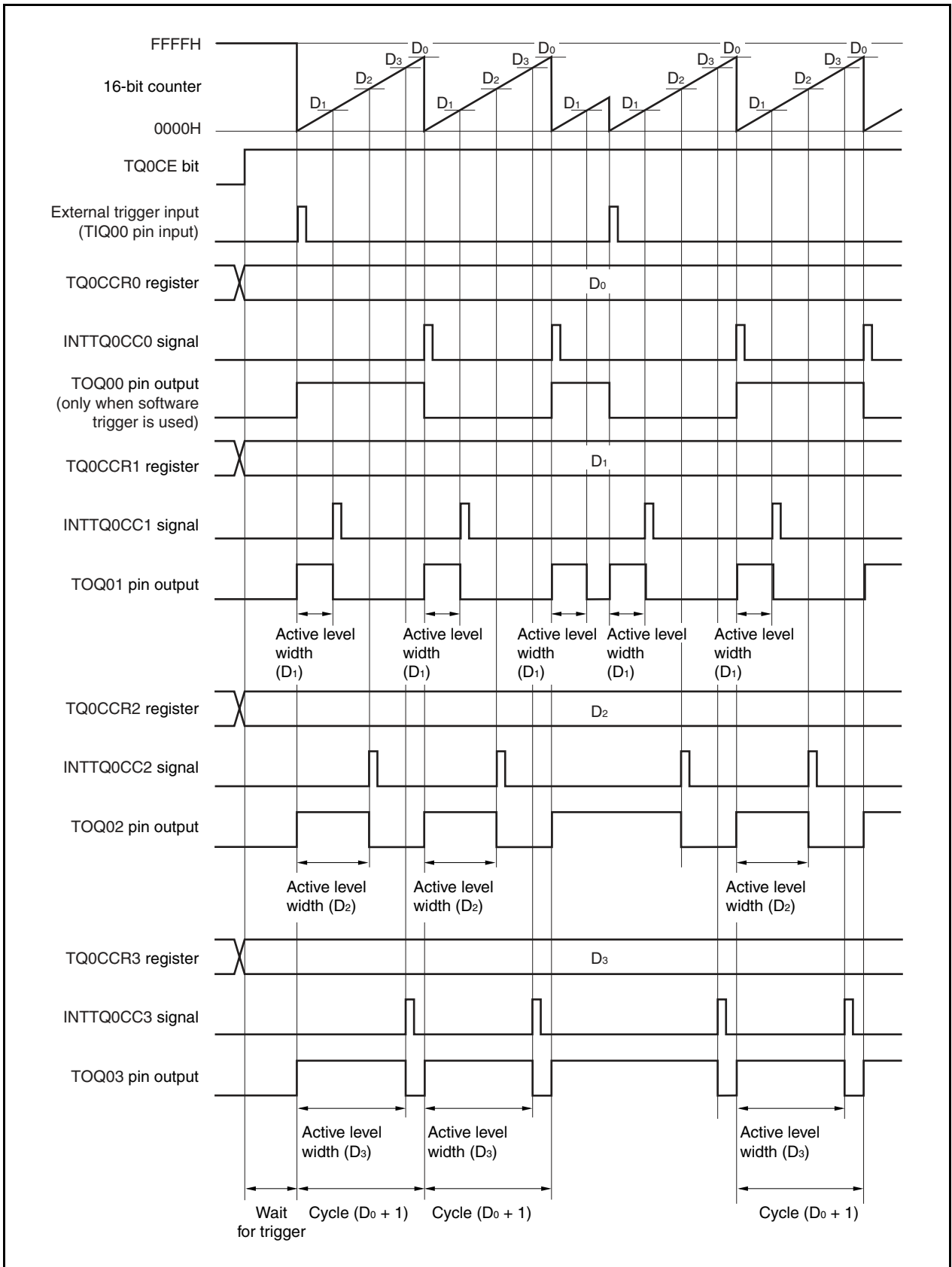


Figure 7-17. Basic Timing in External Trigger Pulse Output Mode



16-bit timer/event counter 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 output of the TOQ00 pin is inverted. The TOQ0k pin outputs a high-level regardless of the status (high/low) when a trigger is generated.)

The active level width, cycle, and duty factor of the PWM waveform can be calculated as follows.

$$\text{Active level width} = (\text{Set value of TQ0CCRk register}) \times \text{Count clock cycle}$$

$$\text{Cycle} = (\text{Set value of TQ0CCR0 register} + 1) \times \text{Count clock cycle}$$

$$\text{Duty factor} = (\text{Set value of TQ0CCRk register}) / (\text{Set value of TQ0CCR0 register} + 1)$$

The compare match 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 7-18. Setting of Registers in External Trigger Pulse Output Mode (1/3)

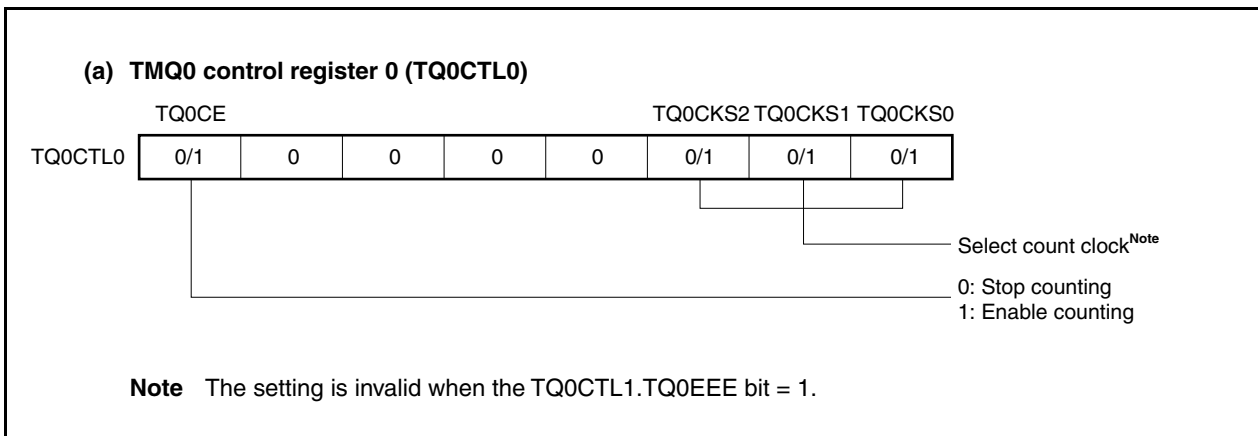


Figure 7-18. Setting of Registers in External Trigger Pulse Output Mode (2/3)

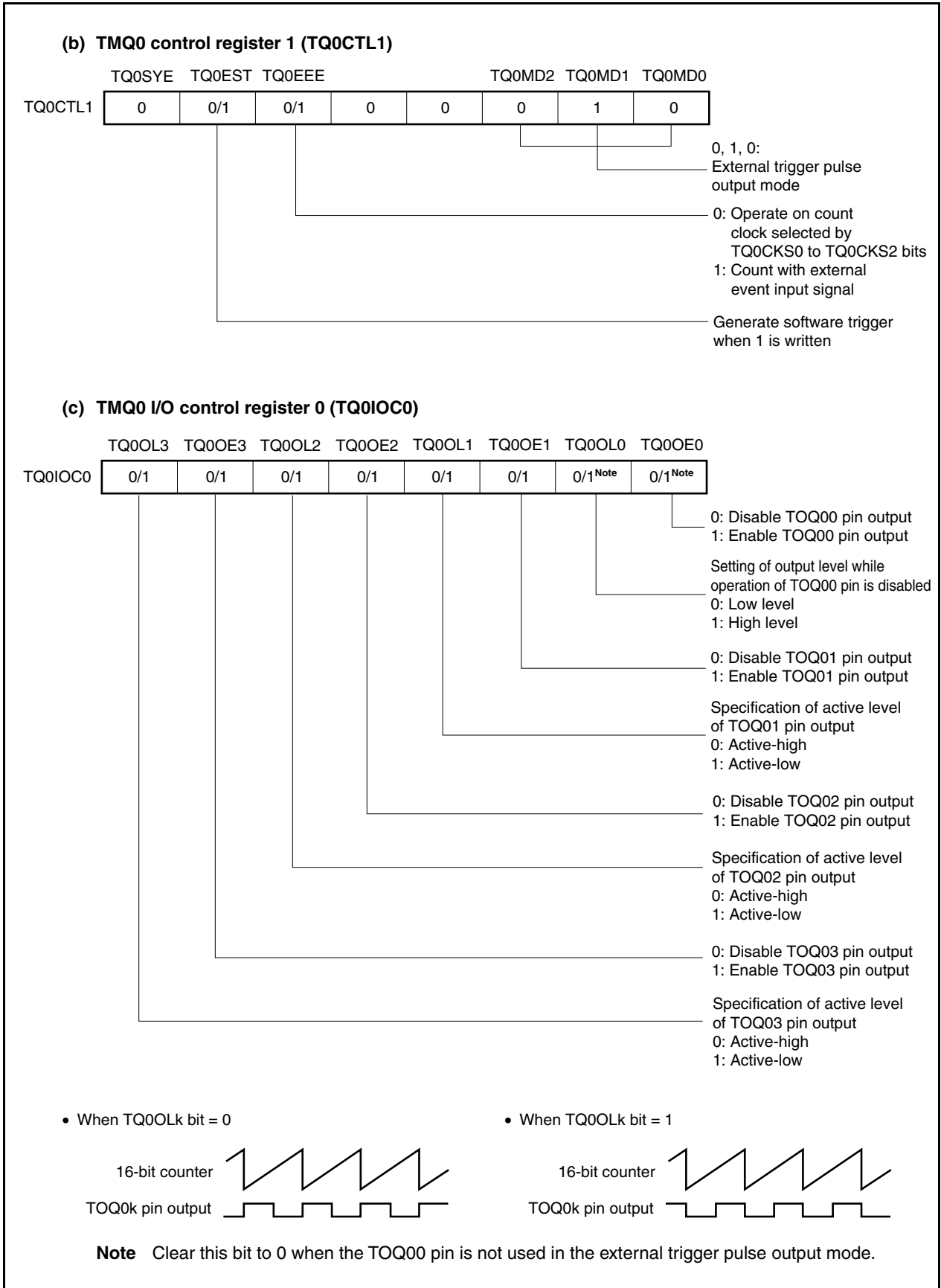
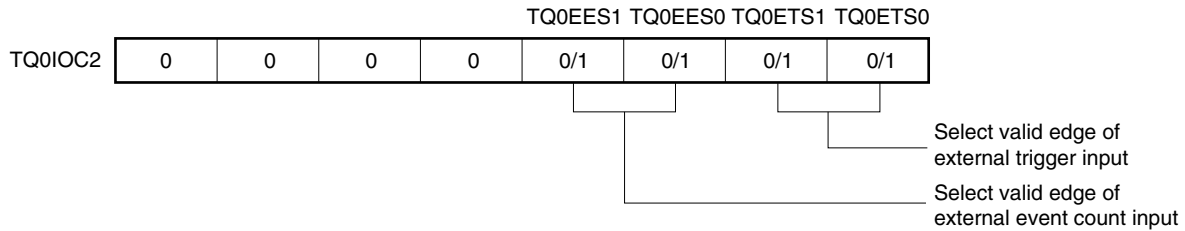


Figure 7-18. Setting of Registers in External Trigger Pulse Output Mode (3/3)

(d) TMQ0 I/O control register 2 (TQ0IOC2)**(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 of the PWM waveform are as follows.

$$\text{Cycle} = (D_0 + 1) \times \text{Count clock cycle}$$

$$\text{TOQ01 pin PWM waveform active level width} = D_1 \times \text{Count clock cycle}$$

$$\text{TOQ02 pin PWM waveform active level width} = D_2 \times \text{Count clock cycle}$$

$$\text{TOQ03 pin PWM waveform active level width} = D_3 \times \text{Count clock cycle}$$

Remarks 1. TMQ0 I/O control register 1 (TQ0IOC1) and TMQ0 option register 0 (TQ0OPT0) are not used in the external trigger pulse output mode.

2. Updating TMQ0 capture/compare register 2 (TQ0CCR2) and TMQ0 capture/compare register 3 (TQ0CCR3) is validated by writing TMQ0 capture/compare register 1 (TQ0CCR1).

(1) Operation flow in external trigger pulse output mode

Figure 7-19. Software Processing Flow in External Trigger Pulse Output Mode (1/2)

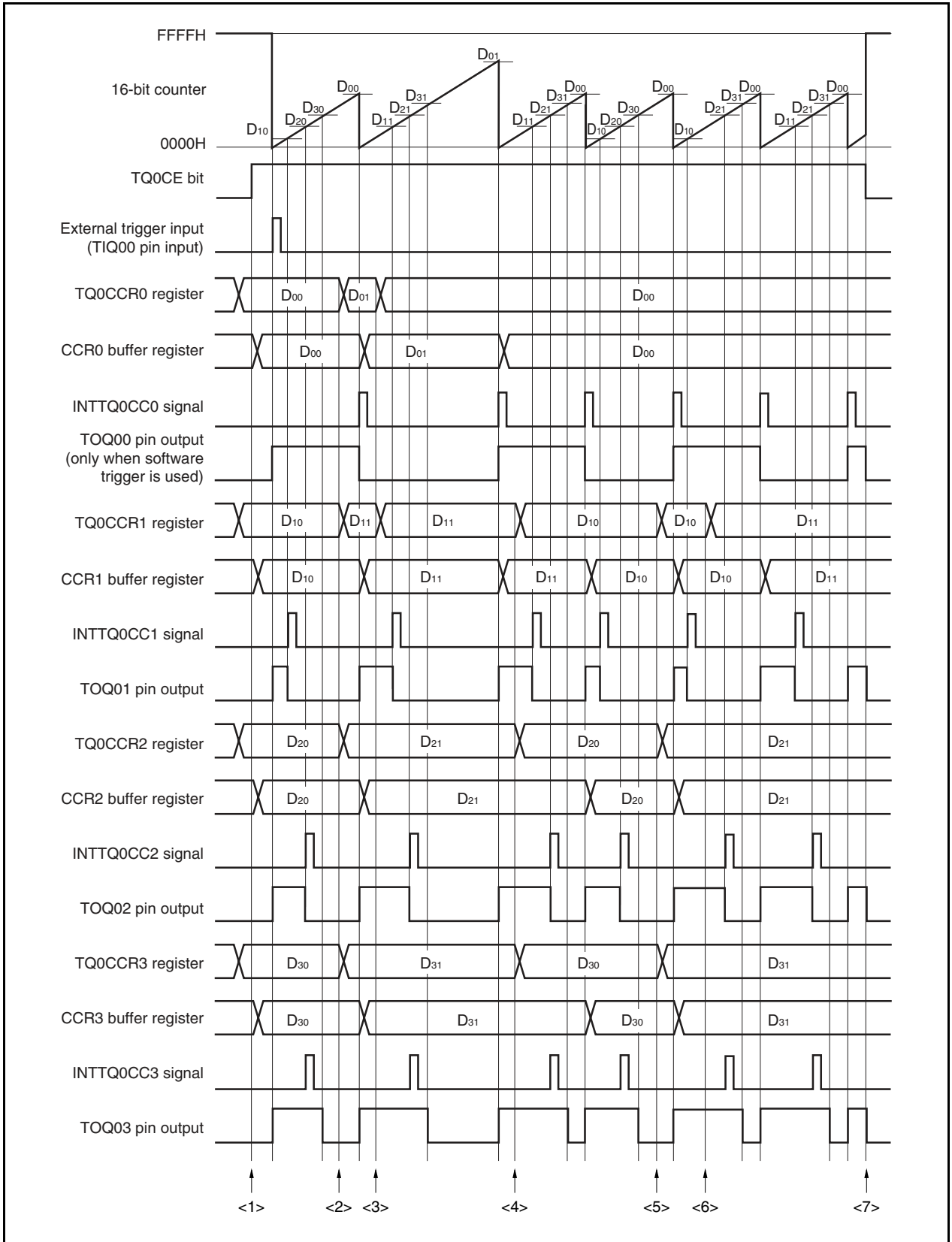
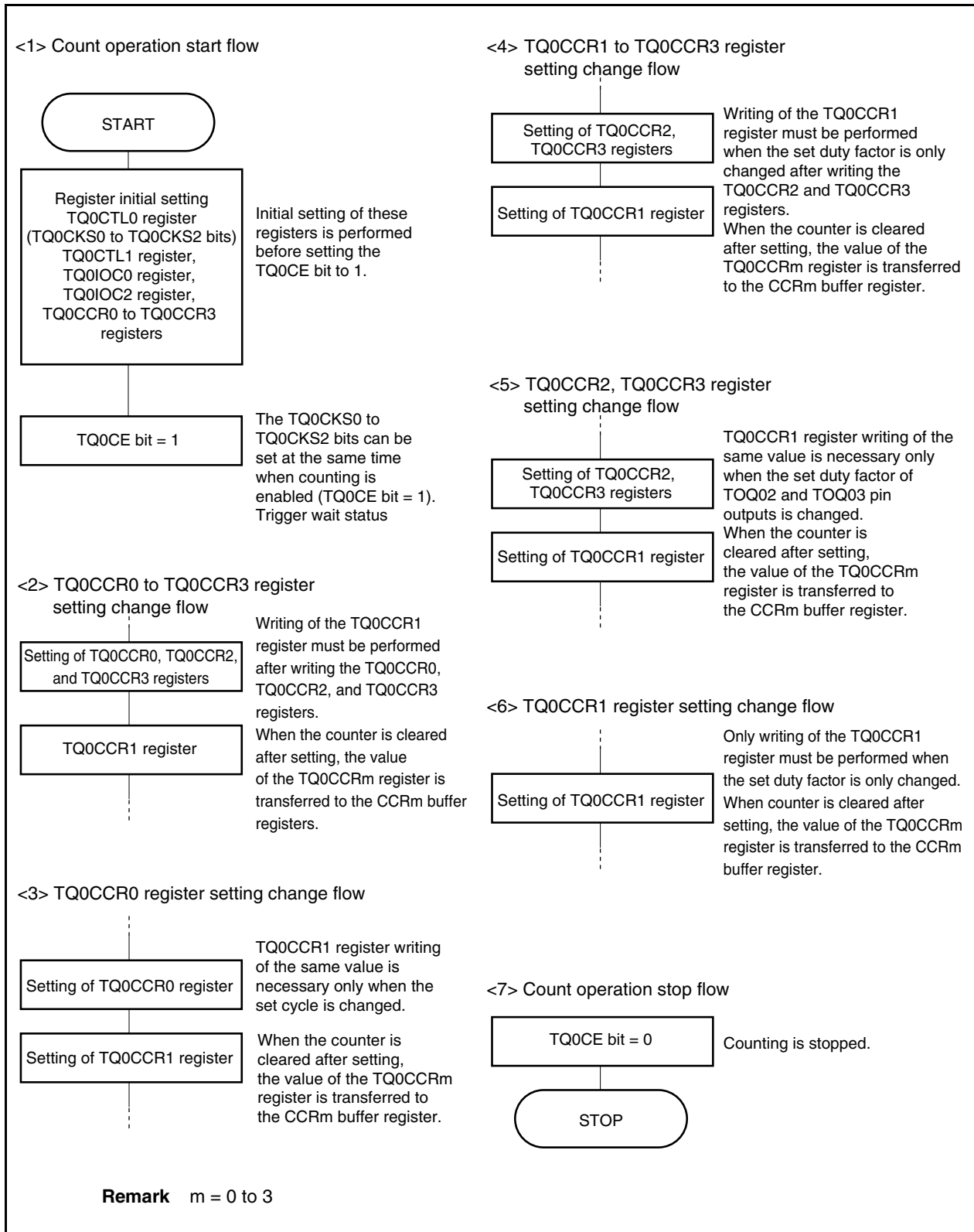


Figure 7-19. Software Processing Flow in External Trigger Pulse Output Mode (2/2)

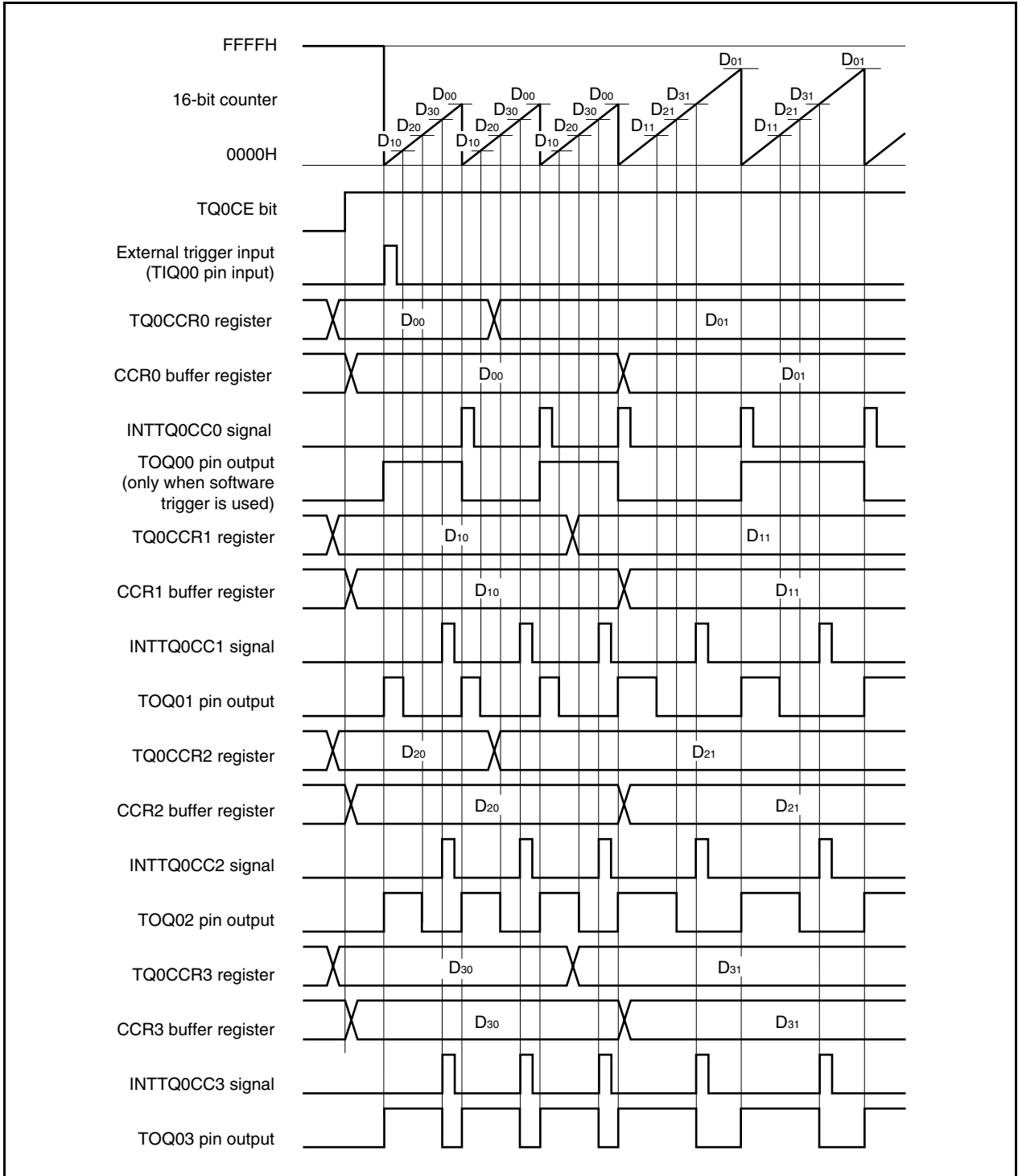


(2) External trigger pulse output mode operation timing

(a) Note on changing pulse width during operation

To change the PWM waveform while the counter is operating, write the 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 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 to the TQ0CCR2 and TQ0CCR3 registers and then set an active level 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.

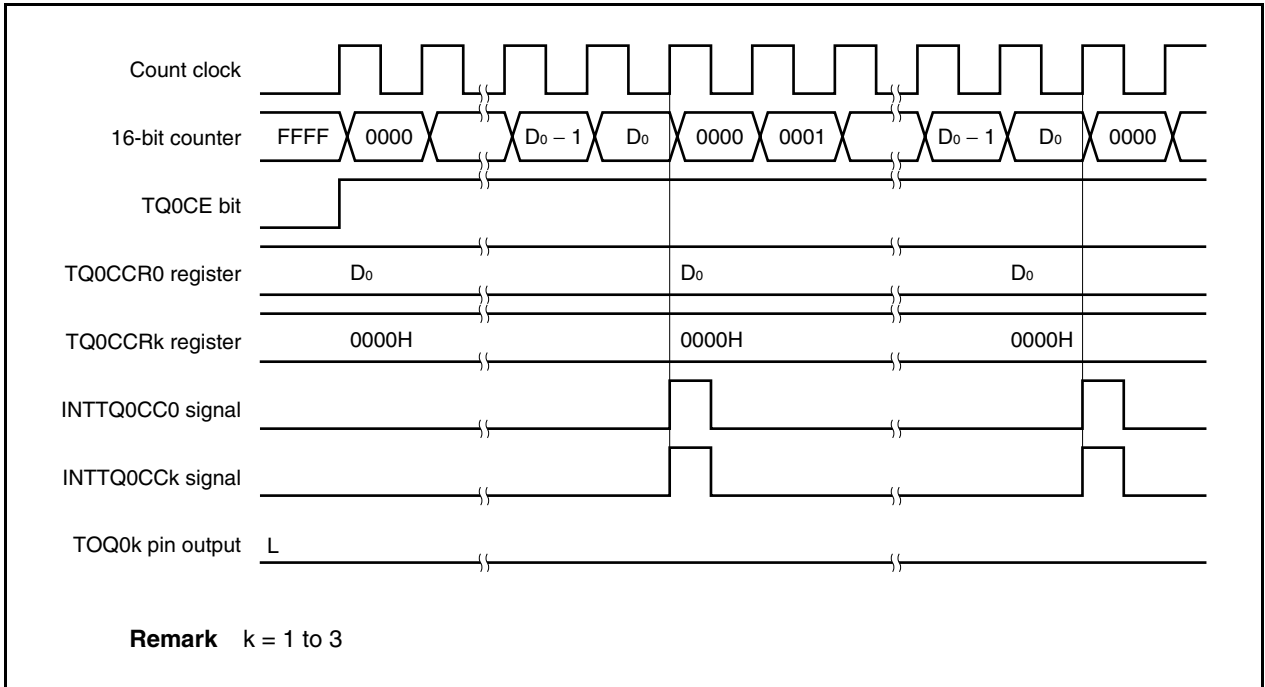
After data is written to the TQ0CCR1 register, the value written to the TQ0CCRm register is transferred to the CCRm buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

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.

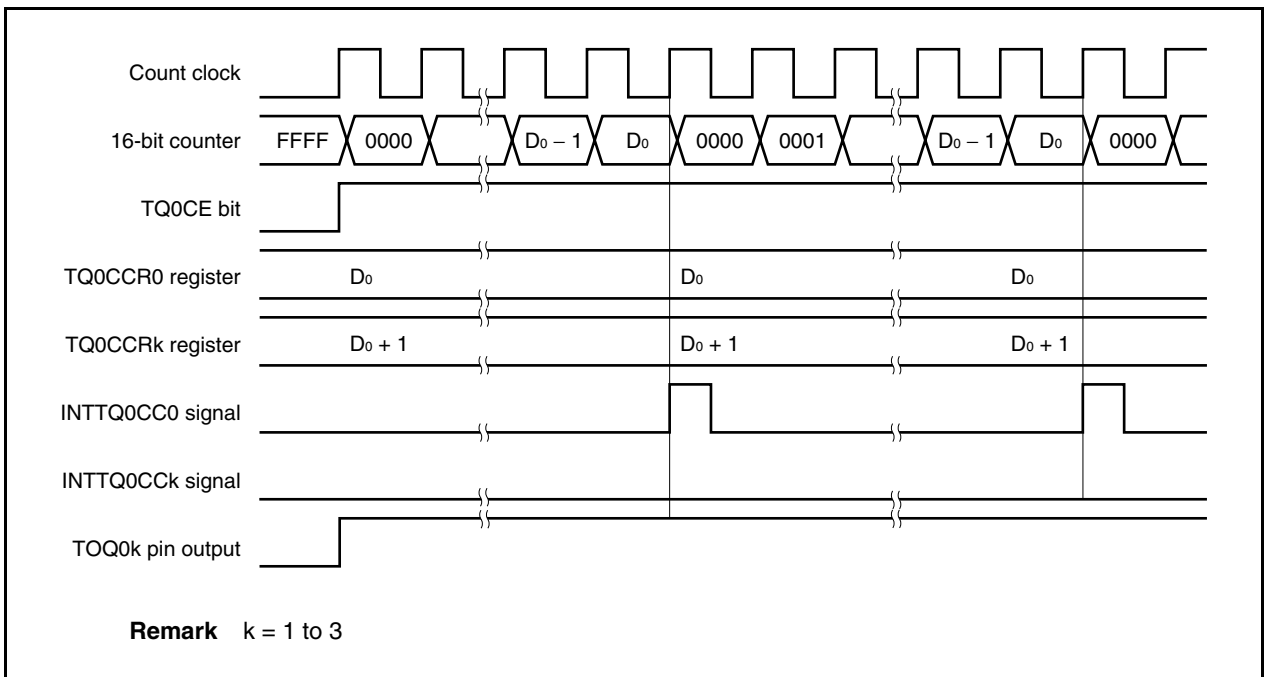
Remark m = 0 to 3

(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.

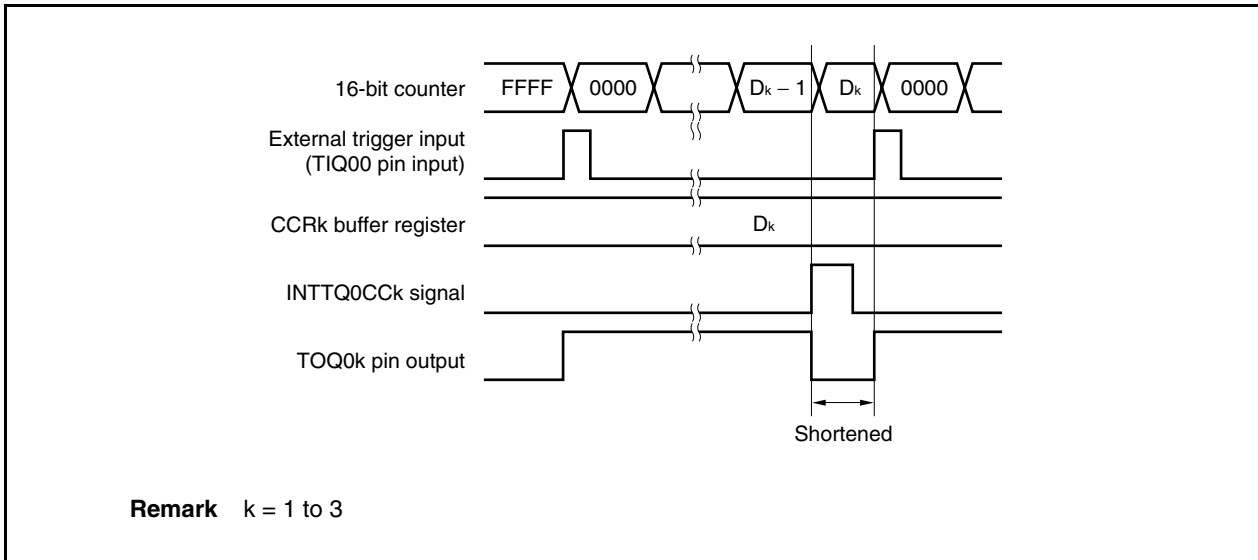


To output a 100% waveform, set a value of (set value of TQ0CCR0 register + 1) to the TQ0CCRk register. If the set value of the TQ0CCR0 register is FFFFH, 100% output cannot be produced.

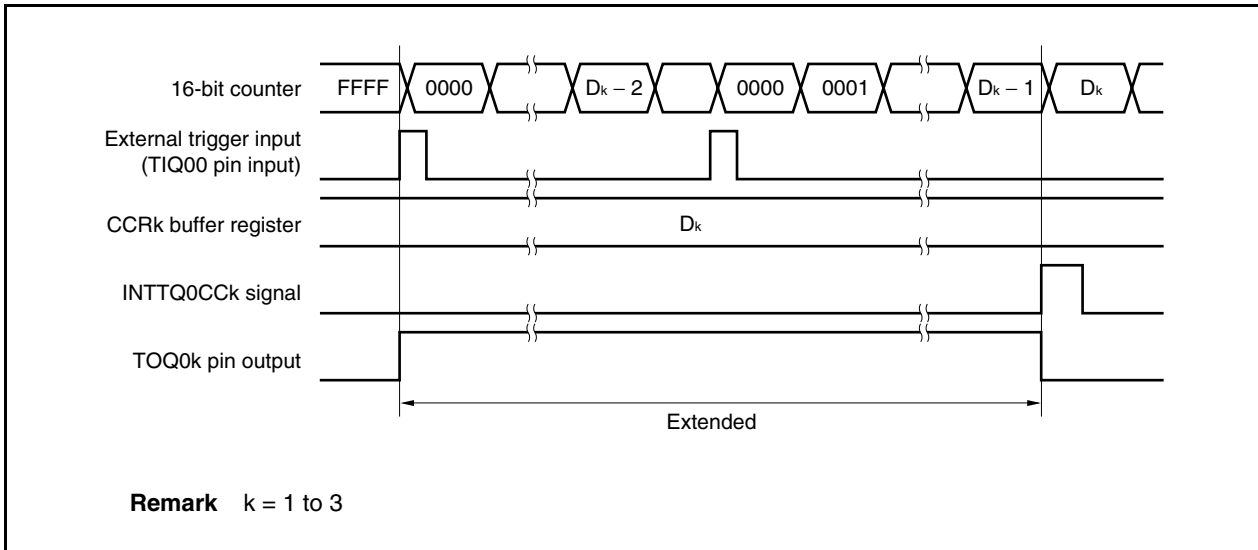


(c) Conflict between trigger detection and match with CCRk buffer register

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 is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.

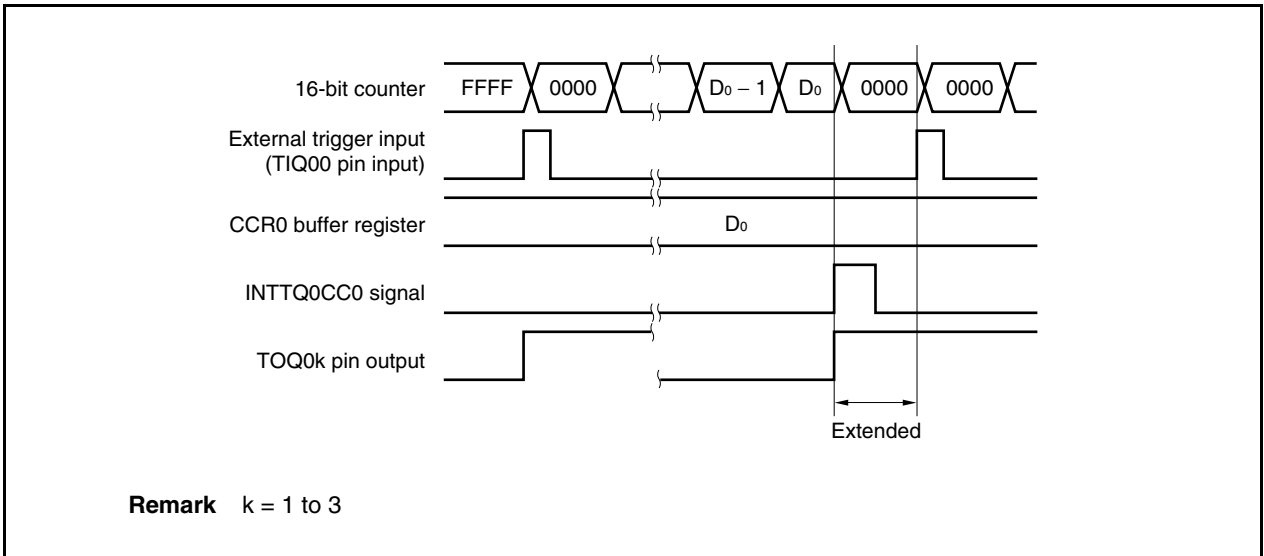


If the trigger is detected immediately before the INTTQ0CCk signal is generated, the INTTQ0CCk signal is not generated, and the 16-bit counter is cleared to 0000H and continues counting. The output signal of the TOQ0k pin remains active. Consequently, the active period of the PWM waveform is extended.

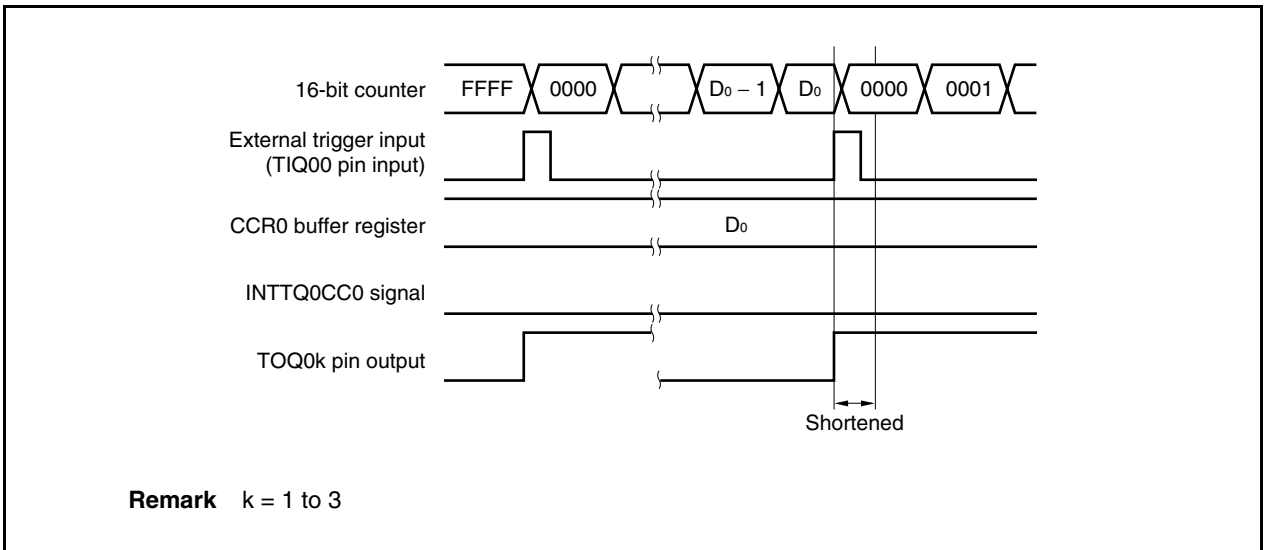


(d) Conflict between trigger detection and match with CCR0 buffer register

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.

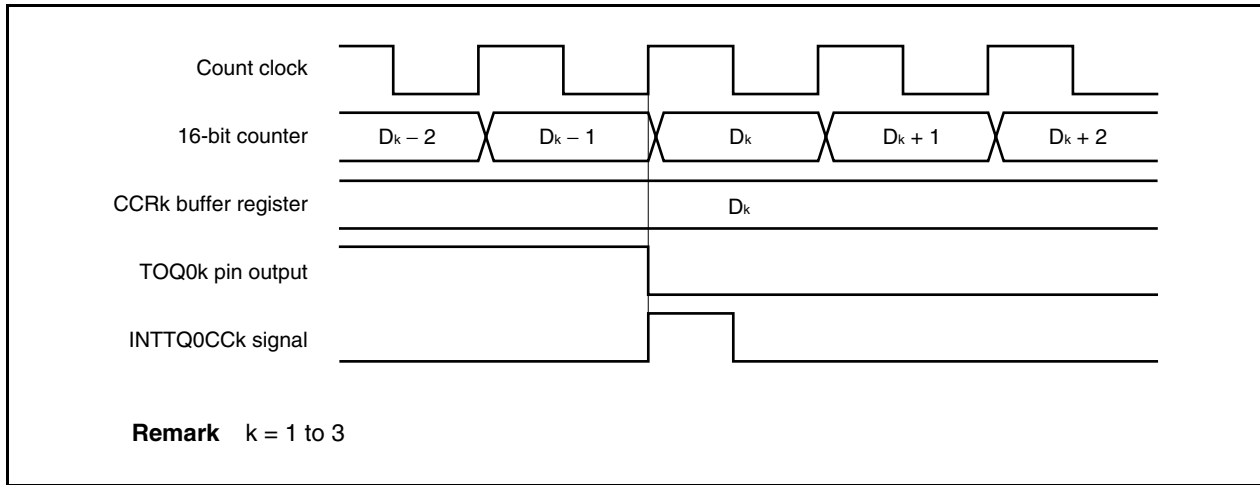


If the trigger is detected immediately before the INTTQ0CC0 signal is generated, the INTTQ0CC0 signal is not generated. The 16-bit counter is cleared to 0000H, the TOQ0k pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



(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.



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.

7.5.4 One-shot pulse output mode (TQ0MD2 to TQ0MD0 bits = 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 7-20. Configuration in One-Shot Pulse Output Mode

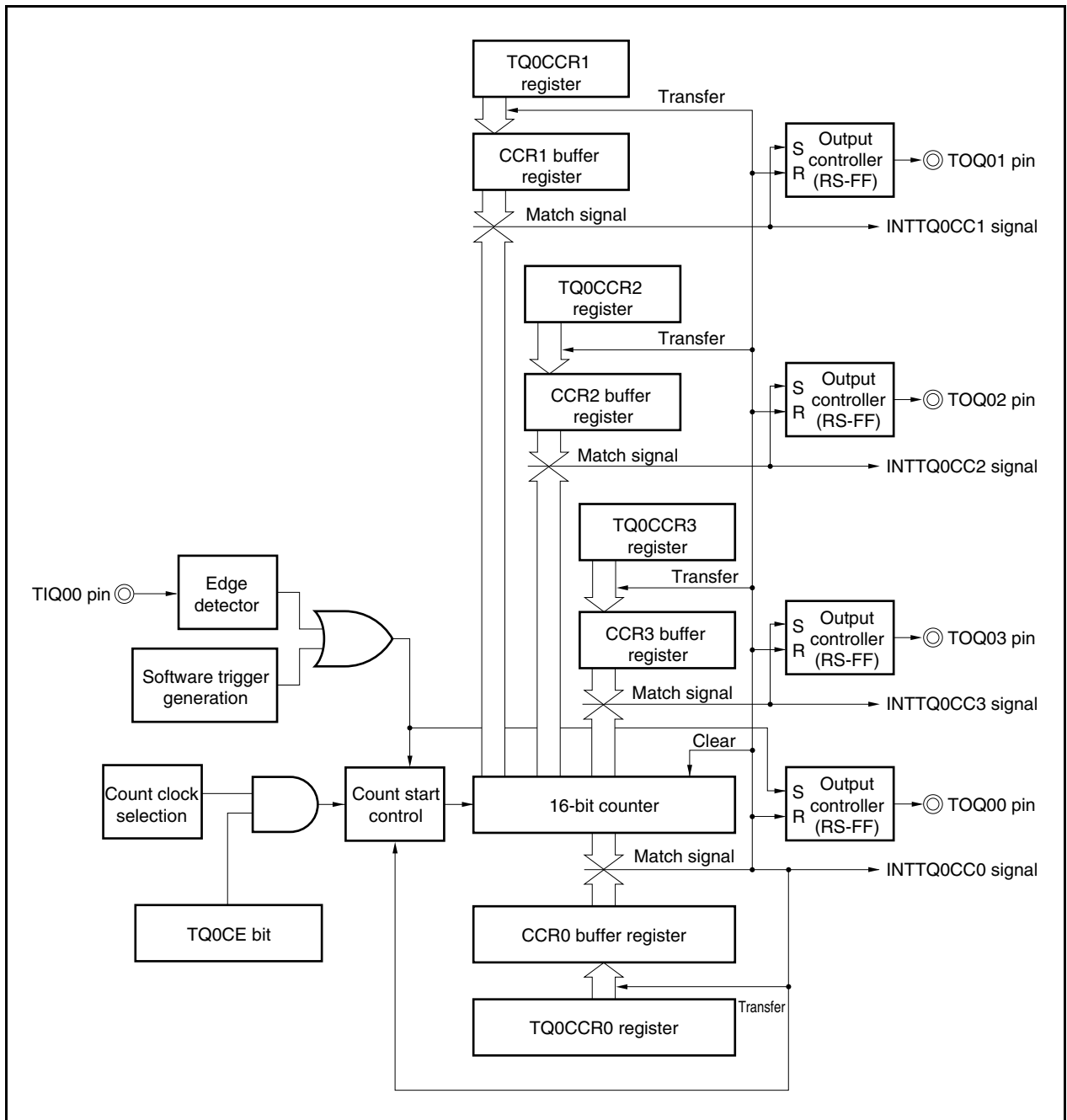
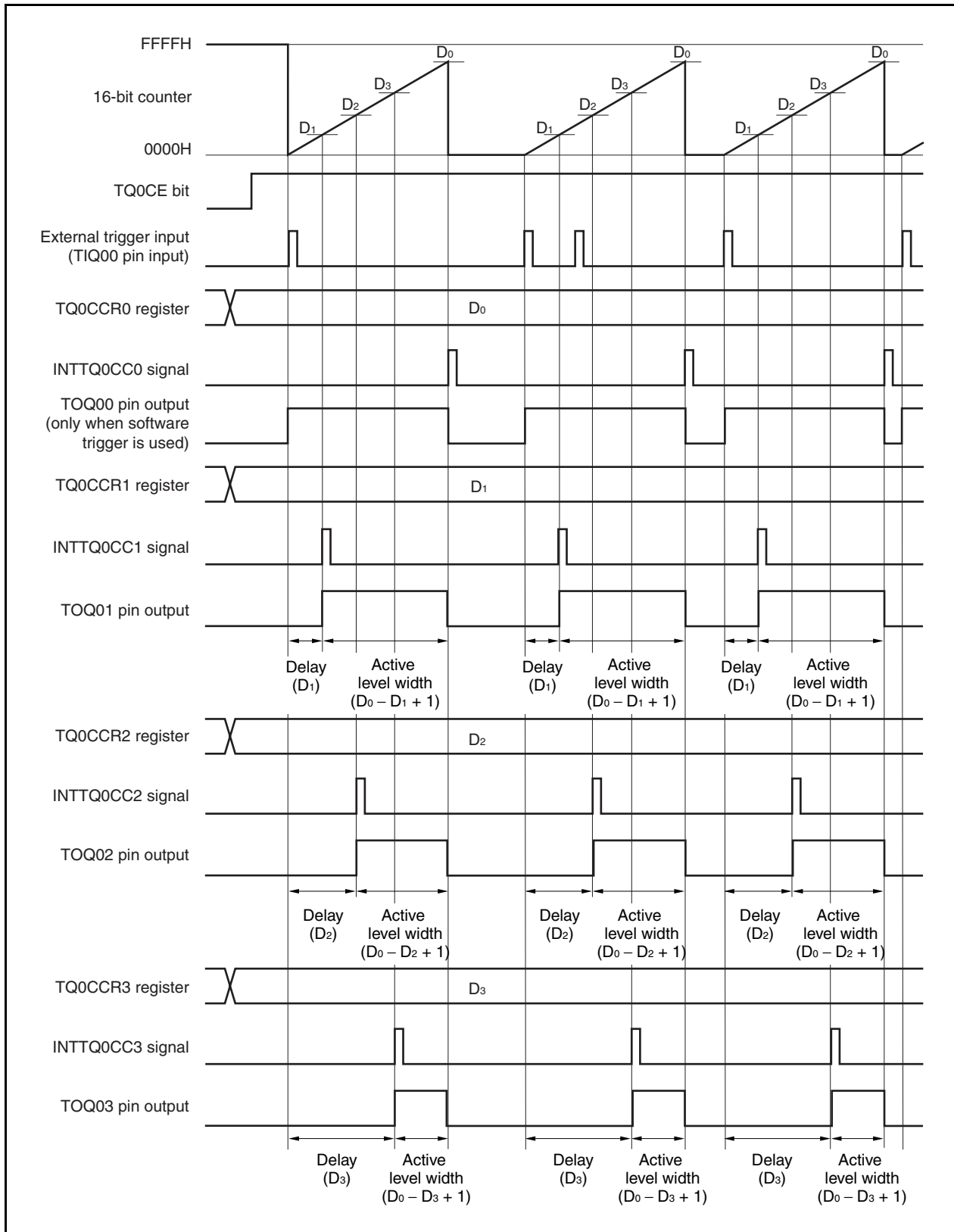


Figure 7-21. Basic Timing in One-Shot Pulse Output Mode



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.

$$\text{Output delay period} = (\text{Set value of TQ0CCRk register}) \times \text{Count clock cycle}$$

$$\text{Active level width} = (\text{Set value of TQ0CCR0 register} - \text{Set value of TQ0CCRk register} + 1) \times \text{Count clock cycle}$$

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 7-22. Setting of Registers in One-Shot Pulse Output Mode (1/3)

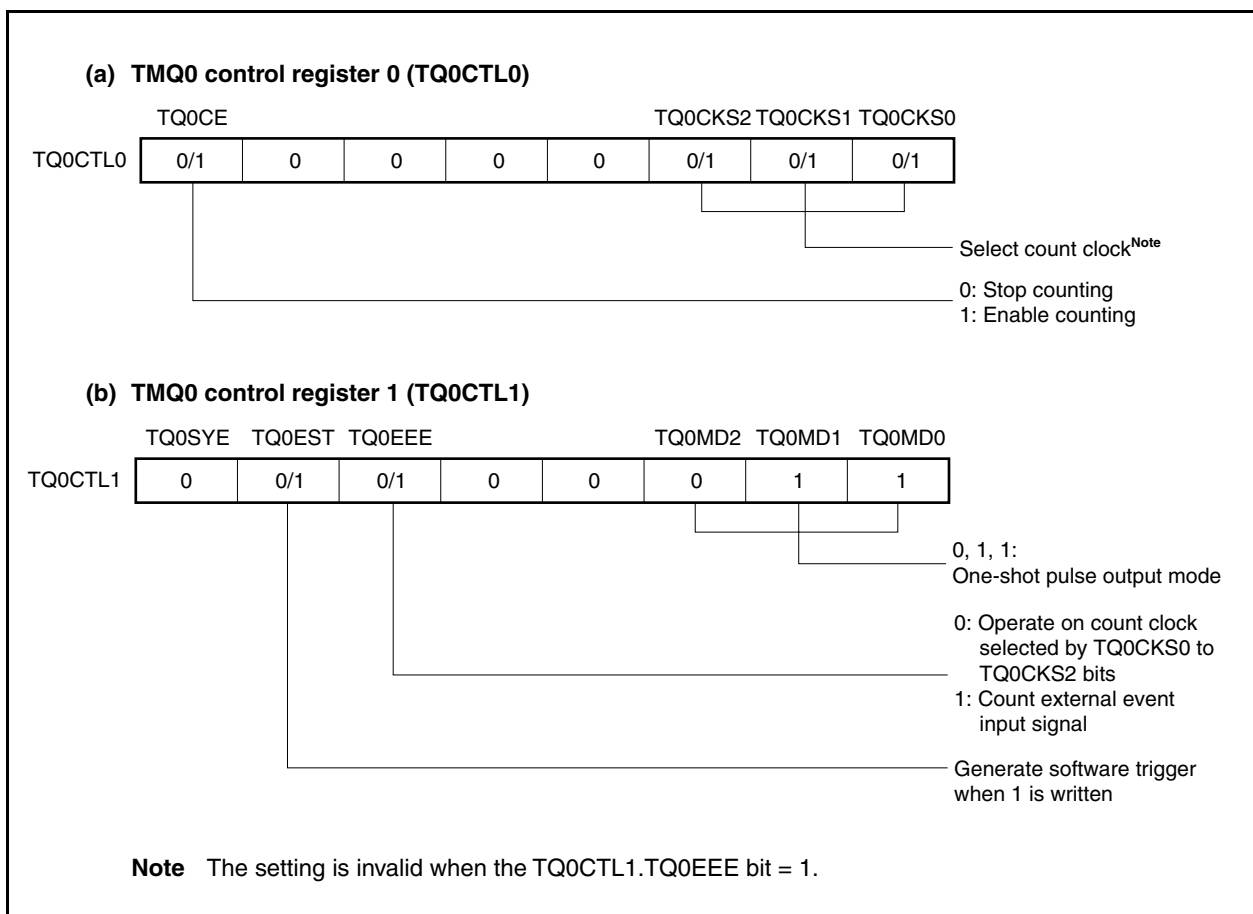


Figure 7-22. Register Setting in One-Shot Pulse Output Mode (2/3)

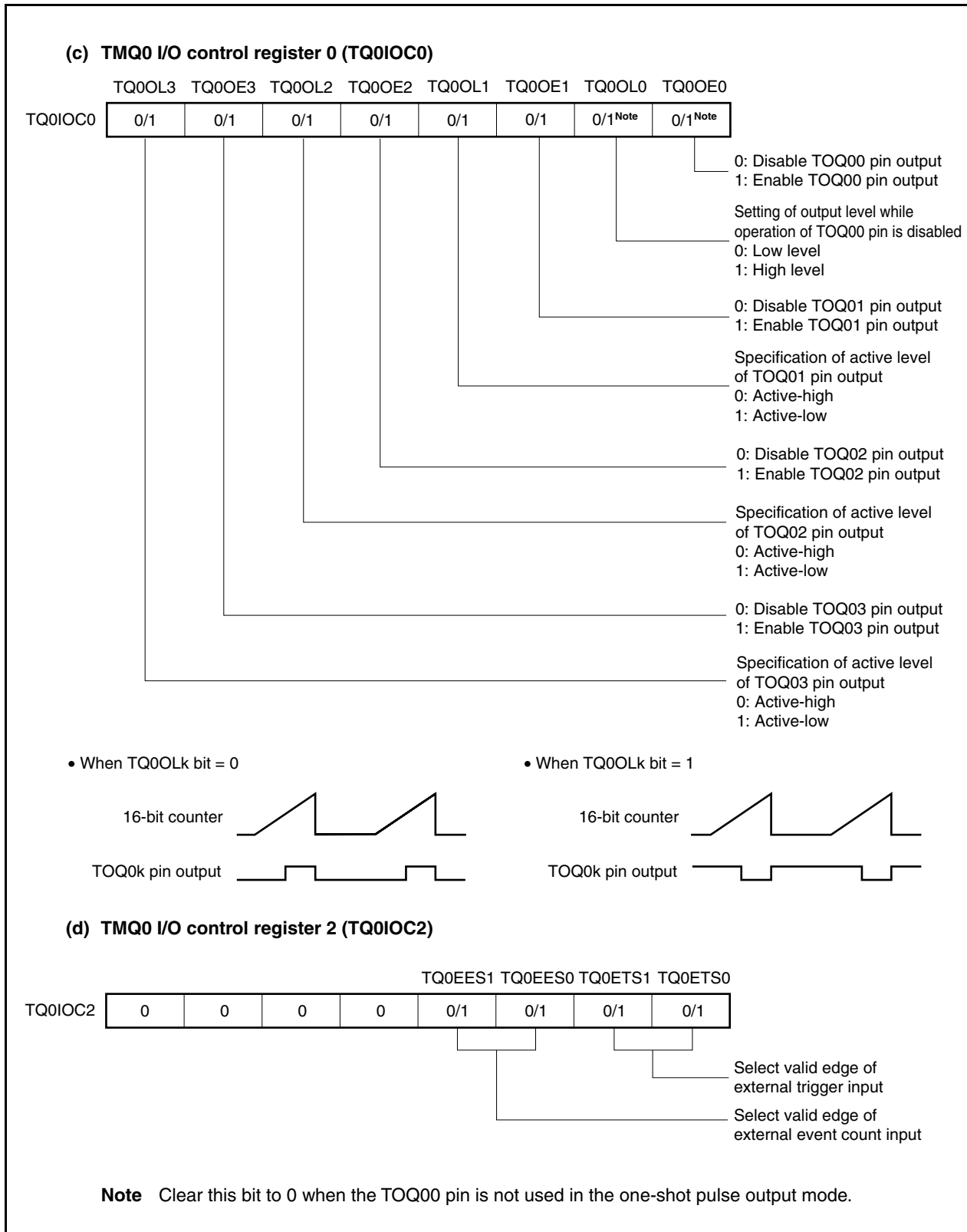


Figure 7-22. Register Setting in One-Shot Pulse Output Mode (3/3)

(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 = $(D_0 - D_k + 1) \times$ Count clock cycle

Output delay period = $(D_k) \times$ Count clock cycle

Caution One-shot pulses are not output even in the one-shot pulse output mode, if the set value in the TQ0CCRk register is greater than that value in the TQ0CCR0 register.

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

<R>

(1) Operation flow in one-shot pulse output mode

Figure 7-23. Software Processing Flow in One-Shot Pulse Output Mode (1/2)

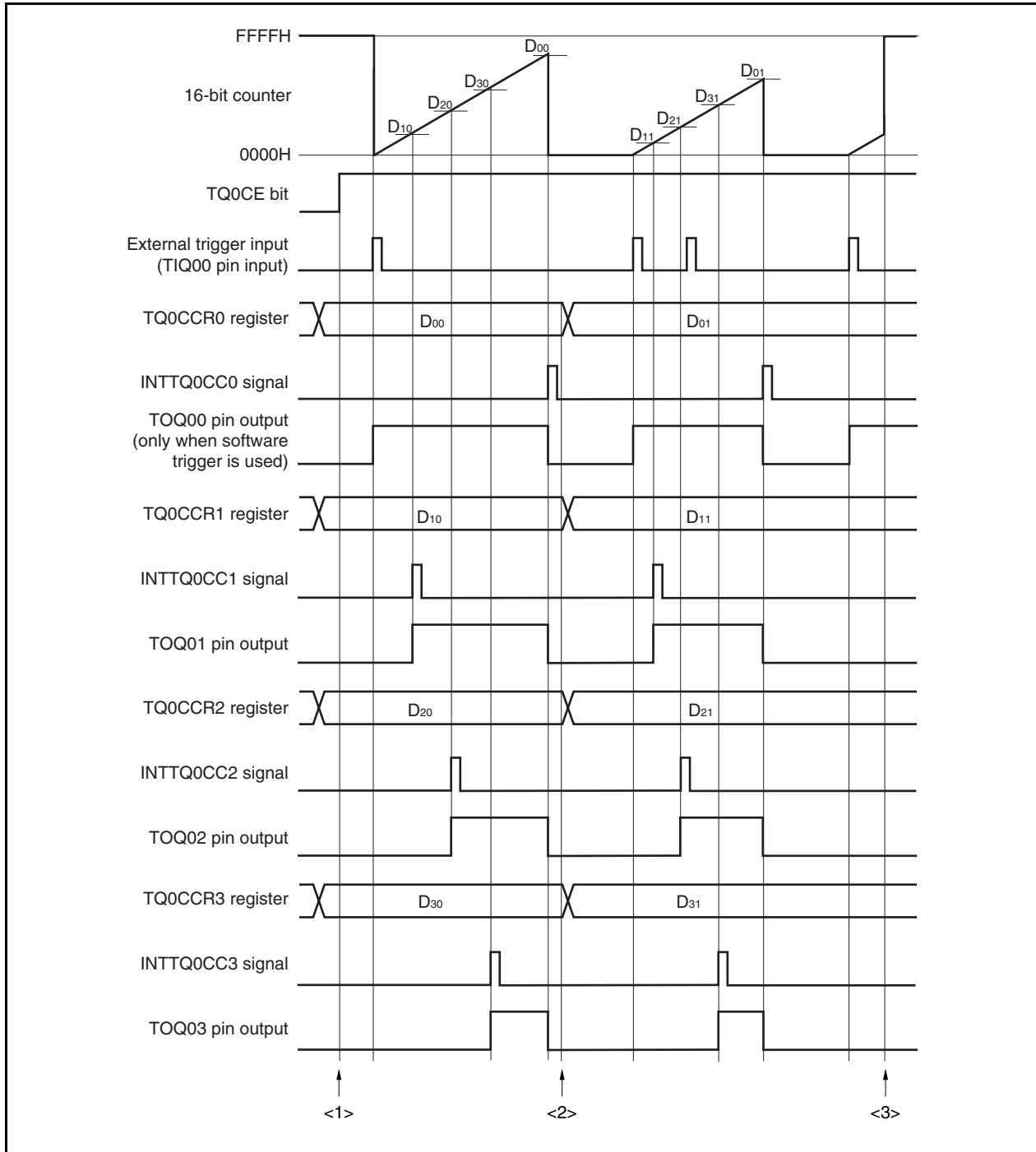
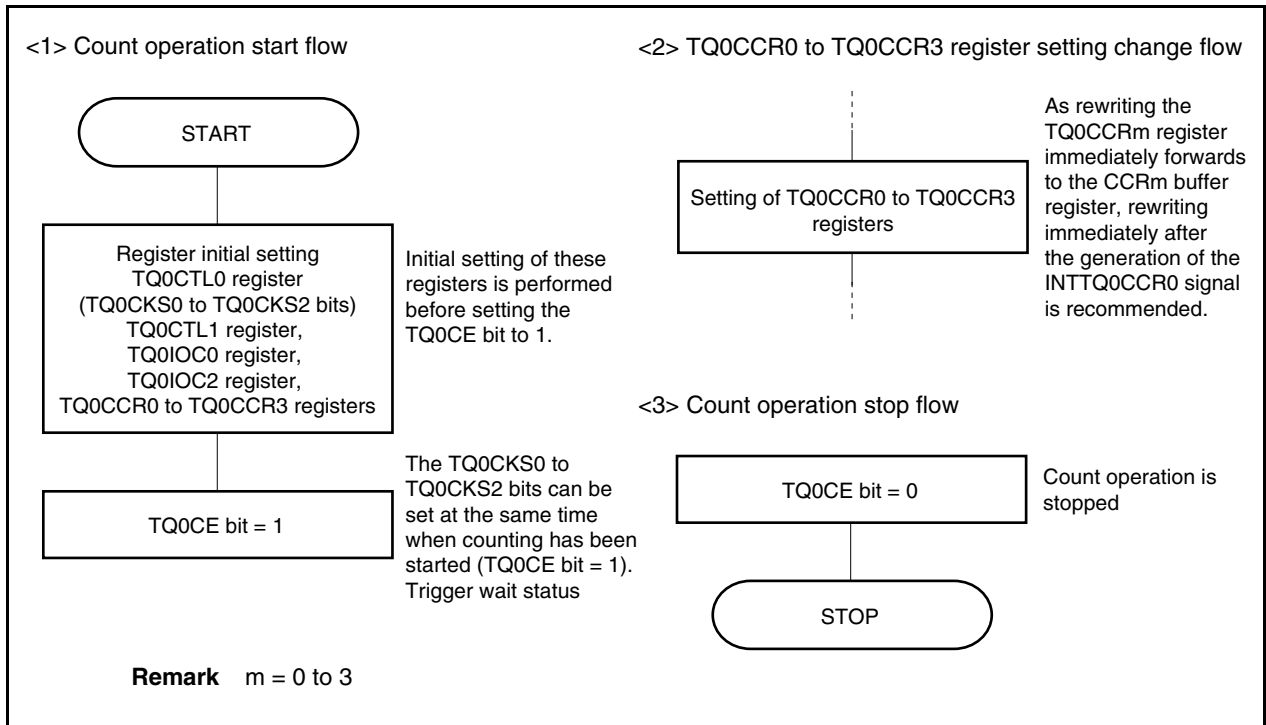


Figure 7-23. Software Processing Flow in One-Shot Pulse Output Mode (2/2)

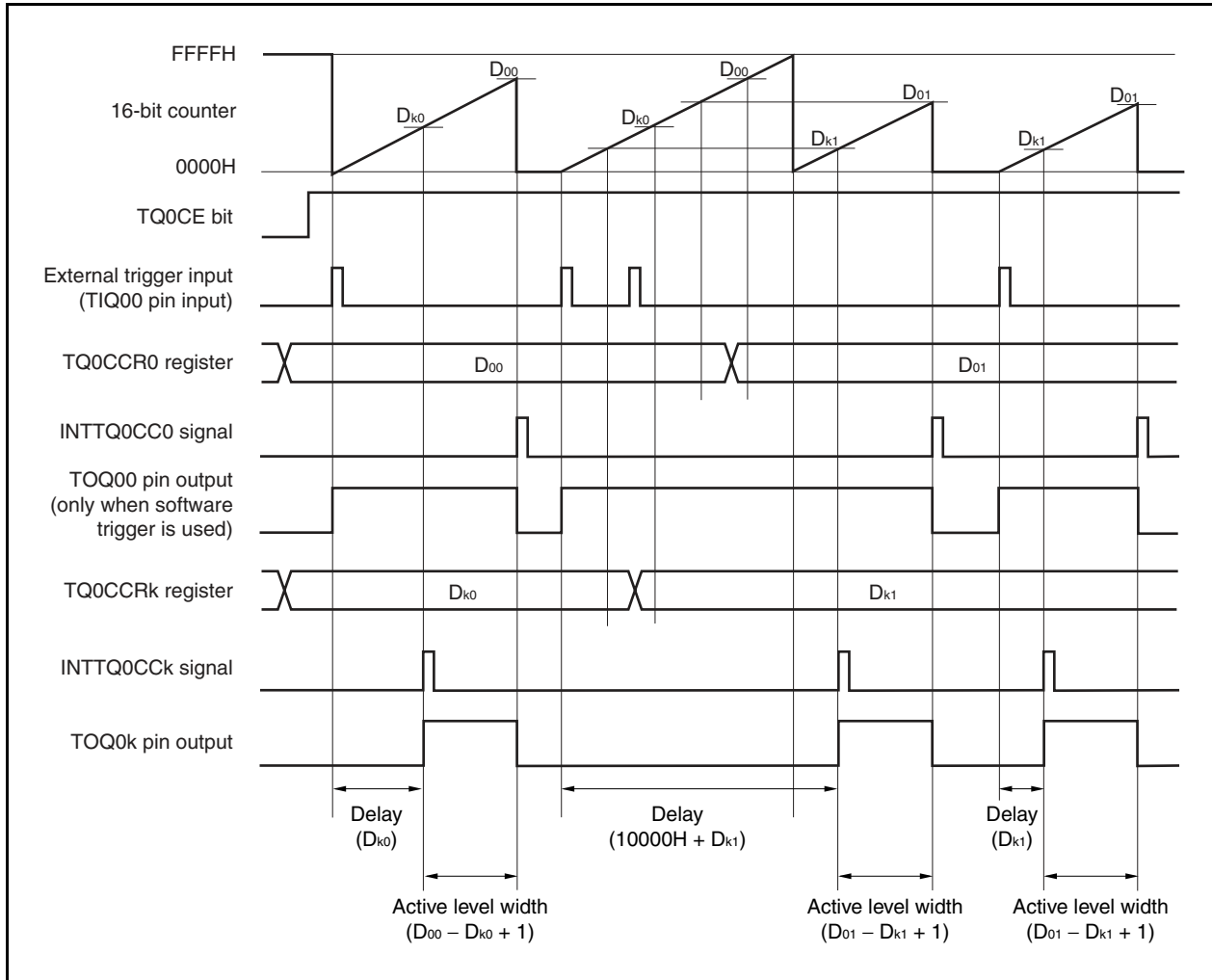


(2) Operation timing in one-shot pulse output mode

(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.



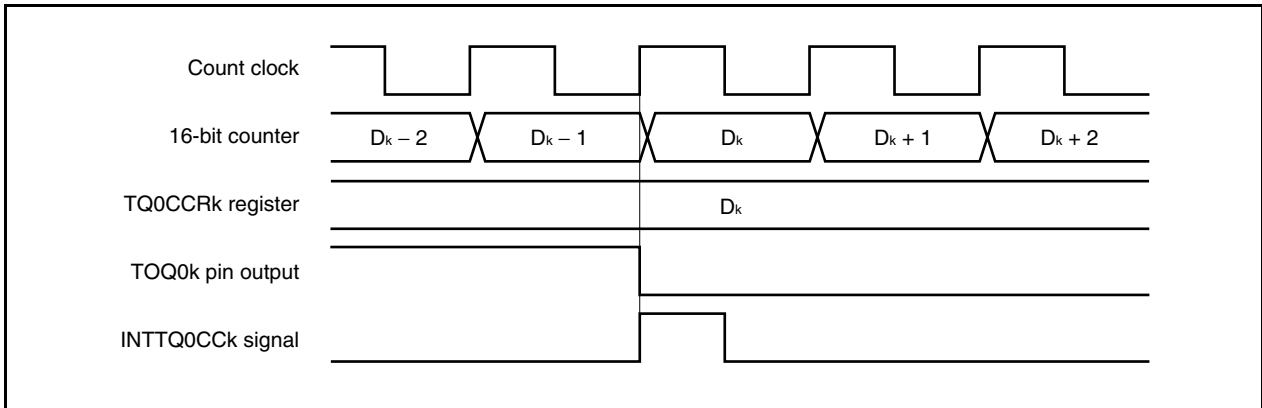
When the TQ0CCR0 register is rewritten from D₀₀ to D₀₁ and the TQ0CCRk register from D_{k0} to D_{k1} where D₀₀ > D₀₁ and D_{k0} > D_{k1}, if the TQ0CCRk register is rewritten when the count value of the 16-bit counter is greater than D_{k1} and less than D_{k0} and if the TQ0CCR0 register is rewritten when the count value is greater than D₀₁ and less than D₀₀, 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 D₀₁, 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

7.5.5 PWM output mode (TQ0MD2 to TQ0MD0 bits = 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 7-24. Configuration in PWM Output Mode

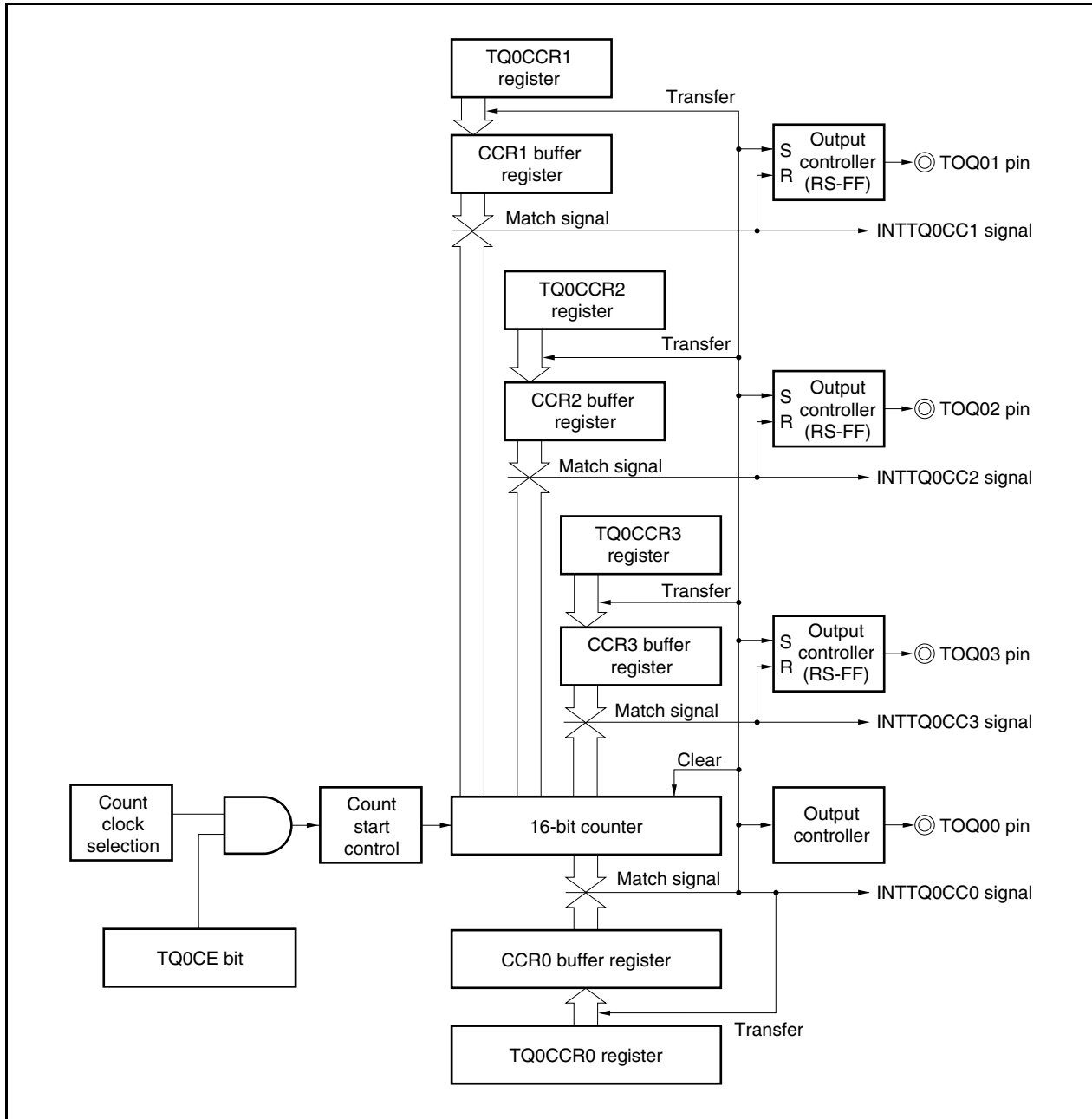
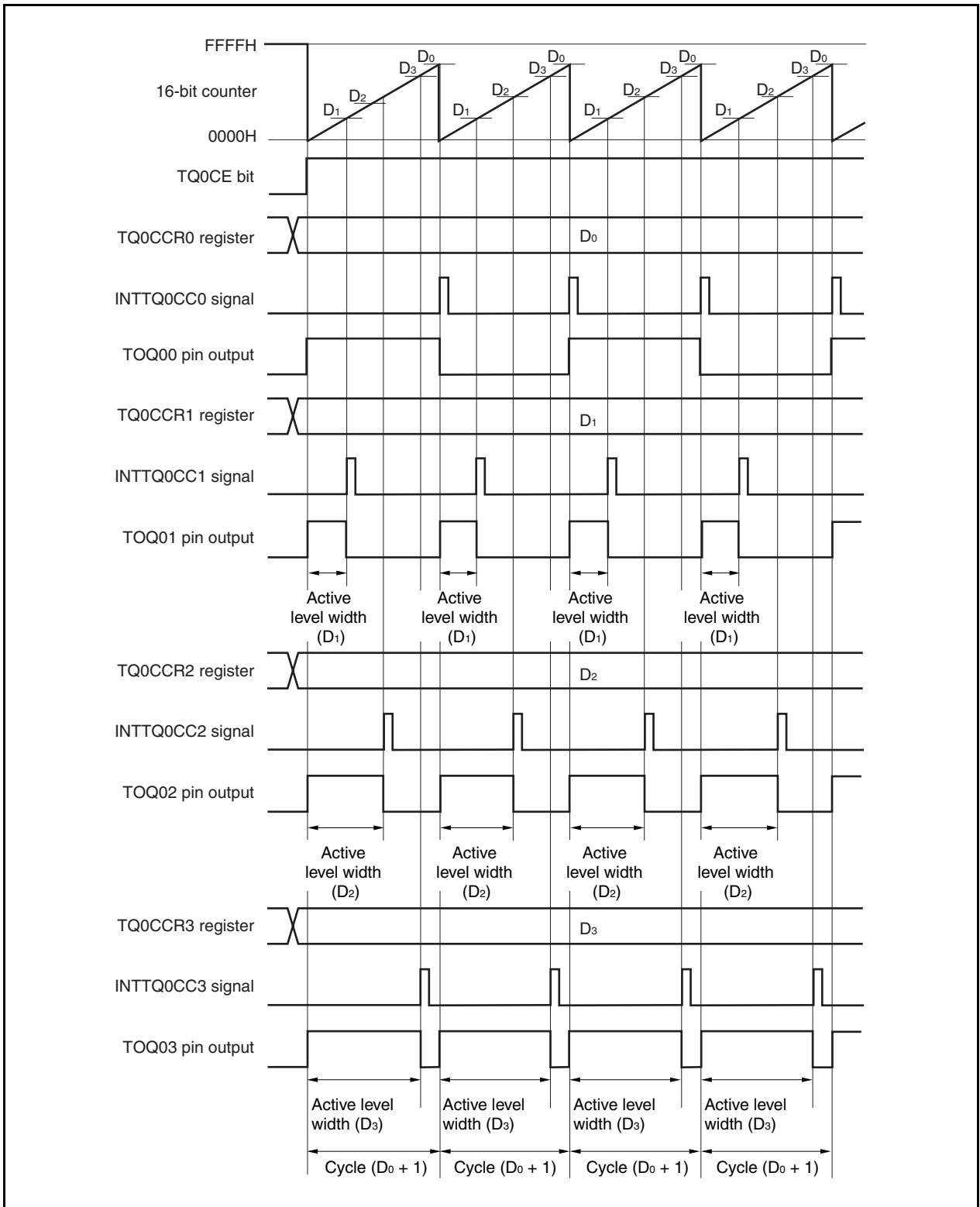


Figure 7-25. Basic Timing in PWM Output Mode



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.

$$\text{Active level width} = (\text{Set value of TQ0CCRk register}) \times \text{Count clock cycle}$$

$$\text{Cycle} = (\text{Set value of TQ0CCR0 register} + 1) \times \text{Count clock cycle}$$

$$\text{Duty factor} = (\text{Set value of TQ0CCRk register}) / (\text{Set value of TQ0CCR0 register} + 1)$$

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 7-26. Setting of Registers in PWM Output Mode (1/3)

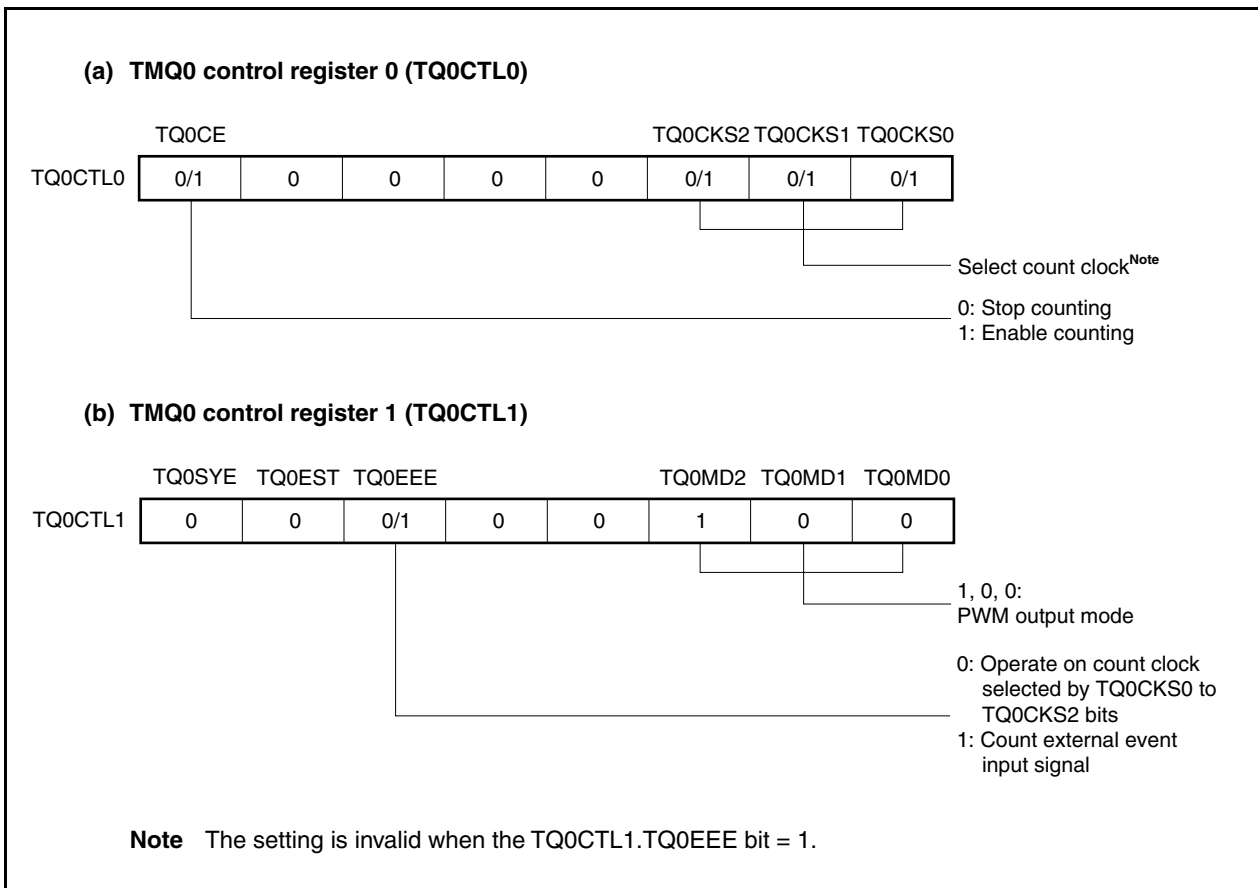


Figure 7-26. Setting of Registers in PWM Output Mode (2/3)

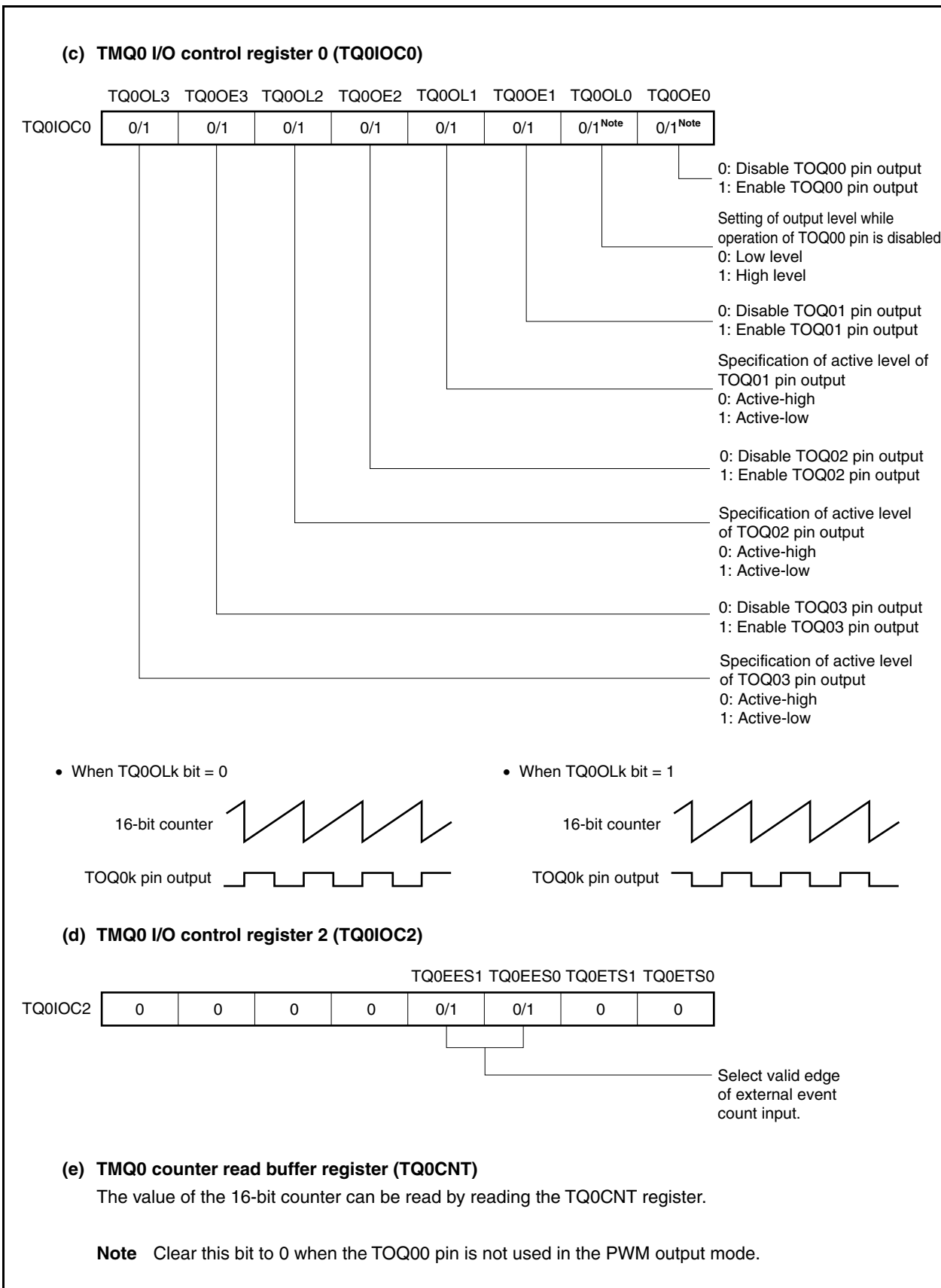


Figure 7-26. Register Setting in PWM Output Mode (3/3)

(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 TQ0CCk register, the cycle and active level of the PWM waveform are as follows.

$$\text{Cycle} = (D_0 + 1) \times \text{Count clock cycle}$$

$$\text{Active level width} = D_k \times \text{Count clock cycle}$$

- Remarks**
1. TMQ0 I/O control register 1 (TQ0IOC1) and TMQ0 option register 0 (TQ0OPT0) are not used in the PWM output mode.
 2. Updating the TMQ0 capture/compare register 2 (TQ0CCR2) and TMQ0 capture/compare register 3 (TQ0CCR3) is validated by writing the TMQ0 capture/compare register 1 (TQ0CCR1).

(1) Operation flow in PWM output mode

Figure 7-27. Software Processing Flow in PWM Output Mode (1/2)

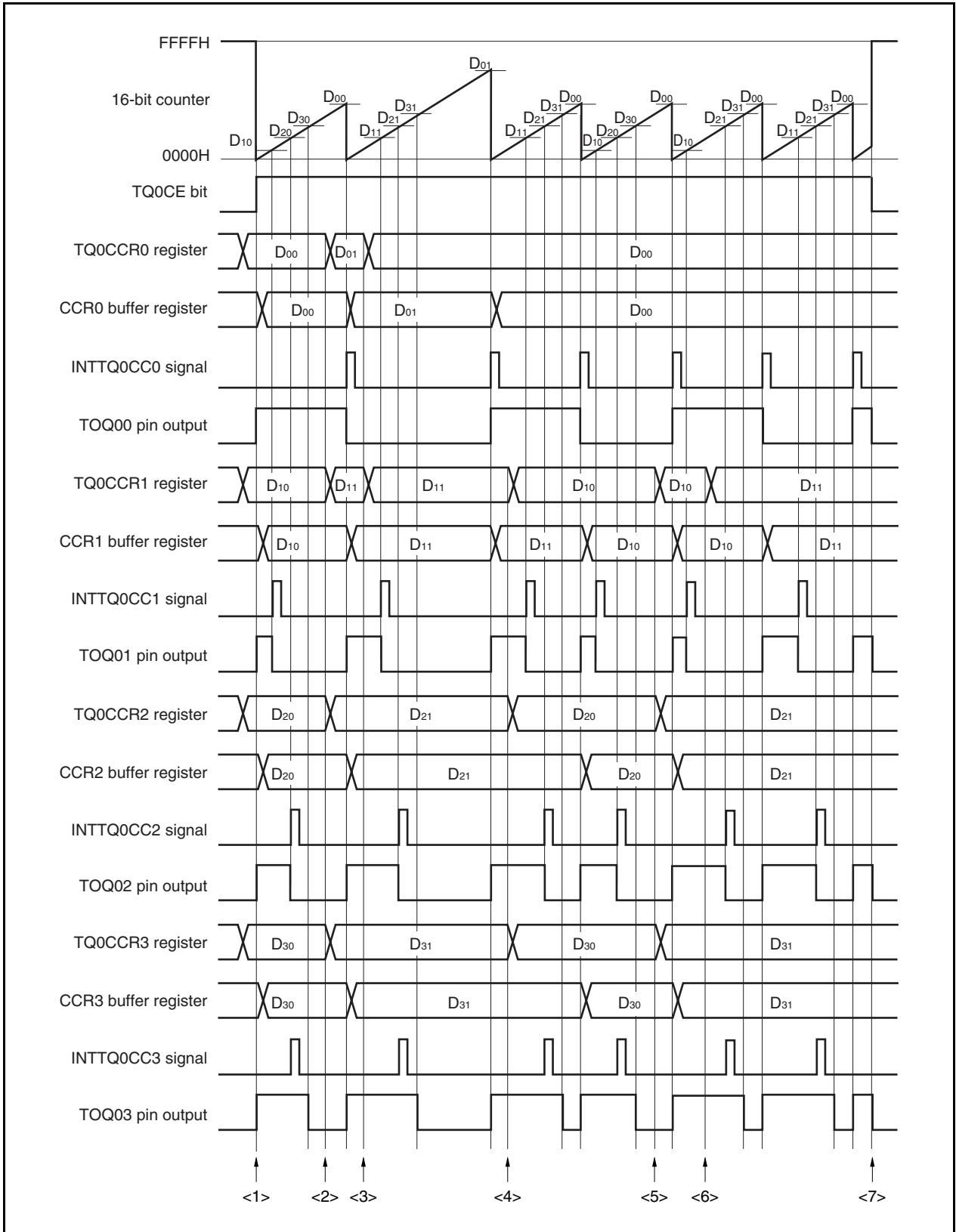
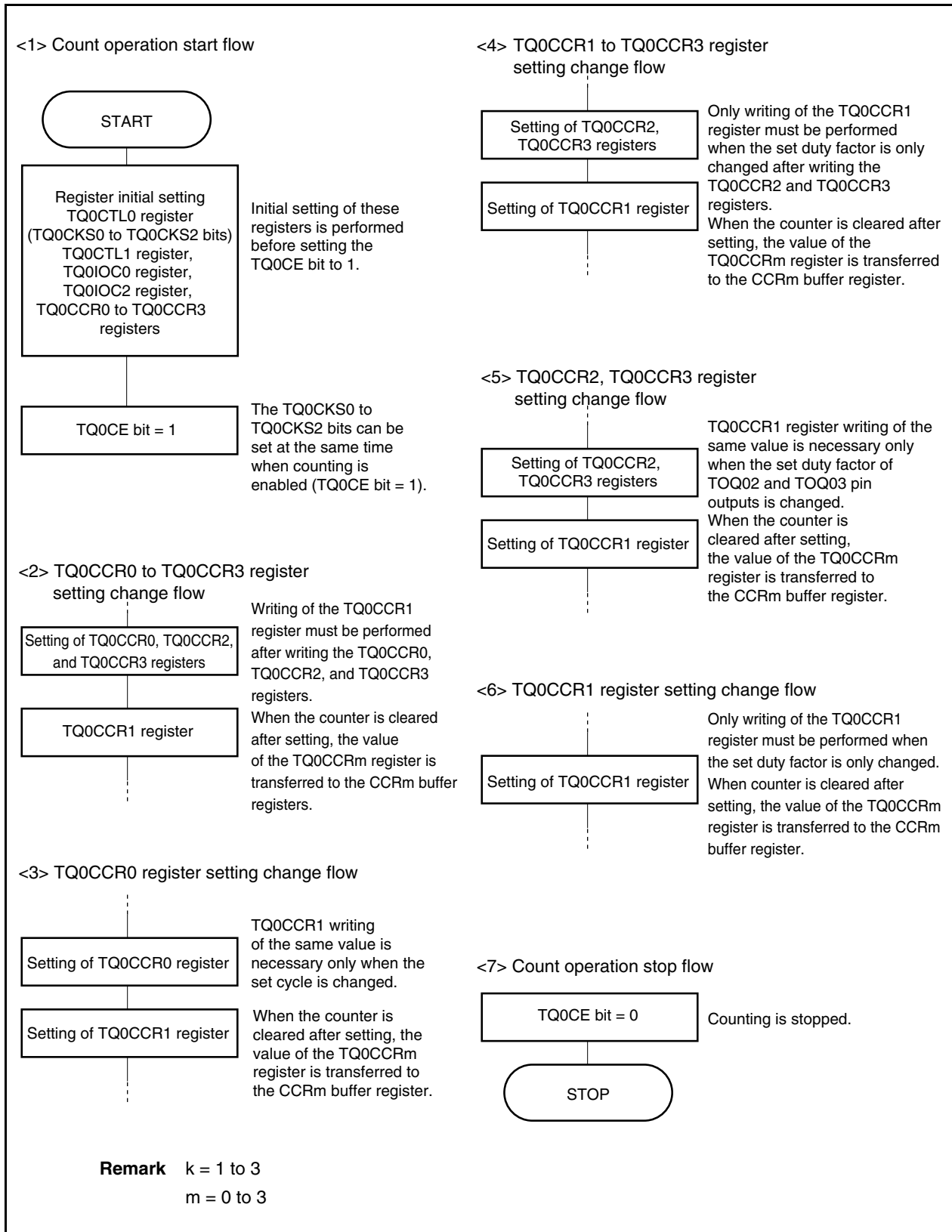


Figure 7-27. Software Processing Flow in PWM Output Mode (2/2)

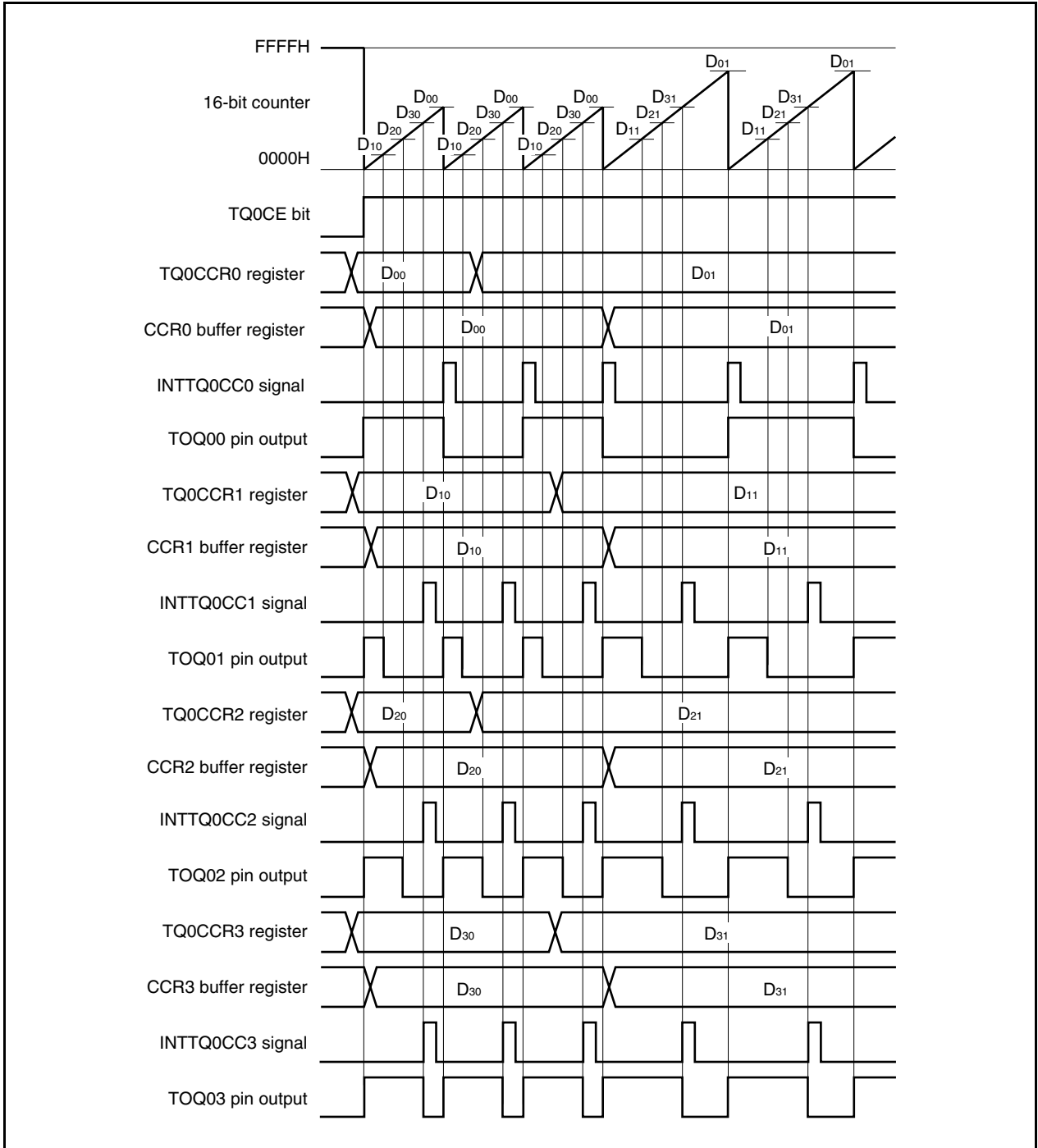


(2) PWM output mode operation timing

(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 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 active level width (duty factor) of PWM wave, first set the active level to the TQ0CCR2 and TQ0CCR3 registers, and then set an active level 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.

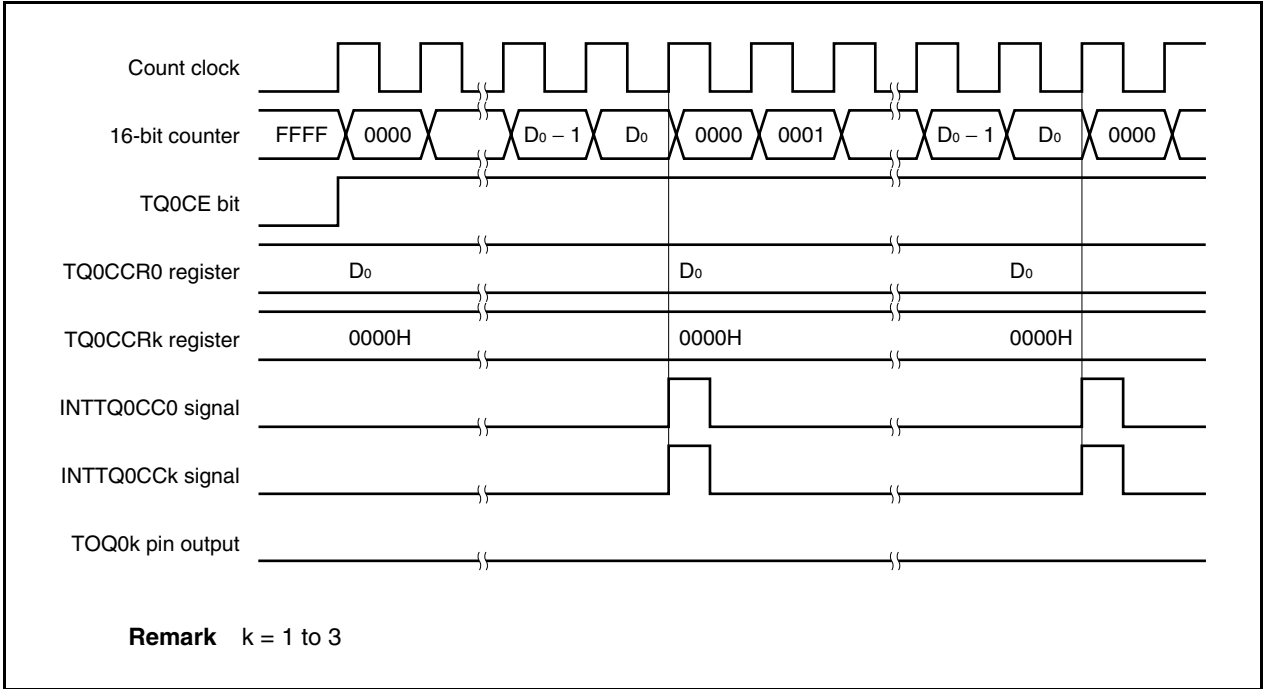
To change only the cycle of the PWM waveform, first set a cycle to the TQ0CCR0 register, and then write the same value to the TQ0CCR1 register.

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.

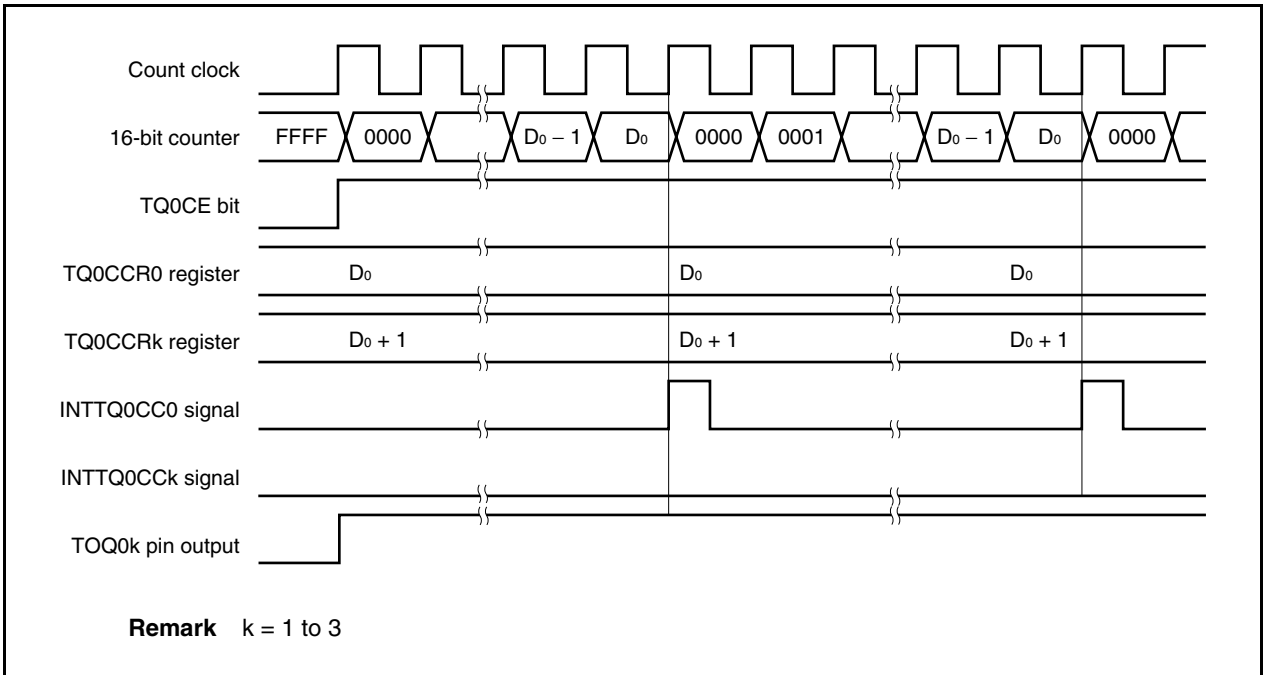
Remark m = 0 to 3

(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.

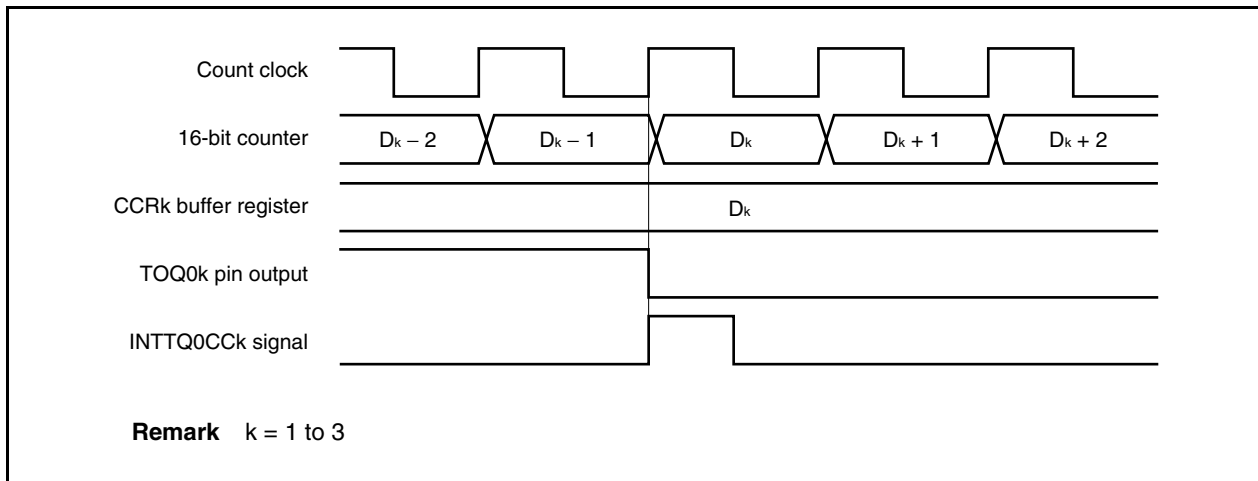


To output a 100% waveform, set a value of (set value of TQ0CCR0 register + 1) to the TQ0CCRk register. If the set value of the TQ0CCR0 register is FFFFH, 100% output cannot be produced.



(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.



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.

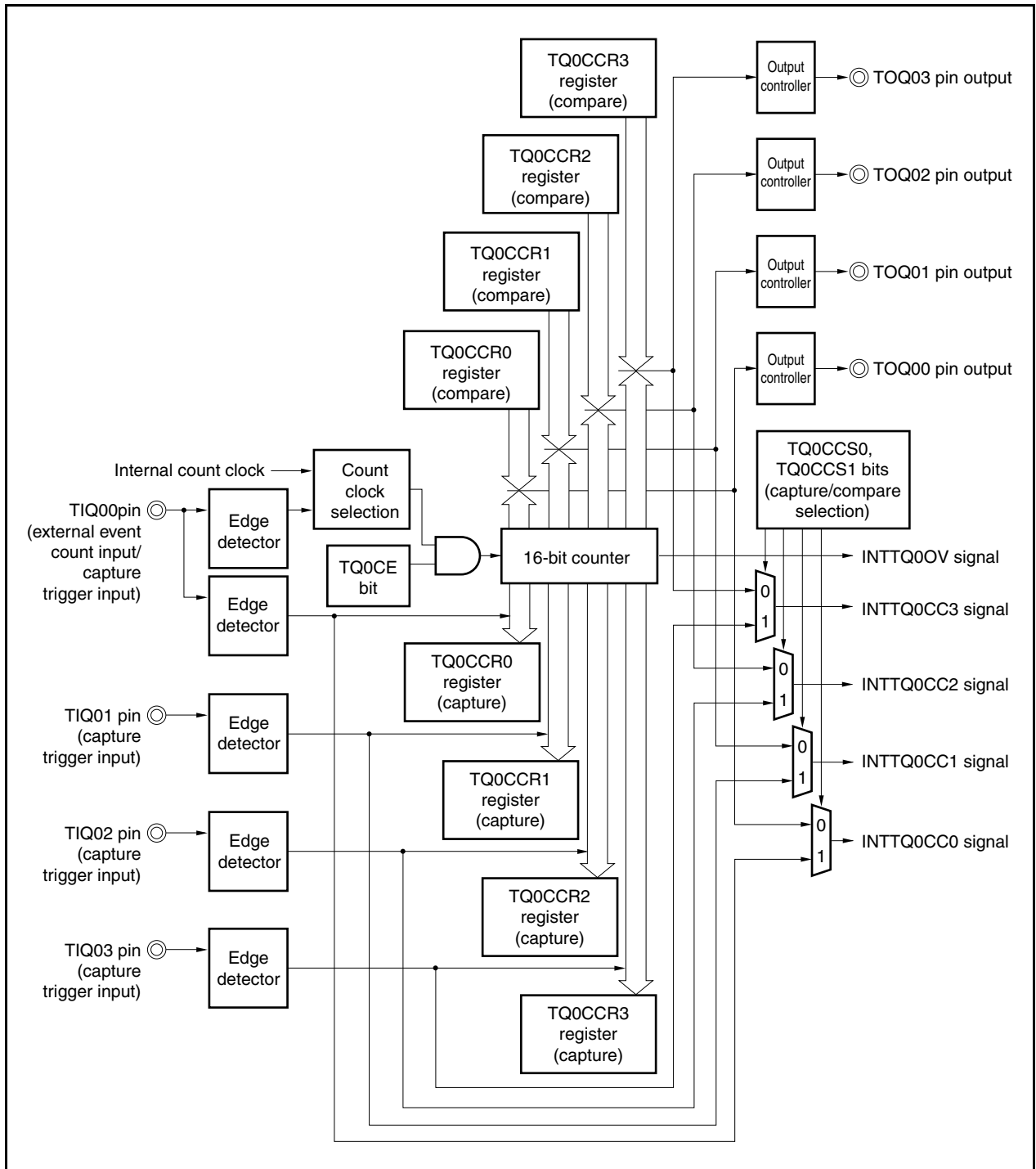
In the PWM output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the output signal of the TOQ0k pin.

7.5.6 Free-running timer mode (TQ0MD2 to TQ0MD0 bits = 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 7-28. Configuration in Free-Running Timer Mode

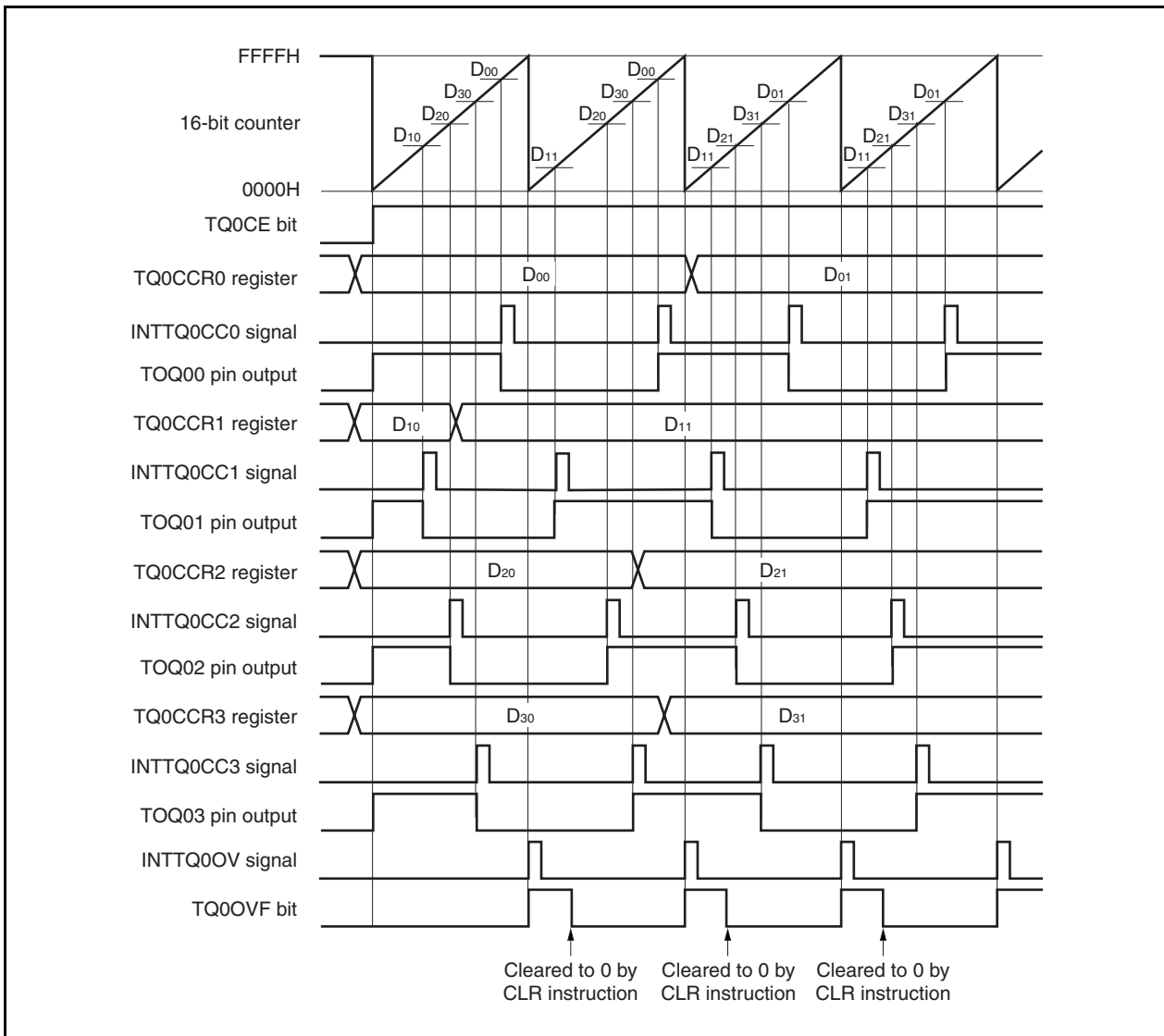


When the TQ0CE bit is set to 1, 16-bit timer/event counter Q starts counting, and the output signals of the TOQ00 to TOQ03 pins are inverted. When the count value of the 16-bit counter later matches the set value of the TQ0CCRm register, a compare match interrupt request signal (INTTQ0CCm) is generated, 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 7-29. 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 7-30. Basic Timing in Free-Running Timer Mode (Capture Function)

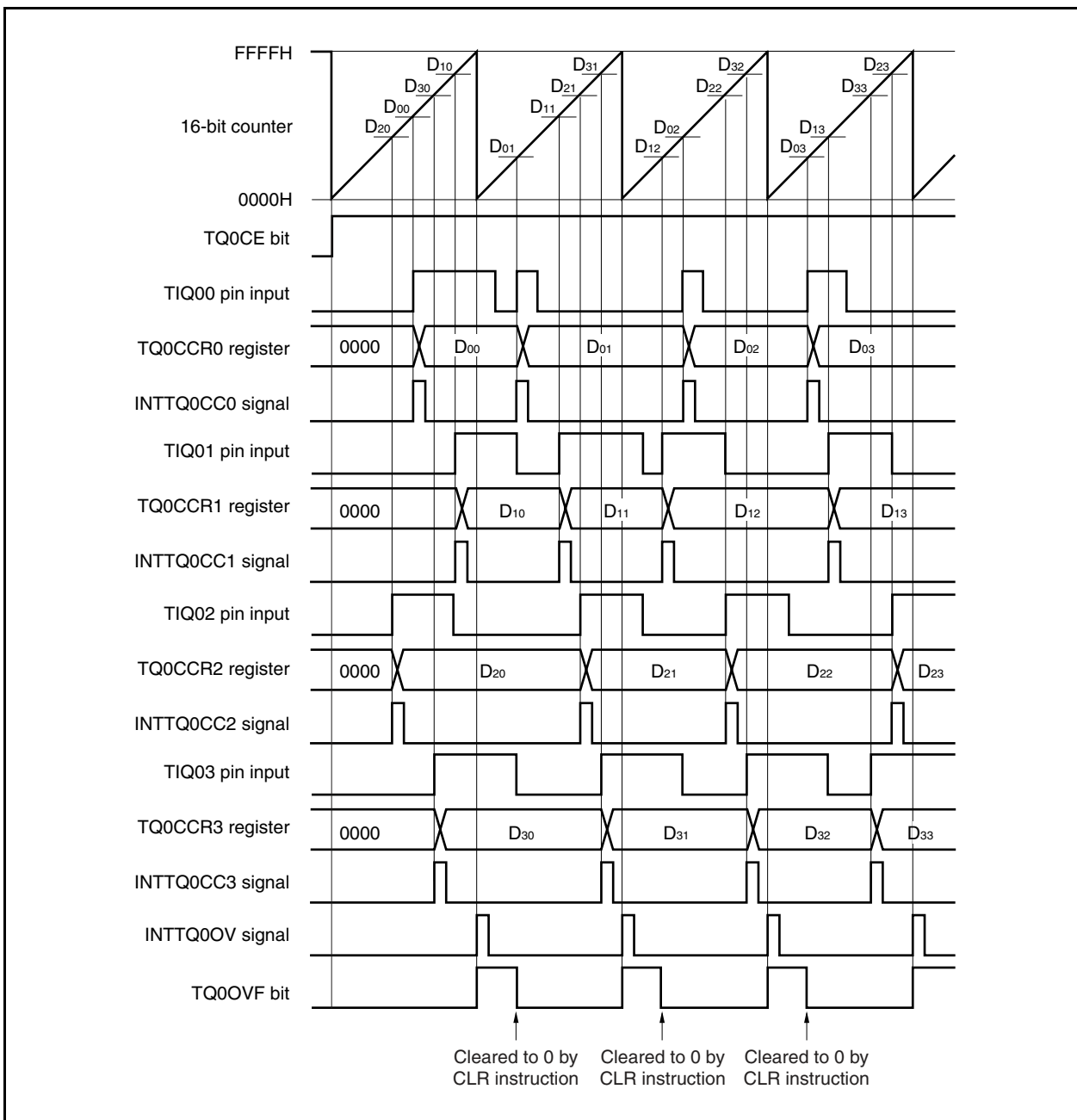


Figure 7-31. Register Setting in Free-Running Timer Mode (1/3)

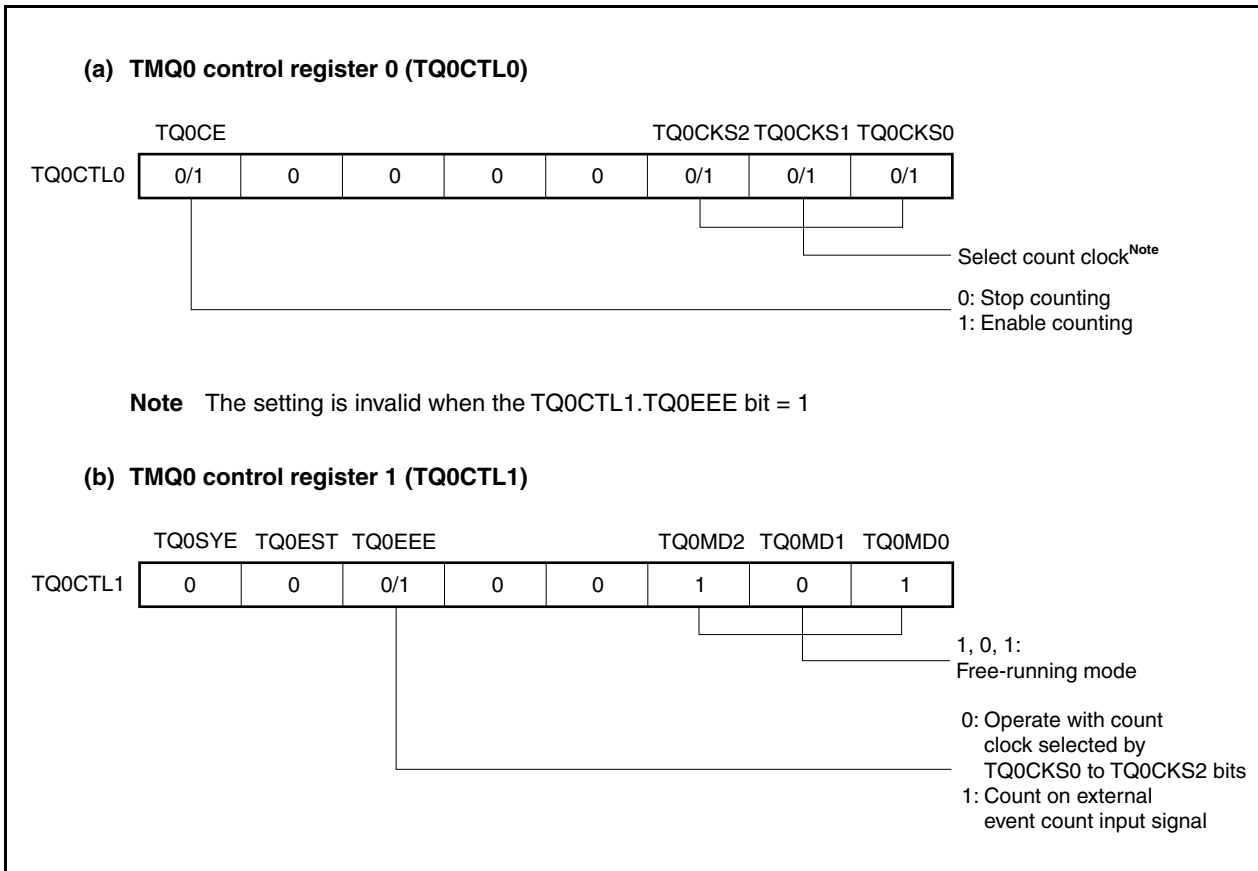


Figure 7-31. Register Setting in Free-Running Timer Mode (2/3)

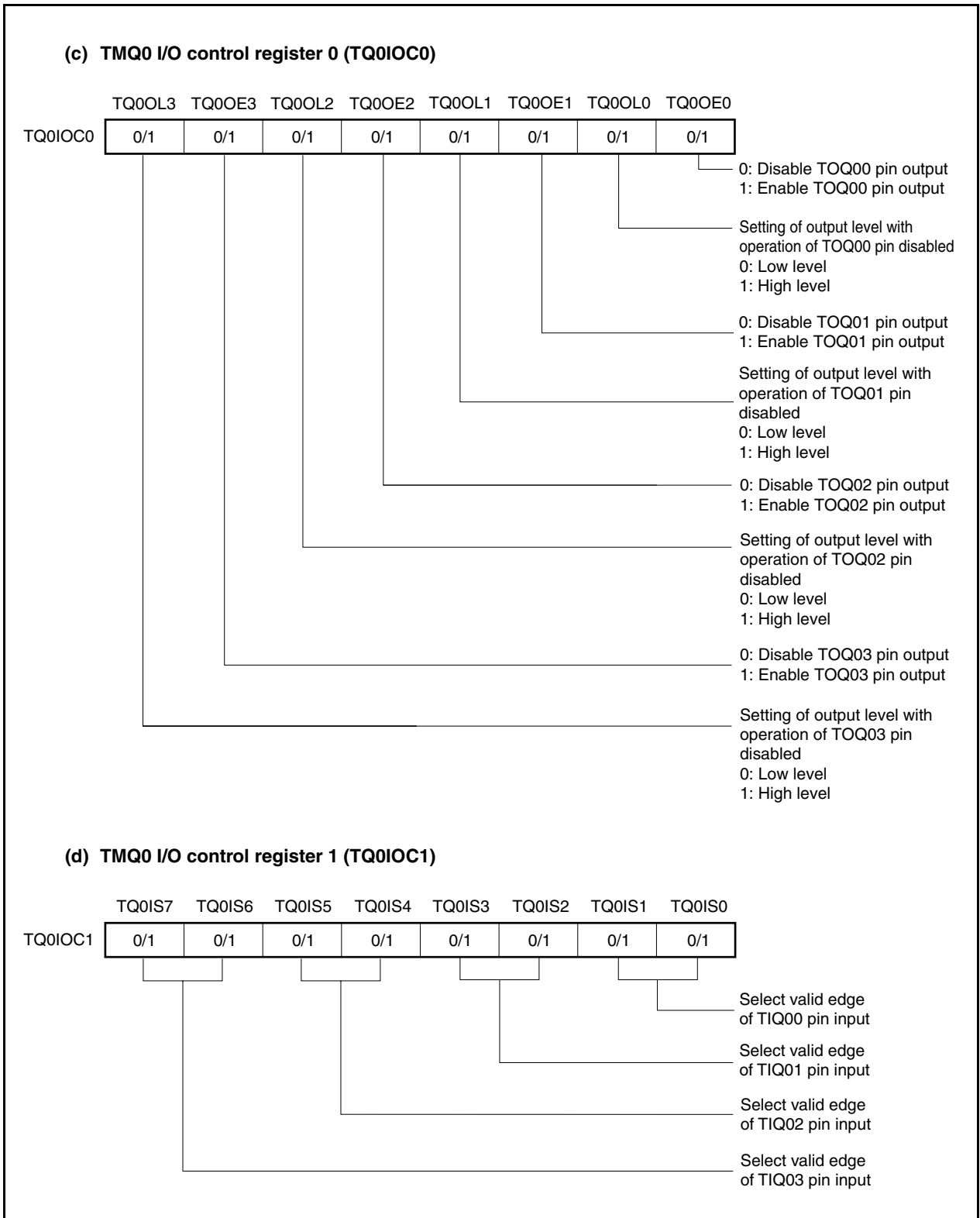
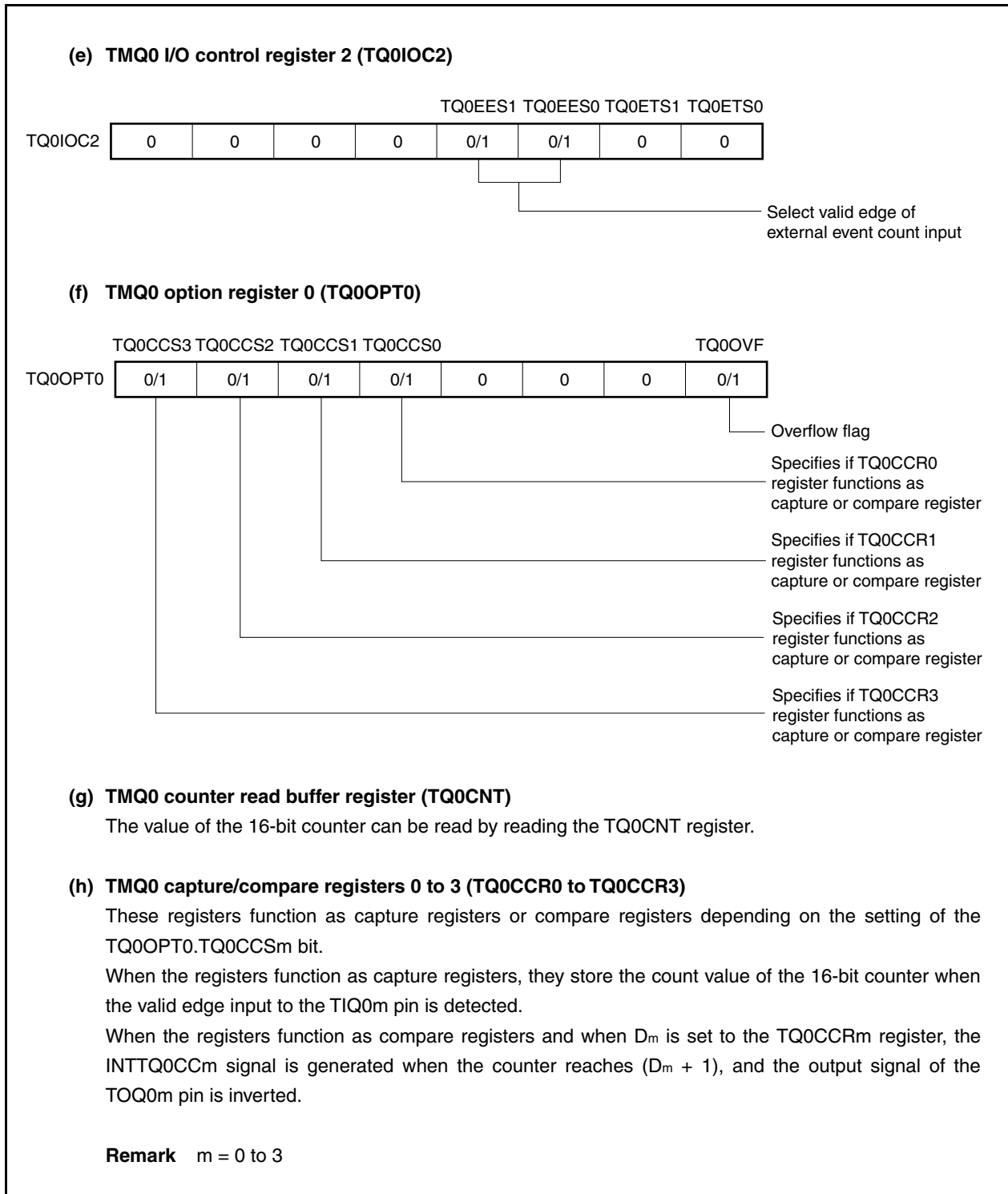


Figure 7-31. Register Setting in Free-Running Timer Mode (3/3)



(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)

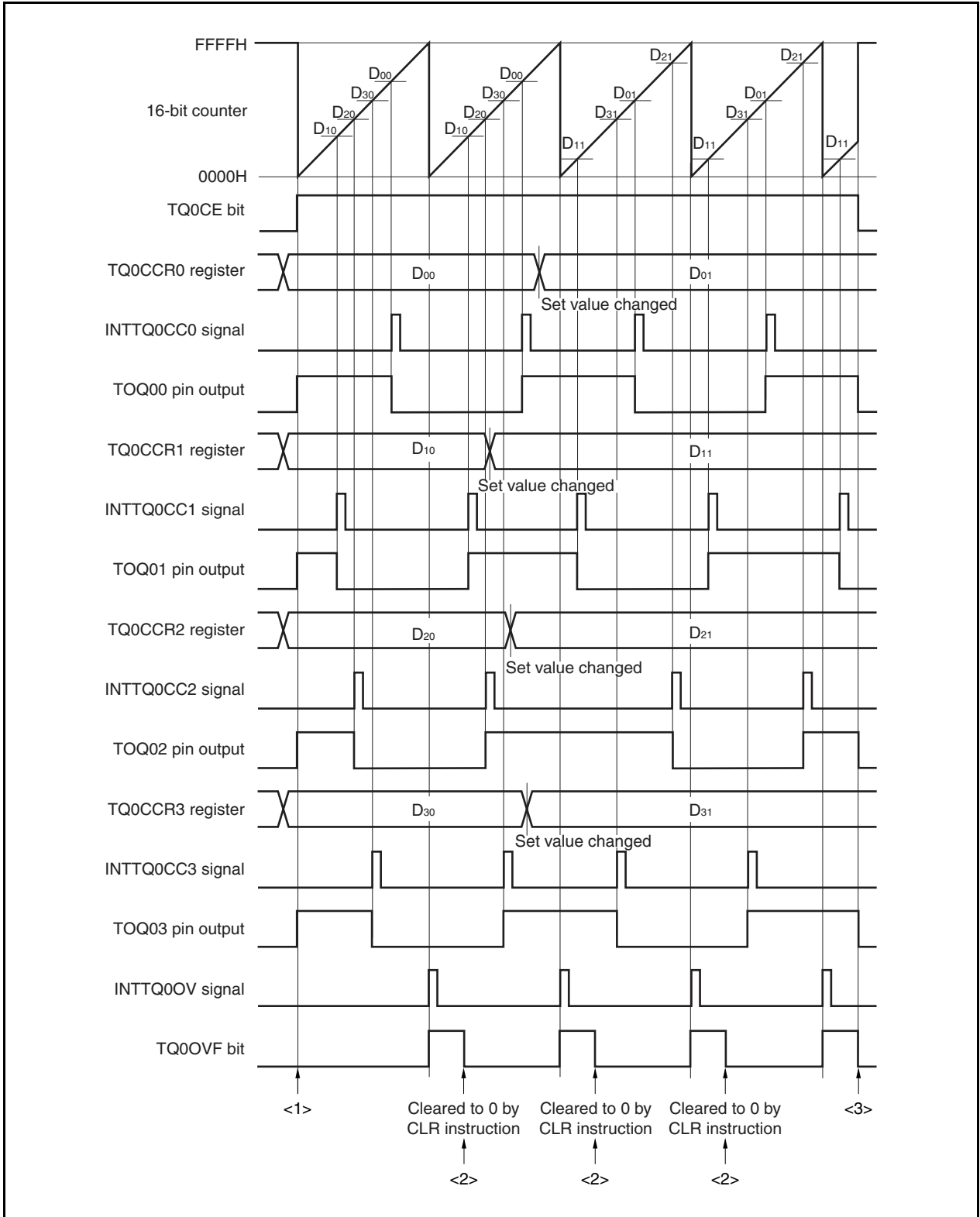
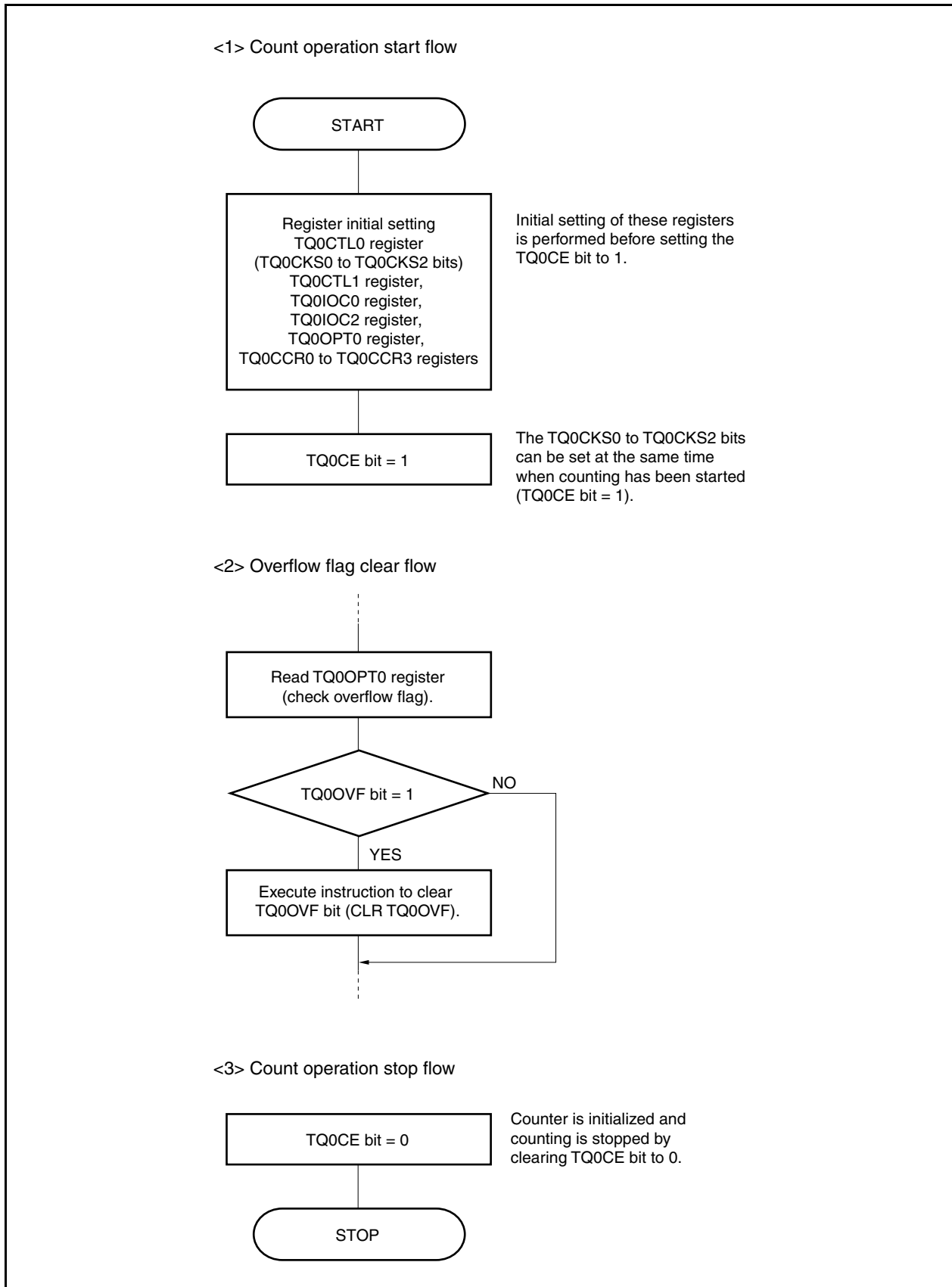


Figure 7-32. Software Processing Flow in Free-Running Timer Mode (Compare Function) (2/2)



(b) When using capture/compare register as capture register

Figure 7-33. Software Processing Flow in Free-Running Timer Mode (Capture Function) (1/2)

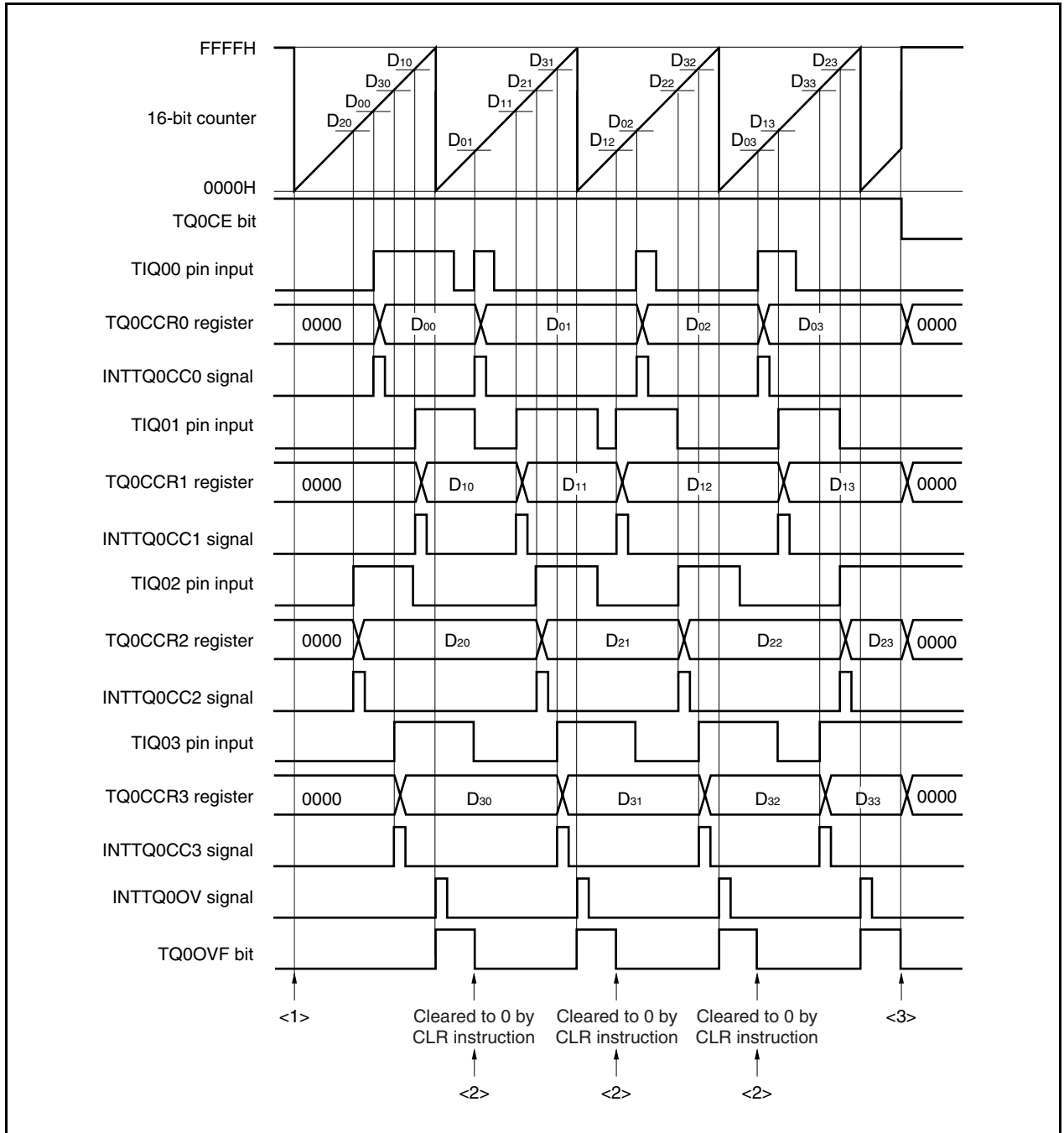
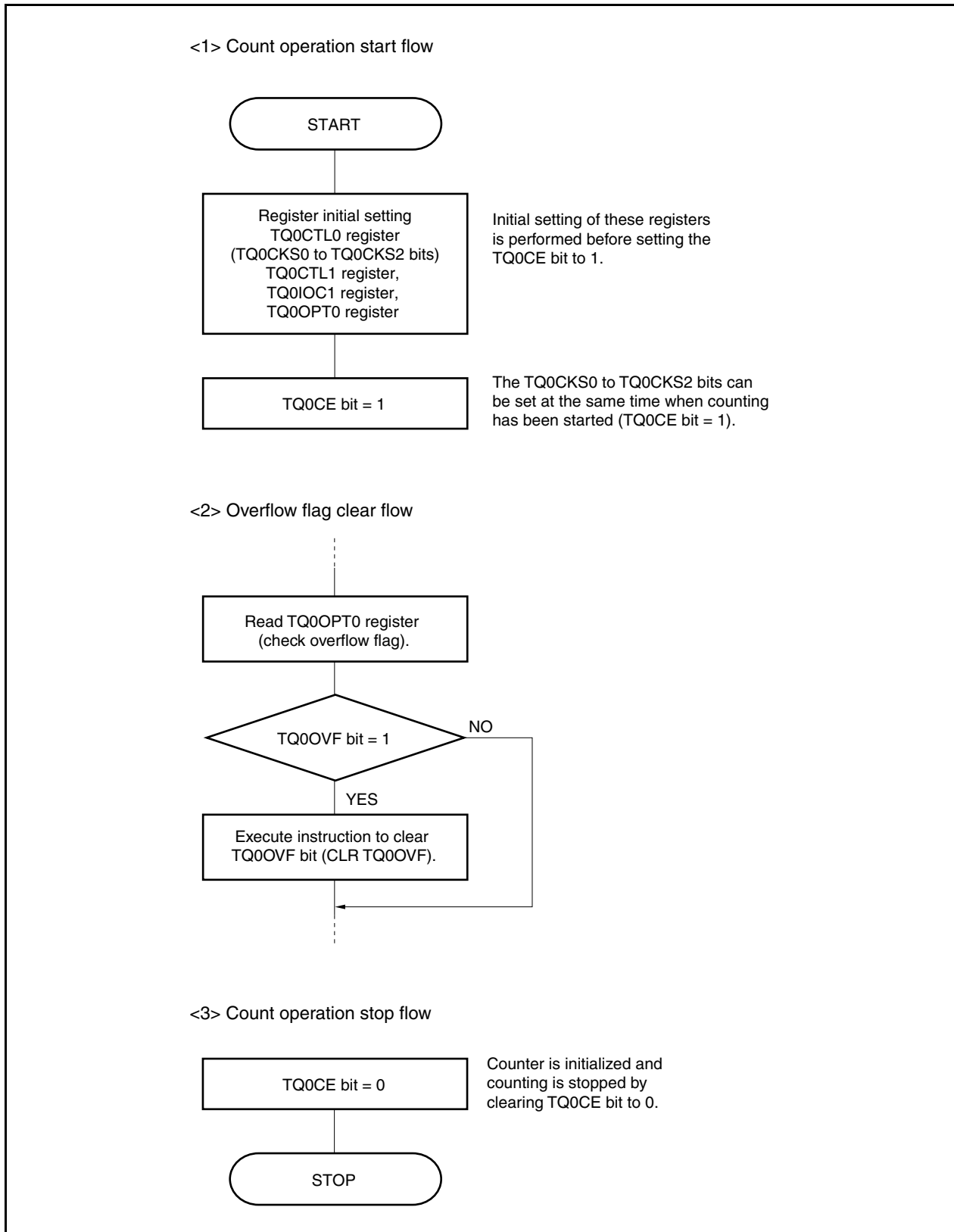


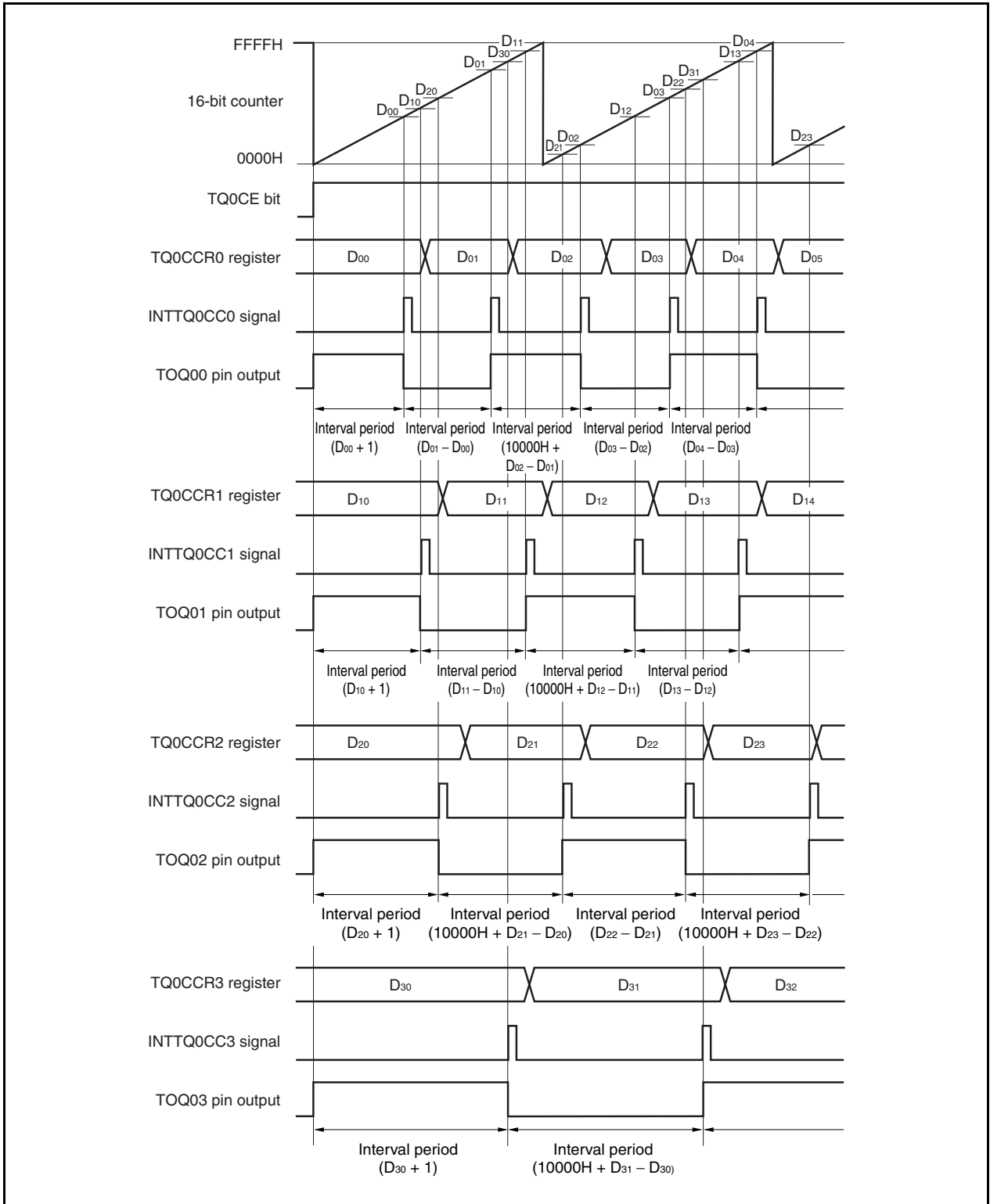
Figure 7-33. Software Processing Flow in Free-Running Timer Mode (Capture Function) (2/2)



(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, two intervals can be set with one channel.

To perform the interval operation, the value of the corresponding TQ0CCR_m register must be re-set in the interrupt servicing that is executed when the INTTQ0CC_m signal is detected.

The set value for re-setting the TQ0CCR_m register can be calculated by the following expression, where “D_m” is the interval period.

Compare register default value: $D_m - 1$

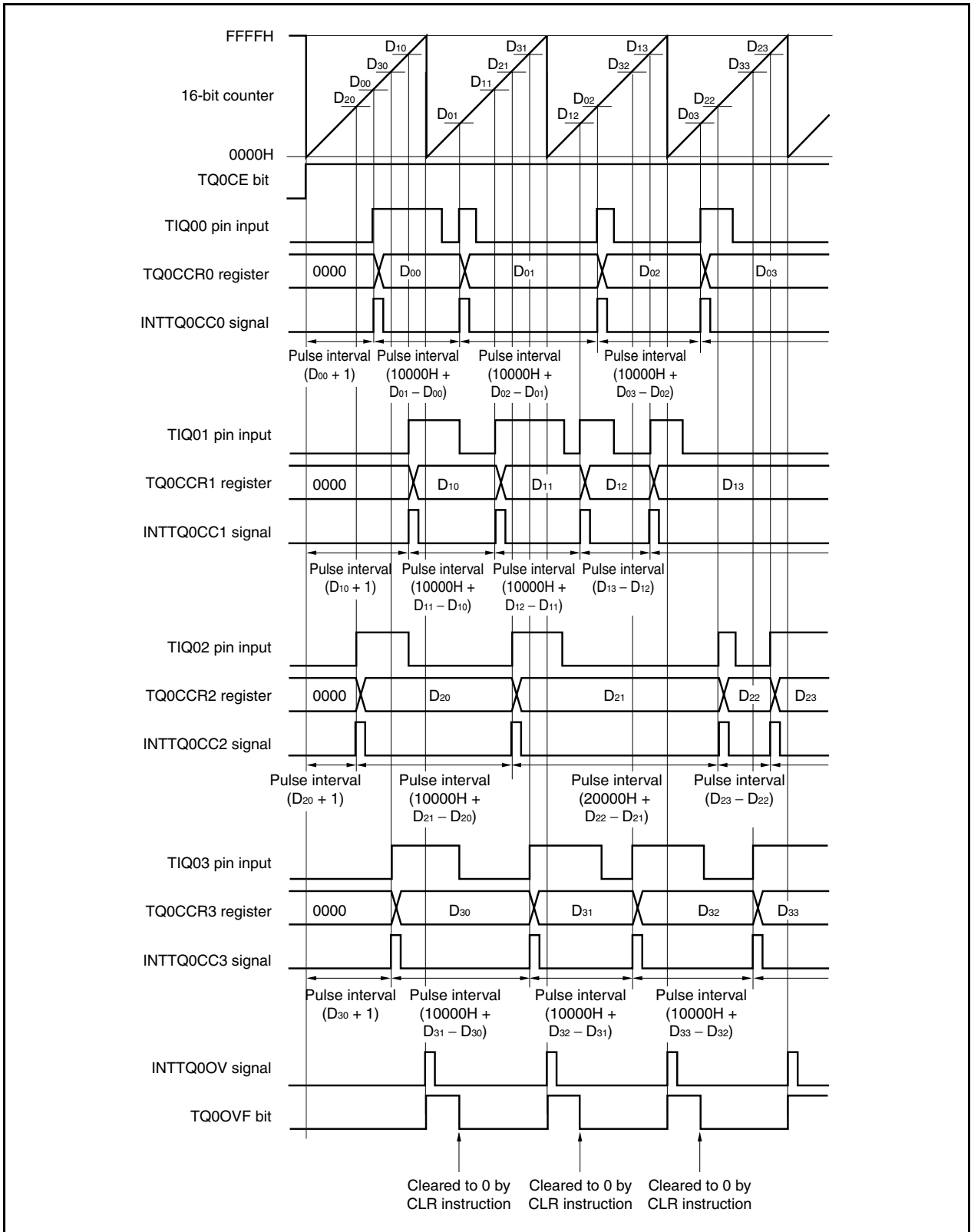
Value set to compare register second and subsequent time: Previous set value + D_m

(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.



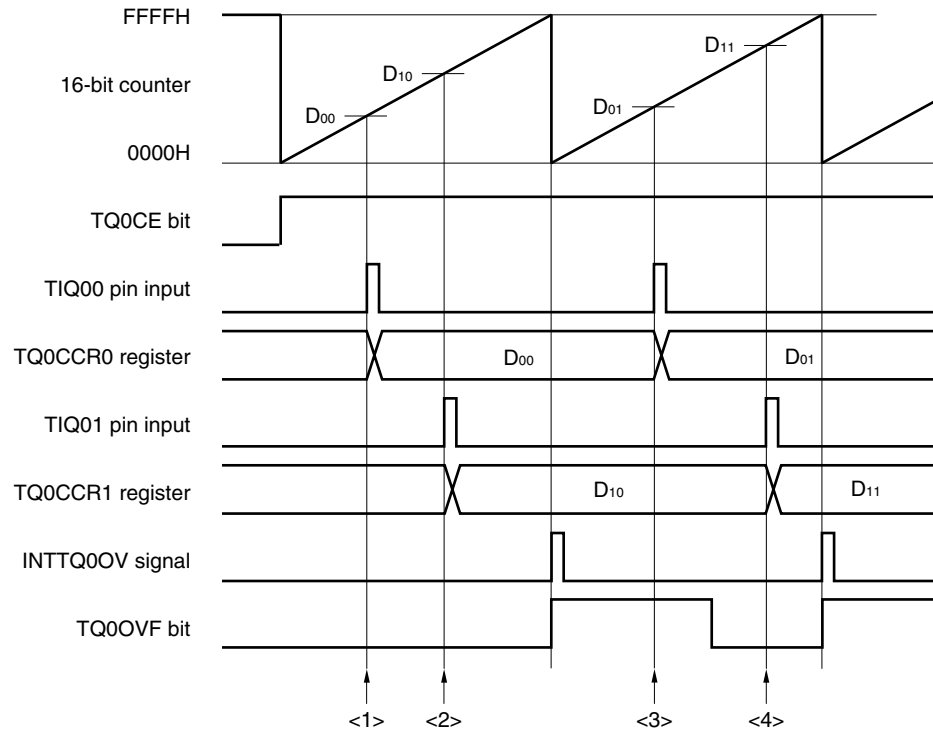
When executing pulse width measurement in the free-running timer mode, four pulse widths can be measured with one channel.

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 or more 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 or more capture registers are used

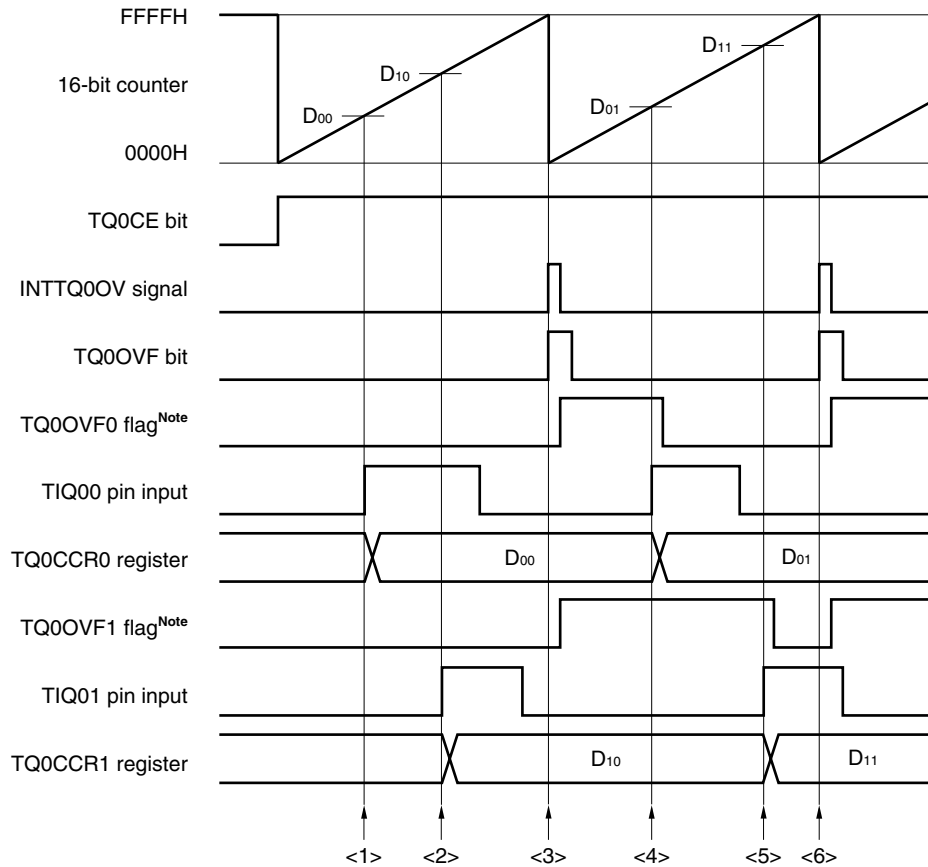
The following problem may occur when two pulse widths are measured in the free-running timer mode.

- <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.

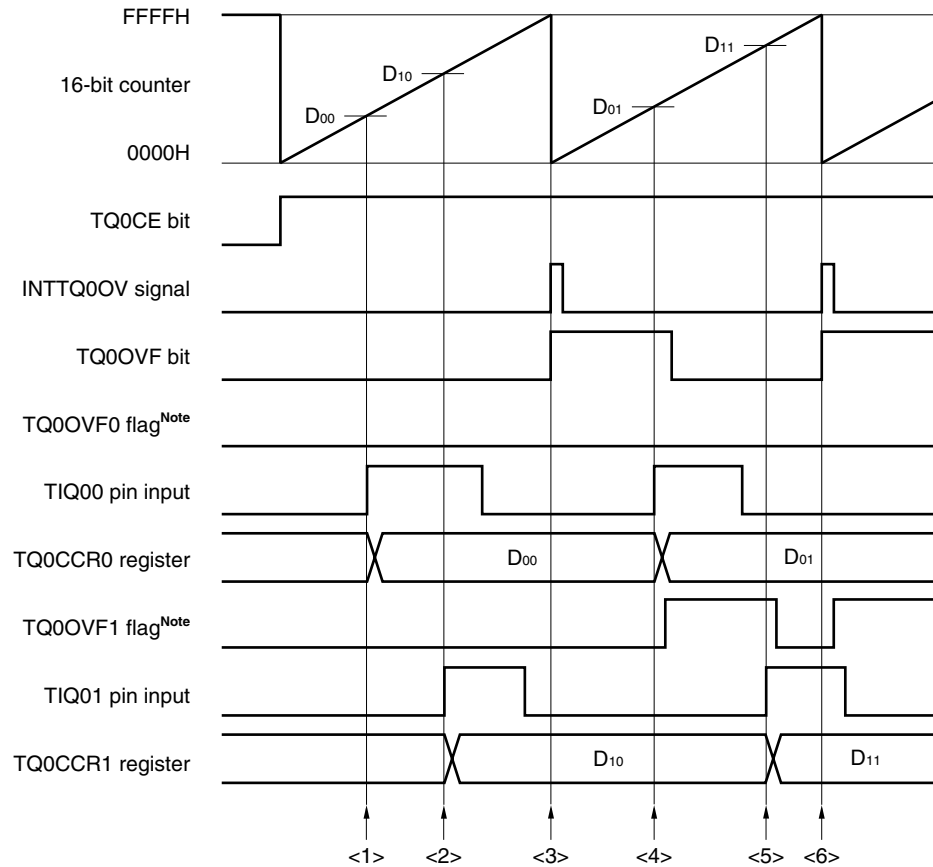
Use software when using two capture registers. An example of how to use software is shown below.

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>

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.

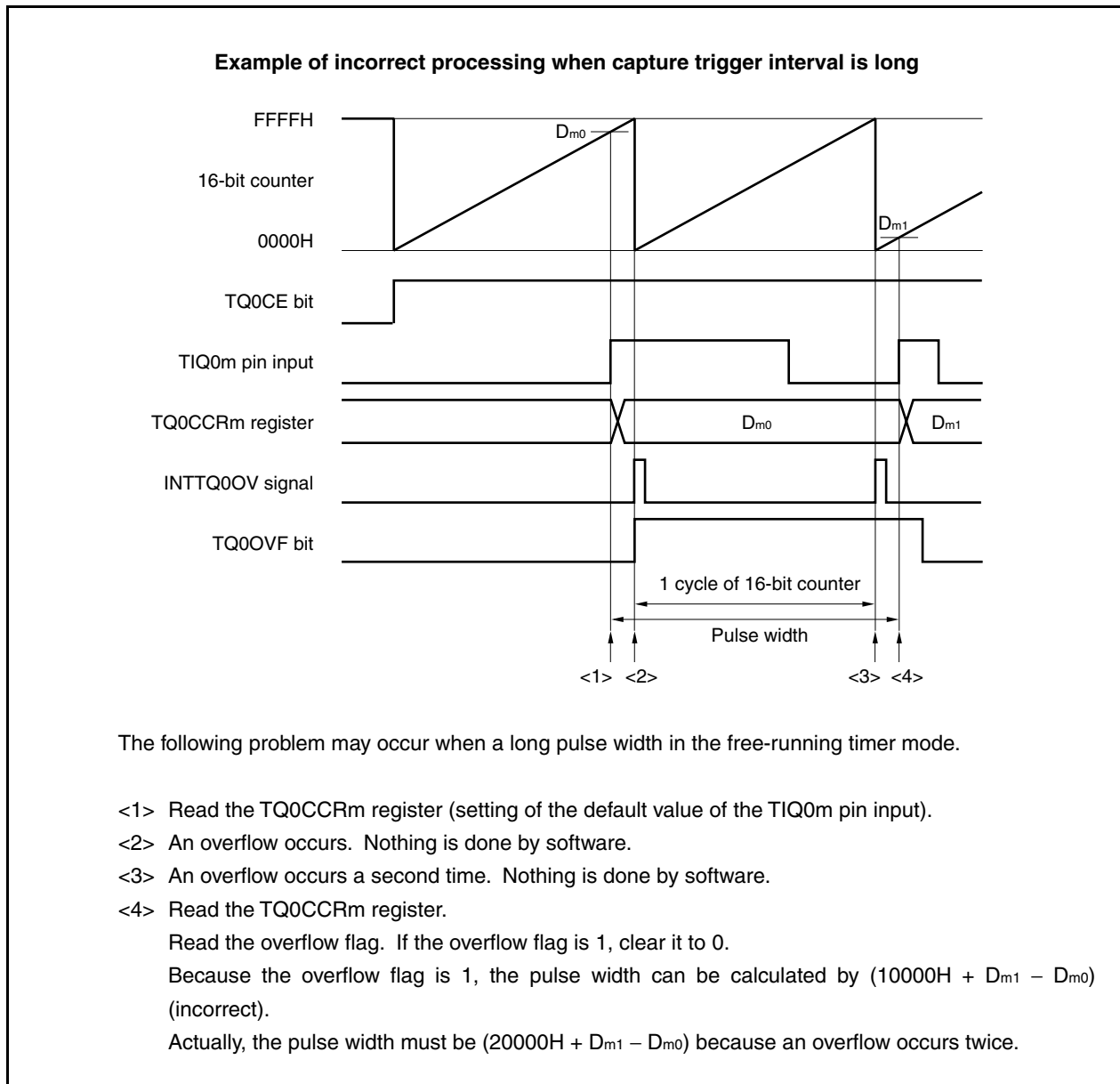
Read the TQ0OVF1 flag. If the TQ0OVF1 flag is 1, clear it to 0.

Because the TQ0OVF1 flag is 1, the pulse width can be calculated by $(10000H + D_{11} - D_{10})$ (correct).

<6> Same as <3>

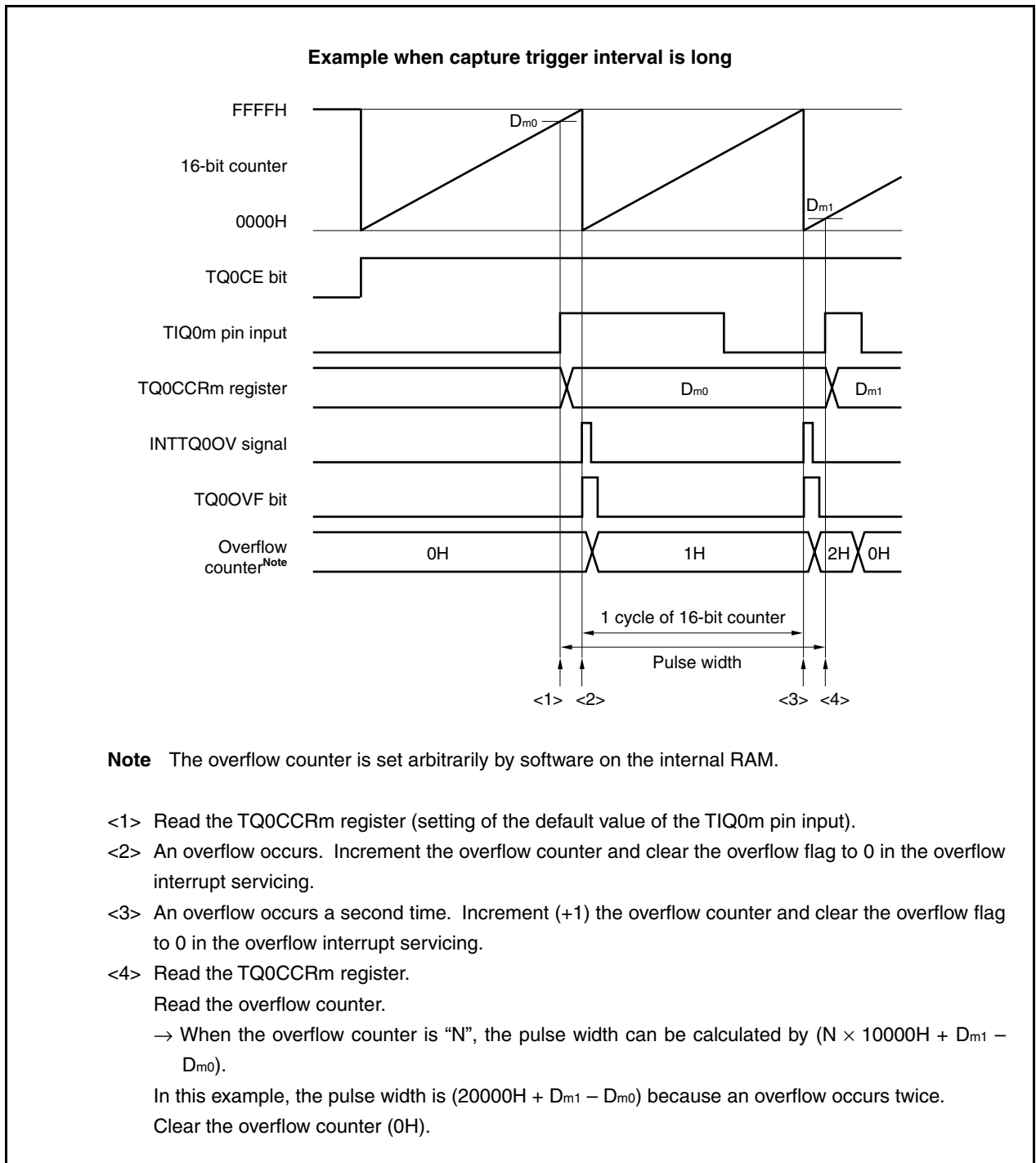
(d) Processing of overflow if capture trigger interval is long

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.



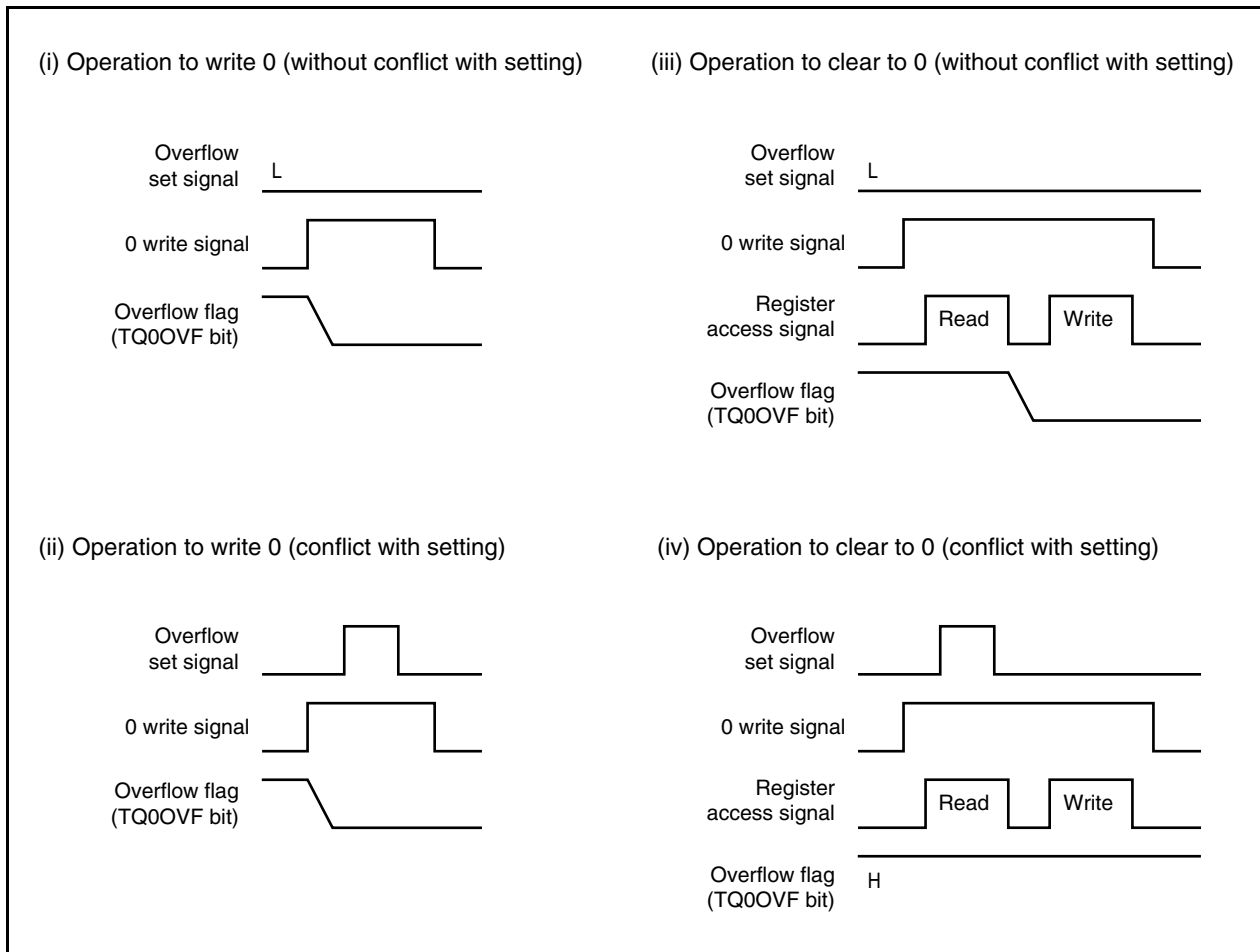
If an overflow occurs twice or more when the capture trigger interval is long, the correct pulse width may not be obtained.

If the capture trigger interval is long, slow the count clock to lengthen one cycle of the 16-bit counter, or use software. An example of how to use software is shown next.



(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.

7.5.7 Pulse width measurement mode (TQ0MD2 to TQ0MD0 bits = 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 7-34. Configuration in Pulse Width Measurement Mode

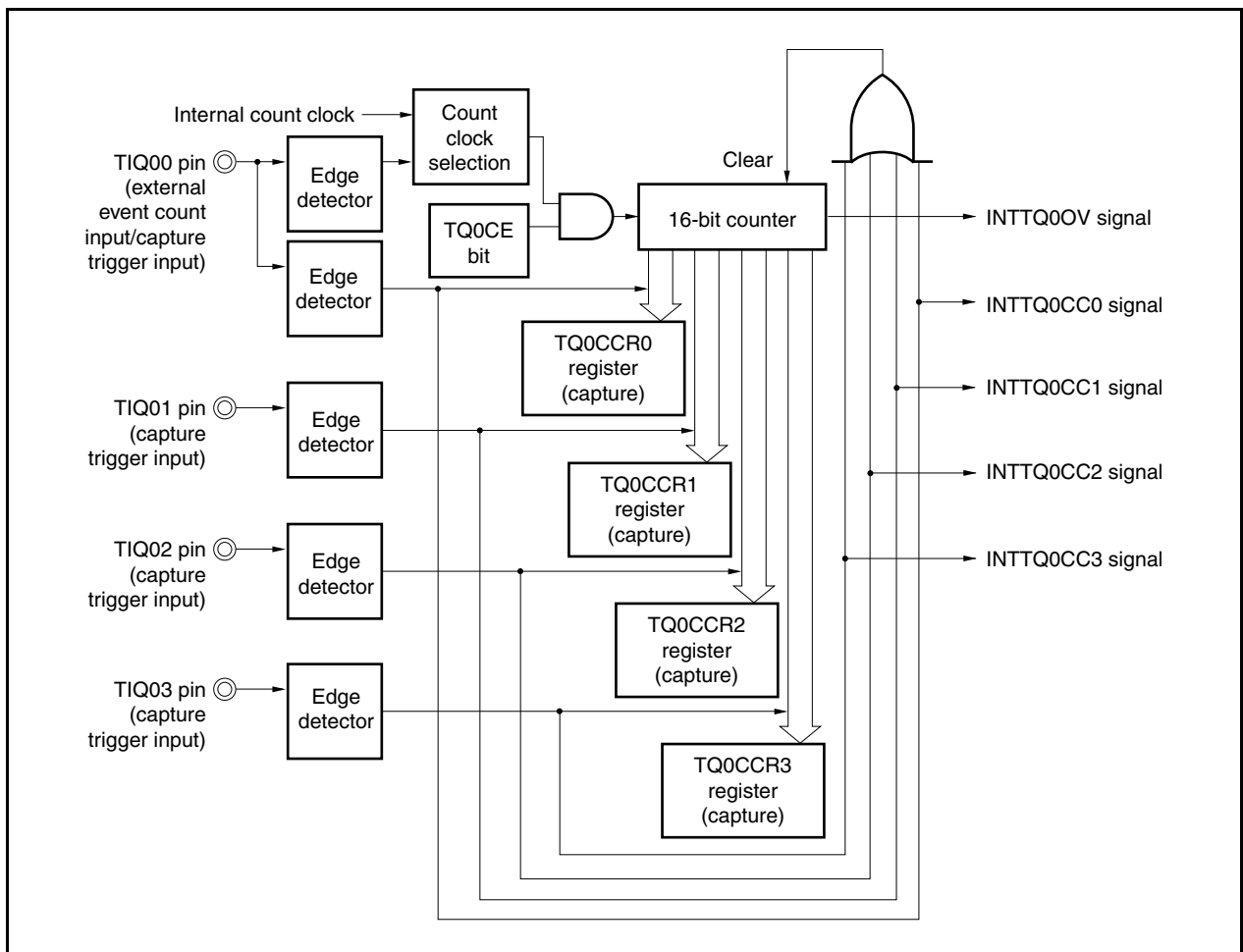
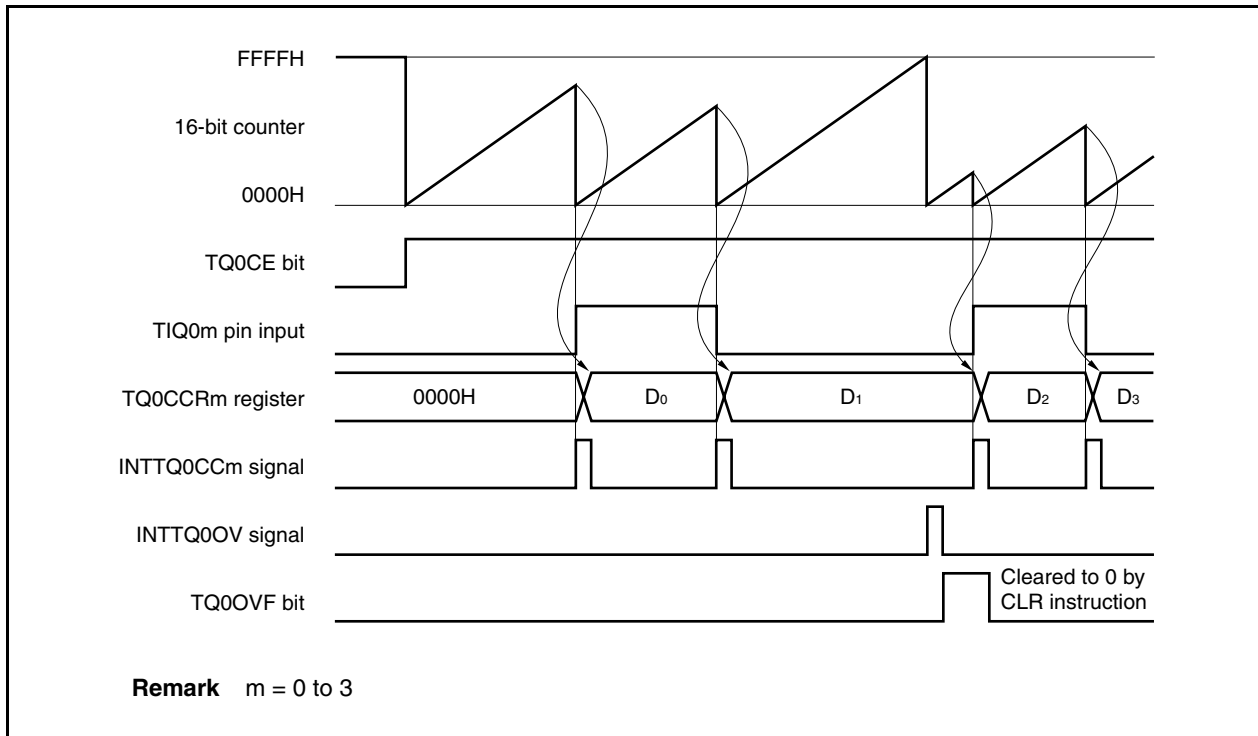


Figure 7-35. Basic Timing in Pulse Width Measurement Mode



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.

$$\text{Pulse width} = \text{Captured value} \times \text{Count clock cycle}$$

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.

$$\text{Pulse width} = (10000\text{H} \times \text{TQ0OVF bit set (1) count} + \text{Captured value}) \times \text{Count clock cycle}$$

Remark m = 0 to 3

Figure 7-36. Register Setting in Pulse Width Measurement Mode (1/2)

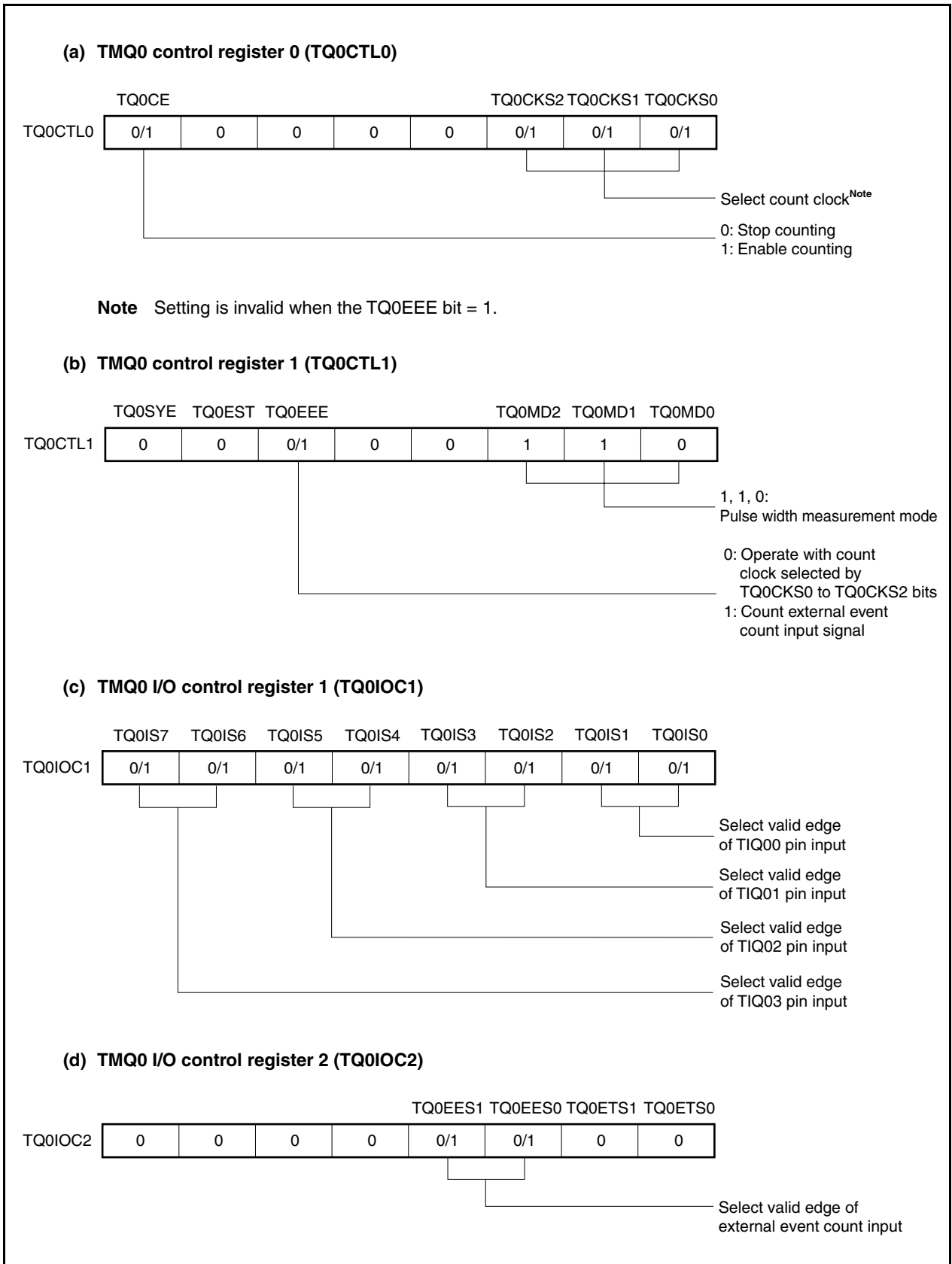
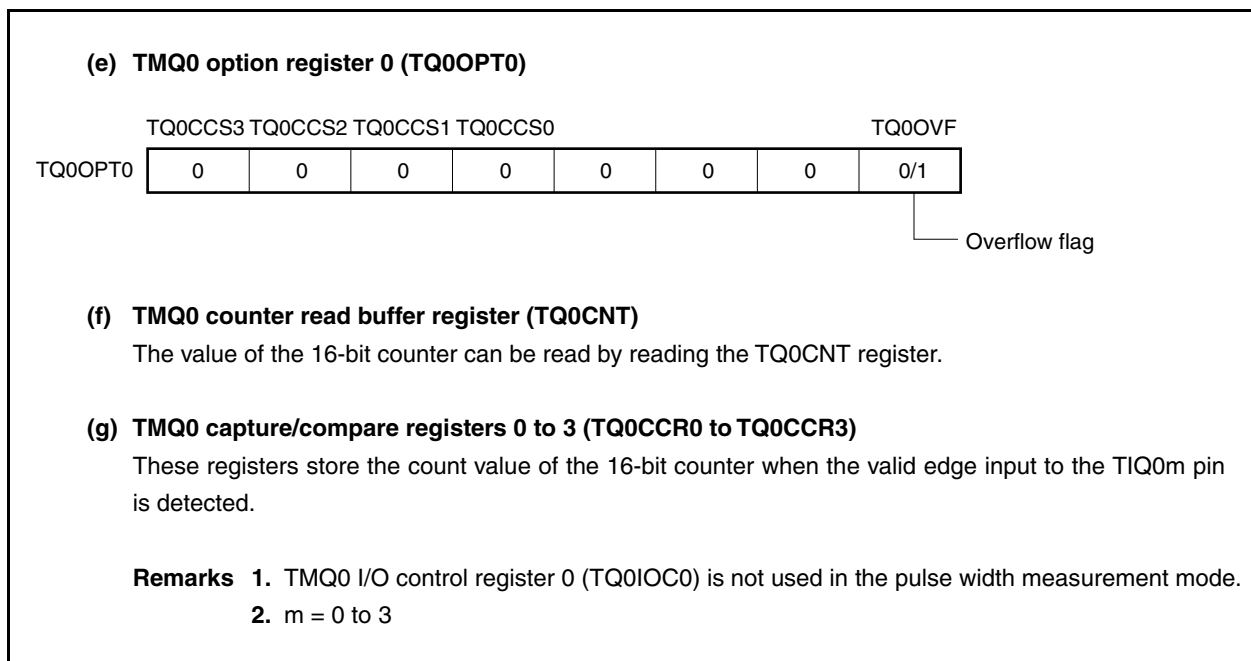
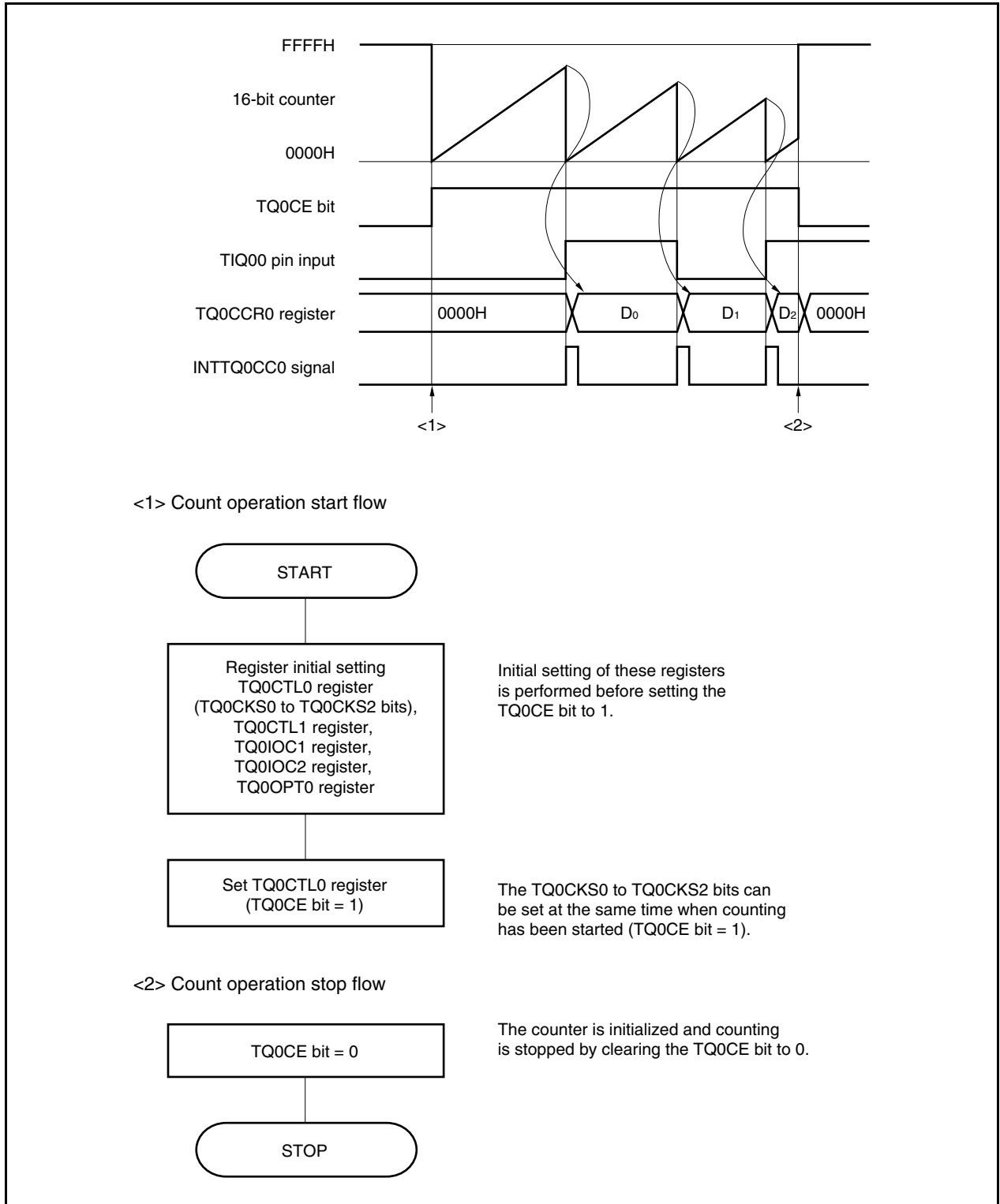


Figure 7-36. Register Setting in Pulse Width Measurement Mode (2/2)



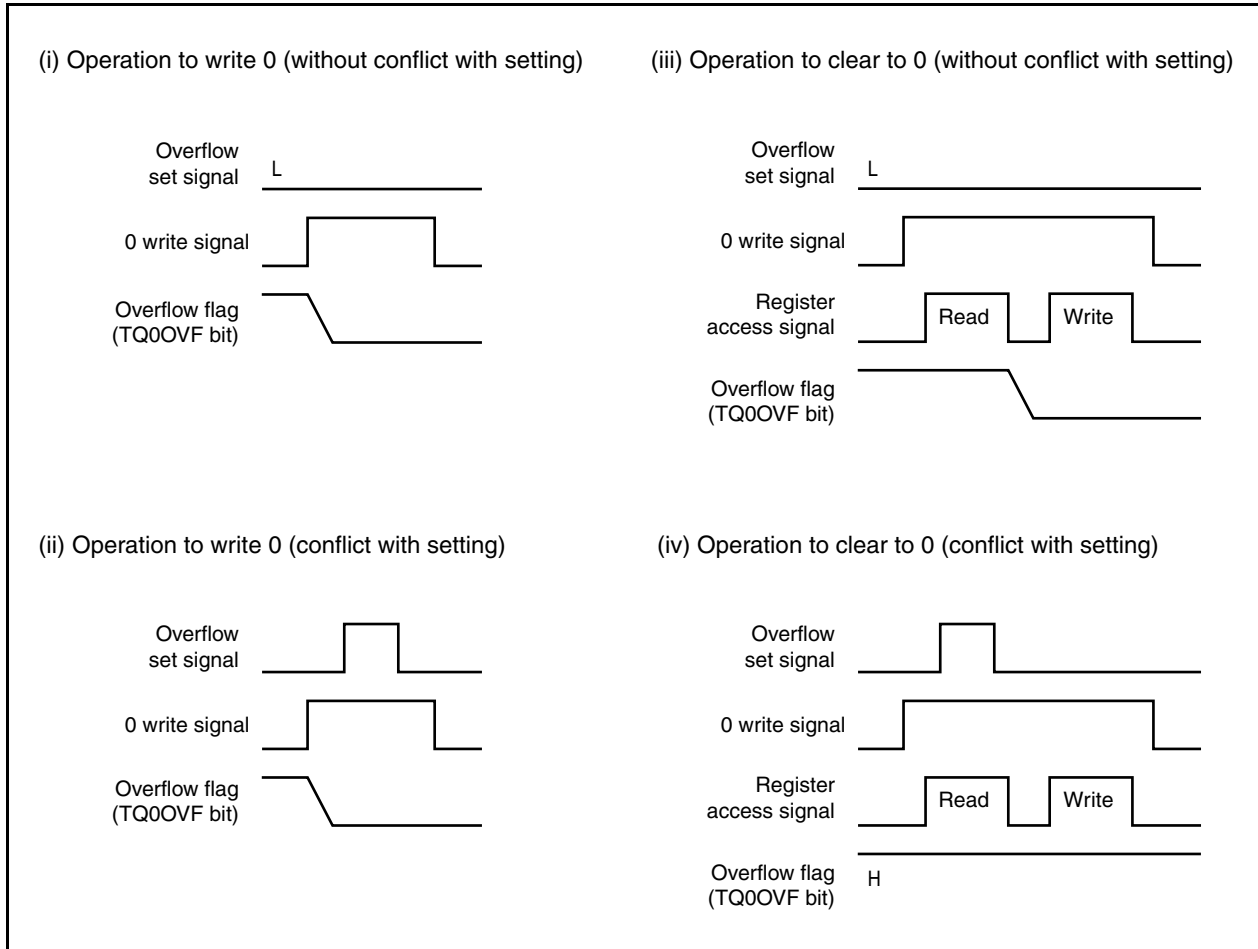
(1) Operation flow in pulse width measurement mode

Figure 7-37. Software Processing Flow in Pulse Width Measurement Mode



(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.

7.5.8 Triangular wave PWM mode (TQ0MD2 to TQ0MD0 = 111)

In the triangular wave PWM mode, TMQ0 capture/compare register k (TQ0CCRk) is used to set the duty factor, and TMQ0 capture/compare register 0 (TQ0CCR0) is used to set the cycle.

By using these four registers and operating the timer, triangular wave PWM with a variable cycle is output.

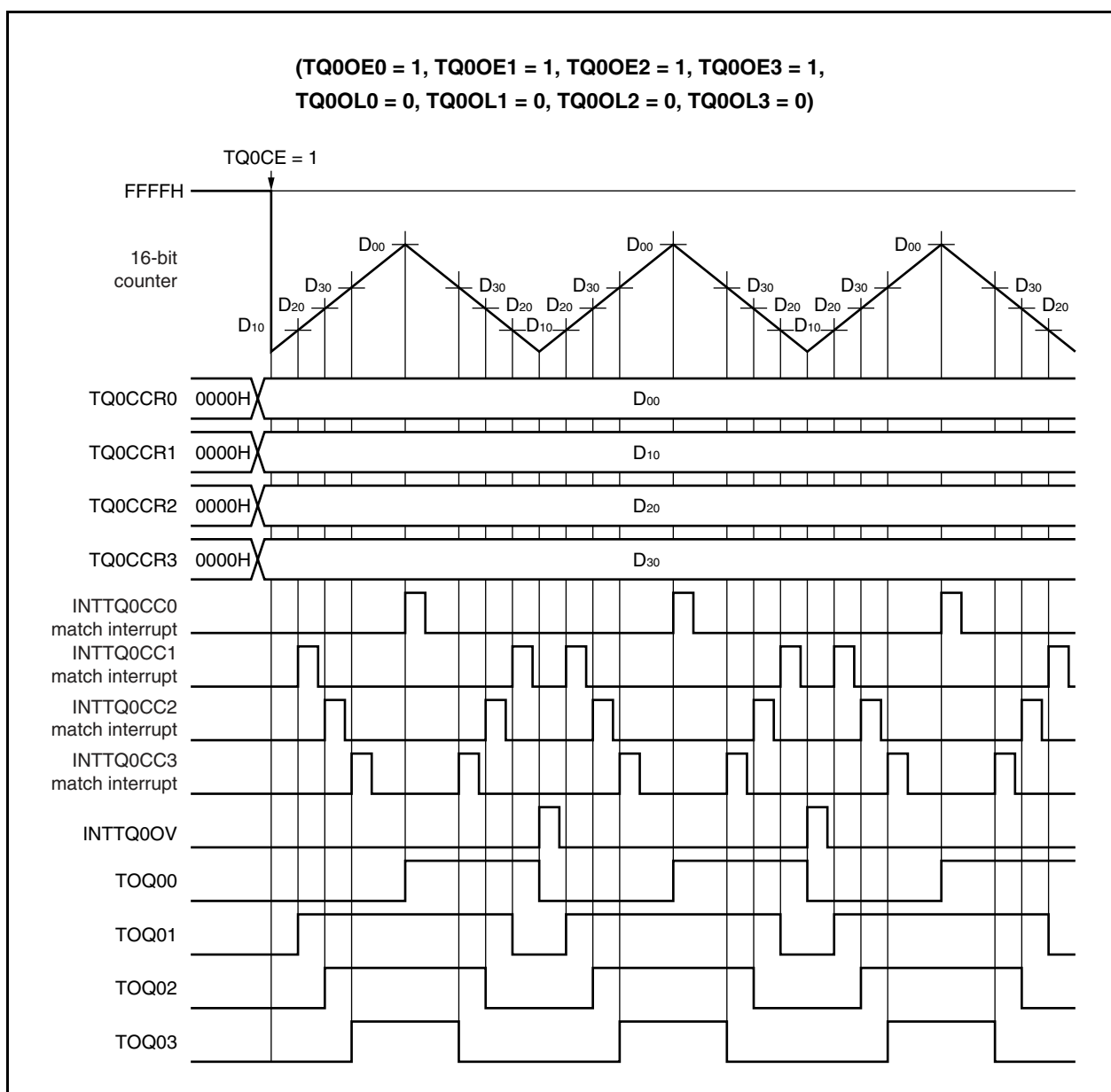
The value of the TQ0CCRm register can be rewritten when TQ0CE = 1.

To stop timer Q, clear TQ0CE to 0. The waveform of PWM is output from the TOQ0k pin. The TOQ00 pin produces a toggle output when the value of the 16-bit counter matches the value of the TQ0CCR0 register and when the counter underflows.

Caution In the PWM mode, the capture function of the TQ0CCRm register cannot be used because this register can be used only as a compare register.

Remark m = 0 to 3, k = 1 to 3

Figure 7-38. Timing of Basic Operation in Triangular Wave PWM Mode



7.5.9 Timer output operations

The following table shows the operations and output levels of the TOQ00 to TOQ03 pins.

Table 7-6. 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 | – | | | |
| Triangular wave PWM output mode | Square wave output | Triangular wave PWM output | Triangular wave PWM output | Triangular wave PWM output |

Table 7-7. 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

7.6 Timer Tuned Operation Function

Timer P and timer Q have a timer tuned operation function.
The timers that can be synchronized are listed in Table 7-8.

Table 7-8. Tuned Operation Mode of Timers

| Master Timer | Slave Timer | |
|--------------|-------------|------|
| TMP0 | TMP1 | – |
| TMP2 | TMP3 | TMQ0 |

- Cautions**
- The tuned operation mode is enabled or disabled by the TPmCTL1.TPmSYE and TQ0CTL1.TQ0SYE bits. For TMQ2, either or both TMQ3 and TMQ0 can be specified as slaves.
 - Set the tuned operation mode using the following procedure.
 - <1> Set the TPmCTL1.TPmSYE and TQ0CTL1.TQ0SYE bits of the slave timer to enable the tuned operation.
Set the TPmCTL1.TPmMD2 to TPmCTL1.TPmMD0 and TQ0CTL1.TQ0MD2 to TQ0CTL1.TQ0MD0 bits of the slave timer to the free-running mode.
 - <2> Set the timer mode by using the TPnCTL1.TPnMD2 to TPnCTL1.TPnMD0 bits.
At this time, do not set the TPnCTL1.TPnSYE bit of the master timer.
 - <3> Set the compare register value of the master and slave timers.
 - <4> Set the TPmCTL0.TPmCE and TQ0CTL0.TQ0CE bits of the slave timer to enable operation on the internal operating clock.
 - <5> Set the TPnCTL0.TPnCE bit of the master timer to enable operation on the internal operating clock.

Remark m = 1, 3

Tables 7-9 and 7-10 show the timer modes that can be used in the tuned operation mode (√: Settable, ×: Not settable).

Table 7-9. Timer Modes Usable in Tuned Operation Mode

| Master Timer | Free-Running Mode | PWM Mode | Triangular Wave PWM Mode |
|--------------|-------------------|----------|--------------------------|
| TMP0 | √ | √ | × |
| TMP2 | √ | √ | × |

Table 7-10. Timer Output Functions

| Tuned Channel | Timer | Pin | Free-Running Mode | | PWM Mode | | Triangular Wave PWM Mode | |
|---------------|---------------|----------------|-------------------|-----------|------------|-----------|--------------------------|-----------|
| | | | Tuning OFF | Tuning ON | Tuning OFF | Tuning ON | Tuning OFF | Tuning ON |
| Ch0 | TMP0 (master) | TOP00 | PPG | ← | Toggle | ← | N/A | ← |
| | | TOP01 | PPG | ← | PWM | ← | N/A | ← |
| | TMP1 (slave) | TOP10 | PPG | ← | Toggle | PWM | N/A | ← |
| | | TOP11 | PPG | ← | PWM | ← | N/A | ← |
| Ch1 | TMP2 (master) | TOP20 | PPG | ← | Toggle | ← | N/A | ← |
| | | TOP21 | PPG | ← | PWM | ← | N/A | ← |
| | TMP3 (slave) | TOP30 | PPG | ← | Toggle | PWM | N/A | ← |
| | | TOP31 | PPG | ← | PWM | ← | N/A | ← |
| | TMQ0 (slave) | TOQ00 | PPG | ← | Toggle | PWM | Toggle | N/A |
| | | TOQ01 to TOQ03 | PPG | ← | PWM | ← | Triangular wave PWM | N/A |

Remark The timing of transmitting data from the compare register of the master timer to the compare register of the slave timer is as follows.

PPG: CPU write timing

Toggle, PWM, triangular wave PWM: Timing at which timer counter and compare register match TOPn0 and TOQ00 (n = 0 to 3)

Figure 7-39. Tuned Operation Image (TMP2, TMP3, TMQ0)

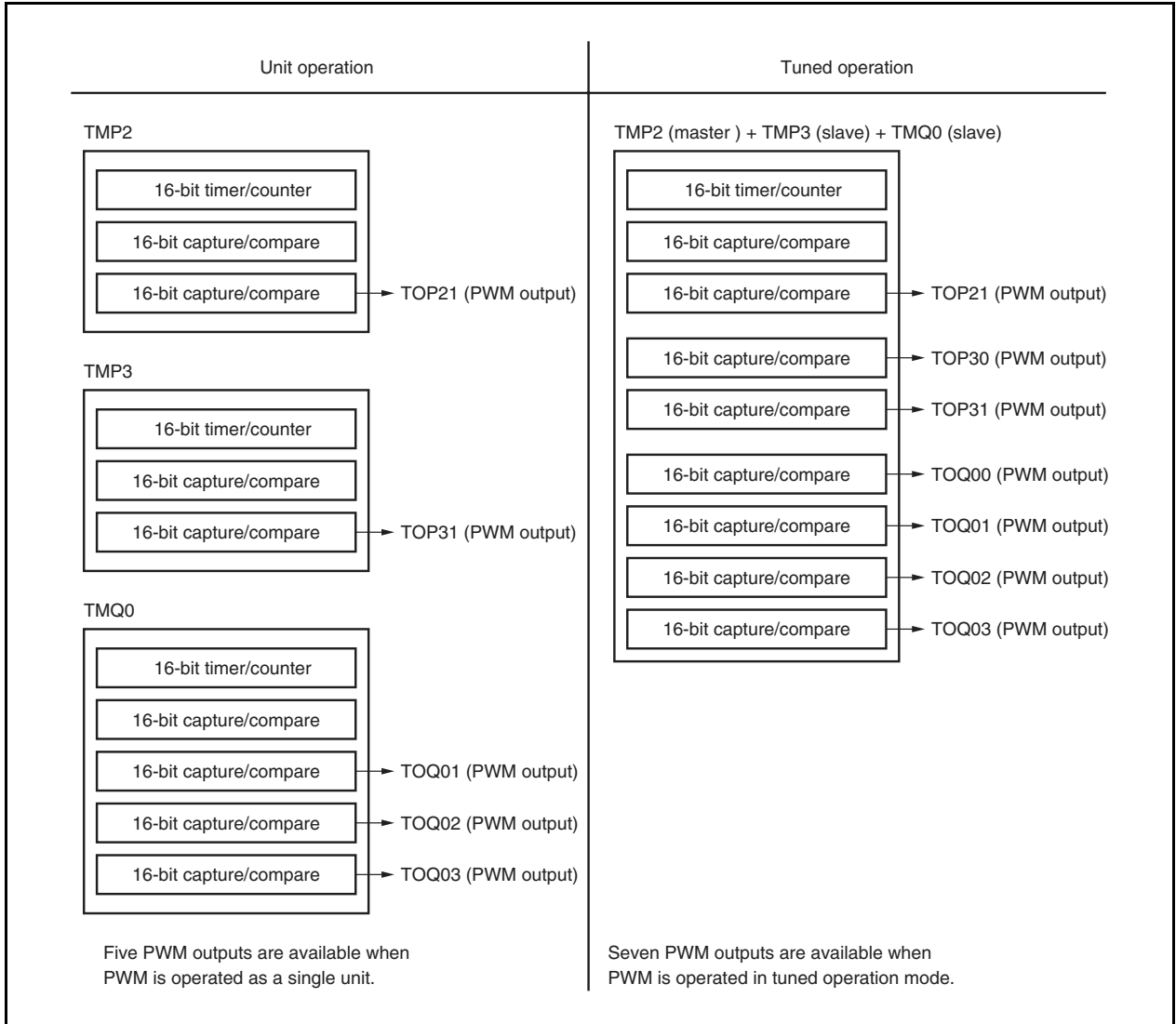
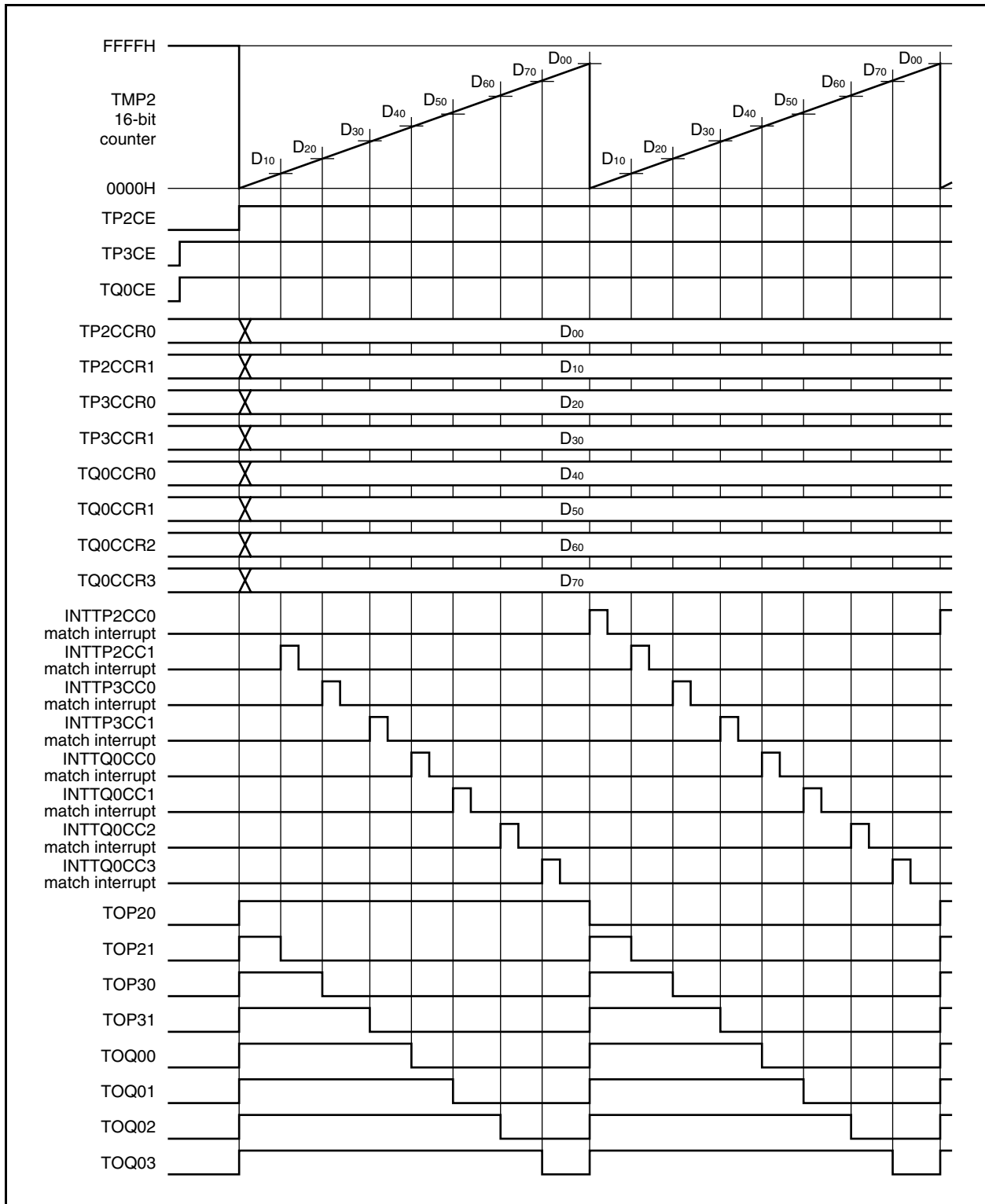


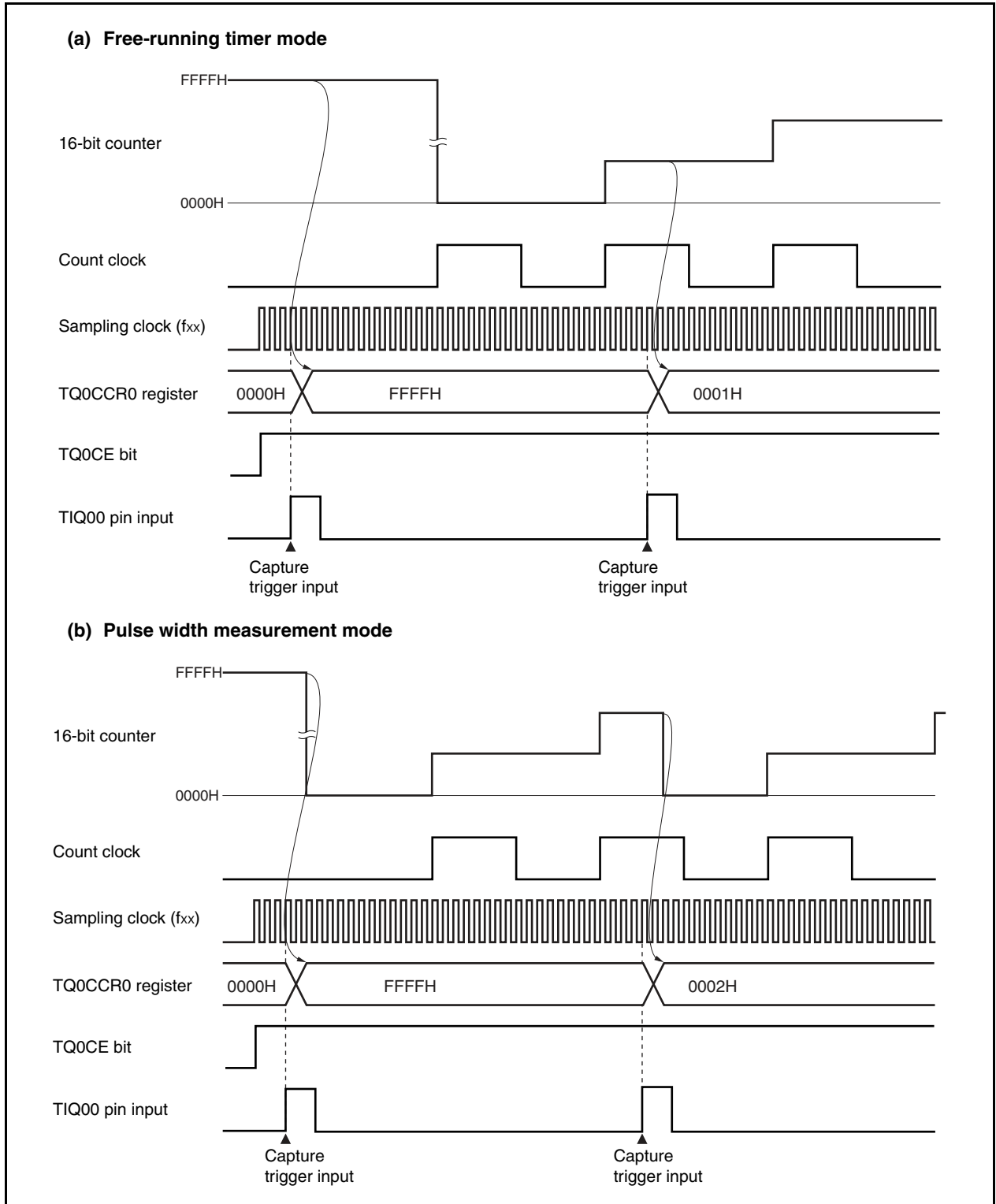
Figure 7-40. Basic Operation Timing of Tuned PWM Function (TMP2, TMP3, TMQ0)



7.7 Cautions

(1) Capture operation

When the capture operation is used and a slow clock is selected as the count clock, FFFFH, not 0000H, may be captured in the TQ0CCR0, TQ0CCR1, TQ0CCR2, and TQ0CCR3 registers if the capture trigger is input immediately after the TQ0CE bit is set to 1.



CHAPTER 8 16-BIT INTERVAL TIMER M (TMM)

8.1 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.

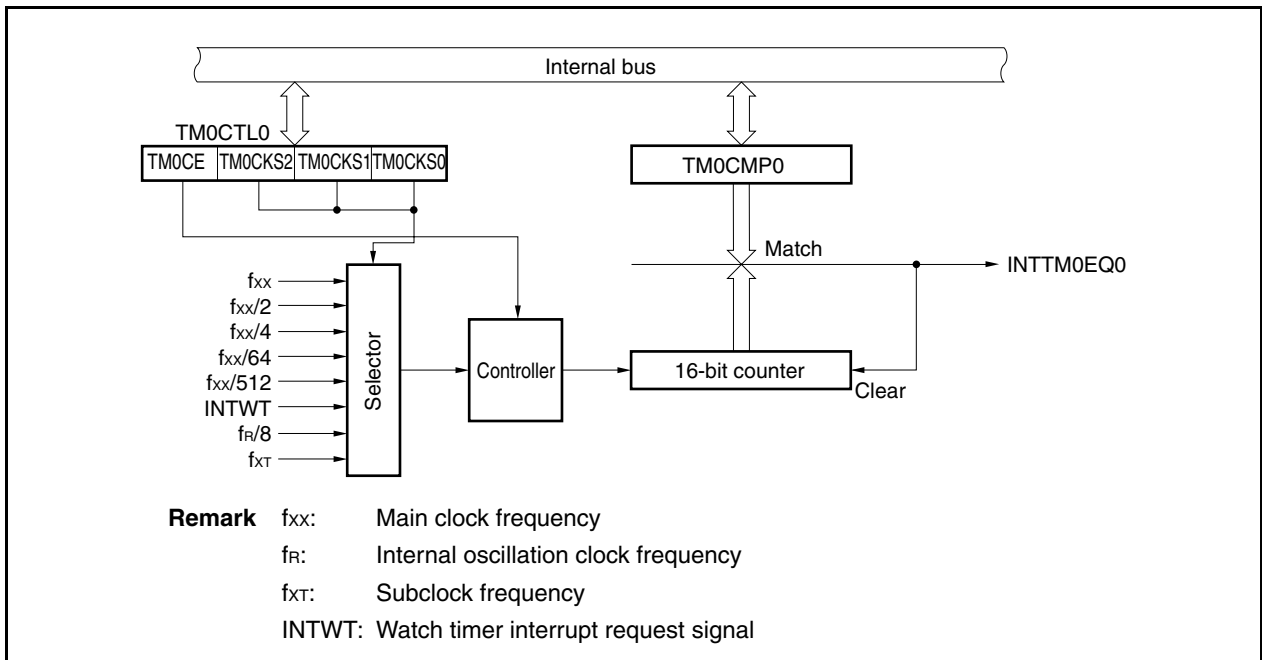
8.2 Configuration

TMM0 includes the following hardware.

Table 8-1. Configuration of TMM0

| Item | Configuration |
|------------------|-----------------------------------|
| Timer register | 16-bit counter |
| Register | TMM0 compare register 0 (TM0CMP0) |
| Control register | TMM0 control register 0 (TM0CTL0) |

Figure 8-1. Block Diagram of TMM0

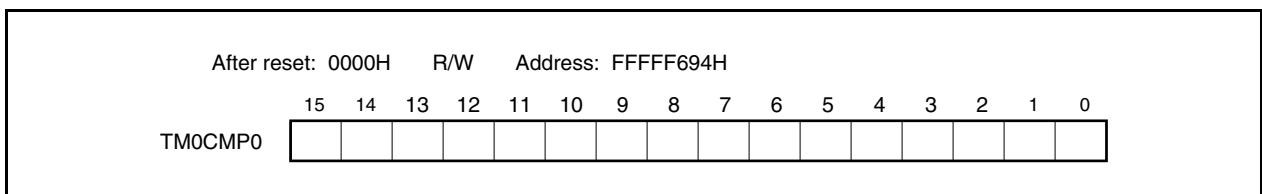


(1) 16-bit counter

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 sets 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.



8.3 Register

(1) TMM0 control register (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 sets this register to 00H.

The same value can always be written to the TM0CTL0 register by software. Rewriting this register, except the TM0CE bit, is prohibited while the timer is operating.

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 application stopped. |
| 1 | TMM0 operation enabled. Operation clock application 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 disabled (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 TM0CE from 0 to 1, it is not possible to set the value of the TM0CKS2 to TM0CKS0 bits simultaneously.
 2. Be sure to clear bits 3 to 6 to "0".

Remark

f_{xx}: Main clock frequency
 f_R: Internal oscillation clock frequency
 f_{XT}: Subclock frequency

8.4 Operation

Caution Do not set the TM0CMP0 register to FFFFH.

8.4.1 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 8-2. Configuration of Interval Timer

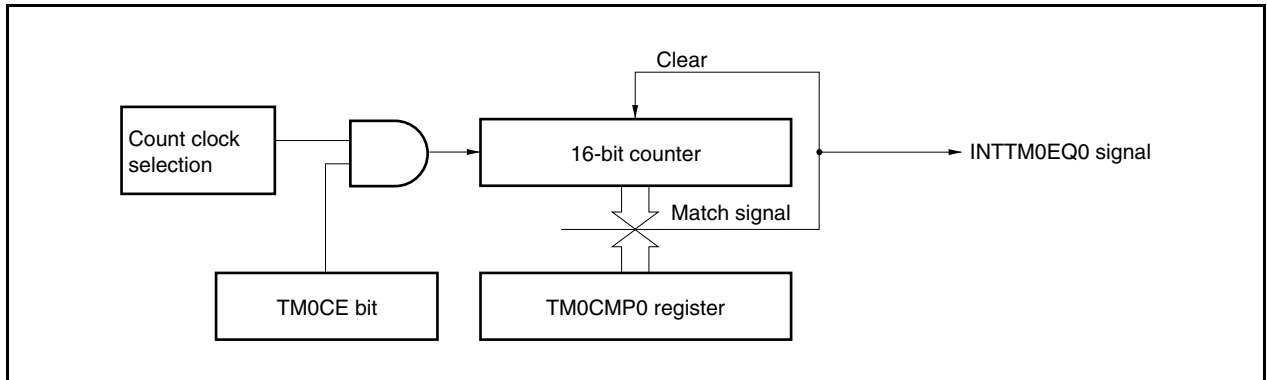
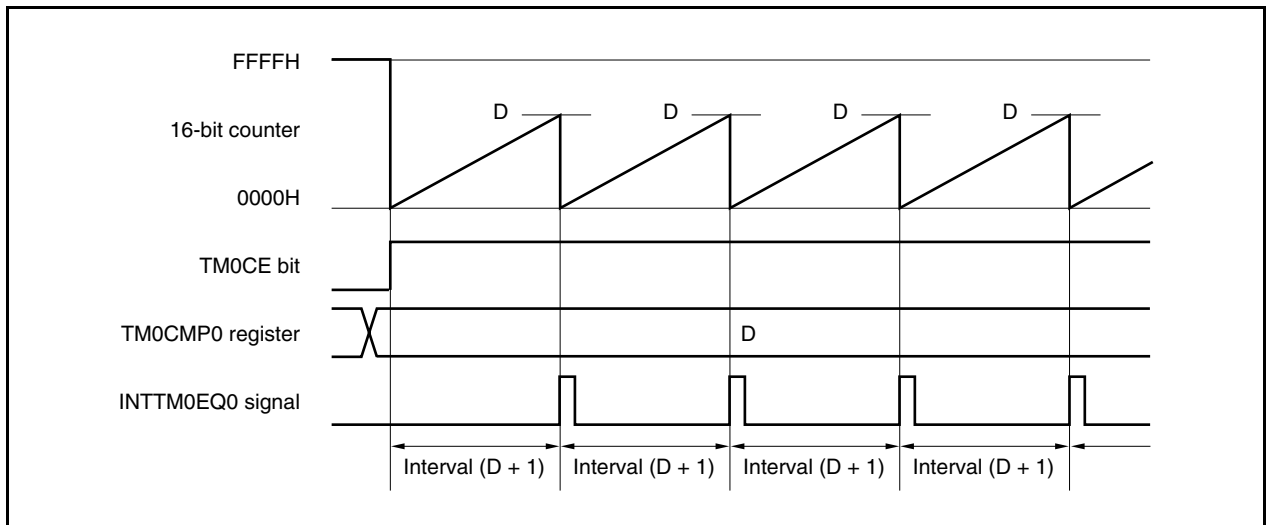


Figure 8-3. Basic Timing of Operation in Interval Timer Mode



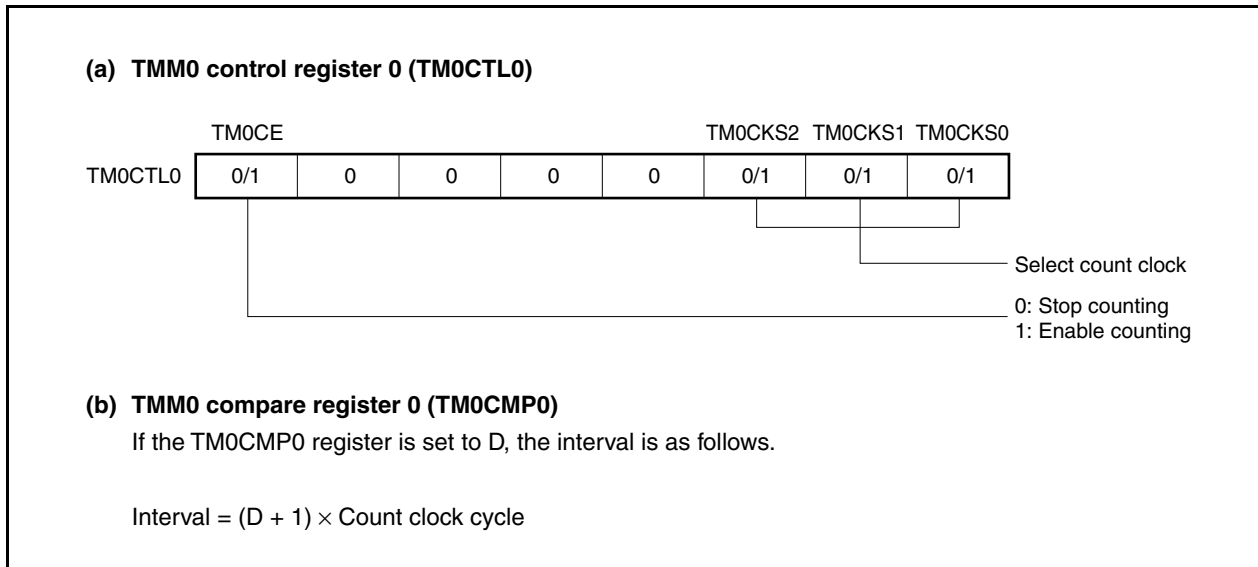
When the TM0CE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H in synchronization with the count clock, and the counter starts counting.

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.

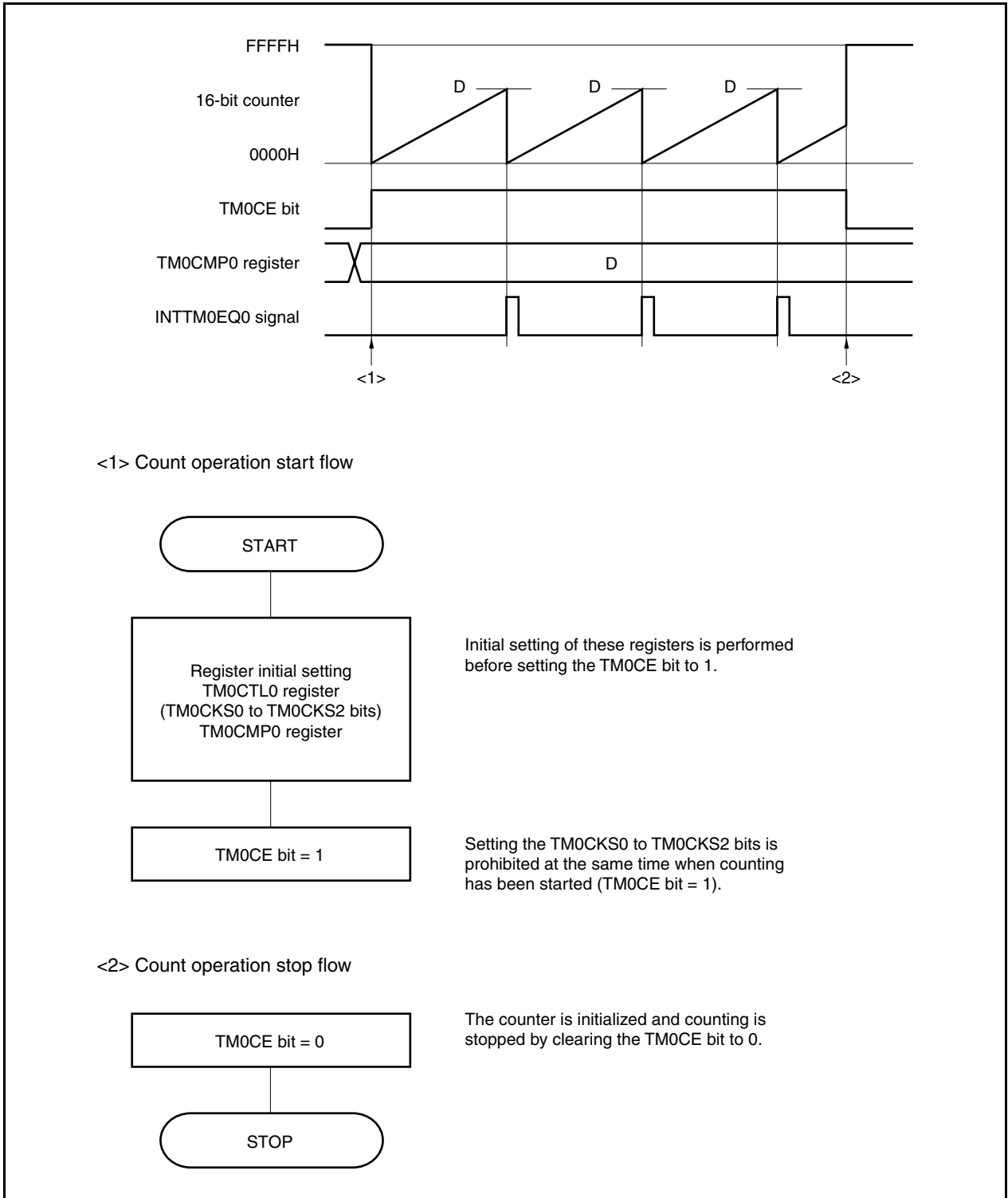
$$\text{Interval} = (\text{Set value of TM0CMP0 register} + 1) \times \text{Count clock cycle}$$

Figure 8-4. Register Setting for Interval Timer Mode Operation



(1) Interval timer mode operation flow

Figure 8-5. Software Processing Flow in Interval Timer Mode

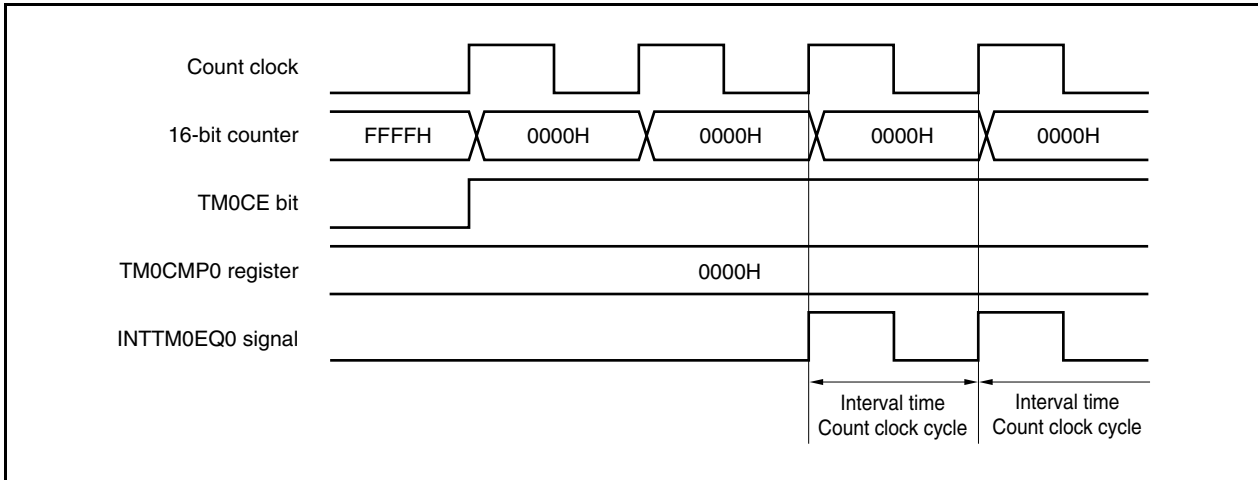


(2) Interval timer mode operation timing

Caution Do not set the TM0CMP0 register to FFFFH.

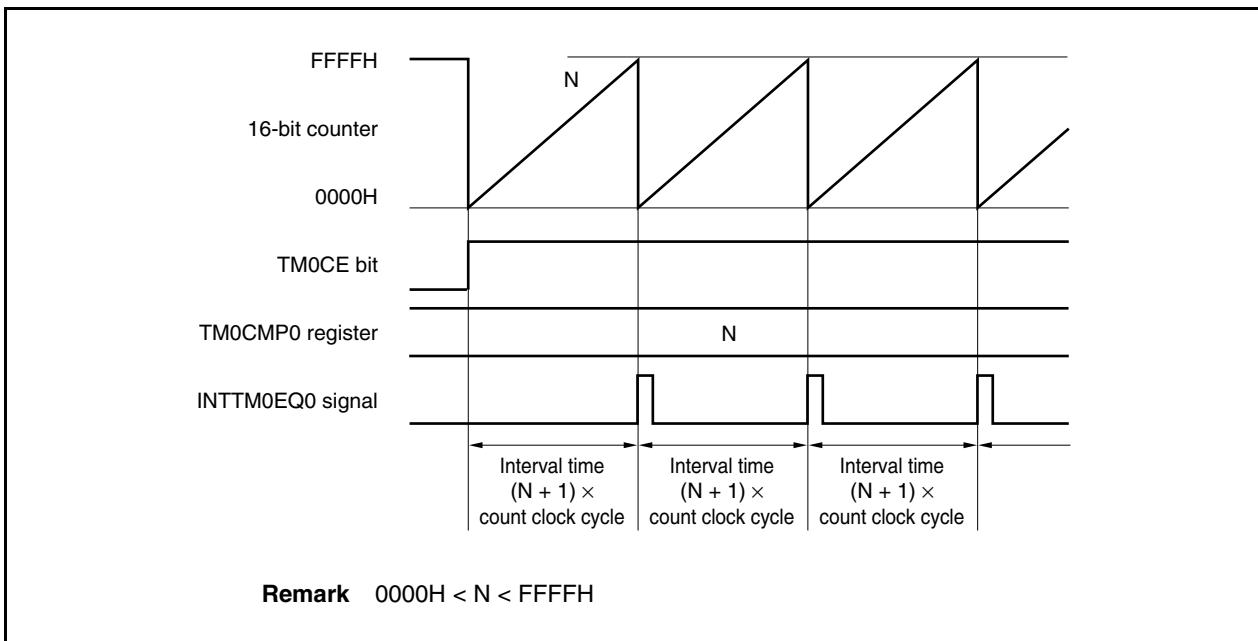
(a) Operation if TM0CMP0 register is set to 0000H

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 N

If the TM0CMP0 register is set to N, the 16-bit counter counts up to N. The counter is cleared to 0000H in synchronization with the next count-up timing and the INTTM0EQ0 signal is generated.



8.4.2 Cautions

- (1) It takes the 16-bit counter up to the following time to start counting after the TMOCTL0.TMOCE 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 TMOCMP0 and TMOCTL0 registers is prohibited while TMM0 is operating. If these registers are rewritten while the TMOCE bit is 1, the operation cannot be guaranteed. If they are rewritten by mistake, clear the TMOCTL0.TMOCE bit to 0, and re-set the registers.

CHAPTER 9 WATCH TIMER FUNCTIONS

9.1 Functions

The watch timer has the following functions.

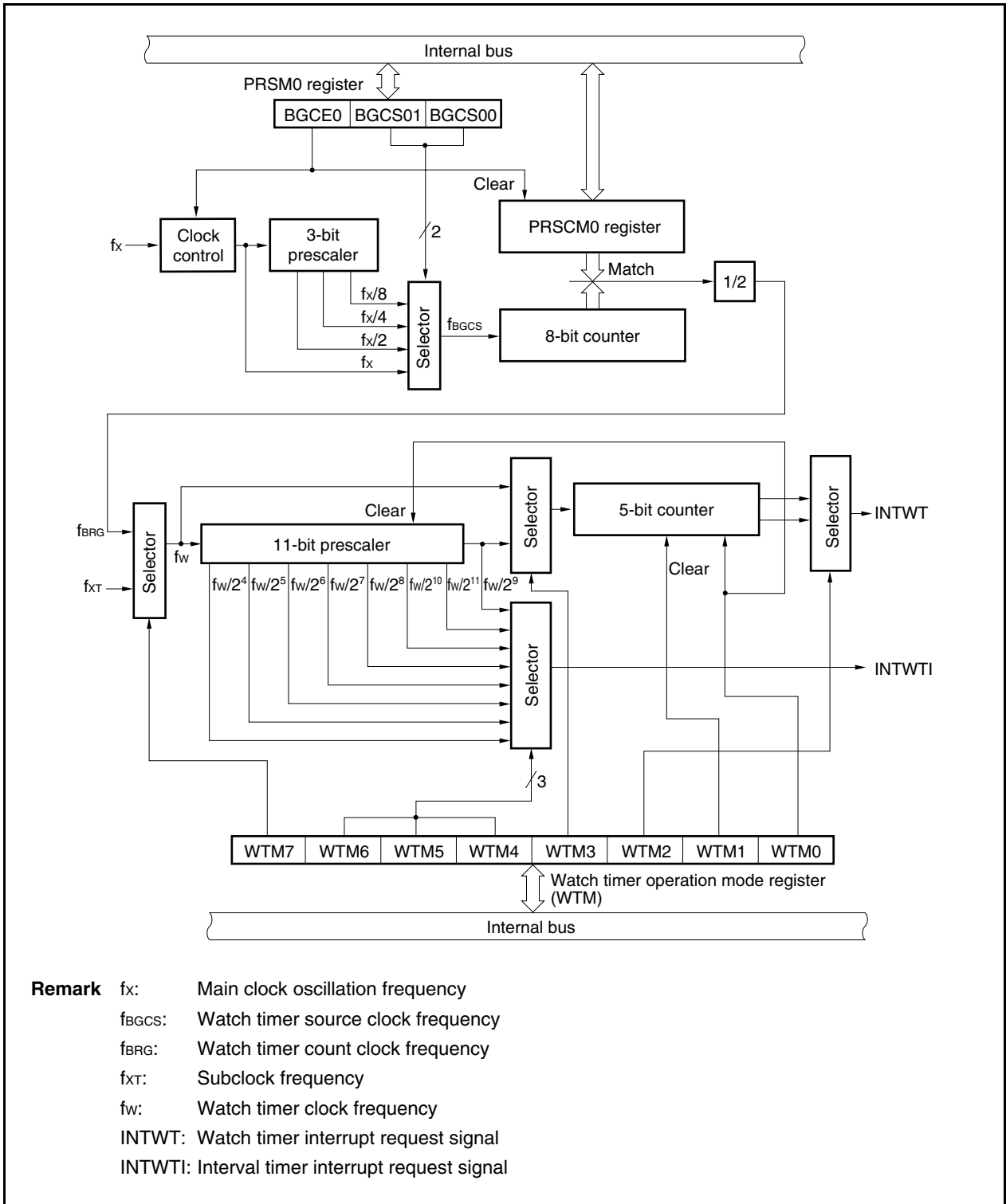
- Watch timer: An interrupt request signal (INTWT) is generated at intervals of 0.5 or 0.25 seconds by using the main clock or subclock.
- Interval timer: An interrupt request signal (INTWTI) is generated at set intervals.

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

9.2 Configuration

The block diagram of the watch timer is shown below.

Figure 9-1. Block Diagram of Watch Timer



(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_{BGS}).

(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 $2^4/f_w$, $2^5/f_w$, $2^{12}/f_w$, 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.

9.3 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 sets 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_{BGS}) | 5 MHz | 4 MHz |
| 0 | 0 | f_x | 200 ns | 250 ns |
| 0 | 1 | $f_x/2$ | 400 ns | 500 ns |
| 1 | 0 | $f_x/4$ | 800 ns | 1 μ s |
| 1 | 1 | $f_x/8$ | 1.6 μ s | 2 μ s |

Cautions

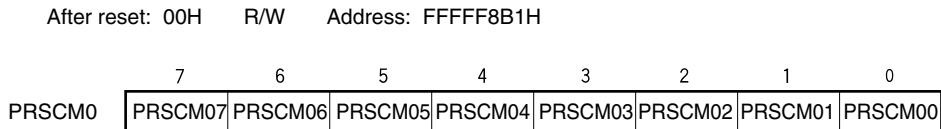
1. Do not change the values of the BGCS00 and BGCS01 bits during watch timer operation.
2. Set the PRSM0 register before setting the BGCE0 bit to 1.
3. Set the PRSM0 and PRSCM0 registers according to the main clock frequency that is used so as to obtain an f_{BRG} frequency of 32.768 kHz.

(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 sets this register to 00H.



- 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.

The calculation for f_{BRG} is shown below.

$$f_{BRG} = f_{BGCS}/2N$$

Remark f_{BGCS} : Watch timer source clock set by the PRSM0 register

N: Set value of PRSCM0 register = 1 to 256

However, N = 256 only when PRSCM0 register is set to 00H.

(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.

Set the PRSM0 register before setting the WTM register.

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

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: FFFFF680H

| | | | | | | | |
|------|------|------|------|------|------|------|------|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 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 μ s: $f_w = f_{XT}$) |
| 0 | 0 | 0 | 1 | $2^5/f_w$ (977 μ s: $f_w = f_{XT}$) |
| 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 μ s: $f_w = f_{BRG}$) |
| 1 | 0 | 0 | 1 | $2^5/f_w$ (977 μ s: $f_w = f_{BRG}$) |
| 1 | 0 | 1 | 0 | $2^6/f_w$ (1.95 ms: $f_w = f_{BRG}$) |
| 1 | 0 | 1 | 1 | $2^7/f_w$ (3.90 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 | $2^{10}/f_w$ (31.2 ms: $f_w = f_{BRG}$) |
| 1 | 1 | 1 | 1 | $2^{11}/f_w$ (62.5 ms: $f_w = f_{BRG}$) |

| 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 μ s: $f_w = f_{XT}$) |
| 0 | 1 | 1 | $2^4/f_w$ (488 μ s: $f_w = f_{XT}$) |
| 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 μ s: $f_w = f_{BRG}$) |
| 1 | 1 | 1 | $2^4/f_w$ (488 μ s: $f_w = f_{BRG}$) |

| 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
 3. f_{XT} : Subclock frequency
 4. f_{BRG} : Watch timer count clock frequency

9.4 Operation

9.4.1 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, set the count clock using the PRSM0.BGCS01 and BGCS00 bits, the 8-bit comparison value using the PRSCM0 register, and the count clock frequency (f_{BRG}) of the watch timer to 32.768 kHz.

When the PRSM0.BGCE0 bit is set (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 $N = f_x / 65,536$. Set $m = 0$.

<2> When the value resulting from rounding up the first decimal place of N is even, set N before the roundup as $N/2$ and m as $m + 1$.

<3> Repeat <2> until N is odd or $m = 3$.

<4> Set the value resulting from rounding up the first decimal place of N to the PRSCM0 register and m to the BGCS01 and BGCS00 bits.

Example: When $f_x = 4.00$ MHz

<1> $N = 4,000,000 / 65,536 = 61.03\dots$, $m = 0$

<2>, <3> Because N (round up the first decimal place) is odd, $N = 61$, $m = 0$.

<4> Set value of PRSCM0 register: 3DH (61), set value of BGCS01 and BGCS00 bits: 00

At this time, the actual f_{BRG} frequency is as follows.

$$\begin{aligned} f_{BRG} &= f_x / (2^{m+1} \times N) = 4,000,000 / (2 \times 61) \\ &= 32.787 \text{ kHz} \end{aligned}$$

Remark m : Division value (set value of BGCS01 and BGCS00 bits) = 0 to 3

N : Set value of PRSCM0 register = 1 to 256

However, $N = 256$ only when PRSCM0 register is set to 00H.

f_x : Main clock oscillation frequency

9.4.2 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.

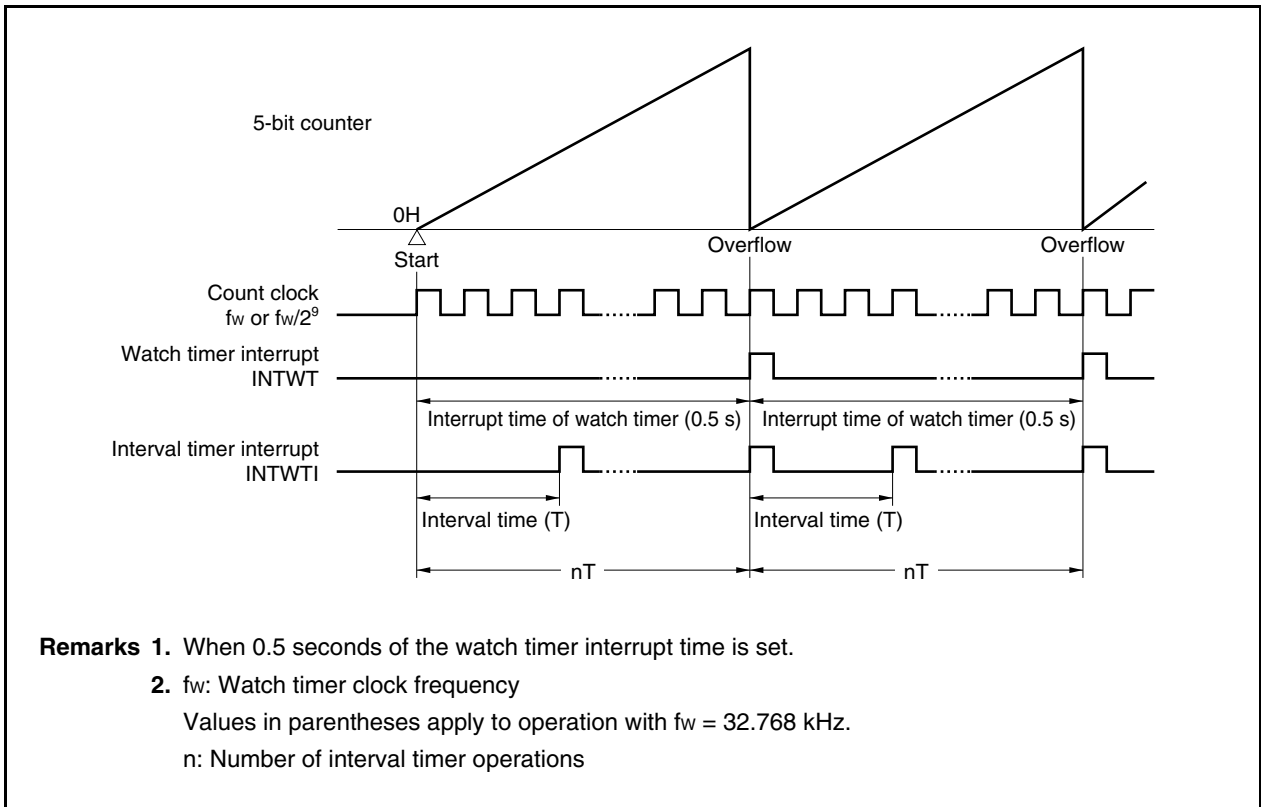
The interval time can be selected by the WTM4 to WTM7 bits of the WTM register.

Table 9-1. Interval Time of Interval Timer

| WTM7 | WTM6 | WTM5 | WTM4 | Interval Time | |
|------|------|------|------|-----------------------|---|
| 0 | 0 | 0 | 0 | $2^4 \times 1/f_w$ | 488 μ s (operating at $f_w = f_{XT} = 32.768$ kHz) |
| 0 | 0 | 0 | 1 | $2^5 \times 1/f_w$ | 977 μ s (operating at $f_w = f_{XT} = 32.768$ kHz) |
| 0 | 0 | 1 | 0 | $2^6 \times 1/f_w$ | 1.95 ms (operating at $f_w = f_{XT} = 32.768$ kHz) |
| 0 | 0 | 1 | 1 | $2^7 \times 1/f_w$ | 3.91 ms (operating at $f_w = f_{XT} = 32.768$ kHz) |
| 0 | 1 | 0 | 0 | $2^8 \times 1/f_w$ | 7.81 ms (operating at $f_w = f_{XT} = 32.768$ kHz) |
| 0 | 1 | 0 | 1 | $2^9 \times 1/f_w$ | 15.6 ms (operating at $f_w = f_{XT} = 32.768$ kHz) |
| 0 | 1 | 1 | 0 | $2^{10} \times 1/f_w$ | 31.3 ms (operating at $f_w = f_{XT} = 32.768$ kHz) |
| 0 | 1 | 1 | 1 | $2^{11} \times 1/f_w$ | 62.5 ms (operating at $f_w = f_{XT} = 32.768$ kHz) |
| 1 | 0 | 0 | 0 | $2^4 \times 1/f_w$ | 488 μ s (operating at $f_w = f_{BRG} = 32.768$ kHz) |
| 1 | 0 | 0 | 1 | $2^5 \times 1/f_w$ | 977 μ s (operating at $f_w = f_{BRG} = 32.768$ kHz) |
| 1 | 0 | 1 | 0 | $2^6 \times 1/f_w$ | 1.95 ms (operating at $f_w = f_{BRG} = 32.768$ kHz) |
| 1 | 0 | 1 | 1 | $2^7 \times 1/f_w$ | 3.91 ms (operating at $f_w = f_{BRG} = 32.768$ kHz) |
| 1 | 1 | 0 | 0 | $2^8 \times 1/f_w$ | 7.81 ms (operating at $f_w = f_{BRG} = 32.768$ kHz) |
| 1 | 1 | 0 | 1 | $2^9 \times 1/f_w$ | 15.6 ms (operating at $f_w = f_{BRG} = 32.768$ kHz) |
| 1 | 1 | 1 | 0 | $2^{10} \times 1/f_w$ | 31.3 ms (operating at $f_w = f_{BRG} = 32.768$ kHz) |
| 1 | 1 | 1 | 1 | $2^{11} \times 1/f_w$ | 62.5 ms (operating at $f_w = f_{BRG} = 32.768$ kHz) |

Remark f_w : Watch timer clock frequency

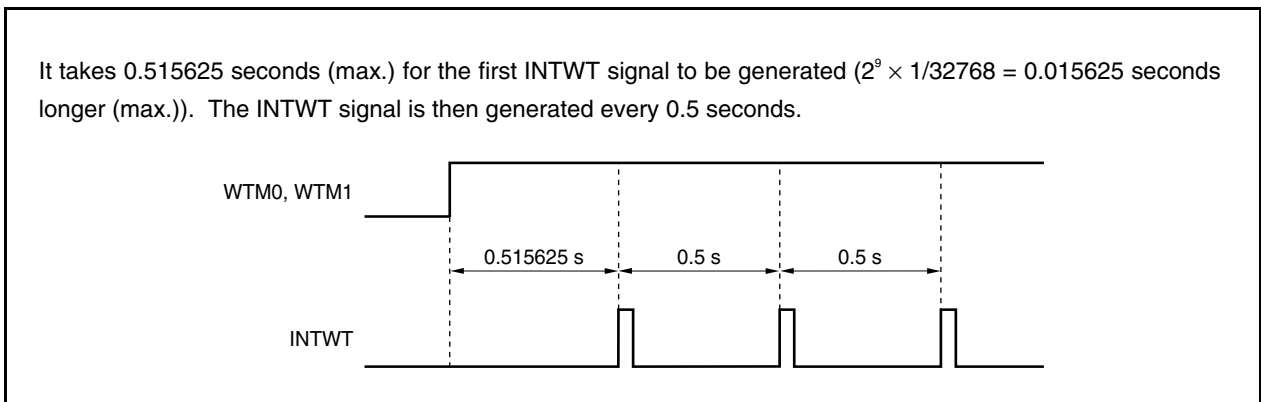
Figure 9-2. Operation Timing of Watch Timer/Interval Timer



9.4.3 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 9-3. Example of Generation of Watch Timer Interrupt Request Signal (INTWT)
(When Interrupt Cycle = 0.5 s)



CHAPTER 10 FUNCTIONS OF WATCHDOG TIMER 2

10.1 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 internal oscillation 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.

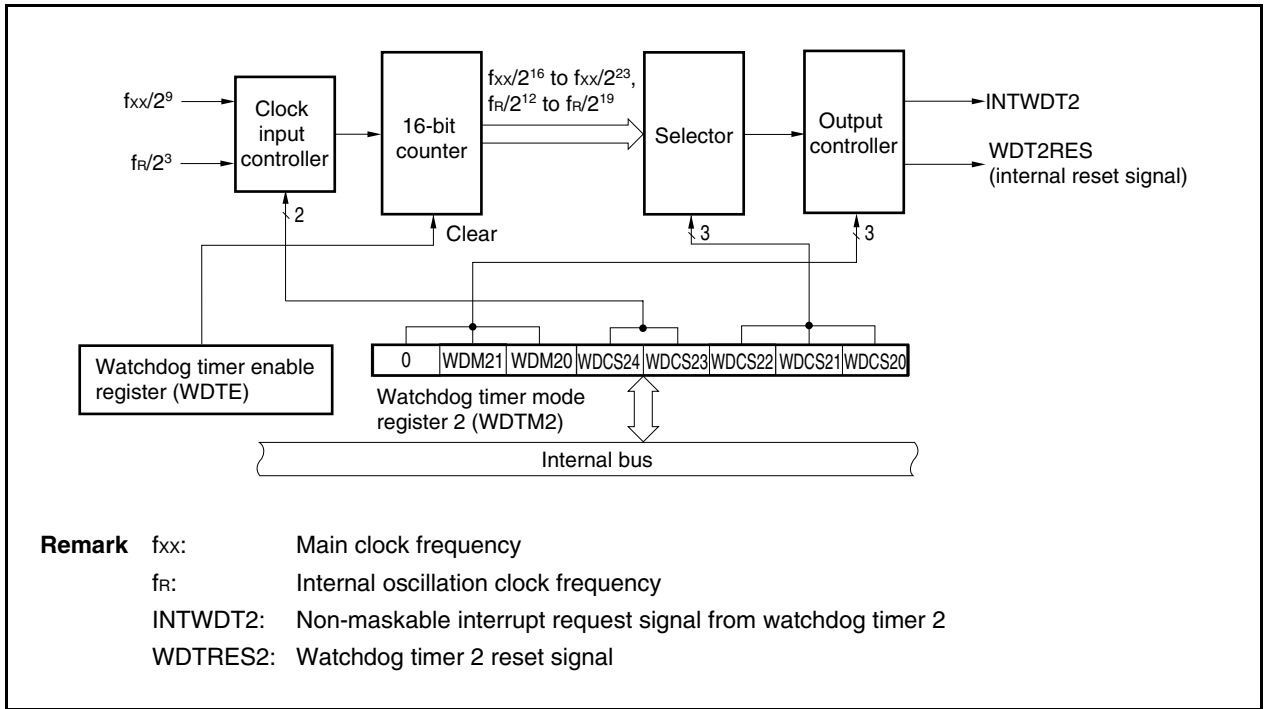
Also, write to the WDTM2 register for verification purposes only once, even if the default settings (reset mode, interval time: $f_R/2^{19}$) do not need to be changed.

2. For the non-maskable interrupt servicing due to a non-maskable interrupt request signal (INTWDT2), see **14.2.2 (2) From INTWDT2 signal**.

10.2 Configuration

The following shows the block diagram of watchdog timer 2.

Figure 10-1. Block Diagram of Watchdog Timer 2



Watchdog timer 2 includes the following hardware.

Table 10-1. Configuration of Watchdog Timer 2

| Item | Configuration |
|-------------------|---|
| Control registers | Watchdog timer mode register 2 (WDTM2) Watchdog timer enable register (WDTE) |

10.3 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 sets this register to 67H.

Caution Accessing the WDTM2 register is prohibited in the following statuses. For details, see 3.4.8

(2) Accessing specific on-chip peripheral I/O registers.

- When the CPU operates with the subclock and the main clock oscillation is stopped
- When the CPU operates with the internal oscillation clock

After reset: 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 ^{Note} |
|-------|-------|---|
| 0 | 0 | Stops operation |
| 0 | 1 | Non-maskable interrupt request mode (generation of INTWDT2 signal) |
| 1 | – | Reset mode (generation of WDT2RES signal) |

Note If the OPB1 bit is set to 1 by using the option byte function (see **CHAPTER 23**), the reset mode is fixed.

Cautions 1. For details of the WDCS20 to WDCS24 bits, see Table 10-2 Watchdog Timer 2 Clock Selection.

2. If the WDTM2 register is rewritten twice after reset, an overflow signal is forcibly generated and the counter is reset.
3. To intentionally generate an overflow signal, write to the WDTM2 register only twice or write a value other than ACH to the WDTE register once.
However, when watchdog timer 2 is set to stop operation, an overflow signal is not generated even if data is written to the WDTM2 register only twice, or a value other than “ACH” is written to the WDTE register only once.
4. To stop the operation of watchdog timer 2, set the RCM.RSTOP bit to 1 (internal oscillator is stopped), and write 1FH to the WDTM2 register. If the OPB1 bit is set to 1 by using the option byte function (see **CHAPTER 23**), however, watchdog timer 2 cannot be stopped by any means other than reset.

<R>

Table 10-2. 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 | 2,621.4 ms | 1,310.7 ms |
| | | | | | | $f_{XX} = 4 \text{ MHz}$ | $f_{XX} = 5 \text{ MHz}$ | |
| 0 | 1 | 0 | 0 | 0 | $2^{16}/f_{XX}$ | 16.4 ms | 13.1 ms | |
| 0 | 1 | 0 | 0 | 1 | $2^{17}/f_{XX}$ | 32.8 ms | 26.2 ms | |
| 0 | 1 | 0 | 1 | 0 | $2^{18}/f_{XX}$ | 65.5 ms | 52.4 ms | |
| 0 | 1 | 0 | 1 | 1 | $2^{19}/f_{XX}$ | 131.1 ms | 104.9 ms | |
| 0 | 1 | 1 | 0 | 0 | $2^{20}/f_{XX}$ | 262.1 ms | 209.7 ms | |
| 0 | 1 | 1 | 0 | 1 | $2^{21}/f_{XX}$ | 524.3 ms | 419.4 ms | |
| 0 | 1 | 1 | 1 | 0 | $2^{22}/f_{XX}$ | 1,048.6 ms | 838.9 ms | |
| 0 | 1 | 1 | 1 | 1 | $2^{23}/f_{XX}$ | 2,097.2 ms | 1,677.7 ms | |
| 1 | 1 | 1 | 1 | 1 | Operation stopped | | | |

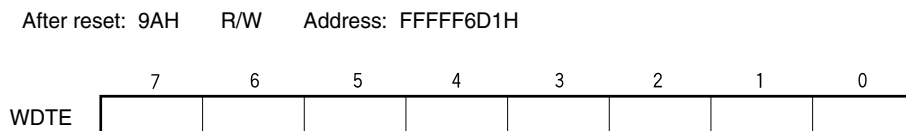
Caution If the OPB1 bit is set to 1 by using the option byte function, the clock is fixed to the internal oscillation clock (f_R) ($2^{12}/f_R$ to $2^{19}/f_R$ can be selected). For details, see CHAPTER 23 OPTION BYTE FUNCTION.

(2) Watchdog timer enable register (WDTE)

The counter of watchdog timer 2 is cleared and counting restarted by writing “ACH” to the WDTE register.

The WDTE register can be read or written in 8-bit units.

Reset sets this register to 9AH.



- 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. To intentionally generate an overflow signal, write to the WDTM2 register only twice or write a value other than ACH to the WDTE register once.
However, when the watchdog timer 2 is set to stop operation, an overflow signal is not generated even if data is written to the WDTM2 register only twice, or a value other than “ACH” is written to the WDTE register only once.
 4. The read value of the WDTE register is “9AH” (which differs from written value “ACH”).

<R>

10.4 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 WDCS24 to WDCS20 bits of the WDTM2 register 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 WDTE 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 WDM21 and WDTM2.WDM20 bits.

When the WDTM2.WDM21 bit is set to 1 (reset mode), if a WDT overflow occurs during oscillation stabilization after a reset or standby is released, no internal reset will occur and the CPU clock will switch to the internal oscillation clock.

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, see **14.2.2 (2) From INTWDT2 signal**.

CHAPTER 11 A/D CONVERTER

11.1 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
 - One-shot 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)

11.2 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.

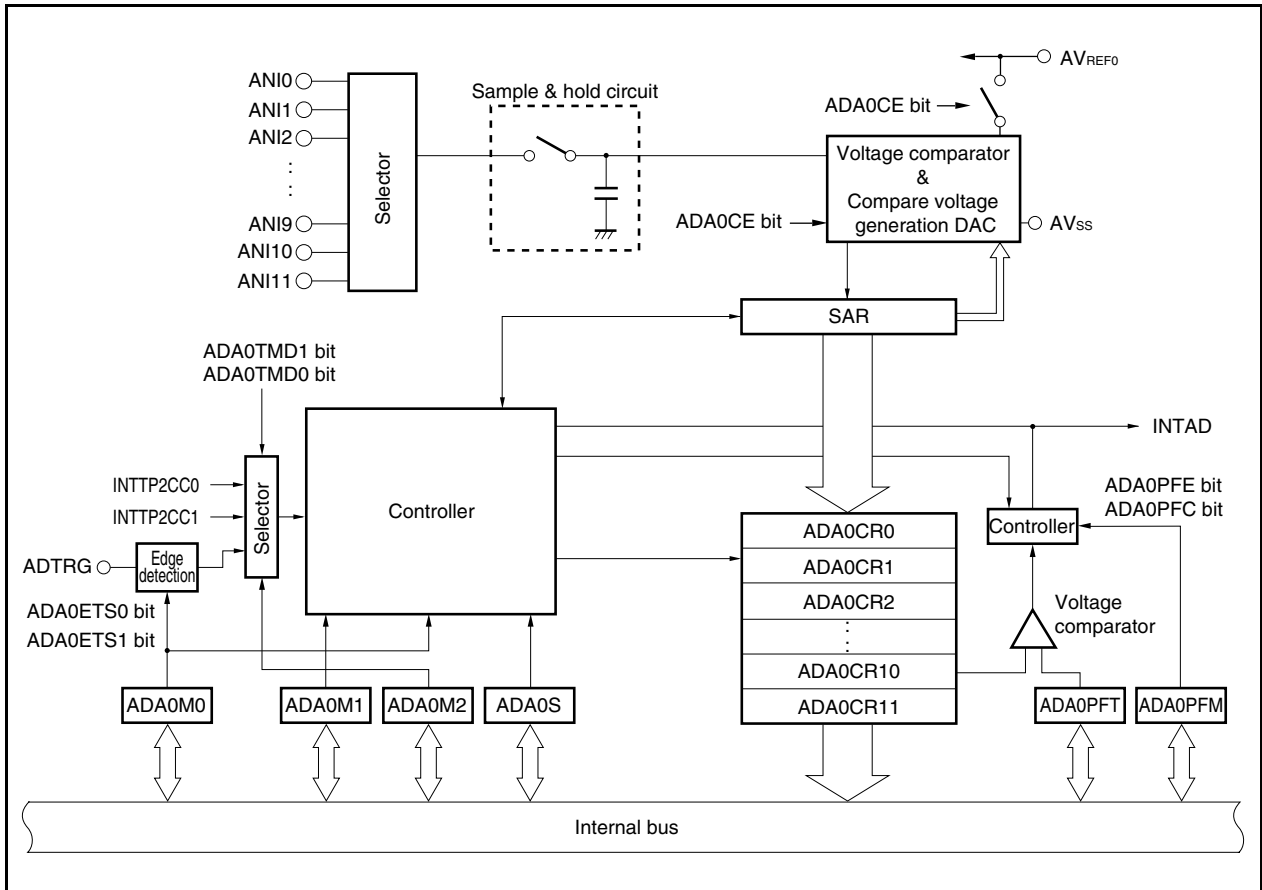
(2) Power-fail detection function

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 ($n = 0$ to 11).

11.3 Configuration

The block diagram of the A/D converter is shown below.

Figure 11-1. Block Diagram of A/D Converter



The A/D converter includes the following hardware.

Table 11-1. 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 output voltage of the compare voltage generation DAC (compare voltage), 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. ADA0ARn consist of 12 registers and the A/D conversion result is stored in the 10 higher 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 voltage value of the compare voltage generation DAC.

(12) Compare voltage generation DAC

This compare voltage generation DAC 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.

- Cautions**
- 1. 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.**
 - 2. The analog input pins (ANI0 to ANI11) function alternately as input port pins (P70 to P711). If any of ANI0 to ANI11 is selected to execute A/D conversion, do not execute an input instruction to port 7 during conversion. If executed, the conversion resolution may be degraded.**

(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.

11.4 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 sets this register to 00H.

Caution Accessing the ADA0M0 register is prohibited in the following statuses. For details, see 3.4.8

(2) Accessing specific on-chip peripheral I/O registers.

- When the CPU operates with the subclock and the main clock oscillation is stopped
- When the CPU operates with the internal oscillation clock

After reset: 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 A/D conversion | |
| 1 | Enables A/D 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 | One-shot scan mode |

| | | |
|----------|----------|--|
| 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. Write operations to bit 0 are ignored.
 2. Changing the ADA0M1 register value is prohibited while A/D conversion is enabled (ADA0CE bit = 1).
 3. If the ADA0M0, ADA0M2, ADA0S, ADA0PFM, and ADA0PFT registers are written during A/D conversion (ADA0EF bit = 1), the following will be performed according to the mode.
 - In software trigger mode
A/D conversion is stopped and started again from the beginning.
 - In hardware trigger mode
A/D conversion is stopped, and the trigger standby state is set.
 4. When not using the A/D converter, stop the operation by setting the ADA0CE bit to 0 to reduce the power consumption.
 5. The resolution for the first conversion of the data of the input pin immediately after the start of A/D conversion may be degraded. For details, see 11.6 (7) AV_{REF0} pin.

(2) A/D converter mode register 1 (ADA0M1)

The ADA0M1 register is an 8-bit register that controls the conversion time specification.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF201H

| | | | | | | | | |
|--------|---------|---|---|---|---------|---------|---------|---------|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| ADA0M1 | ADA0HS1 | 0 | 0 | 0 | ADA0FR3 | ADA0FR2 | ADA0FR1 | ADA0FR0 |

Cautions

1. Be sure to clear bits 6 to 4 to “0”.
2. Be sure to set the ADA0HS1 bit to “1”.

Remark For A/D conversion time setting examples, see **Table 11-2**.

Table 11-2. Conversion Mode Setting Example

| ADA0HS1 | ADA0FR3 to ADA0FR0 | | | | A/D Conversion Time | f _{xx} = 20 MHz | f _{xx} = 16 MHz | f _{xx} = 4 MHz | A/D Stabilization Time ^{Note} |
|---------|--------------------|---|---|---|---------------------|--------------------------|--------------------------|-------------------------|--|
| | 3 | 2 | 1 | 0 | | | | | |
| 1 | 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} |

Note When the ADA0CE bit of the ADA0M0 register is changed from 0 to 1 to secure the A/D converter stabilization time, the first A/D conversion starts after one of the above clock values is input.

- <R> **Cautions**
1. Set as $3.1 \mu\text{s} \leq \text{conversion time} \leq 15.5 \mu\text{s}$.
 2. Rewriting of the ADA0M0, ADA0M2, ADA0S, ADA0PFM, and ADA0PFT registers and trigger input are prohibited during the stabilization time.

(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 sets 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 sets 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 |
| Other than above | | | | Setting prohibited | |

(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.

Caution Accessing the ADA0CRn and ADA0CRnH registers is prohibited in the following statuses. For details, see 3.4.8 (2) Accessing specific on-chip peripheral I/O registers.

- When the CPU operates with the subclock and the main clock oscillation is stopped
- When the CPU operates with the internal oscillation clock

After reset: 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 | 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 | AD9 | AD8 | AD7 | AD6 | AD5 | AD4 | AD3 | AD2 |

Remark n = 0 to 11

Caution 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.

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

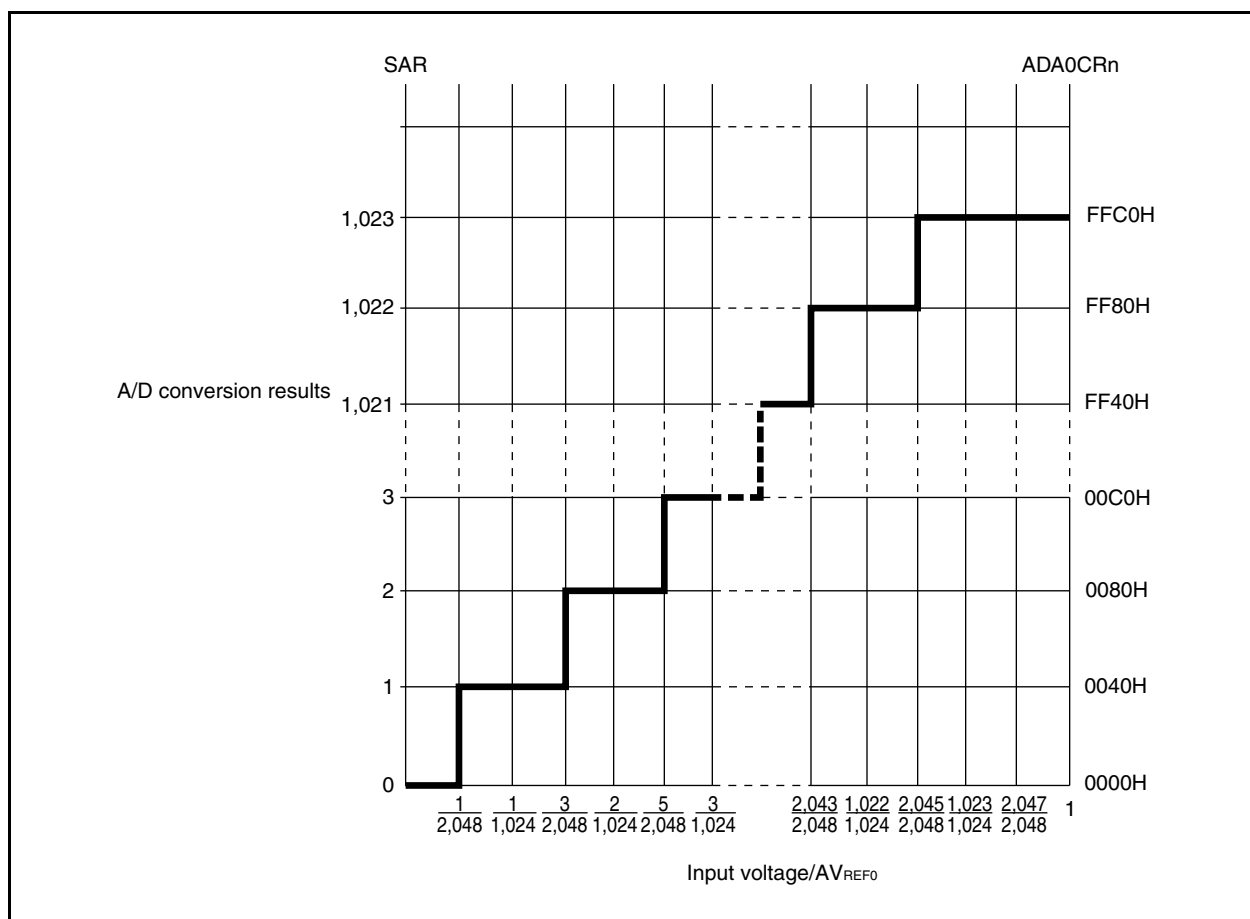
AV_{REF0}: AV_{REF0} pin voltage

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 11-2. 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 sets 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 $ADA0CRnH \geq ADA0PFT$ |
| 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)

<R>

The ADA0PFT register sets the threshold value when comparing with the A/D conversion result register nH (ADA0CRnH).

The 8-bit data set in the ADA0PFT register is compared with the value of the ADA0CRnH register.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF205H

| | | | | | | | | |
|---------|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADA0PFT | | | | | | | | |

11.5 Operation

11.5.1 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) to set the compare voltage generation DAC to $(1/2) AV_{REF0}$.
- <5> The voltage difference between the compare voltage generation DAC 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 compare voltage generation DAC is selected as follows.
- Bit 9 = 1: $(3/4) AV_{REF0}$
 - Bit 9 = 0: $(1/4) AV_{REF0}$
- This compare voltage and the analog input voltage are compared and, depending on the result, bit 8 is manipulated as follows.
- Analog input voltage \geq Compare voltage: Bit 8 = 1
Analog input voltage \leq Compare voltage: 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. After that, an A/D conversion end interrupt request signal (INTAD) is generated.

11.5.2 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 pin (ANI0 to ANI11) 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/scan mode, the next conversion is started, unless the ADA0CE bit is cleared to 0 after completion of the first conversion. Conversion is performed once and ends if the operation mode is the one-shot select/scan mode.

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 pin (ANI0 to ANI11) 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.ATA0ETS0 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, continuous scan, or one-shot 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 the conversion operation, the conversion is not 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 pin (ANI0 to ANI11) 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, continuous scan, or one-shot 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 stopped and the A/D converter waits for the trigger again.

11.5.3 Operation mode

Three operation modes are available as the modes in which to set the ANI0 to ANI11 pins: continuous select mode, continuous scan mode, and one-shot scan mode.

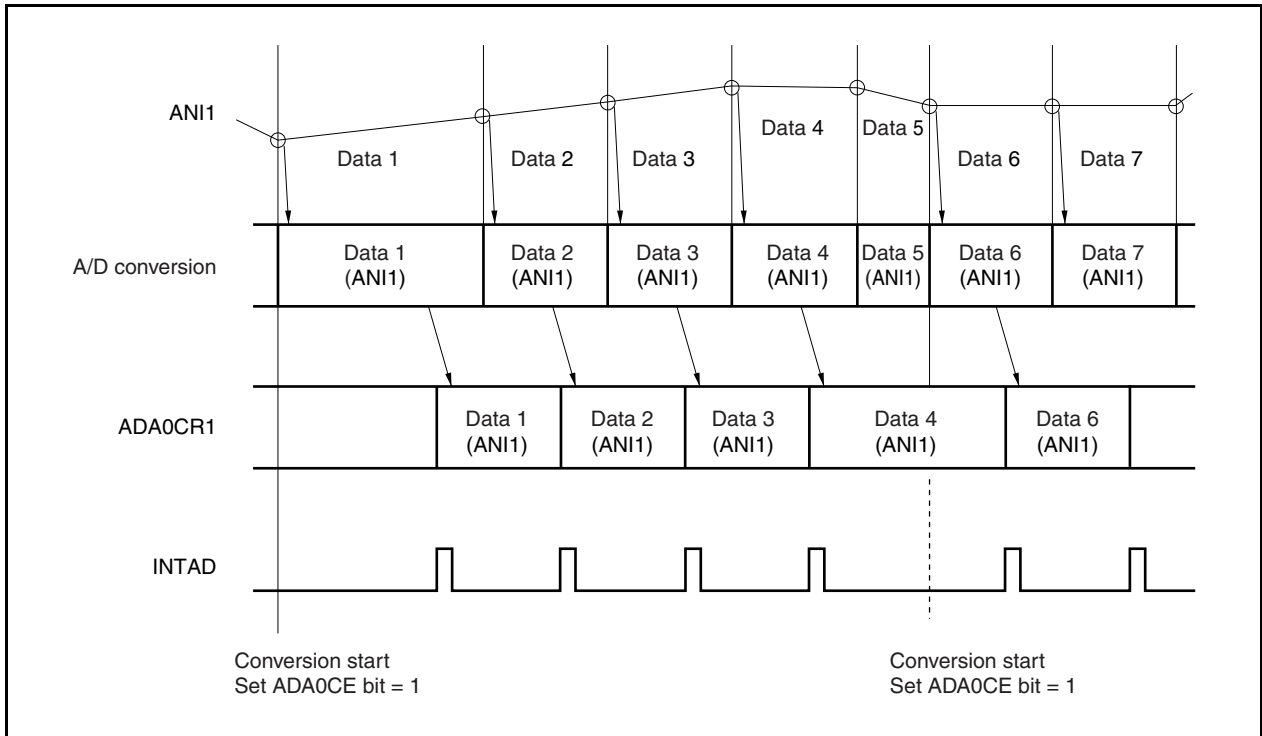
The operation mode is selected by the ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 bits.

(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 11-3. Timing Example of Continuous Select Mode Operation (ADA0S Register = 01H)



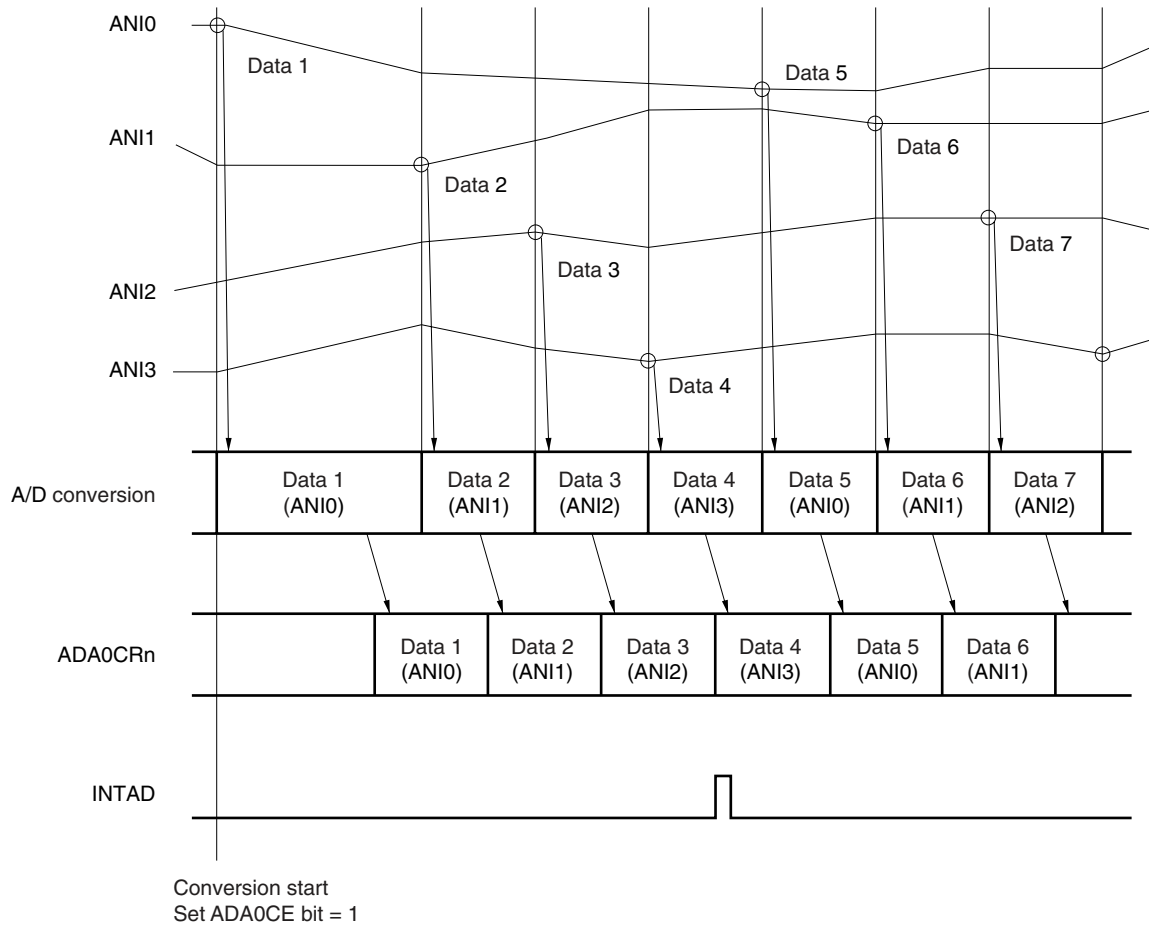
(2) Continuous scan mode

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.

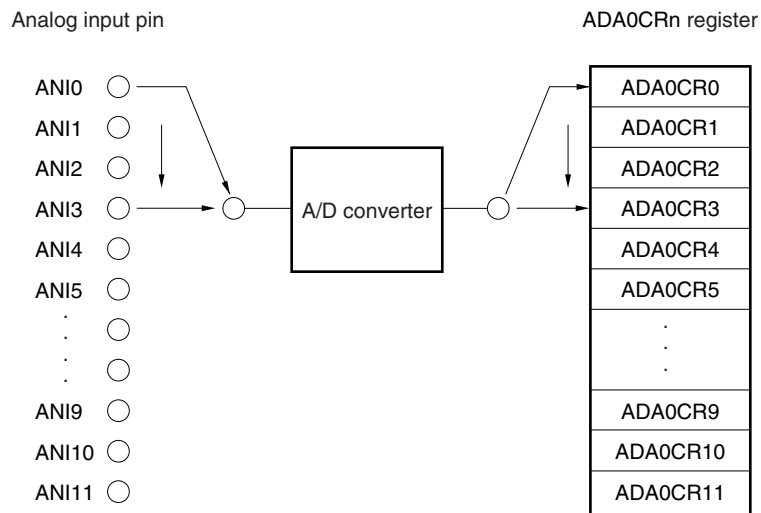
The result of each conversion is stored in the ADA0CRn register corresponding to the analog input pin. When conversion of the analog input pin specified by the ADA0S register is complete, the INTAD signal is generated, and A/D conversion is started again from the ANI0 pin, unless the ADA0CE bit is cleared to 0 ($n = 0$ to 11).

Figure 11-4. Timing Example of Continuous Scan Mode Operation (ADA0S Register = 03H)

(a) Timing example



(b) Block diagram



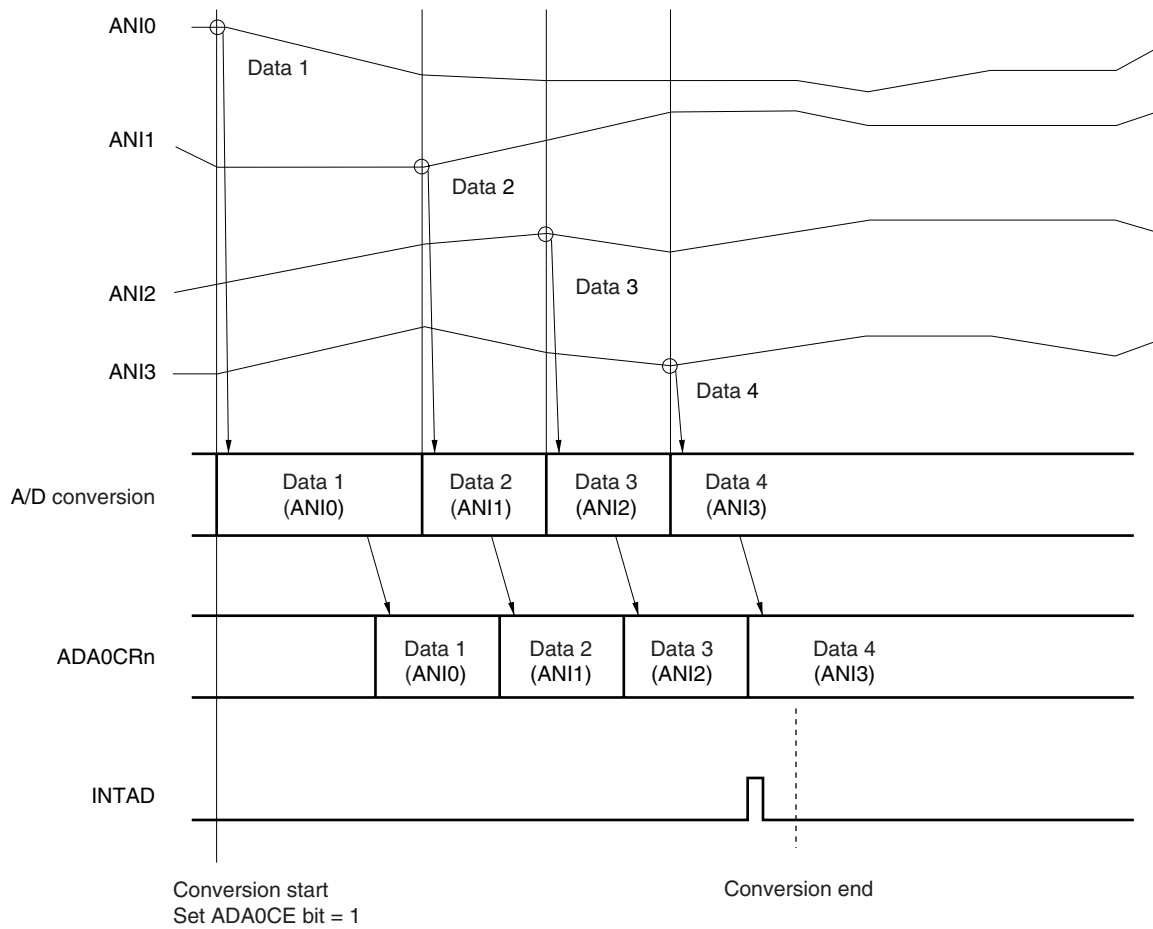
(3) One-shot scan mode

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.

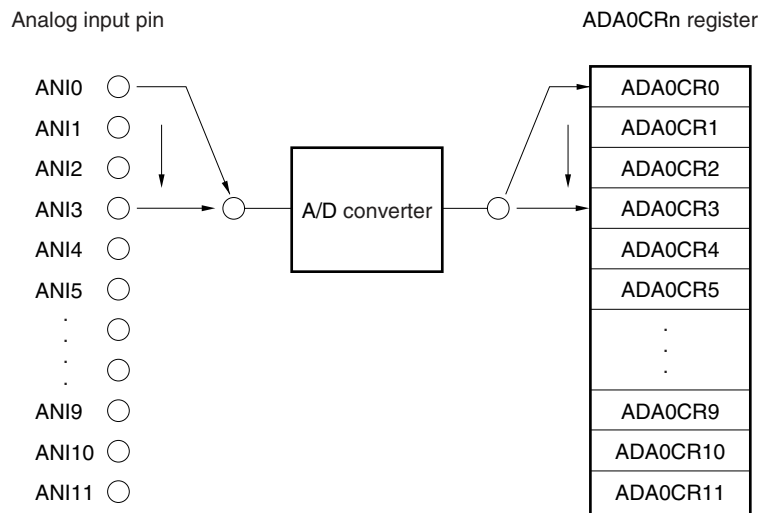
Each conversion result is stored in the ADA0CRn register corresponding to the analog input pin. When conversion of the analog input pin specified by the ADA0S register is complete, the INTAD signal is generated. A/D conversion is stopped after it has been completed (n = 0 to 11).

Figure 11-5. Timing Example of One-Shot Scan Mode Operation (ADA0S Register = 03H)

(a) Timing example



(b) Block diagram



11.5.4 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 $ADA0CRnH \geq ADA0PFT$.
- 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 $ADA0CRnH < ADA0PFT$.

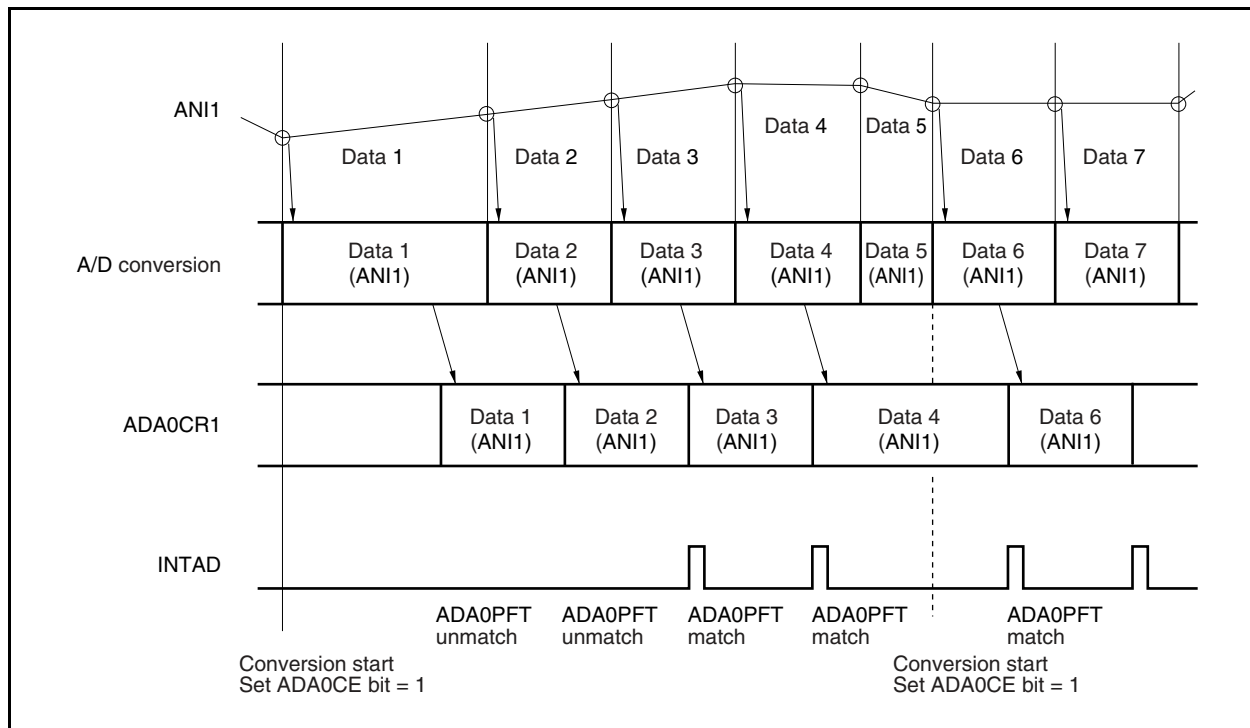
Remark n = 0 to 11

In the power-fail compare mode, three modes are available as modes in which to set the ANI0 to ANI11 pins: continuous select mode, continuous scan mode, and one-shot scan mode.

(1) Continuous select mode

In this mode, the result of converting the voltage of the 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).

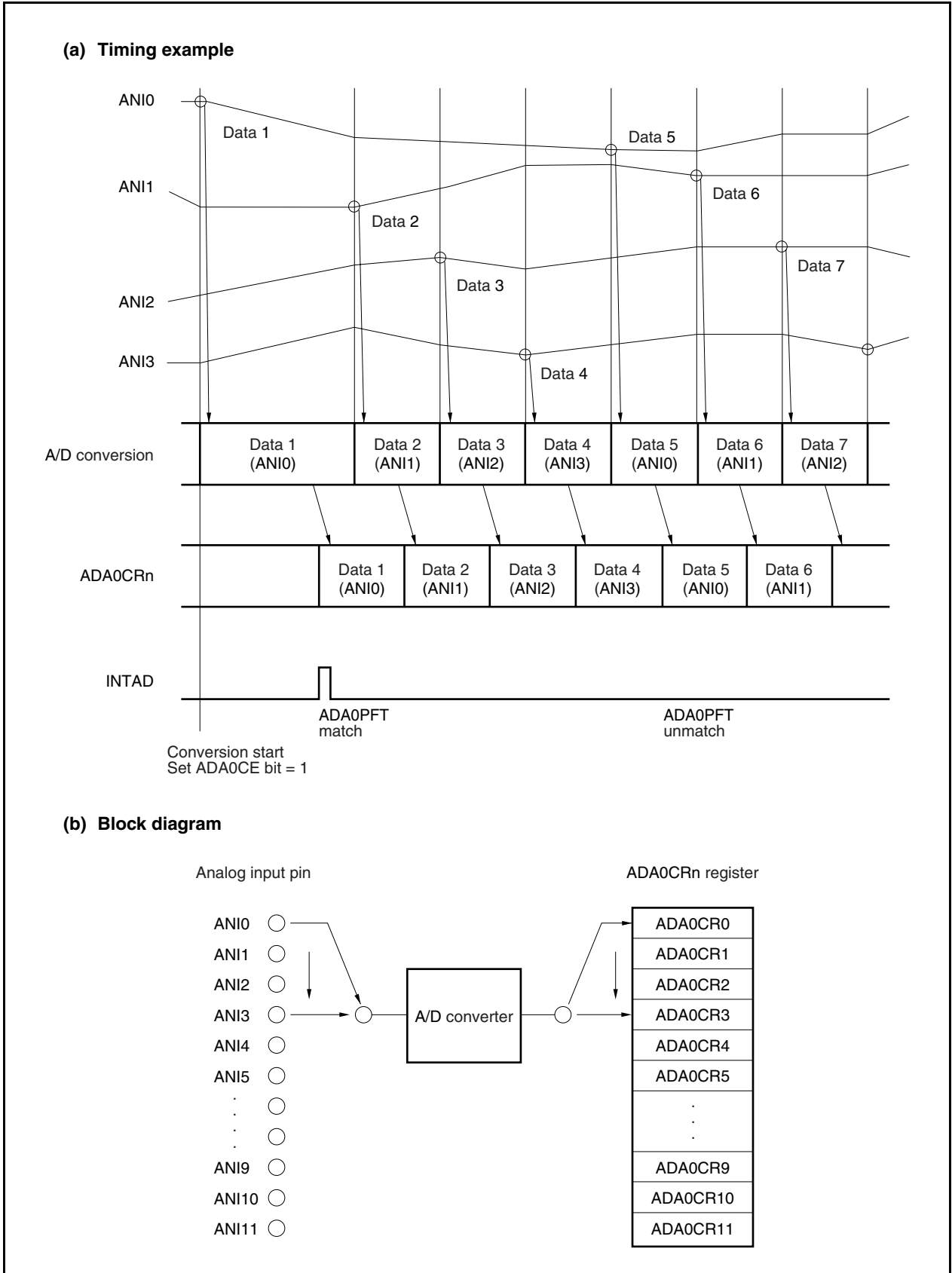
Figure 11-6. Timing Example of Continuous Select Mode Operation
(When Power-Fail Comparison Is Made: ADA0S Register = 01H)

**(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.

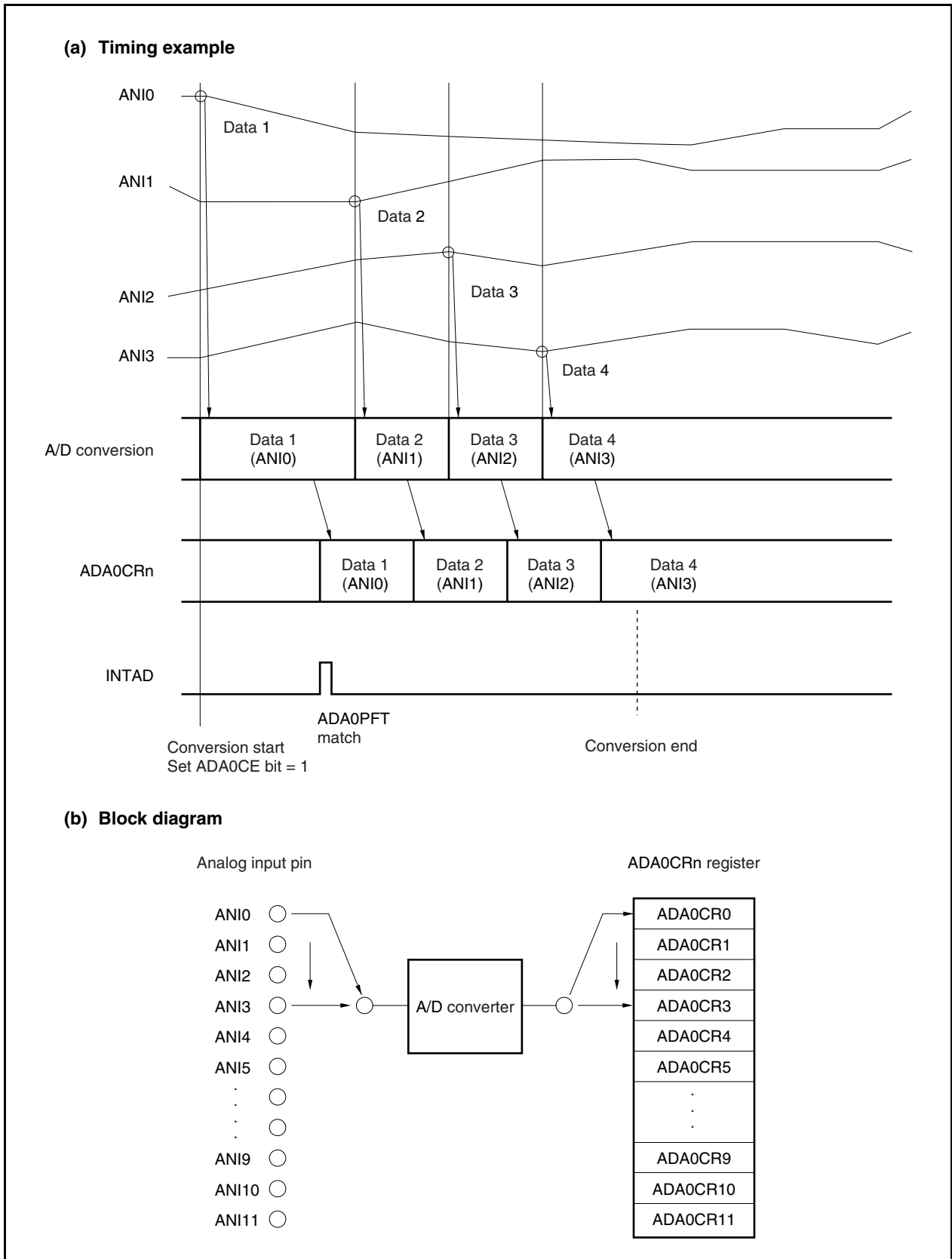
**Figure 11-7. Timing Example of Continuous Scan Mode Operation
(When Power-Fail Comparison Is Made: ADA0S Register = 03H)**



(3) One-shot scan mode

In this mode, the results of converting the voltages of the analog input pins sequentially selected from the ANIO 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 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 ADA0CR0 register and the INTAD signal is generated. If it does not match, the conversion result is stored in the ADA0CR0 register, and the INTAD0 signal is not generated. After the result of the first conversion has been stored in the ADA0CR0 register, the results of converting the signals on the analog input pins specified by the ADA0S register are sequentially stored. The conversion is stopped after it has been completed.

**Figure 11-8. Timing Example of One-Shot Scan Mode Operation
(When Power-Fail Comparison Is Made: ADA0S Register = 03H)**



11.6 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.

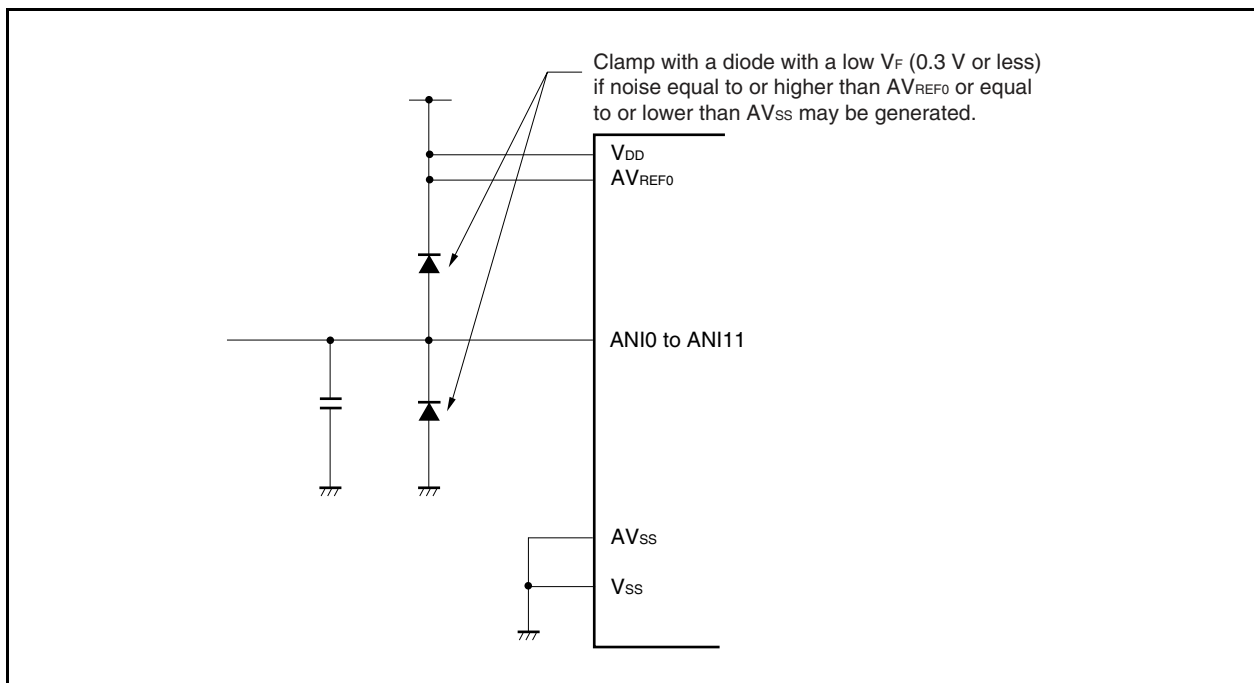
(2) Input range of ANI0 to ANI11 pins

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 11-9 is recommended.

Figure 11-9. 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.

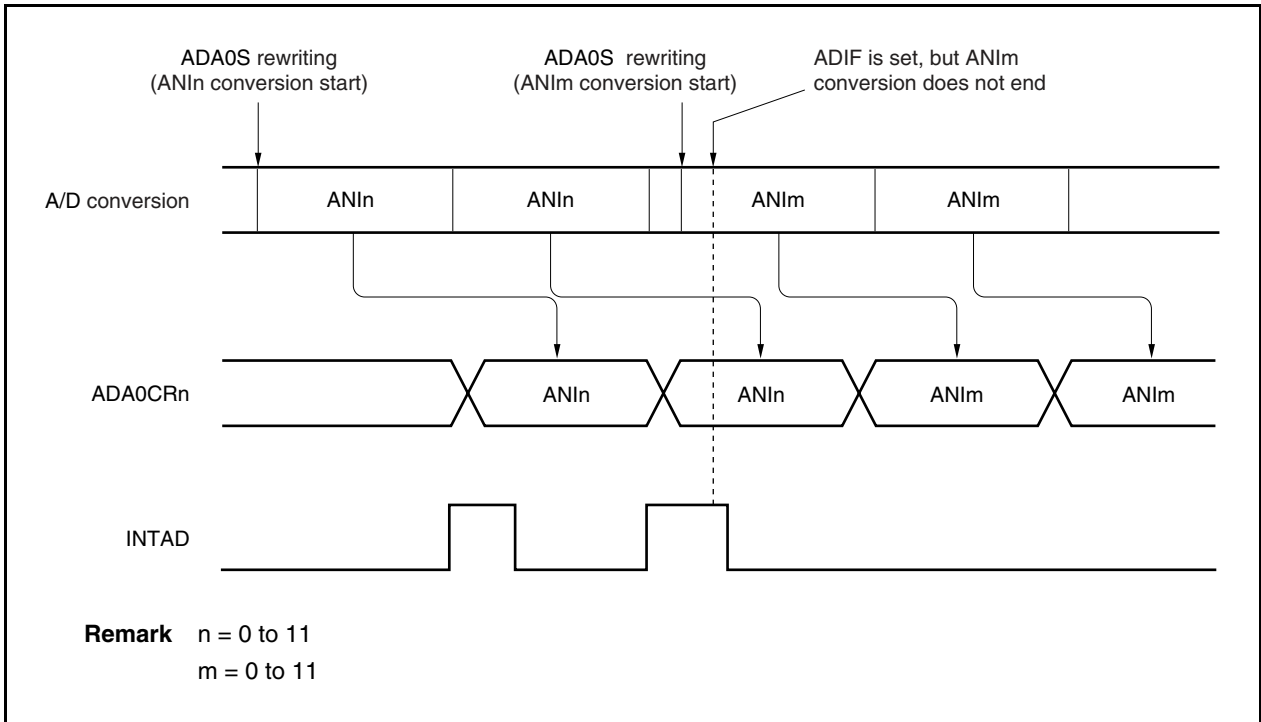
Also the conversion resolution may drop at the pins set as output port pins during A/D conversion if the current flows due to the effect of the external circuit connected to the port pins.

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.

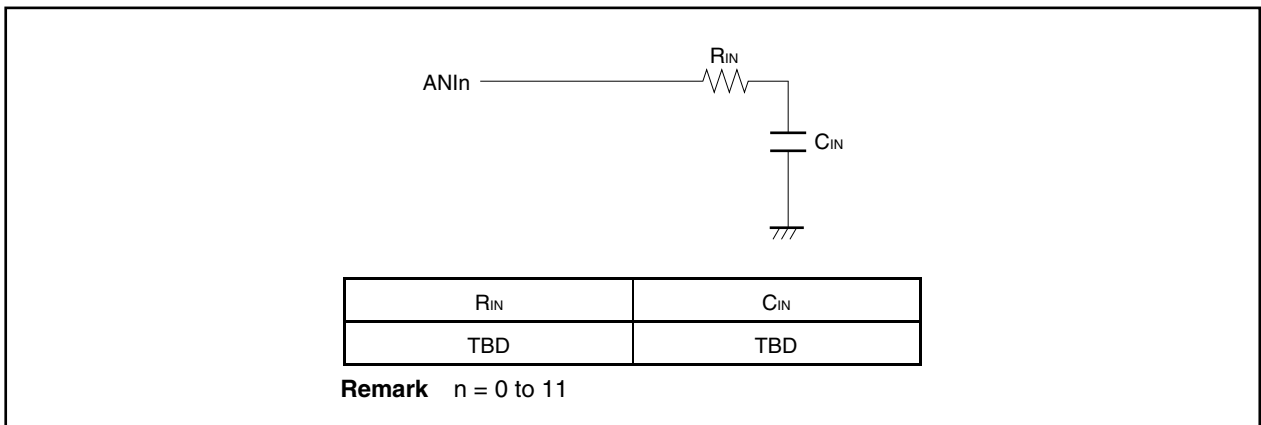
Figure 11-10. Generation Timing of A/D Conversion End Interrupt Request



(6) Internal equivalent circuit

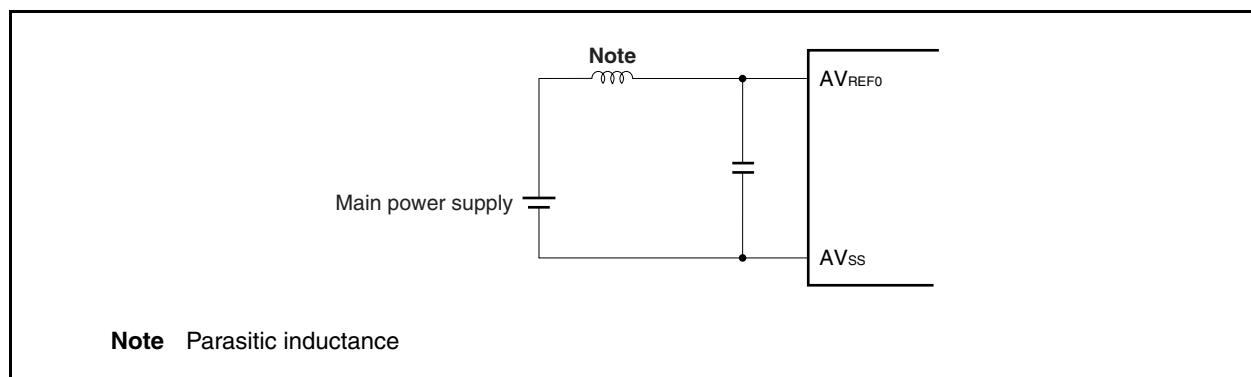
The following shows the equivalent circuit of the analog input block.

Figure 11-11. Internal Equivalent Circuit of ANIn Pin



(7) 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 11-12.
- (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 to suppress the reference voltage fluctuation as shown in Figure 11-12.
- (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 11-12. AV_{REF0} Pin Processing Example**<R> (8) Reading ADA0CRn result**

When the ADA0M0 to ADA0M2, ADA0S, ADA0PFM, or ADA0PFT 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, ADA0S, ADA0PFM, or ADA0PFT registers. Also, when an external/timer trigger is acknowledged, the contents of the ADA0CRn register may be undefined. Read the conversion result after completion of conversion and before the next external/timer trigger is acknowledged. The correct conversion result may not be read at a timing different from the above.

(9) A/D conversion result

If there is noise at the analog input pins and at the reference voltage input pins, that noise may generate an illegal conversion result. Software processing will be needed to avoid a negative effect on the system from this illegal conversion result. An example of this software processing is shown below.

- Take the average result of a number of A/D conversions and use that as the A/D conversion result.
- Execute a number of A/D conversions consecutively and use those results, omitting any exceptional results that may have been obtained.
- If an A/D conversion result that is judged to have generated a system malfunction is obtained, be sure to recheck the system malfunction before performing malfunction processing.

<R>

(10) Standby mode

Because the A/D converter stops operating in the STOP mode, conversion results are invalid, so power consumption can be reduced. Operations are resumed after the STOP mode is released, but the A/D conversion results after the STOP mode is released are invalid. When using the A/D converter after the STOP mode is released, before setting the STOP mode or releasing the STOP mode, clear the ADA0M0.ADA0CE bit to 0 then set the ADA0CE bit to 1 after releasing the STOP mode.

In the IDLE1, IDLE2, or subclock operation mode, operation continues. To lower the power consumption, therefore, clear the ADA0M0.ADA0CE bit to 0. In the IDLE1 and IDLE2 modes, since the analog input voltage value cannot be retained, the A/D conversion results after the IDLE1 and IDLE2 modes are released are invalid. The results of conversions before the IDLE1 and IDLE2 modes were set are valid.

<R>

(11) Rewriting registers and trigger input during the stabilization time

Rewriting of the ADA0M0, ADA0M2, ADA0S, ADA0PFM, and ADA0PFT registers and trigger input during the stabilization time are prohibited.

(12) Variation of A/D conversion results

The results of the A/D conversion may vary depending on the fluctuation of the supply voltage, or may be affected by noise. To reduce the variation, take counteractive measures with the program such as averaging the A/D conversion results.

(13) A/D conversion result hysteresis characteristics

The successive comparison type A/D converter holds the analog input voltage in the internal sample & hold capacitor and then performs A/D conversion. After the A/D conversion has finished, the analog input voltage remains in the internal sample & hold capacitor. As a result, the following phenomena may occur.

- When the same channel is used for A/D conversions, if the voltage is higher or lower than the previous A/D conversion, then hysteresis characteristics may appear where the conversion result is affected by the previous value. Thus, even if the conversion is performed at the same potential, the result may vary.
- When switching the analog input channel, hysteresis characteristics may appear where the conversion result is affected by the previous channel value. This is because one A/D converter is used for the A/D conversions. Thus, even if the conversion is performed at the same potential, the result may vary.

<R>

Therefore, to obtain more accurate conversion result, perform A/D conversion twice successively for the same channel, and discard the first conversion result.

11.7 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.

$$\begin{aligned} 1\%FSR &= (\text{Maximum value of convertible analog input voltage} - \text{Minimum value of convertible analog input voltage})/100 \\ &= (AV_{REF0} - 0)/100 \\ &= AV_{REF0}/100 \end{aligned}$$

When the resolution is 10 bits, 1 LSB is as follows:

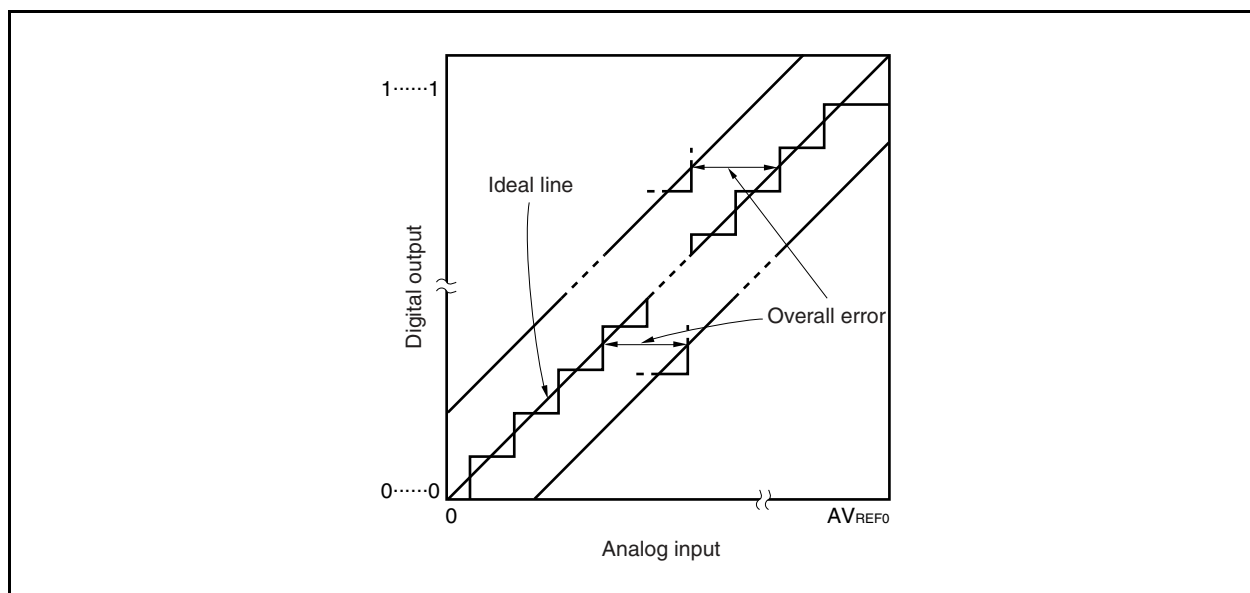
$$\begin{aligned} 1 \text{ LSB} &= 1/2^{10} = 1/1,024 \\ &= 0.098\%FSR \end{aligned}$$

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 11-13. Overall Error

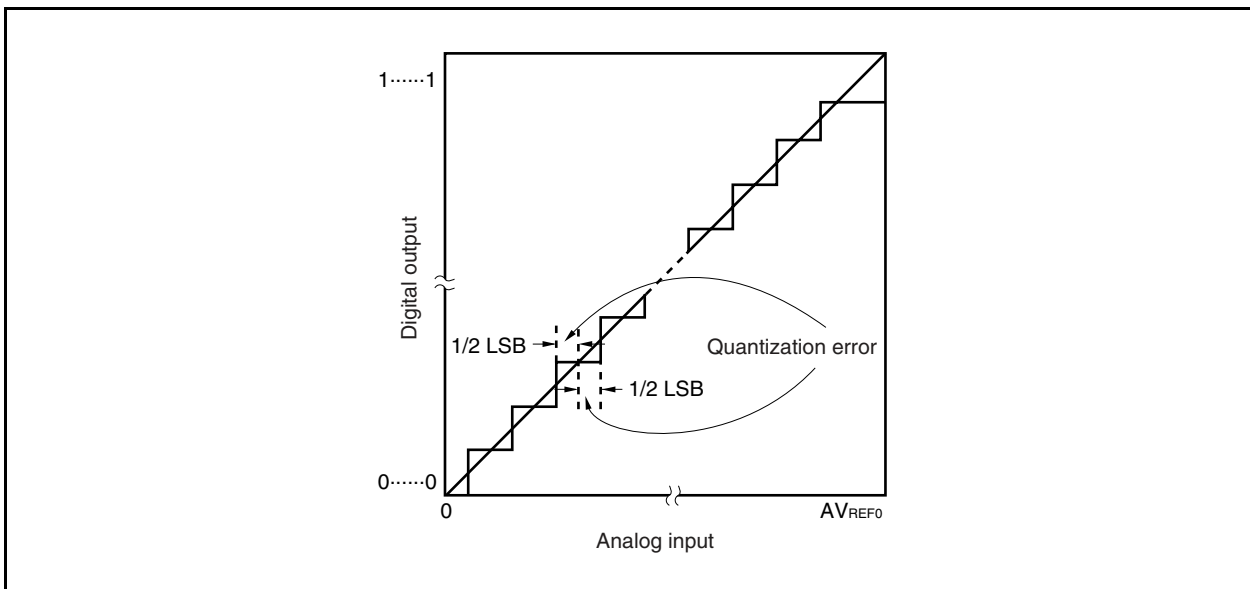


(3) Quantization error

This is an error of $\pm 1/2$ 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 $\pm 1/2$ 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.

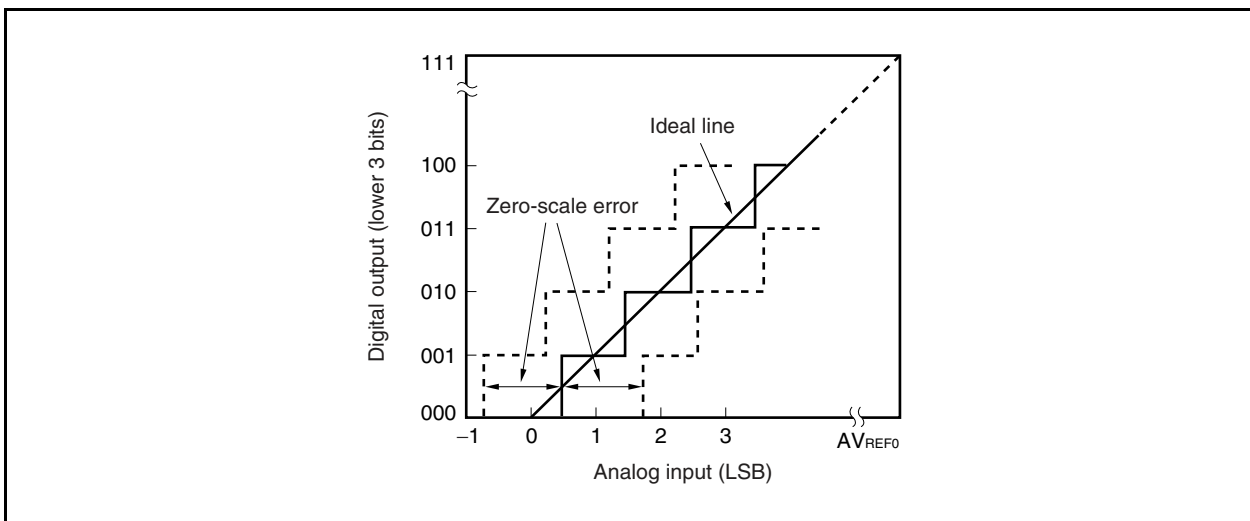
Figure 11-14. Quantization Error



(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).

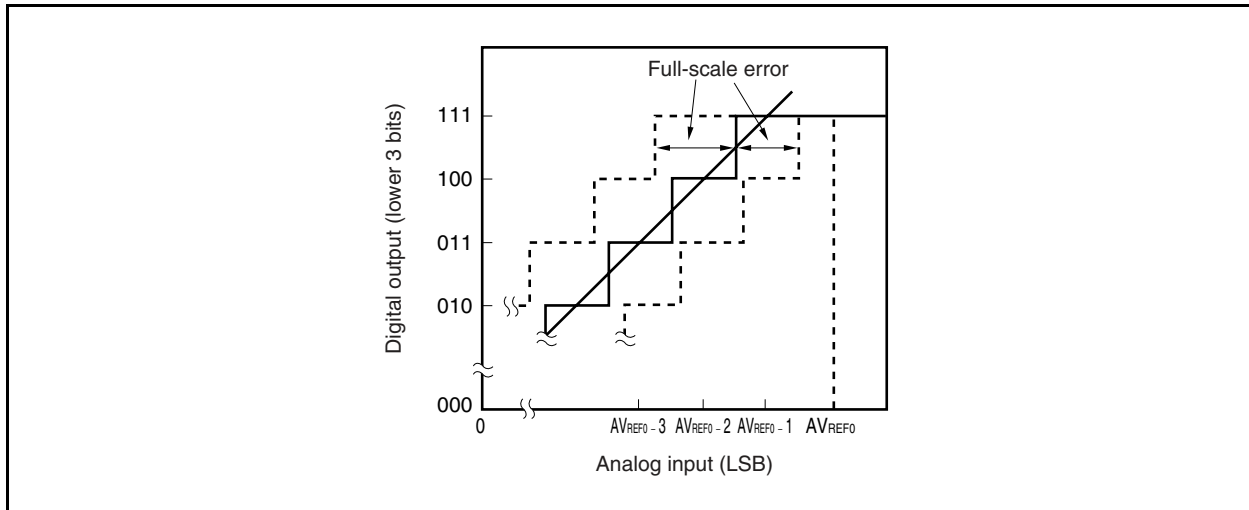
Figure 11-15. Zero-Scale Error



(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).

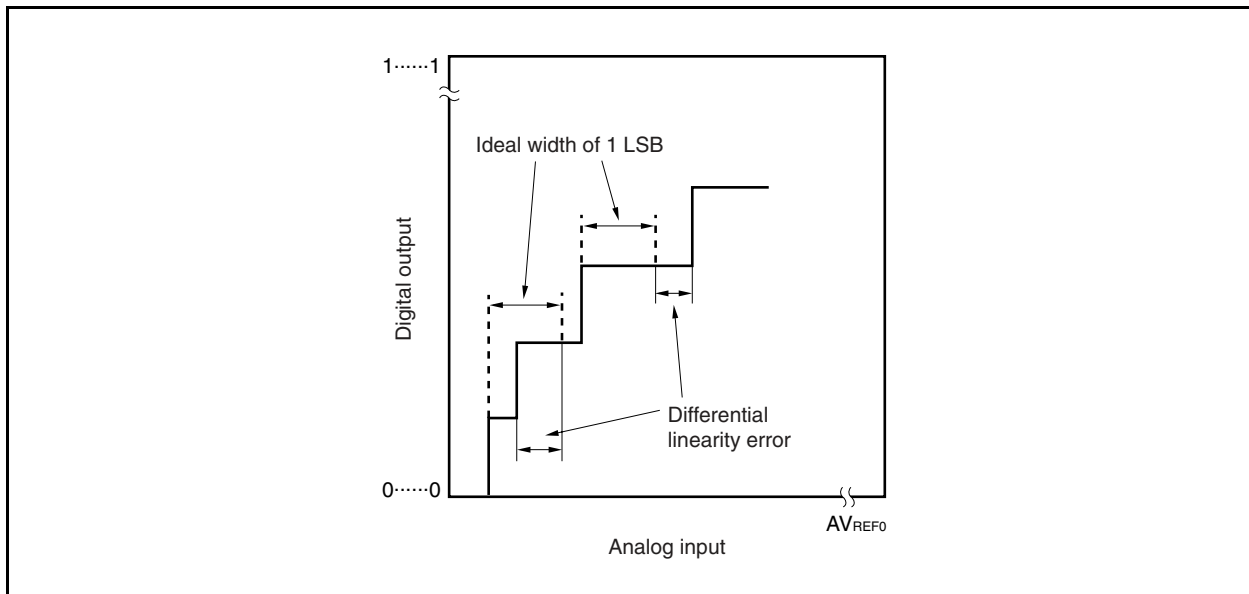
Figure 11-16. Full-Scale Error



(6) Differential linearity error

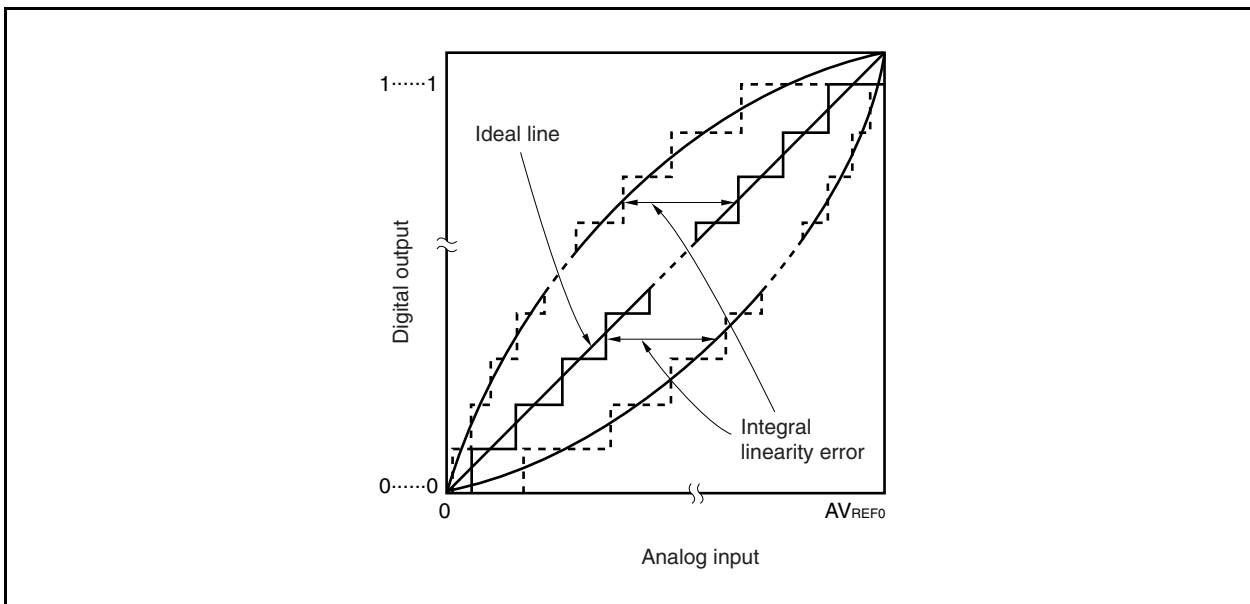
Ideally, the width to output a specific code is 1 LSB. This error indicates the difference between the actually measured value and its theoretical value when a specific code is output. This indicates the basic characteristics of the A/D conversion when the voltage applied to the analog input pins of the same channel is consistently increased bit by bit from AV_{SS} to AV_{REF0} . When the input voltage is increased or decreased, or when two or more channels are used, see 11.7 (2) Overall error.

Figure 11-17. Differential Linearity Error



(7) Integral linearity error

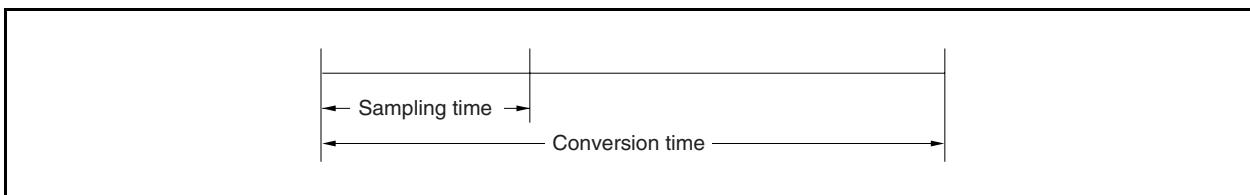
This error indicates the extent to which the conversion characteristics differ from the ideal linear relationship. It indicates the maximum value of the difference between the actually measured value and its theoretical value where the zero-scale error and full-scale error are 0.

Figure 11-18. Integral Linearity Error**(8) Conversion time**

This is the time required to obtain a digital output after each trigger has been generated. 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 11-19. Sampling Time

CHAPTER 12 ASYNCHRONOUS SERIAL INTERFACE A (UARTA)

The V850ES/HF2 includes two channels of asynchronous serial interface A (UARTA).

12.1 Features

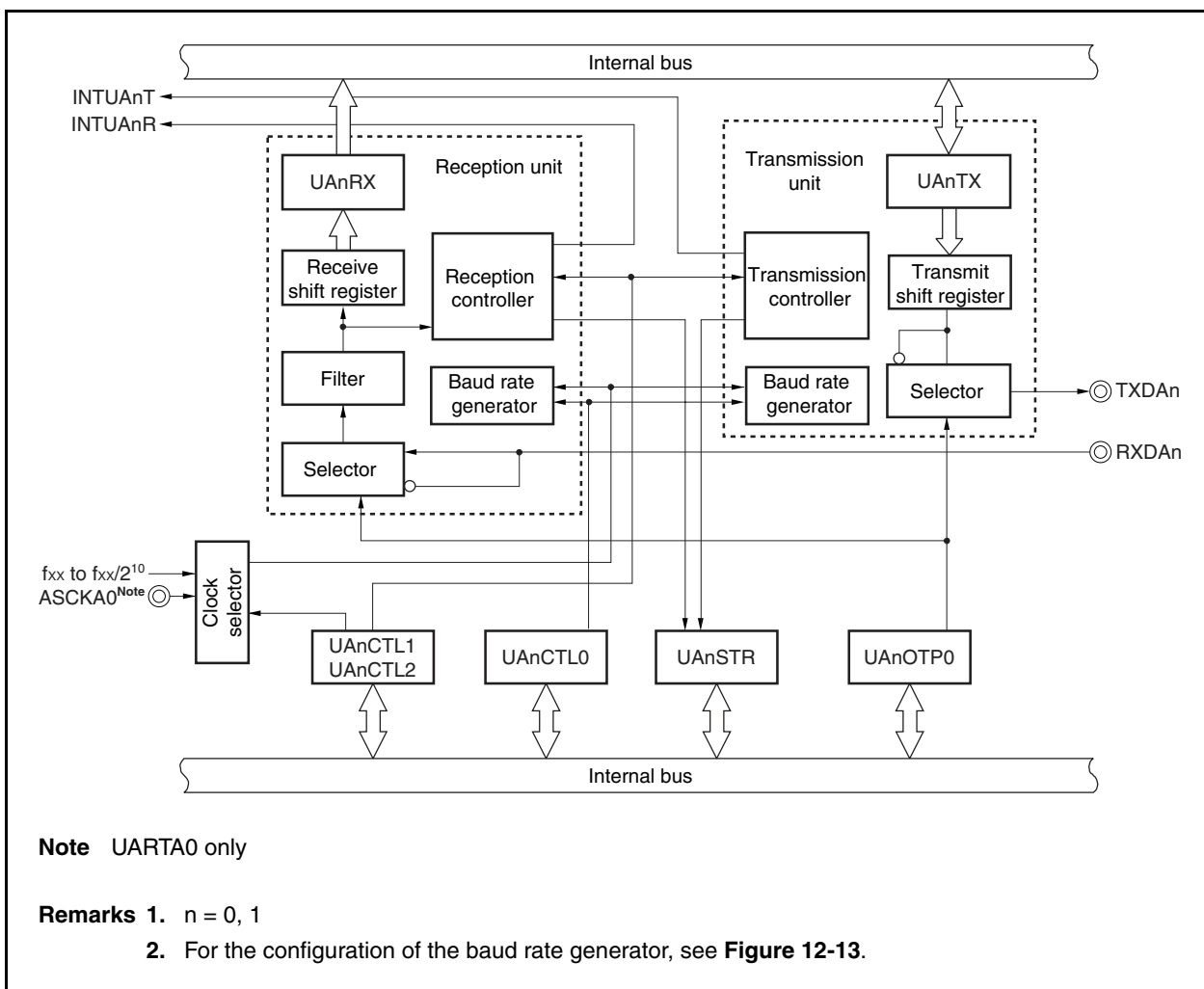
- Transfer rate: 300 bps to 312.5 kbps (using internal system clock of 20 MHz and dedicated baud rate generator)
- Full-duplex communication: Internal UARTAn receive data register (UAnRX)
Internal UARTAn transmit data register (UAnTX)
- 2-pin configuration: TXDAn: Transmit data output pin
RXDAn: Receive data input pin
- Reception error output function
 - Parity error
 - Framing error
 - Overrun error
- Interrupt sources: 2
 - Reception complete interrupt (INTUAnR): An interrupt is generated in the reception enabled status by ORing three types of reception errors. It is also generated when receive data is transferred from the receive shift register to the receive data register after completion of serial transfer.
 - Transmission enable interrupt (INTUAnT): This interrupt occurs upon transfer of transmit data from the transmit data register to the transmit 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
- SBF (Sync Break Field) transmission/reception in the LIN (Local Interconnect Network) communication format possible
 - 13 to 20 bits selectable for SBF transmission
 - Recognition of 11 bits or more possible for SBF reception
 - SBF reception flag provided

Remark n = 0, 1

12.2 Configuration

The block diagram of the UARTAn is shown below.

Figure 12-1. Block Diagram of Asynchronous Serial Interface An



UARTAn includes the following hardware units.

Table 12-1. 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 the UARTAn.

(3) UARTAn control register 2 (UAnCTL2)

The UAnCTL2 register is an 8-bit register used to control the baud rate for the UARTAn.

(4) UARTAn option control register 0 (UAnOPT0)

The UAnOPT0 register is an 8-bit register used to control serial transfer for the UARTAn.

(5) UARTAn status register (UAnSTR)

The UAnSTRn 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 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.

12.3 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 sets this register to 10H.

(1/2)

| After reset: 10H | | R/W | Address: UA0CTL0 FFFFFFFA00H, UA1CTL0 FFFFFFFA10H | | | | | | | |
|--|--|--------|---|--------|--------|--------|-------|-------|---|---|
| | | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UAnCTL0 | UAnPWR | UAnTXE | UAnRXE | UAnDIR | UAnPS1 | UAnPS0 | UAnCL | UAnSL | | |
| (n = 0, 1) | | | | | | | | | | |
| 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 | | | | | | | | | |
| <ul style="list-style-type: none"> 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, see 12.6 (1) (a) Base clock). | | | | | | | | | | |
| UAnRXE | Reception operation enable | | | | | | | | | |
| 0 | Disable reception operation | | | | | | | | | |
| 1 | Enable reception operation | | | | | | | | | |
| <ul style="list-style-type: none"> 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, see 12.6 (1) (a) Base clock). | | | | | | | | | | |

<R>

| 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.
- When transmission and reception are performed in the LIN format, set the UAnDIR bit to 1.

| 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 is 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, the UAnSTR.UAnPE bit is not set.
- When transmission and reception are performed in the LIN format, clear the UAnPS1 and UAnPS0 bits to 00.

<R>

| 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.
- When transmission and reception are performed in the LIN format, set the UAnCL bit to 1.

| 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 12.5.9 Parity types and operations.

(2) UARTAn control register 1 (UAnCTL1)

For details, see 12.6 (2) **UARTAn control register 1 (UAnCTL1)**.

(3) UARTAn control register 2 (UAnCTL2)

For details, see 12.6 (3) **UARTAn control register 2 (UAnCTL2)**.

(4) UARTAn option control register 0 (UAnOPT0)

The UAnOPT0 register is an 8-bit register that controls the serial transfer operation of the UARTAn register.

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

Reset sets this register to 14H.

(1/2)

After reset: 14H R/W Address: UA0OPT0 FFFFFFFA03H, UA1OPT0 FFFFFFFA13H

| | | | | | | | | |
|---------|--------|--------|--------|---------|---------|---------|--------|--------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UAnOPT0 | UAnSRF | UAnSRT | UAnSTT | UAnSLS2 | UAnSLS1 | UAnSLS0 | UAnTDL | UAnRDL |

(n = 0, 1)

| | |
|--|--|
| 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 |
| <ul style="list-style-type: none"> • SBF (Sync Break 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. • The UAnSRF bit is a read-only bit. | |

| | |
|---|-----------------------|
| UAnSRT | SBF reception trigger |
| 0 | – |
| 1 | SBF reception trigger |
| <ul style="list-style-type: none"> • 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 |
| <ul style="list-style-type: none"> • 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. | |

Caution Do not set the UAnSRT and UAnSTT bits (to 1) during SBF reception (UAnSRF bit = 1).

<R>

| UAnSLS2 | UAnSLS1 | UAnSLS0 | SBF transmit length selection |
|---------|---------|---------|-------------------------------|
| 1 | 0 | 1 | 13-bit output (reset value) |
| 1 | 1 | 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 | 0 | 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 input 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 | <ul style="list-style-type: none"> • Reset • UAnCTL0.UAnPWR = 0 |
| UAnTSF bit | <ul style="list-style-type: none"> • UAnCTL0.UAnTXE = 0 |
| UAnPE, UAnFE, UAnOVE bits | <ul style="list-style-type: none"> • 0 write • UAnCTL0.UAnRXE = 0 |

After reset: 00H R/W Address: UA0STR FFFFFFFA04H, UA1STR FFFFFFFA14H

| | | | | | | | | |
|----------------------|--------|---|---|---|---|-------|-------|--------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UAnSTR (n = 0, 1) | UAnTSF | 0 | 0 | 0 | 0 | UAnPE | UAnFE | UAnOVE |

| | |
|--|---|
| UAnTSF | Transfer status flag |
| 0 | <ul style="list-style-type: none"> 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 |
| <p>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.</p> | |

| | |
|---|--|
| UAnPE | Parity error flag |
| 0 | <ul style="list-style-type: none"> 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. |
| <ul style="list-style-type: none"> 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 | <ul style="list-style-type: none"> 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 |
| <ul style="list-style-type: none"> 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 | <ul style="list-style-type: none"> 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 |
| <ul style="list-style-type: none"> 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. When 1 is written to this bit, the value is retained. | |

(6) UARTAn receive data register (UAnRX)

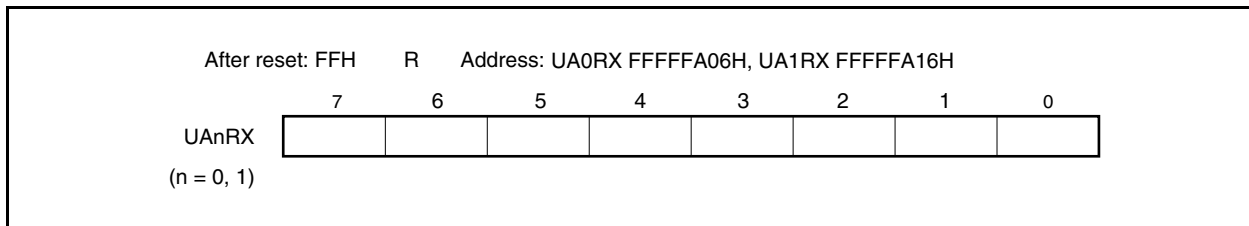
The UAnRX register is an 8-bit buffer register that stores parallel data converted by the receive shift register. The data stored in the receive shift register is transferred to the UAnRX register upon completion of reception of 1 byte of data.

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.

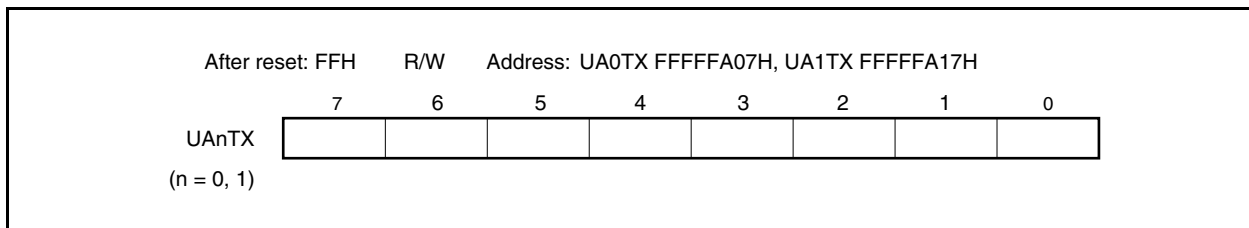


(7) UARTAn transmit data register (UAnTX)

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 sets this register to FFH.



12.4 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 12-2. 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.

<R>

A reception complete interrupt request signal is also output when a reception error occurs. Therefore, 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.

12.5 Operation

12.5.1 Data format

Full-duplex serial data reception and transmission is performed.

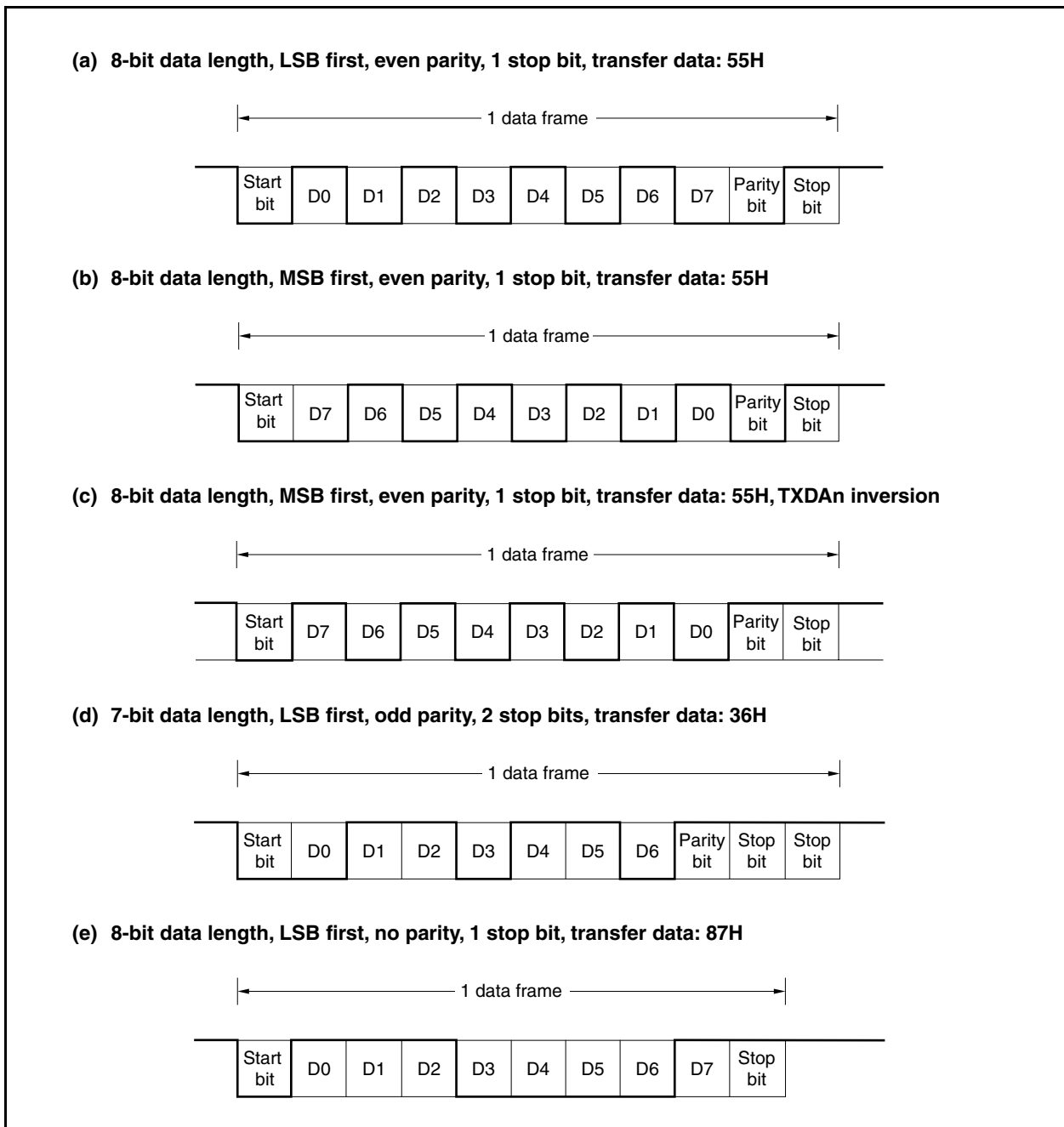
As shown in Figure 12-2, one data frame of transmit/receive data consists of a start bit, character bits, parity bit, and stop bit(s).

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 bit 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 12-2. UARTA Transmit/Receive Data Format



12.5.2 SBF transmission/reception format

The V850ES/HF2 has 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 $\pm 15\%$ or less.

Figures 12-3 and 12-4 outline the transmission and reception manipulations of LIN.

Figure 12-3. LIN Transmission Manipulation Outline

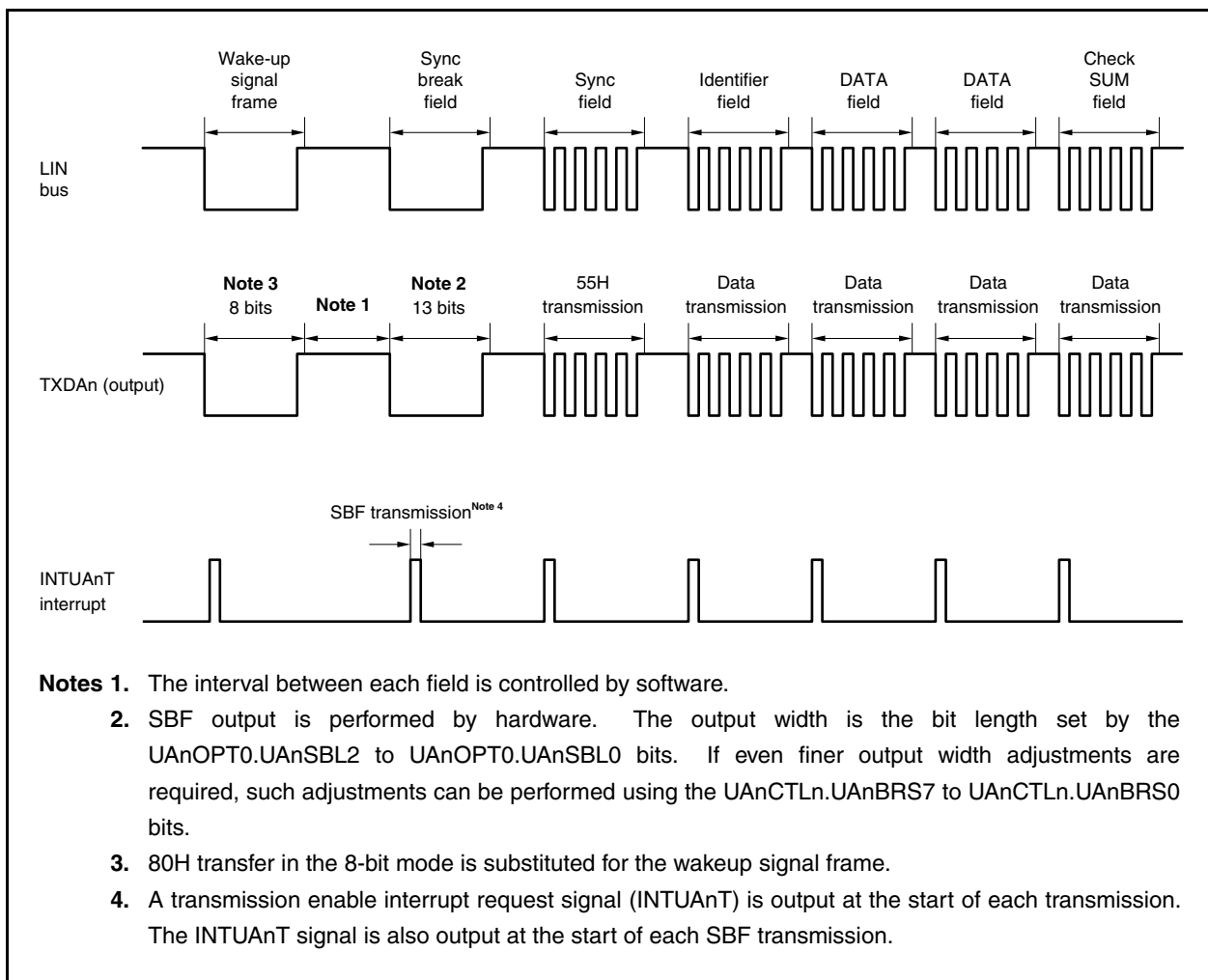
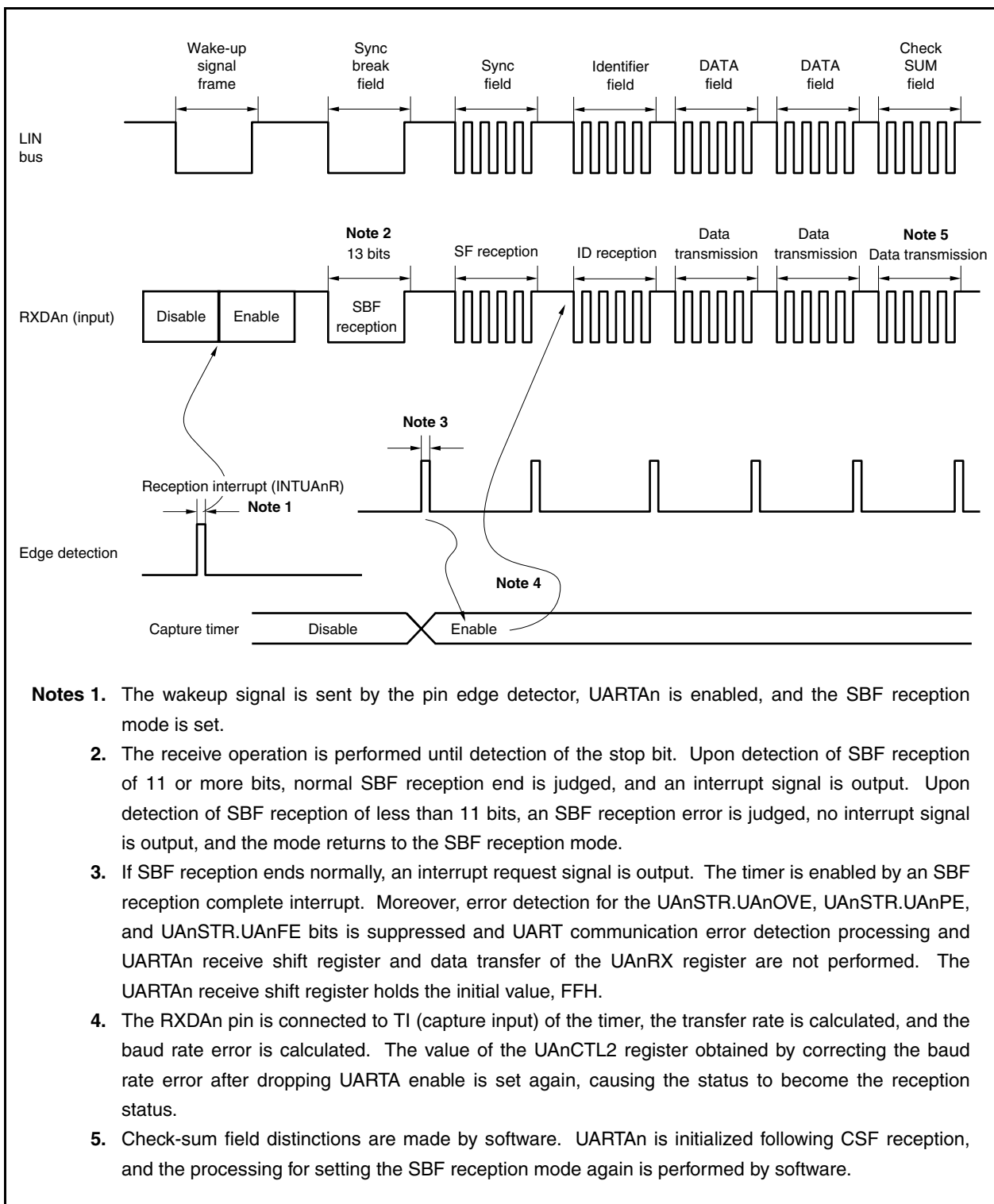


Figure 12-4. LIN Reception Manipulation Outline



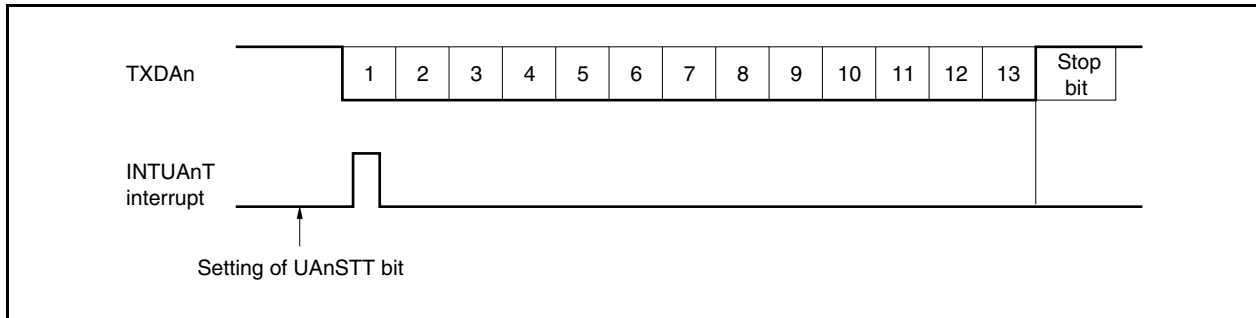
12.5.3 SBF transmission

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 the 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 12-5. SBF Transmission



12.5.4 SBF reception

The reception enabled status is achieved by setting the UAnCTL0.UAnPWR bit to 1 and then setting the UAnCTL0.UAnRXE 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.

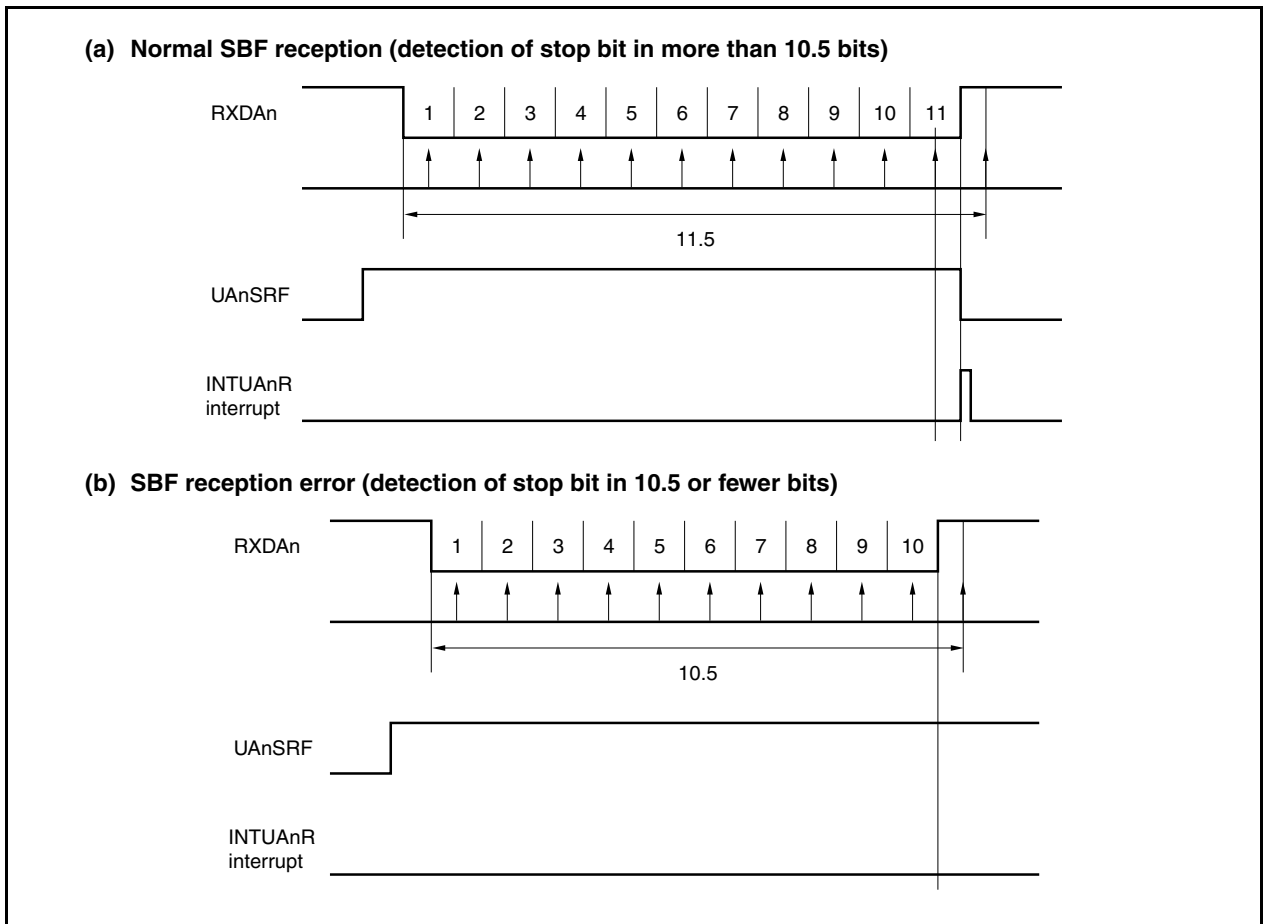
<R>

Cautions 1. If SBF is transmitted during a data reception, a framing error occurs.

<R>

2. Do not set the SBF reception trigger bit (UAnSRT) and SBF transmission trigger bit (UAnSTT) to 1 during an SBF reception (UAnSRF = 1).

Figure 12-6. SBF Reception



12.5.5 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.

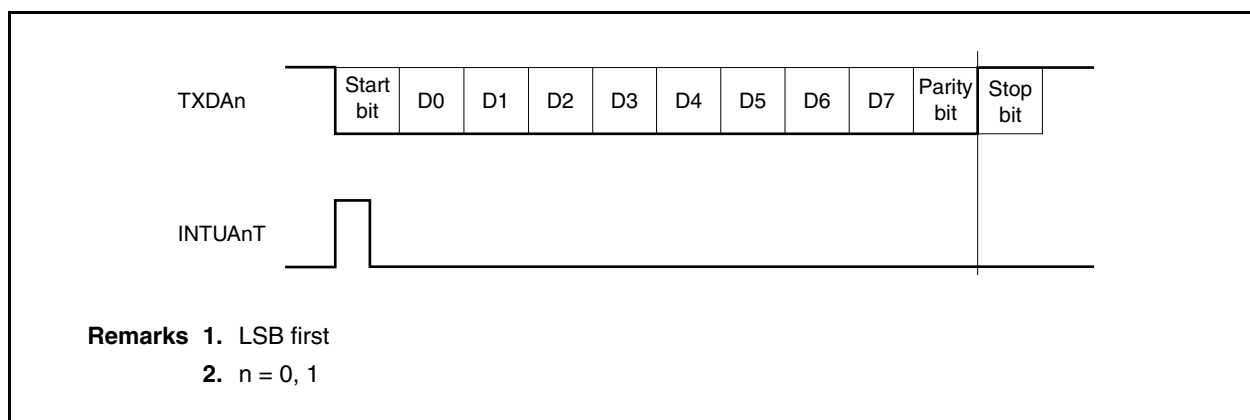
Since the CTS (transmit enable signal) input pin is not provided in UARTAn, use a port to check that reception is enabled at the transmit destination.

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 after the INTUAnT signal is generated.

Figure 12-7. UART Transmission



12.5.6 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.

During continuous transmission, do not write the next transmit data to the UAnTX register before a transmit request interrupt signal (INTUAnT) is generated after transmit data is written to the UAnTX register and transferred to the UARTAn transmit shift register. If a value is written to the UAnTX register before a transmit request interrupt signal is generated, the previously set transmit data is overwritten by the latest transmit data.

Caution When initializing transmissions during the execution of continuous transmissions, make sure that the UAnSTR.UAnTSF bit is 0, then perform the initialization. Transmit data that is initialized when the UAnTSF bit is 1 cannot be guaranteed.

In the case of continuous transmission, the communication rate from the stop bit to the start bit of the next data is extended by two operating clocks from the normal rate.

Figure 12-8. Continuous Transmission Processing Flow

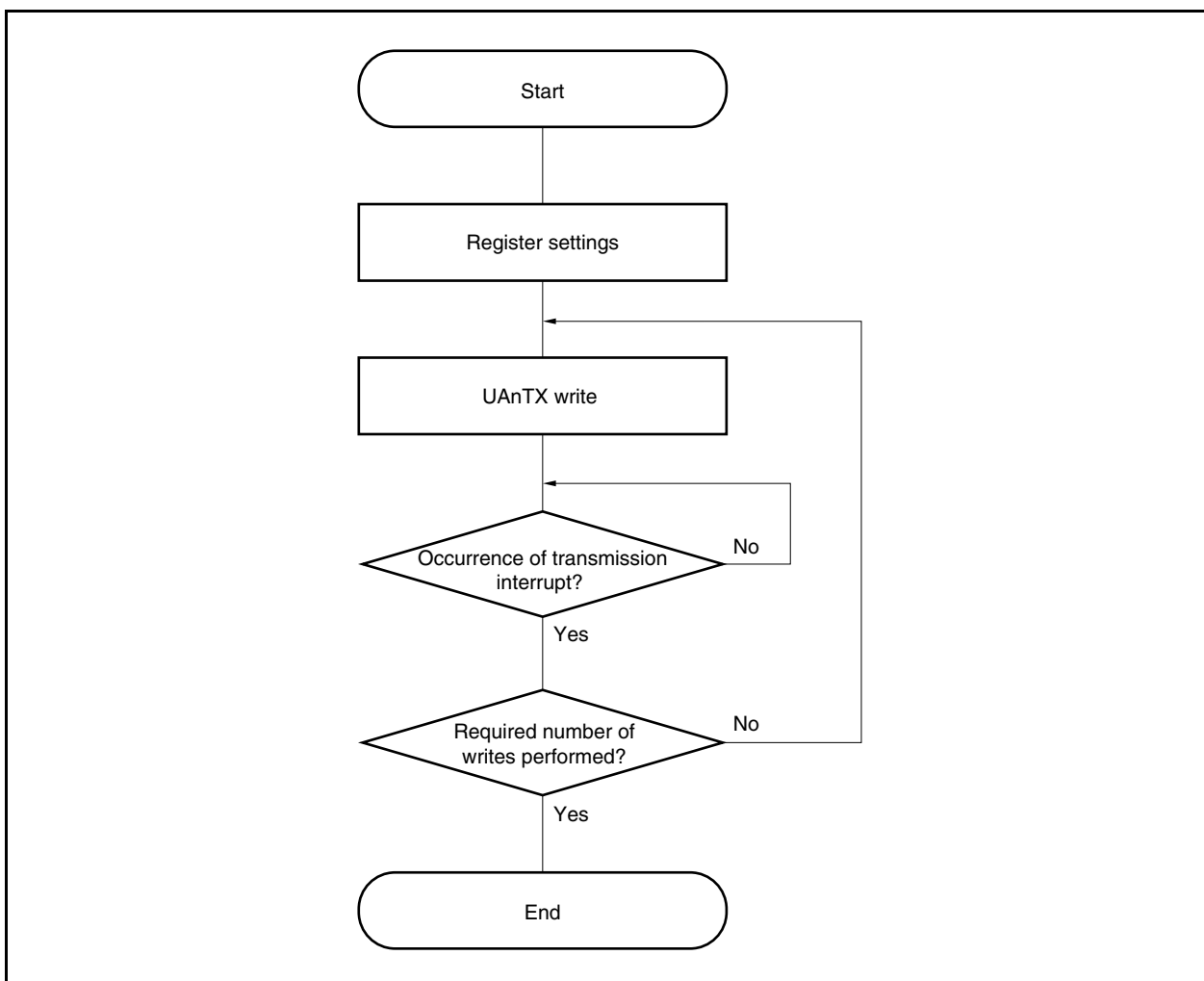
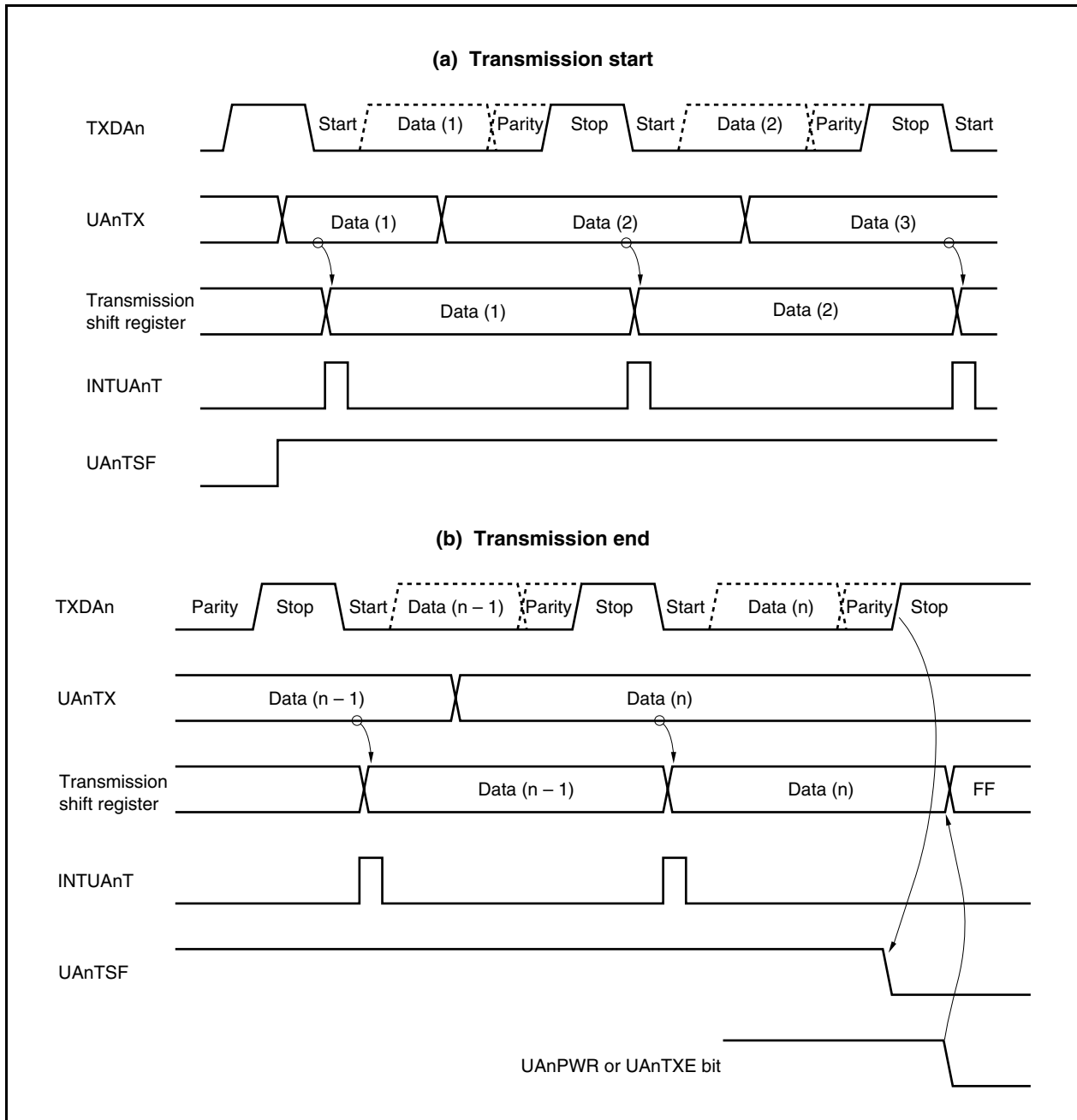


Figure 12-9. Continuous Transmission Operation Timing



12.5.7 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.

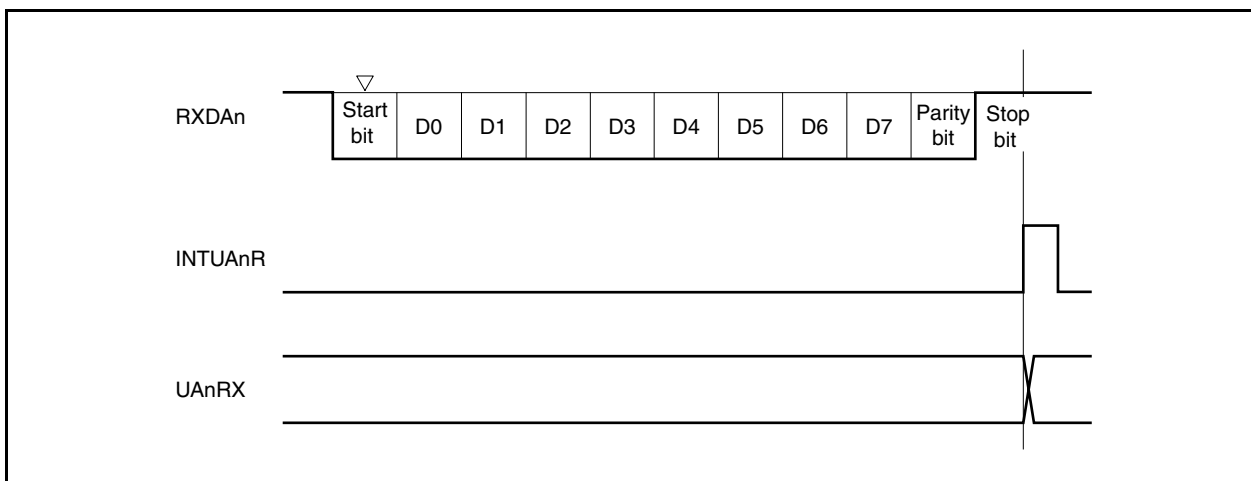
Start bit detection is performed using a two-step detection routine.

First the rising 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 is output following reception completion.

Figure 12-10. 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.

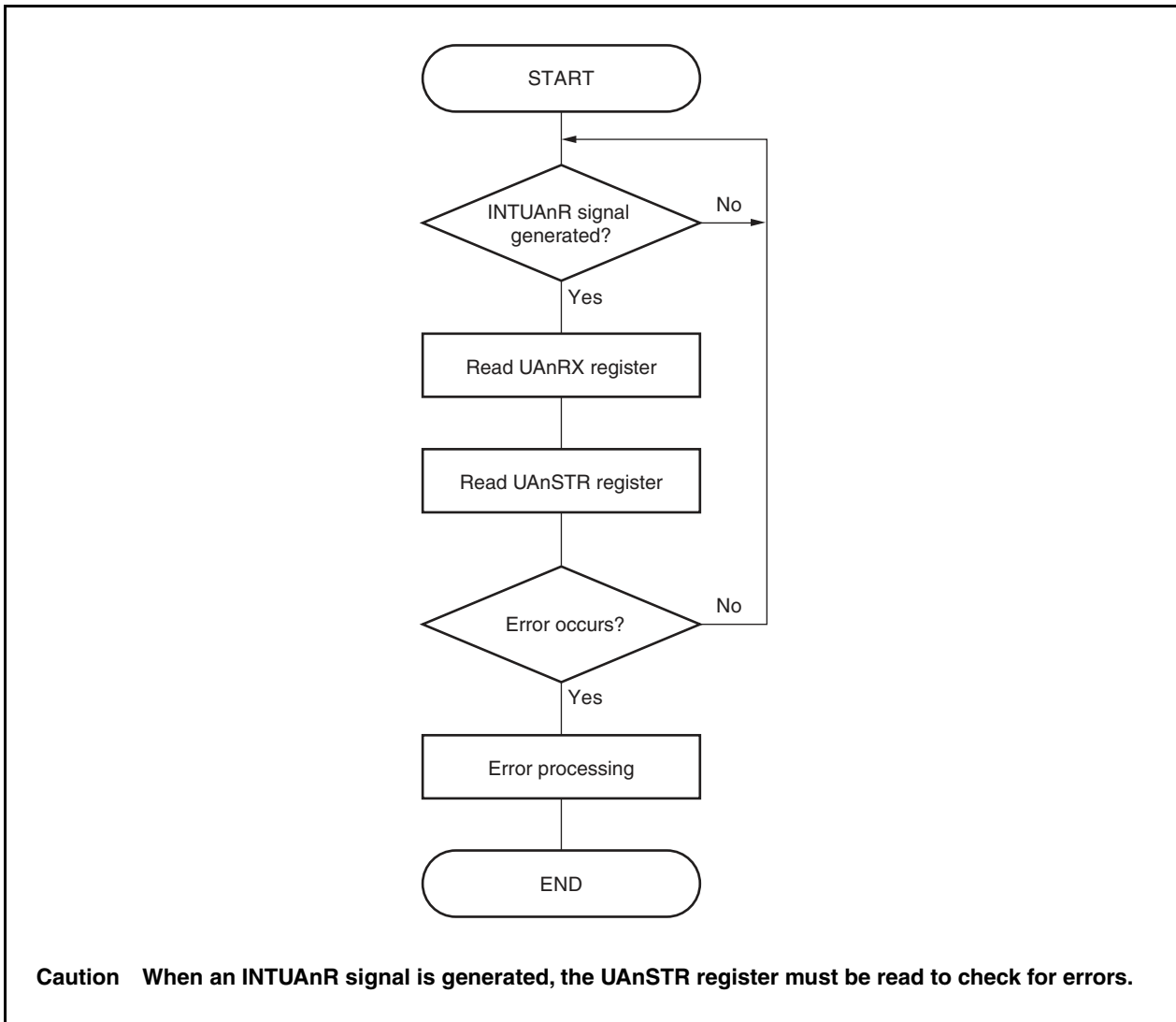
To complete reception without waiting INTUAnR signal generation, be sure to clear (0) the interrupt request flag (UAnRIF) of the UAnRIC register, after setting (1) the interrupt mask flag (UAnRMK) of the interrupt control register (UAnRIC) and then set (1) the UAnPWR bit = 0 or UAnRXE bit = 0.

12.5.8 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.

- Receive data read flow



- 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 |

When reception errors occur, perform the following procedures depending upon the kind of error.

- Parity error
If false data is received due to problems such as noise in the reception line, discard the received data and retransmit.
- Framing error
A baud rate error may have occurred between the reception side and transmission side or the start bit may have been erroneously detected. Since this is a fatal error for the communication format, check the operation stop in the transmission side, perform initialization processing each other, and then start the communication again.
- Overrun error
Since the next reception is completed before reading receive data, 1 frame of data is discarded. If this data was needed, do a retransmission.

Caution If a receive error interrupt occurs during continuous reception, read the contents of the UAnSTR register must be read before the next reception is completed, then perform error processing.

12.5.9 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 reception 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.

12.5.10 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 12-12**). See **12.6 (1) (a) Base clock** regarding the base clock.

Moreover, since the circuit is as shown in **Figure 12-11**, the processing that goes on within the receive operation is delayed by 3 clocks in relation to the external signal status.

Figure 12-11. Noise Filter Circuit

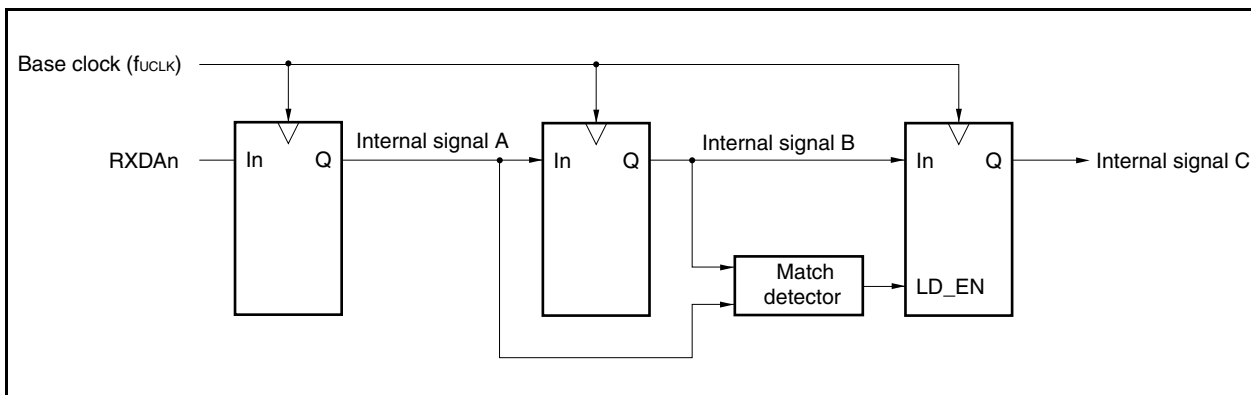
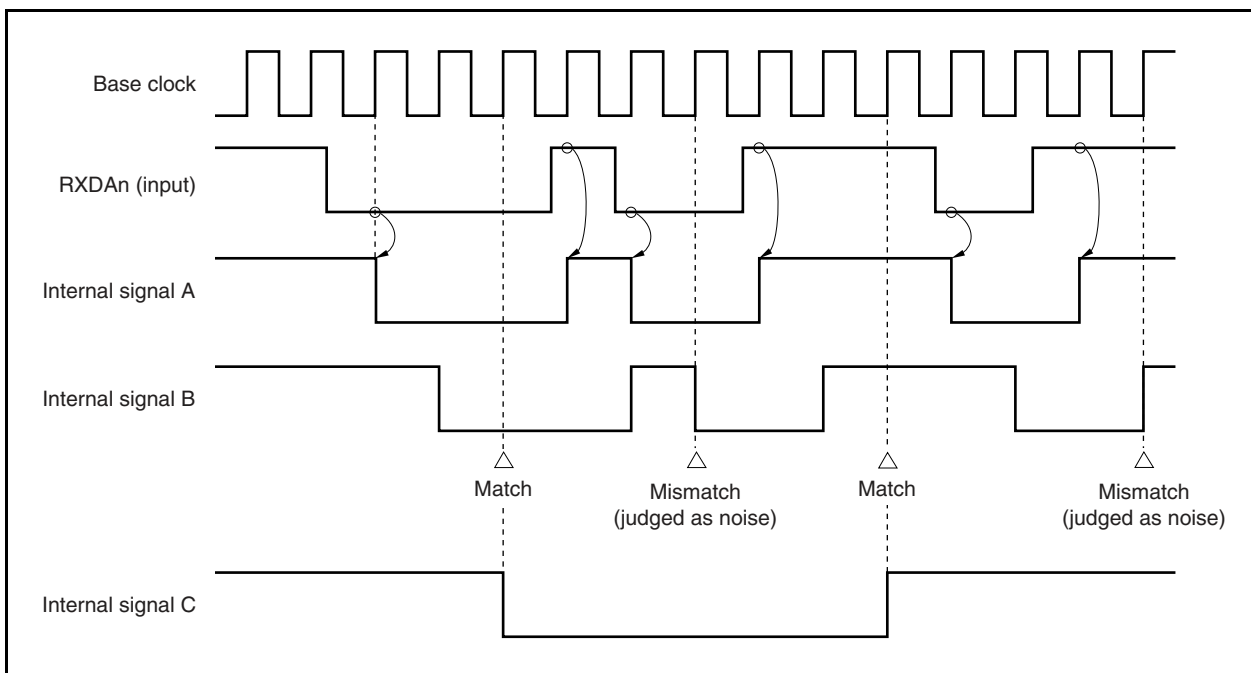


Figure 12-12. Timing of RXDAn Signal Judged as Noise



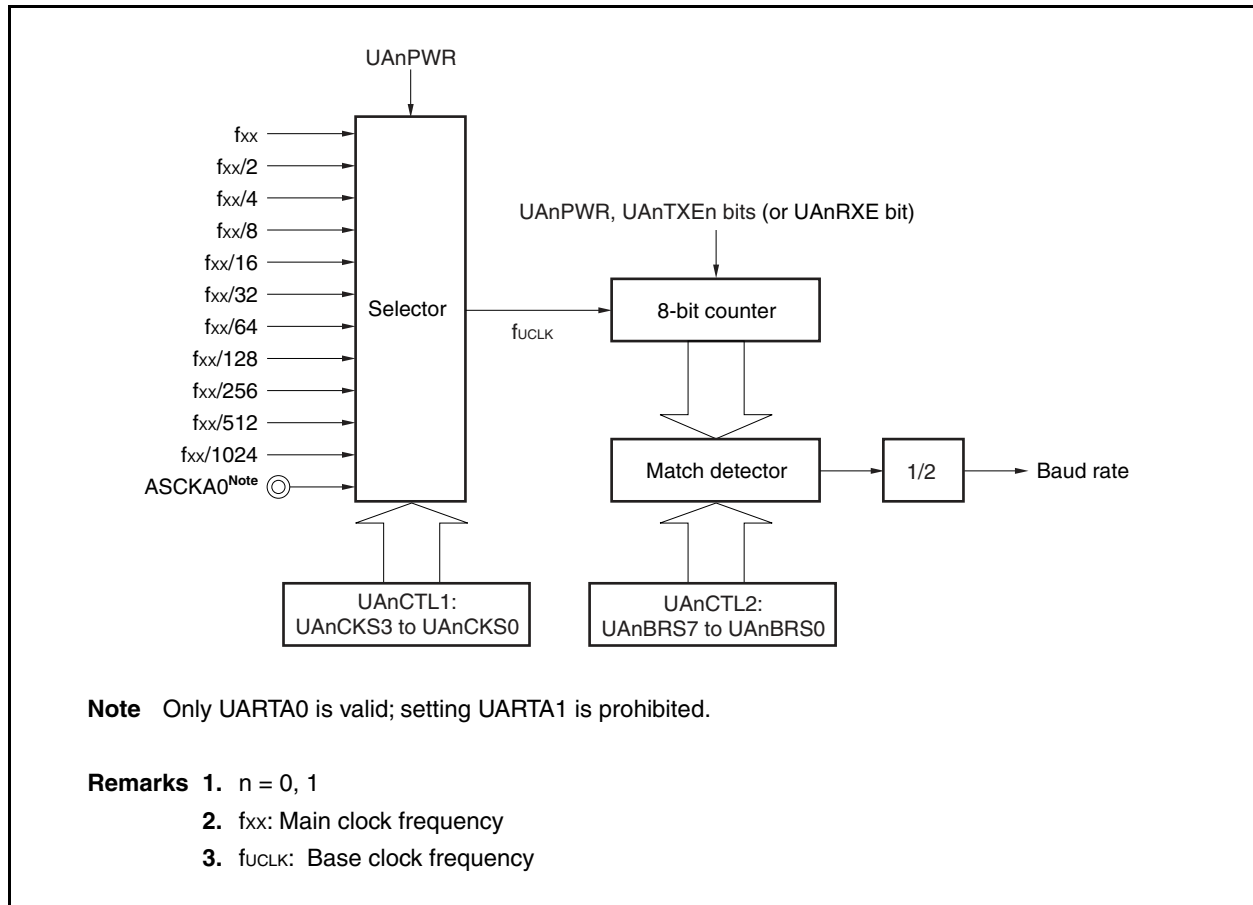
12.6 Dedicated Baud Rate Generator

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 12-13. Configuration of Baud Rate Generator



(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 (f_{UCLK}).

(b) Serial clock generation

A serial clock can be generated by setting the UAnCTL1 register and the UAnCTL2 register ($n = 0, 1$).

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 sets this register to 00H.

Caution Clear the UAnCTL0.UAnPWR bit to 0 before rewriting the UAnCTL1 register.

After reset: 00H R/W Address: UA0CTL1 FFFFFFFA01H, UA1CTL1 FFFFFFFA11H

| | | | | | | | | |
|---------|---|---|---|---|---------|---------|---------|---------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UAnCTL1 | 0 | 0 | 0 | 0 | UAnCKS3 | UAnCKS2 | UAnCKS1 | UAnCKS0 |

(n = 0, 1)

| UAnCKS3 | UAnCKS2 | UAnCKS1 | UAnCKS0 | Base clock (f _{CLK}) selection |
|------------------|---------|---------|---------|---|
| 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 | 0 | f _{xx} /64 |
| 0 | 1 | 1 | 1 | f _{xx} /128 |
| 1 | 0 | 0 | 0 | f _{xx} /256 |
| 1 | 0 | 0 | 1 | f _{xx} /512 |
| 1 | 0 | 1 | 0 | f _{xx} /1,024 |
| 1 | 0 | 1 | 1 | External clock ^{Note} (ASCKA0 pin) |
| Other than above | | | | Setting prohibited |

Note Only UARTA0 is valid; setting UARTA1 is prohibited.

Remark f_{xx}: 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 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 FFFFA02H, UA1CTL2 FFFFA12H

| | | | | | | | | | |
|---------|---------|---------|---------|---------|---------|---------|---------|--|--|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | |
| UAnBRS7 | UAnBRS6 | UAnBRS5 | UAnBRS4 | UAnBRS3 | UAnBRS2 | UAnBRS1 | UAnBRS0 | | |

UAnCTL2 (n = 0, 1)

| 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.

$$\text{Baud rate} = \frac{f_{\text{UCLK}}}{2 \times k} \text{ [bps]}$$

When using the internal clock, the equation will be as follows (when using the ASCKA0 pin as clock at UARTA0, calculate using the above equation).

$$\text{Baud rate} = \frac{f_{\text{xx}}}{2^{m+1} \times k} \text{ [bps]}$$

Remark f_{UCLK} = Frequency of base clock selected by the UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits
 f_{xx} : Main clock frequency
 m = Value set using the UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits ($m = 0$ to 10)
 k = Value set using the UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits ($k = 4$ to 255)

The baud rate error is obtained by the following equation.

$$\begin{aligned} \text{Error (\%)} &= \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Target baud rate (correct baud rate)}} - 1 \right) \times 100 \text{ [\%]} \\ &= \left(\frac{f_{\text{UCLK}}}{2 \times k \times \text{Target baud rate}} - 1 \right) \times 100 \text{ [\%]} \end{aligned}$$

When using the internal clock, the equation will be as follows (when using the ASCKA0 pin as clock at UARTA0, calculate the baud rate error using the above equation).

$$\text{Error (\%)} = \left(\frac{f_{\text{xx}}}{2^{m+1} \times k \times \text{Target baud rate}} - 1 \right) \times 100 \text{ [\%]}$$

- 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 (5) Allowable baud rate range during reception.

To set the baud rate, perform the following calculation and set the UAnCTL1 and UAnCTL2 registers (when using internal clock).

- <1> Set $k = f_{xx}/(2 \times \text{Target baud rate})$. Set $m = 0$.
- <2> Set $k = k/2$ and $m = m + 1$ where $k \geq 256$.
- <3> Repeat <2> until $k < 256$.
- <4> Roundup the first decimal place of k .
If $k = 256$ by the roundup, perform <2> again (k will become 128).
- <5> Set m to the UAnCTL1 register and k to the UAnCTL2 register.

Example: When $f_{xx} = 20 \text{ MHz}$ and target baud rate = 153,600 bps

<1> $k = 20,000,000/(2 \times 153,600) = 65.10\dots, m = 0$

<2>, <3> $k = 65.10\dots < 256, m = 0$

<4> Set value of UAnCTL2 register: $k = 65 = 41\text{H}$, set value of UAnCTL1 register: $m = 0$

Actual baud rate = $20,000,000/(2 \times 65)$
= 153,846 [bps]

Baud rate error = $\{20,000,000/(2 \times 65 \times 153,600) - 1\} \times 100$
= 0.160 [%]

The representative examples of baud rate settings are shown below.

Table 12-3. Baud Rate Generator Setting Data

| Baud Rate (bps) | $f_{xx} = 20 \text{ MHz}$ | | | $f_{xx} = 16 \text{ MHz}$ | | | $f_{xx} = 10 \text{ MHz}$ | | |
|-----------------|---------------------------|---------|---------|---------------------------|---------|---------|---------------------------|---------|---------|
| | UAnCTL1 | UAnCTL2 | ERR (%) | UAnCTL1 | UAnCTL2 | ERR (%) | UAnCTL1 | UAnCTL2 | ERR (%) |
| 300 | 08H | 82H | 0.16 | 0AH | 1AH | 0.16 | 07H | 82H | 0.16 |
| 600 | 07H | 82H | 0.16 | 0AH | 0DH | 0.16 | 06H | 82H | 0.16 |
| 1,200 | 06H | 82H | 0.16 | 09H | 0DH | 0.16 | 05H | 82H | 0.16 |
| 2,400 | 05H | 82H | 0.16 | 08H | 0DH | 0.16 | 04H | 82H | 0.16 |
| 4,800 | 04H | 82H | 0.16 | 07H | 0DH | 0.16 | 03H | 82H | 0.16 |
| 9,600 | 03H | 82H | 0.16 | 06H | 0DH | 0.16 | 02H | 82H | 0.16 |
| 19,200 | 02H | 82H | 0.16 | 05H | 0DH | 0.16 | 01H | 82H | 0.16 |
| 31,250 | 01H | A0H | 0.00 | 01H | 80H | 0.00 | 00H | A0H | 0.00 |
| 38,400 | 01H | 82H | 0.16 | 00H | D0H | 0.16 | 00H | 82H | 0.16 |
| 76,800 | 00H | 82H | 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 | 00H | 20H | 0.00 | 00H | 1AH | -1.54 | 00H | 10H | 0.00 |

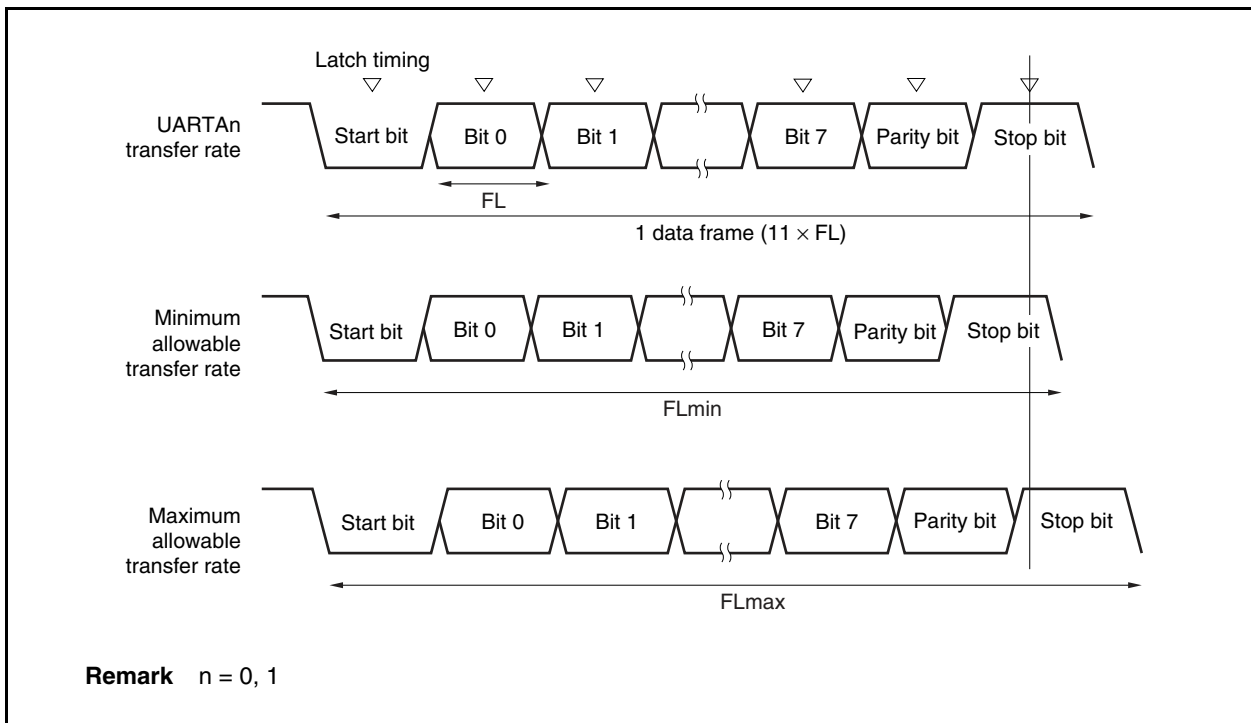
Remark f_{xx} : Main clock frequency
ERR: Baud rate error (%)

(5) 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 12-14. Allowable Baud Rate Range During Reception



As shown in Figure 12-14, 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.

$$FL = (\text{Brate})^{-1}$$

Brate: UARTAn baud rate ($n = 0, 1$)

k: Setting value of UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits ($n = 0, 1$)

FL: 1-bit data length

Latch timing margin: 2 clocks

$$\text{Minimum allowable transfer rate: } FL_{\min} = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

Therefore, the maximum baud rate that can be received by the destination is as follows.

$$BR_{max} = (FL_{min}/11)^{-1} = \frac{22k}{21k + 2} \text{ Brate}$$

Similarly, obtaining the following maximum allowable transfer rate yields the following.

$$\frac{10}{11} \times FL_{max} = 11 \times FL - \frac{k + 2}{2 \times k} \times FL = \frac{21k - 2}{2 \times k} FL$$

$$FL_{max} = \frac{21k - 2}{20k} FL \times 11$$

Therefore, the minimum baud rate that can be received by the destination is as follows.

$$BR_{min} = (FL_{max}/11)^{-1} = \frac{20k}{21k - 2} \text{ Brate}$$

Obtaining the allowable baud rate error for UARTA and the destination from the above-described equations for obtaining the minimum and maximum baud rate values yields the following.

Table 12-4. Maximum/Minimum Allowable Baud Rate Error

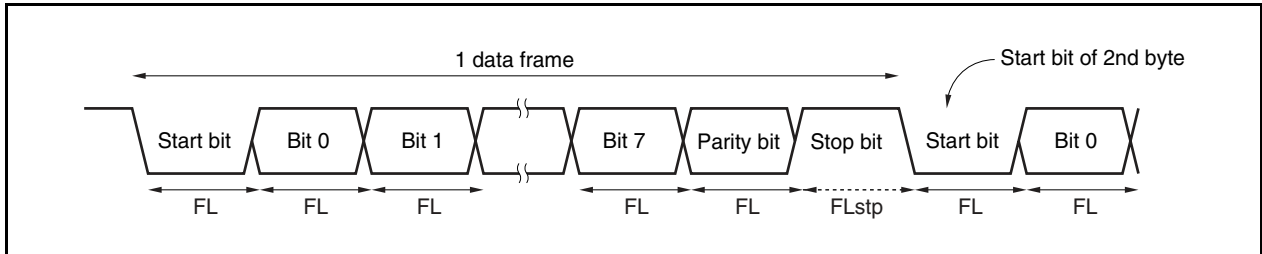
| Division Ratio (k) | Maximum Allowable Baud Rate Error | Minimum Allowable Baud Rate Error |
|--------------------|-----------------------------------|-----------------------------------|
| 4 | +2.32% | -2.43% |
| 8 | +3.52% | -3.61% |
| 20 | +4.26% | -4.30% |
| 50 | +4.56% | -4.58% |
| 100 | +4.66% | -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, 1)

(6) Baud 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 12-15. Transfer Rate During Continuous Transfer



Assuming 1 bit data length: FL; stop bit length: FLstp; and base clock frequency: f_{UCLK} , we obtain the following equation.

$$\text{FLstp} = \text{FL} + 2/f_{\text{UCLK}}$$

Therefore, the transfer rate during continuous transmission is as follows.

$$\text{Transfer rate} = 11 \times \text{FL} + (2/f_{\text{UCLK}})$$

12.7 Cautions

- (1) 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.
- (2) The RXDA1 and KR7 pins must not be used at the same time. To use the RXDA1 pin, do not use the KR7 pin. To use the KR7 pin, do not use the RXDA1 pin (it is recommended to set the PFC91 bit to 1 and clear PFCE91 bit to 0).
- (3) In UARTAn, the interrupt caused by a communication error does not occur. When performing the transfer of receive data, error processing cannot be performed even if errors (parity, overrun, framing) occur during transfer. Read the UAnSTR register during communication to check for errors.
- (4) Start up the UARTAn in the following sequence.
 - <1> Set the UAnCTL0.UAnPWR bit to 1.
 - <2> Set the ports.
 - <3> Set the UAnCTL0.UAnTXE bit to 1, UAnCTL0.UAnRXE bit to 1.
- (5) Stop the UARTAn in the following sequence.
 - <1> Set the UAnCTL0.UAnTXE bit to 0, UAnCTL0.UAnRXE bit to 0.
 - <2> Set the ports and set the UAnCTL0.UAnPWR bit to 0 (it is not a problem if port setting is not changed).
- (6) In transmit mode (UAnCTL0.UAnPWR bit = 1 and UAnCTL0.UAnTXE bit = 1), do not overwrite the same value to the UAnTX register by software because transmission starts by writing to this register. To transmit the same value continuously, overwrite the same value.
- (7) In continuous transmission, the communication rate from the stop bit to the next start bit is extended 2 base clocks more than usual. However, the reception side initializes the timing by detecting the start bit, so the reception result is not affected.
- (8) If the break command is executed in the on-chip debug (OCD) mode and if UART receives data, an overrun error occurs.

CHAPTER 13 3-WIRE VARIABLE-LENGTH SERIAL I/O (CSIB)

The V850ES/HF2 has two channels of 3-wire serial interface (CSIB).

13.1 Features

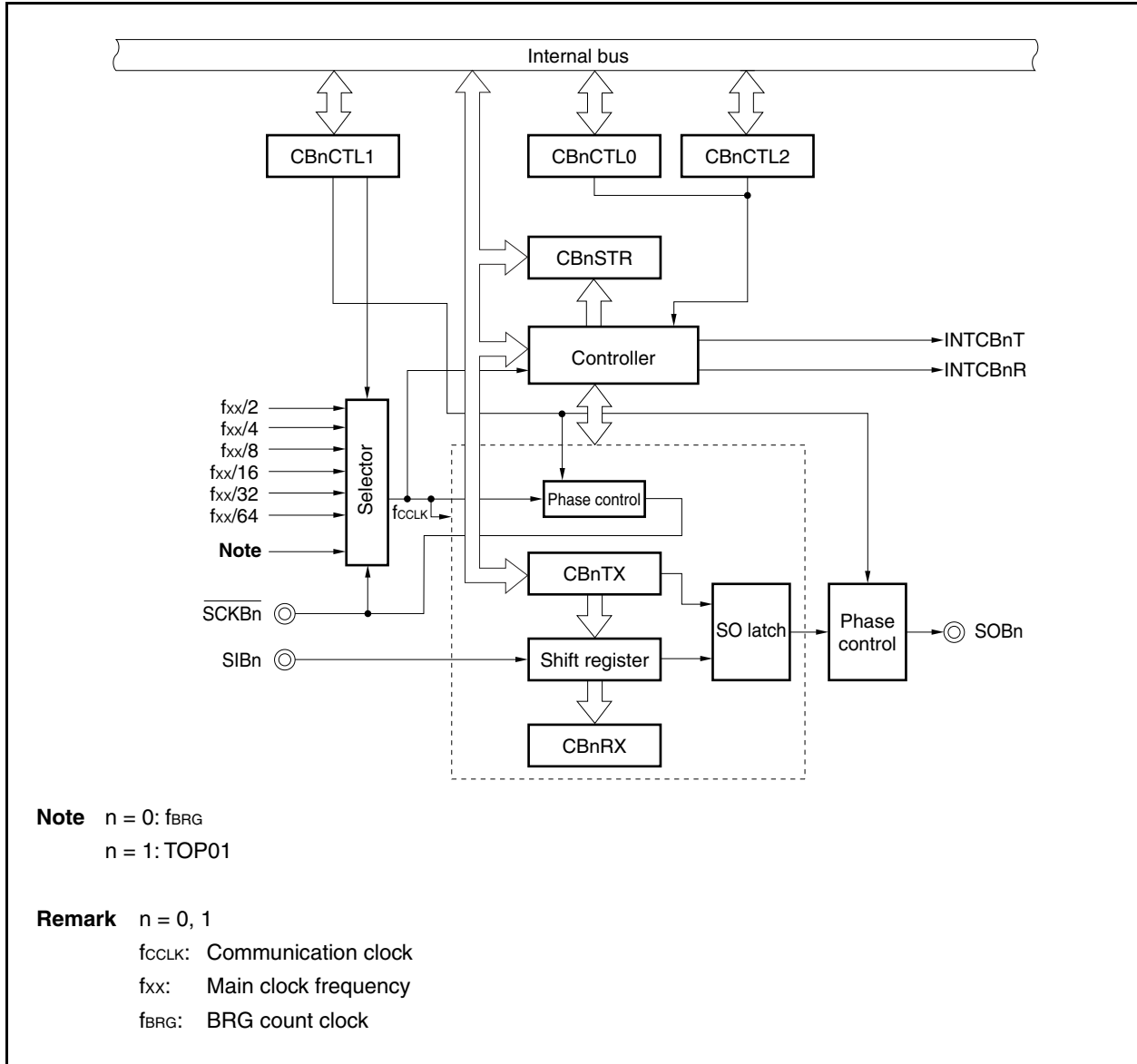
- Transfer rate: 8 Mbps max. ($f_{xx} = 20$ MHz, using internal clock)
 - Master mode and slave mode selectable
 - 8-bit to 16-bit transfer, 3-wire serial interface
 - Interrupt request signals (INTCBnT, INTCBnR) $\times 2$
 - Serial clock and data phase switchable
 - Transfer data length selectable 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

13.2 Configuration

The following shows the block diagram of CSIBn.

Figure 13-1. Block Diagram of CSIBn



CSIBn includes the following hardware.

Table 13-1. 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.

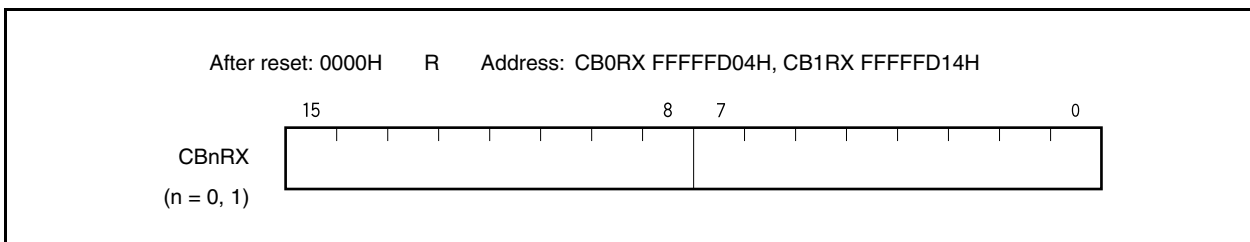
This register is read-only, in 16-bit units.

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 sets 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.



(2) CSIB transmit data register (CBnTX)

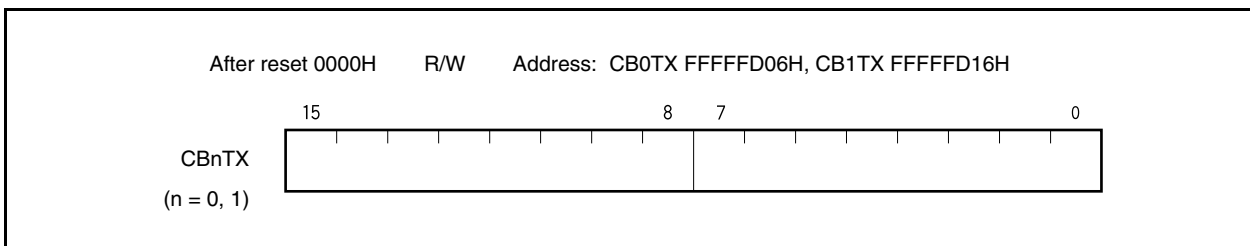
The CBnTX register is a 16-bit buffer register used to write the CSIBn transfer data.

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 are read-only in 8-bit units as the CBnTXL register.

Reset sets this register to 0000H.



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 |

13.3 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 sets this register to 01H.

(1/3)

| | | | | | | | | |
|--|---|------------------------|------------------------|------------------------|---|---|------------------------|--------|
| After reset: 01H R/W Address: CB0CTL0 FFFFFFFD00H, CB1CTL0 FFFFFFFD10H | | | | | | | | |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CBnCTL0 | CBnPWR | CBnTXE ^{Note} | CBnRXE ^{Note} | CBnDIR ^{Note} | 0 | 0 | CBnTMS ^{Note} | CBnSCE |
| (n = 0, 1) | | | | | | | | |
| 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 | | | | | | | |
| • When the CBnRXE bit is cleared to 0, no reception complete interrupt is output even when the prescribed data is transferred in order to disable the receive operation, and the receive data (CBnRX register) is not updated. | | | | | | | | |

Note These bits can only be rewritten when the CBnPWR bit = 0. However, CBnPWR bit = 1 can also be set at the same time as rewriting these bits.

Caution To forcibly suspend transmission/reception, clear the CBnPWR bit to 0 instead of the CBnRXE or CBnTXE bit.
At this time, the clock output is stopped.

<R>

| CBnDIR ^{Note} | Specification of transfer direction mode (MSB/LSB) |
|------------------------|--|
| 0 | MSB-first transfer |
| 1 | LSB-first transfer |

| CBnTMS ^{Note} | Transfer mode specification |
|------------------------|-----------------------------|
| 0 | Single transfer mode |
| 1 | Continuous transfer mode |

[In single transfer mode]

The reception complete interrupt request signal (INTCBnR) is generated.

Even if transmission is enabled (CBnTXE bit = 1), the transmission enable interrupt request signal (INTCBnT) is not generated.

If the next transmit data is written during communication (CBnSTR.CBnTSF bit = 1), it is ignored and the next communication is not started. Also, if reception-only communication is set (CBnTXE bit = 0, CBnRXE bit = 1), the next communication is not started even if the receive data is read during communication (CBnSTR.CBbTSF bit = 1).

[In continuous transfer mode]

The continuous transmission is enabled by writing the next transmit data during communication (CBnSTR.CBnTSF bit = 1). Writing the next transmission data is enabled after a transmission enable interrupt (INTCBnT) occurrence.

If reception-only communication is set (CBnTXE bit = 0, CBnRXE bit = 1) in the continuous transfer mode, the next reception is started continuously after a reception complete interrupt (INTCBnR) regardless of the read operation of the CBnRX register.

Therefore, read immediately the receive data from the CBnRX register. If this read operation is delayed, an overrun error (CBnOVE bit = 1) occurs.

Note These bits can only be rewritten when the CBnPWR bit = 0. However, CBnPWR bit = 1 can also be set at the same time as rewriting these bits.

| 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
The setting of the CBnSCE bit has no influence on communication operation.
 - (b) In single reception mode
Clear the CBnSCE bit to 0 before reading the last receive data because reception is started by reading the receive data (CBnRX register) to disable the reception startup^{Note 1}.
 - (c) In continuous reception mode
Clear the CBnSCE bit to 0 one communication clock before reception of the last data is completed to disable the reception startup after the last data is received^{Note 2}.
- In slave mode
This bit enables or disables the communication start trigger.
Set the CBnSCE bit to 1.

[Usage of CBnSCE bit]

- In single reception mode
 - <1>When reception of the last data is completed by INTCBnR interrupt servicing, clear the CBnSCE bit to 0 before reading the CBnRX register.
 - <2>After confirming the CBnSTR.CBnTSF bit = 0, clear the CBnRXE bit to 0 to disable reception.
To continue reception, set the CBnSCE bit to 1 to start up the next reception by dummy-reading the CBnRX register.
- In continuous reception mode
 - <1>Clear the CBnSCE bit to 0 during the reception of the last data by INTCBnR interrupt servicing.
 - <2>Read the CBnRX register.
 - <3>Read the last reception data by reading the CBnRX register after acknowledging the CBnTIR interrupt.
 - <4>After confirming the CBnSTR.CBnTSF bit = 0, clear the CBnRXE bit to 0 to disable reception.
To continue reception, set the CBnSCE bit to 1 to wait for the next reception by dummy-reading the CBnRX register.

- Notes**
- If the CBnSCE bit is read while it is 1, the next communication operation is started.
 - 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.

Caution Be sure to clear bits 3 and 2 to "0".

(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 sets this register to 00H.

<R>

Caution The CBnCTL1 register can be rewritten only when the CBnCTL0.CBnPWR bit = 0, or CBnCTL0.CBnTXE and CBnRXE bits = 0.

After reset 00H R/W Address: CB0CTL1 FFFFFFFD01H, CB1CTL1 FFFFFFFD11H

| | | | | | | | | |
|------------|---|---|---|--------|--------|---------|---------|---------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CBnCTL1 | 0 | 0 | 0 | CBnCKP | CBnDAP | CBnCKS2 | CBnCKS1 | CBnCKS0 |
| (n = 0, 1) | | | | | | | | |

| | CBnCKP | CBnDAP | Specification of data transmission/reception timing in relation to 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 (f _{CCLK}) ^{Note 1} | | Mode |
|---------|---------|---------|--|--------------|-------------|
| | | | n = 0 | n = 1 | |
| 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 | f _{BRG} ^{Note 2} | TMP0 (TOP01) | Master mode |
| 1 | 1 | 1 | External clock (SCKBn) | | Slave mode |

Notes 1. Set so that communication clock (f_{CCLK}) is 8 MHz or less.
 2. For details, see 13.7 Baud Rate Generator.

<R>

(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 sets 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 FFFFFFFD02H, CB1CTL2 FFFFFFFD12H

| | | | | | | | | |
|---------|---|---|---|---|--------|--------|--------|--------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CBnCTL2 | 0 | 0 | 0 | 0 | CBnCL3 | CBnCL2 | CBnCL1 | CBnCL0 |

(n = 0, 1)

| 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 |

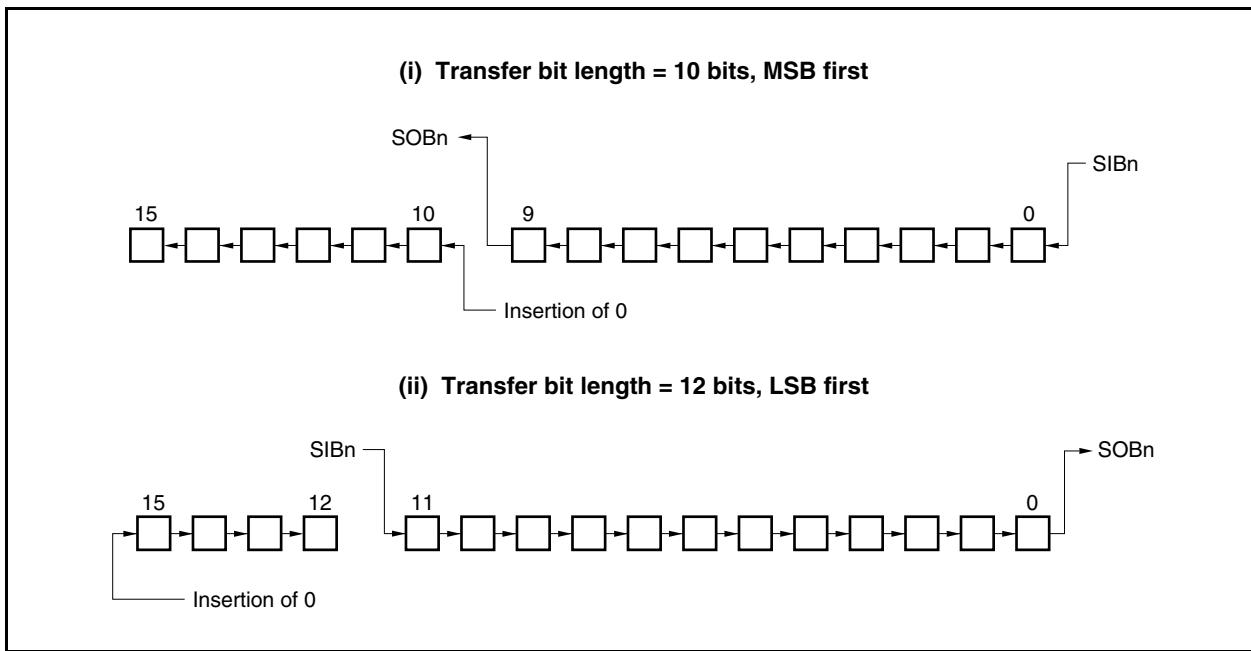
Remarks 1. If the number of transfer bits is other than 8 or 16, prepare and use data stuffed from the LSB of the CBnTX and CBnRX registers.

2. ×: don't care

(a) Transfer data length change function

The CSIB_n 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.



(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 sets 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 FFFFFFFD03H, CB1STR FFFFFFFD13H

| | | | | | | | | |
|--------|--------|---|---|---|---|---|---|--------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CBnSTR | CBnTSF | 0 | 0 | 0 | 0 | 0 | 0 | CBnOVE |

(n = 0, 1)

| | |
|--------|---------------------------|
| 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 ends without reading the value of the receive buffer by CPU, upon completion of the receive operation.
- The CBnOVE flag displays the overrun error occurrence status in this case.
- The CBnOVE bit is valid also in the single transfer mode. Therefore, when only using transmission, note the following.
 - Do not check the CBnOVE flag.
 - Read this bit even if reading the reception data is not required.
- The CBnOVE flag is cleared by writing 0 to it. It cannot be set even by writing 1 to it.

13.4 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 13-2. 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 an overrun error occurs.

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.

In the single transfer mode, the INTCBnR interrupt request signal is generated upon completion of transmission, even when only transmission is executed.

(2) Transmission enable interrupt request signal (INTCBnT)

In the continuous transmission or continuous transmission/reception mode, transmit data is transferred from the CBnTX register and, as soon as writing to CBnTX has been enabled, the transmission enable interrupt request signal is generated.

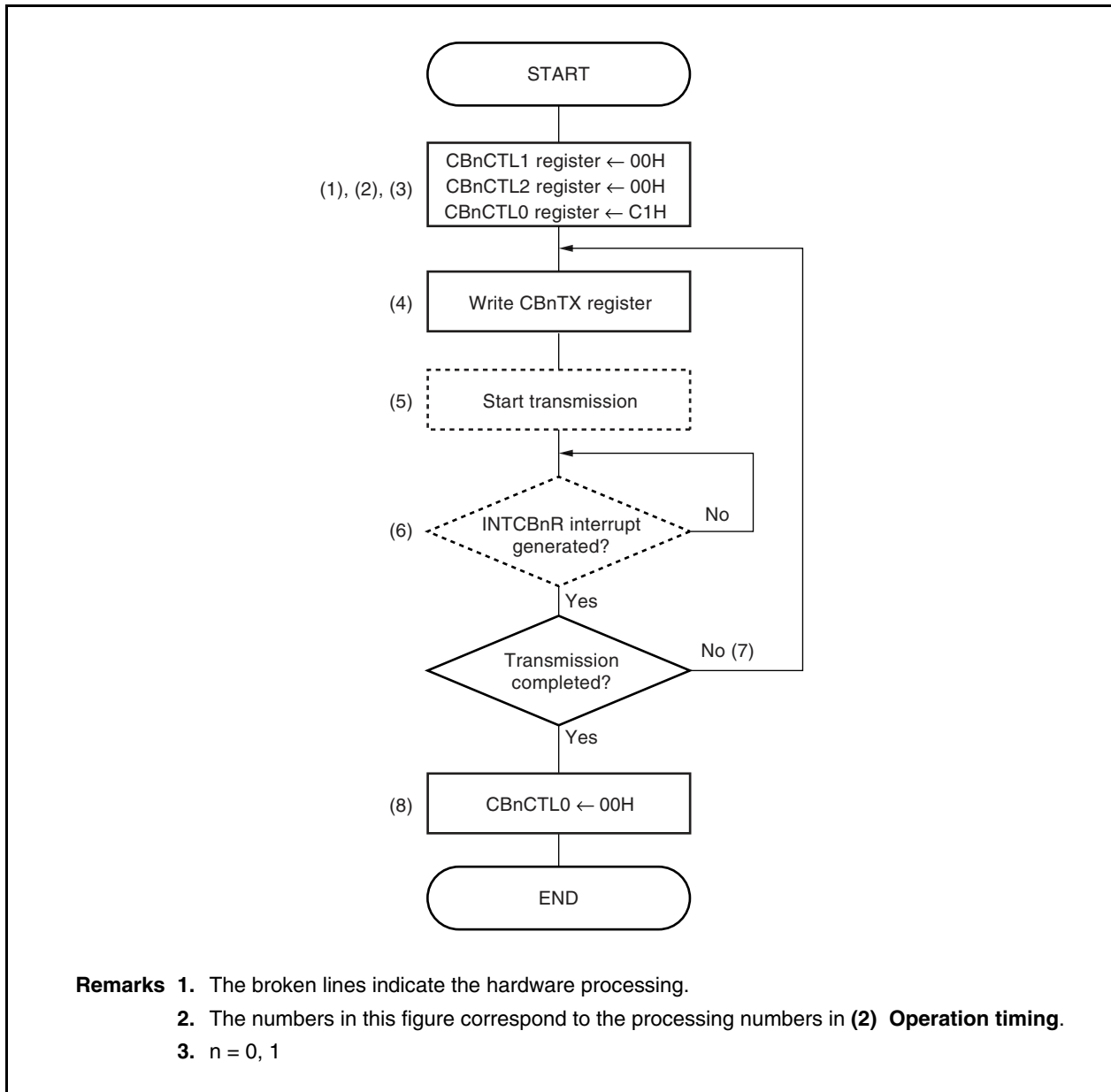
In the single transmission and single transmission/reception modes, the INTCBnT interrupt is not generated.

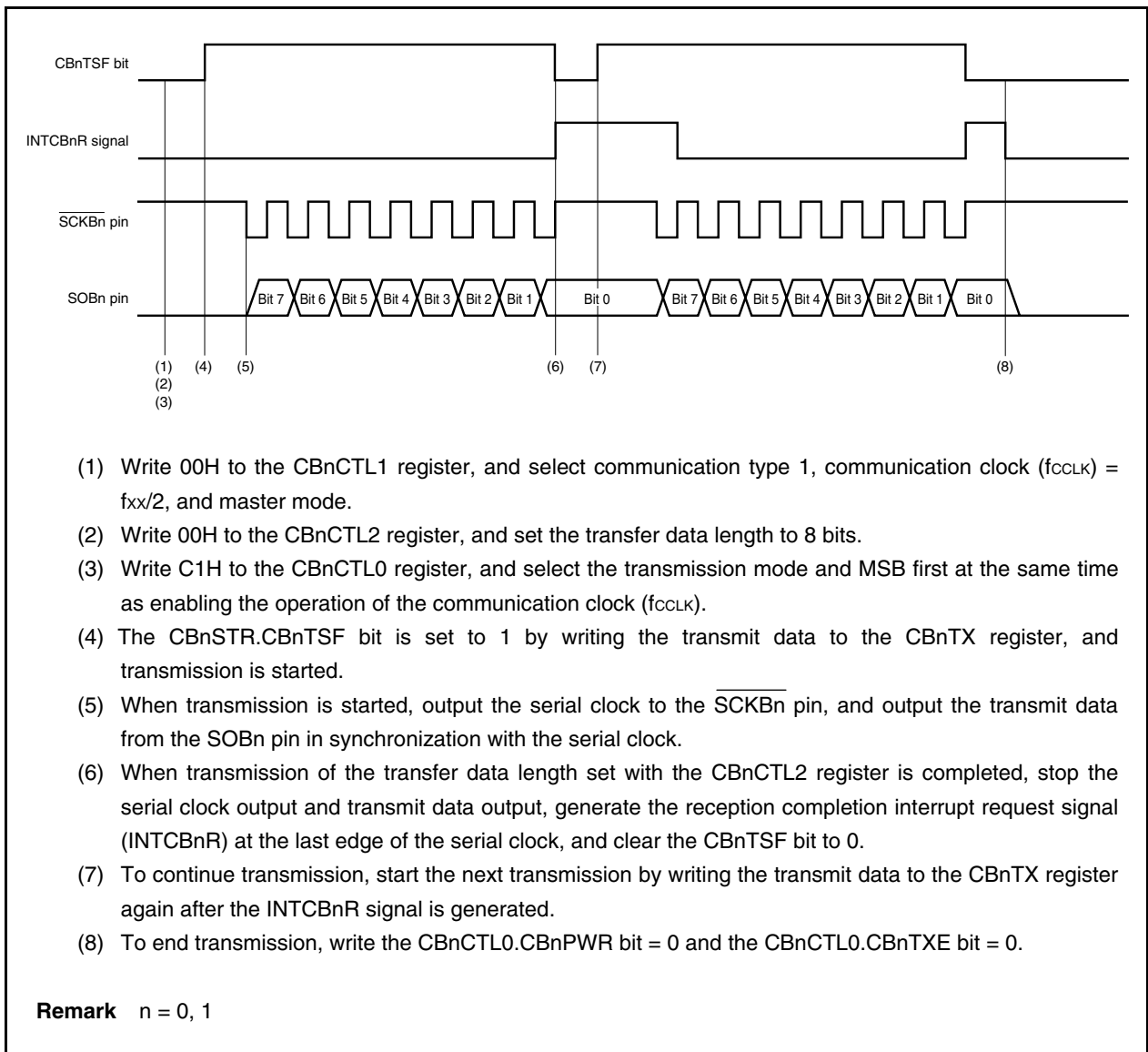
<R> 13.5 Operation

13.5.1 Single transfer mode (master mode, transmission mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CLK}) = $f_{\text{XX}}/2$ (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow



(2) Operation timing

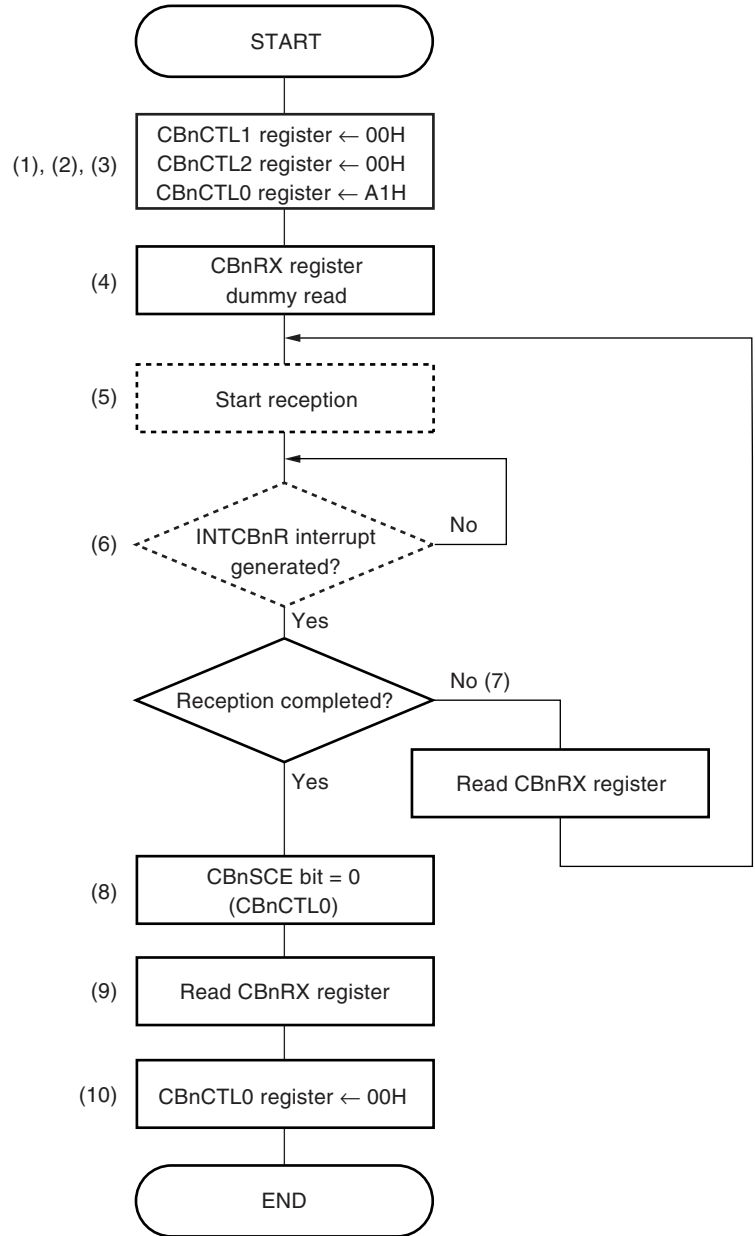
- (1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CLK}) = $f_{xx}/2$, and master mode.
- (2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.
- (3) Write C1H to the CBnCTL0 register, and select the transmission mode and MSB first at the same time as enabling the operation of the communication clock (f_{CLK}).
- (4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and transmission is started.
- (5) When transmission is started, output the serial clock to the \overline{SCKBn} pin, and output the transmit data from the SOBn pin in synchronization with the serial clock.
- (6) When transmission of the transfer data length set with the CBnCTL2 register is completed, stop the serial clock output and transmit data output, generate the reception completion interrupt request signal (INTcBnR) at the last edge of the serial clock, and clear the CBnTSF bit to 0.
- (7) To continue transmission, start the next transmission by writing the transmit data to the CBnTX register again after the INTcBnR signal is generated.
- (8) To end transmission, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnTXE bit = 0.

Remark n = 0, 1

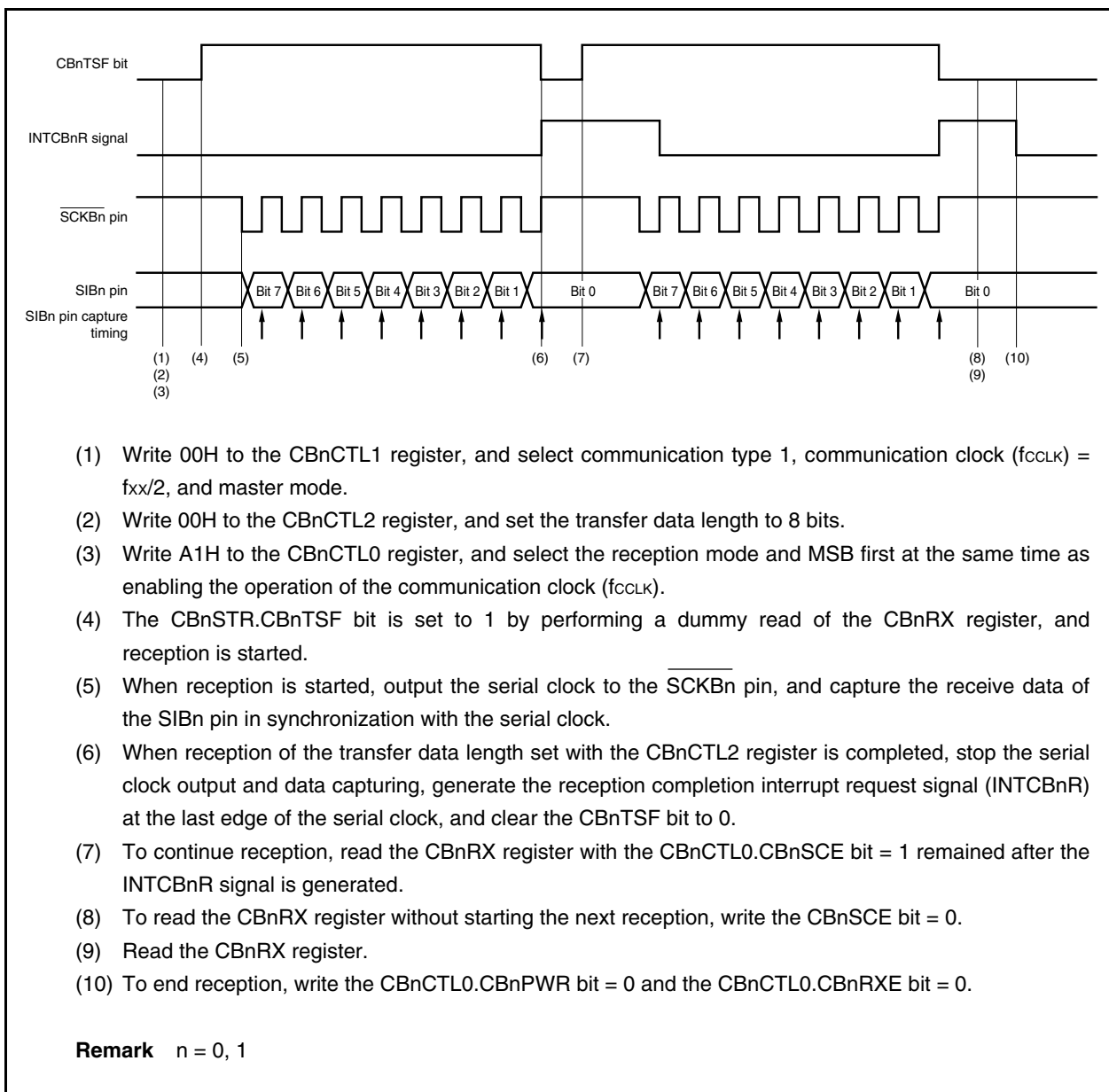
13.5.2 Single transfer mode (master mode, reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CLK}) = $f_{xx}/2$ (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow



- Remarks**
1. The broken lines indicate the hardware processing.
 2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.
 3. n = 0, 1

(2) Operation timing

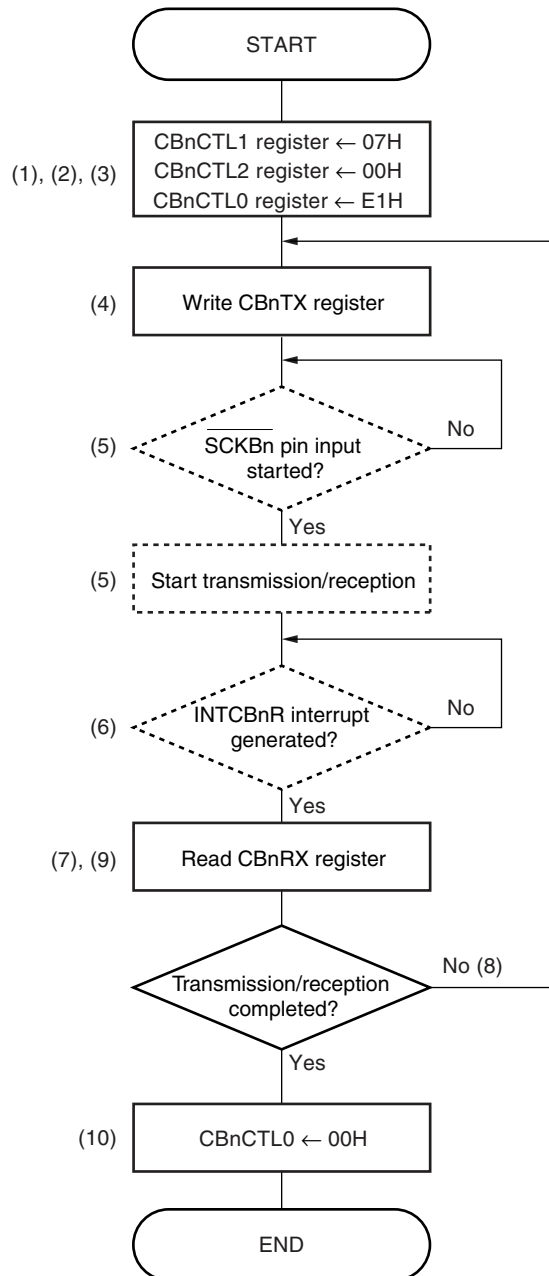
- (1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = $f_{xx}/2$, and master mode.
- (2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.
- (3) Write A1H to the CBnCTL0 register, and select the reception mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).
- (4) The CBnSTR.CBnTSP bit is set to 1 by performing a dummy read of the CBnRX register, and reception is started.
- (5) When reception is started, output the serial clock to the SCKBn pin, and capture the receive data of the SIBn pin in synchronization with the serial clock.
- (6) When reception of the transfer data length set with the CBnCTL2 register is completed, stop the serial clock output and data capturing, generate the reception completion interrupt request signal (INTCBnR) at the last edge of the serial clock, and clear the CBnTSP bit to 0.
- (7) To continue reception, read the CBnRX register with the CBnCTL0.CBnSCE bit = 1 remained after the INTCBnR signal is generated.
- (8) To read the CBnRX register without starting the next reception, write the CBnSCE bit = 0.
- (9) Read the CBnRX register.
- (10) To end reception, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnRXE bit = 0.

Remark n = 0, 1

13.5.3 Single transfer mode (master mode, transmission/reception mode)

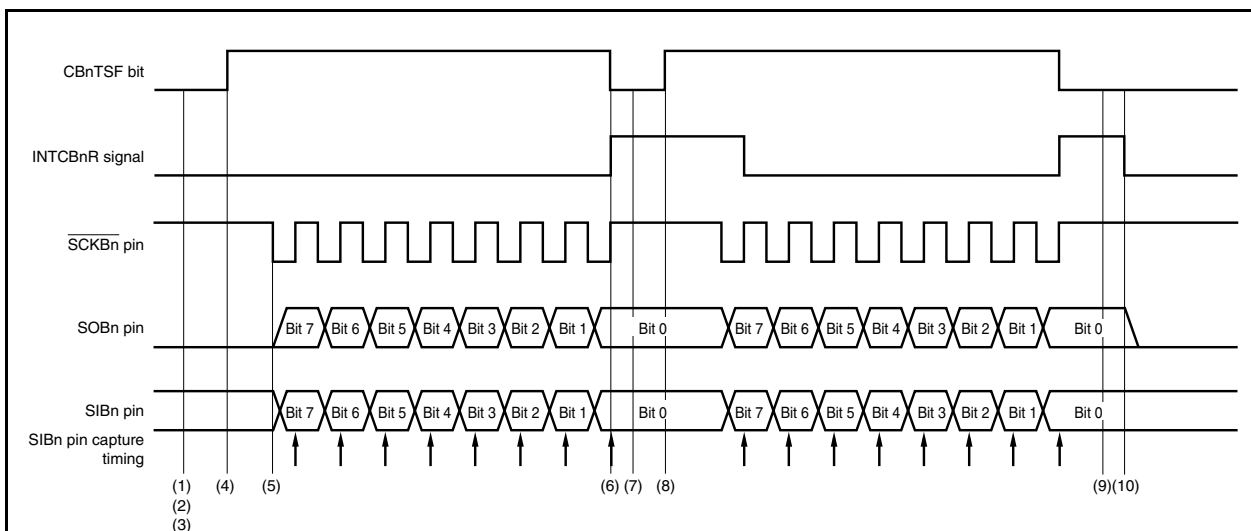
MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CLK}) = $f_{\text{XX}}/2$ (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow



- Remarks**
1. The broken lines indicate the hardware processing.
 2. The numbers in this figure correspond to the processing numbers in (2) **Operation timing**.
 3. $n = 0, 1$

(2) Operation timing



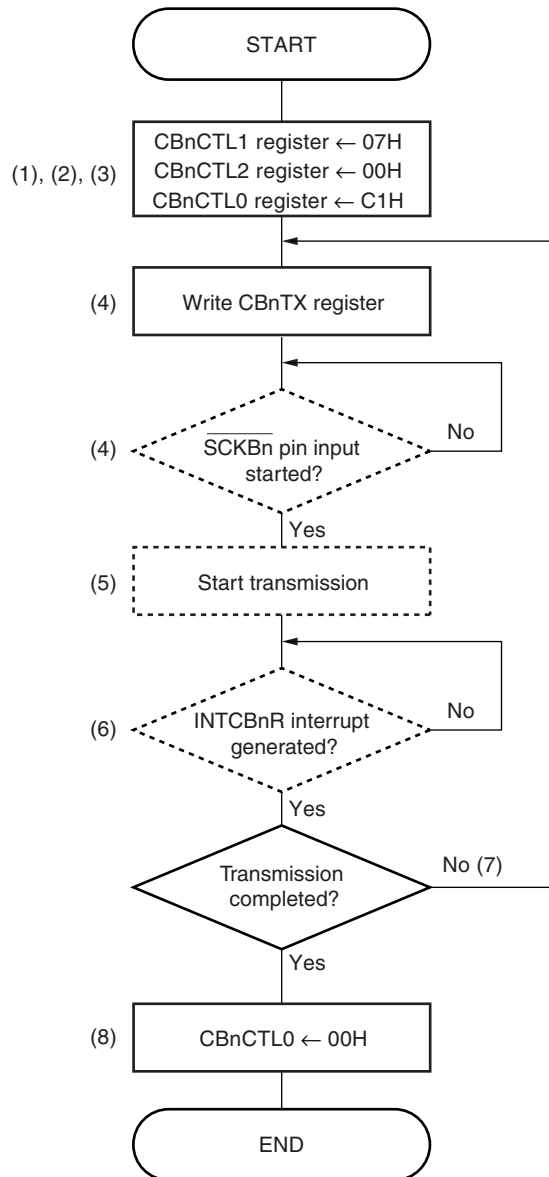
- (1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CLK}) = $f_{\text{xx}}/2$, and master mode.
- (2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.
- (3) Write E1H to the CBnCTL0 register, and select the transmission/reception mode and MSB first at the same time as enabling the operation of the communication clock (f_{CLK}).
- (4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and transmission/reception is started.
- (5) When transmission/reception is started, output the serial clock to the SCKBn pin, output the transmit data to the SOBn pin in synchronization with the serial clock, and capture the receive data of the SIBn pin.
- (6) When transmission/reception of the transfer data length set with the CBnCTL2 register is completed, stop the serial clock output, transmit data output, and data capturing, generate the reception completion interrupt request signal (INTCBnR) at the last edge of the serial clock, and clear the CBnTSF bit to 0.
- (7) Read the CBnRX register.
- (8) To continue transmission/reception, write the transmit data to the CBnTX register again.
- (9) Read the CBnRX register.
- (10) To end transmission/reception, write the CBnCTL0.CBnPWR bit = 0, the CBnCTL0.CBnTXE bit = 0, and the CBnCTL0.CBnRXE bit = 0.

Remark n = 0, 1

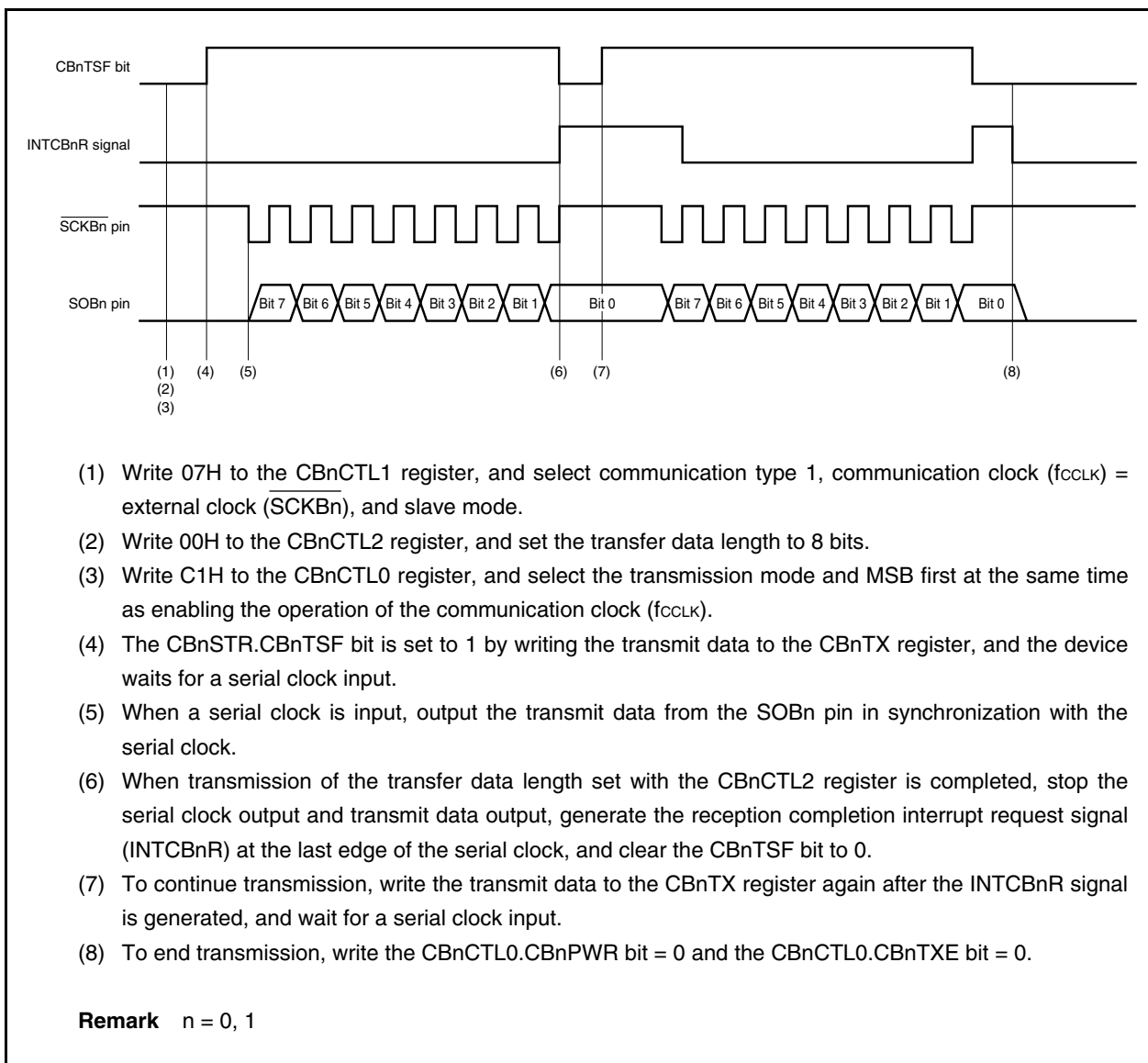
13.5.4 Single transfer mode (slave mode, transmission mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CLK}) = external clock (SCKBn) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow



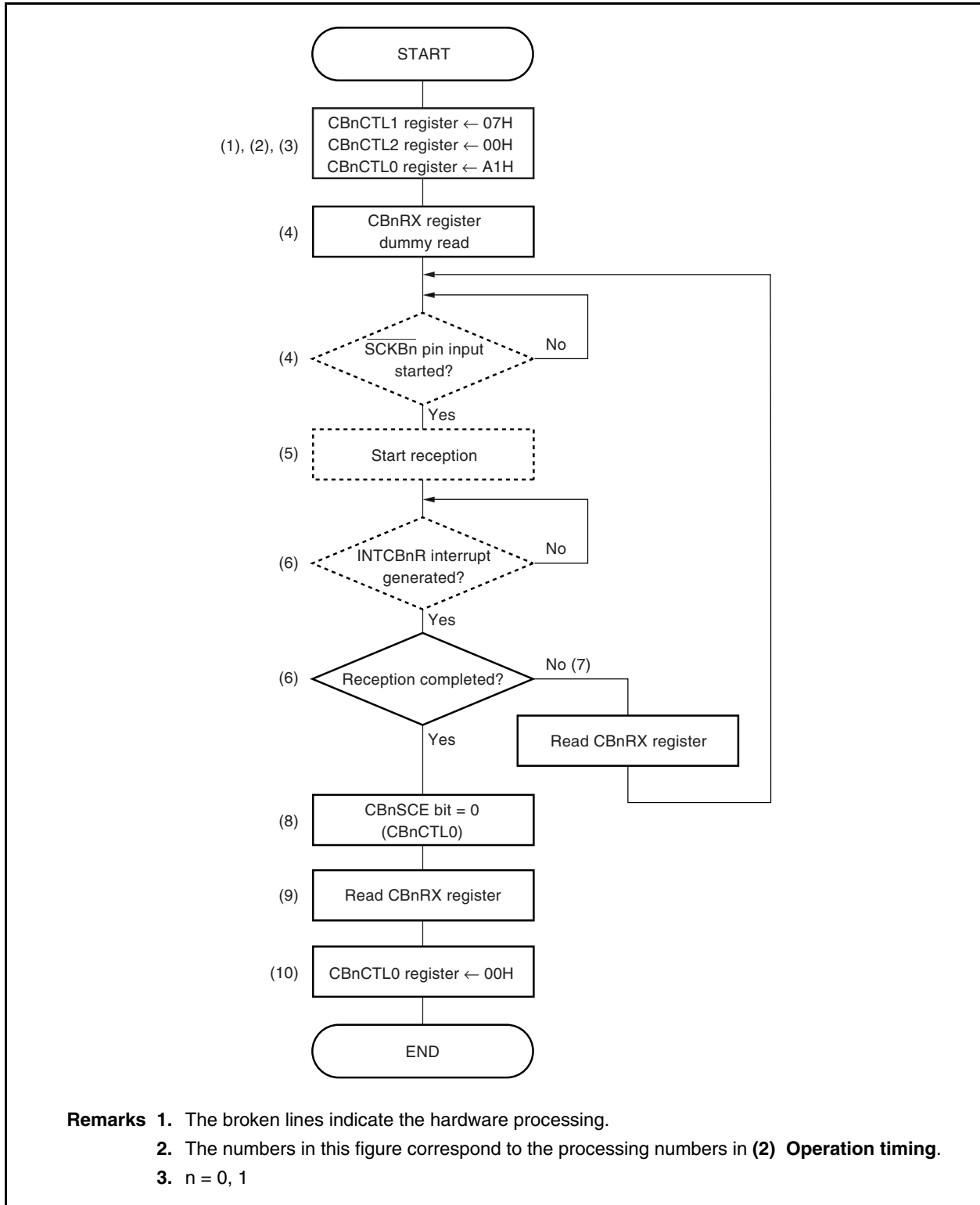
- Remarks**
1. The broken lines indicate the hardware processing.
 2. The numbers in this figure correspond to the processing numbers in (2) **Operation timing**.
 3. n = 0, 1

(2) Operation timing

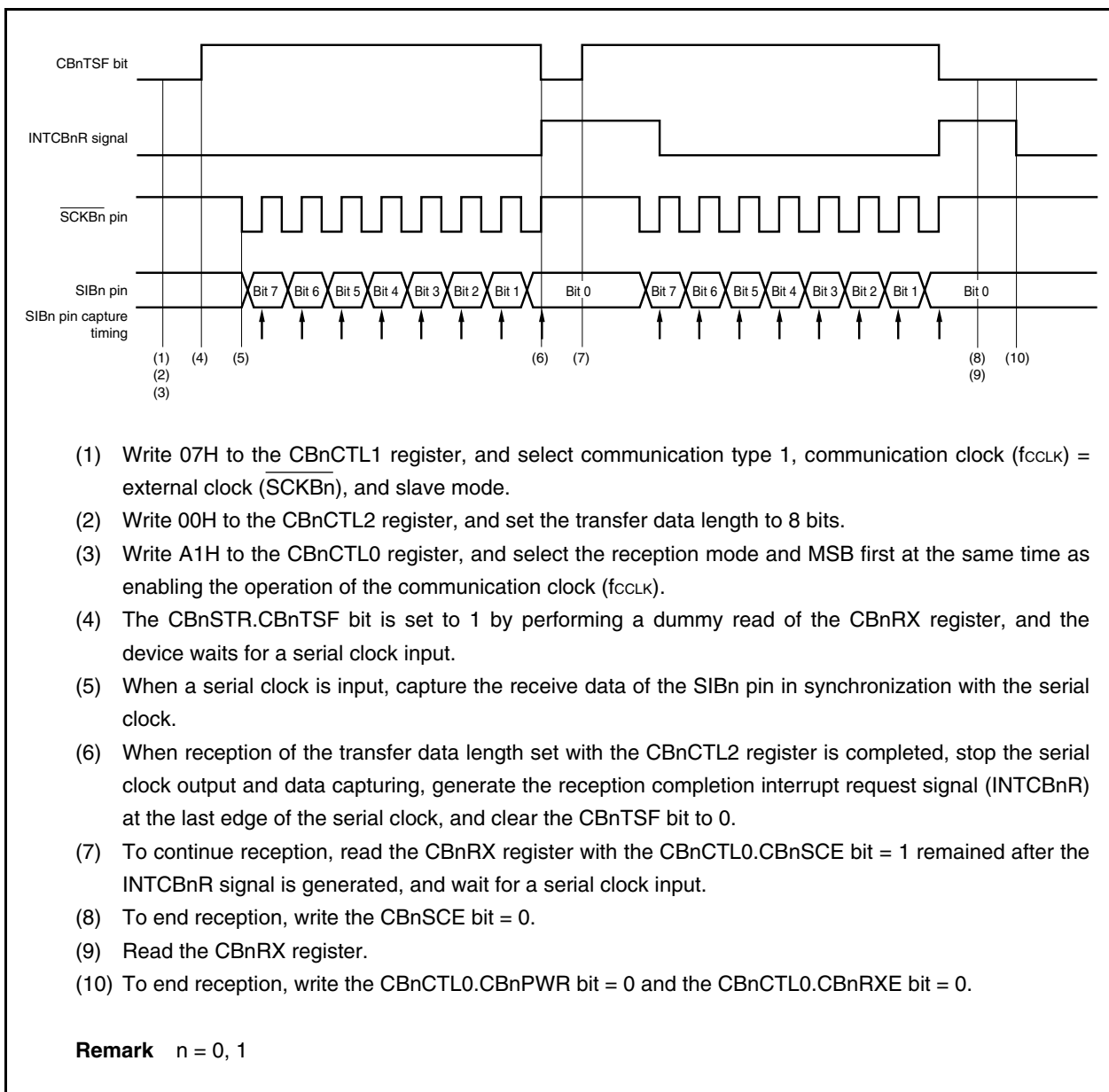
13.5.5 Single transfer mode (slave mode, reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = external clock (SCKBn) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow



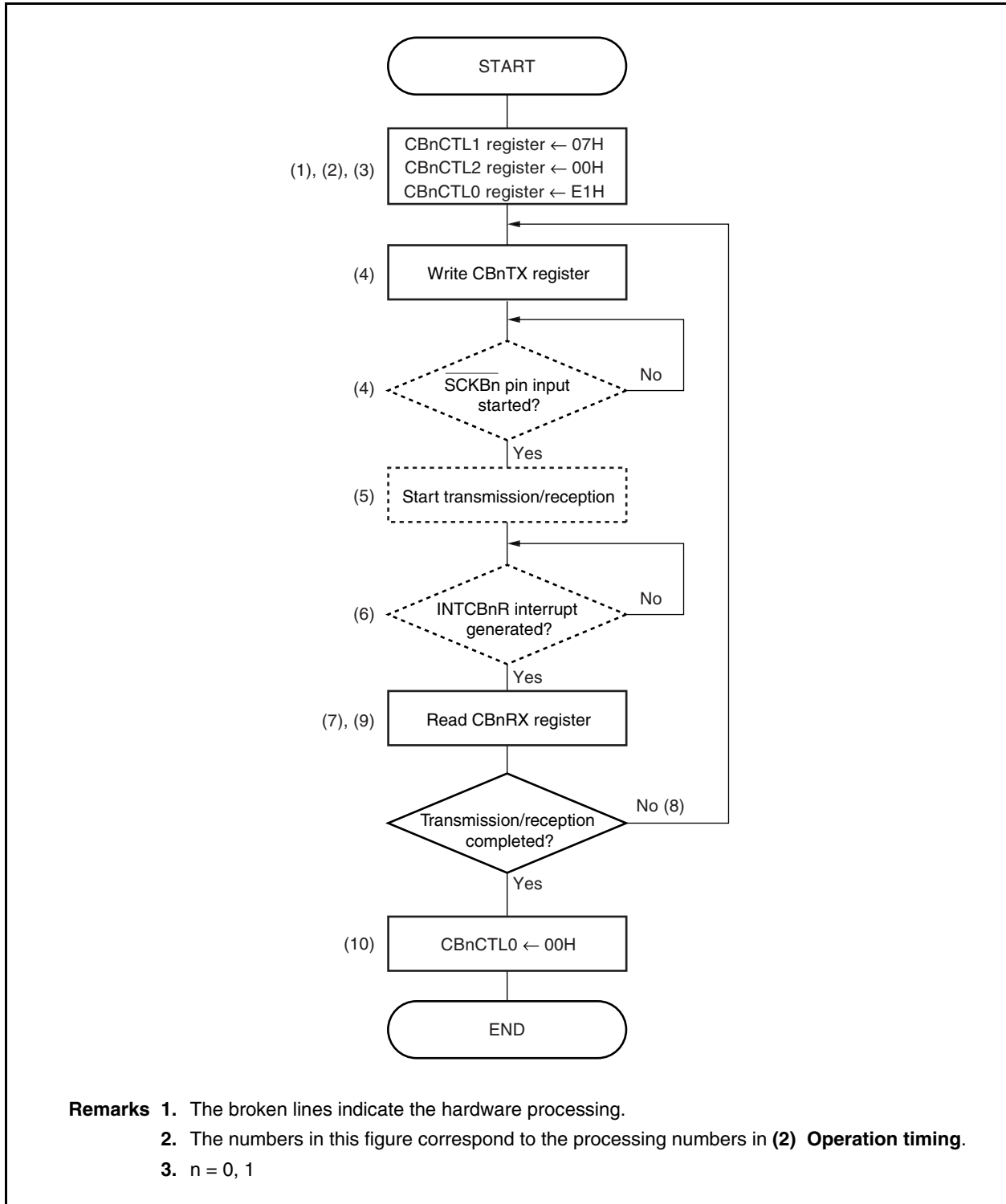
- Remarks**
1. The broken lines indicate the hardware processing.
 2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.
 3. n = 0, 1

(2) Operation timing

13.5.6 Single transfer mode (slave mode, transmission/reception mode)

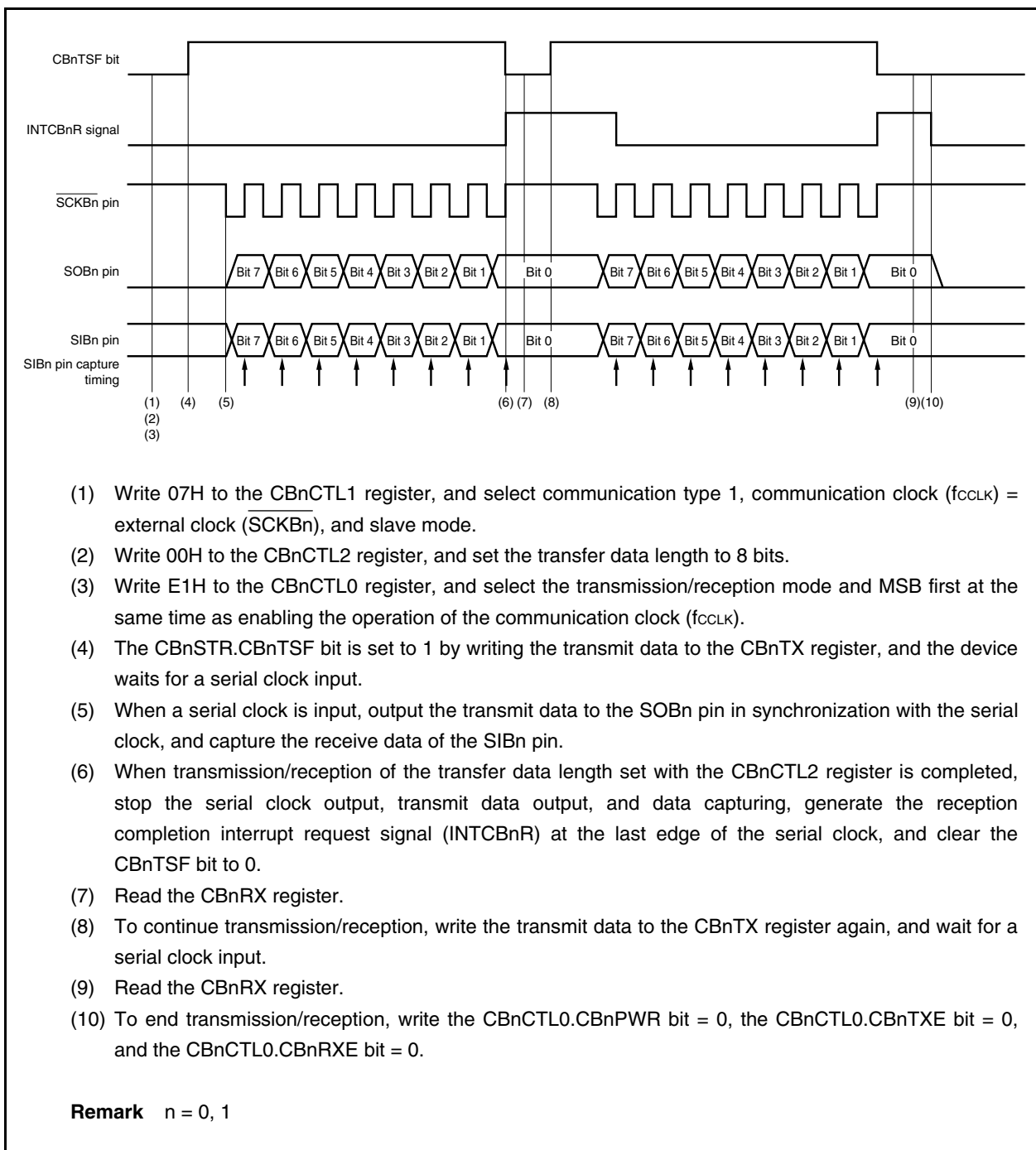
MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = external clock (SCKBn) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow



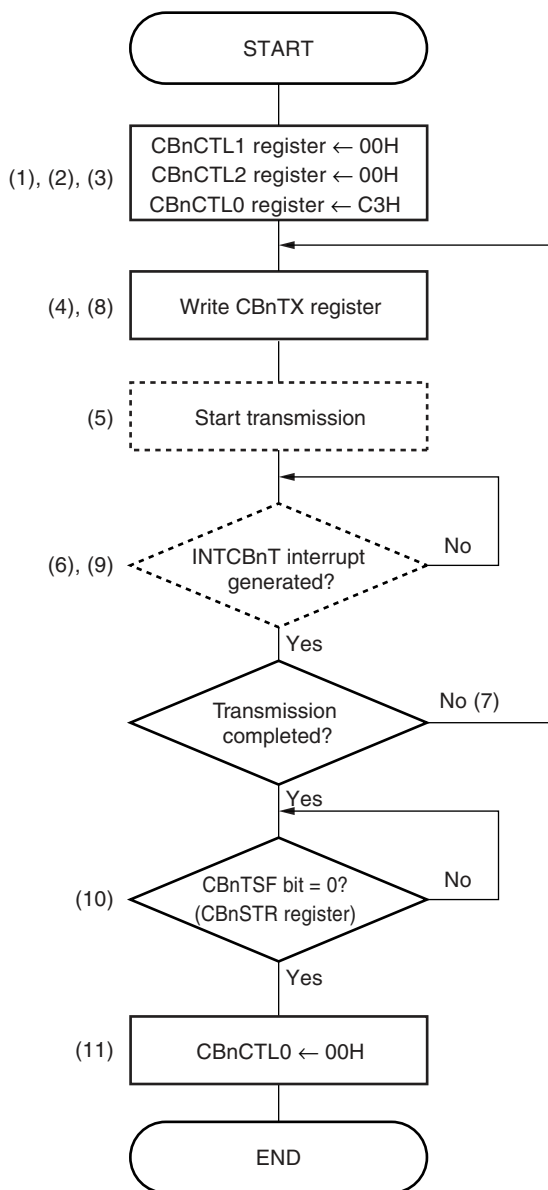
- Remarks**
1. The broken lines indicate the hardware processing.
 2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.
 3. n = 0, 1

(2) Operation timing



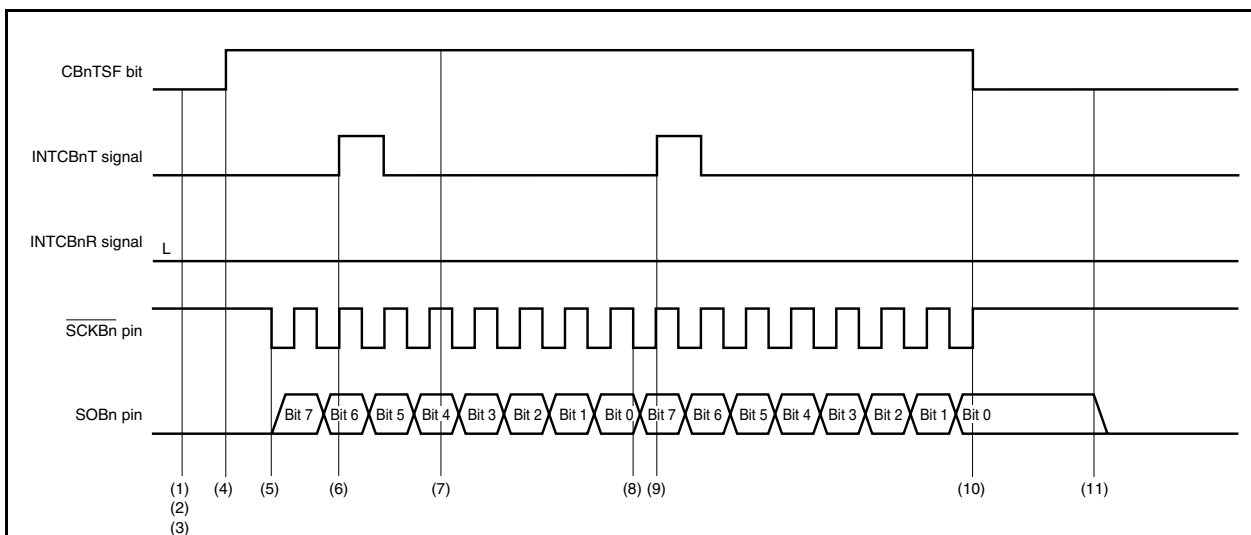
13.5.7 Continuous transfer mode (master mode, transmission mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CLK}) = $f_{\text{xx}}/2$ (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow

- Remarks**
1. The broken lines indicate the hardware processing.
 2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.
 3. $n = 0, 1$

(2) Operation timing



- (1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CLK}) = $f_{\text{xx}}/2$, and master mode.
- (2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.
- (3) Write C3H to the CBnCTL0 register, and select the transmission mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CLK}).
- (4) The CBnSTR.CBnTnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and transmission is started.
- (5) When transmission is started, output the serial clock to the $\overline{\text{SCKn}}$ pin, and output the transmit data from the SOBn pin in synchronization with the serial clock.
- (6) When transfer of the transmit data from the CBnTX register to the shift register is completed and writing to the CBnTX register is enabled, the transmission enable interrupt request signal (INTCnT) is generated.
- (7) To continue transmission, write the transmit data to the CBnTX register again after the INTCnT signal is generated.
- (8) When a new transmit data is written to the CBnTX register before communication completion, the next communication is started following communication completion.
- (9) The transfer of the transmit data from the CBnTX register to the shift register is completed and the INTCnT signal is generated. To end continuous transmission with the current transmission, do not write to the CBnTX register.
- (10) When the next transmit data is not written to the CBnTX register before transfer completion, stop the serial clock output to the $\overline{\text{SCKn}}$ pin after transfer completion, and clear the CBnTnTSF bit to 0.
- (11) To release the transmission enable status, write the CBnCTL0.CBnPWnR bit = 0 and the CBnCTL0.CBnTXE bit = 0 after checking that the CBnTnTSF bit = 0.

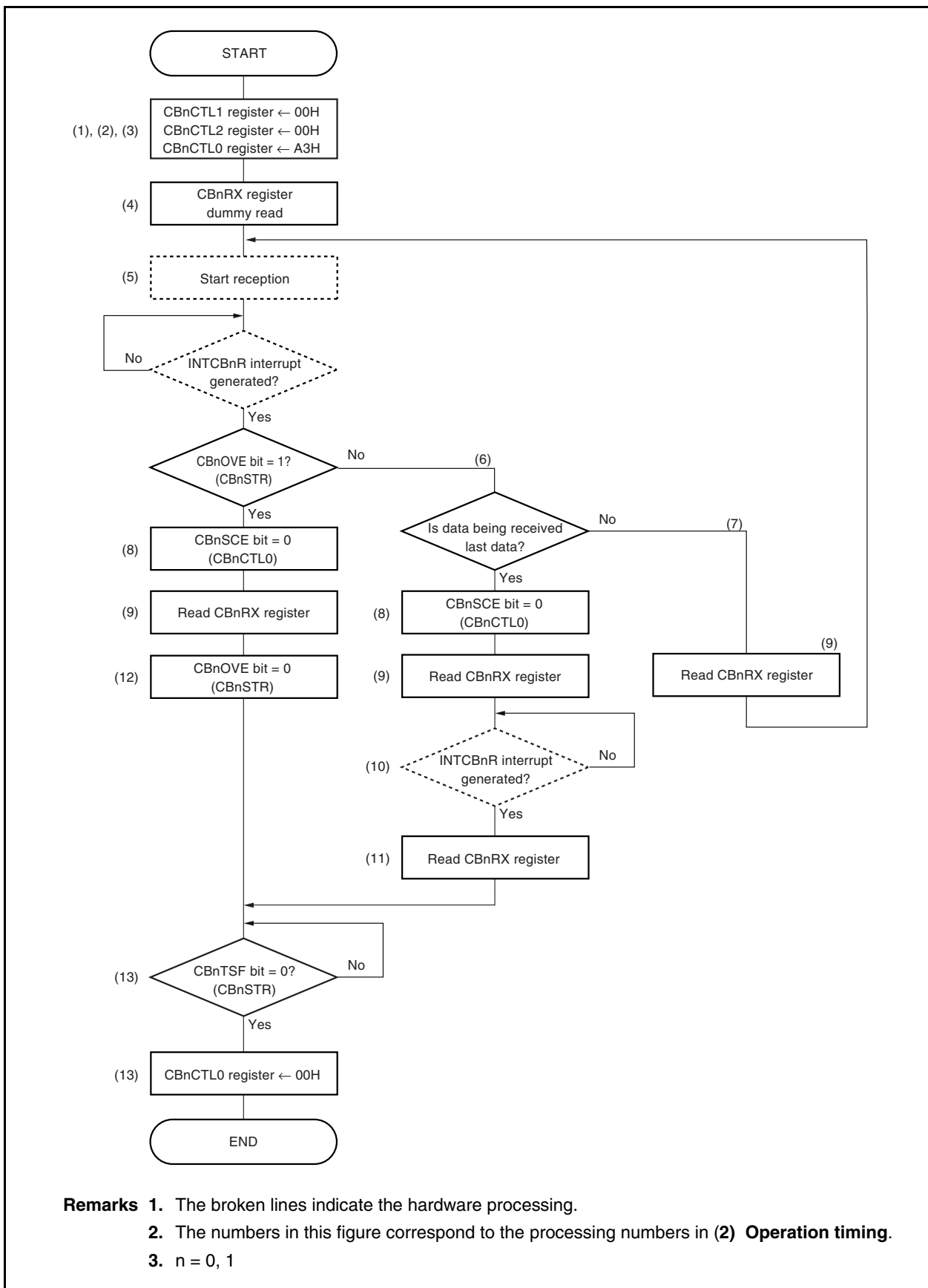
Caution In continuous transmission mode, the reception completion interrupt request signal (INTCnR) is not generated.

Remark $n = 0, 1$

13.5.8 Continuous transfer mode (master mode, reception mode)

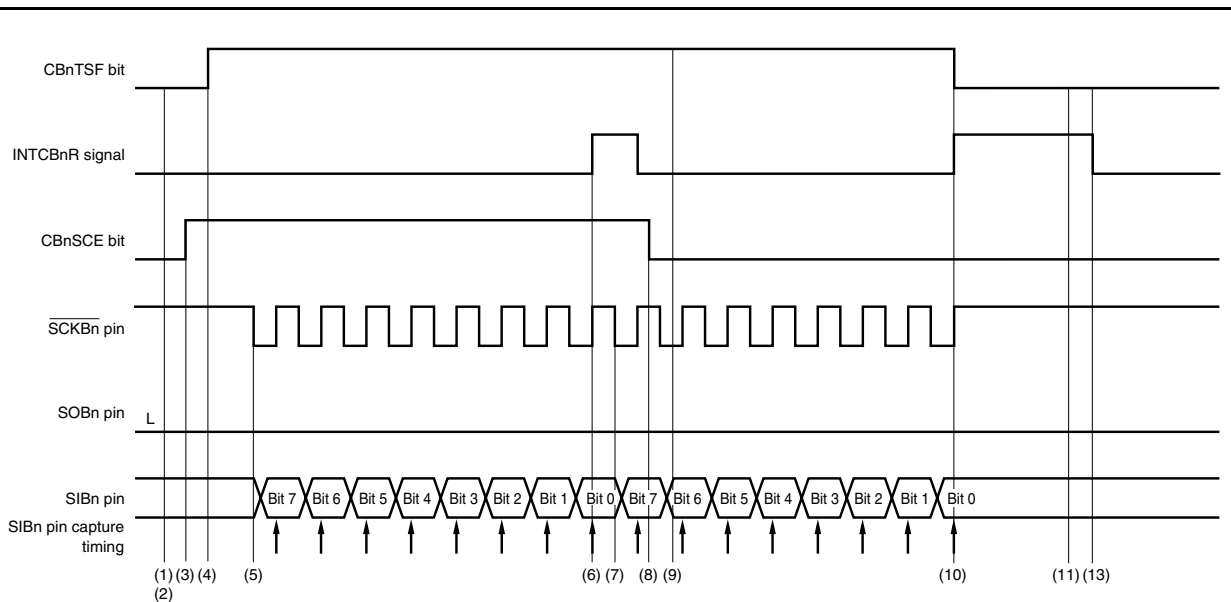
MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CLK}) = $f_{\text{XX}}/2$ (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow



- Remarks**
1. The broken lines indicate the hardware processing.
 2. The numbers in this figure correspond to the processing numbers in (2) Operation timing.
 3. n = 0, 1

(2) Operation timing



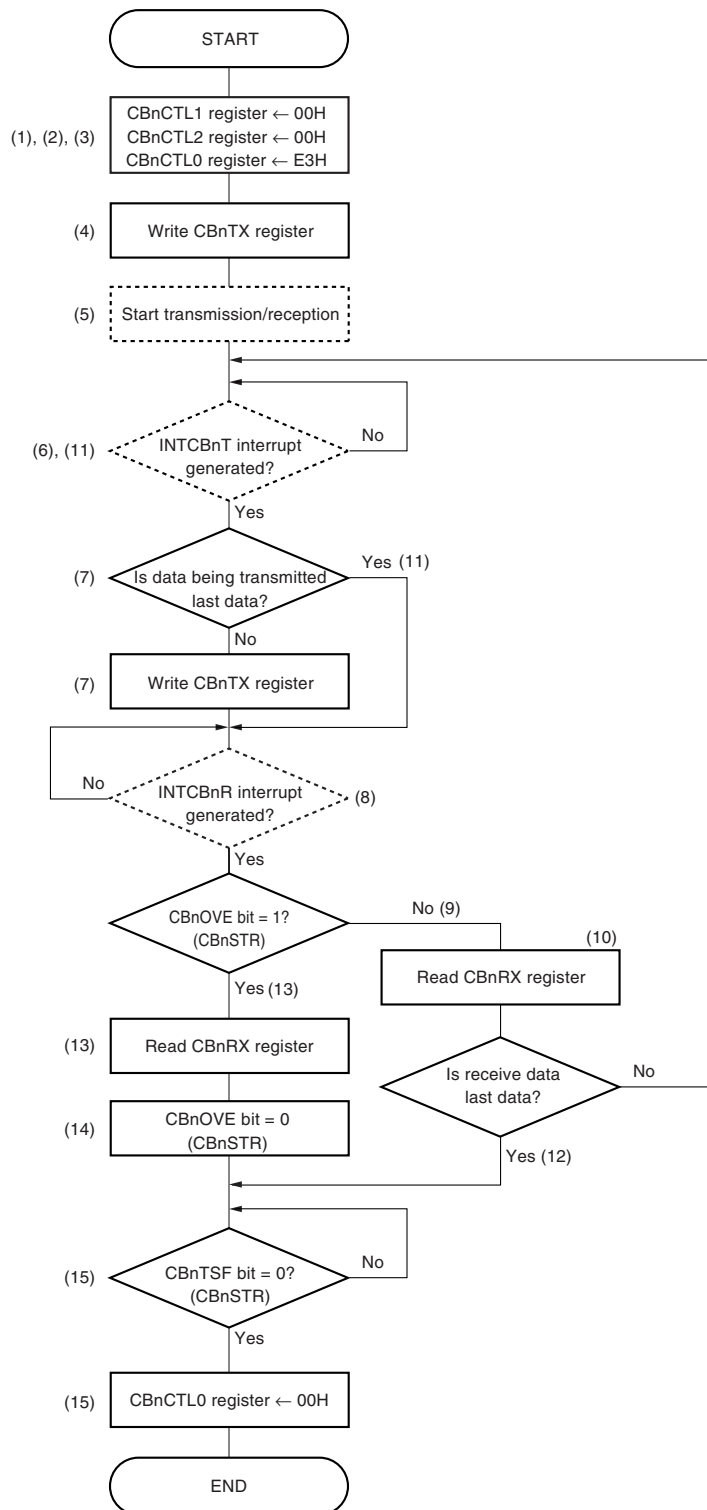
- (1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CLK}) = $f_{\text{xx}}/2$, and master mode.
- (2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.
- (3) Write A3H to the CBnCTL0 register, and select the reception mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CLK}).
- (4) The CBnSTR.CBnTSM bit is set to 1 by performing a dummy read of the CBnRX register, and reception is started.
- (5) When reception is started, output the serial clock to the $\overline{\text{SCKBn}}$ pin, and capture the receive data of the SIBn pin in synchronization with the serial clock.
- (6) When reception is completed, the reception completion interrupt request signal (INTCBnR) is generated, and reading of the CBnRX register is enabled.
- (7) When the CBnCTL0.CBnSCE bit = 1 upon communication completion, the next communication is started following communication completion.
- (8) To end continuous reception with the current reception, write the CBnSCE bit = 0.
- (9) Read the CBnRX register.
- (10) When reception is completed, the INTCBnR signal is generated, and reading of the CBnRX register is enabled. When the CBnSCE bit = 0 is set before communication completion, stop the serial clock output to the $\overline{\text{SCKBn}}$ pin, and clear the CBnTSM bit to 0, to end the receive operation.
- (11) Read the CBnRX register.
- (12) If an overrun error occurs, write the CBnSTR.CBnOVE bit = 0, and clear the error flag.
- (13) To release the reception enable status, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnRXE bit = 0 after checking that the CBnTSM bit = 0.

Remark n = 0, 1

13.5.9 Continuous transfer mode (master mode, transmission/reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CLK}) = $f_{\text{XX}}/2$ (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

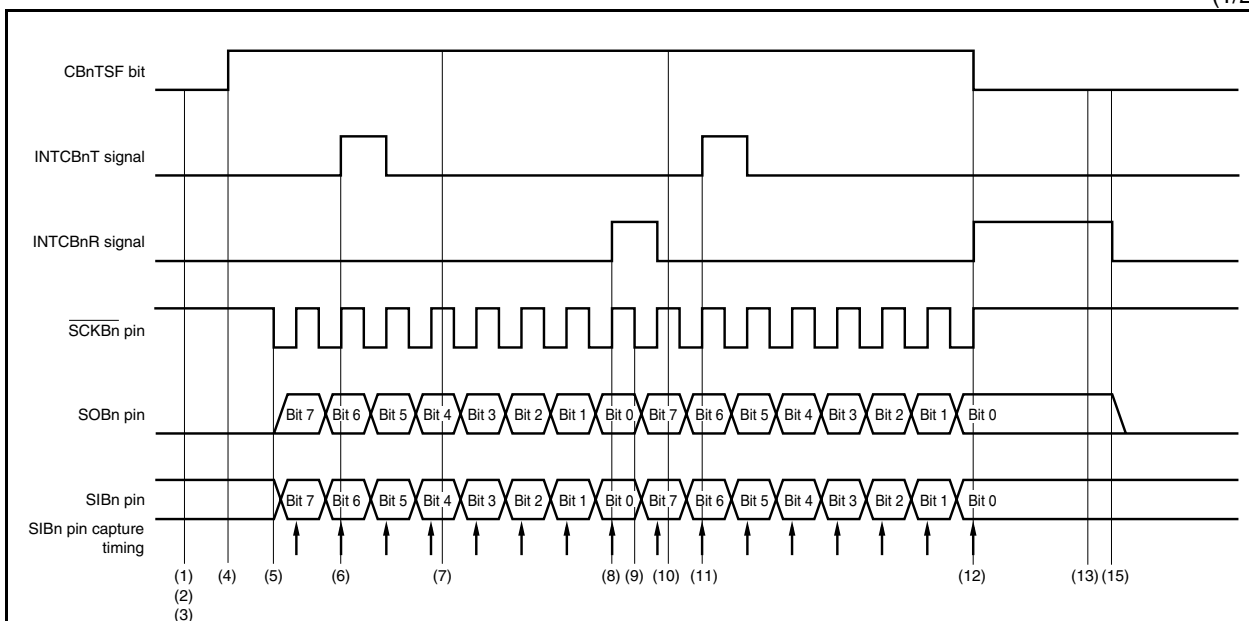
(1) Operation flow



- Remarks**
1. The broken lines indicate the hardware processing.
 2. The numbers in this figure correspond to the processing numbers in (2) Operation timing.
 3. n = 0, 1

(2) Operation timing

(1/2)



- (1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CLK}) = $f_{\text{xx}}/2$, and master mode.
- (2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.
- (3) Write E3H to the CBnCTL0 register, and select the transmission/reception mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CLK}).
- (4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and transmission/reception is started.
- (5) When transmission/reception is started, output the serial clock to the SCKBn pin, output the transmit data to the SOBn pin in synchronization with the serial clock, and capture the receive data of the SIBn pin.
- (6) When transfer of the transmit data from the CBnTX register to the shift register is completed and writing to the CBnTX register is enabled, the transmission enable interrupt request signal (INTCBnT) is generated.
- (7) To continue transmission/reception, write the transmit data to the CBnTX register again after the INTCBnT signal is generated.
- (8) When one transmission/reception is completed, the reception completion interrupt request signal (INTCBnR) is generated, and reading of the CBnRX register is enabled.
- (9) When a new transmit data is written to the CBnTX register before communication completion, the next communication is started following communication completion.
- (10) Read the CBnRX register.

Remark $n = 0, 1$

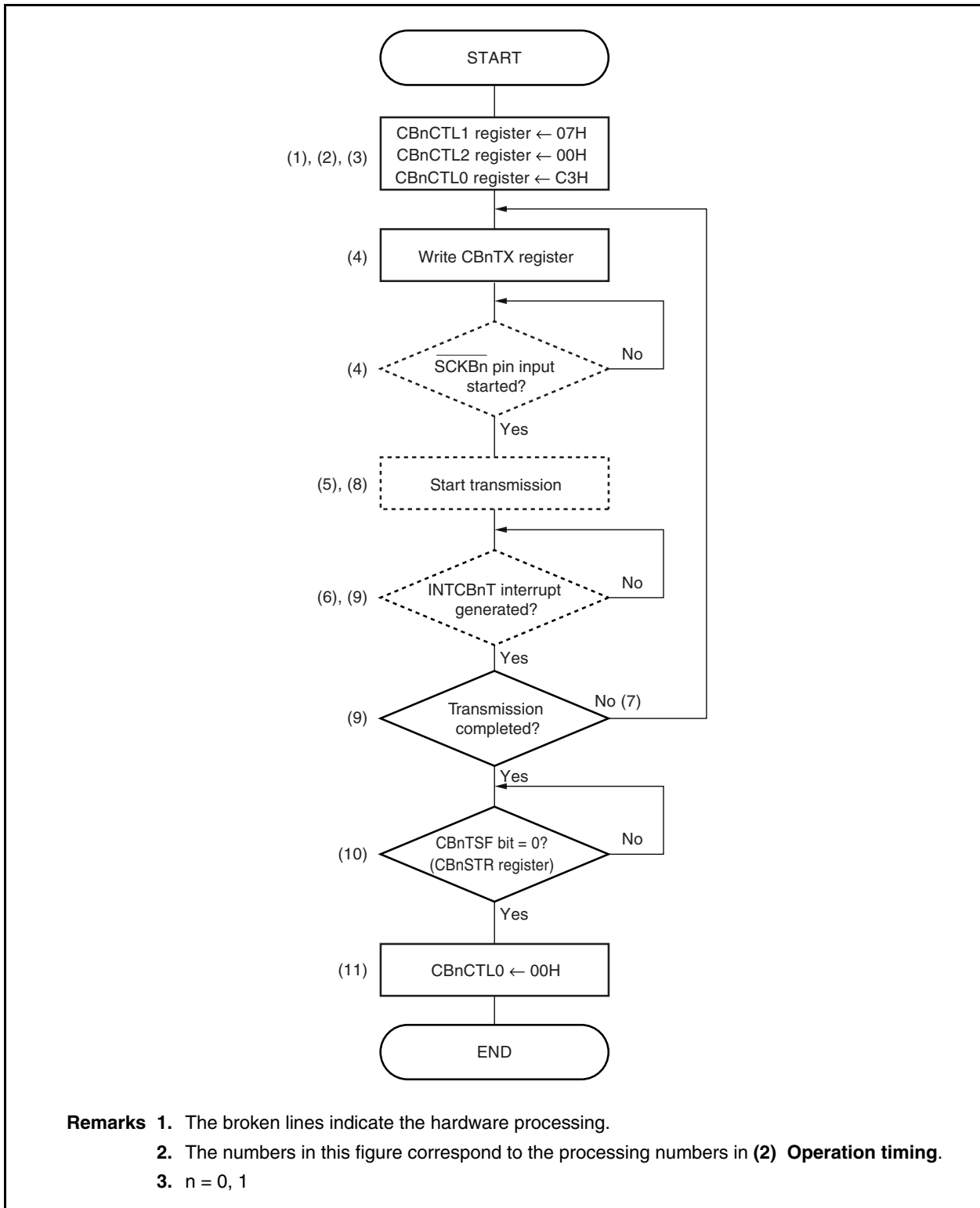
- (11) The transfer of the transmit data from the CBnTX register to the shift register is completed and the INTCBnT signal is generated. To end continuous transmission/reception with the current transmission/reception, do not write to the CBnTX register.
- (12) When the next transmit data is not written to the CBnTX register before transfer completion, stop the serial clock output to the $\overline{\text{SCKBn}}$ pin after transfer completion, and clear the CBnTSF bit to 0.
- (13) When the reception error interrupt request signal (INTCBnR) is generated, read the CBnRX register.
- (14) If an overrun error occurs, write the CBnSTR.CBnOVE bit = 0, and clear the error flag.
- (15) To release the transmission/reception enable status, write the CBnCTL0.CBnPWR bit = 0, the CBnCTL0.CBnTXE bit = 0, and the CBnCTL0.CBnRXE bit = 0 after checking that the CBnTSF bit = 0.

Remark n = 0, 1

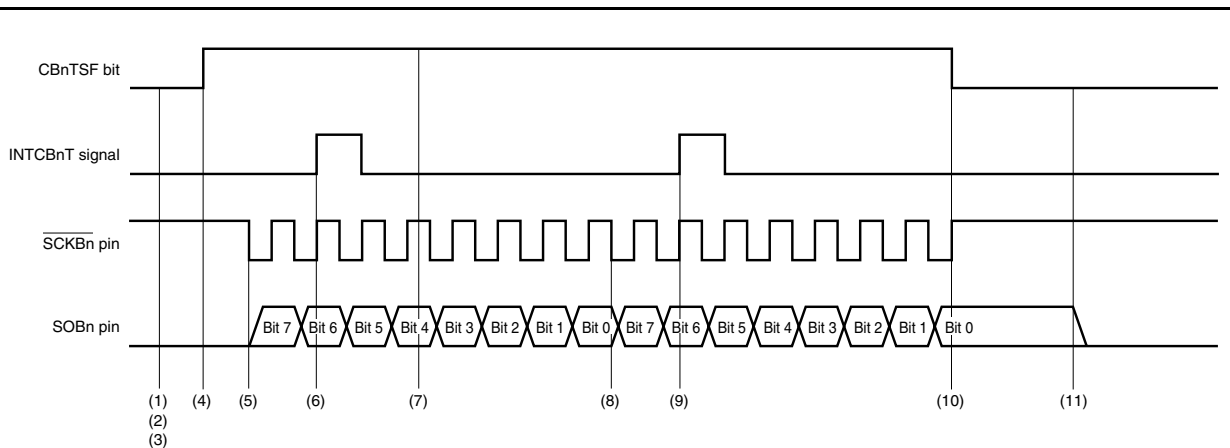
13.5.10 Continuous transfer mode (slave mode, transmission mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CLK}) = external clock (SCKBn) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow



- Remarks**
1. The broken lines indicate the hardware processing.
 2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.
 3. n = 0, 1

(2) Operation timing

- (1) Write 07H to the `CBnCTL1` register, and select communication type 1, communication clock (f_{CCLK}) = external clock (`SCKBn`), and slave mode.
- (2) Write 00H to the `CBnCTL2` register, and set the transfer data length to 8 bits.
- (3) Write C3H to the `CBnCTL0` register, and select the transmission mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).
- (4) The `CBnSTR.CBnTSF` bit is set to 1 by writing the transmit data to the `CBnTX` register, and the device waits for a serial clock input.
- (5) When a serial clock is input, output the transmit data from the `SOBn` pin in synchronization with the serial clock.
- (6) When transfer of the transmit data from the `CBnTX` register to the shift register is completed and writing to the `CBnTX` register is enabled, the transmission enable interrupt request signal (`INTCBnT`) is generated.
- (7) To continue transmission, write the transmit data to the `CBnTX` register again after the `INTCBnT` signal is generated.
- (8) When a serial clock is input following completion of the transmission of the transfer data length set with the `CBnCTL2` register, continuous transmission is started.
- (9) When transfer of the transmit data from the `CBnTX` register to the shift register is completed and writing to the `CBnTX` register is enabled, the `INTCBnT` signal is generated. To end continuous transmission with the current transmission, do not write to the `CBnTX` register.
- (10) When the clock of the transfer data length set with the `CBnCTL2` register is input without writing to the `CBnTX` register, clear the `CBnTSF` bit to 0 to end transmission.
- (11) To release the transmission enable status, write the `CBnCTL0.CBnPWR` bit = 0 and the `CBnCTL0.CBnTXE` bit = 0 after checking that the `CBnTSF` bit = 0.

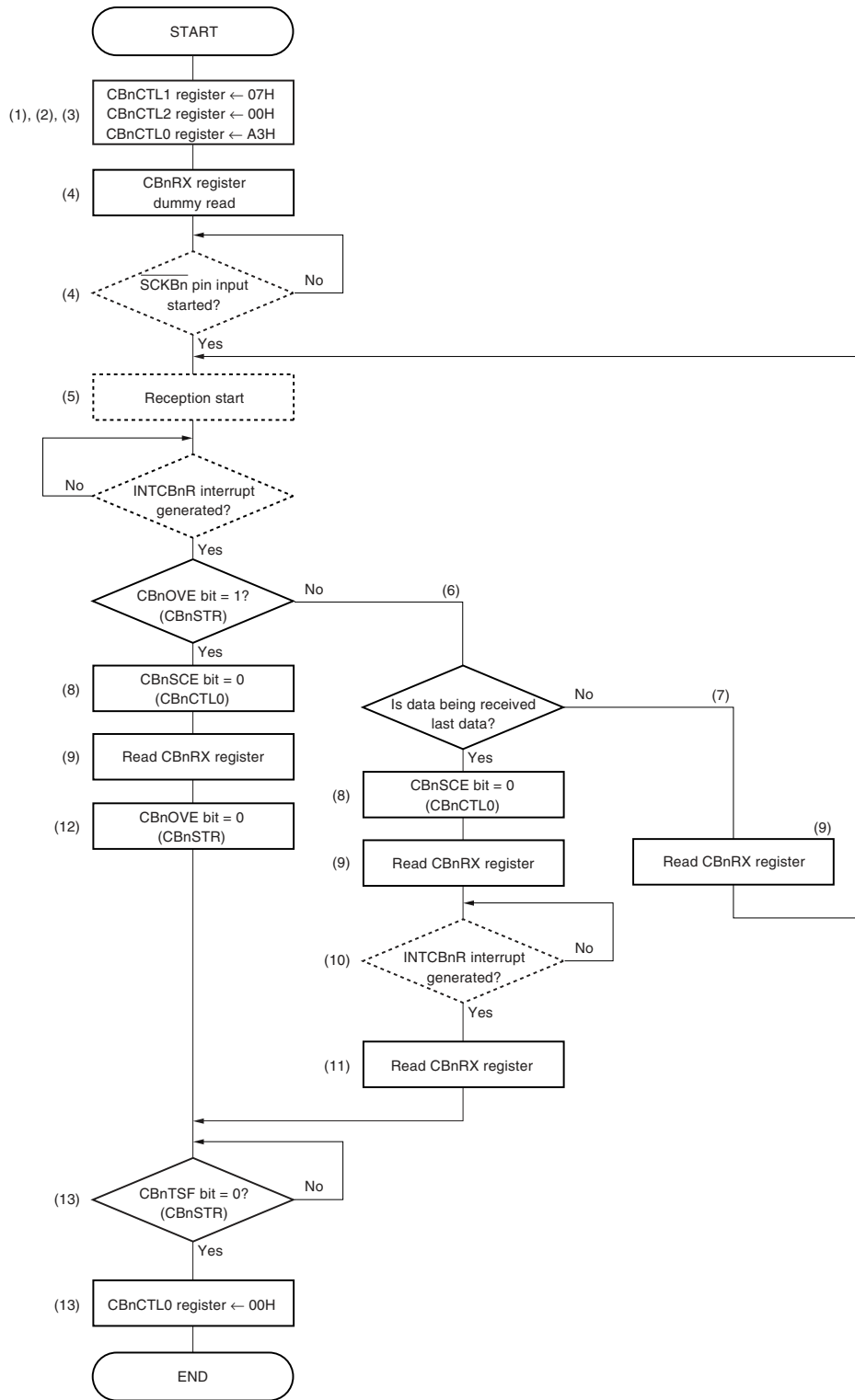
Caution In continuous transmission mode, the reception completion interrupt request signal (`INTCBnR`) is not generated.

Remark $n = 0, 1$

13.5.11 Continuous transfer mode (slave mode, reception mode)

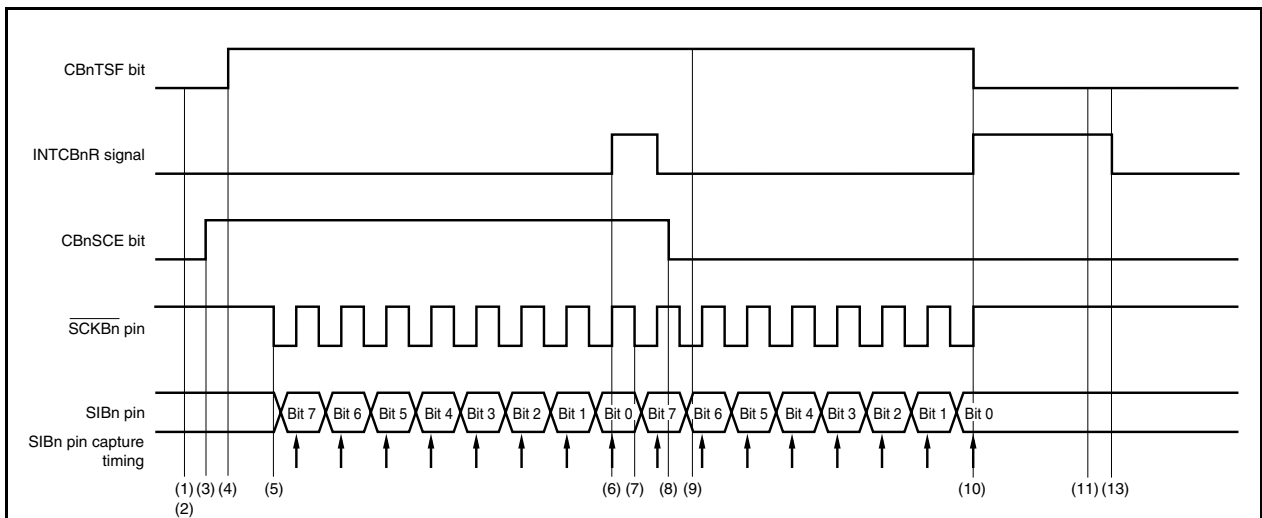
MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CLK}) = external clock ($\overline{\text{SCKBn}}$) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow



- Remarks**
1. The broken lines indicate the hardware processing.
 2. The numbers in this figure correspond to the processing numbers in (2) **Operation timing**.
 3. n = 0, 1

(2) Operation timing



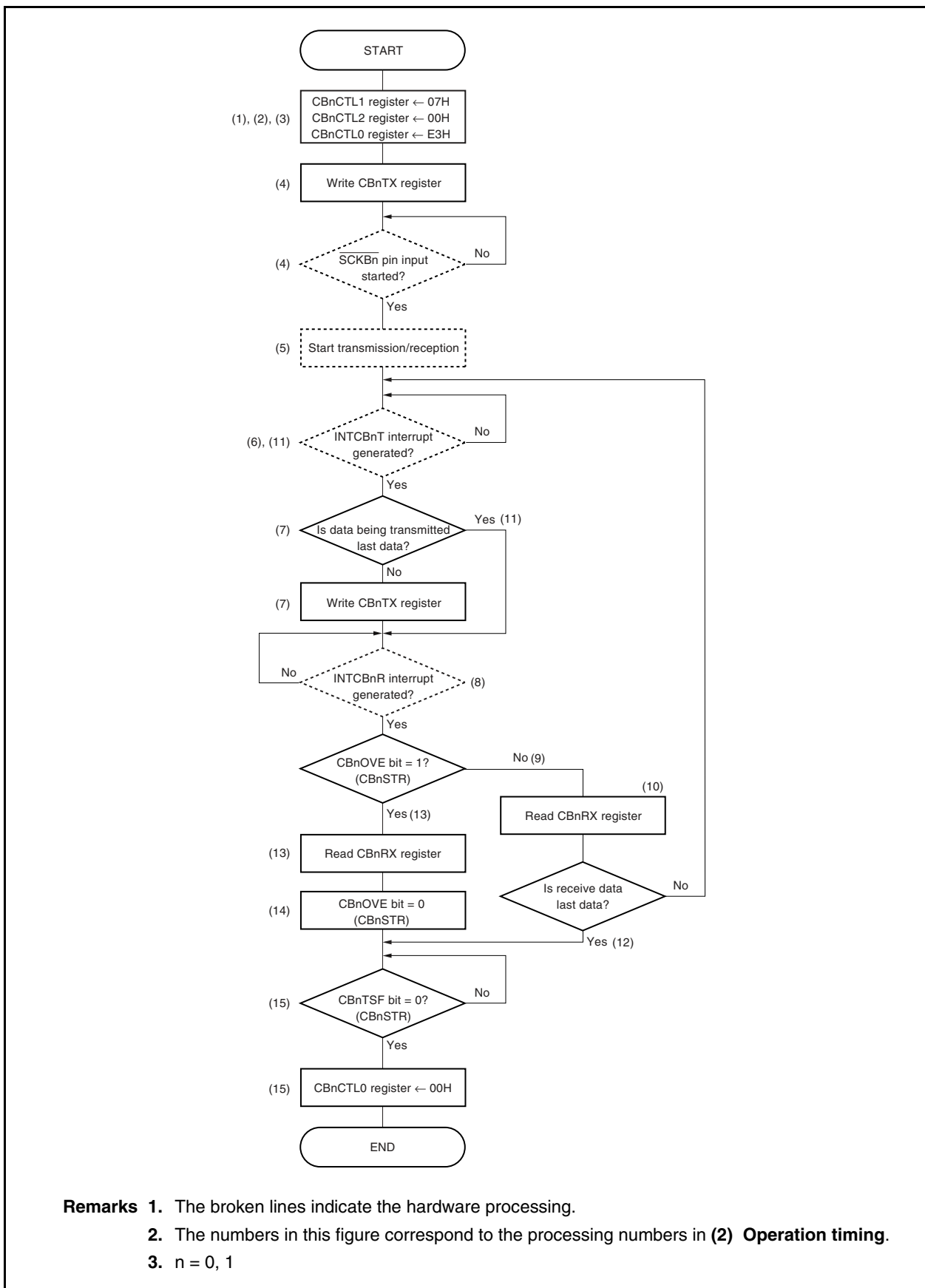
- (1) Write 07H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CLK}) = external clock ($\overline{\text{SCKBn}}$), and slave mode.
- (2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.
- (3) Write A3H to the CBnCTL0 register, and select the reception mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CLK}).
- (4) The CBnSTR.CBnTSF bit is set to 1 by performing a dummy read of the CBnRX register, and the device waits for a serial clock input.
- (5) When a serial clock is input, capture the receive data of the SIBn pin in synchronization with the serial clock.
- (6) When reception is completed, the reception completion interrupt request signal (INTCBnR) is generated, and reading of the CBnRX register is enabled.
- (7) When a serial clock is input in the CBnCTL0.CBnSCE bit = 1 status, continuous reception is started.
- (8) To end continuous reception with the current reception, write the CBnSCE bit = 0.
- (9) Read the CBnRX register.
- (10) When reception is completed, the INTCBnR signal is generated, and reading of the CBnRX register is enabled. When the CBnSCE bit = 0 is set before communication completion, clear the CBnTSF bit to 0 to end the receive operation.
- (11) Read the CBnRX register.
- (12) If an overrun error occurs, write the CBnSTR.CBnOVE bit = 0, and clear the error flag.
- (13) To release the reception enable status, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnRXE bit = 0 after checking that the CBnTSF bit = 0.

Remark n = 0, 1

13.5.12 Continuous transfer mode (slave mode, transmission/reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = external clock ($\overline{\text{SCKBn}}$) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

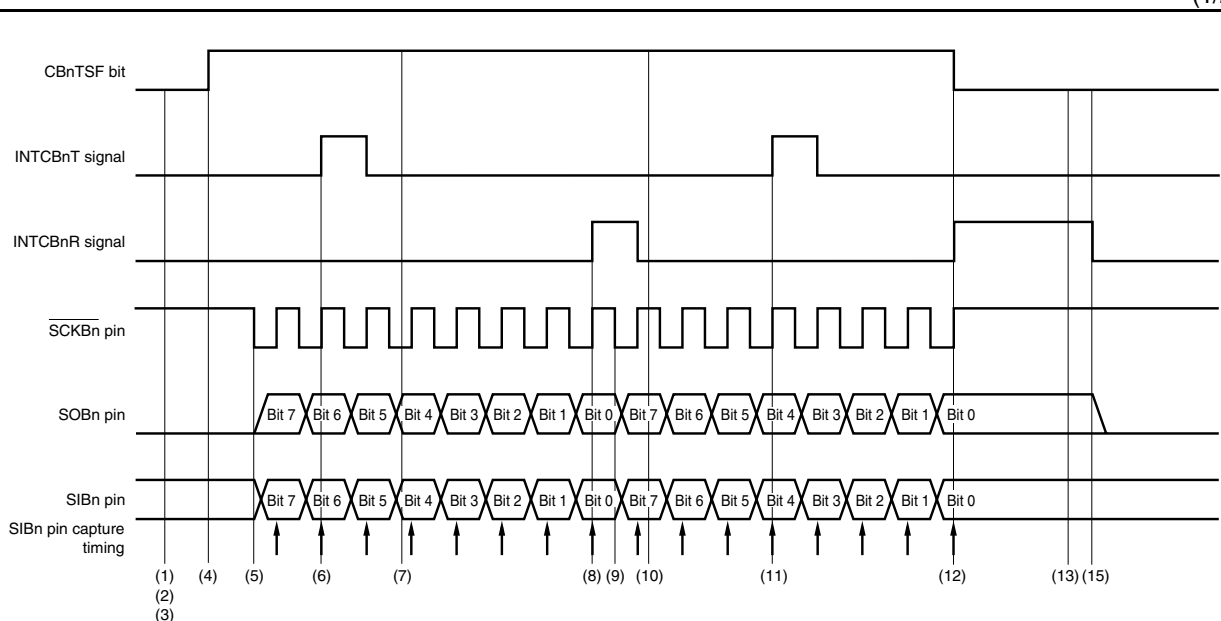
(1) Operation flow



- Remarks**
1. The broken lines indicate the hardware processing.
 2. The numbers in this figure correspond to the processing numbers in (2) **Operation timing**.
 3. n = 0, 1

(2) Operation timing

(1/2)



- (1) Write 07H to the $\overline{\text{CBnCTL1}}$ register, and select communication type 1, communication clock (f_{CLK}) = external clock (SCKBn), and slave mode.
- (2) Write 00H to the $\overline{\text{CBnCTL2}}$ register, and set the transfer data length to 8 bits.
- (3) Write E3H to the $\overline{\text{CBnCTL0}}$ register, and select the transmission/reception mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CLK}).
- (4) The $\overline{\text{CBnSTR.CBnTSF}}$ bit is set to 1 by writing the transmit data to the $\overline{\text{CBnTX}}$ register, and the device waits for a serial clock input.
- (5) When a serial clock is input, output the transmit data to the SOBn pin in synchronization with the serial clock, and capture the receive data of the SIBn pin.
- (6) When transfer of the transmit data from the $\overline{\text{CBnTX}}$ register to the shift register is completed and writing to the $\overline{\text{CBnTX}}$ register is enabled, the transmission enable interrupt request signal (INTCBnT) is generated.
- (7) To continue transmission, write the transmit data to the $\overline{\text{CBnTX}}$ register again after the INTCBnT signal is generated.
- (8) When reception of the transfer data length set with the $\overline{\text{CBnCTL2}}$ register is completed, the reception completion interrupt request signal (INTCBnR) is generated, and reading of the $\overline{\text{CBnRX}}$ register is enabled.
- (9) When a serial clock is input continuously, continuous transmission/reception is started.
- (10) Read the $\overline{\text{CBnRX}}$ register.
- (11) When transfer of the transmit data from the $\overline{\text{CBnTX}}$ register to the shift register is completed and writing to the $\overline{\text{CBnTX}}$ register is enabled, the INTCBnT signal is generated. To end continuous transmission/reception with the current transmission/reception, do not write to the $\overline{\text{CBnTX}}$ register.

Remark $n = 0, 1$

- (12) When the clock of the transfer data length set with the CBnCTL2 register is input without writing to the CBnTX register, the INTCBnR signal is generated. Clear the CBnTSF bit to 0 to end transmission/reception.
- (13) When the INTCBnR signal is generated, read the CBnRX register.
- (14) If an overrun error occurs, write the CBnSTR.CBnOVE bit = 0, and clear the error flag.
- (15) To release the transmission/reception enable status, write the CBnCTL0.CBnPWR bit = 0, the CBnCTL0.CBnTXE bit = 0, and the CBnCTL0.CBnRXE bit = 0 after checking that the CBnTSF bit = 0.

Remark n = 0, 1

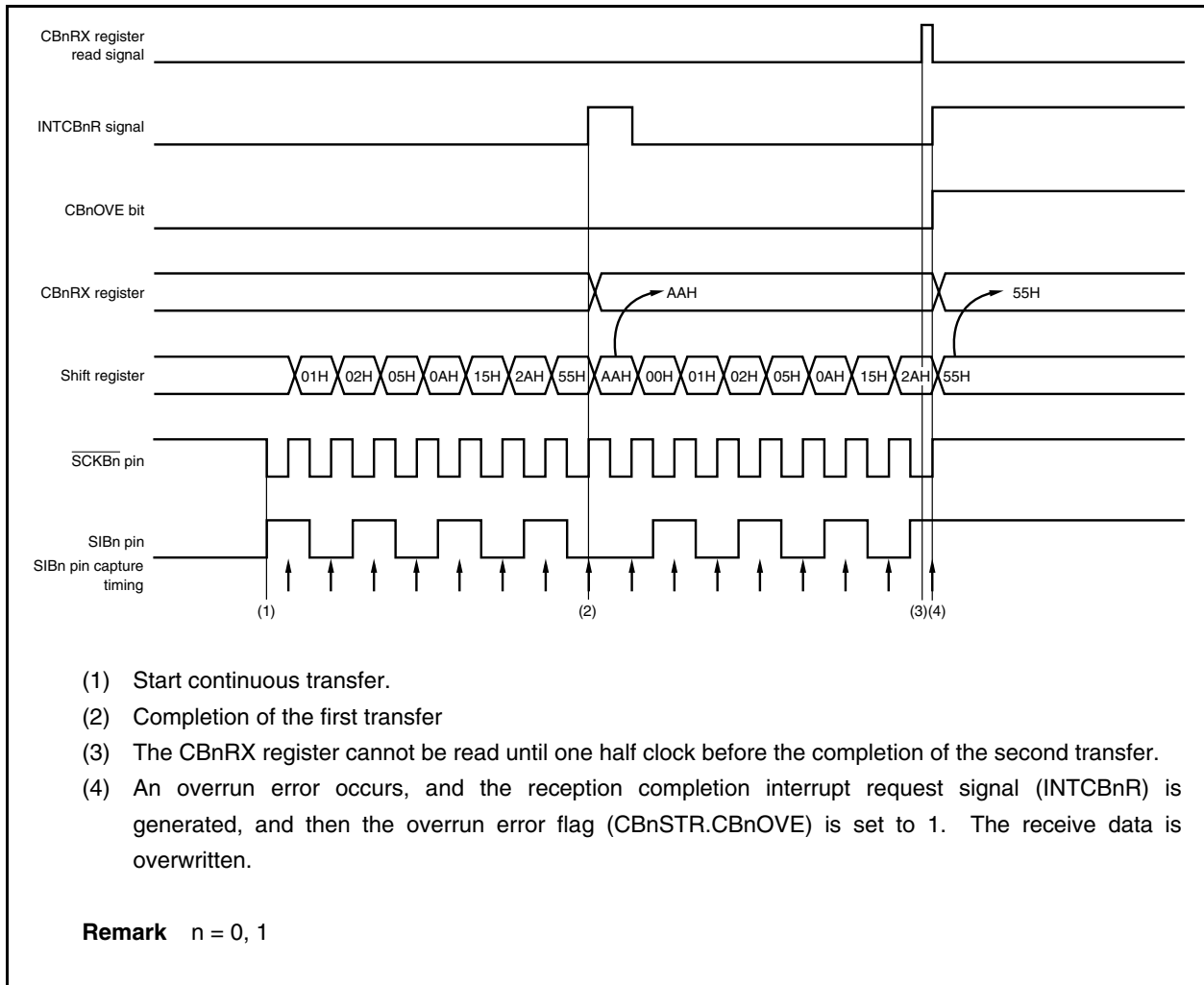
13.5.13 Reception error

When transfer is performed with reception enabled (CBnCTL0.CBnRXE bit = 1) in the continuous transfer mode, the reception completion interrupt request signal (INTCBnR) is generated again when the next receive operation is completed before the CBnRX register is read after the INTCBnR signal is generated, and the overrun error flag (CBnSTR.CBnOVE) is set to 1.

Even if an overrun error has occurred, the previous receive data is lost since the CBnRX register is updated. Even if a reception error has occurred, the INTCBnR signal is generated again upon the next reception completion if the CBnRX register is not read.

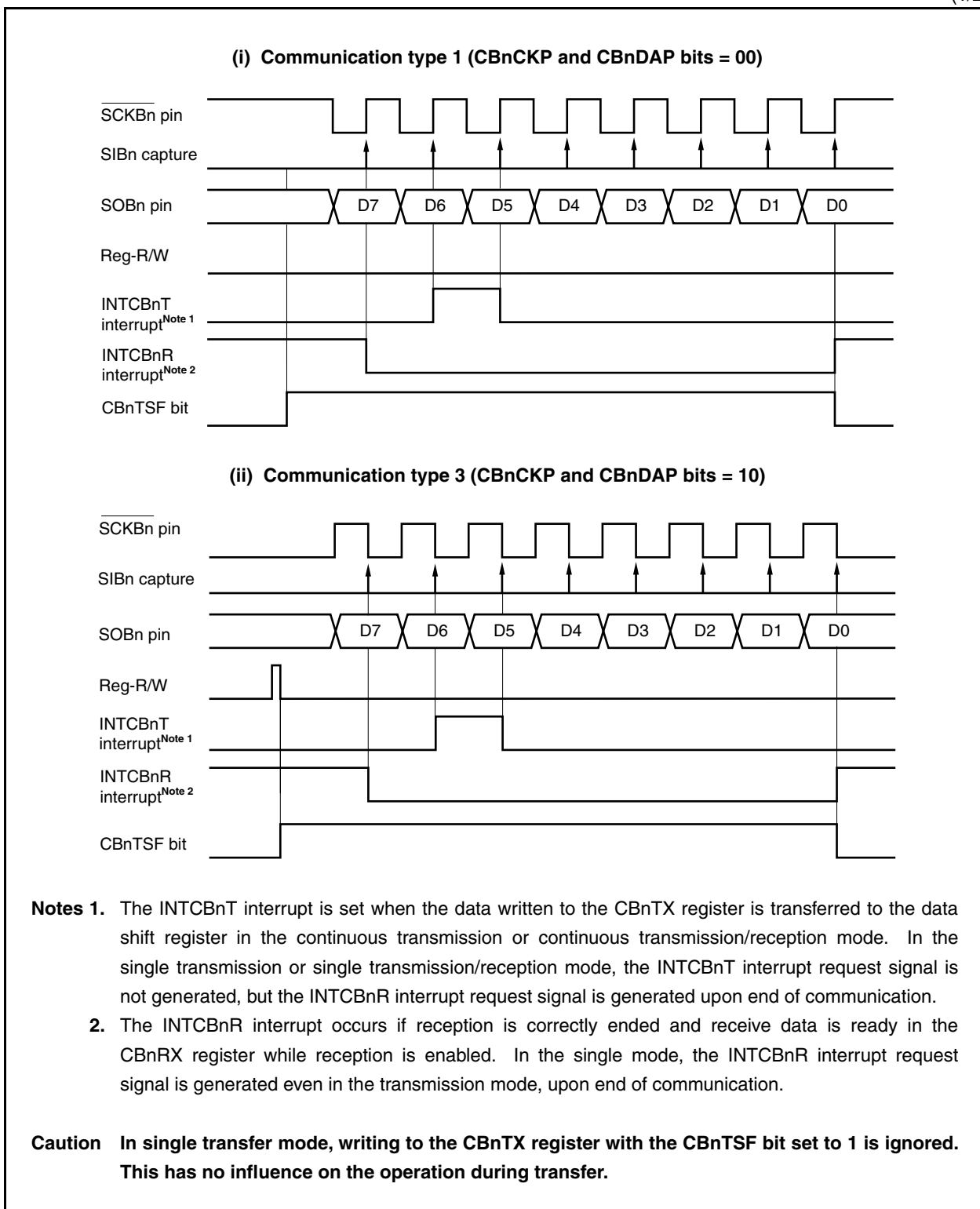
To avoid an overrun error, complete reading the CBnRX register until one half clock before sampling the last bit of the next receive data from the INTCBnR signal generation.

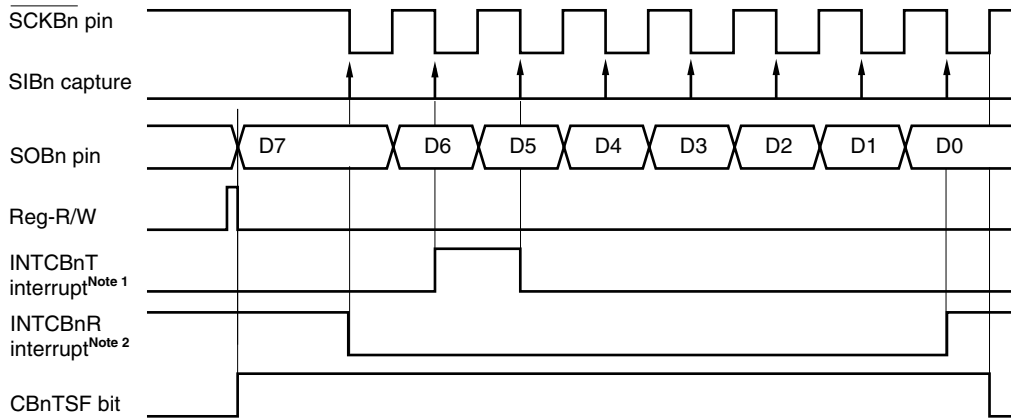
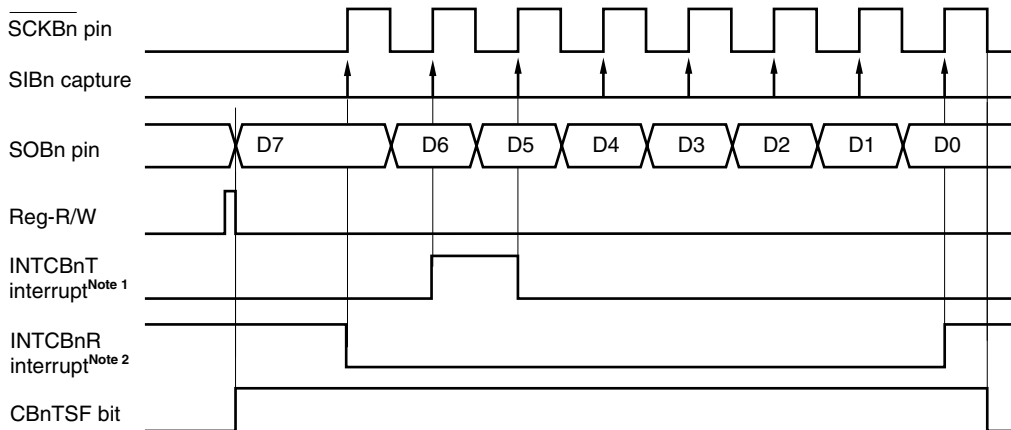
(1) Operation timing



13.5.14 Clock timing

(1/2)



(iii) Communication type 2 (CBnCKP and CBnDAP bits = 01)**(iv) Communication type 4 (CBnCKP and CBnDAP bits = 11)**

Notes 1. The INTCBnT interrupt is set when the data written to the CBnTX register is transferred to the data shift register in the continuous transmission or continuous transmission/reception modes. In the single transmission or single transmission/reception modes, the INTCBnT interrupt request signal is not generated, but the INTCBnR interrupt request signal is generated upon end of communication.

2. The INTCBnR interrupt occurs if reception is correctly ended and receive data is ready in the CBnRX register while reception is enabled. In the single mode, the INTCBnR interrupt request signal is generated even in the transmission mode, upon end of communication.

Caution In single transfer mode, writing to the CBnTX register with the CBnTSF bit set to 1 is ignored. This has no influence on the operation during transfer.

13.6 Output Pin Status with Operation Disabled

(1) $\overline{\text{SCKBn}}$ pin

When CSIBn operation is disabled (CBnCTL0.CBnPWR bit = 0), the $\overline{\text{SCKBn}}$ pin output status is as follows.

| CBnCKS2 | CBnCKS1 | CBnCKS0 | CBnCKP | $\overline{\text{SCKBn}}$ Pin Output |
|------------------|---------|---------|--------|--------------------------------------|
| 1 | 1 | 1 | × | High impedance |
| Other than above | | | 0 | Fixed to high level |
| Other than above | | | 1 | 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
3. ×: don't care

(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 | CBnTX register value (MSB) |
| | | 1 | CBnTX register value (LSB) |

Remarks 1. The SOBn pin output changes when any one of the CBnCTL0.CBnTXE, CBnCTL0.CBnDIR bits, and CBnCTL1.CBnDAP bit is rewritten.

2. n = 0, 1
3. ×: don't care

13.7 Baud Rate Generator

The clock generated by the baud rate generator (prescaler 3) is supplied to the watch timer and CSIB0.

(1) Prescaler mode register 0 (PRSM0)

The PRSM0 register controls generation of the baud rate signal for CSIB.

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

Reset sets this register to 00H.

| | | | | | | | | |
|------------------|-----|-------------------|---|-------|---|---|--------|--------|
| After reset: 00H | R/W | Address: FFFF8B0H | | | | | | |
| | 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 | Count clock selection (f_{BGCS}) | | |
|--------|--------|--------------------------------------|-------------|-----------|
| | | | 5 MHz | 4 MHz |
| 0 | 0 | f_x | 200 ns | 250 ns |
| 0 | 1 | $f_x/2$ | 400 ns | 500 ns |
| 1 | 0 | $f_x/4$ | 800 ns | 1 μ s |
| 1 | 1 | $f_x/8$ | 1.6 μ s | 2 μ s |

Cautions

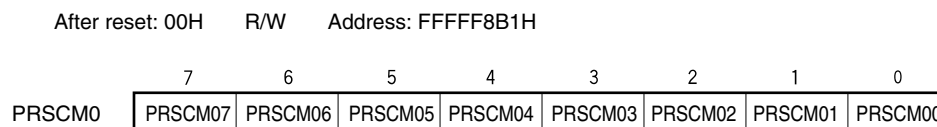
1. Do not rewrite the PRSM0 register while watch timer and CSIB0 are operating.
2. Set the PRSM0 register before setting the BGCE0 bit to 1.

(2) Prescaler compare register 0 (PRSCM0)

The PRSCM0 register is an 8-bit compare registers.

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

Reset sets this register to 00H.



- Cautions**
1. Do not rewrite the PRSCM0 register while watch timer and CSIB are operating.
 2. Set the PRSCM0 register before setting the PRSM0.BGCE0 bit to 1.

13.7.1 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.

$$f_{\text{BRG}} = \frac{f_{\text{xx}}}{2^{k+1} N}$$

<R>

Caution Set so that the f_{BRG} is 8 MHz or less.

Remark

- f_{BRG} : BRG count clock
- f_{xx} : Main clock oscillation frequency
- k : PRSM0 register setting value = 0 to 3
- N : PRSCM0 register setting value = 1 to 256
However, $N = 256$ only when PRSCM0 register is set to 00H.

13.8 Cautions

- (1) In regards to registers that are forbidden from being rewritten during operations (CBnCTL0.CBnPWR bit is 1), if rewriting has been carried out by mistake during operations, set the CBnCTL0.CBnPWR bit to 0 once, then initialize CSIBn.

Registers to which rewriting during operation are prohibited are shown below.

- CBnCTL0 register: CBnTXE, CBnRXE, CBnDIR, CBnTMS bits
- CBnCTL1 register: CBnCKP, CBnDAP, CBnCKS2 to CBnCKS0 bits
- CBnCTL2 register: CBnCL3 to CBnCL0 bits

- (2) In communication type 2 and 4 (CBnCTL1.CBnDAP bit = 1), the CBnSTR.CBnTSF bit is cleared half a $\overline{\text{SCKBn}}$ clock after occurrence of a reception complete interrupt (INTCBnR).

In the single transfer mode, writing the next transmit data is ignored during communication (CBnTSF bit = 1), and the next communication is not started. Also if reception-only communication (CBnCTL0.CBnTXE bit = 0, CBnCTL0.CBnRXE bit = 1) is set, the next communication is not started even if the receive data is read during communication (CBnTSF bit = 1).

Therefore, when using the single transfer mode with communication type 2 or 4 (CBnDAP bit = 1), pay particular attention to the following.

- To start the next transmission, confirm that CBnTSF bit = 0 and then write the transmit data to the CBnTX register.
- To perform the next reception continuously when reception-only communication (CBnTXE bit = 0, CBnRXE bit = 1) is set, confirm that CBnTSF bit = 0 and then read the CBnRX register.

Or, use the continuous transfer mode instead of the single transfer mode

Remark n = 0, 1

CHAPTER 14 INTERRUPT/EXCEPTION PROCESSING FUNCTION

The V850ES/HF2 is provided with a dedicated interrupt controller (INTC) for interrupt servicing and can process a total of 41 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 V850ES/HF2 can process interrupt request signals 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).

14.1 Features

- Interrupts
 - Non-maskable interrupts: 2 sources
 - Maskable interrupts: External: 8, Internal: 31 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 14-1.

Table 14-1. 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 | RESET pin input Reset input by internal source | RESET | 0000H | 00000000H | Undefined | – |
| 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 | POCLVI | 0080H | 00000080H | nextPC | LVIIIC |
| | | 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 |
| | | 17 | INTTP1OV | TMP1 overflow | TMP1 | 0190H | 00000190H | nextPC | TP1OVIC |
| | | 18 | INTTP1CC0 | TMP1 capture 0/compare 0 match | TMP1 | 01A0H | 000001AH | 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 | | |

Notes 1. For the restoring in the case of INTWDT2, see 14.2.2 (2) From INTWDT2 signal.

2. n = 0H to FH

Table 14-1. Interrupt Source List (2/2)

| Type | Classification | Default Priority | Name | Trigger | Generating Unit | Exception Code | Handler Address | Restored PC | Interrupt Control Register |
|----------|----------------|----------------------------|-----------|--|-----------------|----------------|-----------------|-------------|----------------------------|
| Maskable | Interrupt | 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 | CSIB0 reception completion | CSIB0 | 0230H | 00000230H | nextPC | CB0RIC |
| | | 28 | INTCB0T | CSIB0 consecutive transmission write enable | CSIB0 | 0240H | 00000240H | nextPC | CB0TIC |
| | | 29 | INTCB1R | CSIB1 reception completion | CSIB1 | 0250H | 00000250H | nextPC | CB1RIC |
| | | 30 | INTCB1T | CSIB1 consecutive transmission write enable | CSIB1 | 0260H | 00000260H | nextPC | CB1TIC |
| | | 31 | INTUA0R | UARTA0 reception completion | UARTA0 | 0270H | 00000280H | nextPC | UA0RIC |
| | | 32 | INTUA0T | UARTA0 transmission enable | UARTA0 | 0280H | 00000280H | nextPC | UA0TIC |
| | | 33 | INTUA1R | UARTA1 reception completion/UARTA1 reception error | UARTA1 | 0290H | 00000290H | nextPC | UA1RIC |
| | | 34 | INTUA1T | UARTA1 transmission enable | UARTA1 | 02A0H | 000002A0H | nextPC | UA1TIC |
| | | 35 | INTAD | A/D conversion completion | A/D | 02BH | 000002B0H | nextPC | ADIC |
| | | 36 | INTKR | Key return interrupt request | 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 | | |

Remarks 1. Default Priority: The priority order when two or more maskable interrupt requests occur at the same time. The highest priority is 0.

The priority order of non-maskable interrupt is INTWDT2 > NMI.

Restored PC: The value of the program counter (PC) saved to EIPC, FEPC, or DBPC 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 interrupt 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).

14.2 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”.

The function of the NMI pin is enabled by setting the PMC0.PMC02 bit to 1 and the INTF0.INTF02 bit and INTR0.INTR02 bit to a desired value, and specifying a desired valid edge.

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 an 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 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 executed by the non-maskable interrupt request signal (INTWDT2), see 14.2.2 (2) From INTWDT2 signal.

Figure 14-1. Non-Maskable Interrupt Request Signal Acknowledgment Operation (1/2)

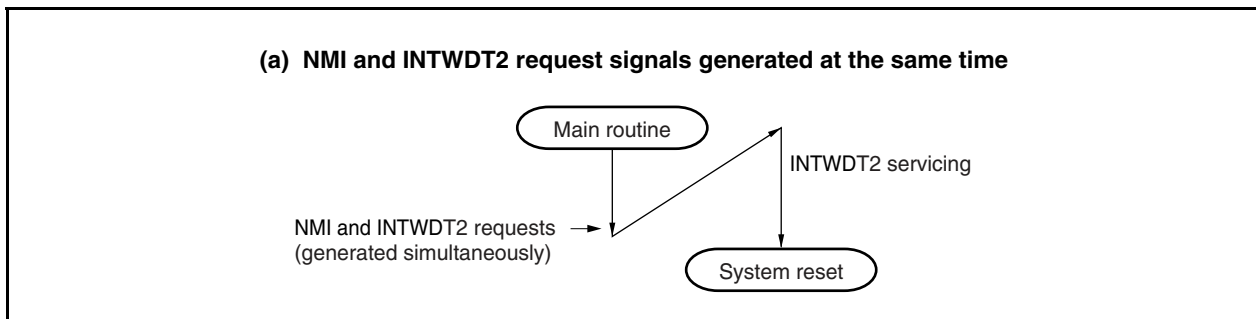
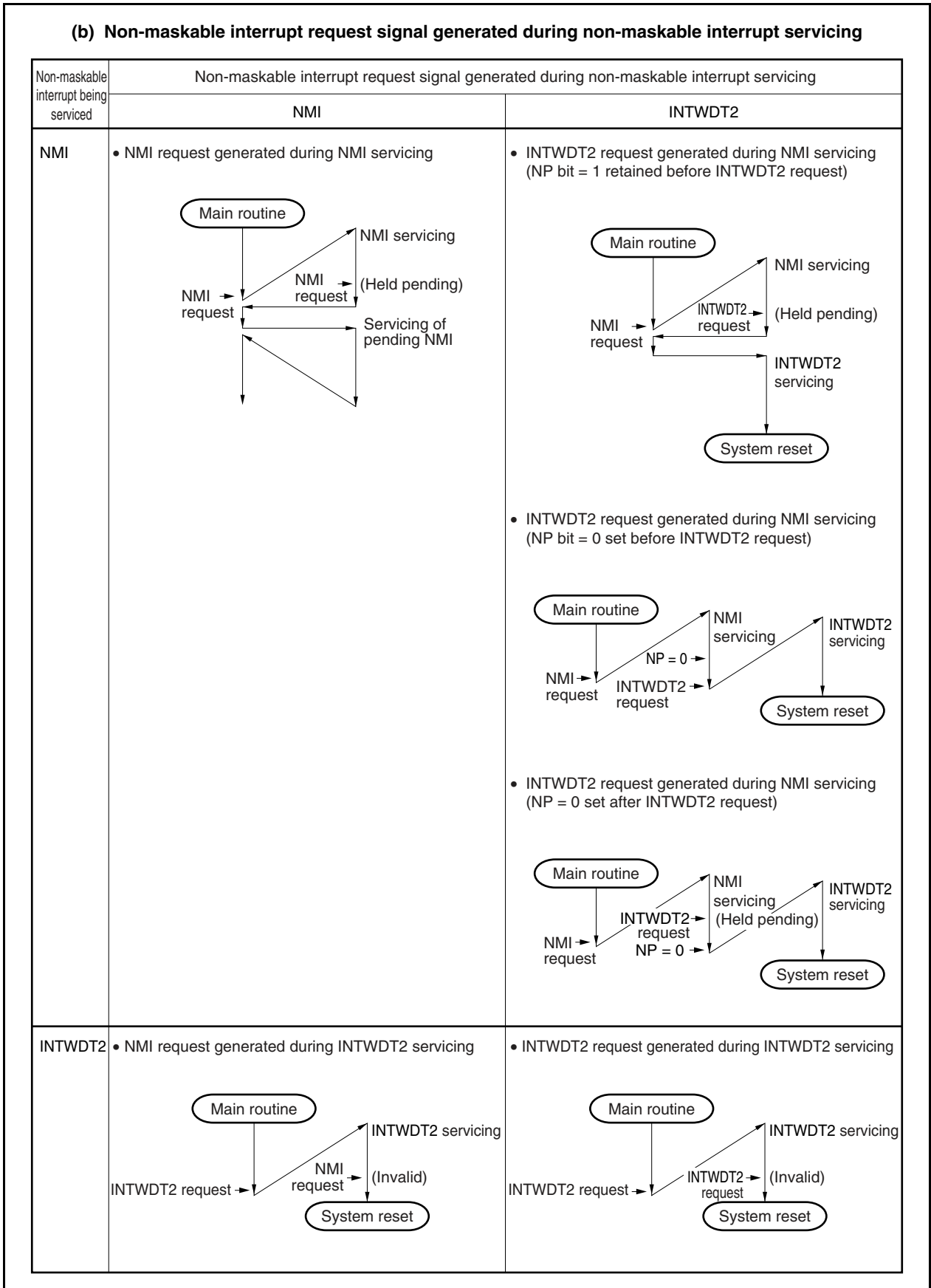


Figure 14-1. Non-Maskable Interrupt Request Signal Acknowledgment Operation (2/2)



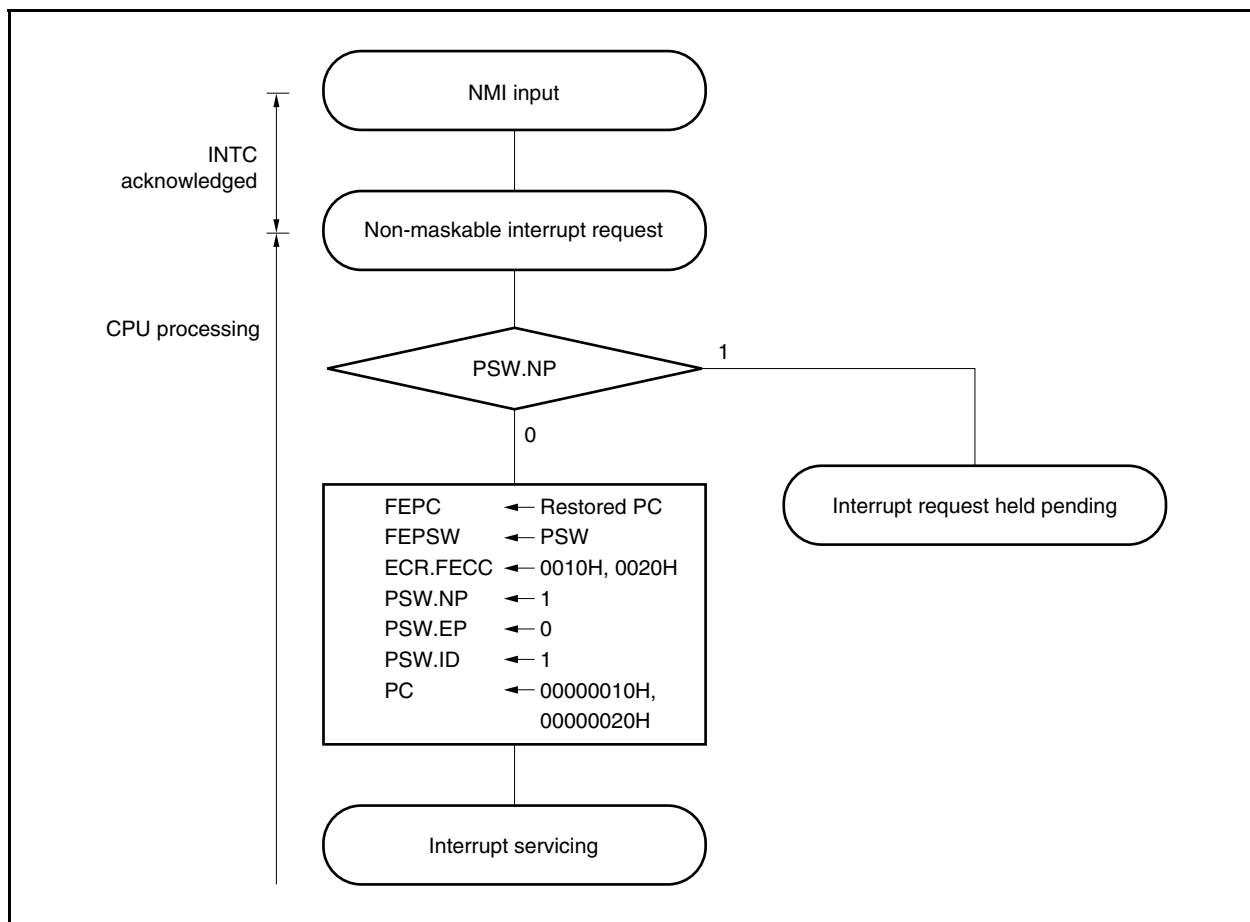
14.2.1 Operation

If a non-maskable interrupt request signal is generated, 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, 0020H) to the higher halfword (FECC) of ECR.
- <4> Sets the PSW.NP and PSW.ID bits to 1 and clears the PSW.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 14-2.

Figure 14-2. Servicing Configuration of Non-Maskable Interrupt



14.2.2 Restore

(1) From NMI pin input

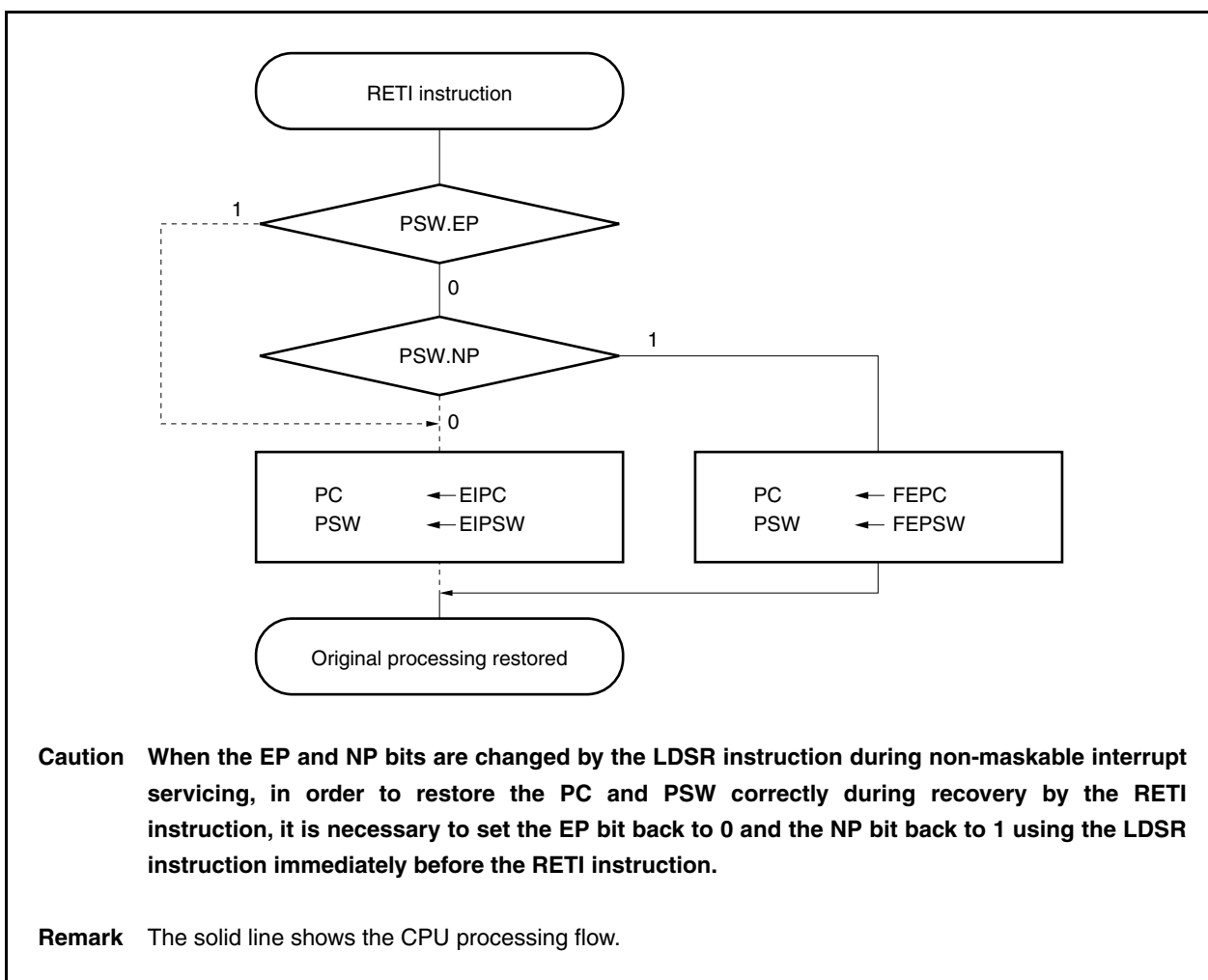
Execution is restored from the NMI servicing by the RETI instruction.

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 14-3 illustrates how the RETI instruction is processed.

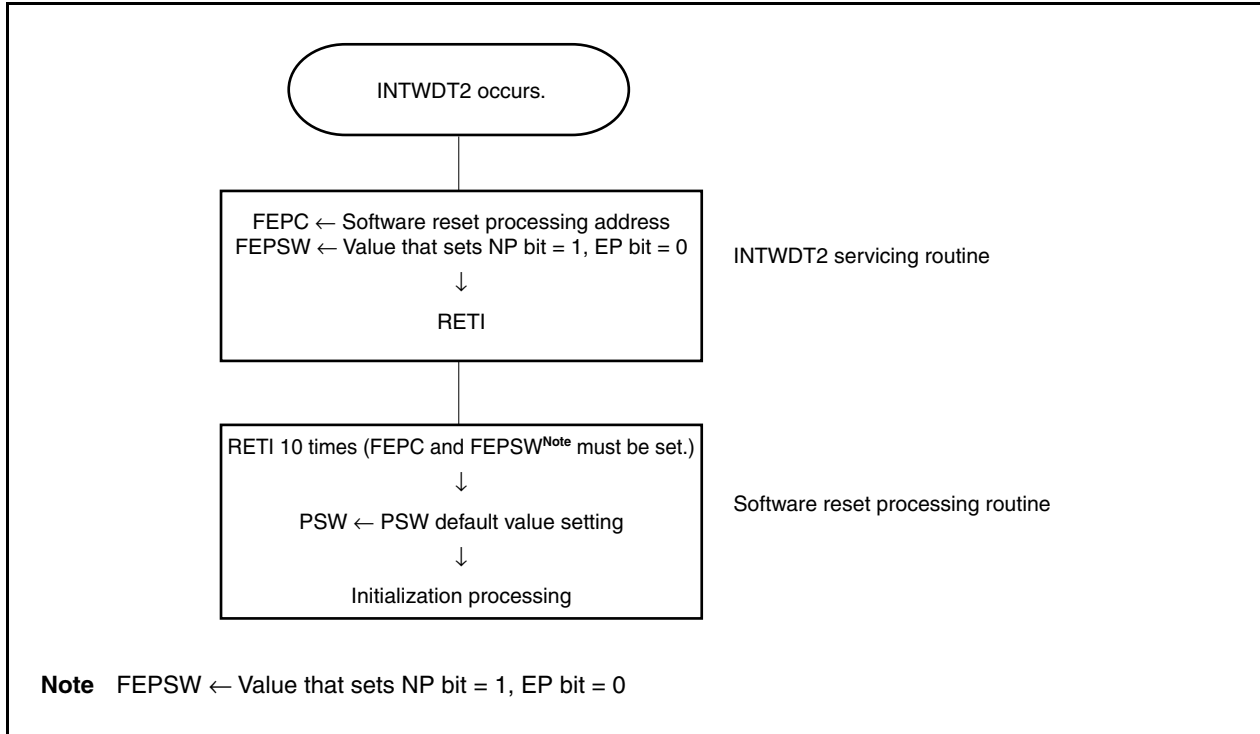
Figure 14-3. RETI Instruction Processing



(2) From INTWDT2 signal

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.

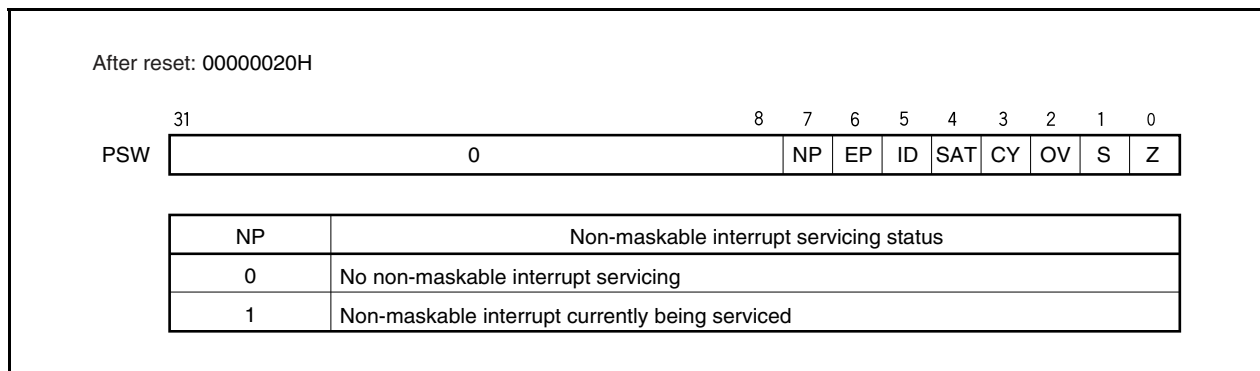
Figure 14-4. Software Reset Processing



14.2.3 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.



14.3 Maskable Interrupts

Maskable interrupt request signals can be masked by interrupt control registers. The V850ES/HF2 has 39 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 general-purpose registers before executing the EI instruction, and execute the DI instruction before the RETI instruction to restore the original values of EIPC and EIPSW.

14.3.1 Operation

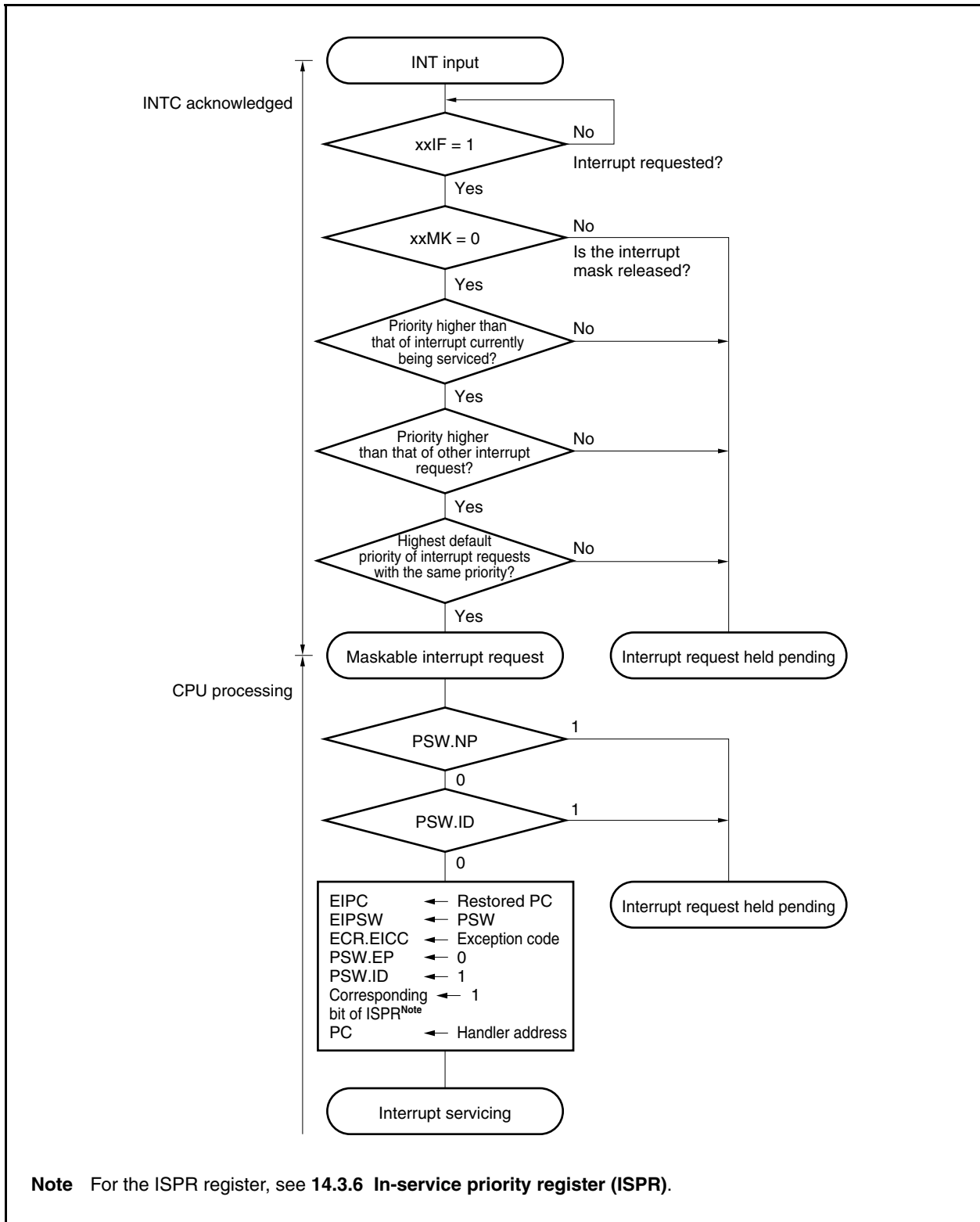
If a maskable interrupt occurs, the CPU performs the following processing, and transfers control to a handler routine.

- <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 the PSW.NP bit = 1 or the PSW.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 14-5. Maskable Interrupt Servicing



14.3.2 Restore

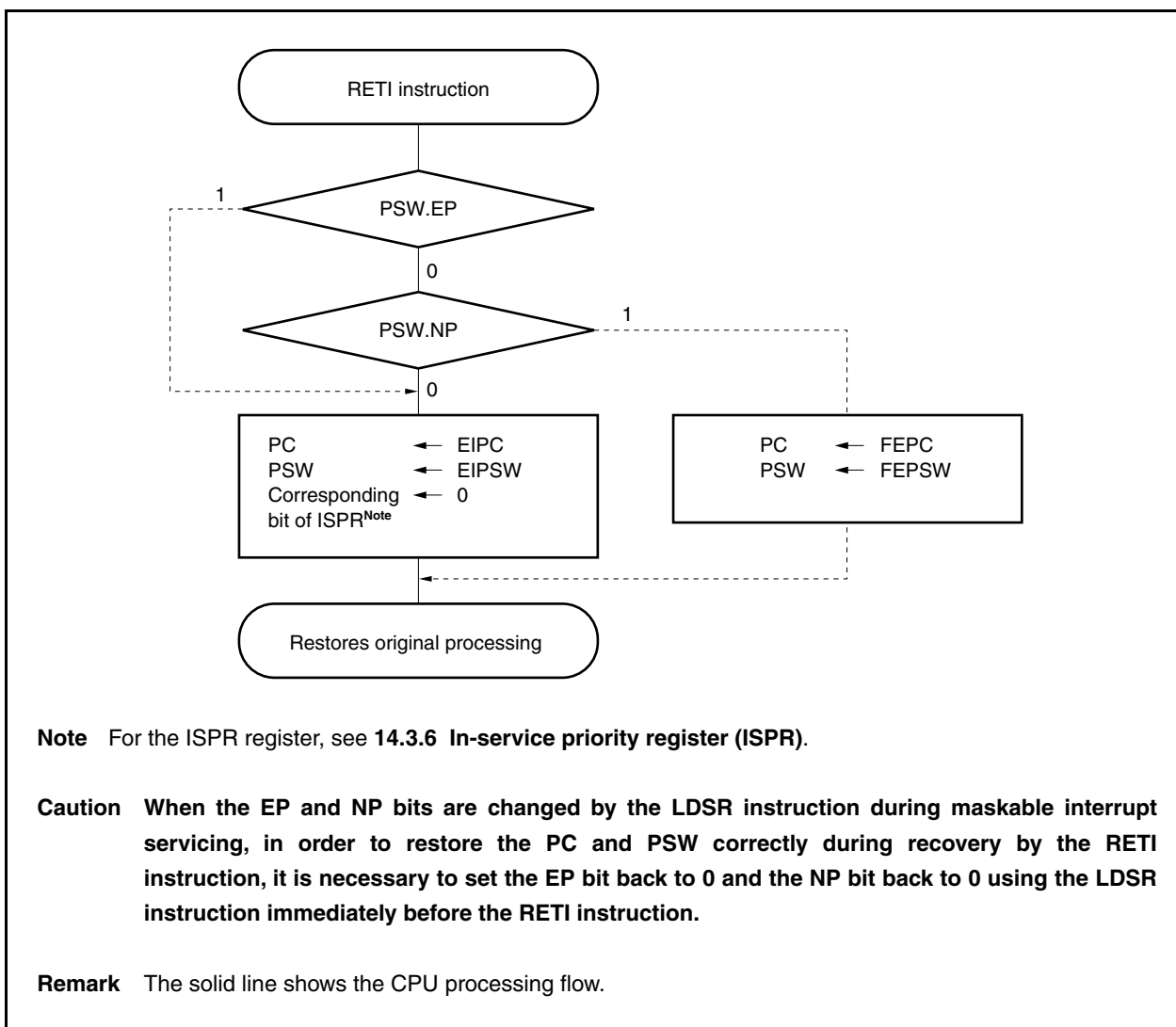
Recovery from maskable interrupt servicing is carried out by the RETI instruction.

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 14-6 illustrates the processing of the RETI instruction.

Figure 14-6. RETI Instruction Processing



14.3.3 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, see **Table 14-1 Interrupt/Exception 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 (see **Table 14-2 Interrupt Control Register (xxICn)**)
n: Peripheral unit number (see **Table 14-2 Interrupt Control Register (xxICn)**).

Figure 14-7. Example of Processing in Which Another Interrupt Request Signal Is Issued While an Interrupt Is Being Serviced (1/2)

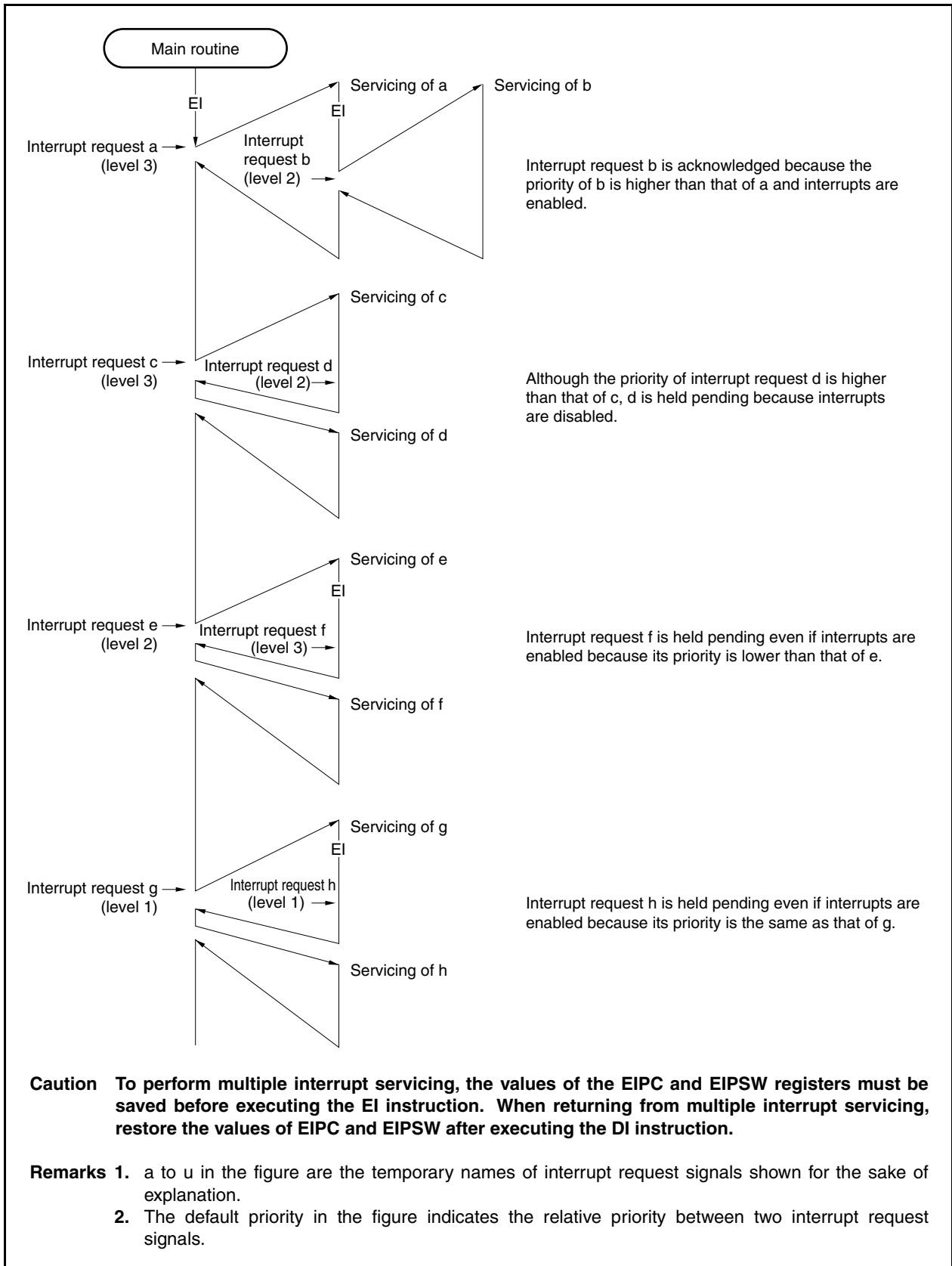


Figure 14-7. Example of Processing in Which Another Interrupt Request Signal Is Issued While an Interrupt Is Being Serviced (2/2)

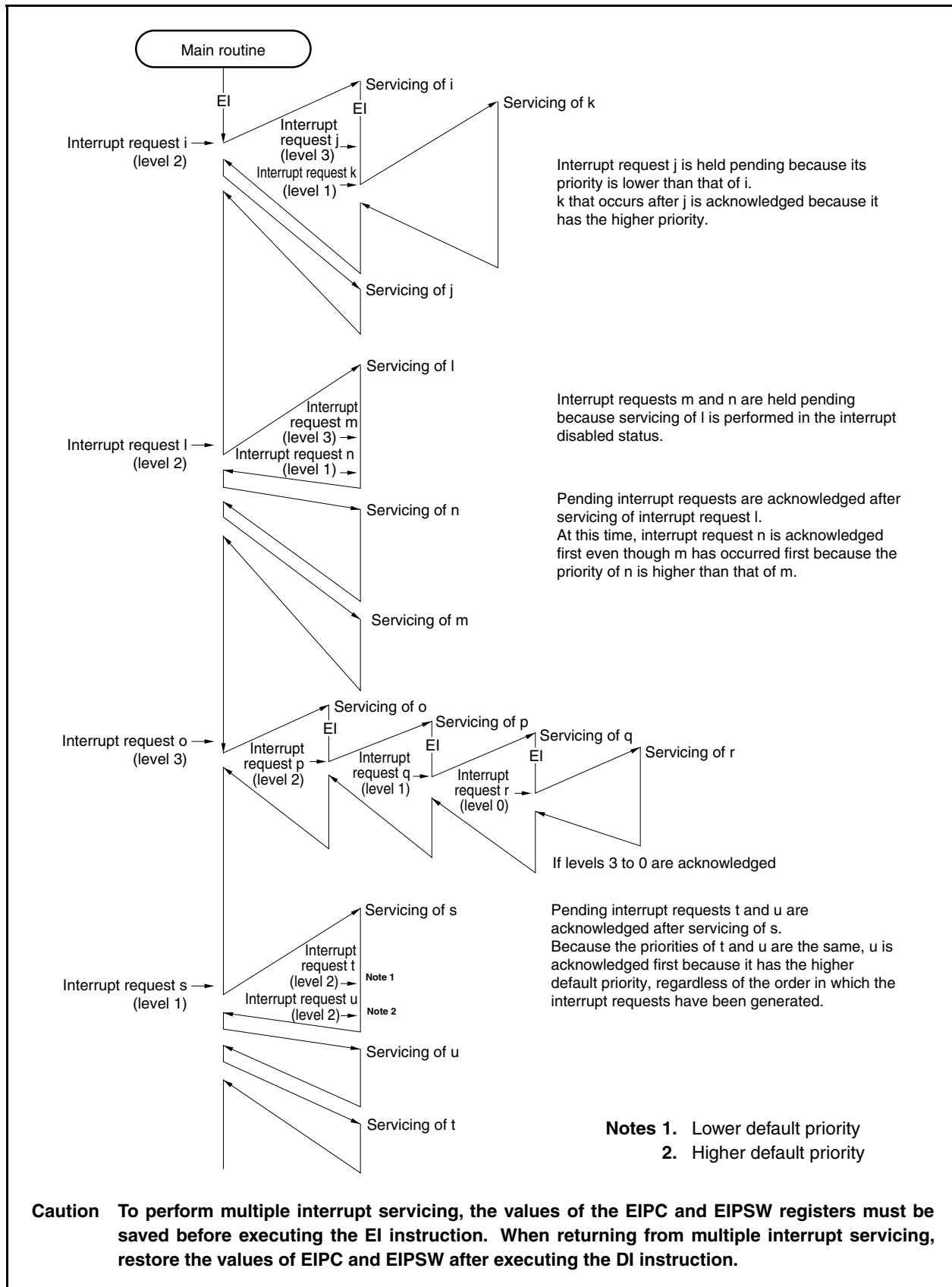
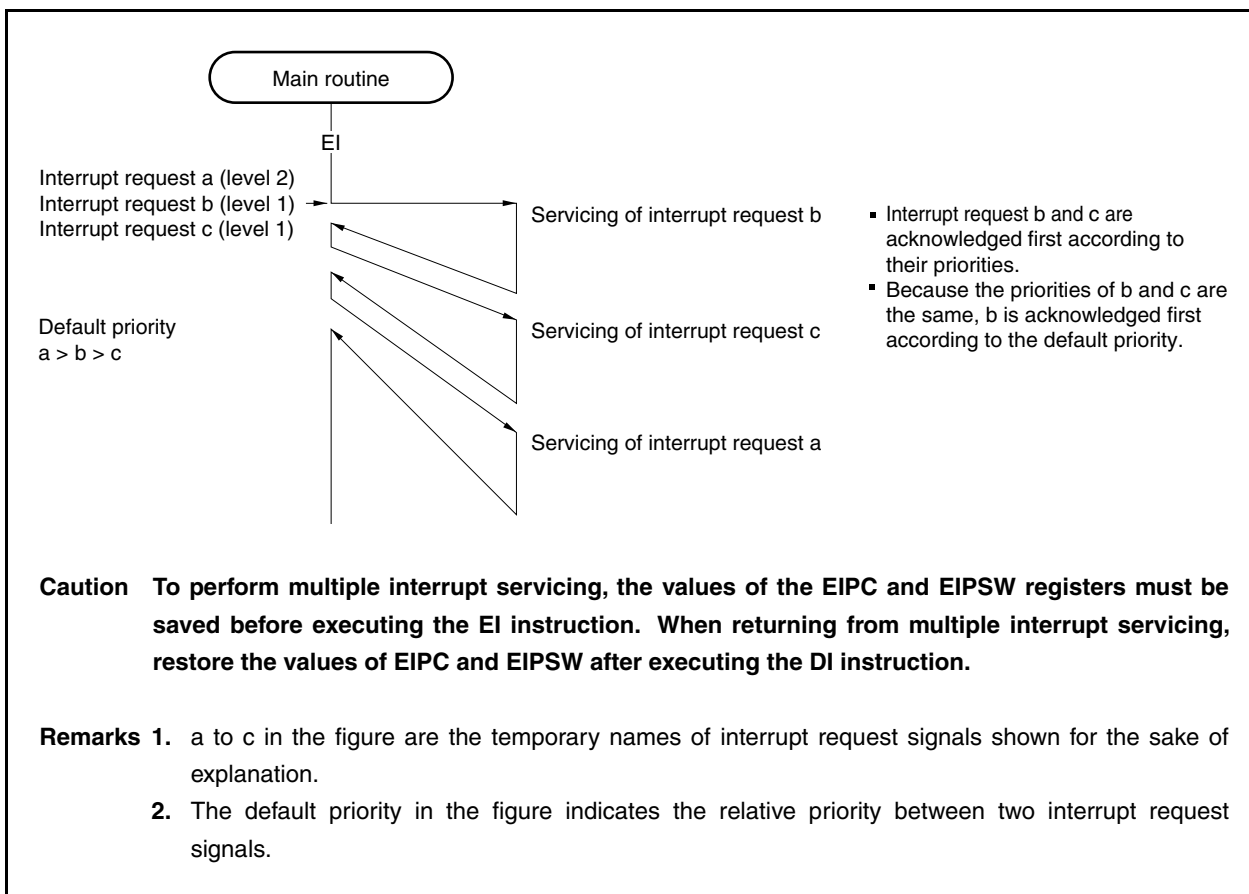


Figure 14-8. Example of Servicing Interrupt Request Signals Simultaneously Generated



14.3.4 Interrupt control register (xxICn)

The xxICn 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 sets this register to 47H.

Caution Disable interrupts (DI) or mask the interrupt to read the xxICn.xxIFn bit. If the xxIFn bit is read while interrupts are enabled (EI) or while the interrupt is unmasked, the correct value may not be read when acknowledging an interrupt and reading the bit conflict.

After reset: 47H R/W Address: FFFFF110H to FFFFF164H

| | | | | | | | | |
|-------|-------|-------|---|---|---|--------|--------|--------|
| | 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 not issued |
| 1 | Interrupt request 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 (see **Table 14-2 Interrupt Control Registers (xxICn)**)
n: Peripheral unit number (see **Table 14-2 Interrupt Control Registers (xxICn)**).

The addresses and bits of the interrupt control registers are as follows.

Table 14-2. Interrupt Control Registers (xxICn)

| Address | Register | Bit | | | | | | | |
|-----------|----------|----------|----------|---|---|---|-----------|-----------|-----------|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FFFFF110H | LVIIIC | LVIIIF | 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 | TQ0OVPR2 | TQ0OVPR1 | TQ0OVPR0 |
| FFFFF124H | TQ0CCIC0 | TQ0CCIF0 | TQ0CCMK0 | 0 | 0 | 0 | TQ0CCPR2 | TQ0CCPR1 | TQ0CCPR0 |
| 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 | TP0CCPR2 | TP0CCPR1 | TP0CCPR0 |
| 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 | TP1CCPR2 | TP1CCPR1 | TP1CCPR0 |
| 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 | TP2CCPR2 | TP2CCPR1 | TP2CCPR0 |
| 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 | TP3CCPR2 | TP3CCPR1 | TP3CCPR0 |
| FFFFF142H | TP3CCIC1 | TP3CCIF1 | TP3CCMK1 | 0 | 0 | 0 | TP3CCPR12 | TP3CCPR11 | TP3CCPR10 |
| FFFFF144H | TM0EQIC0 | TM0EQIF0 | TM0EQMK0 | 0 | 0 | 0 | TM0EQPR2 | TM0EQPR1 | TM0EQPR0 |
| FFFFF146H | CB0RIC | CB0RIF | CB0RMK | 0 | 0 | 0 | CB0RPR2 | CB0RPR1 | CB0RPR0 |
| 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 | ADPR1 | 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 |

14.3.5 Interrupt mask registers 0 to 2 (IMR0 to IMR2)

The IMR0 to IMR2 registers set the interrupt mask state for the maskable interrupts. The xxMKn bit of the IMR0 to IMR2 registers is equivalent to the xxICn.xxMKn bit.

The IMRm register can be read or written in 16-bit units (m = 0 to 2).

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 2).

Reset sets these registers to FFFFH.

Caution The device file defines the xxICn.xxMKn bit 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: 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 | 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 | Setting of interrupt mask flag | | | | | | | |
| 0 | Interrupt servicing enabled | | | | | | | |
| 1 | Interrupt servicing disabled | | | | | | | |

Note To read bits 8 to 15 of the IMR0 to IMR2 registers in 8-bit or 1-bit units, specify them as bits 0 to 7 of the IMR0H to IMR2H registers.

Caution Set bits 15 to 11 and 7 to 4 of the IMR2 register to “1”. If the setting of these bits is changed, the operation is not guaranteed.

Remark xx: Identification name of each peripheral unit (see **Table 14-2 Interrupt Control Registers (xxICn)**).

n: Peripheral unit number (see **Table 14-2 Interrupt Control Registers (xxICn)**)

14.3.6 In-service priority register (ISPR)

The ISPR 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 sets 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).

| | | | | | | | | |
|------------------|-------|---|--------------------|-------|-------|-------|-------|-------|
| 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)

14.3.7 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 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 | Specification of maskable interrupt servicing ^{Note} |
|----|---|
| 0 | Maskable interrupt request signal acknowledgment enabled |
| 1 | Maskable interrupt request signal acknowledgment disabled (pending) |

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.

14.3.8 Watchdog timer mode register 2 (WDTM2)

This register can be read or written in 8-bit units (for details, see **CHAPTER 10 FUNCTIONS OF WATCHDOG TIMER 2**).

Reset sets this register to 67H.

After reset: 67H R/W Address: FFFFF6D0H

| | | | | | | | | |
|-------|---|-------|-------|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WDTM2 | 0 | WDM21 | WDM20 | 0 | 0 | 0 | 0 | 0 |

| WDM21 | WDM20 | Selection of watchdog timer operation mode |
|-------|-------|--|
| 0 | 0 | Stops operation |
| 0 | 1 | Non-maskable interrupt request mode |
| 1 | × | Reset mode (initial-value) |

14.4 Software Exception

A software exception is generated when the CPU executes the TRAP instruction, and can always be acknowledged.

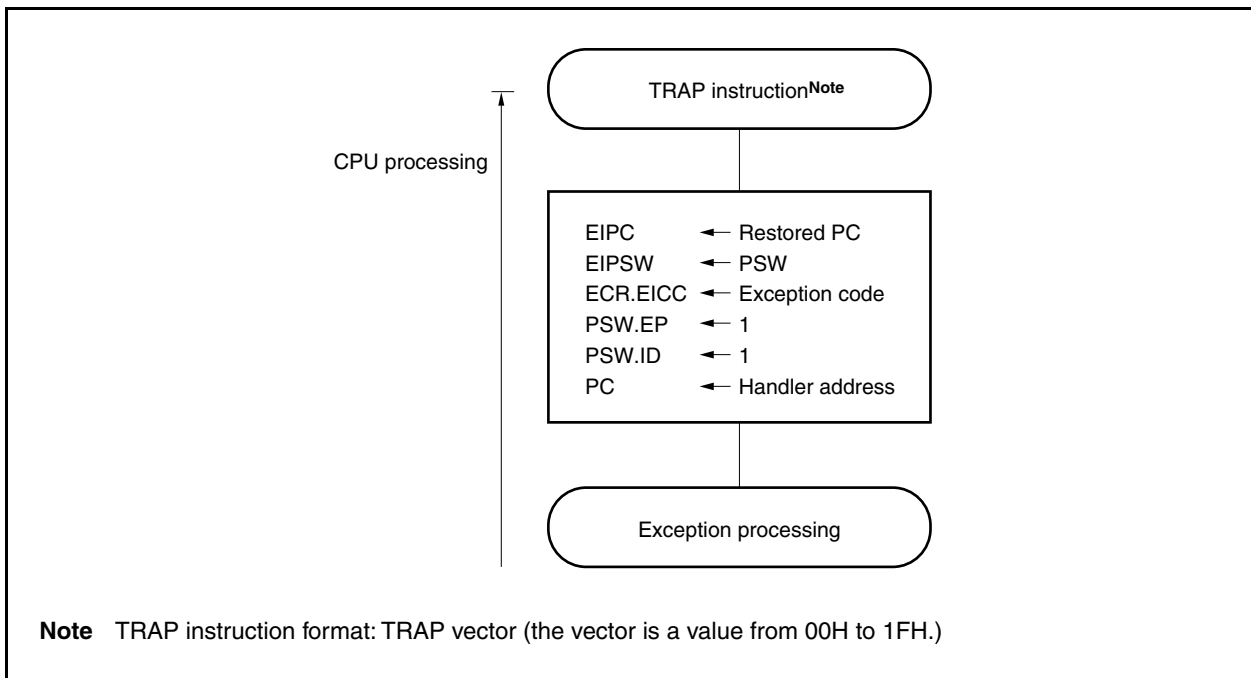
14.4.1 Operation

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 14-9 illustrates the processing of a software exception.

Figure 14-9. Software Exception Processing



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.

14.4.2 Restore

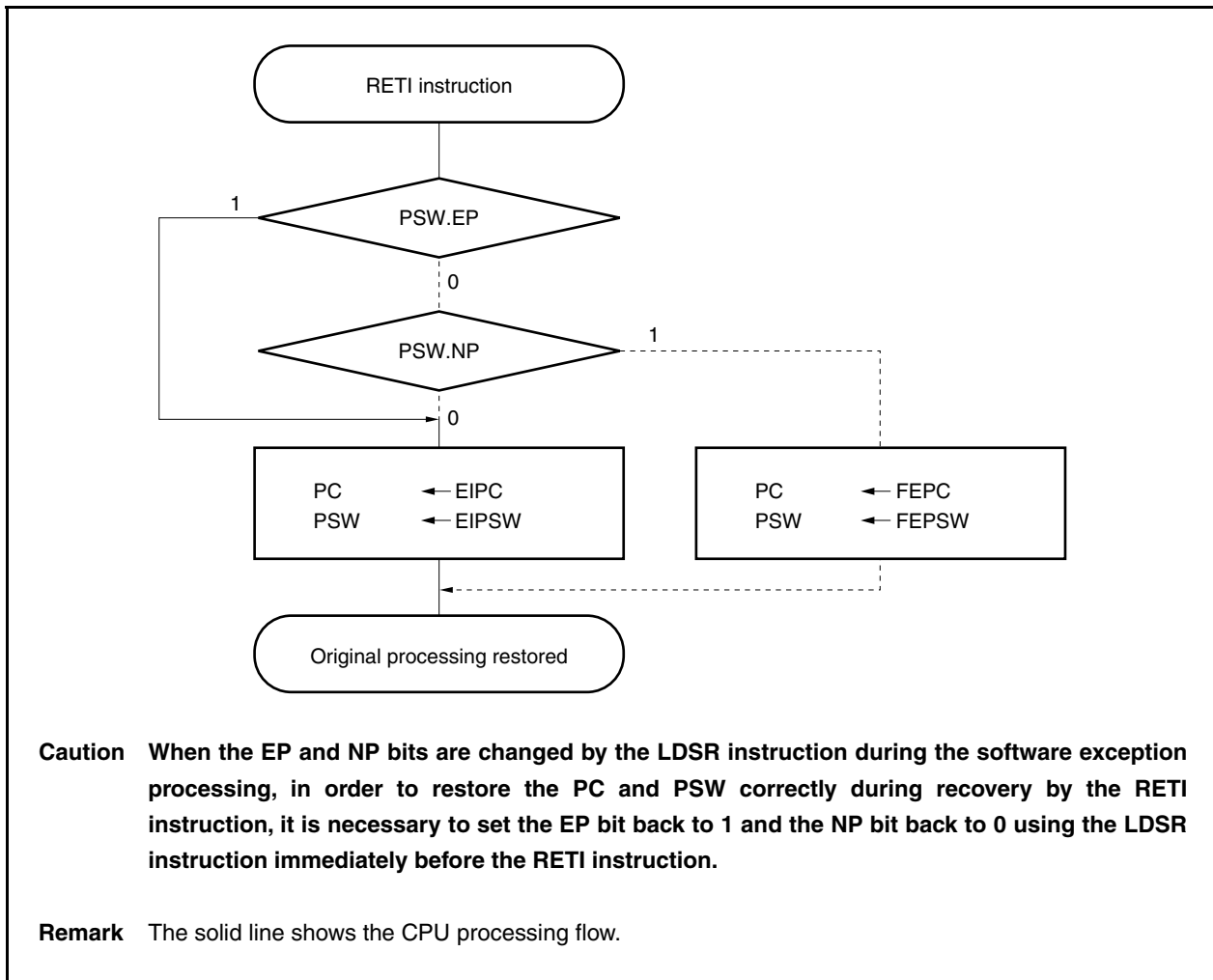
Recovery from software exception processing is carried out by the RETI instruction.

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 14-10 illustrates the processing of the RETI instruction.

Figure 14-10. RETI Instruction Processing



14.4.3 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 |

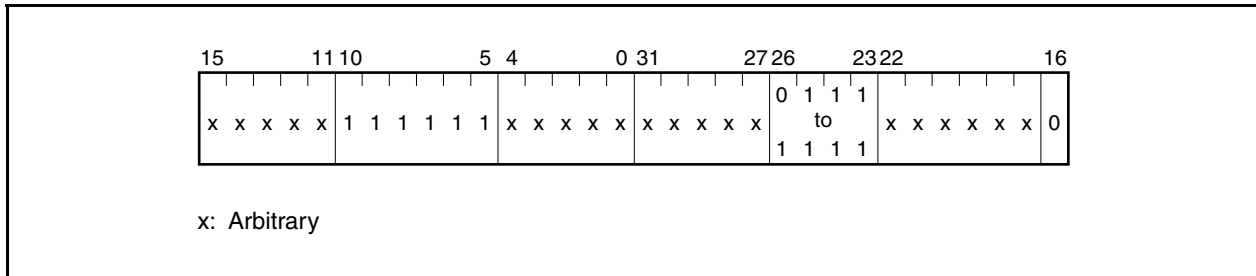
| EP | Exception processing status |
|----|---------------------------------------|
| 0 | Exception processing not in progress. |
| 1 | Exception processing in progress. |

14.5 Exception Trap

An exception trap is an interrupt that is requested when the illegal execution of an instruction takes place. In the V850ES/HF2, an illegal opcode exception (ILGOP: Illegal Opcode Trap) is considered as an exception trap.

14.5.1 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.



Caution Since it is possible to assign this instruction to an illegal opcode in the future, it is recommended that it not be used.

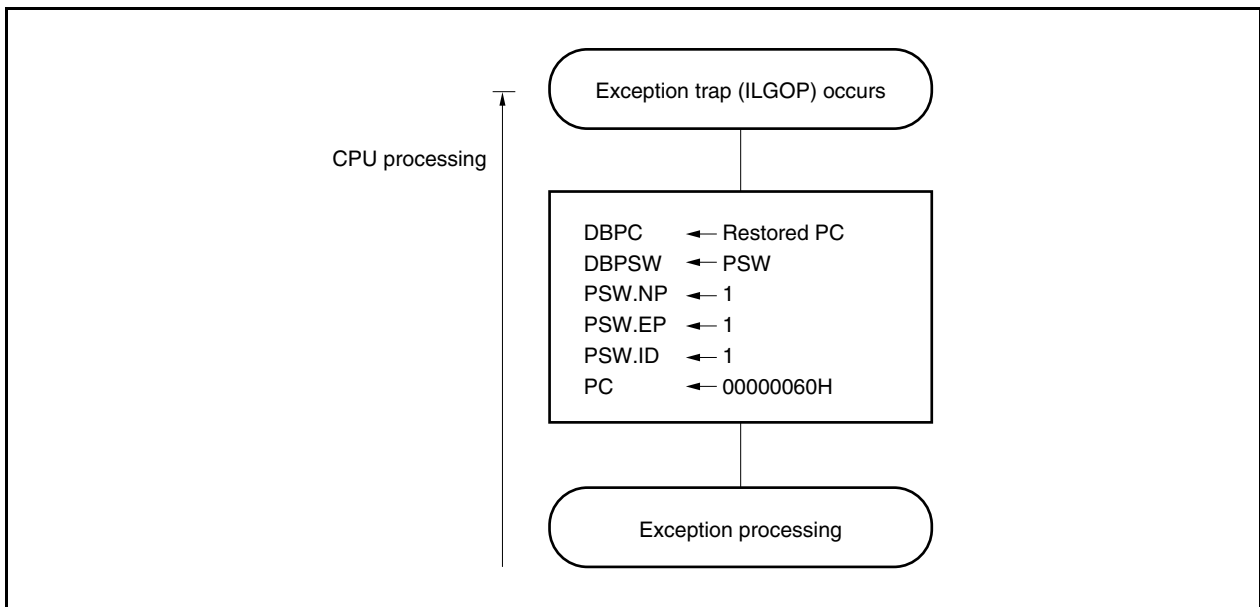
(1) Operation

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 14-11 illustrates the processing of the exception trap.

Figure 14-11. Exception Trap Processing

**(2) Restore**

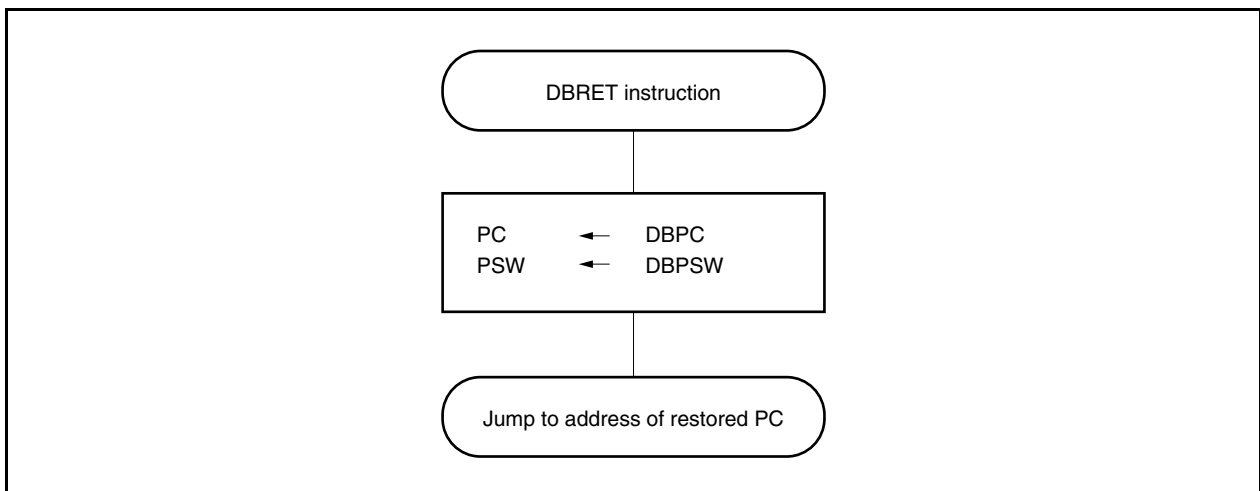
Recovery from an exception trap is carried out by the DBRET instruction. By executing the DBRET instruction, the CPU carries out the following processing and controls the address of the restored PC.

- <1> Loads the restored PC and PSW from DBPC and DBPSW.
- <2> Transfers control to the address indicated by the restored PC and PSW.

Caution DBPC and DBPSW can be accessed only during the interval between the execution of the illegal opcode and the DBRET instruction.

Figure 14-12 illustrates the restore processing from an exception trap.

Figure 14-12. Restore Processing from Exception Trap



14.5.2 Debug trap

A debug trap is an exception that is generated when the DBTRAP instruction is executed and is always acknowledged.

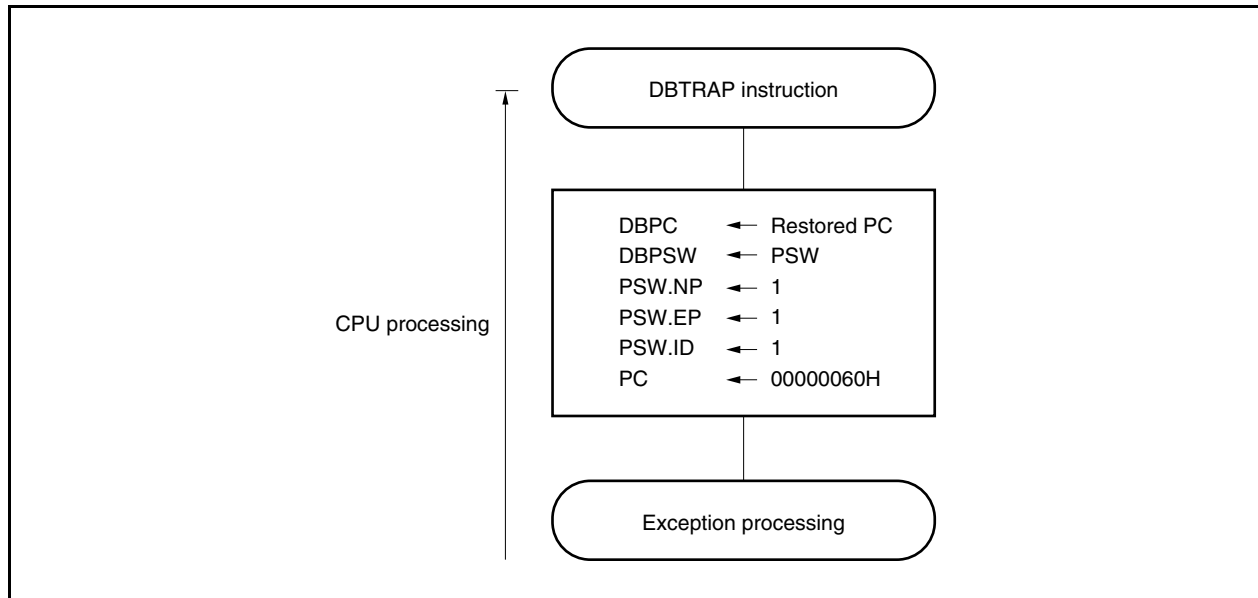
(1) Operation

Upon occurrence of a debug trap, the CPU performs the following processing.

- <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 14-13 shows the debug trap processing format.

Figure 14-13. Debug Trap Processing Format



(2) Restoration

Restoration from a debug trap is executed with the DBRET instruction.

With the DBRET instruction, the CPU performs the following steps and transfers control to the address of the restored PC.

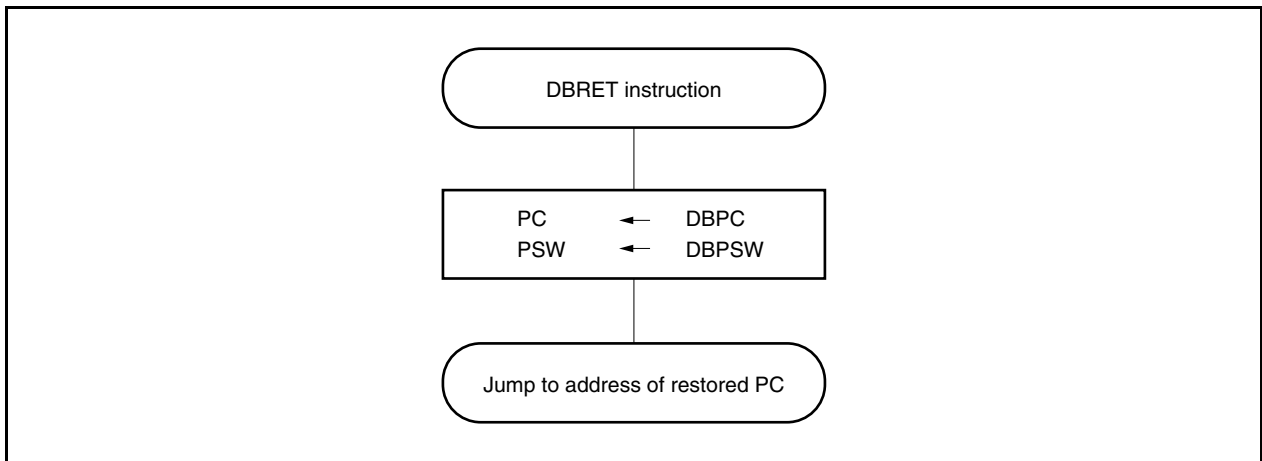
<1> The restored PC and PSW are read from DBPC and DBPSW.

<2> Control is transferred to the fetched address of the restored PC and PSW.

Caution DBPC and DBPSW can be accessed only during the interval between the execution of the DBTRAP instruction and the DBRET instruction.

Figure 14-14 shows the processing format for restoration from a debug trap.

Figure 14-14. Processing Format of Restoration from Debug Trap



14.6 External Interrupt Request Input Pins (NMI and INTP0 to INTP7)

14.6.1 Noise elimination

(1) Eliminating noise on NMI pin

The NMI pin has an internal noise elimination circuit 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 elimination circuit 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.

14.6.2 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 register (the NMI pin functions as a normal port pin).

(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 sets 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.

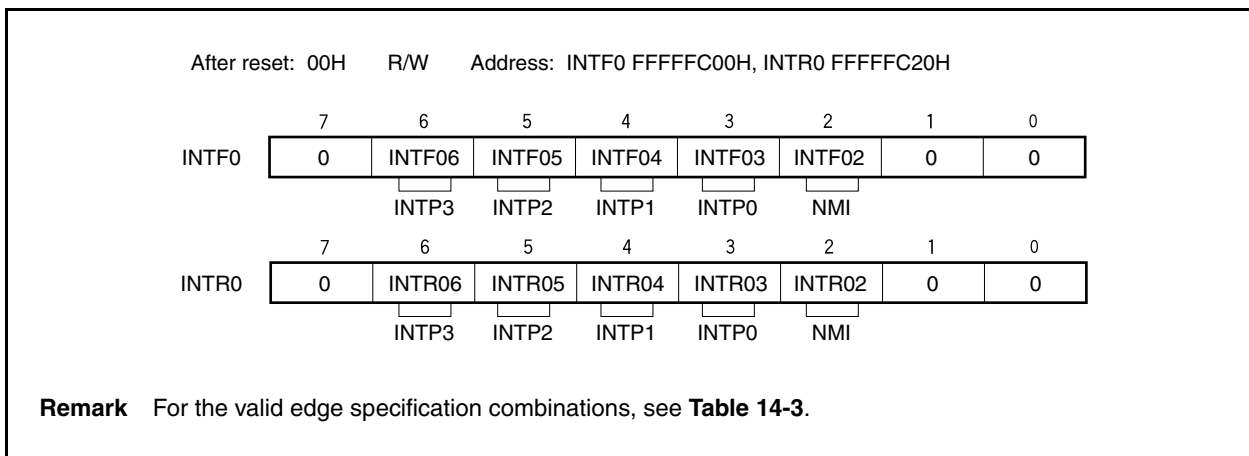


Table 14-3. Valid Edge Specification

| 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 if the corresponding pin is 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 rising, falling edge specification register 3L (INTR3L, INTF3L)

The INTR3L and INTF3L registers are 8-bit registers that specify detection of the rising and falling edges of the INTP7 pin.

These registers can be read or written in 8-bit or 1-bit units.

Reset sets 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 INTF31 and INTR31 bits to 00, and then set the port mode.

After reset: 0000H R/W Address: FFFFC06H

| | | | | | | | | |
|--------|---|---|---|---|---|---|--------|---|
| INTF3L | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 0 | INTF31 | 0 |

└──────────┘
INTP7

After reset: 0000H R/W Address: FFFFC26H

| | | | | | | | | |
|--------|---|---|---|---|---|---|--------|---|
| INTR3L | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 0 | INTR31 | 0 |

└──────────┘
INTP7

Remark For the valid edge specification combinations, see **Table 14-4**.

Table 14-4. 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 if the corresponding pin is not used as the 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 sets 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.

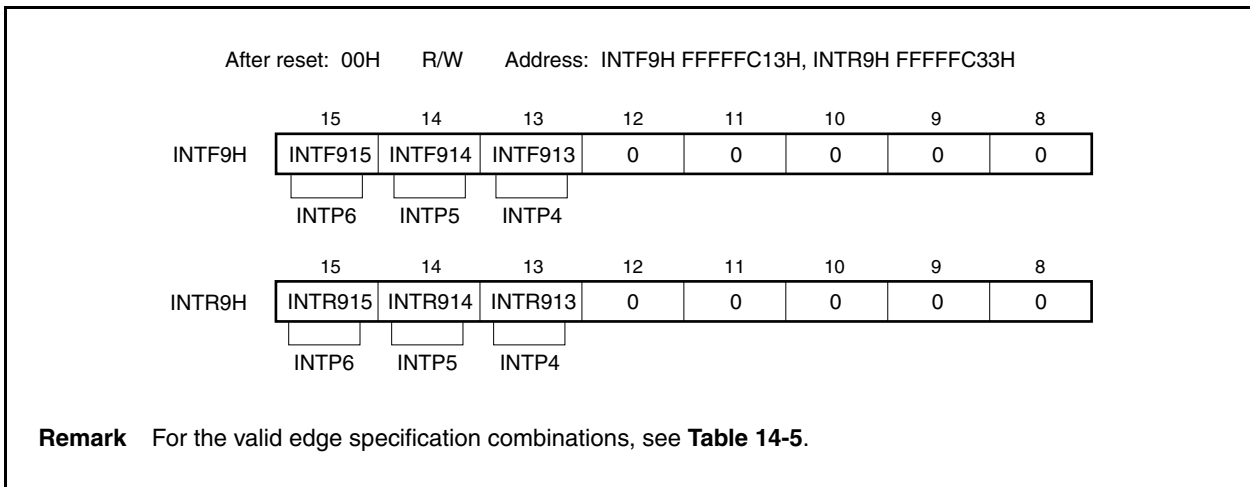


Table 14-5. Valid Edge Specification

| 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 if the corresponding pin is not used as INTP4 to INTP6 pins.

Remark n = 13 to 15: Control of INTP4 to INTP6 pins

(4) Noise elimination control register (NFC)

Digital noise elimination can be selected for the INTP3 pin. The noise elimination settings are performed using the NFC register.

<R> 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 times are set by the NFC.NFSTS bit.

When digital noise elimination is selected, if the clock that performs sampling in the standby mode is stopped, then the INTP3 interrupt request signal cannot be used for releasing the standby mode. When f_{XT} is used as the sampling clock, the INTP3 interrupt request signal can be used for releasing either the subclock operating mode or the IDLE1/IDLE2/STOP/sub-IDLE mode.

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

Reset sets this register to 00H.

Caution Time equal to the sampling clock \times the number of times set by the NFSTS bit is required until the digital noise eliminator is initialized after the sampling clock has been changed. If the valid edge of INTP3 is input after the sampling clock has been changed and before the time of the sampling clock \times the number of times set by the NFSTS bit passes, therefore, the interrupt request signal may be generated. Therefore, note the following points when using the interrupt function.

- When using the interrupt function, after the sampling clock \times the number of times set by the NFSTS bit have elapsed, enable interrupts after the interrupt request flag (PIC3.PIF3 bit) has been cleared.

After reset: 00H R/W Address: FFFF318H

| | | | | | | | | |
|-----|------|-------|---|---|---|------|------|------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| NFC | NFEN | NFSTS | 0 | 0 | 0 | NFC2 | NFC1 | NFC0 |

| | |
|------|---|
| NFEN | Settings of INTP3 pin noise elimination |
| 0 | Analog noise elimination |
| 1 | Digital noise elimination |

| | |
|-------|---|
| NFSTS | Setting of number of times of sampling of digital noise elimination |
| 0 | Number of times of sampling \times 3 times |
| 1 | Number of times of sampling \times twice |

| | | | |
|------------------|------|------|------------------------|
| NFC2 | NFC1 | NFC0 | Digital sampling clock |
| 0 | 0 | 0 | $f_{xx}/64$ |
| 0 | 0 | 1 | $f_{xx}/128$ |
| 0 | 1 | 0 | $f_{xx}/256$ |
| 0 | 1 | 1 | $f_{xx}/512$ |
| 1 | 0 | 0 | $f_{xx}/1,024$ |
| 1 | 0 | 1 | f_{xt} (subclock) |
| Other than above | | | Setting prohibited |

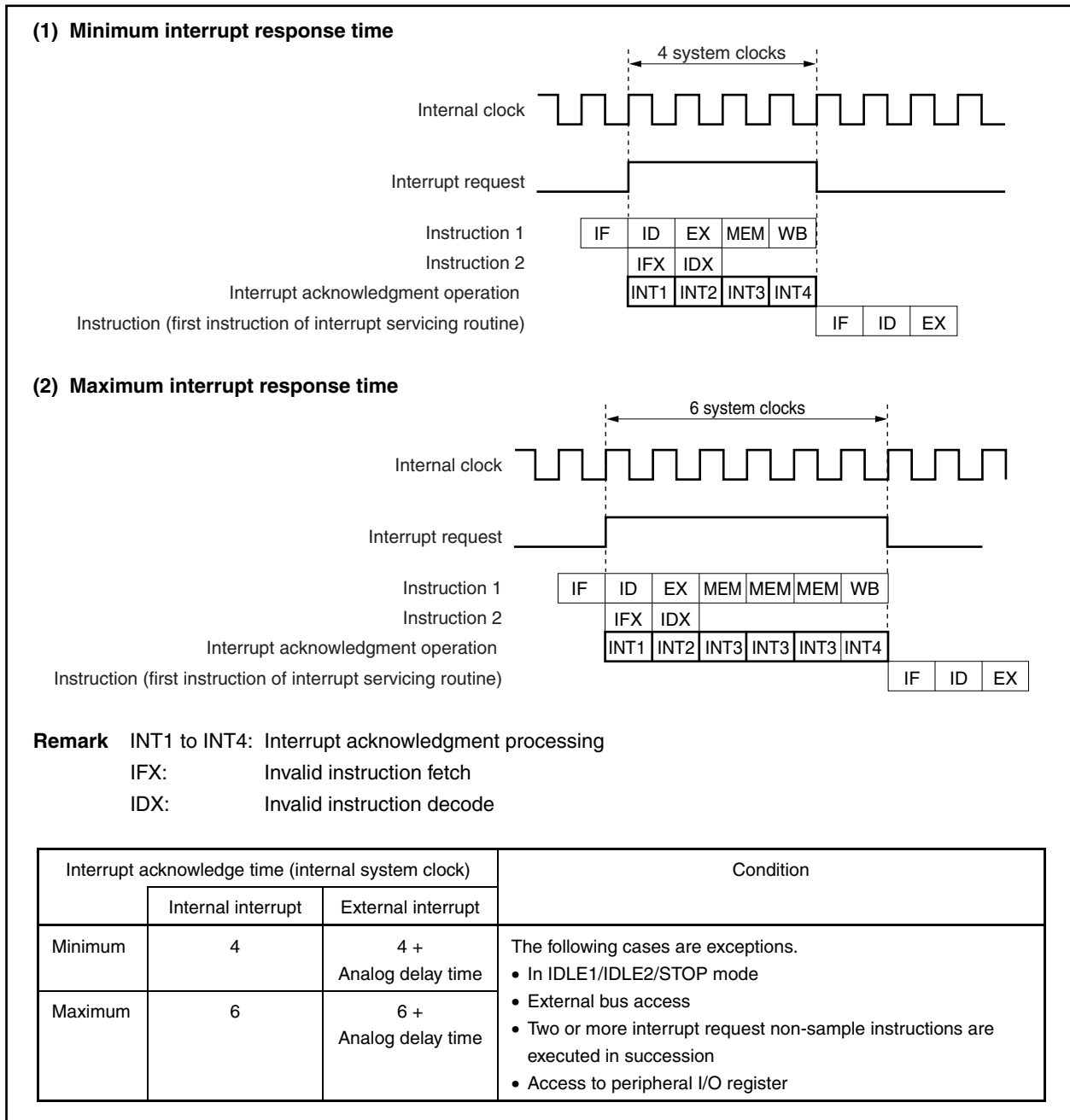
- Remarks**
1. Since sampling is performed three times, the reliably eliminated noise width is 2 sampling clocks.
 2. In the case of noise with a width smaller than 2 sampling clocks, an interrupt request signal is generated if noise synchronized with the sampling clock is input.

14.7 Interrupt Acknowledge Time of CPU

Except the following cases, the interrupt acknowledge time of the CPU is 4 clocks minimum. To input interrupt request signals successively, input the next interrupt request signal at least 5 clocks after the preceding interrupt.

- In IDLE1/IDLE2/STOP mode
- When the external bus is accessed
- When interrupt request non-sampling instructions are successively executed (see 14.8 **Periods in Which Interrupts Are Not Acknowledged by CPU.**)
- When the interrupt control register is accessed

Figure 14-15. Pipeline Operation at Interrupt Request Signal Acknowledgment (Outline)



14.8 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)
- The store instruction for the PRCMD register
- The store, SET1, NOT1, or CLR1 instructions for the following registers:
 - Interrupt-related registers:
Interrupt control register (xxICn), interrupt mask registers 0 to 2 (IMR0 to IMR2)
 - Power save control register (PSC)
 - On-chip debug mode register (OCDM)
 - Peripheral emulation register 1 (PEMU1):

Remark xx: Identification name of each peripheral unit (see **Table 14-2 Interrupt Control Registers (xxICn)**)

n: Peripheral unit number (see **Table 14-2 Interrupt Control Registers (xxICn)**).

14.9 Cautions

The NMI pin alternately functions as the P02 pin. It functions as a normal port pin after reset. To enable the NMI pin, validate the NMI pin with the PMC0 register. The initial setting of the NMI pin is “No edge detected”. Select the NMI pin valid edge using the INTF0 and INTR0 registers.

CHAPTER 15 KEY INTERRUPT FUNCTION

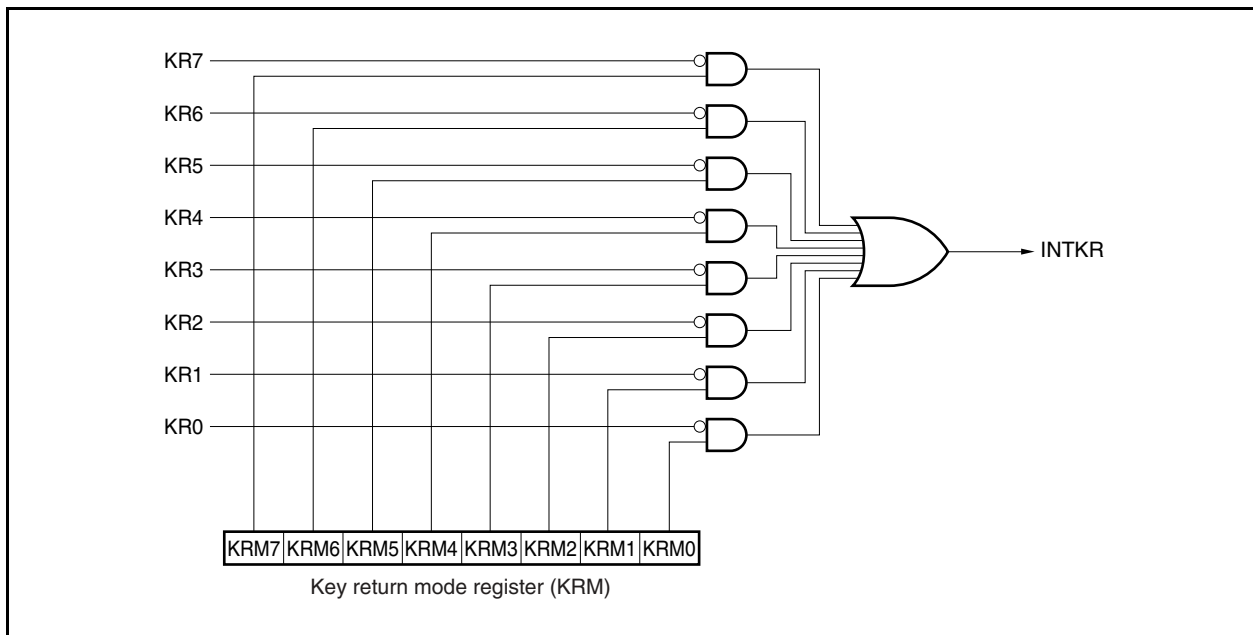
15.1 Function

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 KRM register.

Table 15-1. 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 15-1. Key Return Block Diagram



15.2 Register

(1) Key return mode register (KRM)

The KRM register controls the KRM0 to KRM7 bits using the KR0 to KR7 signals.

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

Reset sets this register to 00H.

| | | | | | | | | |
|------------------|-----------------------------------|------|--------------------|------|------|------|------|------|
| After reset: 00H | | R/W | Address: 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 Rewrite the KRM register after once clearing the KRM register to 00H.

Remark For the alternate-function pin settings, see **Table 4-14 Using Port Pin as Alternate Function Pin**.

15.3 Cautions

- (1) 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.
- (2) The RXDA1 and KR7 pins must not be used at the same time. To use the RXDA1 pin, do not use the KR7 pin. To use the KR7 pin, do not use the RXDA1 pin (it is recommended to set the PFC91 bit to 1 and clear PFCE91 bit to 0).
- (3) 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) or masking, then clear the interrupt request flag (KRIC.KRIF bit) to 0, and enable interrupts (EI) or clear the mask.
- (4) To use the key interrupt function, be sure to set the port pin to the key return pin and then enable the operation with the KRM register. To switch from the key return pin to the port pin, disable the operation with the KRM register and then set the port pin.

CHAPTER 16 STANDBY FUNCTION

16.1 Overview

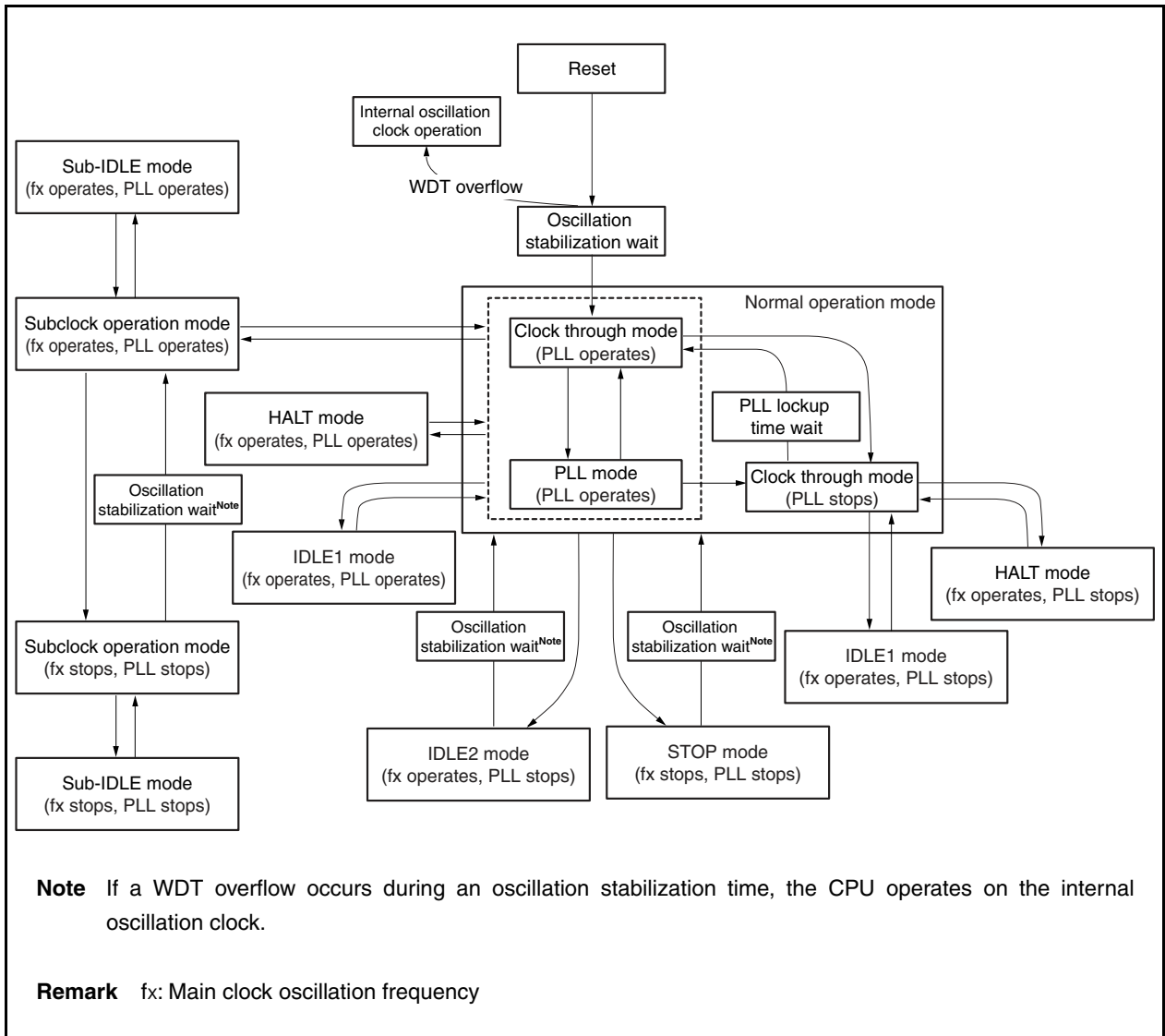
The power consumption of the system can be effectively reduced by using the standby modes in combination and selecting the appropriate mode for the application. The available standby modes are listed in Table 16-1.

Table 16-1. Standby Modes

| Mode | Functional Outline |
|-------------------------|---|
| HALT mode | Mode in which only the operating clock of the CPU is stopped |
| IDLE1 mode | Mode in which all the operations of the internal circuits except the oscillator, PLL ^{Note} , and flash memory are stopped |
| IDLE2 mode | Mode in which all the internal operations of the chip except the oscillator are stopped |
| STOP mode | Mode in which all the internal operations of the chip except the subclock oscillator are stopped |
| Subclock operation mode | Mode in which the subclock is used as the internal system clock |
| Sub-IDLE mode | Mode in which all the internal operations of the chip except the oscillator are stopped, in the subclock operation mode |

Note The PLL holds the previous operating status.

Figure 16-1. Status Transition



16.2 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 sets 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 ^{Note} setting |
| 0 | Normal mode |
| 1 | Standby mode |

Note Standby mode set by STP bit: IDLE1, IDLE2, STOP, or sub-IDLE 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.
 3. If the NMI1M, NMI0M, or INTM bit is set to 1 at the same time the STP bit is set to 1, the setting of NMI1M, NMI0M, or INTM bit becomes invalid. If there is an unmasked interrupt request signal being held pending when the IDLE1/IDLE2/STOP mode is set, set the bit corresponding to the interrupt request signal (NMI1M, NMI0M, or INTM) to 1, and then set the STP bit to 1.

<R>

(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 sets 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 |
| 0 | 1 | STOP mode |
| 1 | 0 | IDLE2, sub-IDLE modes |
| 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.

Sub-IDLE: In this mode, all other operations are halted except for the oscillator. After the IDLE mode has been released by the interrupt request signal, the subclock operation mode will be restored after 12 cycles of the subclock have been secured.

(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 8-bit units.

Reset sets this register to 06H.

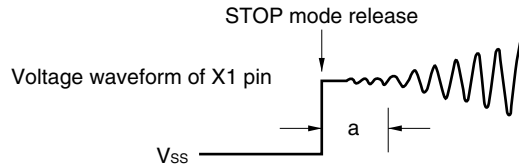
After reset: 06H R/W Address: FFFFF6C0H

| | | | | | | | | |
|------|---|---|---|---|---|-------|-------|-------|
| | 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} | f _x | |
|-------|-------|-------|--|----------------|-----------|
| | | | | 4 MHz | 5 MHz |
| | | | | 0 | 0 |
| 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 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 2¹⁶/f_x (because the initial value of the OSTS register = 06H).

Remark f_x = Main clock oscillation frequency

16.3 HALT Mode

16.3.1 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 16-3 shows the operating status in the HALT mode.

The average current 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. **If 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.**

16.3.2 Releasing HALT mode

The HALT mode is released by a non-maskable interrupt request signal (NMI pin input, 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, power-on-clear circuit (POC), 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 currently being serviced is issued, the HALT mode is released, but 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 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 16-2. 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 16-3. Operating Status in HALT Mode

| Setting of HALT Mode | | Operating Status | |
|-----------------------------|----------------|---|-----------------------|
| | | When Subclock Is Not Used | When Subclock Is Used |
| Main clock oscillator | | Oscillation enabled | |
| Subclock oscillator | | – | Oscillation enabled |
| Internal oscillator | | Oscillation enabled | |
| PLL | | Operable | |
| CPU | | Stops operation | |
| Interrupt controller | | Operable | |
| 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 | |
| | UARTA0, UARTA1 | Operable | |
| A/D converter | | Operable | |
| Key interrupt function (KR) | | Operable | |
| 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. | |

16.4 IDLE1 Mode

16.4.1 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, 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 16-5 shows the operating 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 does 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.

- Cautions**
1. Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the IDLE1 mode.
 2. If the IDLE1 mode is set while an unmasked interrupt request signal is being held pending, the IDLE1 mode is released immediately by the pending interrupt request.

16.4.2 Releasing IDLE1 mode

The IDLE1 mode is released by a non-maskable interrupt request signal (NMI pin input, 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, power-on-clear circuit (POC), 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.

- 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 IDLE1 mode is not released.
 2. If eliminating digital noise is selected by using the NFC register and if the sampling clock is selected from $f_{xx}/64$, $f_{xx}/128$, $f_{xx}/256$, $f_{xx}/512$, and $f_{xx}/1024$, the IDLE1 mode cannot be released by the interrupt request signal of the INTP3 pin. For details, see 14.6.2 (4) Noise elimination control register (NFC).

- (a) If an interrupt request signal with a priority lower than that of the interrupt request currently being serviced is issued, the IDLE1 mode is released, but 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 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 16-4. 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 16-5. Operating Status in IDLE1 Mode

| Item | Setting of IDLE1 Mode | Operating Status | |
|-----------------------------|-----------------------|--|--|
| | | When Subclock Is Not Used | When Subclock Is Used |
| Main clock oscillator | | Oscillation enabled | |
| Subclock oscillator | | – | Oscillation enabled |
| Internal oscillator | | Oscillation enabled | |
| PLL | | Operable | |
| CPU | | Stops operation | |
| Interrupt controller | | Stops operation (but standby mode release is possible) | |
| Timer P (TMP0 to TMP3) | | Stops operation | |
| 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 the \overline{SCKBn} input clock is selected as the count clock ($n = 0, 1$) | |
| | UARTA0, UARTA1 | Stops operation (but UARTA0 is operable when the ASCKA0 input clock is selected) | |
| A/D converter | | Holds operation (conversion result held) ^{Note} | |
| Key interrupt function (KR) | | Operable | |
| 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.

16.5 IDLE2 Mode

16.5.1 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 16-7 shows the operating 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.

- Cautions 1. Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the IDLE2 mode.**
- 2. If the IDLE2 mode is set while an unmasked interrupt request signal is being held pending, the IDLE2 mode is released immediately by the pending interrupt request.**

16.5.2 Releasing IDLE2 mode

The IDLE2 mode is released by a non-maskable interrupt request signal (NMI pin input, 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, power-on-clear circuit (POC), low-voltage detector (LVI), or clock monitor (CLM)). The PLL returns to the operating status it was in 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.

- Cautions 1. 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.**
- 2. If eliminating digital noise is selected by using the NFC register and if the sampling clock is selected from $f_{xx}/64$, $f_{xx}/128$, $f_{xx}/256$, $f_{xx}/512$, and $f_{xx}/1024$, the IDLE2 mode cannot be released by the interrupt request signal of the INTP3 pin. For details, see 14.6.2 (4) Noise elimination control register (NFC).**

- (a) If an interrupt request signal with a priority lower than that of the interrupt request currently being serviced is issued, the IDLE2 mode is released, but 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 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 16-6. 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 16-7. Operating Status in IDLE2 Mode

| Setting of IDLE2 Mode | | Operating Status | |
|-----------------------------|----------------|--|--|
| | | When Subclock Is Not Used | When Subclock Is Used |
| Item | | | |
| Main clock oscillator | | Oscillation enabled | |
| Subclock oscillator | | – | Oscillation enabled |
| Internal oscillator | | Oscillation enabled | |
| PLL | | Stops operation | |
| CPU | | Stops operation | |
| Interrupt controller | | Stops operation (but standby mode release is possible) | |
| Timer P (TMP0 to TMP3) | | Stops operation | |
| 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 the \overline{SCKBn} input clock is selected as the count clock ($n = 0, 1$) | |
| | UARTA0, UARTA1 | Stops operation (but UARTA0 is operable when the ASCKA0 input clock is selected) | |
| A/D converter | | Holds operation (conversion result held) ^{Note} | |
| Key interrupt function (KR) | | Operable | |
| 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.

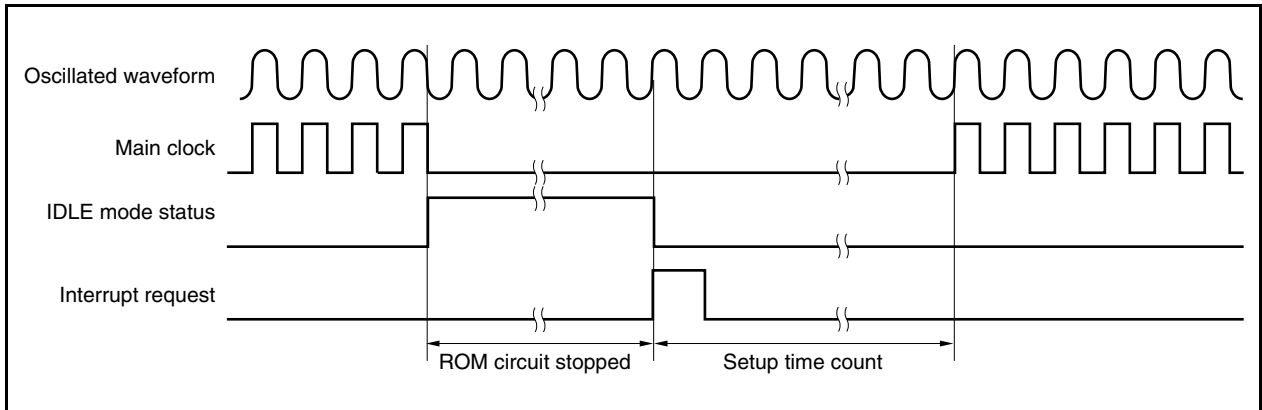
16.5.3 Securing setup time when releasing IDLE2 mode

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 is generated, the dedicated internal timer starts counting according to the OSTS register setting. When it overflows, the normal operation mode is restored.



(2) Release by reset ($\overline{\text{RESET}}$ pin input, WDT2RES generation)

This operation is the same as that of a normal reset.

The oscillation stabilization time is the initial value of the OSTS register, $2^{16}/f_x$.

16.6 STOP Mode

16.6.1 Setting and operation status

The STOP mode is set by setting the PSMR.PSM1 and PSMR.PSM0 bits to 01 or 11 and setting the PSC.STP bit to 1 in the normal operation mode.

In the STOP mode, the subclock oscillator continues operating 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 on-chip peripheral functions that operate with the clock oscillated by the subclock oscillator or an external clock continue operating.

Table 16-9 shows the operating 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, internal oscillator, and external clock are not used, the power consumption can be minimized with only leakage current flowing.

- Cautions**
1. **Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the STOP mode.**
 2. **If the STOP mode is set while an unmasked interrupt request signal is being held pending, the STOP mode is released immediately by the pending interrupt request.**

16.6.2 Releasing STOP mode

The STOP mode is released by a non-maskable interrupt request signal (NMI pin input, 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, power-on-clear circuit (POC), 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.

- Cautions**
1. **The interrupt request that is disabled by setting the PSC.NMI1M, PSC.NMI0M, and PSC.INTM bits to 1 becomes invalid and STOP mode is not released.**
 2. **If eliminating digital noise is selected by using the NFC register and if the sampling clock is selected from $f_{xx}/64$, $f_{xx}/128$, $f_{xx}/256$, $f_{xx}/512$, and $f_{xx}/1024$, the STOP mode cannot be released by the interrupt request signal of the INTP3 pin. For details, see 14.6.2 (4) Noise elimination control register (NFC).**

(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 currently being serviced is issued, the STOP mode is released, but 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 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 16-8. 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 16-9. Operating Status in STOP Mode

| Setting of STOP Mode | | Operating Status | |
|-----------------------------|----------------|---|--|
| | | When Subclock Is Not Used | When Subclock Is Used |
| Item | | | |
| Main clock oscillator | | Stops oscillation | |
| Subclock oscillator | | – | Oscillation enabled |
| Internal oscillator | | Oscillation enabled | |
| PLL | | Stops operation | |
| CPU | | Stops operation | |
| Interrupt controller | | Stops operation (but standby mode release is possible) | |
| Timer P (TMP0 to TMP3) | | Stops operation | |
| 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 the \overline{SCKBn} input clock is selected as the count clock ($n = 0, 1$) | |
| | UARTA0, UARTA1 | 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 | |
| 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 after the STOP mode is released are invalid. 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.

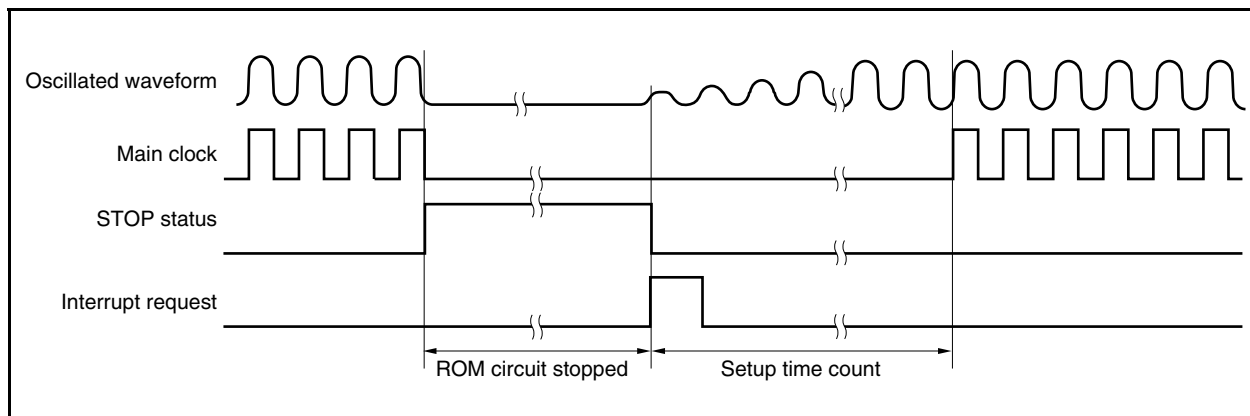
16.6.3 Securing oscillation stabilization time when releasing STOP mode

Secure the oscillation stabilization time for the main clock oscillator after releasing the STOP mode because the operation of the main clock oscillator stops after STOP mode is set.

(1) Releasing STOP mode by non-maskable interrupt request signal or unmasked maskable interrupt request signal

Secure the oscillation stabilization time by setting the OSTS register.

When the releasing source is generated, the dedicated internal timer starts counting according to the OSTS register setting. When it overflows, the normal operation mode is restored.



(2) Release by reset

This operation is the same as that of a normal reset.

The oscillation stabilization time is the initial value of the OSTS register, $2^{16}/f_x$.

16.7 Subclock Operation Mode

16.7.1 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 on 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 16-10 shows the operating 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, see 5.3 (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.**
Internal system clock (f_{CLK}) > Subclock (f_{XT}) \times 4

Remark Internal system clock (f_{CLK}): Clock generated from main clock (f_{XX}) in accordance with the settings of the CK2 to CK0 bits

16.7.2 Releasing subclock operation mode

The subclock operation mode is released by a reset signal (reset by \overline{RESET} pin input, WDT2RES signal, power-on-clear circuit (POC), low-voltage detector (LVI), or clock monitor (CLM)) when the CK3 bit is cleared to 0.

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, see 5.3 (1) Processor clock control register (PCC).

Table 16-10. Operating Status in Subclock Operation Mode

| Setting of Subclock Operation Mode Item | | Operating Status | |
|--|----------------|--------------------------------|--|
| | | When Main Clock Is Oscillating | When Main Clock Is Stopped |
| Subclock oscillator | | Oscillation enabled | |
| Internal oscillator | | Oscillation enabled | |
| PLL | | Operable | Stops operation ^{Note} |
| CPU | | Operable | |
| Interrupt controller | | Operable | |
| Timer P (TMP0 to TMP3) | | Operable | Stops operation |
| 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 the \overline{SCKBn} input clock is selected as the count clock ($n = 0, 1$) |
| | UARTA0, UARTA1 | 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 | |
| Port function | | Settable | |
| Internal data | | Settable | |

Note Be sure to stop the PLL (PLLCTL.PLLON = 0) before stopping the main clock.

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 (see 3.4.8 (2)).

16.8 Sub-IDLE Mode

16.8.1 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 operating 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 current consumption can be reduced to a level as low as that in the STOP mode.

Table 16-12 shows the operating status in the sub-IDLE mode.

- Cautions**
- 1. Following the store instruction to the PSC register for setting the sub-IDLE mode, insert five or more NOP instructions.**
 - 2. If the sub-IDLE mode is set while an unmasked interrupt request signal is being held pending, the sub-IDLE mode is then released immediately by the pending interrupt request.**

16.8.2 Releasing sub-IDLE mode

The sub-IDLE mode is released by a non-maskable interrupt request signal (NMI pin input, 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, power-on-clear circuit (POC), low-voltage detector (LVI), or clock monitor (CLM)). The PLL returns to the operating status it was in before the sub-IDLE mode was set.

When the sub-IDLE mode is released by an interrupt request signal, the subclock operation mode is set.

(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. The 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.

2. When the sub-IDLE mode is released, 12 cycles of the subclock (about 366 μs) elapse from when the interrupt request signal that releases the sub-IDLE mode is generated to when the mode is released.

3. If eliminating digital noise is selected by using the NFC register and if the sampling clock is selected from fxx/64, fxx/128, fxx/256, fxx/512, and fxx/1024, the sub-IDLE mode cannot be released by the interrupt request signal of the INTP3 pin. For details, see 14.6.2 (4) Noise elimination control register (NFC).

(a) If an interrupt request signal with a priority lower than that of the interrupt request currently being serviced is issued, the sub-IDLE mode is released, but 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 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.

Table 16-11. 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 16-12. Operating Status in Sub-IDLE Mode

| Setting of Sub-IDLE Mode | | Operating Status | |
|-----------------------------|----------------|---|---|
| | | When Main Clock Is Oscillating | When Main Clock Is Stopped |
| Subclock oscillator | | Oscillation enabled | |
| Internal oscillator | | Oscillation enabled | |
| PLL | | Operable | Stops operation ^{Note 1} |
| CPU | | Stops operation | |
| Interrupt controller | | Stops operation (but standby mode release is possible) | |
| Timer P (TMP0 to TMP3) | | Stops operation | |
| Timer Q (TMQ0) | | Stops operation | |
| Timer M (TMM0) | | 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 the \overline{SCKBn} input clock is selected as the count clock ($n = 0, 1$) | |
| | UARTA0, UARTA1 | Stops operation (but UARTA0 is operable when the ASCKA0 input clock is selected) | |
| A/D converter | | Holds operation (conversion result held) ^{Note 2} | |
| Key interrupt function (KR) | | Operable | |
| 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 (PLLCTL.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 17 RESET FUNCTIONS

17.1 Overview

The following reset functions are available.

(1) Four kinds of reset sources

- External reset input via the $\overline{\text{RESET}}$ pin
- Reset via the watchdog timer 2 (WDT2) overflow (WDT2RES)
- System reset via the comparison of the low-voltage detector (LVI) supply voltage and detected voltage
- System reset via the detecting clock monitor (CLM) oscillation stop
- System reset via the power-on-clear circuit

After a reset is released, the source of the reset can be confirmed with the reset source flag register (RESF).

(2) Emergency operation mode

If the WDT2 overflows during the main clock oscillation stabilization time inserted after reset, a main clock oscillation anomaly is judged and the CPU starts operating on the internal oscillation clock.

Caution When the CPU is being operated with the internal oscillation clock, access to the register in which a wait state is generated is prohibited. For the register in which a wait state is generated, see 3.4.8 (2) Accessing specific on-chip peripheral I/O registers.

17.2 Registers to Check Reset Source

The V850ES/HF2 has four kinds of reset sources. After a reset has been released, the source of the reset that occurred can be checked with the reset source flag register (RESF).

(1) Reset source flag register (RESF)

The RESF register is a special register that can be written only by a combination of specific sequences (see **3.4.7 Special registers**).

The RESF register indicates the source from which a reset signal is generated.

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

$\overline{\text{RESET}}$ pin input or POC reset sets 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 The value of the RESF register is cleared to 00H when a reset is executed via the $\overline{\text{RESET}}$ pin. When a reset is executed by watchdog timer 2 (WDT2), low-voltage detector (LVI), or clock monitor (CLM), the reset flags of this register (WDT2RF bit, CLMRF bit, and LVIRF bit) are set. However, other sources are retained.

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.

17.3 Operation

17.3.1 Reset operation via $\overline{\text{RESET}}$ 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.

Table 17-1. 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}) | Crystal oscillation | Oscillation continues |
| | RC oscillation | Oscillation stops Oscillation starts |
| Internal oscillator | 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 (initialized to $f_x/8$) |
| CPU | Initialized | Program execution starts after securing oscillation stabilization time |
| Watchdog timer 2 | Operation stops (initialized to 0) | Operation starts |
| Internal RAM | Undefined if power-on reset or CPU access and reset input conflict (data is damaged). Otherwise value immediately after reset input is retained ^{Note 1} . | |
| I/O lines (ports/alternate-function pins) | High impedance ^{Note 2} | |
| 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 |

<R> **Notes 1.** The firmware of the V850ES/HF2 uses a part of the internal RAM after the internal system reset status has been released because it supports a boot swap function. Therefore, the contents of some RAM areas are not retained after power-on reset. For details, see **17.4 Operation After Reset Release**.

2. When the power is turned on, the following pin may output an undefined level temporarily even during reset.

- P53/KR3/TIQ00/TOQ00/DDO pin

Caution The OCDM register is initialized by the $\overline{\text{RESET}}$ pin input. Therefore, note with caution that, if a high level is input to the P05/ $\overline{\text{DRST}}$ pin after a reset release before the OCDM.OCDM0 bit is cleared, the on-chip debug mode is entered. For details, see CHAPTER 4 PORT FUNCTIONS.

Figure 17-1. Timing of Reset Operation by $\overline{\text{RESET}}$ Pin Input

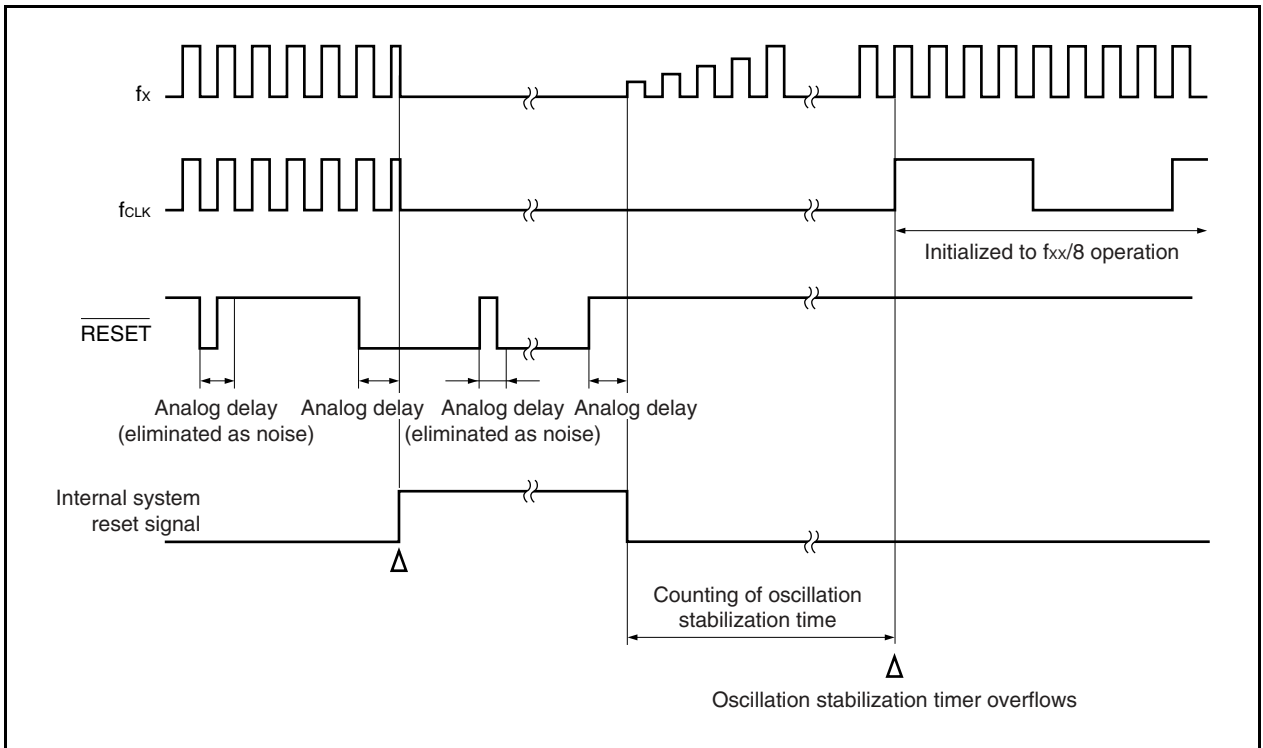
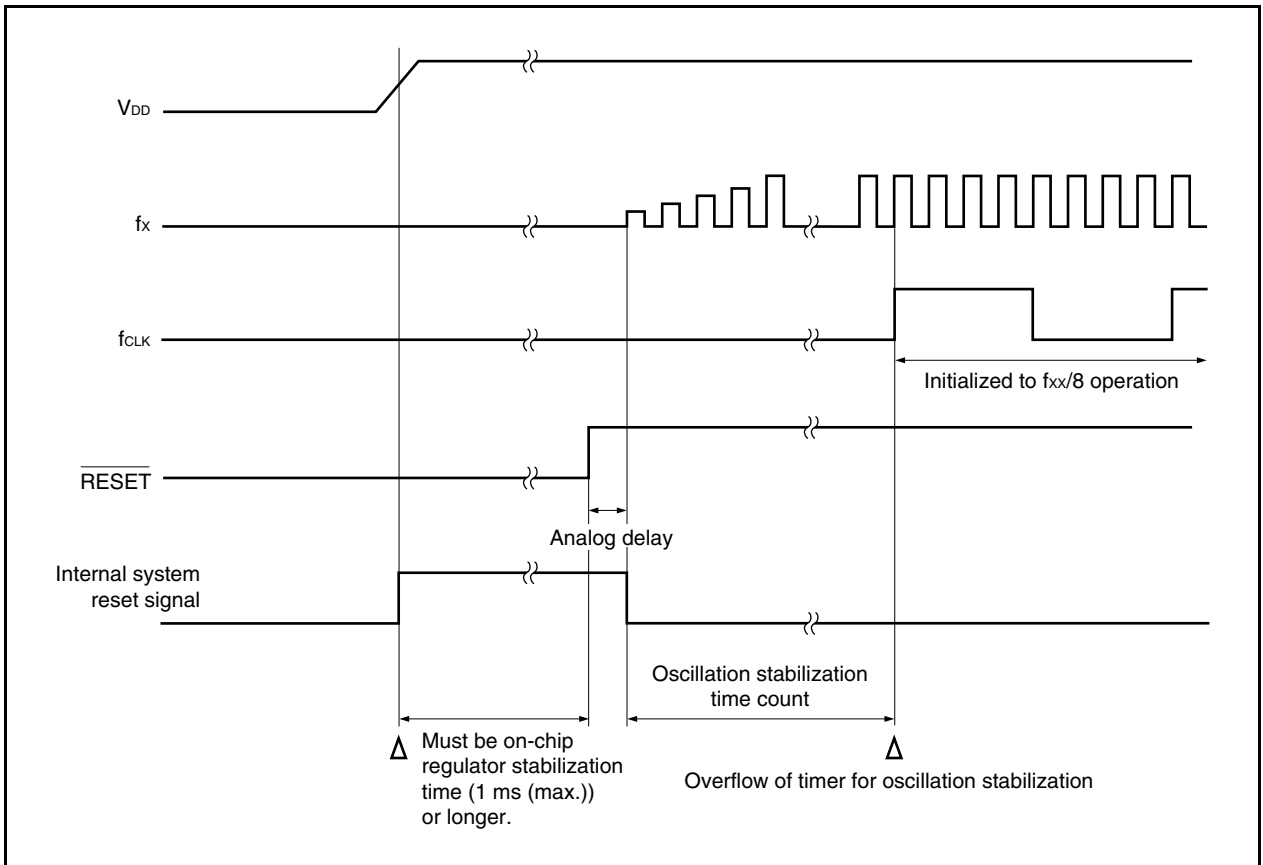


Figure 17-2. Timing of Power-on Reset Operation



17.3.2 Reset operation by watchdog timer 2

When watchdog timer 2 is set to the reset operation mode due to overflow, upon watchdog timer 2 overflow (WDT2RES signal generation), a system reset is executed 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.

The main clock oscillator is stopped during the reset period.

Table 17-2. Hardware Status During Watchdog Timer 2 Reset Operation

| Item | | During Reset | After Reset |
|--|---------------------|--|---|
| Main clock oscillator (f_x) | | Oscillation stops | Oscillation starts |
| Subclock oscillator (f_{XT}) | Crystal oscillation | Oscillation continues | |
| | RC oscillation | Oscillation stops | Oscillation starts |
| Internal oscillator | | Oscillation stops | Oscillation starts |
| Peripheral clock (f_{xx} to $f_{xx}/1,024$) | | Operation stops | Operation starts after securing oscillation stabilization time |
| Internal system clock (f_{xx}), CPU clock (f_{CPU}) | | Operation stops | Operation starts after securing oscillation stabilization time (initialized to $f_{xx}/8$) |
| CPU | | Initialized | Program execution after securing oscillation stabilization time |
| Watchdog timer 2 | | Operation stops (initialized to 0) | Operation starts |
| Internal RAM | | Undefined if power-on reset or CPU access and reset input conflict (data is damaged). Otherwise value immediately after reset input is retained ^{Note} . | |
| I/O lines (ports/alternate-function pins) | | 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. |

<R> **Note** The firmware of the V850ES/HF2 uses a part of the internal RAM after the internal system reset status has been released because it supports a boot swap function. Therefore, the contents of some RAM areas are not retained after power-on reset. For details, see **17.4 Operation After Reset Release**.

17.3.3 Reset operation by power-on-clear circuit

The supply voltage and detection voltage are compared when the power-on-clear operation is enabled. If the supply voltage drops below the detection voltage (including when power is applied), the system is reset and each hardware unit is initialized to the default status.

The reset status lasts since the voltage drop has been detected until the supply voltage rises above the detection voltage, and then is automatically cleared. After the reset status is cleared, time to stabilize oscillation of the main clock oscillator (default value of OSTS register: $2^{16}/f_x$) elapses, and then the CPU starts program execution. For details, see **CHAPTER 19 POWER-ON-CLEAR CIRCUIT**.

17.3.4 Reset operation by low-voltage detector

When LVI operation is enabled and when the LVIM.LVIMD bit is set to “1”, the supply voltage and detection voltage are compared. If the supply voltage drops below the detection voltage, the system is reset and each hardware unit is initialized to the default status.

The reset status lasts from detection of the voltage drop until the supply voltage rises above the detection voltage, and then is automatically cleared. After the reset status is cleared, time to stabilize oscillation of the main clock oscillator (default value of OSTS register: $2^{16}/f_x$) elapses, and then the CPU starts program execution.

For details, see **CHAPTER 20 LOW-VOLTAGE DETECTOR**.

17.3.5 Reset operation by clock monitor

When the clock monitor operation is enabled, the main clock is monitored by using the sampling clock (internal oscillator). If stoppage of the main clock is detected, the system is reset and each hardware unit is initialized to the default status.

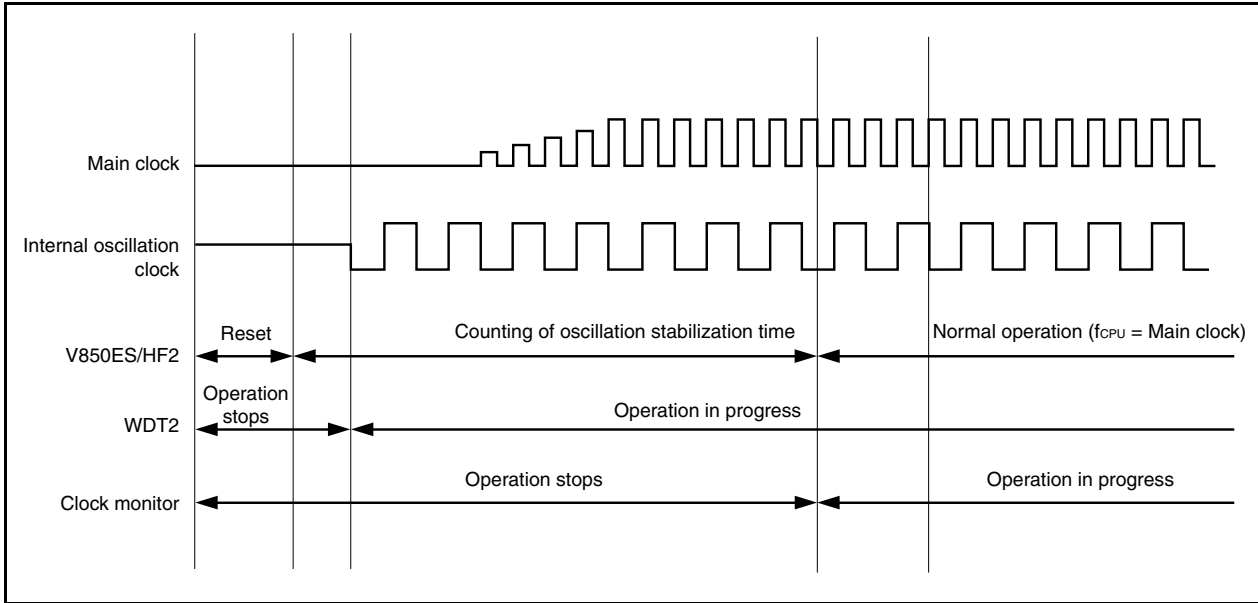
For details, see **CHAPTER 18 CLOCK MONITOR**.

<R> 17.4 Operation After Reset Release

After the reset is released, the main clock starts oscillation and oscillation stabilization time (OSTS register initial value: $2^{16}/f_x$) is secured, and the CPU starts program execution.

WDT2 immediately begins to operate after a reset has been released using the internal oscillation clock as a source clock.

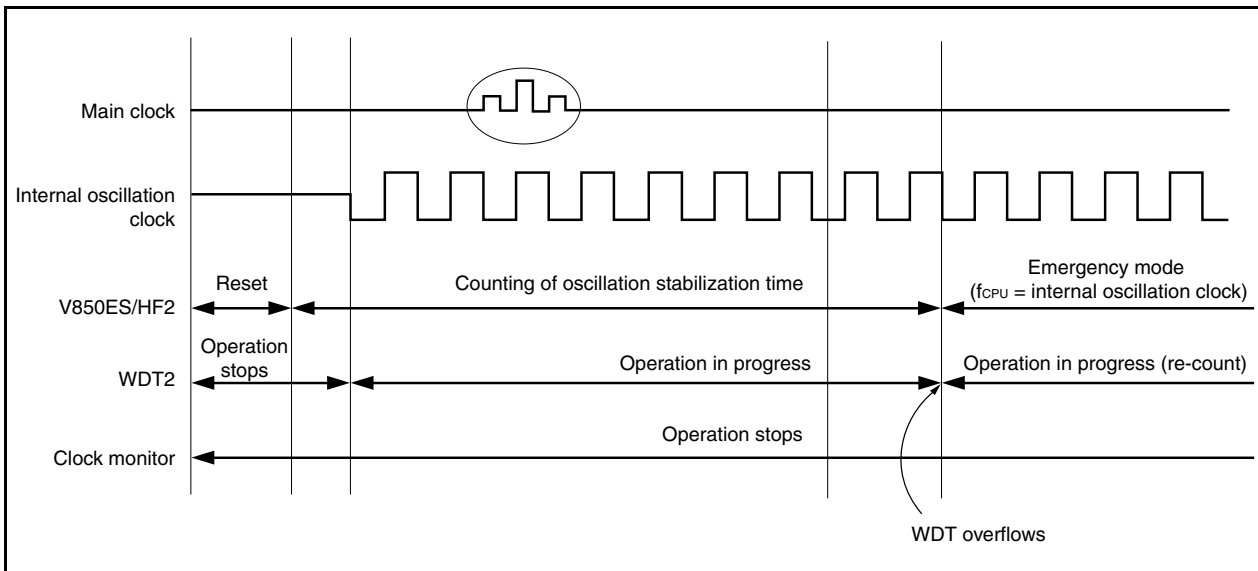
Figure 17-3. Operation After Reset Release



(1) Emergent operation mode

If an anomaly occurs in the main clock before oscillation stabilization time is secured, the WDT2 overflows before executing the CPU program. At this time, the CPU starts program execution by using the internal oscillation clock as the source clock.

Figure 17-4. Operation After Reset Release

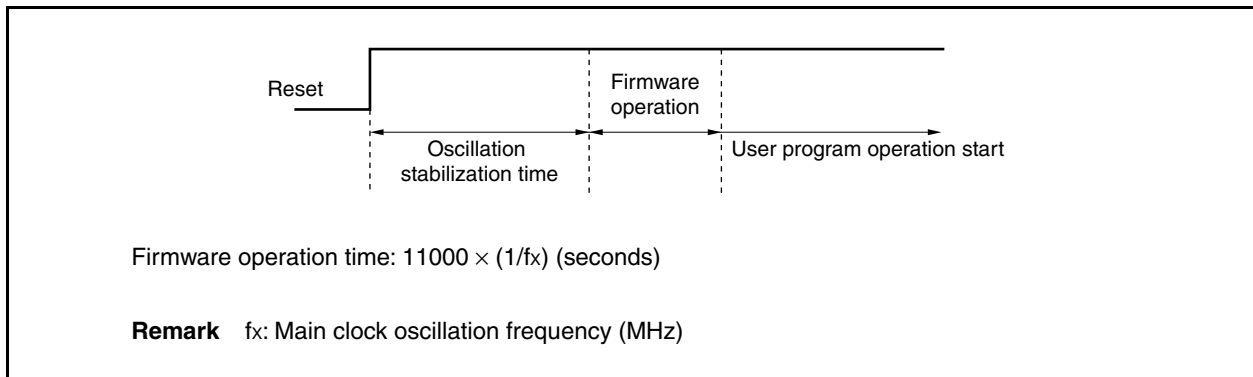


The CPU operation clock states can be checked with the CPU operation clock status register (CCLS).

(2) Firmware operation

In the V850ES/HF2, after a reset is released, the on-chip firmware operates before starting the user program to support the boot switch function.

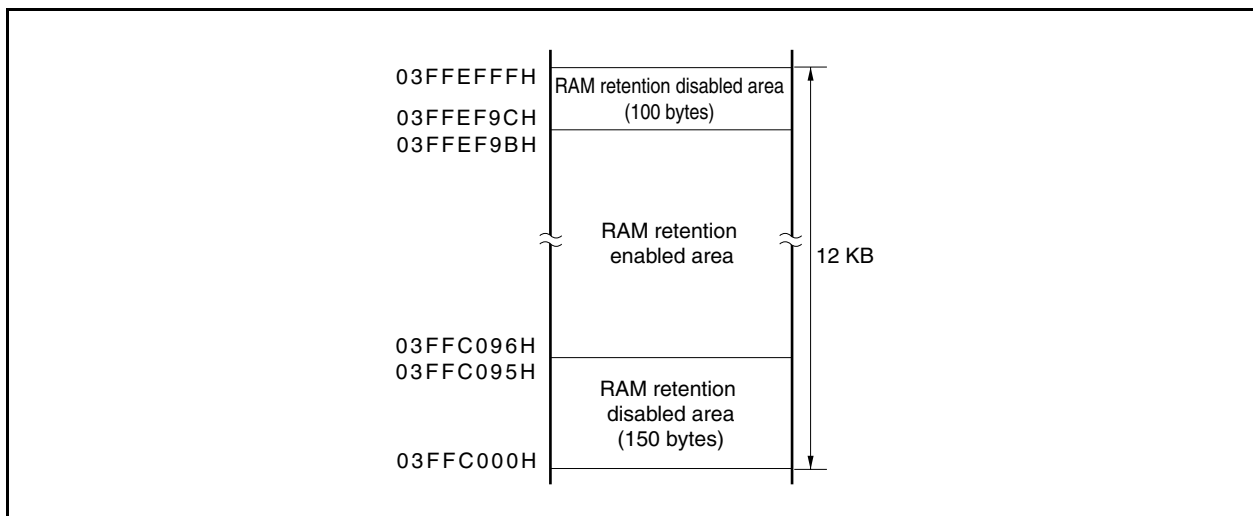
Figure 17-5. Firmware Operation



Since the firmware uses a portion of the internal RAM, the contents of the following RAM areas are not retained through a reset even in power on status.

- Version with 12 KB RAM: 03FFC000H to 03FFC095H, 03FFEF9CH to 03FFEFFFH

Figure 17-6. RAM Retention Enabled Area



CHAPTER 18 CLOCK MONITOR

18.1 Functions

The clock monitor samples the main clock by using the internal oscillation clock and generates a reset request signal when oscillation of the main clock is stopped.

Once the operation of the clock monitor has been enabled by an operation enable flag, it cannot be cleared to 0 by any means other than reset.

When a reset by the clock monitor occurs, the RESF.CLMRF bit is set. For details on the RESF register, see **17.2 Registers to Check Reset Source**.

The clock monitor automatically stops under the following conditions.

- During oscillation stabilization time after STOP mode is released
- When the main clock is stopped (from when the PCC.MCK bit = 1 during subclock operation, until the PCC.CLS bit = 0 during main clock operation)
- When the sampling clock (internal oscillation clock) is stopped
- When the CPU operates with the internal oscillation clock

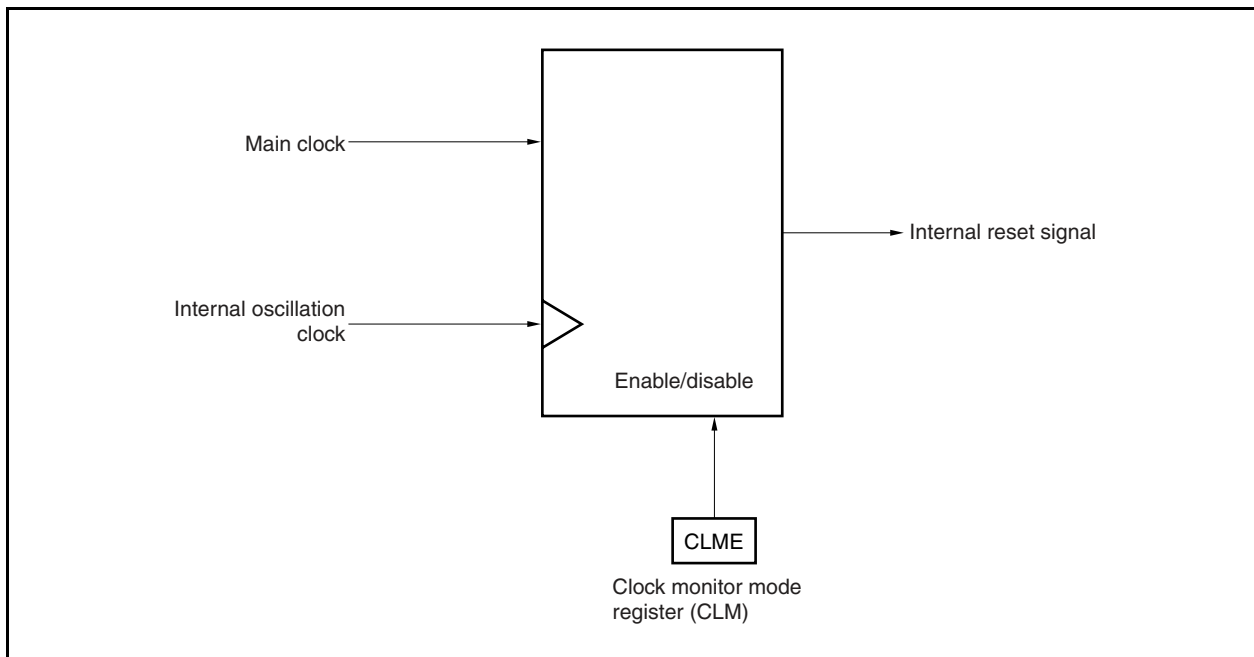
18.2 Configuration

The clock monitor includes the following hardware.

Table 18-1. Configuration of Clock Monitor

| Item | Configuration |
|------------------|-----------------------------------|
| Control register | Clock monitor mode register (CLM) |

Figure 18-1. Block Diagram of Clock Monitor



18.3 Register

The clock monitor is controlled by the clock monitor mode register (CLM).

(1) Clock monitor mode register (CLM)

The CLM register is a special register. This can be written only in a special combination of sequences (see **3.4.7 Special registers**).

This register is used to set the operation mode of the clock monitor.

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

Reset sets this register to 00H.

| | | | | | | | | |
|-----|------------------|---|--------------------|---|---|---|---|------|
| | After reset: 00H | R/W | Address: FFFFF870H | | | | | |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CLM | 0 | 0 | 0 | 0 | 0 | 0 | 0 | CLME |
| | CLME | Clock monitor operation enable or disable | | | | | | |
| | 0 | Disable clock monitor operation. | | | | | | |
| | 1 | Enable clock monitor operation. | | | | | | |

- Cautions**
1. Once the CLME bit has been set to 1, it cannot be cleared to 0 by any means other than reset.
 2. When a reset by the clock monitor occurs, the CLME bit is cleared to 0 and the RESF.CLMRF bit is set to 1.

18.4 Operation

This section explains the functions of the clock monitor. The start and stop conditions are as follows.

<Start condition>

Enabling operation by setting the CLM.CLME bit to 1

<Stop conditions>

- While oscillation stabilization time is being counted after STOP mode is released
- When the main clock is stopped (from when PCC.MCK bit = 1 during subclock operation to when PCC.CLS bit = 0 during main clock operation)
- When the sampling clock (internal oscillation clock) is stopped
- When the CPU operates using the internal oscillation clock

**Table 18-2. Operation Status of Clock Monitor
(When CLM.CLME Bit = 1, During Internal Oscillation Clock Operation)**

| CPU Operating Clock | Operation Mode | Status of Main Clock | Status of Internal Oscillation Clock | Status of Clock Monitor |
|--|--------------------|----------------------|--------------------------------------|----------------------------|
| Main clock | HALT mode | Oscillates | Oscillates ^{Note 1} | Operates ^{Note 2} |
| | IDLE1, IDLE2 modes | Oscillates | Oscillates ^{Note 1} | Operates ^{Note 2} |
| | STOP mode | Stops | Oscillates ^{Note 1} | Stops |
| Subclock (MCK bit of PCC register = 0) | Sub-IDLE mode | Oscillates | Oscillates ^{Note 1} | Operates ^{Note 2} |
| Subclock (MCK bit of PCC register = 1) | Sub-IDLE mode | Stops | Oscillates ^{Note 1} | Stops |
| Internal oscillation clock | – | Stops | Oscillates ^{Note 1} | Stops |
| During reset | – | Stops | Stops | Stops |

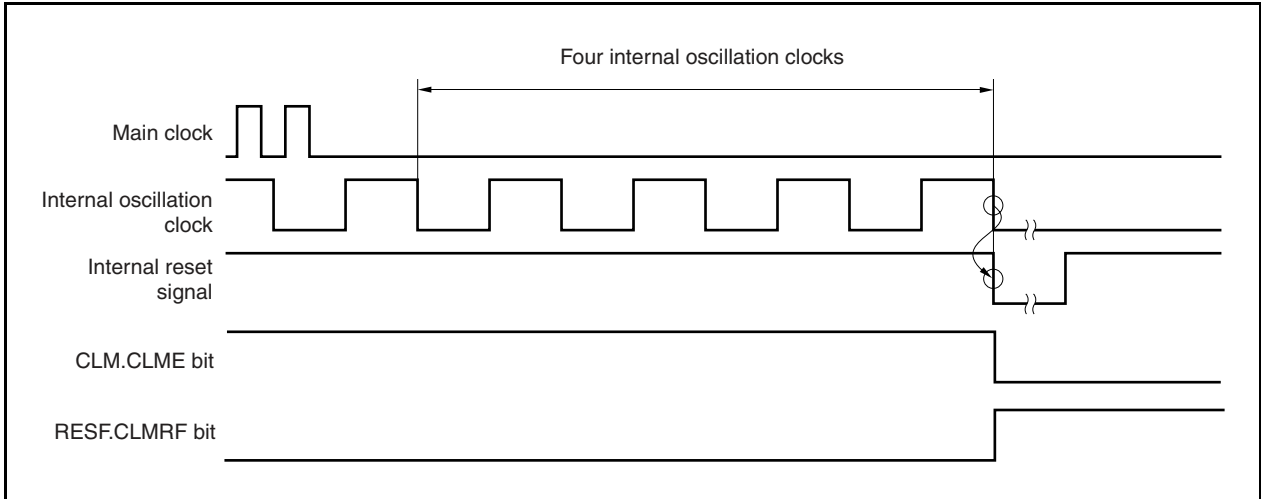
Notes 1. The internal oscillator can be stopped by using the option byte function (see **CHAPTER 23**) to enable the internal oscillator to stop, and setting the RCM.RSTOP bit to 1.

2. The clock monitor is stopped while the internal oscillator is stopped.

(1) Operation when main clock oscillation is stopped (CLME bit = 1)

If oscillation of the main clock is stopped when the CLME bit = 1, an internal reset signal is generated as shown in Figure 18-2.

Figure 18-2. Reset Period Due to That Oscillation of Main Clock Is Stopped

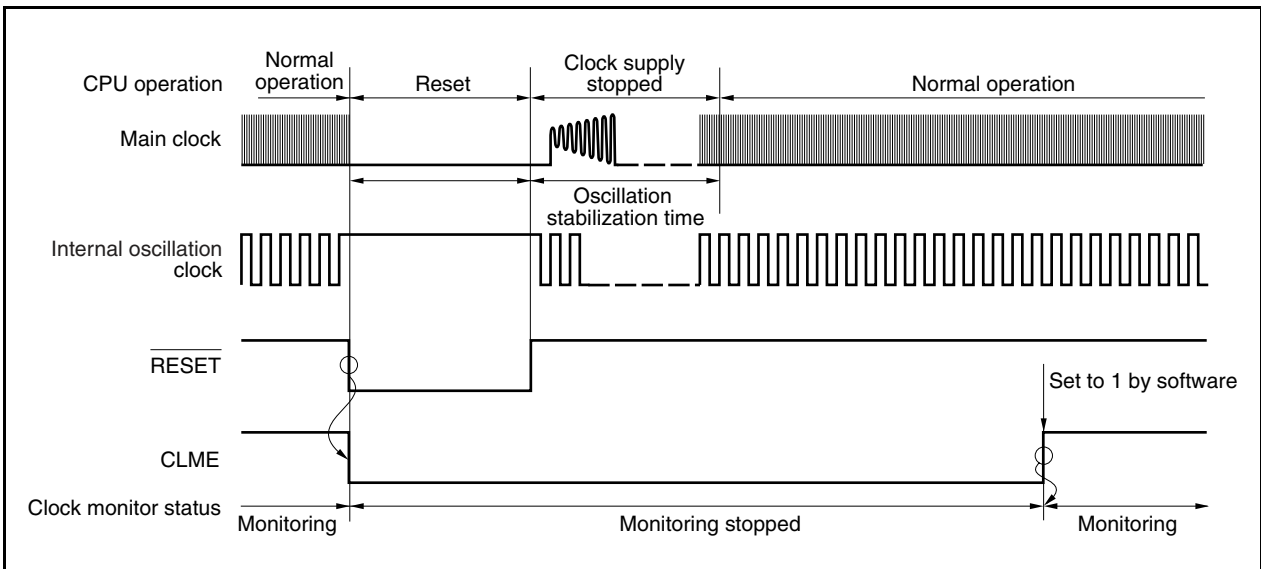


(2) Clock monitor status after $\overline{\text{RESET}}$ input

$\overline{\text{RESET}}$ input clears the CLM.CLME bit to 0 and stops the clock monitor operation. When CLME bit is set to 1 by software at the end of the oscillation stabilization time of the main clock, monitoring is started.

Figure 18-3. Clock Monitor Status After $\overline{\text{RESET}}$ Input

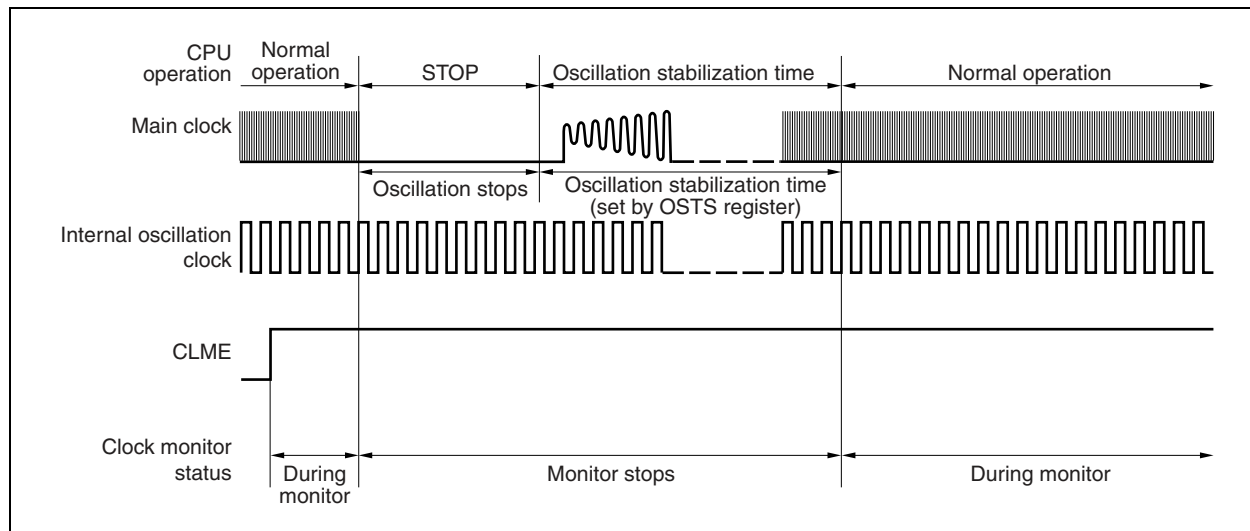
(CLM.CLME bit = 1 is set after $\overline{\text{RESET}}$ input and at the end of main clock oscillation stabilization time)



(3) Operation in STOP mode or after STOP mode is released

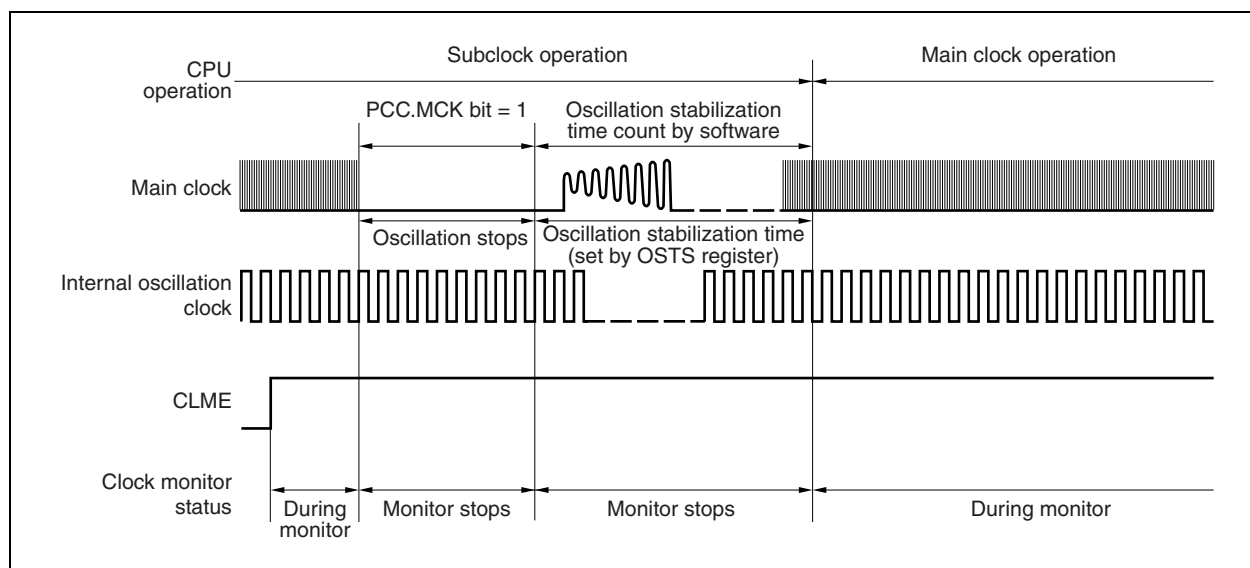
If the STOP mode is set with the CLM.CLME bit = 1, the monitor operation is stopped in the STOP mode and while the oscillation stabilization time is being counted. After the oscillation stabilization time, the monitor operation is automatically started.

Figure 18-4. Operation in STOP Mode or After STOP Mode Is Released

**(4) Operation when main clock is stopped (arbitrary)**

During subclock operation (PCC.CLS bit = 1) or when the main clock is stopped by setting the PCC.MCK bit to 1, the monitor operation is stopped until the main clock operation is started (PCC.CLS bit = 0). The monitor operation is automatically started when the main clock operation is started.

Figure 18-5. Operation When Main Clock Is Stopped (Arbitrary)

**(5) Operation while CPU is operating on internal oscillation clock (CCLS.CCLS bit = 1)**

The monitor operation is not stopped when the CCLS bit is 1, even if the CLME bit is set to 1.

CHAPTER 19 POWER-ON-CLEAR CIRCUIT

19.1 Function

Functions of the power-on-clear (POC) circuit are shown below.

- Generates a reset signal upon power application.
- Compares the supply voltage (V_{DD}) and detection voltage (V_{POC0}), and generates a reset signal when $V_{DD} < V_{POC0}$ (detection voltage (V_{POC0}): $3.7\text{ V} \pm 0.2\text{ V}$).

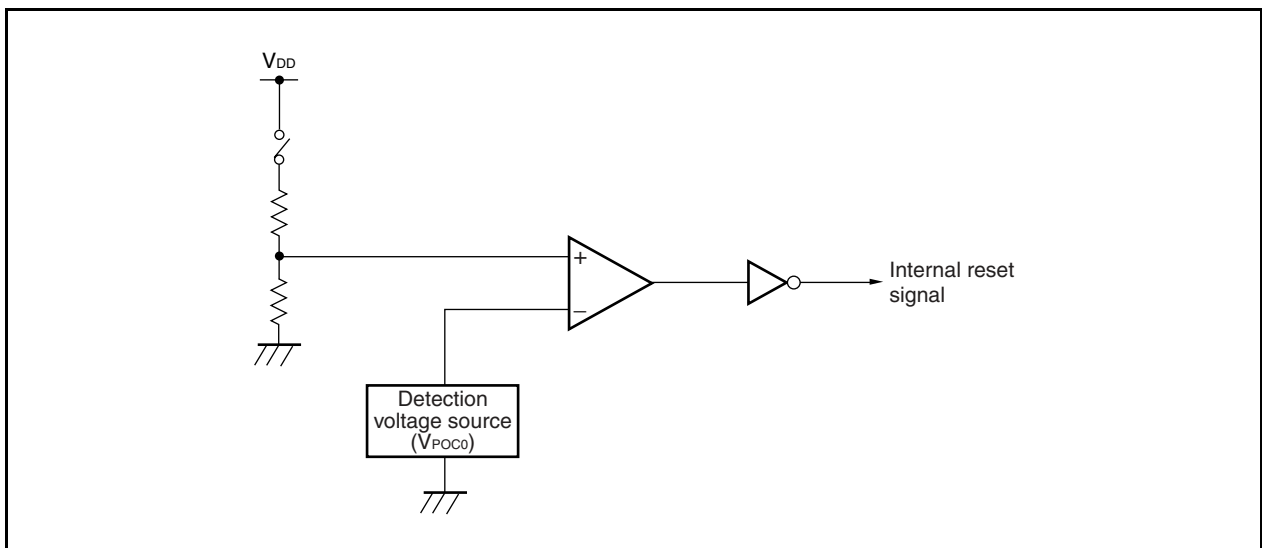
Remarks 1. The V850ES/HF2 has plural internal hardware units that generate an internal reset signal. When the system is reset by watchdog timer 2 (WDT2RES), low-voltage detector (LVI), or clock monitor (CLM), a flag corresponding to the reset source is allocated to the reset source flag register (RESF). The RESF register is not cleared when an internal reset signal is generated by WDT2RES, LVI, or clock monitor, and its flag corresponding to the reset source is set to 1. For details of the RESF register, see **CHAPTER 17 RESET FUNCTIONS**.

2. The time from power application to starting program execution is “Time from power application to releasing reset + 16 ms” if the operating frequency of a resonator externally connected is 5 MHz. However, it varies depending on the external cause (such as a status of supply voltage to the microcontroller and the stabilization time of the resonator).

19.2 Configuration

The block diagram is shown below.

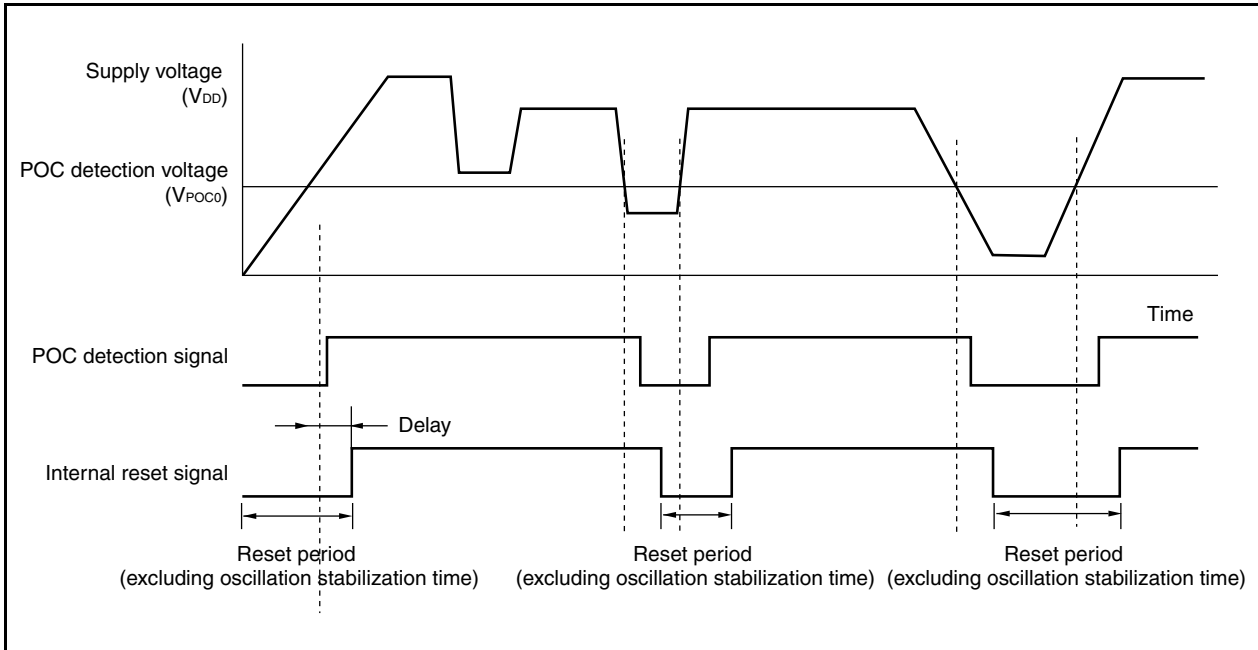
Figure 19-1. Block Diagram of Power-on-Clear Circuit



19.3 Operation

When the supply voltage and detection voltage are compared and if the supply voltage is lower than the detection voltage (including at power application), the system is reset and each hardware is returned to the specific status.

Figure 19-2. Timing of Reset Signal Generation by Power-on-Clear Circuit



CHAPTER 20 LOW-VOLTAGE DETECTOR

20.1 Functions

The low-voltage detector (LVI) has the following functions.

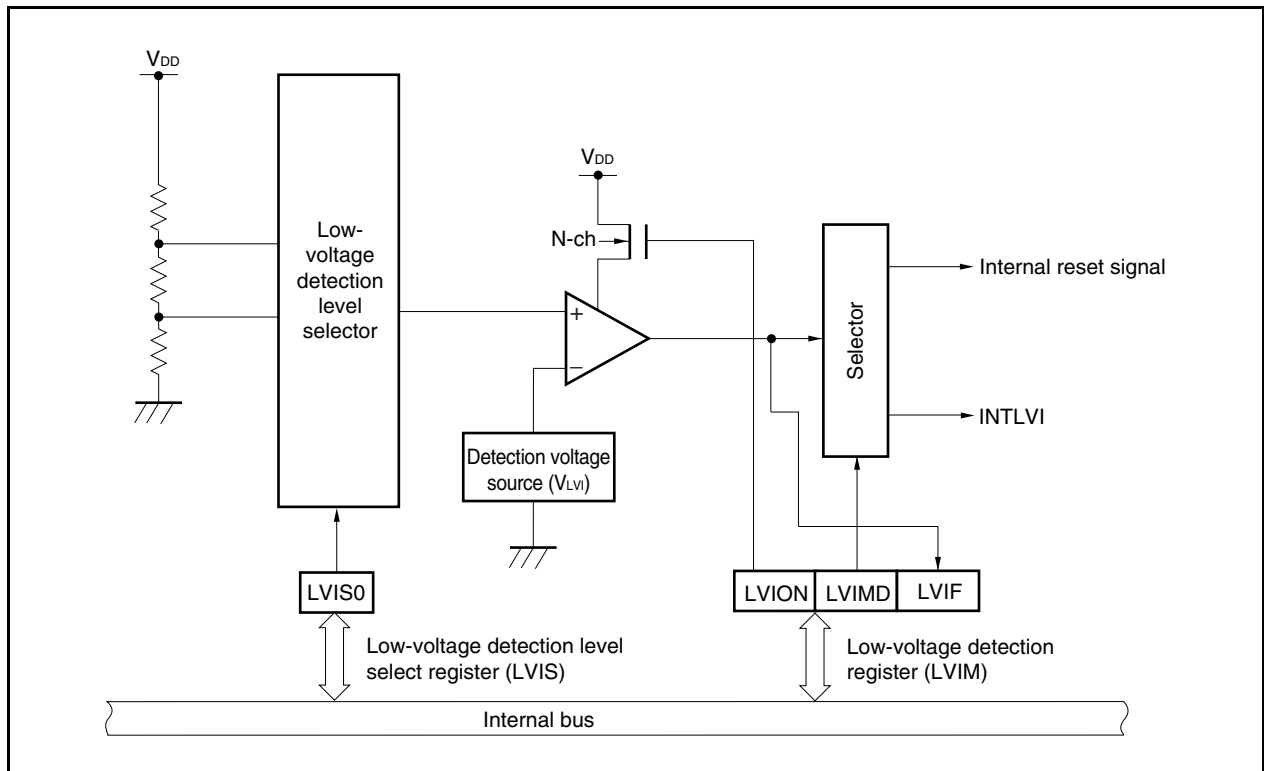
- Compares the supply voltage (V_{DD}) and detection voltage (V_{LVI}) and generates an interrupt request signal or internal reset signal when $V_{DD} < V_{LVI}$.
- The level of the supply voltage to be detected can be changed by software (in two steps).
- An interrupt request signal or internal reset signal can be selected.
- Can operate in STOP mode.
- Operation can be stopped by software.

If the low-voltage detector is used to generate a reset signal, the RESF.LVIRF bit is set to 1 when the reset signal is generated. For details of the RESF register, see **CHAPTER 17 RESET FUNCTIONS**.

20.2 Configuration

The block diagram is shown below.

Figure 20-1. Block Diagram of Low-Voltage Detector



20.3 Registers

(1) Low-voltage detection register (LVIM)

The LVIM register is used to enable or disable low voltage detection, and to set the operation mode of the low-voltage detector. The LVIM register is a special register. It can be written only by a combination of specific sequences (see 3.4.7 Special registers).

This register can be read or written in 8-bit or 1-bit units. However, bit 0 is read-only.

| | | | | | | | | | |
|------------------|---|--------------------|---|---|---|---|---|-------|------|
| 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 | Low voltage detection operation enable or disable | | | | | | | | |
| 0 | Disable operation. | | | | | | | | |
| 1 | Enable operation. | | | | | | | | |
| LVIMD | Selection of operation mode of low voltage detection | | | | | | | | |
| 0 | Generate interrupt request signal INTLVI when supply voltage < detection voltage. | | | | | | | | |
| 1 | Generate internal reset signal LVIRES when supply voltage < detection voltage. | | | | | | | | |
| LVIF | Low voltage detection flag | | | | | | | | |
| 0 | When supply voltage > detection voltage, or when operation is disabled | | | | | | | | |
| 1 | Supply voltage < detection voltage | | | | | | | | |

- Cautions**
1. After setting the LVION bit to 1, wait for 0.2 ms (MAX.) before checking the voltage using the LVIF bit.
 2. The value of the LVIF flag is output as the output signal INTLVI when the LVION bit = 1 and LVIMD bit = 0.
 3. Be sure to clear bits 2 to 6 to "0".
 4. The low-voltage detector cannot be stopped until a reset request due to something other than low-voltage detection is generated after the LVIM.LVION and LVIM.LVIMD bits are set to 1.

<R>

(2) Low-voltage detection level select register (LVIS)

The LVIS register is used to select the level of low voltage to be detected.

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

| | | | | | | | | | |
|------------------|-----|--------------------|---|---|---|---|---|---|-------|
| 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 | Detection level |
| 0 | 4.4 V \pm 0.2 V |
| 1 | 4.2 V \pm 0.2 V |

Cautions 1. This register cannot be written until a reset request due to something other than low-voltage detection is generated after the LVIM.LVION and LVIM.LVIMD bits are set to 1.

2. Be sure to clear bits 1 to 7 to "0".

(3) Internal RAM data status register (RAMS)

The RAMS register is a flag register that indicates whether the internal RAM is valid or not. The RAMS register is a special register. It can be written only by a combination of specific sequences (see 3.4.7 **Special registers**).

For the RAMS register, see 20.5 **RAM Retention Voltage Detection Operation**.

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

Caution The following shows the specific sequence after reset.

- **Setting conditions: Detection of voltage lower than detection level**
 - Set by instruction
 - Generation of reset signal by watchdog timer overflow
 - Generation of reset signal while RAM is being accessed
 - Generation of reset signal by clock monitor
- **Clearing condition: Writing of 0 in specific sequence**

| | | | | | | | | | |
|------------------|-----|--------------------|---|---|---|---|---|---|------|
| After reset: 01H | 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 |

20.4 Operation

Depending on the setting of the LVIM.LVIMD bit, an interrupt request signal (INTLVI) or an internal reset signal is generated.

20.4.1 To use for internal reset signal

<To start operation>

<1> Mask the interrupt of LVI.

<2> Select the voltage to be detected by using the LVIS.LVIS0 bit.

<3> Set the LVIM.LVION bit to 1 (to enable operation).

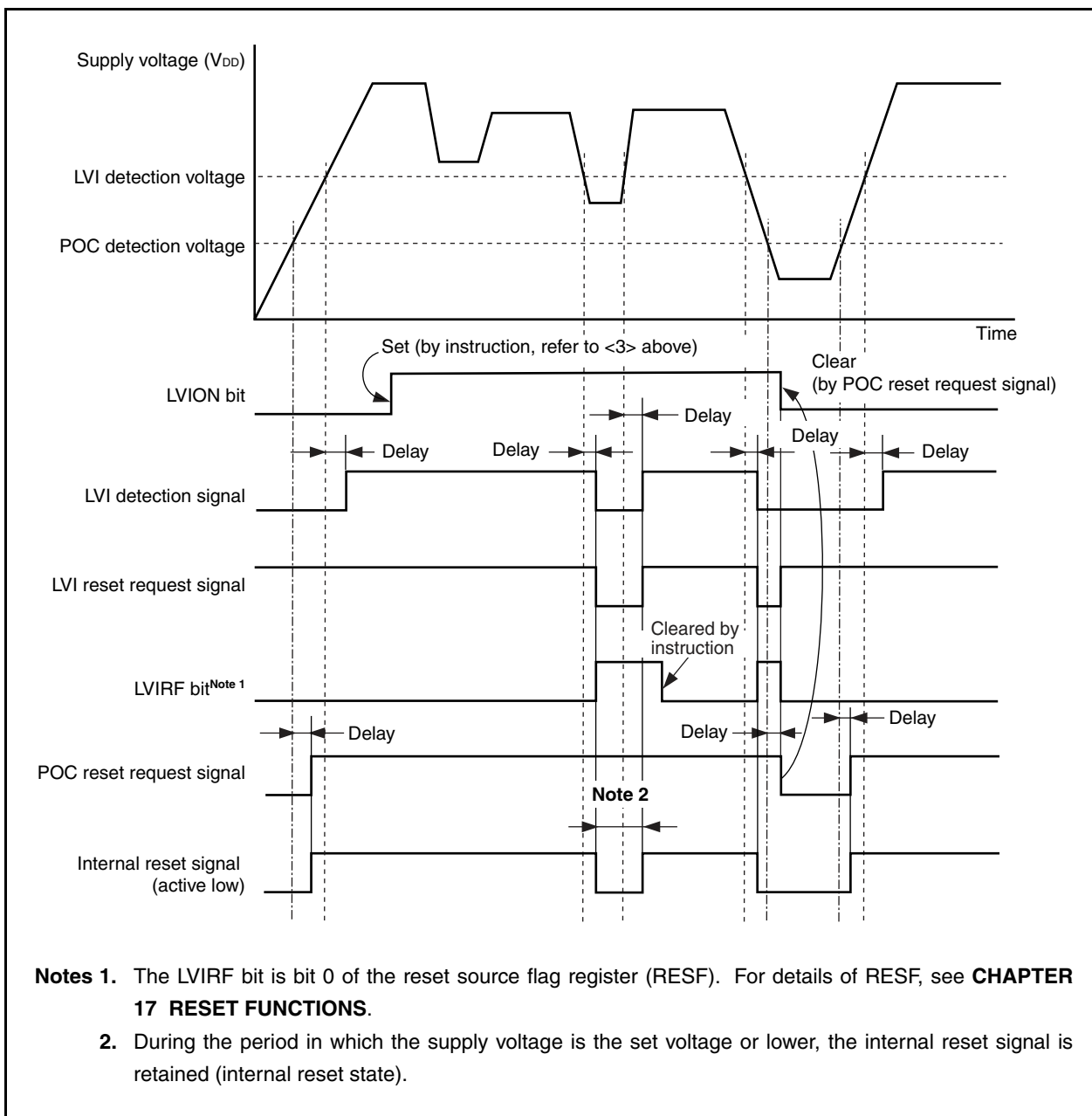
<4> Insert a wait cycle of 0.2 ms MAX. by software.

<5> By using the LVIM.LVIF bit, check if the supply voltage > detection voltage.

<6> Set the LVIM.LVIMD bit to 1 (to generate an internal reset signal).

Caution If the LVIMD bit is set to 1, the contents of the LVIM and LVIS registers cannot be changed until a reset request other than LVI is generated.

Figure 20-2. Operation Timing of Low-Voltage Detector (LVIMD Bit = 1)



<R>

20.4.2 To use for interrupt

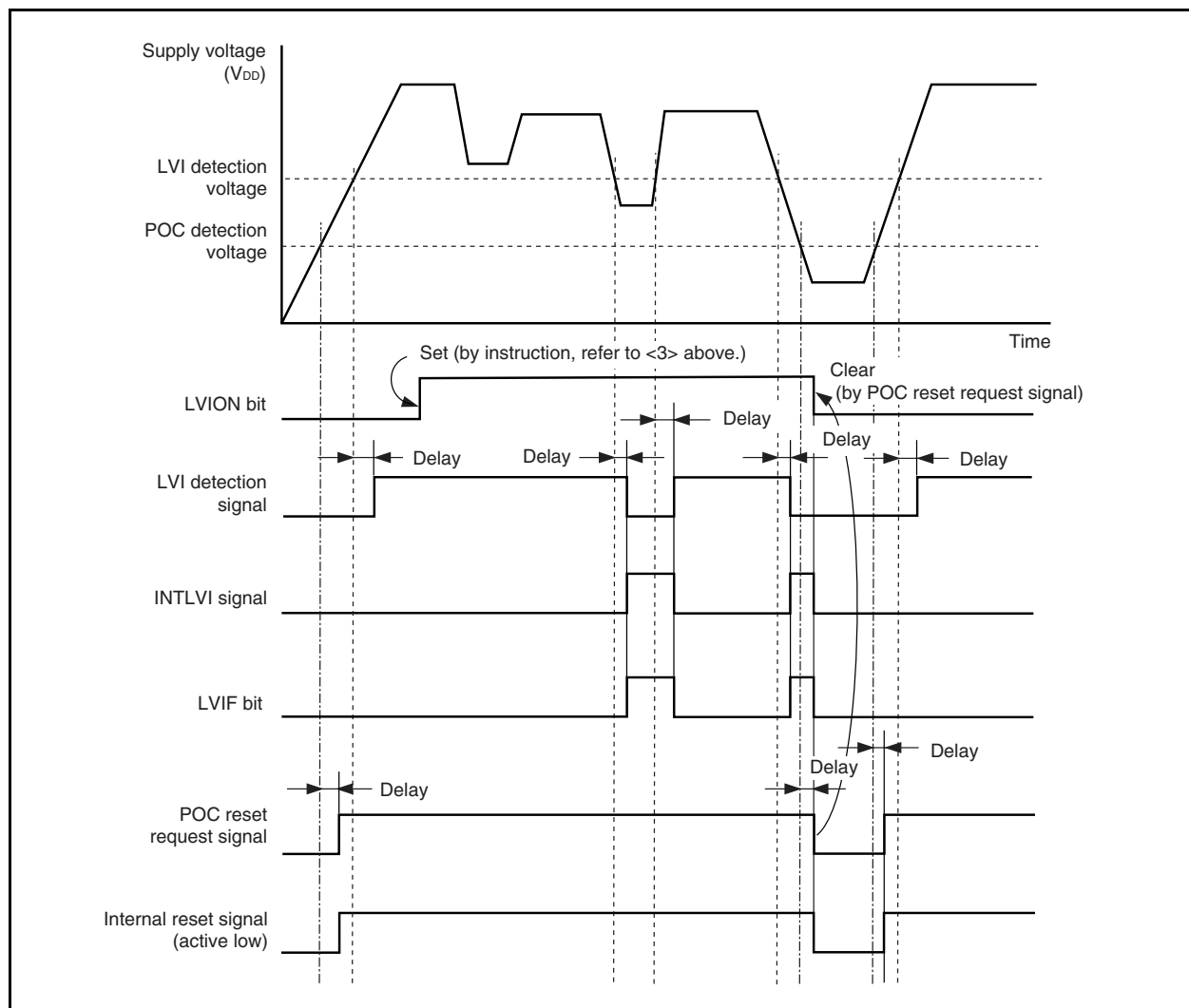
<To start operation>

- <1> Mask the interrupt of LVI.
- <2> Select the voltage to be detected by using the LVIS.LVIS0 bit.
- <3> Set the LVIM.LVION bit to 1 (to enable operation).
- <4> Insert a wait cycle of 0.2 ms MAX, by software.
- <5> By using the LVIM.LVIF bit, check if the supply voltage > detection voltage.
- <6> Clear the interrupt request flag of LVI.
- <7> Unmask the interrupt of LVI.

<To stop operation>

Clear the LVION bit to 0.

Figure 20-3. Operation Timing of Low-Voltage Detector (LVIM Bit = 0)

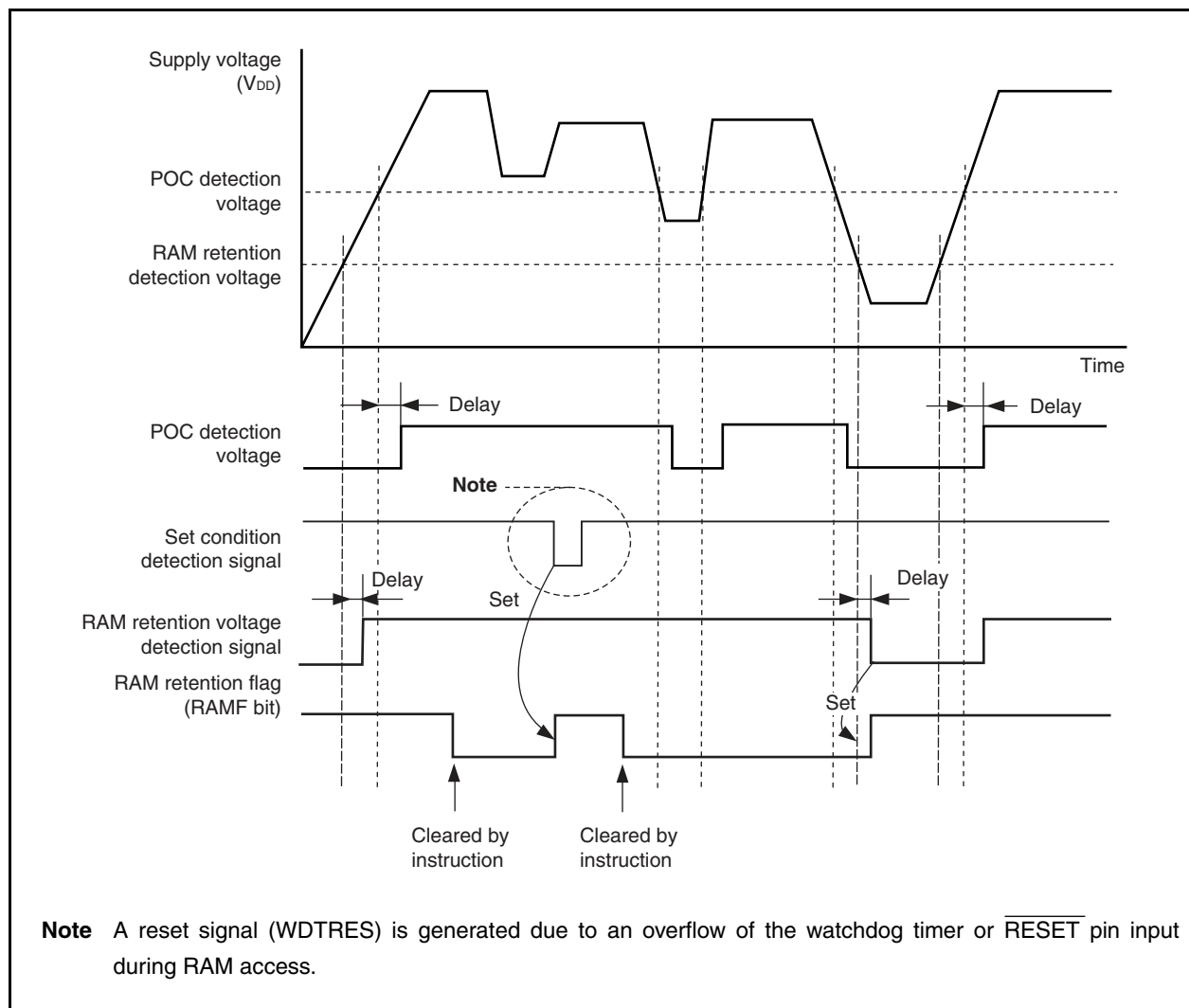


20.5 RAM Retention Voltage Detection Operation

The supply voltage and detection voltage are compared. When the supply voltage drops below the detection voltage (including on power application), the RAMS.RAMF bit is set (1).

When the POC function is not used and when the RAM retention voltage detection function is used, be sure to input an external reset signal if the detected voltage falls below the operating voltage.

Figure 20-4. Operation Timing of RAM Retention Voltage Detection Function



20.6 Emulation Function

When an in-circuit emulator is used, the operation of the RAM retention flag (RAMS.RAMF bit) can be pseudo-controlled and emulated by manipulating the PEMU1 register on the debugger.

This register is valid only in the emulation mode. It is invalid in the normal mode.

(1) Peripheral emulation register 1 (PEMU1)

| | | | | | | | | | |
|------------------|-----|--------------------|---|---|---|---|----------|---|---|
| After reset: 00H | R/W | Address: FFFFF9FEH | | | | | | | |
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PEMU1 | | 0 | 0 | 0 | 0 | 0 | EVARAMIN | 0 | 0 |

| | |
|----------|--|
| EVARAMIN | Pseudo specification of RAM retention voltage detection signal |
| 0 | Do not detect voltage lower than RAM retention voltage. |
| 1 | Detect voltage lower than RAM retention voltage (set RAMF flag). |

Caution This bit is not automatically cleared.

[Usage]

When an in-circuit emulator is used, pseudo emulation of RAMF is realized by rewriting this register on the debugger.

<1> CPU break (CPU operation stops.)

<2> Set the EVARAMIN bit to 1 by using a register write command.

By setting the EVARAMIN bit to 1, the RAMF bit is set to 1 on hardware (the internal RAM data is invalid).

<3> Clear the EVARAMIN bit to 0 by using a register write command again.

Unless this operation is performed (clearing the EVARAMIN bit to 0), the RAMF bit cannot be cleared to 0 by a CPU operation instruction.

<4> Run the CPU and resume emulation.

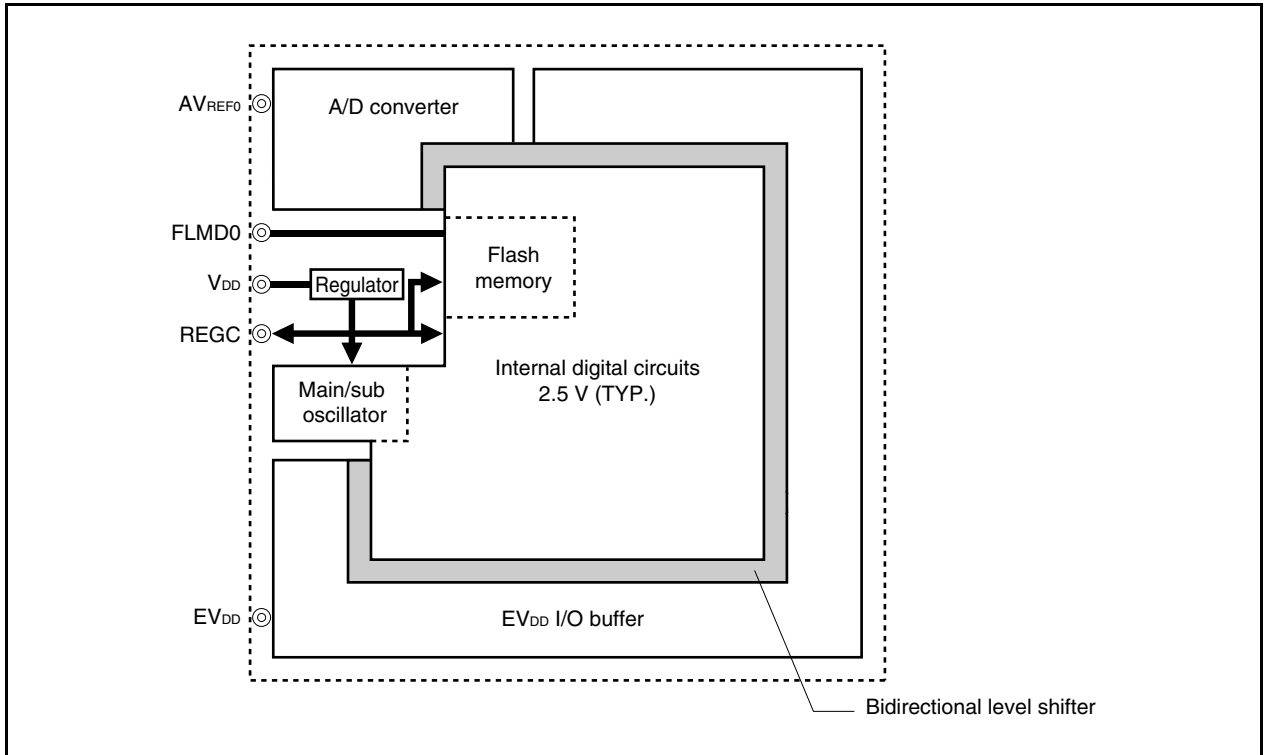
CHAPTER 21 REGULATOR

21.1 Overview

The V850ES/HF2 includes a regulator to reduce 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 buffers). The regulator output voltage is set to 2.5 V (TYP.).

Figure 21-1. Regulator



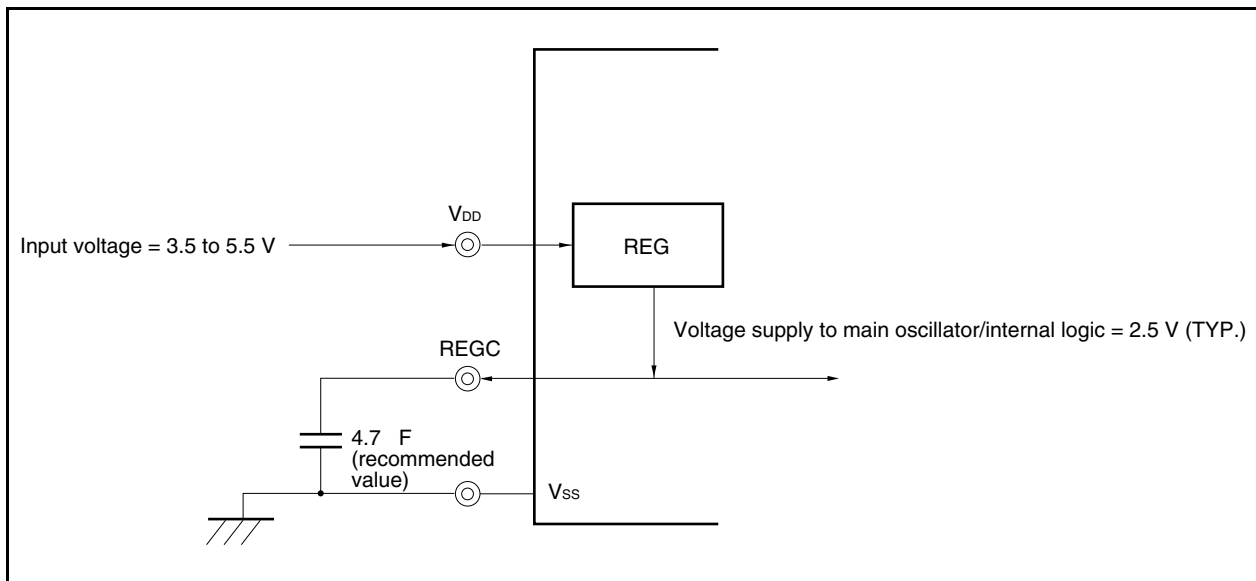
21.2 Operation

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 F (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 FLASH MEMORY

The following can be considered as the development environment and mass production applications using flash memory versions.

- For altering software after the V850ES/HF2 is soldered onto the target system.
- For data adjustment when starting mass production.
- For differentiating software according to the specification in small scale production of various models.
- For facilitating inventory management.
- For updating software after shipment.

22.1 Features

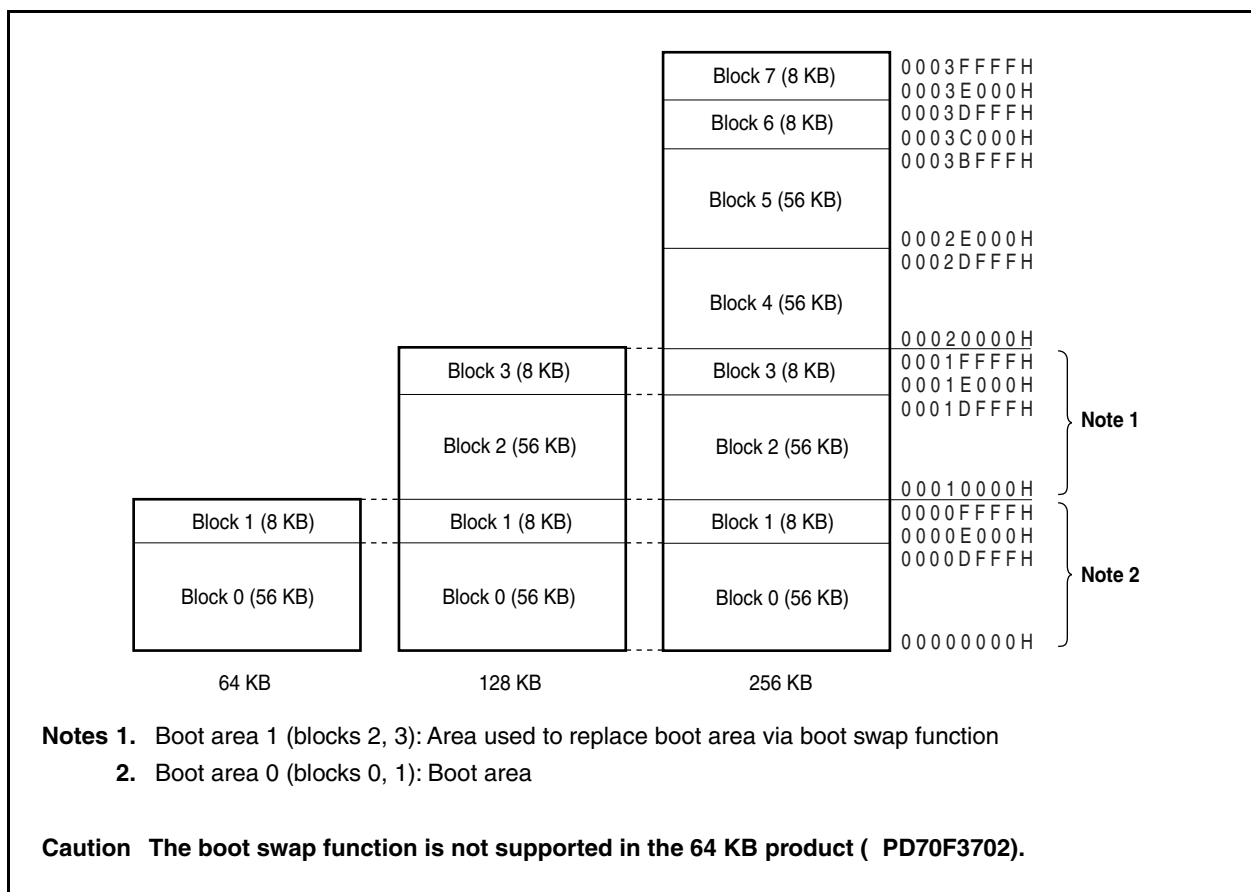
- 4-byte/1-clock access (when instruction is fetched)
- Capacity: 256 KB/128 KB/64 KB
- Write voltage: Erase/write with a single power supply
- Rewriting method
 - Rewriting by communication with dedicated flash programmer via serial interface (on-board/off-board programming)
 - Rewriting flash memory by user program (self programming)
- Flash memory write 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.

<R> 22.2 Memory Configuration

The 256K, 128K, and 64K internal flash memory areas are divided into 8, 4, and 2 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 (blocks 0, 1) located at the addresses of boot area 0 is replaced by the physical memory (blocks 2, 3) located at the addresses of boot area 1. For details of the boot swap function, refer to 22.5 **Rewriting by Self Programming**.

Figure 22-1. Flash Memory Mapping



<R>

22.3 Functional Outline

The internal flash memory of the V850ES/HF2 can be rewritten by using the rewrite function of the dedicated flash programmer, regardless of whether the V850ES/HF2 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 22-1. 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 Densai Machida Mfg. Co., Ltd.

Table 22-2. 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, program command, and read command are prohibited. | ○ | × (Supported only when setting is changed from enable to disable) |

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.

Table22-3. Security Functions

| Function | Function Outline |
|------------------------------|--|
| 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. |
| 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. |
| Program command prohibit | Program and block erase commands on all the blocks are prohibited. Setting of prohibition can be initialized by execution of the chip erase command. |
| Read command prohibit | Read command on all the blocks is prohibited. Setting of prohibition can be initialized by execution of the chip erase command. |
| Boot area rewrite prohibit | Not supported. |

Table 22-4. Security Setting

| Function | Erase, Write, Read Operations When Each Security Is Set (√: Executable, ×: Not Executable, -: Not Supported) | | Notes on Security Setting | |
|------------------------------|---|---|--|--|
| | On-Board/ Off-Board Programming | Self Programming | On-Board/ Off-Board Programming | Self Programming |
| Block erase command prohibit | Block erase command: × Chip erase command: √ Program command: √ Read command: √ | Block erasure (FlashBlockErase): √ Chip erasure: – Write (FlashWordWrite): √ Read (FlashWordRead): √ | Setting of prohibition can be initialized by chip erase command. | Supported only when setting is changed from enable to prohibit |
| Chip erase command prohibit | Block erase command: × Chip erase command: × Program command: √ ^{Note} Read command: √ | Block erasure (FlashBlockErase): √ Chip erasure: – Write (FlashWordWrite): √ Read (FlashWordRead): √ | Setting of prohibition cannot be initialized. | |
| Program command prohibit | Block erase command: × Chip erase command: √ Program command: × Read command: √ | Block erasure (FlashBlockErase): √ Chip erasure: – Write (FlashWordWrite): √ Read (FlashWordRead): √ | Setting of prohibition can be initialized by chip erase command. | |
| Read command prohibit | Block erase command: √ Chip erase command: √ Program command: √ Read command: × | Block erasure (FlashBlockErase): √ Chip erasure: – Write (FlashWordWrite): √ Read (FlashWordRead): √ | Setting of prohibition can be initialized by chip erase command. | |

Note In this case, since the erase command is invalid, data different from the data already written in the flash memory cannot be written.

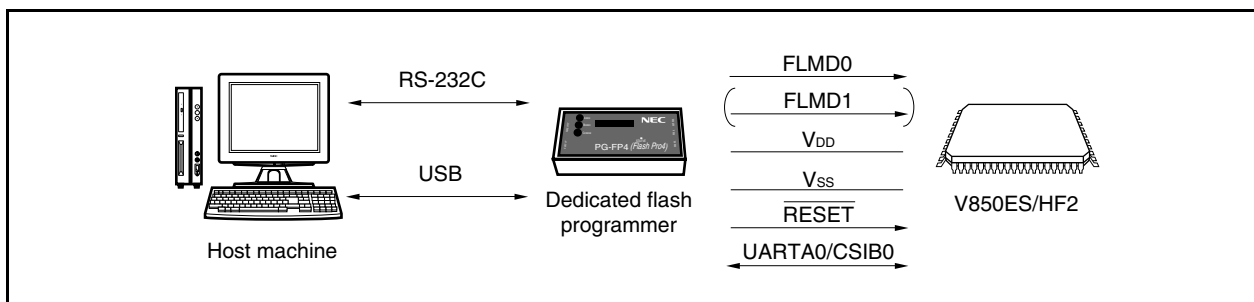
22.4 Rewriting by Dedicated Flash Programmer

The flash memory can be rewritten by using a dedicated flash programmer after the V850ES/HF2 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).

22.4.1 Programming environment

The following shows the environment required for writing programs to the flash memory of the V850ES/HF2.

Figure 22-2. 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 V850ES/HF2 to perform writing, erasing, etc. A dedicated program adapter (FA series) required for off-board writing.

- FA-70F3704GK-9EU-MX (already wired)
- FA-80GK-9EU-A (not wired: wiring required)

Remark The FA series is a product of Naito Densai Machida Mfg. Co., Ltd.

22.4.2 Communication mode

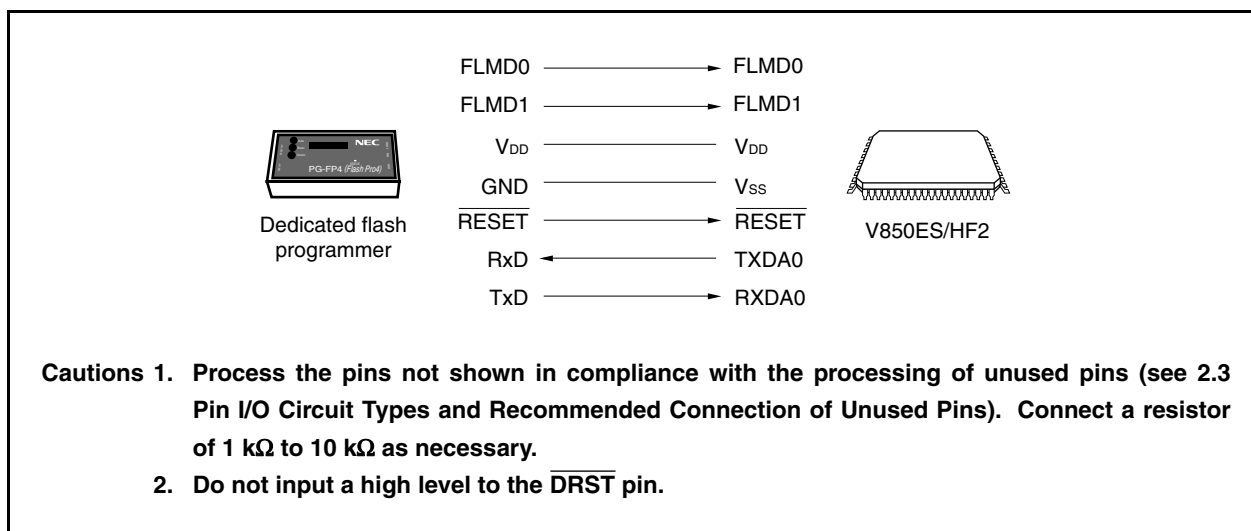
Communication between the dedicated flash programmer and the V850ES/HF2 is performed by serial communication using the UARTA0 or CSIB0 interfaces of the V850ES/HF2.

(1) UARTA0

<R>

Transfer rate: 9,600, 19,200, 31,250, 38,400, 76,800, 153,600 bps
(57,600, 115,200, and 128,000 bps are not supported.)

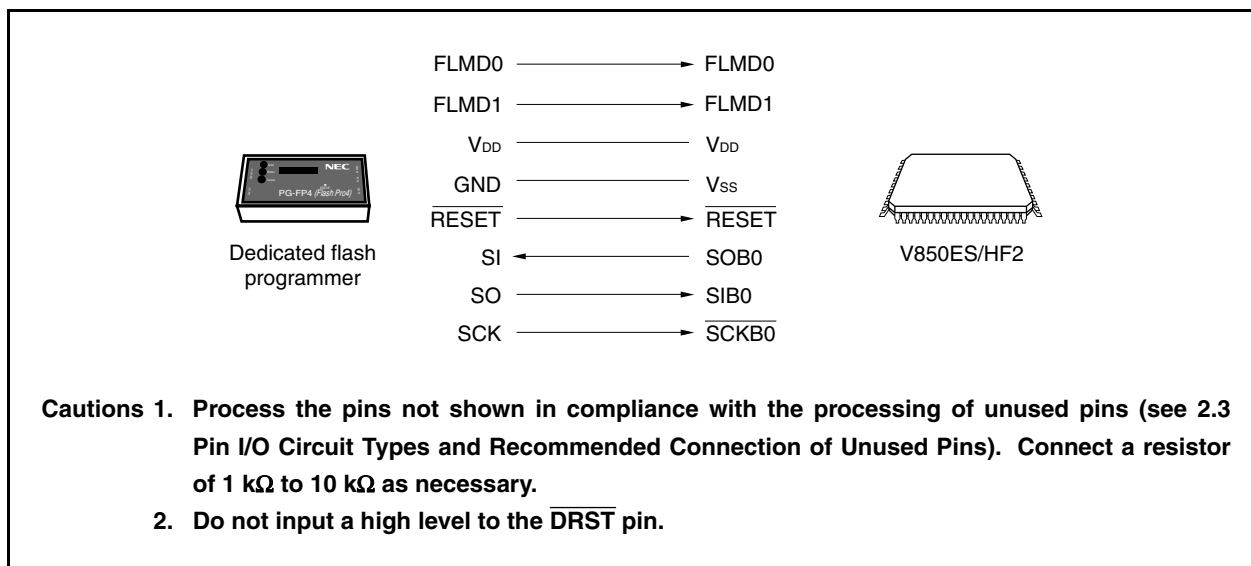
Figure 22-3. Communication with Dedicated Flash Programmer (UARTA0)



(2) CSIB0

Serial clock: 2.4 kHz to 2.5 MHz (MSB first)

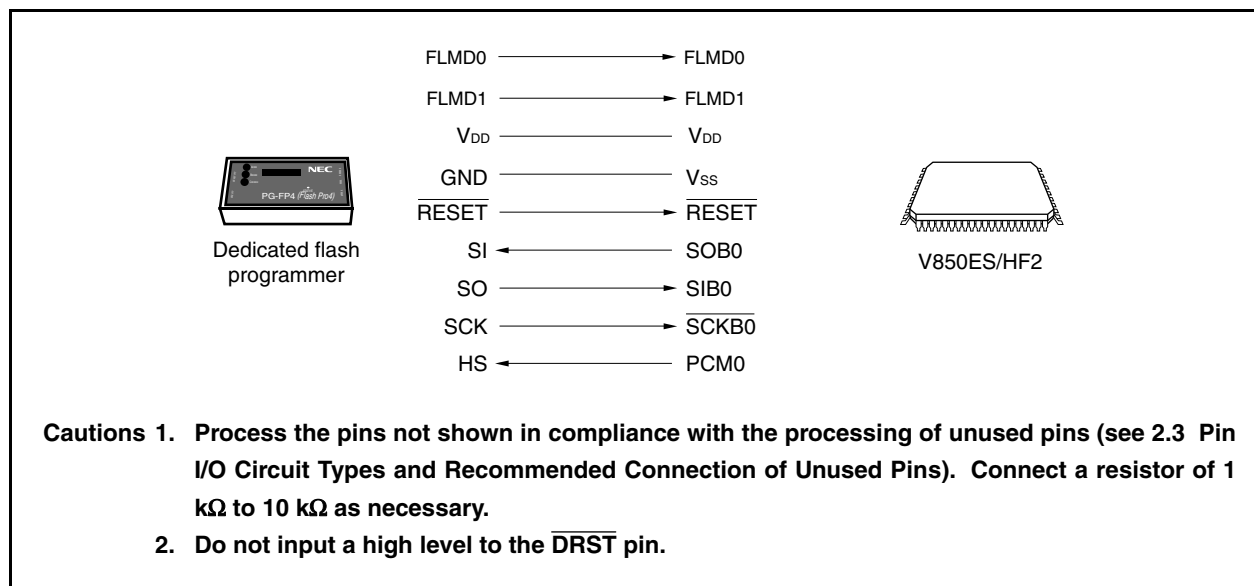
Figure 22-4. Communication with Dedicated Flash Programmer (CSIB0)



(3) CSIB0 + HS

Serial clock: 2.4 kHz to 2.5 MHz (MSB first)

Figure 22-5. Communication with Dedicated Flash Programmer (CSIB0 + HS)



The dedicated flash programmer outputs the transfer clock, and the V850ES/HF2 operates as a slave.

When the PG-FP4 is used as the dedicated flash programmer, it generates the following signals to the V850ES/HF2. For details, refer to the **PG-FP4 User's Manual (U15260E)**.

Table 22-5. Signal Connections of Dedicated Flash Programmer (PG-FP4)

| PG-FP4 | | | V850ES/HF2 | 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} | ○ | ○ | ○ |
| GND | – | Ground | V _{SS} | ○ | ○ | ○ |
| CLK | Output | Clock output to V850ES/HF2 | X1, X2 | × ^{Note 2} | × ^{Note 2} | × ^{Note 2} |
| $\overline{\text{RESET}}$ | 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 22-6, or connect them to GND via pull-down resistor on board.

2. Clock cannot be supplied via the CLK pin of the flash programmer. Create an oscillator on board and supply the clock.

Remark ○: Must be connected.

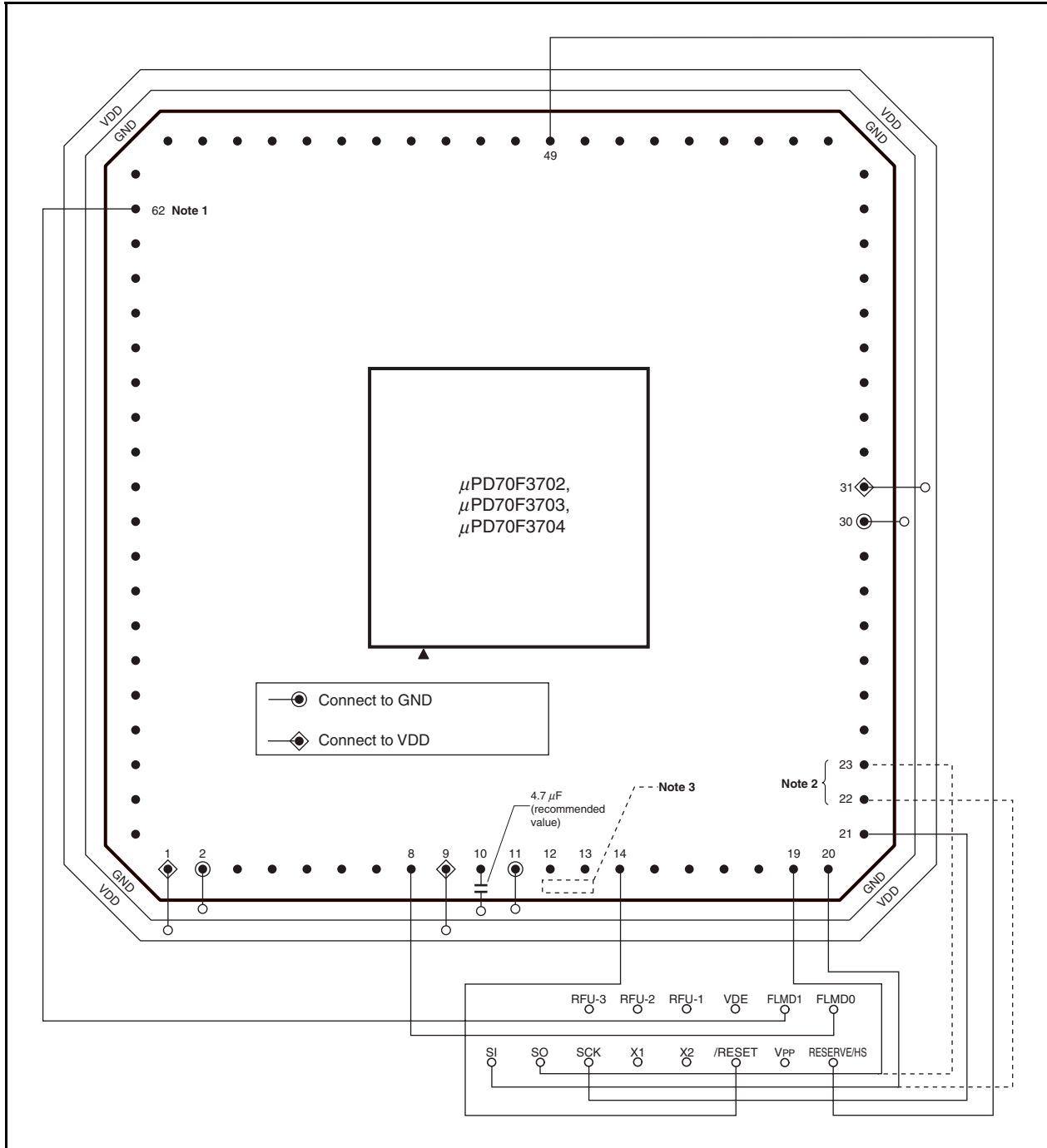
×: Does not have to be connected.

Table 22-6. Wiring of Flash Writing Adapter for V850ES/HF2 (FA-80GK-9EU-A)

| Flash Programmer (PG-FP4) Connection Pins | | | Pin Name on FA Board | When CSIB0 + HS Is Used | | When CSIB0 Is Used | | When UARTA0 Is 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 | 20 | P41/SOB0 | 20 | P30/TXD A0 | 22 |
| SO/TxD | Output | Transmit signal | SO | P40/SIB0 | 19 | P40/SIB0 | 19 | P31/RXD A0/INTP7 | 23 |
| SCK | Output | Transfer clock | SCK | P42/ $\overline{\text{SCKB0}}$ | 21 | P42/ $\overline{\text{SCKB0}}$ | 21 | Not necessary | – |
| CLK | Output | Clock to V850ES/HF2 | X1 | Not necessary | – | Not necessary | – | Not necessary | – |
| | | | X2 | Not necessary | – | Not necessary | – | Not necessary | – |
| /RESET | Output | Reset signal | /RESET | $\overline{\text{RESET}}$ | 14 | $\overline{\text{RESET}}$ | 14 | $\overline{\text{RESET}}$ | 14 |
| FLMD0 | Output | Write voltage | FLMD0 | FLMD0 | 8 | FLMD0 | 8 | FLMD0 | 8 |
| FLMD1 | Output | Write voltage | FLMD1 | PDL5/FLMD1 | 62 | PDL5/FLMD1 | 62 | PDL5/FLMD1 | 62 |
| HS | Input | Handshake signal of CSIO + HS communication | RESERVE/ HS | PCM0 | 49 | Not necessary | – | Not necessary | – |
| VDD | – | VDD voltage generation/ voltage monitor | VDD | V _{DD} | 9 | V _{DD} | 9 | V _{DD} | 9 |
| | | | | EV _{DD} | 31 | EV _{DD} | 31 | EV _{DD} | 31 |
| | | | | 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 |
| | | | | EV _{SS} | 30 | EV _{SS} | 30 | EV _{SS} | 30 |

- Cautions**
1. Be sure to connect the REGC pin to GND via a 4.7 μF (recommended value) capacitor.
 2. A clock cannot be supplied from the CLK pin of the flash programmer. Create an oscillator on the board and supply the clock from that oscillator.

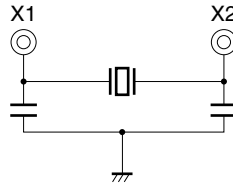
Figure 22-6. Example of Wiring of V850ES/HF2 Flash Writing Adapter (FA-80GK-9EU-A)
(in CSIB0 + HS Mode) (1/2)



**Figure 22-6. Example of Wiring of V850ES/HF2 Flash Writing Adapter (FA-80GK-9EU-A)
(in CSIB0 + HS Mode) (2/2)**

- Notes**
1. Wire the FLMD1 pin as shown below, or connect it to GND on board via a pull-down resistor.
 2. Pins used when UARTA0 is used
 3. Supply a clock by creating an oscillator on the flash writing adapter (enclosed by the broken lines). Here is an example of the oscillator.

Example



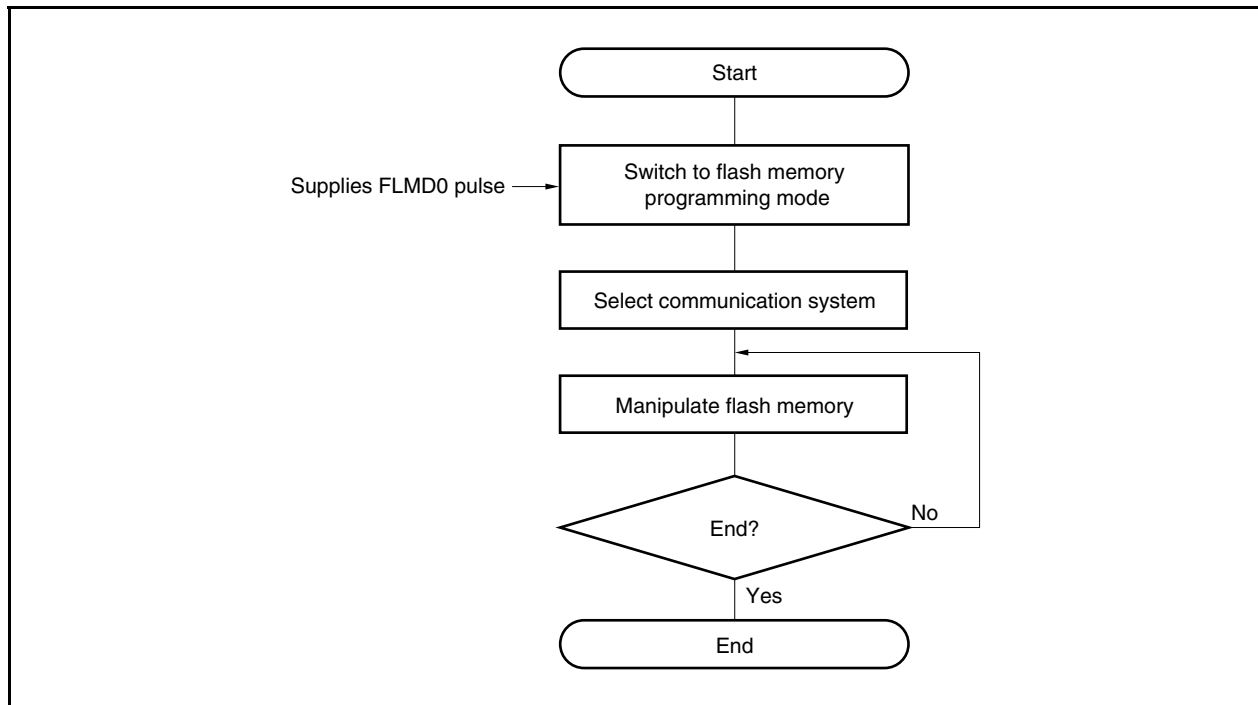
Caution Do not input a high level to the $\overline{\text{DRST}}$ pin.

- Remarks**
1. Process the pins not shown in accordance with processing of unused pins (see 2.3 Pin I/O Circuit Types and Recommended Connection of Unused Pins).
 2. This adapter is used for the 80-pin plastic TQFP package.

22.4.3 Flash memory control

The following shows the procedure for manipulating the flash memory.

Figure 22-7. Procedure for Manipulating Flash Memory

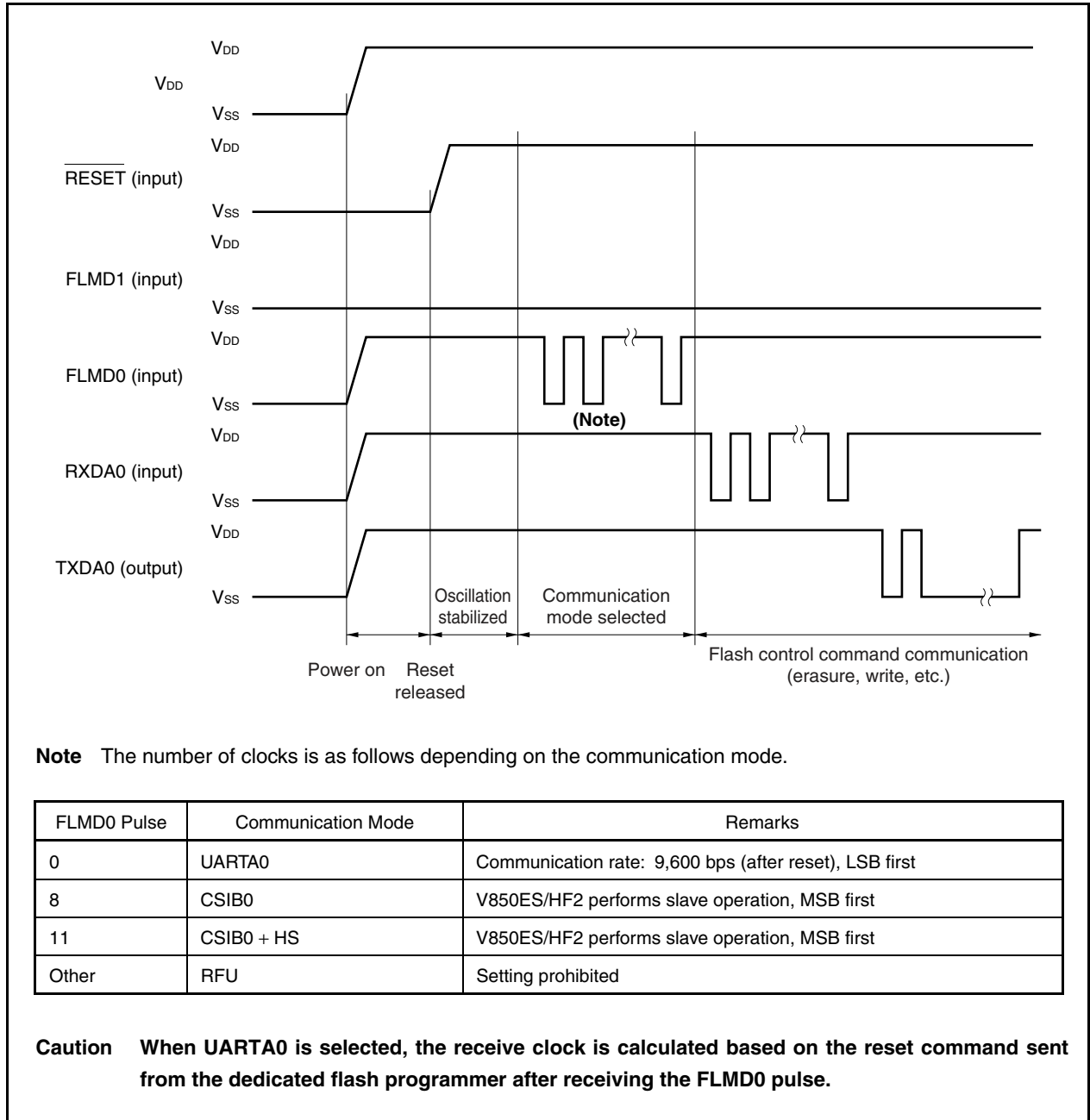


22.4.4 Selection of communication mode

In the V850ES/HF2, 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.

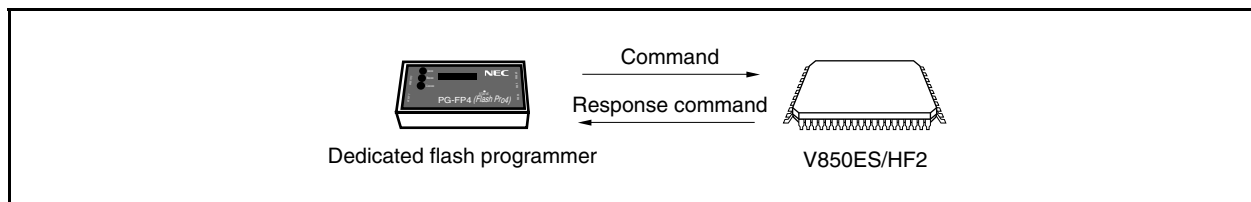
Figure 22-8. Selection of Communication Mode



22.4.5 Communication commands

The V850ES/HF2 communicates with the dedicated flash programmer by means of commands. The signals sent from the dedicated flash programmer to the V850ES/HF2 are called “commands”. The response signals sent from the V850ES/HF2 to the dedicated flash programmer are called “response commands”.

Figure 22-9. Communication Commands



The following shows the commands for flash memory control in the V850ES/HF2. All of these commands are issued from the dedicated flash programmer, and the V850ES/HF2 performs the processing corresponding to the commands.

Table 22-7. 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 | Program 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, block erase command, program command, and read command. |

<R>

22.4.6 Pin connection

When performing on-board writing, 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.

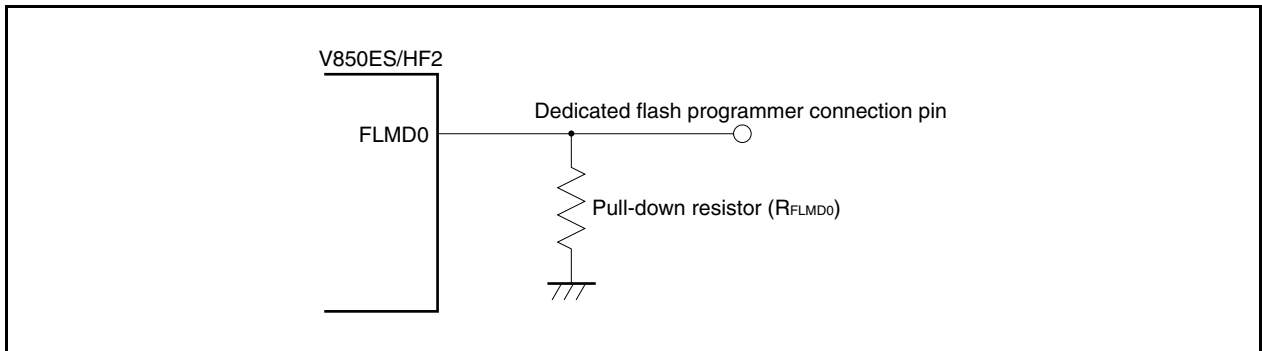
In the flash memory programming mode, all the pins not used for flash memory programming become the same status as that immediately after reset. Therefore, pin handling is required when the external device does not acknowledge the status immediately after a reset.

(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, see **22.5.5 (1) FLMD0 pin**.

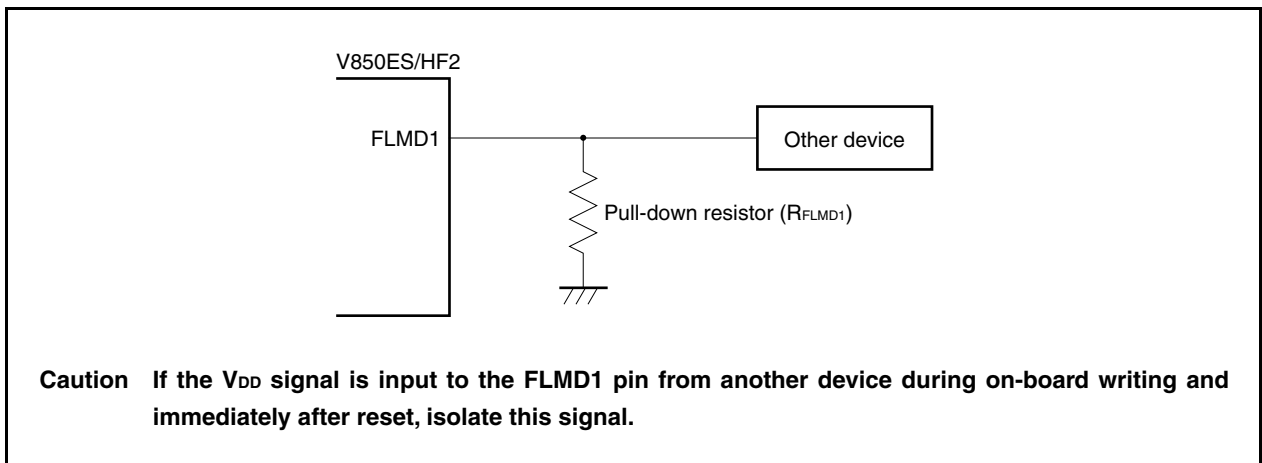
Figure 22-10. FLMD0 Pin Connection Example



(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 22-11. FLMD1 Pin Connection Example



Caution If the V_{DD} signal is input to the FLMD1 pin from another device during on-board writing and immediately after reset, isolate this signal.

Table 22-8. 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 22-9. Pins Used by Serial Interfaces

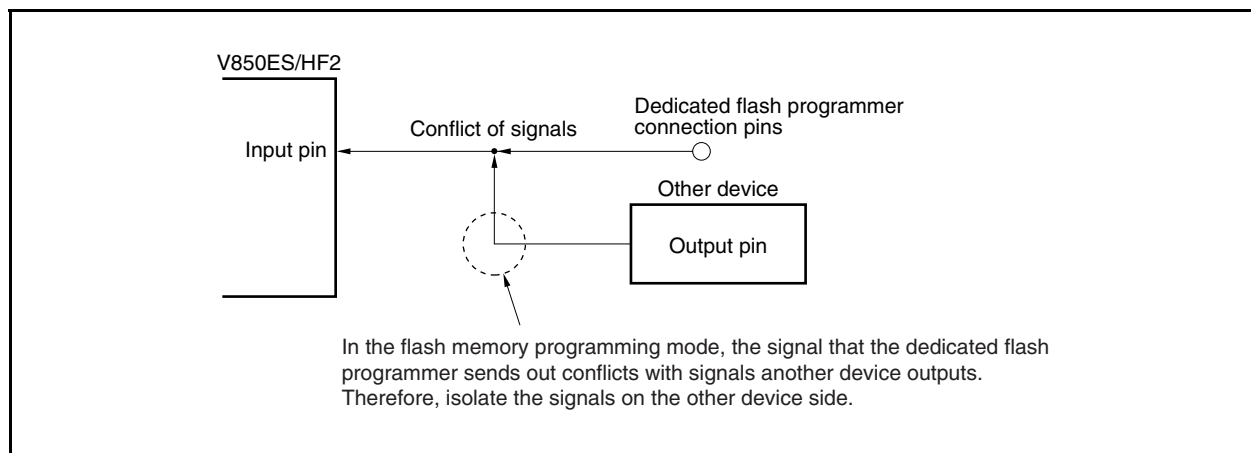
| Serial Interface | Pins Used |
|------------------|---------------------------------------|
| UARTA0 | TXDA0, RXDA0 |
| CSIB0 | SOB0, SIB0, $\overline{SCKB0}$ |
| CSIB0 + HS | SOB0, SIB0, $\overline{SCKB0}$, PCM0 |

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.

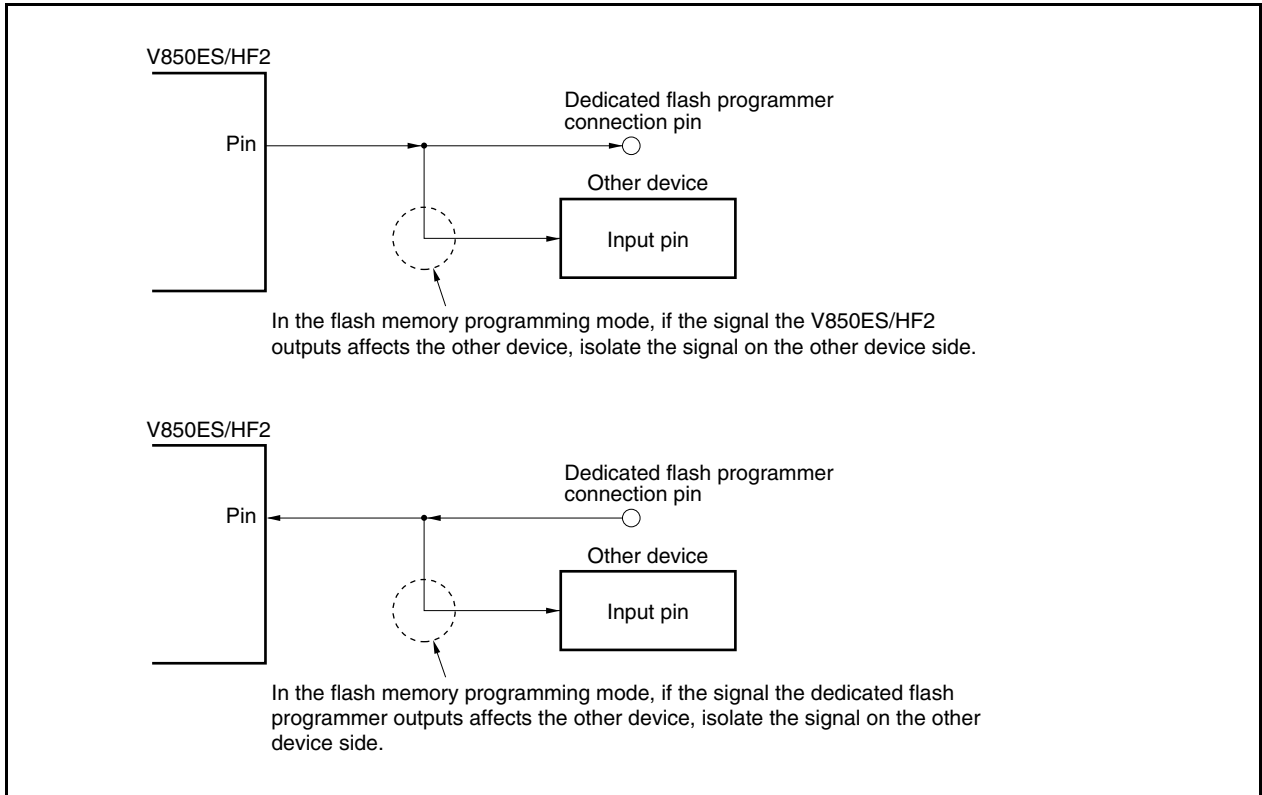
Figure 22-12. Conflict of Signals (Serial Interface Input Pin)



(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 22-13. Malfunction of Other Device

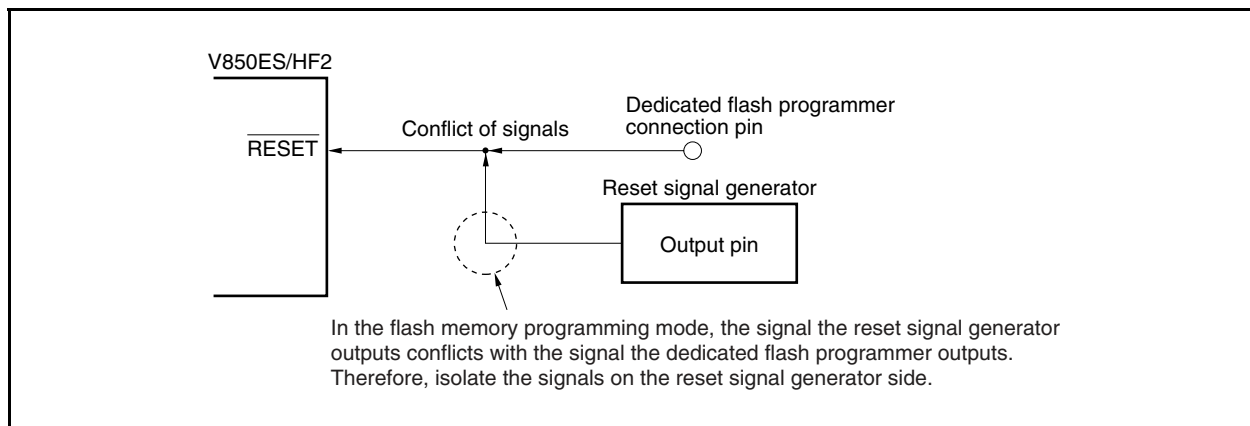


(4) RESET pin

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 22-14. 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, and XT2 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

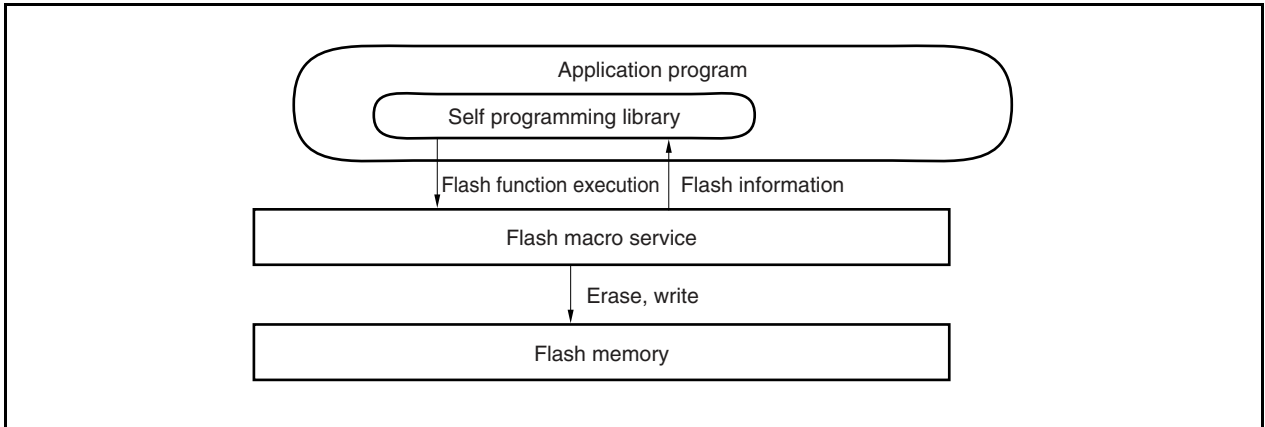
Supply the same power (V_{DD} , V_{SS} , EV_{DD} , EV_{SS} , AV_{REF0} , AV_{SS} , $REGC$) as in normal operation mode.

22.5 Rewriting by Self Programming

22.5.1 Overview

The V850ES/HF2 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 22-15. Concept of Self Programming



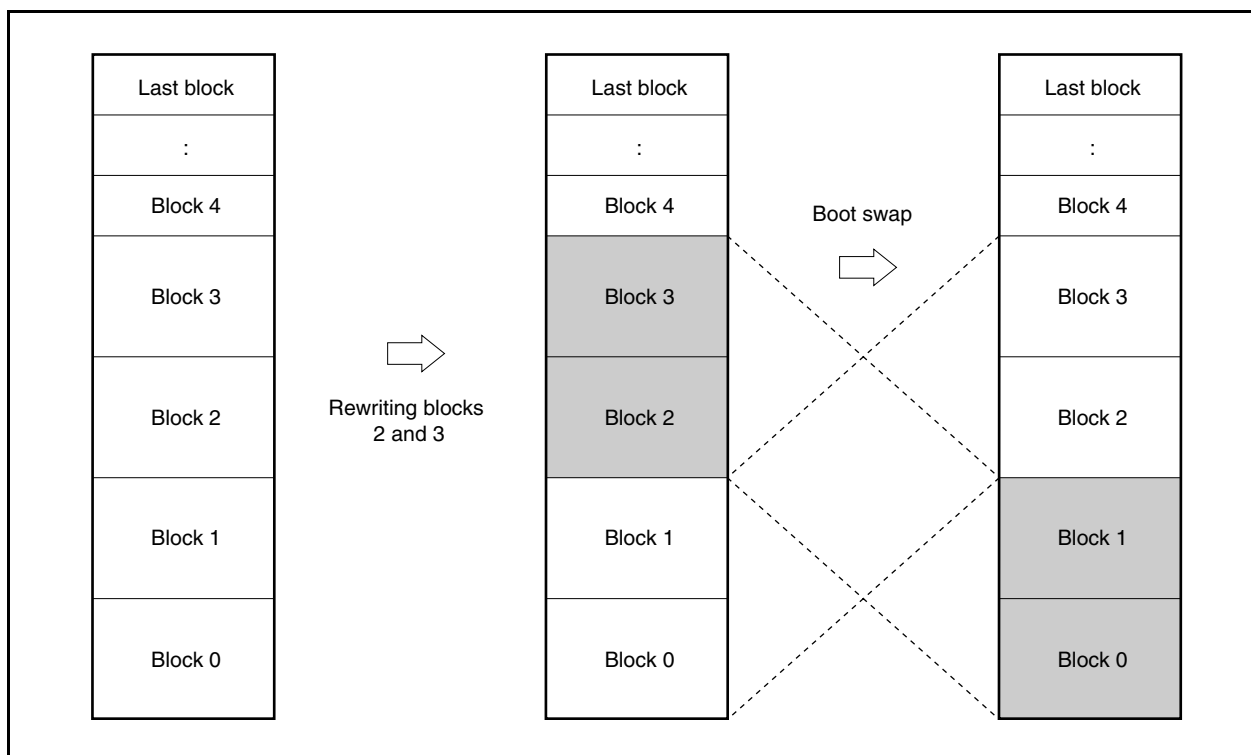
22.5.2 Features

(1) Secure self programming (boot swap function)

The μ PD70F3703 and 70F3704 support 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.

Caution The boot swap function is not supported in the μ PD70F3702.

Figure 22-16. Rewriting Entire Memory Area (Boot Swap)



(2) Interrupt support

Instructions cannot be fetched from the flash memory during self programming. Conventionally, a user handler written to the flash memory could not be used even if an interrupt occurred.

Therefore, in the V850ES/HF2, to use an interrupt during self programming, processing transits to the specific address^{Note} in the internal RAM. Allocate the jump instruction that transits processing to the user interrupt servicing at the specific address^{Note} in the internal RAM.

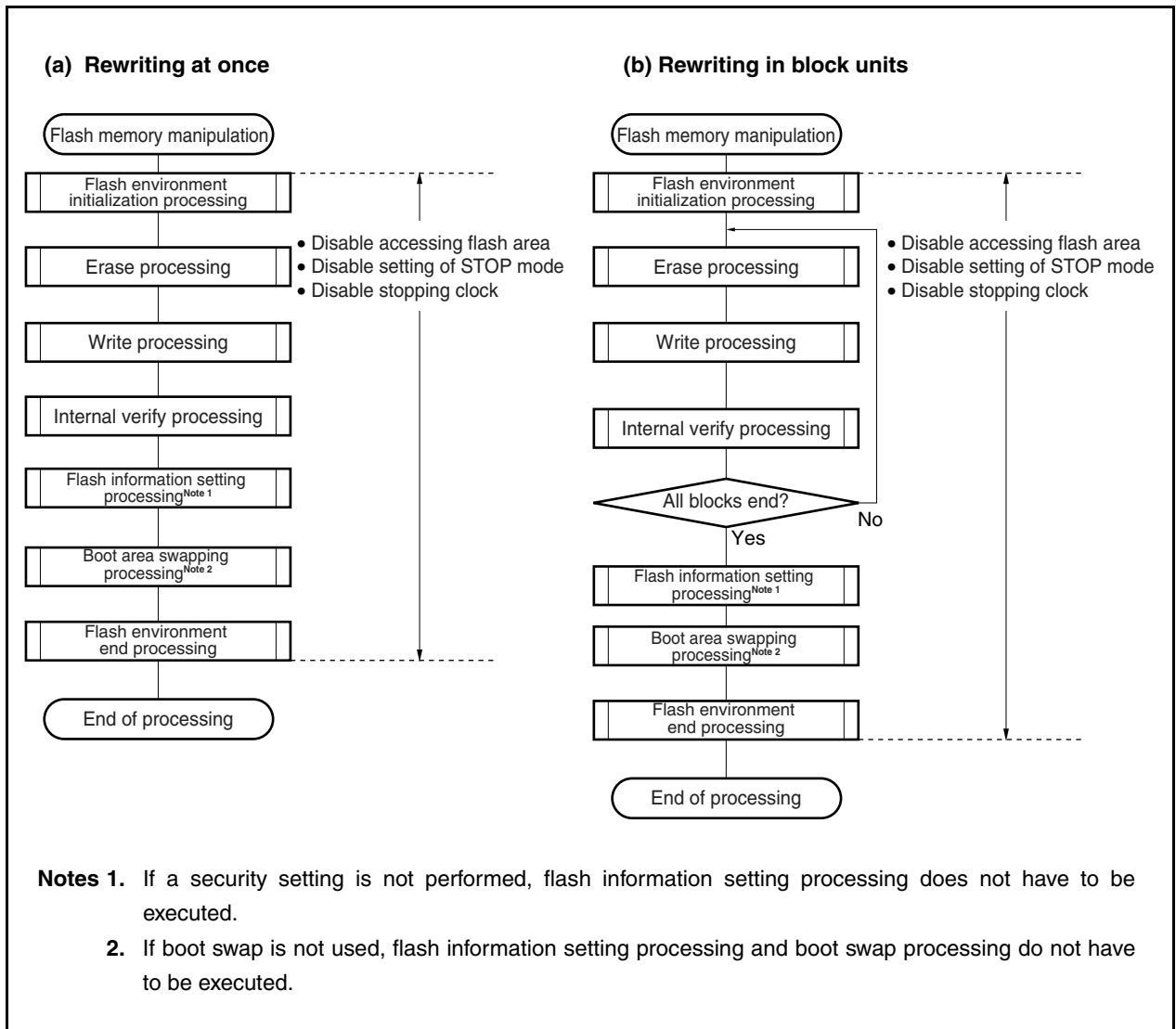
Note NMI interrupt: Start address of internal RAM
 Maskable interrupt: Start address of internal RAM + 4 addresses

22.5.3 Standard self programming flow

The entire processing to rewrite the flash memory by flash self programming is illustrated below.

<R>

Figure 22-17. Standard Self Programming Flow



22.5.4 Flash functions

Table 22-10. Flash Function List

| Function Name | Outline | Support |
|----------------------|--|---------|
| FlashEnv | Initialization of flash control macro | √ |
| FlashBlockErase | Erasure of specified one block | √ |
| FlashWordWrite | Writing from specified address | √ |
| FlashBlockVerify | 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 | √ |
| FlashWordRead | Data read from specified address | √ |
| FlashSetUserHandler | User interrupt handler registration function | √ |

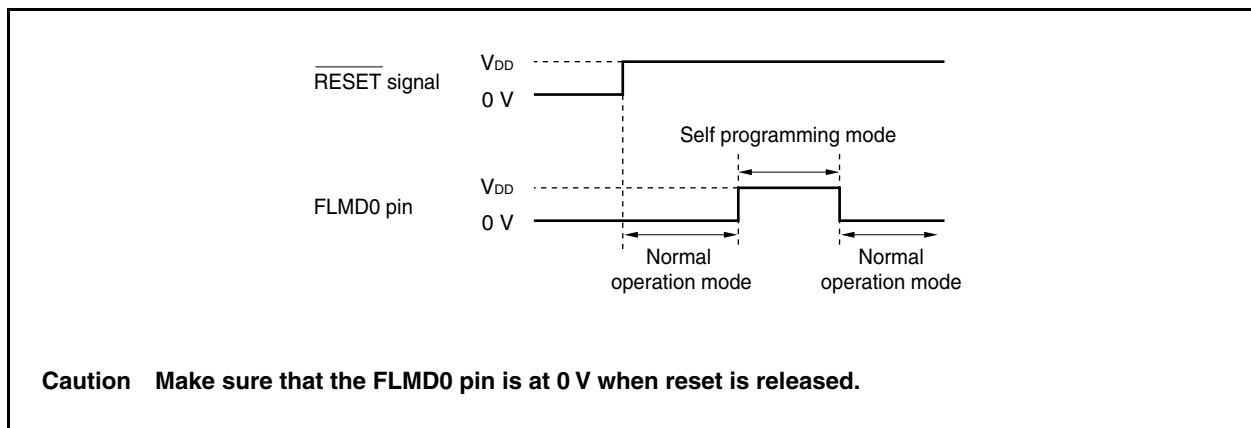
22.5.5 Pin processing

(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.

Figure 22-18. Mode Change Timing



22.5.6 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 22-11. Internal Resources Used

| Resource Name | Description |
|---|--|
| <R> Stack area (user stack + 300 bytes) | 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). |
| <R> Library code (about 2,500 bytes) | Program entity of library (can be used anywhere other than the flash memory block to be manipulated). |
| Application program | Executed as a 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, since the processing transits to the address of the internal RAM start address + 4 addresses, allocate the jump instruction that transits the processing to the user interrupt servicing at the address of the internal RAM start address + 4 addresses in advance. |
| NMI interrupt | Can be used in user application execution status or self programming status. To use this interrupt in the self-programming status, since the processing transits to the address of the internal RAM start address, allocate the jump instruction that transits the processing to the user interrupt servicing at the internal RAM start address in advance. |

CHAPTER 23 OPTION BYTE FUNCTION

The option byte is stored in address 000007AH of the internal flash memory (internal ROM area) as 8-bit data.

When writing a program to the V850ES/HF2, be sure to set the option data corresponding to the following option in the program at address 000007AH as default data.

The data in this area cannot be rewritten during program execution.

| | | | | | | | |
|-------------------|---|---------------------------------|---|---|---|------|------|
| Address: 000007AH | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OPB7 | OPB6 | – | – | – | – | OPB1 | OPB0 |
| OPB7 | OPB6 | Subclock operation mode setting | | | | | |
| 0 | 0 | Crystal resonator mode | | | | | |
| 1 | 1 | RC oscillator mode | | | | | |
| OPB1 | Watchdog timer 2 mode setting | | | | | | |
| 0 | Operating clock (f_x/f_R) selectable INTWDT2 mode/WDTRES mode selectable | | | | | | |
| 1 | Fixed to internal oscillation clock (f_R) Fixed to WDTRES mode | | | | | | |
| OPB0 | Stopping internal oscillator enable/disable | | | | | | |
| 0 | Stopping enabled | | | | | | |
| 1 | Stopping disabled | | | | | | |

<R> A sample program for using the CA850 is shown below.

[Sample Program]

```
#-----  
# OPTION_BYTES  
#-----  
.section "OPTION_BYTES"  
.byte 0b00000001 -- 0x7a  
.byte 0b00000000 -- 0x7b  
.byte 0b00000000 -- 0x7c  
.byte 0b00000000 -- 0x7d  
.byte 0b00000000 -- 0x7e  
.byte 0b00000000 -- 0x7f
```

Caution Be sure to write for 6 bytes in this section. If less than 6 bytes, an error occurs on a linker operation.

Error message: F4112: illegal "OPTION_BYTES" section size.

Remark Set 0x00 to addresses 007BH to 007FH.

CHAPTER 24 ON-CHIP DEBUG FUNCTION

The V850ES/HF2 on-chip debug function can be implemented by the following two methods.

- Using the DCU (debug control unit)
On-chip debug function is implemented by the on-chip DCU in the V850ES/HF2, with using the $\overline{\text{DRST}}$, DCK, DMS, DDI, and DDO pins as the debug interface pins.
- Not using the DCU
On-chip debug function is implemented by MINICUBE2 or the like, using the user resources, instead of the DCU.

The following table shows the features of the two on-chip debug functions.

Table 26-1. On-Chip Debug Function Features

| | | Debugging Using DCU | Debugging Without Using DCU |
|--|-------------------|---|---|
| Debug interface pins | | $\overline{\text{DRST}}$, DCK, DMS, DDI, DDO | <ul style="list-style-type: none"> • When UARTA0 is used RXD0, TXD0 • When CSIB0 is used SIB0, SOB0, SCKB0, HS (PCM0) |
| Securement of user resources | | Not required | Required |
| Hardware break function | | 2 points | 2 points |
| Software break function | Internal ROM area | 4 points | 4 points |
| | Internal RAM area | 2000 points | 2000 points |
| Real-time RAM monitor function ^{Note 1} | | Available | Available |
| Dynamic memory modification (DMM) function ^{Note 2} | | Available | Available |
| Mask function | | Reset, NMI, INTWDT2 | $\overline{\text{RESET}}$ pin |
| ROM security function | | 10-byte ID code authentication | 10-byte ID code authentication |
| Hardware used | | NINICUBE®, etc. | NINICUBE2, etc. |
| Trace function | | Not supported. | Not supported. |
| Debug interrupt interface function (DBINT) | | Not supported. | Not supported. |

Notes 1. This is a function which reads out memory contents during program execution.

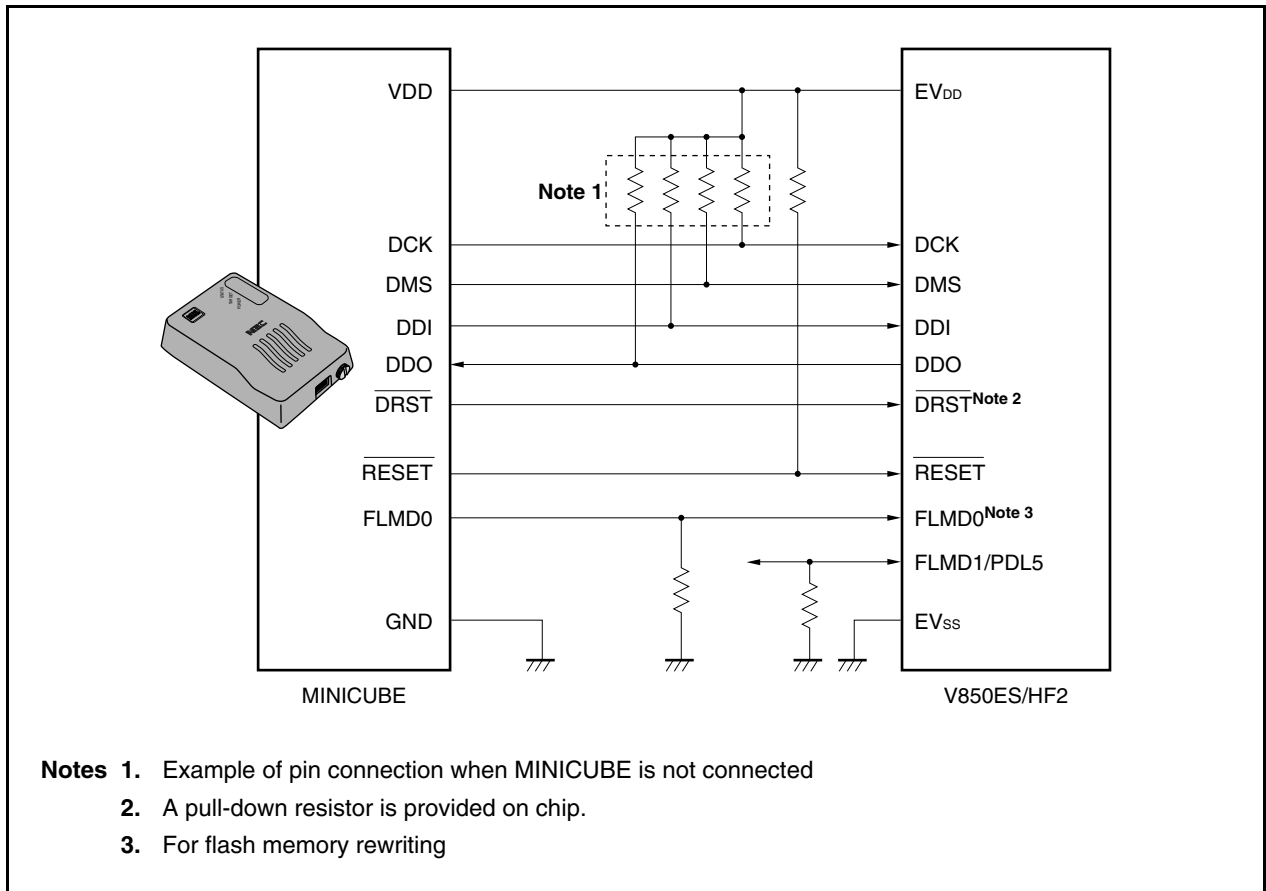
2. This is a function which rewrites RAM contents during program execution.

24.1 Debugging with DCU

Programs can be debugged using the debug interface pins ($\overline{\text{DRST}}$, DCK, DMS, DDI, and DDO) to connect the on-chip debug emulator (MINICUBE).

24.1.1 Connection circuit example

Figure 24-1. Circuit Connection Example When Debug Interface Pins Are Used for Communication Interface



24.1.2 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.

MINICUBE 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 MINICUBE. 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 MINICUBE (\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 MINICUBE

Connect the FLMD0 signal of MINICUBE to the FLMD0 pin.

In the normal mode, nothing is driven by MINICUBE (high impedance).

During a break, MINICUBE 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 after executing the download function.

For details, refer to the **ID850QB Ver. 3.10 Integrated Debugger Operation User's Manual (U17435E)**.

(8) \overline{RESET}

This is a system reset input pin. 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 MINICUBE, using the \overline{RESET} pin, to make the \overline{DRST} pin valid (initialization).

24.1.3 Maskable functions

Reset, NMI, and INTWDT2 signals can be masked.

The maskable functions with the debugger (ID850QB) and the corresponding V850ES/HF2 functions are listed below.

Table 24-2. Maskable Functions

| Maskable Functions with ID850QB | Corresponding V850ES/HF2 Functions |
|---------------------------------|--|
| NMI0 | NMI pin input |
| NMI2 | Non-maskable interrupt request signal (INTWDT2) generation |
| STOP | × |
| HOLD | × |
| RESET | Reset signal generation by $\overline{\text{RESET}}$ pin input, low-voltage detector, clock monitor, power-on clear circuit, or watchdog timer (WDT2) overflow |
| WAIT | × |

24.1.4 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 (see **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 |

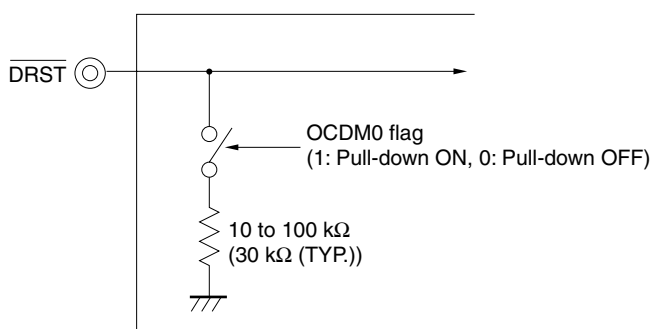
| OCDM0 | Operation mode |
|-------|---|
| 0 | Selects normal operation mode (in which a pin that functions alternately as on-chip debug function pin is used as a port/peripheral function pin) and disconnects the on-chip pull-down resistor of the P05/INTP2/ \overline{DRST} pin. |
| 1 | When \overline{DRST} 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{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{RESET} input sets this register to 01H. After reset by the WDT2RES signal, clock monitor (CLM), or low-voltage detector (LVI), power-on clear circuit (POC), however, the value of the OCDM0 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{DRST} pin.
- Set the OCDM0 bit. In this case, take the following actions.
 - <1> Clear the OCDM0 bit to 0.
 - <2> Fix the P05/INTP2/ \overline{DRST} pin to low level until <1> is completed.

2. The \overline{DRST} pin has an on-chip pull-down resistor. This resistor is disconnected when the OCDM0 flag is cleared to 0.



24.1.5 Operation

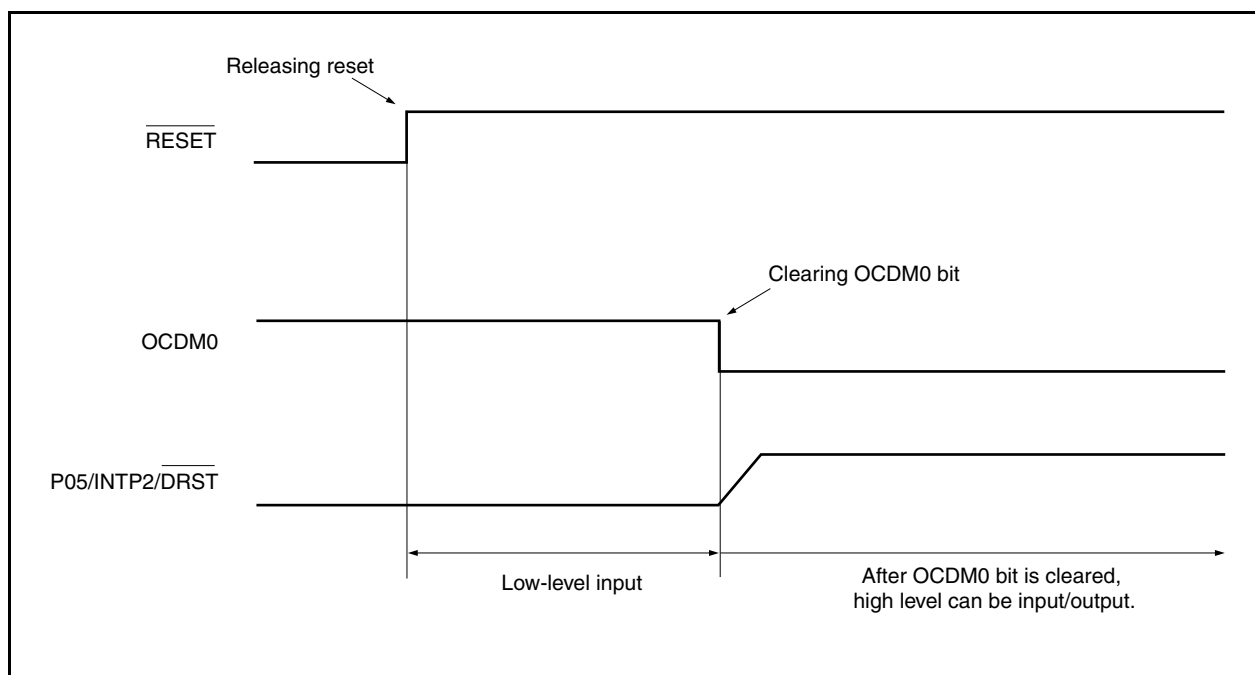
The on-chip debug function is made invalid under the conditions shown in the table below.

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 24-2. Timing When On-Chip Debug Function Is Not Used



24.1.6 Cautions

- (1) 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.
- (2) 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.
- (3) Because a software breakpoint set in the internal flash memory 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.
- (4) 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 DMM 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.
- (5) When the following conditions (a) and (b) are satisfied and operation is stopped on the emulator (IECUBE®, MINICUBE) due to a break, etc., watchdog timer 2 does not stop and a reset or non-maskable interrupt occurs. When a reset occurs, the debugger hangs up.
 - (a) The main clock or subclock is used as the source clock for watchdog timer 2.
 - (b) The internal oscillation clock is stopped (RCM.RSTOP bit = 1).

To avoid this, perform either of the following.

- When an emulator is used, use the internal oscillation clock as the source clock.
- When an emulator is used, do not stop the internal oscillator.

- (6) When the following conditions (a) and (b) are satisfied and operation is stopped on the emulator (IECUBE, MINICUBE) due to a break, etc., TMM does not stop even if the peripheral break function is set to "Break".
 - (a) Either the INTWT, internal oscillation clock ($f_R/8$), or subclock are selected as the TMM source clock.
 - (b) The main clock is stopped.

To avoid this, perform either of the following.

- When an emulator is used, the main clock (f_{xx} , $f_{xx}/2$, $f_{xx}/4$, $f_{xx}/64$, $f_{xx}/512$) is used as the source clock.
- When an emulator is used, disable the main clock oscillation.

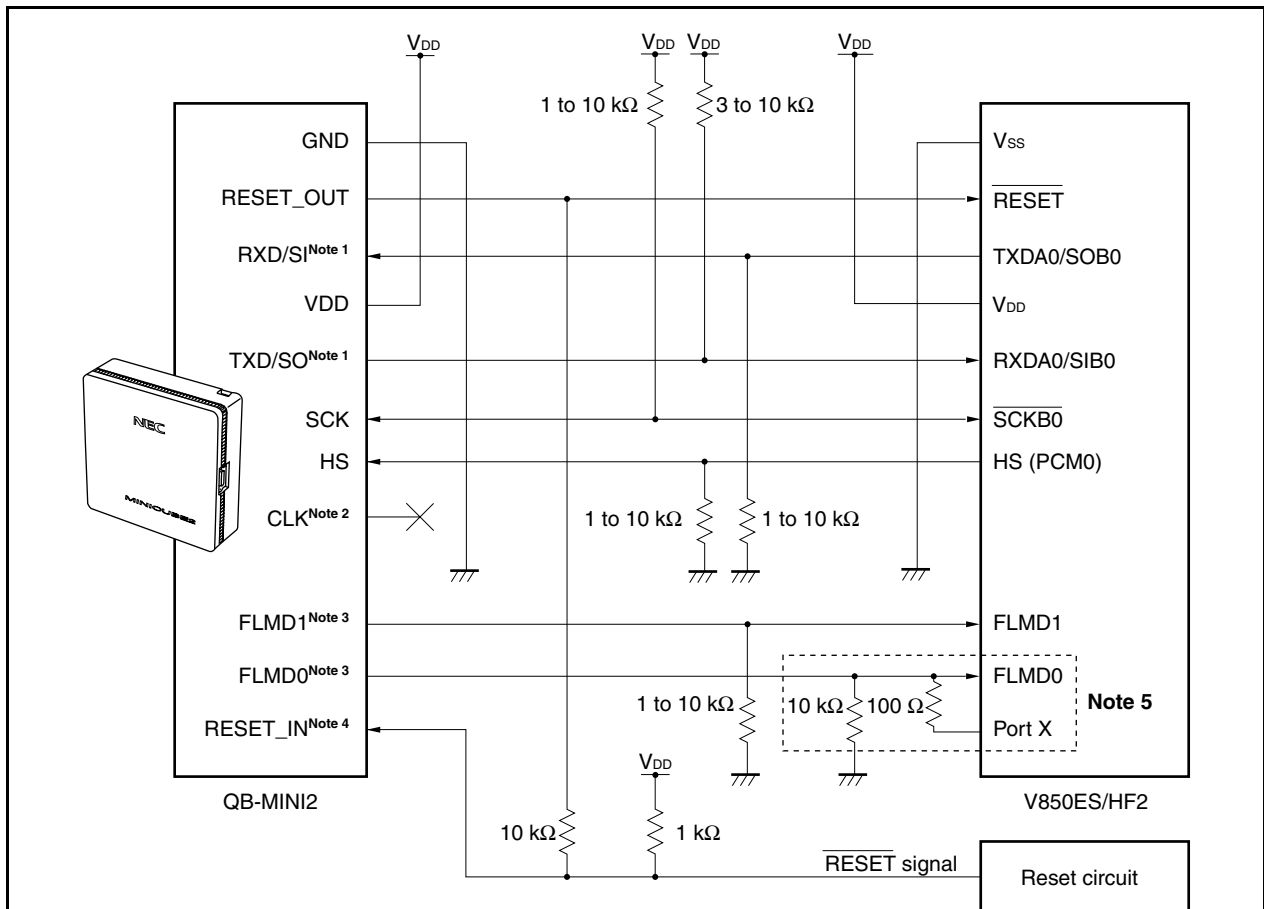
- (7) In the on-chip debug mode, the DDO pin is forcibly set to the high-level output.

24.2 Debugging Without Using DCU

The following describes how to implement an on-chip debug function using MINICUBE2 with pins for UARTA0 (RXDA0 and TXDA0), or pins for CSIB0 (SIB0, SOB0, $\overline{\text{SCKB0}}$, and HS (PMC0)) as debug interfaces, without using the DCU.

24.2.1 Circuit connection examples

Figure 24-3. Circuit Connection Example When UARTA0/CSIB0 Is Used for Communication Interface



- Notes**
1. Connect TXDA0/SOB0 (transmit side) of the V850ES/HF2 to RXD/SI (receive side) of the target connector, and TXD/SO (transmit side) of the target connector to RXDA0/SIB0 (receive side) of the V850ES/HF2.
 2. This pin may be used to supply a clock from MINICUBE2 during flash memory programming. For details, refer to **CHAPTER 22 FLASH MEMORY**.
 3. The V850ES/HF2-side pin connected to this pin (FLMD0, FLMD1) can be used as an alternate-function pin other than while the memory is rewritten during a break in debugging, because this pin is in Hi-Z state.
 4. This connection is designed assuming that the $\overline{\text{RESET}}$ signal is output from the N-ch open-drain buffer (output resistance: 100 Ω or less).
 5. The circuit enclosed by a dashed line is designed for flash self programming, which controls the FLMD0 pin via ports. Use the port for inputting or outputting the high level. When flash self programming is not performed, a pull-down resistance for the FLMD0 pin can be within 1 to 10 k Ω .

Remark Refer to **Table 24-3** for pins used when UARTA0, or CSIB0 is used for communication interface.

Table 24-3. Wiring Between V850ES/HF2 and MINICUBE2

| Pin Configuration of MINICUBE2 (QB-MINI2) | | | With CSIB0-HS | | With UARTA0 | |
|---|--------|--|----------------------------|---------|----------------------------|---------|
| Signal Name | I/O | Pin Function | Pin Name | Pin No. | Pin Name | Pin No. |
| SI/RxD | Input | Pin to receive commands and data from V850ES/HF2 | P41/SOB0 | 20 | P30/TXD0 | 22 |
| SO/TxD | Output | Pin to transmit commands and data to V850ES/HF2 | P40/SIB0 | 19 | P31/RXD0 | 23 |
| SCK | Output | Clock output pin for 3-wire serial communication | P42/SCKB0 | 21 | Not needed | – |
| CLK ^{Note} | Output | Clock output pin to V850ES/HF2 | Not needed ^{Note} | – | Not needed ^{Note} | – |
| | | | Not needed ^{Note} | – | Not needed ^{Note} | – |
| RESET_OUT | Output | Reset output pin to V850ES/HF2 | RESET | 14 | RESET | 14 |
| FLMD0 | Output | Output pin to set V850ES/HF2 to debug mode or programming mode | FLMD0 | 8 | FLMD0 | 8 |
| FLMD1 | Output | Output pin to set programming mode | PDL5/FLMD1 | 62 | PDL5/FLMD1 | 62 |
| HS | Input | Handshake signal for CSI0 + HS communication | PCM0 | 49 | Not needed | – |
| GND | – | Ground | V _{SS} | 11 | V _{SS} | 11 |
| | | | AV _{SS} | 2 | AV _{SS} | 2 |
| | | | EV _{SS} | 30 | EV _{SS} | 30 |
| RESET_IN | Input | Reset input pin on the target system | | | | |

Note It is used as the clock output of the flash programmer for MINICUBE2. For details, refer to **CHAPTER 22 FLASH MEMORY**.

24.2.2 Maskable functions

Only reset signals can be masked.

The maskable functions with the debugger (ID850QB) and the corresponding V850ES/HF2 functions are listed below.

Table 24-4. Maskable Functions

| Maskable Functions with ID850QB | Corresponding V850ES/HF2 Functions |
|---------------------------------|--|
| NMI0 | – |
| NMI1 | – |
| NMI2 | – |
| STOP | – |
| HOLD | – |
| RESET | Reset signal generation by $\overline{\text{RESET}}$ pin input |
| WAIT | – |

24.2.3 Securement of user resources

The user must prepare the following to perform communication between MINICUBE2 and the target device and implement each debug function. These items need to be set in the user program or using the compiler options.

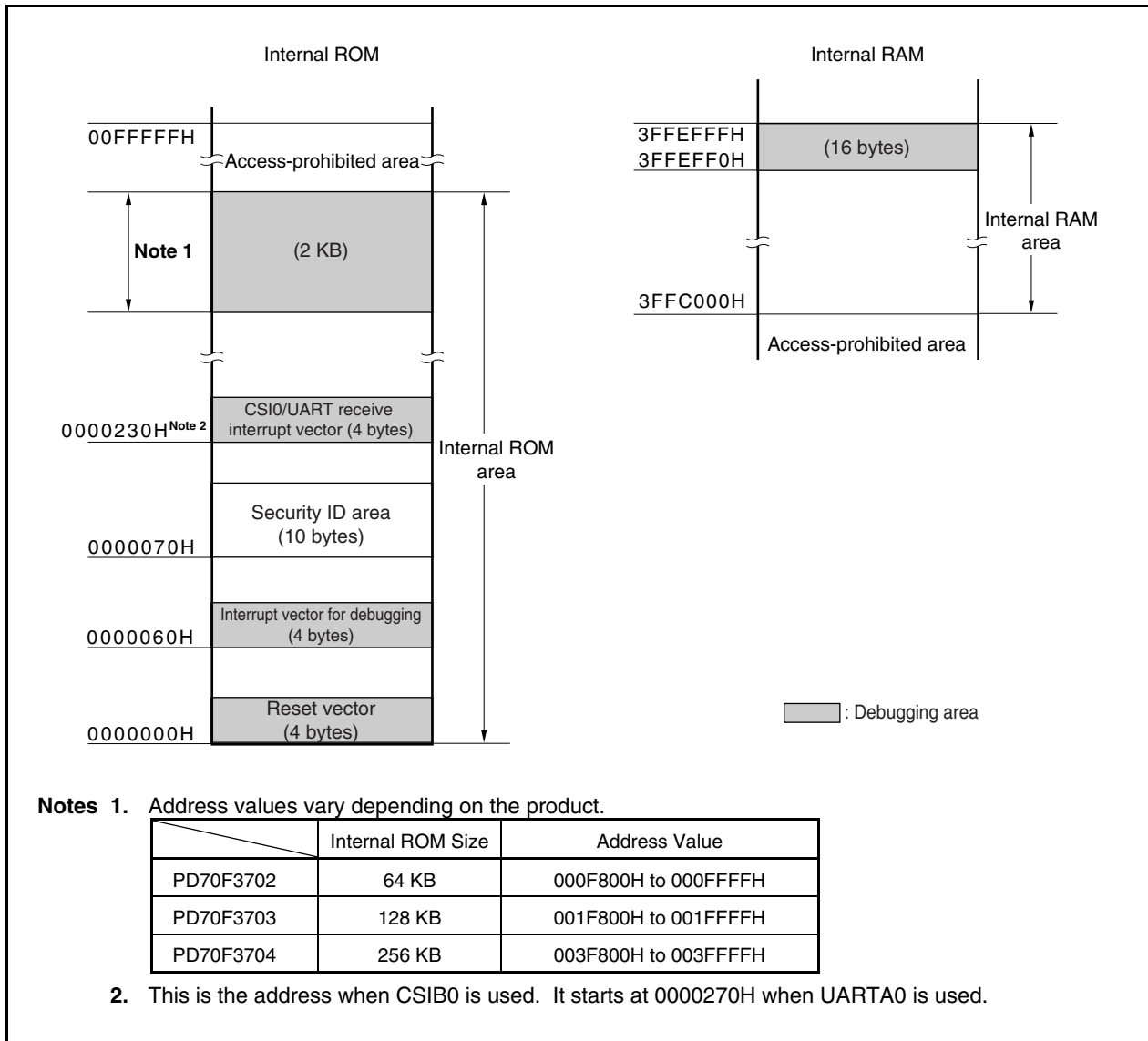
(1) Securement of memory space

The shaded portions in Figure 24-4 are the areas reserved for placing the debug monitor program, so user programs and data cannot be allocated in these spaces. These spaces must be secured so as not to be used by the user program.

(2) Security ID setting

The ID code must be embedded in the area between 0000070H and 0000079H in Figure 24-4, to prevent the memory from being read by an unauthorized person. For details, refer to **24.3 ROM Security Function**.

Figure 24-4. Memory Spaces Where Debug Monitor Programs Are Allocated



(3) Reset vector

A reset vector includes the jump instruction for the debug monitor program.

[How to secure areas]

It is not necessary to secure this area intentionally. When downloading a program, however, the debugger rewrites the reset vector in accordance with the following cases. If the rewritten pattern does not match the following cases, the debugger generates an error (F0C34 when using the ID850QB).

(a) When two nop instructions are placed in succession from address 0

| Before rewriting | | After rewriting |
|------------------|---|---------------------------------------|
| 0x0 nop | → | Jumps to debug monitor program at 0x0 |
| 0x2 nop | | 0x4 xxxx |
| 0x4 xxxx | | |

(b) When two 0xFFFF are successively placed from address 0 (already erased device)

| Before rewriting | | After rewriting |
|------------------|---|---------------------------------------|
| 0x0 0xFFFF | → | Jumps to debug monitor program at 0x0 |
| 0x2 0xFFFF | | 0x4 xxxx |
| 0x4 xxxx | | |

(c) The jr instruction is placed at address 0 (when using CA850)

| Before rewriting | | After rewriting |
|------------------|---|---------------------------------------|
| 0x0 jr disp22 | → | Jumps to debug monitor program at 0x0 |
| | | 0x4 jr disp22 - 4 |

(d) mov32 and jmp are placed in succession from address 0 (when using IAR compiler ICCV850)

| Before rewriting | | After rewriting |
|--------------------|---|---------------------------------------|
| 0x0 mov imm32,reg1 | → | Jumps to debug monitor program at 0x0 |
| 0x6 jmp [reg1] | | 0x4 mov imm32,reg1 |
| | | 0xa jmp [reg1] |

(e) The jump instruction for the debug monitor program is placed at address 0

| Before rewriting | | After rewriting |
|---------------------------------------|---|-----------------|
| Jumps to debug monitor program at 0x0 | → | No change |

(4) Securement of area for debug monitor program

The shaded portions in Figure 24-4 are the areas where the debug monitor program is allocated. The monitor program performs initialization processing for debug communication interface and RUN or break processing for the CPU. The internal ROM area must be filled with 0xFF. This area must not be rewritten by the user program.

[How to secure areas]

It is not necessarily required to secure this area if the user program does not use this area.

To avoid problems that may occur during the debugger startup, however, it is recommended to secure this area in advance, using the compiler.

The following shows examples for securing the area, using the NEC Electronics compiler CA850. Add the assemble source file and link directive code, as shown below.

- Assemble source (Add the following code as an assemble source file.)

```

-- Secures 2 KB space for monitor ROM section
.section "MonitorROM", const
.space 0x800, 0xff

-- Secures interrupt vector for debugging
.section "DBG0"
.space 4, 0xff

-- Secures interrupt vector for serial communication
-- Change the section name according to the serial communication mode used
.section "INTCB0R"
.space 4, 0xff

-- Secures 16-byte space for monitor RAM section
.section "MonitorRAM", bss
.lcomm monitorramsym, 16, 4 -- defines symbol monitorramsym

```

- Link directive (Add the following code to the link directive file.)

The following shows an example when the internal ROM has 256 KB (end address is 003FFFFH) and internal RAM has 24 KB (end address is 3FFEFFFH).

```

MROMSEG      : !LOAD ?R V0x03f800{
              MonitorROM = $PROGBITS ?A MonitorROM;
};

MRAMSEG      : !LOAD ?RW V0x03ffeff0{
              MonitorRAM = $NOBITS ?AW MonitorRAM;
};

```

(5) Securement of communication serial interface

UARTA0 or CSIB0 is used for communication between MINICUBE2 and the target system. The settings related to the serial interface modes are performed by the debug monitor program, but if the setting is changed by the user program, a communication error may occur.

To prevent such a problem from occurring, communication serial interface must be secured in the user program.

[How to secure communication serial interface]

- On-chip debug mode register (OCDM)

For the on-chip debug function using the UARTA0 or CSIB0, set the OCDM register functions to normal mode. Be sure to set as follows.

- Input low level to the P05/INTP2/ $\overline{\text{DRST}}$ pin.
- Set the OCDM0 bit as shown below.
 - <1> Clear the OCDM0 bit to 0.
 - <2> Fix the P05/INTP2/ $\overline{\text{DRST}}$ pin input to low level until the processing of <1> is complete.

- Serial interface registers

Do not set the registers related to CSIB0 or UARTA0 in the user program.

- Interrupt mask register

When CSIB0 is used, do not mask the transmit end interrupt (INTCB0R). When UARTA0 is used, do not mask the receive end interrupt (INTUA0R).

(a) When CSIB0 is used

| | | | | | | | | |
|--------|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CB0RIC | × | 0 | × | × | × | × | × | × |

(b) When UARTA0 is used

| | | | | | | | | |
|--------|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UA0RIC | × | 0 | × | × | × | × | × | × |

Remark ×: don't care

- Port registers when UAR_{TA0} is used

When UAR_{TA0} is used, port registers are set to make the TXDA₀ and RXDA₀ pins valid by the debug monitor program. Do not change the following register settings with the user program during debugging. (The same value can be overwritten.)

| | | | | | | | | |
|-------|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PMC3L | x | x | x | x | x | x | 1 | 1 |

Remark x: don't care

- Port registers when CSIB₀ is used

When CSIB₀ is used, port registers are set to make the SIB₀, SOB₀, $\overline{\text{SCKB0}}$, and HS (PMC₀) pins valid by the debug monitor program. Do not change the following register settings with the user program during debugging. (The same value can be overwritten.)

(a) SIB₀, SOB₀, and $\overline{\text{SCKB0}}$ settings

| | | | | | | | | |
|------|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PMC4 | x | x | x | x | x | 1 | 1 | 1 |

(b) HS (PMC₀ pin) settings

| | | | | | | | | |
|------|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PMCM | x | x | x | x | x | x | x | 0 |

| | | | | | | | | |
|-----|---|---|---|---|---|---|---|-------------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PCM | x | x | x | x | x | x | x | Note |

Note Writing to this bit is prohibited.
The port values corresponding to the HS pin are changed by the monitor program according to the debugger status. To perform port register settings in 8-bit units, the user program can usually use read-modify-write. If an interrupt for debugging occurs before writing, however, an unexpected operation may be performed.

Remark x: don't care

24.2.4 Cautions

(1) Handling of device that was used for debugging

Do not mount a device that was used for debugging on a mass-produced product, because the flash memory was rewritten during debugging and the number of rewrites of the flash memory cannot be guaranteed. Moreover, do not embed the debug monitor program into mass-produced products.

(2) When breaks cannot be executed

Forced breaks cannot be executed if one of the following conditions is satisfied.

- Interrupts are disabled (DI)
- Interrupts issued for the serial interface, which is used for communication between MINICUBE2 and the target device, are masked
- Standby mode is entered while standby release by a maskable interrupt is prohibited
- Mode for communication between MINICUBE2 and the target device is UARTA0, and the main clock has been stopped

(3) When pseudo real-time RAM monitor (RRM) function and DMM function do not operate

The pseudo RRM function and DMM function do not operate if one of the following conditions is satisfied.

- Interrupts are disabled (DI)
- Interrupts issued for the serial interface, which is used for communication between MINICUBE2 and the target device, are masked
- Standby mode is entered while standby release by a maskable interrupt is prohibited
- Mode for communication between MINICUBE2 and the target device is UARTA0, and the main clock has been stopped
- Mode for communication between MINICUBE2 and the target device is UARTA0, and a clock different from the one specified in the debugger is used for communication

(4) Standby release with pseudo RRM and DMM functions enabled

The standby mode is released by the pseudo RRM function and DMM function if one of the following conditions is satisfied.

- Mode for communication between MINICUBE2 and the target device is CSIB0
- Mode for communication between MINICUBE2 and the target device is UARTA0, and the main clock has been supplied.

(5) Writing to peripheral I/O registers that requires a specific sequence, using DMM function

Peripheral I/O registers that requires a specific sequence cannot be written with the DMM function.

(6) Devices for which debugger startup becomes slow

Chip erase and writing of the monitor program for debugging are conducted when the debugger is first started up, but this operation takes about a dozen seconds.

(7) Writing of the monitor program for debugging

When CPU operation clock settings are changed with the debugger, the debugger rewrites the monitor program. The time required is the same as that mentioned just above in (6). For the integrated debugger ID850QB, this applies when settings of the Clock column in the configuration dialog box are changed.

(8) Flash self programming

If a space where the debug monitor program is allocated is rewritten by flash self programming, the debugger can no longer operate normally.

24.3 ROM Security Function

24.3.1 Security ID

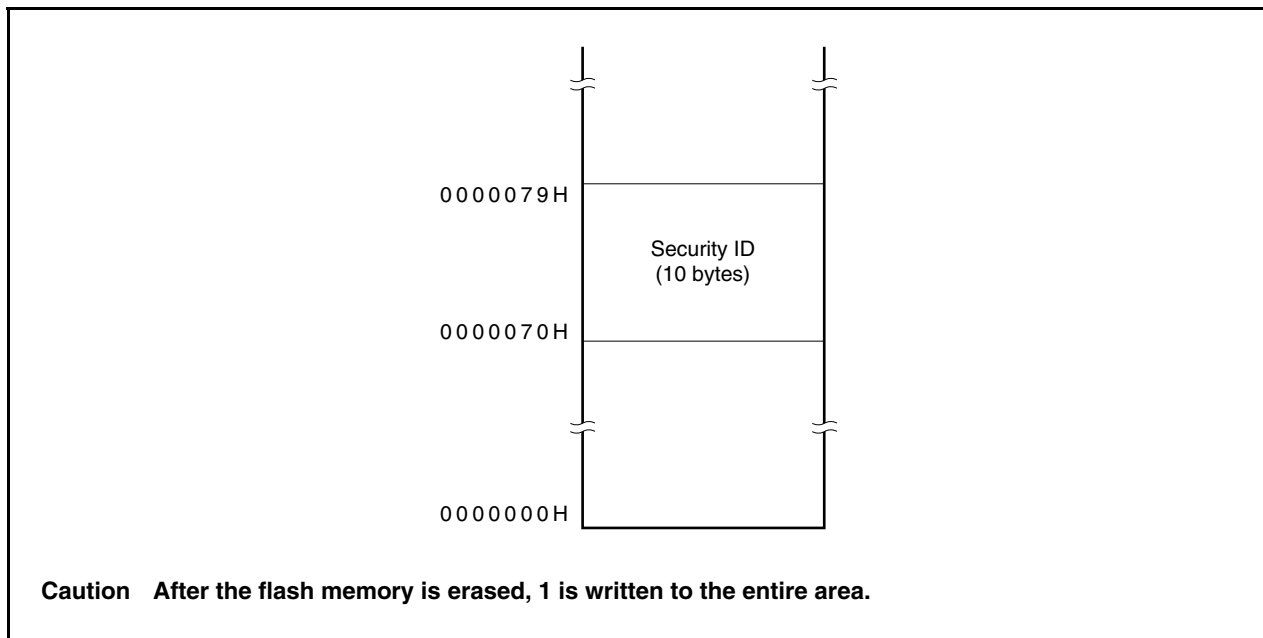
The flash memory versions of the V850ES/HF2 perform authentication using a 10-byte ID code to prevent the contents of the flash memory from being read by an unauthorized person during on-chip debugging by the on-chip debug emulator.

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 on-chip debug emulator are enabled.

- Set the 10-byte ID code to 0000070H to 0000079H.
- Bit 7 of 0000079H is the on-chip debug emulator enable flag.
(0: Disable, 1: Enable)
- When the on-chip debug 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 on-chip debug emulator enable flag is 0, even if the ID codes match.

Figure 24-5. Security ID Area



24.3.2 Setting

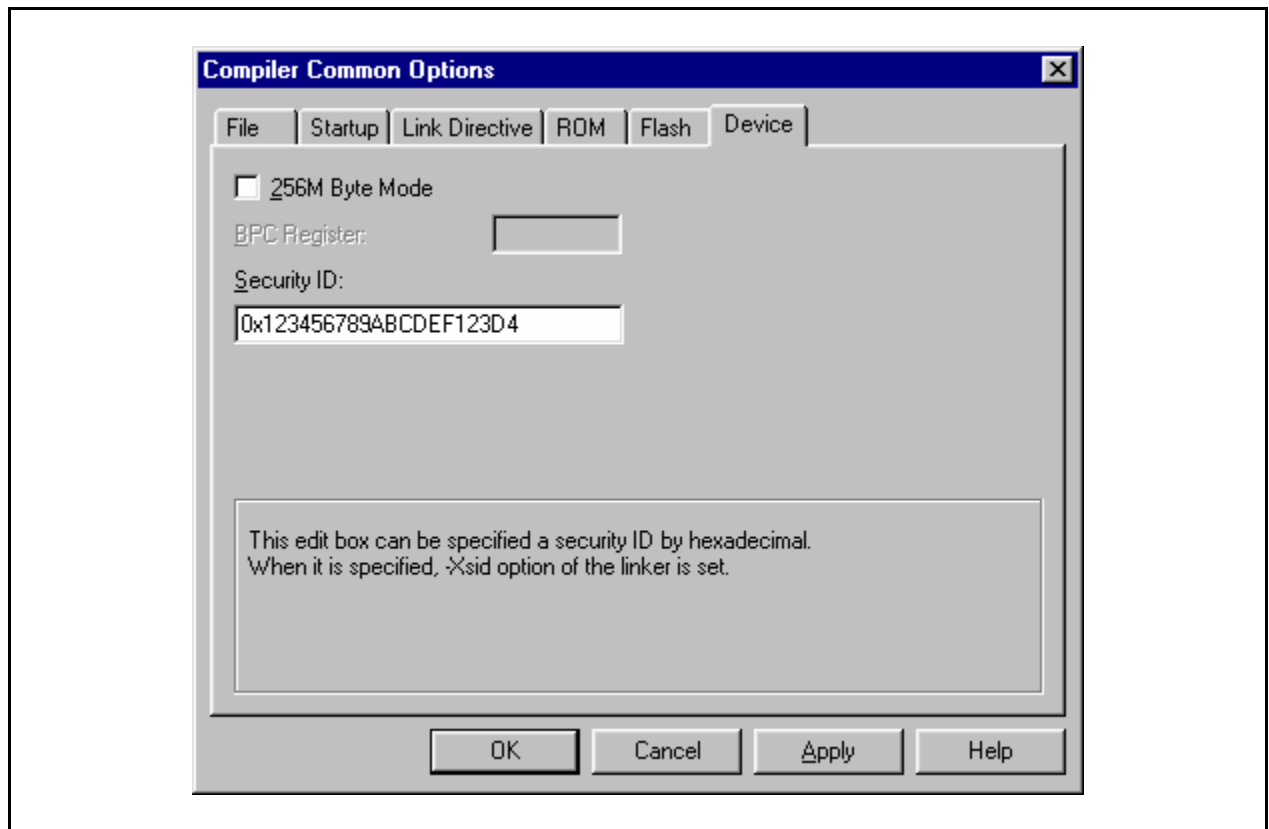
The following shows how to set the ID code as shown in Table 24-5.

When the ID code is set as shown in Table 24-5, the ID code input in the configuration dialog box of the ID850QB is "123456789ABCDEF123D4" (the ID code is case-insensitive).

Table 24-5. ID Code

| Address | Value |
|---------|-------|
| 0x70 | 0x12 |
| 0x71 | 0x34 |
| 0x72 | 0x56 |
| 0x73 | 0x78 |
| 0x74 | 0x9A |
| 0x75 | 0xBC |
| 0x76 | 0xDE |
| 0x77 | 0XF1 |
| 0x78 | 0x23 |
| 0x79 | 0xD4 |

The ID code can be specified for the device file that supports CA850 Ver. 3.10 or later and the security ID using the PM+ compiler common option setting.



[Program example (when using CA850 Ver. 3.10 or later)]

```
#-----  
# SECURITYID  
#-----  
    .section  "SECURITY_ID"  --Interrupt handler address 0x70  
    .word     0x78563412     --0-3 byte code  
    .word     0xF1DEBC9A     --4-7 byte code  
    .hword    0xD423         --8-9 byte code
```

Remark Add the above program example to the startup files.

CHAPTER 25 ELECTRICAL SPECIFICATIONS

25.1 Absolute Maximum Ratings

Absolute Maximum Ratings ($T_A = 25^\circ\text{C}$) (1/2)

| Parameter | Symbol | Conditions | Ratings | Unit |
|----------------------|-------------|--|---|------|
| Supply voltage | V_{DD} | $V_{DD} = EV_{DD}$ | -0.5 to +6.5 | V |
| | EV_{DD} | $V_{DD} = EV_{DD}$ | -0.5 to +6.5 | V |
| | AV_{REF0} | | -0.5 to +6.5 | V |
| | V_{SS} | $V_{SS} = EV_{SS} = AV_{SS}$ | -0.5 to +0.5 | V |
| | AV_{SS} | $V_{SS} = EV_{SS} = AV_{SS}$ | -0.5 to +0.5 | V |
| | EV_{SS} | $V_{SS} = EV_{SS} = AV_{SS}$ | -0.5 to +0.5 | V |
| Input voltage | V_{I1} | P00 to P06, P30 to P35, P38, P39, P40 to P42, P50 to P55, P90, P91, P96 to P99, P913 to P915, PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 to PDL11, $\overline{\text{RESET}}$, FLMD0 | -0.5 to $EV_{DD} + 0.5^{\text{Note}}$ | V |
| | V_{I3} | X1, X2, XT1, XT2 | -0.5 to $V_{RO} + 0.5^{\text{Note}}$ | V |
| Analog input voltage | V_{IAN} | P70 to P711 | -0.5 to $AV_{REF0} + 0.5^{\text{Note}}$ | V |

Note Be sure not to exceed the absolute maximum ratings (MAX. value) of each supply voltage.

- Cautions**
- 1. Avoid direct connections among the IC device output (or I/O) pins and between V_{DD} or V_{CC} and GND.**
 - 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.
 - 3. When directly connecting the external circuit to the pin that becomes high impedance state, the timing must be designed such that the output conflict is avoided on the external circuit.**

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, P30 to P35, P38, P39, P40 to P42, P50 to P55, P90, P91, P96 to P99, P913 to P915, PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 to PDL11 | Per pin | 4 | mA |
| | | | Total of all pins | 50 | mA |
| | | P70 to P711 | Per pin | 4 | mA |
| | | | Total of all pins | 20 | mA |
| Output current, high | I _{OH} | P00 to P06, P30 to P35, P38, P39, P40 to P42, P50 to P55, P90, P91, P96 to P99, P913 to P915, PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 to PDL11 | Per pin | -4 | mA |
| | | | Total of all pins | -50 | mA |
| | | P70 to P711 | Per pin | -4 | mA |
| | | | Total of all pins | -20 | mA |
| Operating ambient temperature | T _A | Normal operation mode | -40 to +85 | °C | |
| | | Flash memory programming mode | | | |
| Storage temperature | T _{stg} | | -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.
 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.
 3. When directly connecting the external circuit to the pin that becomes high impedance state, the timing must be designed such that the output conflict is avoided on the external circuit.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

25.2 Capacitance

($T_A = 25^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|-----------------|----------|--|------|------|------|------|
| I/O capacitance | C_{IO} | $f_x = 1\text{ MHz}$, Unmeasured pins returned to 0 V. | | | 10 | pF |

25.3 Operating Conditions

($T_A = -40\text{ to }+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V to }5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---------------------------------|-----------|---|----------------------|------|----------------------|------|
| Internal system clock frequency | f_{CLK} | REGC = $4.7\ \mu\text{F}$, at operation with main clock | 4 | | 20 | MHz |
| | | REGC = $4.7\ \mu\text{F}$, at operation with subclock (crystal resonator) | 32 | | 35 | kHz |
| | | REGC = $4.7\ \mu\text{F}$, at operation with subclock (RC resonator) | 12.5^{Note} | | 27.5^{Note} | kHz |

Note The internal system clock frequency is half the oscillation frequency.

25.4 Oscillator Characteristics

25.4.1 Main clock oscillator characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V to }5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

| Resonator | Recommended Circuit | Parameter | Conditions | MIN. | TYP. | MAX. | Unit |
|-------------------|---------------------|---|-------------------------|-----------------------|---------------|------|------|
| Ceramic resonator | | Oscillation frequency (f_x) ^{Note 1} | | 4 | | 5 | MHz |
| | | Oscillation stabilization time ^{Note 2} | After reset release | | $2^{16}/f_x$ | | s |
| | | | After STOP mode release | $0.5^{\text{Note 3}}$ | Note 4 | | ms |
| Crystal resonator | | Oscillation frequency (f_x) ^{Note 1} | | 4 | | 5 | MHz |
| | | Oscillation stabilization time ^{Note 2} | After reset release | | $2^{16}/f_x$ | | s |
| | | | After STOP mode release | $0.5^{\text{Note 3}}$ | Note 4 | | ms |
| | | After IDLE2 mode release | $0.35^{\text{Note 3}}$ | Note 4 | | ms | |

- Notes**
1. Indicates only oscillator characteristics.
 2. Time required to stabilize the oscillation after reset or STOP mode is released.
 3. Time required to stabilize access to the internal flash memory.
 4. The value differs depending on the OSTS register settings.

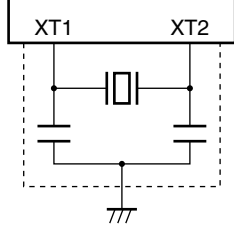
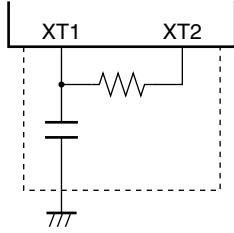
Cautions

1. When using the main clock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

- Keep the wiring length as short as possible.
 - Do not cross the wiring with the other signal lines.
 - Do not route the wiring near a signal line through which a high fluctuating current flows.
 - Always make the ground point of the oscillator capacitor the same potential as V_{SS} .
 - Do not ground the capacitor to a ground pattern through which a high current flows.
 - Do not fetch signals from the oscillator.
2. When the main clock is stopped and the subclock is operating, wait until the oscillation stabilization time has been secured by the program before switching back to the main clock.

25.4.2 Subclock oscillator characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5$ V to 5.5 V, 4.0 V $\leq AV_{REF0} \leq 5.5$ V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V)

| Resonator | Recommended Circuit | Parameter | Conditions | MIN. | TYP. | MAX. | Unit |
|-------------------|---|--|--|------|--------|------|---------------|
| Crystal resonator |  | Oscillation frequency (f_{XT}) ^{Note 1} | | 32 | 32.768 | 35 | kHz |
| | | Oscillation stabilization time ^{Note 2} | | | | 10 | s |
| RC resonator |  | Oscillation frequency ^{Notes 1, 4} | R = 390 k Ω $\pm 5\%$ ^{Note 3} C = 47 pF $\pm 10\%$ ^{Note 3} | 25 | 40 | 55 | kHz |
| | | Oscillation stabilization time ^{Note 2} | | | | 100 | μs |

- Notes**
1. Indicates only oscillator characteristics. For the CPU operation clock, see **25.8 AC Characteristics**.
 2. Time required from when V_{DD} reaches oscillation voltage range (MIN.: 3.5 V) to when the oscillation stabilizes.
 3. To avoid an adverse effect from wiring capacitance, keep the wiring length as short as possible.
 4. RC oscillation frequency is 40 kHz (TYP.). This clock is internally divided by 2. In the case of the RC resonator, the internal system clock frequency is half the oscillation frequency: MIN. = 12.5 kHz, TYP. = 20 kHz, MAX. = 27.5 kHz.

Cautions 1. When using the subclock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

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

25.4.3 PLL characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V to }5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

| 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 MIN.: 3.5 V | | | 800 | μs |

25.4.4 Internal oscillator characteristics

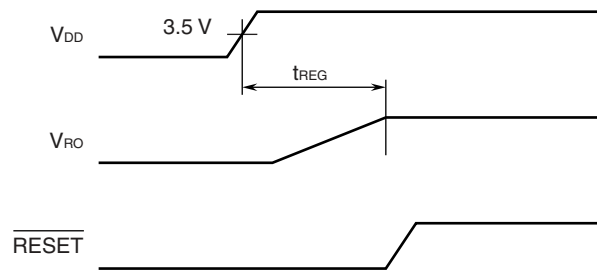
($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V to }5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|------------------|--------|------------|------|------|------|------|
| Output frequency | f_r | | 100 | 200 | 400 | kHz |

25.5 Voltage Regulator Characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|------------------|-----------|--|------|------|------|------|
| Input frequency | V_{DD} | | 3.5 | | 5.5 | V |
| Output frequency | V_{RO} | | | 2.5 | | V |
| Lock time | t_{REG} | After V_{DD} reaches MIN.: 3.5 V, $C = 4.7\ \mu\text{F} \pm 20\%$ connected to REGC pin | | | 1 | ms |



<R> **Caution** Make sure that V_{DD} rises while $\overline{\text{RESET}} = V_{SS} = 0\text{ V}$.

25.6 DC Characteristics

25.6.1 I/O level

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V to }5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

(1/2)

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---------------------|-----------|---|----------------|------|----------------|------|
| Input voltage, high | V_{IH1} | P30, P34, P38, P41, P98, PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 to PDL11 | $0.7EV_{DD}$ | | EV_{DD} | V |
| | V_{IH2} | P00 to P06, P31 to P33, P35, P39, P40, P42, P50 to P55, P90, P91, P96, P97, P99, P913 to P915 | $0.8EV_{DD}$ | | EV_{DD} | V |
| | V_{IH4} | P70 to P711 | $0.7AV_{REF0}$ | | AV_{REF0} | V |
| | V_{IH5} | \overline{RESET} , FLMD0 | $0.8EV_{DD}$ | | EV_{DD} | V |
| Input voltage, low | V_{IL1} | P30, P34, P38, P41, P98, PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 to PDL11 | EV_{SS} | | $0.3EV_{DD}$ | V |
| | V_{IL2} | P00 to P06, P31 to P33, P35, P39, P40, P42, P50 to P55, P90, P91, P96, P97, P99, P913 to P915 | EV_{SS} | | $0.2EV_{DD}$ | V |
| | V_{IL4} | P70 to P711 | AV_{SS} | | $0.3AV_{REF0}$ | V |
| | V_{IL5} | \overline{RESET} , FLMD0 | EV_{SS} | | $0.2EV_{DD}$ | V |

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5$ V to 5.5 V, 4.0 V $\leq AV_{REF0} \leq 5.5$ V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V)

(2/2)

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit | |
|--|------------------|---|---------------------------|--------------------------|------|--------------------|---|
| Output voltage, high ^{Note 1} | V _{OH1} | P00 to P06, P30 to P35, P38, P39, P40 to P42, P50 to P55, P90, P91, P96 to P99, P913 to P915, PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 to PDL11 | I _{OH} = -1.0 mA | EV _{DD} - 1.0 | | EV _{DD} | V |
| | | | I _{OH} = -0.1 mA | EV _{DD} - 0.5 | | EV _{DD} | V |
| | V _{OH3} | P70 to P711 | I _{OH} = -1.0 mA | AV _{REF0} - 1.0 | | AV _{REF0} | V |
| | | | I _{OH} = -0.1 mA | AV _{REF0} - 0.5 | | AV _{REF0} | V |
| Output voltage, low ^{Note 1} | V _{OL1} | P00 to P06, P30 to P35, P38, P39, P40 to P42, P50 to P55, P90, P91, P96 to P99, P913 to P915, PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 to PDL11 | I _{OL} = 1.0 mA | 0 | | 0.4 | V |
| | V _{OL3} | P70 to P711 | I _{OL} = 1.0 mA | 0 | | 0.4 | V |
| Pull-up resistor | R ₁ | V _I = 0 V | 10 | 30 | 100 | k Ω | |
| Pull-down resistor ^{Note 2} | R ₂ | V _I = V _{DD} | 10 | 30 | 100 | k Ω | |

- Notes**
- The maximum value of the total of I_{OH}/I_{OL} is 20 mA/-20 mA for each power supply (EV_{DD}, AV_{REF0}).
 - DRST pin only

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

25.6.2 Pin leakage current

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5$ V to 5.5 V, 4.0 V $\leq AV_{REF0} \leq 5.5$ V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V)

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|-------------------|-----------------------------------|-----------------------|------|------|---------------|
| Input leakage current, high ^{Note} | I _{LIH1} | V _{IN} = V _{DD} | Analog pin | | +0.2 | μA |
| | | | Other than analog pin | | +0.5 | |
| Input leakage current, low ^{Note} | I _{LIL1} | V _{IN} = 0 V | Analog pin | | -0.2 | μA |
| | | | Other than analog pin | | -0.5 | |
| Output leakage current, high | I _{LOH1} | V _O = V _{DD} | Analog pin | | +0.2 | μA |
| | | | Other than analog pin | | +0.5 | |
| Output leakage current, low | I _{LOL1} | V _O = 0 V | Analog pin | | -0.2 | μA |
| | | | Other than analog pin | | -0.5 | |

Note The value of the FLMD0 pin is as follows.

- Input leakage current, high: 2 μA (MAX.)
- Input leakage current, low: -2 μA (MAX.)

25.6.3 Supply current

 (T_A = -40 to +85°C, V_{DD} = EV_{DD} = 3.5 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit | |
|----------------------------------|------------------|---|--|-----------------------------------|------|------|------|----|
| Supply current ^{Note 1} | I _{DD1} | Normal operation mode | f _{XX} = 20 MHz (f _X = 5 MHz) | All peripheral function operating | | 25 | 40 | mA |
| | | | | All peripheral function stopped | | 20 | | mA |
| | I _{DD2} | HALT mode | f _{XX} = 20 MHz (f _X = 5 MHz) | All peripheral function operating | | 14 | 24 | mA |
| | | | | All peripheral function stopped | | 9 | | mA |
| | I _{DD3} | IDLE1 mode | f _{XX} = 5 MHz (f _X = 5 MHz), PLL off | | | 0.6 | 0.9 | mA |
| | I _{DD4} | IDLE2 mode | f _{XX} = 5 MHz (f _X = 5 MHz), PLL off | | | 0.25 | 0.7 | mA |
| | I _{DD5} | Subclock operation mode ^{Notes 2, 3} | Crystal resonator (f _{XT} = 32.768 kHz) | | | 200 | 400 | μA |
| | | | RC resonator (f _{XT} = 40 kHz ^{Note 4}) | | | 200 | 400 | μA |
| | I _{DD6} | Sub-IDLE mode ^{Notes 2, 3} | Crystal resonator (f _{XT} = 32.768 kHz) | | | 20 | 120 | μA |
| | | | RC resonator (f _{XT} = 40 kHz ^{Note 4}) | | | 35 | 140 | μA |
| | I _{DD7} | Stop mode ^{Notes 2, 5} | POC stopped, internal oscillator stopped | | | 7 | 50 | μA |
| | | | POC operating, internal oscillator stopped | | | 10 | 55 | μA |
| | | | POC stopped, internal oscillator operating | | | 15 | 65 | μA |
| | | | POC operating, internal oscillator operating | | | 18 | 70 | μA |

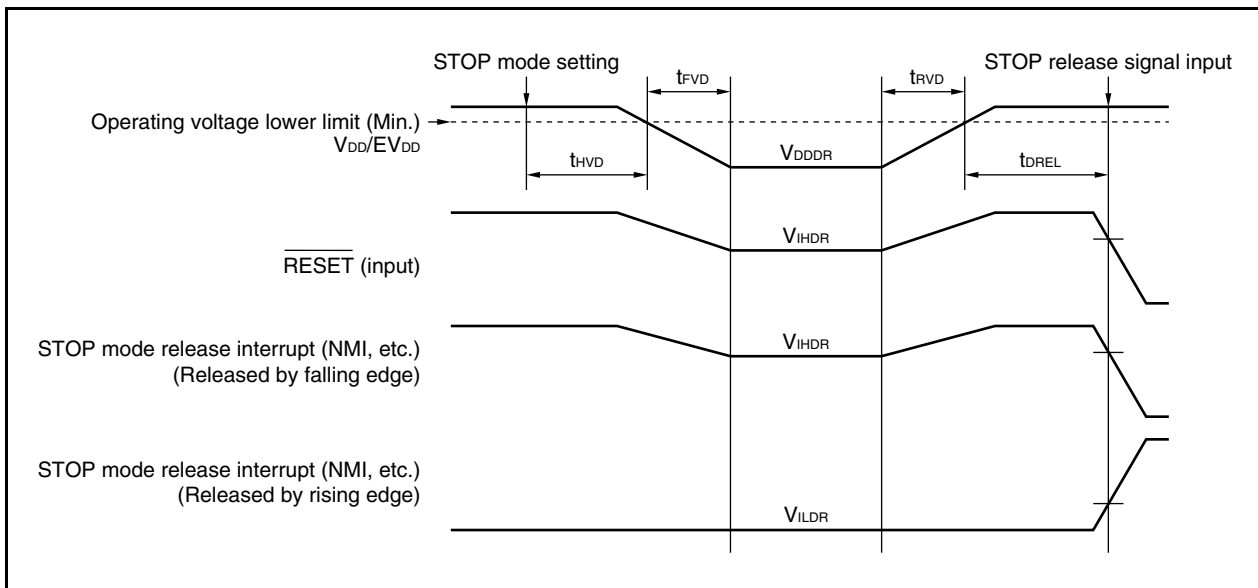
- Notes**
- Total current of V_{DD} and EV_{DD} (all ports stopped). The current of AV_{REF0} and the port buffer current including the current flowing through the on-chip pull-up/pull-down resistors are not included.
 - When the main clock oscillation is stopped.
 - POC operating, internal oscillator operating.
 - The RC oscillation frequency is 40 kHz (TYP.). This clock is internally divided by 2.
 - When the subclock oscillation is not used.

25.7 Data Retention Characteristics

STOP Mode ($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 1.9$ V to 5.5 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V)

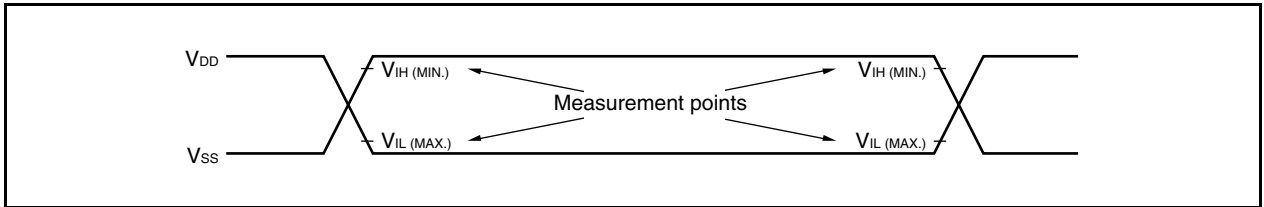
| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|------------------------------------|------------|--|---------------|------|---------------|---------------|
| Data retention voltage | V_{DDDR} | In STOP mode (all functions stopped) | 1.9 | | 5.5 | V |
| Data retention current | I_{DDDR} | $V_{DDDR} = 2.0$ V (all functions stopped) | | 6 | 45 | 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 release | 0 | | | ms |
| STOP release signal input time | t_{DREL} | After V_{DD} reaches MIN.: 3.5 V | 0 | | | μs |
| Data retention input voltage, high | V_{IHDR} | All input ports | $0.9V_{DDDR}$ | | V_{DDDR} | V |
| Data retention input voltage, low | V_{ILDR} | All input ports | 0 | | $0.1V_{DDDR}$ | V |

Caution Shifting to STOP mode and restoring from STOP mode must be performed within the rated operating range.

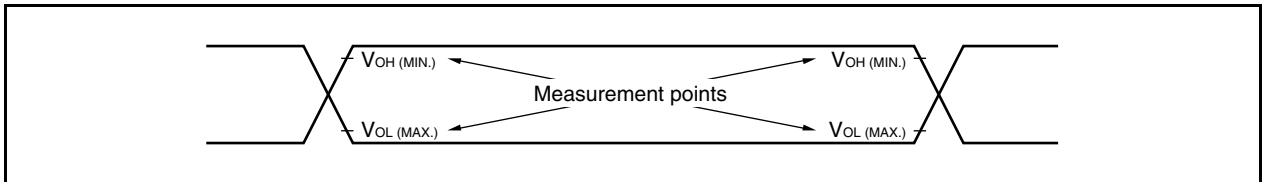


25.8 AC Characteristics

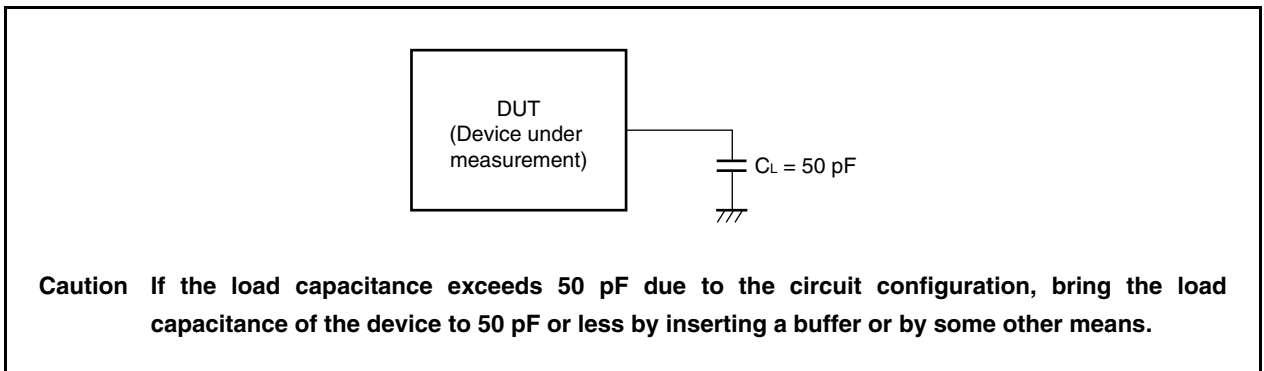
(1) AC test input measurement points (V_{DD} , AV_{REF0} , EV_{DD})



(2) AC test output measurement points



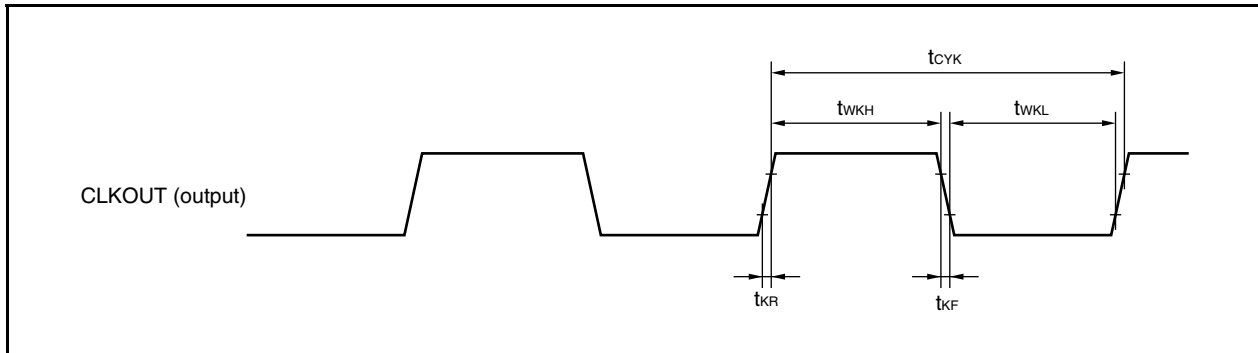
(3) Load conditions



25.8.1 CLKOUT output timing

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V to }5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

| Parameter | Symbol | Conditions | MIN. | MAX. | Unit |
|------------------|-----------|------------|------------------|------------------|------|
| Output cycle | t_{CYK} | | 50 ns | 80 μs | |
| High-level width | t_{WKH} | | $t_{CYK}/2 - 15$ | | ns |
| Low-level width | t_{WKL} | | $t_{CYK}/2 - 15$ | | ns |
| Rise time | t_{KR} | | | 15 | ns |
| Fall time | t_{KF} | | | 15 | ns |



25.9 Basic Operation

(1) Reset, interrupt timing

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5$ V to 5.5 V, 4.0 V $\leq AV_{REF0} \leq 5.5$ V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

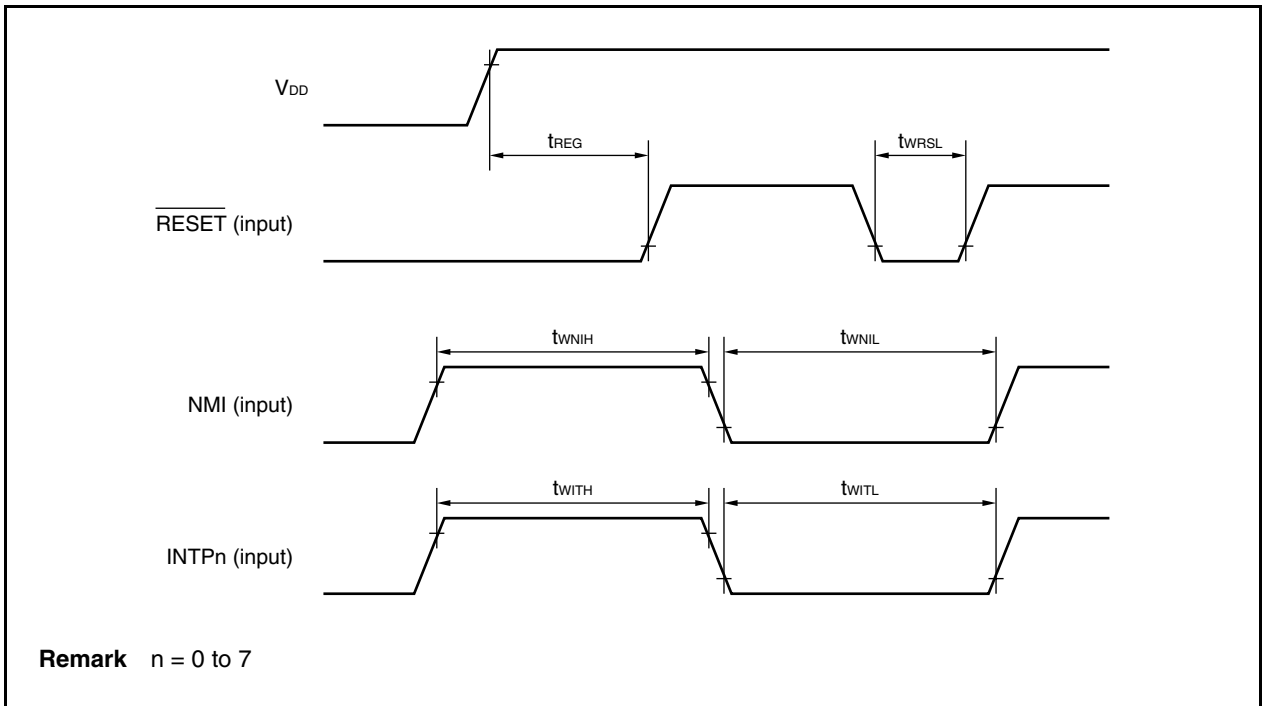
| Parameter | Symbol | Conditions | MIN. | MAX. | Unit |
|--|------------|---------------------------------------|---------------|------|------|
| RESET low-level width | t_{WRSL} | | 500 | | ns |
| NMI high-level width | t_{WNIH} | Analog noise elimination | 500 | | ns |
| NMI low-level width | t_{WNIL} | Analog noise elimination | 500 | | ns |
| INTPn ^{Note 1} high-level width | t_{WITH} | Analog noise elimination (n = 0 to 7) | 500 | | ns |
| | | Digital noise elimination (n = 3) | Note 2 | | ns |
| INTPn ^{Note 1} low-level width | t_{WITL} | Analog noise elimination (n = 0 to 7) | 500 | | ns |
| | | Digital noise elimination (n = 3) | Note 2 | | ns |

Notes 1. The same value as the INTP0/P03 pin applies in the case of the ADTRG pin. The same value as the INTP2/P05 pin applies in the case of the DRST pin.

2. $2T_{\text{samp}} + 20$ or $3T_{\text{samp}} + 20$

T_{samp} : Sampling clock for noise elimination

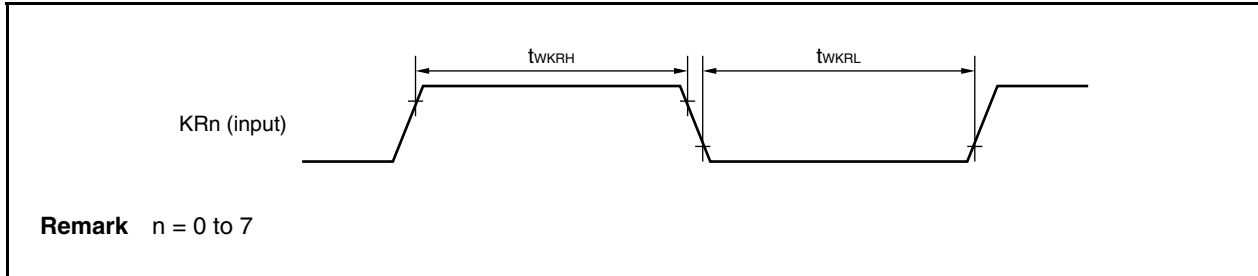
Reset/Interrupt



(2) Key interrupt timing

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V to }5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

| Parameter | Symbol | Conditions | MIN. | MAX. | Unit |
|----------------------------|------------|---|------|------|------|
| KRn input high-level width | t_{WKRH} | Analog noise elimination ($n = 0$ to 7) | 500 | | ns |
| KRn input low-level width | t_{WKRL} | | 500 | | ns |

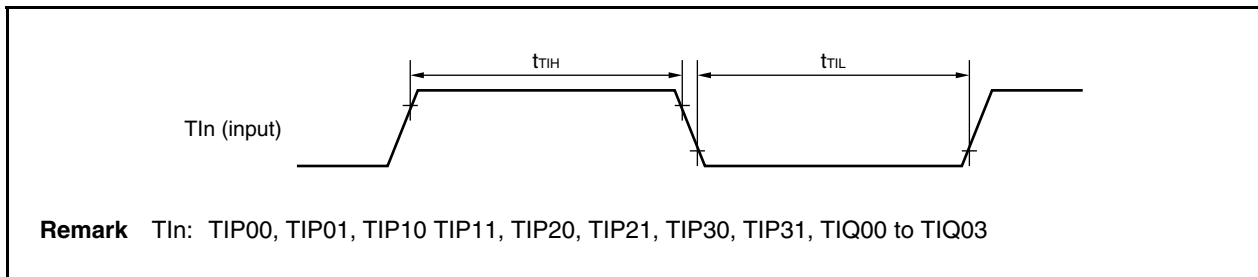


(3) Timer input timing

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V to }5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

| Parameter | Symbol | Conditions | MIN. | MAX. | Unit |
|----------------------|-----------|--|---------------|------|------|
| TIn high-level width | t_{TIH} | TIP00, TIP01, TIP10, TIP11, TIP20, TIP21, TIP30, TIP31, TIQ00 to TIQ03 ^{Note 1} | Note 2 | | ns |
| TIn low-level width | t_{TIL} | | Note 2 | | ns |

- Notes**
- Noise on the TIP00, TIP10, TIP20, TIP30, and TIQ00 pins can be eliminated only when a capture signal is input.
The noise cannot be eliminated when an external trigger signal or an external event counter signal is input.
 - $2T_{\text{samp}} + 20$ or $3T_{\text{samp}} + 20$
 T_{samp} : Sampling clock for noise elimination



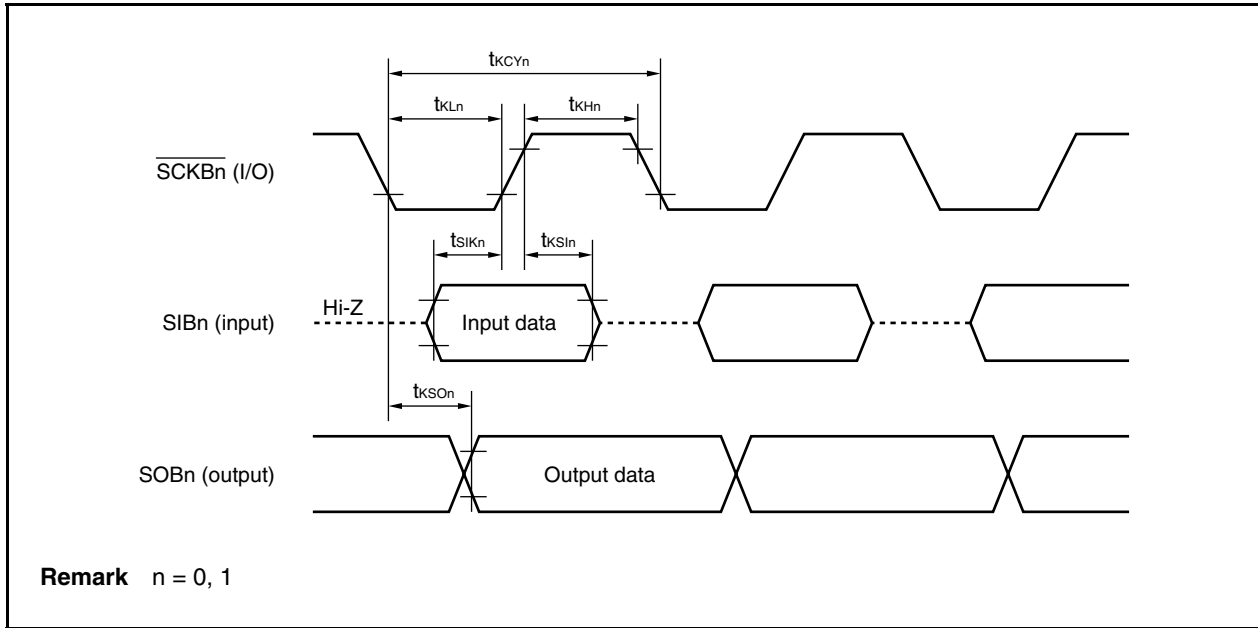
(4) CSIB timing**(a) Master mode****($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5$ V to 5.5 V, 4.0 V $\leq AV_{REF0} \leq 5.5$ V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)**

| Parameter | Symbol | Conditions | MIN. | MAX. | Unit |
|---|--------------------|------------|--------------------------|------|------|
| $\overline{\text{SCKBn}}$ cycle time | t_{KCYn} | | 125 | | ns |
| $\overline{\text{SCKBn}}$ high-level width | t_{KHn} | | $t_{\text{KCYn}}/2 - 15$ | | ns |
| $\overline{\text{SCKBn}}$ low-level width | t_{KLn} | | $t_{\text{KCYn}}/2 - 15$ | | ns |
| SIBn setup time (to $\overline{\text{SCKBn}}\uparrow$) | t_{SIKn} | | 30 | | ns |
| SIBn hold time (from $\overline{\text{SCKBn}}\uparrow$) | t_{KSIn} | | 25 | | ns |
| Output delay time from $\overline{\text{SCKBn}}\downarrow$ to SOBn | t_{KSOIn} | | | 25 | ns |

Remark $n = 0, 1$ **(b) Slave mode****($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5$ V to 5.5 V, 4.0 V $\leq AV_{REF0} \leq 5.5$ V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)**

| Parameter | Symbol | Conditions | MIN. | MAX. | Unit |
|---|--------------------|------------|------|------|------|
| $\overline{\text{SCKBn}}$ cycle time | t_{KCYn} | | 200 | | ns |
| $\overline{\text{SCKBn}}$ high-level width | t_{KHn} | | 90 | | ns |
| $\overline{\text{SCKBn}}$ low-level width | t_{KLn} | | 90 | | ns |
| SIBn setup time (to $\overline{\text{SCKBn}}\uparrow$) | t_{SIKn} | | 50 | | ns |
| SIBn hold time (from $\overline{\text{SCKBn}}\uparrow$) | t_{KSIn} | | 50 | | ns |
| Output delay time from $\overline{\text{SCKBn}}\downarrow$ to SOBn | t_{KSOIn} | | | 50 | ns |

Remark $n = 0, 1$



(5) UARTA timing

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V to }5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

| Parameter | Symbol | Conditions | MIN. | MAX. | Unit |
|--------------------|--------|------------|------|-------|------|
| Communication rate | | | | 312.5 | kbps |
| ASCK0 cycle time | | | | 10 | MHz |

(6) A/D converter

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V to }5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|-------------------------------|-------------|--|-----------|------------|-------------|---------------|
| Resolution | | | | | 10 | bit |
| Overall error ^{Note} | | $4.0 \leq AV_{REF0} \leq 5.5\text{ V}$ | | ± 0.15 | ± 0.3 | %FSR |
| Conversion time | t_{CONV} | | 3.1 | | 16 | μs |
| Analog input voltage | V_{IAN} | | AV_{SS} | | AV_{REF0} | V |
| AV_{REF0} current | I_{AREF0} | When using A/D converter | | 5 | 10 | mA |
| | | When not using A/D converter | | 1 | 10 | μA |

Note Excluding quantization error ($\pm 0.05\%$ FSR). Indicates the ratio to the full-scale value (%FSR).

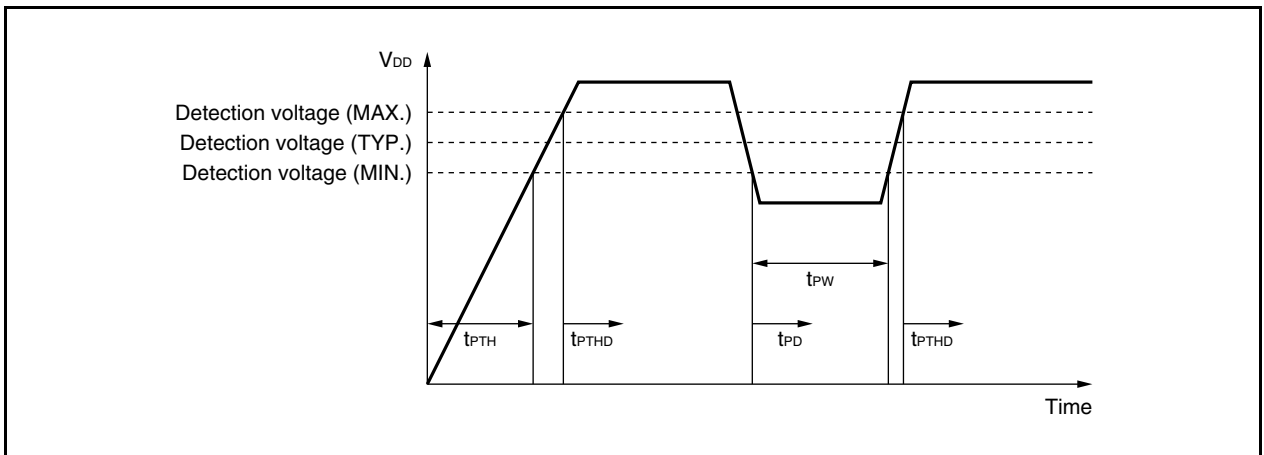
Remark FSR: Full Scale Range

(7) POC circuit characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5$ V to 5.5 V, 4.0 V $\leq AV_{REF0} \leq 5.5$ V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|------------|---|-------|------|------|------|
| Detection voltage | V_{POC0} | | 3.5 | 3.7 | 3.9 | V |
| Power supply startup time | t_{PTH} | $V_{DD} = 0$ V \rightarrow 3.5 V | 0.002 | | | ms |
| Response delay time 1 ^{Note 1} | t_{PTHD} | After V_{DD} reaches 3.9 V on power application | | | 3.0 | ms |
| Response delay time 2 ^{Note 2} | t_{PD} | After V_{DD} drops below 3.5 V on power drop | | | 1 | ms |
| Minimum V_{DD} width | t_{PW} | | 0.2 | | | ms |

- Notes**
1. The time required to release a reset after the detection voltage is detected.
 2. The time required to output a reset after the detection voltage is detected.

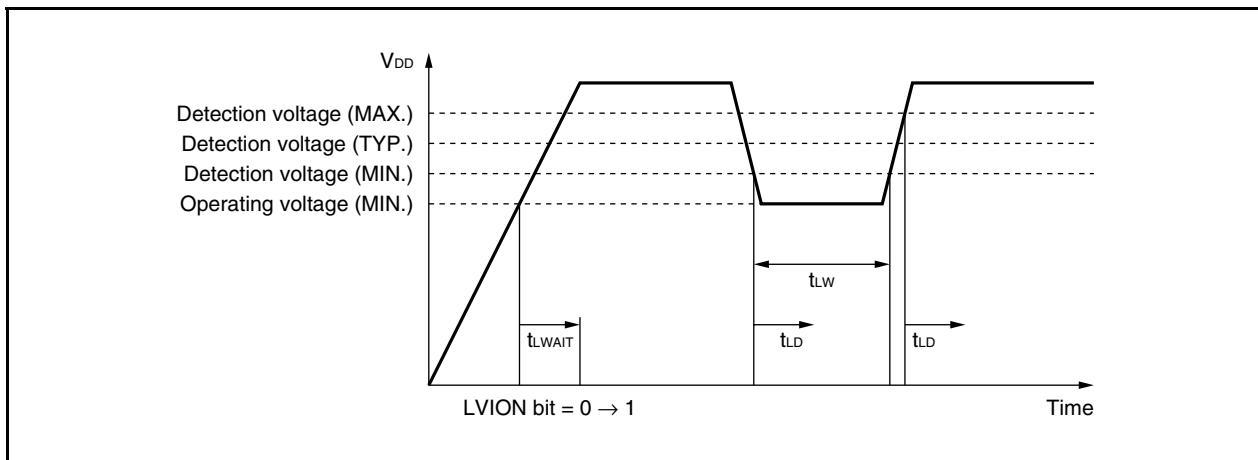


(8) LVI circuit characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5$ V to 5.5 V, 4.0 V $\leq AV_{REF0} \leq 5.5$ V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|-------------|---|------|------|------|------|
| Detection voltage | V_{LV10} | | 4.2 | 4.4 | 4.6 | V |
| | V_{LV11} | | 4.0 | 4.2 | 4.4 | V |
| Response time ^{Note 1} | t_{LD} | After V_{DD} reaches V_{LV10}/V_{LV11} (MAX.) or drops below V_{LV10}/V_{LV11} (MIN.) | | 0.2 | 2 | ms |
| Minimum V_{DD} width | t_{LW} | | 0.2 | | | ms |
| Reference voltage stabilization wait time ^{Note 2} | t_{LWAIT} | After V_{DD} reaches 3.5 V or LVION bit (LVIM.bit7) changes from 0 to 1 | | 0.1 | 0.2 | ms |

- Notes**
1. The time required to output an interrupt/reset after the detection voltage is detected.
 2. Unnecessary when the POC function is used.

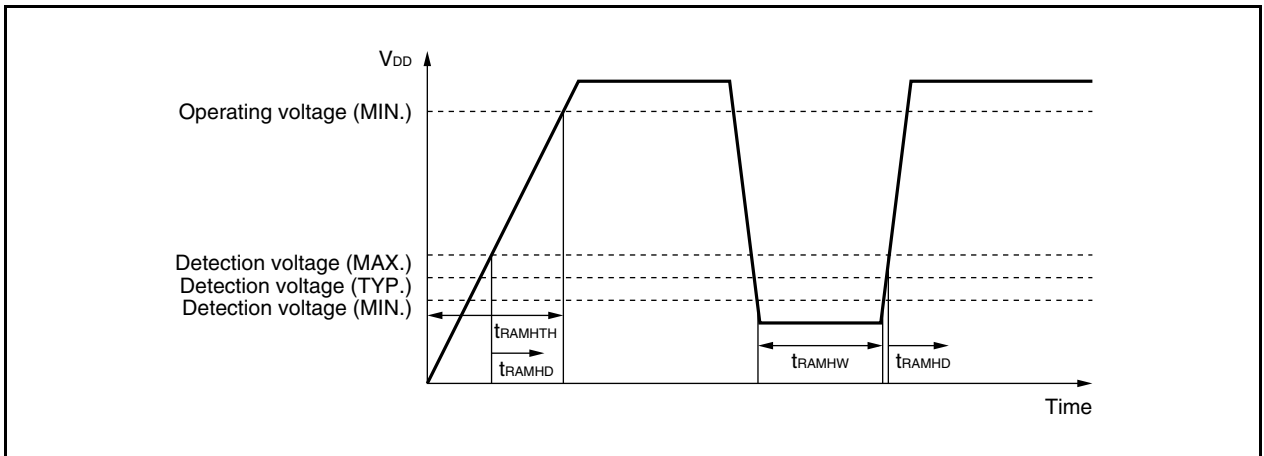


(9) RAM retention flag characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5$ V to 5.5 V, 4.0 V $\leq AV_{REF0} \leq 5.5$ V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ 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 \rightarrow 3.5 V | 0.002 | | 1800 | ms |
| Response time ^{Note} | t_{RAMHD} | After the supply voltage reaches the detection voltage (MAX.) | | 0.2 | 2.0 | ms |
| Minimum V_{DD} width | t_{RAMHW} | | 0.2 | | | ms |

Note Time required to set the RAMF bit after the detection voltage is detected.



25.10 Flash Memory Programming Characteristics

(1) Basic characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V to }5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

| 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 writes | C_{WRT}^{Note} | | | | 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 | $t_{WRT} + t_{ERASE}$ | | | | TBD | s |
| Programming temperature | t_{PRG} | | -40 | | +85 | $^\circ\text{C}$ |

Note When writing initially to shipped products, it is counted as one rewrite for both “erase to write” and “write only”.

Example (P: Write, E: Erase)

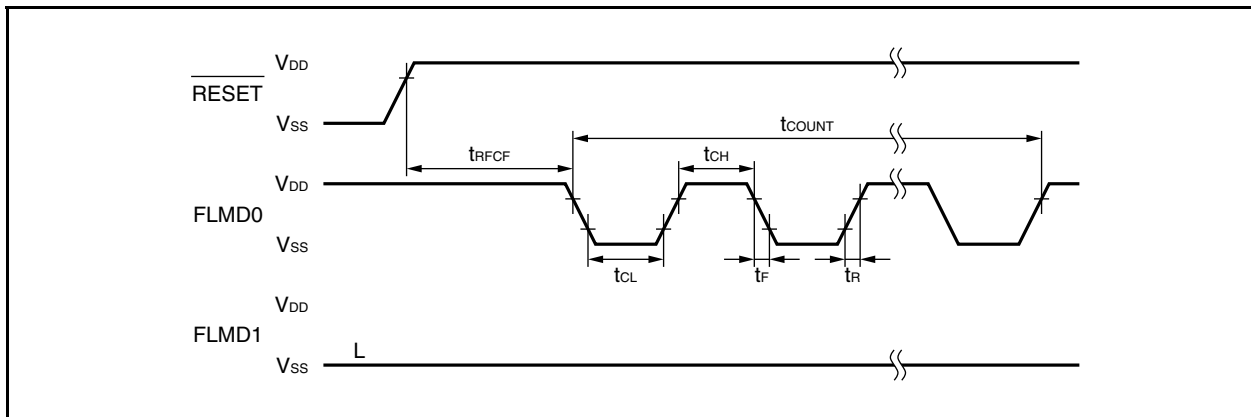
Shipped product → P → E → P → E → P: 3 rewrites

Shipped product → E → P → E → P → E → P: 3 rewrites

(2) Serial write operation characteristics

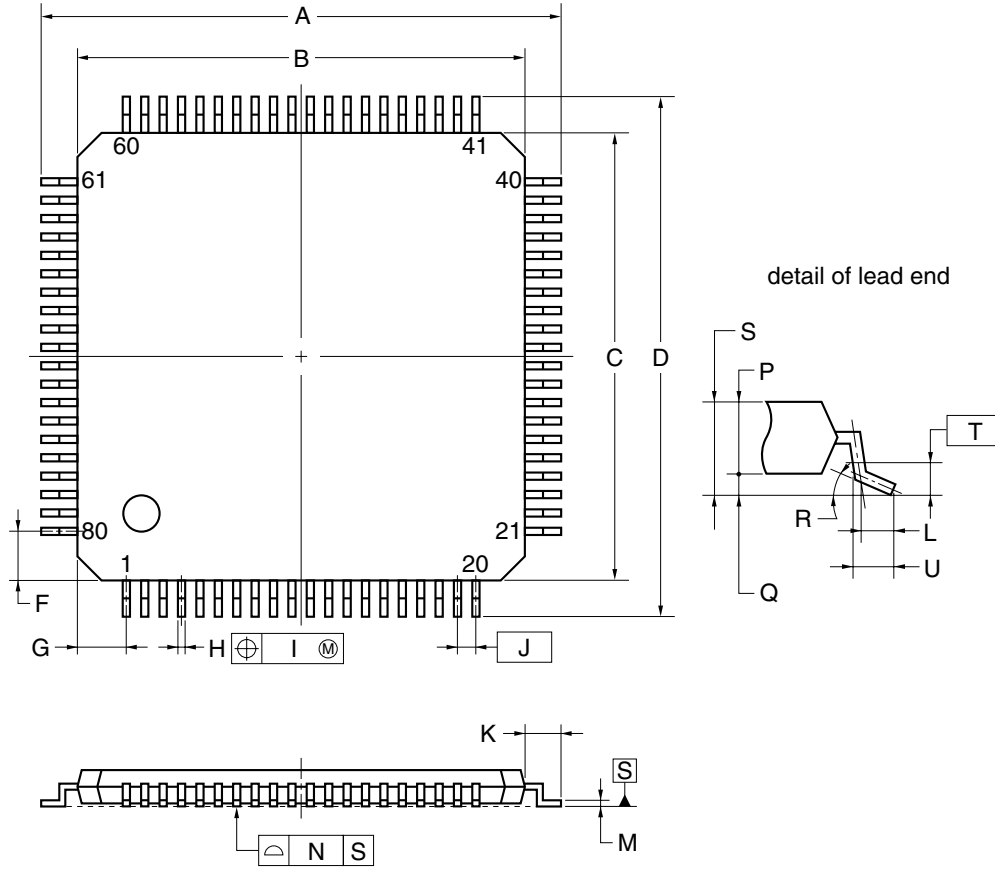
($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = 3.5\text{ V to }5.5\text{ V}$, $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|-------------|------------|-------------|------|------|---------------|
| FLMD0 setup time from $\overline{\text{RESET}}\uparrow$ | t_{RFCF} | | $70536/f_x$ | | | s |
| Count execution time | t_{COUNT} | | | | 3 | ms |
| FLMD0 high-level width | t_{CH} | | 10 | | 100 | μs |
| FLMD0 low-level width | t_{CL} | | 10 | | 100 | μs |
| FLMD0 rise time | t_R | | | | 50 | ns |
| FLMD0 fall time | t_F | | | | 50 | ns |



CHAPTER 26 PACKAGE DRAWING

80-PIN PLASTIC TQFP (FINE PITCH) (12x12)



NOTE

Each lead centerline is located within 0.08 mm of its true position (T.P.) at maximum material condition.

| ITEM | MILLIMETERS |
|------|---------------------------------|
| A | 14.0±0.2 |
| B | 12.0±0.2 |
| C | 12.0±0.2 |
| D | 14.0±0.2 |
| F | 1.25 |
| G | 1.25 |
| H | 0.22±0.05 |
| I | 0.08 |
| J | 0.5 (T.P.) |
| K | 1.0±0.2 |
| L | 0.5 |
| M | 0.145±0.05 |
| N | 0.08 |
| P | 1.0 |
| Q | 0.1±0.05 |
| R | 3 ⁺⁴ ₋₃ ° |
| S | 1.1±0.1 |
| T | 0.25 |
| U | 0.6±0.15 |

P80GK-50-9EU-1

CHAPTER 27 RECOMMENDED SOLDERING CONDITIONS

The V850ES/HF2 should be soldered and mounted under the following recommended conditions. For technical information, see the following website.

Semiconductor Device Mount Manual (<http://www.necel.com/pkg/en/mount/index.html>)

Table 27-1. Surface Mounting Type Soldering Conditions

PD70F3702GK-9EU-A: 80-pin plastic TQFP (fine pitch) (12 12)
PD70F3703GK-9EU-A: 80-pin plastic TQFP (fine pitch) (12 12)
PD70F3704GK-9EU-A: 80-pin plastic TQFP (fine pitch) (12 12)

| Soldering Method | Soldering Conditions | Recommended Condition Symbol |
|------------------|---|------------------------------|
| Infrared reflow | Package peak temperature: 260°C, Time: 60 seconds max. (at 220°C or higher), Count: Three times or less, Exposure limit: 7 days ^{Note} (after that, prebake at 125°C for 20 to 72 hours) | IR60-207-3 |
| Partial heating | Pin temperature: 350°C max., Time: 3 seconds max. (per pin row) | – |

Note After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Caution Do not use different soldering methods together (except for partial heating).

Remarks

1. Products with -A at the end of the part number are lead-free products.
2. For soldering methods and conditions other than those recommended above, please contact an NEC Electronics sales representative.

APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for the development of systems that employ the V850ES/HF2. Figure A-1 shows the development tool configuration.

- **Support for PC98-NX series**

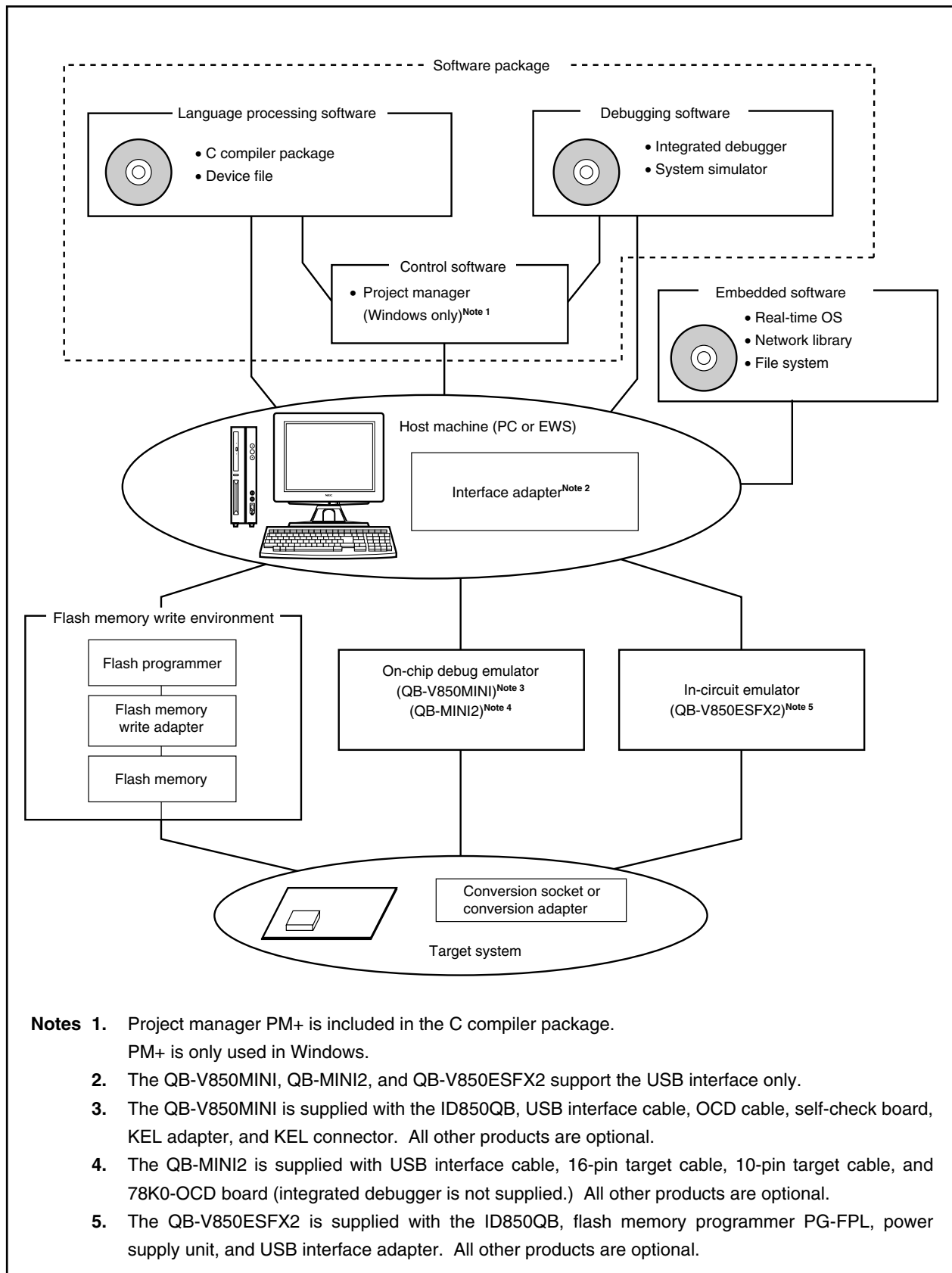
Unless otherwise specified, products supported by IBM PC/AT™ compatibles are compatible with PC98-NX series computers. When using PC98-NX series computers, refer to the explanation for IBM PC/AT compatibles.

- **Windows™**

Unless otherwise specified, "Windows" means the following OSs.

- Windows 98, 2000
- Windows Me
- Windows XP
- Windows NT™ Ver. 4.0

Figure A-1. Development Tool Configuration



A.1 Software Package

| | |
|--|---|
| SP850 Software package for V850 microcontrollers | Development tools (software) commonly used with V850 microcontrollers are included this package. |
| | Part number: μ SxxxxSP850 |

Remark xxxx in the part number differs depending on the host machine and OS used.

μ SxxxxSP850

| xxxx | Host Machine | OS | Supply Medium |
|------|-----------------------|----------------------------|---------------|
| AB17 | PC-9800 series, | Windows (Japanese version) | CD-ROM |
| BB17 | IBM PC/AT compatibles | Windows (English version) | |

A.2 Language Processing Software

| | |
|-----------------------------|--|
| CA850 C compiler package | This compiler converts programs written in C into object codes executable with a microcontroller. This compiler is started from project manager PM+. |
| | Part number: μ SxxxxCA703000 |
| DF703724 Device file | This file contains information peculiar to the device. This device file should be used in combination with a tool (CA850, SM+ for V850ES/Hx2, or ID850QB). The corresponding OS and host machine differ depending on the tool to be used. |

Remark xxxx in the part number differs depending on the host machine and OS used.

μ SxxxxCA703000

| xxxx | Host Machine | OS | Supply Medium |
|------|-----------------------|---|---------------|
| AB17 | PC-9800 series, | Windows (Japanese version) | CD-ROM |
| BB17 | IBM PC/AT compatibles | Windows (English version) | |
| 3K17 | SPARCstation™ | SunOS™ (Rel. 4.1.4), Solaris™ (Rel. 2.5.1) | |

A.3 Control Software

| | |
|------------------------|--|
| PM+ Project manager | This is control software designed to enable efficient user program development in the Windows environment. All operations used in development of a user program, such as starting the editor, building, and starting the debugger, can be performed from PM+. <Caution> PM+ is included in C compiler package CA850. It can only be used in Windows. |
|------------------------|--|

A.4 Debugging Tools (Hardware)

A.4.1 When using IECUBE QB-V850ESFX2

The system configuration when connecting the QB-V850ESFX2 to the host machine (PC-9821 series, PC/AT compatible) is shown below. Even if optional products are not prepared, connection is possible.

Figure A-2. System Configuration (When Using QB-V850ESFX2) (1/2)

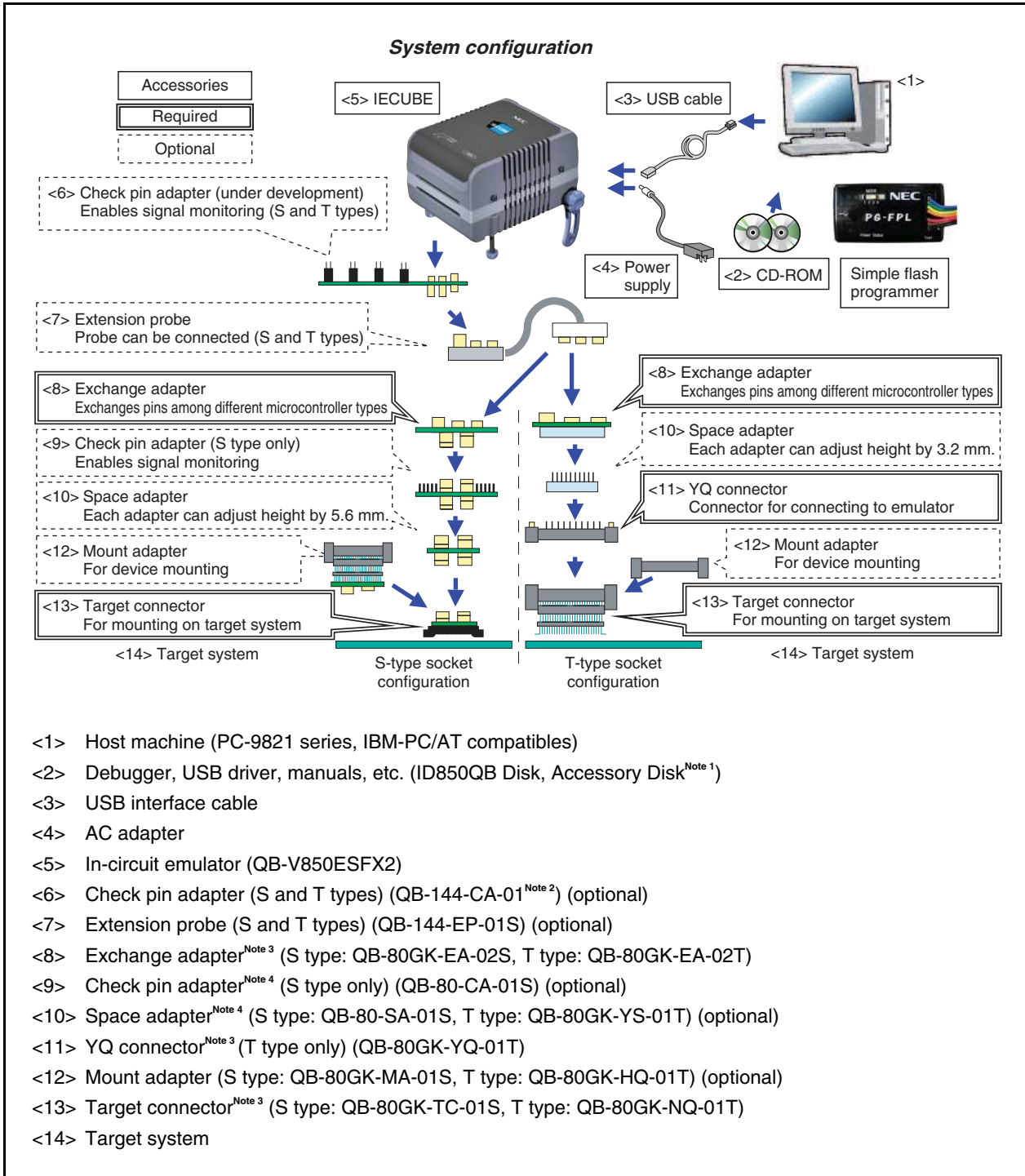


Figure A-2. System Configuration (When Using QB-V850ESFX2) (2/2)

- Notes**
1. Download the device file from the NEC Electronics website.
<http://www.necel.com/micro/ods/eng/>
 2. Under development
 3. Supplied with the device depending on the ordering number.
 - When QB-V850ESFX2-ZZZ is ordered
 The exchange adapter and the target connector are not supplied.
 - When QB-V850ESFX2-S80GK is ordered
 The QB-80GK-EA-02S and QB-80GK-TC-01S are supplied.
 - When QB-V850ESFX2-T80GK is ordered
 The QB-80GK-EA-02T, QB-80GK-YQ-01T, and QB-80GK-NQ-01T are supplied.
 4. When using both <9> and <10>, the order between <9> and <10> is not cared.

| | |
|---|---|
| <5> QB-V850ESFX2 ^{Note} In-circuit emulator | The in-circuit emulator serves to debug hardware and software when developing application systems using the V850ES/HF2. It supports to the integrated debugger ID850QB. This emulator should be used in combination with a power supply unit and emulation probe. Use the USB interface cable to connect this emulator to the host machine. |
| <3> USB interface cable | Cable to connect the host machine and the QB-V850ESFX2. |
| <4> AC adapter | 100 to 240 V can be supported by replacing the AC plug. |
| <8> QB-80GK-EA-02S QB-80GK-EA-02T Exchange adapter | Adapter to perform pin conversion. |
| <9> QB-80-CA-01S Check pin adapter | Adapter used in waveform monitoring using the oscilloscope, etc. |
| <10> QB-80-SA-01S QB-80GK-YS-01T Space adapter | Adapter to adjust the height. |
| <11> QB-80GK-YQ-01T YQ connector | Connector to connect to target connector and exchange adapter |
| <12> QB-80GK-MA-01S QB-80GK-HQ-01T Mount adapter | Adapter to mount the V850ES/HF2 with socket. |
| <13> QB-80GK-TC-01S QB-80GK-NQ-01T Target connector | Connector to solder on the target system. |

Note The QB-V850ESFX2 is supplied with a power supply unit, USB interface cable, and simple programmer PG-FPL. It is also supplied with integrated debugger ID850QB as control software.

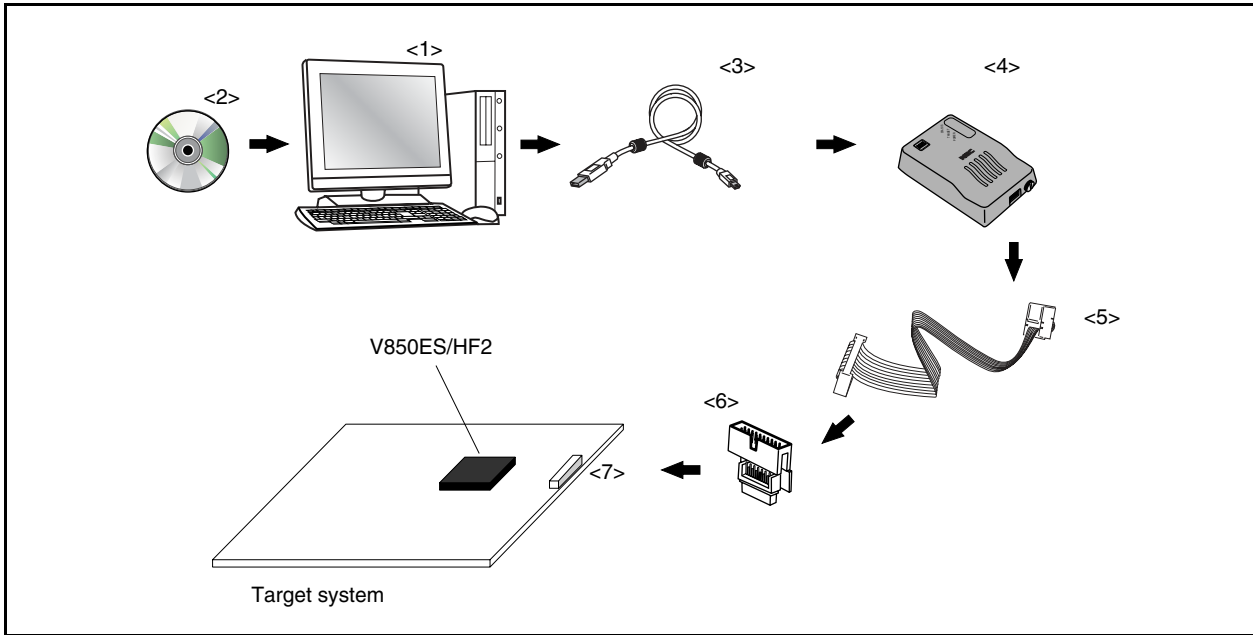
Remark The numbers in the angle brackets correspond to the numbers in Figure A-2.

A.4.2 When using MINICUBE QB-V850MINI

(1) On-chip emulation using MINICUBE

The system configuration when connecting MINICUBE to the host machine (PC-9821 series, PC/AT compatible) is shown below.

Figure A-3. On-Chip Emulation System Configuration



| | |
|---|--|
| <1> Host machine | PC with USB ports |
| <2> CD-ROM ^{Note 1} | Contents such as integrated debugger ID850QB, N-Wire Checker, device driver, and documents are included in CD-ROM. It is supplied with MINICUBE. |
| <3> USB interface cable | USB cable to connect the host machine and MINICUBE. It is supplied with MINICUBE. The cable length is approximately 2 m. |
| <4> MINICUBE On-chip debug emulator | This on-chip debug emulator serves to debug hardware and software when developing application systems using the V850ES/HF2. It supports integrated debugger ID850QB. |
| <5> OCD cable | Cable to connect MINICUBE and the target system. It is supplied with MINICUBE. The cable length is approximately 20 cm. |
| <6> Connector conversion board KEL adapter | This conversion board is supplied with MINICUBE. |
| <7> MINICUBE connector KEL connector ^{Note 2} | 8830E-026-170S (supplied with MINICUBE) 8830E-026-170L (sold separately) |

Notes 1. Download the device file from the NEC Electronics website.

<http://www.necel.com/micro/ods/eng/index.html>

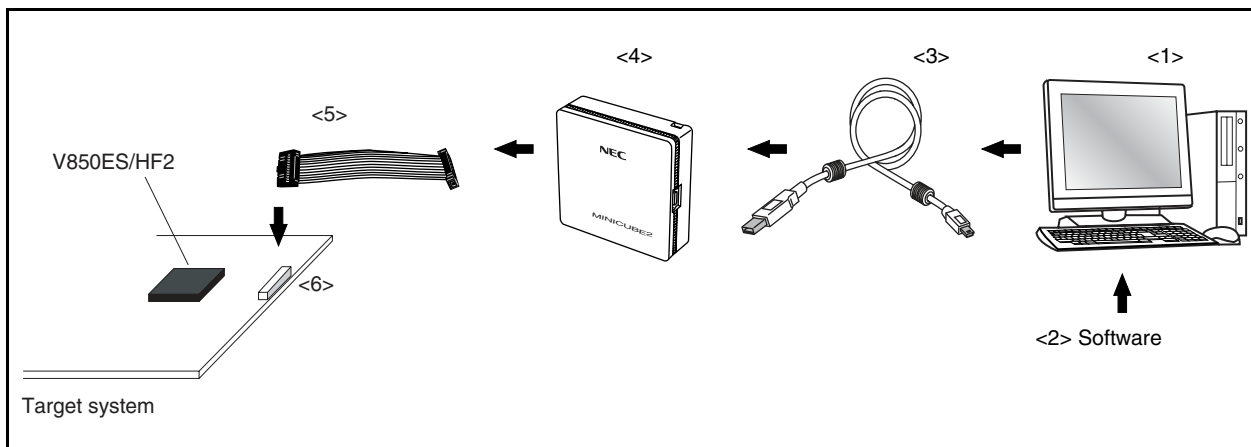
2. Product of KEL Corporation

Remark The numbers in the angular brackets correspond to the numbers in Figure A-3.

A.4.3 When using MINICUBE2 QB-MINI2

The system configuration when connecting MINICUBE2 to the host machine (PC-9821 series, PC/AT compatible) is shown below.

Figure A-4. System Configuration of On-Chip Emulation System



| | |
|---|---|
| <1> Host machine | PC with USB ports |
| <2> Software | The integrated debugger ID850QB, device file, etc. Download the device file from the NEC Electronics website. http://www.necel.com/micro/ods/eng/ |
| <3> USB interface cable | USB cable to connect the host machine and MINICUBE. It is supplied with MINICUBE. The cable length is approximately 2 m. |
| <4> MINICUBE2 On-chip debug emulator | This on-chip debug emulator serves to debug hardware and software when developing application systems using the V850ES/HF2. It supports integrated debugger ID850QB. |
| <5> 16-pin target cable | Cable to connect MINICUBE2 and the target system. It is supplied with MINICUBE. The cable length is approximately 15 cm. |
| <6> Target connector (sold separately) | Use a 16-pin general-purpose connector with 2.54 mm pitch. |

Remark The numbers in the angular brackets correspond to the numbers in Figure A-4.

A.5 Debugging Tools (Software)

| | |
|--|---|
| SM+ for V850ES/Hx2 (under development) System simulator | This simulator is used with V850 microcontrollers. SM+ for V850ES/Hx2 is Windows-based software. Debugging of C source and assembler files is possible during simulation of the target system operation on the host machine. By using SM+ for V850ES/Hx2, logic verification and performance verification of applications can be performed independently from hardware development. Therefore, development efficiency and software quality can be improved. It should be used in combination with the device file. |
| Part number: μ SxxxxSM703712-B | |
| ID850QB Integrated debugger | This debugger supports the in-circuit emulators for V850 microcontrollers. The ID850QB is Windows-based software. It has improved C-compatible debugging functions and can display the results of tracing with the source program using an integrating window function that associates the source program, disassemble display, and memory display with the trace result. It should be used in combination with the device file. |
| Part number: μ Sxxxx ID703000-QB (ID850QB) | |

Remark xxxx in the part number differs depending on the host machine and OS used.

μ SxxxxID703000-QB

| xxxx | Host Machine | OS | Supply Medium |
|------|-----------------------|----------------------------|---------------|
| AB17 | PC-9800 series, | Windows (Japanese version) | CD-ROM |
| BB17 | IBM PC/AT compatibles | Windows (English version) | |

A.6 Embedded Software

| | |
|----------------------------------|--|
| RX850, RX850 Pro Real-time OS | The RX850 and RX850 Pro are real-time OSs conforming to μ ITRON 3.0 specifications. A tool (configurator) for generating multiple information tables is supplied. RX850 Pro has more functions than the RX850. |
| | Part number: μ SxxxxRX703000- $\Delta\Delta\Delta\Delta$ (RX850) μ SxxxxRX703100- $\Delta\Delta\Delta\Delta$ (RX850 Pro) |
| Applilet ^{®Note} | This is a driver configurator that automatically generates sample programs for the V850ES/HF2. |
| RX-FS850 (File system) | This is a FAT file system function. It is a file system that supports the CD-ROM file system function. This file system is used with the real-time OS RX850 Pro. |

Note For how to obtain Applilet, consult an NEC Electronics sales representative.

Caution To purchase the RX850 or RX850 Pro, first fill in the purchase application form and sign the license agreement.

Remark xxxx and $\Delta\Delta\Delta\Delta$ in the part number differ depending on the host machine and OS used.

μ SxxxxRX703000- $\Delta\Delta\Delta\Delta$

μ SxxxxRX703100- $\Delta\Delta\Delta\Delta$

| $\Delta\Delta\Delta\Delta$ | Product Outline | Maximum Number for Use in Mass Production |
|----------------------------|------------------------|---|
| 001 | Evaluation object | Do not use for mass-produced product. |
| 100K | Mass-production object | 0.1 million units |
| 001M | | 1 million units |
| 010M | | 10 million units |
| S01 | Source program | Object source program for mass production |

| xxxx | Host Machine | OS | Supply Medium |
|------|--|----------------------------|---------------|
| AB17 | PC-9800 series, IBM PC/AT compatibles | Windows (Japanese version) | CD-ROM |
| BB17 | | Windows (English version) | |
| 3K17 | SPARCstation | Solaris (Rel. 2.5.1) | |

A.7 Flash Memory Writing Tools

| | |
|--|---|
| Flashpro IV (part number: PG-FP4) Flash programmer | Flash programmer dedicated to microcontrollers with on-chip flash memory. |
| QB-MINI2 (MINICUBE2) | On-chip debug emulator with programming function. |
| FA-80GK-9EU-A Flash memory writing adapter | Flash memory writing adapter used connected to the Flashpro IV, etc. (not wired). |
| FA-70F3704GK-9EU-MX Flash memory writing adapter | Flash memory writing adapter used connected to the Flashpro IV, etc. (already wired). |

Remark FA-80GK-9EU-A and FA-70F3704GK-9EU-MX are products of Naito Densei Machida Mfg. Co., Ltd.
TEL: +81-42-750-4172

APPENDIX B REGISTER INDEX

(1/7)

| Symbol | Name | Unit | Page |
|-----------|--|------|------|
| ADA0CR0 | A/D conversion result register 0 | ADC | 395 |
| ADA0CR0H | A/D conversion result register 0H | ADC | 395 |
| ADA0CR1 | A/D conversion result register 1 | ADC | 395 |
| ADA0CR1H | A/D conversion result register 1H | ADC | 395 |
| ADA0CR10 | A/D conversion result register 10 | ADC | 395 |
| ADA0CR10H | A/D conversion result register 10H | ADC | 395 |
| ADA0CR11 | A/D conversion result register 11 | ADC | 395 |
| ADA0CR11H | A/D conversion result register 11H | ADC | 395 |
| ADA0CR2 | A/D conversion result register 2 | ADC | 395 |
| ADA0CR2H | A/D conversion result register 2H | ADC | 395 |
| ADA0CR3 | A/D conversion result register 3 | ADC | 395 |
| ADA0CR3H | A/D conversion result register 3H | ADC | 395 |
| ADA0CR4 | A/D conversion result register 4 | ADC | 395 |
| ADA0CR4H | A/D conversion result register 4H | ADC | 395 |
| ADA0CR5 | A/D conversion result register 5 | ADC | 395 |
| ADA0CR5H | A/D conversion result register 5H | ADC | 395 |
| ADA0CR6 | A/D conversion result register 6 | ADC | 395 |
| ADA0CR6H | A/D conversion result register 6H | ADC | 395 |
| ADA0CR7 | A/D conversion result register 7 | ADC | 395 |
| ADA0CR7H | A/D conversion result register 7H | ADC | 395 |
| ADA0CR8 | A/D conversion result register 8 | ADC | 395 |
| ADA0CR8H | A/D conversion result register 8H | ADC | 395 |
| ADA0CR9 | A/D conversion result register 9 | ADC | 395 |
| ADA0CR9H | A/D conversion result register 9H | ADC | 395 |
| ADA0M0 | A/D converter mode register 0 | ADC | 390 |
| ADA0M1 | A/D converter mode register 1 | ADC | 392 |
| ADA0M2 | A/D converter mode register 2 | ADC | 393 |
| ADA0PFM | Power-fail compare mode register | ADC | 397 |
| ADA0PFT | Power-fail compare threshold value register | ADC | 397 |
| ADA0S | A/D converter channel specification register 0 | ADC | 394 |
| ADIC | Interrupt control register | INTC | 514 |
| CB0CTL0 | CSIB0 control register 0 | CSI | 454 |
| CB0CTL1 | CSIB0 control register 1 | CSI | 457 |
| CB0CTL2 | CSIB0 control register 2 | CSI | 458 |
| CB0RIC | Interrupt control register | INTC | 514 |
| CB0RX | CSIB0 receive data register | CSI | 453 |
| CB0RXL | CSIB0 receive data register L | CSI | 453 |
| CB0STR | CSIB0 status register | CSI | 460 |
| CB0TIC | Interrupt control register | INTC | 514 |
| CB0TX | CSIB0 transmit data register | CSI | 453 |
| CB0TXL | CSIB0 transmit data register L | CSI | 453 |

| Symbol | Name | Unit | Page |
|---------|---|------|------|
| CB1CTL0 | CSIB1 control register 0 | CSI | 454 |
| CB1CTL1 | CSIB1 control register 1 | CSI | 457 |
| CB1CTL2 | CSIB1 control register 2 | CSI | 458 |
| CB1RIC | Interrupt control register | INTC | 514 |
| CB1RX | CSIB1 receive data register | CSI | 453 |
| CB1RXL | CSIB1 receive data register L | CSI | 453 |
| CB1STR | CSIB1 status register | CSI | 460 |
| CB1TIC | Interrupt control register | INTC | 514 |
| CB1TX | CSIB1 transmit data register | CSI | 453 |
| CB1TXL | CSIB1 transmit data register L | CSI | 453 |
| CCLS | CPU operation clock status register | CG | 153 |
| CLM | Clock monitor mode register | CLM | 465 |
| CTBP | CALLT base pointer | CPU | 49 |
| CTPC | CALLT execution status saving register | CPU | 48 |
| CTPSW | CALLT execution status saving register | CPU | 48 |
| DBPC | Exception/debug trap status saving register | CPU | 49 |
| DBPSW | Exception/debug trap status saving register | CPU | 49 |
| ECR | Interrupt source register | CPU | 46 |
| EIPC | Interrupt status saving register | CPU | 45 |
| EIPSW | Interrupt status saving register | CPU | 45 |
| FEPC | NMI status saving register | CPU | 46 |
| FEPSW | NMI status saving register | CPU | 46 |
| IMR0 | Interrupt mask register 0 | INTC | 516 |
| IMR0H | Interrupt mask register 0H | INTC | 516 |
| IMR0L | Interrupt mask register 0L | INTC | 516 |
| IMR1 | Interrupt mask register 1 | INTC | 516 |
| IMR1H | Interrupt mask register 1H | INTC | 516 |
| IMR1L | Interrupt mask register 1L | INTC | 516 |
| IMR2 | Interrupt mask register 2 | INTC | 516 |
| IMR2H | Interrupt mask register 2H | INTC | 516 |
| IMR2L | Interrupt mask register 2L | INTC | 516 |
| INTF0 | External interrupt falling edge specification register 0 | INTC | 527 |
| INTF3L | External interrupt falling edge specification register 3L | INTC | 528 |
| INTF9H | External interrupt falling edge specification register 9H | INTC | 529 |
| INTR0 | External interrupt rising edge specification register 0 | INTC | 527 |
| INTR3L | External interrupt rising edge specification register 3L | INTC | 528 |
| INTR9H | External interrupt rising edge specification register 9H | INTC | 529 |
| ISPR | In-service priority register | INTC | 517 |
| KRIC | Interrupt control register | INTC | 514 |
| KRM | Key return mode register | KR | 535 |
| LOCKR | Lock register | CG | 156 |
| LVIIC | Interrupt control register | INTC | 514 |
| LVIM | Low-voltage detection register | LVI | 572 |
| LVIS | Low-voltage detection level select register | LVI | 573 |

| Symbol | Name | Unit | Page |
|--------|--|---------|------|
| NFC | Noise elimination control register | INTC | 530 |
| OCDM | On-chip debug mode register | DCU | 609 |
| OSTS | Oscillation stabilization time select register | Standby | 540 |
| P0 | Port 0 | Port | 80 |
| P00NFC | TIP00 pin noise elimination control register | Timer | 176 |
| P01NFC | TIP01 pin noise elimination control register | Timer | 176 |
| P10NFC | TIP10 pin noise elimination control register | Timer | 176 |
| P11NFC | TIP11 pin noise elimination control register | Timer | 176 |
| P20NFC | TIP20 pin noise elimination control register | Timer | 176 |
| P21NFC | TIP21 pin noise elimination control register | Timer | 176 |
| P3 | Port 3 | Port | 84 |
| P30NFC | TIP30 pin noise elimination control register | Timer | 176 |
| P31NFC | TIP31 pin noise elimination control register | Timer | 176 |
| P3H | Port 3H | Port | 84 |
| P3L | Port 3L | Port | 84 |
| P4 | Port 4 | Port | 89 |
| P5 | Port 5 | Port | 92 |
| P7H | Port 7H | Port | 98 |
| P7L | Port 7L | Port | 98 |
| P9 | Port 9 | Port | 100 |
| P9H | Port 9H | Port | 100 |
| P9L | Port 9L | Port | 100 |
| PC | Program counter | CPU | 43 |
| PCC | Processor clock control register | CG | 149 |
| PCLM | Programmable clock mode register | CG | 158 |
| PCM | Port CM | Port | 108 |
| PCS | Port CS | Port | 110 |
| PCT | Port CT | Port | 112 |
| PDL | Port DL | Port | 114 |
| PDLH | Port DLH | Port | 114 |
| PDLL | Port DLL | Port | 114 |
| PEMU1 | Peripheral emulation register 1 | LVI | 578 |
| PFC0 | Port function control register 0 | Port | 82 |
| PFC3L | Port function control register 3L | Port | 86 |
| PFC5 | Port function control register 5 | Port | 94 |
| PFC9 | Port function control register 9 | Port | 102 |
| PFC9H | Port function control register 9H | Port | 102 |
| PFC9L | Port function control register 9L | Port | 102 |
| PFCE3L | Port function control expansion register 3L | Port | 86 |
| PFCE5 | Port function control expansion register 5 | Port | 94 |
| PFCE9 | Port function control expansion register 9 | Port | 103 |
| PFCE9H | Port function control expansion register 9H | Port | 103 |
| PFCE9L | Port function control expansion register 9L | Port | 103 |
| PIC0 | Interrupt control register | INTC | 514 |

| Symbol | Name | Unit | Page |
|--------|--|---------|----------|
| PIC1 | Interrupt control register | INTC | 514 |
| PIC2 | Interrupt control register | INTC | 514 |
| PIC3 | Interrupt control register | INTC | 514 |
| PIC4 | Interrupt control register | INTC | 514 |
| PIC5 | Interrupt control register | INTC | 514 |
| PIC6 | Interrupt control register | INTC | 514 |
| PIC7 | Interrupt control register | INTC | 514 |
| PLLCTL | PLL control register | CG | 155 |
| PLLS | PLL lockup time specification register | CG | 157 |
| PM0 | Port mode register 0 | Port | 80 |
| PM3 | Port mode register 3 | Port | 84 |
| PM3H | Port mode register 3H | Port | 84 |
| PM3L | Port mode register 3L | Port | 84 |
| PM4 | Port mode register 4 | Port | 89 |
| PM5 | Port mode register 5 | Port | 92 |
| PM7H | Port mode register 7H | Port | 98 |
| PM7L | Port mode register 7L | Port | 98 |
| PM9 | Port mode register 9 | Port | 100 |
| PM9H | Port mode register 9H | Port | 100 |
| PM9L | Port mode register 9L | Port | 100 |
| PMC0 | Port mode control register 0 | Port | 81 |
| PMC3L | Port mode control register 3L | Port | 85 |
| PMC4 | Port mode control register 4 | Port | 90 |
| PMC5 | Port mode control register 5 | Port | 93 |
| PMC9 | Port mode control register 9 | Port | 101 |
| PMC9H | Port mode control register 9H | Port | 101 |
| PMC9L | Port mode control register 9L | Port | 101 |
| PMCCM | Port mode control register CM | Port | 108 |
| PMCM | Port mode register CM | Port | 108 |
| PMCS | Port mode register CS | Port | 110 |
| PMCT | Port mode register CT | Port | 112 |
| PMDL | Port mode register DL | Port | 114 |
| PMDLH | Port mode register DLH | Port | 114 |
| PMDLL | Port mode register DLL | Port | 114 |
| PRCMD | Command register | CPU | 70 |
| PRSCM0 | Prescaler compare register 0 | WT | 374, 497 |
| PRSM0 | Prescaler mode register 0 | WT | 374, 497 |
| PSC | Power save control register | Standby | 538 |
| PSMR | Power save mode register | Standby | 539 |
| PSW | Program status word | CPU | 47 |
| PU0 | Pull-up resistor option register 0 | Port | 82 |
| PU3 | Pull-up resistor option register 3 | Port | 87 |
| PU3H | Pull-up resistor option register 3H | Port | 87 |
| PU3L | Pull-up resistor option register 3L | Port | 87 |

| Symbol | Name | Unit | Page |
|-----------|--|-------|------|
| PU4 | Pull-up resistor option register 4 | Port | 90 |
| PU5 | Pull-up resistor option register 5 | Port | 96 |
| PU9 | Pull-up resistor option register 9 | Port | 106 |
| PU9H | Pull-up resistor option register 9H | Port | 106 |
| PU9L | Pull-up resistor option register 9L | Port | 106 |
| Q00NFC | TIQ00 pin noise elimination control register | Timer | 276 |
| Q01NFC | TIQ01 pin noise elimination control register | Timer | 276 |
| Q02NFC | TIQ02 pin noise elimination control register | Timer | 276 |
| Q03NFC | TIQ03 pin noise elimination control register | Timer | 276 |
| r0 to r31 | General-purpose register | CPU | 43 |
| RAMS | Internal RAM data status register | LVI | 573 |
| RCM | Internal oscillation mode register | CG | 153 |
| RESF | Reset source flag register | Reset | 557 |
| SELCNT0 | Selector operation control register 0 | Timer | 253 |
| SYS | System status register | CPU | 71 |
| TM0CMP0 | TMM0 compare register 0 | Timer | 363 |
| TM0CTL0 | TMM0 control register 0 | Timer | 364 |
| TM0EQIC0 | Interrupt control register | INTC | 514 |
| TP0CCIC0 | Interrupt control register | INTC | 514 |
| TP0CCIC1 | Interrupt control register | INTC | 514 |
| TP0CCR0 | TMP0 capture/compare register 0 | Timer | 171 |
| TP0CCR1 | TMP0 capture/compare register 1 | Timer | 173 |
| TP0CNT | TMP0 counter read buffer register | Timer | 175 |
| TP0CTL0 | TMP0 control register 0 | Timer | 164 |
| TP0CTL1 | TMP0 control register 1 | Timer | 165 |
| TP0IOC0 | TMP0 I/O control register 0 | Timer | 167 |
| TP0IOC1 | TMP0 I/O control register 1 | Timer | 168 |
| TP0IOC2 | TMP0 I/O control register 2 | Timer | 169 |
| TP0OPT0 | TMP0 option register 0 | Timer | 170 |
| TP0OVIC | Interrupt control register | INTC | 514 |
| TP1CCIC0 | Interrupt control register | INTC | 514 |
| TP1CCIC1 | Interrupt control register | INTC | 514 |
| TP1CCR0 | TMP1 capture/compare register 0 | Timer | 171 |
| TP1CCR1 | TMP1 capture/compare register 1 | Timer | 173 |
| TP1CNT | TMP1 counter read buffer register | Timer | 175 |
| TP1CTL0 | TMP1 control register 0 | Timer | 164 |
| TP1CTL1 | TMP1 control register 1 | Timer | 165 |
| TP1IOC0 | TMP1 I/O control register 0 | Timer | 167 |
| TP1IOC1 | TMP1 I/O control register 1 | Timer | 168 |
| TP1IOC2 | TMP1 I/O control register 2 | Timer | 169 |
| TP1OPT0 | TMP1 option register 0 | Timer | 170 |
| TP1OVIC | Interrupt control register | INTC | 514 |
| TP2CCIC0 | Interrupt control register | INTC | 514 |
| TP2CCIC1 | Interrupt control register | INTC | 514 |

| Symbol | Name | Unit | Page |
|----------|-----------------------------------|-------|------|
| TP2CCR0 | TMP2 capture/compare register 0 | Timer | 171 |
| TP2CCR1 | TMP2 capture/compare register 1 | Timer | 173 |
| TP2CNT | TMP2 counter read buffer register | Timer | 175 |
| TP2CTL0 | TMP2 control register 0 | Timer | 164 |
| TP2CTL1 | TMP2 control register 1 | Timer | 165 |
| TP2IOC0 | TMP2 I/O control register 0 | Timer | 167 |
| TP2IOC1 | TMP2 I/O control register 1 | Timer | 168 |
| TP2IOC2 | TMP2 I/O control register 2 | Timer | 169 |
| TP2OPT0 | TMP2 option register 0 | Timer | 170 |
| TP2OVIC | Interrupt control register | INTC | 514 |
| TP3CCIC0 | Interrupt control register | INTC | 514 |
| TP3CCIC1 | Interrupt control register | INTC | 514 |
| TP3CCR0 | TMP3 capture/compare register 0 | Timer | 171 |
| TP3CCR1 | TMP3 capture/compare register 1 | Timer | 173 |
| TP3CNT | TMP3 counter read buffer register | Timer | 175 |
| TP3CTL0 | TMP3 control register 0 | Timer | 164 |
| TP3CTL1 | TMP3 control register 1 | Timer | 165 |
| TP3IOC0 | TMP3 I/O control register 0 | Timer | 167 |
| TP3IOC1 | TMP3 I/O control register 1 | Timer | 168 |
| TP3IOC2 | TMP3 I/O control register 2 | Timer | 169 |
| TP3OPT0 | TMP3 option register 0 | Timer | 170 |
| TP3OVIC | Interrupt control register | INTC | 514 |
| TQ0CCIC0 | Interrupt control register | INTC | 514 |
| TQ0CCIC1 | Interrupt control register | INTC | 514 |
| TQ0CCIC2 | Interrupt control register | INTC | 514 |
| TQ0CCIC3 | Interrupt control register | INTC | 514 |
| TQ0CCR0 | TMQ0 capture/compare register 0 | Timer | 267 |
| TQ0CCR1 | TMQ0 capture/compare register 1 | Timer | 269 |
| TQ0CCR2 | TMQ0 capture/compare register 2 | Timer | 271 |
| TQ0CCR3 | TMQ0 capture/compare register 3 | Timer | 273 |
| TQ0CNT | TMQ0 counter read buffer register | Timer | 275 |
| TQ0CTL0 | TMQ0 control register 0 | Timer | 260 |
| TQ0CTL1 | TMQ0 control register 1 | Timer | 261 |
| TQ0IOC0 | TMQ0 I/O control register 0 | Timer | 263 |
| TQ0IOC1 | TMQ0 I/O control register 1 | Timer | 264 |
| TQ0IOC2 | TMQ0 I/O control register 2 | Timer | 265 |
| TQ0OPT0 | TMQ0 option register 0 | Timer | 266 |
| TQ0OVIC | Interrupt control register | INTC | 514 |
| UA0CTL0 | UARTA0 control register 0 | UART | 421 |
| UA0CTL1 | UARTA0 control register 1 | UART | 443 |
| UA0CTL2 | UARTA0 control register 2 | UART | 444 |
| UA0OPT0 | UARTA0 option control register 0 | UART | 423 |
| UA0RIC | Interrupt control register | INTC | 514 |
| UA0RX | UARTA0 receive data register | UART | 426 |

| Symbol | Name | Unit | Page |
|---------|-------------------------------------|------|----------|
| UA0STR | UARTA0 status register | UART | 424 |
| UA0TIC | Interrupt control register | INTC | 514 |
| UA0TX | UARTA0 transmit data register | UART | 426 |
| UA1CTL0 | UARTA1 control register 0 | UART | 421 |
| UA1CTL1 | UARTA1 control register 1 | UART | 443 |
| UA1CTL2 | UARTA1 control register 2 | UART | 444 |
| UA1OPT0 | UARTA1 option control register 0 | UART | 423 |
| UA1RIC | Interrupt control register | INTC | 514 |
| UA1RX | UARTA1 receive data register | UART | 426 |
| UA1STR | UARTA1 status register | UART | 424 |
| UA1TIC | Interrupt control register | INTC | 514 |
| UA1TX | UARTA1 transmit data register | UART | 426 |
| VSWC | System wait control register | CPU | 72 |
| WDTE | Watchdog timer enable register | WDT | 384 |
| WDTM2 | Watchdog timer mode register 2 | WDT | 382, 518 |
| WTIC | Interrupt control register | INTC | 514 |
| WTIIC | Interrupt control register | INTC | 514 |
| WTM | Watch timer operation mode register | WT | 375 |

APPENDIX C INSTRUCTION SET LIST

C.1 Conventions

(1) Register symbols used to describe operands

| Register Symbol | Explanation |
|-----------------|--|
| reg1 | General-purpose registers: Used as source registers. |
| reg2 | General-purpose registers: Used mainly as destination registers. Also used as source register in some instructions. |
| reg3 | General-purpose registers: Used mainly to store the remainders of division results and the higher 32 bits of multiplication results. |
| 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 conditions code |
| 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 |
| l | 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 | Half word (16 bits) |
| Word | Word (32 bits) |
| + | Addition |
| − | Subtraction |
| | 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

| Identifier | Explanation |
|------------|--|
| (Blank) | No change |
| 0 | Clear to 0 |
| X | Set or cleared in accordance with the results. |
| R | Previously saved values are restored. |

(6) Condition codes

| Condition Code (cccc) | Condition Formula | Explanation |
|--------------------------|---|---|
| 0 0 0 0 | $OV = 1$ | Overflow |
| 1 0 0 0 | $OV = 0$ | No overflow |
| 0 0 0 1 | $CY = 1$ | Carry Lower (Less than) |
| 1 0 0 1 | $CY = 0$ | No carry Not lower (Greater than or equal) |
| 0 0 1 0 | $Z = 1$ | Zero |
| 1 0 1 0 | $Z = 0$ | Not zero |
| 0 0 1 1 | $(CY \text{ or } Z) = 1$ | Not higher (Less than or equal) |
| 1 0 1 1 | $(CY \text{ or } Z) = 0$ | Higher (Greater than) |
| 0 1 0 0 | $S = 1$ | Negative |
| 1 1 0 0 | $S = 0$ | Positive |
| 0 1 0 1 | – | Always (Unconditional) |
| 1 1 0 1 | $SAT = 1$ | Saturated |
| 0 1 1 0 | $(S \text{ xor } OV) = 1$ | Less than signed |
| 1 1 1 0 | $(S \text{ xor } OV) = 0$ | Greater than or equal signed |
| 0 1 1 1 | $((S \text{ xor } OV) \text{ or } Z) = 1$ | Less than or equal signed |
| 1 1 1 1 | $((S \text{ xor } OV) \text{ or } Z) = 0$ | Greater than signed |

C.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 | x | x | x | x | | |
| | imm5,reg2 | rrrrr010010iiii | GR[reg2]←GR[reg2]+sign-extend(imm5) | 1 | 1 | 1 | x | x | x | x | | |
| ADDI | imm16,reg1,reg2 | rrrrr110000RRRRR iiiiiiiiiiiiiiii | GR[reg2]←GR[reg1]+sign-extend(imm16) | 1 | 1 | 1 | x | x | x | x | | |
| AND | reg1,reg2 | rrrrr001010RRRRR | GR[reg2]←GR[reg2]AND GR[reg1] | 1 | 1 | 1 | | 0 | x | x | | |
| ANDI | imm16,reg1,reg2 | rrrrr110110RRRRR iiiiiiiiiiiiiiii | GR[reg2]←GR[reg1]AND zero-extend(imm16) | 1 | 1 | 1 | | 0 | x | x | | |
| Bcond | disp9 | dddd1011dddcccc Note 1 | if conditions are satisfied then PC←PC+sign-extend(disp9) | When conditions are satisfied | 2 | 2 | 2 | | | | | |
| | | | | When conditions are not satisfied | 1 | 1 | 1 | | | | | |
| BSH | reg2,reg3 | rrrrr11111100000 wwwww01101000010 | GR[reg3]←GR[reg2] (23 : 16) GR[reg2] (31 : 24) GR[reg2] (7 : 0) GR[reg2] (15 : 8) | 1 | 1 | 1 | x | 0 | x | x | | |
| BSW | reg2,reg3 | rrrrr11111100000 wwwww01101000000 | GR[reg3]←GR[reg2] (7 : 0) GR[reg2] (15 : 8) GR [reg2] (23 : 16) GR[reg2] (31 : 24) | 1 | 1 | 1 | x | 0 | x | x | | |
| CALLT | imm6 | 0000001000iiii | 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 dddddddddddddd | adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,0) | 3 | 3 | 3 | | | | x | | |
| | reg2,[reg1] | rrrrr111111RRRRR 0000000011100100 | adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,0) | 3 | 3 | 3 | | | | x | | |
| CMOV | cccc,imm5,reg2,reg3 | rrrrr111111iiii wwwww01100cccc0 | if conditions are satisfied then GR[reg3]←sign-extended(imm5) else GR[reg3]←GR[reg2] | 1 | 1 | 1 | | | | | | |
| | cccc,reg1,reg2,reg3 | rrrrr111111RRRRR 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 | x | x | x | x | | |
| | imm5,reg2 | rrrrr010011iiii | result←GR[reg2]-sign-extend(imm5) | 1 | 1 | 1 | x | x | x | x | | |
| CTRET | | 000001111100000 0000000101000100 | PC←CTPC PSW←CTPSW | 3 | 3 | 3 | R | R | R | R | R | |
| DBRET | | 000001111100000 0000000101000110 | PC←DBPC PSW←DBPSW | 3 | 3 | 3 | R | R | R | R | R | |

APPENDIX C INSTRUCTION SET LIST

(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 | 0000011001iiiiL LLLLLLLLLLLL00000 | 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] | 0000011001iiiiL LLLLLLLLLLLLRRRRR 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 | rrrrr11111RRRRR wwwww01011000000 | GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1] | 35 | 35 | 35 | | × | × | × | | |
| DIVH | reg1,reg2 | rrrrr000010RRRRR | GR[reg2]←GR[reg2]÷GR[reg1] ^{Note 6} | 35 | 35 | 35 | | × | × | × | | |
| | reg1,reg2,reg3 | rrrrr11111RRRRR wwwww01010000000 | GR[reg2]←GR[reg2]÷GR[reg1] ^{Note 6} GR[reg3]←GR[reg2]%GR[reg1] | 35 | 35 | 35 | | × | × | × | | |
| DIVHU | reg1,reg2,reg3 | rrrrr11111RRRRR wwwww01010000010 | GR[reg2]←GR[reg2]÷GR[reg1] ^{Note 6} GR[reg3]←GR[reg2]%GR[reg1] | 34 | 34 | 34 | | × | × | × | | |
| DIVU | reg1,reg2,reg3 | rrrrr11111RRRRR wwwww01011000010 | 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 | rrrrr11111100000 wwwww01101000100 | GR[reg3]←GR[reg2](15 : 0) GR[reg2] (31 : 16) | 1 | 1 | 1 | × | 0 | × | × | | |
| JARL | disp22,reg2 | rrrrr11110dddddd ddddddddddddddd0 Note 7 | GR[reg2]←PC+4 PC←PC+sign-extend(disp22) | 2 | 2 | 2 | | | | | | |
| JMP | [reg1] | 0000000011RRRRR | PC←GR[reg1] | 3 | 3 | 3 | | | | | | |
| JR | disp22 | 0000011110dddddd ddddddddddddddd0 Note 7 | PC←PC+sign-extend(disp22) | 2 | 2 | 2 | | | | | | |
| LD.B | disp16[reg1],reg2 | rrrrr111000RRRRR ddddddddddddddd | adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Byte)) | 1 | 1 | Note 11 | | | | | | |
| LD.BU | disp16[reg1],reg2 | rrrrr11110bRRRRR ddddddddddddddd1 Notes 8, 10 | adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Byte)) | 1 | 1 | Note 11 | | | | | | |

| Mnemonic | Operand | Opcode | Operation | Execution Clock | | | Flags | | | | | |
|----------|--------------------|--|--|------------------------|--------|---------|-------|----|---|---|-----|---|
| | | | | i | r | l | CY | OV | S | Z | SAT | |
| LD.H | disp16[reg1],reg2 | rrrrr111001RRRRR dddddddddddddd0 | adr←GR[reg1]+sign-extend(dispatch16) GR[reg2]←sign-extend(Load-memory(adr,Halfword)) | 1 | 1 | Note 11 | | | | | | |
| LDSR | reg2,regID | rrrrr111111RRRRR 0000000000100000 | SR[regID]←GR[reg2] | Other than regID = PSW | 1 | 1 | 1 | | | | | |
| | | | | regID = PSW | 1 | 1 | 1 | × | × | × | × | × |
| LD.HU | disp16[reg1],reg2 | rrrrr111111RRRRR dddddddddddddd1 | adr←GR[reg1]+sign-extend(dispatch16) GR[reg2]←zero-extend(Load-memory(adr,Halfword)) | 1 | 1 | Note 11 | | | | | | |
| LD.W | disp16[reg1],reg2 | rrrrr111001RRRRR dddddddddddddd1 | adr←GR[reg1]+sign-extend(dispatch16) GR[reg2]←Load-memory(adr,Word) | 1 | 1 | Note 11 | | | | | | |
| MOV | reg1,reg2 | rrrrr00000RRRRR | GR[reg2]←GR[reg1] | 1 | 1 | 1 | | | | | | |
| | imm5,reg2 | rrrrr010000iiii | GR[reg2]←sign-extend(imm5) | 1 | 1 | 1 | | | | | | |
| | imm32,reg1 | 00000110001RRRRR iiiiiiiiiiiiiiii llllllllllllllllll | GR[reg1]←imm32 | 2 | 2 | 2 | | | | | | |
| MOVEA | imm16,reg1,reg2 | rrrrr110001RRRRR iiiiiiiiiiiiiiii | GR[reg2]←GR[reg1]+sign-extend(imm16) | 1 | 1 | 1 | | | | | | |
| MOVHI | imm16,reg1,reg2 | rrrrr110010RRRRR iiiiiiiiiiiiiiii | 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] | 1 | 4 | 5 | | | | | | |
| | imm9,reg2,reg3 | rrrrr111111iiii wwwww01001lllll00 | 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 | rrrrr010111iiii | GR[reg2]←GR[reg2] ^{Note 6} xsign-extend(imm5) | 1 | 1 | 2 | | | | | | |
| MULHI | imm16,reg1,reg2 | rrrrr110111RRRRR iiiiiiiiiiiiiiii | 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] | 1 | 4 | 5 | | | | | | |
| | imm9,reg2,reg3 | rrrrr111111iiii wwwww01001lllll10 | 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 dddddddddddddd | adr←GR[reg1]+sign-extend(dispatch16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,Z flag) | Note 3 | Note 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) | Note 3 | Note 3 | Note 3 | | | | × | | |

APPENDIX C INSTRUCTION SET LIST

(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 iiiiiiiiiiiiiiii | GR[reg2]←GR[reg1]OR zero-extend(imm16) | 1 | 1 | 1 | | 0 | × | × | |
| PREPARE | list12,imm5 | 0000011110iiiiL LLLLLLLLLLLL00001 | 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} | 0000011110iiiiL LLLLLLLLLLLLff011 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 | | 000001111100000 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 | rrrrr11111RRRRR 0000000010100000 | GR[reg2]←GR[reg2]arithmetically shift right by GR[reg1] | 1 | 1 | 1 | × | 0 | × | × | |
| | imm5,reg2 | rrrrr010101iiii | GR[reg2]←GR[reg2]arithmetically shift right by zero-extend (imm5) | 1 | 1 | 1 | × | 0 | × | × | |
| SASF | cccc,reg2 | rrrrr111110cccc 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 | rrrrr010001iiii | 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 iiiiiiiiiiiiiiii | 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 | rrrrr111110cccc 0000000000000000 | If conditions are satisfied then GR[reg2]←00000001H else GR[reg2]←00000000H | 1 | 1 | 1 | | | | | |

APPENDIX C INSTRUCTION SET LIST

(5/6)

| Mnemonic | Operand | Opcode | Operation | Execution Clock | | | Flags | | | | | |
|----------|--------------------|---|--|-----------------|-------------|-------------|-------|----|---|---|-----|--|
| | | | | i | r | l | CY | OV | S | Z | SAT | |
| SET1 | bit#3,disp16[reg1] | 00bbb111110RRRRR dddddddddddddd | 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] | rrrrr11111RRRRR 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 | rrrrr11111RRRRR 0000000011000000 | GR[reg2]←GR[reg2] logically shift left by GR[reg1] | 1 | 1 | 1 | × | 0 | × | × | | |
| | imm5,reg2 | rrrrr010110iiii | GR[reg2]←GR[reg2] logically shift left by zero-extend(imm5) | 1 | 1 | 1 | × | 0 | × | × | | |
| SHR | reg1,reg2 | rrrrr11111RRRRR 0000000010000000 | GR[reg2]←GR[reg2] logically shift right by GR[reg1] | 1 | 1 | 1 | × | 0 | × | × | | |
| | imm5,reg2 | rrrrr010100iiii | GR[reg2]←GR[reg2] logically shift right by zero-extend(imm5) | 1 | 1 | 1 | × | 0 | × | × | | |
| SLD.B | disp7[ep],reg2 | rrrrr0110dddddd | 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 | rrrrr1000dddddd 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 | rrrrr1010dddddd0 Note 21 | adr←ep+zero-extend(disp8) GR[reg2]←Load-memory(adr,Word) | 1 | 1 | Note 9 | | | | | | |
| SST.B | reg2,disp7[ep] | rrrrr0111dddddd | adr←ep+zero-extend(disp7) Store-memory(adr,GR[reg2],Byte) | 1 | 1 | 1 | | | | | | |
| SST.H | reg2,disp8[ep] | rrrrr1001dddddd Note 19 | adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Halfword) | 1 | 1 | 1 | | | | | | |
| SST.W | reg2,disp8[ep] | rrrrr1010dddddd1 Note 21 | adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Word) | 1 | 1 | 1 | | | | | | |
| ST.B | reg2,disp16[reg1] | rrrrr111010RRRRR dddddddddddddd | 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 | rrrrr11111RRRRR 0000000010000000 | GR[reg2]←SR[regID] | 1 | 1 | 1 | | | | | | |

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. iiii: 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. ddddddd: Higher 6 bits of disp8.

APPENDIX D LIST OF CAUTIONS

This appendix lists cautions described in this document.

“Classification (hard/soft)” in table is as follows.

Hard: Cautions for microcontroller internal/external hardware

Soft: Cautions for software such as register settings or programs

(1/29)

| Chapter | Classification | Function | Details of Function | Cautions | Page |
|-----------|----------------|---------------|--|---|--------------------------------|
| Chapter 1 | Hard | Introduction | FLMD0 | Connect this pin to V _{ss} in the normal mode. | p. 20 <input type="checkbox"/> |
| | | | REGC | Connect the REGC pin to V _{ss} via a 4.7 μF (recommended value) capacitor. | p. 20 <input type="checkbox"/> |
| Chapter 2 | Soft | Pin functions | NMI | The NMI pin alternately functions as the P02 pin. It functions as the P02 pin after reset. To enable the NMI pin, set the PMC0.PMC02 bit to 1. The initial setting of the NMI pin is “No edge detected”. Select the NMI pin valid edge using INTF0 and INTRO registers. | p. 28 <input type="checkbox"/> |
| | | | FLMD0 | If noise that exceeds the noise elimination width is input to the RESET pin during self programming, the flash on-board mode may be entered depending on the capacitance charge end timing when a capacitor is connected to the FLMD0 pin. Therefore, do not connect a capacitor to the FLMD0 pin. | p. 38 <input type="checkbox"/> |
| | | | When power is turned on | Note that the following pin may temporarily output an undefined level, even during reset upon power application. P53/KR3/TIQ00/TOQ00/DDO pin | p. 40 <input type="checkbox"/> |
| Chapter 3 | Soft | CPU function | EIPC register EIPSW register FEPC register FEPSW register | Because only one set of these registers is available, the contents of these registers must be saved by program if multiple interrupts are enabled. | p. 44 <input type="checkbox"/> |
| | | | EIPC, FEPC | 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). | p. 44 <input type="checkbox"/> |
| | | | Program space | Because the 4 KB area of addresses 03FFF00H to 03FFFFFFH is an on-chip 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. | p. 52 <input type="checkbox"/> |
| | | | On-chip peripheral I/O area | 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. | p. 57 <input type="checkbox"/> |
| | | | | 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. | p. 57 <input type="checkbox"/> |
| | | | | Addresses not defined as registers are reserved for future expansion. The operation is undefined and not guaranteed when these addresses are accessed. | p. 57 <input type="checkbox"/> |
| | | | Internal RAM area | If a branch instruction is at the upper limit of the internal RAM area, a prefetch operation (invalid fetch) straddling the on-chip peripheral I/O area does not occur. | p. 58 <input type="checkbox"/> |

| Chapter | Chapter Classification | Function | Details of Function | Cautions | Page |
|---|--|--------------------------------|-----------------------------------|--|--------------------------------|
| Chapter 3 | Soft | CPU function | Setting data to special registers | When switching to the IDLE1, IDLE2, STOP, or sub-IDLE mode (PSC.STP bit = 1), five NOP instructions must be inserted immediately after switching is performed. | p. 69 <input type="checkbox"/> |
| | | | | 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 <2> and <3> above are performed by successive store instructions. If another instruction is placed between <2> and <3>, and if an interrupt is acknowledged by that instruction, the above sequence may not be established, causing malfunction. | p. 69 <input type="checkbox"/> |
| | | | | Although dummy data is written to the PRCMD register, use the same general-purpose register used to set the special register (<3> in Example) to write data to the PRCMD register (<2> in Example). The same applies when a general-purpose register is used for addressing. | p. 69 <input type="checkbox"/> |
| | | | SYS register | 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). | p. 71 <input type="checkbox"/> |
| | | | | 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. | p. 71 <input type="checkbox"/> |
| | | | Registers to be set first | Be sure to set the following registers first when using the V850ES/HF2. <ul style="list-style-type: none"> • System wait control register (VSWC) • On-chip debug mode register (OCDM) • Watchdog timer mode register 2 (WDTM2) | p. 72 <input type="checkbox"/> |
| | | | VSWC register | Three clocks are required to access an on-chip peripheral I/O register (without a wait cycle). The V850ES/HF2 requires wait cycles according to the operating frequency. Set the following value to the VSWC register in accordance with the frequency used. | p. 72 <input type="checkbox"/> |
| Accessing specific on-chip peripheral I/O registers | Accessing the above registers is prohibited in the following statuses. If a wait cycle is generated, it can only be cleared by a reset. <ul style="list-style-type: none"> • When the CPU operates with the subclock and the main clock oscillation is stopped • When the CPU operates with the internal oscillation clock | p. 73 <input type="checkbox"/> | | | |
| Chapter 4 | Hard | Port functions | Port function | Although a 1-bit memory manipulation instruction manipulates 1 bit, it accesses a port in 8-bit units. If a port has a mixture of input and output pins, therefore, the contents of the output latch of a pin set in the input mode become undefined, even if the pin is not subject to manipulation | p. 77 <input type="checkbox"/> |
| | Soft | | Port 0 | The NMI pin alternately functions as the P02 pin. It functions as the P02 pin after reset. To enable the NMI pin, set the PMC0.PMC02 bit to 1. The initial setting of the NMI pin is "No edge detected". Select the NMI pin valid edge using INTF0 and INTRO registers. | p. 79 <input type="checkbox"/> |
| | Hard, Soft | | | The alternate function of the P05 pin is the on-chip debug function. After external reset, the P05/INTP2/DRST pin is initialized as the on-chip debug pin (DRST). To use the P05 pin as a port pin, not as an on-chip debug pin, the following actions must be taken. <p><1> Clear the OCDM.OCDM0 bit (special register) to 0.</p> <p><2> Fix the P05/INTP2/DRST pin to the low level until the above action has been taken.</p> <p>When the on-chip debug function is not used, inputting a high level to the DRST pin before the above actions are taken may cause a malfunction (CPU deadlock). Exercise utmost care in handling the P05 pin.</p> <p>When a high level is not input to the P05/INTP2/DRST pin (when this pin is fixed to low level), it is not necessary to manipulate the OCDM.OCDM0 bit. Because a pull-down resistor (30 kΩ TYP.) is connected to the buffer of the P05/INTP2/DRST pin, the pin does not have to be fixed to the low level by an external source. The pull-down resistor is disconnected by clearing the OCDM0 bit to 0.</p> | p. 79 <input type="checkbox"/> |

| Chapter | Classification | Function | Details of Function | Cautions | Page | | |
|-----------|----------------|----------------|---------------------|---|---|--|--------------------------|
| Chapter 4 | Hard | Port functions | Port 0 | The P00 to P06 pins have hysteresis characteristics in the input mode of the alternate function, but do not have hysteresis characteristics in the port mode. | p. 80 | <input type="checkbox"/> | |
| | | | PMC0 register | The P05/INTP2/DRST pin functions as the DRST pin when the OCDM.OCDM0 bit is 1, regardless of the value of the PMC05 bit. | p. 81 | <input type="checkbox"/> | |
| | | | Port 3 | The P31 to P35 pins have hysteresis characteristics in the input mode of the alternate function, but do not have hysteresis characteristics in the port mode. | p. 83 | <input type="checkbox"/> | |
| | | | P3 register | To read or write bits 8 to 15 of the P3 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the P3H register. | p. 84 | <input type="checkbox"/> | |
| | Soft | | PM3 register | To read or write bits 8 to 15 of the PM3 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PM3H register. | p. 84 | <input type="checkbox"/> | |
| | | | PMC3L register | The INTP7 pin functions alternately as the RXDA0 pin. To use as the RXDA0 pin, invalidate the edge detection function of the alternate-function INTP7 pin (by fixing the INTF3.INTF31 and INTR3.INTR31 bits to 0). To use as the INTP7 pin, stop the reception operation of UARTA0 (by clearing the UA0CTL0.UA0RXE bit to 0). | p. 85 | <input type="checkbox"/> | |
| | Soft | | PU3 register | To read/write bits 8 to 15 of the PU3 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PU3H register. | p. 87 | <input type="checkbox"/> | |
| | | | Port 4 | The P40 and P42 pins have hysteresis characteristics in the input mode of the alternate function, but do not have hysteresis characteristics in the port mode. | p. 88 | <input type="checkbox"/> | |
| | | | Hard, Soft | Port 5 | <p>The DDI, DDO, DCK, and DMS pins are for the on-chip debug function. To use the DDI, DDO, DCK, and DMS pins as port pins, not as on-chip debug pins, the following actions must be taken.</p> <p><1> Clear the OCDM0 bit of the OCDM register (special register) to 0.</p> <p><2> Fix the P05/INTP2/DRST pin to the low level until the above action has been taken.</p> <p>When the on-chip debug function is not used, inputting a high level to the DRST pin before the above actions are taken may cause a malfunction (CPU deadlock). Exercise utmost care in handling the P05 pin.</p> <p>When a high level is not input to the P05/INTP2/DRST pin (when this pin is fixed to low level), it is not necessary to manipulate the OCDM.OCDM0 bit.</p> <p>Because a pull-down resistor (30 kΩ TYP.) is connected to the buffer of the P05/INTP2/DRST pin, the pin does not have to be fixed to the low level by an external source. The pull-down resistor is disconnected by clearing the OCDM0 bit to 0.</p> | p. 91 | <input type="checkbox"/> |
| | | | | Hard | Port 5 PMC5 register | <p>The P50 to P55 pins have hysteresis characteristics in the input mode of the alternate function, but do not have hysteresis characteristics in the port mode.</p> <p>If the control mode is specified by using the PMC5 register when the PFC5.PFC5n and PFCE5.PFCE5n bits are the default values (0), the output becomes undefined.</p> <p>For this reason, first set the PFC5.PFC5n and PFCE5.PFCE5n bits, and then set the PMC5n bit to 1 to set the control mode.</p> | p. 91 |
| | Soft | | Soft | Specification of port 5 alternate function | The KRn pin functions alternately as the TIQ0m pin. To use this pin as the TIQ0m pin, invalidate the key return detection function of the alternate-function KRn pin (by clearing the KRM.KRMn bit to 0). To use this pin as the KRn pin, invalidate the edge detection function of the alternate-function TIQ0m pin (n = 0 to 3, m = 0 to 3). | pp. 93, 94 | <input type="checkbox"/> |
| | | | | | | | |

| Chapter | Classification | Function | Details of Function | Cautions | Page |
|---|--|---------------------------------|-----------------------------------|---|---------------------------------|
| Chapter 4 | Soft | Port functions | P7H register, P7L register | Do not read the P7H and P7L registers during A/D conversion. | p. 98 <input type="checkbox"/> |
| | | | PM7H register, PM7L register | To use the alternate function of P7n (ANIn), set PM7n to 1. | p. 98 <input type="checkbox"/> |
| | Soft | Hard | Port 9 | The P90, P91, P96, P97, P99, and P913 to P915 pins have hysteresis characteristics in the input mode of the alternate function, but do not have hysteresis characteristics in the port mode. | p. 99 <input type="checkbox"/> |
| | | | P9 register | To read or write bits 8 to 15 of the P9 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the P9H register. | p. 100 <input type="checkbox"/> |
| | | | PM9 register | To read or write bits 8 to 15 of the PM9 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PM9H register. | p. 100 <input type="checkbox"/> |
| | | | PMC9 register | If the control mode is specified by using the PMC9 register when the PFC9.PFC9n bit and the PFCE9.PFCE9n bit are the default values (0), the output becomes undefined. For this reason, first set the PFC9.PFC9n bit and the PFCE9.PFCE9n bit to 1, and then set the PMC9n bit to 1 to set the control mode. | p. 101 <input type="checkbox"/> |
| | | | | To read or write bits 8 to 15 of the PMC9 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PMC9H register. | p. 101 <input type="checkbox"/> |
| | | | PFC9 register | To read or write bits 8 to 15 of the PFC9 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PFC9H register. | p. 102 <input type="checkbox"/> |
| | | | PFCE9 register | To read or write bits 8 to 15 of the PFCE9 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PFCE9H register. | p. 103 <input type="checkbox"/> |
| | | | Setting of control mode of P9 pin | If the control mode is specified by using the PMC9 register when the PFC9.PFC9n and PFCE9.PFCE9n bits are the default values (0), the output becomes undefined. For this reason, first set the PFC9.PFC9n and PFCE9.PFCE9n bits, and then set the PMC9n bit to 1 to set the control mode. | p. 104 <input type="checkbox"/> |
| | | | | The KR7 pin and RXDA1 pin are alternate-function pins. When using the pin as the RXDA1 pin, disable KR7 pin key return detection. (Clear the KRM7 bit of the KRM register to 0.) Also, when using the pin as the KR7 pin, it is recommended to set the PFC91 bit to 1 and clear the PFCE91 bit to 0. | p. 105 <input type="checkbox"/> |
| | | | PU9 register | To read/write bits 8 to 15 of the PU9 register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PU9H register. | p. 106 <input type="checkbox"/> |
| | | | PMCCM register | Be sure to set bits 7 to 2 and 0 to "0". | p. 108 <input type="checkbox"/> |
| | | | Port DL | Because the FLMD1 pin is used in the flash programming mode, it does not have to be manipulated by using a port control register. For details, see CHAPTER 22 FLASH MEMORY. | p. 113 <input type="checkbox"/> |
| | | | PDL register | To read or write bits 8 to 15 of the PDL register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PDLH register. | p. 114 <input type="checkbox"/> |
| | | | PMDL register | To read or write bits 8 to 15 of the PMDL register in 8-bit or 1-bit units, specify these bits as bits 0 to 7 of the PMDLH register. | p. 114 <input type="checkbox"/> |
| Register settings to use port pins as alternate-function pins | After an external reset, the P05/INTP2/DRST pin is initialized as an on-chip debug pin (DRST). To not use the P05/INTP2/DRST pin as an on-chip debug pin, see CHAPTER 24 ON-CHIP DEBUG FUNCTION. | p. 116 <input type="checkbox"/> | | | |

| Chapter | Classification | Function | Details of Function | Cautions | Page | |
|-----------|----------------|----------------|---|--|--|---------------------------------|
| Chapter 4 | Soft | Port functions | Register settings to use port pins as alternate-function pins | <p>The INTP7 pin functions alternately as the RXDA0 pin. To use this pin as the RXDA0 pin, invalidate the edge detection function of the alternate-function INTP7 pin (by clearing the INTF3.INTF31 bit to 0 and the INTR3.INTR31 bit to 0). To use this pin as the INTP7 pin, stop the reception operation of UARTA0 (by clearing the UA0CTL0.UA0RXE bit to 0).</p> | p. 116 <input type="checkbox"/> | |
| | | | | <p>The KRn pin functions alternately as the TIQ0m pin. To use this pin as the TIQ0m pin, invalidate the key return detection function of the alternate-function KRn pin (by clearing the KRMn bit of the KRM register to 0). To use this pin as the KRn pin, invalidate the edge detection function of the alternate-function TIQ0m pin.</p> | p. 117 <input type="checkbox"/> | |
| | | | | <p>The DDI, DDO, DCK, and DMS pins are on-chip debug pins. To not use these pins as on-chip debug pins after an external reset, see CHAPTER 24 ON-CHIP DEBUG FUNCTION.</p> | p. 117 <input type="checkbox"/> | |
| | | | | <p>If the control mode is specified by using the PMC5 register when the PFC5.PFC5n bit and the PFCE5.PFCE5n bit are the default values (0), the output becomes undefined. For this reason, first set the PFC5.PFC5n bit and the PFCE5.PFCE5n bit, and then set the PMC5n bit to 1 to set the control mode.</p> | p. 117 <input type="checkbox"/> | |
| | | | | <p>Set PM7n to 1 to use the alternate function of P7n (ANIn).</p> | p. 118 <input type="checkbox"/> | |
| | | | | <p>The KR7 pin and RXDA1 pin are alternate-function pins. When using the pin as the RXDA1 pin, disable KR7 pin key return detection. (Clear the KRM.KRM7 bit to 0.) Also, when using the pin as the KR7 pin, it is recommended to set the PFC91 bit to 1 and clear the PFCE91 bit to 0.</p> | p. 118 <input type="checkbox"/> | |
| | | | | <p>If the control mode is specified by using the PMC9 register when the PFC9.PFC9n bit and the PFCE9.PFCE9n bit are the default values (0), the output becomes undefined. For this reason, first set the PFC9.PFC9n bit and the PFCE9.PFCE9n bit, and then set the PMC9n bit to 1 to set the control mode.</p> | p. 118 <input type="checkbox"/> | |
| | | | | <p>The FLMD1 pin does not have to be manipulated by using a port control register because it is used in the flash programming mode. For details, see CHAPTER 22 FLASH MEMORY.</p> | p. 119 <input type="checkbox"/> | |
| | | | | <p>Switching from port mode to alternate-function mode</p> | <p>To switch from the port mode to alternate-function mode in the following order.</p> <p><1> Set the PFCn and PFCEn registers: Alternate-function selection</p> <p><2> Set the corresponding bit of the PMc register to 1: Switch to alternate-function mode</p> <p>If the PMc register is set first, note with caution that, at that moment or depending on the change of the pin states in accordance with the setting of the PFCn, and PFCEn registers, unexpected operations may occur.</p> | p. 145 <input type="checkbox"/> |
| | | | | | <p>IRegardless of the port mode/alternate-function mode, the Pn register is read and written as follows.</p> <ul style="list-style-type: none"> • Pn register read: Read the port output latch value (when PMn.PMnm bit = 0), or read the pin states (PMn.PMnm bit = 1). • Pn register write: Write to the port output latch | p. 145 <input type="checkbox"/> |

| Chapter | Chapter Classification | Function | Details of Function | Cautions | Page |
|---------------|--|---------------------------------|--|---|---------------------------------|
| Chapter 4 | Soft | Port functions | Caution on alternate-function mode (input) | <p>The input signal to the alternate-function block is low level when the PMCn.PMCnm bit is 0 due to the AND output of the PMCn register set value and the pin level. Thus, depending on the port setting and alternate-function operation enable timing, unexpected operations may occur. Therefore, switch between the port mode and alternate-function mode in the following sequence.</p> <ul style="list-style-type: none"> To switch from port mode to alternate-function mode (input) Set the pins to the alternate-function mode using the PMCn register and then enable the alternate-function operation. To switch from alternate-function mode (input) to port mode Stop the alternate-function operation and then switch the pins to the port mode. | p. 145 <input type="checkbox"/> |
| Chapter 5 | Soft | Clock generation function | Clock generator | The internal oscillation clock is selected when watchdog timer 2 overflows during the oscillation stabilization time. | p. 147 <input type="checkbox"/> |
| | | | PCC register | Do not change the CPU clock (by using the CK3 to CK0 bits) while CLKOUT is being output. | p. 150 <input type="checkbox"/> |
| | | | | 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. | p. 150 <input type="checkbox"/> |
| | | | | When stopping the main clock, stop the PLL. Also stop the operations of the on-chip peripheral functions operating with the main clock. | p. 151 <input type="checkbox"/> |
| | | | | 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. Internal system clock (fCLK) > Subclock (fXT) × 4 | p. 151 <input type="checkbox"/> |
| | | | | Enable operation of the on-chip peripheral functions operating with the main clock only after the oscillation of the main clock stabilizes. If their operations are enabled before the lapse of the oscillation stabilization time, a malfunction may occur. | p. 152 <input type="checkbox"/> |
| | | | RCM register | The settings of the RCM register are valid by setting the option byte. For details, see CHAPTER 23 OPTION BYTE FUNCTION. | p. 153 <input type="checkbox"/> |
| | | | | The internal oscillator cannot be stopped while the CPU is operating on the internal oscillation clock (CCLS.CCLS bit = 1). Do not set the RSTOP bit to 1. | p. 153 <input type="checkbox"/> |
| | | | | The internal oscillator oscillates if the CCLS.CCLS bit is set to 1 (when WDT overflow occurs during oscillation stabilization) even when the RSTOP bit is set to 1. At this time, the RSTOP bit remains being set to 1. | p. 153 <input type="checkbox"/> |
| | | | CCLS register | If WDT overflow occurs during oscillation stabilization after a reset is released, the CCLS bit is set to 1 and the reset value is 01H. | p. 153 <input type="checkbox"/> |
| | | | PLLCTL register | When the PLLON bit is cleared to 0, the SELPLL bit is automatically cleared to 0 (clock-through mode). | p. 155 <input type="checkbox"/> |
| | | | | 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. | p. 155 <input type="checkbox"/> |
| | | | LOCKR register | The LOCK register does not reflect the lock status of the PLL in real time. | p. 156 <input type="checkbox"/> |
| | | | PLLS register | Set so that the lockup time is 800 μs or longer. | p. 157 <input type="checkbox"/> |
| | | | | Do not change the PLLS register setting during the lockup period. | p. 157 <input type="checkbox"/> |
| PLCM register | Set the port-related control registers (PM, PMC, PFC, and PFCE registers, etc.) first, and then set the PCLE bit to 1. | p. 158 <input type="checkbox"/> | | | |
| | Set the PCLE bit to 1 only during PLL operation. To stop the PLL, clear the PCLE bit to 0. | p. 158 <input type="checkbox"/> | | | |

| Chapter | Classification | Function | Details of Function | Cautions | Page |
|-----------|----------------|---|---------------------|--|---------------------------------|
| Chapter 6 | Soft | 16-bit timer/ event counter P (TMP) | TPnCTL0 register | 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. | p. 164 <input type="checkbox"/> |
| | | | | Be sure to clear bits 3 to 6 to "0". | p. 164 <input type="checkbox"/> |
| | | | TPnCTL1 register | Be sure to clear the TP0SYE and TP2SYE bits to 0. | p. 165 <input type="checkbox"/> |
| | | | | 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. | p. 165 <input type="checkbox"/> |
| | | | | Be sure to clear bits 3 and 4 to "0". | p. 165 <input type="checkbox"/> |
| | | | | External event count input is selected in the external event count mode regardless of the value of the TPnEEE bit. | p. 166 <input type="checkbox"/> |
| | | | | 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. | p. 166 <input type="checkbox"/> |
| | | | TP0nIOC0 register | 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. | p. 167 <input type="checkbox"/> |
| | | | | Even if the TPnOLm bit is manipulated when the TPnCE and TPnOEm bits are 0, the TOPnm pin output level varies. | p. 167 <input type="checkbox"/> |
| | | | TPnIOC1 register | 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. | p. 168 <input type="checkbox"/> |
| | | | | 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. | p. 168 <input type="checkbox"/> |
| | | | TPnIOC2 register | 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. | p. 169 <input type="checkbox"/> |
| | | | | 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. | p. 169 <input type="checkbox"/> |
| | | | | The TPnETS1 and TPnETS0 bits are valid only when the external trigger pulse output mode (TPnCTL1.TPnMD2 to TPnCTL1.TPnMD0 bits = 010) or the one-shot pulse output mode (TPnCTL1.TPnMD2 to TPnCTL1.TPnMD0 = 011) is set. | p. 169 <input type="checkbox"/> |
| | | | TPnOPT0 register | 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. | p. 170 <input type="checkbox"/> |
| | | | | Be sure to clear bits 1 to 3, 6, and 7 to "0". | p. 170 <input type="checkbox"/> |
| | | | TPnCCR0 register | Accessing the TPnCCR0 register is prohibited in the following statuses. For details, see 3.4.8 (2) Accessing specific on-chip peripheral I/O registers. <ul style="list-style-type: none"> When the CPU operates with the subclock and the main clock oscillation is stopped When the CPU operates with the internal oscillation clock | p. 171 <input type="checkbox"/> |

| Chapter | Chapter Classification | Function | Details of Function | Cautions | Page |
|---|--|------------------------------------|---|--|---------------------------------|
| Chapter 6 | Soft | 16-bit timer/event counter P (TMP) | TPnCCR1 register | Accessing the TPnCCR1 register is prohibited in the following statuses. For details, see 3.4.8 (2) Accessing specific on-chip peripheral I/O registers. <ul style="list-style-type: none"> • When the CPU operates with the subclock and the main clock oscillation is stopped • When the CPU operates with the internal oscillation clock | p. 173 <input type="checkbox"/> |
| | | | TPnCNT register | Accessing the TPnCNT register is prohibited in the following statuses. For details, see 3.4.8 (2) Accessing specific on-chip peripheral I/O registers. <ul style="list-style-type: none"> • When the CPU operates with the subclock and the main clock oscillation is stopped • When the CPU operates with the internal oscillation clock | p. 175 <input type="checkbox"/> |
| | | | PnmNFC register | Be sure to clear bits 3 to 5 and 7 to "0". | p. 176 <input type="checkbox"/> |
| | | | | A signal input to the timer input pin (TIPnm) before the PnmNFC register is set is output with digital noise eliminated. Therefore, set the sampling clock (NFC2 to NFC0) and the number of times of sampling (NFSTS) by using the PnmNFC register, wait for initialization time = (Sampling clock) × (Number of times of sampling), and enable the timer operation. | p. 176 <input type="checkbox"/> |
| | | | Operation | 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"). | p. 177 <input type="checkbox"/> |
| | | | | 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). | p. 177 <input type="checkbox"/> |
| | | | Interval timer mode (TPnMD2 to TPnMD0 bits = 000) | This bit can be set to 1 only when the interrupt request signals (INTTPnCC0 and INTTPnCC1) are masked by the interrupt mask flags (TPnCCMK0 and TPnCCMK1) and timer output (TOPn1) is performed at the same time. However, set the TPnCCR0 and TPnCCR1 registers to the same value (see 6.5.1 (2) (d) Operation of TPnCCR1 register). | p. 179 <input type="checkbox"/> |
| | | | 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. | p. 184 <input type="checkbox"/> |
| | | | Register setting for operation in external event count mode | When an external clock is used as the count clock, the external clock can be input only from the TIPn0 pin. At this time, set the TPnIOC1.TPnIS1 and TPnIOC1.TPnIS0 bits to 00 (capture trigger input (TIPn0 pin): no edge detection). | p. 190 <input type="checkbox"/> |
| | | | Operation timing in external event count mode | In the external event count mode, do not set the TPnCCR0 register to 0000H. | p. 192 <input type="checkbox"/> |
| | | | | In the external event count mode, use of the timer output is disabled. If performing timer output using external event count input, set the interval timer mode, and select the operation enabled by the external event count input for the count clock (TPnCTL1.TPnMD2 to TPnCTL1.TPnMD0 bits = 000, TPnCTL1.TPnEEE bit = 1). | p. 192 <input type="checkbox"/> |
| 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. | p. 193 <input type="checkbox"/> | | | |
| TPnIOC0, TPnCE0, TPnOL0 bits | Clear this bit to 0 when the TOPn0 pin is not used in the external trigger pulse output mode. | p. 198 <input type="checkbox"/> | | | |

| Chapter | Classification | Function | Details of Function | Cautions | Page |
|-----------|----------------|---|--|--|---------------------------------|
| Chapter 6 | Soft | 16-bit timer/ event counter P (TMP) | 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. | p. 202 <input type="checkbox"/> |
| | | | TPnIOC0, TPnOE0, TPnOLO bits | Clear this bit to 0 when the TOPn0 pin is not used in the one-shot pulse output mode. | p. 210 <input type="checkbox"/> |
| | | | Setting of registers in one-shot pulse output mode | One-shot pulses are not output even in the one-shot pulse output mode, if the value set in the TPnCCR1 register is greater than that set in the TPnCCR0 register. | p. 211 <input type="checkbox"/> |
| | | | 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. | p. 213 <input type="checkbox"/> |
| | | | TPnIOC0, TPnOE0, TPnOLO bits | Clear this bit to 0 when the TOPn0 pin is not used in the PWM output mode. | p. 217 <input type="checkbox"/> |
| | | | Timer tuned operation function | The tuned operation mode is enabled or disabled by the TPmCTL1.TPmSYE and TQ0CTL1.TQ0SYE bits. For TMP2, either or both TMP3 and TMQ0 can be specified as slaves. | p. 248 <input type="checkbox"/> |
| | | | | Set the tuned operation mode using the following procedure. <1> Set the TPmCTL1.TPmSYE and TQ0CTL1.TQ0SYE bits of the slave timer to enable the tuned operation. Set the TPmCTL1.TPmMD2 to TPmCTL1.TPmMD0 and TQ0CTL1.TQ0MD2 to TQ0CTL1.TQ0MD0 bits of the slave timer to the free-running mode. <2> Set the timer mode by using the TPnCTL1.TPnMD2 to TPnCTL1.TPnMD0 bits. At this time, do not set the TPnCTL1.TPnSYE bit of the master timer. <3> Set the compare register value of the master and slave timers. <4> Set the TPmCTL0.TPmCE and TQ0CTL0.TQ0CE bits of the slave timer to enable operation on the internal operating clock. <5> Set the TPnCTL0.TPnCE bit of the master timer to enable operation on the internal operating clock. | p. 248 <input type="checkbox"/> |
| | | | Selector function | When using the selector function, set the capture trigger input of TMP before connecting the timer. | p. 252 <input type="checkbox"/> |
| | | | | When setting the selector function, first disable the peripheral I/O to be connected (TMP, TMM0, or UARTA). | p. 252 <input type="checkbox"/> |
| | | | SELCNT0 register | Use the INTTM0EQ0 interrupt signal as the TIP01 input signal under the following condition. TMM0 operation clock \geq TMP0 operation clock \times 4 | p. 253 <input type="checkbox"/> |
| | | | | To set the ISEL02 to ISEL04 bits to 1, set the corresponding pin in the capture input mode. | p. 253 <input type="checkbox"/> |
| | | | | Set the ISEL02 to ISEL06 bits when operation of the target (TMP0, TMP1, TMM0, UARTA0, or UARTA1) is stopped. | p. 253 <input type="checkbox"/> |
| | | | Capture operation | When the capture operation is used and a slow clock is selected as the count clock, FFFFH, not 0000H, may be captured in the TPnCCR0 and TPnCCR1 registers if the capture trigger is input immediately after the TPnCE bit is set to 1. | p. 254 <input type="checkbox"/> |

| Chapter | Classification | Function | Details of Function | Cautions | Page |
|-----------|----------------|---|---------------------|---|---------------------------------|
| Chapter 7 | Soft | 16-bit timer/ event counter Q (TMQ) | TQ0CTL0 register | 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. | p. 260 <input type="checkbox"/> |
| | | | | Be sure to clear bits 3 to 6 to "0". | p. 260 <input type="checkbox"/> |
| | | | TQ0CTL1 register | 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. | p. 261 <input type="checkbox"/> |
| | | | | Be sure to clear bits 3 and 4 to "0". | p. 261 <input type="checkbox"/> |
| | | | | External event count input is selected in the external event count mode regardless of the value of the TQ0EEE bit. | p. 262 <input type="checkbox"/> |
| | | | | 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. | p. 262 <input type="checkbox"/> |
| | | | TQ0IOC0 register | 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. | p. 263 <input type="checkbox"/> |
| | | | | Even if the TQ0OLm bit is manipulated when the TQ0CE and TQ0OEm bits are 0, the TQ0m pin output level varies. | p. 263 <input type="checkbox"/> |
| | | | TQ0IOC1 register | 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. | p. 264 <input type="checkbox"/> |
| | | | | 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. | p. 264 <input type="checkbox"/> |
| | | | TQ0IOC2 register | 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. | p. 265 <input type="checkbox"/> |
| | | | | 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. | p. 265 <input type="checkbox"/> |
| | | | | The TQ0ETS1 and TQ0ETS0 bits are valid only when the external trigger pulse output mode (TQ0CTL1.TQ0MD2 to TQ0CTL1.TQ0MD0 bits = 010) or the one-shot pulse output mode (TQ0CTL1.TQ0MD2 to TQ0CTL1.TQ0MD0 = 011) is set. | p. 265 <input type="checkbox"/> |
| | | | TQ0OPT0 register | Rewrite the TQ0CCS3 to TQ0CCS0 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. | p. 266 <input type="checkbox"/> |
| | | | | Be sure to clear bits 1 to 3 to "0". | p. 266 <input type="checkbox"/> |
| | | | TQ0CCR0 register | Accessing the TQ0CCR0 register is prohibited in the following statuses. For details, refer to 3.4.8 (2) Accessing specific on-chip peripheral I/O registers. <ul style="list-style-type: none"> When the CPU operates with the subclock and the main clock oscillation is stopped When the CPU operates with the internal oscillation clock | p. 267 <input type="checkbox"/> |
| | | | TQ0CCR1 register | Accessing the TQ0CCR1 register is prohibited in the following statuses. For details, refer to 3.4.8 (2) Accessing specific on-chip peripheral I/O registers. <ul style="list-style-type: none"> When the CPU operates with the subclock and the main clock oscillation is stopped When the CPU operates with the internal oscillation clock | p. 269 <input type="checkbox"/> |

| Chapter | Classification | Function | Details of Function | Cautions | Page |
|---|--|---|--|--|---------------------------------------|
| Chapter 7 | Soft | 16-bit timer/ event counter Q (TMQ) | TQ0CCR2 register | Accessing the TQ0CCR2 register is prohibited in the following statuses. For details, refer to 3.4.8 (2) Accessing specific on-chip peripheral I/O registers. <ul style="list-style-type: none"> • When the CPU operates with the subclock and the main clock oscillation is stopped • When the CPU operates with the internal oscillation clock | p. 271 <input type="checkbox"/> |
| | | | TQ0CCR3 register | Accessing the TQ0CCR3 register is prohibited in the following statuses. For details, refer to 3.4.8 (2) Accessing specific on-chip peripheral I/O registers. <ul style="list-style-type: none"> • When the CPU operates with the subclock and the main clock oscillation is stopped • When the CPU operates with the internal oscillation clock | p. 273 <input type="checkbox"/> |
| | | | TQ0CNT register | Accessing the TQ0CNT register is prohibited in the following statuses. For details, refer to 3.4.8 (2) Accessing specific on-chip peripheral I/O registers. <ul style="list-style-type: none"> • When the CPU operates with the subclock and the main clock oscillation is stopped • When the CPU operates with the internal oscillation clock | p. 275 <input type="checkbox"/> |
| | | | Q0mNFC register | Be sure to clear bits 3 to 5 and 7 to "0". | p. 276 <input type="checkbox"/> |
| | | | | A signal input to the timer input pin (TIQ0m) before the Q0mNFC register is set is output with digital noise eliminated. Therefore, set the sampling clock (NFC2 to NFC0) and the number of times of sampling (NFSTS) by using the Q0mNFC register, wait for initialization time = (Sampling clock) × (Number of times of sampling), and enable the timer operation. | p. 276 <input type="checkbox"/> |
| | | | External event count mode | 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"). | p. 277 <input type="checkbox"/> |
| | | | External trigger pulse output mode, one-shot pulse output mode, and pulse width measurement mode | 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). | p. 277 <input type="checkbox"/> |
| | | | TQ0CTL1.TQ0EEE bit | This bit can be set to 1 only when the interrupt request signals (INTTQ0CC0 and INTTQ0CCk) are masked by the interrupt mask flags (TQ0CCMK0 to TQ0CCMKk) and the timer output (TOQ0k) is performed at the same time. However, the TQ0CCR0 and TQ0CCRk registers must be set to the same value (refer to 7.5.1 (2) (d) Operation of TQ0CCR1 to TQ0CCR3 registers) (k = 1 to 3). | p. 279 <input type="checkbox"/> |
| | | | 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. | pp. 283, 292 <input type="checkbox"/> |
| Register setting for operation in external event count mode | When an external clock is used as the count clock, the external clock can be input only from the TIQ00 pin. At this time, set the TQ0IOC1.TQ0IS1 and TQ0IOC1.TQ0IS0 bits to 00 (capture trigger input (TIQ00 pin): no edge detection). | p. 289 <input type="checkbox"/> | | | |

| Chapter | Chapter Classification | Function | Details of Function | Cautions | Page |
|--|--|------------------------------------|---|--|---------------------------------|
| Chapter 7 | Soft | 16-bit timer/event counter Q (TMQ) | Operation timing in external event count mode | In the external event count mode, do not set the TQ0CCR0 register to 0000H. | p. 291 <input type="checkbox"/> |
| | | | | In the external event count mode, use of the timer output is disabled. If performing timer output using external event count input, set the interval timer mode, and select the operation enabled by the external event count input for the count clock (TQ0CTL1.TQ0MD2 to TQ0CTL1.TQ0MD0 bits = 000, TQ0CTL1.TQ0EEE bit = 1). | p. 291 <input type="checkbox"/> |
| | | | TQ0IOC0.TQ0OE0, TQ0OLO bits | Clear this bit to 0 when the TOQ00 pin is not used in the external trigger pulse output mode. | p. 299 <input type="checkbox"/> |
| | | | 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. | p. 303 <input type="checkbox"/> |
| | | | TQ0IOC0.TQ0OE0, TQ0OLO bits | Clear this bit to 0 when the TOQ00 pin is not used in the one-shot pulse output mode. | p. 312 <input type="checkbox"/> |
| | | | Register setting in one-shot pulse output mode | One-shot pulses are not output even in the one-shot pulse output mode, if the value set in the TQ0CCRk register is greater than that set in the TQ0CCR0 register. | p. 313 <input type="checkbox"/> |
| | | | 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. | p. 316 <input type="checkbox"/> |
| | | | TQ0IOC0.TQ0OE0, TQ0OLO bit | Clear this bit to 0 when the TOQ00 pin is not used in the PWM output mode. | p. 321 <input type="checkbox"/> |
| | | | Triangular wave PWM mode (TQ0MD2 to TQ0MD0 = 111) | In the PWM mode, the capture function of the TQ0CCRm register cannot be used because this register can be used only as a compare register. | p. 355 <input type="checkbox"/> |
| | | | Timer tuned operation function | The tuned operation mode is enabled or disabled by the TPmCTL1.TPmSYE and TQ0CTL1.TQ0SYE bits. For TMQ2, either or both TMQ3 and TMQ0 can be specified as slaves. | p. 357 <input type="checkbox"/> |
| Set the tuned operation mode using the following procedure. <1> Set the TPmCTL1.TPmSYE and TQ0CTL1.TQ0SYE bits of the slave timer to enable the tuned operation. Set the TPmCTL1.TPmMD2 to TPmCTL1.TPmMD0 and TQ0CTL1.TQ0MD2 to TQ0CTL1.TQ0MD0 bits of the slave timer to the free-running mode. <2> Set the timer mode by using the TPnCTL1.TPnMD2 to TPnCTL1.TPnMD0 bits. At this time, do not set the TPnCTL1.TPnSYE bit of the master timer. <3> Set the compare register value of the master and slave timers. <4> Set the TPmCTL0.TPmCE and TQ0CTL0.TQ0CE bits of the slave timer to enable operation on the internal operating clock. <5> Set the TPnCTL0.TPnCE bit of the master timer to enable operation on the internal operating clock. | p. 357 <input type="checkbox"/> | | | | |
| Capture operation | When the capture operation is used and a slow clock is selected as the count clock, FFFFH, not 0000H, may be captured in the TQ0CCR0, TQ0CCR1, TQ0CCR2, and TQ0CCR3 registers if the capture trigger is input immediately after the TQ0CE bit is set to 1. | p. 361 <input type="checkbox"/> | | | |

| Chapter | Classification | Function | Details of Function | Cautions | Page |
|--|----------------|---|---|--|--|
| Chapter 8 | Soft | 16-bit interval timer M (TMM) | TM0CTL0 register | Set the TMOCKS2 to TMOCKS0 bits when TM0CE bit = 0. When changing the value of TM0CE from 0 to 1, it is not possible to set the value of the TMOCKS2 to TMOCKS0 bits simultaneously. | p. 364 <input type="checkbox"/> |
| | | | | Be sure to clear bits 3 to 6 to "0". | p. 364 <input type="checkbox"/> |
| | | | Interval timer mode | Do not set the TM0CMP0 register to FFFFH. | pp. 365, 368 <input type="checkbox"/> |
| | | | Start counting | 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. | p. 369 <input type="checkbox"/> |
| | | | TM0CMP0, TM0CTL0 registers | 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. | p. 369 <input type="checkbox"/> |
| Chapter 9 | Soft | Watch timer functions | PRSM0 register | Do not change the values of the BGCS00 and BGCS01 bits during watch timer operation. | p. 373 <input type="checkbox"/> |
| | | | | Set the PRSM0 register before setting the BGCE0 bit to 1. | p. 373 <input type="checkbox"/> |
| | | | | 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. | p. 373 <input type="checkbox"/> |
| | | | PRSCM0 register | Do not rewrite the PRSCM0 register during watch timer operation. | p. 374 <input type="checkbox"/> |
| | | | | Set the PRSCM0 register before setting the PRSM0.BGCE0 bit to 1. | p. 374 <input type="checkbox"/> |
| | | | | 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. | p. 374 <input type="checkbox"/> |
| | | | WTM register | Rewrite the WTM2 to WTM7 bits while both the WTM0 and WTM1 bits are 0. | p. 376 <input type="checkbox"/> |
| | 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). | p. 379 <input type="checkbox"/> | | |
| | Hard | | 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. | p. 379 <input type="checkbox"/> | |
| | Chapter 10 | Soft | Functions of Watchdog Timer 2 | Functions | 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: $f_{\text{H}}/2^{19}$) do not need to be changed. |
| For the non-maskable interrupt servicing due to a non-maskable interrupt request signal (INTWDT2), see 14.2.2 (2) From INTWDT2 signal. | | | | | p. 380 <input type="checkbox"/> |
| WDTM2 register | | | | Accessing the WDTM2 register is prohibited in the following statuses. For details, see 3.4.8 (2) Accessing specific on-chip peripheral I/O registers. <ul style="list-style-type: none"> When the CPU operates with the subclock and the main clock oscillation is stopped When the CPU operates with the internal oscillation clock | p. 382 <input type="checkbox"/> |
| | | | | If the OPB1 bit is set to 1 by using the option byte function (see CHAPTER 23), the reset mode is fixed. | p. 382 <input type="checkbox"/> |
| | | | | For details of the WDCS20 to WDCS24 bits, see Table 10-2 Watchdog Timer 2 Clock Selection. | p. 382 <input type="checkbox"/> |

| Chapter | Chapter Classification | Function | Details of Function | Cautions | Page |
|------------|---|---------------------------------|--|---|---------------------------------|
| Chapter 10 | Soft | Functions of watchdog timer 2 | WDTM2 register | If the WDTM2 register is rewritten twice after reset, an overflow signal is forcibly generated and the counter is reset. | p. 382 <input type="checkbox"/> |
| | | | | To intentionally generate an overflow signal, write to the WDTM2 register only twice or write a value other than ACH to the WDTE register once. However, when watchdog timer 2 is set to stop operation, an overflow signal is not generated even if data is written to the WDTM2 register only twice, or a value other than "ACH" is written to the WDTE register only once. | p. 382 <input type="checkbox"/> |
| | | | | To stop the operation of watchdog timer 2, write 1FH to the WDTM2 register. If the OPB1 bit is set to 1 by using the option byte function (see CHAPTER 23), however, watchdog timer 2 cannot be stopped by any means other than reset. | p. 382 <input type="checkbox"/> |
| | | | | If the OPB1 bit is set to 1 by using the option byte function, the clock is fixed to the internal oscillation clock (f_R) ($2^{12}/f_R$ to $2^{19}/f_R$ can be selected). For details, see CHAPTER 23 OPTION BYTE FUNCTION. | p. 383 <input type="checkbox"/> |
| | | | WDTE register | When a value other than "ACH" is written to the WDTE register, an overflow signal is forcibly output. | p. 384 <input type="checkbox"/> |
| | | | | When a 1-bit memory manipulation instruction is executed for the WDTE register, an overflow signal is forcibly output. | p. 384 <input type="checkbox"/> |
| | | | | To intentionally generate an overflow signal, write to the WDTM2 register only twice or write a value other than ACH to the WDTE register once. However, when the watchdog timer 2 is set to stop operation, an overflow signal is not generated even if data is written to the WDTM2 register only twice, or a value other than "ACH" is written to the WDTE register only once. | p. 384 <input type="checkbox"/> |
| | | | The read value of the WDTE register is "9AH" (which differs from written value "ACH"). | p. 384 <input type="checkbox"/> | |
| Chapter 11 | Hard | A/D converter | ANI0 to ANI11 pins | 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. | p. 389 <input type="checkbox"/> |
| | | | | The analog input pins (ANI0 to ANI11) function alternately as input port pins (P70 to P711). If any of ANI0 to ANI11 is selected to execute A/D conversion, do not execute an input instruction to port 7 during conversion. If executed, the conversion resolution may be degraded. | p. 389 <input type="checkbox"/> |
| | Soft | ADA0M0 register | Accessing the ADA0M0 register is prohibited in the following statuses. For details, see 3.4.8 (2) Accessing specific on-chip peripheral I/O registers. | p. 390 <input type="checkbox"/> | |
| | | | <ul style="list-style-type: none"> When the CPU operates with the subclock and the main clock oscillation is stopped When the CPU operates with the internal oscillation clock | | |
| | | | Write operations to bit 0 are ignored. | p. 391 <input type="checkbox"/> | |
| | Changing the ADA0M1 register value is prohibited while A/D conversion is enabled (ADA0CE bit = 1). | p. 391 <input type="checkbox"/> | | | |
| | If the ADA0M0, ADA0M2, ADA0S, ADA0PFM, and ADA0PFT registers are written during A/D conversion (ADA0EF bit = 1), the following will be performed according to the mode. | p. 391 <input type="checkbox"/> | | | |
| | <ul style="list-style-type: none"> In software trigger mode A/D conversion is stopped and started again from the beginning. In hardware trigger mode A/D conversion is stopped, and the trigger standby state is set. | | | | |

| Chapter | Classification | Function | Details of Function | Cautions | Page |
|------------|----------------|---------------|-----------------------------------|---|---------------------------------|
| Chapter 11 | Soft | A/D converter | ADA0M0 register | When not using the A/D converter, stop the operation by setting the ADA0CE bit to 0 to reduce the power consumption. | p. 391 <input type="checkbox"/> |
| | | | | The resolution for the first conversion of the data of the input pin immediately after the start of A/D conversion may be degraded. For details, see 11.6 (7) AV _{REF0} pin. | p. 391 <input type="checkbox"/> |
| | | | ADA0M1 register | Be sure to clear bits 6 to 4 to "0". | p. 392 <input type="checkbox"/> |
| | | | | Be sure to set the ADA0HS1 bit to "1". | p. 392 <input type="checkbox"/> |
| | | | Conversion mode setting example | Set as $3.1 \mu s \leq \text{conversion time} \leq 15.5 \mu s$. | p. 392 <input type="checkbox"/> |
| | | | | Rewriting of the ADA0M0, ADA0M2, ADA0S, ADA0PFM, and ADA0PFT registers and trigger input are prohibited during the stabilization time. | p. 392 <input type="checkbox"/> |
| | | | ADA0M2 register | Be sure to clear bits 7 to 2 to "0". | p. 393 <input type="checkbox"/> |
| | | | ADA0CRn, ADA0CRnH register | Accessing the ADA0CRn and ADA0CRnH registers is prohibited in the following statuses. For details, see 3.4.8 (2) Accessing specific on-chip peripheral I/O registers. <ul style="list-style-type: none"> When the CPU operates with the subclock and the main clock oscillation is stopped When the CPU operates with the internal oscillation clock | p. 395 <input type="checkbox"/> |
| | | | | 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. | p. 395 <input type="checkbox"/> |
| | | | ADA0PFM register | 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. | p. 397 <input type="checkbox"/> |
| | | | | 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. | p. 397 <input type="checkbox"/> |
| | | | 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. | p. 410 <input type="checkbox"/> |
| | | | 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. | p. 410 <input type="checkbox"/> |
| | | | 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 11-9 is recommended. | p. 410 <input type="checkbox"/> |

| Chapter | Classification | Function | Details of Function | Cautions | Page |
|------------|----------------|---------------|-------------------------------|--|---------------------------------|
| Chapter 11 | Soft | A/D converter | 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. Also the conversion resolution may drop at the pins set as output port pins during A/D conversion if the current flows due to the effect of the external circuit connected to the port pins. 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. | p. 410 <input type="checkbox"/> |
| | | | 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. | p. 411 <input type="checkbox"/> |
| | Hard | A/D converter | 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 11-12. (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 to suppress the reference voltage fluctuation as shown in Figure 11-12. (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. | p. 412 <input type="checkbox"/> |
| | | | Reading ADA0CRn result | When the ADA0M0 to ADA0M2, ADA0S, ADA0PFM, or ADA0PFT 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, ADA0S, ADA0PFM, or ADA0PFT registers. Also, when an external/timer trigger is acknowledged, the contents of the ADA0CRn register may be undefined. Read the conversion result after completion of conversion and before the next external/timer trigger is acknowledged. The correct conversion result may not be read at a timing different from the above. | p. 412 <input type="checkbox"/> |
| | | | A/D conversion result | If there is noise at the analog input pins and at the reference voltage input pins, that noise may generate an illegal conversion result. Software processing will be needed to avoid a negative effect on the system from this illegal conversion result. An example of this software processing is shown below. <ul style="list-style-type: none"> Take the average result of a number of A/D conversions and use that as the A/D conversion result. Execute a number of A/D conversions consecutively and use those results, omitting any exceptional results that may have been obtained. If an A/D conversion result that is judged to have generated a system malfunction is obtained, be sure to recheck the system malfunction before performing malfunction processing. | p. 412 <input type="checkbox"/> |

| Chapter | Classification | Function | Details of Function | Cautions | Page |
|------------|----------------|---|---|---|---------------------------------|
| Chapter 11 | Soft | A/D converter | Standby mode | Because the A/D converter stops operating in the STOP mode, conversion results are invalid, so power consumption can be reduced. Operations are resumed after the STOP mode is released, but the A/D conversion results after the STOP mode is released are invalid. When using the A/D converter after the STOP mode is released, before setting the STOP mode or releasing the STOP mode, clear the ADA0M0.ADA0CE bit to 0 then set the ADA0CE bit to 1 after releasing the STOP mode. In the IDLE1, IDLE2, or subclock operation mode, operation continues. To lower the power consumption, therefore, clear the ADA0M0.ADA0CE bit to 0. In the IDLE1 and IDLE2 modes, since the analog input voltage value cannot be retained, the A/D conversion results after the IDLE1 and IDLE2 modes are released are invalid. The results of conversions before the IDLE1 and IDLE2 modes were set are valid. | p. 413 <input type="checkbox"/> |
| | | | Rewriting registers and trigger input during the stabilization time | Rewriting of the ADA0M0, ADA0M2, ADA0S, ADA0PEM, and ADA0PFT registers and trigger input during the stabilization time are prohibited. | p. 413 <input type="checkbox"/> |
| | | | Variation of A/D conversion results | The results of the A/D conversion may vary depending on the fluctuation of the supply voltage, or may be affected by noise. To reduce the variation, take counteractive measures with the program such as averaging the A/D conversion results. | p. 413 <input type="checkbox"/> |
| | | | A/D conversion result hysteresis characteristics | The successive comparison type A/D converter holds the analog input voltage in the internal sample & hold capacitor and then performs A/D conversion. After the A/D conversion has finished, the analog input voltage remains in the internal sample & hold capacitor. As a result, the following phenomena may occur. <ul style="list-style-type: none"> When the same channel is used for A/D conversions, if the voltage is higher or lower than the previous A/D conversion, then hysteresis characteristics may appear where the conversion result is affected by the previous value. Thus, even if the conversion is performed at the same potential, the result may vary. When switching the analog input channel, hysteresis characteristics may appear where the conversion result is affected by the previous channel value. This is because one A/D converter is used for the A/D conversions. Thus, even if the conversion is performed at the same potential, the result may vary. Thus, even if the conversion is performed at the same potential, the result may vary. | p. 413 <input type="checkbox"/> |
| | | | UAnOPT0 register | Do not set the UAnSRT and UAnSTT bits (to 1) during SBF reception (UAnSRF bit = 1). | p. 423 <input type="checkbox"/> |
| Chapter 12 | Soft | Asynchronous serial interface A (UARTA) | SBF reception | If SBF is transmitted during a data reception, a framing error occurs. | p. 433 <input type="checkbox"/> |
| | | | | Do not set the SBF reception trigger bit (UAnSRT) and SBF transmission trigger bit (UAnSTT) to 1 during an SBF reception (UAnSRF = 1). | p. 433 <input type="checkbox"/> |
| | | | Continuous transmission | When initializing transmissions during the execution of continuous transmissions, make sure that the UAnSTR.UAnTSF bit is 0, then perform the initialization. Transmit data that is initialized when the UAnTSF bit is 1 cannot be guaranteed. In the case of continuous transmission, the communication rate from the stop bit to the start bit of the next data is extended by two operating clocks from the normal rate. | p. 435 <input type="checkbox"/> |

| Chapter | Chapter Classification | Function | Details of Function | Cautions | Page |
|------------|------------------------|---|--|---|---------------------------------|
| Chapter 12 | Soft | Asynchronous serial interface A (UARTA) | UART reception | 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. | p. 437 <input type="checkbox"/> |
| | | | | The operation during reception is performed assuming that there is only one stop bit. A second stop bit is ignored. | p. 437 <input type="checkbox"/> |
| | | | | 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. | p. 437 <input type="checkbox"/> |
| | | | | 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 (1) the UAnPWR bit = 0 or UAnRXE bit = 0. | p. 437 <input type="checkbox"/> |
| | | | Reception errors | When an INTUAnR signal is generated, the UAnSTR register must be read to check for errors. | p. 438 <input type="checkbox"/> |
| | | | | If a receive error interrupt occurs during continuous reception, read the contents of the UAnSTR register must be read before the next reception is completed, then perform error processing. | p. 439 <input type="checkbox"/> |
| | | | LIN function | When using the LIN function, fix the UAnCTL0.UAnPS1 and UAnCTL0.UAnPS0 bits to 00. | p. 440 <input type="checkbox"/> |
| | | | UAnCTL1 register | Clear the UAnCTL0.UAnPWR bit to 0 before rewriting the UAnCTL1 register. | p. 443 <input type="checkbox"/> |
| | | | UAnCTL2 register | Clear the UAnCTL0.UAnPWR bit to 0 or clear the UAnTXE and UAnRXE bits to 00 before rewriting the UAnCTL2 register. | p. 444 <input type="checkbox"/> |
| | | | Baud rate error | The baud rate error during transmission must be within the error tolerance on the receiving side. | p. 445 <input type="checkbox"/> |
| | | | | The baud rate error during reception must satisfy the range indicated in (5) Allowable baud rate range during reception. | p. 445 <input type="checkbox"/> |
| | | | Allowable baud rate range during reception | The baud rate error during reception must be set within the allowable error range using the following equation. | p. 447 <input type="checkbox"/> |
| | | | When the clock supply to UARTAn is stopped | 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. | p. 450 <input type="checkbox"/> |
| | | | RXDA1 pin KR7 pin | The RXDA1 and KR7 pins must not be used at the same time. To use the RXDA1 pin, do not use the KR7 pin. To use the KR7 pin, do not use the RXDA1 pin (it is recommended to set the PFC91 bit to 1 and clear PFCE91 bit to 0). | p. 450 <input type="checkbox"/> |

| Chapter | Classification | Function | Details of Function | Cautions | Page |
|----------------------|--|--|---|---|---------------------------------------|
| Chapter 12 | Soft | Asynchronous serial interface A (UARTA) | When performing the transfer of receive data | In UARTAn, the interrupt caused by a communication error does not occur. When performing the transfer of receive data, error processing cannot be performed even if errors (parity, overrun, framing) occur during transfer. Read the UAnSTR register during communication to check for errors. | p. 450 <input type="checkbox"/> |
| | | | Start up the UARTAn | Start up the UARTAn in the following sequence. <1> Set the UAnCTL0.UAnPWR bit to 1. <2> Set the ports. <3> Set the UAnCTL0.UAnTXE bit to 1, UAnCTL0.UAnRXE bit to 1. | p. 450 <input type="checkbox"/> |
| | | | Stop the UARTAn | Stop the UARTAn in the following sequence. <1> Set the UAnCTL0.UAnTXE bit to 0, UAnCTL0.UAnRXE bit to 0. <2> Set the ports and set the UAnCTL0.UAnPWR bit to 0 (it is not a problem if port setting is not changed). | p. 450 <input type="checkbox"/> |
| | | | In transmit mode | In transmit mode (UAnCTL0.UAnPWR bit = 1 and UAnCTL0.UAnTXE bit = 1), do not overwrite the same value to the UAnTX register by software because transmission starts by writing to this register. To transmit the same value continuously, overwrite the same value. | p. 450 <input type="checkbox"/> |
| | | | In continuous transmission | In continuous transmission, the communication rate from the stop bit to the next start bit is extended 2 base clocks more than usual. However, the reception side initializes the timing by detecting the start bit, so the reception result is not affected. | p. 450 <input type="checkbox"/> |
| | | | On-chip debug mode | If the break command is executed in the on-chip debug (OCD) mode and if UART receives data, an overrun error occurs. | p. 450 <input type="checkbox"/> |
| Chapter 13 | Soft | 3-wire variable-length serial I/O (CSIB) | CBnCTL0 register | To forcibly suspend transmission/reception, clear the CBnPWR, CBnTXE bit instead of the CBnRXE bit to 0. At this time, the clock output is stopped. | p. 454 <input type="checkbox"/> |
| | | | | Be sure to clear bits 3 and 2 to "0". | p. 456 <input type="checkbox"/> |
| | | | CBnCTL1 register | The CBnCTL1 register can be rewritten only when the CBnCTL0.CBnPWR bit = 0, or CBnCTL0.CBnTXE and CBnRXE bits = 0. | p. 457 <input type="checkbox"/> |
| | | | | Set so that communication clock (f _{CLK}) is 8 MHz or less. | p. 457 <input type="checkbox"/> |
| | | | CBnCTL2 register | The CBnCTL2 register can be rewritten only when the CBnCTL0.CBnPWR bit = 0 or when both the CBnTXE and CBnRXE bits = 0. | p. 458 <input type="checkbox"/> |
| | | | Continuous transfer mode (master mode, transmission mode) | In continuous transmission mode, the reception completion interrupt request signal (INTCBnR) is not generated. | p. 475 <input type="checkbox"/> |
| | | | Continuous transfer mode (slave mode, transmission mode) | In continuous transmission mode, the reception completion interrupt request signal (INTCBnR) is not generated. | p. 484 <input type="checkbox"/> |
| | | | Clock timing | In single transfer mode, writing to the CBnTX register with the CBnTSF bit set to 1 is ignored. This has no influence on the operation during transfer. | pp. 493, 494 <input type="checkbox"/> |
| | | | PRSM0 register | Do not rewrite the PRSM0 register while watch timer and CSIB0 are operating. | p. 496 <input type="checkbox"/> |
| | | | | Set the PRSM0 register before setting the BGCE0 bit to 1. | p. 496 <input type="checkbox"/> |
| PRSCM0 register | Do not rewrite the PRSCM0 register while watch timer and CSIB are operating. | p. 497 <input type="checkbox"/> | | | |
| | Set the PRSCM0 register before setting the PRSM0.BGCE0 bit to 1. | p. 497 <input type="checkbox"/> | | | |
| Baud rate generation | Set so that the f _{BRG} is 8 MHz or less. | p. 497 <input type="checkbox"/> | | | |

| Chapter | Classification | Function | Details of Function | Cautions | Page |
|------------|----------------|--|--|---|--|
| Chapter 13 | Soft | 3-wire variable-length serial I/O (CSIB) | CBnCTL0 register CBnCTL1 register CBnCTL2 register | In regards to registers that are forbidden from being rewritten during operations (CBnCTL0.CBnPWR bit is 1), if rewriting has been carried out by mistake during operations, set the CBnCTL0.CBnPWR bit to 0 once, then initialize CSIBn. Registers to which rewriting during operation are prohibited are shown below. <ul style="list-style-type: none"> • CBnCTL0 register: CBnTXE, CBnRXE, CBnDIR, CBnTMS bits • CBnCTL1 register: CBnCKP, CBnDAP, CBnCKS2 to CBnCKS0 bits • CBnCTL2 register: CBnCL3 to CBnCL0 bits | p. 498 <input type="checkbox"/> |
| | | | Communication type 2 and 4 | In communication type 2 and 4 (CBnCTL1.CBnDAP bit = 1), the CBnSTR.CBnTSF bit is cleared half a $\overline{\text{SCKBn}}$ clock after occurrence of a reception complete interrupt (INTCBnR). In the single transfer mode, writing the next transmit data is ignored during communication (CBnTSF bit = 1), and the next communication is not started. Also if reception-only communication (CBnCTL0.CBnTXE bit = 0, CBnCTL0.CBnRXE bit = 1) is set, the next communication is not started even if the receive data is read during communication (CBnTSF bit = 1). Therefore, when using the single transfer mode with communication type 2 or 4 (CBnDAP bit = 1), pay particular attention to the following. <ul style="list-style-type: none"> • To start the next transmission, confirm that CBnTSF bit = 0 and then write the transmit data to the CBnTX register. • To perform the next reception continuously when reception-only communication (CBnTXE bit = 0, CBnRXE bit = 1) is set, confirm that CBnTSF bit = 0 and then read the CBnRX register. Or, use the continuous transfer mode instead of the single transfer mode | p. 498 <input type="checkbox"/> |
| Chapter 14 | Soft | Interrupt/exception processing function | Non-maskable interrupts | For the non-maskable interrupt servicing executed by the non-maskable interrupt request signal (INTWDT2), see 14.2.2 (2) From INTWDT2 signal. | p. 502 <input type="checkbox"/> |
| | | | | 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 the NP bit back to 1 using the LDSR instruction immediately before the RETI instruction. | p. 505 <input type="checkbox"/> |
| | | | Maskable interrupts | 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. | p. 509 <input type="checkbox"/> |
| | | | Multiple interrupt | 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. | pp.511 to 513 <input type="checkbox"/> |
| | | | Interrupt control register | Disable interrupts (DI) or mask the interrupt to read the xxlCn.xxIFn bit. If the xxIFn bit is read while interrupts are enabled (EI) or while the interrupt is unmasked, the correct value may not be read when acknowledging an interrupt and reading the bit conflict. The flag xxIFn is reset automatically by the hardware if an interrupt request signal is acknowledged. | p. 514 <input type="checkbox"/> p. 514 <input type="checkbox"/> |
| | | | IMPO to IMR2 register | The device file defines the xxlCn.xxMKn bit as a reserved word. If a bit is manipulated using the name of xxMKn, the contents of the xxlCn register, instead of the IMRm register, are rewritten (as a result, the contents of the IMRm register are also rewritten). To read bits 8 to 15 of the IMR0 to IMR2 registers in 8-bit or 1-bit units, specify them as bits 0 to 7 of the IMR0H to IMR2H registers. | p. 516 <input type="checkbox"/> p. 516 <input type="checkbox"/> |
| | | | | | |

| Chapter | Classification | Function | Details of Function | Cautions | Page |
|------------|----------------|---|---|---|---------------------------------|
| Chapter 14 | Soft | Interrupt/exception processing function | IMP0 to IMR2 register | Set bits 15 to 11 and 7 to 4 of the IMR2 register to "1". If the setting of these bits is changed, the operation is not guaranteed. | p. 516 <input type="checkbox"/> |
| | | | ISPR register | 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). | p. 517 <input type="checkbox"/> |
| | | | Recovery from software exception processing | When the EP and NP bits are changed by the LDSR instruction during the 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. | p. 520 <input type="checkbox"/> |
| | | | Illegal opcode definition | Since it is possible to assign this instruction to an illegal opcode in the future, it is recommended that it not be used. | p. 522 <input type="checkbox"/> |
| | | | Restoration from illegal opcode | DBPC and DBPSW can be accessed only during the interval between the execution of the illegal opcode and the DBRET instruction. | p. 523 <input type="checkbox"/> |
| | | | Restoration from a debug trap | DBPC and DBPSW can be accessed only during the interval between the execution of the DBTRAP instruction and the DBRET instruction. | p. 525 <input type="checkbox"/> |
| | | | INTF0, INTR0 registers | 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. | p. 527 <input type="checkbox"/> |
| | | | | Be sure to clear the INTF0n and INTR0n bits to 00 if the corresponding pin is not used as the NMI or INTP0 to INTP3 pins. | p. 527 <input type="checkbox"/> |
| | | | INTR3L, INTF3L registers | 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. | p. 528 <input type="checkbox"/> |
| | | | | Be sure to clear the INTF31 and INTR31 bits to 00 if the corresponding pin is not used as the INTP7 pin. | p. 528 <input type="checkbox"/> |
| | | | INTF9H, INTR9H registers | 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. | p. 529 <input type="checkbox"/> |
| | | | | Be sure to clear the INTF9n and INTR9n bits to 00 if the corresponding pin is not used as INTP4 to INTP6 pins. | p. 529 <input type="checkbox"/> |
| | | | NFC register | Time equal to the sampling clock × the number of times set by the NFSTS bit is required until the digital noise eliminator is initialized after the sampling clock has been changed. If the valid edge of INTP3 is input after the sampling clock has been changed and before the time of the sampling clock × the number of times set by the NFSTS bit passes, therefore, the interrupt request signal may be generated. Therefore, note the following points when using the interrupt function. <ul style="list-style-type: none"> When using the interrupt function, after the sampling clock × the number of times set by the NFSTS bit have elapsed, enable interrupts after the interrupt request flag (PIC3.PIF3 bit) has been cleared. | p. 530 <input type="checkbox"/> |
| | | | NMI pin | The NMI pin alternately functions as the P02 pin. It functions as a normal port pin after reset. To enable the NMI pin, validate the NMI pin with the PMC0 register. The initial setting of the NMI pin is "No edge detected". Select the NMI pin valid edge using the INTF0 and INTR0 registers. | p. 533 <input type="checkbox"/> |

| Chapter | Chapter Classification | Function | Details of Function | Cautions | Page |
|----------------------|--|-----------------------------------|---|--|---------------------------------|
| Chapter 15 | Soft | Key interrupt function | KRM register | Rewrite the KRM register after once clearing the KRM register to 00H. | p. 535 <input type="checkbox"/> |
| | | | | 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) or masking, then clear the interrupt request flag (KRIC.KRIF bit) to 0, and enable interrupts (EI) or clear the mask. | p. 535 <input type="checkbox"/> |
| | | | KR0 to KR7 pin | 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. | p. 535 <input type="checkbox"/> |
| | | | RXDA1 pin KR7 pin | The RXDA1 and KR7 pins must not be used at the same time. To use the RXDA1 pin, do not use the KR7 pin. To use the KR7 pin, do not use the RXDA1 pin (it is recommended to set the PFC91 bit to 1 and clear PFCE91 bit to 0). | p. 535 <input type="checkbox"/> |
| | | To use the key interrupt function | To use the key interrupt function, be sure to set the port pin to the key return pin and then enable the operation with the KRM register. To switch from the key return pin to the port pin, disable the operation with the KRM register and then set the port pin. | p. 535 <input type="checkbox"/> | |
| Chapter 16 | Soft | Standby function | PSC register | Before setting the IDLE1, IDLE2, STOP, or sub-IDLE mode, set the PSMR.PSM1 and PSMR.PSM0 bits and then set the STP bit. | p. 538 <input type="checkbox"/> |
| | | | | Settings of the NMI1M, NMI0M, and INTM bits are invalid when HALT mode is released. | p. 538 <input type="checkbox"/> |
| | | | | If the NMI1M, NMI0M, or INTM bit is set to 1 at the same time the STP bit is set to 1, the setting of NMI1M, NMI0M, or INTM bit becomes invalid. If there is an unmasked interrupt request signal being held pending when the IDLE1/IDLE2/STOP mode is set, set the bit corresponding to the interrupt request signal (NMI1M, NMI0M, or INTM) to 1, and then set the STP bit to 1. | p. 538 <input type="checkbox"/> |
| | | | PSMR register | Be sure to clear bits 2 to 7 to "0". | p. 539 <input type="checkbox"/> |
| | | | | The PSM0 and PSM1 bits are valid only when the PSC.STP bit is 1. | p. 539 <input type="checkbox"/> |
| | | | OSTS register | 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 or the occurrence of an interrupt request signal. | p. 540 <input type="checkbox"/> |
| | | | | Be sure to clear bits 3 to 7 to "0". | p. 540 <input type="checkbox"/> |
| | | | | The oscillation stabilization time following reset release is $2^{16}/fx$ (because the initial value of the OSTS register = 06H). | p. 540 <input type="checkbox"/> |
| | | | HALT mode | Insert five or more NOP instructions after the HALT instruction. | p. 541 <input type="checkbox"/> |
| | | | | If 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. | p. 541 <input type="checkbox"/> |
| | | | IDLE1 mode | Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the IDLE1 mode. | p. 543 <input type="checkbox"/> |
| | | | | If the IDLE1 mode is set while an unmasked interrupt request signal is being held pending, the IDLE1 mode is released immediately by the pending interrupt request. | p. 543 <input type="checkbox"/> |
| Releasing IDLE1 mode | 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. | p. 543 <input type="checkbox"/> | | | |
| | If eliminating digital noise is selected by using the NFC register and if the sampling clock is selected from fxx/64, fxx/128, fxx/256, fxx/512, and fxx/1024, the IDLE1 mode cannot be released by the interrupt request signal of the INTP3 pin. For details, see 14.6.2 (4) Noise elimination control register (NFC). | p. 543 <input type="checkbox"/> | | | |

| Chapter | Classification | Function | Details of Function | Cautions | Page |
|------------|----------------|------------------|---|---|---------------------------------|
| Chapter 16 | Soft function | Standby function | IDLE2 mode | Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the IDLE2 mode. | p. 545 <input type="checkbox"/> |
| | | | | If the IDLE2 mode is set while an unmasked interrupt request signal is being held pending, the IDLE2 mode is released immediately by the pending interrupt request. | p. 545 <input type="checkbox"/> |
| | | | Releasing IDLE2 mode | 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. | p. 545 <input type="checkbox"/> |
| | | | | If eliminating digital noise is selected by using the NFC register and if the sampling clock is selected from fxx/64, fxx/128, fxx/256, fxx/512, and fxx/1024, the IDLE2 mode cannot be released by the interrupt request signal of the INTP3 pin. For details, see 14.6.2 (4) Noise elimination control register (NFC). | p. 545 <input type="checkbox"/> |
| | | | Stop mode | Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the STOP mode. | p. 548 <input type="checkbox"/> |
| | | | | If the STOP mode is set while an unmasked interrupt request signal is being held pending, the STOP mode is released immediately by the pending interrupt request. | p. 548 <input type="checkbox"/> |
| | | | Releasing STOP mode | The interrupt request that is disabled by setting the PSC.NMI1M, PSC.NMI0M, and PSC.INTM bits to 1 becomes invalid and STOP mode is not released. | p. 548 <input type="checkbox"/> |
| | | | | If eliminating digital noise is selected by using the NFC register and if the sampling clock is selected from fxx/64, fxx/128, fxx/256, fxx/512, and fxx/1024, the STOP mode cannot be released by the interrupt request signal of the INTP3 pin. For details, see 14.6.2 (4) Noise elimination control register (NFC). | p. 548 <input type="checkbox"/> |
| | | | Operating status in STOP mode | 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 after the STOP mode is released are invalid. All the A/D conversion results before the STOP mode is set are invalid. | p. 549 <input type="checkbox"/> |
| | | | | 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. | p. 549 <input type="checkbox"/> |
| | | | Subclock operation mode | 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, see 5.3 (1) Processor clock control register (PCC). | p. 551 <input type="checkbox"/> |
| | | | | 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. Internal system clock (f_{CLK}) > Subclock (f_{XT}) × 4 | p. 551 <input type="checkbox"/> |
| | | | Releasing subclock operation mode | 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, see 5.3 (1) Processor clock control register (PCC). | p. 551 <input type="checkbox"/> |
| | | | Operating status in subclock operation mode | Be sure to stop the PLL (PLLCTL.PLLON = 0) before stopping the main clock. | p. 552 <input type="checkbox"/> |
| | | | | 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 (see 3.4.8 (2)). | p. 552 <input type="checkbox"/> |
| | | | Sub-IDLE mode | Following the store instruction to the PSC register for setting the sub-IDLE mode, insert five or more NOP instructions. | p. 553 <input type="checkbox"/> |

| Chapter | Classification | Function | Details of Function | Cautions | Page |
|--|---------------------------------|----------------------|------------------------------------|---|--------------------------------------|
| Chapter 16 | Soft | Standby function | Sub-IDLE mode | If the sub-IDLE mode is set while an unmasked interrupt request signal is being held pending, the sub-IDLE mode is then released immediately by the pending interrupt request. | p. 553 <input type="checkbox"/> |
| | | | Releasing sub-IDLE mode | The 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. | p. 554 <input type="checkbox"/> |
| | | | | When the sub-IDLE mode is released, 12 cycles of the subclock (about 366 μ s) elapse from when the interrupt request signal that releases the sub-IDLE mode is generated to when the mode is released. | p. 554 <input type="checkbox"/> |
| | | | | If eliminating digital noise is selected by using the NFC register and if the sampling clock is selected from fxx/64, fxx/128, fxx/256, fxx/512, and fxx/1024, the sub-IDLE mode cannot be released by the interrupt request signal of the INTP3 pin. For details, see 14.6.2 (4) Noise elimination control register (NFC). | p. 554 <input type="checkbox"/> |
| | | | Operating status in sub-IDLE Mode | Be sure to stop the PLL (PLLCTL.PLLON bit = 0) before stopping the main clock. | p. 555 <input type="checkbox"/> |
| To realize low power consumption, stop the A/D converter before shifting to the sub-IDLE mode. | p. 555 <input type="checkbox"/> | | | | |
| Chapter 17 | Soft | Reset functions | Emergency operation mode | When the CPU is being operated with the internal oscillation clock, access to the register in which a wait state is generated is prohibited. For the register in which a wait state is generated, see 3.4.8 (2) Accessing specific on-chip peripheral I/O registers. | p. 556 <input type="checkbox"/> |
| | | | RESF register | 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. | p. 557 <input type="checkbox"/> |
| | Hard | | Internal RAM status after reset | The firmware of the V850ES/HF2 uses a part of the internal RAM after the internal system reset status has been released because it supports a boot swap function. Therefore, the contents of some RAM areas are not retained after power-on reset. For details, see 17.4 Operation After Reset Release. | pp.558, 560 <input type="checkbox"/> |
| | | | Hardware status on RESET pin input | When the power is turned on, the following pin may output an undefined level temporarily even during reset. • P53/KR3/TIQ00/TOQ00/DDO pin | p. 558 <input type="checkbox"/> |
| | | | | The OCDM register is initialized by the $\overline{\text{RESET}}$ pin input. Therefore, note with caution that, if a high level is input to the P05/DRST pin after a reset release before the OCDM.OCDM0 bit is cleared, the on-chip debug mode is entered. For details, see CHAPTER 4 PORT FUNCTIONS. | p. 558 <input type="checkbox"/> |
| Chapter 18 | Soft | Clock monitor | CLM register | Once the CLME bit has been set to 1, it cannot be cleared to 0 by any means other than reset. | p. 565 <input type="checkbox"/> |
| | | | | When a reset by the clock monitor occurs, the CLME bit is cleared to 0 and the RESF.CLMRF bit is set to 1. | p. 565 <input type="checkbox"/> |
| | | | Internal oscillator | The internal oscillator can be stopped by using the option byte function (see CHAPTER 23) to enable the internal oscillator to stop, and setting the RCM.RSTOP bit to 1. | p. 566 <input type="checkbox"/> |
| | | | | The clock monitor is stopped while the internal oscillator is stopped. | p. 566 <input type="checkbox"/> |
| Chapter 20 | Soft | Low-voltage detector | LVIM register | After setting the LVION bit to 1, wait for 0.2 ms (MAX.) before checking the voltage using the LVIF bit. | p. 572 <input type="checkbox"/> |
| | | | | The value of the LVIF flag is output as the output signal INTLVI when the LVION bit = 1 and LVIMD bit = 0. | p. 572 <input type="checkbox"/> |
| | | | | Be sure to clear bits 2 to 6 to "0". | p. 572 <input type="checkbox"/> |
| | | | | The low-voltage detector cannot be stopped until a reset request due to something other than low-voltage detection is generated after the LVIM.LVION and LVIM.LVIMD bits are set to 1. | p. 572 <input type="checkbox"/> |

| Chapter | Classification | Function | Details of Function | Cautions | Page |
|----------------------|--|---------------------------------|--|--|---------------------------------------|
| Chapter 20 | Soft | Low-voltage detector | LVIS register | This register cannot be written until a reset request due to something other than low-voltage detection is generated after the LVIM.LVION and LVIM.LVIMD bits are set to 1. Be sure to clear bits 1 to 7 to "0". | p. 573 <input type="checkbox"/> |
| | | | RAMS register | The following shows the specific sequence after reset. <ul style="list-style-type: none"> Setting conditions: <ul style="list-style-type: none"> Detection of voltage lower than detection level Set by instruction Generation of reset signal by watchdog timer overflow Generation of reset signal while RAM is being accessed Generation of reset signal by clock monitor Clearing condition: Writing of 0 in specific sequence | p. 573 <input type="checkbox"/> |
| | | | To use for internal reset signal | If the LVIMD bit is set to 1, the contents of the LVIM and LVIS registers cannot be changed until a reset request other than LVI is generated. | p. 574 <input type="checkbox"/> |
| | | | PEMU1 register | EVARAMIN bit is not automatically cleared. | p. 578 <input type="checkbox"/> |
| | | | | | |
| Chapter 22 | Hard | Flash memory | Flash memory mapping | 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. | p. 582 <input type="checkbox"/> |
| | | | Communication mode | Process the pins not shown in compliance with the processing of unused pins (see 2.3 Pin I/O Circuit Types and Recommended Connection of Unused Pins). Connect a resistor of 1 kΩ to 10 kΩ as necessary. | pp. 587, 588 <input type="checkbox"/> |
| | | | | Do not input a high level to the \overline{DRST} pin. | pp. 587, 588 <input type="checkbox"/> |
| | | | PG-FP4 | Wire these pins as shown in Figure 22-6, or connect them to GND via pull-down resistor on board. | p. 588 <input type="checkbox"/> |
| | | | | Clock cannot be supplied via the CLK pin of the flash programmer. Create an oscillator on board and supply the clock. | p. 588 <input type="checkbox"/> |
| | | | FA-80GK-9EU-A | Be sure to connect the REGC pin to GND via a 4.7 μF (recommended value) capacitor. | p. 589 <input type="checkbox"/> |
| | | | | A clock cannot be supplied from the CLK pin of the flash programmer. Create an oscillator on the board and supply the clock from that oscillator. | p. 589 <input type="checkbox"/> |
| | | | FA-80GK-9EU-A (in CSIB0 + HS Mode) | Wire the FLMD1 pin as shown below, or connect it to GND on board via a pull-down resistor. | p. 591 <input type="checkbox"/> |
| | | | | Supply a clock by creating an oscillator on the flash writing adapter (enclosed by the broken lines). Here is an example of the oscillator. | p. 591 <input type="checkbox"/> |
| | | | | Do not input a high level to the \overline{DRST} pin. | p. 591 <input type="checkbox"/> |
| | | | Selection of communication mode | 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. | p. 593 <input type="checkbox"/> |
| | | | FLMD1 pin connection | If the VDD signal is input to the FLMD1 pin from another device during on-board writing and immediately after reset, isolate this signal. | p. 595 <input type="checkbox"/> |
| | | | Secure self programming (boot swap function) | The boot swap function is not supported in the μPD70F3702. | p. 600 <input type="checkbox"/> |
| FLMD0 pin processing | Make sure that the FLMD0 pin is at 0 V when reset is released. | p. 602 <input type="checkbox"/> | | | |

| Chapter | Classification | Function | Details of Function | Cautions | Page |
|------------|----------------|--|---------------------------------|---|---------------------------------|
| Chapter 23 | Hard | Option byte function | CA850 sample program | Be sure to write for 6 bytes in this section. If less than 6 bytes, an error occurs on a linker operation. Error message: F4112: illegal "OPTION_BYTES" section size. | p. 605 <input type="checkbox"/> |
| Chapter 24 | Hard, Soft | On-chip debug function | OCDM register | 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. <ul style="list-style-type: none"> • Input a low level to the P05/INTP2/$\overline{\text{DRST}}$ pin. • Set the OCDM0 bit. In this case, take the following actions. <ul style="list-style-type: none"> <1> Clear the OCDM0 bit to 0. <2> Fix the P05/INTP2/$\overline{\text{DRST}}$ pin to low level until <1> is completed. | p. 610 <input type="checkbox"/> |
| | | | | The $\overline{\text{DRST}}$ pin has an on-chip pull-down resistor. This resistor is disconnected when the OCDM0 flag is cleared to 0. | p. 610 <input type="checkbox"/> |
| | | | Cautions (with DCU) | 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. | p. 612 <input type="checkbox"/> |
| | | | | 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. | p. 612 <input type="checkbox"/> |
| | | | | Because a software breakpoint set in the internal flash memory 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. | p. 612 <input type="checkbox"/> |
| | | | | 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 DMM 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. | p. 612 <input type="checkbox"/> |
| | | | | When the following conditions (a) and (b) are satisfied and operation is stopped on the emulator (IECUBE, MINICUBE) due to a break, etc., watchdog timer 2 does not stop and a reset or non-maskable interrupt occurs. When a reset occurs, the debugger hangs up. <ul style="list-style-type: none"> (a) The main clock or subclock is used as the source clock for watchdog timer 2. (b) The internal oscillation clock is stopped (RCM.RSTOP bit = 1). To avoid this, perform either of the following. <ul style="list-style-type: none"> • When an emulator is used, use the internal oscillation clock as the source clock. • When an emulator is used, do not stop the internal oscillator. | p. 612 <input type="checkbox"/> |
| | | | | When the following conditions (a) and (b) are satisfied and operation is stopped on the emulator (IECUBE, MINICUBE) due to a break, etc., TMM does not stop even if the peripheral break function is set to "Break". <ul style="list-style-type: none"> (a) Either the INTWT, internal oscillation clock ($f_{R/8}$), or subclock are selected as the TMM source clock. (b) The main clock is stopped. To avoid this, perform either of the following. <ul style="list-style-type: none"> • When an emulator is used, the main clock (f_{xx}, $f_{xx}/2$, $f_{xx}/4$, $f_{xx}/64$, $f_{xx}/512$) is used as the source clock. • When an emulator is used, disable the main clock oscillation. | p. 612 <input type="checkbox"/> |
| Hard | | In the on-chip debug mode, the DDO pin is forcibly set to the high-level output. | p. 612 <input type="checkbox"/> | | |

| Chapter | Classification | Function | Details of Function | Cautions | Page | |
|---|---------------------------------------|------------------------|------------------------|--|--|---------------------------------|
| Chapter 24 | Hard Soft | On-chip debug function | Cautions (without DCU) | Do not mount a device that was used for debugging on a mass-produced product, because the flash memory was rewritten during debugging and the number of rewrites of the flash memory cannot be guaranteed. Moreover, do not embed the debug monitor program into mass-produced products. | p. 621 <input type="checkbox"/> | |
| | | | | Forced breaks cannot be executed if one of the following conditions is satisfied. <ul style="list-style-type: none"> • Interrupts are disabled (DI) • Interrupts issued for the serial interface, which is used for communication between MINICUBE2 and the target device, are masked • Standby mode is entered while standby release by a maskable interrupt is prohibited • Mode for communication between MINICUBE2 and the target device is UARTA0, and the main clock has been stopped | p. 621 <input type="checkbox"/> | |
| | | | | The pseudo RRM function and DMM function do not operate if one of the following conditions is satisfied. <ul style="list-style-type: none"> • Interrupts are disabled (DI) • Interrupts issued for the serial interface, which is used for communication between MINICUBE2 and the target device, are masked • Standby mode is entered while standby release by a maskable interrupt is prohibited • Mode for communication between MINICUBE2 and the target device is UARTA0, and the main clock has been stopped • Mode for communication between MINICUBE2 and the target device is UARTA0, and a clock different from the one specified in the debugger is used for communication | p. 621 <input type="checkbox"/> | |
| | | | | The standby mode is released by the pseudo RRM function and DMM function if one of the following conditions is satisfied. <ul style="list-style-type: none"> • Mode for communication between MINICUBE2 and the target device is CSIB0 • Mode for communication between MINICUBE2 and the target device is UARTA0, and the main clock has been supplied. | p. 621 <input type="checkbox"/> | |
| | | | | Peripheral I/O registers that requires a specific sequence cannot be written with the DMM function. | p. 621 <input type="checkbox"/> | |
| | | | | Chip erase and writing of the monitor program for debugging are conducted when the debugger is first started up, but this operation takes about a dozen seconds. | p. 621 <input type="checkbox"/> | |
| | | | | When CPU operation clock settings are changed with the debugger, the debugger rewrites the monitor program. The time required is the same as that mentioned just above in (6). For the integrated debugger ID850QB, this applies when settings of the Clock column in the configuration dialog box are changed. | p. 621 <input type="checkbox"/> | |
| | | | | If a space where the debug monitor program is allocated is rewritten by flash self programming, the debugger can no longer operate normally. | p. 621 <input type="checkbox"/> | |
| | | | | Security ID | After the flash memory is erased, 1 is written to the entire area. | p. 622 <input type="checkbox"/> |
| | | | | Chapter 25 | Hard | Electrical specifications |
| Avoid direct connections among the IC device output (or I/O) pins and between VDD or VCC and GND. | pp. 625, 626 <input type="checkbox"/> | | | | | |

| Chapter | Classification | Function | Details of Function | Cautions | Page | |
|------------|----------------|---------------------------|--------------------------|--|--|---|
| Chapter 25 | Hard | Electrical specifications | Absolute maximum ratings | 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. | pp. 625, 626 <input type="checkbox"/> | |
| | | | | When directly connecting the external circuit to the pin that becomes high impedance state, the timing must be designed such that the output conflict is avoided on the external circuit. | pp. 625, 626 <input type="checkbox"/> | |
| | Soft | | | Main clock oscillator characteristics | When using the main clock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance. <ul style="list-style-type: none"> • 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. | p. 628 <input type="checkbox"/> |
| | | | | | When the main clock is stopped and the subclock is operating, wait until the oscillation stabilization time has been secured by the program before switching back to the main clock. | p. 628 <input type="checkbox"/> |
| | Hard | | | Subclock oscillator characteristics | When using the subclock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance. <ul style="list-style-type: none"> • 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. | p. 629 <input type="checkbox"/> |
| | | | | | 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. | p. 629 <input type="checkbox"/> |
| | | | | | Voltage regulator characteristics | The be sure that V _{DD} rises while RESET = V _{SS} = 0 V. |
| | | | | Pin leakage current | The value of the FLMD0 pin is as follows. <ul style="list-style-type: none"> • Input leakage current, high: 2 μA (MAX.) • Input leakage current, low: -2 μA (MAX.) | p. 632 <input type="checkbox"/> |
| | | | | Data retention characteristics | Shifting to STOP mode and restoring from STOP mode must be performed within the rated operating range. | p. 634 <input type="checkbox"/> |

APPENDIX D LIST OF CAUTIONS

(29/29)

| Chapter | Classification | Function | Details of Function | Cautions | Page |
|------------|----------------|----------------------------------|----------------------------------|--|---------------------------------|
| Chapter 25 | Hard | Electrical specifications | AC characteristics | 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. | p. 635 <input type="checkbox"/> |
| | Soft | | Programming characteristics | When writing initially to shipped products, it is counted as one rewrite for both "erase to write" and "write only". Example (P: Write, E: Erase) Shipped product → P → E → P → E → P: 3 rewrites Shipped product → E → P → E → P → E → P: 3 rewrites | p. 644 <input type="checkbox"/> |
| Chapter 27 | Hard | Recommended soldering conditions | Recommended Soldering conditions | Do not use different soldering methods together (except for partial heating). | p. 646 <input type="checkbox"/> |
| Appendix A | Soft | Development tools | RX850, RX850 Pro | To purchase the RX850 or RX850 Pro, first fill in the purchase application form and sign the license agreement. | p. 655 <input type="checkbox"/> |
| Appendix C | Soft | Instruction set list | Instruction set | Do not specify the same register for general-purpose registers reg1 and reg3. | p. 673 <input type="checkbox"/> |

APPENDIX E REVISION HISTORY

E.1 Major Revisions in This Edition

(1/2)

| Page | Description |
|--------|---|
| p. 40 | Addition of 2.5 Caution |
| p. 68 | Addition of description to 3.4.7 Special registers |
| p. 145 | Addition of 4.5.1 (b) Cautions on alternate-function mode (input) |
| p. 147 | Modification of Figure 5-1 Clock Generator |
| p. 148 | Modification of description in 5.2 (8) Prescaler 4 |
| p. 154 | Modification of Table 5-1 Operation Status of Each Clock |
| p. 158 | Modification of 5.5.2 (4) Programmable clock mode register (PCLM) |
| p. 167 | Modification of 6.4 (3) TMPn I/O control register 0 (TPnIOC0) |
| p. 189 | Addition of Caution to Figure 6-11 Register Setting for Operation in External Event Count Mode |
| p. 210 | Addition of Caution to Figure 6-22 Setting of Registers in One-Shot Pulse Output Mode |
| p. 253 | Addition of Caution 2 to 6.7 (1) Selector operation control register 0 (SELCNT0) |
| p. 263 | Addition of Note to 7.4 (3) TMQ0 I/O control register 0 (TQ0IOC0) |
| p. 289 | Addition of Caution to Figure 7-11 Register Setting for Operation in External Event Count Mode |
| p. 313 | Addition of Caution to Figure 7-22 Setting of Registers in One-Shot Pulse Output Mode |
| p. 382 | Modification of Cautions 3, 4 in 10.3 (1) Watchdog timer mode register 2 (WDTM2) |
| p. 384 | Modification of Caution 3 in 10.3 (2) Watchdog timer enable register (WDTE) |
| p. 392 | Addition of Cautions to Table 11-2 Conversion Mode Setting Example |
| p. 397 | Modification of description in 11.4 (7) Power-fail compare threshold value register (ADA0PFT) |
| p. 412 | Modification of description to 11.6 (8) Reading ADA0CRn result |
| p. 413 | Addition of 11.6 (10) Standby mode |
| p. 413 | Addition of 11.6 (11) Rewriting registers and trigger input during the stabilization time |
| p. 413 | Modification of description to 11.6 (13) A/D conversion result hysteresis characteristics |
| p. 421 | Addition of description to 12.3 (1) UARTAn control register 0 (UAnCTL0) |
| p. 423 | Addition of description to 12.3 (4) UARTAn option control register 0 (UAnOPT0) |
| p. 427 | Addition of description to 12.4 (1) Reception complete interrupt request signal (INTUAnR) |
| p. 433 | Addition of Caution to 12.5.4 SBF reception |
| p. 454 | Modification of Caution in 13.3 (1) CSIBn control register 0 (CBnCTL0) |
| p. 457 | Modification of Caution in and addition of Note 1 to 13.3 (2) CSIBn control register 1 (CBnCTL1) |
| p. 462 | Modification of 13.5 Operation |
| p. 497 | Addition of Caution to 13.7.1 Baud rate generation |
| p. 530 | Modification of description to 14.6.2 (4) Noise elimination control register (NFC) |
| p. 538 | Addition of Caution 3 to 16.2 (1) Power save control register (PSC) |
| p. 558 | Modification of description to Note 1 in Table 17-1 Hardware Status on RESET Pin Input |
| p. 560 | Modification of description to Note in Table 17-2 Hardware Status During Watchdog Timer 2 Reset Operation |
| p. 562 | Addition of 17.4 Operation After Reset Release |
| p. 572 | Addition of Caution 4 to 20.3 (1) Low-voltage detection register (LVIM) |
| p. 575 | Modification of Figure 20-2 Operation Timing of Low-Voltage Detector (LVIMD Bit = 1) |

| Page | Description |
|--------|---|
| p. 582 | Modification of 22.2 Memory Configuration |
| p. 583 | Addition of 22.3 Functional Outline |
| p. 587 | Modification of transfer rate in 22.4 2 (1) UARTA0 |
| p. 594 | Modification of Table 22-7 Flash Memory Control Commands |
| p. 601 | Modification of Figure 22-17 Standard Self Programming Flow |
| p. 603 | Modification of Table 22-11 Internal Resources Used |
| p. 604 | Addition of description to CHAPTER 23 OPTION BYTE FUNCTION |
| p. 606 | Modification of CHAPTER 24 ON-CHIP DEBUG FUNCTION |
| p. 630 | Addition of Caution to 25.5 Voltage Regulator Characteristics |
| p. 646 | Addition of CHAPTER 27 RECOMMENDED SOLDERING CONDITIONS |
| p. 647 | Addition of APPENDIX A DEVELOPMENT TOOLS |
| p. 674 | Addition of APPENDIX D LIST OF CAUTIONS |
| p. 703 | Addition of APPENDIX E REVISION HISTORY |

*For further information,
please contact:*

NEC Electronics Corporation

1753, Shimonumabe, Nakahara-ku,
Kawasaki, Kanagawa 211-8668,
Japan
Tel: 044-435-5111
<http://www.necel.com/>

[America]

NEC Electronics America, Inc.

2880 Scott Blvd.
Santa Clara, CA 95050-2554, U.S.A.
Tel: 408-588-6000
800-366-9782
<http://www.am.necel.com/>

[Europe]

NEC Electronics (Europe) GmbH

Arcadiastrasse 10
40472 Düsseldorf, Germany
Tel: 0211-65030
<http://www.eu.necel.com/>

Hanover Office

Podbielskistrasse 166 B
30177 Hannover
Tel: 0 511 33 40 2-0

Munich Office

Werner-Eckert-Strasse 9
81829 München
Tel: 0 89 92 10 03-0

Stuttgart Office

Industriestrasse 3
70565 Stuttgart
Tel: 0 711 99 01 0-0

United Kingdom Branch

Cygnus House, Sunrise Parkway
Linford Wood, Milton Keynes
MK14 6NP, U.K.
Tel: 01908-691-133

Succursale Française

9, rue Paul Dautier, B.P. 52
78142 Velizy-Villacoublay Cédex
France
Tel: 01-3067-5800

Sucursal en España

Juan Esplandiu, 15
28007 Madrid, Spain
Tel: 091-504-2787

Tyskland Filial

Täby Centrum
Entrance S (7th floor)
18322 Täby, Sweden
Tel: 08 638 72 00

Filiale Italiana

Via Fabio Filzi, 25/A
20124 Milano, Italy
Tel: 02-667541

Branch The Netherlands

Steijgerweg 6
5616 HS Eindhoven
The Netherlands
Tel: 040 265 40 10

[Asia & Oceania]

NEC Electronics (China) Co., Ltd

7th Floor, Quantum Plaza, No. 27 ZhiChunLu Haidian
District, Beijing 100083, P.R.China
Tel: 010-8235-1155
<http://www.cn.necel.com/>

NEC Electronics Shanghai Ltd.

Room 2511-2512, Bank of China Tower,
200 Yincheng Road Central,
Pudong New Area, Shanghai P.R. China P.C:200120
Tel: 021-5888-5400
<http://www.cn.necel.com/>

NEC Electronics Hong Kong Ltd.

Unit 1601-1613, 16/F., Tower 2, Grand Century Place,
193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: 2886-9318
<http://www.hk.necel.com/>

NEC Electronics Taiwan Ltd.

7F, No. 363 Fu Shing North Road
Taipei, Taiwan, R. O. C.
Tel: 02-8175-9600
<http://www.tw.necel.com/>

NEC Electronics Singapore Pte. Ltd.

238A Thomson Road,
#12-08 Novena Square,
Singapore 307684
Tel: 6253-8311
<http://www.sg.necel.com/>

NEC Electronics Korea Ltd.

11F., Samik Lavied'or Bldg., 720-2,
Yeoksam-Dong, Kangnam-Ku,
Seoul, 135-080, Korea
Tel: 02-558-3737
<http://www.kr.necel.com/>