

To our customers,

---

## Old Company Name in Catalogs and Other Documents

---

On April 1<sup>st</sup>, 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 1<sup>st</sup>, 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

## V850E/IF3, V850E/IG3

32-bit Single-Chip Microcontrollers

Hardware

---

**V850E/IF3:**

***μ*PD70F3451**

***μ*PD70F3452**

**V850E/IG3:**

***μ*PD70F3453**

***μ*PD70F3454**

**[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 VIL (MAX) and VIH (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 VIL (MAX) and VIH (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 VDD or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.
- (3) **PRECAUTION AGAINST ESD:** A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.
- (4) **STATUS BEFORE INITIALIZATION:** Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.
- (5) **POWER ON/OFF SEQUENCE:** In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current. The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.
- (6) **INPUT OF SIGNAL DURING POWER OFF STATE :** Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

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

**EEPROM is a trademark of NEC Electronics Corporation.**

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

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

- The information in this document is current as of March, 2010. 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.

(Note 2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

(M8E0909E)

## PREFACE

**Readers** This manual is intended for users who wish to understand the functions of the V850E/IF3 ( $\mu$ PD70F3451, 70F3452) and V850E/IG3 ( $\mu$ PD70F3453, 70F3454) and design application systems using the V850E/IF3 and V850E/IG3.

**Purpose** This manual is intended to give users an understanding of the hardware functions of the V850E/IF3 and V850E/IG3 shown in the **Organization** below.

**Organization** This manual is divided into two parts: Hardware (this manual) and Architecture (**V850E1 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 V850E/IF3 and V850E/IG3  
→ Read this manual according to the **CONTENTS**.

To find the details of a register where the name is known  
→ See **APPENDIX B REGISTER INDEX**.

Register format

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

To understand the details of an instruction function  
→ Refer to the **V850E1 Architecture User's Manual**.

To know the electrical specifications of the V850E/IF3 and V850E/IG3  
→ See **CHAPTER 28 ELECTRICAL SPECIFICATIONS**.

The "yyy bit of the xxx register" is described as the "xxx.yyy bit" in this manual. Note with caution that even 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:                                     | $\overline{xxx}$ (overscore over pin or signal name)  |
| Memory map address:  | Higher addresses on the top and lower addresses on the bottom                                 |
| <b>Note:</b>   | Footnote for item marked with <b>Note</b> in the text   |
| <b>Caution:</b>  | Information requiring particular attention  |
| <b>Remark:</b>   | Supplementary information   |
| Numeric representation:  | Binary           ... xxxx or xxxxB<br>Decimal           ... xxxx<br>Hexadecimal     ... xxxxH |
| Prefix indicating power of 2 (address space, memory capacity): | K (kilo): $2^{10} = 1,024$<br>M (mega): $2^{20} = 1,024^2$<br>G (giga): $2^{30} = 1,024^3$    |
| Data type:   | Word       ... 32 bits<br>Halfword   ... 16 bits<br>Byte        ... 8 bits                    |



## Related Documents

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

### Documents related to V850E/IF3 and V850E/IG3

| Document Name   | Document No. |
|---|--------------|
| V850E1 Architecture User's Manual   | U14559E      |
| V850E/IF3, V850E/IG3 Hardware User's Manual   | This manual  |
| V850E/IF3, V850E/IG3 Sample Programs for Serial Communication (UARTA) Application Note                                      | U18723E      |
| V850E/IF3, V850E/IG3 Sample Programs for Serial Communication (UARTB) Application Note                                      | U18724E      |
| V850E/IF3, V850E/IG3 Sample Programs for Serial Communication (CSIB) Application Note                                       | U18725E      |
| V850E/IF3, V850E/IG3 Sample Programs for Serial Communication (I <sup>2</sup> C) Application Note                           | U18726E      |
| V850E/IF3, V850E/IG3 Sample Programs for DMA Function Application Note  | U18727E      |
| V850E/IF3, V850E/IG3 Sample Programs for Timer M Application Note   | U18728E      |
| V850E/IF3, V850E/IG3 Sample Programs for Watchdog Timer Application Note  | U18729E      |
| V850E/IF3, V850E/IG3 Sample Programs for Timer AA Application Note  | U18730E      |
| V850E/IF3, V850E/IG3 Sample Programs for Timer AB Application Note  | U18731E      |
| V850E/IF3, V850E/IG3 Sample Programs for Timer T Application Note   | U18732E      |
| V850E/IF3, V850E/IG3 Sample Programs for Port Function Application Note   | U18733E      |
| V850E/IF3, V850E/IG3 Sample Programs for Clock Generator Application Note   | U18734E      |
| V850E/IF3, V850E/IG3 Sample Programs for Standby Function Application Note  | U18735E      |
| V850E/IF3, V850E/IG3 Sample Programs for Interrupt Function Application Note  | U18736E      |
| V850E/IF3, V850E/IG3 Sample Programs for A/D Converters 0 and 1 Application Note  | U18737E      |
| V850E/IF3, V850E/IG3 Sample Programs for A/D Converter 2 Application Note   | U18738E      |
| V850E/IF3, V850E/IG3 Sample Programs for Low-Voltage Detector (LVI) Function Application Note                               | U18739E      |
| V850E/IF3, V850E/IG3 6-Phase PWM Output Control by Timer AB, Timer Q Option, Timer AA, A/D Converters 0, 1 Application Note | U18717E      |

**Documents related to development tools (user's manuals)**

| Document Name   |                     | Document No. |
|---|---------------------|--------------|
| QB-V850EIX3 In-Circuit Emulator                           |                     | U18651E      |
| QB-V850MINI On-Chip Debug Emulator                        |                     | U17638E      |
| QB-MINI2 On-Chip Debug Emulator with Programming Function |                     | U18371E      |
| QB-Programmer Programming GUI                             | Operation           | U18527E      |
| CA850 Ver. 3.20 C Compiler Package                        | Operation           | U18512E      |
|   | C Language          | U18513E      |
|   | Assembly Language   | U18514E      |
|   | Link Directives     | U18515E      |
| PM+ Ver. 6.30 Project Manager                             |                     | U18416E      |
| ID850QB Ver. 3.40 Integrated Debugger                     | Operation           | U18604E      |
| TW850 Ver. 2.00 Performance Analysis Tuning Tool          |                     | U17241E      |
| SM+ System Simulator                                      | Operation           | U18601E      |
|   | User Open Interface | U18212E      |
| RX850 Ver. 3.20 Real-Time OS                              | Basics              | U13430E      |
|   | Installation        | U17419E      |
|   | Technical           | U13431E      |
|   | Task Debugger       | U17420E      |
| RX850 Pro Ver. 3.21 Real-Time OS                          | Basics              | U18165E      |
|   | In-Structure        | U18164E      |
|   | Task Debugger       | U17422E      |
| AZ850 Ver. 3.30 System Performance Analyzer               |                     | U17423E      |
| PG-FP4 Flash Memory Programmer                            |                     | U15260E      |
| PG-FP5 Flash Memory Programmer                            |                     | U18865E      |

## CONTENTS

|   |           |
|---|-----------|
| <b>CHAPTER 1 INTRODUCTION .....</b>   | <b>20</b> |
| <b>1.1 Overview .....</b>   | <b>20</b> |
| <b>1.2 V850E/IF3 .....</b>  | <b>22</b> |
| 1.2.1 Features (V850E/IF3).....   | 22        |
| 1.2.2 Application fields (V850E/IF3) .....                                  | 24        |
| 1.2.3 Ordering information (V850E/IF3) .....                                | 24        |
| 1.2.4 Pin configuration (V850E/IF3) .....                                   | 25        |
| 1.2.5 Function blocks (V850E/IF3) .....                                     | 27        |
| <b>1.3 V850E/IG3.....</b>   | <b>30</b> |
| 1.3.1 Features (V850E/IG3) .....  | 30        |
| 1.3.2 Application fields (V850E/IG3).....                                   | 32        |
| 1.3.3 Ordering information (V850E/IG3).....                                 | 32        |
| 1.3.4 Pin configuration (V850E/IG3).....                                    | 33        |
| 1.3.5 Function blocks (V850E/IG3).....                                      | 38        |
| <b>CHAPTER 2 PIN FUNCTIONS.....</b>   | <b>42</b> |
| <b>2.1 List of Pin Functions .....</b>                                      | <b>42</b> |
| <b>2.2 Pin Status.....</b>  | <b>54</b> |
| <b>2.3 Pin I/O Circuits and Recommended Connection of Unused Pins .....</b> | <b>55</b> |
| <b>2.4 Pin I/O Circuits .....</b>   | <b>59</b> |
| <b>CHAPTER 3 CPU FUNCTION.....</b>  | <b>60</b> |
| <b>3.1 Features .....</b>   | <b>60</b> |
| <b>3.2 CPU Register Set .....</b>   | <b>61</b> |
| 3.2.1 Program register set.....   | 62        |
| 3.2.2 System register set.....  | 63        |
| <b>3.3 Operating Modes.....</b>   | <b>69</b> |
| 3.3.1 Operating modes.....  | 69        |
| 3.3.2 Operating mode specification .....                                    | 69        |
| <b>3.4 Address Space .....</b>  | <b>70</b> |
| 3.4.1 CPU address space .....   | 70        |
| 3.4.2 Image .....   | 71        |
| 3.4.3 Wraparound of CPU address space.....                                  | 72        |
| 3.4.4 Memory map .....  | 73        |
| 3.4.5 Area.....   | 74        |
| 3.4.6 Recommended use of address space .....                                | 77        |
| 3.4.7 On-chip peripheral I/O registers .....                                | 79        |
| 3.4.8 Special registers.....  | 93        |
| 3.4.9 System wait control register (VSWC) .....                             | 97        |
| <b>CHAPTER 4 PORT FUNCTIONS.....</b>  | <b>98</b> |
| <b>4.1 Features .....</b>   | <b>98</b> |
| 4.1.1 V850E/IF3 .....   | 98        |

|            |   |            |
|------------|---|------------|
| 4.1.2      | V850E/IG3 .....   | 98         |
| <b>4.2</b> | <b>Port Configuration.....</b>  | <b>99</b>  |
| 4.2.1      | V850E/IF3.....  | 99         |
| 4.2.2      | V850E/IG3.....  | 100        |
| <b>4.3</b> | <b>Port Configuration.....</b>  | <b>101</b> |
| 4.3.1      | Port 0.....   | 106        |
| 4.3.2      | Port 1.....   | 113        |
| 4.3.3      | Port 2.....   | 119        |
| 4.3.4      | Port 3.....   | 125        |
| 4.3.5      | Port 4.....   | 131        |
| 4.3.6      | Port 7.....   | 137        |
| 4.3.7      | Port DL .....   | 139        |
| <b>4.4</b> | <b>Output Data and Port Read Value for Each Setting.....</b>            | <b>144</b> |
| <b>4.5</b> | <b>Port Register Settings When Alternate Function Is Used.....</b>      | <b>157</b> |
| <b>4.6</b> | <b>Noise Eliminator .....</b>   | <b>166</b> |
| <b>4.7</b> | <b>Cautions .....</b>   | <b>172</b> |
| 4.7.1      | Cautions on setting port pins .....                                     | 172        |
| 4.7.2      | Cautions on bit manipulation instruction for port n register (Pn) ..... | 173        |

## **CHAPTER 5 CLOCK GENERATOR .....** 174

|            |                                |            |
|------------|--------------------------------|------------|
| <b>5.1</b> | <b>Overview .....</b>          | <b>174</b> |
| <b>5.2</b> | <b>Configuration.....</b>      | <b>175</b> |
| <b>5.3</b> | <b>Control Registers .....</b> | <b>178</b> |
| <b>5.4</b> | <b>PLL Function .....</b>      | <b>184</b> |
| 5.4.1      | Overview.....                  | 184        |
| 5.4.2      | PLL mode .....                 | 184        |
| 5.4.3      | Clock-through mode .....       | 184        |
| <b>5.5</b> | <b>Operation .....</b>         | <b>185</b> |
| 5.5.1      | Operation of each clock.....   | 185        |
| 5.5.2      | Clock output function .....    | 185        |
| 5.5.3      | Operation timing .....         | 186        |
| <b>5.6</b> | <b>Clock Monitor .....</b>     | <b>189</b> |

## **CHAPTER 6 16-BIT TIMER/EVENT COUNTER AA (TAA).....** 190

|            |  |            |
|------------|--|------------|
| <b>6.1</b> | <b>Overview .....</b>  | <b>190</b> |
| <b>6.2</b> | <b>Functions .....</b>   | <b>191</b> |
| <b>6.3</b> | <b>Configuration.....</b>  | <b>192</b> |
| <b>6.4</b> | <b>Registers .....</b>   | <b>199</b> |
| <b>6.5</b> | <b>Timer Output Operations.....</b>  | <b>212</b> |
| <b>6.6</b> | <b>Operation .....</b>   | <b>213</b> |
| 6.6.1      | Interval timer mode (TAA <sub>n</sub> MD2 to TAA <sub>n</sub> MD0 bits = 000) .....                | 221        |
| 6.6.2      | External event count mode (TAA <sub>n</sub> MD2 to TAA <sub>n</sub> MD0 bits = 001) .....          | 233        |
| 6.6.3      | External trigger pulse output mode (TAA <sub>n</sub> MD2 to TAA <sub>n</sub> MD0 bits = 010) ..... | 244        |
| 6.6.4      | One-shot pulse output mode (TAA <sub>n</sub> MD2 to TAA <sub>n</sub> MD0 bits = 011).....          | 256        |
| 6.6.5      | PWM output mode (TAA <sub>n</sub> MD2 to TAA <sub>n</sub> MD0 bits = 100) .....                    | 263        |
| 6.6.6      | Free-running timer mode (TAA <sub>n</sub> MD2 to TAA <sub>n</sub> MD0 bits = 101) .....            | 272        |
| 6.6.7      | Pulse width measurement mode (TAA <sub>n</sub> MD2 to TAA <sub>n</sub> MD0 bits = 110).....        | 288        |

|   |            |
|---|------------|
| <b>CHAPTER 7 16-BIT TIMER/EVENT COUNTER AB (TAB)</b> .....                    | <b>294</b> |
| <b>7.1 Overview</b> .....   | <b>294</b> |
| <b>7.2 Functions</b> .....  | <b>294</b> |
| <b>7.3 Configuration</b> .....  | <b>295</b> |
| <b>7.4 Registers</b> .....  | <b>297</b> |
| <b>7.5 Timer Output Operations</b> .....                                      | <b>312</b> |
| <b>7.6 Operation</b> .....  | <b>313</b> |
| 7.6.1 Interval timer mode (TABnMD2 to TABnMD0 bits = 000).....                | 321        |
| 7.6.2 External event count mode (TABnMD2 to TABnMD0 bits = 001).....          | 333        |
| 7.6.3 External trigger pulse output mode (TABnMD2 to TABnMD0 bits = 010)..... | 344        |
| 7.6.4 One-shot pulse output mode (TABnMD2 to TABnMD0 bits = 011) .....        | 357        |
| 7.6.5 PWM output mode (TABnMD2 to TABnMD0 bits = 100).....                    | 366        |
| 7.6.6 Free-running timer mode (TABnMD2 to TABnMD0 bits = 101) .....           | 377        |
| 7.6.7 Pulse width measurement mode (TABnMD2 to TABnMD0 bits = 110) .....      | 396        |
| <br>  |            |
| <b>CHAPTER 8 16-BIT TIMER/EVENT COUNTER T (TMT)</b> .....                     | <b>402</b> |
| <b>8.1 Overview</b> .....   | <b>402</b> |
| <b>8.2 Functions</b> .....  | <b>403</b> |
| <b>8.3 Configuration</b> .....  | <b>404</b> |
| <b>8.4 Registers</b> .....  | <b>408</b> |
| <b>8.5 Timer Output Operations</b> .....                                      | <b>428</b> |
| <b>8.6 Operation</b> .....  | <b>429</b> |
| 8.6.1 Interval timer mode (TTnMD3 to TTnMD0 bits = 0000).....                 | 437        |
| 8.6.2 External event count mode (TTmMD3 to TTmMD0 bits = 0001) .....          | 446        |
| 8.6.3 External trigger pulse output mode (TTmMD3 to TTmMD0 bits = 0010).....  | 457        |
| 8.6.4 One-shot pulse output mode (TTmMD3 to TTmMD0 bits = 0011) .....         | 470        |
| 8.6.5 PWM output mode (TTmMD3 to TTmMD0 bits = 0100).....                     | 477        |
| 8.6.6 Free-running timer mode (TTnMD3 to TTnMD0 bits = 0101) .....            | 486        |
| 8.6.7 Pulse width measurement mode (TTmMD3 to TTmMD0 bits = 0110).....        | 502        |
| 8.6.8 Triangular-wave PWM output mode (TTmMD3 to TTmMD0 bits = 0111) .....    | 508        |
| 8.6.9 Encoder count function.....   | 511        |
| 8.6.10 Encoder compare mode (TTmMD3 to TTmMD0 bits = 1000) .....              | 527        |
| <br>  |            |
| <b>CHAPTER 9 16-BIT INTERVAL TIMER M (TMM)</b> .....                          | <b>535</b> |
| <b>9.1 Overview</b> .....   | <b>535</b> |
| <b>9.2 Configuration</b> .....  | <b>536</b> |
| <b>9.3 Control Register</b> .....   | <b>537</b> |
| <b>9.4 Operation</b> .....  | <b>538</b> |
| 9.4.1 Interval timer mode.....  | 538        |
| <b>9.5 Cautions</b> .....   | <b>542</b> |
| <br>  |            |
| <b>CHAPTER 10 MOTOR CONTROL FUNCTION</b> .....                                | <b>543</b> |
| <b>10.1 Functional Overview</b> .....   | <b>543</b> |
| <b>10.2 Configuration</b> .....   | <b>544</b> |
| <b>10.3 Control Registers</b> .....   | <b>548</b> |
| <b>10.4 Operation</b> .....   | <b>562</b> |

|  |   |            |
|--|---|------------|
| 10.4.1   | System outline .....  | 562        |
| 10.4.2   | Dead-time control (generation of negative-phase wave signal) .....  | 567        |
| 10.4.3   | Interrupt culling function.....   | 574        |
| 10.4.4   | Operation to rewrite register with transfer function .....  | 581        |
| 10.4.5   | TAA <sub>n</sub> tuning operation for A/D conversion start trigger signal output .....  | 599        |
| 10.4.6   | A/D conversion start trigger output function.....   | 602        |
| <b>CHAPTER 11 WATCHDOG TIMER FUNCTIONS .....</b> |   | <b>607</b> |
| 11.1   | <b>Functions .....</b>  | <b>607</b> |
| 11.2   | <b>Configuration.....</b>   | <b>607</b> |
| 11.3   | <b>Control Registers .....</b>  | <b>608</b> |
| 11.4   | <b>Operation .....</b>  | <b>609</b> |
| 11.5   | <b>Caution .....</b>  | <b>609</b> |
| <b>CHAPTER 12 A/D CONVERTERS 0 AND 1 .....</b>   |   | <b>610</b> |
| 12.1   | <b>Features .....</b>   | <b>610</b> |
| 12.2   | <b>Configuration.....</b>   | <b>612</b> |
| 12.3   | <b>Control Registers .....</b>  | <b>622</b> |
| 12.4   | <b>Operation .....</b>  | <b>656</b> |
| 12.4.1   | Basic operation .....   | 656        |
| 12.4.2   | Input voltage and conversion result .....   | 658        |
| 12.4.3   | Operation mode.....   | 660        |
| 12.4.4   | Operation setting .....   | 660        |
| 12.4.5   | Operation of 1-channel conversion .....   | 661        |
| 12.4.6   | Operation of multiple channel conversion.....   | 662        |
| 12.4.7   | A/D trigger mode (normal operation mode) .....  | 664        |
| 12.4.8   | A/D trigger polling mode (normal operation mode) .....  | 666        |
| 12.4.9   | Hardware trigger mode (normal operation mode).....  | 668        |
| 12.4.10  | Conversion channel specification mode (extension operation mode).....   | 670        |
| 12.4.11  | Extension buffer mode (extension operation mode) .....  | 672        |
| 12.5   | <b>Internal Equivalent Circuit.....</b>   | <b>678</b> |
| 12.6   | <b>Cautions .....</b>   | <b>680</b> |
| 12.6.1   | Stopping conversion operation .....   | 680        |
| 12.6.2   | Interval of trigger during conversion operation in hardware trigger mode, conversion channel specification mode, and extension buffer mode..... | 680        |
| 12.6.3   | Writing to AD <sub>n</sub> SCM register.....  | 680        |
| 12.6.4   | A/D conversion start timing.....  | 680        |
| 12.6.5   | Operation in standby mode.....  | 681        |
| 12.6.6   | Timing of accepting trigger in conversion channel specification mode and extension buffer mode .....  | 681        |
| 12.6.7   | Variation of A/D conversion results.....  | 681        |
| 12.6.8   | A/D conversion result hysteresis characteristics.....   | 681        |
| 12.6.9   | A/D conversion trigger interval for continuous conversion .....   | 682        |
| 12.7   | <b>How to Read A/D Converter Characteristics Table .....</b>  | <b>683</b> |
| <b>CHAPTER 13 A/D CONVERTER 2 .....</b>          |   | <b>687</b> |
| 13.1   | <b>Features .....</b>   | <b>687</b> |

|  |   |            |
|--|---|------------|
| <b>13.2</b>  | <b>Configuration .....</b>  | <b>688</b> |
| <b>13.3</b>  | <b>Control Registers.....</b>                                       | <b>691</b> |
| <b>13.4</b>  | <b>Operation .....</b>  | <b>697</b> |
| 13.4.1   | Basic operation.....  | 697        |
| 13.4.2   | Trigger mode.....   | 699        |
| 13.4.3   | Operation mode.....   | 700        |
| <b>13.5</b>  | <b>Operation in Software Trigger Mode.....</b>                      | <b>707</b> |
| <b>13.6</b>  | <b>Internal Equivalent Circuit .....</b>                            | <b>711</b> |
| <b>13.7</b>  | <b>Cautions.....</b>  | <b>713</b> |
| <b>13.8</b>  | <b>How to Read A/D Converter Characteristics Table .....</b>        | <b>716</b> |
| <br><b>CHAPTER 14 ASYNCHRONOUS SERIAL INTERFACE A (UARTA).....</b> |   | <b>717</b> |
| <b>14.1</b>  | <b>Mode Switching Between UARTA and Other Serial Interface.....</b> | <b>717</b> |
| 14.1.1   | Mode switching between UARTA0 and CSIB0.....                        | 717        |
| 14.1.2   | Mode switching between UARTA1 and I <sup>2</sup> C.....             | 718        |
| 14.1.3   | Mode switching between UARTA2 and CSIB1 .....                       | 719        |
| <b>14.2</b>  | <b>Features .....</b>   | <b>720</b> |
| <b>14.3</b>  | <b>Configuration .....</b>  | <b>721</b> |
| <b>14.4</b>  | <b>Control Registers.....</b>                                       | <b>723</b> |
| <b>14.5</b>  | <b>Interrupt Request Signals .....</b>                              | <b>728</b> |
| <b>14.6</b>  | <b>Operation .....</b>  | <b>729</b> |
| 14.6.1   | Data format .....   | 729        |
| 14.6.2   | UART transmission .....   | 731        |
| 14.6.3   | Continuous transmission procedure .....                             | 732        |
| 14.6.4   | UART reception.....   | 734        |
| 14.6.5   | Reception errors.....   | 735        |
| 14.6.6   | Parity types and operations.....                                    | 736        |
| 14.6.7   | Receive data noise filter .....                                     | 737        |
| <b>14.7</b>  | <b>Dedicated Baud Rate Generator.....</b>                           | <b>738</b> |
| <b>14.8</b>  | <b>Cautions.....</b>  | <b>745</b> |
| <br><b>CHAPTER 15 ASYNCHRONOUS SERIAL INTERFACE B (UARTB).....</b> |   | <b>746</b> |
| <b>15.1</b>  | <b>Mode Switching Between UARTB and CSIB2.....</b>                  | <b>746</b> |
| <b>15.2</b>  | <b>Features .....</b>   | <b>747</b> |
| <b>15.3</b>  | <b>Configuration .....</b>  | <b>748</b> |
| <b>15.4</b>  | <b>Control Registers.....</b>                                       | <b>752</b> |
| <b>15.5</b>  | <b>Interrupt Request Signals .....</b>                              | <b>768</b> |
| <b>15.6</b>  | <b>Control Modes .....</b>  | <b>771</b> |
| <b>15.7</b>  | <b>Operation .....</b>  | <b>775</b> |
| 15.7.1   | Data format .....   | 775        |
| 15.7.2   | Transmit operation .....  | 776        |
| 15.7.3   | Continuous transmission operation .....                             | 779        |
| 15.7.4   | Receive operation .....   | 780        |
| 15.7.5   | Reception error.....  | 783        |
| 15.7.6   | Parity types and corresponding operation .....                      | 784        |
| 15.7.7   | Receive data noise filter .....                                     | 785        |
| <b>15.8</b>  | <b>Dedicated Baud Rate Generator (BRG) .....</b>                    | <b>786</b> |

|                   |   |            |
|-------------------|---|------------|
| <b>15.9</b>       | <b>Control Flow .....</b>   | <b>793</b> |
| <b>15.10</b>      | <b>Cautions .....</b>   | <b>804</b> |
| <br>              |   |            |
| <b>CHAPTER 16</b> | <b>CLOCKED SERIAL INTERFACE B (CSIB).....</b>                                   | <b>806</b> |
| <b>16.1</b>       | <b>Mode Switching Between CSIB and Other Serial Interface .....</b>             | <b>806</b> |
| 16.1.1            | Mode switching between CSIB0 and UARТА0 .....                                   | 806        |
| 16.1.2            | Mode switching between CSIB1 and UARТА2 .....                                   | 807        |
| 16.1.3            | Mode switching between CSIB2 and UARТB .....                                    | 808        |
| <b>16.2</b>       | <b>Features .....</b>   | <b>809</b> |
| <b>16.3</b>       | <b>Configuration.....</b>   | <b>810</b> |
| <b>16.4</b>       | <b>Control Registers .....</b>  | <b>812</b> |
| <b>16.5</b>       | <b>Operation .....</b>  | <b>819</b> |
| 16.5.1            | Single transfer mode (master mode, transmission mode) .....                     | 819        |
| 16.5.2            | Single transfer mode (master mode, reception mode).....                         | 821        |
| 16.5.3            | Single transfer mode (master mode, transmission/reception mode).....            | 823        |
| 16.5.4            | Single transfer mode (slave mode, transmission mode) .....                      | 825        |
| 16.5.5            | Single transfer mode (slave mode, reception mode) .....                         | 827        |
| 16.5.6            | Single transfer mode (slave mode, transmission/reception mode) .....            | 829        |
| 16.5.7            | Continuous transfer mode (master mode, transmission mode) .....                 | 831        |
| 16.5.8            | Continuous transfer mode (master mode, reception mode).....                     | 833        |
| 16.5.9            | Continuous transfer mode (master mode, transmission/reception mode).....        | 836        |
| 16.5.10           | Continuous transfer mode (slave mode, transmission mode).....                   | 840        |
| 16.5.11           | Continuous transfer mode (slave mode, reception mode) .....                     | 842        |
| 16.5.12           | Continuous transfer mode (slave mode, transmission/reception mode) .....        | 845        |
| 16.5.13           | Reception error .....   | 849        |
| 16.5.14           | Clock timing .....  | 850        |
| <b>16.6</b>       | <b>Output Pins .....</b>  | <b>852</b> |
| <br>              |   |            |
| <b>CHAPTER 17</b> | <b>I<sup>2</sup>C BUS .....</b>   | <b>853</b> |
| <b>17.1</b>       | <b>Mode Switching Between I<sup>2</sup>C and UARТА1 .....</b>                   | <b>853</b> |
| <b>17.2</b>       | <b>Features .....</b>   | <b>854</b> |
| <b>17.3</b>       | <b>Configuration.....</b>   | <b>857</b> |
| <b>17.4</b>       | <b>Registers .....</b>  | <b>859</b> |
| <b>17.5</b>       | <b>Functions .....</b>  | <b>873</b> |
| 17.5.1            | Pin configuration .....   | 873        |
| <b>17.6</b>       | <b>I<sup>2</sup>C Bus Definitions and Control Methods .....</b>                 | <b>874</b> |
| 17.6.1            | Start condition.....  | 874        |
| 17.6.2            | Addresses.....  | 875        |
| 17.6.3            | Transfer direction specification .....  | 876        |
| 17.6.4            | $\overline{\text{ACK}}$ .....   | 877        |
| 17.6.5            | Stop condition .....  | 878        |
| 17.6.6            | Wait state .....  | 879        |
| 17.6.7            | Wait state cancellation method.....   | 881        |
| <b>17.7</b>       | <b>I<sup>2</sup>C Interrupt Request Signals (INTIIC).....</b>                   | <b>882</b> |
| 17.7.1            | Master device operation .....   | 883        |
| 17.7.2            | Slave device operation (when receiving slave address data (address match))..... | 886        |
| 17.7.3            | Slave device operation (when receiving extension code) .....                    | 890        |



|                   |   |            |
|-------------------|---|------------|
| 17.7.4            | Operation without communication .....   | 894        |
| 17.7.5            | Arbitration loss operation (operation as slave after arbitration loss) .....          | 895        |
| 17.7.6            | Operation when arbitration loss occurs (no communication after arbitration loss)..... | 897        |
| <b>17.8</b>       | <b>Interrupt Request Signal (INTIIC) Generation Timing and Wait Control .....</b>     | <b>904</b> |
| <b>17.9</b>       | <b>Address Match Detection Method.....</b>  | <b>905</b> |
| <b>17.10</b>      | <b>Error Detection.....</b>   | <b>905</b> |
| <b>17.11</b>      | <b>Extension Code.....</b>  | <b>905</b> |
| <b>17.12</b>      | <b>Arbitration.....</b>   | <b>906</b> |
| <b>17.13</b>      | <b>Wakeup Function .....</b>  | <b>907</b> |
| <b>17.14</b>      | <b>Communication Reservation .....</b>  | <b>908</b> |
| 17.14.1           | When communication reservation function is enabled (IICF0.IICRSV0 bit = 0).....       | 908        |
| 17.14.2           | When communication reservation function is disabled (IICF0.IICRSV0 bit = 1) .....     | 911        |
| <b>17.15</b>      | <b>Cautions.....</b>  | <b>912</b> |
| <b>17.16</b>      | <b>Communication Operations.....</b>  | <b>913</b> |
| 17.16.1           | Master operation in single master system .....  | 914        |
| 17.16.2           | Master operation in multimaster system .....  | 915        |
| 17.16.3           | Slave operation .....   | 918        |
| <b>17.17</b>      | <b>Timing of Data Communication.....</b>  | <b>921</b> |
| <b>CHAPTER 18</b> | <b>BUS CONTROL FUNCTION.....</b>  | <b>928</b> |
| <b>18.1</b>       | <b>Features .....</b>   | <b>928</b> |
| <b>18.2</b>       | <b>Bus Control Pins.....</b>  | <b>929</b> |
| 18.2.1            | Pin status during internal ROM, internal RAM, and on-chip peripheral I/O access.....  | 929        |
| <b>18.3</b>       | <b>Memory Block Function .....</b>  | <b>930</b> |
| 18.3.1            | Chip select control function .....  | 930        |
| <b>18.4</b>       | <b>Bus Cycle Type Control Function .....</b>  | <b>931</b> |
| <b>18.5</b>       | <b>Bus Access.....</b>  | <b>932</b> |
| 18.5.1            | Number of access clocks .....   | 932        |
| 18.5.2            | Bus sizing function .....   | 933        |
| 18.5.3            | Endian function.....  | 934        |
| 18.5.4            | Bus width.....  | 934        |
| <b>18.6</b>       | <b>Wait Function .....</b>  | <b>941</b> |
| 18.6.1            | Programmable wait function .....  | 941        |
| 18.6.2            | External wait function .....  | 944        |
| 18.6.3            | Relationship between programmable wait and external wait.....                         | 944        |
| 18.6.4            | Bus cycles in which wait function is valid.....                                       | 945        |
| <b>18.7</b>       | <b>Idle State Insertion Function.....</b>   | <b>946</b> |
| <b>18.8</b>       | <b>Bus Timing.....</b>  | <b>948</b> |
| <b>18.9</b>       | <b>Bus Priority Order.....</b>  | <b>958</b> |
| <b>18.10</b>      | <b>Boundary Operation Conditions .....</b>  | <b>958</b> |
| 18.10.1           | Program space.....  | 958        |
| 18.10.2           | Data space.....   | 958        |
| <b>CHAPTER 19</b> | <b>DMA FUNCTIONS (DMA CONTROLLER).....</b>  | <b>959</b> |
| <b>19.1</b>       | <b>Features .....</b>   | <b>959</b> |
| <b>19.2</b>       | <b>Configuration .....</b>  | <b>960</b> |
| <b>19.3</b>       | <b>Control Registers.....</b>   | <b>961</b> |

|              |  |            |
|--------------|--|------------|
| 19.3.1       | DMA source address registers 0 to 3 (DSA0 to DSA3).....        | 961        |
| 19.3.2       | DMA destination address registers 0 to 3 (DDA0 to DDA3).....   | 963        |
| 19.3.3       | DMA transfer count registers 0 to 3 (DBC0 to DBC3).....        | 965        |
| 19.3.4       | DMA addressing control registers 0 to 3 (DADC0 to DADC3) ..... | 966        |
| 19.3.5       | DMA channel control registers 0 to 3 (DCHC0 to DCHC3).....     | 967        |
| 19.3.6       | DMA trigger factor registers 0 to 3 (DTFR0 to DTFR3) .....     | 969        |
| <b>19.4</b>  | <b>Transfer Modes.....</b>                                     | <b>973</b> |
| 19.4.1       | Single transfer mode .....                                     | 973        |
| 19.4.2       | Single-step transfer mode.....                                 | 975        |
| 19.4.3       | Block transfer mode.....                                       | 976        |
| <b>19.5</b>  | <b>Transfer Types.....</b>                                     | <b>977</b> |
| 19.5.1       | 2-cycle transfer .....   | 977        |
| <b>19.6</b>  | <b>Transfer Target.....</b>                                    | <b>977</b> |
| 19.6.1       | Transfer type and transfer target .....                        | 977        |
| <b>19.7</b>  | <b>DMA Channel Priorities .....</b>                            | <b>977</b> |
| <b>19.8</b>  | <b>Next Address Setting Function.....</b>                      | <b>978</b> |
| <b>19.9</b>  | <b>DMA Transfer Start Factors .....</b>                        | <b>979</b> |
| <b>19.10</b> | <b>Forcible Termination.....</b>                               | <b>980</b> |
| <b>19.11</b> | <b>Times Related to DMA Transfer.....</b>                      | <b>981</b> |
| <b>19.12</b> | <b>Cautions .....</b>  | <b>981</b> |
| <b>19.13</b> | <b>DMA Transfer End.....</b>                                   | <b>982</b> |

**CHAPTER 20 INTERRUPT/EXCEPTION PROCESSING FUNCTION..... 983**

|             |   |             |
|-------------|---|-------------|
| <b>20.1</b> | <b>Features .....</b>   | <b>983</b>  |
| <b>20.2</b> | <b>Non-Maskable Interrupts .....</b>  | <b>988</b>  |
| 20.2.1      | Operation.....  | 989         |
| 20.2.2      | Restore .....   | 991         |
| 20.2.3      | Non-maskable interrupt status flag (NP).....  | 992         |
| <b>20.3</b> | <b>Maskable Interrupts .....</b>  | <b>993</b>  |
| 20.3.1      | Operation.....  | 993         |
| 20.3.2      | Restore .....   | 995         |
| 20.3.3      | Priorities of maskable interrupts .....   | 996         |
| 20.3.4      | Interrupt control registers (xxICn) .....   | 1000        |
| 20.3.5      | Interrupt mask registers 0 to 5 (IMR0 to IMR5).....                                     | 1005        |
| 20.3.6      | In-service priority register (ISPR).....  | 1008        |
| 20.3.7      | Maskable interrupt status flag (ID).....  | 1009        |
| <b>20.4</b> | <b>External Interrupt Request Input Pins (INTP00 to INTP18, INTADT0, INTADT1) .....</b> | <b>1010</b> |
| 20.4.1      | Noise elimination .....   | 1010        |
| 20.4.2      | Edge detection.....   | 1010        |
| <b>20.5</b> | <b>Software Exception .....</b>   | <b>1015</b> |
| 20.5.1      | Operation.....  | 1015        |
| 20.5.2      | Restore.....  | 1016        |
| 20.5.3      | Exception status flag (EP) .....  | 1017        |
| <b>20.6</b> | <b>Exception Trap .....</b>   | <b>1017</b> |
| 20.6.1      | Illegal opcode definition .....   | 1017        |
| 20.6.2      | Debug trap.....   | 1019        |
| <b>20.7</b> | <b>Multiple Interrupt Servicing Control.....</b>  | <b>1021</b> |
| <b>20.8</b> | <b>Interrupt Response Time of CPU.....</b>  | <b>1023</b> |

|  |  |             |
|--|--|-------------|
| 20.9   | Periods in Which CPU Does Not Acknowledge Interrupts ..... | 1024        |
| 20.10  | Caution .....  | 1024        |
| <b>CHAPTER 21 STANDBY FUNCTION.....</b>        |  | <b>1025</b> |
| 21.1   | Overview .....   | 1025        |
| 21.2   | Control Registers.....                                     | 1027        |
| 21.3   | HALT Mode .....  | 1029        |
| 21.3.1   | Setting and operation status.....                          | 1029        |
| 21.3.2   | Releasing HALT mode .....                                  | 1029        |
| 21.4   | IDLE Mode.....   | 1031        |
| 21.4.1   | Setting and operation status.....                          | 1031        |
| 21.4.2   | Releasing IDLE mode .....                                  | 1031        |
| 21.5   | STOP Mode .....  | 1033        |
| 21.5.1   | Setting and operation status.....                          | 1033        |
| 21.5.2   | Releasing STOP mode.....                                   | 1033        |
| 21.6   | Securing Oscillation Stabilization Time .....              | 1035        |
| <b>CHAPTER 22 RESET FUNCTIONS.....</b>         |  | <b>1036</b> |
| 22.1   | Overview .....   | 1036        |
| 22.2   | Control Register.....                                      | 1036        |
| 22.3   | Operation .....  | 1038        |
| <b>CHAPTER 23 LOW-VOLTAGE DETECTOR.....</b>    |  | <b>1041</b> |
| 23.1   | Functions .....  | 1041        |
| 23.2   | Configuration .....  | 1041        |
| 23.3   | Control Registers.....                                     | 1042        |
| 23.4   | Operation .....  | 1044        |
| 23.4.1   | To use for internal reset signal .....                     | 1044        |
| 23.4.2   | To use for interrupt .....                                 | 1046        |
| <b>CHAPTER 24 POWER-ON CLEAR CIRCUIT .....</b> |  | <b>1047</b> |
| 24.1   | Function.....  | 1047        |
| 24.2   | Configuration .....  | 1047        |
| 24.3   | Operation .....  | 1048        |
| <b>CHAPTER 25 REGULATOR.....</b>               |  | <b>1049</b> |
| 25.1   | Overview .....   | 1049        |
| 25.2   | Operation .....  | 1050        |
| <b>CHAPTER 26 ON-CHIP DEBUG FUNCTION.....</b>  |  | <b>1051</b> |
| 26.1   | Debugging Using DCU .....                                  | 1052        |
| 26.1.1   | Circuit connection examples.....                           | 1052        |
| 26.1.2   | Interface signals .....                                    | 1056        |
| 26.1.3   | Maskable functions.....                                    | 1057        |
| 26.1.4   | Cautions .....   | 1058        |

|     |  |                 |
|-----|--|-----------------|
|     | <b>26.2 Debugging Without Using DCU .....</b>                      | <b>1059</b>     |
|     | 26.2.1 Circuit connection examples .....                           | 1059            |
|     | 26.2.2 Maskable functions .....                                    | 1062            |
|     | 26.2.3 Securing of user resources .....                            | 1062            |
|     | 26.2.4 Cautions .....  | 1068            |
|     | <b>26.3 ROM Security Function .....</b>                            | <b>1069</b>     |
|     | 26.3.1 Security ID .....   | 1069            |
|     | 26.3.2 Setting .....   | 1070            |
|     | <br><b>CHAPTER 27 FLASH MEMORY .....</b>                           | <br><b>1071</b> |
|     | <b>27.1 Features .....</b>   | <b>1071</b>     |
| <R> | <b>27.2 Memory Configuration .....</b>                             | <b>1072</b>     |
|     | <b>27.3 Functional Overview .....</b>                              | <b>1073</b>     |
|     | 27.3.1 Erase units .....   | 1075            |
|     | 27.3.2 Security function .....                                     | 1075            |
|     | <b>27.4 Writing with Flash Memory Programmer .....</b>             | <b>1077</b>     |
|     | <b>27.5 Flash Memory Programming Environment .....</b>             | <b>1078</b>     |
|     | <b>27.6 Communication Method of Flash Memory Programming .....</b> | <b>1079</b>     |
|     | <b>27.7 Pin Processing During Flash Memory Programming .....</b>   | <b>1085</b>     |
|     | 27.7.1 Power supply .....  | 1085            |
|     | 27.7.2 Pins used .....   | 1085            |
|     | 27.7.3 RESET pin .....   | 1088            |
|     | 27.7.4 FLMD0 and FLMD1 pins .....                                  | 1088            |
|     | 27.7.5 Port pins .....   | 1089            |
|     | 27.7.6 Other signal pins .....                                     | 1089            |
|     | <b>27.8 Flash Memory Programming Mode .....</b>                    | <b>1090</b>     |
|     | 27.8.1 Flash memory control .....                                  | 1090            |
|     | 27.8.2 Selection of communication mode .....                       | 1091            |
|     | 27.8.3 Communication commands .....                                | 1092            |
|     | <b>27.9 Rewriting by Self Programming .....</b>                    | <b>1094</b>     |
|     | 27.9.1 Overview .....  | 1094            |
|     | 27.9.2 Features .....  | 1095            |
|     | 27.9.3 Standard self programming flow .....                        | 1096            |
|     | 27.9.4 Flash functions .....                                       | 1097            |
|     | 27.9.5 Pin processing .....  | 1097            |
|     | 27.9.6 Internal resources used .....                               | 1098            |
|     | <br><b>CHAPTER 28 ELECTRICAL SPECIFICATIONS .....</b>              | <br><b>1099</b> |
|     | <b>28.1 V850E/IF3 .....</b>  | <b>1099</b>     |
|     | 28.1.1 Absolute maximum ratings .....                              | 1099            |
|     | 28.1.2 Capacitance .....   | 1100            |
|     | 28.1.3 Operating conditions .....                                  | 1100            |
|     | 28.1.4 Clock oscillator characteristics .....                      | 1101            |
|     | 28.1.5 Regulator characteristics .....                             | 1102            |
|     | 28.1.6 DC characteristics .....                                    | 1103            |
|     | 28.1.7 Data retention characteristics .....                        | 1105            |
|     | 28.1.8 AC characteristics .....                                    | 1106            |

|  |   |             |
|--|---|-------------|
| 28.1.9   | Characteristics of A/D converters 0 and 1 .....                                   | 1116        |
| 28.1.10  | Characteristics of A/D converter 2.....   | 1117        |
| 28.1.11  | Operational amplifier characteristics .....                                       | 1118        |
| 28.1.12  | Comparator characteristics.....   | 1119        |
| 28.1.13  | Power-on-clear circuit (POC).....   | 1120        |
| 28.1.14  | Low-voltage detector (LVI) .....  | 1121        |
| 28.1.15  | Flash memory programming characteristics.....                                     | 1122        |
| <b>28.2</b>  | <b>V850E/IG3.....</b>   | <b>1123</b> |
| 28.2.1   | Absolute maximum ratings .....  | 1123        |
| 28.2.2   | Capacitance .....   | 1124        |
| 28.2.3   | Operating conditions .....  | 1124        |
| 28.2.4   | Clock oscillator characteristics .....  | 1125        |
| 28.2.5   | Regulator characteristics .....   | 1126        |
| 28.2.6   | DC characteristics .....  | 1127        |
| 28.2.7   | Data retention characteristics .....  | 1129        |
| 28.2.8   | AC characteristics .....  | 1130        |
| 28.2.9   | Characteristics of A/D converters 0, 1 .....                                      | 1148        |
| 28.2.10  | Characteristics of A/D converter 2.....   | 1149        |
| 28.2.11  | Operational amplifier characteristics .....                                       | 1150        |
| 28.2.12  | Comparator characteristics.....   | 1151        |
| 28.2.13  | Power-on-clear circuit (POC).....   | 1152        |
| 28.2.14  | Low-voltage detector (LVI) .....  | 1153        |
| 28.2.15  | Flash memory programming characteristics.....                                     | 1154        |
| <b>CHAPTER 29 PACKAGE DRAWINGS.....</b>                  |   | <b>1155</b> |
| <b>CHAPTER 30 RECOMMENDED SOLDERING CONDITIONS .....</b> |   | <b>1159</b> |
| <b>APPENDIX A CAUTIONS.....</b>                          |   | <b>1161</b> |
| <b>A.1</b>   | <b>Restriction on Conflict Between sld Instruction and Interrupt Request.....</b> | <b>1161</b> |
| A.1.1  | Description .....   | 1161        |
| A.1.2  | Countermeasure.....   | 1161        |
| <b>APPENDIX B REGISTER INDEX.....</b>                    |   | <b>1162</b> |
| <b>APPENDIX C INSTRUCTION SET LIST.....</b>              |   | <b>1175</b> |
| <b>C.1</b>   | <b>Conventions .....</b>  | <b>1175</b> |
| <b>C.2</b>   | <b>Instruction Set (in Alphabetical Order).....</b>                               | <b>1178</b> |
| <b>APPENDIX D REVISION HISTORY .....</b>                 |   | <b>1185</b> |
| <b>D.1</b>   | <b>Major Revisions in This Edition.....</b>                                       | <b>1185</b> |
| <b>D.2</b>   | <b>Revision History up to Previous Edition .....</b>                              | <b>1186</b> |

## CHAPTER 1 INTRODUCTION

The V850E/IF3 and V850E/IG3 are products of the NEC Electronics V850 single-chip microcontrollers. This chapter gives an outline of the V850E/IF3 and V850E/IG3.

### 1.1 Overview

The V850E/IF3 and V850E/IG3 are 32-bit single-chip microcontrollers that use the V850E1 CPU core and incorporate ROM/RAM and various peripheral functions such as DMA controller, timer/counter, watchdog timer, serial interfaces, A/D converter, and on-chip debug function.

In addition to high real-time response characteristics and 1-clock-pitch basic instructions, the V850E/IF3 and V850E/IG3 feature instructions such as multiply instructions, saturated operation instructions, and bit manipulation instructions realized by a hardware multiplier, as optimum instructions for digital servo control applications. Moreover, as a real-time control system, the V850E/IF3 and V850E/IG3 enable an extremely high cost-performance for applications such as motor inverter control.

Table 1-1 lists the V850E/IF3 and V850E/IG3 products.

**Table 1-1. V850E/IF3, V850E/IG3 Product List**

| Function<br>Part Number |                 | Package | ROM          |        | RAM Size | Operating Frequency (MAX.) | Maskable Interrupt |          | Non-Maskable Interrupt |
|-------------------------|-----------------|---------|--------------|--------|----------|----------------------------|--------------------|----------|------------------------|
|                         |                 |         | Type         | Size   |          |                            | External           | Internal |                        |
| V850E/IF3               | $\mu$ PD70F3451 | 80GC    | Flash memory | 128 KB | 8 KB     | 64 MHz                     | 15                 | 73       | 1                      |
|                         | $\mu$ PD70F3452 | 80GC    |              | 256 KB | 12 KB    |                            |                    |          |                        |
| V850E/IG3               | $\mu$ PD70F3453 | 100GC   |              | 128 KB | 8 KB     |                            | 21                 | 74       |                        |
|                         | $\mu$ PD70F3453 | 100GF   |              | 128 KB | 8 KB     |                            |                    |          |                        |
|                         | $\mu$ PD70F3454 | 100GC   |              | 256 KB | 12 KB    |                            |                    |          |                        |
|                         | $\mu$ PD70F3454 | 100GF   |              | 256 KB | 12 KB    |                            |                    |          |                        |
|                         | $\mu$ PD70F3454 | 161F1   |              | 256 KB | 12 KB    |                            |                    |          |                        |

- Remarks**
- 80GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)  
 100GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)  
 100GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)  
 161F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)
  - The part numbers of the V850E/IG3 are shown as follows in this manual.
    - GF versions  
 $\mu$ PD70F3453GF-GAS-AX, 70F3454GF-GAS-AX

Table 1-2 shows the differences in functions between the V850E/IF3 and V850E/IG3.

**Table 1-2. Differences in Functions Between V850E/IF3 and V850E/IG3**

| Item                          |                                   | V850E/IF3   | V850E/IG3  |
|-------------------------------|-----------------------------------|---|--|
| Port function                 | I/O                               | 44  | 56   |
|                               | Input                             | 4   | 8  |
|                               | On-chip pull-up resistor          | 44  | 56   |
| Interrupt source              |                                   | External interrupt: 15<br>Internal interrupt: 74  | External interrupt: 21<br>Internal interrupt: 75   |
| External bus function         |                                   | None  | Provided ( $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only)                                       |
| Timers AA0 to AA4             |                                   | Timer AA0 (without I/O)<br>Timer AA1 (without I/O)<br>Timer AA2<br>Timer AA3 (without I/O)<br>Timer AA4 | Timer AA0 (without I/O)<br>Timer AA1 (without I/O)<br>Timer AA2<br>Timer AA3<br>Timer AA4          |
| Timers T0, T1                 |                                   | Timer T0 (without I/O)<br>Timer T1  | Timer T0<br>Timer T1   |
| Motor control function        | High-impedance output control pin | TOA2OFF<br>TOB0OFF<br>TOB1OFF   | TOA2OFF<br>TOA3OFF<br>TOB0OFF<br>TOB1OFF   |
| A/D converter 2               | Analog input                      | 4 channels  | 8 channels   |
| On-chip debug function        | On-chip debug emulator            | NIMICUBE2   | NIMICUBE<br>NIMICUBE2  |
| Power supply for external pin |                                   | EV <sub>DD0</sub> , EV <sub>DD1</sub>   | EV <sub>DD0</sub> to EV <sub>DD2</sub>   |
| Package                       |                                   | 80-pin plastic LQFP (14 × 14)   | 100-pin plastic LQFP (14 × 14)<br>100-pin plastic LQFP (14 × 20)<br>161-pin plastic FBGA (10 × 10) |

1.2 V850E/IF3

1.2.1 Features (V850E/IF3)

- Minimum instruction execution time:  
15.6 ns (at internal 64 MHz operation)
- General-purpose registers: 32 bits × 32
- CPU features:
  - Signed multiplication (16 bits × 16 bits → 32 bits or 32 bits × 32 bits → 64 bits):  
1 to 2 clocks
  - Saturated operation instructions (with overflow/underflow detection function)
  - 32-bit shift instructions: 1 clock
  - Bit manipulation instructions
  - Load/store instructions with long/short format
  - Signed load instructions

○ Internal memory:

| Part Number | Internal ROM          | Internal RAM |
|-------------|-----------------------|--------------|
| μPD70F3451  | 128 KB (flash memory) | 8 KB         |
| μPD70F3452  | 256 KB (flash memory) | 12 KB        |

- On-chip debug function: Supports MINICUBE<sup>®</sup>2.
- Interrupts/exceptions:
  - Non-maskable interrupts: 1 source (external: none, internal: 1)
  - Maskable interrupts: 88 sources (external: 15, internal: 73)
  - Software exceptions: 32 sources
  - Exception traps: 2 sources
- DMA controller:
  - 4 channels
  - Transfer unit: 8 bits/16 bits
  - Maximum transfer count: 65536 (2<sup>16</sup>)
  - Transfer type: 2-cycle
  - Transfer modes: Single/single step/block
  - Transfer targets: On-chip peripheral I/O ↔ Internal RAM  
On-chip peripheral I/O ↔ On-chip peripheral I/O
  - Transfer request: On-chip peripheral I/O/software
  - Next address setting function
- I/O lines: Total: 48 (Input ports: 4, I/O ports: 44)



- Timer/counter function:
  - 16-bit interval timer M (TMM): 4 channels
  - 16-bit timer/event counter AA (TAA): 5 channels
  - 16-bit timer/event counter AB (TAB): 2 channels
  - 16-bit timer/event counter T (TMT): 2 channels
  - Motor control function (uses timer TAB: 2 channels (TAB0, TAB1), TAA: 2 channels (TAA0, TAA1))
    - 16-bit accuracy 6-phase PWM function with deadtime: 2 channels
    - High-impedance output control function
    - A/D trigger generation by timer tuning operation function
    - Arbitrary cycle setting function
    - Arbitrary deadtime setting function
  - Watchdog timer: 1 channel
  
- Serial interfaces:
  - Asynchronous serial interface A (UARTA)
  - Asynchronous serial interface B (UARTB)
  - Clocked serial interface B (CSIB)
  - I<sup>2</sup>C bus interface (I<sup>2</sup>C)
  
  - UARTA0/CSIB0: 1 channel
  - UARTA1/I<sup>2</sup>C: 1 channel
  - UARTA2/CSIB1: 1 channel
  - UARTB/CSIB2: 1 channel
  
- A/D converter:
  - 12-bit resolution A/D converters (A/D converters 0 and 1): 5 channels + 5 channels (2 units)
  - The one A/D converter 0 channel and three A/D converter 1 channels are provided with an operational amplifier for input level amplification and a comparator for overvoltage detection.
  - 10-bit resolution A/D converter (A/D converter 2): 4 channels
  
- Clock generator:
  - 4 to 8 MHz resonator connectable (external clock input prohibited)
  - Multiplication function by PLL clock synthesizer (fixed to multiplication by eight,  $f_{xx} = 32$  to 64 MHz)
  - CPU clock division function ( $f_{xx}$ ,  $f_{xx}/2$ ,  $f_{xx}/4$ ,  $f_{xx}/8$ )
  
- Power-save function: HALT/IDLE/STOP mode
  
- Power-on-clear function:
  
- Low-voltage detection function:
  
- Package: 80-pin plastic LQFP (14 × 14)
  
- Operation supply voltage:
  - $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0$  to 5.5 V (when A/D converter 0, 1 or 2 is operating)
  - $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5$  to 5.5 V (when none of A/D converters 0 to 2 is operating)

### 1.2.2 Application fields (V850E/IF3)

- Consumer equipment (such as inverter air conditioners, washing machines, driers, refrigerators, etc.)
- Industrial equipment (such as motor control, general-purpose inverters, etc.)

### 1.2.3 Ordering information (V850E/IF3)

| Part Number             | Package                       | Internal ROM          |
|-------------------------|-------------------------------|-----------------------|
| $\mu$ PD70F3451GC-UBT-A | 80-pin plastic LQFP (14 × 14) | Flash memory (128 KB) |
| $\mu$ PD70F3452GC-UBT-A | 80-pin plastic LQFP (14 × 14) | Flash memory (256 KB) |

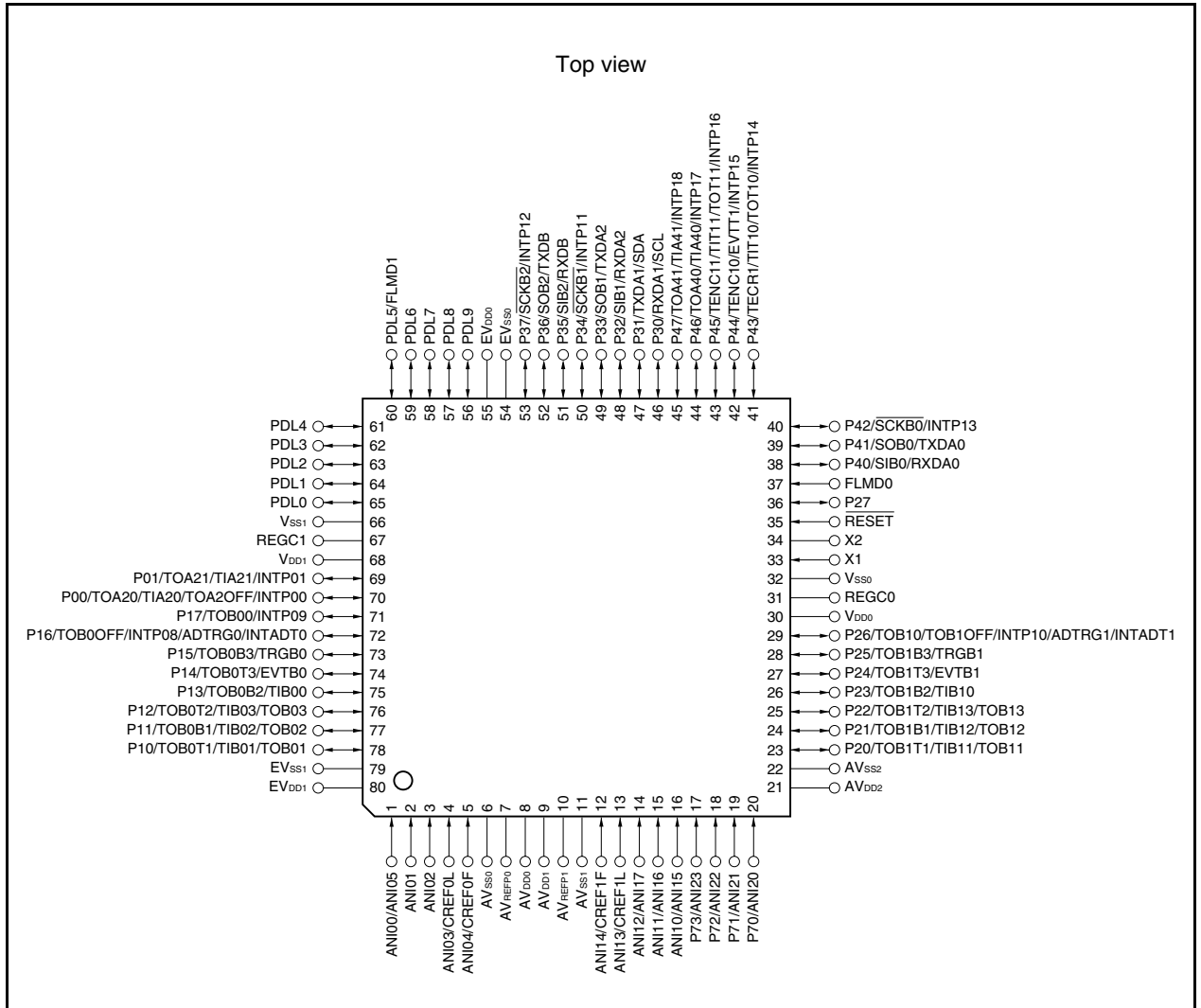
**Remark** The V850E/IF3 microcontrollers are lead-free products.

1.2.4 Pin configuration (V850E/IF3)

• 80-pin plastic LQFP (14 × 14)

μPD70F3451GC-UBT-A

μPD70F3452GC-UBT-A

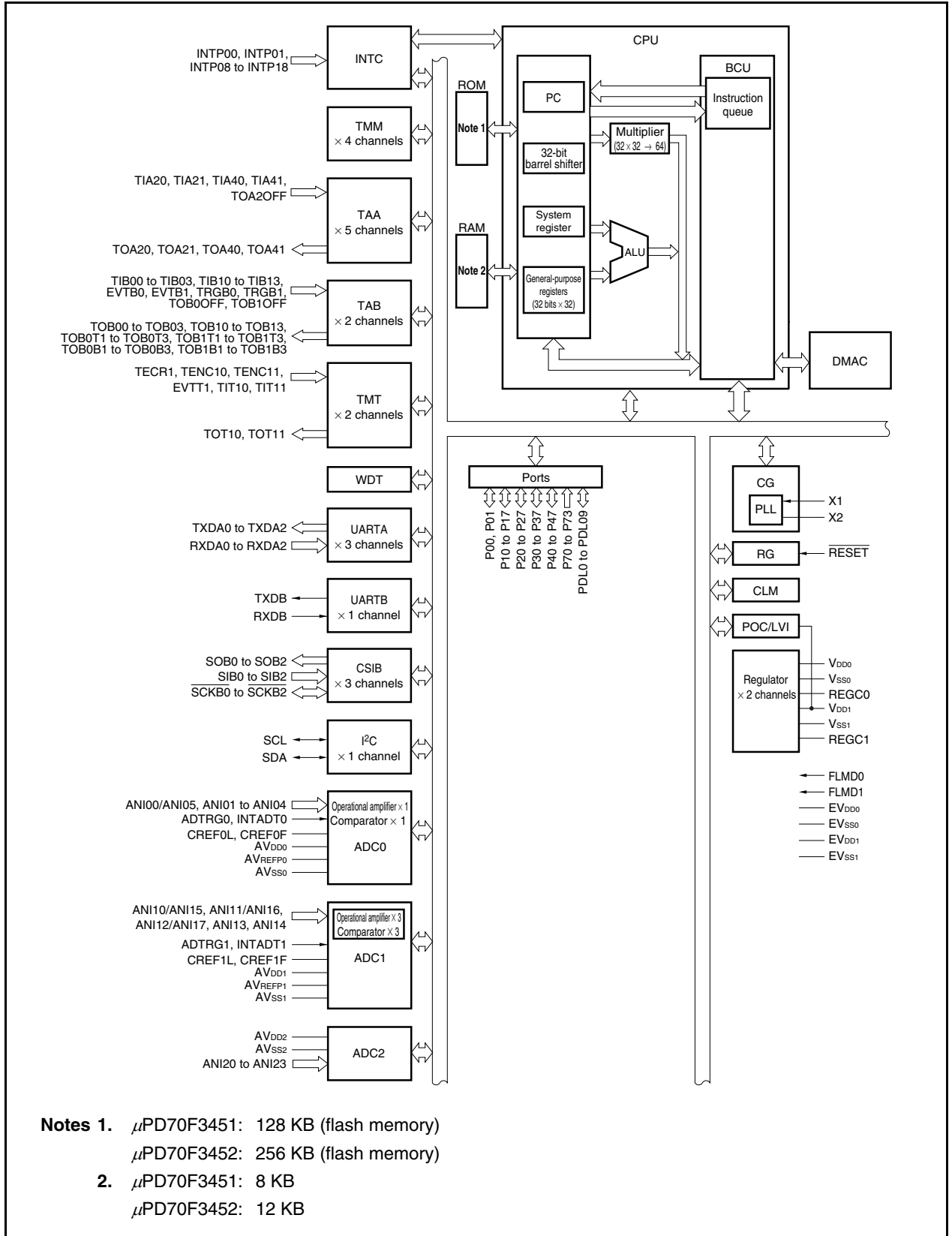


**Pin Identification (V850E/IF3)**

|   |                              |  |                           |
|---|------------------------------|--|---------------------------|
| ADTRG0, ADTRG1:   | A/D trigger input            | SCL:   | Serial clock              |
| ANI00 to ANI05,<br>ANI10 to ANI17,<br>ANI20 to ANI23:     | Analog input                 | SDA:   | Serial data               |
| AVDD0 to AVDD2:   | Analog power supply          | SIB0 to SIB2:  | Serial input              |
| AVREFP0, AVREFP1:   | Analog reference voltage     | SOB0 to SOB2:  | Serial output             |
| AVSS0 to AVSS2:   | Analog ground                | TECR1:   | Timer encoder clear input |
| CREFOF, CREF1F,<br>CREF0L, CREF1L:                        | Comparator reference voltage | TENC10, TENC11:  | Timer encoder input       |
| EVDD0, EVDD1:   | Power supply for port        | TIA20, TIA21,<br>TIA40, TIA41,<br>TIB00 to TIB03,<br>TIB10 to TIB13,<br>TIT10, TIT11:  | Timer trigger input       |
| EVSS0, EVSS1:   | Ground for port              | TOA20, TOA21,<br>TOA40, TOA41,<br>TOB00 to TOB03,<br>TOB0B1 to TOB0B3,<br>TOB0T1 to TOB0T3,<br>TOB10 to TOB13,<br>TOB1B1 to TOB1B3,<br>TOB1T1 to TOB1T3, |                           |
| EVTB0, EVTB1,<br>EVTT1:                                   | Timer event count input      | TOT10, TOT11:  | Timer output              |
| FLMD0, FLMD1:   | Flash programming mode       | TOA2OFF,<br>TOB0OFF, TOB1OFF:  | Timer output off          |
| INTADT0, INTADT1,<br>INTP00, INTP01,<br>INTP08 to INTP18: | External interrupt input     | TRGB0, TRGB1:  | Timer trigger input       |
| P00, P01:   | Port 0                       | TXDA0 to TXDA2,<br>TXDB:   | Transmit data             |
| P10 to P17:   | Port 1                       | VDD0, VDD1:  | Power supply              |
| P20 to P27:   | Port 2                       | VSS0, VSS1:  | Ground                    |
| P30 to P37:   | Port 3                       | X1, X2:  | Clock oscillator pin      |
| P40 to P47:   | Port 4                       |  |                           |
| P70 to P73:   | Port 7                       |  |                           |
| PDL0 to PDL9:   | Port DL                      |  |                           |
| REGC0, REGC1:   | Regulator control            |  |                           |
| RESET:  | Reset                        |  |                           |
| RXDA0 to RXDA2,<br>RXDB:                                  | Receive data                 |  |                           |
| SCKB0 to SCKB2:   | Serial clock                 |  |                           |

1.2.5 Function blocks (V850E/IF3)

(1) Internal block diagram



**(2) Internal units**

**(a) 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 (32 bits × 32 bits → 64 bits) and a barrel shifter (32 bits), help accelerate complex processing.

**(b) Bus control unit (BCU)**

The BCU controls the internal bus.

**(i) DMA controller (DMAC)**

This controller controls data transfer between on-chip peripheral I/O and internal RAM or on-chip peripheral I/O instead of the CPU.

The transfer type is two-cycle transfer, and single transfer, single-step transfer, and block transfer are used in transfer mode.

**(c) ROM**

This is flash memory that is mapped from address 00000000H.

During instruction fetch, ROM/flash memory can be accessed from the CPU in 1-clock cycles. The internal ROM capacity and area differ as follows depending on the product.

| Part Number | Internal ROM Capacity | Internal ROM Area      |
|-------------|-----------------------|------------------------|
| μPD70F3451  | 128 KB (flash memory) | x0000000H to x001FFFFH |
| μPD70F3452  | 256 KB (flash memory) | x0000000H to x003FFFFH |

**(d) RAM**

The internal RAM capacity and area differ as follows depending on the product.

During instruction fetch or data access, data can be accessed from the CPU in 1-clock cycles.

| Part Number | Internal RAM Capacity | Internal RAM Area      |
|-------------|-----------------------|------------------------|
| μPD70F3451  | 8 KB                  | xFFFC000H to xFFFDFFFH |
| μPD70F3452  | 12 KB                 | xFFFC000H to xFFFEFFFH |

**(e) Interrupt controller (INTC)**

This controller handles hardware interrupt requests (INTP00, INTP01, INTP08 to INTP18, INTADT0, INTADT1) from on-chip peripheral hardware and external hardware. Eight levels of interrupt priorities can be specified for these interrupt requests, and multiple-interrupt servicing control can be performed.

**(f) Clock generator (CG)**

The clock generator includes two basic operation modes: PLL mode (fixed to multiplication by eight) and clock-through mode. It generates four types of clocks ( $f_{xx}$ ,  $f_{xx}/2$ ,  $f_{xx}/4$ ,  $f_{xx}/8$ ), and supplies one of them as the operating clock for the CPU ( $f_{CPU}$ ).

**(g) Timer/counter**

The V850E/IF3 incorporates four 16-bit interval timer M (TMM) channels, five 16-bit timer/event counter AA (TAA) channels, two 16-bit timer/event counter AB (TAB) channels, and two 16-bit timer/event counter T (TMT) channels, and can measure pulse interval widths or frequency, enable an inverter function for motor control, and output a programmable pulse.

**(h) Watchdog timer (WDT)**

A watchdog timer is equipped to detect program loops, system abnormalities, etc. It generates a non-maskable interrupt request signal (INTWDT) or internal reset signal (WDTRES) after an overflow occurs.

**(i) Serial interface**

The V850E/IF3 incorporates eight serial interface channels: for three asynchronous serial interface A (UARTA) channels, one asynchronous serial interface B (UARTB) channel, three clocked serial interface B (CSIB) channels, and one I<sup>2</sup>C bus interface (I<sup>2</sup>C) channel. Of these, UARTA0 and CSIB0, UARTA1 and I<sup>2</sup>C, UARTA2 and CSIB1, and UARTB and CSIB2 share a pin.

For UARTA, data is transferred via the TXDAn and RXDAn pins (n = 0 to 2).

For UARTB, data is transferred via the TXDB and RXDB pins.

For CSIB, data is transferred via the SOBn, SIBn, and  $\overline{\text{SCKBn}}$  pins (n = 0 to 2).

For I<sup>2</sup>C, data is transferred via the SCL and SDA pins.

**(j) A/D converter (ADC)**

One channel is provided for each of the high-speed, high-resolution 12-bit A/D converters (ADC0, ADC1) (total of two channels), which have five analog input pins respectively, and one channel is provided for the 10-bit A/D converter (ADC2), which has four analog input pins.

Both one of the ADC0 channels and three of the ADC1 channels include an operational amplifier and a comparator so that these A/D converters can amplify an analog input voltage and detect overvoltage input.

**(k) On-chip debug function**

An on-chip debug function supporting MINICUBE2 can be used, so that a simple, inexpensive debug environment can be organized.

**(l) Ports**

As shown below, the following ports have general-purpose port functions and control pin functions.

| Port    | I/O         | Alternate Function   |
|---------|-------------|--|
| Port 0  | 2-bit I/O   | Timer/counter I/O, external interrupt input  |
| Port 1  | 8-bit I/O   | Timer/counter I/O, external trigger input of A/D converter 0, external interrupt input |
| Port 2  | 8-bit I/O   | Timer/counter I/O, external trigger input of A/D converter 1, external interrupt input |
| Port 3  | 8-bit I/O   | Serial interface I/O, external interrupt input   |
| Port 4  | 8-bit I/O   | Serial interface I/O, timer/counter I/O, external interrupt input                      |
| Port 7  | 4-bit input | A/D converter 2 input  |
| Port DL | 10-bit I/O  | —  |

## 1.3 V850E/IG3

### 1.3.1 Features (V850E/IG3)

- Minimum instruction execution time:  
15.6 ns (at internal 64 MHz operation)
- General-purpose registers: 32 bits × 32
- CPU features:
  - Signed multiplication (16 bits × 16 bits → 32 bits or 32 bits × 32 bits → 64 bits):  
1 to 2 clocks
  - Saturated operation instructions (with overflow/underflow detection function)
  - 32-bit shift instructions: 1 clock
  - Bit manipulation instructions
  - Load/store instructions with long/short format
  - Signed load instructions
- Memory space ( $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only):
  - 256 MB of linear address space (program/data sharing)
  - Chip select output function: 2 spaces
  - Memory block division function: 2 MB/block
  - External bus interface:
    - Multiplexed bus mode: 16-bit address bus  
8-bit/16-bit data bus
    - Separate bus mode: 8-bit address bus  
8-bit/16-bit data bus
    - 8-bit/16-bit data bus sizing function
    - External bus frequency switch function: 32/16 MHz
    - Wait function
      - Programmable wait function
      - External wait function
    - Idle state function
    - Address setup wait function
- Internal memory:
 

| Part Number     | Internal ROM          | Internal RAM |
|-----------------|-----------------------|--------------|
| $\mu$ PD70F3453 | 128 KB (flash memory) | 8 KB         |
| $\mu$ PD70F3454 | 256 KB (flash memory) | 12 KB        |
- On-chip debug function: Supports MINICUBE and MINICUBE2.
- Interrupts/exceptions:
  - Non-maskable interrupts: 1 source (external: none, internal: 1)
  - Maskable interrupts: 95 sources (external: 21, internal: 74)
  - Software exceptions: 32 sources
  - Exception traps: 2 sources



- DMA controller:
  - 4 channels
  - Transfer unit: 8 bits/16 bits
  - Maximum transfer count: 65,536 ( $2^{16}$ )
  - Transfer type: 2-cycle
  - Transfer mode: Single/single step/block
  - Transfer target: On-chip peripheral I/O ↔ Internal RAM  
On-chip peripheral I/O ↔ On-chip peripheral I/O
  - Transfer request: On-chip peripheral I/O/software
  - Next address setting function
  
- I/O lines:
  - Total: 64 (input ports: 8, I/O ports: 56)
  
- Timer/counter function:
  - 16-bit interval timer M (TMM): 4 channels
  - 16-bit timer/event counter AA (TAA): 5 channels
  - 16-bit timer/event counter AB (TAB): 2 channels
  - 16-bit timer/event counter T (TMT): 2 channels
  - Motor control function (uses timer TAB: 2 channels (TAB0 and TAB1),  
TAA: 2 channels (TAA0 and TAA1))
    - 16-bit accuracy 6-phase PWM function with deadtime: 2 channels
    - High-impedance output control function
    - A/D trigger generation by timer tuning operation function
    - Arbitrary cycle setting function
    - Arbitrary deadtime setting function
  - Watchdog timer: 1 channel
  
- Serial interfaces:
  - Asynchronous serial interface A (UARTA)
  - Asynchronous serial interface B (UARTB)
  - Clocked serial interface B (CSIB)
  - I<sup>2</sup>C bus interface (I<sup>2</sup>C)
  
  - UARTA0/CSIB0: 1 channel
  - UARTA1/ I<sup>2</sup>C: 1 channel
  - UARTA2/CSIB1: 1 channel
  - UARTB/CSIB2: 1 channel
  
- A/D converter:
  - 12-bit resolution A/D converters (A/D converters 0 and 1): 5 channels + 5 channels  
(2 units)
  - The one A/D converter 0 channel and three A/D converter 1 channels are provided with an operational amplifier for input level amplification and a comparator for overvoltage detection.
  - 10-bit resolution A/D converter (A/D converter 2): 8 channels
  
- Clock generator:
  - 4 to 8 MHz resonator connectable (external clock input prohibited)
  - Multiplication function by PLL clock synthesizer (fixed to multiplication by eight,  $f_{xx} = 32$  to 64 MHz)
  - CPU clock division function ( $f_{xx}$ ,  $f_{xx}/2$ ,  $f_{xx}/4$ ,  $f_{xx}/8$ )

- Power-save function:       HALT/IDLE/STOP mode
  
- Power-on-clear function
  
- Low-voltage detection function
  
- Package:                   100-pin plastic LQFP (fine pitch) (14 × 14)  
                                   100-pin plastic LQFP (14 × 20)  
                                   161-pin plastic FBGA (10 × 10)
  
- Operation supply voltage:  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0$  to 5.5 V  
                                   (when A/D converter 0, 1 or 2 is operating)  
 $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5$  to 5.5 V  
                                   (when none of A/D converters 0 to 2 is operating)

### 1.3.2 Application fields (V850E/IG3)

- Consumer equipment (such as inverter air conditioners, washing machines, driers, refrigerators, etc.)
- Industrial equipment (such as motor control, general-purpose inverters, etc.)

### 1.3.3 Ordering information (V850E/IG3)

| Part Number              | Package                                     | Internal ROM          |
|--------------------------|---|-----------------------|
| $\mu$ PD70F3453GC-8EA-A  | 100-pin plastic LQFP (fine pitch) (14 × 14) | Flash memory (128 KB) |
| $\mu$ PD70F3453GF-GAS-AX | 100-pin plastic LQFP (14 × 20)              | Flash memory (128 KB) |
| $\mu$ PD70F3454GC-8EA-A  | 100-pin plastic LQFP (fine pitch) (14 × 14) | Flash memory (256 KB) |
| $\mu$ PD70F3454GF-GAS-AX | 100-pin plastic LQFP (14 × 20)              | Flash memory (256 KB) |
| $\mu$ PD70F3454F1-DA9-A  | 161-pin plastic FBGA (10 × 10)              | Flash memory (256 KB) |

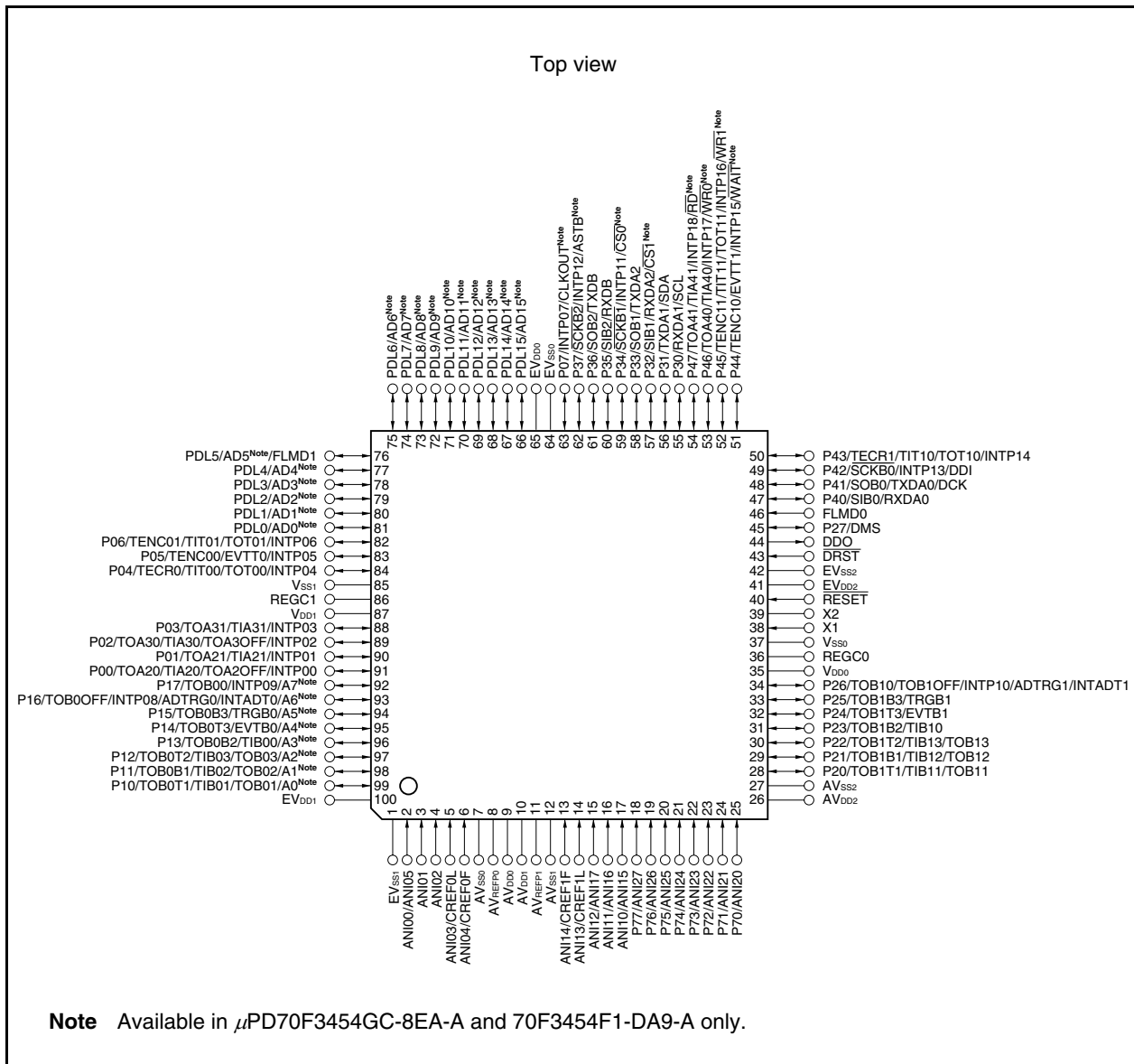
**Remark** The V850E/IG3 microcontrollers are lead-free products.

1.3.4 Pin configuration (V850E/IG3)

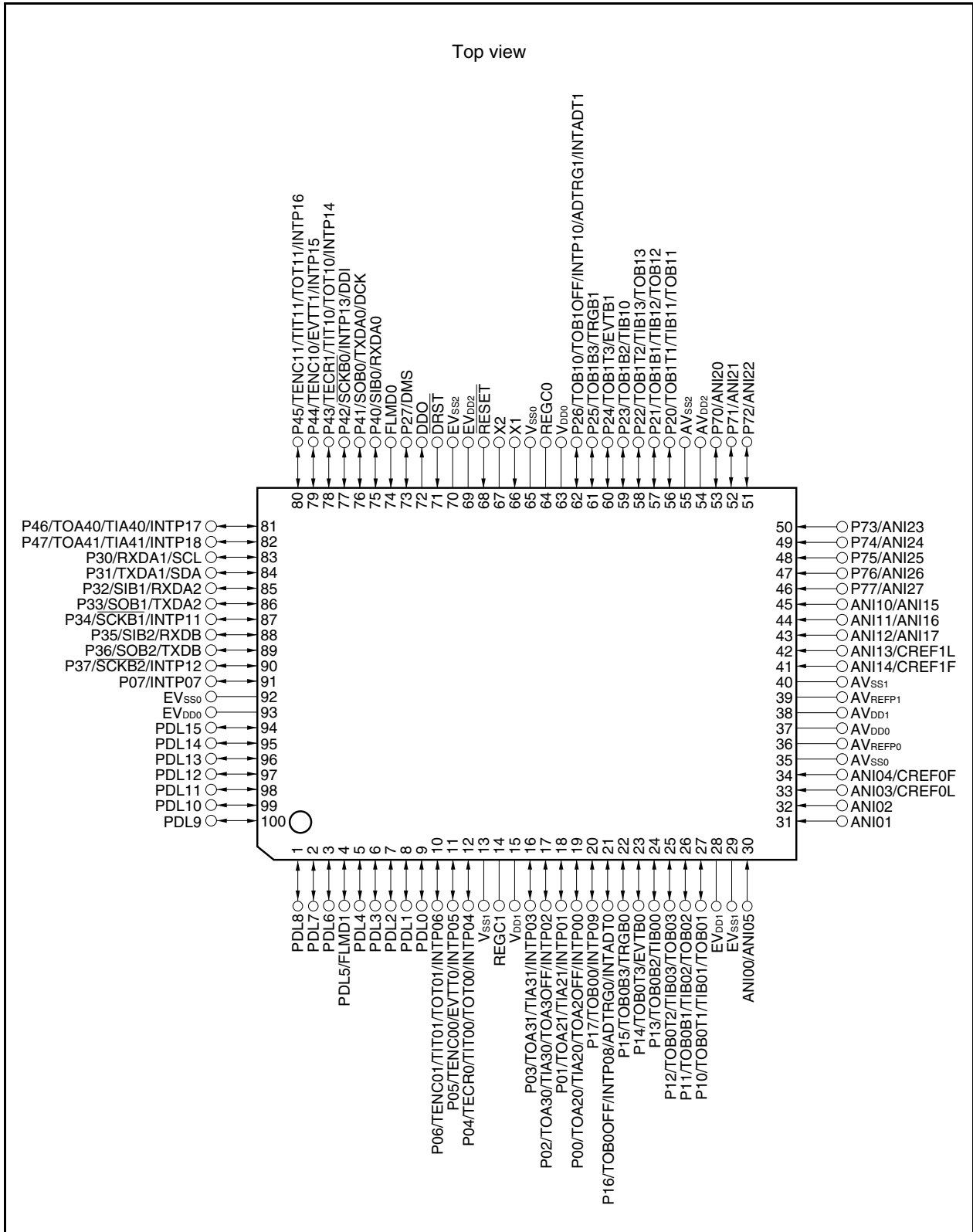
- 100-pin plastic LQFP (fine pitch) (14 × 14)

μPD70F3453GC-8EA-A

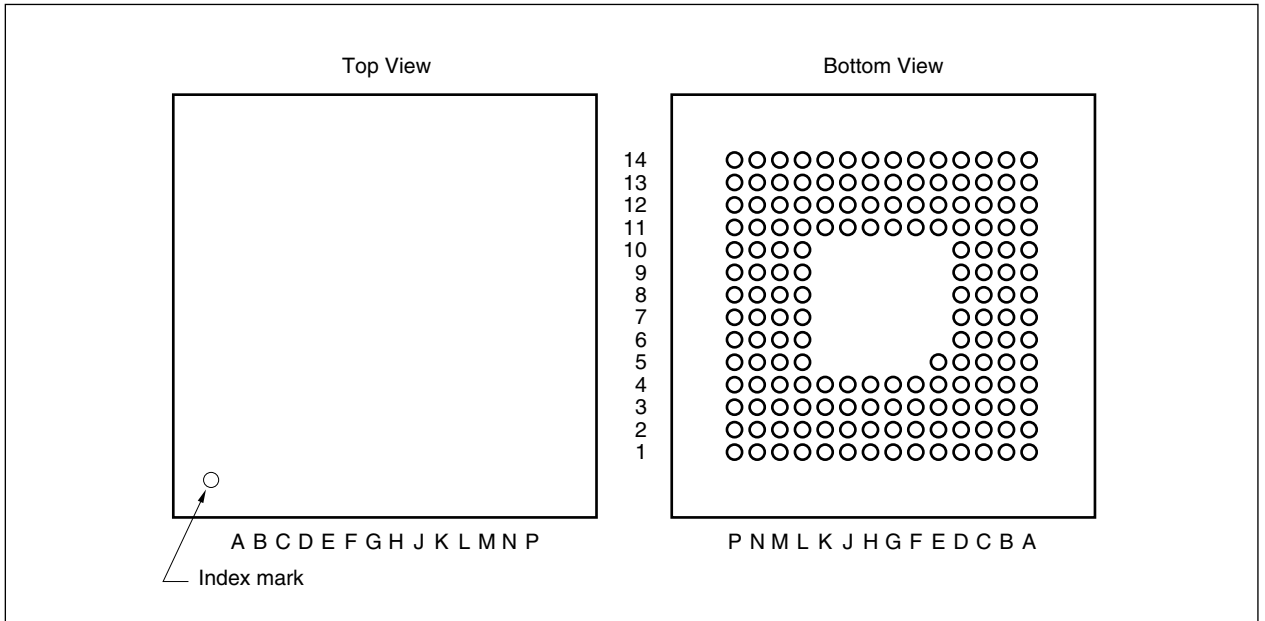
μPD70F3454GC-8EA-A



- 100-pin plastic LQFP (14 × 20)
  - μPD70F3453GF-GAS-AX
  - μPD70F3454GF-GAS-AX



- 161-pin plastic FBGA (10 × 10)  
 $\mu$ PD70F3454F1-DA9-A



(1/2)

| Pin No. | Name                                 | Pin No. | Name                           | Pin No. | Name                |
|---------|--------------------------------------|---------|--------------------------------|---------|---------------------|
| A1      | EV <sub>SS1</sub>                    | B10     | PDL1/AD1                       | D5      | IC1                 |
| A2      | IC1                                  | B11     | PDL3/AD3                       | D6      | IC1                 |
| A3      | EV <sub>DD1</sub>                    | B12     | PDL6/AD6                       | D7      | IC1                 |
| A4      | P12/TOB0T2/TIB03/TOB03/A2            | B13     | IC1                            | D8      | IC1                 |
| A5      | P15/TOB0B3/TRGB0/A5                  | B14     | IC1                            | D9      | IC1                 |
| A6      | P00/TOA20/TIA20/TOA2OFF/INTP00       | C1      | EV <sub>SS1</sub>              | D10     | IC1                 |
| A7      | V <sub>DD1</sub>                     | C2      | EV <sub>SS1</sub>              | D11     | IC1                 |
| A8      | REGC1                                | C3      | P11/TOB0B1/TIB02/TOB02/A1      | D12     | PDL11/AD11          |
| A9      | V <sub>SS1</sub>                     | C4      | P14/TOB0T3/EVTB0/A4            | D13     | PDL10/AD10          |
| A10     | PDL0/AD0                             | C5      | P17/TOB00/INTP09/A7            | D14     | PDL9/AD9            |
| A11     | PDL2/AD2                             | C6      | P02/TOA30/TIA30/TOA3OFF/INTP02 | E1      | AV <sub>SS0</sub>   |
| A12     | PDL5/AD5/FLMD1                       | C7      | P03/TOA31/TIA31/INTP03         | E2      | AV <sub>SS0</sub>   |
| A13     | IC1                                  | C8      | P04/TECR0/TIT00/TOT00/INTP04   | E3      | ANI02               |
| A14     | EV <sub>SS0</sub>                    | C9      | P05/TENC00/EVTT0/INTP05        | E4      | EV <sub>SS1</sub>   |
| B1      | EV <sub>SS1</sub>                    | C10     | P06/TENC01/TIT01/TOT01/INTP06  | E5      | IC0                 |
| B2      | EV <sub>DD1</sub>                    | C11     | PDL4/AD4                       | E11     | IC1                 |
| B3      | P10/TOB0T1/TIB01/TOB01/A0            | C12     | IC1                            | E12     | PDL14/AD14          |
| B4      | P13/TOB0B2/TIB00/A3                  | C13     | PDL8/AD8                       | E13     | PDL13/AD13          |
| B5      | P16/TOB0OFF/INTP08/ADTRG0/INTADT0/A6 | C14     | PDL7/AD7                       | E14     | PDL12/AD12          |
| B6      | P01/TOA21/TIA21/INTP01               | D1      | ANI00/ANI05                    | F1      | AV <sub>DD0</sub>   |
| B7      | V <sub>DD1</sub>                     | D2      | ANI01                          | F2      | AV <sub>REFP0</sub> |
| B8      | REGC1                                | D3      | EV <sub>SS1</sub>              | F3      | ANI03/CREFO1        |
| B9      | V <sub>SS1</sub>                     | D4      | IC1                            | F4      | ANI04/CREFOF        |

- Remarks**
1. Connect the IC0 pin to EV<sub>SS0</sub>, EV<sub>SS1</sub>, or EV<sub>SS2</sub>.
  2. Leave the IC1 pin open.

**CHAPTER 1 INTRODUCTION**

(2/2)

| Pin No. | Name  | Pin No. | Name                                       | Pin No. | Name              |
|---------|---|---------|--|---------|-------------------|
| F11     | IC1   | L6      | IC1  | P5      | V <sub>DD0</sub>  |
| F12     | PDL15/AD15  | L7      | IC1  | P6      | REGC0             |
| F13     | EV <sub>DD0</sub>                                 | L8      | IC1  | P7      | V <sub>SS0</sub>  |
| F14     | EV <sub>DD0</sub>                                 | L9      | IC1  | P8      | X1                |
| G1      | AV <sub>DD1</sub>                                 | L10     | IC1  | P9      | RESET             |
| G2      | AV <sub>REFP1</sub>                               | L11     | IC1  | P10     | EV <sub>DD2</sub> |
| G3      | ANI13/CREF1L                                      | L12     | P46/TOA40/TIA40/INTP17/ $\overline{WR0}$   | P11     | EV <sub>SS2</sub> |
| G4      | ANI14/CREF1F                                      | L13     | P45/TENC11/TIT11/TOT11/INTP16/WR1          | P12     | P40/SIB0/RXDA0    |
| G11     | IC1   | L14     | P44/TENC10/EVTT1/INTP15/ $\overline{WAIT}$ | P13     | IC1               |
| G12     | P07/INTP07/CLKOUT                                 | M1      | AV <sub>DD2</sub>                          | P14     | EV <sub>SS2</sub> |
| G13     | EV <sub>SS0</sub>                                 | M2      | AV <sub>DD2</sub>                          |         |                   |
| G14     | EV <sub>SS0</sub>                                 | M3      | AV <sub>SS2</sub>                          |         |                   |
| H1      | AV <sub>SS1</sub>                                 | M4      | P24/TOB1T3/EVTB1                           |         |                   |
| H2      | AV <sub>SS1</sub>                                 | M5      | P25/TOB1B3/TRGB1                           |         |                   |
| H3      | ANI12/ANI17                                       | M6      | P26/TOB10/TOB1OFF/INTP10/ADTRG1/INTADT1    |         |                   |
| H4      | IC2   | M7      | IC1  |         |                   |
| H11     | IC1   | M8      | IC1  |         |                   |
| H12     | P37/ $\overline{SCKB2}$ /INTP12/ASTB              | M9      | $\overline{DRST}$                          |         |                   |
| H13     | P36/SOB2/TXDB                                     | M10     | DDO  |         |                   |
| H14     | P35/SIB2/RXDB                                     | M11     | P27/DMS                                    |         |                   |
| J1      | ANI10/ANI15                                       | M12     | IC1  |         |                   |
| J2      | ANI11/ANI16                                       | M13     | P43/TECR1/TIT10/TOT10/INTP14               |         |                   |
| J3      | IC2   | M14     | P42/ $\overline{SCKB0}$ /INTP13/DDI        |         |                   |
| J4      | P77/ANI27   | N1      | IC1  |         |                   |
| J11     | IC1   | N2      | AV <sub>SS2</sub>                          |         |                   |
| J12     | P34/ $\overline{SCKB1}$ /INTP11/ $\overline{CS0}$ | N3      | P21/TOB1B1/TIB12/TOB12                     |         |                   |
| J13     | P33/SOB1/TXDA2                                    | N4      | P23/TOB1B2/TIB10                           |         |                   |
| J14     | P32/SIB1/RXDA2/ $\overline{CS1}$                  | N5      | V <sub>DD0</sub>                           |         |                   |
| K1      | P73/ANI23   | N6      | REGC0                                      |         |                   |
| K2      | P74/ANI24   | N7      | V <sub>SS0</sub>                           |         |                   |
| K3      | P75/ANI25   | N8      | X2   |         |                   |
| K4      | P76/ANI26   | N9      | FLMD0                                      |         |                   |
| K11     | IC1   | N10     | EV <sub>DD2</sub>                          |         |                   |
| K12     | P31/TXDA1/SDA                                     | N11     | EV <sub>SS2</sub>                          |         |                   |
| K13     | P30/RXDA1/SCL                                     | N12     | P41/SOB0/TXDA0/DCK                         |         |                   |
| K14     | P47/TOA41/TIA41/INTP18/ $\overline{RD}$           | N13     | IC1  |         |                   |
| L1      | P70/ANI20   | N14     | IC1  |         |                   |
| L2      | P71/ANI21   | P1      | AV <sub>SS2</sub>                          |         |                   |
| L3      | P72/ANI22   | P2      | IC1  |         |                   |
| L4      | AV <sub>SS2</sub>                                 | P3      | P20/TOB1T1/TIB11/TOB11                     |         |                   |
| L5      | IC1   | P4      | P22/TOB1T2/TIB13/TOB13                     |         |                   |

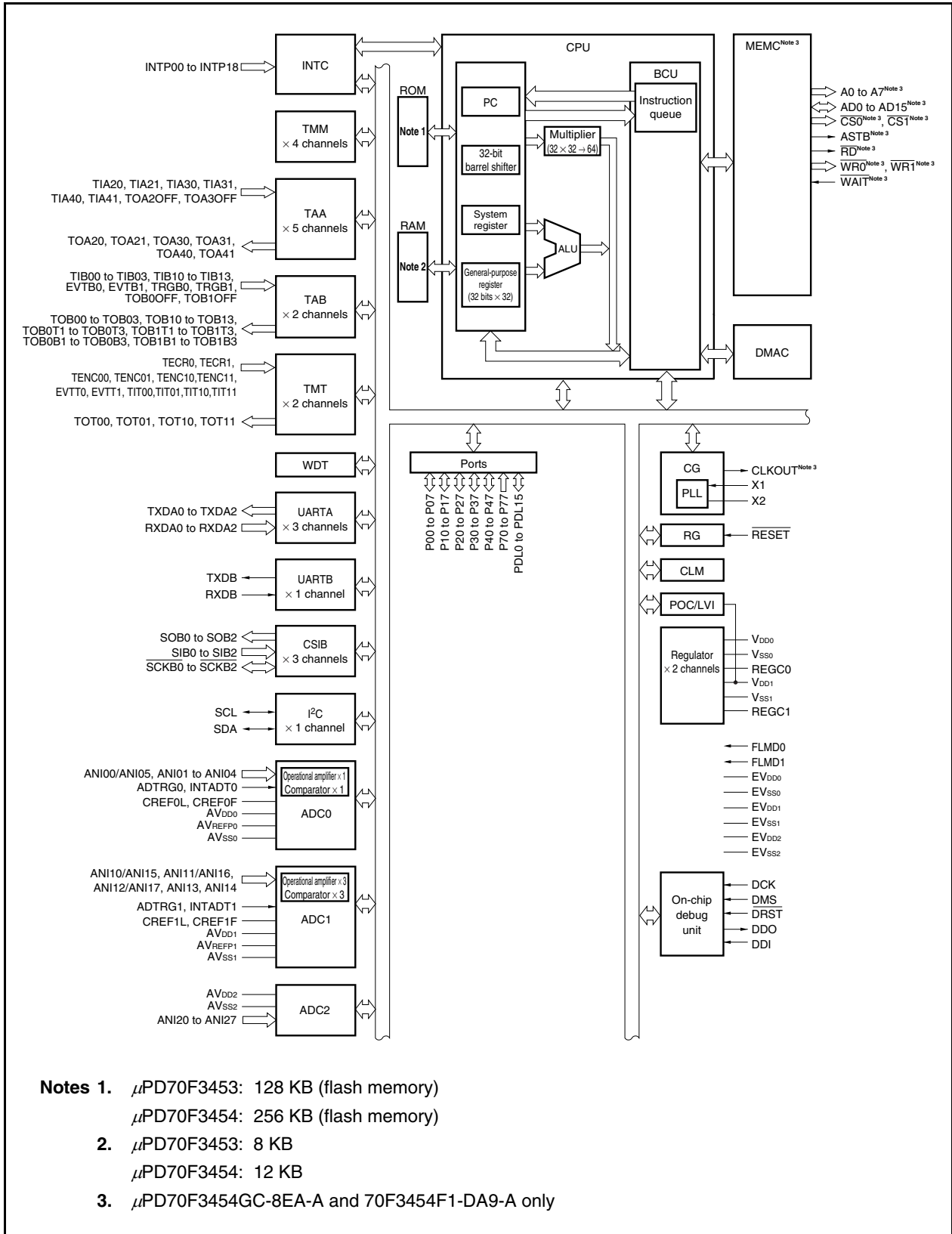
- Remarks**
1. Leave the IC1 pin open.
  2. Independently connect the IC2 pin to AV<sub>SS2</sub> via a resistor.

**Pin Identification (V850E/IG3)**

|   |                              |   |                           |
|---|------------------------------|---|---------------------------|
| A0 to A7:   | Address bus                  | $\overline{\text{SCKB0}}$ to $\overline{\text{SCKB2}}$ :  | Serial clock              |
| AD0 to AD15:  | Address/data bus             | SCL:  | Serial clock              |
| ADTRG0, ADTRG1:   | A/D trigger input            | SDA:  | Serial data               |
| ANI00 to ANI05,<br>ANI10 to ANI17,<br>ANI20 to ANI27:   | Analog input                 | SIB0 to SIB2:   | Serial input              |
| ASTB:   | Address strobe               | SOB0 to SOB2:   | Serial output             |
| AV <sub>DD0</sub> to AV <sub>DD2</sub> :                | Analog power supply          | TECR0, TECR1:   | Timer encoder clear input |
| AV <sub>REFP0</sub> , AV <sub>REFP1</sub> :             | Analog reference voltage     | TENC00, TENC01,<br>TENC10, TENC11:  | Timer encoder input       |
| AV <sub>SS0</sub> to AV <sub>SS2</sub> :                | Analog ground                | TIA20, TIA21,<br>TIA30, TIA31,<br>TIA40, TIA41,<br>TIB00 to TIB03,<br>TIB10 to TIB13,<br>TIT00, TIT01,<br>TIT10, TIT11:   | Timer trigger input       |
| CLKOUT:   | Clock output                 | TOA20, TOA21,<br>TOA30, TOA31,<br>TOA40, TOA41,<br>TOB00 to TOB03,<br>TOB0B1 to TOB0B3,<br>TOB0T1 to TOB0T3,<br>TOB10 to TOB13,<br>TOB1B1 to TOB1B3,<br>TOB1T1 to TOB1T3,<br>TOT00, TOT01,<br>TOT10, TOT11: | Timer output              |
| CREF0F, CREF1F,<br>CREF0L, CREF1L:                      | Comparator reference voltage | TOA2OFF, TOA3OFF,<br>TOB0OFF, TOB1OFF:  | Timer output off          |
| $\overline{\text{CS0}}$ , $\overline{\text{CS1}}$ :     | Chip select                  | TRGB0, TRGB1:   | Timer trigger input       |
| DCK:  | Debug clock                  | TXDA0 to TXDA2,<br>TXDB:  | Transmit data             |
| DDI:  | Debug data input             | V <sub>DD0</sub> , V <sub>DD1</sub> :   | Power supply              |
| DDO:  | Debug data output            | V <sub>SS0</sub> , V <sub>SS1</sub> :   | Ground                    |
| DDM:  | Debug mode select            | $\overline{\text{WAIT}}$ :  | Wait                      |
| DRST:   | Debug reset                  | $\overline{\text{WR0}}$ , $\overline{\text{WR1}}$ :   | Write strobe              |
| EV <sub>DD0</sub> to EV <sub>DD2</sub> :                | Power supply for port        | X1, X2:   | Clock oscillator pin      |
| EV <sub>SS0</sub> to EV <sub>SS2</sub> :                | Ground for port              |   |                           |
| EVTB0, EVTB1,<br>EVTT0, EVTT1:                          | Timer event count input      |   |                           |
| FLMD0, FLMD1:   | Flash programming mode       |   |                           |
| IC0 to IC2:   | Internally connected         |   |                           |
| INTADT0, INTADT1,<br>INTP00 to INTP18:                  | External interrupt input     |   |                           |
| P00 to P07:   | Port 0                       |   |                           |
| P10 to P17:   | Port 1                       |   |                           |
| P20 to P27:   | Port 2                       |   |                           |
| P30 to P37:   | Port 3                       |   |                           |
| P40 to P47:   | Port 4                       |   |                           |
| P70 to P77:   | Port 7                       |   |                           |
| $\overline{\text{PDL0}}$ to $\overline{\text{PDL15}}$ : | Port DL                      |   |                           |
| $\overline{\text{RD}}$ :                                | Read strobe                  |   |                           |
| REGC0, REGC1:   | Regulator control            |   |                           |
| $\overline{\text{RESET}}$ :                             | Reset                        |   |                           |
| RXDA0 to RXDA2,<br>RXDB:                                | Receive data                 |   |                           |

1.3.5 Function blocks (V850E/IG3)

(1) Internal block diagram





**(2) Internal units**

**(a) 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 (32 bits × 32 bits → 64 bits) and a barrel shifter (32 bits), help accelerate complex processing.

**(b) Bus control unit (BCU)**

The BCU starts the required external bus cycles in accordance with the physical address obtained by the CPU. If the CPU does not request the start of a bus cycle when an instruction is fetched from the external memory area ( $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only), the BCU generates a prefetch address and prefetches an instruction code. The prefetched instruction code is loaded to the internal instruction queue.

The BCU controls a memory controller (MEMC) and performs external memory access ( $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only).

**(i) Memory controller (MEMC) ( $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only)**

Controls access to SRAM, external ROM, and external I/O.

**(ii) DMA controller (DMAC)**

This controller controls data transfer between on-chip peripheral I/O and internal RAM or on-chip peripheral I/O and on-chip peripheral I/O instead of the CPU.

The transfer type is two-cycle transfer, and single transfer, single-step transfer, and block transfer are used in transfer mode.

**(c) ROM**

This is flash memory that is mapped from address 00000000H.

During instruction fetch, ROM/flash memory can be accessed from the CPU in 1-clock cycles. The internal ROM capacity and area differ as follows depending on the product.

| Part Number     | Internal ROM Capacity | Internal ROM Area      |
|-----------------|-----------------------|------------------------|
| $\mu$ PD70F3453 | 128 KB (flash memory) | x0000000H to x001FFFFH |
| $\mu$ PD70F3454 | 256 KB (flash memory) | x0000000H to x003FFFFH |

**(d) RAM**

The internal RAM capacity and area differ as follows depending on the product.

During instruction fetch or data access, data can be accessed from the CPU in 1-clock cycles.

| Part Number     | Internal RAM Capacity | Internal RAM Area      |
|-----------------|-----------------------|------------------------|
| $\mu$ PD70F3453 | 8 KB                  | xFFFC000H to xFFFDFFFH |
| $\mu$ PD70F3454 | 12 KB                 | xFFFC000H to xFFFEFFFH |

**(e) Interrupt controller (INTC)**

This controller handles hardware interrupt requests (INTP00 to INTP18, INTADT0, INTADT1) from on-chip peripheral hardware and external hardware. Eight levels of interrupt priorities can be specified for these interrupt requests, and multiple-interrupt servicing control can be performed.

**(f) Clock generator (CG)**

The clock generator includes two basic operation modes: PLL mode (fixed to multiplication by eight) and clock-through mode. It generates four types of clocks ( $f_{xx}$ ,  $f_{xx}/2$ ,  $f_{xx}/4$ ,  $f_{xx}/8$ ), and supplies one of them as the operating clock for the CPU ( $f_{CPU}$ ).

**(g) Timer/counter**

The V850E/IG3 incorporates four 16-bit interval timer M (TMM) channels, five 16-bit timer/event counter AA (TAA) channels, two 16-bit timer/event counter AB (TAB) channels, and two 16-bit timer/event counter T (TMT) channels, and can measure pulse interval widths or frequency, enable an inverter function for motor control, and output a programmable pulse.

**(h) Watchdog timer (WDT)**

A watchdog timer is equipped to detect program loops, system abnormalities, etc.

It generates a non-maskable interrupt request signal (INTWDT) or internal reset signal (WDTRES) after an overflow occurs.

**(i) Serial interface**

The V850E/IG3 incorporates eight serial interface channels: for three asynchronous serial interface A (UARTA) channels, one asynchronous serial interface B (UARTB) channel, three clocked serial interface B (CSIB) channels, and one I<sup>2</sup>C bus interface (I<sup>2</sup>C) channel. Of these, UART0 and CSIB0, UARTA1 and I<sup>2</sup>C, UART2 and CSIB1, and UARTB and CSIB2 share a pin.

For UARTA, data is transferred via the TXDAn and RXDAn pins ( $n = 0$  to 2).

For UARTB, data is transferred via the TXDB and RXDB pins.

For CSIB, data is transferred via the SOBn, SIBn, and  $\overline{SCKBn}$  pins ( $n = 0$  to 2).

For I<sup>2</sup>C, data is transferred via the SCL and SDA pins.

**(j) A/D converter (ADC)**

One channel is provided for each of the high-speed, high-resolution 12-bit A/D converters (ADC0, ADC1) (total of two channels), which have five analog input pins respectively, and one channel is provided for the 10-bit A/D converter (ADC2), which has eight analog input pins.

Both one of the ADC0 channels and three of the ADC1 channels include an operational amplifier and a comparator so that these A/D converters can amplify an analog input voltage and detect overvoltage input.

**(k) On-chip debug function**

An on-chip debug function supporting MINICUBE and MINICUBE2 can be used, so that a simple, inexpensive debug environment can be organized.

**(I) Ports**

As shown below, the following ports have general-purpose port functions and control pin functions.

| Port    | I/O         | Alternate Function   |
|---------|-------------|--|
| Port 0  | 8-bit I/O   | Timer/counter I/O, external interrupt input, external bus interface control signal output  |
| Port 1  | 8-bit I/O   | Timer/counter I/O, external bus interface control signal output, external trigger input of A/D converter 0, external interrupt input |
| Port 2  | 8-bit I/O   | Timer/counter I/O, external trigger input of A/D converter 1, external interrupt input, debug input                                  |
| Port 3  | 8-bit I/O   | Serial interface I/O, external bus interface control signal output, external interrupt input   |
| Port 4  | 8-bit I/O   | Serial interface I/O, timer/counter I/O, debug input, external interrupt input, external bus interface control signal I/O            |
| Port 7  | 8-bit input | A/D converter 2 input  |
| Port DL | 16-bit I/O  | External bus interface control signal I/O  |

## CHAPTER 2 PIN FUNCTIONS

The names and functions of the pins in the V850E/IF3 and V850E/IG3 are listed below. These pins can be divided into port pins and non-port pins according to their function.

### 2.1 List of Pin Functions

There are two power supplies for the I/O buffer of a pin:  $AV_{DD2}$  and  $EV_{DD0}$ ,  $EV_{DD1}$ ,  $EV_{DD2}$  (V850E/IG3 only). The relationship between each power supply and the pins is shown below.

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

**(a) V850E/IF3**

| Power Supply            | Corresponding Pins   |
|-------------------------|--|
| $AV_{DD2}$              | P70 to P73   |
| $EV_{DD0}$ , $EV_{DD1}$ | P00, P01, P10 to P17, P20 to P27, P30 to P37, P40 to P47, PDL0 to PDL9, $\overline{RESET}$ |

**(b) V850E/IG3**

| Power Supply                         | Corresponding Pins  |
|--------------------------------------|---|
| $AV_{DD2}$                           | P70 to P77  |
| $EV_{DD0}$ , $EV_{DD1}$ , $EV_{DD2}$ | P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, PDL0 to PDL15, $\overline{RESET}$ , DCK, DDI, DDO, DMS, $\overline{DRST}$ |

(1) Port pins

(1/3)

| Pin Name              | Pin No. |    |     |     | I/O | Function  | Alternate-Function Pin  |
|-----------------------|---------|----|-----|-----|-----|---|---|
|                       | IF3     |    | IG3 |     |     |   |   |
|                       | GC      | GC | GF  | F1  |     |   |   |
| P00                   | 70      | 91 | 19  | A6  | I/O | Port 0<br>V850E/IF3: 2-bit I/O port<br>V850E/IG3: 8-bit I/O port<br>Input data read/output data write is enabled in 1-bit units.<br><br>An on-chip pull-up resistor can be specified in 1-bit units (the on-chip pull-up resistor can be connected when the pins are in the port mode and input mode, and when the pins function as input pins of the alternate function, and when TOA21 and TOA31 (V850E/IG3 only) pins (output pins of the alternate function) go into a high-impedance state). | TOA20/TIA20/TOA2OFF/INTP00  |
| P01                   | 69      | 90 | 18  | B6  |     |   | TOA21/TIA21/INTP01  |
| P02 <sup>Note 1</sup> | –       | 89 | 17  | C6  |     |   | TOA30 <sup>Note 1</sup> /TIA30 <sup>Note 1</sup> /TOA3OFF <sup>Note 1</sup> /INTP02 <sup>Note 1</sup> |
| P03 <sup>Note 1</sup> | –       | 88 | 16  | C7  |     |   | TOA31 <sup>Note 1</sup> /TIA31 <sup>Note 1</sup> /INTP03 <sup>Note 1</sup>                            |
| P04 <sup>Note 1</sup> | –       | 84 | 12  | C8  |     |   | TECR0 <sup>Note 1</sup> /TIT00 <sup>Note 1</sup> /TOT00 <sup>Note 1</sup> /INTP04 <sup>Note 1</sup>   |
| P05 <sup>Note 1</sup> | –       | 83 | 11  | C9  |     |   | TENC00 <sup>Note 1</sup> /EVTT0 <sup>Note 1</sup> /INTP05 <sup>Note 1</sup>                           |
| P06 <sup>Note 1</sup> | –       | 82 | 10  | C10 |     |   | TENC01 <sup>Note 1</sup> /TIT01 <sup>Note 1</sup> /TOT01 <sup>Note 1</sup> /INTP06 <sup>Note 1</sup>  |
| P07 <sup>Note 1</sup> | –       | 63 | 91  | G12 |     |   | INTP07 <sup>Note 1</sup> /CLKOUT <sup>Note 2</sup>  |
| P10                   | 78      | 99 | 27  | B3  | I/O | Port 1<br>8-bit I/O port<br>Input data read/output data write is enabled in 1-bit units.<br><br>An on-chip pull-up resistor can be specified in 1-bit units (the on-chip pull-up resistor can be connected when the pins are in the port mode and input mode, and when the pins function as input pins of the alternate function, and when TOB0B1 to TOB0B3 and TOB0T1 to TOB0T3 pins (output pins of the alternate function) go into a high-impedance state).                                    | TOB0T1/TIB01/TOB01/A0 <sup>Note 2</sup>   |
| P11                   | 77      | 98 | 26  | C3  |     |   | TOB0B1/TIB02/TOB02/A1 <sup>Note 2</sup>   |
| P12                   | 76      | 97 | 25  | A4  |     |   | TOB0T2/TIB03/TOB03/A2 <sup>Note 2</sup>   |
| P13                   | 75      | 96 | 24  | B4  |     |   | TOB0B2/TIB00/A3 <sup>Note 2</sup>   |
| P14                   | 74      | 95 | 23  | C4  |     |   | TOB0T3/EVTB0/A4 <sup>Note 2</sup>   |
| P15                   | 73      | 94 | 22  | A5  |     |   | TOB0B3/TRGB0/A5 <sup>Note 2</sup>   |
| P16                   | 72      | 93 | 21  | B5  |     |   | TOB0OFF/INTP08/ADTRG0/INTADT0/A6 <sup>Note 2</sup>  |
| P17                   | 71      | 92 | 20  | C5  |     |   | TOB00/INTP09/A7 <sup>Note 2</sup>   |

Notes 1. V850E/IG3 only

2.  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

Remark IF3: V850E/IF3

IG3: V850E/IG3

GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)

GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)

GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

CHAPTER 2 PIN FUNCTIONS

(2/3)

| Pin Name | Pin No. |     |    |     | I/O | Function   | Alternate-Function Pin                          |
|----------|---------|-----|----|-----|-----|--|---|
|          | IF3     | IG3 |    |     |     |  |   |
|          | GC      | GC  | GF | F1  |     |  |   |
| P20      | 23      | 28  | 56 | P3  | I/O | Port 2<br>8-bit I/O port<br>Input data read/output data write is enabled in 1-bit units.<br>An on-chip pull-up resistor can be specified in 1-bit units (the on-chip pull-up resistor can be connected when the pins are in the port mode and input mode, and when the pins function as input pins of the alternate function, and when TOB1B1 to TOB1B3 and TOB1T1 to TOB1T3 pins (output pins of the alternate function) go into a high-impedance state).   | TOB1T1/TIB11/TOB11                              |
| P21      | 24      | 29  | 57 | N3  |     |  | TOB1B1/TIB12/TOB12                              |
| P22      | 25      | 30  | 58 | P4  |     |  | TOB1T2/TIB13/TOB13                              |
| P23      | 26      | 31  | 59 | N4  |     |  | TOB1B2/TIB10                                    |
| P24      | 27      | 32  | 60 | M4  |     |  | TOB1T3/EVTB1                                    |
| P25      | 28      | 33  | 61 | M5  |     |  | TOB1B3/TRGB1                                    |
| P26      | 29      | 34  | 62 | M6  |     |  | TOB10/TOB1OFF/INTP10/ADTRG1/INTADT1             |
| P27      | 36      | 45  | 73 | M11 |     |  | DMS <sup>Note 1</sup>                           |
| P30      | 46      | 55  | 83 | K13 | I/O | Port 3<br>8-bit I/O port<br>Input data read/output data write is enabled in 1-bit units.<br>An on-chip pull-up resistor can be specified in 1-bit units (the on-chip pull-up resistor can be connected when the pins are in the port mode and input mode, and when the pins function as input pins of the alternate function (including the SCKB1 and SCKB2 pins in the slave mode)).<br>If the SCL or SDA pin is selected when the alternate function is to be used, N-ch open-drain output can be specified. | RXDA1/SCL                                       |
| P31      | 47      | 56  | 84 | K12 |     |  | TXDA1/SDA                                       |
| P32      | 48      | 57  | 85 | J14 |     |  | SIB1/RXDA2/CS1 <sup>Note 2</sup>                |
| P33      | 49      | 58  | 86 | J13 |     |  | SOB1/TXDA2                                      |
| P34      | 50      | 59  | 87 | J12 |     |  | SCKB1/INTP11/CS0 <sup>Note 2</sup>              |
| P35      | 51      | 60  | 88 | H14 |     |  | SIB2/RXDB                                       |
| P36      | 52      | 61  | 89 | H13 |     |  | SOB2/TXDB                                       |
| P37      | 53      | 62  | 90 | H12 |     |  | SCKB2/INTP12/ASTB <sup>Note 2</sup>             |
| P40      | 38      | 47  | 75 | P12 | I/O | Port 4<br>8-bit I/O port<br>Input data read/output data write is enabled in 1-bit units.<br>An on-chip pull-up resistor can be specified in 1-bit units (the on-chip pull-up resistor can be connected when the pins are in the port mode and input mode, and when the pins function as input pins of the alternate function (including the SCKB0 pin in the slave mode)).   | SIB0/RXDA0                                      |
| P41      | 39      | 48  | 76 | N12 |     |  | SOB0/TXDA0/DCK <sup>Note 1</sup>                |
| P42      | 40      | 49  | 77 | M14 |     |  | SCKB0/INTP13/DD1 <sup>Note 1</sup>              |
| P43      | 41      | 50  | 78 | M13 |     |  | TECR1/TIT10/TOT10/INTP14                        |
| P44      | 42      | 51  | 79 | L14 |     |  | TENC10/EVTT1/INTP15/WAIT <sup>Note 2</sup>      |
| P45      | 43      | 52  | 80 | L13 |     |  | TENC11/TIT11/TOT11/INTP16/WR1 <sup>Note 2</sup> |
| P46      | 44      | 53  | 81 | L12 |     |  | TOA40/TIA40/INTP17/WR0 <sup>Note 2</sup>        |
| P47      | 45      | 54  | 82 | K14 |     |  | TOA41/TIA41/INTP18/RD <sup>Note 2</sup>         |

- Notes** 1. V850E/IG3 only  
 2.  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**Remark** IF3: V850E/IF3  
 IG3: V850E/IG3  
 GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)  
 GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)  
 GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)  
 F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

| Pin Name                | Pin No. |    |     |     | I/O   | Function   | Alternate-Function Pin       |
|-------------------------|---------|----|-----|-----|-------|--|------------------------------|
|                         | IF3     |    | IG3 |     |       |  |                              |
|                         | GC      | GC | GF  | F1  |       |  |                              |
| P70                     | 20      | 25 | 53  | L1  | Input | Port 7<br>V850E/IF3: 4-bit input port<br>V850E/IG3: 8-bit input port   | ANI20                        |
| P71                     | 19      | 24 | 52  | L2  |       |  | ANI21                        |
| P72                     | 18      | 23 | 51  | L3  |       |  | ANI22                        |
| P73                     | 17      | 22 | 50  | K1  |       |  | ANI23                        |
| P74 <sup>Note 1</sup>   | –       | 21 | 49  | K2  |       |  | ANI24 <sup>Note 1</sup>      |
| P75 <sup>Note 1</sup>   | –       | 20 | 48  | K3  |       |  | ANI25 <sup>Note 1</sup>      |
| P76 <sup>Note 1</sup>   | –       | 19 | 47  | K4  |       |  | ANI26 <sup>Note 1</sup>      |
| P77 <sup>Note 1</sup>   | –       | 18 | 46  | J4  |       |  | ANI27 <sup>Note 1</sup>      |
| PDL0                    | 65      | 81 | 9   | A10 | I/O   | Port DL<br>V850E/IF3: 10-bit I/O port<br>V850E/IG3: 16-bit I/O port<br>Input data read/output data write is enabled in 1-bit units.<br><br>An on-chip pull-up resistor can be specified in 1-bit units (the on-chip pull-up resistor can be connected only when the pins are in the port mode and input mode). | AD0 <sup>Note 2</sup>        |
| PDL1                    | 64      | 80 | 8   | B10 |       |  | AD1 <sup>Note 2</sup>        |
| PDL2                    | 63      | 79 | 7   | A11 |       |  | AD2 <sup>Note 2</sup>        |
| PDL3                    | 62      | 78 | 6   | B11 |       |  | AD3 <sup>Note 2</sup>        |
| PDL4                    | 61      | 77 | 5   | C11 |       |  | AD4 <sup>Note 2</sup>        |
| PDL5                    | 60      | 76 | 4   | A12 |       |  | AD5 <sup>Note 2</sup> /FLMD1 |
| PDL6                    | 59      | 75 | 3   | B12 |       |  | AD6 <sup>Note 2</sup>        |
| PDL7                    | 58      | 74 | 2   | C14 |       |  | AD7 <sup>Note 2</sup>        |
| PDL8                    | 57      | 73 | 1   | C13 |       |  | AD8 <sup>Note 2</sup>        |
| PDL9                    | 56      | 72 | 100 | D14 |       |  | AD9 <sup>Note 2</sup>        |
| PDL10 <sup>Note 1</sup> | –       | 71 | 99  | D13 |       |  | AD10 <sup>Note 2</sup>       |
| PDL11 <sup>Note 1</sup> | –       | 70 | 98  | D12 |       |  | AD11 <sup>Note 2</sup>       |
| PDL12 <sup>Note 1</sup> | –       | 69 | 97  | E14 |       |  | AD12 <sup>Note 2</sup>       |
| PDL13 <sup>Note 1</sup> | –       | 68 | 96  | E13 |       |  | AD13 <sup>Note 2</sup>       |
| PDL14 <sup>Note 1</sup> | –       | 67 | 95  | E12 |       |  | AD14 <sup>Note 2</sup>       |
| PDL15 <sup>Note 1</sup> | –       | 66 | 94  | F12 |       |  | AD15 <sup>Note 2</sup>       |

- Notes 1.** V850E/IG3 only  
**2.**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**Remark** IF3: V850E/IF3  
 IG3: V850E/IG3  
 GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)  
 GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)  
 GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)  
 F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

(2) Non-port pins

(1/8)

| Pin Name               | Pin No. |    |     |     | I/O                     | Function                                    | Alternate-Function Pin                          |
|------------------------|---------|----|-----|-----|-------------------------|---|---|
|                        | IF3     |    | IG3 |     |                         |   |   |
|                        | GC      | GC | GF  | F1  |                         |   |   |
| A0 <sup>Note 1</sup>   | -       | 99 | -   | B3  | Output                  | 8-bit address bus for external memory       | P10/TOB0T1/TIB01/TOB01                          |
| A1 <sup>Note 1</sup>   | -       | 98 | -   | C3  |                         |   | P11/TOB0B1/TIB02/TOB02                          |
| A2 <sup>Note 1</sup>   | -       | 97 | -   | A4  |                         |   | P12/TOB0T2/TIB03/TOB03                          |
| A3 <sup>Note 1</sup>   | -       | 96 | -   | B4  |                         |   | P13/TOB0B2/TIB00                                |
| A4 <sup>Note 1</sup>   | -       | 95 | -   | C4  |                         |   | P14/TOB0T3/EVTB0                                |
| A5 <sup>Note 1</sup>   | -       | 94 | -   | A5  |                         |   | P15/TOB0B3/TRGB0                                |
| A6 <sup>Note 1</sup>   | -       | 93 | -   | B5  |                         |   | P16/TOB0OFF/INTP08/ADTRG0/INTADT0               |
| A7 <sup>Note 1</sup>   | -       | 92 | -   | C5  |                         |   | P17/TOB00/INTP09                                |
| AD0 <sup>Note 1</sup>  | -       | 81 | -   | A10 | I/O                     | 16-bit address/data bus for external memory | PDL0  |
| AD1 <sup>Note 1</sup>  | -       | 80 | -   | B10 |                         |   | PDL1  |
| AD2 <sup>Note 1</sup>  | -       | 79 | -   | A11 |                         |   | PDL2  |
| AD3 <sup>Note 1</sup>  | -       | 78 | -   | B11 |                         |   | PDL3  |
| AD4 <sup>Note 1</sup>  | -       | 77 | -   | C11 |                         |   | PDL4  |
| AD5 <sup>Note 1</sup>  | -       | 76 | -   | A12 |                         |   | FLMD1/PDL5                                      |
| AD6 <sup>Note 1</sup>  | -       | 75 | -   | B12 |                         |   | PDL6  |
| AD7 <sup>Note 1</sup>  | -       | 74 | -   | C14 |                         |   | PDL7  |
| AD8 <sup>Note 1</sup>  | -       | 73 | -   | C13 |                         |   | PDL8  |
| AD9 <sup>Note 1</sup>  | -       | 72 | -   | D14 |                         |   | PDL9  |
| AD10 <sup>Note 1</sup> | -       | 71 | -   | D13 |                         |   | PDL10 <sup>Note 2</sup>                         |
| AD11 <sup>Note 1</sup> | -       | 70 | -   | D12 |                         |   | PDL11 <sup>Note 2</sup>                         |
| AD12 <sup>Note 1</sup> | -       | 69 | -   | E14 |                         |   | PDL12 <sup>Note 2</sup>                         |
| AD13 <sup>Note 1</sup> | -       | 68 | -   | E13 |                         |   | PDL13 <sup>Note 2</sup>                         |
| AD14 <sup>Note 1</sup> | -       | 67 | -   | E12 |                         |   | PDL14 <sup>Note 2</sup>                         |
| AD15 <sup>Note 1</sup> | -       | 66 | -   | F12 | PDL15 <sup>Note 2</sup> |   |   |
| ADTRG0                 | 72      | 93 | 21  | B5  | Input                   | External trigger input for A/D converter 0  | P16/TOB0OFF/INTP08/INTADT0/A6 <sup>Note 1</sup> |
| ADTRG1                 | 29      | 34 | 62  | M6  | Input                   | External trigger input for A/D converter 1  | P26/TOB10/TOB1OFF/INTP10/INTADT1                |

Notes 1.  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

2. V850E/IG3 only

Remark IF3: V850E/IF3

IG3: V850E/IG3

GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)

GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)

GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)



| Pin Name                | Pin No. |     |    |        | I/O    | Function                                    | Alternate-Function Pin |
|-------------------------|---------|-----|----|--------|--------|---|------------------------|
|                         | IF3     | IG3 |    |        |        |   |                        |
|                         | GC      | GC  | GF | F1     |        |   |                        |
| ANI00                   | 1       | 2   | 30 | D1     | Input  | Analog input for A/D converter 0            | ANI05                  |
| ANI01                   | 2       | 3   | 31 | D2     |        |   | -                      |
| ANI02                   | 3       | 4   | 32 | E3     |        |   | -                      |
| ANI03                   | 4       | 5   | 33 | F3     |        |   | CREF0L                 |
| ANI04                   | 5       | 6   | 34 | F4     |        |   | CREF0F                 |
| ANI05                   | 1       | 2   | 30 | D1     |        |   | ANI00                  |
| ANI10                   | 16      | 17  | 45 | J1     | Input  | Analog input for A/D converter 1            | ANI15                  |
| ANI11                   | 15      | 16  | 44 | J2     |        |   | ANI16                  |
| ANI12                   | 14      | 15  | 43 | H3     |        |   | ANI17                  |
| ANI13                   | 13      | 14  | 42 | G3     |        |   | CREF1L                 |
| ANI14                   | 12      | 13  | 41 | G4     |        |   | CREF1F                 |
| ANI15                   | 16      | 17  | 45 | J1     |        |   | ANI10                  |
| ANI16                   | 15      | 16  | 44 | J2     |        |   | ANI11                  |
| ANI17                   | 14      | 15  | 43 | H3     |        |   | ANI12                  |
| ANI20                   | 20      | 25  | 53 | L1     | Input  | Analog input for A/D converter 2            | P70                    |
| ANI21                   | 19      | 24  | 52 | L2     |        |   | P71                    |
| ANI22                   | 18      | 23  | 51 | L3     |        |   | P72                    |
| ANI23                   | 17      | 22  | 50 | K1     |        |   | P73                    |
| ANI24 <sup>Note 1</sup> | -       | 21  | 49 | K2     |        |   | P74 <sup>Note 1</sup>  |
| ANI25 <sup>Note 1</sup> | -       | 20  | 48 | K3     |        |   | P75 <sup>Note 1</sup>  |
| ANI26 <sup>Note 1</sup> | -       | 19  | 47 | K4     |        |   | P76 <sup>Note 1</sup>  |
| ANI27 <sup>Note 1</sup> | -       | 18  | 46 | J4     |        |   | P77 <sup>Note 1</sup>  |
| ASTB <sup>Note 2</sup>  | -       | 62  | -  | H12    | Output | Address strobe output of external data bus  | P37/SCKB2/INTP12       |
| AV <sub>DD0</sub>       | 8       | 9   | 37 | F1     | -      | Positive power supply for A/D converter 0   | -                      |
| AV <sub>DD1</sub>       | 9       | 10  | 38 | G1     | -      | Positive power supply for A/D converter 1   | -                      |
| AV <sub>DD2</sub>       | 21      | 26  | 54 | M1, M2 | -      | Positive power supply for A/D converter 2   | -                      |
| AV <sub>REFP0</sub>     | 7       | 8   | 36 | F2     | -      | Reference voltage input for A/D converter 0 | -                      |
| AV <sub>REFP1</sub>     | 10      | 11  | 39 | G2     | -      | Reference voltage input for A/D converter 1 | -                      |

- Notes 1.** V850E/IG3 only  
**2.**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**Remark** IF3: V850E/IF3  
 IG3: V850E/IG3  
 GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)  
 GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)  
 GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)  
 F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

| Pin Name                                  | Pin No. |     |    |               | I/O    | Function   | Alternate-Function Pin                          |
|---|---------|-----|----|---------------|--------|--|---|
|   | IF3     | IG3 |    |               |        |  |   |
|   | GC      | GC  | GF | F1            |        |  |   |
| AV <sub>SS0</sub>                         | 6       | 7   | 35 | E1, E2        | –      | Ground potential for A/D converter 0                       | –   |
| AV <sub>SS1</sub>                         | 11      | 12  | 40 | H1, H2        | –      | Ground potential for A/D converter 1                       | –   |
| AV <sub>SS2</sub>                         | 22      | 27  | 55 | <b>Note 1</b> | –      | Ground potential for A/D converter 2                       | –   |
| CLKOUT <sup>Note 2</sup>                  | –       | 63  | –  | G12           | Output | External bus clock output                                  | P07 <sup>Note 3</sup> /INTP07 <sup>Note 3</sup> |
| CREF0L                                    | 4       | 5   | 33 | F3            | –      | Low range comparator reference voltage of A/D converter 0  | ANI03   |
| CREF1L                                    | 13      | 14  | 42 | G3            | –      | Low range comparator reference voltage of A/D converter 1  | ANI13   |
| CREF0F                                    | 5       | 6   | 34 | F4            | –      | Full range comparator reference voltage of A/D converter 0 | ANI04   |
| CREF1F                                    | 12      | 13  | 41 | G4            | –      | Full range comparator reference voltage of A/D converter 1 | ANI14   |
| $\overline{\text{CS0}}$ <sup>Note 2</sup> | –       | 59  | –  | J12           | Output | Chip select output   | P34/ $\overline{\text{SCKB1}}$ /INTP11          |
| $\overline{\text{CS1}}$ <sup>Note 2</sup> | –       | 57  | –  | J14           |        |  | P32/SIB1/RXDA2                                  |
| DCK <sup>Note 3</sup>                     | –       | 48  | 76 | N12           | Input  | Debug clock input for on-chip debug emulator               | P41/SOB0/TXDA0                                  |
| DDI <sup>Note 3</sup>                     | –       | 49  | 77 | M14           | Input  | Debug data input for on-chip debug emulator                | P42/ $\overline{\text{SCKB0}}$ /INTP13          |
| DDO <sup>Note 3</sup>                     | –       | 44  | 72 | M10           | Output | Debug data output for on-chip debug emulator               | –   |
| DMS <sup>Note 3</sup>                     | –       | 45  | 73 | M11           | Input  | Debug mode select for on-chip debug emulator               | P27   |
| DRST <sup>Note 3</sup>                    | –       | 43  | 71 | M9            | Input  | Debug reset input for on-chip debug emulator               | –   |
| EV <sub>DD0</sub>                         | 55      | 65  | 93 | F13, F14      | –      | Positive power supply for external pin                     | –   |
| EV <sub>DD1</sub>                         | 80      | 100 | 28 | A3, B2        |        |  | –   |
| EV <sub>DD2</sub> <sup>Note 3</sup>       | –       | 41  | 69 | N10, P10      |        |  | –   |
| EV <sub>SS0</sub>                         | 54      | 64  | 92 | Note 4        | –      | Ground potential for external pin                          | –   |
| EV <sub>SS1</sub>                         | 79      | 1   | 29 | Note 5        |        |  | –   |
| EV <sub>SS2</sub> <sup>Note 3</sup>       | –       | 42  | 70 | Note 6        |        |  | –   |
| EVTB0                                     | 74      | 95  | 23 | C4            | Input  | External event count input of TAB0, TAB1                   | A4 <sup>Note 1</sup> /P14/TOB0T3                |
| EVTB1                                     | 27      | 32  | 60 | M4            |        |  | P24/TOB1T3                                      |

- Notes**
- L4, M3, N2, P1
  - $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only
  - V850E/IG3 only
  - A14, G13, G14
  - A1, B1, C1, C2, D3, E4
  - N11, P11, P14

**Remark**

IF3: V850E/IF3  
 IG3: V850E/IG3  
 GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)  
 GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)  
 GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)  
 F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

| Pin Name                 | Pin No. |     |    |               | I/O   | Function  | Alternate-Function Pin   |
|--------------------------|---------|-----|----|---------------|---|---|--|
|                          | IF3     | IG3 |    |               |   |   |  |
|                          | GC      | GC  | GF | F1            |   |   |  |
| EVTT0 <sup>Note 1</sup>  | –       | 83  | 11 | C9            | Input   | External event count input of TMT0, TMT1/external trigger input | P05 <sup>Note 1</sup> /TENC00 <sup>Note 1</sup> /INTP05 <sup>Note 1</sup>                          |
| EVTT1                    | 42      | 51  | 79 | L14           |   |   | P44/TENC10/INTP15/ $\overline{\text{WAIT}}$ <sup>Note 2</sup>                                      |
| FLMD0                    | 37      | 46  | 74 | N9            | Input   | Pin for setting flash memory programming mode                   | –  |
| FLMD1                    | 60      | 76  | 4  | A12           |   |   | PDL5/AD5 <sup>Note 2</sup>   |
| IC0 <sup>Note 3</sup>    | –       | –   | –  | E5            | –   | Internally connected pins                                       | –  |
| IC1 <sup>Note 3</sup>    | –       | –   | –  | <b>Note 4</b> | –   |   | –  |
| IC2 <sup>Note 3</sup>    | –       | –   | –  | <b>Note 5</b> | –   |   | –  |
| INTADT0                  | 72      | 93  | 21 | B5            | Input   | External maskable interrupt request input                       | P16/TOB0OFF/INTP08/ADTRG0/A6 <sup>Note 2</sup>   |
| INTADT1                  | 29      | 34  | 62 | M6            |   |   | P26/TOB10/TOB1OFF/INTP10/ADTRG1  |
| INTP00                   | 70      | 91  | 19 | A6            |   |   | P00/TOA20/TIA20/TOA2OFF  |
| INTP01                   | 69      | 90  | 18 | B6            |   |   | P01/TOA21/TIA21  |
| INTP02 <sup>Note 1</sup> | –       | 89  | 17 | C6            |   |   | P02 <sup>Note 1</sup> /TOA30 <sup>Note 1</sup> /TIA30 <sup>Note 1</sup> /TOA3OFF <sup>Note 1</sup> |
| INTP03 <sup>Note 1</sup> | –       | 88  | 16 | C7            |   |   | P03 <sup>Note 1</sup> /TOA31 <sup>Note 1</sup> /TIA31 <sup>Note 1</sup>                            |
| INTP04 <sup>Note 1</sup> | –       | 84  | 12 | C8            |   |   | P04 <sup>Note 1</sup> /TECR0 <sup>Note 1</sup> /TIT00 <sup>Note 1</sup> /TOT00 <sup>Note 1</sup>   |
| INTP05 <sup>Note 1</sup> | –       | 83  | 11 | C9            |   |   | P05 <sup>Note 1</sup> /TENC00 <sup>Note 1</sup> /EVTT0 <sup>Note 1</sup>                           |
| INTP06 <sup>Note 1</sup> | –       | 82  | 10 | C10           |   |   | P06 <sup>Note 1</sup> /TENC01 <sup>Note 1</sup> /TIT01 <sup>Note 1</sup> /TOT01 <sup>Note 1</sup>  |
| INTP07 <sup>Note 1</sup> | –       | 63  | 91 | G12           |   |   | P07 <sup>Note 1</sup> /CLKOUT <sup>Note 2</sup>  |
| INTP08                   | 72      | 93  | 21 | B5            |   |   | P16/TOB0OFF/ADTRG0/INTADT0/A6 <sup>Note 2</sup>  |
| INTP09                   | 71      | 92  | 20 | C5            |   |   | P17/TOB00/A7 <sup>Note 2</sup>   |
| INTP10                   | 29      | 34  | 62 | M6            |   |   | P26/TOB10/TOB1OFF/ADTRG1/INTADT1   |
| INTP11                   | 50      | 59  | 87 | J12           |   |   | P34/SCKB1/CS0 <sup>Note 2</sup>  |
| INTP12                   | 53      | 62  | 90 | H12           |   |   | P37/SCKB2/ASTB <sup>Note 2</sup>   |
| INTP13                   | 40      | 49  | 77 | M14           |   |   | P42/SCKB0/DDI <sup>Note 1</sup>  |
| INTP14                   | 41      | 50  | 78 | M13           |   |   | P43/TECR1/TIT10/TOT10  |
| INTP15                   | 42      | 51  | 79 | L14           |   |   | P44/TENC10/EVTT1/ $\overline{\text{WAIT}}$ <sup>Note 2</sup>                                       |
| INTP16                   | 43      | 52  | 80 | L13           | P45/TENC11/TIT11/TOT11/ $\overline{\text{WR1}}$ <sup>Note 2</sup> |   |  |
| INTP17                   | 44      | 53  | 81 | L12           | P46/TOA40/TIA40/ $\overline{\text{WR0}}$ <sup>Note 2</sup>        |   |  |
| INTP18                   | 45      | 54  | 82 | K14           | P47/TOA41/TIA41/ $\overline{\text{RD}}$ <sup>Note 2</sup>         |   |  |

**Notes 1.** V850E/IG3 only

2.  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only
3.  $\mu$ PD70F3454F1-DA9-A only
4. A2, A13, B13, B14, C12, D4 to D11, E11, F11, G11, H11, J11, K11, L5 to L11, M7, M8, M12, N1, N13, N14, P2, P13
5. H4, J3

**Remark** IF3: V850E/IF3

IG3: V850E/IG3

GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)

GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)

GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

| Pin Name                          | Pin No. |    |     |        | I/O    | Function  | Alternate-Function Pin  |
|-----------------------------------|---------|----|-----|--------|--------|---|---|
|                                   | IF3     |    | IG3 |        |        |   |   |
|                                   | GC      | GC | GF  | F1     |        |   |   |
| $\overline{RD}$ <sup>Note 2</sup> | –       | 54 | –   | K14    | Output | Read strobe output of external data bus   | P47/TOA41/TIA41/INTP18  |
| REGC0                             | 31      | 36 | 64  | N6, P6 | –      | Regulator output stabilization capacitance connection                           | –   |
| REGC1                             | 67      | 86 | 14  | A8, B8 | –      |   | –   |
| $\overline{RESET}$                | 35      | 40 | 68  | P9     | Input  | System reset input  | –   |
| RXDA0                             | 38      | 47 | 75  | P12    | Input  | Serial receive data input of UARТА0 to UARТА2                                   | P40/SIB0  |
| RXDA1                             | 46      | 55 | 83  | K13    |        |   | P30/SCL   |
| RXDA2                             | 48      | 57 | 85  | J14    |        |   | P32/SIB1/ $\overline{CS1}$ <sup>Note 1</sup>  |
| RXDB                              | 51      | 60 | 88  | H14    | Input  | Serial receive data input of UARТB0   | P35/SIB2  |
| $\overline{SCKB0}$                | 40      | 49 | 77  | M14    | I/O    | Serial clock I/O of CSIB0 to CSIB2  | P42/INTP13/DD1 <sup>Note 2</sup>  |
| $\overline{SCKB1}$                | 50      | 59 | 87  | J12    |        |   | P34/INTP11/ $\overline{CS0}$ <sup>Note 1</sup>  |
| $\overline{SCKB2}$                | 53      | 62 | 90  | H12    |        |   | P37/INTP12/ASTB <sup>Note 1</sup>   |
| SCL                               | 46      | 55 | 83  | K13    | I/O    | Serial clock I/O  | P30/RXDA1   |
| SDA                               | 47      | 56 | 84  | K12    | I/O    | Serial transmit/receive data I/O  | P31/TXDA1   |
| SIB0                              | 38      | 47 | 75  | P12    | Input  | Serial receive data input of CSIB0 to CSIB2                                     | P40/RXDA0   |
| SIB1                              | 48      | 57 | 85  | J14    |        |   | P32/RXDA2/ $\overline{CS1}$ <sup>Note 1</sup>   |
| SIB2                              | 51      | 60 | 88  | H14    |        |   | P35/RXDB  |
| SOB0                              | 39      | 48 | 76  | N12    | Output | Serial transmit data output of CSIB0 to CSIB2                                   | P41/TXDA0/DCK <sup>Note 2</sup>   |
| SOB1                              | 49      | 58 | 86  | J13    |        |   | P33/TXDA2   |
| SOB2                              | 52      | 61 | 89  | H13    |        |   | P36/TXDB  |
| TECRO <sup>Note 2</sup>           | –       | 84 | 12  | C8     | Input  | Encoder clear input of TMT0, TMT1   | P04 <sup>Note 2</sup> /TIT00 <sup>Note 2</sup> /TOT00 <sup>Note 2</sup> /INTP04 <sup>Note 2</sup> |
| TECR1                             | 41      | 50 | 78  | M13    |        |   | P43/TIT10/TOT10/INTP14  |
| TENC00 <sup>Note 2</sup>          | –       | 83 | 11  | C9     | Input  | Encoder input of TMT0, TMT1   | P05 <sup>Note 2</sup> /EVTT0 <sup>Note 2</sup> /INTP05 <sup>Note 2</sup>                          |
| TENC01 <sup>Note 2</sup>          | –       | 82 | 10  | C10    |        |   | P06 <sup>Note 2</sup> /TIT01 <sup>Note 2</sup> /TOT01 <sup>Note 2</sup> /INTP06 <sup>Note 2</sup> |
| TENC10                            | 42      | 51 | 79  | L14    |        |   | P44/EVTT1/INTP15/ $\overline{WAIT}$ <sup>Note 1</sup>   |
| TENC11                            | 43      | 52 | 80  | L13    |        |   | P45/TIT11/TOT11/INTP16/ $\overline{WR1}$ <sup>Note 1</sup>  |
| TIA20                             | 70      | 91 | 19  | A6     | Input  | External event count input/external trigger input/capture trigger input of TAA2 | P00/TOA20/TOA2OFF/INTP00  |

- Notes 1.**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only  
**2.** V850E/IG3 only

**Remark** IF3: V850E/IF3  
 IG3: V850E/IG3  
 GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)  
 GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)  
 GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)  
 F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

CHAPTER 2 PIN FUNCTIONS

(6/8)

| Pin Name                  | Pin No. |    |     |     | I/O  | Function  | Alternate-Function Pin   |
|---------------------------|---------|----|-----|-----|--|---|--|
|                           | IF3     |    | IG3 |     |  |   |  |
|                           | GC      | GC | GF  | F1  |  |   |  |
| TIA21                     | 69      | 90 | 18  | B6  | Input  | Capture trigger input of TAA2   | P01/TOA21/INTP01   |
| TIA30 <sup>Note 2</sup>   | –       | 89 | 17  | C6  | Input  | External event count input/external trigger input/capture trigger input of TAA3 | P02 <sup>Note 2</sup> /TOA30 <sup>Note 2</sup> /TOA3OFF <sup>Note 2</sup><br>/INTP02 <sup>Note 2</sup> |
| TIA31 <sup>Note 2</sup>   | –       | 88 | 16  | C7  |  | Capture trigger input of TAA3   | P03 <sup>Note 2</sup> /TOA31 <sup>Note 2</sup> /INTP03 <sup>Note 2</sup>                               |
| TIA40                     | 44      | 53 | 81  | L12 | Input  | External event count input/external trigger input/capture trigger input of TAA4 | P46/TOA40/INTP17/WR0 <sup>Note 1</sup>   |
| TIA41                     | 45      | 54 | 82  | K14 | Input  | Capture trigger input of TAA4   | P47/TOA41/INTP18/RD <sup>Note 1</sup>  |
| TIB00                     | 75      | 96 | 24  | B4  | Input  | Capture trigger input of TAB0, TAB1   | P13/TOB0B2/A3 <sup>Note 1</sup>  |
| TIB01                     | 78      | 99 | 27  | B3  |  |   | P10/TOB0T1/TOB01/A0 <sup>Note 1</sup>  |
| TIB02                     | 77      | 98 | 26  | C3  |  |   | P11/TOB0B1/TOB02/A1 <sup>Note 1</sup>  |
| TIB03                     | 76      | 97 | 25  | A4  |  |   | P12/TOB0T2/TOB03/A2 <sup>Note 1</sup>  |
| TIB10                     | 26      | 31 | 59  | N4  |  |   | P23/TOB1B2   |
| TIB11                     | 23      | 28 | 56  | P3  |  |   | P20/TOB1T1/TOB11   |
| TIB12                     | 24      | 29 | 57  | N3  |  |   | P21/TOB1B1/TOB12   |
| TIB13                     | 25      | 30 | 58  | P4  |  |   | P22/TOB1T2/TOB13   |
| TIT00 <sup>Note 2</sup>   | –       | 84 | 12  | C8  |  |   | Input  |
| TIT01 <sup>Note 2</sup>   | –       | 82 | 10  | C10 | P06 <sup>Note 2</sup> /TENC01 <sup>Note 2</sup> /TOT01 <sup>Note 2</sup> /INTP06 <sup>Note 2</sup> |   |  |
| TIT10                     | 41      | 50 | 78  | M13 | Input  | Capture trigger input of TIT1   | P43/TECR1/TOT10/INTP14   |
| TIT11                     | 43      | 52 | 80  | L13 |  |   | P45/TENC11/TOT11/INTP16/WR1 <sup>Note 1</sup>  |
| TOA20                     | 70      | 91 | 19  | A6  | Output   | Timer output of TAA2  | P00/TIA20/TOA2OFF/INTP00   |
| TOA21                     | 69      | 90 | 18  | B6  |  |   | P01/TIA21/INTP01   |
| TOA2OFF                   | 70      | 91 | 19  | A6  | Input  | High-impedance output control signal input                                      | P00/TOA20/TIA20/INTP00   |
| TOA30 <sup>Note 2</sup>   | –       | 89 | 17  | C6  | Output   | Timer output of TAA3  | P02 <sup>Note 2</sup> /TIA30 <sup>Note 2</sup> /TOA3OFF <sup>Note 2</sup> /INTP02 <sup>Note 2</sup>    |
| TOA31 <sup>Note 2</sup>   | –       | 88 | 16  | C7  |  |   | P03 <sup>Note 2</sup> /TIA31 <sup>Note 2</sup> /INTP03 <sup>Note 2</sup>                               |
| TOA3OFF <sup>Note 2</sup> | –       | 89 | 17  | C6  | Input  | High-impedance output control signal input                                      | P02 <sup>Note 2</sup> /TOA30 <sup>Note 2</sup> /TIA30 <sup>Note 2</sup> /INTP02 <sup>Note 2</sup>      |
| TOA40                     | 44      | 53 | 81  | L12 | Output   | Timer output of TAA4  | P46/TIA40/INTP17/WR0 <sup>Note 1</sup>   |
| TOA41                     | 45      | 54 | 82  | K14 |  |   | P47/TIA41/INTP18/RD <sup>Note 1</sup>  |

Notes 1.  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

2. V850E/IG3 only

Remark IF3: V850E/IF3

IG3: V850E/IG3

GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)

GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)

GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

CHAPTER 2 PIN FUNCTIONS

(7/8)

| Pin Name                | Pin No. |    |     |     | I/O    | Function   | Alternate-Function Pin  |
|-------------------------|---------|----|-----|-----|--------|--|---|
|                         | IF3     |    | IG3 |     |        |  |   |
|                         | GC      | GC | GF  | F1  |        |  |   |
| TOB00                   | 71      | 92 | 20  | C5  | Output | Timer output of TAB0   | P17/INTP09/A7 <sup>Note 1</sup>   |
| TOB01                   | 78      | 99 | 27  | B3  |        |  | P10/TOB0T1/TIB01/A0 <sup>Note 1</sup>   |
| TOB02                   | 77      | 98 | 26  | C3  |        |  | P11/TOB0B1/TIB02/A1 <sup>Note 1</sup>   |
| TOB03                   | 76      | 97 | 25  | A4  |        |  | P12/TOB0T2/TIB03/A2 <sup>Note 1</sup>   |
| TOB0B1                  | 77      | 98 | 26  | C3  | Output | Pulse signal output for 6-phase PWM low arm of TAB0            | P11/TIB02/TOB02/A1 <sup>Note 1</sup>  |
| TOB0B2                  | 75      | 96 | 24  | B4  |        |  | P13/TIB00/A3 <sup>Note 1</sup>  |
| TOB0B3                  | 73      | 94 | 22  | A5  |        |  | P15/TRGB0/A5 <sup>Note 1</sup>  |
| TOB0OFF                 | 72      | 93 | 21  | B5  | Input  | 6-phase PWM high-impedance output control signal input of TAB0 | P16/INTP08/ADTRG0/INTADT0/A6 <sup>Note 1</sup>  |
| TOB0T1                  | 78      | 99 | 27  | B3  | Output | Pulse signal output for 6-phase PWM high arm of TAB0           | P10/TIB01/TOB01/A0 <sup>Note 1</sup>  |
| TOB0T2                  | 76      | 97 | 25  | A4  |        |  | P12/TIB03/TOB03/A2 <sup>Note 1</sup>  |
| TOB0T3                  | 74      | 95 | 23  | C4  |        |  | P14/EVTB0/A4 <sup>Note 1</sup>  |
| TOB10                   | 29      | 34 | 62  | M6  | Output | Timer output of TAB1   | P26/TOB1OFF/INTP10/ADTRG1/INTADT1   |
| TOB11                   | 23      | 28 | 56  | P3  |        |  | P20/TOB1T1/TIB11  |
| TOB12                   | 24      | 29 | 57  | N3  |        |  | P21/TOB1B1/TIB12  |
| TOB13                   | 25      | 30 | 58  | P4  |        |  | P22/TOB1T2/TIB13  |
| TOB1B1                  | 24      | 29 | 57  | N3  | Output | Pulse signal output for 6-phase PWM low arm of TAB1            | P21/TIB12/TOB12   |
| TOB1B2                  | 26      | 31 | 59  | N4  |        |  | P23/TIB10   |
| TOB1B3                  | 28      | 33 | 61  | M5  |        |  | P25/TRGB1   |
| TOB1OFF                 | 29      | 34 | 62  | M6  | Input  | 6-phase PWM high-impedance output control signal input of TAB1 | P26/TOB10/INTP10/ADTRG1/INTADT1   |
| TOB1T1                  | 23      | 28 | 56  | P3  | Output | Pulse signal output for 6-phase PWM high arm of TAB1           | P20/TIB11/TOB11   |
| TOB1T2                  | 25      | 30 | 58  | P4  |        |  | P22/TIB13/TOB13   |
| TOB1T3                  | 27      | 32 | 60  | M4  |        |  | P24/EVTB1   |
| TOT0 <sup>Note 2</sup>  | –       | 84 | 12  | C8  | Output | Timer output of TMT0, TMT1                                     | P04 <sup>Note 2</sup> /TECR0 <sup>Note 2</sup> /TIT00 <sup>Note 2</sup> /INTP04 <sup>Note 2</sup>     |
| TOT01 <sup>Note 2</sup> | –       | 82 | 10  | C10 |        |  | P06 <sup>Note 2</sup> /TENC01 <sup>Note 2</sup> /TIT01 <sup>Note 2</sup><br>/INTP06 <sup>Note 2</sup> |
| TOT10                   | 41      | 50 | 78  | M13 |        |  | P43/TECR1/TIT10/INTP14  |
| TOT11                   | 43      | 52 | 80  | L13 |        |  | P45/TENC11/TIT11/INTP16/WR1 <sup>Note 1</sup>   |

**Notes 1.**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**2.** V850E/IG3 only

**Remark** IF3: V850E/IF3

IG3: V850E/IG3

GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)

GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)

GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

| Pin Name             | Pin No. |     |    |        | I/O    | Function  | Alternate-Function Pin          |
|----------------------|---------|-----|----|--------|--------|---|---------------------------------|
|                      | IF3     | IG3 |    |        |        |   |                                 |
|                      | GC      | GC  | GF | F1     |        |   |                                 |
| TRGB0                | 73      | 94  | 22 | A5     | Input  | External trigger input of TAB0, TAB1            | P15/TOB0B3/A5 <sup>Note 1</sup> |
| TRGB1                | 28      | 33  | 61 | M5     |        |   | P25/TOB1B3                      |
| TXDA0                | 39      | 48  | 76 | N12    | Output | Serial transmit data output of UARTA0 to UARTA2 | P41/SOB0/DCK <sup>Note 2</sup>  |
| TXDA1                | 47      | 56  | 84 | K12    |        |   | P31/SDA                         |
| TXDA2                | 49      | 58  | 86 | J13    |        |   | P33/SOB1                        |
| TXDB                 | 52      | 61  | 89 | H13    | Output | Serial transmit data output of UARB0            | P36/SOB2                        |
| V <sub>DD0</sub>     | 30      | 35  | 63 | N5, P5 | -      | Positive power supply for internal unit         | -                               |
| V <sub>DD1</sub>     | 68      | 87  | 15 | A7, B7 |        |   | -                               |
| V <sub>SS0</sub>     | 32      | 37  | 65 | N7, P7 | -      | Ground potential for internal unit              | -                               |
| V <sub>SS1</sub>     | 66      | 85  | 13 | A9, B9 |        |   | -                               |
| WAIT <sup>Note</sup> | -       | 51  | -  | L14    | Input  | External wait request input                     | P44/TENC10/EVTT1/INTP15         |
| WRO <sup>Note</sup>  | -       | 53  | -  | L12    | Output | Write strobe output of external data bus        | P46/TOA40/TIA40/INTP17          |
| WR1 <sup>Note</sup>  | -       | 52  | -  | L13    |        |   | P45/TENC11/TIT11/TOT11/INTP16   |
| X1                   | 33      | 38  | 66 | P8     | Input  | Resonator connection pin for system clock       | -                               |
| X2                   | 34      | 39  | 67 | N8     |        |   | -                               |

**Note**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**Remark** IF3: V850E/IF3

IG3: V850E/IG3

GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)

GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)

GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

## 2.2 Pin Status

The operation statuses of pins in the various operation modes are described below.

**Table 2-2. Pin Operation Status in Operation Modes**

| Pin \ Operating Status   | Reset                  | HALT Mode/<br>During DMA Transfer | IDLE Mode/<br>STOP Mode | Idle State |
|--|------------------------|-----------------------------------|-------------------------|------------|
| AD0 to AD15 <sup>Note 1</sup> (PDL0 to PDL15)                                  | Hi-Z <sup>Note 2</sup> | Operating                         | Hi-Z                    | Held       |
| A0 to A7 <sup>Note 1</sup> (P10 to P17)  | Hi-Z <sup>Note 2</sup> | Operating                         | Hi-Z                    | Held       |
| $\overline{CS0}^{\text{Note 1}}$ , $\overline{CS1}^{\text{Note 1}}$ (P34, P32) | Hi-Z <sup>Note 2</sup> | Operating                         | H                       | Held       |
| $\overline{WR0}^{\text{Note 1}}$ , $\overline{WR1}^{\text{Note 1}}$ (P46, P45) | Hi-Z <sup>Note 2</sup> | Operating                         | H                       | H          |
| $\overline{RD}^{\text{Note 1}}$ (P47)  | Hi-Z <sup>Note 2</sup> | Operating                         | H                       | H          |
| $\overline{ASTB}^{\text{Note 1}}$ (P37)  | Hi-Z <sup>Note 2</sup> | Operating                         | H                       | H          |
| $\overline{WAIT}^{\text{Note 1}}$ (P44)  | Hi-Z <sup>Note 2</sup> | Operating                         | –                       | –          |
| CLKOUT <sup>Note 1</sup> (P07)   | Hi-Z <sup>Note 2</sup> | Operating                         | Held                    | Operating  |

**Notes 1.**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

2. Since the bus control pin is also used as a port pin, it is initialized to the input mode (port mode) after reset.

**Remark** Hi-Z: High impedance

Held: The state during the immediately preceding external bus cycle is held.

H: High-level output

–: Input without sampling (not acknowledged)



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

It is recommended that 1 to 10 kΩ resistors be used when connecting to AV<sub>SS2</sub>, EV<sub>DD0</sub>, EV<sub>DD1</sub>, EV<sub>DD2</sub> (V850E/IG3 only) or EV<sub>SS0</sub>, EV<sub>SS1</sub>, EV<sub>SS2</sub> (V850E/IG3 only) via resistors.

(1/4)

| Pin                   | Alternate-Function Pin Name   | Pin No. |    |     |     | I/O Circuit Type  | Recommended Connection |
|-----------------------|---|---------|----|-----|-----|---|------------------------|
|                       |   | IF3     |    | IG3 |     |   |                        |
|                       |   | GC      | GC | GF  | F1  |   |                        |
| P00                   | TOA20/TIA20/TOA2OFF/INTP00  | 70      | 91 | 19  | A6  | 5-AH<br>Input: Independently connect to EV <sub>DD0</sub> , EV <sub>DD1</sub> , EV <sub>DD2</sub> <sup>Note 1</sup> or EV <sub>SS0</sub> , EV <sub>SS1</sub> , EV <sub>SS2</sub> <sup>Note 1</sup> via a resistor.<br>Output: Leave open. |                        |
| P01                   | TOA21/TIA21/INTP01  | 69      | 90 | 18  | B6  |   |                        |
| P02 <sup>Note 1</sup> | TOA30 <sup>Note 1</sup> /TIA30 <sup>Note 1</sup> /TOA3OFF <sup>Note 1</sup> /INTP02 <sup>Note 1</sup> | –       | 89 | 17  | C6  |   |                        |
| P03 <sup>Note 1</sup> | TOA31 <sup>Note 1</sup> /TIA31 <sup>Note 1</sup> /INTP03 <sup>Note 1</sup>                            | –       | 88 | 16  | C7  |   |                        |
| P04 <sup>Note 1</sup> | TECR0 <sup>Note 1</sup> /TIT00 <sup>Note 1</sup> /TOT00 <sup>Note 1</sup> /INTP04 <sup>Note 1</sup>   | –       | 84 | 12  | C8  |   |                        |
| P05 <sup>Note 1</sup> | TENC00 <sup>Note 1</sup> /EVTT0 <sup>Note 1</sup> /INTP05 <sup>Note 1</sup>                           | –       | 83 | 11  | C9  |   |                        |
| P06 <sup>Note 1</sup> | TENC01 <sup>Note 1</sup> /TIT01 <sup>Note 1</sup> /TOT01 <sup>Note 1</sup> /INTP06 <sup>Note 1</sup>  | –       | 82 | 10  | C10 |   |                        |
| P07 <sup>Note 1</sup> | INTP07 <sup>Note 1</sup> /CLKOUT <sup>Note 2</sup>  | –       | 63 | 91  | G12 |   |                        |
| P10                   | TOB0T1/TIB01/TOB01/A0 <sup>Note 2</sup>   | 78      | 99 | 27  | B3  |   |                        |
| P11                   | TOB0B1/TIB02/TOB02/A1 <sup>Note 2</sup>   | 77      | 98 | 26  | C3  |   |                        |
| P12                   | TOB0T2/TIB03/TOB03/A2 <sup>Note 2</sup>   | 76      | 97 | 25  | A4  |   |                        |
| P13                   | TOB0B2/TIB00/A3 <sup>Note 2</sup>   | 75      | 96 | 24  | B4  |   |                        |
| P14                   | TOB0T3/EVTB0/A4 <sup>Note 2</sup>   | 74      | 95 | 23  | C4  |   |                        |
| P15                   | TOB0B3/TRGB0/A5 <sup>Note 2</sup>   | 73      | 94 | 22  | A5  |   |                        |
| P16                   | TOB0OFF/INTP08/ADTRG0/INTADT0/A6 <sup>Note 2</sup>  | 72      | 93 | 21  | B5  |   |                        |
| P17                   | TOB00/INTP09/A7 <sup>Note 2</sup>   | 71      | 92 | 20  | C5  |   |                        |
| P20                   | TOB1T1/TIB11/TOB11  | 23      | 28 | 56  | P3  |   |                        |
| P21                   | TOB1B1/TIB12/TOB12  | 24      | 29 | 57  | N3  |   |                        |
| P22                   | TOB1T2/TIB13/TOB13  | 25      | 30 | 58  | P4  |   |                        |
| P23                   | TOB1B2/TIB10  | 26      | 31 | 59  | N4  |   |                        |
| P24                   | TOB1T3/EVTB1  | 27      | 32 | 60  | M4  |   |                        |
| P25                   | TOB1B3/TRGB1  | 28      | 33 | 61  | M5  |   |                        |
| P26                   | TOB10/TOB1OFF/INTP10/ADTRG1/INTADT1   | 29      | 34 | 62  | M6  |   |                        |
| P27                   | DMS <sup>Note 1</sup>   | 36      | 45 | 73  | M11 |   |                        |

**Notes 1.** V850E/IG3 only

**2.** μPD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**Remark** IF3: V850E/IF3

IG3: V850E/IG3

GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)

GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)

GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

CHAPTER 2 PIN FUNCTIONS

(2/4)

| Pin                   | Alternate-Function Pin Name                                     | Pin No. |    |     |     | I/O Circuit Type | Recommended Connection  |
|-----------------------|---|---------|----|-----|-----|------------------|---|
|                       |   | IF3     |    | IG3 |     |                  |   |
|                       |   | GC      | GC | GF  | F1  |                  |   |
| P30                   | RXDA1/SCL   | 46      | 55 | 83  | K13 | 5-AH             | Input: Independently connect to EV <sub>DD0</sub> , EV <sub>DD1</sub> , EV <sub>DD2</sub> <sup>Note 2</sup> or EV <sub>SS0</sub> , EV <sub>SS1</sub> , EV <sub>SS2</sub> <sup>Note 2</sup> via a resistor.                        |
| P31                   | TXDA1/SDA   | 47      | 56 | 84  | K12 |                  |   |
| P32                   | SIB1/RXDA2/ $\overline{CS1}$ <sup>Note 1</sup>                  | 48      | 57 | 85  | J14 |                  |   |
| P33                   | SOB1/TXDA2  | 49      | 58 | 86  | J13 | 5-AG             | Output: Leave open.   |
| P34                   | $\overline{SCKB1}$ /INTP11/ $\overline{CS0}$ <sup>Note 1</sup>  | 50      | 59 | 87  | J12 | 5-AH             |   |
| P35                   | SIB2/RXDB   | 51      | 60 | 88  | H14 |                  |   |
| P36                   | SOB2/TXDB   | 52      | 61 | 89  | H13 | 5-AG             |   |
| P37                   | $\overline{SCKB2}$ /INTP12/ $\overline{ASTB}$ <sup>Note 1</sup> | 53      | 62 | 90  | H12 | 5-AH             |   |
| P40                   | SIB0/RXDA0  | 38      | 47 | 75  | P12 |                  |   |
| P41                   | SOB0/TXDA0/DCK <sup>Note 2</sup>                                | 39      | 48 | 76  | N12 | 5-AG             |   |
| P42                   | $\overline{SCKB0}$ /INTP13/DDI <sup>Note 2</sup>                | 40      | 49 | 77  | M14 | 5-AH             |   |
| P43                   | TECR1/TIT10/TOT10/INTP14  | 41      | 50 | 78  | M13 |                  |   |
| P44                   | TENC10/EVTT1/INTP15/ $\overline{WAIT}$ <sup>Note 1</sup>        | 42      | 51 | 79  | L14 |                  |   |
| P45                   | TENC11/TIT11/TOT11/INTP16/ $\overline{WR1}$ <sup>Note 1</sup>   | 43      | 52 | 80  | L13 |                  |   |
| P46                   | TOA40/TIA40/INTP17/ $\overline{WR0}$ <sup>Note 1</sup>          | 44      | 53 | 81  | L12 |                  |   |
| P47                   | TOA41/TIA41/INTP18/ $\overline{RD}$ <sup>Note 1</sup>           | 45      | 54 | 82  | K14 |                  |   |
| P70                   | ANI20   | 20      | 25 | 53  | L1  | 11-G             | Independently connect to AV <sub>SS2</sub> via a resistor.  |
| P71                   | ANI21   | 19      | 24 | 52  | L2  |                  |   |
| P72                   | ANI22   | 18      | 23 | 51  | L3  |                  |   |
| P73                   | ANI23   | 17      | 22 | 50  | K1  |                  |   |
| P74 <sup>Note 2</sup> | ANI24 <sup>Note 2</sup>   | –       | 21 | 49  | K2  |                  |   |
| P75 <sup>Note 2</sup> | ANI25 <sup>Note 2</sup>   | –       | 20 | 48  | K3  |                  |   |
| P76 <sup>Note 2</sup> | ANI26 <sup>Note 2</sup>   | –       | 19 | 47  | K4  |                  |   |
| P77 <sup>Note 2</sup> | ANI27 <sup>Note 2</sup>   | –       | 18 | 46  | J4  |                  |   |
| PDL0                  | AD0 <sup>Note 1</sup>   | 65      | 81 | 9   | A10 | 5-AG             | Input: Independently connect to EV <sub>DD0</sub> , EV <sub>DD1</sub> , EV <sub>DD2</sub> <sup>Note 2</sup> or EV <sub>SS0</sub> , EV <sub>SS1</sub> , EV <sub>SS2</sub> <sup>Note 2</sup> via a resistor.<br>Output: Leave open. |
| PDL1                  | AD1 <sup>Note 1</sup>   | 64      | 80 | 8   | B10 |                  |   |
| PDL2                  | AD2 <sup>Note 1</sup>   | 63      | 79 | 7   | A11 |                  |   |
| PDL3                  | AD3 <sup>Note 1</sup>   | 62      | 78 | 6   | B11 |                  |   |

- Notes**
1.  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only
  2. V850E/IG3 only

**Remark**

IF3: V850E/IF3  
 IG3: V850E/IG3  
 GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)  
 GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)  
 GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)  
 F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

CHAPTER 2 PIN FUNCTIONS

(3/4)

| Pin                     | Alternate-Function Pin Name  | Pin No. |     |     |     | I/O Circuit Type | Recommended Connection  |     |   |
|-------------------------|------------------------------|---------|-----|-----|-----|------------------|---|-----|---|
|                         |                              | IF3     | IG3 |     |     |                  |   |     |   |
|                         |                              | GC      | GC  | GF  | F1  |                  |   |     |   |
| PDL4                    | AD4 <sup>Note 1</sup>        | 61      | 77  | 5   | C11 | 5-AG             | Input: Independently connect to EV <sub>DD0</sub> , EV <sub>DD1</sub> , EV <sub>DD2</sub> <sup>Note 2</sup> or EV <sub>SS0</sub> , EV <sub>SS1</sub> , EV <sub>SS2</sub> <sup>Note 2</sup> via a resistor.<br>Output: Leave open. |     |   |
| PDL5                    | AD5 <sup>Note 1</sup> /FLMD1 | 60      | 76  | 4   | A12 |                  |   |     |   |
| PDL6                    | AD6 <sup>Note 1</sup>        | 59      | 75  | 3   | B12 |                  |   |     |   |
| PDL7                    | AD7 <sup>Note 1</sup>        | 58      | 74  | 2   | C14 |                  |   |     |   |
| PDL8                    | AD8 <sup>Note 1</sup>        | 57      | 73  | 1   | C13 |                  |   |     |   |
| PDL9                    | AD9 <sup>Note 1</sup>        | 56      | 72  | 100 | D14 |                  |   |     |   |
| PDL10 <sup>Note 2</sup> | AD10 <sup>Note 1</sup>       | –       | 71  | 99  | D13 |                  |   |     |   |
| PDL11 <sup>Note 2</sup> | AD11 <sup>Note 1</sup>       | –       | 70  | 98  | D12 |                  |   |     |   |
| PDL12 <sup>Note 2</sup> | AD12 <sup>Note 1</sup>       | –       | 69  | 97  | E14 |                  |   |     |   |
| PDL13 <sup>Note 2</sup> | AD13 <sup>Note 1</sup>       | –       | 68  | 96  | E13 |                  |   |     |   |
| PDL14 <sup>Note 2</sup> | AD14 <sup>Note 1</sup>       | –       | 67  | 95  | E12 |                  |   |     |   |
| PDL15 <sup>Note 2</sup> | AD15 <sup>Note 1</sup>       | –       | 66  | 94  | F12 |                  |   |     |   |
| ANI00                   | ANI05                        | 1       | 2   | 30  | D1  |                  |   | 7-C | Connect to AV <sub>SS0</sub> or AV <sub>SS1</sub> . |
| ANI01                   | –                            | 2       | 3   | 31  | D2  |                  |   |     |   |
| ANI02                   | –                            | 3       | 4   | 32  | E3  |                  |   |     |   |
| ANI03                   | CREF0L                       | 4       | 5   | 33  | F3  |                  |   |     |   |
| ANI04                   | CREF0F                       | 5       | 6   | 34  | F4  |                  |   |     |   |
| ANI10                   | ANI15                        | 16      | 17  | 45  | J1  |                  |   |     |   |
| ANI11                   | ANI16                        | 15      | 16  | 44  | J2  |                  |   |     |   |
| ANI12                   | ANI17                        | 14      | 15  | 43  | H3  |                  |   |     |   |
| ANI13                   | CREF1L                       | 13      | 14  | 42  | G3  |                  |   |     |   |
| ANI14                   | CREF1F                       | 12      | 13  | 41  | G4  |                  |   |     |   |

**Notes 1.**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**2.** V850E/IG3 only

**Remark** IF3: V850E/IF3

IG3: V850E/IG3

GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)

GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)

GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

| Pin  | Alternate-Function Pin Name | Pin No. |     |               |  | I/O Circuit Type | Recommended Connection  |
|--|-----------------------------|---------|-----|---------------|--|------------------|---|
|  |                             | IF3     | IG3 |               |  |                  |   |
|  |                             | GC      | GC  | GF            | F1   |                  |   |
| DDO <sup>Note 1</sup>                      | –                           | –       | 44  | 72            | M10  | 3-C              | Leave open (output when $\overline{\text{DRST}}$ is high-level).          |
| $\overline{\text{DRST}}$ <sup>Note 1</sup> | –                           | –       | 43  | 71            | M9   | 2-M              | Leave open (on-chip pull-down resistor).                                  |
| FLMD0                                      | –                           | 37      | 46  | 74            | N9   | 2                | –   |
| IC0 <sup>Note 2</sup>                      | –                           | –       | –   | –             | E5   | –                | Connect to EV <sub>SS0</sub> , EV <sub>SS1</sub> , or EV <sub>SS2</sub> . |
| IC1 <sup>Note 2</sup>                      | –                           | –       | –   | <b>Note 3</b> | Leave open   |                  |   |
| IC2 <sup>Note 2</sup>                      | –                           | –       | –   | <b>Note 4</b> | Independently connect to AV <sub>SS2</sub> via a resistor. |                  |   |
| $\overline{\text{RESET}}$                  | –                           | 35      | 40  | 68            | P9   | 2                | Pull this pin up when the power-on-clear circuit (POC) is used.           |

**Notes 1.** V850E/IG3 only

**2.**  $\mu$ PD70F3454F1-DA9-A only

**3.** A2, A13, B13, B14, C12, D4 to D11, E11, F11, G11, H11, J11, K11, L5 to L11, M7, M8, M12, N1, N13, N14, P2, P13

**4.** H4, J3

**Remark** IF3: V850E/IF3

IG3: V850E/IG3

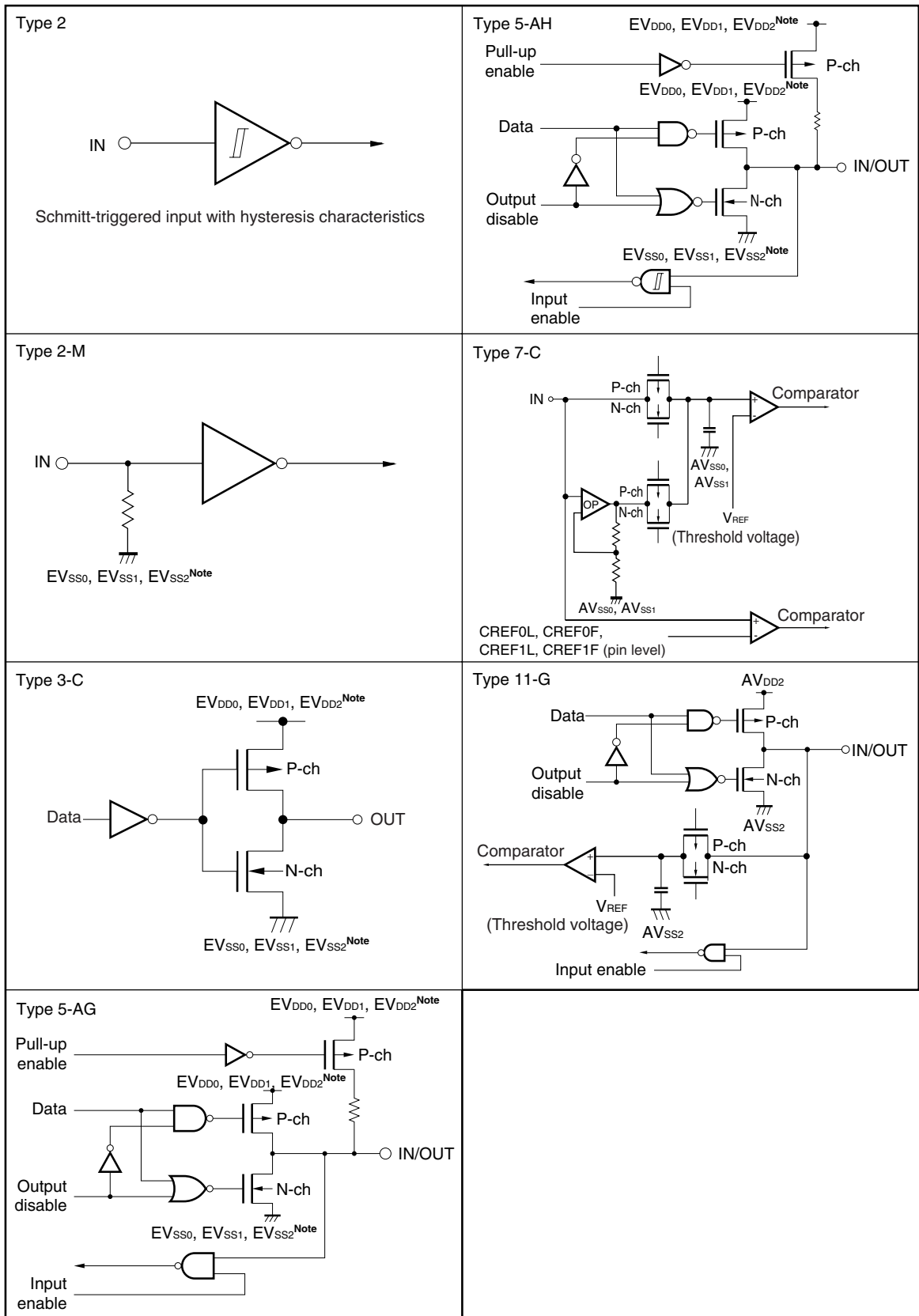
GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)

GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)

GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

2.4 Pin I/O Circuits



**Note** V850E/IG3 only

## CHAPTER 3 CPU FUNCTION

The CPU of the V850E/IF3 and V850E/IG3 is based on RISC architecture and executes almost all the instructions in one clock cycle using 5-stage pipeline control.

### 3.1 Features

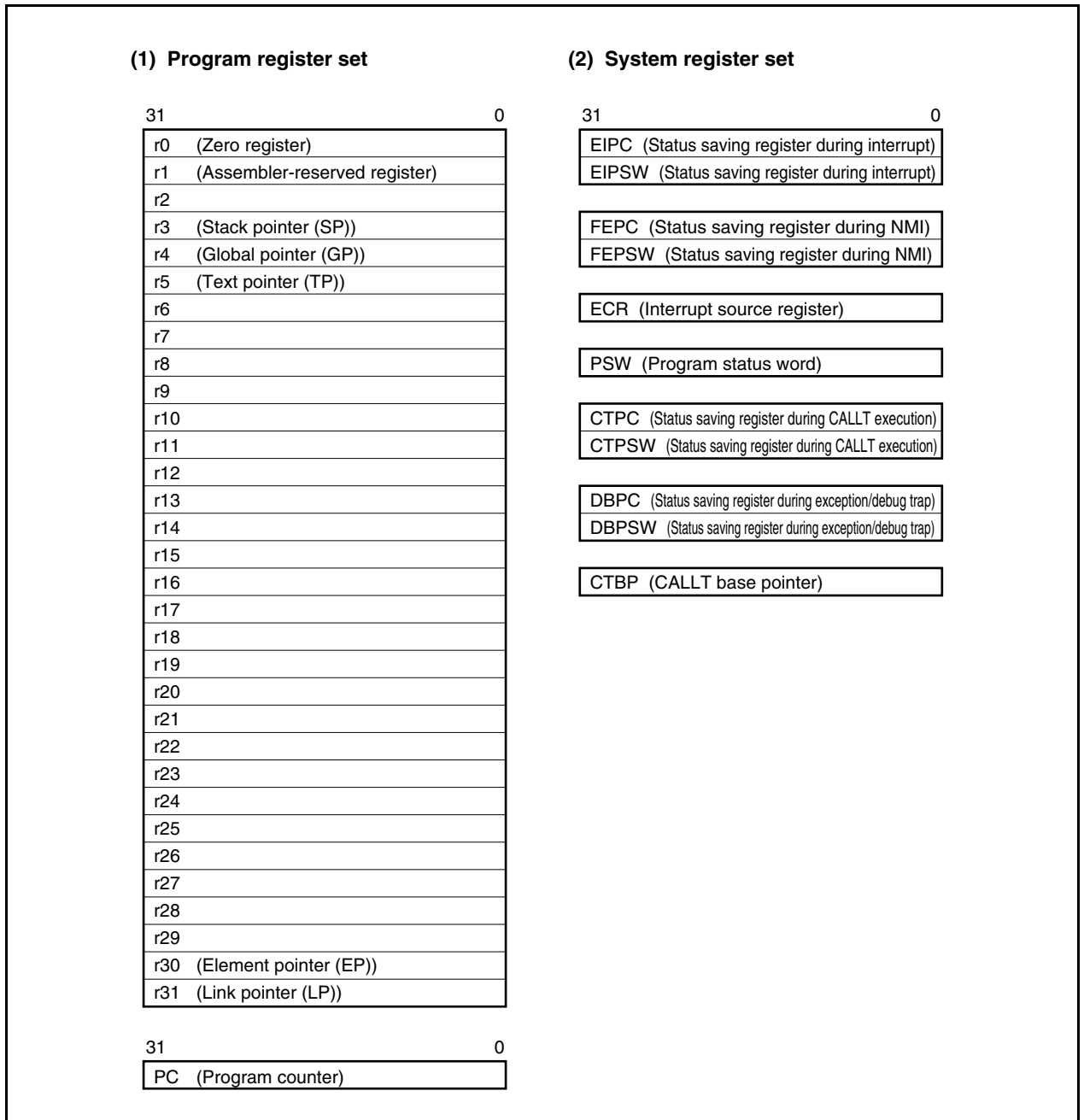
- Minimum instruction execution time: 15.6 ns (@ 64 MHz internal operation)
- Thirty-two 32-bit general-purpose registers
- Internal 32-bit architecture
- Five-stage pipeline control
- Multiply/divide instructions
- Saturated operation instructions
- One-clock 32-bit shift instruction
- Load/store instruction with long/short instruction format
- Four types of bit manipulation instructions
  - SET1
  - CLR1
  - NOT1
  - TST1

### 3.2 CPU Register Set

The registers of the V850E/IF3 and V850E/IG3 can be classified into two categories: a general-purpose program register set and a dedicated system register set. All the registers have a 32-bit width.

For details, refer to **V850E1 Architecture User's Manual**.

Figure 3-1. CPU Register Set



**3.2.1 Program register set**

The program register set includes 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 as a data variable or address variable.

However, r0 and r30 are implicitly used by instructions, and care must be exercised when using these registers. r0 is a register that always holds 0, and is used for operations using 0 and offset 0 addressing. r30 is used, by means of the SLD and SST instructions, as a base pointer for when memory is accessed. Also, r1, r3 to r5, and r31 are implicitly used by the assembler and C compiler. Therefore, before using these registers, their contents must be saved so that they are not lost. The contents must be restored to the registers after the registers have been used. r2 may be used by the real-time OS. If the real-time OS does not use r2, it can be used as a variable register.

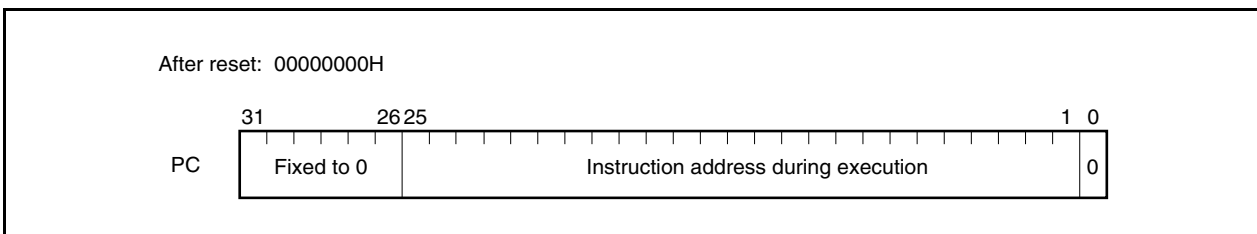
**Table 3-1. General-Purpose Registers**

| Name      | Usage  | Operation   |
|-----------|--|---|
| r0        | Zero register  | Always holds 0  |
| r1        | Assembler-reserved register  | Working register for generating 32-bit immediate data                           |
| r2        | Address/data variable register (when r2 is not used by the real-time OS) |   |
| r3        | Stack pointer  | Used to generate stack frame when function is called                            |
| r4        | Global pointer   | Used to access global variable in data area                                     |
| r5        | Text pointer   | Register to indicate the start of the text area (where program code is located) |
| r6 to r29 | Address/data variable registers  |   |
| r30       | Element pointer  | Base pointer when memory is accessed  |
| r31       | Link pointer   | Used by compiler when calling function  |

**(2) Program counter (PC)**

This register holds the instruction address during program execution. The lower 26 bits of this register are valid, and bits 31 to 26 are fixed to 0. If a carry occurs from bit 25 to 26, it is ignored.

Bit 0 is fixed to 0, and branching to an odd address cannot be performed.





**3.2.2 System register set**

System registers control the status of the CPU and hold interrupt information.

To read/write these system registers, specify a system register number indicated below using the system register load/store instruction (LDSR or STSR instruction).

**Table 3-2. System Register Numbers**

| System Register No. | System Register Name   | Operand Specification |                     |
|---------------------|--|-----------------------|---------------------|
|                     |  | LDSR Instruction      | STSR Instruction    |
| 0                   | Interrupt status saving register (EIPC) <sup>Note 1</sup>  | √                     | √                   |
| 1                   | Interrupt status saving register (EIPSW) <sup>Note 1</sup>   | √                     | √                   |
| 2                   | NMI status saving register (FEPC)  | √                     | √                   |
| 3                   | NMI status saving register (FEPSW)   | √                     | √                   |
| 4                   | Interrupt source register (ECR)  | ×                     | √                   |
| 5                   | Program status word (PSW)  | √                     | √                   |
| 6 to 15             | Reserved for future function expansion (operations that access these register numbers cannot be guaranteed). | ×                     | ×                   |
| 16                  | CALLT execution status saving register (CTPC)  | √                     | √                   |
| 17                  | CALLT execution status saving register (CTPSW)   | √                     | √                   |
| 18                  | Exception/debug trap status saving register (DBPC)   | √ <sup>Note 2</sup>   | √ <sup>Note 2</sup> |
| 19                  | Exception/debug trap status saving register (DBPSW)  | √ <sup>Note 2</sup>   | √ <sup>Note 2</sup> |
| 20                  | CALLT base pointer (CTBP)  | √                     | √                   |
| 21 to 31            | Reserved for future function expansion (operations that access these register numbers cannot be guaranteed). | ×                     | ×                   |

- Notes 1.** Because this register has only one set, to enable multiple interrupts, it is necessary to save this register by program.
- 2.** These registers can be read/written only in the period between DBTRAP instruction or illegal opcode execution and DBRET instruction execution.

**Caution** Even if bit 0 of EIPC, FEPC, or CTPC is set to 1 by the LDSR instruction, bit 0 will be ignored when the program is returned by the RETI instruction after interrupt servicing (because bit 0 of the PC is fixed to 0). When setting the value of EIPC, FEPC, and CTPC, use an even value (bit 0 = 0).

**Remark** √: Access allowed  
 ×: Access prohibited

**(1) Interrupt status saving registers (EIPC, EIPSW)**

There are two interrupt status saving registers, EIPC and EIPSW.

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

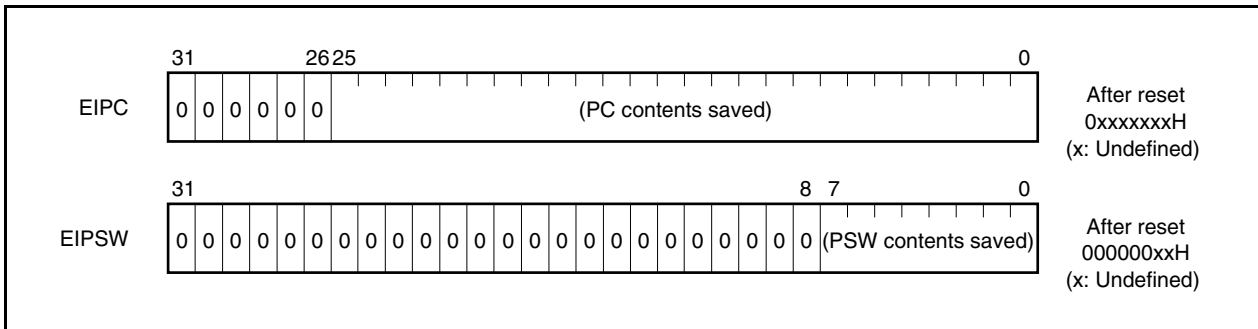
The address of the next instruction following the instruction executed when a software exception or maskable interrupt occurs is saved to EIPC, except for some instructions (see **20.9 Periods in Which CPU Does Not Acknowledge Interrupts**).

The current PSW contents are saved to EIPSW.

Since there is only one set of interrupt status saving registers, the contents of these registers must be saved by the program when multiple interrupt servicing is enabled.

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

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



**(2) NMI status saving registers (FEPC, FEPSW)**

There are two NMI status saving registers, FEPC and FEPSW.

Upon occurrence of a non-maskable interrupt (NMI), the contents of the program counter (PC) are saved to FEPC and the contents of the program status word (PSW) are saved to FEPSW.

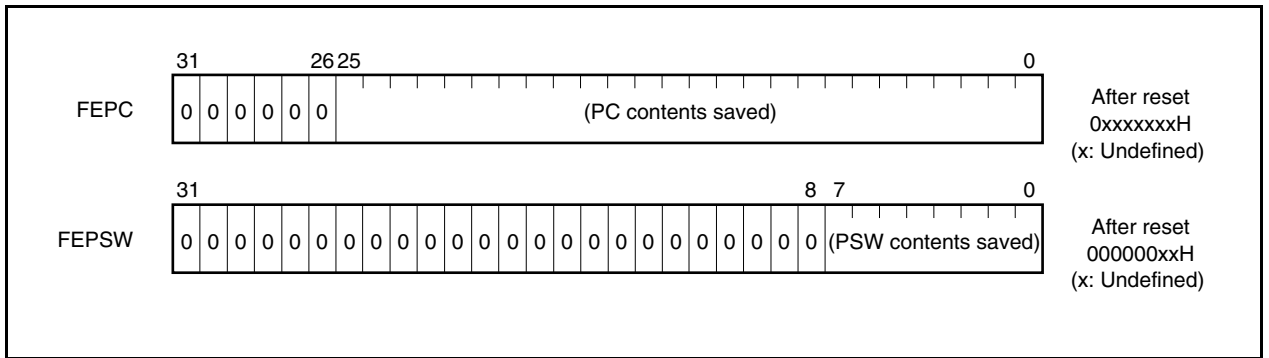
The address of the next instruction following the instruction executed when a non-maskable interrupt occurs is saved to FEPC, except for some instructions.

The current PSW contents are saved to FEPSW.

Since there is only one set of NMI status saving registers, the contents of these registers must be saved by the program when multiple interrupt servicing is enabled.

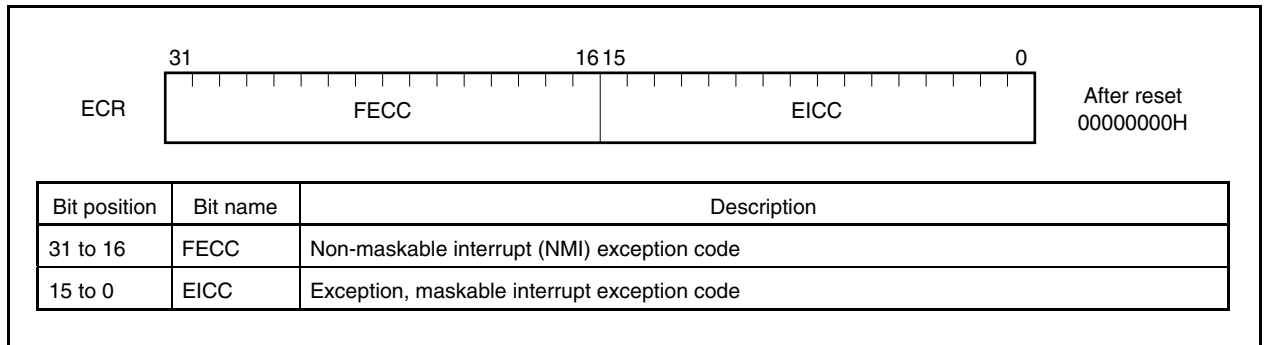
Bits 31 to 26 of FEPC and bits 31 to 8 of FEPSW are reserved (fixed to 0) for future function expansion.

When the RETI instruction has been executed, the values of FEPC and FEPSW are restored to the PC and PSW, respectively.



**(3) Interrupt source register (ECR)**

Upon occurrence of an interrupt or an exception, the interrupt source register (ECR) holds the source of an interrupt or an exception. The value held by ECR is the exception code coded for each interrupt source. This register is a read-only register, and thus data cannot be written to it using the LDSR instruction.



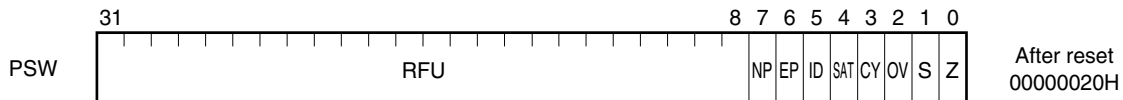
**(4) Program status word (PSW)**

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

When the contents of this register are changed using the LDSR instruction, the new contents become valid immediately following completion of LDSR instruction execution. Interrupt request acknowledgment is held pending while a write to the PSW is being executed by the LDSR instruction.

Bits 31 to 8 are reserved (fixed to 0) for future function expansion.

(1/2)



| Bit position | Flag name           | Description  |
|--------------|---------------------|--|
| 31 to 8      | RFU                 | Reserved field. Fixed to 0.  |
| 7            | NP                  | Indicates that non-maskable interrupt (NMI) servicing is in progress. This flag is set to 1 when an NMI request is acknowledged, and disables multiple interrupts.<br>0: NMI servicing not in progress<br>1: NMI servicing in progress   |
| 6            | EP                  | Indicates that exception processing is in progress. This flag is set to 1 when an exception occurs. Moreover, interrupt requests can be acknowledged even when this bit is set.<br>0: Exception processing not in progress<br>1: Exception processing in progress  |
| 5            | ID                  | Indicates whether maskable interrupt request acknowledgment is enabled.<br>0: Interrupt enabled (EI)<br>1: Interrupt disabled (DI)   |
| 4            | SAT <sup>Note</sup> | Indicates that the result of executing a saturated operation instruction has overflowed and that the calculation result is saturated. Since this is a cumulative flag, it is set to 1 when the result of a saturated operation instruction becomes saturated, and it is not cleared to 0 even if the operation results of successive instructions do not become saturated. This flag is neither set nor cleared when arithmetic operation instructions are executed.<br>0: Not saturated<br>1: Saturated |
| 3            | CY                  | Indicates whether carry or borrow occurred as the result of an operation.<br>0: No carry or borrow occurred<br>1: Carry or borrow occurred   |
| 2            | OV <sup>Note</sup>  | Indicates whether overflow occurred during an operation.<br>0: No overflow occurred<br>1: Overflow occurred.   |
| 1            | S <sup>Note</sup>   | Indicates whether the result of an operation is negative.<br>0: Operation result is positive or 0.<br>1: Operation result is negative.   |
| 0            | Z                   | Indicates whether operation result is 0.<br>0: Operation result is not 0.<br>1: Operation result is 0.   |

**Remark** Note is explained on the following page.

**Note** During saturated operation, the saturated operation results are determined by the contents of the OV flag and S flag. The SAT flag is set (to 1) only when the OV flag is set (to 1) during saturated operation.

| Operation result status               | Flag status                  |    |   | Saturated operation result |
|---------------------------------------|------------------------------|----|---|----------------------------|
|                                       | SAT                          | OV | S |                            |
| Maximum positive value exceeded       | 1                            | 1  | 0 | 7FFFFFFFH                  |
| Maximum negative value exceeded       | 1                            | 1  | 1 | 80000000H                  |
| Positive (maximum value not exceeded) | Holds value before operation | 0  | 0 | Actual operation result    |
| Negative (maximum value not exceeded) |                              |    | 1 |                            |

**(5) CALLT execution status saving registers (CTPC, CTPSW)**

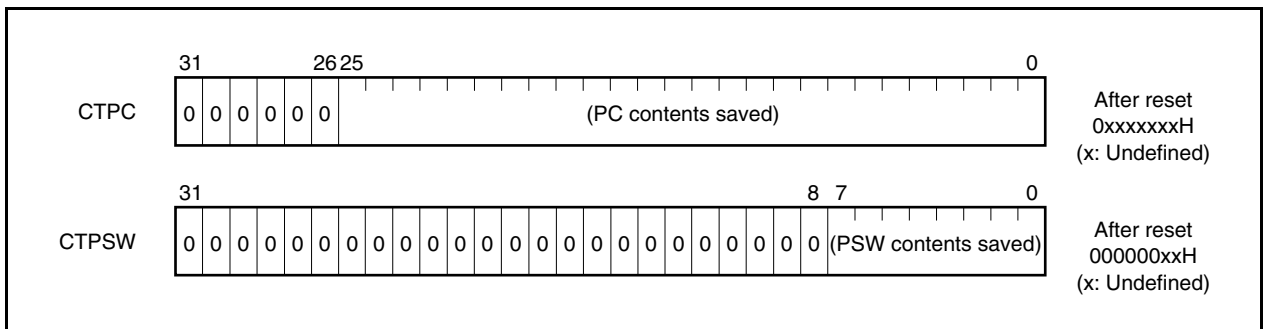
There are two CALLT execution status saving registers, CTPC and CTPSW.

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

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

The current PSW contents are saved to CTPSW.

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



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

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

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

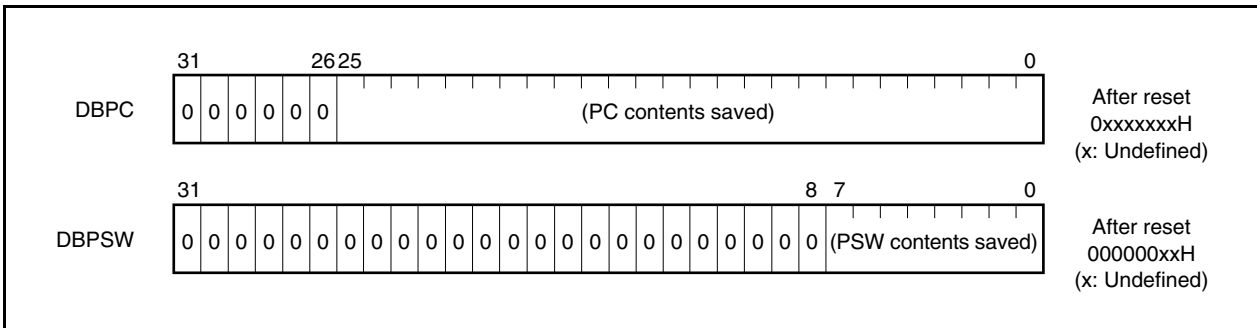
The contents saved to DBPC consist of the address of the next instruction after the instruction executed when an exception trap or debug trap occurs.

The current PSW contents are saved to DBPSW.

These registers can be read or written only in the period between DBTRAP instruction or illegal opcode execution and DBRET instruction execution.

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

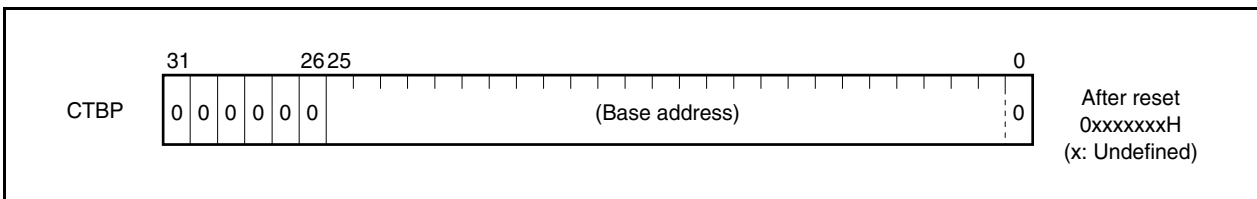
When the DBRET instruction has been executed, the values of DBPC and DBPSW are restored to the PC and PSW, respectively.



**(7) CALLT base pointer (CTBP)**

The CALLT base pointer (CTBP) is used to specify table addresses and generate target addresses (bit 0 is fixed to 0).

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



### 3.3 Operating Modes

#### 3.3.1 Operating modes

The V850E/IF3 and V850E/IG3 have the following operating modes. Mode specification is carried out using the FLMD0 and FLMD1 pins.

##### (1) Normal operation mode

In this mode, execution branches to the reset entry address in the internal ROM and instruction processing is started when system reset is released.

##### (2) Flash memory programming mode

If this mode is specified, a program can be written to the internal flash memory by the flash memory programmer.

#### 3.3.2 Operating mode specification

The operating mode is specified according to the status (input level) of the FLMD0 and FLMD1 pins.

In the normal operating mode, input a low level to the FLMD0 pin after reset.

When the flash memory programmer is connected, a high level is input to the FLMD0 pin by the flash memory programmer in the flash memory programming mode; however, in the self-programming mode, input a high level via an external circuit.

Other than in the self-programming mode, fix the specifications of these pins in the application system, and do not change them during operation.

| FLMD1 | FLMD0 | Operating Mode                | Remarks  |
|-------|-------|-------------------------------|--|
| ×     | L     | Normal operation mode         | Internal ROM area is allocated from address 000000H. |
| L     | H     | Flash memory programming mode | –  |
| H     | H     | Setting prohibited            |  |

**Remark** L: Low-level input  
H: High-level input

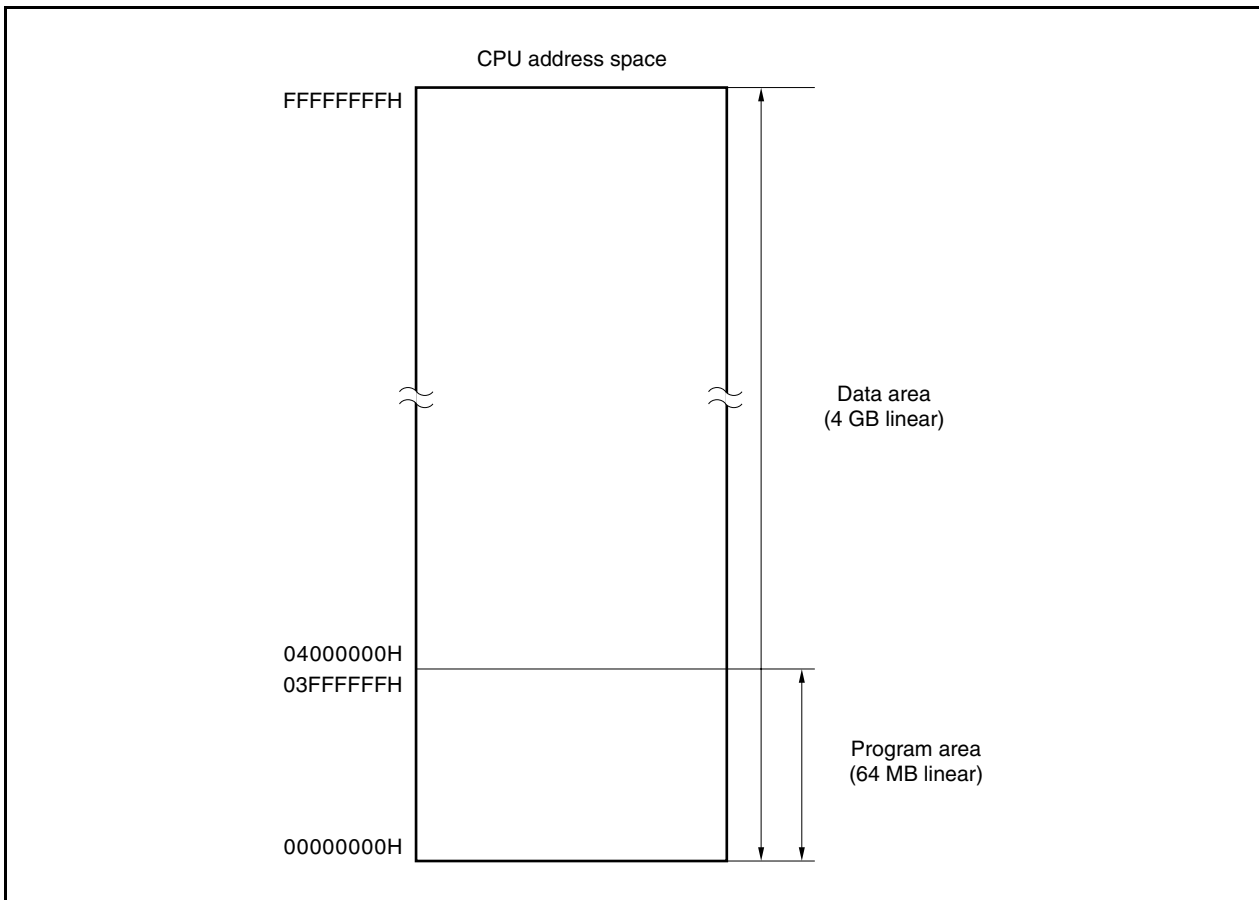
### 3.4 Address Space

#### 3.4.1 CPU address space

The CPU of the V850E/IF3 and V850E/IG3 has 32-bit architecture and supports up to 4 GB of linear address space (data space) during operand addressing (data access). Also, in instruction address addressing, a maximum of 64 MB of linear address space (program space) is supported.

Figure 3-2 shows the CPU address space.

Figure 3-2. CPU Address Space



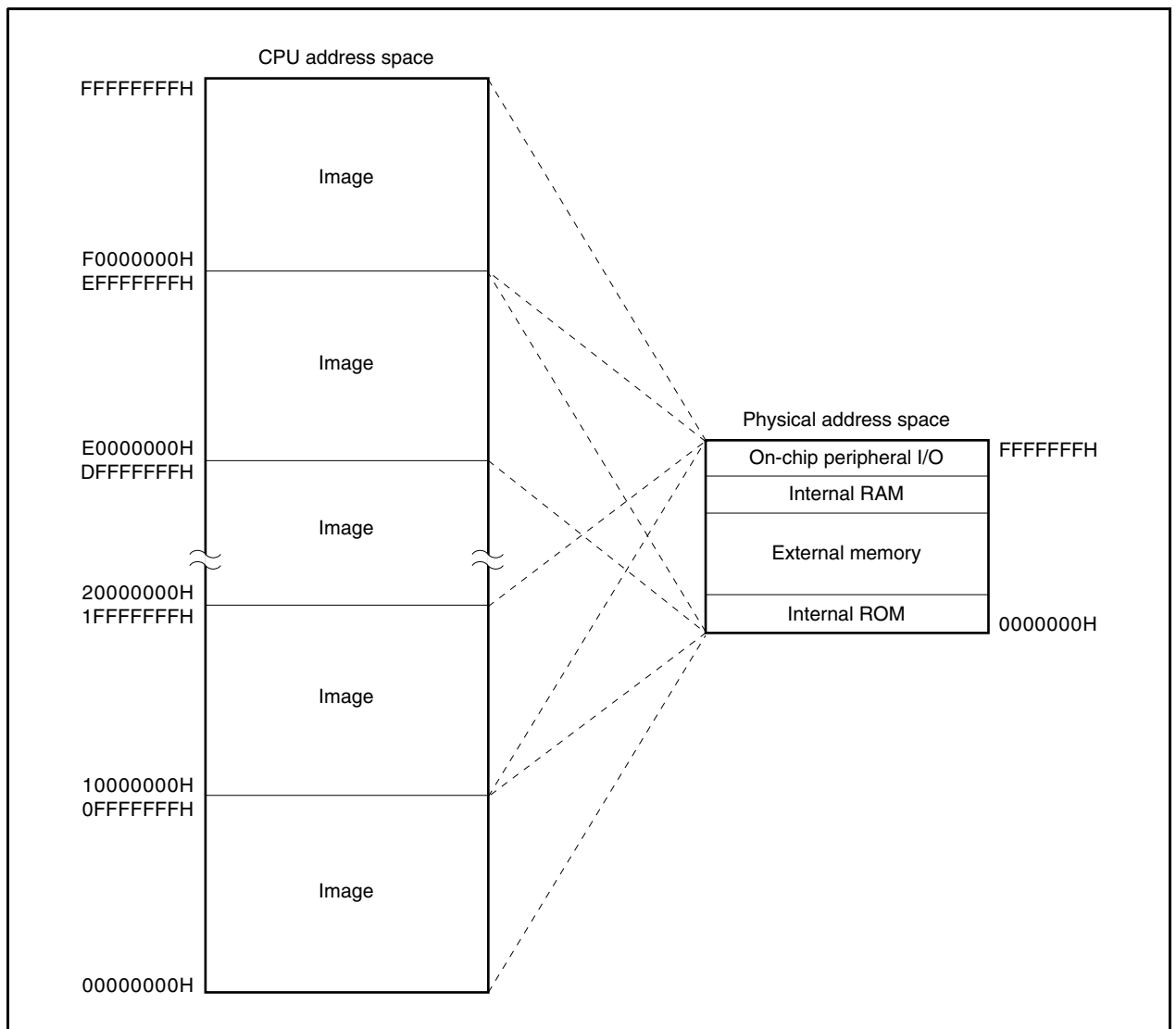


3.4.2 Image

A 256 MB physical address space is seen as 16 images in the 4 GB CPU address space. In actuality, the same 256 MB physical address space is accessed regardless of the values of bits 31 to 28 of the CPU address. Figure 3-3 shows the image of the virtual addressing space.

Physical address x0000000H can be seen as CPU address 00000000H, and in addition, can be seen as address 10000000H, address 20000000H, ... , address E0000000H, or address F0000000H.

Figure 3-3. Images on Address Space



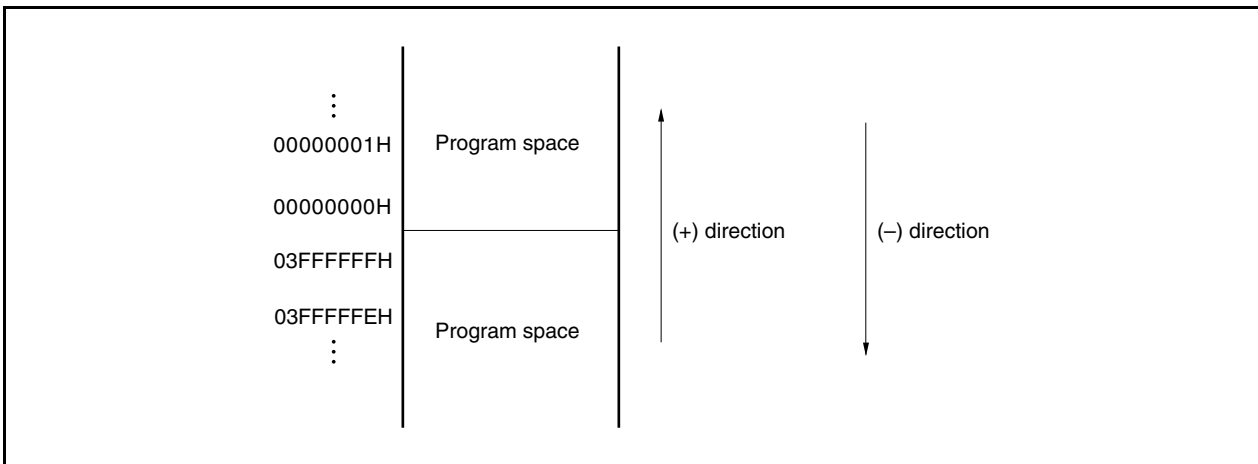
3.4.3 Wraparound of CPU address space

(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. Even if a carry or borrow occurs from bit 25 to 26 as a result of a branch address calculation, the higher 6 bits ignore the carry or borrow.

Therefore, the upper-limit address of the program space, address 03FFFFFFH, and the lower-limit address 00000000H become contiguous addresses. Wraparound refers to a situation like this whereby the lower-limit address and upper-limit address become contiguous.

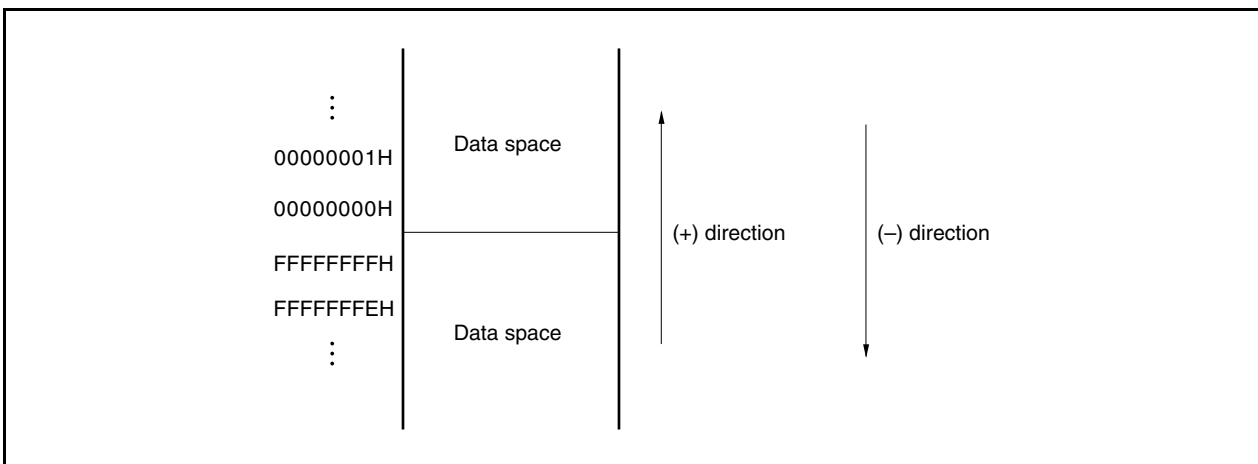
**Caution** The 4 KB area of 03FFF000H to 03FFFFFFH can be seen as an image of 0FFFF000H to 0FFFFFFFH. This area is access-prohibited. Therefore, do not execute any branch address calculation in which the result will reside in any part of this area.



(2) Data space

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

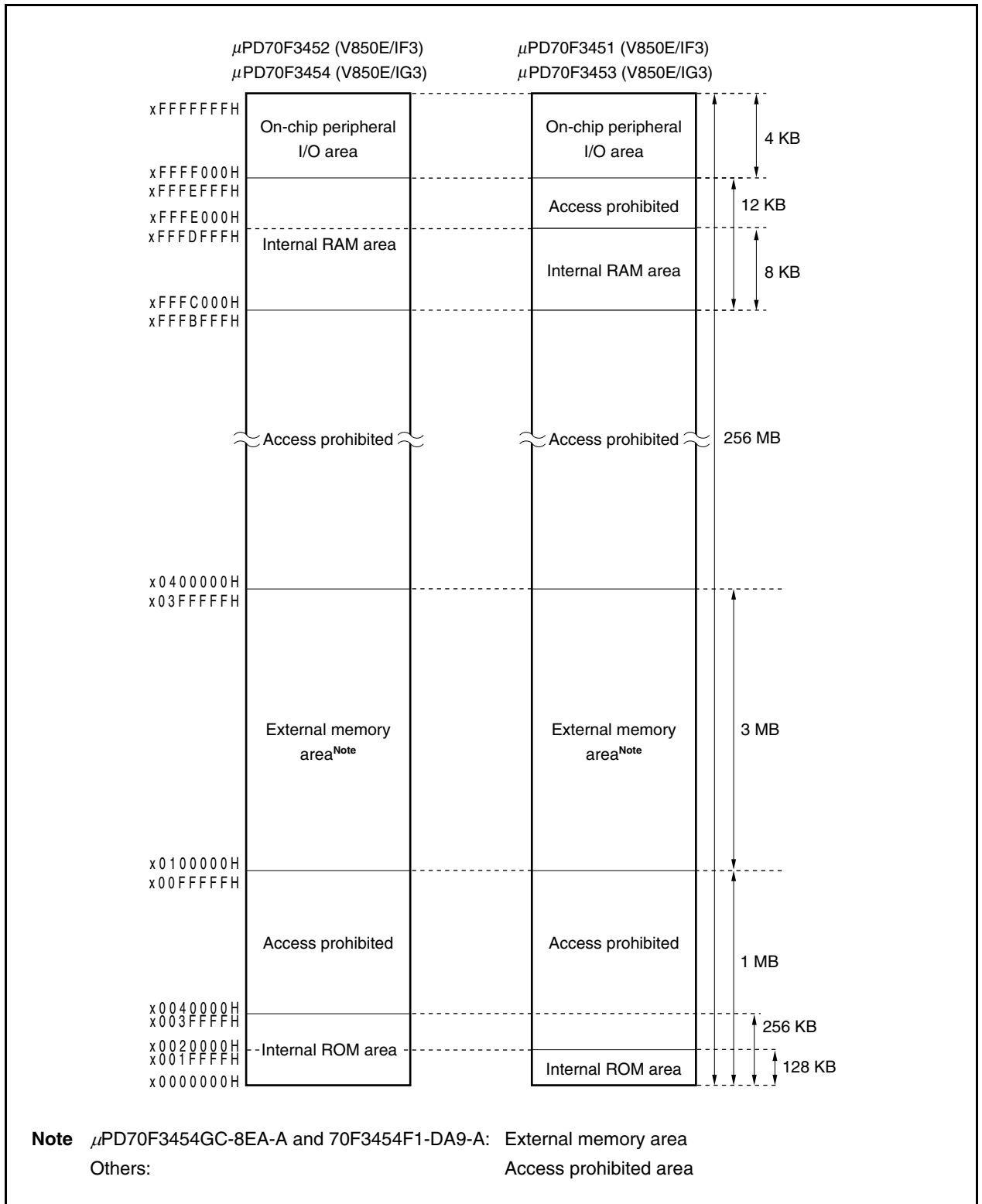
Therefore, the upper-limit address of the program space, address FFFFFFFFH, and the lower-limit address 00000000H are contiguous addresses, and the data space is wrapped around at the boundary of these addresses.



3.4.4 Memory map

The V850E/IF3 and V850E/IG3 reserve areas as shown in Figure 3-4.

Figure 3-4. Memory Map



3.4.5 Area

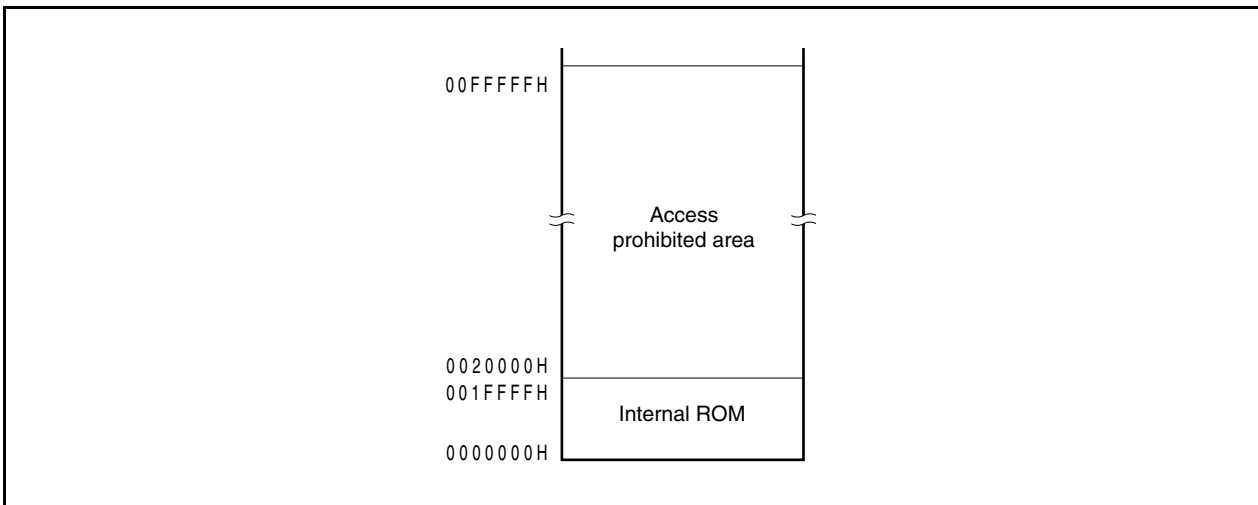
(1) Internal ROM area

1 MB of internal ROM area, addresses 00000H to FFFFFH, is reserved.

(a)  $\mu$ PD70F3451 (V850E/IF3),  $\mu$ PD70F3453 (V850E/IG3)

128 KB are provided at addresses 000000H to 01FFFFH as physical internal ROM.

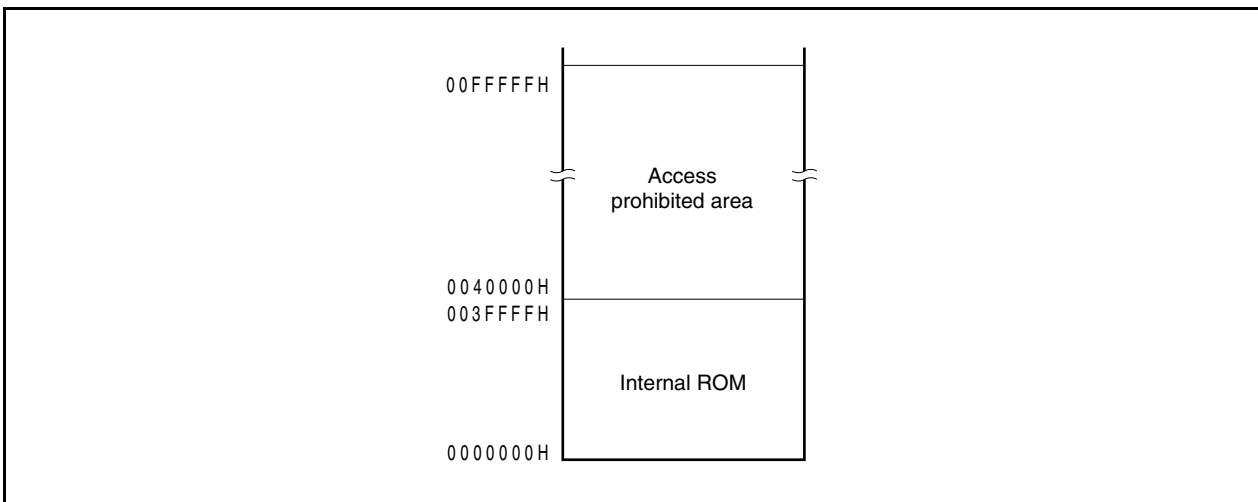
Figure 3-5. Internal ROM Area (128 KB)



(b)  $\mu$ PD70F3452 (V850E/IF3),  $\mu$ PD70F3454 (V850E/IG3)

256 KB are provided at addresses 000000H to 03FFFFH as physical internal ROM.

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



**(2) Internal RAM area**

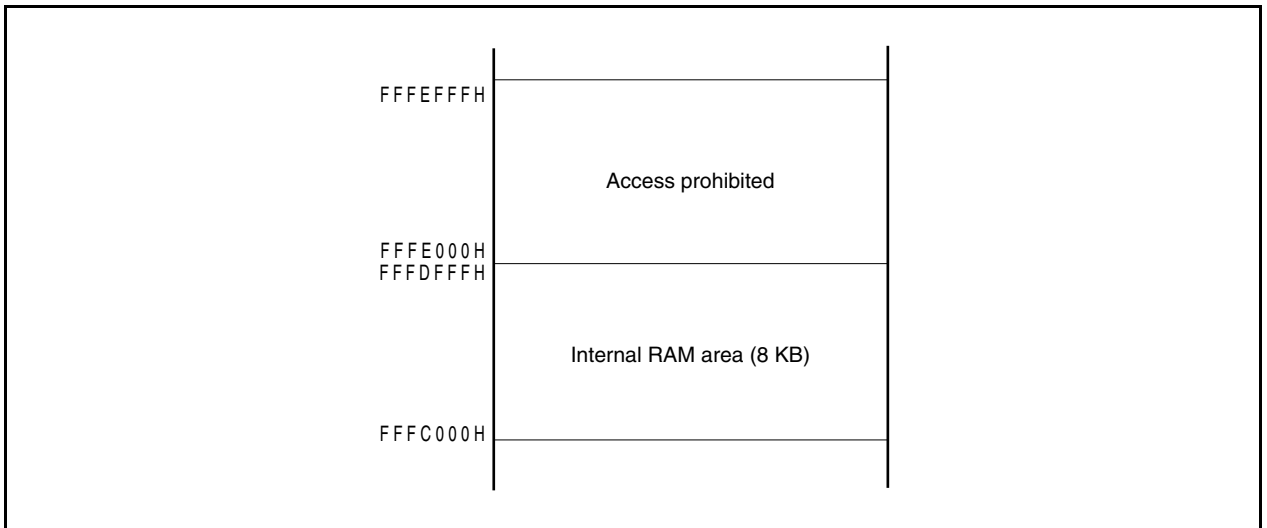
The 12 KB area of addresses FFFC000H to FFFEFFFFH is reserved for the internal RAM area.

**(a)  $\mu$ PD70F3451 (V850E/IF3),  $\mu$ PD70F3453 (V850E/IG3)**

The 8 KB area of addresses FFFC000H to FFFDFFFFH is provided as physical internal RAM.

**Caution** The following areas are access-prohibited.  
**Addresses FFFE000H to FFFEFFFFH**

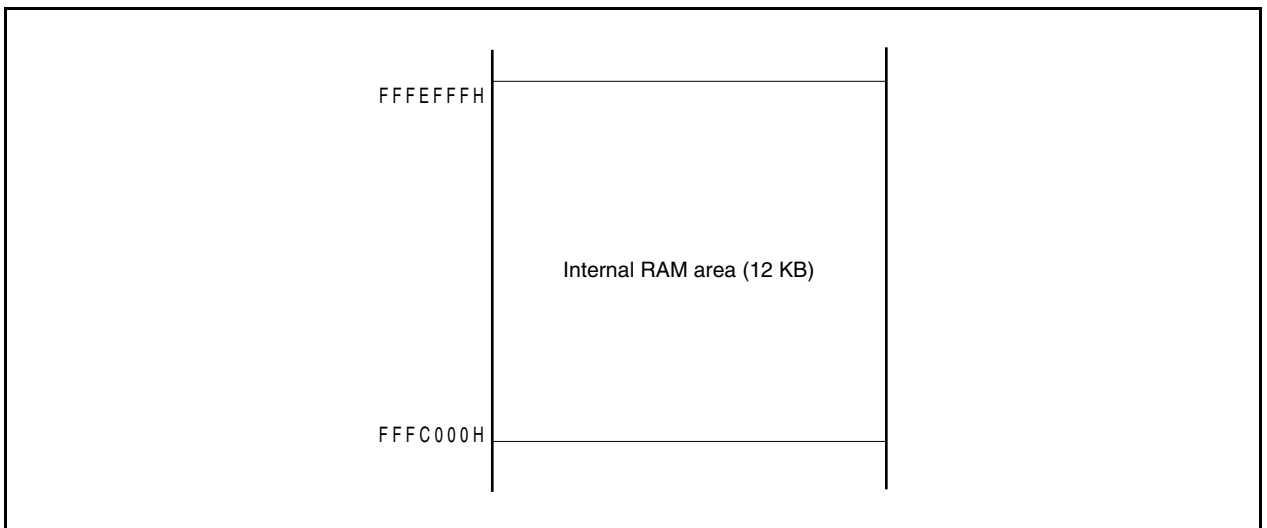
**Figure 3-7. Internal RAM Area (8 KB)**



**(b)  $\mu$ PD70F3452 (V850E/IF3),  $\mu$ PD70F3454 (V850E/IG3)**

The 12 KB area of addresses FFFC000H to FFFEFFFFH is provided as physical internal RAM.

**Figure 3-8. Internal RAM Area (12 KB)**

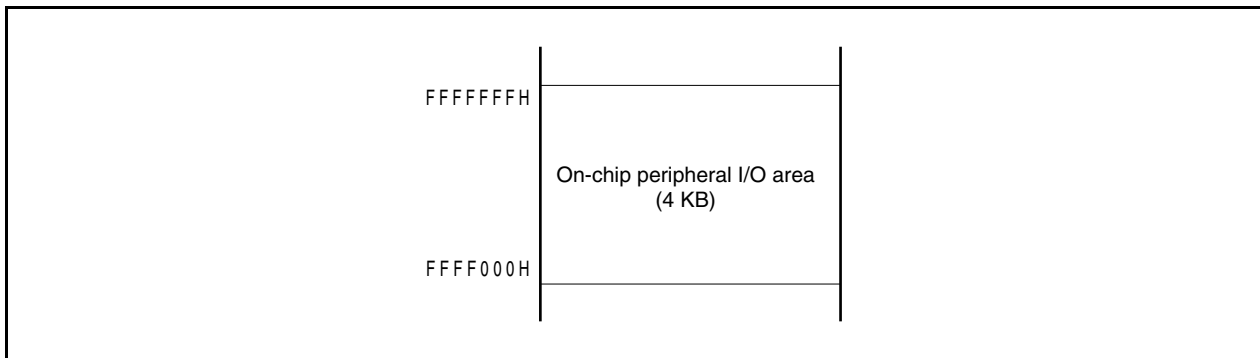


**(3) On-chip peripheral I/O area**

4 KB of memory, addresses FFFF000H to FFFFFFFFH, is provided as an on-chip peripheral I/O area. An image of addresses FFFF000H to FFFFFFFFH can be seen at addresses 3FFF000H to 3FFFFFFFH<sup>Note</sup>.

**Note** Addresses 3FFF000H to 3FFFFFFFH are access-prohibited. To access the on-chip peripheral I/O, specify addresses FFFF000H to FFFFFFFFH.

**Figure 3-9. On-Chip Peripheral I/O Area**



On-chip peripheral I/O registers associated with the operating mode specification and the state monitoring for the on-chip peripheral I/O are all memory-mapped to the on-chip peripheral I/O area. Program fetches cannot be executed from this area.

- Cautions**
1. In the V850E/IF3 and V850E/IG3, if a register is word accessed, halfword access is performed twice in the order of lower address, then higher address of the word area, disregarding the lower 2 bits of the address.
  2. For registers in which byte access is possible, if halfword access is executed, the higher 8 bits become undefined during the read operation, and the lower 8 bits of data are written to the register during the write operation.
  3. Addresses that are not defined as registers are reserved for future expansion. If these addresses are accessed, the operation is undefined and not guaranteed.  
Addresses 3FFF000H to 3FFFFFFFH cannot be specified as the source/destination address of DMA transfer. Be sure to use addresses FFFF000H to FFFFFFFFH for the source/destination address of DMA transfer.

**(4) External memory area ( $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only)**

3 MB (0100000H to 03FFFFFFH) are available for the external memory area. For details, see **CHAPTER 18 BUS CONTROL FUNCTION**.

**3.4.6 Recommended use of address space**

The architecture of the V850E/IF3 and V850E/IG3 requires that a register that serves as a pointer be secured for address generation in operand data accessing of data space. Operand data access from instruction can be directly executed at the address in this pointer register area  $\pm 32$  KB. However, because the general-purpose registers that can be used as a pointer register are limited, by minimizing the deterioration of address calculation performance when changing the pointer value, the number of usable general-purpose registers for handling variables is maximized, and the program size can be saved.

**(1) Program space**

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

**(2) Data space**

With the V850E/IF3 and V850E/IG3, a 256 MB physical address space is seen as 16 images in the 4 GB CPU address space. The highest bit (bit 25) of this 26-bit address is assigned as an address sign-extended to 32 bits.

**(a) Application examples using wraparound**

When R = r0 (zero register) is specified by the LD/ST disp16 [R] instruction, an addressing range of 00000000H  $\pm 32$  KB can be referenced by the sign-extended disp16.

The zero register (r0) is a register set to 0 by the hardware, and eliminates the need for additional registers for the pointer.

**Example  $\mu$ PD70F3454 (V850E/IG3)**

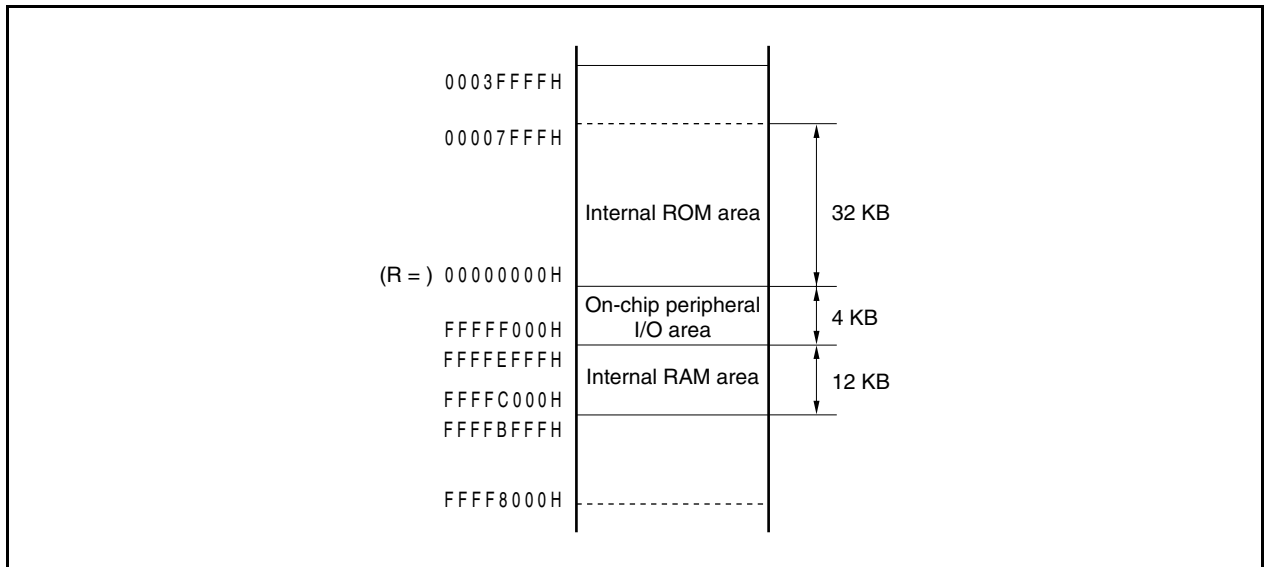
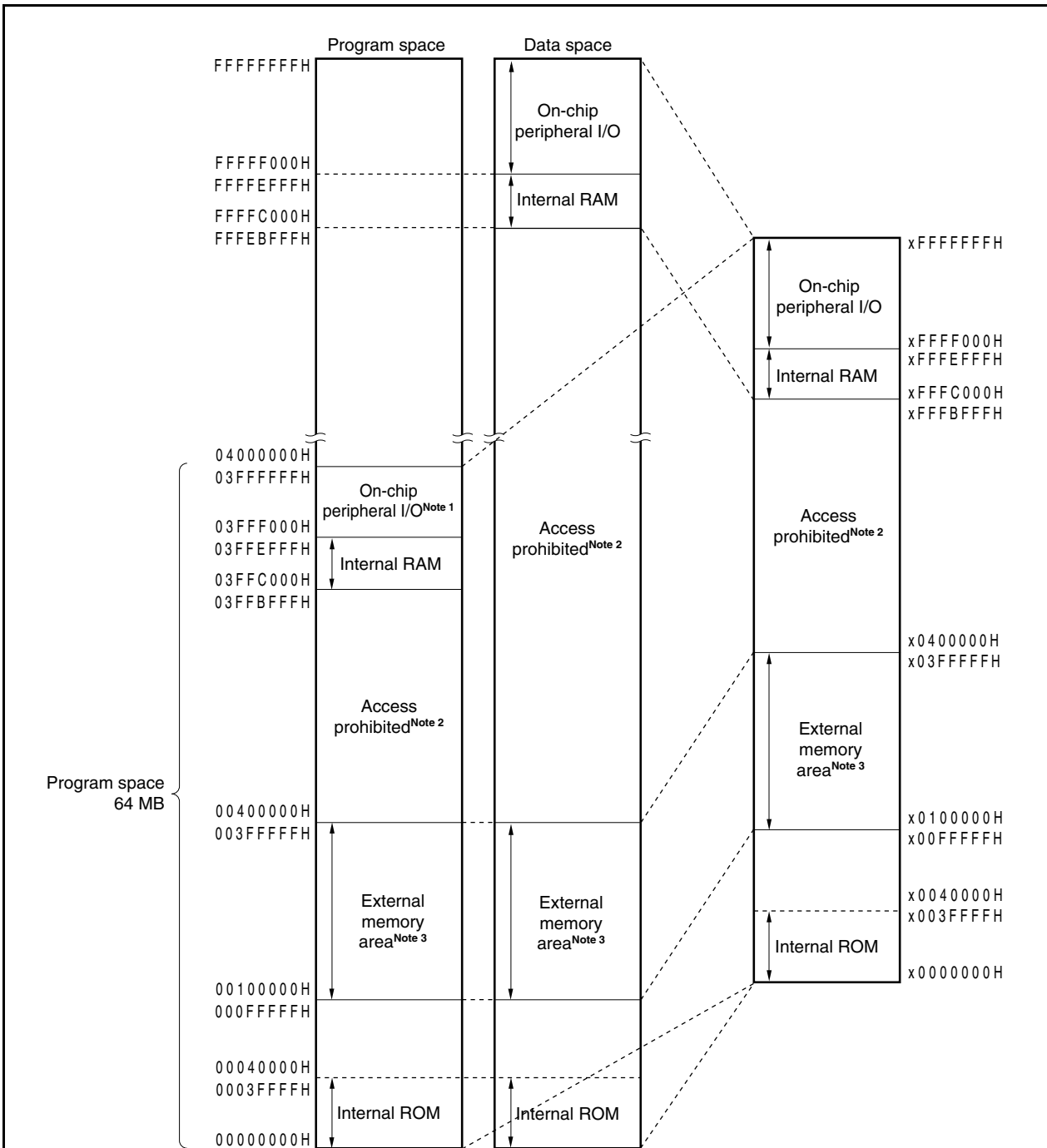


Figure 3-10. Recommended Memory Map



- Notes**
1. This area is access-prohibited. To access the on-chip peripheral I/O, specify addresses FFFF000H to FFFFFFFH.
  2. The operation is not guaranteed if an access-prohibited area is accessed.
  3.  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A: External memory area  
Others: Access prohibited area<sup>Note 2</sup>

- Remarks**
1. The arrows indicate the recommended area.
  2. This is a recommended memory map for the  $\mu$ PD70F3454 (V850E/IG3).



3.4.7 On-chip peripheral I/O registers

(1/14)

| Address   | Function Register Name              | Symbol | R/W | Bit Units for Manipulation |   |    | After Reset |
|-----------|-------------------------------------|--------|-----|----------------------------|---|----|-------------|
|           |                                     |        |     | 1                          | 8 | 16 |             |
| FFFFF004H | Port DL register                    | PDL    | R/W |                            |   | √  | Undefined   |
| FFFFF004H | Port DLL register                   | PDLL   |     | √                          | √ |    | Undefined   |
| FFFFF005H | Port DLH register                   | PDLH   |     | √                          | √ |    | Undefined   |
| FFFFF024H | Port DL mode register               | PMDL   |     |                            |   | √  | FFFFH       |
| FFFFF024H | Port DL mode register L             | PMDLL  |     | √                          | √ |    | FFH         |
| FFFFF025H | Port DL mode register H             | PMDLH  |     | √                          | √ |    | FFH         |
| FFFFF044H | Port DL mode control register       | PMCDL  |     |                            |   | √  | 0000H       |
| FFFFF044H | Port DL mode control register L     | PMCDLL |     | √                          | √ |    | 00H         |
| FFFFF045H | Port DL mode control register H     | PMCDLH |     | √                          | √ |    | 00H         |
| FFFFF066H | Bus size configuration register     | BSC    |     |                            |   | √  | 5555H       |
| FFFFF06EH | System wait control register        | VSWC   |     |                            | √ |    | 77H         |
| FFFFF080H | DMA source address register 0L      | DSA0L  |     |                            |   | √  | Undefined   |
| FFFFF082H | DMA source address register 0H      | DSA0H  |     |                            |   | √  | Undefined   |
| FFFFF084H | DMA destination address register 0L | DDA0L  |     |                            |   | √  | Undefined   |
| FFFFF086H | DMA destination address register 0H | DDA0H  |     |                            |   | √  | Undefined   |
| FFFFF088H | DMA source address register 1L      | DSA1L  |     |                            |   | √  | Undefined   |
| FFFFF08AH | DMA source address register 1H      | DSA1H  |     |                            |   | √  | Undefined   |
| FFFFF08CH | DMA destination address register 1L | DDA1L  |     |                            |   | √  | Undefined   |
| FFFFF08EH | DMA destination address register 1H | DDA1H  |     |                            |   | √  | Undefined   |
| FFFFF090H | DMA source address register 2L      | DSA2L  |     |                            |   | √  | Undefined   |
| FFFFF092H | DMA source address register 2H      | DSA2H  |     |                            |   | √  | Undefined   |
| FFFFF094H | DMA destination address register 2L | DDA2L  |     |                            |   | √  | Undefined   |
| FFFFF096H | DMA destination address register 2H | DDA2H  |     |                            |   | √  | Undefined   |
| FFFFF098H | DMA source address register 3L      | DSA3L  |     |                            |   | √  | Undefined   |
| FFFFF09AH | DMA source address register 3H      | DSA3H  |     |                            |   | √  | Undefined   |
| FFFFF09CH | DMA destination address register 3L | DDA3L  |     |                            |   | √  | Undefined   |
| FFFFF09EH | DMA destination address register 3H | DDA3H  |     |                            |   | √  | Undefined   |
| FFFFF0C0H | DMA transfer count register 0       | DBC0   |     |                            |   | √  | Undefined   |
| FFFFF0C2H | DMA transfer count register 1       | DBC1   |     |                            |   | √  | Undefined   |
| FFFFF0C4H | DMA transfer count register 2       | DBC2   |     |                            |   | √  | Undefined   |
| FFFFF0C6H | DMA transfer count register 3       | DBC3   |     |                            |   | √  | Undefined   |
| FFFFF0D0H | DMA addressing control register 0   | DADC0  |     |                            |   | √  | 0000H       |
| FFFFF0D2H | DMA addressing control register 1   | DADC1  |     |                            |   | √  | 0000H       |
| FFFFF0D4H | DMA addressing control register 2   | DADC2  |     |                            |   | √  | 0000H       |
| FFFFF0D6H | DMA addressing control register 3   | DADC3  |     |                            |   | √  | 0000H       |
| FFFFF0E0H | DMA channel control register 0      | DCHC0  |     | √                          | √ |    | 00H         |
| FFFFF0E2H | DMA channel control register 1      | DCHC1  |     | √                          | √ |    | 00H         |
| FFFFF0E4H | DMA channel control register 2      | DCHC2  |     | √                          | √ |    | 00H         |
| FFFFF0E6H | DMA channel control register 3      | DCHC3  |     | √                          | √ |    | 00H         |

| Address   | Function Register Name     | Symbol                | R/W | Bit Units for Manipulation |   |     | After Reset |
|-----------|----------------------------|-----------------------|-----|----------------------------|---|-----|-------------|
|           |                            |                       |     | 1                          | 8 | 16  |             |
| FFFFF100H | Interrupt mask register 0  | IMR0                  | R/W |                            |   | √   | FFFFH       |
| FFFFF100H | Interrupt mask register 0L | IMR0L                 |     | √                          | √ |     | FFH         |
| FFFFF101H | Interrupt mask register 0H | IMR0H                 |     | √                          | √ |     | FFH         |
| FFFFF102H | Interrupt mask register 1  | IMR1                  |     |                            |   | √   | FFFFH       |
| FFFFF102H | Interrupt mask register 1L | IMR1L                 |     | √                          | √ |     | FFH         |
| FFFFF103H | Interrupt mask register 1H | IMR1H                 |     | √                          | √ |     | FFH         |
| FFFFF104H | Interrupt mask register 2  | IMR2                  |     |                            |   | √   | FFFFH       |
| FFFFF104H | Interrupt mask register 2L | IMR2L                 |     | √                          | √ |     | FFH         |
| FFFFF105H | Interrupt mask register 2H | IMR2H                 |     | √                          | √ |     | FFH         |
| FFFFF106H | Interrupt mask register 3  | IMR3                  |     |                            |   | √   | FFFFH       |
| FFFFF106H | Interrupt mask register 3L | IMR3L                 |     | √                          | √ |     | FFH         |
| FFFFF107H | Interrupt mask register 3H | IMR3H                 |     | √                          | √ |     | FFH         |
| FFFFF108H | Interrupt mask register 4  | IMR4                  |     |                            |   | √   | FFFFH       |
| FFFFF108H | Interrupt mask register 4L | IMR4L                 |     | √                          | √ |     | FFH         |
| FFFFF109H | Interrupt mask register 4H | IMR4H                 |     | √                          | √ |     | FFH         |
| FFFFF10AH | Interrupt mask register 5  | IMR5                  |     |                            |   | √   | FFFFH       |
| FFFFF10AH | Interrupt mask register 5L | IMR5L                 |     | √                          | √ |     | FFH         |
| FFFFF10BH | Interrupt mask register 5H | IMR5H                 |     | √                          | √ |     | FFH         |
| FFFFF110H | Interrupt control register | LVILIC                |     | √                          | √ |     | 47H         |
| FFFFF112H | Interrupt control register | LVIHIC                |     | √                          | √ |     | 47H         |
| FFFFF114H | Interrupt control register | PIC00                 |     | √                          | √ |     | 47H         |
| FFFFF116H | Interrupt control register | PIC01                 |     | √                          | √ |     | 47H         |
| FFFFF118H | Interrupt control register | PIC02 <sup>Note</sup> |     | √                          | √ |     | 47H         |
| FFFFF11AH | Interrupt control register | PIC03 <sup>Note</sup> |     | √                          | √ |     | 47H         |
| FFFFF11CH | Interrupt control register | PIC04 <sup>Note</sup> |     | √                          | √ |     | 47H         |
| FFFFF11EH | Interrupt control register | PIC05 <sup>Note</sup> |     | √                          | √ |     | 47H         |
| FFFFF120H | Interrupt control register | PIC06 <sup>Note</sup> |     | √                          | √ |     | 47H         |
| FFFFF122H | Interrupt control register | PIC07 <sup>Note</sup> |     | √                          | √ |     | 47H         |
| FFFFF124H | Interrupt control register | PIC08                 |     | √                          | √ |     | 47H         |
| FFFFF126H | Interrupt control register | PIC09                 |     | √                          | √ |     | 47H         |
| FFFFF128H | Interrupt control register | PIC10                 |     | √                          | √ |     | 47H         |
| FFFFF12AH | Interrupt control register | PIC11                 |     | √                          | √ |     | 47H         |
| FFFFF12CH | Interrupt control register | PIC12                 |     | √                          | √ |     | 47H         |
| FFFFF12EH | Interrupt control register | PIC13                 | √   | √                          |   | 47H |             |
| FFFFF130H | Interrupt control register | PIC14                 | √   | √                          |   | 47H |             |
| FFFFF132H | Interrupt control register | PIC15                 | √   | √                          |   | 47H |             |
| FFFFF134H | Interrupt control register | PIC16                 | √   | √                          |   | 47H |             |
| FFFFF136H | Interrupt control register | PIC17                 | √   | √                          |   | 47H |             |
| FFFFF138H | Interrupt control register | PIC18                 | √   | √                          |   | 47H |             |
| FFFFF13AH | Interrupt control register | CMPIC0L               | √   | √                          |   | 47H |             |

**Note** V850E/IG3 only

| Address   | Function Register Name     | Symbol                   | R/W | Bit Units for Manipulation |   |     | After Reset |
|-----------|----------------------------|--------------------------|-----|----------------------------|---|-----|-------------|
|           |                            |                          |     | 1                          | 8 | 16  |             |
| FFFFF13CH | Interrupt control register | CMPIC0F                  | R/W | √                          | √ |     | 47H         |
| FFFFF13EH | Interrupt control register | CMPIC1L                  |     | √                          | √ |     | 47H         |
| FFFFF140H | Interrupt control register | CMPIC1F                  |     | √                          | √ |     | 47H         |
| FFFFF142H | Interrupt control register | TB0OVIC                  |     | √                          | √ |     | 47H         |
| FFFFF144H | Interrupt control register | TB0CCIC0                 |     | √                          | √ |     | 47H         |
| FFFFF146H | Interrupt control register | TB0CCIC1                 |     | √                          | √ |     | 47H         |
| FFFFF148H | Interrupt control register | TB0CCIC2                 |     | √                          | √ |     | 47H         |
| FFFFF14AH | Interrupt control register | TB0CCIC3                 |     | √                          | √ |     | 47H         |
| FFFFF14CH | Interrupt control register | TB1OVIC                  |     | √                          | √ |     | 47H         |
| FFFFF14EH | Interrupt control register | TB1CCIC0                 |     | √                          | √ |     | 47H         |
| FFFFF150H | Interrupt control register | TB1CCIC1                 |     | √                          | √ |     | 47H         |
| FFFFF152H | Interrupt control register | TB1CCIC2                 |     | √                          | √ |     | 47H         |
| FFFFF154H | Interrupt control register | TB1CCIC3                 |     | √                          | √ |     | 47H         |
| FFFFF156H | Interrupt control register | TT0OVIC                  |     | √                          | √ |     | 47H         |
| FFFFF158H | Interrupt control register | TT0CCIC0                 |     | √                          | √ |     | 47H         |
| FFFFF15AH | Interrupt control register | TT0CCIC1                 |     | √                          | √ |     | 47H         |
| FFFFF15CH | Interrupt control register | TT0IECIC <sup>Note</sup> |     | √                          | √ |     | 47H         |
| FFFFF15EH | Interrupt control register | TT1OVIC                  |     | √                          | √ |     | 47H         |
| FFFFF160H | Interrupt control register | TT1CCIC0                 |     | √                          | √ |     | 47H         |
| FFFFF162H | Interrupt control register | TT1CCIC1                 |     | √                          | √ |     | 47H         |
| FFFFF164H | Interrupt control register | TT1IECIC                 |     | √                          | √ |     | 47H         |
| FFFFF166H | Interrupt control register | TA0OVIC                  |     | √                          | √ |     | 47H         |
| FFFFF168H | Interrupt control register | TA0CCIC0                 |     | √                          | √ |     | 47H         |
| FFFFF16AH | Interrupt control register | TA0CCIC1                 |     | √                          | √ |     | 47H         |
| FFFFF16CH | Interrupt control register | TA1OVIC                  |     | √                          | √ |     | 47H         |
| FFFFF16EH | Interrupt control register | TA1CCIC0                 |     | √                          | √ |     | 47H         |
| FFFFF170H | Interrupt control register | TA1CCIC1                 |     | √                          | √ |     | 47H         |
| FFFFF172H | Interrupt control register | TA2OVIC                  |     | √                          | √ |     | 47H         |
| FFFFF174H | Interrupt control register | TA2CCIC0                 |     | √                          | √ |     | 47H         |
| FFFFF176H | Interrupt control register | TA2CCIC1                 |     | √                          | √ |     | 47H         |
| FFFFF178H | Interrupt control register | TA3OVIC                  |     | √                          | √ |     | 47H         |
| FFFFF17AH | Interrupt control register | TA3CCIC0                 |     | √                          | √ |     | 47H         |
| FFFFF17CH | Interrupt control register | TA3CCIC1                 | √   | √                          |   | 47H |             |
| FFFFF17EH | Interrupt control register | TA4OVIC                  | √   | √                          |   | 47H |             |
| FFFFF180H | Interrupt control register | TA4CCIC0                 | √   | √                          |   | 47H |             |
| FFFFF182H | Interrupt control register | TA4CCIC1                 | √   | √                          |   | 47H |             |
| FFFFF184H | Interrupt control register | DMAIC0                   | √   | √                          |   | 47H |             |
| FFFFF186H | Interrupt control register | DMAIC1                   | √   | √                          |   | 47H |             |
| FFFFF188H | Interrupt control register | DMAIC2                   | √   | √                          |   | 47H |             |
| FFFFF18AH | Interrupt control register | DMAIC3                   | √   | √                          |   | 47H |             |

**Note** V850E/IG3 only

| Address   | Function Register Name             | Symbol   | R/W | Bit Units for Manipulation |   |     | After Reset |       |
|-----------|------------------------------------|----------|-----|----------------------------|---|-----|-------------|-------|
|           |                                    |          |     | 1                          | 8 | 16  |             |       |
| FFFFF18CH | Interrupt control register         | UREIC    | R/W | √                          | √ |     | 47H         |       |
| FFFFF18EH | Interrupt control register         | URIC     |     | √                          | √ |     | 47H         |       |
| FFFFF190H | Interrupt control register         | UTIC     |     | √                          | √ |     | 47H         |       |
| FFFFF192H | Interrupt control register         | UIFIC    |     | √                          | √ |     | 47H         |       |
| FFFFF194H | Interrupt control register         | UTOIC    |     | √                          | √ |     | 47H         |       |
| FFFFF196H | Interrupt control register         | UA0REIC  |     | √                          | √ |     | 47H         |       |
| FFFFF198H | Interrupt control register         | UA0RIC   |     | √                          | √ |     | 47H         |       |
| FFFFF19AH | Interrupt control register         | UA0TIC   |     | √                          | √ |     | 47H         |       |
| FFFFF19CH | Interrupt control register         | CB0REIC  |     | √                          | √ |     | 47H         |       |
| FFFFF19EH | Interrupt control register         | CB0RIC   |     | √                          | √ |     | 47H         |       |
| FFFFF1A0H | Interrupt control register         | CB0TIC   |     | √                          | √ |     | 47H         |       |
| FFFFF1A2H | Interrupt control register         | UA1REIC  |     | √                          | √ |     | 47H         |       |
| FFFFF1A4H | Interrupt control register         | UA1RIC   |     | √                          | √ |     | 47H         |       |
| FFFFF1A6H | Interrupt control register         | UA1TIC   |     | √                          | √ |     | 47H         |       |
| FFFFF1A8H | Interrupt control register         | CB1REIC  |     | √                          | √ |     | 47H         |       |
| FFFFF1AAH | Interrupt control register         | CB1RIC   |     | √                          | √ |     | 47H         |       |
| FFFFF1ACH | Interrupt control register         | CB1TIC   |     | √                          | √ |     | 47H         |       |
| FFFFF1AEH | Interrupt control register         | UA2REIC  |     | √                          | √ |     | 47H         |       |
| FFFFF1B0H | Interrupt control register         | UA2RIC   |     | √                          | √ |     | 47H         |       |
| FFFFF1B2H | Interrupt control register         | UA2TIC   |     | √                          | √ |     | 47H         |       |
| FFFFF1B4H | Interrupt control register         | CB2REIC  |     | √                          | √ |     | 47H         |       |
| FFFFF1B6H | Interrupt control register         | CB2RIC   |     | √                          | √ |     | 47H         |       |
| FFFFF1B8H | Interrupt control register         | CB2TIC   |     | √                          | √ |     | 47H         |       |
| FFFFF1BAH | Interrupt control register         | IICIC    |     | √                          | √ |     | 47H         |       |
| FFFFF1BCH | Interrupt control register         | AD0IC    |     | √                          | √ |     | 47H         |       |
| FFFFF1BEH | Interrupt control register         | AD1IC    |     | √                          | √ |     | 47H         |       |
| FFFFF1C0H | Interrupt control register         | AD2IC    |     | √                          | √ |     | 47H         |       |
| FFFFF1C2H | Interrupt control register         | TM0EQIC0 |     | √                          | √ |     | 47H         |       |
| FFFFF1C4H | Interrupt control register         | TM1EQIC0 |     | √                          | √ |     | 47H         |       |
| FFFFF1C6H | Interrupt control register         | TM2EQIC0 |     | √                          | √ |     | 47H         |       |
| FFFFF1C8H | Interrupt control register         | TM3EQIC0 |     | √                          | √ |     | 47H         |       |
| FFFFF1CAH | Interrupt control register         | ADT0IC   |     | √                          | √ |     | 47H         |       |
| FFFFF1CCH | Interrupt control register         | ADT1IC   | √   | √                          |   | 47H |             |       |
| FFFFF1FAH | In-service priority register       | ISPR     | R   | √                          | √ |     | 00H         |       |
| FFFFF1FCH | Command register                   | PRCMD    | W   |                            | √ |     | Undefined   |       |
| FFFFF1FEH | Power save control register        | PSC      | R/W | √                          | √ |     | 00H         |       |
| FFFFF200H | A/D0 conversion result register 0  | AD0CR0   | R   |                            |   | √   | 0000H       |       |
| FFFFF201H | A/D0 conversion result register 0H | AD0CR0H  |     |                            |   | √   | 00H         |       |
| FFFFF202H | A/D0 conversion result register 1  | AD0CR1   |     |                            |   |     | √           | 0000H |
| FFFFF203H | A/D0 conversion result register 1H | AD0CR1H  |     |                            |   | √   | 00H         |       |

| Address   | Function Register Name                                      | Symbol   | R/W | Bit Units for Manipulation |   |       | After Reset |
|-----------|---|----------|-----|----------------------------|---|-------|-------------|
|           |   |          |     | 1                          | 8 | 16    |             |
| FFFFF204H | A/D0 conversion result register 2                           | AD0CR2   | R   |                            |   | √     | 0000H       |
| FFFFF205H | A/D0 conversion result register 2H                          | AD0CR2H  |     |                            | √ |       | 00H         |
| FFFFF206H | A/D0 conversion result register 3                           | AD0CR3   |     |                            |   | √     | 0000H       |
| FFFFF207H | A/D0 conversion result register 3H                          | AD0CR3H  |     |                            | √ |       | 00H         |
| FFFFF208H | A/D0 conversion result register 4                           | AD0CR4   |     |                            |   | √     | 0000H       |
| FFFFF209H | A/D0 conversion result register 4H                          | AD0CR4H  |     |                            | √ |       | 00H         |
| FFFFF20AH | A/D0 conversion result register 5                           | AD0CR5   |     |                            |   | √     | 0000H       |
| FFFFF20BH | A/D0 conversion result register 5H                          | AD0CR5H  |     |                            | √ |       | 00H         |
| FFFFF20CH | A/D0 conversion result register 6                           | AD0CR6   |     |                            |   | √     | 0000H       |
| FFFFF20DH | A/D0 conversion result register 6H                          | AD0CR6H  |     |                            | √ |       | 00H         |
| FFFFF20EH | A/D0 conversion result register 7                           | AD0CR7   |     |                            |   | √     | 0000H       |
| FFFFF20FH | A/D0 conversion result register 7H                          | AD0CR7H  |     |                            | √ |       | 00H         |
| FFFFF210H | A/D0 conversion result register 8                           | AD0CR8   |     |                            |   | √     | 0000H       |
| FFFFF211H | A/D0 conversion result register 8H                          | AD0CR8H  |     |                            | √ |       | 00H         |
| FFFFF212H | A/D0 conversion result register 9                           | AD0CR9   |     |                            |   | √     | 0000H       |
| FFFFF213H | A/D0 conversion result register 9H                          | AD0CR9H  |     |                            | √ |       | 00H         |
| FFFFF214H | A/D0 conversion result register 10                          | AD0CR10  |     |                            |   | √     | 0000H       |
| FFFFF215H | A/D0 conversion result register 10H                         | AD0CR10H |     |                            | √ |       | 00H         |
| FFFFF216H | A/D0 conversion result register 11                          | AD0CR11  |     |                            |   | √     | 0000H       |
| FFFFF217H | A/D0 conversion result register 11H                         | AD0CR11H |     |                            | √ |       | 00H         |
| FFFFF218H | A/D0 conversion result register 12                          | AD0CR12  |     |                            |   | √     | 0000H       |
| FFFFF219H | A/D0 conversion result register 12H                         | AD0CR12H |     |                            | √ |       | 00H         |
| FFFFF21AH | A/D0 conversion result register 13                          | AD0CR13  |     |                            |   | √     | 0000H       |
| FFFFF21BH | A/D0 conversion result register 13H                         | AD0CR13H |     |                            | √ |       | 00H         |
| FFFFF21CH | A/D0 conversion result register 14                          | AD0CR14  |     |                            | √ | 0000H |             |
| FFFFF21DH | A/D0 conversion result register 14H                         | AD0CR14H |     | √                          |   | 00H   |             |
| FFFFF21EH | A/D0 conversion result register 15                          | AD0CR15  |     |                            | √ | 0000H |             |
| FFFFF21FH | A/D0 conversion result register 15H                         | AD0CR15H |     | √                          |   | 00H   |             |
| FFFFF220H | A/D converter 0 scan mode register                          | AD0SCM   | R/W |                            |   | √     | 0000H       |
| FFFFF220H | A/D converter 0 scan mode register L                        | AD0SCML  |     | √                          | √ |       | 00H         |
| FFFFF221H | A/D converter 0 scan mode register H                        | AD0SCMH  |     | √                          | √ |       | 00H         |
| FFFFF222H | A/D converter 0 conversion time control register            | AD0CTC   |     | √                          | √ |       | 00H         |
| FFFFF224H | A/D converter 0 conversion channel specification register   | AD0CHEN  |     |                            |   | √     | 0000H       |
| FFFFF224H | A/D converter 0 conversion channel specification register L | AD0CHENL |     | √                          | √ |       | 00H         |
| FFFFF225H | A/D converter 0 conversion channel specification register H | AD0CHENH |     | √                          | √ |       | 00H         |
| FFFFF230H | A/D converter 0 control register                            | AD0CTL0  |     | √                          | √ |       | 00H         |
| FFFFF231H | A/D converter 0 trigger select register                     | AD0TSEL  |     | √                          | √ |       | 10H         |
| FFFFF232H | A/D converter 0 channel specification register 1            | AD0CH1   |     | √                          | √ |       | 00H         |

| Address   | Function Register Name                                  | Symbol   | R/W | Bit Units for Manipulation |   |       | After Reset |
|-----------|---|----------|-----|----------------------------|---|-------|-------------|
|           |   |          |     | 1                          | 8 | 16    |             |
| FFFFF233H | A/D converter 0 channel specification register 2        | AD0CH2   | R/W | √                          | √ |       | 00H         |
| FFFFF240H | A/D0 conversion result expansion register 0             | AD0ECR0  | R   |                            |   | √     | 0000H       |
| FFFFF241H | A/D0 conversion result expansion register 0H            | AD0ECR0H |     |                            | √ |       | 00H         |
| FFFFF242H | A/D0 conversion result expansion register 1             | AD0ECR1  |     |                            |   | √     | 0000H       |
| FFFFF243H | A/D0 conversion result expansion register 1H            | AD0ECR1H |     |                            | √ |       | 00H         |
| FFFFF244H | A/D0 conversion result expansion register 2             | AD0ECR2  |     |                            |   | √     | 0000H       |
| FFFFF245H | A/D0 conversion result expansion register 2H            | AD0ECR2H |     |                            | √ |       | 00H         |
| FFFFF246H | A/D0 conversion result expansion register 3             | AD0ECR3  |     |                            |   | √     | 0000H       |
| FFFFF247H | A/D0 conversion result expansion register 3H            | AD0ECR3H |     |                            | √ |       | 00H         |
| FFFFF248H | A/D0 conversion result expansion register 4             | AD0ECR4  |     |                            |   | √     | 0000H       |
| FFFFF249H | A/D0 conversion result expansion register 4H            | AD0ECR4H |     |                            | √ |       | 00H         |
| FFFFF254H | A/D converter 0 flag register                           | AD0FLG   |     |                            | √ |       | 00H         |
| FFFFF255H | A/D converter 0 flag buffer register                    | AD0FLGB  |     |                            | √ |       | 00H         |
| FFFFF260H | Operational amplifier 0 control register 0              | OP0CTL0  |     | R/W                        |   | √     | 00H         |
| FFFFF261H | Comparator 0 control register 0                         | CMP0CTL0 |     |                            | √ |       | 00H         |
| FFFFF262H | Comparator 0 control register 1                         | CMP0CTL1 |     | R                          |   | √     | 00H         |
| FFFFF263H | Comparator 0 control register 2                         | CMP0CTL2 |     | R/W                        |   | √     | 00H         |
| FFFFF264H | Comparator 0 control register 3                         | CMP0CTL3 |     |                            | √ | 00H   |             |
| FFFFF270H | A/D converter 0 clock select register                   | AD0OCKS  |     |                            | √ | 00H   |             |
| FFFFF274H | A/D converter 1 clock select register                   | AD1OCKS  |     |                            | √ | 00H   |             |
| FFFFF278H | Comparator output digital noise elimination register 0L | CMPNFC0L |     |                            | √ | 00H   |             |
| FFFFF27AH | Comparator output digital noise elimination register 0F | CMPNFC0F |     |                            | √ | 00H   |             |
| FFFFF27CH | Comparator output digital noise elimination register 1L | CMPNFC1L |     |                            | √ | 00H   |             |
| FFFFF27EH | Comparator output digital noise elimination register 1F | CMPNFC1F |     |                            | √ | 00H   |             |
| FFFFF280H | A/D1 conversion result register 0                       | AD1CR0   | R   |                            |   | √     | 0000H       |
| FFFFF281H | A/D1 conversion result register 0H                      | AD1CR0H  |     |                            | √ |       | 00H         |
| FFFFF282H | A/D1 conversion result register 1                       | AD1CR1   |     |                            |   | √     | 0000H       |
| FFFFF283H | A/D1 conversion result register 1H                      | AD1CR1H  |     |                            | √ |       | 00H         |
| FFFFF284H | A/D1 conversion result register 2                       | AD1CR2   |     |                            |   | √     | 0000H       |
| FFFFF285H | A/D1 conversion result register 2H                      | AD1CR2H  |     |                            | √ |       | 00H         |
| FFFFF286H | A/D1 conversion result register 3                       | AD1CR3   |     |                            |   | √     | 0000H       |
| FFFFF287H | A/D1 conversion result register 3H                      | AD1CR3H  |     |                            | √ |       | 00H         |
| FFFFF288H | A/D1 conversion result register 4                       | AD1CR4   |     |                            |   | √     | 0000H       |
| FFFFF289H | A/D1 conversion result register 4H                      | AD1CR4H  |     |                            | √ |       | 00H         |
| FFFFF28AH | A/D1 conversion result register 5                       | AD1CR5   |     |                            |   | √     | 0000H       |
| FFFFF28BH | A/D1 conversion result register 5H                      | AD1CR5H  |     |                            | √ |       | 00H         |
| FFFFF28CH | A/D1 conversion result register 6                       | AD1CR6   |     |                            |   | √     | 0000H       |
| FFFFF28DH | A/D1 conversion result register 6H                      | AD1CR6H  |     |                            | √ |       | 00H         |
| FFFFF28EH | A/D1 conversion result register 7                       | AD1CR7   |     |                            |   | √     | 0000H       |
| FFFFF28FH | A/D1 conversion result register 7H                      | AD1CR7H  |     |                            | √ |       | 00H         |
| FFFFF290H | A/D1 conversion result register 8                       | AD1CR8   |     |                            | √ | 0000H |             |
| FFFFF291H | A/D1 conversion result register 8H                      | AD1CR8H  |     | √                          |   | 00H   |             |

| Address   | Function Register Name                                      | Symbol   | R/W | Bit Units for Manipulation |   |    | After Reset |       |
|-----------|---|----------|-----|----------------------------|---|----|-------------|-------|
|           |   |          |     | 1                          | 8 | 16 |             |       |
| FFFFF292H | A/D1 conversion result register 9                           | AD1CR9   | R   |                            |   | √  | 0000H       |       |
| FFFFF293H | A/D1 conversion result register 9H                          | AD1CR9H  |     |                            | √ |    |             | 00H   |
| FFFFF294H | A/D1 conversion result register 10                          | AD1CR10  |     |                            |   |    | √           | 0000H |
| FFFFF295H | A/D1 conversion result register 10H                         | AD1CR10H |     |                            | √ |    |             | 00H   |
| FFFFF296H | A/D1 conversion result register 11                          | AD1CR11  |     |                            |   |    | √           | 0000H |
| FFFFF297H | A/D1 conversion result register 11H                         | AD1CR11H |     |                            | √ |    |             | 00H   |
| FFFFF298H | A/D1 conversion result register 12                          | AD1CR12  |     |                            |   |    | √           | 0000H |
| FFFFF299H | A/D1 conversion result register 12H                         | AD1CR12H |     |                            | √ |    |             | 00H   |
| FFFFF29AH | A/D1 conversion result register 13                          | AD1CR13  |     |                            |   |    | √           | 0000H |
| FFFFF29BH | A/D1 conversion result register 13H                         | AD1CR13H |     |                            | √ |    |             | 00H   |
| FFFFF29CH | A/D1 conversion result register 14                          | AD1CR14  |     |                            |   |    | √           | 0000H |
| FFFFF29DH | A/D1 conversion result register 14H                         | AD1CR14H |     |                            | √ |    |             | 00H   |
| FFFFF29EH | A/D1 conversion result register 15                          | AD1CR15  |     |                            |   |    | √           | 0000H |
| FFFFF29FH | A/D1 conversion result register 15H                         | AD1CR15H |     |                            | √ |    |             | 00H   |
| FFFFF2A0H | A/D converter 1 scan mode register                          | AD1SCM   |     | R/W                        |   |    | √           | 0000H |
| FFFFF2A0H | A/D converter 1 scan mode register L                        | AD1SCML  | √   |                            | √ |    |             | 00H   |
| FFFFF2A1H | A/D converter 1 scan mode register H                        | AD1SCMH  | √   |                            | √ |    |             | 00H   |
| FFFFF2A2H | A/D converter 1 conversion time control register            | AD1CTC   | √   |                            | √ |    |             | 00H   |
| FFFFF2A4H | A/D converter 1 conversion channel specification register   | AD1CHEN  |     |                            |   |    | √           | 0000H |
| FFFFF2A4H | A/D converter 1 conversion channel specification register L | AD1CHENL | √   |                            | √ |    |             | 00H   |
| FFFFF2A5H | A/D converter 1 conversion channel specification register H | AD1CHENH | √   |                            | √ |    |             | 00H   |
| FFFFF2B0H | A/D converter 1 control register                            | AD1CTL0  | √   |                            | √ |    |             | 00H   |
| FFFFF2B1H | A/D converter 1 trigger select register                     | AD1TSEL  | √   |                            | √ |    |             | 10H   |
| FFFFF2B2H | A/D converter 1 channel specification register 1            | AD1CH1   | √   |                            | √ |    |             | 00H   |
| FFFFF2B3H | A/D converter 1 channel specification register 2            | AD1CH2   | √   | √                          |   |    | 00H         |       |
| FFFFF2C0H | A/D1 conversion result expansion register 0                 | AD1ECR0  | R   |                            |   | √  | 0000H       |       |
| FFFFF2C1H | A/D1 conversion result expansion register 0H                | AD1ECR0H |     |                            | √ |    |             | 00H   |
| FFFFF2C2H | A/D1 conversion result expansion register 1                 | AD1ECR1  |     |                            |   |    | √           | 0000H |
| FFFFF2C3H | A/D1 conversion result expansion register 1H                | AD1ECR1H |     |                            | √ |    |             | 00H   |
| FFFFF2C4H | A/D1 conversion result expansion register 2                 | AD1ECR2  |     |                            |   |    | √           | 0000H |
| FFFFF2C5H | A/D1 conversion result expansion register 2H                | AD1ECR2H |     |                            | √ |    |             | 00H   |
| FFFFF2C6H | A/D1 conversion result expansion register 3                 | AD1ECR3  |     |                            |   |    | √           | 0000H |
| FFFFF2C7H | A/D1 conversion result expansion register 3H                | AD1ECR3H |     |                            | √ |    |             | 00H   |
| FFFFF2C8H | A/D1 conversion result expansion register 4                 | AD1ECR4  |     |                            |   |    | √           | 0000H |
| FFFFF2C9H | A/D1 conversion result expansion register 4H                | AD1ECR4H |     |                            | √ |    |             | 00H   |
| FFFFF2D4H | A/D converter 1 flag register                               | AD1FLG   |     |                            |   |    | √           | 00H   |
| FFFFF2D5H | A/D converter 1 flag buffer register                        | AD1FLGB  |     |                            |   |    | √           | 00H   |
| FFFFF2E0H | Operational amplifier 1 control register 0                  | OP1CTL0  | R/W |                            |   | √  | 00H         |       |
| FFFFF2E1H | Comparator 1 control register 0                             | CMP1CTL0 |     |                            |   |    | √           | 00H   |
| FFFFF2E2H | Comparator 1 control register 1                             | CMP1CTL1 | R   |                            |   | √  | 00H         |       |

| Address   | Function Register Name  | Symbol   | R/W | Bit Units for Manipulation |   |       | After Reset |
|-----------|---|----------|-----|----------------------------|---|-------|-------------|
|           |   |          |     | 1                          | 8 | 16    |             |
| FFFFF2E3H | Comparator 1 control register 2                                 | CMP1CTL2 | R/W |                            | √ |       | 00H         |
| FFFFF2E4H | Comparator 1 control register 3                                 | CMP1CTL3 |     |                            | √ |       | 00H         |
| FFFFF2F0H | A/D trigger falling edge specification register                 | ADTF     |     | √                          | √ |       | 00H         |
| FFFFF2F2H | A/D trigger rising edge specification register                  | ADTR     |     | √                          | √ |       | 00H         |
| FFFFF2F4H | Comparator output interrupt falling edge specification register | CMPOF    |     | √                          | √ |       | 00H         |
| FFFFF2F6H | Comparator output interrupt rising edge specification register  | CMPOR    |     | √                          | √ |       | 00H         |
| FFFFF2F8H | A/DLDTRG1 input select register                                 | ADLTS1   |     |                            | √ |       | 00H         |
| FFFFF2FAH | A/DLDTRG2 input select register                                 | ADLTS2   |     |                            | √ |       | 00H         |
| FFFFF310H | Digital noise elimination 0 control register 14                 | INTNFC14 |     |                            | √ |       | 00H         |
| FFFFF312H | Digital noise elimination 0 control register 15                 | INTNFC15 |     |                            | √ |       | 00H         |
| FFFFF314H | Digital noise elimination 0 control register 16                 | INTNFC16 |     |                            | √ |       | 00H         |
| FFFFF400H | Port 0 register   | P0       |     | √                          | √ |       | Undefined   |
| FFFFF402H | Port 1 register   | P1       |     | √                          | √ |       | Undefined   |
| FFFFF404H | Port 2 register   | P2       |     | √                          | √ |       | Undefined   |
| FFFFF406H | Port 3 register   | P3       |     | √                          | √ |       | Undefined   |
| FFFFF408H | Port 4 register   | P4       |     | √                          | √ |       | Undefined   |
| FFFFF420H | Port 0 mode register  | PM0      |     | √                          | √ |       | FFH         |
| FFFFF422H | Port 1 mode register  | PM1      |     | √                          | √ |       | FFH         |
| FFFFF424H | Port 2 mode register  | PM2      |     | √                          | √ |       | FFH         |
| FFFFF426H | Port 3 mode register  | PM3      |     | √                          | √ |       | FFH         |
| FFFFF428H | Port 4 mode register  | PM4      |     | √                          | √ |       | FFH         |
| FFFFF440H | Port 0 mode control register                                    | PMC0     |     | √                          | √ |       | 00H         |
| FFFFF442H | Port 1 mode control register                                    | PMC1     |     | √                          | √ |       | 00H         |
| FFFFF444H | Port 2 mode control register                                    | PMC2     |     | √                          | √ |       | 00H         |
| FFFFF446H | Port 3 mode control register                                    | PMC3     |     | √                          | √ |       | 00H         |
| FFFFF448H | Port 4 mode control register                                    | PMC4     |     | √                          | √ |       | 00H         |
| FFFFF460H | Port 0 function control register                                | PFC0     |     | √                          | √ |       | 00H         |
| FFFFF462H | Port 1 function control register                                | PFC1     |     | √                          | √ |       | 00H         |
| FFFFF464H | Port 2 function control register                                | PFC2     |     | √                          | √ |       | 00H         |
| FFFFF466H | Port 3 function control register                                | PFC3     |     | √                          | √ |       | 00H         |
| FFFFF468H | Port 4 function control register                                | PFC4     |     | √                          | √ |       | 00H         |
| FFFFF480H | Bus cycle type configuration register 0                         | BCT0     |     |                            |   | √     | CCCCH       |
| FFFFF484H | Data wait control register 0                                    | DWC0     |     |                            | √ | 7777H |             |
| FFFFF488H | Address wait control register                                   | AWC      |     |                            | √ | FFFFH |             |
| FFFFF48AH | Bus cycle control register                                      | BCC      |     |                            | √ | AAAAH |             |
| FFFFF48EH | Bus clock division control register                             | DVC      |     | √                          |   | 81H   |             |
| FFFFF540H | TMM0 control register 0   | TM0CTL0  | √   | √                          |   | 00H   |             |
| FFFFF544H | TMM0 compare register 0   | TM0CMP0  |     |                            | √ | 0000H |             |
| FFFFF550H | TMM1 control register 0   | TM1CTL0  | √   | √                          |   | 00H   |             |



| Address   | Function Register Name                         | Symbol                  | R/W | Bit Units for Manipulation |   |    | After Reset |
|-----------|--|-------------------------|-----|----------------------------|---|----|-------------|
|           |  |                         |     | 1                          | 8 | 16 |             |
| FFFFF554H | TMM1 compare register 0                        | TM1CMP0                 | R/W |                            |   | √  | 0000H       |
| FFFFF560H | TMM2 control register 0                        | TM2CTL0                 |     | √                          | √ |    | 00H         |
| FFFFF564H | TMM2 compare register 0                        | TM2CMP0                 |     |                            |   | √  | 0000H       |
| FFFFF570H | TMM3 control register 0                        | TM3CTL0                 |     | √                          | √ |    | 00H         |
| FFFFF574H | TMM3 compare register 0                        | TM3CMP0                 |     |                            |   | √  | 0000H       |
| FFFFF580H | TMT0 control register 0                        | TT0CTL0                 |     | √                          | √ |    | 00H         |
| FFFFF581H | TMT0 control register 1                        | TT0CTL1                 |     | √                          | √ |    | 00H         |
| FFFFF582H | TMT0 control register 2                        | TT0CTL2 <sup>Note</sup> |     | √                          | √ |    | 00H         |
| FFFFF583H | TMT0 I/O control register 0                    | TT0IOC0 <sup>Note</sup> |     | √                          | √ |    | 00H         |
| FFFFF584H | TMT0 I/O control register 1                    | TT0IOC1 <sup>Note</sup> |     | √                          | √ |    | 00H         |
| FFFFF585H | TMT0 I/O control register 2                    | TT0IOC2 <sup>Note</sup> |     | √                          | √ |    | 00H         |
| FFFFF586H | TMT0 I/O control register 3                    | TT0IOC3 <sup>Note</sup> |     | √                          | √ |    | 00H         |
| FFFFF587H | TMT0 option register 0                         | TT0OPT0                 |     | √                          | √ |    | 00H         |
| FFFFF588H | TMT0 option register 1                         | TT0OPT1 <sup>Note</sup> |     | √                          | √ |    | 00H         |
| FFFFF58AH | TMT0 capture/compare register 0                | TT0CCR0                 |     |                            |   | √  | 0000H       |
| FFFFF58CH | TMT0 capture/compare register 1                | TT0CCR1                 |     |                            |   | √  | 0000H       |
| FFFFF58EH | TMT0 counter read buffer register              | TT0CNT                  | R   |                            |   | √  | 0000H       |
| FFFFF590H | TMT0 counter write register                    | TT0TCW <sup>Note</sup>  | R/W |                            |   | √  | 0000H       |
| FFFFF5A0H | Digital noise elimination 2 control register 0 | TTNFC0 <sup>Note</sup>  |     |                            | √ |    | 00H         |
| FFFFF5A2H | Digital noise elimination 2 control register 1 | TTNFC1                  |     |                            | √ |    | 00H         |
| FFFFF5A4H | TMT0 capture input select register             | TTISL0 <sup>Note</sup>  |     |                            | √ |    | Undefined   |
| FFFFF5A6H | TMT1 capture input select register             | TTISL1                  |     |                            | √ |    | Undefined   |
| FFFFF5C0H | TMT1 control register 0                        | TT1CTL0                 |     | √                          | √ |    | 00H         |
| FFFFF5C1H | TMT1 control register 1                        | TT1CTL1                 |     | √                          | √ |    | 00H         |
| FFFFF5C2H | TMT1 control register 2                        | TT1CTL2                 |     | √                          | √ |    | 00H         |
| FFFFF5C3H | TMT1 I/O control register 0                    | TT1IOC0                 |     | √                          | √ |    | 00H         |
| FFFFF5C4H | TMT1 I/O control register 1                    | TT1IOC1                 |     | √                          | √ |    | 00H         |
| FFFFF5C5H | TMT1 I/O control register 2                    | TT1IOC2                 |     | √                          | √ |    | 00H         |
| FFFFF5C6H | TMT1 I/O control register 3                    | TT1IOC3                 |     | √                          | √ |    | 00H         |
| FFFFF5C7H | TMT1 option register 0                         | TT1OPT0                 |     | √                          | √ |    | 00H         |
| FFFFF5C8H | TMT1 option register 1                         | TT1OPT1                 |     | √                          | √ |    | 00H         |
| FFFFF5CAH | TMT1 capture/compare register 0                | TT1CCR0                 |     |                            |   | √  | 0000H       |
| FFFFF5CCH | TMT1 capture/compare register 1                | TT1CCR1                 |     |                            |   | √  | 0000H       |
| FFFFF5CEH | TMT1 counter read buffer register              | TT1CNT                  | R   |                            |   | √  | 0000H       |
| FFFFF5D0H | TMT1 counter write register                    | TT1TCW                  | R/W |                            |   | √  | 0000H       |
| FFFFF5E0H | TAB0 control register 0                        | TAB0CTL0                |     | √                          | √ |    | 00H         |
| FFFFF5E1H | TAB0 control register 1                        | TAB0CTL1                |     | √                          | √ |    | 00H         |
| FFFFF5E2H | TAB0 I/O control register 0                    | TAB0IOC0                |     | √                          | √ |    | 00H         |
| FFFFF5E3H | TAB0 I/O control register 1                    | TAB0IOC1                |     | √                          | √ |    | 00H         |
| FFFFF5E4H | TAB0 I/O control register 2                    | TAB0IOC2                |     | √                          | √ |    | 00H         |

**Note** V850E/IG3 only

| Address   | Function Register Name                    | Symbol                   | R/W | Bit Units for Manipulation |   |    | After Reset |       |
|-----------|---|--------------------------|-----|----------------------------|---|----|-------------|-------|
|           |   |                          |     | 1                          | 8 | 16 |             |       |
| FFFFF5E5H | TAB0 option register 0                    | TAB0OPT0                 | R/W | √                          | √ |    | 00H         |       |
| FFFFF5E6H | TAB0 capture/compare register 0           | TAB0CCR0                 |     |                            |   | √  | 0000H       |       |
| FFFFF5E8H | TAB0 capture/compare register 1           | TAB0CCR1                 |     |                            |   | √  | 0000H       |       |
| FFFFF5EAH | TAB0 capture/compare register 2           | TAB0CCR2                 |     |                            |   | √  | 0000H       |       |
| FFFFF5ECH | TAB0 capture/compare register 3           | TAB0CCR3                 |     |                            |   | √  | 0000H       |       |
| FFFFF5EEH | TAB0 counter read buffer register         | TAB0CNT                  | R   |                            |   | √  | 0000H       |       |
| FFFFF600H | TAB0 option register 1                    | TAB0OPT1                 | R/W | √                          | √ |    | 00H         |       |
| FFFFF601H | TAB0 option register 2                    | TAB0OPT2                 |     | √                          | √ |    | 00H         |       |
| FFFFF602H | TAB0 I/O control register 3               | TAB0IOC3                 |     | √                          | √ |    | A8H         |       |
| FFFFF603H | TAB0 option register 3                    | TAB0OPT3                 |     | √                          | √ |    | 00H         |       |
| FFFFF604H | TAB0 deadtime compare register            | TAB0DTC                  |     |                            |   | √  | 0000H       |       |
| FFFFF610H | High-impedance output control register 00 | HZA0CTL0                 |     | √                          | √ |    | 00H         |       |
| FFFFF611H | High-impedance output control register 01 | HZA0CTL1                 |     | √                          | √ |    | 00H         |       |
| FFFFF618H | High-impedance output control register 10 | HZA1CTL0                 |     | √                          | √ |    | 00H         |       |
| FFFFF619H | High-impedance output control register 11 | HZA1CTL1 <sup>Note</sup> |     | √                          | √ |    | 00H         |       |
| FFFFF620H | TAB1 control register 0                   | TAB1CTL0                 |     | √                          | √ |    | 00H         |       |
| FFFFF621H | TAB1 control register 1                   | TAB1CTL1                 |     | √                          | √ |    | 00H         |       |
| FFFFF622H | TAB1 I/O control register 0               | TAB1IOC0                 |     | √                          | √ |    | 00H         |       |
| FFFFF623H | TAB1 I/O control register 1               | TAB1IOC1                 |     | √                          | √ |    | 00H         |       |
| FFFFF624H | TAB1 I/O control register 2               | TAB1IOC2                 |     | √                          | √ |    | 00H         |       |
| FFFFF625H | TAB1 option register 0                    | TAB1OPT0                 |     | √                          | √ |    | 00H         |       |
| FFFFF626H | TAB1 capture/compare register 0           | TAB1CCR0                 |     |                            |   | √  | 0000H       |       |
| FFFFF628H | TAB1 capture/compare register 1           | TAB1CCR1                 |     |                            |   | √  | 0000H       |       |
| FFFFF62AH | TAB1 capture/compare register 2           | TAB1CCR2                 |     |                            |   | √  | 0000H       |       |
| FFFFF62CH | TAB1 capture/compare register 3           | TAB1CCR3                 |     |                            |   | √  | 0000H       |       |
| FFFFF62EH | TAB1 counter read buffer register         | TAB1CNT                  | R   |                            |   | √  | 0000H       |       |
| FFFFF640H | TAB1 option register 1                    | TAB1OPT1                 | R/W | √                          | √ |    | 00H         |       |
| FFFFF641H | TAB1 option register 2                    | TAB1OPT2                 |     | √                          | √ |    | 00H         |       |
| FFFFF642H | TAB1 I/O control register 3               | TAB1IOC3                 |     | √                          | √ |    | A8H         |       |
| FFFFF643H | TAB1 option register 3                    | TAB1OPT3                 |     | √                          | √ |    | 00H         |       |
| FFFFF644H | TAB1 deadtime compare register            | TAB1DTC                  |     |                            |   | √  | 0000H       |       |
| FFFFF650H | High-impedance output control register 20 | HZA2CTL0                 |     | √                          | √ |    | 00H         |       |
| FFFFF651H | High-impedance output control register 21 | HZA2CTL1                 |     | √                          | √ |    | 00H         |       |
| FFFFF658H | High-impedance output control register 30 | HZA3CTL0                 |     | √                          | √ |    | 00H         |       |
| FFFFF659H | High-impedance output control register 31 | HZA3CTL1                 |     | √                          | √ |    | 00H         |       |
| FFFFF660H | TAA0 control register 0                   | TAA0CTL0                 |     | √                          | √ |    | 00H         |       |
| FFFFF661H | TAA0 control register 1                   | TAA0CTL1                 |     | √                          | √ |    | 00H         |       |
| FFFFF665H | TAA0 option register 0                    | TAA0OPT0                 |     | √                          | √ |    | 00H         |       |
| FFFFF666H | TAA0 capture/compare register 0           | TAA0CCR0                 |     |                            |   | √  | 0000H       |       |
| FFFFF668H | TAA0 capture/compare register 1           | TAA0CCR1                 |     |                            |   | √  | 0000H       |       |
| FFFFF66AH | TAA0 counter read buffer register         | TAA0CNT                  |     | R                          |   |    | √           | 0000H |

**Note** V850E/IG3 only

| Address   | Function Register Name                         | Symbol   | R/W | Bit Units for Manipulation |   |             | After Reset |
|-----------|--|----------|-----|----------------------------|---|-------------|-------------|
|           |  |          |     | 1                          | 8 | 16          |             |
| FFFFF680H | TAA1 control register 0                        | TAA1CTL0 | R/W | √                          | √ |             | 00H         |
| FFFFF681H | TAA1 control register 1                        | TAA1CTL1 |     | √                          | √ |             | 00H         |
| FFFFF685H | TAA1 option register 0                         | TAA1OPT0 |     | √                          | √ |             | 00H         |
| FFFFF686H | TAA1 capture/compare register 0                | TAA1CCR0 |     |                            |   | √           | 0000H       |
| FFFFF688H | TAA1 capture/compare register 1                | TAA1CCR1 |     |                            |   | √           | 0000H       |
| FFFFF68AH | TAA1 counter read buffer register              | TAA1CNT  | R   |                            |   | √           | 0000H       |
| FFFFF6A0H | TAA2 control register 0                        | TAA2CTL0 | R/W | √                          | √ |             | 00H         |
| FFFFF6A1H | TAA2 control register 1                        | TAA2CTL1 |     | √                          | √ |             | 00H         |
| FFFFF6A2H | TAA2 I/O control register 0                    | TAA2IOC0 |     | √                          | √ |             | 00H         |
| FFFFF6A3H | TAA2 I/O control register 1                    | TAA2IOC1 |     | √                          | √ |             | 00H         |
| FFFFF6A4H | TAA2 I/O control register 2                    | TAA2IOC2 |     | √                          | √ |             | 00H         |
| FFFFF6A5H | TAA2 option register 0                         | TAA2OPT0 |     | √                          | √ |             | 00H         |
| FFFFF6A6H | TAA2 capture/compare register 0                | TAA2CCR0 |     |                            |   | √           | 0000H       |
| FFFFF6A8H | TAA2 capture/compare register 1                | TAA2CCR1 |     |                            |   | √           | 0000H       |
| FFFFF6AAH | TAA2 counter read buffer register              | TAA2CNT  | R   |                            |   | √           | 0000H       |
| FFFFF6C0H | Oscillation stabilization time select register | OSTS     | R/W |                            | √ |             | 04H         |
| FFFFF6D0H | Watchdog timer mode register                   | WDTM     |     |                            | √ |             | 67H         |
| FFFFF6D1H | Watchdog timer enable register                 | WDTE     |     |                            | √ |             | 1AH         |
| FFFFF700H | Port 0 function control expansion register     | PFCE0    |     | √                          | √ |             | 00H         |
| FFFFF702H | Port 1 function control expansion register     | PFCE1    |     | √                          | √ |             | 00H         |
| FFFFF704H | Port 2 function control expansion register     | PFCE2    |     | √                          | √ |             | 00H         |
| FFFFF706H | Port 3 function control expansion register     | PFCE3    |     | √                          | √ |             | 00H         |
| FFFFF708H | Port 4 function control expansion register     | PFCE4    |     | √                          | √ |             | 00H         |
| FFFFF802H | System status register                         | SYS      |     | √                          | √ |             | 00H         |
| FFFFF810H | DMA trigger factor register 0                  | DTFR0    |     | √                          | √ |             | 00H         |
| FFFFF812H | DMA trigger factor register 1                  | DTFR1    |     | √                          | √ |             | 00H         |
| FFFFF814H | DMA trigger factor register 2                  | DTFR2    |     | √                          | √ |             | 00H         |
| FFFFF816H | DMA trigger factor register 3                  | DTFR3    |     | √                          | √ |             | 00H         |
| FFFFF820H | Power save mode register                       | PSMR     |     | √                          | √ |             | 00H         |
| FFFFF828H | Processor clock control register               | PCC      |     | √                          | √ |             | 03H         |
| FFFFF82CH | PLL control register                           | PLLCTL   | √   | √                          |   | 01H         |             |
| FFFFF870H | Clock monitor mode register                    | CLM      | √   | √                          |   | 00H         |             |
| FFFFF888H | Reset source flag register                     | RESF     | √   | √                          |   | 00H/10H/01H |             |
| FFFFF890H | Low-voltage detection register                 | LVIM     | √   | √                          |   | 00H         |             |
| FFFFF891H | Low-voltage detection level select register    | LVIS     |     | √                          |   | 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         |             |

| Address   | Function Register Name            | Symbol                   | R/W | Bit Units for Manipulation |   |    | After Reset |
|-----------|-----------------------------------|--------------------------|-----|----------------------------|---|----|-------------|
|           |                                   |                          |     | 1                          | 8 | 16 |             |
| FFFFFA07H | UARTA0 transmit data register     | UA0TX                    | R/W |                            | √ |    | FFH         |
| FFFFFA10H | UARTA1 control register 0         | UA1CTL0                  |     | √                          | √ |    | 10H         |
| FFFFFA11H | UARTA1 control register 1         | UA1CTL1                  |     |                            | √ |    | 00H         |
| FFFFFA12H | UARTA1 control register 2         | UA1CTL2                  |     |                            | √ |    | FFH         |
| FFFFFA13H | UARTA1 option control register 0  | UA1OPT0                  |     | √                          | √ |    | 14H         |
| FFFFFA14H | UARTA1 status register            | UA1STR                   |     | √                          | √ |    | 00H         |
| FFFFFA16H | UARTA1 receive data register      | UA1RX                    | R   |                            | √ |    | FFH         |
| FFFFFA17H | UARTA1 transmit data register     | UA1TX                    | R/W |                            | √ |    | FFH         |
| FFFFFA20H | UARTA2 control register 0         | UA2CTL0                  |     | √                          | √ |    | 10H         |
| FFFFFA21H | UARTA2 control register 1         | UA2CTL1                  |     |                            | √ |    | 00H         |
| FFFFFA22H | UARTA2 control register 2         | UA2CTL2                  |     |                            | √ |    | FFH         |
| FFFFFA23H | UARTA2 option control register 0  | UA2OPT0                  |     | √                          | √ |    | 14H         |
| FFFFFA24H | UARTA2 status register            | UA2STR                   |     | R/W                        | √ | √  |             |
| FFFFFA26H | UARTA2 receive data register      | UA2RX                    | R   |                            | √ |    | FFH         |
| FFFFFA27H | UARTA2 transmit data register     | UA2TX                    | R/W |                            | √ |    | FFH         |
| FFFFFA40H | UARTB control register 0          | UBCTL0                   |     | √                          | √ |    | 10H         |
| FFFFFA42H | UARTB control register 2          | UBCTL2                   |     |                            |   | √  | FFFFH       |
| FFFFFA44H | UARTB status register             | UBSTR                    |     | √                          | √ |    | 00H         |
| FFFFFA46H | UARTB receive data register AP    | UBRXAP                   | R   |                            |   | √  | 00FFH       |
| FFFFFA46H | UARTB receive data register       | UBRX                     |     |                            | √ |    | FFH         |
| FFFFFA48H | UARTB transmit data register      | UBTX                     | W   |                            | √ |    | FFH         |
| FFFFFA4AH | UARTBFIFO control register 0      | UBFIC0                   | R/W | √                          | √ |    | 00H         |
| FFFFFA4BH | UARTBFIFO control register 1      | UBFIC1                   |     | √                          | √ |    | 00H         |
| FFFFFA4CH | UARTBFIFO control register 2      | UBFIC2                   |     |                            |   | √  | 0000H       |
| FFFFFA4CH | UARTBFIFO control register 2L     | UBFIC2L                  |     |                            | √ |    | 00H         |
| FFFFFA4DH | UARTBFIFO control register 2H     | UBFIC2H                  |     |                            | √ |    | 00H         |
| FFFFFA4EH | UARTBFIFO status register 0       | UBFIS0                   |     | R                          |   | √  |             |
| FFFFFA4FH | UARTBFIFO status register 1       | UBFIS1                   |     |                            | √ |    | 10H         |
| FFFFFB00H | TAA3 control register 0           | TAA3CTL0                 | R/W | √                          | √ |    | 00H         |
| FFFFFB01H | TAA3 control register 1           | TAA3CTL1                 |     | √                          | √ |    | 00H         |
| FFFFFB02H | TAA3 I/O control register 0       | TAA3IOC0 <sup>Note</sup> |     | √                          | √ |    | 00H         |
| FFFFFB03H | TAA3 I/O control register 1       | TAA3IOC1 <sup>Note</sup> |     | √                          | √ |    | 00H         |
| FFFFFB04H | TAA3 I/O control register 2       | TAA3IOC2 <sup>Note</sup> |     | √                          | √ |    | 00H         |
| FFFFFB05H | TAA3 option register 0            | TAA3OPT0                 |     | √                          | √ |    | 00H         |
| FFFFFB06H | TAA3 capture/compare register 0   | TAA3CCR0                 |     |                            |   | √  | 0000H       |
| FFFFFB08H | TAA3 capture/compare register 1   | TAA3CCR1                 |     |                            |   | √  | 0000H       |
| FFFFFB0AH | TAA3 counter read buffer register | TAA3CNT                  |     | R                          |   | √  | 0000H       |
| FFFFFB20H | TAA4 control register 0           | TAA4CTL0                 |     | R/W                        | √ | √  |             |
| FFFFFB21H | TAA4 control register 1           | TAA4CTL1                 | √   |                            | √ |    | 00H         |
| FFFFFB22H | TAA4 I/O control register 0       | TAA4IOC0                 | √   |                            | √ |    | 00H         |
| FFFFFB23H | TAA4 I/O control register 1       | TAA4IOC1                 | √   |                            | √ |    | 00H         |

**Note** V850E/IG3 only

| Address   | Function Register Name                                   | Symbol                  | R/W | Bit Units for Manipulation |   |     | After Reset |
|-----------|--|-------------------------|-----|----------------------------|---|-----|-------------|
|           |  |                         |     | 1                          | 8 | 16  |             |
| FFFFFB24H | TAA4 I/O control register 2                              | TAA4IOC2                | R/W | √                          | √ |     | 00H         |
| FFFFFB25H | TAA4 option register 0                                   | TAA4OPT0                |     | √                          | √ |     | 00H         |
| FFFFFB26H | TAA4 capture/compare register 0                          | TAA4CCR0                |     |                            |   | √   | 0000H       |
| FFFFFB28H | TAA4 capture/compare register 1                          | TAA4CCR1                |     |                            |   | √   | 0000H       |
| FFFFFB2AH | TAA4 counter read buffer register                        | TAA4CNT                 | R   |                            |   | √   | 0000H       |
| FFFFFB40H | Digital noise elimination 1 control register 2           | TANFC2                  | R/W |                            | √ |     | 00H         |
| FFFFFB42H | Digital noise elimination 1 control register 3           | TANFC3 <sup>Note</sup>  |     |                            | √ |     | 00H         |
| FFFFFB44H | Digital noise elimination 1 control register 4           | TANFC4                  |     |                            | √ |     | 00H         |
| FFFFFB80H | A/D converter 2 mode register 0                          | AD2M0                   |     | √                          | √ |     | 00H         |
| FFFFFB81H | A/D converter 2 mode register 1                          | AD2M1                   |     | √                          |   | 00H |             |
| FFFFFB82H | A/D converter 2 channel specification register           | AD2S                    |     | √                          |   | 00H |             |
| FFFFFB90H | A/D2 conversion result register 0                        | AD2CR0                  | R   |                            |   | √   | 0000H       |
| FFFFFB91H | A/D2 conversion result register 0H                       | AD2CR0H                 |     |                            | √ |     | 00H         |
| FFFFFB92H | A/D2 conversion result register 1                        | AD2CR1                  |     |                            |   | √   | 0000H       |
| FFFFFB93H | A/D2 conversion result register 1H                       | AD2CR1H                 |     |                            | √ |     | 00H         |
| FFFFFB94H | A/D2 conversion result register 2                        | AD2CR2                  |     |                            |   | √   | 0000H       |
| FFFFFB95H | A/D2 conversion result register 2H                       | AD2CR2H                 |     |                            | √ |     | 00H         |
| FFFFFB96H | A/D2 conversion result register 3                        | AD2CR3                  |     |                            |   | √   | 0000H       |
| FFFFFB97H | A/D2 conversion result register 3H                       | AD2CR3H                 |     |                            | √ |     | 00H         |
| FFFFFB98H | A/D2 conversion result register 4                        | AD2CR4 <sup>Note</sup>  |     |                            |   | √   | 0000H       |
| FFFFFB99H | A/D2 conversion result register 4H                       | AD2CR4H <sup>Note</sup> |     |                            | √ |     | 00H         |
| FFFFFB9AH | A/D2 conversion result register 5                        | AD2CR5 <sup>Note</sup>  |     |                            |   | √   | 0000H       |
| FFFFFB9BH | A/D2 conversion result register 5H                       | AD2CR5H <sup>Note</sup> |     |                            | √ |     | 00H         |
| FFFFFB9CH | A/D2 conversion result register 6                        | AD2CR6 <sup>Note</sup>  |     |                            |   | √   | 0000H       |
| FFFFFB9DH | A/D2 conversion result register 6H                       | AD2CR6H <sup>Note</sup> |     |                            | √ |     | 00H         |
| FFFFFB9EH | A/D2 conversion result register 7                        | AD2CR7 <sup>Note</sup>  |     |                            |   | √   | 0000H       |
| FFFFFB9FH | A/D2 conversion result register 7H                       | AD2CR7H <sup>Note</sup> |     |                            | √ |     | 00H         |
| FFFFFB0H  | Port 7 register  | P7                      | R   | √                          | √ |     | Undefined   |
| FFFFFB8H  | Port 7 mode control register                             | PMC7                    | R/W | √                          | √ |     | 00H         |
| FFFFC00H  | External interrupt falling edge specification register 0 | INTF0                   |     | √                          | √ |     | 00H         |
| FFFFC02H  | External interrupt falling edge specification register 1 | INTF1                   |     | √                          | √ |     | 00H         |
| FFFFC04H  | External interrupt falling edge specification register 2 | INTF2                   |     | √                          | √ |     | 00H         |
| FFFFC20H  | External interrupt rising edge specification register 0  | INTR0                   |     | √                          | √ |     | 00H         |
| FFFFC22H  | External interrupt rising edge specification register 1  | INTR1                   |     | √                          | √ |     | 00H         |
| FFFFC24H  | External interrupt rising edge specification register 2  | INTR2                   |     | √                          | √ |     | 00H         |
| FFFFC40H  | Pull-up resistor option register 0                       | PU0                     |     | √                          | √ |     | 00H         |
| FFFFC42H  | Pull-up resistor option register 1                       | PU1                     |     | √                          | √ |     | 00H         |
| FFFFC44H  | Pull-up resistor option register 2                       | PU2                     |     | √                          | √ |     | 00H         |
| FFFFC46H  | Pull-up resistor option register 3                       | PU3                     |     | √                          | √ |     | 00H         |
| FFFFC48H  | Pull-up resistor option register 4                       | PU4                     |     | √                          | √ |     | 00H         |

**Note** V850E/IG3 only

CHAPTER 3 CPU FUNCTION

(14/14)

| Address   | Function Register Name               | Symbol  | R/W | Bit Units for Manipulation |   |    | After Reset |       |
|-----------|--------------------------------------|---------|-----|----------------------------|---|----|-------------|-------|
|           |                                      |         |     | 1                          | 8 | 16 |             |       |
| FFFFFC66H | Port 3 function register             | PF3     | R/W | √                          | √ |    | 00H         |       |
| FFFFFD00H | CSIB0 control register 0             | CB0CTL0 | R/W | √                          | √ |    | 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 | R/W | √                          | √ |    | 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   |
| FFFFFD20H | CSIB2 control register 0             | CB2CTL0 | R/W | √                          | √ |    | 01H         |       |
| FFFFFD21H | CSIB2 control register 1             | CB2CTL1 |     | √                          | √ |    | 00H         |       |
| FFFFFD22H | CSIB2 control register 2             | CB2CTL2 |     |                            | √ |    | 00H         |       |
| FFFFFD23H | CSIB2 status register                | CB2STR  |     | √                          | √ |    | 00H         |       |
| FFFFFD24H | CSIB2 receive data register          | CB2RX   |     | R                          |   |    | √           | 0000H |
| FFFFFD24H | CSIB2 receive data register L        | CB2RXL  |     |                            | √ |    | 00H         |       |
| FFFFFD26H | CSIB2 transmit data register         | CB2TX   | R/W |                            |   | √  | 0000H       |       |
| FFFFFD26H | CSIB2 transmit data register L       | CB2TXL  |     |                            |   | √  |             | 00H   |
| FFFFFD80H | IIC shift register 0                 | IIC0    | R/W |                            | √ |    | 00H         |       |
| FFFFFD82H | IIC control register 0               | IICC0   |     | √                          | √ |    | 00H         |       |
| FFFFFD83H | Slave address register 0             | SVA0    |     |                            | √ |    | 00H         |       |
| FFFFFD84H | IIC clock select register 0          | IICCL0  |     | √                          | √ |    | 00H         |       |
| FFFFFD85H | IIC function expansion register 0    | IICX0   |     | √                          | √ |    | 00H         |       |
| FFFFFD86H | IIC status register 0                | IICS0   | R   | √                          | √ |    | 00H         |       |
| FFFFFD8AH | IIC flag register 0                  | IICF0   | R/W | √                          | √ |    | 00H         |       |
| FFFFFD90H | IIC OPS clock select register        | IICOCKS |     |                            |   | √  |             | 00H   |
| FFFFF44H  | Pull-up resistor option register DL  | PUDL    |     |                            |   |    | √           | 0000H |
| FFFFF44H  | Pull-up resistor option register DLL | PUDLL   |     | √                          | √ |    | 00H         |       |
| FFFFF45H  | Pull-up resistor option register DLH | PUDLH   |     | √                          | √ |    | 00H         |       |

### 3.4.8 Special registers

Special registers are registers that are protected from being written with illegal data due to a program loop. The V850E/IF3 and V850E/IG3 have the following five special registers.

- Power save control register (PSC)
- Processor clock control register (PCC)
- Reset source flag register (RESF)
- Clock monitor mode register (CLM)
- Low-voltage detection register (LVIM)

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

**(1) Setting data to special registers**

Set data to the special registers in the following sequence.

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

[Example] With PSC register (setting standby mode)

```

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

```

There is no special sequence to read a special register.

**Note** Five NOP instructions or more must be inserted immediately after setting the IDLE mode or STOP mode (by setting the PSC.STB bit to 1).

- Cautions**
1. When a store instruction is executed to store data in the command register, interrupts are not acknowledged. This is because it is assumed that steps <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 command register, use the same general-purpose register used to set the special register (<3> in Example) by using the store instruction to write data to the command register (<2> in Example). The same applies when a general-purpose register is used for addressing.  
An example of setting the special register (<3> in Example) by using the bit manipulation instruction is shown below.

```
CLR1 4, RESF[r0]
```

3. Before executing this processing, terminate all DMA transfer operations.



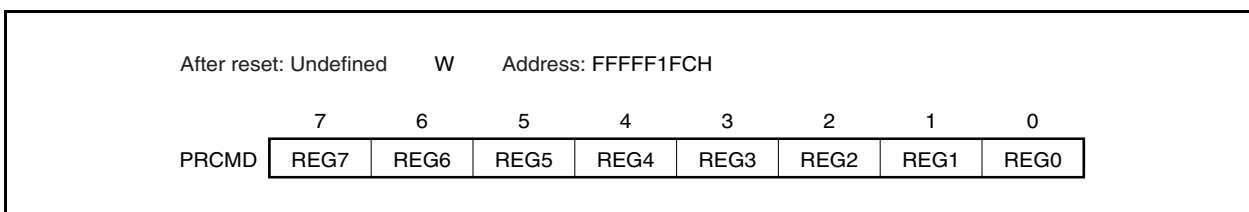
**(2) Command register (PRCMD)**

The PRCMD register is an 8-bit register that protects the registers that may seriously affect the application system from being written, so that the system does not inadvertently stop due to a program 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.

An illegal write operation to a special register can be checked by using the SYS.PRERR bit.

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

Reset makes this register undefined.



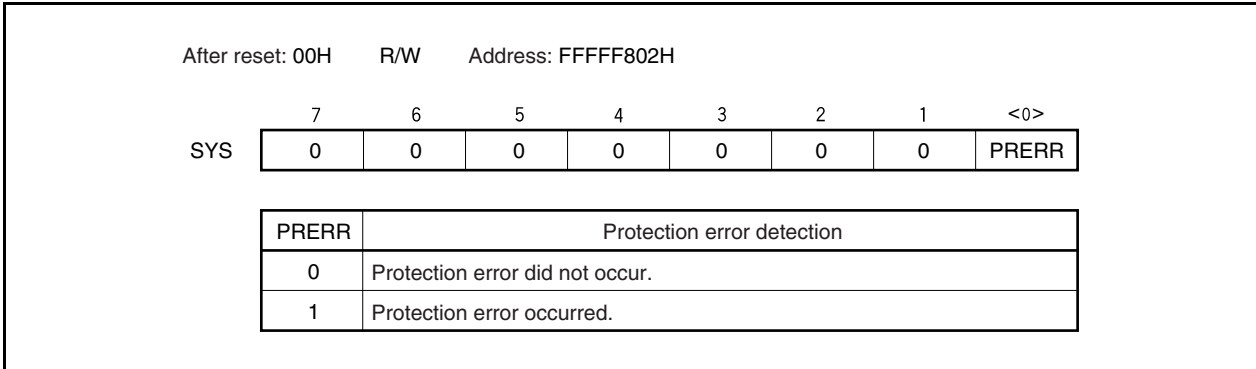
**(3) System status register (SYS)**

Status flags that indicate the operation status of the overall system are allocated to this register.

If this register is not written in the correct sequence including an access to the PRCMD register, data is not written to the intended register, a protection error occurs, and the PRERR flag is set. This register is cleared by writing “0” to it by an instruction from CPU.

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

Reset sets this register to 00H.



The PRERR flag operates under the following conditions.

**(a) Set condition (PRERR flag = 1)**

- When data is written to a special register without writing anything to the PRCMD register (when <3> is executed without executing <2> in **3.4.8 (1) Setting data to special registers**)
- 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 **3.4.8 (1) Setting data to special registers** is not the setting of a special register)

**Remark** Even if an on-chip peripheral I/O register is read (excluding execution of a bit manipulation instruction) between a write access to the PRCMD register and a write access to a special register (such as an access to the internal RAM), the PRERR flag is not set and data can be written to the special register.

**(b) Clear condition (PRERR flag = 0)**

- (i) When 0 is written to the SYS.PRERR flag
- (ii) When the system is reset

- Cautions**
1. If 0 is written to the SYS.PRERR bit 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.

### 3.4.9 System wait control register (VSWC)

The VSWC register is a register that controls the bus access wait for the on-chip peripheral I/O registers.

Access to on-chip peripheral I/O registers of the V850E1 CPU core is basically made in 3 clocks; however, in the V850E/IF3 and V850E/IG3, a wait set by the VSWC register is required in addition to those 3 clocks. Set 13H (set wait for 4 clocks) to VSWC.

This register can be read or written in 8-bit units (address: FFFFF06EH, initial value: 77H).

| CPU Clock Frequency ( $f_{CPU}$ )                  | VSWC Set Value |
|--|----------------|
| $500 \text{ kHz} \leq f_{CPU} \leq 64 \text{ MHz}$ | 13H            |

**Caution** When using the V850E/IF3 and V850E/IG3, the VSWC register must be set first. Set other registers if necessary after setting the VSWC register.

**Remark** When a register includes status flags that indicate the statuses of the on-chip peripheral functions (such as UAnSTR) or a register that indicates the count value of a timer (such as TAAncNT) is accessed, a register access retry operation takes place if the timing at which the flag and count value changes and the timing of the register access overlap. Consequently, access to the on-chip peripheral I/O register may take a long time.

## CHAPTER 4 PORT FUNCTIONS

### 4.1 Features

#### 4.1.1 V850E/IF3

- Input-only ports: 4  
I/O ports: 44
- Input data read/output data write is enabled in 1-bit units.
- On-chip pull-up resistor can be connected in 1-bit units (ports 0 to 4 and DL only).

However, an on-chip pull-up resistor can only be connected when the pins are in input mode in the port mode, or when the pins function as input pins in the alternate-function mode. Moreover, an on-chip pull-up resistor can be connected to the TOB0T1 to TOB0T3, TOB0B1 to TOB0B3, and TOA21 pins, these are output pins in the alternate-function mode, when these pins go into a high-impedance state due to the TOB0OFF, TOA2OFF, or TOB1OFF pin or software processing.

#### 4.1.2 V850E/IG3

- Input-only ports: 8  
I/O ports: 56
- Input data read/output data write is enabled in 1-bit units.
- On-chip pull-up resistor can be connected in 1-bit units (ports 0 to 4 and DL only).

However, an on-chip pull-up resistor can only be connected when the pins are in input mode in the port mode, or when the pins function as input pins in the alternate-function mode. Moreover, an on-chip pull-up resistor can be connected to the TOB0T1 to TOB0T3, TOB0B1 to TOB0B3, TOA21, TOB1T1 to TOB1T3, TOB1B1 to TOB1B3, and TOA31 pins, these are output pins in the alternate-function mode, when these pins go into a high-impedance state due to the TOB0OFF, TOB1OFF, TOA2OFF, or TOA3OFF pin or software processing.

## 4.2 Port Configuration

### 4.2.1 V850E/IF3

The V850E/IF3 incorporates a total of 48 input/output ports (including 4 input-only ports) labeled ports 0 to 4, 7, and DL. The port configuration is shown in Figure 4-1.

There are two power supply systems for the I/O buffer of a pin:  $AV_{DD2}$  and  $EV_{DD0}$ ,  $EV_{DD1}$ . The relationship between each of these power supplies and the pin is shown in Table 4-1.

Figure 4-1. Port Configuration (V850E/IF3)

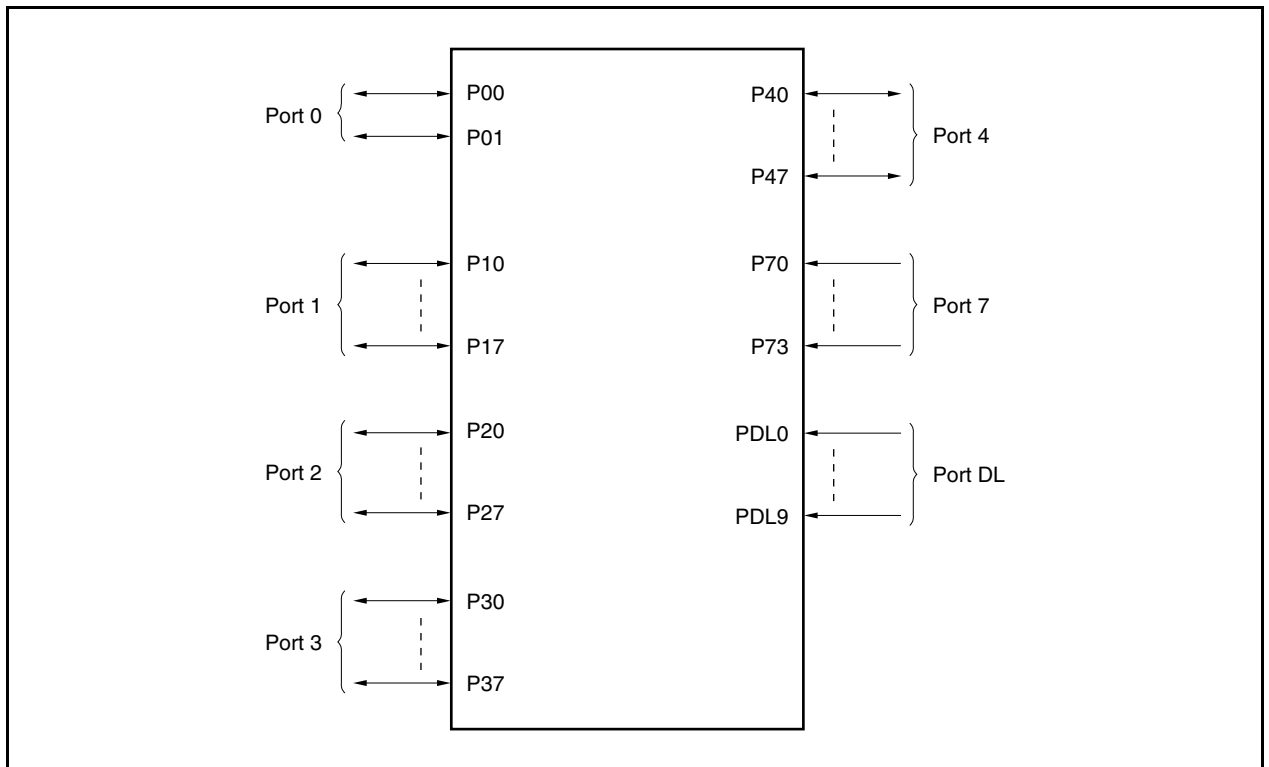


Table 4-1. Power Supplies for I/O Buffer of Each Pin (V850E/IF3)

| Power Supply            | Corresponding Pins   |
|-------------------------|--|
| $AV_{DD2}$              | P70 to P73   |
| $EV_{DD0}$ , $EV_{DD1}$ | P00, P01, P10 to P17, P20 to P27, P30 to P37, P40 to P47, PDL0 to PDL9, $\overline{RESET}$ |

4.2.2 V850E/IG3

The V850E/IG3 incorporates a total of 64 input/output ports (including 8 input-only ports) labeled ports 0 to 4, 7, and DL. The port configuration is shown in Figure 4-2.

There are two power supply systems for the I/O buffer of a pin:  $AV_{DD2}$  and  $EV_{DD0}$ ,  $EV_{DD1}$ ,  $EV_{DD2}$ . The relationship between each of these power supplies and the pin is shown in Table 4-2.

Figure 4-2. Port Configuration (V850E/IG3)

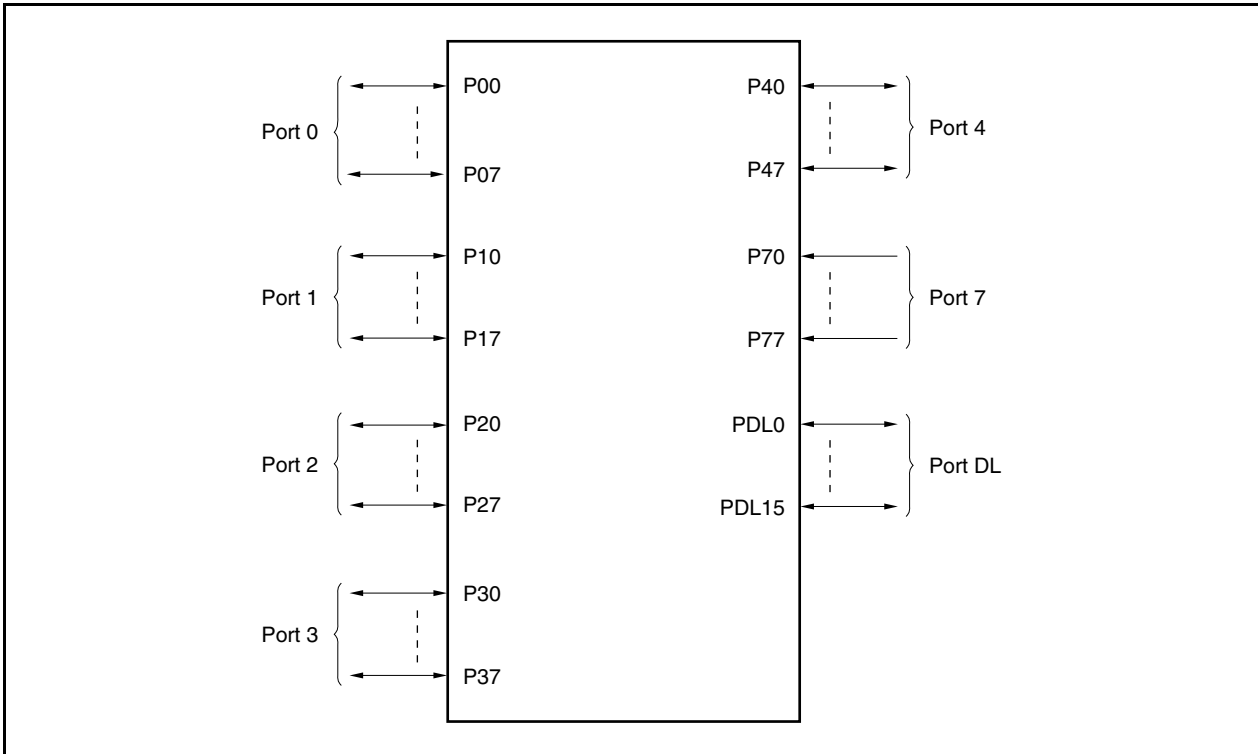


Table 4-2. Power Supplies for I/O Buffer of Each Pin (V850E/IG3)

| Power Supply                         | Corresponding Pins  |
|--------------------------------------|---|
| $AV_{DD2}$                           | P70 to P77  |
| $EV_{DD0}$ , $EV_{DD1}$ , $EV_{DD2}$ | P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, PDL0 to PDL15, $\overline{RESET}$ , DCK, DDI, DDO, DMS, $\overline{DRST}$ |

4.3 Port Configuration

**Table 4-3. Port Configuration (V850E/IF3)**

| Item              | Configuration  |
|-------------------|--|
| Control registers | Port n register (Pn: n = 0 to 4, 7, DL)<br>Port n mode register (PMn: n = 0 to 4, DL)<br>Port n mode control register (PMcn: n = 0 to 4, 7, DL)<br>Port n function control register (PFCn: n = 0 to 4)<br>Port n function control expansion register (PFCEn: n = 0 to 4)<br>Pull-up resistor option register (PUn: n = 0 to 4, DL)<br>Port 3 function register (PF3) |
| Ports             | Input-only: 4, I/O: 44   |
| Pull-up resistor  | Software control: 44   |

**Table 4-4. Port Configuration (V850E/IG3)**

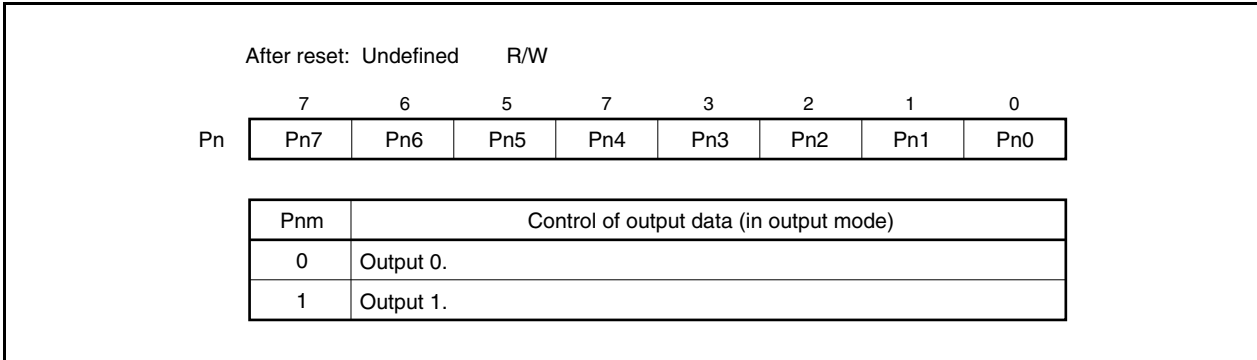
| Item              | Configuration  |
|-------------------|--|
| Control registers | Port n register (Pn: n = 0 to 4, 7, DL)<br>Port n mode register (PMn: n = 0 to 4, DL)<br>Port n mode control register (PMcn: n = 0 to 4, 7, DL)<br>Port n function control register (PFCn: n = 0 to 4)<br>Port n function control expansion register (PFCEn: n = 0 to 4)<br>Pull-up resistor option register (PUn: n = 0 to 4, DL)<br>Port 3 function register (PF3) |
| Ports             | Input-only: 8, I/O: 56   |
| Pull-up resistor  | Software control: 56   |

**(1) Port n register (Pn)**

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

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

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



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

**Table 4-5. Writing/Reading Pn Register**

| Setting of PMn Register   | Writing to Pn Register   | Reading from Pn Register                                  |
|---------------------------|--|---|
| Output mode<br>(PMnm = 0) | Data is written to the output latch <sup>Note 1</sup> .<br>In the port mode (PMcn = 0), the contents of the output latch are output from the pins. | The value of the output latch is read <sup>Note 2</sup> . |
| Input mode<br>(PMnm = 1)  | Data is written to the output latch.<br>The pin status is not affected <sup>Note 1</sup> .   | The pin status is read <sup>Note 3</sup> .                |

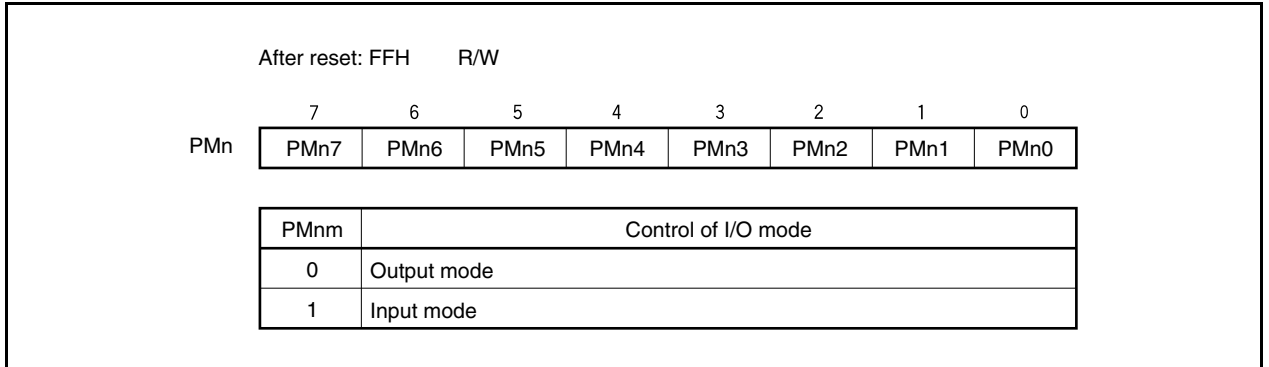
- Notes**
1. The value written to the output latch is retained until a new value is written to the output latch.
  2. Also, the value of the Pn register is read when the PMn register is in the output mode while the alternate function is set.
  3. If the PMn register is in the input mode while the alternate function is set, the statuses of the pins at that time are read regardless of whether the alternate function is an input or output function.



**(2) Port n mode register (PMn)**

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

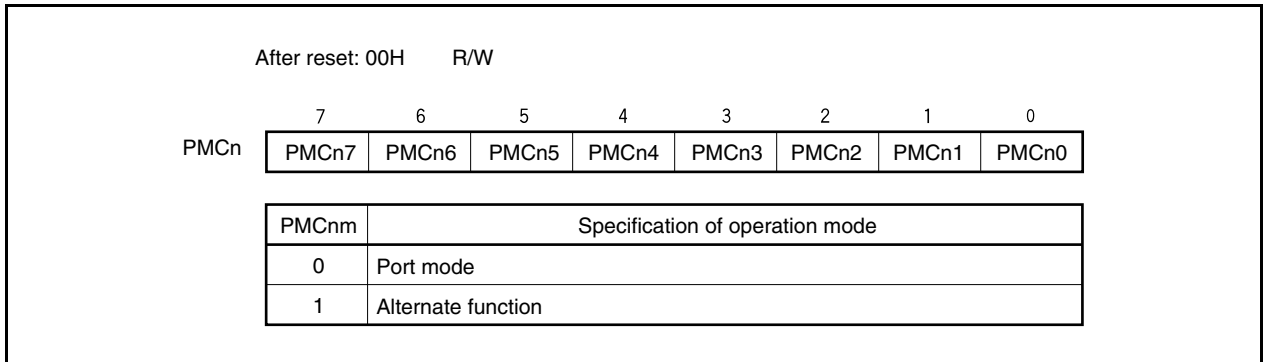
Each bit of this register corresponds to one pin of port n, and the input or output mode can be specified in 1-bit units.



**(3) Port n mode control register (PMCn)**

The PMCn register specifies the port mode or alternate function.

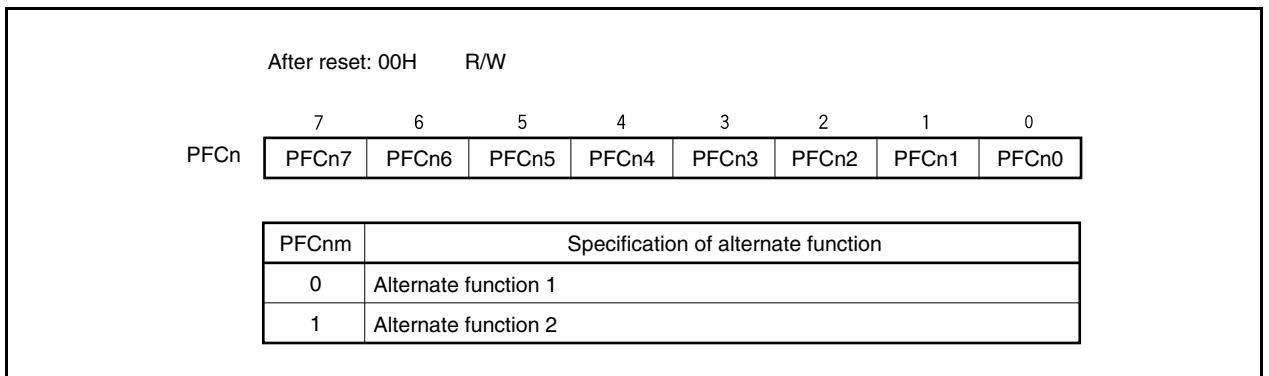
Each bit of this register corresponds to one pin of port n, and the mode of the port can be specified in 1-bit units.



**(4) Port n function control register (PFCn)**

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

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



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

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

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

After reset: 00H    R/W

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

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

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

**(6) Pull-up resistor option register (PUn)**

PUn is a register that specifies the connection of an on-chip pull-up resistor.

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

After reset: 00H    R/W

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

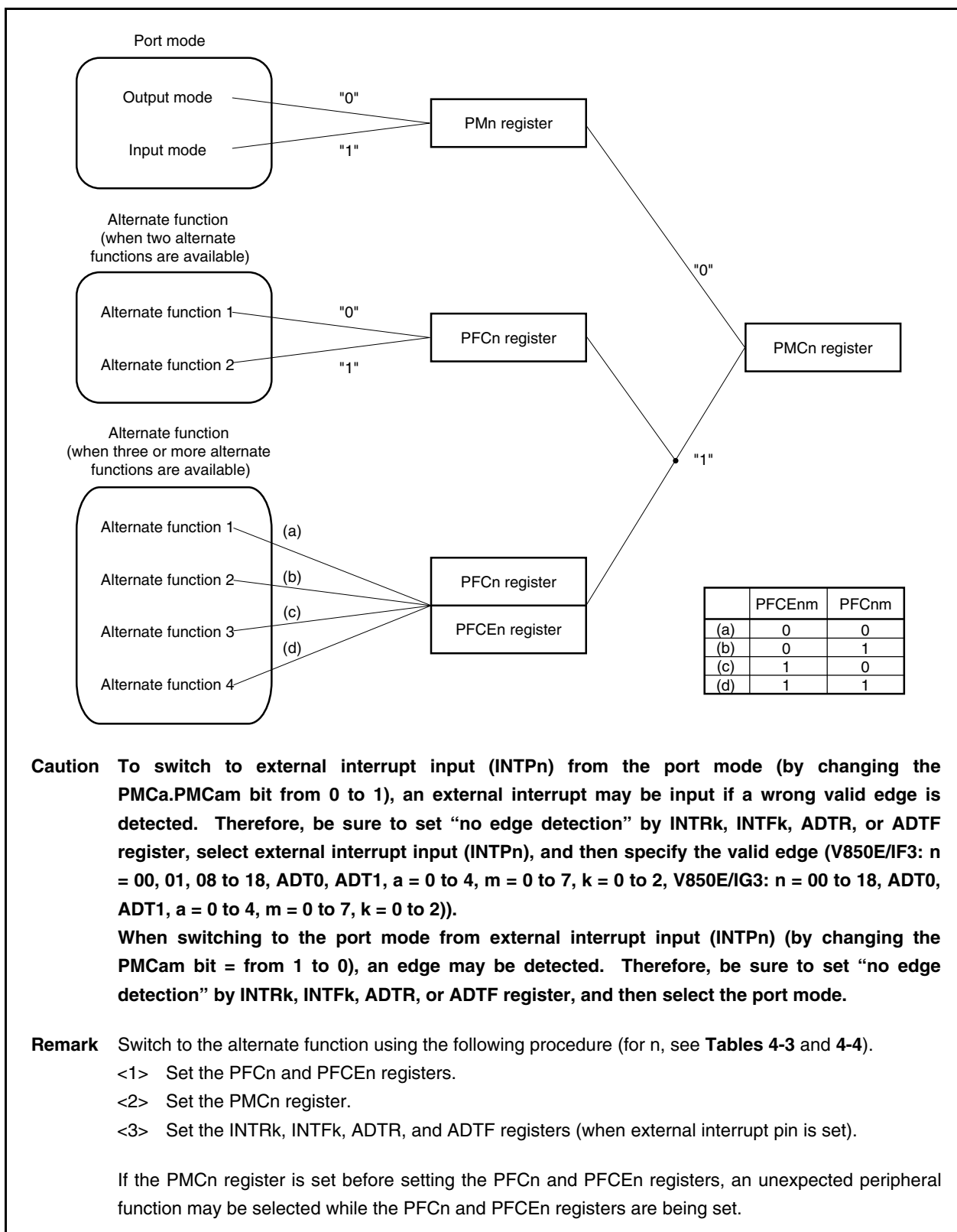
  

| PUnm | Control of on-chip pull-up resistor connection |
|------|--|
| 0    | Not connected                                  |
| 1    | Connected                                      |

(7) Port settings

Set the ports as follows.

Figure 4-3. Register Settings and Pin Functions



**Caution** To switch to external interrupt input (INTPn) from the port mode (by changing the PMCa.PMCam bit from 0 to 1), an external interrupt may be input if a wrong valid edge is detected. Therefore, be sure to set “no edge detection” by INTRk, INTFk, ADTR, or ADTF register, select external interrupt input (INTPn), and then specify the valid edge (V850E/IF3: n = 00, 01, 08 to 18, ADT0, ADT1, a = 0 to 4, m = 0 to 7, k = 0 to 2, V850E/IG3: n = 00 to 18, ADT0, ADT1, a = 0 to 4, m = 0 to 7, k = 0 to 2)).

When switching to the port mode from external interrupt input (INTPn) (by changing the PMCam bit = from 1 to 0), an edge may be detected. Therefore, be sure to set “no edge detection” by INTRk, INTFk, ADTR, or ADTF register, and then select the port mode.

**Remark** Switch to the alternate function using the following procedure (for n, see Tables 4-3 and 4-4).  
 <1> Set the PFCn and PFCEn registers.  
 <2> Set the PMCn register.  
 <3> Set the INTRk, INTFk, ADTR, and ADTF registers (when external interrupt pin is set).

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

### 4.3.1 Port 0

Port 0 can be set to the input or output mode in 1-bit units.  
The number of I/O pins differs from one product to another.

| Generic Name | Number of I/O Ports |
|--------------|---------------------|
| V850E/IF3    | 2-bit I/O port      |
| V850E/IG3    | 8-bit I/O port      |

Port 0 has an alternate function as the following pins.

**Table 4-6. Alternate-Function Pins of Port 0**

| Pin Name              | Pin No. |    |     |     | Alternate-Function Pin Name   | I/O   | Pull-Up <sup>Note 1</sup> |
|-----------------------|---------|----|-----|-----|---|-------|---------------------------|
|                       | IF3     |    | IG3 |     |   |       |                           |
|                       | GC      | GC | GF  | F1  |   |       |                           |
| P00                   | 70      | 91 | 19  | A6  | TOA20/TIA20/TOA2OFF/INTP00  | I/O   | Provided                  |
| P01                   | 69      | 90 | 18  | B6  | TOA21/TIA21/INTP01  | I/O   |                           |
| P02 <sup>Note 2</sup> | –       | 89 | 17  | C6  | TOA30 <sup>Note 2</sup> /TIA30 <sup>Note 2</sup> /TOA3OFF <sup>Note 2</sup> /INTP02 <sup>Note 2</sup> | I/O   |                           |
| P03 <sup>Note 2</sup> | –       | 88 | 16  | C7  | TOA31 <sup>Note 2</sup> /TIA31 <sup>Note 2</sup> /INTP03 <sup>Note 2</sup>                            | I/O   |                           |
| P04 <sup>Note 2</sup> | –       | 84 | 12  | C8  | TECR0 <sup>Note 2</sup> /TIT00 <sup>Note 2</sup> /TOT00 <sup>Note 2</sup> /INTP04 <sup>Note 2</sup>   | I/O   |                           |
| P05 <sup>Note 2</sup> | –       | 83 | 11  | C9  | TENC00 <sup>Note 2</sup> /EVTT0 <sup>Note 2</sup> /INTP05 <sup>Note 2</sup>                           | Input |                           |
| P06 <sup>Note 2</sup> | –       | 82 | 10  | C10 | TENC01 <sup>Note 2</sup> /TIT01 <sup>Note 2</sup> /TOT01 <sup>Note 2</sup> /INTP06 <sup>Note 2</sup>  | I/O   |                           |
| P07 <sup>Note 2</sup> | –       | 63 | 91  | C12 | INTP07 <sup>Note 2</sup> /CLKOUT <sup>Note 3</sup>  | I/O   |                           |

**Remark** IF3: V850E/IF3  
IG3: V850E/IG3  
GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)  
GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)  
GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)  
F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

**Notes** 1. Software pull-up function  
2. V850E/IG3 only  
3.  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**Cautions** 1. To control the high-impedance output of a timer for motor control, be sure to set the PMC0.PMC0n bit to 1 and then specify the edge to be detected and enable the operation of the high-impedance output controller, because the output of the motor control timer may go into a high-impedance state if a wrong valid edge is detected (V850E/IF3: n = 0, V850E/IG3: n = 0, 2).  
2. When P01 and P03 (V850E/IG3 only) are used as TOA21 and TOA31 (V850E/IG3 only), they go into a high-impedance state by inputting the following active signal.

- Output of high impedance setting signal from high impedance output controller
- Output of clock stop detection signal from clock monitor

**Cautions 3.** To switch to external interrupt input (INTP0n) from the port mode (by changing the PMC0.PMC0n bit from 0 to 1), an external interrupt may be input if a wrong valid edge is detected. Therefore, be sure to disable edge detection (INTF0.INTF0n bit = 0 and INTR0.INTR0n bit = 0), select external interrupt input (INTP0n), and then specify the valid edge (V850E/IF3: n = 0, 1, V850E/IG3: n = 0 to 7).

When switching to the port mode from external interrupt input (INTP0n) (by changing the PMC0n bit from 1 to 0), an edge may be detected. Therefore, be sure to disable edge detection (INTF0n bit = 0, INTR0n bit = 0), and then select the port mode.

**4.** To control high-impedance output of the external interrupt function and motor output control function, set the PMC0n bit to 1 (V850E/IF3: n = 0, 1, V850E/IG3: n = 0 to 7).

(1) Registers

(a) Port 0 register (P0)

After reset: Undefined R/W Address: FFFFF400H

|    |                     |                     |                     |                     |                     |                     |     |     |
|----|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|-----|-----|
|    | 7                   | 6                   | 5                   | 4                   | 3                   | 2                   | 1   | 0   |
| P0 | P07 <sup>Note</sup> | P06 <sup>Note</sup> | P05 <sup>Note</sup> | P04 <sup>Note</sup> | P03 <sup>Note</sup> | P02 <sup>Note</sup> | P01 | P00 |

|     |   |
|-----|---|
| P0n | Control of output data (in output mode) |
| 0   | Output 0.                               |
| 1   | Output 1.                               |

**Note** Valid only for the V850E/IG3.  
With the V850E/IF3, the read value of this register is undefined.

**Remark** V850E/IF3: n = 0, 1  
V850E/IG3: n = 0 to 7

(b) Port 0 mode register (PM0)

After reset: FFH R/W Address: FFFFF420H

|     |                      |                      |                      |                      |                      |                      |      |      |
|-----|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|------|------|
|     | 7                    | 6                    | 5                    | 4                    | 3                    | 2                    | 1    | 0    |
| PM0 | PM07 <sup>Note</sup> | PM06 <sup>Note</sup> | PM05 <sup>Note</sup> | PM04 <sup>Note</sup> | PM03 <sup>Note</sup> | PM02 <sup>Note</sup> | PM01 | PM00 |

|      |                                    |
|------|------------------------------------|
| PM0n | Control of I/O mode (in port mode) |
| 0    | Output mode                        |
| 1    | Input mode                         |

**Note** Valid only for the V850E/IG3.  
With the V850E/IF3, be sure to set this bit to 1.

**Remark** V850E/IF3: n = 0, 1  
V850E/IG3: n = 0 to 7

(c) Port 0 mode control register (PMC0)

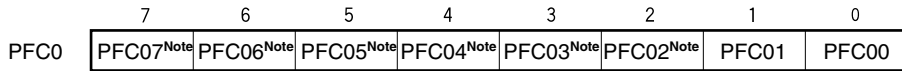
After reset: 00H R/W Address: FFFFF440H

|                         | 7  | 6                       | 5                       | 4                       | 3                       | 2                       | 1     | 0     |
|-------------------------|--|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------|-------|
| PMC0                    | PMC07 <sup>Note 1</sup>  | PMC06 <sup>Note 1</sup> | PMC05 <sup>Note 1</sup> | PMC04 <sup>Note 1</sup> | PMC03 <sup>Note 1</sup> | PMC02 <sup>Note 1</sup> | PMC01 | PMC00 |
| PMC07 <sup>Note 1</sup> | Specification of operating mode of P07 pin   |                         |                         |                         |                         |                         |       |       |
| 0                       | I/O port   |                         |                         |                         |                         |                         |       |       |
| 1                       | INTP07 input <sup>Note 2</sup> /CLKOUT output <sup>Note 3</sup>  |                         |                         |                         |                         |                         |       |       |
| PMC06 <sup>Note 1</sup> | Specification of operating mode of P06 pin   |                         |                         |                         |                         |                         |       |       |
| 0                       | I/O port   |                         |                         |                         |                         |                         |       |       |
| 1                       | TENC01 input <sup>Note 2</sup> /TIT01 input <sup>Note 2</sup> /TOT01 output <sup>Note 2</sup> /INTP06 input <sup>Note 1</sup>  |                         |                         |                         |                         |                         |       |       |
| PMC05 <sup>Note 1</sup> | Specification of operating mode of P05 pin   |                         |                         |                         |                         |                         |       |       |
| 0                       | I/O port   |                         |                         |                         |                         |                         |       |       |
| 1                       | TENC00 input <sup>Note 2</sup> /EVTT0 input <sup>Note 2</sup> /INTP05 input <sup>Note 2</sup>                                  |                         |                         |                         |                         |                         |       |       |
| PMC04 <sup>Note 1</sup> | Specification of operating mode of P04 pin   |                         |                         |                         |                         |                         |       |       |
| 0                       | I/O port   |                         |                         |                         |                         |                         |       |       |
| 1                       | TECR0 input <sup>Note 2</sup> /TIT00 input <sup>Note 2</sup> /TOT00 output <sup>Note 2</sup> /INTP04 input <sup>Note 2</sup>   |                         |                         |                         |                         |                         |       |       |
| PMC03 <sup>Note 1</sup> | Specification of operating mode of P03 pin   |                         |                         |                         |                         |                         |       |       |
| 0                       | I/O port   |                         |                         |                         |                         |                         |       |       |
| 1                       | TOA31 output <sup>Note 2</sup> /TIA31 input <sup>Note 2</sup> /INTP03 input <sup>Note 2</sup>                                  |                         |                         |                         |                         |                         |       |       |
| PMC02 <sup>Note 1</sup> | Specification of operating mode of P02 pin   |                         |                         |                         |                         |                         |       |       |
| 0                       | I/O port   |                         |                         |                         |                         |                         |       |       |
| 1                       | TOA30 output <sup>Note 2</sup> /TIA30 input <sup>Note 2</sup> /TOA3OFF input <sup>Note 2</sup> /INTP02 input <sup>Note 2</sup> |                         |                         |                         |                         |                         |       |       |
| PMC01                   | Specification of operating mode of P01 pin   |                         |                         |                         |                         |                         |       |       |
| 0                       | I/O port   |                         |                         |                         |                         |                         |       |       |
| 1                       | TOA21 output/TIA21 input/INTP01 input  |                         |                         |                         |                         |                         |       |       |
| PMC00                   | Specification of operating mode of P00 pin   |                         |                         |                         |                         |                         |       |       |
| 0                       | I/O port   |                         |                         |                         |                         |                         |       |       |
| 1                       | TOA20 output/TIA20 input/TOA2OFF input/INTP00 input  |                         |                         |                         |                         |                         |       |       |

- Notes**
- Valid only in the V850E/IG3.  
With the V850E/IF3, be sure to set these bits to 0.
  - V850E/IG3 only
  - $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**(d) Port 0 function control register (PFC0)**

After reset: 00H    R/W    Address: FFFFF460H



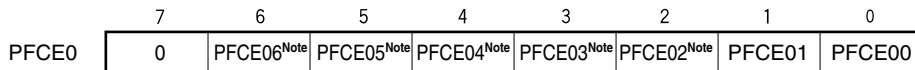
**Note** Valid only in the V850E/IG3.

With the V850E/IF3, be sure to set these bits to 0.

**Remark** For the specifications of alternate functions, see **4.3.1 (1) (f) Settings of alternate functions of port 0.**

**(e) Port 0 function control expansion register (PFCE0)**

After reset: 00H    R/W    Address: FFFFF700H



**Note** Valid only in the V850E/IG3.

With the V850E/IF3, be sure to set these bits to 0.

**Remark** For the specifications of alternate functions, see **4.3.1 (1) (f) Settings of alternate functions of port 0.**



(f) Setting of alternate function of port 0

| PFC07 <sup>Note 1</sup> | Specification of Alternate Function of P07 <sup>Note 1</sup> Pin |
|-------------------------|--|
| 0                       | INTP07 input <sup>Note 1</sup>                                   |
| 1                       | CLKOUT output <sup>Note 2</sup>                                  |

| PFCE06 <sup>Note 1</sup> | PFC06 <sup>Note 1</sup> | Specification of Alternate Function of P06 <sup>Note 1</sup> Pin                                   |
|--------------------------|-------------------------|--|
| 0                        | 0                       | TENC01 input <sup>Note 1</sup> /TIT01 input <sup>Note 1</sup> (two functions are alternately used) |
| 0                        | 1                       | TOT01 output <sup>Note 1</sup>   |
| 1                        | 0                       | INTP06 input <sup>Note 1</sup>   |
| 1                        | 1                       | Setting prohibited   |

| PFCE05 <sup>Note 1</sup> | PFC05 <sup>Note 1</sup> | Specification of Alternate Function of P05 <sup>Note 1</sup> Pin |
|--------------------------|-------------------------|--|
| 0                        | 0                       | TENC00 input <sup>Note 1</sup>                                   |
| 0                        | 1                       | EVT0 input <sup>Note 1</sup>                                     |
| 1                        | 0                       | INTP05 input <sup>Note 1</sup>                                   |
| 1                        | 1                       | Setting prohibited   |

| PFCE04 <sup>Note 1</sup> | PFC04 <sup>Note 1</sup> | Specification of Alternate Function of P04 <sup>Note 1</sup> Pin                                  |
|--------------------------|-------------------------|---|
| 0                        | 0                       | TECR0 input <sup>Note 1</sup> /TIT00 input <sup>Note 1</sup> (two functions are alternately used) |
| 0                        | 1                       | TOT00 output <sup>Note 1</sup>  |
| 1                        | 0                       | INTP04 input <sup>Note 1</sup>  |
| 1                        | 1                       | Setting prohibited  |

| PFCE03 <sup>Note 1</sup> | PFC03 <sup>Note 1</sup> | Specification of Alternate Function of P03 <sup>Note 1</sup> Pin |
|--------------------------|-------------------------|--|
| 0                        | 0                       | TOA31 output <sup>Note 1</sup>                                   |
| 0                        | 1                       | TIA31 input <sup>Note 1</sup>                                    |
| 1                        | 0                       | INTP03 input <sup>Note 1</sup>                                   |
| 1                        | 1                       | Setting prohibited   |

| PFCE02 <sup>Note 1</sup> | PFC02 <sup>Note 1</sup> | Specification of Alternate Function of P02 <sup>Note 1</sup> Pin                                      |
|--------------------------|-------------------------|---|
| 0                        | 0                       | TOA30 output <sup>Note 1</sup>  |
| 0                        | 1                       | TIA30 input <sup>Note 1</sup>   |
| 1                        | 0                       | TOA30OFF input <sup>Note 1</sup> /INTP02 input <sup>Note 1</sup> (two functions are alternately used) |
| 1                        | 1                       | Setting prohibited  |

Notes 1. V850E/IG3 only

2.  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

| PFCE01 | PFC01 | Specification of Alternate Function of P01 Pin |
|--------|-------|--|
| 0      | 0     | TOA21 output                                   |
| 0      | 1     | TIA21 input                                    |
| 1      | 0     | INTP01 input                                   |
| 1      | 1     | Setting prohibited                             |

| PFCE00 | PFC00 | Specification of Alternate Function of P00 Pin |
|--------|-------|--|
| 0      | 0     | TOA20 output                                   |
| 0      | 1     | TIA20 input                                    |
| 1      | 0     | TOA2OFF input/INTP00 input                     |
| 1      | 1     | Setting prohibited                             |

**(g) Pull-up resistor option register 0 (PU0)**

After reset: 00H R/W Address: FFFFC40H

|     |                        |                        |                        |                        |                        |                        |      |      |
|-----|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------|------|
|     | 7                      | 6                      | 5                      | 4                      | 3                      | 2                      | 1    | 0    |
| PU0 | PU07 <sup>Note 1</sup> | PU06 <sup>Note 1</sup> | PU05 <sup>Note 1</sup> | PU04 <sup>Note 1</sup> | PU03 <sup>Note 1</sup> | PU02 <sup>Note 1</sup> | PU01 | PU00 |

| PU0n | Control of on-chip pull-up resistor connection |
|------|--|
| 0    | Do not connect                                 |
| 1    | Connect <sup>Note 2</sup>                      |

**Notes 1.** Valid only in the V850E/IG3.

With the V850E/IF3, be sure to set this bit to 0.

- 2.** An on-chip pull-up resistor can be connected only when the pins are in input mode in the port mode or when the pins function as input pins in the alternate-function mode. Moreover, an on-chip pull-up resistor can be connected to the TOA21 and TOA31 (V850E/IG3 only) pins, these are output pins in the alternate-function mode, when these pins go into a high-impedance state due to the TOA2OFF or TOA3OFF (V850E/IG3 only) pin, or software processing. An on-chip pull-up resistor cannot be connected when the pins are in output mode.

**Remark** V850E/IF3: n = 0, 1  
V850E/IG3: n = 0 to 7

4.3.2 Port 1

Port 1 can be set to the input or output mode in 1-bit units.

Port 1 has an alternate function as the following pins.

Table 4-7. Alternate-Function Pins of Port 1

| Pin Name | Pin No. |     |    |    | Alternate-Function Pin Name                        | I/O | Pull-Up <sup>Note 1</sup> |
|----------|---------|-----|----|----|--|-----|---------------------------|
|          | IF3     | IG3 |    |    |  |     |                           |
|          | GC      | GC  | GF | F1 |  |     |                           |
| P10      | 78      | 99  | 27 | B3 | TOB0T1/TIB01/TOB01/A0 <sup>Note 2</sup>            | I/O | Provided                  |
| P11      | 77      | 98  | 26 | C3 | TOB0B1/TIB02/TOB02/A1 <sup>Note 2</sup>            | I/O |                           |
| P12      | 76      | 97  | 25 | A4 | TOB0T2/TIB03/TOB03/A2 <sup>Note 2</sup>            | I/O |                           |
| P13      | 75      | 96  | 24 | B4 | TOB0B2/TIB00/A3 <sup>Note 2</sup>                  | I/O |                           |
| P14      | 74      | 95  | 23 | C4 | TOB0T3/EVTB0/A4 <sup>Note 2</sup>                  | I/O |                           |
| P15      | 73      | 94  | 22 | A5 | TOB0B3/TRGB0/A5 <sup>Note 2</sup>                  | I/O |                           |
| P16      | 72      | 93  | 21 | B5 | TOB0OFF/INTP08/ADTRG0/INTADT0/A6 <sup>Note 2</sup> | I/O |                           |
| P17      | 71      | 92  | 20 | C5 | TOB00/INTP09/A7 <sup>Note 2</sup>                  | I/O |                           |

- Notes**
1. Software pull-up function
  2.  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**Caution** When P10 to P15 are used as TOB0T1 to TOB0T3 and TOB0B1 to TOB0B3, they go into a high-impedance state by inputting the following active signal.

- Output of high impedance setting signal from high impedance output controller
- Output of clock stop detection signal from clock monitor

**Remark**

IF3: V850E/IF3  
 IG3: V850E/IG3  
 GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)  
 GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)  
 GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)  
 F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

(1) Registers

(a) Port 1 register (P1)

After reset: Undefined    R/W    Address: FFFFF402H

|    |     |     |     |     |     |     |     |     |
|----|-----|-----|-----|-----|-----|-----|-----|-----|
|    | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| P1 | P17 | P16 | P15 | P14 | P13 | P12 | P11 | P10 |

|     |   |
|-----|---|
| P1n | Control of output data (in output mode) |
| 0   | Output 0.                               |
| 1   | Output 1.                               |

**Remark**    n = 0 to 7

(b) Port 1 mode register (PM1)

After reset: FFH    R/W    Address: FFFFF422H

|     |      |      |      |      |      |      |      |      |
|-----|------|------|------|------|------|------|------|------|
|     | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| PM1 | PM17 | PM16 | PM15 | PM14 | PM13 | PM12 | PM11 | PM10 |

|      |                                    |
|------|------------------------------------|
| PM1n | Control of I/O mode (in port mode) |
| 0    | Output mode                        |
| 1    | Input mode                         |

**Remark**    n = 0 to 7

(c) Port 1 mode control register (PMC1)

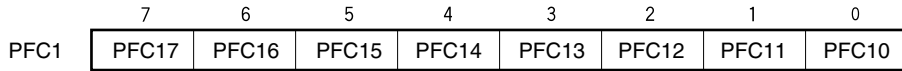
After reset: 00H R/W Address: FFFF442H

|      | 7  | 6   | 5     | 4     | 3     | 2     | 1     | 0     |
|------|--|---|-------|-------|-------|-------|-------|-------|
| PMC1 | PMC17                                      | PMC16   | PMC15 | PMC14 | PMC13 | PMC12 | PMC11 | PMC10 |
|      | Specification of operating mode of P17 pin |   |       |       |       |       |       |       |
|      | 0  | I/O port  |       |       |       |       |       |       |
|      | 1  | TOB00 output/INTP09 input/A7 output <sup>Note</sup>                             |       |       |       |       |       |       |
|      | Specification of operating mode of P16 pin |   |       |       |       |       |       |       |
|      | 0  | I/O port  |       |       |       |       |       |       |
|      | 1  | TOB00FF input/INTP08 input/ADTRG0 input/INTADT0 input/A6 output <sup>Note</sup> |       |       |       |       |       |       |
|      | Specification of operating mode of P15 pin |   |       |       |       |       |       |       |
|      | 0  | I/O port  |       |       |       |       |       |       |
|      | 1  | TOB0B3 output/TRGB0 input/A5 output <sup>Note</sup>                             |       |       |       |       |       |       |
|      | Specification of operating mode of P14 pin |   |       |       |       |       |       |       |
|      | 0  | I/O port  |       |       |       |       |       |       |
|      | 1  | TOB0T3 output/EVTB0 input/A4 output <sup>Note</sup>                             |       |       |       |       |       |       |
|      | Specification of operating mode of P13 pin |   |       |       |       |       |       |       |
|      | 0  | I/O port  |       |       |       |       |       |       |
|      | 1  | TOB0B2 output/TIB00 input/A3 output <sup>Note</sup>                             |       |       |       |       |       |       |
|      | Specification of operating mode of P12 pin |   |       |       |       |       |       |       |
|      | 0  | I/O port  |       |       |       |       |       |       |
|      | 1  | TOB0T2 output/TIB03 input/TOB03 output/A2 output <sup>Note</sup>                |       |       |       |       |       |       |
|      | Specification of operating mode of P11 pin |   |       |       |       |       |       |       |
|      | 0  | I/O port  |       |       |       |       |       |       |
|      | 1  | TOB0B1 output/TIB02 input/TOB02 output/A1 output <sup>Note</sup>                |       |       |       |       |       |       |
|      | Specification of operating mode of P10 pin |   |       |       |       |       |       |       |
|      | 0  | I/O port  |       |       |       |       |       |       |
|      | 1  | TOB0T1 output/TIB01 input/TOB01 output/A0 output <sup>Note</sup>                |       |       |       |       |       |       |

**Note**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**(d) Port 1 function control register (PFC1)**

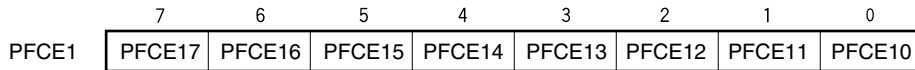
After reset: 00H    R/W    Address: FFFFF462H



**Remark** For the specifications of alternate functions, see **4.3.2 (1) (f) Settings of alternate functions of port 1.**

**(e) Port 1 function control expansion register (PFCE1)**

After reset: 00H    R/W    Address: FFFFF702H



**Remark** For the specifications of alternate functions, see **4.3.2 (1) (f) Settings of alternate functions of port 1.**

(f) Settings of alternate functions of port 1

| PFCE17 | PFC17 | Specification of Alternate Function of P17 Pin |
|--------|-------|--|
| 0      | 0     | TOB00 output                                   |
| 0      | 1     | INTP09 input                                   |
| 1      | 0     | A7 output <sup>Note</sup>                      |
| 1      | 1     | Setting prohibited                             |

| PFCE16 | PFC16 | Specification of Alternate Function of P16 Pin                  |
|--------|-------|---|
| 0      | 0     | TOB0OFF input/INTP08 input (two functions are alternately used) |
| 0      | 1     | ADTRG0 input/INTADT0 input (two functions are alternately used) |
| 1      | 0     | A6 output <sup>Note</sup>                                       |
| 1      | 1     | Setting prohibited  |

| PFCE15 | PFC15 | Specification of Alternate Function of P15 Pin |
|--------|-------|--|
| 0      | 0     | TOB0B3 output                                  |
| 0      | 1     | TRGB0 input                                    |
| 1      | 0     | A5 output <sup>Note</sup>                      |
| 1      | 1     | Setting prohibited                             |

| PFCE14 | PFC14 | Specification of Alternate Function of P14 Pin |
|--------|-------|--|
| 0      | 0     | TOB0T3 output                                  |
| 0      | 1     | EVTB0 input                                    |
| 1      | 0     | A4 output <sup>Note</sup>                      |
| 1      | 1     | Setting prohibited                             |

| PFCE13 | PFC13 | Specification of Alternate Function of P13 Pin |
|--------|-------|--|
| 0      | 0     | TOB0B2 output                                  |
| 0      | 1     | TIB00 input                                    |
| 1      | 0     | A3 output <sup>Note</sup>                      |
| 1      | 1     | Setting prohibited                             |

| PFCE12 | PFC12 | Specification of Alternate Function of P12 Pin |
|--------|-------|--|
| 0      | 0     | TOB0T2 output                                  |
| 0      | 1     | TIB03 input                                    |
| 1      | 0     | TOB03 output                                   |
| 1      | 1     | A2 output <sup>Note</sup>                      |

**Note**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

| PFCE11 | PFC11 | Specification of Alternate Function of P11 Pin |
|--------|-------|--|
| 0      | 0     | TOB0B1 output                                  |
| 0      | 1     | TIB02 input                                    |
| 1      | 0     | TOB02 output                                   |
| 1      | 1     | A1 output <sup>Note</sup>                      |

| PFCE10 | PFC10 | Specification of Alternate Function of P10 Pin |
|--------|-------|--|
| 0      | 0     | TOB0T1 output                                  |
| 0      | 1     | TIB01 input                                    |
| 1      | 0     | TOB01 output                                   |
| 1      | 1     | A0 output <sup>Note</sup>                      |

**Note**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**(g) Pull-up resistor option register 1 (PU1)**

After reset: 00H R/W Address: FFFFC42H

|     |      |      |      |      |      |      |      |      |
|-----|------|------|------|------|------|------|------|------|
|     | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| PU1 | PU17 | PU16 | PU15 | PU14 | PU13 | PU12 | PU11 | PU10 |

| PU1n | Control of on-chip pull-up resistor connection |
|------|--|
| 0    | Do not connect                                 |
| 1    | Connect <sup>Note</sup>                        |

**Note** An on-chip pull-up resistor can be connected only when the pins are in input mode in the port mode or when the pins function as input pins in the alternate-function mode. Moreover, an on-chip pull-up resistor can be connected to the TOB0T1 to TOB0T3 and TOB0B1 to TOB0B3 pins, these are output pins in the alternate-function mode, when these pins go into a high-impedance state due to the TOB0OFF pin, or software processing. An on-chip pull-up resistor cannot be connected when the pins are in output mode.

**Remark** n = 0 to 7



4.3.3 Port 2

Port 2 can be set to the input or output mode in 1-bit units.

Port 2 has an alternate function as the following pins.

Table 4-8. Alternate-Function Pins of Port 2

| Pin Name | Pin No. |     |    |     | Alternate-Function Pin Name         | I/O   | Pull-Up <sup>Note 1</sup> |
|----------|---------|-----|----|-----|-------------------------------------|-------|---------------------------|
|          | IF3     | IG3 |    |     |                                     |       |                           |
|          | GC      | GC  | GF | F1  |                                     |       |                           |
| P20      | 23      | 28  | 56 | P3  | TOB1T1/TIB11/TOB11                  | I/O   | Provided                  |
| P21      | 24      | 29  | 57 | N3  | TOB1B1/TIB12/TOB12                  | I/O   |                           |
| P22      | 25      | 30  | 58 | P4  | TOB1T2/TIB13/TOB13                  | I/O   |                           |
| P23      | 26      | 31  | 59 | N4  | TOB1B2/TIB10                        | I/O   |                           |
| P24      | 27      | 32  | 60 | M4  | TOB1T3/EVTB1                        | I/O   |                           |
| P25      | 28      | 33  | 61 | M5  | TOB1B3/TRGB1                        | I/O   |                           |
| P26      | 29      | 34  | 62 | M6  | TOB10/TOB1OFF/INTP10/ADTRG1/INTADT1 | I/O   |                           |
| P27      | 36      | 45  | 73 | M11 | DMS <sup>Notes 2, 3</sup>           | Input |                           |

- Notes**
1. Software pull-up function
  2. V850E/IG3 only
  3. The P27 pin also functions as an on-chip debug pin. The on-chip debug function or port function (including the alternate functions) can be selected by using the level of the  $\overline{DRST}$  pin, as shown in the table below.

| Port 2 Functions                         |   |
|--|---|
| Low-Level Input to $\overline{DRST}$ Pin | High-Level Input to $\overline{DRST}$ Pin |
| P27                                      | DMS                                       |

**Caution** When P20 to P25 are used as TOB1T1 to TOB1T3 and TOB1B1 to TOB1B3, they go into a high-impedance state by inputting the following active signal.

- Output of high impedance setting signal from high impedance output controller
- Output of clock stop detection signal from clock monitor

**Remark**

IF3: V850E/IF3  
 IG3: V850E/IG3  
 GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)  
 GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)  
 GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)  
 F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

(1) Registers

(a) Port 2 register (P2)

After reset: Undefined    R/W    Address: FFFFF404H

|    |     |     |     |     |     |     |     |     |
|----|-----|-----|-----|-----|-----|-----|-----|-----|
|    | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| P2 | P27 | P26 | P25 | P24 | P23 | P22 | P21 | P20 |

|     |   |
|-----|---|
| P2n | Control of output data (in output mode) |
| 0   | Output 0.                               |
| 1   | Output 1.                               |

**Remark**    n = 0 to 7

(b) Port 2 mode register (PM2)

After reset: FFH    R/W    Address: FFFFF424H

|     |      |      |      |      |      |      |      |      |
|-----|------|------|------|------|------|------|------|------|
|     | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| PM2 | PM27 | PM26 | PM25 | PM24 | PM23 | PM22 | PM21 | PM20 |

|      |                                    |
|------|------------------------------------|
| PM2n | Control of I/O mode (in port mode) |
| 0    | Output mode                        |
| 1    | Input mode                         |

**Remark**    n = 0 to 7

(c) Port 2 mode control register (PMC2)

After reset: 00H    R/W    Address: FFFFF444H

|      |   |       |       |       |       |       |       |       |
|------|---|-------|-------|-------|-------|-------|-------|-------|
|      | 7 | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| PMC2 | 0 | PMC26 | PMC25 | PMC24 | PMC23 | PMC22 | PMC21 | PMC20 |

|       |  |
|-------|--|
| PMC26 | Specification of operating mode of P26 pin                         |
| 0     | I/O port   |
| 1     | TOB10 output/TOB1OFF input/INTP10 input/ADTRG1 input/INTADT1 input |
| PMC25 | Specification of operating mode of P25 pin                         |
| 0     | I/O port   |
| 1     | TOB1B3 output/TRGB1 input  |
| PMC24 | Specification of operating mode of P24 pin                         |
| 0     | I/O port   |
| 1     | TOB1T3 output/EVTB1 input  |
| PMC23 | Specification of operating mode of P23 pin                         |
| 0     | I/O port   |
| 1     | TOB1B2 output/TIB10 input  |
| PMC22 | Specification of operating mode of P22 pin                         |
| 0     | I/O port   |
| 1     | TOB1T2 output/TIB13 input/TOB13 output                             |
| PMC21 | Specification of operating mode of P21 pin                         |
| 0     | I/O port   |
| 1     | TOB1B1 output/TIB12 input/TOB12 output                             |
| PMC20 | Specification of operating mode of P20 pin                         |
| 0     | I/O port   |
| 1     | TOB1T1 output/TIB11 input/TOB11 output                             |

**(d) Port 2 function control register (PFC2)**

After reset: 00H    R/W    Address: FFFFF464H

|      |   |       |       |       |       |       |       |       |
|------|---|-------|-------|-------|-------|-------|-------|-------|
|      | 7 | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| PFC2 | 0 | PFC26 | PFC25 | PFC24 | PFC23 | PFC22 | PFC21 | PFC20 |

**Remark** For the specifications of alternate functions, see **4.3.3 (1) (f) Settings of alternate functions of port 2.**

**(e) Port 2 function control expansion register (PFCE2)**

After reset: 00H    R/W    Address: FFFFF704H

|       |   |        |   |   |   |        |        |        |
|-------|---|--------|---|---|---|--------|--------|--------|
|       | 7 | 6      | 5 | 4 | 3 | 2      | 1      | 0      |
| PFCE2 | 0 | PFCE26 | 0 | 0 | 0 | PFCE22 | PFCE21 | PFCE20 |

**Remark** For the specifications of alternate functions, see **4.3.3 (1) (f) Settings of alternate functions of port 2.**

(f) Settings of alternate functions of port 2

| PFCE26 | PFC26 | Specification of Alternate Function of P26 Pin                  |
|--------|-------|---|
| 0      | 0     | TOB10 output  |
| 0      | 1     | TOB1OFF input/INTP10 input (two functions are alternately used) |
| 1      | 0     | ADTRG1 input/INTADT1 input (two functions are alternately used) |
| 1      | 1     | Setting prohibited  |

| PFC25 | Specification of Alternate Function of P25 Pin |
|-------|--|
| 0     | TOB1B3 output                                  |
| 1     | TRGB1 input                                    |

| PFC24 | Specification of Alternate Function of P24 Pin |
|-------|--|
| 0     | TOB1T3 output                                  |
| 1     | EVTB1 input                                    |

| PFC23 | Specification of Alternate Function of P23 Pin |
|-------|--|
| 0     | TOB1B2 output                                  |
| 1     | TIB10 input                                    |

| PFCE22 | PFC22 | Specification of Alternate Function of P22 Pin |
|--------|-------|--|
| 0      | 0     | TOB1T2 output                                  |
| 0      | 1     | TIB13 input                                    |
| 1      | 0     | TOB13 output                                   |
| 1      | 1     | Setting prohibited                             |

| PFCE21 | PFC21 | Specification of Alternate Function of P21 Pin |
|--------|-------|--|
| 0      | 0     | TOB1B1 output                                  |
| 0      | 1     | TIB12 input                                    |
| 1      | 0     | TOB12 output                                   |
| 1      | 1     | Setting prohibited                             |

| PFCE20 | PFC20 | Specification of Alternate Function of P20 Pin |
|--------|-------|--|
| 0      | 0     | TOB1T1 output                                  |
| 0      | 1     | TIB11 input                                    |
| 1      | 0     | TOB11 output                                   |
| 1      | 1     | Setting prohibited                             |

(g) Pull-up resistor option register 2 (PU2)

After reset: 00H R/W Address: FFFFC44H

|     |      |      |      |      |      |      |      |      |
|-----|------|------|------|------|------|------|------|------|
|     | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| PU2 | PU27 | PU26 | PU25 | PU24 | PU23 | PU22 | PU21 | PU20 |

|      |  |
|------|--|
| PU2n | Control of on-chip pull-up resistor connection |
| 0    | Do not connect                                 |
| 1    | Connect <sup>Note</sup>                        |

**Note** An on-chip pull-up resistor can be connected only when the pins are in input mode in the port mode or when the pins function as input pins in the alternate-function mode. Moreover, an on-chip pull-up resistor can be connected to the TOB1T1 to TOB1T3 and TOB1B1 to TOB1B3 pins, these are output pins in the alternate-function mode, when these pins go into a high-impedance state due to the TOB1OFF pin, or software processing. An on-chip pull-up resistor cannot be connected when the pins are in output mode.

**Remark** n = 0 to 7

4.3.4 Port 3

Port 3 can be set to the input or output mode in 1-bit units.

Port 3 has an alternate function as the following pins.

Table 4-9. Alternate-Function Pins of Port 3

| Pin Name | Pin No. |     |    |     | Alternate-Function Pin Name                                    | I/O    | Pull-Up <sup>Note 1</sup> |
|----------|---------|-----|----|-----|--|--------|---------------------------|
|          | IF3     | IG3 |    |     |  |        |                           |
|          | GC      | GC  | GF | F1  |  |        |                           |
| P30      | 46      | 55  | 83 | K13 | RXDA1/SCL  | I/O    | Provided                  |
| P31      | 47      | 56  | 84 | K12 | TXDA1/SDA  | I/O    |                           |
| P32      | 48      | 57  | 85 | J14 | SIB1/RXDA2/ $\overline{CS1}$ <sup>Note 2</sup>                 | I/O    |                           |
| P33      | 49      | 58  | 86 | J13 | SOB1/TXDA2   | Output |                           |
| P34      | 50      | 59  | 87 | J12 | $\overline{SCKB1}$ /INTP11/ $\overline{CS0}$ <sup>Note 2</sup> | I/O    |                           |
| P35      | 51      | 60  | 88 | H14 | SIB2/RXDB  | Input  |                           |
| P36      | 52      | 61  | 89 | H13 | SOB2/TXDB  | Output |                           |
| P37      | 53      | 62  | 90 | H12 | $\overline{SCKB2}$ /INTP12/ASTB <sup>Note 2</sup>              | I/O    |                           |

- Notes**
1. Software pull-up function
  2.  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**Remark**

IF3: V850E/IF3  
 IG3: V850E/IG3  
 GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)  
 GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)  
 GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)  
 F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

(1) Registers

(a) Port 3 register (P3)

After reset: Undefined    R/W    Address: FFFFF406H

|    |     |     |     |     |     |     |     |     |
|----|-----|-----|-----|-----|-----|-----|-----|-----|
|    | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| P3 | P37 | P36 | P35 | P34 | P33 | P32 | P31 | P30 |

|     |   |
|-----|---|
| P3n | Control of output data (in output mode) |
| 0   | Output 0.                               |
| 1   | Output 1.                               |

**Remark**    n = 0 to 7

(b) Port 3 mode register (PM3)

After reset: FFH    R/W    Address: FFFFF426H

|     |      |      |      |      |      |      |      |      |
|-----|------|------|------|------|------|------|------|------|
|     | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| PM3 | PM37 | PM36 | PM35 | PM34 | PM33 | PM32 | PM31 | PM30 |

|      |                                    |
|------|------------------------------------|
| PM3n | Control of I/O mode (in port mode) |
| 0    | Output mode                        |
| 1    | Input mode                         |

**Remark**    n = 0 to 7



(c) Port 3 mode control register (PMC3)

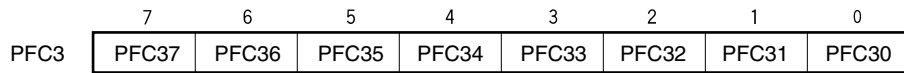
After reset: 00H R/W Address: FFFF446H

|      | 7  | 6   | 5     | 4     | 3     | 2     | 1     | 0     |
|------|--|---|-------|-------|-------|-------|-------|-------|
| PMC3 | PMC37                                      | PMC36   | PMC35 | PMC34 | PMC33 | PMC32 | PMC31 | PMC30 |
|      | Specification of operating mode of P37 pin |   |       |       |       |       |       |       |
|      | 0  | I/O port  |       |       |       |       |       |       |
|      | 1  | SCKB2 I/O/NTP12 input/ASTB output <sup>Note</sup>               |       |       |       |       |       |       |
|      | Specification of operating mode of P36 pin |   |       |       |       |       |       |       |
|      | 0  | I/O port  |       |       |       |       |       |       |
|      | 1  | SOB2 output/TXDB output   |       |       |       |       |       |       |
|      | Specification of operating mode of P35 pin |   |       |       |       |       |       |       |
|      | 0  | I/O port  |       |       |       |       |       |       |
|      | 1  | SIB2 input/RXDB input   |       |       |       |       |       |       |
|      | Specification of operating mode of P34 pin |   |       |       |       |       |       |       |
|      | 0  | I/O port  |       |       |       |       |       |       |
|      | 1  | SCKB1 I/O/INTP11 input/ $\overline{CS0}$ output <sup>Note</sup> |       |       |       |       |       |       |
|      | Specification of operating mode of P33 pin |   |       |       |       |       |       |       |
|      | 0  | I/O port  |       |       |       |       |       |       |
|      | 1  | SOB1 output/TXDA2 output  |       |       |       |       |       |       |
|      | Specification of operating mode of P32 pin |   |       |       |       |       |       |       |
|      | 0  | I/O port  |       |       |       |       |       |       |
|      | 1  | SIB1 input/RXDA2 input/ $\overline{CS1}$ output <sup>Note</sup> |       |       |       |       |       |       |
|      | Specification of operating mode of P31 pin |   |       |       |       |       |       |       |
|      | 0  | I/O port  |       |       |       |       |       |       |
|      | 1  | TXDA1 output/SDA I/O  |       |       |       |       |       |       |
|      | Specification of operating mode of P30 pin |   |       |       |       |       |       |       |
|      | 0  | I/O port  |       |       |       |       |       |       |
|      | 1  | RXDA1 input/SCL I/O   |       |       |       |       |       |       |

**Note**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**(d) Port 3 function control register (PFC3)**

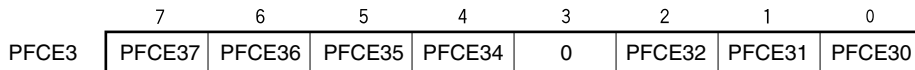
After reset: 00H    R/W    Address: FFFFF466H



**Remark** For the specifications of alternate functions, see **4.3.4 (1) (f) Settings of alternate functions of port 3.**

**(e) Port 3 function control expansion register (PFCE3)**

After reset: 00H    R/W    Address: FFFFF706H



**Remark** For the specifications of alternate functions, see **4.3.4 (1) (f) Settings of alternate functions of port 3.**

(f) Settings of alternate functions of port 3

| PFCE37 | PFC37 | Specification of Alternate Function of P37 Pin |
|--------|-------|--|
| 0      | 0     | SCKB2 input/output                             |
| 0      | 1     | INTP12 input                                   |
| 1      | 0     | ASTB output <sup>Note</sup>                    |
| 1      | 1     | Setting prohibited                             |

| PFCE36 | PFC36 | Specification of Alternate Function of P36 Pin |
|--------|-------|--|
| 0      | 0     | SOB2 output                                    |
| 0      | 1     | TXDB output                                    |
| 1      | 0     | Setting prohibited                             |
| 1      | 1     | Setting prohibited                             |

| PFCE35 | PFC35 | Specification of Alternate Function of P35 Pin |
|--------|-------|--|
| 0      | 0     | SIB2 input                                     |
| 0      | 1     | RXDB input                                     |
| 1      | 0     | Setting prohibited                             |
| 1      | 1     | Setting prohibited                             |

| PFCE34 | PFC34 | Specification of Alternate Function of P34 Pin |
|--------|-------|--|
| 0      | 0     | SCKB1 input/output                             |
| 0      | 1     | INTP11 input                                   |
| 1      | 0     | $\overline{CS0}$ output <sup>Note</sup>        |
| 1      | 1     | Setting prohibited                             |

| PFC33 | Specification of Alternate Function of P33 Pin |
|-------|--|
| 0     | SOB1 output                                    |
| 1     | TXDA2 output                                   |

| PFCE32 | PFC32 | Specification of Alternate Function of P32 Pin |
|--------|-------|--|
| 0      | 0     | SIB1 input                                     |
| 0      | 1     | RXDA2 input                                    |
| 1      | 0     | $\overline{CS1}$ output <sup>Note</sup>        |
| 1      | 1     | Setting prohibited                             |

**Note**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

| PFCE31 | PFC31 | Specification of Alternate Function of P31 Pin |
|--------|-------|--|
| 0      | 0     | TXDA1 output                                   |
| 0      | 1     | SDA input/output                               |
| 1      | 0     | Setting prohibited                             |
| 1      | 1     | Setting prohibited                             |

| PFCE30 | PFC30 | Specification of Alternate Function of P30 Pin |
|--------|-------|--|
| 0      | 0     | RXDA1 input                                    |
| 0      | 1     | SCL input/output                               |
| 1      | 0     | Setting prohibited                             |
| 1      | 1     | Setting prohibited                             |

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

After reset: 00H R/W Address: FFFFC46H

|     |      |      |      |      |      |      |      |      |
|-----|------|------|------|------|------|------|------|------|
|     | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| PU3 | PU37 | PU36 | PU35 | PU34 | PU33 | PU32 | PU31 | PU30 |

| PU3n | Control of on-chip pull-up resistor connection |
|------|--|
| 0    | Do not connect                                 |
| 1    | Connect <sup>Note</sup>                        |

**Note** An on-chip pull-up resistor can be connected only when the pins are in input mode in the port mode or when the pins function as input pins in the alternate-function mode (including the  $\overline{\text{SCKB1}}$  and  $\text{SCKB2}$  pins in the slave mode). An on-chip pull-up resistor cannot be connected when the pins are in output mode.

**Remark** n = 0 to 7

**(h) Port 3 function register (PF3)**

After reset: 00H R/W Address: FFFFC66H

|     |   |   |   |   |   |   |      |      |
|-----|---|---|---|---|---|---|------|------|
|     | 7 | 6 | 5 | 4 | 3 | 2 | 1    | 0    |
| PF3 | 0 | 0 | 0 | 0 | 0 | 0 | PF31 | PF30 |

| PF3n | Control of normal output/N-ch open-drain output |
|------|---|
| 0    | Normal output (CMOS output)                     |
| 1    | N-ch open-drain output <sup>Note</sup>          |

**Note** When using I<sup>2</sup>C, set as N-ch open-drain output.

**Remark** n = 0, 1

4.3.5 Port 4

Port 4 can be set to the input or output mode in 1-bit units.

Port 4 has an alternate function as the following pins.

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

| Pin Name | Pin No. |     |    |     | Alternate-Function Pin Name  | I/O   | Pull-Up <sup>Note 1</sup> |
|----------|---------|-----|----|-----|--|-------|---------------------------|
|          | IF3     | IG3 |    |     |  |       |                           |
|          | GC      | GC  | GF | F1  |  |       |                           |
| P40      | 38      | 47  | 75 | P12 | SIB0/RXDA0   | Input | Provided                  |
| P41      | 39      | 48  | 76 | N12 | SOB0/TXDA0/DCK <sup>Notes 2, 3</sup>                                 | I/O   |                           |
| P42      | 40      | 49  | 77 | M14 | SCKB0/INTP13/DDI <sup>Notes 2, 3</sup>                               | I/O   |                           |
| P43      | 41      | 50  | 78 | M13 | TECR1/TIT10/TOT10/INTP14   | I/O   |                           |
| P44      | 42      | 51  | 79 | L14 | TENC10/EVTT1/INTP15/ $\overline{\text{WAIT}}$ <sup>Note 4</sup>      | Input |                           |
| P45      | 43      | 52  | 80 | L13 | TENC11/TIT11/TOT11/INTP16/ $\overline{\text{WR1}}$ <sup>Note 4</sup> | I/O   |                           |
| P46      | 44      | 53  | 81 | L12 | TOA40/TIA40/INTP17/ $\overline{\text{WR0}}$ <sup>Note 4</sup>        | I/O   |                           |
| P47      | 45      | 54  | 82 | K14 | TOA41/TIA41/INTP18/ $\overline{\text{RD}}$ <sup>Note 4</sup>         | I/O   |                           |

**Notes** 1. Software pull-up function

2. V850E/IG3 only

3. The P41 and P42 pins also function as on-chip debug pins. The on-chip debug function or port function (including the alternate functions) can be selected by using the level of the  $\overline{\text{DRST}}$  pin, as shown in the table below.

| Port 4 Functions                                |  |
|---|--|
| Low-Level Input to $\overline{\text{DRST}}$ Pin | High-Level Input to $\overline{\text{DRST}}$ Pin |
| P41/SOB0/TXDA0                                  | DCK  |
| P42/SCKB0/INTP13                                | DDI  |

4.  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**Remark** IF3: V850E/IF3

IG3: V850E/IG3

GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)

GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)

GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

(1) Registers

(a) Port 4 register (P4)

After reset: Undefined    R/W    Address: FFFFF408H

|     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| P47 | P46 | P45 | P44 | P43 | P42 | P41 | P40 |

|     |   |
|-----|---|
| P4n | Control of output data (in output mode) |
| 0   | Output 0.                               |
| 1   | Output 1.                               |

**Remark**    n = 0 to 7

(b) Port 4 mode register (PM4)

After reset: FFH    R/W    Address: FFFFF428H

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| PM47 | PM46 | PM45 | PM44 | PM43 | PM42 | PM41 | PM40 |

|      |                                    |
|------|------------------------------------|
| PM4n | Control of I/O mode (in port mode) |
| 0    | Output mode                        |
| 1    | Input mode                         |

**Remark**    n = 0 to 7

(c) Port 4 mode control register (PMC4)

After reset: 00H R/W Address: FFFFF448H

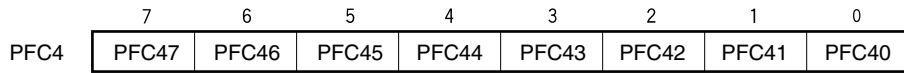
|      | 7  | 6   | 5     | 4     | 3     | 2     | 1     | 0     |
|------|--|---|-------|-------|-------|-------|-------|-------|
| PMC4 | PMC47                                      | PMC46   | PMC45 | PMC44 | PMC43 | PMC42 | PMC41 | PMC40 |
|      | Specification of operating mode of P47 pin |   |       |       |       |       |       |       |
|      | 0  | I/O port  |       |       |       |       |       |       |
|      | 1  | TOA41 output/TIA41 input/INTP18 input/ $\overline{RD}$ output <sup>Note 1</sup>               |       |       |       |       |       |       |
|      | Specification of operating mode of P46 pin |   |       |       |       |       |       |       |
|      | 0  | I/O port  |       |       |       |       |       |       |
|      | 1  | TOA40 output/TIA40 input/INTP17 input/ $\overline{WR0}$ output <sup>Note 1</sup>              |       |       |       |       |       |       |
|      | Specification of operating mode of P45 pin |   |       |       |       |       |       |       |
|      | 0  | I/O port  |       |       |       |       |       |       |
|      | 1  | TENC11 input/TIT11 input/TOT11 output/INTP16 input/ $\overline{WR1}$ output <sup>Note 1</sup> |       |       |       |       |       |       |
|      | Specification of operating mode of P44 pin |   |       |       |       |       |       |       |
|      | 0  | I/O port  |       |       |       |       |       |       |
|      | 1  | TENC10 input/EVTT1 input/INTP15 input/ $\overline{WAIT}$ input <sup>Note 1</sup>              |       |       |       |       |       |       |
|      | Specification of operating mode of P43 pin |   |       |       |       |       |       |       |
|      | 0  | I/O port  |       |       |       |       |       |       |
|      | 1  | TECR1 input/TIT10 input/TOT10 output/INTP14 input   |       |       |       |       |       |       |
|      | Specification of operating mode of P42 pin |   |       |       |       |       |       |       |
|      | 0  | I/O port  |       |       |       |       |       |       |
|      | 1  | $\overline{SCKB0}$ input/output/INTP13 input/DDI input <sup>Note 2</sup>                      |       |       |       |       |       |       |
|      | Specification of operating mode of P41 pin |   |       |       |       |       |       |       |
|      | 0  | I/O port  |       |       |       |       |       |       |
|      | 1  | SOB0 output/TXDA0 output/DCK input <sup>Note 2</sup>  |       |       |       |       |       |       |
|      | Specification of operating mode of P40 pin |   |       |       |       |       |       |       |
|      | 0  | I/O port  |       |       |       |       |       |       |
|      | 1  | SIB0 input/RXDA0 input  |       |       |       |       |       |       |

**Notes** 1.  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

2. V850E/IG3 only

**(d) Port 4 function control register (PFC4)**

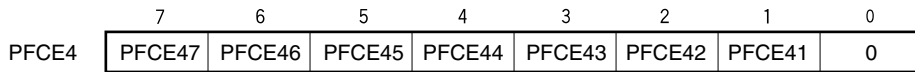
After reset: 00H    R/W    Address: FFFFF468H



**Remark** For the specifications of alternate functions, see **4.3.5 (1) (f) Settings of alternate functions of port 4.**

**(e) Port 4 function control expansion register (PFCE4)**

After reset: 00H    R/W    Address: FFFFF708H



**Remark** For the specifications of alternate functions, see **4.3.5 (1) (f) Settings of alternate functions of port 4.**



(f) Settings of alternate functions of port 4

| PFCE47 | PFC47 | Specification of Alternate Function of P47 Pin |
|--------|-------|--|
| 0      | 0     | TOA41 output                                   |
| 0      | 1     | TIA41 input                                    |
| 1      | 0     | INTP18 input                                   |
| 1      | 1     | $\overline{\text{RD}}$ output <sup>Note</sup>  |

| PFCE46 | PFC46 | Specification of Alternate Function of P46 Pin |
|--------|-------|--|
| 0      | 0     | TOA40 output                                   |
| 0      | 1     | TIA40 input                                    |
| 1      | 0     | INTP17 input                                   |
| 1      | 1     | $\overline{\text{WR0}}$ output <sup>Note</sup> |

| PFCE45 | PFC45 | Specification of Alternate Function of P45 Pin                |
|--------|-------|---|
| 0      | 0     | TENC11 input/TIT11 input (two functions are alternately used) |
| 0      | 1     | TOT11 output  |
| 1      | 0     | INTP16 input  |
| 1      | 1     | $\overline{\text{WR1}}$ output <sup>Note</sup>                |

| PFCE44 | PFC44 | Specification of Alternate Function of P44 Pin |
|--------|-------|--|
| 0      | 0     | TENC10 input                                   |
| 0      | 1     | EVTT1 input                                    |
| 1      | 0     | INTP15 input                                   |
| 1      | 1     | $\overline{\text{WAIT}}$ input <sup>Note</sup> |

| PFCE43 | PFC43 | Specification of Alternate Function of P43 Pin               |
|--------|-------|--|
| 0      | 0     | TECR1 input/TIT10 input (two functions are alternately used) |
| 0      | 1     | TOT10 output   |
| 1      | 0     | INTP14 input   |
| 1      | 1     | Setting prohibited   |

| PFCE42 | PFC42 | Specification of Alternate Function of P42 Pin |
|--------|-------|--|
| 0      | 0     | $\overline{\text{SCKB0}}$ I/O                  |
| 0      | 1     | INTP13 input                                   |
| 1      | 0     | Setting prohibited                             |
| 1      | 1     | Setting prohibited                             |

**Note**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

| PFCE41 | PFC41 | Specification of Alternate Function of P41 Pin |
|--------|-------|--|
| 0      | 0     | SOB0 output                                    |
| 0      | 1     | TXDA0 output                                   |
| 1      | 0     | Setting prohibited                             |
| 1      | 1     | Setting prohibited                             |

| PFC40 | Specification of Alternate Function of P40 Pin |
|-------|--|
| 0     | SIB0 input                                     |
| 1     | RXDA0 input                                    |

**(g) Pull-up resistor option register 4 (PU4)**

After reset: 00H    R/W    Address: FFFFC48H

|     |      |      |      |      |      |      |      |      |
|-----|------|------|------|------|------|------|------|------|
|     | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| PU4 | PU47 | PU46 | PU45 | PU44 | PU43 | PU42 | PU41 | PU40 |

| PU4n | Control of on-chip pull-up resistor connection |
|------|--|
| 0    | Do not connect                                 |
| 1    | Connect <sup>Note</sup>                        |

**Note** An on-chip pull-up resistor can be connected only when the pins are in input mode in the port mode or when the pins function as input pins in the alternate-function mode (including the  $\overline{\text{SCKB0}}$  pin in the slave mode). An on-chip pull-up resistor cannot be connected when the pins are in output mode.

**Remark** n = 0 to 7

4.3.6 Port 7

Port 7 is an input port with all its pins fixed to the input mode.  
The number of input port pins differs depending on the product.

| Generic Name | Number of I/O Ports   |
|--------------|-----------------------|
| V850E/IF3    | 4-bit input-only port |
| V850E/IG3    | 8-bit input-only port |

Port 7 has an alternate function as the following pins.

Table 4-11. Alternate-Function Pins of Port 7

| Pin Name              | Pin No. |     |    |    | Alternate-Function Pin Name | I/O   | Pull-Up <sup>Note 1</sup> |
|-----------------------|---------|-----|----|----|-----------------------------|-------|---------------------------|
|                       | IF3     | IG3 |    |    |                             |       |                           |
|                       | GC      | GC  | GF | F1 |                             |       |                           |
| P70                   | 20      | 25  | 53 | L1 | ANI20                       | Input | None                      |
| P71                   | 19      | 24  | 52 | L2 | ANI21                       | Input |                           |
| P72                   | 18      | 23  | 51 | L3 | ANI22                       | Input |                           |
| P73                   | 17      | 22  | 50 | K1 | ANI23                       | Input |                           |
| P74 <sup>Note 2</sup> | –       | 21  | 49 | K2 | ANI24 <sup>Note 2</sup>     | Input |                           |
| P75 <sup>Note 2</sup> | –       | 20  | 48 | K3 | ANI25 <sup>Note 2</sup>     | Input |                           |
| P76 <sup>Note 2</sup> | –       | 19  | 47 | K4 | ANI26 <sup>Note 2</sup>     | Input |                           |
| P77 <sup>Note 2</sup> | –       | 18  | 46 | J4 | ANI27 <sup>Note 2</sup>     | Input |                           |

- Notes** 1. Software pull-up function  
2. V850E/IG3 only

**Remark** IF3: V850E/IF3  
IG3: V850E/IG3  
GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)  
GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)  
GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)  
F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

(1) Registers

(a) Port 7 register (P7)

After reset: Undefined R Address: FFFFBB0H

|    |                     |                     |                     |                     |     |     |     |     |
|----|---------------------|---------------------|---------------------|---------------------|-----|-----|-----|-----|
|    | 7                   | 6                   | 5                   | 4                   | 3   | 2   | 1   | 0   |
| P7 | P77 <sup>Note</sup> | P76 <sup>Note</sup> | P75 <sup>Note</sup> | P74 <sup>Note</sup> | P73 | P72 | P71 | P70 |

|     |                       |
|-----|-----------------------|
| P7n | Control of input data |
| 0   | Input low level.      |
| 1   | Input high level.     |

**Note** Valid only in the V850E/IG3.

With the V850E/IF3, the read value of this register is undefined.

**Caution** When using a port input pin and analog input pin (ANI2n) together, be sure to set (1) the bit (PMC7n) of the PMC7 register to be used as the ANI2n pin.

**Remark** V850E/IF3: n = 0 to 3  
V850E/IG3: n = 0 to 7

(b) Port 7 mode control register (PMC7)

After reset: 00H R/W Address: FFFFBB8H

|      |                       |                       |                       |                       |       |       |       |       |
|------|-----------------------|-----------------------|-----------------------|-----------------------|-------|-------|-------|-------|
|      | 7                     | 6                     | 5                     | 4                     | 3     | 2     | 1     | 0     |
| PMC7 | PMC77 <sup>Note</sup> | PMC76 <sup>Note</sup> | PMC75 <sup>Note</sup> | PMC74 <sup>Note</sup> | PMC73 | PMC72 | PMC71 | PMC70 |

|       |   |
|-------|---|
| PMC7n | Specification of operating mode of P7n pin                                    |
| 0     | Input port (reading P7n enabled. Input buffer is on when this bit is read)    |
| 1     | ANI2n input (reading P7n disabled. Input buffer is off when this bit is read) |

**Note** Valid only in the V850E/IG3.

With the V850E/IF3, be sure to set these bits to 0.

**Cautions** 1. Do not change to the port mode using A/D converter 2 during A/D conversion.  
2. The PMC7 register enables or disables reading of the P7 register. When the PMC7n bit = 1, the input buffer does not turn on even when the P7 register is read. In this case, the read value of the P7n bit is fixed to the low level (V850E/IF3: n = 0 to 3, V850E/IG3: n = 0 to 7). This is to prevent through-current that may flow when the ANI2n input (intermediate level) is read.

**Remark** V850E/IF3: n = 0 to 3  
V850E/IG3: n = 0 to 7

4.3.7 Port DL

Port DL can be set to the input or output mode in 1-bit units.

Port DL has an alternate function as the following pins.

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

| Pin Name                | Pin No, |     |     |     | Alternate-Function Pin Name                    | I/O | Pull-Up <sup>Note 1</sup> |
|-------------------------|---------|-----|-----|-----|--|-----|---------------------------|
|                         | IF3     | IG3 |     |     |  |     |                           |
|                         | GC      | GC  | GF  | F1  |  |     |                           |
| PDL0                    | 65      | 81  | 9   | A10 | AD0 <sup>Note 3</sup>                          | I/O | Provided                  |
| PDL1                    | 64      | 80  | 8   | B10 | AD1 <sup>Note 3</sup>                          | I/O |                           |
| PDL2                    | 63      | 79  | 7   | A11 | AD2 <sup>Note 3</sup>                          | I/O |                           |
| PDL3                    | 62      | 78  | 6   | B11 | AD3 <sup>Note 3</sup>                          | I/O |                           |
| PDL4                    | 61      | 77  | 5   | C11 | AD4 <sup>Note 3</sup>                          | I/O |                           |
| PDL5                    | 60      | 76  | 4   | A12 | AD5 <sup>Note 3</sup> /FLMD1 <sup>Note 4</sup> | I/O |                           |
| PDL6                    | 59      | 75  | 3   | B12 | AD6 <sup>Note 3</sup>                          | I/O |                           |
| PDL7                    | 58      | 74  | 2   | C14 | AD7 <sup>Note 3</sup>                          | I/O |                           |
| PDL8                    | 57      | 73  | 1   | C13 | AD8 <sup>Note 3</sup>                          | I/O |                           |
| PDL9                    | 56      | 72  | 100 | D14 | AD9 <sup>Note 3</sup>                          | I/O |                           |
| PDL10 <sup>Note 2</sup> | –       | 71  | 99  | D13 | AD10 <sup>Note 3</sup>                         | I/O |                           |
| PDL11 <sup>Note 2</sup> | –       | 70  | 98  | D12 | AD11 <sup>Note 3</sup>                         | I/O |                           |
| PDL12 <sup>Note 2</sup> | –       | 69  | 97  | E14 | AD12 <sup>Note 3</sup>                         | I/O |                           |
| PDL13 <sup>Note 2</sup> | –       | 68  | 96  | E13 | AD13 <sup>Note 3</sup>                         | I/O |                           |
| PDL14 <sup>Note 2</sup> | –       | 67  | 95  | E12 | AD14 <sup>Note 3</sup>                         | I/O |                           |
| PDL15 <sup>Note 2</sup> | –       | 66  | 94  | F12 | AD15 <sup>Note 3</sup>                         | I/O |                           |

**Notes** 1. Software pull-up function

2. V850E/IG3 only

3.  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

4. This pin is used in the flash programming mode and does not have to be manipulated by a port control register. For details, see **CHAPTER 27 FLASH MEMORY**.

**Remark** IF3: V850E/IF3

IG3: V850E/IG3

GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)

GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)

GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

(1) Registers

(a) Port DL register (PDL)

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

|                               |   |                         |                         |                         |                         |                         |      |      |
|-------------------------------|---|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|------|------|
|                               | 15                                      | 14                      | 13                      | 12                      | 11                      | 10                      | 9    | 8    |
| PDL (PDLH <sup>Note 1</sup> ) | PDL15 <sup>Note 2</sup>                 | PDL14 <sup>Note 2</sup> | PDL13 <sup>Note 2</sup> | PDL12 <sup>Note 2</sup> | PDL11 <sup>Note 2</sup> | PDL10 <sup>Note 2</sup> | PDL9 | PDL8 |
|                               | 7                                       | 6                       | 5                       | 4                       | 3                       | 2                       | 1    | 0    |
| (PDLL)                        | PDL7                                    | PDL6                    | PDL5                    | PDL4                    | PDL3                    | PDL2                    | PDL1 | PDL0 |
|                               | Control of output data (in output mode) |                         |                         |                         |                         |                         |      |      |
| PDLn                          |   |                         |                         |                         |                         |                         |      |      |
| 0                             | Output 0.                               |                         |                         |                         |                         |                         |      |      |
| 1                             | Output 1.                               |                         |                         |                         |                         |                         |      |      |

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

2. Valid only in the V850E/IG3.

With the V850E/IF3, the read value of this register is undefined.

**Remarks** 1. The PDL register can be read or written in 16-bit units.

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

2. V850E/IF3: n = 0 to 9

V850E/IG3: n = 0 to 15

**(b) Port DL mode register (PMDL)**

After reset: FFFFH    R/W    Address: PMDL FFFF024H  
 PMDLL FFFF024H, PMDLH FFFF025H

|                                 |                          |                          |                          |                          |                          |                          |       |       |
|---------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|-------|-------|
|                                 | 15                       | 14                       | 13                       | 12                       | 11                       | 10                       | 9     | 8     |
| PMDL (PMDLH <sup>Note 1</sup> ) | PMDL15 <sup>Note 2</sup> | PMDL14 <sup>Note 2</sup> | PMDL13 <sup>Note 2</sup> | PMDL12 <sup>Note 2</sup> | PDAL11 <sup>Note 2</sup> | PDAL10 <sup>Note 2</sup> | 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 (in port mode) |
| 0     | Output mode                        |
| 1     | Input mode                         |

**Notes**

1. To read/write bits 8 to 15 of the PMDL register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PMDLH register.
2. Valid only in the V850E/IG3.  
 With the V850E/IF3, be sure to set these bits to 1.

**Remarks**

1. The PMDL register can be read or written in 16-bit units.  
 When the higher 8 bits of the PMDL register are used as the PMDLH register, and the lower 8 bits, as the PMDLL register, these registers can be read or written in 8-bit or 1-bit units.
2. V850E/IF3: n = 0 to 9  
 V850E/IG3: n = 0 to 15

(c) Port DL mode control register (PMCDL)

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

|                                   |                           |                           |                           |                           |                           |                           |        |        |
|-----------------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|--------|--------|
|                                   | 15                        | 14                        | 13                        | 12                        | 11                        | 10                        | 9      | 8      |
| PMCDL (PMCDLH <sup>Note 1</sup> ) | PMCDL15 <sup>Note 2</sup> | PMCDL14 <sup>Note 2</sup> | PMCDL13 <sup>Note 2</sup> | PMCDL12 <sup>Note 2</sup> | PMCDL11 <sup>Note 2</sup> | PMCDL10 <sup>Note 2</sup> | PMCDL9 | PMCDL8 |

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

|        |   |
|--------|---|
| PMCDLn | Specification of operating mode of PMCDLn pin |
| 0      | I/O port                                      |
| 1      | ADn input/output                              |

- Notes**
- To read/write bits 8 to 15 of the PMCDL register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PMCDLH register.
  - Valid only in the V850E/IG3.  
 With the V850E/IF3, be sure to set these bits to 0.

- Remarks**
- The PMCDL register can be read or written in 16-bit units.  
 When the higher 8 bits of the PMCDL register are used as the PMCDLH register, and the lower 8 bits, as the PMCDLL register, these registers can be read or written in 8-bit or 1-bit units.
  - V850E/IF3: n = 0 to 9  
 V850E/IG3: n = 0 to 15



(d) Pull-up resistor option register DL (PUDL)

After reset: 0000H R/W Address: PUDL FFFFFFF44H  
PUDLL FFFFFFF44H, PUDLH FFFFFFF45H

|                                 |                          |                          |                          |                          |                          |                          |       |       |
|---------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|-------|-------|
|                                 | 15                       | 14                       | 13                       | 12                       | 11                       | 10                       | 9     | 8     |
| PUDL (PUDLH <sup>Note 1</sup> ) | PUDL15 <sup>Note 2</sup> | PUDL14 <sup>Note 2</sup> | PUDL13 <sup>Note 2</sup> | PUDL12 <sup>Note 2</sup> | PUDL11 <sup>Note 2</sup> | PUDL10 <sup>Note 2</sup> | PUDL9 | PUDL8 |

|         |       |       |       |       |       |       |       |       |
|---------|-------|-------|-------|-------|-------|-------|-------|-------|
|         | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| (PUDLL) | PUDL7 | PUDL6 | PUDL5 | PUDL4 | PUDL3 | PUDL2 | PUDL1 | PUDL0 |

| PUDLn | Control of on-chip pull-up resistor connection |
|-------|--|
| 0     | Do not connect                                 |
| 1     | Connect <sup>Note 3</sup>                      |

- Notes**
- To read/write bits 8 to 15 of the PUDL register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PUDLH register.
  - Valid only in the V850E/IG3.  
With the V850E/IF3, be sure to set these bits to 0.
  - An on-chip pull-up resistor can be connected only when the pins are in input mode in the port mode.  
An on-chip pull-up resistor cannot be connected when the pins are in output mode.

- Remarks**
- The PUDL register can be read or written in 16-bit units.  
When the higher 8 bits of the PUDL register are used as the PUDLH register, and the lower 8 bits, as the PUDLL register, these registers can be read or written in 8-bit or 1-bit units.
  - V850E/IF3: n = 0 to 9  
V850E/IG3: n = 0 to 15

#### 4.4 Output Data and Port Read Value for Each Setting

Table 4-14 shows the values used to select the alternate function of the respective pins, output data and port read values for each setting. In addition to the settings shown in Table 4-14, the setting of each peripheral function control register is required.

**Table 4-13. Output Data and Port Read Value for Each Setting (1/12)**

| Port Name   | Function  | PMCmn | PFCEmn | PFCmn | PMmn      | Output Data                        | Pmn Read Value         |           |
|---|---|-------|--------|-------|-----------|------------------------------------|------------------------|-----------|
| P00, P01, P02 <sup>Note</sup> ,<br>P03 <sup>Note</sup>  | Output port   | 0     | ×      | ×     | 0         | Port latch                         | Port latch             |           |
|   | Input port  |       |        |       | 1         | –                                  | Pin level              |           |
|   | TOA20, TOA21,<br>TOA30 <sup>Note</sup> , TOA31 <sup>Note</sup>                                    | 1     | 0      | 0     | 0         | Alternate output<br>(timer output) | Port latch             |           |
|   |   |       |        |       | 1         |                                    | Pin level              |           |
|   | TIA20, TIA21,<br>TIA30 <sup>Note</sup> , TIA31 <sup>Note</sup>                                    | 1     | 0      | 1     | 0         | –                                  | Port latch             | Alternate |
| 1   |   |       |        |       | Pin level |                                    |                        |           |
| TOA2OFF,<br>INTP00, INTP01,<br>TOA3OFF <sup>Note</sup> ,<br>INTP02 <sup>Note</sup> , INTP03 <sup>Note</sup> | 1   | 1     | 0      | 0     | –         | Port latch                         | Alternate<br>input (ne |           |
|   |   |       |        | 1     |           | Pin level                          |                        |           |
| P04 <sup>Note</sup> , P06 <sup>Note</sup>   | Output port   | 0     | ×      | ×     | 0         | Port latch                         | Port latch             |           |
|   | Input port  |       |        |       | 1         | –                                  | Pin level              |           |
|   | TECR0 <sup>Note</sup> , TIT00 <sup>Note</sup> ,<br>TENC01 <sup>Note</sup> , TIT01 <sup>Note</sup> | 1     | 0      | 0     | 0         | –                                  | Port latch             | Alternate |
|   |   |       |        |       | 1         |                                    | Pin level              |           |
|   | TOT00 <sup>Note</sup> , TOT01 <sup>Note</sup>   | 1     | 0      | 1     | 0         | Alternate output<br>(timer output) | Port latch             |           |
| 1   |   |       |        |       | Pin level |                                    |                        |           |
| INTP04 <sup>Note</sup> ,<br>INTP06 <sup>Note</sup>  | 1   | 1     | 0      | 0     | –         | Port latch                         | Alternate<br>(necessa  |           |
|   |   |       |        | 1     |           | Pin level                          |                        |           |

**Note** V850E/IG3 only

**Remark** ×: don't care

Table 4-13. Output Data and Port Read Value for Each Setting (2/12)

| Port Name             | Function                 | PMCmn | PFCEmn | PFCmn | PMmn | Output Data                      | Pmn Read Value |           |
|-----------------------|--------------------------|-------|--------|-------|------|----------------------------------|----------------|-----------|
| P05 <sup>Note 1</sup> | Output port              | 0     | ×      | ×     | 0    | Port latch                       | Port latch     |           |
|                       | Input port               |       |        |       | 1    | –                                | Pin level      |           |
|                       | TENC00 <sup>Note 1</sup> | 1     | 0      | 0     | 0    | –                                | Port latch     | Alternate |
|                       |                          |       |        |       | 1    |                                  | Pin level      |           |
|                       | EVTT0 <sup>Note 1</sup>  | 1     | 0      | 1     | 0    | –                                | Port latch     | Alternate |
|                       |                          |       |        |       | 1    |                                  | Pin level      |           |
|                       | INTP05 <sup>Note 1</sup> | 1     | 1      | 0     | 0    | –                                | Port latch     | Alternate |
|                       |                          |       |        |       | 1    |                                  | Pin level      |           |
| P07 <sup>Note 1</sup> | Output port              | 0     | None   | ×     | 0    | Port latch                       | Port latch     |           |
|                       | Input port               |       |        |       | 1    | –                                | Pin level      |           |
|                       | INTP07 <sup>Note 1</sup> | 1     | None   | 0     | 0    | –                                | Port latch     | Alternate |
|                       |                          |       |        |       | 1    |                                  | Pin level      |           |
|                       | CLKOUT <sup>Note 2</sup> | 1     | None   | 1     | 0    | Alternate output<br>(bus output) | Port latch     |           |
|                       |                          |       |        |       | 1    |                                  | Pin level      |           |

- Notes** 1. V850E/IG3 only  
2.  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**Remark** ×: don't care

Table 4-13. Output Data and Port Read Value for Each Setting (3/12)

| Port Name  | Function                  | PMCmn | PFCEmn | PFCmn | PMmn                                 | Output Data                          | Pmn Read Value |           |
|--|---------------------------|-------|--------|-------|--------------------------------------|--------------------------------------|----------------|-----------|
| P10 to P12   | Output port               | 0     | ×      | ×     | 0                                    | Port latch                           | Port latch     |           |
|  | Input port                |       |        |       | 1                                    | –                                    | Pin level      |           |
|  | TOB0T1, TOB0B1,<br>TOB0T2 | 1     | 0      | 0     | 0                                    | Alternate output 1<br>(timer output) | Port latch     |           |
|  |                           |       |        |       | 1                                    |                                      | Pin level      |           |
|  | TIB01 to TIB03            | 1     | 0      | 1     | 0                                    | –                                    | Port latch     | Alternate |
|  |                           |       |        |       | 1                                    |                                      | Pin level      |           |
| TOB01 to TOB03   | 1                         | 1     | 0      | 0     | Alternate output 2<br>(timer output) | Port latch                           |                |           |
|  |                           |       |        | 1     |                                      | Pin level                            |                |           |
| A0 <sup>Note</sup> , A1 <sup>Note</sup> , A2 <sup>Note</sup> | 1                         | 1     | 1      | 0     | Alternate output 3<br>(bus output)   | Port latch                           |                |           |
|  |                           |       |        | 1     |                                      | Pin level                            |                |           |
| P13 to P15   | Output port               | 0     | ×      | ×     | 0                                    | Port latch                           | Port latch     |           |
|  | Input port                |       |        |       | 1                                    | –                                    | Pin level      |           |
|  | TOB0B2, TOB0T3,<br>TOB0B3 | 1     | 0      | 0     | 0                                    | Alternate output 1<br>(timer output) | Port latch     |           |
|  |                           |       |        |       | 1                                    |                                      | Pin level      |           |
|  | TIB00, EVTB0,<br>TRGB0    | 1     | 0      | 1     | 0                                    | –                                    | Port latch     | Alternate |
|  |                           |       |        |       | 1                                    |                                      | Pin level      |           |
| A3 <sup>Note</sup> , A4 <sup>Note</sup> , A5 <sup>Note</sup> | 1                         | 1     | 0      | 0     | Alternate output 2<br>(bus output)   | Port latch                           |                |           |
|  |                           |       |        | 1     |                                      | Pin level                            |                |           |

**Note**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**Remark** ×: don't care

Table 4-13. Output Data and Port Read Value for Each Setting (4/12)

| Port Name          | Function                  | PMCmn | PFCEmn | PFCmn | PMmn                                 | Output Data                          | Pmn Read Value |                        |
|--------------------|---------------------------|-------|--------|-------|--------------------------------------|--------------------------------------|----------------|------------------------|
| P16                | Output port               | 0     | ×      | ×     | 0                                    | Port latch                           | Port latch     |                        |
|                    | Input port                |       |        |       | 1                                    | –                                    | Pin level      |                        |
|                    | TOB0OFF/INTP08            | 1     | 0      | 0     | 0                                    | Alternate output 1<br>(timer output) | Port latch     |                        |
|                    |                           |       |        |       | 1                                    |                                      | Pin level      |                        |
|                    | ADTRG0/INTADT0            | 1     | 0      | 1     | 0                                    | –                                    | Port latch     | Alternate<br>input (ne |
|                    | 1                         |       |        |       | Pin level                            |                                      |                |                        |
| A6 <sup>Note</sup> | 1                         | 1     | 0      | 0     | Alternate output 2<br>(bus output)   | Port latch                           |                |                        |
|                    |                           |       |        | 1     |                                      | Pin level                            |                |                        |
| P17                | Output port               | 0     | ×      | ×     | 0                                    | Port latch                           | Port latch     |                        |
|                    | Input port                |       |        |       | 1                                    | –                                    | Pin level      |                        |
|                    | TOB00                     | 1     | 0      | 0     | 0                                    | Alternate output 1<br>(timer output) | Port latch     |                        |
|                    |                           |       |        |       | 1                                    |                                      | Pin level      |                        |
|                    | INTP09                    | 1     | 0      | 1     | 0                                    | –                                    | Port latch     | Alternate<br>(necessa  |
|                    | 1                         |       |        |       | Pin level                            |                                      |                |                        |
| A7 <sup>Note</sup> | 1                         | 1     | 0      | 0     | Alternate output 2<br>(bus output)   | Port latch                           |                |                        |
|                    |                           |       |        | 1     |                                      | Pin level                            |                |                        |
| P20 to P22         | Output port               | 0     | ×      | ×     | 0                                    | Port latch                           | Port latch     |                        |
|                    | Input port                |       |        |       | 1                                    | –                                    | Pin level      |                        |
|                    | TOB1T1, TOB1B1,<br>TOB1T2 | 1     | 0      | 0     | 0                                    | Alternate output 1<br>(timer output) | Port latch     |                        |
|                    |                           |       |        |       | 1                                    |                                      | Pin level      |                        |
|                    | TIB11 to TIB13            | 1     | 0      | 1     | 0                                    | –                                    | Port latch     | Alternate              |
|                    | 1                         |       |        |       | Pin level                            |                                      |                |                        |
| TOB11 to TOB13     | 1                         | 1     | 0      | 0     | Alternate output 2<br>(timer output) | Port latch                           |                |                        |
|                    |                           |       |        | 1     |                                      | Pin level                            |                |                        |

**Note**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**Remark** ×: don't care

Table 4-13. Output Data and Port Read Value for Each Setting (5/12)

| Port Name              | Function                  | PMCmn | PFCEmn | PFCmn | PMmn      | Output Data                        | Pmn Read Value         |                        |
|------------------------|---------------------------|-------|--------|-------|-----------|------------------------------------|------------------------|------------------------|
| P23 to P25             | Output port               | 0     | None   | ×     | 0         | Port latch                         | Port latch             |                        |
|                        | Input port                |       |        |       | 1         | –                                  | Pin level              |                        |
|                        | TOB1B2, TOB1T3,<br>TOB1B3 | 1     | None   | 0     | 0         | Alternate output<br>(timer output) | Port latch             |                        |
|                        |                           |       |        |       | 1         |                                    | Pin level              |                        |
| TIB10, EVTB1,<br>TRGB1 | 1                         | None  | 1      | 0     | –         | Port latch                         | Alternate              |                        |
|                        |                           |       |        | 1     |           | Pin level                          |                        |                        |
| P26                    | Output port               | 0     | ×      | ×     | 0         | Port latch                         | Port latch             |                        |
|                        | Input port                |       |        |       | 1         | –                                  | Pin level              |                        |
|                        | TOB10                     | 1     | 0      | 0     | 0         | Alternate output<br>(timer output) | Port latch             |                        |
|                        |                           |       |        |       | 1         |                                    | Pin level              |                        |
|                        | TOB1OFF/INTP10            | 1     | 0      | 1     | 0         | –                                  | Port latch             | Alternate<br>input (ne |
| 1                      |                           |       |        |       | Pin level |                                    |                        |                        |
| ADTRG1/INTADT1         | 1                         | 1     | 0      | 0     | –         | Port latch                         | Alternate<br>input (ne |                        |
|                        |                           |       |        | 1     |           | Pin level                          |                        |                        |
| P27 <sup>Note</sup>    | Output port               | None  | None   | None  | 0         | Port latch                         | Port latch             |                        |
|                        | Input port                |       |        |       | 1         | –                                  | Pin level              |                        |

**Note** The P27 pin is also used for on-chip debugging (V850E/IG3 only). Switching between on-chip debug function and port function can be set by using the  $\overline{\text{DRST}}$  pin level. The following shows the setting method.

| Port 2 Functions                                |  |
|---|--|
| Low-Level Input to $\overline{\text{DRST}}$ Pin | High-Level Input to $\overline{\text{DRST}}$ Pin |
| P27   | DMS  |

**Remark** ×: don't care

Table 4-13. Output Data and Port Read Value for Each Setting (6/12)

| Port Name                               | Function    | PMCmn | PFCEmn | PFCmn | PMmn                             | Output Data                         | Pmn Read Value |                         |
|---|-------------|-------|--------|-------|----------------------------------|-------------------------------------|----------------|-------------------------|
| P30                                     | Output port | 0     | ×      | ×     | 0                                | Port latch                          | Port latch     |                         |
|   | Input port  |       |        |       | 1                                | –                                   | Pin level      |                         |
|   | RXDA1       | 1     | 0      | 0     | 0                                | –                                   | Port latch     | Alternate               |
|   |             |       |        |       | 1                                |                                     | Pin level      |                         |
|   | SCL         | 1     | 0      | 1     | 0                                | Alternate I/O<br>(serial I/O)       | Port latch     | Output in<br>Input in s |
|   |             |       |        |       | 1                                |                                     | Pin level      |                         |
| P31                                     | Output port | 0     | ×      | ×     | 0                                | Port latch                          | Port latch     |                         |
|   | Input port  |       |        |       | 1                                | –                                   | Pin level      |                         |
|   | TXDA1       | 1     | 0      | 0     | 0                                | Alternate output<br>(serial output) | Port latch     |                         |
|   |             |       |        |       | 1                                |                                     | Pin level      |                         |
|   | SDA         | 1     | 0      | 1     | 0                                | Alternate I/O<br>(serial I/O)       | Port latch     | Output in<br>Input in s |
|   |             |       |        |       | 1                                |                                     | Pin level      |                         |
| P32                                     | Output port | 0     | ×      | ×     | 0                                | Port latch                          | Port latch     |                         |
|   | Input port  |       |        |       | 1                                | –                                   | Pin level      |                         |
|   | SIB1        | 1     | 0      | 0     | 0                                | –                                   | Port latch     | Alternate               |
|   |             |       |        |       | 1                                |                                     | Pin level      |                         |
|   | RXDA2       | 1     | 0      | 1     | 0                                | –                                   | Port latch     | Alternate               |
|   |             |       |        |       | 1                                |                                     | Pin level      |                         |
| $\overline{\text{CS1}}$ <sup>Note</sup> | 1           | 1     | 0      | 0     | Alternate output<br>(bus output) | Port latch                          |                |                         |
|   |             |       |        | 1     |                                  | Pin level                           |                |                         |

**Note**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**Remark** ×: don't care



Table 4-13. Output Data and Port Read Value for Each Setting (7/12)

| Port Name           | Function    | PMCmn | PFCEmn | PFCmn | PMmn                             | Output Data                           | Pmn Read Value |            |
|---------------------|-------------|-------|--------|-------|----------------------------------|---------------------------------------|----------------|------------|
| P33                 | Output port | 0     | None   | ×     | 0                                | Port latch                            | Port latch     |            |
|                     | Input port  |       |        |       | 1                                | –                                     | Pin level      |            |
|                     | SOB1        | 1     | None   | 0     | 0                                | Alternate output 1<br>(serial output) | Port latch     |            |
|                     |             |       |        |       | 1                                |                                       | Pin level      |            |
|                     | TXDA2       | 1     | None   | 1     | 0                                | Alternate output 2<br>(serial output) | Port latch     |            |
|                     |             |       |        |       | 1                                |                                       | Pin level      |            |
| P34                 | Output port | 0     | ×      | ×     | 0                                | Port latch                            | Port latch     |            |
|                     | Input port  |       |        |       | 1                                | –                                     | Pin level      |            |
|                     | SCKB1       | 1     | 0      | 0     | 0                                | Alternate I/O<br>(serial I/O)         | Port latch     | Output in  |
|                     |             |       |        |       | 1                                |                                       | Pin level      | Input in s |
|                     | INTP11      | 1     | 0      | 1     | 0                                | –                                     | Port latch     | Alternate  |
|                     |             |       |        |       | 1                                |                                       | Pin level      | (necessa   |
| CS0 <sup>Note</sup> | 1           | 1     | 0      | 0     | Alternate output<br>(bus output) | Port latch                            |                |            |
|                     |             |       |        | 1     |                                  | Pin level                             |                |            |
| P35                 | Output port | 0     | ×      | ×     | 0                                | Port latch                            | Port latch     |            |
|                     | Input port  |       |        |       | 1                                | –                                     | Pin level      |            |
|                     | SIB2        | 1     | 0      | 0     | 0                                | –                                     | Port latch     | Alternate  |
|                     |             |       |        |       | 1                                |                                       | Pin level      |            |
|                     | RXDB        | 1     | 0      | 1     | 0                                | –                                     | Port latch     | Alternate  |
|                     |             |       |        |       | 1                                |                                       | Pin level      |            |

**Note**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**Remark** ×: don't care

Table 4-13. Output Data and Port Read Value for Each Setting (8/12)

| Port Name            | Function                  | PMCmn | PFCEmn | PFCmn | PMmn                             | Output Data                           | Pmn Read Value |            |
|----------------------|---------------------------|-------|--------|-------|----------------------------------|---------------------------------------|----------------|------------|
| P36                  | Output port               | 0     | ×      | ×     | 0                                | Port latch                            | Port latch     |            |
|                      | Input port                |       |        |       | 1                                | –                                     | Pin level      |            |
|                      | SOB2                      | 1     | 0      | 0     | 0                                | Alternate output 1<br>(serial output) | Port latch     |            |
|                      |                           |       |        |       | 1                                |                                       | Pin level      |            |
|                      | TXDB                      | 1     | 0      | 1     | 0                                | Alternate output 2<br>(serial output) | Port latch     |            |
|                      |                           |       |        |       | 1                                |                                       | Pin level      |            |
| P37                  | Output port               | 0     | ×      | ×     | 0                                | Port latch                            | Port latch     |            |
|                      | Input port                |       |        |       | 1                                | –                                     | Pin level      |            |
|                      | $\overline{\text{SCKB2}}$ | 1     | 0      | 0     | 0                                | Alternate I/O<br>(serial I/O)         | Port latch     | Output in  |
|                      |                           |       |        |       | 1                                |                                       | Pin level      | Input in s |
|                      | INTP12                    | 1     | 0      | 1     | 0                                | –                                     | Port latch     | Alternate  |
|                      |                           |       |        |       | 1                                |                                       | Pin level      | (necessa   |
| ASTB <sup>Note</sup> | 1                         | 1     | 0      | 0     | Alternate output<br>(bus output) | Port latch                            |                |            |
|                      |                           |       |        | 1     |                                  | Pin level                             |                |            |
| P40                  | Output port               | 0     | None   | ×     | 0                                | Port latch                            | Port latch     |            |
|                      | Input port                |       |        |       | 1                                | –                                     | Pin level      |            |
|                      | SIB0                      | 1     | None   | 0     | 0                                | –                                     | Port latch     | Alternate  |
|                      |                           |       |        |       | 1                                |                                       | Pin level      |            |
|                      | RXDA0                     | 1     | None   | 1     | 0                                | –                                     | Port latch     | Alternate  |
|                      |                           |       |        |       | 1                                |                                       | Pin level      |            |

**Note**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**Remark** ×: don't care

Table 4-13. Output Data and Port Read Value for Each Setting (9/12)

| Port Name           | Function    | PMCmn | PFCEmn | PFCmn | PMmn | Output Data                           | Pmn Read Value |            |
|---------------------|-------------|-------|--------|-------|------|---------------------------------------|----------------|------------|
| P41 <sup>Note</sup> | Output port | 0     | ×      | ×     | 0    | Port latch                            | Port latch     |            |
|                     | Input port  |       |        |       | 1    | –                                     | Pin level      |            |
|                     | SOB0        | 1     | 0      | 0     | 0    | Alternate output 1<br>(serial output) | Port latch     |            |
|                     |             |       |        |       | 1    |                                       | Pin level      |            |
|                     | TXDA0       | 1     | 0      | 1     | 0    | Alternate output 2<br>(serial output) | Port latch     |            |
|                     |             |       |        |       | 1    |                                       | Pin level      |            |
| P42 <sup>Note</sup> | Output port | 0     | ×      | ×     | 0    | Port latch                            | Port latch     |            |
|                     | Input port  |       |        |       | 1    | –                                     | Pin level      |            |
|                     | SCKB0       | 1     | 0      | 0     | 0    | Alternate I/O<br>(serial I/O)         | Port latch     | Output in  |
|                     |             |       |        |       | 1    |                                       | Pin level      | Input in s |
|                     | INTP13      | 1     | 0      | 1     | 0    | –                                     | Port latch     | Alternate  |
|                     |             |       |        |       | 1    |                                       | Pin level      | (necessa   |

**Note** The P41 and P42 pins are also used for on-chip debugging (V850E/IG3 only). Switching between on-chip debug function (alternate function) can be set by using the  $\overline{\text{DRST}}$  pin level. The following shows the setting method.

| Port 4 Functions                                |  |
|---|--|
| Low-Level Input to $\overline{\text{DRST}}$ Pin | High-Level Input to $\overline{\text{DRST}}$ Pin |
| P41/SOB0/TXDA0                                  | DCK  |
| P42/SCKB0/INTP13                                | DDI  |

**Remark** ×: don't care

**Table 4-13. Output Data and Port Read Value for Each Setting (10/12)**

| Port Name | Function             | PMCmn | PFCEmn | PFCmn | PMmn | Output Data                        | Pmn Read Value |           |
|-----------|----------------------|-------|--------|-------|------|------------------------------------|----------------|-----------|
| P43       | Output port          | 0     | ×      | ×     | 0    | Port latch                         | Port latch     |           |
|           | Input port           |       |        |       | 1    | –                                  | Pin level      |           |
|           | TECR1/TIT10          | 1     | 0      | 0     | 0    | –                                  | Port latch     | Alternate |
|           |                      |       |        |       | 1    |                                    | Pin level      |           |
|           | TOT10                | 1     | 0      | 1     | 0    | Alternate output<br>(timer output) | Port latch     |           |
|           |                      |       |        |       | 1    |                                    | Pin level      |           |
|           | INTP14               | 1     | 1      | 0     | 0    | –                                  | Port latch     | Alternate |
|           |                      |       |        |       | 1    |                                    | Pin level      |           |
| P44       | Output port          | 0     | ×      | ×     | 0    | Port latch                         | Port latch     |           |
|           | Input port           |       |        |       | 1    | –                                  | Pin level      |           |
|           | TENC10               | 1     | 0      | 0     | 0    | –                                  | Port latch     | Alternate |
|           |                      |       |        |       | 1    |                                    | Pin level      |           |
|           | EVTT1                | 1     | 0      | 1     | 0    | Alternate output<br>(timer output) | Port latch     |           |
|           |                      |       |        |       | 1    |                                    | Pin level      |           |
|           | INTP15               | 1     | 1      | 0     | 0    | –                                  | Port latch     | Alternate |
|           |                      |       |        |       | 1    |                                    | Pin level      |           |
|           | WAIT <sup>Note</sup> | 1     | 1      | 1     | 0    | –                                  | Port latch     | Alternate |
|           |                      |       |        |       | 1    |                                    | Pin level      |           |

**Note**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**Remark** ×: don't care

Table 4-13. Output Data and Port Read Value for Each Setting (11/12)

| Port Name   | Function       | PMCmn | PFCEmn | PFCmn | PMmn                               | Output Data                          | Pmn Read Value |           |
|---|----------------|-------|--------|-------|------------------------------------|--------------------------------------|----------------|-----------|
| P45   | Output port    | 0     | ×      | ×     | 0                                  | Port latch                           | Port latch     |           |
|   | Input port     |       |        |       | 1                                  | –                                    | Pin level      |           |
|   | TENC11/TIT11   | 1     | 0      | 0     | 0                                  | –                                    | Port latch     | Alternate |
|   |                |       |        |       | 1                                  |                                      | Pin level      |           |
|   | TOT11          | 1     | 0      | 1     | 0                                  | Alternate output<br>1 (timer output) | Port latch     |           |
|   |                |       |        |       | 1                                  |                                      | Pin level      |           |
|   | INTP16         | 1     | 1      | 0     | 0                                  | –                                    | Port latch     | Alternate |
|   |                |       |        |       | 1                                  |                                      | Pin level      |           |
| $\overline{WR1}^{\text{Note}}$                              | 1              | 1     | 1      | 0     | Alternate output<br>2 (bus output) | Port latch                           |                |           |
|   |                |       |        | 1     |                                    | Pin level                            |                |           |
| P46, P47  | Output port    | 0     | ×      | ×     | 0                                  | Port latch                           | Port latch     |           |
|   | Input port     |       |        |       | 1                                  | –                                    | Pin level      |           |
|   | TOA40, TOA41   | 1     | 0      | 0     | 0                                  | Alternate output<br>1 (timer output) | Port latch     |           |
|   |                |       |        |       | 1                                  |                                      | Pin level      |           |
|   | TIA40, TIA41   | 1     | 0      | 1     | 0                                  | –                                    | Port latch     | Alternate |
|   |                |       |        |       | 1                                  |                                      | Pin level      |           |
|   | INTP17, INTP18 | 1     | 1      | 0     | 0                                  | –                                    | Port latch     | Alternate |
|   |                |       |        |       | 1                                  |                                      | Pin level      |           |
| $\overline{WR0}^{\text{Note}}, \overline{RD}^{\text{Note}}$ | 1              | 1     | 1      | 0     | Alternate output<br>2 (bus output) | Port latch                           |                |           |
|   |                |       |        | 1     |                                    | Pin level                            |                |           |

**Note**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**Remark** ×: don't care

Table 4-13. Output Data and Port Read Value for Each Setting (12/12)

| Port Name  | Function  | PMCmn | PFCEmn | PFCmn | PMmn | Output Data                | Pmn Read Value |           |
|--|---|-------|--------|-------|------|----------------------------|----------------|-----------|
| P70 to P73,<br>P74 to P77 <sup>Note 1</sup>          | Input port  | 0     | None   | None  | None | –                          | Pin level      | Input-onl |
|  | ANI20 to ANI23,<br>ANI24 to ANI27 <sup>Note 1</sup> | 1     |        |       |      | –                          | Low level      |           |
| PDL0 to PDL9,<br>PDL10 to<br>PDL15 <sup>Note 1</sup> | Output port   | 0     | None   | None  | 0    | Port latch                 | Port latch     |           |
|  | Input port  |       |        |       | 1    | –                          | Pin level      |           |
|  | AD0 to AD15 <sup>Note 2</sup>                       | 1     | None   | None  | 0    | Alternate I/O<br>(bus I/O) | Port latch     |           |
|  |   |       |        |       | 1    |                            | Pin level      |           |

- Notes**
1. V850E/IG3 only
  2.  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only
  3. The PDL5 pin is also used in flash programming mode. This pin does not have to be manipulated by a port control register.  
**FLASH MEMORY.**

**Remark** ×: don't care

#### **4.5 Port Register Settings When Alternate Function Is Used**

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

Table 4-14. Using Port Pin as Alternate-Function Pin (1/8)

| Pin Name            | Alternate Pin           |        | Pnx Bit of Pn Register     | PMnx Bit of PMn Register    | PMCnx Bit of PMCn Register | PFCEnx Bit of PFCEn Register | PFCnx Bit of PFCn Register |
|---------------------|-------------------------|--------|----------------------------|-----------------------------|----------------------------|------------------------------|----------------------------|
|                     | Name                    | I/O    |                            |                             |                            |                              |                            |
| P00                 | TOA20                   | Output | P00 = Setting not required | PM00 = Setting not required | PMC00 = 1                  | PFCE00 = 0                   | PFC00 = 0                  |
|                     | TIA20                   | Input  | P00 = Setting not required | PM00 = Setting not required | PMC00 = 1                  | PFCE00 = 0                   | PFC00 = 1                  |
|                     | TOA2OFF                 | Input  | P00 = Setting not required | PM00 = Setting not required | PMC00 = 1                  | PFCE00 = 1                   | PFC00 = 0                  |
|                     | INTP00                  | Input  | P00 = Setting not required | PM00 = Setting not required | PMC00 = 1                  | PFCE00 = 1                   | PFC00 = 0                  |
| P01                 | TOA21                   | Output | P01 = Setting not required | PM01 = Setting not required | PMC01 = 1                  | PFCE01 = 0                   | PFC01 = 0                  |
|                     | TIA21                   | Input  | P01 = Setting not required | PM01 = Setting not required | PMC01 = 1                  | PFCE01 = 0                   | PFC01 = 1                  |
|                     | INTP01                  | Input  | P01 = Setting not required | PM01 = Setting not required | PMC01 = 1                  | PFCE01 = 1                   | PFC01 = 0                  |
| P02 <sup>Note</sup> | TOA30 <sup>Note</sup>   | Output | P02 = Setting not required | PM02 = Setting not required | PMC02 = 1                  | PFCE02 = 0                   | PFC02 = 0                  |
|                     | TIA30 <sup>Note</sup>   | Input  | P02 = Setting not required | PM02 = Setting not required | PMC02 = 1                  | PFCE02 = 0                   | PFC02 = 1                  |
|                     | TOA3OFF <sup>Note</sup> | Input  | P02 = Setting not required | PM02 = Setting not required | PMC02 = 1                  | PFCE02 = 1                   | PFC02 = 0                  |
|                     | INTP02 <sup>Note</sup>  | Input  | P02 = Setting not required | PM02 = Setting not required | PMC02 = 1                  | PFCE02 = 1                   | PFC02 = 0                  |
| P03 <sup>Note</sup> | TOA31 <sup>Note</sup>   | Output | P03 = Setting not required | PM03 = Setting not required | PMC03 = 1                  | PFCE03 = 0                   | PFC03 = 0                  |
|                     | TIA31 <sup>Note</sup>   | Input  | P03 = Setting not required | PM03 = Setting not required | PMC03 = 1                  | PFCE03 = 0                   | PFC03 = 1                  |
|                     | INTP03 <sup>Note</sup>  | Input  | P03 = Setting not required | PM03 = Setting not required | PMC03 = 1                  | PFCE03 = 1                   | PFC03 = 0                  |
| P04 <sup>Note</sup> | TECR0 <sup>Note</sup>   | Input  | P04 = Setting not required | PM04 = Setting not required | PMC04 = 1                  | PFCE04 = 0                   | PFC04 = 0                  |
|                     | TIT00 <sup>Note</sup>   | Input  | P04 = Setting not required | PM04 = Setting not required | PMC04 = 1                  | PFCE04 = 0                   | PFC04 = 0                  |
|                     | TOT00 <sup>Note</sup>   | Output | P04 = Setting not required | PM04 = Setting not required | PMC04 = 1                  | PFCE04 = 0                   | PFC04 = 1                  |
|                     | INTP04 <sup>Note</sup>  | Input  | P04 = Setting not required | PM04 = Setting not required | PMC04 = 1                  | PFCE04 = 1                   | PFC04 = 0                  |
| P05 <sup>Note</sup> | TENC00 <sup>Note</sup>  | Input  | P05 = Setting not required | PM05 = Setting not required | PMC05 = 1                  | PFCE05 = 0                   | PFC05 = 0                  |
|                     | EVTT0 <sup>Note</sup>   | Input  | P05 = Setting not required | PM05 = Setting not required | PMC05 = 1                  | PFCE05 = 0                   | PFC05 = 1                  |
|                     | INTP05 <sup>Note</sup>  | Input  | P05 = Setting not required | PM05 = Setting not required | PMC05 = 1                  | PFCE05 = 1                   | PFC05 = 0                  |

**Note** V850E/IG3 only



Table 4-14. Using Port Pin as Alternate-Function Pin (2/8)

| Pin Name              | Alternate Pin            |        | Pnx Bit of Pn Register     | PMnx Bit of PMn Register    | PMCnx Bit of PMCn Register | PFCEnx Bit of PFCEn Register | PFCnx Bit of PFCn Register |
|-----------------------|--------------------------|--------|----------------------------|-----------------------------|----------------------------|------------------------------|----------------------------|
|                       | Name                     | I/O    |                            |                             |                            |                              |                            |
| P06 <sup>Note 1</sup> | TENC01 <sup>Note 1</sup> | Input  | P06 = Setting not required | PM06 = Setting not required | PMC06 = 1                  | PFCE06 = 0                   | PFC06 = 0                  |
|                       | TIT01 <sup>Note 1</sup>  | Input  | P06 = Setting not required | PM06 = Setting not required | PMC06 = 1                  | PFCE06 = 0                   | PFC06 = 0                  |
|                       | TOT01 <sup>Note 1</sup>  | Output | P06 = Setting not required | PM06 = Setting not required | PMC06 = 1                  | PFCE06 = 0                   | PFC06 = 1                  |
|                       | INTP06 <sup>Note 1</sup> | Input  | P06 = Setting not required | PM06 = Setting not required | PMC06 = 1                  | PFCE06 = 1                   | PFC06 = 0                  |
| P07 <sup>Note 1</sup> | INTP07 <sup>Note 1</sup> | Input  | P07 = Setting not required | PM07 = Setting not required | PMC07 = 1                  | –                            | PFC07 = 0                  |
|                       | CLKOUT <sup>Note 2</sup> | Output | P07 = Setting not required | PM07 = Setting not required | PMC07 = 1                  | –                            | PFC07 = 1                  |
| P10                   | TOB0T1                   | Output | P10 = Setting not required | PM10 = Setting not required | PMC10 = 1                  | PFCE10 = 0                   | PFC10 = 0                  |
|                       | TIB01                    | Input  | P10 = Setting not required | PM10 = Setting not required | PMC10 = 1                  | PFCE10 = 0                   | PFC10 = 1                  |
|                       | TOB01                    | Output | P10 = Setting not required | PM10 = Setting not required | PMC10 = 1                  | PFCE10 = 1                   | PFC10 = 0                  |
|                       | A0 <sup>Note 2</sup>     | Output | P10 = Setting not required | PM10 = Setting not required | PMC10 = 1                  | PFCE10 = 1                   | PFC10 = 1                  |
| P11                   | TOB0B1                   | Output | P11 = Setting not required | PM11 = Setting not required | PMC11 = 1                  | PFCE11 = 0                   | PFC11 = 0                  |
|                       | TIB02                    | Input  | P11 = Setting not required | PM11 = Setting not required | PMC11 = 1                  | PFCE11 = 0                   | PFC11 = 1                  |
|                       | TOB02                    | Output | P11 = Setting not required | PM11 = Setting not required | PMC11 = 1                  | PFCE11 = 1                   | PFC11 = 0                  |
|                       | A1 <sup>Note 2</sup>     | Output | P11 = Setting not required | PM11 = Setting not required | PMC11 = 1                  | PFCE11 = 1                   | PFC11 = 1                  |
| P12                   | TOB0T2                   | Output | P12 = Setting not required | PM12 = Setting not required | PMC12 = 1                  | PFCE12 = 0                   | PFC12 = 0                  |
|                       | TIB03                    | Input  | P12 = Setting not required | PM12 = Setting not required | PMC12 = 1                  | PFCE12 = 0                   | PFC12 = 1                  |
|                       | TOB03                    | Output | P12 = Setting not required | PM12 = Setting not required | PMC12 = 1                  | PFCE12 = 1                   | PFC12 = 0                  |
|                       | A2 <sup>Note 2</sup>     | Output | P12 = Setting not required | PM12 = Setting not required | PMC12 = 1                  | PFCE12 = 1                   | PFC12 = 1                  |
| P13                   | TOB0B2                   | Output | P13 = Setting not required | PM13 = Setting not required | PMC13 = 1                  | PFCE13 = 0                   | PFC13 = 0                  |
|                       | TIB00                    | Input  | P13 = Setting not required | PM13 = Setting not required | PMC13 = 1                  | PFCE13 = 0                   | PFC13 = 1                  |
|                       | A3 <sup>Note 2</sup>     | Output | P13 = Setting not required | PM13 = Setting not required | PMC13 = 1                  | PFCE13 = 1                   | PFC13 = 0                  |

- Notes**
1. V850E/IG3 only
  2.  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

Table 4-14. Using Port Pin as Alternate-Function Pin (3/8)

| Pin Name | Alternate Pin      |        | Pnx Bit of Pn Register     | PMnx Bit of PMn Register    | PMCnx Bit of PMCn Register | PFCEnx Bit of PFCEn Register | PFCnx Bit of PFCn Register |
|----------|--------------------|--------|----------------------------|-----------------------------|----------------------------|------------------------------|----------------------------|
|          | Name               | I/O    |                            |                             |                            |                              |                            |
| P14      | TOB0T3             | Output | P14 = Setting not required | PM14 = Setting not required | PMC14 = 1                  | PFCE14 = 0                   | PFC14 = 0                  |
|          | EVTB0              | Input  | P14 = Setting not required | PM14 = Setting not required | PMC14 = 1                  | PFCE14 = 0                   | PFC14 = 1                  |
|          | A4 <sup>Note</sup> | Output | P14 = Setting not required | PM14 = Setting not required | PMC14 = 1                  | PFCE14 = 1                   | PFC14 = 0                  |
| P15      | TOB0B3             | Output | P15 = Setting not required | PM15 = Setting not required | PMC15 = 1                  | PFCE15 = 0                   | PFC15 = 0                  |
|          | TRGB0              | Input  | P15 = Setting not required | PM15 = Setting not required | PMC15 = 1                  | PFCE15 = 0                   | PFC15 = 1                  |
|          | A5 <sup>Note</sup> | Output | P15 = Setting not required | PM15 = Setting not required | PMC15 = 1                  | PFCE15 = 1                   | PFC15 = 0                  |
| P16      | TOB0OFF            | Input  | P16 = Setting not required | PM16 = Setting not required | PMC16 = 1                  | PFCE16 = 0                   | PFC16 = 0                  |
|          | INTP08             | Input  | P16 = Setting not required | PM16 = Setting not required | PMC16 = 1                  | PFCE16 = 0                   | PFC16 = 0                  |
|          | ADTRG0             | Input  | P16 = Setting not required | PM16 = Setting not required | PMC16 = 1                  | PFCE16 = 0                   | PFC16 = 1                  |
|          | INTADT0            | Input  | P16 = Setting not required | PM16 = Setting not required | PMC16 = 1                  | PFCE16 = 0                   | PFC16 = 1                  |
|          | A6 <sup>Note</sup> | Output | P16 = Setting not required | PM16 = Setting not required | PMC16 = 1                  | PFCE16 = 1                   | PFC16 = 0                  |
| P17      | TOB00              | Output | P17 = Setting not required | PM17 = Setting not required | PMC17 = 1                  | PFCE17 = 0                   | PFC17 = 0                  |
|          | INTP09             | Input  | P17 = Setting not required | PM17 = Setting not required | PMC17 = 1                  | PFCE17 = 0                   | PFC17 = 1                  |
|          | A7 <sup>Note</sup> | Output | P17 = Setting not required | PM17 = Setting not required | PMC17 = 1                  | PFCE17 = 1                   | PFC17 = 0                  |
| P20      | TOB1T1             | Output | P20 = Setting not required | PM20 = Setting not required | PMC20 = 1                  | PFCE20 = 0                   | PFC20 = 0                  |
|          | TIB11              | Input  | P20 = Setting not required | PM20 = Setting not required | PMC20 = 1                  | PFCE20 = 0                   | PFC20 = 1                  |
|          | TOB11              | Output | P20 = Setting not required | PM20 = Setting not required | PMC20 = 1                  | PFCE20 = 1                   | PFC20 = 0                  |
| P21      | TOB1B1             | Output | P21 = Setting not required | PM21 = Setting not required | PMC21 = 1                  | PFCE21 = 0                   | PFC21 = 0                  |
|          | TIB12              | Input  | P21 = Setting not required | PM21 = Setting not required | PMC21 = 1                  | PFCE21 = 0                   | PFC21 = 1                  |
|          | TOB12              | Output | P21 = Setting not required | PM21 = Setting not required | PMC21 = 1                  | PFCE21 = 1                   | PFC21 = 0                  |
| P22      | TOB1T2             | Output | P22 = Setting not required | PM22 = Setting not required | PMC22 = 1                  | PFCE22 = 0                   | PFC22 = 0                  |
|          | TIB13              | Input  | P22 = Setting not required | PM22 = Setting not required | PMC22 = 1                  | PFCE22 = 0                   | PFC22 = 1                  |
|          | TOB13              | Output | P22 = Setting not required | PM22 = Setting not required | PMC22 = 1                  | PFCE22 = 1                   | PFC22 = 0                  |

**Note**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

Table 4-14. Using Port Pin as Alternate-Function Pin (4/8)

| Pin Name | Alternate Pin             |        | Pnx Bit of Pn Register     | PMnx Bit of PMn Register    | PMCnx Bit of PMCn Register | PFCEnx Bit of PFCEn Register | PFCnx Bit of PFCn Register |
|----------|---------------------------|--------|----------------------------|-----------------------------|----------------------------|------------------------------|----------------------------|
|          | Name                      | I/O    |                            |                             |                            |                              |                            |
| P23      | TOB1B2                    | Output | P23 = Setting not required | PM23 = Setting not required | PMC23 = 1                  | –                            | PFC23 = 0                  |
|          | TIB10                     | Input  | P23 = Setting not required | PM23 = Setting not required | PMC23 = 1                  | –                            | PFC23 = 1                  |
| P24      | TOB1T3                    | Output | P24 = Setting not required | PM24 = Setting not required | PMC24 = 1                  | –                            | PFC24 = 0                  |
|          | EVTB1                     | Input  | P24 = Setting not required | PM24 = Setting not required | PMC24 = 1                  | –                            | PFC24 = 1                  |
| P25      | TOB1B3                    | Output | P25 = Setting not required | PM25 = Setting not required | PMC25 = 1                  | –                            | PFC25 = 0                  |
|          | TRGB1                     | Input  | P25 = Setting not required | PM25 = Setting not required | PMC25 = 1                  | –                            | PFC25 = 1                  |
| P26      | TOB10                     | Output | P26 = Setting not required | PM26 = Setting not required | PMC26 = 1                  | PFCE26 = 0                   | PFC26 = 0                  |
|          | TOB1OFF                   | Input  | P26 = Setting not required | PM26 = Setting not required | PMC26 = 1                  | PFCE26 = 0                   | PFC26 = 1                  |
|          | INTP10                    | Input  | P26 = Setting not required | PM26 = Setting not required | PMC26 = 1                  | PFCE26 = 0                   | PFC26 = 1                  |
|          | ADTRG1                    | Input  | P26 = Setting not required | PM26 = Setting not required | PMC26 = 1                  | PFCE26 = 1                   | PFC26 = 0                  |
|          | INTADT1                   | Input  | P26 = Setting not required | PM26 = Setting not required | PMC26 = 1                  | PFCE26 = 1                   | PFC26 = 0                  |
| P27      | DMS <sup>Notes 1, 2</sup> | Input  | P27 = Setting not required | PM27 = Setting not required | –                          | –                            | –                          |
| P30      | RXDA1                     | Input  | P30 = Setting not required | PM30 = Setting not required | PMC30 = 1                  | PFCE30 = 0                   | PFC30 = 0                  |
|          | SCL                       | I/O    | P30 = Setting not required | PM30 = Setting not required | PMC30 = 1                  | PFCE30 = 0                   | PFC30 = 1                  |
| P31      | TXDA1                     | Output | P31 = Setting not required | PM31 = Setting not required | PMC31 = 1                  | PFCE31 = 0                   | PFC31 = 0                  |
|          | SDA                       | I/O    | P31 = Setting not required | PM31 = Setting not required | PMC31 = 1                  | PFCE31 = 0                   | PFC31 = 1                  |

**Notes** 1. V850E/IG3 only

2. The P27 pin is also used for on-chip debugging. Switching between on-chip debug function and port function can be set. The following shows the setting method.

| Port 2 Functions                                |  |
|---|--|
| Low-Level Input to $\overline{\text{DRST}}$ Pin | High-Level Input to $\overline{\text{DRST}}$ Pin |
| P27   | DMS  |

Table 4-14. Using Port Pin as Alternate-Function Pin (5/8)

| Pin Name | Alternate Pin                           |        | Pnx Bit of Pn Register     | PMnx Bit of PMn Register    | PMCnx Bit of PMCn Register | PFCEnx Bit of PFCEn Register | PFCnx Bit of PFCn Register |
|----------|---|--------|----------------------------|-----------------------------|----------------------------|------------------------------|----------------------------|
|          | Name                                    | I/O    |                            |                             |                            |                              |                            |
| P32      | SIB1                                    | Input  | P32 = Setting not required | PM32 = Setting not required | PMC32 = 1                  | PFCE32 = 0                   | PFC32 = 0                  |
|          | RXDA2                                   | Input  | P32 = Setting not required | PM32 = Setting not required | PMC32 = 1                  | PFCE32 = 0                   | PFC32 = 1                  |
|          | $\overline{\text{CS1}}$ <sup>Note</sup> | Output | P32 = Setting not required | PM32 = Setting not required | PMC32 = 1                  | PFCE32 = 1                   | PFC32 = 0                  |
| P33      | SOB1                                    | Output | P33 = Setting not required | PM33 = Setting not required | PMC33 = 1                  | –                            | PFC33 = 0                  |
|          | TXDA2                                   | Output | P33 = Setting not required | PM33 = Setting not required | PMC33 = 1                  | –                            | PFC33 = 1                  |
| P34      | $\overline{\text{SCKB1}}$               | I/O    | P34 = Setting not required | PM34 = Setting not required | PMC34 = 1                  | PFCE34 = 0                   | PFC34 = 0                  |
|          | INTP11                                  | Input  | P34 = Setting not required | PM34 = Setting not required | PMC34 = 1                  | PFCE34 = 0                   | PFC34 = 1                  |
|          | $\overline{\text{CS0}}$ <sup>Note</sup> | Output | P34 = Setting not required | PM34 = Setting not required | PMC34 = 1                  | PFCE34 = 1                   | PFC34 = 0                  |
| P35      | SIB2                                    | Input  | P35 = Setting not required | PM35 = Setting not required | PMC35 = 1                  | PFCE35 = 0                   | PFC35 = 0                  |
|          | RXDB                                    | Input  | P35 = Setting not required | PM35 = Setting not required | PMC35 = 1                  | PFCE35 = 0                   | PFC35 = 1                  |
| P36      | SOB2                                    | Output | P36 = Setting not required | PM36 = Setting not required | PMC36 = 1                  | PFCE36 = 0                   | PFC36 = 0                  |
|          | TXDB                                    | Output | P36 = Setting not required | PM36 = Setting not required | PMC36 = 1                  | PFCE36 = 0                   | PFC36 = 1                  |
| P37      | $\overline{\text{SCKB2}}$               | I/O    | P30 = Setting not required | PM37 = Setting not required | PMC37 = 1                  | PFCE37 = 0                   | PFC37 = 0                  |
|          | INTP12                                  | Input  | P30 = Setting not required | PM37 = Setting not required | PMC37 = 1                  | PFCE37 = 0                   | PFC37 = 1                  |
|          | ASTB <sup>Note</sup>                    | Output | P30 = Setting not required | PM37 = Setting not required | PMC37 = 1                  | PFCE37 = 1                   | PFC37 = 0                  |
| P40      | SIB0                                    | Input  | P40 = Setting not required | PM40 = Setting not required | PMC40 = 1                  | –                            | PFC40 = 0                  |
|          | RXDA0                                   | Input  | P40 = Setting not required | PM40 = Setting not required | PMC40 = 1                  | –                            | PFC40 = 1                  |

**Note**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

Table 4-14. Using Port Pin as Alternate-Function Pin (6/8)

| Pin Name | Alternate Pin                              |        | Pnx Bit of Pn Register     | PMnx Bit of PMn Register    | PMCnx Bit of PMCn Register   | PFCEnx Bit of PFCEn Register  | PFCnx Bit of PFCn Register |
|----------|--|--------|----------------------------|-----------------------------|------------------------------|-------------------------------|----------------------------|
|          | Name                                       | I/O    |                            |                             |                              |                               |                            |
| P41      | SOB0                                       | Output | P41 = Setting not required | PM41 = Setting not required | PMC41 = 1                    | PFCE41 = 0                    | PFC41 = 0                  |
|          | TXDA0                                      | Output | P41 = Setting not required | PM41 = Setting not required | PMC41 = 1                    | PFCE41 = 0                    | PFC41 = 1                  |
|          | DCK <sup>Notes 1, 2</sup>                  | Input  | P41 = Setting not required | PM41 = Setting not required | PMC41 = Setting not required | PFCE41 = Setting not required | PFC41 = Setting required   |
| P42      | $\overline{\text{SCKB0}}$                  | I/O    | P42 = Setting not required | PM42 = Setting not required | PMC42 = 1                    | PFCE42 = 0                    | PFC42 = 0                  |
|          | INTP13                                     | Input  | P42 = Setting not required | PM42 = Setting not required | PMC42 = 1                    | PFCE42 = 0                    | PFC42 = 1                  |
|          | DDI <sup>Notes 1, 2</sup>                  | Input  | P42 = Setting not required | PM42 = Setting not required | PMC42 = Setting not required | PFCE42 = Setting not required | PFC42 = Setting required   |
| P43      | TECR1                                      | Input  | P43 = Setting not required | PM43 = Setting not required | PMC43 = 1                    | PFCE43 = 0                    | PFC43 = 0                  |
|          | TIT10                                      | Input  | P43 = Setting not required | PM43 = Setting not required | PMC43 = 1                    | PFCE43 = 0                    | PFC43 = 0                  |
|          | TOT10                                      | Output | P43 = Setting not required | PM43 = Setting not required | PMC43 = 1                    | PFCE43 = 0                    | PFC43 = 1                  |
|          | INTP14                                     | Input  | P43 = Setting not required | PM43 = Setting not required | PMC43 = 1                    | PFCE43 = 1                    | PFC43 = 0                  |
| P44      | TENC10                                     | Input  | P44 = Setting not required | PM44 = Setting not required | PMC44 = 1                    | PFCE44 = 0                    | PFC44 = 0                  |
|          | EVTT1                                      | Input  | P44 = Setting not required | PM44 = Setting not required | PMC44 = 1                    | PFCE44 = 0                    | PFC44 = 1                  |
|          | INTP15                                     | Input  | P44 = Setting not required | PM44 = Setting not required | PMC44 = 1                    | PFCE44 = 1                    | PFC44 = 0                  |
|          | $\overline{\text{WAIT}}$ <sup>Note 3</sup> | Input  | P44 = Setting not required | PM44 = Setting not required | PMC44 = 1                    | PFCE44 = 1                    | PFC44 = 1                  |

**Notes** 1. V850E/IG3 only

2. The P41 and P42 pins are also used for on-chip debugging. Switching between on-chip debug function and port function be set by using the  $\overline{\text{DRST}}$  pin level. The following shows the setting method.

| Port 4 Functions                                |  |
|---|--|
| Low-Level Input to $\overline{\text{DRST}}$ Pin | High-Level Input to $\overline{\text{DRST}}$ Pin |
| P41/SOB0/TXDA0                                  | DCK  |
| P42/ $\overline{\text{SCKB0}}$ /INTP13          | DDI  |

3.  $\mu\text{PD70F3454GC-8EA-A}$ , 70F3454F1-DA9-A only

Table 4-14. Using Port Pin as Alternate-Function Pin (7/8)

| Pin Name              | Alternate Pin           |        | Pnx Bit of Pn Register     | PMnx Bit of PMn Register    | PMCnx Bit of PMCn Register | PFCEnx Bit of PFCEn Register | PFCnx Bit of PFCn Register |
|-----------------------|-------------------------|--------|----------------------------|-----------------------------|----------------------------|------------------------------|----------------------------|
|                       | Name                    | I/O    |                            |                             |                            |                              |                            |
| P45                   | TENC11                  | Input  | P45 = Setting not required | PM45 = Setting not required | PMC45 = 1                  | PFCE45 = 0                   | PFC45 = 0                  |
|                       | TIT11                   | Input  | P45 = Setting not required | PM45 = Setting not required | PMC45 = 1                  | PFCE45 = 0                   | PFC45 = 0                  |
|                       | TOT11                   | Output | P45 = Setting not required | PM45 = Setting not required | PMC45 = 1                  | PFCE45 = 0                   | PFC45 = 1                  |
|                       | INTP16                  | Input  | P45 = Setting not required | PM45 = Setting not required | PMC45 = 1                  | PFCE45 = 1                   | PFC45 = 0                  |
|                       | WR1 <sup>Note 1</sup>   | Output | P45 = Setting not required | PM45 = Setting not required | PMC45 = 1                  | PFCE45 = 1                   | PFC45 = 1                  |
| P46                   | TOA40                   | Output | P46 = Setting not required | PM46 = Setting not required | PMC46 = 1                  | PFCE46 = 0                   | PFC46 = 0                  |
|                       | TIA40                   | Input  | P46 = Setting not required | PM46 = Setting not required | PMC46 = 1                  | PFCE46 = 0                   | PFC46 = 1                  |
|                       | INTP17                  | Input  | P46 = Setting not required | PM46 = Setting not required | PMC46 = 1                  | PFCE46 = 1                   | PFC46 = 0                  |
|                       | WR0 <sup>Note 1</sup>   | Output | P46 = Setting not required | PM46 = Setting not required | PMC46 = 1                  | PFCE46 = 1                   | PFC46 = 1                  |
| P47                   | TOA41                   | Output | P47 = Setting not required | PM47 = Setting not required | PMC47 = 1                  | PFCE47 = 0                   | PFC47 = 0                  |
|                       | TIA41                   | Input  | P47 = Setting not required | PM47 = Setting not required | PMC47 = 1                  | PFCE47 = 0                   | PFC47 = 1                  |
|                       | INTP18                  | Input  | P47 = Setting not required | PM47 = Setting not required | PMC47 = 1                  | PFCE47 = 1                   | PFC47 = 0                  |
|                       | RD <sup>Note 1</sup>    | Output | P47 = Setting not required | PM47 = Setting not required | PMC47 = 1                  | PFCE47 = 1                   | PFC47 = 1                  |
| P70                   | ANI20                   | Input  | P70 = Setting not required | –                           | PMC70 = 1                  | –                            | –                          |
| P71                   | ANI21                   | Input  | P71 = Setting not required | –                           | PMC71 = 1                  | –                            | –                          |
| P72                   | ANI22                   | Input  | P72 = Setting not required | –                           | PMC72 = 1                  | –                            | –                          |
| P73                   | ANI23                   | Input  | P73 = Setting not required | –                           | PMC73 = 1                  | –                            | –                          |
| P74 <sup>Note 2</sup> | ANI24 <sup>Note 2</sup> | Input  | P74 = Setting not required | –                           | PMC74 = 1                  | –                            | –                          |
| P75 <sup>Note 2</sup> | ANI25 <sup>Note 2</sup> | Input  | P75 = Setting not required | –                           | PMC75 = 1                  | –                            | –                          |
| P76 <sup>Note 2</sup> | ANI26 <sup>Note 2</sup> | Input  | P76 = Setting not required | –                           | PMC76 = 1                  | –                            | –                          |
| P77 <sup>Note 2</sup> | ANI27 <sup>Note 2</sup> | Input  | P77 = Setting not required | –                           | PMC77 = 1                  | –                            | –                          |

**Notes** 1.  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

2. V850E/IG3 only

**Table 4-14. Using Port Pin as Alternate-Function Pin (8/8)**

| Pin Name                | Alternate Pin           |       | Pnx Bit of Pn Register       | PMnx Bit of PMn Register      | PMCnx Bit of PMCn Register    | PFCEnx Bit of PFCEn Register | PFCnx Bit of PFCn Register |
|-------------------------|-------------------------|-------|------------------------------|-------------------------------|-------------------------------|------------------------------|----------------------------|
|                         | Name                    | I/O   |                              |                               |                               |                              |                            |
| PDL0                    | AD0 <sup>Note 1</sup>   | I/O   | PDL0 = Setting not required  | PMDL0 = Setting not required  | PMCDL0 = 1                    | –                            | –                          |
| PDL1                    | AD1 <sup>Note 1</sup>   | I/O   | PDL1 = Setting not required  | PMDL1 = Setting not required  | PMCDL1 = 1                    | –                            | –                          |
| PDL2                    | AD2 <sup>Note 1</sup>   | I/O   | PDL2 = Setting not required  | PMDL2 = Setting not required  | PMCDL2 = 1                    | –                            | –                          |
| PDL3                    | AD3 <sup>Note 1</sup>   | I/O   | PDL3 = Setting not required  | PMDL3 = Setting not required  | PMCDL3 = 1                    | –                            | –                          |
| PDL4                    | AD4 <sup>Note 1</sup>   | I/O   | PDL4 = Setting not required  | PMDL4 = Setting not required  | PMCDL4 = 1                    | –                            | –                          |
| PDL5                    | AD5 <sup>Note 1</sup>   | I/O   | PDL5 = Setting not required  | PMDL5 = Setting not required  | PMCDL5 = 1                    | –                            | –                          |
|                         | FLMD1 <sup>Note 2</sup> | Input | PDL5 = Setting not required  | PMDL5 = Setting not required  | PMCDL5 = Setting not required | –                            | –                          |
| PDL6                    | AD6 <sup>Note 1</sup>   | I/O   | PDL6 = Setting not required  | PMDL6 = Setting not required  | PMCDL6 = 1                    | –                            | –                          |
| PDL7                    | AD7 <sup>Note 1</sup>   | I/O   | PDL7 = Setting not required  | PMDL7 = Setting not required  | PMCDL7 = 1                    | –                            | –                          |
| PDL8                    | AD8 <sup>Note 1</sup>   | I/O   | PDL8 = Setting not required  | PMDL8 = Setting not required  | PMCDL8 = 1                    | –                            | –                          |
| PDL9                    | AD9 <sup>Note 1</sup>   | I/O   | PDL9 = Setting not required  | PMDL9 = Setting not required  | PMCDL9 = 1                    | –                            | –                          |
| PDL10 <sup>Note 3</sup> | AD10 <sup>Note 1</sup>  | I/O   | PDL10 = Setting not required | PMDL10 = Setting not required | PMCDL10 = 1                   | –                            | –                          |
| PDL11 <sup>Note 3</sup> | AD11 <sup>Note 1</sup>  | I/O   | PDL11 = Setting not required | PMDL11 = Setting not required | PMCDL11 = 1                   | –                            | –                          |
| PDL12 <sup>Note 3</sup> | AD12 <sup>Note 1</sup>  | I/O   | PDL12 = Setting not required | PMDL12 = Setting not required | PMCDL12 = 1                   | –                            | –                          |
| PDL13 <sup>Note 3</sup> | AD13 <sup>Note 1</sup>  | I/O   | PDL13 = Setting not required | PMDL13 = Setting not required | PMCDL13 = 1                   | –                            | –                          |
| PDL14 <sup>Note 3</sup> | AD14 <sup>Note 1</sup>  | I/O   | PDL14 = Setting not required | PMDL14 = Setting not required | PMCDL14 = 1                   | –                            | –                          |
| PDL15 <sup>Note 3</sup> | AD15 <sup>Note 1</sup>  | I/O   | PDL15 = Setting not required | PMDL15 = Setting not required | PMCDL15 = 1                   | –                            | –                          |

**Notes** 1.  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

2. The PDL5 pin is also used for a pin (FLMD1) to be set in the flash programming mode. This pin does not need to be set in the PDL register. For details, see **CHAPTER 27 FLASH MEMORY**.

3. V850E/IG3 only

4.6 Noise Eliminator

A timing controller used to secure the noise elimination time is provided for the following pins. Input signals that change within the noise elimination time are not internally acknowledged.

Table 4-15. Noise Eliminator (1/2)

| Target Pin   | Filter Type                                       | Noise Elimination Width | Sampling Clock |  |
|--|---|-------------------------|----------------|--|
| RESET  | Analog filter                                     | Several 10 ns           | -              |  |
| DRST <sup>Note 1</sup>   |   |                         |                |  |
| FLMD0  |   |                         |                |  |
| P00/TOA20/TIA20/TOA2OFF/INTP00   | TIA20   | Digital filter          | 2, 3 clocks    | f <sub>xx</sub> , f <sub>xx</sub> /4 selectable  |
|  | TOA2OFF   | Analog filter           | Several 10 ns  | -  |
|  | INTP00  |                         |                |  |
| P01/TOA21/TIA21/INTP01   | TIA21   | Digital filter          | 2, 3 clocks    | f <sub>xx</sub> , f <sub>xx</sub> /4 selectable  |
|  | INTP01  | Analog filter           | Several 10 ns  | -  |
| P02 <sup>Note 1</sup> /TOA30 <sup>Note 1</sup> /TIA30 <sup>Note 1</sup> /TOA3OFF <sup>Note 1</sup> /INTP02 <sup>Note 1</sup> | TIA30 <sup>Note 1</sup>                           | Digital filter          | 2, 3 clocks    | f <sub>xx</sub> , f <sub>xx</sub> /4 selectable  |
|  | TOA3OFF <sup>Note 1</sup>                         | Analog filter           | Several 10 ns  | -  |
|  | INTP02 <sup>Note 1</sup>                          |                         |                |  |
| P03 <sup>Note 1</sup> /TOA31 <sup>Note 1</sup> /TIA31 <sup>Note 1</sup> /INTP03 <sup>Note 1</sup>                            | TIA31 <sup>Note 1</sup>                           | Digital filter          | 2, 3 clocks    | f <sub>xx</sub> , f <sub>xx</sub> /4 selectable  |
|  | INTP03 <sup>Note 1</sup>                          | Analog filter           | Several 10 ns  | -  |
| P04 <sup>Note 1</sup> /TECR0 <sup>Note 1</sup> /TIT00 <sup>Note 1</sup> /TOT00 <sup>Note 1</sup> /INTP04 <sup>Note 1</sup>   | TECR0 <sup>Note 1</sup> /TIT00 <sup>Note 1</sup>  | Digital filter          | 2, 3 clocks    | f <sub>xx</sub> , f <sub>xx</sub> /4, f <sub>xx</sub> /8, f <sub>xx</sub> /16, f <sub>xx</sub> /32, f <sub>xx</sub> /64 selectable |
|  | INTP04 <sup>Note 1</sup>                          | Analog filter           | Several 10 ns  | -  |
| P05 <sup>Note 1</sup> /TENC00 <sup>Note 1</sup> /EVTT0 <sup>Note 1</sup> /INTP05 <sup>Note 1</sup>                           | TENC00 <sup>Note 1</sup>                          | Digital filter          | 2, 3 clocks    | f <sub>xx</sub> , f <sub>xx</sub> /4, f <sub>xx</sub> /8, f <sub>xx</sub> /16, f <sub>xx</sub> /32, f <sub>xx</sub> /64 selectable |
|  | EVTT0 <sup>Note 1</sup>                           |                         |                |  |
|  | INTP05 <sup>Note 1</sup>                          | Analog filter           | Several 10 ns  | -  |
| P06 <sup>Note 1</sup> /TENC01 <sup>Note 1</sup> /TIT01 <sup>Note 1</sup> /TOT01 <sup>Note 1</sup> /INTP06 <sup>Note 1</sup>  | TENC01 <sup>Note 1</sup> /TIT01 <sup>Note 1</sup> | Digital filter          | 2, 3 clocks    | f <sub>xx</sub> , f <sub>xx</sub> /4, f <sub>xx</sub> /8, f <sub>xx</sub> /16, f <sub>xx</sub> /32, f <sub>xx</sub> /64 selectable |
|  | INTP06 <sup>Note 1</sup>                          | Analog filter           | Several 10 ns  | -  |
| P07 <sup>Note 1</sup> /INTP07 <sup>Note 1</sup> /CLKOUT <sup>Note 2</sup>  | INTP07 <sup>Note 1</sup>                          |                         |                |  |
| P10/TOB0T1/TIB01/TOB01/A0 <sup>Note 2</sup>  | TIB01   | Digital filter          | 2, 3 clocks    | f <sub>xx</sub> /4   |
| P11/TOB0B1/TIB02/TOB02/A1 <sup>Note 2</sup>  | TIB02   |                         |                |  |
| P12/TOB0T2/TIB03/TOB03/A2 <sup>Note 2</sup>  | TIB03   |                         |                |  |
| P13/TOB0B2/TIB00/A3 <sup>Note 2</sup>  | TIB00   |                         |                |  |
| P14/TOB0T3/EVTB0/A4 <sup>Note 2</sup>  | EVTB0   |                         |                |  |
| P15/TOB0B3/TRGB0/A5 <sup>Note 2</sup>  | TRGB0   |                         |                |  |

- Notes 1. V850E/IG3 only
- 2.  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

- Cautions 1. The maskable interrupt pins can be used to release the standby mode. For details, see CHAPTER 21 STANDBY FUNCTION.
- 2. The digital filter uses clock sampling and therefore cannot acknowledge an input signal when the peripheral clock (f<sub>xx</sub>) is stopped (STOP or IDLE mode).
- 3. The noise eliminator is valid only in the alternate-function mode.



Table 4-15. Noise Eliminator (2/2)

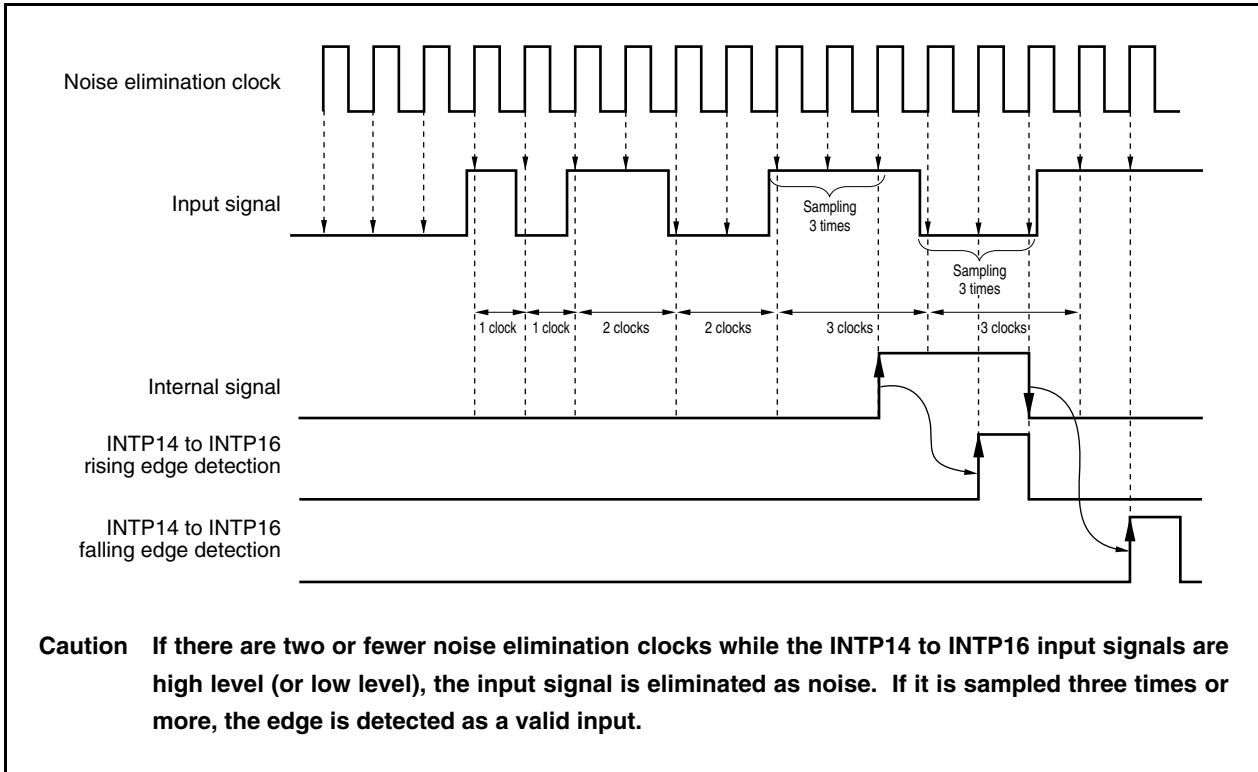
| Target Pin  |                  | Filter Type    | Noise Elimination Width | Sampling Clock   |
|---|------------------|----------------|-------------------------|--|
| P16/TOB0OFF/INTP08/ADTRG0/INTADT0/<br>A6 <sup>Note 1</sup>                        | TOB0OFF          | Analog filter  | Several 10 ns           | -  |
|   | INTP08           |                |                         |  |
|   | ADTRG0           |                |                         |  |
|   | INTADT0          |                |                         |  |
| P17/TOB00/INTP09/A7 <sup>Note 1</sup>   | INTP09           |                |                         |  |
| P20/TOB1T1/TIB11/TOB11  | TIB11            | Digital filter | 2, 3 clocks             | f <sub>xx</sub> /4   |
| P21/TOB1B1/TIB12/TOB12  | TIB12            |                |                         |  |
| P22/TOB1T2/TIB13/TOB13  | TIB13            |                |                         |  |
| P23/TOB1B2/TIB10  | TIB10            |                |                         |  |
| P24/TOB1T3/EVTB1  | EVTB1            |                |                         |  |
| P25/TOB1B3/TRGB1  | TRGB1            |                |                         |  |
| P26/TOB10/TOB1OFF/INTP10/ADTRG1/<br>INTADT1                                       | TOB1OFF          | Analog filter  | Several 10 ns           | -  |
|   | INTP10           |                |                         |  |
|   | ADTRG1           |                |                         |  |
|   | INTADT1          |                |                         |  |
| P34/ $\overline{\text{SCKB1}}$ /INTP11/ $\overline{\text{CS0}}$ <sup>Note 1</sup> | INTP11           |                |                         |  |
| P37/ $\overline{\text{SCKB2}}$ /INTP12/ASTB <sup>Note 1</sup>                     | INTP12           |                |                         |  |
| P42/ $\overline{\text{SCKB0}}$ /INTP13/DDI <sup>Note 2</sup>                      | INTP13           |                |                         |  |
| P43/TECR1/TIT10/TOT10/INTP14  | TECR1/<br>TIT10  | Digital filter | 2, 3 clocks             | f <sub>xx</sub> , f <sub>xx</sub> /4, f <sub>xx</sub> /8, f <sub>xx</sub> /16, f <sub>xx</sub> /32,<br>f <sub>xx</sub> /64 selectable        |
|   | INTP14           |                |                         | f <sub>xx</sub> /4, f <sub>xx</sub> /16, f <sub>xx</sub> /64, f <sub>xx</sub> /128,<br>f <sub>xx</sub> /256, f <sub>xx</sub> /512 selectable |
| P44/TENC10/EVTT1/INTP15/ $\overline{\text{WAIT}}$ <sup>Note 1</sup>               | TENC10           |                |                         | f <sub>xx</sub> , f <sub>xx</sub> /4, f <sub>xx</sub> /8, f <sub>xx</sub> /16, f <sub>xx</sub> /32,<br>f <sub>xx</sub> /64 selectable        |
|   | EVTT1            |                |                         |  |
|   | INTP15           |                |                         | f <sub>xx</sub> /4, f <sub>xx</sub> /16, f <sub>xx</sub> /64, f <sub>xx</sub> /128,<br>f <sub>xx</sub> /256, f <sub>xx</sub> /512 selectable |
| P45/TENC11/TIT11/TOT11/INTP16/ $\overline{\text{WR1}}$ <sup>Note 1</sup>          | TENC11/<br>TIT11 |                |                         | f <sub>xx</sub> /, f <sub>xx</sub> /4, f <sub>xx</sub> /8, f <sub>xx</sub> /16,<br>f <sub>xx</sub> /32, f <sub>xx</sub> /64 selectable       |
|   | INTP16           |                |                         | f <sub>xx</sub> /4, f <sub>xx</sub> /16, f <sub>xx</sub> /64, f <sub>xx</sub> /128,<br>f <sub>xx</sub> /256, f <sub>xx</sub> /512 selectable |
| P46/TOA40/TIA40/INTP17/ $\overline{\text{WR0}}$ <sup>Note 1</sup>                 | TIA40            |                |                         | f <sub>xx</sub> , f <sub>xx</sub> /4 selectable  |
|   | INTP17           | Analog filter  | Several 10 ns           | -  |
| P47/TOA41/TIA41/INTP18/ $\overline{\text{RD}}$ <sup>Note 1</sup>                  | TIA41            | Digital filter | 2, 3 clocks             | f <sub>xx</sub> , f <sub>xx</sub> /4 selectable  |
|   | INTP18           | Analog filter  | Several 10 ns           | -  |

- Notes 1.**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only  
**2.** V850E/IG3 only

- Cautions 1.** The maskable interrupt pins can be used to release the standby mode. For details, see **CHAPTER 21 STANDBY FUNCTION**.  
**2.** The digital filter uses clock sampling and therefore cannot acknowledge an input signal when the peripheral clock (f<sub>xx</sub>) is stopped (STOP or IDLE mode).  
**3.** The noise eliminator is valid only in the alternate-function mode.

An example of timing of noise elimination by digital filter for INTP14 to INTP16, timer AA input pin, and timer T input pin is shown below.

**Figure 4-4. Example of Noise Elimination Timing**



**(1) Digital noise elimination 0 control register n (INTNFCn)**

The INTNFCn register is used to select the sampling clock that is used to eliminate digital noise on the INTPn pin. If the same level is not detected on this pin three times in sequence using the clock selected by the INTNFCn register, the signal is eliminated as noise.

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

Reset sets this register to 00H.

- Cautions**
1. If the input signal lasts for the duration of 2 or 3 clocks, it is undefined whether the signal is detected as a valid edge or eliminated as noise. So that the signal is actually detected as a valid edge, the same signal level must be input for a duration of 3 clocks or more.
  2. If noise is generated in synchronization with the sampling clock, eliminate the noise by attaching a filter to the input pin.
  3. Noise is not eliminated if the pin is used as a normal input port pin.

After reset: 00H    R/W    Address: INTNFC14 FFFFF310H, INTNFC15 FFFFF312H,  
INTNFC16 FFFFF314H

|                           |          |   |   |   |   |          |          |          |
|---------------------------|----------|---|---|---|---|----------|----------|----------|
|                           | 7        | 6 | 5 | 4 | 3 | 2        | 1        | 0        |
| INTNFCn<br>(n = 14 to 16) | INTNFENn | 0 | 0 | 0 | 0 | INTNFCn2 | INTNFCn1 | INTNFCn0 |

|          |                                      |
|----------|--------------------------------------|
| INTNFENn | Setting of digital noise elimination |
| 0        | <b>Note</b>                          |
| 1        | Enables digital noise elimination    |

|                  |          |          |                          |
|------------------|----------|----------|--------------------------|
| INTNFCn2         | INTNFCn1 | INTNFCn0 | Sampling clock selection |
| 0                | 0        | 0        | fxx/4                    |
| 0                | 0        | 1        | fxx/16                   |
| 0                | 1        | 0        | fxx/64                   |
| 0                | 1        | 1        | fxx/128                  |
| 1                | 0        | 0        | fxx/256                  |
| 1                | 0        | 1        | fxx/512                  |
| Other than above |          |          | Setting prohibited       |

**Note** When using the INTPn pin, be sure to set the INTNFENn bit to 1 (digital noise elimination enabled). At the same time, a sampling clock must be selected. When using an alternate function of the INTPn pin, set the INTNFENn bit to 0.

**(2) Digital noise elimination 1 control register n (TANFCn)**

The TANFCn register is used to select the sampling clock that is used to eliminate digital noise on the TIA<sub>n0</sub> or TIA<sub>n1</sub> pin. If the same level is not detected on these pins three times in sequence using the clock selected by the TANFCn register, the signal is eliminated as noise.

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

Reset sets this register to 00H.

- Cautions**
1. If the input signal lasts for the duration of 2 or 3 clocks, it is undefined whether the signal is detected as a valid edge or eliminated as noise. So that the signal is actually detected as a valid edge, the same signal level must be input for a duration of 3 clocks or more.
  2. If noise is generated in synchronization with the sampling clock, eliminate the noise by attaching a filter to the input pin.
  3. Noise is not eliminated if the pin is used as a normal input port pin.
  4. The noise elimination function starts operating when the TAA<sub>n</sub>CTL0.TAA<sub>n</sub>CE bit is set to 1 (enabling count operations).

After reset: 00H    R/W    Address: TANFC2 FFFFFFFB40H, TANFC3 FFFFFFFB42H<sup>Note</sup>,  
TANFC4 FFFFFFFB44H

|        |                     |   |   |   |   |   |   |         |
|--------|---------------------|---|---|---|---|---|---|---------|
|        | 7                   | 6 | 5 | 4 | 3 | 2 | 1 | 0       |
| TANFCn | TANFEN <sub>n</sub> | 0 | 0 | 0 | 0 | 0 | 0 | TANFCn0 |

[ V850E/IF3  
n = 2, 4 ]

|                     |                                      |
|---------------------|--------------------------------------|
| TANFEN <sub>n</sub> | Setting of digital noise elimination |
| 0                   | Disables digital noise elimination   |
| 1                   | Enables digital noise elimination    |

[ V850E/IG3  
n = 2 to 4 ]

|         |                          |
|---------|--------------------------|
| TANFCn0 | Sampling clock selection |
| 0       | f <sub>xx</sub>          |
| 1       | f <sub>xx</sub> /4       |

**Note** V850E/IG3 only

**(3) Digital noise elimination 2 control register n (TTNFCn)**

The TTNFCn register is used to select the sampling clock that is used to eliminate digital noise on the TITn0, TITn1, EVTTn, TENCn0, TENCn1, or TECRn pin. If the same level is not detected on these pins three times in sequence using the clock selected by the TTNFCn register, the signal is eliminated as noise.

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

Reset sets this register to 00H.

- Cautions**
1. If the input signal lasts for the duration of 2 or 3 clocks, it is undefined whether the signal is detected as a valid edge or eliminated as noise. So that the signal is actually detected as a valid edge, the same signal level must be input for a duration of 3 clocks or more.
  2. If noise is generated in synchronization with the sampling clock, eliminate the noise by attaching a filter to the input pin.
  3. Noise is not eliminated if the pin is used as a normal input port pin.
  4. The noise elimination function starts operating when the TTnCTL0.TTnCE bit is set to 1 (enabling count operations).

|                           |   |                                    |  |                          |   |         |         |         |
|---------------------------|---|------------------------------------|--|--------------------------|---|---------|---------|---------|
| After reset: 00H          |   | R/W                                | Address: TTNFC0 FFFFF5A0H <sup>Note</sup> , TTNFC1 FFFFF5A2H |                          |   |         |         |         |
|                           | 7   | 6                                  | 5  | 4                        | 3 | 2       | 1       | 0       |
| TTNFCn                    | TTNFENn   | 0                                  | 0  | 0                        | 0 | TTNFCn2 | TTNFCn1 | TTNFCn0 |
| ( V850E/IF3<br>n = 1 )    | TTNFENn      Setting of digital noise elimination |                                    |  |                          |   |         |         |         |
| ( V850E/IG3<br>n = 0, 1 ) | 0   | Disables digital noise elimination |  |                          |   |         |         |         |
|                           | 1   | Enables digital noise elimination  |  |                          |   |         |         |         |
|                           | TTNFCn2   | TTNFCn1                            | TTNFCn0  | Sampling clock selection |   |         |         |         |
|                           | 0   | 0                                  | 0  | fxx                      |   |         |         |         |
|                           | 0   | 0                                  | 1  | fxx/4                    |   |         |         |         |
|                           | 0   | 1                                  | 0  | fxx/8                    |   |         |         |         |
|                           | 0   | 1                                  | 1  | fxx/16                   |   |         |         |         |
|                           | 1   | 0                                  | 0  | fxx/32                   |   |         |         |         |
|                           | 1   | 0                                  | 1  | fxx/64                   |   |         |         |         |
|                           | Other than above                                  |                                    |  | Setting prohibited       |   |         |         |         |

**Note** V850E/IG3 only

## 4.7 Cautions

### 4.7.1 Cautions on setting port pins

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

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

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

(2) An on-chip pull-up resistor can only be connected when the pins are in input mode in the port mode, or when the pins function as input pins in the alternate-function mode.

Moreover, for the V850E/IF3, an on-chip pull-up resistor can be connected to the TOB0T1 to TOB0T3, TOB0B1 to TOB0B3, and TOA21 pins, these are output pins in the alternate-function mode, when these pins go into a high-impedance state due to the TOB0OFF, TOA2OFF, and TOB1OFF pins or software processing.

For the V850E/IG3, an on-chip pull-up resistor can be connected to the TOB0T1 to TOB0T3, TOB0B1 to TOB0B3, TOA21, TOB1T1 to TOB1T3, TOB1B1 to TOB1B3, and TOA31 pins, these are output pins in the alternate-function mode, when these pins go into a high-impedance state due to the TOB0OFF, TOA2OFF, TOB1OFF, and TOA3OFF pins or software processing.

Set the on-chip pull-up resistor in the following sequence.

- <1> Set PMcN register.
- <2> Set PMn register.
- <3> Set PU register.

(3) Set the N-ch open-drain in the following sequence.

- Used in port mode

- <1> Set PMcN register.
- <2> Set PFn register.

- Used as output pin in alternate-function mode of I<sup>2</sup>C

- <1> Set PFCn and PFCEn registers.
- <2> Set PFn register.
- <3> Set PMcN register.

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

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

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

<Example> When P20 pin is an output port, P21 to P27 pins are input ports (all pin statuses are high level), and the value of the port latch is 00H, if the output of P20 pin is changed from low level to high level via a bit manipulation instruction, the value of the port latch is FFH.

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

A bit manipulation instruction is executed in the following order in the V850E/IF3 and V850E/IG3.

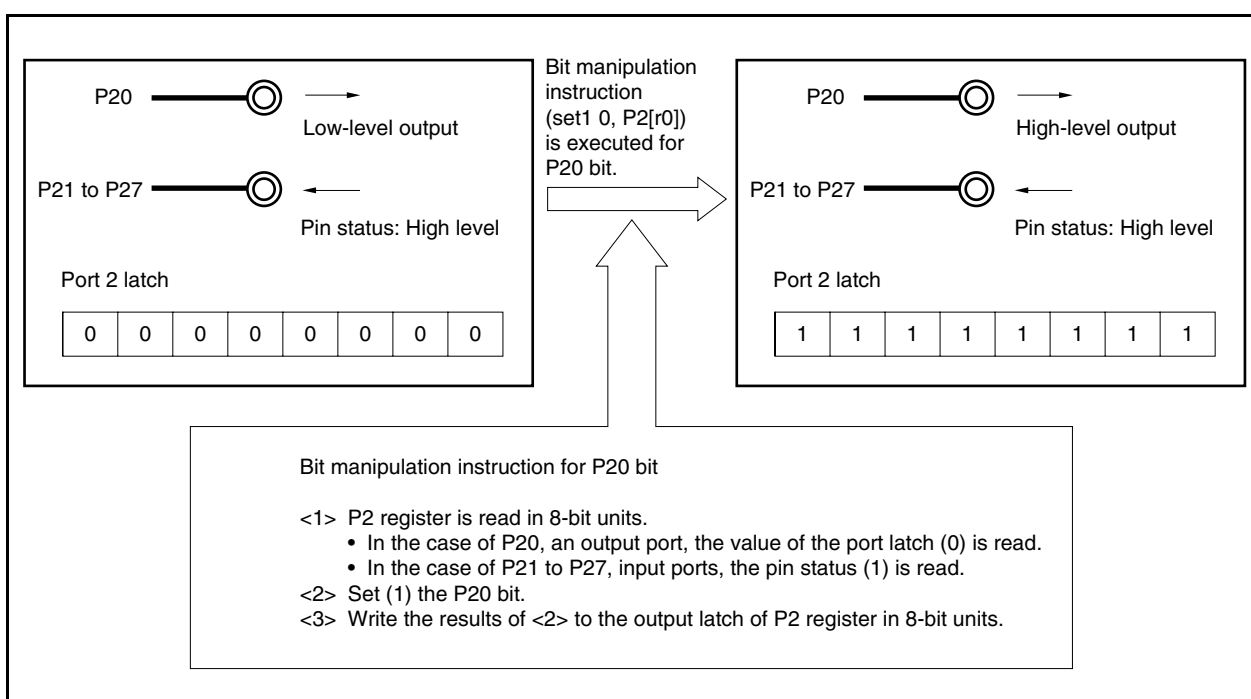
- <1> The Pn register is read in 8-bit units.
- <2> The targeted one bit is manipulated.
- <3> The Pn register is written in 8-bit units.

In step <1>, the value of the output latch (0) of P20 pin, which is an output port, is read, while the pin statuses of P21 to P27 pins, which are input ports, are read. If the pin statuses of P21 to P27 pins are high level at this time, the read value is FEH.

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

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

Figure 4-5. Bit Manipulation Instruction (P20 Pin)



## CHAPTER 5 CLOCK GENERATOR

### 5.1 Overview

The features of clock generator are as follows.

- Oscillator
  - In PLL mode:  $f_x = 4$  to  $8$  MHz ( $f_{xx} = 32$  to  $64$  MHz)
  - In clock-through mode:  $f_x = 4$  to  $8$  MHz ( $f_{xx} = 4$  to  $8$  MHz)
- Multiply ( $\times 8$  fixed) function by PLL (Phase Locked Loop)
  - Clock-through mode/PLL mode selectable
- Internal system clock generation
  - 4 steps ( $f_{xx}$ ,  $f_{xx}/2$ ,  $f_{xx}/4$ ,  $f_{xx}/8$ )
- Peripheral clock generation
- Oscillation stabilization time selection

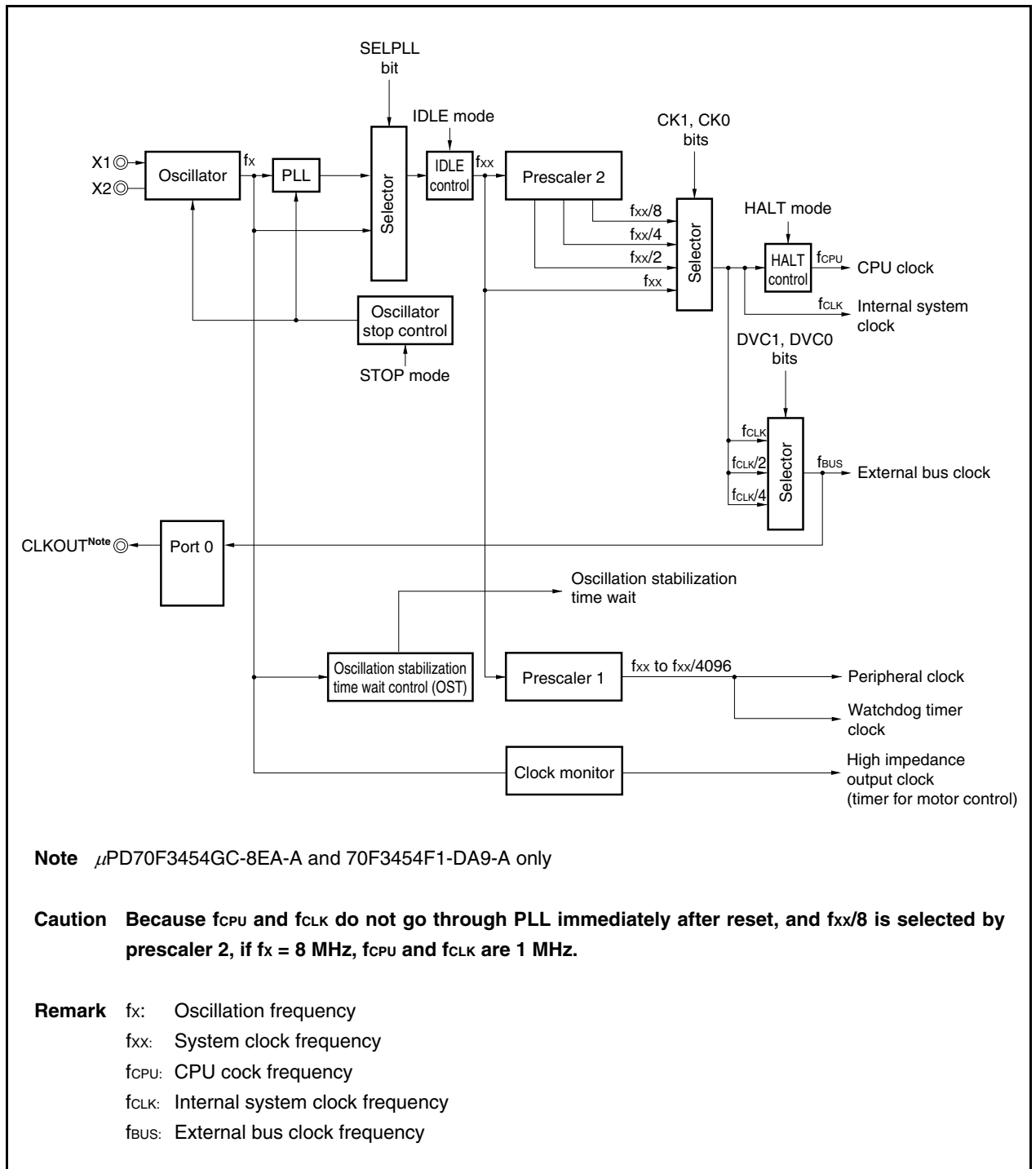
**Caution** The oscillation guaranteed range is 4 to 8 MHz.

**Remark**  $f_x$ : Oscillation frequency  
 $f_{xx}$ : System clock frequency



5.2 Configuration

Figure 5-1. Clock Generator



**Table 5-1. Operation Clock of Each Function Block**

| Function Block            |              | Operation Clock   |
|---------------------------|--------------|---|
| CPU                       |              | $f_{CPU}$ (selected from $f_{XX}$ to $f_{XX}/8$ by PCC register)                  |
| DMA, interrupt controller |              | $f_{CLK}$ (selected from $f_{XX}$ to $f_{XX}/8$ by PCC register)                  |
| TAA                       | TAA0, TAA1   | $f_{XX}$  |
|                           | TAA2 to TAA4 | $f_{XX}/2$  |
| TAB                       |              | $f_{XX}$  |
| TMT                       |              | $f_{XX}/2$  |
| TMM                       |              | $f_{XX}/2$  |
| Watchdog timer            |              | $f_{XX}/1024$   |
| UARTA                     |              | $f_{UCLK}$ (selected from $f_{XX}/2$ to $f_{XX}/4096$ by UAnCTL1 register)        |
| UARTB                     |              | $f_{XX}$  |
| CSIB                      |              | $f_{CCLK}$ (selected from $f_{XX}/4$ to $f_{XX}/256$ by CBnCTL1 register)         |
| I <sup>2</sup> C          |              | $f_{XX}/2$  |
| Bus control function      |              | $f_{BUS}$ (selected from $f_{CLK}/1$ , $f_{CLK}/2$ , $f_{CLK}/4$ by DVC register) |
| A/D converters 0, 1       |              | $f_{AD01}$ (selected from $f_{XX}/2$ to $f_{XX}/4$ by ADnOCKS register)           |
| A/D converter 2           |              | $f_{AD2} = f_{XX}/2$  |

- Remarks 1.**  $f_{CPU}$ : CPU clock frequency  
 $f_{XX}$ : Peripheral clock frequency  
 $f_{CLK}$ : Internal system clock frequency  
 $f_{UCLK}$ : Base clock frequency of UARTA0 to UARTA2  
 $f_{CCLK}$ : Base clock frequency of CSIB0 to CSIB2  
 $f_{AD01}$ : Base clock frequency of A/D converters 0 and 1  
 $f_{AD2}$ : Operating clock frequency of A/D converter 2  
 $f_{BUS}$ : External bus clock frequency
- 2.**  $n = 0, 1$

**(1) Oscillator**

The main resonator oscillates the following frequencies ( $f_x$ ):

- In PLL mode ( $\times 8$  fixed):  $f_x = 4$  to  $8$  MHz ( $f_{xx} = 32$  to  $64$  MHz)
- In clock-through mode:  $f_x = 4$  to  $8$  MHz ( $f_{xx} = 4$  to  $8$  MHz)

**(2) IDLE control**

All functions other than the oscillator, PLL, clock monitor operation, CSIB in slave mode, low-voltage detector (LVI), and power-on-clear circuit (POC) are stopped.

**(3) HALT control**

Only the CPU clock ( $f_{CPU}$ ) is stopped.

**(4) PLL**

This circuit multiplies the clock generated by the oscillator ( $f_x$ ) by 8.

It operates in two modes: clock-through mode in which  $f_x$  is output as is by setting the SELPLL bit of the PLL control register (PLLCTL), and PLL mode in which a multiplied clock is output.

**(5) Prescaler 1**

This prescaler generates the clock ( $f_{xx}$  to  $f_{xx}/4096$ ) to be supplied to on-chip peripheral functions.

**(6) Prescaler 2**

This circuit divides the system clock ( $f_{xx}$ ).

The clock ( $f_{xx}$  to  $f_{xx}/8$ ) to be supplied to the CPU clock ( $f_{CPU}$ ) and internal system clock ( $f_{CLK}$ ) is generated.

**(7) Oscillation stabilization time wait control (OST)**

This unit measures the time from when the clock generated by the oscillator was input until oscillation is stabilized. It also counts the PLL lockup time.

The count clock can be selected from  $2^{14}/f_x$  to  $2^{18}/f_x$ .

**(8) Clock monitor**

The clock monitor samples the clock generated by the oscillator ( $f_x$ ), by using the internal oscillation clock. When it detects stop of oscillation, output of the timer for motor control goes into a high-impedance state (for details, see **CHAPTER 10 MOTOR CONTROL FUNCTION**).

### 5.3 Control Registers

The clock generator is controlled by the following six registers.

- PLL control register (PLLCTL)
- Processor clock control register (PCC)
- Power save control register (PSC)
- Power save mode register (PSMR)
- Oscillation stabilization time select register (OSTS)
- Clock monitor mode register (CLM)

#### (1) PLL control register (PLLCTL)

The PLLCTL register selects CPU operation clock.

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 |   |   |   |        |   |
| PLLCTL           | 7 | 6                             | 5                  | 4 | 3 | 2 | <1>    | 0 |
|                  | 0 | 0                             | 0                  | 0 | 0 | 0 | SELPLL | 1 |
| SELPLL           |   | CPU operation clock selection |                    |   |   |   |        |   |
| 0                |   | Clock-through mode            |                    |   |   |   |        |   |
| 1                |   | PLL mode                      |                    |   |   |   |        |   |

- Cautions**
1. Be sure to set bits 7 to 2 to “0” and set bit 0 to “1”.
  2. Setting the SELPLL bit to 1 is enabled only when the PLL clock frequency is stabilized. If the SELPLL bit is rewritten when the PLL clock frequency is not stabilized (during unlock), 0 is written to the bit. Therefore, be sure to confirm that the PLL mode has been set. Use the following program for reference.

```

_loop:  setl  1, PLLCTL
        tstl  1, PLLCTL
        bz   _loop
(next instruction)
    
```

**(2) Processor clock control register (PCC)**

The PCC register is a special register. Data can be written to this register only in a combination of specific sequences (see **3.4.8 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 | 0 | 0 | 0 | 0 | 0 | 0 | CK1 | CK0 |

| CK1 | CK0 | Clock selection ( $f_{CLK}/f_{CPU}$ ) |
|-----|-----|---------------------------------------|
| 0   | 0   | $f_{XX}$                              |
| 0   | 1   | $f_{XX}/2$                            |
| 1   | 0   | $f_{XX}/4$                            |
| 1   | 1   | $f_{XX}/8$                            |

- Cautions**
1. Be sure to set bits 2 to 7 to “0”.
  2. Set the PCC register to 00H after the PLL mode is selected (PLLCTL.SELPLL bit = 1).

**(3) Power save control register (PSC)**

The PSC register is an 8-bit register that controls the standby function and specifies the standby mode by setting the STB bit. The PSC register is a special register (see 3.4.8 Special registers). Data can be written to this register only in a combination of specific sequences.

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 | 0 | 0 | INTM | 0 | 0 | STB | 0 |

|      |   |
|------|---|
| INTM | Standby mode control <sup>Note 2</sup> by maskable interrupt request (INTxx <sup>Note 1</sup> ) |
| 0    | Standby mode release by INTxx request enabled   |
| 1    | Standby mode release by INTxx request disabled  |

|     |                     |
|-----|---------------------|
| STB | Sets operation mode |
| 0   | Normal mode         |
| 1   | Standby mode        |

**Notes** 1. For details, see Table 20-1 Interrupt Source List.

2. Setting is valid only in the IDLE mode and STOP mode.

**Cautions** 1. Be sure to set bits 0, 2, 3, and 5 to 7 to “0”.

2. Before setting a standby mode by setting the STB bit to 1, be sure to set the PCC register to 03H and then set the STB bit to 1. Otherwise, the standby mode may not be set or released. After releasing the standby mode, change the value of the PCC register to the desired value.

3. To set the IDLE mode or STOP mode, set the PCC register to 03H, and the PSMR.PSM0 bit in that order and then set the STB bit to 1.

**(4) Power save mode register (PSMR)**

The PSMR register is an 8-bit register that controls the operation in the software standby 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: FFFF820H |
|------------------|-----|-------------------|

|      |   |   |   |   |   |   |   |      |
|------|---|---|---|---|---|---|---|------|
|      | 7 | 6 | 5 | 4 | 3 | 2 | 1 | <0>  |
| PSMR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | PSM0 |

|      |  |
|------|--|
| PSM0 | Specifies operation in software standby mode |
| 0    | IDLE mode                                    |
| 1    | STOP mode                                    |

**Cautions**

1. Be sure to set bits 1 to 7 to “0”.
2. The PSM0 bit is valid only when the PSC.STB bit is 1.

**(5) Oscillation stabilization time select register (OSTS)**

The OSTS register selects the oscillation stabilization time until the oscillation stabilizes after the STOP mode is released by interrupt request.

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

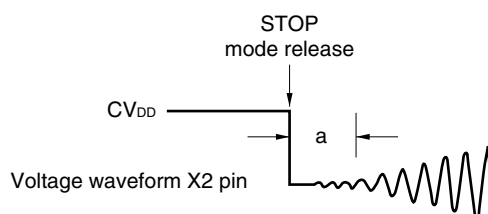
Reset sets this register to 04H.

After reset: 04H    R/W    Address: FFFFF6C0H

|      |   |   |   |   |       |       |       |       |
|------|---|---|---|---|-------|-------|-------|-------|
|      | 7 | 6 | 5 | 4 | 3     | 2     | 1     | 0     |
| OSTS | 0 | 0 | 0 | 0 | OSTS3 | OSTS2 | OSTS1 | OSTS0 |

| OSTS3            | OSTS2 | OSTS1 | OSTS0 | Selection of oscillation stabilization time (fx = 8 MHz) |
|------------------|-------|-------|-------|--|
| 0                | 1     | 0     | 0     | $2^{14}/f_x$ (2.05 ms)                                   |
| 0                | 1     | 0     | 1     | $2^{15}/f_x$ (4.10 ms)                                   |
| 0                | 1     | 1     | 0     | $2^{16}/f_x$ (8.19 ms)                                   |
| 0                | 1     | 1     | 1     | $2^{17}/f_x$ (16.4 ms)                                   |
| 1                | 0     | 0     | 0     | $2^{18}/f_x$ (32.8 ms)                                   |
| Other than above |       |       |       | Setting prohibited                                       |

**Cautions** 1. The wait time does not include the time until the clock oscillation starts (“a” in the figure below) following release of the STOP mode.



2. The default value of the OSTS register after reset is 04H. If an 8 MHz resonator is used, therefore, the oscillation stabilization time is about 2 ms. Half the oscillation stabilization time is consumed by waiting for the lockup of PLL. Therefore, the actual stabilization time of the resonator is about 1 ms. When releasing reset, therefore, make sure that the oscillation stabilization time is secured during the active period of the reset signal. To release the STOP mode by an interrupt input other than a reset signal ( $\overline{\text{RESET}}$  pin input, reset signal (LVIREs) generation by low-voltage detector (LVI), reset signal (POCRES) generation by power-on-clear circuit (POC)), the oscillation stabilization time is determined by the set value of the OSTS register. Therefore, set a time twice as long as that required for the resonator to stabilize to the OSTS register (because half the oscillation stabilization time is the stabilization time of PLL).
3. Be sure to set bits 4 to 7 to “0”.

**Remark** fx: Oscillation frequency



**(6) Clock monitor mode register (CLM)**

The CLM register sets clock monitor operation mode. The CLM register is a special register. It can be written only in a combination of specific sequences (see 3.4.8 **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: FFFFF870H

|     |   |   |   |   |   |   |   |      |
|-----|---|---|---|---|---|---|---|------|
|     | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0    |
| CLM | 0 | 0 | 0 | 0 | 0 | 0 | 0 | CLME |

|      |                                  |
|------|----------------------------------|
| CLME | Clock monitor operation control  |
| 0    | Clock monitor operation disabled |
| 1    | Clock monitor operation enabled  |

- Cautions**
1. The CLME bit is cleared to 0 only after reset.
  2. When the CLME bit = 1, the clock monitor function is forcibly stopped if the following conditions are satisfied.
    - During oscillation stabilization time count after release of STOP mode
  3. When the CLME bit = 1, output of the timer for motor control goes into a high-impedance state if oscillation (fx) stop is detected. See Figure 10-4 for the target timer output.

<R>

## 5.4 PLL Function

### 5.4.1 Overview

The CPU and the operating clock of the peripheral macro can be switched between output of the oscillation frequency multiplied by 8, and clock-through mode.

When PLL function is used: Input clock ( $f_x$ ) = 4 to 8 MHz, output clock ( $f_{xx}$ ) = 32 to 64 MHz

Clock-through mode: Input clock ( $f_x$ ) = 4 to 8 MHz, output clock ( $f_{xx}$ ) = 4 to 8 MHz

### 5.4.2 PLL mode

In the PLL mode, the oscillation frequency ( $f_x$ ) is multiplied by 8 with the PLL to generate a system clock ( $f_{xx}$ ).

In the PLL mode, the clock is input from the oscillator to the PLL. A clock at a stable frequency must be supplied to the internal circuit after the lapse of the lockup time (frequency stabilization time) during which the phase is locked at a specific frequency and oscillation is stabilized. In the V850E/IF3 and V850E/IG3, the lockup time after release of reset is secured automatically.

**Caution** When a resonator of  $f_x = 8$  MHz is used and if the oscillation stabilization time of that resonator must be 3 ms (MAX.), the reset input ( $\overline{\text{RESET}}$  active) width must be 2 ms (MIN.).

### 5.4.3 Clock-through mode

In the clock-through mode, a system clock ( $f_{xx}$ ) of the same frequency as the oscillation frequency ( $f_x$ ) is generated.

## 5.5 Operation

### 5.5.1 Operation of each clock

The following table shows the operation status of each clock.

**Table 5-2. Operation Status of Each Clock**

| Power Save Mode  | Oscillator (fx)     | PLL                 | Internal System Clock (f <sub>CLK</sub> ) | Peripheral Clock (f <sub>xx</sub> to f <sub>xx</sub> /4096) | External Bus Clock (f <sub>BUS</sub> ) <sup>Note 1</sup> | CPU Clock (f <sub>CPU</sub> ) | Watchdog Timer Clock <sup>Note 2</sup> |
|--|---------------------|---------------------|---|---|--|-------------------------------|--|
| Normal operation   | √                   | √                   | √   | √   | √  | √                             | √                                      |
| HALT mode  | √                   | √                   | √   | √   | √  | ×                             | √                                      |
| IDLE mode  | √                   | √                   | ×   | ×   | ×  | ×                             | ×                                      |
| In STOP mode and during oscillation stabilization time count after release of STOP mode                          | × <sup>Note 3</sup> | × <sup>Note 3</sup> | ×   | ×   | ×  | ×                             | ×                                      |
| During $\overline{\text{RESET}}$ pin input <sup>Note 4</sup> and subsequent oscillation stabilization time count | √                   | × → √               | √   | × <sup>Note 5</sup>   | × <sup>Note 6</sup>                                      | √                             | ×                                      |

- Notes**
1.  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only
  2. The peripheral clock (f<sub>xx</sub>/1024) is used as the watchdog timer clock.
  3. Operation continues during on-chip debugging.
  4.  $\overline{\text{RESET}}$  pin input, reset signal (WDTRES) generation by watchdog timer, reset signal (LVIRET) generation by low-voltage detector (LVI), or reset signal (POCRET) generation by power-on-clear circuit (POC)
  5. The output from the prescaler (PRS) is not performed.
  6. The clock is not output from the CLKOUT pin.

**Remark** √: Operating  
 ×: Stopped

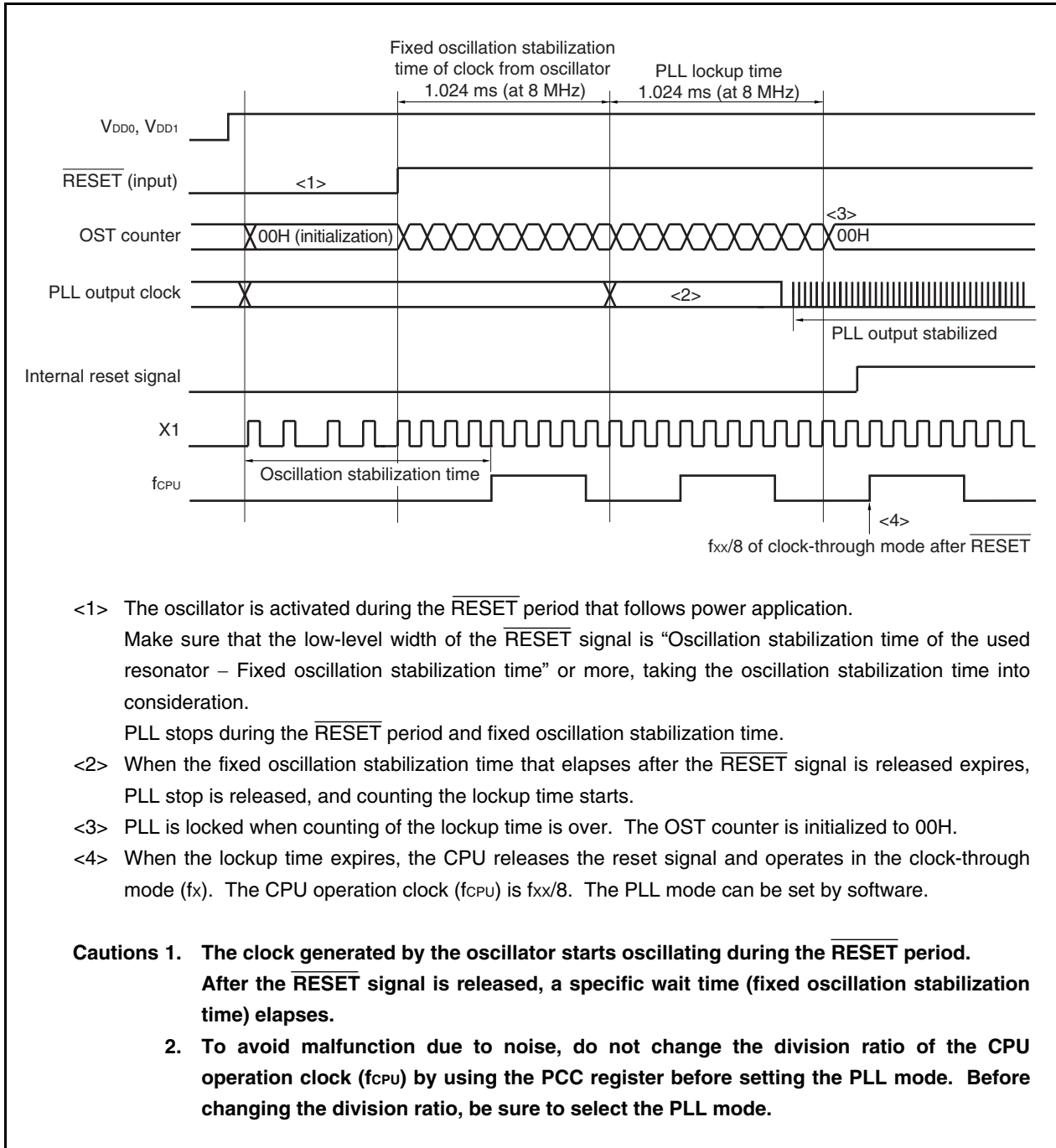
### 5.5.2 Clock output function

The clock output function is used to output the external bus clock (f<sub>BUS</sub>) from the CLKOUT pin and supported only in the  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A.

When the internal system clock (f<sub>CLK</sub>) in Table 5-2 is in the operable status (√), the pin can output the clock. When it is in the stopped status (×), the pin cannot output the clock.

5.5.3 Operation timing

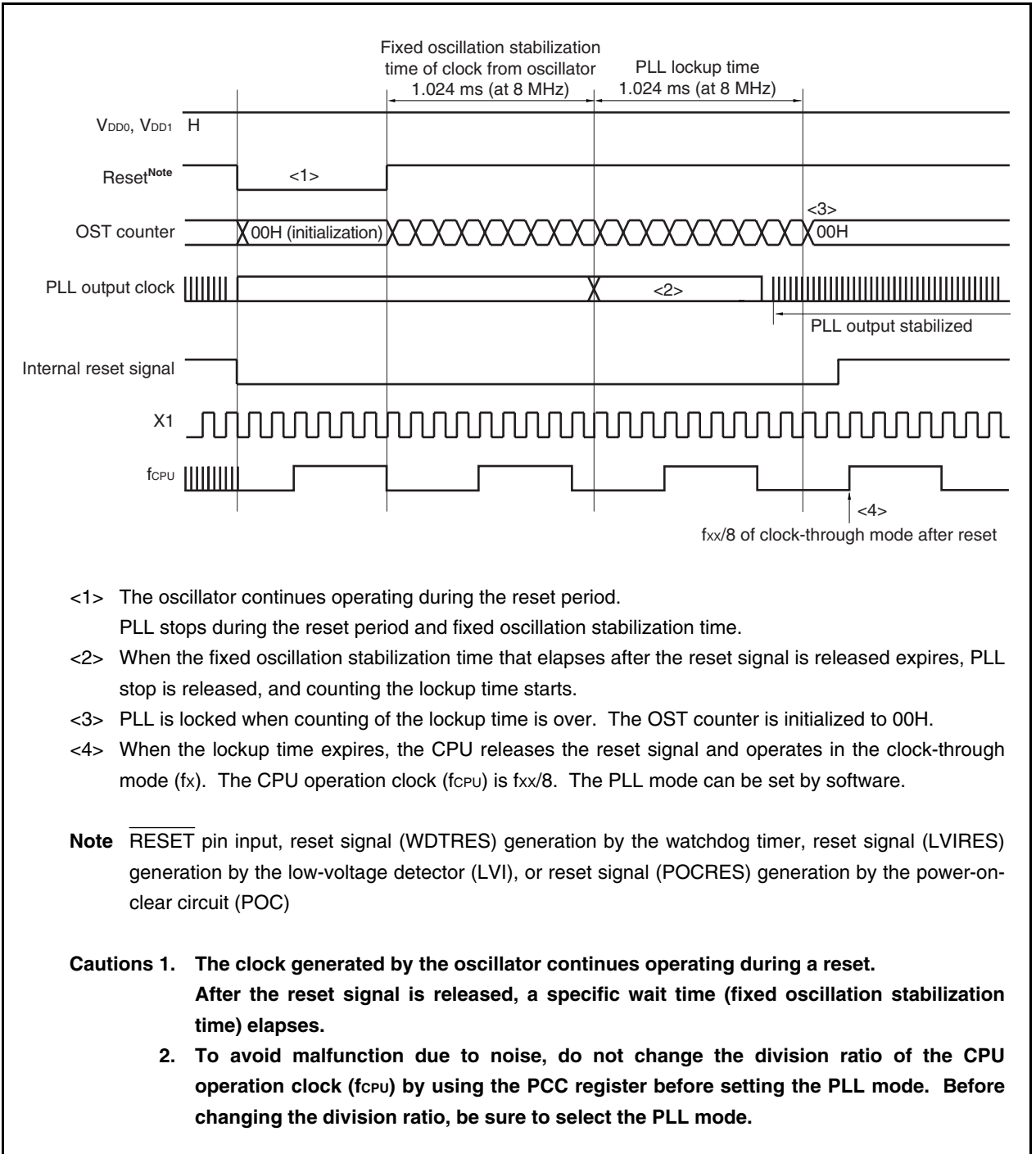
(1) Power on (power-on reset)



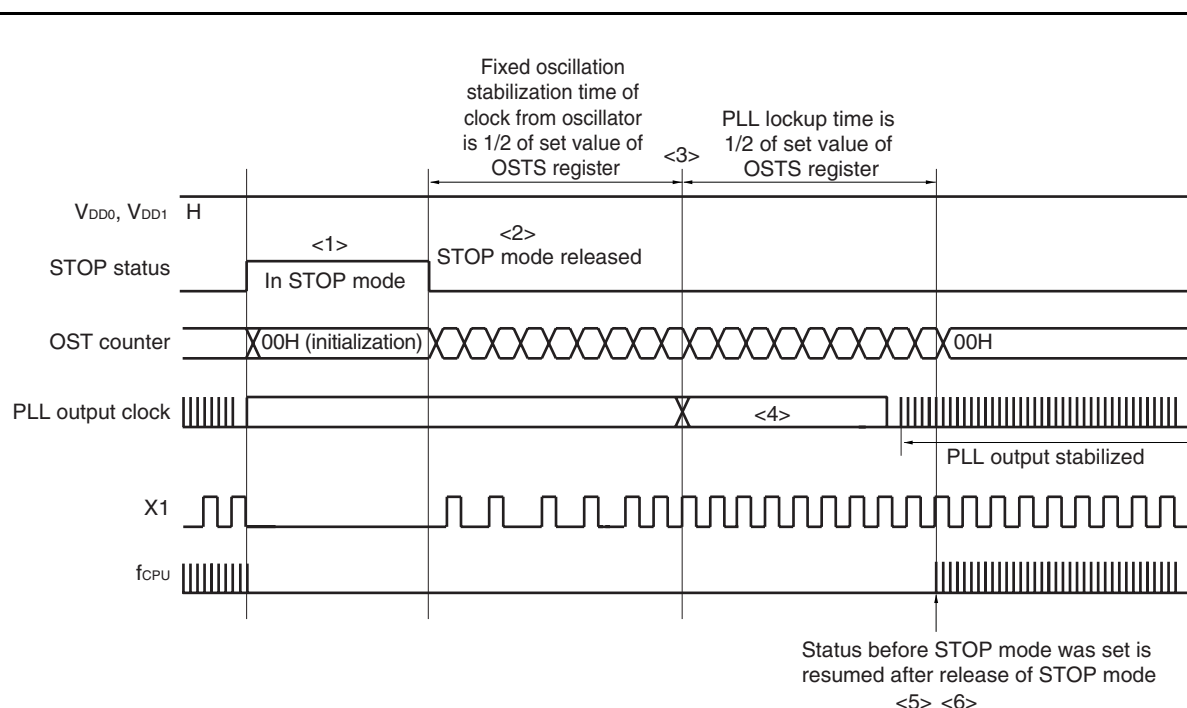
- <1> The oscillator is activated during the  $\overline{\text{RESET}}$  period that follows power application. Make sure that the low-level width of the  $\overline{\text{RESET}}$  signal is “Oscillation stabilization time of the used resonator – Fixed oscillation stabilization time” or more, taking the oscillation stabilization time into consideration. PLL stops during the  $\overline{\text{RESET}}$  period and fixed oscillation stabilization time.
- <2> When the fixed oscillation stabilization time that elapses after the  $\overline{\text{RESET}}$  signal is released expires, PLL stop is released, and counting the lockup time starts.
- <3> PLL is locked when counting of the lockup time is over. The OST counter is initialized to 00H.
- <4> When the lockup time expires, the CPU releases the reset signal and operates in the clock-through mode (f<sub>x</sub>). The CPU operation clock (f<sub>CPU</sub>) is f<sub>x</sub>/8. The PLL mode can be set by software.

- Cautions 1. The clock generated by the oscillator starts oscillating during the  $\overline{\text{RESET}}$  period. After the  $\overline{\text{RESET}}$  signal is released, a specific wait time (fixed oscillation stabilization time) elapses.**
- 2. To avoid malfunction due to noise, do not change the division ratio of the CPU operation clock (f<sub>CPU</sub>) by using the PCC register before setting the PLL mode. Before changing the division ratio, be sure to select the PLL mode.**

(2) Reset input with power on



(3) When releasing STOP mode by interrupt request



- <1> When the STOP mode is set, both the oscillator and PLL stop. At this time, PLL is stopped in the STOP mode. The OST counter is initialized.
- <2> When the STOP mode is released, the oscillator is activated and the OST counter starts counting the oscillation stabilization time. At this time, PLL remains stopped.
- <3> When half the oscillation stabilization time set to the OSTs has elapsed, PLL starts operating. The clock generated by the oscillator must be stabilized before PLL starts operating. The actual oscillation stabilization time is "1/2 the oscillation stabilization time". Take this into consideration when setting a value to the OSTs register.
- <4> After half the oscillation stabilization time has elapsed, the lockup wait time starts. The remaining count time of the OST counter is the lockup wait time.
- <5> When the lockup time of PLL is over, clock supply to the internal circuitry is started. At this time, the status before the STOP mode was set is recovered.
- <6> The operation to be performed when the STOP mode is released by a reset signal ( $\overline{\text{RESET}}$  pin input, reset signal (LVIRES) generation by the low-voltage detector (LVI), reset signal (POCRES) generation by the power-on-clear circuit (POC)) is the same as that in (1) **Power on (power-on reset)** and (2) **Reset input with power on**.

## 5.6 Clock Monitor

### (1) Clock monitor function

The clock monitor samples the clock generated by the oscillator, by using the internal oscillation clock. When it detects stop of oscillation, output of the timer for motor control goes into a high-impedance state (for details, see **CHAPTER 10 MOTOR CONTROL FUNCTION**). The high-impedance state created by the clock monitor function is released by a reset signal ( $\overline{\text{RESET}}$  pin input, reset signal (POCRES) generation by the power-on-clear circuit (POC)) and the pin enters the status after reset.

## CHAPTER 6 16-BIT TIMER/EVENT COUNTER AA (TAA)

Timer AA (TAA) is a 16-bit timer/event counter.

The V850E/IF3 and V850E/IG3 incorporate TAA0 to TAA4.

### 6.1 Overview

The TAA<sub>n</sub> of channels are outlined below (n = 0 to 4).

**Table 6-1. TAA<sub>n</sub> Overview**

| Item   | TAA0                | TAA1                | TAA2   | TAA3                | TAA4   |
|--|---------------------|---------------------|--------|---------------------|--------|
| Clock selection                                | 8 ways              | 8 ways              | 8 ways | 8 ways              | 8 ways |
| Capture trigger input pin                      | None                | None                | 2      | <b>Note 1</b>       | 2      |
| External event count input pin                 | None                | None                | 1      | <b>Note 2</b>       | 1      |
| External trigger input pin                     | None                | None                | 1      | <b>Note 2</b>       | 1      |
| Timer counter                                  | 1                   | 1                   | 1      | 1                   | 1      |
| Capture/compare register                       | 2 <sup>Note 3</sup> | 2 <sup>Note 3</sup> | 2      | 2 <sup>Note 4</sup> | 2      |
| Capture/compare match interrupt request signal | 2 <sup>Note 3</sup> | 2 <sup>Note 3</sup> | 2      | 2 <sup>Note 4</sup> | 2      |
| Overflow interrupt request signal              | 1                   | 1                   | 1      | 1                   | 1      |
| Timer output pin                               | None                | None                | 2      | <b>Note 1</b>       | 2      |

- Notes**
1. V850E/IF3: None  
V850E/IG3: 2
  2. V850E/IF3: None  
V850E/IG3: 1
  3. Compare function only
  4. In the V850E/IF3, compare function only



## 6.2 Functions

The functions of TAA<sub>n</sub> that can be realized differ from one channel to another, as shown in the table below (n = 0 to 4).

**Table 6-2. TAA<sub>n</sub> Functions**

| Function                      | TAA0                | TAA1                | TAA2 | TAA3                | TAA4 |
|-------------------------------|---------------------|---------------------|------|---------------------|------|
| Interval timer                | √                   | √                   | √    | √                   | √    |
| External event counter        | ×                   | ×                   | √    | <b>Note 1</b>       | √    |
| External trigger pulse output | ×                   | ×                   | √    | <b>Note 1</b>       | √    |
| One-shot pulse output         | ×                   | ×                   | √    | <b>Note 1</b>       | √    |
| PWM output                    | ×                   | ×                   | √    | <b>Note 1</b>       | √    |
| Free-running timer            | √ <sup>Note 2</sup> | √ <sup>Note 2</sup> | √    | √ <sup>Note 3</sup> | √    |
| Pulse width measurement       | ×                   | ×                   | √    | <b>Note 1</b>       | √    |
| Timer tuning operation        | √ (TAB0)            | √ (TAB1)            | ×    | ×                   | ×    |

- Notes 1.** V850E/IF3: ×  
V850E/IG3: √
- 2.** Compare function only
- 3.** In the V850E/IF3, compare function only

### 6.3 Configuration

TAA<sub>n</sub> includes the following hardware (n = 0 to 4).

**Table 6-3. Configuration of TAA<sub>n</sub>**

| Item              | Configuration  |
|-------------------|--|
| Timer register    | 16-bit counter × 1   |
| Registers         | TAA <sub>n</sub> capture/compare registers 0, 1 (TAA <sub>n</sub> CCR0, TAA <sub>n</sub> CCR1)<br>TAA <sub>n</sub> counter read buffer register (TAA <sub>n</sub> CNT)<br>CCR0 and CCR1 buffer registers   |
| Timer input       | 6 in total (TIA20, TIA21, TIA30 <sup>Note 1</sup> , TIA31 <sup>Note 1</sup> , TIA40, TIA41 pins) <sup>Notes 2, 3</sup>   |
| Timer output      | 6 in total (TOA20, TOA21, TOA30 <sup>Note 1</sup> , TOA31 <sup>Note 1</sup> , TOA40, TOA41 pins) <sup>Notes 2, 3</sup>   |
| Control registers | TAA <sub>n</sub> control registers 0, 1 (TAA <sub>n</sub> CTL0, TAA <sub>n</sub> CTL1)<br>TAA <sub>m</sub> I/O control registers 0 to 2 (TAA <sub>m</sub> IOC0 to TAA <sub>m</sub> IOC2)<br>TAA <sub>n</sub> option register 0 (TAA <sub>n</sub> OPT0) |

**Notes 1.** V850E/IG3 only

**2.** Not provided for TMP0 and TMP1

**3.** TIA20, TIA30, and TIA40 pins function as capture trigger input signal, external event count input signal, and external trigger input signal, and function alternately (alternate-function) as timer output pins (TOA20, TOA30, TOA40).

TIA21, TIA31, and TIA41 pins function alternately as capture trigger input signal and timer output pins (TOA21, TOA31, TOA41).

**Remark** V850E/IF3: n = 0 to 4, m = 2, 4

V850E/IG3: n = 0 to 4, m = 2 to 4

Figure 6-1. TAA0 Block Diagram

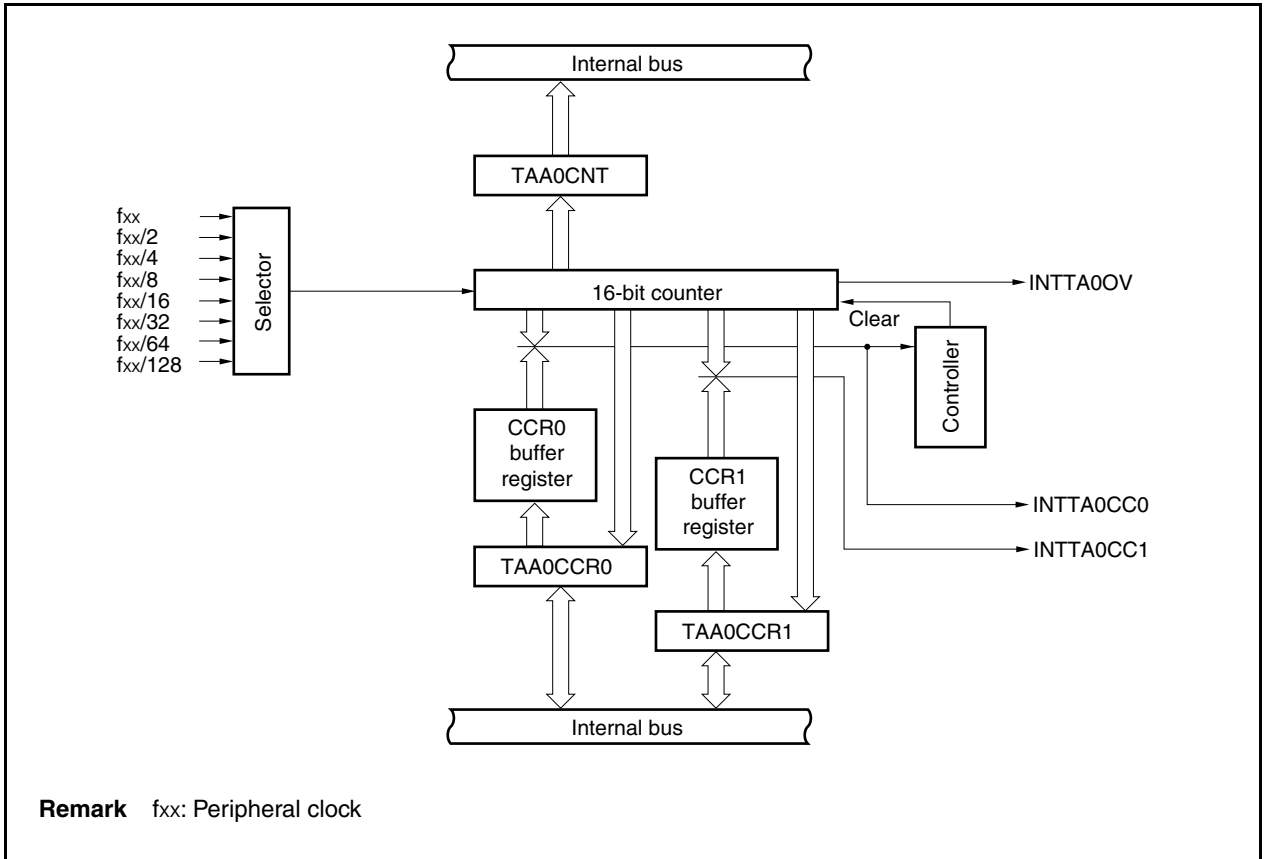


Figure 6-2. TAA1 Block Diagram

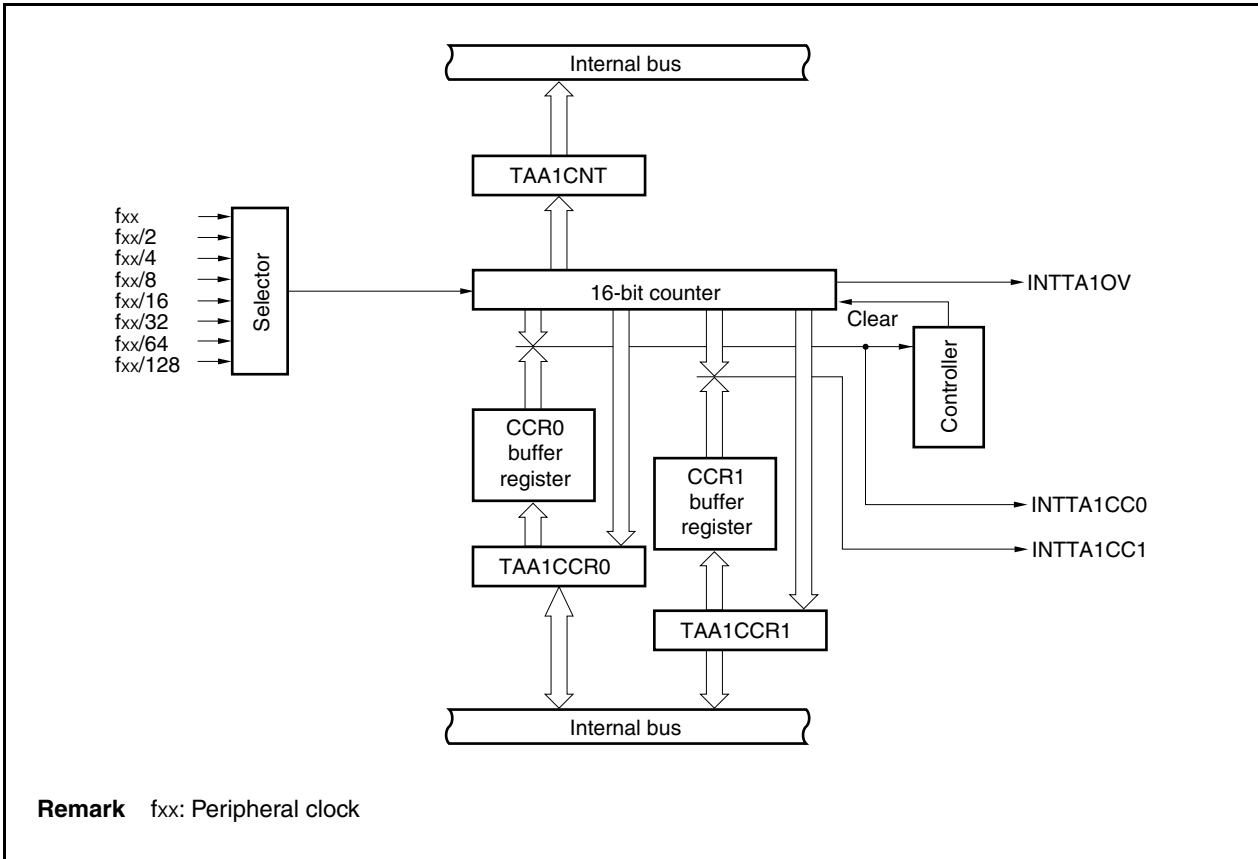


Figure 6-3. TAA2 Block Diagram

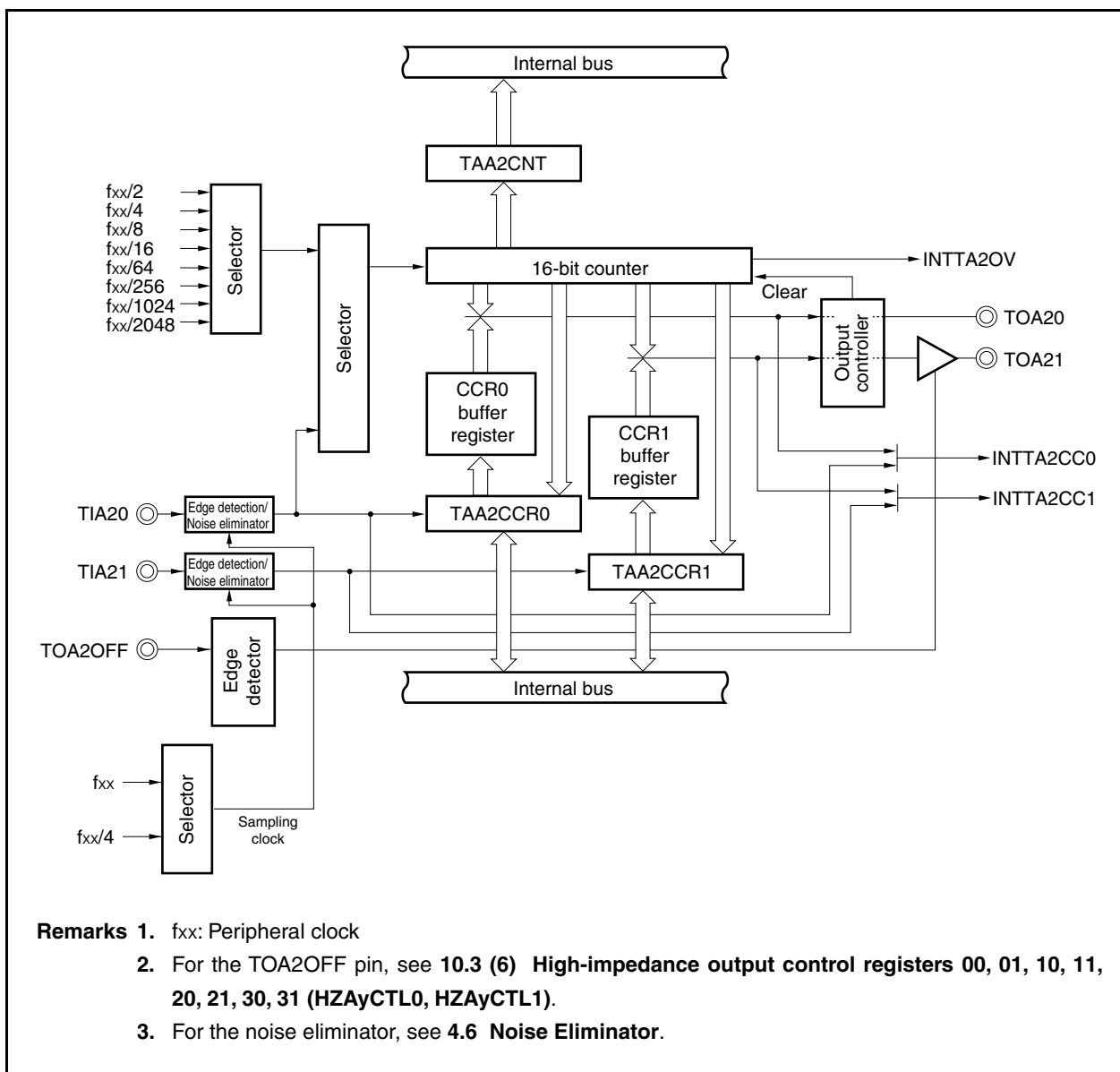


Figure 6-4. TAA3 Block Diagram

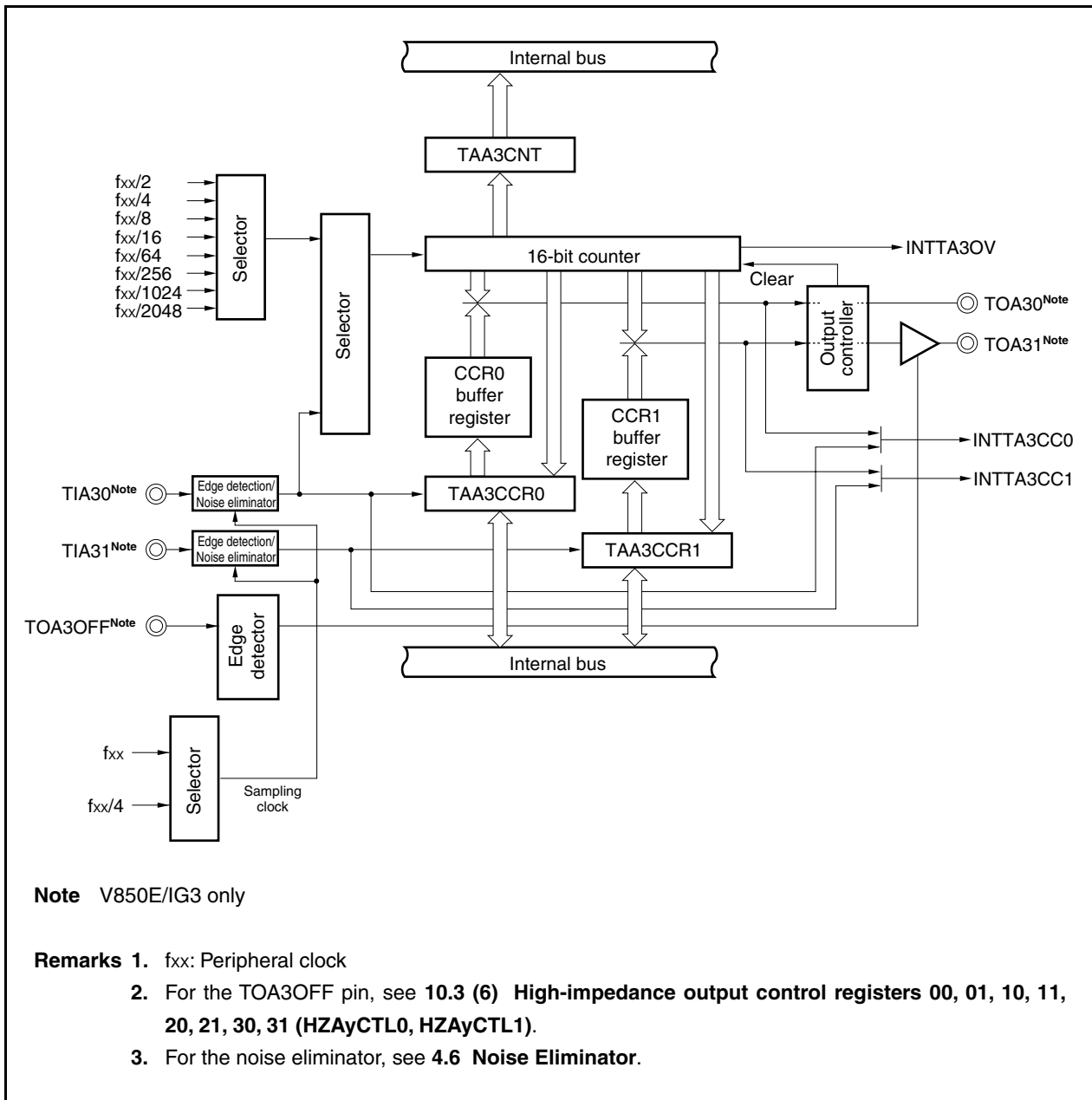
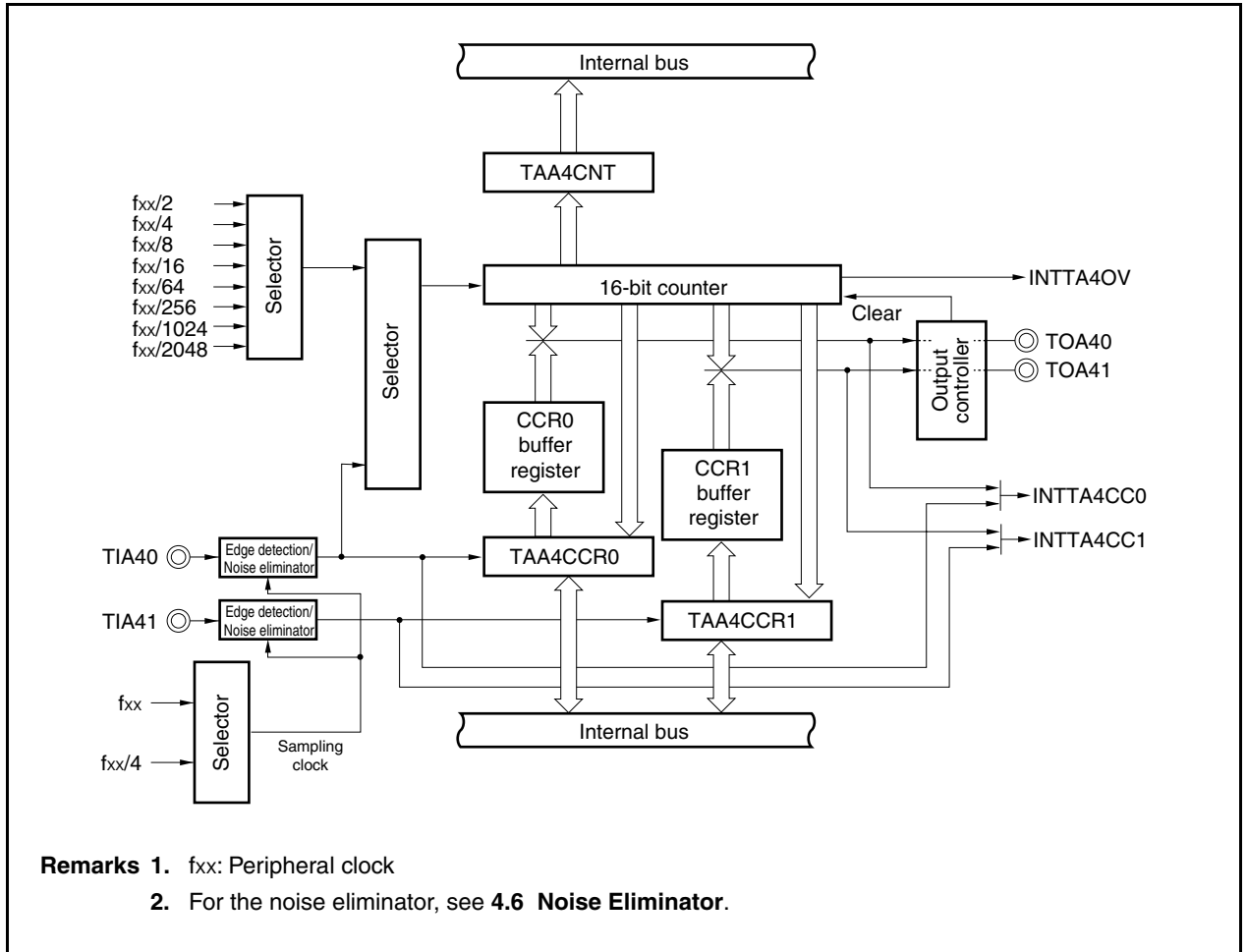


Figure 6-5. TAA4 Block Diagram



**(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 TAAncNT register.

When the TAAncTL0.TAAncCE bit = 0, the value of the 16-bit counter is FFFFH. If the TAAncNT register is read at this time, 0000H is read.

Reset sets the TAAncCE bit to 0.

**(2) CCR0 buffer register**

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

When the TAAncCCR0 register is used as a compare register, the value written to the TAAncCCR0 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 (INTTAAncC0) is generated.

The CCR0 buffer register cannot be read or written directly.

The CCR0 buffer register is cleared to 0000H after reset, and the TAAncCCR0 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 TAAncCCR1 register is used as a compare register, the value written to the TAAncCCR1 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 (INTTAAncC1) is generated.

The CCR1 buffer register cannot be read or written directly.

The CCR1 buffer register is cleared to 0000H after reset, and the TAAncCCR1 register is cleared to 0000H.

**(4) Edge detector**

This circuit detects the valid edges input to the TIA20, TIA21, TIA30 (V850E/IG3 only), TIA31 (V850E/IG3 only), TIA40, and TIA41 pins. No edge, rising edge, falling edge, or both the rising and falling edges can be selected as the valid edge by using the TAAmIOC1 and TAAmIOC2 registers.

**(5) Output controller**

This circuit controls the output of the TOA20, TOA21, TOA30 (V850E/IG3 only), TOA31 (V850E/IG3 only), TOA40, and TOA41 pins. The output controller is controlled by the TAAmIOC0 registers.

**(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

(1) TAA<sub>n</sub> control register 0 (TAA<sub>n</sub>CTL0)

The TAA<sub>n</sub>CTL0 register is an 8-bit register that controls the operation of TAA<sub>n</sub>.

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 TAA<sub>n</sub>CTL0 register by software.

After reset: 00H    R/W    Address: TAA0CTL0 FFFFF660H, TAA1CTL0 FFFFF680H,  
 TAA2CTL0 FFFFF6A0H, TAA3CTL0 FFFFFB00H,  
 TAA4CTL0 FFFFFB20H

|                                       |                     |   |   |   |   |                       |                       |                       |
|---------------------------------------|---------------------|---|---|---|---|-----------------------|-----------------------|-----------------------|
|                                       | <7>                 | 6 | 5 | 4 | 3 | 2                     | 1                     | 0                     |
| TAA <sub>n</sub> CTL0<br>(n = 0 to 4) | TAA <sub>n</sub> CE | 0 | 0 | 0 | 0 | TAA <sub>n</sub> CKS2 | TAA <sub>n</sub> CKS1 | TAA <sub>n</sub> CKS0 |

|                     |  |
|---------------------|--|
| TAA <sub>n</sub> CE | TAA <sub>n</sub> operation control   |
| 0                   | TAA <sub>n</sub> operation disabled (TAA <sub>n</sub> reset asynchronously <sup>Note</sup> ) |
| 1                   | TAA <sub>n</sub> operation enabled. TAA <sub>n</sub> operation start                         |

| TAA <sub>n</sub> CKS2 | TAA <sub>n</sub> CKS1 | TAA <sub>n</sub> CKS0 | Internal count clock selection |                       |
|-----------------------|-----------------------|-----------------------|--------------------------------|-----------------------|
|                       |                       |                       | TAA0, TAA1                     | TAA2 to TAA4          |
| 0                     | 0                     | 0                     | f <sub>xx</sub>                | f <sub>xx</sub> /2    |
| 0                     | 0                     | 1                     | f <sub>xx</sub> /2             | f <sub>xx</sub> /4    |
| 0                     | 1                     | 0                     | f <sub>xx</sub> /4             | f <sub>xx</sub> /8    |
| 0                     | 1                     | 1                     | f <sub>xx</sub> /8             | f <sub>xx</sub> /16   |
| 1                     | 0                     | 0                     | f <sub>xx</sub> /16            | f <sub>xx</sub> /64   |
| 1                     | 0                     | 1                     | f <sub>xx</sub> /32            | f <sub>xx</sub> /256  |
| 1                     | 1                     | 0                     | f <sub>xx</sub> /64            | f <sub>xx</sub> /1024 |
| 1                     | 1                     | 1                     | f <sub>xx</sub> /128           | f <sub>xx</sub> /2048 |

**Note** The TAA<sub>n</sub>OPT0.TAA<sub>n</sub>OVF bit and the 16-bit counter are reset simultaneously. Moreover, timer outputs (TOA20, TOA21, TOA30 (V850E/IG3 only), TOA31 (V850E/IG3 only), TOA40, TOA41 pins) are reset to the TAA<sub>m</sub>IOC0 register set status at the same time as the 16-bit counter is reset (V850E/IF3: m = 2, 4, V850E/IG3: m = 2 to 4).

- Cautions**
1. Set the TAA<sub>n</sub>CKS2 to TAA<sub>n</sub>CKS0 bits when the TAA<sub>n</sub>CE bit = 0.  
 When the value of the TAA<sub>n</sub>CE bit is changed from 0 to 1, the TAA<sub>n</sub>CKS2 to TAA<sub>n</sub>CKS0 bits can be set simultaneously.
  2. Be sure to set bits 3 to 6 to “0”.

**Remark** f<sub>xx</sub>: Peripheral clock

**(2) TAA<sub>n</sub> control register 1 (TAA<sub>n</sub>CTL1)**

The TAA<sub>n</sub>CTL1 register is an 8-bit register that controls the TAA<sub>n</sub> operation.

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: TAA0CTL1 FFFFF661H, TAA1CTL1 FFFFF681H,  
TAA2CTL1 FFFFF6A1H, TAA3CTL1 FFFFFB01H,  
TAA4CTL1 FFFFFB21H

|   |  |   |  |   |   |                      |                      |                      |
|---|--|---|--|---|---|----------------------|----------------------|----------------------|
|   | 7  | 6   | 5                                      | 4 | 3 | 2                    | 1                    | 0                    |
| TAA <sub>n</sub> CTL1                                       | TAA <sub>a</sub> SYE <sup>Note 1</sup>   | TAA <sub>m</sub> EST <sup>Note 2</sup>  | TAA <sub>m</sub> EEE <sup>Note 2</sup> | 0 | 0 | TAA <sub>n</sub> MD2 | TAA <sub>n</sub> MD1 | TAA <sub>n</sub> MDO |
| V850E/IF3<br>n = 0 to 4<br>m = 2, 4<br>a = 0, 1             | TAA <sub>a</sub> SYE <sup>Note 1</sup>   |   | Operation mode selection               |   |   |                      |                      |                      |
|   | 0  | TAA <sub>a</sub> single mode  |  |   |   |                      |                      |                      |
| V850E/IG3<br>n = 0 to 4<br>m = 2 to 4<br>a = 0, 1           | 1  | Tuning operation mode (see 10.4.5)  |  |   |   |                      |                      |                      |
|   | TAA <sub>a</sub> can be used only as an A/D conversion start trigger factor of A/D converters 0 and 1 during the tuning operation. In the tuning operation mode, this bit always operates in synchronization with TAB <sub>a</sub> . |   |  |   |   |                      |                      |                      |
|   | TAA <sub>m</sub> EST <sup>Note 2</sup>   |   | Software trigger control               |   |   |                      |                      |                      |
|   | 0  | -   |  |   |   |                      |                      |                      |
|   | 1  | Generates a valid signal for external trigger input. <ul style="list-style-type: none"> <li>In one-shot pulse output mode:<br/>A one-shot pulse is output with writing 1 to the TAA<sub>m</sub>EST bit as the trigger.</li> <li>In external trigger pulse output mode:<br/>A PWM waveform is output with writing 1 to the TAA<sub>m</sub>EST bit as the trigger.</li> </ul> |  |   |   |                      |                      |                      |
| The read value of the TAA <sub>m</sub> EST bit is always 0. |  |   |  |   |   |                      |                      |                      |

**Notes 1.** This bit can be set only in TAA0 and TAA1. Be sure to set bit 7 of TAA2 to TAA4 to "0".

For details of tuning operation mode, see **CHAPTER 10 MOTOR CONTROL FUNCTION**.

**2.** In the V850E/IF3, this bit can be set only in TAA2 and TAA4. Be sure to set bits 5 and 6 of TAA0, TAA1, and TAA3 to "0".

In the V850E/IG3, this bit can be set only in TAA2 to TAA4. Be sure to set bits 5 and 6 of TAA0 and TAA1 to "0".

| TAAmEEE <sup>Note 1</sup> | Count clock selection  |
|---------------------------|--|
| 0                         | Disable operation with external event count input (TIAm0 pin). (Perform counting with the count clock selected by the TAAmCTL0.TAAmCKS0 to TAAmCTL0.TAAmCKS2 bits.)          |
| 1                         | Enable operation <sup>Note 2</sup> with external event count input (TIAm0 pin). (Perform counting at every valid edge of the external event count input signal (TIAm0 pin).) |

The TAAmEEE bit selects whether counting is performed with the internal count clock or the valid edge of the external event count input.

| TAAmMD2 | TAAmMD1 | TAAmMD0 | Timer mode selection         |
|---------|---------|---------|------------------------------|
| 0       | 0       | 0       | Interval timer mode          |
| 0       | 0       | 1       | External event count mode    |
| 0       | 1       | 0       | External trigger pulse 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           |

**Notes 1.** With the V850E/IF3, this bit can be set only in TAA2 and TAA4. Be sure to set bits 5 and 6 of TAA0, TAA1, and TAA3 to “0”.

With the V850E/IG3, this bit can be set only in TAA2 to TAA4. Be sure to set bits 5 and 6 of TAA0 and TAA1 to “0”.

- Set the valid edge selection of capture trigger input (TIAm0 pin) and external trigger input (TIAm0 pin) to “No edge detection”.

**Cautions 1.** The TAAmEST bit is valid only in the external trigger pulse output mode or one-shot pulse output mode. In any other mode, writing 1 to this bit is ignored.

2. External event count input is selected in the external event count mode regardless of the value of the TAAmEEE bit.

3. Set the TAAaSYE, TAAmEEE, and TAAmMD2 to TAAmMD0 bits when the TAAmCTL0.TAAmCE bit = 0. (The same value can be written when the TAAmCE bit = 1.) The operation is not guaranteed when rewriting is performed with the TAAmCE bit = 1. If rewriting was mistakenly performed, clear the TAAmCE bit to 0 and then set the bits again.

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

**(3) TAAm I/O control register 0 (TAAmIOC0)**

The TAAmIOC0 register is an 8-bit register that controls the timer output (TOAm0, TOAm1 pins). This register can be read or written in 8-bit or 1-bit units. Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: TAA2IOC0 FFFFF6A2H, TAA3IOC0 FFFFFB02H<sup>Note 1</sup>  
TAA4IOC0 FFFFFB22H

|          |   |   |   |   |         |         |         |         |
|----------|---|---|---|---|---------|---------|---------|---------|
|          | 7 | 6 | 5 | 4 | 3       | <2>     | 1       | <0>     |
| TAAmIOC0 | 0 | 0 | 0 | 0 | TAAmOL1 | TAAmOE1 | TAAmOL0 | TAAmOE0 |

( V850E/IF3 )  
m = 2, 4  
a = 0, 1

|         |  |
|---------|--|
| TAAmOL1 | TOAm1 pin output level setting <sup>Note 2</sup> |
| 0       | TOAm1 pin starts output at high level.           |
| 1       | TOAm1 pin starts output at low level.            |

( V850E/IG3 )  
m = 2 to 4  
a = 0, 1

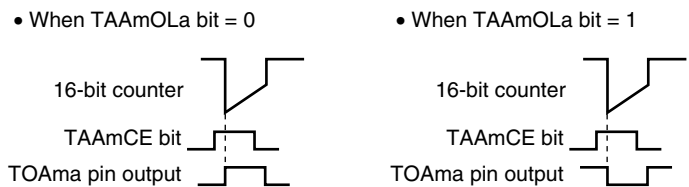
|         |  |
|---------|--|
| TAAmOE1 | TOAm1 pin output setting   |
| 0       | Timer output prohibited<br>• Low level is output from the TOAm1 pin when the TAAmOL1 bit = 0.<br>• High level is output from the TOAm1 pin when the TAAmOL1 bit = 1. |
| 1       | Timer output enabled (A pulse is output from the TOAm1 pin.)   |

|         |  |
|---------|--|
| TAAmOL0 | TOAm0 pin output level setting <sup>Note 2</sup> |
| 0       | TOAm0 pin starts output at high level.           |
| 1       | TOAm0 pin starts output at low level.            |

|         |  |
|---------|--|
| TAAmOE0 | TOAm0 pin output setting   |
| 0       | Timer output prohibited<br>• Low level is output from the TOAm0 pin when the TAAmOL0 bit = 0.<br>• High level is output from the TOAm0 pin when the TAAmOL0 bit = 1. |
| 1       | Timer output enabled (A pulse is output from the TOAm0 pin.)   |

**Notes 1.** V850E/IG3 only

**2.** The output level of the timer output pins (TOAm0 and TOAm1) specified by the TAAmOLa bit is shown below.



- Cautions**
1. If the setting of the TAAmIOC0 register is changed when TOAm0 and TOAm1 are set in the output mode, the output of the pins change. Set the port in the input mode and make the port go into a high-impedance state, noting changes in the pin status.
  2. Rewrite the TAAmOL1, TAAmOE1, TAAmOL0, and TAAmOE0 bits when the TAAmCTL0.TAAmCE bit = 0. (The same value can be written when the TAAmCE bit = 1.) If rewriting was mistakenly performed, clear the TAAmCE bit to 0 and then set the bits again.
  3. Even if the TAAmOL0 or TAAmOL1 bit is manipulated when the TAAmCE, TAAmOE0, and TAAmOE1 bits are 0, the output level of the TOAm0 and TOAm1 pins changes.

**(4) TAAm I/O control register 1 (TAAmIOC1)**

The TAAmIOC1 register is an 8-bit register that controls the valid edge for the capture trigger input signals (TIAm0, TIAm1 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: TAA2IOC1 FFFFF6A3H, TAA3IOC1 FFFFFB03H<sup>Note</sup>,  
TAA4IOC1 FFFFFB23H

|                             |         |         |   |   |         |         |         |         |
|-----------------------------|---------|---------|---|---|---------|---------|---------|---------|
|                             | 7       | 6       | 5   | 4 | 3       | 2       | 1       | 0       |
| TAAmIOC1                    | 0       | 0       | 0   | 0 | TAAmIS3 | TAAmIS2 | TAAmIS1 | TAAmIS0 |
| [ V850E/IF3<br>m = 2, 4 ]   |         |         |   |   |         |         |         |         |
|                             | TAAmIS3 | TAAmIS2 | Capture trigger input signal (TIAm1 pin) valid edge setting |   |         |         |         |         |
| [ V850E/IG3<br>m = 2 to 4 ] | 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                                     |   |         |         |         |         |
|                             | TAAmIS1 | TAAmIS0 | Capture trigger input signal (TIAm0 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                                     |   |         |         |         |         |

**Note** V850E/IG3 only

- Cautions**
1. Rewrite the TAAmIS3 to TAAmIS0 bits when the TAAmCTL0.TAAmCE bit = 0. (The same value can be written when the TAAmCE bit = 1.) If rewriting was mistakenly performed, clear the TAAmCE bit to 0 and then set the bits again.
  2. The TAAmIS3 to TAAmIS0 bits are valid only in the free-running timer mode (only when the TAAmOPT0.TAAmCCS1 and TAAmOPT0.TAAmCCS0 bits = 11) and the pulse width measurement mode. In all other modes, a capture operation is not possible.

**(5) TAAm I/O control register 2 (TAAmIOC2)**

The TAAmIOC2 register is an 8-bit register that controls the valid edge for the external event count input signal (TIAm0 pin) and external trigger input signal (TIAm0 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: TAA2IOC2 FFFFF6A4H, TAA3IOC2 FFFFFB04H<sup>Note</sup>,  
TAA4IOC2 FFFFFB24H

|                             |          |          |  |   |          |          |          |          |
|-----------------------------|----------|----------|--|---|----------|----------|----------|----------|
|                             | 7        | 6        | 5  | 4 | 3        | 2        | 1        | 0        |
| TAAmIOC2                    | 0        | 0        | 0  | 0 | TAAmEES1 | TAAmEES0 | TAAmETS1 | TAAmETS0 |
| [ V850E/IF3<br>m = 2, 4 ]   |          |          |  |   |          |          |          |          |
|                             | TAAmEES1 | TAAmEES0 | External event count input signal (TIAm0 pin) valid edge setting |   |          |          |          |          |
| [ V850E/IG3<br>m = 2 to 4 ] | 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  |   |          |          |          |          |
|                             | TAAmETS1 | TAAmETS0 | External trigger input signal (TIAm0 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  |   |          |          |          |          |

**Note** V850E/IG3 only

- Cautions**
1. Rewrite the TAAmEES1, TAAmEES0, TAAmETS1, and TAAmETS0 bits when the TAAmCTL0.TAAmCE bit = 0. (The same value can be written when the TAAmCE bit = 1.) If rewriting was mistakenly performed, clear the TAAmCE bit to 0 and then set the bits again.
  2. The TAAmEES1 and TAAmEES0 bits are valid only when the TAAmCTL1.TAAmEEE bit = 1 or when the external event count mode (the TAAmCTL1.TAAmMD2 to TAAmCTL1.TAAmMD0 bits = 001) has been set.
  3. The TAAmETS1 and TAAmETS0 bits are valid only in the external trigger pulse mode or one-shot pulse output mode.

**(6) TAA<sub>n</sub> option register 0 (TAA<sub>n</sub>OPT0)**

The TAA<sub>n</sub>OPT0 register is an 8-bit register that sets the capture/compare operation and detects 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: TAA0OPT0 FFFFF665H, TAA1OPT0 FFFFF685H,  
TAA2OPT0 FFFFF6A5H, TAA3OPT0 FFFFFB05H,  
TAA4OPT0 FFFFFB25H

|                       |   |   |                          |                          |   |   |   |                      |
|-----------------------|---|---|--------------------------|--------------------------|---|---|---|----------------------|
|                       | 7 | 6 | 5                        | 4                        | 3 | 2 | 1 | <0>                  |
| TAA <sub>n</sub> OPT0 | 0 | 0 | TAAmCCS1 <sup>Note</sup> | TAAmCCS0 <sup>Note</sup> | 0 | 0 | 0 | TAA <sub>n</sub> OVF |

[ V850E/IF3  
n = 0 to 4  
m = 2, 4 ]

[ V850E/IG3  
n = 0 to 4  
m = 2 to 4 ]

|  |  |
|--|--|
| TAAmCCS1 <sup>Note</sup>   | TAAmCCR1 register capture/compare selection                    |
| 0  | Compare register selected                                      |
| 1  | Capture register selected (cleared by TAAmCTL0.TAAmCE bit = 0) |
| The TAAmCCS1 bit setting is valid only in the free-running timer mode. |  |

|  |  |
|--|--|
| TAAmCCS0 <sup>Note</sup>   | TAAmCCR0 register capture/compare selection                    |
| 0  | Compare register selected                                      |
| 1  | Capture register selected (cleared by TAAmCTL0.TAAmCE bit = 0) |
| The TAAmCCS0 bit setting is valid only in the free-running timer mode. |  |

|   |   |
|---|---|
| TAA <sub>n</sub> OVF  | TAA <sub>n</sub> overflow detection flag  |
| Set (1)   | Overflow occurred   |
| Reset (0)   | 0 is written to TAA <sub>n</sub> OVF bit or TAA <sub>n</sub> CTL0.TAA <sub>n</sub> CE bit = 0 |
| <ul style="list-style-type: none"> <li>The TAA<sub>n</sub>OVF bit is set to 1 when the 16-bit counter value overflows from FFFFH to 0000H in the free-running timer mode or the pulse width measurement mode.</li> <li>An overflow interrupt request signal (INTTAA<sub>n</sub>OV) is generated at the same time that the TAA<sub>n</sub>OVF bit is set to 1. The INTTAA<sub>n</sub>OV signal is not generated in modes other than the free-running timer mode and the pulse width measurement mode.</li> <li>The TAA<sub>n</sub>OVF bit is not cleared to 0 even when the TAA<sub>n</sub>OVF bit or the TAA<sub>n</sub>OPT0 register are read when the TAA<sub>n</sub>OVF bit = 1.</li> <li>Before clearing the TAA<sub>n</sub>OVF bit to 0 after generation of the INTTAA<sub>n</sub>OV signal, be sure to confirm (by reading) that the TAA<sub>n</sub>OVF bit is set to 1.</li> <li>The TAA<sub>n</sub>OVF bit can be both read and written, but the TAA<sub>n</sub>OVF bit cannot be set to 1 by software. Writing 1 has no effect on the operation of TAA<sub>n</sub>.</li> </ul> |   |

**Note** With the V850E/IF3, this bit can be set only in TAA2 and TAA4. Be sure to set bits 4 and 5 of TAA0, TAA1, and TAA3 to “0”.

With the V850E/IG3, this bit can be set only in TAA2 to TAA4. Be sure to set bits 4 and 5 of TAA0 and TAA1 to “0”.

**Cautions 1.** Rewrite the TAAmCCS1 and TAAmCCS0 bits when the TAAmCE bit = 0. (The same value can be written when the TAAmCE bit = 1.) If rewriting was mistakenly performed, clear the TAAmCE bit to 0 and then set the bits again.

**2.** Be sure to set bits 1 to 3, 6, and 7 to “0”.



**(7) TAA<sub>n</sub> capture/compare register 0 (TAA<sub>n</sub>CCR0)**

The TAA<sub>m</sub>CCR0 register is a 16-bit register that can be used as a capture register or compare register depending on the mode. The TAA<sub>k</sub>CCR0 register is a 16-bit register that can only be used as a compare register.

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 TAA<sub>m</sub>OPT0.TAA<sub>m</sub>CCS0 bit. In the pulse width measurement mode, the TAA<sub>m</sub>CCR0 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

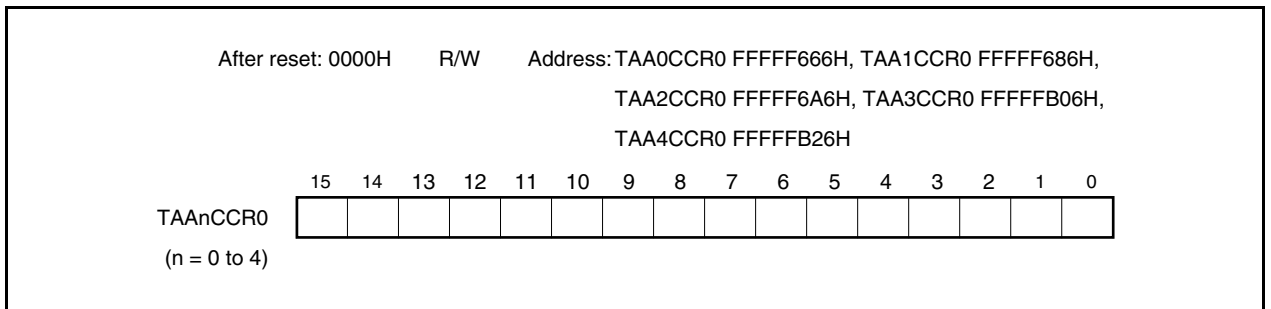
The TAA<sub>n</sub>CCR0 register can be read or written during operation.

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

Reset sets this register to 0000H.

**Remark** V850E/IF3: n = 0 to 4, m = 2, 4, k = 0, 1, 3

V850E/IG3: n = 0 to 4, m = 2 to 4, k = 0, 1



**(a) Function as compare register**

The TAAAnCCR0 register can be rewritten even when the TAAAnCTL0.TAAAnCE bit = 1.

The set value of the TAAAnCCR0 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 (INTTAAAnCC0) is generated. If TOAm0 pin output is enabled at this time, the output of the TOAm0 pin is inverted.

When the TAAAnCCR0 register is used as a cycle register in the interval timer mode or the TAAmCCR0 register is used as a cycle register in 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.

The compare register is not cleared by setting the TAAAnCTL0.TAAAnCE bit to 0.

**(b) Function as capture register**

When the TAAmCCR0 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 TAAmCCR0 register if the valid edge of the capture trigger input pin (TIAm0 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TAAmCCR0 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIAm0 pin) is detected.

Even if the capture operation and reading the TAAmCCR0 register conflict, the correct value of the TAAmCCR0 register can be read.

The capture register is cleared by setting the TAAmCTL0.TAAmCE bit to 0.

**Remark** V850E/IF3: n = 0 to 4, m = 2, 4  
 V850E/IG3: n = 0 to 4, m = 2 to 4

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

**Table 6-4. 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 <sup>Note 1</sup>        | Compare register         | Anytime write                 |
| External trigger pulse output <sup>Note 1</sup> | Compare register         | Batch write <sup>Note 2</sup> |
| One-shot pulse output <sup>Note 1</sup>         | Compare register         | Anytime write                 |
| PWM output <sup>Note 1</sup>                    | Compare register         | Batch write <sup>Note 2</sup> |
| Free-running timer                              | Capture/compare register | Anytime write                 |
| Pulse width measurement <sup>Note 1</sup>       | Capture register         | None                          |

**Notes 1.** With the V850E/IF3, this mode is only for TAA2 and TAA4. With the V850E/IG3, this mode is only for TAA2 to TAA4.

**2.** Writing to the TAAmCCR1 register is the trigger.

**Remark** For anytime write and batch write, see 6.6 (2) **Anytime write and batch write.**

**(8) TAA<sub>n</sub> capture/compare register 1 (TAA<sub>n</sub>CCR1)**

The TAA<sub>m</sub>CCR1 register is a 16-bit register that can be used as a capture register or compare register depending on the mode. The TAA<sub>k</sub>CCR1 register is a 16-bit register that can only be used as a compare register.

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 TAA<sub>m</sub>OPT0.TAA<sub>m</sub>CCS1 bit. In the pulse width measurement mode, the TAA<sub>m</sub>CCR1 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

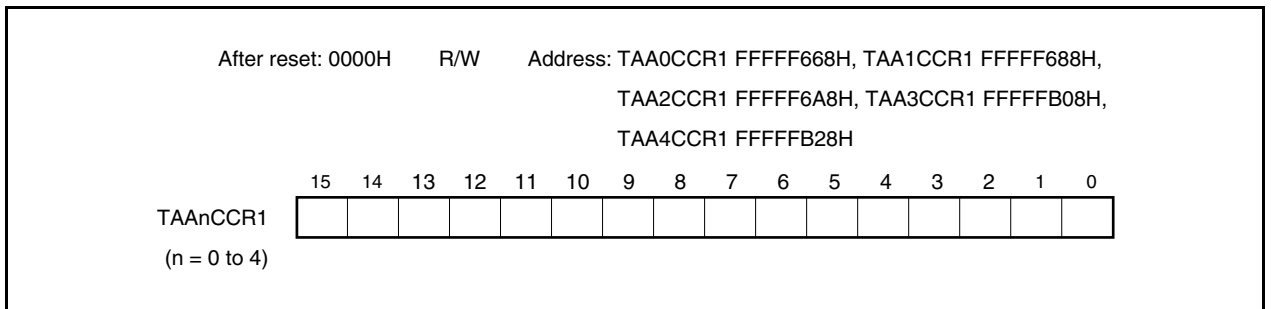
The TAA<sub>n</sub>CCR1 register can be read or written during operation.

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

Reset sets this register to 0000H.

**Remark** V850E/IF3: n = 0 to 4, m = 2, 4, k = 0, 1, 3

V850E/IG3: n = 0 to 4, m = 2 to 4, k = 0, 1



**(a) Function as compare register**

The TAA<sub>n</sub>CCR1 register can be rewritten even when the TAA<sub>n</sub>CTL0.TAA<sub>n</sub>CE bit = 1.

The set value of the TAA<sub>n</sub>CCR1 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 (INTTAA<sub>n</sub>CC1) is generated. If TOA<sub>m</sub>1 pin output is enabled at this time, the output of the TOA<sub>m</sub>1 pin is inverted.

The compare register is not cleared by setting the TAA<sub>n</sub>CTL0.TAA<sub>n</sub>CE bit to 0.

**(b) Function as capture register**

When the TAA<sub>m</sub>CCR1 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 TAA<sub>m</sub>CCR1 register if the valid edge of the capture trigger input pin (TIA<sub>m</sub>1 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TAA<sub>m</sub>CCR1 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIA<sub>m</sub>1 pin) is detected.

Even if the capture operation and reading the TAA<sub>m</sub>CCR1 register conflict, the correct value of the TAA<sub>m</sub>CCR1 register can be read.

The capture register is cleared by setting the TAA<sub>m</sub>CTL0.TAA<sub>m</sub>CE bit to 0.

**Remark** V850E/IF3: n = 0 to 4, m = 2, 4  
 V850E/IG3: n = 0 to 4, m = 2 to 4

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

**Table 6-5. 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 <sup>Note 1</sup>        | Compare register         | Anytime write                 |
| External trigger pulse output <sup>Note 1</sup> | Compare register         | Batch write <sup>Note 2</sup> |
| One-shot pulse output <sup>Note 1</sup>         | Compare register         | Anytime write                 |
| PWM output <sup>Note 1</sup>                    | Compare register         | Batch write <sup>Note 2</sup> |
| Free-running timer                              | Capture/compare register | Anytime write                 |
| Pulse width measurement <sup>Note 1</sup>       | Capture register         | None                          |

**Notes 1.** In the V850E/IF3, this mode is only for TAA2 and TAA4. In the V850E/IG3, this mode is only for TAA2 to TAA4.

**2.** Writing to the TAA<sub>m</sub>CCR1 register is the trigger.

**Remark** For anytime write and batch write, see **6.6 (2) Anytime write and batch write**.

**(9) TAA<sub>n</sub> counter read buffer register (TAA<sub>n</sub>CNT)**

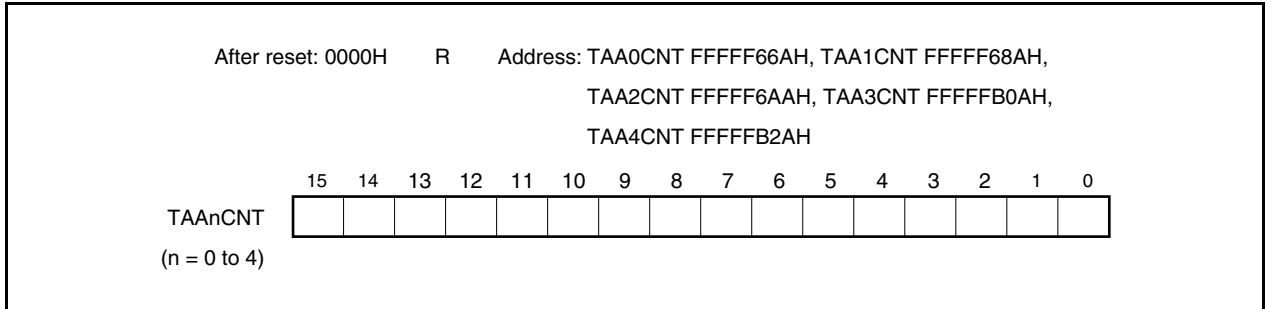
The TAA<sub>n</sub>CNT register is a read buffer register that can read the count value of the 16-bit counter.

If this register is read when the TAA<sub>n</sub>CTL0.TAA<sub>n</sub>CE 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 TAA<sub>n</sub>CNT register is cleared to 0000H when the TAA<sub>n</sub>CE bit = 0. If the TAA<sub>n</sub>CNT 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 TAA<sub>n</sub>CNT register is cleared to 0000H after reset, and the TAA<sub>n</sub>CE bit is cleared to 0.



## 6.5 Timer Output Operations

The following table shows the operations and output levels of the TOAm0 and TOAm1 pins.

**Table 6-6. Timer Output Control in Each Mode**

| Operation Mode                     | TOAm1 Pin                                       | TOAm0 Pin  |
|------------------------------------|---|------------|
| Interval timer mode                | PWM output                                      |            |
| External event count mode          | None  |            |
| External trigger pulse output mode | External trigger pulse output                   | PWM output |
| One-shot pulse output mode         | One-shot pulse output                           |            |
| PWM output mode                    | PWM output                                      |            |
| Free-running timer mode            | PWM output (only when compare function is used) |            |
| Pulse width measurement mode       | None  |            |

**Remark** V850E/IF3: m = 2, 4  
 V850E/IG3: m = 2 to 4

**Table 6-7. Truth Table of TOAm0 and TOAm1 Pins Under Control of Timer Output Control Bits**

| TAAmIOC0.TAAmOLa Bit | TAAmIOC0.TAAmOEa Bit | TAAmCTL0.TAAmCE Bit | Level of TOAma 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** V850E/IF3: m = 2, 4, a = 0, 1  
 V850E/IG3: m = 2 to 4, a = 0, 1

## 6.6 Operation

The functions of TAA<sub>n</sub> that can be achieved differ from one channel to another. The functions of each channel are shown below.

**Table 6-8. TAA0 and TAA1 Specifications in Each Mode**

| Operation                          | Software Trigger Bit | External Trigger Input | Capture/Compare Register Setting | Compare Register Write Method |
|------------------------------------|----------------------|------------------------|----------------------------------|-------------------------------|
| Interval timer mode                | Invalid              | Invalid                | Compare only                     | Anytime write                 |
| External event count mode          | None                 |                        |                                  |                               |
| External trigger pulse output mode | None                 |                        |                                  |                               |
| One-shot pulse output mode         | None                 |                        |                                  |                               |
| PWM output mode                    | None                 |                        |                                  |                               |
| Free-running timer mode            | Invalid              | Invalid                | Compare only                     | Anytime write                 |
| Pulse width measurement mode       | None                 |                        |                                  |                               |

- Remarks**
1. TAA<sub>a</sub> does not have timer input pins (TIA<sub>a0</sub>, TIA<sub>a1</sub>) and timer output pins (TOA<sub>a0</sub>, TOA<sub>a1</sub>). It has interrupt request signals (INTTAA<sub>CC0</sub>, INTTAA<sub>CC1</sub>) on a match between the value of the 16-bit counter and the values of the TAA<sub>a</sub>CCR0 and TAA<sub>a</sub>CCR1 registers.
  2. TAA<sub>a</sub> has a function to execute tuning with TAB<sub>a</sub>. For details, see **CHAPTER 10 MOTOR CONTROL FUNCTION**.
  3. a = 0, 1

**Table 6-9. TAA2 to TAA4 Specifications in Each Mode**

| Operation  | TAAmCTL1.TAAmEST Bit (Software Trigger Bit) | TIAM0 Pin (External Trigger Input) | Capture/Compare Register Setting | Compare Register Write Method |
|--|---|------------------------------------|----------------------------------|-------------------------------|
| Interval timer mode                                  | Invalid                                     | Invalid                            | Compare only                     | Anytime write                 |
| External event count mode <sup>Note 1</sup>          | Invalid                                     | Invalid                            | Compare only                     | Anytime write                 |
| External trigger pulse output mode <sup>Note 2</sup> | Valid                                       | Valid                              | Compare only                     | Batch write                   |
| One-shot pulse output mode <sup>Note 2</sup>         | Valid                                       | Valid                              | Compare only                     | Anytime write                 |
| PWM output mode                                      | Invalid                                     | Invalid                            | Compare only                     | Batch write                   |
| Free-running timer mode                              | Invalid                                     | Invalid                            | Switchable <sup>Note 3</sup>     | Anytime write                 |
| Pulse width measurement mode <sup>Note 2</sup>       | Invalid                                     | Invalid                            | Capture only                     | Not applicable                |

- Notes**
1. When using the external event count mode, set the TIAM0 pin capture trigger input valid edge selection to “No edge detection”. (Clear the TAAmIOC1.TAAmIS1 and TAAmIOC1.TAAmIS0 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 TAAmCTL1.TAAmEEE bit to 0).
  3. In TAA3 of the V850E/IF3, this setting is compare only.

**Remark** V850E/IF3: m = 2, 4  
V850E/IG3: m = 2 to 4

**(1) Counter basic operation**

This section explains the basic operation of the 16-bit counter. For details, refer to the description of the operation in each mode.

**Remark** V850E/IF3: n = 0 to 4, m = 2, 4  
V850E/IG3: n = 0 to 4, m = 2 to 4

**(a) Counter start operation**

- In external event count mode  
When the TAAmCTL0.TAAmCE bit is set from 0 to 1, the 16-bit counter is set to 0000H.  
After that, it counts up to 0001H, 0002H, 0003H, ... each time the valid edge of external event count input (TIAm0) is detected.
- In modes other than the above  
Starts counting from the default value FFFFH.  
It counts up from FFFFH to 0000H, 0001H, 0002H, 0003H, and so on.

**(b) Clear operation**

The 16-bit counter is cleared to 0000H when its value matches the value of the compare register and is cleared, and when its value is captured and cleared. The counting operation from FFFFH to 0000H that takes place immediately after the counter has started counting or when the counter overflows is not a clearing operation. Therefore, the INTTAnCC0 and INTTAnCC1 interrupt signals are not generated.

**(c) Overflow operation**

The 16-bit counter overflows when the counter counts up from FFFFH to 0000H in the free-running mode or pulse width measurement mode. If the counter overflows, the TAAAnOPT0.TAAAnOVF bit is set to 1 and an interrupt request signal (INTTAnOV) is generated. Note that the INTTAnOV signal is not generated under the following conditions.

- Immediately after a counting operation has been started
- If the counter value matches the compare value FFFFH and is cleared
- When FFFFH is captured and cleared in the pulse width measurement mode and the counter counts up from FFFFH to 0000H

**Caution** After the overflow interrupt request signal (INTTAnOV) has been generated, be sure to check that the overflow flag (TAAAnOVF bit) is set to 1.

**(d) Counter read operation during counting operation**

The value of the 16-bit counter of TAAAn can be read by using the TAAAnCNT register during the count operation. When the TAAAnCTL0.TAAAnCE bit = 1, the value of the 16-bit counter can be read by reading the TAAAnCNT register. When the TAAAnCTL0.TAAAnCE bit = 0, the 16-bit counter is FFFFH and the TAAAnCNT register is 0000H.



**(e) Interrupt operation**

TAA<sub>n</sub> generates the following three types of interrupt request signals.

- **INTTAA<sub>n</sub>CC0 interrupt:** This signal functions as a match interrupt request signal of the CCR0 buffer register and as a capture interrupt request signal to the TAA<sub>n</sub>CCR0 register.
- **INTTAA<sub>n</sub>CC1 interrupt:** This signal functions as a match interrupt request signal of the CCR1 buffer register and as a capture interrupt request signal to the TAA<sub>n</sub>CCR1 register.
- **INTTAA<sub>n</sub>OV interrupt:** This signal functions as an overflow interrupt request signal.

**(2) Anytime write and batch write**

The TAAAnCCR0 and TAAAnCCR1 registers in TAAAn can be rewritten during timer operation (TAAAnCTL0.TAAAnCE bit = 1), but the write method (anytime write, batch write) of the CCR0 and CCR1 buffer registers differs depending on the mode.

**(a) Anytime write**

In this mode, data is transferred at any time from the TAAAnCCR0 and TAAAnCCR1 registers to the CCR0 and CCR1 buffer registers during timer operation.

**Remark** n = 0 to 4

**Figure 6-6. Flowchart of Basic Operation for Anytime Write**

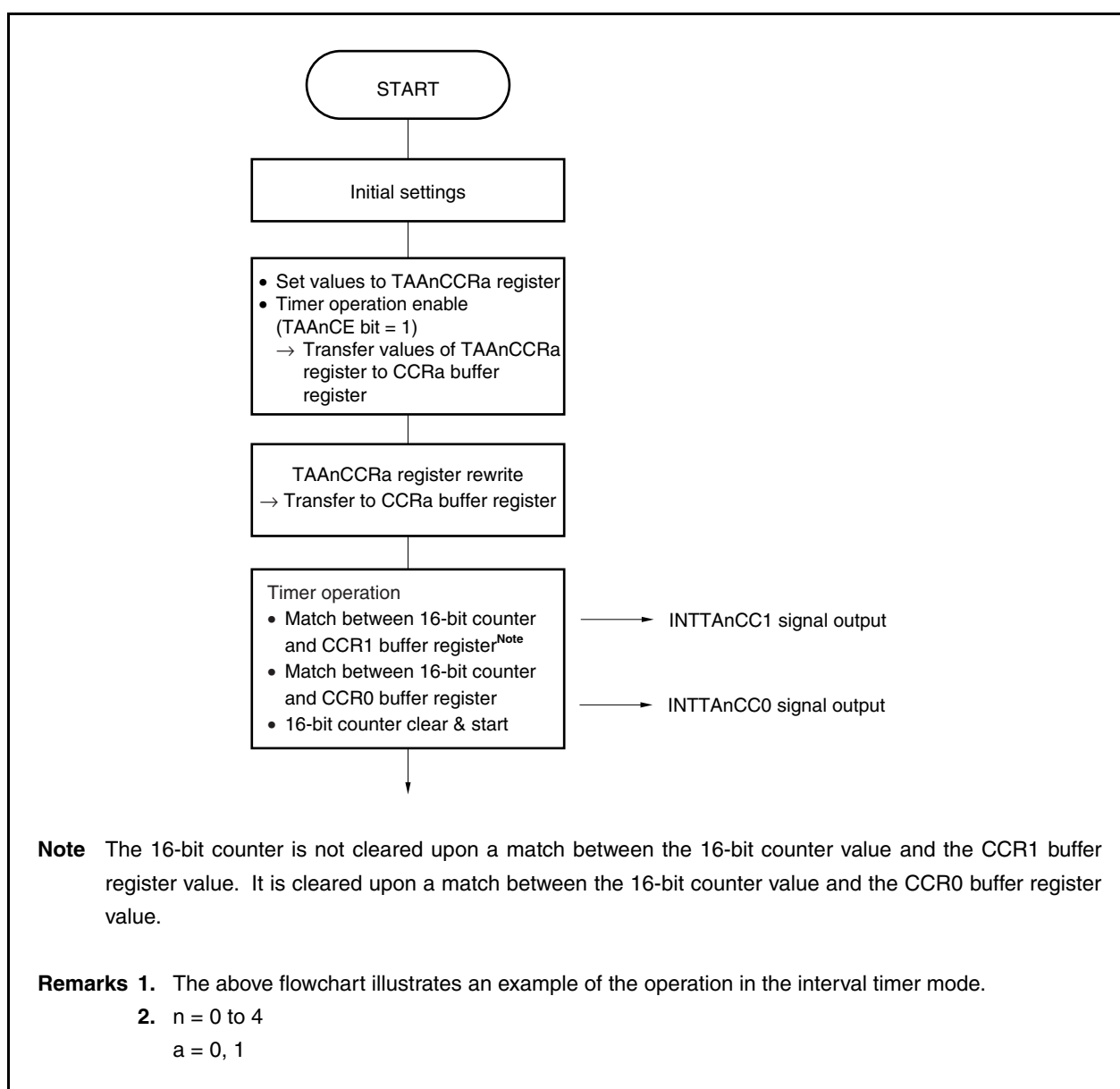
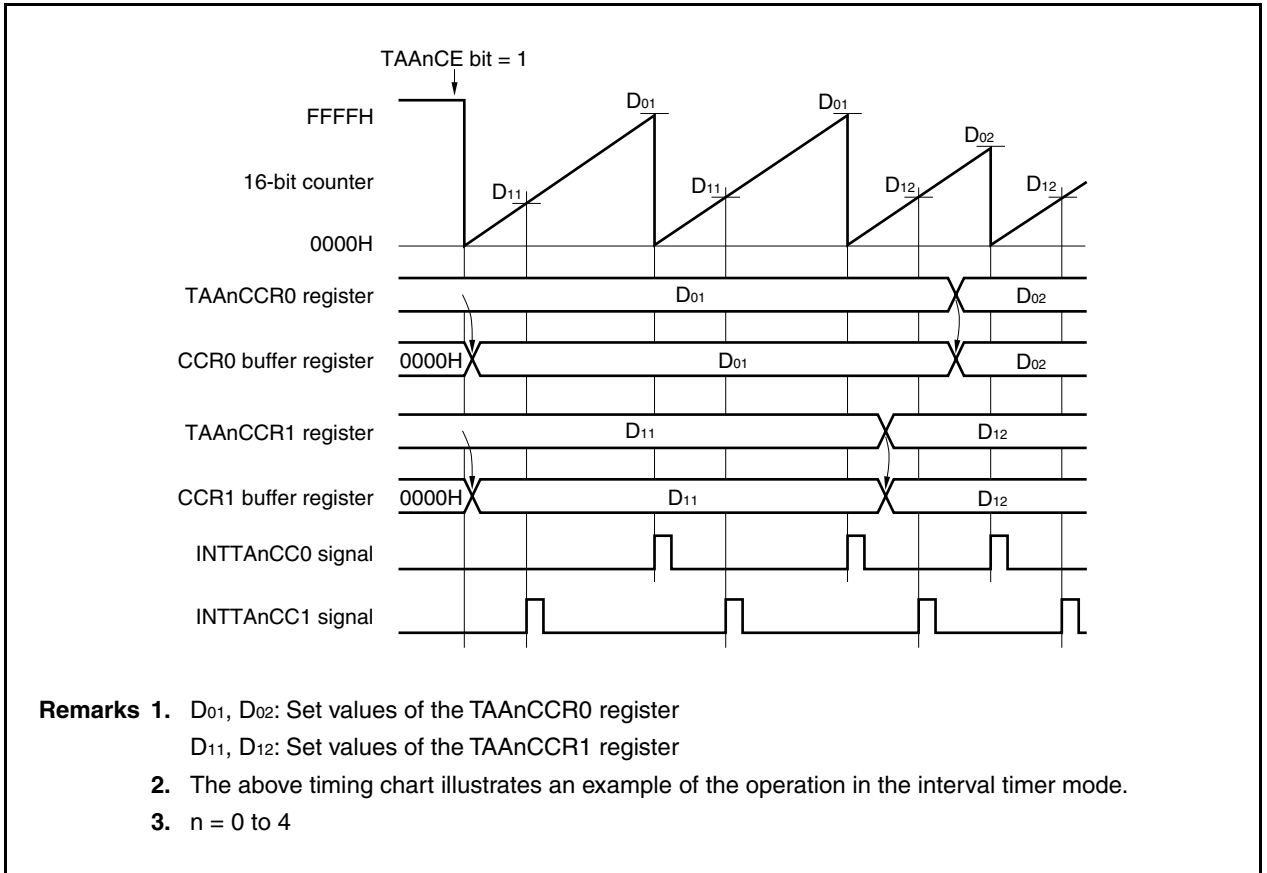


Figure 6-7. Timing of Anytime Write



**(b) Batch write**

In this mode, data is transferred all at once from the TAAmCCR0 and TAAmCCR1 registers to the CCR0 and CCR1 buffer registers during timer operation. This data is transferred upon a match between the value of the CCR0 buffer register and the value of the 16-bit counter. Transfer is enabled by writing to the TAAmCCR1 register. Whether to enable or disable the next transfer timing is controlled by writing or not writing to the TAAmCCR1 register.

In order for the set value when the TAAmCCR0 and TAAmCCR1 registers are rewritten to become the 16-bit counter comparison value (in other words, in order for this value to be transferred to the CCR0 and CCR1 buffer registers), it is necessary to rewrite the TAAmCCR0 register and then write to the TAAmCCR1 register before the 16-bit counter value and the CCR0 buffer register value match. Therefore, the values of the TAAmCCR0 and TAAmCCR1 registers are transferred to the CCR0 and CCR1 buffer registers upon a match between the count value of the 16-bit counter and the value of the CCR0 buffer register. Thus even when wishing only to rewrite the value of the TAAmCCR0 register, also write the same value (same as preset value of the TAAmCCR1 register) to the TAAmCCR1 register.

**Remark** V850E/IF3: m = 2, 4  
V850E/IG3: m = 2 to 4

Figure 6-8. Flowchart of Basic Operation for Batch Write

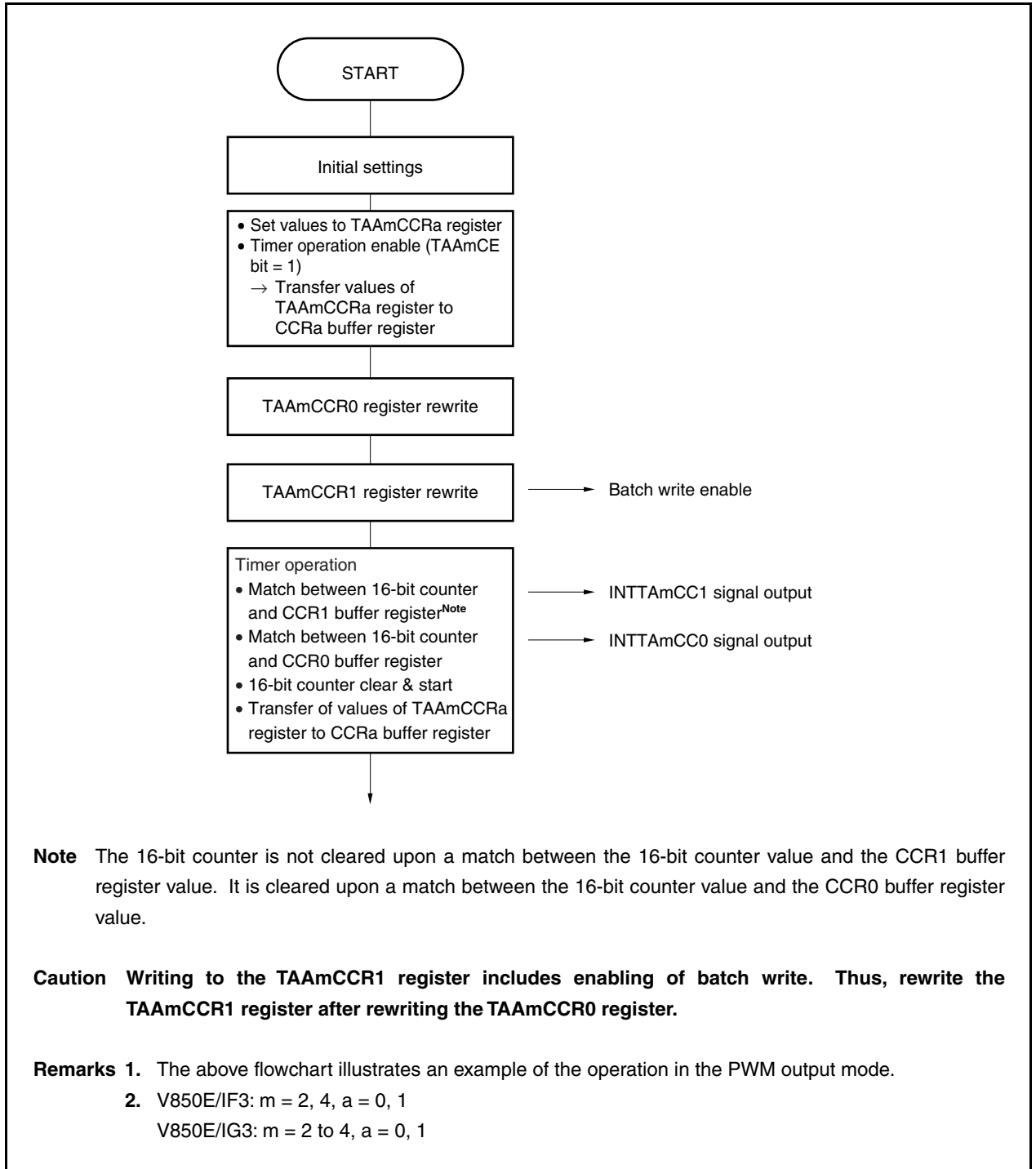
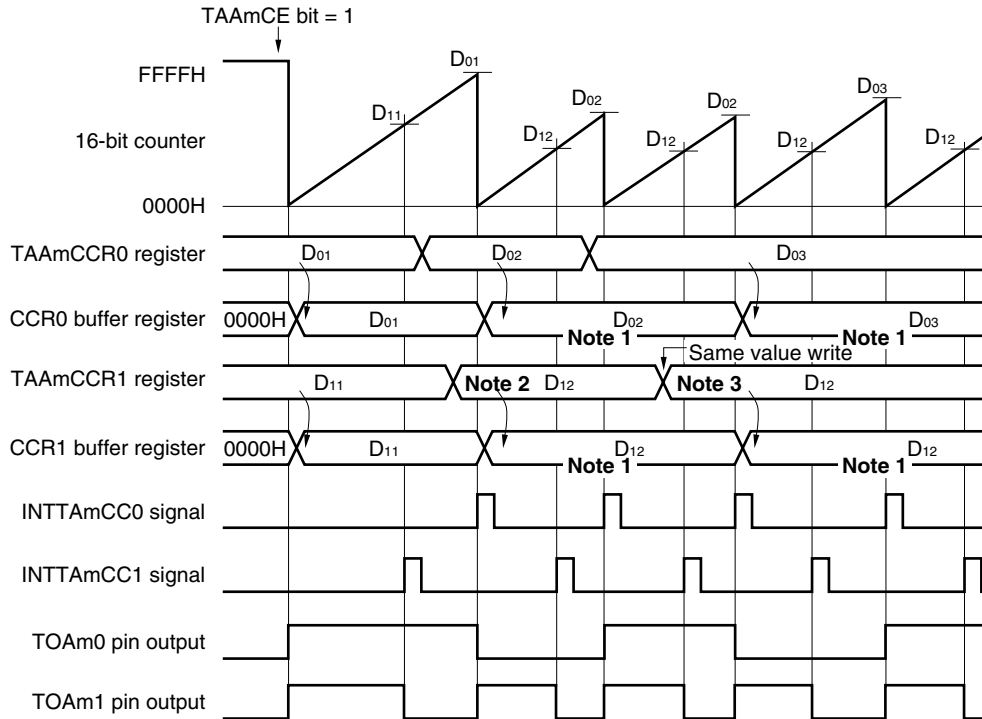


Figure 6-9. Timing of Batch Write



- Notes 1.** Because the TAAmCCR1 register was not rewritten, D<sub>03</sub> is not transferred.
- 2.** Because the TAAmCCR1 register has been written (D<sub>12</sub>), data is transferred to the CCR1 buffer register upon a match between the value of the 16-bit counter and the value of the TAAmCCR0 register (D<sub>01</sub>).
- 3.** Because the TAAmCCR1 register has been written (D<sub>12</sub>), data is transferred to the CCR1 buffer register upon a match between the value of the 16-bit counter and the value of the TAAmCCR0 register (D<sub>02</sub>).

- Remarks 1.** D<sub>01</sub>, D<sub>02</sub>, D<sub>03</sub>: Set values of TAAmCCR0 register  
 D<sub>11</sub>, D<sub>12</sub>: Set values of TAAmCCR1 register
- 2.** The above timing chart illustrates the operation in the PWM output mode as an example.
- 3.** V850E/IF3: m = 2, 4  
 V850E/IG3: m = 2 to 4

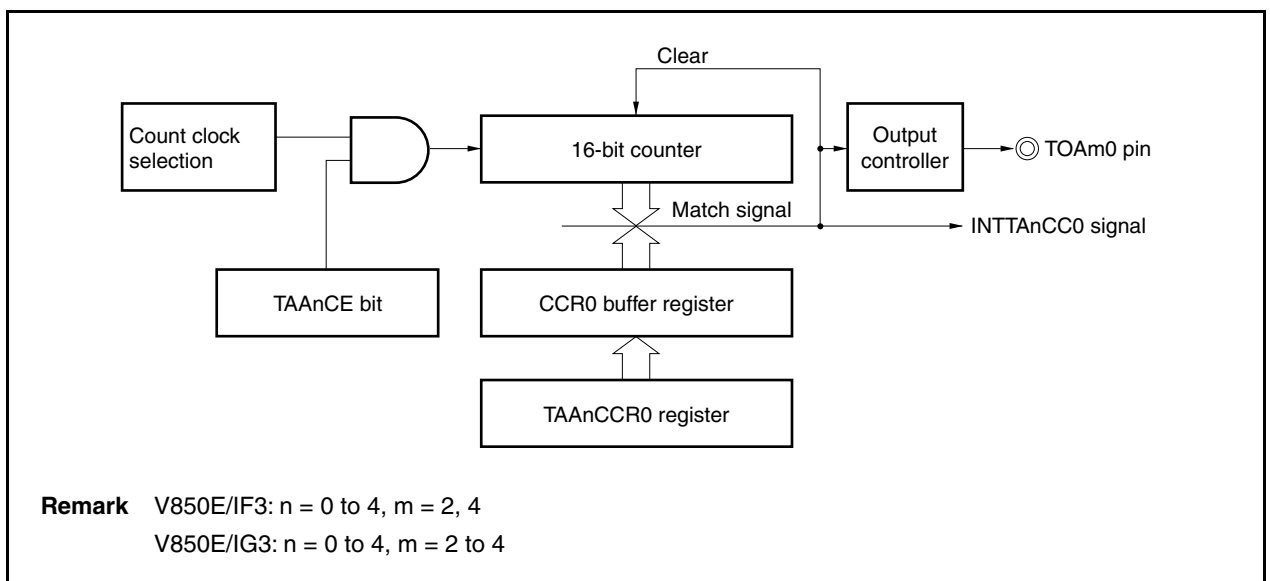
**6.6.1 Interval timer mode (TAA<sub>n</sub>MD2 to TAA<sub>n</sub>MD0 bits = 000)**

In the interval timer mode, an interrupt request signal (INTTAA<sub>n</sub>CC0) is generated at the interval set by the TAA<sub>n</sub>CCR0 register if the TAA<sub>n</sub>CTL0.TAA<sub>n</sub>CE bit is set to 1. A PWM waveform with a duty factor of 50% whose half cycle is equal to the interval can be output from the TOA<sub>m</sub>0 pin.

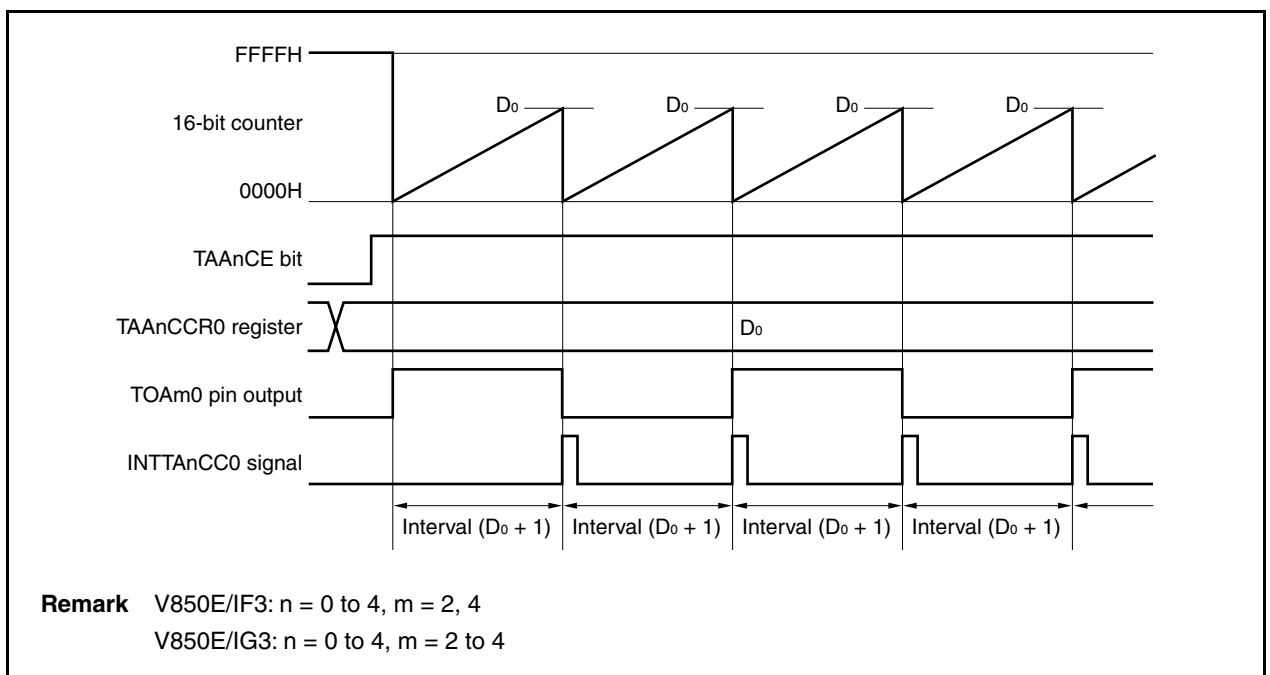
The TAA<sub>n</sub>CCR1 register is not used in the interval timer mode. However, the set value of the TAA<sub>n</sub>CCR1 register is transferred to the CCR1 buffer register, and when the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTAA<sub>n</sub>CC1) is generated. In addition, a PWM waveform with a duty factor of 50%, which is inverted when the INTTAA<sub>n</sub>CC1 signal is generated, can be output from the TOA<sub>m</sub>1 pin.

The value of the TAA<sub>n</sub>CCR0 and TAA<sub>n</sub>CCR1 registers can be rewritten even while the timer is operating.

**Figure 6-10. Configuration of Interval Timer**



**Figure 6-11. Basic Timing of Operation in Interval Timer Mode**



When the TAA<sub>n</sub>CE 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 TOA<sub>m</sub>0 pin is inverted. Additionally, the set value of the TAA<sub>n</sub>CCR0 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 TOA<sub>m</sub>0 pin is inverted, and a compare match interrupt request signal (INTTAA<sub>n</sub>CC0) is generated.

The interval can be calculated by the following expression.

$$\text{Interval} = (\text{Set value of TAA}_{n}\text{CCR0 register} + 1) \times \text{Count clock cycle}$$

**Remark** V850E/IF3: n = 0 to 4, m = 2, 4  
 V850E/IG3: n = 0 to 4, m = 2 to 4

Figure 6-12. Register Setting for Interval Timer Mode Operation (1/3)

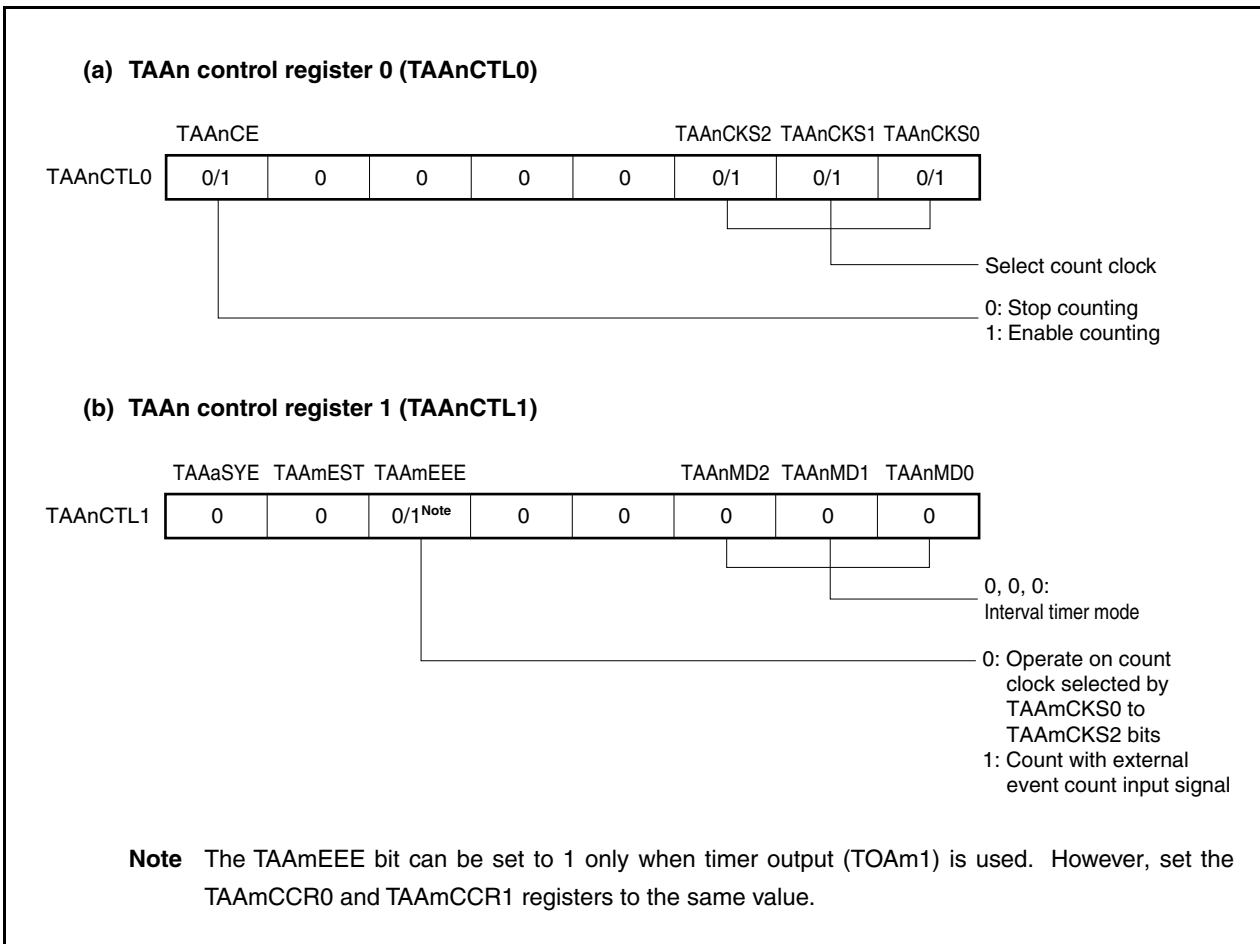




Figure 6-12. Register Setting for Interval Timer Mode Operation (2/3)

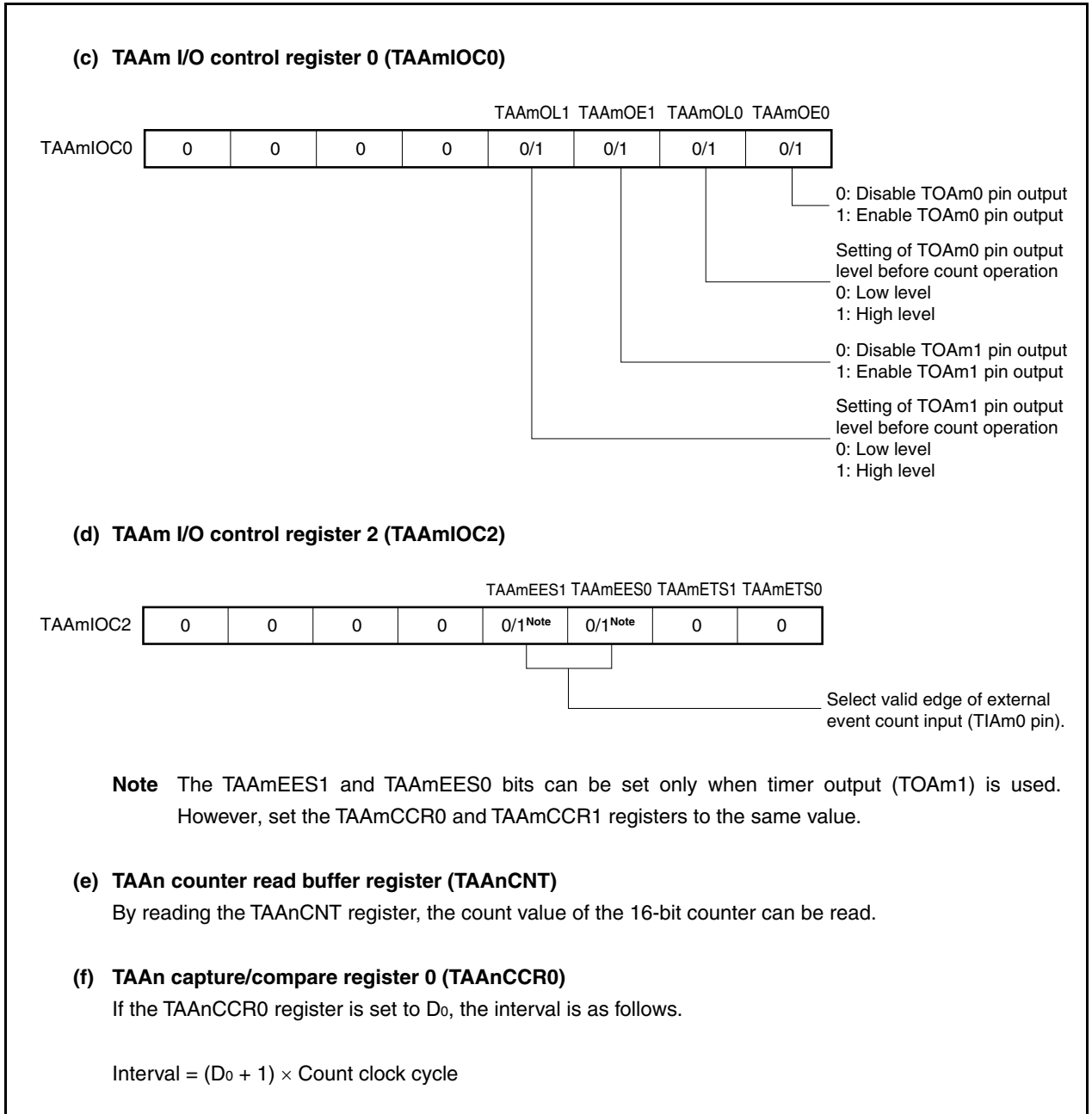


Figure 6-12. Register Setting for Interval Timer Mode Operation (3/3)

**(g) TAAAn capture/compare register 1 (TAAAnCCR1)**

The TAAAnCCR1 register is not used in the interval timer mode. However, the set value of the TAAAnCCR1 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, the TOAm1 pin output is inverted and a compare match interrupt request signal (INTTAAAnCC1) is generated.

By setting this register to the same value as the value set in the TAAAnCCR0 register, a PWM waveform with a duty factor of 50% can be output from the TOAm1 pin.

When the TAAAnCCR1 register is not used, it is recommended to set the value to FFFFH. Also mask the register by the interrupt mask flag (TAAAnCCIC1.TAAAnCCMK1).

- Remarks**
1. TAAAn I/O control register 1 (TAAAnIOC1) and TAAAn option register 0 (TAAAnOPT0) are not used in the interval timer mode.
  2. V850E/IF3: n = 0 to 4, m = 2, 4, a = 0, 1  
V850E/IG3: n = 0 to 4, m = 2 to 4, a = 0, 1

(1) Interval timer mode operation flow

Figure 6-13. Software Processing Flow in Interval Timer Mode (1/2)

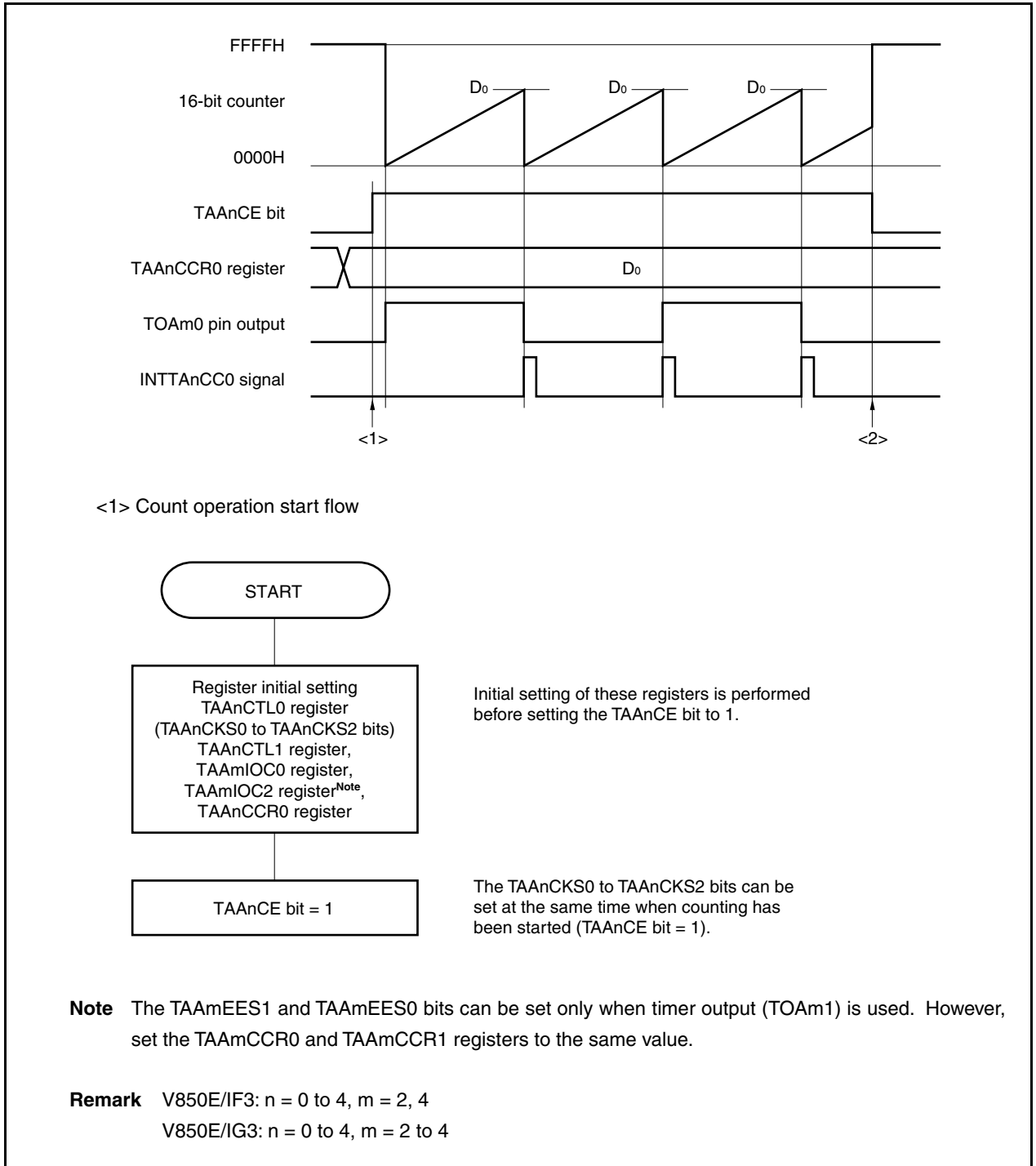
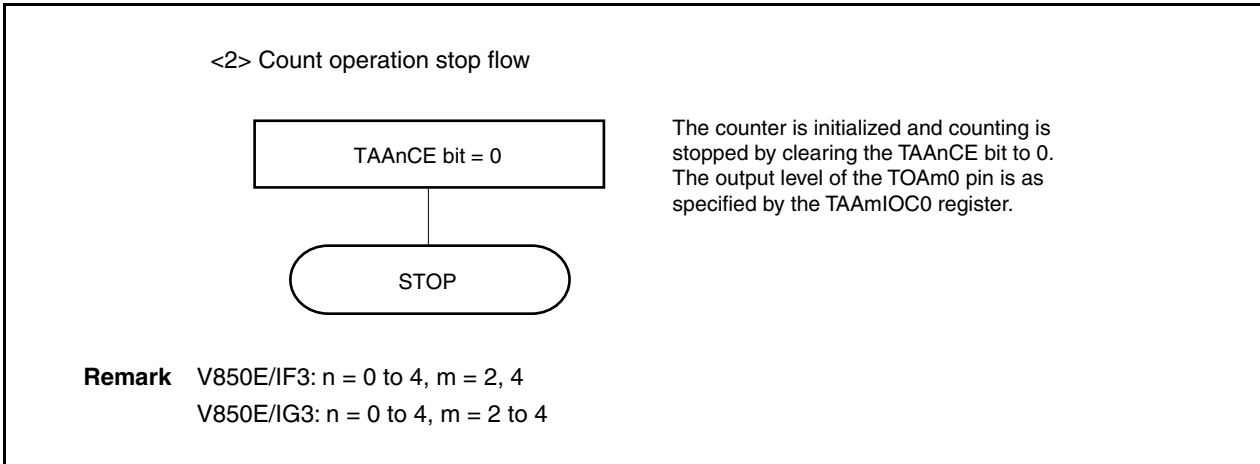


Figure 6-13. Software Processing Flow in Interval Timer Mode (2/2)

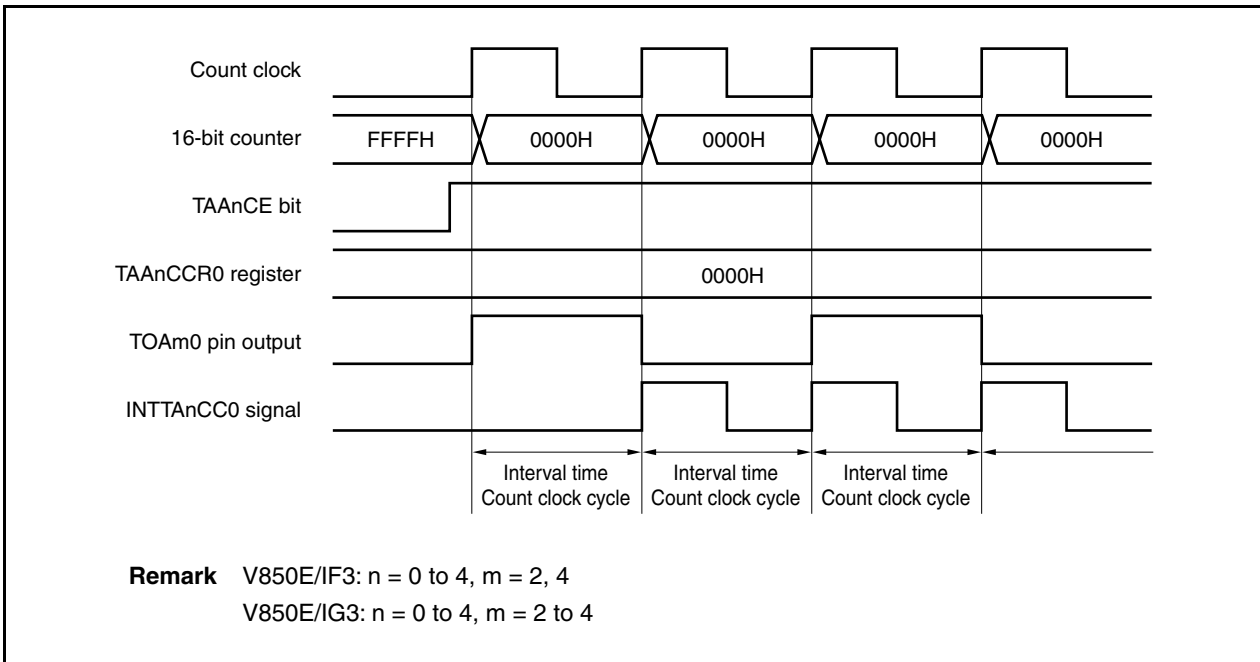


(2) Interval timer mode operation timing

(a) Operation if TAAAnCCR0 register is set to 0000H

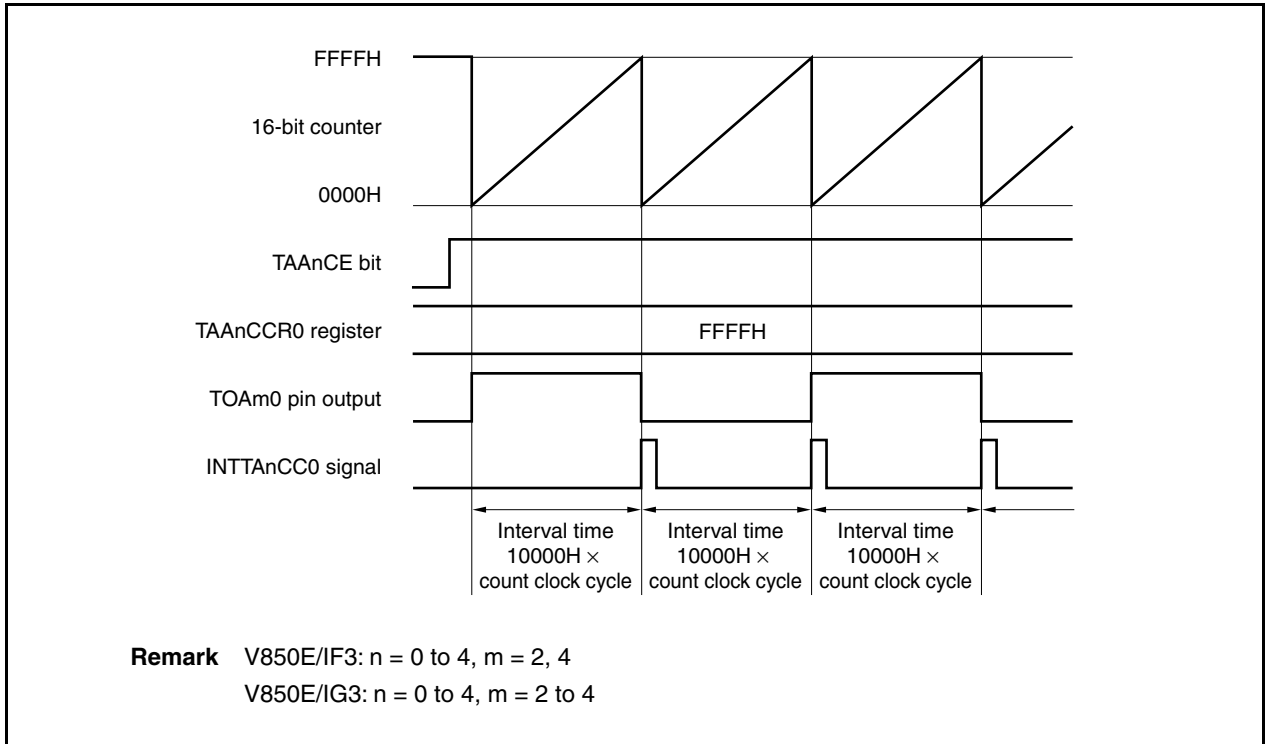
If the TAAAnCCR0 register is set to 0000H, the INTTAAAnCC0 signal is generated at each count clock, and the output of the TOAm0 pin is inverted.

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



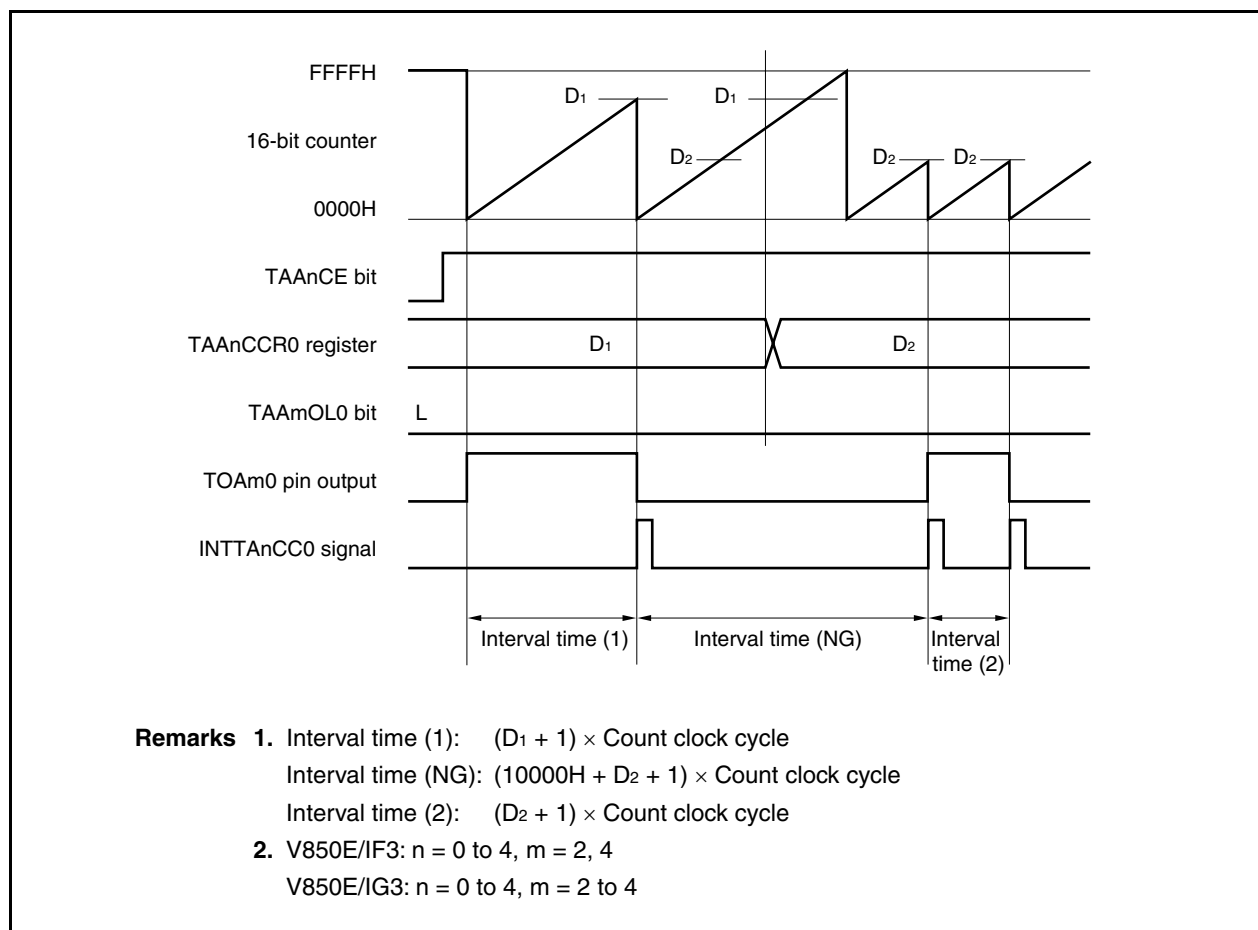
**(b) Operation if TAAAnCCR0 register is set to FFFFH**

If the TAAAnCCR0 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 INTTAnCC0 signal is generated and the output of the TOAm0 pin is inverted. At this time, an overflow interrupt request signal (INTTAnOV) is not generated, nor is the overflow flag (TAAAnOPT0.TAAAnOVF bit) set to 1.



**(c) Notes on rewriting TAAAnCCR0 register**

If the value of the TAAAnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When an overflow may occur, stop counting and then change the set value.



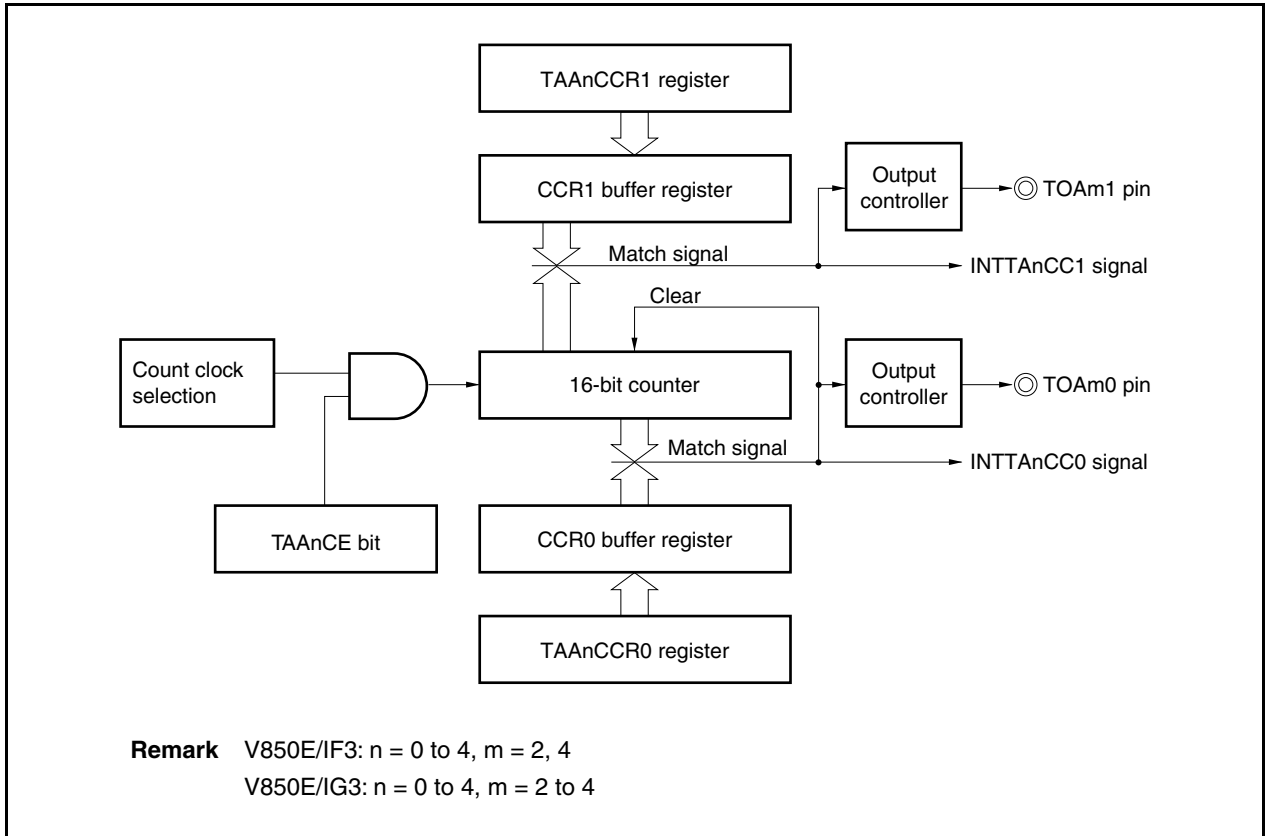
If the value of the TAAAnCCR0 register is changed from  $D_1$  to  $D_2$  while the count value is greater than  $D_2$  but less than  $D_1$ , the count value is transferred to the CCR0 buffer register as soon as the TAAAnCCR0 register has been rewritten. Consequently, the value of the 16-bit counter that is compared is  $D_2$ .

Because the count value has already exceeded  $D_2$ , however, the 16-bit counter counts up again from 0000H, overflows, and then counts up again from 0000H. When the count value matches  $D_2$ , the INTTAAAnCC0 signal is generated and the output of the TOAm0 pin is inverted.

Therefore, the INTTAAAnCC0 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 " $(10000\text{H} + D_2 + 1) \times \text{Count clock cycle}$ ".

(d) Operation of TAA<sub>n</sub>CCR1 register

Figure 6-14. Configuration of TAA<sub>n</sub>CCR1 Register

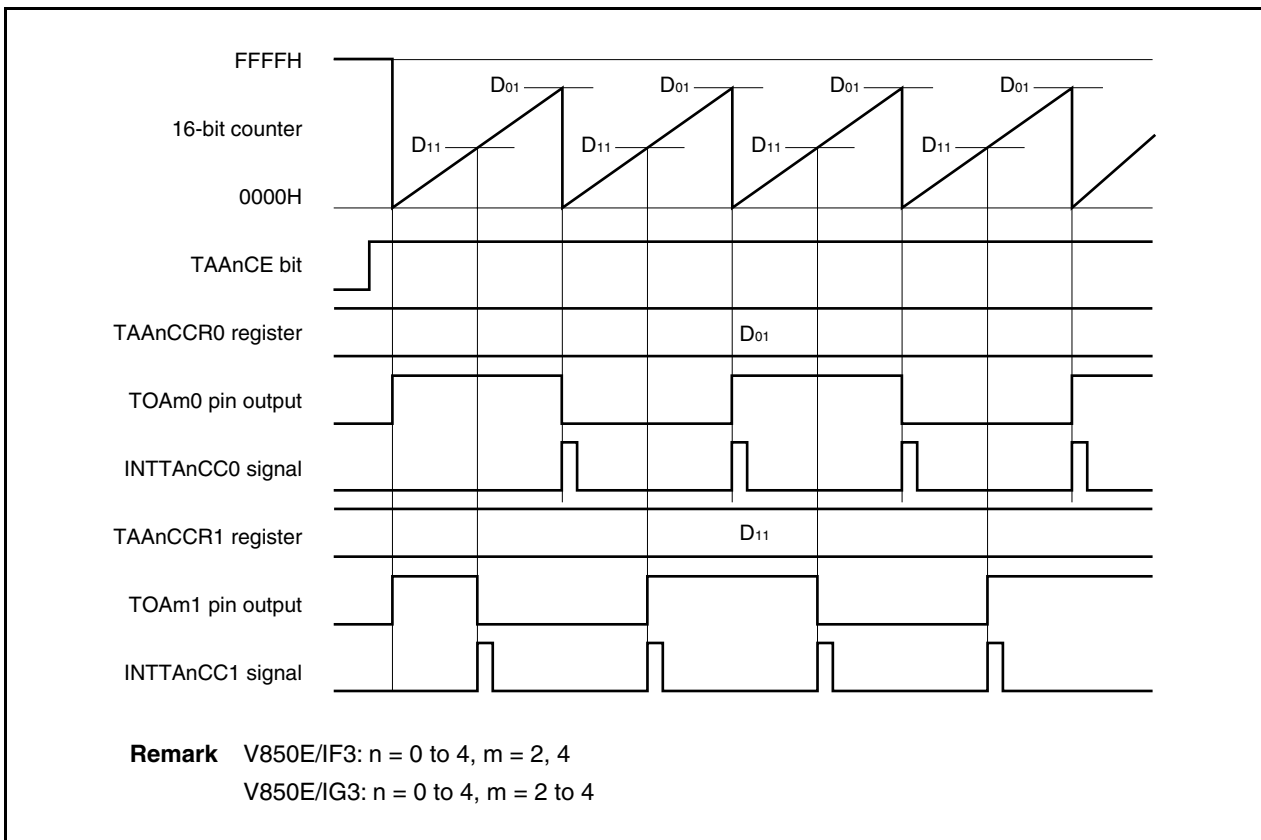


When the TAAAnCCR1 register is set to the same value as the TAAAnCCR0 register, the INTTAAAnCC0 signal is generated at the same timing as the INTTAAAnCC1 signal and the TOAm1 pin output is inverted. In other words, a PWM waveform with a duty factor of 50% can be output from the TOAm1 pin.

The following shows the operation when the TAAAnCCR1 register is set to other than the value set in the TAAAnCCR0 register.

If the set value of the TAAAnCCR1 register is less than the set value of the TAAAnCCR0 register, the INTTAAAnCC1 signal is generated once per cycle. At the same time, the output of the TOAm1 pin is inverted. The TOAm1 pin outputs a PWM waveform with a duty factor of 50% after outputting a short-width pulse.

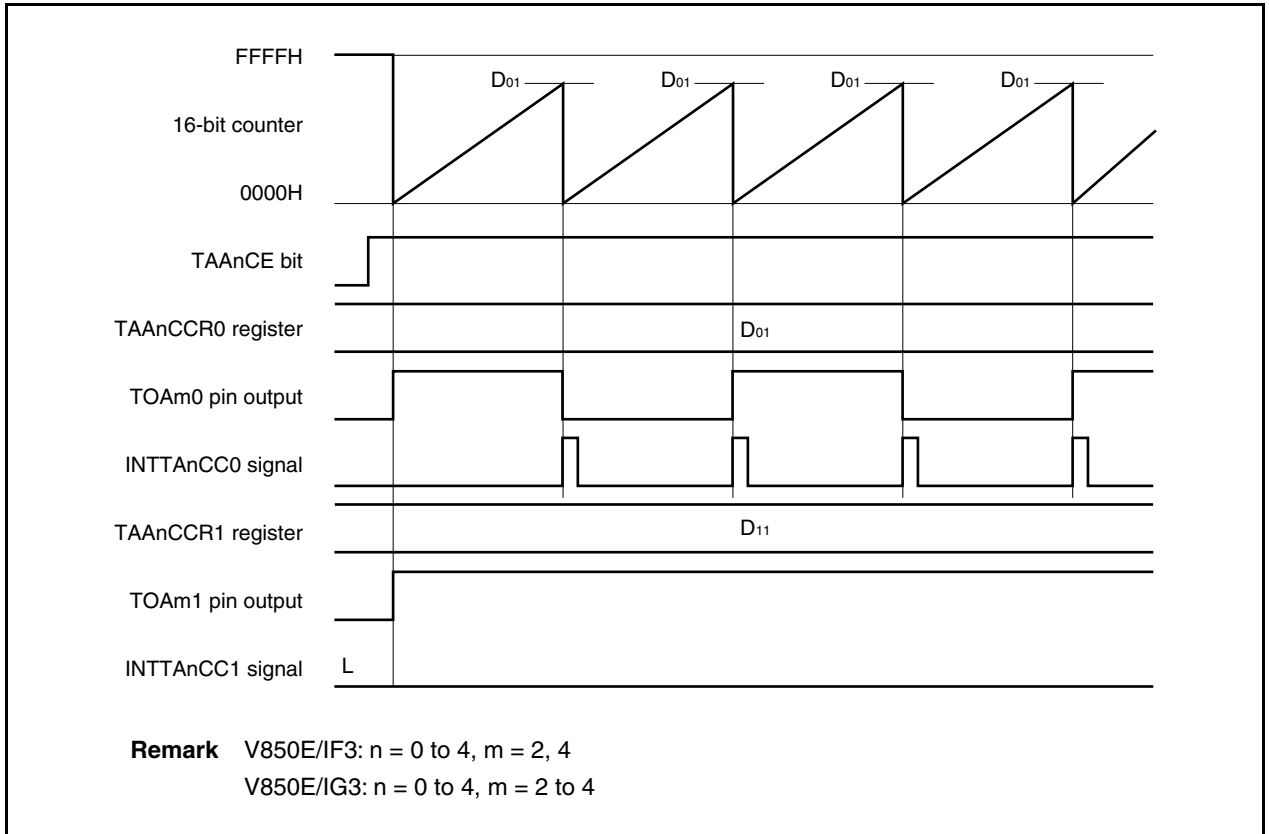
Figure 6-15. Timing Chart When  $D_{01} \geq D_{11}$





If the set value of the TAAAnCCR1 register is greater than the set value of the TAAAnCCR0 register, the count value of the 16-bit counter does not match the value of the TAAAnCCR1 register. Consequently, the INTTAAAnCC1 signal is not generated, nor is the output of the TOAm1 pin changed. When the TAAAnCCR1 register is not used, it is recommended to set its value to FFFFH.

Figure 6-16. Timing Chart When  $D_{01} < D_{11}$



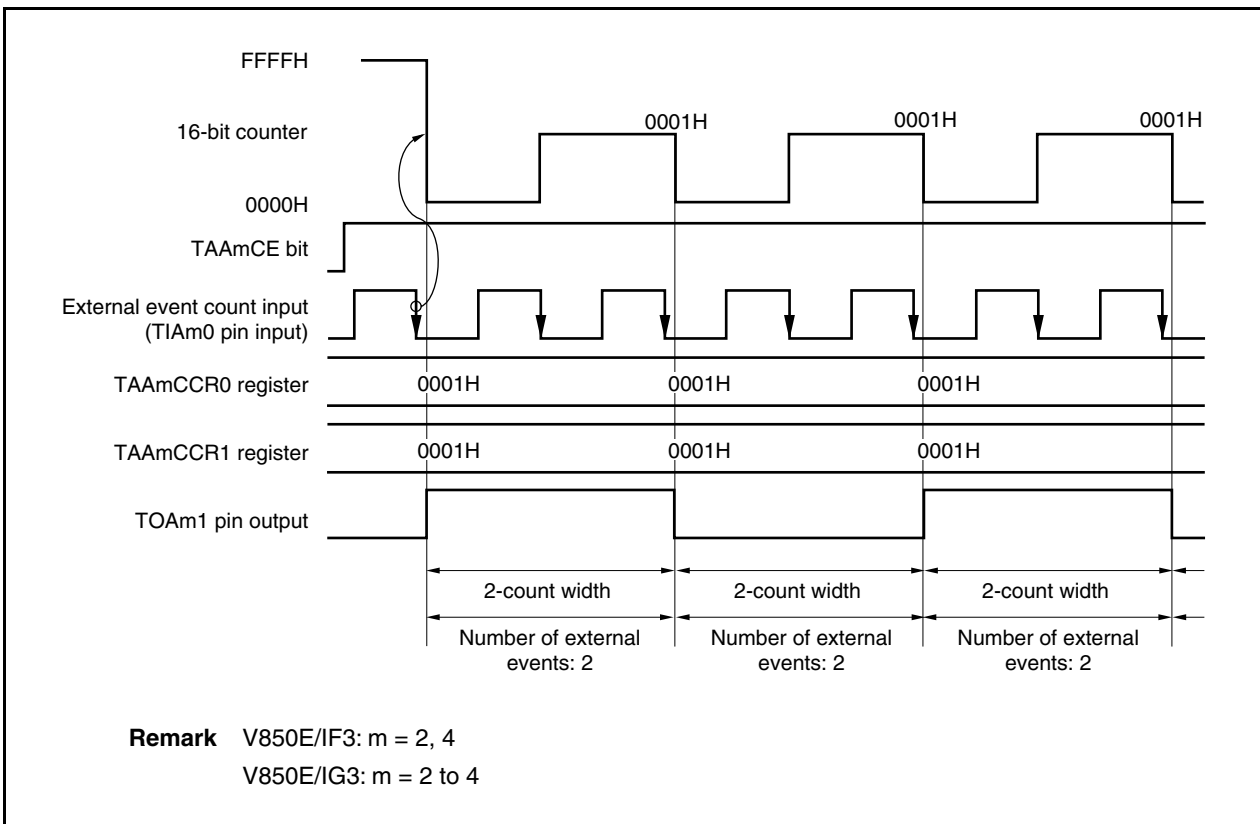
(3) Operation by external event count input (TIAM0)

(a) Operation

To count the 16-bit counter at the valid edge of the external event count input (TIAM0) in the interval timer mode, the 16-bit counter is cleared from FFFFH to 0000H by the valid edge of the external event count after the TAAmCE bit is set from 0 to 1.

When 0001H is set to both the TAAmCCR0 and TAAmCCR1 registers, the TOAm1 pin output is inverted each time the 16-bit counter counts twice.

The TAAmCTL1.TAAmEEE bit can be set to 1 in the interval timer mode only when the timer output (TOAm1) is used with the external event count input.



**6.6.2 External event count mode (TAAmMD2 to TAAmMD0 bits = 001)**

This mode is valid only in TAA2, TAA3 (V850E/IG3 only), and TAA4.

In the external event count mode, the valid edge of the external event count input (TIAm0) is counted when the TAAmCTL0.TAAmCE bit is set to 1, and an interrupt request signal (INTTAmCC0) is generated each time the number of edges set by the TAAmCCR0 register have been counted. The TOAm0 and TOAm1 pins cannot be used. When using the TOAm1 pin for external event count input, set the TAAmCTL1.TAAmEEE bit to 1 in the interval timer mode (see 6.6.1 (3) **Operation by external event count input (TIAm0)**).

The TAAmCCR1 register is not used in the external event count mode.

**Figure 6-17. Configuration in External Event Count Mode**

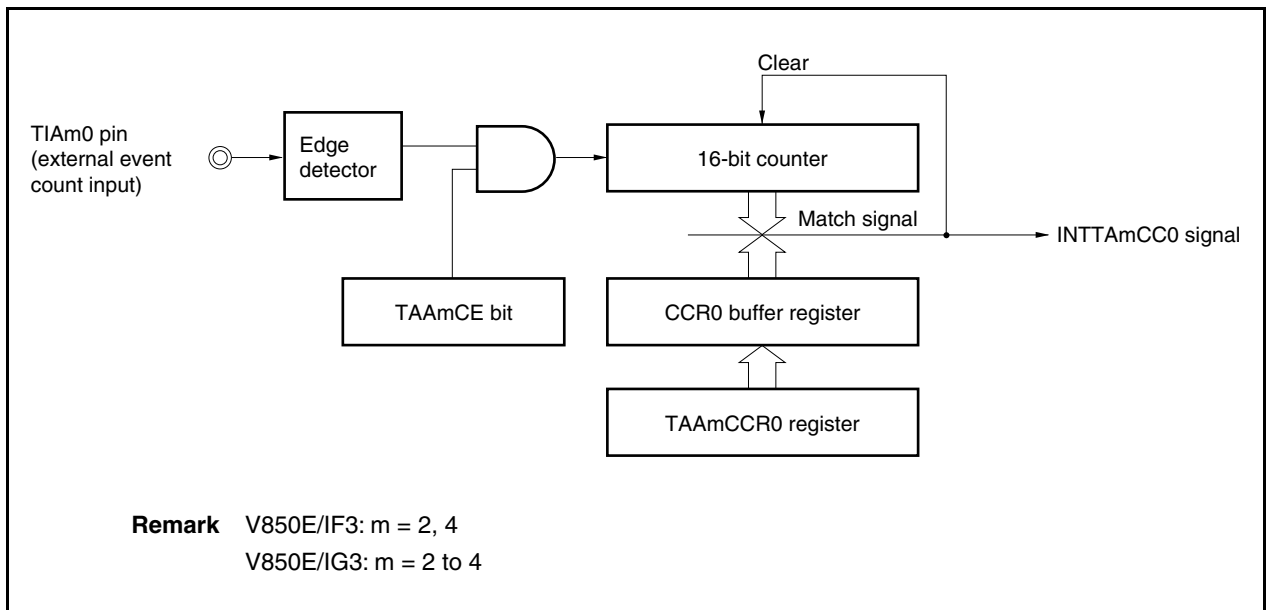
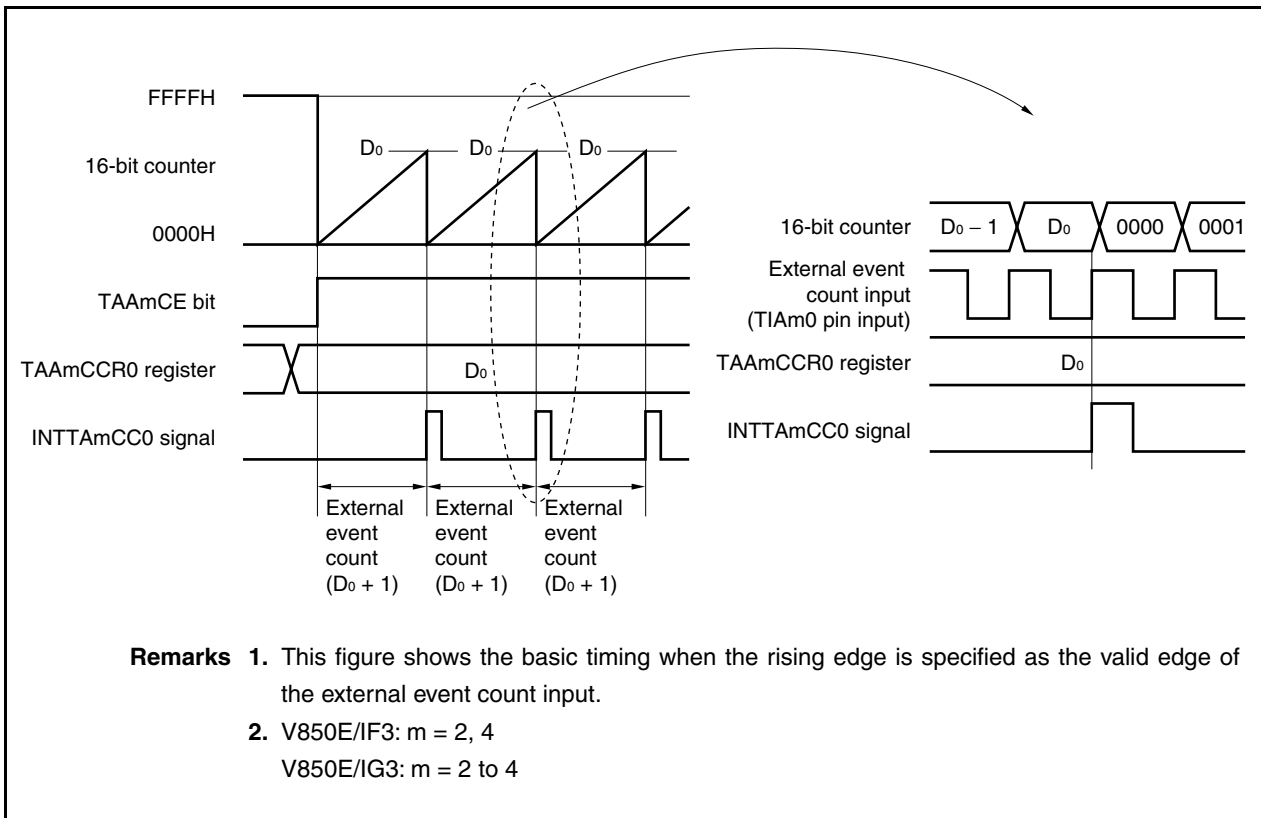


Figure 6-18. Basic Timing in External Event Count Mode



When the TAAmCE 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 TAAmCCR0 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 (INTTAmCC0) is generated.

The INTTAmCC0 signal is generated each time the valid edge of the external event count input has been detected “value set to TAAmCCR0 register + 1” times.

**Figure 6-19. Register Setting for Operation in External Event Count Mode (1/2)**

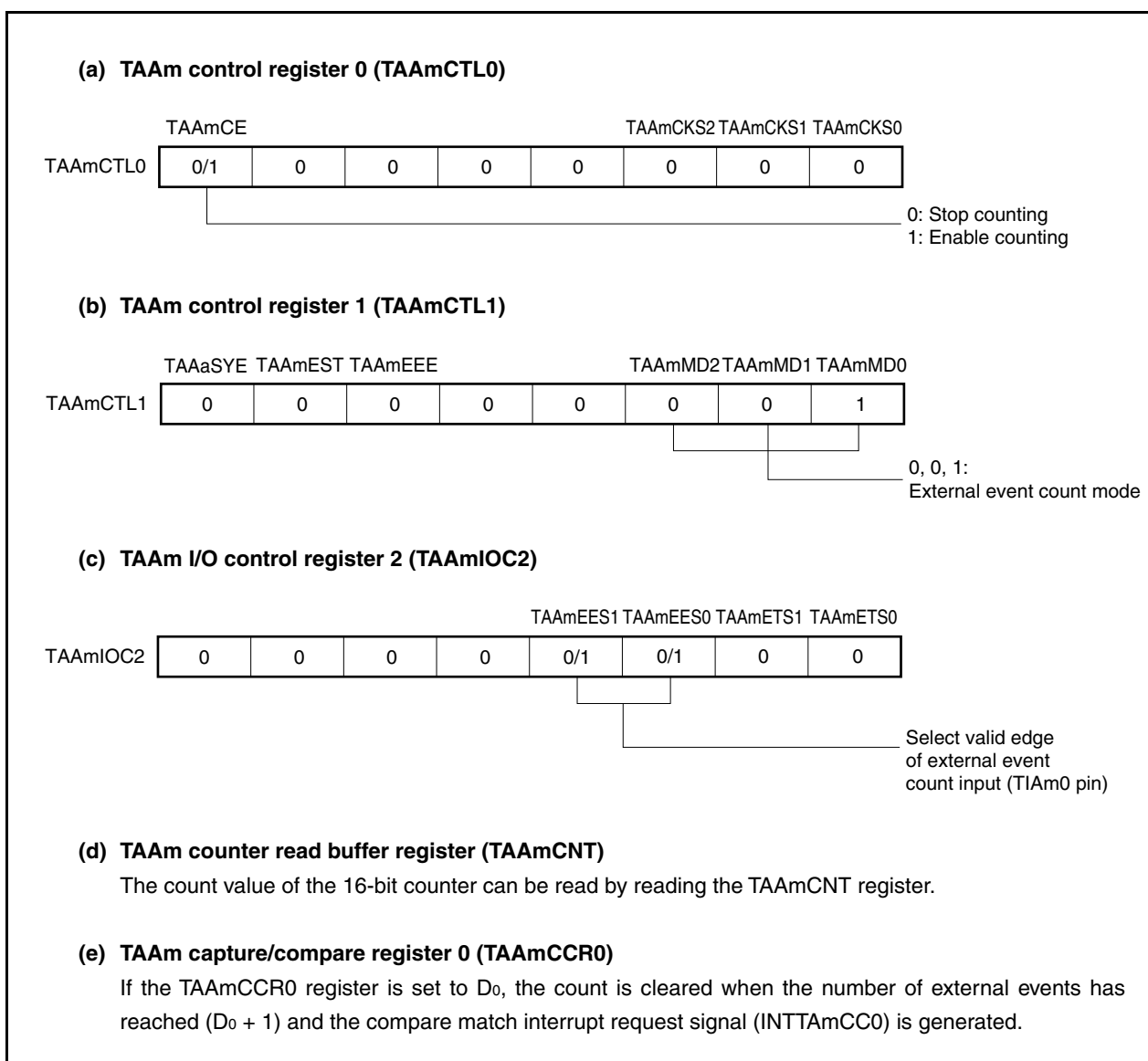


Figure 6-19. Register Setting for Operation in External Event Count Mode (2/2)

**(f) TAAm capture/compare register 1 (TAAmCCR1)**

The TAAmCCR1 register is not used in the external event count mode. However, the set value of the TAAmCCR1 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 (INTTAmCC1) is generated.

When the TAAmCCR1 register is not used, it is recommended to set the value to FFFFH. Also mask the register by the interrupt mask flag (TAAmCCIC1.TAAmCCMK1).

**Cautions** 1. Set the TAAmIOC0 register to 00H.

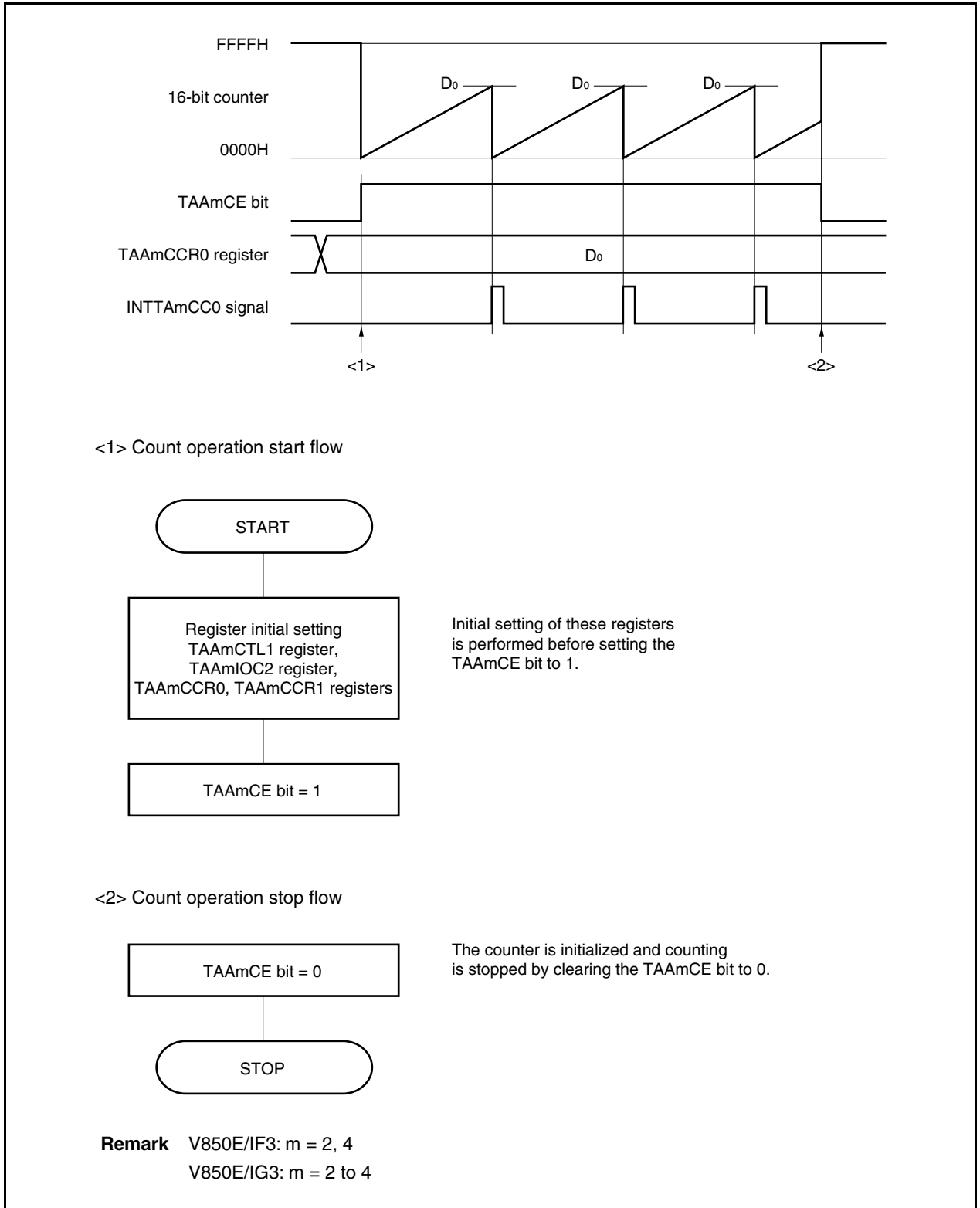
2. When an external clock is used as the count clock, the external clock can be input only from the TIAm0 pin. At this time, set the TAAmIOC1.TAAmIS1 and TAAmIOC1.TAAmIS0 bits to 00 (capture trigger input (TIAm0 pin): no edge detection)

**Remarks** 1. TAAm I/O control register 1 (TAAmIOC1) and TAAm option register 0 (TAAmOPT0) are not used in the external event count mode.

2. V850E/IF3: m = 2, 4, a = 0, 1  
V850E/IG3: m = 2 to 4, a = 0, 1

(1) External event count mode operation flow

Figure 6-20. Software Processing Flow in External Event Count Mode



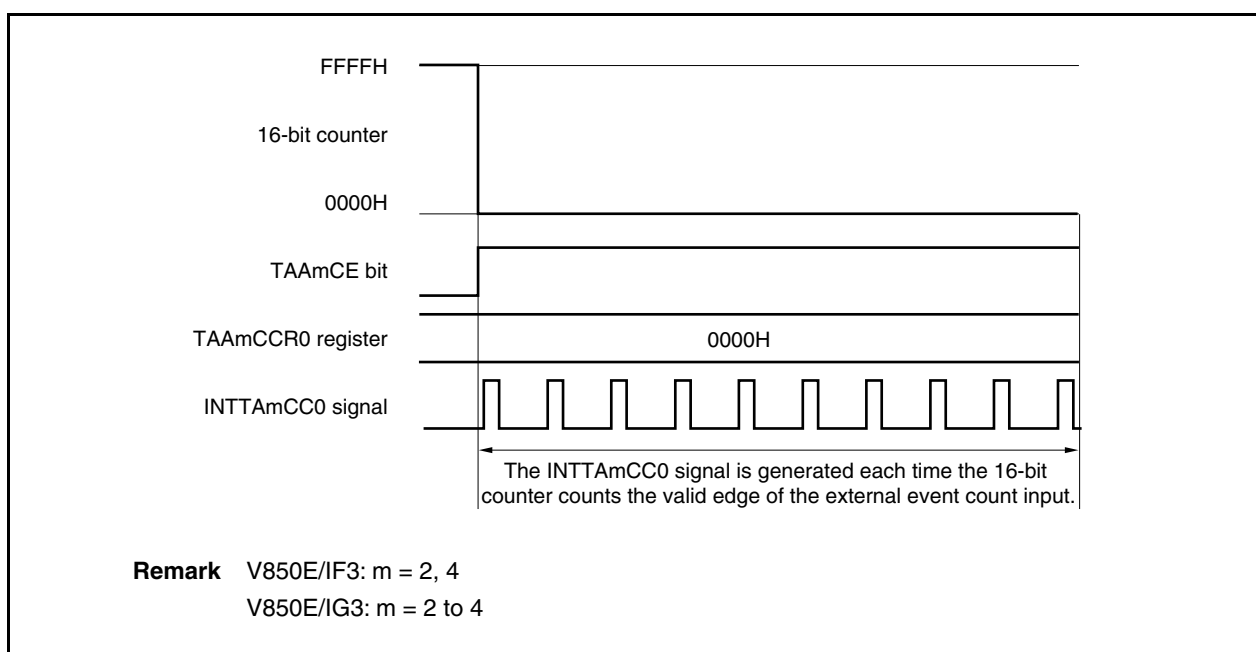
(2) Operation timing in external event count mode

**Caution** In the external event count mode, use of the timer output (TOAm0, TOAm1) is disabled. If using timer output (TOAm1) with external event count input (TIAm0), set the interval timer mode, and select the operation enabled by the external event count input for the count clock (TAAmCTL1.TAAmEEE bit = 1) (see 6.6.1 (3) Operation by external event count input (TIAm0)).

(a) Operation if TAAmCCR0 register is set to 0000H

When the TAAmCCR0 register is set to 0000H, the 16-bit counter is repeatedly cleared to 0000H and generates the INTTAmCC0 signal each time it has detected the valid edge of the external event count signal and its value has matched that of the CCR0 buffer register.

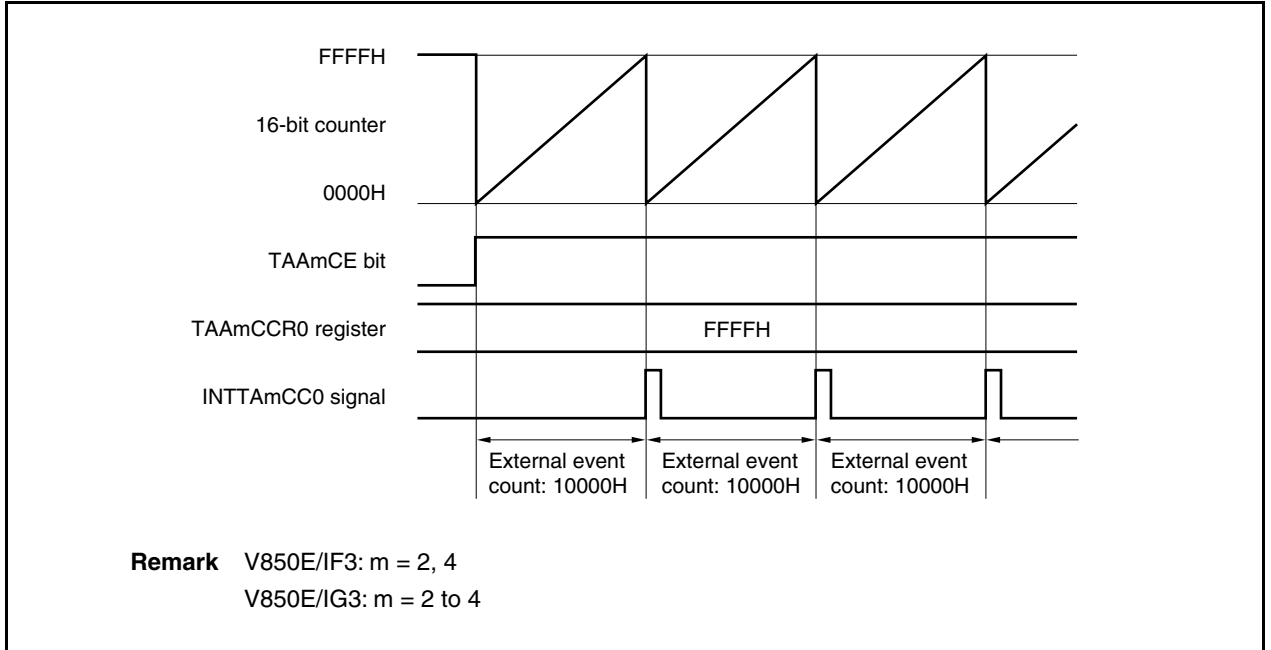
The value of the 16-bit counter is always 0000H.





**(b) Operation if TAAmCCR0 register is set to FFFFH**

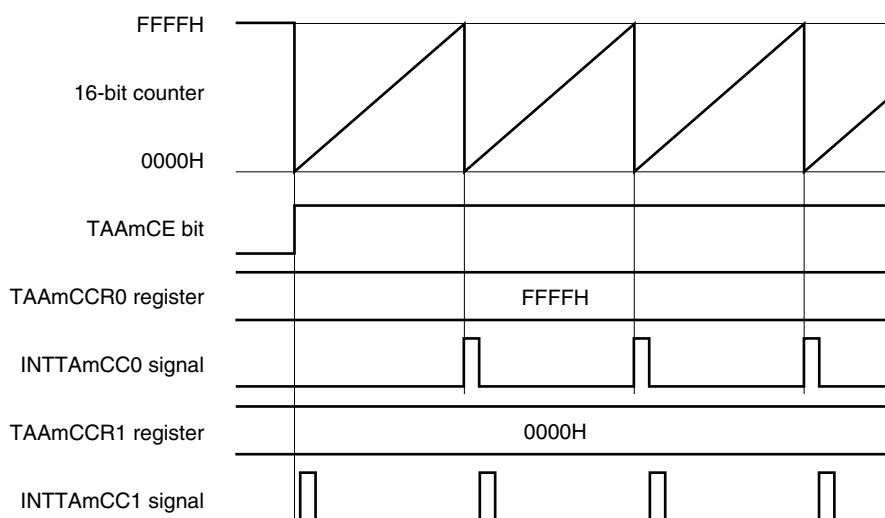
If the TAAmCCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH each time the valid edge of the external event count signal has been detected. The 16-bit counter is cleared to 0000H in synchronization with the next count-up timing, and the INTTAmCC0 signal is generated. At this time, the TAAmOPT0.TAAmOVF bit is not set.



**(c) Operation with TAAmCCR0 register set to FFFFH and TAAmCCR1 register to 0000H**

When the TAAmCCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH each time it has detected the valid edge of the external event count signal. The counter is then cleared to 0000H in synchronization with the next count-up timing and the INTTAmCC0 signal is generated. At this time, the TAAmOPT0.TAAmOVF bit is not set.

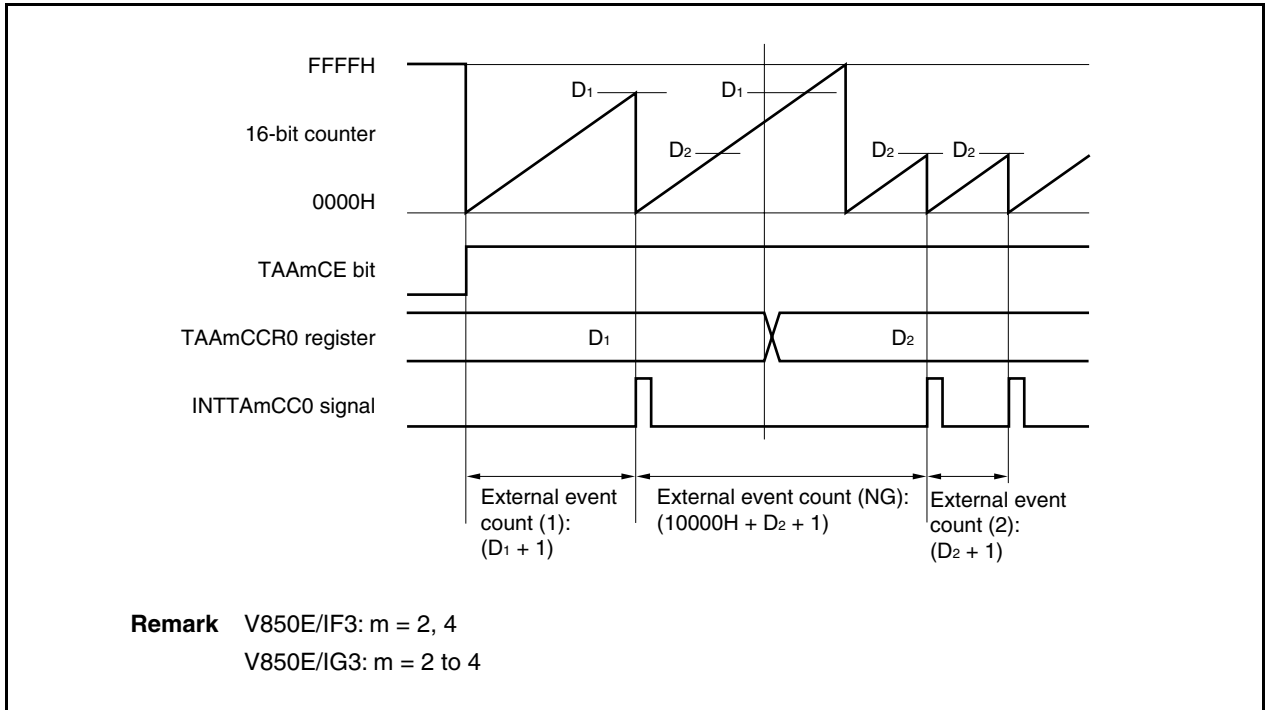
If the TAAmCCR1 register is set to 0000H, the INTTAmCC1 signal is generated when the 16-bit counter is cleared to 0000H.



**Remark** V850E/IF3: m = 2, 4  
 V850E/IG3: m = 2 to 4

**(d) Notes on rewriting the TAAmCCR0 register**

If the value of the TAAmCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When the overflow may occur, stop counting once and then change the set value.

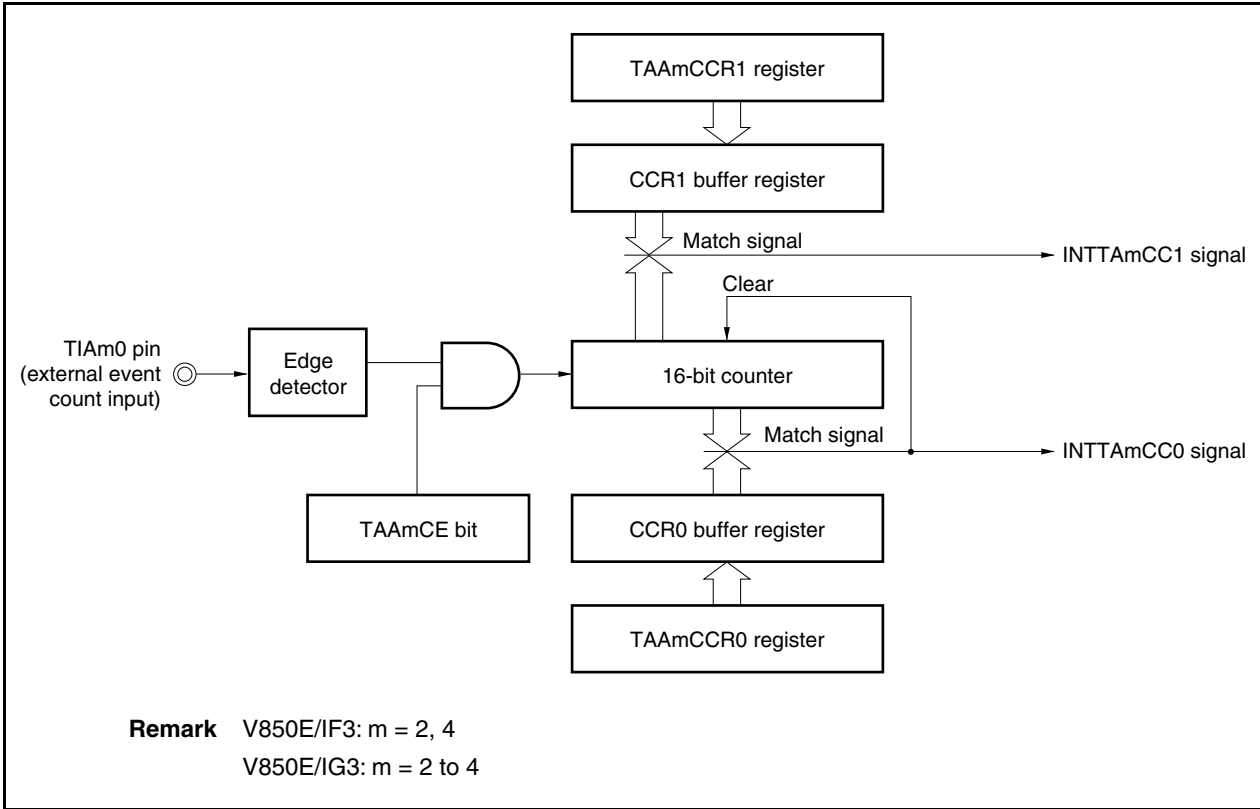


If the value of the TAAmCCR0 register is changed from  $D_1$  to  $D_2$  while the count value is greater than  $D_2$  but less than  $D_1$ , the count value is transferred to the CCR0 buffer register as soon as the TAAmCCR0 register has been rewritten. Consequently, the value that is compared with the 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 INTTAmCC0 signal is generated.

Therefore, the INTTAmCC0 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”.

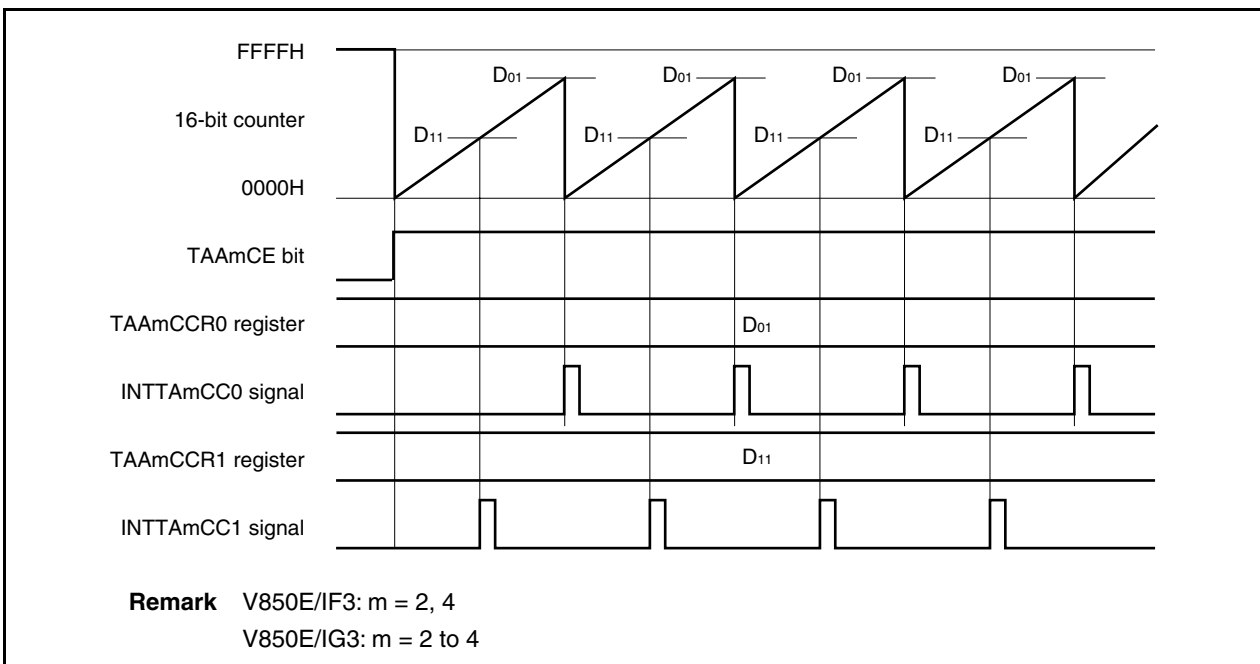
(e) Operation of TAAmCCR1 register

Figure 6-21. Configuration of TAAmCCR1 Register



If the set value of the TAAmCCR1 register is smaller than the set value of the TAAmCCR0 register, the INTTAmCC1 signal is generated once per cycle.

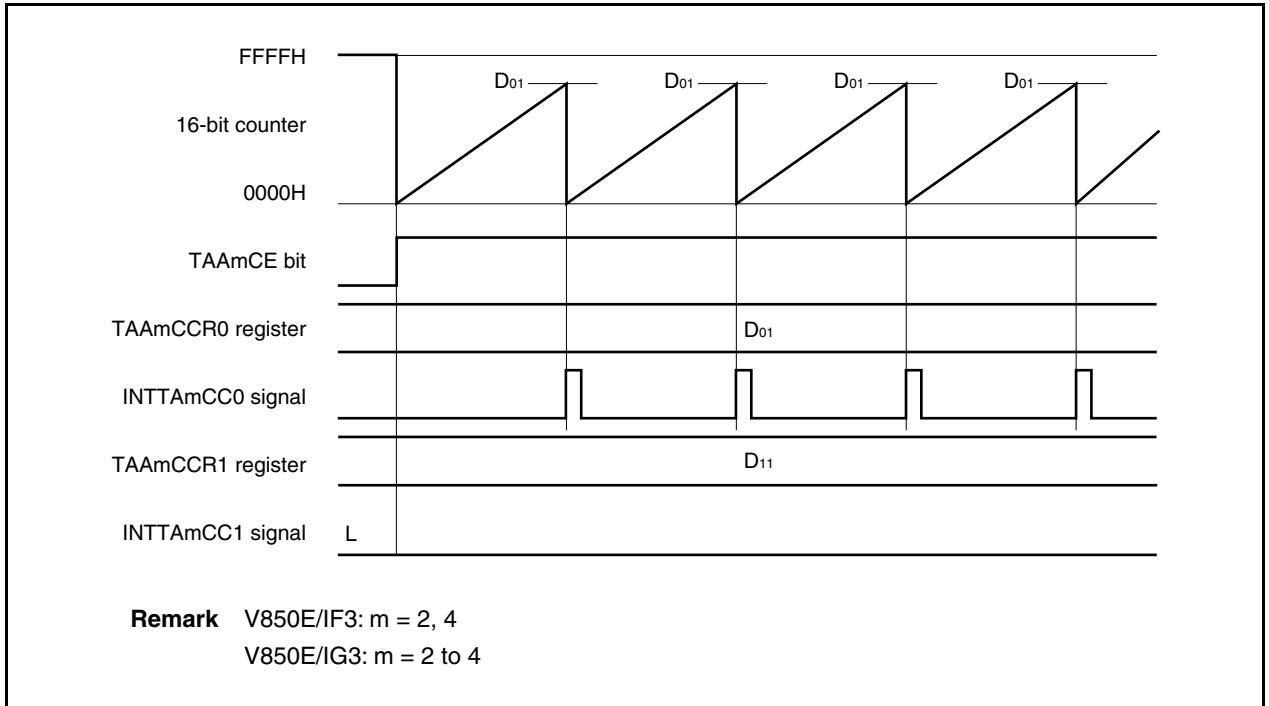
Figure 6-22. Timing Chart When  $D_{01} \geq D_{11}$



If the set value of the TAAmCCR1 register is greater than the set value of the TAAmCCR0 register, the INTTAmCC1 signal is not generated because the count value of the 16-bit counter and the value of the TAAmCCR1 register do not match.

When the TAAmCCR1 register is not used, it is recommended to set its value to FFFFH.

Figure 6-23. Timing Chart When  $D_{01} < D_{11}$



**6.6.3 External trigger pulse output mode (TAAmMD2 to TAAmMD0 bits = 010)**

This mode is valid only in TAA2, TAA3 (V850E/IG3 only), and TAA4.

In the external trigger pulse output mode, 16-bit timer/event counter AA waits for a trigger when the TAAmCTL0.TAAmCE bit is set to 1. When the valid edge of an external trigger input (TIAm0) is detected, 16-bit timer/event counter AA starts counting, and outputs a PWM waveform from the TOAm1 pin.

Pulses can also be output by generating a software trigger instead of using the external trigger. When using a software trigger, a PWM waveform with a duty factor of 50% that has the set value of the TAAmCCR0 register + 1 as half its cycle can also be output from the TOAm0 pin.

**Figure 6-24. Configuration in External Trigger Pulse Output Mode**

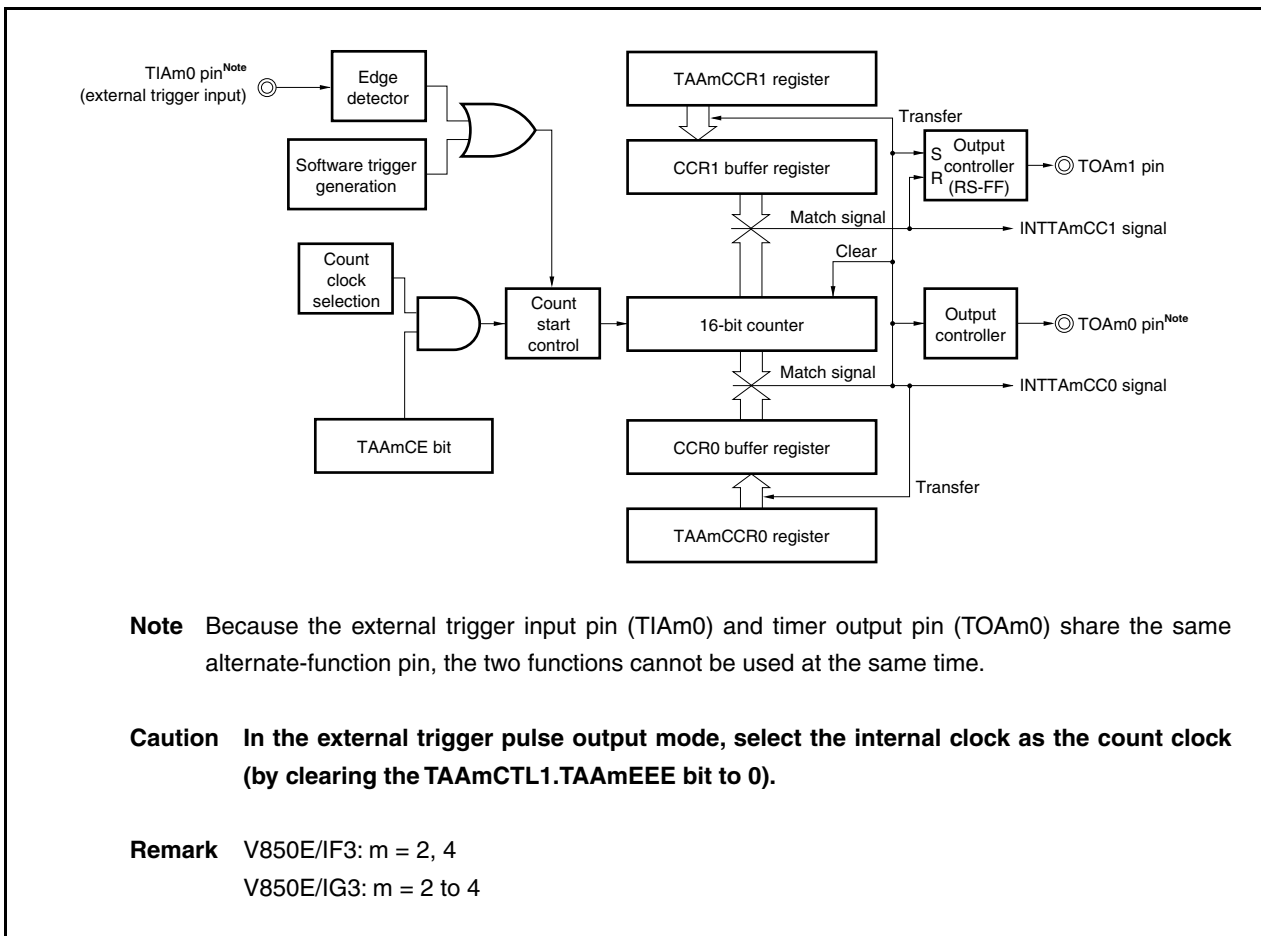
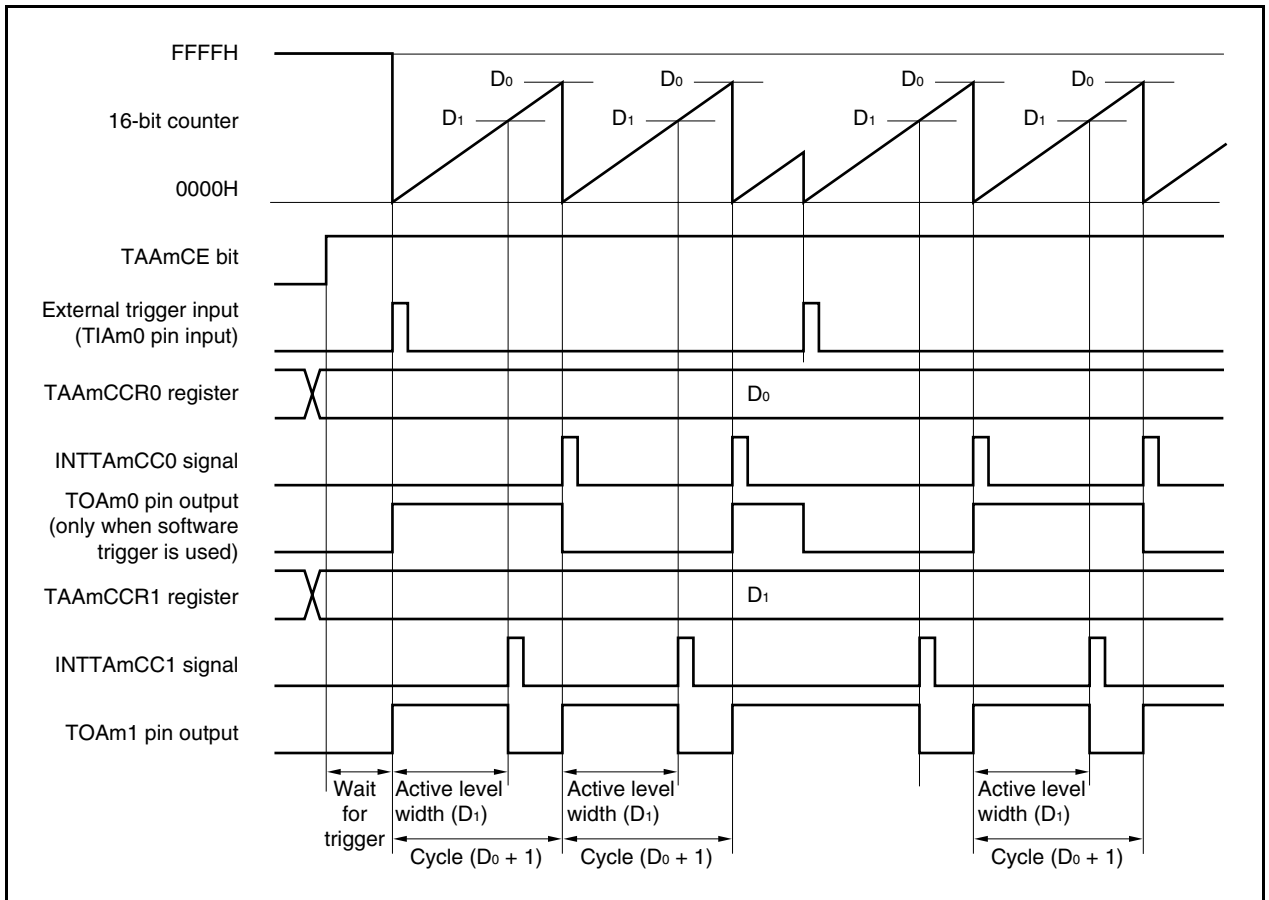


Figure 6-25. Basic Timing in External Trigger Pulse Output Mode



16-bit timer/event counter AA waits for a trigger when the TAAmCE 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 TOAm1 pin. If the trigger is generated again while the counter is operating, the counter is cleared to 0000H and restarted. (The output of the TOAm0 pin is inverted. The TOAm1 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.

$$\text{Active level width} = (\text{Set value of TAAmCCR1 register}) \times \text{Count clock cycle}$$

$$\text{Cycle} = (\text{Set value of TAAmCCR0 register} + 1) \times \text{Count clock cycle}$$

$$\text{Duty factor} = (\text{Set value of TAAmCCR1 register}) / (\text{Set value of TAAmCCR0 register} + 1)$$

The compare match interrupt request signal INTTAmCC0 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 INTTAmCC1 is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

The value set to the TAAmCCRa register is transferred to the CCRa buffer register when the count value of the 16-bit counter matches the value of the CCRa buffer register and the 16-bit counter is cleared to 0000H.

The valid edge of an external trigger input (TIAm0), or setting the software trigger (TAAmCTL1.TAAmEST bit) to 1 is used as the trigger.

**Remark** V850E/IF3: m = 2, 4, a = 0, 1, V850E/IG3: m = 2 to 4, a = 0, 1

Figure 6-26. Setting of Registers in External Trigger Pulse Output Mode (1/2)

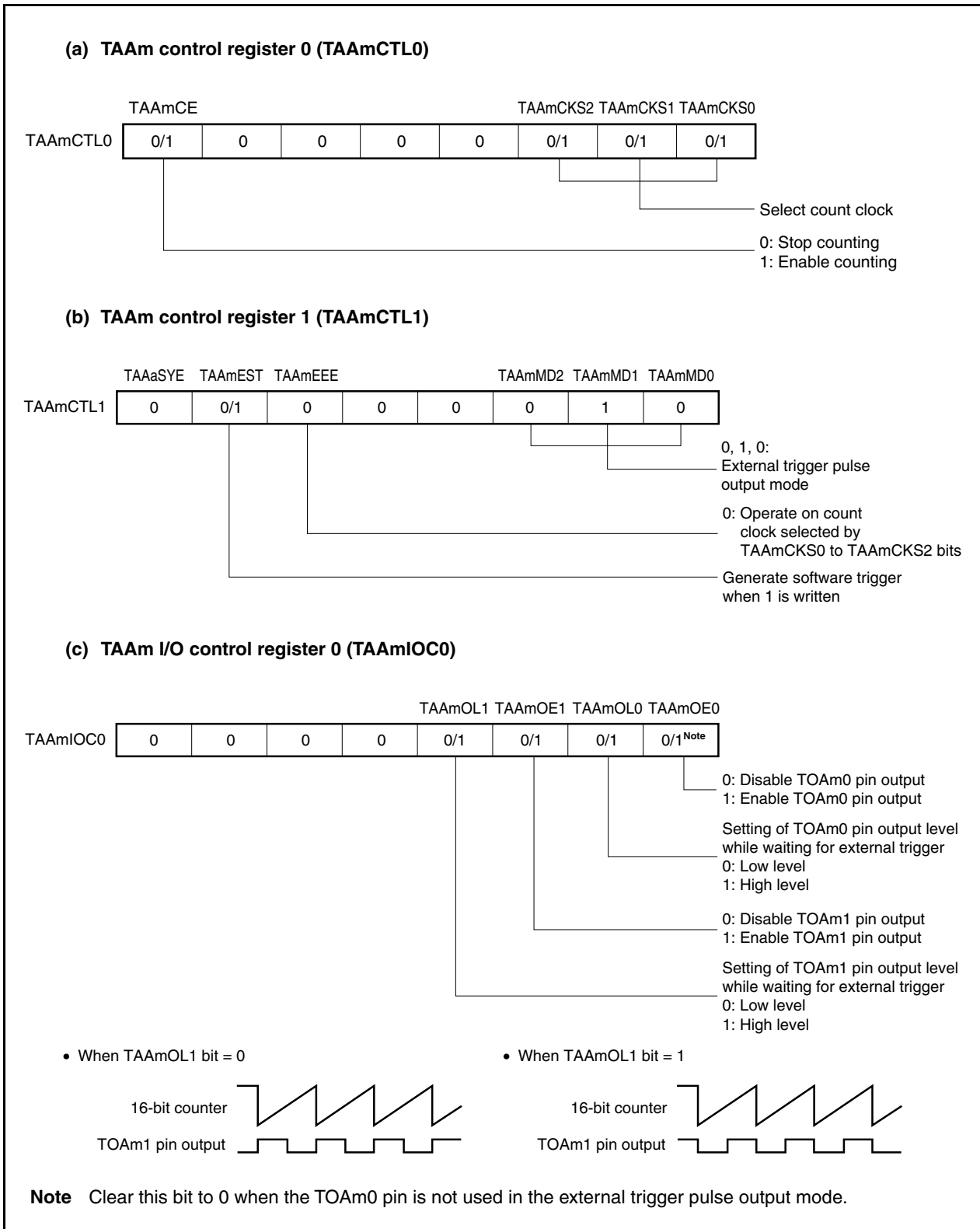
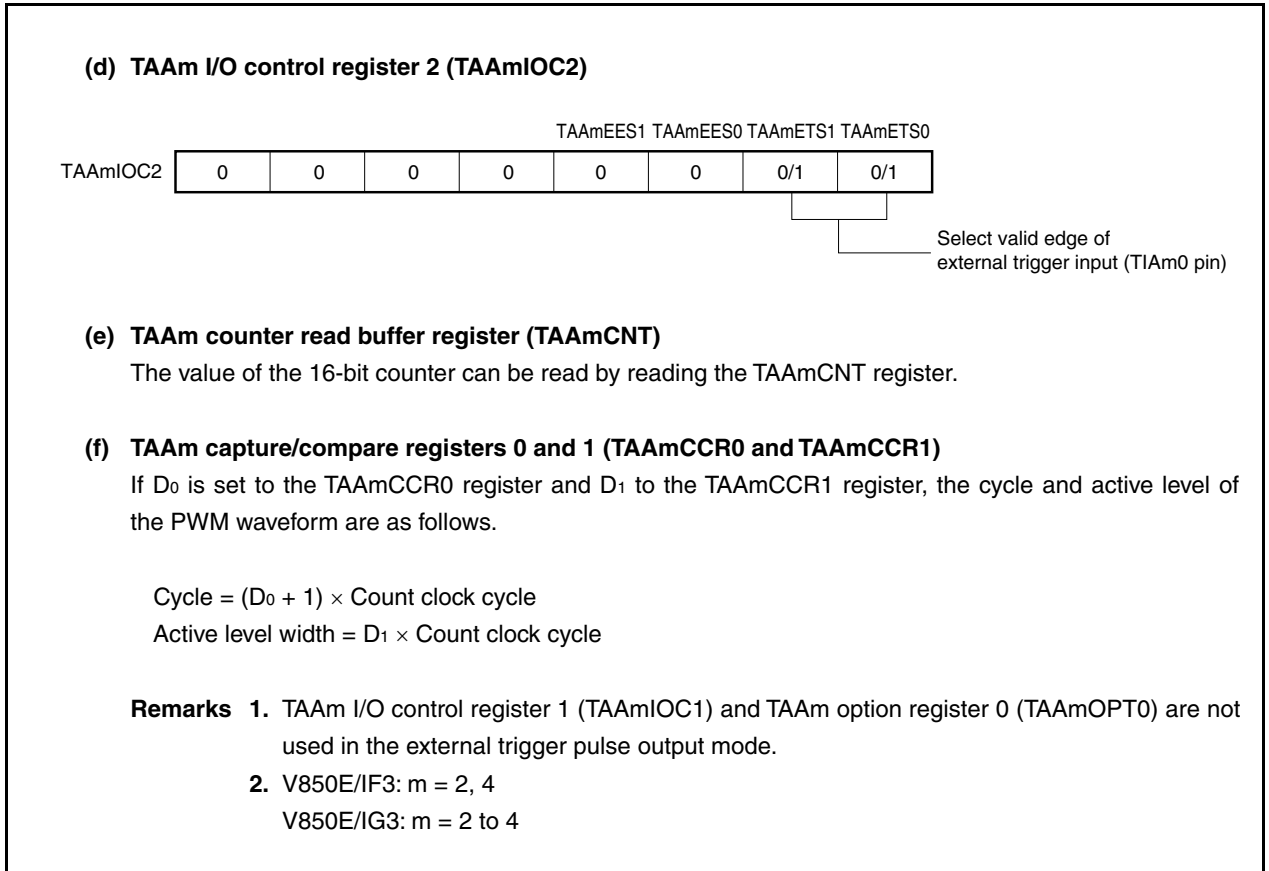




Figure 6-26. Setting of Registers in External Trigger Pulse Output Mode (2/2)



(1) Operation flow in external trigger pulse output mode

Figure 6-27. Software Processing Flow in External Trigger Pulse Output Mode (1/2)

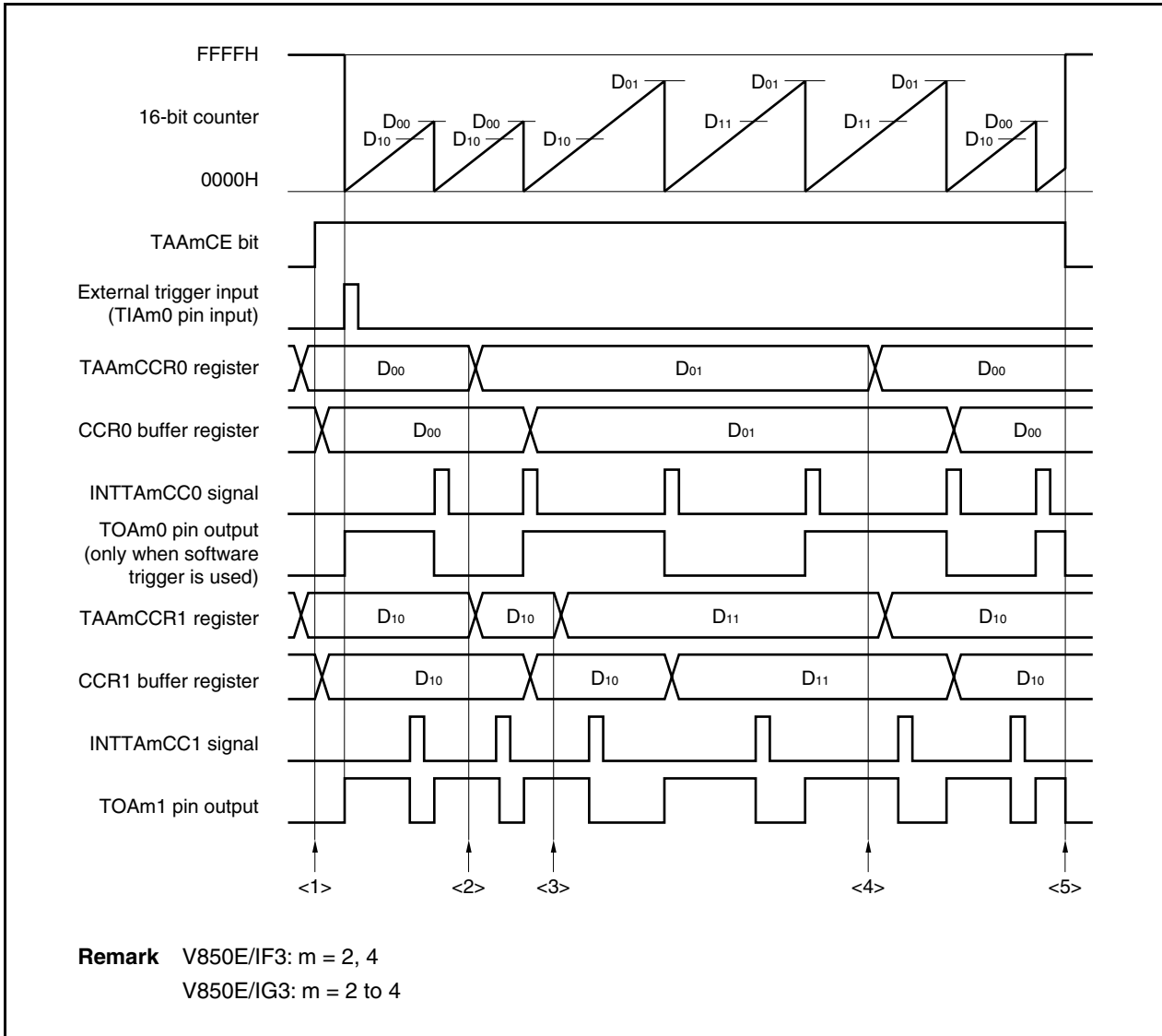
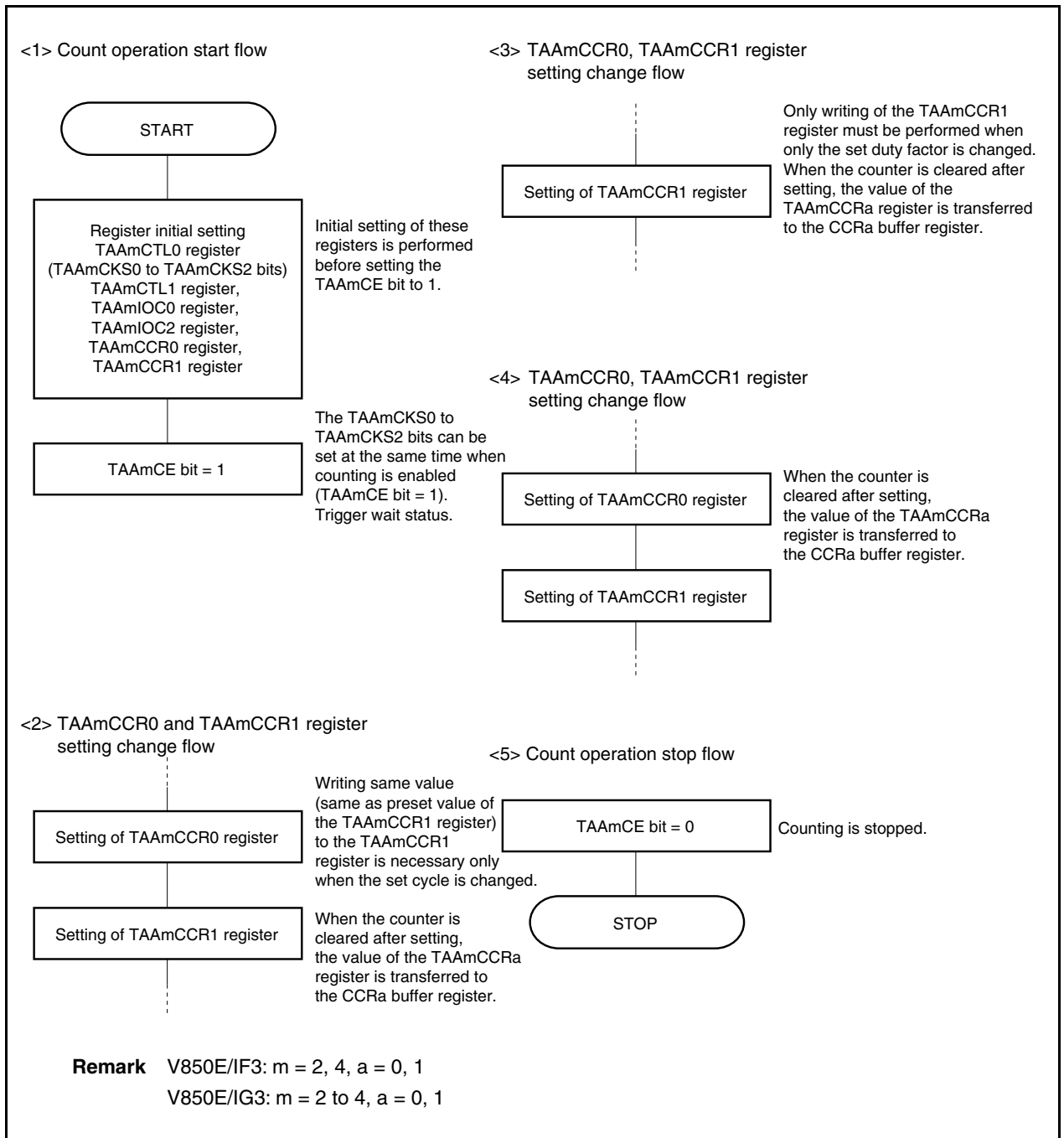


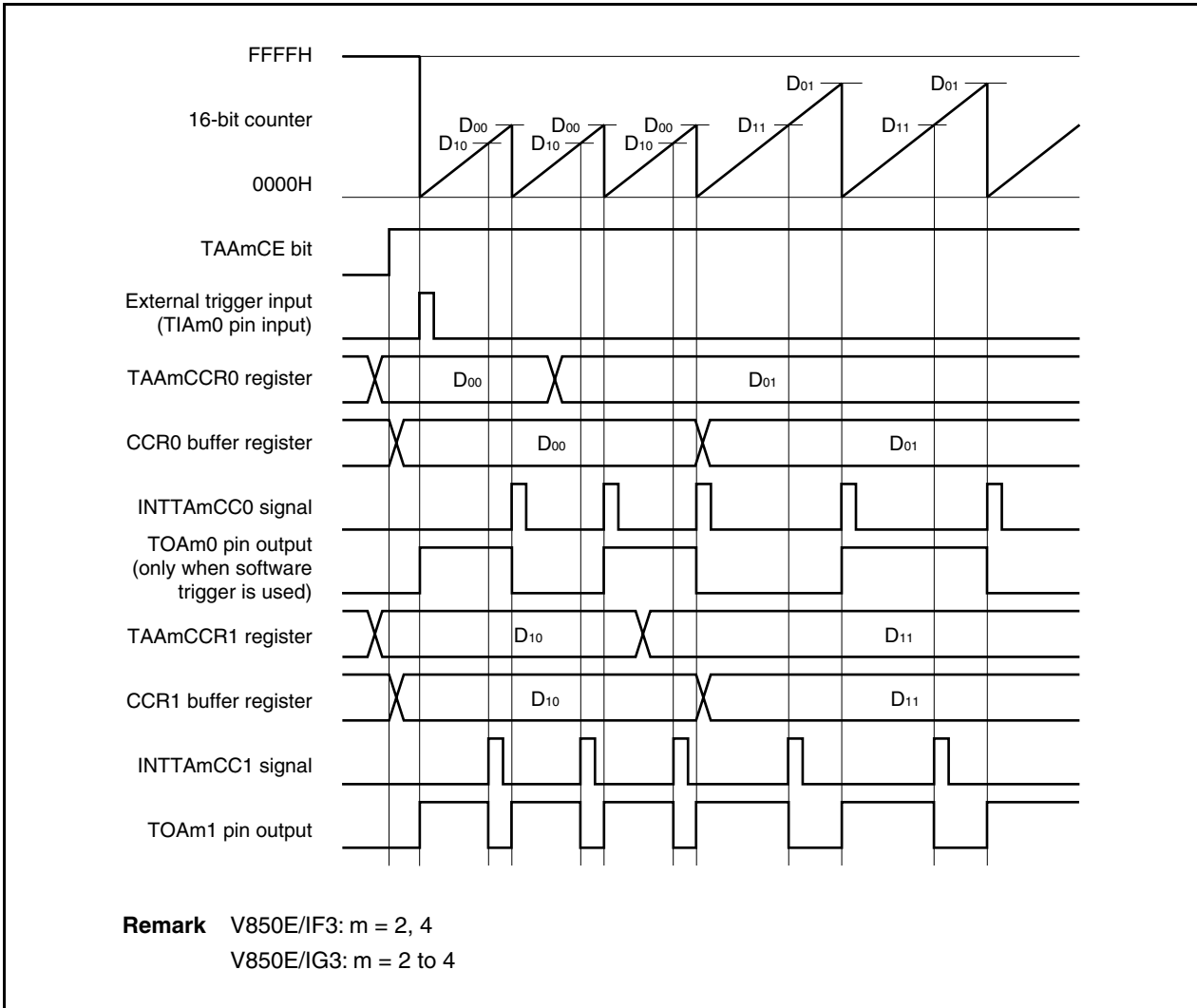
Figure 6-27. Software Processing Flow in External Trigger Pulse Output Mode (2/2)



(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 TAAmCCR1 register last. Rewrite the TAAmCCRa register after writing the TAAmCCR1 register after the INTTAmCC0 signal is detected.



In order to transfer data from the TAAmCCRa register to the CCRa buffer register, the TAAmCCR1 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 TAAmCCR0 register and then set the active level width to the TAAmCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TAAmCCR0 register, and then write the same value (same as preset value of the TAAmCCR1 register) to the TAAmCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, only the TAAmCCR1 register has to be set.

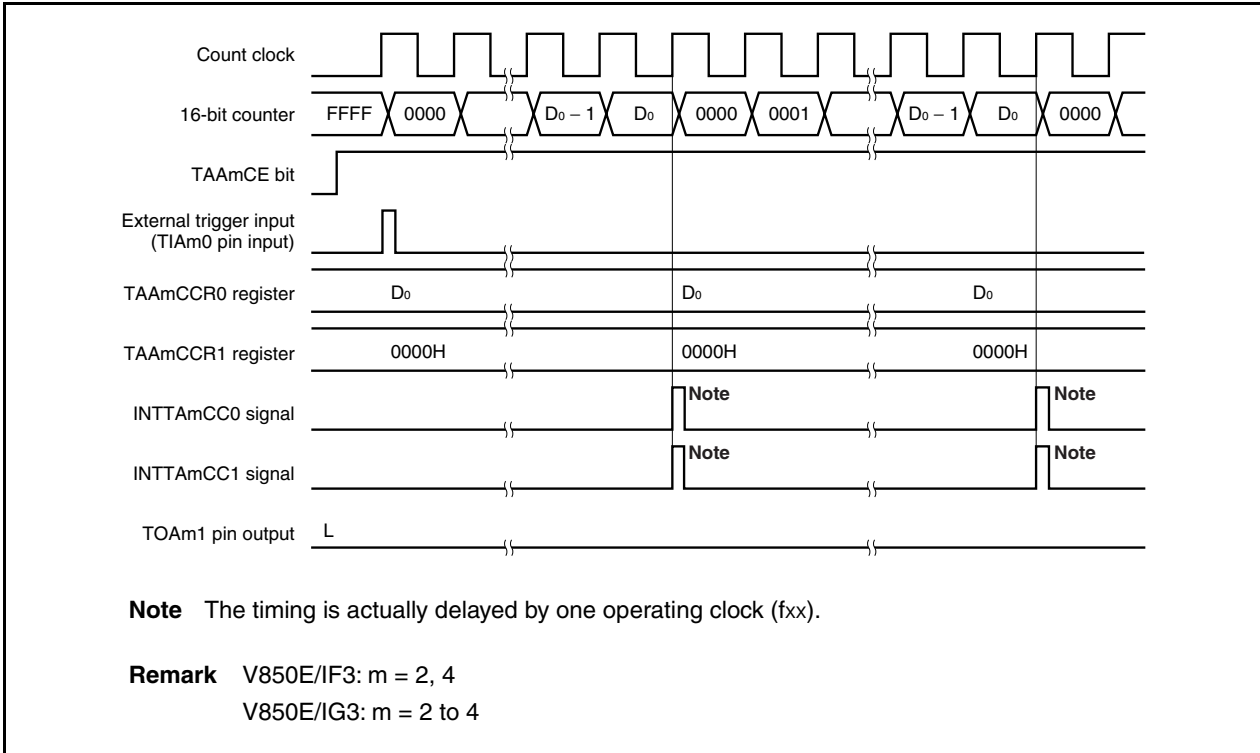
After data is written to the TAAmCCR1 register, the value written to the TAAmCCRa register is transferred to the CCRa buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TAAmCCR0 or TAAmCCR1 register again after writing the TAAmCCR1 register once, do so after the INTTAmCC0 signal is generated. Otherwise, the value of the CCRa buffer register may become undefined because the timing of transferring data from the TAAmCCRa register to the CCRa buffer register conflicts with writing the TAAmCCRa register.

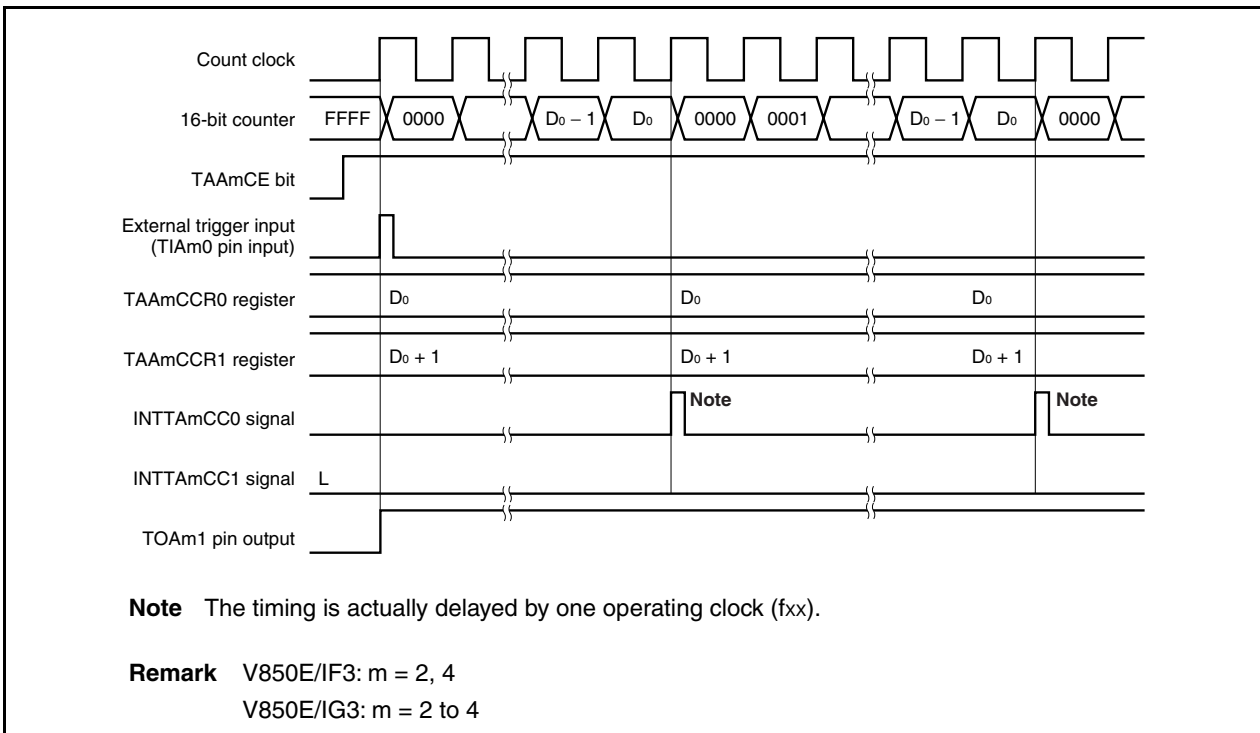
**Remark** V850E/IF3: m = 2, 4, a = 0, 1  
V850E/IG3: m = 2 to 4, a = 0, 1

**(b) 0%/100% output of PWM waveform**

To output a 0% waveform, set the TAAmCCR1 register to 0000H. The 16-bit counter is cleared to 0000H and the INTTAmCC0 and INTTAmCC1 signals are generated at the next timing after a match between the count value of the 16-bit counter and the value of the CCR0 buffer register.

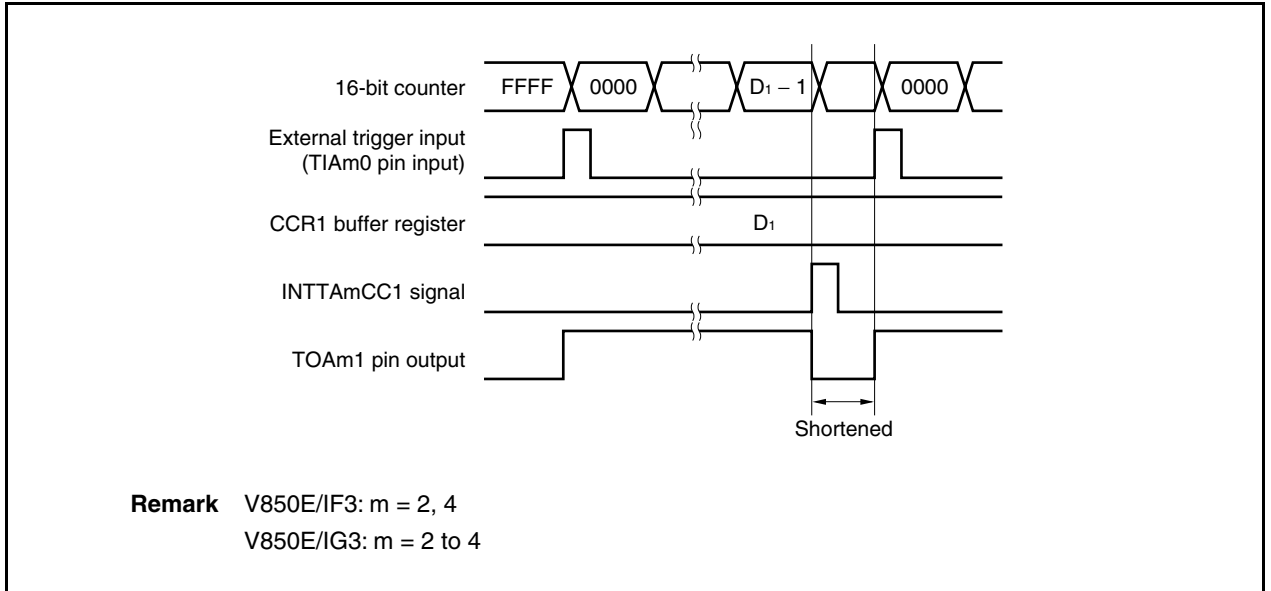


To output a 100% waveform, set a value of (set value of TAAmCCR0 register + 1) to the TAAmCCR1 register. If the set value of the TAAmCCR0 register is FFFFH, 100% output cannot be produced.

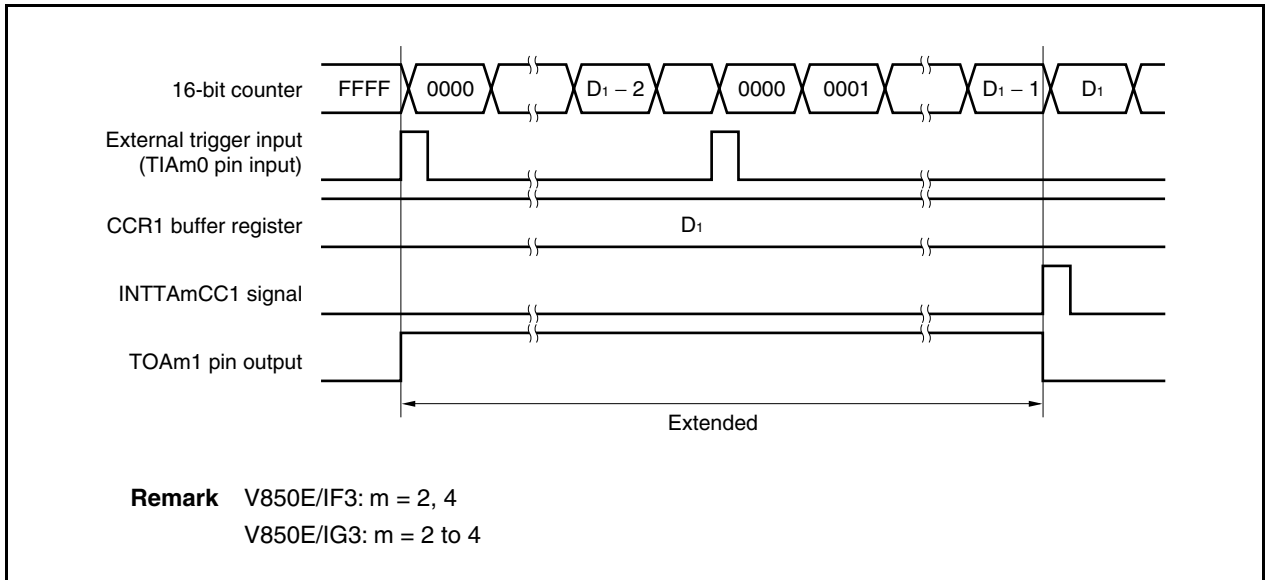


**(c) Conflict between trigger detection and match with CCR1 buffer register**

If the trigger is detected immediately after the INTTAmCC1 signal is generated, the 16-bit counter is immediately cleared to 0000H, the output signal of the TOAm1 pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.

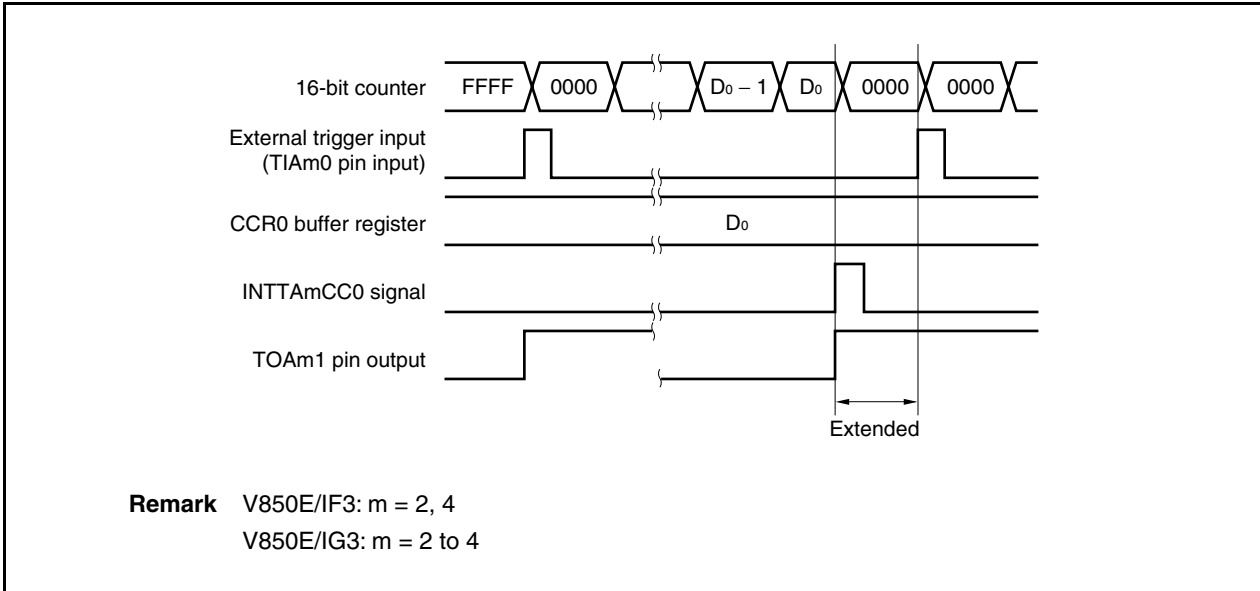


If the trigger is detected immediately before the INTTAmCC1 signal is generated, the INTTAmCC1 signal is not generated, and the 16-bit counter is cleared to 0000H and continues counting. The output signal of the TOAm1 pin remains active. Consequently, the active period of the PWM waveform is extended.

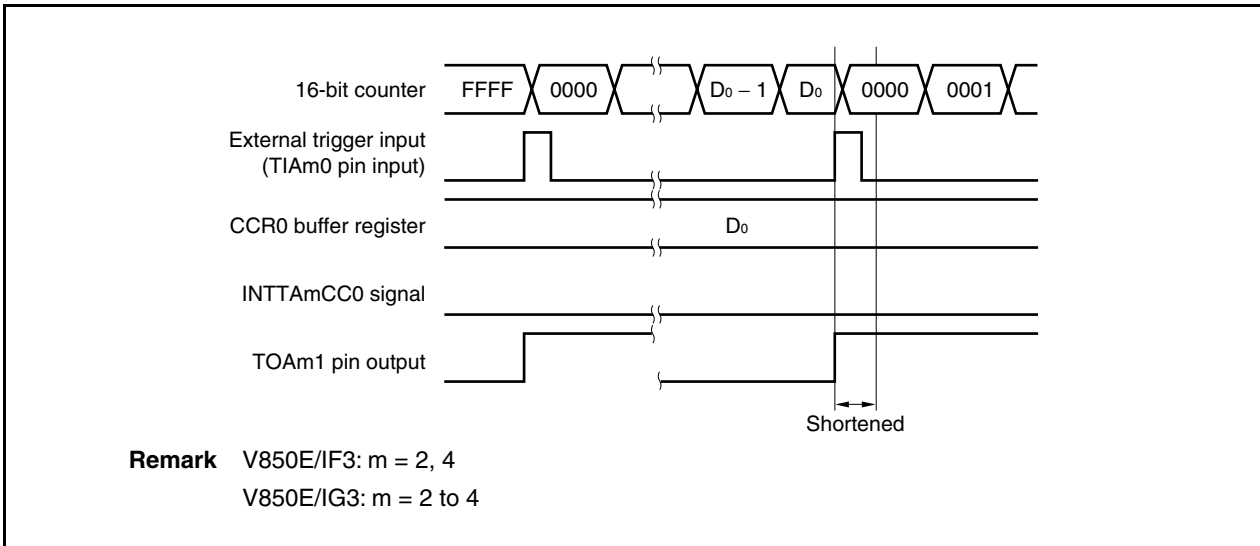


**(d) Conflict between trigger detection and match with CCR0 buffer register**

If the trigger is detected immediately after the INTTAmCC0 signal is generated, the 16-bit counter is cleared to 0000H and continues counting up. Therefore, the active period of the TOAm1 pin is extended by time from generation of the INTTAmCC0 signal to trigger detection.



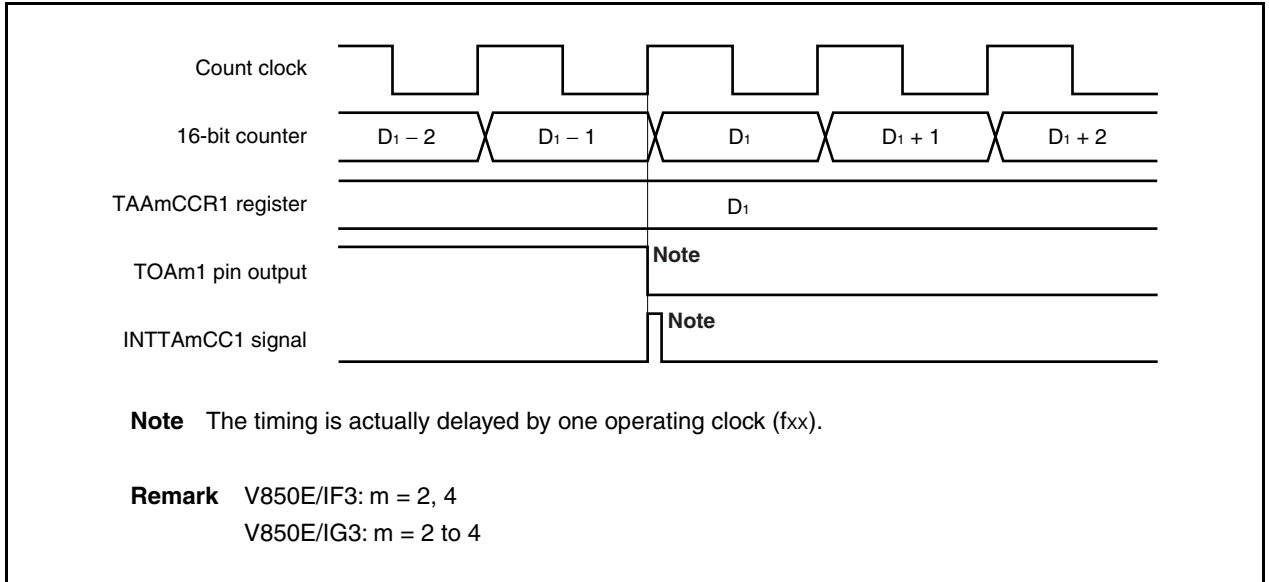
If the trigger is detected immediately before the INTTAmCC0 signal is generated, the INTTAmCC0 signal is not generated. The 16-bit counter is cleared to 0000H, the TOAm1 pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.





**(e) Generation timing of compare match interrupt request signal (INTTAmCC1)**

The timing of generation of the INTTAmCC1 signal in the external trigger pulse output mode differs from the timing of INTTAmCC1 signals in other mode; the INTTAmCC1 signal is generated when the count value of the 16-bit counter matches the value of the TAAmCCR1 register.



Usually, the INTTAmCC1 signal is generated in synchronization with the next count-up, after the count value of the 16-bit counter matches the value of the TAAmCCR1 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 TOAm1 pin.

### 6.6.4 One-shot pulse output mode (TAAmMD2 to TAAmMD0 bits = 011)

This mode is valid only in TAA2, TAA3 (V850E/IG3 only), and TAA4.

In the one-shot pulse output mode, 16-bit timer/event counter AA waits for a trigger when the TAAmCTL0.TAAmCE bit is set to 1. When the valid edge of an external trigger input is detected, 16-bit timer/event counter AA starts counting, and outputs a one-shot pulse from the TOAm1 pin.

Instead of the external trigger input (TIAm0), a software trigger can also be generated to output the pulse. When the software trigger is used, the TOAm0 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 6-28. Configuration in One-Shot Pulse Output Mode

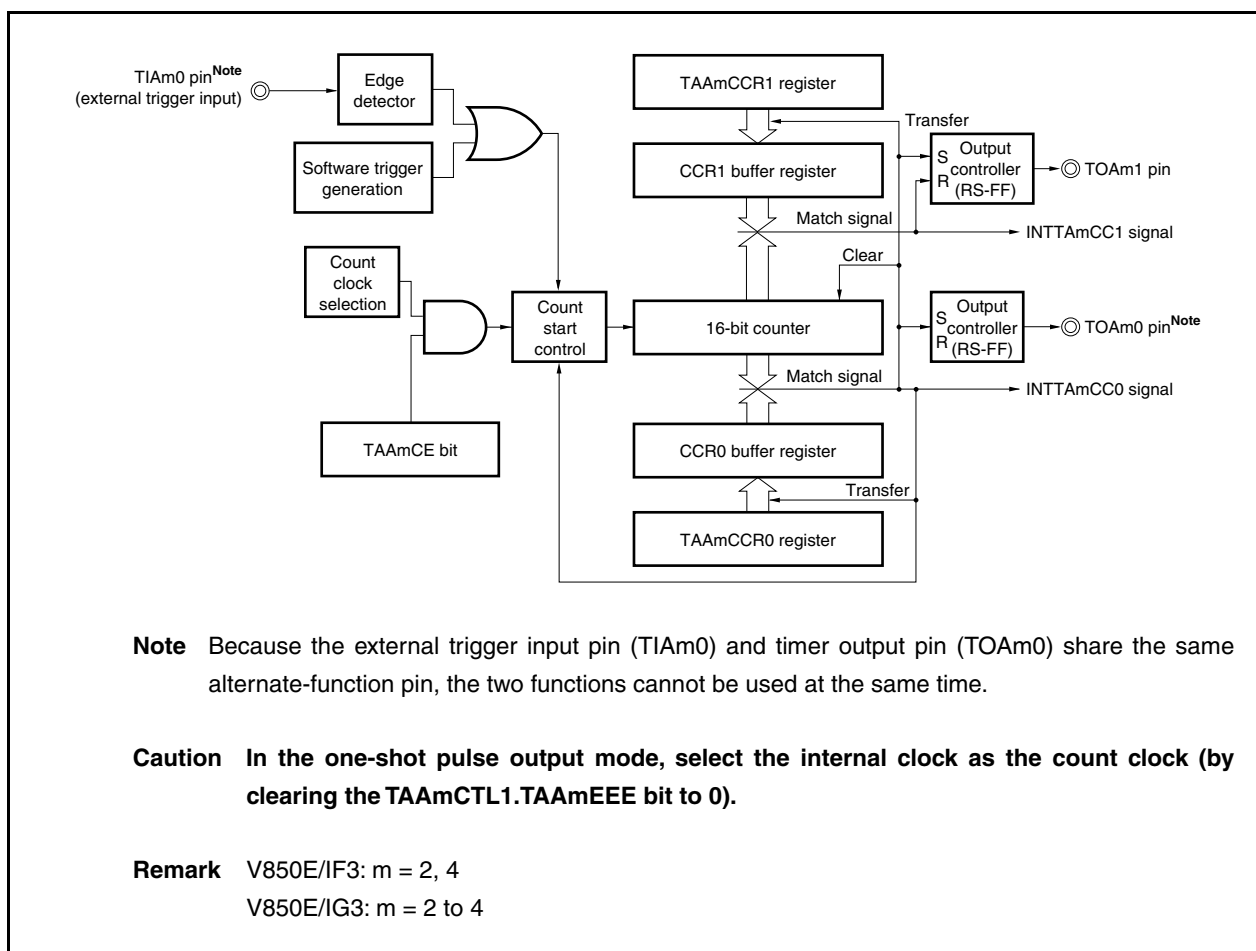
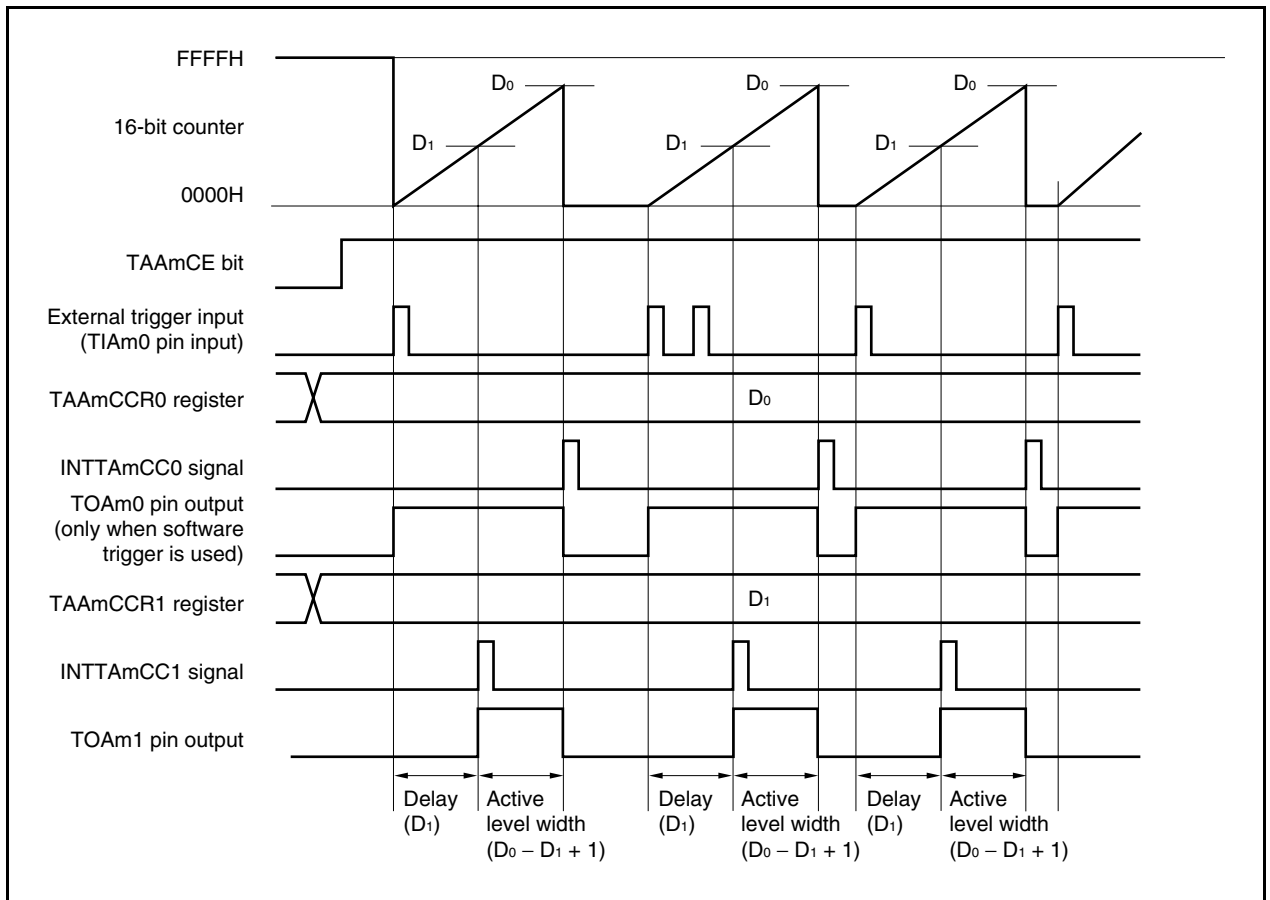


Figure 6-29. Basic Timing in One-Shot Pulse Output Mode



When the TAAmCE bit is set to 1, 16-bit timer/event counter AA 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 TOAm1 pin. After the one-shot pulse is output, the 16-bit counter is cleared to 0000H, stops counting, and waits for a trigger. When the trigger is generated again, the 16-bit counter starts counting from 0000H. 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.

$$\text{Output delay period} = (\text{Set value of TAAmCCR1 register}) \times \text{Count clock cycle}$$

$$\text{Active level width} = (\text{Set value of TAAmCCR0 register} - \text{Set value of TAAmCCR1 register} + 1) \times \text{Count clock cycle}$$

The compare match interrupt request signal (INTTAmCC0) 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 (INTTAmCC1) 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 (TIAm0 pin) or setting the software trigger (TAAmCTL1.TAAmEST bit) to 1 is used as the trigger.

**Remark** V850E/IF3: m = 2, 4  
V850E/IG3: m = 2 to 4

Figure 6-30. Setting of Registers in One-Shot Pulse Output Mode (1/2)

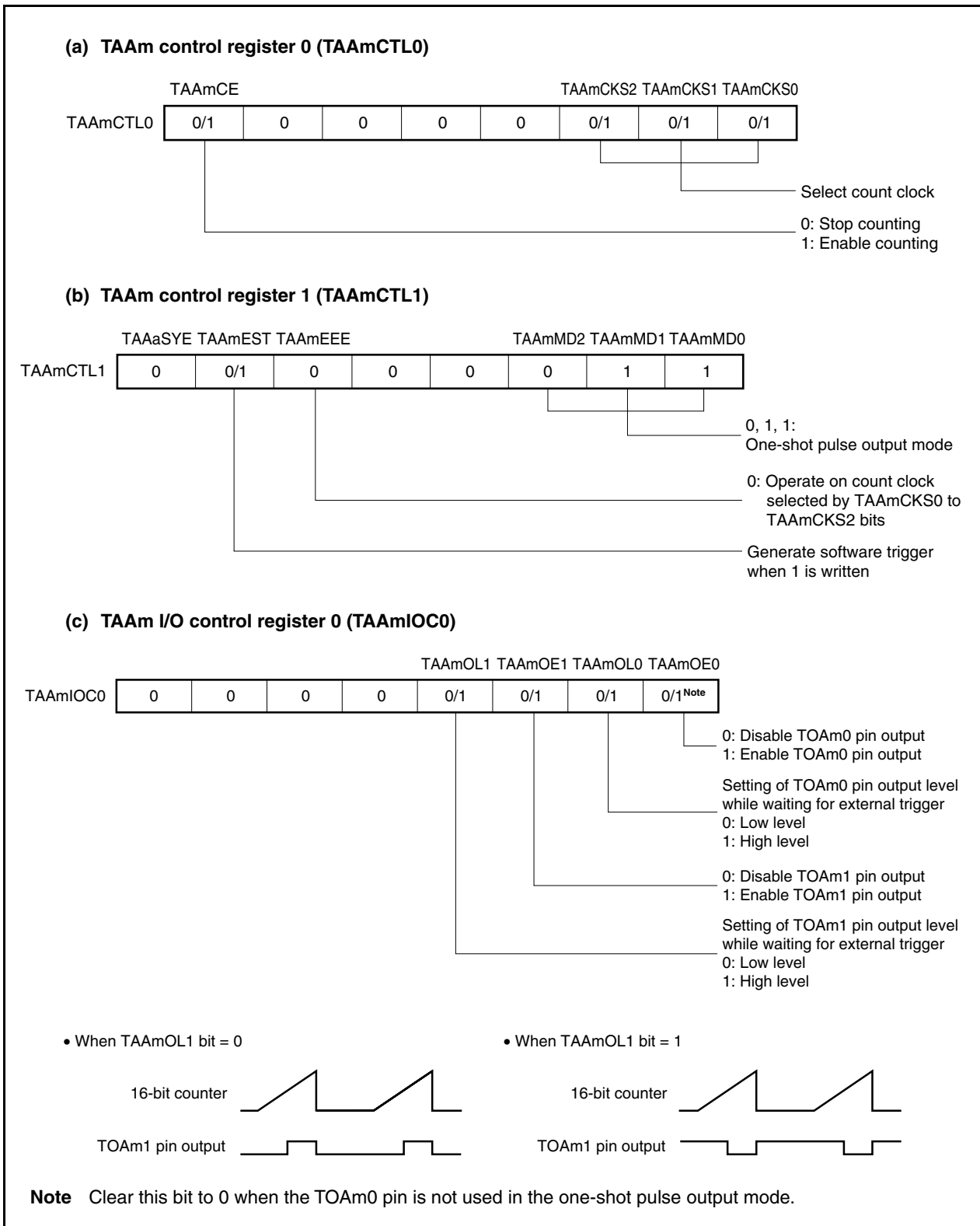
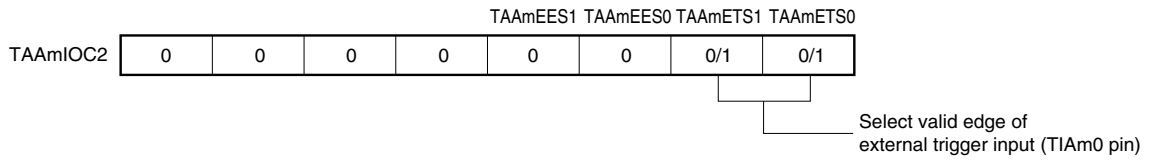


Figure 6-30. Setting of Registers in One-Shot Pulse Output Mode (2/2)

**(d) TAAm I/O control register 2 (TAAmIOC2)**



**(e) TAAm counter read buffer register (TAAmCNT)**

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

**(f) TAAm capture/compare registers 0 and 1 (TAAmCCR0 and TAAmCCR1)**

If  $D_0$  is set to the TAAmCCR0 register and  $D_1$  to the TAAmCCR1 register, the active level width and output delay period of the one-shot pulse are as follows.

Active level width =  $(D_0 - D_1 + 1) \times \text{Count clock cycle}$

Output delay period =  $D_1 \times \text{Count clock cycle}$

**Caution** One-shot pulses are not output even in the one-shot pulse output mode, if the value set in the TAAmCCR1 register is greater than that set in the TAAmCCR0 register.

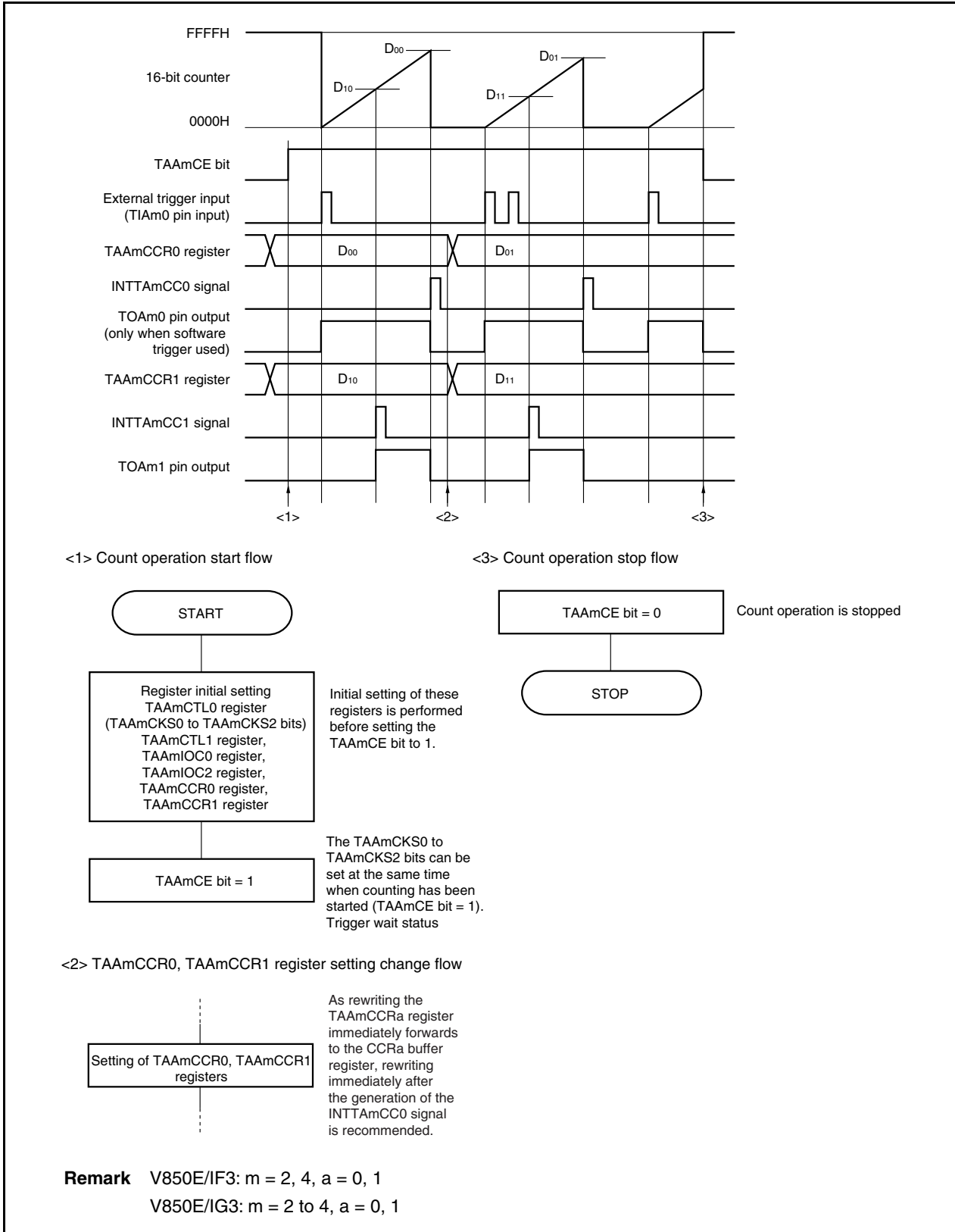
**Remarks** 1. TAAm I/O control register 1 (TAAmIOC1) and TAAm option register 0 (TAAmOPT0) are not used in the one-shot pulse output mode.

2. V850E/IF3:  $m = 2, 4, a = 0, 1$

V850E/IG3:  $m = 2 \text{ to } 4, a = 0, 1$

(1) Operation flow in one-shot pulse output mode

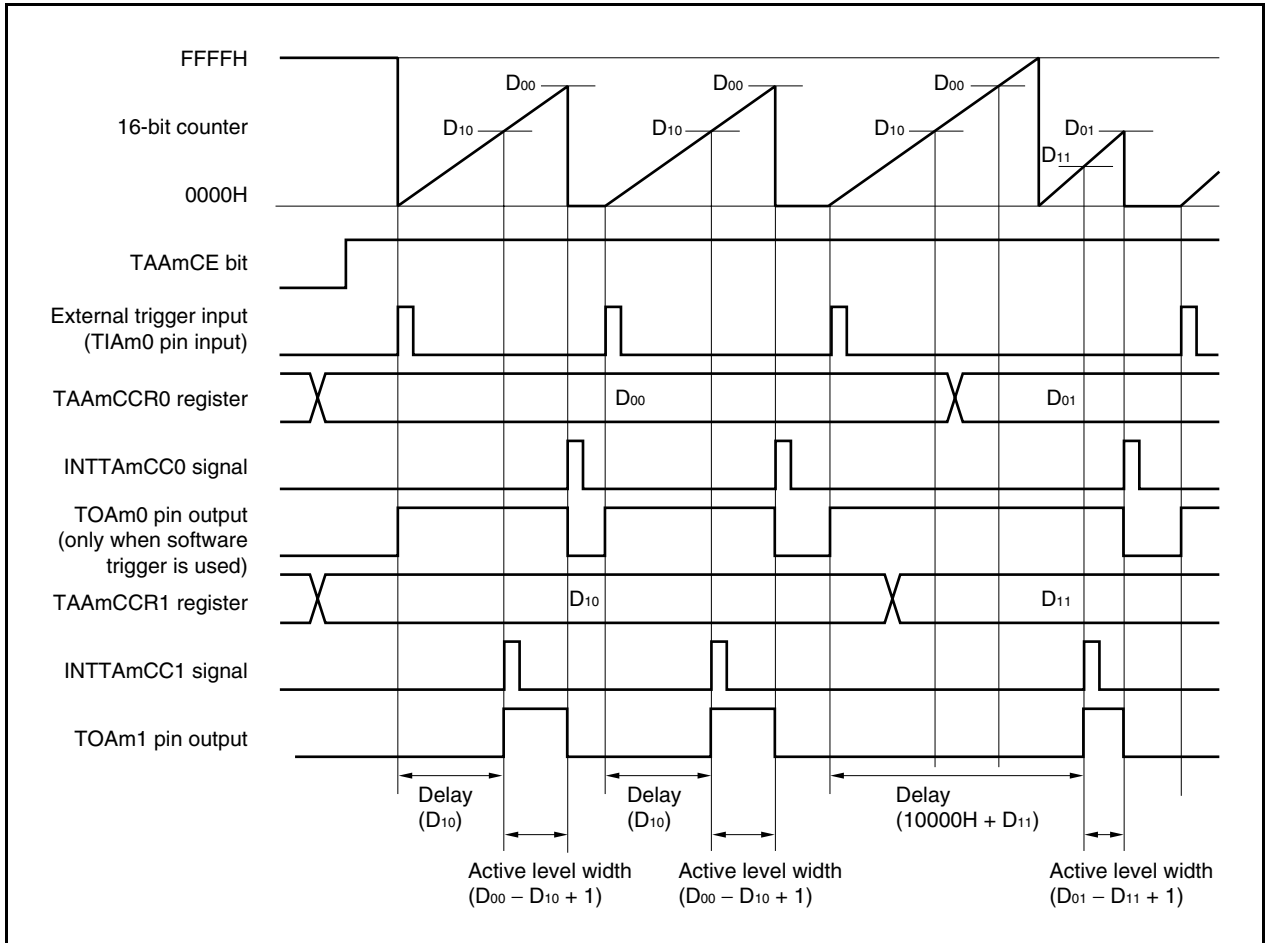
Figure 6-31. Software Processing Flow in One-Shot Pulse Output Mode



(2) Operation timing in one-shot pulse output mode

(a) Note on rewriting TAAmCCRa register

If the value of the TAAmCCRa register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When an overflow may occur, stop counting and then change the set value.

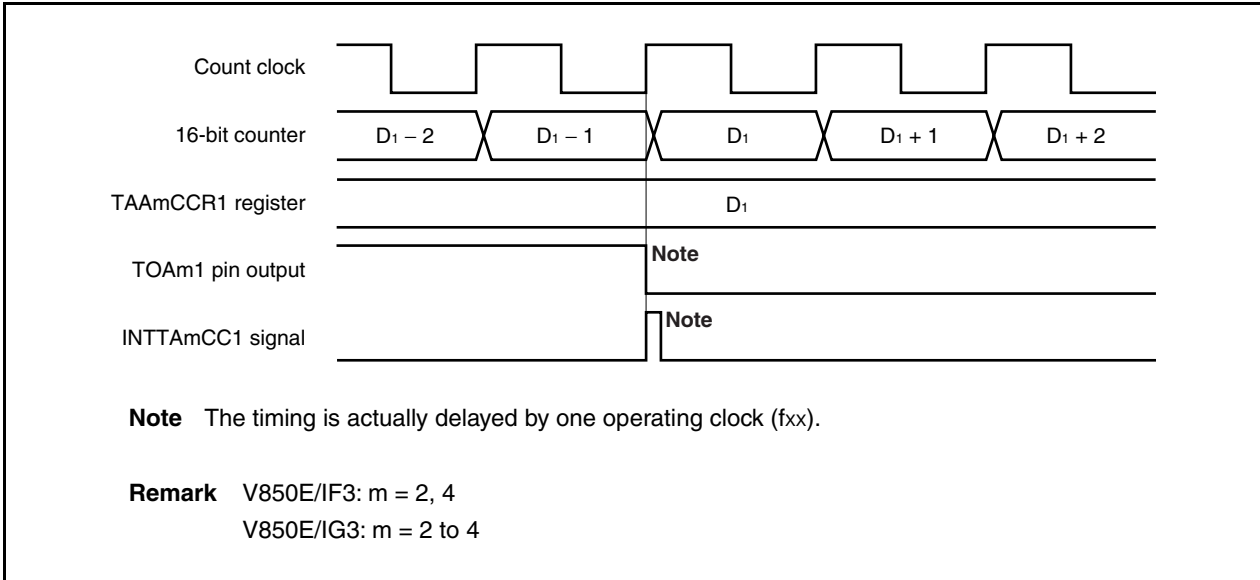


When the TAAmCCR0 register is rewritten from D<sub>00</sub> to D<sub>01</sub> and the TAAmCCR1 register from D<sub>10</sub> to D<sub>11</sub> where D<sub>00</sub> > D<sub>01</sub> and D<sub>10</sub> > D<sub>11</sub>, if the TAAmCCR1 register is rewritten when the count value of the 16-bit counter is greater than D<sub>11</sub> and less than D<sub>10</sub> and if the TAAmCCR0 register is rewritten when the count value is greater than D<sub>01</sub> and less than D<sub>00</sub>, 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<sub>11</sub>, the counter generates the INTTAmCC1 signal and asserts the TOAm1 pin output. When the count value matches D<sub>01</sub>, the counter generates the INTTAmCC0 signal, deasserts the TOAm1 pin output, 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** V850E/IF3: m = 2, 4, a = 0, 1  
 V850E/IG3: m = 2 to 4, a = 0, 1

**(b) Generation timing of compare match interrupt request signal (INTTAmCC1)**

The generation timing of the INTTAmCC1 signal in the one-shot pulse output mode is different from INTTAmCC1 signals; the INTTAmCC1 signal is generated when the count value of the 16-bit counter matches the value of the TAAmCCR1 register.



Usually, the INTTAmCC1 signal is generated when the 16-bit counter counts up next time after its count value matches the value of the TAAmCCR1 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 TOAm1 pin.



**6.6.5 PWM output mode (TAAmMD2 to TAAmMD0 bits = 100)**

This mode is valid only in TAA2, TAA3 (V850E/IG3 only), and TAA4.

In the PWM output mode, a PWM waveform is output from the TOAm1 pin when the TAAmCTL0.TAAmCE bit is set to 1.

In addition, a PWM waveform with a duty factor of 50% with the set value of the TAAmCCR0 register + 1 as half its cycle is output from the TOAm0 pin.

**Figure 6-32. Configuration in PWM Output Mode**

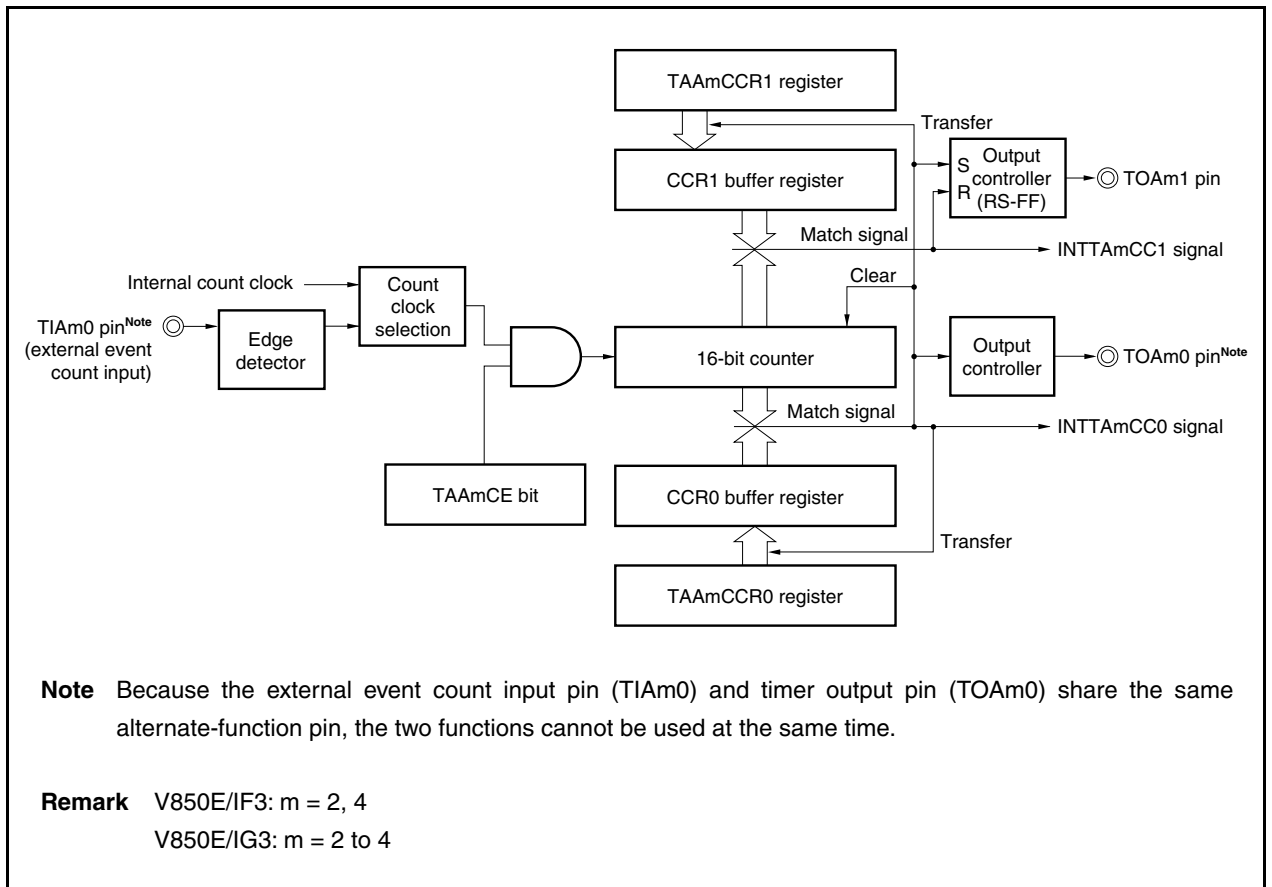
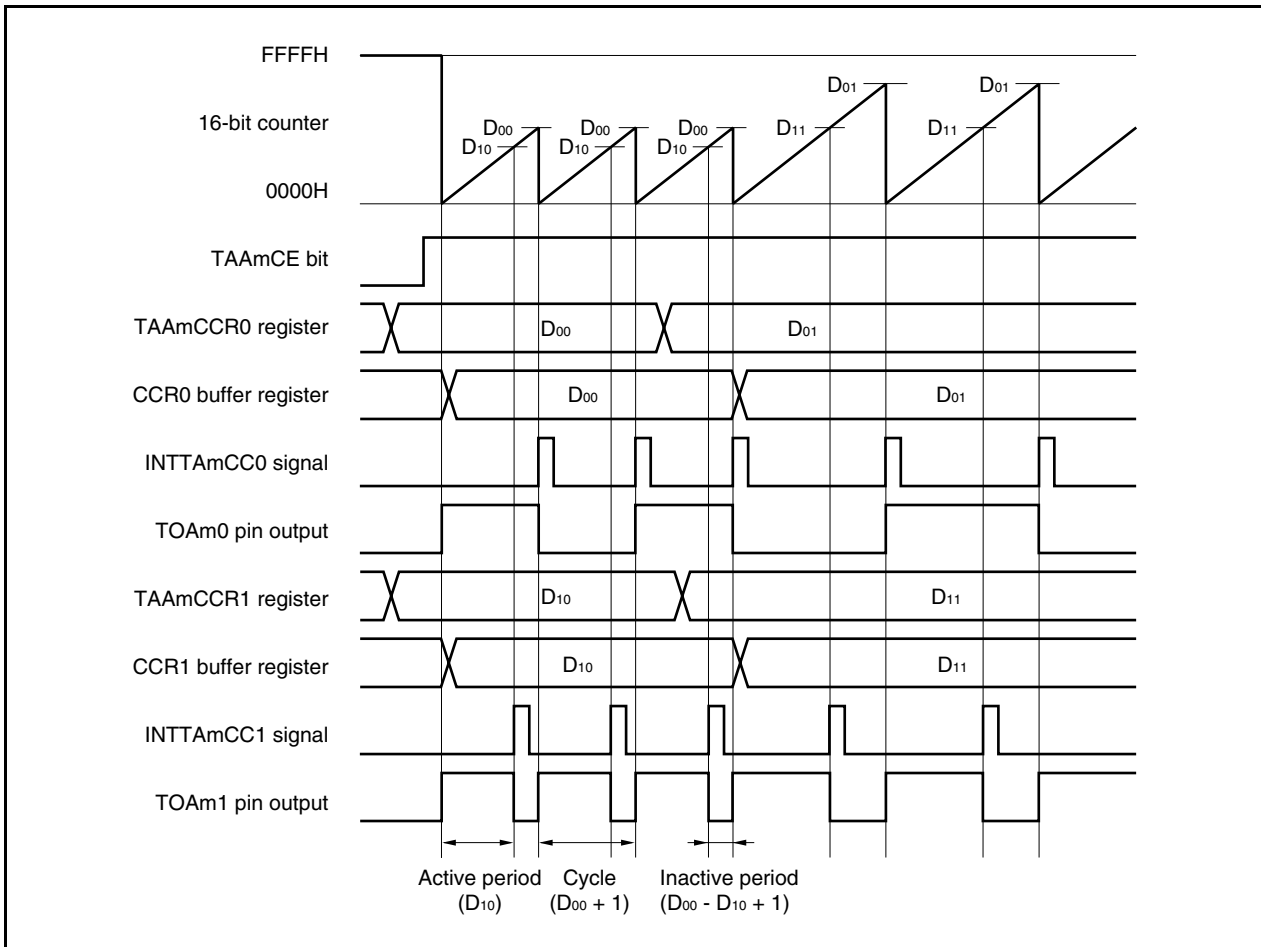


Figure 6-33. Basic Timing in PWM Output Mode



When the TAAmCE bit is set to 1, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs a PWM waveform from the TOAm1 pin.

The active level width, cycle, and duty factor of the PWM waveform can be calculated as follows.

$$\text{Active level width} = (\text{Set value of TAAmCCR1 register}) \times \text{Count clock cycle}$$

$$\text{Cycle} = (\text{Set value of TAAmCCR0 register} + 1) \times \text{Count clock cycle}$$

$$\text{Duty factor} = (\text{Set value of TAAmCCR1 register}) / (\text{Set value of TAAmCCR0 register} + 1)$$

The PWM waveform can be changed by rewriting the TAAmCCR<sub>a</sub> 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 INTTAmCC0 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 INTTAmCC1 is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

The value set to the TAAmCCR<sub>a</sub> register is transferred to the CCR<sub>a</sub> buffer register when the count value of the 16-bit counter matches the value of the CCR<sub>a</sub> buffer register and the 16-bit counter is cleared to 0000H.

**Remark** V850E/IF3: m = 2, 4, a = 0, 1  
 V850E/IG3: m = 2 to 4, a = 0, 1

Figure 6-34. Setting of Registers in PWM Output Mode (1/2)

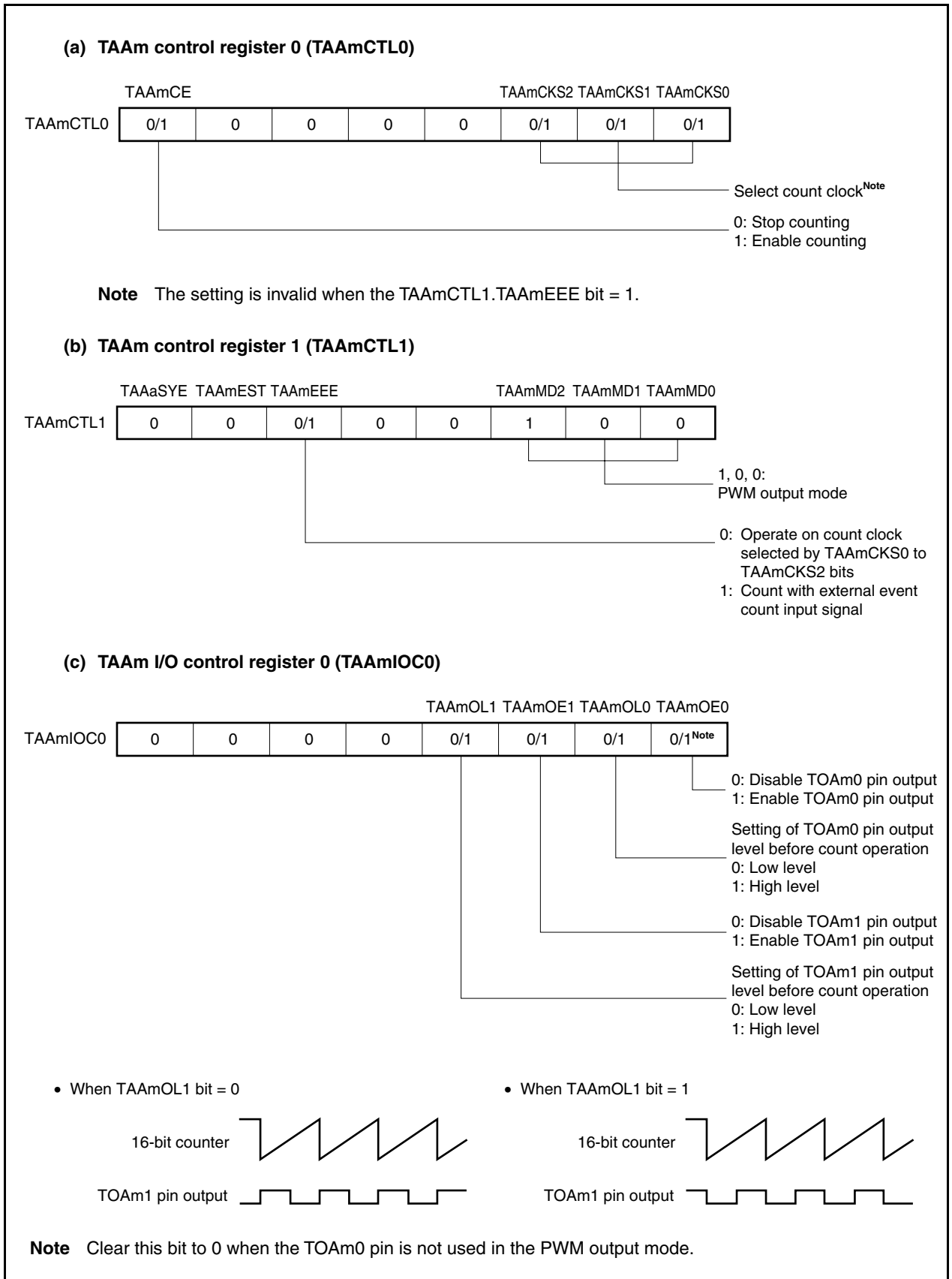
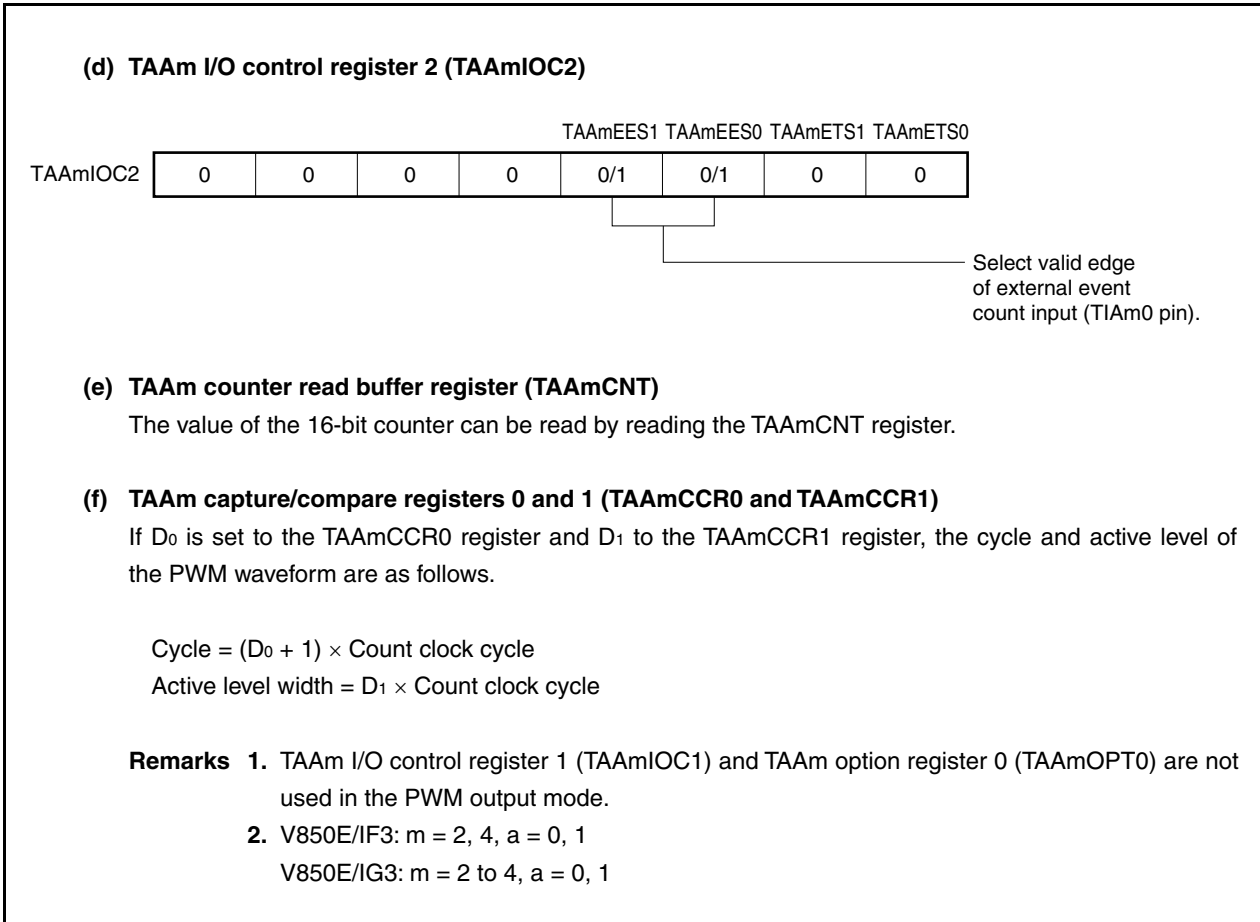


Figure 6-34. Register Setting in PWM Output Mode (2/2)



(1) Operation flow in PWM output mode

Figure 6-35. Software Processing Flow in PWM Output Mode (1/2)

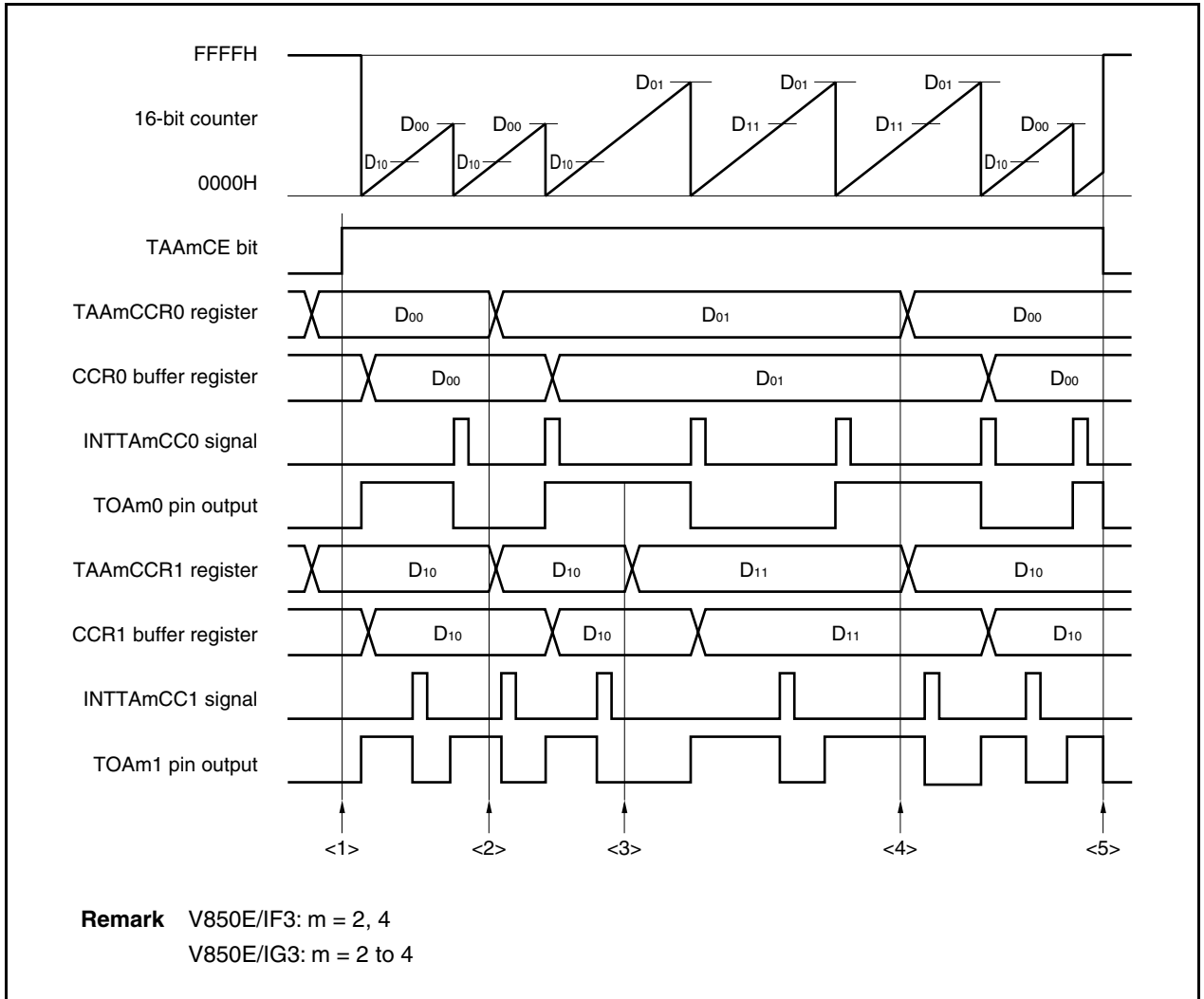
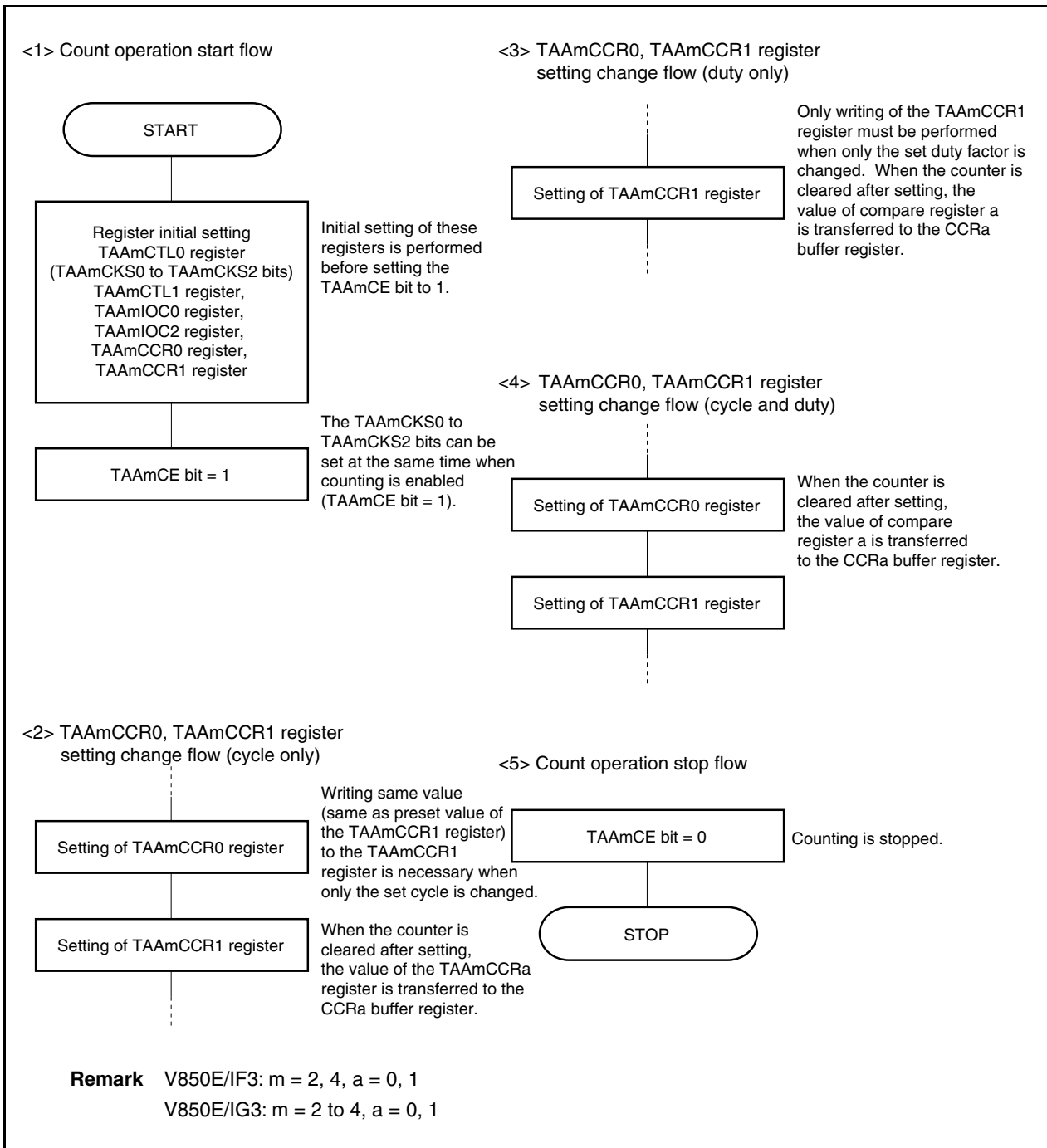


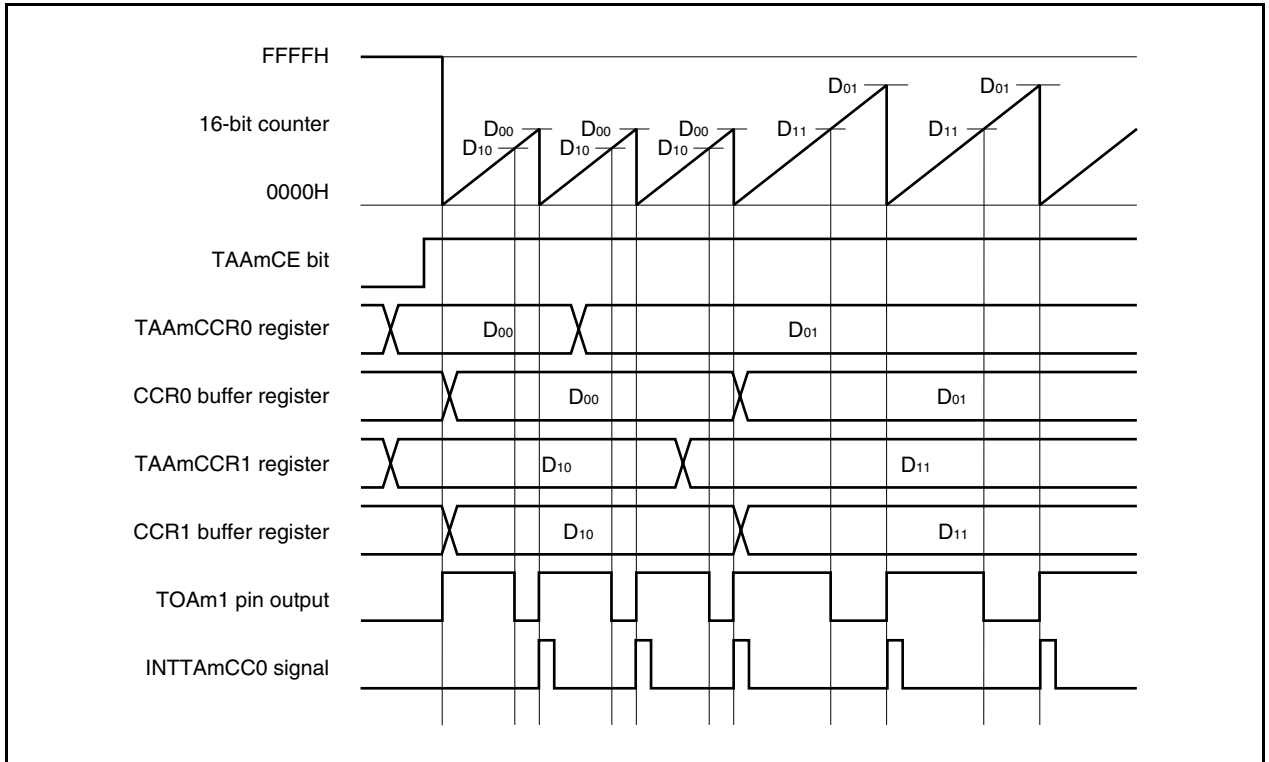
Figure 6-35. Software Processing Flow in PWM Output Mode (2/2)



(2) PWM output mode operation timing

(a) Changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TAAmCCR1 register last. Rewrite the TAAmCCRa register after writing the TAAmCCR1 register after the INTTAmCC0 signal is detected.



To transfer data from the TAAmCCRa register to the CCRa buffer register, the TAAmCCR1 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 TAAmCCR0 register and then set the active level width to the TAAmCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TAAmCCR0 register, and then write the same value (same as preset value of the TAAmCCR1 register) to the TAAmCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, only the TAAmCCR1 register has to be set.

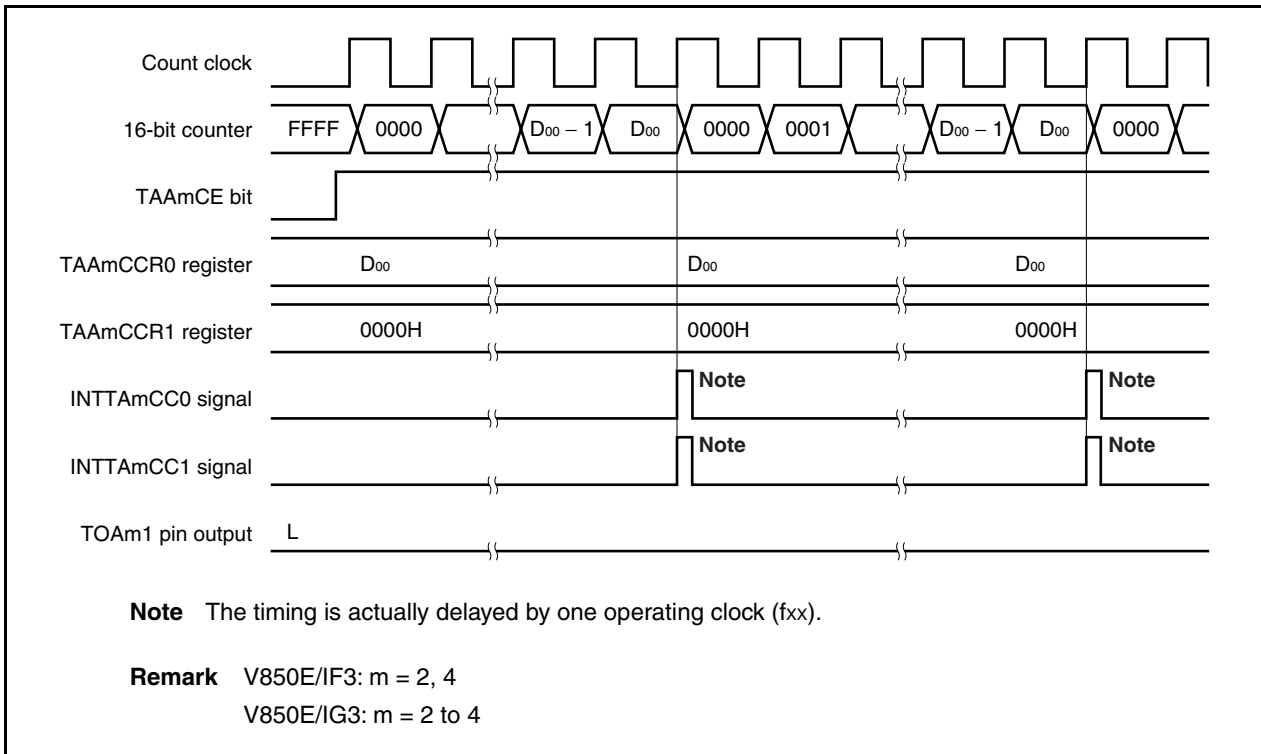
After data is written to the TAAmCCR1 register, the value written to the TAAmCCRa register is transferred to the CCRa buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TAAmCCR0 or TAAmCCR1 register again after writing the TAAmCCR1 register once, do so after the INTTAmCC0 signal is generated. Otherwise, the value of the CCRa buffer register may become undefined because the timing of transferring data from the TAAmCCRa register to the CCRa buffer register conflicts with writing the TAAmCCRa register.

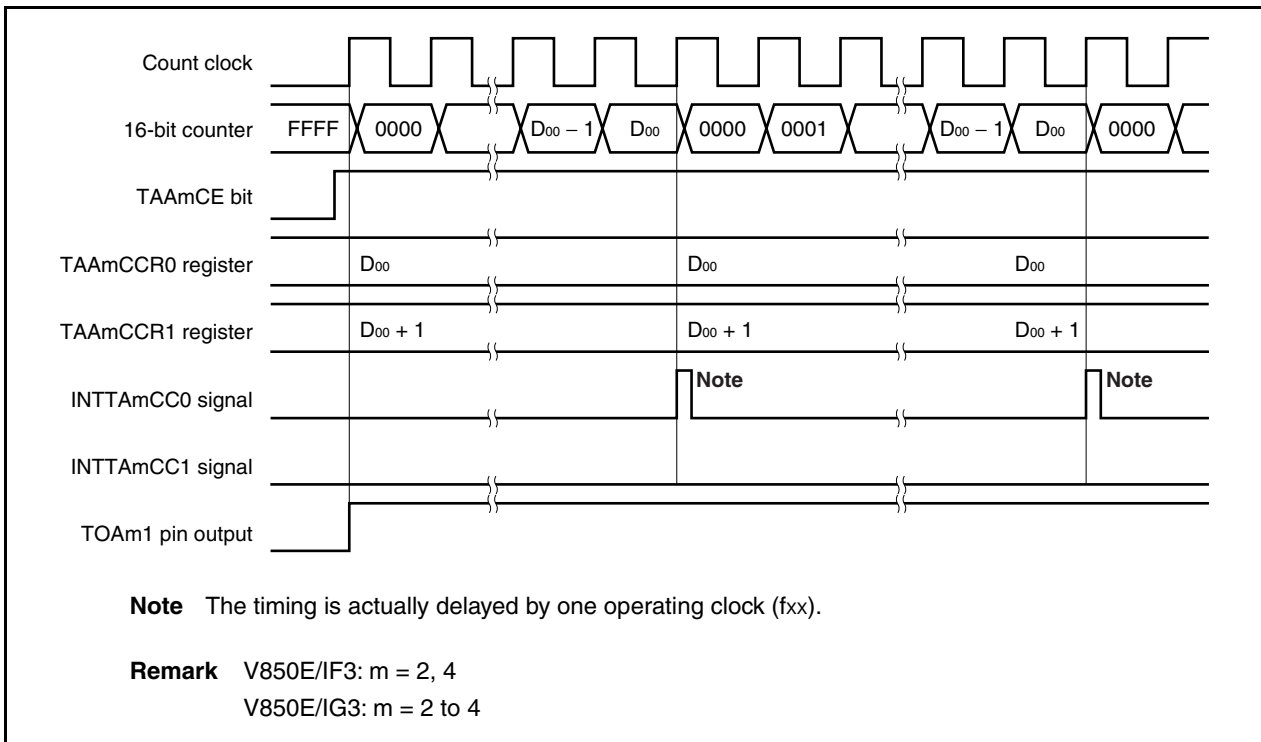
**Remark** V850E/IF3: m = 2, 4, a = 0, 1  
 V850E/IG3: m = 2 to 4, a = 0, 1

**(b) 0%/100% output of PWM waveform**

To output a 0% waveform, set the TAAmCCR1 register to 0000H. The 16-bit counter is cleared to 0000H and the INTTAmCC0 and INTTAmCC1 signals are generated at the next timing after a match between the count value of the 16-bit counter and the value of the CCR0 buffer register.



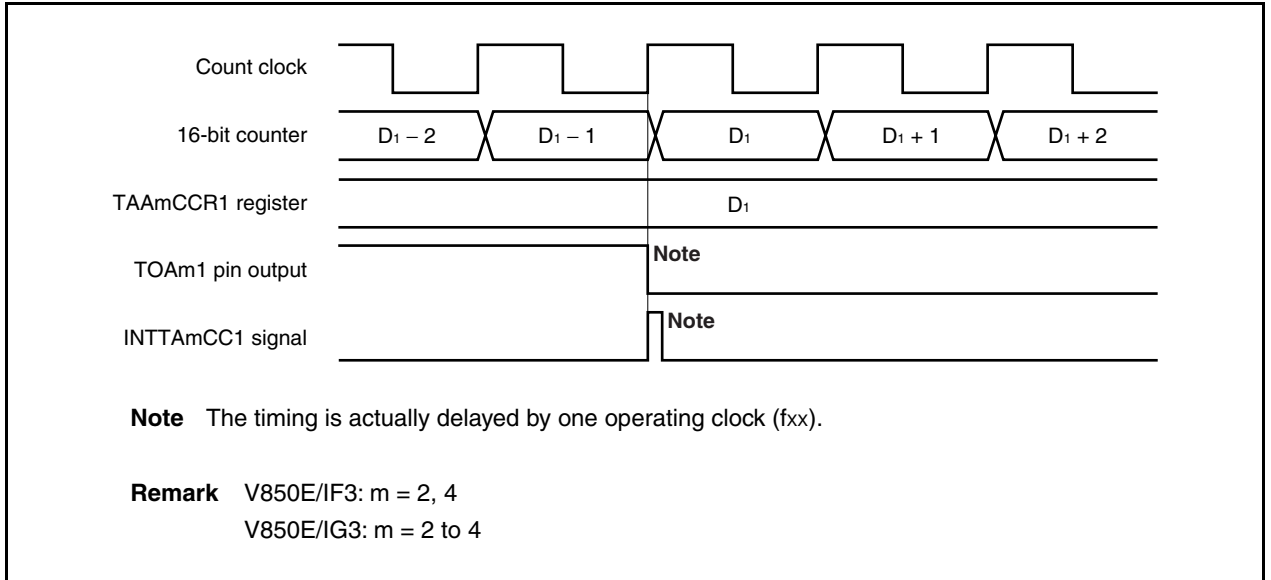
To output a 100% waveform, set a value of (set value of TAAmCCR0 register + 1) to the TAAmCCR1 register. If the set value of the TAAmCCR0 register is FFFFH, 100% output cannot be produced.





**(c) Generation timing of compare match interrupt request signal (INTTAmCC1)**

The timing of generation of the INTTAmCC1 signal in the PWM output mode differs from the timing of INTTAmCC1 signals; the INTTAmCC1 signal is generated when the count value of the 16-bit counter matches the value of the TAAmCCR1 register.



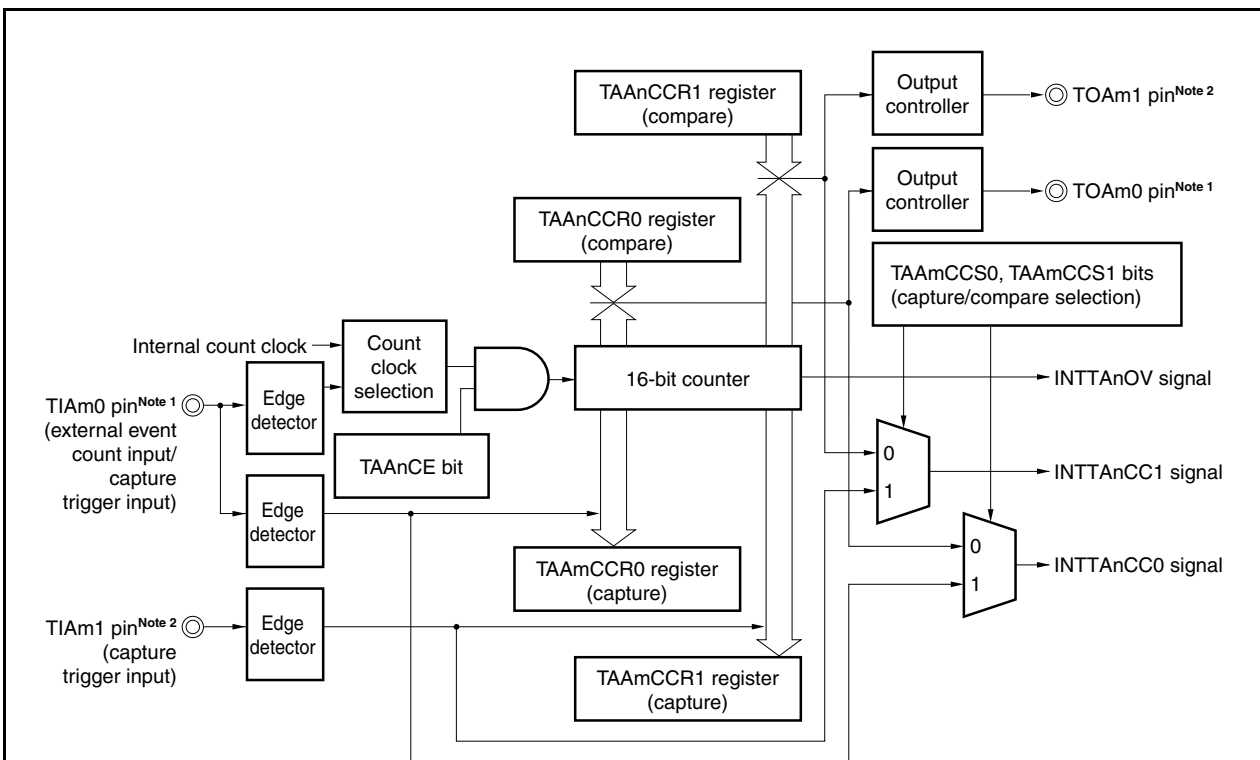
Usually, the INTTAmCC1 signal is generated in synchronization with the next counting up after the count value of the 16-bit counter matches the value of the TAAmCCR1 register. In the PWM output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the output signal of the TOAm1 pin.

**6.6.6 Free-running timer mode (TAA<sub>n</sub>MD2 to TAA<sub>n</sub>MD0 bits = 101)**

The compare function is valid for all of TAA0 to TAA4. The capture function is valid only for TAA2, TAA3 (V850E/IG3 only), and TAA4.

In the free-running timer mode, 16-bit timer/event counter AA starts counting when the TAA<sub>n</sub>CTL0.TAA<sub>n</sub>CE bit is set to 1. At this time, the TAA<sub>m</sub>CCR0 and TAA<sub>m</sub>CCR1 registers can be used as compare registers or capture registers, depending on the setting of the TAA<sub>m</sub>OPT0.TAA<sub>m</sub>CCS0 and TAA<sub>m</sub>OPT0.TAA<sub>m</sub>CCS1 bits.

**Figure 6-36. Configuration in Free-Running Timer Mode**



- Notes 1.** Because the external event count input pin (TIAm0), capture trigger input pin (TIAm0), and timer output pin (TOAm0) share the same alternate-function pin, the two or more functions cannot be used at the same time.
- 2.** Because the capture trigger input pin (TIAm1) and timer output pin (TOAm1) share the same alternate-function pin, the two functions cannot be used at the same time.

**Remark** V850E/IF3: n = 0 to 4, m = 2, 4  
 V850E/IG3: n = 0 to 4, m = 2 to 4

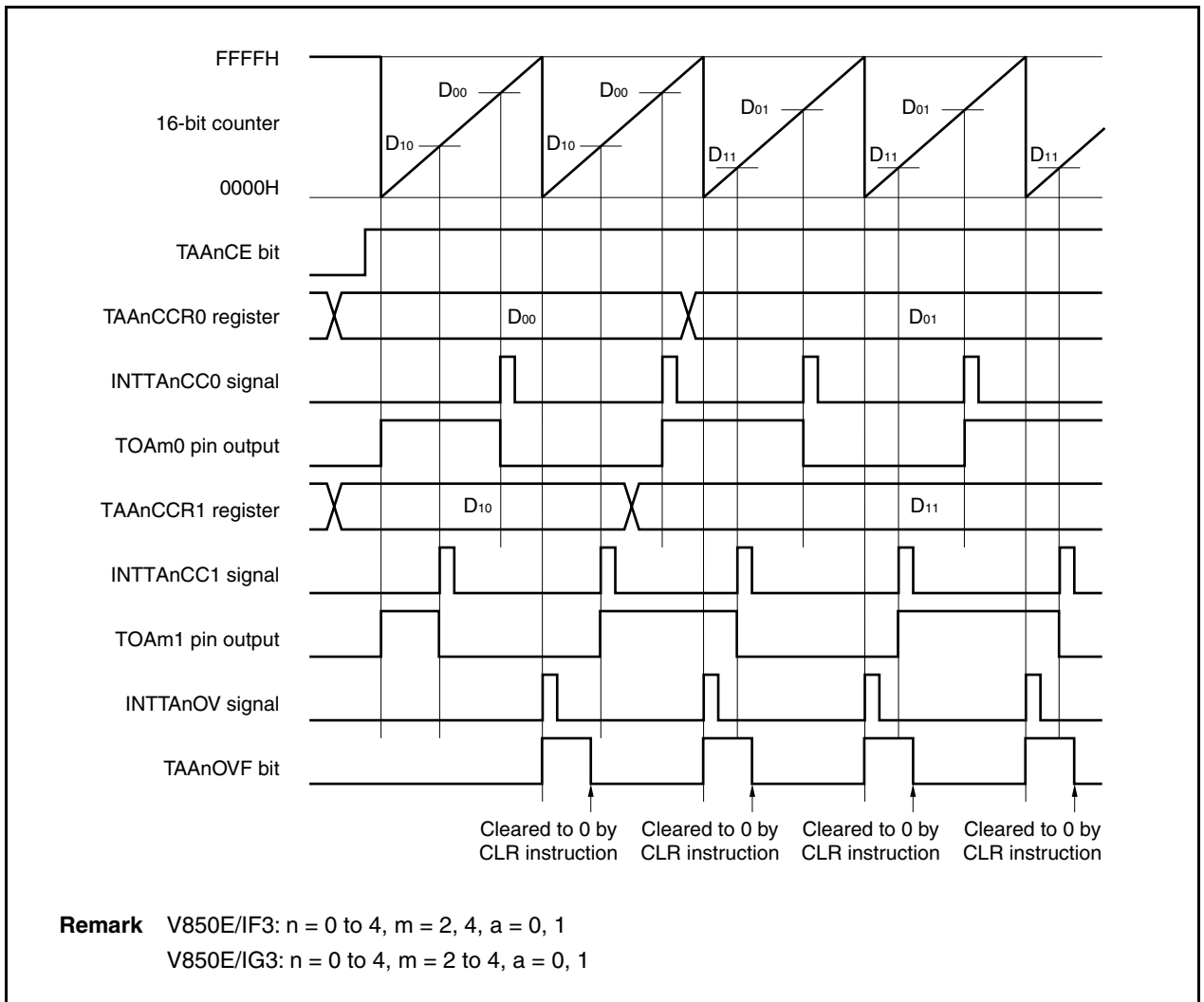
- Compare operation

When the TAA<sub>n</sub>CE bit is set to 1, 16-bit timer/event counter AA starts counting, and the output signal of the TOA<sub>m</sub> pin is inverted. When the count value of the 16-bit counter later matches the set value of the TAA<sub>n</sub>CCR<sub>a</sub> register, a compare match interrupt request signal (INTTA<sub>n</sub>CC<sub>a</sub>) is generated, and the output signal of the TOA<sub>m</sub> 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 (INTTA<sub>n</sub>OV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TAA<sub>n</sub>OPT0.TAA<sub>n</sub>OVF bit) is also set to 1. Confirm that the overflow flag is set to 1 and then clear it to 0 by executing the CLR instruction via software.

The TAA<sub>n</sub>CCR<sub>a</sub> register can be rewritten while the counter is operating. If it is rewritten, the new value is reflected at that time by anytime write, and compared with the count value.

**Figure 6-37. Basic Timing in Free-Running Timer Mode (Compare Function)**



- Capture operation

When the TAAmCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIAm<sub>a</sub> pin is detected, the count value of the 16-bit counter is stored in the TAAmCCR<sub>a</sub> register, and a capture interrupt request signal (INTTAmCC<sub>a</sub>) 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 (INTTAmOV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TAAmOPT0.TAAmOVF bit) is also set to 1. Confirm that the overflow flag is set to 1 and then clear it to 0 by executing the CLR instruction via software.

**Figure 6-38. Basic Timing in Free-Running Timer Mode (Capture Function)**

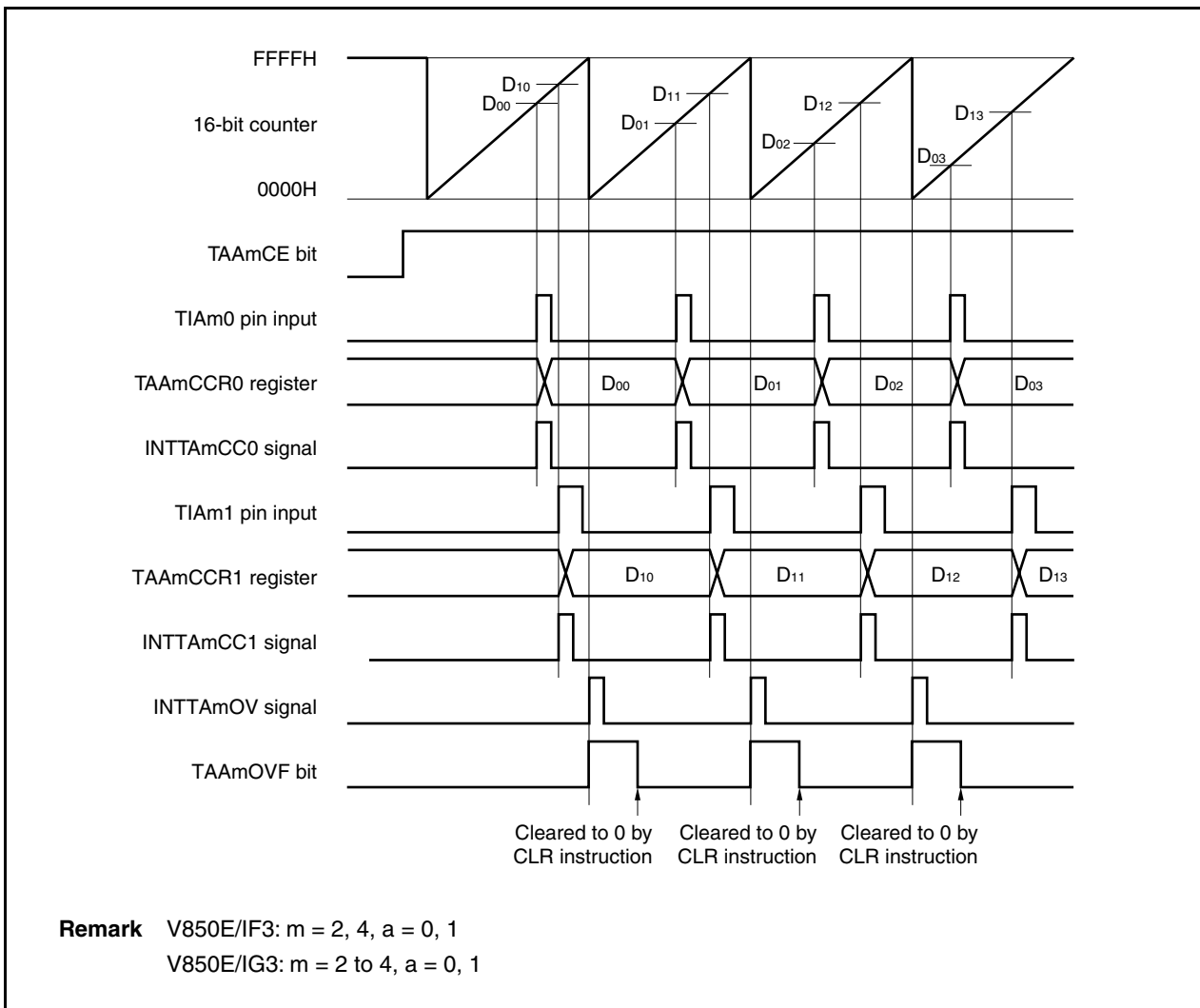


Figure 6-39. Register Setting in Free-Running Timer Mode (1/2)

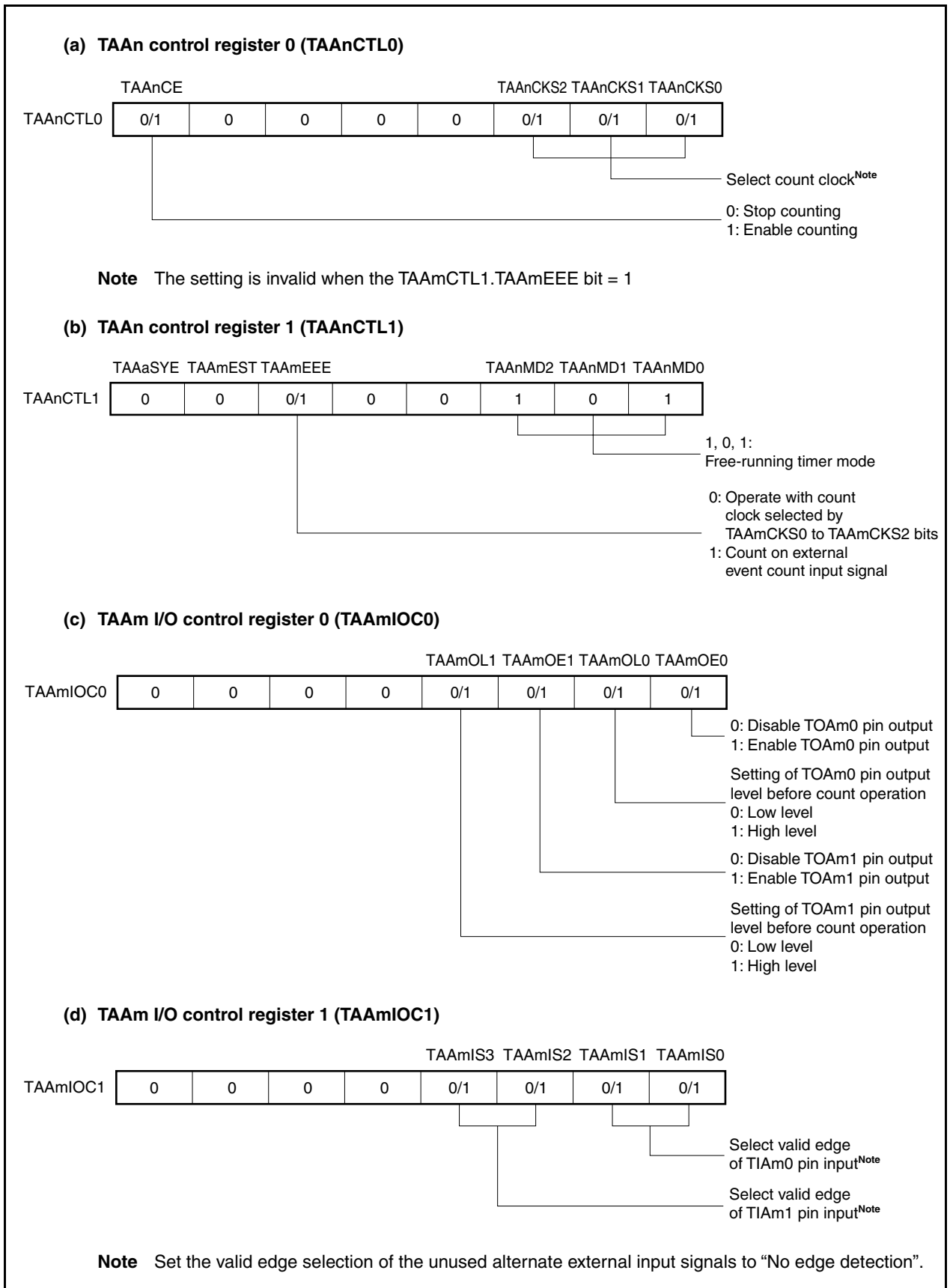
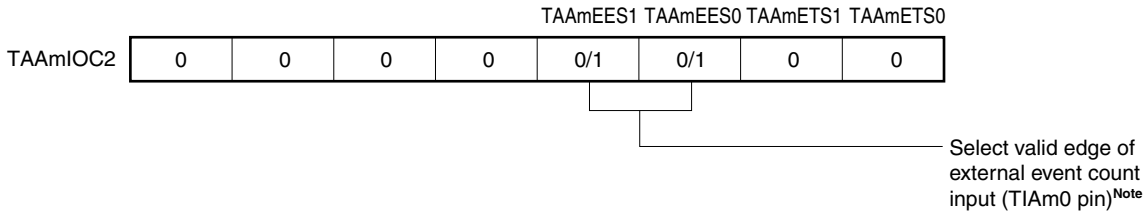


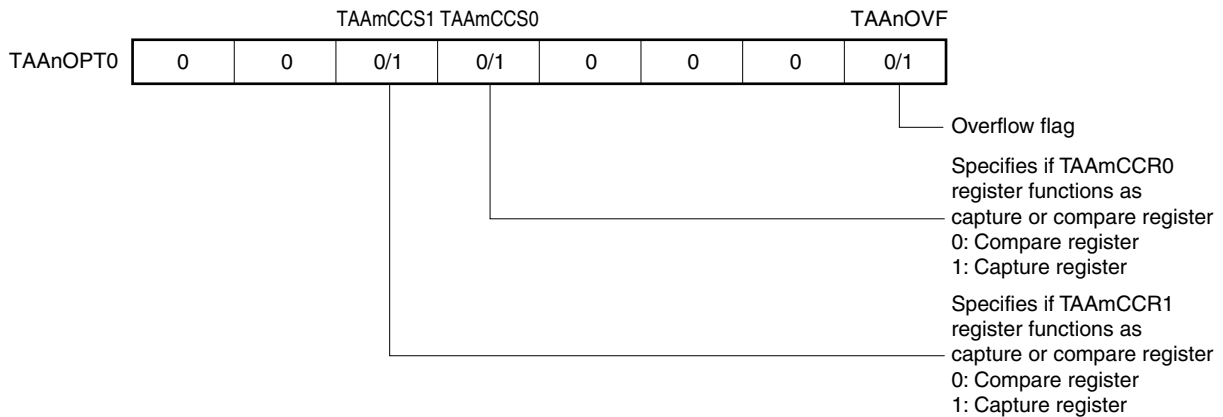
Figure 6-39. Register Setting in Free-Running Timer Mode (2/2)

(e) TAAm I/O control register 2 (TAAmIOC2)



**Note** Set the valid edge selection of the unused alternate external input signals to “No edge detection”.

(f) TAAm option register 0 (TAAmOPT0)



(g) TAAm counter read buffer register (TAAmCNT)

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

(h) TAAm capture/compare registers 0 and 1 (TAAmCCR0 and TAAmCCR1)

These registers function as capture registers or compare registers depending on the setting of the TAAmOPT0.TAAmCCSa bit.

When the registers function as capture registers, they store the count value of the 16-bit counter when the valid edge input to the TIAMA pin is detected.

When the registers function as compare registers and when  $D_a$  is set to the TAAmCCRa register, the INTTANCCa signal is generated when the counter reaches  $(D_a + 1)$ , and the output signals of the TOAm0 and TOAm1 pins are inverted.

**Remark** V850E/IF3:  $n = 0$  to  $4$ ,  $m = 2, 4$ ,  $a = 0, 1$   
V850E/IG3:  $n = 0$  to  $4$ ,  $m = 2$  to  $4$ ,  $a = 0, 1$

(1) Operation flow in free-running timer mode

(a) When using capture/compare register as compare register

Figure 6-40. Software Processing Flow in Free-Running Timer Mode (Compare Function) (1/2)

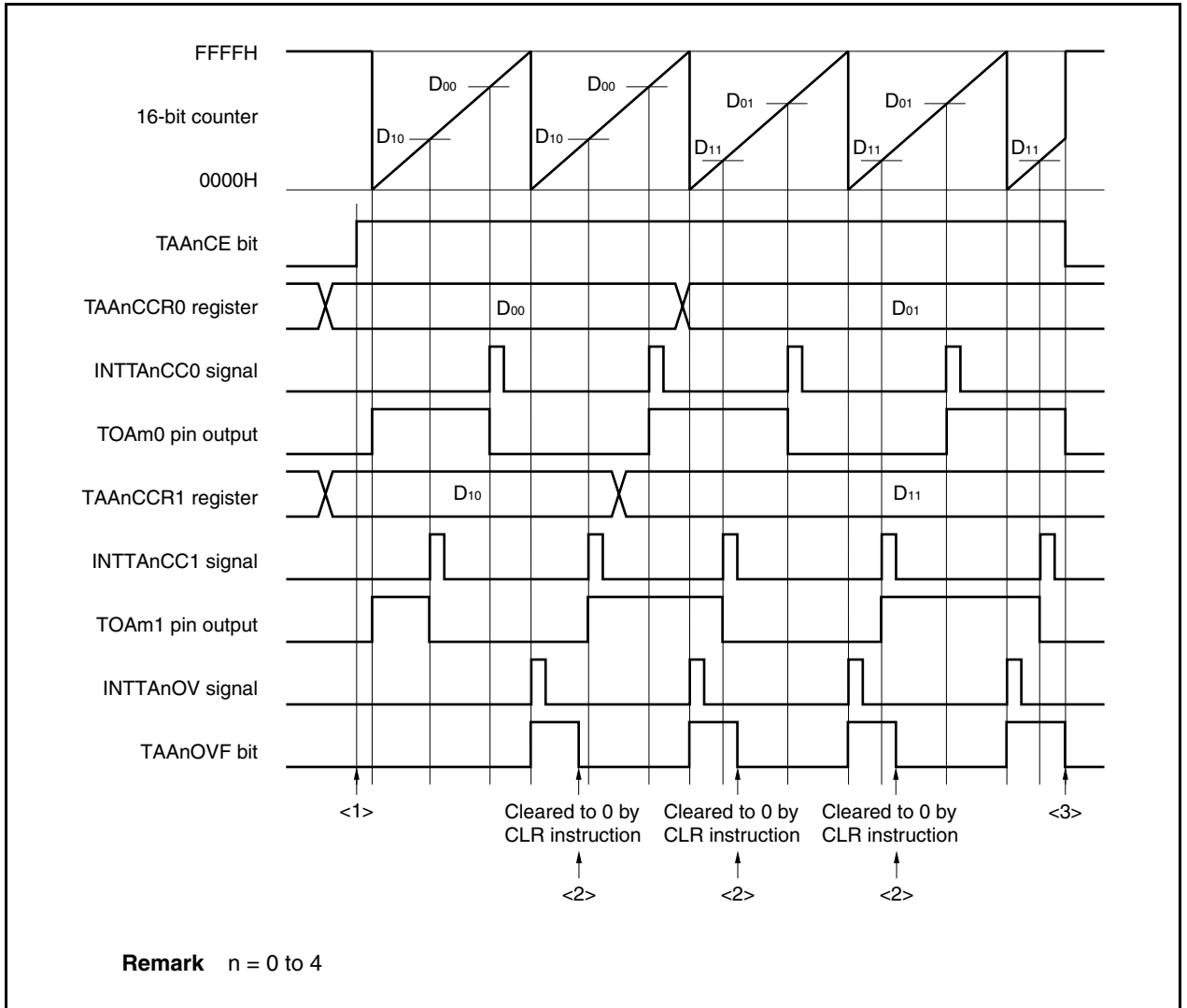
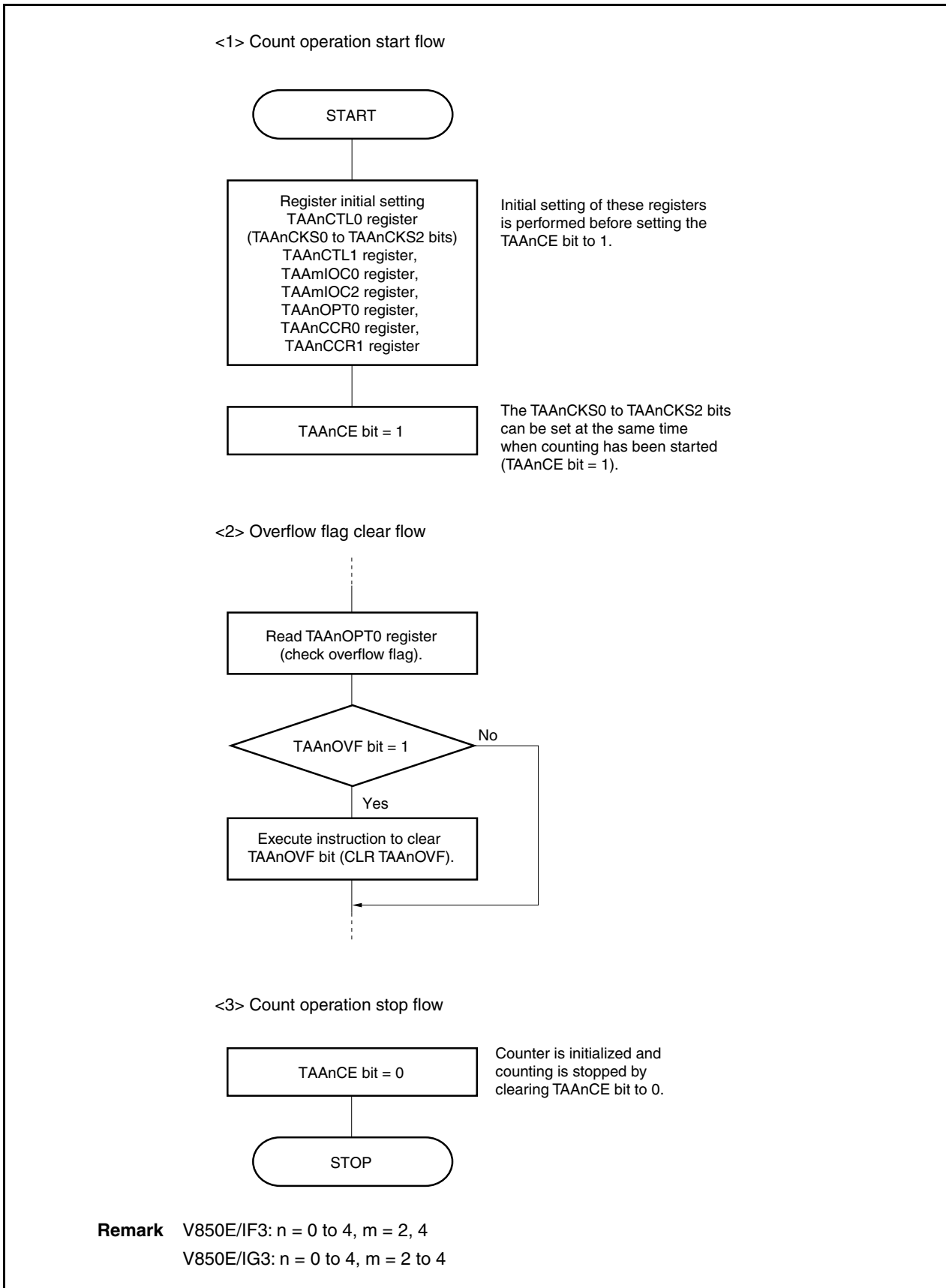


Figure 6-40. Software Processing Flow in Free-Running Timer Mode (Compare Function) (2/2)





(b) When using capture/compare register as capture register

Figure 6-41. Software Processing Flow in Free-Running Timer Mode (Capture Function) (1/2)

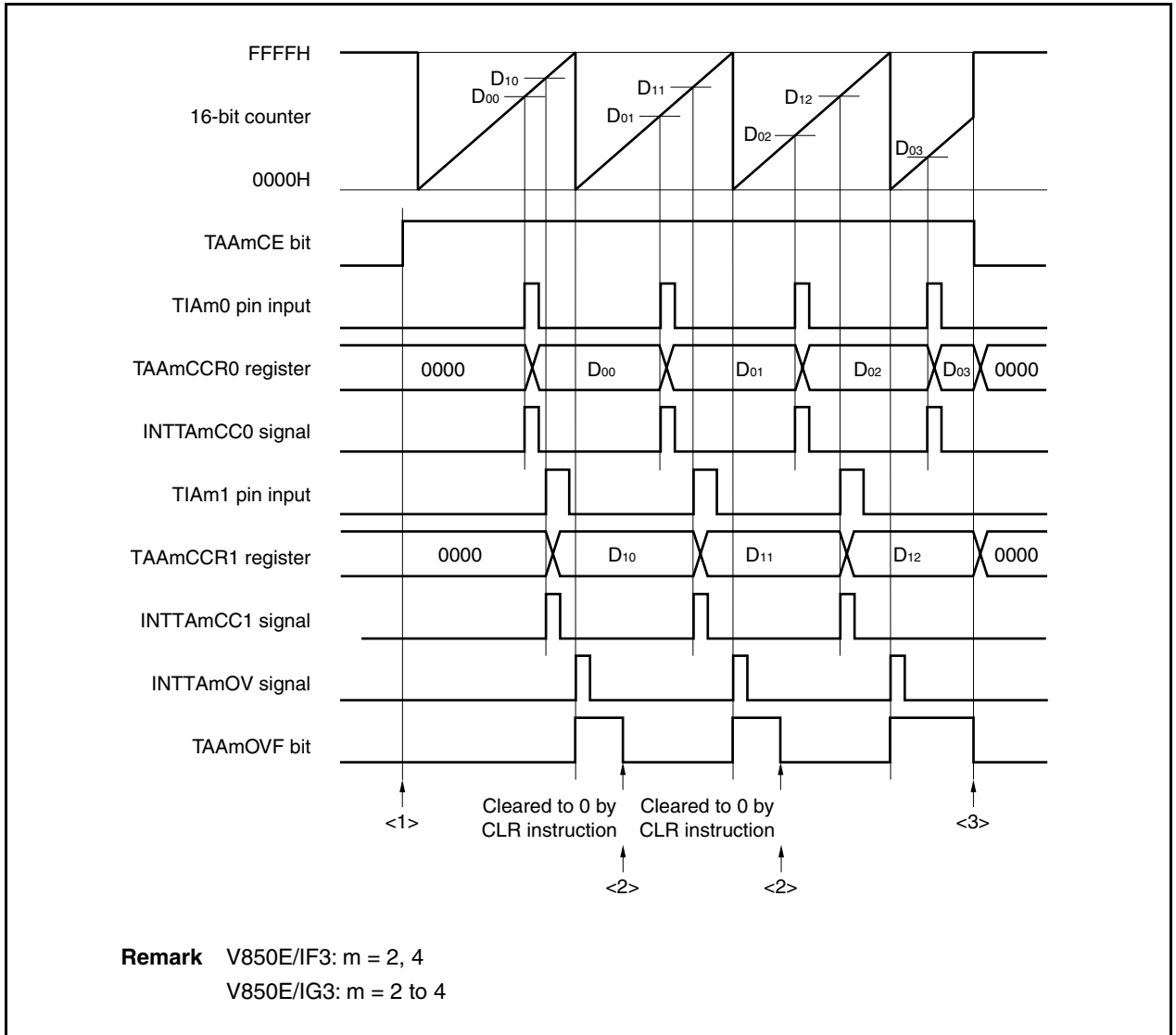
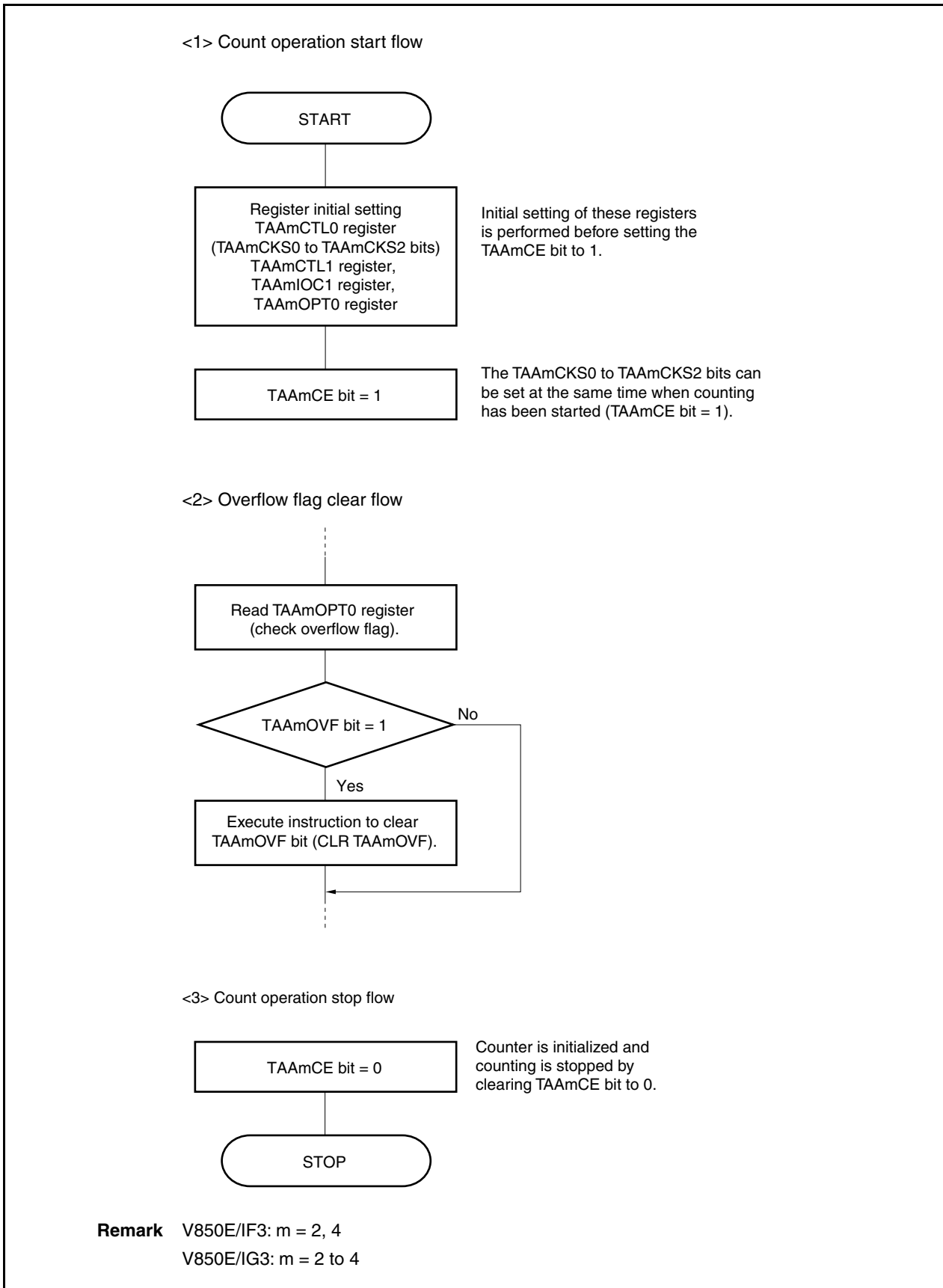


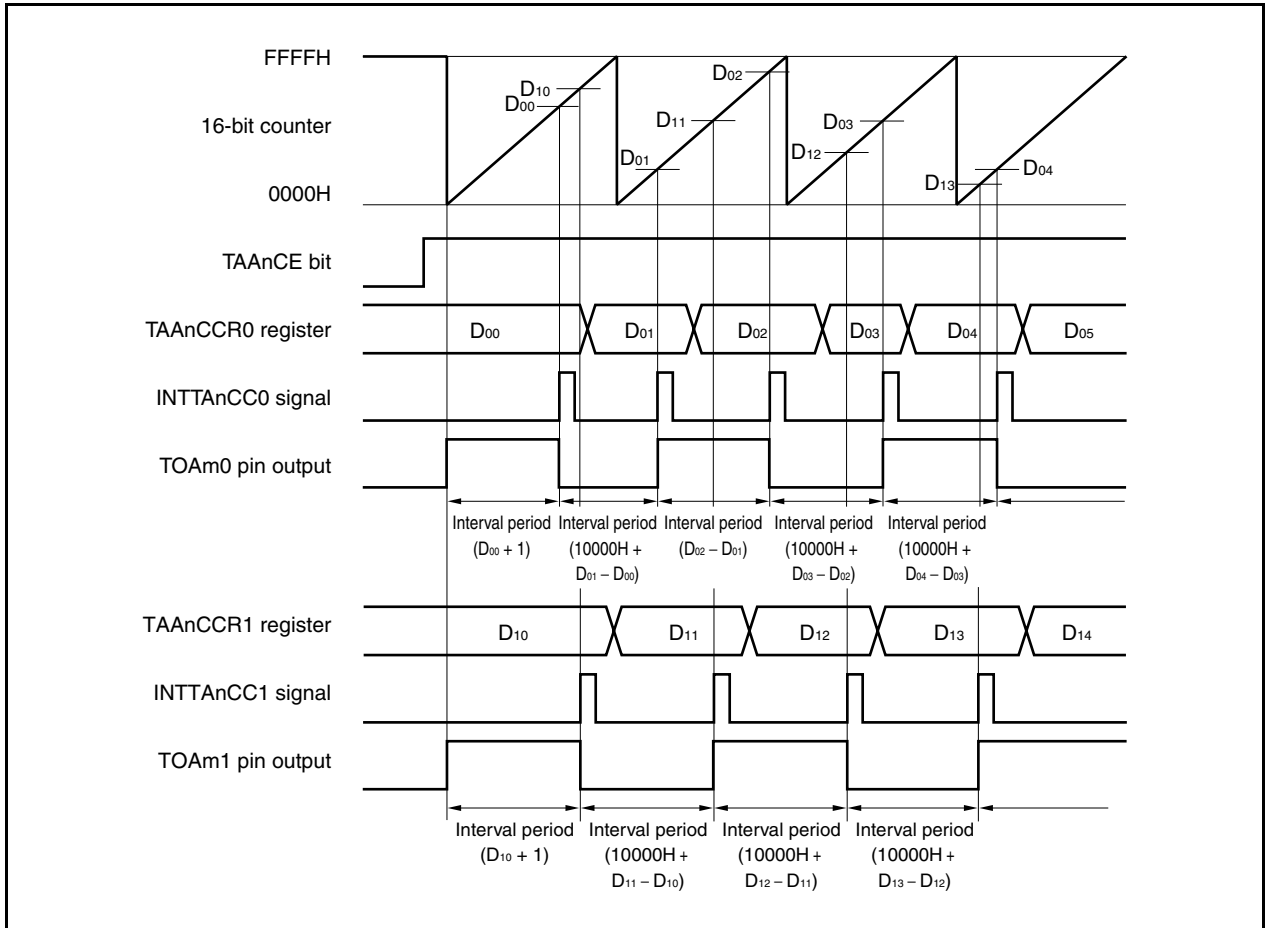
Figure 6-41. Software Processing Flow in Free-Running Timer Mode (Capture Function) (2/2)



(2) Operation timing in free-running timer mode

(a) Interval operation with compare register

When 16-bit timer/event counter AA is used as an interval timer with the TAAAnCCRa 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 INTTAnCCa 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 TAAAnCCRa register must be re-set in the interrupt servicing that is executed when the INTTAnCCa signal is detected.

The set value for re-setting the TAAAnCCRa register can be calculated by the following expression, where “D<sub>a</sub>” is the interval period.

Compare register default value:  $D_a - 1$

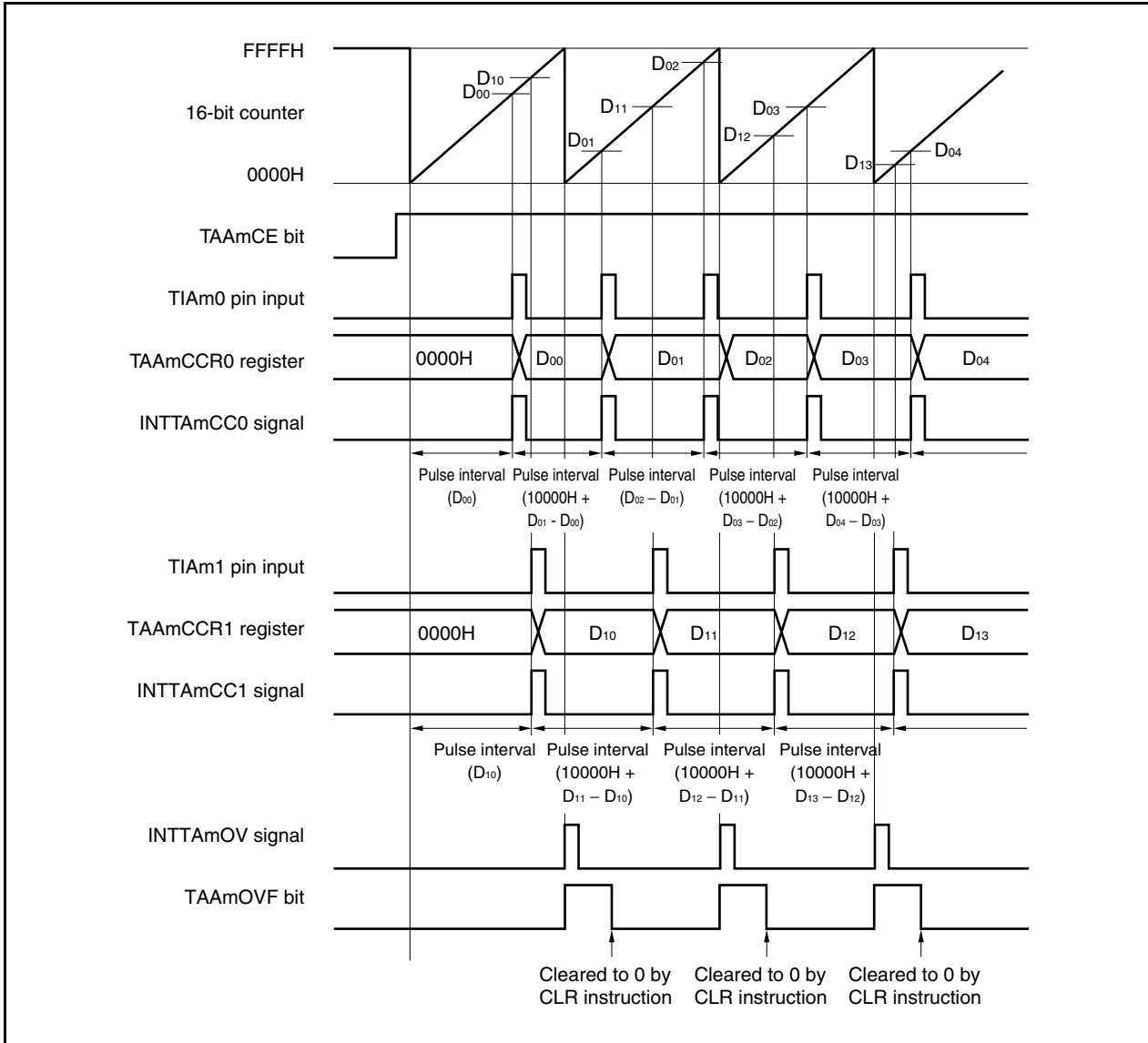
Value set to compare register second and subsequent time: Previous set value +  $D_a$

(If the calculation result is greater than FFFFH, subtract 10000H from the result and set this value to the register.)

**Remark** V850E/IF3: n = 0 to 4, m = 2, 4, a = 0, 1  
 V850E/IG3: n = 0 to 4, m = 2 to 4, a = 0, 1

**(b) Pulse width measurement with capture register**

When pulse width measurement is performed with the TAAmCCRa register used as a capture register, software processing is necessary for reading the capture register each time the INTTAmCCa signal has been detected and for calculating an interval.



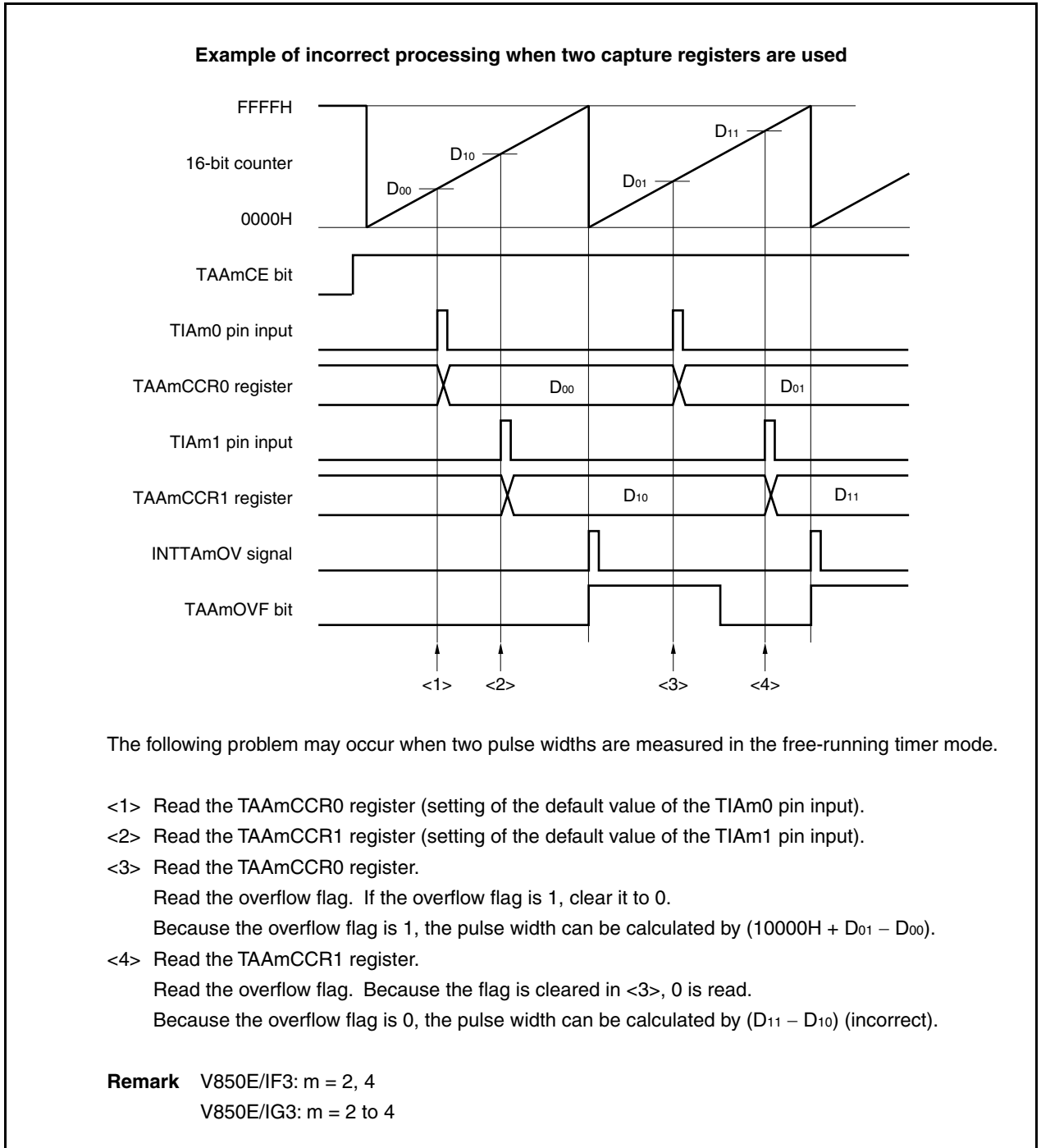
When executing pulse width measurement in the free-running timer mode, two pulse widths can be measured with one channel.

To measure a pulse width, the pulse width can be calculated by reading the value of the TAAmCCRa register in synchronization with the INTTAmCCa signal, and calculating the difference between the read value and the previously read value.

**Remark** V850E/IF3: m = 2, 4, a = 0, 1  
 V850E/IG3: m = 2 to 4, a = 0, 1

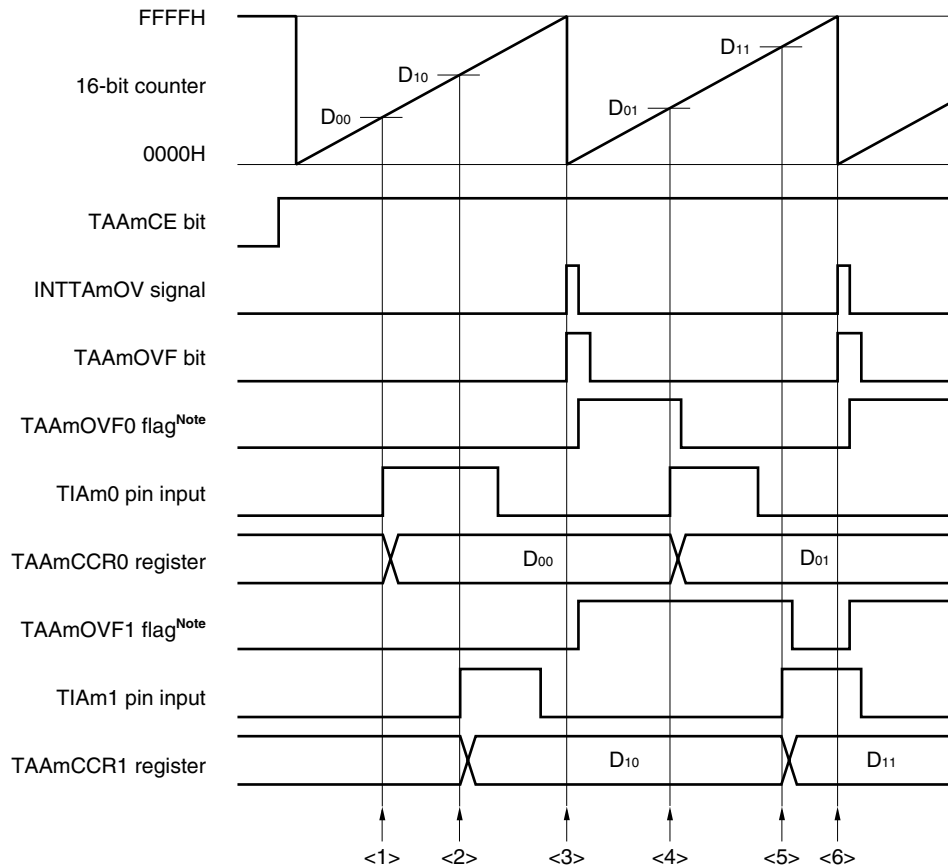
(c) Processing of overflow when two capture registers are used

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.

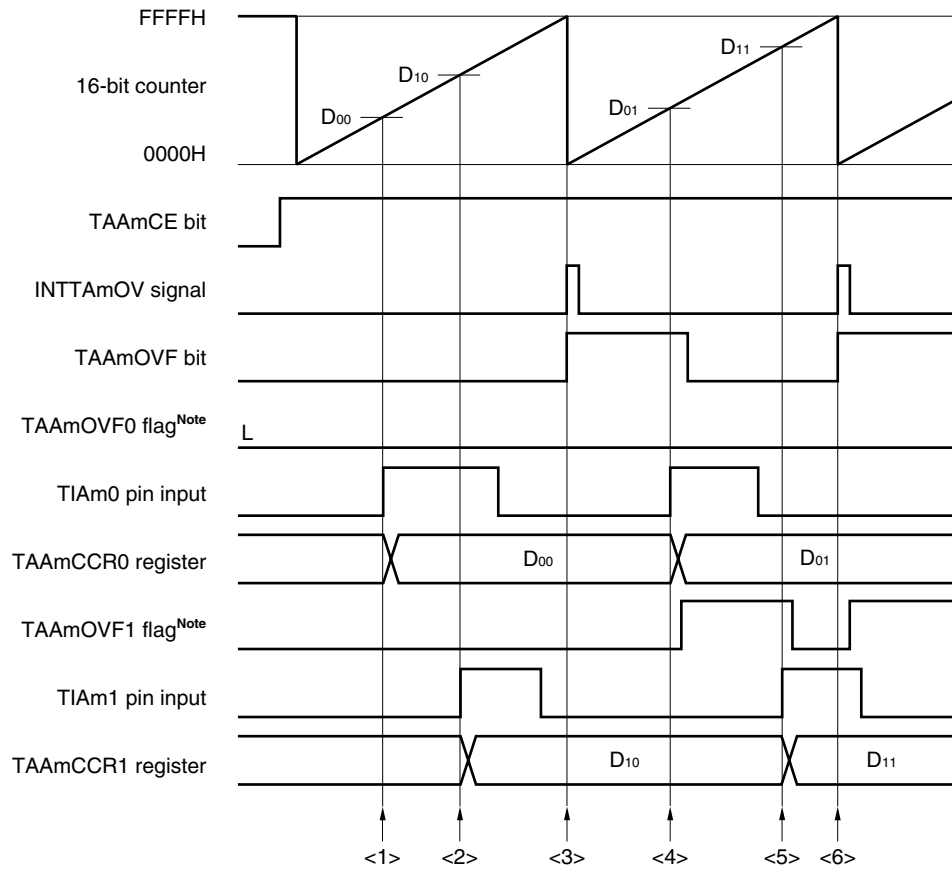
### Example when two capture registers are used (using overflow interrupt)



**Note** The TAAmOVF0 and TAAmOVF1 flags are set on the internal RAM by software.

- <1> Read the TAAmCCR0 register (setting of the default value of the TIAm0 pin input).
- <2> Read the TAAmCCR1 register (setting of the default value of the TIAm1 pin input).
- <3> An overflow occurs. Set the TAAmOVF0 and TAAmOVF1 flags to 1 in the overflow interrupt servicing, and clear the overflow flag to 0.
- <4> Read the TAAmCCR0 register.  
Read the TAAmOVF0 flag. If the TAAmOVF0 flag is 1, clear it to 0.  
Because the TAAmOVF0 flag is 1, the pulse width can be calculated by  $(10000H + D_{01} - D_{00})$ .
- <5> Read the TAAmCCR1 register.  
Read the TAAmOVF1 flag. If the TAAmOVF1 flag is 1, clear it to 0 (the TAAmOVF0 flag is cleared in <4>, and the TAAmOVF1 flag remains 1).  
Because the TAAmOVF1 flag is 1, the pulse width can be calculated by  $(10000H + D_{11} - D_{10})$  (correct).
- <6> Same as <3>

**Remark** V850E/IF3: m = 2, 4  
V850E/IG3: m = 2 to 4

**Example when two capture registers are used (without using overflow interrupt)**

**Note** The TAAmOVF0 and TAAmOVF1 flags are set on the internal RAM by software.

<1> Read the TAAmCCR0 register (setting of the default value of the TIAm0 pin input).

<2> Read the TAAmCCR1 register (setting of the default value of the TIAm1 pin input).

<3> An overflow occurs. Nothing is done by software.

<4> Read the TAAmCCR0 register.

Read the overflow flag. If the overflow flag is 1, set only the TAAmOVF1 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 TAAmCCR1 register.

Read the overflow flag. Because the overflow flag is cleared in <4>, 0 is read.

Read the TAAmOVF1 flag. If the TAAmOVF1 flag is 1, clear it to 0.

Because the TAAmOVF1 flag is 1, the pulse width can be calculated by  $(10000H + D_{11} - D_{10})$  (correct).

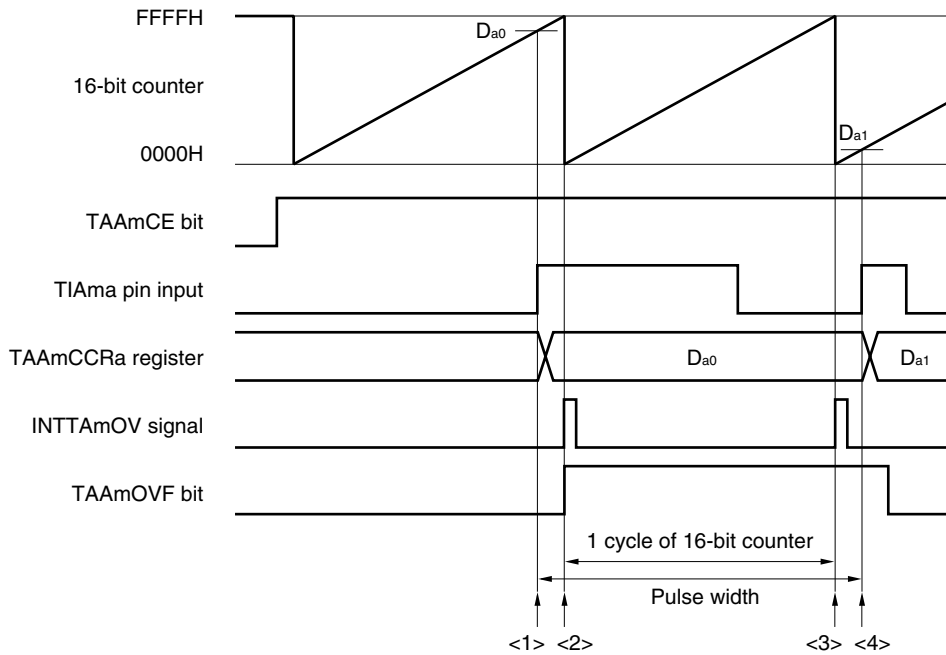
<6> Same as <3>

**Remark** V850E/IF3: m = 2, 4  
V850E/IG3: m = 2 to 4

**(d) Processing of overflow if capture trigger interval is long**

If the pulse width is greater than one cycle of the 16-bit counter, care must be exercised because an overflow may occur more than once from the first capture trigger to the next. First, an example of incorrect processing is shown below.

**Example of incorrect processing when capture trigger interval is long**



The following problem may occur when long pulse width is measured in the free-running timer mode.

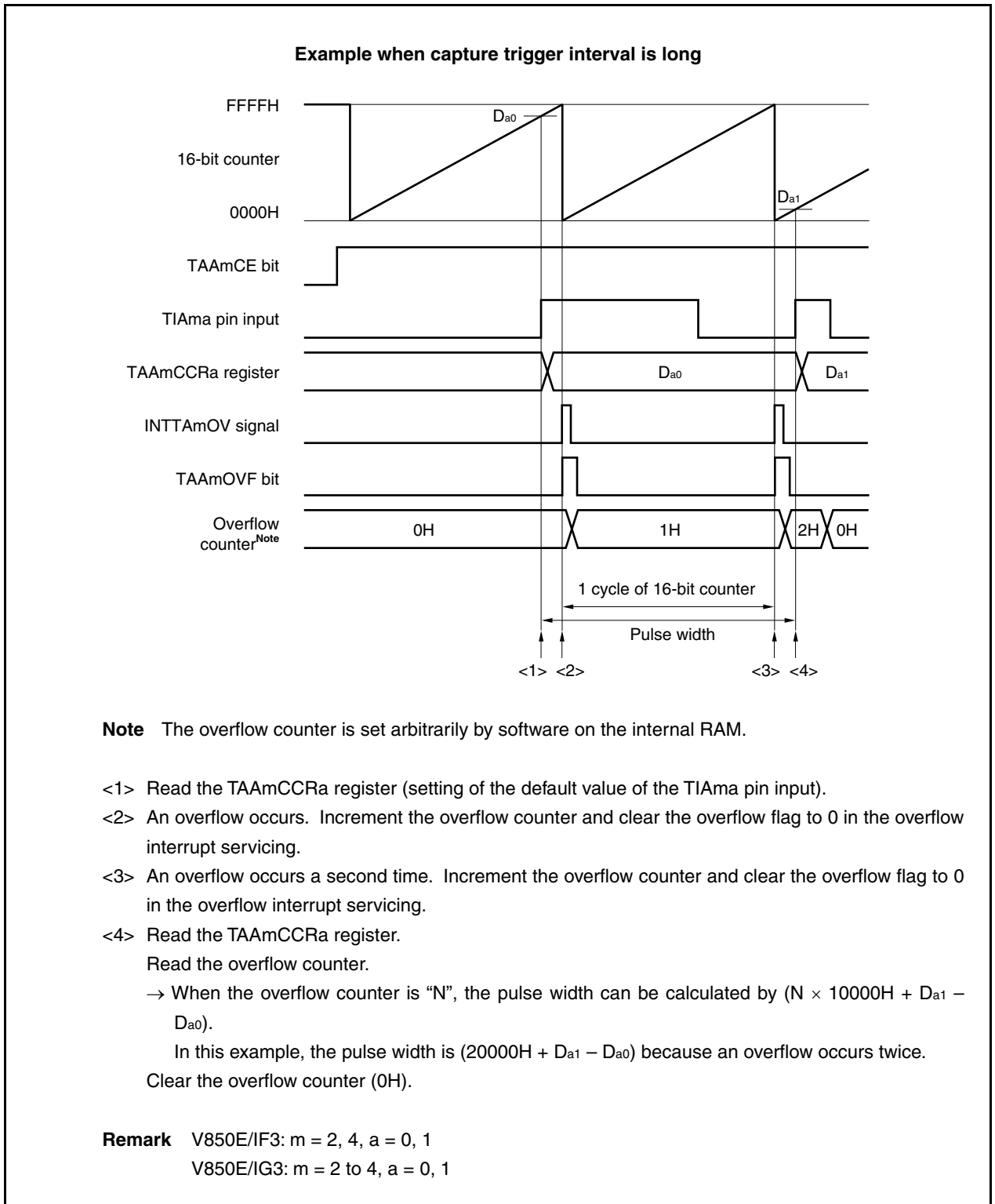
- <1> Read the TAAmCCRa register (setting of the default value of the TIama pin input).
- <2> An overflow occurs. Nothing is done by software.
- <3> An overflow occurs a second time. Nothing is done by software.
- <4> Read the TAAmCCRa register.  
 Read the overflow flag. If the overflow flag is 1, clear it to 0.  
 Because the overflow flag is 1, the pulse width can be calculated by  $(10000H + D_{a1} - D_{a0})$  (incorrect).  
 Actually, the pulse width must be  $(20000H + D_{a1} - D_{a0})$  because an overflow occurs twice.

**Remark** V850E/IF3: m = 2, 4, a = 0, 1  
 V850E/IG3: m = 2 to 4, a = 0, 1

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 TAAmOVF bit to 0 with the CLR instruction after reading the TAAmOVF bit when it is 1 and by writing 8-bit data (bit 0 is 0) to the TAAmOPT0 register after reading the TAAmOVF bit when it is 1.

**6.6.7 Pulse width measurement mode (TAAmMD2 to TAAmMD0 bits = 110)**

This mode is valid only in TAA2, TAA3 (V850E/IG3 only), and TAA4.

In the pulse width measurement mode, 16-bit timer/event counter AA starts counting when the TAAmCTL0.TAAmCE bit is set to 1. Each time the valid edge input to the TIAm<sub>a</sub> pin has been detected, the count value of the 16-bit counter is stored in the TAAmCCRa register, and the 16-bit counter is cleared to 0000H.

The interval of the valid edge can be measured by reading the TAAmCCRa register after a capture interrupt request signal (INTTAmCCa) occurs.

As shown in Figure 6-43, select either the TIAm<sub>0</sub> or TIAm<sub>1</sub> pin as the capture trigger input pin and set the unused pins to “No edge detection” by using the TAAmLOC1 register.

**Figure 6-42. Configuration in Pulse Width Measurement Mode**

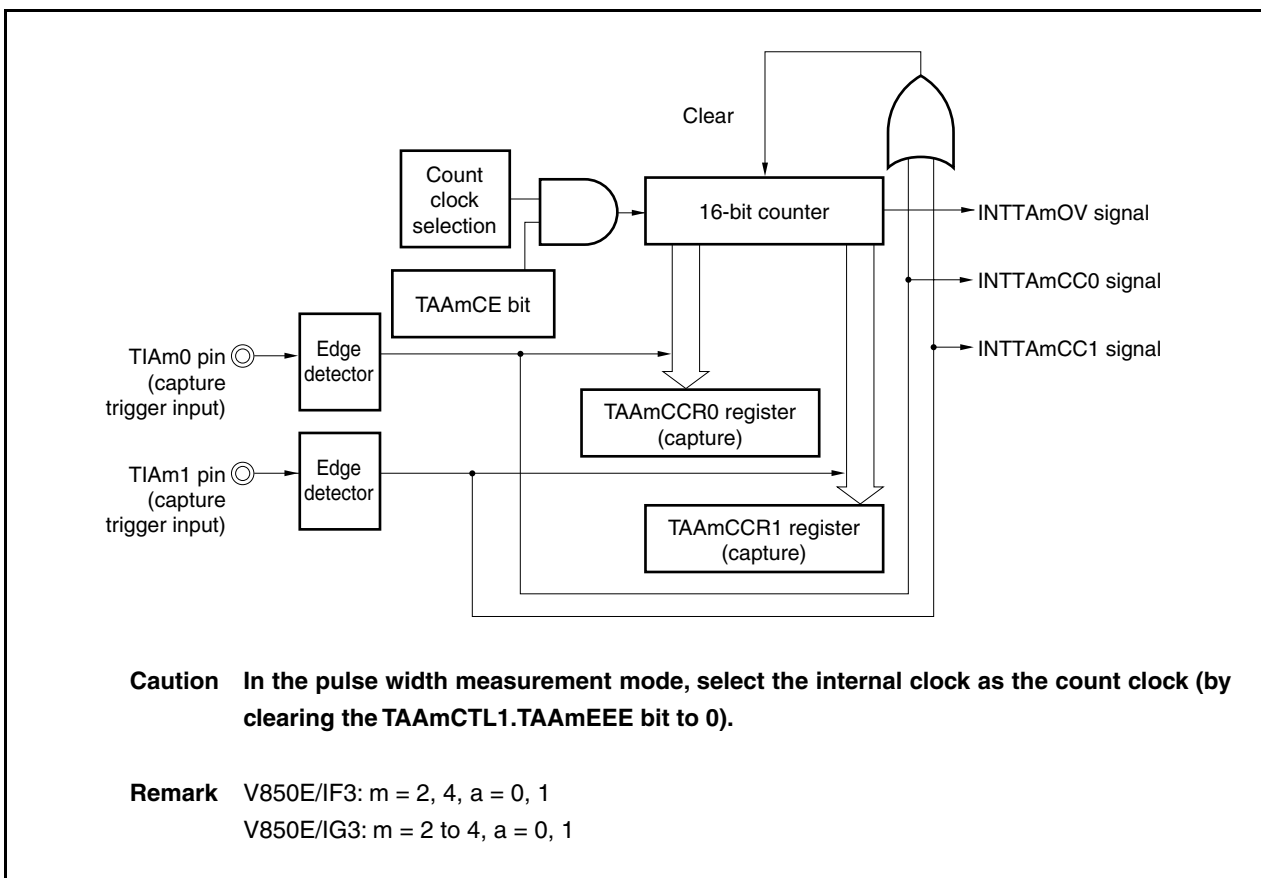
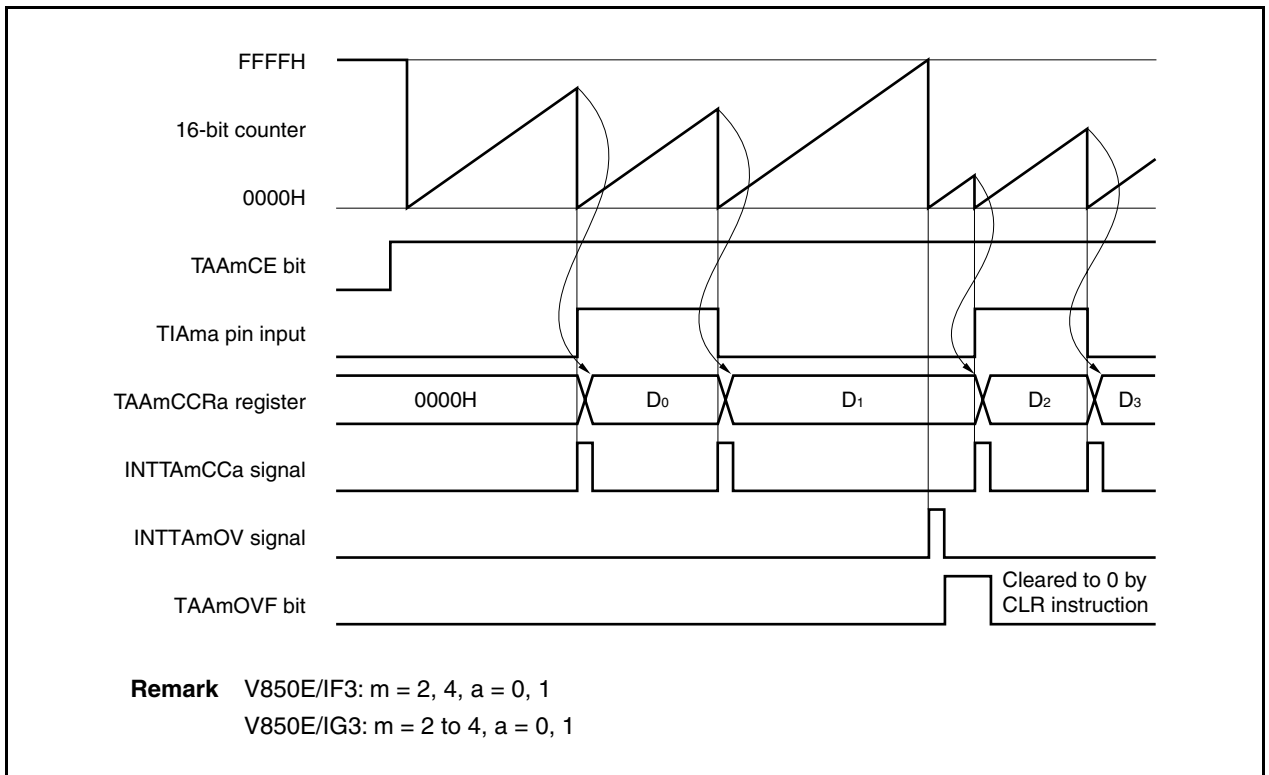


Figure 6-43. Basic Timing in Pulse Width Measurement Mode



When the TAAmCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIAMA pin is later detected, the count value of the 16-bit counter is stored in the TAAmCCRa register, the 16-bit counter is cleared to 0000H, and a capture interrupt request signal (INTTAmCCa) 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 TIAMA pin even when the 16-bit counter counted up to FFFFH, an overflow interrupt request signal (INTTAmOV) is generated at the next count clock, and the counter is cleared to 0000H and continues counting. At this time, the overflow flag (TAAmOPT0.TAAmOVF 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.

$$\text{Pulse width} = (10000\text{H} \times \text{TAAmOVF bit set (1) count} + \text{Captured value}) \times \text{Count clock cycle}$$

**Remark** V850E/IF3: m = 2, 4, a = 0, 1  
 V850E/IG3: m = 2 to 4, a = 0, 1

Figure 6-44. Register Setting in Pulse Width Measurement Mode (1/2)

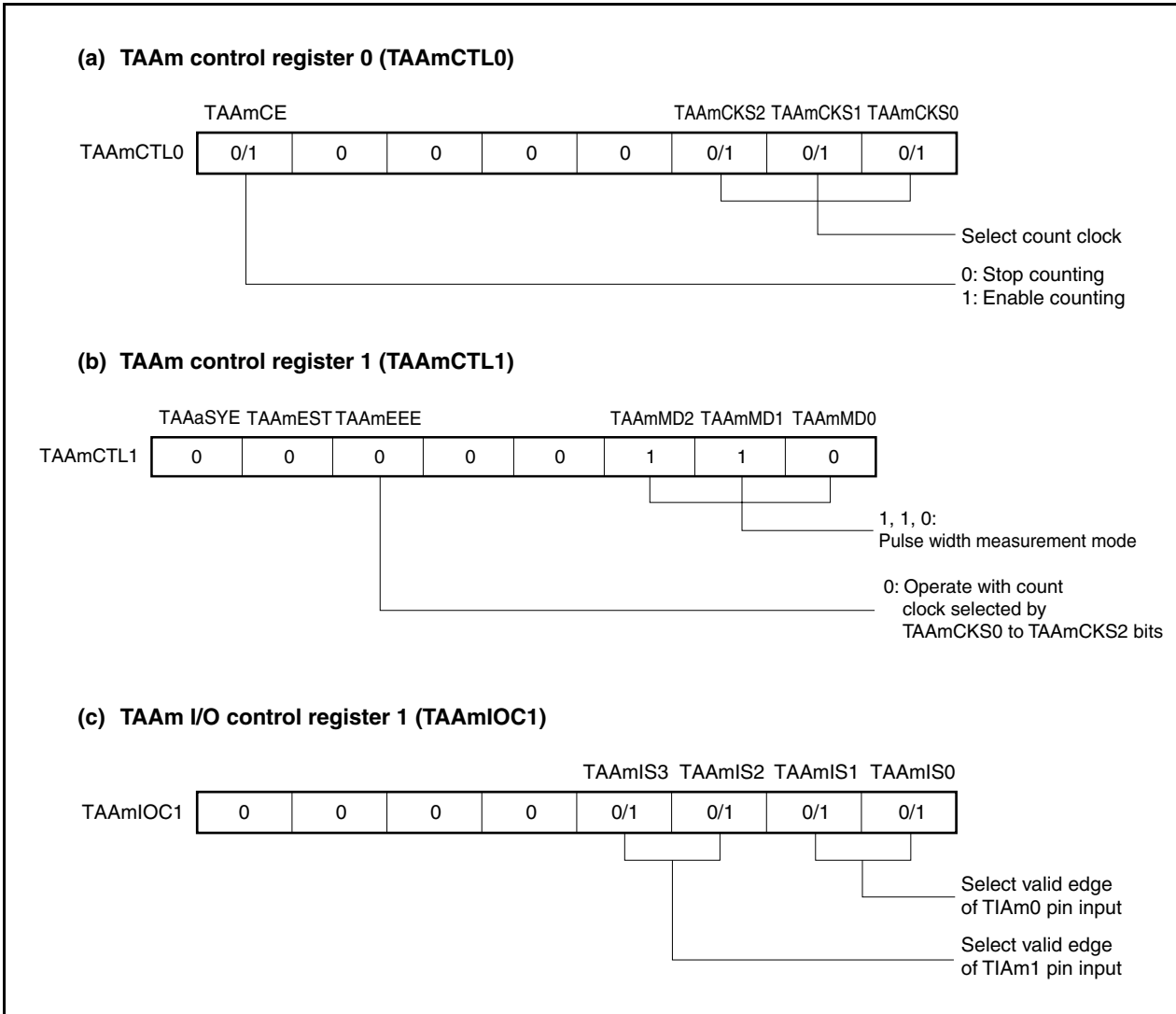
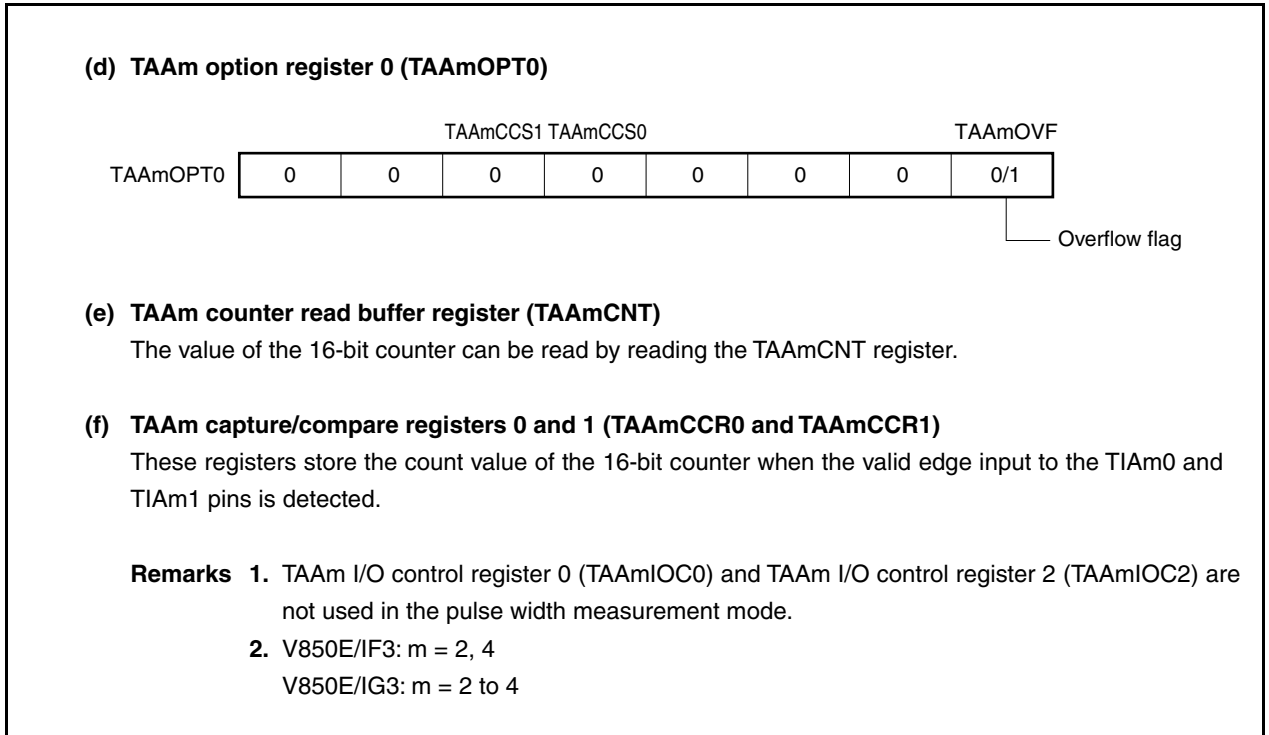
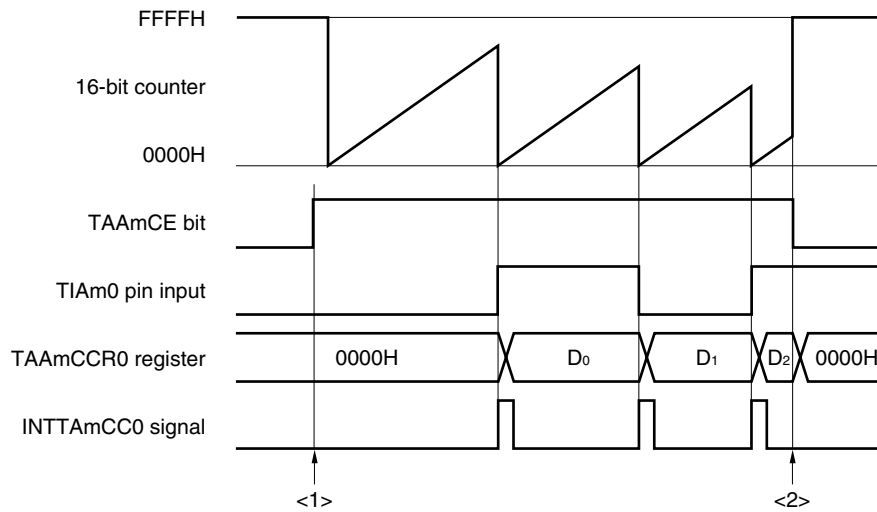


Figure 6-44. Register Setting in Pulse Width Measurement Mode (2/2)

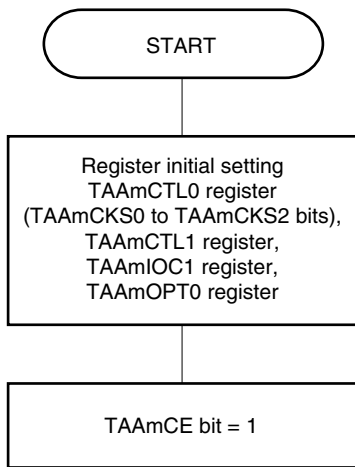


(1) Operation flow in pulse width measurement mode

Figure 6-45. Software Processing Flow in Pulse Width Measurement Mode



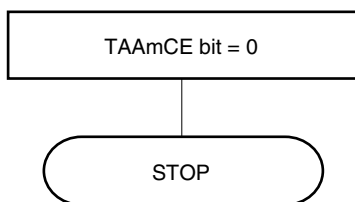
<1> Count operation start flow



Initial setting of these registers is performed before setting the TAAmCE bit to 1.

The TAAmCKS0 to TAAmCKS2 bits can be set at the same time when counting has been started (TAAmCE bit = 1).

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TAAmCE bit to 0.

**Remark** V850E/IF3: m = 2, 4  
V850E/IG3: m = 2 to 4

**(2) Operation timing in pulse width measurement mode**

**(a) Clearing overflow flag**

The overflow flag can be cleared to 0 by clearing the TAAmOVF bit to 0 with the CLR instruction after reading the TAAmOVF bit when it is 1 and by writing 8-bit data (bit 0 is 0) to the TAAmOPT0 register after reading the TAAmOVF bit when it is 1.

## CHAPTER 7 16-BIT TIMER/EVENT COUNTER AB (TAB)

Timer AB (TAB) is a 16-bit timer/event counter.

The V850E/IF3 and V850E/IG3 incorporate TAB0 and TAB1.

### 7.1 Overview

An outline of TAB<sub>n</sub> is shown below (n = 0, 1).

- 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
- Overflow interrupt request signal: 1
- Timer output pins<sup>Note</sup>: 4

**Note** This is the number of output pins of TAB<sub>n</sub>; it does not include the output pins of TMQOP<sub>n</sub>. For details of the output pins of TMQOP<sub>n</sub>, see **CHAPTER 10 MOTOR CONTROL FUNCTION**.

### 7.2 Functions

TAB<sub>n</sub> has the following functions (n = 0, 1).

- 6-phase PWM output<sup>Note</sup>
- Interval timer
- External event counter
- External trigger pulse output
- One-shot pulse output
- PWM output
- Free-running timer
- Pulse width measurement

**Note** This is connected to TMQOP<sub>n</sub>. For details, see **CHAPTER 10 MOTOR CONTROL FUNCTION**.



### 7.3 Configuration

TABn includes the following hardware (n = 0, 1).

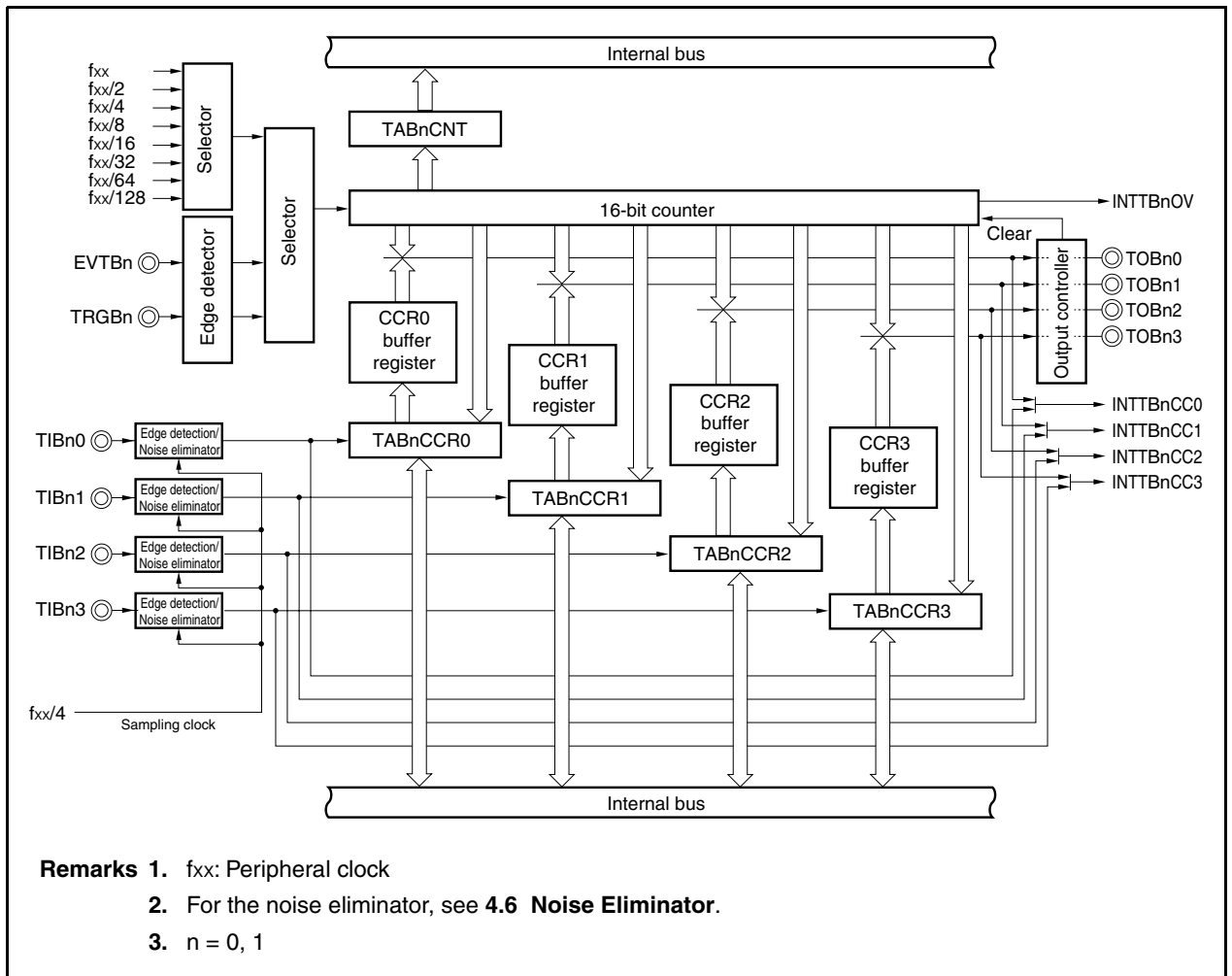
**Table 7-1. TABn Configuration**

| Item              | Configuration   |
|-------------------|---|
| Timer register    | 16-bit counter × 1  |
| Registers         | TABn counter read buffer register (TABnCNT)<br>TABn capture/compare registers 0 to 3 (TABnCCR0 to TABnCCR3)<br>CCR0 to CCR3 buffer registers      |
| Timer input       | 12 in total (TIB00 to TIB03, TIB10 to TIB13, EVTb0, EVTb1, TRGB0, TRGB1 pins) <sup>Note</sup>   |
| Timer output      | 8 in total (TOB00 to TOB03, TOB10 to TOB13 pins) <sup>Note</sup>  |
| Control registers | TABn control registers 0, 1 (TABnCTL0, TABnCTL1)<br>TABn I/O control registers 0 to 2 (TABnIOC0 to TABnIOC2)<br>TABn option register 0 (TABnOPT0) |

**Note** The TIBn1 to TIBn3 pins function alternately as timer output pins (TOBn1 to TOBn3).

**Remark** n = 0, 1

**Figure 7-1. TABn Block Diagram**



**(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 TABnCNT register.

When the TABnCTL0.TABnCE bit = 0, the value of the 16-bit counter is FFFFH. If the TABnCNT register is read at this time, 0000H is read.

Reset sets the TABnCE bit to 0.

**(2) CCR0 buffer register**

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

When the TABnCCR0 register is used as a compare register, the value written to the TABnCCR0 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 (INTTBnCC0) is generated.

The CCR0 buffer register cannot be read or written directly.

The CCR0 buffer register is cleared to 0000H after reset, and the TABnCCR0 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 TABnCCR1 register is used as a compare register, the value written to the TABnCCR1 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 (INTTBnCC1) is generated.

The CCR1 buffer register cannot be read or written directly.

The CCR1 buffer register is cleared to 0000H after reset, and the TABnCCR1 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 TABnCCR2 register is used as a compare register, the value written to the TABnCCR2 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 (INTTBnCC2) is generated.

The CCR2 buffer register cannot be read or written directly.

The CCR2 buffer register is cleared to 0000H after reset, and the TABnCCR2 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 TABnCCR3 register is used as a compare register, the value written to the TABnCCR3 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 (INTTBnCC3) is generated.

The CCR3 buffer register cannot be read or written directly.

The CCR3 buffer register is cleared to 0000H after reset, and the TABnCCR3 register is cleared to 0000H.

**(6) Edge detector**

This circuit detects the valid edges input to the TIBn0 to TIBn3, EVTn, and TRGBn pins. No edge, rising edge, falling edge, or both the rising and falling edges can be selected as the valid edge by using the TABnIOC1 and TABnIOC2 registers.

**(7) Output controller**

This circuit controls the output of the TOBn0 to TOBn3 pins. The output controller is controlled by the TABnIOC0 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

### (1) TABn control register 0 (TABnCTL0)

The TABnCTL0 register is an 8-bit register that controls the operation of TABn.

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 TABnCTL0 register by software.

After reset: 00H R/W Address: TAB0CTL0 FFFFF5E0H, TAB1CTL0 FFFFF620H

|          |        |   |   |   |   |          |          |          |
|----------|--------|---|---|---|---|----------|----------|----------|
|          | <7>    | 6 | 5 | 4 | 3 | 2        | 1        | 0        |
| TABnCTL0 | TABnCE | 0 | 0 | 0 | 0 | TABnCKs2 | TABnCKs1 | TABnCKs0 |

(n = 0, 1)

|        |  |
|--------|--|
| TABnCE | TABn operation control   |
| 0      | TABn operation disabled (TABn reset asynchronously <sup>Note</sup> ) |
| 1      | TABn operation enabled.  |

|          |          |          |                                |
|----------|----------|----------|--------------------------------|
| TABnCKs2 | TABnCKs1 | TABnCKs0 | Internal count clock selection |
| 0        | 0        | 0        | f <sub>xx</sub>                |
| 0        | 0        | 1        | f <sub>xx</sub> /2             |
| 0        | 1        | 0        | f <sub>xx</sub> /4             |
| 0        | 1        | 1        | f <sub>xx</sub> /8             |
| 1        | 0        | 0        | f <sub>xx</sub> /16            |
| 1        | 0        | 1        | f <sub>xx</sub> /32            |
| 1        | 1        | 0        | f <sub>xx</sub> /64            |
| 1        | 1        | 1        | f <sub>xx</sub> /128           |

**Note** The TABnOPT0.TABnOVF bit and the 16-bit counter are reset simultaneously. Moreover, timer outputs (TOBn0 to TOBn3 pins) are reset to the TABnIOC0 register set status at the same time as the 16-bit counter is reset.

**Cautions 1.** Set the TABnCKs2 to TABnCKs0 bits when the TABnCE bit = 0.

When the value of the TABnCE bit is changed from 0 to 1, the TABnCKs2 to TABnCKs0 bits can be set simultaneously.

**2.** Be sure to set bits 3 to 6 to "0".

**Remark** f<sub>xx</sub>: Peripheral clock

**(2) TABn control register 1 (TABnCTL1)**

The TABnCTL1 register is an 8-bit register that controls the operation of TABn.

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: TAB0CTL1 FFFF5E1H, TAB1CTL1 FFFF621H

|          |   |         |         |   |   |         |         |         |
|----------|---|---------|---------|---|---|---------|---------|---------|
|          | 7 | 6       | 5       | 4 | 3 | 2       | 1       | 0       |
| TABnCTL1 | 0 | TABnEST | TABnEEE | 0 | 0 | TABnMD2 | TABnMD1 | TABnMD0 |

(n = 0, 1)

|  |   |
|--|---|
| TABnEST                                    | Software trigger control  |
| 0  | –   |
| 1  | Generate a valid signal for external trigger input.<br><ul style="list-style-type: none"> <li>In one-shot pulse output mode:<br/>A one-shot pulse is output with writing 1 to the TABnEST bit as the trigger.</li> <li>In external trigger pulse output mode:<br/>A PWM waveform is output with writing 1 to the TABnEST bit as the trigger.</li> </ul> |
| Read value of the TABnEST bit is always 0. |   |

|  |   |
|--|---|
| TABnEEE  | Count clock selection   |
| 0  | Disable operation with external event count input (EVTBn pin).<br>(Perform counting with the count clock selected by the TABnCTL0.TABnCKS0 to TABnCKS2 bits.) |
| 1  | Enable operation with external event count input (EVTBn pin).<br>(Perform counting at the valid edge of the external event count input signal (EVTBn pin).)   |
| The TABnEEE bit selects whether counting is performed with the internal count clock or the valid edge of the external event count input. |   |

|         |         |         |   |
|---------|---------|---------|---|
| TABnMD2 | TABnMD1 | TABnMD0 | 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       | 6-phase PWM output mode <sup>Note</sup> |

**Note** The 6-phase PWM output mode cannot be used when only TABn is used. For details, see **CHAPTER 10 MOTOR CONTROL FUNCTION**.

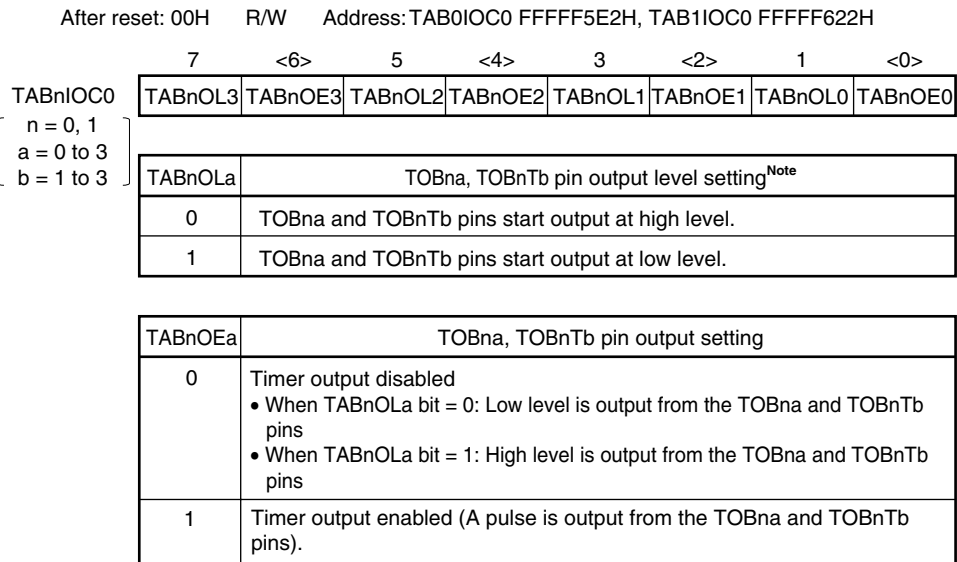
- Cautions**
1. The TABnEST bit is valid only in the external trigger pulse output mode or one-shot pulse output mode. In any other mode, writing 1 to this bit is ignored.
  2. External event count input is selected in the external event count mode regardless of the value of the TABnEEE bit.
  3. Set the TABnEEE and TABnMD2 to TABnMD0 bits when the TABnCTL0.TABnCE bit = 0. (The same value can be written when the TABnCE bit = 1.) The operation is not guaranteed when rewriting is performed with the TABnCE bit = 1. If rewriting was mistakenly performed, clear the TABnCE bit to 0 and then set the bits again.
  4. Be sure to set bits 3, 4, and 7 to “0”.

**(3) TABn I/O control register 0 (TABnIOC0)**

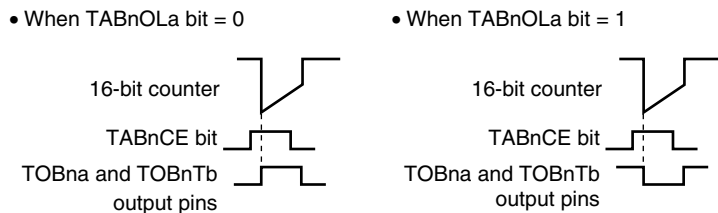
The TABnIOC0 register is an 8-bit register that controls the timer output (TOBn0 to TOBn3, TOBnT1 to TOBnT3 pins).

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

Reset sets this register to 00H.



**Note** The output level of the timer output pins (TOBna and TOBnTb) specified by the TABnOLa bit is shown below.



- Cautions**
1. If the setting of the TABnIOC0 register is changed when TOBna and TOBnTb are set in the output mode, the output of the pins change. Set the port in the input mode and make the port go into a high-impedance state, noting changes in the pin status.
  2. Rewrite the TABnOLa and TABnOEa bits when the TABnCTL.TABnCE bit = 0. (The same value can be written when the TABnCE bit = 1.) If rewriting was mistakenly performed, clear (0) the TABnCE bit and then set the bits again.
  3. If the TABnOLa bit is manipulated when the TABnCE and TABnOEa bits are 0, the output level of the TOBna and TOBnTb pins changes.
  4. To generate the TOBnTb pin output and the A/D conversion start trigger signal of A/D converters 0 and 1 in the 6-phase PWM output mode, be sure to set the TOBnTb pin output using the TABnIOC0 register. At this time, be sure to clear the TABnOL0 bit to 0 and set the TABnOE0 bit to 1.

**(4) TABn I/O control register 1 (TABnIOC1)**

The TABnIOC1 register is an 8-bit register that controls the valid edge of the capture trigger input signals (TIBn0 to TIBn3 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: TAB0IOC1 FFFFF5E3H, TAB1IOC1 FFFFF623H

|          |         |         |         |         |         |         |         |         |
|----------|---------|---------|---------|---------|---------|---------|---------|---------|
|          | 7       | 6       | 5       | 4       | 3       | 2       | 1       | 0       |
| TABnIOC1 | TABnIS7 | TABnIS6 | TABnIS5 | TABnIS4 | TABnIS3 | TABnIS2 | TABnIS1 | TABnIS0 |

(n = 0, 1)

| TABnIS7 | TABnIS6 | Capture trigger input signal (TIBn3 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                                     |

| TABnIS5 | TABnIS4 | Capture trigger input signal (TIBn2 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                                       |

| TABnIS3 | TABnIS2 | Capture trigger input signal (TIBn1 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                                     |

| TABnIS1 | TABnIS0 | Capture trigger input signal (TIBn0 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 TABnIS7 to TABnIS0 bits when the TABnCTL0.TABnCE bit = 0. (The same value can be written when the TABnCE bit = 1.) If rewriting was mistakenly performed, clear the TABnCE bit to 0 and then set the bits again.
  2. The TABnIS7 to TABnIS0 bits are valid only in the free-running timer mode (only when the TABnOPT0.TABnCCSa bit = 1) and the pulse width measurement mode (a = 0 to 3). In all other modes, a capture operation is not possible.

**(5) TABn I/O control register 2 (TABnIOC2)**

The TABnIOC2 register is an 8-bit register that controls the valid edge of the external event count input signal (EVTBn pin) and external trigger input signal (TRGBn 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: TAB0IOC2 FFFFF5E4H, TAB1IOC2 FFFFF624H

|          |   |   |   |   |          |          |          |          |
|----------|---|---|---|---|----------|----------|----------|----------|
|          | 7 | 6 | 5 | 4 | 3        | 2        | 1        | 0        |
| TABnIOC2 | 0 | 0 | 0 | 0 | TABnEES1 | TABnEES0 | TABnETS1 | TABnETS0 |

(n = 0, 1)

| TABnEES1 | TABnEES0 | External event count input signal (EVTBn 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  |

| TABnETS1 | TABnETS0 | External trigger input signal (TRGBn 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 TABnEES1, TABnEES0, TABnETS1, and TABnETS0 bits when the TABnCTL0.TABnCE bit = 0. (The same value can be written when the TABnCE bit = 1.) If rewriting was mistakenly performed, clear the TABnCE bit to 0 and then set the bits again.
  2. The TABnEES1 and TABnEES0 bits are valid only when the TABnCTL1.TABnEEE bit = 1 or when the external event count mode (TABnCTL1.TABnMD2 to TABnCTL1.TABnMD0 bits = 001) has been set.
  3. The TABnETS1 and TABnETS0 bits are valid only in the external trigger pulse output mode or one-shot pulse output mode.

**(6) TABn option register 0 (TABnOPT0)**

The TABnOPT0 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: TAB0OPT0 FFFFF5E5H, TAB1OPT0 FFFFF625H

|          |          |          |          |          |   |                         |                         |         |
|----------|----------|----------|----------|----------|---|-------------------------|-------------------------|---------|
|          | <7>      | <6>      | <5>      | <4>      | 3 | <2>                     | <1>                     | <0>     |
| TABnOPT0 | TABnCCS3 | TABnCCS2 | TABnCCS1 | TABnCCS0 | 0 | TABnCMS <sup>Note</sup> | TABnCUF <sup>Note</sup> | TABnOVF |

$\left[ \begin{array}{l} n = 0, 1 \\ a = 0 \text{ to } 3 \end{array} \right]$

|  |  |
|--|--|
| TABnCCSa   | TABnCCRa register capture/compare selection                    |
| 0  | Compare register selected                                      |
| 1  | Capture register selected (cleared by TABnCTL0.TABnCE bit = 0) |
| The TABnCCSa bit setting is valid only in the free-running timer mode. |  |

|   |  |
|---|--|
| TABnOVF   | TABn overflow flag                               |
| Set (1)   | Overflow occurred                                |
| Reset (0)   | TABnOVF bit 0 written or TABnCTL0.TABnCE bit = 0 |
| <ul style="list-style-type: none"> <li>• The TABnOVF 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.</li> <li>• An overflow interrupt request signal (INTTBnOV) is generated at the same time that the TABnOVF bit is set to 1. The INTTBnOV signal is not generated in modes other than the free-running timer mode and the pulse width measurement mode.</li> <li>• The TABnOVF bit is not cleared to 0 even when the TABnOVF bit or the TABnOPT0 register are read when the TABnOVF bit = 1.</li> <li>• Before clearing the TABnOVF bit to 0 after generation of the INTTBnOV signal, be sure to confirm (by reading) that the TABnOVF bit is set to 1.</li> <li>• The TABnOVF bit can be both read and written, but the TABnOVF bit cannot be set to 1 by software. Writing 1 has no influence on the operation of TABn.</li> </ul> |  |

**Note** For details of the TABnCMS and TABnCUF bits, see **CHAPTER 10 MOTOR CONTROL FUNCTION**.

**Cautions**

1. Rewrite the TABnCCS3 to TABnCCS0 bits when the TABnCE bit = 0. (The same value can be written when the TABnCE bit = 1.) If rewriting was mistakenly performed, clear the TABnCE bit to 0 and then set the bits again.
2. Be sure to set bit 3 to “0”.



**(7) TABn capture/compare register 0 (TABnCCR0)**

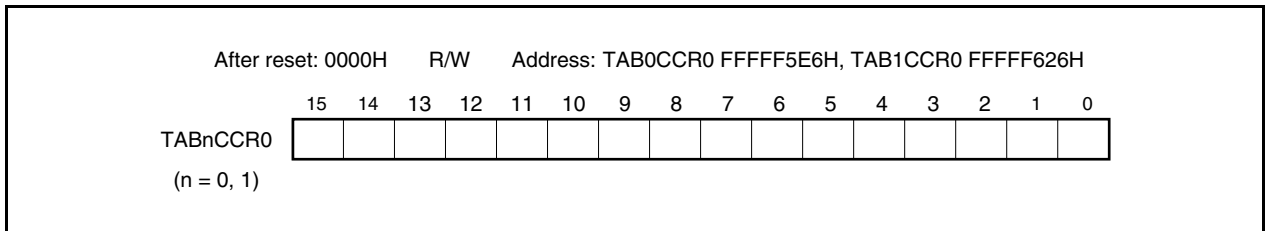
The TABnCCR0 register is a 16-bit register that can be used as a capture register or a compare register depending on the mode.

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 TABnOPT0.TABnCCS0 bit. In the pulse width measurement mode, the TABnCCR0 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TABnCCR0 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.



**(a) Function as compare register**

The TABnCCR0 register can be rewritten even when the TABnCTL0.TABnCE bit = 1.

The set value of the TABnCCR0 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 (INTTBnCC0) is generated. If TOBn0 pin output is enabled at this time, the output of the TOBn0 pin is inverted.

When the TABnCCR0 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.

The compare register is not cleared by setting the TABnCTL0.TABnCE bit to 0.

**(b) Function as capture register**

When the TABnCCR0 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 TABnCCR0 register if the valid edge of the capture trigger input pin (TIBn0 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TABnCCR0 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIBn0 pin) is detected.

Even if the capture operation and reading the TABnCCR0 register conflict, the correct value of the TABnCCR0 register can be read.

The capture register is cleared by setting the TABnCTL0.TABnCE bit = 0.

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 <sup>Note</sup>   |
| One-shot pulse output         | Compare register         | Anytime write                 |
| PWM output                    | Compare register         | Batch write <sup>Note</sup>   |
| Free-running timer            | Capture/compare register | Anytime write                 |
| Pulse width measurement       | Capture register         | None                          |

**Note** Writing to the TABnCCR1 register is the trigger.

**Remark** For anytime write and batch write, see 7.6 (2) **Anytime write and batch write**.

**(8) TABn capture/compare register 1 (TABnCCR1)**

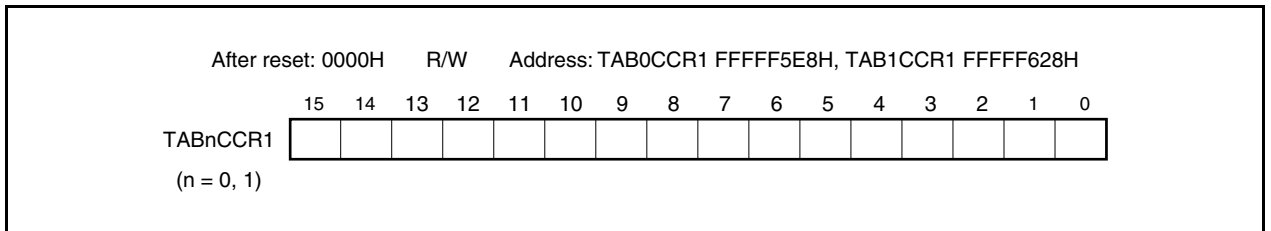
The TABnCCR1 register, which consists of 16 bits, can be used as a capture register or a compare register depending on the mode.

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 TABnOPT0.TABnCCS1 bit. In the pulse width measurement mode, the TABnCCR1 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TABnCCR1 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.



**(a) Function as compare register**

The TABnCCR1 register can be rewritten even when the TABnCTL0.TABnCE bit = 1.

The set value of the TABnCCR1 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 (INTTBnCC1) is generated. If TOBn1 pin output is enabled at this time, the output of the TOBn1 pin is inverted.

The compare register is not cleared by setting the TABnCTL0.TABnCE bit to 0.

**(b) Function as capture register**

When the TABnCCR1 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 TABnCCR1 register if the valid edge of the capture trigger input pin (TIBn1 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TABnCCR1 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIBn1 pin) is detected.

Even if the capture operation and reading the TABnCCR1 register conflict, the correct value of the TABnCCR1 register can be read.

The capture register is cleared by setting the TABnCTL0.TABnCE bit to 0.

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 <sup>Note</sup>   |
| One-shot pulse output         | Compare register         | Anytime write                 |
| PWM output                    | Compare register         | Batch write <sup>Note</sup>   |
| Free-running timer            | Capture/compare register | Anytime write                 |
| Pulse width measurement       | Capture register         | None                          |

**Note** Writing to the TABnCCR1 register is the trigger.

**Remark** For anytime write and batch write, see 7.6 (2) **Anytime write and batch write**.

**(9) TABn capture/compare register 2 (TABnCCR2)**

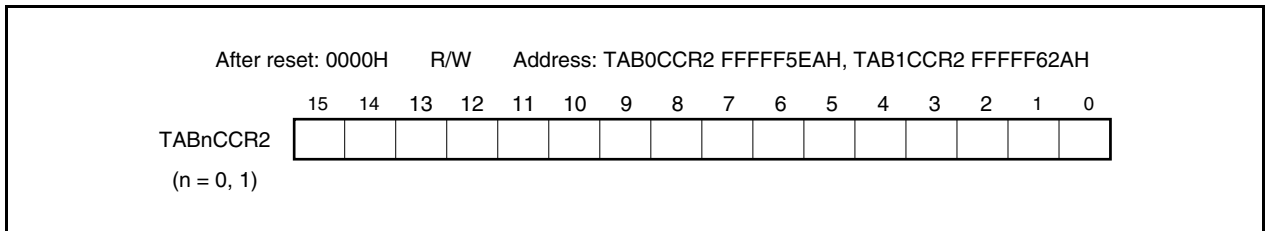
The TABnCCR2 register is a 16-bit register that can be used as a capture register or a compare register depending on the mode.

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 TABnOPT0.TABnCCS2 bit. In the pulse width measurement mode, the TABnCCR2 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TABnCCR2 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.



**(a) Function as compare register**

The TABnCCR2 register can be rewritten even when the TABnCTL0.TABnCE bit = 1.

The set value of the TABnCCR2 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 (INTTBnCC2) is generated. If TOBn2 pin output is enabled at this time, the output of the TOBn2 pin is inverted.

The compare register is not cleared by setting the TABnCTL0.TABnCE bit to 0.

**(b) Function as capture register**

When the TABnCCR2 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 TABnCCR2 register if the valid edge of the capture trigger input pin (TIBn2 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TABnCCR2 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIBn2 pin) is detected.

Even if the capture operation and reading the TABnCCR2 register conflict, the correct value of the TABnCCR2 register can be read.

The capture register is cleared by setting the TABnCTL0.TABnCE bit to 0.

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 <sup>Note</sup>   |
| One-shot pulse output         | Compare register         | Anytime write                 |
| PWM output                    | Compare register         | Batch write <sup>Note</sup>   |
| Free-running timer            | Capture/compare register | Anytime write                 |
| Pulse width measurement       | Capture register         | None                          |

**Note** Writing to the TABnCCR1 register is the trigger.

**Remark** For anytime write and batch write, see 7.6 (2) **Anytime write and batch write**.

**(10) TABn capture/compare register 3 (TABnCCR3)**

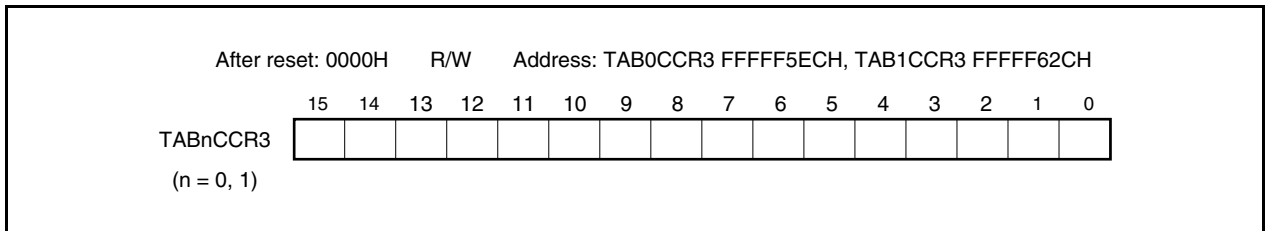
The TABnCCR3 register, which consists of 16 bits, can be used as a capture register or a compare register depending on the mode.

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 TABnOPT0.TABnCCS3 bit. In the pulse width measurement mode, the TABnCCR3 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TABnCCR3 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.



**(a) Function as compare register**

The TABnCCR3 register can be rewritten even when the TABnCTL0.TABnCE bit = 1.

The set value of the TABnCCR3 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 (INTTBnCC3) is generated. If TOBn3 pin output is enabled at this time, the output of the TOBn3 pin is inverted.

The compare register is not cleared by setting the TABnCTL0.TABnCE bit to 0.

**(b) Function as capture register**

When the TABnCCR3 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 TABnCCR3 register if the valid edge of the capture trigger input pin (TIBn3 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TABnCCR3 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIBn3 pin) is detected.

Even if the capture operation and reading the TABnCCR3 register conflict, the correct value of the TABnCCR3 register can be read.

The capture register is cleared by setting the TABnCTL0.TABnCE bit to 0.

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 <sup>Note</sup>   |
| One-shot pulse output         | Compare register         | Anytime write                 |
| PWM output                    | Compare register         | Batch write <sup>Note</sup>   |
| Free-running timer            | Capture/compare register | Anytime write                 |
| Pulse width measurement       | Capture register         | None                          |

**Note** Writing to the TABnCCR1 register is the trigger.

**Remark** For anytime write and batch write, see 7.6 (2) **Anytime write and batch write**.



**(11) TABn counter read buffer register (TABnCNT)**

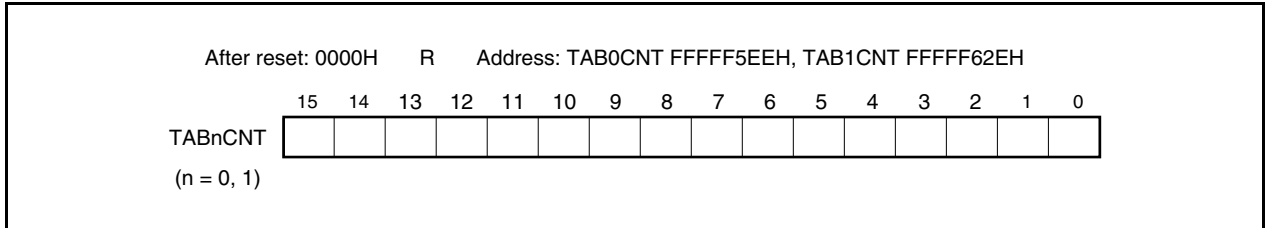
The TABnCNT register is a read buffer register that can read the count value of the 16-bit counter.

If this register is read when the TABnCTL0.TABnCE 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 TABnCNT register is set to 0000H when the TABnCE bit = 0. If the TABnCNT 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 TABnCNT register is set to 0000H after reset, and the TABnCE bit is cleared to 0.



## 7.5 Timer Output Operations

The following table shows the operations and output levels of the TOBn0 to TOBn3 pins.

**Table 7-6. Timer Output Control in Each Mode**

| Operation Mode                     | TOBn0 Pin                                       | TOBn1 Pin                     | TOBn2 Pin                     | TOBn3 Pin                     |
|------------------------------------|---|-------------------------------|-------------------------------|-------------------------------|
| Interval timer mode                | PWM output                                      |                               |                               |                               |
| External event count mode          | None  |                               |                               |                               |
| External trigger pulse output mode | PWM 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            | PWM output (only when compare function is used) |                               |                               |                               |
| Pulse width measurement mode       | None  |                               |                               |                               |

**Remark** n = 0, 1

**Table 7-7. Truth Table of TOBn0 to TOBn3 Pins Under Control of Timer Output Control Bits**

| TABnIOC0.TABnOLa Bit | TABnIOC0.TABnOEa Bit | TABnCTL0.TABnCE bit | Level of TOBna 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, 1  
a = 0 to 3

## 7.6 Operation

TABn can perform the following functions.

**Table 7-8. TABn Specifications in Each Mode**

| Operation                          | TABnCTL1.TABnEST Bit<br>(Software Trigger Bit) | TRGBn Pin<br>(External Trigger Input) | Capture/Compare<br>Register Setting | Compare Register<br>Write |
|------------------------------------|--|---------------------------------------|-------------------------------------|---------------------------|
| Interval timer mode                | Invalid  | Invalid                               | Compare only                        | Anytime write             |
| External event count mode          | Invalid  | Invalid                               | Compare only                        | Anytime write             |
| External trigger pulse output mode | Valid  | Valid                                 | Compare only                        | Batch write               |
| One-shot pulse output mode         | 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       | Invalid  | Invalid                               | Capture only                        | Not applicable            |

**Remarks 1.** TABn has a function to execute tuning with TAA<sub>n</sub>. For details, see **CHAPTER 10 MOTOR CONTROL FUNCTION.**

**2.** n = 0, 1

**(1) Counter basic operation**

This section explains the basic operation of the 16-bit counter. For details, refer to the description of the operation in each mode.

**Remark** n = 0, 1  
a = 0 to 3

**(a) Counter start operation**

- In external event count mode  
When the TABnCTL0.TABnCE bit is set from 0 to 1, the 16-bit counter is set to 0000H.  
After that, it counts up to 0001H, 0002H, 0003H, ... each time the valid edge of external event count input (EVTBn) is detected.
- In modes other than the above  
Starts counting from the default value FFFFH in all modes.  
It counts up from FFFFH to 0000H, 0001H, 0002H, 0003H, and so on.

**(b) Clear operation**

The 16-bit counter is cleared to 0000H when its value matches the value of the compare register and when its value is captured. The count operation from FFFFH to 0000H that takes place immediately after the counter has started counting or when the counter overflows is not a clearing operation. Therefore, the INTTBnCCa interrupt signal is not generated.

**(c) Overflow operation**

The 16-bit counter overflows when the counter counts up from FFFFH to 0000H in the free-running mode or pulse width measurement mode. If the counter overflows, the TABnOPT0.TABnOVF bit is set to 1 and an interrupt request signal (INTTBnOV) is generated. Note that the INTTBnOV signal is not generated under the following conditions.

- Immediately after a count operation has been started
- If the counter value matches the compare value FFFFH and is cleared
- When FFFFH is captured in the pulse width measurement mode and the counter counts up from FFFFH to 0000H

**Caution** After the overflow interrupt request signal (INTTBnOV) has been generated, be sure to check that the overflow flag (TABnOVF bit) is set to 1.

**(d) Counter read operation during count operation**

The value of the 16-bit counter of TABn can be read by using the TABnCNT register during the count operation.

When the TABnCTL0.TABnCE bit = 1, the value of the 16-bit counter can be read by reading the TABnCNT register. When the TABnCE bit = 0, the 16-bit counter is FFFFH and the TABnCNT register is 0000H.

**(e) Interrupt operation**

TABn generates the following five interrupt request signals.

- INTTBnCC0 interrupt: This signal functions as a match interrupt request signal of the CCR0 buffer register and as a capture interrupt request signal to the TABnCCR0 register.
- INTTBnCC1 interrupt: This signal functions as a match interrupt request signal of the CCR1 buffer register and as a capture interrupt request signal to the TABnCCR1 register.

- INTTBnCC2 interrupt: This signal functions as a match interrupt request signal of the CCR2 buffer register and as a capture interrupt request signal to the TABnCCR2 register.
- INTTBnCC3 interrupt: This signal functions as a match interrupt request signal of the CCR3 buffer register and as a capture interrupt request signal to the TABnCCR3 register.
- INTTBnOV interrupt: This signal functions as an overflow interrupt request signal.

**(2) Anytime write and batch write**

The TABnCCR0 to TABnCCR3 registers can be rewritten in the TABn during timer operation (TABnCTL0.TABnCE bit = 1), but the write method (anytime write, batch write) of the CCR0 to CCR3 buffer registers differs depending on the mode.

**(a) Anytime write**

In this mode, data is transferred at any time from the TABnCCR0 to TABnCCR3 registers to the CCR0 to CCR3 buffer registers during the timer operation.

**Figure 7-2. Flowchart of Basic Operation for Anytime Write**

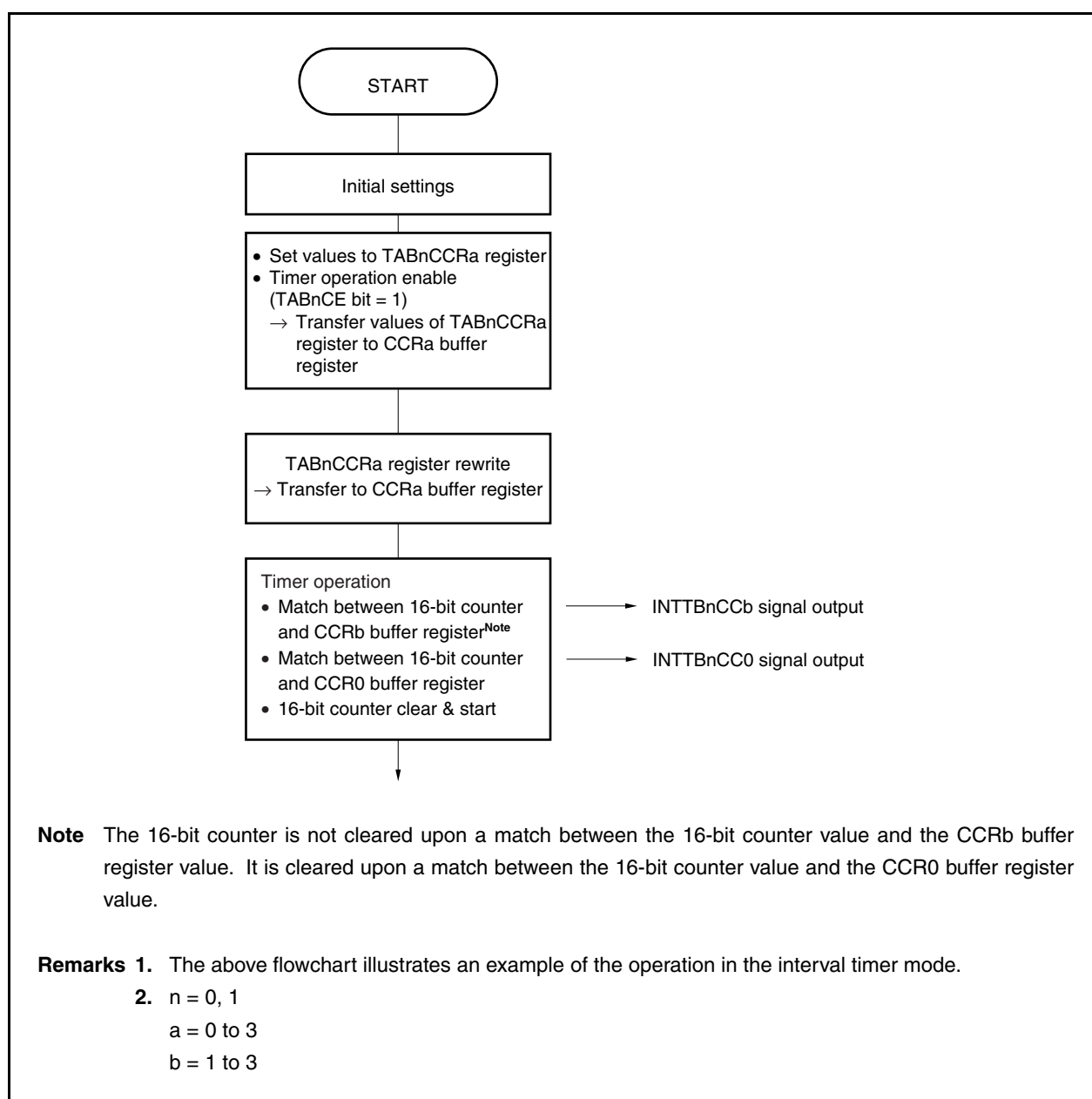
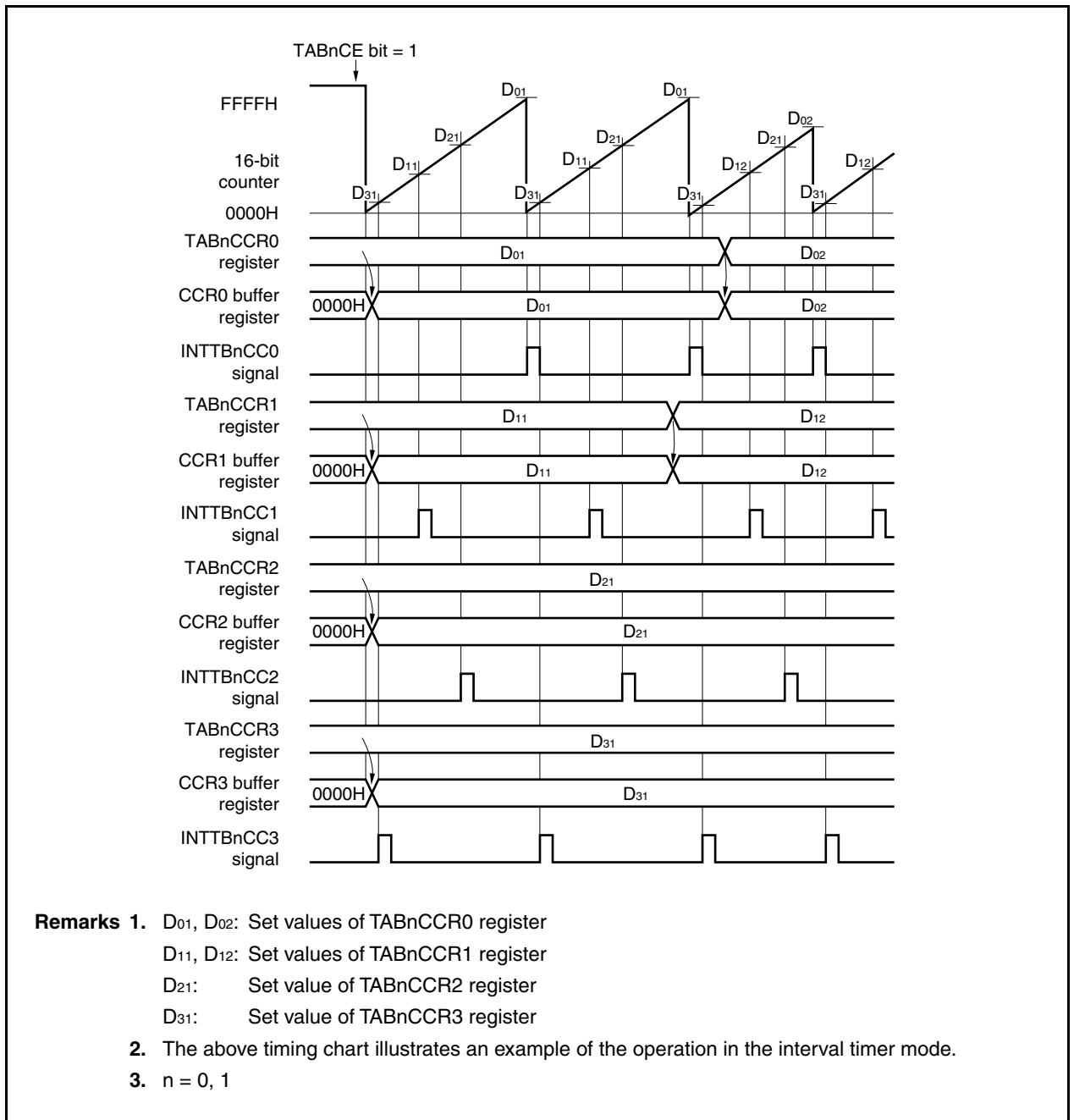


Figure 7-3. Timing of Anytime Write



**(b) Batch write**

In this mode, data is transferred all at once from the TABnCCR0 to TABnCCR3 registers to the CCR0 to CCR3 buffer registers during timer operation. This data is transferred upon a match between the value of the CCR0 buffer register and the value of the 16-bit counter. Transfer is enabled by writing to the TABnCCR1 register.

Whether to enable or disable the next transfer timing is controlled by writing or not writing to the TABnCCR1 register.

In order for the set value when the TABnCCR0 to TABnCCR3 registers are rewritten to become the 16-bit counter comparison value (in other words, in order for this value to be transferred to the CCR0 to CCR3 buffer registers), it is necessary to rewrite TABnCCR0 and finally write to the TABnCCR1 register before the 16-bit counter value and the CCR0 buffer register value match. The values of the TABnCCR0 to TABnCCR3 registers are transferred to the CCR0 to CCR3 buffer registers upon a match between the count value of the 16-bit counter and the value of the CCR0 buffer register. Thus, even when wishing only to rewrite the value of the TABnCCR0, TABnCCR2, or TABnCCR3 register, also write the same value (same as preset value of the TABnCCR1 register) to the TABnCCR1 register.



Figure 7-4. Flowchart of Basic Operation for Batch Write

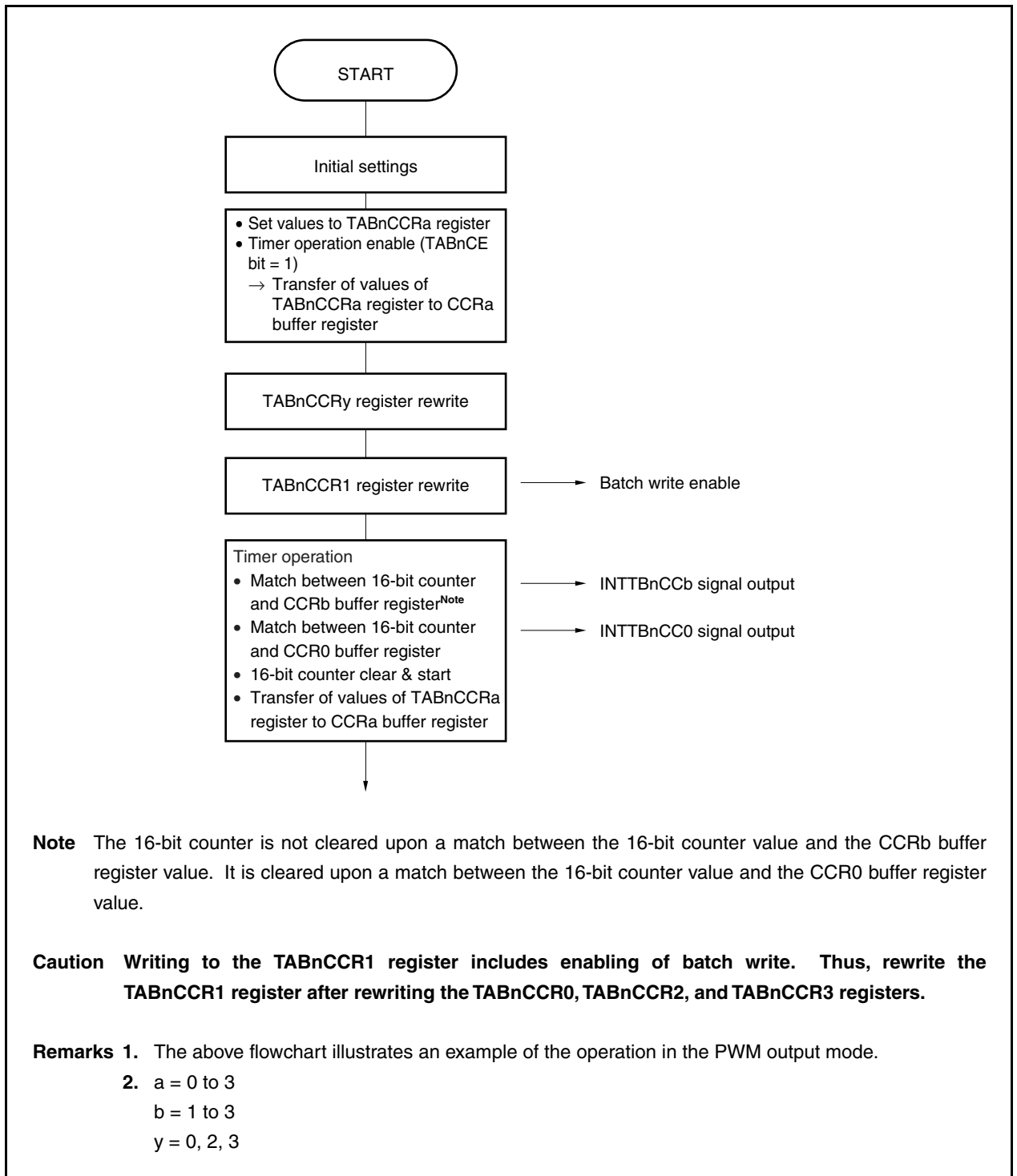
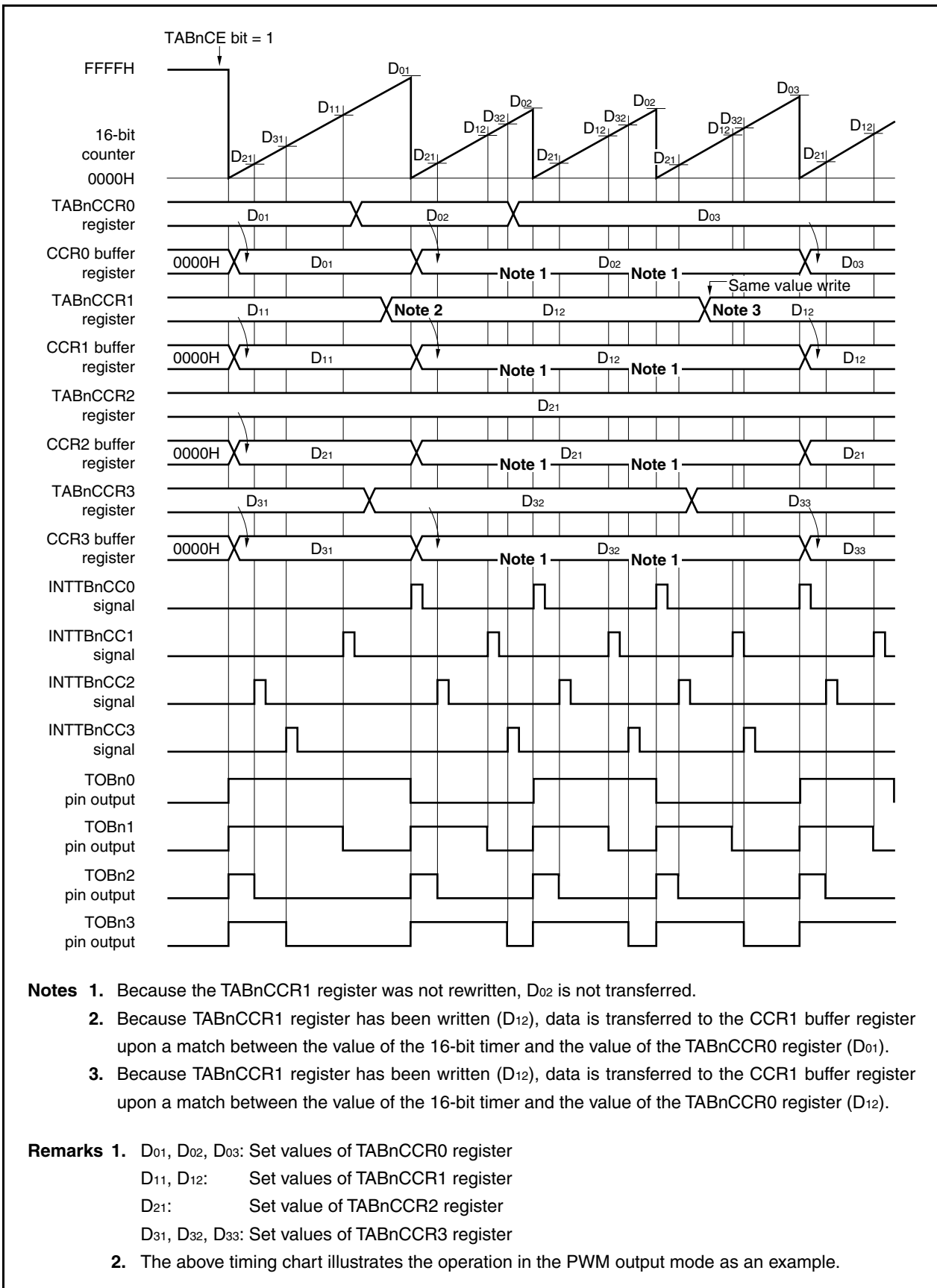


Figure 7-5. Timing of Batch Write



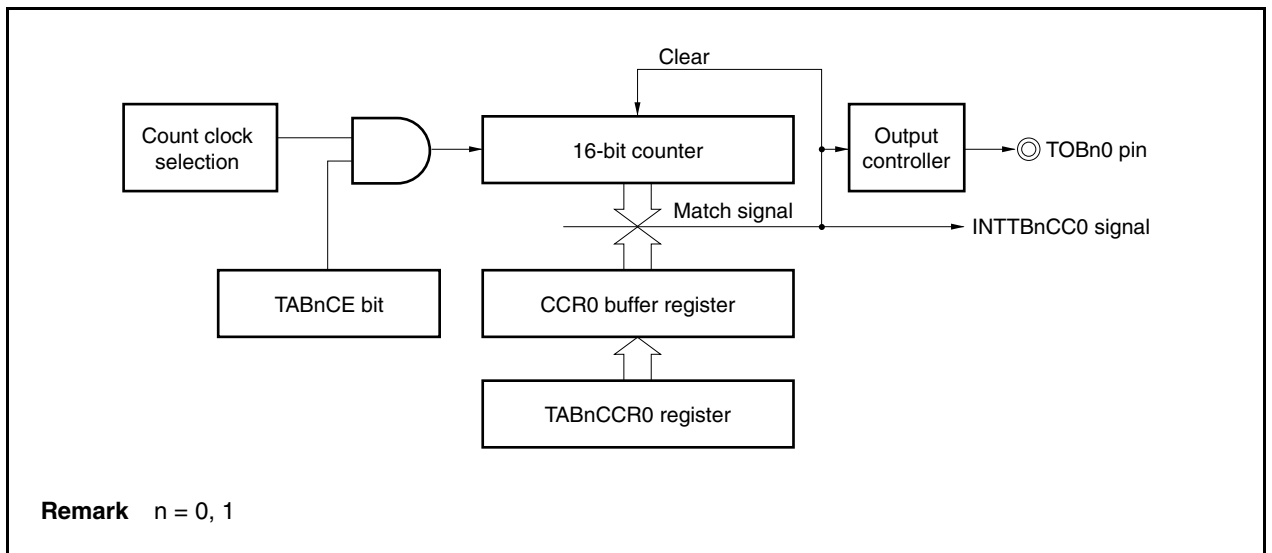
**7.6.1 Interval timer mode (TABnMD2 to TABnMD0 bits = 000)**

In the interval timer mode, an interrupt request signal (INTTBnCC0) is generated at the interval set by the TABnCCR0 register if the TABnCTL0.TABnCE bit is set to 1. A PWM waveform with a duty factor of 50% whose half cycle is equal to the interval can be output from the TOBn0 pin.

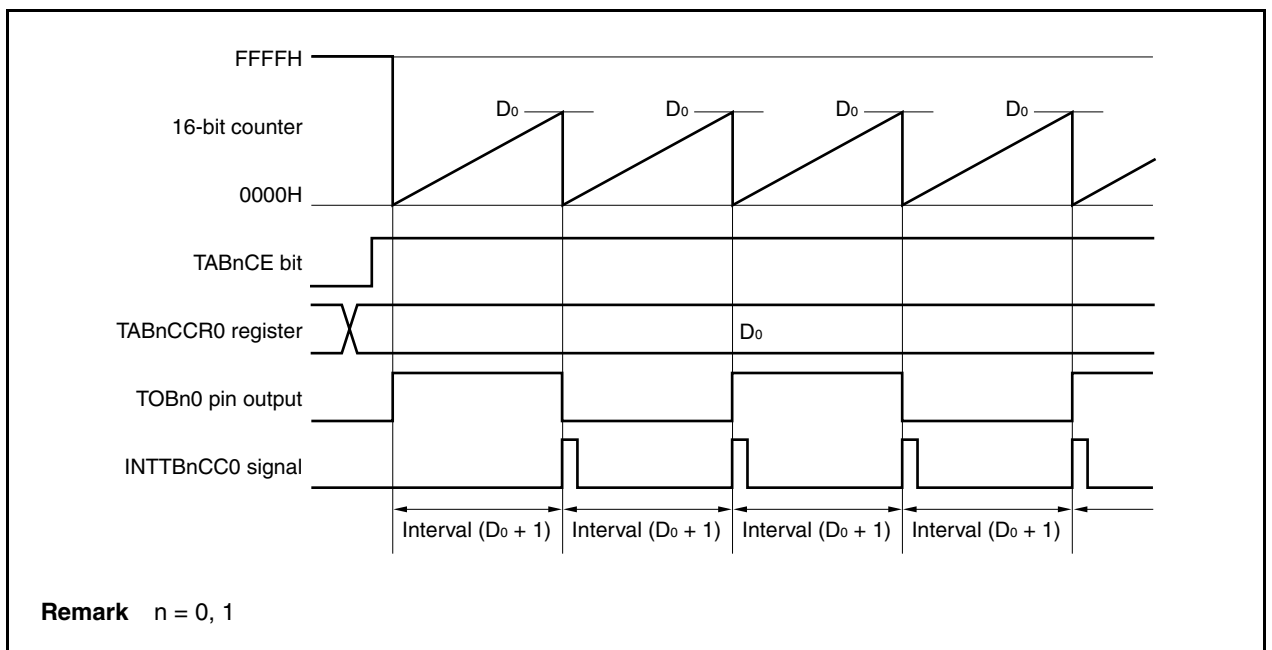
The TABnCCR1 to TABnCCR3 registers are not used in the interval timer mode. However, the set value of the TABnCCR1 to TABnCCR3 registers is transferred to the CCR1 to CCR3 buffer registers and, when the count value of the 16-bit counter matches the value of the CCR1 to CCR3 buffer registers, compare match interrupt request signals (INTTBnCC1 to INTTBnCC3) are generated. In addition, a PWM waveform with a duty factor of 50%, which is inverted when the INTTBnCC1 to INTTBnCC3 signals are generated, can be output from the TOBn1 to TOBn3 pins.

The value of the TABnCCR1 to TABnCCR3 registers can be rewritten even while the timer is operating.

**Figure 7-6. Interval Timer Configuration**



**Figure 7-7. Basic Timing of Operation in Interval Timer Mode**



When the TABnCE 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 TOBn0 pin is inverted. Additionally, the set value of the TABnCCR0 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 TOBn0 pin is inverted, and a compare match interrupt request signal (INTTBnCC0) is generated.

The interval can be calculated by the following expression.

$$\text{Interval} = (\text{Set value of TABnCCR0 register} + 1) \times \text{Count clock cycle}$$

**Remark** n = 0, 1

**Figure 7-8. Register Setting for Interval Timer Mode Operation (1/3)**

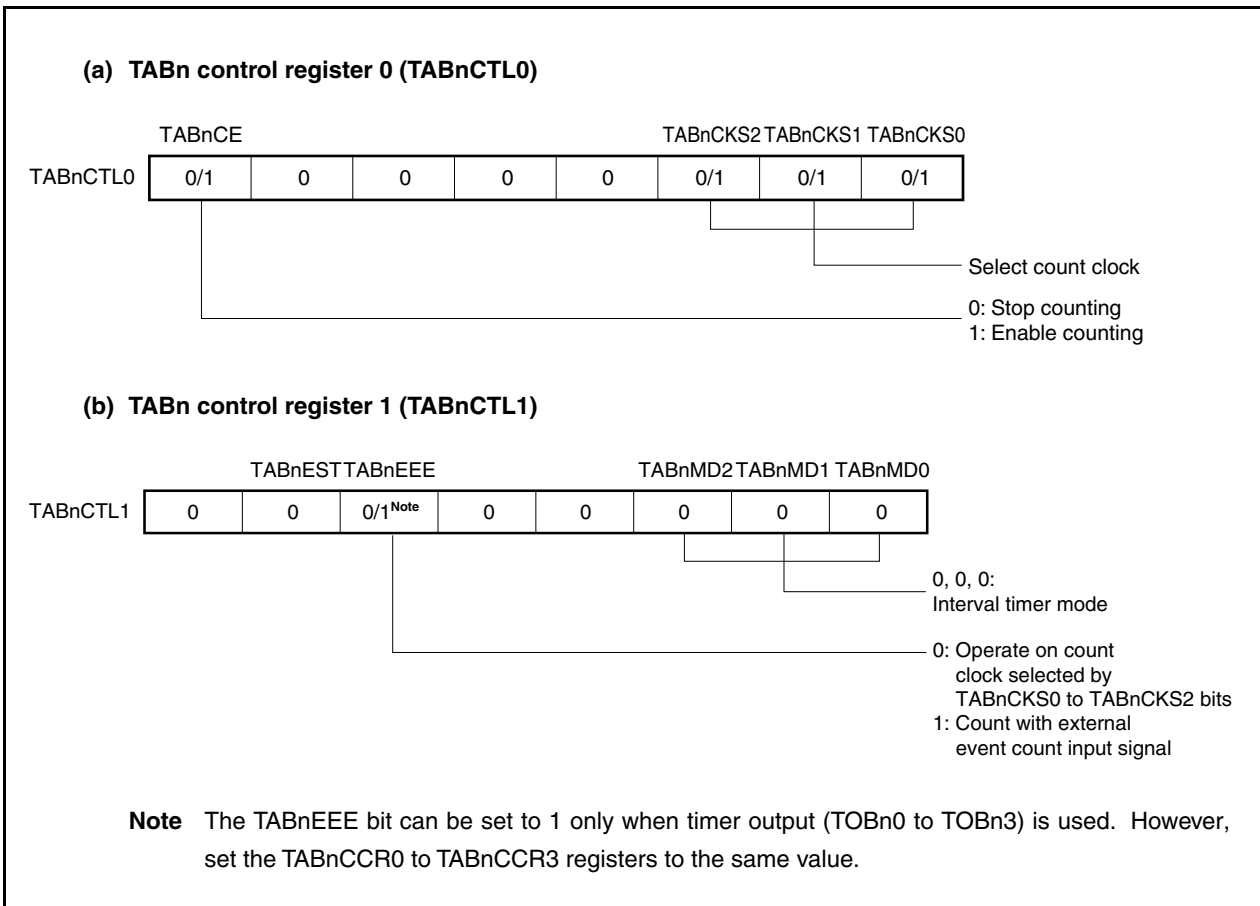


Figure 7-8. Register Setting for Interval Timer Mode Operation (2/3)

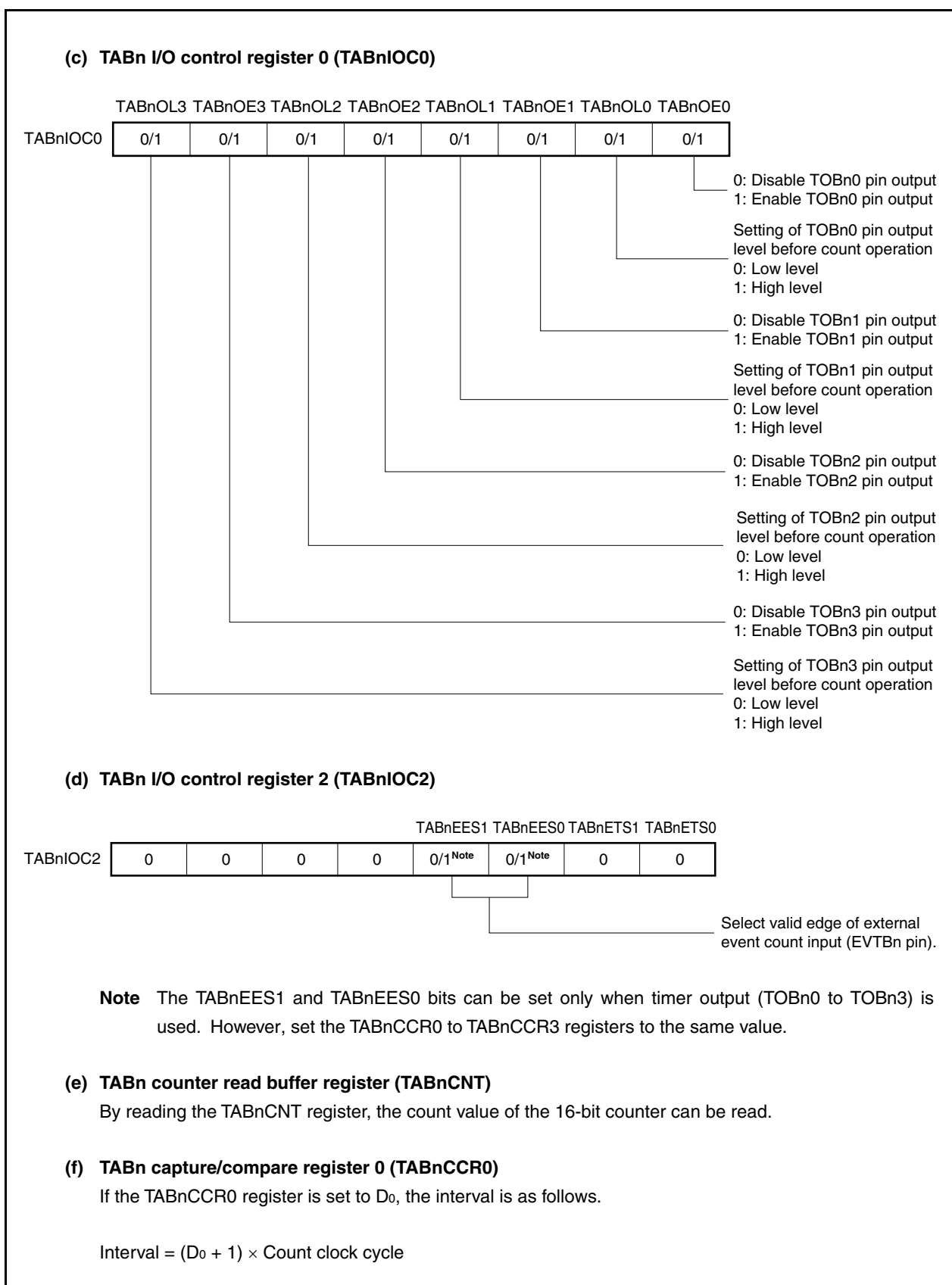


Figure 7-8. Register Setting for Interval Timer Mode Operation (3/3)

**(g) TABn capture/compare registers 1 to 3 (TABnCCR1 to TABnCCR3)**

The TABnCCR1 to TABnCCR3 registers are not used in the interval timer mode. However, the set values of the TABnCCR1 to TABnCCR3 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, the TOBn1 to TOBn3 pin outputs are inverted and a compare match interrupt request signal (INTTBnCC1 to INTTBnCC3) is generated.

When the TABnCCR1 to TABnCCR3 registers are not used, it is recommended to set their values to FFFFH. Also mask the registers by the interrupt mask flags (TABnCCIC1.TABnCCMK1 to TABnCCIC3.TABnCCMK3).

**Remarks** 1. TABn I/O control register 1 (TABnIOC1) and TABn option register 0 (TABnOPT0) are not used in the interval timer mode.

2. n = 0, 1

(1) Interval timer mode operation flow

Figure 7-9. Software Processing Flow in Interval Timer Mode (1/2)

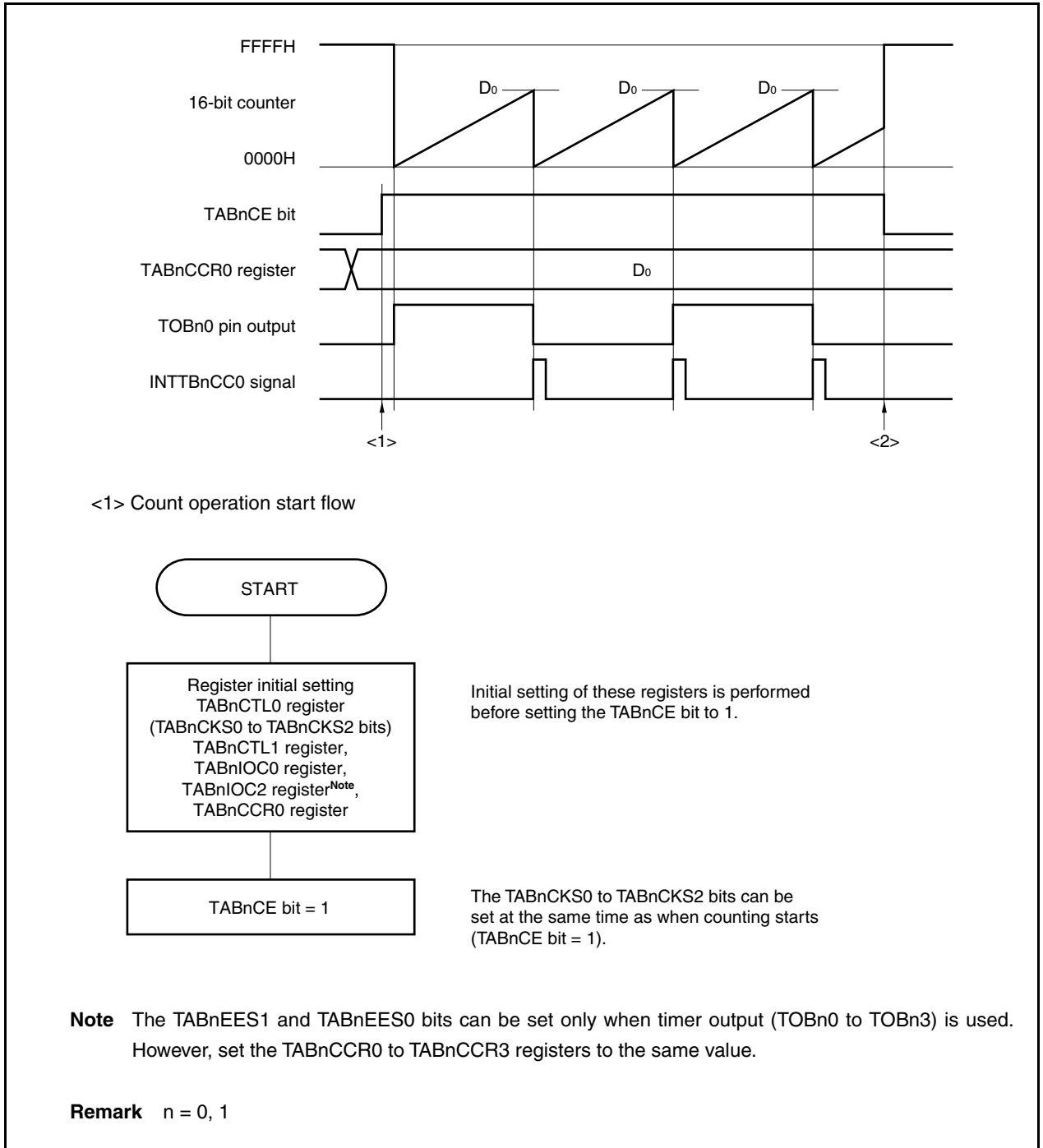
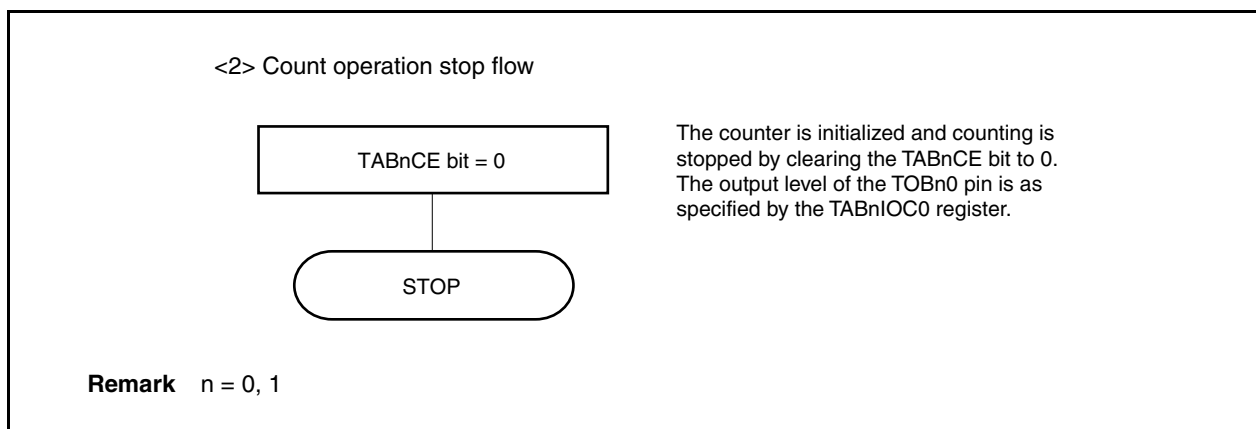


Figure 7-9. Software Processing Flow in Interval Timer Mode (2/2)



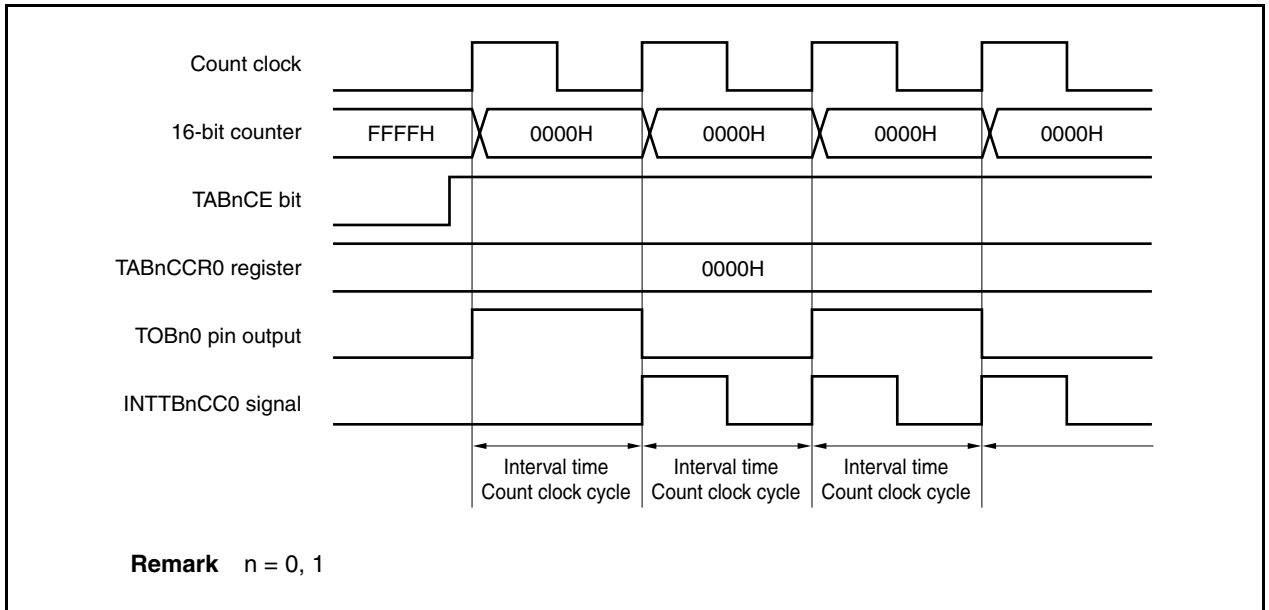


(2) Interval timer mode operation timing

(a) Operation if TABnCCR0 register is set to 0000H

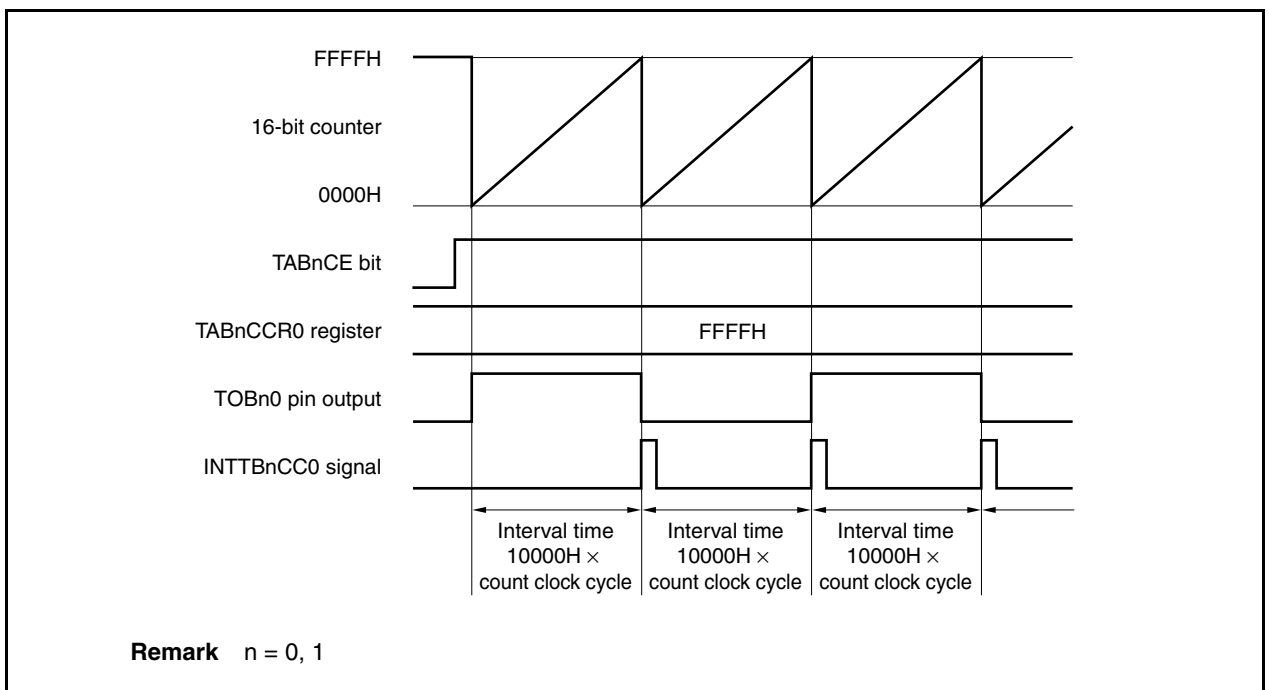
If the TABnCCR0 register is set to 0000H, the INTTBnCC0 signal is generated at each count clock, and the output of the TOBn0 pin is inverted.

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



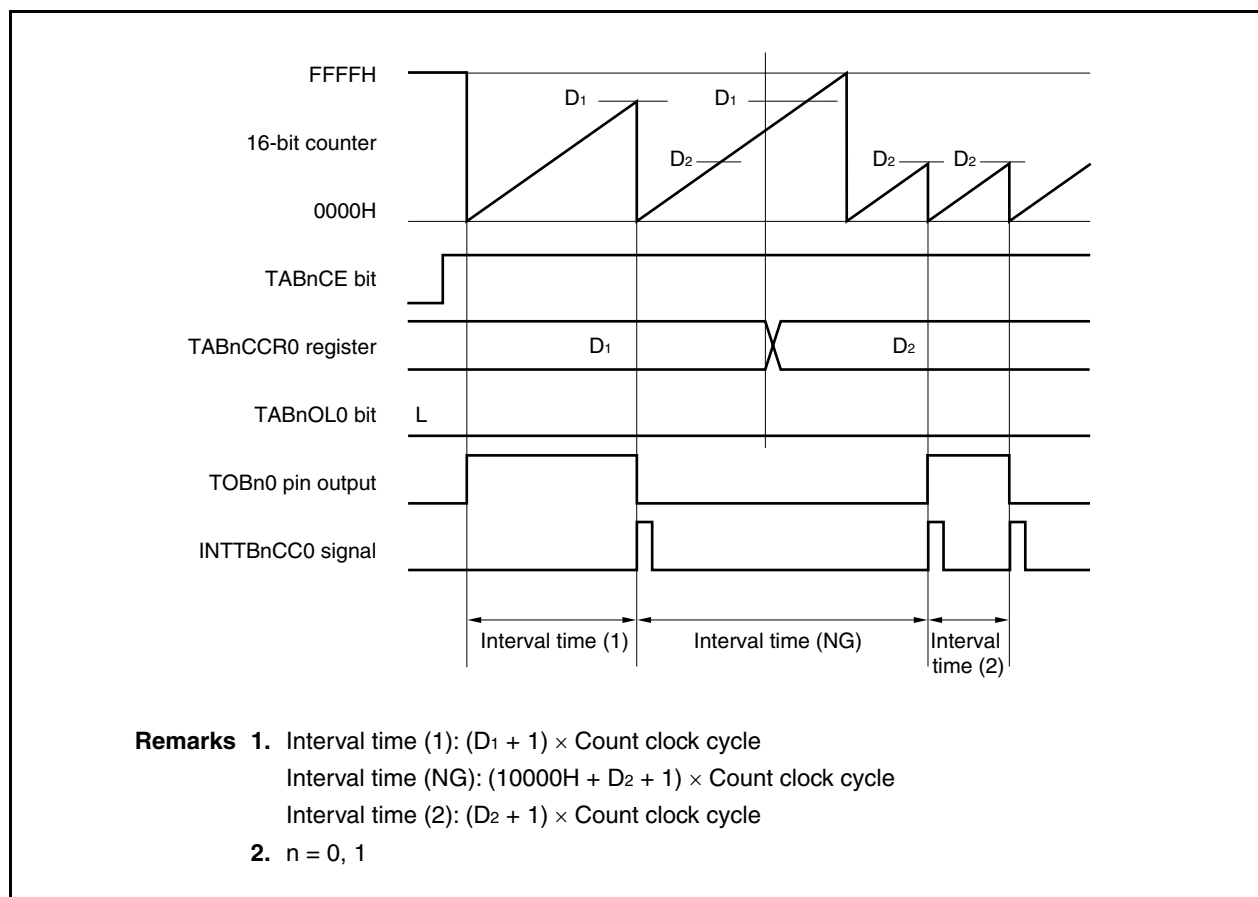
(b) Operation if TABnCCR0 register is set to FFFFH

If the TABnCCR0 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 INTTBnCC0 signal is generated and the output of the TOBn0 pin is inverted. At this time, an overflow interrupt request signal (INTTBnOV) is not generated, nor is the overflow flag (TABnOPT0.TABnOVF bit) set to 1.



**(c) Notes on rewriting TABnCCR0 register**

If the value of the TABnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When the overflow may occur, stop counting once and then change the set value.



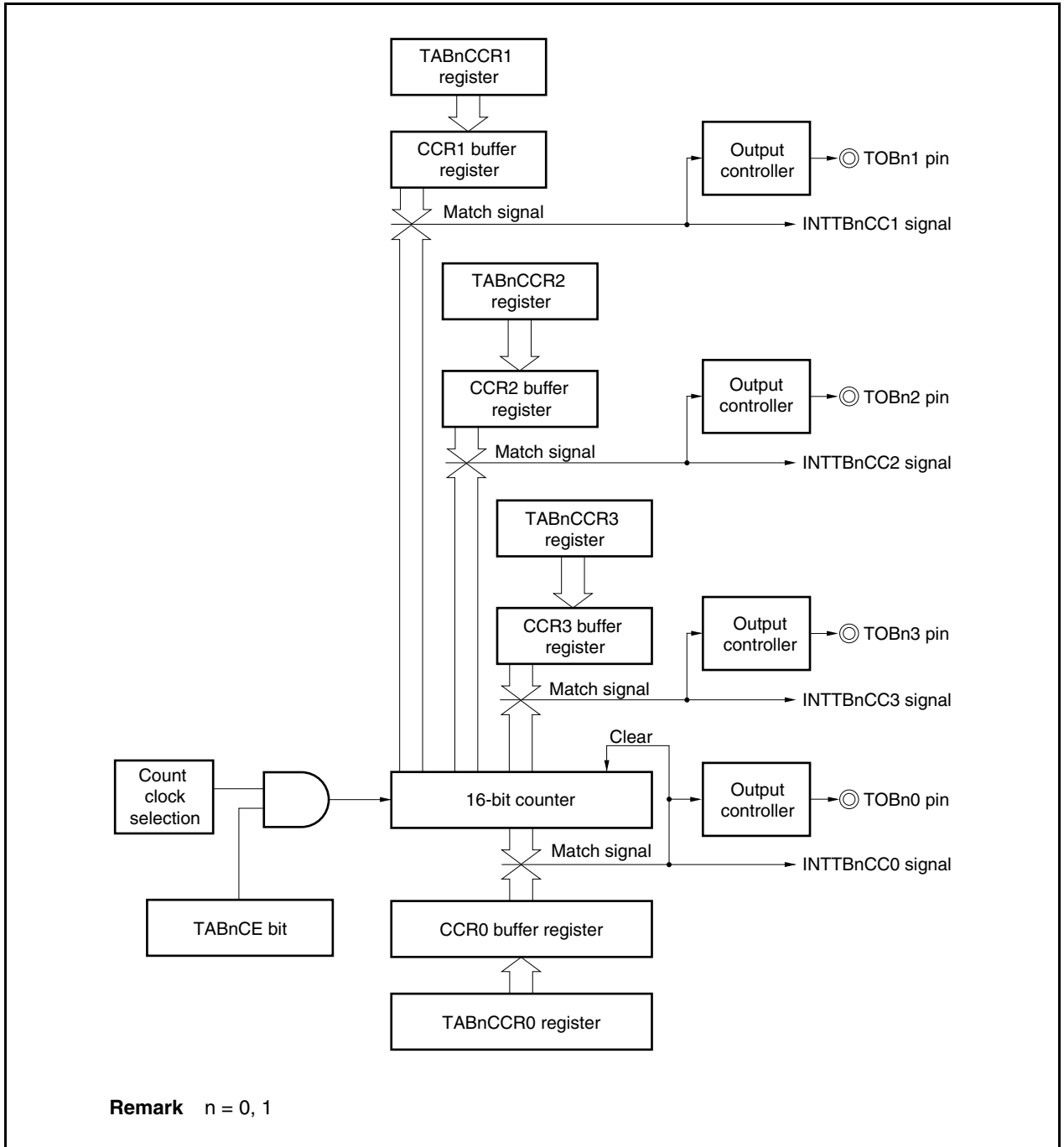
If the value of the TABnCCR0 register is changed from  $D_1$  to  $D_2$  while the count value is greater than  $D_2$  but less than  $D_1$ , the count value is transferred to the CCR0 buffer register as soon as the TABnCCR0 register has been rewritten. Consequently, the value of the 16-bit counter that 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 INTTBnCC0 signal is generated and the output of the TOBn0 pin is inverted.

Therefore, the INTTBnCC0 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 " $(10000\text{H} + D_2 + 1) \times \text{Count clock period}$ ".

(d) Operation of TABnCCR1 to TABnCCR3 registers

Figure 7-10. Configuration of TABnCCR1 to TABnCCR3 Registers

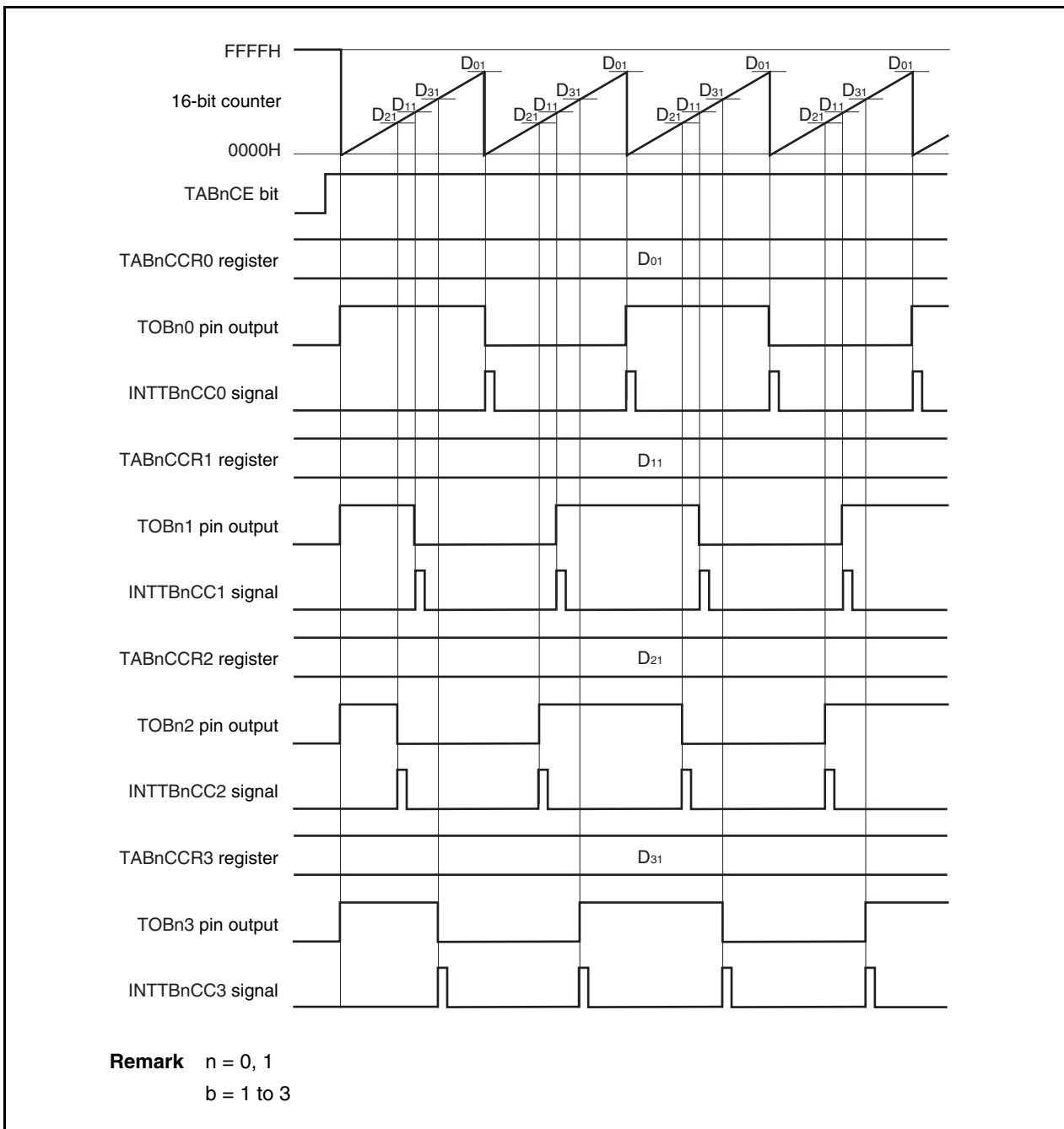


When the TABnCCRb register is set to the same value as the TABnCCR0 register, the INTTBnCCb signal is generated at the same timing as the INTTBnCC0 signal and the TOBnb pin output is inverted. In other words, a PWM waveform with a duty factor of 50% can be output from the TOBnb pin.

The following shows the operation when the TABnCCRb register is set to other than the value set in the TABnCCR0 register.

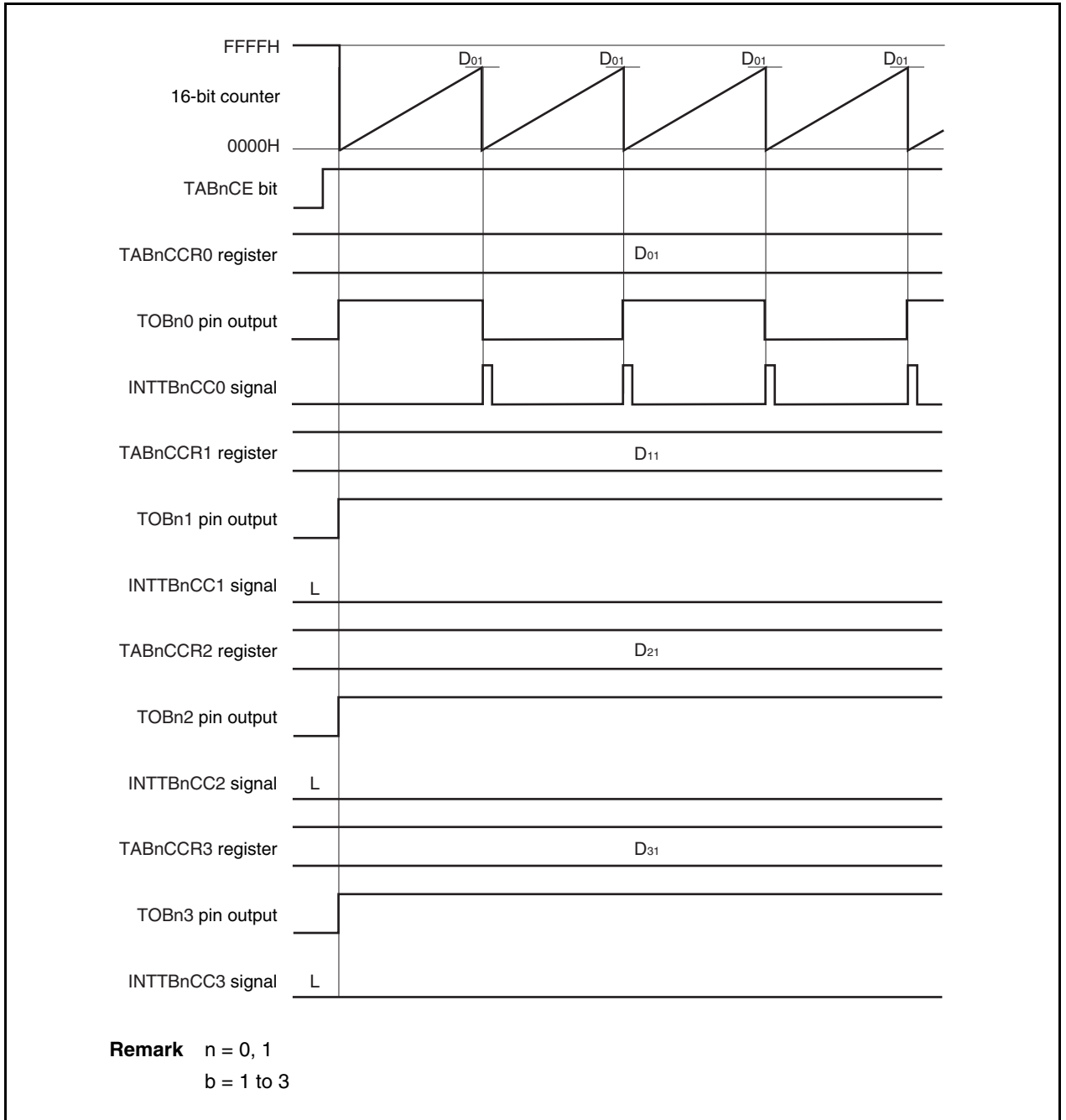
If the set value of the TABnCCRb register is less than the set value of the TABnCCR0 register, the INTTBnCCb signal is generated once per cycle. At the same time, the output of the TOBnb pin is inverted. The TOBnb pin outputs a PWM waveform with a duty factor of 50% after outputting a short-width pulse.

Figure 7-11. Timing Chart When  $D_{01} \geq D_{b1}$



If the set value of the TABnCCRb register is greater than the set value of the TABnCCR0 register, the count value of the 16-bit counter does not match the value of the TABnCCRb register. Consequently, the INTTBnCCb signal is not generated, nor is the output of the TOBnb pin changed. When the TABnCCRb register is not used, it is recommended to set its value to FFFFH.

Figure 7-12. Timing Chart When  $D_{01} < D_{b1}$



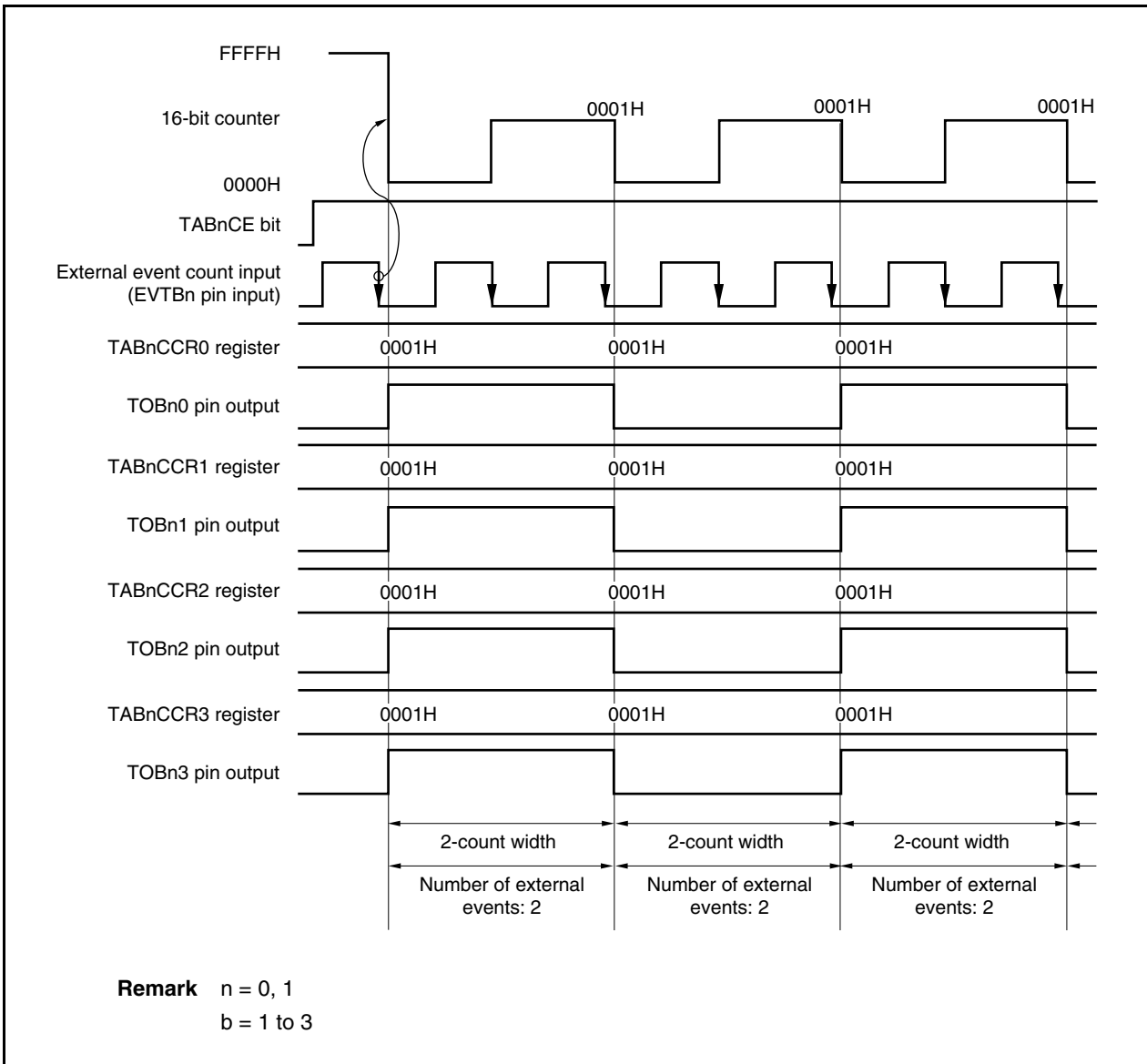
(3) Operation by external event count input (EVTBn)

(a) Operation

To count the 16-bit counter at the valid edge of the external event count input (EVTBn) in the interval timer mode, the 16-bit counter is cleared from FFFFH to 0000H by the valid edge of the external event count input after the TABnCE bit is set from 0 to 1.

When 0001H is set to both the TABnCCR0 and TABnCCRb registers, the output of the TOBn0 and TOBnb pins is inverted each time the 16-bit counter counts twice (b = 1 to 3).

The TABnCTL1.TABnEEE bit can be set to 1 in the interval timer mode only when the timer output (TOBn0, TOBnb) is used with the external event count input.

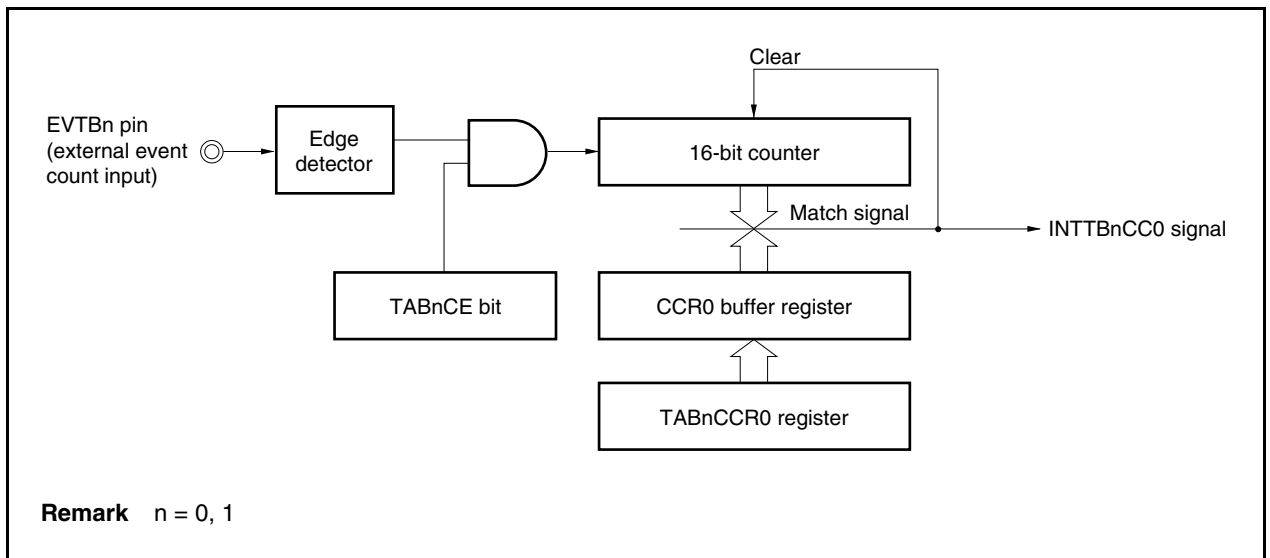


**7.6.2 External event count mode (TABnMD2 to TABnMD0 bits = 001)**

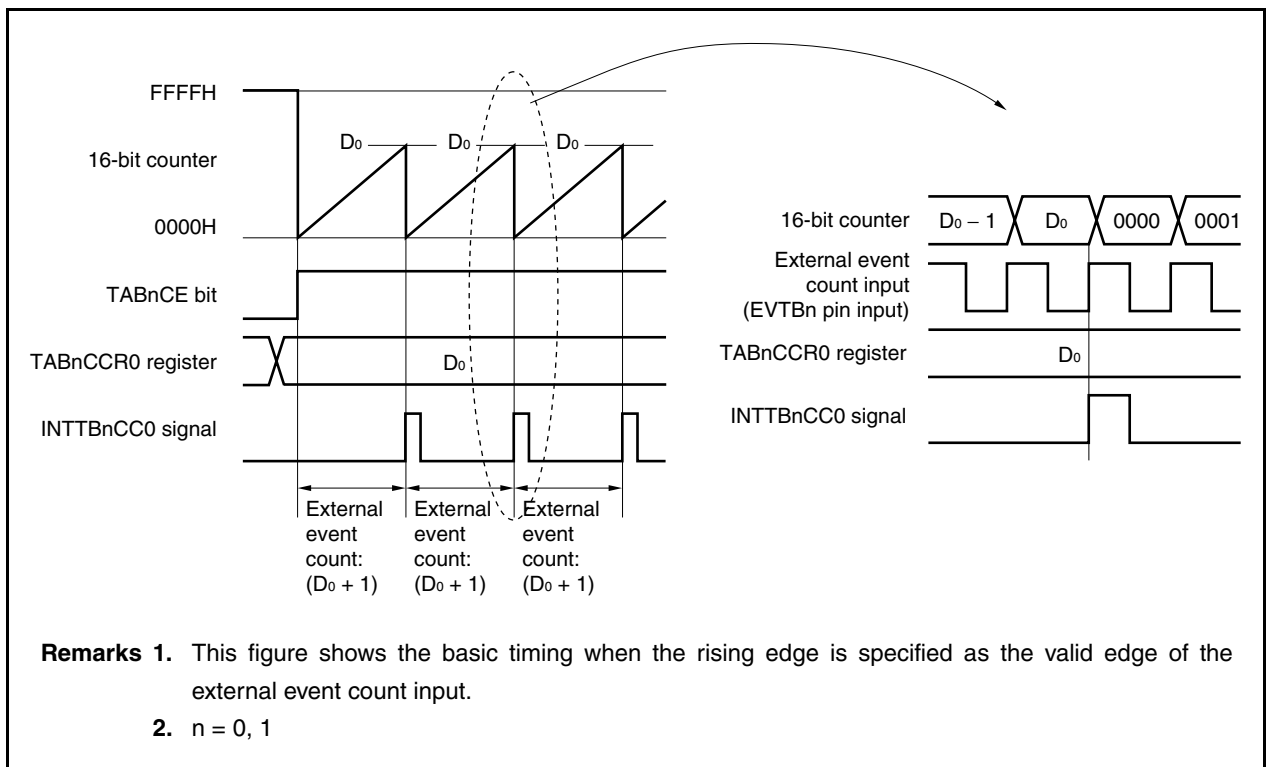
In the external event count mode, the valid edge of the external event count input (EVTBn) is counted when the TABnCTL0.TABnCE bit is set to 1, and an interrupt request signal (INTTBnCC0) is generated each time the specified number of edges set by the TABnCCR0 register have been counted. The TOBn0 to TOBn3 pins cannot be used. When using the TOBn0 to TOBn3 pins for external event count input, set the TABnCTL1.TABnEEE bit to 1 in the interval timer mode (see 7.6.1 (3) **Operation by external event count input (EVTBn)**).

The TABnCCR1 to TABnCCR3 registers are not used in the external event count mode.

**Figure 7-13. Configuration in External Event Count Mode**



**Figure 7-14. Basic Timing in External Event Count Mode**



When the TABnCE 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 TABnCCR0 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 (INTTBnCC0) is generated.

The INTTBnCC0 signal is generated each time the valid edge of the external event count has been detected “value set to TABnCCR0 register + 1” times.

Figure 7-15. Register Setting for Operation in External Event Count Mode (1/2)

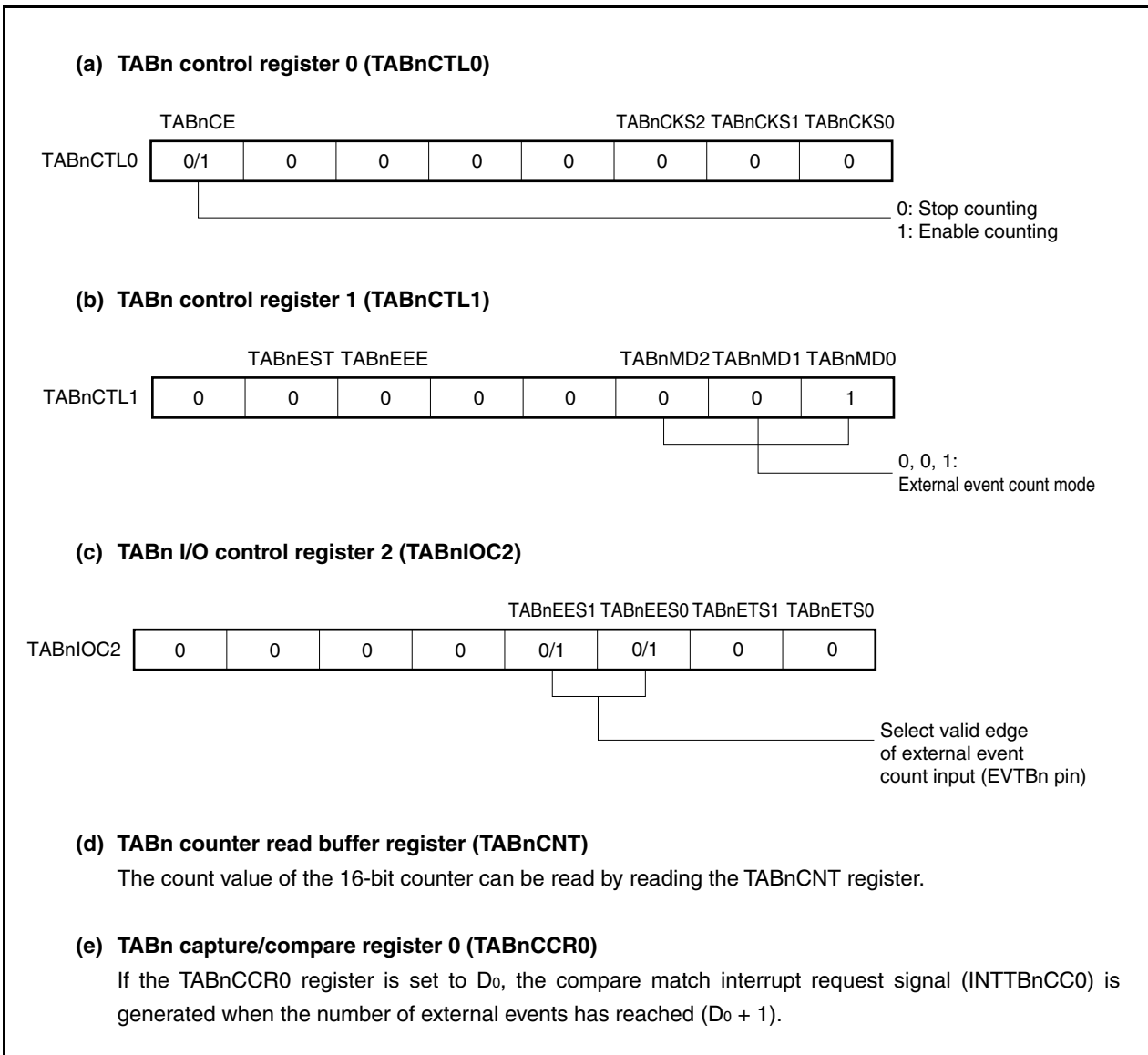




Figure 7-15. Register Setting for Operation in External Event Count Mode (2/2)

**(f) TABn capture/compare registers 1 to 3 (TABnCCR1 to TABnCCR3)**

The TABnCCR1 to TABnCCR3 registers are not used in the external event count mode. However, the set value of the TABnCCR1 to TABnCCR3 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 (INTTBnCC1 to INTTBnCC3) are generated.

When the TABnCCR1 to TABnCCR3 registers are not used, it is recommended to set their values to FFFFH. Also mask the registers by the interrupt mask flags (TABnCCIC1.TABnCCMK1 to TABnCCIC3.TABnCCMK3).

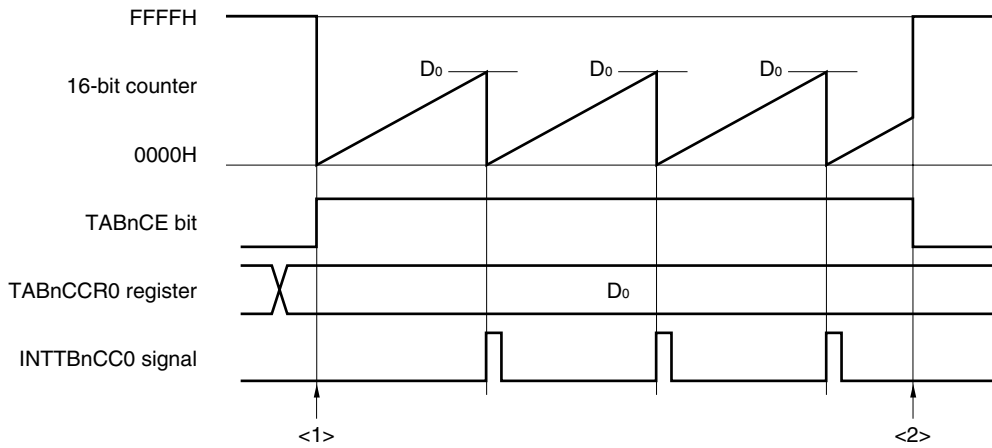
**Caution** Set the TABnIOC0 register to 00H.

**Remarks** 1. TABn I/O control register 1 (TABnIOC1) and TABn option register 0 (TABnOPT0) are not used in the external event count mode.

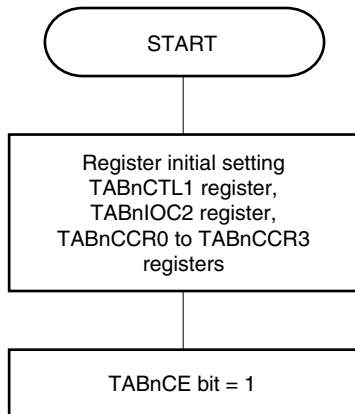
2. n = 0, 1

(1) External event count mode operation flow

Figure 7-16. Software Processing Flow in External Event Count Mode

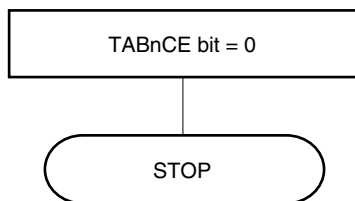


<1> Count operation start flow



Initial setting of these registers is performed before setting the TABnCE bit to 1.

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TABnCE bit to 0.

**Remark** n = 0, 1

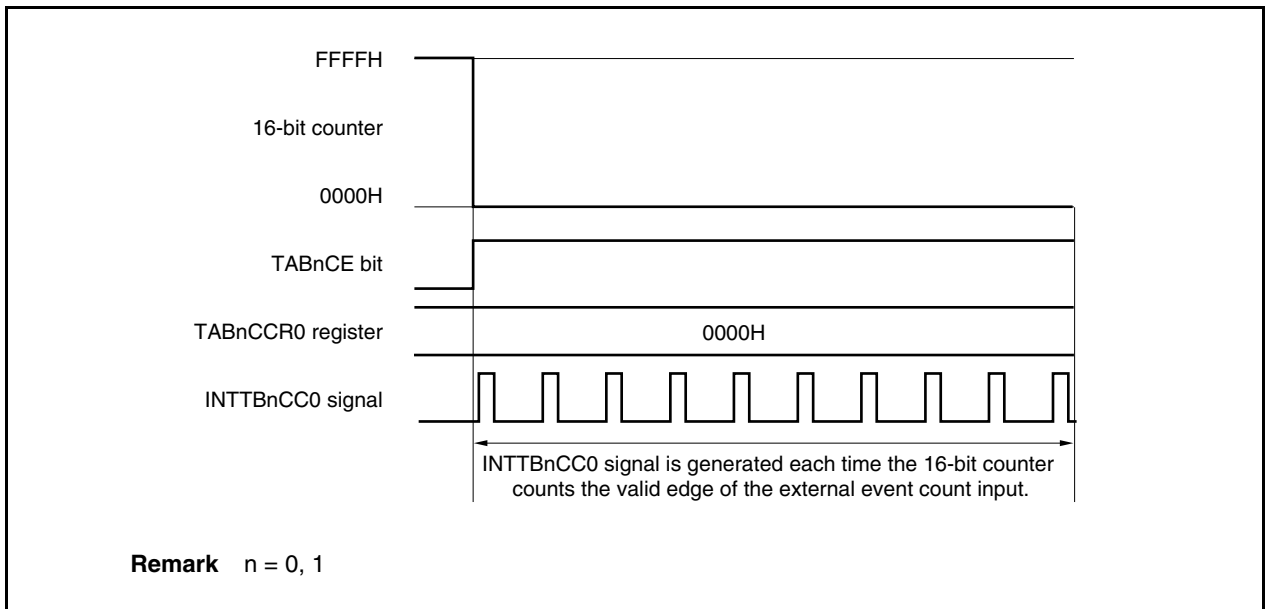
(2) Operation timing in external event count mode

**Caution** In the external event count mode, use of the timer output (TOBn0 to TOBn3) is disabled. If using timer output (TOBn0 to TOBn3) with external event count input (EVTBn), set the interval timer mode, and select the operation enabled by the external event count input for the count clock (TABnCTL1.TABnEEE bit = 1) (see 7.6.1 (3) Operation by external event count input (EVTBn)).

(a) Operation if TABnCCR0 register is set to 0000H

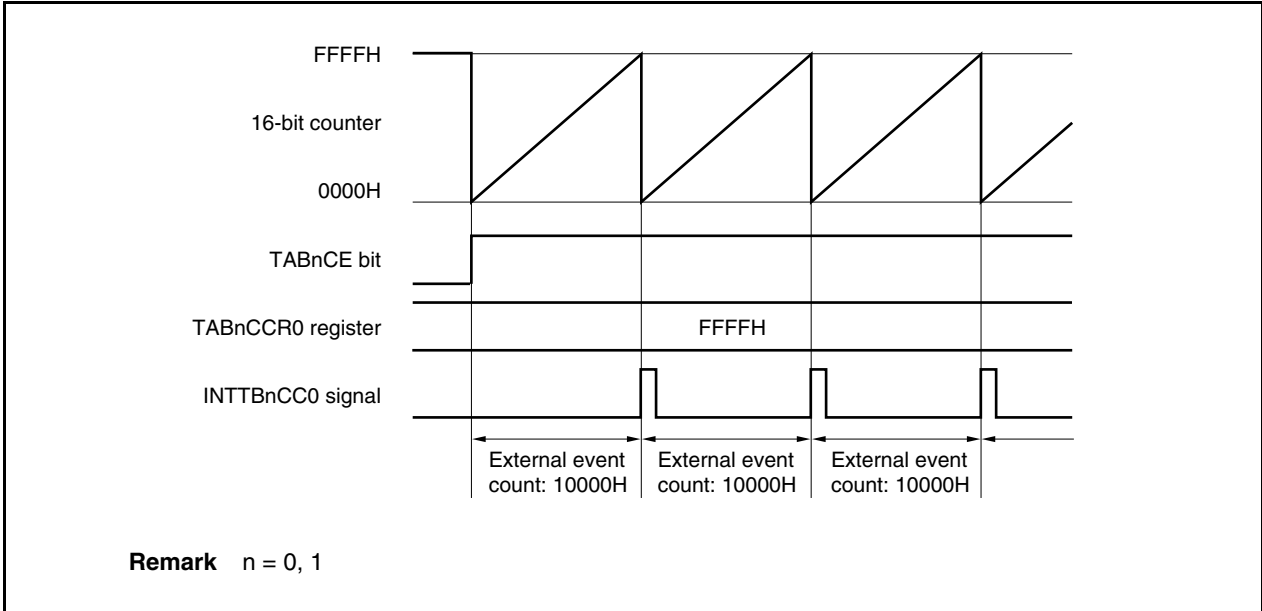
When the TABnCCR0 register is set to 0000H, the 16-bit counter is repeatedly cleared to 0000H and generates an INTTBnCC0 signal each time it has detected the valid edge of the external event count signal and its value has matched that of the CCR0 buffer register.

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



**(b) Operation if TABnCCR0 register is set to FFFFH**

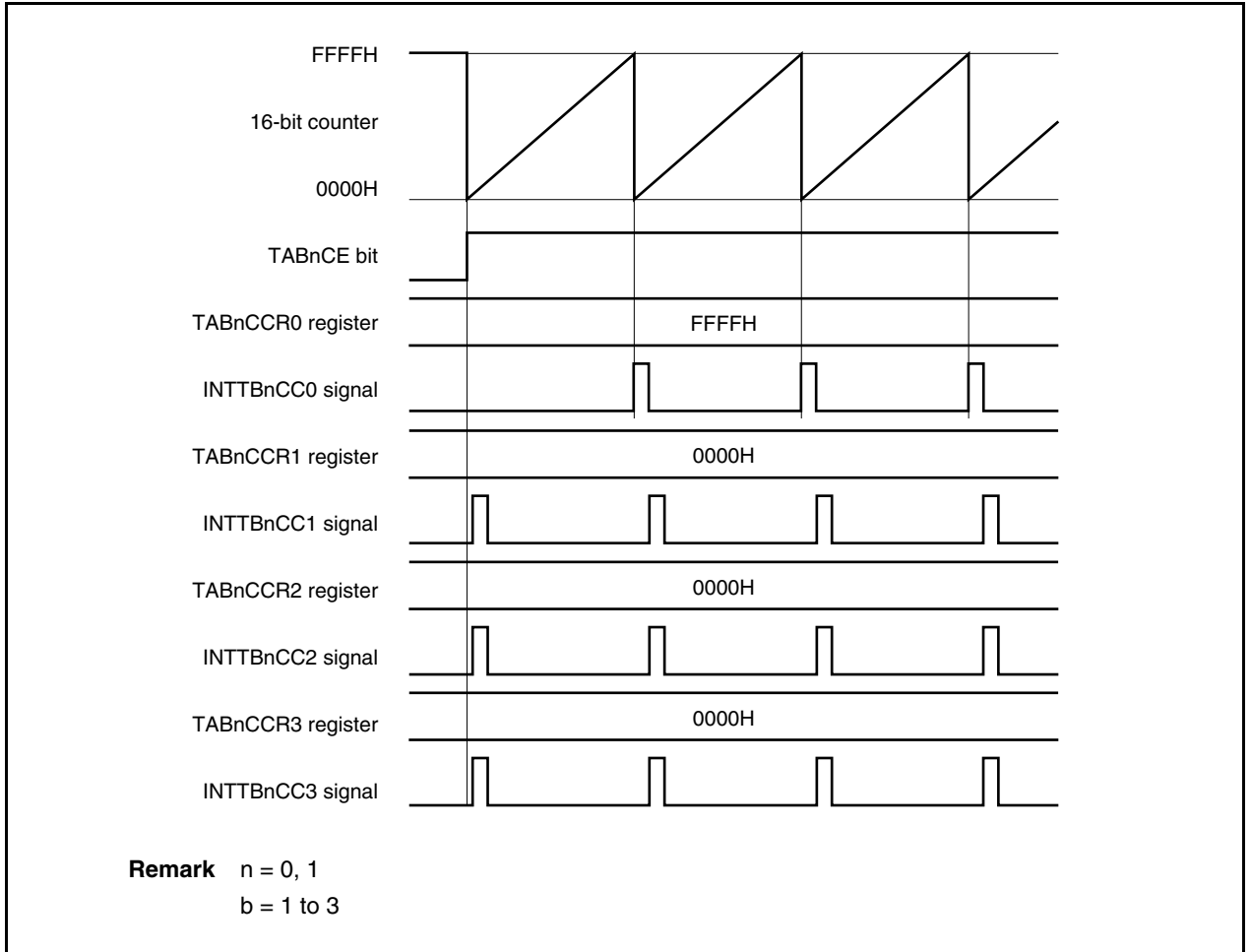
If the TABnCCR0 register is set to FFFFH, the 16-bit counter counts to FFFFH each time the valid edge of the external event count signal has been detected. The 16-bit counter is cleared to 0000H in synchronization with the next count-up timing, and the INTTBnCC0 signal is generated. At this time, the TABnOPT0.TABnOVF bit is not set.



**(c) Operation with TABnCCR0 set to FFFFH and TABnCCRb register to 0000H**

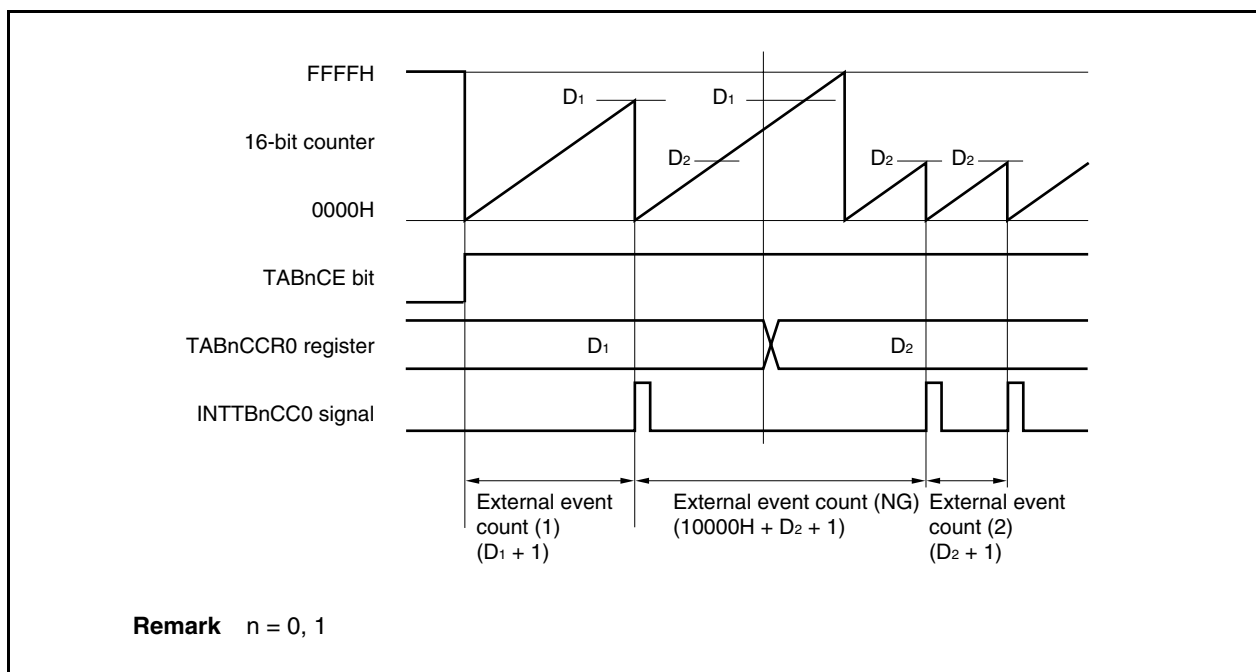
When the TABnCCR0 register is set to FFFFH, the 16-bit counter counts to FFFFH each time it has detected the valid edge of the external event count signal. The counter is then cleared to 0000H in synchronization with the next count-up timing and the INTTBnCC0 signal is generated. At this time, the TABnOPT0.TABnOVF bit is not set.

If the TABnCCRb register is set to 0000H, the INTTBnCCb signal is generated when the 16-bit counter is cleared to 0000H.



**(d) Notes on rewriting the TABnCCR0 register**

If the value of the TABnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When the overflow may occur, stop counting once and then change the set value.



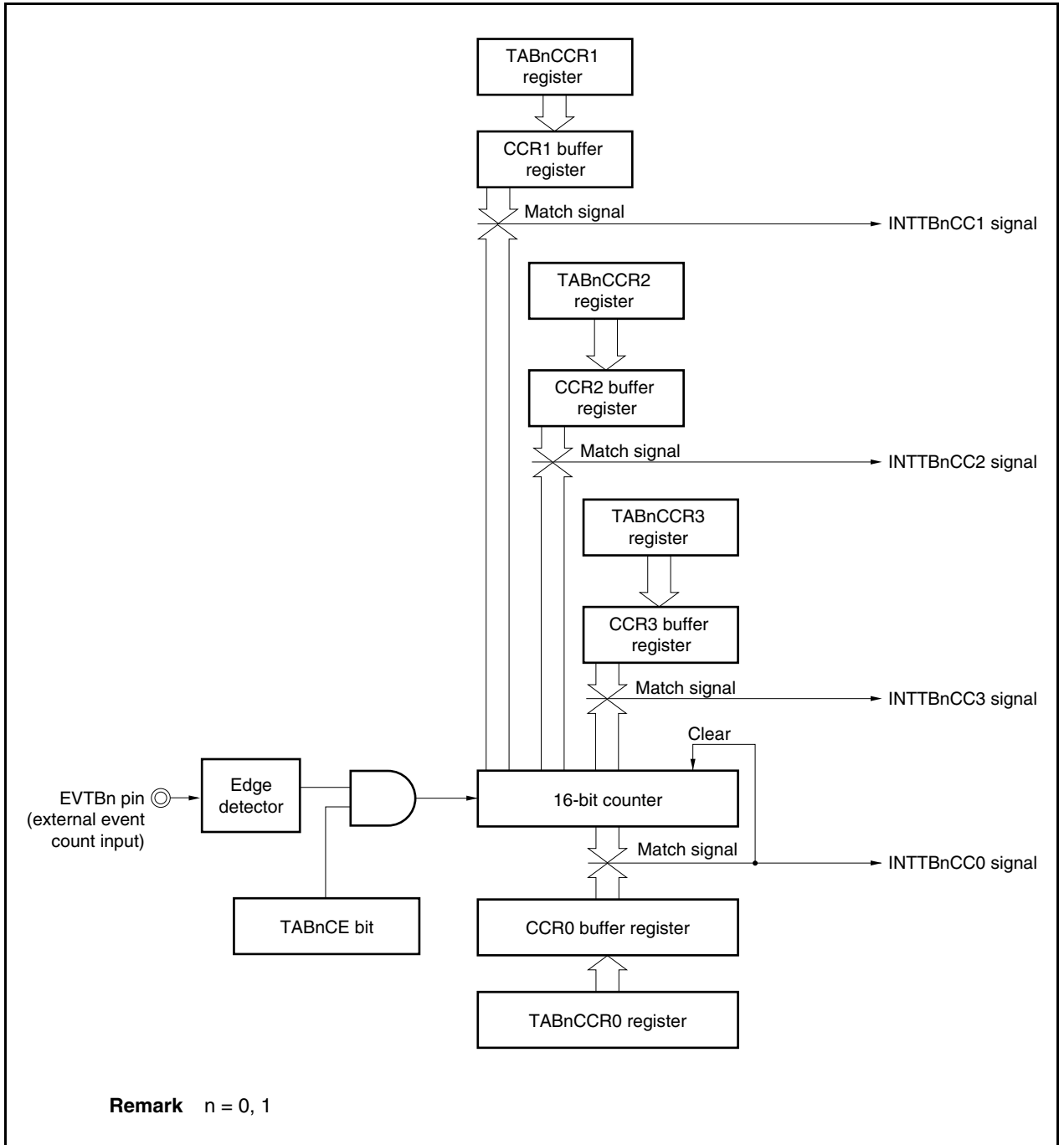
If the value of the TABnCCR0 register is changed from D<sub>1</sub> to D<sub>2</sub> while the count value is greater than D<sub>2</sub> but less than D<sub>1</sub>, the count value is transferred to the CCR0 buffer register as soon as the TABnCCR0 register has been rewritten. Consequently, the value that is compared with the 16-bit counter is D<sub>2</sub>.

Because the count value has already exceeded D<sub>2</sub>, however, the 16-bit counter counts up to FFFFH, overflows, and then counts up again from 0000H. When the count value matches D<sub>2</sub>, the INTTBnCC0 signal is generated.

Therefore, the INTTBnCC0 signal may not be generated at the valid edge count of “(D<sub>1</sub> + 1) times” or “(D<sub>2</sub> + 1) times” originally expected, but may be generated at the valid edge count of “(10000H + D<sub>2</sub> + 1) times”.

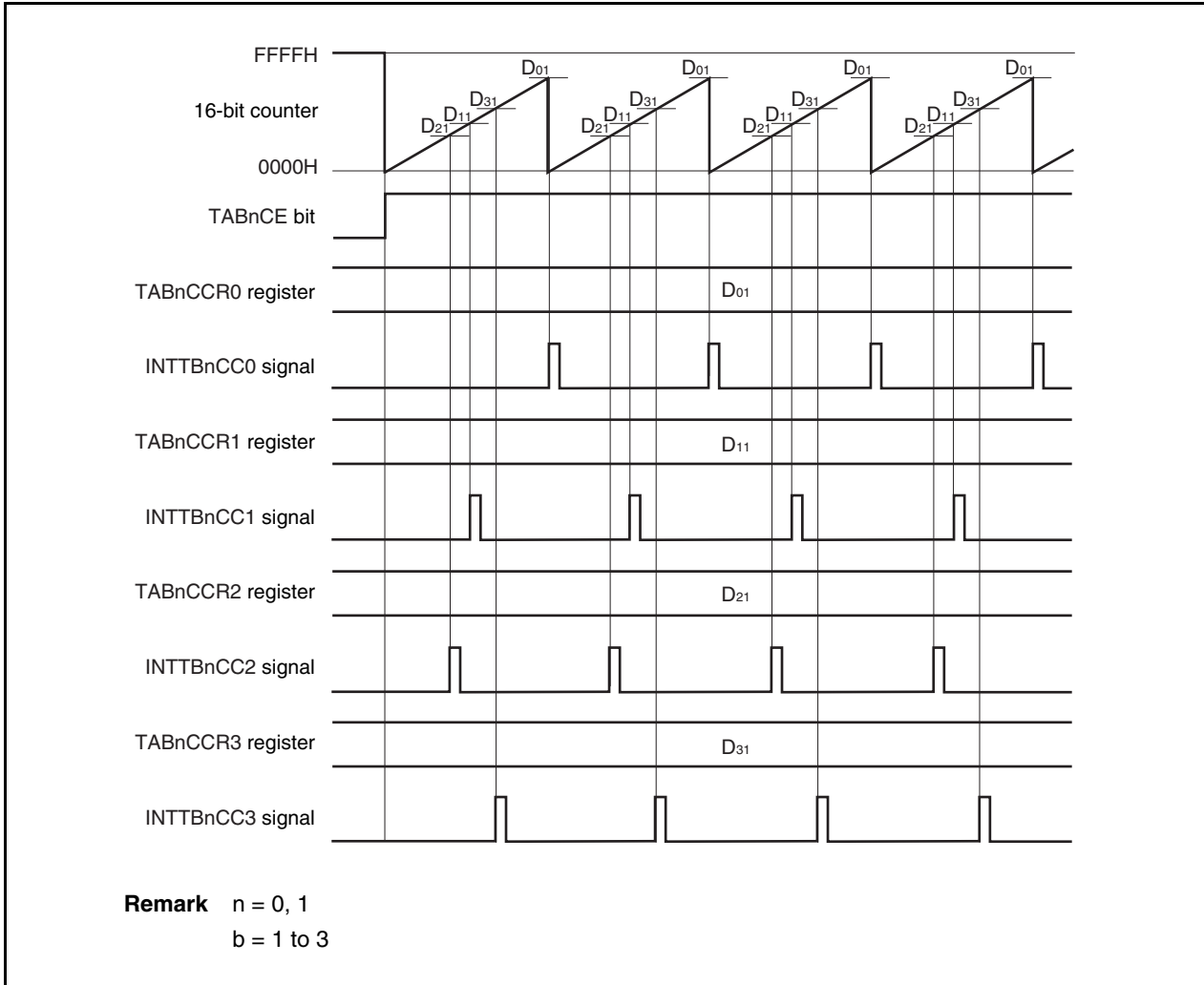
(e) Operation of TABnCCR1 to TABnCCR3 registers

Figure 7-17. Configuration of TABnCCR1 to TABnCCR3 Registers



If the set value of the TABnCCRb register is smaller than the set value of the TABnCCR0 register, the INTTBnCCb signal is generated once per cycle.

Figure 7-18. Timing Chart When  $D_{01} \geq D_{b1}$

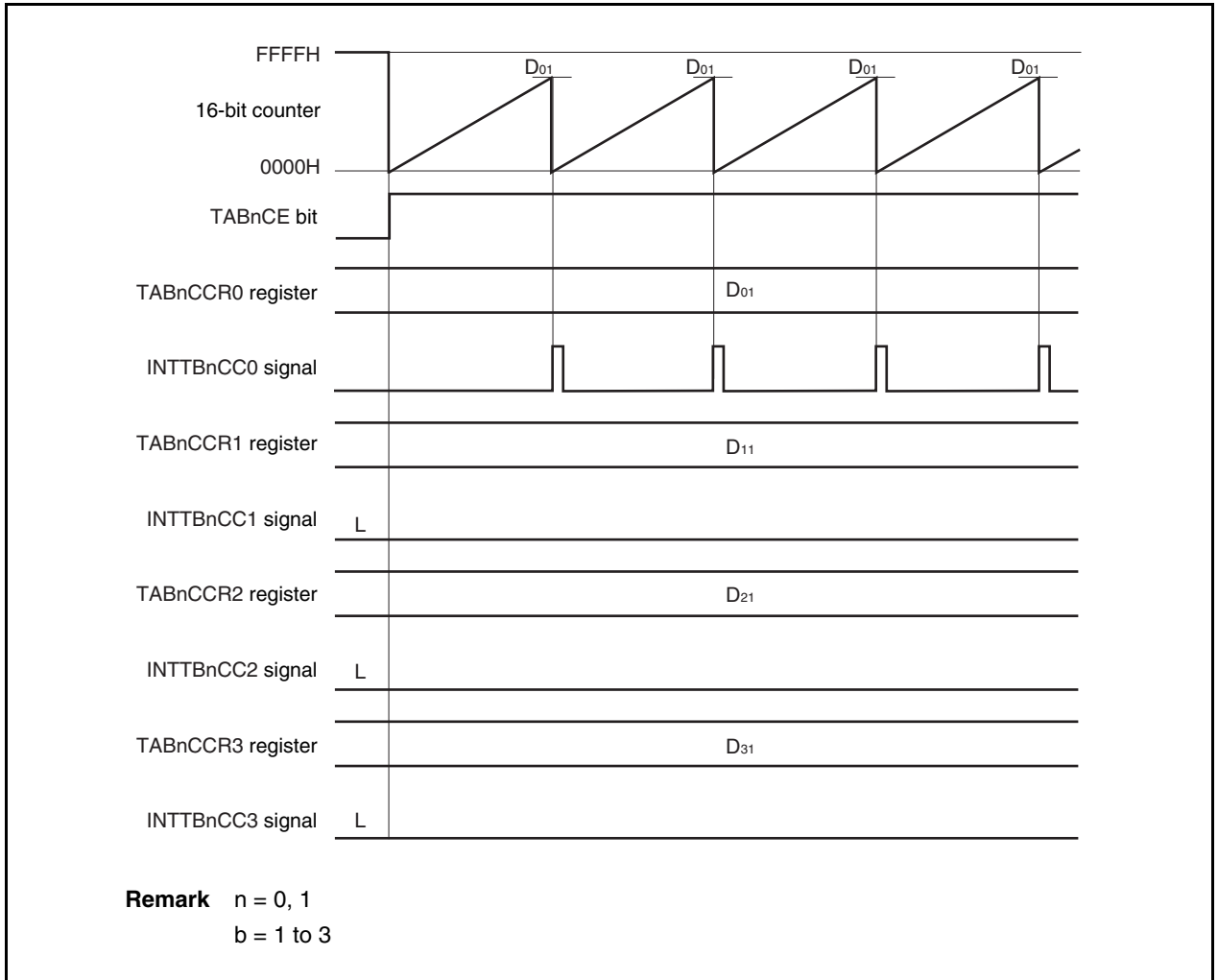




If the set value of the TABnCCRb register is greater than the set value of the TABnCCR0 register, the INTTBnCCb signal is not generated because the count value of the 16-bit counter and the value of the TABnCCRb register do not match.

When the TABnCCRb register is not used, it is recommended to set its value to FFFFH.

Figure 7-19. Timing Chart When  $D_{01} < D_{b1}$



7.6.3 External trigger pulse output mode (TABnMD2 to TABnMD0 bits = 010)

In the external trigger pulse output mode, 16-bit timer/event counter AB waits for a trigger when the TABnCTL0.TABnCE bit is set to 1. When the valid edge of an external trigger input signal (TRGBn) is detected, 16-bit timer/event counter AB starts counting, and outputs a PWM waveform (up to 3-phase) from the TOBn1 to TOBn3 pins. A PWM waveform with a duty factor of 50% whose half cycle is the set value of the TABnCCR0 register + 1 can also be output from the TOBn0 pin.

Pulses can also be output by generating a software trigger instead of using the external trigger input.

Figure 7-20. Configuration in External Trigger Pulse Output Mode

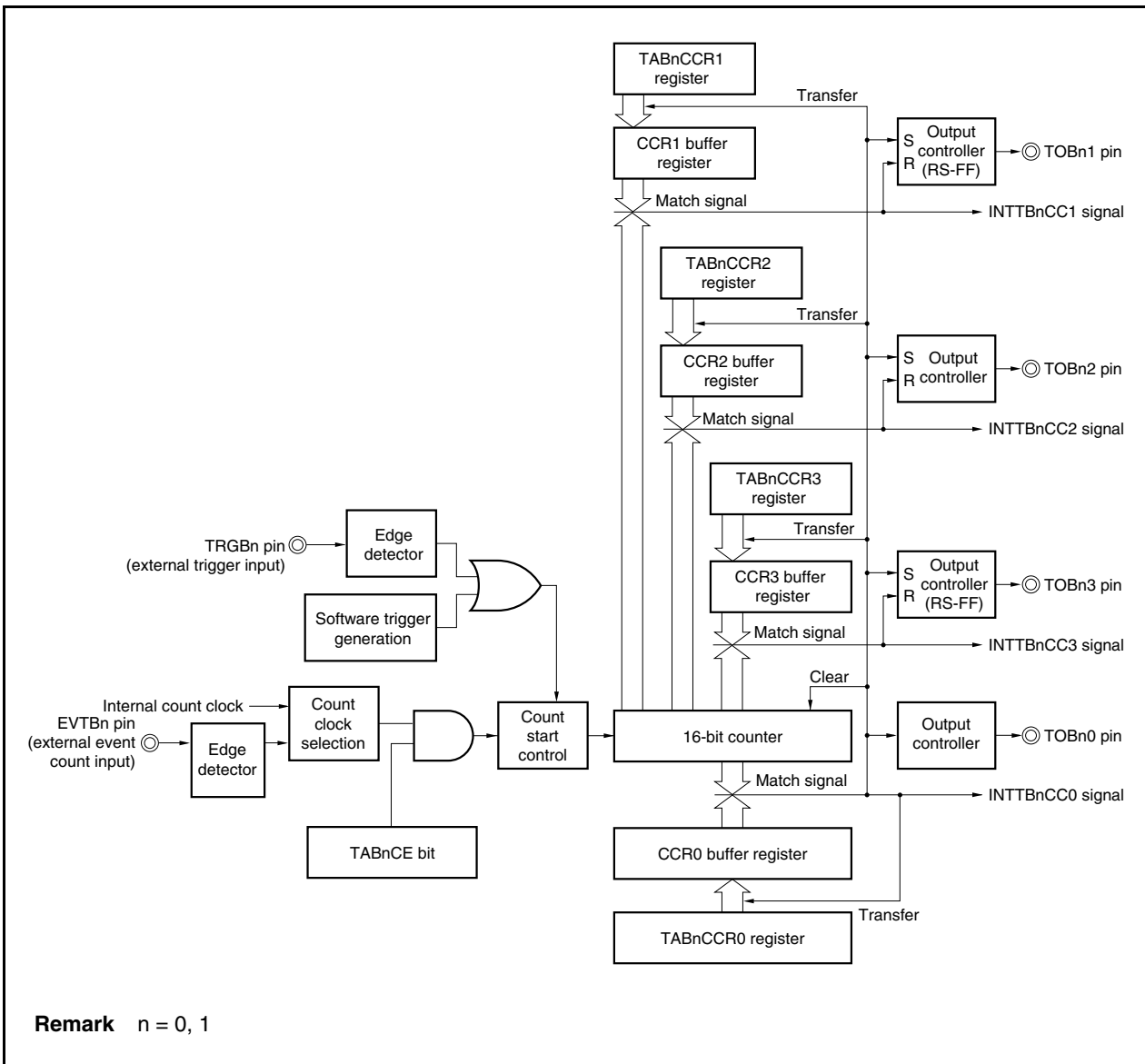
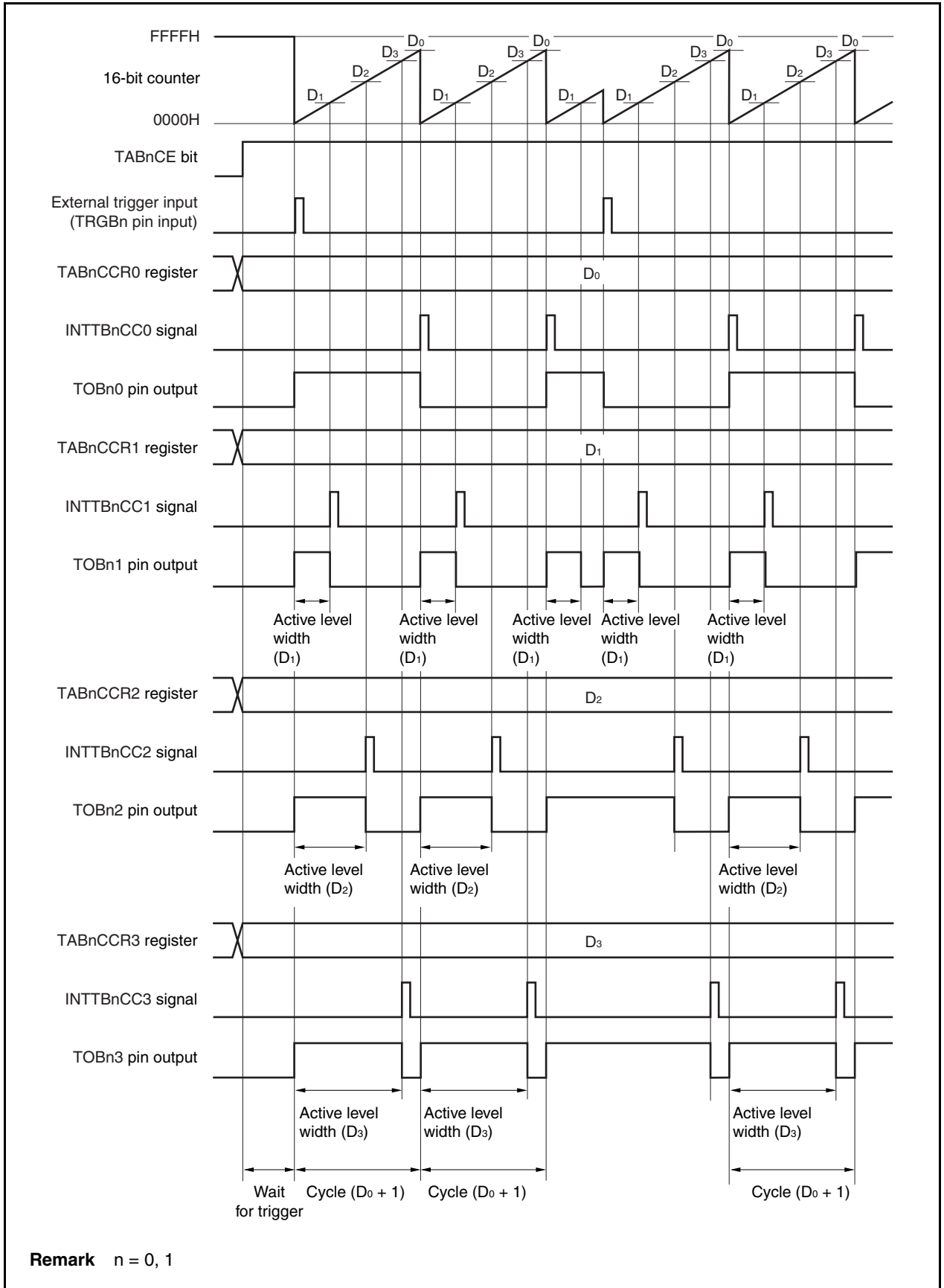


Figure 7-21. Basic Timing in External Trigger Pulse Output Mode



16-bit timer/event counter AB waits for a trigger when the TABnCE 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 TOBnb pin. If the trigger is generated again while the counter is operating, the counter is cleared to 0000H and restarted. (The output of the TOBn0 pin is inverted. The TOBnb 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.

$$\text{Active level width} = (\text{Set value of TABnCCRb register}) \times \text{Count clock cycle}$$

$$\text{Cycle} = (\text{Set value of TABnCCR0 register} + 1) \times \text{Count clock cycle}$$

$$\text{Duty factor} = (\text{Set value of TABnCCRb register}) / (\text{Set value of TABnCCR0 register} + 1)$$

The compare match request signal INTTBnCC0 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 INTTBnCCb is generated when the count value of the 16-bit counter matches the value of the CCRb buffer register.

The value set to the TABnCCRa register is transferred to the CCRa 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 (TRGBn), or setting the software trigger (TABnCTL1.TABnEST bit) to 1 is used as the trigger.

**Remark** n = 0, 1  
a = 0 to 3  
b = 1 to 3

Figure 7-22. Setting of Registers in External Trigger Pulse Output Mode (1/3)

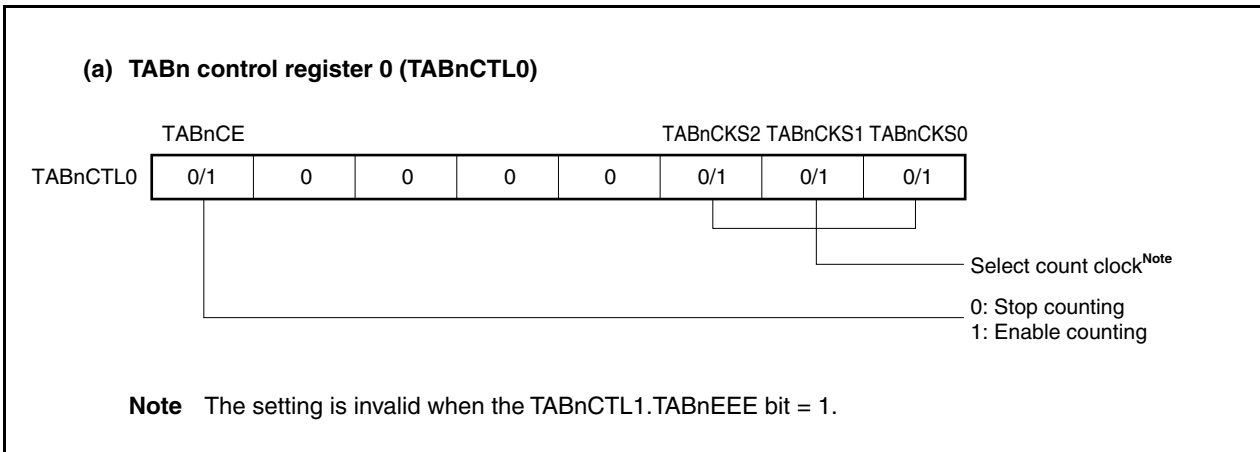


Figure 7-22. Setting of Registers in External Trigger Pulse Output Mode (2/3)

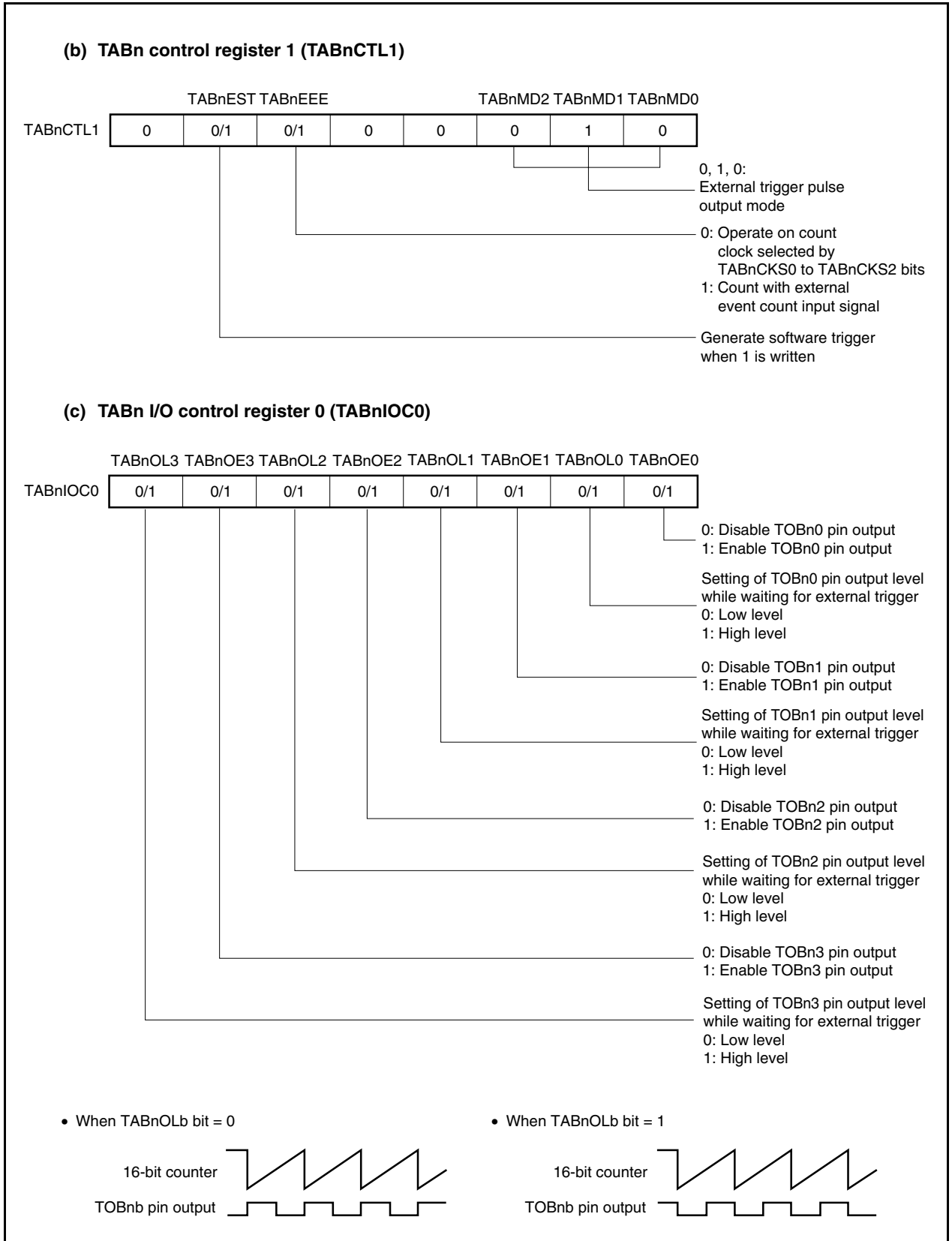
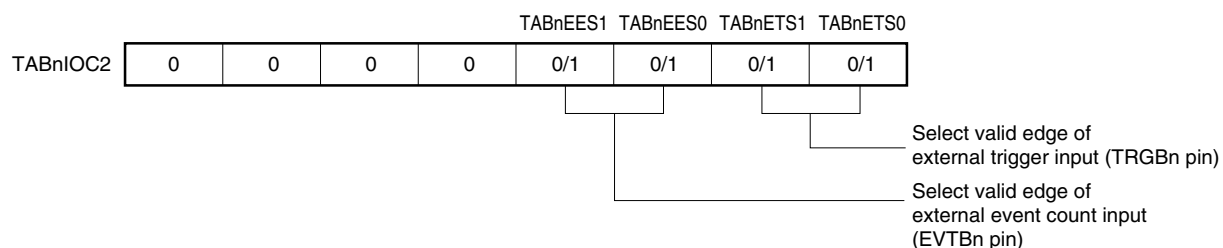


Figure 7-22. Setting of Registers in External Trigger Pulse Output Mode (3/3)

**(d) TABn I/O control register 2 (TABnIOC2)****(e) TABn counter read buffer register (TABnCNT)**

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

**(f) TABn capture/compare registers 0 to 3 (TABnCCR0 to TABnCCR3)**

If  $D_0$  is set to the TABnCCR0 register,  $D_1$  to the TABnCCR1 register,  $D_2$  to the TABnCCR2 register, and  $D_3$ , to the TABnCCR3 register, the cycle and active level of the PWM waveform are as follows.

$$\text{Cycle} = (D_0 + 1) \times \text{Count clock cycle}$$

$$\text{TOBn1 pin PWM waveform active level width} = D_1 \times \text{Count clock cycle}$$

$$\text{TOBn2 pin PWM waveform active level width} = D_2 \times \text{Count clock cycle}$$

$$\text{TOBn3 pin PWM waveform active level width} = D_3 \times \text{Count clock cycle}$$

**Remarks 1.** TABn I/O control register 1 (TABnIOC1) and TABn option register 0 (TABnOPT0) are not used in the external trigger pulse output mode.

2.  $n = 0, 1$   
 $b = 1 \text{ to } 3$

(1) Operation flow in external trigger pulse output mode

Figure 7-23. Software Processing Flow in External Trigger Pulse Output Mode (1/2)

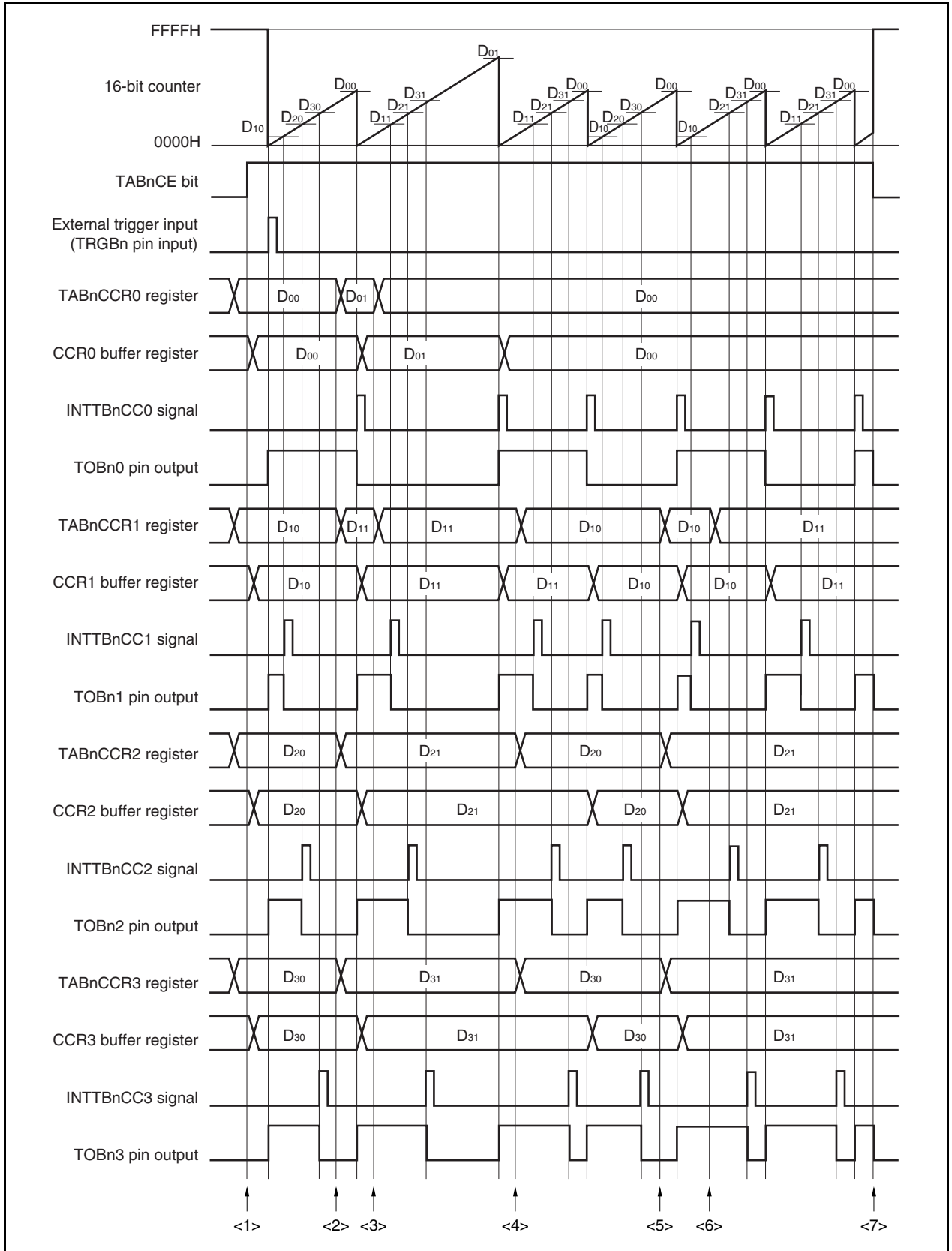
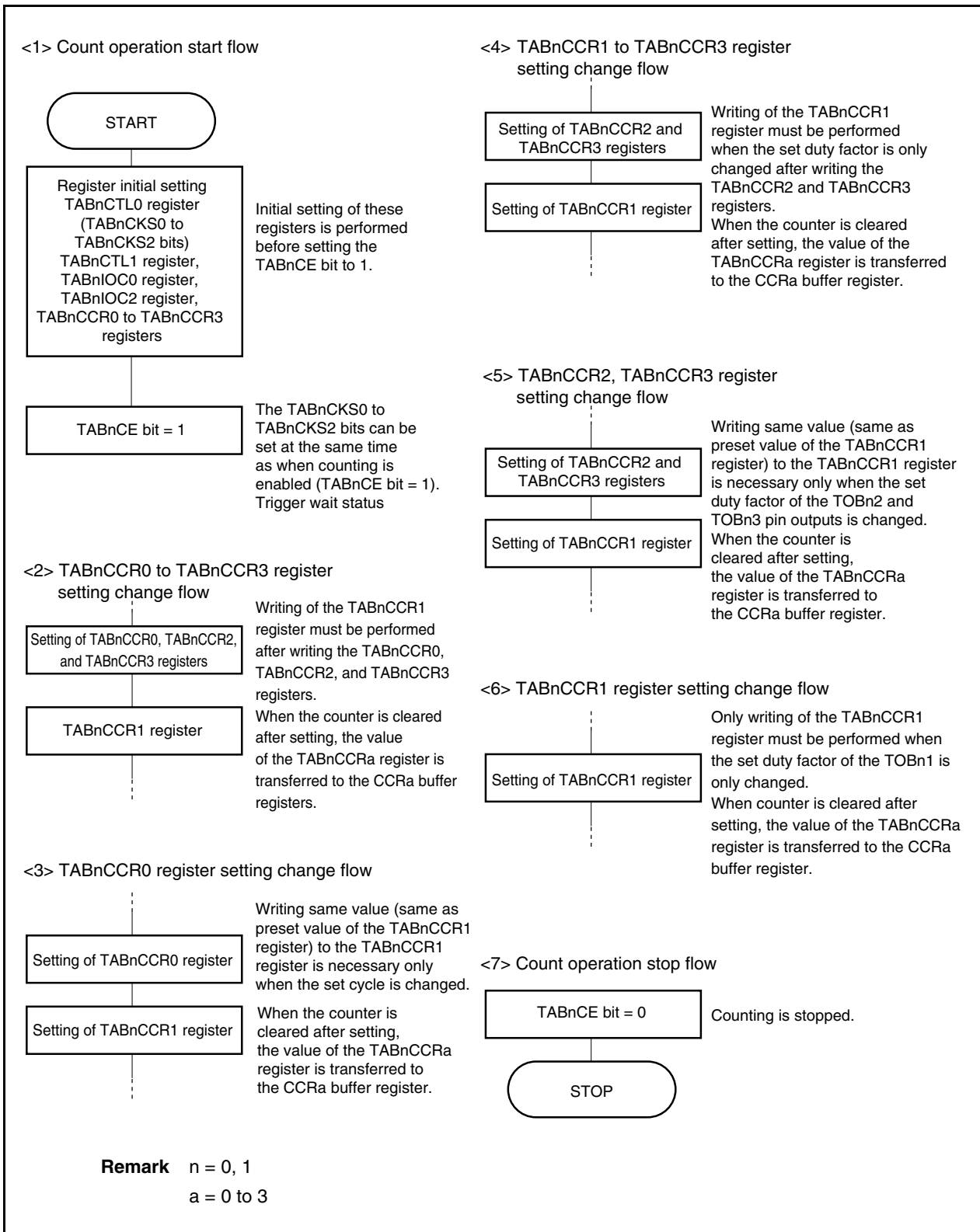


Figure 7-23. Software Processing Flow in External Trigger Pulse Output Mode (2/2)



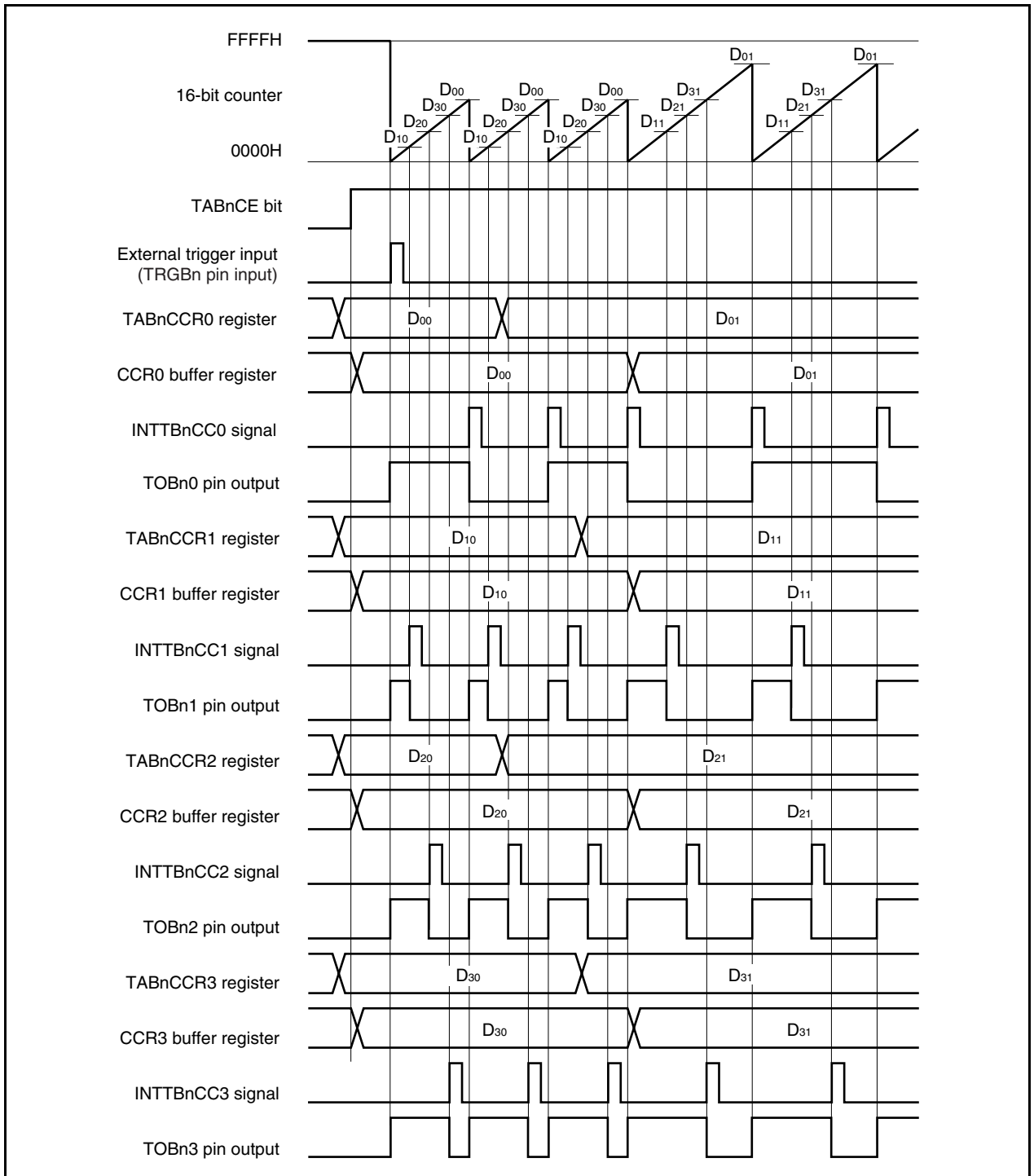


(2) External trigger pulse output mode operation timing

(a) Note on changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TABnCCR1 register last. Rewrite the TABnCCRB register after writing the TABnCCR1 register after the INTTBnCC0 signal is detected.

**Remark** n = 0, 1  
b = 1 to 3



In order to transfer data from the TABnCCRa register to the CCRa buffer register, the TABnCCR1 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 TABnCCR0 register, set the active level width to the TABnCCR2 and TABnCCR3 registers, and then set an active level to the TABnCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TABnCCR0 register, and then write the same value (same as preset value of the TABnCCR1 register) to the TABnCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, first set an active level to the TABnCCR2 and TABnCCR3 registers and then set an active level to the TABnCCR1 register.

To change only the active level width (duty factor) of the PWM waveform output by the TOBn1 pin, only the TABnCCR1 register has to be set.

To change only the active level width (duty factor) of the PWM waveform output by the TOBn2 and TOBn3 pins, first set an active level width to the TABnCCR2 and TABnCCR3 registers, and then write the same value (same as preset value of the TABnCCR1 register) to the TABnCCR1 register.

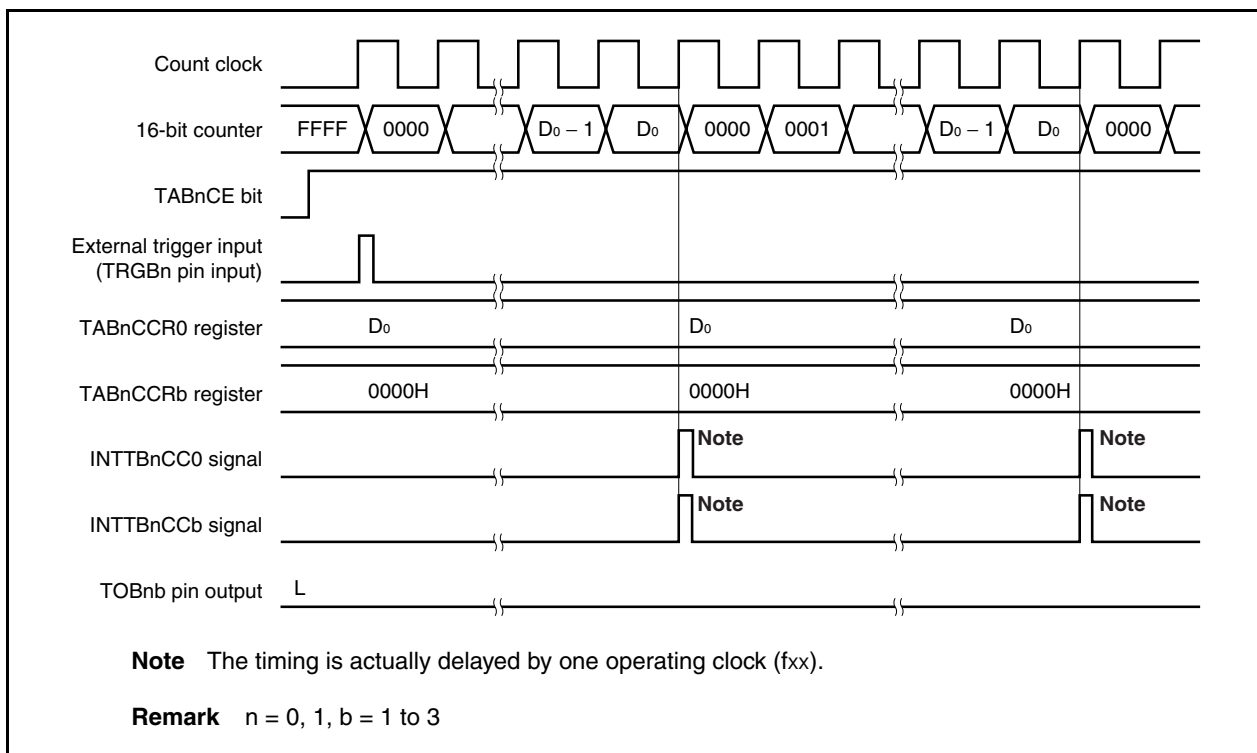
After data is written to the TABnCCR1 register, the value written to the TABnCCRa register is transferred to the CCRa buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TABnCCR0 to TABnCCR3 registers again after writing the TABnCCR1 register once, do so after the INTTBnCC0 signal is generated. Otherwise, the value of the CCRa buffer register may become undefined because timing of transferring data from the TABnCCRa register to the CCRa buffer register conflicts with writing the TABnCCRa register.

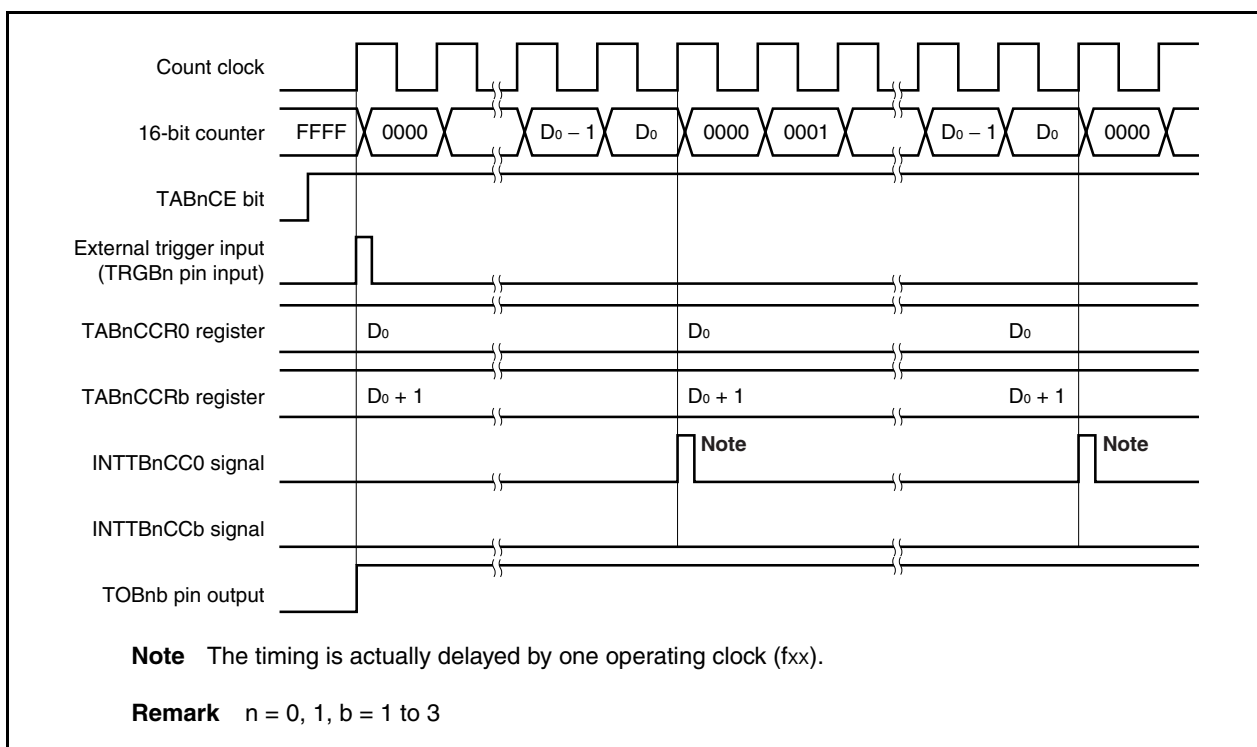
**Remark** n = 0, 1  
a = 0 to 3

**(b) 0%/100% output of PWM waveform**

To output a 0% waveform, set the TABnCCRb register to 0000H. The 16-bit counter is cleared to 0000H and the INTTBnCC0 and INTTBnCCb signals are generated at the next timing after a match between the count value of the 16-bit counter and the value of the CCR0 buffer register.

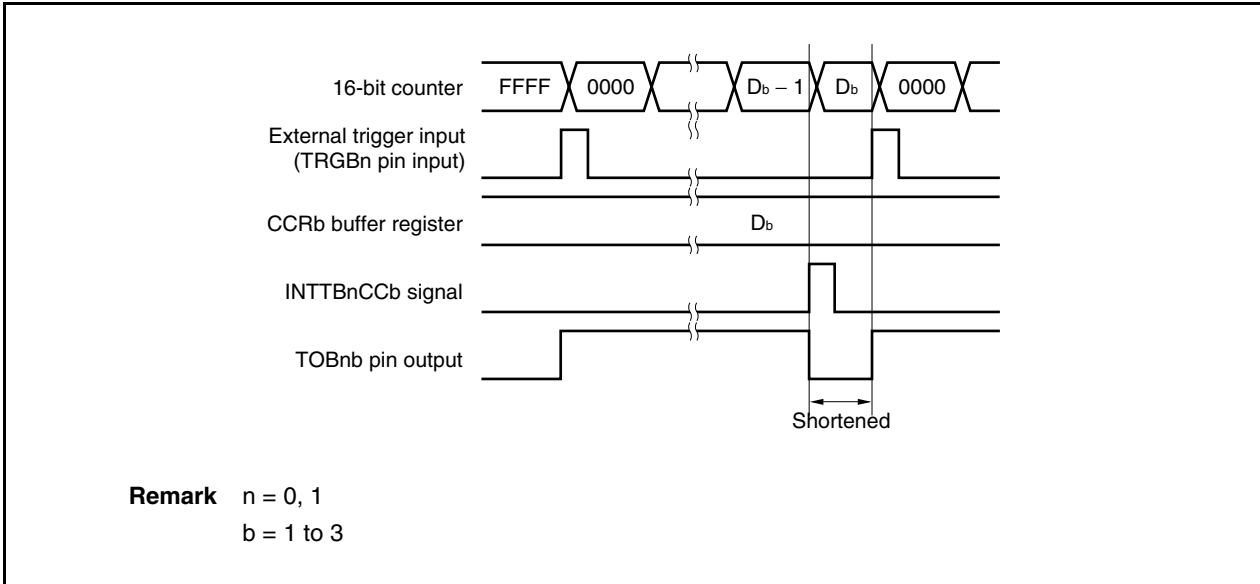


To output a 100% waveform, set a value of (set value of TABnCCR0 register + 1) to the TABnCCRb register. If the set value of the TABnCCR0 register is FFFFH, 100% output cannot be produced.

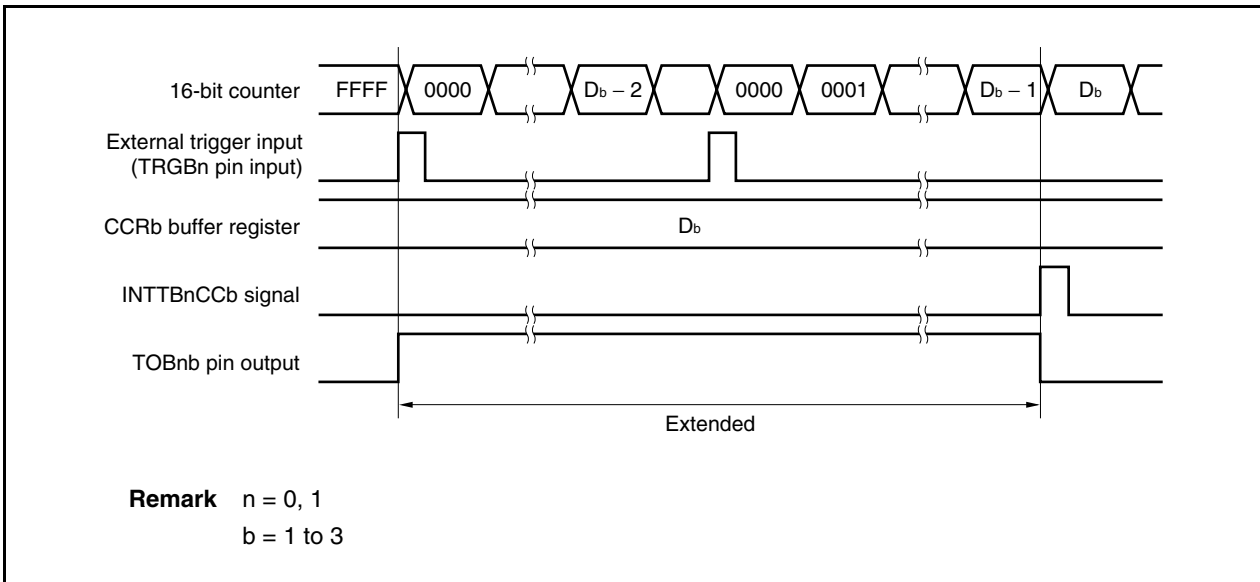


**(c) Conflict between trigger detection and match with CCRb buffer register**

If the trigger is detected immediately after the INTTBnCCb signal is generated, the 16-bit counter is immediately cleared to 0000H, the output signal of the TOBnb pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.

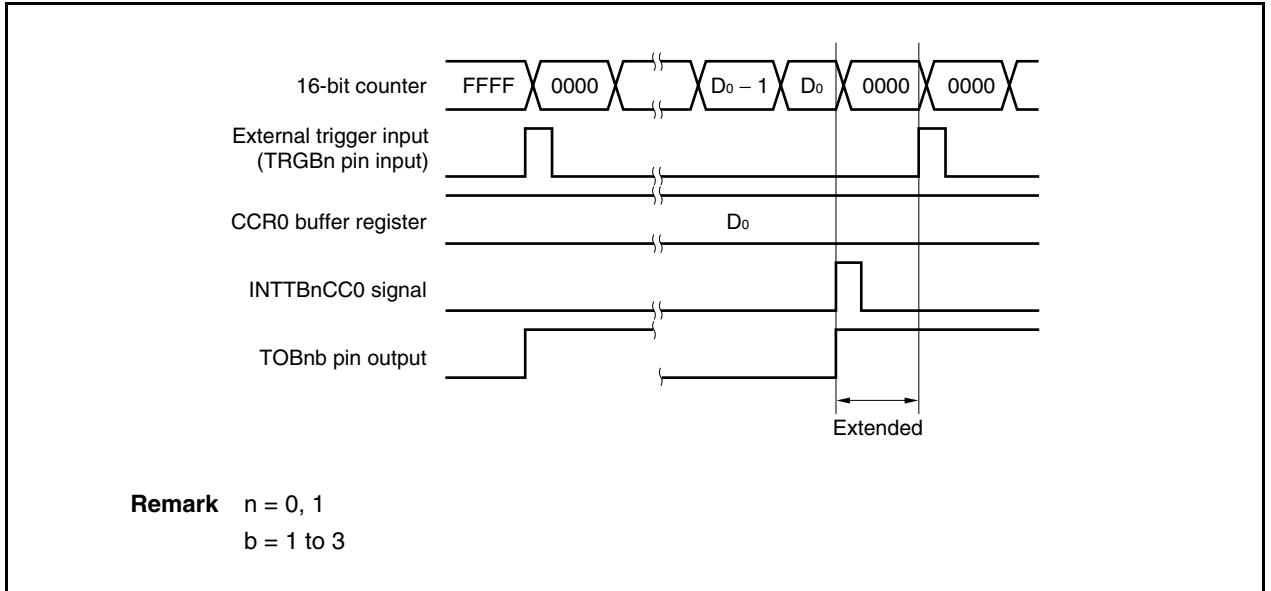


If the trigger is detected immediately before the INTTBnCCb signal is generated, the INTTBnCCb signal is not generated, and the 16-bit counter is cleared to 0000H and continues counting. The output signal of the TOBnb pin remains active. Consequently, the active period of the PWM waveform is extended.

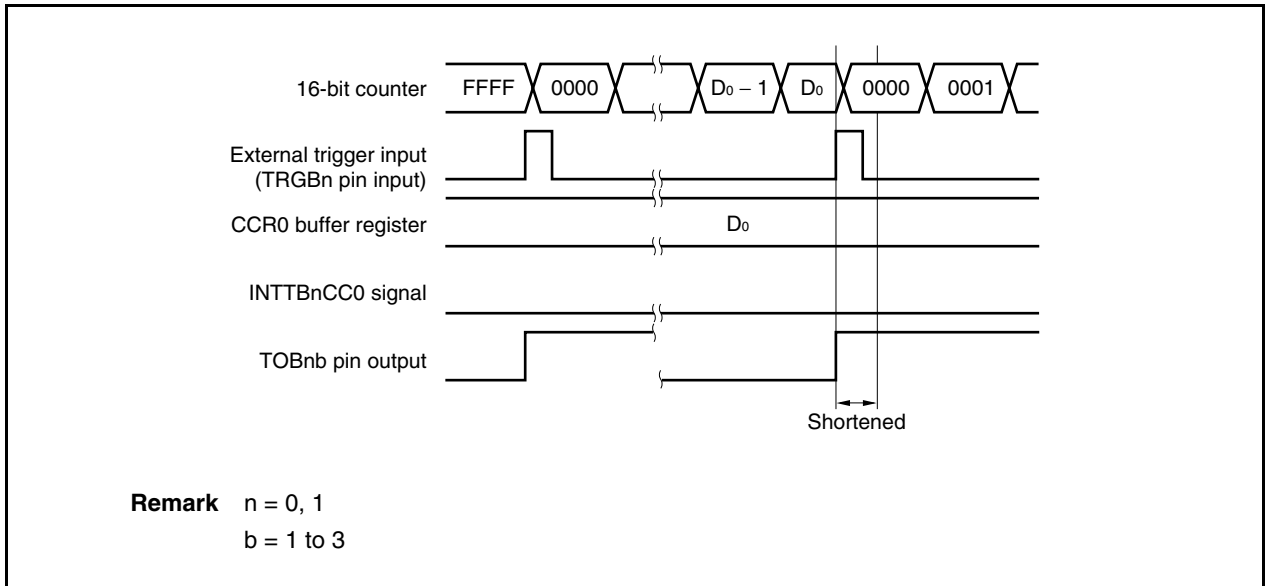


**(d) Conflict between trigger detection and match with CCR0 buffer register**

If the trigger is detected immediately after the INTTBnCC0 signal is generated, the 16-bit counter is cleared to 0000H and continues counting up. Therefore, the active period of the TOBnb pin is extended by time from generation of the INTTBnCC0 signal to trigger detection.

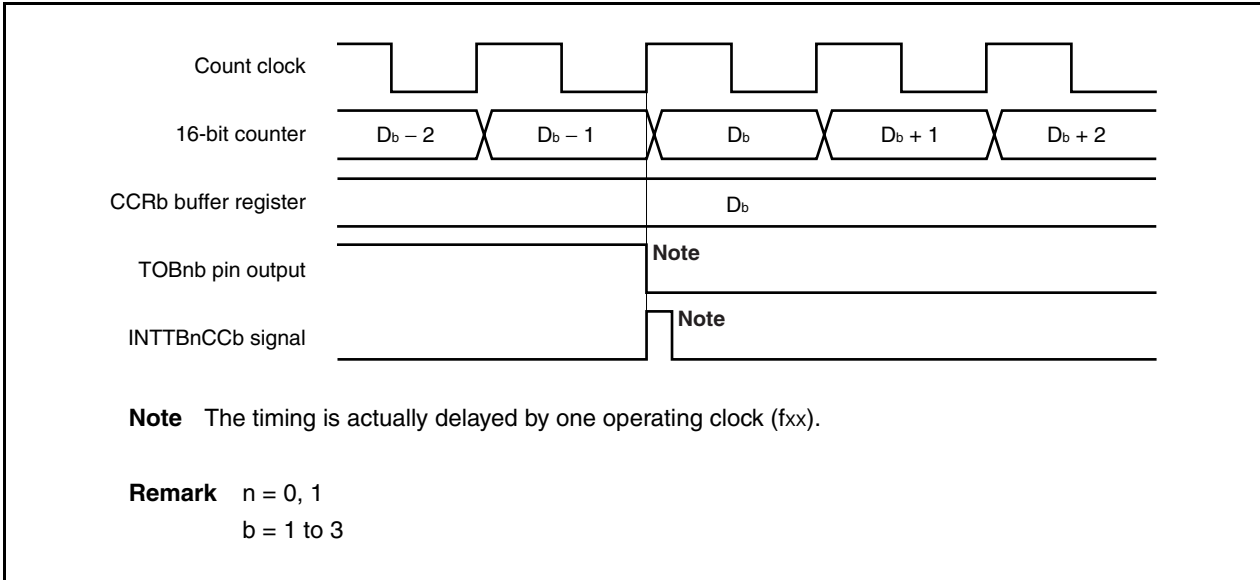


If the trigger is detected immediately before the INTTBnCC0 signal is generated, the INTTBnCC0 signal is not generated. The 16-bit counter is cleared to 0000H, the TOBnb pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



**(e) Generation timing of compare match interrupt request signal (INTTBnCCb)**

The timing of generation of the INTTBnCCb signal in the external trigger pulse output mode differs from the timing of INTTBnCCb signals in other mode; the INTTBnCCb signal is generated when the count value of the 16-bit counter matches the value of the CCRb buffer register.



Usually, the INTTBnCCb signal is generated in synchronization with the next count up after the count value of the 16-bit counter matches the value of the CCRb 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 TOBnb pin.

7.6.4 One-shot pulse output mode (TABnMD2 to TABnMD0 bits = 011)

In the one-shot pulse output mode, 16-bit timer/event counter AB waits for a trigger when the TABnCTL0.TABnCE bit is set to 1. When the valid edge of an external trigger input (TRGBn) is detected, 16-bit timer/event counter AB starts counting, and outputs a one-shot pulse from the TOBn1 to TOBn3 pins. The TOBn0 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).

Instead of the external trigger input, a software trigger can also be generated to output the pulse.

Figure 7-24. Configuration in One-Shot Pulse Output Mode

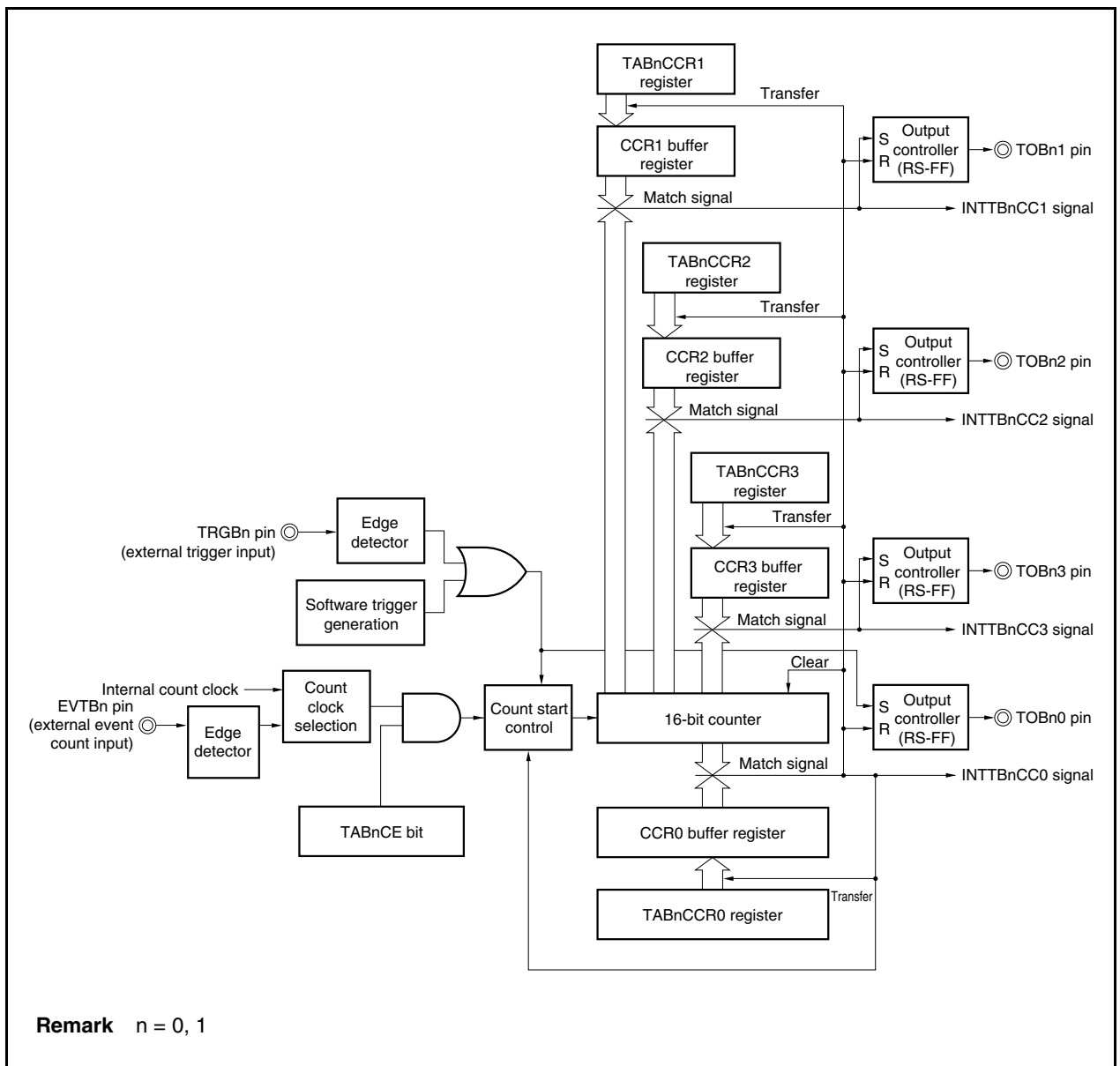
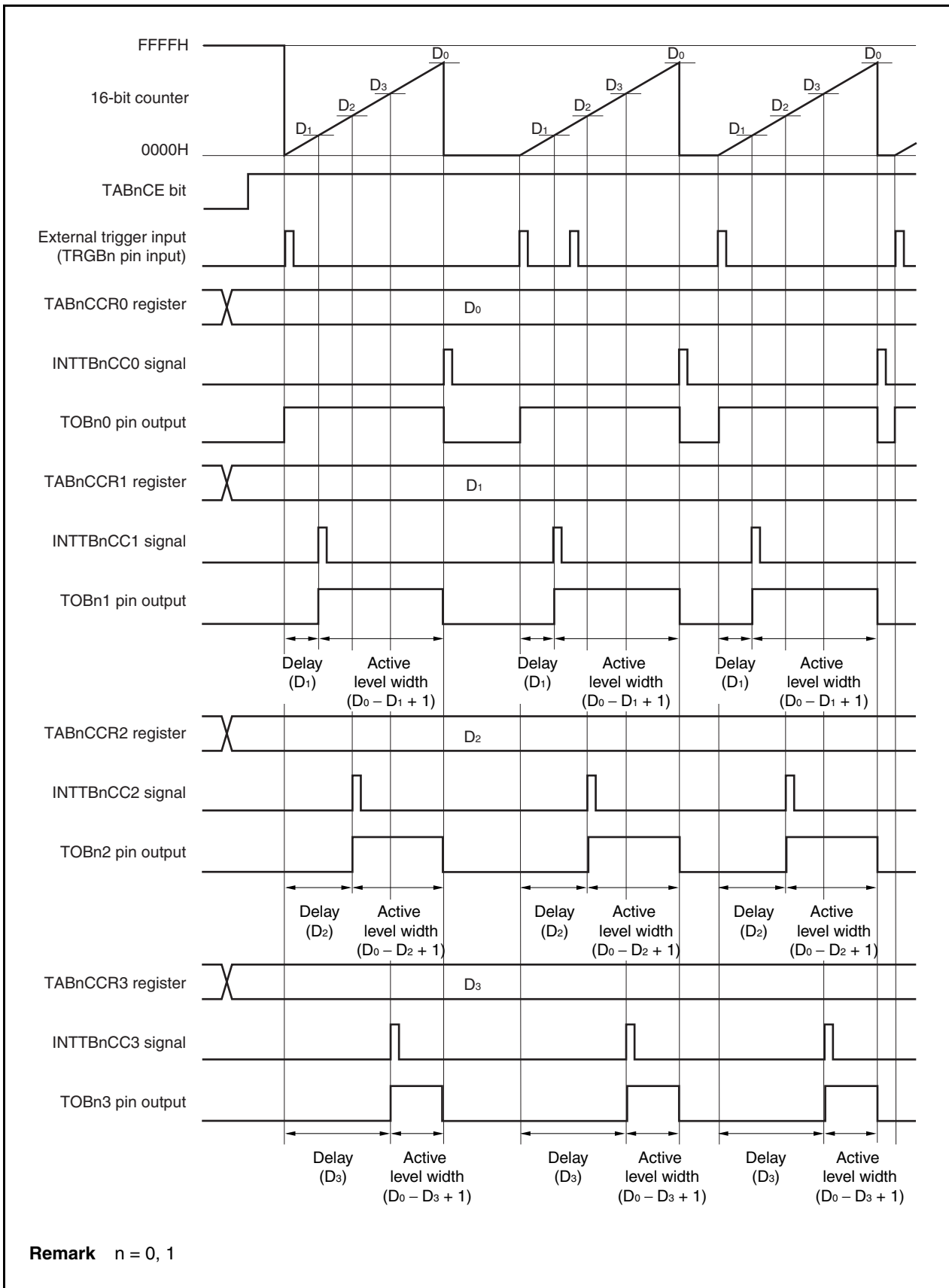


Figure 7-25. Basic Timing in One-Shot Pulse Output Mode





When the TABnCE bit is set to 1, 16-bit timer/event counter AB 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 TOBnb pin. After the one-shot pulse is output, the 16-bit counter is set to 0000H, stops counting, and waits for a trigger. When the trigger is generated again, the 16-bit counter starts counting from 0000H. 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.

$$\text{Output delay period} = (\text{Set value of TABnCCRb register}) \times \text{Count clock cycle}$$

$$\text{Active level width} = (\text{Set value of TABnCCR0 register} - \text{Set value of TABnCCRb register} + 1) \times \text{Count clock cycle}$$

The compare match interrupt request signal INTTBnCC0 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 INTTBnCCb is generated when the count value of the 16-bit counter matches the value of the CCRb buffer register.

The valid edge of an external trigger input (TRGBn) or setting the software trigger (TABnCTL1.TABnEST bit) to 1 is used as the trigger.

**Remark** n = 0, 1, b = 1 to 3

**Figure 7-26. Setting of Registers in One-Shot Pulse Output Mode (1/3)**

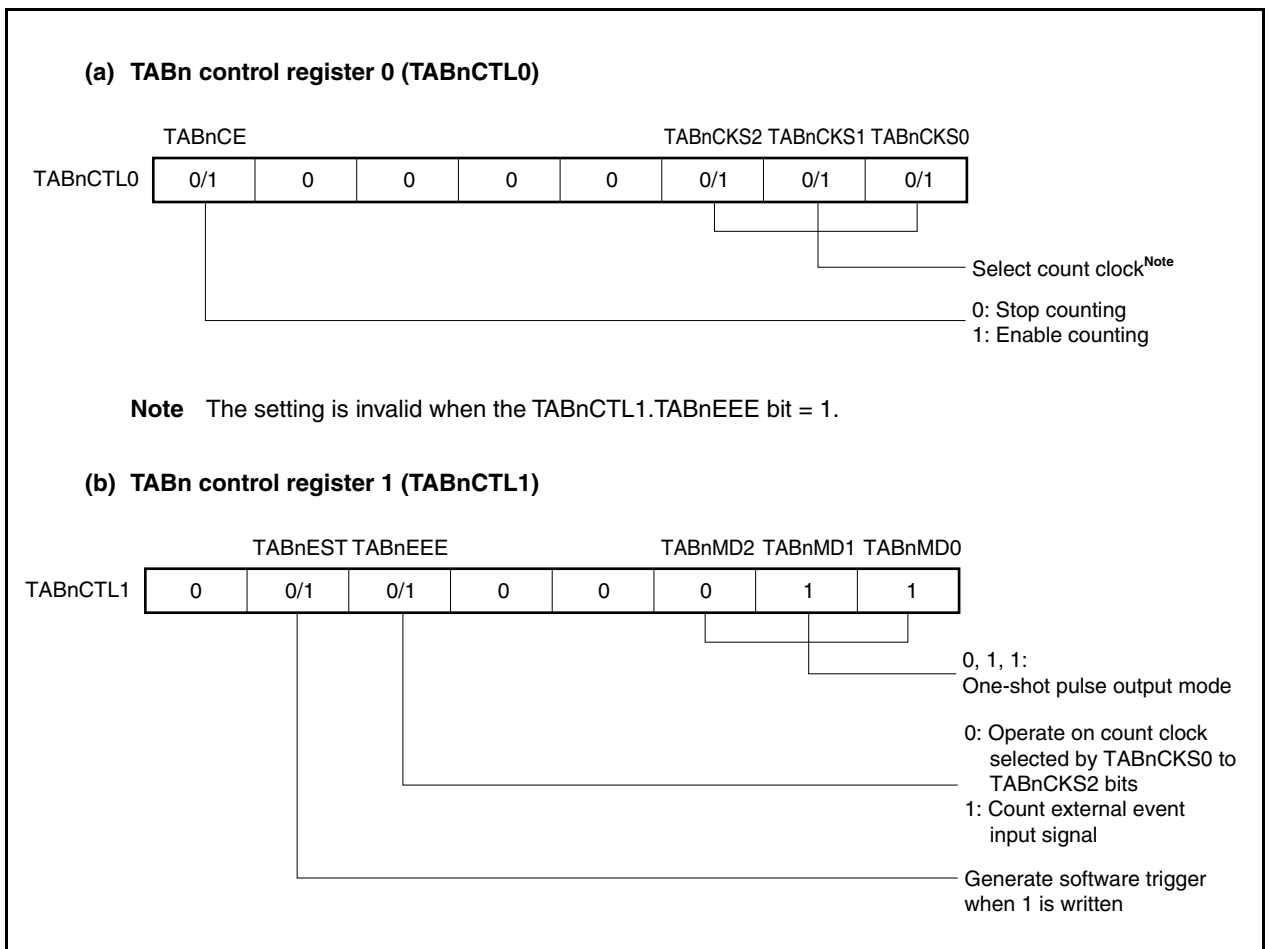


Figure 7-26. Register Setting in One-Shot Pulse Output Mode (2/3)

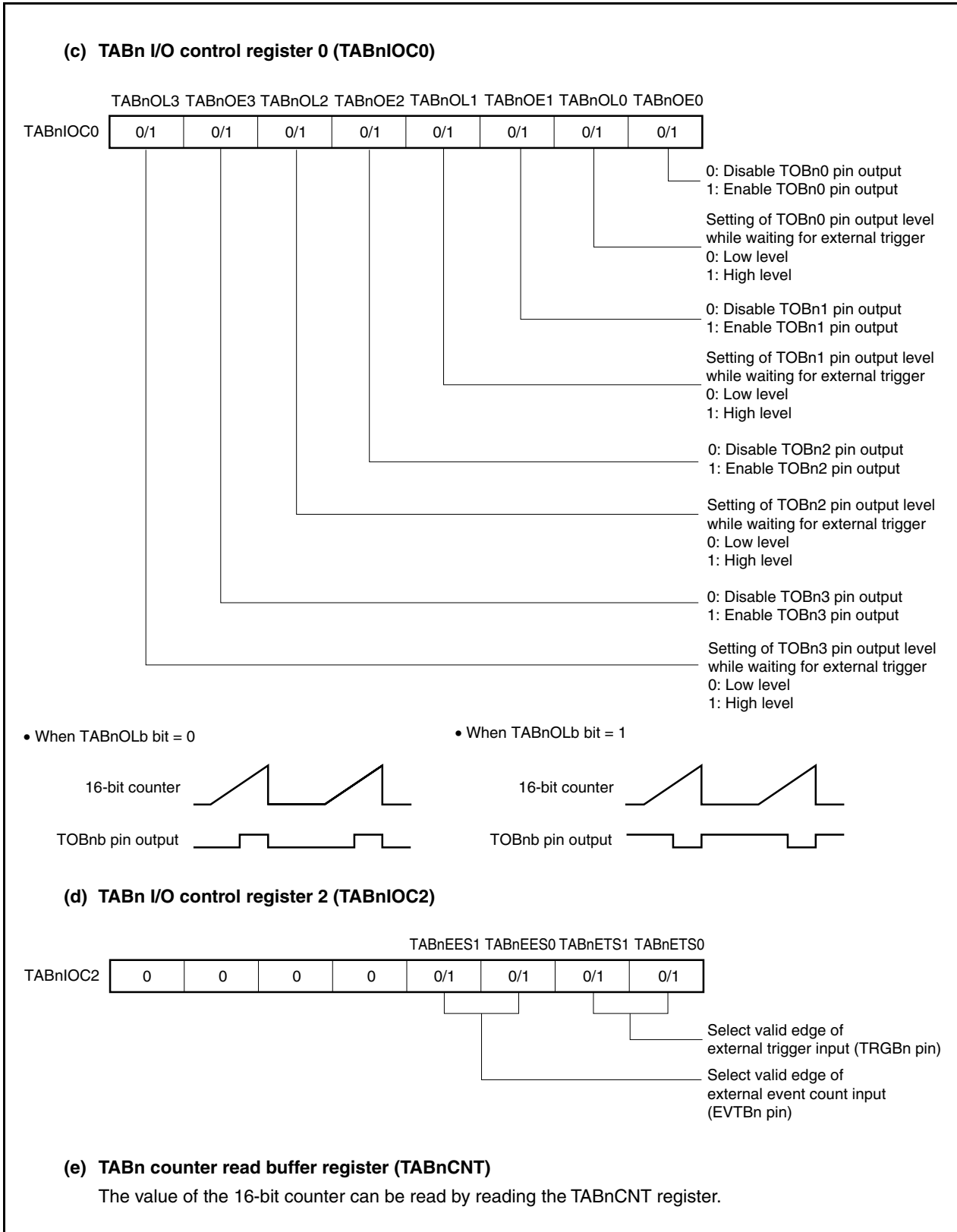


Figure 7-26. Register Setting in One-Shot Pulse Output Mode (3/3)

**(f) TABn capture/compare registers 0 to 3 (TABnCCR0 to TABnCCR3)**

If  $D_0$  is set to the TABnCCR0 register and  $D_b$  to the TABnCCRb register, the active level width and output delay period of the one-shot pulse are as follows.

Active level width =  $(D_b - D_0 + 1) \times$  Count clock cycle

Output delay period =  $D_b \times$  Count clock cycle

**Caution** One-shot pulses are not output even in the one-shot pulse output mode, if the value set in the TABnCCRb register is greater than that set in the TABnCCR0 register.

**Remarks** 1. TABn I/O control register 1 (TABnIOC1) and TABn option register 0 (TABnOPT0) are not used in the one-shot pulse output mode.

2.  $n = 0, 1$

$b = 1$  to 3

(1) Operation flow in one-shot pulse output mode

Figure 7-27. Software Processing Flow in One-Shot Pulse Output Mode (1/2)

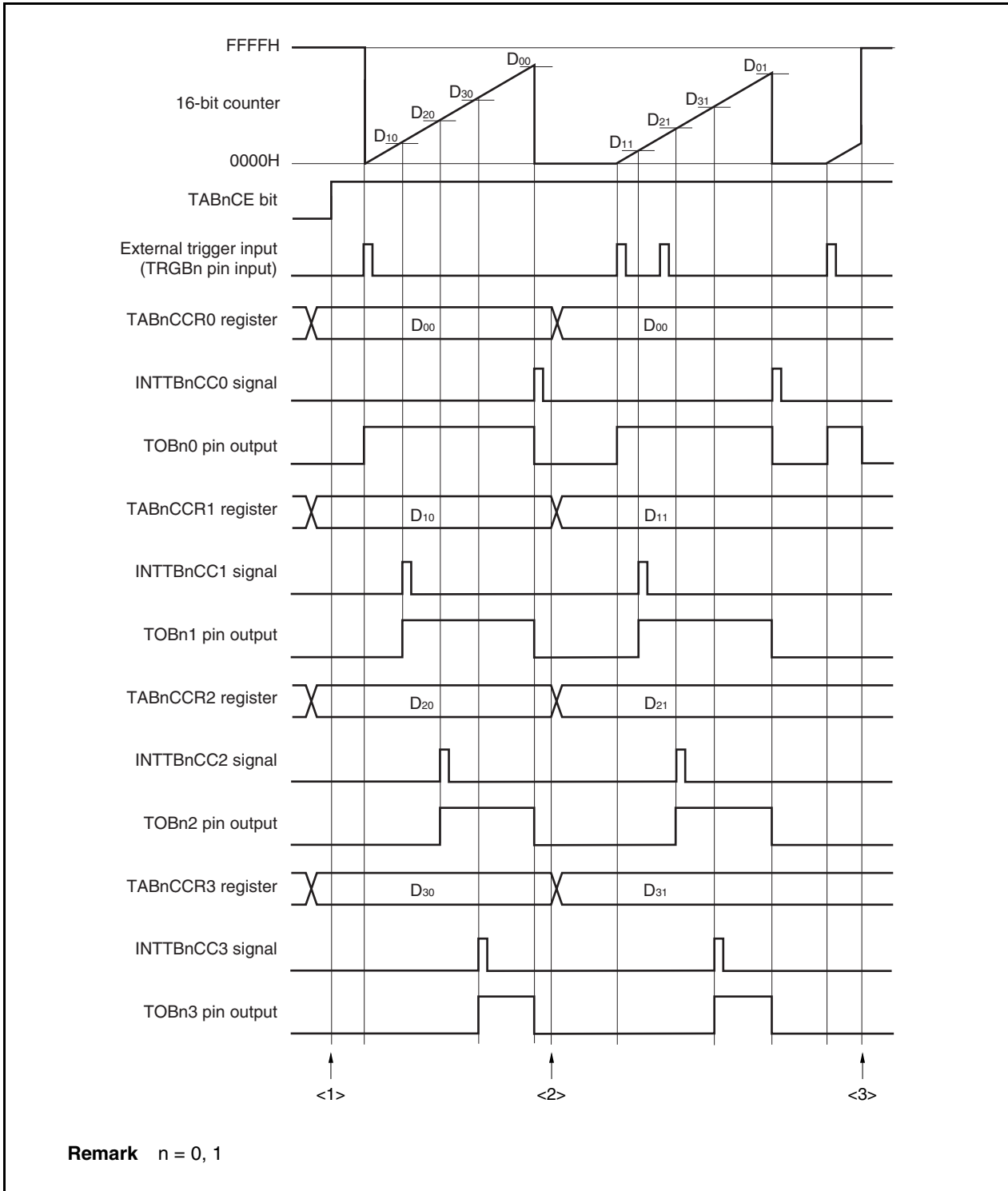
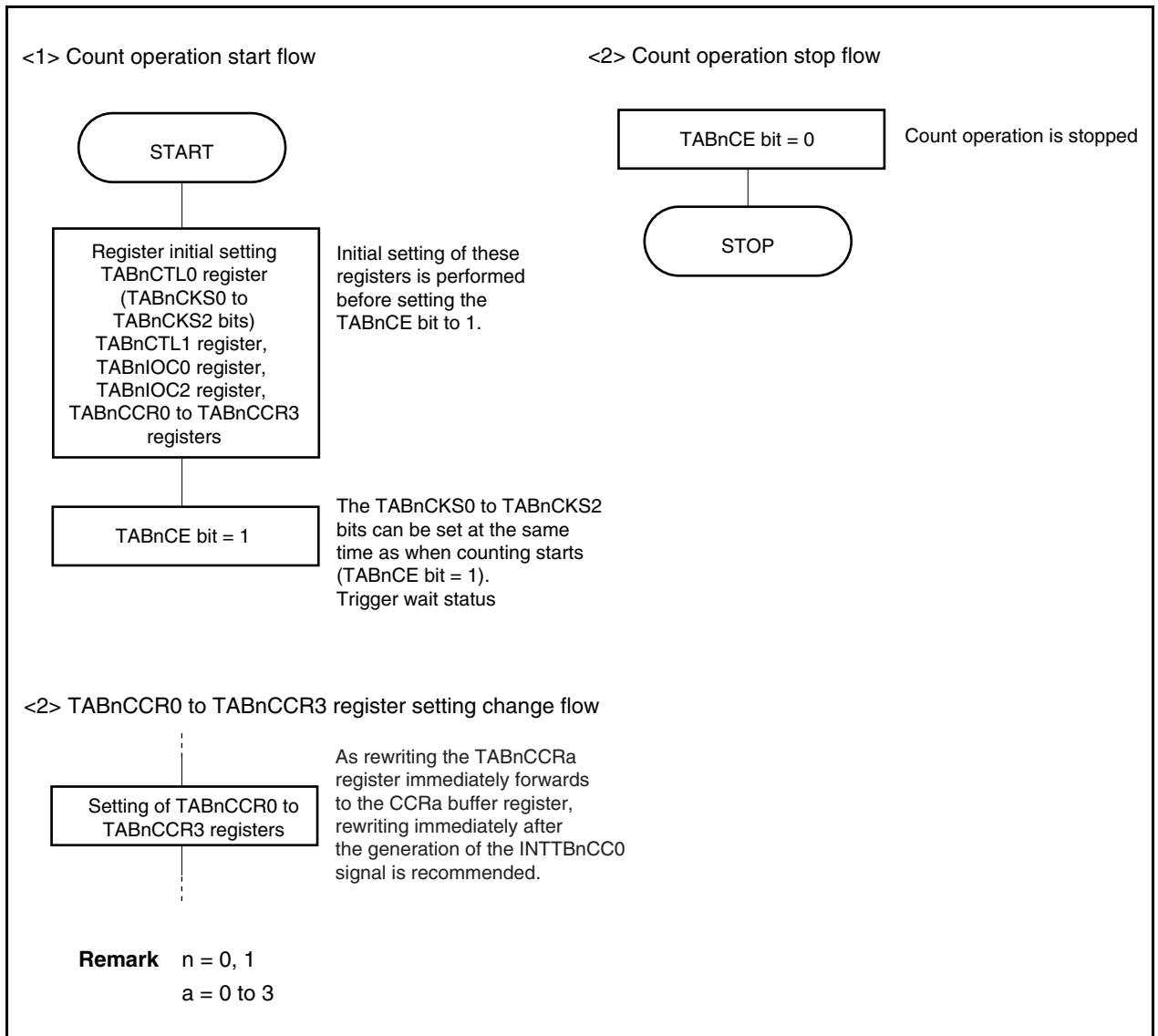


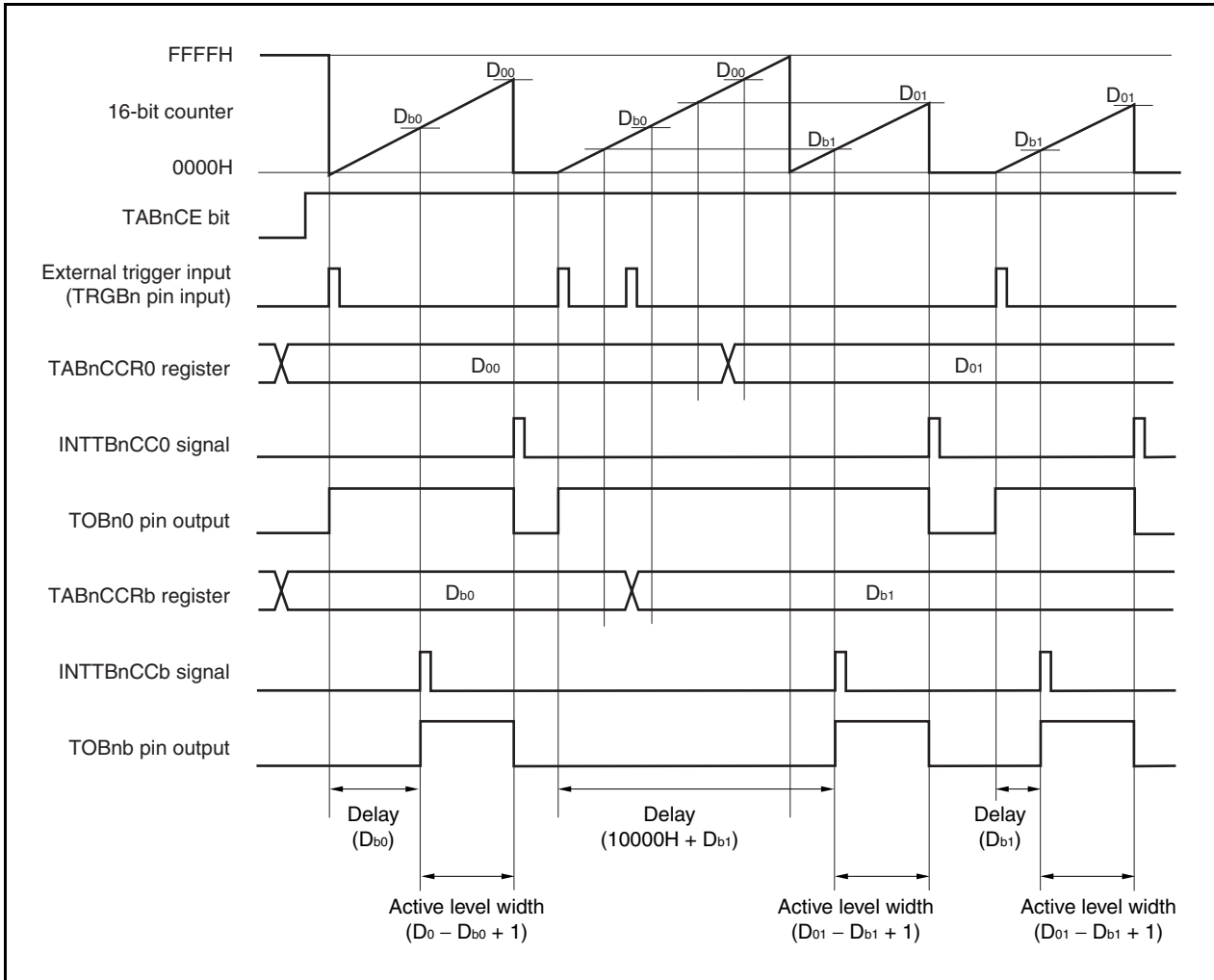
Figure 7-27. Software Processing Flow in One-Shot Pulse Output Mode (2/2)



(2) Operation timing in one-shot pulse output mode

(a) Note on rewriting TABnCCRa register

To change the set value of the TABnCCRa register to a smaller value, stop counting once, and then change the set value. When the overflow may occur, stop counting once, and then change the set value.



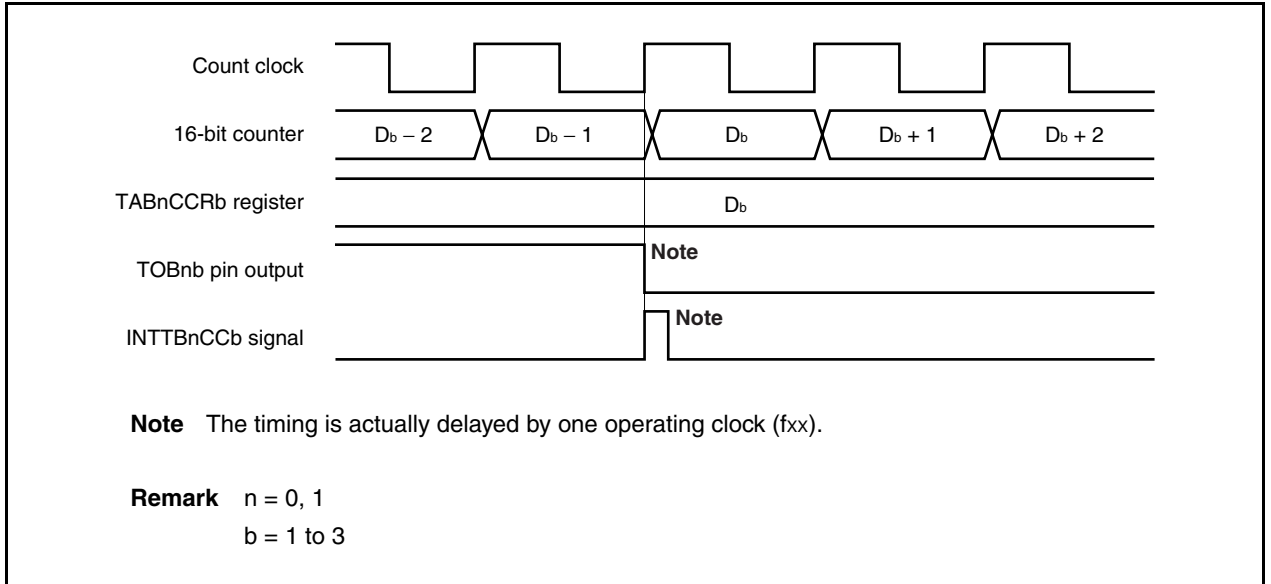
When the TABnCCR0 register is rewritten from D<sub>00</sub> to D<sub>01</sub> and the TABnCCRb register from D<sub>b0</sub> to D<sub>b1</sub> where D<sub>00</sub> > D<sub>01</sub> and D<sub>b0</sub> > D<sub>b1</sub>, if the TABnCCRb register is rewritten when the count value of the 16-bit counter is greater than D<sub>b1</sub> and less than D<sub>b0</sub> and if the TABnCCR0 register is rewritten when the count value is greater than D<sub>01</sub> and less than D<sub>00</sub>, 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<sub>b1</sub>, the counter generates the INTTBnCCb signal and asserts the TOBnb pin. When the count value matches D<sub>01</sub>, the counter generates the INTTBnCC0 signal, deasserts the TOBnb 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, 1, a = 0 to 3, b = 1 to 3

**(b) Generation timing of compare match interrupt request signal (INTTBnCCb)**

The generation timing of the INTTBnCCb signal in the one-shot pulse output mode is different from INTTBnCCb signals in other mode; the INTTBnCCb signal is generated when the count value of the 16-bit counter matches the value of the TABnCCRb register.



Usually, the INTTBnCCb signal is generated when the 16-bit counter counts up next time after its count value matches the value of the TABnCCRb 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 TOBnb pin.

7.6.5 PWM output mode (TABnMD2 to TABnMD0 bits = 100)

In the PWM output mode, a PWM waveform is output from the TOBn1 to TOBn3 pins when the TABnCTL0.TABnCE bit is set to 1.

In addition, a PWM waveform with a duty factor of 50% with the set value of the TABnCCR0 register + 1 as half its cycle is output from the TOBn0 pin.

Figure 7-28. Configuration in PWM Output Mode

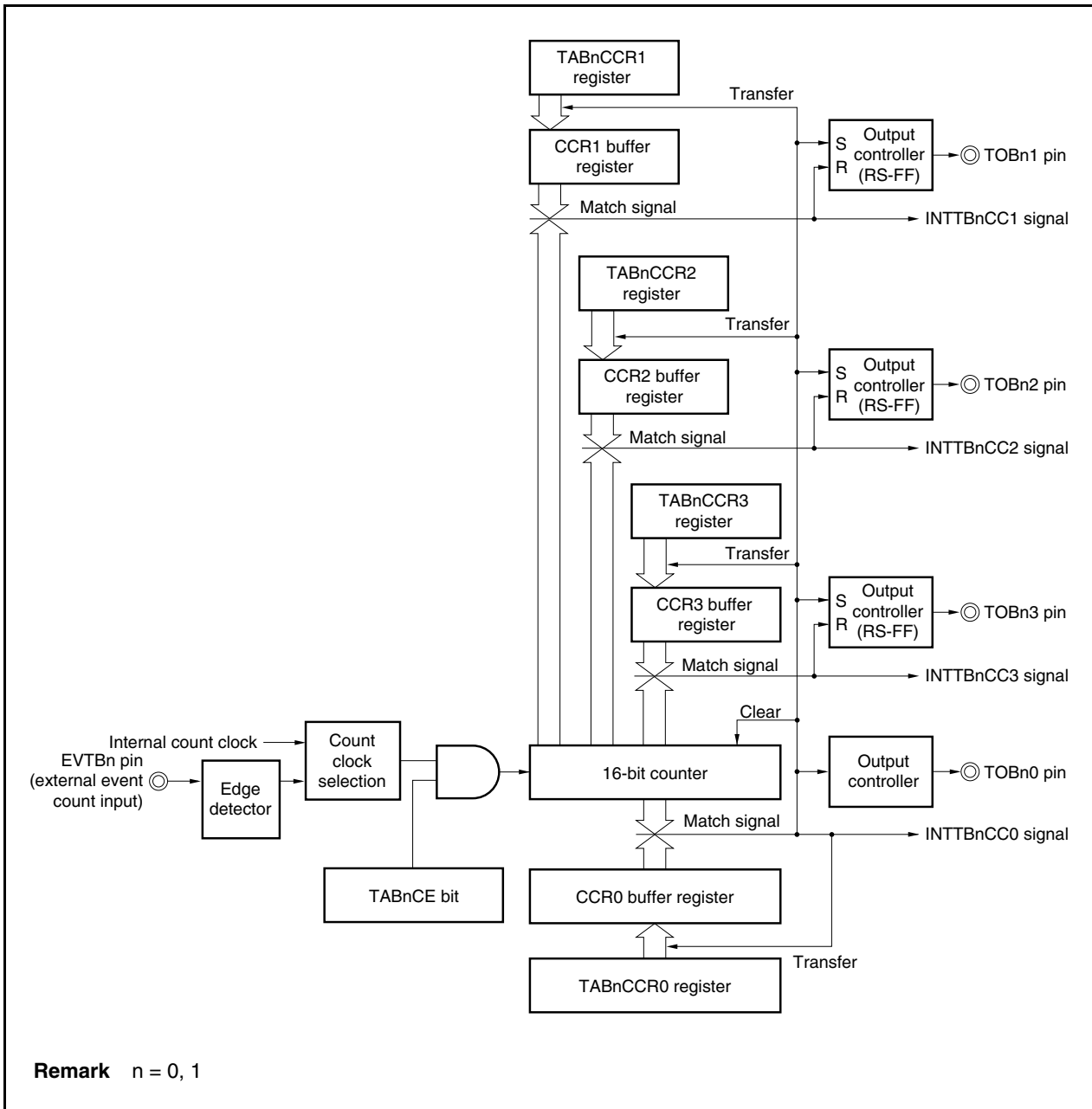
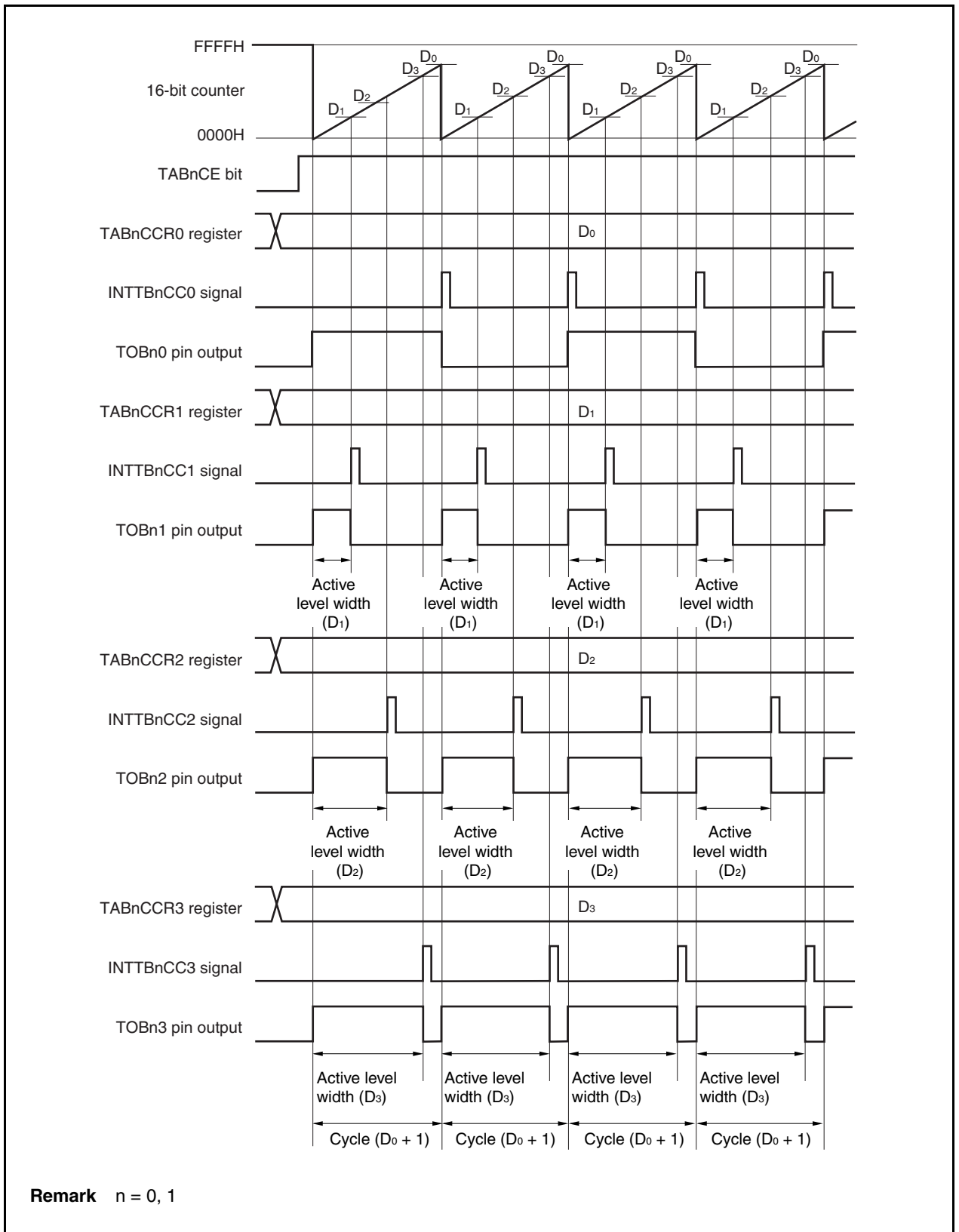




Figure 7-29. Basic Timing in PWM Output Mode



When the TABnCE bit is set to 1, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs PWM waveform from the TOBnb pin.

The active level width, cycle, and duty factor of the PWM waveform can be calculated as follows.

$$\text{Active level width} = (\text{Set value of TABnCCRb register}) \times \text{Count clock cycle}$$

$$\text{Cycle} = (\text{Set value of TABnCCR0 register} + 1) \times \text{Count clock cycle}$$

$$\text{Duty factor} = (\text{Set value of TABnCCRb register}) / (\text{Set value of TABnCCR0 register} + 1)$$

The PWM waveform can be changed by rewriting the TABnCCRa 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 INTTBnCC0 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 INTTBnCCb is generated when the count value of the 16-bit counter matches the value of the CCRb buffer register.

**Remark** n = 0, 1  
a = 0 to 3  
b = 1 to 3

Figure 7-30. Setting of Registers in PWM Output Mode (1/3)

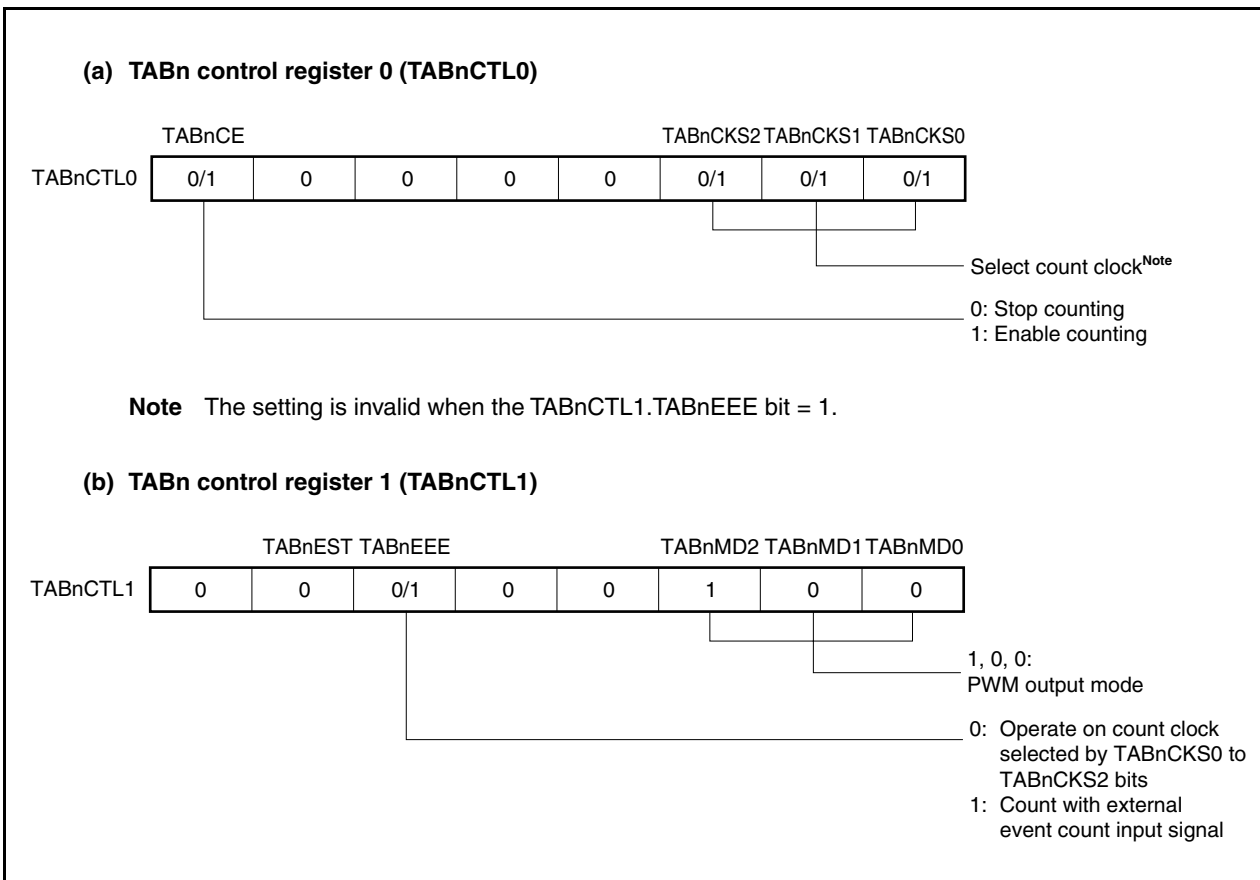


Figure 7-30. Setting of Registers in PWM Output Mode (2/3)

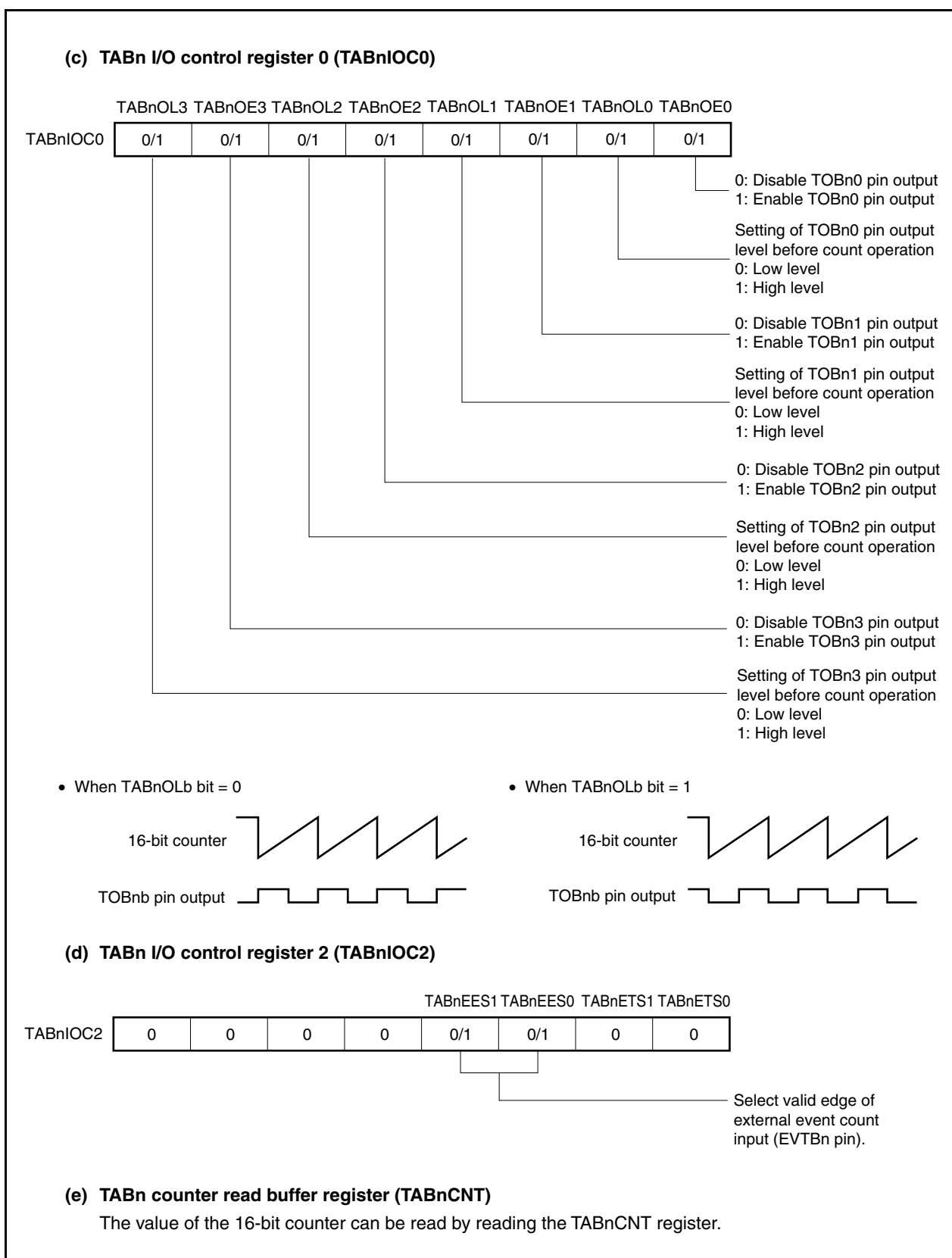


Figure 7-30. Register Setting in PWM Output Mode (3/3)

**(f) TABn capture/compare registers 0 to 3 (TABnCCR0 to TABnCCR3)**

If  $D_0$  is set to the TABnCCR0 register and  $D_b$  to the TABnCCRb register, the cycle and active level of the PWM waveform are as follows.

PWM waveform cycle =  $(D_0 + 1) \times$  Count clock cycle

PWM waveform active level width =  $D_b \times$  Count clock cycle

**Remarks 1.** TABn I/O control register 1 (TABnIOC1) and TABn option register 0 (TABnOPT0) are not used in the PWM output mode.

**2.**  $n = 0, 1$

$b = 1$  to 3

(1) Operation flow in PWM output mode

Figure 7-31. Software Processing Flow in PWM Output Mode (1/2)

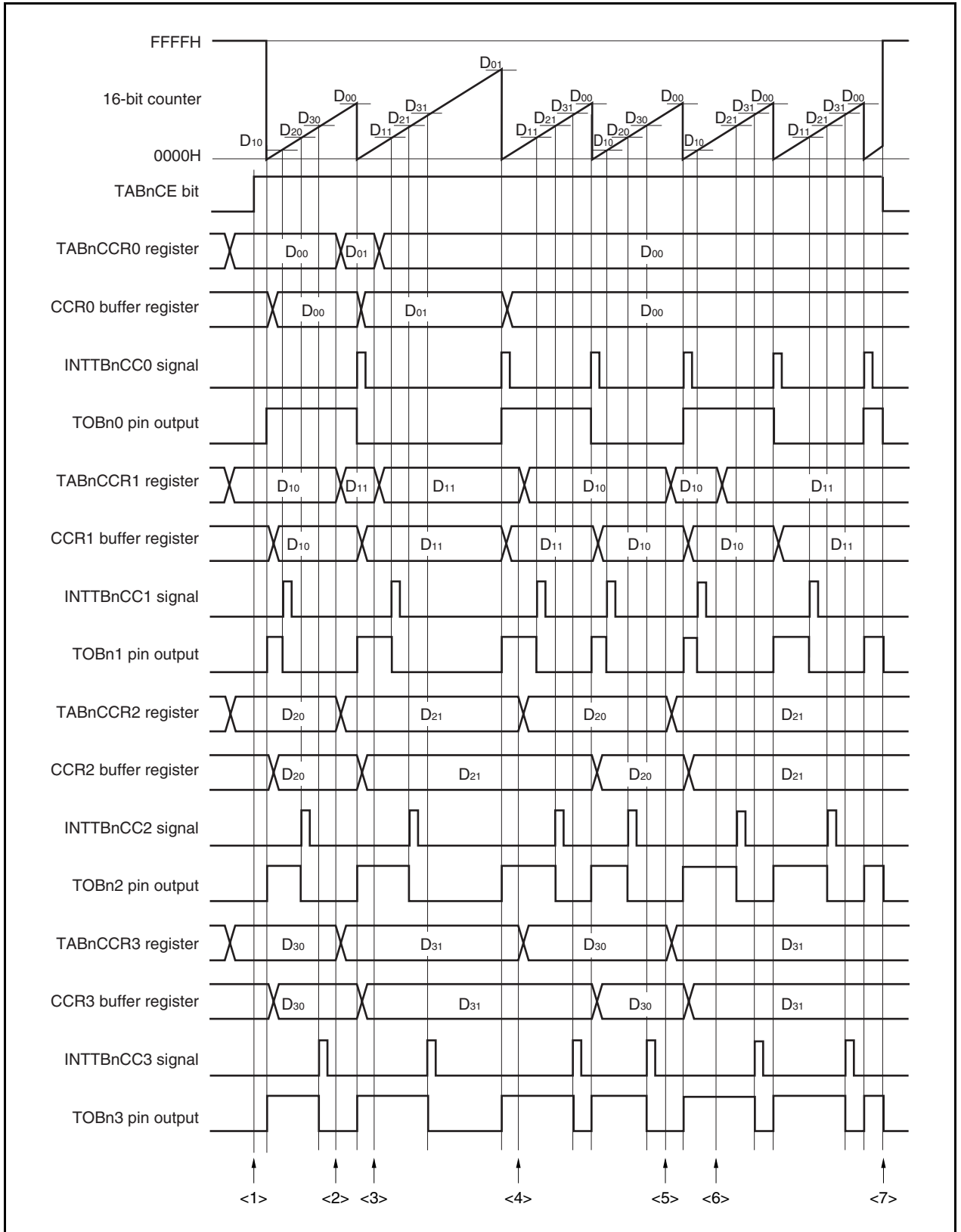
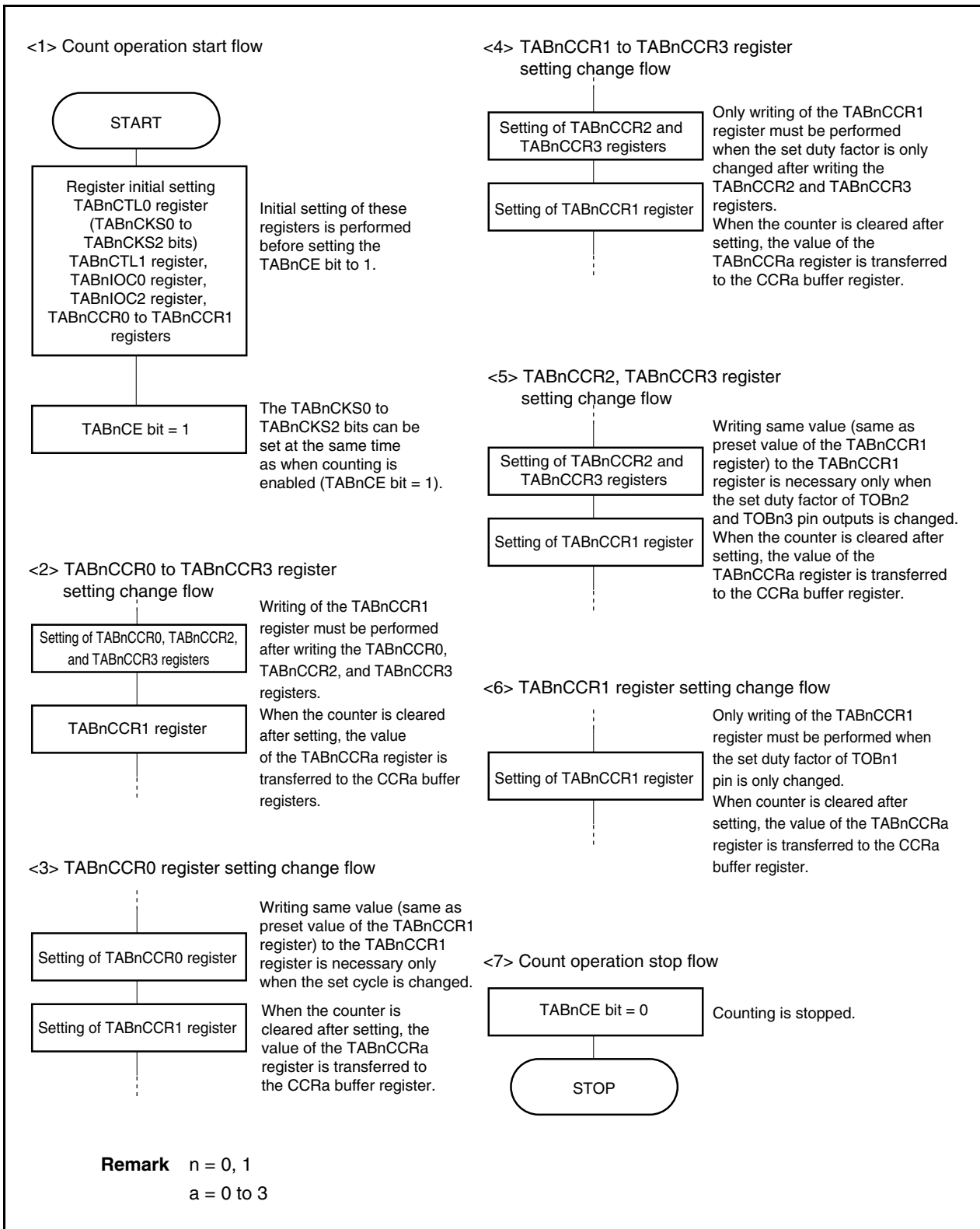


Figure 7-31. Software Processing Flow in PWM Output Mode (2/2)

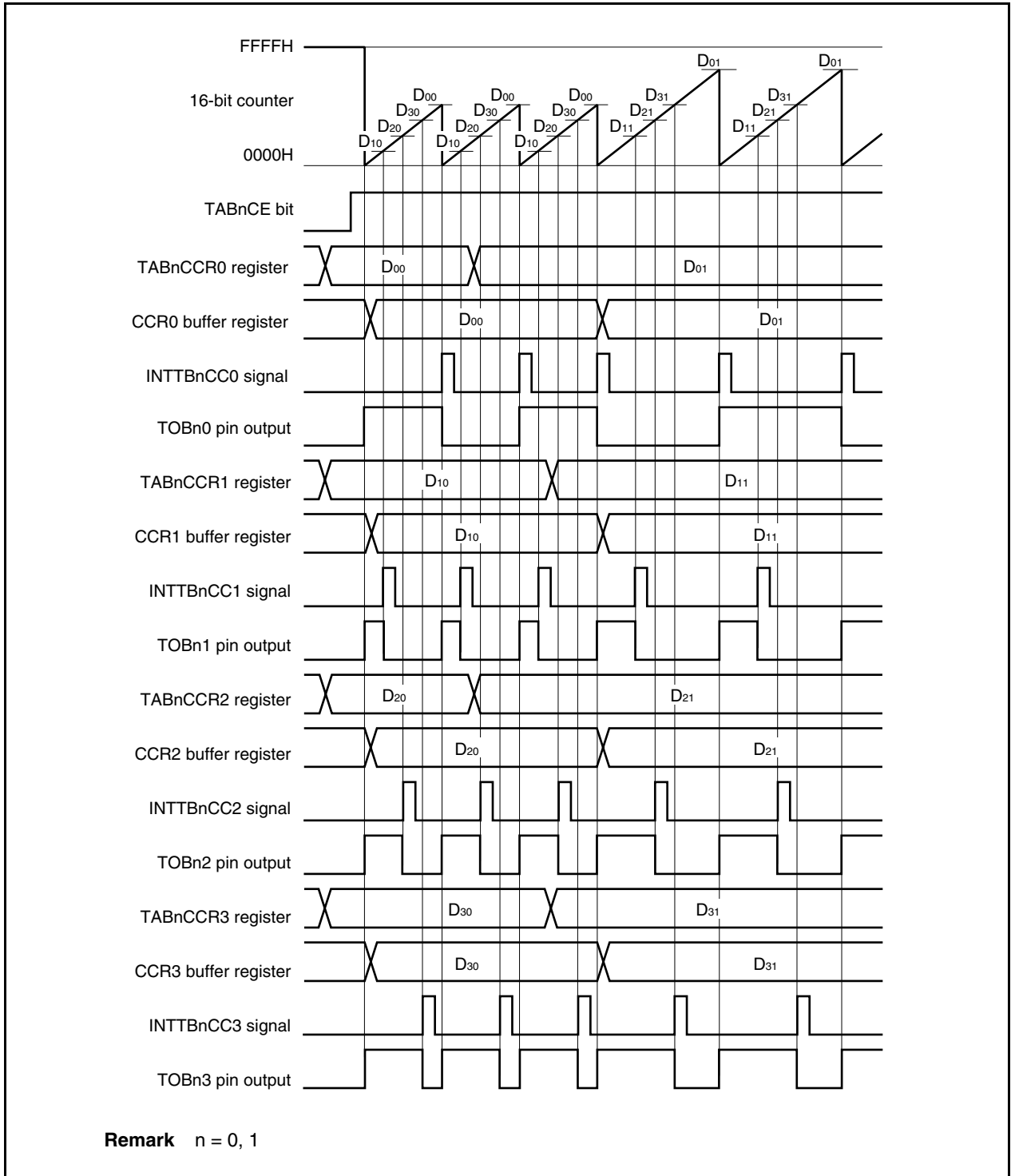


(2) PWM output mode operation timing

(a) Changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TABnCCR1 register last.

Rewrite the TABnCCRa register after writing the TABnCCR1 register after the INTTBnCC0 signal is detected.



To transfer data from the TABnCCRa register to the CCRa buffer register, the TABnCCR1 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 TABnCCR0 register, set the active level width to the TABnCCR2 and TABnCCR3 registers, and then set an active level width to the TABnCCR1 register.

To change only the cycle of the PWM waveform, first set a cycle to the TABnCCR0 register, and then write the same value (same as preset value of the TABnCCR1 register) to the TABnCCR1 register.

To change only the active level width (duty factor) of PWM waveform, first set the active level to the TABnCCR2 and TABnCCR3 registers, and then set an active level to the TABnCCR1 register.

To change only the active level width (duty factor) of the PWM waveform output by the TOBn1 pin, only the TABnCCR1 register has to be set.

To change only the active level width (duty factor) of the PWM waveform output by the TOBn2 and TOBn3 pins, first set an active level width to the TABnCCR2 and TABnCCR3 registers, and then write the same value (same as preset value of the TABnCCR1 register) to the TABnCCR1 register.

After the TABnCCR1 register is written, the value written to the TABnCCRa register is transferred to the CCRa buffer register in synchronization with the timing of clearing the 16-bit counter, and is used as a value to be compared with the value of the 16-bit counter.

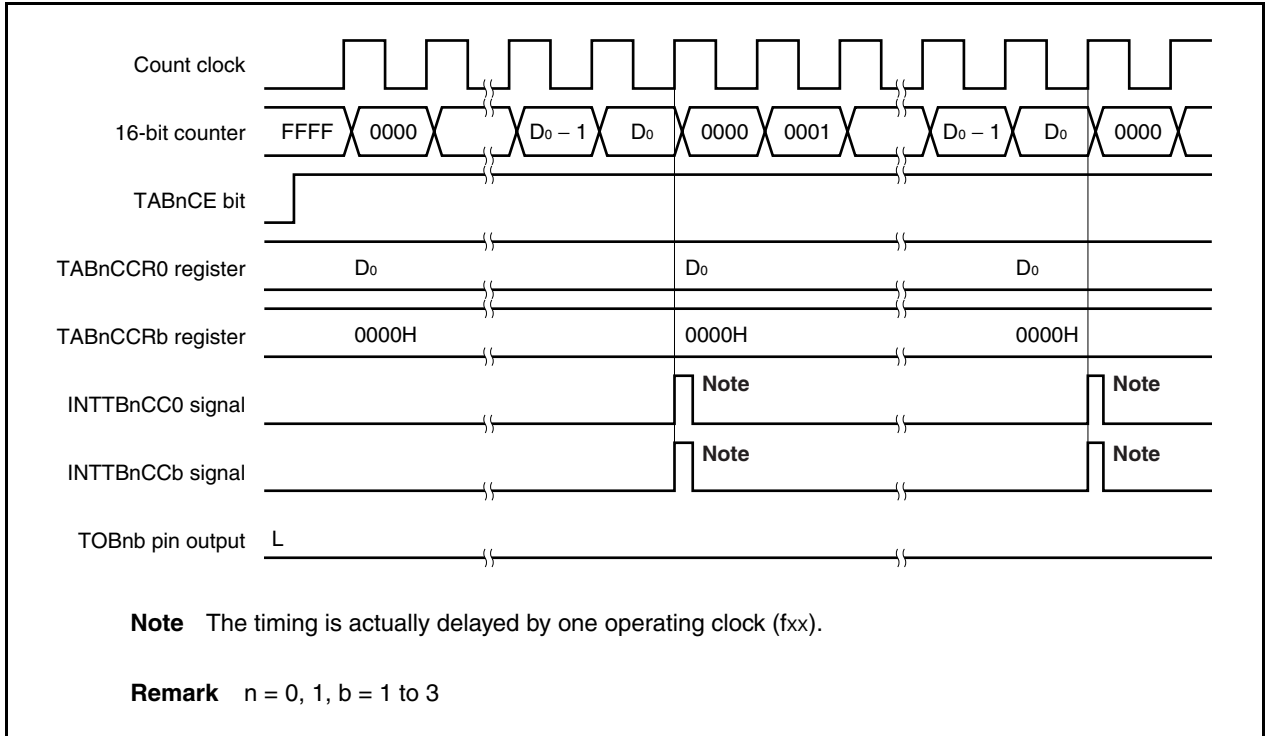
To write the TABnCCR0 to TABnCCR3 registers again after writing the TABnCCR1 register once, do so after the INTTBnCC0 signal is generated. Otherwise, the value of the CCRa buffer register may become undefined because the timing of transferring data from the TABnCCRa register to the CCRa buffer register conflicts with writing the TABnCCRa register.

**Remark** n = 0, 1  
a = 0 to 3

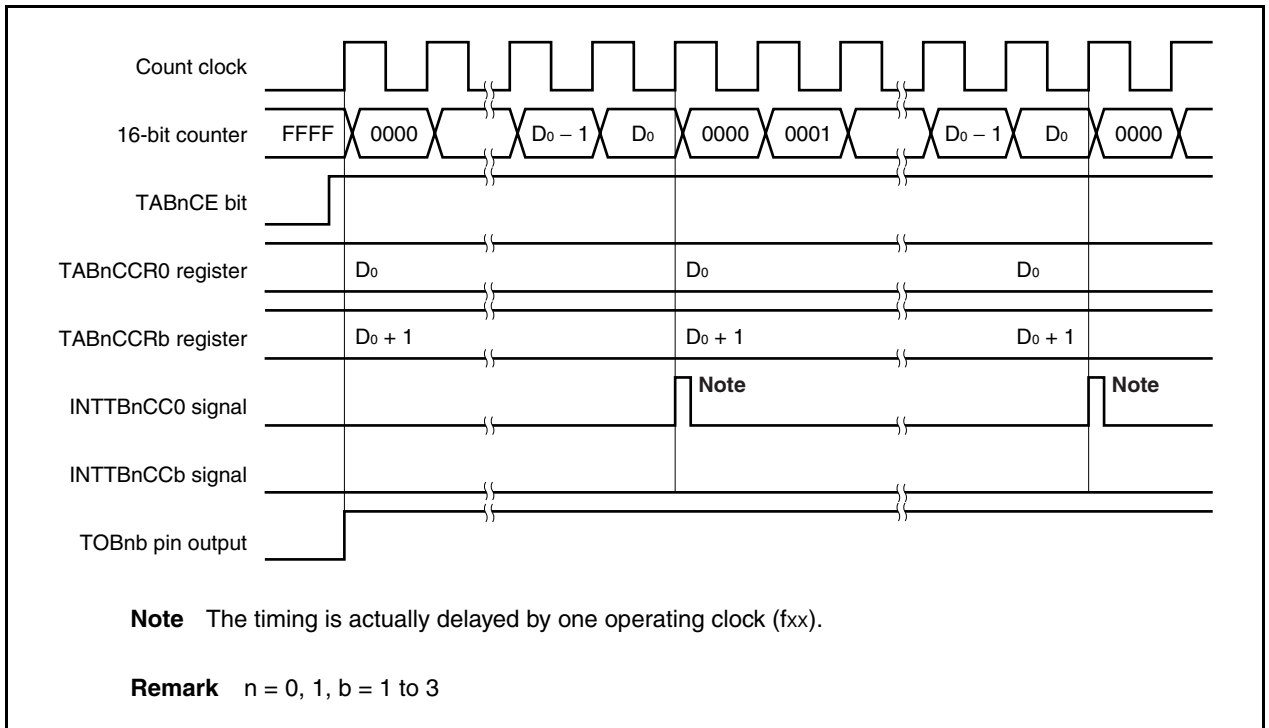


**(b) 0%/100% output of PWM waveform**

To output a 0% waveform, set the TABnCCRb register to 0000H. The 16-bit counter is cleared to 0000H and the INTTBnCC0 and INTTBnCCb signals are generated at the next timing after a match between the count value of the 16-bit counter and the value of the CCR0 buffer register.

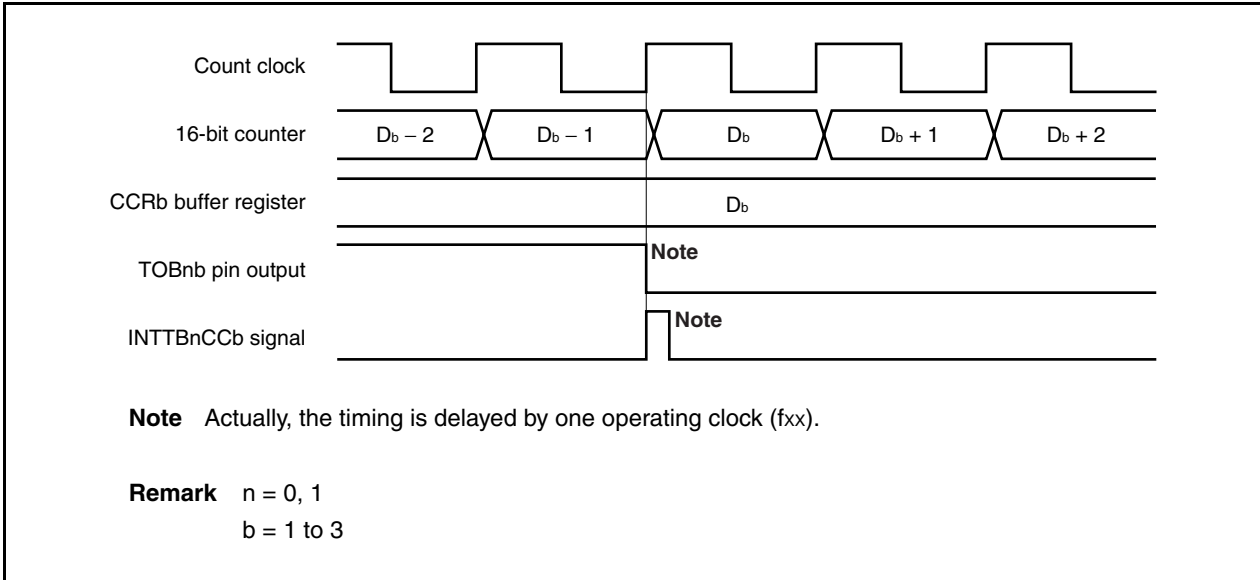


To output a 100% waveform, set a value of (set value of TABnCCR0 register + 1) to the TABnCCRb register. If the set value of the TABnCCR0 register is FFFFH, 100% output cannot be produced.



**(c) Generation timing of compare match interrupt request signal (INTTBnCCb)**

The timing of generation of the INTTBnCCb signal in the PWM output mode differs from the timing of INTTBnCCb signals in other mode; the INTTBnCCb signal is generated when the count value of the 16-bit counter matches the value of the TABnCCRb register.

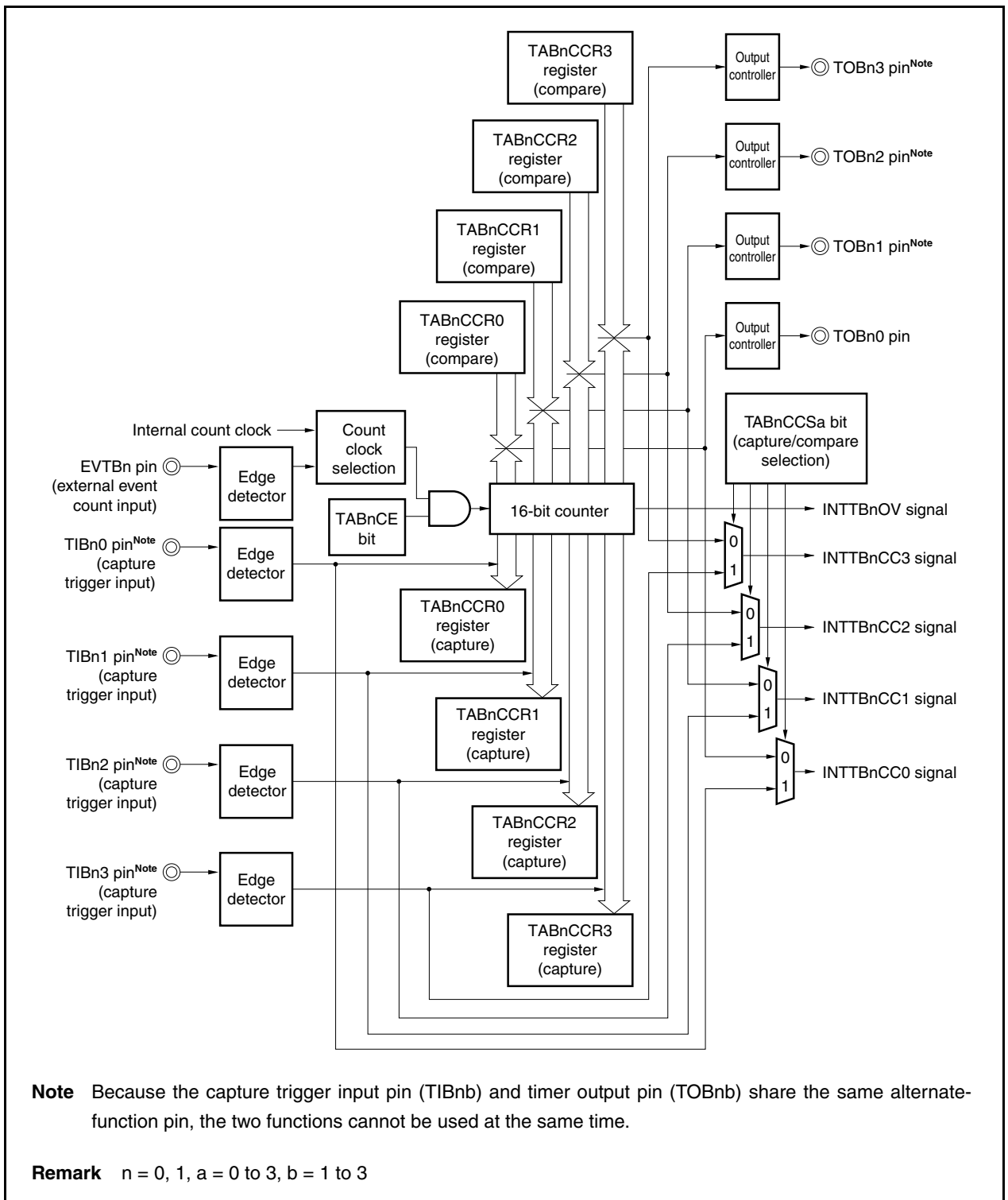


Usually, the INTTBnCCb signal is generated in synchronization with the next counting up after the count value of the 16-bit counter matches the value of the TABnCCRb register. In the PWM output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the output signal of the TOBnb pin.

7.6.6 Free-running timer mode (TABnMD2 to TABnMD0 bits = 101)

In the free-running timer mode, 16-bit timer/event counter AB starts counting when the TABnCTL0.TABnCE bit is set to 1. At this time, the TABnCCRa register can be used as a compare register or a capture register, depending on the setting of the TABnOPT0.TABnCCSa bit.

Figure 7-32. Configuration in Free-Running Timer Mode



- Compare operation

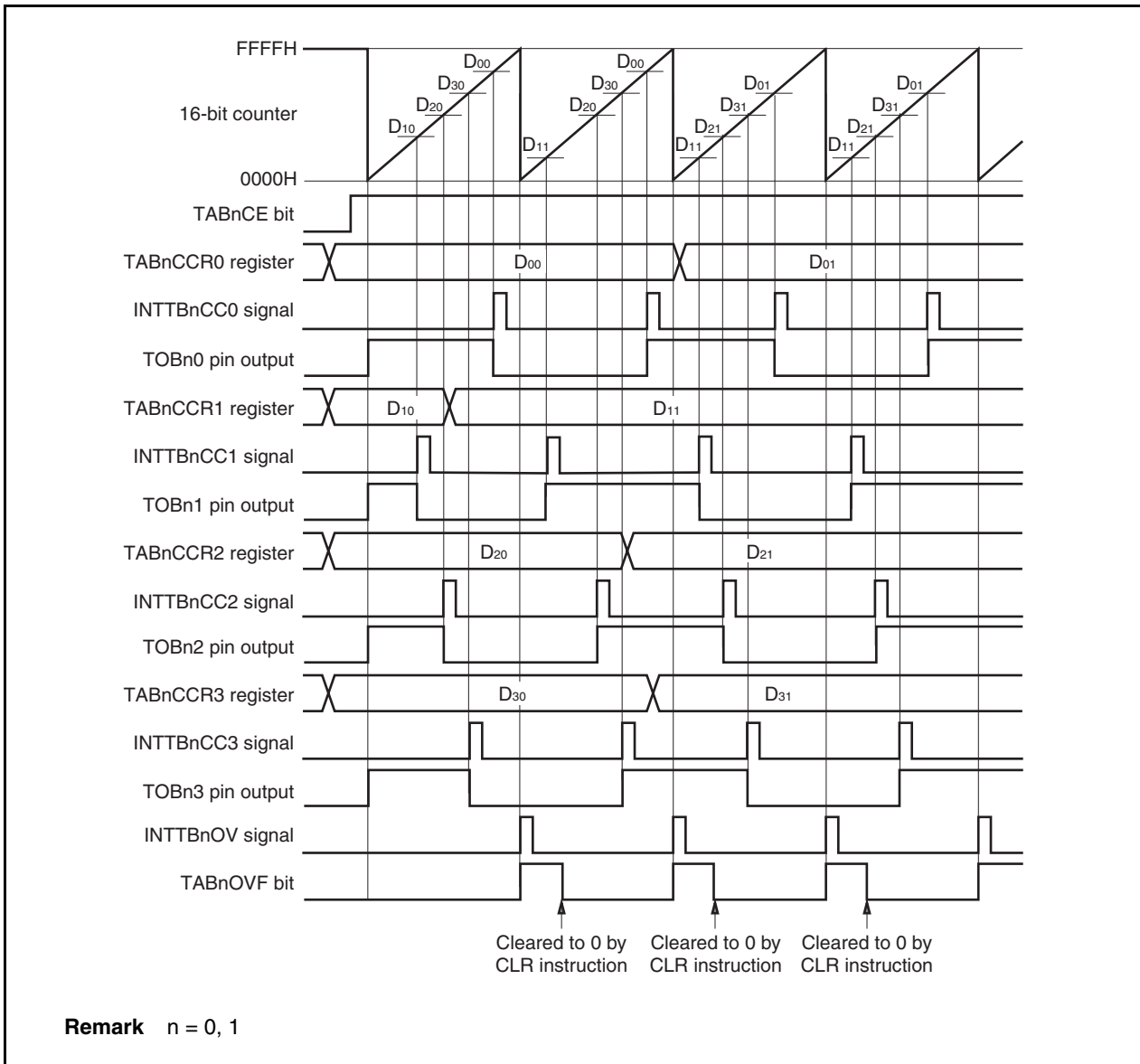
When the TABnCE bit is set to 1, 16-bit timer/event counter AB starts counting, and the output signals of the TOBn0 to TOBn3 pins are inverted. When the count value of the 16-bit counter later matches the set value of the TABnCCRa register, a compare match interrupt request signal (INTTBnCCa) is generated, and the output signals of the TOBn0 to TOBn3 pins are 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 (INTTBnOV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TABnOPT0.TABnOVF bit) is also set to 1. Confirm that the overflow flag is set to 1 and then clear it to 0 by executing the CLR instruction via software.

The TABnCCRa 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.

**Remark** n = 0, 1  
a = 0 to 3

**Figure 7-33. Basic Timing in Free-Running Timer Mode (Compare Function)**



- Capture operation

When the TABnCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIBna pin is detected, the count value of the 16-bit counter is stored in the TABnCCRa register, and a capture interrupt request signal (INTTBnCCa) 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 (INTTBnOV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TABnOPT0.TABnOVF bit) is also set to 1. Confirm that the overflow flag is set to 1 and then clear it to 0 by executing the CLR instruction via software.

**Figure 7-34. Basic Timing in Free-Running Timer Mode (Capture Function)**

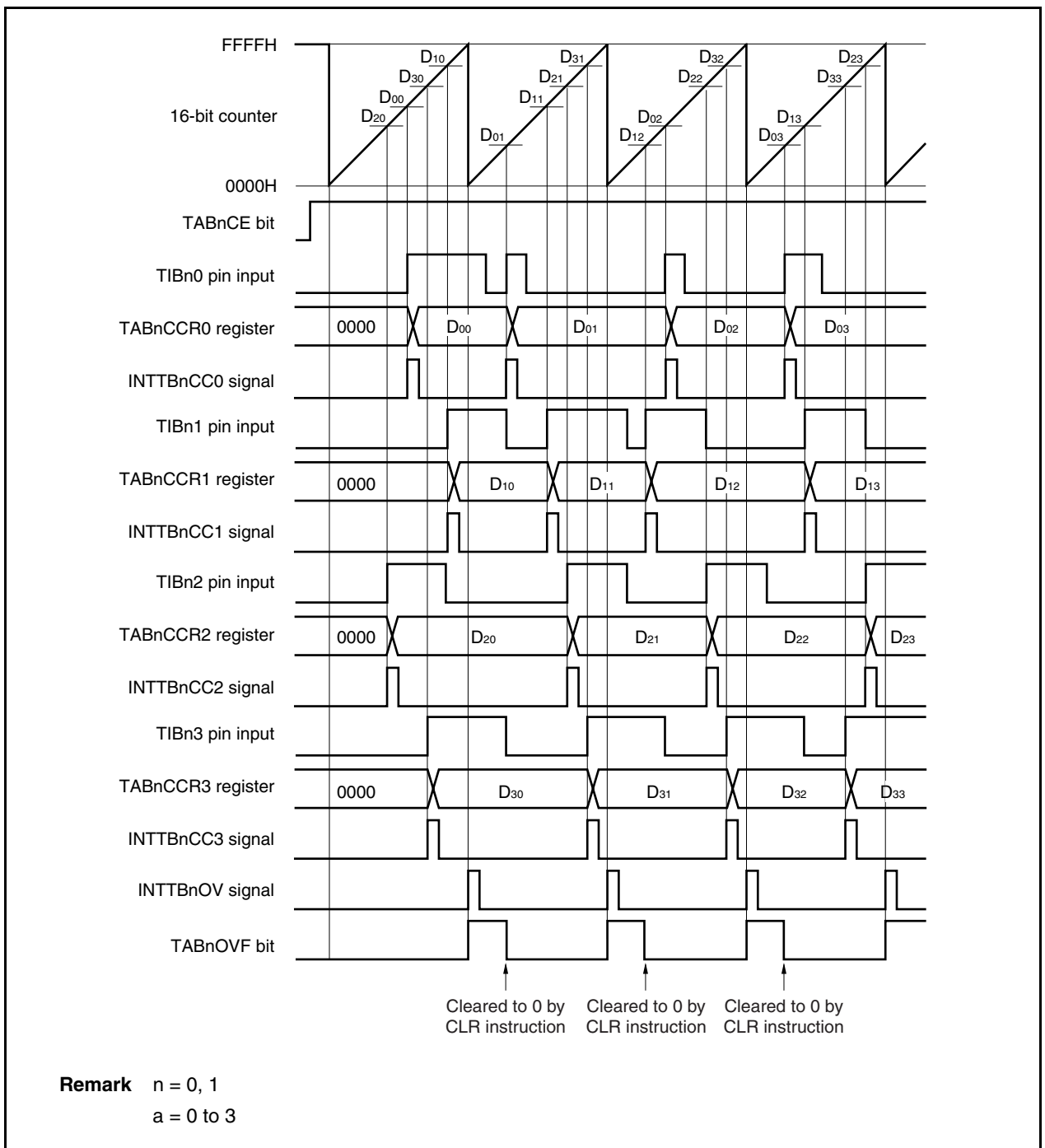


Figure 7-35. Register Setting in Free-Running Timer Mode (1/3)

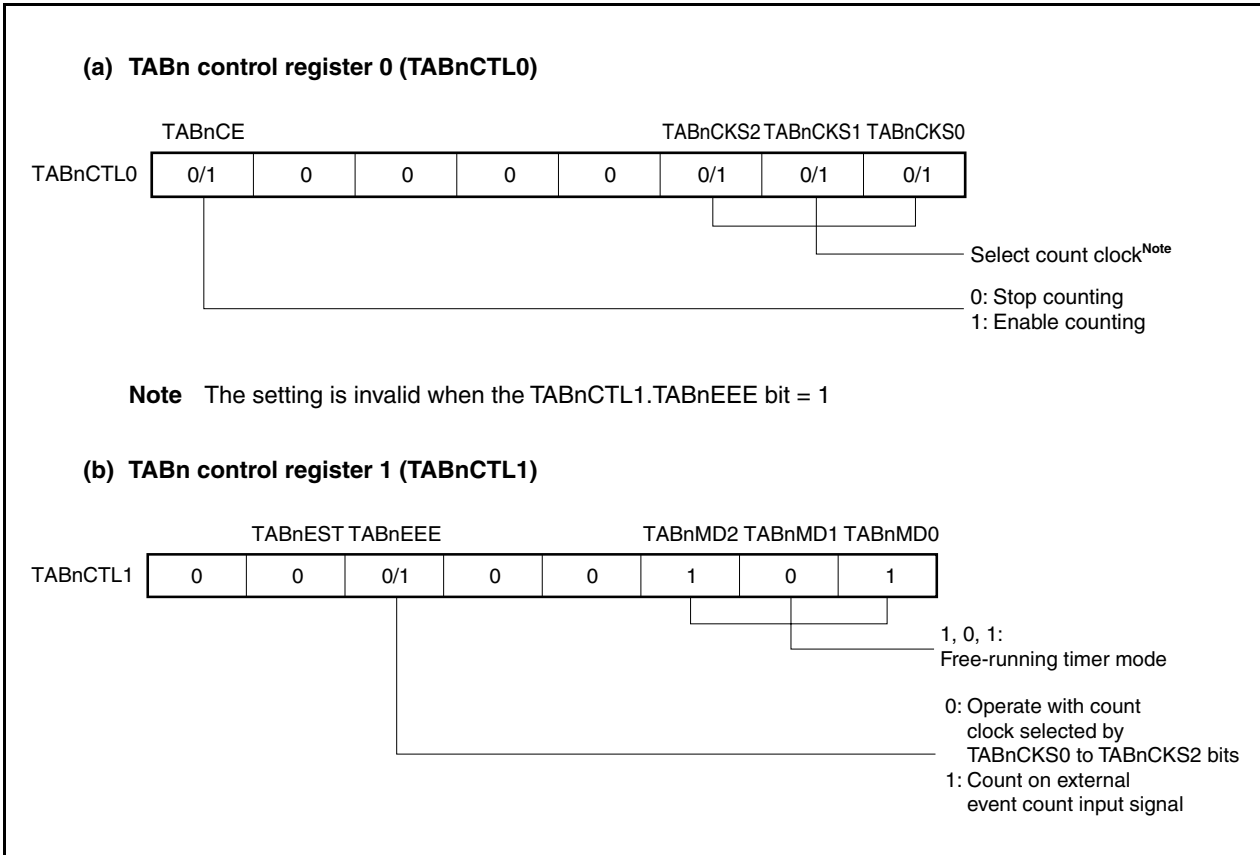


Figure 7-35. Register Setting in Free-Running Timer Mode (2/3)

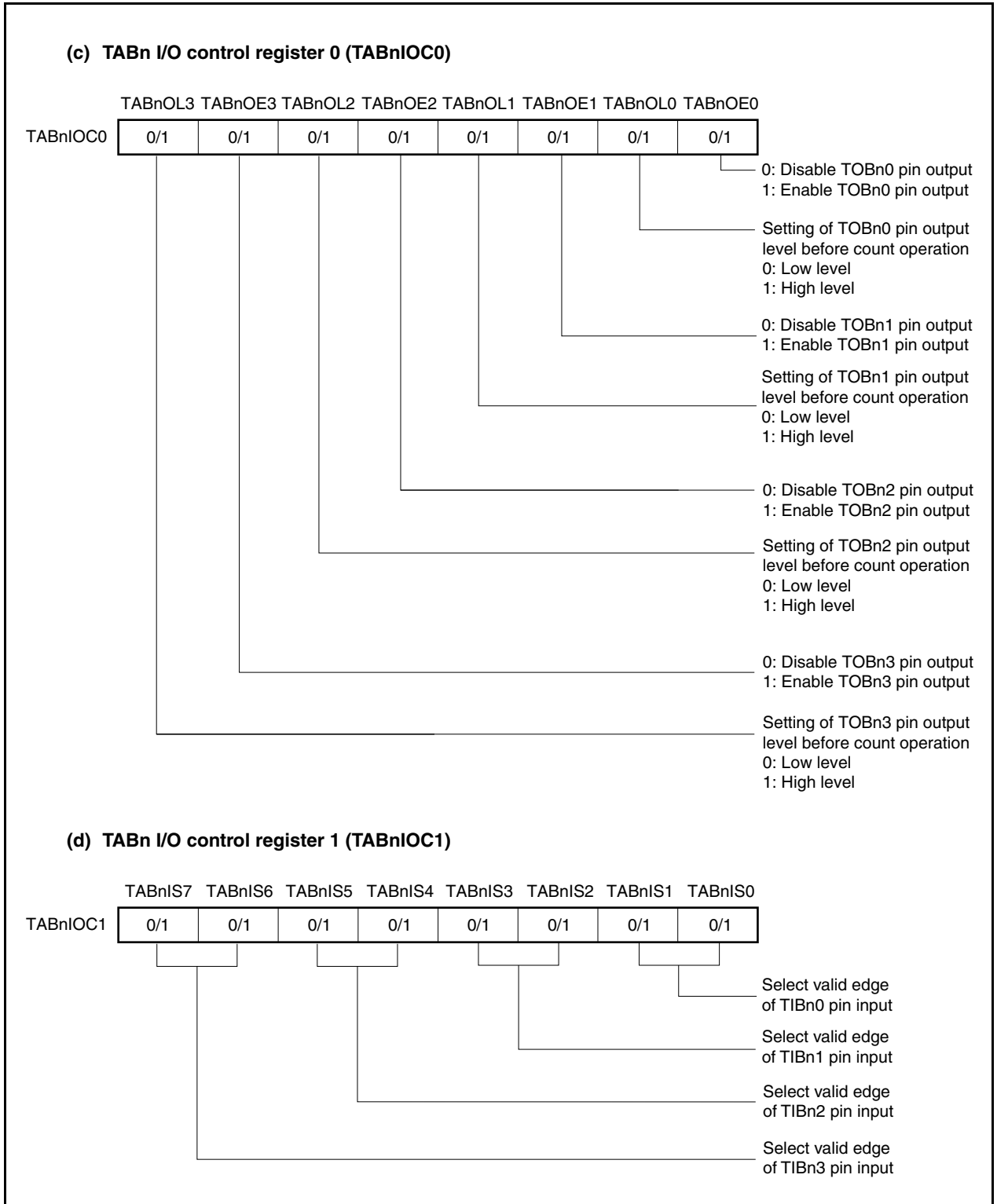
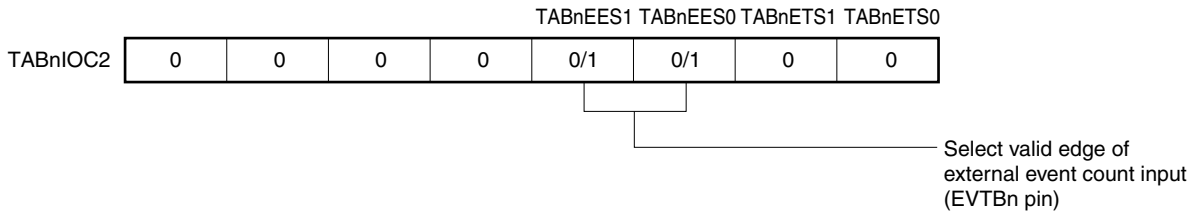
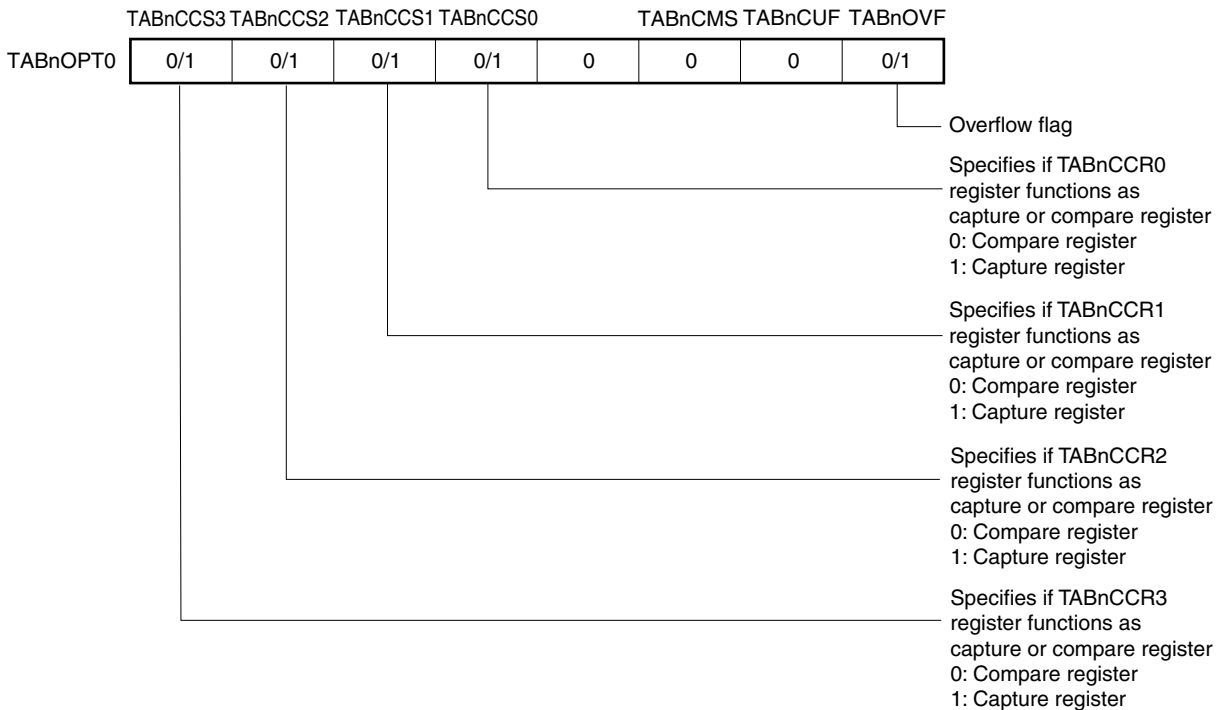


Figure 7-35. Register Setting in Free-Running Timer Mode (3/3)

(e) TABn I/O control register 2 (TABnIOC2)



(f) TABn option register 0 (TABnOPT0)



(g) TABn counter read buffer register (TABnCNT)

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

(h) TABn capture/compare registers 0 to 3 (TABnCCR0 to TABnCCR3)

These registers function as capture registers or compare registers depending on the setting of the TABnOPT0.TABnCCSa bit.

When the registers function as capture registers, they store the count value of the 16-bit counter when the valid edge input to the TIBna pin is detected.

When the registers function as compare registers and when Da is set to the TABnCCRa register, the INTTBnCCa signal is generated when the counter reaches (Da + 1), and the output signals of the TOBn0 to TOBn3 pins are inverted.

**Remark** n = 0, 1  
a = 0 to 3



(1) Operation flow in free-running timer mode

(a) When using capture/compare register as compare register

Figure 7-36. Software Processing Flow in Free-Running Timer Mode (Compare Function) (1/2)

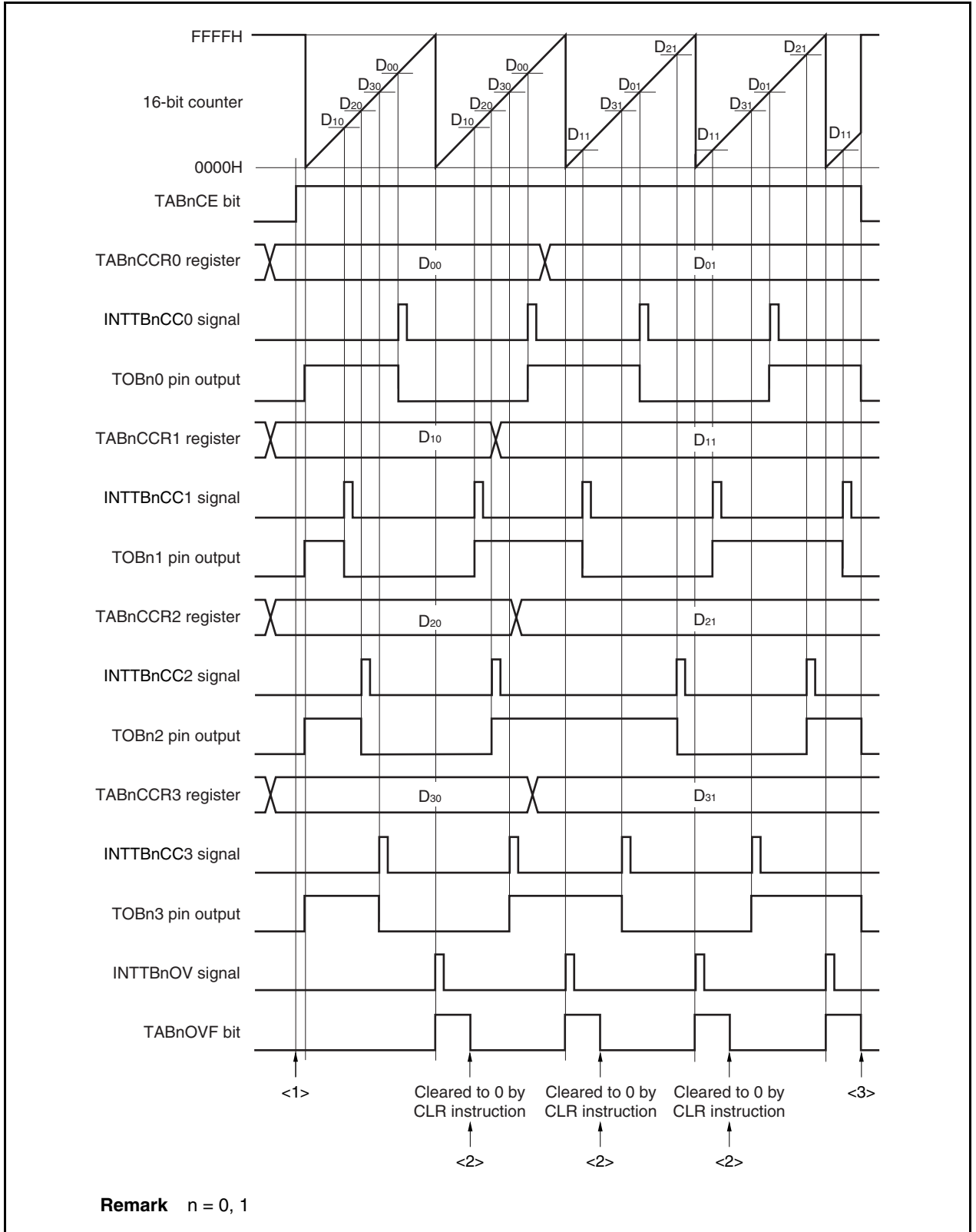
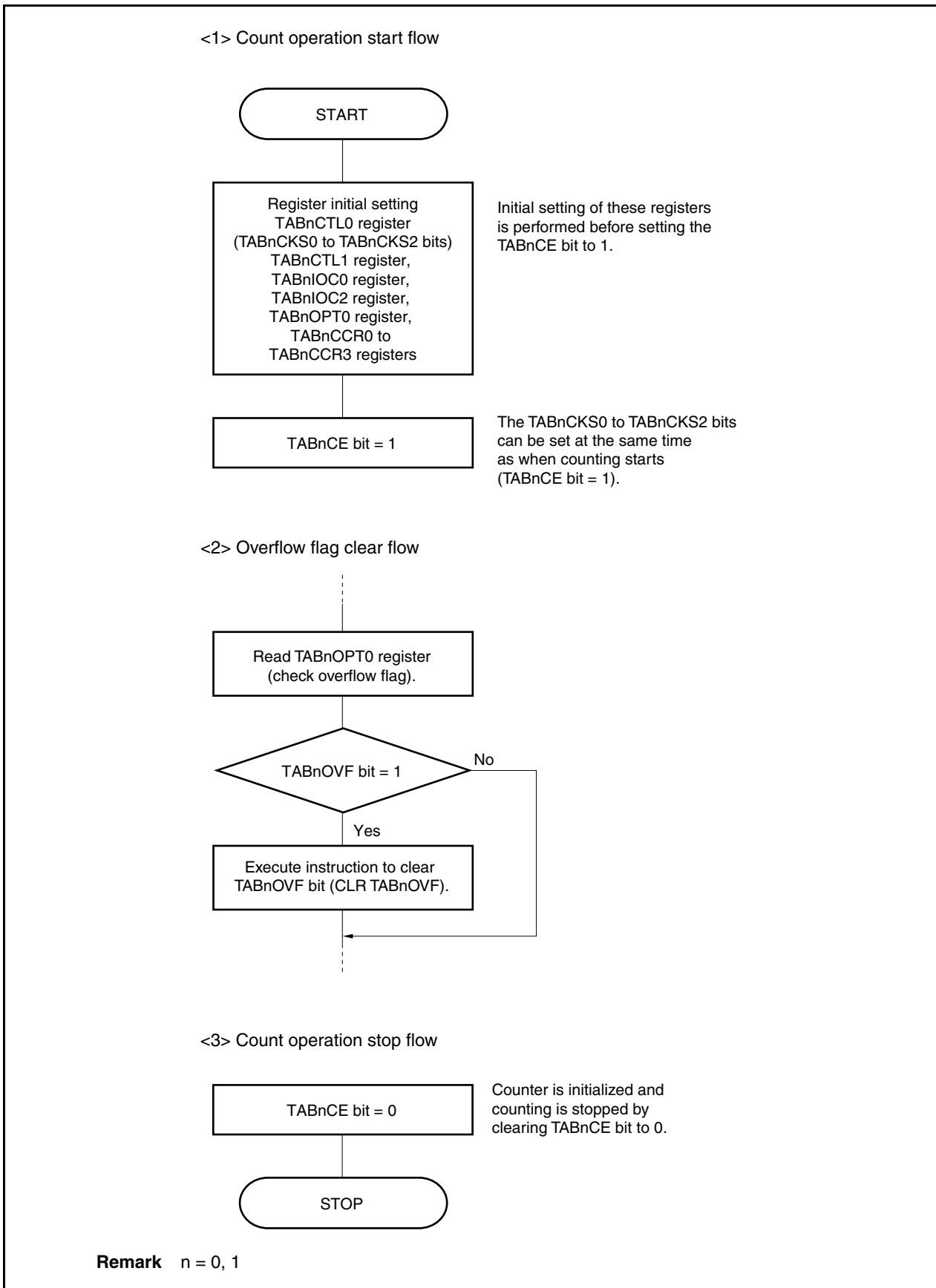


Figure 7-36. Software Processing Flow in Free-Running Timer Mode (Compare Function) (2/2)



(b) When using capture/compare register as capture register

Figure 7-37. Software Processing Flow in Free-Running Timer Mode (Capture Function) (1/2)

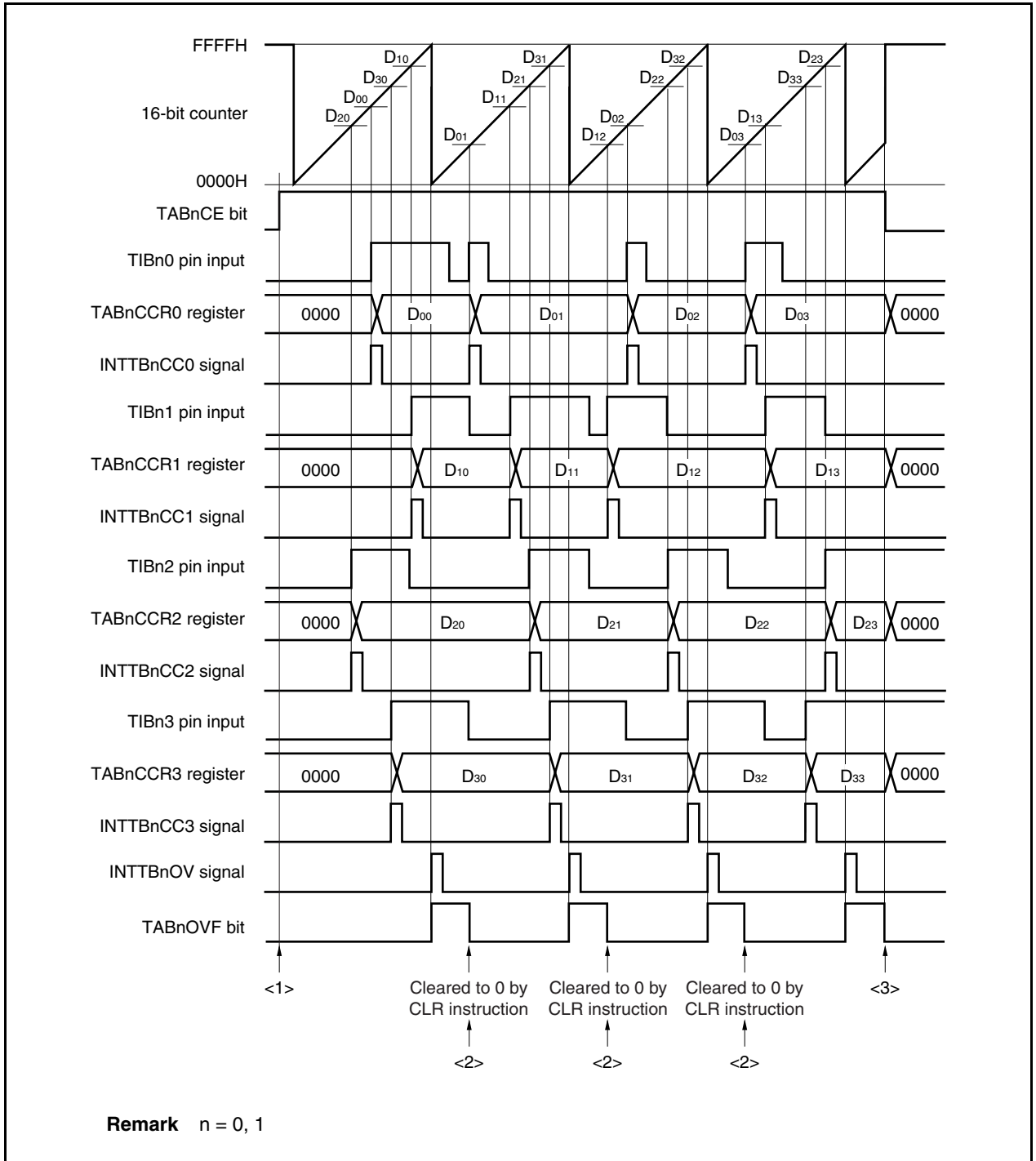
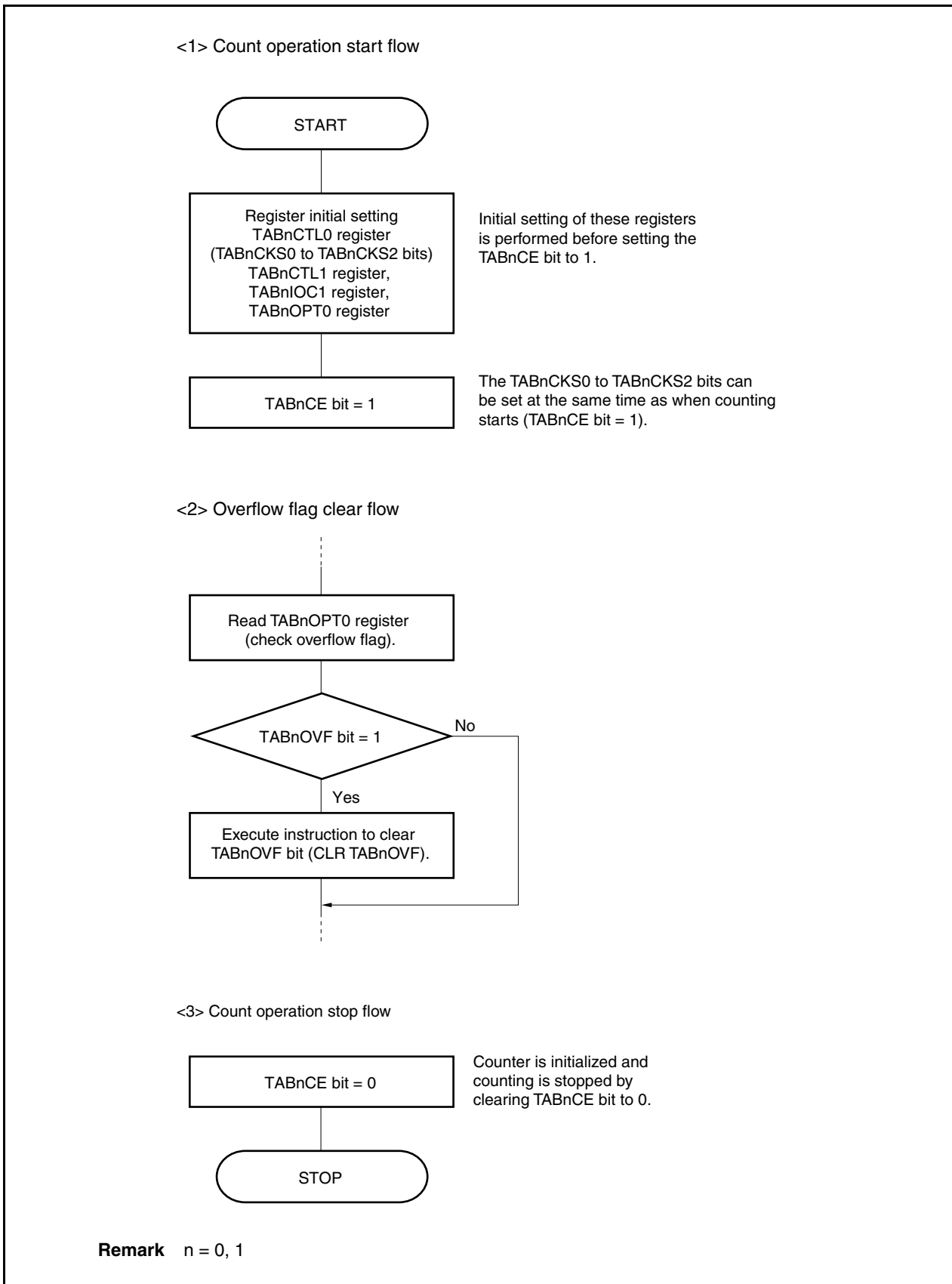


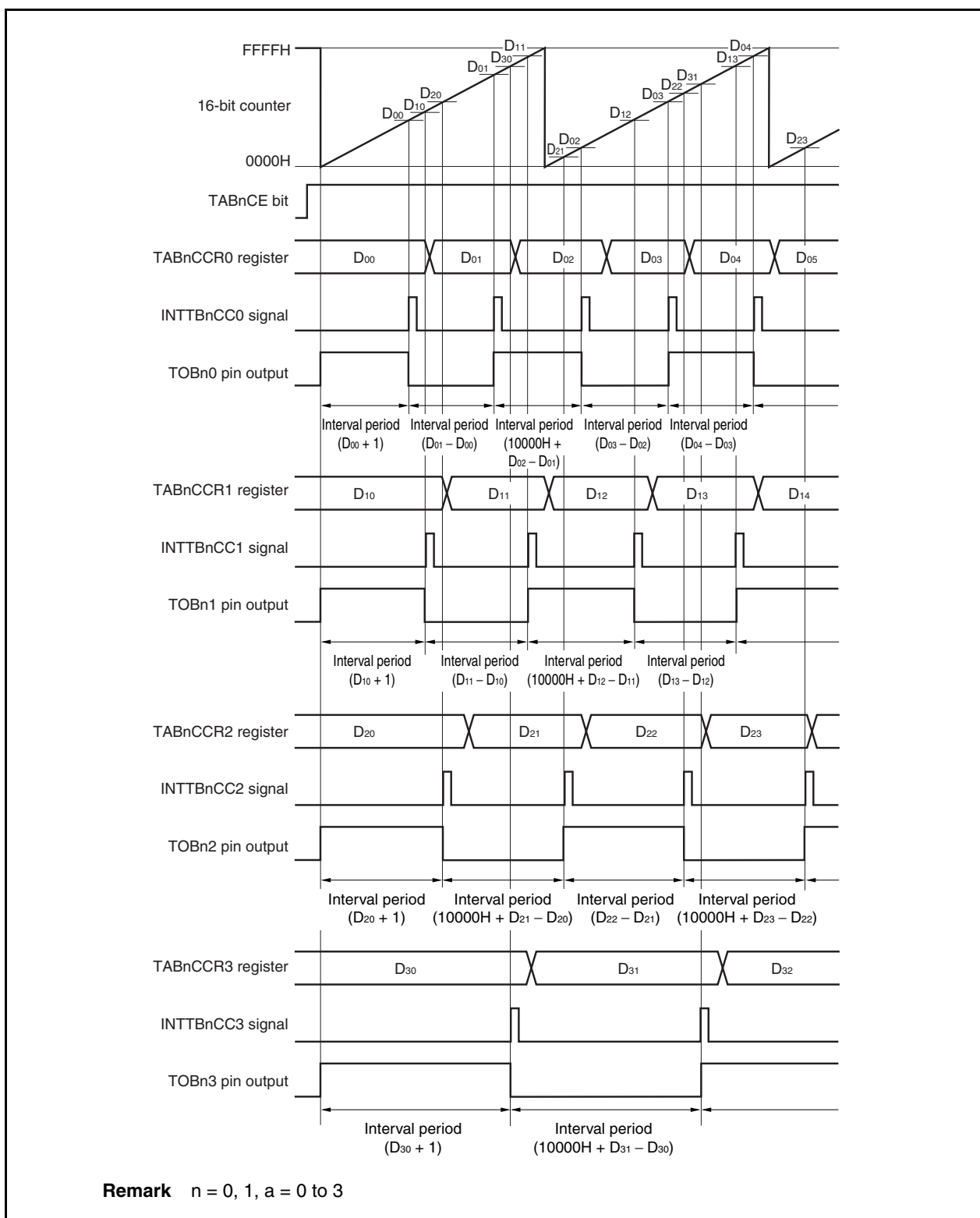
Figure 7-37. Software Processing Flow in Free-Running Timer Mode (Capture Function) (2/2)



(2) Operation timing in free-running timer mode

(a) Interval operation with compare register

When 16-bit timer/event counter AB is used as an interval timer with the TABnCCRa 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 INTTBnCCa signal has been detected.



When performing an interval operation in the free-running timer mode, four intervals can be set with one channel.

To perform the interval operation, the value of the corresponding TABnCCRa register must be re-set in the interrupt servicing that is executed when the INTTBnCCa signal is detected.

The set value for re-setting the TABnCCRa register can be calculated by the following expression, where “Da” is the interval period.

Compare register default value:  $D_a - 1$

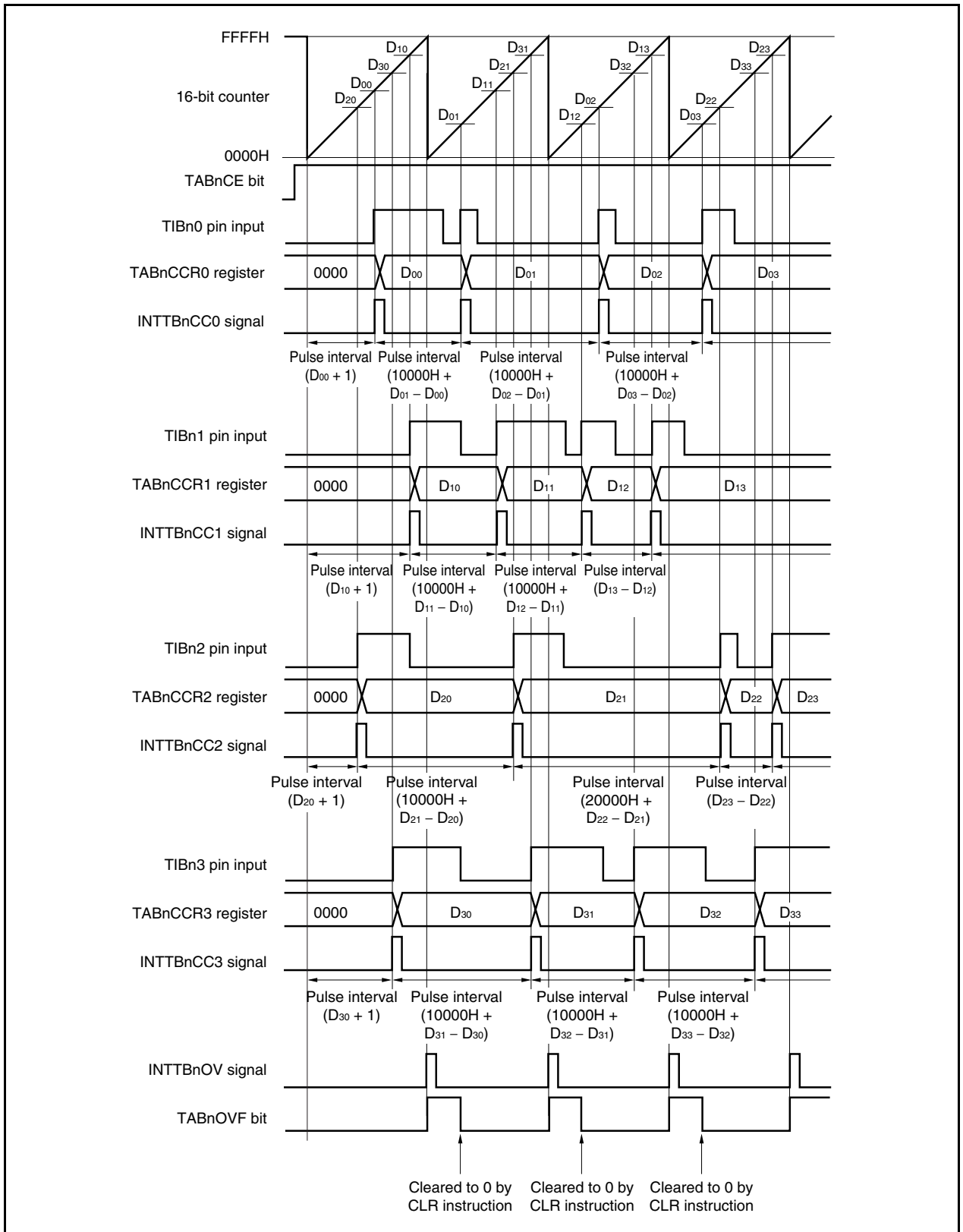
Value set to compare register second and subsequent time: Previous set value +  $D_a$

(If the calculation result is greater than FFFFH, subtract 10000H from the result and set this value to the register.)

**Remark**     $n = 0, 1$   
               $a = 0 \text{ to } 3$

**(b) Pulse width measurement with capture register**

When pulse width measurement is performed with the TABnCCRa register used as a capture register, software processing is necessary for reading the capture register each time the INTTBnCCa 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.

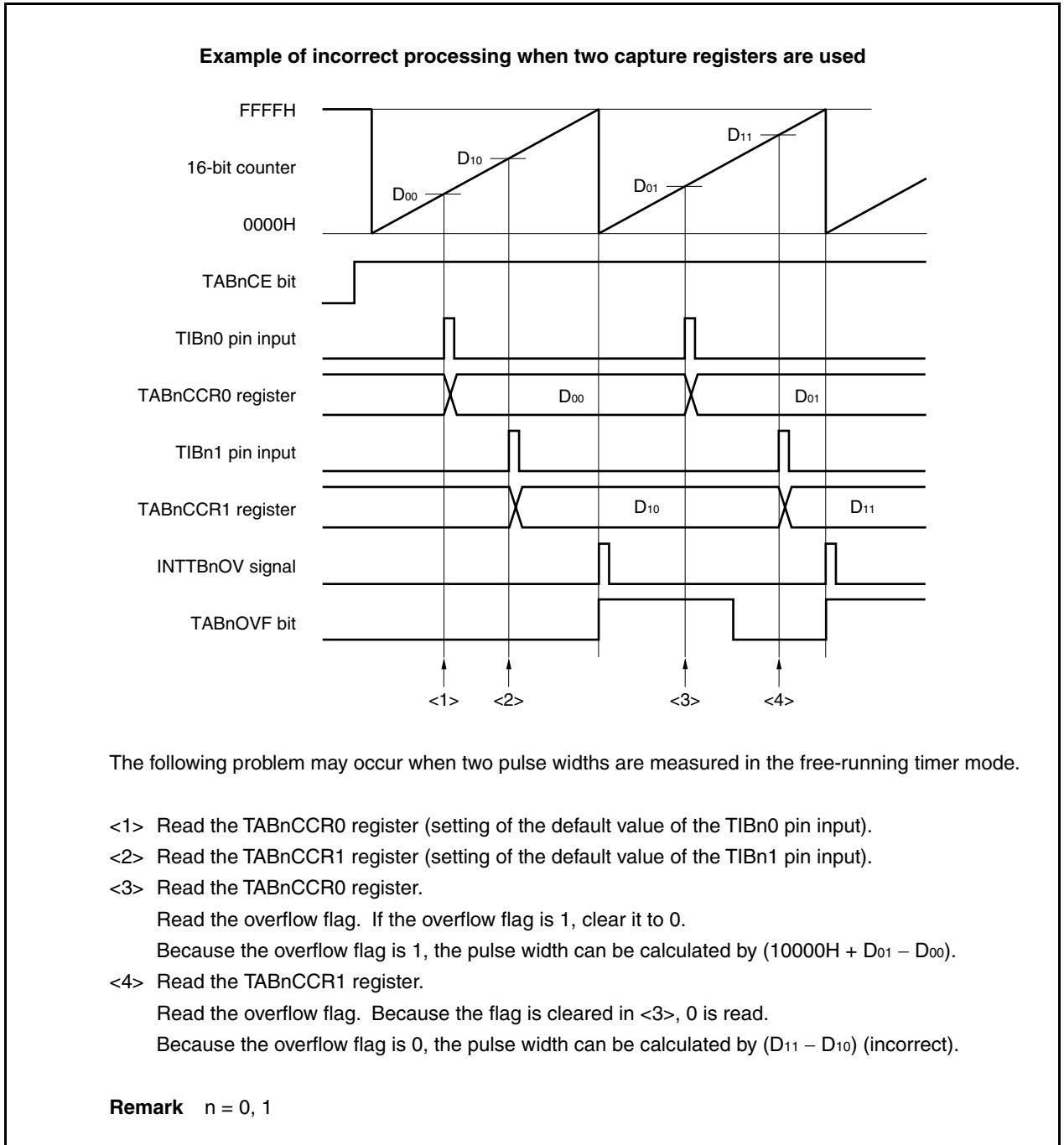
To measure a pulse width, the pulse width can be calculated by reading the value of the TABnCCRa register in synchronization with the INTTBnCCa signal, and calculating the difference between the read value and the previously read value.

**Remark** n = 0, 1  
a = 0 to 3



(c) Processing of overflow when two capture registers are used

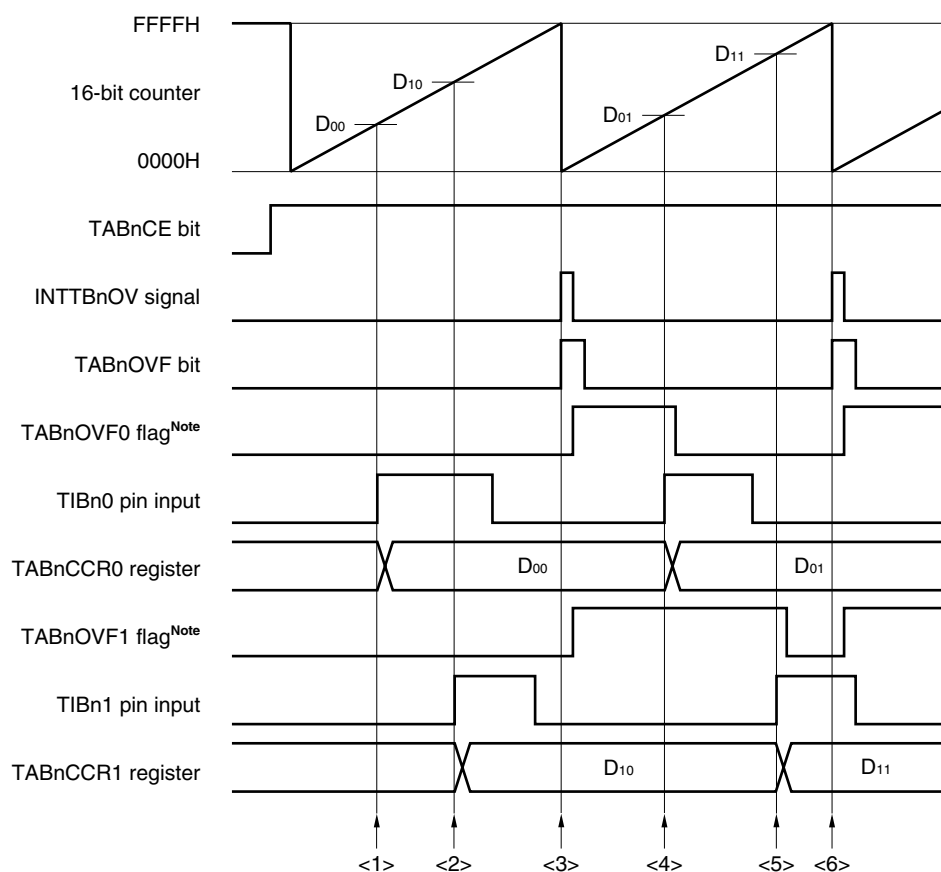
Care must be exercised in processing the overflow flag when two capture registers are used. First, an example of incorrect processing is shown below.



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.

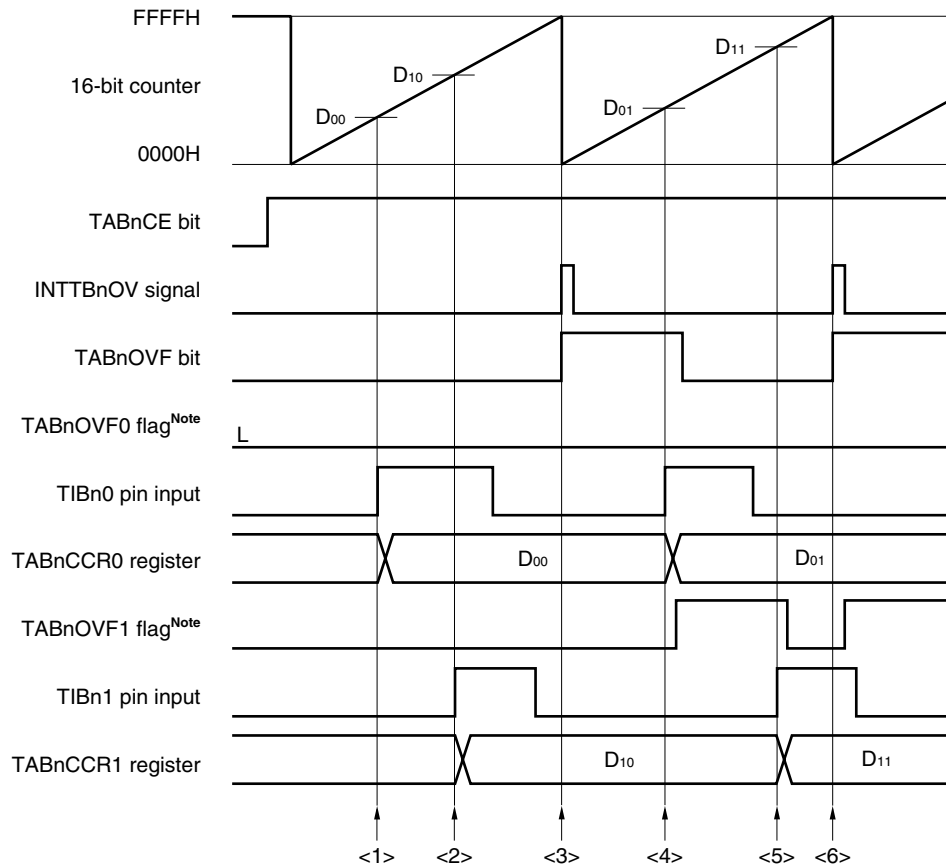
### Example when two capture registers are used (using overflow interrupt)



**Note** The TABnOVF0 and TABnOVF1 flags are set on the internal RAM by software.

- <1> Read the TABnCCR0 register (setting of the default value of the TIBn0 pin input).
- <2> Read the TABnCCR1 register (setting of the default value of the TIBn1 pin input).
- <3> An overflow occurs. Set the TABnOVF0 and TABnOVF1 flags to 1 in the overflow interrupt servicing, and clear the overflow flag to 0.
- <4> Read the TABnCCR0 register.  
Read the TABnOVF0 flag. If the TABnOVF0 flag is 1, clear it to 0.  
Because the TABnOVF0 flag is 1, the pulse width can be calculated by  $(10000H + D_{01} - D_{00})$ .
- <5> Read the TABnCCR1 register.  
Read the TABnOVF1 flag. If the TABnOVF1 flag is 1, clear it to 0 (the TABnOVF0 flag is cleared in <4>, and the TABnOVF1 flag remains 1).  
Because the TABnOVF1 flag is 1, the pulse width can be calculated by  $(10000H + D_{11} - D_{10})$  (correct).
- <6> Same as <3>

**Remark**  $n = 0, 1$

**Example when two capture registers are used (without using overflow interrupt)**

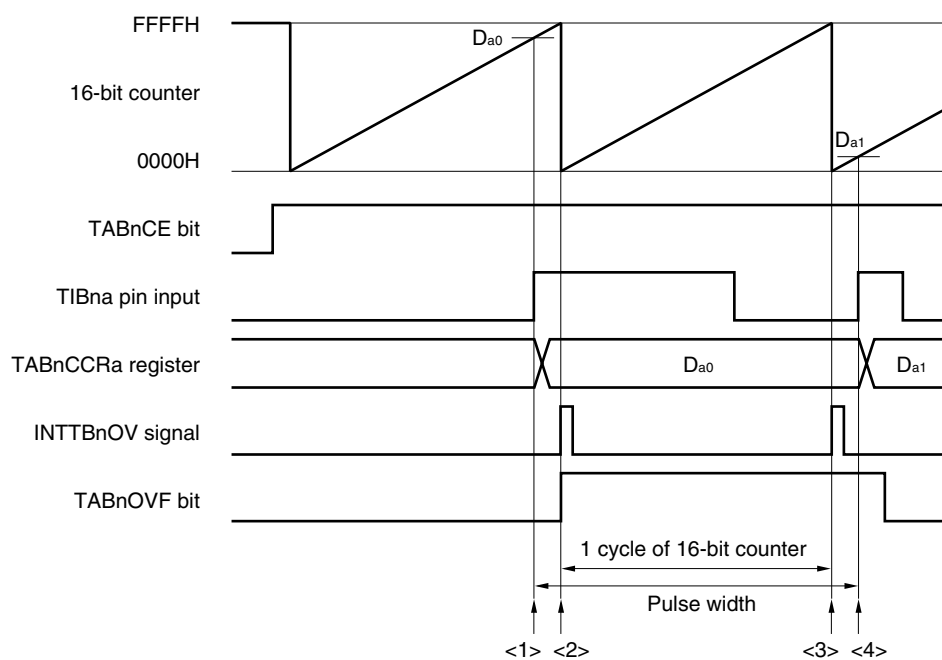
**Note** The TABnOVF0 and TABnOVF1 flags are set on the internal RAM by software.

- <1> Read the TABnCCR0 register (setting of the default value of the TIBn0 pin input).
- <2> Read the TABnCCR1 register (setting of the default value of the TIBn1 pin input).
- <3> An overflow occurs. Nothing is done by software.
- <4> Read the TABnCCR0 register.  
Read the overflow flag. If the overflow flag is 1, set only the TABnOVF1 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 TABnCCR1 register.  
Read the overflow flag. Because the overflow flag is cleared in <4>, 0 is read.  
Read the TABnOVF1 flag. If the TABnOVF1 flag is 1, clear it to 0.  
Because the TABnOVF1 flag is 1, the pulse width can be calculated by  $(10000H + D_{11} - D_{10})$  (correct).
- <6> Same as <3>

**Remark**  $n = 0, 1$

**(d) Processing of overflow if capture trigger interval is long**

If the pulse width is greater than one cycle of the 16-bit counter, care must be exercised because an overflow may occur more than once from the first capture trigger to the next. First, an example of incorrect processing is shown below.

**Example of incorrect processing when capture trigger interval is long**

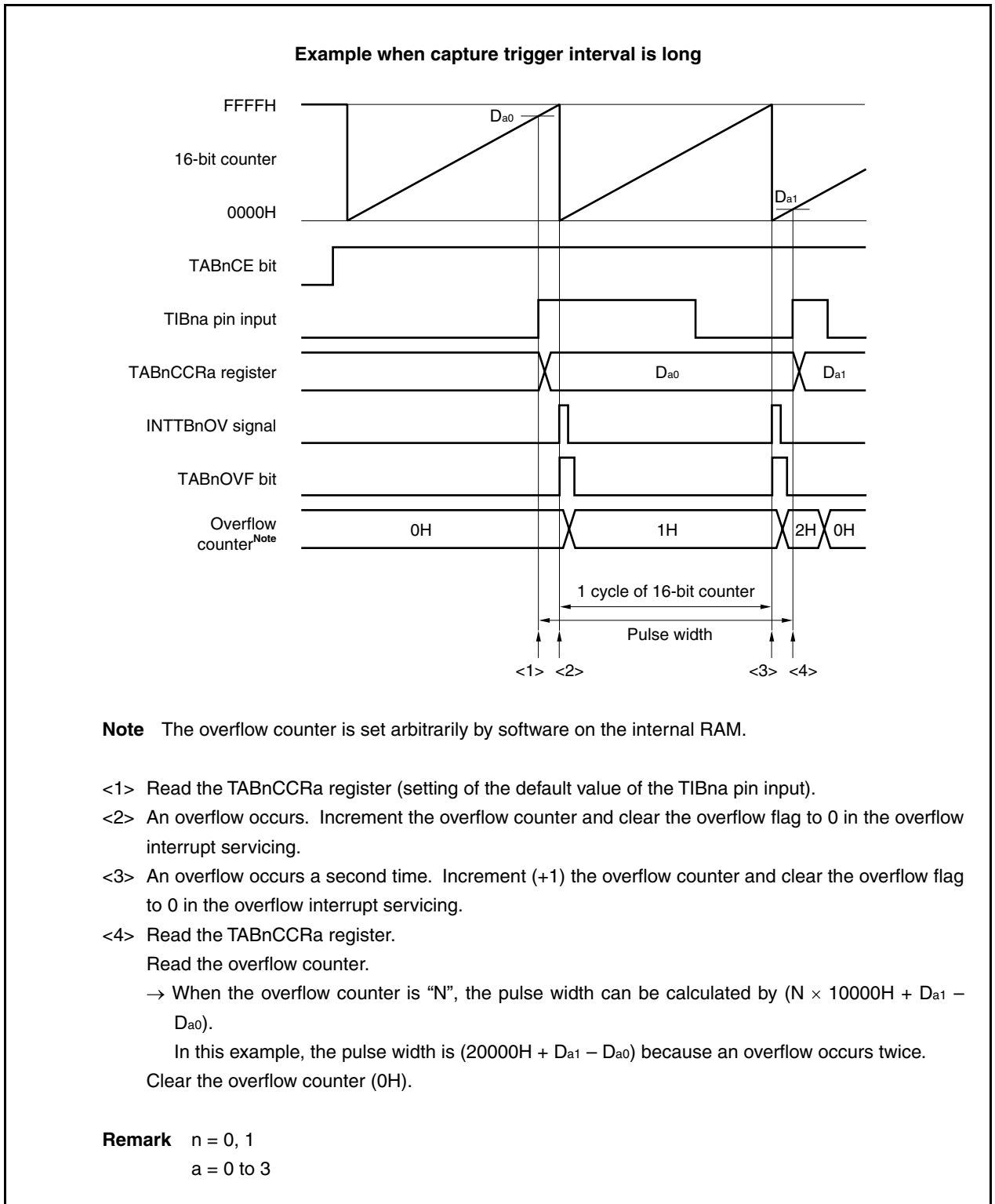
The following problem may occur when a long pulse width in the free-running timer mode.

- <1> Read the TABnCCRa register (setting of the default value of the TIBna pin input).
- <2> An overflow occurs. Nothing is done by software.
- <3> An overflow occurs a second time. Nothing is done by software.
- <4> Read the TABnCCRa 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  $(10000H + D_{a1} - D_{a0})$  (incorrect).  
 Actually, the pulse width must be  $(20000H + D_{a1} - D_{a0})$  because an overflow occurs twice.

**Remark**  $n = 0, 1, a = 0$  to 3

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 TABnOVF bit to 0 with the CLR instruction after reading the TABnOVF bit when it is 1 and by writing 8-bit data (bit 0 is 0) to the TABnOPT0 register after reading the TABnOVF bit when it is 1.

**7.6.7 Pulse width measurement mode (TABnMD2 to TABnMD0 bits = 110)**

In the pulse width measurement mode, 16-bit timer/event counter AB starts counting when the TABnCTL0.TABnCE bit is set to 1. Each time the valid edge input to the TIBna pin has been detected, the count value of the 16-bit counter is stored in the TABnCCRa register, and the 16-bit counter is cleared to 0000H.

The interval of the valid edge can be measured by reading the TABnCCRa register after a capture interrupt request signal (INTTBnCCa) occurs.

As shown in Figure 7-39, select either of the TIBn0 to TIBn3 pins as the capture trigger input pin. Specify “No edge detection” by using the TABnIOC1 register for the unused pins.

**Figure 7-38. Configuration in Pulse Width Measurement Mode**

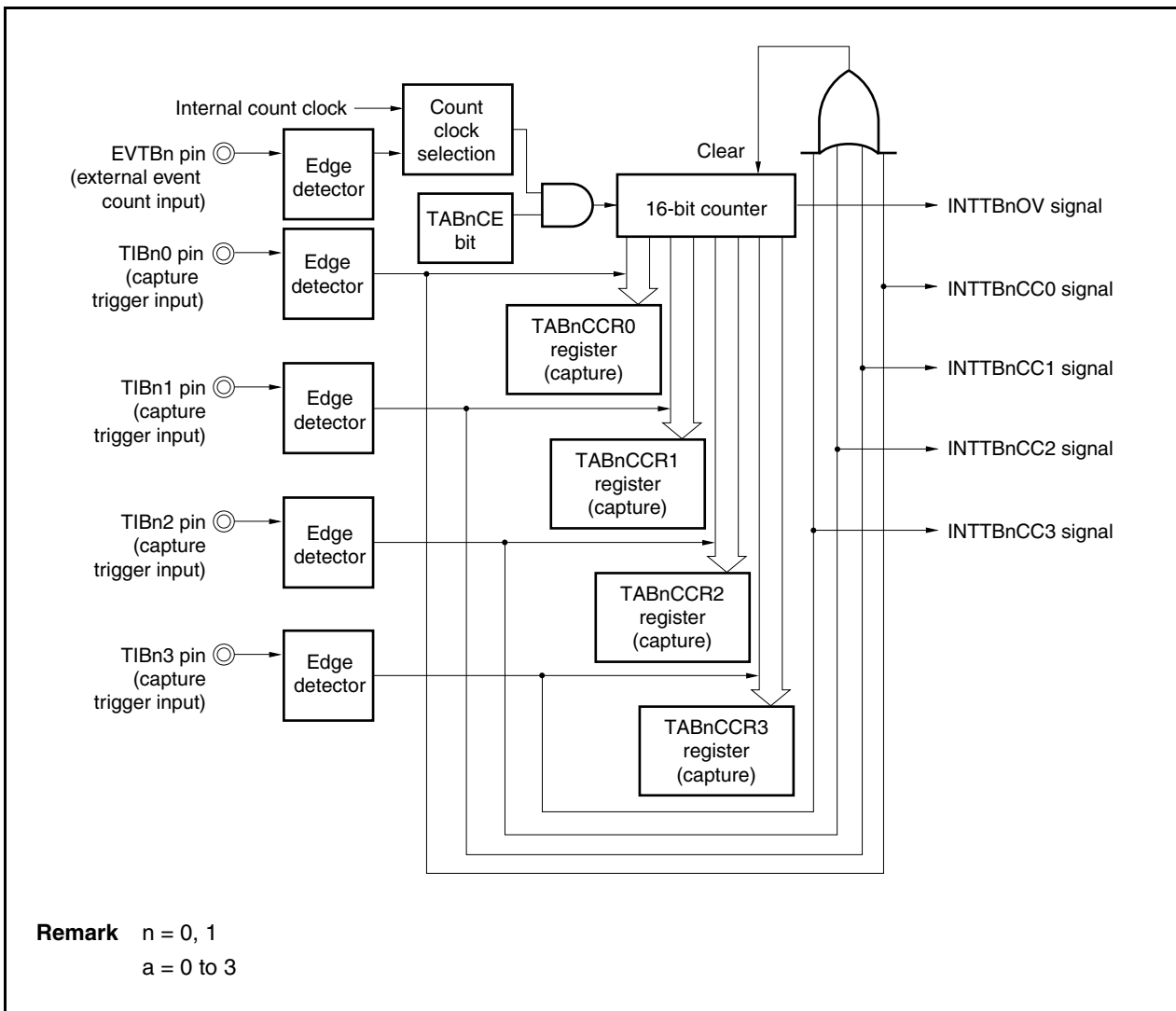
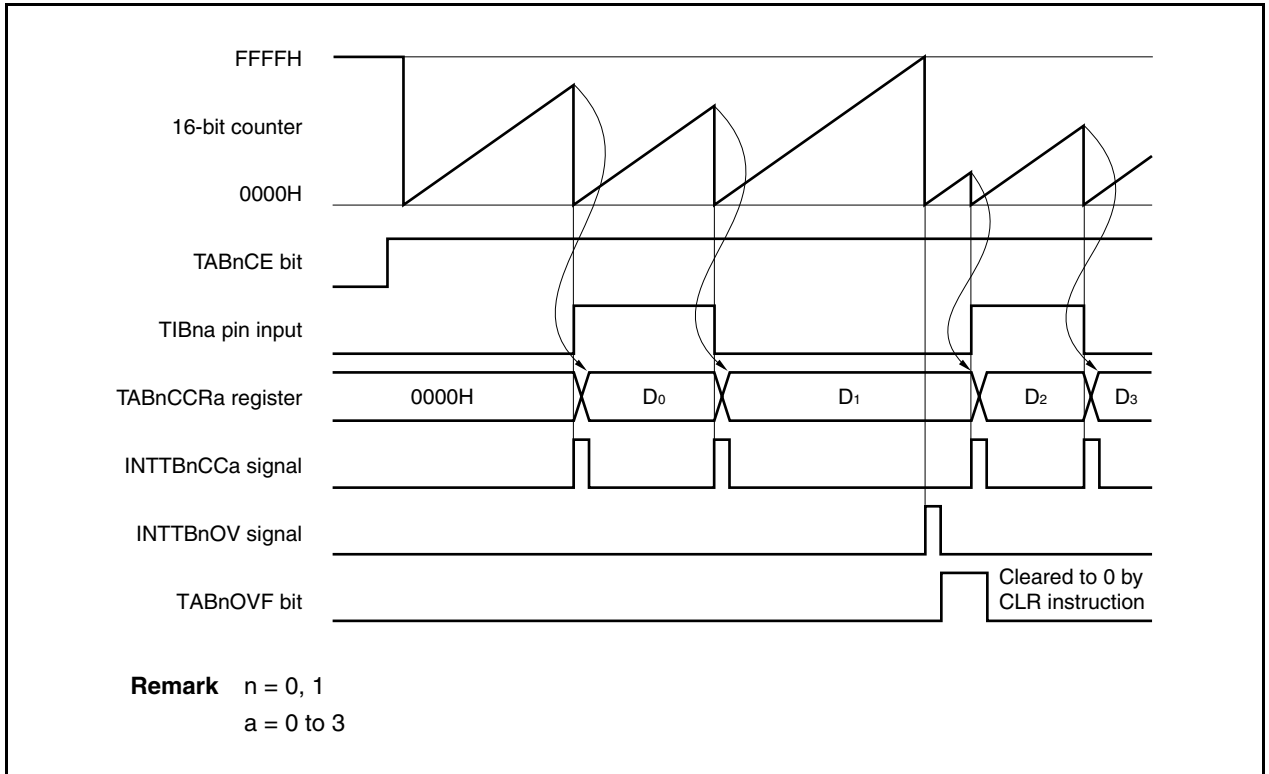


Figure 7-39. Basic Timing in Pulse Width Measurement Mode



When the TABnCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIBna pin is later detected, the count value of the 16-bit counter is stored in the TABnCCRa register, the 16-bit counter is cleared to 0000H, and a capture interrupt request signal (INTTBnCCa) 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 even when the 16-bit counter counted up to FFFFH, an overflow interrupt request signal (INTTBnOV) is generated at the next count clock, and the counter is cleared to 0000H and continues counting. At this time, the overflow flag (TABnOPT0.TABnOVF bit) is also set to 1. Clear the overflow flag to 0 by executing the CLR instruction via software.

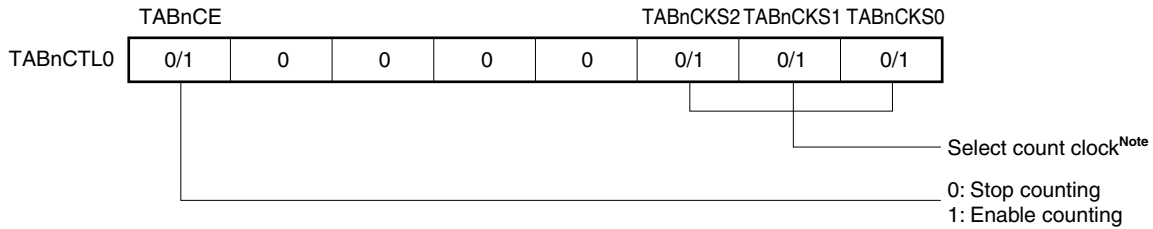
If the overflow flag is set to 1, the pulse width can be calculated as follows.

$$\text{Pulse width} = (10000\text{H} \times \text{TABnOVF bit set (1) count} + \text{Captured value}) \times \text{Count clock cycle}$$

**Remark** n = 0, 1  
a = 0 to 3

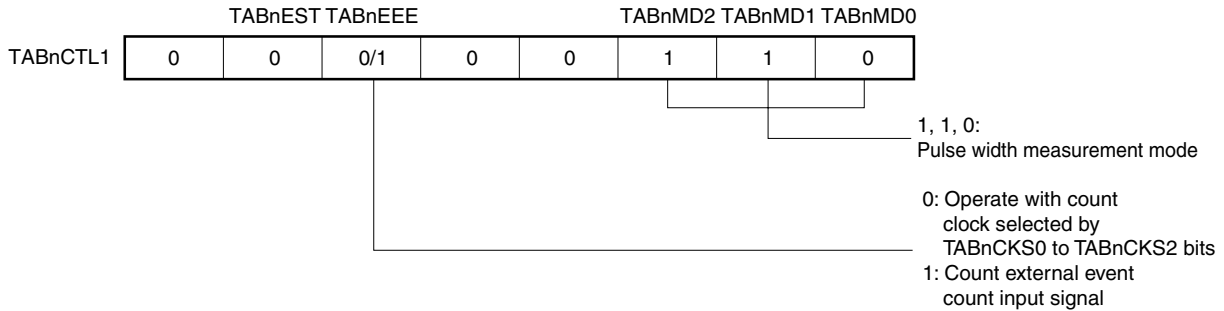
Figure 7-40. Register Setting in Pulse Width Measurement Mode (1/2)

(a) TABn control register 0 (TABnCTL0)

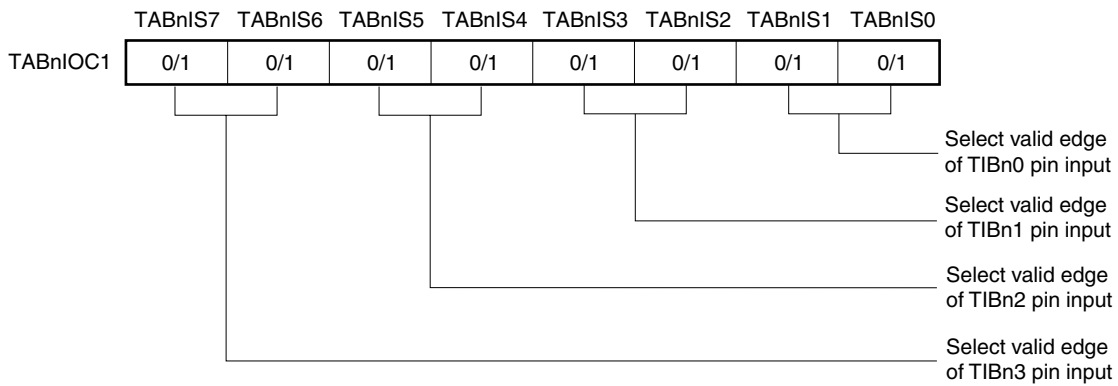


**Note** Setting is invalid when the TABnCTL1.TABnEEE bit = 1.

(b) TABn control register 1 (TABnCTL1)



(c) TABn I/O control register 1 (TABnIOC1)



(d) TABn I/O control register 2 (TABnIOC2)

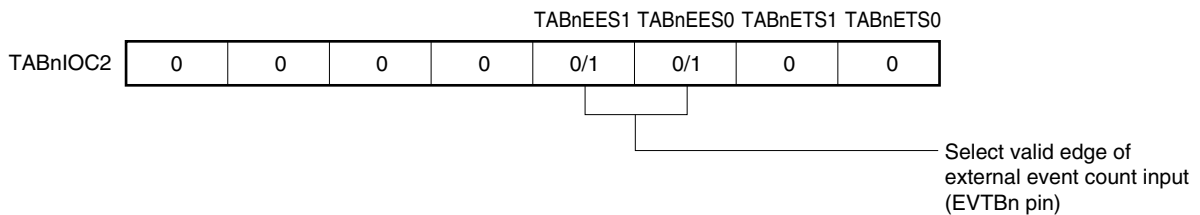
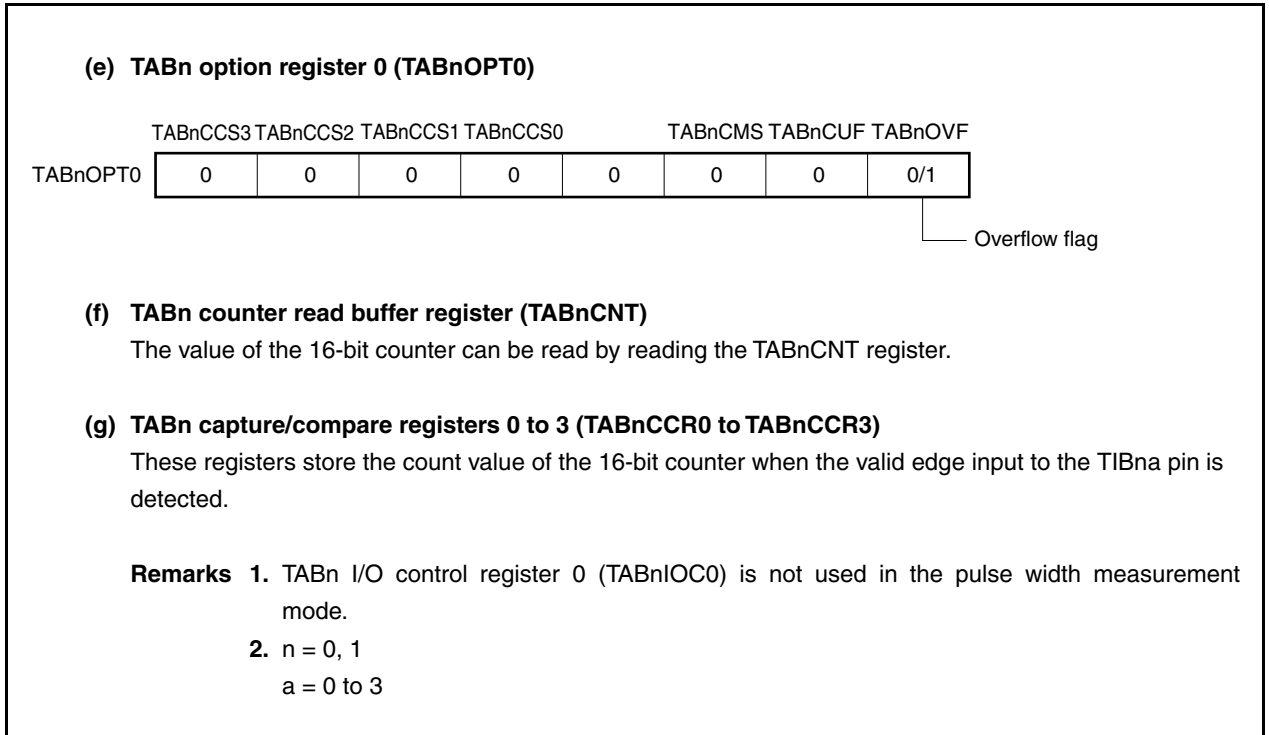


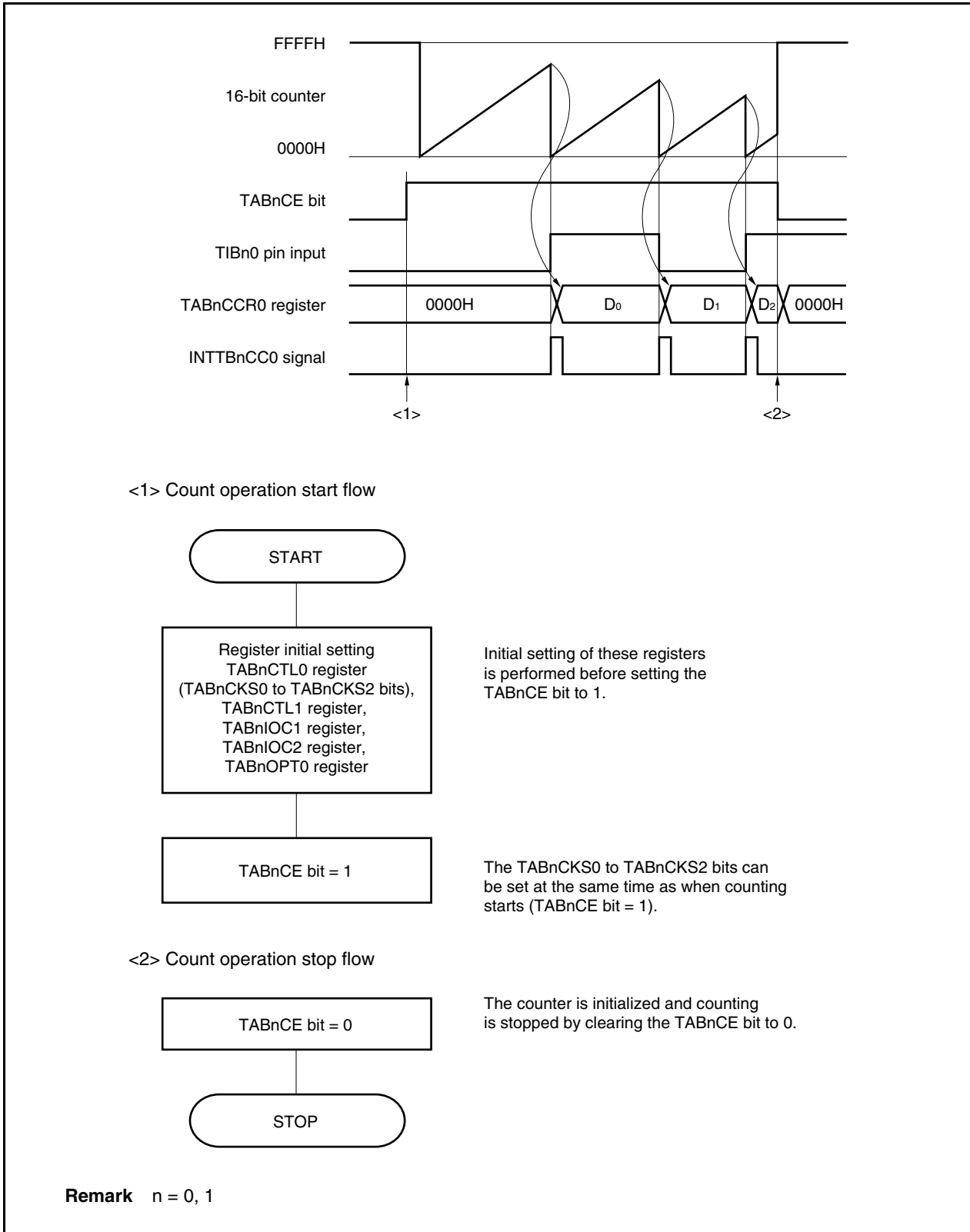


Figure 7-40. Register Setting in Pulse Width Measurement Mode (2/2)



(1) Operation flow in pulse width measurement mode

Figure 7-41. 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 TABnOVF bit to 0 with the CLR instruction after reading the TABnOVF bit when it is 1 and by writing 8-bit data (bit 0 is 0) to the TABnOPT0 register after reading the TABnOVF bit when it is 1.

## CHAPTER 8 16-BIT TIMER/EVENT COUNTER T (TMT)

Timer T (TMT) is a 16-bit timer/event counter.

An encoder count function and other functions are added to the timer AA (TAA). However, TMT does not have a function to operate with an external event count input when it operates in the interval timer mode.

The V850E/IF3 and V850E/IG3 incorporate TMT0 and TMT1.

### 8.1 Overview

The TMTn channels are outlined below (n = 0, 1).

**Table 8-1. TMTn Overview**

| Item   | TMT0                | TMT1   |
|--|---------------------|--------|
| Clock selection                                | 8 ways              | 8 ways |
| Capture trigger input pin                      | <b>Note 1</b>       | 2      |
| External event count input pin                 | <b>Note 2</b>       | 1      |
| External trigger input pin                     | <b>Note 2</b>       | 1      |
| Encoder input pin                              | <b>Note 1</b>       | 2      |
| Encoder clear input pin                        | <b>Note 2</b>       | 1      |
| Timer counter                                  | 1                   | 1      |
| Capture/compare register                       | 2 <sup>Note 3</sup> | 2      |
| Capture/compare match interrupt request signal | 2 <sup>Note 3</sup> | 2      |
| Overflow interrupt request signal              | 1                   | 1      |
| Encoder clear interrupt request signal         | <b>Note 1</b>       | 1      |
| Timer output pin                               | <b>Note 4</b>       | 2      |

**Notes 1.** V850E/IF3: None  
V850E/IG3: 2

**2.** V850E/IF3: None  
V850E/IG3: 1

**3.** In the V850E/IF3, compare function only

## 8.2 Functions

The TMTn functions that can be implemented differ from one channel to another, as shown in the table below (n = 0, 1).

**Table 8-2. TMTn Functions**

| Function                        | TMT0                | TMT1 |
|---------------------------------|---------------------|------|
| Interval timer                  | √                   | √    |
| External event counter          | <b>Note 1</b>       | √    |
| External trigger pulse output   | <b>Note 1</b>       | √    |
| One-shot pulse output           | <b>Note 1</b>       | √    |
| PWM output                      | <b>Note 1</b>       | √    |
| Free-running timer              | √ <sup>Note 2</sup> | √    |
| Pulse width measurement         | ×                   | √    |
| Triangular-wave PWM output mode | <b>Note 1</b>       | √    |
| Encoder count function          | <b>Note 1</b>       | √    |

**Notes 1.** V850E/IF3: ×

V850E/IG3: √

**2.** In the V850E/IF3, compare function only

### 8.3 Configuration

TMTn includes the following hardware (n = 0, 1).

**Table 8-3. Configuration of TMTn**

| Item              | Configuration  |
|-------------------|--|
| Timer register    | 16-bit counter × 1   |
| Registers         | TMTn capture/compare registers 0, 1 (TTnCCR0, TTnCCR1)<br>TMTn counter read buffer register (TTnCNT)<br>TMTm counter write register (TTmTCW)<br>CCR0 and CCR1 buffer registers   |
| Timer input       | 2 in total (TIT00 <sup>Note 1</sup> , TIT01 <sup>Note 1</sup> , TIT10, TIT11, EVTT0 <sup>Note 1</sup> , EVTT1, TENC00 <sup>Note 1</sup> , TENC01 <sup>Note 1</sup> , TENC10, TENC11, TECR0 <sup>Note 1</sup> , TECR1 pins) <sup>Note 2</sup>                         |
| Timer output      | 4 in total (TOT00 <sup>Note 1</sup> , TOT01 <sup>Note 1</sup> , TOT10, TOT11 pins) <sup>Note 2</sup>   |
| Control registers | TMTn control registers 0, 1 (TTnCTL0, TTnCTL1)<br>TMTm control register 2 (TTmCTL2)<br>TMTm I/O control registers 0 to 3 (TTmIOC0 to TTmIOC3)<br>TMTn option register 0 (TTnOPT0)<br>TMTm option register 1 (TTmOPT1)<br>TMTm capture input select register (TTISLm) |

**Notes 1.** V850E/IG3 only

- TIT00/TECR0 and TIT10/TECR1 pins function alternately as capture trigger input pins (TIT00, TIT10), encoder clear input pins (TECR0, TECR1), and timer output pins (TOT00, TOT10). TENC00/EVTT0 and TENC10/EVTT1 pins function alternately as encoder input pins (TENC00, TENC10), external event count input pins (EVTT0, EVTT1), and external trigger input pins (EVTT0, EVTT1). TIT01/TENC01 and TIT11/TENC11 pins function alternately as capture trigger input pins (TIT01, TIT11), encoder input pins (TENC01, TENC11), and timer output pins (TOT01, TOT11).

**Remark** V850E/IF3: n = 0, 1, m = 1  
V850E/IG3: n = 0, 1, m = 0, 1

Figure 8-1. Block Diagram of TMT0 for V850E/IF3

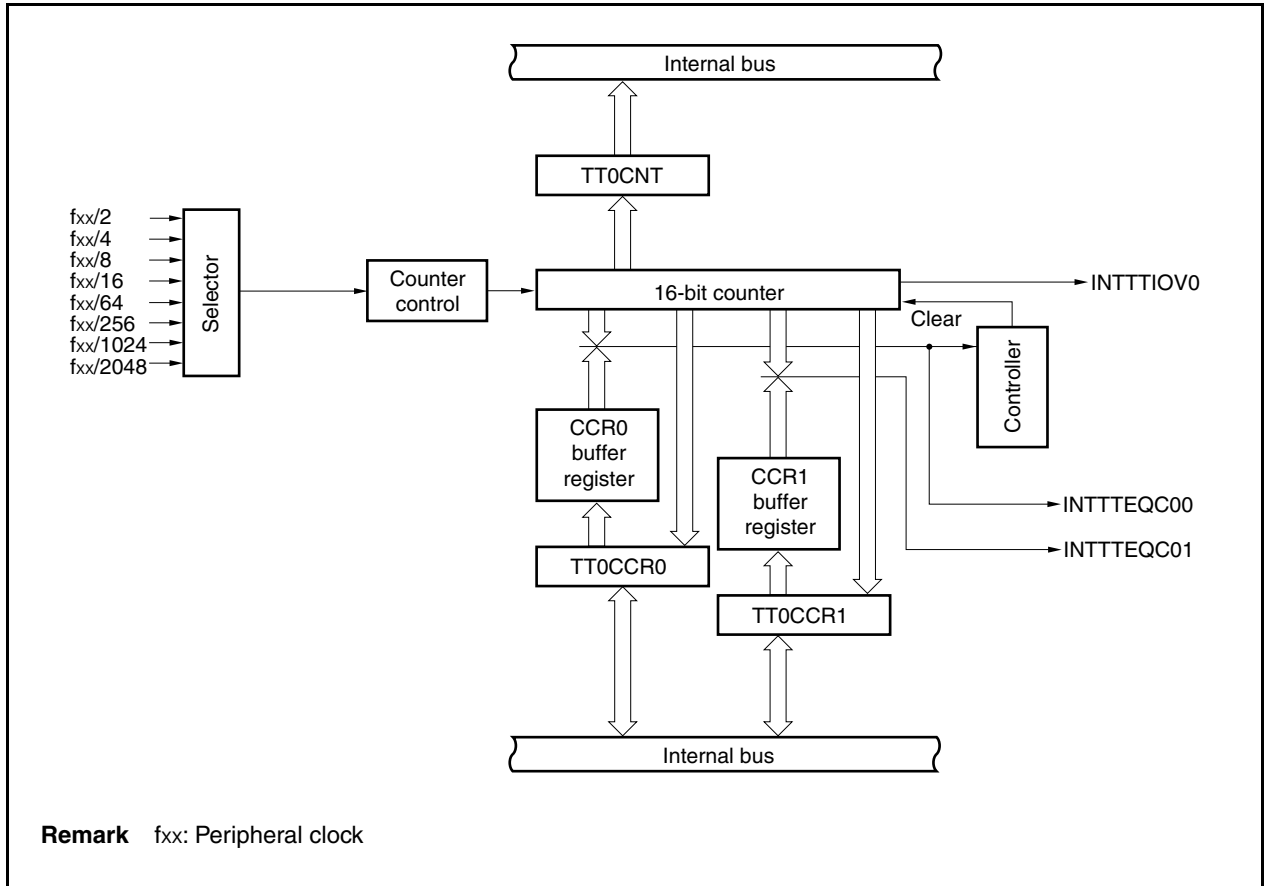
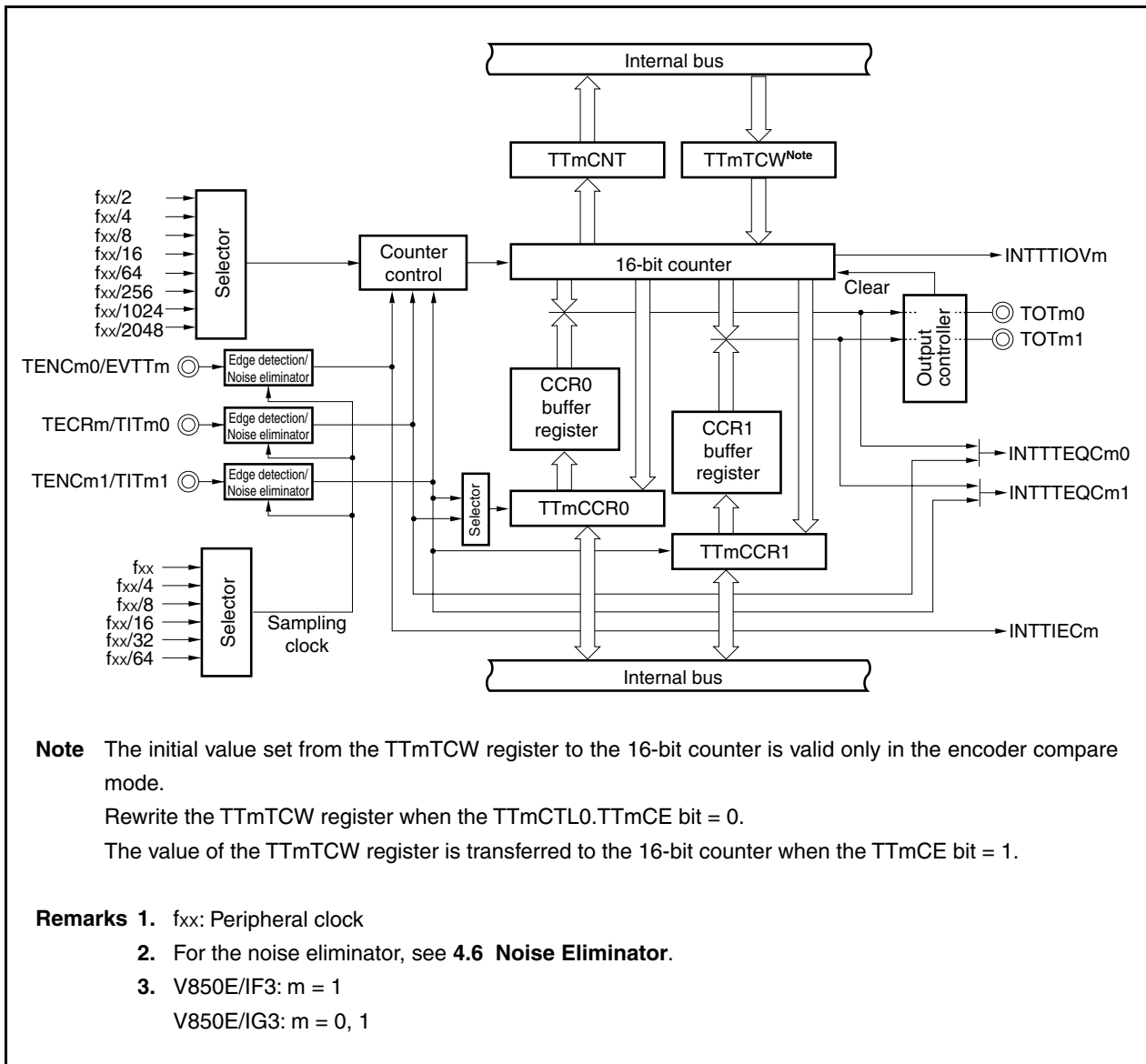


Figure 8-2. Block Diagram of TMT1 for V850E/IF3 and TMT0 and TMT1 for V850E/IG3





**(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 TTnCNT register.

When the TTnCTL0.TTnCE bit = 0, the value of the 16-bit counter is FFFFH. If the TTnCNT register is read at this time, 0000H is read.

Reset sets the TTnCE bit to 0.

**(2) CCR0 buffer register**

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

When the TTnCCR0 register is used as a compare register, the value written to the TTnCCR0 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 (INTTTEQCn0) is generated.

The CCR0 buffer register cannot be read or written directly.

The CCR0 buffer register is set to 0000H after reset, and the TTnCCR0 register is set 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 TTnCCR1 register is used as a compare register, the value written to the TTnCCR1 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 (INTTTEQCn1) is generated.

The CCR1 buffer register cannot be read or written directly.

The CCR1 buffer register is set to 0000H after reset, and the TTnCCR1 register is set to 0000H.

**(4) Edge detector**

This circuit detects the valid edges input to the TIT00 (V850E/IG3 only), TIT01 (V850E/IG3 only), TIT10, TIT11, EVTT0 (V850E/IG3 only), EVTT1, TENC00 (V850E/IG3 only), TENC01 (V850E/IG3 only), TENC10, TENC11, TECR0 (V850E/IG3 only), and TECR1 pins. No edge, rising edge, falling edge, or both the rising and falling edges can be selected as the valid edge by using the TTmIOC1, TTmIOC2, and TTmIOC3 registers.

**(5) Output controller**

This circuit controls the output of the TOT00 (V850E/IG3 only), TOT01 (V850E/IG3 only), TOT10, and TOT11 pins. The output controller is controlled by the TTmIOC0 registers.

**(6) Selector**

This selector selects the count clock for the 16-bit counter. Eight types of internal clocks or an external event can be selected as the count clock.

**(7) Counter control**

The count operation is controlled by the timer mode selected by the TTnCTL1 register.

8.4 Registers

(1) TMTn control register 0 (TTnCTL0)

The TTnCTL0 register is an 8-bit register that controls the operation of TMTn.

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 TTnCTL0 register by software.

After reset: 00H    R/W    Address: TT0CTL0 FFFFFFF580H, TT1CTL0 FFFFFFF5C0H

|                       |       |   |   |   |   |         |         |         |
|-----------------------|-------|---|---|---|---|---------|---------|---------|
|                       | <7>   | 6 | 5 | 4 | 3 | 2       | 1       | 0       |
| TTnCTL0<br>(n = 0, 1) | TTnCE | 0 | 0 | 0 | 0 | TTnCKS2 | TTnCKS1 | TTnCKS0 |

|       |  |
|-------|--|
| TTnCE | TMTn operation control   |
| 0     | TMTn operation disabled (TMTn reset asynchronously <sup>Note</sup> ) |
| 1     | TMTn operation enabled. TMTn operation start                         |

|         |         |         |                                |
|---------|---------|---------|--------------------------------|
| TTnCKS2 | TTnCKS1 | TTnCKS0 | Internal count clock selection |
| 0       | 0       | 0       | f <sub>xx</sub> /2             |
| 0       | 0       | 1       | f <sub>xx</sub> /4             |
| 0       | 1       | 0       | f <sub>xx</sub> /8             |
| 0       | 1       | 1       | f <sub>xx</sub> /16            |
| 1       | 0       | 0       | f <sub>xx</sub> /64            |
| 1       | 0       | 1       | f <sub>xx</sub> /256           |
| 1       | 1       | 0       | f <sub>xx</sub> /1024          |
| 1       | 1       | 1       | f <sub>xx</sub> /2048          |

**Note** The TTnOPT0.TTnOVF bit and the 16-bit counter are reset simultaneously. Moreover, timer outputs (TOT00 (V850E/IG3 only), TOT01 (V850E/IG3 only), TOT10, and TOT11 pins) are reset to the TTmIOC0 register set status at the same time as the 16-bit counter is reset (V850E/IF3: m = 1, V850E/IG3: m = 0, 1).

- Cautions**
1. Set the TTnCKS2 to TTnCKS0 bits when the TTnCE bit = 0.  
When the value of the TTnCE bit is changed from 0 to 1, the TTnCKS2 to TTnCKS0 bits can be set simultaneously.
  2. Be sure to set bits 3 to 6 to “0”.

**Remark** f<sub>xx</sub>: Peripheral clock

**(2) TMTn control register 1 (TTnCTL1)**

The TTnCTL1 register is an 8-bit register that controls the TMTn operation.

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: TT0CTL1 FFFFF581H, TT1CTL1 FFFFF5C1H

|         |   |                        |                        |   |        |        |        |        |
|---------|---|------------------------|------------------------|---|--------|--------|--------|--------|
|         | 7 | 6                      | 5                      | 4 | 3      | 2      | 1      | 0      |
| TTnCTL1 | 0 | TTmEST <sup>Note</sup> | TTmEEE <sup>Note</sup> | 0 | TTnMD3 | TTnMD2 | TTnMD1 | TTnMD0 |

V850E/IF3  
n = 0, 1  
m = 1

V850E/IG3  
n = 0, 1  
m = 0, 1

|                        |  |   |
|------------------------|--|---|
| TTmEST <sup>Note</sup> | 0  | - |
| 1                      | Generate a valid signal for external trigger input. <ul style="list-style-type: none"> <li>• In one-shot pulse output mode: A one-shot pulse is output with writing 1 to the TTmEST bit as the trigger.</li> <li>• In external trigger pulse output mode: A PWM waveform is output with writing 1 to the TTmEST bit as the trigger.</li> </ul> |   |

The read value of the TTmEST bit is always 0.

|                        |   |  |
|------------------------|---|--|
| TTmEEE <sup>Note</sup> | Count clock selection   |  |
| 0                      | Disable operation with external event count input (EVTTm pin). (Perform counting with the count clock selected by the TTmCTL0.TTmCKS0 to TTmCTL0.TTmCKS2 bits.) |  |
| 1                      | Enable operation with external event count input (EVTTm pin). (Perform counting at the valid edge of the external event count input signal (EVTTm pin).)        |  |

The TTmEEE bit selects whether counting is performed with the internal count clock or the valid edge of the external event count input.

|                  |        |        |        |                                    |
|------------------|--------|--------|--------|------------------------------------|
| TTnMD3           | TTnMD2 | TTnMD1 | TTnMD0 | Timer mode selection               |
| 0                | 0      | 0      | 0      | Interval timer mode                |
| 0                | 0      | 0      | 1      | External event count mode          |
| 0                | 0      | 1      | 0      | External trigger pulse output mode |
| 0                | 0      | 1      | 1      | One-shot pulse output mode         |
| 0                | 1      | 0      | 0      | PWM output mode                    |
| 0                | 1      | 0      | 1      | Free-running timer mode            |
| 0                | 1      | 1      | 0      | Pulse width measurement mode       |
| 0                | 1      | 1      | 1      | Triangular-wave PWM output mode    |
| 1                | 0      | 0      | 0      | Encoder compare mode               |
| Other than above |        |        |        | Setting prohibited                 |

**Note** In the V850E/IF3, this bit can be set only in TMT1. Be sure to set bits 5 and 6 of TMT0 to "0".

- Cautions**
1. The TTmEST 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. The TTmEEE bit is valid only in the interval timer mode, external trigger pulse output mode, one-shot pulse output mode, PWM output mode, free-running timer mode, pulse width measurement mode, or triangular-wave PWM output mode. In any other mode, writing 1 to this bit is ignored.
  3. External event count input (EVTTm) or encoder inputs (TENCm0, TENCm1) is selected in the external event count mode or encoder compare mode regardless of the value of the TTmEEE bit.
  4. Set the TTmEEE and TTnMD3 to TTnMD0 bits when the TTnCTL0.TTnCE bit = 0. (The same value can be written when the TTnCE bit = 1.) The operation is not guaranteed when rewriting is performed with the TTnCE bit = 1. If rewriting was mistakenly performed, clear the TTnCE bit to 0 and then set the bits again.
  5. Be sure to set bits 4 and 7 to "0".

**(3) TMTm control register 2 (TTmCTL2)**

The TTmCTL2 register is an 8-bit register that controls the encoder count function operation.

The TTmCTL2 register is valid only in the encoder compare mode.

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

Reset sets this register to 00H.

**Caution** For details of each bit of the TTmCTL2 register, see 8.6.9 (5) Controlling bits of TTmCTL2 register.

(1/2)

|                            |         |   |  |        |         |         |         |         |   |   |
|----------------------------|---------|---|--|--------|---------|---------|---------|---------|---|---|
| After reset: 00H           |         | R/W   | Address: TT0CTL2 FFFFF582H <sup>Note</sup> , TT1CTL2 FFFFF5C2H |        |         |         |         |         |   |   |
|                            |         |   | 7  | 6      | 5       | 4       | 3       | 2       | 1 | 0 |
| TTmCTL2                    | TTmECC  | 0   | 0  | TTmLDE | TTmECM1 | TTmECM0 | TTmUDS1 | TTmUDS0 |   |   |
| [ V850E/IF3<br>m = 1 ]     | TTmECC  | Encoder counter control   |  |        |         |         |         |         |   |   |
| [ V850E/IG3<br>m = 0, 1 ]  | 0       | Normal operation  |  |        |         |         |         |         |   |   |
|                            | 1       | Holds count value of 16-bit counter when TTmCTL0.TTmCE bit = 0.                                 |  |        |         |         |         |         |   |   |
|                            | TTmLDE  | Transfer setting to 16-bit counter  |  |        |         |         |         |         |   |   |
|                            | 0       | Disables transfer of set value of TTmCCR0 to 16-bit counter in case of underflow.               |  |        |         |         |         |         |   |   |
|                            | 1       | Enables transfer of set value of TTmCCR0 to 16-bit counter in case of underflow.                |  |        |         |         |         |         |   |   |
|                            | TTmECM1 | Control of encoder clear operation 1  |  |        |         |         |         |         |   |   |
|                            | 0       | The 16-bit counter is not cleared to 0000H when its count value matches value of CCR1 register. |  |        |         |         |         |         |   |   |
|                            | 1       | The 16-bit counter is cleared to 0000H when its count value matches value of CCR1 register.     |  |        |         |         |         |         |   |   |
|                            | TTmECM0 | Control of encoder clear operation 0  |  |        |         |         |         |         |   |   |
|                            | 0       | The 16-bit counter is not cleared to 0000H when its count value matches value of CCR0 register. |  |        |         |         |         |         |   |   |
|                            | 1       | The 16-bit counter is cleared to 0000H when its count value matches value of CCR0 register.     |  |        |         |         |         |         |   |   |
| <b>Note</b> V850E/IG3 only |         |   |  |        |         |         |         |         |   |   |

| TTmUDS1 | TTmUDS0 | Up/down count selection   |
|---------|---------|---|
| 0       | 0       | When valid edge of TENCm0 input is detected<br>Counts down when TENCm1 = high level.<br>Counts up when TENCm1 = low level.  |
| 0       | 1       | Counts up when valid edge of TENCm0 input is detected.<br>Counts down when valid edge of TENCm1 input is detected.  |
| 1       | 0       | Counts down when rising edge of TENCm0 input is detected.<br>Counts up when falling edge of TENCm0 input is detected.<br>However, count operation is performed only when<br>TENCm1 = low level. |
| 1       | 1       | Both rising and falling edges of TENCm0 and TENCm1 are<br>detected. Count operation is automatically identified by<br>combination of edge detection and level detection.                        |

**Cautions 1.** The TTmECC bit is valid only in the encoder compare mode. In any other mode, writing “1” to this bit is ignored.

If the TTmCTL0.TTmCE bit is cleared to 0 while the TTmECC bit = 1, the values of the timer/counter and capture registers (TTmCCR0 and TTmCCR1), and the TTmOPT1, TTmEUF, TTmEOF, and TTmESF flags are retained.

If the TTmCE bit is set from 0 to 1 when the TTmECC bit = 1, the value of the TTmTCW register is not transferred to the 16-bit counter.

2. The TTmLDE bit is valid only when the TTmECM1 and TTmECM0 bits = 00, 01. Writing “1” to this bit is ignored when the TTmECM1 and TTmECM0 bits = 10, 11.
3. The edge detection of the TENCm0 and TENCm1 inputs specified by the TTmIOC3.TTmEIS1 and TTmIOC3.TTmEIS0 bits is invalid and fixed to both the rising and falling edges when the TTmUDS1 and TTmUDS0 bits = 10, 11.
4. Set the TTmLDE, TTmECM1, TTmECM0, TTmUDS1, and TTmUDS0 bits when the TTmCTL0.TTmCE bit = 0 (the same value can be written to these bits when the TTmCE bit = 1). If the value of these bits is changed when the TTmCE bit = 1, the operation cannot be guaranteed. If it is changed by mistake, clear the TTmCE bit and then set the correct value.
5. Be sure to set bits 5 and 6 to “0”.

**(4) TMTm I/O control register 0 (TTmIOC0)**

The TTmIOC0 register is an 8-bit register that controls the timer output (TOTm0, TOTm1 pins).

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

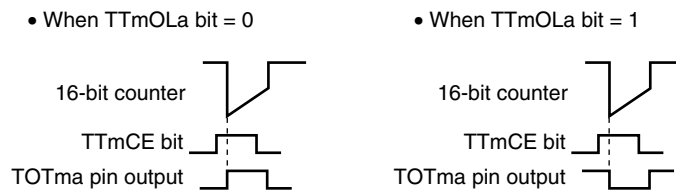
Reset sets this register to 00H.

After reset: 00H R/W Address: TT0IOC0 FFFFF583H<sup>Note 1</sup>, TT1IOC0 FFFFF5C3H

|                                 |        |  |  |   |        |        |        |        |
|---------------------------------|--------|--|--|---|--------|--------|--------|--------|
|                                 | 7      | 6  | 5  | 4 | 3      | <2>    | 1      | <0>    |
| TTmIOC0<br>(V850E/IF3<br>m = 1) | 0      | 0  | 0  | 0 | TTmOL1 | TTmOE1 | TTmOL0 | TTmOE0 |
|                                 | TTmOL1 |  | TOTm1 pin output level setting <sup>Note 2</sup> |   |        |        |        |        |
| (V850E/IG3<br>m = 0, 1)         | 0      | TOTm1 pin starts output at high level.   |  |   |        |        |        |        |
|                                 | 1      | TOTm1 pin starts output at low level.  |  |   |        |        |        |        |
|                                 | TTmOE1 |  | TOTm1 pin output setting                         |   |        |        |        |        |
|                                 | 0      | Timer output prohibited<br>• Low level is output from the TOTm1 pin when the TTmOL1 bit = 0.<br>• High level is output from the TOTm1 pin when the TTmOL1 bit = 1. |  |   |        |        |        |        |
|                                 | 1      | Timer output enabled (A pulse is output from the TOTm1 pin.)   |  |   |        |        |        |        |
|                                 | TTmOL0 |  | TOTm0 pin output level setting <sup>Note 2</sup> |   |        |        |        |        |
|                                 | 0      | TOTm0 pin starts output at high level.   |  |   |        |        |        |        |
|                                 | 1      | TOTm0 pin starts output at low level.  |  |   |        |        |        |        |
|                                 | TTmOE0 |  | TOTm0 pin output setting                         |   |        |        |        |        |
|                                 | 0      | Timer output prohibited<br>• Low level is output from the TOTm0 pin when the TTmOL0 bit = 0.<br>• High level is output from the TOTm0 pin when the TTmOL0 bit = 1. |  |   |        |        |        |        |
|                                 | 1      | Timer output enabled (A pulse is output from the TOTm0 pin.)   |  |   |        |        |        |        |

**Notes 1.** V850E/IG3 only

- 2.** The output level of the timer output pins (TOTm0, TOTm1) in modes other than the triangular-wave PWM output mode, which is specified by the TTmOLa bit, is as follows (a = 0, 1).



For the output level in the triangular-wave PWM output mode, see **Figure 8-44 Basic Timing in Triangular-Wave PWM Output Mode.**

- Cautions 1.** If the setting of the TTmIOC0 register is changed when TOTm0 and TOTm1 outputs are set for the port mode, the output of the pins change. Set the port in the input mode and make the port go into a high-impedance state, noting changes in the pin status.
- 2.** Rewrite the TTmOL1, TTmOE1, TTmOL0, and TTmOE0 bits when the TTmCTL0.TTmCE bit = 0. (The same value can be written when the TTmCE bit = 1.) If rewriting was mistakenly performed, clear the TTmCE bit to 0 and then set the bits again.
- 3.** Even if the TTmOL0 or TTmOL1 bit is manipulated when the TTmCE, TTmOE0, and TTmOE1 bits are 0, the output level of the TOTm0 and TOTm1 pins changes.



**(5) TMTm I/O control register 1 (TTmIOC1)**

The TTmIOC1 register is an 8-bit register that controls the valid edge for the capture trigger input signals (TITm0, TITm1 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: TT0IOC1 FFFFF584H <sup>Note</sup> , TT1IOC1 FFFFF5C4H |   |   |   |        |        |        |        |
|                  |  |     | 7  | 6 | 5 | 4 | 3      | 2      | 1      | 0      |
| TTmIOC1          |  |     | 0  | 0 | 0 | 0 | TTmIS3 | TTmIS2 | TTmIS1 | TTmIS0 |

|                           |        |        |   |
|---------------------------|--------|--------|---|
| [ V850E/IF3<br>m = 1 ]    | TTmIS3 | TTmIS2 | Capture trigger input signal (TITm1 pin) valid edge setting |
|                           | 0      | 0      | No edge detection (capture operation invalid)               |
| [ V850E/IG3<br>m = 0, 1 ] | 0      | 1      | Detection of rising edge                                    |
|                           | 1      | 0      | Detection of falling edge                                   |
|                           | 1      | 1      | Detection of both edges                                     |

|        |        |   |
|--------|--------|---|
| TTmIS1 | TTmIS0 | Capture trigger input signal (TITm0 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                                     |

**Note** V850E/IG3 only

**Cautions**

1. Rewrite the TTmIS3 to TTmIS0 bits when the TTmCTL0.TTmCE bit = 0.  
(The same value can be written when the TTmCE bit = 1.) If rewriting was mistakenly performed, clear the TTmCE bit to 0 and then set the bits again.
2. The TTmIS3 and TTmIS2 bits are valid only in the free-running timer mode (only when the TTmOPT0.TTmCCS1 bit = 1) and the pulse width measurement mode. In all other modes, a capture operation is not possible.  
The TTmIS1 and TTmIS0 bits are valid only in the free-running timer mode (only when the TTmOPT0.TTmCCS0 bit = 1) and the pulse width measurement mode. In all other modes, a capture operation is not possible.

**(6) TMTm I/O control register 2 (TTmIOC2)**

The TTmIOC2 register is an 8-bit register that controls the valid edge for the external event count input signal (EVTTm pin) and external trigger input signal (EVTTm 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: TT0IOC2 FFFFF585H<sup>Note</sup>, TT1IOC2 FFFFF5C5H

|         |   |   |   |   |         |         |         |         |
|---------|---|---|---|---|---------|---------|---------|---------|
|         | 7 | 6 | 5 | 4 | 3       | 2       | 1       | 0       |
| TTmIOC2 | 0 | 0 | 0 | 0 | TTmEES1 | TTmEES0 | TTmETS1 | TTmETS0 |

[ V850E/IF3  
m = 1 ]

|         |         |  |
|---------|---------|--|
| TTmEES1 | TTmEES0 | External event count input signal (EVTTm 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  |

[ V850E/IG3  
m = 0, 1 ]

|         |         |  |
|---------|---------|--|
| TTmETS1 | TTmETS0 | External trigger input signal (EVTTm 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                                      |

**Note** V850E/IG3 only

**Cautions**

1. Rewrite the TTmEES1, TTmEES0, TTmETS1, and TTmETS0 bits when the TTmCTL0.TTmCE bit = 0. (The same value can be written when the TTmCE bit = 1.) If rewriting was mistakenly performed, clear the TTmCE bit to 0 and then set the bits again.
2. The TTmEES1 and TTmEES0 bits are valid only when the TTmCTL1.TTmEEE bit = 1 or when the external event count mode (the TTmCTL1.TTmMD3 to TTmCTL1.TTmMD0 bits = 0001) has been set.
3. The TTmETS1 and TTmETS0 bits are valid only in the external trigger pulse mode or one-shot pulse output mode.

**(7) TMTm I/O control register 3 (TTmIOC3)**

The TTmIOC3 register is an 8-bit register that controls the encoder clear function operation.

The TTmIOC3 register is valid only in the encoder compare mode.

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: TT0IOC3 FFFFF586H <sup>Note</sup> , TT1IOC3 FFFFF5C6H |   |  |        |        |         |         |         |         |
|   | 7   | 6  | 5      | 4      | 3       | 2       | 1       | 0       |
| TTmIOC3   | TTmSCE  | TTmZCL   | TTmBCL | TTmACL | TTmECS1 | TTmECS0 | TTmEIS1 | TTmEIS0 |
| [ V850E/IF3<br>m = 1 ]  |   |  |        |        |         |         |         |         |
| [ V850E/IG3<br>m = 0, 1 ]   | TTmSCE  | Encoder clear selection  |        |        |         |         |         |         |
|   | 0   | Clears 16-bit counter on detection of edge of encoder clear signal (TECRm pin).                    |        |        |         |         |         |         |
|   | 1   | Clears 16-bit counter on detection of clear level condition of the TENCm0, TENCm1, and TECRm pins. |        |        |         |         |         |         |
|   | <ul style="list-style-type: none"> <li>• Clears 16-bit counter to 0000H when valid edge of TECRm pin specified by the TTmECS1 and TTmECS0 bits is detected when the TTmSCE bit = 0.</li> <li>• Clears 16-bit counter to 0000H when clear level conditions of the TTmZCL, TTmBCL, and TTmACL bits match input levels of the TECRm, TENCm1, and TENCm0 pins when TTmSCE bit = 1.</li> <li>• Setting of the TTmZCL, TTmBCL, and TTmACL bits is valid and that of the TTmECS1 and TTmECS0 bits is invalid when the TTmSCE bit = 1. Encoder clear interrupt request signal (INTTIECm) is not generated.</li> <li>• Setting of the TTmZCL, TTmBCL, and TTmACL bits is invalid and setting of the TTmECS1 and TTmECS0 bits is valid when the TTmSCE bit = 0. The INTTIECm signal is generated when valid edge specified by the TTmECS1 and TTmECS0 bits is detected.</li> <li>• Be sure to set the TTmCTL2.TTmUDS1 and TTmCTL2.TTmUDS0 bits to 10 or 11 when the TTmSCE bit = 1. Operation is not guaranteed if the TTmUDS1 and TTmUDS0 bits = 00 or 01 and the TTmSCE bit = 1.</li> </ul> |  |        |        |         |         |         |         |
|   | TTmZCL  | Clear level selection of encoder clear signal (TECRm pin)  |        |        |         |         |         |         |
|   | 0   | Clears low level of the TECRm pin.   |        |        |         |         |         |         |
|   | 1   | Clears high level of the TECRm pin.  |        |        |         |         |         |         |
|   | Setting of the TTmZCL bit is valid only when the TTmSCE bit = 1.  |  |        |        |         |         |         |         |
|   | TTmBCL  | Clear level selection of encoder input signal (TENCm1 pin)   |        |        |         |         |         |         |
|   | 0   | Clears low level of the TENCm1 pin.  |        |        |         |         |         |         |
|   | 1   | Clears high level of the TENCm1 pin.   |        |        |         |         |         |         |
|   | Setting of the TTmBCL bit is valid only when the TTmSCE bit = 1.  |  |        |        |         |         |         |         |
|   | TTmACL  | Clear level selection of encoder input signal (TENCm0 pin)   |        |        |         |         |         |         |
|   | 0   | Clears low level of the TENCm0 pin.  |        |        |         |         |         |         |
|   | 1   | Clears high level of the TENCm0 pin.   |        |        |         |         |         |         |
|   | Setting of the TTmACL bit is valid only when the TTmSCE bit = 1.  |  |        |        |         |         |         |         |
| <b>Note</b> V850E/IG3 only  |   |  |        |        |         |         |         |         |

| TTmECS1 | TTmECS0 | Valid edge setting of encoder clear signal (TECRm pin) |
|---------|---------|--|
| 0       | 0       | Detects no edge (clearing encoder is invalid).         |
| 0       | 1       | Detects rising edge.                                   |
| 1       | 0       | Detects falling edge.                                  |
| 1       | 1       | Detects both edges.                                    |

| TTmEIS1 | TTmEIS0 | Valid edge setting of encoder input signals (TENCm0, TENCm1 pins) |
|---------|---------|---|
| 0       | 0       | Detects no edge (inputting encoder is invalid).                   |
| 0       | 1       | Detects rising edge.  |
| 1       | 0       | Detects falling edge.   |
| 1       | 1       | Detects both edges.   |

- Cautions**
1. Rewrite the TTmSCE, TTmZCL, TTmBCL, TTmACL, TTmECS1, TTmECS0, TTmEIS1, and TTmEIS0 bits when the TTmCTL0.TTmCE bit = 0. (The same value can be written to these bits when the TTmCE bit = 1.) If rewriting was mistakenly performed, clear the TTmCE bit to 0 and then set these bits again.
  2. The TTmECS1 and TTmECS0 bits are valid only when the TTmSCE bit = 0 and the encoder compare mode is set.
  3. The TTmEIS1 and TTmEIS0 bits are valid only when the TTmCTL2.TTmUDS1 and TTmCTL2.TTmUDS0 bits = 00 or 01.

**(8) TMTn option register 0 (TTnOPT0)**

The TTnOPT0 register is an 8-bit register that sets the capture/compare operation and detects 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: TT0OPT0 FFFFF587H, TT1OPT0 FFFFF5C7H

|         |   |   |                         |                         |   |   |   |        |
|---------|---|---|-------------------------|-------------------------|---|---|---|--------|
|         | 7 | 6 | 5                       | 4                       | 3 | 2 | 1 | <0>    |
| TTnOPT0 | 0 | 0 | TTmCCS1 <sup>Note</sup> | TTmCCS0 <sup>Note</sup> | 0 | 0 | 0 | TTnOVF |

V850E/IF3  
n = 0, 1  
m = 1

|   |  |
|---|--|
| TTmCCS1 <sup>Note</sup>   | TTmCCR1 register capture/compare selection                       |
| 0   | Compare register selected  |
| 1   | Capture register selected (cleared by the TTmCTL0.TTmCE bit = 0) |
| The TTmCCS1 bit setting is valid only in the free-running timer mode. |  |

V850E/IG3  
n = 0, 1  
m = 0, 1

|   |  |
|---|--|
| TTmCCS0 <sup>Note</sup>   | TTmCCR0 register capture/compare selection                       |
| 0   | Compare register selected  |
| 1   | Capture register selected (cleared by the TTmCTL0.TTmCE bit = 0) |
| The TTmCCS0 bit setting is valid only in the free-running timer mode. |  |

|   |  |
|---|--|
| TTnOVF  | TMTn overflow detection flag                     |
| Set (1)   | Overflow occurred                                |
| Reset (0)   | 0 written to TTnOVF bit or TTnCTL0.TTnCE bit = 0 |
| <ul style="list-style-type: none"> <li>The TTnOVF bit is set to 1 when the 16-bit counter value overflows from FFFFH to 0000H in the free-running timer mode or the pulse width measurement mode.</li> <li>An overflow interrupt request signal (INTTTIOVn) is generated at the same time that the TTnOVF bit is set to 1. The INTTTIOVn signal is not generated in modes other than the free-running timer mode and the pulse width measurement mode.</li> <li>The TTnOVF bit is not cleared to 0 even when the TTnOVF bit or the TTnOPT0 register are read when the TTnOVF bit = 1.</li> <li>Before clearing the TTnOVF bit to 0 after generation of the INTTTIOVn signal, be sure to confirm (by reading) that the TTnOVF bit is set to 1.</li> <li>The TTnOVF bit can be both read and written, but the TTnOVF bit cannot be set to 1 by software. Writing 1 has no effect on the operation of TMTn.</li> </ul> |  |

**Note** In the V850E/IF3, this bit can be set only in TMT1. Be sure to set bits 4 and 5 of TMT0 to “0”.

- Cautions**
1. Rewrite the TTmCCS1 and TTmCCS0 bits when the TTmCE bit = 0. (The same value can be written when the TTmCE bit = 1.) If rewriting was mistakenly performed, clear the TTmCE bit to 0 and then set these bits again.
  2. Be sure to set bits 1 to 3, 6, and 7 to “0”.

**(9) TMTm option register 1 (TTmOPT1)**

The TTmOPT1 register is an 8-bit register that detects the overflow, underflow, and count-up/down operation of the encoder count function.

The TTmOPT1 register is valid only in the encoder compare mode.

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

Reset sets this register to 00H.

This register can be rewritten even when the TTmCTL0.TTmCE bit = 1.

(1/2)

|   |   |   |   |   |   |        |        |        |        |                               |         |                   |           |  |
|---|---|---|---|---|---|--------|--------|--------|--------|-------------------------------|---------|-------------------|-----------|--|
| After reset: 00H    R/W    Address: TT0OPT1 FFFFF588H <sup>Note</sup> , TT1OPT1 FFFFF5C8H |   |   |   |   |   |        |        |        |        |                               |         |                   |           |  |
|   | 7   | 6 | 5 | 4 | 3 | <2>    | <1>    | <0>    |        |                               |         |                   |           |  |
| TTmOPT1   | 0   | 0 | 0 | 0 | 0 | TTmEUF | TTmEOF | TTmESF |        |                               |         |                   |           |  |
| [ V850E/IF3<br>m = 1 ]  | <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">TTmEUF</td> <td>TMTm underflow detection flag</td> </tr> <tr> <td>Set (1)</td> <td>Underflow occurs.</td> </tr> <tr> <td>Reset (0)</td> <td>Cleared by writing to TTmEUF bit or when TTmCTL0.TTmCE bit = 0</td> </tr> </table>  |   |   |   |   |        |        |        | TTmEUF | TMTm underflow detection flag | Set (1) | Underflow occurs. | Reset (0) | Cleared by writing to TTmEUF bit or when TTmCTL0.TTmCE bit = 0 |
| TTmEUF  | TMTm underflow detection flag   |   |   |   |   |        |        |        |        |                               |         |                   |           |  |
| Set (1)   | Underflow occurs.   |   |   |   |   |        |        |        |        |                               |         |                   |           |  |
| Reset (0)   | Cleared by writing to TTmEUF bit or when TTmCTL0.TTmCE bit = 0  |   |   |   |   |        |        |        |        |                               |         |                   |           |  |
| [ V850E/IG3<br>m = 0, 1 ]   | <ul style="list-style-type: none"> <li>• The TTmEUF bit is set to 1 when 16-bit counter underflows from 0000H to FFFFH in encoder compare mode.</li> <li>• When the TTmCTL2.TTmLDE bit = 1, TTmEUF bit is set to 1 when value of 16-bit counter is changed from 0000H to set value of the TTmCCR0 register.</li> <li>• Overflow interrupt request signal (INTTTIOVm) is generated as soon as the TTmEUF bit is set to 1.</li> <li>• The TTmEUF bit is not cleared to 0 even if the TTmEUF bit or TTmOPT1 register is read when the TTmEUF bit = 1.</li> <li>• Status of the TTmEUF bit is retained even if the TTmCTL0.TTmCE bit is cleared to 0 when the TTmCTL2.TTmECC bit = 1.</li> <li>• Before clearing the TTmEUF bit to 0 after the INTTTIOVm signal is generated, be sure to confirm (read) that the TTmEUF bit is set to 1.</li> <li>• The TTmEUF bit can be read or written, but it cannot be set to 1 by software. Setting this bit to 1 does not affect operation of TMTm.</li> </ul> |   |   |   |   |        |        |        |        |                               |         |                   |           |  |

**Note** V850E/IG3 only

| TTmEOF   | Overflow detection flag for TMTm encoder function                        |
|--|--|
| Set (1)  | Overflow occurs.   |
| Reset (0)  | Cleared by writing 0 to the TTmEOF bit or when the TTmCTL0.TTmCE bit = 0 |
| <ul style="list-style-type: none"> <li>• The TTmEOF bit is set to 1 when 16-bit counter overflows from FFFFH to 0000H in encoder compare mode.</li> <li>• As soon as the TTmEOF bit has been set to 1, an overflow interrupt request signal (INTTTIOVm) is generated. At this time, the TTmOPT0.TTmOVF bit is not set to 1.</li> <li>• The TTmEOF bit is not cleared to 0 even if the TTmEOF bit or TTmOPT1 register is read when the TTmEOF bit = 1.</li> <li>• Status of the TTmEOF bit is retained even if the TTmCTL0.TTmCE bit is cleared to 0 when the TTmCTL2.TTmECC bit = 1.</li> <li>• Before clearing the TTmEOF bit to 0 after the INTTTIOVm signal is generated, be sure to confirm (read) that the TTmEOF bit is set to 1.</li> <li>• The TTmEOF bit can be read or written, but it cannot be set to 1 by software. Writing 1 to this bit does not affect operation of TMTm.</li> </ul> |  |

| TTmESF  | TMTm count-up/-down operation status detection flag |
|---|---|
| 0   | TMTm is counting up.                                |
| 1   | TMTm is counting down.                              |
| <ul style="list-style-type: none"> <li>• This bit is cleared to 0 if the TTmCTL0.TTmCE bit = 0 when the TTmCTL2.TTmECC bit = 0.</li> <li>• Status of the TTmESF bit is retained even if the TTmCE bit = 0 when the TTmECC bit = 1.</li> </ul> |   |

**Caution** Be sure to set bits 3 to 7 to “0”.

**(10) TMTm capture input select register (TTISLm)**

The TTISLm register is used to select which of TITm0 or TITm1 pin is used to input a capture trigger input signal when the TTmCCR0 register is used as a capture register.

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

Reset makes this register undefined.

|                           |        |  |   |   |   |   |   |   |        |
|---------------------------|--------|--|---|---|---|---|---|---|--------|
| After reset: Undefined    | R/W    | Address: TTISL0 FFFFF5A4H <sup>Note</sup> , TTISL1 FFFFF5A6H |   |   |   |   |   |   |        |
|                           |        | 7  | 6 | 5 | 4 | 3 | 2 | 1 | 0      |
| TTISLm                    |        | 0  | 0 | 0 | 0 | 0 | 0 | 0 | TTISLm |
| [ V850E/IF3<br>m = 1 ]    | TTISLm | Capture trigger input signal selection of TTmCCR0 register   |   |   |   |   |   |   |        |
| [ V850E/IG3<br>m = 0, 1 ] | 0      | TITm0 input  |   |   |   |   |   |   |        |
|                           | 1      | TITm1 input  |   |   |   |   |   |   |        |

**Note** V850E/IG3 only



**(11) TMTn capture/compare register 0 (TTnCCR0)**

The TTmCCR0 register is a 16-bit register that can be used as a capture register or compare register depending on the mode. The TT0CCR0 register of the V850E/IF3 is a 16-bit registers that can only be used as a compare register.

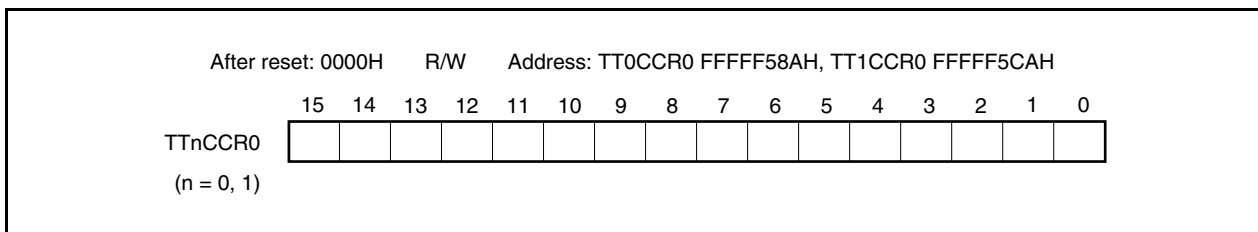
This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TTmOPT0.TTmCCS0 bit. In the pulse width measurement mode, the TTmCCR0 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TTnCCR0 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.

**Remark** V850E/IF3: n = 0, 1, m = 1  
 V850E/IG3: n = 0, 1, m = 0, 1



**(a) Function as compare register**

The TTnCCR0 register can be rewritten even when the TTnCTL0.TTnCE bit = 1.

The set value of the TTnCCR0 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 (INTTTEQCn0) is generated. If TOTm0 pin output is enabled at this time, the output of the TOTm0 pin is inverted.

When the TTnCCR0 register is used as a cycle register in the interval timer mode, or when the TTmCCR0 register is used as a cycle register in the external event count mode, external trigger pulse output mode, one-shot pulse output mode, PWM output mode, triangular-wave PWM output mode, or encoder compare mode, the value of the 16-bit counter is cleared (0000H) if its count value matches the value of the CCR0 buffer register.

The compare register is not cleared by setting the TTnCTL0.TTnCE bit to 0.

**(b) Function as capture register**

When the TTmCCR0 register is used as a capture register in the free-running timer mode (when the TTmCCR0 register is used as a capture register), the count value of the 16-bit counter is stored in the TTmCCR0 register if the valid edge of the capture trigger input pin (TITm0 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TTmCCR0 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TITm0 pin) is detected. Even if the capture operation and reading the TTmCCR0 register conflict, the correct value of the TTmCCR0 register can be read.

The capture register is cleared by setting the TTmCTL0.TTmCE bit to 0.

**Remark** V850E/IF3: m = 1  
 V850E/IG3: m = 0, 1

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

**Table 8-4. Function of Capture/Compare Register in Each Mode and How to Write Compare Register**

| Operation Mode                                  | TTnCCR0 Register         | How to Write Compare Register |
|---|--------------------------|-------------------------------|
| Interval timer                                  | Compare register         | Anytime write                 |
| External event counter <sup>Note 1</sup>        | Compare register         | Anytime write                 |
| External trigger pulse output <sup>Note 1</sup> | Compare register         | Batch write <sup>Note 2</sup> |
| One-shot pulse output <sup>Note 1</sup>         | Compare register         | Anytime write                 |
| PWM output <sup>Note 1</sup>                    | Compare register         | Batch write <sup>Note 2</sup> |
| Free-running timer                              | Capture/compare register | Anytime write                 |
| Pulse width measurement <sup>Note 1</sup>       | Capture register         | None                          |
| Triangular-wave WPM output <sup>Note 1</sup>    | Compare register         | Batch write <sup>Note 2</sup> |
| Encoder compare <sup>Note 1</sup>               | Compare register         | Anytime write                 |

**Notes 1.** In the V850E/IF3, this mode can be set only in TMT1.

**2.** Writing to the TTnCCR1 register is the trigger.

**Remark** For anytime write and batch write, see **8.6 (2) Anytime write and batch write.**

**(12) TMTn capture/compare register 1 (TTnCCR1)**

The TTmCCR1 register is a 16-bit register that can be used as a capture register or compare register depending on the mode. The TT0CCR1 register of the V850E/IF3 is a 16-bit registers that can only be used as a compare register.

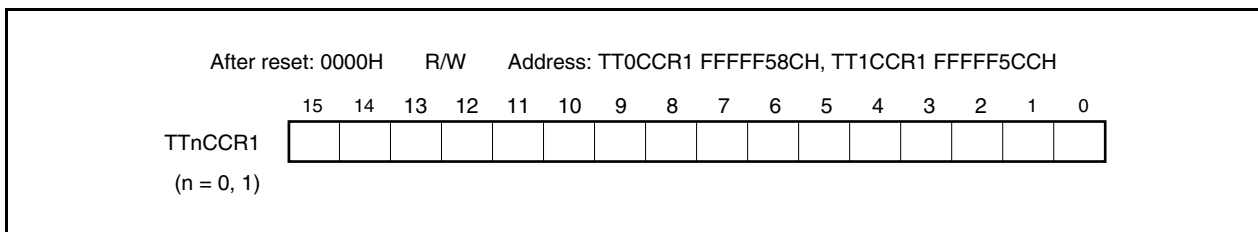
This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TTmOPT0.TTmCCS1 bit. In the pulse width measurement mode, the TTmCCR1 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TTnCCR1 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.

**Remark** V850E/IF3: n = 0, 1, m = 1  
 V850E/IG3: n = 0, 1, m = 0, 1



**(a) Function as compare register**

The TTnCCR1 register can be rewritten even when the TTnCTL0.TTnCE bit = 1.

The set value of the TTnCCR1 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 (INTTTEQCn1) is generated. If TOTm1 pin output is enabled at this time, the output of the TOTm1 pin is inverted.

The compare register is not cleared by setting the TTnCTL0.TTnCE bit to 0.

**(b) Function as capture register**

When the TTnCCR1 register is used as a capture register in the free-running timer mode (when the TTmCCR1 register is used as a capture register), the count value of the 16-bit counter is stored in the TTmCCR1 register if the valid edge of the capture trigger input pin (TITm1 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TTmCCR1 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TITm1 pin) is detected. Even if the capture operation and reading the TTmCCR1 register conflict, the correct value of the TTmCCR1 register can be read.

The capture register is cleared by setting the TTmCTL0.TTmCE bit to 0.

**Remark** V850E/IF3: m = 1  
 V850E/IG3: m = 0, 1

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

**Table 8-5. Function of Capture/Compare Register in Each Mode and How to Write Compare Register**

| Operation Mode                                  | TTnCCR1 Register         | How to Write Compare Register |
|---|--------------------------|-------------------------------|
| Interval timer                                  | Compare register         | Anytime write                 |
| External event counter <sup>Note 1</sup>        | Compare register         | Anytime write                 |
| External trigger pulse output <sup>Note 1</sup> | Compare register         | Batch write <sup>Note 2</sup> |
| One-shot pulse output <sup>Note 1</sup>         | Compare register         | Anytime write                 |
| PWM output <sup>Note 1</sup>                    | Compare register         | Batch write <sup>Note 2</sup> |
| Free-running timer                              | Capture/compare register | Anytime write                 |
| Pulse width measurement <sup>Note 1</sup>       | Capture register         | None                          |
| Triangular-wave PWM output <sup>Note 1</sup>    | Compare register         | Batch write <sup>Note 2</sup> |
| Encoder compare <sup>Note 1</sup>               | Compare register         | Anytime write                 |

**Notes 1.** In the V850E/IF3, this mode can be set only in TMT1.

**2.** Writing to the TTnCCR1 register is the trigger.

**Remark** For anytime write and batch write, see **8.6 (2) Anytime write and batch write**.

**(13) TMTm counter write register (TTmTCW)**

The TTmTCW register is used to set the initial value of the 16-bit counter.

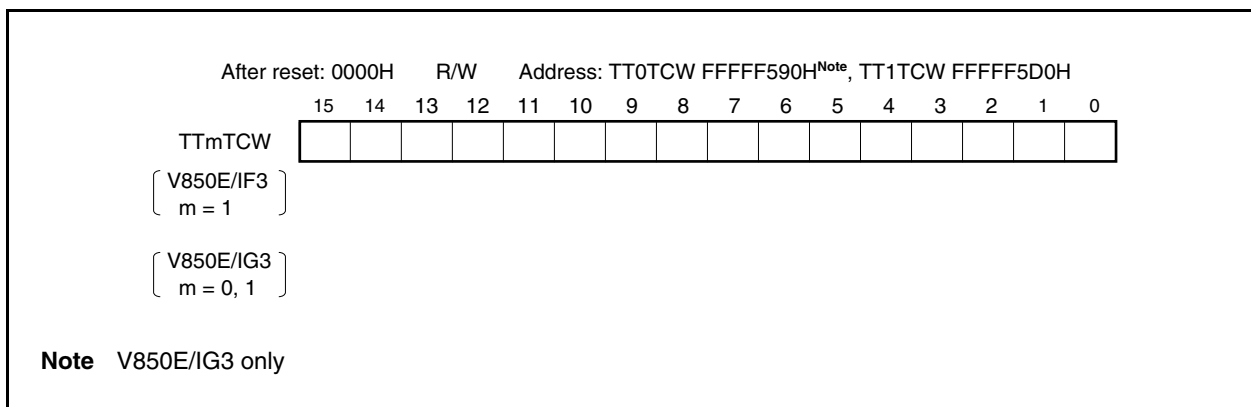
The TTmTCW register is valid only in the encoder compare mode.

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

Rewrite the TTmTCW register when the TTmCTL0.TTmCE bit = 0.

The value of the TTmTCW register is transferred to the 16-bit counter when the TTmCE bit is set (1).

Reset sets this register to 0000H.



**(14) TMTn counter read buffer register (TTnCNT)**

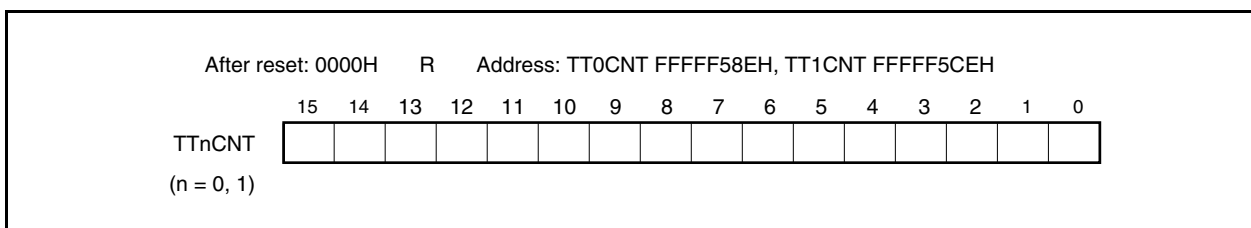
The TTnCNT register is a read buffer register that can read the count value of the 16-bit counter.

If this register is read when the TTnCTL0.TTnCE 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 TTmCNT register is set to 0000H when the TTmCTL2.TTmECC and TTmCE bits = 0. If the TTmCNT register is read at this time, the value of the 16-bit counter (FFFFH) is not read, but 0000H is read. The TTmCNT register is not set to 0000H but the previous value is read when the TTmECC bit = 1 and TTmCE bit = 0.

The TTmECC and TTmCE bits are set to 0 after reset, and the value of the TTmCNT register is set to 0000H.



## 8.5 Timer Output Operations

The following table shows the operations and output levels of the TOTm0 and TOTm1 pins.

**Table 8-6. Timer Output Control in Each Mode**

| Operation Mode                                     | TOTm1 Pin                                       | TOTm0 Pin  |
|--|---|------------|
| Interval timer mode                                | PWM output                                      |            |
| External event count mode <sup>Note</sup>          | None  |            |
| External trigger pulse output mode <sup>Note</sup> | External trigger pulse output                   | PWM output |
| One-shot pulse output mode <sup>Note</sup>         | One-shot pulse output                           |            |
| PWM output mode <sup>Note</sup>                    | PWM output                                      |            |
| Free-running timer mode                            | PWM output (only when compare function is used) |            |
| Pulse width measurement mode <sup>Note</sup>       | None  |            |
| Triangular-wave PWM output mode <sup>Note</sup>    | Triangular-wave PWM output                      |            |
| Encoder compare mode <sup>Note</sup>               | None  |            |

**Note** In the V850E/IF3, this mode can be set only in TMT1.

**Remark** V850E/IF3: m = 1, a = 0, 1  
 V850E/IG3: m = 0, 1, a = 0, 1

**Table 8-7. Truth Table of TOTm0 and TOTm1 Pins Under Control of Timer Output Control Bits**

| TTmIOC0.TTmOLa Bit | TTmIOC0.TTmOEa Bit | TTmCTL0.TTmCE Bit | Level of TOTma 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** V850E/IF3: m = 1, a = 0, 1  
 V850E/IG3: m = 0, 1, a = 0, 1

## 8.6 Operation

The functions of TMT<sub>n</sub> that can be implemented differ from one channel to another. The functions of each channel are shown below (n = 0, 1).

**Table 8-8. TMT<sub>m</sub> Specifications in Each Mode**

| Operation                          | TTmCTL1.TTmEST Bit<br>(Software Trigger Bit) | EVTTm Pin<br>(External Trigger Input) | Capture/Compare<br>Register Setting | Compare Register<br>Write Method |
|------------------------------------|--|---------------------------------------|-------------------------------------|----------------------------------|
| Interval timer mode                | Invalid                                      | Invalid                               | Compare only                        | Anytime write                    |
| External event count mode          | Invalid                                      | Invalid                               | Compare only                        | Anytime write                    |
| External trigger pulse output mode | Valid  | Valid                                 | Compare only                        | Batch write                      |
| One-shot pulse output mode         | Valid  | Valid                                 | Compare only                        | Anytime write                    |
| PWM output mode                    | Invalid                                      | Invalid                               | Compare only                        | Batch write                      |
| Free-running timer mode            | Invalid                                      | Invalid                               | Switchable                          | Anytime write                    |
| Pulse width measurement mode       | Invalid                                      | Invalid                               | Capture only                        | Not applicable                   |
| Triangular-wave PWM output mode    | Invalid                                      | Invalid                               | Compare only                        | Batch write                      |
| Encoder compare mode               | Invalid                                      | Invalid                               | Compare only                        | Anytime write                    |

**Remark** V850E/IF3: m = 1  
V850E/IG3: m = 0, 1

**Table 8-9. TMT0 Specifications in Each Mode of V850E/IF3**

| Operation                          | Software Trigger Bit | External Trigger Input | Capture/Compare<br>Register Setting | Compare Register<br>Write Method |
|------------------------------------|----------------------|------------------------|-------------------------------------|----------------------------------|
| Interval timer mode                | Invalid              | Invalid                | Compare only                        | Anytime write                    |
| External event count mode          | None                 |                        |                                     |                                  |
| External trigger pulse output mode | None                 |                        |                                     |                                  |
| One-shot pulse output mode         | None                 |                        |                                     |                                  |
| PWM output mode                    | None                 |                        |                                     |                                  |
| Free-running timer mode            | Invalid              | Invalid                | Compare only                        | Anytime write                    |
| Pulse width measurement mode       | None                 |                        |                                     |                                  |
| Triangular-wave PWM output mode    | None                 |                        |                                     |                                  |
| Encoder compare mode               | None                 |                        |                                     |                                  |

**Remark** TMT0 of the V850E/IF3 does not have timer input pins (TIT00, TIT01, TECR0, TENC00, TENC01, EVTT0) and timer output pins (TOT00, TOT01). It has interrupt request signals (INTTTEQC00, INTTTEQC01) indicating a match between the value of the 16-bit counter and the values of the TT0CCR0 and TT0CCR1 registers.

**(1) Counter basic operation**

This section explains the basic operation of the 16-bit counter. For details, refer to the description of the operation in each mode.

**Remark** V850E/IF3:  $n = 0, 1, m = 1$   
 V850E/IG3:  $n = 0, 1, m = 0, 1$

**(a) Counter start operation**

- In external event count mode  
 When the TTmCTL0.TTmCE bit is set from 0 to 1, the 16-bit counter is set to 0000H.  
 After that, it counts up to 0001H, 0002H, 0003H, ... each time the valid edge of external event count input (EVTTm) is detected.
- Encoder compare mode  
 The count operation is controlled by TENCm0 and TENCm1 phases.  
 When the 16-bit counter initial setting is performed by transferring the set value of the TTmTCW register to the 16-bit counter and the count operation is started. (When the TTmCTL2.TTmECC bit = 0, the TTmTCW register set value is transferred to the 16-bit counter at the timing when the TTmCTL0.TTmCE bit changes from 0 to 1.)
- Triangular-wave PWM mode  
 The 16-bit counter starts counting from the initial value FFFFH.  
 It counts up FFFFH, 0000H, 0001H, 0002H, 0003H, and so on.  
 Following count up operation, the counter counts down upon a match between the 16-bit count value and the CCR0 buffer register.
- Mode other than above  
 The 16-bit counter starts counting from the initial value FFFFH.  
 It counts up FFFFH, 0000H, 0001H, 0002H, 0003H, and so on.

**(b) Clear operation**

The 16-bit counter is cleared to 0000H when its value matches the value of the compare register and cleared, when the value of the 16-bit counter is captured and cleared, when the edge of the encoder clear signal is detected and cleared, and when the clear level condition of the TENCm0, TENCm1, and TECRm pins is detected and cleared. The count operation from FFFFH to 0000H that takes place immediately after the counter has started counting or when the counter overflows is not a clearing operation. Therefore, the INTTTEQCn0 and INTTTEQCn1 interrupt signals are not generated.

**(c) Overflow operation**

The 16-bit counter overflows when the counter counts up from FFFFH to 0000H in the free-running mode, pulse width measurement mode, and encoder compare mode. If the counter overflows, the TTnOPT0.TTnOVF bit is set to 1 and an interrupt request signal (INTTTIOVn) is generated in the free-running mode and pulse width measurement mode.

If the counter overflows, the TTnOPT1.TTnEOF bit is set to 1 and an interrupt request signal (INTTTIOVn) is generated in the encoder compare mode.

Note that the INTTTIOVn signal is not generated under the following conditions.

- Immediately after a count operation has been started
- If the counter value matches the compare value FFFFH and is cleared
- When FFFFH is captured and cleared to 0000H in the pulse width measurement mode



**Caution** After the overflow interrupt request signal (INTTTIOVn) has been generated, be sure to check that the overflow flag (TTnOVF, TTmEOF bits) is set to 1.

**(d) Count value holding operation**

The value of the 16-bit counter is held by the TTmCTL2.TTmECC bit in the encoder compare mode. The value of the 16-bit counter is reset to FFFFH when the TTmECC bit = 0 and TTmCTL0.TTmCE bit = 0. When the TTmCE bit is set to 1 next time, the set value of the TTmTCW register is transferred to the 16-bit counter and the counter continues its count operation.

If the TTmECC bit = 1 and TTmCE bit = 0, the value of the 16-bit counter is held. When the TTmCE bit is set to 1 next time, the counter resumes the count operation from the held value.

**(e) Counter read operation during count operation**

The value of the 16-bit counter of TMTn can be read by using the TTnCNT register during the count operation. When the TTnCTL0.TTnCE bit = 1, the value of the 16-bit counter can be read by reading the TTnCNT register. If the TTnCNT register is read when the TTmCTL2.TTmECC bit = 0 and TTmCE bit = 0, however, it is 0000H. The held value of the TTmCNT register is read if the register is read when the TTmECC bit = 1 and TTmCE bit = 0.

**(f) Underflow operation**

The 16-bit counter underflow occurs at the timing when the 16-bit counter value changes from 0000H to FFFFH in the encoder compare mode. When underflow occurs, the TTmOPT1.TTmEUF bit is set to 1 and an interrupt request signal (INTTTIOVm) is generated.

**(g) Interrupt operation**

TMTn generates the following four types of interrupt request signals.

- INTTTEQCn0 interrupt: This signal functions as a match interrupt request signal of the CCR0 buffer register and as a capture interrupt request signal to the TTnCCR0 register.
- INTTTEQCn1 interrupt: This signal functions as a match interrupt request signal of the CCR1 buffer register and as a capture interrupt request signal to the TTnCCR1 register.
- INTTTIOVn interrupt: This signal functions as an overflow interrupt request signal.
- INTTIECm interrupt: This signal functions as a valid edge detection interrupt request signal of the encoder clear input (TECRm pin).

**(2) Anytime write and batch write**

The TTnCCR0 and TTnCCR1 registers in TMTn can be rewritten during timer operation (TTnCTL0.TTnCE bit = 1), but the write method (anytime write, batch write) of the CCR0 and CCR1 buffer registers differs depending on the mode.

**(a) Anytime write**

In this mode, data is transferred at any time from the TTnCCR0 and TTnCCR1 registers to the CCR0 and CCR1 buffer registers during timer operation (n = 0, 1).

**Figure 8-3. Flowchart of Basic Operation for Anytime Write**

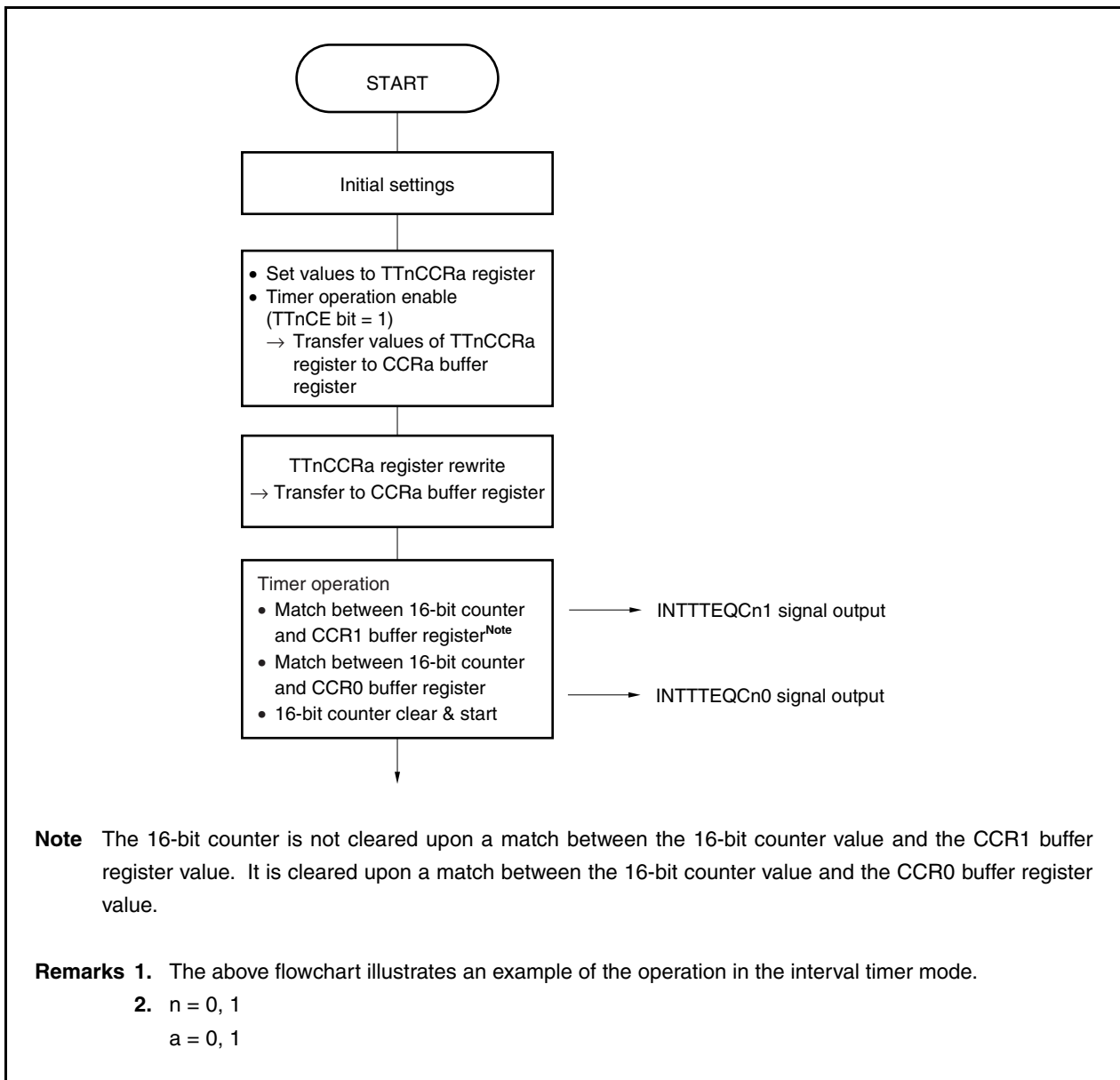
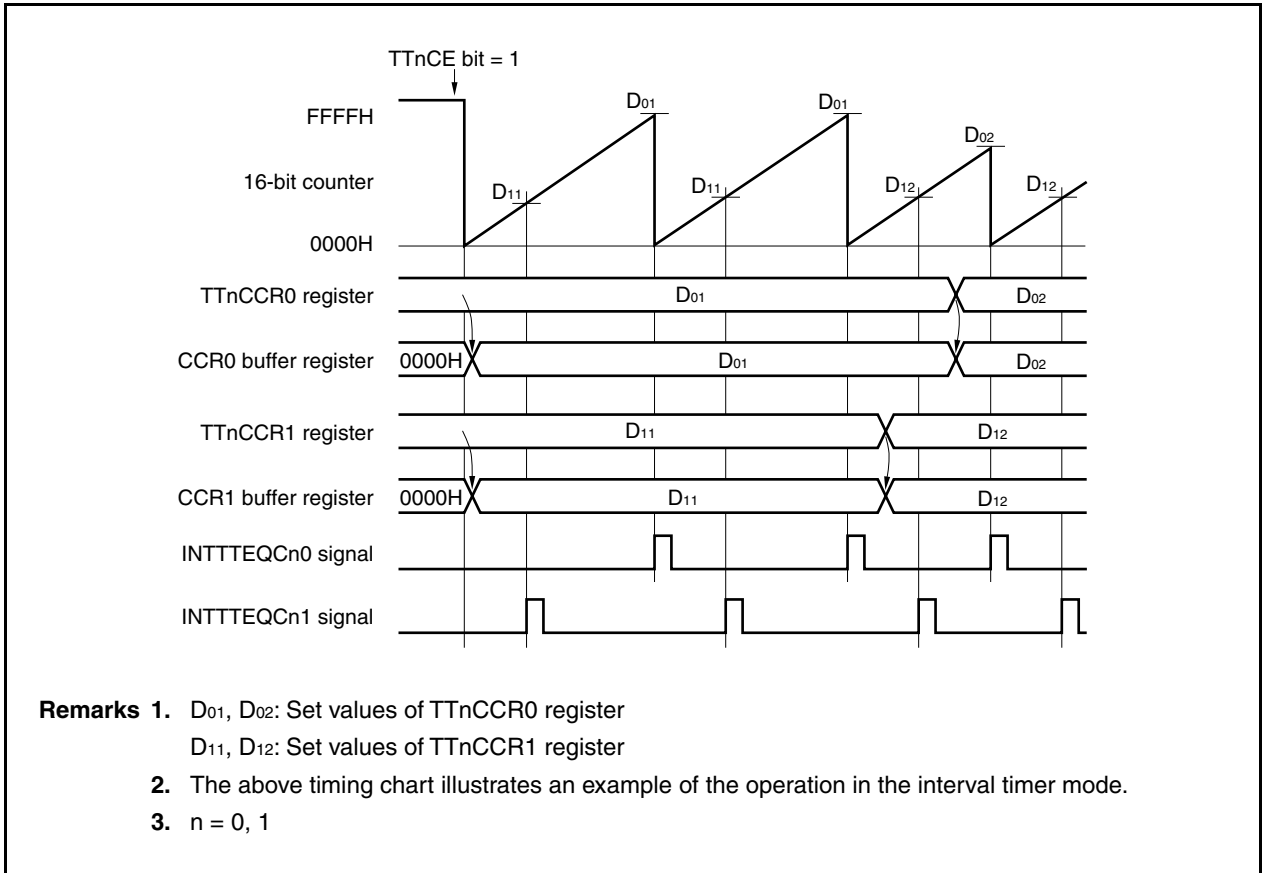


Figure 8-4. Timing of Anytime Write



**(b) Batch write**

In this mode, data is transferred all at once from the TTmCCR0 and TTmCCR1 registers to the CCR0 and CCR1 buffer registers during timer operation. This data is transferred upon a match between the value of the CCR0 buffer register and the value of the 16-bit counter. Transfer is enabled by writing to the TTmCCR1 register. Whether to enable or disable the next transfer timing is controlled by writing or not writing to the TTmCCR1 register.

In order for the set value when the TTmCCR0 and TTmCCR1 registers are rewritten to become the 16-bit counter comparison value (in other words, in order for this value to be transferred to the CCR0 and CCR1 buffer registers), it is necessary to rewrite the TTmCCR0 register and then write to the TTmCCR1 register before the 16-bit counter value and the CCR0 buffer register value match. Therefore, the values of the TTmCCR0 and TTmCCR1 registers are transferred to the CCR0 and CCR1 buffer registers upon a match between the count value of the 16-bit counter and the value of the CCR0 buffer register. Thus even when wishing only to rewrite the value of the TTmCCR0 register, also write the same value (same as preset value of the TTmCCR1 register) to the TTmCCR1 register.

**Remark** V850E/IF3:  $m = 1$   
V850E/IG3:  $m = 0, 1$

Figure 8-5. Flowchart of Basic Operation for Batch Write

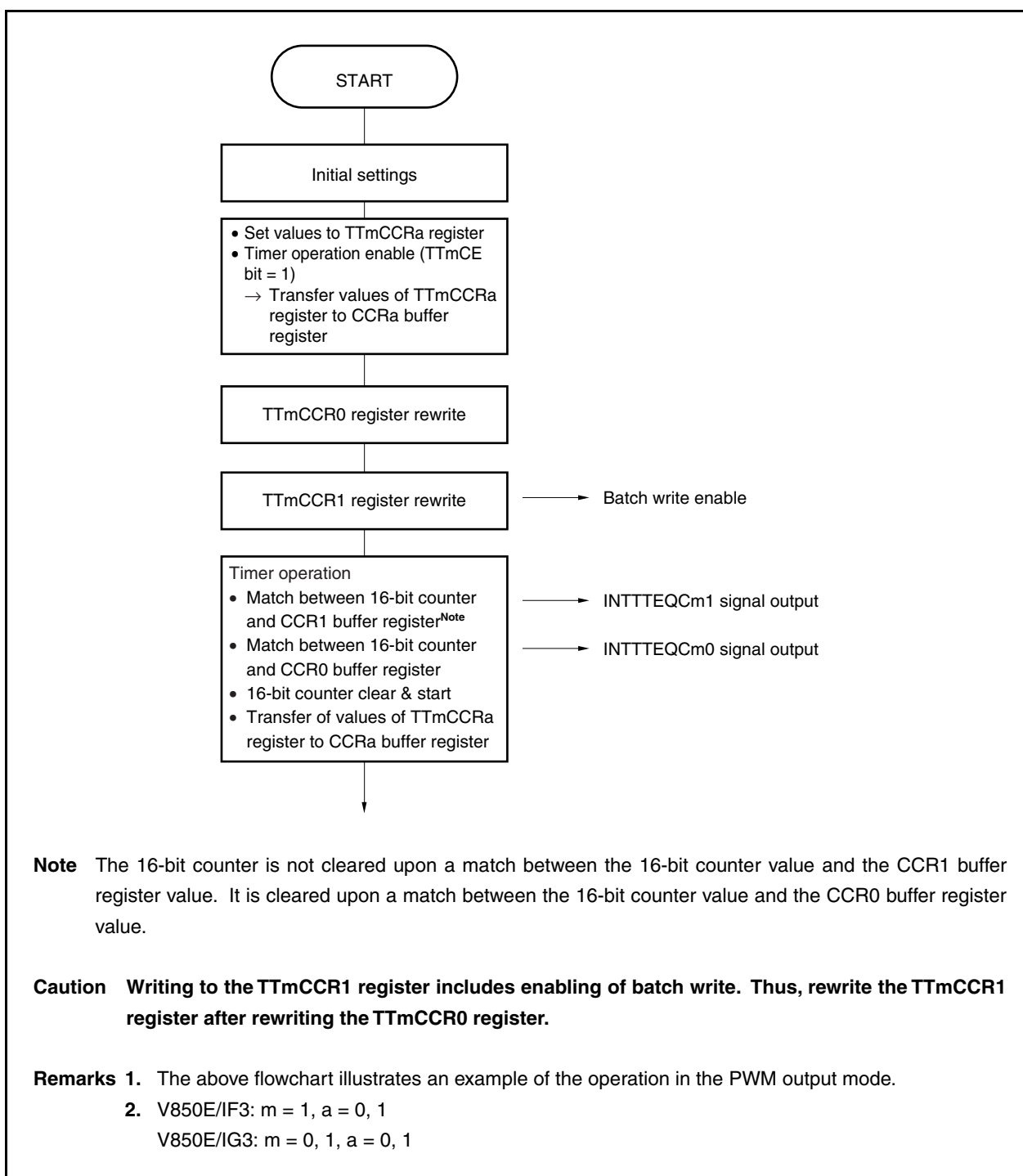
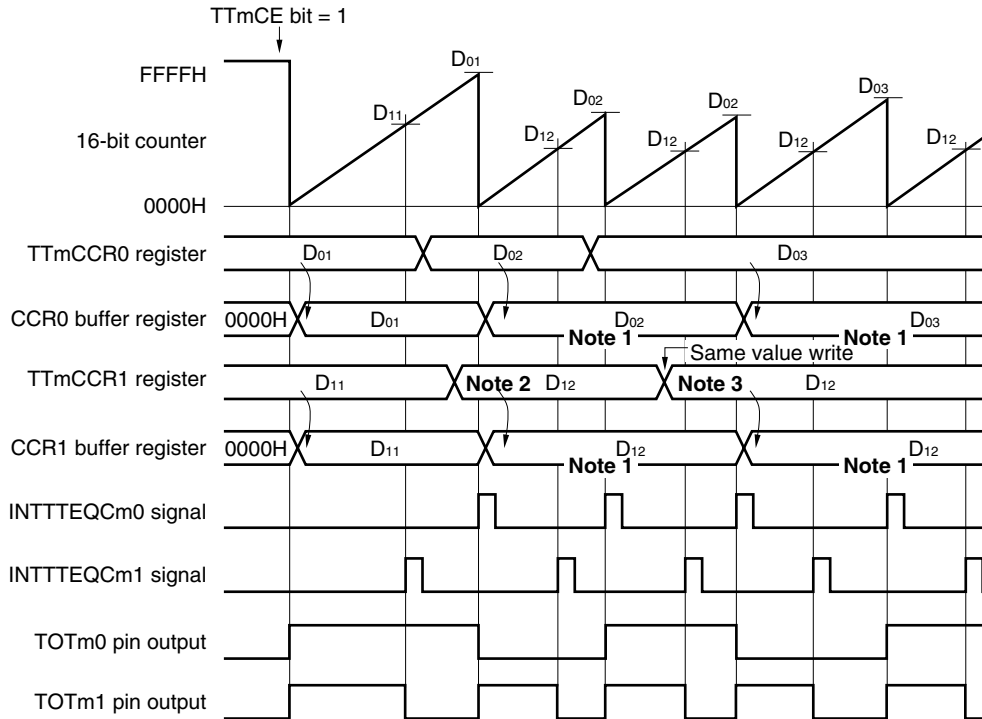


Figure 8-6. Timing of Batch Write



- Notes 1.** Because the TTmCCR1 register was not rewritten, D<sub>03</sub> is not transferred.
- 2.** Because the TTmCCR1 register has been written (D<sub>12</sub>), data is transferred to the CCR1 buffer register upon a match between the value of the 16-bit counter and the value of the TTmCCR0 register (D<sub>01</sub>).
- 3.** Because the TTmCCR1 register has been written (D<sub>12</sub>), data is transferred to the CCR1 buffer register upon a match between the value of the 16-bit counter and the value of the TTmCCR0 register (D<sub>02</sub>).

- Remarks 1.** D<sub>01</sub>, D<sub>02</sub>, D<sub>03</sub>: Set values of TTmCCR0 register  
 D<sub>11</sub>, D<sub>12</sub>: Set values of TTmCCR1 register
- 2.** The above timing chart illustrates the operation in the PWM output mode as an example.
- 3.** V850E/IF3: m = 1  
 V850E/IG3: m = 0, 1

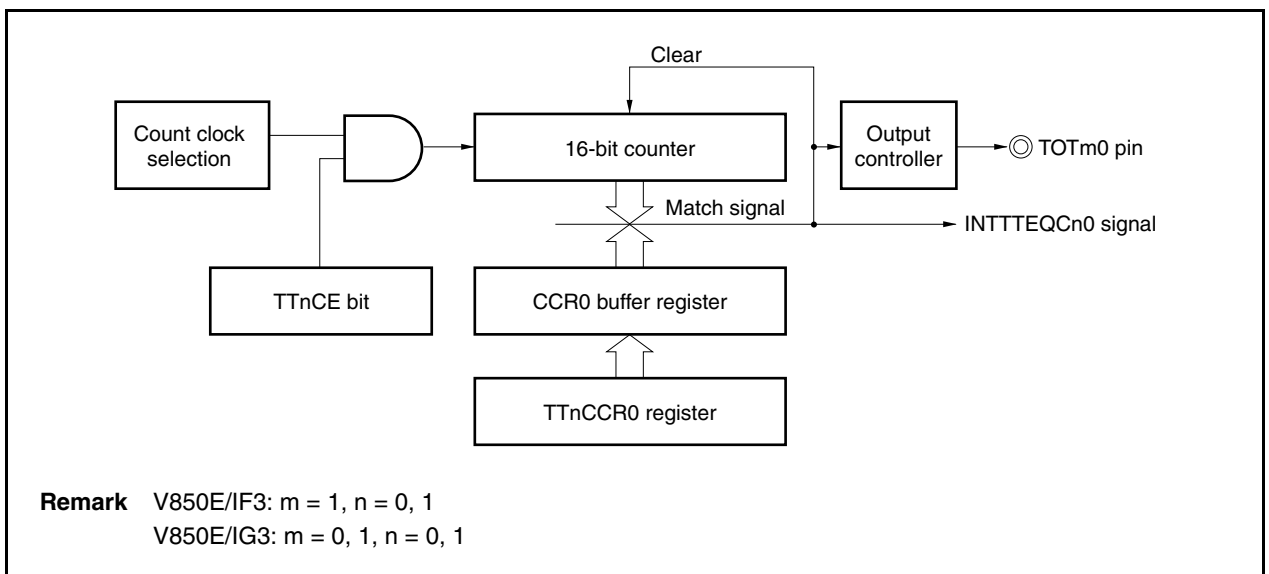
**8.6.1 Interval timer mode (TTnMD3 to TTnMD0 bits = 0000)**

In the interval timer mode, an interrupt request signal (INTTTEQCn0) is generated at the interval set by the TTnCCR0 register if the TTnCTL0.TTnCE bit is set to 1. A PWM waveform with a duty factor of 50% whose half cycle is equal to the interval can be output from the TOTm0 pin.

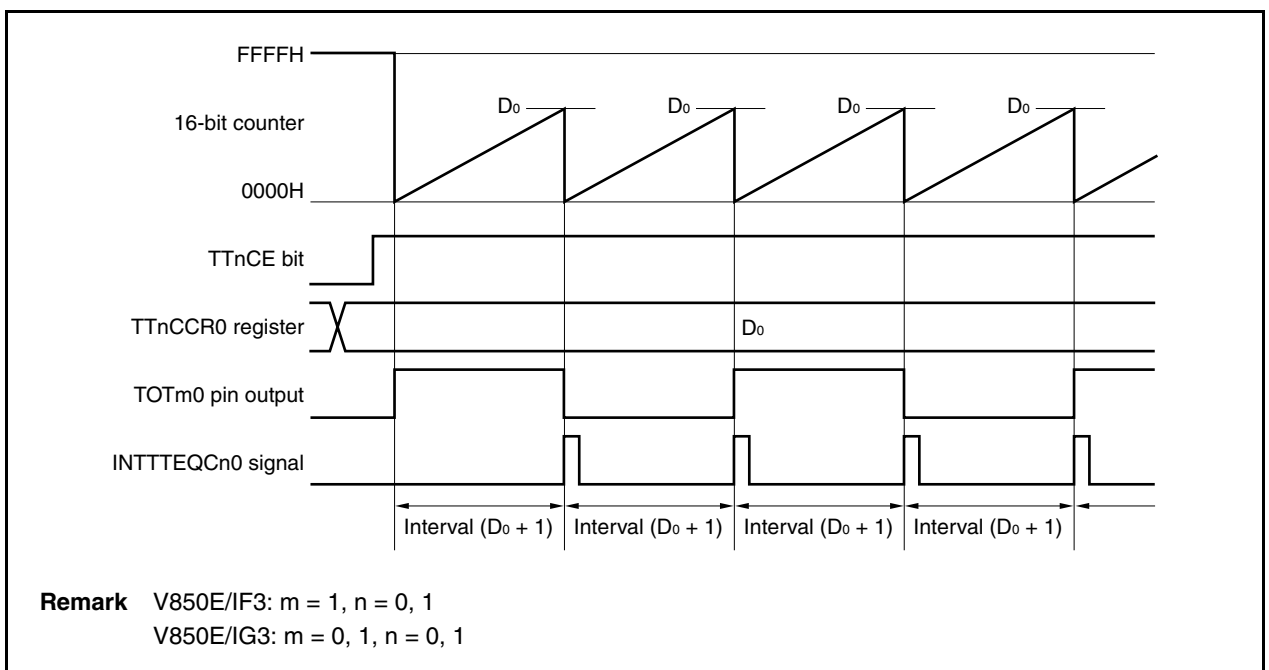
The TTnCCR1 register is not used in the interval timer mode. However, the set value of the TTnCCR1 register is transferred to the CCR1 buffer register, and when the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTTEQCn1) is generated. In addition, a PWM waveform with a duty factor of 50%, which is inverted when the INTTTEQCm1 signal is generated, can be output from the TOTm1 pin.

The value of the TTnCCR0 and TTnCCR1 registers can be rewritten even while the timer is operating.

**Figure 8-7. Configuration of Interval Timer**



**Figure 8-8. Basic Timing of Operation in Interval Timer Mode**



When the TTnCE 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 TOTm0 pin is inverted. Additionally, the set value of the TTnCCR0 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 TOTm0 pin is inverted, and a compare match interrupt request signal (INTTTEQCn0) is generated.

The interval can be calculated by the following expression.

$$\text{Interval} = (\text{Set value of TTnCCR0 register} + 1) \times \text{Count clock cycle}$$

**Remark** V850E/IF3: n = 0, 1, m = 1  
 V850E/IG3: n = 0, 1, m = 0, 1

Figure 8-9. Register Setting for Interval Timer Mode Operation (1/2)

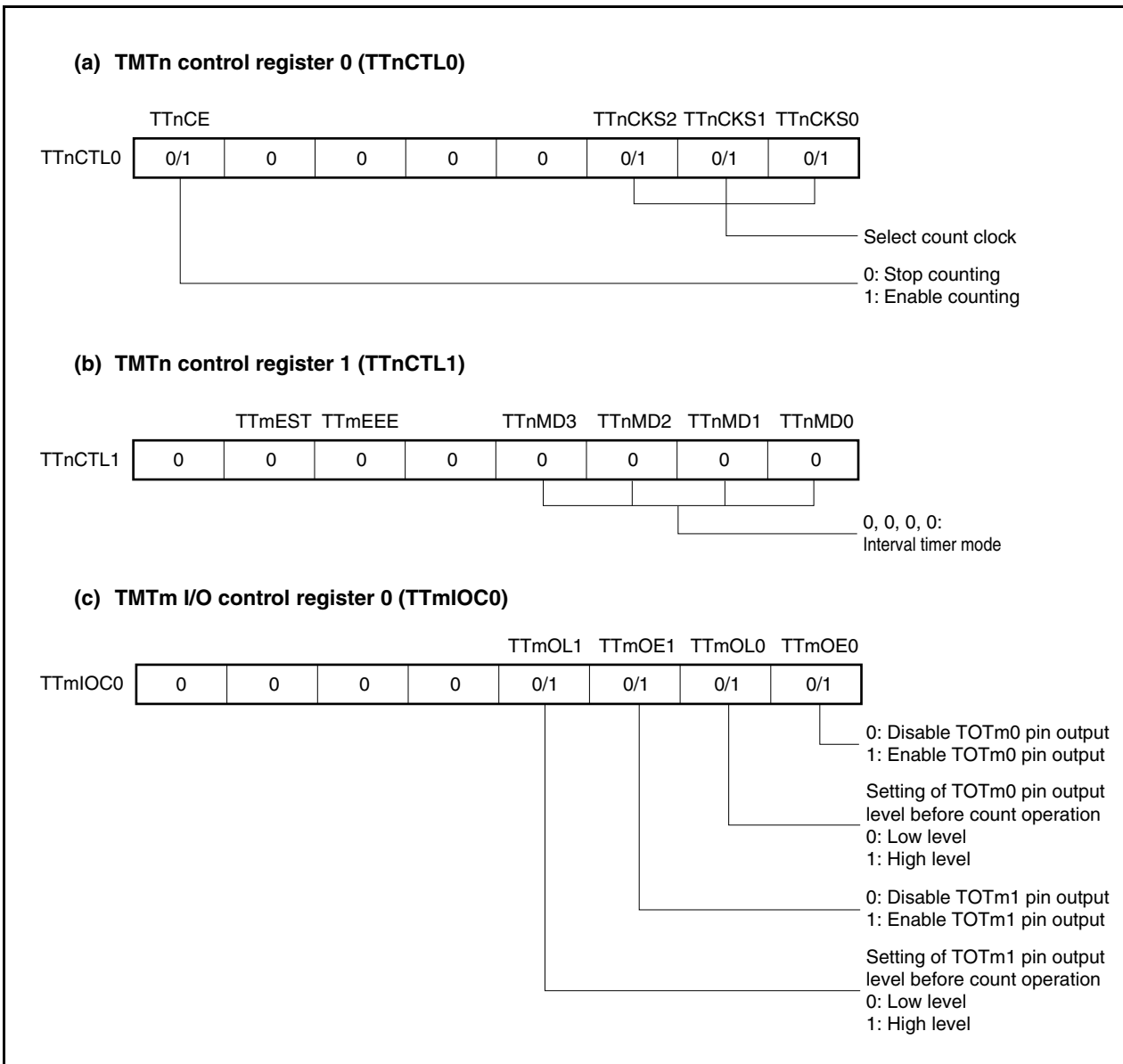




Figure 8-9. Register Setting for Interval Timer Mode Operation (2/2)

**(d) TMTn counter read buffer register (TTnCNT)**

By reading the TTnCNT register, the count value of the 16-bit counter can be read.

**(e) TMTn capture/compare register 0 (TTnCCR0)**

If the TTnCCR0 register is set to  $D_0$ , the interval is as follows.

$$\text{Interval} = (D_0 + 1) \times \text{Count clock cycle}$$

**(f) TMTn capture/compare register 1 (TTnCCR1)**

The TTnCCR1 register is not used in the interval timer mode. However, the set value of the TTnCCR1 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, the TOTm1 pin output is inverted and a compare match interrupt request signal (INTTTEQCn1) is generated.

By setting this register to the same value as the value set in the TTnCCR0 register, a PWM waveform with a duty factor of 50% can be output from the TOTm1 pin.

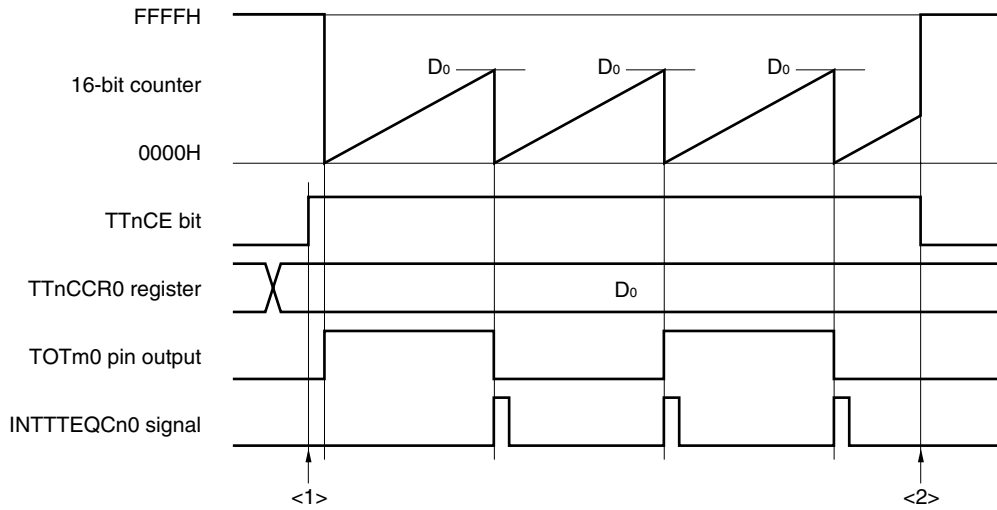
When the TTnCCR1 register is not used, it is recommended to set its value to FFFFH. Also mask the register by the interrupt mask flag (TTnCCIC1.TTnCCMK1).

**Remarks 1.** TMTm control register 2 (TTmCTL2), TMTm I/O control register 1 (TTmIOC1), TMTm I/O control register 2 (TTmIOC2), TMTm I/O control register 3 (TTmIOC3), TMTn option register 0 (TTnOPT0), TMTm option register 1 (TTmOPT1), TMTm capture input select register (TTISLm), and TMTm counter write register (TTmTCW) are not used in the interval timer mode.

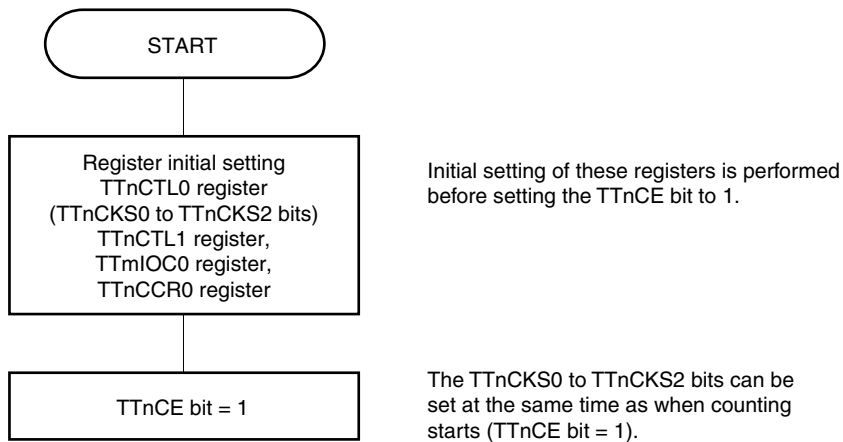
2. V850E/IF3: m = 1, n = 0, 1  
V850E/IG3: m = 0, 1, n = 0, 1

(1) Interval timer mode operation flow

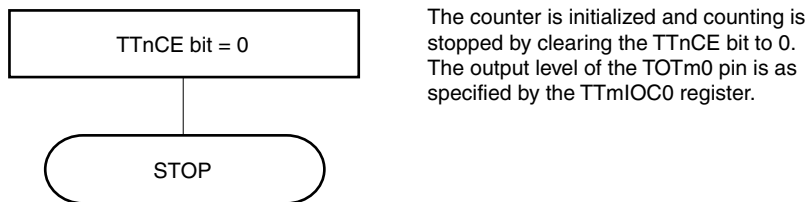
Figure 8-10. Software Processing Flow in Interval Timer Mode



<1> Count operation start flow



<2> Count operation stop flow



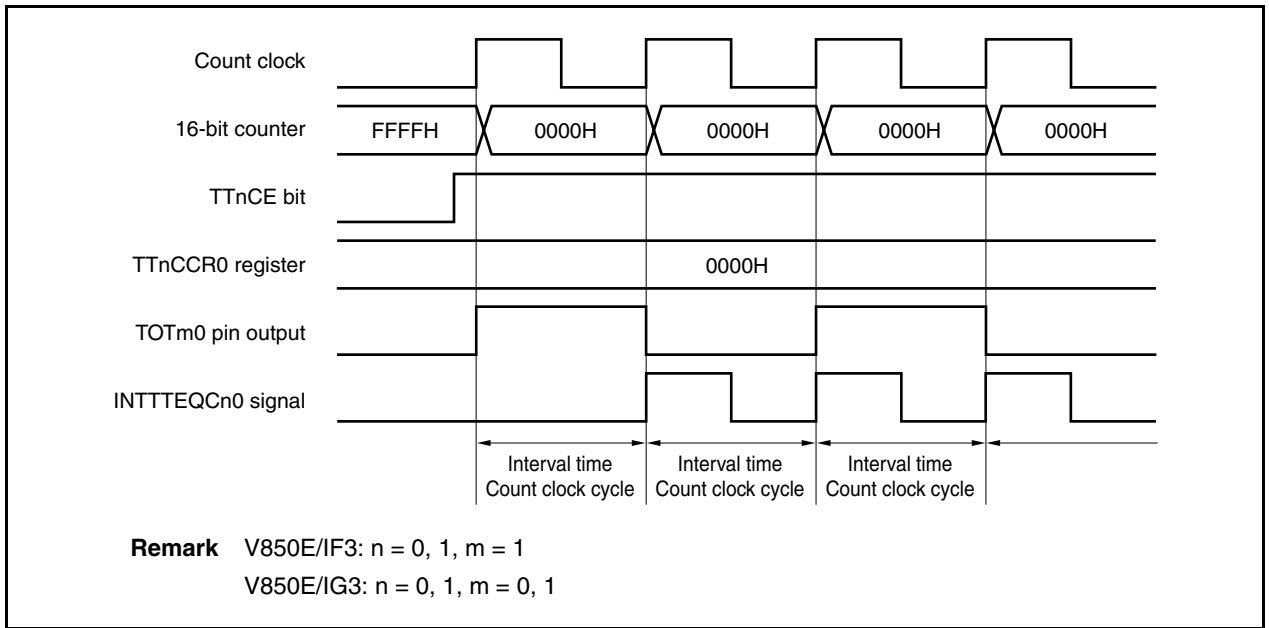
**Remark** V850E/IF3: n = 0, 1, m = 1  
V850E/IG3: n = 0, 1, m = 0, 1

(2) Interval timer mode operation timing

(a) Operation if TTnCCR0 register is set to 0000H

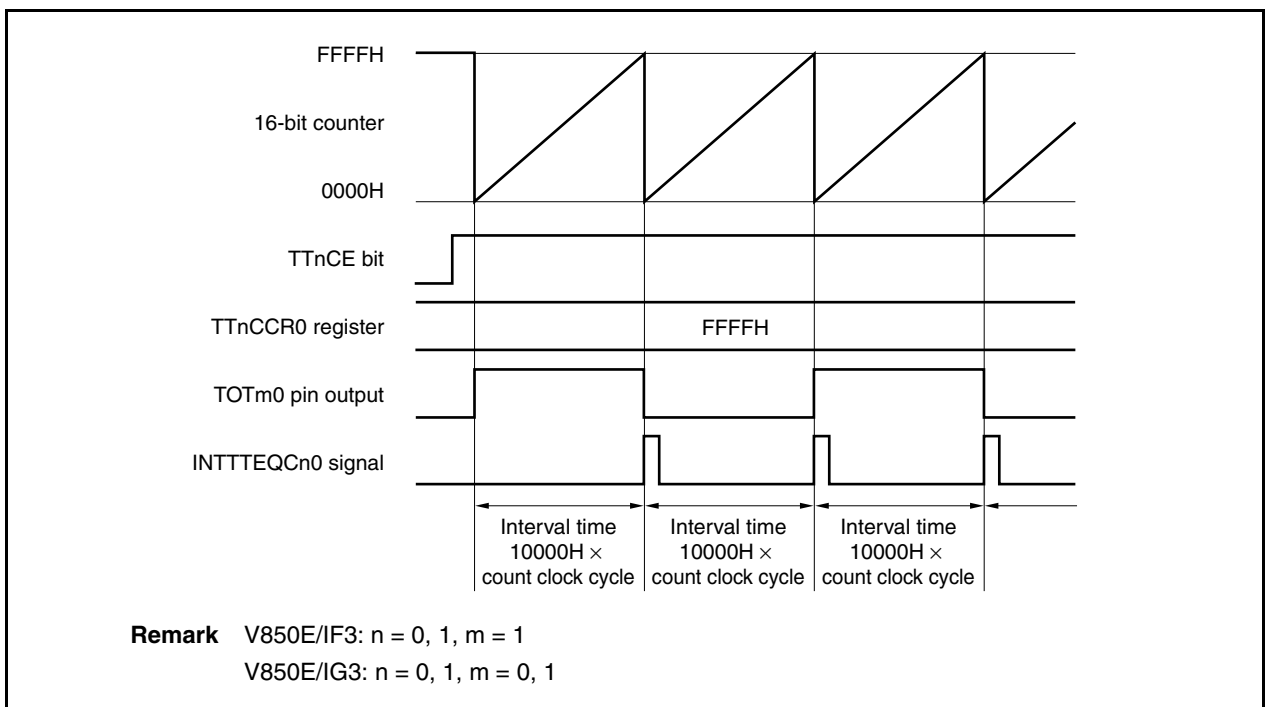
If the TTnCCR0 register is set to 0000H, the INTTTEQCn0 signal is generated at each count clock, and the output of the TOTm0 pin is inverted.

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



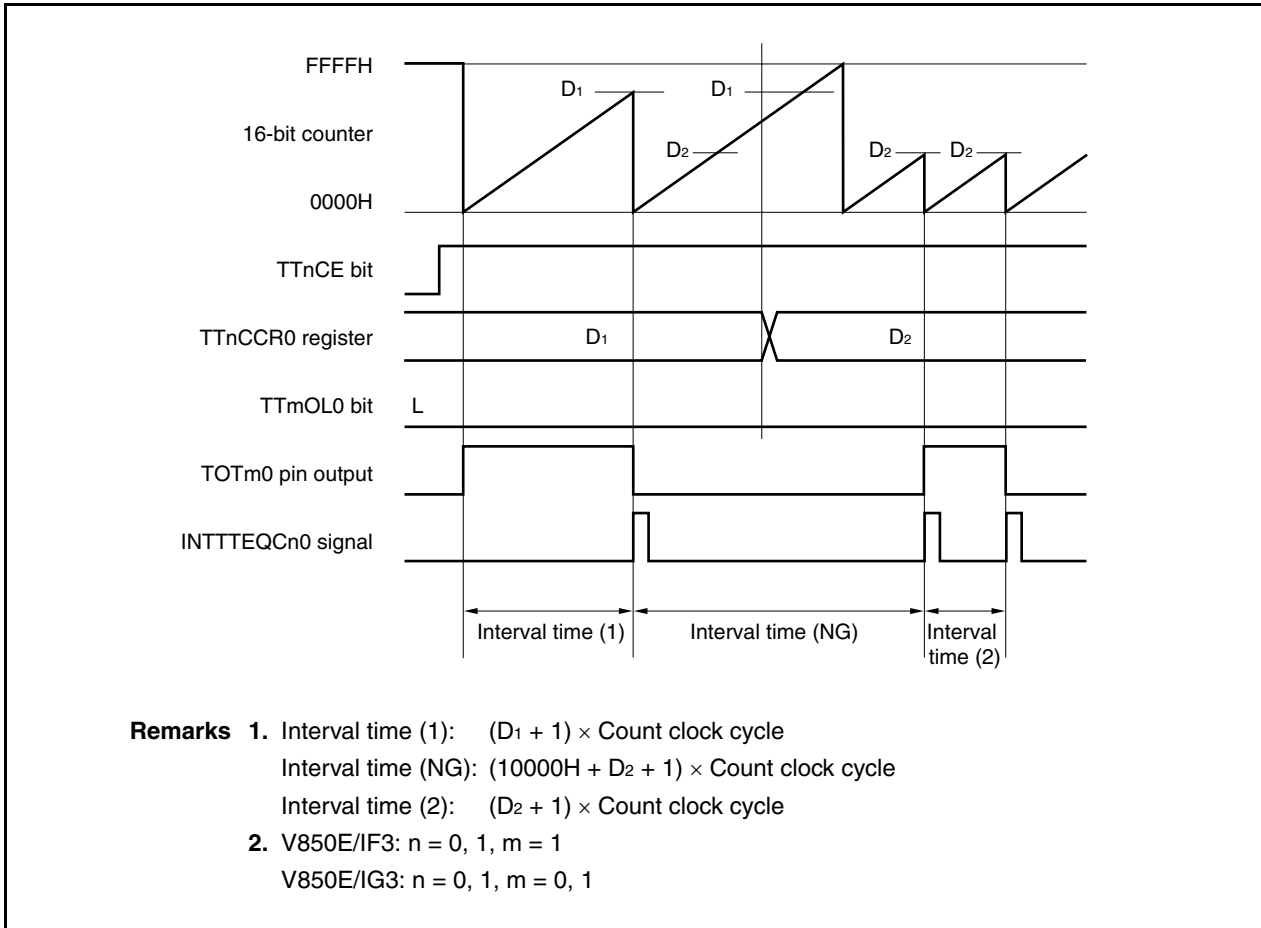
(b) Operation if TTnCCR0 register is set to FFFFH

If the TTnCCR0 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 INTTTEQCn0 signal is generated and the output of the TOTm0 pin is inverted. At this time, an overflow interrupt request signal (INTTTIOVn) is not generated, nor is the overflow flag (TTnOPT0.TTnOVF bit) set to 1.



(c) Notes on rewriting TTnCCR0 register

If the value of the TTnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When an overflow may occur, stop counting and then change the set value.



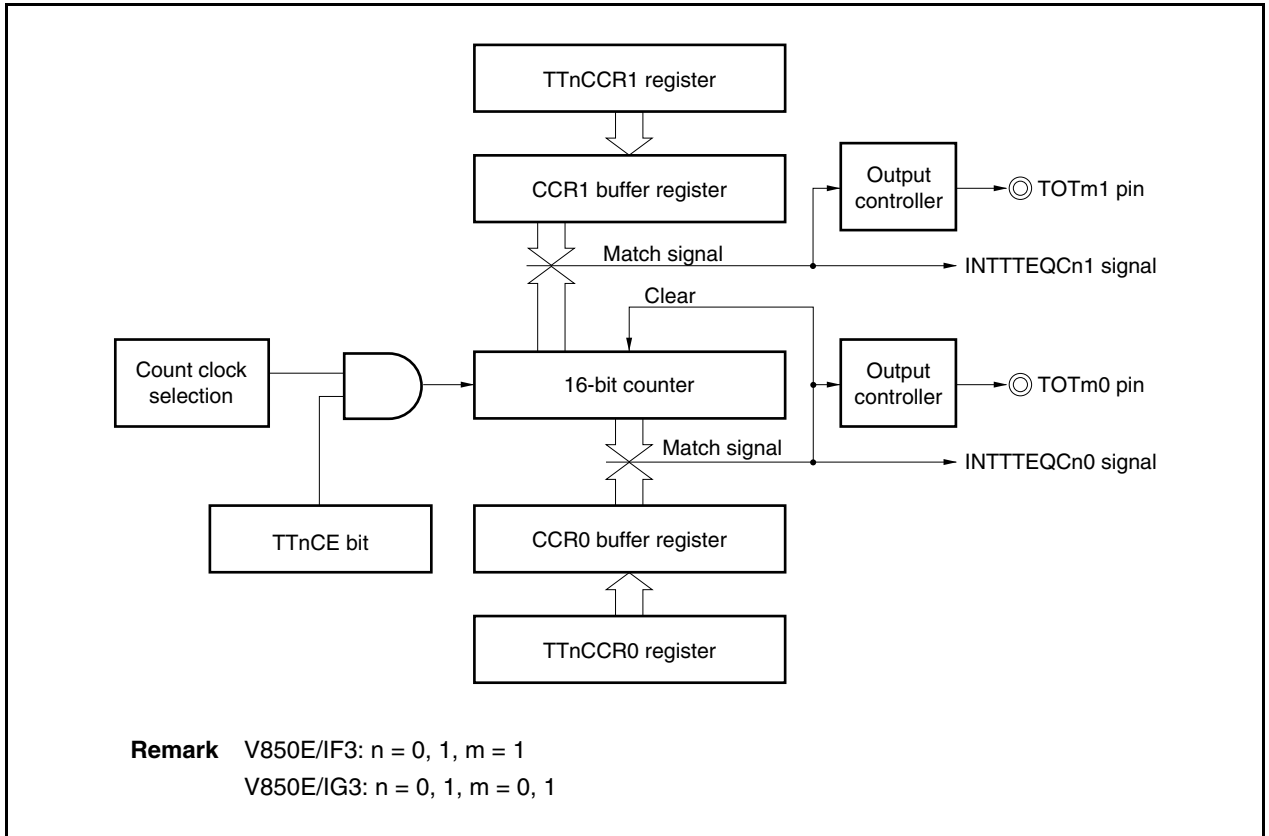
If the value of the TTnCCR0 register is changed from  $D_1$  to  $D_2$  while the count value is greater than  $D_2$  but less than  $D_1$ , the count value is transferred to the CCR0 buffer register as soon as the TTnCCR0 register has been rewritten. Consequently, the value of the 16-bit counter that is compared is  $D_2$ .

Because the count value has already exceeded  $D_2$ , however, the 16-bit counter counts up again from 0000H, overflows, and then counts up again from 0000H. When the count value matches  $D_2$ , the INTTTEQCn0 signal is generated and the output of the TOTm0 pin is inverted.

Therefore, the INTTTEQCn0 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 " $(10000H + D_2 + 1) \times \text{Count clock cycle}$ ".

(d) Operation of TTnCCR1 register

Figure 8-11. Configuration of TTnCCR1 Register



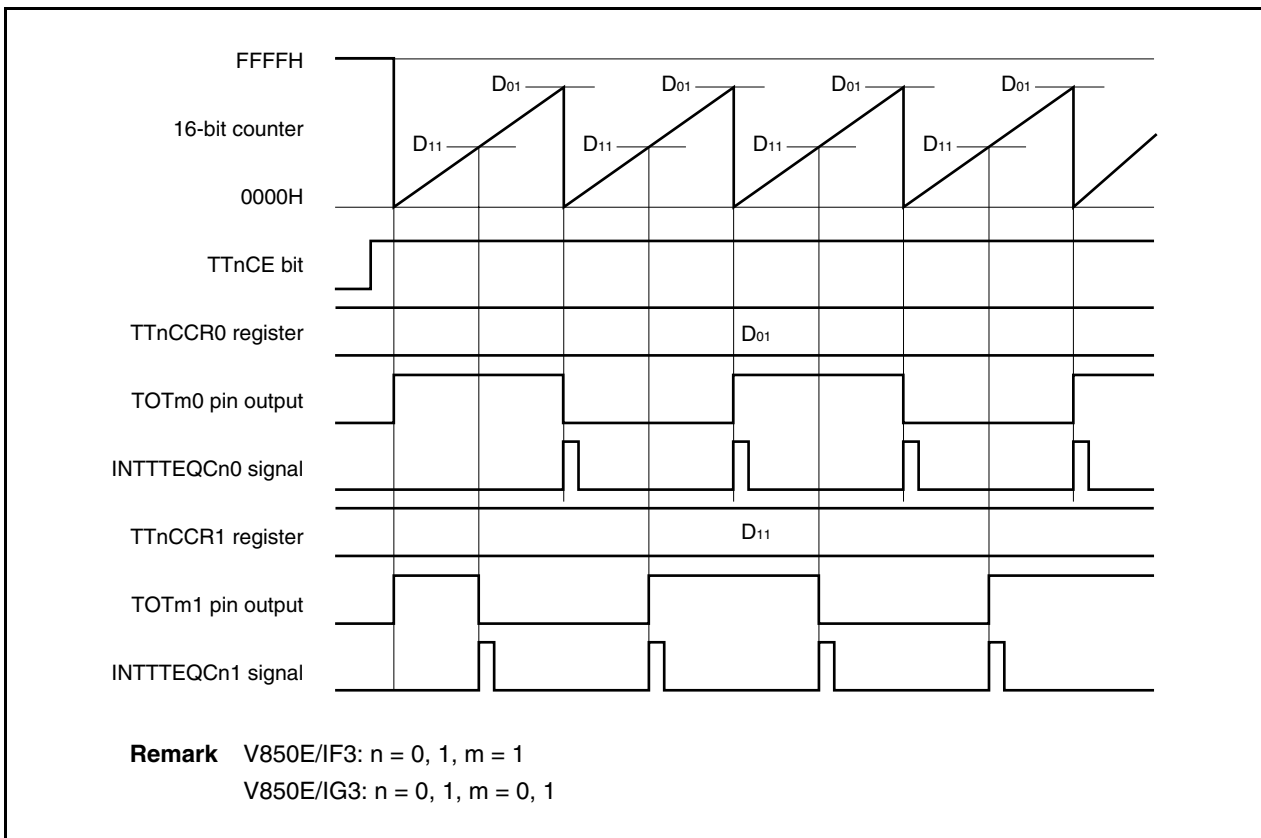
When the TTnCCR1 register is set to the same value as the TTnCCR0 register, the INTTTEQCn0 signal is generated at the same timing as the INTTTEQCn1 signal and the TOTm1 pin output is inverted. In other words, a PWM waveform with a duty factor of 50% can be output from the TOTm1 pin.

The following shows the operation when the TTnCCR1 register is set to other than the value set in the TTnCCR0 register.

If the set value of the TTnCCR1 register is less than the set value of the TTnCCR0 register, the INTTTEQCn1 signal is generated once per cycle. At the same time, the output of the TOTm1 pin is inverted.

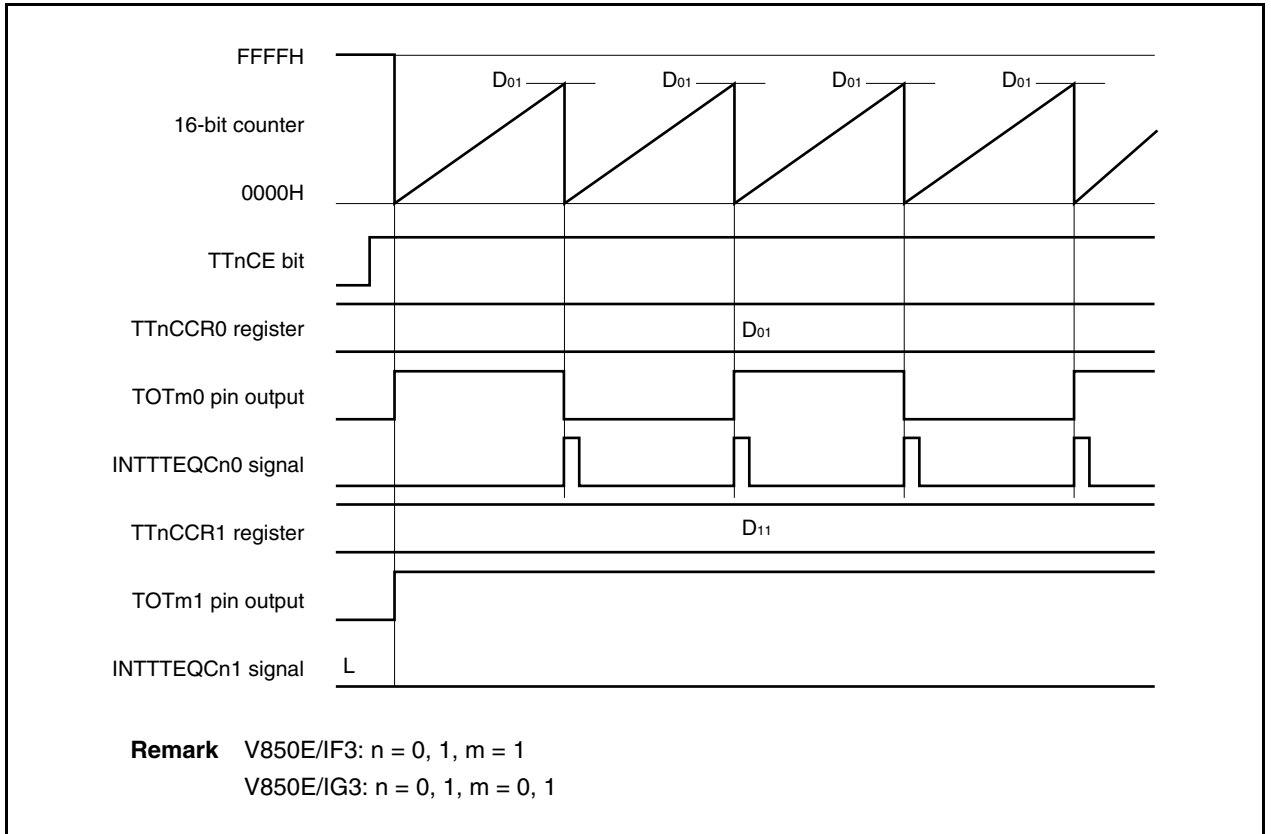
The TOTm1 pin outputs a PWM waveform with a duty factor of 50% after outputting a short-width pulse.

Figure 8-12. Timing Chart When  $D_{01} \geq D_{11}$



If the set value of the TTnCCR1 register is greater than the set value of the TTnCCR0 register, the count value of the 16-bit counter does not match the value of the TTnCCR1 register. Consequently, the INTTTEQCn1 signal is not generated, nor is the output of the TOTm1 pin changed. When the TTnCCR1 register is not used, it is recommended to set its value to FFFFH.

Figure 8-13. Timing Chart When  $D_{01} < D_{11}$



**8.6.2 External event count mode (TTmMD3 to TTmMD0 bits = 0001)**

This mode is valid only in TMT0 (V850E/IG3 only) and TMT1.

In the external event count mode, the valid edge of the external event count input (EVTTm) is counted when the TTmCTL0.TTmCE bit is set to 1, and an interrupt request signal (INTTTEQCm0) is generated each time the number of edges set by the TTmCCR0 register have been counted. The TOTm0 and TOTm1 pins cannot be used.

The TTmCCR1 register is not used in the external event count mode.

**Figure 8-14. Configuration in External Event Count Mode**

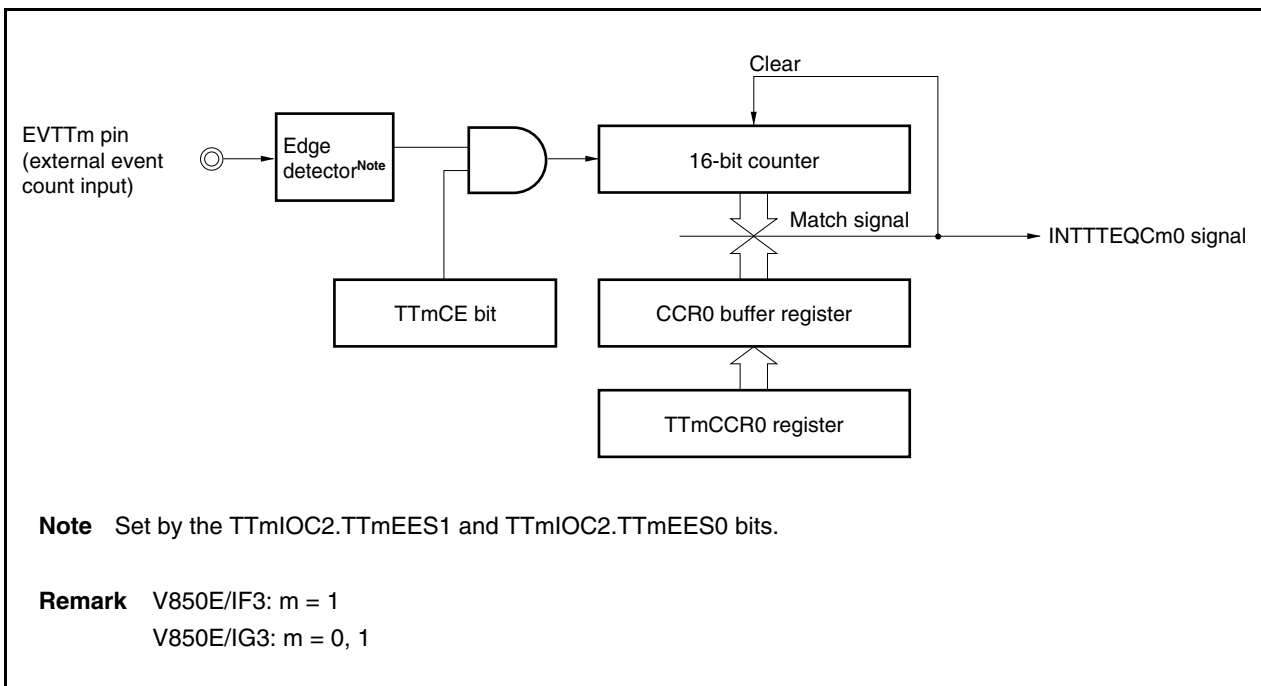
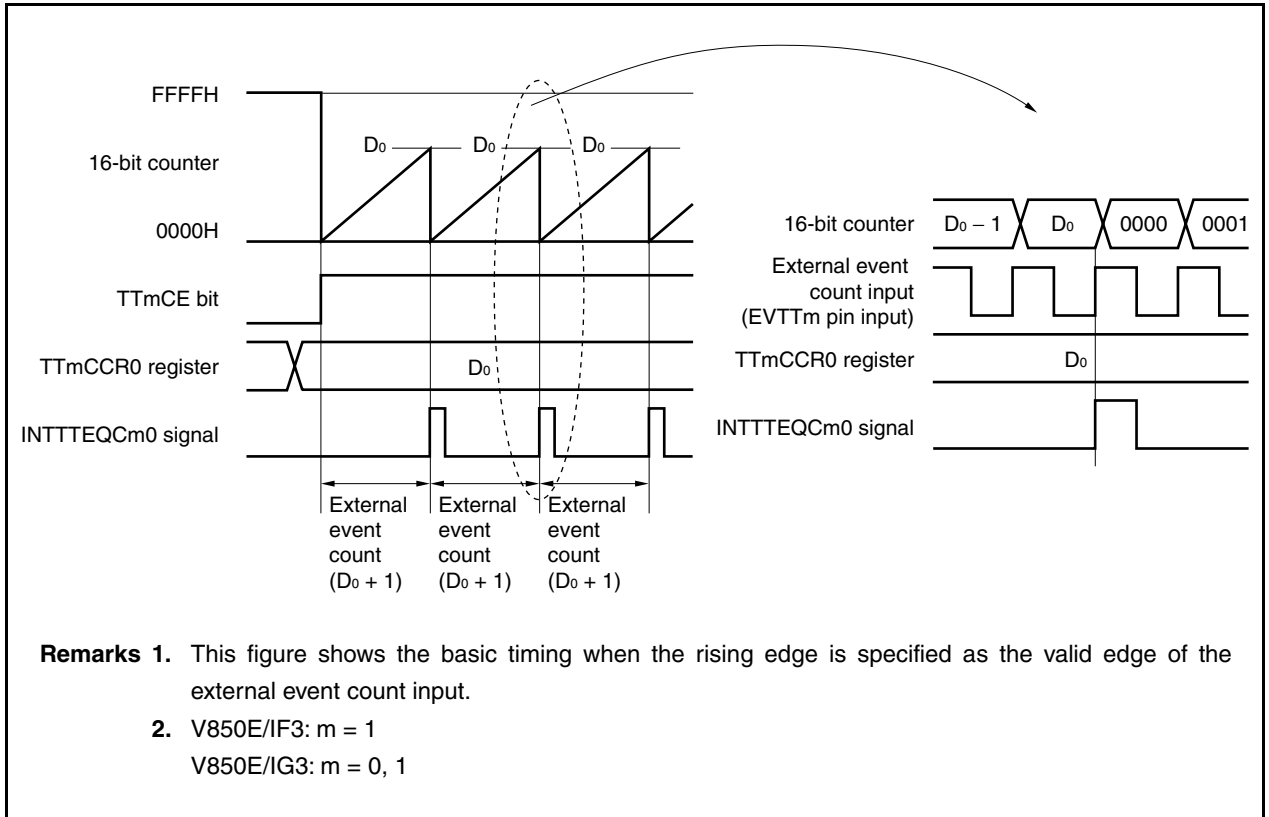




Figure 8-15. Basic Timing in External Event Count Mode



When the TTmCE 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 TTmCCR0 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 (INTTTEQCm0) is generated.

The INTTTEQCm0 signal is generated each time the valid edge of the external event count input has been detected “value set to TTmCCR0 register + 1” times.

**Figure 8-16. Register Setting for Operation in External Event Count Mode (1/2)**

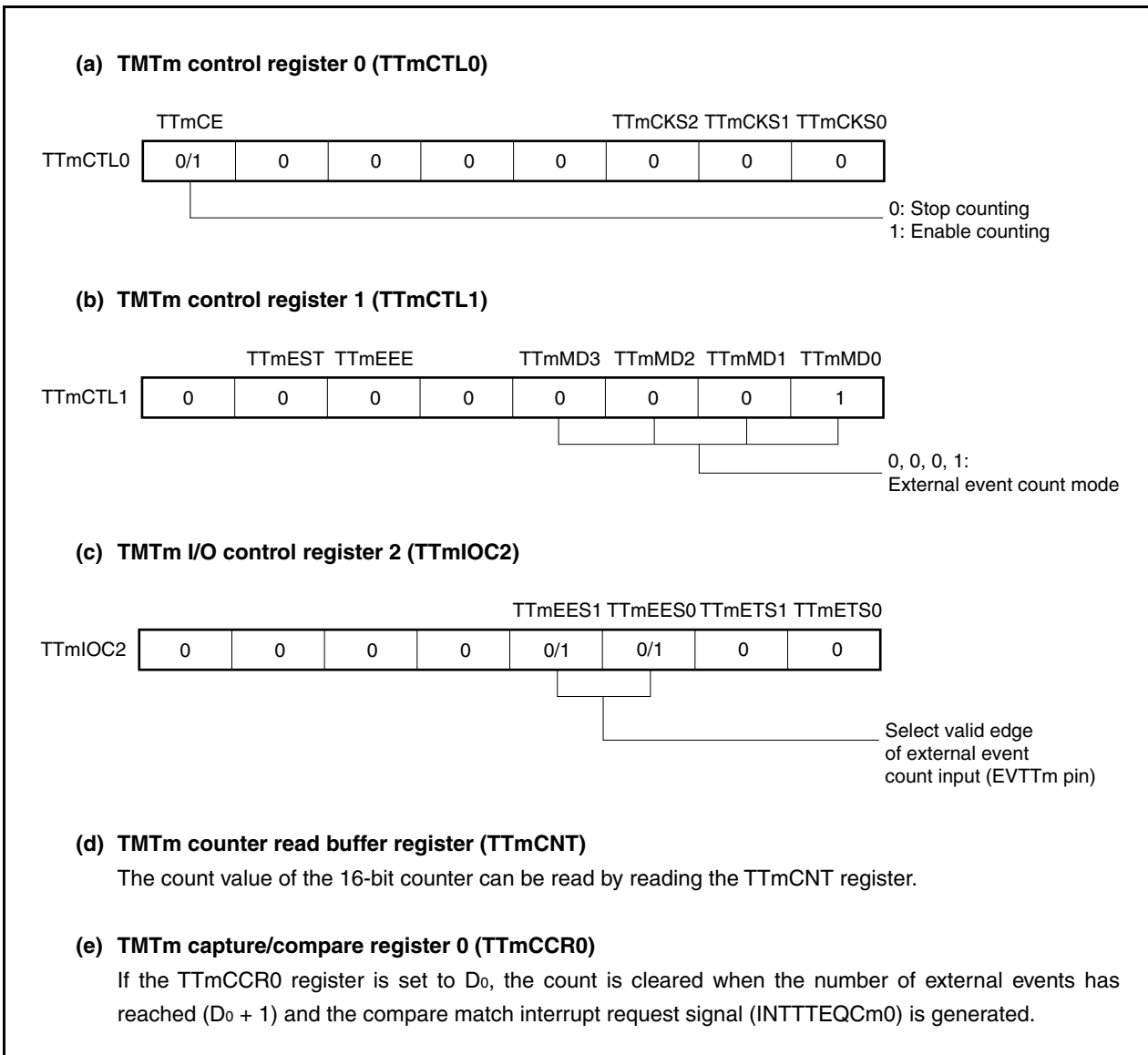


Figure 8-16. Register Setting for Operation in External Event Count Mode (2/2)

**(f) TMTm capture/compare register 1 (TTmCCR1)**

The TTmCCR1 register is not used in the external event count mode. However, the set value of the TTmCCR1 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 (INTTTEQCm1) is generated.

When the TTmCCR1 register is not used, it is recommended to set its value to FFFFH. Also mask the register by the interrupt mask flag (TTmCCIC1.TTmCCMK1).

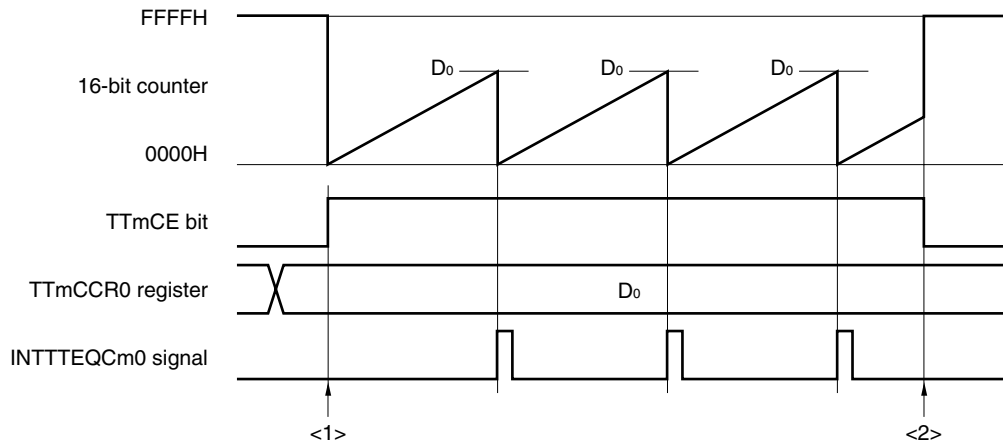
**Caution** Be sure to set the TTmIOC0 register to 00H.

**Remarks** 1. TMTm control register 2 (TTmCTL2), TMTm I/O control register 1 (TTmIOC1), TMTm I/O control register 3 (TTmIOC3), TMTm option register 0 (TTmOPT0), TMTm option register 1 (TTmOPT1), TMTm capture input select register (TTISLm), and TMTm counter write register (TTmTCW) are not used in the external event count mode.

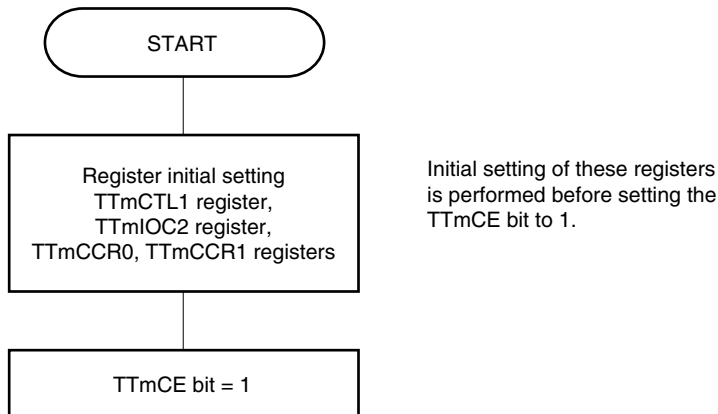
2. V850E/IF3: m = 1  
V850E/IG3: m = 0, 1

(1) External event count mode operation flow

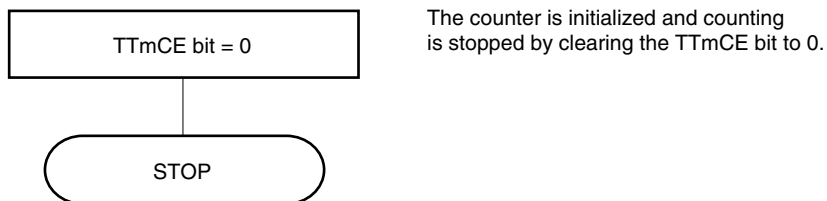
Figure 8-17. Software Processing Flow in External Event Count Mode



<1> Count operation start flow



<2> Count operation stop flow



**Remark** V850E/IF3: m = 1  
V850E/IG3: m = 0, 1

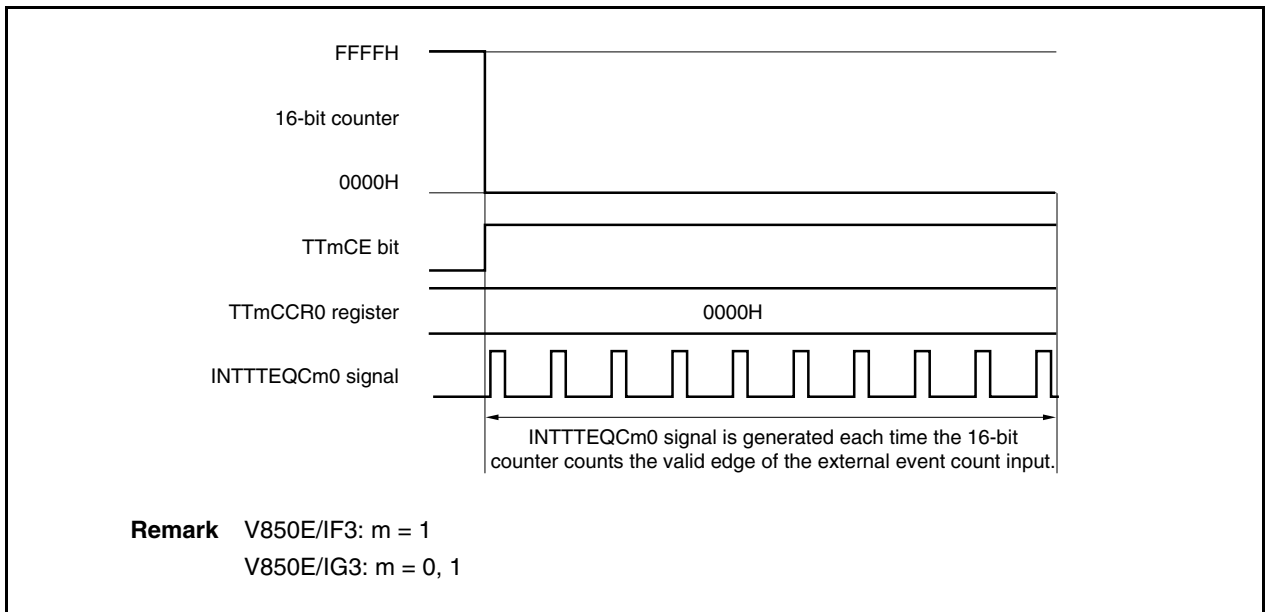
(2) Operation timing in external event count mode

**Caution** The use of timer output (TOTm0, TOTm1) is prohibited in the external event count mode.

(a) Operation if TTmCCR0 register is set to 0000H

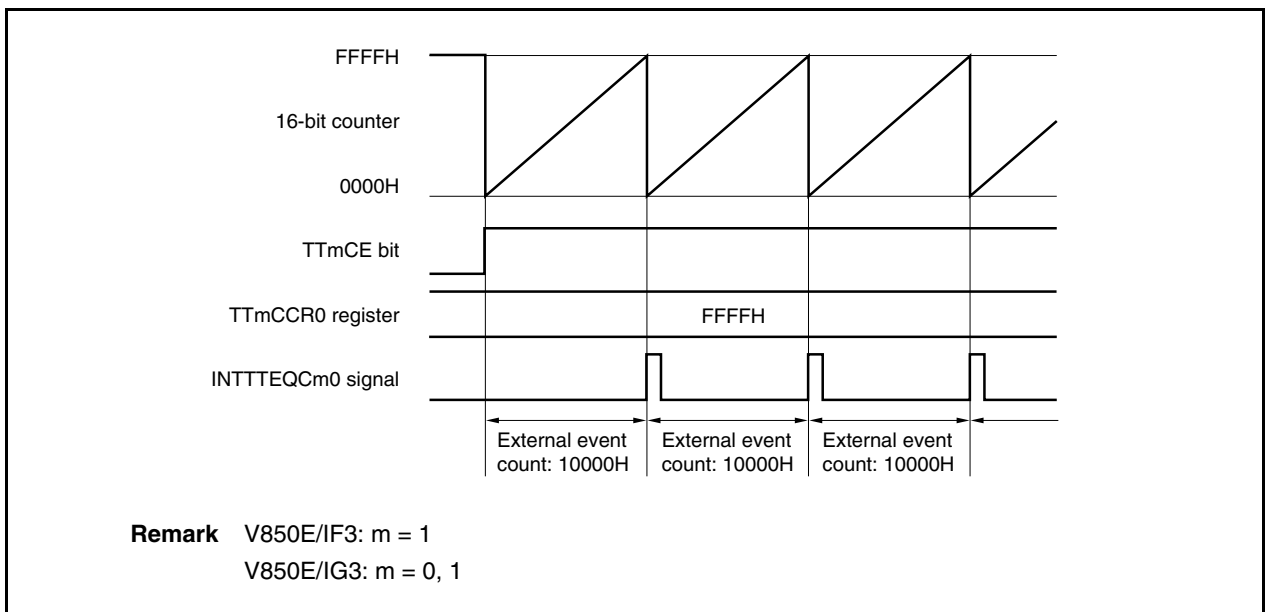
When the TTmCCR0 register is set to 0000H, the 16-bit counter is repeatedly cleared to 0000H and generates the INTTTEQm0 signal each time it has detected the valid edge of the external event count signal and its value has matched that of the CCR0 buffer register.

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



(b) Operation if TTmCCR0 register is set to FFFFH

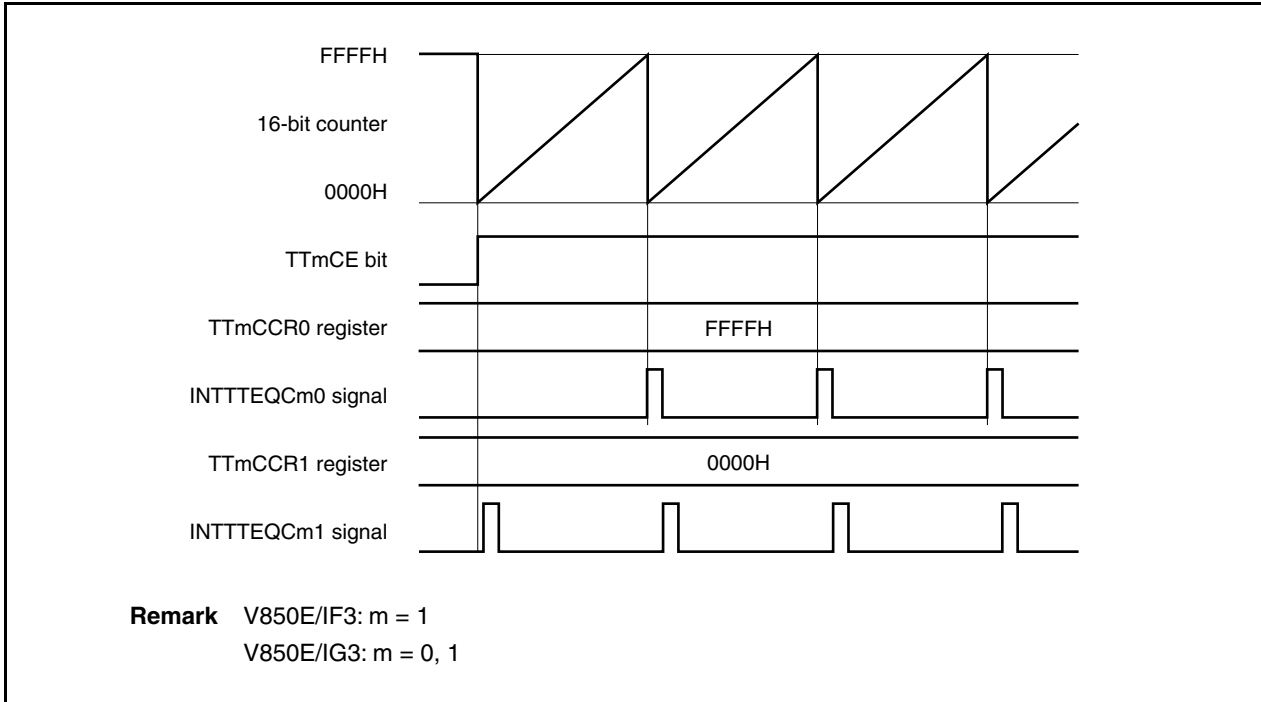
If the TTmCCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH each time the valid edge of the external event count signal has been detected. The 16-bit counter is cleared to 0000H in synchronization with the next count-up timing, and the INTTTEQm0 signal is generated. At this time, the TTmOPT0.TTmOVF bit is not set.



**(c) Operation with TTmCCR0 set to FFFFH and TTmCCR1 register to 0000H**

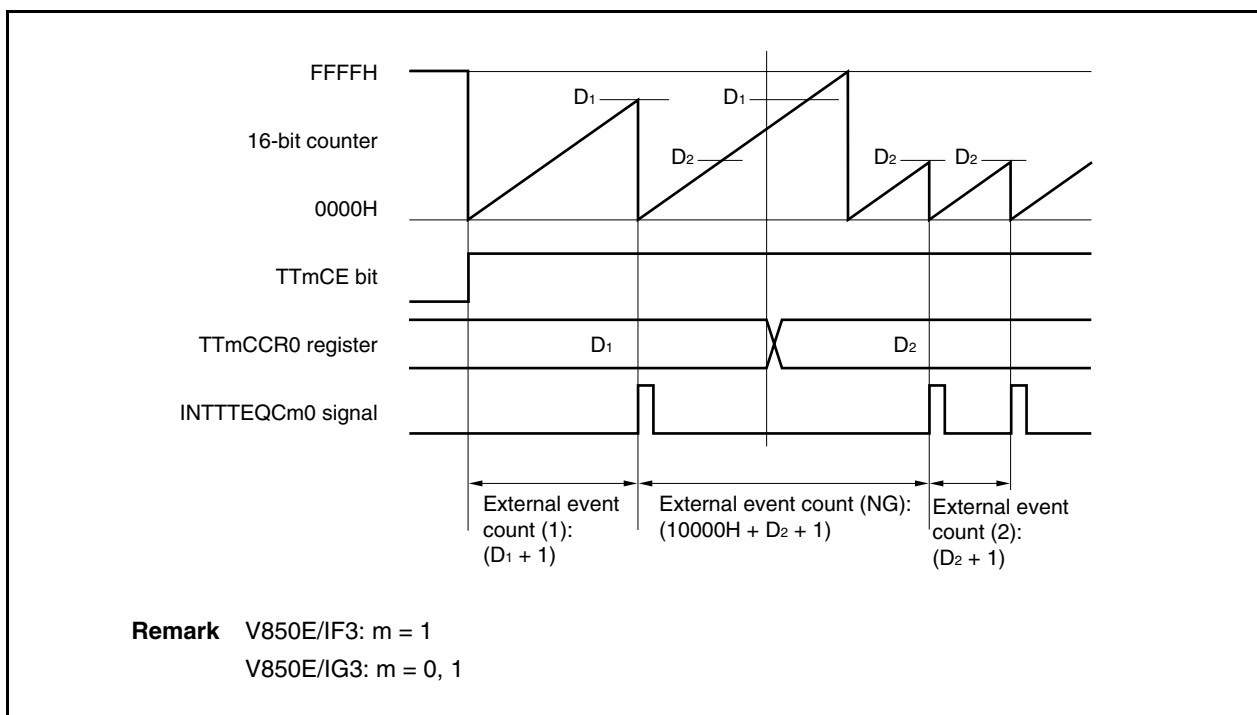
When the TTmCCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH each time it has detected the valid edge of the external event count signal. The counter is then cleared to 0000H in synchronization with the next count-up timing and the INTTTEQm0 signal is generated. At this time, the TTmOPT0.TTmOVF bit is not set.

If the TTmCCR1 register is set to 0000H, the INTTTEQm1 signal is generated when the 16-bit counter is cleared to 0000H.



**(d) Notes on rewriting the TTmCCR0 register**

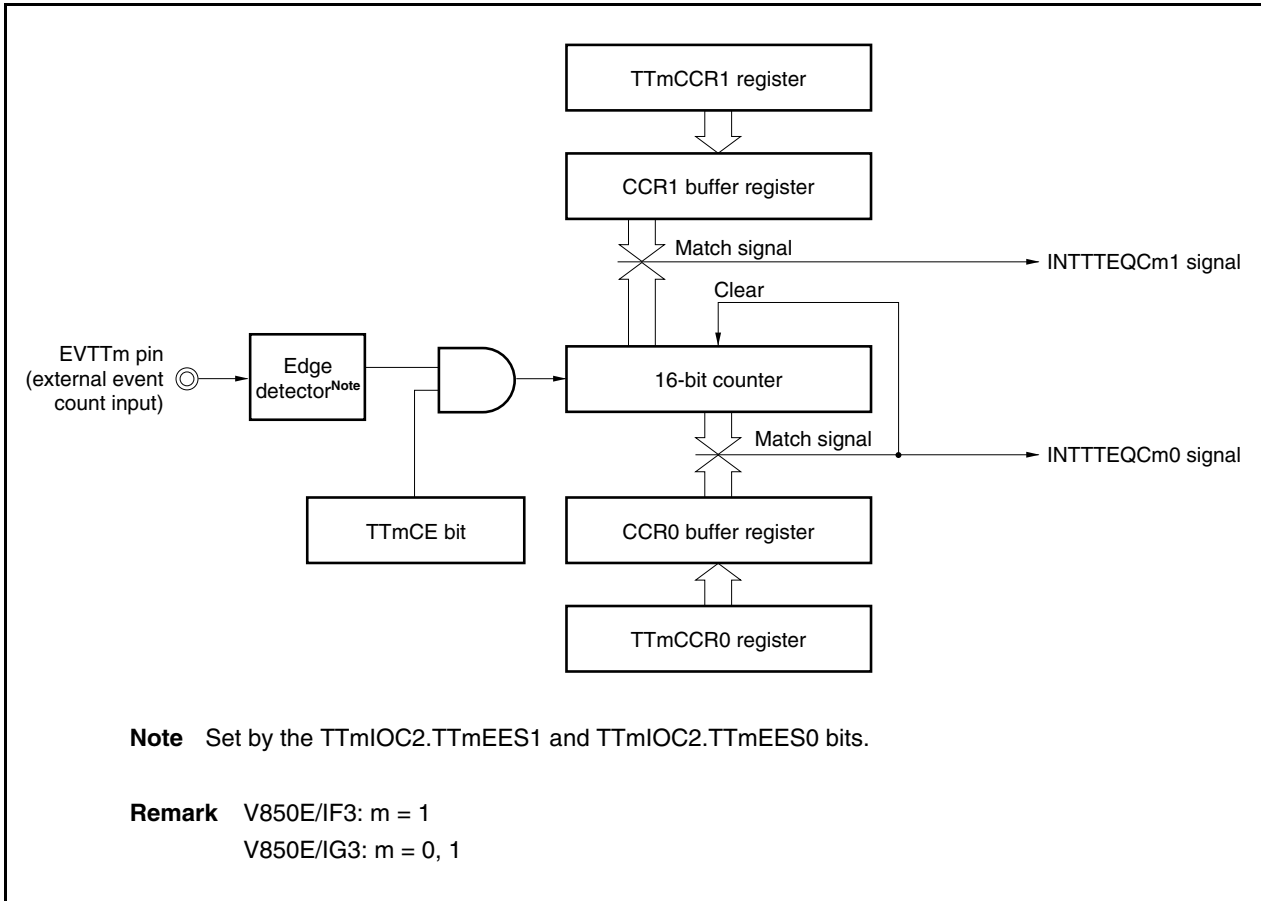
If the value of the TTmCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When the overflow may occur, stop counting once and then change the set value.



If the value of the TTmCCR0 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 TTmCCR0 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 INTTTEQCm0 signal is generated. Therefore, the INTTTEQCm0 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”.

(e) Operation of TTmCCR1 register

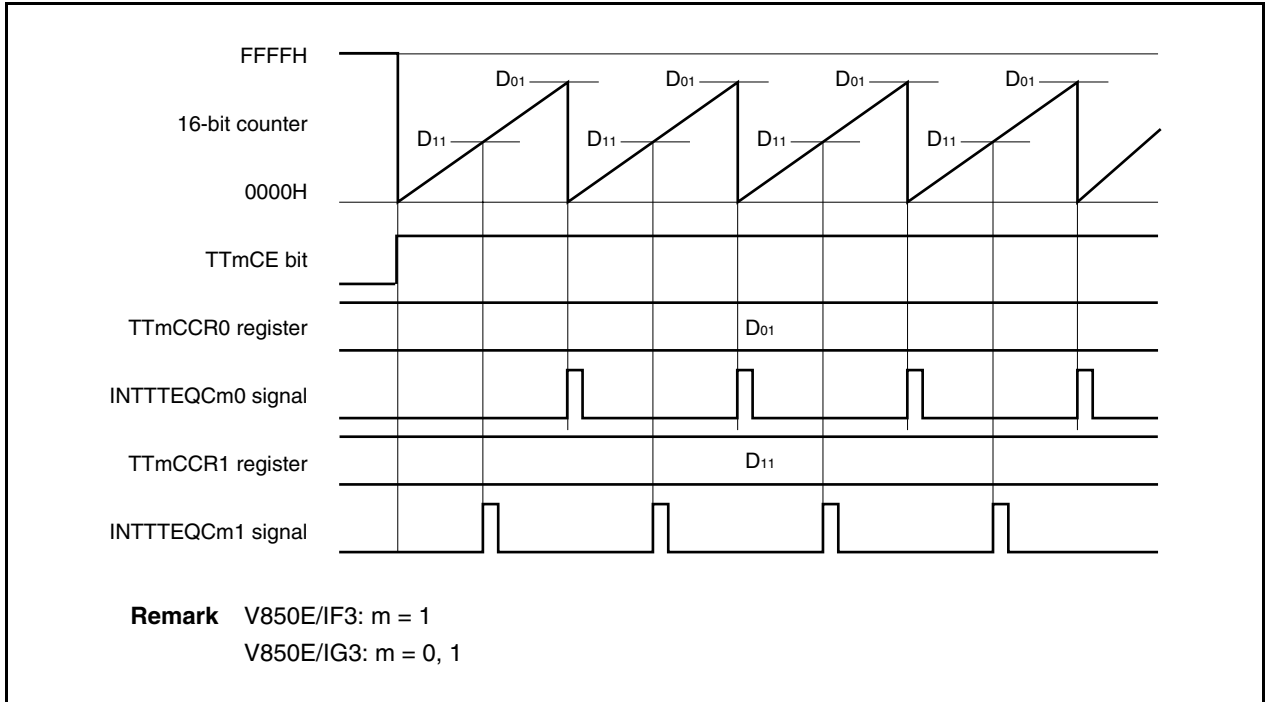
Figure 8-18. Configuration of TTmCCR1 Register





If the set value of the TTmCCR1 register is smaller than the set value of the TTmCCR0 register, the INTTTEQm1 signal is generated once per cycle.

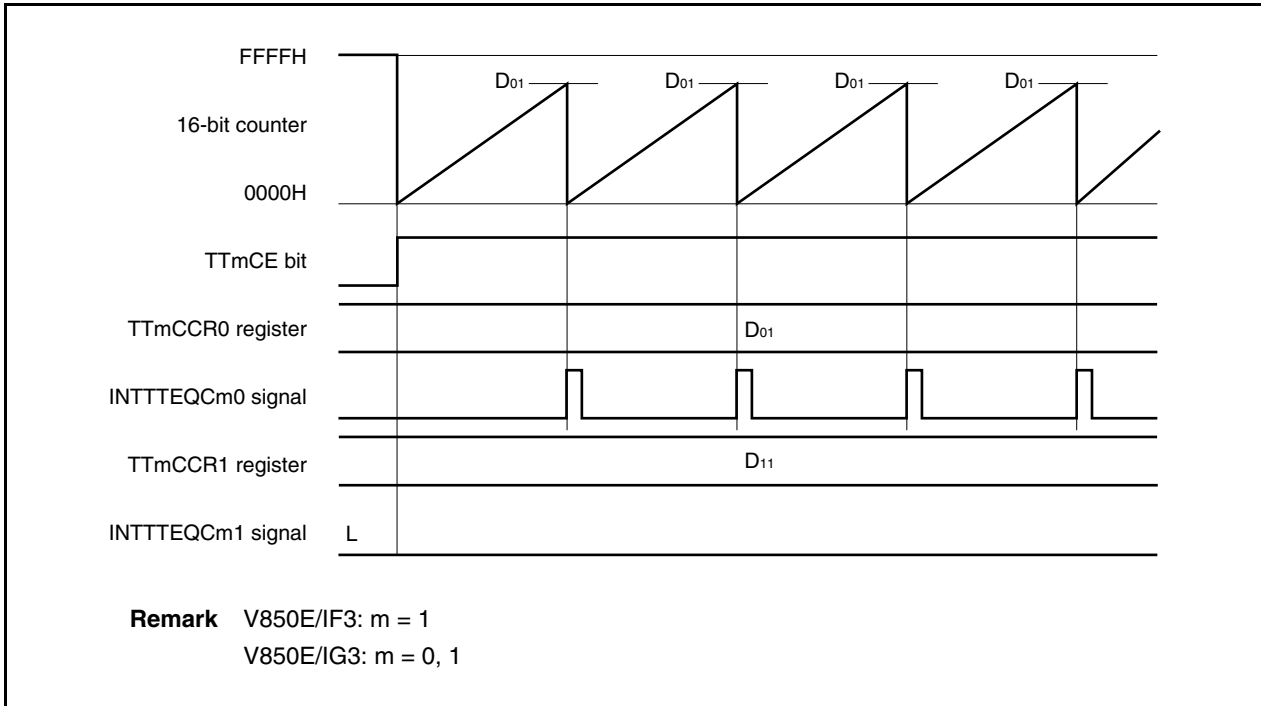
Figure 8-19. Timing Chart When  $D_{01} \geq D_{11}$



If the set value of the TTmCCR1 register is greater than the set value of the TTmCCR0 register, the INTTTEQCm1 signal is not generated because the count value of the 16-bit counter and the value of the TTmCCR1 register do not match.

When the TTmCCR1 register is not used, it is recommended to set its value to FFFFH.

Figure 8-20. Timing Chart When  $D_{01} < D_{11}$



### 8.6.3 External trigger pulse output mode (TTmMD3 to TTmMD0 bits = 0010)

This mode is valid only in TMT0 (V850E/IG3 only) and TMT1.

In the external trigger pulse output mode, 16-bit timer/event counter T waits for a trigger when the TTmCTL0.TTmCE bit is set to 1. When the valid edge of an external trigger input (EVTTm) is detected, 16-bit timer/event counter T starts counting, and outputs a PWM waveform from the TOTm1 pin. A PWM waveform with a duty factor of 50% that has the set value of the TTmCCR0 register + 1 as half its cycle can also be output from the TOTm0 pin.

Pulses can also be output by generating a software trigger instead of using the external trigger.

Figure 8-21. Configuration in External Trigger Pulse Output Mode

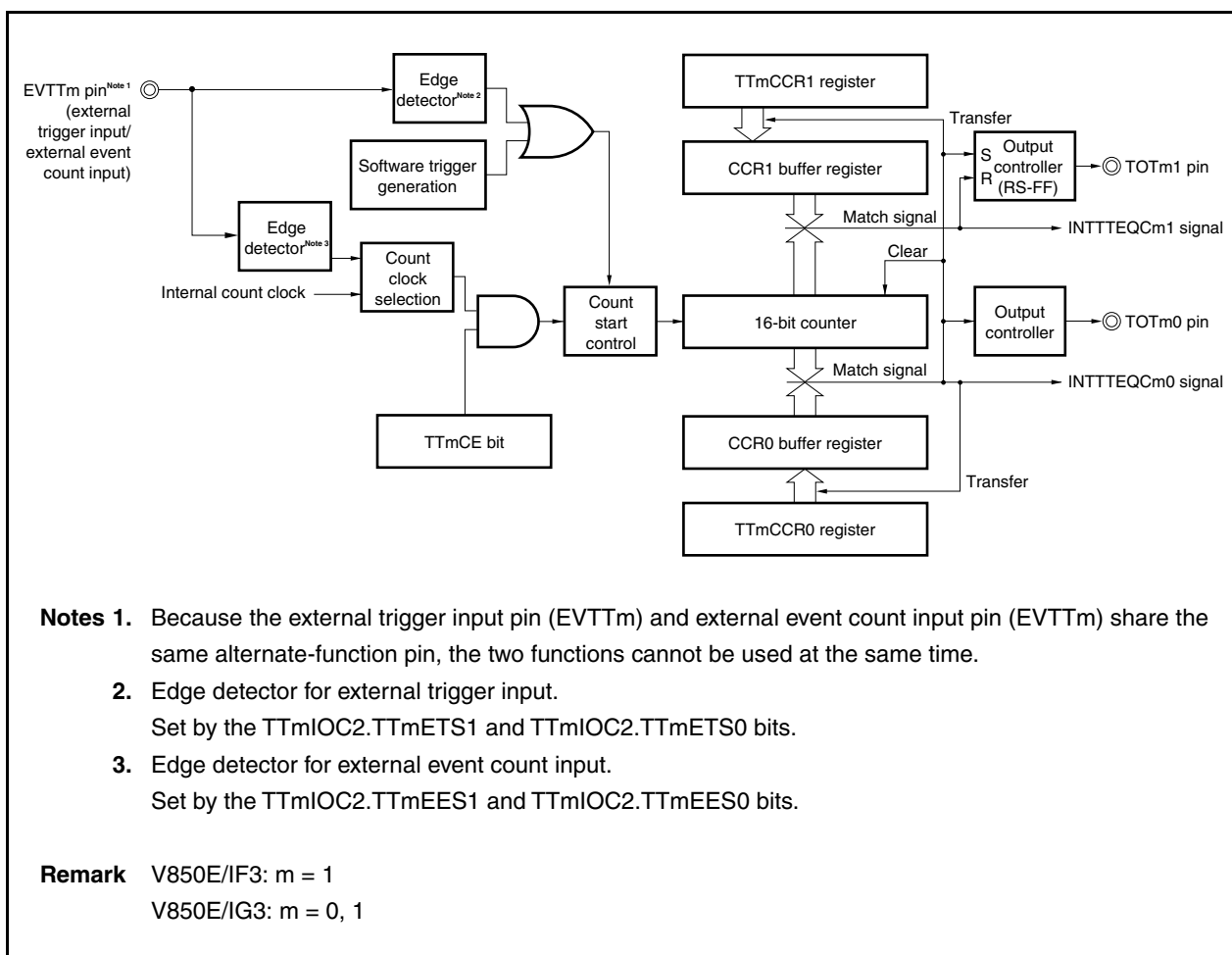
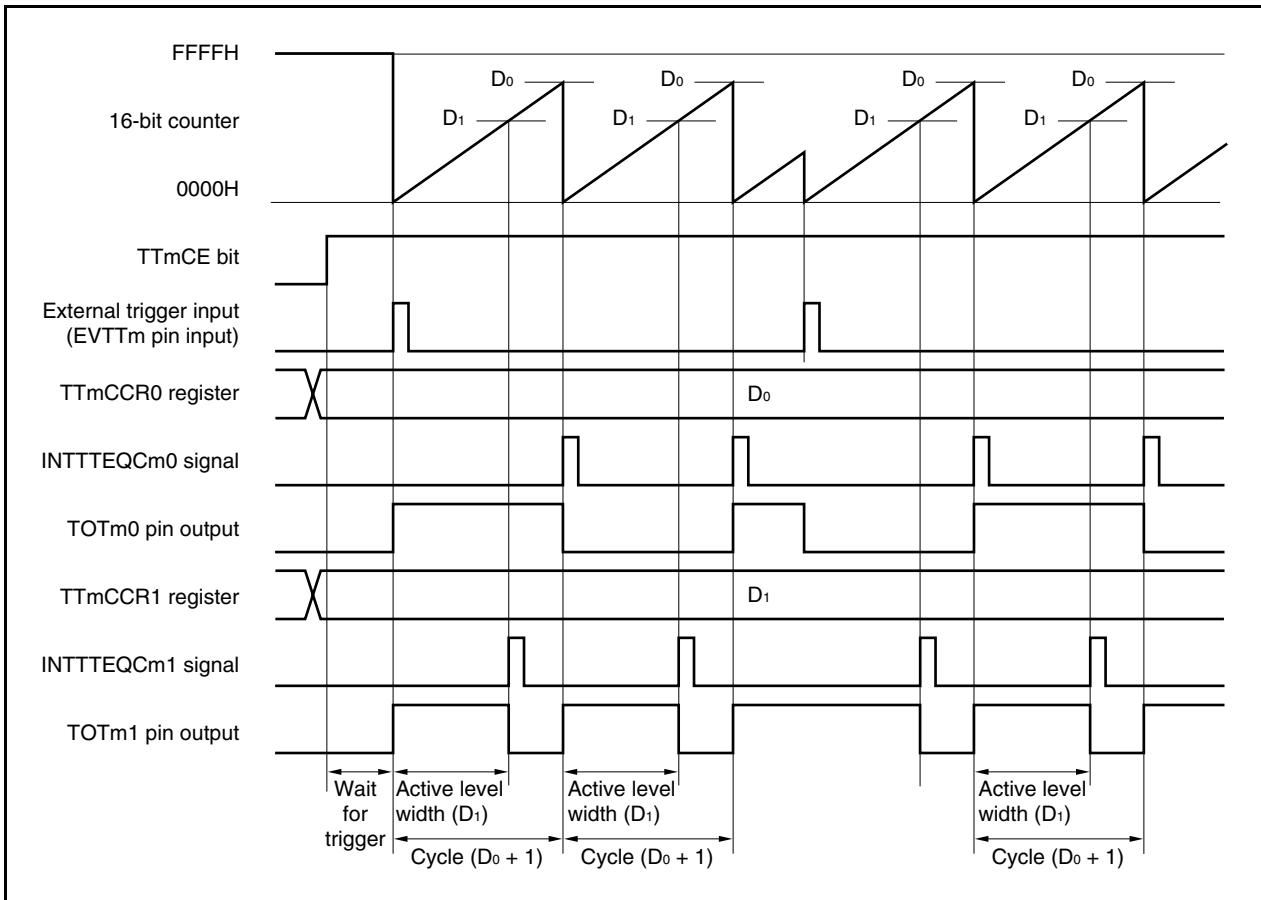


Figure 8-22. Basic Timing in External Trigger Pulse Output Mode



16-bit timer/event counter T waits for a trigger when the TmCE 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 TOTm1 pin. If the trigger is generated again while the counter is operating, the counter is cleared to 0000H and restarted. (The output of the TOTm0 pin is inverted. The TOTm1 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.

$$\text{Active level width} = (\text{Set value of TmCCR1 register}) \times \text{Count clock cycle}$$

$$\text{Cycle} = (\text{Set value of TmCCR0 register} + 1) \times \text{Count clock cycle}$$

$$\text{Duty factor} = (\text{Set value of TmCCR1 register}) / (\text{Set value of TmCCR0 register} + 1)$$

The compare match request signal (INTTTEQcm0) 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 (INTTTEQcm1) is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

The value set to the TmCCR<sub>a</sub> register is transferred to the CCR<sub>a</sub> buffer register when the count value of the 16-bit counter matches the value of the CCR<sub>a</sub> buffer register and the 16-bit counter is cleared to 0000H.

The valid edge of an external trigger input (EVTTm), or setting the software trigger (TmCTL1.TmEST bit) to 1 is used as the trigger.

**Remark** V850E/IF3: m = 1, a = 0, 1  
V850E/IG3: m = 0, 1, a = 0, 1

Figure 8-23. Setting of Registers in External Trigger Pulse Output Mode (1/2)

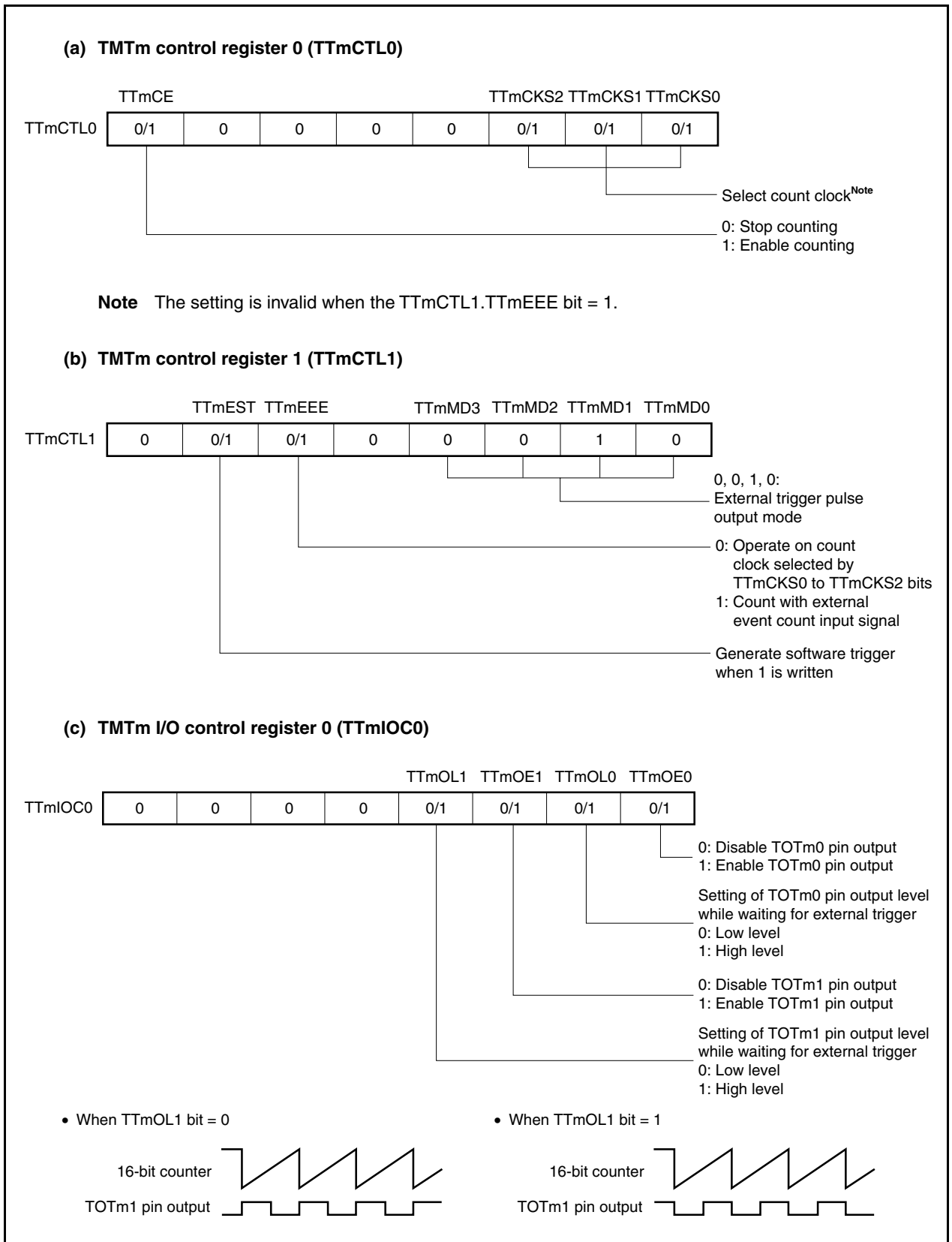
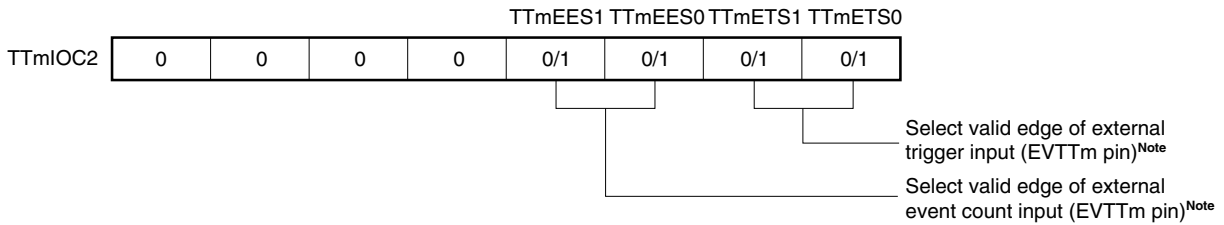


Figure 8-23. Setting of Registers in External Trigger Pulse Output Mode (2/2)

(d) TMTm I/O control register 2 (TTmIOC2)



**Note** Set the valid edge selection of the unused alternate external input signals to “No edge detection”.

(e) TMTm counter read buffer register (TTmCNT)

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

(f) TMTm capture/compare registers 0 and 1 (TTmCCR0 and TTmCCR1)

If D<sub>0</sub> is set to the TTmCCR0 register and D<sub>1</sub> to the TTmCCR1 register, the cycle and active level of the PWM waveform are as follows.

$$\text{Cycle} = (D_0 + 1) \times \text{Count clock cycle}$$

$$\text{Active level width} = D_1 \times \text{Count clock cycle}$$

**Remarks** 1. TMTm control register 2 (TTmCTL2), TMTm I/O control register 1 (TTmIOC1), TMTm I/O control register 3 (TTmIOC3), TMTm option register 0 (TTmOPT0), TMTm option register 1 (TTmOPT1), TMTm capture input select register (TTISLm), and TMTm counter write register (TTmTCW) are not used in the external trigger pulse output mode.

- 2. V850E/IF3: m = 1  
V850E/IG3: m = 0, 1

(1) Operation flow in external trigger pulse output mode

Figure 8-24. Software Processing Flow in External Trigger Pulse Output Mode (1/2)

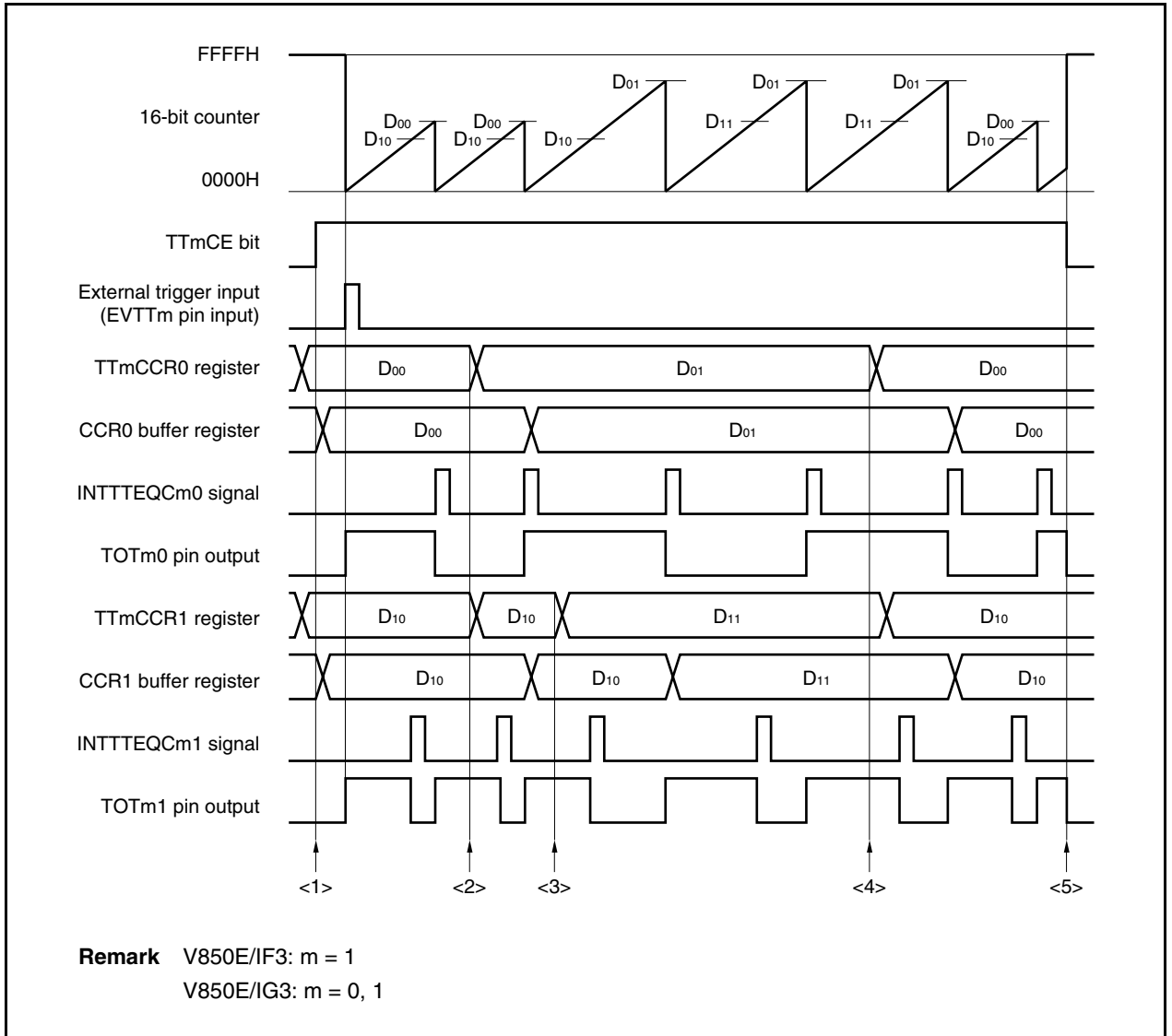
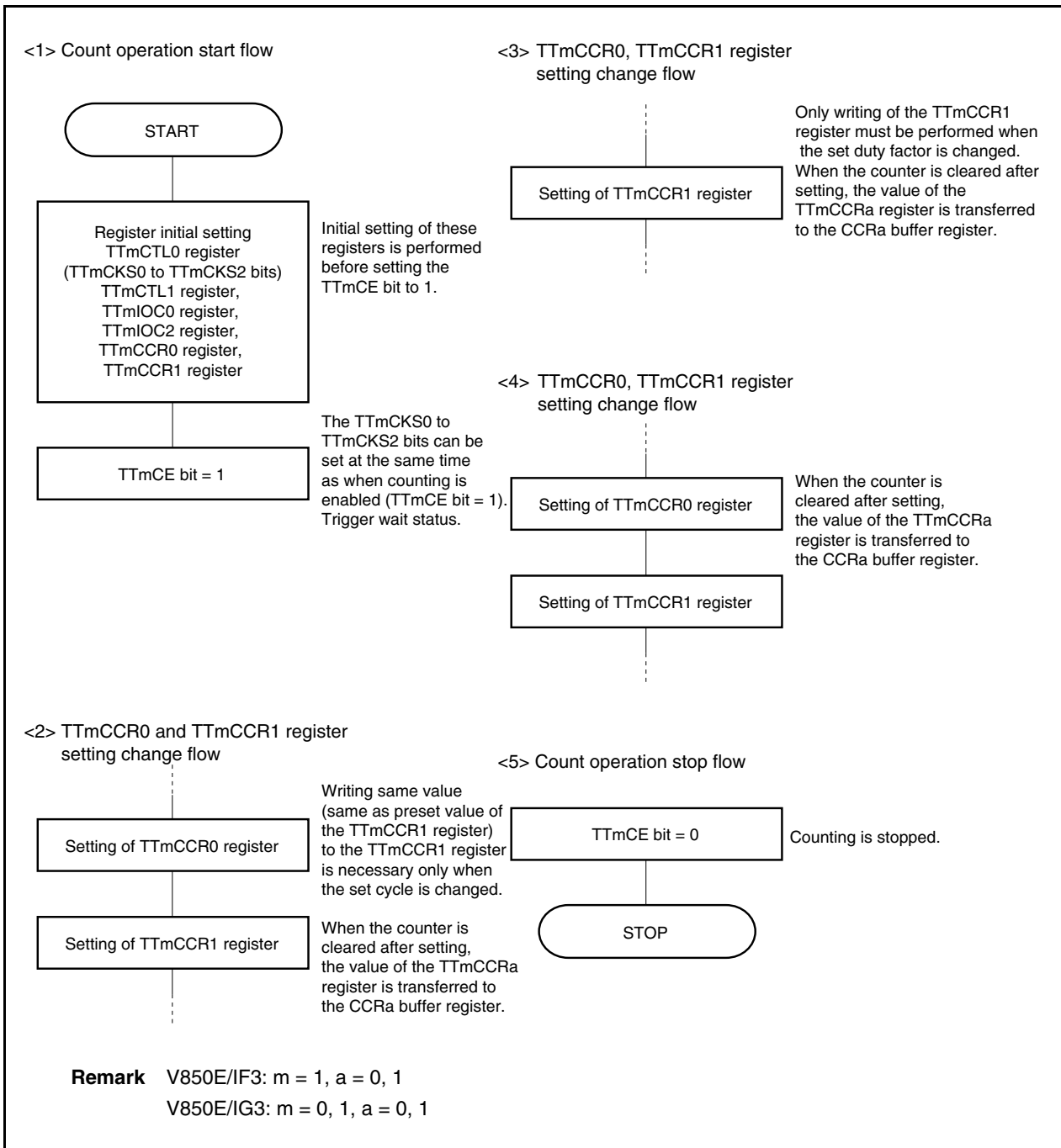


Figure 8-24. Software Processing Flow in External Trigger Pulse Output Mode (2/2)



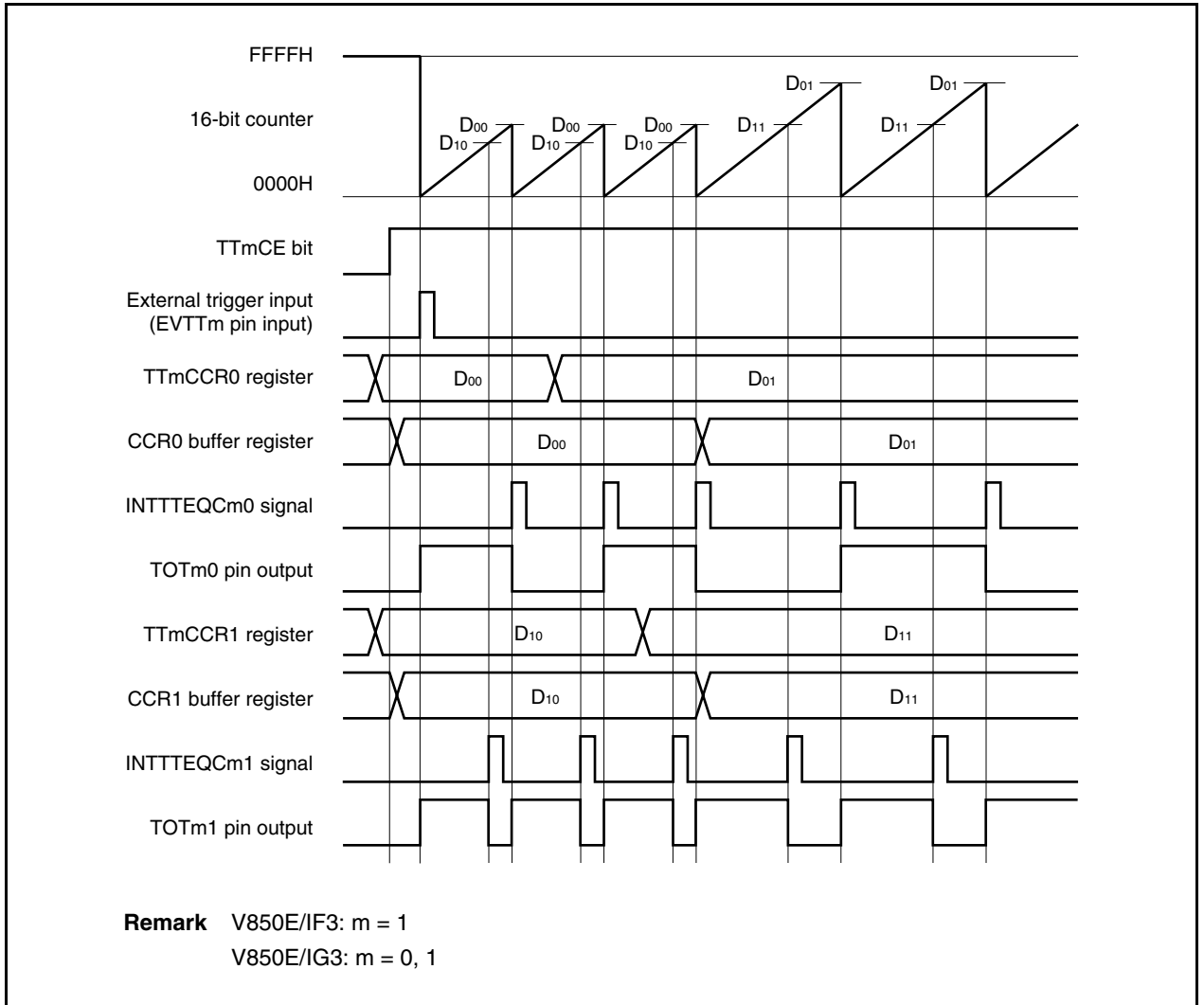


(2) External trigger pulse output mode operation timing

(a) Note on changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TTmCCR1 register last.

Rewrite the TTmCCRa register after writing the TTmCCR1 register after the INTTTEQCm0 signal is detected.



In order to transfer data from the TTmCCRa register to the CCRa buffer register, the TTmCCR1 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 TTmCCR0 register and then set the active level width to the TTmCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TTmCCR0 register, and then write the same value (same as preset value of the TTmCCR1 register) to the TTmCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, only the TTmCCR1 register has to be set.

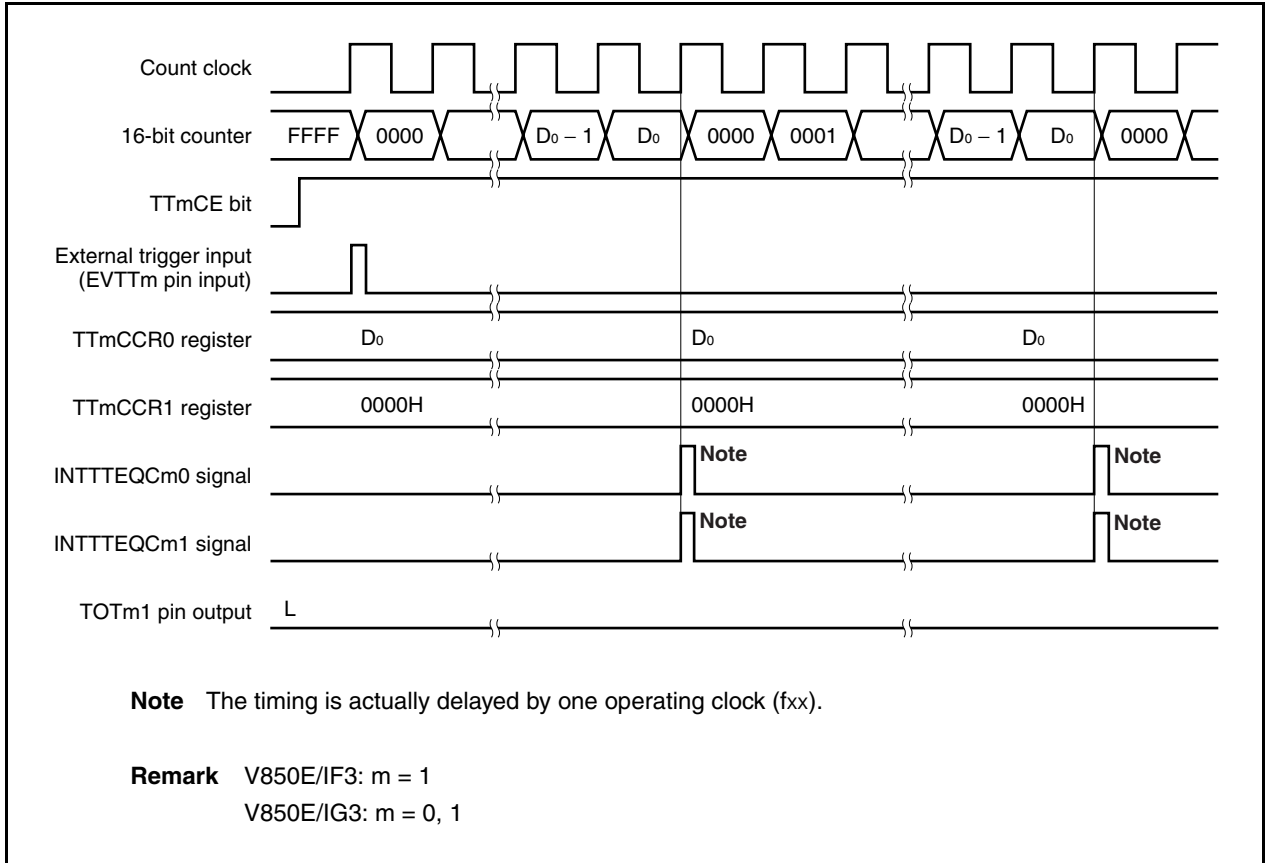
After data is written to the TTmCCR1 register, the value written to the TTmCCRa register is transferred to the CCRa buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TTmCCR0 or TTmCCR1 register again after writing the TTmCCR1 register once, do so after the INTTTEQCm0 signal is generated. Otherwise, the value of the CCRa buffer register may become undefined because the timing of transferring data from the TTmCCRa register to the CCRa buffer register conflicts with writing the TTmCCRa register.

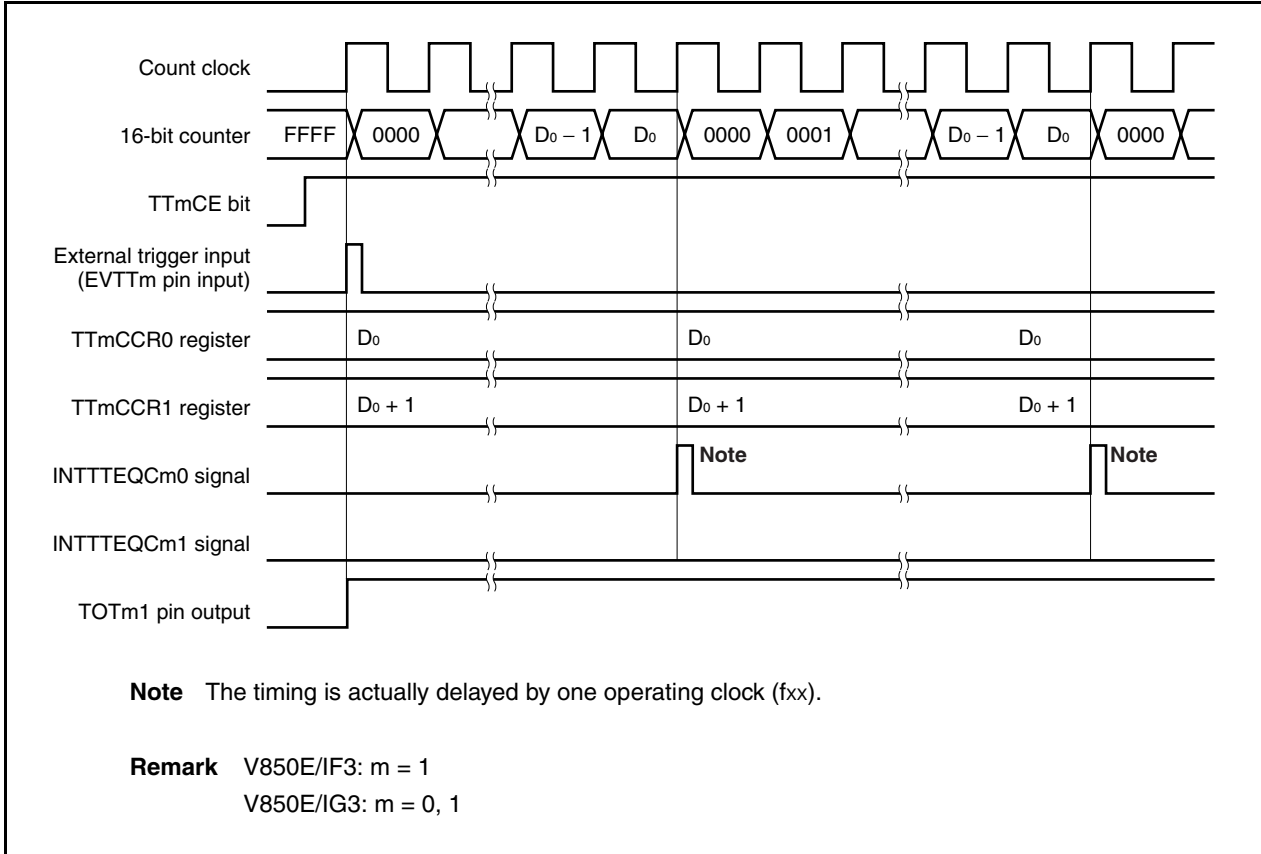
**Remark** V850E/IF3: m = 1, a = 0, 1  
V850E/IG3: m = 0, 1, a = 0, 1

**(b) 0%/100% output of PWM waveform**

To output a 0% waveform, set the TTmCCR1 register to 0000H. The 16-bit counter is cleared to 0000H and the INTTTEQm0 and INTTTEQm1 signals are generated at the next timing after a match between the count value of the 16-bit counter and the value of the CCR0 buffer register.

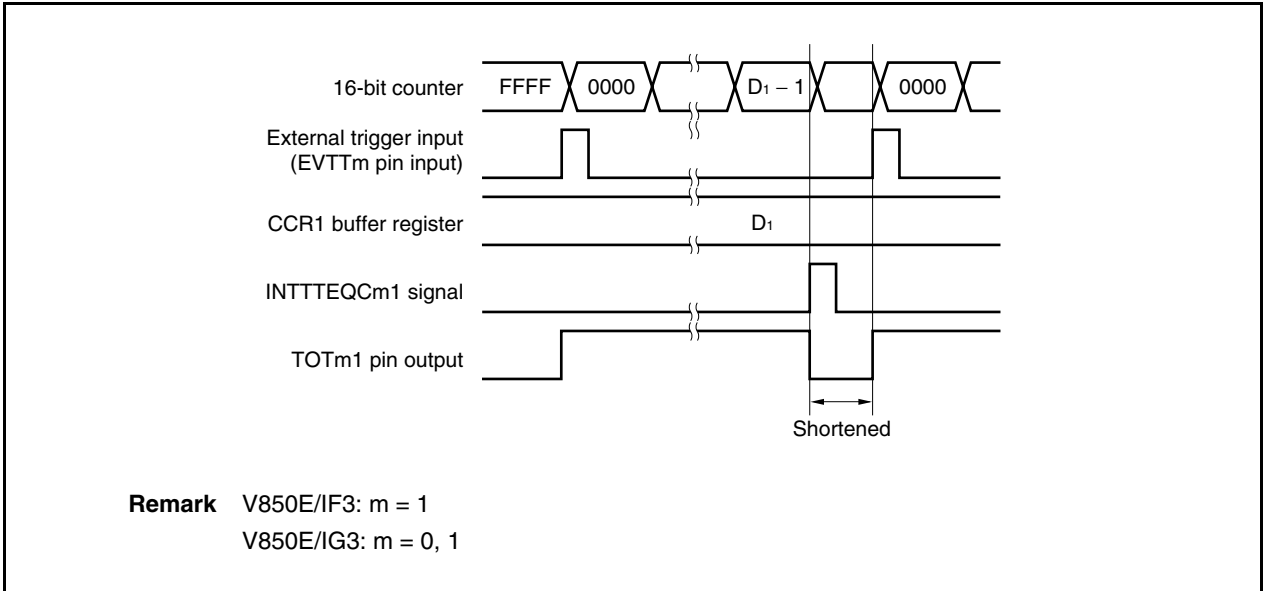


To output a 100% waveform, set a value of (set value of TTmCCR0 register + 1) to the TTmCCR1 register. If the set value of the TTmCCR0 register is FFFFH, 100% output cannot be produced.

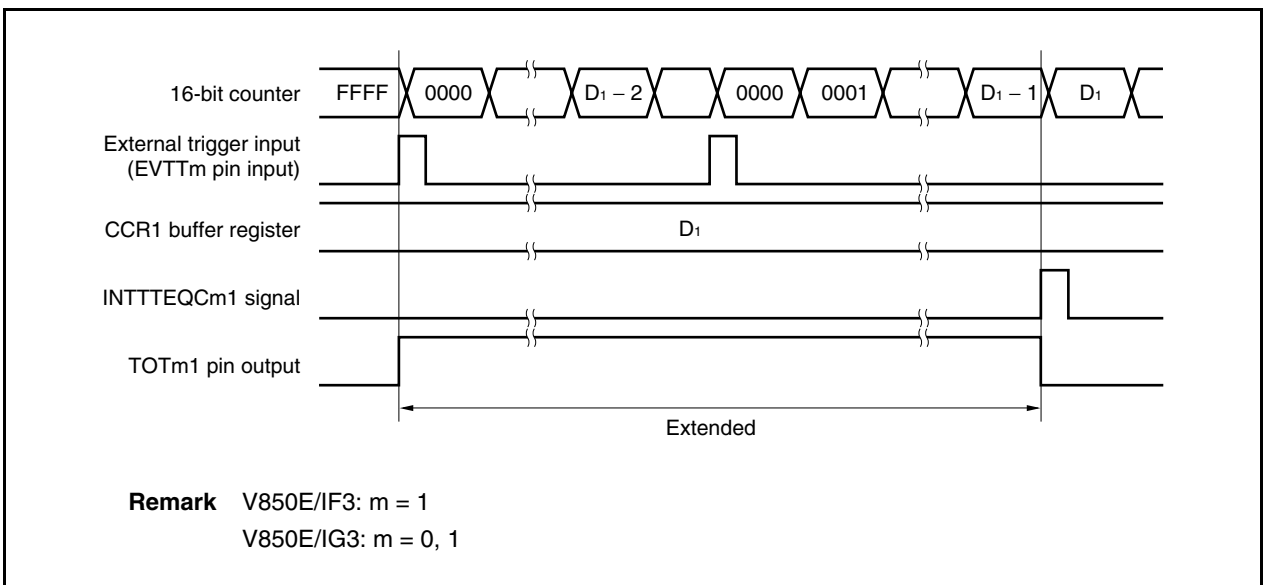


**(c) Conflict between trigger detection and match with CCR1 buffer register**

If the trigger is detected immediately after the INTTTEQCm1 signal is generated, the 16-bit counter is immediately cleared to 0000H, the output signal of the TOTm1 pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.

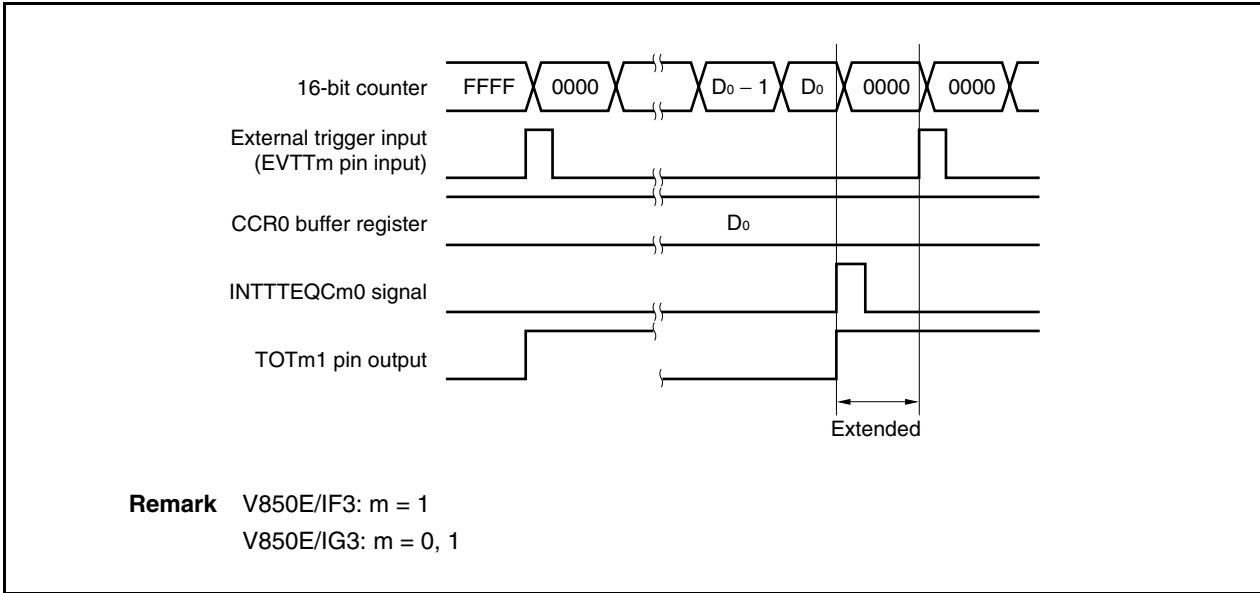


If the trigger is detected immediately before the INTTTEQCm1 signal is generated, the INTTTEQCm1 signal is not generated, and the 16-bit counter is cleared to 0000H and continues counting. The output signal of the TOTm1 pin remains active. Consequently, the active period of the PWM waveform is extended.

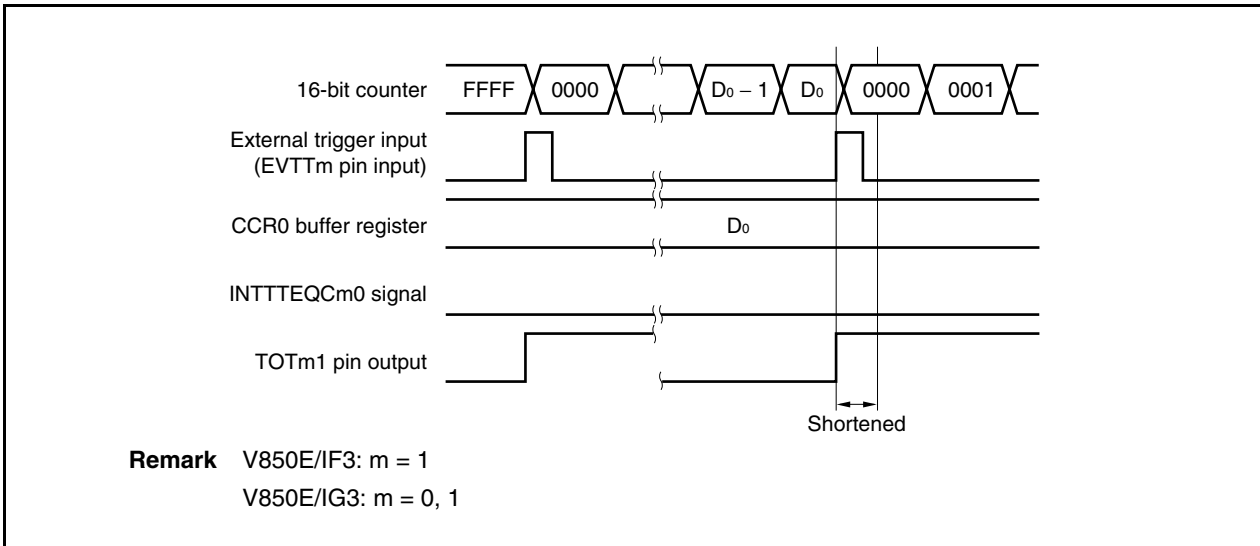


**(d) Conflict between trigger detection and match with CCR0 buffer register**

If the trigger is detected immediately after the INTTTEQCm0 signal is generated, the 16-bit counter is cleared to 0000H and continues counting up. Therefore, the active period of the TOTm1 pin is extended by time from generation of the INTTTEQCm0 signal to trigger detection.

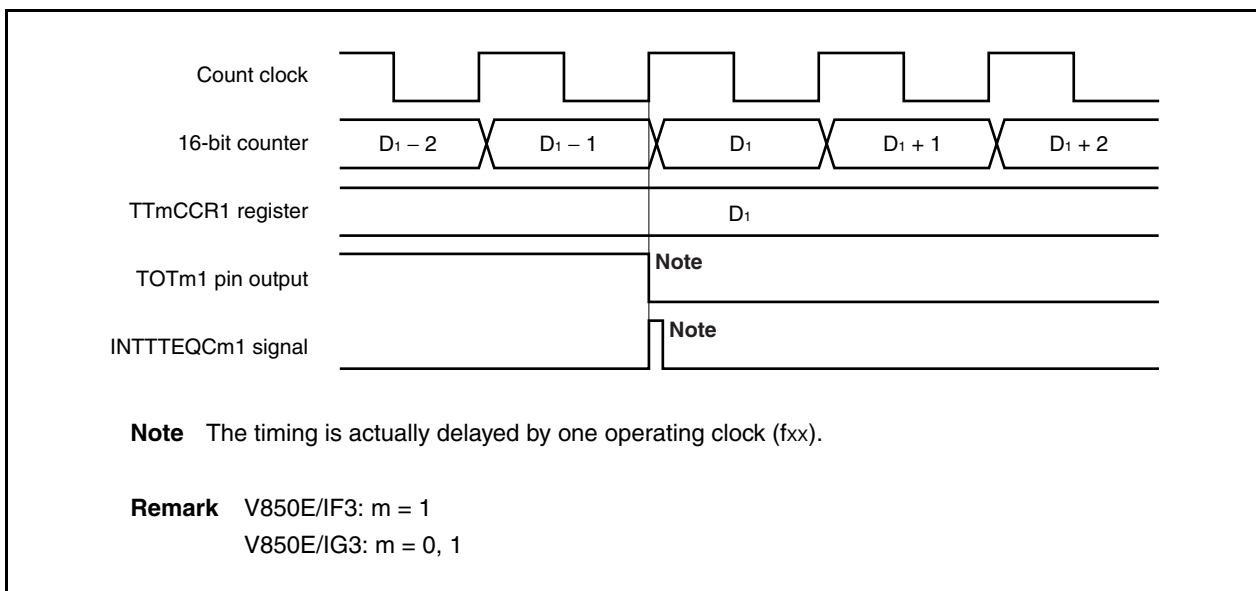


If the trigger is detected immediately before the INTTTEQCm0 signal is generated, the INTTTEQCm0 signal is not generated. The 16-bit counter is cleared to 0000H, the TOTm1 pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



**(e) Generation timing of compare match interrupt request signal (INTTTEQCm1)**

The timing of generation of the INTTTEQCm1 signal in the external trigger pulse output mode differs from the timing of INTTTEQCm1 signals in other mode; the INTTTEQCm1 signal is generated when the count value of the 16-bit counter matches the value of the TTmCCR1 register.



Usually, the INTTTEQCm1 signal is generated in synchronization with the next count-up, after the count value of the 16-bit counter matches the value of the TTmCCR1 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 TOTm1 pin.

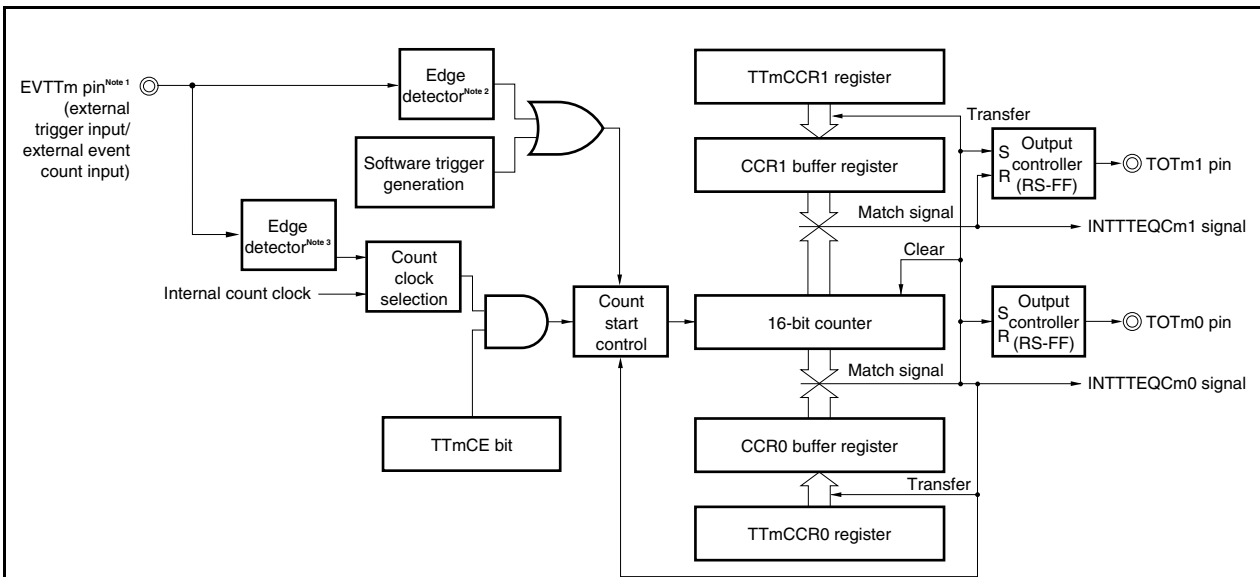
**8.6.4 One-shot pulse output mode (TTmMD3 to TTmMD0 bits = 0011)**

This mode is valid only in TMT0 (V850E/IG3 only) and TMT1.

In the one-shot pulse output mode, 16-bit timer/event counter T waits for a trigger when the TTmCTL0.TTmCE bit is set to 1. When the valid edge of an external trigger input (EVTTm) is detected, 16-bit timer/event counter T starts counting, and outputs a one-shot pulse from the TOTm1 pin. The TOTm0 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).

Instead of the external trigger input (EVTTm), a software trigger can also be generated to output the pulse.

**Figure 8-25. Configuration in One-Shot Pulse Output Mode**

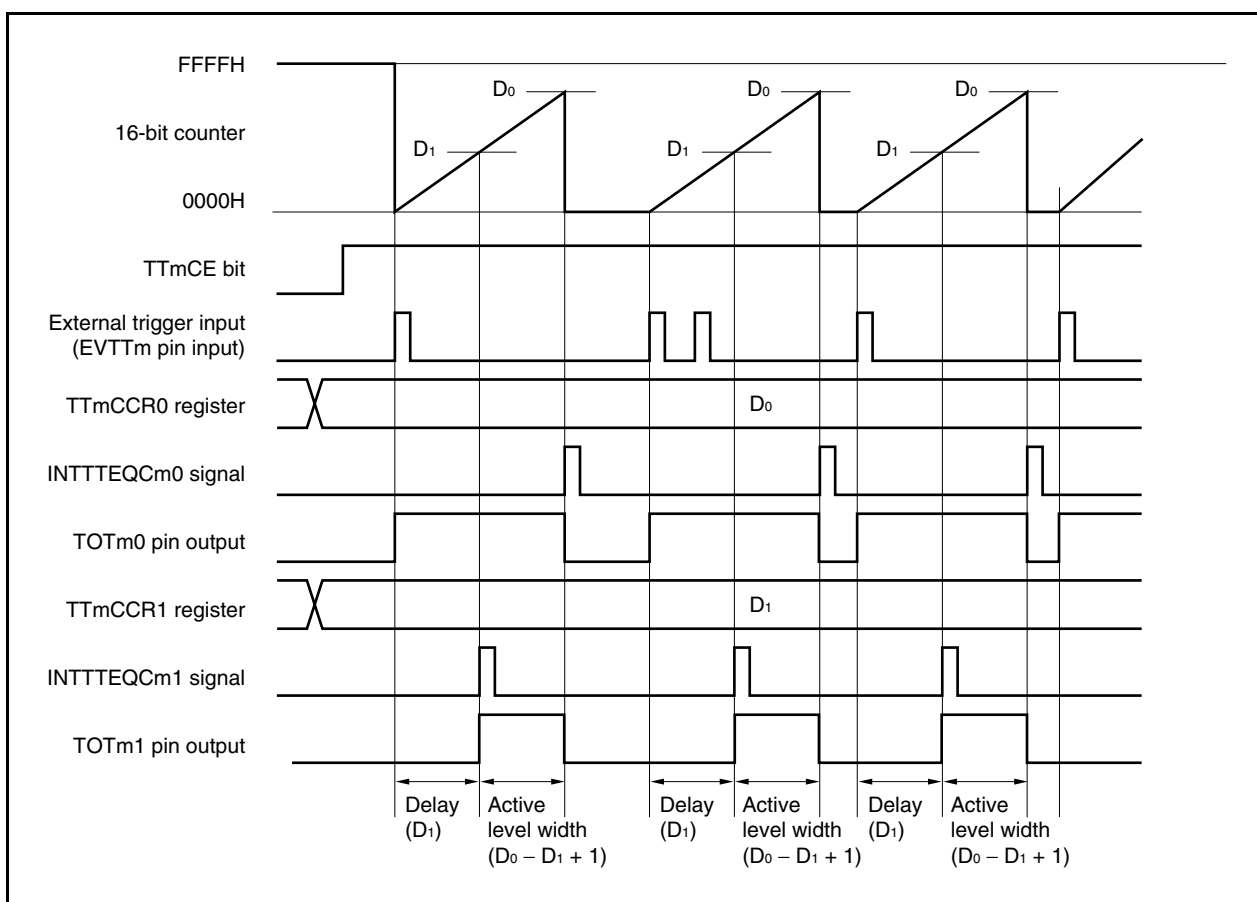


- Notes**
1. Because the external trigger input pin (EVTTm) and external event count input pin (EVTm) share the same alternate-function pin, the two functions cannot be used at the same time.
  2. Edge detector for external trigger input.  
Set by the TTmIIOC2.TTmETS1 and TTmIIOC2.TTmETS0 bits.
  3. Edge detector for external event count input.  
Set by the TTmIIOC2.TTmEES1 and TTmIIOC2.TTmEES0 bits.

**Remark** V850E/IF3: m = 1  
V850E/IG3: m = 0, 1



Figure 8-26. Basic Timing in One-Shot Pulse Output Mode



When the TTmCE bit is set to 1, 16-bit timer/event counter T 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 TOTm1 pin. After the one-shot pulse is output, the 16-bit counter is cleared to 0000H, stops counting, and waits for a trigger. When the trigger is generated again, the 16-bit counter starts counting from 0000H. 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.

$$\text{Output delay period} = (\text{Set value of TTmCCR1 register}) \times \text{Count clock cycle}$$

$$\text{Active level width} = (\text{Set value of TTmCCR0 register} - \text{Set value of TTmCCR1 register} + 1) \times \text{Count clock cycle}$$

The compare match interrupt request signal (INTTTEQm0) 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 (INTTTEQm1) 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 (EVTTm pin) or setting the software trigger (TTmCTL1.TTmEST bit) to 1 is used as the trigger.

**Remark** V850E/IF3: m = 1  
V850E/IG3: m = 0, 1

Figure 8-27. Setting of Registers in One-Shot Pulse Output Mode (1/2)

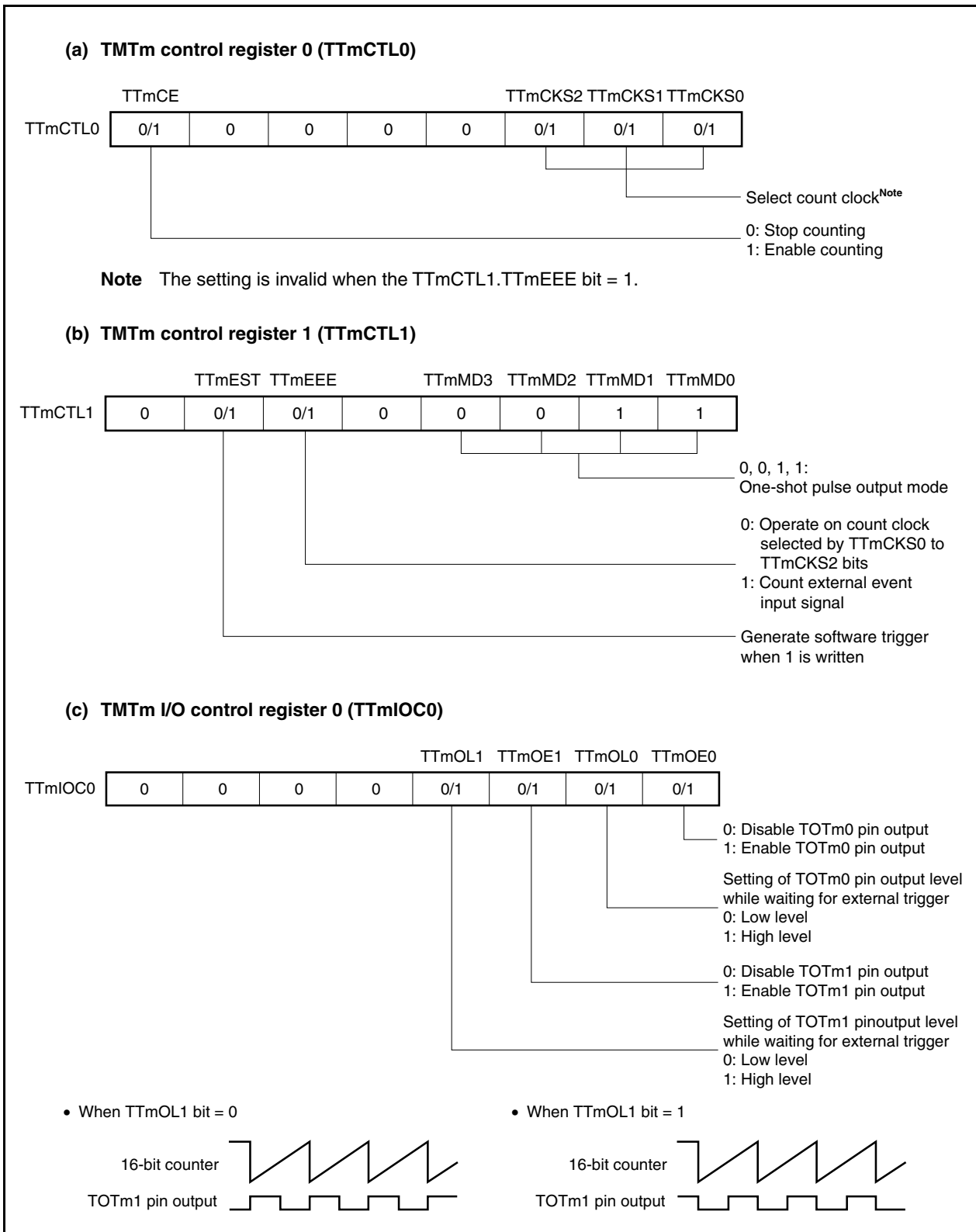
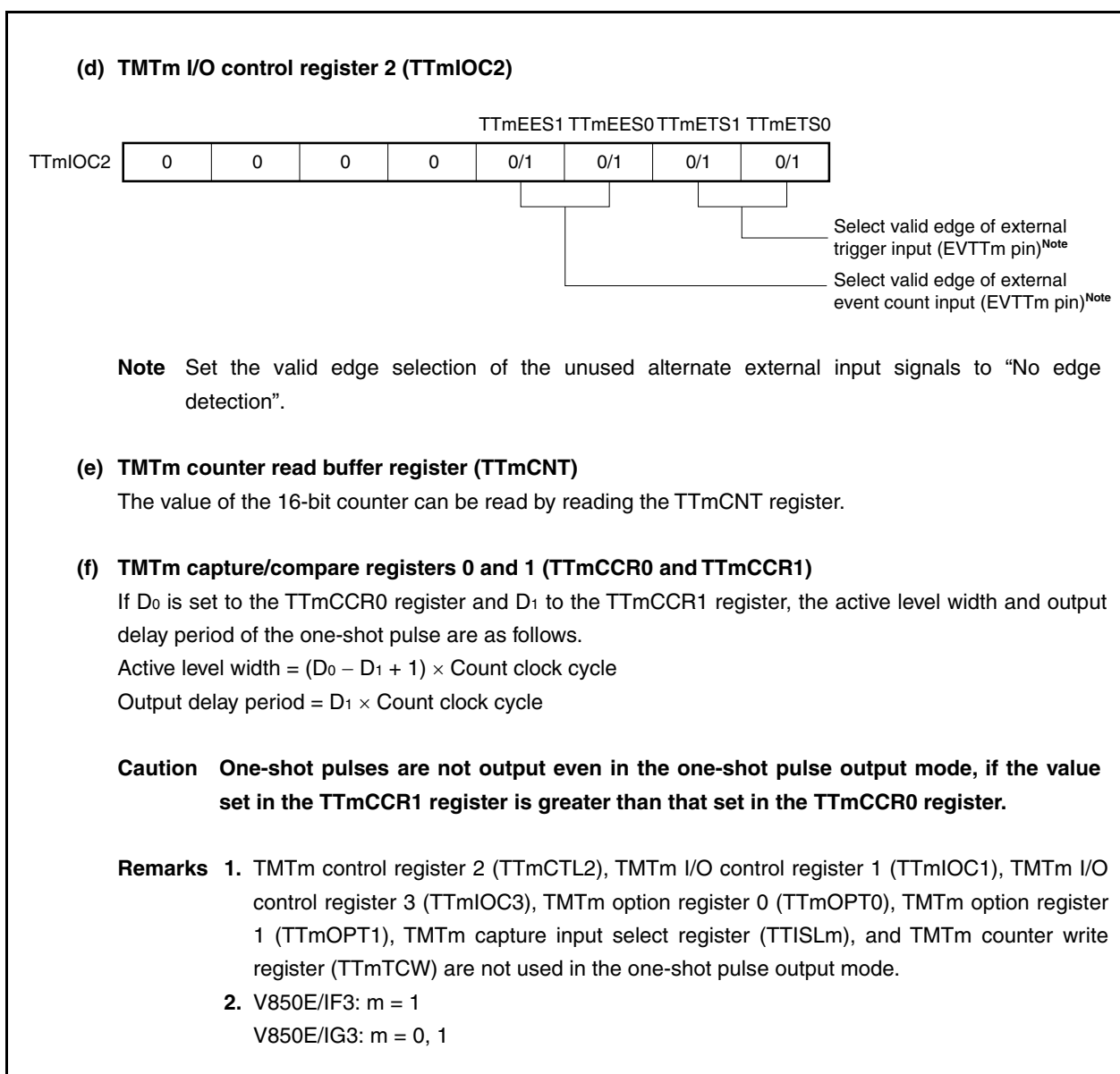
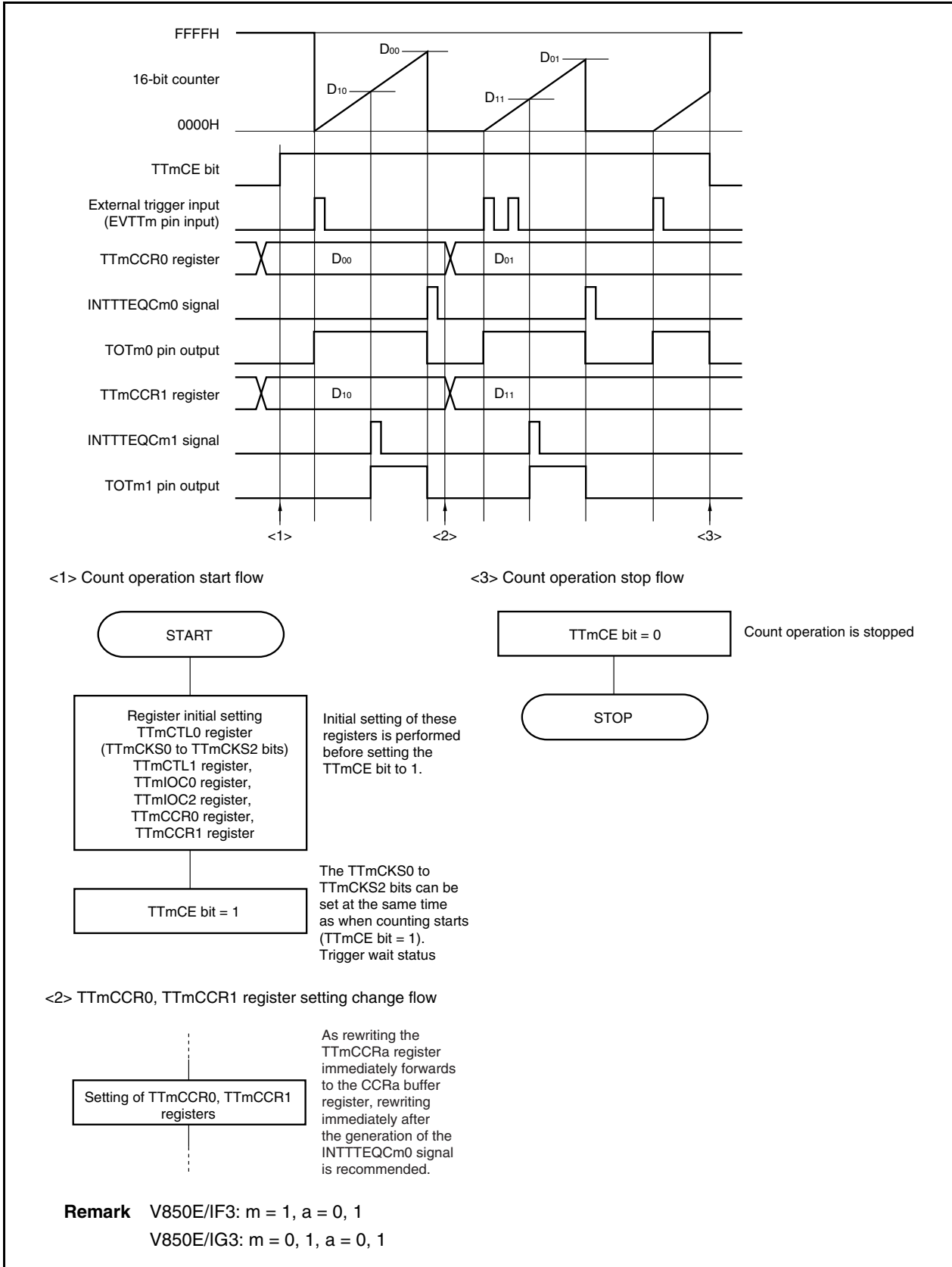


Figure 8-27. Setting of Registers in One-Shot Pulse Output Mode (2/2)



(1) Operation flow in one-shot pulse output mode

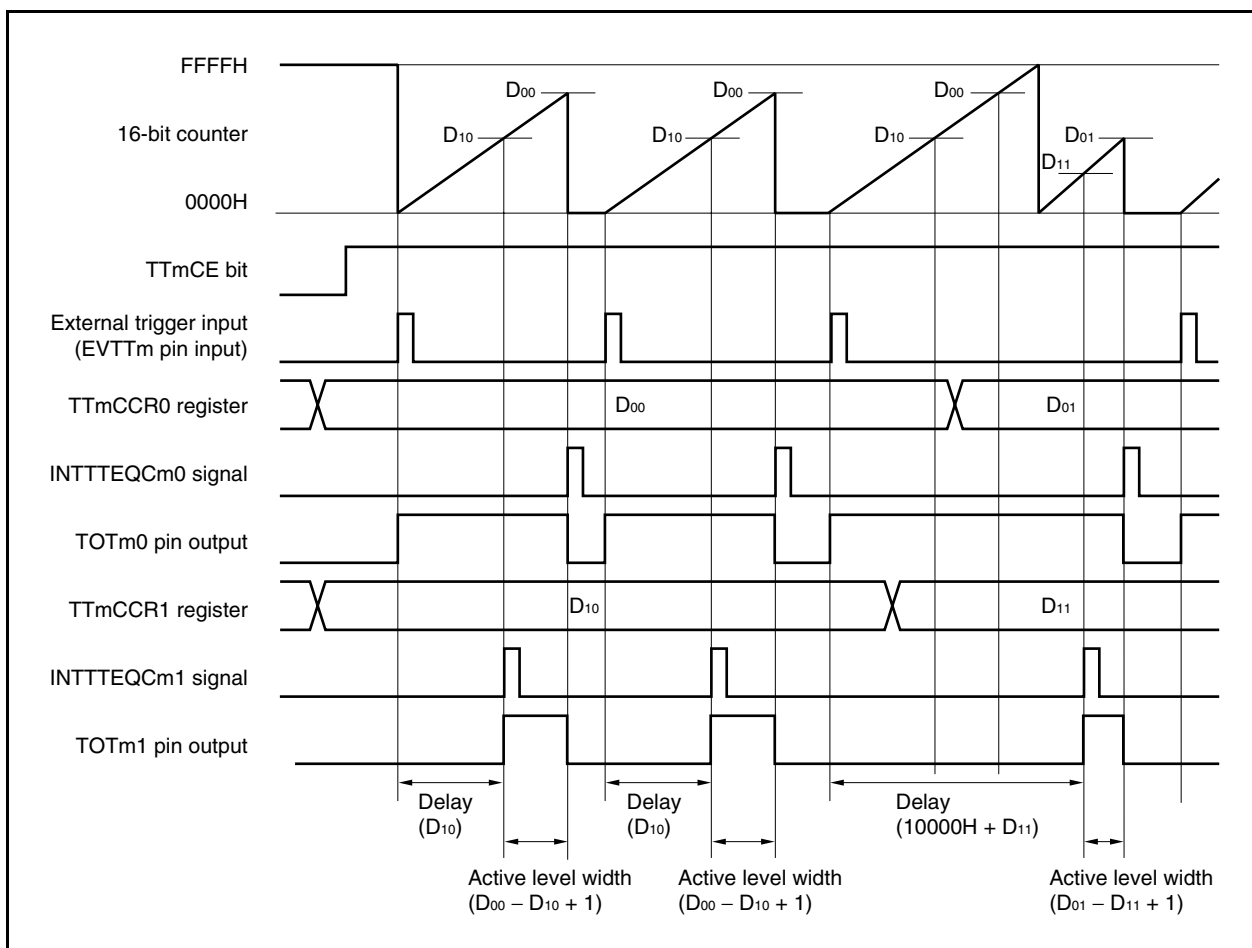
Figure 8-28. Software Processing Flow in One-Shot Pulse Output Mode



(2) Operation timing in one-shot pulse output mode

(a) Note on rewriting TTmCCRa register

If the value of the TTmCCRa register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When an overflow may occur, stop counting and then change the set value.



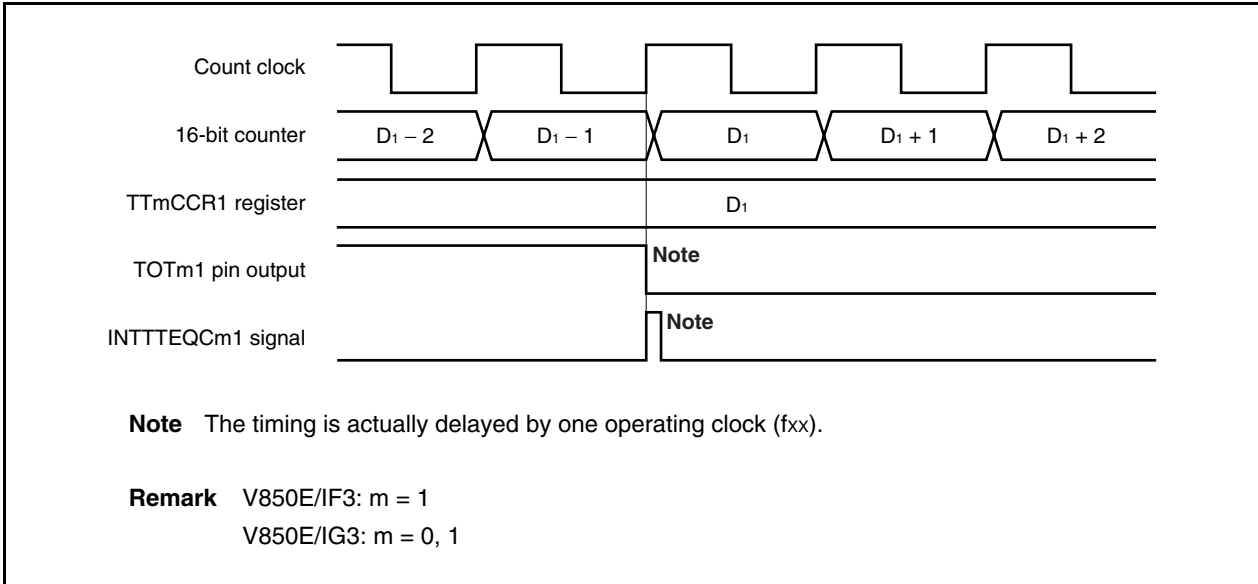
When the TTmCCR0 register is rewritten from D<sub>00</sub> to D<sub>01</sub> and the TTmCCR1 register from D<sub>10</sub> to D<sub>11</sub> where D<sub>00</sub> > D<sub>01</sub> and D<sub>10</sub> > D<sub>11</sub>, if the TTmCCR1 register is rewritten when the count value of the 16-bit counter is greater than D<sub>11</sub> and less than D<sub>10</sub> and if the TTmCCR0 register is rewritten when the count value is greater than D<sub>01</sub> and less than D<sub>00</sub>, 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<sub>11</sub>, the counter generates the INTTTEQCm1 signal and asserts the TOTm1 pin. When the count value matches D<sub>01</sub>, the counter generates the INTTTEQCm0 signal, deasserts the TOTm1 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** V850E/IF3: m = 1, a = 0, 1  
 V850E/IG3: m = 0, 1, a = 0, 1

**(b) Generation timing of compare match interrupt request signal (INTTTEQCm1)**

The generation timing of the INTTTEQCm1 signal in the one-shot pulse output mode is different from INTTTEQCm1 signals in other mode; the INTTTEQCm1 signal is generated when the count value of the 16-bit counter matches the value of the TTmCCR1 register.



Usually, the INTTTEQCm1 signal is generated when the 16-bit counter counts up next time after its count value matches the value of the TTmCCR1 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 TOTm1 pin.

**8.6.5 PWM output mode (TTmMD3 to TTmMD0 bits = 0100)**

This mode is valid only in TMT0 (V850E/IG3 only) and TMT1.

In the PWM output mode, a PWM waveform is output from the TOTm1 pin when the TTmCTL0.TTmCE bit is set to 1.

In addition, a PWM waveform with a duty factor of 50% with the set value of the TTmCCR0 register + 1 as half its cycle is output from the TOTm0 pin.

**Figure 8-29. Configuration in PWM Output Mode**

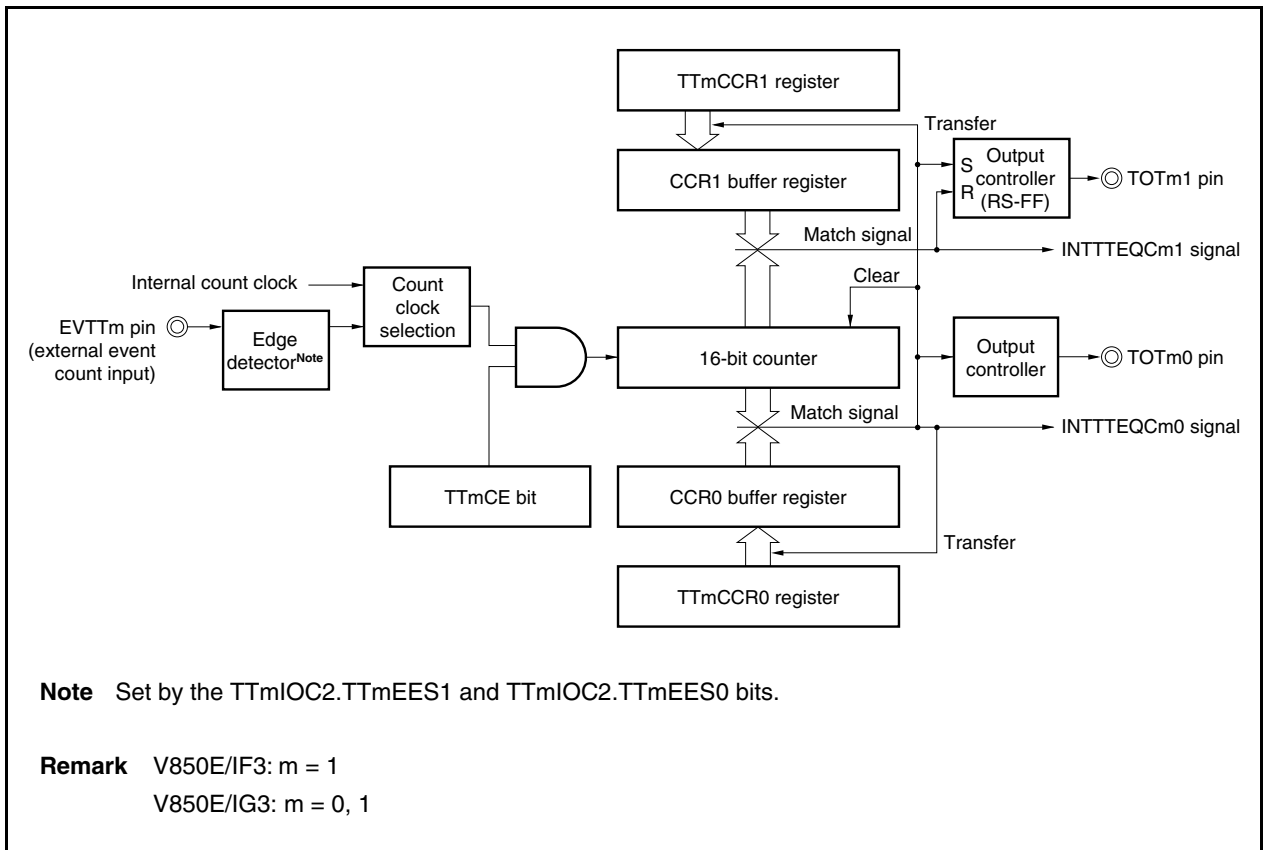
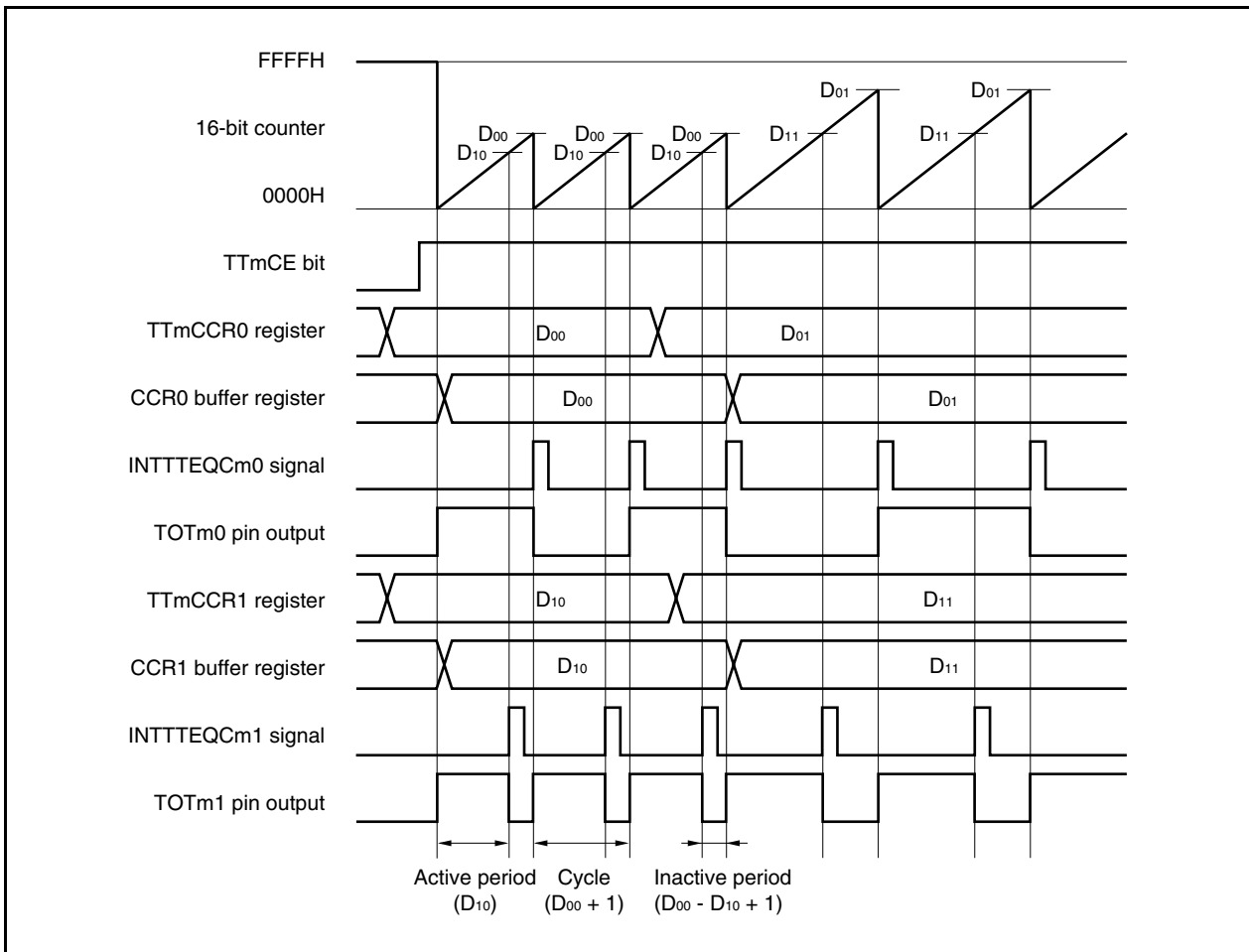


Figure 8-30. Basic Timing in PWM Output Mode



When the TTmCE bit is set to 1, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs a PWM waveform from the TOTm1 pin.

The active level width, cycle, and duty factor of the PWM waveform can be calculated as follows.

$$\text{Active level width} = (\text{Set value of TTmCCR1 register}) \times \text{Count clock cycle}$$

$$\text{Cycle} = (\text{Set value of TTmCCR0 register} + 1) \times \text{Count clock cycle}$$

$$\text{Duty factor} = (\text{Set value of TTmCCR1 register}) / (\text{Set value of TTmCCR0 register} + 1)$$

The PWM waveform can be changed by rewriting the TTmCCRa 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 (INTTTEQm0) 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 (INTTTEQm1) is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

The value set to the TTmCCRa register is transferred to the CCRa buffer register when the count value of the 16-bit counter matches the value of the CCRa buffer register and the 16-bit counter is cleared to 0000H.

**Remark** V850E/IF3: m = 1, a = 0, 1, V850E/IG3: m = 0, 1, a = 0, 1



Figure 8-31. Setting of Registers in PWM Output Mode (1/2)

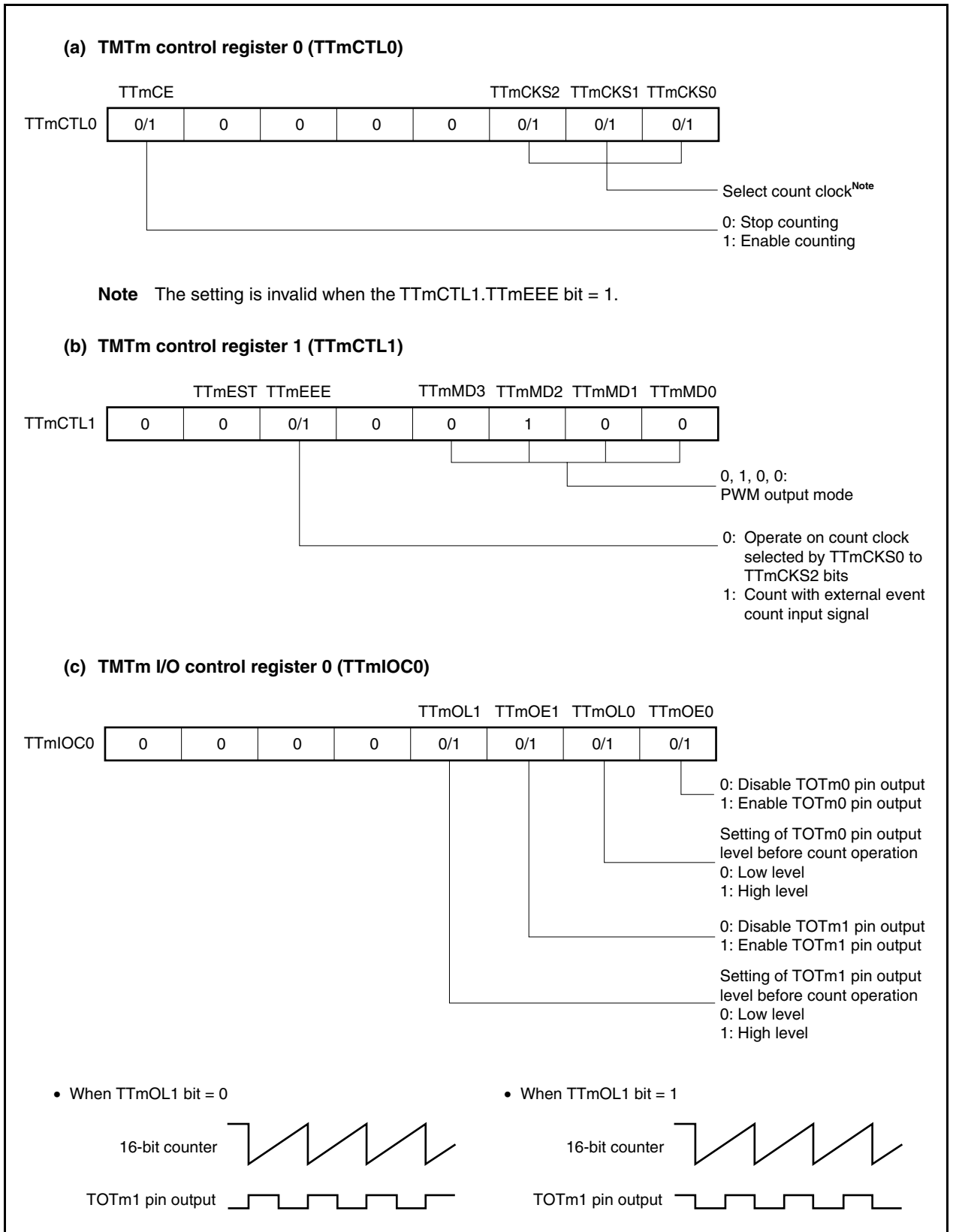
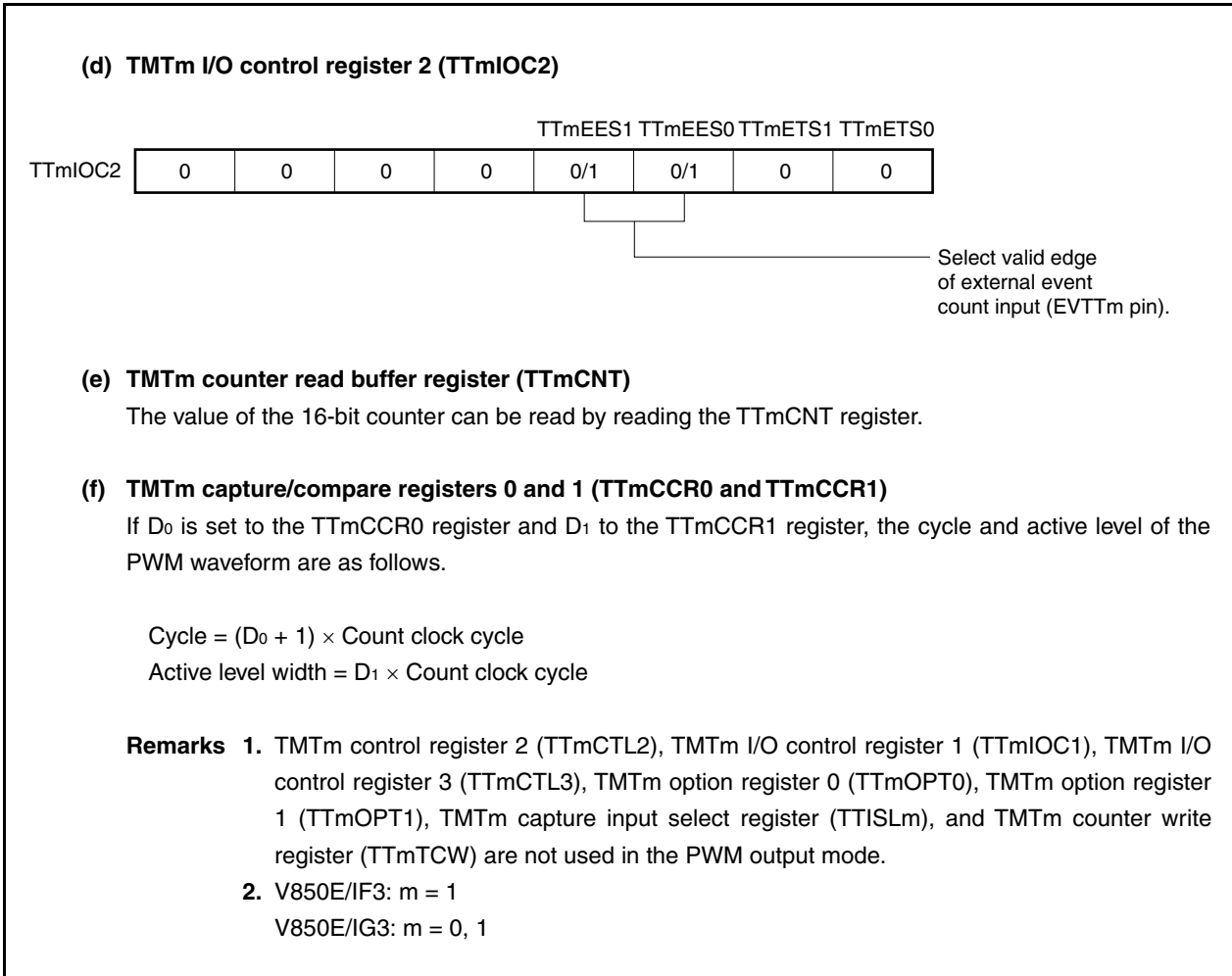


Figure 8-31. Register Setting in PWM Output Mode (2/2)



(1) Operation flow in PWM output mode

Figure 8-32. Software Processing Flow in PWM Output Mode (1/2)

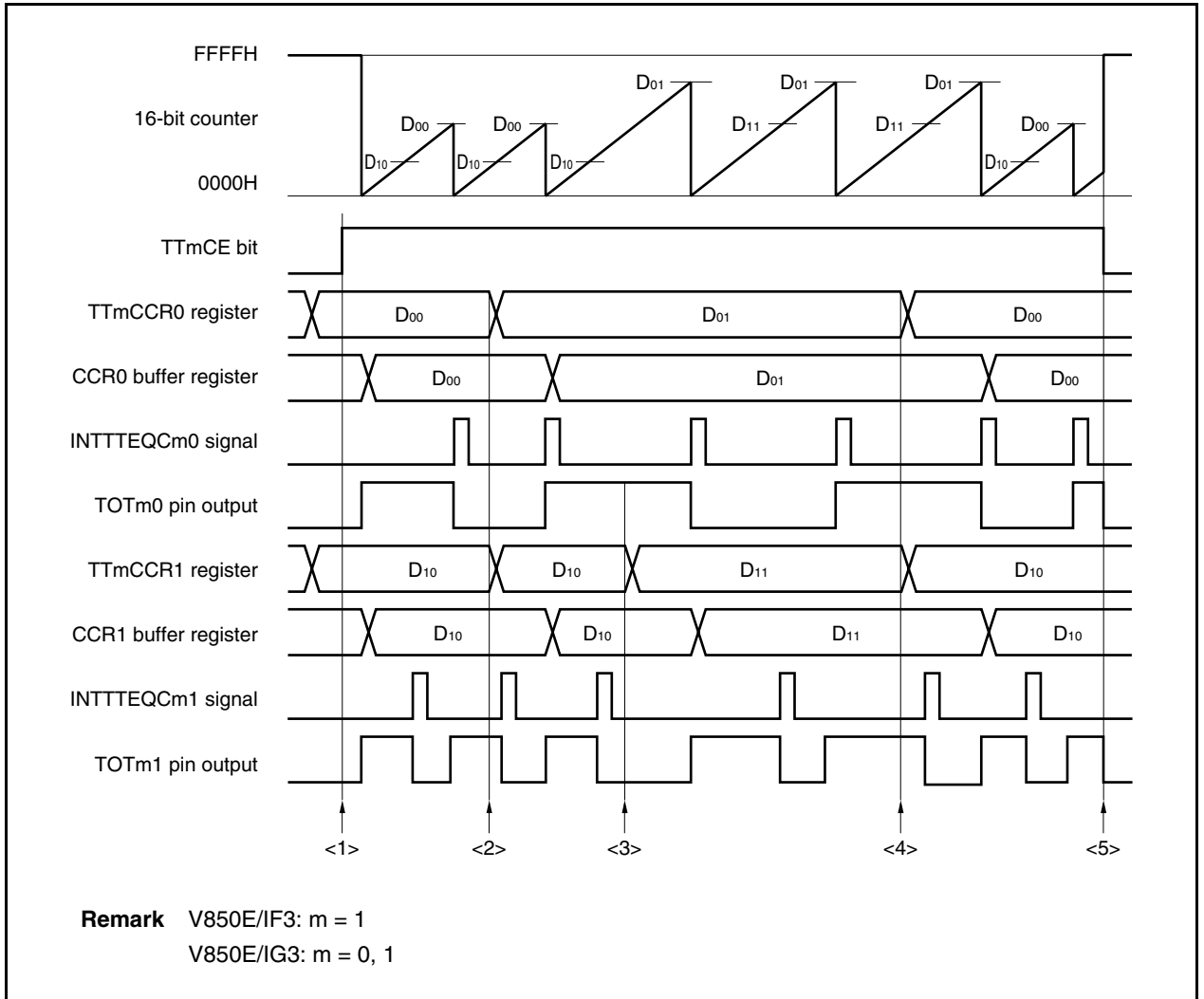
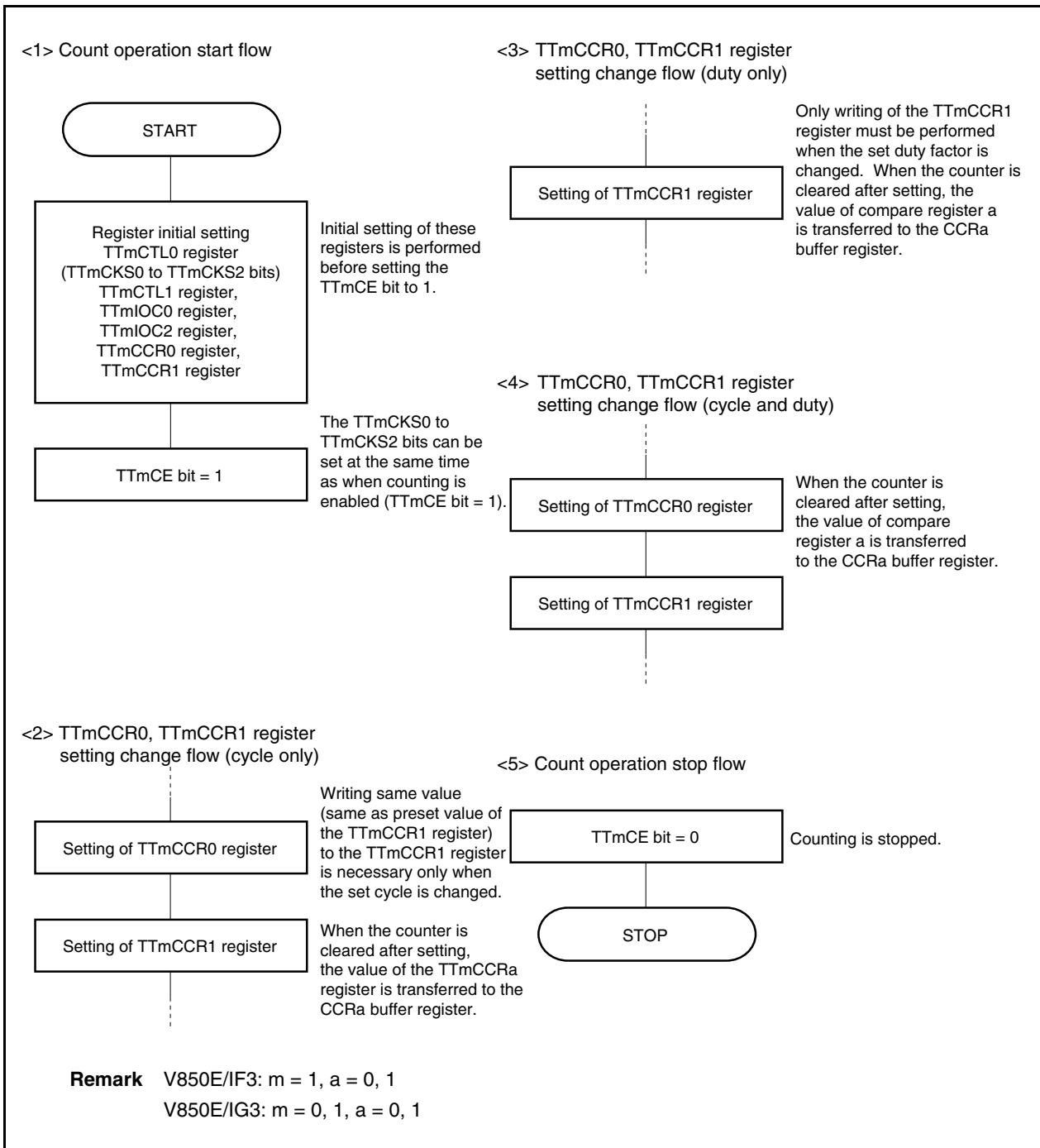


Figure 8-32. Software Processing Flow in PWM Output Mode (2/2)

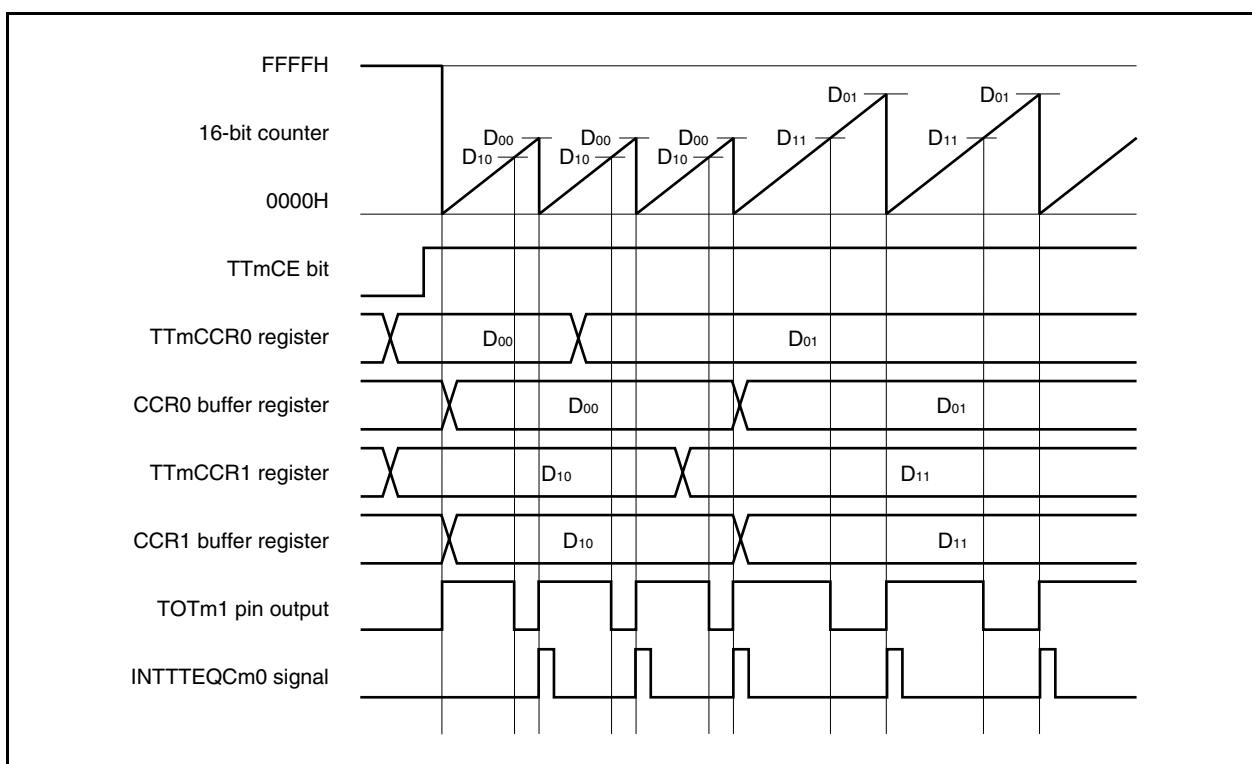


(2) PWM output mode operation timing

(a) Changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TTmCCR1 register last.

Rewrite the TTmCCRa register after writing the TTmCCR1 register after the INTTTEQCm0 signal is detected.



To transfer data from the TTmCCRa register to the CCRa buffer register, the TTmCCR1 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 TTmCCR0 register and then set the active level to the TTmCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TTmCCR0 register, and then write the same value (same as preset value of the TTmCCR1 register) to the TTmCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, only the TTmCCR1 register has to be set.

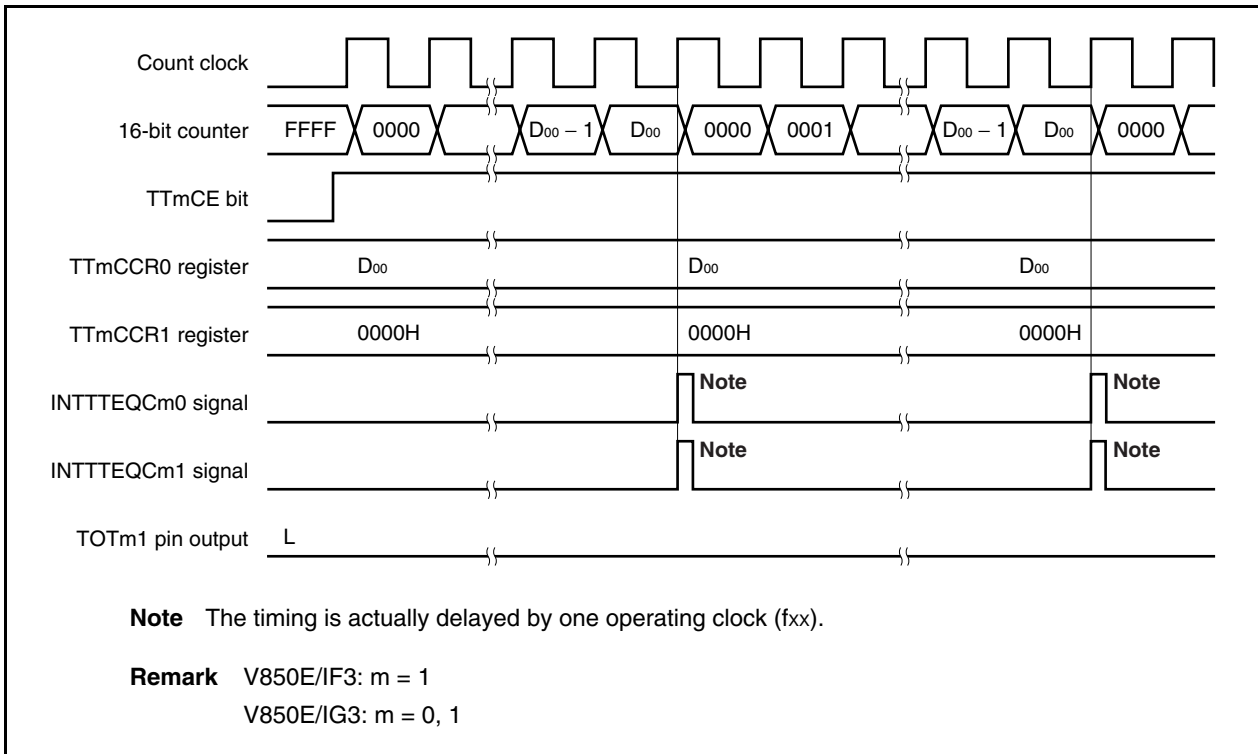
After data is written to the TTmCCR1 register, the value written to the TTmCCRa register is transferred to the CCRa buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TTmCCR0 or TTmCCR1 register again after writing the TTmCCR1 register once, do so after the INTTTEQCm0 signal is generated. Otherwise, the value of the CCRa buffer register may become undefined because the timing of transferring data from the TTmCCRa register to the CCRa buffer register conflicts with writing the TTmCCRa register.

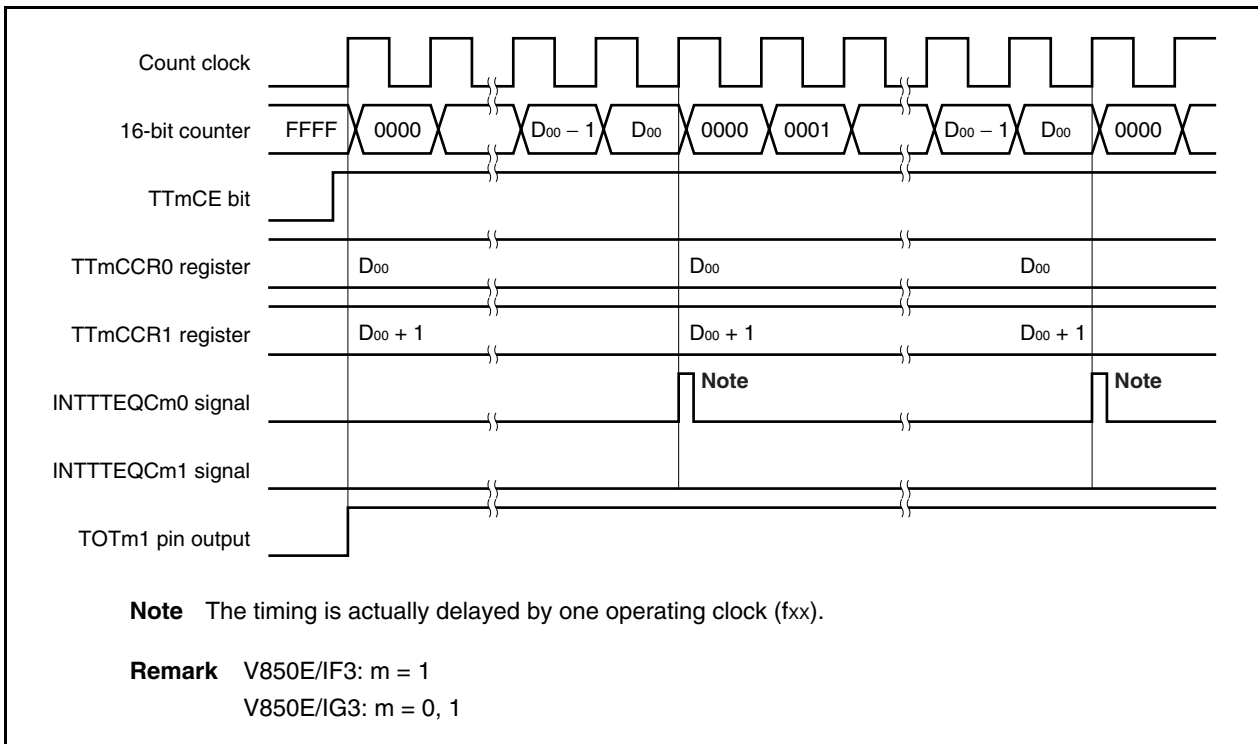
**Remark** V850E/IF3: m = 1, a = 0, 1, V850E/IG3: m = 0, 1, a = 0, 1

**(b) 0%/100% output of PWM waveform**

To output a 0% waveform, set the TTmCCR1 register to 0000H. The 16-bit counter is cleared to 0000H and the INTTTEQm0 and INTTTEQm1 signals are generated at the next timing after a match between the count value of the 16-bit counter and the value of the CCR0 buffer register.

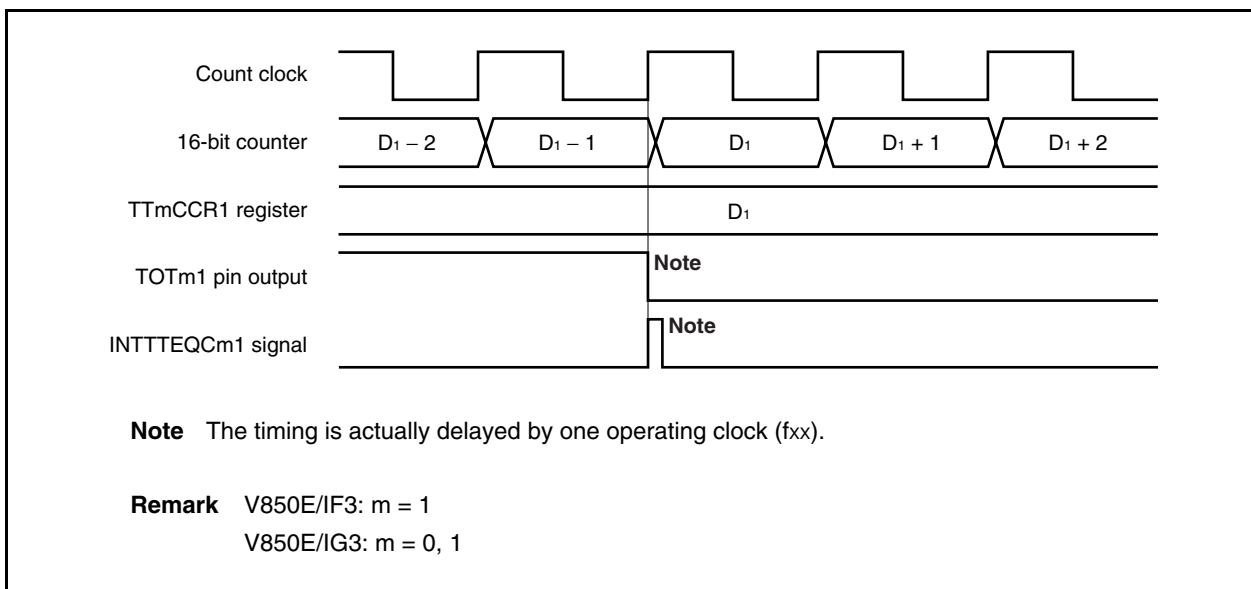


To output a 100% waveform, set a value of (set value of TTmCCR0 register + 1) to the TTmCCR1 register. If the set value of the TTmCCR0 register is FFFFH, 100% output cannot be produced.



**(c) Generation timing of compare match interrupt request signal (INTTTEQCm1)**

The timing of generation of the INTTTEQCm1 signal in the PWM output mode differs from the timing of INTTTEQCm1 signals in other modes; the INTTTEQCm1 signal is generated when the count value of the 16-bit counter matches the value of the TTmCCR1 register.



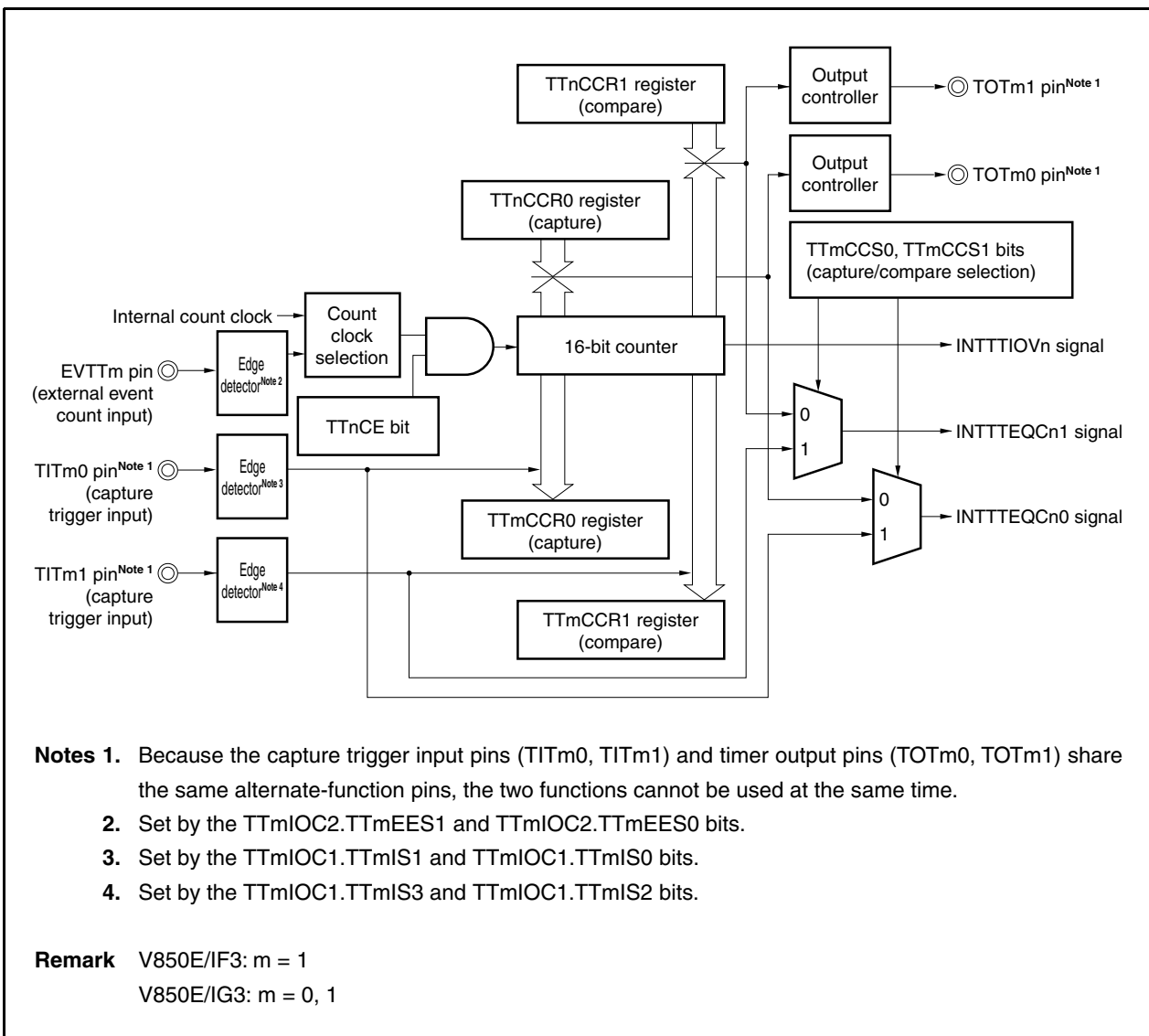
Usually, the INTTTEQCm1 signal is generated in synchronization with the next counting up after the count value of the 16-bit counter matches the value of the TTmCCR1 register. In the PWM output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the output signal of the TOTm1 pin.

**8.6.6 Free-running timer mode (TTnMD3 to TTnMD0 bits = 0101)**

The compare function is valid for all of TMT0 and TMT1. The capture function is valid only for TMT0 (V850E/IG3 only) and TMT1.

In the free-running timer mode, 16-bit timer/event counter T starts counting when the TTnCTL0.TTnCE bit is set to 1. At this time, the TTmCCR0 and TTmCCR1 registers can be used as compare registers or capture registers, depending on the setting of the TTmOPT0.TTmCCS0 and TTmOPT0.TTmCCS1 bits.

**Figure 8-33. Configuration in Free-Running Timer Mode**





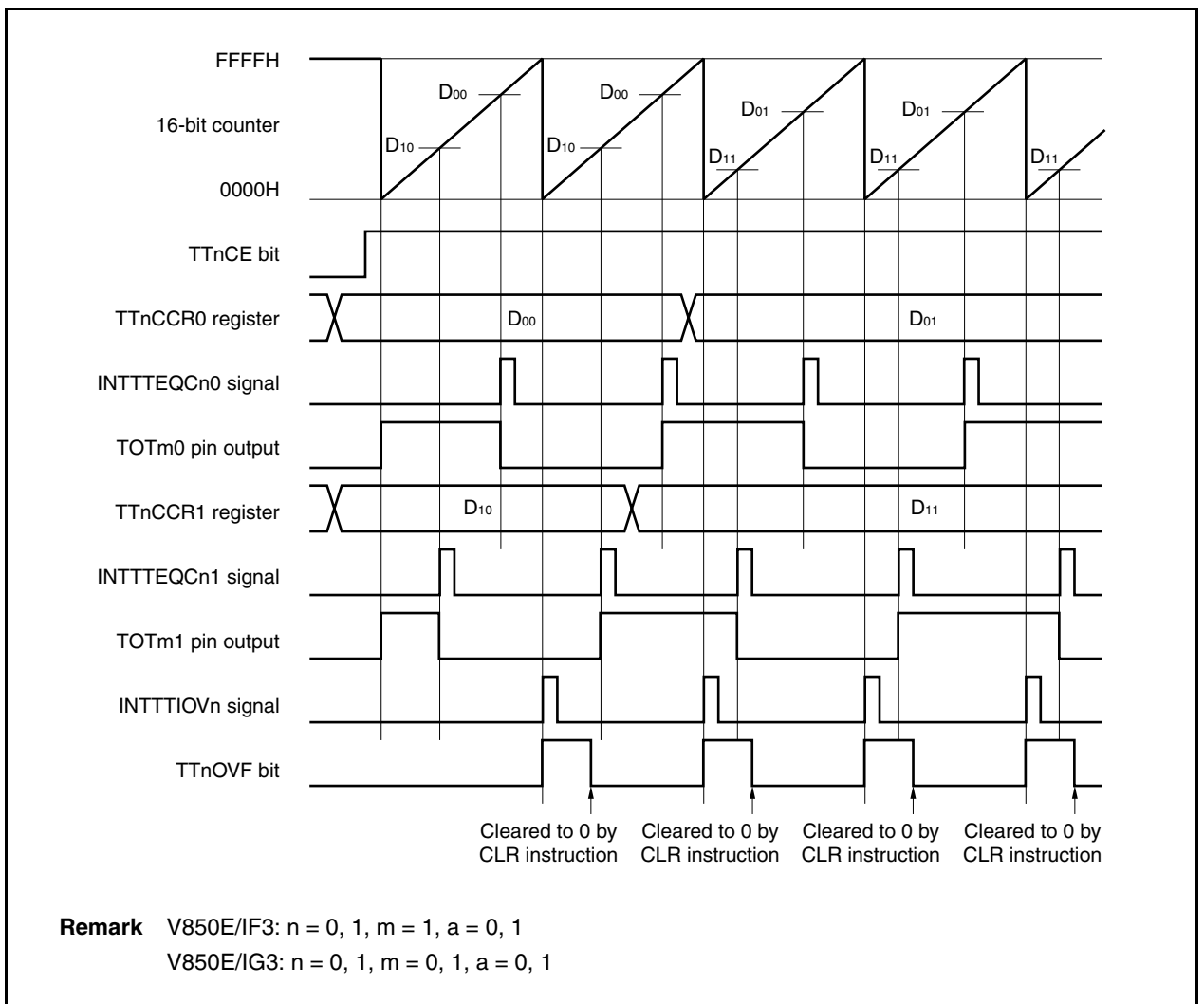
- Compare operation

When the TTnCE bit is set to 1, 16-bit timer/event counter T starts counting, and the output signal of the TOTma pin is inverted. When the count value of the 16-bit counter later matches the set value of the TTnCCRa register, a compare match interrupt request signal (INTTTEQCna) is generated, and the output signal of the TOTna 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 (INTTTIOVn) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TTnOPT0.TTnOVF bit) is also set to 1. Confirm that the overflow flag is set to 1 and then clear it to 0 by executing the CLR instruction via software.

The TTnCCRa register can be rewritten while the counter is operating. If it is rewritten, the new value is reflected at that time by anytime write, and compared with the count value.

**Figure 8-34. Basic Timing in Free-Running Timer Mode (Compare Function)**



- Capture operation

When the TTmCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TITma pin is detected, the count value of the 16-bit counter is stored in the TTmCCRm register, and a capture interrupt request signal (INTTTEQcm) 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 (INTTTIOVm) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TTmOPT0.TTmOVF bit) is also set to 1. Confirm that the overflow flag is set to 1 and then clear it to 0 by executing the CLR instruction via software.

**Figure 8-35. Basic Timing in Free-Running Timer Mode (Capture Function)**

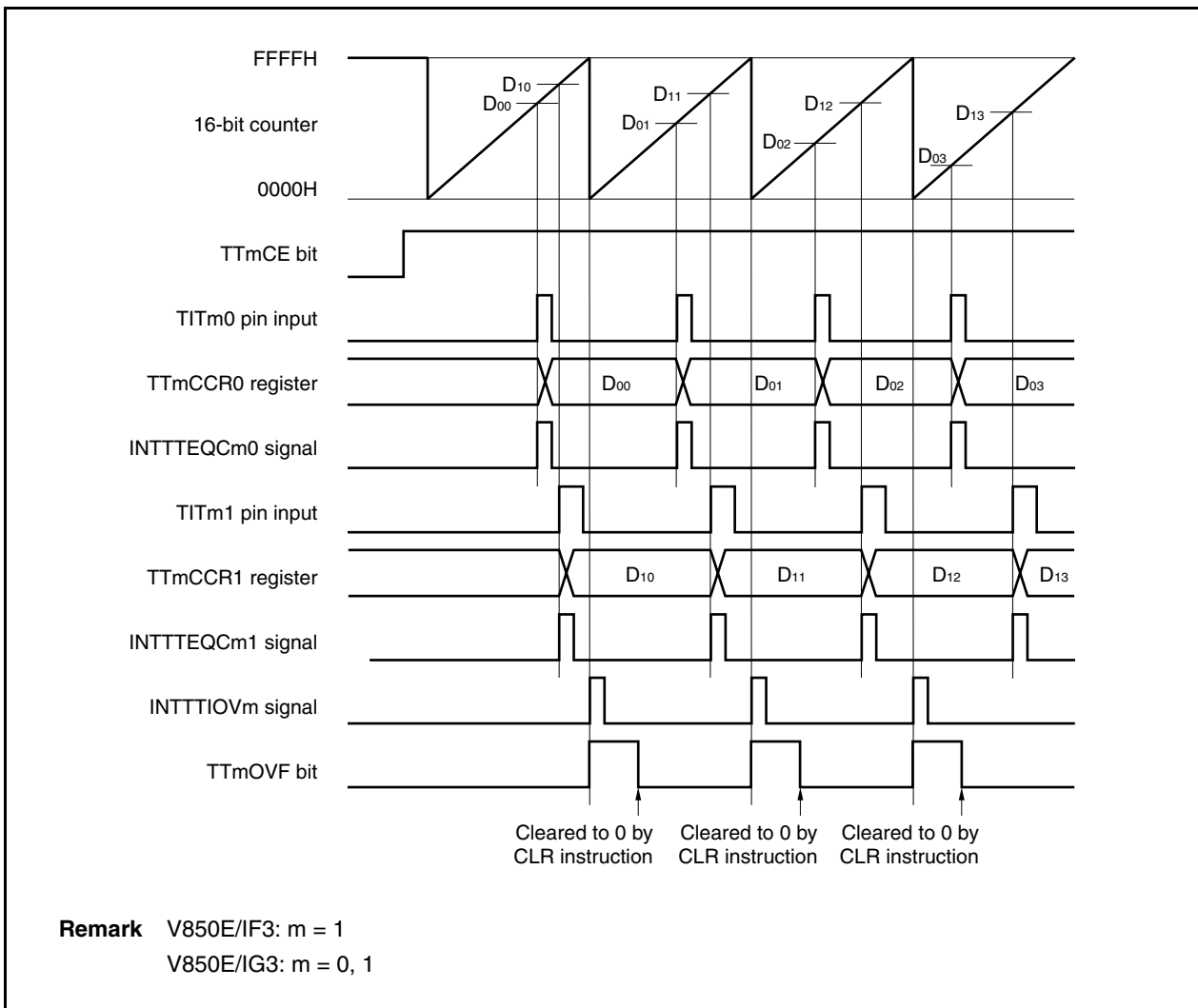


Figure 8-36. Register Setting in Free-Running Timer Mode (1/2)

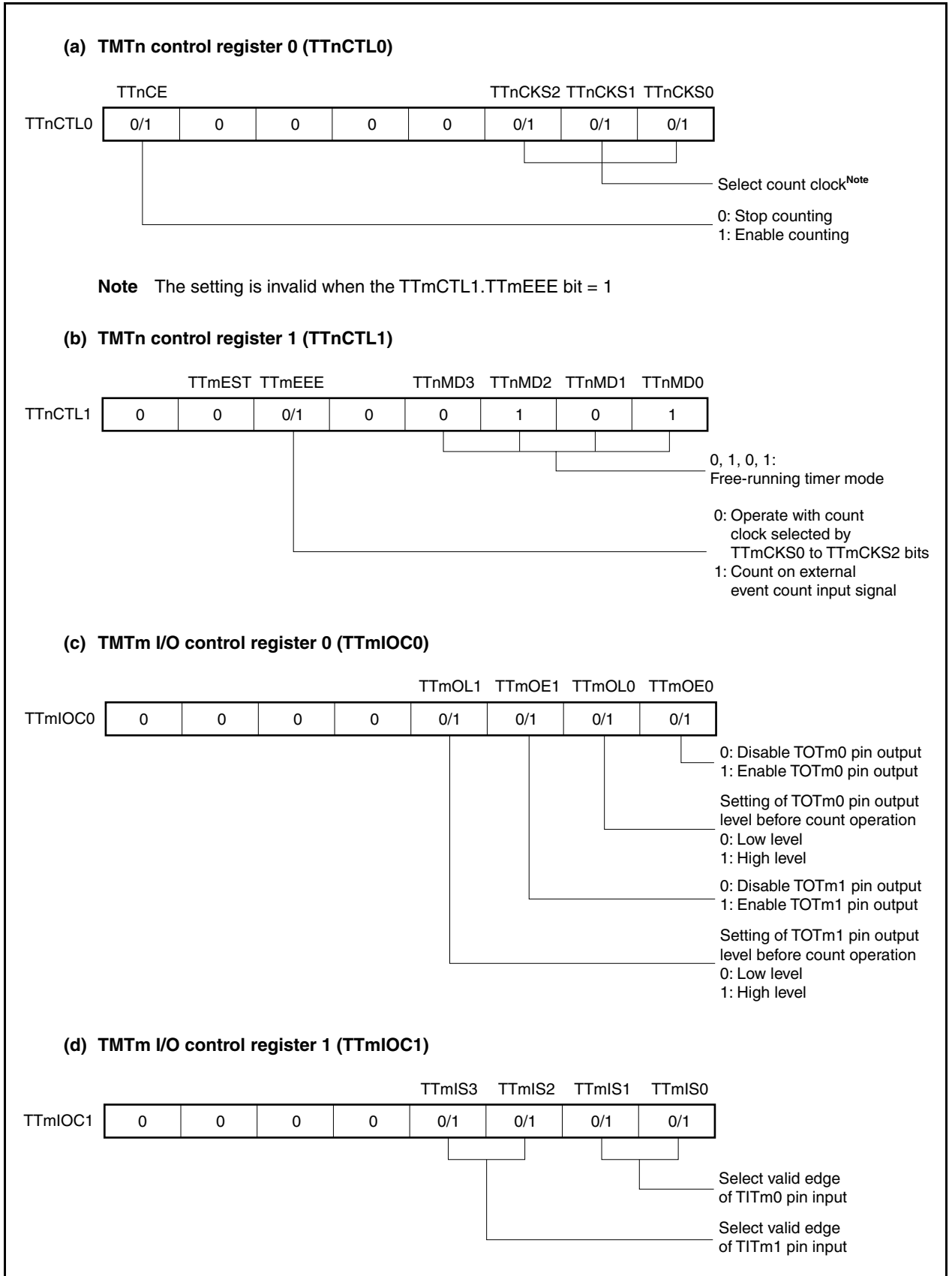
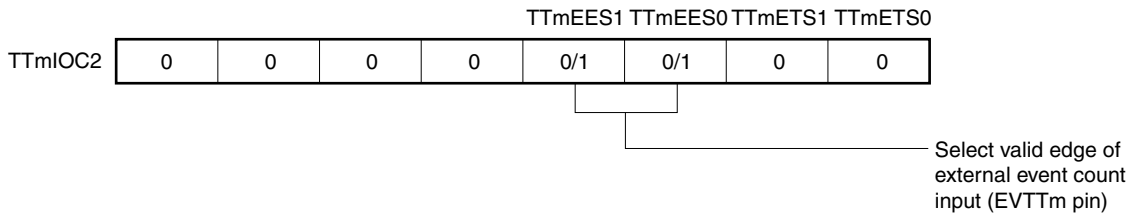
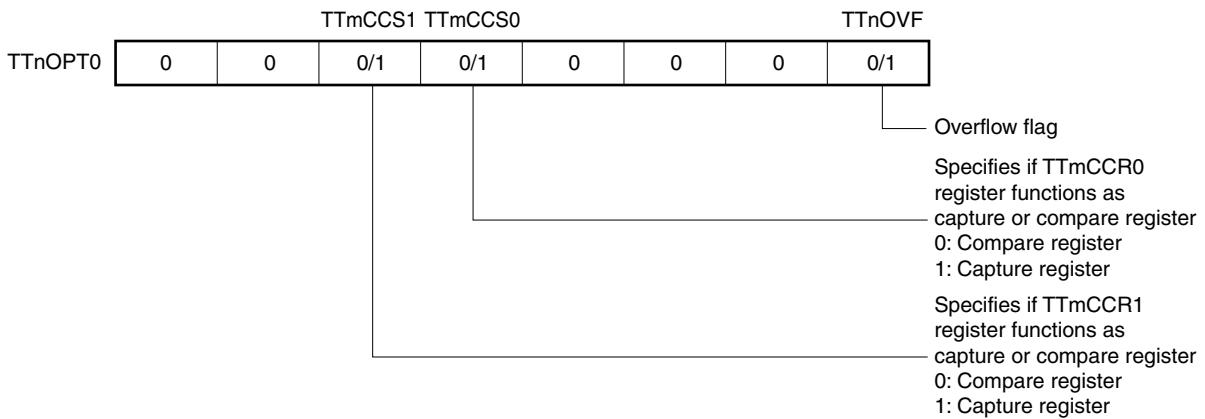


Figure 8-36. Register Setting in Free-Running Timer Mode (2/2)

(e) TMTm I/O control register 2 (TTmIOC2)



(f) TMTn option register 0 (TTnOPT0)



(g) TMTn counter read buffer register (TTnCNT)

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

(h) TMTn capture/compare registers 0 and 1 (TTnCCR0 and TTnCCR1)

These registers function as capture registers or compare registers depending on the setting of the TTmOPT0.TTmCCSa bit.

When the registers function as capture registers, they store the count value of the 16-bit counter when the valid edge input to the TITma pin is detected.

When the registers function as compare registers and when Da is set to the TTnCCRa register, the INTTTEQCna signal is generated when the counter reaches (Da + 1), and the output signals of the TOTm0 and TOTm1 pins are inverted.

**Remark** V850E/IF3: n = 0, 1, m = 1, a = 0, 1  
V850E/IG3: n = 0, 1, m = 0, 1, a = 0, 1

(1) Operation flow in free-running timer mode

(a) When using capture/compare register as compare register

Figure 8-37. Software Processing Flow in Free-Running Timer Mode (Compare Function) (1/2)

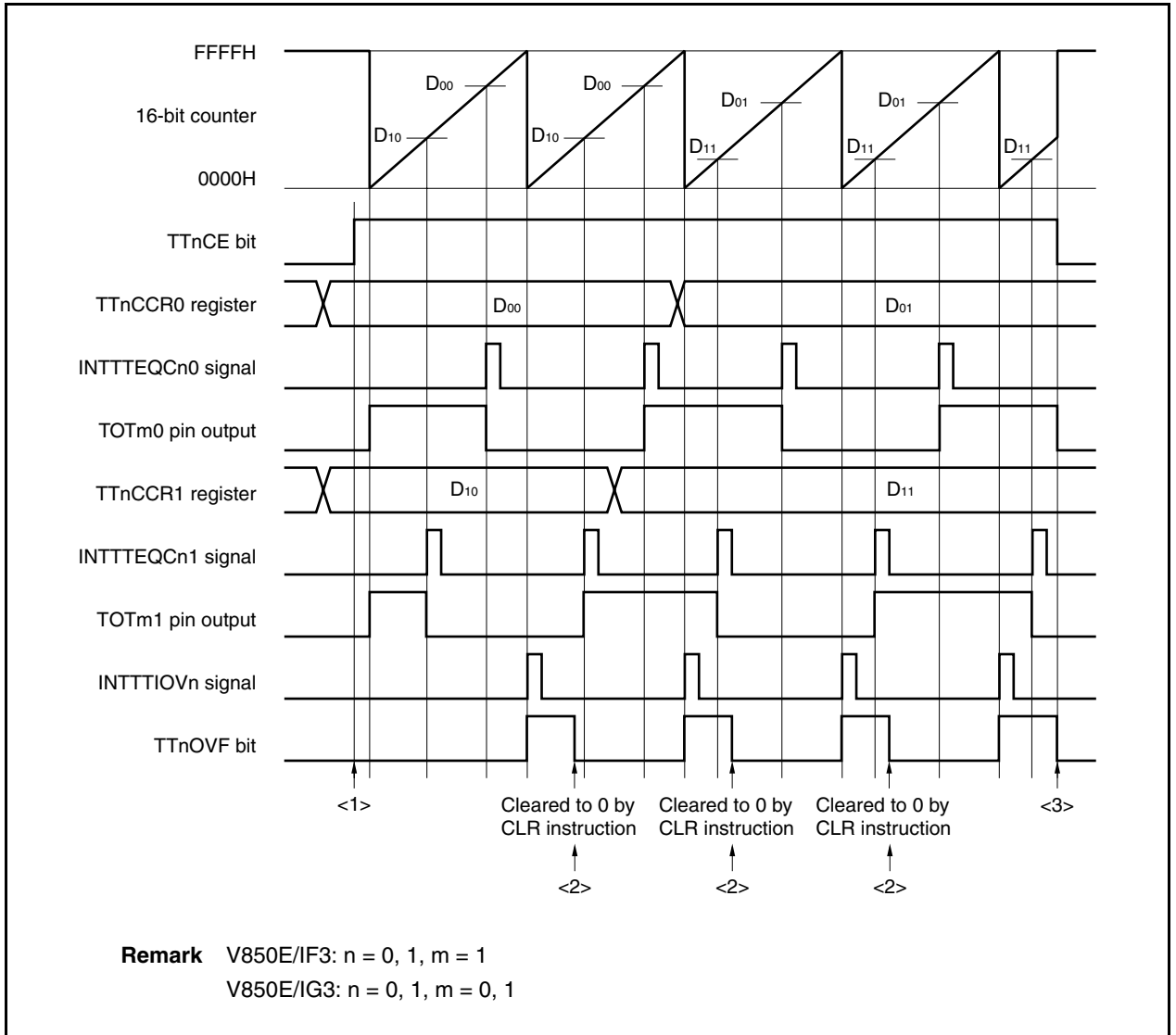
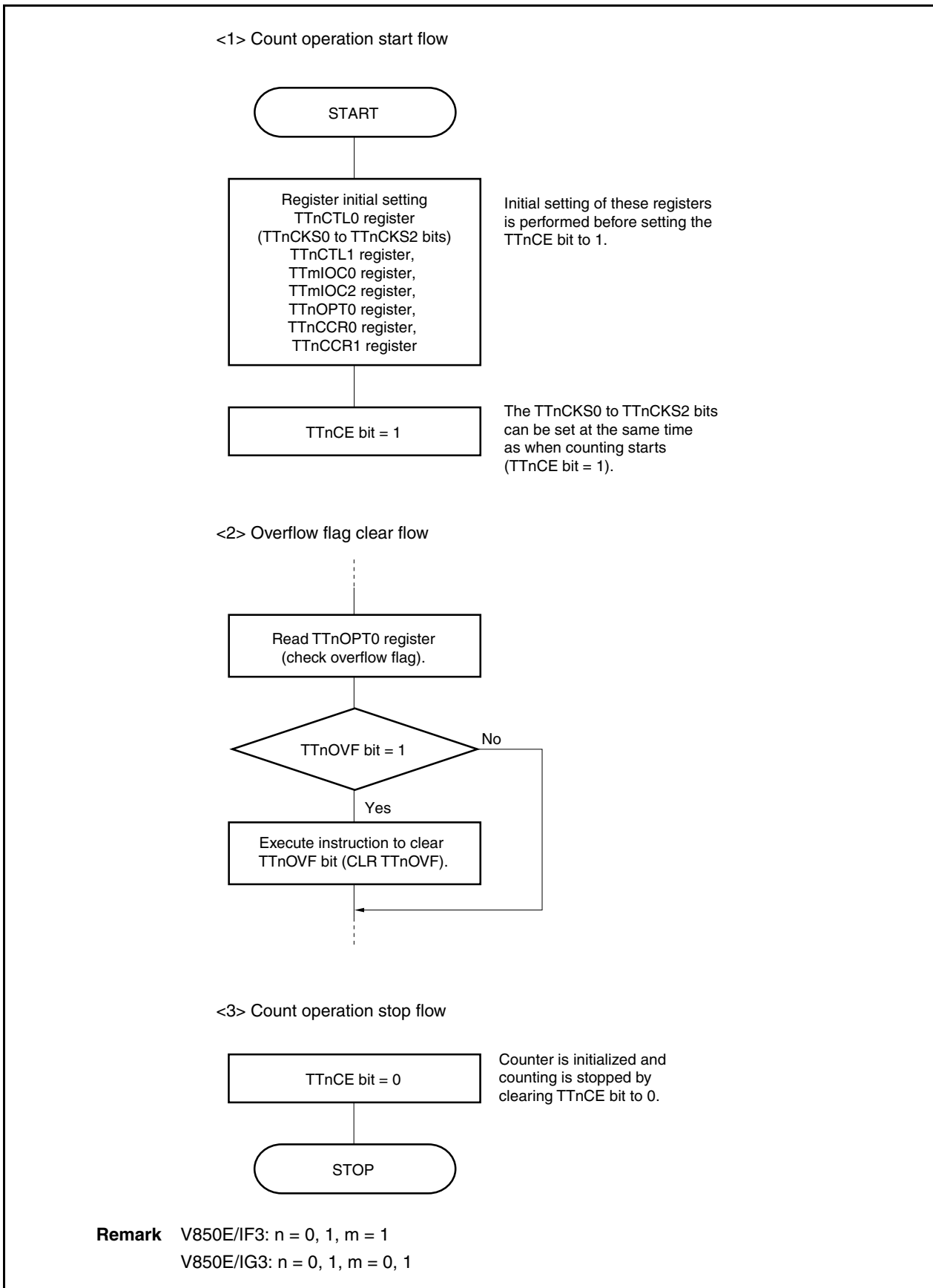


Figure 8-37. Software Processing Flow in Free-Running Timer Mode (Compare Function) (2/2)



(b) When using capture/compare register as capture register

Figure 8-38. Software Processing Flow in Free-Running Timer Mode (Capture Function) (1/2)

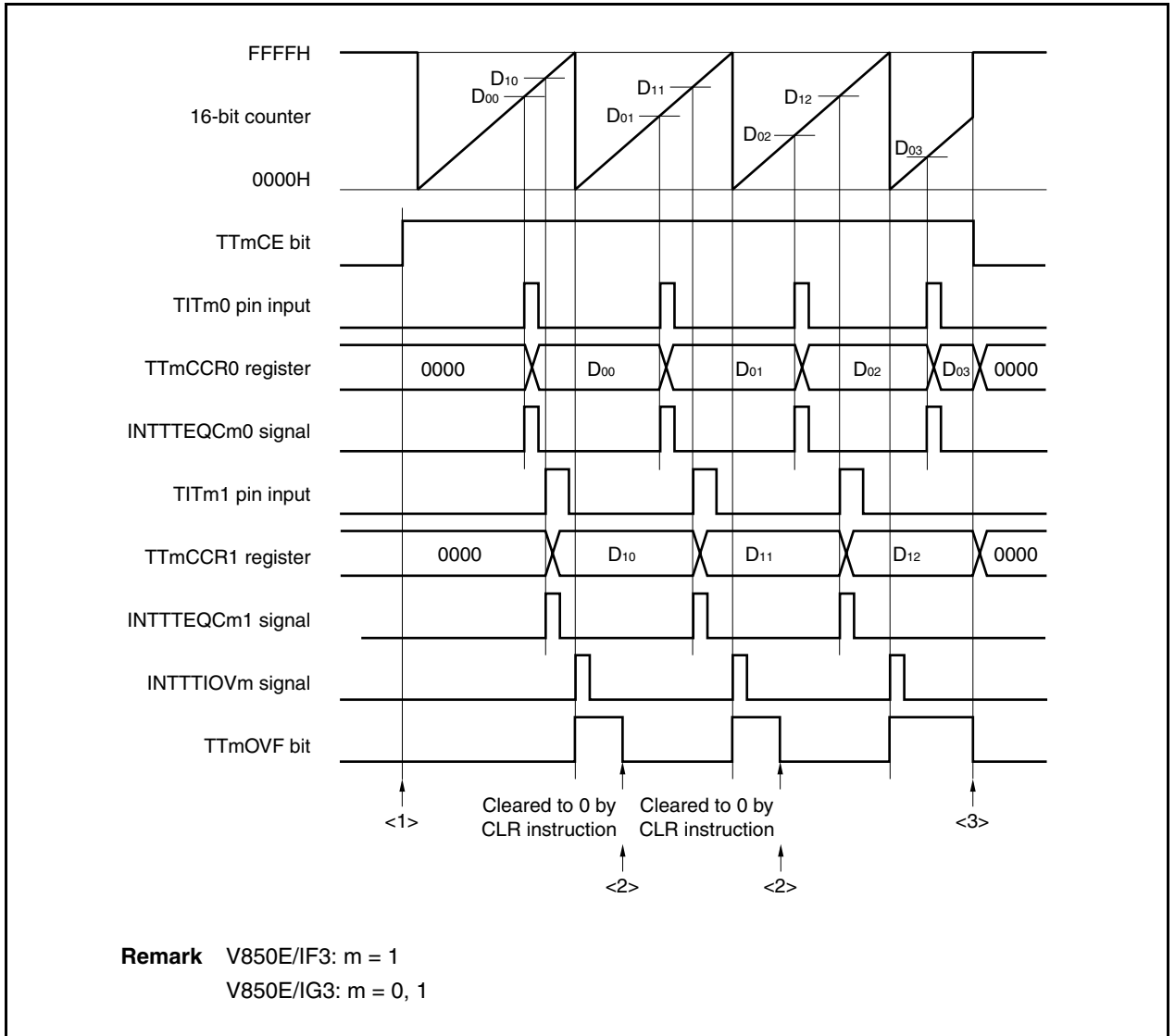
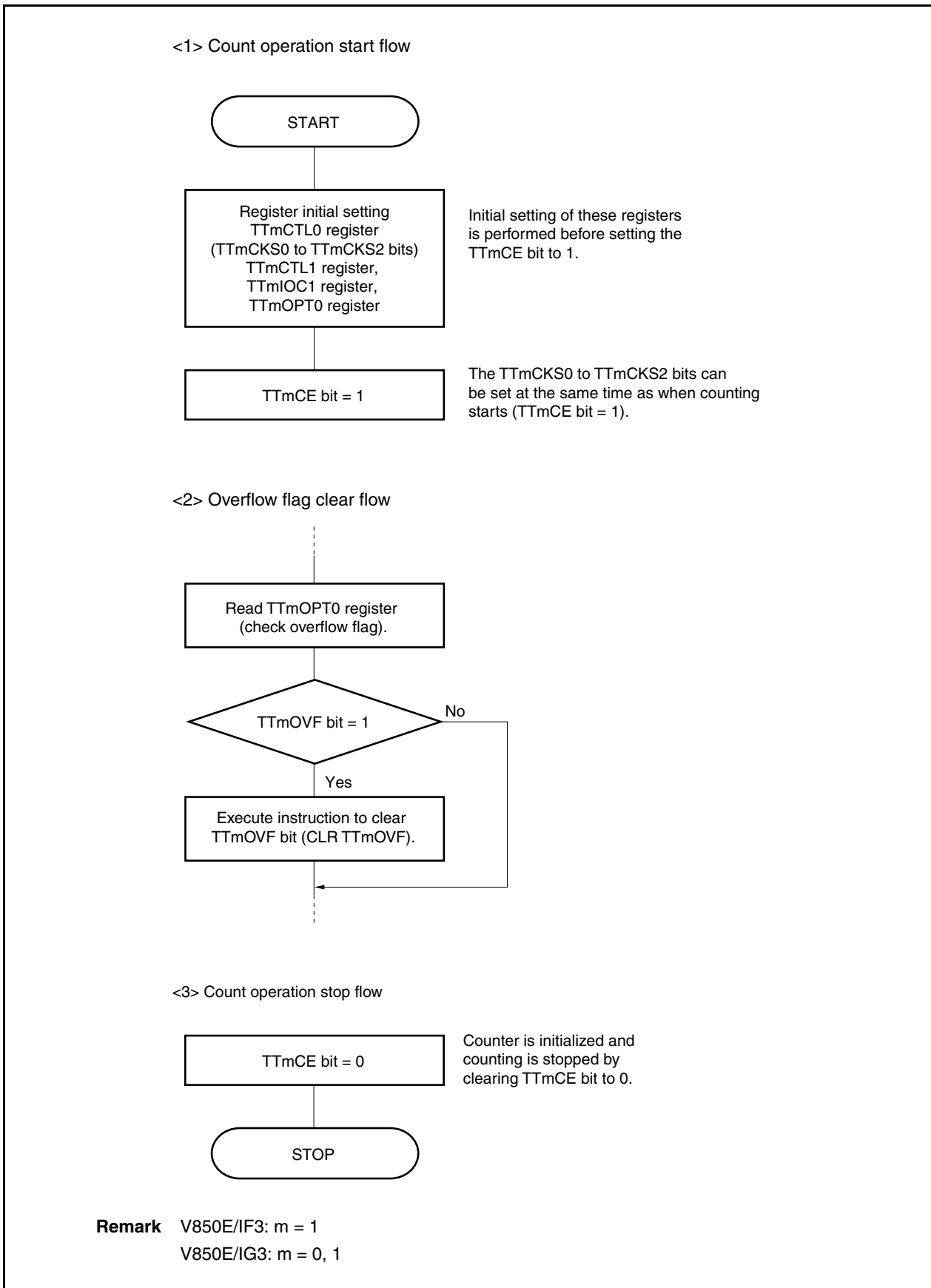


Figure 8-38. Software Processing Flow in Free-Running Timer Mode (Capture Function) (2/2)

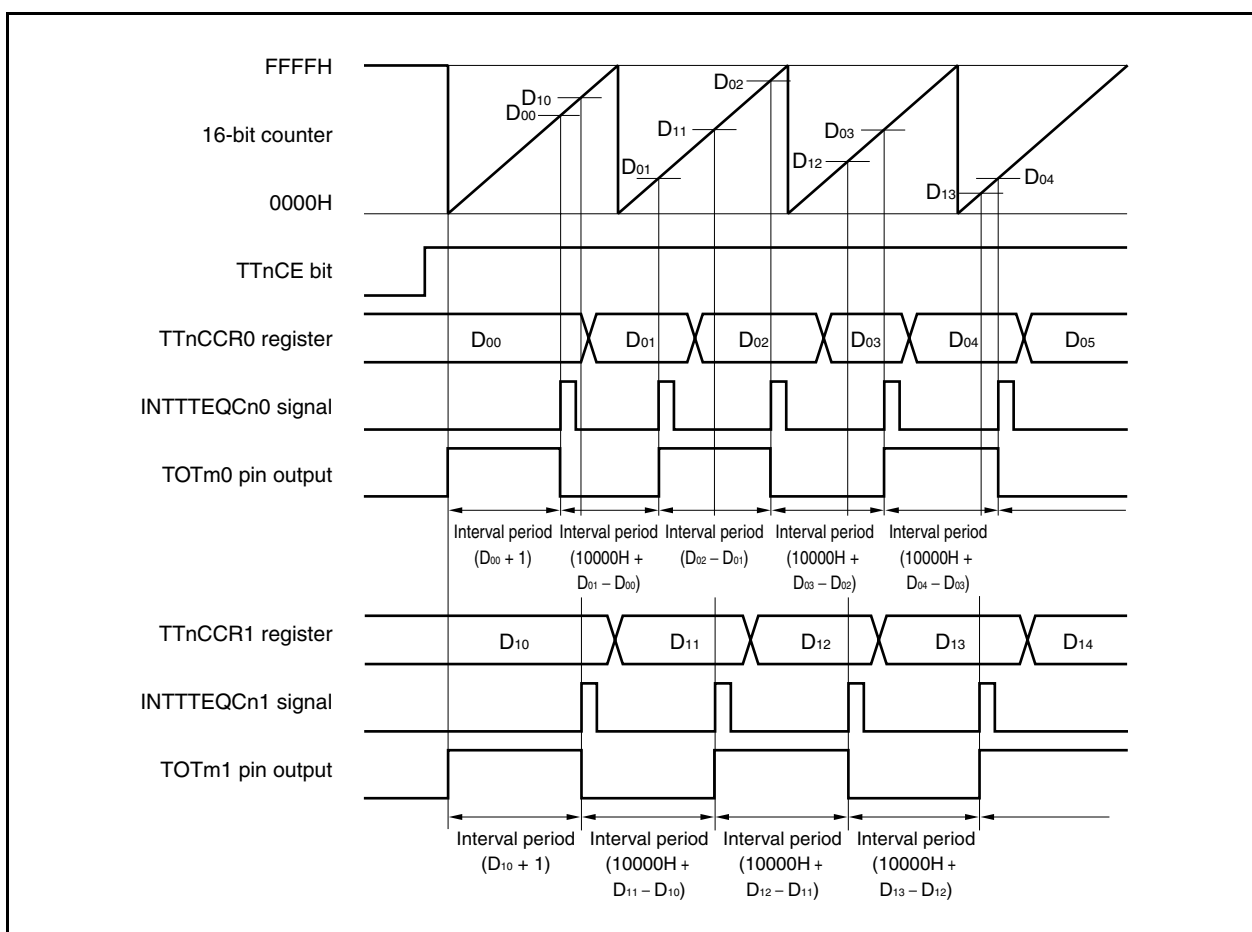




(2) Operation timing in free-running timer mode

(a) Interval operation with compare register

When 16-bit timer/event counter T is used as an interval timer with the TTnCCRa 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 INTTTEQCna 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 TTnCCRa register must be re-set in the interrupt servicing that is executed when the INTTTEQCna signal is detected.

The set value for re-setting the TTnCCRa register can be calculated by the following expression, where “D<sub>a</sub>” is the interval period.

Compare register default value:  $D_a - 1$

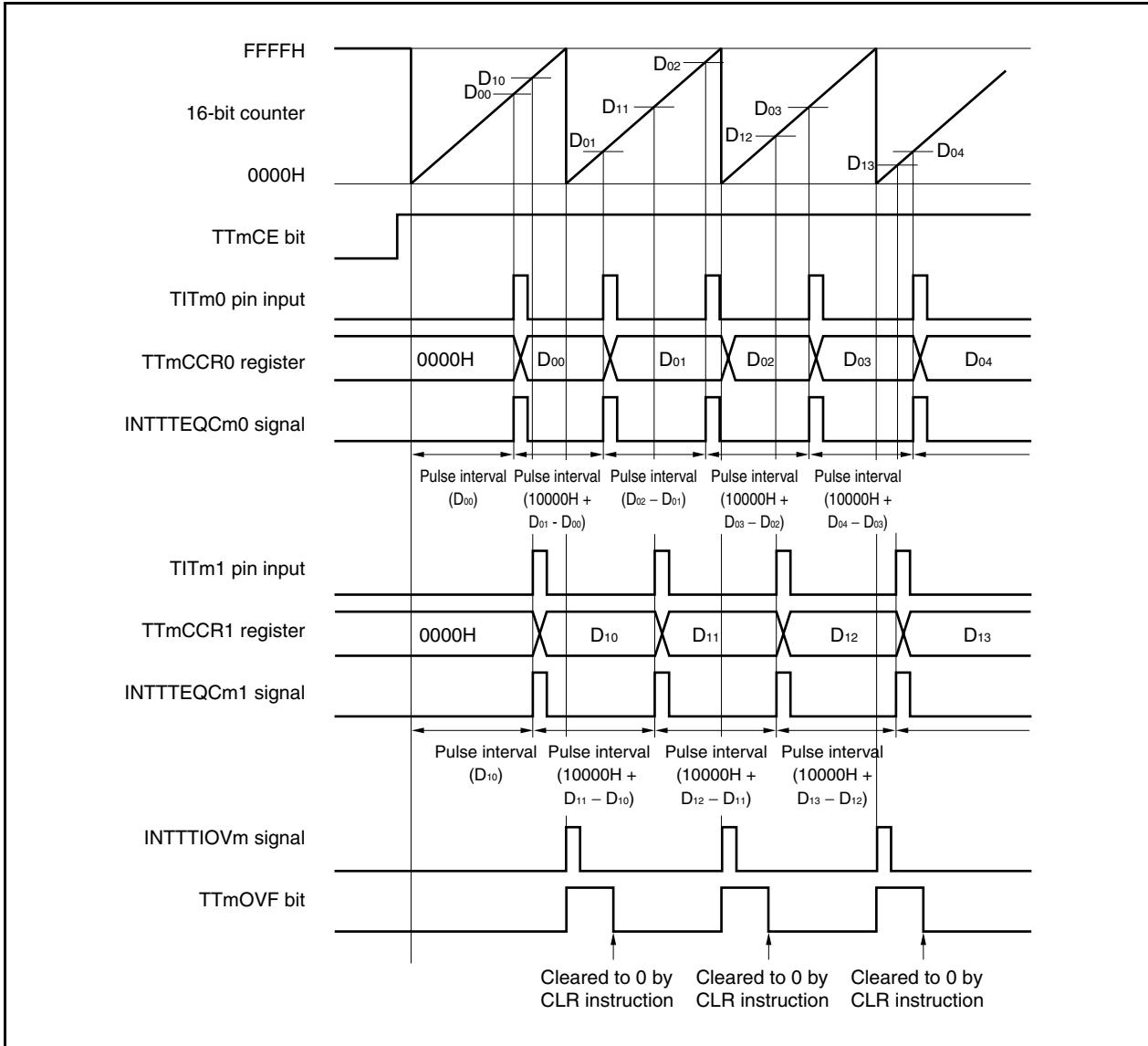
Value set to compare register second and subsequent time: Previous set value +  $D_a$

(If the calculation result is greater than FFFFH, subtract 10000H from the result and set this value to the register.)

**Remark** V850E/IF3: n = 0, 1, m = 1, a = 0, 1  
V850E/IG3: n = 0, 1, m = 0, 1, a = 0, 1

**(b) Pulse width measurement with capture register**

When pulse width measurement is performed with the TTmCCRa register used as a capture register, software processing is necessary for reading the capture register each time the INTTTEQCma signal has been detected and for calculating an interval.



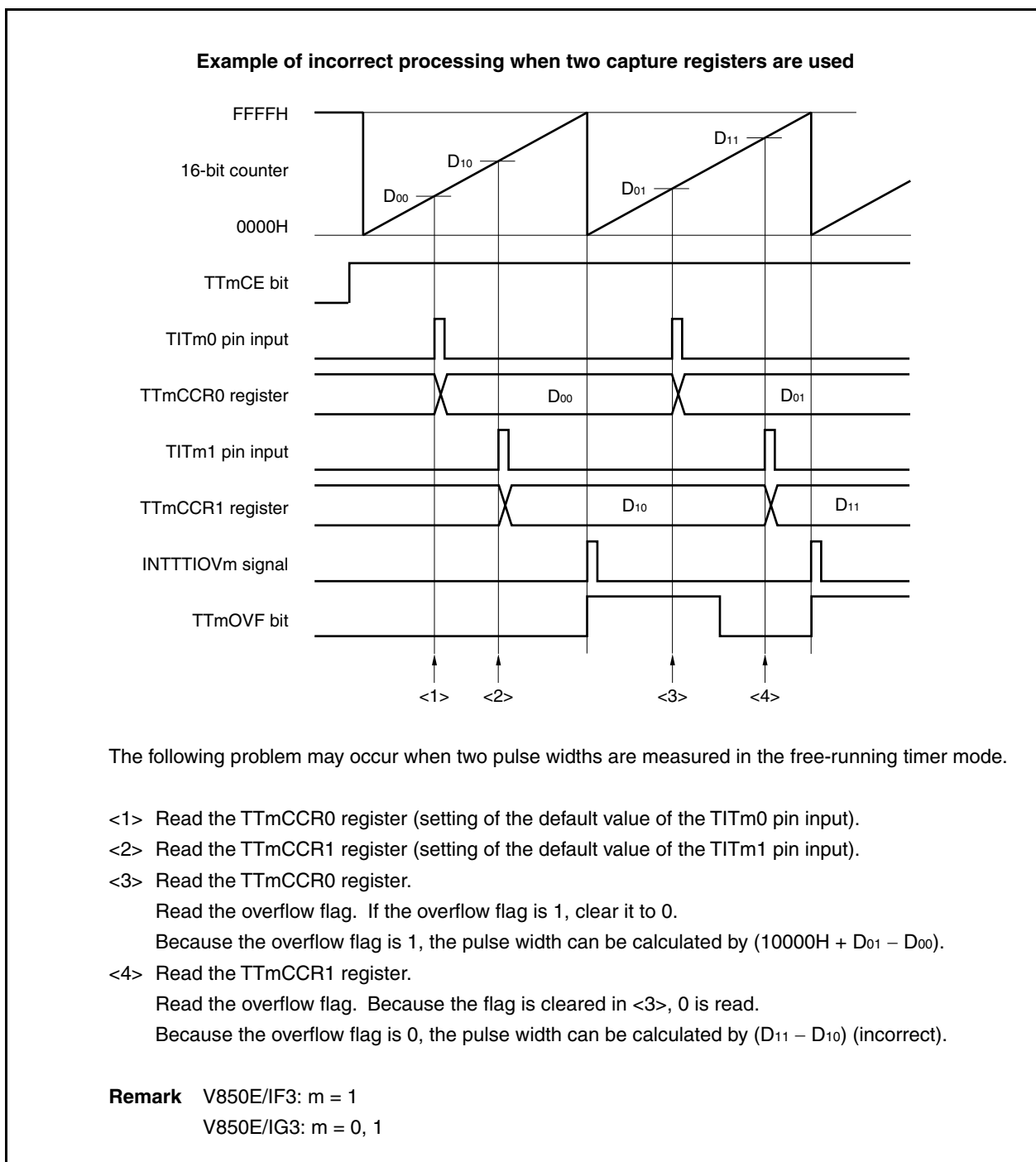
When executing pulse width measurement in the free-running timer mode, two pulse widths can be measured with one channel.

To measure a pulse width, the pulse width can be calculated by reading the value of the TTmCCRa register in synchronization with the INTTTEQCma signal, and calculating the difference between the read value and the previously read value.

**Remark** V850E/IF3: m = 1, a = 0, 1  
 V850E/IG3: m = 0, 1, a = 0, 1

**(c) Processing of overflow when two capture registers are used**

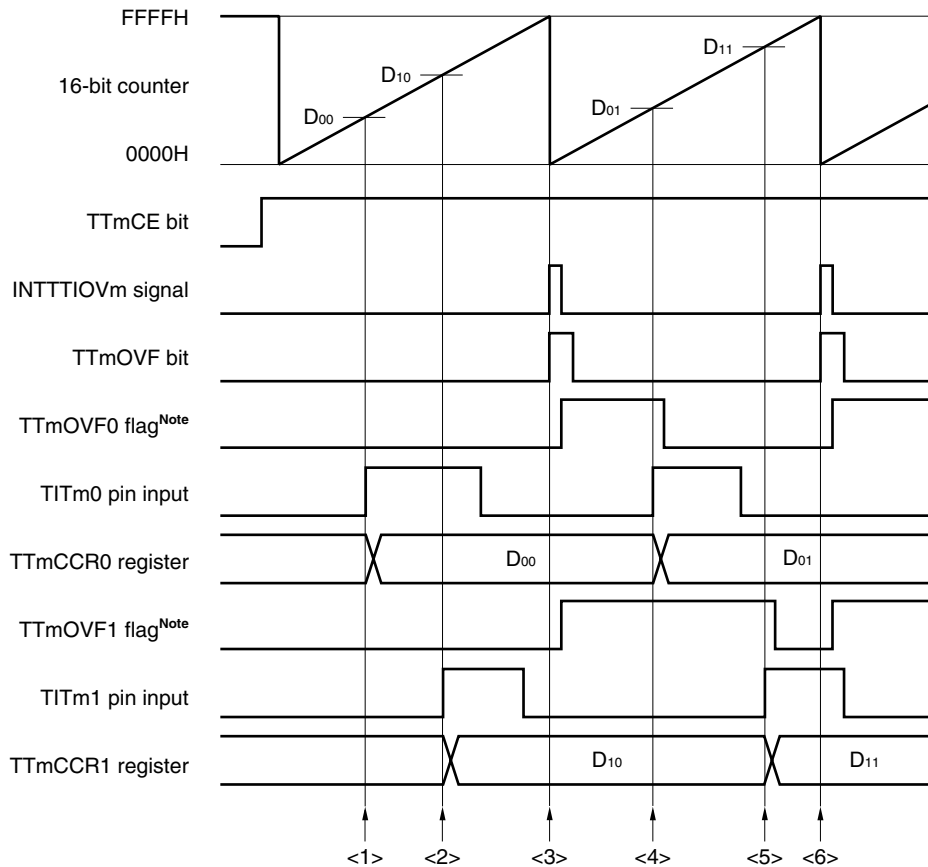
Care must be exercised in processing the overflow flag when two capture registers are used. First, an example of incorrect processing is shown below.



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.

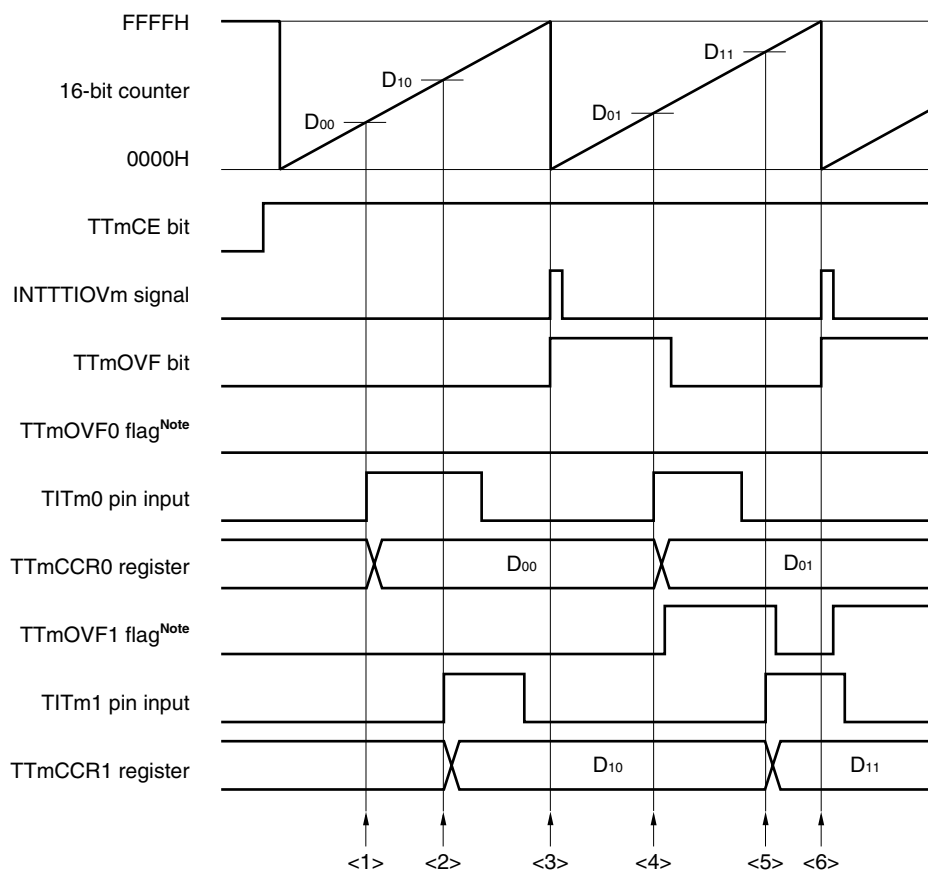
### Example when two capture registers are used (using overflow interrupt)



**Note** The TTmOVF0 and TTmOVF1 flags are set on the internal RAM by software.

- <1> Read the TTmCCR0 register (setting of the default value of the TITm0 pin input).
- <2> Read the TTmCCR1 register (setting of the default value of the TITm1 pin input).
- <3> An overflow occurs. Set the TTmOVF0 and TTmOVF1 flags to 1 in the overflow interrupt servicing, and clear the overflow flag to 0.
- <4> Read the TTmCCR0 register.  
Read the TTmOVF0 flag. If the TTmOVF0 flag is 1, clear it to 0.  
Because the TTmOVF0 flag is 1, the pulse width can be calculated by  $(10000H + D_{01} - D_{00})$ .
- <5> Read the TTmCCR1 register.  
Read the TTmOVF1 flag. If the TTmOVF1 flag is 1, clear it to 0 (the TTmOVF0 flag is cleared in <4>, and the TTmOVF1 flag remains 1).  
Because the TTmOVF1 flag is 1, the pulse width can be calculated by  $(10000H + D_{11} - D_{10})$  (correct).
- <6> Same as <3>

**Remark** V850E/IF3:  $m = 1$   
V850E/IG3:  $m = 0, 1$

**Example when two capture registers are used (without using overflow interrupt)**

**Note** The TTmOVf0 and TTmOVf1 flags are set on the internal RAM by software.

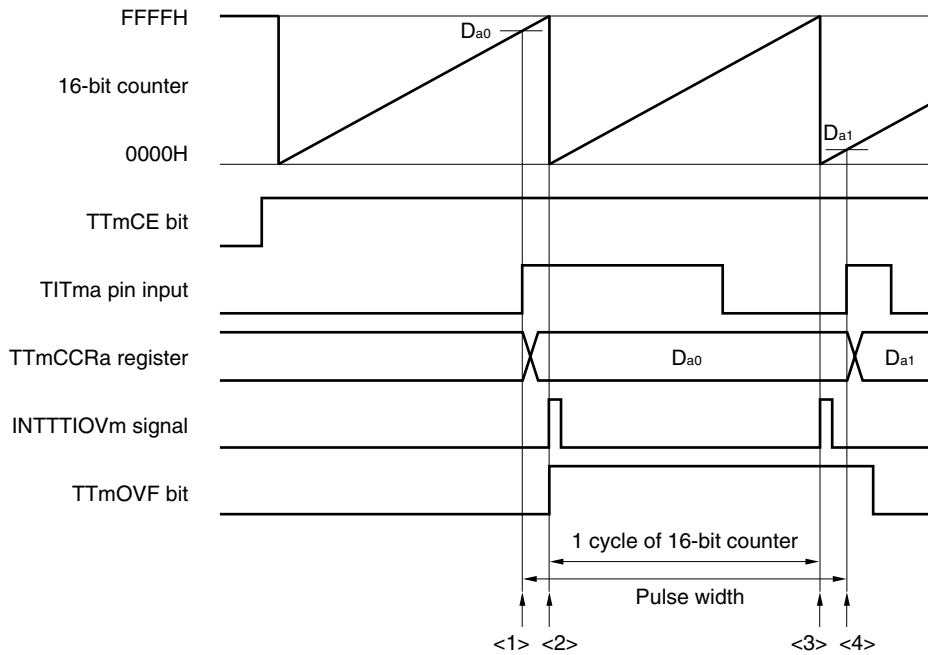
- <1> Read the TTmCCR0 register (setting of the default value of the TITm0 pin input).
- <2> Read the TTmCCR1 register (setting of the default value of the TITm1 pin input).
- <3> An overflow occurs. Nothing is done by software.
- <4> Read the TTmCCR0 register.  
Read the overflow flag. If the overflow flag is 1, set only the TTmOVf1 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 TTmCCR1 register.  
Read the overflow flag. Because the overflow flag is cleared in <4>, 0 is read.  
Read the TTmOVf1 flag. If the TTmOVf1 flag is 1, clear it to 0.  
Because the TTmOVf1 flag is 1, the pulse width can be calculated by  $(10000H + D_{11} - D_{10})$  (correct).
- <6> Same as <3>

**Remark** V850E/IF3:  $m = 1$   
V850E/IG3:  $m = 0, 1$

**(d) Processing of overflow if capture trigger interval is long**

If the pulse width is greater than one cycle of the 16-bit counter, care must be exercised because an overflow may occur more than once from the first capture trigger to the next. First, an example of incorrect processing is shown below.

**Example of incorrect processing when capture trigger interval is long**



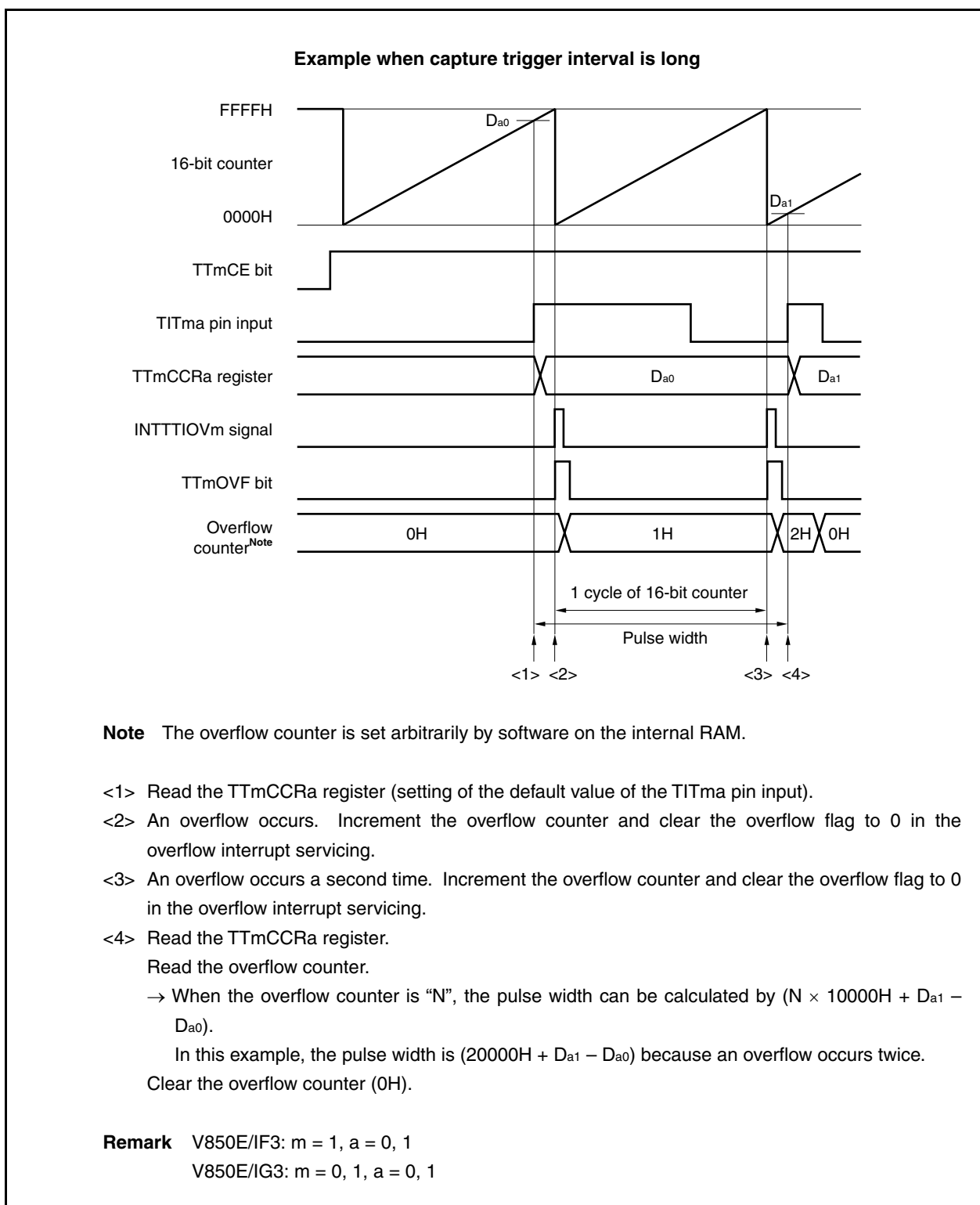
The following problem may occur when long pulse width is measured in the free-running timer mode.

- <1> Read the TTmCCRa register (setting of the default value of the TITma pin input).
- <2> An overflow occurs. Nothing is done by software.
- <3> An overflow occurs a second time. Nothing is done by software.
- <4> Read the TTmCCRa 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  $(10000H + D_{a1} - D_{a0})$  (incorrect).  
 Actually, the pulse width must be  $(20000H + D_{a1} - D_{a0})$  because an overflow occurs twice.

**Remark** V850E/IF3: m = 1, a = 0, 1  
 V850E/IG3: m = 0, 1, a = 0, 1

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 TTmOVF bit to 0 with the CLR instruction after reading the TTmOVF bit when it is 1 and by writing 8-bit data (bit 0 is 0) to the TTmOPT0 register after reading the TTmOVF bit when it is 1.

**8.6.7 Pulse width measurement mode (TTmMD3 to TTmMD0 bits = 0110)**

This mode is valid only in TMT0 (V850E/IG3 only) and TMT1.

In the pulse width measurement mode, 16-bit timer/event counter T starts counting when the TTmCTL0.TTmCE bit is set to 1. Each time the valid edge input to the TITma pin has been detected, the count value of the 16-bit counter is stored in the TTmCCRa register, and the 16-bit counter is cleared to 0000H.

The interval of the valid edge can be measured by reading the TTmCCRa register after a capture interrupt request signal (INTTTEQCma) occurs.

As shown in Figure 8-40, select either the TITm0 or TITm1 pin as the capture trigger input pin and set the unused pins to “No edge detection” by using the TTmIOC1 register.

**Figure 8-39. Configuration in Pulse Width Measurement Mode**

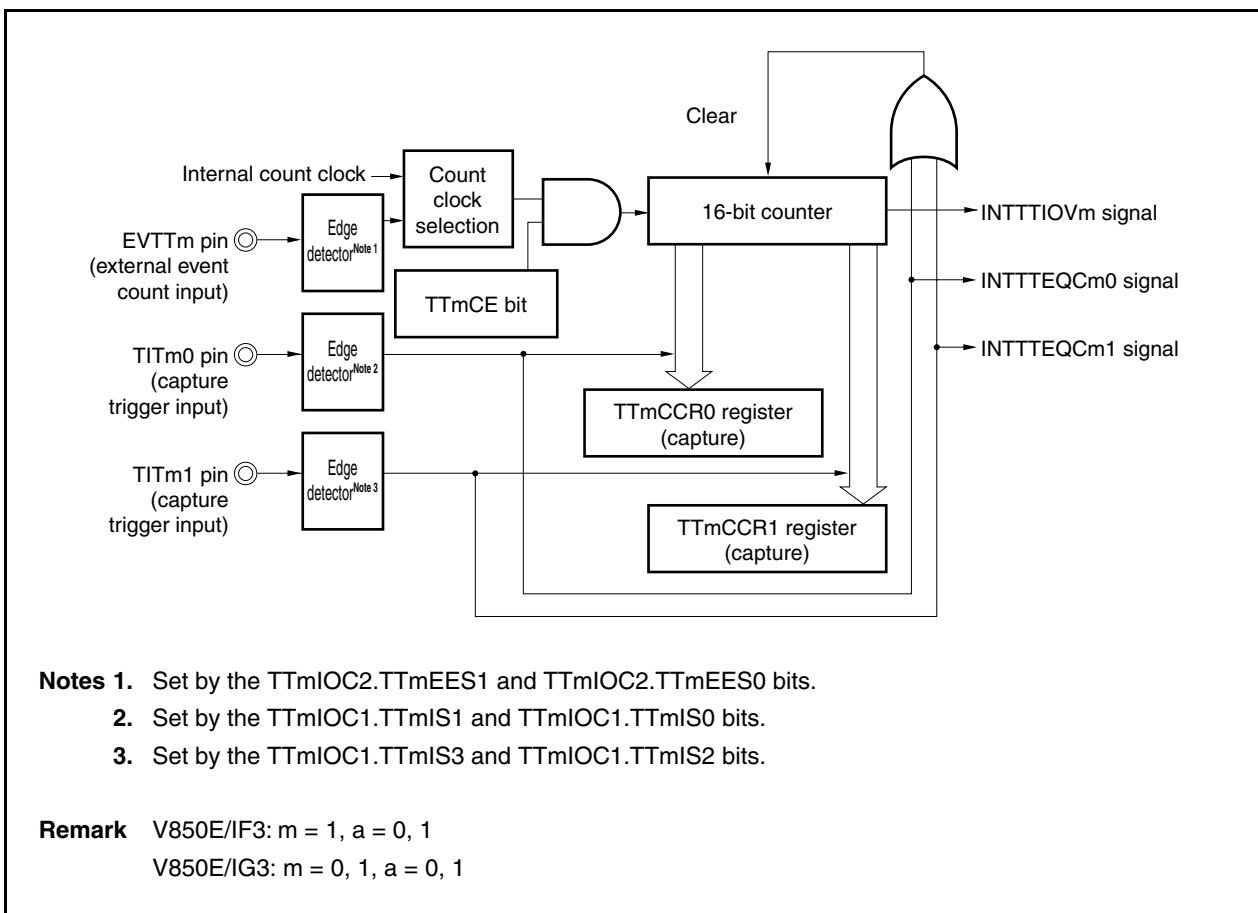
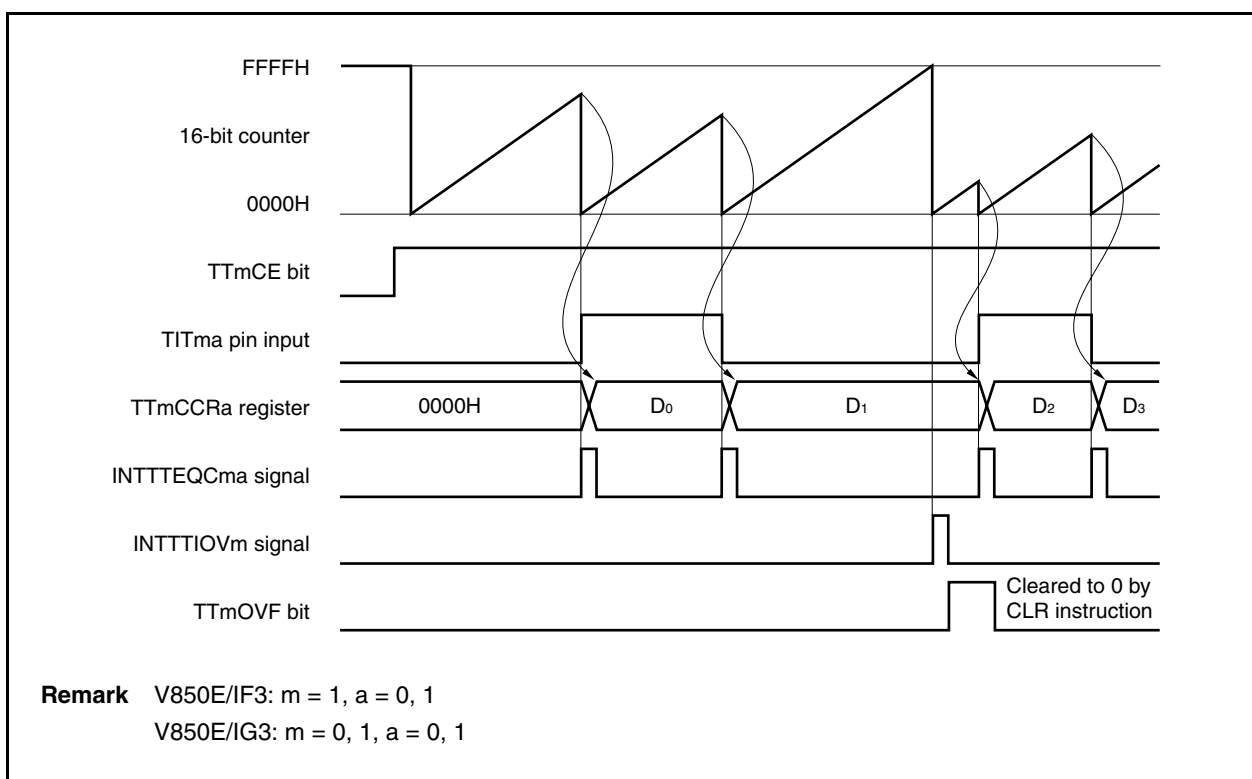




Figure 8-40. Basic Timing in Pulse Width Measurement Mode



When the TmCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TITma pin is later detected, the count value of the 16-bit counter is stored in the TmCCRa register, the 16-bit counter is cleared to 0000H, and a capture interrupt request signal (INTTTEQCma) 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 TITma pin even when the 16-bit counter counted up to FFFFH, an overflow interrupt request signal (INTTTIOVm) is generated at the next count clock, and the counter is cleared to 0000H and continues counting. At this time, the overflow flag (TmOPT0.TmOVf 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.

$$\text{Pulse width} = (10000\text{H} \times \text{TmOVf bit set (1) count} + \text{Captured value}) \times \text{Count clock cycle}$$

**Remark** V850E/IF3: m = 1, a = 0, 1  
V850E/IG3: m = 0, 1, a = 0, 1

Figure 8-41. Register Setting in Pulse Width Measurement Mode (1/2)

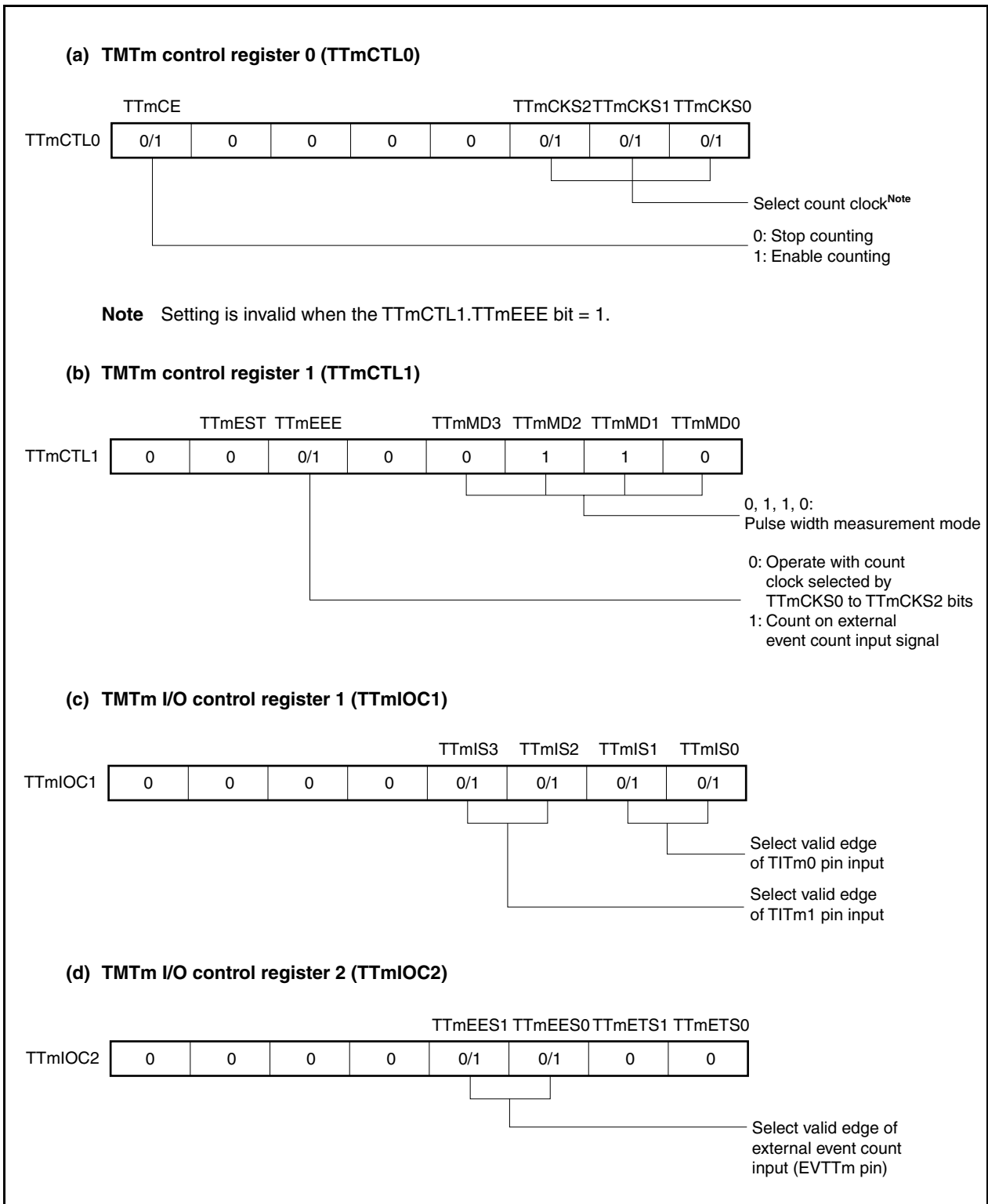
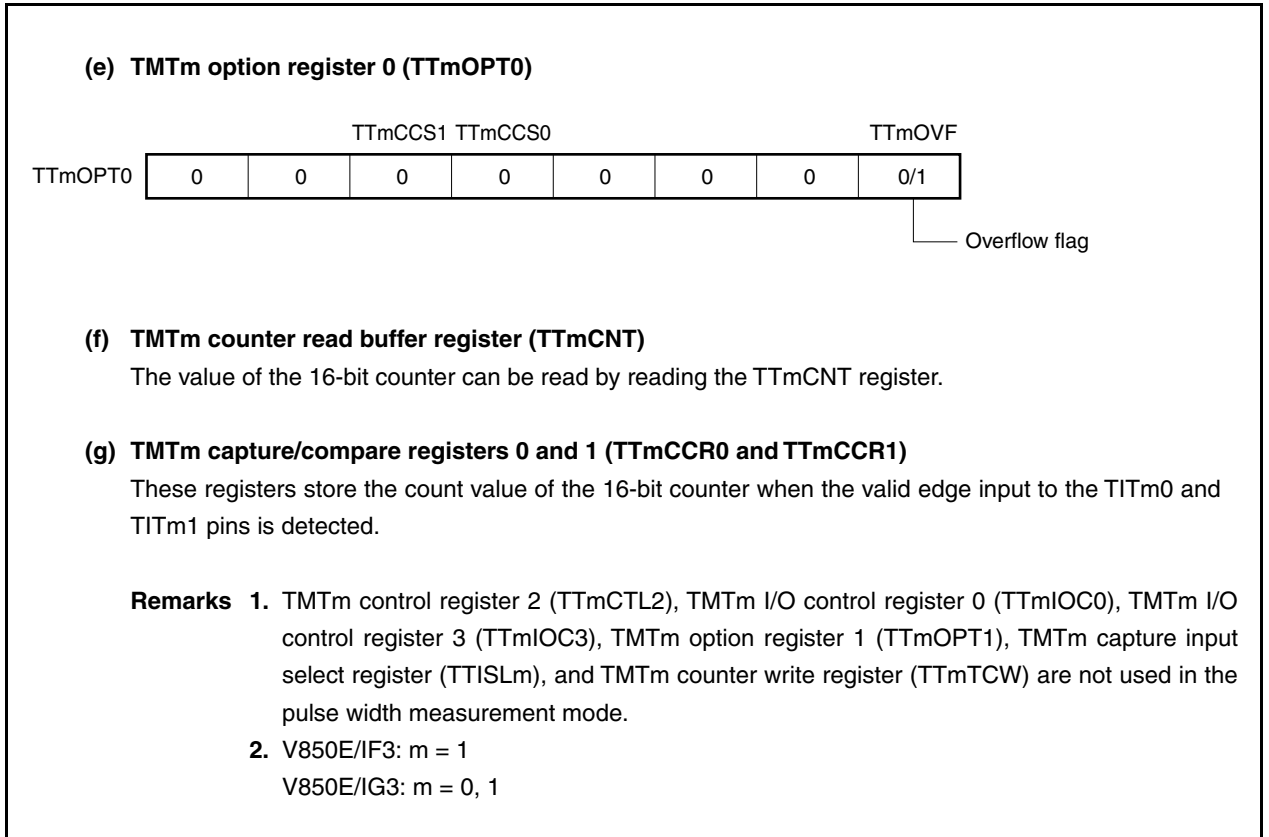
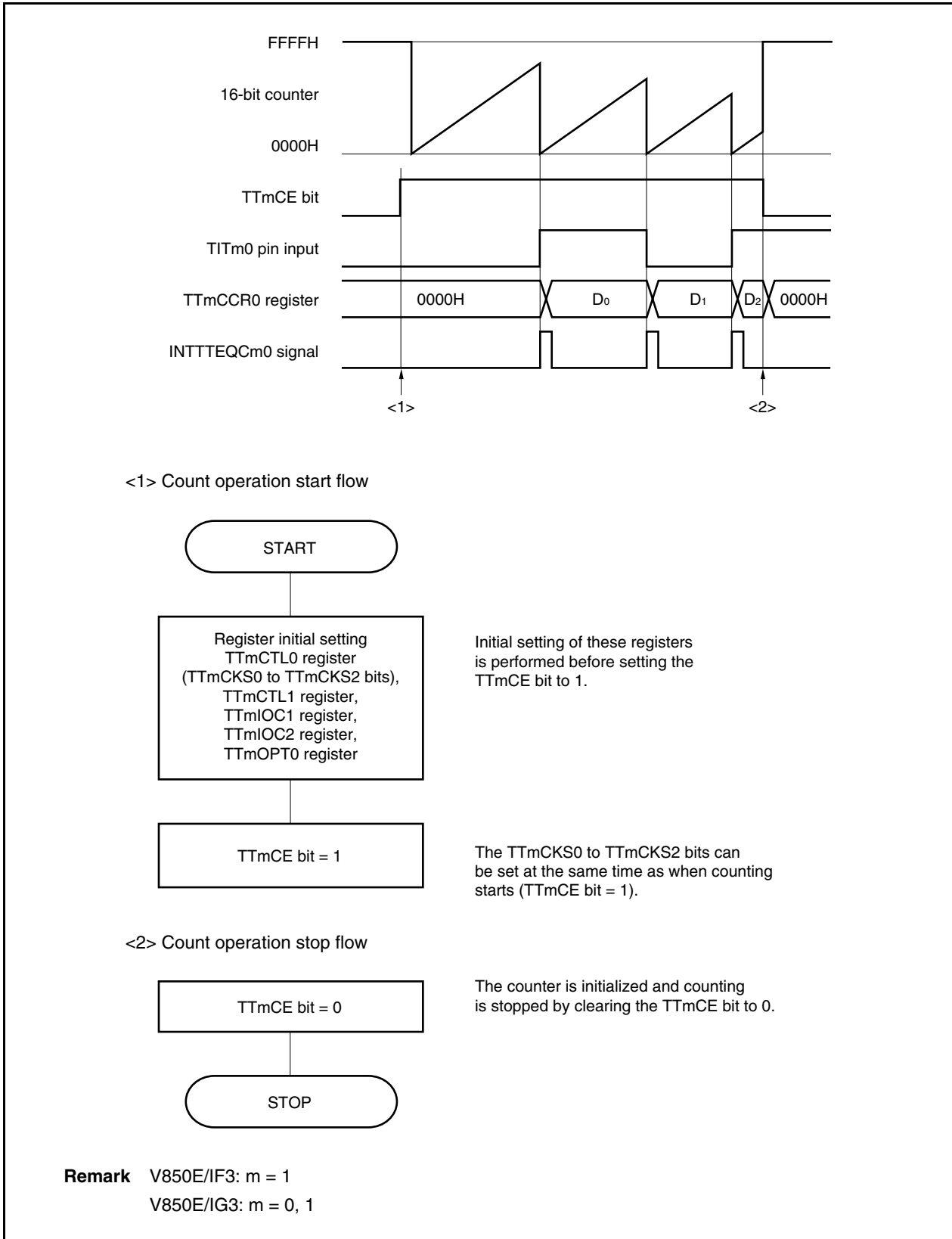


Figure 8-41. Register Setting in Pulse Width Measurement Mode (2/2)



(1) Operation flow in pulse width measurement mode

Figure 8-42. 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 TTmOVF bit to 0 with the CLR instruction after reading the TTmOVF bit when it is 1 and by writing 8-bit data (bit 0 is 0) to the TTmOPT0 register after reading the TTmOVF bit when it is 1.

**8.6.8 Triangular-wave PWM output mode (TTmMD3 to TTmMD0 bits = 0111)**

In the triangular-wave PWM output mode, a triangular-wave PWM waveform is output from the TOTm1 pin when the TTmCTL0.TTmCE bit is set to 1.

An inverted PWM waveform is output from the TOTm0 pin when the count value of the 16-bit counter matches the value of the CCR0 buffer register and when the 16-bit counter is set to 0000H.

**Figure 8-43. Configuration in Triangular-Wave PWM Output Mode**

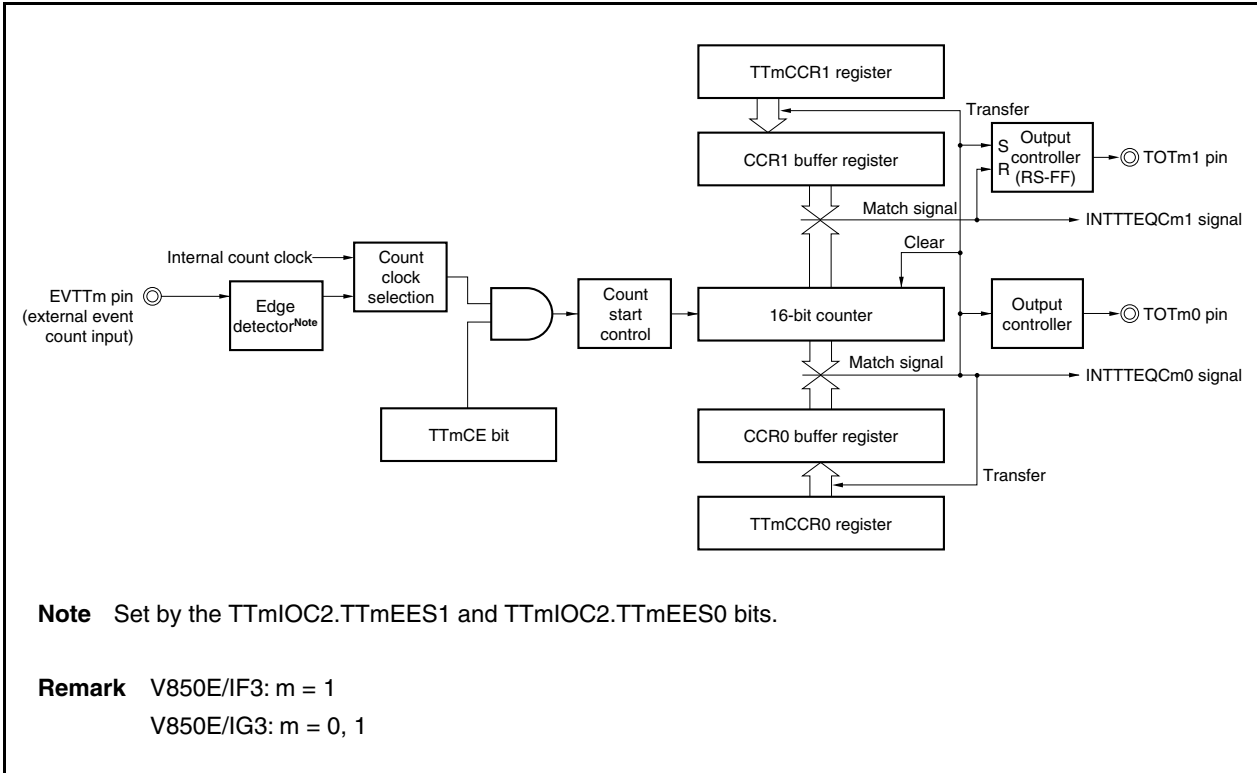
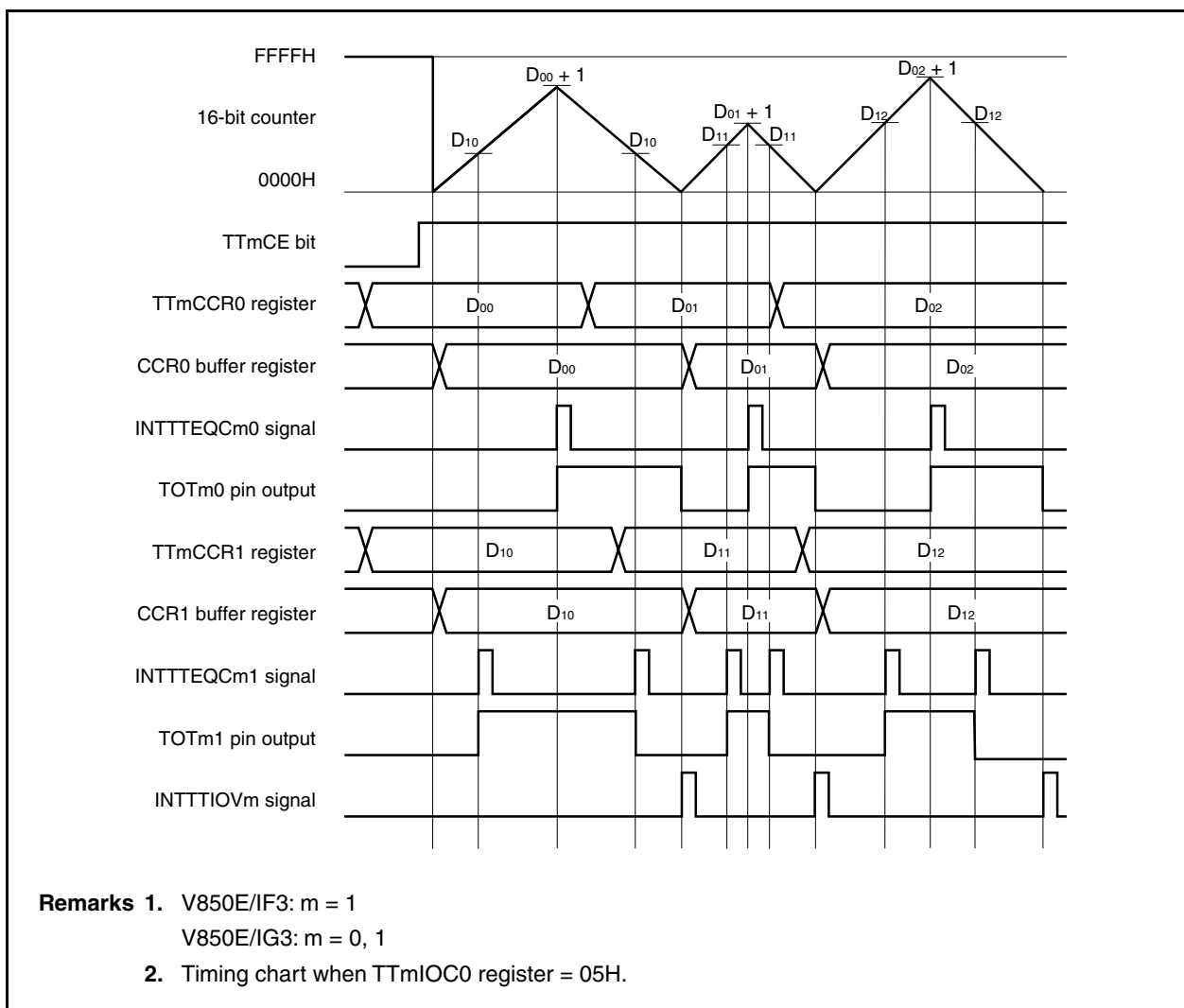


Figure 8-44. Basic Timing in Triangular-Wave PWM Output Mode

<R>



The 16-bit counter is cleared from FFFFH and 0000H and starts counting when the TTMCE bit is set to 1. The triangular PWM waveform is output from the TOTm1 pin.

In the triangular-wave PWM output mode, the counter counts up or down. When the 16-bit counter reaches 0000H while it is counting down, an overflow interrupt request signal (INTTTIOVm) is generated. At this time, the TTMOPT0.TTMOVF bit is not set to 1. If the count value of the 16-bit counter matches the value of the CCR0 buffer register while the counter is counting up, a compare match interrupt request signal (INTTTEQCm0) is generated.

The counting direction is changed from up to down when the value of the 16-bit counter matches that of the CCR0 buffer register, and from down to up when the counter is cleared to 0000H.

The PWM waveform can be changed by rewriting the TTMCCRa register during operation. To change the PWM waveform during operation, write the TTMCCR1 register last.

The cycle of the triangular PWM waveform is set by the TTMCCR0 register and its duty factor is set by the TTMCCR1 register. Set a value to the TTMCCR0 register in a range of  $0 \leq \text{TTMCCR0} \leq \text{FFFEH}$ . The rewritten value is reflected when the 16-bit counter reaches 0000H while it is counting down.

Even when changing only the cycle of the PWM waveform, first set a period to the TTMCCR0 register, and then write the same value (value same as that set to the TTMCCR1 register) to the TTMCCR1 register.

To transfer data from the TTMCCRa register to the CCRa buffer register, the data must be written to the TTMCCR1 register ( $a = 0, 1$ ).

**(1) PWM output of 0%/100%**

In the triangular-wave PWM output mode, 0% waveform output and 100% waveform output are available for PWM output.

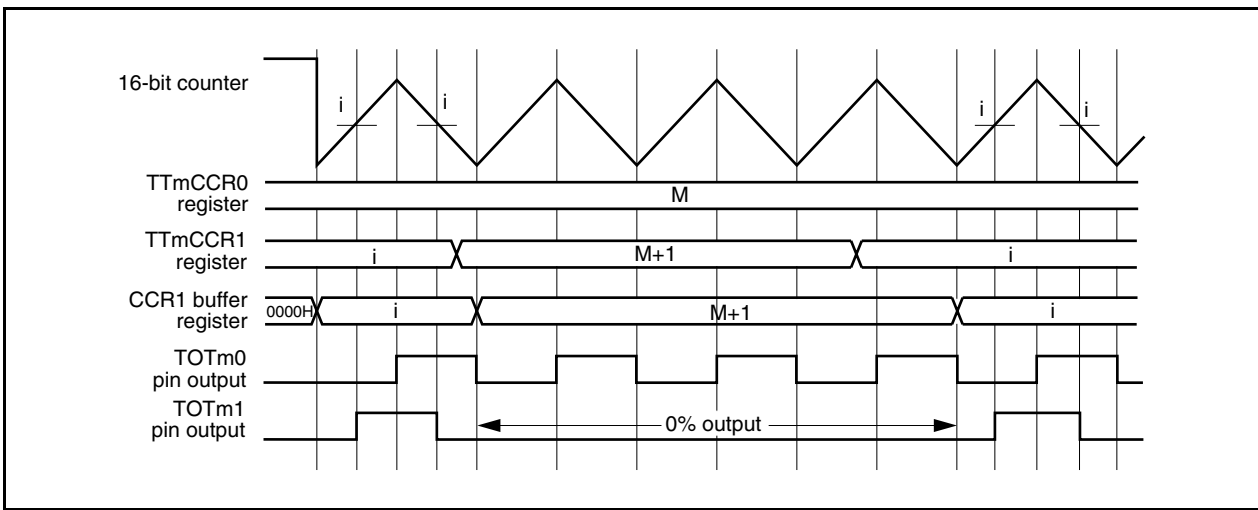
The 0% waveform is output by setting the TTmCCR1 register to “M + 1” when the TTmCCR0 register = M.

The 100% waveform is output by setting the TTmCCR1 register to “0000H”.

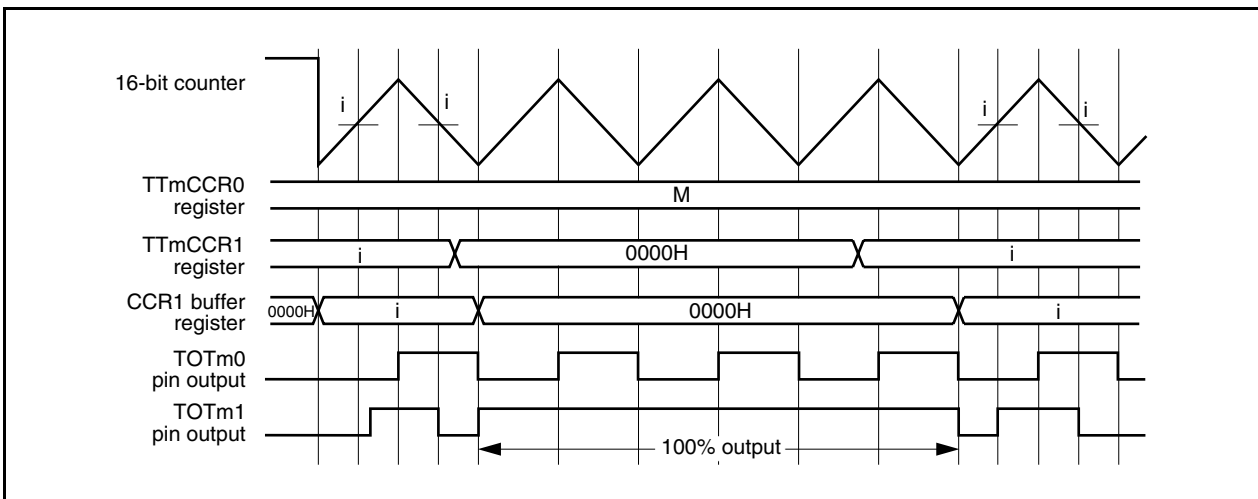
The output level of TOTm0 and TOTm1 can be set in the TTmILOC0 register.

**Remark** V850E/IF3: m = 1  
 V850E/IG3: m = 0, 1

**Figure 8-45. 0% PWM Output Waveform (TTmILOC0 Register = 05H)**



**Figure 8-46. 100% PWM Output Waveform (TTmILOC0 Register = 05H)**





**8.6.9 Encoder count function**

The encoder count function includes an encoder compare mode (see **8.6.10 Encoder compare mode (TTmMD3 to TTmMD0 bits = 1000)**).

| Mode                 | TTmCCR0 Register | TTmCCR1 Register |
|----------------------|------------------|------------------|
| Encoder compare mode | Compare only     | Compare only     |

**(1) Count-up/-down control**

Counting up or down by the 16-bit counter is controlled by the phase of input encoder signals (TENCm0 and TENCm1) and setting of the TTmCTL2.TTmUDS1 and TTmCTL2.TTmUDS0 bits.

When the encoder count function is used, the internal count clock and external event count input (EVTTm) cannot be used. Set the TTmCTL0.TTmCKS2 to TTmCTL0.TTmCKS0 bits to 000 and the TTmCTL1.TTmEEE bit to 0.

**(2) Setting initial value of 16-bit counter**

The initial count value set to the TTmTCW register when the TTmCTL2.TTmECC bit = 0 is transferred to the 16-bit counter immediately after the counter starts its operation (TTmCTL0.TTmCE bit = 0 → 1), and the counter starts the operation after it detects the valid edge of the encoder input signal (TENCm0 or TENCm1).

**(3) Basic operation**

The TTmCCRa register generates a compare match interrupt request signal (INTTTEQCma) when the count value of the 16-bit counter matches the value of the CCRa buffer register.

**(4) Clear operation**

The 16-bit counter is cleared when the following conditions are satisfied in the encoder compare mode.

- When the value of the 16-bit counter matches the value of the compare register (the TTmCTL2.TTmECM1 and TTmCTL2.TTmECM0 bits are set)
- When the edge of the encoder clear input signal (TECRm) is detected and cleared (the TTmECS1 and TTmECS0 bits are set when the TTmIOC3.TTmSCE bit = 0)
- When the clear level condition of the TENCm0, TENCm1, and TECRm pins is detected (the TTmZCL, TTmBCL, and TTmACL bits are set when the TTmSCE bit = 1)

**Remark** V850E/IF3: m = 1, a = 0, 1  
 V850E/IG3: m = 0, 1, a = 0, 1

**(5) Controlling bits of TTmCTL2 register**

The setting of the TTmCTL2 register in the encoder compare mode is shown below.

**Table 8-10. Setting of TTmCTL2 Register**

| Mode                 | TTmUDS1, TTmUDS0 Bits (<1>)      | TTmECM1 Bit (<2>) | TTmECM0 Bit (<2>) | TTmLDE Bit (<3>) | Counter Clear (Target Compare Register) | Transfer to Counter      |
|----------------------|----------------------------------|-------------------|-------------------|------------------|---|--------------------------|
| Encoder compare mode | Can be set to 00, 01, 10, or 11. | 0                 | 0                 | 0                | -                                       | -                        |
|                      |                                  |                   |                   | 1                |   | Possible                 |
|                      |                                  | 1                 | 0                 | 0                | TTmCCR0                                 | -                        |
|                      |                                  |                   |                   | 1                |   | Possible <sup>Note</sup> |
|                      |                                  | 1                 | 0                 | Invalid          | TTmCCR1                                 | -                        |
|                      |                                  |                   |                   | 1                |   | Invalid                  |

**Note** The counter can operate in a range from 0000H to the set value of the TTmCCR0 register.

**Remark** V850E/IF3: m = 1  
V850E/IG3: m = 0, 1

**(a) Outline of each bit**

- <1> The TTmUDS1 and TTmUDS0 bits identify the counting direction (up or down) of the 16-bit counter by the phase input from the encoder input pin (TENCm0 or TENCm1).
- <2> The TTmECM1 and TTmECM0 bits control clearing of the 16-bit counter when its count value matches the value of the CCR0 or CCR1 buffer register.
- <3> The TTmLDE bit controls a function to transfer the set value of the TTmCCR0 register to the 16-bit counter when the counter underflows. The TTmLDE bit is valid only when the TTmECM1 and TTmECM0 bits are 00 or 01. It is invalid when these bits are set to any other value.

**(b) Detailed explanation of each bit**

<1> TTmUDS1 and TTmUDS0 bits: Count-up/-down selection

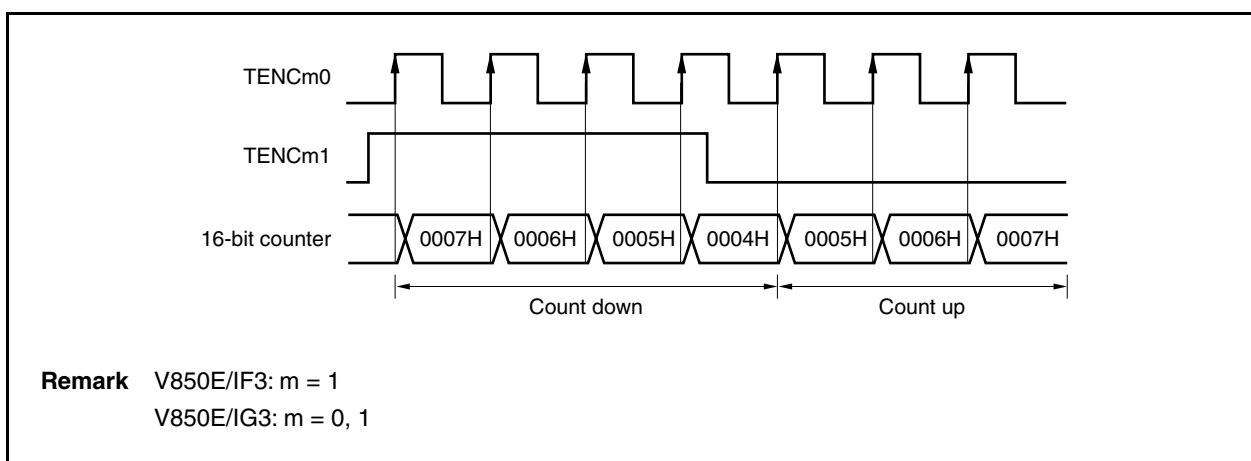
Whether the 16-bit counter is counting up or down is identified by the phase input from the TENCm0 or TENCm1 pin and depending on the setting of the TTmUDS1 and TTmUDS0 bits. These bits are valid only in the encoder compare mode.

- When TTmUDS1 and TTmUDS0 bits = 00

| TENCm0 Pin   | TENCm1 Pin | Count Operation |
|--------------|------------|-----------------|
| Rising edge  | High level | Count down      |
| Falling edge |            |                 |
| Both edges   |            |                 |
| Rising edge  | Low level  | Count up        |
| Falling edge |            |                 |
| Both edges   |            |                 |

**Remark** Detecting the edge of the TENCm0 pin is specified by the TTmIOC3.TTmEIS1 and TTmEIS0 bits.

**Figure 8-47. Operation Example (When Valid Edge of TENCm0 Pin Is Specified to Be Rising Edge and No Edge Is Specified as Valid Edge of TENCm1 Pin)**

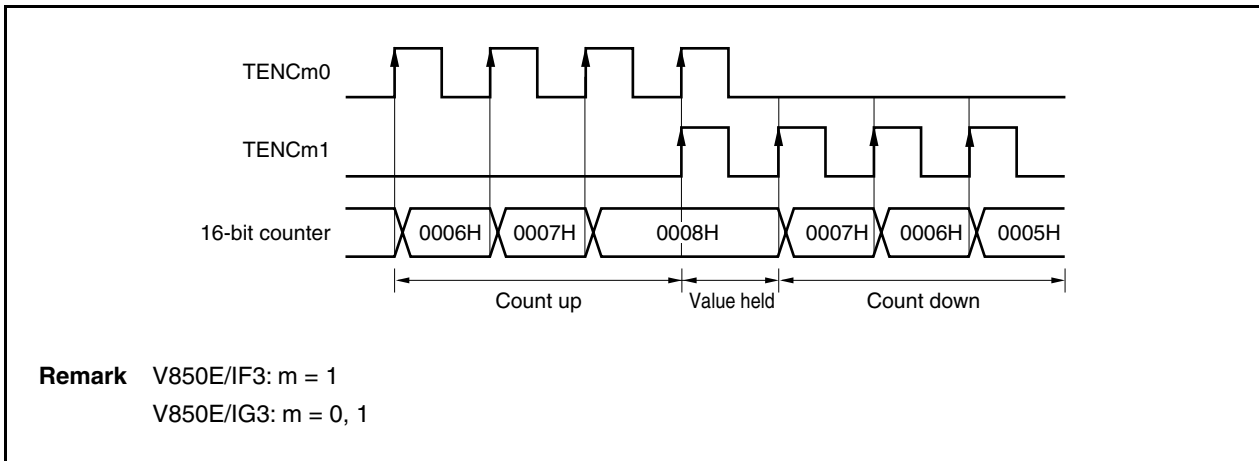


- When TTmUDS1 and TTmUDS0 bits = 01

| TENCM0 Pin                                   | TENCM1 Pin   | Count Operation  |
|--|--------------|--|
| Low level                                    | Rising edge  | Count down   |
|  | Falling edge |  |
|  | Both edges   |  |
| High level                                   | Rising edge  |  |
|  | Falling edge |  |
|  | Both edges   |  |
| Rising edge                                  | High level   | Count up   |
| Falling edge                                 |              |  |
| Both edges                                   |              |  |
| Rising edge                                  | Low level    |  |
| Falling edge                                 |              |  |
| Both edges                                   |              |  |
| Simultaneous input to TENCM0 and TENCM1 pins |              | Counter does not perform count operation but holds value immediately before. |

**Remark** Detecting the edge of the TENCM0 pin is specified by the TTmIOC3.TTmEIS1 and TTmIOC3.TTmEIS0 bits.

**Figure 8-48. Operation Example (When Rising Edge Is Specified as Valid Edge of TENCM0 and TENCM1 Pins)**



- When TTmUDS1 and TTmUDS0 bits = 10

| TENCM0 Pin   | TENCM1 Pin   | Count Operation  |
|--------------|--------------|--|
| Low level    | Falling edge | Counter does not perform count operation but holds value immediately before. |
| Rising edge  | Low level    | Count down   |
| High level   | Rising edge  | Counter does not perform count operation but holds value immediately before. |
| Falling edge | High level   |  |
| Rising edge  | High level   |  |
| High level   | Falling edge | Count up   |
| Falling edge | Low level    | Count up   |
| Low level    | Rising edge  | Counter does not perform count operation but holds value immediately before. |
| Rising edge  | Rising edge  |  |
| Falling edge | Rising edge  |  |
| Rising edge  | Falling edge | Count down   |
| Falling edge | Falling edge | Count up   |

**Caution** Specification of the valid edge of the TENCM0 and TENCM1 pins is invalid.

Figure 8-49. Operation Example (Count Operation When Valid Edges of TENCM0 and TENCM1 Pins Do Not Overlap)

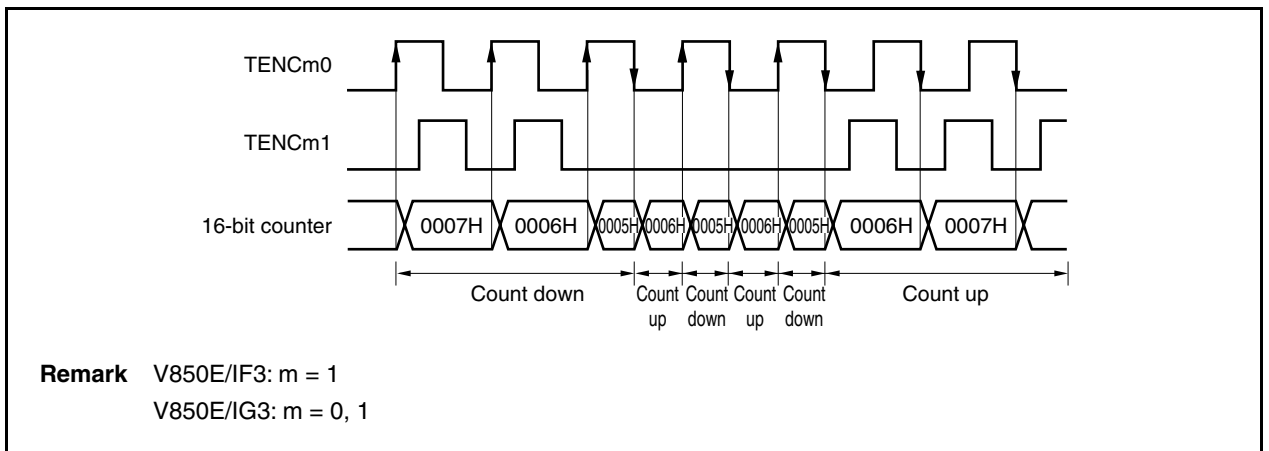
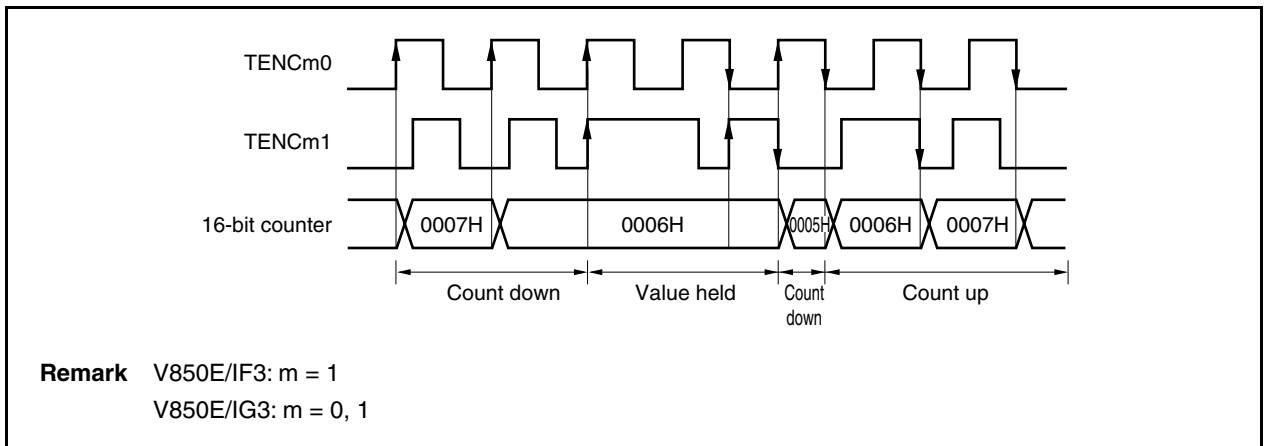


Figure 8-50. Operation Example (Count Operation When Valid Edges of TENCM0 and TENCM1 Pins Overlap)



- When TTmUDS1 and TTmUDS0 bits = 11

| TENCM0 Pin                                   | TENCM1 Pin   | Count Operation  |
|--|--------------|--|
| Low level                                    | Falling edge | Count down   |
| Rising edge                                  | Low level    |  |
| High level                                   | Rising edge  |  |
| Falling edge                                 | High level   |  |
| Rising edge                                  | High level   | Count up   |
| High level                                   | Falling edge |  |
| Falling edge                                 | Low level    |  |
| Low level                                    | Rising edge  |  |
| Simultaneous input to TENCM0 and TENCM1 pins |              | Counter does not perform count operation but holds value immediately before. |

**Caution** Specification of the valid edge of the TENCM0 and TENCM1 pins is invalid.

Figure 8-51. Operation Example (Count Operation When Valid Edges of TENCM0 and TENCM1 Pins Do Not Overlap)

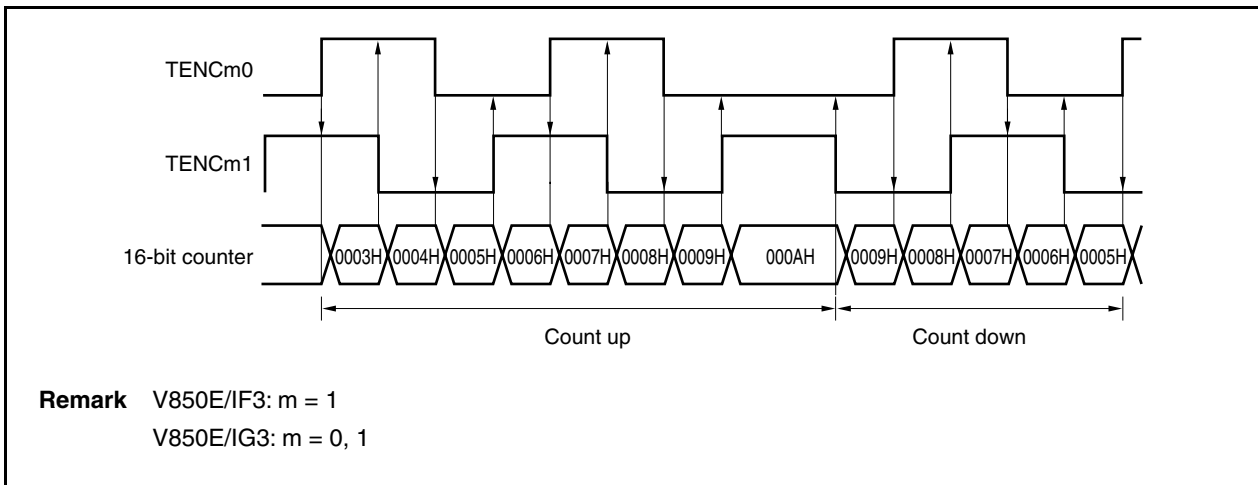
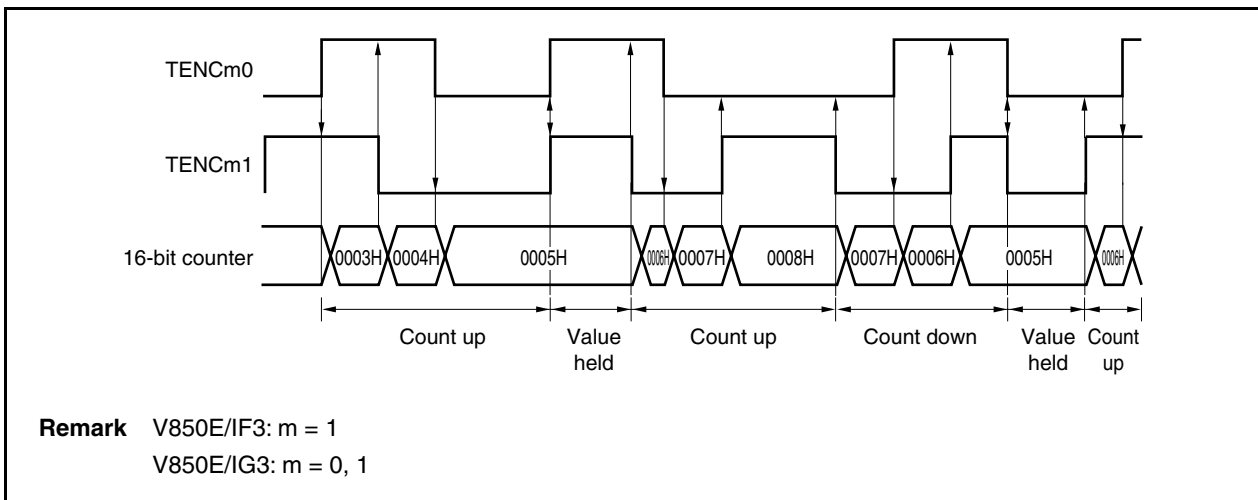


Figure 8-52. Operation Example (Count Operation When Valid Edges of TENCM0 and TENCM1 Pins Overlap)



<2> TTmECM1 and TTmECM0 bits: Timer/counter clear function upon match of the compare register  
 The 16-bit counter performs its count operation in accordance with the set value of the TTmECM1 and TTmECM0 bits when the count value of the counter matches the value of the CCRa buffer register.

- When TTmECM1 and TTmECM0 bits = 00  
 The 16-bit counter is not cleared when its count value matches the value of the CCRa buffer register.
- When TTmECM1 and TTmECM0 bits = 01  
 The 16-bit counter performs a count operation under the following condition when its count value matches the value of the CCR0 buffer register.

| Next Count Operation | Description                                    |
|----------------------|--|
| Count up             | 16-bit counter is cleared to 0000H.            |
| Count down           | Count value of 16-bit counter is counted down. |

- When TTmECM1 and TTmECM0 bits = 10  
 The 16-bit counter performs a count operation under the following condition when its count value matches the value of the CCR1 buffer register.

| Next Count Operation | Description                                  |
|----------------------|--|
| Count up             | Count value of 16-bit counter is counted up. |
| Count down           | 16-bit counter is cleared to 0000H.          |

- When TTmECM1 and TTmECM0 bits = 11  
 The 16-bit counter performs a count operation under the following condition when its count value matches the value of the CCR0 buffer register.

| Next Count Operation | Description                                    |
|----------------------|--|
| Count up             | 16-bit counter is cleared to 0000H.            |
| Count down           | Count value of 16-bit counter is counted down. |

The 16-bit counter performs a count operation under the following condition when its count value matches the value of the CCR1 buffer register.

| Next Count Operation | Description                                  |
|----------------------|--|
| Count up             | Count value of 16-bit counter is counted up. |
| Count down           | 16-bit counter is cleared to 0000H.          |

<3> TTmLDE bit: Transfer function of the set value of the TTmCCR0 register to the 16-bit counter when the counter underflows

When the TTmLDE bit = 1, the set value of the TTmCCR0 register can be transferred to the 16-bit counter when the counter underflows.

The TTmLDE bit is valid only in the encoder compare mode.

- Count operation in range from 0000H to set value of the TTmCCR0 register

If the 16-bit counter performs a count operation when the TTmLDE bit = 1 and TTmECM1 and TTmECM0 bits = 01, and when the count value of the counter matches the set value of the CCR0 buffer register when the TTmECM0 bit = 1, the 16-bit counter is cleared to 0000H if the next count operation is counting up.

If the 16-bit counter underflows when the TTmLDE bit = 1, the set value of the TTmCCR0 register is transferred to the counter.

Therefore, the counter can operate in a range from 0000H to the set value of the TTmCCR0 register in which the upper-limit count value is the set value of the TTmCCR0 register and the lower-limit value is 0000H.

**Figure 8-53. Operation Example (Count Operation in Range from 0000H to Set Value of TTmCCR0 Register)**

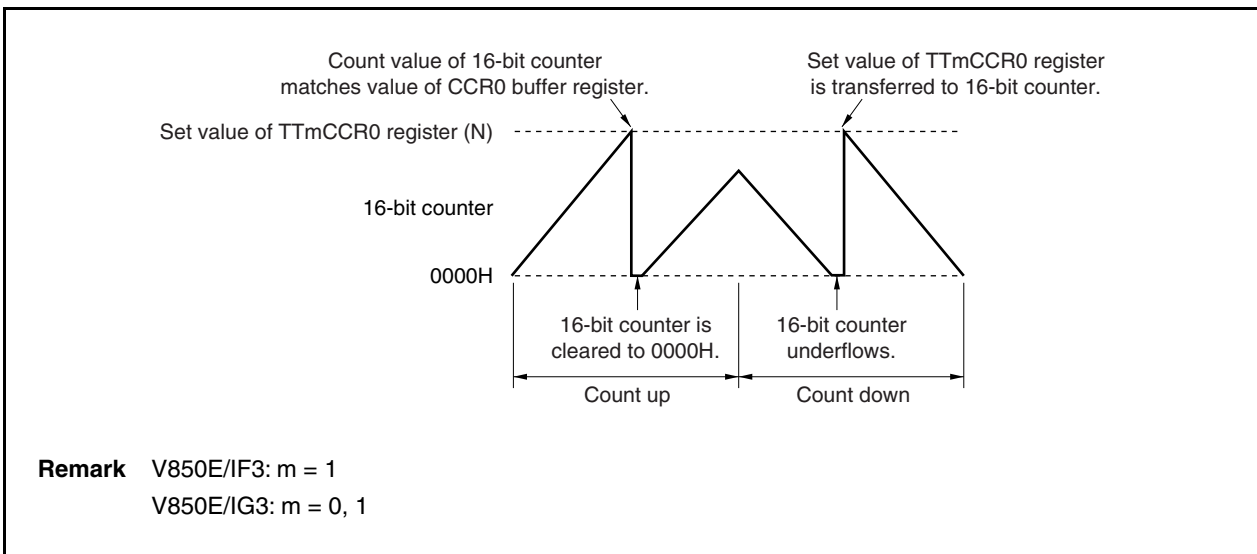
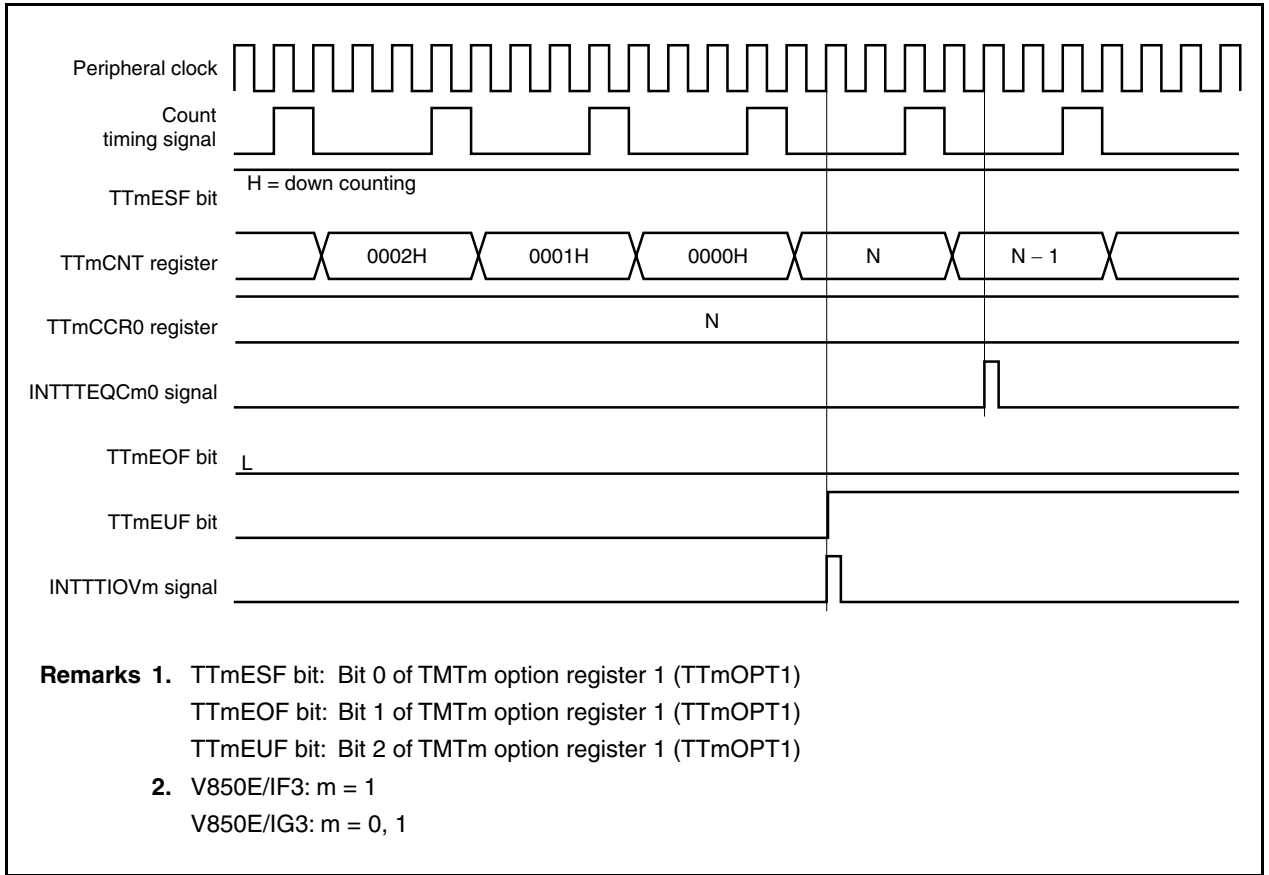




Figure 8-54. Operation Timing (Count Operation in Range from 0000H to Set Value of TTmCCR0 Register)



**(6) Clearing counter to 0000H by encoder clear signal (TECRm pin)**

The 16-bit counter can be cleared to 0000H by the input signal of the TECRm pin in two ways which are selected by the TTmIOC3.TTmSCE bit. The TTmSCE bit also controls, depending its setting, the TTmIOC3.TTmZCL, TTmIOC3.TTmBCL, TTmIOC3.TTmACL, TTmIOC3.TTmESC1, and TTmIOC3.TTmECS0 bits.

The counter can be cleared by the methods described below only in the encoder compare mode.

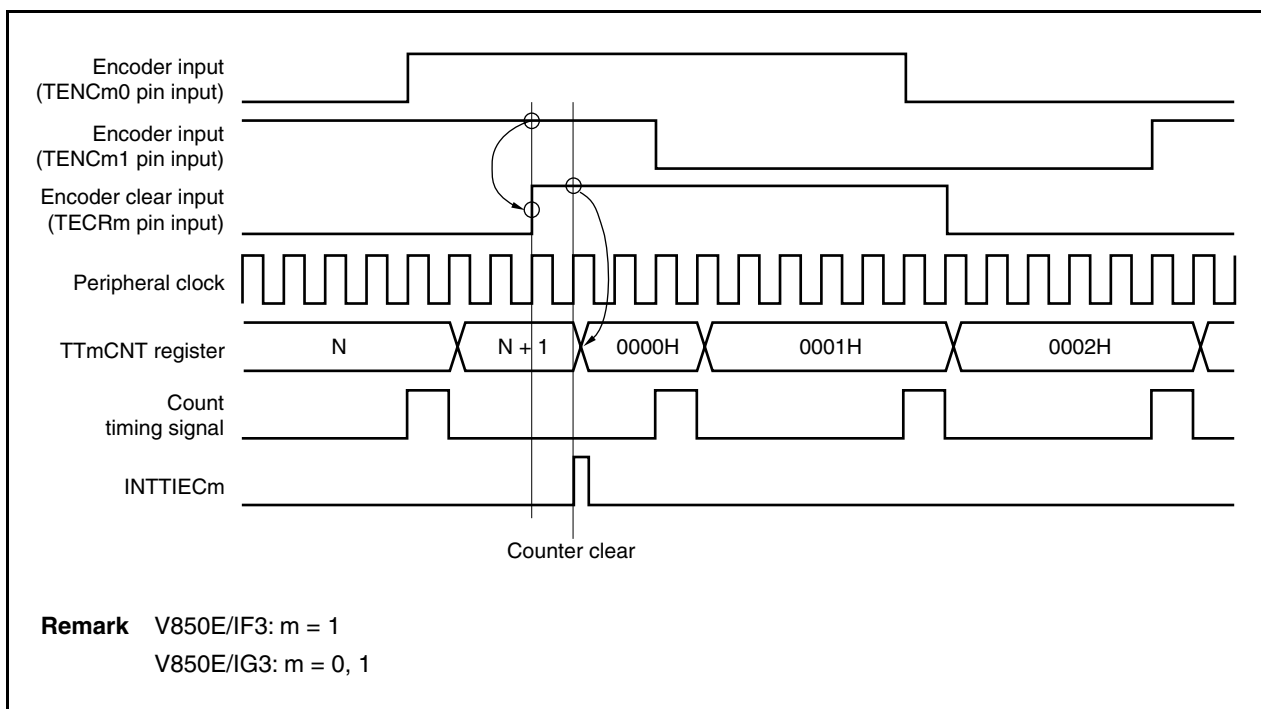
**Table 8-11. Relationship Between TTmSCE Bit and TTmZCL, TTmBCL, TTmACL, TTmECS1, and TTmECS0 Bits**

| Clearing Method | TTmSCE Bit | TTmZCL Bit | TTmBCL Bit | TTmACL Bit | TTmECS1, TTmECS0 Bits |
|-----------------|------------|------------|------------|------------|-----------------------|
| <1>             | 0          | Invalid    | Invalid    | Invalid    | Valid                 |
| <2>             | 1          | Valid      | Valid      | Valid      | Invalid               |

**(a) Clearing method <1>: By detecting edge of encoder clear signal (TECRm pin) (TTmSCE bit = 0)**

When the TTmSCE bit = 0, the 16-bit counter is cleared to 0000H in synchronization with the peripheral clock if the valid edge of the TECRm pin specified by the TTmECS1 and TTmECS0 bits is detected. At this time, an encoder clear interrupt request signal (INTTIECm) is generated. When the TTmSCE bit = 0, setting of the TTmZCL, TTmBCL, and TTmACL bits is invalid.

**Figure 8-55. Operation Example (When TTmSCE Bit = 0, TTmECS1 and TTmECS0 Bits = 01, and TTmUDS1 and TTmUDS0 Bits = 11)**



**(b) Clearing method <2>: By detecting clear level condition of the TENCm0, TENCm1, and TECRm pins (TTmSCE bit = 1)**

When the TTmSCE bit = 1, the 16-bit counter is cleared to 0000H if the clear level condition of the TECRm, TENCm0, or TENCm1 pin specified by the TTmZCL, TTmBCL, and TTmACL bits is detected. At this time, the encoder clear interrupt request signal (INTTIECm) is not generated. Setting of the TTmECS1 and TTmECS0 bits is invalid when the TTmSCE bit = 1.

**Table 8-12. 16-bit Counter Clearing Condition When TTmSCE Bit = 1**

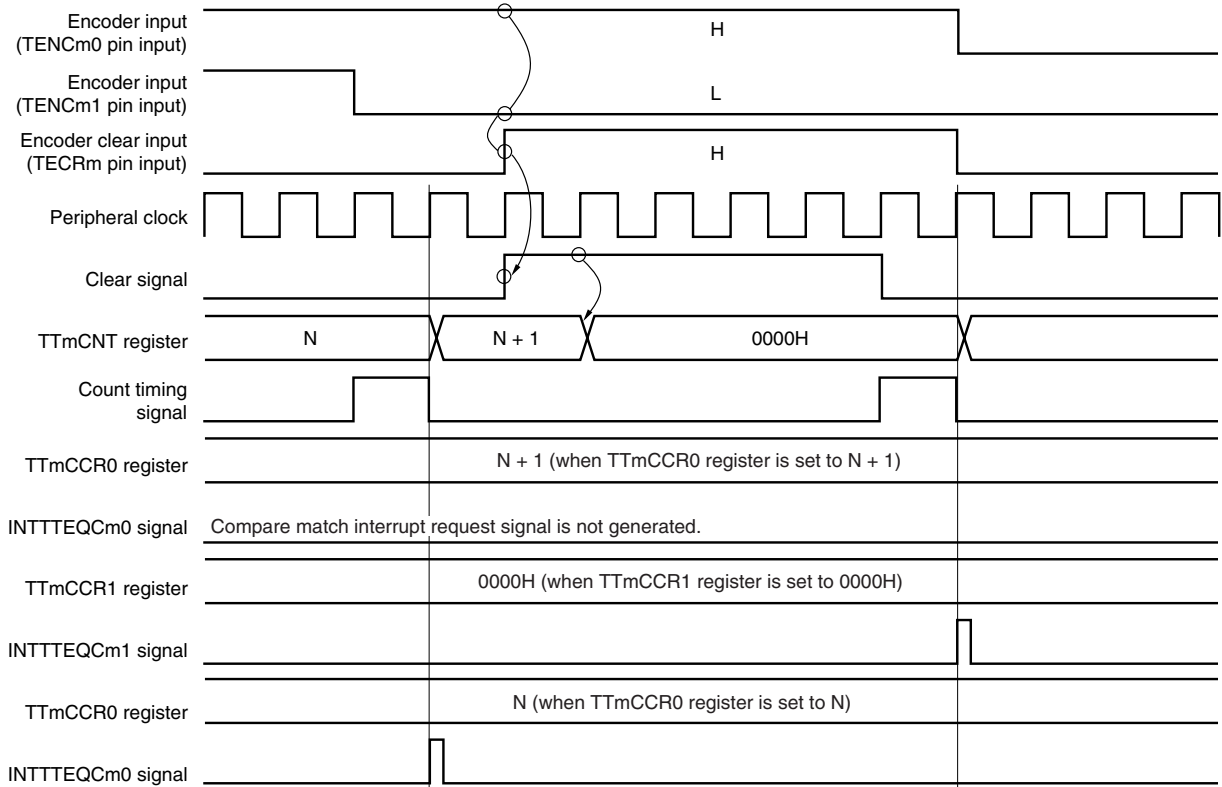
| Clear Level Condition Setting |            |            | Input Level of Encoder Pin |            |            |
|-------------------------------|------------|------------|----------------------------|------------|------------|
| TTmZCL Bit                    | TTmBCL Bit | TTmACL Bit | TECRm Pin                  | TENCm1 Pin | TENCm0 Pin |
| 0                             | 0          | 0          | L                          | L          | L          |
| 0                             | 0          | 1          | L                          | L          | H          |
| 0                             | 1          | 0          | L                          | H          | L          |
| 0                             | 1          | 1          | L                          | H          | H          |
| 1                             | 0          | 0          | H                          | L          | L          |
| 1                             | 0          | 1          | H                          | L          | H          |
| 1                             | 1          | 0          | H                          | H          | L          |
| 1                             | 1          | 1          | H                          | H          | H          |

**Caution** The 16-bit counter is cleared to 0000H when the clear level condition of the TTmZCL, TTmBCL, and TTmACL bits match the input level of the TECRm, TENCm1, or TENCm0 pin.

**Remark** V850E/IF3: m = 1  
 V850E/IG3: m = 0, 1

**Figure 8-56. Operation Example (When TTmSCE Bit = 1, TTmZCL Bit = 1, TTmBCL Bit = 0, TTmACL Bit = 1, TTmUDS1 and TTmUDS0 Bits = 11, TECRm = High Level, TENCm1 = Low Level, and TENCm0 = High Level) (1/3)**

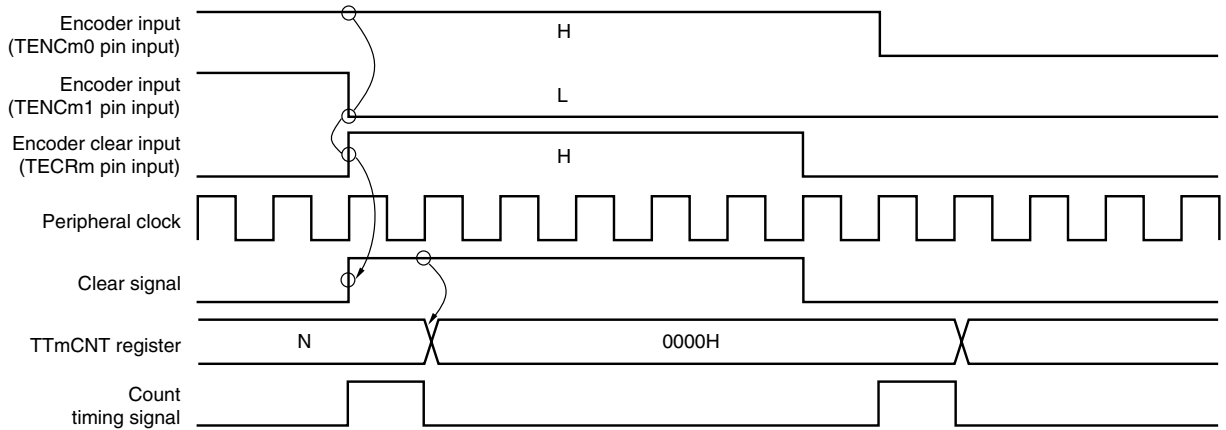
(i) If inputting the high level to the TECRm pin lags behind inputting the low level to the TENCm1 pin while the counter is counting up, the counter is cleared after it counts up.



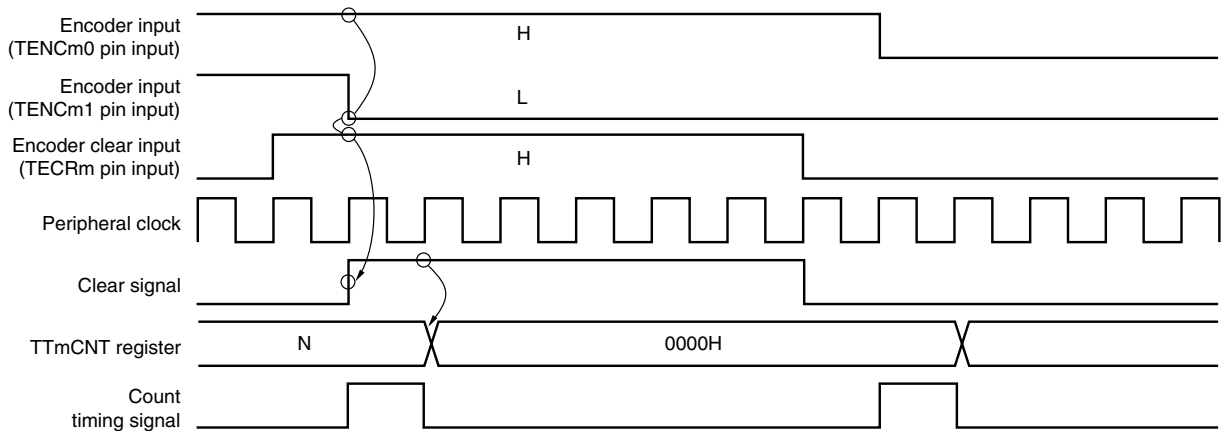
**Remark** V850E/IF3: m = 1  
V850E/IG3: m = 0, 1

**Figure 8-56. Operation Example (When TTmSCE Bit = 1, TTmZCL Bit = 1, TTmBCL Bit = 0, TTmACL Bit = 1, TTmUDS1 and TTmUDS0 Bits = 11, TECRm = High Level, TENCm1 = Low Level, and TENCm0 = High Level) (2/3)**

**(ii) If the high level is input to the TECRm pin at the same time as the low level is input to the TENCm1 pin while the counter is counting up, the counter is cleared without counting up.**

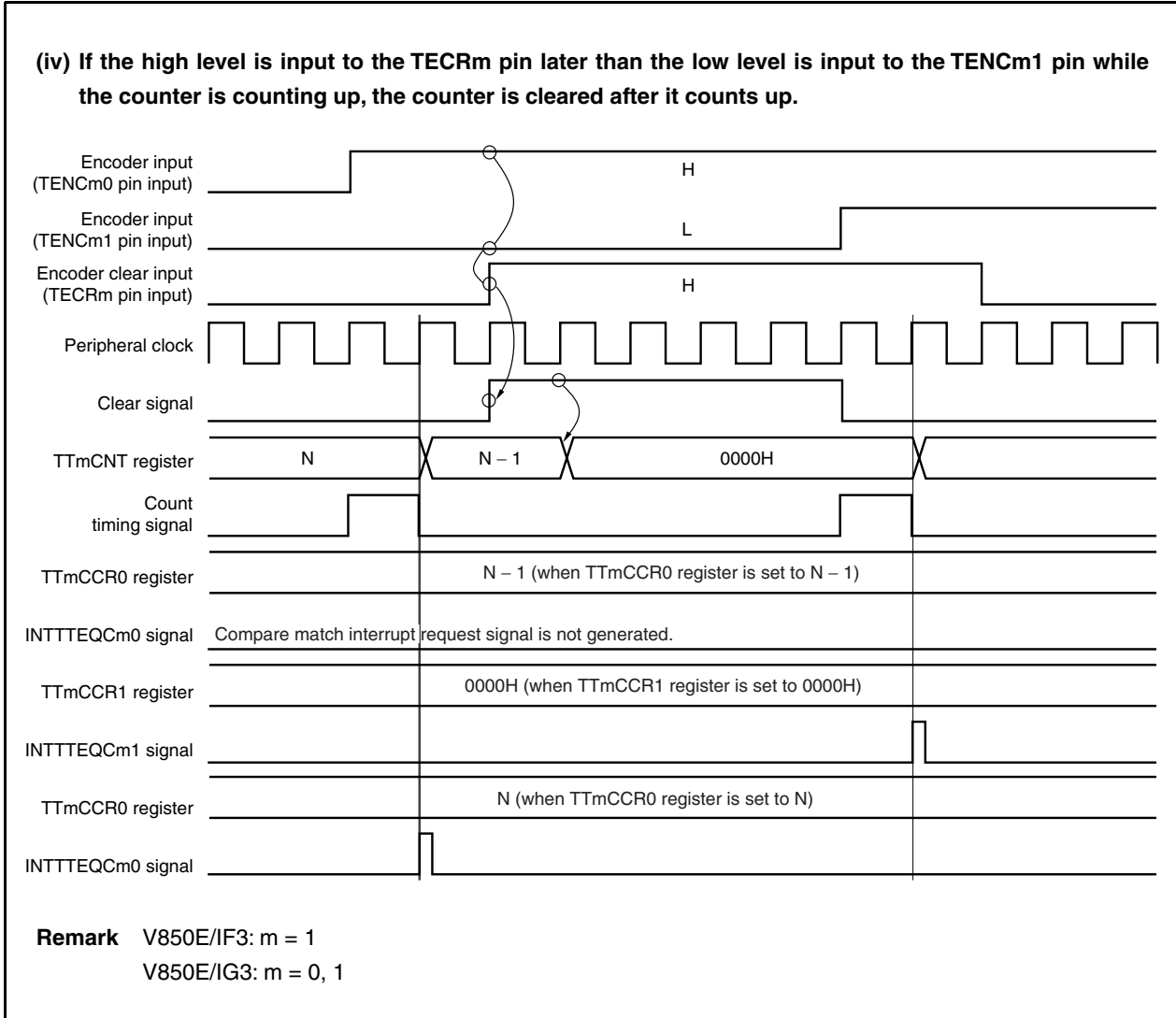


**(iii) If the high level is input to the TECRm pin earlier than the low level is input to the TENCm1 pin while the counter is counting up, the counter is cleared without counting up.**



**Remark** V850E/IF3: m = 1  
V850E/IG3: m = 0, 1

**Figure 8-56. Operation Example (When TTmSCE Bit = 1, TTmZCL Bit = 1, TTmBCL Bit = 0, TTmACL Bit = 1, TTmUDS1 and TTmUDS0 Bits = 11, TECRm = High Level, TENCm1 = Low Level, and TENCm0 = High Level) (3/3)**

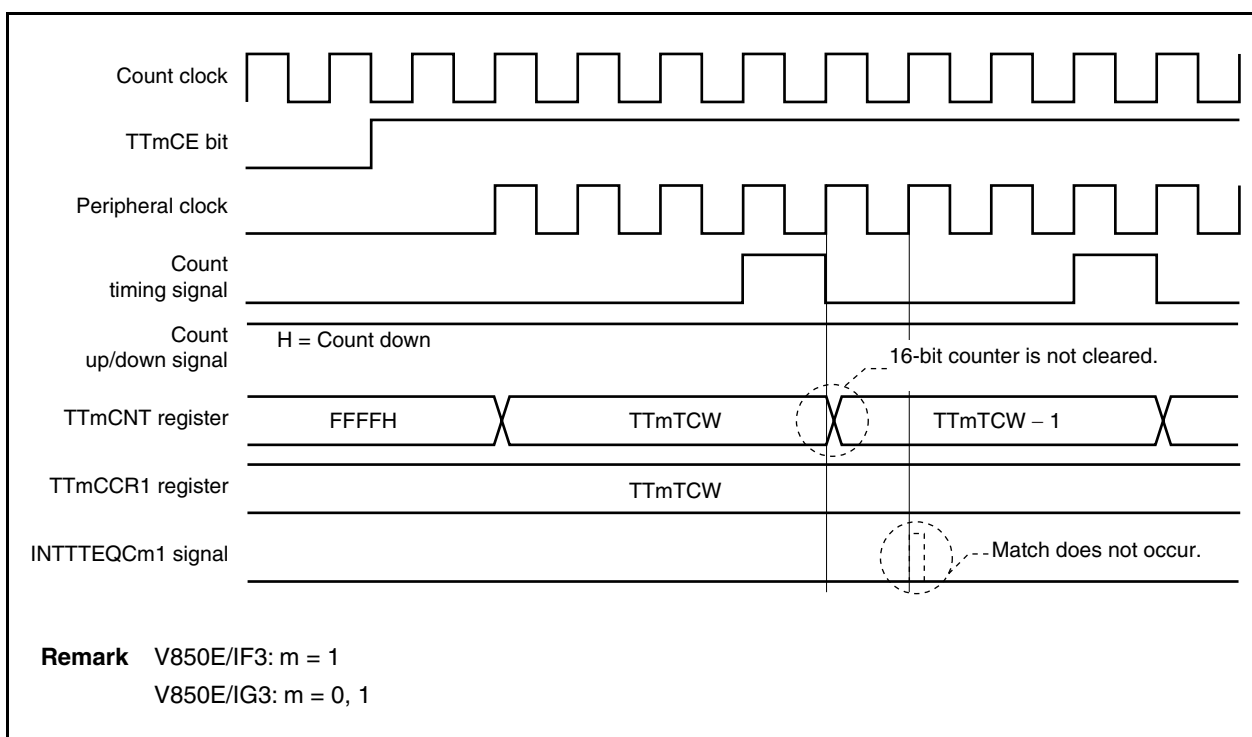


If the counter is cleared in this way, a miscount does not occur even if inputting the signal to the TECRm pin is late, because the clear level condition of the TECRm, TENCm1, and TENCm0 pins is set and the 16-bit counter is cleared to 0000H when the clear level condition is detected.

(7) Notes on using encoder count function

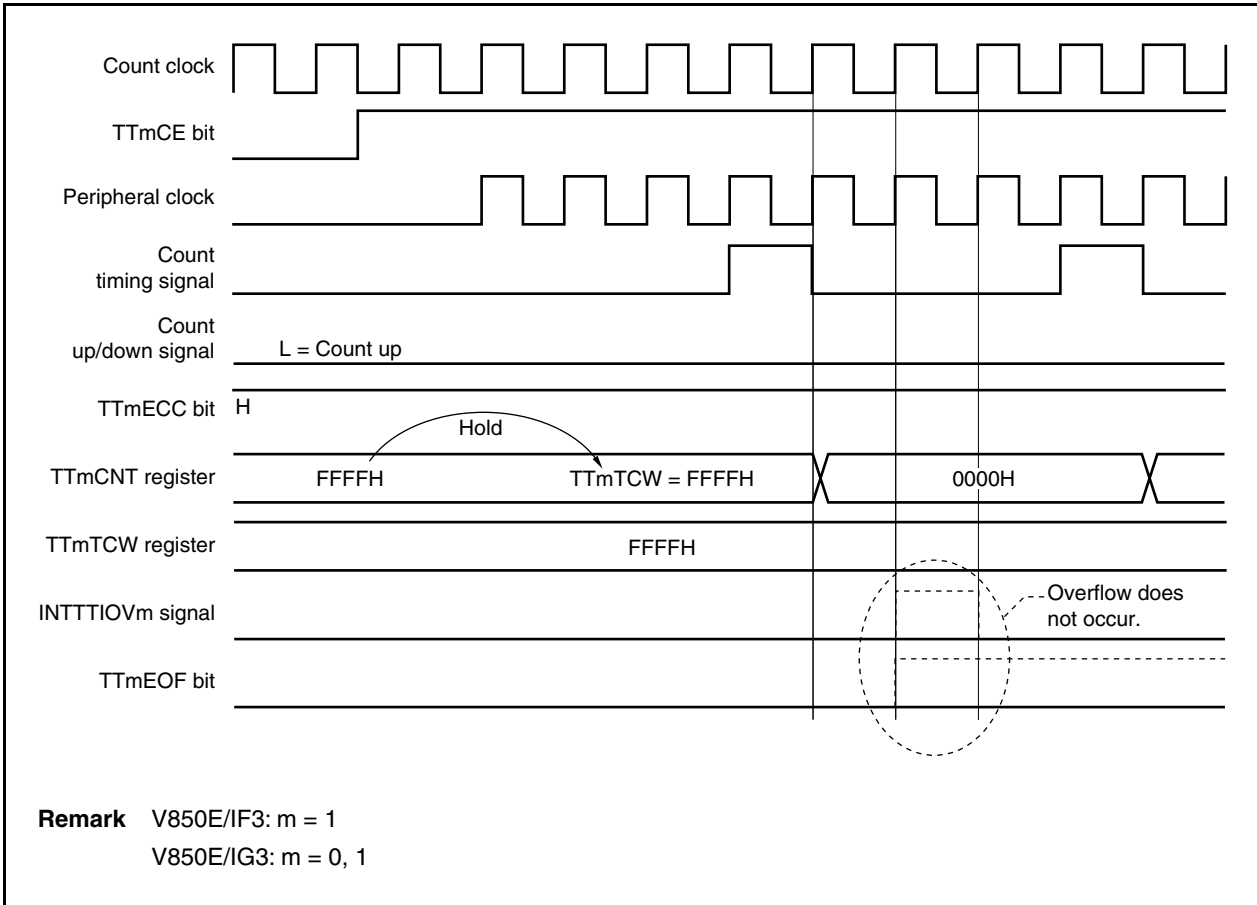
(a) If compare match interrupt is not generated immediately after operation is started

If a value which is the same as that of the TTmTCW register is set to the TTmCCR0 or TTmCCR1 register and the counter operation is started when the TTmCTL2.TTmECC bit = 0, and if the count value (TTmTCW) of the 16-bit counter matches the value of the CCRa buffer register immediately after the start of the operation, the match is masked and the compare match interrupt request signal (INTTTEQCma) is not generated (a = 0, 1). In addition, the 16-bit counter is not cleared to 0000H by setting the TTmCTL2.TTmECM1 and TTmCTL2.TTmECM0 bits.



**(b) If overflow does not occur immediately after start of operation**

If the count operation is resumed when the TTmCTL2.TTmECC bit = 1, the 16-bit counter does not overflow if its count value that has been held is FFFFH and if the next count operation is counting up. After the counter starts operating and counts up from a count value (value of TTmTCW register = FFFFH), the counter overflows from FFFFH to 0000H. However, detection of the overflow is masked, the overflow flag (TTmEOF) is not set, and the overflow interrupt request signal (INTTTIOVm) is not generated.





**8.6.10 Encoder compare mode (TTmMD3 to TTmMD0 bits = 1000)**

In the encoder compare mode, the encoder is controlled by using both the TTmCCR0 and TTmCCR1 registers as compare registers and the input pins for encoder count function (TENCm0, TENCm1, and TECRm).

In this mode, the 16-bit counter can be cleared to 0000H in three ways: when the count value of the counter matches the value of the CCRa buffer register (compare match interrupt request signal (INTTTEQCma) is generated), when the edge of the encoder clear input (TECRm pin) is detected and cleared, and when the clear level condition of TENCm0, TENCm1, and TECRm pins is detected and cleared.

When the 16-bit counter underflows, the set value of the TTmCCR0 register can be transferred to the counter.

**(1) Encoder compare mode operation flow**

**Figure 8-57. Encoder Compare Mode Operation Flow**

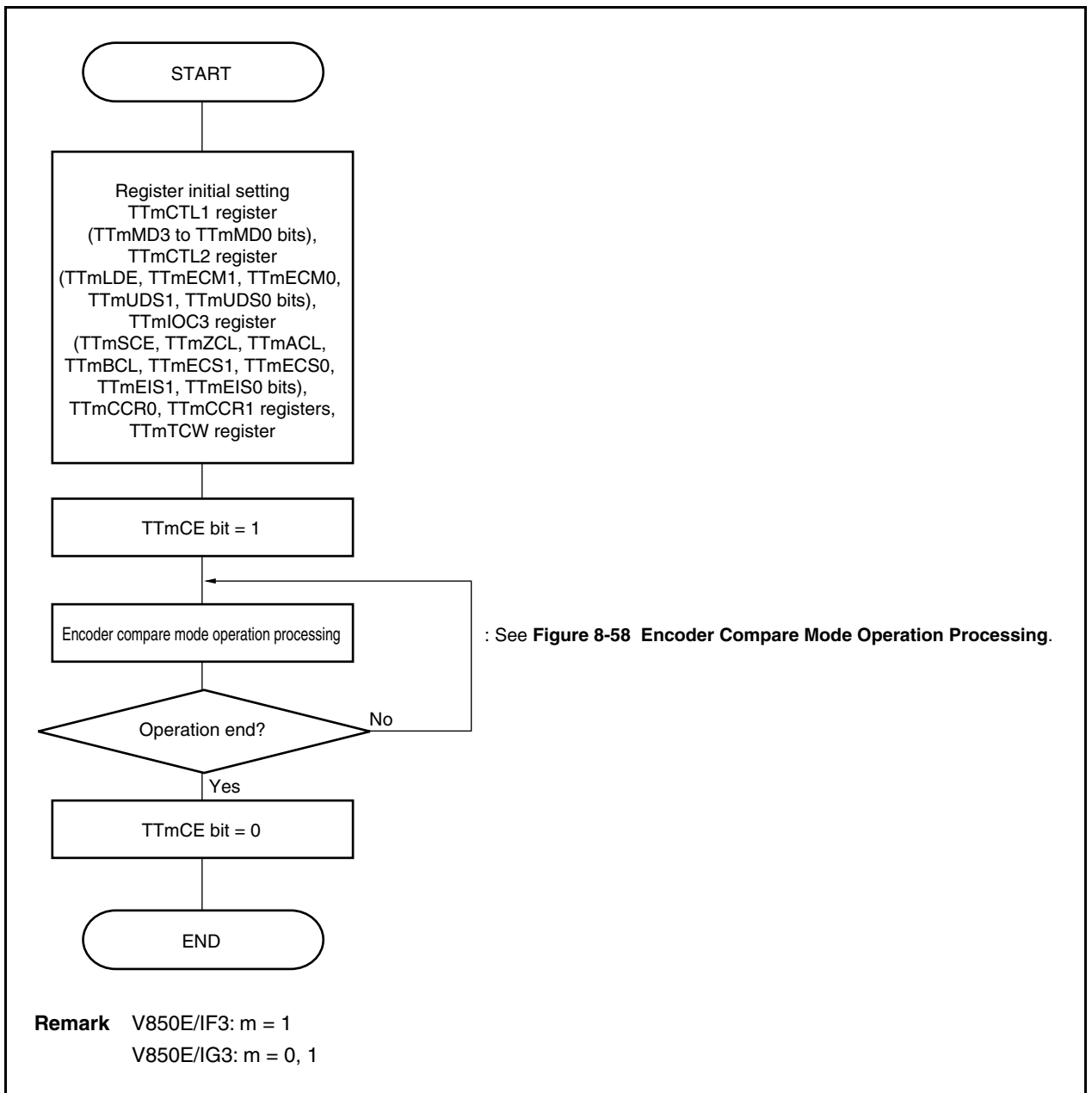
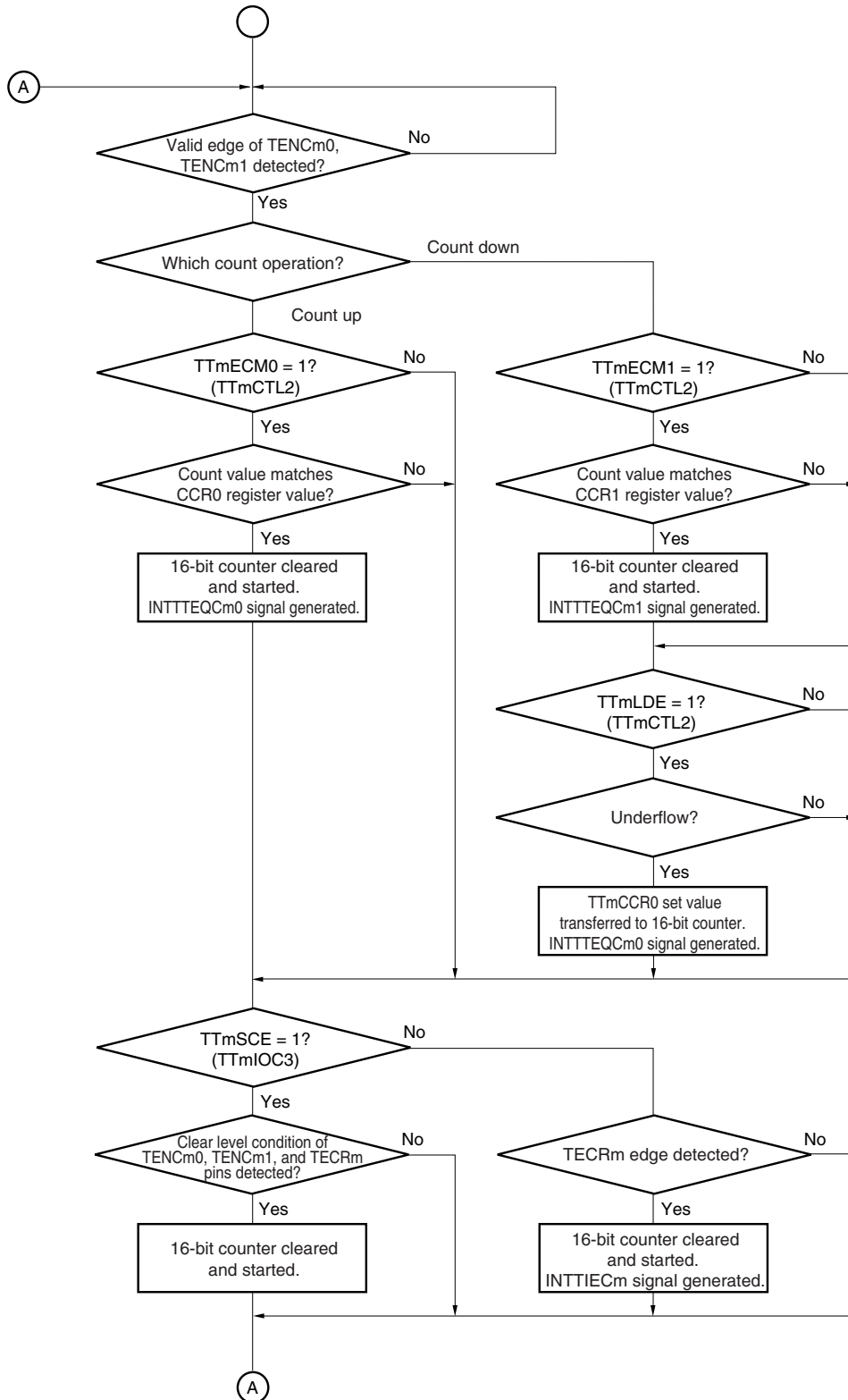


Figure 8-58. Encoder Compare Mode Operation Processing



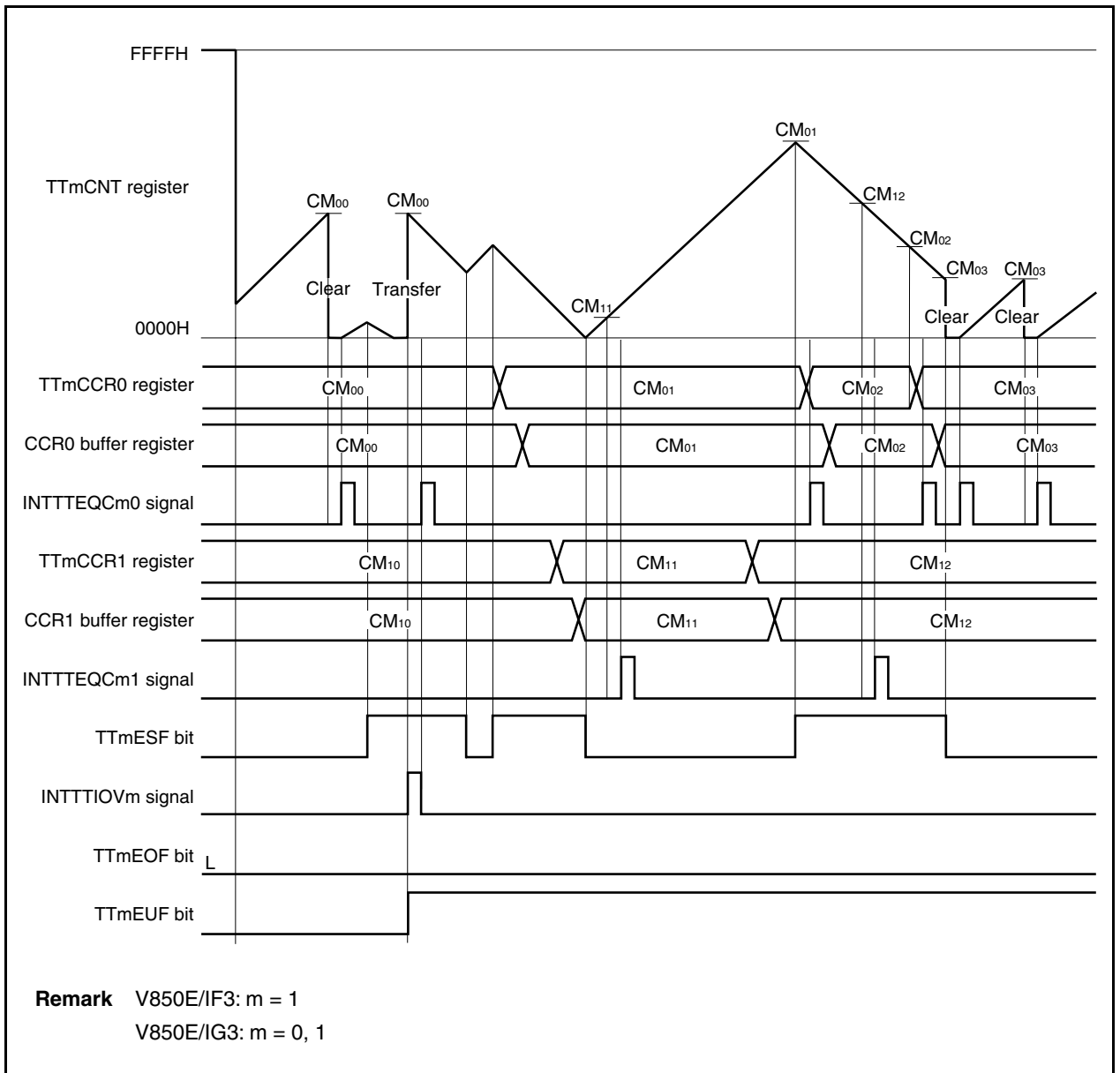
**Remark** V850E/IF3: m = 1  
V850E/IG3: m = 0, 1

(2) Encoder compare mode operation timing

(a) Basic timing 1

[Register setting conditions]

- TTmCTL2.TTmECM1 and TTmCTL2.TTmECM0 bits = 01  
The 16-bit counter is cleared to 0000H when its count value matches the value of the CCR0 buffer register.
- TTmCTL2.TTmLDE bit = 1  
The set value of the TTmCCR0 register is transferred to the 16-bit counter when it overflows.
- TTmIOC3.TTmSCE bit = 0, and TTmIOC3.TTmECS1 and TTmIOC3.TTmECS0 bits = 00  
Specification of the edge of encoder clear input signal (TECRm pin) to be detected and cleared (no edge specified)



When the 16-bit counter starts operating (TTmCE bit = 0 → 1), the set value of the TTmTCW register is transferred to the counter and the 16-bit counter starts operating.

When the count value of the counter matches the value of the CCR0 buffer register, the compare match interrupt request signal (INTTTEQm0) is generated. Because the TTmECM0 bit = 1, the 16-bit counter is cleared to 0000H if the next count operation is counting up.

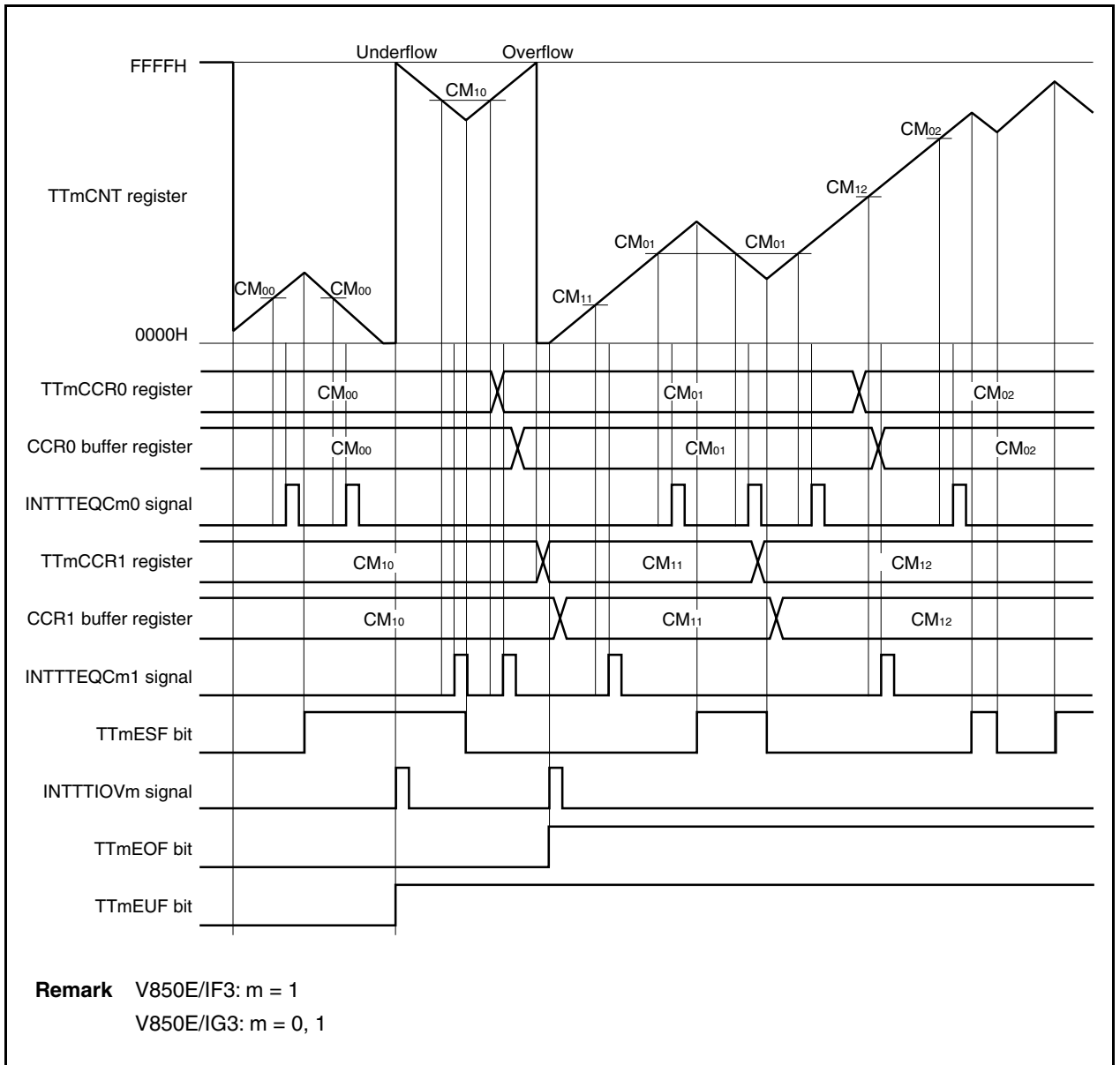
When the count value of the 16-bit counter matches the value of the CCR1 buffer register, the compare match interrupt request signal (INTTTEQm1) is generated. Because the TTmECM1 bit = 0, the 16-bit counter is not cleared to 0000H when its value matches that of the CCR1 buffer register.

When the TTmLDE bit = 1 and TTmECM0 bit = 1, the counter can operate in a range from 0000H to the set value of the TTmCCR0 register.

(b) Basic timing 2

[Register setting condition]

- TTmCTL2.TTmECM1 and TTmCTL2.TTmECM0 bits = 00  
The 16-bit counter is not cleared even when its count value matches the value of the CCRa buffer register (a = 0, 1).
- TTmCTL2.TTmLDE bit = 0  
The set value of the TTmCCR0 register is not transferred to the 16-bit counter after the counter underflows.
- TTmIOC3.TTmSCE bit = 0, and TTmIOC3.TTmECS1 and TTmIOC3.TTmECS0 bits = 00  
Specification of the edge of the encoder clear input signal (TECRm pin) to be detected and cleared (no edge specified)



When the 16-bit counter starts operating (TTmCE bit = 0 → 1), the set value of the TTmTCW register is transferred to the 16-bit counter and the counter starts operating.

When the count value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTTEQCm0) is generated.

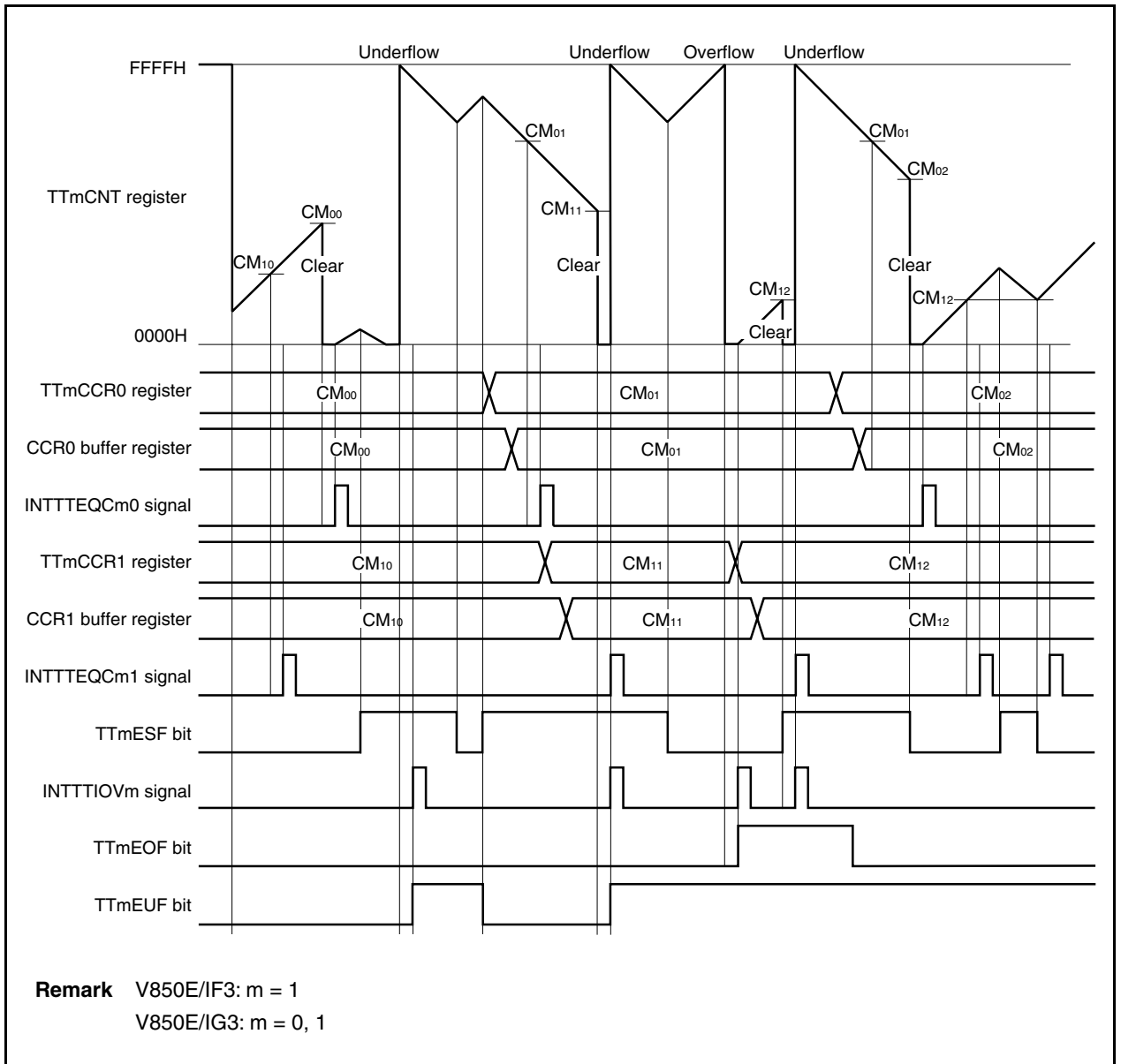
When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTTEQCm1) is generated.

The 16-bit counter is not cleared to 0000H even when its count value matches the value of the CCRa buffer register because the TTmECM1 and TTmECM0 bits = 00 (a = 0, 1).

(c) Basic timing 3

[Register setting condition]

- TTmCTL2.TTmECM1 and TTmCTL2.TTmECM0 bits = 11  
The count value of the 16-bit counter is cleared to 0000H when its value matches the value of the CCR0 buffer register.
- Setting of the TTmCTL2.TTmLDE bit is invalid.
- TTmIOC3.TTmSCE bit = 0, and TTmIOC3.TTmECS1 and TTmIOC3.TTmECS0 bits = 00  
Specification of the edge of the encoder clear input signal (TECRm pin) to be detected and cleared (no edge specified)



When the 16-bit counter starts operating (TTmCE bit = 0 → 1), the set value of the TTmTCW register is transferred to the 16-bit counter and the counter starts operating.

When the count value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTTEQm0) is generated. At this time, the 16-bit counter is cleared to 0000H if the next count operation is counting up.

When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTTEQm1) is generated. At this time, the 16-bit counter is cleared to 0000H if the next count operation is counting down.



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

Timer M (TMM) is a 16-bit interval timer.

The V850E/IF3 and V850E/IG3 incorporate TMM0 to TMM3.

### 9.1 Overview

An outline of TMMn is shown below (n = 0 to 3).

- 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 count operation.)
- Compare match interrupt × 1

Timer M supports only the clear & start mode. The free-running timer mode is not supported.

## 9.2 Configuration

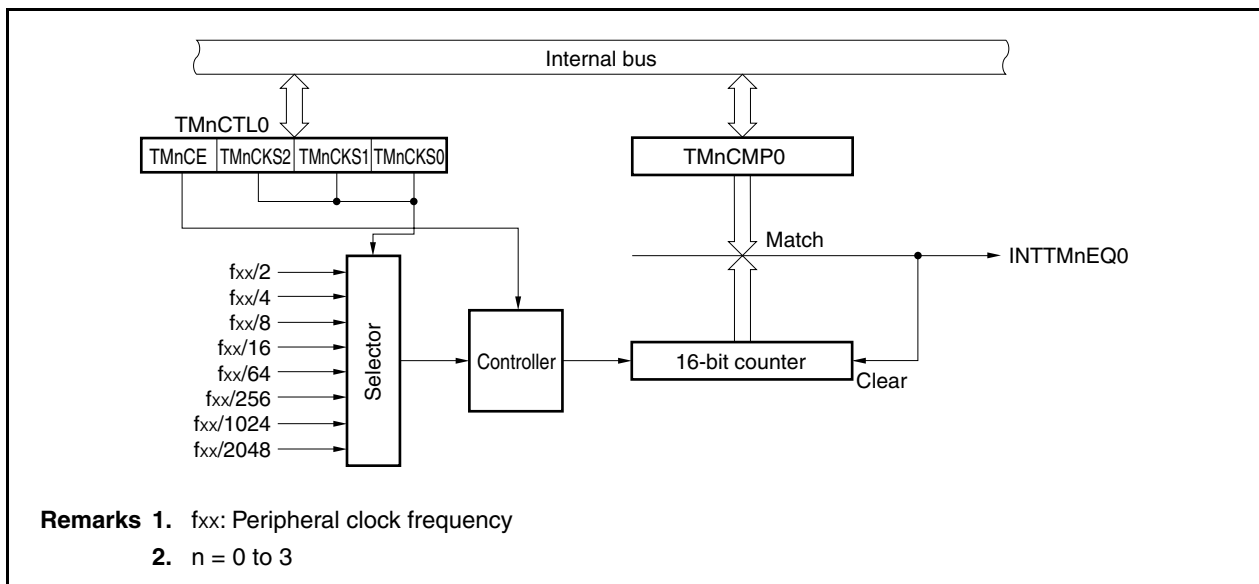
TMMn includes the following hardware (n = 0 to 3).

**Table 9-1. Configuration of TMMn**

| Item             | Configuration                     |
|------------------|-----------------------------------|
| Timer register   | 16-bit counter × 1                |
| Register         | TMMn compare register 0 (TMnCMP0) |
| Control register | TMMn control register 0 (TMnCTL0) |

**Remark** n = 0 to 3

**Figure 9-1. Block Diagram of TMMn**

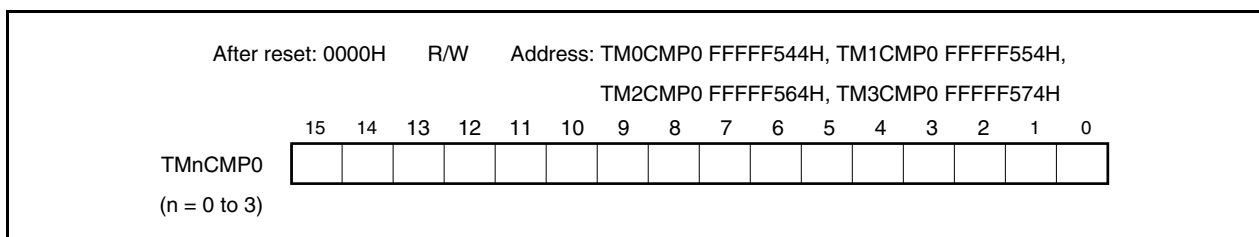


**(1) 16-bit counter**

This is a 16-bit counter that counts the internal clock.  
The 16-bit counter cannot be read or written.

**(2) TMMn compare register 0 (TMnCMP0)**

The TMnCMP0 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 TMnCMP0 register by software.  
Rewriting the TMnCMP0 register is prohibited during TMMn operation (TMnCTL0.TMnCE bit = 1).



### 9.3 Control Register

#### (1) TMMn control register 0 (TMnCTL0)

The TMnCTL0 register is an 8-bit register that controls the TMMn 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 TMnCTL0 register by software.

After reset: 00H    R/W    Address: TM0CTL0 FFFFF540H, TM1CTL0 FFFFF550H,  
TM2CTL0 FFFFF560H, TM3CTL0 FFFFF570H

|                         |       |   |   |   |   |         |         |         |
|-------------------------|-------|---|---|---|---|---------|---------|---------|
|                         | <7>   | 6 | 5 | 4 | 3 | 2       | 1       | 0       |
| TMnCTL0<br>(n = 0 to 3) | TMnCE | 0 | 0 | 0 | 0 | TMnCKS2 | TMnCKS1 | TMnCKS0 |

|  |   |
|--|---|
| TMnCE  | Internal clock operation enable/disable specification                       |
| 0  | TMMn operation disabled (16-bit counter reset asynchronously)               |
| 1  | TMMn operation enabled. Start operation clock supply. Start TMMn operation. |
| The internal clock control and internal circuit reset for TMMn are performed asynchronously with the TMnCE bit. When the TMnCE bit is cleared to 0, the internal clock of TMMn is stopped (fixed to low level) and 16-bit counter is reset asynchronously. |   |

| TMnCKS2 | TMnCKS1 | TMnCKS0 | Count clock selection |
|---------|---------|---------|-----------------------|
| 0       | 0       | 0       | f <sub>xx</sub> /2    |
| 0       | 0       | 1       | f <sub>xx</sub> /4    |
| 0       | 1       | 0       | f <sub>xx</sub> /8    |
| 0       | 1       | 1       | f <sub>xx</sub> /16   |
| 1       | 0       | 0       | f <sub>xx</sub> /64   |
| 1       | 0       | 1       | f <sub>xx</sub> /256  |
| 1       | 1       | 0       | f <sub>xx</sub> /1024 |
| 1       | 1       | 1       | f <sub>xx</sub> /2048 |

- Cautions**
1. Set the TMnCKS2 to TMnCKS0 bits when the TMnCE bit = 0.  
However, when changing the value of the TMnCE bit from 0 to 1, it is impossible to set the value of the TMnCKS2 to TMnCKS0 bits simultaneously.
  2. Be sure to clear bits 3 to 6 to "0".

**Remark** f<sub>xx</sub>: Peripheral clock frequency

## 9.4 Operation

### 9.4.1 Interval timer mode

In the interval timer mode, an interrupt request signal (INTTMnEQ0) is generated at the interval set by the TMnCMP0 register if the TMnCTL0.TMnCE bit is set to 1.

Figure 9-2. Configuration of Interval Timer

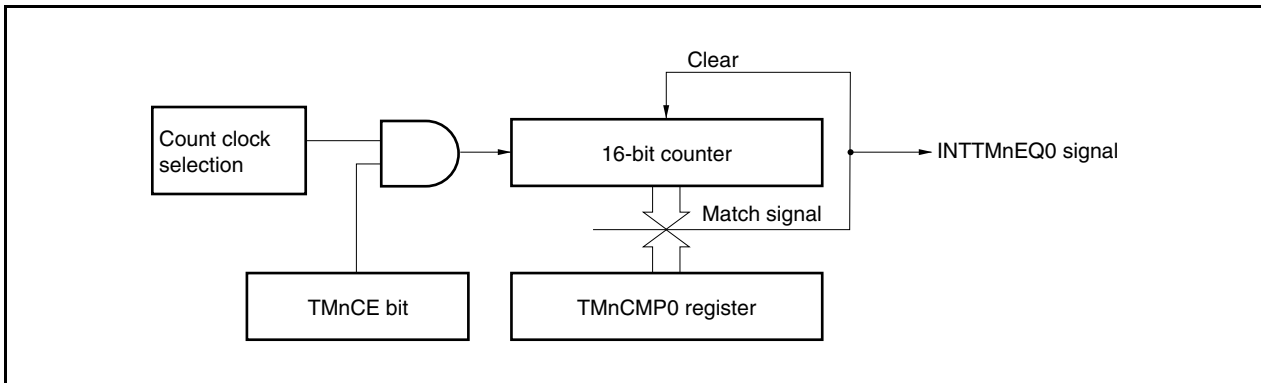
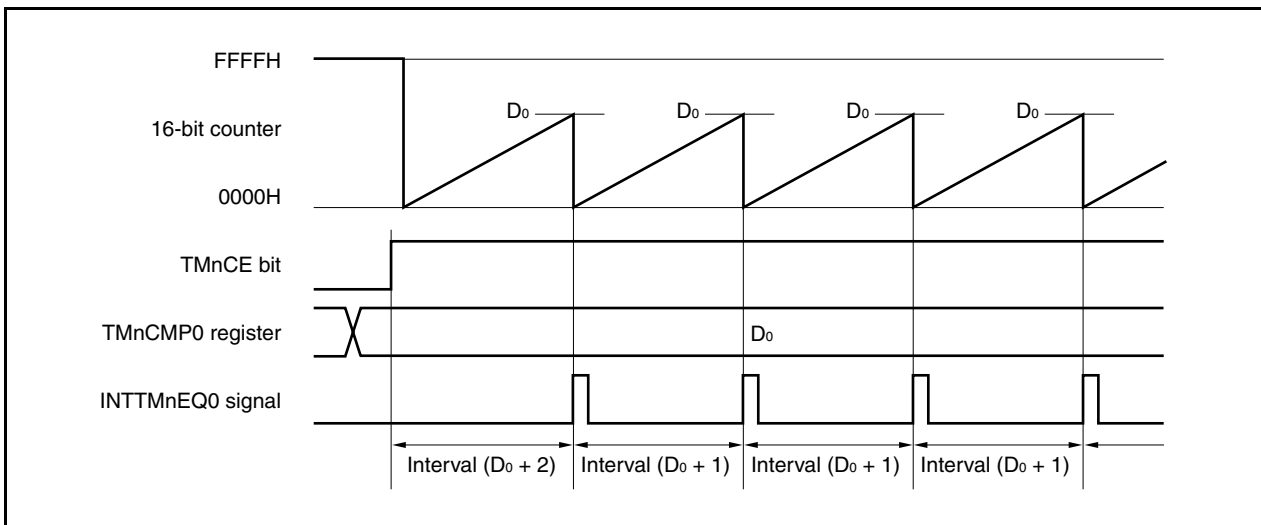


Figure 9-3. Basic Timing of Operation in Interval Timer Mode



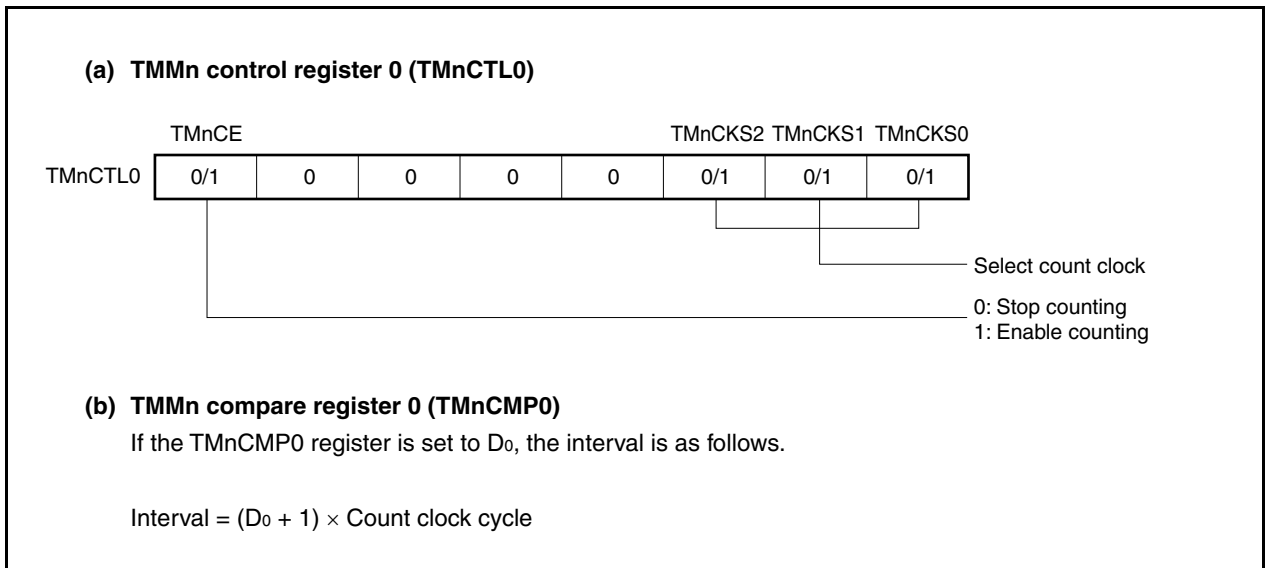
When the TMnCE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H in synchronization with the count clock, and the counter starts counting.

When the count value of the 16-bit counter matches the value of the TMnCMP0 register, the 16-bit counter is cleared to 0000H, and a compare match interrupt request signal (INTTMnEQ0) is generated.

The interval can be calculated by the following expression.

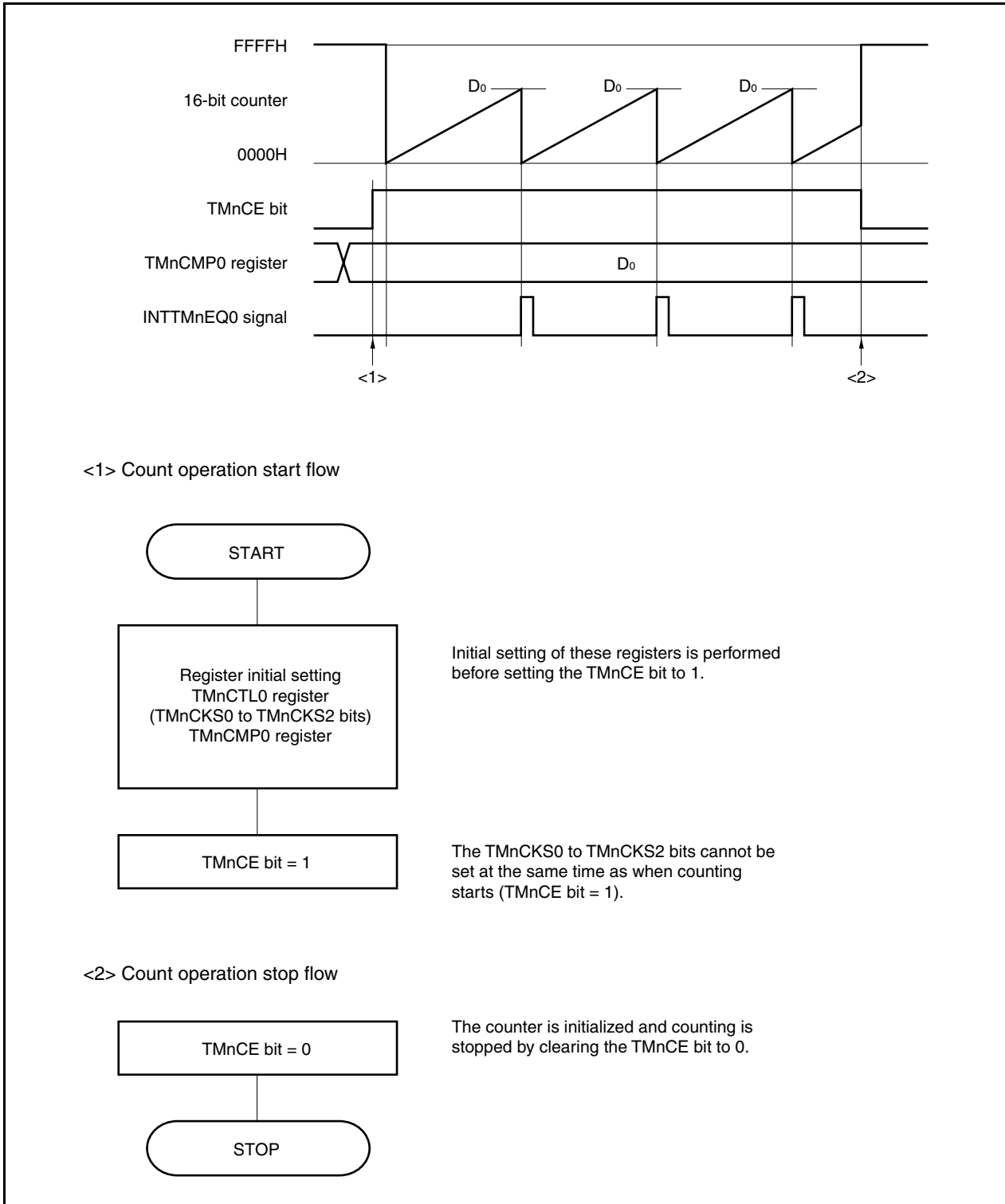
$$\text{Interval} = (\text{Set value of TMnCMP0 register} + 1) \times \text{Count clock cycle}$$

**Figure 9-4. Register Setting for Interval Timer Mode Operation**



(1) Interval timer mode operation flow

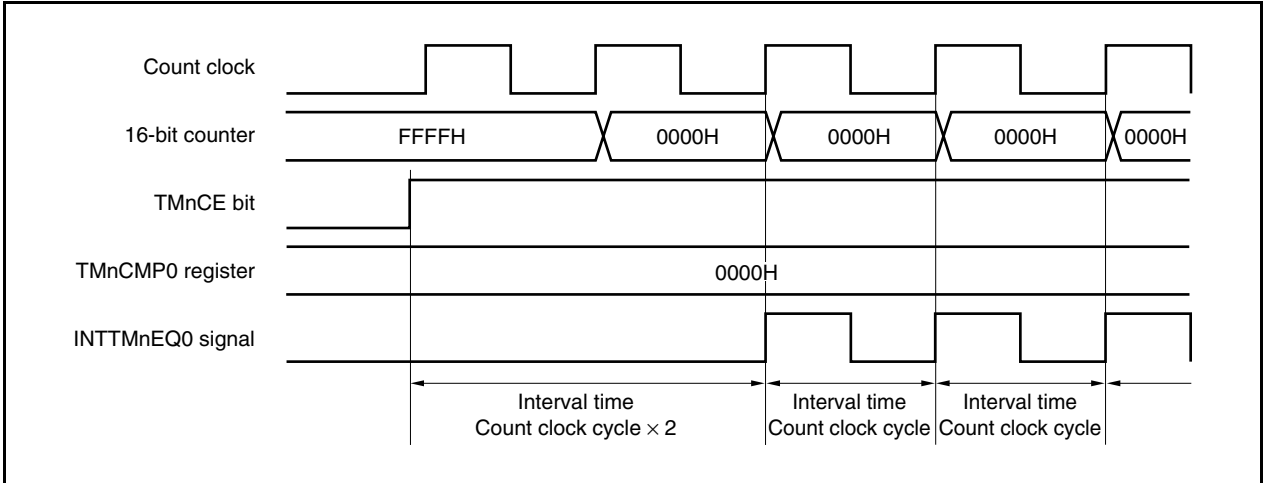
Figure 9-5. Software Processing Flow in Interval Timer Mode



(2) Interval timer mode operation timing

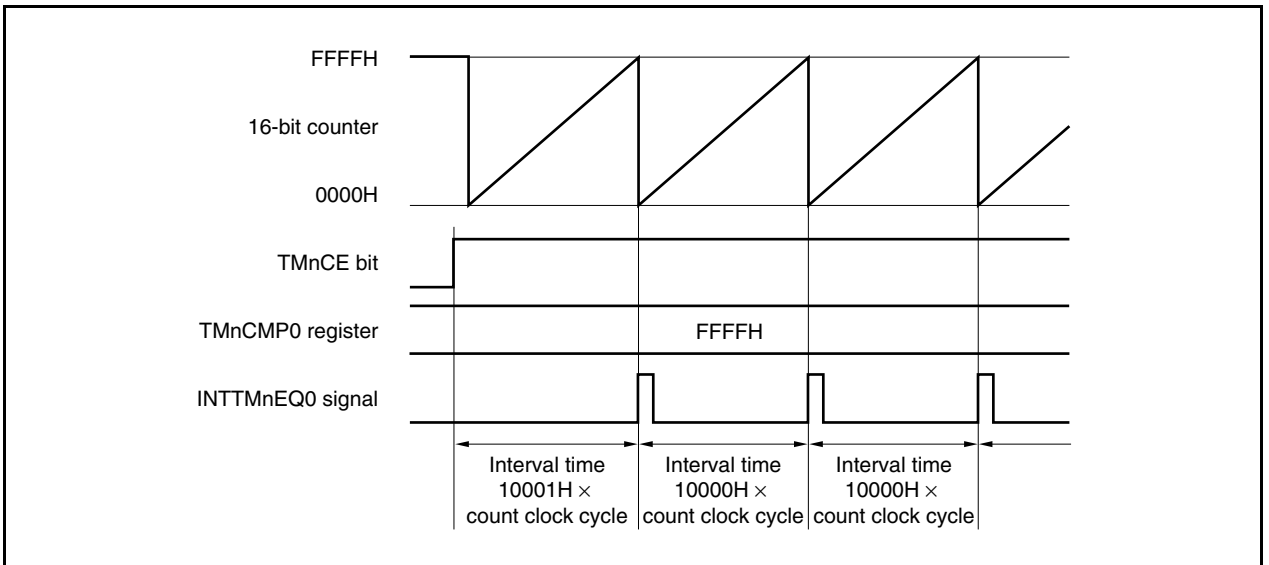
(a) Operation if TMnCMP0 register is set to 0000H

If the TMnCMP0 register is set to 0000H, the INTTMnEQ0 signal is generated at each count clock. The value of the 16-bit counter is always 0000H.



(b) Operation if TMnCMP0 register is set to FFFFH

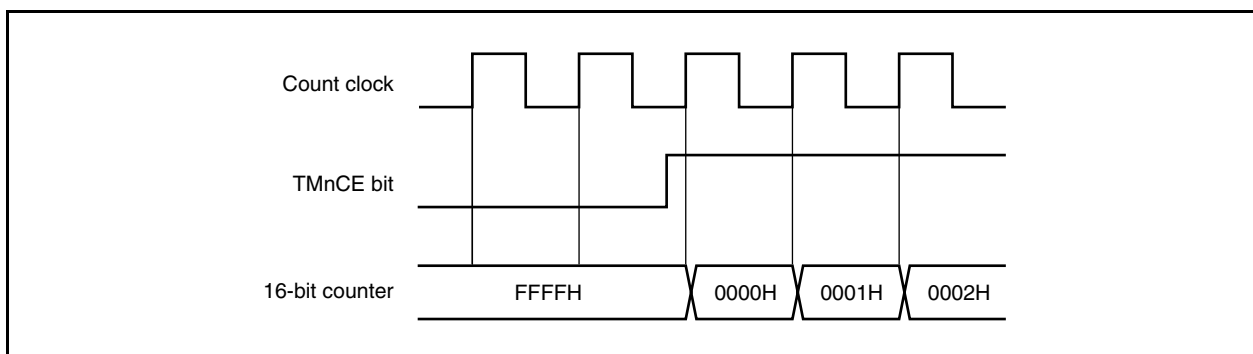
If the TMnCMP0 register is set to FFFFH, the 16-bit counter counts up to FFFFH. The counter is cleared to 0000H in synchronization with the next count-up timing. The INTTMnEQ0 signal is generated.



## 9.5 Cautions

### (1) Error on starting timer

It takes one clock to generate the first compare match interrupt request signal (INTTMMnEQ0) after the TMnCTL0.TMnCE bit is set to 1 and TMMn is started. This is because the value of the 16-bit counter is FFFFH when the TMnCE bit = 0 and TMMn is started asynchronously to the count clock.



### (2) Rewriting the TMnCMP0 and TMnCTL0 registers is prohibited while TMMn is operating.

If these registers are rewritten while the TMnCTL0.TMnCE bit is 1, the operation cannot be guaranteed. If they are rewritten by mistake, clear the TMnCE bit to 0, and re-set the registers.



## CHAPTER 10 MOTOR CONTROL FUNCTION

### 10.1 Functional Overview

Timer ABn (TABn) and the TMQn option (TMQOPn) can be used as an inverter function that controls a motor. It performs a tuning operation with timer AAn (TAAAn) and A/D conversion of A/D converters 0 and 1 can be started when the value of TABn matches the value of TAAAn. The following operations can be performed as motor control functions.

- 6-phase PWM output function with 16-bit accuracy (with dead-timer, for upper and lower arms)
- Timer tuning operation function (tunable with TAAAn)
- Period setting function (period can be changed during operation of crest or valley interrupt)
- Compare register rewriting: Anytime rewrite, batch write, or intermittent rewrite (selectable during TABn operation)
- Interrupt and transfer culling functions
- Dead-time setting function
- A/D trigger timing function of A/D converters 0 and 1 (four types of timing can be generated)
- 0% output and 100% output available
- 0% output and 100% output selectable by crest interrupt and valley interrupt
- Forced output stop function
  - At valid edge detection by external pin input (TOBnOFF, TOAmOFF)
  - At overvoltage detection by comparator function of A/D converter
  - At main clock oscillation stop detection by clock monitor function

**Remark** V850E/IF3: n = 0, 1, m = 2  
V850E/IG3: n = 0, 1, m = 2, 3

## 10.2 Configuration

The motor control function consists of the following hardware.

| Item              | Configuration   |
|-------------------|---|
| Timer register    | Dead-time counter m   |
| Compare register  | TABn dead-time compare register (TABnDTC register)  |
| Control registers | TABn option register 0 (TABnOPT0)<br>TABn option register 1 (TABnOPT1)<br>TABn option register 2 (TABnOPT2)<br>TABn option register 3 (TABnOPT3)<br>TABn I/O control register 3 (TABnIOC3)<br>High-impedance output control registers 0, 1 (HZAyCTLa) |

**Remark** V850E/IF3: m = 0 to 3, n = 0, 1, y = 0, 2, 3, a = 0 when y = 1, a = 0, 1  
V850E/IG3: m = 0 to 3, n = 0, 1, y = 0 to 3, a = 0, 1

- 6-phase PWM output can be produced with dead time by using the output of TABn (TOBn1, TOBn2, TOBn3)
- The output level of the 6-phase PWM output can be set individually.
- The 16-bit timer/counter of TABn counts up/down triangular waves. When the timer/counter underflows and when a period match occurs, an interrupt is generated. Interrupt generation, however, can be suppressed up to 31 times.
- TAA<sub>n</sub> can execute counting at the same time as TABn (timer tuning operation function). TAA<sub>n</sub> can be set in four ways as it can generate two types of A/D trigger sources (INTTAnCC0 and INTTAnCC1), and two types of interrupts: on underflow interrupt of TABn (INTTBnOV) and period match interrupt (INTTBnCC0).

Figure 10-1. Block Diagram of Motor Control

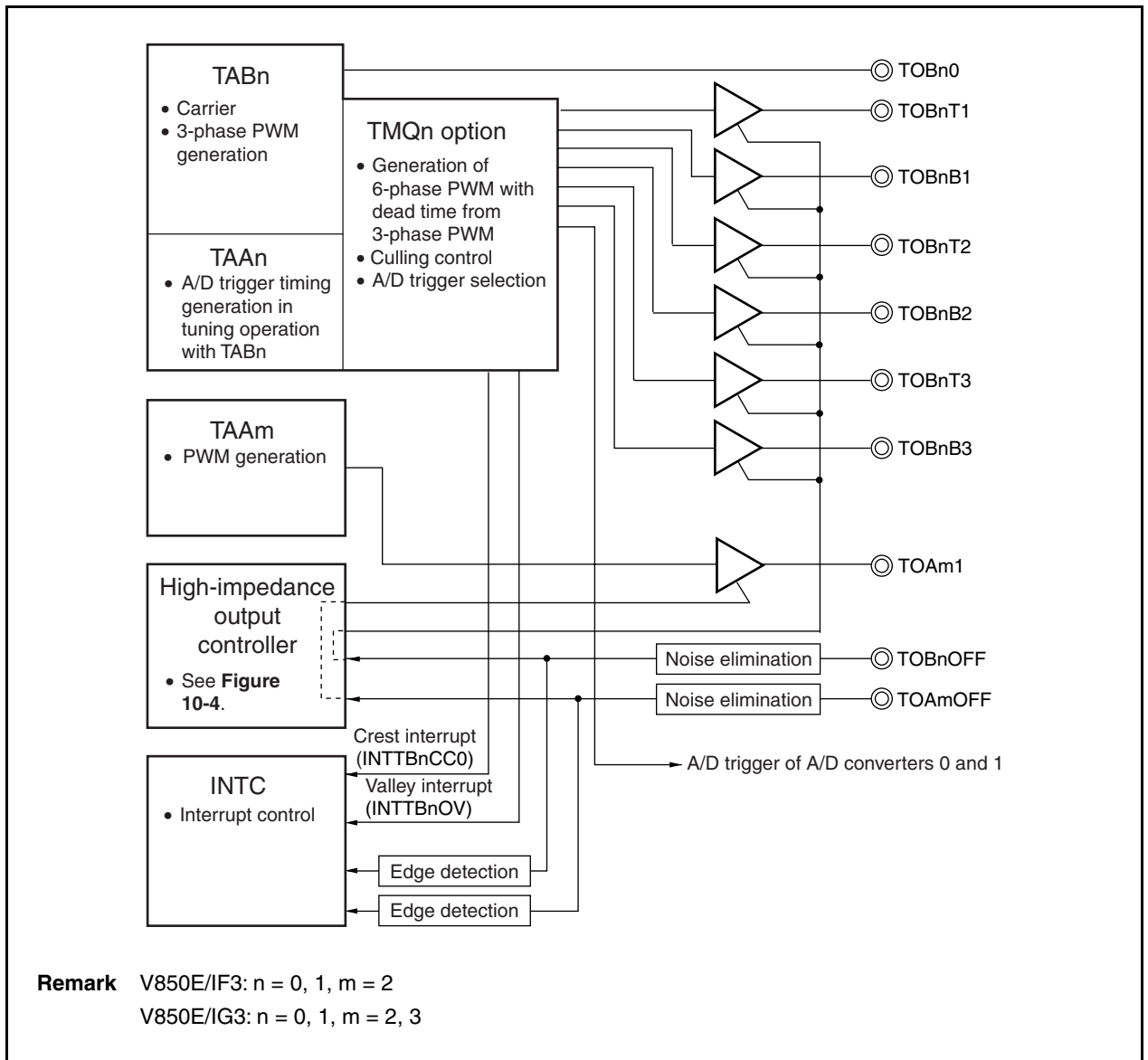
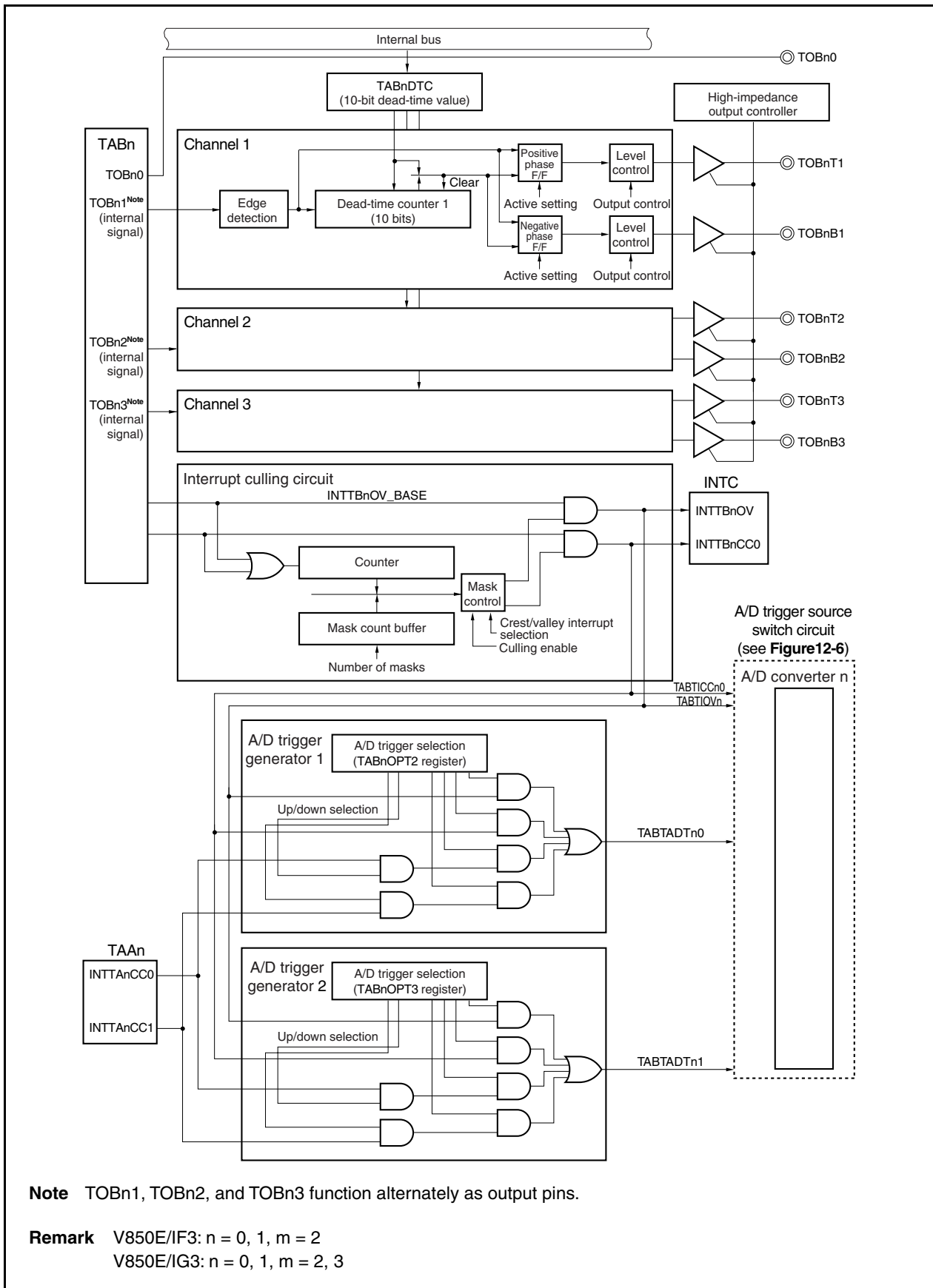


Figure 10-2. TMQn Option



**Note**  $TOBn1$ ,  $TOBn2$ , and  $TOBn3$  function alternately as output pins.

**Remark** V850E/IF3:  $n = 0, 1, m = 2$   
 V850E/IG3:  $n = 0, 1, m = 2, 3$

**(1) TABn dead-time compare register (TABnDTC)**

The TABnDTC register is a 10-bit compare register that specifies a dead-time value.

Rewriting this register is prohibited when the TABnCTL0.TABnCE bit = 1.

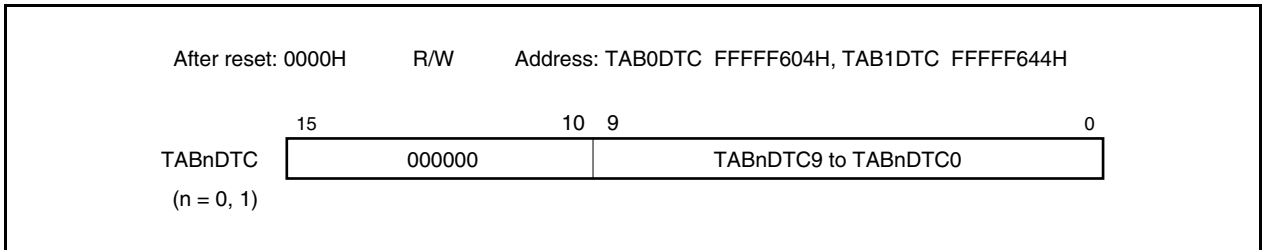
This register can be read or written in 16-bit units.

Reset sets this register to 0000H.

**Caution** To generate a dead time period, set a value of 1 or greater to the TABnDTC register.

While the operation is stopped (TABnCTL0.TABnCE bit = 0), the dead time period is not generated and the output levels of the TOBnT1 to TOBnT3 and TOBnB1 to TOBnB3 pins are in the initial status. To protect the system, therefore, allow the TOBnT1 to TOBnT3 and TOBnB1 to TOBnB3 pins to go into a high-impedance state or select the port mode with setting the output levels of the pins, before stopping the operation.

If the dead time period is not necessary, set the TABnDTC register to 0.



**(2) Dead-time counters 1 to 3**

The dead-time counters are 10-bit counters that count dead time.

These counters are cleared or count up at the rising or falling edge of the TOBnm output signal by TABn, and are cleared and stopped when their count value matches the value of the TABnDTC register. The count clock of these counters is the same as that set by the TABnCTL0.TABnCKS2 to TABnCTL0.TABnCKS0 bits of TABn.

**Remarks 1.** The operation differs when the TABnOPT2.TABnDTM bit = 1. For details, see **10.4.2 (4) Automatic dead-time width narrowing function (TABnOPT2.TABnDTM bit = 1).**

**2.** n = 0, 1, m = 1 to 3

### 10.3 Control Registers

#### (1) TABn option register 0 (TABnOPT0)

The TABnOPT0 register is an 8-bit register that controls the timer Qn option function. This register can be read or written in 8-bit or 1-bit units. However, the TABnCUF bit is read-only. Reset sets this register to 00H.

**Caution** The TABnCMS and TABnCUF bits can be set only in the 6-phase PWM output mode. Be sure to clear these bits to 0 when TABn is used alone.

After reset: 00H      R/W      Address: TAB0OPT0 FFFF5E5H, TAB1OPT0 FFFF625H

|                        |                            |                            |                            |                            |   |         |         |                           |
|------------------------|----------------------------|----------------------------|----------------------------|----------------------------|---|---------|---------|---------------------------|
|                        | <7>                        | <6>                        | <5>                        | <4>                        | 3 | <2>     | <1>     | <0>                       |
| TABnOPT0<br>(n = 0, 1) | TABnCCS3 <sup>Note 1</sup> | TABnCCS2 <sup>Note 1</sup> | TABnCCS1 <sup>Note 1</sup> | TABnCCS0 <sup>Note 1</sup> | 0 | TABnCMS | TABnCUF | TABnOVF <sup>Note 2</sup> |

|  |   |
|--|---|
| <b>TABnCMS</b>   | Compare register rewrite mode selection |
| 0  | Batch write mode (transfer operation)   |
| 1  | Anytime write mode                      |
| <ul style="list-style-type: none"> <li>• The TABnCMS bit is valid only when the 6-phase PWM output mode is set (when the TABnCTL1.TABnMD2 to TABnCTL1.TABnMD0 bits = 111). Clear the TABnCMS bit to 0 in any other mode.</li> <li>• The TABnCMS bit can be rewritten while the timer is operating (when the TABnCTL0.TABnCE bit = 1).</li> <li>• The following compare registers are rewritten in the batch write mode. TABnCCR0 to TABnCCR3, TAnCCR0, TAnCCR1, TABnOPT1, and TABnDTC registers</li> </ul> |   |

|   |                                       |
|---|---------------------------------------|
| <b>TABnCUF</b>  | Up-count/down-count flag of timer ABn |
| 0   | Timer ABn is counting up.             |
| 1   | Timer ABn is counting down.           |
| The TABnCUF bit is valid only when the 6-phase PWM output mode is set (when the TABnCTL1.TABnMD2 to TABnCTL1.TABnMD0 bits = 111). |                                       |

**Notes**

1. Be sure to clear the TABnCCS3 to TABnCCS0 bits to 0 in the 6-phase PWM output mode.
2. For details of the TABnOVF bit, see **CHAPTER 7 16-BIT TIMER/EVENT COUNTER AB (TAB)**.

**(2) TABn option register 1 (TABnOPT1)**

The TABnOPT1 register is an 8-bit register that controls the interrupt request signal generated by the timer Qn option function.

The TABnOPT1 register generates the signals output to the interrupt culling circuit, A/D trigger generator 1, and A/D trigger generator 2 shown in Figure 10-2.

This register can be rewritten when the TABnCTL0.TABnCE bit is 1.

Two rewriting modes (batch write mode and anytime write mode) can be selected, depending on the setting of the TABnOPT0.TABnCMS bit.

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: TAB0OPT1 FFFFF600H, TAB1OPT1 FFFFF640H           |         |         |         |  |         |         |
| TABnOPT1<br>(n = 0, 1) | <7>  | <6>   | 5       | 4       | 3       | 2  | 1       | 0       |
|                        | TABnICE  | TABnIOE   | 0       | TABnID4 | TABnID3 | TABnID2                                      | TABnID1 | TABnID0 |
|                        | TABnICE  | Crest interrupt (INTTBnCC0 signal) enable <sup>Note</sup> |         |         |         |  |         |         |
| 0                      | Do not use INTTBnCC0 signal (do not use it as count signal for interrupt culling). |   |         |         |         |  |         |         |
| 1                      | Use INTTBnCC0 signal (use it as count signal for interrupt culling).               |   |         |         |         |  |         |         |
|                        | TABnIOE  | Valley interrupt (INTTBnOV signal) enable <sup>Note</sup> |         |         |         |  |         |         |
| 0                      | Do not use INTTBnOV signal (do not use it as count signal for interrupt culling).  |   |         |         |         |  |         |         |
| 1                      | Use INTTBnOV signal (use it as count signal for interrupt culling).                |   |         |         |         |  |         |         |
|                        | TABnID4  | TABnID3   | TABnID2 | TABnID1 | TABnID0 | Number of times of interrupt                 |         |         |
| 0                      | 0  | 0   | 0       | 0       | 0       | Not culled (all interrupts are output)       |         |         |
| 0                      | 0  | 0   | 0       | 0       | 1       | 1 masked (one of two interrupts is output)   |         |         |
| 0                      | 0  | 0   | 0       | 1       | 0       | 2 masked (one of three interrupts is output) |         |         |
| 0                      | 0  | 0   | 0       | 1       | 1       | 3 masked (one of four interrupts is output)  |         |         |
| :                      | :  | :   | :       | :       | :       | :  |         |         |
| 1                      | 1  | 1   | 1       | 0       | 0       | 28 masked (one of 29 interrupts is output)   |         |         |
| 1                      | 1  | 1   | 1       | 0       | 1       | 29 masked (one of 30 interrupts is output)   |         |         |
| 1                      | 1  | 1   | 1       | 1       | 0       | 30 masked (one of 31 interrupts is output)   |         |         |
| 1                      | 1  | 1   | 1       | 1       | 1       | 31 masked (one of 32 interrupts is output)   |         |         |

**Note** When using the crest interrupt (INTTBnCC0 signal) and the valley interrupt (INTTBnOV signal) as the count signal for interrupt culling or as the A/D trigger signal, set the signal to be used to 1. An A/D trigger is generated at the culled interrupt timing.

**(3) TABn option register 2 (TABnOPT2)**

The TABnOPT2 register is an 8-bit register that controls the timer Qn option function.

This register can be rewritten when the TABnCTL0.TABnCE bit is 1. However, rewriting the TABnDTM bit is prohibited when the TABnCE bit is 1. The same value can be rewritten.

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: TAB0OPT2 FFFFF601H, TAB1OPT2 FFFFF641H |          |          |         |         |         |         |
| TABnOPT2<br>n = 0, 1<br>m = 1 to 3   | <7>  | <6>   | <5>      | <4>      | <3>     | <2>     | <1>     | <0>     |
|  | TABnRDE  | TABnDTM   | TABnATM3 | TABnATM2 | TABnAT3 | TABnAT2 | TABnAT1 | TABnAT0 |
|  | TABnRDE  | Transfer culling enable                         |          |          |         |         |         |         |
| 0  | Do not cull transfer (transfer timing is generated every time at crest and valley).  |   |          |          |         |         |         |         |
| 1  | Cull transfer at the same interval as interrupt culling set by the TABnOPT1 register.  |   |          |          |         |         |         |         |
|  | TABnDTM  | Dead-time counter operation mode selection      |          |          |         |         |         |         |
| 0  | Dead-time counter counts up normally and, if TOBnm output of TABn is at a narrow interval (TOBnm output width < dead-time width), the dead-time counter is cleared and counts up again.  |   |          |          |         |         |         |         |
| 1  | Dead-time counter counts up normally and, if TOBnm output of TABn is at a narrow interval (TOBnm output width < dead-time width), the dead-time counter counts down and the dead-time control width is automatically narrowed. |   |          |          |         |         |         |         |
| Rewriting the TABnDTM bit is disabled during timer operation. If it is rewritten by mistake, stop the timer operation by clearing the TABnCE bit to 0, and re-set the TABnDTM bit. |  |   |          |          |         |         |         |         |

**Cautions 1.** When using interrupt culling (the TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits are set to other than 00000), be sure to set the TABnRDE bit to 1. Therefore, the interrupt and transfer are generated at the same timing. The interrupt and transfer cannot be set separately. If the interrupt and transfer are set separately (TABnRDE bit = 0), transfer is not performed normally.

**2.** To generate a dead time period, set a value 1 or greater to the TABnDTC register. While the operation is stopped (TABnCTL0.TABnCE bit = 0), the dead time period is not generated and the output levels of the TOBnT1 to TOBnT3 and TOBnB1 to TOBnB3 pins are in the initial status. To protect the system, therefore, allow the TOBnT1 to TOBnT3 and TOBnB1 to TOBnB3 pins go into a high-impedance state or select the port mode with setting the output levels of the pins, before stopping the operation. If the dead time period is not necessary, set the TABnDTC register to 0.



| TABnATM3 | TABnATM3 mode selection   |
|----------|---|
| 0        | Output A/D trigger signal (TABTADn0) for INTTAnCC1 interrupt while 16-bit counter is counting up.   |
| 1        | Output A/D trigger signal (TABTADn0) for INTTAnCC1 interrupt while 16-bit counter is counting down. |

| TABnATM2 | TABnATM2 mode selection   |
|----------|---|
| 0        | Output A/D trigger signal (TABTADn0) for INTTAnCC0 interrupt while 16-bit counter is counting up.   |
| 1        | Output A/D trigger signal (TABTADn0) for INTTAnCC0 interrupt while 16-bit counter is counting down. |

| TABnAT3 <sup>Note</sup> | A/D trigger output control 3   |
|-------------------------|--|
| 0                       | Disable output of A/D trigger signal (TABTADn0) for INTTAnCC1 interrupt. |
| 1                       | Enable output of A/D trigger signal (TABTADn0) for INTTAnCC1 interrupt.  |

| TABnAT2 <sup>Note</sup> | A/D trigger output control 2   |
|-------------------------|--|
| 0                       | Disable output of A/D trigger signal (TABTADn0) for INTTAnCC0 interrupt. |
| 1                       | Enable output of A/D trigger signal (TABTADn0) for INTTAnCC0 interrupt.  |

| TABnAT1 <sup>Note</sup> | A/D trigger output control 1   |
|-------------------------|--|
| 0                       | Disable output of A/D trigger signal (TABTADn0) for INTTBnCC0 (crest interrupt). |
| 1                       | Enable output of A/D trigger signal (TABTADn0) for INTTBnCC0 (crest interrupt).  |

| TABnAT0 <sup>Note</sup> | A/D trigger output control 0   |
|-------------------------|--|
| 0                       | Disable output of A/D trigger signal (TABTADn0) for INTTBnOV (valley interrupt). |
| 1                       | Enable output of A/D trigger signal (TABTADn0) for INTTBnOV (valley interrupt).  |

**Note** For the setting of the TABnAT3 to TABnAT0 bits, see **CHAPTER 12 A/D CONVERTERS 0 AND 1**.

**(4) TABn option register 3 (TABnOPT3)**

The TABnOPT3 register is an 8-bit register that controls the timer Qn option function.

This register can be rewritten when the TABnCTL0.TABnCE bit is 1.

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: TAB0OPT3 FFFFF603H, TAB1OPT3 FFFFF643H  |          |         |         |         |         |
|                         | 7 | 6   | <5>  | <4>      | <3>     | <2>     | <1>     | <0>     |
| TABnOPT3<br>(n = 0, 1)  | 0 | 0   | TABnATM7   | TABnATM6 | TABnAT7 | TABnAT6 | TABnAT5 | TABnAT4 |
| TABnATM7                |   |     | TABnATM7 mode selection  |          |         |         |         |         |
| 0                       |   |     | Output A/D trigger signal (TABTADTn1) of INTTAnCC1 interrupt while dead-time counter is counting up.   |          |         |         |         |         |
| 1                       |   |     | Output A/D trigger signal (TABTADTn1) of INTTAnCC1 interrupt while dead-time counter is counting down. |          |         |         |         |         |
| TABnATM6                |   |     | TABnATM6 mode selection  |          |         |         |         |         |
| 0                       |   |     | Output A/D trigger signal (TABTADTn1) of INTTAnCC0 interrupt while dead-time counter is counting up.   |          |         |         |         |         |
| 1                       |   |     | Output A/D trigger signal (TABTADTn1) of INTTAnCC0 interrupt while dead-time counter is counting down. |          |         |         |         |         |
| TABnAT7 <sup>Note</sup> |   |     | A/D trigger output control 3   |          |         |         |         |         |
| 0                       |   |     | Disable output of A/D trigger signal (TABTADTn1) for INTTAnCC1 interrupt.                              |          |         |         |         |         |
| 1                       |   |     | Enable output of A/D trigger signal (TABTADTn1) for INTTAnCC1 interrupt.                               |          |         |         |         |         |
| TABnAT6 <sup>Note</sup> |   |     | A/D trigger output control 2   |          |         |         |         |         |
| 0                       |   |     | Disable output of A/D trigger signal (TABTADTn1) for INTTAnCC0 interrupt.                              |          |         |         |         |         |
| 1                       |   |     | Enable output of A/D trigger signal (TABTADTn1) for INTTAnCC0 interrupt.                               |          |         |         |         |         |
| TABnAT5 <sup>Note</sup> |   |     | A/D trigger output control 1   |          |         |         |         |         |
| 0                       |   |     | Disable output of A/D trigger signal (TABTADTn1) for INTTBnCC0 interrupt (crest interrupt).            |          |         |         |         |         |
| 1                       |   |     | Enable output of A/D trigger signal (TABTADTn1) for INTTBnCC0 interrupt (crest interrupt).             |          |         |         |         |         |
| TABnAT4 <sup>Note</sup> |   |     | A/D trigger output control 0   |          |         |         |         |         |
| 0                       |   |     | Disable output of A/D trigger signal (TABTADTn1) for INTTBnOV interrupt (valley interrupt).            |          |         |         |         |         |
| 1                       |   |     | Enable output of A/D trigger signal (TABTADTn1) for INTTBnOV interrupt (valley interrupt).             |          |         |         |         |         |

**Note** For the setting of the TABnAT7 to TABnAT4 bits, see **CHAPTER 12 A/D CONVERTERS 0 AND 1**.

**(5) TABn I/O control register 3 (TABnIOC3)**

The TABnIOC3 register is an 8-bit register that controls the output of the timer Qn option function.

To output from the TOBnTm pin, set the TABnIOC0.TABnOEm bit to 1 and then set the TABnIOC3 register.

The TABnIOC3 register can be rewritten only when the TABnCTL0.TABnCE bit is 0.

Rewriting each bit of the TABnIOC3 register is prohibited when the TABnCTL0.TABnCE bit is 1; however the same value can be rewritten to each bit of the TABnIOC3 register when the TABnCTL0.TABnCE bit is 1.

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

Reset sets this register to A8H.

**Caution** Set the TABnIOC3 register to the default value (A8H) when the timer is used in a mode other than the 6-phase PWM output mode.

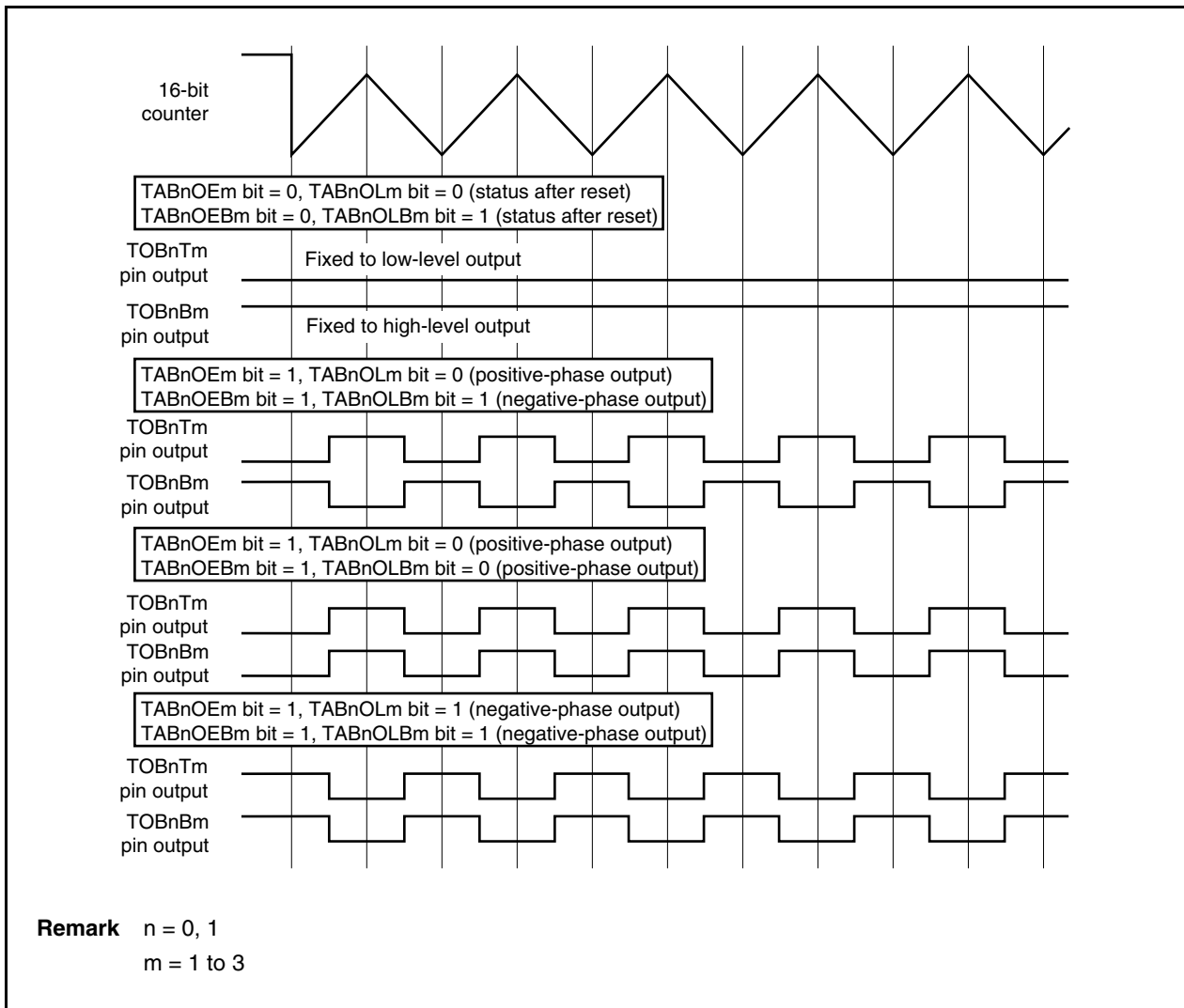
**Remark** Set the output level of the TOBnTm pin by the TABnIOC0 register.

|                                    |          |   |   |          |          |          |   |   |
|------------------------------------|----------|---|---|----------|----------|----------|---|---|
| After reset: A8H                   |          | R/W   | Address: TAB0IOC3 FFFFF602H, TAB1IOC3 FFFFF642H |          |          |          |   |   |
| TABnIOC3<br>n = 0, 1<br>m = 1 to 3 | <7>      | <6>   | <5>   | <4>      | <3>      | <2>      | 1 | 0 |
|                                    | TABnOLB3 | TABnOEB3  | TABnOLB2  | TABnOEB2 | TABnOLB1 | TABnOEB1 | 0 | 0 |
|                                    |          |   |   |          |          |          |   |   |
| TABnOLBm                           |          | Setting of TOBnBm pin output level  |   |          |          |          |   |   |
| 0                                  |          | Disable inversion of output of TOBnBm pin   |   |          |          |          |   |   |
| 1                                  |          | Enable inversion of output of TOBnBm pin  |   |          |          |          |   |   |
| TABnOEBm                           |          | Setting of TOBnBm pin output  |   |          |          |          |   |   |
| 0                                  |          | Disable TOBnBm pin output.<br><ul style="list-style-type: none"> <li>• When TABnOLBm bit = 0, low level is output from TOBnBm pin.</li> <li>• When TABnOLBm bit = 1, high level is output from TOBnBm pin.</li> </ul> |   |          |          |          |   |   |
| 1                                  |          | Enable TOBnBm pin output.   |   |          |          |          |   |   |

(a) Output from TOBnTm and TOBnBm pins

The TOBnTm pin output is controlled by the TABnIOC0.TABnOLm and TABnIOC0.TABnOEm bits. The TOBnBm pin output is controlled by the TABnIOC3.TABnOLBm and TABnIOC3.TABnOEBm bits. A timer output with each setting in the 6-phase PWM output mode is shown below.

Figure 10-3. TOBnTm and TOBnBm Pin Output Control (Without Dead Time)



**Table 10-1. TOBnTm Pin Output**

| TABnOLm Bit | TABnOEm Bit | TABnCE Bit | TOBnTm Pin Output            |
|-------------|-------------|------------|------------------------------|
| 0           | 0           | x          | Low-level output             |
|             | 1           | 0          | Low-level output             |
|             |             | 1          | TOBnTm positive-phase output |
| 1           | 0           | x          | High-level output            |
|             | 1           | 0          | High-level output            |
|             |             | 1          | TOBnTm negative-phase output |

**Remark** n = 0, 1  
m = 1 to 3

**Table 10-2. TOBnBm Pin Output**

| TABnOLBm Bit | TABnOEBm Bit | TABnCE Bit | TOBnBm Pin Output            |
|--------------|--------------|------------|------------------------------|
| 0            | 0            | x          | Low-level output             |
|              | 1            | 0          | Low-level output             |
|              |              | 1          | TOBnBm positive-phase output |
| 1            | 0            | x          | High-level output            |
|              | 1            | 0          | High-level output            |
|              |              | 1          | TOBnBm negative-phase output |

**Remark** n = 0, 1  
m = 1 to 3

**(6) High-impedance output control registers 00, 01, 10, 11, 20, 21, 30, 31 (HZAyCTL0, HZAyCTL1)**

The HZAyCTL0 and HZAyCTL1 registers are 8-bit registers that control the high-impedance state of the output buffer.

These registers can be read or written in 8-bit or 1-bit units. However, the HZAyDCFn bit is a read-only bit and cannot be written.

16-bit access is not possible.

Reset sets these registers to 00H.

The same value can be always rewritten to the HZAyCTLn register by software.

The relationship between detection factor and the control registers is shown below.

| Pins Subject to High-Impedance Control                               | High-Impedance Control Factor |  | Control Register         |
|--|-------------------------------|--|--------------------------|
|  | External Pin                  | A/D Unit (Comparator)  |                          |
| When TOB0T1 to TOB0T3 are output<br>When TOB0B1 to TOB0B3 are output | TOB0OFF                       | –  | HZA0CTL0                 |
|  | –                             | When low range reference voltage of ANI00/ANI05 input is exceeded (rising edge) or not attained (falling edge)                         | HZA2CTL0                 |
|  | –                             | When full range reference voltage of ANI00/ANI05 input is exceeded (rising edge) or not attained (falling edge)                        | HZA2CTL1                 |
| When TOA21 is output   | TOA2OFF                       | –  | HZA0CTL1                 |
| When TOB1T1 to TOB1T3 are output<br>When TOB1B1 to TOB1B3 are output | TOB1OFF                       | –  | HZA1CTL0 <sup>Note</sup> |
|  | –                             | When low range reference voltage of ANI10 to ANI12 and ANI15 to ANI17 inputs is exceeded (rising edge) or not attained (falling edge)  | HZA3CTL0                 |
|  | –                             | When full range reference voltage of ANI10 to ANI12 and ANI15 to ANI17 inputs is exceeded (rising edge) or not attained (falling edge) | HZA3CTL1                 |
| When TOA31 <sup>Note</sup> is output                                 | TOA3OFF <sup>Note</sup>       | –  | HZA1CTL1 <sup>Note</sup> |

**Note** V850E/IG3 only

**Caution** High-impedance control is performed only when a port pin is set to function as indicated in the above table.

After reset: 00H R/W Address: HZA0CTL0 FFFF610H, HZA0CTL1 FFFF611H,  
 HZA1CTL0 FFFF618H, HZA1CTL1 FFFF619H<sup>Note 1</sup>,  
 HZA2CTL0 FFFF650H, HZA2CTL1 FFFF651H,  
 HZA3CTL0 FFFF658H, HZA3CTL1 FFFF659H

|          |          |          |          |          |          |          |   |          |
|----------|----------|----------|----------|----------|----------|----------|---|----------|
|          | <7>      | <6>      | 5        | 4        | <3>      | <2>      | 1 | <0>      |
| HZAyCTLn | HZAyDCEn | HZAyDCMn | HZAyDCNn | HZAyDCPn | HZAyDCTn | HZAyDCCn | 0 | HZAyDCFn |

|  |          |  |
|--|----------|--|
| ( V850E/IF3<br>n = 0, 1<br>y = 0, 2, 3<br>n = 0 when y = 1 ) | HZAyDCEn | High-impedance output control  |
|  | 0        | Disable high-impedance output control operation. Pins can function as output pins. |
|  | 1        | Enable high-impedance output control operation.                                    |

|   |  |
|---|--|
| HZAyDCMn  | Condition of clearing high-impedance state by HZAyDCCn bit   |
| 0   | Setting of the HZAyDCCn bit is valid regardless of the external pin <sup>Note 2</sup> input.   |
| 1   | Setting of the HZAyDCCn bit is invalid while the external pin <sup>Note 2</sup> input holds a level detected as abnormal (active level). |
| Rewrite the HZAyDCMn bit when the HZAyDCEn bit = 0. |  |

**Notes 1.** V850E/IG3 only

**2.** • V850E/IF3

HZA0CTL0: TOB0OFF pin,            HZA0CTL1: TOA2OFF pin,  
 HZA1CTL0: TOB1OFF pin,  
 HZA2CTL0: ANI00/ANI05 pin,      HZA2CTL1: ANI00/ANI05 pin,  
 HZA3CTL0: ANI10 to ANI12, ANI15 to ANI17 pins,  
 HZA3CTL1: ANI10 to ANI12, ANI15 to ANI17 pins

• V850E/IG3

HZA0CTL0: TOB0OFF pin,            HZA0CTL1: TOA2OFF pin,  
 HZA1CTL0: TOB1OFF pin,            HZA1CTL1: TOA3OFF pin,  
 HZA2CTL0: ANI00/ANI05 pin,      HZA2CTL1: ANI00/ANI05 pin,  
 HZA3CTL0: ANI10 to ANI12, ANI15 to ANI17 pins,  
 HZA3CTL1: ANI10 to ANI12, ANI15 to ANI17 pins

| HZAyDCNn | HZAyDCPn | External pin <sup>Note 1</sup> input edge specification   |
|----------|----------|---|
| 0        | 0        | No valid edge (setting the HZAyDCFn bit by external pin <sup>Note 1</sup> input is prohibited).                                       |
| 0        | 1        | Rising edge of the external pin <sup>Note 1</sup> input is valid (abnormality is detected by rising edge input) <sup>Note 2</sup> .   |
| 1        | 0        | Falling edge of the external pin <sup>Note 1</sup> input is valid (abnormality is detected by falling edge input) <sup>Note 2</sup> . |
| 1        | 1        | Setting prohibited  |

- Rewrite the HZAyDCNn and HZAyDCPn bits when the HZAyDCEn bit is 0.
- For the edge specification of the INTP00, INTP02, INTP08, and INTP10 pins, see **20.4.2 (1) External interrupt rising edge specification register 0 (INTR0), external interrupt falling edge specification register 0 (INTF0) and (2) External interrupt rising edge specification register 1 (INTR1), external interrupt falling edge specification register 1 (INTF1)**.
- The edge of the external pins must be specified starting from the TOB0OFF, TOB1OFF, TOA2OFF, and TOA3OFF<sup>Note 3</sup> pins. Then the edge of the external pins other than the TOB0OFF, TOB1OFF, TOA2OFF, and TOA3OFF<sup>Note 3</sup> pins must be specified. Otherwise, the undefined edge may be detected when the edges of the TOB0OFF, TOB1OFF, TOA2OFF, and TOA3OFF<sup>Note 3</sup> pins are specified.
- High-impedance output control is performed when the valid edge is input after the operation is enabled (by setting HZAyDCEn bit to 1). If the external pin<sup>Note 1</sup> is at the active level when the operation is enabled, therefore, high-impedance output control is not performed.

**Notes 1.** • V850E/IF3

HZA0CTL0: TOB0OFF pin,           HZA0CTL1: TOA2OFF pin,  
 HZA1CTL0: TOB1OFF pin,  
 HZA2CTL0: ANI00/ANI05 pin,    HZA2CTL1: ANI00/ANI05 pin,  
 HZA3CTL0: ANI10 to ANI12, ANI15 to ANI17 pins,  
 HZA3CTL1: ANI10 to ANI12, ANI15 to ANI17 pins

• V850E/IG3

HZA0CTL0: TOB0OFF pin,           HZA0CTL1: TOA2OFF pin,  
 HZA1CTL0: TOB1OFF pin,           HZA1CTL1: TOA3OFF pin,  
 HZA2CTL0: ANI00/ANI05 pin,    HZA2CTL1: ANI00/ANI05 pin,  
 HZA3CTL0: ANI10 to ANI12, ANI15 to ANI17 pins,  
 HZA3CTL1: ANI10 to ANI12, ANI15 to ANI17 pins

2. For detecting the voltage of a comparator exceeding the reference voltage, set the rising edge input. For detecting the voltage of a comparator which has not attained the reference voltage, set the falling edge input.
3. V850E/IG3 only



| HZAyDCTn | High-impedance output trigger bit   |
|----------|---|
| 0        | No operation  |
| 1        | Pins are made to go into a high-impedance state by software and the HZAyDCFn bit is set to 1. |

- If an edge indicating abnormality is input to the external pin<sup>Note 2</sup> (which is detected according to the setting of the HZAyDCNn and HZAyDCPn bits), the HZAyDCTn bit is invalid even if it is set to 1.
- The HZAyDCTn bit is always 0 when it is read because it is a software-triggered bit.
- The HZAyDCTn bit is invalid even if it is set to 1 when the HZAyDCEn bit = 0.
- Simultaneously setting the HZAyDCTn and HZAyDCCn bits to 1 is prohibited.

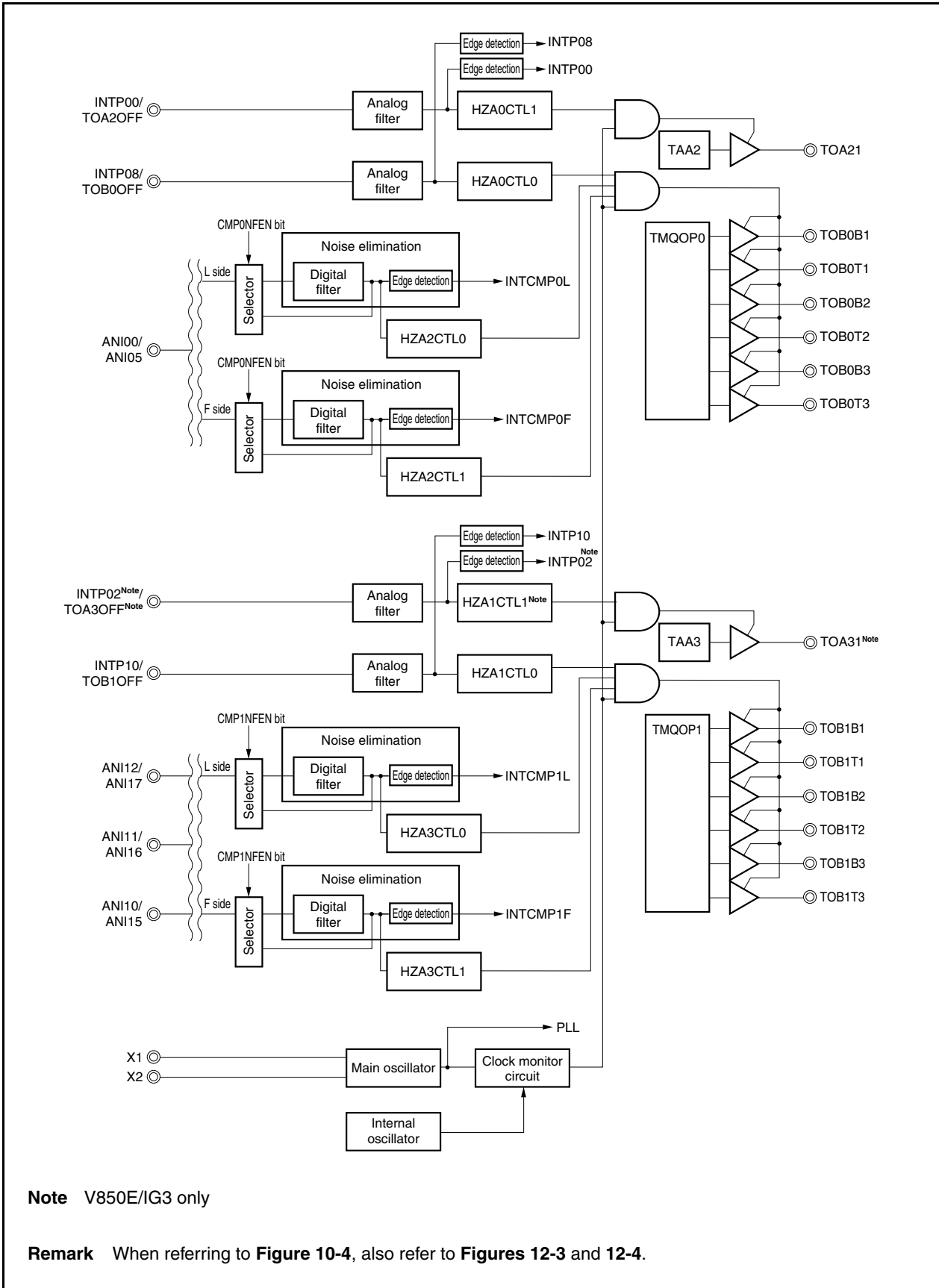
| HZAyDCCn | High-impedance output control clear bit  |
|----------|--|
| 0        | No operation   |
| 1        | Pins that have gone into a high-impedance state are output-enabled by software and the HZAyDCFn bit is cleared to 0. |

- Pins can function as output pins when the HZAyDCM bit = 0, regardless of the status of the external pin<sup>Note</sup>.
- If an edge indicating abnormality is input to the external pin<sup>Note</sup> (which is set by the HZAyDCNn and HZAyDCPn bits) when the HZAyDCM bit = 1, the HZAyDCCn bit is invalid even if it is set to 1.
- The HZAyDCCn bit is always 0 when it is read.
- The HZAyDCCn bit is invalid even if it is set to 1 when the HZAyDCEn bit = 0.
- Simultaneously setting the HZAyDCTn and HZAyDCCn bits to 1 is prohibited.

| HZAyDCFn | High-impedance output status flag   |
|----------|---|
| 0        | Indicates that output of the pin is enabled.<br><ul style="list-style-type: none"> <li>• This bit is cleared to 0 when the HZAyDCEn bit = 0.</li> <li>• This bit is cleared to 0 when the HZAyDCCn bit = 1.</li> </ul>  |
| 1        | Indicates that the pin goes into a high-impedance state.<br><ul style="list-style-type: none"> <li>• This bit is set to 1 when the HZAyDCTn bit = 1.</li> <li>• This bit is set to 1 when an edge indicating abnormality is input to the external pin<sup>Note</sup> (which is detected according to the setting of the HZAyDCNn and HZAyDCPn bits).</li> </ul> |

- Note**
- V850E/IF3
    - HZA0CTL0: TOB0OFF pin,      HZA0CTL1: TOA2OFF pin,
    - HZA1CTL0: TOB1OFF pin,
    - HZA2CTL0: ANI00/ANI05 pin,      HZA2CTL1: ANI00/ANI05 pin,
    - HZA3CTL0: ANI10 to ANI12, ANI15 to ANI17 pins,
    - HZA3CTL1: ANI10 to ANI12, ANI15 to ANI17 pins
  - V850E/IG3
    - HZA0CTL0: TOB0OFF pin,      HZA0CTL1: TOA2OFF pin,
    - HZA1CTL0: TOB1OFF pin,      HZA1CTL1: TOA3OFF pin,
    - HZA2CTL0: ANI00/ANI05 pin,      HZA2CTL1: ANI00/ANI05 pin,
    - HZA3CTL0: ANI10 to ANI12, ANI15 to ANI17 pins,
    - HZA3CTL1: ANI10 to ANI12, ANI15 to ANI17 pins

Figure 10-4. High-Impedance Output Controller Configuration



**(a) Setting procedure****(i) Setting of high-impedance control operation**

- <1> Set the HZAyDCMn, HZAyDCNn, and HZAyDCPn bits.
- <2> Set the HZAyDCEn bit to 1 (enable high-impedance control).

**(ii) Changing setting after enabling high-impedance control operation**

- <1> Clear the HZAyDCEn bit to 0 (to stop the high-impedance control operation).
- <2> Change the setting of the HZAyDCMn, HZAyDCNn, and HZAyDCPn bits.
- <3> Set the HZAyDCEn bit to 1 (to enable the high-impedance control operation again).

**(iii) Resuming output when pins are in high-impedance state**

If the HZAyDCMn bit is 1, set the HZAyDCCn bit to 1 to clear the high-impedance state after the valid edge of the external pin<sup>Note</sup> is detected. However, the high-impedance state cannot be cleared unless this bit is set while the input level of the external pin<sup>Note</sup> is inactive.

- <1> Set the HZAyDCCn bit to 1 (command signal to clear the high-impedance state).
- <2> Read the HZAyDCFn bit and check the flag status.
- <3> Return to <1> if the HZAyDCFn bit is 1. The input level of the external pin<sup>Note</sup> must be checked. The pin can function as an output pin if the HZAyDCFn bit is 0.

**(iv) To make the pin to go into a high-impedance state by software**

The HZAyDCTn bit must be set to 1 by software to make the pin to go into a high-impedance state while the input level of the external pin<sup>Note</sup> is inactive. The following procedure is an example in which the setting is not dependent upon the setting of the HZAyDCMn bit.

- <1> Set the HZAyDCTn bit to 1 (high-impedance output command).
- <2> Read the HZAyDCFn bit to check the flag status.
- <3> Return to <1> if the HZAyDCFn bit is 0. The input level of the external pin<sup>Note</sup> must be checked. The pin is in a high-impedance state if the HZAyDCFn bit is 1.

However, if the external pin<sup>Note</sup> is not used with the HZAyDCPn bit and HZAyDCNn bit cleared to 0, the pin goes into a high-impedance state when the HZAyDCTn bit is set to 1.

**Note** • V850E/IF3

HZA0CTL0: TOB0OFF pin,            HZA0CTL1: TOA2OFF pin,  
 HZA1CTL0: TOB1OFF pin,  
 HZA2CTL0: ANI00/ANI05 pin,       HZA2CTL1: ANI00/ANI05 pin,  
 HZA3CTL0: ANI10 to ANI12, ANI15 to ANI17 pins,  
 HZA3CTL1: ANI10 to ANI12, ANI15 to ANI17 pins

## • V850E/IG3

HZA0CTL0: TOB0OFF pin,            HZA0CTL1: TOA2OFF pin,  
 HZA1CTL0: TOB1OFF pin,            HZA1CTL1: TOA3OFF pin,  
 HZA2CTL0: ANI00/ANI05 pin,       HZA2CTL1: ANI00/ANI05 pin,  
 HZA3CTL0: ANI10 to ANI12, ANI15 to ANI17 pins,  
 HZA3CTL1: ANI10 to ANI12, ANI15 to ANI17 pins

## 10.4 Operation

### 10.4.1 System outline

#### (1) Outline of 6-phase PWM output

The 6-phase PWM output mode is used to generate a 6-phase PWM output waveform, by using TABn and the TABn option in combination.

The 6-phase PWM output mode is enabled by setting the TABnCTL1.TABnMD2 to TABnCTL1.TABnMD0 bits of TABn to "111".

One 16-bit counter and four 16-bit compare registers of TABn are used to generate a basic 3-phase wave.

The functions of the compare registers are as follows.

TAA<sub>n</sub> can perform a tuning operation with TABn to start a conversion trigger source for A/D converters 0 and 1.

**Remark** n = 0, 1

| Compare Register  | Function                           | Settable Range    |
|-------------------|------------------------------------|-------------------|
| TABnCCR0 register | Setting of cycle                   | 0002H ≤ m ≤ FFEEH |
| TABnCCR1 register | Specifying output width of phase U | 0000H ≤ i ≤ m + 1 |
| TABnCCR2 register | Specifying output width of phase V | 0000H ≤ j ≤ m + 1 |
| TABnCCR3 register | Specifying output width of phase W | 0000H ≤ k ≤ m + 1 |

**Remark** m = Set value of TABnCCR0 register

i = Set value of TABnCCR1 register

j = Set value of TABnCCR2 register

k = Set value of TABnCCR3 register

A dead-time interval is generated from the basic 3-phase wave generated by using three 10-bit dead-time counters and one compare register to create a wave with a reverse phase to that of the basic 3-phase wave. Then a 6-phase PWM output waveform (U,  $\bar{U}$ , V,  $\bar{V}$ , W, and  $\bar{W}$ ) is generated.

The 16-bit counter for generating the basic 3-phase wave counts up or down. After the operation has been started, this counter counts up. When its count value matches the cycle set to the TABnCCR0 register, the counter starts counting down. When the count value matches 0001H, the counter counts up again. This means that a value two times higher than the value set to the TABnCCR0 register +1 is the carrier cycle.

10-bit dead-time counters 1 to 3 that generate the dead-time interval count up. Therefore, the value set to the TABn dead-time compare register (TABnDTC) is used as a dead-time value as is. Because three counters are used, dead time can be generated independently in phases U, V, and W. However, because there is only one register that specifies a dead-time value (TABnDTC), the same dead-time value is used in the three phases.

Figure 10-5. Outline of 6-Phase PWM Output Mode

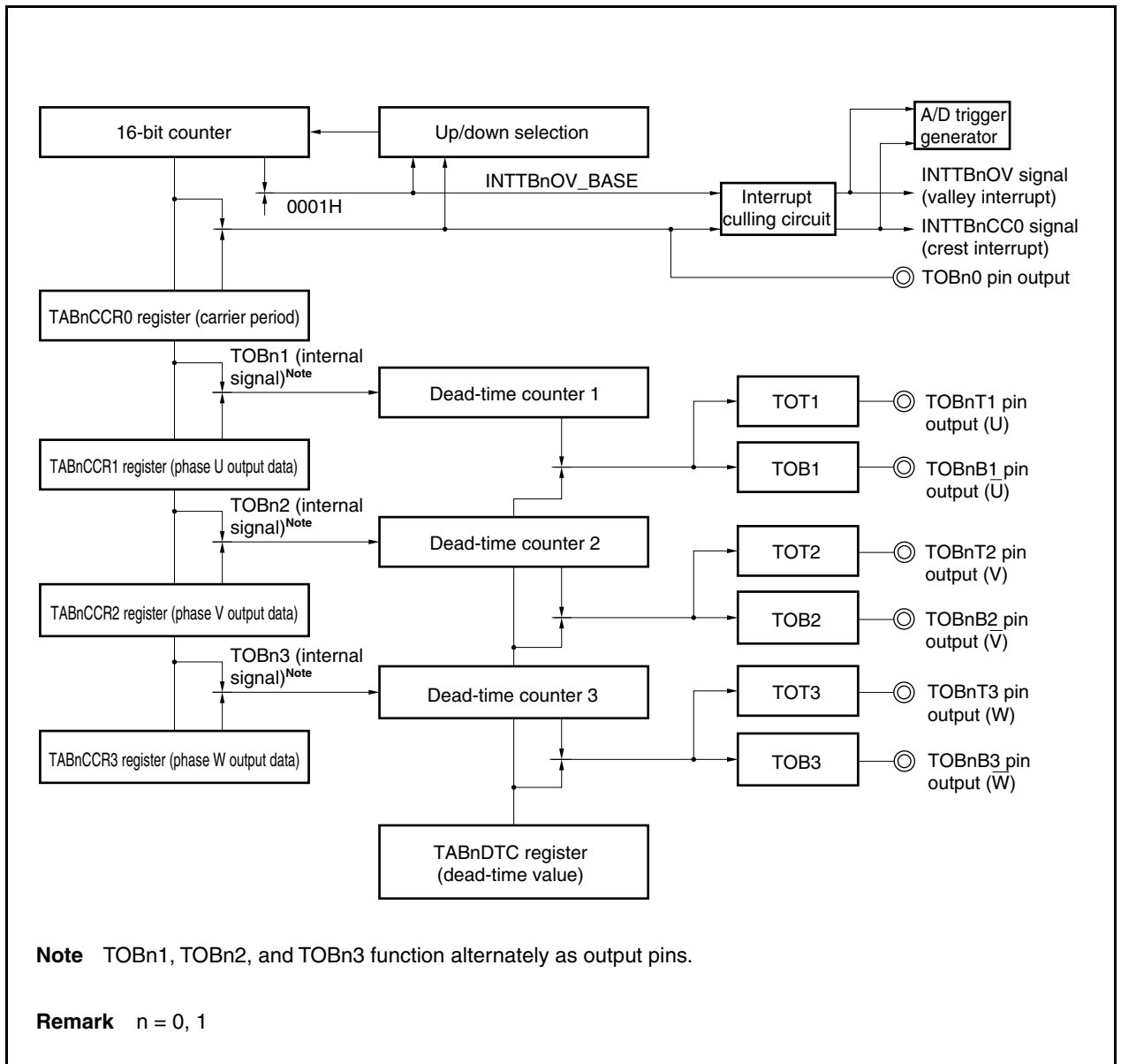
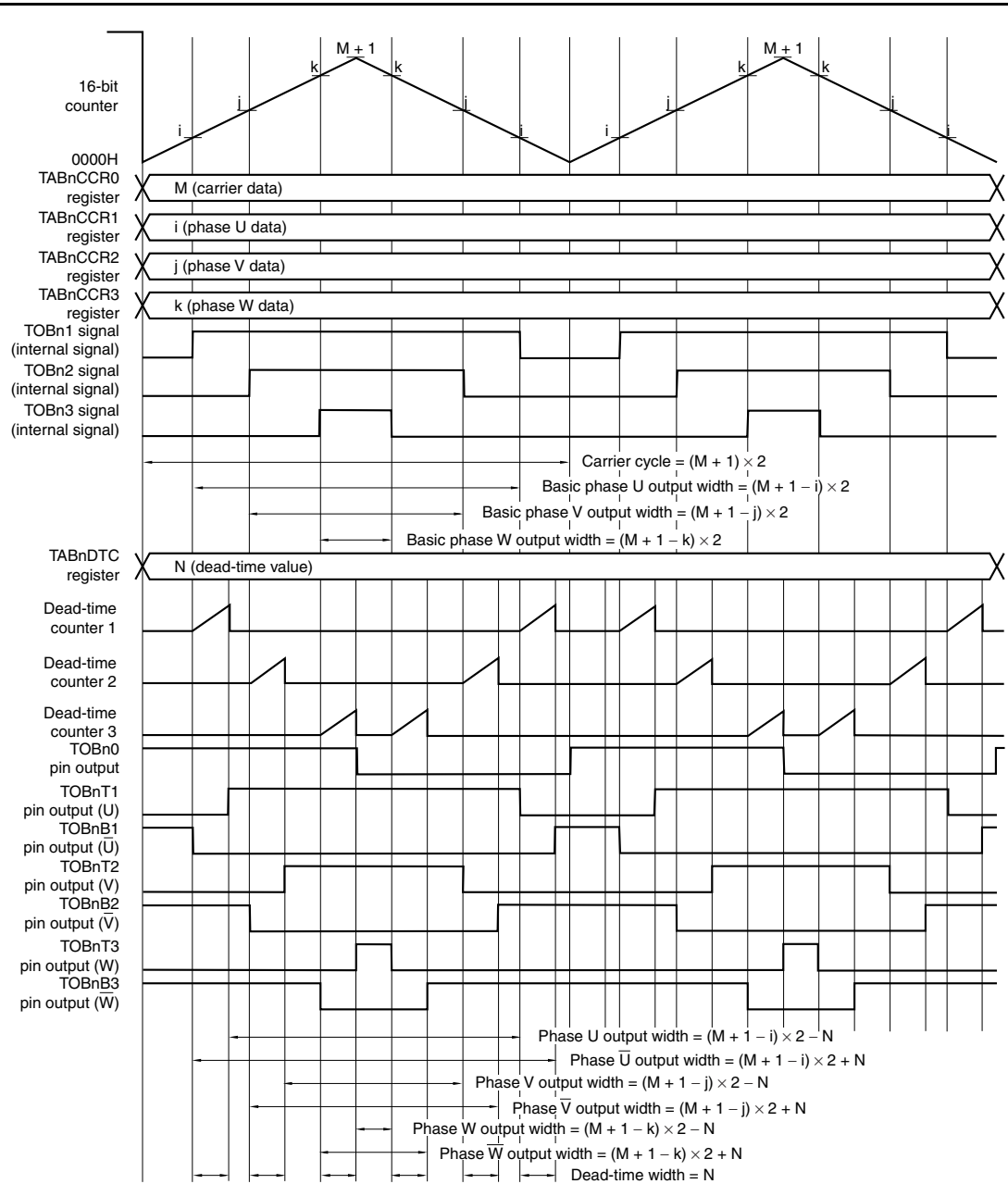


Figure 10-6. Timing Chart of 6-Phase PWM Output Mode



- Cautions**
1. Set the value "M" of the TABnCCR0 register in a range of  $0002H \leq M \leq FFEH$  in the 6-phase PWM output mode.
  2. Only a value of up to "M + 1" can be set to the TABnCCR1, TABnCCR2, and TABnCCR3 registers.
  3. The output is 100% if "0000H" is set to the TABnCCR1, TABnCCR2, and TABnCCR3 registers. The output is 0% if "M + 1" is set to the TABnCCR1, TABnCCR2, and TABnCCR3 registers. The output (duty 50%) rises at the crest (M + 1) of the 16-bit counter and falls at the valley (0000H) if "M + 2" or higher is set to the TABnCCR1, TABnCCR2, and TABnCCR3 registers.
  4. If the value calculated using an equation for the output width of the positive phase side of the U, V, or W phase (such as  $((M + 1 - i) \times 2 - N)$ ) is 0 or less, the output converges at 0 (0%). If the range of the calculated value is from  $((M + 1) \times 2 - N)$  to  $((M + 1) \times 2)$ , the output converges at  $((M + 1) \times 2)$  (100%).

**Remark** n = 0, 1

<R>

**(2) Interrupt requests**

Two types of interrupt requests are available: the INTTBnCC0 (crest interrupt) signal and INTTBnOV (valley interrupt) signal.

The INTTBnCC0 and INTTBnOV signals can be culled by using the TABnOPT1 register.

For details of culling interrupts, see **10.4.3 Interrupt culling function**.

- INTTBnCC0 (crest interrupt) signal: Interrupt signal indicating matching between the value of the 16-bit counter that counts up and the value of the TABnCCR0 register
- INTTBnOV (valley interrupt) signal: Interrupt signal indicating matching between the value of the 16-bit counter that counts down and the value 0001H

**(3) Rewriting registers during timer operation**

The following registers have a buffer register and can be rewritten in the anytime rewriting mode, batch rewrite mode, or intermittent batch rewrite mode.

| Related Unit    | Register   |
|-----------------|--|
| Timer AAn       | TAAAn capture/compare register 0 (TAAAnCCR0)<br>TAAAn capture/compare register 1 (TAAAnCCR1)   |
| Timer ABn       | TABn capture/compare register 0 (TABnCCR0)<br>TABn capture/compare register 1 (TABnCCR1)<br>TABn capture/compare register 2 (TABnCCR2)<br>TABn capture/compare register 3 (TABnCCR3) |
| Timer Qn option | TABn option register 1 (TABnOPT1)  |

**Remark** n = 0, 1

For details of the transfer function of the compare register, see **10.4.4 Operation to rewrite register with transfer function**.

**(4) Counting-up/-down operation of 16-bit counter**

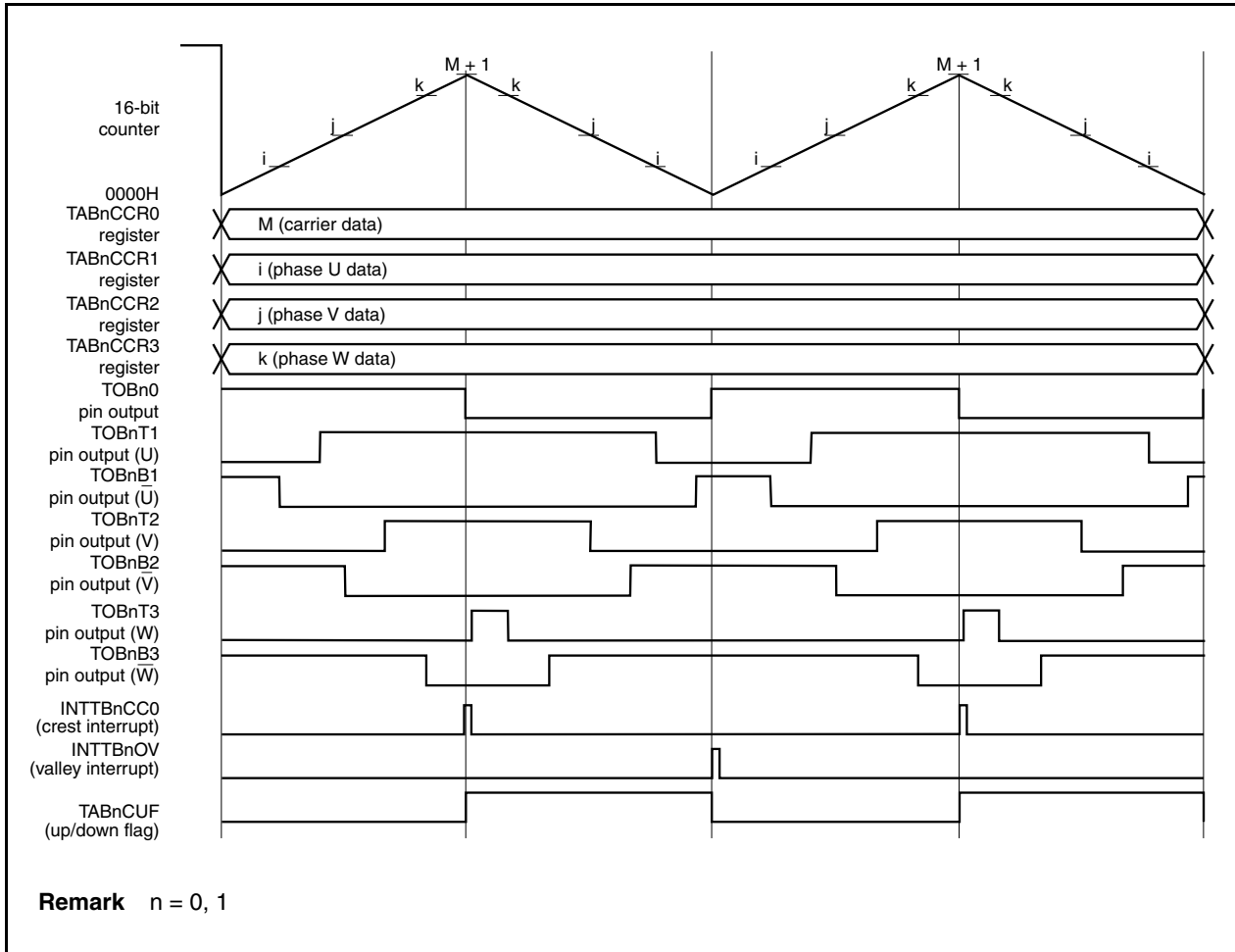
The operation status of the 16-bit counter can be checked by using the TABnCUF bit of TABn option register 0 (TABnOPT0).

| Status of TABnCUF Bit | Status of 16-Bit Counter | Range of 16-Bit Counter Value |
|-----------------------|--------------------------|-------------------------------|
| TABnCUF bit = 0       | Counting up              | 0000H – m                     |
| TABnCUF bit = 1       | Counting down            | (m+1) – 0001H                 |

**Remarks** 1. m = Set value of TABnCCR0 register

2. n = 0, 1

Figure 10-7. Interrupt and Up/Down Flag



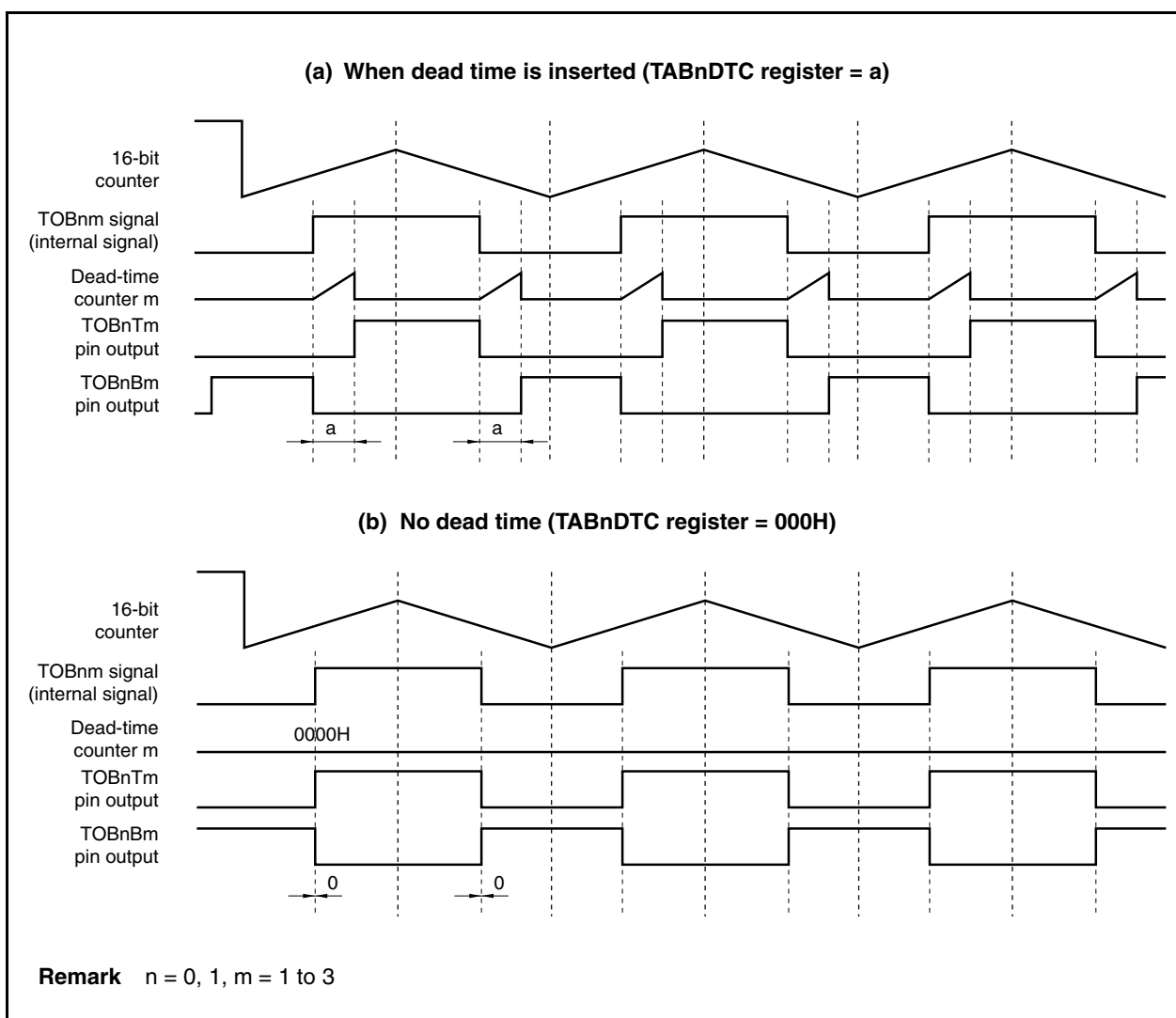


10.4.2 Dead-time control (generation of negative-phase wave signal)

(1) Dead-time control mechanism

In the 6-phase PWM output mode, compare registers 1 to 3 (TABnCCR1, TABnCCR2, and TABnCCR3) are used to set the duty factor, and compare register 0 (TABnCCR0) is used to set the cycle. By setting these four registers and by starting the operation of TAB, three types of PWM output waves (basic 3-phase waves) with a variable duty factor are generated. These three PWM output waves are input to the timer Qn option unit (TMQOPn) and their inverted signal with dead-time is created to generate three sets of (six) PWM waves. The TMQOPn unit consists of three 10-bit counters (dead-time counters 1 to 3) that operate in synchronization with the count clock of TABn, and a TABn dead-time compare register (TABnDTC) that specifies dead time. If "a" is set to the TABnDTC register, the dead-time value is "a", and interval "a" is created between a positive-phase wave and a negative-phase wave.

Figure 10-8. PWM Output Waveform with Dead Time (1)



**(2) PWM output of 0%/100%**

The V850E/IF3 and V850E/IG3 are capable of 0% waveform output and 100% waveform output for PWM output.

A low level is continuously output from TOBnTm pin as the 0% waveform output. A high level is continuously output from TOBnTm pin as the 100% waveform output.

The 0% waveform is output by setting the TABnCCRm register to “M + 1” when the TABnCCR0 register = M.

The 100% waveform is output by setting the TABnCCRm register to “0000H”.

Rewriting the TABnCCRm register is enabled while the timer is operating, and 0% waveform output or 100% waveform output can be selected at the point of the crest interrupt (INTTBnCC0) and valley interrupt (INTTBnOV).

**Remark** n = 0, 1, m = 1 to 3

**Figure 10-9. 0% PWM Output Waveform (With Dead Time)**

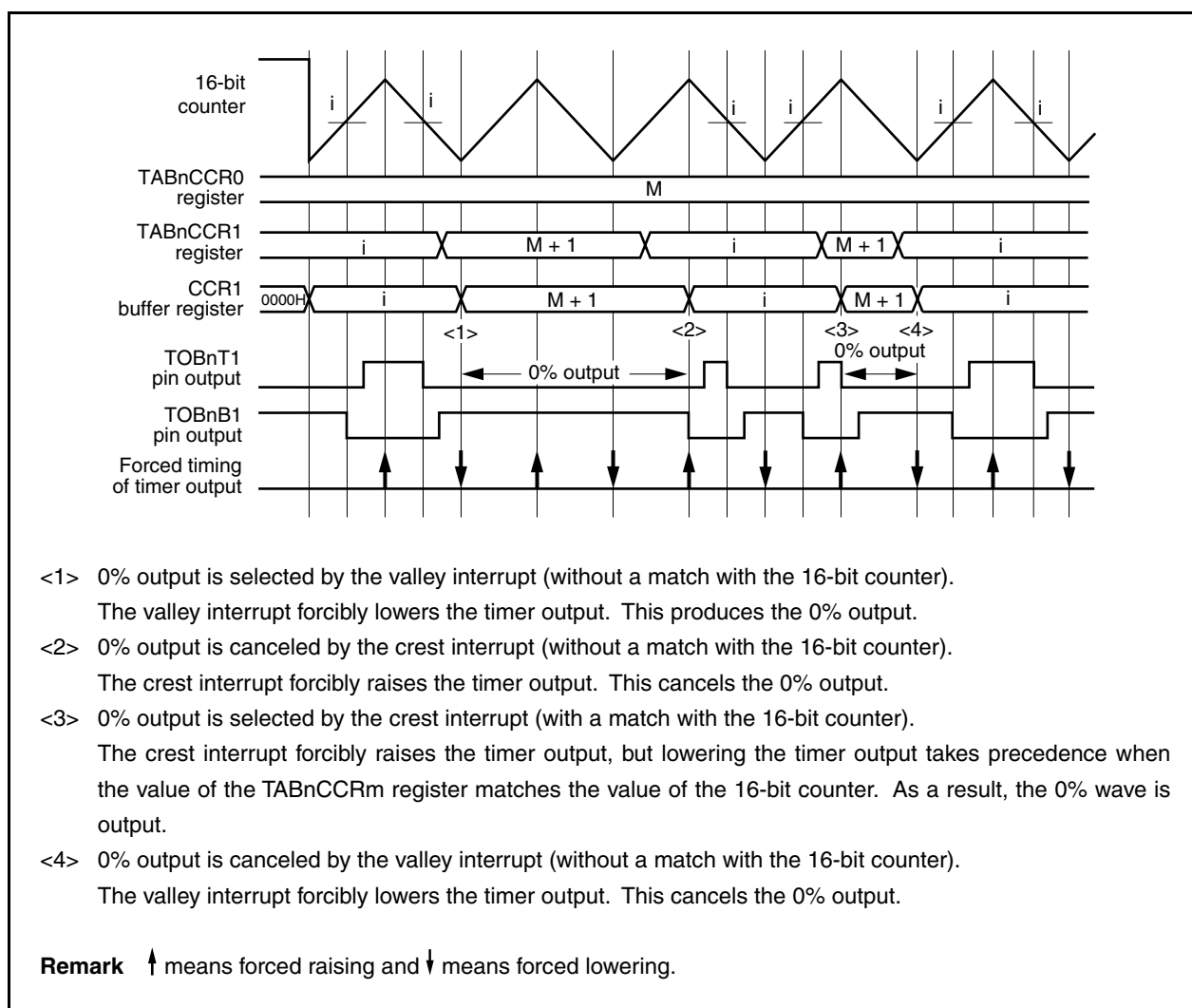


Figure 10-10. 100% PWM Output Waveform (With Dead Time)

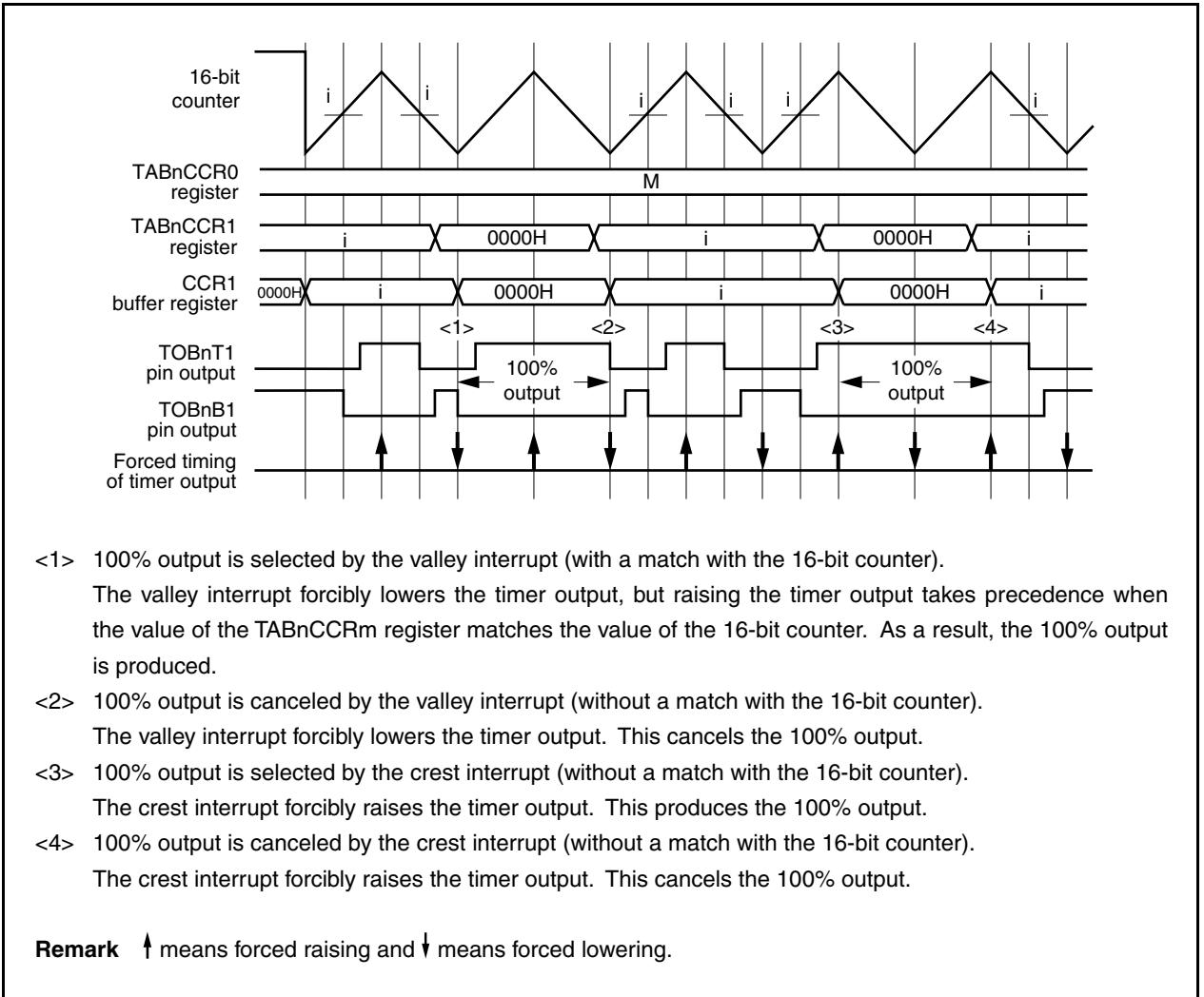
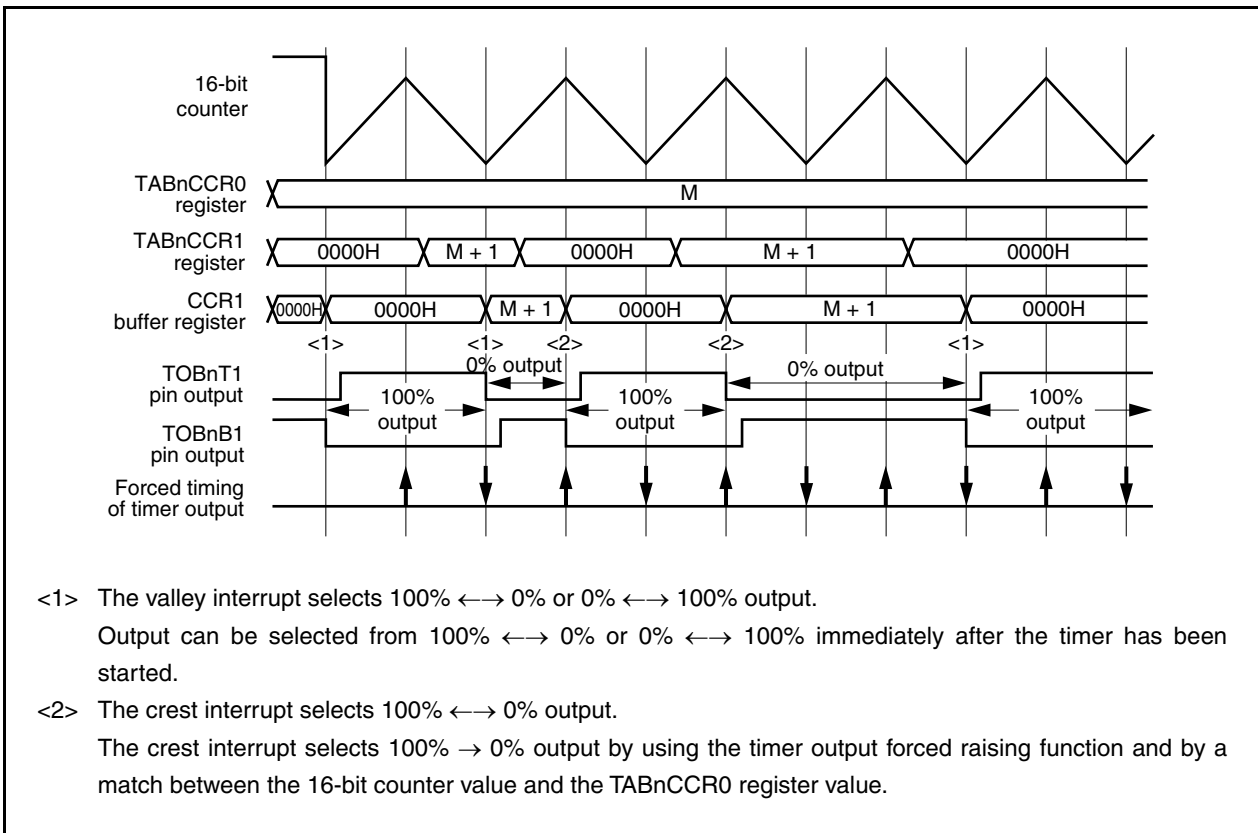


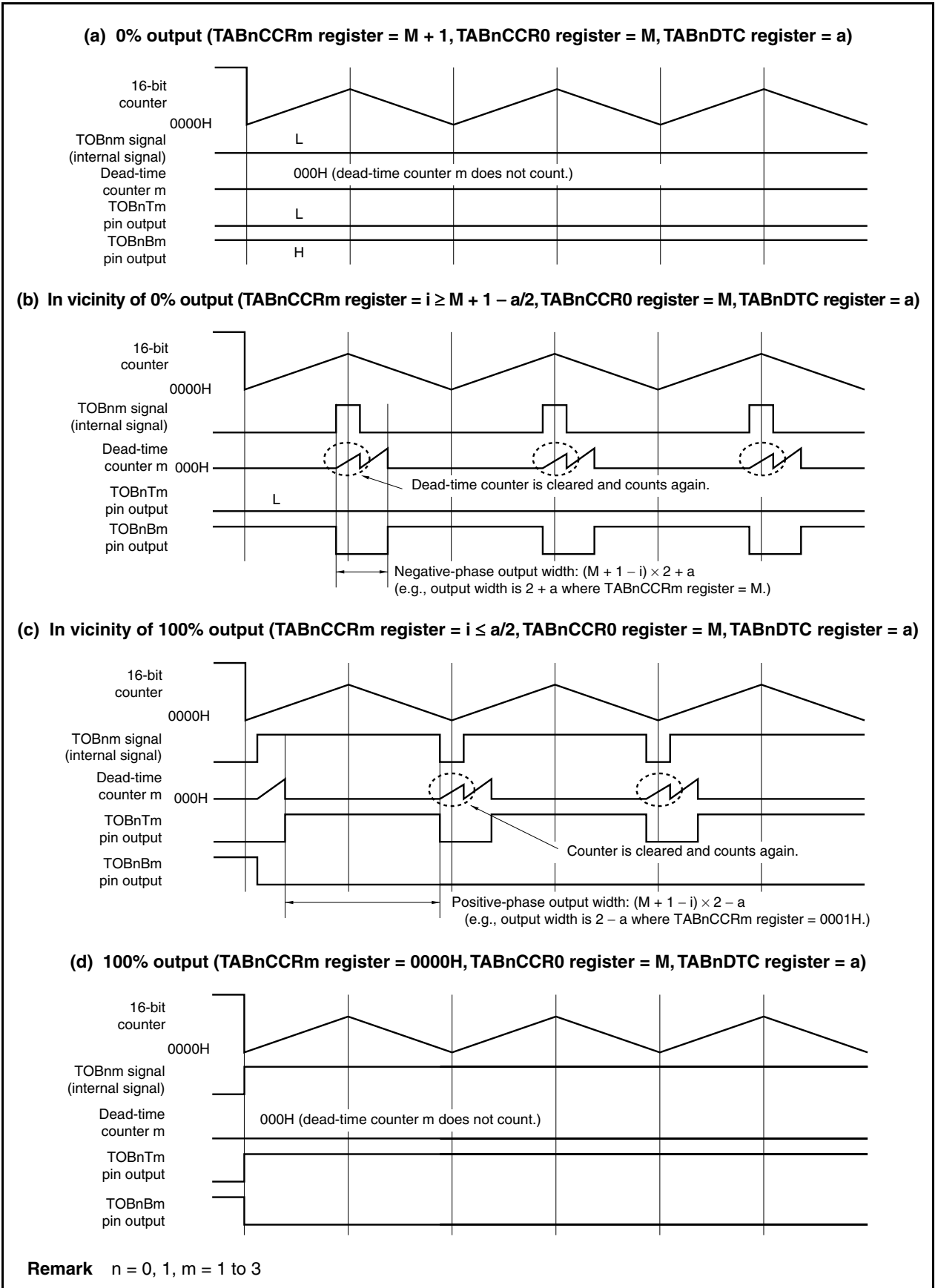
Figure 10-11. PWM Output Waveform from 0% to 100% and from 100% to 0% (With Dead Time)



**(3) Output wave in vicinity of 0% and 100% output**

If an interrupt is generated because the value of the 16-bit counter matches the value of the compare register while dead time is being counted, the dead-time counter is cleared and starts its count operation again. The output waveform of dead-time control in the vicinity of 0% and 100% output is shown below.

Figure 10-12. PWM Output Waveform with Dead Time (2)



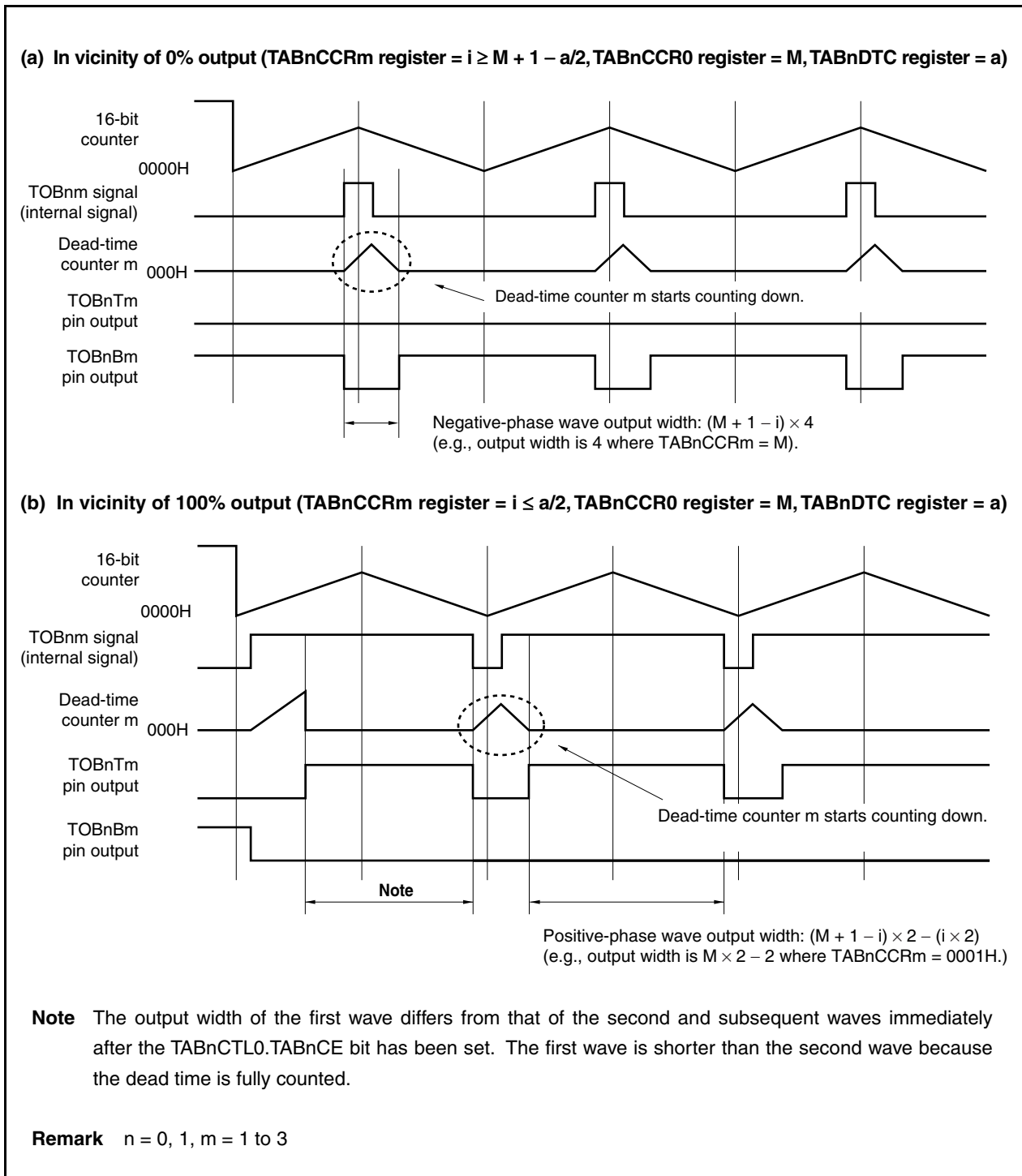
**(4) Automatic dead-time width narrowing function (TABnOPT2.TABnDTM bit = 1)**

The dead-time width can be automatically narrowed in the vicinity of 0% output or 100% output by setting the TABnOPT2.TABnDTM bit to 1.

By setting the TABnDTM bit to 1, the dead-time counter is not cleared, but starts down counting if the TOBnm (internal signal) output of timer AB changes during dead-time counting.

The following timing chart shows the operation of the dead-time counter when the TABnDTM bit is set to 1.

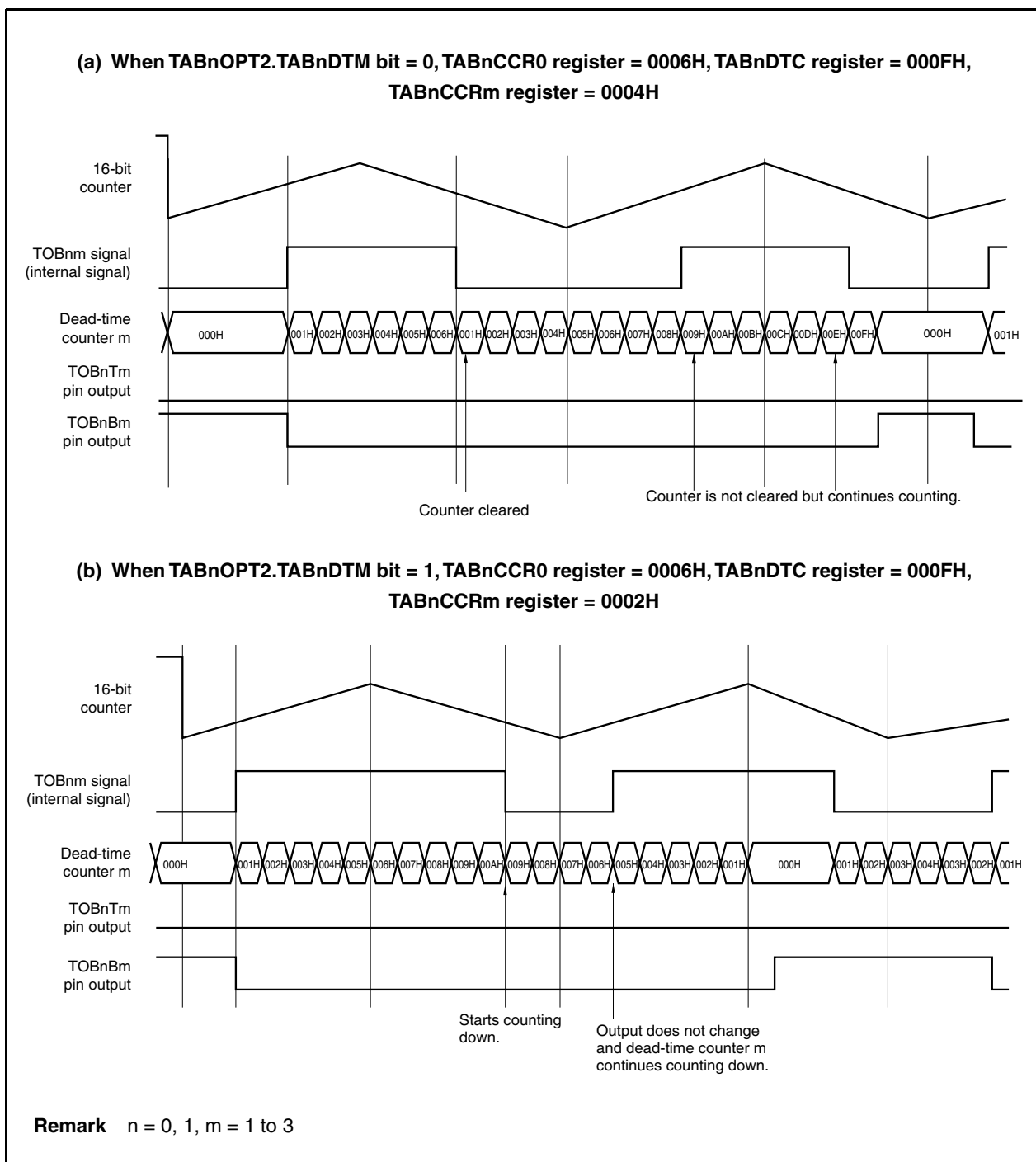
**Figure 10-13. Operation of Dead-Time Counter m (1)**



**(5) Dead-time control in case of incorrect setting**

Usually, the TOBnm (internal signal) output of TABn changes only once during dead-time counting, only in the vicinity of 0% and 100% output. This section shows an example where the TABnCCR0 register (carrier cycle) and TABnDTC register (dead-time value) are incorrectly set. If these registers are incorrectly set, the TOBnm (internal signal) output of TABn changes more than once during dead-time counting. The following flowchart shows the 6-phase PWM output waveform in this case.

**Figure 10-14. Operation of Dead-Time Counter m (2)**



### 10.4.3 Interrupt culling function

- The interrupts to be culled are INTTBnCC0 (crest interrupt) and INTTBnOV (valley interrupt).
- The TABnOPT1.TABnICE bit is used to enable output of the INTTBnCC0 interrupt and specify the count signal for interrupt culling.
- The TABnOPT1.TABnIOE bit is used to enable output of the INTTBnOV interrupt and specify the count signal for interrupt culling.
- The TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits are used to specify the number of interrupts to be culled, specified for the count signals for interrupt culling.

The interrupts are masked for the specified number of culling counts and an interrupt occurs at the next interrupt timing.

- The TABnOPT2.TABnRDE bit is used to specify whether transfer is to be culled or not.  
If it is specified that transfer is to be culled, transfer is executed at the same timing as the interrupt output after culling. If it is specified that transfer is not to be culled, transfer is executed at the transfer timing after the TABnCCR1 register has been written.
- The TABnOPT0.TABnCMS bit is used to specify whether the registers with a transfer function are batch rewritten or anytime rewritten.

The values of the registers are updated in synchronization with transferring when the TABnCMS bit is 0. When the TABnCMS bit is 1, the values of the registers are immediately updated when a new value is written to the registers.

Transfer is performed from the TABnCCRm register to the CCRm buffer register in synchronization with interrupt culling timing.

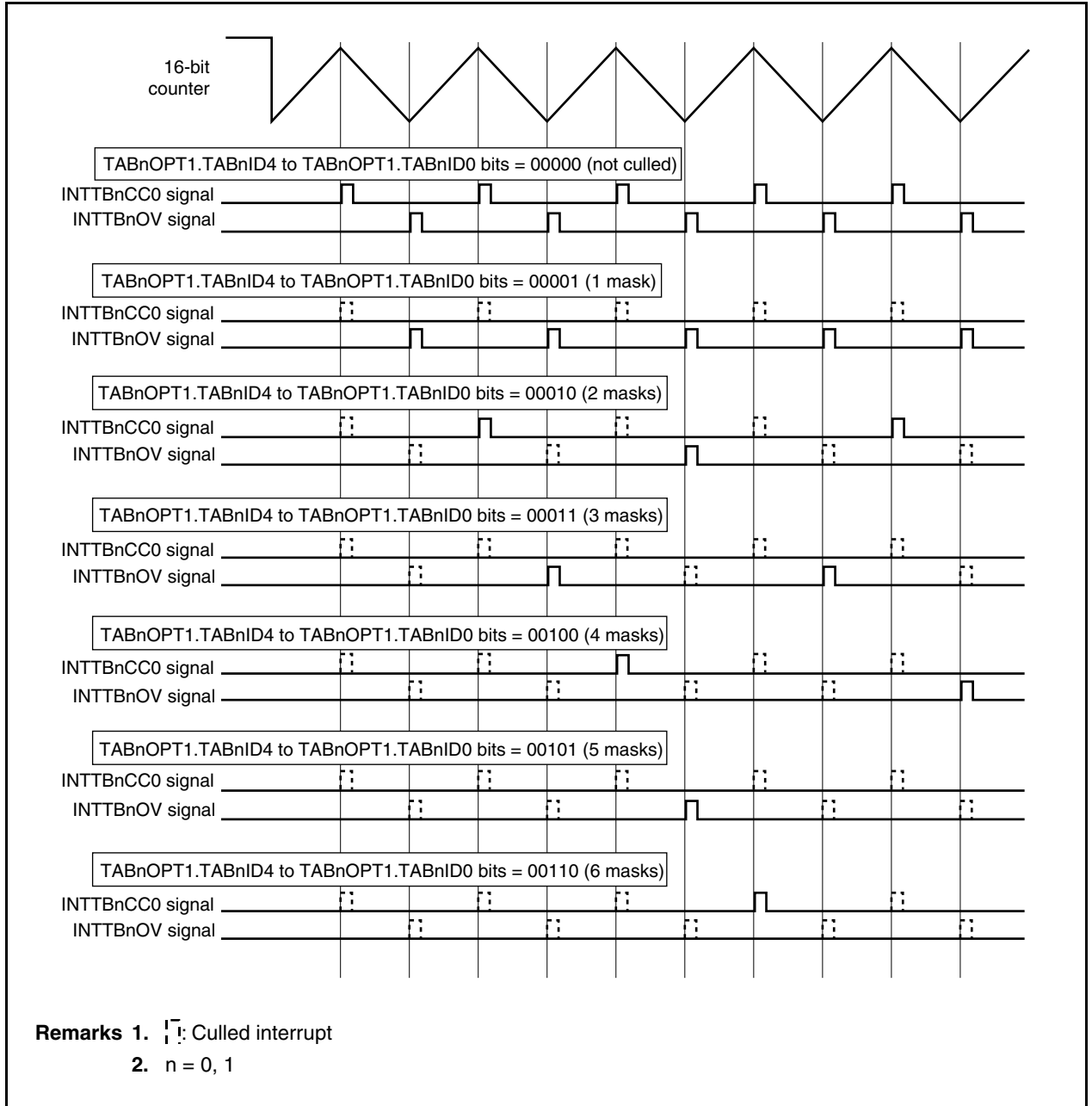
**Cautions 1. When using the interrupt culling function in the batch rewrite mode (transfer mode), execute the function in the intermittent batch rewrite mode (transfer culling mode).**

**2. An interrupt is generated at the timing after culling.**

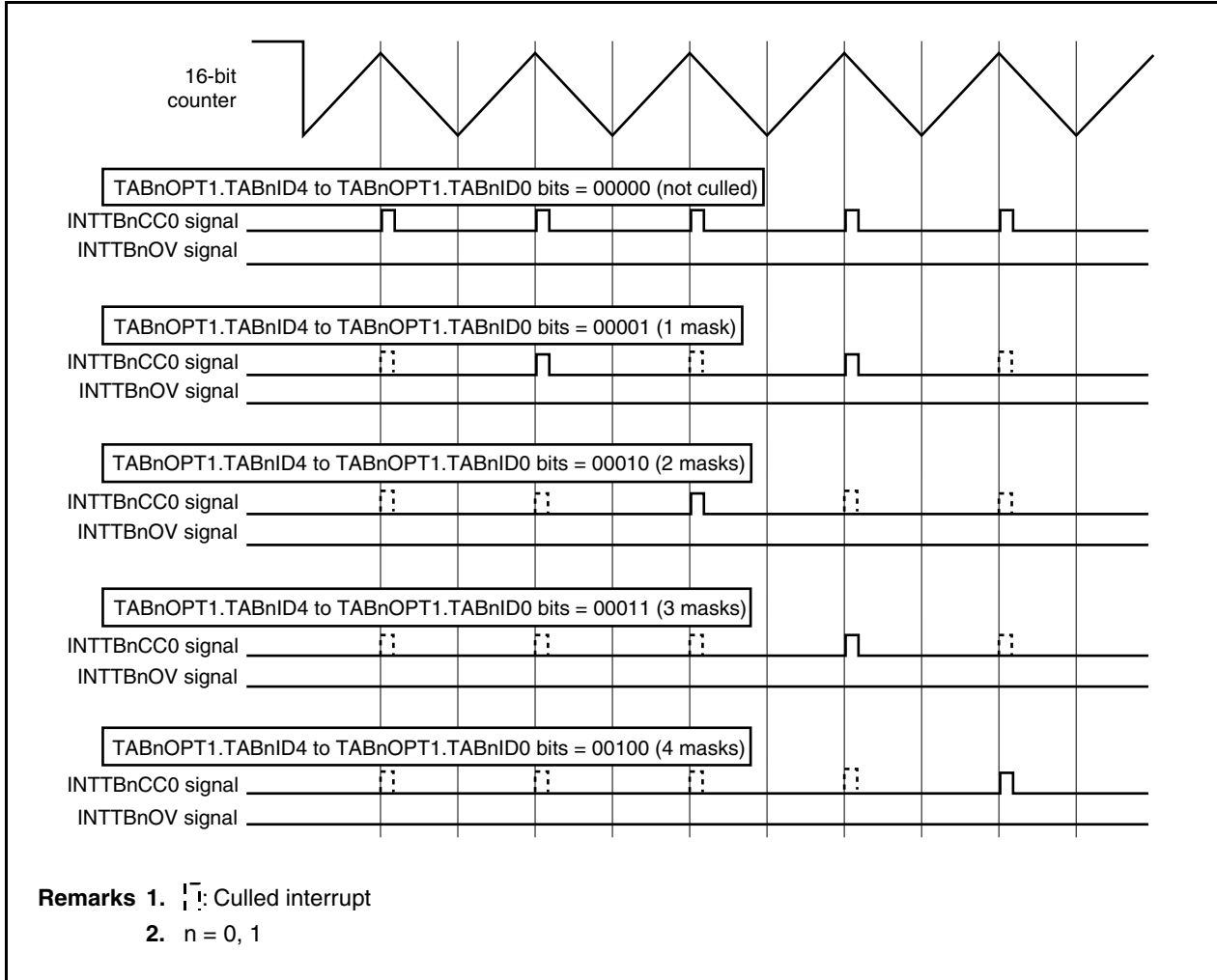


(1) Interrupt culling operation

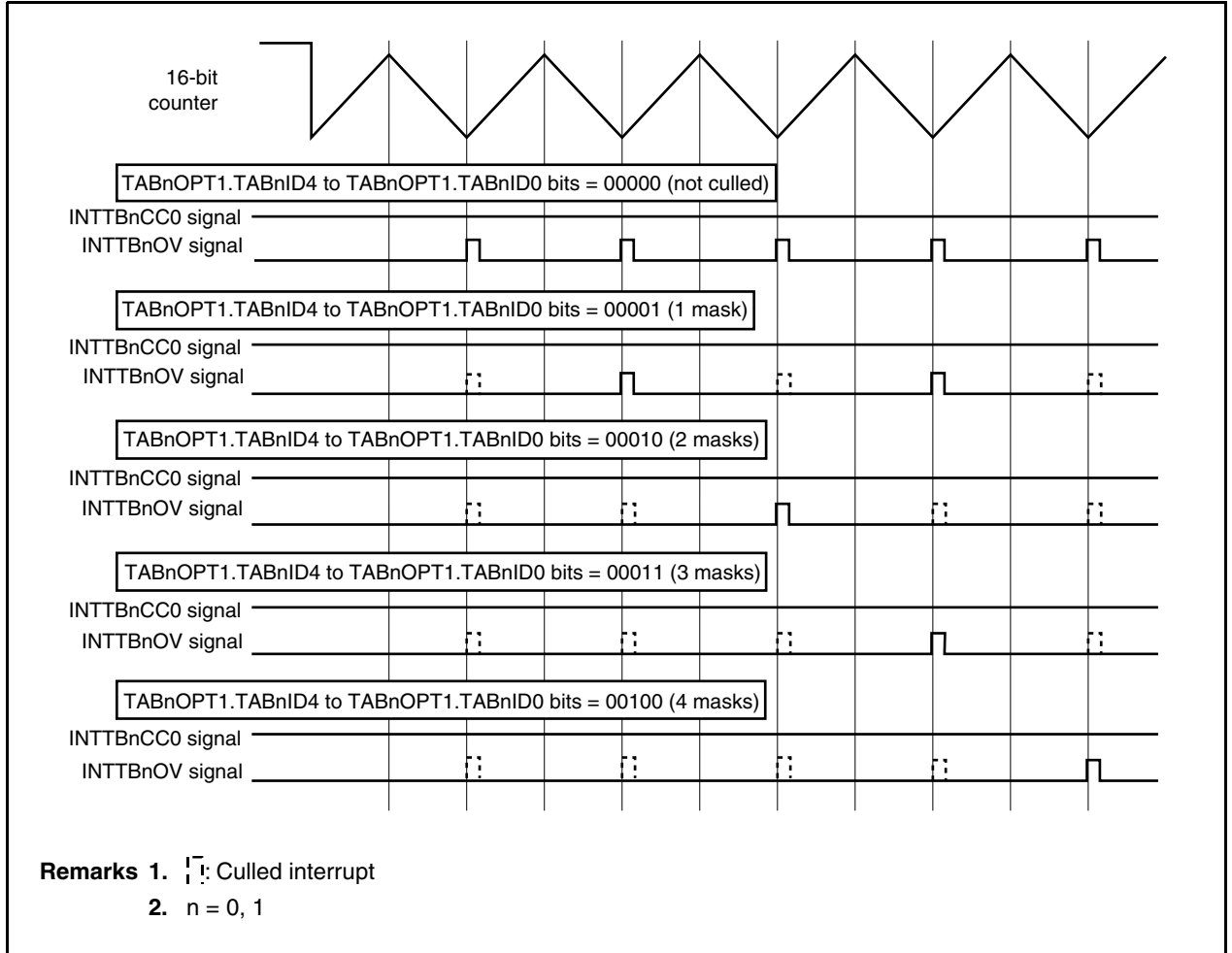
Figure 10-15. Interrupt Culling Operation When TABnOPT1.TABnICE Bit = 1, TABnOPT1.TABnIOE Bit = 1, TABnOPT2.TABnRDE Bit = 1 (Crest/Valley Interrupt Output)



**Figure 10-16. Interrupt Culling Operation When TABnOPT1.TABnICE Bit = 1, TABnOPT1.TABnIOE Bit = 0, TABnOPT2.TABnRDE Bit = 1 (Crest Interrupt Output)**



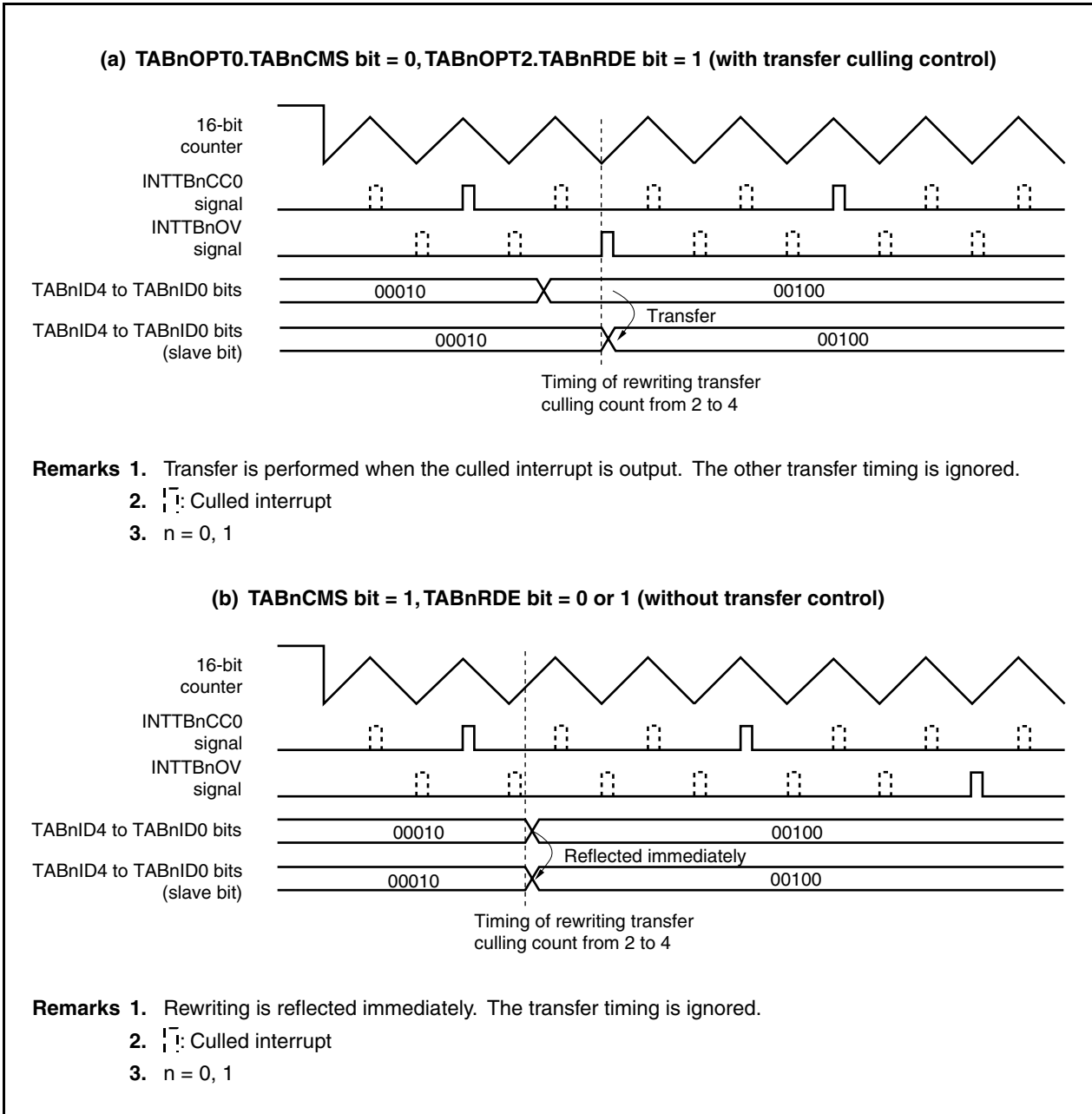
**Figure 10-17. Interrupt Culling Operation When TABnOPT1.TABnICE Bit = 0, TABnOPT1.TABnIOE Bit = 1, TABnOPT2.TABnRDE Bit = 1 (Valley Interrupt Output)**



**(2) To alternately output crest interrupt (INTTBnCC0) and valley interrupt (INTTBnOV)**

To alternately output the crest and valley interrupts, set both the TABnOPT1.TABnICE and TABnOPT1.TABnIOE bits to 1.

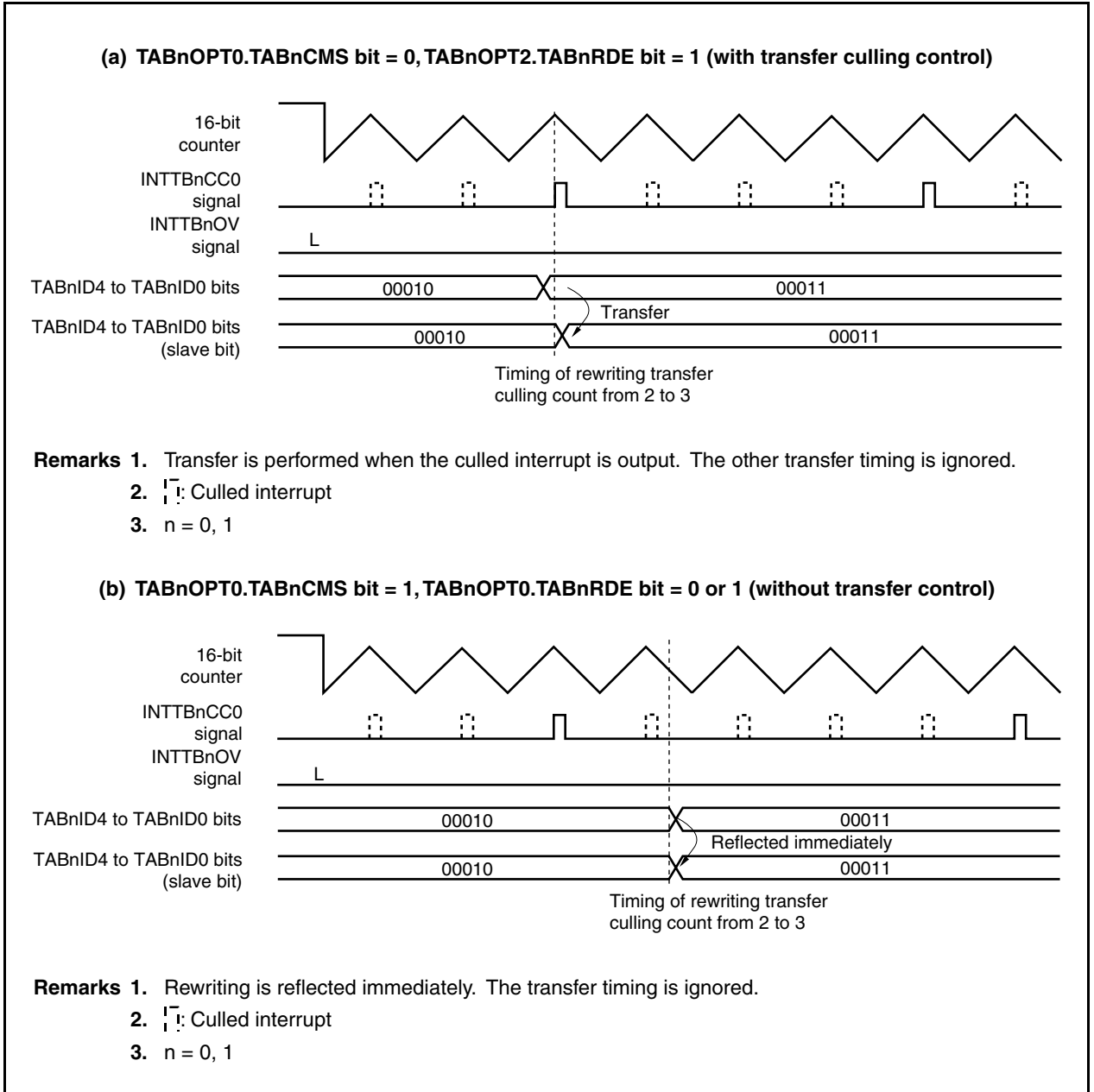
**Figure 10-18. Crest/Valley Interrupt Output**



(3) To output only crest interrupt (INTTBnCC0)

Set the TABnOPT1.TABnICE bit to 1 and set the TABnOPT1.TABnIOE bit to 0.

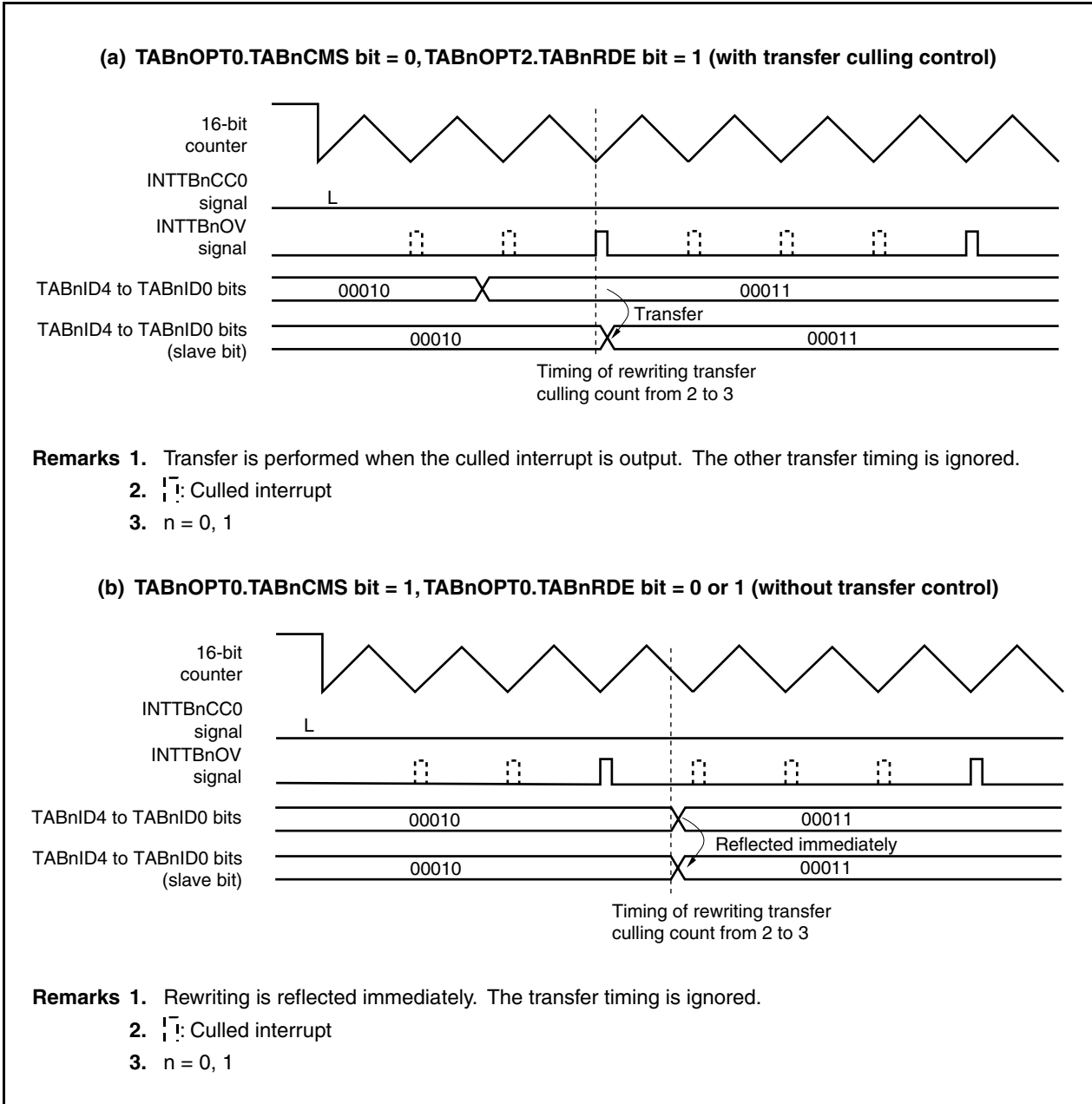
Figure 10-19. Crest Interrupt Output



(4) To output only valley interrupt (INTTBnOV)

Set the TABnOPT1.TABnICE bit to 0 and set the TABnOPT1.TABnIOE bit to 1.

Figure 10-20. Valley Interrupt Output



#### 10.4.4 Operation to rewrite register with transfer function

The following seven registers are provided with a transfer function and used to control a motor. Each of registers has a buffer register.

- TABnCCR0: Register that specifies the cycle of the 16-bit counter (TAB)
- TABnCCR1: Register that specifies the duty factor of TOBnT1 (U) and TOBnB1 ( $\bar{U}$ )
- TABnCCR2: Register that specifies the duty factor of TOBnT2 (V) and TOBnB2 ( $\bar{V}$ )
- TABnCCR3: Register that specifies the duty factor of TOBnT3 (W) and TOBnB3 ( $\bar{W}$ )
- TABnOPT1: Register that specifies the culling of interrupts
- TAAAnCCR0: Register that specifies the A/D conversion start trigger generation timing (TAAAn during tuning operation)
- TAAAnCCR1: Register that specifies the A/D conversion start trigger generation timing (TAAAn during tuning operation)

The following three rewrite modes are provided in the registers with a transfer function.

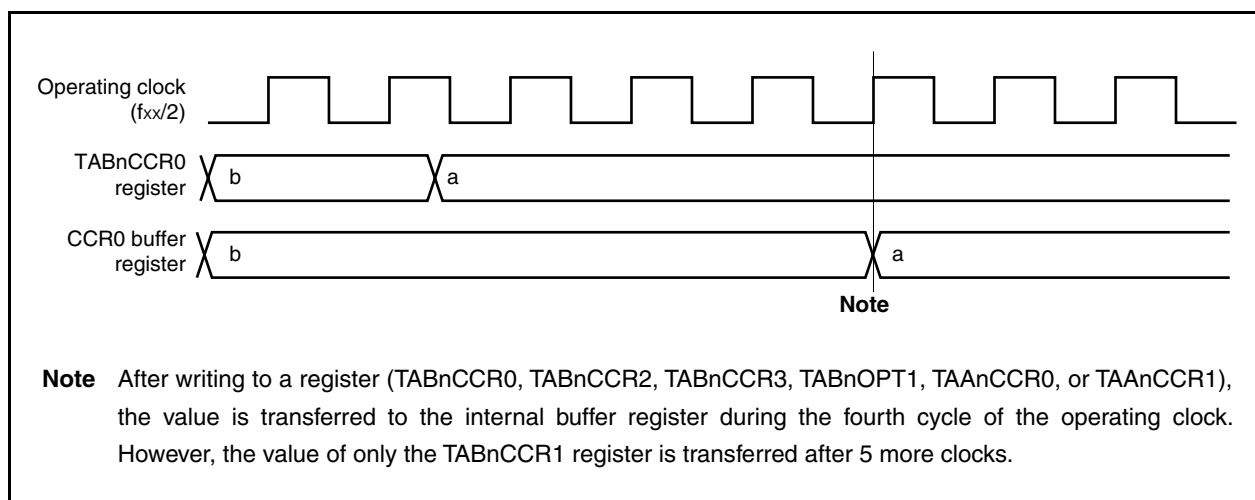
- Anytime rewriting mode  
This mode is specified by setting the TABnOPT0.TABnCMS bit to 1. The setting of the TABnOPT2.TABnRDE bit is ignored.  
In this mode, each compare register is updated independently, and the value of the compare register is updated as soon as a new value is written to it.
- Batch rewrite mode (transfer mode)  
This mode is specified by setting the TABnOPT0.TABnCMS bit to 0, the TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits to 00000, and the TABnOPT2.TABnRDE bit to 0.  
When data is written to the TABnCCR1 register, the seven registers are transferred to the buffer register all at once at the next transfer timing. Unless the TABnCCR1 register is rewritten, the transfer operation is not performed even if the other six registers are rewritten.  
The transfer timing is the timing of each crest (match between the 16-bit counter value and TABnCCR0 register value) and valley (match between the 16-bit counter value and 0001H) regardless of the interrupt.
- Intermittent batch rewrite mode (transfer culling mode)  
This mode is specified by setting the TABnOPT0.TABnCMS bit to 0 and the TABnOPT2.TABnRDE bit to 1.  
When data is written to the TABnCCR1 register, the seven registers are transferred to the buffer register all at once at the next transfer timing. Unless the TABnCCR1 register is rewritten, the transfer operation is not performed even if the other six registers are rewritten.  
If interrupt culling is specified by the TABnOPT1 register, the transfer timing is also culled as the interrupts are culled, and the seven registers are transferred all at once at the culled timing of crest interrupt (match between the 16-bit counter value and TABnCCR0 register value) or valley interrupt (match between the 16-bit counter value and 0001H).  
For details of the interrupt culling function, see **10.4.3 Interrupt culling function**.

**(1) Anytime rewriting mode**

This mode is specified by setting the TABnOPT0.TABnCMS bit is 1. The setting of the TABnOPT2.TABnRDE bit is ignored.

In this mode, the value written to each register with a transfer function is immediately transferred to an internal buffer register and compared with the count value. If a register with transfer function is rewritten in this mode after the count value of the 16-bit counter matches the value of the TABnCCRm register, the rewritten value is not reflected because the next match is ignored after the first match has occurred. If the register is rewritten during up counting, the new register value becomes valid after the counter has started counting down.

**Figure 10-21. Timing of Reflecting Rewritten Value**

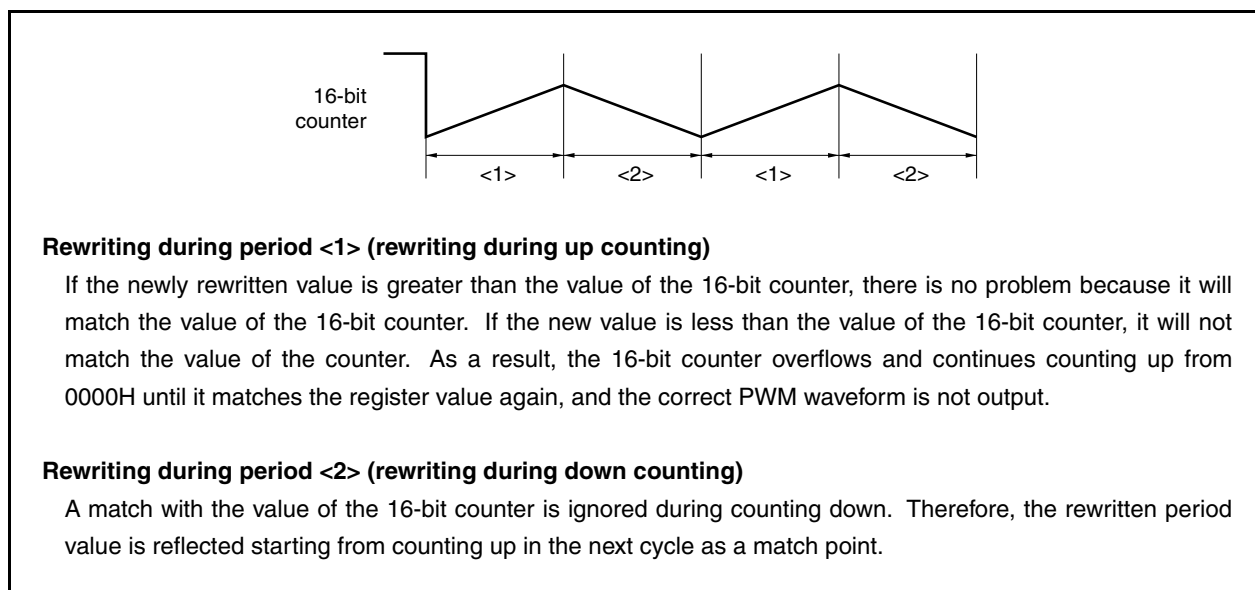


<R>

**(a) Rewriting TABnCCR0 register**

Even if the TABnCCR0 register is rewritten in the anytime rewriting mode, the new value may not be reflected in some cases.

**Figure 10-22. Example of Rewriting TABnCCR0 Register**





**(b) Rewriting TABnCCRM register**

Figure 10-24 shows the timing of rewriting before the value of the 16-bit counter matches the value of the TABnCCRM register (<1> in Figure 10-23), and Figure 10-25 shows the timing of rewriting after the value of the 16-bit counter matches the value of the TABnCCRM register (<2> in Figure 10-23).

**Figure 10-23. Basic Operation of 16-Bit Counter and TABnCCRM Register**

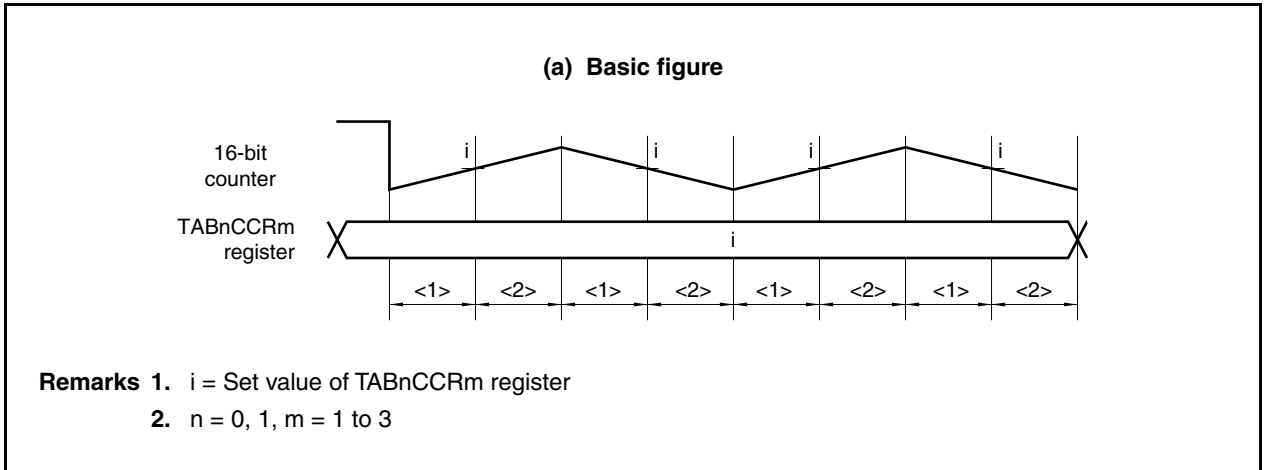
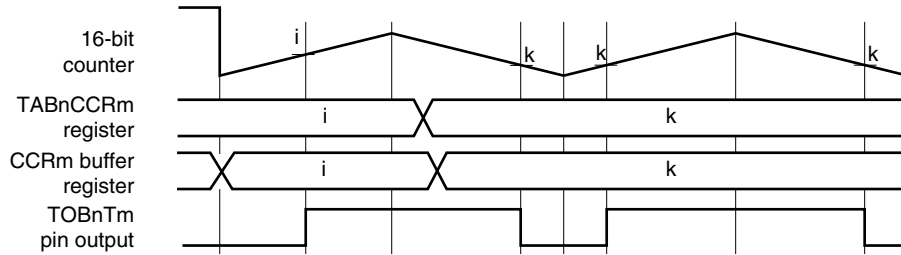


Figure 10-24. Example of Rewriting TABnCCR1 to TABnCCR3 Registers (Rewriting Before Match Occurs)

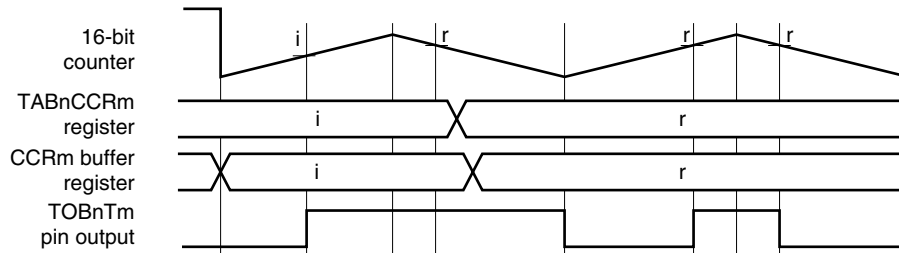
(a)

If the TABnCCRm register is rewritten before its value matches the value of the 16-bit counter, the register value will match the value of the 16-bit counter after the register has been rewritten. Consequently, the new register value is immediately reflected.



(b)

If a value less than the value of the 16-bit counter (greater if the counter is counting down) is written to the TABnCCRm register, the output waveform is as follows because the register value does not match the counter value.



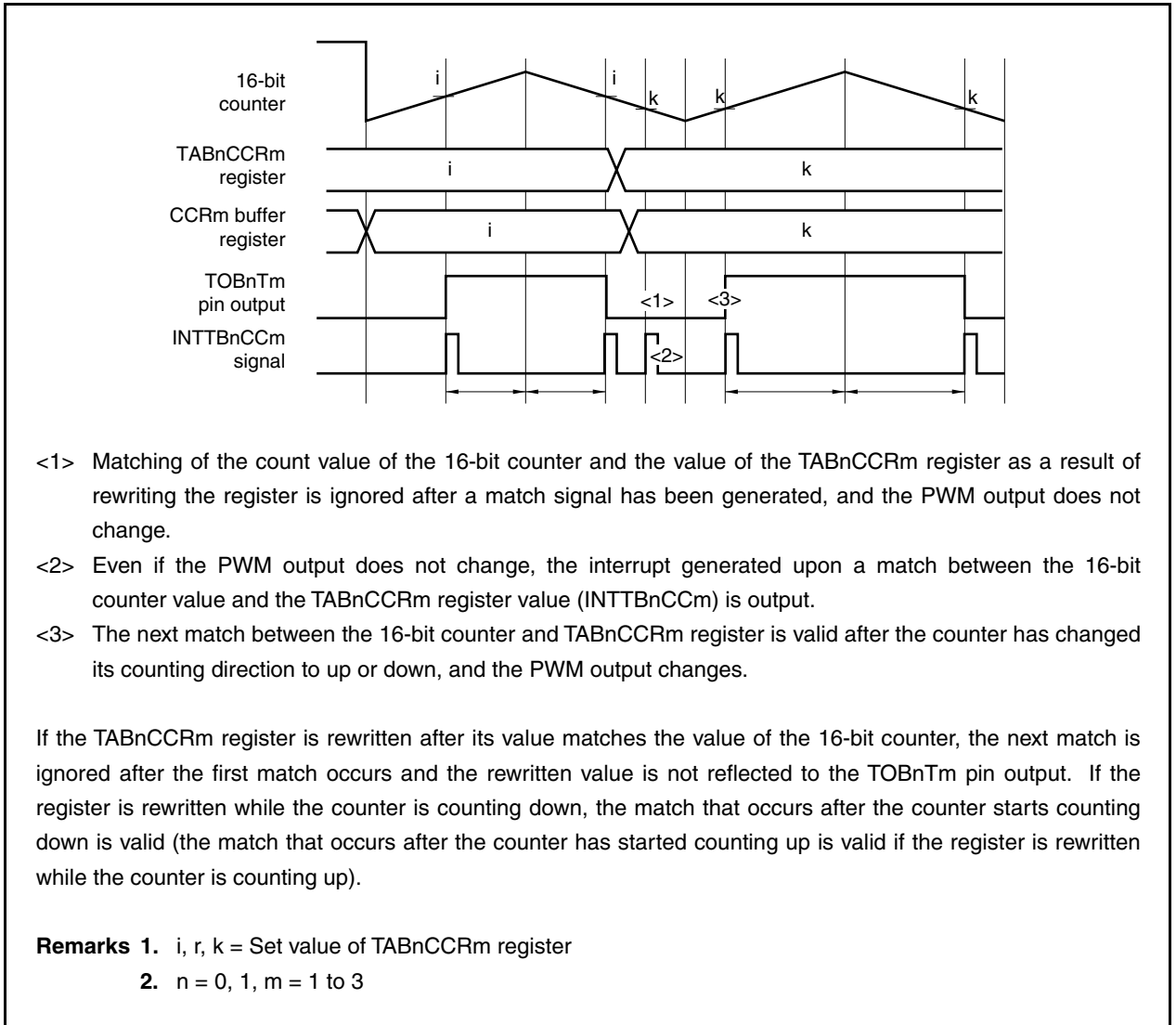
If the register value does not match the counter value, the TOBnTm pin output does not change. Even if the value of the 16-bit counter does not match the value of the TABnCCRm register, the TOBnTm pin output always changes to the high level if the crest interrupt occurs and to the low level if the valley interrupt occurs.

This is a function provided for 0% output and 100% output.

For details, see 10.4.2 (2) PWM output of 0%/100%.

- Remarks**
1. i, r, k = Set values of TABnCCRm register
  2. n = 0, 1, m = 1 to 3

Figure 10-25. Example of Rewriting TABnCCR1 to TABnCCR3 Registers (Rewriting After Match Occurs)



**(c) Rewriting TABnOPT1 register**

The interrupt culling counter is cleared when the TABnOPT1 register is written. When the interrupt culling counter has been cleared, the measured number of times the interrupt has occurred is discarded. Consequently, the interrupt generation interval is temporarily extended.

To avoid this operation, rewrite the TABnOPT1 register in the intermittent batch rewriting mode (transfer culling mode).

For details of rewriting the TABnOPT1 register, see **10.4.3 Interrupt culling function**.

**(2) Batch rewrite mode (transfer mode)**

This mode is specified by setting the TABnOPT0.TABnCMS bit to 0, the TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits to 00000, and the TABnOPT2.TABnRDE bit to 0.

In this mode, the values written to each compare register are transferred to the internal buffer register all at once at the transfer timing and compared with the count value.

**(a) Rewriting procedure**

If data is written to the TABnCCR1 register, the values set to the TABnCCR0 to TABnCCR3, TABnOPT1, TAAAnCCR0, and TAAAnCCR1 registers are transferred all at once to the internal buffer register at the next transfer timing. Therefore, write to the TABnCCR1 register last. Writing to the register is prohibited after the TABnCCR1 register has been written and before the transfer timing is generated (until the crest (match between the 16-bit counter value and TABnCCR0 register value) or the valley (match between the 16-bit counter value and 0001H)). The operation procedure is as follows.

<1> Rewriting the TABnCCR0, TABnCCR2, TABnCCR3, TABnOPT1, TAAAnCCR0, and TAAAnCCR1 registers.

Do not rewrite registers that do not have to be rewritten.

<2> Rewriting the TABnCCR1 register.

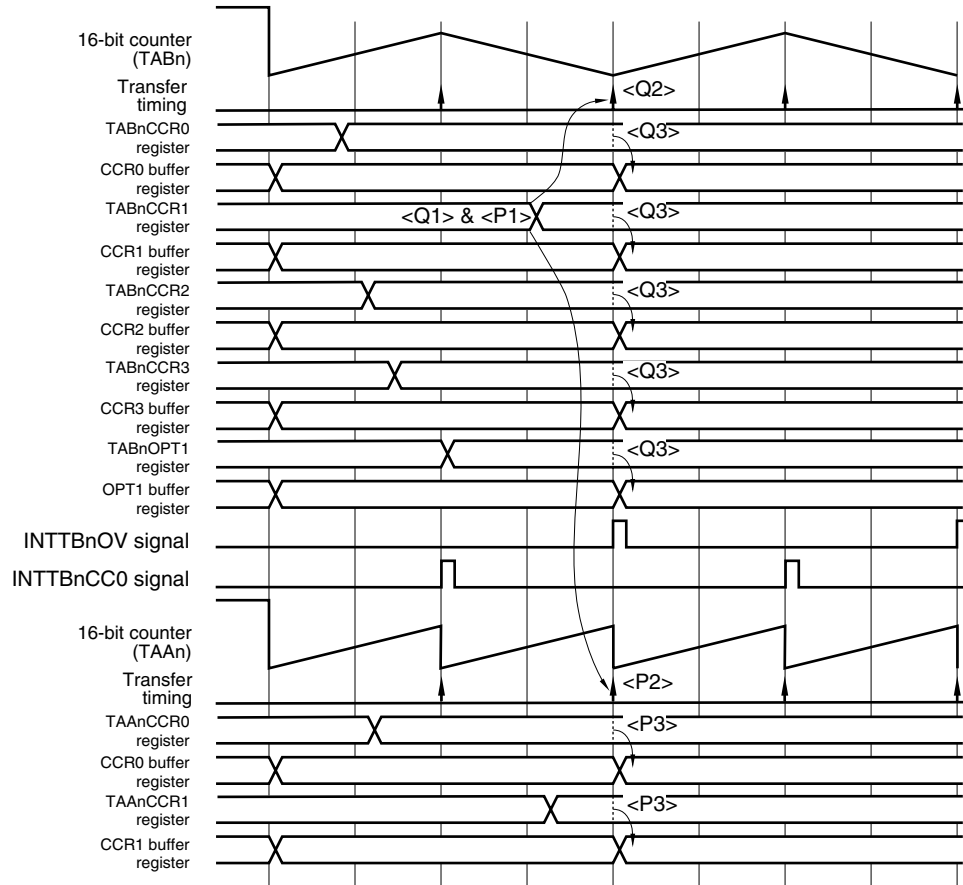
Rewrite the same value to the register even when it is not necessary to rewrite the TABnCCR1 register.

<3> Holding the next rewriting pending until the transfer timing is generated.

Rewrite the register next time after the INTTBnOV or INTTBnCC0 interrupt has occurred.

<4> Return to <1>.

Figure 10-26. Basic Operation in Batch Mode



[Operation of TABn]

<Q1> Write the TABnCCR1 register

<Q2> The target timing is the first transfer timing after a write to the TABnCCR1 register.

<Q3> The values are transferred all at once at the transfer timing.

[Operation of TAA n]

<P1> Write the TABnCCR1 register

<P2> The target timing is the first transfer timing after a write to the TABnCCR1 register.

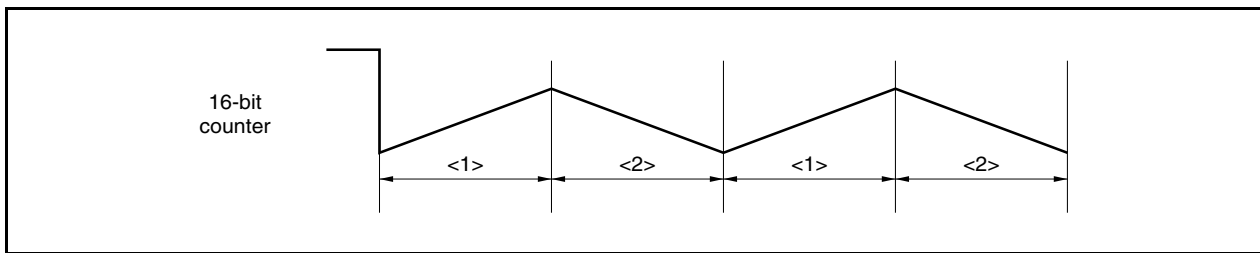
<P3> The values are transferred all at once at the transfer timing.

**(b) Rewriting TABnCCR0 register**

When rewriting the TABnCCR0 register in the batch rewrite mode, the output waveform differs depending on whether transfer occurs at the crest (match between the 16-bit counter value and TABnCCR0 register value) or at the valley (match between the 16-bit counter value and 0001H). Usually, it is recommended to rewrite the TABnCCR0 register while the 16-bit counter is counting down, and transfer the register value at the transfer timing of the crest timing.

Figure 10-28 shows an example of rewriting the TABnCCR0 register while the 16-bit counter is counting up (during period <1> in Figure 10-27). Figure 10-29 shows an example of rewriting the TABnCCR0 register while the counter is counting down (during period <2> in Figure 10-27).

**Figure 10-27. Basic Operation of 16-Bit Counter**



The transfer timing in Figure 10-28 is at the point where the crest timing occurs. While the 16-bit counter is counting down, the cycle changes and an asymmetrical triangular wave is output. Because the cycle changes, rewrite the duty factor (voltage data value).

Figure 10-28. Example of Rewriting TABnCCR0 Register (During Up Counting)

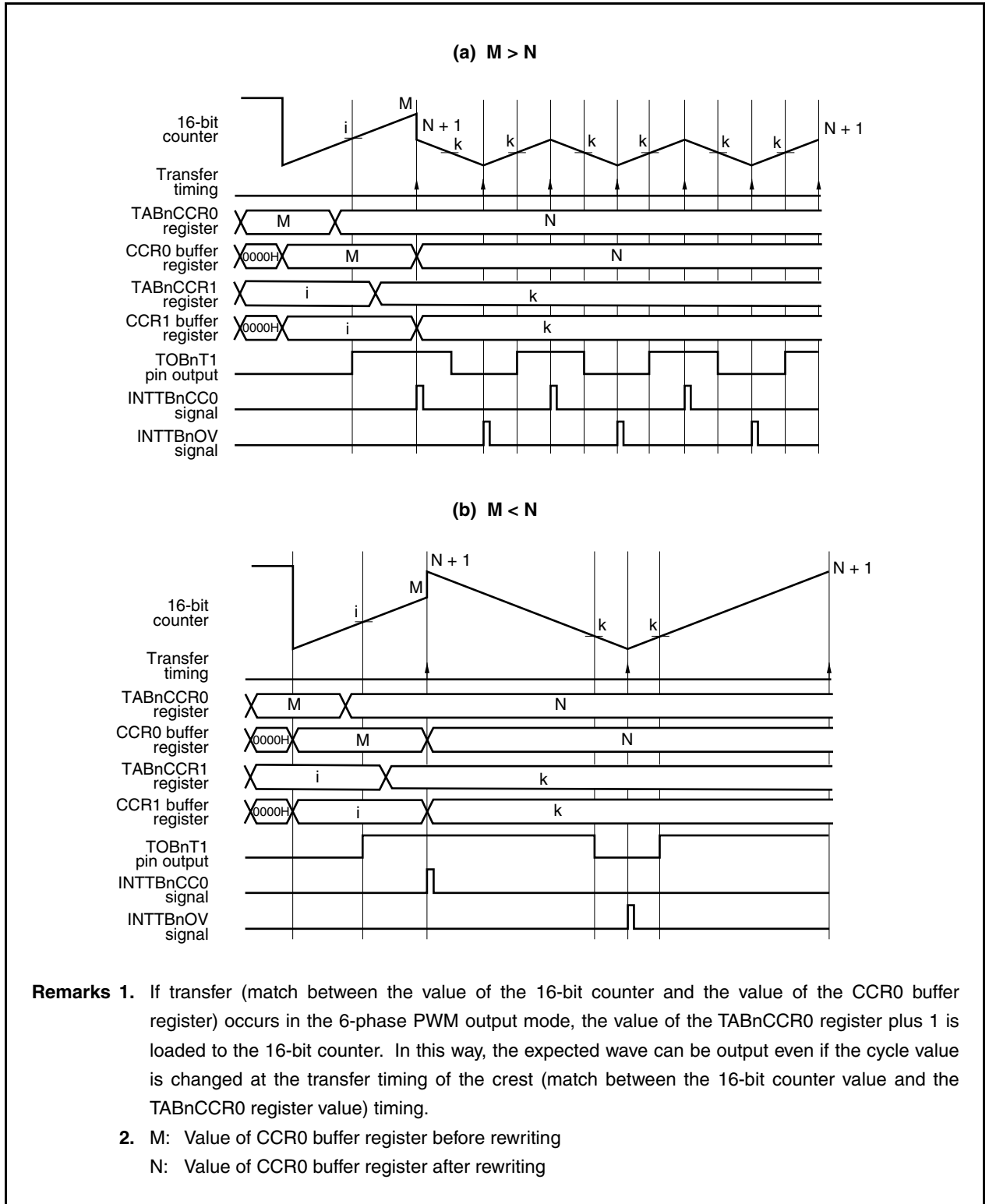
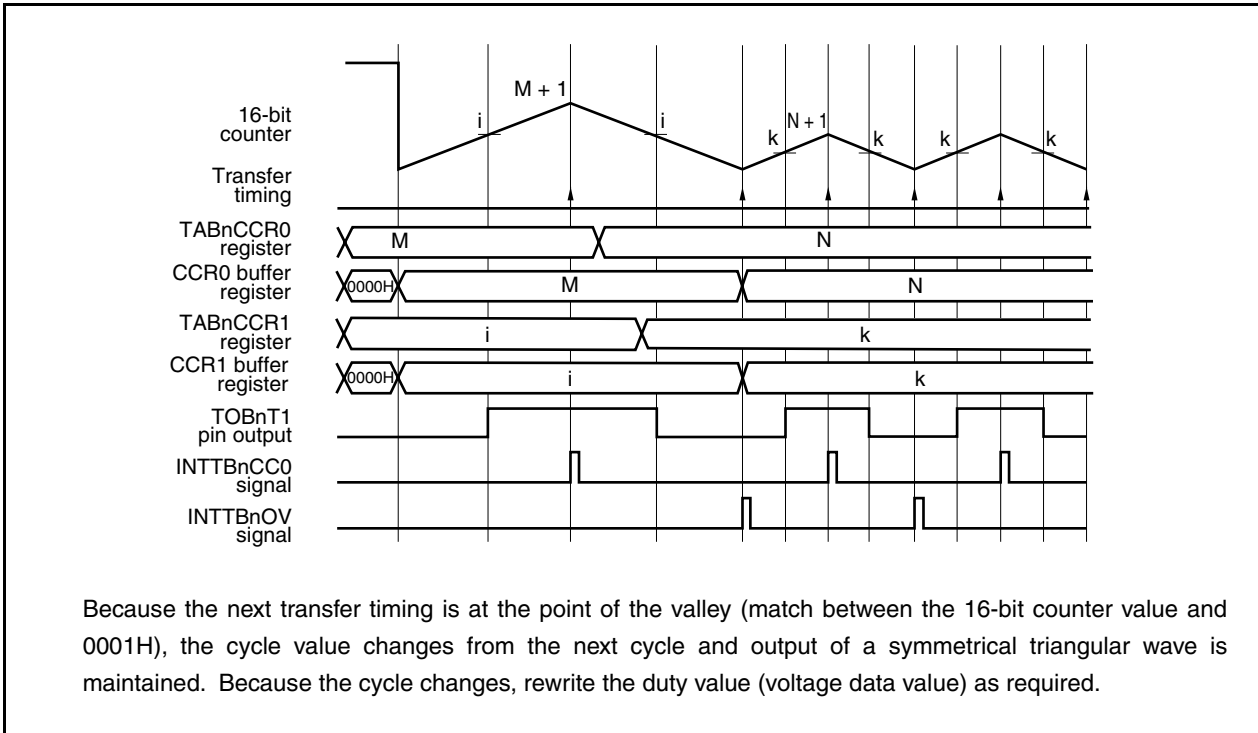


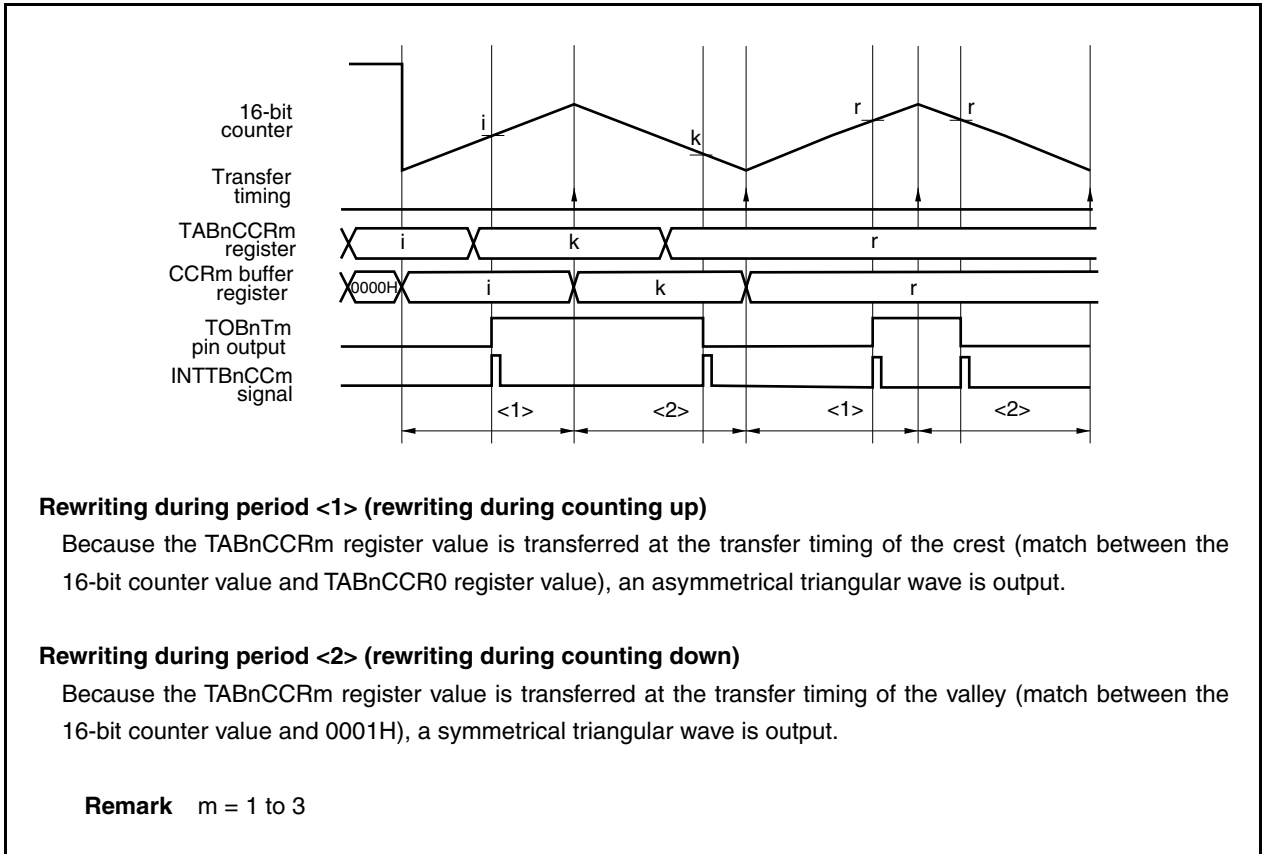
Figure 10-29. Example of Rewriting TABnCCR0 Register (During Down Counting)





(c) Rewriting TABnCCRm register

Figure 10-30. Example of Rewriting TABnCCRm Register



(d) Transferring TABnOPT1 register value

Do not set the TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits to other than 00000. When using the interrupt culling function, rewrite the TABnOPT1 register in the intermittent batch rewrite mode (transfer culling mode).

For details of rewriting the TABnOPT1 register, see 10.4.3 Interrupt culling function.

**(3) Intermittent batch rewriting mode (transfer culling mode)**

This mode is specified by setting the TABnOPT0.TABnCMS bit is 0 and the TABnOPT2.TABnRDE bit is 1.

In this mode, the values written to each compare register are transferred to the internal buffer register all at once at the culled transfer timing and compared with the count value.

The transfer timing is the timing at which an interrupt is generated (INTTBnCC0, INTTBnOV) by interrupt culling.

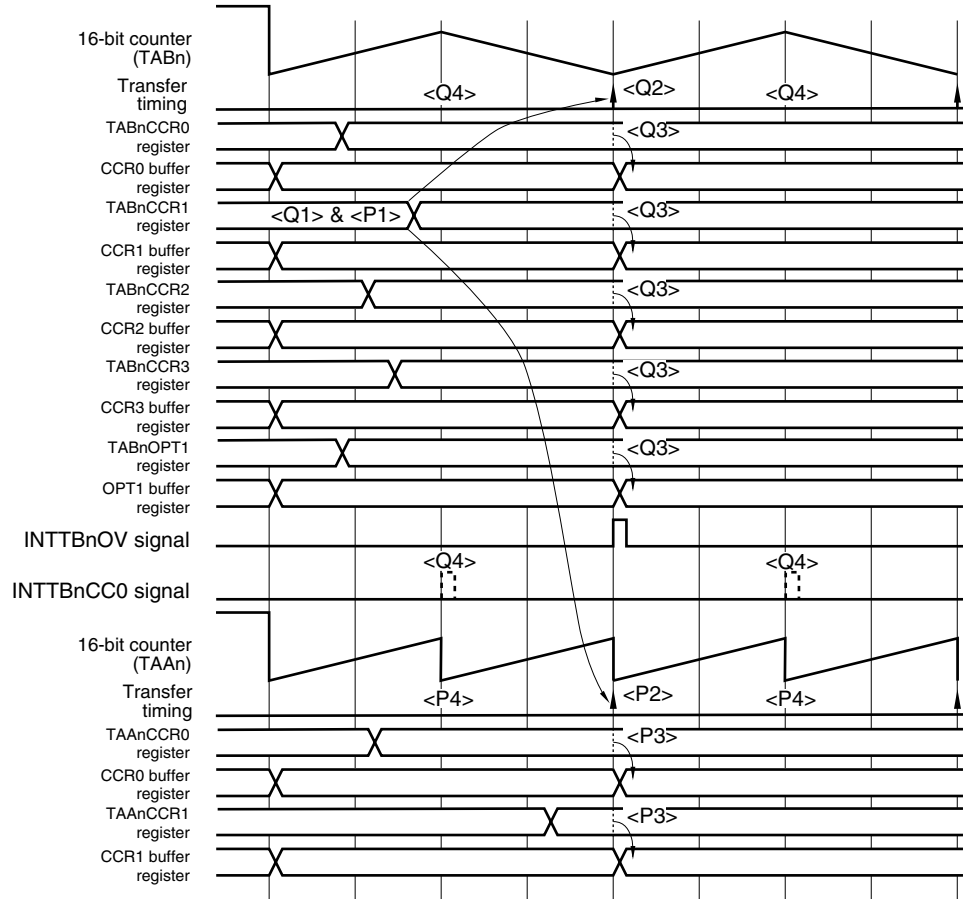
For details of the interrupt culling function, see **10.4.3 Interrupt culling function**.

**(a) Rewriting procedure**

If data is written to the TABnCCR1 register, the TABnCCR0 to TABnCCR3, TABnOPT1, TAAAnCCR0, and TAAAnCCR1 registers are transferred all at once to the internal buffer register at the next transfer timing. Therefore, write to the TABnCCR1 register last. Writing to the register is prohibited after the TABnCCR1 register has been written until the transfer timing is generated (until the INTTBnOV or INTTBnCC0 interrupt occurs). The operation procedure is as follows.

- <1> Rewrite the TABnCCR0, TABnCCR2, TABnCCR3, TABnOPT1, TAAAnCCR0, and TAAAnCCR1 registers.  
Do not rewrite registers that do not have to be rewritten.
- <2> Rewrite the TABnCCR1 register.  
Rewrite the same value to the register even when it is not necessary to rewrite the TABnCCR1 register.
- <3> Hold the next rewriting pending until the transfer timing is generated.  
Perform the next rewrite after the INTTBnOV or INTTBnCC0 interrupt has occurred.
- <4> Return to <1>.

Figure 10-31. Basic Operation in Intermittent Batch Rewriting Mode



**[TABn operation]**

- <Q1> Write the TABnCCR1 register.
- <Q2> Rewrite the register at the transfer timing that is generated after the TABnCCR1 register has been rewritten.
- <Q3> The registers are transferred all at once at the transfer timing.
- <Q4> The transfer timing is also called as the interrupts are called.

**[TAAn operation]**

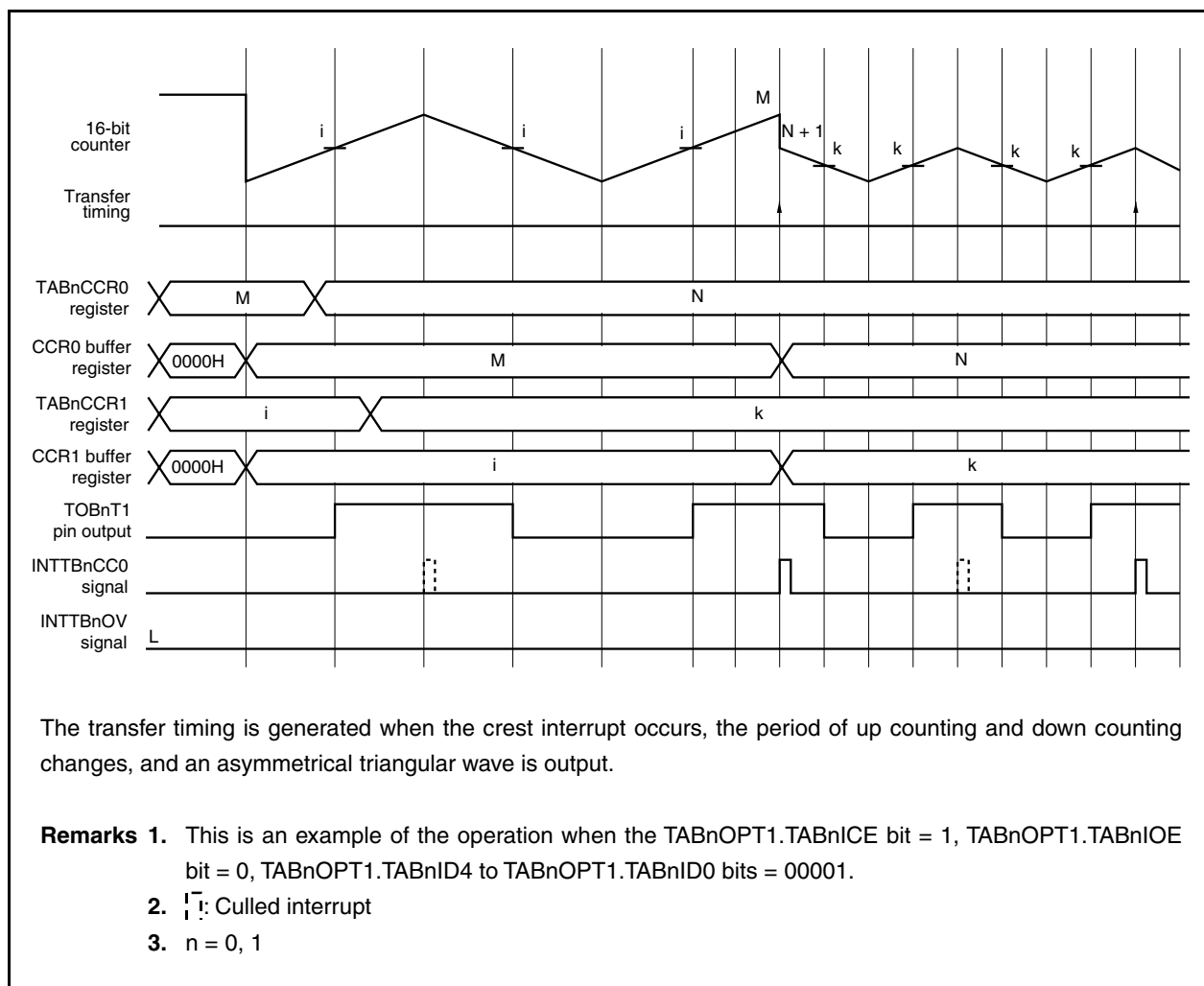
- <P1> Write the TABnCCR1 register.
- <P2> Rewrite the register at the transfer timing that is generated after the TABnCCR1 register has been rewritten.
- <P3> The registers are transferred all at once at the transfer timing.
- <P4> The transfer timing is also called as the interrupts are called.

**Remark** This is an example of the operation when the TABnOPT1.TABnICE bit = 1, TABnOPT1.TABnIOE bit = 1, TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00001.

**(b) Rewriting TABnCCR0 register**

When rewriting the TABnCCR0 register in the intermittent batch mode, the output waveform differs depending on where the occurrence of the crest or valley interrupt is specified by the interrupt culling setting. The following figure illustrates the change of the output waveform when interrupts are culled.

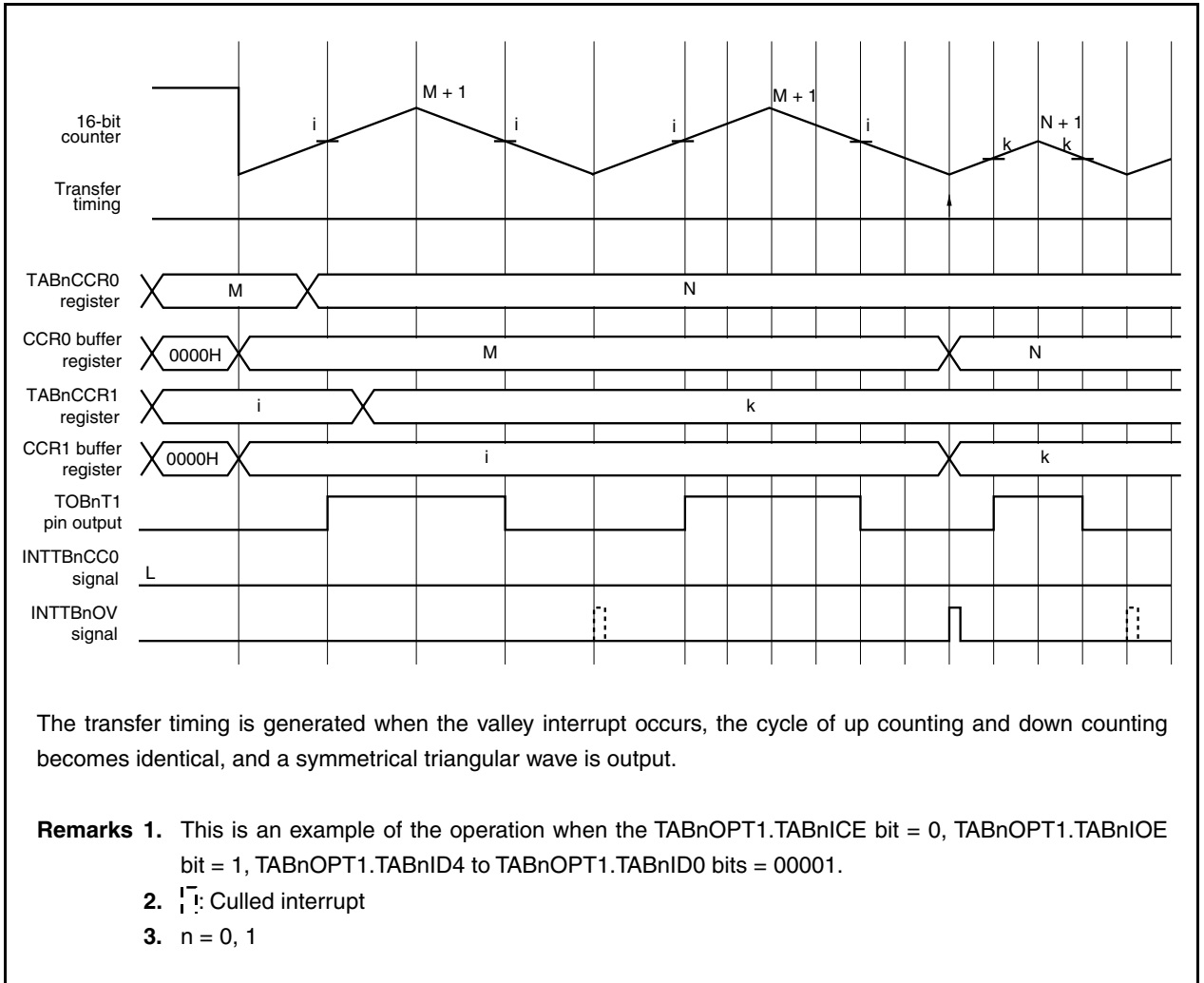
**Figure 10-32. Rewriting TABnCCR0 Register (When Crest Interrupt Is Set)**



The transfer timing is generated when the crest interrupt occurs, the period of up counting and down counting changes, and an asymmetrical triangular wave is output.

- Remarks**
1. This is an example of the operation when the TABnOPT1.TABnICE bit = 1, TABnOPT1.TABnIOE bit = 0, TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00001.
  2.  $\overline{i}$ : Culled interrupt
  3. n = 0, 1

Figure 10-33. Rewriting TABnCCR0 Register (When Valley Interrupt Is Set)

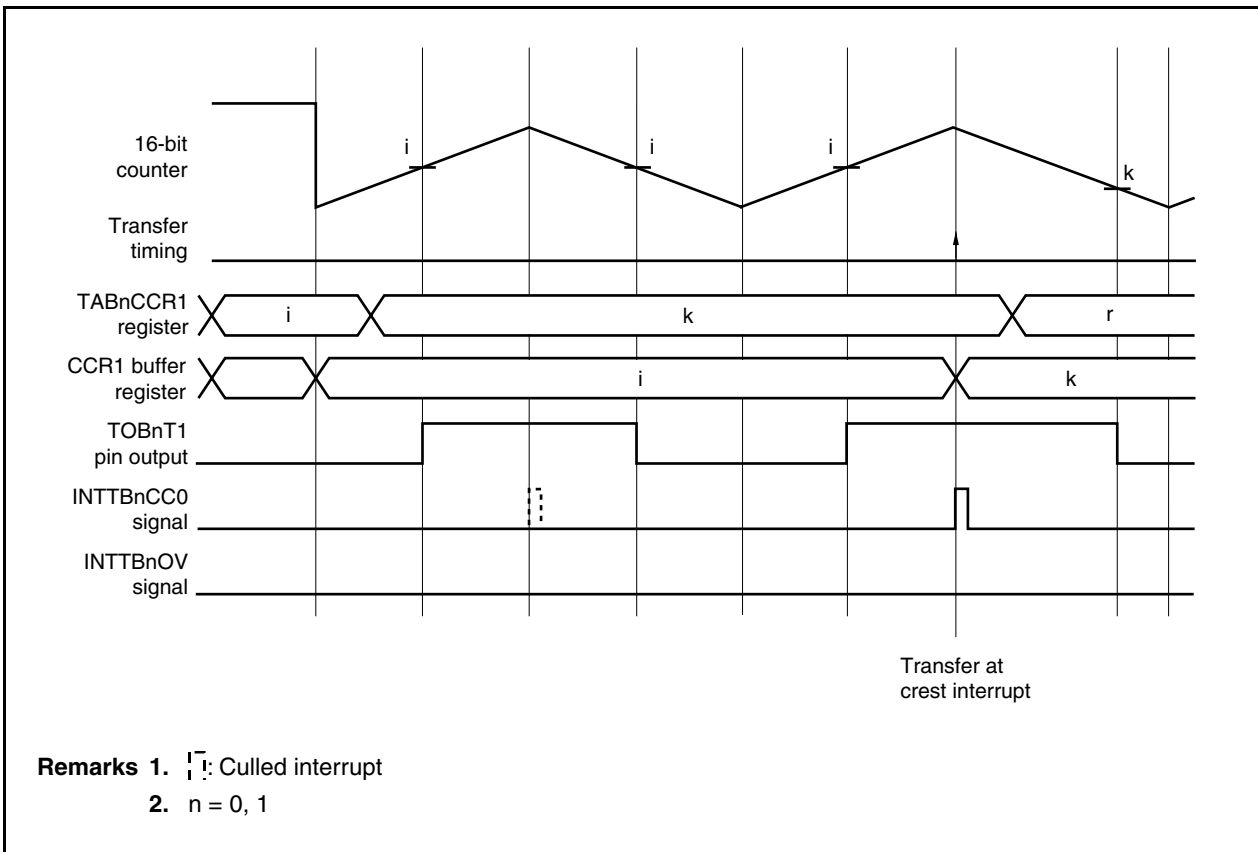


(c) Rewriting TABnCCR1 to TABnCCR3 registers

- Transfer at crest when crest interrupt is set  
Because the register is transferred at the transfer timing of the crest interrupt, an asymmetrical triangular wave is output.

Figure 10-34. Rewriting TABnCCR1 Register

(TABnOPT1.TABnICE Bit = 1, TABnOPT1.TABnIOE Bit = 0, TABnOPT1.TABnID4 to TABnOPT1.TABnID0 = 00001)

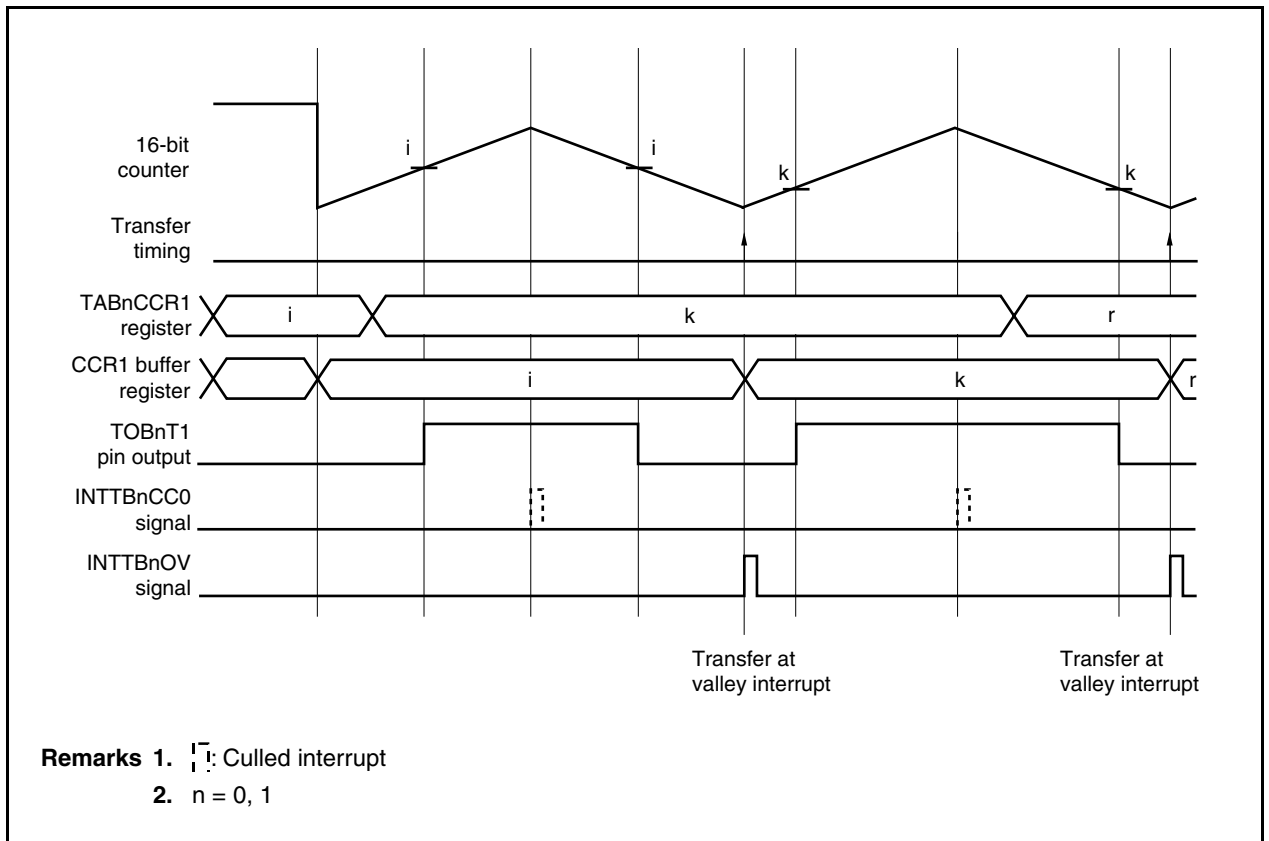


- Transfer at valley when valley interrupt is set

Because the register is transferred at the transfer timing of the valley interrupt, a symmetrical triangular wave is output.

**Figure 10-35. Rewriting TABnCCR1 Register**

(TABnOPT1.TABnICE Bit = 1, TABnOPT1.TABnIOE Bit = 1, TABnOPT1.TABnID4 to TABnOPT1.TABnID0 = 00001)



**(d) Rewriting TABnOPT1 register**

Because a new interrupt culling value is transferred when the value of the interrupt culling counter matches the value of the 16-bit counter, the next interrupt and those that follow occur at the set interval.

For details of rewriting the TABnOPT1 register, see **10.4.3 Interrupt culling function**.

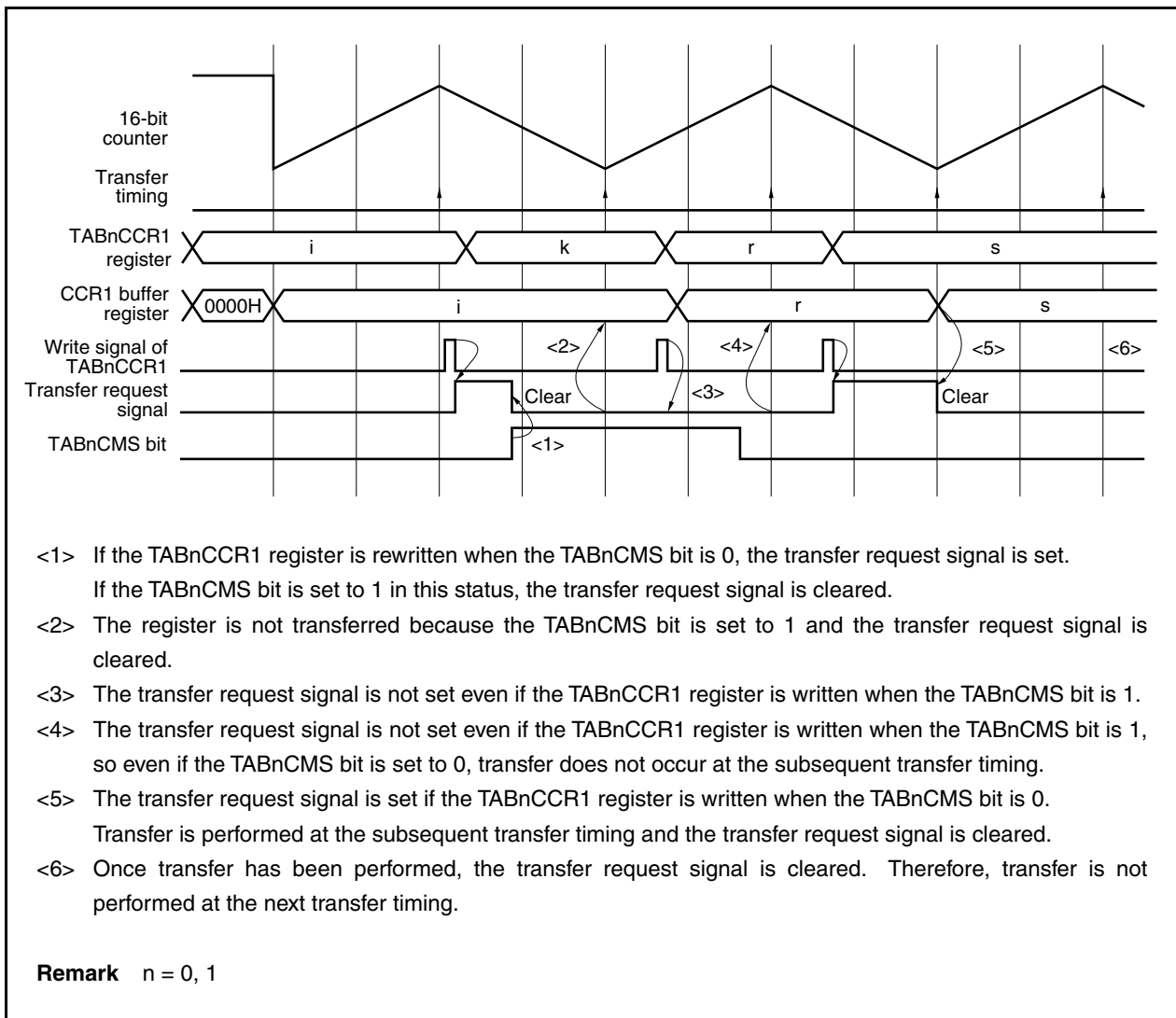
**(4) Rewriting TABnOPT0.TABnCMS bit**

The TABnCMS bit can select the anytime rewrite mode and batch rewrite mode. This bit can be rewritten during timer operation (when TABnCTL0.TABnCE bit = 1). However, the operation and caution illustrated in Figure 10-36 are necessary.

If the TABnCCR1 register is written when the TABnCMS bit is set to 0, a transfer request signal (internal signal) is set.

When the transfer request signal is set, the register is transferred at the next transfer timing, and the transfer request signal is cleared. This transfer request signal is also cleared when the TABnCMS bit is set to 1.

**Figure 10-36. Rewriting TABnCMS Bit**





#### 10.4.5 TAA<sub>n</sub> tuning operation for A/D conversion start trigger signal output

This section explains the tuning operation of TAA<sub>n</sub> and TAB<sub>n</sub> in the 6-phase PWM output mode.

In the 6-phase PWM output mode, the tuning operation is performed with TAB<sub>n</sub> serving as the master and TAA<sub>n</sub> as a slave. The conversion start trigger signal of A/D converters 0 and 1 can be set as the A/D conversion start trigger source by the INTTAnCC0 and INTTAnCC1 signals of TAA<sub>n</sub> and the INTTBnOV and INTTBnCC0 signals of TAB<sub>n</sub>.

**Remark** n = 0, 1

##### (1) Tuning operation starting procedure

The TAA<sub>n</sub> and TAB<sub>n</sub> registers should be set using the following procedure to perform the tuning operation.

##### (a) Setting of TAA<sub>n</sub> register (stop the operations of TAB<sub>n</sub> and TAA<sub>n</sub> (by setting the TABnCTL0.TABnCE bit and TAA<sub>n</sub>CTL0.TAA<sub>n</sub>CE bit to 0))

- Set the TAA<sub>n</sub>CTL1 register to 85H (set the tuning operation slave mode and free-running timer mode).
- Set the TAA<sub>n</sub>OPT0 register to 00H (select the compare register).
- Set an appropriate value to the TAA<sub>n</sub>CCR0 and TAA<sub>n</sub>CCR1 registers (set the default value for comparison for starting the operation).

##### (b) Setting of TAB<sub>n</sub> register

- Set the TAB<sub>n</sub>CTL1 register to 07H (set the master mode and 6-phase PWM output mode).
- Set an appropriate value to the TAB<sub>n</sub>IOC0 register (set the output mode of TOB<sub>n</sub>T1 to TOB<sub>n</sub>T3).  
However, set the TAB<sub>n</sub>OL0 bit to 0 and the TAB<sub>n</sub>OEO bit to 1 (enable positive phase output). Unless this setting is made, the crest interrupt (INTTBnCC0) and valley interrupt (INTTBnOV) do not occur. Consequently, the conversion start trigger signal of A/D converters 0 and 1 is not correctly generated.
- Clear the TAB<sub>n</sub>IOC1 and TAB<sub>n</sub>IOC2 registers to 00H (the TIB<sub>n</sub>0 to TIB<sub>n</sub>3, EVT<sub>Bn</sub>, and TRGB<sub>n</sub> pins of TAB<sub>n</sub> are not used).
- Clear the TAB<sub>n</sub>OPT0 register to 00H (select the compare register).
- Set an appropriate value to the TAB<sub>n</sub>CCR0 to TAB<sub>n</sub>CCR3 registers (set the default value for comparison for starting the operation).
- Set the TAB<sub>n</sub>CTL0 register to 0xH (set the TAB<sub>n</sub>CE bit to 0 and the operating clock of TAB<sub>n</sub>).  
The operating clock of TAB<sub>n</sub> set by the TAB<sub>n</sub>CTL0 register is also supplied to TAA<sub>n</sub>, and the count operation is performed at the same timing. The operating clock of TAA<sub>n</sub> set by the TAA<sub>n</sub>CTL0 register is ignored.

##### (c) Setting of TMQOP<sub>n</sub> (TMQ<sub>n</sub> option) register

- Set an appropriate value to the TAB<sub>n</sub>OPT1 and TAB<sub>n</sub>OPT2 registers.
- Set an appropriate value to the TAB<sub>n</sub>IOC3 register (set TOB<sub>n</sub>B1 to TOB<sub>n</sub>B3 in the output mode).
- Set an appropriate value to the TAB<sub>n</sub>DTC register (set the default value for comparison for starting the operation).

##### (d) Setting of alternate function

- Select the alternate function of the port by setting the port to the port control mode.

**(e) Set the TAA<sub>n</sub>CE bit to 1 and set the TAB<sub>n</sub>CE bit to 1 immediately after that to start the 6-phase PWM output operation.**

Rewriting the TAB<sub>n</sub>CTL0, TAB<sub>n</sub>CTL1, TAB<sub>n</sub>IOC1, TAB<sub>n</sub>IOC2, TAA<sub>n</sub>CTL0, and TAA<sub>n</sub>CTL1 registers is prohibited during operation. The operation and the PWM output waveform are not guaranteed if any of these registers is rewritten during operation. However, rewriting the TAB<sub>n</sub>CTL0.TAB<sub>n</sub>CE bit to clear it is permitted. Manipulating (reading/writing) the other TAB<sub>n</sub>, TAA<sub>n</sub>, and TMQ<sub>n</sub> option registers is prohibited until the TAA<sub>n</sub>CTL0.TAA<sub>n</sub>CE bit is set to 1 and then the TAB<sub>n</sub>CE bit is set to 1.

**(2) Tuning operation clearing procedure**

To clear the tuning operation and exit the 6-phase PWM output mode, set the TAA<sub>n</sub> and TAB<sub>n</sub> registers using the following procedure.

- <1> Clear the TAB<sub>n</sub>CTL0.TAB<sub>n</sub>CE bit to 0 and stop the timer operation.
- <2> Clear the TAA<sub>n</sub>CTL0.TAA<sub>n</sub>CE bit to 0 so that TAA<sub>n</sub> can be separated.
- <3> Stop the timer output by using the TAB<sub>n</sub>IOC0 register.
- <4> Clear the TAA<sub>n</sub>CTL1.TAA<sub>n</sub>SYE bit to 0 to clear the tuning operation.

**Caution** Manipulating (reading/writing) the other TAB<sub>n</sub>, TAA<sub>n</sub>, and TMQ<sub>n</sub> option registers is prohibited until the TAB<sub>n</sub>CE bit is set to 0 and then the TAA<sub>n</sub>CE bit is set to 0.

**(3) When not tuning TAA<sub>n</sub>**

When the match interrupt signal of TAA<sub>n</sub> is not necessary as the conversion trigger source that starts A/D converters 0 and 1, TAA<sub>n</sub> can be used independently as a separate timer without being tuned. In this case, the match interrupt signal of TAA<sub>n</sub> cannot be used as a trigger source to start A/D conversion in the 6-phase PWM output mode. Therefore, fix the TAB<sub>n</sub>OPT2.TAB<sub>n</sub>AT2, TAB<sub>n</sub>OPT2.TAB<sub>n</sub>AT3, TAB<sub>n</sub>OPT3.TAB<sub>n</sub>AT6, and TAB<sub>n</sub>OPT3.TAB<sub>n</sub>AT7 bits to 0.

The other control bits can be used in the same manner as when TAA<sub>n</sub> is tuned.

If TAA<sub>n</sub> is not tuned, the compare registers (TAA<sub>n</sub>CCR0 and TAA<sub>n</sub>CCR1) of TAA<sub>n</sub> are not affected by the settings of the TAB<sub>n</sub>OPT0.TAB<sub>n</sub>CMS and TAB<sub>n</sub>OPT2.TAB<sub>n</sub>RDE bits. For the initialization procedure when TAA<sub>n</sub> is not tuned, see (b) to (e) in **10.4.5 (1) Tuning operation starting procedure**. (a) is not necessary because it is a step used to set TAA<sub>n</sub> for the tuning operation.

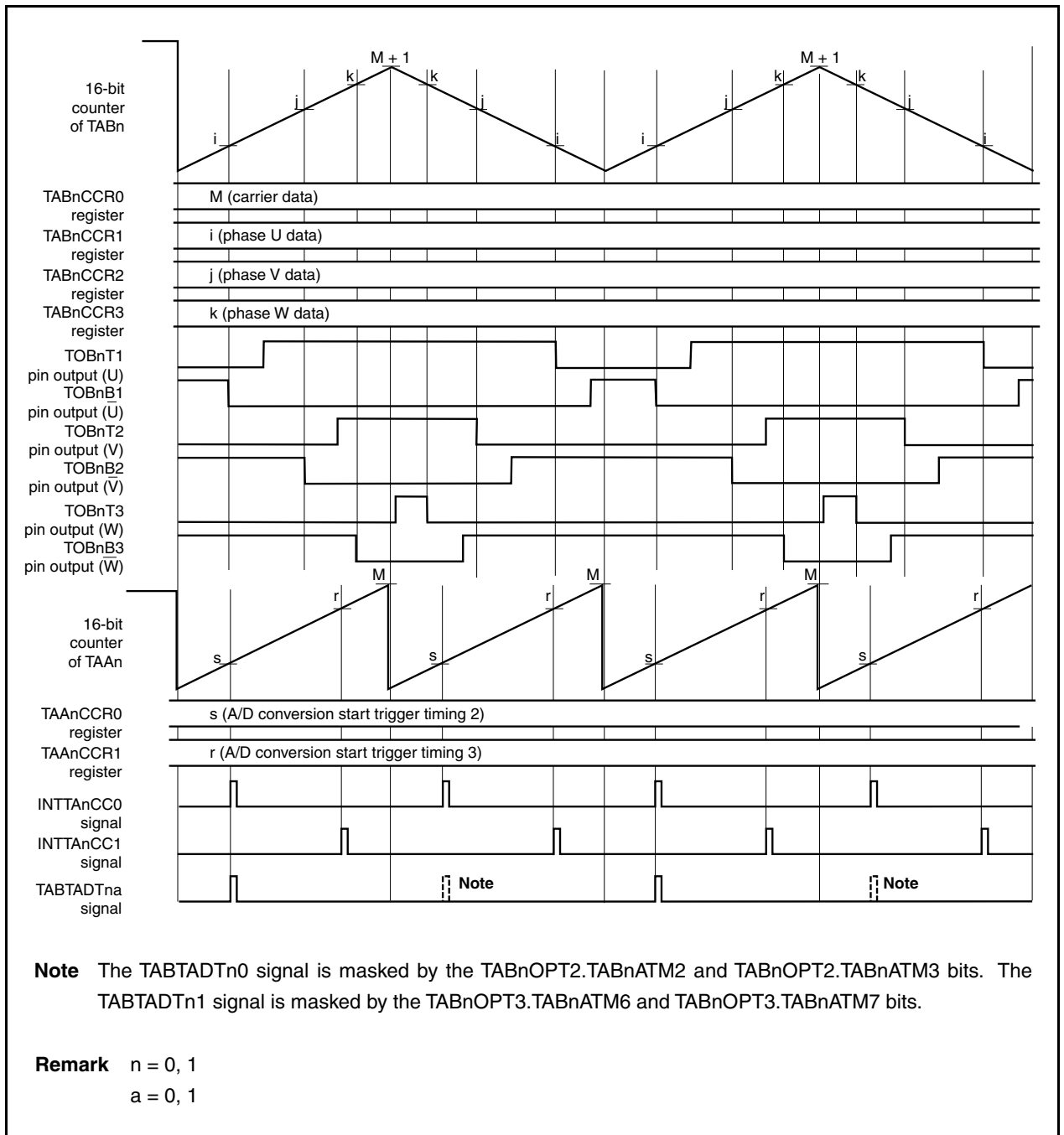
**(4) Basic operation of TAA<sub>n</sub> during tuning operation**

The 16-bit counter of TAA<sub>n</sub> only counts up. The 16-bit counter is cleared by the set cycle value of the TAB<sub>n</sub>CCR0 register and starts counting from 0000H again. The count value of this counter is the same as the value of the 16-bit counter of TAA<sub>n</sub> when it counts up. However, it is not the same when the 16-bit counter of TAB<sub>n</sub> counts down.

- When TAB<sub>n</sub> counts up (same value)
  - 16-bit counter of TAB<sub>n</sub>: 0000H → M (up counting)
  - 16-bit counter of TAA<sub>n</sub>: 0000H → M (up counting)
- When TAB<sub>n</sub> counts down (not same value)
  - 16-bit counter of TAB<sub>n</sub>: M + 1 → 0001H (down counting)
  - 16-bit counter of TAA<sub>n</sub>: 0000H → M (up counting)

<R>

Figure 10-37. TAA<sub>n</sub> During Tuning Operation



#### 10.4.6 A/D conversion start trigger output function

The V850E/IF3 and V850E/IG3 have a function to select four trigger sources (INTTBnOV, INTTBnCC0, INTTAnCC0, INTTAnCC1) to generate the A/D conversion start trigger signal (TABTADTn0, TABTADTn1) of A/D converters 0 and 1.

The trigger sources are specified by the TABnOPT2.TABnAT0 to TABnOPT2.TABnAT3 and TABnOPT3.TABnAT4 to TABnOPT3.TABnAT7 bits.

- TABnAT0, TABnAT4 bits = 1:  
A/D conversion start trigger signal generated when INTTBnOV (counter underflow) occurs.
- TABnAT1, TABnAT5 bits = 1:  
A/D conversion start trigger signal generated when INTTBnCC0 (cycle match) occurs.
- TABnAT2, TABnAT6 bits = 1:  
A/D conversion start trigger signal generated when INTTAnCC0 (match of TAAAnCCR0 register of TAAAn during tuning operation) occurs.
- TABnAT3, TABnAT7 bits = 1:  
A/D conversion start trigger signal generated when INTTAnCC1 (match of TAAAnCCR1 register of TAAAn during tuning operation) occurs.

The A/D conversion start trigger signals selected by the TABnAT0 to TABnAT3 and TABnAT4 to TABnAT7 bits are ORed and output. Therefore, two or more trigger sources can be specified at the same time.

The INTTBnOV and INTTBnCC0 signals selected by the TABnAT0, TABnAT1, TABnAT4, and TABnAT5 bits are called interrupt signals.

Therefore, these signals are output after the interrupts have been culled and, unless interrupt output is enabled (TABnOPT1.TABnICE, TABnOPT1.TABnIOE bits), the A/D conversion start trigger is not output.

The trigger sources (INTTAnCC0 and INTTAnCC1) from TAAAn have a function to mask the A/D conversion start trigger signal depending on the status of the up-count/down-count of the 16-bit counter, if so set by the TABnAT2, TABnAT3, TABnAT6, and TABnAT7 bits.

- TABnATM2, TABnATM6 bits:  
Correspond to the TABnAT2 and TABnAT6 bits and control INTTAnCC0 (match interrupt signal) of TAAAn.
- TABnATM2, TABnATM6 bits = 0  
The A/D conversion start trigger signal is output when the 16-bit counter counts up (TABnOPT0.TABnCUF bit = 0), and the A/D conversion start trigger signal is not output when the 16-bit counter counts down (TABnOPT0.TABnCUF bit = 1).
- TABnATM2, TABnATM6 bits = 1  
The A/D conversion start trigger signal is output when the 16-bit counter counts down (TABnOPT0.TABnCUF bit = 1), and the A/D conversion start trigger signal is not output when the 16-bit counter counts up (TABnOPT0.TABnCUF bit = 0).
- TABnATM3, TABnATM7 bits:  
Correspond to the TABnAT3 and TABnAT7 bits and control INTTAnCC1 (match interrupt signal) of TAAAn.
- TABnATM3, TABnATM7 bits = 0  
The A/D conversion start trigger signal is output when the 16-bit counter counts up (TABnCUF bit = 0), and the A/D conversion start trigger signal is not output when the 16-bit counter counts down (TABnCUF bit = 1).
- TABnATM3, TABnATM7 bits = 1  
The A/D conversion start trigger signal is output when the 16-bit counter counts down (TABnCUF bit = 1), and the A/D conversion start trigger signal is not output when the 16-bit counter counts up (TABnCUF bit = 0).

The TABnATM3, TABnATM2, TABnAT3 to TABnAT0, TABnATM7, TABnATM6, and TABnAT7 to TABnAT4 bits can be rewritten while the timer is operating. If the bit that sets the A/D conversion start trigger signal is rewritten while the timer is operating, the new setting is immediately reflected on the output status of the A/D conversion start trigger. These control bits do not have a transfer function and can be used only in the anytime rewriting mode.

- Cautions**
- 1. The A/D conversion start trigger signal output that is set by the TABnAT2, TABnAT3, TABnAT6, and TABnAT7 bits can be used only when TAA<sub>n</sub> is performing a tuning operation as the slave timer of TAB<sub>n</sub>. If TAB<sub>n</sub> and TAA<sub>n</sub> are not performing a tuning operation, or if a mode other than the 6-phase PWM output mode is used, the output cannot be guaranteed.**
  - 2. The TOB<sub>n</sub>0 signal output is internally used to identify whether the 16-bit counter is counting up or down. Therefore, enable TOB<sub>n</sub>0 pin output by setting the TABnIOC0.TABnOL0 bit to 0 and the TABnIOC0.TABnOE0 bit to 1.**

Figure 10-38. Example of A/D Conversion Start Trigger (TABTADTn0) Signal Output (TABnOPT1.TABnICE Bit = 1, TABnOPT1.TABnIOE Bit = 1, TABnOPT1.TABnID4 to TABnOPT1.TABnID0 Bits = 00000: Without Interrupt Culling)

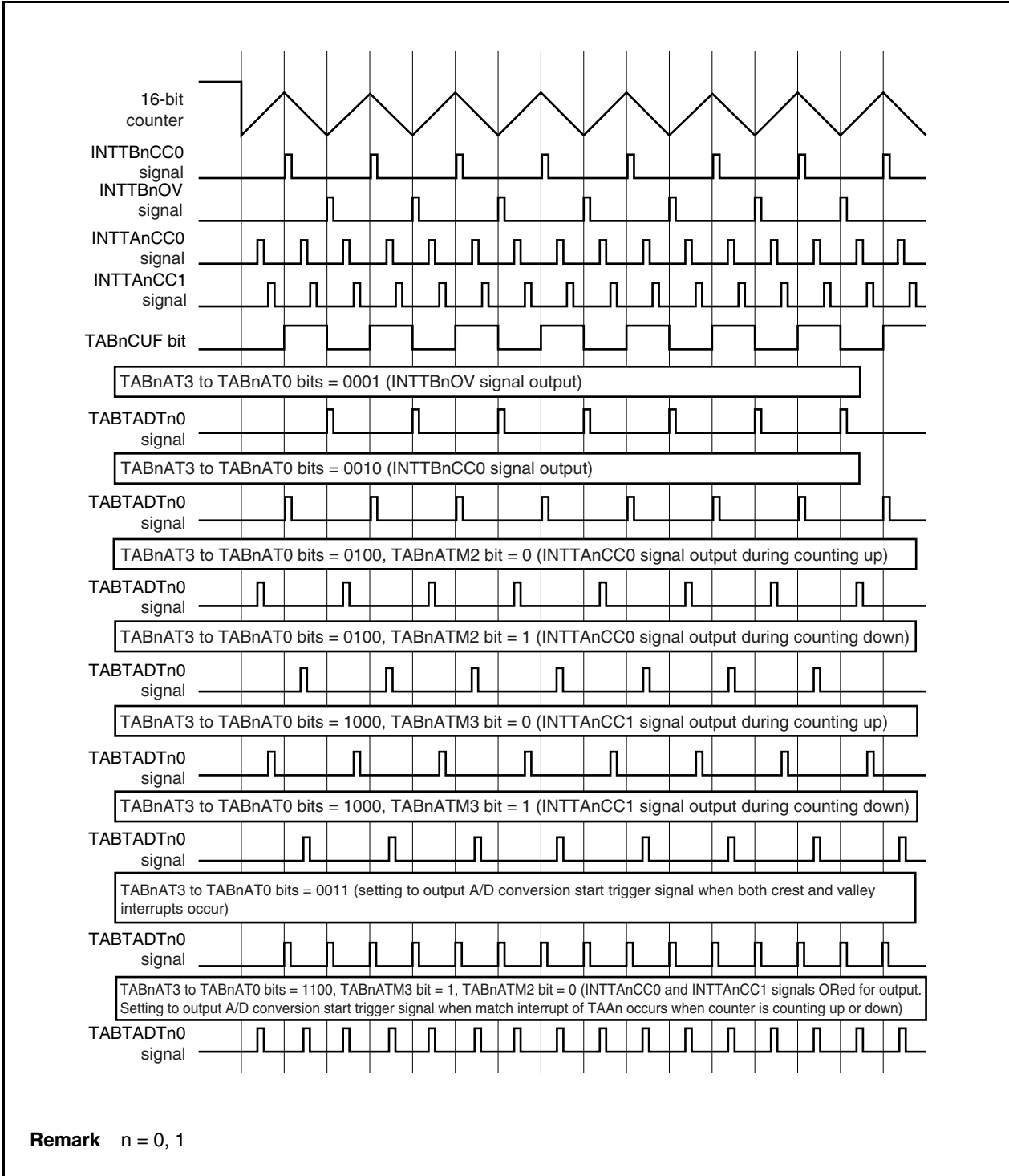


Figure 10-39. Example of A/D Conversion Start Trigger (TABTADTn0) Signal Output (TABnOPT1.TABnICE Bit = 0, TABnOPT1.TABnIOE Bit = 1, TABnOPT1.TABnID4 to TABnOPT1.TABnID0 Bits = 00010: With Interrupt Culling) (1)

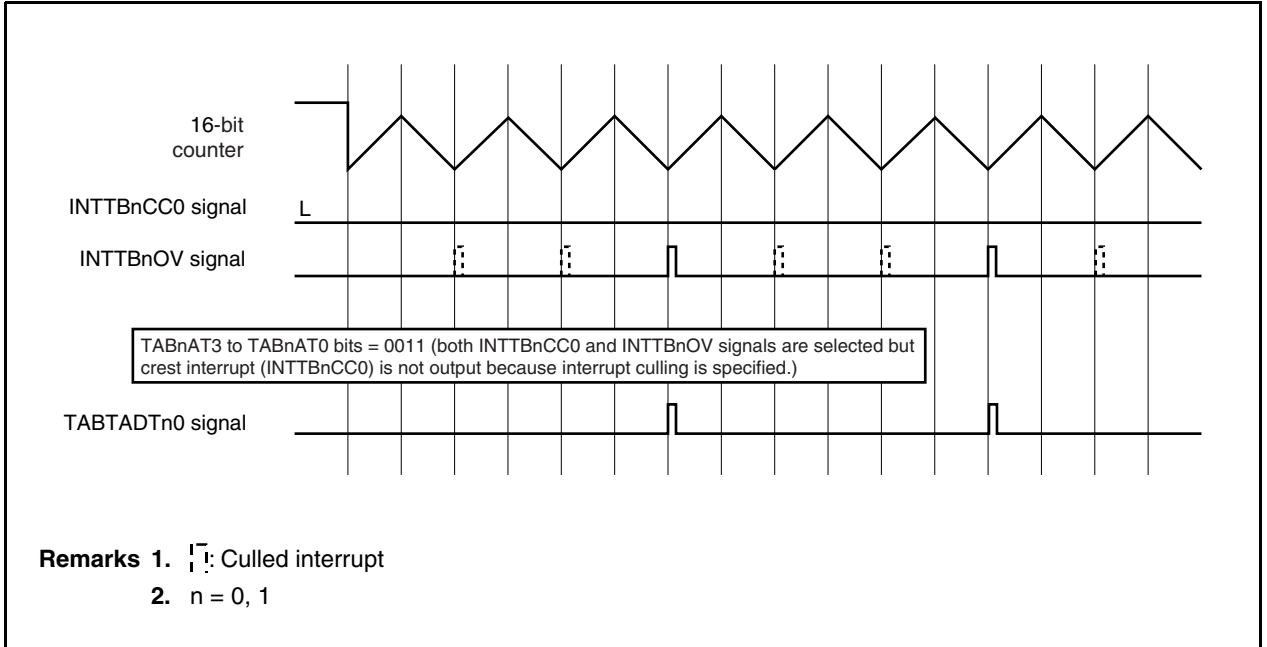
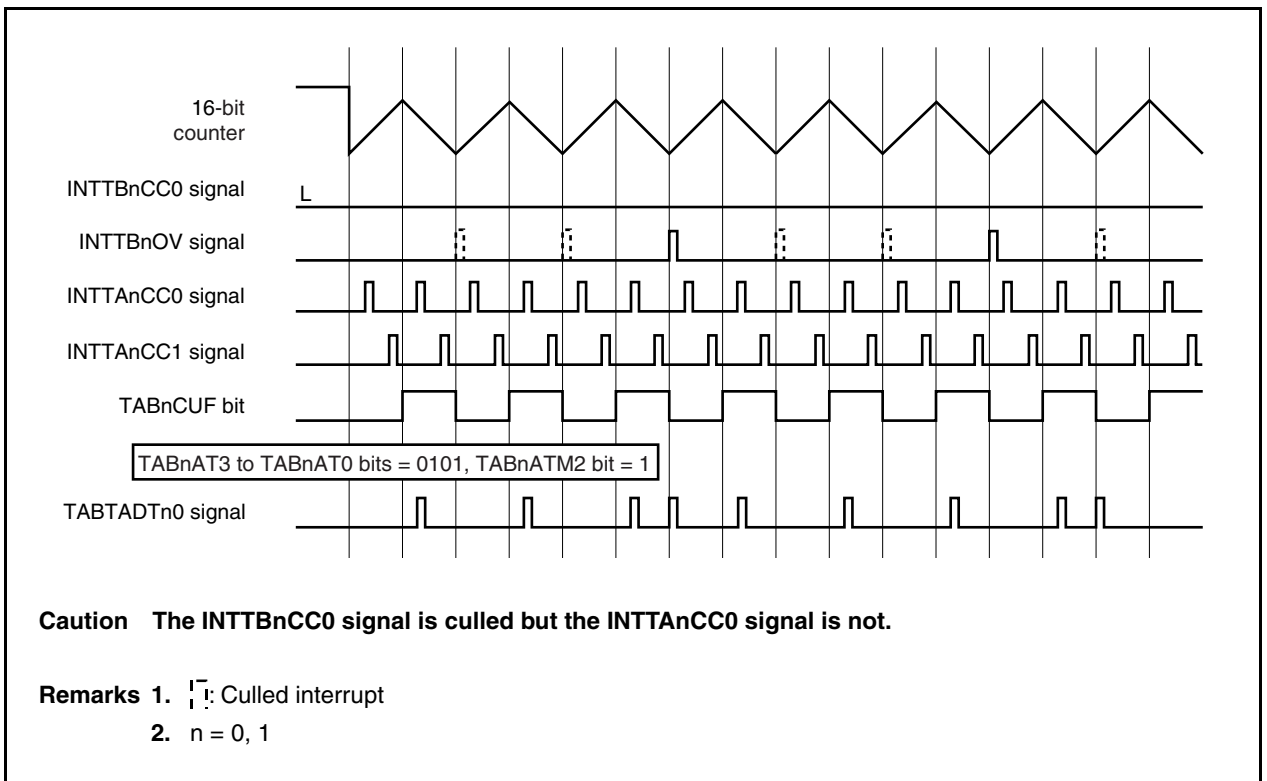


Figure 10-40. Example of A/D Conversion Start Trigger (TABTADTn0) Signal Output (TABnOPT1.TABnICE Bit = 0, TABnOPT1.TABnIOE Bit = 1, TABnOPT1.TABnID4 to TABnOPT1.TABnID0 Bits = 00010: With Interrupt Culling) (2)



(1) Operation under boundary condition (operation when 16-bit counter matches INTTAnCC0 signal)

**Table 10-3. Operation When TABnCCR0 Register = M, TABnAT2 Bit = 1, TABnAT6 Bit = 1, TABnATM2 Bit = 0, TABnATM6 Bit = 0 (Up Counting Period Selected)**

| Value of TAAAnCCR0 Register | Value of 16-bit Counter of TABn | Value of 16-bit Counter of TAAAn | Status of 16-bit Counter of TABn | Output of INTTAnCC0 Signal from TABTADTna Signal |
|-----------------------------|---------------------------------|----------------------------------|----------------------------------|--|
| 0000H                       | 0000H                           | 0000H                            | –                                | Output   |
| 0000H                       | M + 1                           | 0000H                            | –                                | Not output                                       |
| 0001H                       | 0001H                           | 0001H                            | Up count                         | Output   |
| 0001H                       | M                               | 0001H                            | Down count                       | Not output                                       |
| M                           | M                               | M                                | Up count                         | Output   |
| M                           | 0001H                           | M                                | Down count                       | Not output                                       |

**Table 10-4. Operation When TABnCCR0 Register = M, TABnAT2 Bit = 1, TABnAT6 Bit = 1, TABnATM2 Bit = 1, TABnATM6 Bit = 1 (Down Counting Period Selected)**

| Value of TAAAnCCR0 Register | Value of 16-bit Counter of TABn | Value of 16-bit Counter of TAAAn | Status of 16-bit Counter of TABn | Output of INTTAnCC0 Signal from TABTADTna Signal |
|-----------------------------|---------------------------------|----------------------------------|----------------------------------|--|
| 0000H                       | 0000H                           | 0000H                            | –                                | Not output                                       |
| 0000H                       | M + 1                           | 0000H                            | –                                | Output   |
| 0001H                       | 0001H                           | 0001H                            | Up count                         | Not output                                       |
| 0001H                       | M                               | 0001H                            | Down count                       | Output   |
| M                           | M                               | M                                | Up count                         | Not output                                       |
| M                           | 0001H                           | M                                | Down count                       | Output   |

**Caution** The TAAAnCCR0 register enables setting of “0” to “M” when the TABnCCR0 register = M. Setting of a value of “M + 1” or higher is prohibited.

If a value higher than “M + 1” is set, the 16-bit counter of TAAAn is cleared by “M”. Therefore, the TABTADTna signal is not output.

**Remark** n = 0, 1  
a = 0, 1



## CHAPTER 11 WATCHDOG TIMER FUNCTIONS

### 11.1 Functions

The watchdog timer has the following functions.

- Reset mode: Reset operation upon overflow of the watchdog timer (generation of WDTRES signal)
- Non-maskable interrupt request mode:  
Non-maskable interrupt operation upon overflow of the watchdog timer (generation of INTWDT signal)

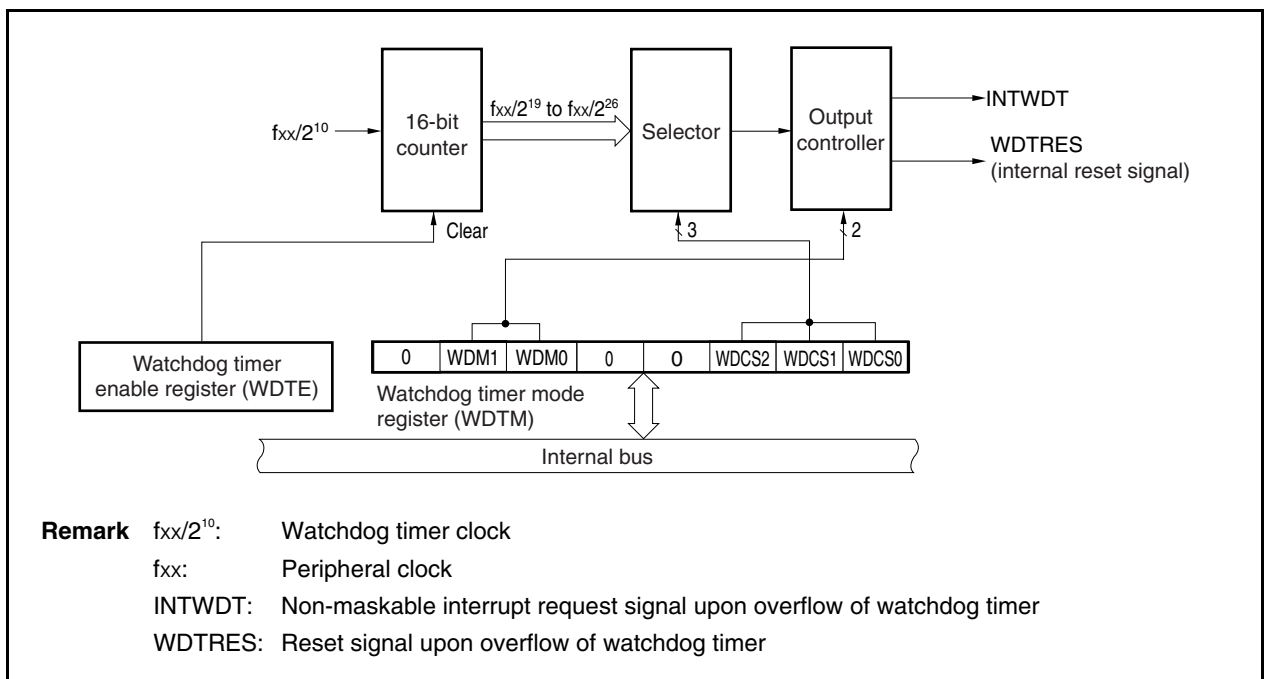
**Caution** The watchdog timer is stopped after reset is released.

It starts operating when “ACH” is written to the WDTE register. Also, write to the WDTM register for verification purposes only once, even if the default settings (reset mode, interval time:  $2^{26}/f_{xx}$ ) do not need to be changed.

### 11.2 Configuration

The block diagram of the watchdog timer is shown below.

Figure 11-1. Block Diagram of Watchdog Timer



The watchdog timer consists of the following hardware.

Table 11-1. Configuration of Watchdog Timer

| Item              | Configuration  |
|-------------------|--|
| Control registers | Watchdog timer mode register (WDTM)<br>Watchdog timer enable register (WDTE) |

### 11.3 Control Registers

#### (1) Watchdog timer mode register (WDTM)

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

This register can be read or written in 8-bit units. This register can be read any number of times, but can be written only once following reset release; it cannot then be written a second or subsequent time.

Reset sets this register to 67H.

|                  |     |                    |      |   |   |       |       |       |
|------------------|-----|--------------------|------|---|---|-------|-------|-------|
| After reset: 67H | R/W | Address: FFFFF6D0H |      |   |   |       |       |       |
| WDTM             | 7   | 6                  | 5    | 4 | 3 | 2     | 1     | 0     |
|                  | 0   | WDM1               | WDM0 | 0 | 0 | WDCS2 | WDCS1 | WDCS0 |

| WDM1 | WDM0 | Selection of operation mode of watchdog timer                     |
|------|------|---|
| 0    | 0    | Stop operation  |
| 0    | 1    | Non-maskable interrupt request mode (generation of INTWDT signal) |
| 1    | ×    | Reset mode (generation of WDTRES signal)                          |

**Cautions**

1. For details of the WDCS2 to WDCS0 bits, see Table 11-2 Overflow Time.
2. If the WDTM register is rewritten while the watchdog timer is counting, the counter of the watchdog timer is cleared to 0000H.
3. Be sure to clear bits 3, 4, and 7 to "0".

<R>

**Table 11-2. Overflow Time**

| WDCS2 | WDCS1 | WDCS0 | Overflow Time   | $f_{xx} = 64 \text{ MHz}$ | $f_{xx} = 32 \text{ MHz}$ |
|-------|-------|-------|-----------------|---------------------------|---------------------------|
| 0     | 0     | 0     | $2^{19}/f_{xx}$ | 8.2 ms                    | 16.4 ms                   |
| 0     | 0     | 1     | $2^{20}/f_{xx}$ | 16.4 ms                   | 32.8 ms                   |
| 0     | 1     | 0     | $2^{21}/f_{xx}$ | 32.8 ms                   | 65.5 ms                   |
| 0     | 1     | 1     | $2^{22}/f_{xx}$ | 65.5 ms                   | 131.1 ms                  |
| 1     | 0     | 0     | $2^{23}/f_{xx}$ | 131.1 ms                  | 262.1 ms                  |
| 1     | 0     | 1     | $2^{24}/f_{xx}$ | 262.1 ms                  | 524.3 ms                  |
| 1     | 1     | 0     | $2^{25}/f_{xx}$ | 524.3 ms                  | 1,048.5 ms                |
| 1     | 1     | 1     | $2^{26}/f_{xx}$ | 1,048.5 ms                | 2,097.1 ms                |

**(2) Watchdog timer enable register (WDTE)**

The counter of the watchdog timer is cleared and counting restarted by writing “ACH” to the WDTE register. This register can be read or written in 8-bit units. Reset sets this register to 1AH.

|                  |  |                    |   |   |   |   |   |  |  |  |  |  |  |  |  |
|------------------|--|--------------------|---|---|---|---|---|--|--|--|--|--|--|--|--|
| After reset: 1AH | R/W  | Address: FFFFF6D1H |   |   |   |   |   |  |  |  |  |  |  |  |  |
| 7                | 6  | 5                  | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |  |  |
| WDTE             | <table border="1" style="width: 100%; height: 20px; border-collapse: collapse;"> <tr> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> </tr> </table> |                    |   |   |   |   |   |  |  |  |  |  |  |  |  |
|                  |  |                    |   |   |   |   |   |  |  |  |  |  |  |  |  |

**Cautions**

1. If “ACH” is written to the WDTE register to enable the watchdog timer operation and then a value other than “ACH” is written to the WDTE register, a non-maskable interrupt request signal (INTWDT) or a reset signal (WDTRES) is generated due to watchdog timer overflow, depending on the specification of the WDTM.WDM1 and WDTM.WDM0 bits.
2. When the WDTE register is read or written in 1-bit units, an internal reset signal is output.
3. The read value of the WDTE register is “1AH” before the watchdog timer operates, and “9AH” after it operates. The value read from this register is different from the written value (ACH).

<R>

**11.4 Operation**

The watchdog timer is stopped after reset is released.  
 The WDTM register can be written only once after reset is released.  
 To use the watchdog timer, write the operation mode and the interval time to the WDTM register in 8-bit units. After this, the operation of the watchdog timer cannot be stopped.  
 To not use the watchdog timer, write 00H to the WDTM register.

**11.5 Caution**

The cycle of the non-maskable interrupt request signal (INTWDT) that is generated due to watchdog timer overflow can be calculated from “Interval time set to WDTM register + 2<sup>7</sup> peripheral clock pulse width”, if INTWDT occurs successively without the watchdog timer being cleared.  
 Note that the pulse width until generation of the first interrupt request signal after the watchdog timer has been started is not included.

## CHAPTER 12 A/D CONVERTERS 0 AND 1

### 12.1 Features

- Two 12-bit resolution A/D converter circuits (A/D converters 0 and 1)  
Simultaneous sampling of two circuits possible
- Analog input
  - When comparator is not used  
Total of 10 channels in two circuits  
A/D converter 0: ANI00/ANI05, ANI01 to ANI04 (5 channels)  
A/D converter 1: ANI10/ANI15, ANI11/ANI16, ANI12/ANI17, ANI13, ANI14 (5 channels)
  - When comparator is used  
Total of 6/8 channels in two circuits  
[6 channels (when comparators of low-range and full-range are used)]  
A/D converter 0: ANI00/ANI05, ANI01, ANI02 (3 channels)  
A/D converter 1: ANI10/ANI15, ANI11/ANI16, ANI12/ANI17 (3 channels)  
[8 channels (when comparator of low-range or full-range is used)]  
A/D converter 0: ANI00/ANI05, ANI01, ANI02, ANI03 or ANI04 (4 channels)  
A/D converter 1: ANI10/ANI15, ANI11/ANI16, ANI12/ANI17, ANI13 or ANI14 (4 channels)
- A/D conversion result registers  
12 bits × 16 + 12 bits × 16  
A/D converter 0: AD0CR0 to AD0CR15  
A/D converter 1: AD1CR0 to AD1CR15
- A/D conversion result extension registers  
Can be used only in the extension buffer mode  
12 bits × 5 + 12 bits × 5  
A/D converter 0: AD0ECR0 to AD0ECR4  
A/D converter 1: AD1ECR0 to AD1ECR4
- Operation modes
  - Normal operation modes  
A/D trigger mode  
A/D trigger polling mode  
Hardware trigger mode
  - Extension operation modes  
Conversion channel specification mode  
Extension buffer mode
- Operational amplifiers for input level amplification (×2.5 to ×10)  
These channels can be used only when the operational amplifier for input level amplification is used.  
Total of 4 units in two circuits  
A/D converter 0: ANI05 (1 unit)  
A/D converter 1: ANI15 to ANI17 (3 units)

- Overvoltage detection comparator
  - These channels can be used only when the overvoltage detection comparator is used.
  - Total of 4 units in two circuits
    - A/D converter 0: 1 unit
    - A/D converter 1: 3 units
  - Reference voltage
    - Input voltage range from CREF0L, CREF1L pins (low-range side) =  $0.02AV_{DD} + 0.1$  to  $0.5AV_{DD} - 0.1$  V
    - Input voltage range from CREF0F, CREF1F pins (full-range side) =  $0.02AV_{DD} + 0.1$  to  $0.92AV_{DD} - 0.1$  V
  - An interrupt occurs when an overvoltage is detected. Interrupt requests are output by two overvoltage detection signal pins ANI00/ANI05 (full-range side and low-range side) and as the logical sum (OR) of the overvoltage detection signals of three channels, ANI10/ANI15, ANI11/ANI16, and ANI12/ANI17, or two output signals (full-range side and low-range side) of a logical product (AND).
  - The output of a timer for motor control can be set to a high-impedance state when an overvoltage is detected.
- Successive approximation method
- Operating voltage range
  - $EV_{DD0} = EV_{DD1} = EV_{DD2}$  (V850E/IG3 only) =  $AV_{DD0} = AV_{DD1} = AV_{REFF0} = AV_{REFF1} = 4.0$  to  $5.5$  V

## 12.2 Configuration

The block diagram is shown below.

Figure 12-1. Block Diagram of A/D Converter 0

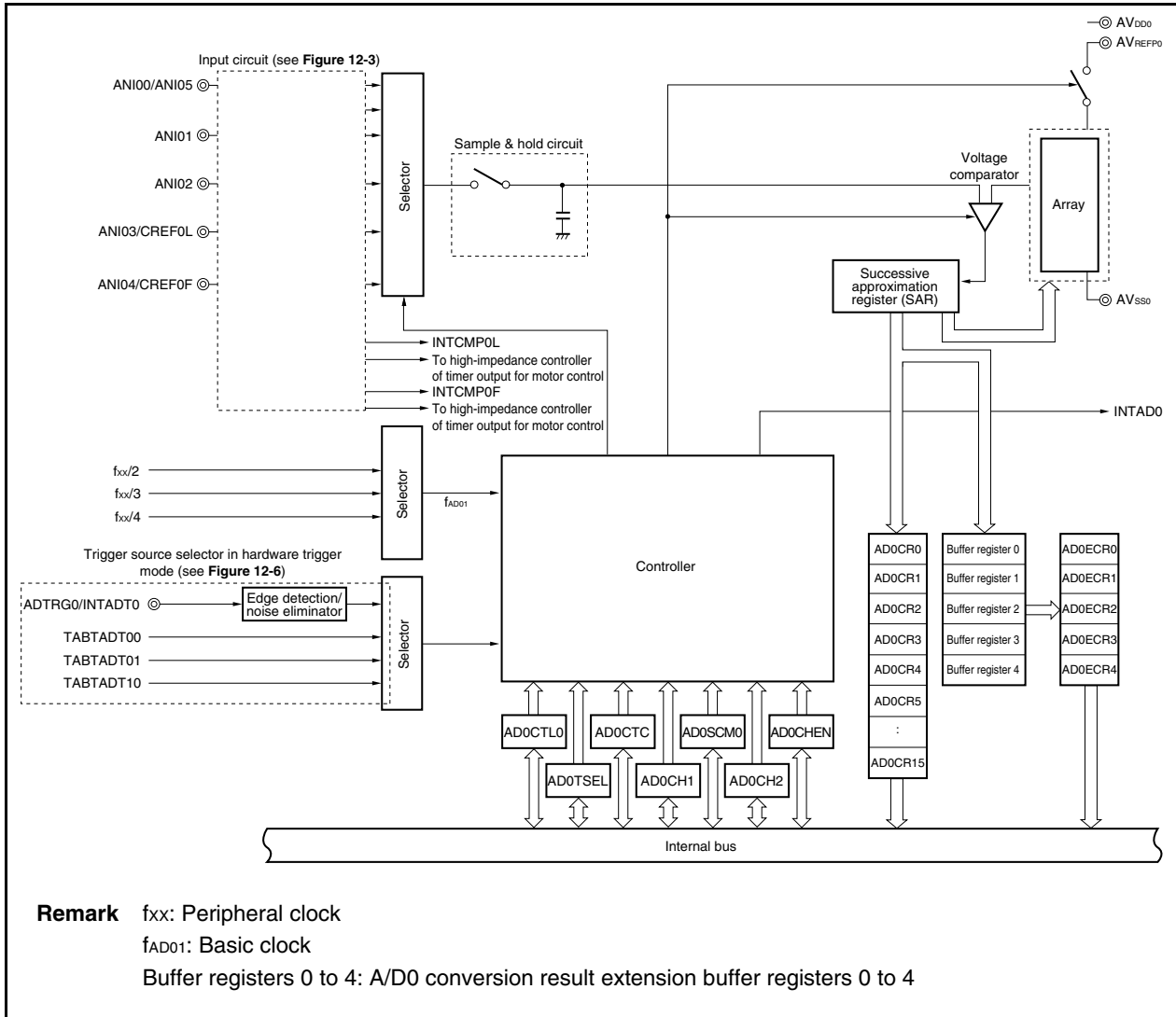
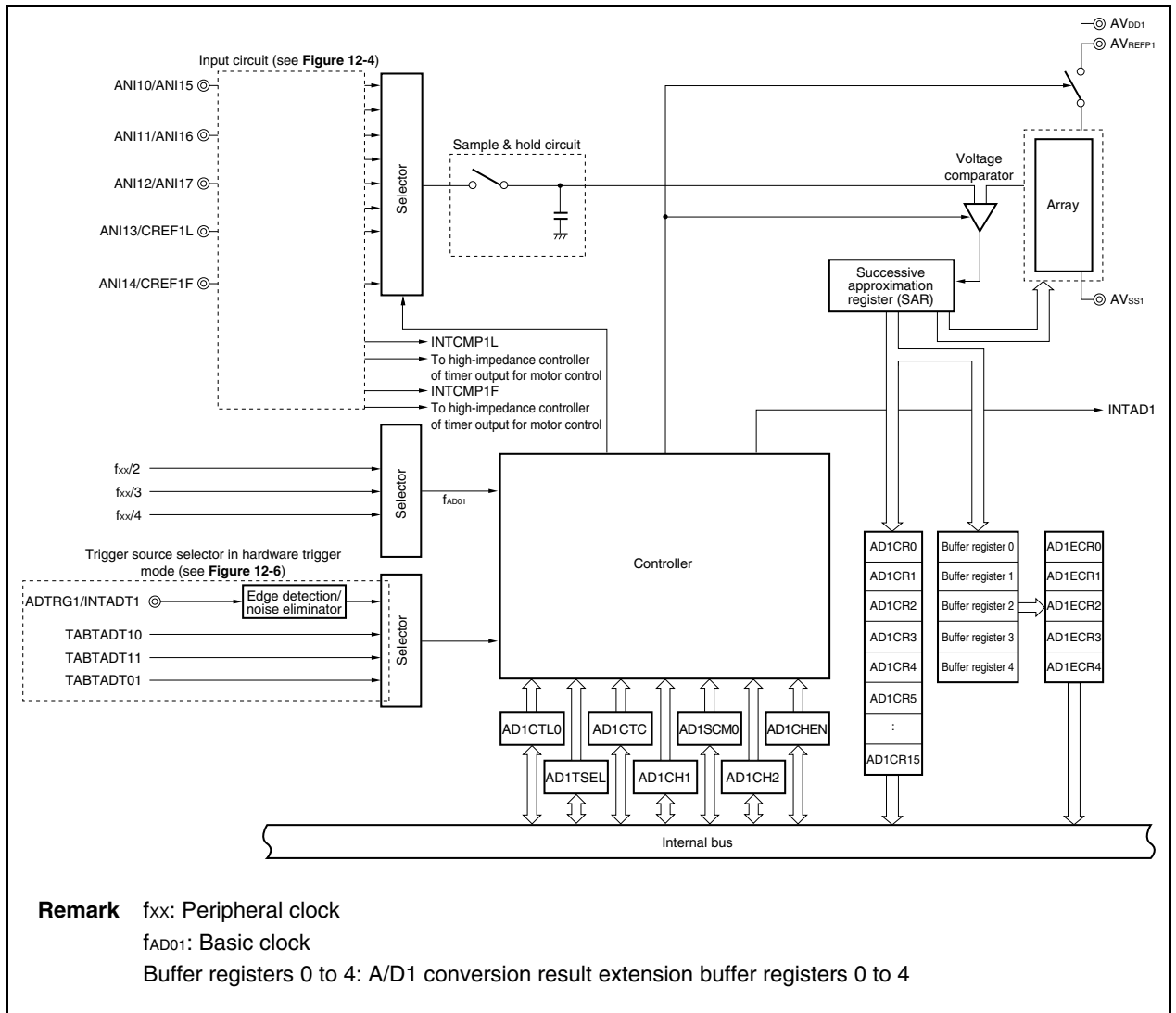
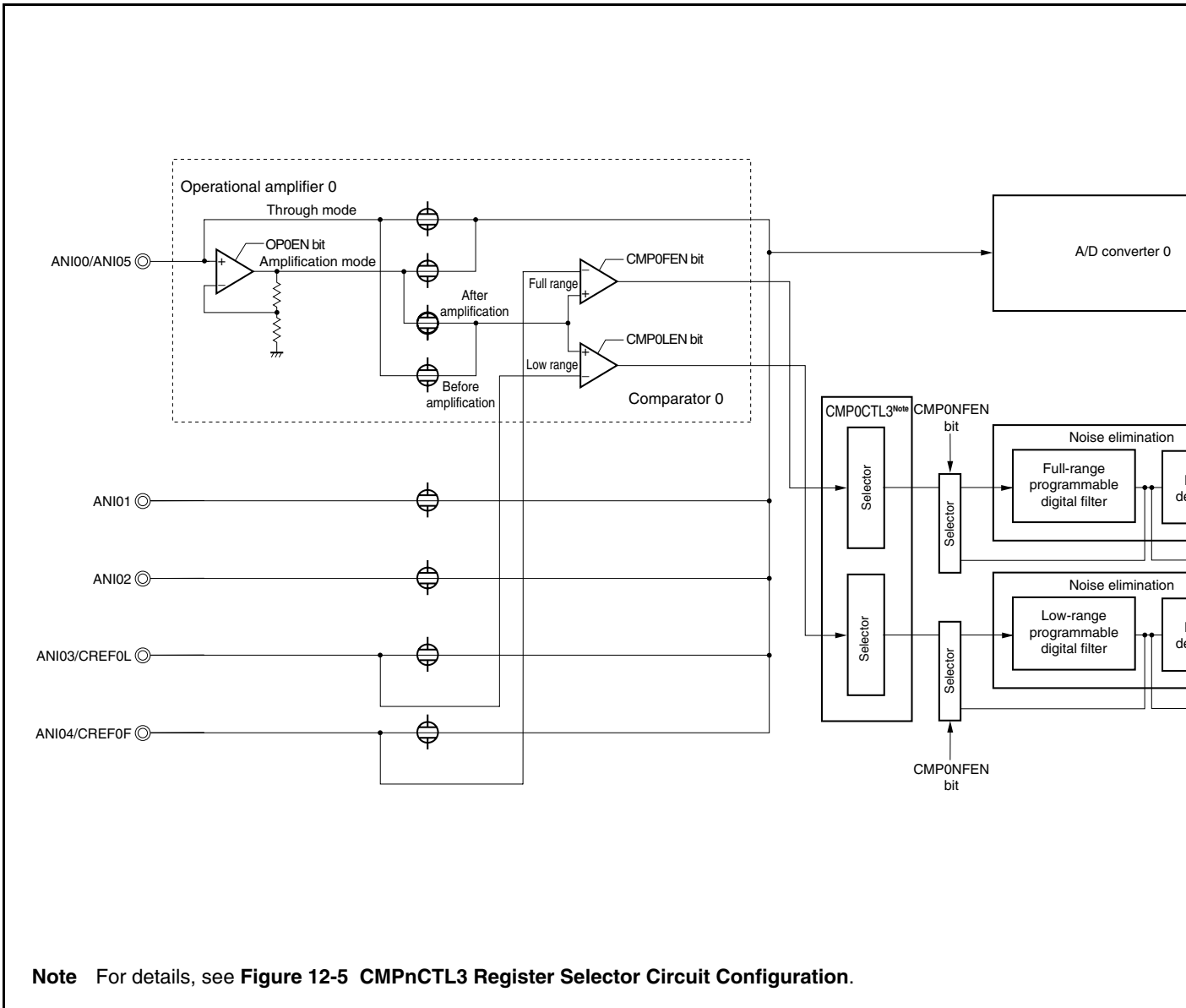


Figure 12-2. Block Diagram of A/D Converter 1



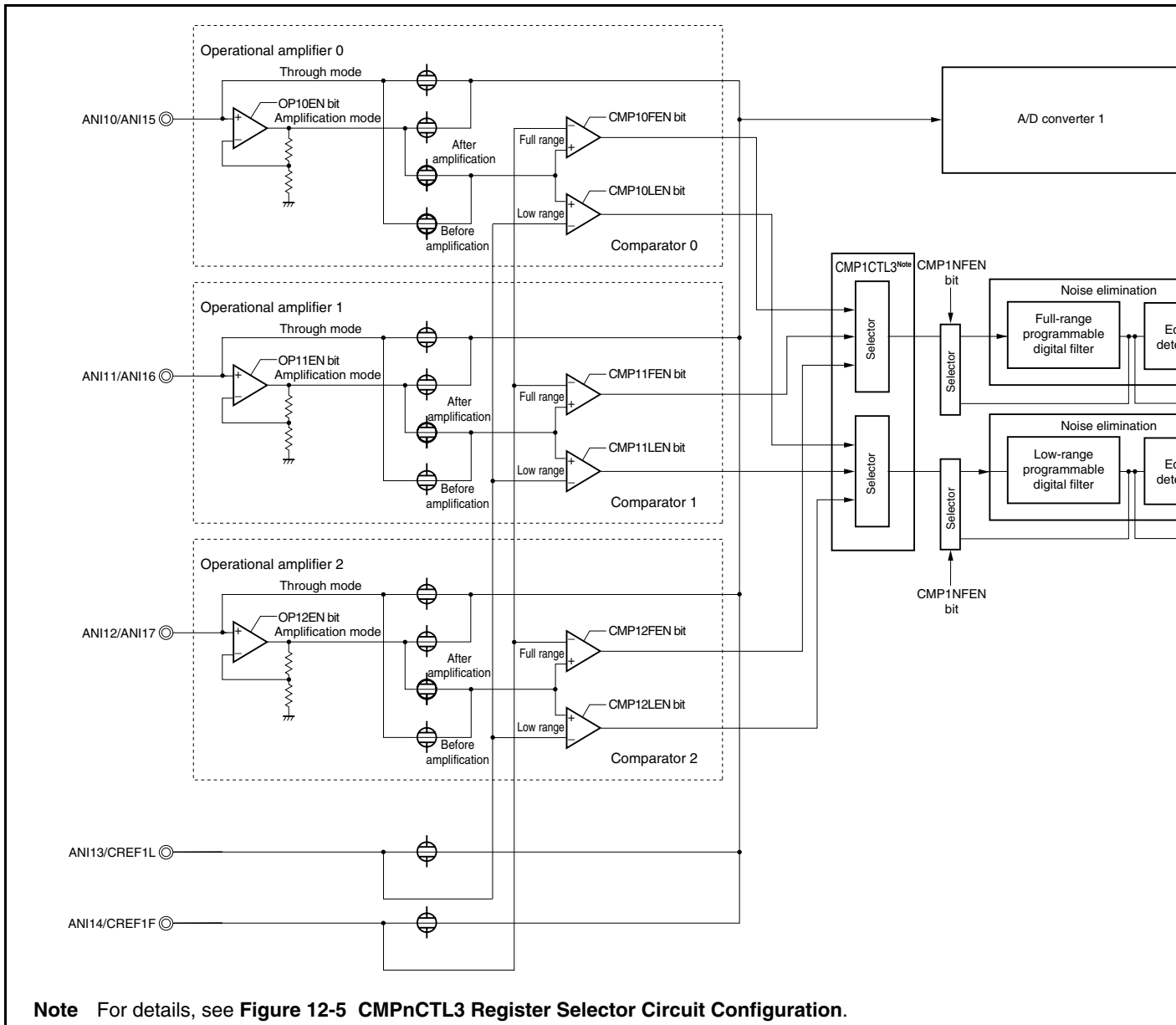
- Cautions 1.** If there is noise at the analog input pins (ANI00 to ANI05, ANI10 to ANI17) or at the A/D converter reference voltage input pins (AV<sub>REFP0</sub>, AV<sub>REFP1</sub>), that noise may generate an illegal conversion result. Software processing will be needed to avoid a negative effect 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.
- 2.** Do not apply a voltage outside the AV<sub>SSn</sub> to AV<sub>REFPn</sub> range to the pins that are used as input pins of A/D converters 0 and 1 (n = 0, 1).

<R> Figure 12-3. Block Diagram of Operational Amplifier for Input Level Amplification and Overvoltage Detection Com





<R> **Figure 12-4. Block Diagram of Operational Amplifier for Input Level Amplification and Overvoltage Detection Comp**



**Note** For details, see Figure 12-5 CMPnCTL3 Register Selector Circuit Configuration.

<R>

Figure 12-5. CMPnCTL3 Register Selector Circuit Configuration

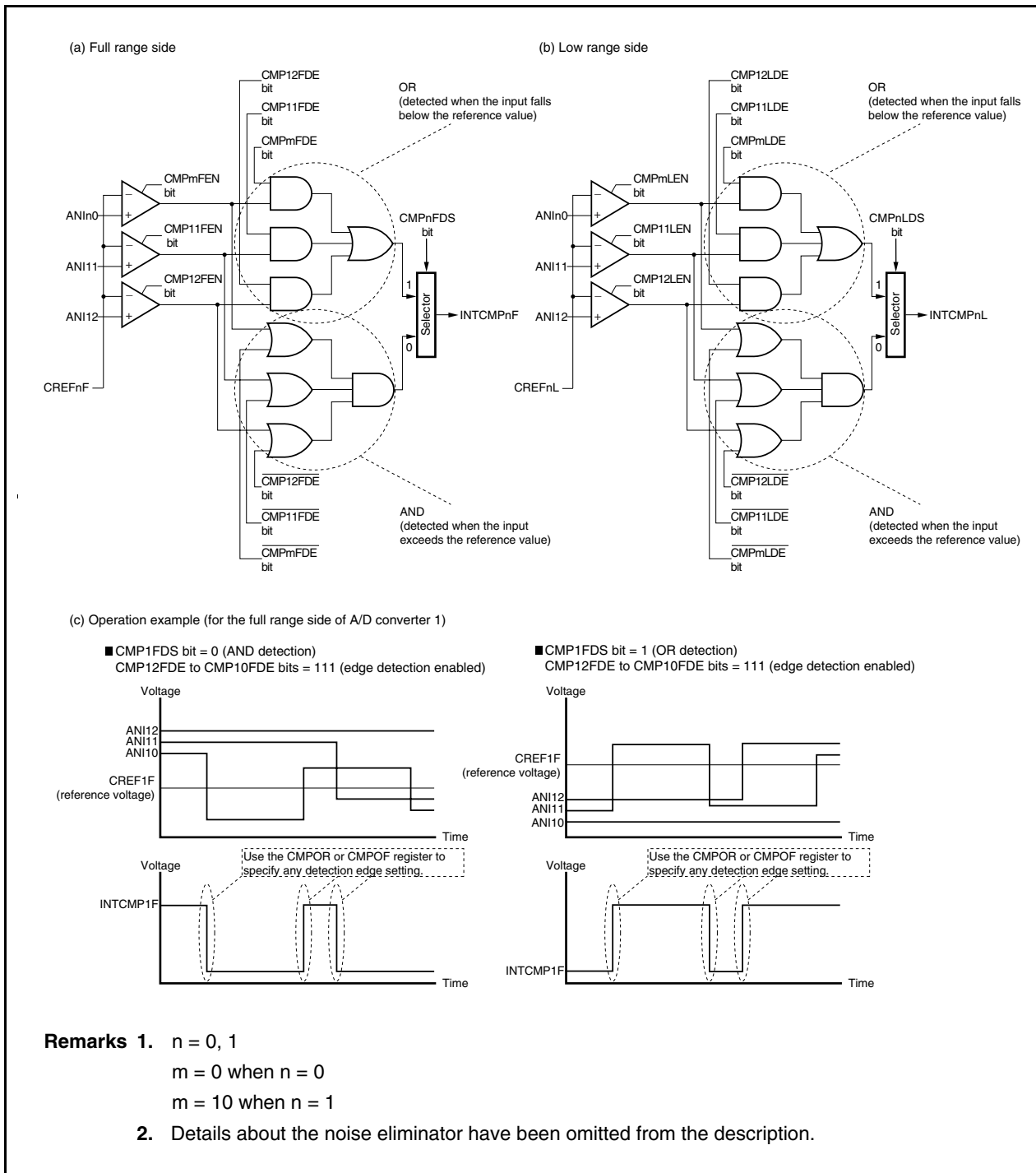
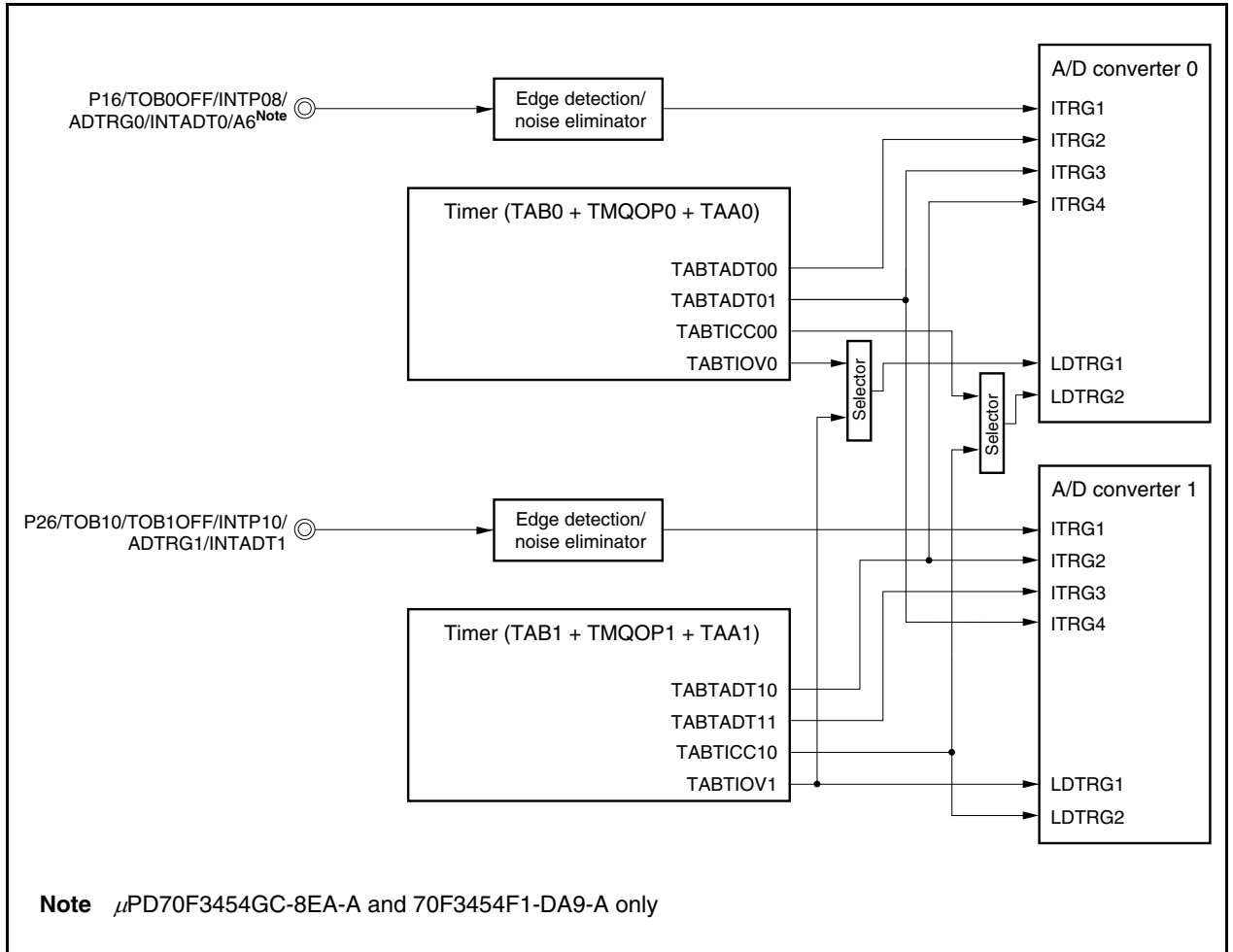


Figure 12-6. Block Diagram of Trigger Source Selector in Hardware Trigger Mode



A/D converters 0 and 1 consist of the following hardware.

**Table 12-1. Configuration of A/D Converters 0 and 1**

| Item              | Configuration   |
|-------------------|---|
| Analog input      | When comparator is not used: ANI00/ANI05, ANI01 to ANI04, ANI10/ANI15 to ANI12/ANI17, ANI13, ANI14 (total of 10 channels in two circuits)<br>When comparator is used (when comparators of low-range and full-range are used): ANI00/ANI05, ANI01, ANI02, ANI10/ANI15 to ANI12/ANI17 (total of 6 channels in two circuits)<br>When comparator is used (when comparator of low-range or full-range is used): ANI00/ANI05, ANI01, ANI02, ANI03 or ANI04, ANI10/ANI15 to ANI12/ANI17, ANI13 or ANI14 (total of 8 channels in two circuits)  |
| Registers         | Successive approximation register (SAR)<br>A/Dn conversion result registers 0 to 15 (ADnCR0 to ADnCR15)<br>A/Dn conversion result registers 0H to 15H (ADnCR0H to ADnCR15H)<br>A/Dn conversion result extension registers 0 to 4 (ADnECR0 to ADnECR4) (only in extension operation mode (extension buffer mode))<br>A/Dn conversion result extension registers 0H to 4H (ADnECR0H to ADnECR4H) (only in extension operation mode (extension buffer mode))   |
| Control registers | A/D converter n scan mode register (ADnSCM)<br>A/D converter n scan mode register L (ADnSCML)<br>A/D converter n scan mode register H (ADnSCMH)<br>A/D converter n conversion time control register (ADnCTC)<br>A/D converter n conversion channel specification register (ADnCHEN)<br>A/D converter n conversion channel specification register L (ADnCHENL)<br>A/D converter n conversion channel specification register H (ADnCHENH)<br>A/D converter n control register (ADnCTL0)<br>A/D converter n trigger select register (ADnTSEL)<br>A/D converter n channel specification register 1 (ADnCH1)<br>A/D converter n channel specification register 2 (ADnCH2)<br>A/D converter n flag register (ADnFLG)<br>A/D converter n flag buffer register (ADnFLGB)<br>A/DLDTRG1 input select register (ADLTS1)<br>A/DLDTRG2 input select register (ADLTS2)<br>A/D converter n clock select register (ADnOCKS)<br>A/D trigger falling edge specification register (ADTF)<br>A/D trigger rising edge specification register (ADTR)<br>Operational amplifier n control register 0 (OPnCTL0)<br>Comparator n control register 0 (CMPnCTL0)<br>Comparator n control register 1 (CMPnCTL1)<br>Comparator n control register 2 (CMPnCTL2)<br>Comparator n control register 3 (CMPnCTL3)<br>Comparator output digital noise elimination register nL (CMPNFCnL)<br>Comparator output digital noise elimination register nF (CMPNFCnF)<br>Comparator output interrupt rising edge specification register (CMPOR)<br>Comparator output interrupt falling edge specification register (CMPOF) |

**Remark** n = 0, 1

**(1) Selector**

The selector selects the analog input pin according to the mode set by the ADnSCM, ADnCTC, ADnCHEN, ADnCTL0, ADnTSEL, ADnCH1, ADnCH2, ADLTS1, ADLTS2, and ADnOCKS registers and sends the input to the sample & hold circuit ( $n = 0, 1$ ).

ANI05, ANI15 to ANI17 are provided with an operational amplifier for input level amplification and an overvoltage detection comparator. The operational amplifier and comparator of each analog input pin can be specified to be on or off. The amplification (gain) of the operational amplifier can be selected from 2.5 to 10 times for ANI05, ANI15 to ANI17.

**(2) Sample & hold circuit**

The sample & hold circuit samples each of the analog input voltages sequentially sent from the input circuit, and sends them to the voltage comparator. When the operational amplifier for input level amplification is used, the gain specified by the OPnCTL0.OPnGA3 to OPnCTL0.OPnGA0 bits  $\times$  the input voltage is sampled. This circuit also holds the sampled analog input voltage during A/D conversion.

**(3) Voltage comparator**

This comparator compares the voltage generated from the voltage tap of the array with the analog input voltage. If the analog input voltage is found to be greater than the reference voltage ( $1/2 AV_{REFPn}$ ) as a result of the comparison, the most significant bit (MSB) of the successive approximation register (SAR) is set. If the analog input voltage is less than the reference voltage ( $1/2 AV_{REFPn}$ ), the MSB of the SAR is reset.

After that, bit 10 of the SAR is automatically set, and the next comparison is made. The voltage tap of the array is selected by the value of bit 11, to which the result has been already set.

Bit 11 = 0: ( $1/4 AV_{REFPn}$ )

Bit 11 = 1: ( $3/4 AV_{REFPn}$ )

The voltage tap of the array and the analog input voltage are compared and bit 10 of the SAR is manipulated according to the result of the comparison.

Analog input voltage  $\geq$  Voltage tap of array: Bit 10 = 1

Analog input voltage  $\leq$  Voltage tap of array: Bit 10 = 0

Comparison is continued like this to bit 0 of the SAR.

**(4) Array**

The array generates the comparison voltage input from an analog input pin.

**(5) Successive approximation register (SAR)**

The SAR is a 12-bit register that sets voltage tap data whose values from the array match the voltage values of the analog input pins, 1 bit at a time starting from the most significant bit (MSB).

If data is set in the SAR all the way to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR (conversion results) are held in A/Dn conversion result registers 0 to 15 (ADnCR0 to ADnCR15) ( $n = 0, 1$ ). In the extension buffer mode, however, the conversion result is stored in A/Dn conversion result extension buffer registers 0 to 4 and, when selection load trigger  $x$  is generated, shifted to and stored in the ADnECR0 to ADnECR4 registers ( $x = 1, 2$ ). When all the specified A/D conversion operations have ended, an A/Dn conversion end interrupt request signal (INTADn) is generated.

**(6) A/Dn conversion result registers 0 to 15 (ADnCR0 to ADnCR15), A/Dn conversion result registers 0H to 15H (ADnCR0H to ADnCR15H) (n = 0, 1)**

The ADnCR0 to ADnCR15 and ADnCR0H to ADnCR15H registers are registers that hold the A/D conversion results. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR) and stored in the higher 12 bits of the ADnCR0 to ADnCR15 registers. The lower 4 bits of these registers are always 0 when read.

The higher 8 bits of the result of A/D conversion are read from the ADnCR0H to ADnCR15H registers.

To read the result of A/D conversion in 16-bit units, specify the ADnCR0 to ADnCR15 registers. To read the higher 8 bits, specify the ADnCR0H to ADnCR15H registers.

**(7) A/Dn conversion result extension registers 0 to 4 (ADnECR0 to ADnECR4), A/Dn conversion result extension registers 0H to 4H (ADnECR0H to ADnECR4H) (n = 0, 1)**

The ADnECR0 to ADnECR4 and ADnECR0H to ADnECR4H registers are registers that hold the A/D conversion results. These registers can be used only in extension buffer mode. When A/D conversion is completed, the A/D conversion result is stored in the A/Dn conversion result extension buffer register. If selection load trigger 1 is generated after that, the A/D conversion result is shifted from A/Dn conversion result extension buffer registers 0 to 2 to the higher 12 bits of the ADnECR0 to ADnECR2 registers for storage. Bits 1 to 3 are always 0 when read. If selection load trigger 2 is generated, the A/D conversion result is shifted from A/Dn conversion result extension buffer registers 3 and 4 to the higher 12 bits of the ADnECR3 and ADnECR4 registers. Bits 1 to 3 are always 0 when read.

The higher 8 bits of the result of A/D conversion are read from the ADnECR0H to ADnECR4H registers.

To read the result of A/D conversion in 16-bit units, specify the ADnECR0 to ADnECR4 registers. To read the higher 8 bits, specify the ADnECR0H to ADnECR4H registers.

**(8) ANI00 to ANI05, ANI10 to ANI17 pins (n = 0, 1)**

The ANI00 to ANI05 and ANI10 to ANI17 pins are analog input pins for A/D converters 0 and 1. They input the analog signals to be A/D converted.

**Caution** Make sure that the voltages input to the ANI00 to ANI05 and ANI10 to ANI17 pins do not exceed the rated values. If a voltage higher than or equal to  $AV_{REFPn}$  or lower than or equal to  $AV_{SSn}$  (even within the range of the absolute maximum ratings) is input to a channel, the conversion value of the channel is undefined, and the conversion values of the other channels may also be affected.

**(9)  $AV_{REFPn}$  pin (n = 0, 1)**

This pin is used for inputting the reference voltage of A/D converters 0 and 1. It converts signals input to the analog input pin to digital signals based on the voltage applied between  $AV_{REFPn}$  and  $AV_{SSn}$  (n = 0, 1).

Always make the potential at this pin the same as that at the  $EV_{DD0}$ ,  $EV_{DD1}$ , and  $EV_{DD2}$  (V850E/IG3 only) pins even when A/D converters 0 and 1 are not used.

The operating voltage range of the  $AV_{REFPn}$  pin is  $EV_{DD0} = EV_{DD1} = EV_{DD2}$  (V850E/IG3 only) =  $AV_{DDn} = AV_{REFPn} = 4.0$  to  $5.5$  V.

**(10)  $AV_{SSn}$  pin (n = 0, 1)**

This is the ground pin of A/D converters 0 and 1. Always make the potential at this pin the same as that at the  $EV_{SS0}$ ,  $EV_{SS1}$ , and  $EV_{SS2}$  (V850E/IG3 only) pins even when A/D converters 0 and 1 are not used.

**(11) AV<sub>DDn</sub> pin (n = 0, 1)**

This pin is the analog power supply pin of A/D converters 0 and 1.

Supply the same potential to the AV<sub>DD0</sub> and AV<sub>DD1</sub> pins.

Always make the potential at this pin the same as that at the EV<sub>DD0</sub>, EV<sub>DD1</sub>, and EV<sub>DD2</sub> (V850E/IG3 only) pins even when A/D converters 0 and 1 are not used.

The operating voltage range of the AV<sub>DDn</sub> pin is EV<sub>DD0</sub> = EV<sub>DD1</sub> = EV<sub>DD2</sub> (V850E/IG3 only) = AV<sub>REFPn</sub> = AV<sub>DDn</sub> = 4.0 to 5.5 V.

**(12) CREFnL, CREFnF pins (n = 0, 1)**

The CREFnL pin supplies the low range of the reference voltage of the comparator for overvoltage detection, and the CREFnF pin supplies the full range of the reference voltage (input voltage range of CREF0L and CREF1L pins =  $0.02AV_{DD} + 0.1$  to  $0.5AV_{DD} - 0.1$  V, input range of CREF0F and CREF1F pins =  $0.02AV_{DD} + 0.1$  to  $0.92AV_{DD} - 0.1$  V).

## 12.3 Control Registers

A/D converters 0 and 1 are controlled by the following registers.

- A/D converter n scan mode register (ADnSCM)
- A/D converter n scan mode register L (ADnSCML)
- A/D converter n scan mode register H (ADnSCMH)
- A/D converter n conversion time control register (ADnCTC)
- A/D converter n conversion channel specification register (ADnCHEN)
- A/D converter n conversion channel specification register L (ADnCHENL)
- A/D converter n conversion channel specification register H (ADnCHENH)
- A/D converter n control register (ADnCTL0)
- A/D converter n trigger select register (ADnTSEL)
- A/D converter n channel specification registers 1 and 2 (ADnCH1, ADnCH2)
- A/D converter n flag register (ADnFLG)
- A/D converter n flag buffer register (ADnFLGB)
- A/DLDTRG1 input select register (ADLTS1)
- A/DLDTRG2 input select register (ADLTS2)
- A/D converter n clock select register (ADnOCKS)
- A/D trigger falling edge specification register (ADTF)
- A/D trigger rising edge specification register (ADTR)
- Operational amplifier n control register 0 (OPnCTL0)
- Comparator n control registers 0 to 3 (CMPnCTL0 to CMPnCTL3)
- Comparator output digital noise elimination registers nL, nF (CMPNFCnL, CMPNFCnF)
- Comparator output interrupt rising edge specification register (CMPOR)
- Comparator output interrupt falling edge specification register (CMPOF)

The following registers are also used.

- A/Dn conversion result registers 0 to 15 (ADnCR0 to ADnCR15)
- A/Dn conversion result registers 0H to 15H (ADnCR0H to ADnCR15H)
- A/Dn conversion result extension registers 0 to 4 (ADnECR0 to ADnECR4)
- A/Dn conversion result extension registers 0H to 4H (ADnECR0H to ADnECR4H)



**(1) A/D converter n scan mode register (ADnSCM)**

The ADnSCM register is a register that specifies the normal operation mode and controls conversion operations.

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

When the higher 8 bits of the ADnSCM register are used as the ADnSCMH register and the lower 8 bits, as the ADnSCML register, these registers can be read or written in 1-bit or 8-bit units. However, bit 14 is read-only.

Reset sets this register to 0000H.

(1/2)

After reset: 0000H R/W Address: AD0SCM FFFF220H, AD1SCM FFFF2A0H

|                      |           |           |    |    |    |            |             |             |           |   |   |   |   |   |                     |   |
|----------------------|-----------|-----------|----|----|----|------------|-------------|-------------|-----------|---|---|---|---|---|---------------------|---|
|                      | <15>      | 14        | 13 | 12 | 11 | 10         | 9           | 8           | 7         | 6 | 5 | 4 | 3 | 2 | 1                   | 0 |
| ADnSCM<br>(n = 0, 1) | ADn<br>CE | ADn<br>CS | 0  | 0  | 0  | ADn<br>PLM | ADn<br>TRG1 | ADn<br>TRG0 | ADn<br>PS | 0 | 0 | 0 | 0 | 0 | 0 <sup>Note 1</sup> | 0 |

**Note 1.** When using A/D converters 1 and 0, be sure to set bit 1 to "1".  
This setting can be performed at the same time as other ADnSCM register bits.

|       |                                  |
|-------|----------------------------------|
| ADnCE | A/D conversion operation control |
| 0     | Stop conversion operation        |
| 1     | Start conversion operation       |

|       |   |
|-------|---|
| ADnCS | Status of A/D converter n <sup>Note 2</sup>   |
| 0     | A/D conversion stopped  |
| 1     | A/D conversion operating<br>(remains "1" even when the channel is changed during successive conversion) |

|                  |         |         |   |
|------------------|---------|---------|---|
| ADnPLM           | ADnTRG1 | ADnTRG0 | Normal operation mode specification     |
| 0                | 0       | 0       | A/D trigger mode                        |
| 0                | 0       | 1       | Hardware trigger mode <sup>Note 3</sup> |
| 1                | 0       | 0       | A/D trigger polling mode                |
| Other than above |         |         | Setting prohibited                      |

|       |                                   |
|-------|-----------------------------------|
| ADnPS | A/D power save mode specification |
| 0     | A/D power save mode               |
| 1     | A/D operational mode              |

**Notes 2.** The ADnCS bit is set to 1 five basic clocks ( $f_{AD01}$ ) after the ADnCE bit has been set to 1 and A/D conversion has been started.

A/D conversion is started when a trigger signal, such as one from a timer, is input in the hardware trigger mode, conversion channel specification mode, or extension buffer mode. In the A/D trigger mode and A/D trigger polling mode, it is started when the ADnCE bit is 1.

**3.** In the extended operation mode (conversion channel specification mode or extension buffer mode), be sure to set the hardware trigger mode.

- Cautions**
1. In the A/D trigger mode or the A/D trigger polling mode, conversion is triggered when 1 is written to the ADnCE bit.  
 In the hardware trigger mode, the conversion channel specification mode, or the extension buffer mode, the trigger signal wait state starts when 1 is written to the ADnCE bit.  
 The ADnCE bit is not cleared to 0 even after the A/Dn conversion end interrupt request signal (INTADn) is generated in all modes. To stop the A/D conversion operation, therefore, write 0 to the ADnCE bit.
  2. If the ADnSCM register is written during A/D conversion operation (ADnCS bit = 1), the operation is performed as follows in each mode. The corresponding conversion result register is undefined during A/D conversion operation.
    - In A/D trigger mode, A/D trigger polling mode  
 A/D conversion is stopped and executed again from the beginning.
    - In hardware trigger mode, conversion channel specification mode, extension buffer mode  
 A/D conversion is stopped and the trigger standby state is restored again.
  3. Make sure that time of at least five basic clocks ( $f_{AD01}$ ) passes before successively writing data to the ADnSCM register when the conversion operation is enabled (ADnCE bit = 1). Otherwise, the register may not be set correctly.  
 The register can be successively written if the ADnCE bit is set to 1 after the ADnSCM register is written when ADnCE bit = 0.
  4. The ADnCS bit remains set (1) when the conversion channel is changed during successive conversion.
  5. It is recommended to set the A/D power save mode (ADnPS bit = 0) when the A/D converter is not used.
  6. The setting procedure is as follows when an A/D conversion operation is started (after reset release and after recovery from the A/D power save mode (ADnPS bit = 0)).
    - <1> Select an input clock ( $f_{AD01}$ ) by using the ADnOCKS register and set the ADnOCKSEN bit to 1 (enable supplying the operating clock to A/D converter n).
    - <2> Set the A/D conversion time by using the ADnCTC.ADnFR3 to ADnFR0 bits.
    - <3> Set the ADnPS bit to 1 (A/D operation mode).
    - <4> Wait for 1  $\mu$ s or longer after <3>.
    - <5> Initialize A/D converters 0 and 1.
    - <6> Set the ADnCE bit to 1 (enable conversion operation).
  7. The setting procedure is as follows when an A/D conversion operation is stopped.
    - <1> Set the ADnCE bit to 0 (stop conversion operation) (retaining ADnPS bit = 1).
    - <2> Set the ADnPS bit to 0 (A/D power save mode).
    - <3> Set the ADnOCKS.ADnOCKSEN bit to 0 (stop supplying the operating clock to A/D converter n).
  8. It is recommended to set the A/D power save mode even in the IDLE and STOP modes.  
 Follow the setting procedure in Caution 6 above when releasing the IDLE or STOP mode by using the reset signal.
  9. Be sure to set bits 0, 2 to 6 and 11 to 13 to "0".

&lt;R&gt;

**(2) A/D converter n conversion time control register (ADnCTC)**

The ADnCTC register is a register that specifies the number of A/D conversion clocks and A/D conversion time. 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: AD0CTC FFFFF222H, AD1CTC FFFFF2A2H

|                      |   |   |   |   |        |        |        |        |
|----------------------|---|---|---|---|--------|--------|--------|--------|
|                      | 7 | 6 | 5 | 4 | 3      | 2      | 1      | 0      |
| ADnCTC<br>(n = 0, 1) | 0 | 0 | 0 | 0 | ADnFR3 | ADnFR2 | ADnFR1 | ADnFR0 |

**Cautions**

1. See Table 12-2 Number of A/D Conversion Clocks and A/D Conversion Time for the ADnFR3 to ADnFR0 bits.
2. Set the ADnFR3 to ADnFR0 bits when the ADnSCM.ADnCE bit = 0 (conversion operation is stopped).
3. Be sure to set bits 4 to 7 to “0”.

**Table 12-2. Number of A/D Conversion Clocks and A/D Conversion Time**

| ADnFR3 | ADnFR2 | ADnFR1 | ADnFR0 | Number of A/D Conversion Clocks <sup>Note 1</sup> | A/D Conversion Time ( $\mu$ s) <sup>Note 2</sup> |   |
|--------|--------|--------|--------|---|--|---|
|        |        |        |        |   | $f_{AD01} = 16$ MHz<br>( $f_{XX} = 64$ MHz)      | $f_{AD01} = 12$ MHz<br>( $f_{XX} = 48$ MHz) |
| 0      | 0      | 0      | 0      | 89  | 5.56   | 7.42  |
| 0      | 0      | 0      | 1      | 88  | 5.50   | 7.33  |
| 0      | 0      | 1      | 0      | 57  | 3.56   | 4.75  |
| 0      | 0      | 1      | 1      | 56  | 3.50   | 4.67  |
| 0      | 1      | 0      | 0      | 41  | 2.56   | 3.42  |
| 0      | 1      | 0      | 1      | 40  | 2.50   | 3.33  |
| 0      | 1      | 1      | 0      | 35  | 2.19   | 2.92  |
| 0      | 1      | 1      | 1      | 34  | 2.13   | 2.83  |
| 1      | 0      | 0      | 0      | 34  | 2.13   | 2.83  |
| 1      | 0      | 0      | 1      | 33  | 2.06   | 2.75  |
| 1      | 0      | 1      | 0      | 33  | 2.06   | 2.75  |
| 1      | 0      | 1      | 1      | 32  | 2.00   | 2.67  |
| 1      | 1      | 0      | 0      | 32  | 2.00   | 2.67  |
| 1      | 1      | 0      | 1      | 31  | Setting prohibited                               | 2.58  |
| 1      | 1      | 1      | 0      | 31  | Setting prohibited                               | 2.58  |
| 1      | 1      | 1      | 1      | 30  | Setting prohibited                               | 2.50  |

- Notes**
1. The number of clocks ( $f_{AD01}$ ) from the start to the end of A/D conversion.  
The number of clocks ( $f_{AD01}$ ) per conversion during successive conversion (1-channel conversion (repeat), multiple channel conversion, or multiple channel conversion (repeat)) is the same.
  2. Set the A/D conversion time in a range of 2 to 7.42  $\mu$ s.  
A/D Conversion time =  $1/f_{AD01} \times$  Number of A/D conversion clocks

**(3) A/D converter n conversion channel specification register (ADnCHEN)**

The ADnCHEN register is a register that specifies the analog input pin, number of conversion times, and conversion result register.

This register is used to specify an analog input pin in the A/D trigger mode, A/D trigger polling mode, and hardware trigger mode. The ADnCRm register corresponds to an analog input pin on a one-to-one basis. Use the bits (AD0CHEN00 to AD0CHEN05 and AD1CHEN00 to AD1CHEN07) corresponding to the ANI00 to ANI05 and ANI10 to ANI17 pins. If two or more analog input pins are specified, they are sequentially selected, starting from the one with the lowest number, for conversion (when AD1CHEN register = 004DH: ANI10 → ANI12 → ANI13 → ANI16). If an analog input pin that is not specified is skipped during successive conversion. In the conversion channel specification mode, specify the number of times of conversion and a conversion result register. Specify an analog input pin by using the ADnCH1 register. A value set to the lower bits of the ADnCHEN register, justified to the lowest bit, is the number of times of conversion. These bits correspond to the ADnCRm and ADnCHmH registers on a one-to-one basis.

Because the ADnCHEN register is of master/slave configuration, a new analog input pin can be set to the master register during A/D conversion operation. The set value of the master register is transferred to a slave register after completion of A/D conversion (after the A/Dn conversion end interrupt request signal (INTADn) is generated).

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

When the higher 8 bits of the ADnCHEN register are used as the ADnCHENH register and the lower 8 bits, as the ADnCHENL register, these registers can be read or written in 1-bit or 8-bit units.

Reset sets this register to 0000H.

After reset: 0000H R/W Address: AD0CHEN FFFFF224H, AD1CHEN FFFFF2A4H

|                       |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |
|-----------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                       | 15          | 14          | 13          | 12          | 11          | 10          | 9           | 8           | 7           | 6           | 5           | 4           | 3           | 2           | 1           | 0           |
| ADnCHEN<br>(n = 0, 1) | ADn<br>CHEN | ADn<br>CHEN | ADn<br>CHEN | ADn<br>CHEN | ADn<br>CHEN | ADn<br>CHEN | ADn<br>CHEN | ADn<br>CHEN | ADn<br>CHEN | ADn<br>CHEN | ADn<br>CHEN | ADn<br>CHEN | ADn<br>CHEN | ADn<br>CHEN | ADn<br>CHEN | ADn<br>CHEN |
|                       | 15          | 14          | 13          | 12          | 11          | 10          | 9           | 8           | 7           | 6           | 5           | 4           | 3           | 2           | 1           | 0           |

**Remark** See **Table 12-3 Specifying Analog Input Pin in A/D Trigger Mode, A/D Trigger Polling Mode, and Hardware Trigger Mode** for how to specify an analog input pin in the A/D trigger mode, A/D trigger polling mode, and hardware trigger mode. For how to specify the number of times of conversion and the A/D conversion result register in the conversion channel specification mode, see **Table 12-4 Correspondence Among Set Value of ADnCHEN Register, Number of Times of Conversion, and A/D Conversion Result Register in Conversion Channel Specification Mode**.

- Cautions**
1. The A/D conversion operation is prohibited when the ADnCHEN register = 0000H. If the ADnCHEN register = 0000H, the operation is the same as when the ADnCHEN register = 0001H.
  2. Do not write the ADnCHEN register when the ADnSCM.ADnPS bit = 0. If it is written, the CPU deadlocks.
  3. To change the setting of the ADnCHEN register when the ADnSCM.ADnCE bit = 1 in the hardware trigger mode, be sure to set the ADnCE bit to 0.

**Table 12-3. Specifying Analog Input Pin in A/D Trigger Mode, A/D Trigger Polling Mode, and Hardware Trigger Mode**

| ADnCHENm Bit | Specification of Analog Input Pin   |
|--------------|-------------------------------------|
| 0            | Specifying ANInk pin is prohibited. |
| 1            | Specifying ANInk pin is enabled.    |

**Remark** A/D converter 0: n = 0, k = 0 to 5, m = 0 to 15  
A/D converter 1: n = 1, k = 0 to 7, m = 0 to 15

**Table 12-4. Correspondence Among Set Value of ADnCHEN Register, Number of Times of Conversion, and A/D Conversion Result Register in Conversion Channel Specification Mode**

| ADnCHEN Register Value | Number of Times of Conversion | A/D Conversion Result Register |                     |
|------------------------|-------------------------------|--------------------------------|---------------------|
|                        |                               | ADnCR0                         | ADnCR0H             |
| 0001H                  | 1                             | ADnCR0                         | ADnCR0H             |
| 0003H                  | 2                             | ADnCR0, ADnCR1                 | ADnCR0H, ADnCR1H    |
| 0007H                  | 3                             | ADnCR0 to ADnCR2               | ADnCR0H to ADnCR2H  |
| 000FH                  | 4                             | ADnCR0 to ADnCR3               | ADnCR0H to ADnCR3H  |
| 001FH                  | 5                             | ADnCR0 to ADnCR4               | ADnCR0H to ADnCR4H  |
| 003FH                  | 6                             | ADnCR0 to ADnCR5               | ADnCR0H to ADnCR5H  |
| 007FH                  | 7                             | ADnCR0 to ADnCR6               | ADnCR0H to ADnCR6H  |
| 00FFH                  | 8                             | ADnCR0 to ADnCR7               | ADnCR0H to ADnCR7H  |
| 01FFH                  | 9                             | ADnCR0 to ADnCR8               | ADnCR0H to ADnCR8H  |
| 03FFH                  | 10                            | ADnCR0 to ADnCR9               | ADnCR0H to ADnCR9H  |
| 07FFH                  | 11                            | ADnCR0 to ADnCR10              | ADnCR0H to ADnCR10H |
| 0FFFH                  | 12                            | ADnCR0 to ADnCR11              | ADnCR0H to ADnCR11H |
| 1FFFH                  | 13                            | ADnCR0 to ADnCR12              | ADnCR0H to ADnCR12H |
| 3FFFH                  | 14                            | ADnCR0 to ADnCR13              | ADnCR0H to ADnCR13H |
| 7FFFH                  | 15                            | ADnCR0 to ADnCR14              | ADnCR0H to ADnCR14H |
| FFFFH                  | 16                            | ADnCR0 to ADnCR15              | ADnCR0H to ADnCR15H |
| Others                 | Setting prohibited            |                                |                     |

**Caution** An analog input pin is specified by the ADnCH1 register in the conversion channel specification mode.

**Remark** n = 0, 1

**(4) A/Dn conversion result registers 0 to 15, 0H to 15H (ADnCR0 to ADnCR15, ADnCR0H to ADnCR15H)**

The ADnCRm and ADnCRmH registers are registers that hold the A/D conversion results in the A/D trigger mode, A/D trigger polling mode, hardware trigger mode, or conversion channel specification mode. Sixteen of these registers are provided per circuit, and two circuits are available. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR) and stored in the higher 12 bits of the ADnCRm register. The lower 4 bits of these registers are always 0 when read.

The higher 8 bits of A/D conversion result are read to the ADnCRmH register.

These registers can only be read in 16-bit or 8-bit units. When the A/D conversion results are read in 16-bit units, the ADnCRm register is specified, and when the higher 8 bits are read, the ADnCRmH register is specified.

Reset sets these registers to 0000H.

**Remark** While the result of A/D conversion is stored in the ADnCRm register, a read access to the same register is held pending. The pending read access is executed after the A/D conversion result is stored. Similarly, storing the result of A/D conversion in the ADnCRm register is held pending while a read access to that register is made. The pending A/D conversion result storing processing is executed after completion of the read access.

After reset: 0000H R Address: AD0CR0 FFFF200H, AD0CR1 FFFF202H,  
 AD0CR2 FFFF204H, AD0CR3 FFFF206H,  
 AD0CR4 FFFF208H, AD0CR5 FFFF20AH,  
 AD0CR6 FFFF20CH, AD0CR7 FFFF20EH,  
 AD0CR8 FFFF210H, AD0CR9 FFFF212H,  
 AD0CR10 FFFF214H, AD0CR11 FFFF216H,  
 AD0CR12 FFFF218H, AD0CR13 FFFF21AH,  
 AD0CR14 FFFF21CH, AD0CR15 FFFF21EH,  
 AD1CR0 FFFF280H, AD1CR1 FFFF282H,  
 AD1CR2 FFFF284H, AD1CR3 FFFF286H,  
 AD1CR4 FFFF288H, AD1CR5 FFFF28AH,  
 AD1CR6 FFFF28CH, AD1CR7 FFFF28EH,  
 AD1CR8 FFFF290H, AD1CR9 FFFF292H,  
 AD1CR10 FFFF294H, AD1CR11 FFFF296H,  
 AD1CR12 FFFF298H, AD1CR13 FFFF29AH,  
 AD1CR14 FFFF29CH, AD1CR15 FFFF29EH

|                                       |            |            |            |            |            |            |            |            |            |            |            |            |            |   |   |   |   |
|---------------------------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|---|---|---|---|
|                                       | 15         | 14         | 13         | 12         | 11         | 10         | 9          | 8          | 7          | 6          | 5          | 4          | 3          | 2 | 1 | 0 |   |
| ADnCRm<br>(n = 0, 1)<br>(m = 0 to 15) | ADn<br>CRm | ADn<br>CRm | ADn<br>CRm | ADn<br>CRm | ADn<br>CRm | ADn<br>CRm | ADn<br>CRm | ADn<br>CRm | ADn<br>CRm | ADn<br>CRm | ADn<br>CRm | ADn<br>CRm | ADn<br>CRm | 0 | 0 | 0 | 0 |
|                                       | 11         | 10         | 9          | 8          | 7          | 6          | 5          | 4          | 3          | 2          | 1          | 0          |            |   |   |   |   |

After reset: 0000H R Address: AD0CR0H FFFF201H, AD0CR1H FFFF203H,  
 AD0CR2H FFFF205H, AD0CR3H FFFF207H,  
 AD0CR4H FFFF209H, AD0CR5H FFFF20BH,  
 AD0CR6H FFFF20DH, AD0CR7H FFFF20FH,  
 AD0CR8H FFFF211H, AD0CR9H FFFF213H,  
 AD0CR10H FFFF215H, AD0CR11H FFFF217H,  
 AD0CR12H FFFF219H, AD0CR13H FFFF21BH,  
 AD0CR14H FFFF21DH, AD0CR15H FFFF21FH,  
 AD1CR0H FFFF281H, AD1CR1H FFFF283H,  
 AD1CR2H FFFF285H, AD1CR3H FFFF287H,  
 AD1CR4H FFFF289H, AD1CR5H FFFF28BH,  
 AD1CR6H FFFF28DH, AD1CR7H FFFF28FH,  
 AD1CR8H FFFF291H, AD1CR9H FFFF293H,  
 AD1CR10H FFFF295H, AD1CR11H FFFF297H,  
 AD1CR12H FFFF299H, AD1CR13H FFFF29BH,  
 AD1CR14H FFFF29DH, AD1CR15H FFFF29FH

|  |          |          |         |         |         |         |         |         |
|--|----------|----------|---------|---------|---------|---------|---------|---------|
|  | 7        | 6        | 5       | 4       | 3       | 2       | 1       | 0       |
| ADnCRmH<br>(n = 0, 1)<br>(m = 0 to 15) | ADnCRm11 | ADnCRm10 | ADnCRm9 | ADnCRm8 | ADnCRm7 | ADnCRm6 | ADnCRm5 | ADnCRm4 |

The correspondence between the analog input pins and the A/D conversion result registers in the A/D trigger mode, A/D trigger polling mode, hardware trigger mode, and conversion channel specification mode is shown below.

**Table 12-5. Correspondence Between Analog Input Pins and A/D Conversion Result Registers in A/D Trigger Mode, A/D Trigger Polling Mode, Hardware Trigger Mode**

| A/D Converter   | Analog Input Pin | A/D Conversion Result Register |
|-----------------|------------------|--------------------------------|
| A/D converter 0 | ANI00            | AD0CR0, AD0CR0H                |
|                 | ANI01            | AD0CR1, AD0CR1H                |
|                 | ANI02            | AD0CR2, AD0CR2H                |
|                 | ANI03            | AD0CR3, AD0CR3H                |
|                 | ANI04            | AD0CR4, AD0CR4H                |
|                 | ANI05            | AD0CR5, AD0CR5H                |
| A/D converter 1 | ANI10            | AD1CR0, AD1CR0H                |
|                 | ANI11            | AD1CR1, AD1CR1H                |
|                 | ANI12            | AD1CR2, AD1CR2H                |
|                 | ANI13            | AD1CR3, AD1CR3H                |
|                 | ANI14            | AD1CR4, AD1CR4H                |
|                 | ANI15            | AD1CR5, AD1CR5H                |
|                 | ANI16            | AD1CR6, AD1CR6H                |
|                 | ANI17            | AD1CR7, AD1CR7H                |



**Table 12-6. Correspondence Between Analog Input Pins and A/D Conversion Result Registers in Conversion Channel Specification Mode**

| ADnCHEN Register Set Value | Analog Input Pin                                   | A/D Conversion Result Register |                     |
|----------------------------|--|--------------------------------|---------------------|
|                            |  | ADnCR0                         | ADnCR0H             |
| 0001H                      | Set by ADnCH1.ADnTRGCH12 to ADnCH1.ADnTRGCH10 bits | ADnCR0, ADnCR1                 | ADnCR0H, ADnCR1H    |
| 0003H                      |  | ADnCR0 to ADnCR2               | ADnCR0H to ADnCR2H  |
| 0007H                      |  | ADnCR0 to ADnCR3               | ADnCR0H to ADnCR3H  |
| 000FH                      |  | ADnCR0 to ADnCR4               | ADnCR0H to ADnCR4H  |
| 001FH                      |  | ADnCR0 to ADnCR5               | ADnCR0H to ADnCR5H  |
| 003FH                      |  | ADnCR0 to ADnCR6               | ADnCR0H to ADnCR6H  |
| 007FH                      |  | ADnCR0 to ADnCR7               | ADnCR0H to ADnCR7H  |
| 00FFH                      |  | ADnCR0 to ADnCR8               | ADnCR0H to ADnCR8H  |
| 01FFH                      |  | ADnCR0 to ADnCR9               | ADnCR0H to ADnCR9H  |
| 03FFH                      |  | ADnCR0 to ADnCR10              | ADnCR0H to ADnCR10H |
| 07FFH                      |  | ADnCR0 to ADnCR11              | ADnCR0H to ADnCR11H |
| 0FFFH                      |  | ADnCR0 to ADnCR12              | ADnCR0H to ADnCR12H |
| 1FFFH                      |  | ADnCR0 to ADnCR13              | ADnCR0H to ADnCR13H |
| 3FFFH                      |  | ADnCR0 to ADnCR14              | ADnCR0H to ADnCR14H |
| 7FFFH                      |  | ADnCR0 to ADnCR15              | ADnCR0H to ADnCR15H |
| FFFFH                      |  |                                |                     |
| 0001H                      | Set by ADnCH1.ADnTRGCH16 to ADnCH1.ADnTRGCH14 bits | ADnCR0                         | ADnCR0H             |
| 0003H                      |  | ADnCR0, ADnCR1                 | ADnCR0H, ADnCR1H    |
| 0007H                      |  | ADnCR0 to ADnCR2               | ADnCR0H to ADnCR2H  |
| 000FH                      |  | ADnCR0 to ADnCR3               | ADnCR0H to ADnCR3H  |
| 001FH                      |  | ADnCR0 to ADnCR4               | ADnCR0H to ADnCR4H  |
| 003FH                      |  | ADnCR0 to ADnCR5               | ADnCR0H to ADnCR5H  |
| 007FH                      |  | ADnCR0 to ADnCR6               | ADnCR0H to ADnCR6H  |
| 00FFH                      |  | ADnCR0 to ADnCR7               | ADnCR0H to ADnCR7H  |
| 01FFH                      |  | ADnCR0 to ADnCR8               | ADnCR0H to ADnCR8H  |
| 03FFH                      |  | ADnCR0 to ADnCR9               | ADnCR0H to ADnCR9H  |
| 07FFH                      |  | ADnCR0 to ADnCR10              | ADnCR0H to ADnCR10H |
| 0FFFH                      |  | ADnCR0 to ADnCR11              | ADnCR0H to ADnCR11H |
| 1FFFH                      |  | ADnCR0 to ADnCR12              | ADnCR0H to ADnCR12H |
| 3FFFH                      |  | ADnCR0 to ADnCR13              | ADnCR0H to ADnCR13H |
| 7FFFH                      |  | ADnCR0 to ADnCR14              | ADnCR0H to ADnCR14H |
| FFFFH                      |  | ADnCR0 to ADnCR15              | ADnCR0H to ADnCR15H |
| Others                     | Setting prohibited                                 |                                |                     |

**Remark** n = 0, 1

**(5) A/D converter n control register (ADnCTL0)**

The ADnCTL0 register is a register that specifies the operation 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: AD0CTL0 FFFFF230H, AD1CTL0 FFFFF2B0H

|                       |   |   |   |   |   |   |        |        |
|-----------------------|---|---|---|---|---|---|--------|--------|
|                       | 7 | 6 | 5 | 4 | 3 | 2 | 1      | 0      |
| ADnCTL0<br>(n = 0, 1) | 0 | 0 | 0 | 0 | 0 | 0 | ADnMD1 | ADnMD0 |

| ADnMD1 | ADnMD0 | Extended operating mode specification |
|--------|--------|---------------------------------------|
| 0      | 0      | Normal operating mode                 |
| 0      | 1      | Setting prohibited                    |
| 1      | 0      | Conversion channel specification mode |
| 1      | 1      | Extension buffer mode                 |

- Cautions**
1. Set the ADnMD1 and ADnMD0 bits when the ADnSCM.ADnCE bit = 0 (conversion operation is stopped) (the same value can be written to these bits when the ADnCE bit = 1 (conversion operation is enabled)).
  2. In the conversion channel specification mode and extension buffer mode, start of A/D conversion is delayed up to 1.5 basic clocks ( $f_{AD01}$ ) as compared with the normal operating mode.
  3. Be sure to set the hardware trigger mode in the conversion channel specification mode and extension buffer mode.

**(6) A/D converter n trigger select register (ADnTSEL)**

The ADnTSEL register is a register that specifies trigger in the hardware trigger mode and conversion channel specification mode, and trigger (selection trigger 1, selection trigger 2, selection load trigger 1, and selection load trigger 2) in the extension buffer mode.

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

Reset sets this register to 10H.

After reset: 10H R/W Address: AD0TSEL FFFFFFF231H, AD1TSEL FFFFFFF2B1H

|                       |                                |   |                                 |                                 |                                |   |                                 |                                 |
|-----------------------|--------------------------------|---|---------------------------------|---------------------------------|--------------------------------|---|---------------------------------|---------------------------------|
|                       | 7                              | 6 | 5                               | 4                               | 3                              | 2 | 1                               | 0                               |
| ADnTSEL<br>(n = 0, 1) | ADn <sup>Note</sup><br>LDTSEL2 | 0 | ADn <sup>Note</sup><br>TRGSEL21 | ADn <sup>Note</sup><br>TRGSEL20 | ADn <sup>Note</sup><br>LDTSEL1 | 0 | ADn <sup>Note</sup><br>TRGSEL11 | ADn <sup>Note</sup><br>TRGSEL10 |

|                            |  |
|----------------------------|--|
| ADnLDTSEL2 <sup>Note</sup> | Specification of selection load trigger 2 for ADnECR3, ADnECR4 registers |
| 0                          | LDTRG1   |
| 1                          | LDTRG2   |

|                             |                             |   |
|-----------------------------|-----------------------------|---|
| ADnTRGSEL21 <sup>Note</sup> | ADnTRGSEL20 <sup>Note</sup> | Specification of selection trigger 2 for ADnECR3, ADnECR4 registers |
| 0                           | 0                           | ITRG1   |
| 0                           | 1                           | ITRG2   |
| 1                           | 0                           | ITRG3   |
| 1                           | 1                           | ITRG4   |

|                            |  |
|----------------------------|--|
| ADnLDTSEL1 <sup>Note</sup> | Specification of selection load trigger 1 for ADnECR0 to ADnECR2 registers |
| 0                          | LDTRG1   |
| 1                          | LDTRG2   |

|             |             |   |
|-------------|-------------|---|
| ADnTRGSEL11 | ADnTRGSEL10 | <ul style="list-style-type: none"> <li>In hardware trigger mode or conversion channel specification mode:<br/>Trigger specification</li> <li>In expansion buffer mode:<br/>Specification of selection trigger 1 for ADnECR0 to ADnECR2 registers</li> </ul> |
| 0           | 0           | ITRG1   |
| 0           | 1           | ITRG2   |
| 1           | 0           | ITRG3   |
| 1           | 1           | ITRG4   |

**Note** Be sure to set bits 3, 5, and 7 to “0” and set bit 4 to “1” in the hardware trigger mode and conversion channel specification mode.

**Caution** Set the ADnTSEL register when the ADnSCM.ADnCE bit = 0 (conversion operation is stopped) (the same value can be written to the register when the ADnCE bit = 1 (conversion operation is enabled)).

**(7) A/D converter n channel specification register 1 (ADnCH1)**

The ADnCH1 register is a register that specifies the analog input pin for selection trigger 1 in the conversion channel specification mode and extension buffer 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: AD0CH1 FFFFF232H, AD1CH1 FFFFF2B2H

|                      |   |                |                |                |   |                |                |                |
|----------------------|---|----------------|----------------|----------------|---|----------------|----------------|----------------|
|                      | 7 | 6              | 5              | 4              | 3 | 2              | 1              | 0              |
| ADnCH1<br>(n = 0, 1) | 0 | ADn<br>TRGCH16 | ADn<br>TRGCH15 | ADn<br>TRGCH14 | 0 | ADn<br>TRGCH12 | ADn<br>TRGCH11 | ADn<br>TRGCH10 |

| ADnTRGCH16 | ADnTRGCH15 | ADnTRGCH14 | Specification of analog input pin for selection trigger 1 |
|------------|------------|------------|---|
| 0          | 0          | 0          | ANIn0   |
| 0          | 0          | 1          | ANIn1   |
| 0          | 1          | 0          | ANIn2   |
| 0          | 1          | 1          | ANIn3   |
| 1          | 0          | 0          | ANIn4   |
| 1          | 0          | 1          | ANIn5   |
| 1          | 1          | 0          | ANI16   |
| 1          | 1          | 1          | ANI17   |

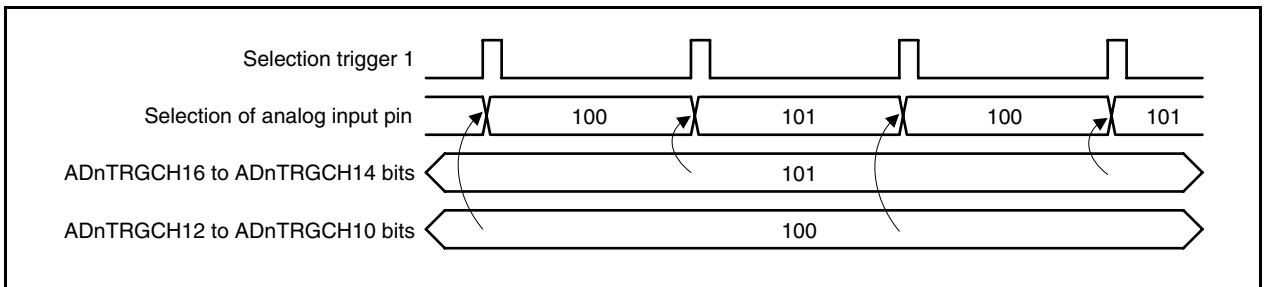
| ADnTRGCH12 | ADnTRGCH11 | ADnTRGCH10 | Specification of analog input pin for selection trigger 1 |
|------------|------------|------------|---|
| 0          | 0          | 0          | ANIn0   |
| 0          | 0          | 1          | ANIn1   |
| 0          | 1          | 0          | ANIn2   |
| 0          | 1          | 1          | ANIn3   |
| 1          | 0          | 0          | ANIn4   |
| 1          | 0          | 1          | ANIn5   |
| 1          | 1          | 0          | ANI16   |
| 1          | 1          | 1          | ANI17   |

**Cautions 1. Set the ADnCH1 register when the ADnSCM.ADnCE bit = 0 (conversion operation is stopped) (the same value can be written to the register when the ADnCE bit = 1 (conversion operation is enabled)).**

**2. Be sure to set bits 3 and 7 to "0".**

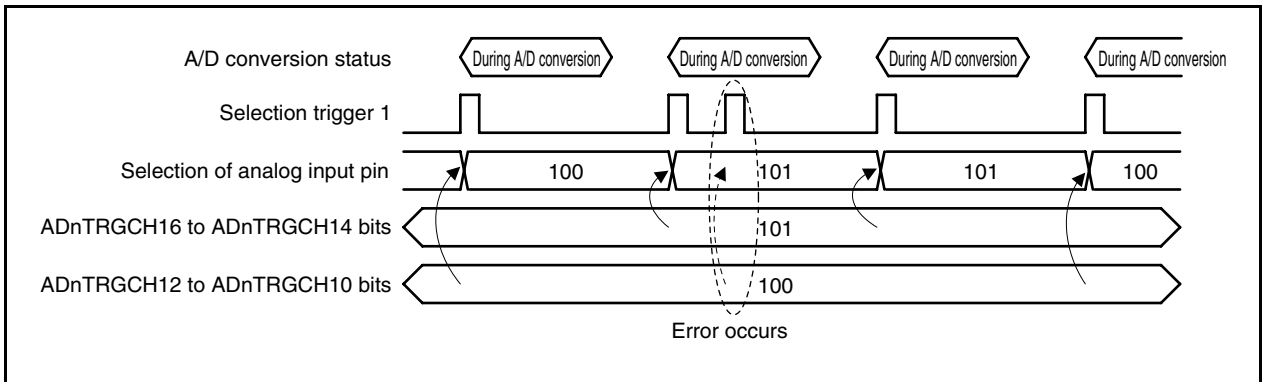
Setting the ADnCH1 register is enabled when a conversion operation is enabled (ADnSCM.ADnCE bit = 1) in the conversion channel specification mode or extension buffer mode. When the first selection trigger 1 is generated after the conversion operation is enabled (ADnCE bit = 1), the analog input pin specified by the ADnTRGCH12 to ADnTRGCH10 bits is selected and A/D conversion is executed. When the next selection trigger 1 is later generated, the analog input pin specified by the ADnTRGCH16 to ADnTRGCH14 bits is selected and A/D conversion is executed. After that, the analog input pins are alternately selected for output each time selection trigger 1 is generated.

Figure 12-7. ADnCH1 Register Operation



If an error occurs (when selection trigger 1 is generated during A/D conversion), the analog input pin specified by the ADnTRGCH12 to ADnTRGCH10 bits and the analog input pin specified by the ADnTRGCH16 to ADnTRGCH14 bits are alternately selected, but the selected analog input pin is not changed because A/D conversion is in progress.

Figure 12-8. ADnCH1 Register Operation In Case of Error



**(8) A/D converter n channel specification register 2 (ADnCH2)**

The ADnCH2 register is a register that specifies the analog input pin for selection trigger 2 in the extension buffer 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: AD0CH2 FFFFF233H, AD1CH2 FFFFF2B3H

|                      |   |                |                |                |   |                |                |                |
|----------------------|---|----------------|----------------|----------------|---|----------------|----------------|----------------|
|                      | 7 | 6              | 5              | 4              | 3 | 2              | 1              | 0              |
| ADnCH2<br>(n = 0, 1) | 0 | ADn<br>TRGCH26 | ADn<br>TRGCH25 | ADn<br>TRGCH24 | 0 | ADn<br>TRGCH22 | ADn<br>TRGCH21 | ADn<br>TRGCH20 |

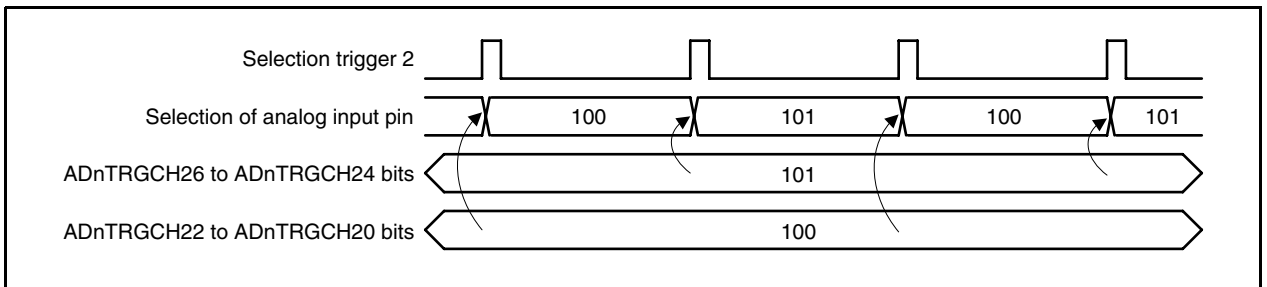
| ADnTRGCH26 | ADnTRGCH25 | ADnTRGCH24 | Specification of analog input pin for selection trigger 2 |
|------------|------------|------------|---|
| 0          | 0          | 0          | ANIn0   |
| 0          | 0          | 1          | ANIn1   |
| 0          | 1          | 0          | ANIn2   |
| 0          | 1          | 1          | ANIn3   |
| 1          | 0          | 0          | ANIn4   |
| 1          | 0          | 1          | ANIn5   |
| 1          | 1          | 0          | ANI16   |
| 1          | 1          | 1          | ANI17   |

| ADnTRGCH22 | ADnTRGCH21 | ADnTRGCH20 | Specification of analog input pin for selection trigger 2 |
|------------|------------|------------|---|
| 0          | 0          | 0          | ANIn0   |
| 0          | 0          | 1          | ANIn1   |
| 0          | 1          | 0          | ANIn2   |
| 0          | 1          | 1          | ANIn3   |
| 1          | 0          | 0          | ANIn4   |
| 1          | 0          | 1          | ANIn5   |
| 1          | 1          | 0          | ANI16   |
| 1          | 1          | 1          | ANI17   |

- Cautions**
1. Set the ADnCH2 register when the ADnSCM.ADnCE bit = 0 (conversion operation is stopped) (the same value can be written to the register when the ADnCE bit = 1 (conversion operation is enabled)).
  2. The ADnCH2 register is valid only in the extension buffer mode; it is invalid in any other mode.
  3. Be sure to set bits 3 and 7 to "0".

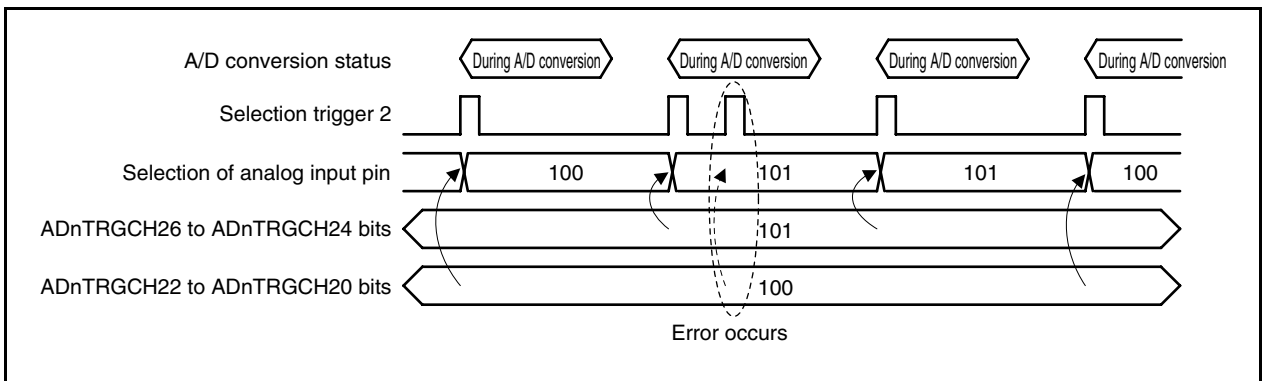
Setting the ADnCH2 register is enabled when a conversion operation is enabled (ADnSCM.ADnCE bit = 1) in the extension buffer mode. When the first selection trigger 2 is generated after the conversion operation is enabled (ADnCE bit = 1), the analog input pin specified by the ADnTRGCH22 to ADnTRGCH20 bits is selected and A/D conversion is executed. When the next selection trigger 2 is later generated, the analog input pin specified by the ADnTRGCH26 to ADnTRGCH24 bits is selected and A/D conversion is executed. After that, the analog input pins are alternately selected for output each time selection trigger 2 is generated.

Figure 12-9. ADnCH2 Register Operation



If an error occurs (when selection trigger 2 is generated during A/D conversion), the analog input pin specified by the ADnTRGCH22 to ADnTRGCH20 bits and the analog input pin specified by the ADnTRGCH26 to ADnTRGCH24 bits are alternately selected, but the selected analog input pin is not changed because A/D conversion is in progress.

Figure 12-10. ADnCH2 Register Operation In Case of Error



**(9) A/Dn conversion result extension registers 0 to 4, 0H to 4H (ADnECR0 to ADnECR4, ADnECR0H to ADnECR4H)**

The ADnECRa and ADnECRaH registers hold the result of A/D conversion in their higher 12 bits and indicate the status (information on the A/D conversion result of the analog input pin specified by the ADnCHx.ADnTRGCHx2 to ADnTRGCHx0 bits or ADnTRGCHx6 to ADnTRGCHx4 bits) of the A/D conversion result with the lower 1 bit in the extension buffer mode. Five of these registers are provided per circuit and two circuits are available. When A/D conversion is completed, the A/D conversion result is stored in A/Dn conversion result extension buffer register a. When selection load trigger 1 is later generated, the A/D conversion result is shifted from A/Dn conversion result extension buffer registers 0 to 2 to the higher 12 bits of the ADnECR0 to ADnECR2 registers and stored. Bits 1 to 3 are always 0 when read. When selection load trigger 2 is generated, the A/D conversion result is shifted from the A/Dn conversion result extension buffer registers 3 and 4 to the higher 12 bits of the ADnECR3 and ADnECR4 registers and stored. Bits 1 to 3 are always 0 when read.

The higher 8 bits of the A/D conversion result are read from the ADnECRaH register.

These registers are read-only in 16-bit or 8-bit units. To read the A/D conversion result in 16-bit units, specify the ADnECRa register. Specify the ADnECRaH register to read the higher 8 bits of the A/D conversion result. Reset sets these registers to 0000H.

**Remark** While the result of A/D conversion is stored in the ADnECRa register, a read access to that register is held pending. The pending read access is executed when storing the A/D conversion result is completed. Similarly, storing the A/D conversion result in the ADnECRa register is held pending while a read access is made to that register. The pending A/D conversion result is stored in the register after the read access is completed.



After reset: 0000H R Address: AD0ECR0 FFFFF240H, AD0ECR1 FFFFF242H,  
 AD0ECR2 FFFFF244H, AD0ECR3 FFFFF246H,  
 AD0ECR4 FFFFF248H,  
 AD1ECR0 FFFFF2C0H, AD1ECR1 FFFFF2C2H,  
 AD1ECR2 FFFFF2C4H, AD1ECR3 FFFFF2C6H,  
 AD1ECR4 FFFFF2C8H

|                                       |      |      |      |      |      |      |      |      |      |      |      |      |   |   |   |      |
|---------------------------------------|------|------|------|------|------|------|------|------|------|------|------|------|---|---|---|------|
|                                       | 15   | 14   | 13   | 12   | 11   | 10   | 9    | 8    | 7    | 6    | 5    | 4    | 3 | 2 | 1 | 0    |
| ADnECRa<br>(n = 0, 1)<br>(a = 0 to 4) | ADn  | ADn  | ADn  | ADn  | ADn  | ADn  | ADn  | ADn  | ADn  | ADn  | ADn  | ADn  | 0 | 0 | 0 | ADn  |
|                                       | ECRa | ECRa | ECRa | ECRa | ECRa | ECRa | ECRa | ECRa | ECRa | ECRa | ECRa | ECRa |   |   |   | CH   |
|                                       | 11   | 10   | 9    | 8    | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |   |   |   | FLGa |

|           |   |
|-----------|---|
| ADnCHFLGa | Status of A/D conversion result (x = 1, 2)  |
| 0         | A/D conversion result for analog input pin set by ADnCHx.ADnTRGCHx2 to ADnCHx.ADnTRGCHx0 bits |
| 1         | A/D conversion result for analog input pin set by ADnCHx.ADnTRGCHx6 to ADnCHx.ADnTRGCHx4 bits |

After reset: 00H R Address: AD0ECR0H FFFFF241H, AD0ECR1H FFFFF243H,  
 AD0ECR2H FFFFF245H, AD0ECR3H FFFFF247H,  
 AD0ECR4H FFFFF249H,  
 AD1ECR0H FFFFF2C1H, AD1ECR1H FFFFF2C3H,  
 AD1ECR2H FFFFF2C5H, AD1ECR3H FFFFF2C7H,  
 AD1ECR4H FFFFF2C9H

|  |          |           |          |          |          |          |          |          |
|--|----------|-----------|----------|----------|----------|----------|----------|----------|
|  | 7        | 6         | 5        | 4        | 3        | 2        | 1        | 0        |
| ADnECRaH<br>(n = 0, 1)<br>(a = 0 to 4) | ADnECRa1 | ADnECRa10 | ADnECRa9 | ADnECRa8 | ADnECRa7 | ADnECRa6 | ADnECRa5 | ADnECRa4 |
|  |          |           |          |          |          |          |          |          |

**Caution** The ADnECRa and ADnECRaH registers are valid only in the extension buffer mode; they are invalid in any other mode.

The correspondence between the analog input pins and the A/Dn conversion result extension registers is shown below.

**Table 12-7. Correspondence Between Analog Input Pins and A/D Conversion Result Extension Registers**

| Analog Input Pin   | A/Dn Conversion Result Register |
|--|---------------------------------|
| Set with ADnCH1 register's ADnTRGCH12 to ADnTRGCH10, ADnTRGCH16 to ADnTRGCH14 bits | ADnECR0, ADnECR0H               |
|  | ADnECR1, ADnECR1H               |
|  | ADnECR2, ADnECR2H               |
| Set with ADnCH2 register's ADnTRGCH22 to ADnTRGCH20, ADnTRGCH26 to ADnTRGCH24 bits | ADnECR3, ADnECR3H               |
|  | ADnECR4, ADnECR4H               |

**Remark** n = 0, 1

**(10) A/D converter n flag register (ADnFLG)**

The ADnFLG register indicates that an error has occurred when selection load trigger x is generated in the extension buffer mode (x = 1 or 2). The ADnTERR2 and ADnTERR1 flags can only be read and cleared when the conversion operation is stopped (ADnSCM.ADnCE bit = 0).

This register is read-only in 8-bit units.

Reset sets this register to 00H.

After reset: 00H R Address: AD0FLG FFFFF254H, AD1FLG FFFFF2D4H

|                      |   |   |   |   |   |   |                              |                              |
|----------------------|---|---|---|---|---|---|------------------------------|------------------------------|
|                      | 7 | 6 | 5 | 4 | 3 | 2 | 1                            | 0                            |
| ADnFLG<br>(n = 0, 1) | 0 | 0 | 0 | 0 | 0 | 0 | ADn<br>TERR2 <sup>Note</sup> | ADn<br>TERR1 <sup>Note</sup> |

|                          |  |
|--------------------------|--|
| ADnTERR2 <sup>Note</sup> | Occurrence timing error flag of selection load trigger 2             |
| 0                        | Occurrence timing error of selection load trigger 2 has not occurred |
| 1                        | Occurrence timing error of selection load trigger 2 has occurred     |

|                          |  |
|--------------------------|--|
| ADnTERR1 <sup>Note</sup> | Occurrence timing error flag of selection load trigger 1             |
| 0                        | Occurrence timing error of selection load trigger 1 has not occurred |
| 1                        | Occurrence timing error of selection load trigger 1 has occurred     |

**Note** The ADnTERR2 and ADnTERR1 flags are valid only in the extension buffer mode; they are fixed to 0 in any other mode.

**(11) A/D converter n flag buffer register (ADnFLGB)**

The ADnFLGB register indicates that an error has occurred when selection trigger x is generated in the extension buffer mode (x = 1 or 2). The ADnTERRB2 and ADnTERRB1 flags can only be read and cleared when the conversion operation is stopped (ADnSCM.ADnCE bit = 0).

This register is read-only in 8-bit units.

Reset sets this register to 00H.

After reset: 00H R Address: AD0FLGB FFFFF255H, AD1FLGB FFFFF2D5H

|                       |   |   |   |   |   |   |                               |                               |
|-----------------------|---|---|---|---|---|---|-------------------------------|-------------------------------|
|                       | 7 | 6 | 5 | 4 | 3 | 2 | 1                             | 0                             |
| ADnFLGB<br>(n = 0, 1) | 0 | 0 | 0 | 0 | 0 | 0 | ADn<br>TERRB2 <sup>Note</sup> | ADn<br>TERRB1 <sup>Note</sup> |

|                           |   |
|---------------------------|---|
| ADnTERRB2 <sup>Note</sup> | Occurrence timing error flag of selection trigger 2             |
| 0                         | Occurrence timing error of selection trigger 2 has not occurred |
| 1                         | Occurrence timing error of selection trigger 2 has occurred     |

|                           |   |
|---------------------------|---|
| ADnTERRB1 <sup>Note</sup> | Occurrence timing error flag of selection trigger 1             |
| 0                         | Occurrence timing error of selection trigger 1 has not occurred |
| 1                         | Occurrence timing error of selection trigger 1 has occurred     |

**Note** The ADnTERRB2 and ADnTERRB1 flags are valid only in the extension buffer mode; they are fixed to 0 in any other mode.

**(12) A/D LDTRG1 input select register (ADLTS1)**

The ADLTS1 register is a register that specifies the input signal for selection load trigger (LDTRG1) in the extension buffer mode.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF2F8H

|        |   |   |   |   |   |   |   |   |         |
|--------|---|---|---|---|---|---|---|---|---------|
|        | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |         |
| ADLTS1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ADLTS10 |

|         |  |
|---------|--|
| ADLTS10 | Specification of input signal for LDTRG1 |
| 0       | TABTIOV0 signal                          |
| 1       | TABTIOV1 signal                          |

**Note** The ADLTS1 register is valid only in the extension buffer mode; it is invalid in any other mode.

**(13) A/D LDTRG2 input select register (ADLTS2)**

The ADLTS2 register is a register that specifies the input signal for selection load trigger (LDTRG2) in the extension buffer mode.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF2FAH

|        |   |   |   |   |   |   |   |   |         |
|--------|---|---|---|---|---|---|---|---|---------|
|        | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |         |
| ADLTS2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ADLTS20 |

|         |  |
|---------|--|
| ADLTS20 | Specification of input signal for LDTRG2 |
| 0       | TABTICC00 signal                         |
| 1       | TABTICC10 signal                         |

**Note** The ADLTS2 register is valid only in the extension buffer mode; it is invalid in any other mode.

**(14) A/D converter n clock select register (ADnOCKS)**

The ADnOCKS register is a register that selects the clock ( $f_{AD01}$ ) to be input to the A/D converter n. This register can be read or written in 8-bit units. Reset sets this register to 00H.

After reset: 00H R/W Address: AD0OCKS FFFFF270H, AD1OCKS FFFFF274H

|                       |   |   |   |           |   |   |          |          |
|-----------------------|---|---|---|-----------|---|---|----------|----------|
|                       | 7 | 6 | 5 | 4         | 3 | 2 | 1        | 0        |
| ADnOCKS<br>(n = 0, 1) | 0 | 0 | 0 | ADnOCKSEN | 0 | 0 | ADnOCKS1 | ADnOCKS0 |

|           |  |
|-----------|--|
| ADnOCKSEN | Clock operation control                          |
| 0         | Stop operation clock supply of A/D converter n   |
| 1         | Enable operation clock supply of A/D converter n |

|          |          |   |
|----------|----------|---|
| ADnOCKS1 | ADnOCKS0 | Input clock selection of A/D converter n ( $f_{AD01}$ ) |
| 0        | 0        | $f_{xx}/2$  |
| 0        | 1        | $f_{xx}/3$  |
| 1        | 0        | $f_{xx}/4$ (when $f_{xx} = 64$ MHz)                     |
| 1        | 1        | Setting prohibited                                      |

- Cautions**
1. Set  $f_{AD01}$  to 16 MHz or lower.
  2. When A/D converter n is used, be sure to set the ADnOCKS register and set the ADnSCM.ADnPS bit to 1, as well as to read the A/D conversion result register.
  3. Be sure to set bits 2, 3, and 5 to 7 to "0".

**(15) A/D trigger rising edge, falling edge specification registers (ADTR, ADTF)**

The ADTR and ADTF registers are registers that specify the trigger mode of the ADTRG0/INTADT0 and ADTRG1/INTADT1 pins and can specify the valid edge independently for each pin (rising edge, falling edge, or both rising and falling edges).

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 trigger input of the A/D converter n (alternate function)/external interrupt function (alternate function) to the port mode, an edge may be detected. Therefore, be sure to set the ADTFn and ADTRn bits to 00, and then set the port mode.

After reset: 00H R/W Address: FFFFF2F2H

|      |   |   |   |   |   |   |       |       |
|------|---|---|---|---|---|---|-------|-------|
|      | 7 | 6 | 5 | 4 | 3 | 2 | 1     | 0     |
| ADTR | 0 | 0 | 0 | 0 | 0 | 0 | ADTR1 | ADTR0 |

After reset: 00H R/W Address: FFFFF2F0H

|      |   |   |   |   |   |   |       |       |
|------|---|---|---|---|---|---|-------|-------|
|      | 7 | 6 | 5 | 4 | 3 | 2 | 1     | 0     |
| ADTF | 0 | 0 | 0 | 0 | 0 | 0 | ADTF1 | ADTF0 |

**Remark** For the valid edge specification, see Table 12-8.

**Table 12-8. Valid Edge Specification of ADTRG0/INTADT0 and ADTRG1/INTADT1 Pins**

| ADTFn | ADTRn | Valid Edge Specification      |
|-------|-------|-------------------------------|
| 0     | 0     | No edge detected              |
| 0     | 1     | Rising edge                   |
| 1     | 0     | Falling edge                  |
| 1     | 1     | Both rising and falling edges |

**Caution** When not using these pins as the ADTRGn/INTADTn pins, be sure to set the ADTFn and ADTRn bits to 00.

**Remark** n = 0, 1

**(16) Operational amplifier n control register 0 (OPnCTL0)**

The OPnCTL0 register is used to control the operation of an operational amplifier that amplifies the input level, and specify its gain.

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

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: FFFFF260H

|         |   |   |   |       |        |        |        |        |
|---------|---|---|---|-------|--------|--------|--------|--------|
|         | 7 | 6 | 5 | 4     | 3      | 2      | 1      | 0      |
| OP0CTL0 | 0 | 0 | 0 | OP0EN | OP0GA3 | OP0GA2 | OP0GA1 | OP0GA0 |

|       |  |
|-------|--|
| OP0EN | Operation control of operational amplifier 0 for A/D converter 0 |
| 0     | Operation disabled (not used)                                    |
| 1     | Operation enabled (used)   |

| OP0GA3 | OP0GA2 | OP0GA1 | OP0GA0 | Gain specification of operational amplifier |
|--------|--------|--------|--------|---|
| 0      | 0      | 0      | 0      | ×2.500                                      |
| 0      | 0      | 0      | 1      | ×2.667                                      |
| 0      | 0      | 1      | 0      | ×2.857                                      |
| 0      | 0      | 1      | 1      | ×3.077                                      |
| 0      | 1      | 0      | 0      | ×3.333                                      |
| 0      | 1      | 0      | 1      | ×3.636                                      |
| 0      | 1      | 1      | 0      | ×4.000                                      |
| 0      | 1      | 1      | 1      | ×4.444                                      |
| 1      | 0      | 0      | 0      | ×5.000                                      |
| 1      | 0      | 0      | 1      | ×5.714                                      |
| 1      | 0      | 1      | 0      | ×6.667                                      |
| 1      | 0      | 1      | 1      | ×8.000                                      |
| 1      | 1      | 0      | 0      | ×10.00                                      |
| Others |        |        |        | Setting prohibited                          |

- Cautions**
1. Be sure to set bits 5 to 7 of the OP0CTL0 register to “0”.
  2. After enabling the operation of the operational amplifier, stabilization time of 10  $\mu$ s is required.  
If the OP0GA3 to OP0GA0 bit settings are changed, stabilization time of 5  $\mu$ s is required.

<R>

After reset: 00H R/W Address: FFFFF2E0H

|         |   |        |        |        |        |        |        |        |
|---------|---|--------|--------|--------|--------|--------|--------|--------|
|         | 7 | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
| OP1CTL0 | 0 | OP12EN | OP11EN | OP10EN | OP1GA3 | OP1GA2 | OP1GA1 | OP1GA0 |

|        |  |
|--------|--|
| OP12EN | Operation control of operational amplifier 2 for A/D converter 1 |
| 0      | Operation disabled (not used)                                    |
| 1      | Operation enabled (used)   |

|        |  |
|--------|--|
| OP11EN | Operation control of operational amplifier 1 for A/D converter 1 |
| 0      | Operation disabled (not used)                                    |
| 1      | Operation enabled (used)   |

|        |  |
|--------|--|
| OP10EN | Operation control of operational amplifier 0 for A/D converter 1 |
| 0      | Operation disabled (not used)                                    |
| 1      | Operation enabled (used)   |

| OP1GA3 | OP1GA2 | OP1GA1 | OP1GA0 | Gain specification of operational amplifier |
|--------|--------|--------|--------|---|
| 0      | 0      | 0      | 0      | ×2.500                                      |
| 0      | 0      | 0      | 1      | ×2.667                                      |
| 0      | 0      | 1      | 0      | ×2.857                                      |
| 0      | 0      | 1      | 1      | ×3.077                                      |
| 0      | 1      | 0      | 0      | ×3.333                                      |
| 0      | 1      | 0      | 1      | ×3.636                                      |
| 0      | 1      | 1      | 0      | ×4.000                                      |
| 0      | 1      | 1      | 1      | ×4.444                                      |
| 1      | 0      | 0      | 0      | ×5.000                                      |
| 1      | 0      | 0      | 1      | ×5.714                                      |
| 1      | 0      | 1      | 0      | ×6.667                                      |
| 1      | 0      | 1      | 1      | ×8.000                                      |
| 1      | 1      | 0      | 0      | ×10.00                                      |
| Others |        |        |        | Setting prohibited                          |

- Cautions**
1. Be sure to set bit 7 of the OP1CTL0 register to "0".
  2. After enabling the operation of the operational amplifier, stabilization time of 10  $\mu\text{s}$  is required.  
If the OP1GA3 to OP1GA0 bit settings are changed, stabilization time of 5  $\mu\text{s}$  is required.

&lt;R&gt;



**(17) Comparator n control register 0 (CMPnCTL0)**

The CMPnCTL0 register is a register that controls the operation of the overvoltage detection comparator. This register can be read or written in 8-bit units. Reset sets this register to 00H.

(1/2)

|  |  |          |          |          |   |          |          |          |
|--|--|----------|----------|----------|---|----------|----------|----------|
| After reset: 00H R/W Address: FFFF261H |  |          |          |          |   |          |          |          |
| CMP0CTL0                               | 7  | 6        | 5        | 4        | 3 | 2        | 1        | 0        |
|  | 0  | 0        | 0        | CMP0FEN  | 0 | 0        | 0        | CMP0LEN  |
| CMP0FEN                                | Operation control of comparator 0 (full range) for A/D converter 0 |          |          |          |   |          |          |          |
| 0                                      | Operation disabled (not used)                                      |          |          |          |   |          |          |          |
| 1                                      | Operation enabled (used)   |          |          |          |   |          |          |          |
| CMP0LEN                                | Operation control of comparator 0 (low range) for A/D converter 0  |          |          |          |   |          |          |          |
| 0                                      | Operation disabled (not used)                                      |          |          |          |   |          |          |          |
| 1                                      | Operation enabled (used)   |          |          |          |   |          |          |          |
| After reset: 00H R/W Address: FFFF2E1H |  |          |          |          |   |          |          |          |
| CMP1CTL0                               | 7  | 6        | 5        | 4        | 3 | 2        | 1        | 0        |
|  | 0  | CMP12FEN | CMP11FEN | CMP10FEN | 0 | CMP12LEN | CMP11LEN | CMP10LEN |
| CMP12FEN                               | Operation control of comparator 2 (full range) for A/D converter 1 |          |          |          |   |          |          |          |
| 0                                      | Operation disabled (not used)                                      |          |          |          |   |          |          |          |
| 1                                      | Operation enabled (used)   |          |          |          |   |          |          |          |
| CMP11FEN                               | Operation control of comparator 1 (full range) for A/D converter 1 |          |          |          |   |          |          |          |
| 0                                      | Operation disabled (not used)                                      |          |          |          |   |          |          |          |
| 1                                      | Operation enabled (used)   |          |          |          |   |          |          |          |
| CMP10FEN                               | Operation control of comparator 0 (full range) for A/D converter 1 |          |          |          |   |          |          |          |
| 0                                      | Operation disabled (not used)                                      |          |          |          |   |          |          |          |
| 1                                      | Operation enabled (used)   |          |          |          |   |          |          |          |
| CMP12LEN                               | Operation control of comparator 2 (low range) for A/D converter 1  |          |          |          |   |          |          |          |
| 0                                      | Operation disabled (not used)                                      |          |          |          |   |          |          |          |
| 1                                      | Operation enabled (used)   |          |          |          |   |          |          |          |
| CMP11LEN                               | Operation control of comparator 1 (low range) for A/D converter 1  |          |          |          |   |          |          |          |
| 0                                      | Operation disabled (not used)                                      |          |          |          |   |          |          |          |
| 1                                      | Operation enabled (used)   |          |          |          |   |          |          |          |
| CMP10LEN                               | Operation control of comparator 0 (low range) for A/D converter 1  |          |          |          |   |          |          |          |
| 0                                      | Operation disabled (not used)                                      |          |          |          |   |          |          |          |
| 1                                      | Operation enabled (used)   |          |          |          |   |          |          |          |

**Cautions** 1. Be sure to set bits 1 to 3, 5 to 7 of the CMP0CTL0 register and bits 3, 7 of the CMP1CTL0 register to “0”.

2. After enabling the operation of the comparator, stabilization time of 10  $\mu$ s is required.

3. The input voltage range of the comparator is as follows, regardless of whether it is amplified by the operational amplifier or not.

Input voltage range of CREF0L and CREF1L pins (low-range side):  $0.02AV_{DD} + 0.1$  to  $0.5AV_{DD} - 0.1$  V

Input voltage range of CREF0F and CREF1F pins (full-range side):  $0.02AV_{DD} + 0.1$  to  $0.92AV_{DD} - 0.1$  V

For details, see CHAPTER 28 ELECTRICAL SPECIFICATIONS.

&lt;R&gt;

**(18) Comparator n control register 1 (CMPnCTL1)**

The CMPnCTL1 register is a register that monitors the output of the overvoltage detection comparator. This register is read-only in 8-bit units. Reset sets this register to 00H.

(1/2)

After reset: 00H R Address: FFFFF262H

|          |   |   |   |          |   |   |   |          |
|----------|---|---|---|----------|---|---|---|----------|
|          | 7 | 6 | 5 | 4        | 3 | 2 | 1 | 0        |
| CMP0CTL1 | 0 | 0 | 0 | CMP0FOUT | 0 | 0 | 0 | CMP0LOUT |

|          |  |
|----------|--|
| CMP0FOUT | Output level status of comparator 0 (full range) for A/D converter 0 |
| 0        | Comparator output = 0 (without overvoltage detection)                |
| 1        | Comparator output = 1 (with overvoltage detection)                   |

|          |   |
|----------|---|
| CMP0LOUT | Output level status of comparator 0 (low range) for A/D converter 0 |
| 0        | Comparator output = 0 (without overvoltage detection)               |
| 1        | Comparator output = 1 (with overvoltage detection)                  |

After reset: 00H R Address: FFFFF2E2H

|          |   |           |           |           |   |           |           |           |
|----------|---|-----------|-----------|-----------|---|-----------|-----------|-----------|
|          | 7 | 6         | 5         | 4         | 3 | 2         | 1         | 0         |
| CMP1CTL1 | 0 | CMP12FOUT | CMP11FOUT | CMP10FOUT | 0 | CMP12LOUT | CMP11LOUT | CMP10LOUT |

|           |  |
|-----------|--|
| CMP12FOUT | Output level status of comparator 2 (full range) for A/D converter 1 |
| 0         | Comparator output = 0 (without overvoltage detection)                |
| 1         | Comparator output = 1 (with overvoltage detection)                   |

|           |  |
|-----------|--|
| CMP11FOUT | Output level status of comparator 1 (full range) for A/D converter 1 |
| 0         | Comparator output = 0 (without overvoltage detection)                |
| 1         | Comparator output = 1 (with overvoltage detection)                   |

|           |  |
|-----------|--|
| CMP10FOUT | Output level status of comparator 0 (full range) for A/D converter 1 |
| 0         | Comparator output = 0 (without overvoltage detection)                |
| 1         | Comparator output = 1 (with overvoltage detection)                   |

|           |   |
|-----------|---|
| CMP12LOUT | Output level status of comparator 2 (low range) for A/D converter 1 |
| 0         | Comparator output = 0 (without overvoltage detection)               |
| 1         | Comparator output = 1 (with overvoltage detection)                  |

|           |   |
|-----------|---|
| CMP11LOUT | Output level status of comparator 1 (low range) for A/D converter 1 |
| 0         | Comparator output = 0 (without overvoltage detection)               |
| 1         | Comparator output = 1 (with overvoltage detection)                  |

|           |   |
|-----------|---|
| CMP10LOUT | Output level status of comparator 0 (low range) for A/D converter 1 |
| 0         | Comparator output = 0 (without overvoltage detection)               |
| 1         | Comparator output = 1 (with overvoltage detection)                  |

- Cautions**
1. The **CMP12FOUT**, **CMP11FOUT**, **CMP0FOUT**, **CMP10FOUT**, **CMP12LOUT**, **CMP11LOUT**, **CMP0LOUT**, and **CMP10LOUT** bits are set to 0 when the input voltage falls to a level at which an overvoltage is not detected.
  2. Be sure to set bits 1 to 3, 5 to 7 of the **CMP0CTL1** register and bits 3, 7 of the **CMP1CTL1** register to “0”.

**(19) Comparator n control register 2 (CMPnCTL2)**

The CMPnCTL2 register is a register that specifies the compare signal of the overvoltage detection comparator. This register can be read or written in 8-bit units. Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF263H

|          |   |   |   |   |   |   |   |         |
|----------|---|---|---|---|---|---|---|---------|
|          | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0       |
| CMP0CTL2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | CMP0SEL |

|         |   |
|---------|---|
| CMP0SEL | Specification of compare signal of comparator 0 for A/D converter 0 |
| 0       | Before operational amplifier 0 amplification                        |
| 1       | After operational amplifier 0 amplification                         |

After reset: 00H R/W Address: FFFFF2E3H

|          |   |   |   |   |   |          |          |          |
|----------|---|---|---|---|---|----------|----------|----------|
|          | 7 | 6 | 5 | 4 | 3 | 2        | 1        | 0        |
| CMP1CTL2 | 0 | 0 | 0 | 0 | 0 | CMP12SEL | CMP11SEL | CMP10SEL |

|          |   |
|----------|---|
| CMP12SEL | Specification of compare signal of comparator 2 for A/D converter 1 |
| 0        | Before operational amplifier 2 amplification                        |
| 1        | After operational amplifier 2 amplification                         |

|          |   |
|----------|---|
| CMP11SEL | Specification of compare signal of comparator 1 for A/D converter 1 |
| 0        | Before operational amplifier 1 amplification                        |
| 1        | After operational amplifier 1 amplification                         |

|          |   |
|----------|---|
| CMP10SEL | Specification of compare signal of comparator 0 for A/D converter 1 |
| 0        | Before operational amplifier 0 amplification                        |
| 1        | After operational amplifier 0 amplification                         |

**Caution** Be sure to set bits 1 to 7 of the CMP0CTL2 register and bits 3 to 7 of the CMP1CTL2 register to “0”.

**(20) Comparator n control register 3 (CMPnCTL3)**

The CMPnCTL3 register is a register that specifies the detection direction of the overvoltage detection comparator and selects the edge detection.

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

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: FFFFF264H

|          |         |   |   |         |         |   |   |         |
|----------|---------|---|---|---------|---------|---|---|---------|
|          | 7       | 6 | 5 | 4       | 3       | 2 | 1 | 0       |
| CMP0CTL3 | CMP0FDS | 0 | 0 | CMP0FDE | CMP0LDS | 0 | 0 | CMP0LDE |

|         |   |
|---------|---|
| CMP0FDS | Specification of detection direction for comparator (full range) for A/D converter 0      |
| 0       | Logical product (AND) detection (Detects if input voltage is lower than reference value.) |
| 1       | Logical sum (OR) detection (Detects if input voltage is more than reference value.)       |

|         |   |
|---------|---|
| CMP0FDE | Selection of edge detection for comparator 0 (full range) for A/D converter 0 |
| 0       | Edge detection disabled (comparator not used)                                 |
| 1       | Edge detection enabled (comparator used)                                      |

|         |   |
|---------|---|
| CMP0LDS | Specification of detection direction for comparator (low range) for A/D converter 0       |
| 0       | Logical product (AND) detection (Detects if input voltage is lower than reference value.) |
| 1       | Logical sum (OR) detection (Detects if input voltage is more than reference value.)       |

|         |  |
|---------|--|
| CMP0LDE | Selection of edge detection for comparator 0 (low range) for A/D converter 0 |
| 0       | Edge detection disabled (comparator not used)                                |
| 1       | Edge detection enabled (comparator used)                                     |

**Remark** The reference value means the reference voltage input to the CREF0L and CREF0F pins.

After reset: 00H R/W Address: FFFFF2E4H

|          |         |          |          |          |         |          |          |          |
|----------|---------|----------|----------|----------|---------|----------|----------|----------|
|          | 7       | 6        | 5        | 4        | 3       | 2        | 1        | 0        |
| CMP1CTL3 | CMP1FDS | CMP12FDE | CMP11FDE | CMP10FDE | CMP1LDS | CMP12LDE | CMP11LDE | CMP10LDE |

|         |   |
|---------|---|
| CMP1FDS | Specification of detection direction for comparator (full range) for A/D converter 1      |
| 0       | Logical product (AND) detection (Detects if input voltage is lower than reference value.) |
| 1       | Logical sum (OR) detection (Detects if input voltage is more than reference value.)       |

|          |   |
|----------|---|
| CMP12FDE | Selection of edge detection for comparator 2 (full range) for A/D converter 1 |
| 0        | Edge detection disabled (comparator not used)                                 |
| 1        | Edge detection enabled (comparator used)                                      |

|          |   |
|----------|---|
| CMP11FDE | Selection of edge detection for comparator 1 (full range) for A/D converter 1 |
| 0        | Edge detection disabled (comparator not used)                                 |
| 1        | Edge detection enabled (comparator used)                                      |

|          |   |
|----------|---|
| CMP10FDE | Selection of edge detection for comparator 0 (full range) for A/D converter 1 |
| 0        | Edge detection disabled (comparator not used)                                 |
| 1        | Edge detection enabled (comparator used)                                      |

|         |   |
|---------|---|
| CMP1LDS | Specification of detection direction for comparator (low range) for A/D converter 1       |
| 0       | Logical product (AND) detection (Detects if input voltage is lower than reference value.) |
| 1       | Logical sum (OR) detection (Detects if input voltage is more than reference value.)       |

|          |  |
|----------|--|
| CMP12LDE | Selection of edge detection for comparator 2 (low range) for A/D converter 1 |
| 0        | Edge detection disabled (comparator not used)                                |
| 1        | Edge detection enabled (comparator used)                                     |

|          |  |
|----------|--|
| CMP11LDE | Selection of edge detection for comparator 1 (low range) for A/D converter 1 |
| 0        | Edge detection disabled (comparator not used)                                |
| 1        | Edge detection enabled (comparator used)                                     |

|          |  |
|----------|--|
| CMP10LDE | Selection of edge detection for comparator 0 (low range) for A/D converter 1 |
| 0        | Edge detection disabled (comparator not used)                                |
| 1        | Edge detection enabled (comparator used)                                     |

**Remark** The reference value means the reference voltage input to the CREF1L and CREF1F pins.

**(21) Comparator output digital noise elimination register nL, nF (CMPNFCnL, CMPNFCnF)**

The CMPNFCnL and CMPNFCnF registers are control the digital noise elimination of the overvoltage detection comparator output.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: CMPNFC0L FFFFF278H, CMPNFC1L FFFFF27CH

|                        |          |   |   |   |   |          |          |          |
|------------------------|----------|---|---|---|---|----------|----------|----------|
|                        | 7        | 6 | 5 | 4 | 3 | 2        | 1        | 0        |
| CMPNFCnL<br>(n = 0, 1) | CMPnNFEN | 0 | 0 | 0 | 0 | CMPnNFC2 | CMPnNFC1 | CMPnNFC0 |

After reset: 00H R/W Address: CMPNFC0F FFFFF27AH, CMPNFC1F FFFFF27EH

|                        |          |   |   |   |   |          |          |          |
|------------------------|----------|---|---|---|---|----------|----------|----------|
|                        | 7        | 6 | 5 | 4 | 3 | 2        | 1        | 0        |
| CMPNFCnF<br>(n = 0, 1) | CMPnNFEN | 0 | 0 | 0 | 0 | CMPnNFC2 | CMPnNFC1 | CMPnNFC0 |

|          |                                      |
|----------|--------------------------------------|
| CMPnNFEN | Setting of digital noise elimination |
| 0        | <b>Note</b>                          |
| 1        | Performs digital noise elimination   |

|          |          |          |                          |
|----------|----------|----------|--------------------------|
| CMPnNFC2 | CMPnNFC1 | CMPnNFC0 | Sampling clock selection |
| 0        | 0        | 0        | f <sub>xx</sub> /32      |
| 0        | 0        | 1        | f <sub>xx</sub> /64      |
| 0        | 1        | 0        | f <sub>xx</sub> /128     |
| 0        | 1        | 1        | f <sub>xx</sub> /256     |
| 1        | 0        | 0        | f <sub>xx</sub> /1024    |
| 1        | 0        | 1        | f <sub>xx</sub> /4096    |
| Others   |          |          | Setting prohibited       |

**Note** When using the comparator output, be sure to set the CMPnNFEN bit to 1 (digital noise eliminated). At the same time, a sampling clock must be selected.

When not using the comparator output, set the CMPnNFEN bit to 0.

**Caution** Be sure to set bits 3 to 6 to “0”.



**(22) Comparator output interrupt rising edge, falling edge specification registers (CMPOR, CMPOF)**

The CMPOR and CMPOF registers are registers that specify the trigger mode of the INTCMP0L, INTCMP0F, INTCMP1L, and INTCMP1F signals and can specify the valid edge independently for each interrupt request signal (rising edge, falling edge, or both rising and falling edges).

These registers can be read or written in 8-bit or 1-bit units.

Reset sets these registers to 00H.

|   |   |   |   |   |         |         |         |         |   |       |   |   |   |   |         |         |         |         |
|---|---|---|---|---|---------|---------|---------|---------|---|-------|---|---|---|---|---------|---------|---------|---------|
| After reset: 00H R/W Address: FFFF2F6H  |   |   |   |   |         |         |         |         |   |       |   |   |   |   |         |         |         |         |
| <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;"></td> <td style="width: 10%; text-align: center;">7</td> <td style="width: 10%; text-align: center;">6</td> <td style="width: 10%; text-align: center;">5</td> <td style="width: 10%; text-align: center;">4</td> <td style="width: 10%; text-align: center;">3</td> <td style="width: 10%; text-align: center;">2</td> <td style="width: 10%; text-align: center;">1</td> <td style="width: 10%; text-align: center;">0</td> </tr> <tr> <td style="border: none;">CMPOR</td> <td style="border: 1px solid black; text-align: center;">0</td> <td style="border: 1px solid black; text-align: center;">0</td> <td style="border: 1px solid black; text-align: center;">0</td> <td style="border: 1px solid black; text-align: center;">0</td> <td style="border: 1px solid black; text-align: center;">CMPOR1F</td> <td style="border: 1px solid black; text-align: center;">CMPOR1L</td> <td style="border: 1px solid black; text-align: center;">CMPOR0F</td> <td style="border: 1px solid black; text-align: center;">CMPOR0L</td> </tr> </table> |   | 7 | 6 | 5 | 4       | 3       | 2       | 1       | 0 | CMPOR | 0 | 0 | 0 | 0 | CMPOR1F | CMPOR1L | CMPOR0F | CMPOR0L |
|   | 7 | 6 | 5 | 4 | 3       | 2       | 1       | 0       |   |       |   |   |   |   |         |         |         |         |
| CMPOR   | 0 | 0 | 0 | 0 | CMPOR1F | CMPOR1L | CMPOR0F | CMPOR0L |   |       |   |   |   |   |         |         |         |         |
| After reset: 00H R/W Address: FFFF2F4H  |   |   |   |   |         |         |         |         |   |       |   |   |   |   |         |         |         |         |
| <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;"></td> <td style="width: 10%; text-align: center;">7</td> <td style="width: 10%; text-align: center;">6</td> <td style="width: 10%; text-align: center;">5</td> <td style="width: 10%; text-align: center;">4</td> <td style="width: 10%; text-align: center;">3</td> <td style="width: 10%; text-align: center;">2</td> <td style="width: 10%; text-align: center;">1</td> <td style="width: 10%; text-align: center;">0</td> </tr> <tr> <td style="border: none;">CMPOF</td> <td style="border: 1px solid black; text-align: center;">0</td> <td style="border: 1px solid black; text-align: center;">0</td> <td style="border: 1px solid black; text-align: center;">0</td> <td style="border: 1px solid black; text-align: center;">0</td> <td style="border: 1px solid black; text-align: center;">CMPOF1F</td> <td style="border: 1px solid black; text-align: center;">CMPOF1L</td> <td style="border: 1px solid black; text-align: center;">CMPOF0F</td> <td style="border: 1px solid black; text-align: center;">CMPOF0L</td> </tr> </table> |   | 7 | 6 | 5 | 4       | 3       | 2       | 1       | 0 | CMPOF | 0 | 0 | 0 | 0 | CMPOF1F | CMPOF1L | CMPOF0F | CMPOF0L |
|   | 7 | 6 | 5 | 4 | 3       | 2       | 1       | 0       |   |       |   |   |   |   |         |         |         |         |
| CMPOF   | 0 | 0 | 0 | 0 | CMPOF1F | CMPOF1L | CMPOF0F | CMPOF0L |   |       |   |   |   |   |         |         |         |         |
| <b>Remark</b> For the valid edge specification, see <b>Tables 12-9</b> and <b>12-10</b> .   |   |   |   |   |         |         |         |         |   |       |   |   |   |   |         |         |         |         |

**Table 12-9. Valid Edge Specification of INTCMP0F and INTCMP1F Signals**

| CMPOFnF | CMPORnF | Valid Edge Specification      |
|---------|---------|-------------------------------|
| 0       | 0       | No edge detected              |
| 0       | 1       | Rising edge                   |
| 1       | 0       | Falling edge                  |
| 1       | 1       | Both rising and falling edges |

**Remark** n = 0, 1

**Table 12-10. Valid Edge Specification of INTCMP0L and INTCMP1L Signals**

| CMPOFnL | CMPORnL | Valid Edge Specification      |
|---------|---------|-------------------------------|
| 0       | 0       | No edge detected              |
| 0       | 1       | Rising edge                   |
| 1       | 0       | Falling edge                  |
| 1       | 1       | Both rising and falling edges |

**Remark** n = 0, 1

## 12.4 Operation

**Cautions 1.** A/D converters 0 and 1 are capable of simultaneous sampling of two circuits.

**2.** For details of operation setting, see 12.3 (1) A/D converter n scan mode register (ADnSCM).

### 12.4.1 Basic operation

A/D conversion is executed by the following procedure.

- (1) Select an input clock ( $f_{AD01}$ ) by using the ADnOCKS register and set the ADnOCKSEN bit to 1 (enable supply of the operating clock to A/D converter n).
- (2) Set ADnSCM.ADnPS bit = 1.
- (3) Wait for 1  $\mu$ s or more after <2>.
- (4) Select an analog input pin and operation mode, by using the ADnSCM<sup>Note</sup>, ADnCTC, ADnCHEN, ADnCTL0, ADnTSEL, ADnCH1, ADnCH2, ADLTS1, and ADLTS2 registers ( $n = 0, 1$ ). Number of A/D conversion clocks and A/D conversion time are determined by the specification of the ADnCTC.ADnFR3 to ADnCTC.ADnFR0 bits.

**Note** Be sure to set bit 1 of the ADnSCM register to “1”.

This setting can be performed at the same time as other ADnSCM register bits.

- (5) In the A/D trigger mode and the A/D trigger polling mode, setting the ADnSCM.ADnCE bit to 1 starts A/D conversion ( $n = 0, 1$ ). If the ADnCE bit is set to 1 in the hardware trigger mode, conversion channel specification mode, and extension buffer mode, the A/D converter enters the trigger wait status.
- (6) When A/D conversion is started, the voltage input to the selected analog input channel is sampled by the sample & hold circuit. When the operational amplifier for input level amplification is used, the gain specified by the OPnCTL0.OPnGA3 to OPnCTL0.OPnGA0 bits  $\times$  the input voltage is sampled.
- (7) When sampling has been performed for a specific time, the sample & hold circuit enters the hold status, and holds the input analog voltage until A/D conversion ends.
- (8) Set bit 11 of the successive approximation register (SAR). The tap selector changes the level of the voltage tap of the array to the reference voltage ( $1/2AV_{REFn}$ ).
- (9) The voltage generated by the voltage tap of the array is compared with the analog input voltage by a comparator. If the analog input voltage is found to be greater than the reference voltage ( $1/2AV_{REFn}$ ) as a result of comparison, the most significant bit (MSB) of the successive approximation register (SAR) remains set. If the analog input voltage is less than the reference voltage ( $1/2AV_{REFn}$ ), the MSB of the SAR is reset.

(10) Next, bit 10 of the successive approximation register (SAR) is automatically set, and the next comparison is started. The voltage tap of the array is selected according to the value of bit 11, to which the result has been already set.

Bit 11 = 0:  $(1/4AV_{REFPn})$

Bit 11 = 1:  $(3/4AV_{REFPn})$

The voltage tap of the array and the analog input voltage are compared and bit 10 of the SAR is manipulated according to the result of the comparison.

Analog input voltage  $\geq$  Voltage tap of array: Bit 10 = 1

Analog input voltage  $\leq$  Voltage tap of array: Bit 10 = 0

Comparison is continued like this to bit 0 of the SAR.

(11) When comparison of 12 bits has been completed, the valid digital value result remains in the successive approximation register (SAR). This value is transferred to A/Dn conversion result register m (ADnCRm) and the conversion result is stored in this register in the A/D trigger mode, A/D trigger polling mode, hardware trigger mode, and conversion channel specification mode ( $n = 0, 1, m = 0$  to 15). The valid digital value is stored in the A/Dn conversion result extension buffer register a in the extension buffer mode, and is shifted to A/Dn conversion result extension register a when selection load trigger x is generated and stored ( $x = 1, 2, a = 0$  to 4). When A/D conversion has ended the specified number of times, an A/Dn conversion end interrupt request signal (INTADn) is generated.

### 12.4.2 Input voltage and conversion result

The relationship between the analog voltage input to the analog input pin (ANInk) and the A/D conversion result (of A/Dn conversion result register m (ADnCRm) or A/Dn conversion result extension register a (ADnECRa)) is as follows:

$$\text{ADCR} = \text{INT} \left( \frac{V_{\text{IN}}}{\text{AV}_{\text{REFP}}} \times 4,096 + 0.5 \right)$$

or,

$$(\text{ADCR} - 0.5) \times \frac{\text{AV}_{\text{REFP}}}{4,096} \leq V_{\text{IN}} < (\text{ADCR} + 0.5) \times \frac{\text{AV}_{\text{REFP}}}{4,096}$$

INT( ): Function that returns the integer of the value in ( )

$V_{\text{IN}}$ : Analog input voltage

$\text{AV}_{\text{REFP}}$ :  $\text{AV}_{\text{REFP}n}$  pin voltage

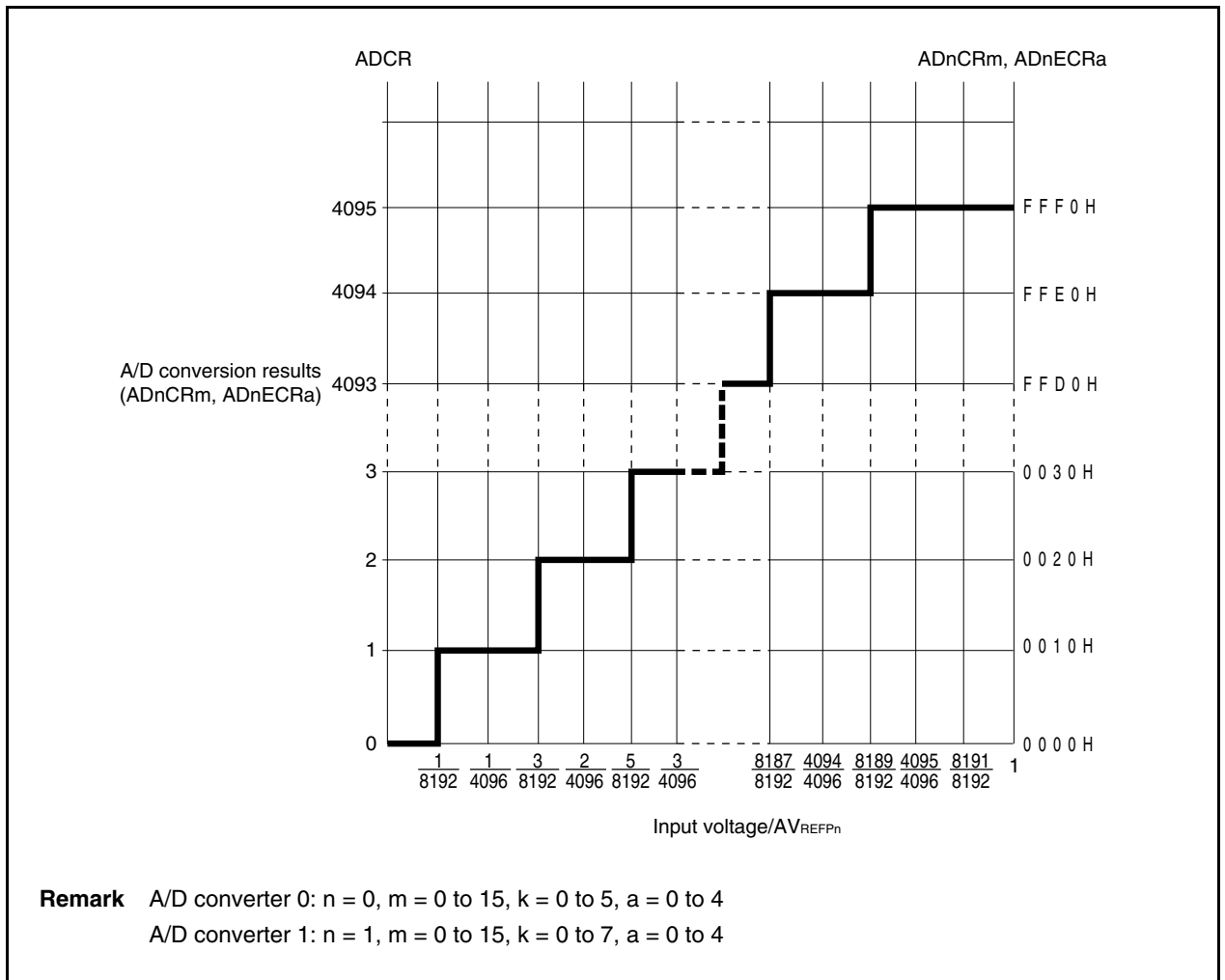
ADCR: Value of A/Dn conversion result register m (ADnCRm) or A/Dn conversion result extension register a (ADnECRa)

The relationship between the analog input voltage and the A/D conversion result is shown below.

**Remark** A/D converter 0: n = 0, m = 0 to 15, k = 0 to 5, a = 0 to 4

A/D converter 1: n = 1, m = 0 to 15, k = 0 to 7, a = 0 to 4

Figure 12-11. Relationship Between Analog Input Voltage and A/D Conversion Results

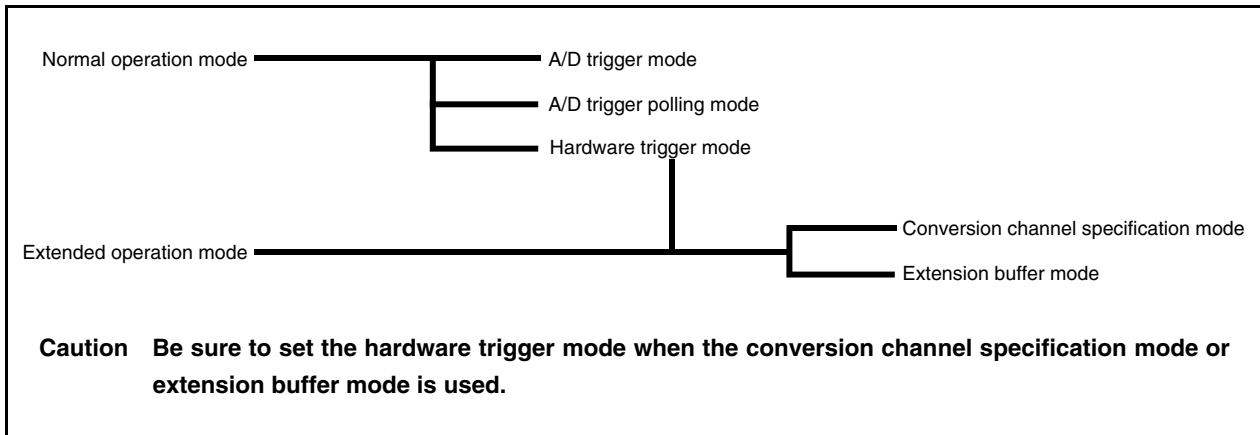


### 12.4.3 Operation mode

Various conversion operations can be specified for the A/D converters 0 and 1 by specifying the operation mode. The operation mode is set by the ADnSCM, ADnCTC, ADnCHEN, ADnCTL0, ADnTSEL, ADnCH1, ADnCH2, ADLTS1, ADLTS2, and ADnOCKS registers.

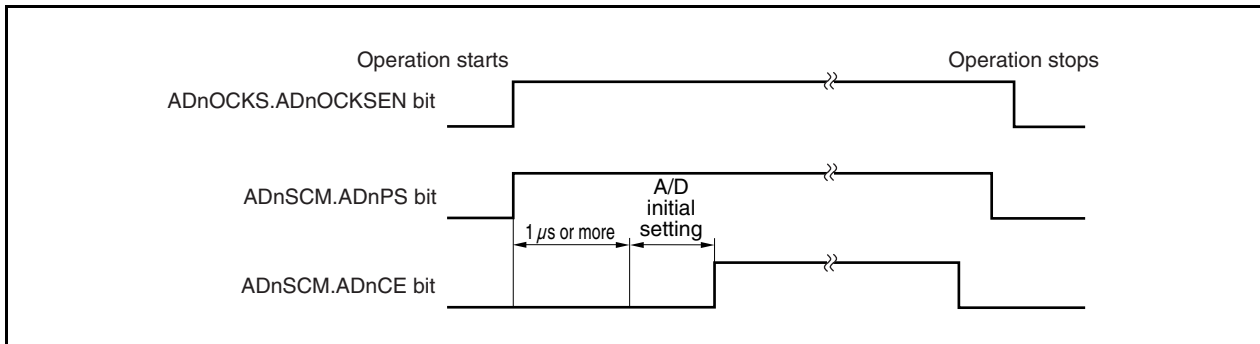
The following shows the relationship between the operation modes.

**Remark** n = 0, 1



### 12.4.4 Operation setting

Start or stop the operation of A/D converters 0 and 1 in the following procedure.



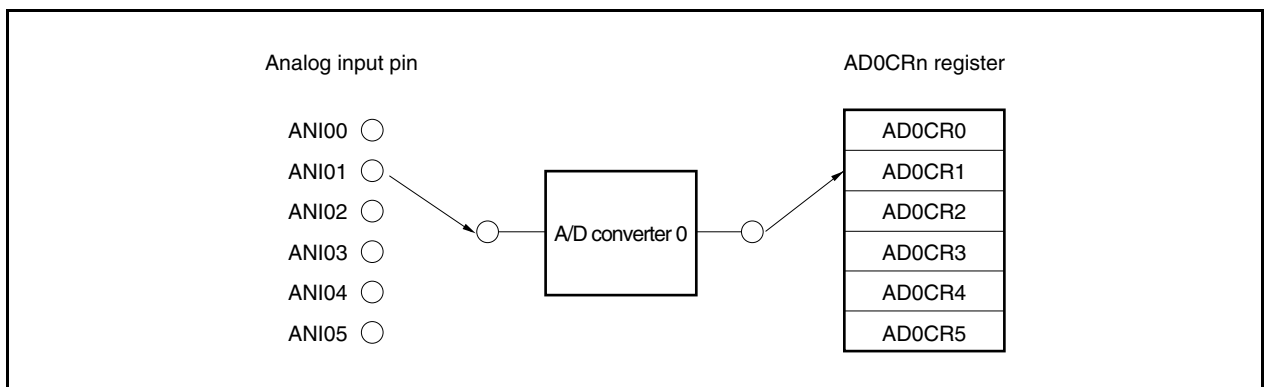
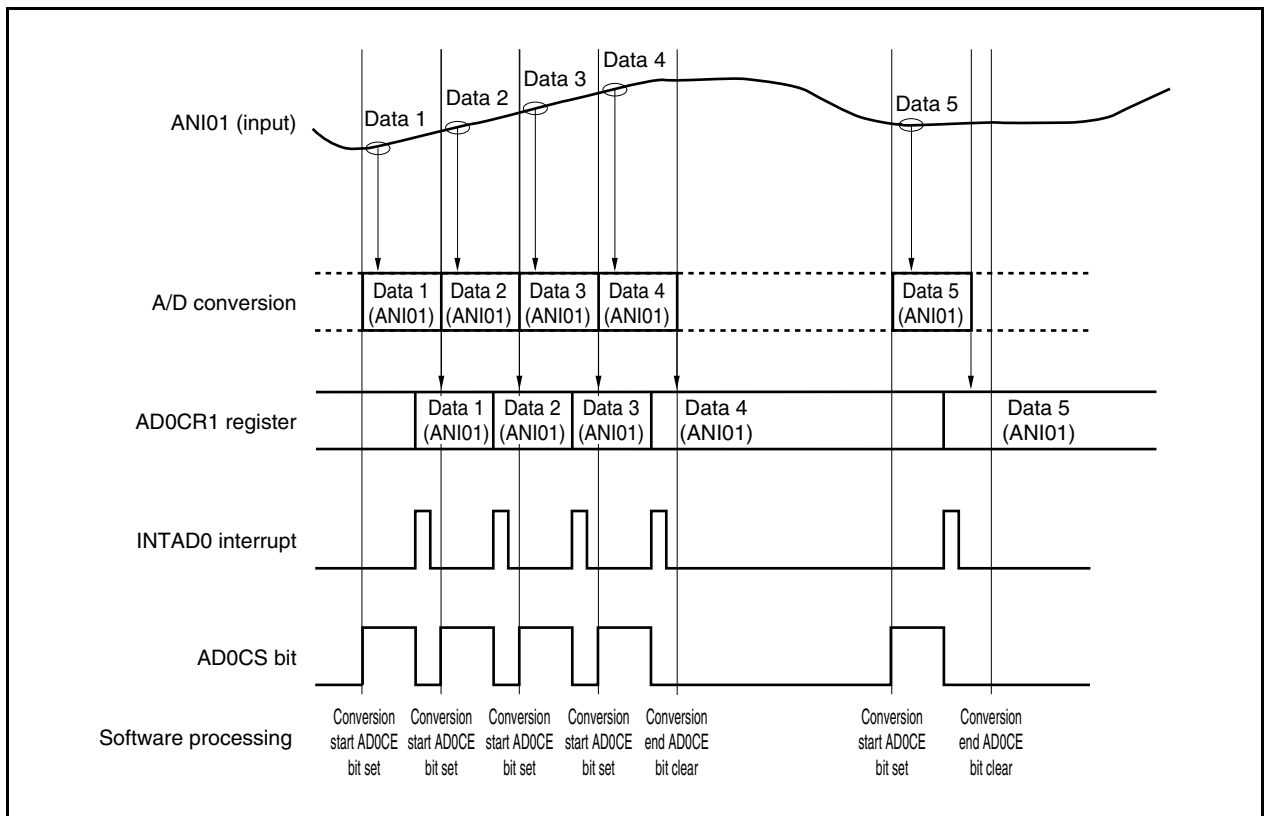
**12.4.5 Operation of 1-channel conversion**

The signal of one analog input pin (ANInk) specified by the ADnCHEN register is converted. The result of conversion is stored in the ADnCRk register corresponding to the ANInk pin. The ANInk pin and ADnCRk register correspond to each other on a one-to-one basis, and an A/Dn conversion end interrupt request signal (INTADn) is generated each time conversion has been completed.

After completion of A/D conversion, the conversion operation is stopped in the A/D trigger mode or A/D trigger polling mode. In the hardware trigger mode, the A/D converter waits for a trigger.

**Remark** A/D converter 0: n = 0, k = 0 to 5  
 A/D converter 1: n = 1, k = 0 to 7

**Figure 12-12. Operation of 1-Channel Conversion (in A/D Trigger Mode): A/D Converter 0**



#### 12.4.6 Operation of multiple channel conversion

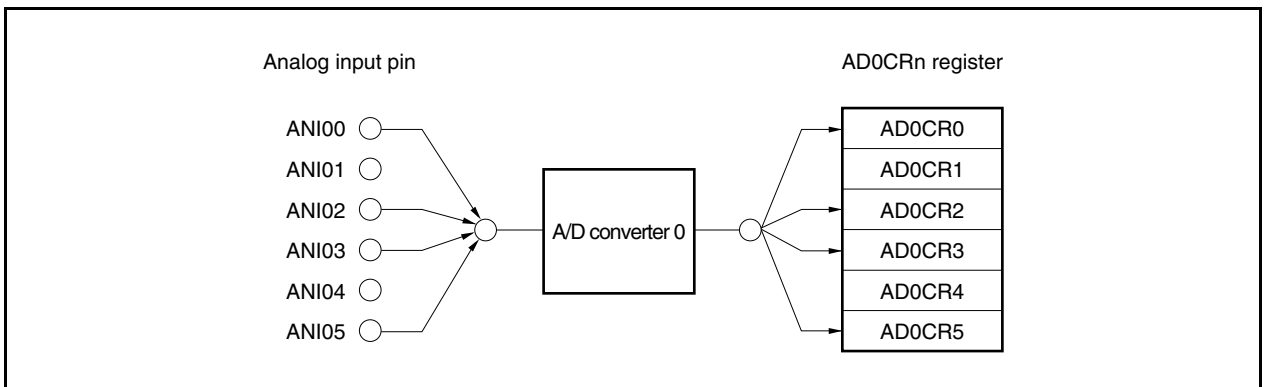
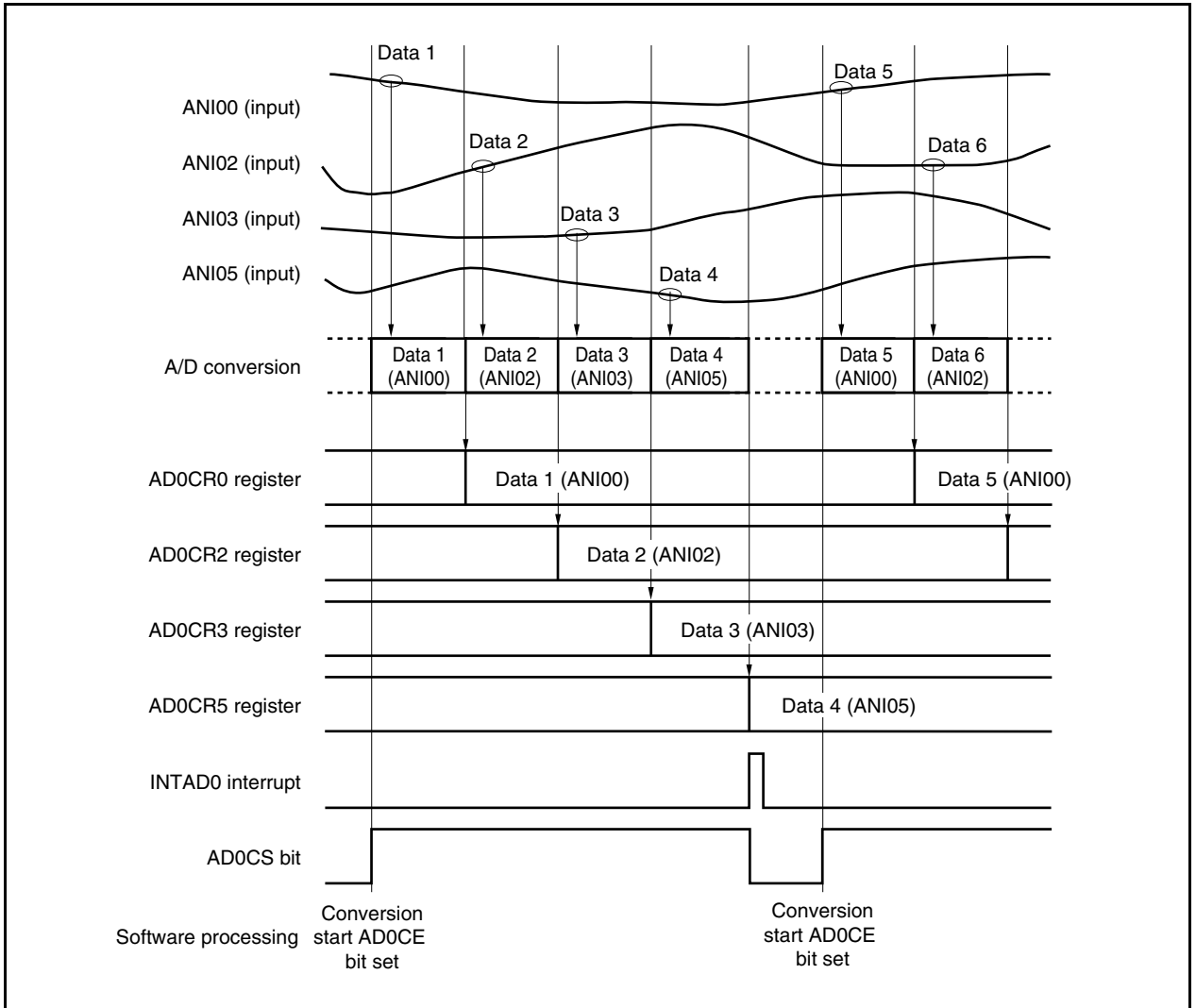
The signals of two or more analog input pins (ANInk) specified by the ADnCHEN register are converted. The signals are sequentially converted starting from the pin with the lowest number (in the example in Figure 12-13, ANI00 → ANI02 → ANI03 → ANI05). An analog input pin that is not specified is skipped. The result of conversion is stored in the ADnCRk register corresponding to the ANInk pin. The ANInk pin and ADnCRk register correspond to each other on a one-to-one basis. When conversion of the signal of the specified analog input pins is completed, an A/Dn conversion end interrupt request signal (INTADn) is generated.

After completion of A/D conversion, the conversion operation is stopped in the A/D trigger mode or A/D trigger polling mode. In the hardware trigger mode, the A/D converter waits for a trigger.

**Remark** A/D converter 0: n = 0, k = 0 to 5  
A/D converter 1: n = 1, k = 0 to 7



Figure 12-13. Operation of Multiple Channel Conversion (in A/D Trigger Mode): A/D Converter 0



**12.4.7 A/D trigger mode (normal operation mode)**

A/D conversion is started when the ADnSCM.ADnCE bit is set to 1.

When conversion is started, the ADnSCM.ADnCS bit is set to 1 (conversion is in progress).

If the ADnSCM register is written during A/D conversion, the conversion is stopped and started again from the beginning.

**(1) Operation of 1-channel conversion**

The signal of one analog input pin (ANInk) is converted once and the result is stored in one ADnCRk register. The ANInk pin and ADnCRk register correspond to each other on a one-to-one basis.

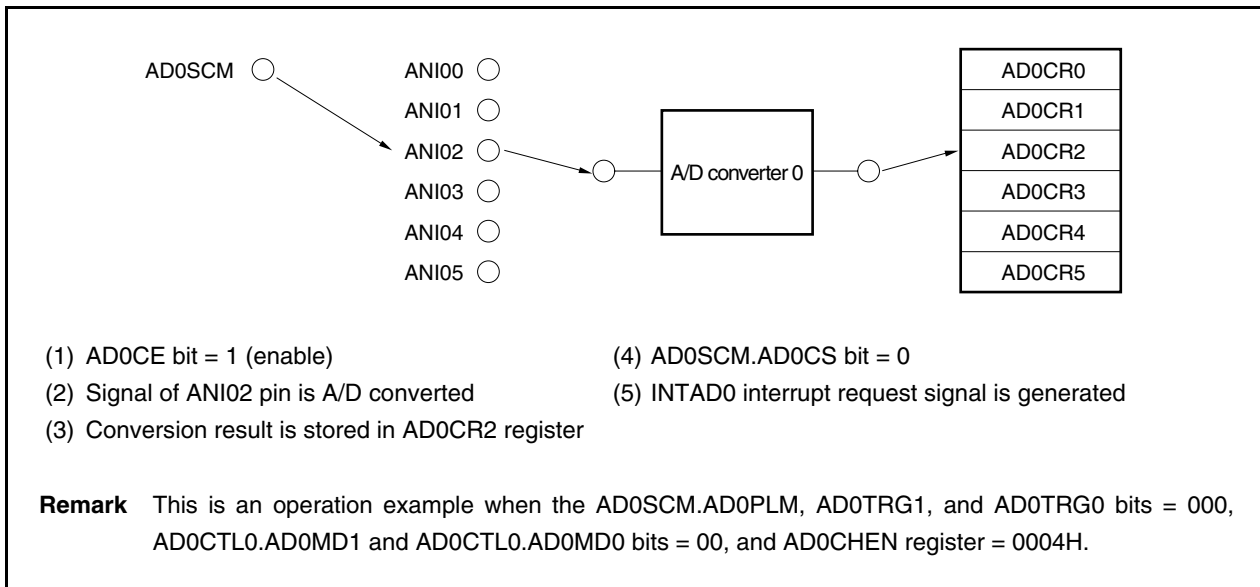
Each time conversion has been completed, an A/Dn conversion end interrupt request signal (INTADn) is generated. After A/D conversion is completed, The A/D converter stops conversion operation with the ADnSCM.ADnCE bit remaining set to 1. The A/D conversion can be restarted by setting the ADnCE bit to 1.

This operation is suitable for an application where the result of A/D conversion should be read each time conversion has been completed once.

| Analog Input Pin | A/D Conversion Result Register |
|------------------|--------------------------------|
| ANInk            | ADnCRk                         |

**Remark** A/D converter 0: n = 0, k = 0 to 5  
 A/D converter 1: n = 1, k = 0 to 7

**Figure 12-14. Example of 1-Channel Conversion Operation (A/D Trigger Mode): A/D Converter 0**



**(2) Operation of multiple channel conversion**

The signals of two or more analog input pins specified by the ADnCHEN register are converted sequentially starting from the pin with the lowest number. The result of conversion is stored in the ADnCRk register corresponding to the analog input pin.

When conversion of the signals of all the specified analog input pins is completed, an A/Dn conversion end interrupt request signal (INTADn) is generated. After A/D conversion is completed, the A/D converter stops conversion operation with the ADnSCM.ADnCE bit remaining set to 1. The A/D conversion can be restarted by setting the ADnCE bit to 1.

This operation is suitable for an application where two or more analog input signals should be monitored.

| Analog Input Pin      | A/D Conversion Result Register |
|-----------------------|--------------------------------|
| ANInk <sup>Note</sup> | ADnCRk                         |
|                       |                                |
| ANInk <sup>Note</sup> | ADnCRk                         |

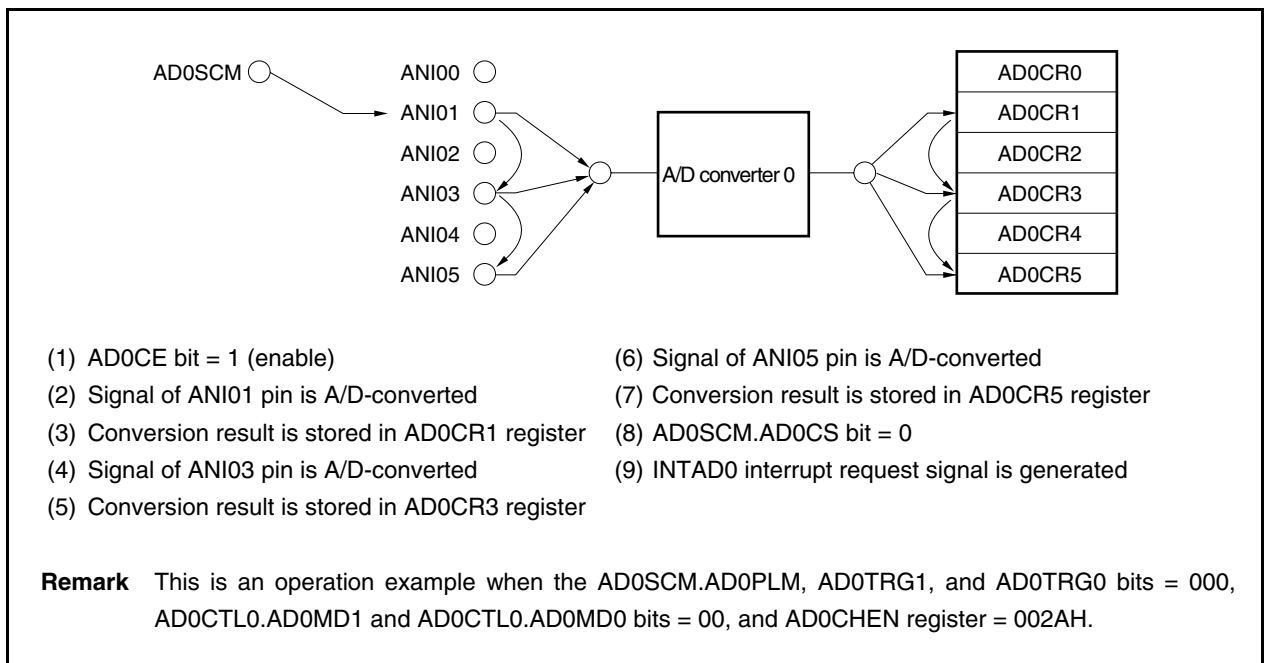
**Note** Two or more can be specified by the ADnCHEN register.

However, A/D conversion is sequentially executed starting from the pin with the lowest number.

**Remark** A/D converter 0: n = 0, k = 0 to 5

A/D converter 1: n = 1, k = 0 to 7

**Figure 12-15. Example of Multiple Channel Conversion Operation (A/D Trigger Mode): A/D Converter 0**





**(2) Operation of multiple channel conversion**

The signals of two or more analog input pins specified by the ADnCHEN register are converted sequentially starting from the pin with the lowest number. The result of conversion is stored in the ADnCRk register corresponding to the analog input pin.

When conversion of the signals of all the specified analog input pins is completed, an A/Dn conversion end interrupt request signal (INTADn) is generated. A/D conversion is repeated until the ADnSCM.ADnCE bit is set to 0. The conversion operation is stopped when the ADnCE bit is cleared to 0.

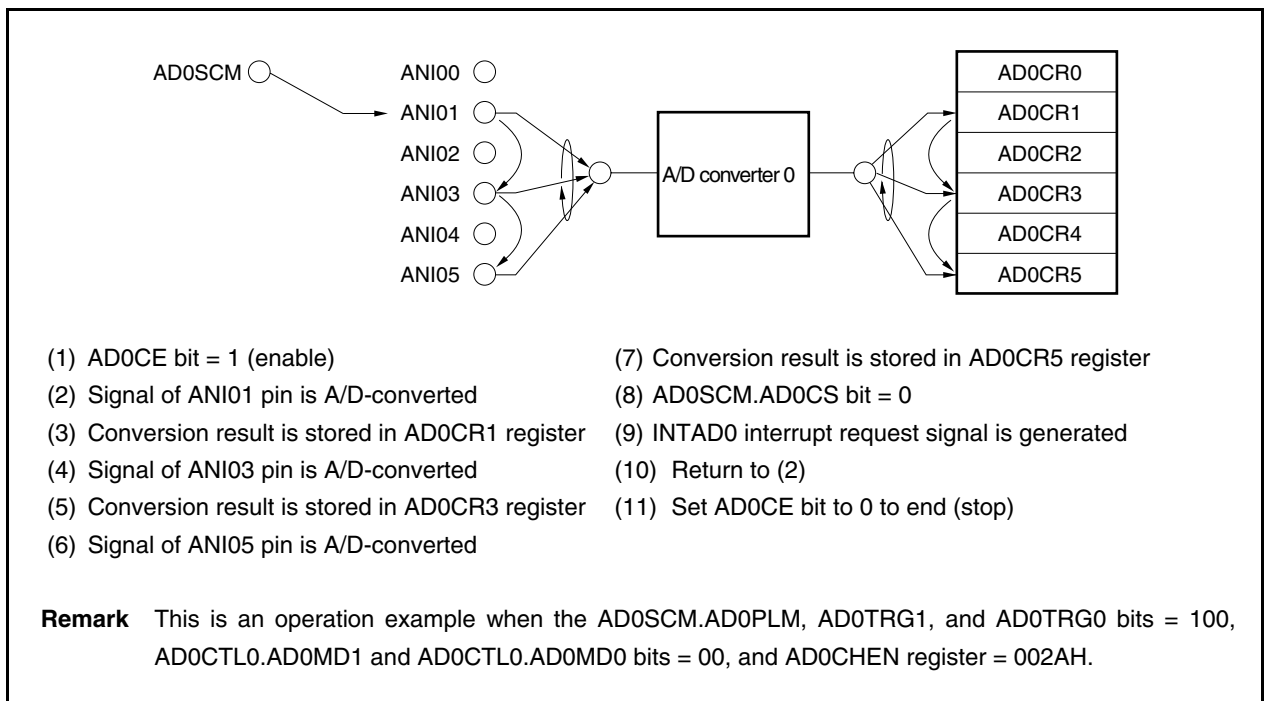
It is not necessary to set the ADnCE bit to restart the conversion operation in the A/D trigger polling mode. This operation is suitable for an application where the A/D conversion value is always read.

| Analog Input Pin      | A/D Conversion Result Register |
|-----------------------|--------------------------------|
| ANInk <sup>Note</sup> | ADnCRk                         |
|                       |                                |
| ANInk <sup>Note</sup> | ADnCRk                         |

**Note** Two or more can be specified by the ADnCHEN register. However, A/D conversion is sequentially executed starting from the pin with the lowest number.

**Remark** A/D converter 0: n = 0, k = 0 to 5  
A/D converter 1: n = 1, k = 0 to 7

**Figure 12-17. Example of Multiple Channel Conversion Operation (A/D Trigger Polling Mode): A/D Converter 0**



**12.4.9 Hardware trigger mode (normal operation mode)**

The A/D converter waits for a trigger when the ADnSCM.ADnCE bit is set to 1, and starts A/D conversion when a trigger specified by the ADnTSEL.ADnTRGSEL11 and ADnTSEL.ADnTRGSEL10 bits is generated.

When conversion is started, the ADnSCM.ADnCS bit is set to 1 (conversion is in progress).

If the ADnSCM register is written during A/D conversion, the conversion is stopped and becomes trigger wait status again.

**(1) Operation of 1-channel conversion**

The signal of one analog input pin (ANInk) is converted once, using a signal specified by the ADnTSEL.ADnTRGSEL11 and ADnTSEL.ADnTRGSEL10 bits as a trigger, and the result of conversion is stored in one ADnCRk register. The ANInk pin and ADnCRk register correspond to each other on a one-to-one basis.

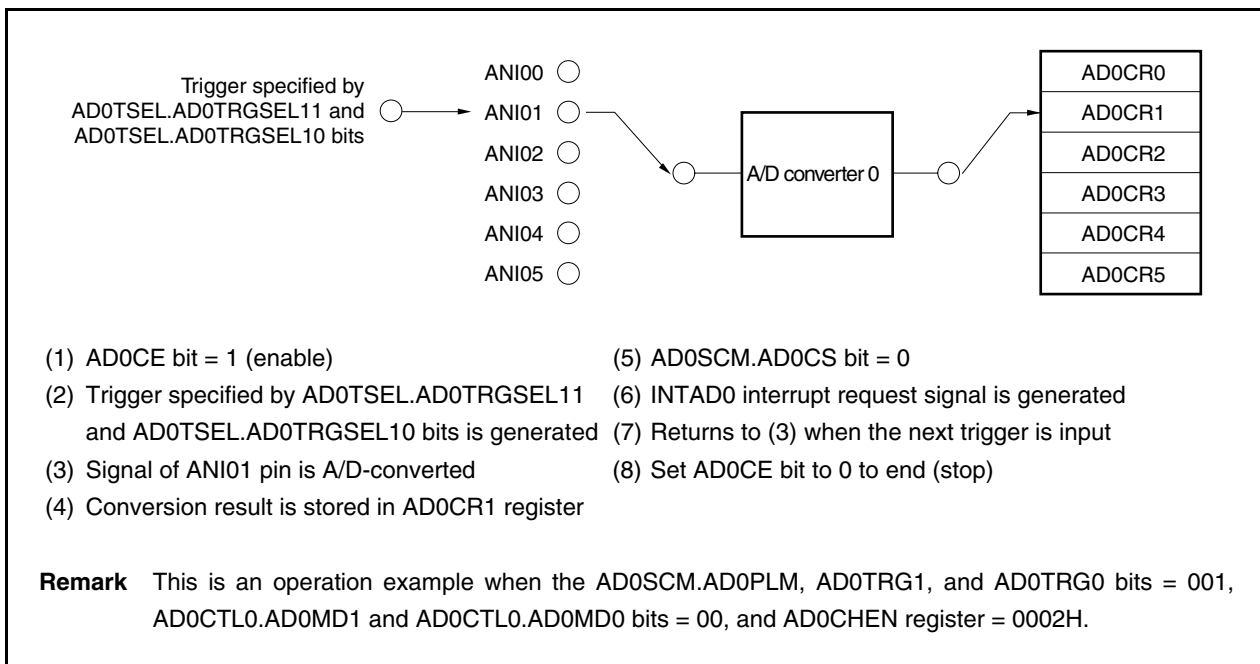
Each time conversion has been completed, an A/Dn conversion end interrupt request signal (INTADn) is generated. After completing the conversion, the converter waits for the trigger with the ADnSCM.ADnCE bit set to 1.

This operation is suitable for an application where the result of A/D conversion should be read each time conversion by one trigger has been completed.

| Analog Input Pin | A/D Conversion Result Register |
|------------------|--------------------------------|
| ANInk            | ADnCRk                         |

**Remark** A/D converter 0: n = 0, k = 0 to 5  
 A/D converter 1: n = 1, k = 0 to 7

**Figure 12-18. Example of 1-Channel Conversion Operation (Hardware Trigger Mode): A/D Converter 0**



**(2) Operation of multiple channel conversion**

The signals of two or more analog input pins specified by the ADnCHEN register are sequentially converted, starting from the pin with the lowest number, using a signal specified by the ADnTSEL.ADnTRGSEL11 and ADnTSEL.ADnTRGSEL10 bits as a trigger. The result of conversion is stored in the ADnCRk register corresponding to the analog input pin.

When conversion of the signals of all the specified analog input pins is completed, an A/Dn conversion end interrupt request signal (INTADn) is generated. After completion of conversion, the A/D converter waits for the trigger with the ADnSCM.ADnCE bit remaining set to 1.

This operation is suitable for an application where two or more analog input signals should be monitored when the trigger is generated.

| Analog Input Pin      | A/D Conversion Result Register |
|-----------------------|--------------------------------|
| ANInk <sup>Note</sup> | ADnCRk                         |
|                       |                                |
| ANInk <sup>Note</sup> | ADnCRk                         |

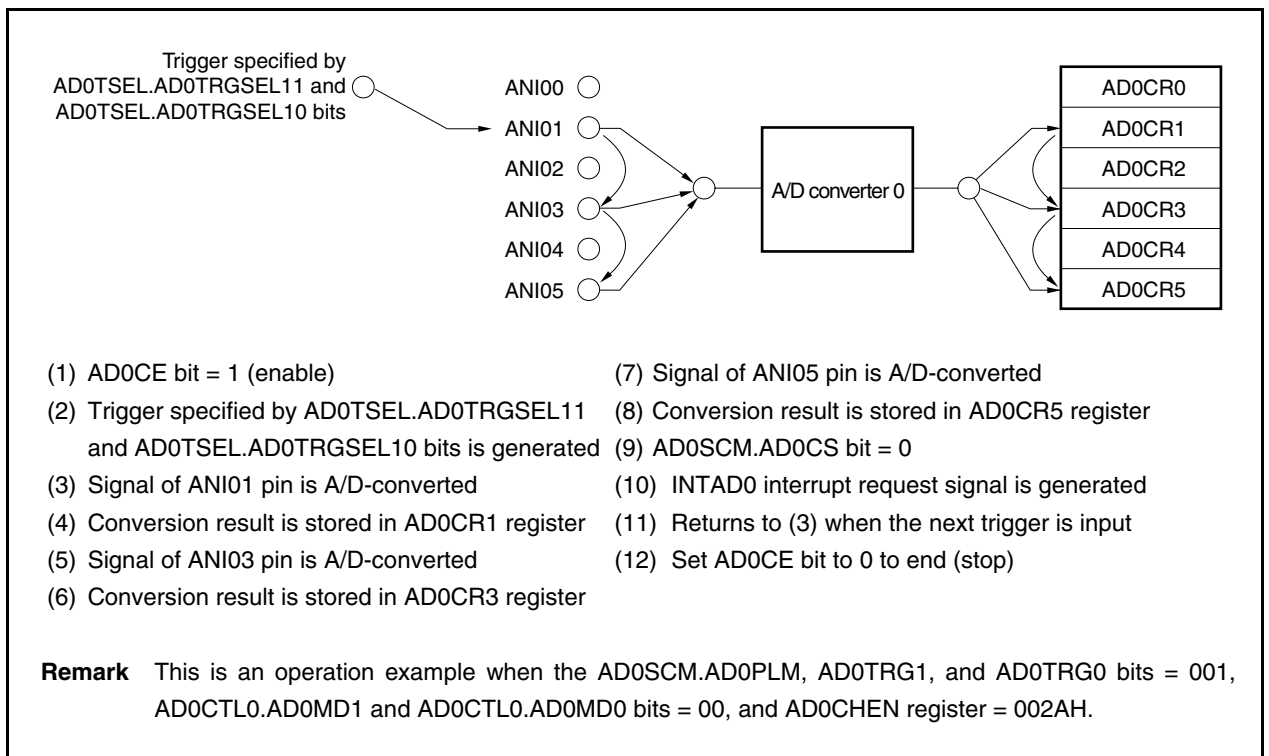
**Note** Two or more can be specified by the ADnCHEN register.

However, A/D conversion is sequentially executed starting from the pin with the lowest number.

**Remark** A/D converter 0: n = 0, k = 0 to 5

A/D converter 1: n = 1, k = 0 to 7

**Figure 12-19. Example of Multiple Channel Conversion Operation (Hardware Trigger Mode): A/D Converter 0**



**12.4.10 Conversion channel specification mode (extension operation mode)**

When the ADnSCM.ADnCE bit is set to 1, the A/D converter waits for a trigger. When selection trigger 1 specified by the ADnTSEL.ADnTRGSEL11 and ADnTSEL.ADnTRGSEL10 bits is generated, the converter starts A/D conversion.

When conversion is started, the ADnSCM.ADnCS bit is set to 1 (conversion is in progress).

If the ADnSCM register is written during A/D conversion operation, the conversion is stopped and the converter waits for the trigger again.

The analog input pin is specified by the ADnCH1.ADnTRGCH12 to ADnCH1.ADnTRGCH10 and ADnCH1.ADnTRGCH16 to ADnCH1.ADnTRGCH14 bits. Each time selection trigger 1 is generated, the analog input pins specified by the ADnCH1.ADnTRGCH12 to ADnCH1.ADnTRGCH10 and ADnCH1.ADnTRGCH16 to ADnCH1.ADnTRGCH14 bits are sequentially selected.

The signal of a specified analog input pin is converted the number of times specified by the ADnCHEN register (up to 16 times), using selection trigger 1 as the trigger, and the result is stored in the ADnCRm register specified by the ADnCHEN register. The conversion results are sequentially stored from ADnCR0.

When the signal of the specified analog input pin has been converted the number of times (up to 16 times) specified by the ADnCHEN register, an A/Dn conversion end interrupt request signal (INTADn) is generated. After A/D conversion is completed, the A/D converter waits for the trigger with the ADnSCM.ADnCE bit remaining set to 1.

This operation is suitable for an application where two or more analog input signals should be monitored.

| Selection Trigger   | Analog Input Pin        | A/D Conversion Result Extension Register |
|---------------------|-------------------------|--|
| Selection trigger 1 | ANInx <sup>Note 1</sup> | ADnCR0 <sup>Note 3</sup>                 |
|                     | ANInx <sup>Note 1</sup> |  |
|                     | ANInx <sup>Note 1</sup> | ADnCRm <sup>Note 3</sup>                 |
| Selection trigger 2 | ANIny <sup>Note 2</sup> | ADnCR0 <sup>Note 3</sup>                 |
|                     | ANIny <sup>Note 2</sup> |  |
|                     | ANIny <sup>Note 2</sup> | ADnCRm <sup>Note 3</sup>                 |

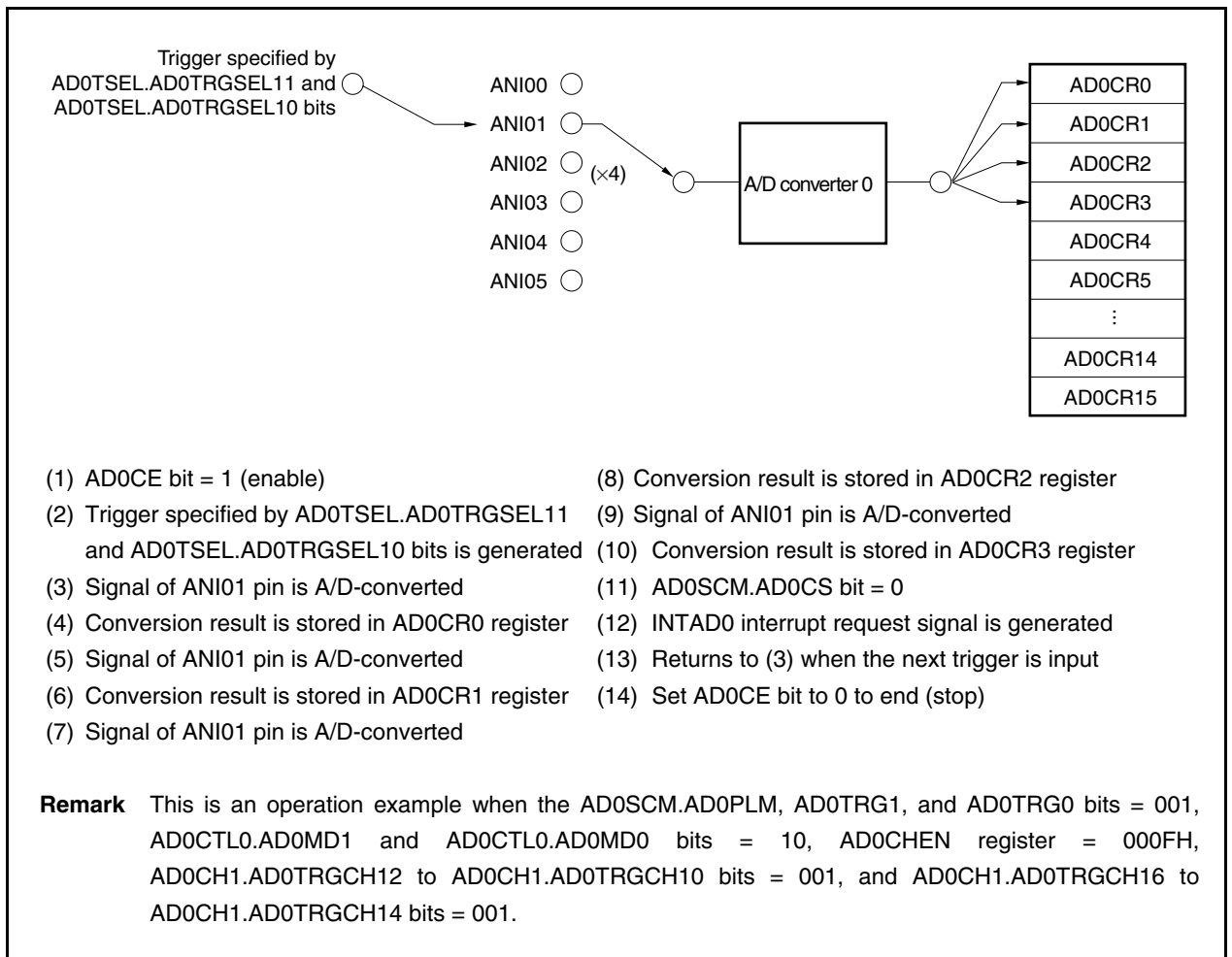
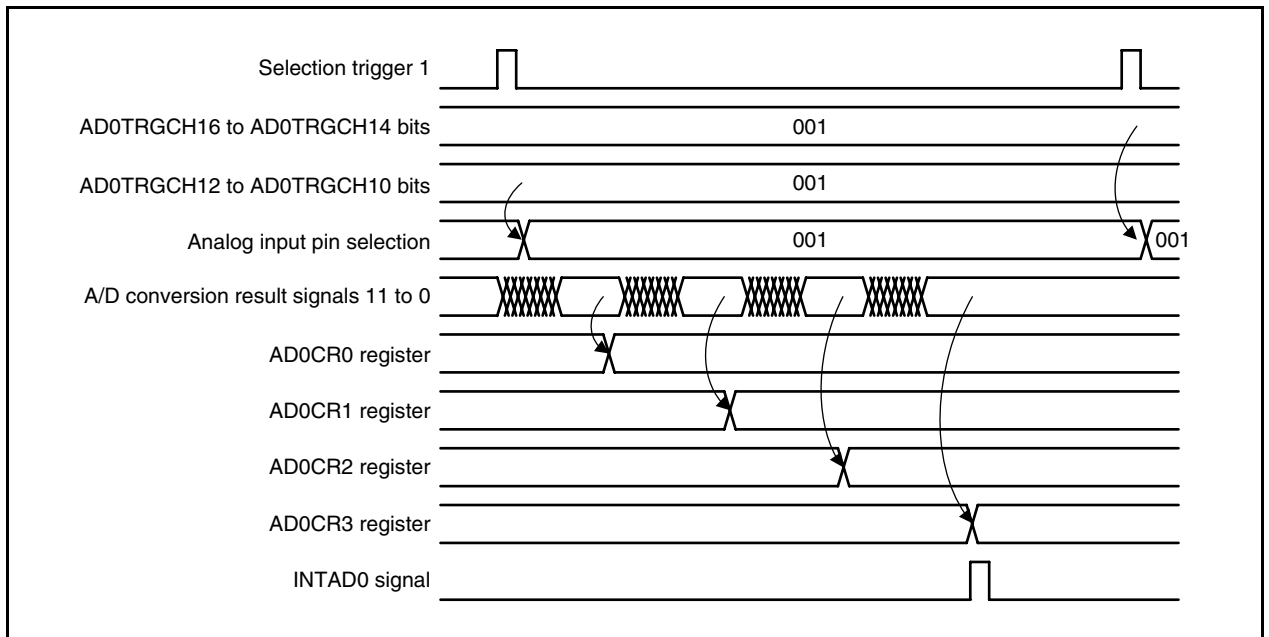
- Notes**
1. Set by ADnCH1.ADnTRGCH12 to ADnCH1.ADnTRGCH10 bits
  2. Set by ADnCH1.ADnTRGCH16 to ADnCH1.ADnTRGCH14 bits
  3. Two or more times can be set by the ADnCHEN register.

- Cautions**
1. Be sure to set the hardware trigger mode as the conversion channel specification mode.
  2. Be sure to set the ADnCHEN register using the lower bits, justifying to the bottom. Any other setting is prohibited.
  3. Setting of the ADnCH2 register is invalid.
  4. The ADnECRa, ADnECRaH, ADnFLG, and ADnFLGB registers are not used. If these registers are read, 0000H and 00H are read.
  5. Selection trigger 1 is ignored if it is generated during A/D conversion operation. The next selection trigger 1 is accepted when a trigger is generated after completion of A/D conversion (after generation of the INTADn signal).

**Remark** A/D converter 0: n = 0, k = 0 to 5, m = 0 to 15  
 A/D converter 1: n = 1, k = 0 to 7, m = 0 to 15



Figure 12-20. Example of Operation in Conversion Channel Specification Mode: A/D Converter 0



**12.4.11 Extension buffer mode (extension operation mode)**

When the ADnSCM.ADnCE bit is set to 1, the A/D converter waits for a trigger. When selection trigger 1 specified by the ADnTSEL.ADnTRGSEL11 and ADnTSEL.ADnTRGSEL10 bits or selection trigger 2 specified by the ADnTSEL.ADnTRGSEL21 and ADnTSEL.ADnTRGSEL20 bits is generated, the converter starts A/D conversion.

When conversion is started, the ADnSCM.ADnCS bit is set to 1 (conversion is in progress).

If the ADnSCM register is written during A/D conversion operation, the conversion is stopped and the converter waits for the trigger again.

The analog input pin for selection trigger x is specified by the ADnCHx.ADnTRGCHx2 to ADnCHx.ADnTRGCHx0 and ADnCHx.ADnTRGCHx6 to ADnCHx.ADnTRGCHx4 bits. Each time selection trigger x is generated, the analog input pins specified by the ADnCHx.ADnTRGCHx2 to ADnCHx.ADnTRGCHx0 and ADnCHx.ADnTRGCHx6 to ADnCHx.ADnTRGCHx4 bits are sequentially selected.

When selection trigger 1 is used, the signal of the analog input pin specified by the ADnTRGCH12 to ADnTRGCH10 bits is converted when the trigger is generated for the first time. The result is stored in the A/Dn conversion result extension buffer register 0 and an A/Dn conversion end interrupt request signal (INTADn) is generated. When the trigger is generated the second time, the signal of the analog input pin specified by the ADnTRGCH16 to ADnTRGCH14 bits is converted. The result is stored in the A/Dn conversion result extension buffer register 0 and, at the same time, the first value stored in the A/Dn conversion result extension buffer register 0 is stored in the A/Dn conversion result extension buffer register 1. Then the INTADn interrupt request signal is generated. For A/D conversion using selection trigger 1, up to three A/Dn conversion result extension buffer registers, 0 to 2, can be used. When selection load trigger 1 is later generated, the values of the A/Dn conversion result extension buffer registers 0 to 2 are transferred to the ADnECR0 to ADnECR2 registers. After A/D conversion is completed, the converter waits for the trigger with the ADnSCM.ADnCE bit remaining set to 1.

When selection trigger 2 is used, the signal of the analog input pin specified by the ADnTRGCH22 to ADnTRGCH20 bits is converted when the trigger is generated for the first time, and the result is stored in the A/Dn conversion end extension buffer register 3. Then an A/Dn conversion end interrupt request signal (INTADn) is generated. When the trigger is generated the second time, the signal of the analog input pin specified by the ADnTRGCH26 to ADnTRGCH24 bits is converted and the result is stored in the A/Dn conversion result extension buffer register 4. At the same time, the value stored first in the A/Dn conversion result extension buffer register 3 is stored in the A/Dn conversion result extension buffer register 4, and the INTADn interrupt request signal is generated. When selection trigger 2 is used for A/D conversion, up to two A/Dn conversion result extension buffer registers, 3 and 4, can be used. When selection load trigger 2 is generated again, the values of the A/Dn conversion result extension buffer registers 3 and 4 are transferred to and stored in the ADnECR3 and ADnECR4 registers. After A/D conversion is completed, the converter waits for the trigger with the ADnCE bit remaining set to 1.

Therefore, the contents of the ADnECR0 to ADnECR4 registers can be saved to RAM all at once.

This operation is suitable for an application where there is little time to save the conversion result and two or more analog input signals should be monitored when a trigger is generated.

| Selection Trigger   | Analog Input Pin        | A/D Conversion Result Extension Register |
|---------------------|-------------------------|--|
| Selection trigger 1 | ANInx <sup>Note 1</sup> | ADnECR0 to ADnECR2                       |
| Selection trigger 1 | ANIny <sup>Note 2</sup> | ADnECR0, ADnECR1                         |
| Selection trigger 1 | ANInx <sup>Note 1</sup> | ADnECR0                                  |
| Selection trigger 2 | ANIns <sup>Note 3</sup> | ADnECR3, ADnECR4                         |
| Selection trigger 2 | ANInt <sup>Note 4</sup> | ADnECR3                                  |

- Notes**
1. Set by ADnCH1.ADnTRGCH12 to ADnCH1.ADnTRGCH10 bits
  2. Set by ADnCH1.ADnTRGCH16 to ADnCH1.ADnTRGCH14 bits
  3. Set by ADnCH2.ADnTRGCH22 to ADnCH2.ADnTRGCH20 bits
  4. Set by ADnCH2.ADnTRGCH26 to ADnCH2.ADnTRGCH24 bits

- Cautions**
1. In the extension buffer mode, be sure to set the hardware trigger mode and the ADnCHEN register to 0001H.
  2. The conversion result is stored in the ADnECRa register. The value of the ADnCRm register is undefined.

**Remark** n = 0, 1

Figure 12-21. Block Diagram in Extension Buffer Mode

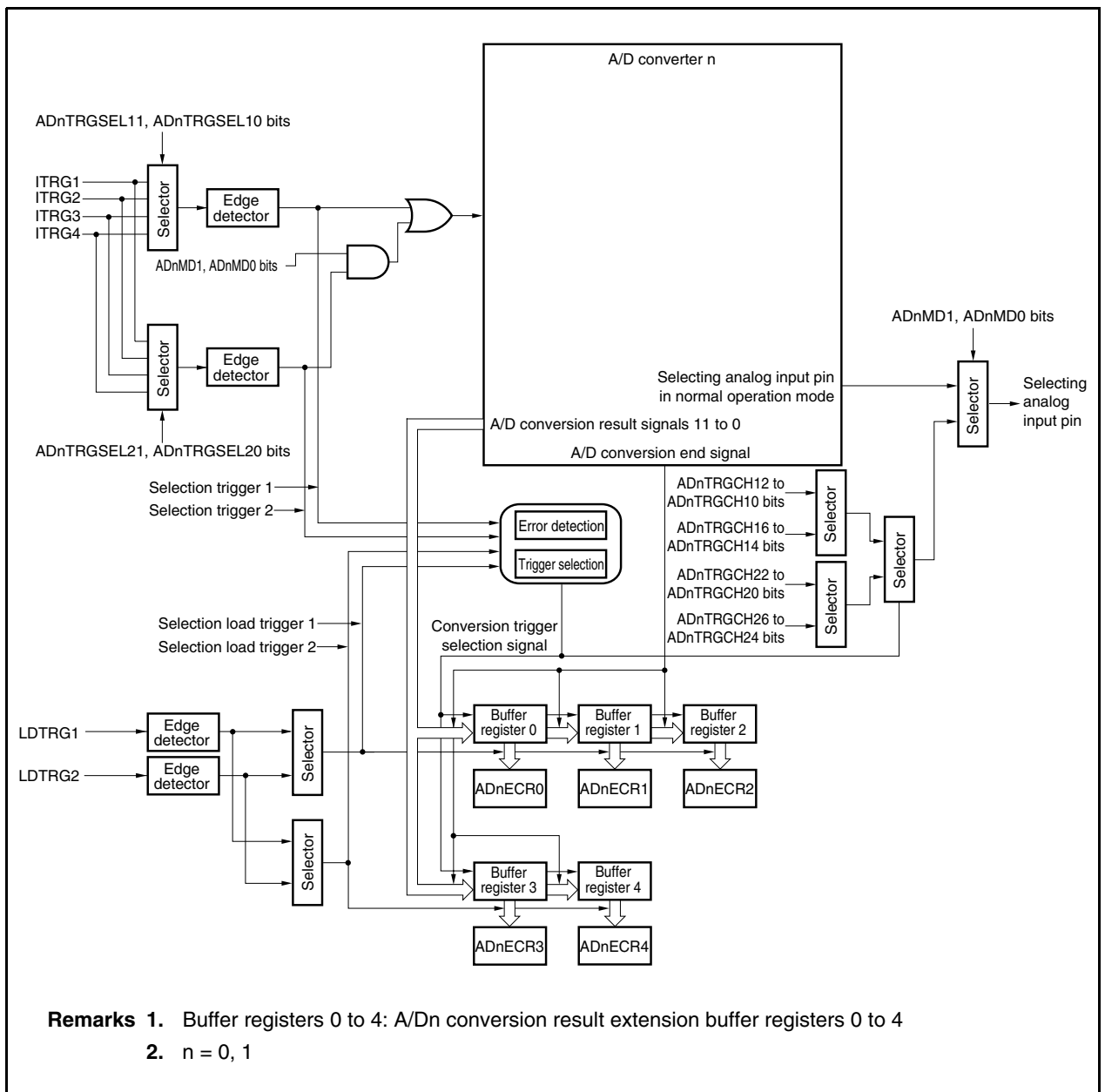


Figure 12-22. Example of Operation in Extension Buffer Mode: A/D Converter 0 (1/2)

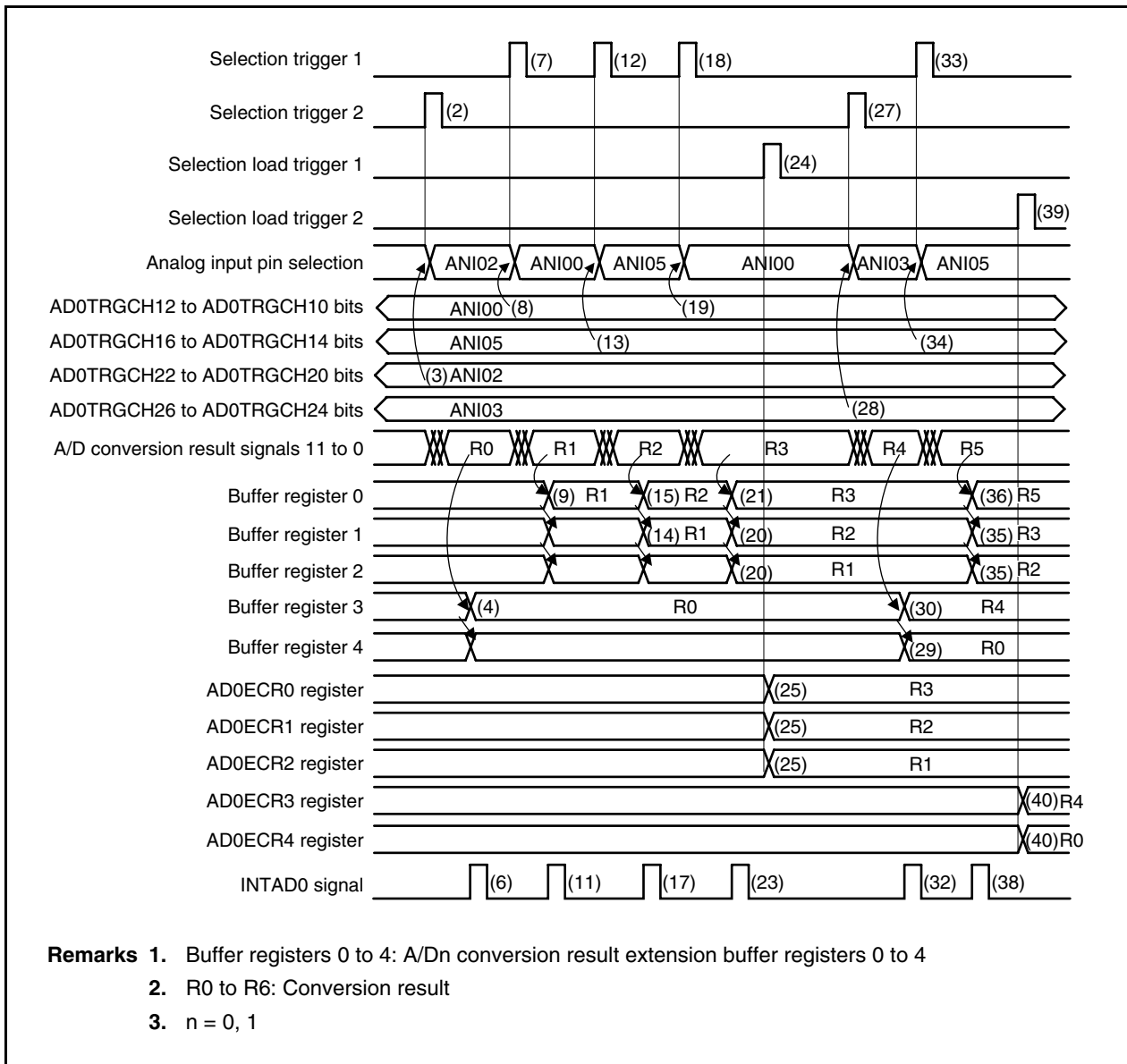


Figure 12-22. Example of Operation in Extension Buffer Mode: A/D Converter 0 (2/2)

- |   |  |
|---|--|
| (1) AD0CE bit = 1 (enable)  | (24) Selection load trigger 1 is generated   |
| (2) Selection trigger 2 is generated  | (25) Shifted from buffer registers 0 to 2, to AD0ECR0 to AD0ECR2                                 |
| (3) Signal of ANI02 pin is A/D-converted                                      | (26) AD0SCM.ADOCS bit = 0  |
| (4) Conversion result is stored in buffer register 3                          | (27) Selection trigger 2 is generated  |
| (5) AD0SCM.ADOCS bit = 0  | (28) Signal of ANI03 pin is A/D-converted  |
| (6) INTAD0 interrupt request signal is generated                              | (29) Shifted from buffer register 3 to buffer register 4   |
| (7) Selection trigger 1 is generated  | (30) Conversion result is stored in buffer register 3  |
| (8) Signal of ANI00 pin is A/D-converted                                      | (31) AD0SCM.ADOCS bit = 0  |
| (9) Conversion result is stored in buffer register 0                          | (32) INTAD0 interrupt request signal is generated  |
| (10) AD0SCM.ADOCS bit = 0   | (33) Selection trigger 1 is generated  |
| (11) INTAD0 interrupt request signal is generated                             | (34) Signal of ANI05 pin is A/D-converted  |
| (12) Selection trigger 1 is generated   | (35) Shifted from buffer register 0 to buffer register 1 to buffer register 2                    |
| (13) Signal of ANI05 pin is A/D-converted                                     | (36) Conversion result is stored in buffer register 0  |
| (14) Shifted from buffer register 0 to buffer register 1                      | (37) AD0SCM.ADOCS bit = 0  |
| (15) Conversion result is stored in buffer register 0                         | (38) INTAD0 interrupt request signal is generated  |
| (16) AD0SCM.ADOCS bit = 0   | (39) Selection load trigger 2 is generated   |
| (17) INTAD0 interrupt request signal is generated                             | (40) Shifted from buffer registers 3 and 4 to AD0ECR3 and AD0ECR4 registers                      |
| (18) Selection trigger 1 is generated   | (41) AD0SCM.ADOCS bit = 0  |
| (19) Signal of ANI00 pin is A/D-converted                                     | (42) When the next trigger is input, the operation is performed in accordance with that trigger. |
| (20) Shifted from buffer register 0 to buffer register 1 to buffer register 2 | (43) Set ADnCE bit to 0 to end (stop)  |
| (21) Conversion result is stored in buffer register 0                         |  |
| (22) AD0SCM.ADOCS bit = 0   |  |
| (23) INTAD0 interrupt request signal is generated                             |  |

**(1) Error detection function**

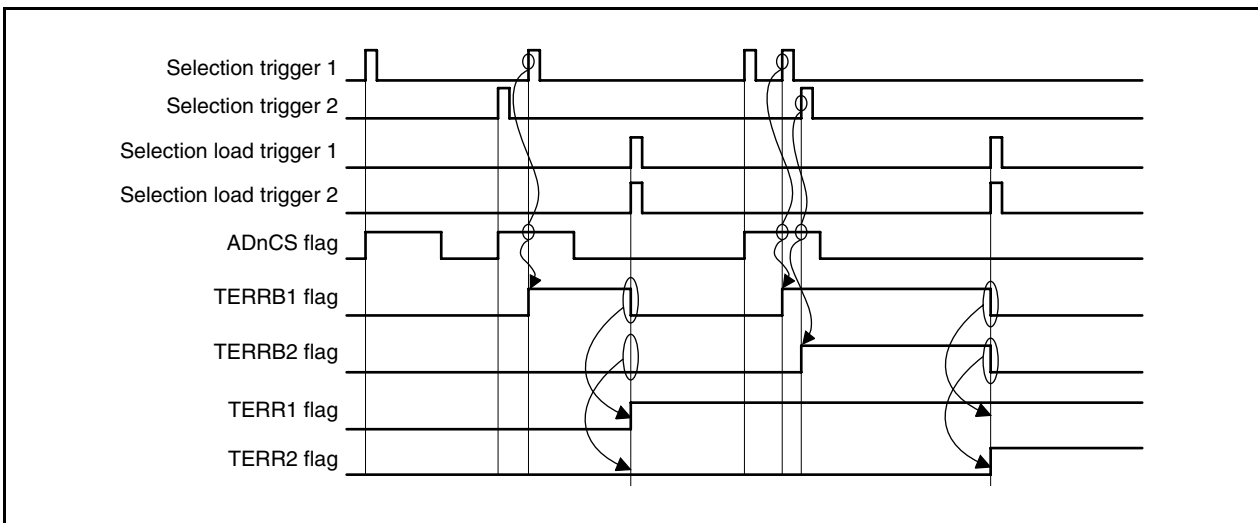
The extension buffer mode has an error detection function. If a trigger (selection trigger 1, selection trigger 2, selection load trigger 1, or selection load trigger 2) is generated during A/D conversion, an error occurs. The error is detected by the ADnFLG.ADnTERR2 and ADnFLG.ADnTERR1 flags, and ADnFLGB.ADnTERRB2 and ADnFLGB.ADnTERRB1 flags.

- Cautions**
1. Selection trigger 1, selection trigger 2, selection load trigger 1, and selection load trigger 2 are generated when asynchronous signals ITRG1 to ITRG4, LDTRG1, and LDTRG2 signals are synchronized. Although the timing of inputting these triggers seems to be the same, their simultaneous operation is not guaranteed because the asynchronous signals are synchronized.
  2. Selection trigger 1 or 2 is ignored, even if it is generated again, during a period of up to 2.5 basic clocks ( $f_{AD01}$ ) after the trigger is once generated (no error occurs).

**(a) Error detection by generation of selection trigger 1 or 2 during A/D conversion**

If selection trigger 1 is generated during A/D conversion, the ADnFLGB.ADnTERRB1 flag is set to 1 and A/D conversion by selection trigger 1 is ignored. If selection load trigger 1 is generated next, the value of the ADnTERRB1 flag is stored in the ADnFLG.ADnTERR1 flag.

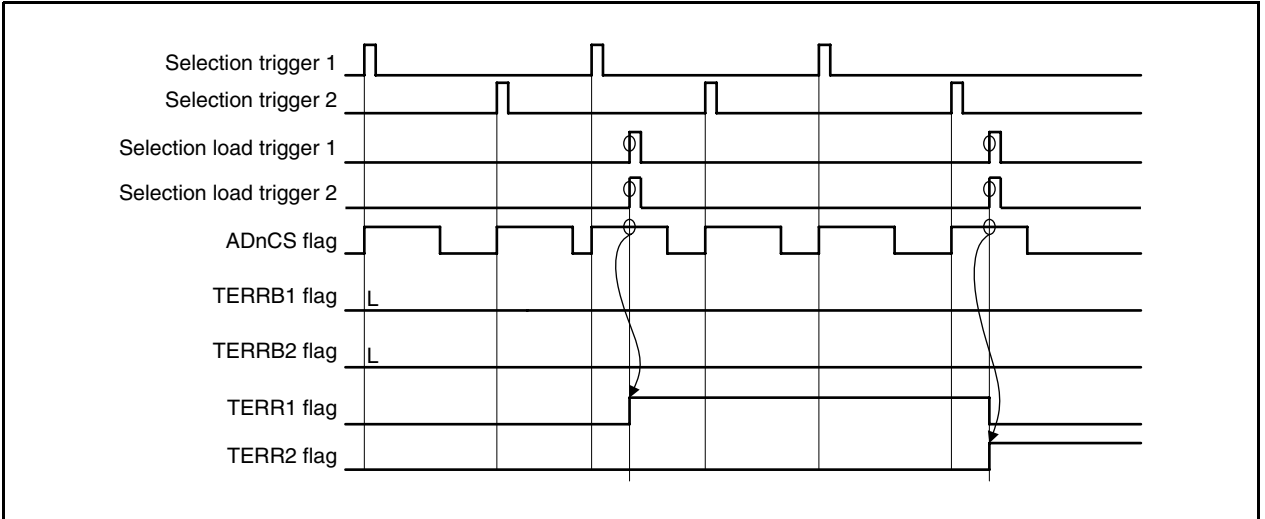
Similarly, if selection trigger 2 is generated during A/D conversion, the ADnFLGB.ADnTERRB2 flag is set to 1 and A/D conversion by selection trigger 2 is ignored. When selection load trigger 2 is generated next, the value of the ADnTERRB2 flag is stored in the ADnFLG.ADnTERR2 flag.



**(b) Error detection by generation of selection load trigger 1 or 2 during A/D conversion**

If selection load trigger 1 is generated during A/D conversion that uses selection trigger 1, the ADnFLG.ADnTERR1 flag is set to 1. A/D conversion and load operation are performed normally.

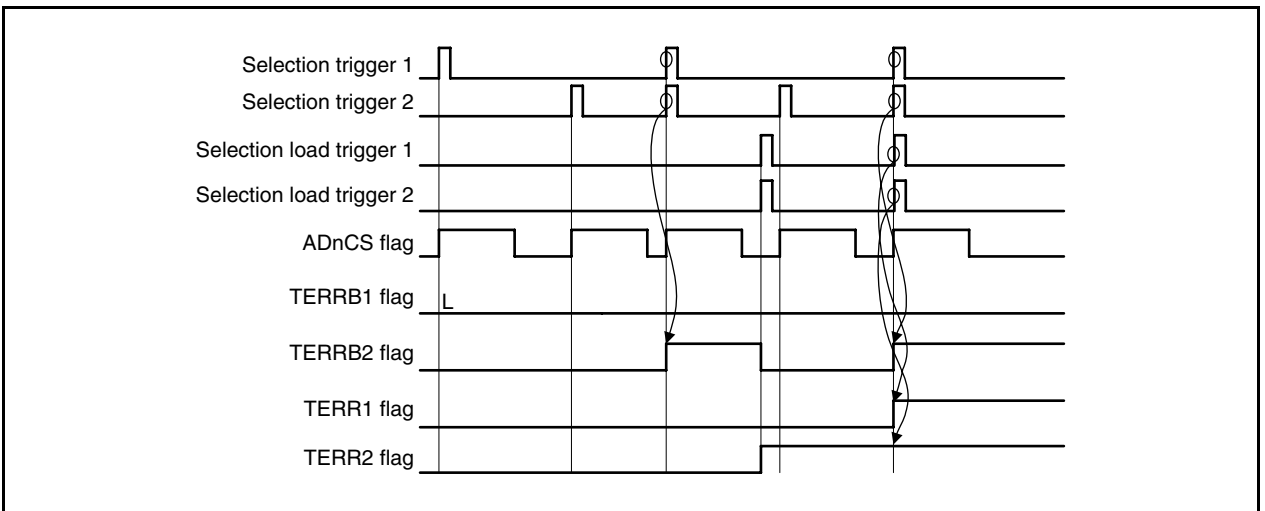
Similarly, if selection load trigger 2 is generated during A/D conversion that uses selection trigger 2, the ADnTERR2 flag is set to 1. A/D conversion and load operation are performed normally.



**(c) Error detection by simultaneous generation of selection triggers 1 and 2, and of selection triggers 1 and 2, and selection load triggers 1 and 2**

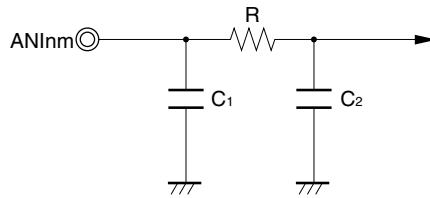
If selection triggers 1 and 2 are simultaneously generated, A/D conversion that uses selection trigger 1 is started and selection trigger 2 is ignored. Therefore, the ADnFLGB.ADnTERRB2 flag is set to 1.

If selection triggers 1 and 2, and selection load triggers 1 and 2 are simultaneously generated, the ADnFLGB.ADnTERRB2, ADnFLG.ADnTERR1, and ADnFLG.ADnTERR2 flags are set to 1. A/D conversion by selection trigger 1 and load operation of selection load triggers 1 and 2 are performed normally. Selection trigger 2 is ignored.



### 12.5 Internal Equivalent Circuit

The following figure shows the equivalent circuit of the analog input block.

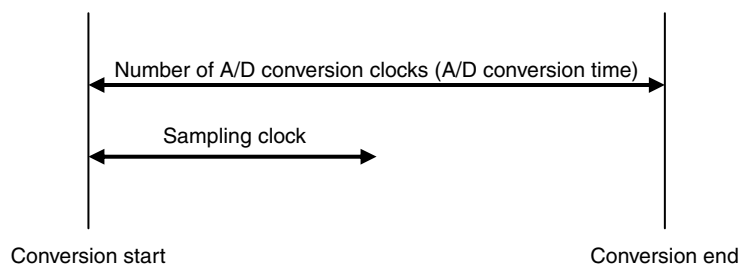


|        |                |                |
|--------|----------------|----------------|
| R      | C <sub>1</sub> | C <sub>2</sub> |
| 5.1 kΩ | 15 pF          | 3.9 pF         |

- Remarks**
1. The maximum values are shown (reference values).
  2. m = 0 to 5 when n = 0  
m = 0 to 7 when n = 1

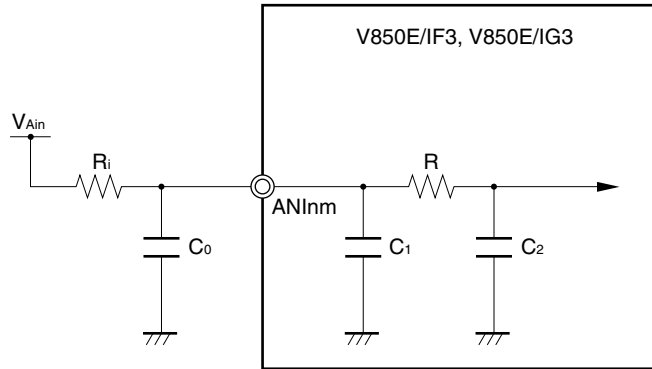
| ADnCTC register |            |            |            | Number of A/D conversion clocks (f <sub>AD01</sub> ) | Number of sampling clocks |
|-----------------|------------|------------|------------|--|---------------------------|
| ADnFR3 bit      | ADnFR2 bit | ADnFR1 bit | ADnFR0 bit |  |                           |
| 0               | 0          | 0          | 0          | 89   | 69.5                      |
| 0               | 0          | 0          | 1          | 88   | 68.5                      |
| 0               | 0          | 1          | 0          | 57   | 37.5                      |
| 0               | 0          | 1          | 1          | 56   | 36.5                      |
| 0               | 1          | 0          | 0          | 41   | 21.5                      |
| 0               | 1          | 0          | 1          | 40   | 20.5                      |
| 0               | 1          | 1          | 0          | 35   | 15.5                      |
| 0               | 1          | 1          | 1          | 34   | 14.5                      |
| 1               | 0          | 0          | 0          | 34   | 14.5                      |
| 1               | 0          | 0          | 1          | 33   | 13.5                      |
| 1               | 0          | 1          | 0          | 33   | 13.5                      |
| 1               | 0          | 1          | 1          | 32   | 12.5                      |
| 1               | 1          | 0          | 0          | 32   | 12.5                      |
| 1               | 1          | 0          | 1          | 31   | 11.5                      |
| 1               | 1          | 1          | 0          | 31   | 11.5                      |
| 1               | 1          | 1          | 1          | 30   | 10.5                      |

**Caution** Number of sampling clocks is included in number of A/D conversion clocks.





An example of calculating an overall error of A/D converters 0 and 1 is shown below.



| $f_{xx}$<br>(MHz) | A/D conversion<br>time ( $\mu s$ ) | Sampling ( $\mu s$ )        | R<br>( $k\Omega$ ) | $C_1$<br>(pF) | $C_2$<br>(pF) | $C_0$<br>(pF) | $R_i$<br>( $k\Omega$ ) | Sampling error<br>(LSB) <sup>Note</sup> |
|-------------------|------------------------------------|-----------------------------|--------------------|---------------|---------------|---------------|------------------------|---|
| 64                | 2.00<br>( $32/f_{AD01}$ )          | 0.78<br>( $12.5/f_{AD01}$ ) | 5.1                | 15            | 3.9           | 100           | 1.0                    | 364.8                                   |
|                   |                                    |                             |                    |               |               | 100           | 0.5                    | 30.4                                    |
|                   |                                    |                             |                    |               |               | 100           | 0.25                   | 0.1 or lower                            |
|                   |                                    |                             |                    |               |               | 100           | 0.125                  | 0.1 or lower                            |
|                   |                                    |                             |                    |               |               | 50            | 1.0                    | 62.4                                    |
|                   |                                    |                             |                    |               |               | 50            | 0.5                    | 0.8                                     |
|                   |                                    |                             |                    |               |               | 50            | 0.25                   | 0.1 or lower                            |
|                   |                                    |                             |                    |               |               | 50            | 0.125                  | 0.1 or lower                            |

**Note** The error when considering the signal source impedance is “sampling error + overall error”.

- Remarks**
1. These values are reference values calculated by simulating what happens to  $C_2$  voltage by  $R_i$  and  $C_0$  when  $V_{Ain}$  is applied from 0 V to 5 V at the same time as sampling start.
  2.  $m = 0$  to 5 when  $n = 0$   
 $m = 0$  to 7 when  $n = 1$
  3.  $f_{xx}$ : System clock frequency  
 $f_{AD01}$ : Basic clock frequency

<R>

## 12.6 Cautions

### 12.6.1 Stopping conversion operation

The ongoing conversion operation is stopped when 0 is written to the ADnSCM.ADnCE bit. At this time, the conversion result in the A/Dn conversion result register m (ADnCRm) and A/Dn conversion result extension register a (ADnECRa) is undefined. Therefore, read the A/D conversion result after A/D conversion has been completed (after the A/Dn conversion end interrupt request signal (INTADn) has been issued), and then write 0 to the ADnCE bit as necessary.

Note that the ADnCE bit is not cleared to 0 in all the modes even after the INTADn signal is generated.

**Remark** n = 0, 1  
m = 0 to 15

### 12.6.2 Interval of trigger during conversion operation in hardware trigger mode, conversion channel specification mode, and extension buffer mode

Inputting a trigger during conversion operation is ignored in the hardware trigger mode, conversion channel specification mode, and extension buffer mode. Therefore, the interval of the trigger (input time) in the hardware trigger mode, conversion channel specification mode, and extension buffer mode must be longer than the A/D conversion time specified by the ADnCTC.ADnFR3 to ADnCTC.ADnFR0 bits (see **Table 12-2 Number of A/D Conversion Clocks and A/D Conversion Time**).

**Remark** n = 0, 1

### 12.6.3 Writing to ADnSCM register

#### (1) Restarting A/D conversion

To restart A/D conversion, write the same value to the ADnSCM register. To change the ADnPLM, ADnTRG1, and ADnTRG0 bits, be sure to set the ADnCE bit to 0.

#### (2) Contention between end of A/D conversion and writing to ADnSCM register

If completion of A/D conversion contends with writing to the ADnSCM register during A/D conversion operation, the conversion result is correctly stored in the ADnCRm and ADnECRa registers, if the A/Dn conversion end interrupt request signal (INTADn) is generated. If the INTADn signal is not generated, the A/D conversion operation is aborted. Therefore, the previous conversion result is held by the ADnCRm and ADnECRa registers.

#### (3) Successive writing to ADnSCM register

To successively write the ADnSCM register when the conversion operation is enabled (ADnCE bit = 1), be sure to wait for time of at least 5 basic clocks ( $f_{AD01}$ ).

The ADnSCM register can be successively written when the ADnCE bit is set to 1 after the ADnSCM register is written while the ADnCE bit = 0.

**Remark** n = 0, 1

### 12.6.4 A/D conversion start timing

In the conversion channel specification mode and extension buffer mode, starting A/D conversion is delayed up to 1.5 basic clocks ( $f_{AD01}$ ) as compared with the normal operation mode.

### 12.6.5 Operation in standby mode

#### (1) HALT mode

The A/D conversion operation continues. If the HALT mode is released by a maskable interrupt request signal that is not masked, the values of the ADnSCM, ADnCRm, and ADnECRa registers are held.

#### (2) IDLE mode and STOP mode

No conversion operation is performed because clock supply to A/D converters 0 and 1 is stopped.

Be sure to set the ADnSCM.ADnCE bit to 0 when the IDLE or STOP mode is set. At this time, setting the A/D power save mode (ADnSCM.ADnPS bit = 0) is recommended.

**Remark** n = 0, 1  
m = 0 to 15

### 12.6.6 Timing of accepting trigger in conversion channel specification mode and extension buffer mode

In the conversion channel specification mode and extension buffer mode, selection trigger 1 or 2 is ignored, even if it is generated again, until the A/Dn conversion end interrupt signal (INTADn) is generated after A/D conversion is started by the first generation of selection trigger 1 or 2. In the extension buffer mode, the error flag is set to 1 in accordance with a specified error condition if selection trigger 1 or 2, or selection load trigger 1 or 2 is generated during this period (except, however, the case in **Caution 2** in **12.4.11 (1) Error detection function**).

**Remark** n = 0, 1

### 12.6.7 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 by averaging the A/D conversion results.

### 12.6.8 A/D conversion result hysteresis characteristics

Successive comparison type A/D converters hold an analog input voltage in an internal sample & hold capacitor and then perform 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 if the output impedance from the analog input source is too high.

- 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. Even if the conversion were to be performed at the same potential, the results may thus 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. Even if the conversion were to be performed at the same potential, the results may thus vary.

To obtain more accurate conversion results, lower the output impedance from the analog input source or execute A/D conversion twice consecutively on the same channel, and discard the first conversion result.

**12.6.9 A/D conversion trigger interval for continuous conversion**

For the A/D conversion trigger interval for continuous conversion, secure at least the minimum trigger interval shown below before inputting the next trigger. Otherwise, the trigger will be invalid (not retained).

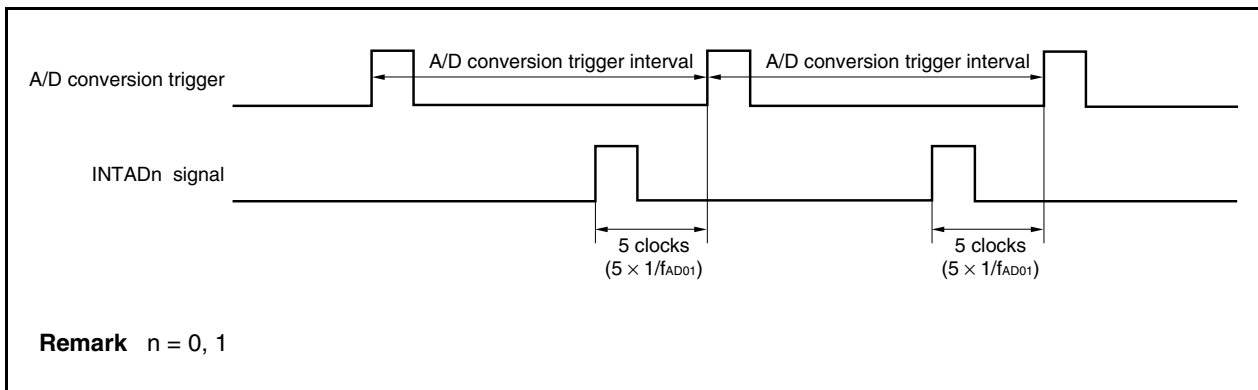
Minimum trigger interval clock count = A/D conversion clock count + 5 clocks

Minimum trigger interval time = Minimum trigger interval clock count  $\times$   $1/f_{AD01}$

**Example**  $f_{AD01} = 16$  MHz, A/D conversion time =  $2 \mu\text{s}$ , A/D conversion clock count = 32 clocks

Minimum trigger interval clock count =  $32 + 5 = 37$

Minimum trigger interval timer =  $37 \times 1/16 = 2.3125 [\mu\text{s}]$



## 12.7 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

### (1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1 LSB (Least Significant Bit). The percentage of 1 LSB with respect to the full scale is expressed by %FSR (Full Scale Range). %FSR indicates the ratio of analog input voltage that can be converted as a percentage, and is always represented by the following formula regardless of the resolution.

$$\begin{aligned} 1\%FSR &= (\text{Max. value of analog input voltage that can be converted} - \text{Min. value of analog input voltage that can be converted})/100 \\ &= (AV_{REFPn} - 0)/100 \\ &= AV_{REFPn}/100 \end{aligned}$$

1 LSB is as follows when the resolution is 12 bits.

$$\begin{aligned} 1 \text{ LSB} &= 1/2^{12} = 1/4,096 \\ &= 0.024\%FSR \end{aligned}$$

Accuracy has no relation to resolution, but is determined by overall error.

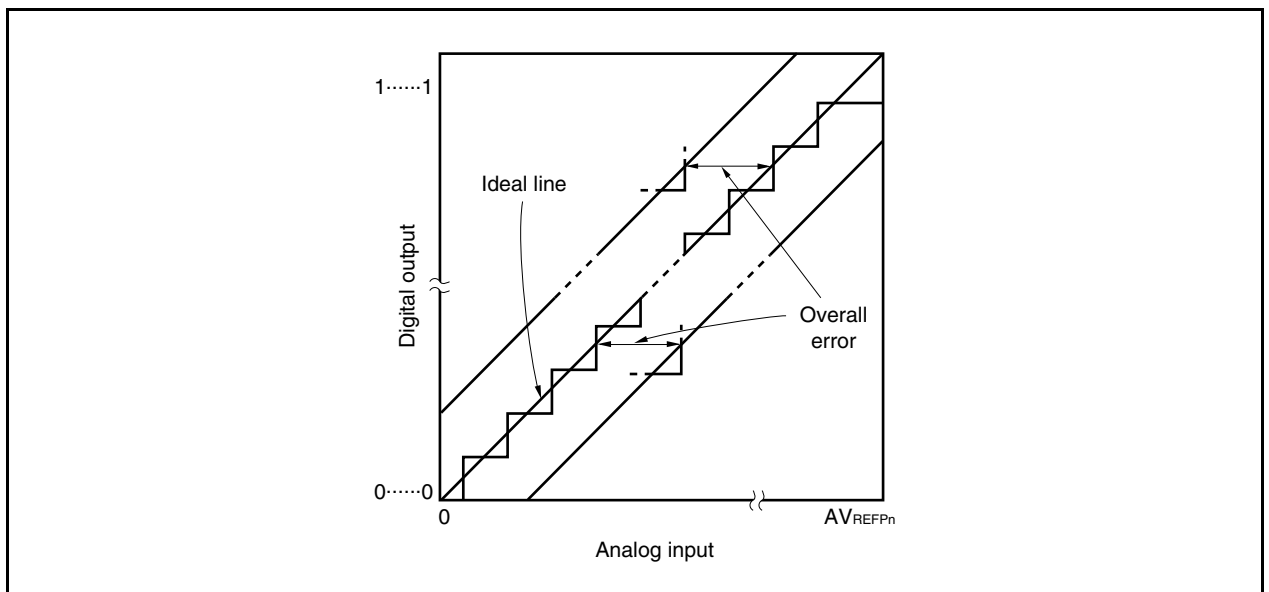
### (2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.

Zero-scale error, full-scale error, linearity error and errors that are combinations of these express the overall error.

Note that the quantization error is not included in the overall error in the characteristics table.

Figure 12-23. Overall Error

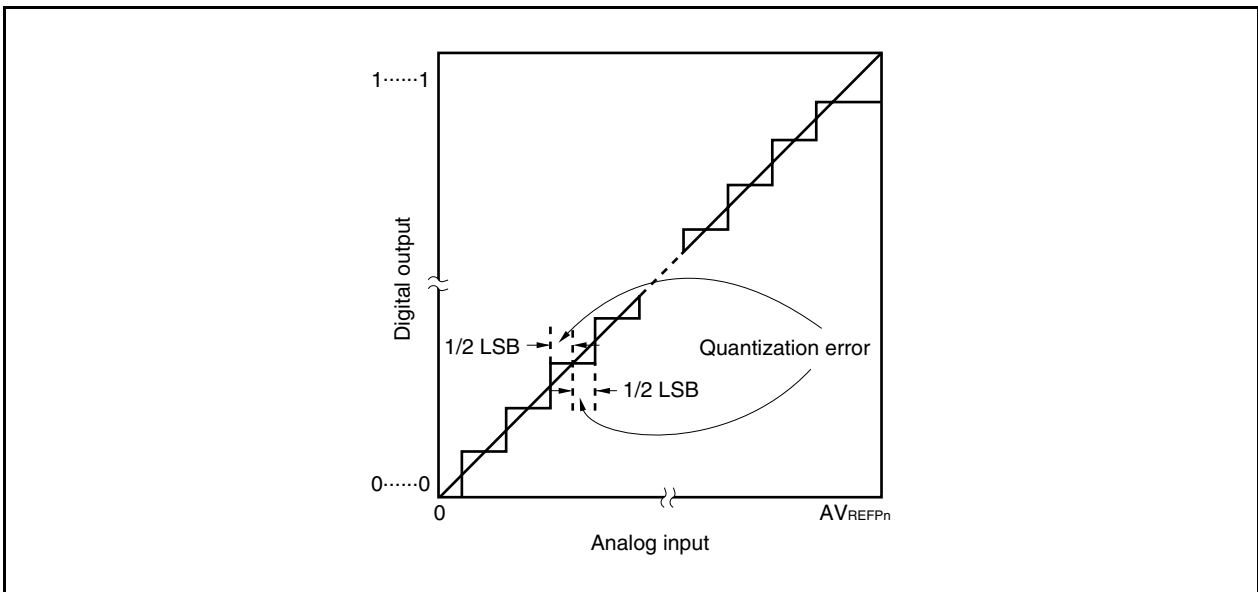


**(3) Quantization error**

When analog values are converted to digital values, a  $\pm 1/2$  LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of  $\pm 1/2$  LSB is converted to the same digital code, so a quantization error cannot be avoided.

Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

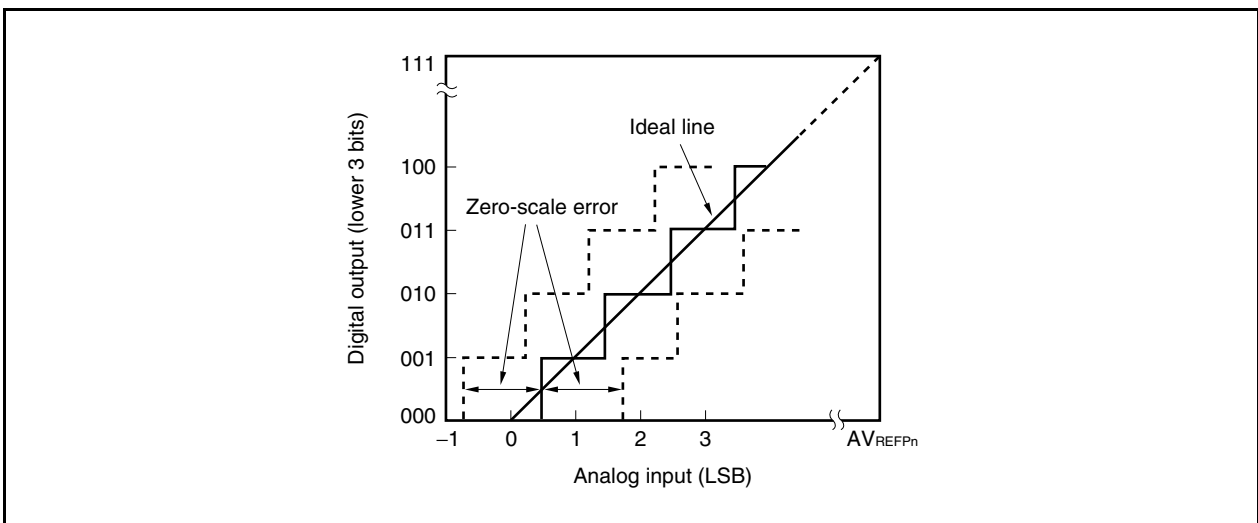
**Figure 12-24. Quantization Error**



**(4) Zero-scale error**

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value ( $1/2$  LSB) when the digital output changes from 0...000 to 0...001.

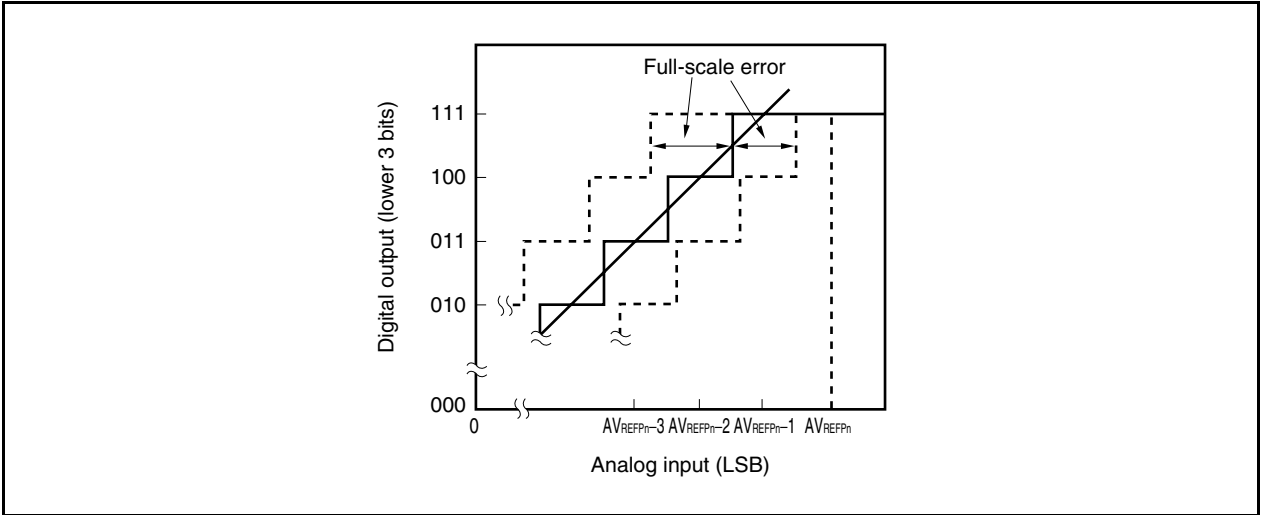
**Figure 12-25. Zero-Scale Error**



**(5) Full-scale error**

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (full-scale value – 3/2 LSB) when the digital output changes from 1.....110 to 1.....111.

**Figure 12-26. Full-Scale Error**

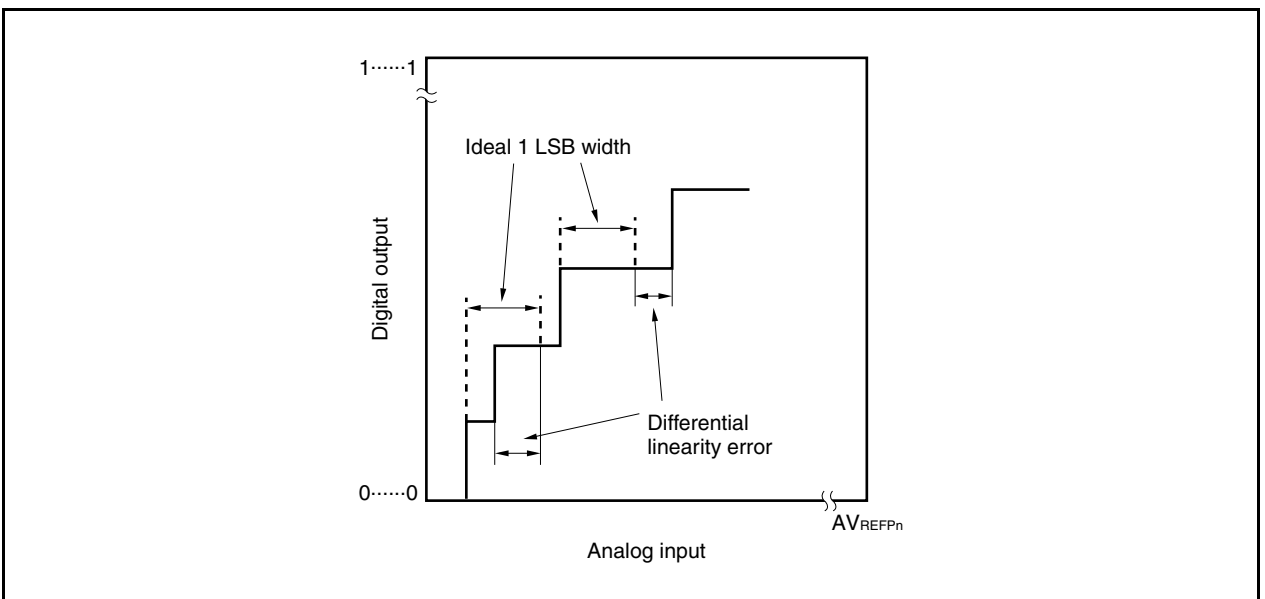


**(6) Differential linearity error**

While the ideal width of code output is 1 LSB, this indicates the difference between the actual measurement value and the ideal value.

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  $AV_{SSn}$  to  $AV_{REFPn}$ . See **12.7 (2) Overall error** for when the input voltage is increased or decreased, or when two or more channels are used.

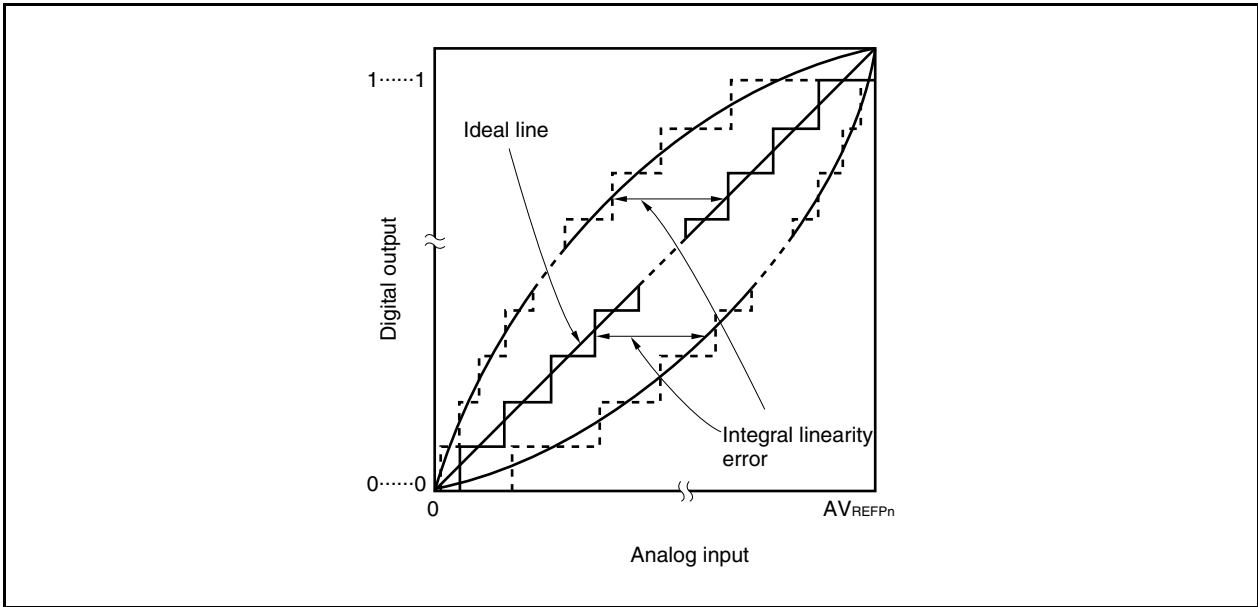
**Figure 12-27. Differential Linearity Error**



**(7) Integral linearity error**

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

**Figure 12-28. Integral Linearity Error**



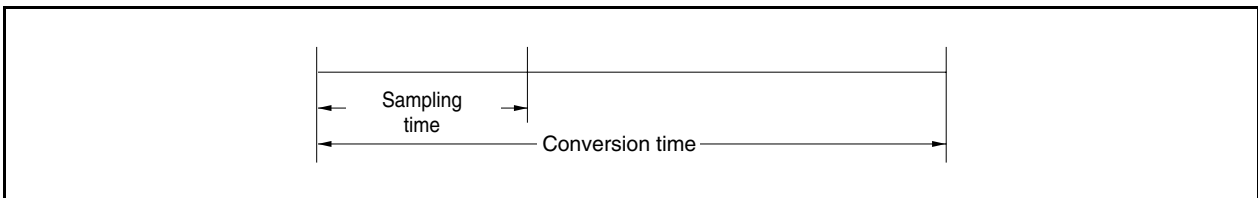
**(8) Conversion time**

This expresses the time from when the trigger is generated to when the digital output is obtained. The sampling time is included in the conversion time in the characteristics table.

**(9) Sampling time**

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.

**Figure 12-29. Sampling Time**





## CHAPTER 13 A/D CONVERTER 2

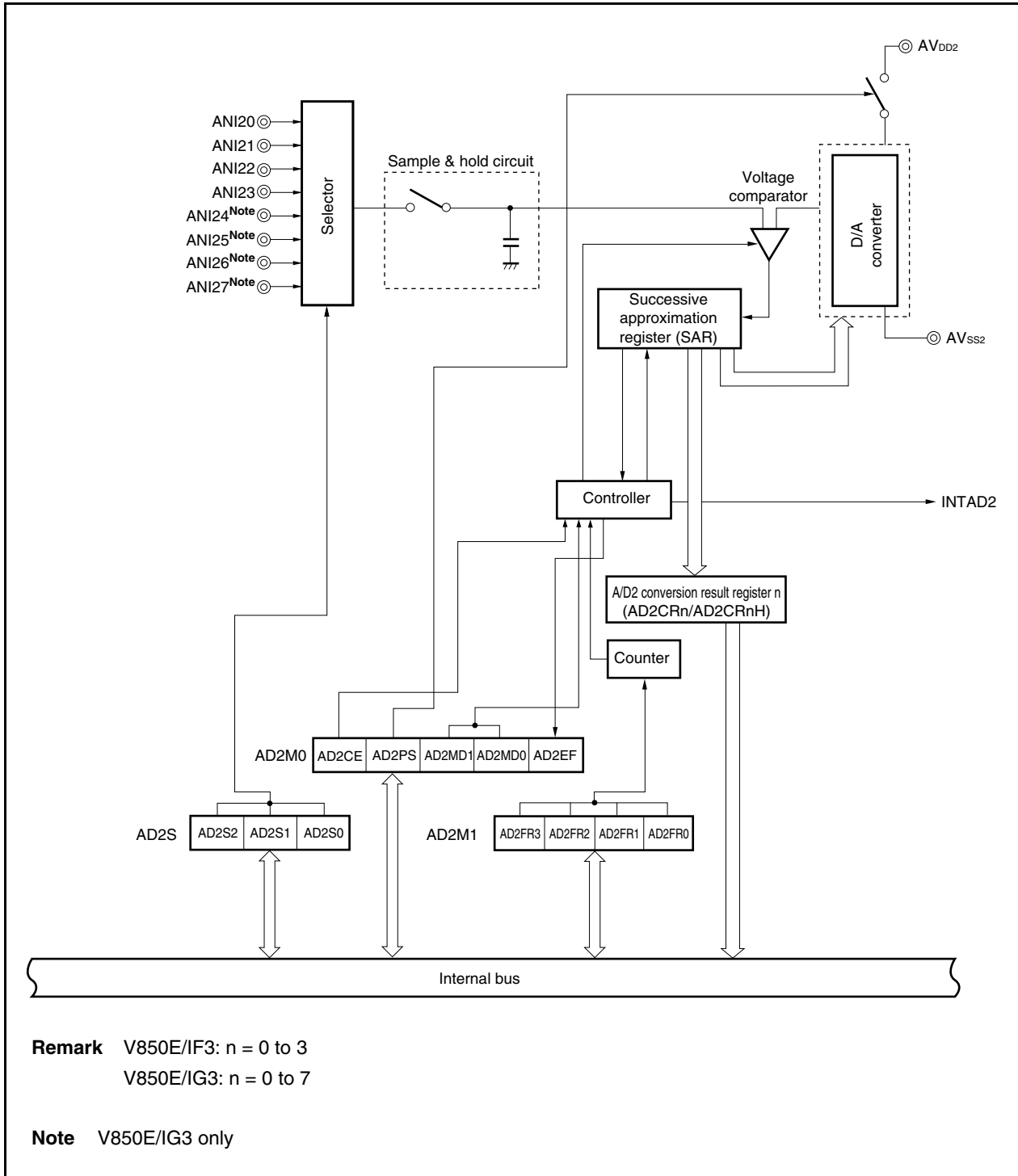
### 13.1 Features

- On-chip 10-bit resolution A/D converter
- Analog input
  - V850E/IF3: ANI20 to ANI23 (4 channels)
  - V850E/IG3: ANI20 to ANI27 (8 channels)
- A/D conversion result register
  - V850E/IF3: AD2CR0 to AD2CR3 (10 bits × 4)
  - V850E/IG3: AD2CR0 to AD2CR7 (10 bits × 8)
- A/D conversion trigger mode
  - Software trigger mode
- A/D conversion operation mode
  - Continuous select mode
  - Continuous scan mode
  - One-shot select mode
  - One-shot scan mode
- Successive comparison approximation method
- Operating voltage:  $EV_{DD0} = EV_{DD1} = EV_{DD2}$  (V850E/IG3 only) =  $AV_{DD2} = 4.0$  to  $5.5$  V

### 13.2 Configuration

The block diagram is shown below.

Figure 13-1. Block Diagram of A/D Converter 2



**Cautions 1.** If there is noise at the analog input pin (ANI2n) and at the A/D converter power supply voltage pin (AV<sub>DD2</sub>), 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 successively 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.
2. Do not apply a voltage outside the AV<sub>SS2</sub> to AV<sub>DD2</sub> range to the pins that are used as input pins of A/D converter 2.

A/D converter 2 consists of the following hardware.

**Table 13-1. Configuration of A/D Converter 2**

| Item              | Configuration  |
|-------------------|--|
| Analog input      | V850E/IF3: ANI20 to ANI23 (4 channels)<br>V850E/IG3: ANI20 to ANI27 (8 channels)   |
| Registers         | Successive approximation register (SAR)<br>V850E/IF3:<br>A/D2 conversion result registers 0 to 3 (AD2CR0 to AD2CR3)<br>A/D2 conversion result registers 0H to 3H (AD2CR0H to AD2CR3H): Only the higher 8 bits can be read V850E/IG3:<br>A/D2 conversion result registers 0 to 7 (AD2CR0 to AD2CR7)<br>A/D2 conversion result registers 0H to 7H (AD2CR0H to AD2CR7H): Only the higher 8 bits can be read |
| Control registers | A/D converter 2 mode registers 0, 1 (AD2M0, AD2M1)<br>A/D converter 2 channel specification register (AD2S)  |

**(1) Successive approximation register (SAR)**

The SAR register is a register that compares the voltage value of an analog input pin with the value of the voltage tap of the D/A converter and holds the result, starting from the most significant bit (MSB).

If data is held in the SAR all the way to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register are transferred in AD2CRn register.

When all the specified A/D conversion operations have ended, an A/D2 conversion end interrupt request signal (INTAD2) is generated.

**(2) A/D conversion result register n (AD2CRn), A/D conversion result register nH (AD2CRnH)**

The AD2CRn register is a register that holds the A/D conversion results. The conversion result is stored in the higher 10 bits of the AD2CRn register corresponding to the analog input. The lower 6 bits of these registers are always 0 when read.

The higher 8 bits of the result of A/D conversion are read from the AD2CRn register.

To read the result of A/D conversion in 16-bit units, specify the AD2CRn register. To read the higher 8 bits, specify the AD2CRnH register.

**Caution** The contents of the AD2CRn register may become undefined depending on the operation to write the AD2M0, AD2M1, and AD2S registers. Read the result of conversion from the AD2CRn register after conversion and before writing the AD2M0, AD2M1, and AD2S registers. The correct conversion result cannot be read from the AD2CRn register if any other procedure is used.

**(3) Sample & hold circuit**

The sample & hold circuit samples the analog input signals selected by the input circuit and sends the sampled data to the voltage comparator. This circuit holds the sampled analog input voltage during A/D conversion.

**(4) Voltage comparator**

The voltage comparator compares the value that is sampled and held with the voltage generated from the voltage tap of the D/A converter.

**(5) D/A converter**

The D/A converter is connected between  $AV_{DD2}$  and  $AV_{SS2}$  and generates a voltage to be compared with an input analog signal.

**(6) ANI2n pin**

The ANI2n pin is an analog input pin for A/D converter 2. This pin inputs the analog signals to be A/D converted. Pins other than the one that is selected by the AD2S register as analog signal input pins can be used as input port pins.

- Cautions**
- 1. Make sure that the voltages input to the ANI2n pin do not exceed the rated values. If a voltage higher than or equal to  $AV_{DD2}$  or lower than or equal to  $AV_{SS2}$  is input to a channel, the conversion value of the channel is undefined, and the conversion values of the other channels may also be affected.**
  - 2. The analog input pin (ANI2n) is alternately used as input port pin (P7n). If an instruction to input a signal to port 7 is executed during conversion when one of ANI2n is selected for A/D conversion, the resolution for conversion may drop.**

**(7)  $AV_{DD2}$  pin**

The  $AV_{DD2}$  pin alternately functions as the pin for inputting the positive power supply and reference voltage of A/D converter 2. This pin converts signals input to the ANI2n pin to digital signals based on the voltage applied between  $AV_{DD2}$  and  $AV_{SS2}$ .

Always make the potential at this pin the same as that at the  $EV_{DD0}$ ,  $EV_{DD1}$ , and  $EV_{DD2}$  pins (V850E/IG3 only) even when A/D converter 2 is not used.

The operating voltage range of the  $AV_{DD2}$  pin is  $EV_{DD0} = EV_{DD1} = EV_{DD2}$  (V850E/IG3 only) =  $AV_{DD2} = 4.0$  to  $5.5$  V.

**(8)  $AV_{SS2}$  pin**

This is the ground pin of A/D converter 2. Always make the potential at this pin the same as that at the  $EV_{SS0}$ ,  $EV_{SS1}$ , and  $EV_{SS2}$  (V850E/IG3 only) pins even when A/D converter 2 is not used.

**Remark** V850E/IF3: n = 0 to 3  
V850E/IG3: n = 0 to 7

**13.3 Control Registers**

A/D converter 2 is controlled by the following registers.

- A/D converter 2 mode registers 0, 1 (AD2M0 to AD2M1)
- A/D converter 2 channel specification register (AD2S)

The following registers are also used.

- A/D2 conversion result register n (AD2CRn)
- A/D2 conversion result register nH (AD2CRnH)

**(1) A/D converter 2 mode register 0 (AD2M0)**

The AD2M0 register is a 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, bit 0 is read-only. Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFFB80H

|       |       |       |        |        |   |   |   |       |
|-------|-------|-------|--------|--------|---|---|---|-------|
|       | <7>   | 6     | 5      | 4      | 3 | 2 | 1 | 0     |
| AD2M0 | AD2CE | AD2PS | AD2MD1 | AD2MD0 | 0 | 0 | 0 | AD2EF |

|       |                                     |
|-------|-------------------------------------|
| AD2CE | Control of A/D conversion operation |
| 0     | Conversion operation stopped        |
| 1     | Conversion operation enabled        |

|       |                        |
|-------|------------------------|
| AD2PS | A/D conversion control |
| 0     | A/D power off          |
| 1     | A/D power on           |

- The first result of conversion by the A/D converter 2 becomes valid when the AD2CE bit is set to 1 (when conversion is enabled) at least 2  $\mu$ s after the AD2PS bit is set to 1 (A/D power is turned on).  
If the AD2CE bit is set to 1 before 2  $\mu$ s pass, the conversion operation is started and ends after the A/D conversion time, but the conversion result is undefined.
- When the A/D converter 2 is not used, set to 0 the AD2CE bit (stop conversion operation) and the AD2PS bit (turn off A/D power) to reduce the power consumption.
- Do not set the AD2PS2 bit during A/D conversion operation (AD2EF bit = 1).  
While the A/D conversion operation is not performed, the AD2CE and AD2PS bits can be simultaneously cleared to 0.

|        |        |                                 |
|--------|--------|---------------------------------|
| AD2MD1 | AD2MD0 | Specification of operation mode |
| 0      | 0      | Successive select mode          |
| 0      | 1      | Successive scan mode            |
| 1      | 0      | One-shot select mode            |
| 1      | 1      | One-shot scan mode              |

|       |                                    |
|-------|------------------------------------|
| AD2EF | Status of A/D converter 2 (status) |
| 0     | During A/D conversion stop         |
| 1     | During A/D conversion operation    |

- Cautions**
1. Writing to bit 0 is ignored.
  2. The conversion resolution of the pin to which an analog signal is input first immediately after A/D conversion is started may drop. For details, see 13.7 (6) About AV<sub>DD2</sub> pin.
  3. A/D conversion is stopped and started again from the beginning if the AD2M0 and AD2S registers are written during A/D conversion operation (AD2EF bit = 1).
  4. Be sure to set bits 1 to 3 to "0".

**(2) A/D converter 2 mode register 1 (AD2M1)**

The AD2M1 register is a register that specifies the number of A/D conversion clocks and A/D conversion time. 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: FFFFFB81H

|       |   |   |   |   |        |        |        |        |
|-------|---|---|---|---|--------|--------|--------|--------|
|       | 7 | 6 | 5 | 4 | 3      | 2      | 1      | 0      |
| AD2M1 | 0 | 0 | 0 | 0 | AD2FR3 | AD2FR2 | AD2FR1 | AD2FR0 |

**Cautions**

1. See Table 13-2 Setting Example During Conversion Mode for the AD2FR3 to AD2FR0 bits.
2. Changing the AD2FR3 to AD2FR0 bits is prohibited during conversion operation (AD2CE bit = 1).
3. Be sure to set bits 4 to 7 to “0”.

**Table 13-2. Setting Example During Conversion Mode**

| AD2FR3           | AD2FR2 | AD2FR1 | AD2FR0 | Number of A/D Conversion Clocks <sup>Note</sup> | A/D Conversion Time  | f <sub>AD2</sub> = 32 MHz<br>(f <sub>XX</sub> = 64 MHz) | f <sub>AD2</sub> = 24 MHz<br>(f <sub>XX</sub> = 48 MHz) |
|------------------|--------|--------|--------|---|----------------------|---|---|
| 0                | 0      | 0      | 1      | 62  | 62/f <sub>AD2</sub>  | Setting prohibited                                      | Setting prohibited                                      |
| 0                | 0      | 1      | 0      | 93  | 93/f <sub>AD2</sub>  | Setting prohibited                                      | 3.86 μs   |
| 0                | 0      | 1      | 1      | 124   | 124/f <sub>AD2</sub> | 3.88 μs   | 5.17 μs   |
| 0                | 1      | 0      | 0      | 155   | 155/f <sub>AD2</sub> | 4.84 μs   | 6.46 μs   |
| 0                | 1      | 0      | 1      | 186   | 186/f <sub>AD2</sub> | 5.81 μs   | 7.75 μs   |
| 0                | 1      | 1      | 0      | 217   | 217/f <sub>AD2</sub> | 6.78 μs   | 9.04 μs   |
| 0                | 1      | 1      | 1      | 248   | 248/f <sub>AD2</sub> | 7.75 μs   | Setting prohibited                                      |
| 1                | 0      | 0      | 0      | 279   | 279/f <sub>AD2</sub> | 8.72 μs   | Setting prohibited                                      |
| 1                | 0      | 0      | 1      | 310   | 310/f <sub>AD2</sub> | 9.69 μs   | Setting prohibited                                      |
| Other than above |        |        |        |   | Setting prohibited   |   |   |

**Note** The number of clocks (f<sub>AD2</sub>) from the start to the end of A/D conversion.

**Caution** Set the A/D conversion time in a range from 3.8 to 10 μs.

**Remark** f<sub>AD2</sub>: Operating clock of A/D converter 2

**(3) A/D converter 2 channel specification register (AD2S)**

The AD2S register is a register that specifies the analog input pin to be A/D-converted.

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: FFFF82H

|      |   |   |   |   |   |       |       |       |
|------|---|---|---|---|---|-------|-------|-------|
|      | 7 | 6 | 5 | 4 | 3 | 2     | 1     | 0     |
| AD2S | 0 | 0 | 0 | 0 | 0 | AD2S2 | AD2S1 | AD2S0 |

| AD2S2 | AD2S1 | AD2S0 | Select mode           | Scan mode                      |
|-------|-------|-------|-----------------------|--------------------------------|
| 0     | 0     | 0     | ANI20                 | ANI20                          |
| 0     | 0     | 1     | ANI21                 | ANI20, ANI21                   |
| 0     | 1     | 0     | ANI22                 | ANI20 to ANI22                 |
| 0     | 1     | 1     | ANI23                 | ANI20 to ANI23                 |
| 1     | 0     | 0     | ANI24 <sup>Note</sup> | ANI20 to ANI24 <sup>Note</sup> |
| 1     | 0     | 1     | ANI25 <sup>Note</sup> | ANI20 to ANI25 <sup>Note</sup> |
| 1     | 1     | 0     | ANI26 <sup>Note</sup> | ANI20 to ANI26 <sup>Note</sup> |
| 1     | 1     | 1     | ANI27 <sup>Note</sup> | ANI20 to ANI27 <sup>Note</sup> |

**Note** V850E/IG3 only.

With the V850E/IF3, this setting is not available because the necessary pins are not provided.

**Caution** Be sure to set bits 3 to 7 to “0”.



**(4) A/D2 conversion result registers n, nH (AD2CRn, AD2CRnH)**

The AD2CRn and AD2CRnH registers are registers that hold the A/D conversion results. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR) and stored in the higher 10 bits of the AD2CRn register. The lower 6 bits of these registers are always 0 when read.

The higher 8 bits of A/D conversion result are read to the AD2CRnH register.

These registers can only be read in 16-bit or 8-bit units. When the A/D conversion results are read in 16-bit units, the AD2CRn register is specified, and when the higher 8 bits are read, the AD2CRnH register is specified.

Reset sets AD2CRn register to 0000H and AD2CRnH register to 00H.

**Caution** If a write operation is performed on the AD2M0, AD2M1, and AD2S registers, the contents of the AD2CRn register may become undefined. Read the conversion result after the conversion operation and before performing a write operation on the AD2M0, AD2M1, and AD2S registers. The correct conversion result may not be read if the timing is other than the above.

After reset: 0000H R Address: AD2CR0 FFFFFFFB90H, AD2CR1 FFFFFFFB92H,  
AD2CR2 FFFFFFFB94H, AD2CR3 FFFFFFFB96H,  
AD2CR4 FFFFFFFB98H<sup>Note</sup>, AD2CR5 FFFFFFFB9AH<sup>Note</sup>,  
AD2CR6 FFFFFFFB9CH<sup>Note</sup>, AD2CR7 FFFFFFFB9EH<sup>Note</sup>

|                             |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |
|-----------------------------|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|
|                             | 15 | 14 | 13 | 12 | 11 | 10 | 9  | 8  | 7  | 6  | 5 | 4 | 3 | 2 | 1 | 0 |
| AD2CRn                      | AD | AD | AD | AD | AD | AD | AD | AD | AD | AD | 0 | 0 | 0 | 0 | 0 | 0 |
| { V850E/IF3<br>n = 0 to 3 } | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 |   |   |   |   |   |   |
| { V850E/IG3<br>n = 0 to 7 } |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |

**Note** V850E/IG3 only.

After reset: 00H R Address: AD2CR0H FFFFFFFB91H, AD2CR1H FFFFFFFB93H,  
AD2CR2H FFFFFFFB95H, AD2CR3H FFFFFFFB97H,  
AD2CR4H FFFFFFFB99H<sup>Note</sup>, AD2CR5H FFFFFFFB9BH<sup>Note</sup>,  
AD2CR6H FFFFFFFB9DH<sup>Note</sup>, AD2CR7H FFFFFFFB9FH<sup>Note</sup>

|                             |      |      |      |      |      |      |      |      |
|-----------------------------|------|------|------|------|------|------|------|------|
|                             | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| AD2CRnH                     | AD29 | AD28 | AD27 | AD26 | AD25 | AD24 | AD23 | AD22 |
| { V850E/IF3<br>n = 0 to 3 } |      |      |      |      |      |      |      |      |
| { V850E/IG3<br>n = 0 to 7 } |      |      |      |      |      |      |      |      |

**Note** V850E/IG3 only.

The correspondence between the analog input pins and the AD2CRn and AD2CRnH registers is shown below.

**Table 13-3. Correspondence Between Analog Input Pins and AD2CRn and AD2CRnH Registers**

| Analog Input Pin      | A/D Conversion Result Register                   |
|-----------------------|--|
| ANI20                 | AD2CR0, AD2CR0H                                  |
| ANI21                 | AD2CR1, AD2CR1H                                  |
| ANI22                 | AD2CR2, AD2CR2H                                  |
| ANI23                 | AD2CR3, AD2CR3H                                  |
| ANI24 <sup>Note</sup> | AD2CR4 <sup>Note</sup> , AD2CR4H <sup>Note</sup> |
| ANI25 <sup>Note</sup> | AD2CR5 <sup>Note</sup> , AD2CR5H <sup>Note</sup> |
| ANI26 <sup>Note</sup> | AD2CR6 <sup>Note</sup> , AD2CR6H <sup>Note</sup> |
| ANI27 <sup>Note</sup> | AD2CR7 <sup>Note</sup> , AD2CR7H <sup>Note</sup> |

**Note** V850E/IG3 only

The relationship between the analog voltage input to the analog input pin (ANI2n) and the A/D conversion result (of A/D2 conversion result register n (AD2CRn)) is as follows:

$$SAR = INT \left( \frac{V_{IN}}{AV_{DD2}} \times 1,024 + 0.5 \right)$$

$$ADCR^{Note} = SAR \times 64$$

or,

$$(SAR - 0.5) \times \frac{AV_{DD2}}{1,024} \leq V_{IN} < (SAR + 0.5) \times \frac{AV_{DD2}}{1,024}$$

INT(): Function that returns the integer of the value in ( )

V<sub>IN</sub>: Analog input voltage

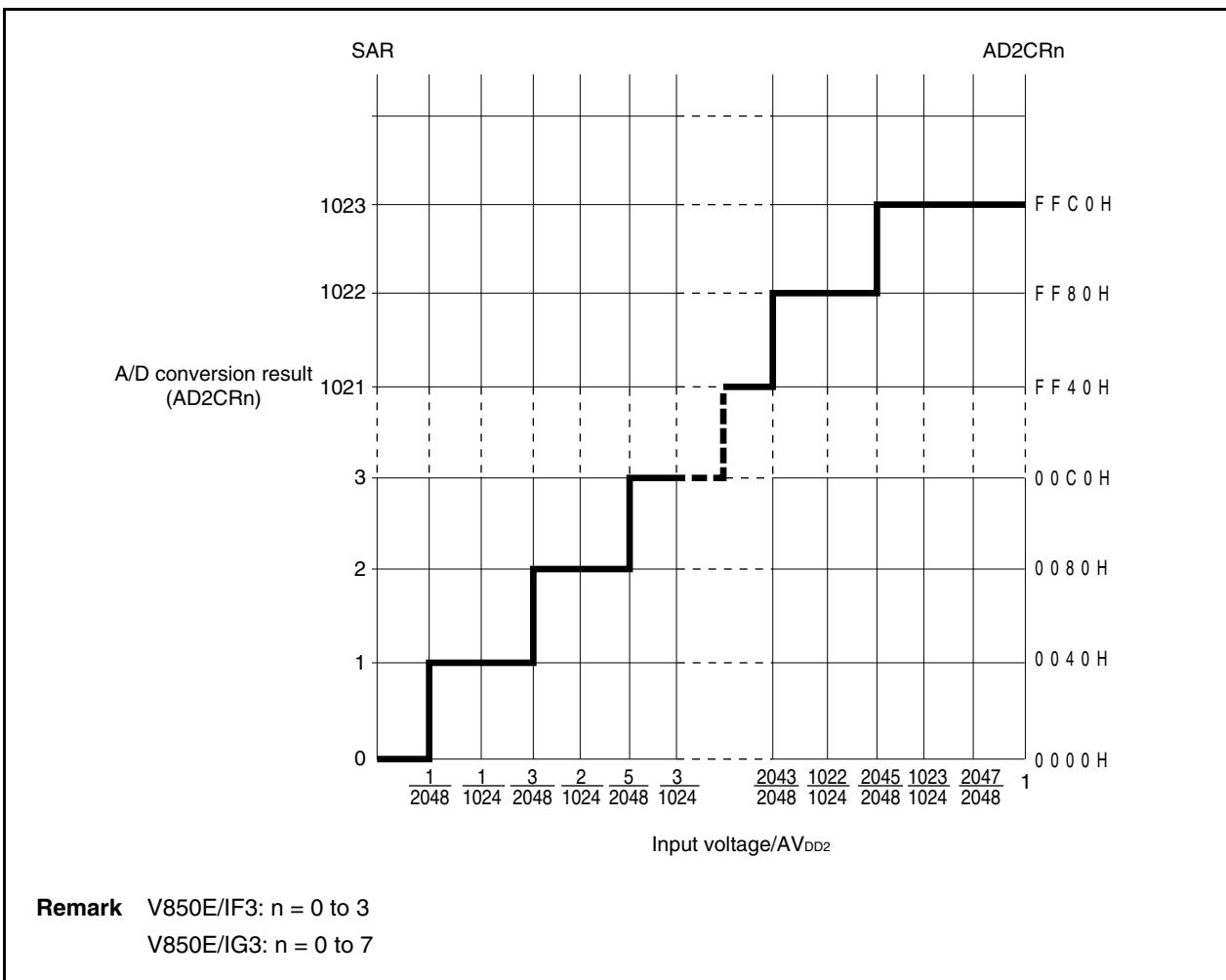
AV<sub>DD2</sub>: AV<sub>DD2</sub> pin voltage

ADCR: Value of A/D2 conversion result register n (AD2CRn)

**Note** The lower 6 bits of the AD2CRn register are fixed to 0.

The relationship between the analog input voltage and the A/D conversion results is shown in Figure 13-2.

**Figure 13-2. Relationship Between Analog Input Voltage and A/D Conversion Results**



## 13.4 Operation

### 13.4.1 Basic operation

- <1> Set the AD2M0.AD2PS bit to 1 to turn on A/D power while the AD2M0.AD2CE bit = 0.  
At this time, bits other than the AD2M0.AD2CE bit can be simultaneously set.
- <2> Select an operation mode of A/D conversion and A/D conversion time by using the AD2M0, AD2M1, and AD2S registers.
- <3> Setting the AD2M0.AD2CE bit to 1 (enable conversion) at least 2  $\mu$ s after turning on A/D power (AD2M0.AD2PS bit = 0  $\rightarrow$  1) starts A/D conversion.  
If the AD2CE bit is set to 1 before 2  $\mu$ s passes, the conversion operation is started and ends after A/D conversion time, but the conversion result is undefined.
- <4> When A/D conversion is started, the voltage input to the selected analog input channel is sampled by the sample & hold circuit.
- <5> When sampling has been performed for a specific time, the sample & hold circuit enters the hold status, and holds the input analog voltage until A/D conversion ends.
- <6> Set bit 9 of the successive approximation register (SAR) and changes the level of the voltage tap of the D/A converter to the reference voltage ( $1/2AV_{DD2}$ ).
- <7> The voltage generated by the voltage tap of the D/A converter is compared with the analog input voltage by a voltage comparator. If the analog input voltage is found to be greater than ( $1/2AV_{DD2}$ ) as a result of comparison, the MSB of the SAR register remains set. If the analog input voltage is less than ( $1/2AV_{DD2}$ ), the MSB is reset.
- <8> Next, bit 8 of the SAR register is automatically set, and the next comparison is started. The voltage tap of the D/A converter is selected according to the value of bit 9, to which the result has been already set as shown below.

Bit 9 = 1: ( $3/4AV_{DD2}$ )

Bit 9 = 0: ( $1/4AV_{DD2}$ )

The voltage tap of the D/A converter and the analog input voltage are compared and bit 8 of the SAR register is manipulated according to the result of the comparison as shown below.

Analog input voltage  $\geq$  Voltage tap of D/A converter: Bit 8 = 1

Analog input voltage  $\leq$  Voltage tap of D/A converter: Bit 8 = 0

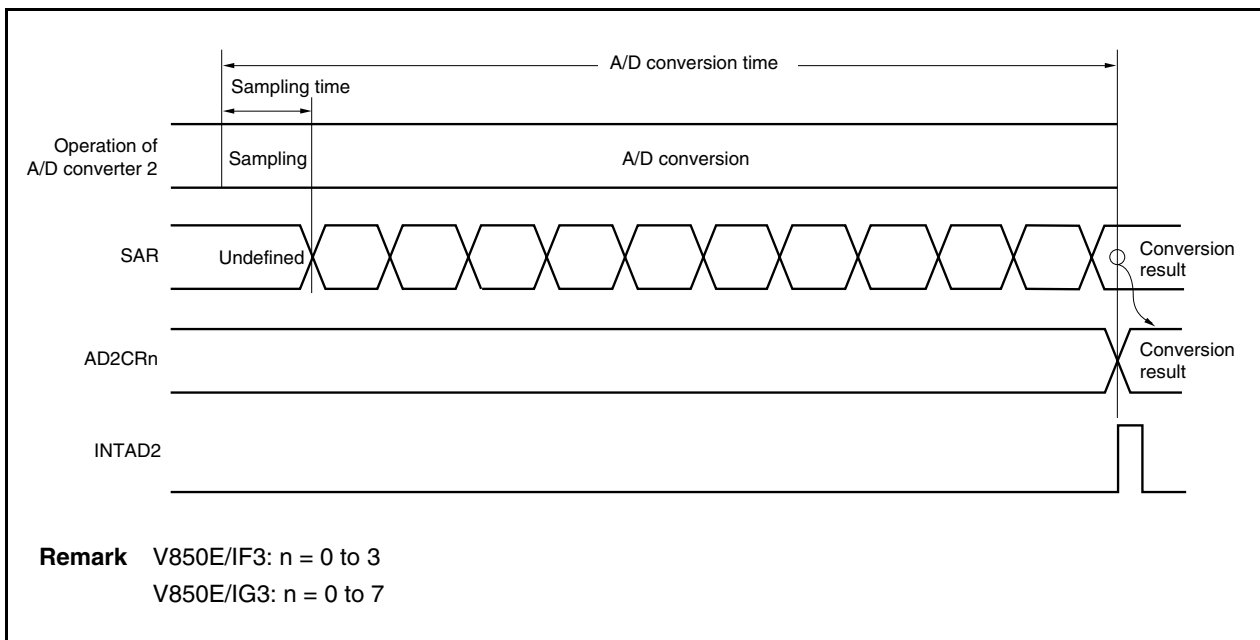
Comparison is continued like this to bit 0 of the SAR register.

<9> When comparison of 10 bits has been completed, the valid digital result remains in the SAR register. This value is transferred to the AD2CRn register and the conversion result is stored in this register (V850E/IF3: n = 0 to 3, V850E/IG3: n = 0 to 7). An A/D2 conversion end interrupt request signal (INTAD2) is generated simultaneously in the select mode and when all the specified A/D conversion operations are completed in the scan mode.

<10> In the continuous select mode or continuous scan mode, <4> to <9> are repeated unless the AD2CE bit is set to 0 after completion of A/D conversion.

In the one-shot select mode or one-shot scan mode, the conversion operation is stopped after it is completed (at this time, the AD2M0.AD2CE bit holds 1 and is not automatically cleared). Write 1 to the AD2CE bit to start conversion operation again.

Figure 13-3. Basic Operation of A/D Converter 2



### 13.4.2 Trigger mode

Trigger mode that serve as the start timing of an A/D conversion operation is software trigger mode. This mode is set by the AD2M0 register.

#### (1) Software trigger mode

In this mode, the analog input pin (ANI2n) specified by the AD2S.AD2S2 to AD2S.AD2S0 bits is used for the A/D conversion start timing by setting the AD2M0.AD2CE bit to 1.

After A/D conversion ends, the conversion result is stored in A/D2 conversion result register n (AD2CRn).

An A/D2 conversion end interrupt request signal (INTAD2) is generated simultaneously when A/D conversion operations are completed in the select mode. INTAD2 interrupt request signal is generated when all the specified A/D conversion operations are completed in the scan mode.

If the operation mode set by the AD2M0.AD2MD1 and AD2M0.AD2MD0 bits is the continuous select mode or continuous scan mode, the conversion operation is repeated unless the AD2M0.AD2CE bit is set to 0. In the one-shot select mode or one-shot scan mode, the conversion operation is stopped after A/D conversion ends.

The AD2M0.AD2EF bit is set to 1 (conversion in progress) when A/D conversion is started, and set to 0 (conversion stops) when it is completed.

If the AD2M0 and AD2S registers are written during A/D conversion, the conversion is stopped and executed again from the beginning.

**Remark** V850E/IF3: n = 0 to 3  
V850E/IG3: n = 0 to 7

### 13.4.3 Operation mode

There are four operation modes to which the ANI2n pin is set: continuous select mode, continuous scan mode, one-shot select mode, and one-shot scan mode. These modes are set by the AD2M0.AD2MD1 and AD2M0.AD2MD0 registers.

The relationship between the AD2M0, AD2M1, and AD2S registers and operation mode is shown below.

| Trigger Mode     | Operation Mode    | Set Value |           |           |
|------------------|-------------------|-----------|-----------|-----------|
|                  |                   | AD2M0     | AD2M1     | AD2S      |
| Software trigger | Continuous select | X100000XB | 0000XXXXB | 00000XXXB |
|                  | Continuous scan   | X101000XB | 0000XXXXB | 00000XXXB |
|                  | One-shot select   | X110000XB | 0000XXXXB | 00000XXXB |
|                  | One-shot scan     | X111000XB | 0000XXXXB | 00000XXXB |

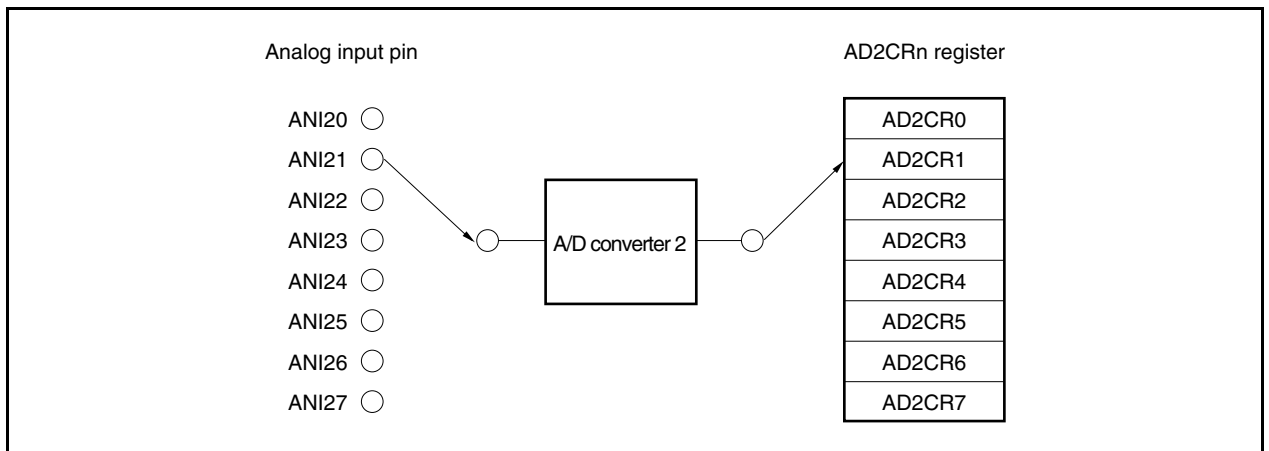
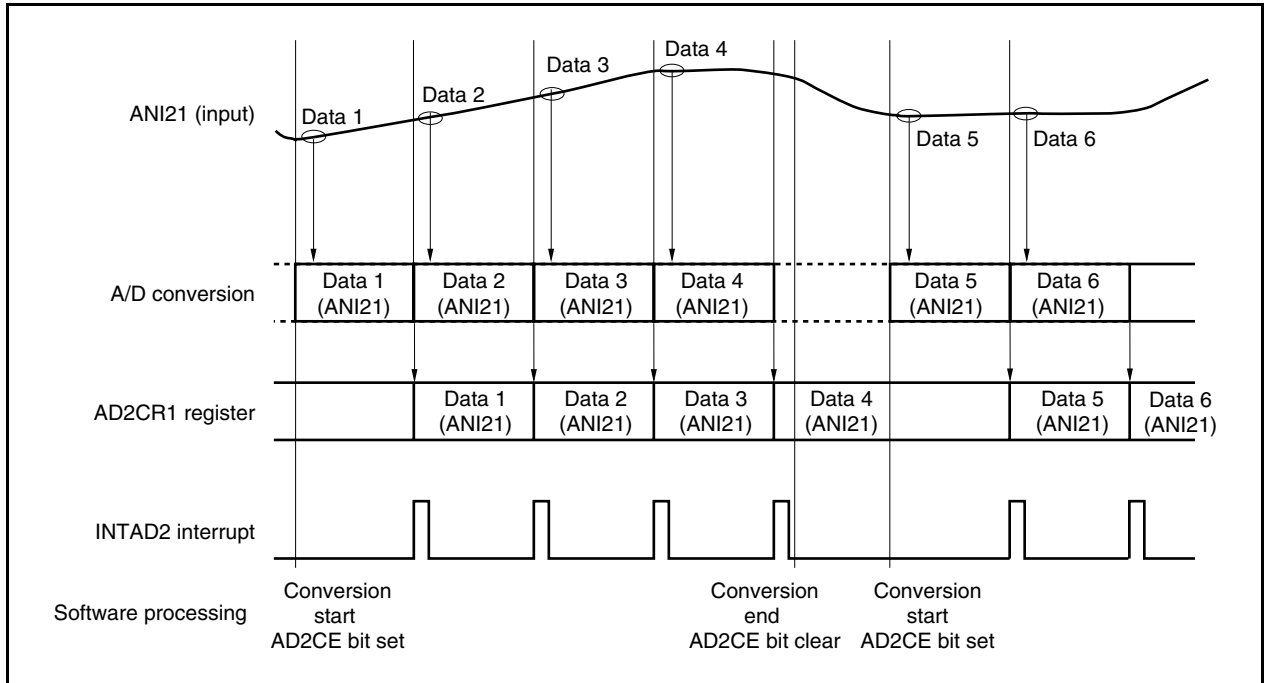
#### (1) Continuous select mode

In this mode, the analog input pin (ANI2n) specified by the AD2S register is A/D-converted continuously. The conversion results are stored in the AD2CRn register corresponding to the ANI2n pin. The ANI2n pin and the AD2CRn register correspond one to one, and an A/D2 conversion end interrupt request signal (INTAD2) is generated each time one A/D conversion ends.

After A/D conversion ends, the conversion is repeated again unless the AD2M0.AD2CE bit is set to 0.

**Remark** V850E/IF3: n = 0 to 3  
V850E/IG3: n = 0 to 7

**Figure 13-4. Continuous Select Mode Operation Timing**  
 (When AD2M0.AD2MD1 and AD2M0.AD2MD0 Bits = 00, AD2S.AD2S2 to AD2S.AD2S0 Bits = 001): V850E/IG3



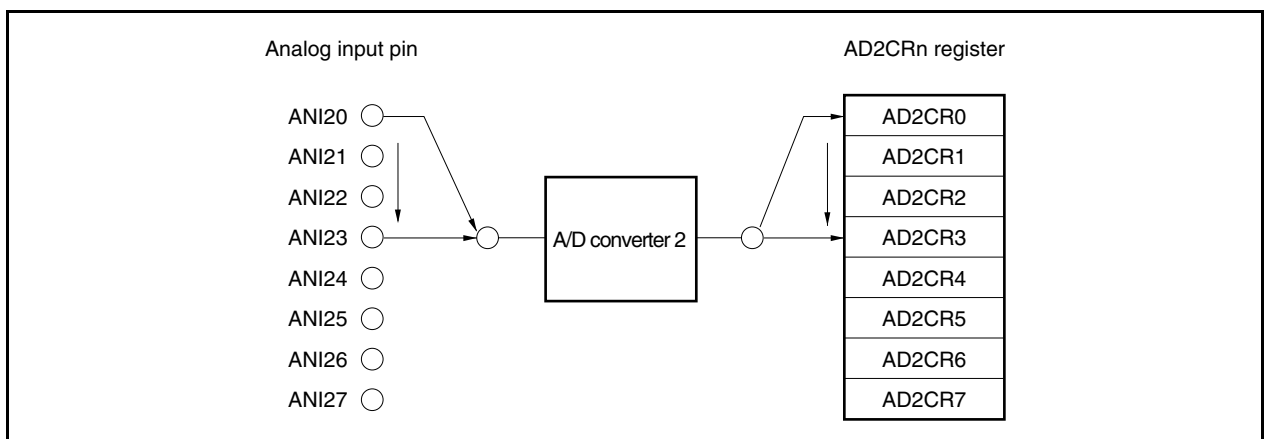
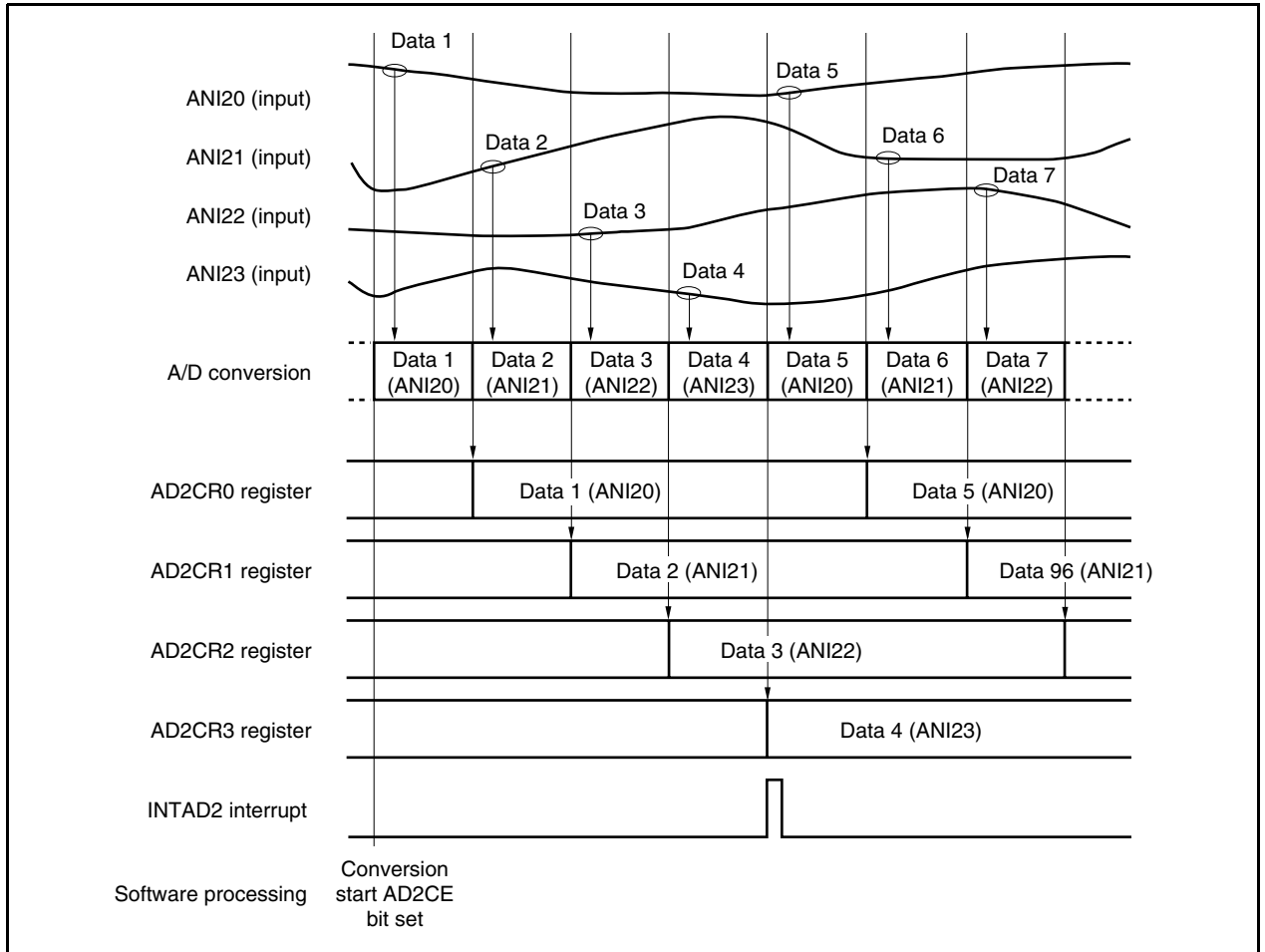
**(2) Continuous scan mode**

In this mode, the analog input pin (ANI2n) specified by the AD2S register is selected sequentially from the ANI20 pin, and A/D conversion is executed continuously. The A/D conversion results are stored in the AD2CRn register corresponding to the analog input pin. When conversion of all the specified analog input pin ends, the A/D2 conversion end interrupt request signal (INTAD2) is generated. After A/D conversion ends, the conversion is started again from the ANI20 pin, unless the AD2M0.AD2CE bit is set to 0.

**Remark** V850E/IF3: n = 0 to 3  
V850E/IG3: n = 0 to 7



**Figure 13-5. Continuous Scan Mode Operation Timing**  
 (When AD2M0.AD2MD1 and AD2M0.AD2MD0 Bits = 01, AD2S.AD2S2 to AD2S.AD2S0 Bits = 011): V850E/IG3



**(3) One-shot select mode**

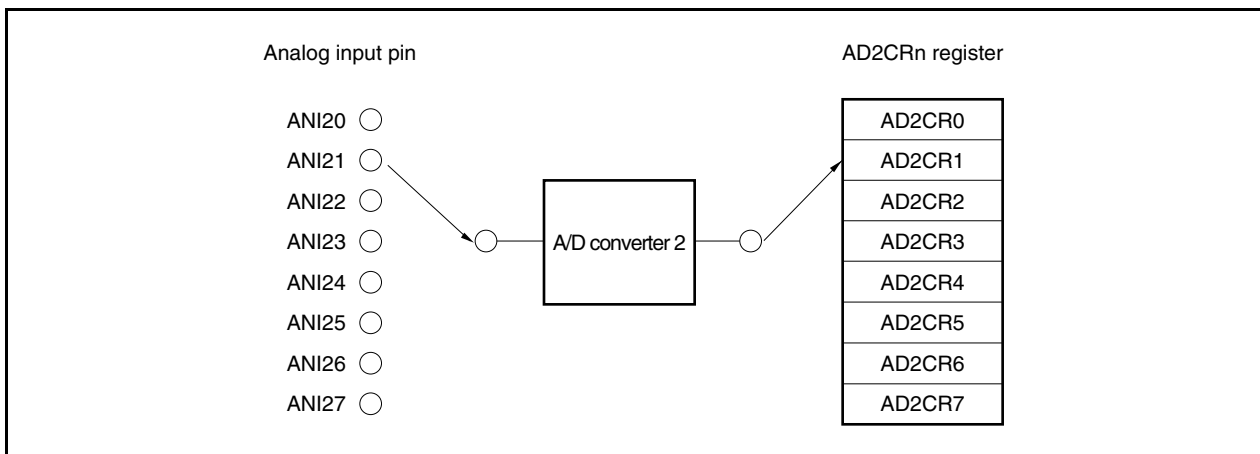
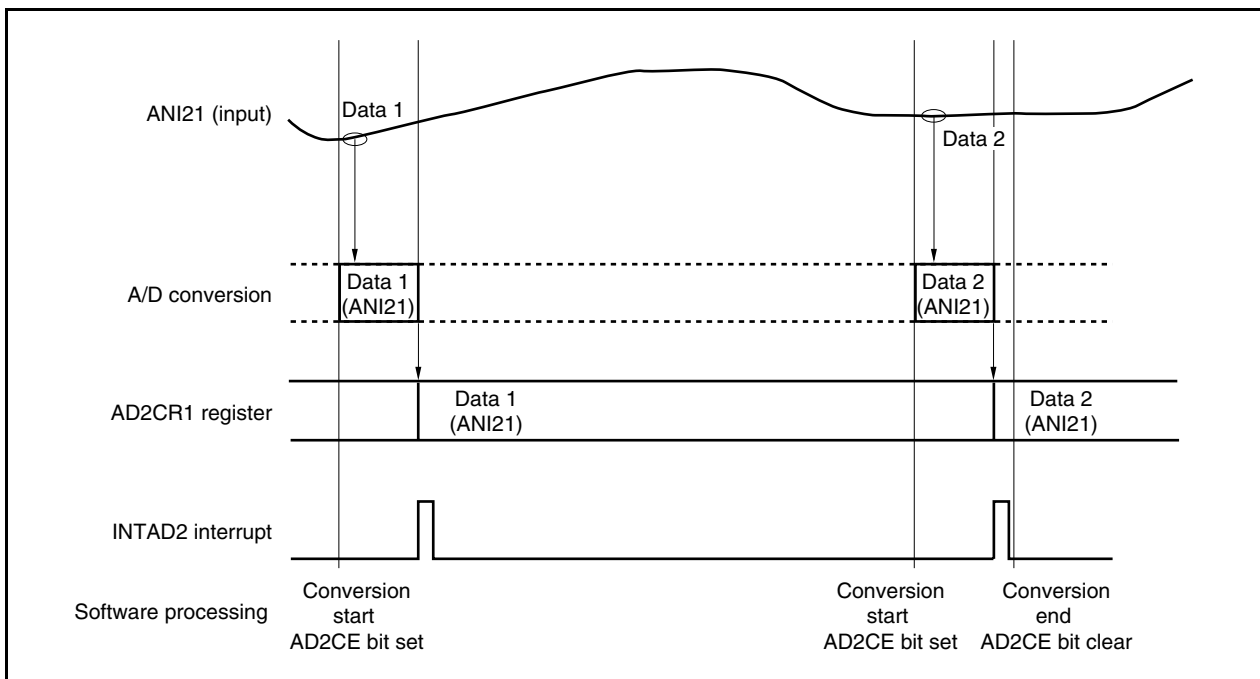
In this mode, the analog input pin (ANI2n) specified by the AD2S register is A/D-converted once. The conversion result is stored in the AD2CRn register corresponding to the ANI2n pin. The ANI2n pin and the AD2CRn register correspond one to one, and an A/D2 conversion end interrupt request signal (INTAD2) is generated each time one A/D conversion ends.

After A/D conversion ends, the conversion operation is stopped.

**Remark** V850E/IF3: n = 0 to 3  
 V850E/IG3: n = 0 to 7

**Figure 13-6. One-Shot Select Mode Operation Timing**

(When AD2M0.AD2MD1 and AD2M0.AD2MD0 Bits = 10, AD2S.AD2S2 to AD2S.AD2S0 Bits = 001): V850E/IG3

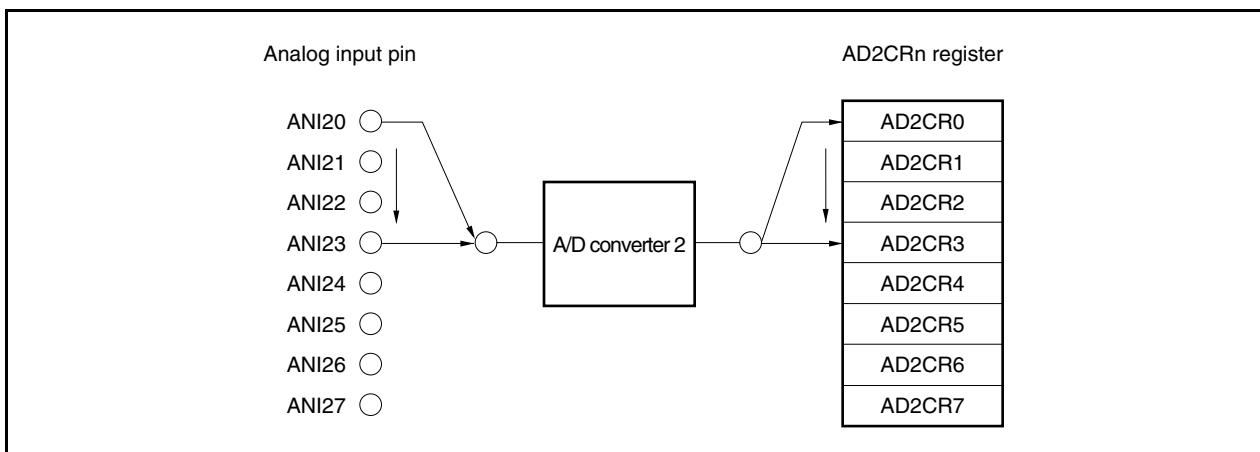
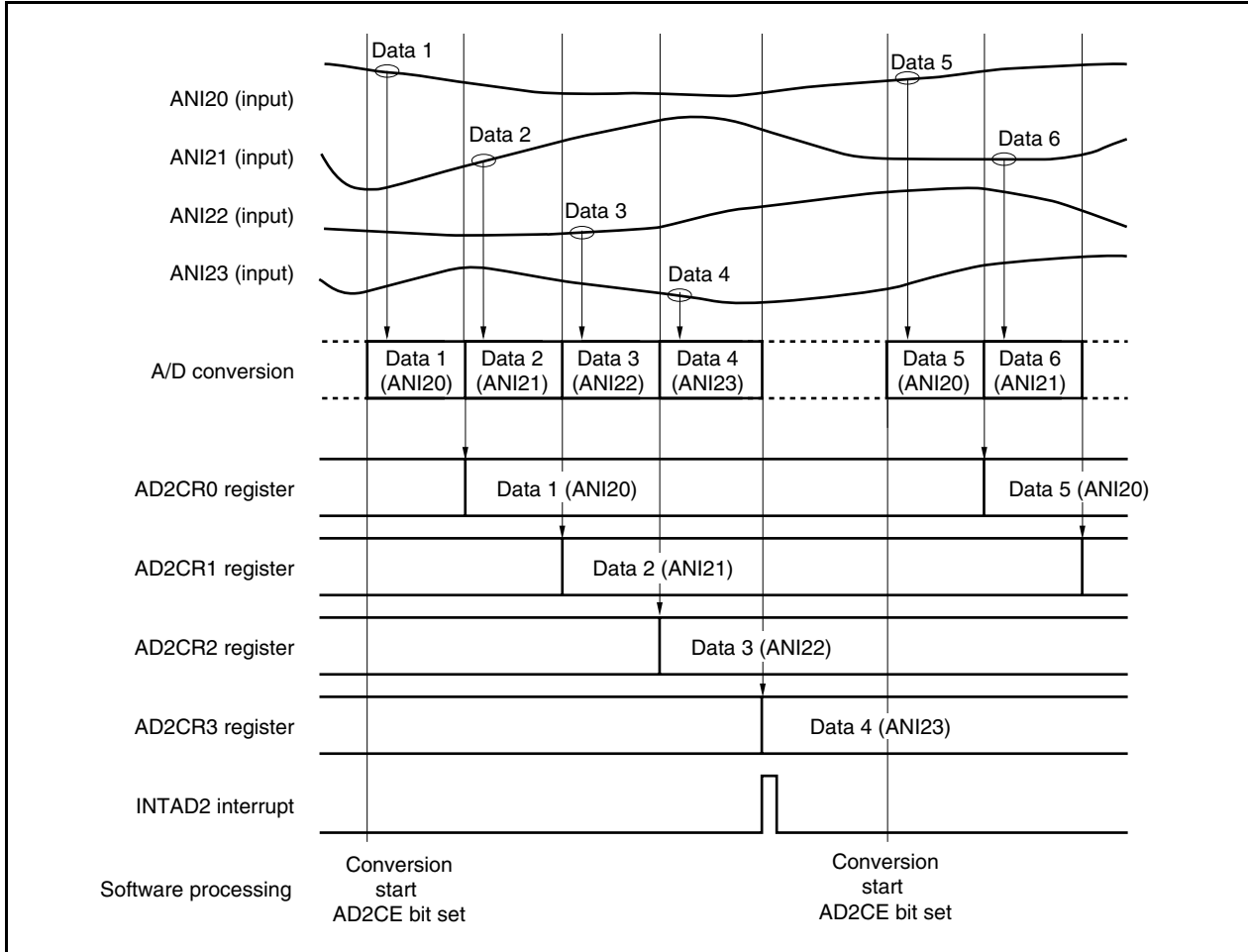


**(4) One-shot scan mode**

In this mode, pins up to the analog input pin (ANI2n) specified by the AD2S register from the ANI20 pin are selected sequentially, and A/D conversion is executed. The A/D conversion results are stored in the AD2CRn register corresponding to the analog input pin. When conversion of all the specified analog input pins ends, the A/D2 conversion end interrupt request signal (INTAD2) is generated. After A/D conversion ends, the conversion operation is stopped.

**Remark** V850E/IF3: n = 0 to 3  
V850E/IG3: n = 0 to 7

**Figure 13-7. One-Shot Scan Mode Operation Timing**  
 (When AD2M0.AD2MD1 and AD2M0.AD2MD0 Bits = 11, AD2S.AD2S2 to AD2S.AD2S0 Bits = 011): V850E/IG3



### 13.5 Operation in Software Trigger Mode

When the AD2M0.AD2CE bit is set to 1, A/D conversion is started.

When A/D conversion is started, the AD2M0.AD2EF bit = 1 (conversion in progress).

If the AD2M0 and AD2S registers are written during A/D conversion, the conversion is stopped and executed again from the beginning.

#### (1) Operation in software trigger continuous select mode

In this mode, one analog input pin (ANI2n) specified by the AD2S register is A/D-converted once. The conversion results are stored in one AD2CRn register. The ANI2n pin and AD2CRn register correspond one to one.

Each time an A/D conversion is executed, an A/D2 conversion end interrupt request signal (INTAD2) is generated and A/D conversion ends. After A/D conversion ends, the conversion is repeated again unless the AD2M0.AD2CE bit is set to 0.

It is not necessary to set (1) the AD2M0.AD2CE bit to restart A/D conversion<sup>Note</sup>.

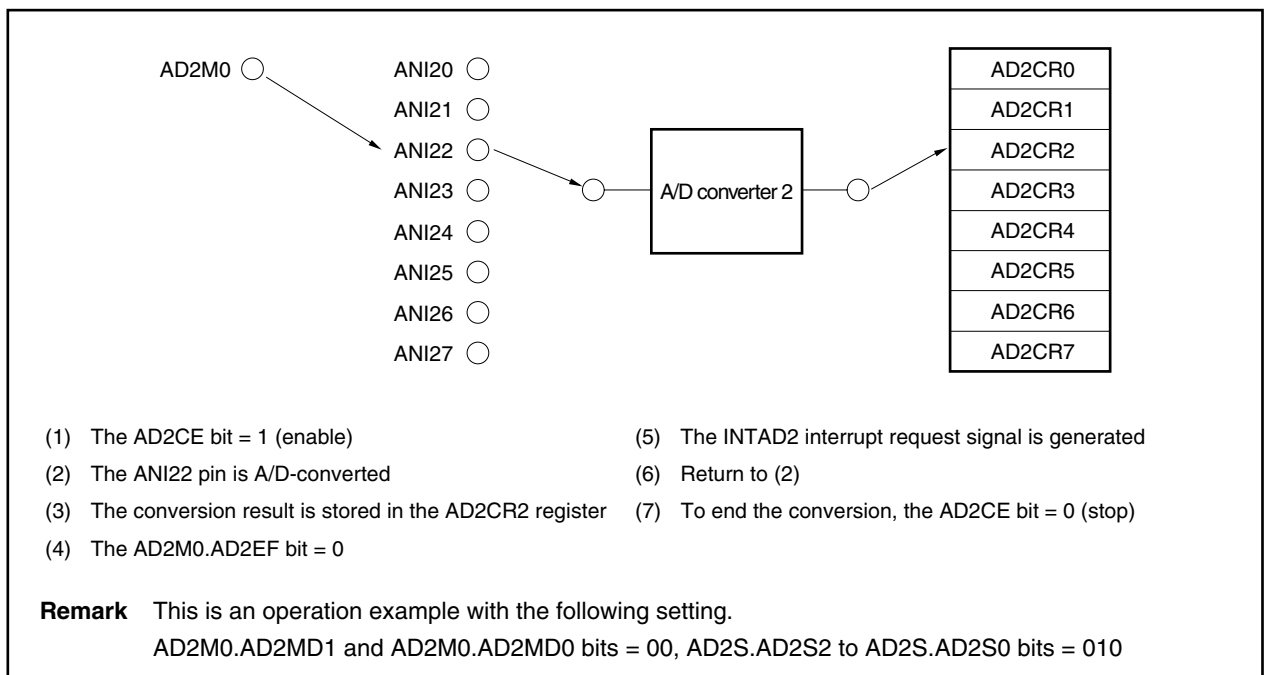
**Note** In the software trigger continuous select mode, the A/D conversion operation is not stopped unless the AD2M0.AD2CE bit is set to 0. If the AD2CRn register is not read before the next A/D conversion ends, it is overwritten.

This mode is suitable for applications in which the A/D conversion value of one analog input pin is read.

| Analog Input Pin | A/D Conversion Result Register |
|------------------|--------------------------------|
| ANI2n            | AD2CRn                         |

**Remark** V850E/IF3: n = 0 to 3  
V850E/IG3: n = 0 to 7

Figure 13-8. Operation Example of Software Trigger Continuous Select Mode: V850E/IG3



**(2) Software trigger continuous scan mode operations**

In this mode, pins up to the analog input pin (ANI2n) specified by the AD2S register from the ANI20 pin are selected sequentially, and A/D conversion is executed continuously. The A/D conversion results are stored in the AD2CRn register corresponding to the analog input pin.

When conversion of all the specified analog input pins ends, the A/D2 conversion end interrupt request signal (INTAD2) is generated. After A/D conversion ends, the conversion is started again from the ANI20 pin, unless the AD2M0.AD2CE bit is set to 0.

It is not necessary to set (1) the AD2M0.AD2CE bit to restart A/D conversion<sup>Note</sup>.

**Note** In the software trigger continuous scan mode, the A/D conversion operation is not stopped unless the AD2M0.AD2CE bit is set to 0. If the AD2CRn register is not read before the next A/D conversion ends, it is overwritten.

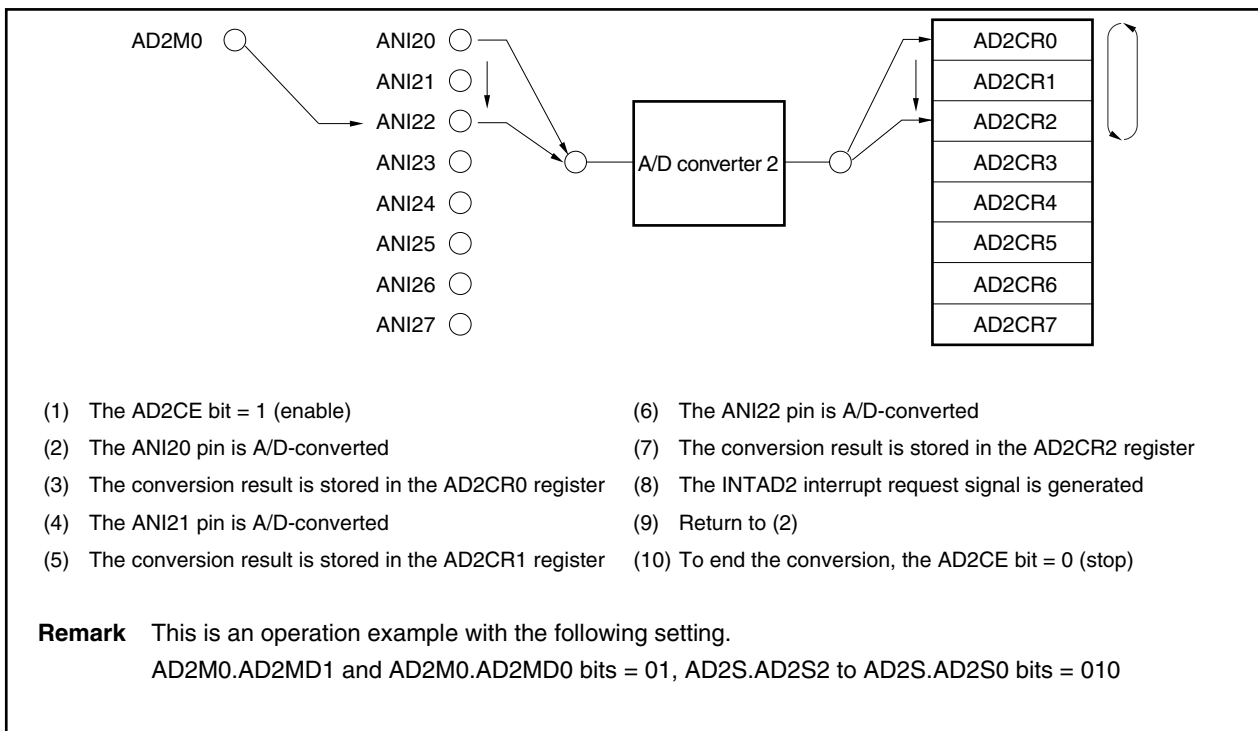
This mode is suitable for applications in which multiple analog inputs are constantly monitored.

| Analog Input Pin      | A/D Conversion Result Register |
|-----------------------|--------------------------------|
| ANI20                 | AD2CR0                         |
| ⋮                     | ⋮                              |
| ANI2n <sup>Note</sup> | AD2CRn                         |

**Note** Set by the AD2S.AD2S0 to AD2S.AD2S2 bits.

**Remark** V850E/IF3: n = 0 to 3  
V850E/IG3: n = 0 to 7

**Figure 13-9. Operation Example of Software Trigger Continuous Scan Mode: V850E/IG3**



**(3) Software trigger one-shot select mode**

In this mode, the voltage of one analog input pin (ANI2n) specified by the AD2S register is A/D-converted once. The conversion result is stored in one AD2CRn register. The ANI2n pin and the AD2CRn register correspond one to one.

Each time an A/D conversion is executed, an A/D2 conversion end interrupt request signal (INTAD2) is generated and A/D conversion ends. After A/D conversion ends, the conversion operation is stopped.

If the AD2M0.AD2CE bit is set to 1, A/D conversion can be restarted.

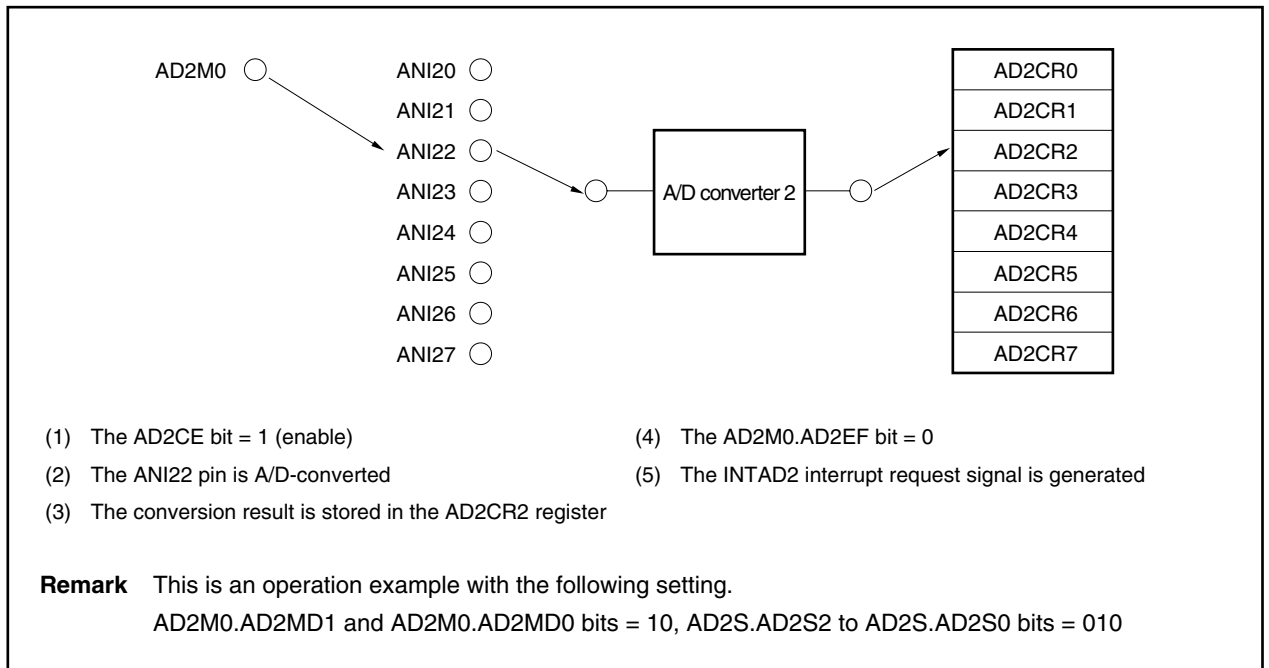
This mode is suitable for applications in which the results of each first-time A/D conversion are read.

| Analog Input Pin | A/D Conversion Result Register |
|------------------|--------------------------------|
| ANI2n            | AD2CRn                         |

**Remark** V850E/IF3: n = 0 to 3

V850E/IG3: n = 0 to 7

**Figure 13-10. Operation Example of Software Trigger One-Shot Select Mode: V850E/IG3**



**(4) Software trigger one-shot scan mode operations**

In this mode, pins up to the analog input pin (ANI2n) specified by the AD2S register from the ANI20 pin are selected sequentially, and A/D conversion is executed continuously. The A/D conversion results are stored in the AD2CRn register corresponding to the analog input pin.

When conversion of all the specified analog input pin ends, the A/D2 conversion end interrupt request signal (INTAD2) is generated. After A/D conversion ends, the conversion operation is stopped.

If the AD2M0.AD2CE bit is set to 1, A/D conversion can be restarted.

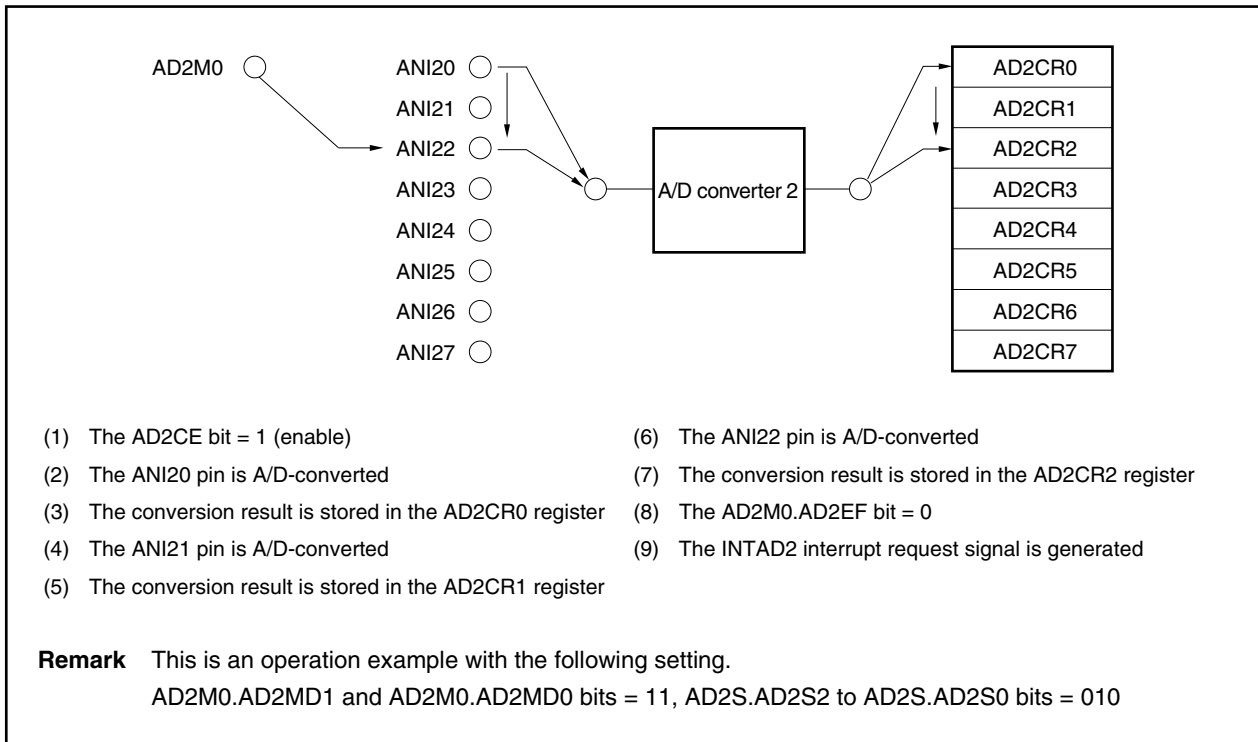
This mode is suitable for applications in which multiple analog inputs are constantly monitored.

| Analog Input Pin      | A/D Conversion Result Register |
|-----------------------|--------------------------------|
| ANI20                 | AD2CR0                         |
| ⋮                     | ⋮                              |
| ANI2n <sup>Note</sup> | AD2CRn                         |

**Note** Set by the AD2S.AD2S0 to AD2S.AD2S2 bits.

**Remark** V850E/IF3: n = 0 to 3  
V850E/IG3: n = 0 to 7

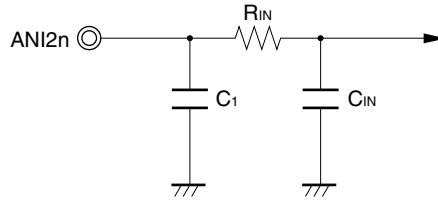
**Figure 13-11. Operation Example of Software Trigger One-Shot Scan Mode: V850E/IG3**





### 13.6 Internal Equivalent Circuit

The following figure shows the equivalent circuit of the analog input block.

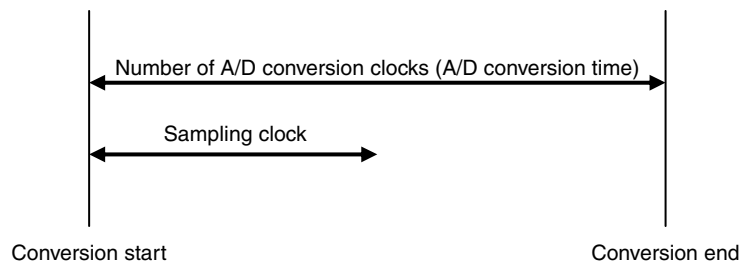


| R      | C <sub>1</sub> | C <sub>2</sub> |
|--------|----------------|----------------|
| 2.6 kΩ | 15 pF          | 6.2 pF         |

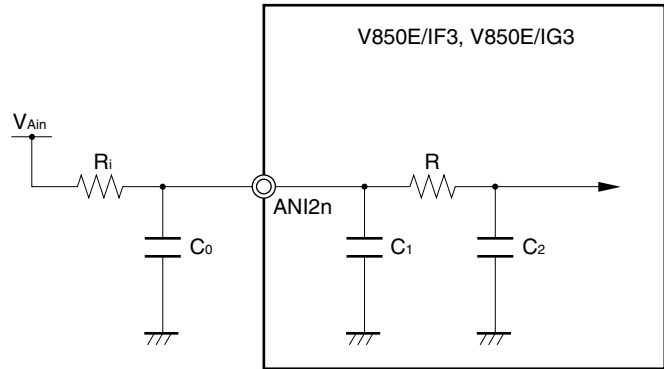
- Remarks**
1. The maximum values are shown (reference values).
  2. V850E/IF3: n = 0 to 3  
V850E/IG3: n = 0 to 7

| AD2M1 register |            |            |            | Number of A/D conversion clocks (f <sub>AD2</sub> ) | Number of sampling clocks |
|----------------|------------|------------|------------|---|---------------------------|
| AD2FR3 bit     | AD2FR2 bit | AD2FR1 bit | AD2FR0 bit |   |                           |
| 0              | 0          | 0          | 1          | 62  | 33                        |
| 0              | 0          | 1          | 0          | 93  | 49.5                      |
| 0              | 0          | 1          | 1          | 124   | 66                        |
| 0              | 1          | 0          | 0          | 155   | 82.5                      |
| 0              | 1          | 0          | 1          | 186   | 99                        |
| 0              | 1          | 1          | 0          | 217   | 115.5                     |
| 0              | 1          | 1          | 1          | 248   | 132                       |
| 1              | 0          | 0          | 0          | 279   | 148.5                     |
| 1              | 0          | 0          | 1          | 310   | 165                       |

**Caution** Number of sampling clocks is included in number of A/D conversion clocks.



An example of calculating an overall error of A/D converter 2 is shown below.



| $f_{xx}$<br>(MHz) | A/D conversion<br>time ( $\mu\text{s}$ ) | Sampling ( $\mu\text{s}$ ) | R<br>( $\text{k}\Omega$ ) | $C_1$<br>(pF) | $C_2$<br>(pF) | $C_0$<br>(pF) | $R_i$<br>( $\text{k}\Omega$ ) | Sampling error<br>(LSB) <sup>Note</sup> |
|-------------------|--|----------------------------|---------------------------|---------------|---------------|---------------|-------------------------------|---|
| 64                | 3.88<br>( $62/f_{AD2}$ )                 | 1.03<br>( $33/f_{AD2}$ )   | 2.6                       | 15            | 6.2           | 100           | 1.0                           | 0.1 or lower                            |
|                   |  |                            |                           |               |               | 100           | 0.5                           | 0.1 or lower                            |
|                   |  |                            |                           |               |               | 100           | 0.25                          | 0.1 or lower                            |
|                   |  |                            |                           |               |               | 100           | 0.125                         | 0.1 or lower                            |
|                   |  |                            |                           |               |               | 50            | 1.0                           | 0.1 or lower                            |
|                   |  |                            |                           |               |               | 50            | 0.5                           | 0.1 or lower                            |
|                   |  |                            |                           |               |               | 50            | 0.25                          | 0.1 or lower                            |
|                   |  |                            |                           |               |               | 50            | 0.125                         | 0.1 or lower                            |

**Note** The error when considering the signal source impedance is “sampling error + overall error”.

- Remarks**
1. These values are reference values calculated by simulating what happens to  $C_2$  voltage by  $R_i$  and  $C_0$  when  $V_{Ain}$  is applied from 0 V to 5 V at the same time as sampling start.
  2. V850E/IF3:  $n = 0$  to 3  
V850E/IG3:  $n = 0$  to 7
  3.  $f_{xx}$ : System clock frequency  
 $f_{AD2}$ : Operating clock frequency

13.7 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 AD2M0.AD2CE and AD2M0.AD2PS bits to 0.

(2) Input range of ANI2n pin

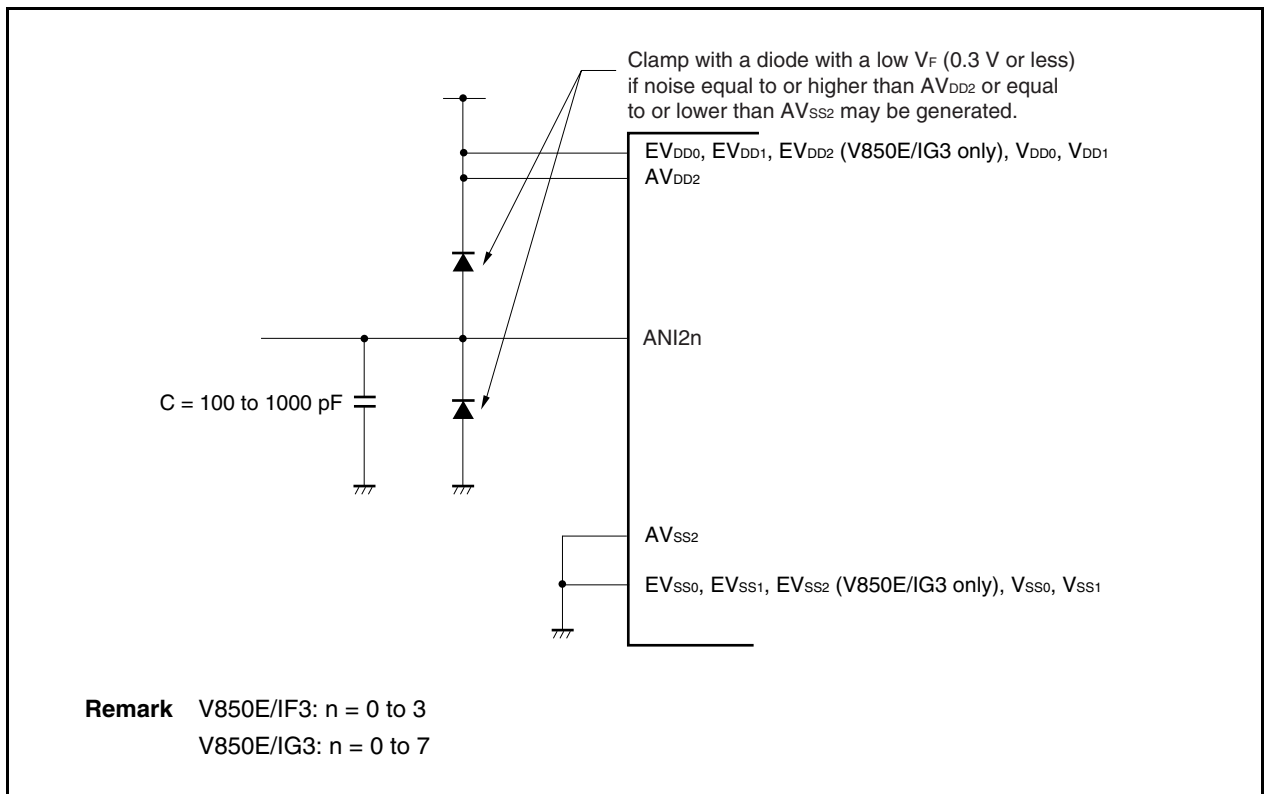
Input the voltage within the specified range to the ANI2n pin. If a voltage equal to or higher than  $AV_{DD2}$  or equal to or lower than  $AV_{SS2}$  (even within the range of the absolute maximum ratings) is input to this pin, the conversion value of that channel is undefined, and the conversion value of the other channels may also be affected.

(3) Countermeasures against noise

To maintain the 10-bit resolution, the ANI2n pin 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 13-12 is recommended.

<R>

Figure 13-12. Processing of Analog Input Pin



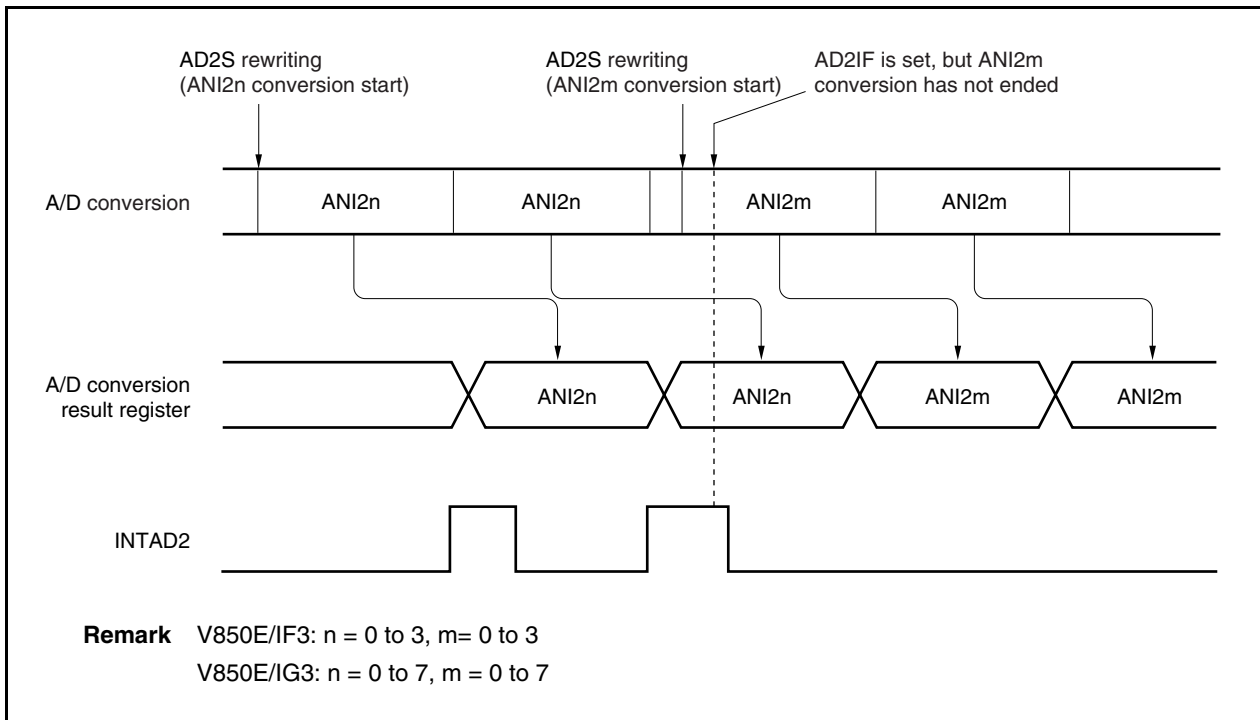
(4) Alternate input

The analog input pin (ANI2n) functions alternately as input port (P7n). When selecting one of the ANI2n pin to execute A/D conversion, do not execute an input instruction to port 7 during conversion as the conversion resolution may drop.

(5) Interrupt request flag (AD2IF)

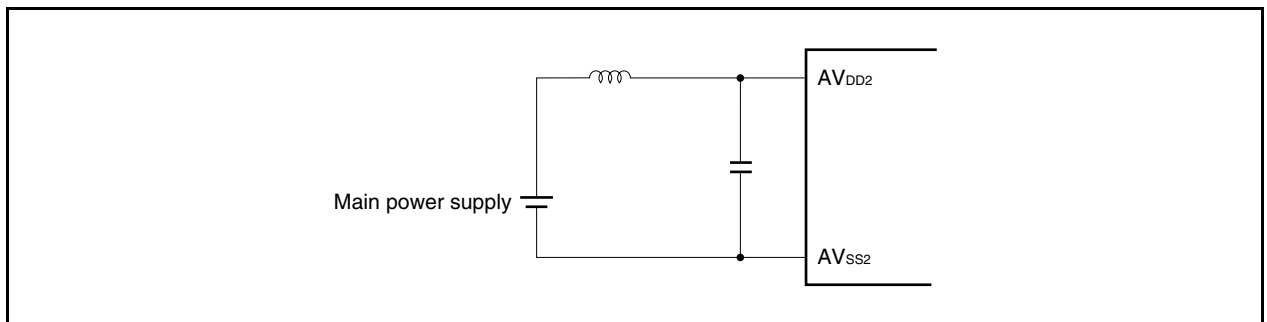
The interrupt request flag (AD2IF) is not cleared even if the contents of the AD2S register are changed. If the analog input pin is changed during A/D conversion, therefore, the result of converting the previously selected analog input signal may be stored and the A/D2 conversion end interrupt request flag may be set immediately before the AD2S register is rewritten. If the AD2IF flag is read immediately after the AD2S register is rewritten, the AD2IF 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 AD2IF flag before resuming conversion.

Figure 13-13. Generation Timing of A/D2 Conversion End Interrupt Request



**(6) AV<sub>DD2</sub> pin**

- (a) The AV<sub>DD2</sub> pin is used as the power supply pin of the A/D converter 2 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 potential as EV<sub>DD0</sub>, EV<sub>DD1</sub>, and EV<sub>DD2</sub> (V850E/IG3 only) to the AV<sub>DD2</sub> pin as shown in Figure 13-12.
- (b) The AV<sub>DD2</sub> pin is also used as the reference voltage pin of the A/D converter 2. If the source supplying power to the AV<sub>DD2</sub> 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 (AD2CE bit = 1)). As a result, the conversion accuracy may drop. To avoid this, it is recommended to connect a capacitor across the AV<sub>DD2</sub> and AV<sub>SS2</sub> pins to suppress the reference voltage fluctuation as shown in Figure 13-14.
- (c) If the source supplying power to the AV<sub>DD2</sub> 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 13-14. AV<sub>DD2</sub> Pin Connection Example****(7) Reading AD2CRn register**

When the AD2M0, AD2M1, or AD2S register is written, the contents of the AD2CRn register may be undefined. Read the conversion result after completion of conversion and before writing to the AD2M0, AD2M1, and AD2S registers. The correct conversion result may not be read at a timing different from the above.

**(8) A/D conversion result**

If there is noise at the analog input pin (ANI2n) or at the power supply voltage pin (AV<sub>DD2</sub>), 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 successively 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 counteractive measures.

**(9) Standby mode**

Because the A/D converter 2 stops operating in the IDLE and STOP modes, conversion results are invalid, so power consumption can be reduced. Operations are resumed after the IDLE and STOP modes are released, but the A/D conversion results after the IDLE and STOP modes are released are invalid. When using the A/D converter 2 after the IDLE and STOP modes are released, before setting the IDLE and STOP modes or releasing the IDLE and STOP modes, set the AD2M0.AD2CE bit to 0 then set the AD2CE bit to 1 after releasing the IDLE and STOP modes.

**(10) 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 by averaging the A/D conversion results.

**(11) A/D conversion result hysteresis characteristics**

Successive comparison type A/D converters hold an analog input voltage in an internal sample & hold capacitor and then perform 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 if the output impedance from the analog input source is too high.

- 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. Even if the conversion were to be performed at the same potential, the results may thus 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. Even if the conversion were to be performed at the same potential, the results may thus vary.

To obtain more accurate conversion results, lower the output impedance from the analog input source or execute A/D conversion twice consecutively on the same channel, and discard the first conversion result.

**13.8 How to Read A/D Converter Characteristics Table**

For details about the A/D converter characteristics table, see **12.7 How to Read A/D Converter Characteristics Table**.

## CHAPTER 14 ASYNCHRONOUS SERIAL INTERFACE A (UARTA)

### 14.1 Mode Switching Between UARTA and Other Serial Interface

#### 14.1.1 Mode switching between UARTA0 and CSIB0

In the V850E/IF3 and V850E/IG3, UARTA0 and CSIB0 function alternately, and these pins cannot be used at the same time. To switch between UARTA0 and CSIB0, the PMC4, PFC4, and PFCE4 registers must be set in advance.

**Caution** The operations related to transmission and reception of UARTA0 or CSIB0 are not guaranteed if the mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 14-1. Mode Switch Settings of UARTA0 and CSIB0

|                  |        |  |  |        |        |        |        |        |
|------------------|--------|--|--|--------|--------|--------|--------|--------|
| After reset: 00H |        | R/W  | Address: FFFFF448H                             |        |        |        |        |        |
| PMC4             | 7      | 6  | 5  | 4      | 3      | 2      | 1      | 0      |
|                  | PMC47  | PMC46  | PMC45  | PMC44  | PMC43  | PMC42  | PMC41  | PMC40  |
| After reset: 00H |        | R/W  | Address: FFFFF468H                             |        |        |        |        |        |
| PFC4             | 7      | 6  | 5  | 4      | 3      | 2      | 1      | 0      |
|                  | PFC47  | PFC46  | PFC45  | PFC44  | PFC43  | PFC42  | PFC41  | PFC40  |
| After reset: 00H |        | R/W  | Address: FFFFF708H                             |        |        |        |        |        |
| PFCE4            | 7      | 6  | 5  | 4      | 3      | 2      | 1      | 0      |
|                  | PFCE47 | PFCE46   | PFCE45   | PFCE44 | PFCE43 | PFCE42 | PFCE41 | PFCE40 |
| PMC42            | PFCE42 | PFC42  | Specification of alternate function of P42 pin |        |        |        |        |        |
| 0                | ×      | ×  | I/O port                                       |        |        |        |        |        |
| 1                | 0      | 0  | SCKB0 I/O                                      |        |        |        |        |        |
| 1                | 0      | 1  | INTP13 input                                   |        |        |        |        |        |
| 1                | 1      | 0  | Setting prohibited                             |        |        |        |        |        |
| 1                | 1      | 1  | Setting prohibited                             |        |        |        |        |        |
| PMC41            | PFCE41 | PFC41  | Specification of alternate function of P41 pin |        |        |        |        |        |
| 0                | ×      | ×  | I/O port                                       |        |        |        |        |        |
| 1                | 0      | 0  | SOB0 output                                    |        |        |        |        |        |
| 1                | 0      | 1  | TXDA0 output                                   |        |        |        |        |        |
| 1                | 1      | 0  | Setting prohibited                             |        |        |        |        |        |
| 1                | 1      | 1  | Setting prohibited                             |        |        |        |        |        |
| PMC40            | PFC40  | Specification of alternate function of P40 pin |  |        |        |        |        |        |
| 0                | ×      | I/O port                                       |  |        |        |        |        |        |
| 1                | 0      | SIB0 input                                     |  |        |        |        |        |        |
| 1                | 1      | RXDA0 input                                    |  |        |        |        |        |        |

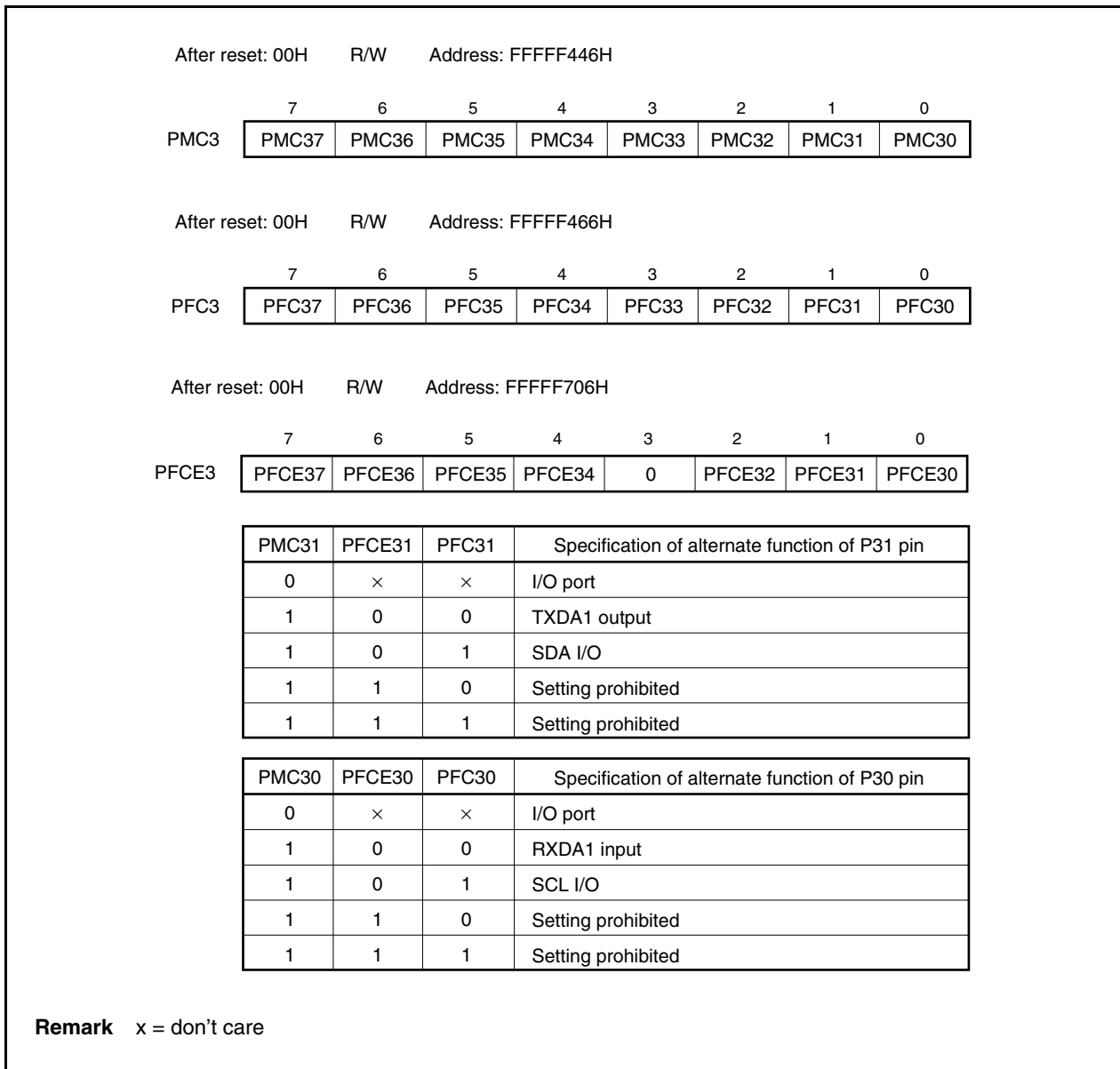
**Remark** × = don't care

14.1.2 Mode switching between UARTA1 and I<sup>2</sup>C

In the V850E/IF3 and V850E/IG3, UARTA1 and I<sup>2</sup>C function alternately, and their pins cannot be used at the same time. To switch between UARTA1 and I<sup>2</sup>C, the PMC3, PFC3, and PFCE3 registers must be set in advance.

**Caution** The operations related to transmission and reception of UARTA1 or I<sup>2</sup>C are not guaranteed if the mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 14-2. Mode Switch Settings of UARTA1 and I<sup>2</sup>C



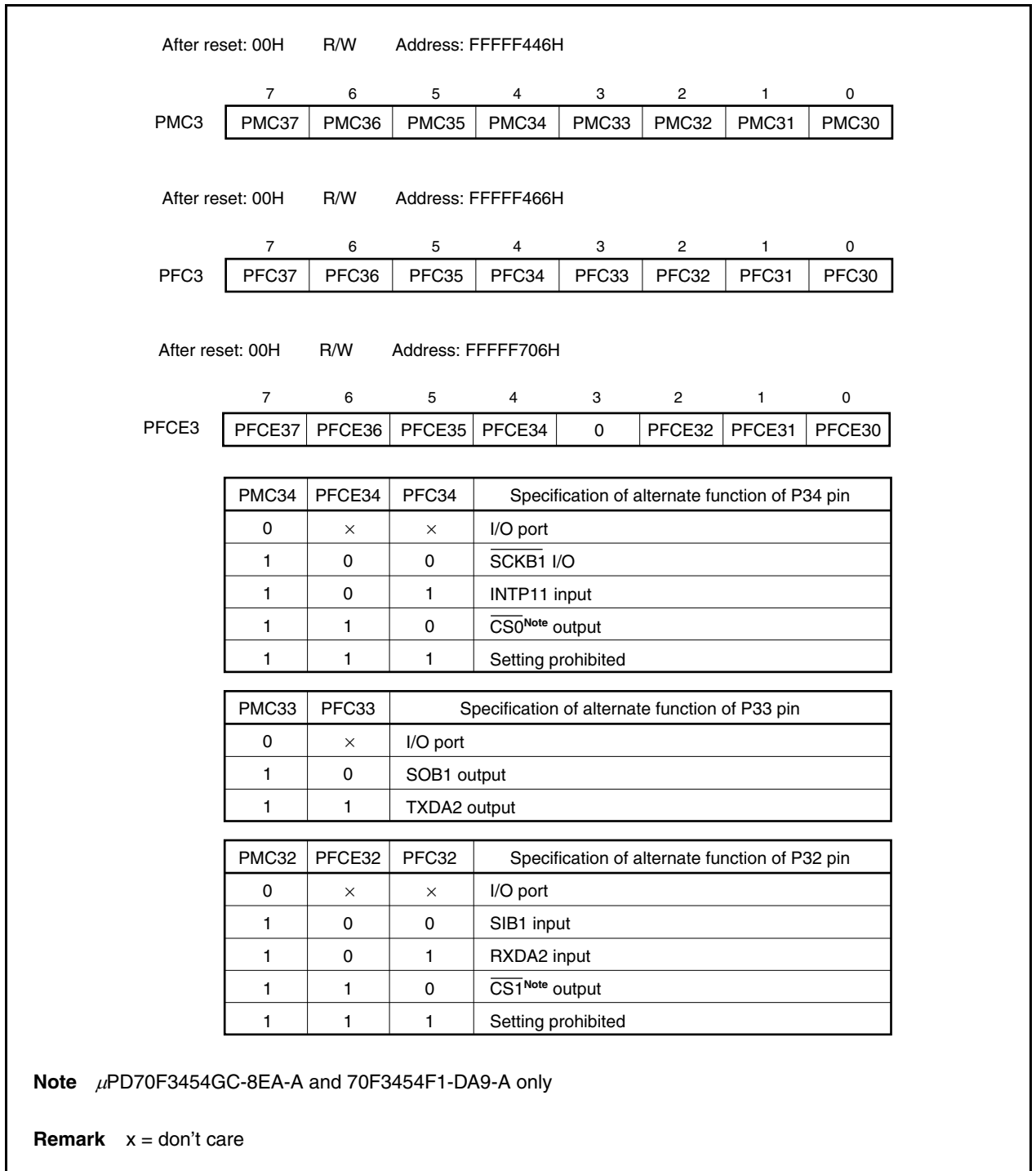


14.1.3 Mode switching between UARTA2 and CSIB1

In the V850E/IF3 and V850E/IG3, UARTA2 and CSIB1 function alternately, and their pins cannot be used at the same time. To switch between UARTA2 and CSIB1, the PMC3, PFC3, and PFCE3 registers must be set in advance.

**Caution** The operations related to transmission and reception of UARTA2 or CSIB1 are not guaranteed if the mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 14-3. Mode Switch Settings of UARTA2 and CSIB1



## 14.2 Features

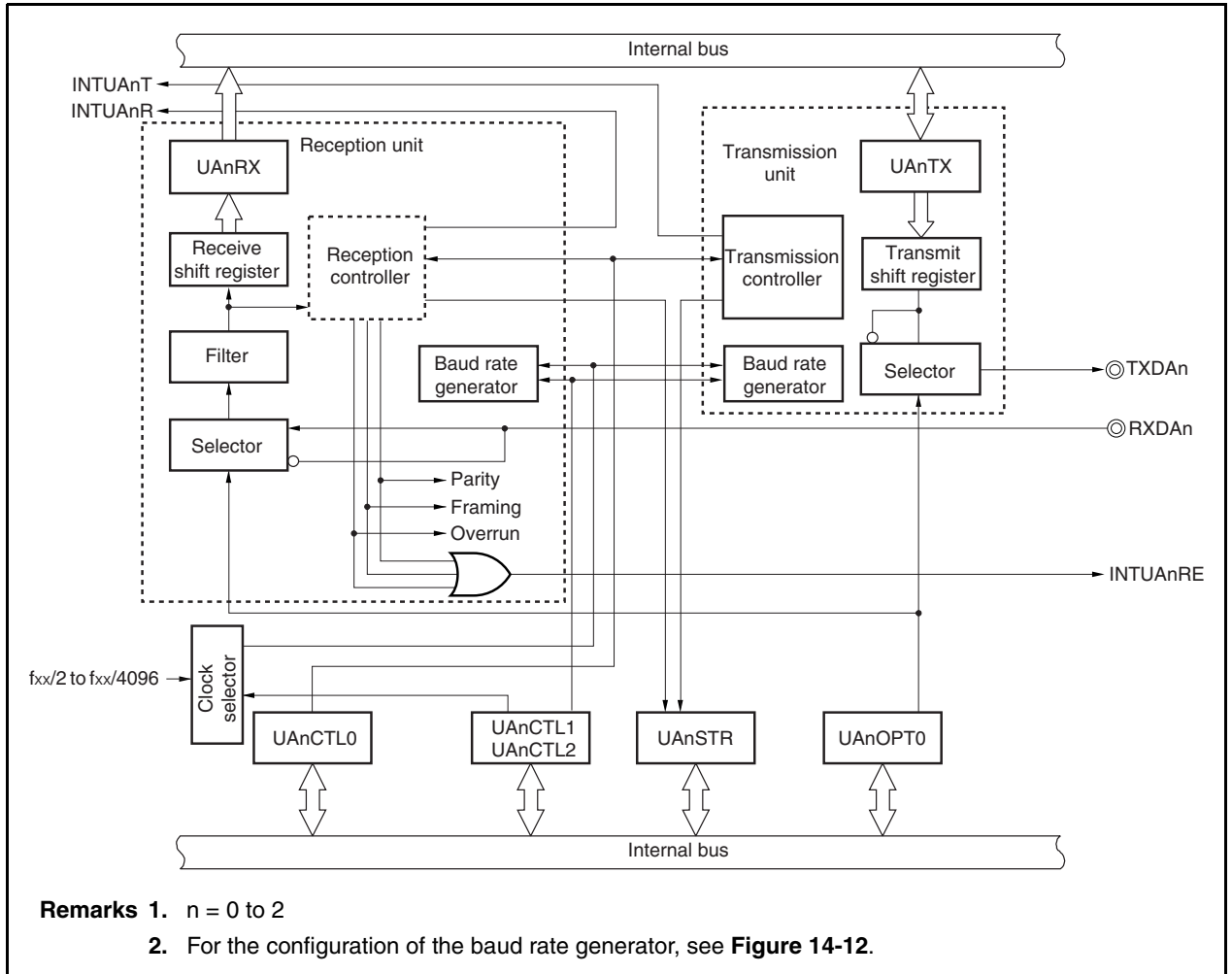
- Transfer rate: 300 bps to 1.25 Mbps (using peripheral clock ( $f_{xx}$ ) of 64 MHz and dedicated baud rate generator)
- Full-duplex communication: Internal UARTA receive data register n (UAnRX)  
Internal UARTA transmit data register n (UAnTX)
- 2-pin configuration: TXDAn: Transmit data output pin  
RXDAn: Receive data input pin
- Reception error output function
  - Parity error
  - Framing error
  - Overrun error
- Interrupt sources: 3
  - Reception error interrupt (INTUAnRE): This interrupt is generated by ORing the three types of reception errors
  - Reception end interrupt (INTUAnR): This interrupt occurs upon transfer of receive data from the shift register to the UAnRX register after serial transfer end, in the reception enabled status.
  - Transmission enable interrupt (INTUAnT): This interrupt occurs upon transfer of transmit data from the UAnTX register to the shift register in the transmission enabled status.
- Character length: 7, 8 bits
- Parity function: Odd, even, 0, none
- Transmission stop bit: 1, 2 bits
- On-chip dedicated baud rate generator
- MSB-/LSB-first transfer selectable
- Transmit/receive data inverted input/output possible

**Remark** n = 0 to 2

### 14.3 Configuration

The block diagram of the UARTAn is shown below.

Figure 14-4. Block Diagram of UARTAn



UARTAn consists of the following hardware units.

Table 14-1. Configuration of UARTAn

| Item      | Configuration  |
|-----------|--|
| Registers | UARTAn control register 0 (UANCTL0)<br>UARTAn control register 1 (UANCTL1)<br>UARTAn control register 2 (UANCTL2)<br>UARTAn option control register 0 (UANOPT0)<br>UARTAn status register (UANSTR)<br>UARTAn receive shift register<br>UARTAn receive data register (UANRX)<br>UARTAn transmit shift register<br>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 base clock ( $f_{UCLK}$ ) 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 UAnSTR register consists of flags indicating the error contents when a reception error occurs. Each one of the reception error flags is set (to 1) upon occurrence of a reception error.

**(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 end interrupt request signal (INTUAnR) to be output.

**(8) UARTAn transmit shift register**

The UARTAn 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 UARTAn transmit 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.

14.4 Control Registers

(1) UARTAn control register 0 (UAnCTL0)

The UAnCTL0 register is an 8-bit register that controls the UARTAn serial transfer operation.

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

Reset sets this register to 10H.

(1/2)

After reset: 10H    R/W    Address: UA0CTL0 FFFFFFFA00H, UA1CTL0 FFFFFFFA10H,  
UA2CTL0 FFFFFFFA20H

|                         |        |        |        |        |        |        |       |       |
|-------------------------|--------|--------|--------|--------|--------|--------|-------|-------|
|                         | <7>    | <6>    | <5>    | <4>    | 3      | 2      | 1     | 0     |
| UAnCTL0<br>(n = 0 to 2) | UAnPWR | UAnTXE | UAnRXE | UAnDIR | UAnPS1 | UAnPS0 | UAnCL | UAnSL |

|  |  |
|--|--|
| UAnPWR   | UARTAn operation control                               |
| 0  | Disable UARTAn operation (UARTAn reset asynchronously) |
| 1  | Enable UARTAn operation                                |
| <p>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).</p> |  |

|  |                                |
|--|--------------------------------|
| UAnTXE   | Transmission operation enable  |
| 0  | Disable transmission operation |
| 1  | Enable transmission operation  |
| <ul style="list-style-type: none"> <li>• To start transmission, set the UAnPWR bit to 1 and then set the UAnTXE bit to 1.</li> <li>• To initialize the transmission unit, clear the UAnTXE bit to 0, wait for two cycles of the base clock (<math>f_{CLK}</math>), and then set the UAnTXE bit to 1 again. Otherwise, initialization may not be executed (for the base clock, see <b>14.7 (1) (a) Base clock</b>).</li> <li>• When the operation is enabled (UAnPWR bit = 1), the transmission operation is enabled after two or more cycles of the base clock (<math>f_{CLK}</math>) have elapsed since UAnTXE = 1.</li> <li>• When the UAnPWR bit is cleared to 0, the status of the internal circuit becomes the same status as UAnTXE bit = 0 by the UAnPWR bit even if the UAnTXE bit is 1. The transmission operation is enabled when the UAnPWR bit is set to 1 again.</li> </ul> |                                |

|   |                             |
|---|-----------------------------|
| UAnRXE  | Reception operation enable  |
| 0   | Disable reception operation |
| 1   | Enable reception operation  |
| <ul style="list-style-type: none"> <li>• To start reception, set the UAnPWR bit to 1 and then set the UAnRXE bit to 1.</li> <li>• To initialize the reception unit, clear the UAnRXE bit to 0, wait for two cycles of the base clock, and then set the UAnRXE bit to 1 again. Otherwise, initialization may not be executed (for the base clock, see <b>14.7 (1) (a) Base clock</b>).</li> <li>• When the operation is enabled (UAnPWR bit = 1), the reception operation is enabled after two or more cycles of the base clock (<math>f_{CLK}</math>) have elapsed since UAnRXE = 1. The start bit is ignored if it is received before the reception operation is enabled.</li> <li>• When the UAnPWR bit is cleared to 0, the status of the internal circuit becomes the same status as UAnRXE bit = 0 by the UAnPWR bit even if the UAnRXE bit is 1. The reception operation is enabled when the UAnPWR bit is set to 1 again.</li> </ul> |                             |

| UAnDIR <sup>Note</sup> | Transfer direction selection |
|------------------------|------------------------------|
| 0                      | MSB-first transfer           |
| 1                      | LSB-first transfer           |

| UAnPS1 <sup>Note</sup> | UAnPS0 <sup>Note</sup> | 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                 |

If “reception with 0 parity” is selected during reception, a parity check is not performed. Therefore, since the UAnSTR.UAnPE bit is not set, no error interrupt due to a parity error is output.

| UAnCL <sup>Note</sup> | Specification of data character length of 1 frame of transmit/receive data |
|-----------------------|--|
| 0                     | 7 bits   |
| 1                     | 8 bits   |

| UAnSL <sup>Note</sup> | Specification of length of stop bit for transmit data |
|-----------------------|---|
| 0                     | 1 bit   |
| 1                     | 2 bits  |

Only the first bit of the receive data stop bits is checked, regardless of the value of the UAnSL bit.

**Note** This register can be rewritten only when the UAnPWR bit = 0 or the UAnTXE bit = UAnRXE bit = 0. However, setting any or all of the UAnPWR, UAnTXE, and UAnRXE bits to 1 at the same time is possible.

**Remark** For details of parity, see 14.6.6 Parity types and operations.

**(2) UARTAn control register 1 (UAnCTL1)**

For details, see 14.7 (2) UARTAn control register 1 (UAnCTL1).

**(3) UARTAn control register 2 (UAnCTL2)**

For details, see 14.7 (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 UARTAn. This register can be read or written in 8-bit or 1-bit units. Reset sets this register to 14H.

After reset: 14H    R/W    Address: UA0OPT0 FFFFA03H, UA1OPT0 FFFFA13H,  
UA2OPT0 FFFFA23H

|                         |   |   |   |   |   |   |        |        |
|-------------------------|---|---|---|---|---|---|--------|--------|
|                         | 7 | 6 | 5 | 4 | 3 | 2 | 1      | 0      |
| UAnOPT0<br>(n = 0 to 2) | 0 | 0 | 0 | 1 | 0 | 1 | UAnTDL | UAnRDL |

|   |                                  |
|---|----------------------------------|
| UAnTDL  | Transmit data level bit          |
| 0   | Normal output of transfer data   |
| 1   | Inverted output of transfer data |
| <ul style="list-style-type: none"> <li>• The output level of the TXDAn pin can be inverted using the UAnTDL bit.</li> <li>• This register can be set when the UAnCTL0.UAnPWR bit = 0 or when the UAnCTL0.UAnTXE bit = 0.</li> </ul> |                                  |

|  |                                 |
|--|---------------------------------|
| UAnRDL   | Receive data level bit          |
| 0  | Normal input of transfer data   |
| 1  | Inverted input of transfer data |
| <ul style="list-style-type: none"> <li>• The input level of the RXDAn pin can be inverted using the UAnRDL bit.</li> <li>• This register can be set when the UAnPWR bit = 0 or the UAnCTL0.UAnRXE bit = 0.</li> <li>• When the UAnRDL bit is set to 1 (inverted input of receive data), reception must be enabled (UAnCTL0.UAnRXE bit = 1) after setting the data reception pin to the UART reception pin (RXDAn) when reception is started. When the pin mode is changed after reception is enabled, the start bit will be mistakenly detected if the pin level is high.</li> </ul> |                                 |

**Caution** Be sure to set bits 3 and 5 to 7 to “0”, and set bits 2 and 4 to “1”. Operation with other settings is not guaranteed.

**(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"> <li>• After reset</li> <li>• UAnCTL0.UAnPWR bit = 0</li> </ul> |
| UAnTSF bit                | <ul style="list-style-type: none"> <li>• UAnCTL0.UAnTXE bit = 0</li> </ul>                        |
| UAnPE, UAnFE, UAnOVE bits | <ul style="list-style-type: none"> <li>• 0 write</li> <li>• UAnCTL0.UAnRXE bit = 0</li> </ul>     |

**Caution** Be sure to read and check the error flags of the UAnPE, UAnFE, and UAnOVE bits, and clear the flags by writing “0” to them.

After reset: 00H R/W Address: UA0STR FFFFA04H, UA1STR FFFFA14H,  
UA2STR FFFFA24H

|                        |        |   |   |   |   |       |       |        |
|------------------------|--------|---|---|---|---|-------|-------|--------|
|                        | <7>    | 6 | 5 | 4 | 3 | <2>   | <1>   | <0>    |
| UAnSTR<br>(n = 0 to 2) | UAnTSF | 0 | 0 | 0 | 0 | UAnPE | UAnFE | UAnOVE |

|  |  |
|--|--|
| UAnTSF   | Transfer status flag   |
| 0  | <ul style="list-style-type: none"> <li>• When the UAnPWR bit = 0 or the UAnTXE bit = 0 has been set.</li> <li>• When, following transfer end, there was no next data transfer from UAnTX register</li> </ul> |
| 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"> <li>• When the UAnPWR bit = 0 or the UAnRXE bit = 0 has been set.</li> <li>• When 0 has been written</li> </ul> |
| 1   | When parity of data and parity bit do not match during reception.  |
| <ul style="list-style-type: none"> <li>• The operation of the UAnPE bit is controlled by the settings of the UAnCTL0.UAnPS1 and UAnCTL0.UAnPS0 bits.</li> <li>• 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.</li> </ul> |  |

|   |  |
|---|--|
| UAnFE   | Framing error flag   |
| 0   | <ul style="list-style-type: none"> <li>• When the UAnPWR bit = 0 or the UAnRXE bit = 0 has been set.</li> <li>• When 0 has been written</li> </ul> |
| 1   | When no stop bit is detected during reception  |
| <ul style="list-style-type: none"> <li>• Only the first bit of the receive data stop bits is checked, regardless of the value of the UAnCTL0.UAnSL bit.</li> <li>• 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.</li> </ul> |  |

|  |  |
|--|--|
| UAnOVE   | Overrun error flag   |
| 0  | <ul style="list-style-type: none"> <li>• When the UAnPWR bit = 0 or the UAnRXE bit = 0 has been set.</li> <li>• When 0 has been written</li> </ul> |
| 1  | When receive data has been set to the UAnRX register and the next receive operation is ended before that receive data has been read.               |
| <ul style="list-style-type: none"> <li>• When an overrun error occurs, the data is discarded without the next receive data being written to the UAnRX register.</li> <li>• The UAnOVE bit can be both read and written, but it can only be cleared by writing 0 to it, and it cannot be set by writing 1 to it. When 1 is written to this bit, the value is retained.</li> </ul> |  |



**(6) UARTAn receive data register (UAnRX)**

The UAnRX register is an 8-bit buffer register that stores parallel data converted by the UARTAn receive shift register.

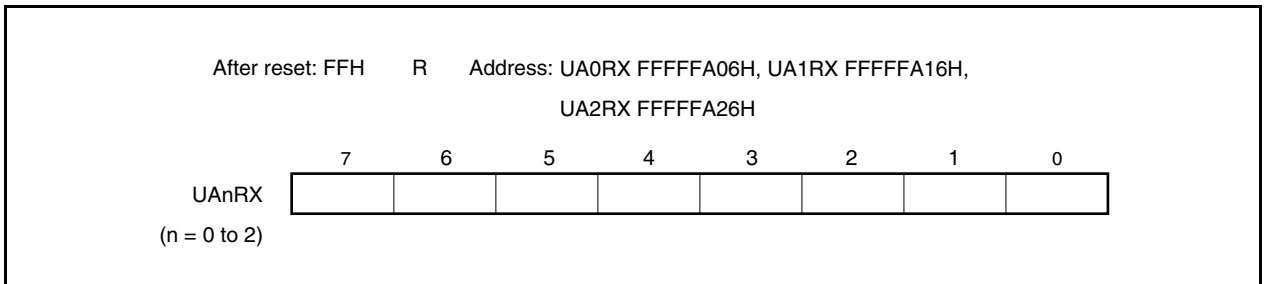
The data stored in the UARTAn receive shift register is transferred to the UAnRX register upon end of reception of 1 byte of data. A reception end interrupt request signal (INTUAnR) is generated at this timing.

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 occurs (UAnSTR.UAnOVE bit = 1), 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, the UAnRX register can be set to FFH by clearing the UAnCTL0.UAnPWR bit to 0.



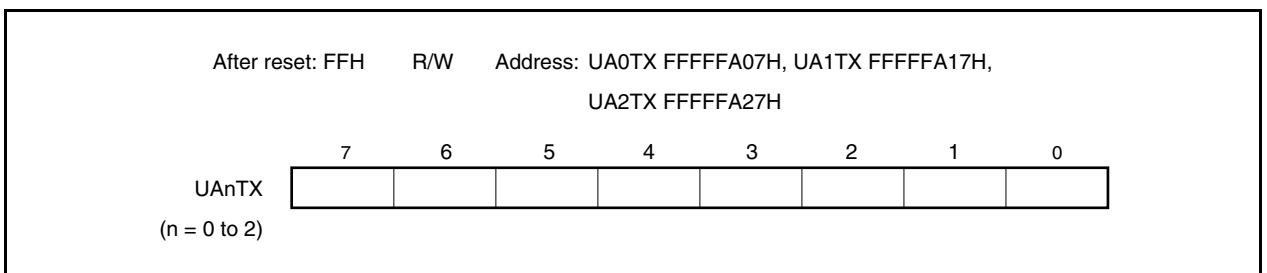
**(7) UARTAn transmit data register (UAnTX)**

The UAnTX register is an 8-bit register used to set transmit data.

Transmission starts when transmit data is written to the UAnTX register in the transmission enabled status (UAnCTL0.UAnTXE bit = 1). Upon end of the transfer of the data of the UAnTX register to the UARTAn transmit shift register, the transmission enable interrupt request signal (INTUAnT) is generated.

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

Reset sets this register to FFH.



## 14.5 Interrupt Request Signals

The following three interrupt request signals are generated from UARTAn.

- Reception error interrupt request signal (INTUAnRE)
- Reception end interrupt request signal (INTUAnR)
- Transmission enable interrupt request signal (INTUAnT)

Among these three interrupt signals, the reception error interrupt signal has the highest default priority, and the reception end interrupt request signal and transmission enable interrupt request signal follow in this order.

**Table 14-2. Interrupts and Their Default Priorities**

| Interrupt           | Priority |
|---------------------|----------|
| Reception error     | High     |
| Reception end       | ↕        |
| Transmission enable | Low      |

### (1) Reception error interrupt request signal (INTUAnRE)

A reception error interrupt request signal is generated while reception is enabled by ORing the three types of reception errors (parity error, framing error, and overrun error) explained in the UAnSTR register section.

### (2) Reception end interrupt request signal (INTUAnR)

A reception end interrupt request signal is output when data is shifted into the UARTAn receive shift register and transferred to the UAnRX register in the reception enabled status.

No reception end interrupt request signal is generated in the reception disabled status.

### (3) 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.

## 14.6 Operation

### 14.6.1 Data format

Full-duplex serial data reception and transmission is performed.

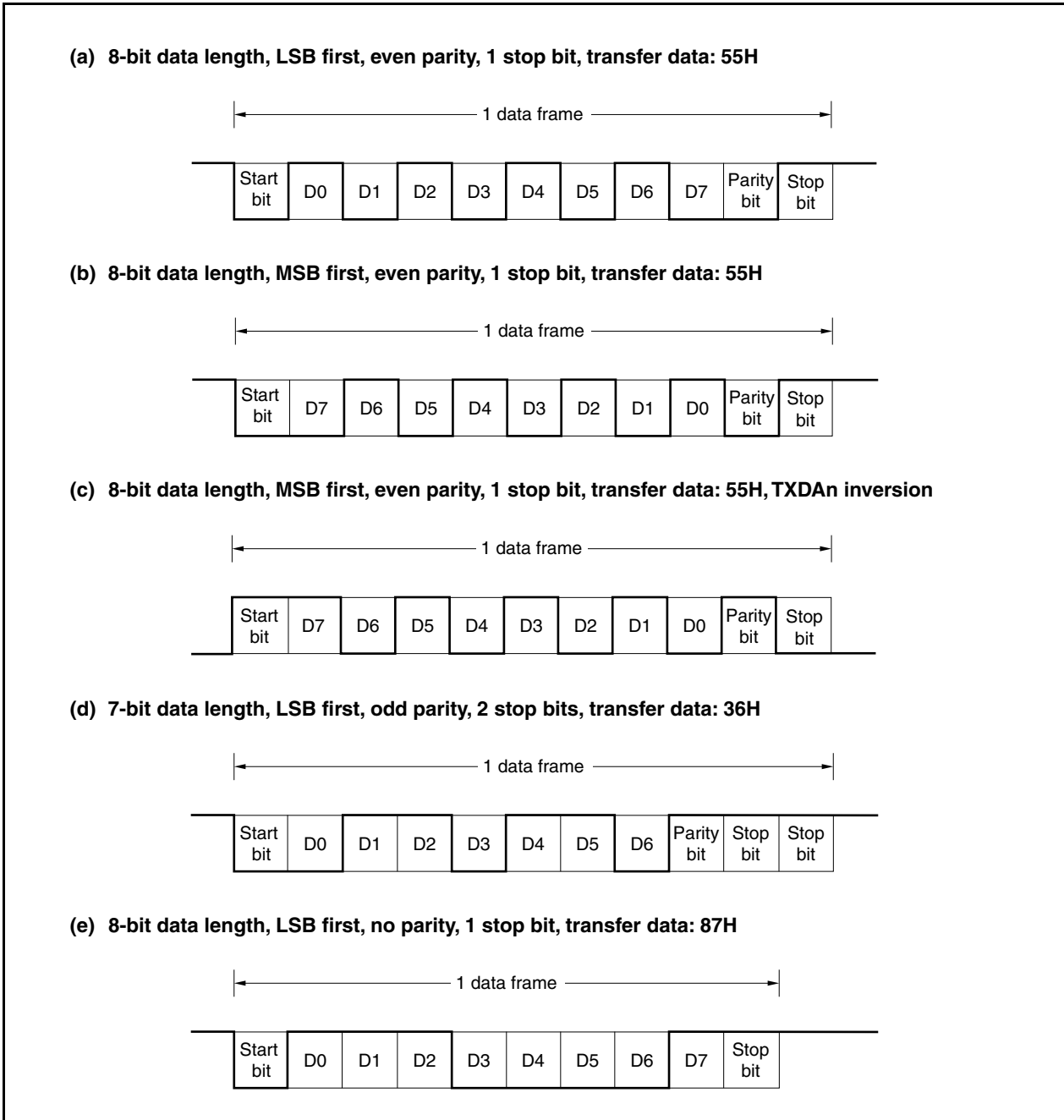
As shown in Figure 14-5, one data frame of transmit/receive data consists of a start bit, character bits, parity bit, and stop bit(s).

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 UARTAn output/inverted output for the TXDAn pin is performed using the UAnOPT0.UAnTDL bit.

- Start bit ..... 1 bit
- Character bits..... 7 bits/8 bits
- Parity bit ..... Even parity/odd parity/0 parity/no parity
- Stop bit..... 1 bit/2 bits

Figure 14-5. UARTA Transmit/Receive Data Format



**14.6.2 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.

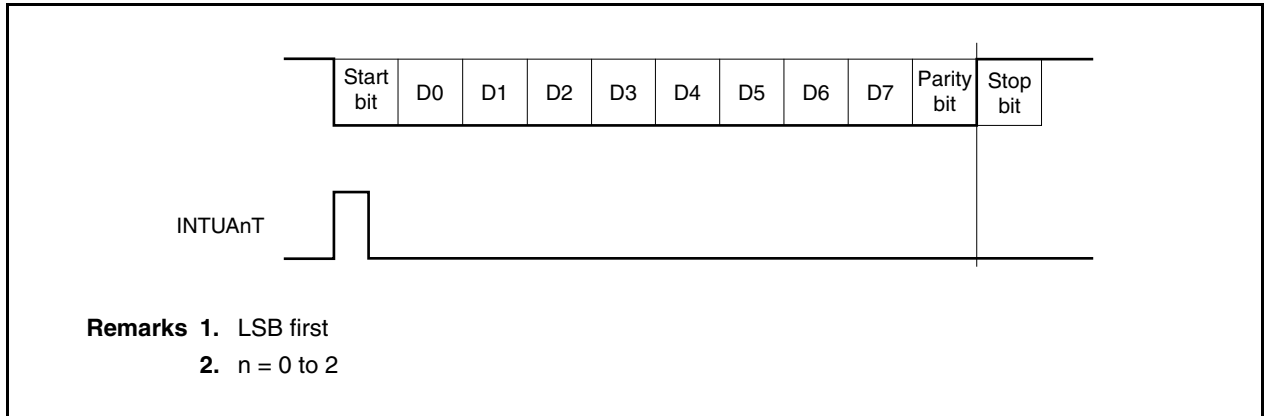
Since the CTS (transmit enable signal) input pin is not provided in UARTAn, use a port to check that reception is enabled at the transmit destination.

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 end of transmission of the data of the UAnTX register to the UARTAn transmit shift register, and thereafter the contents of the UARTAn transmit shift register are output to the TXDAn pin.

Write of the next transmit data to the UAnTX register is enabled by generating the INTUAnT signal.

**Figure 14-6. UART Transmission**



**14.6.3 Continuous transmission procedure**

UARTAn can write the next transmit data to the UAnTX register when the UARTAn transmit shift register starts the shift operation. The transmit timing of the UARTAn transmit shift register can be judged from the transmission enable interrupt request signal (INTUAnT). An efficient communication rate is realized by writing the data to be transmitted next to the UAnTX register during transfer.

**Caution** During continuous transmission execution, perform initialization after checking that the UAnSTR.UAnTSF bit is 0. The transmit data cannot be guaranteed when initialization is performed while the UAnTSF bit is 1.

**Remark** n = 0 to 2

**Figure 14-7. Continuous Transmission Processing Flow**

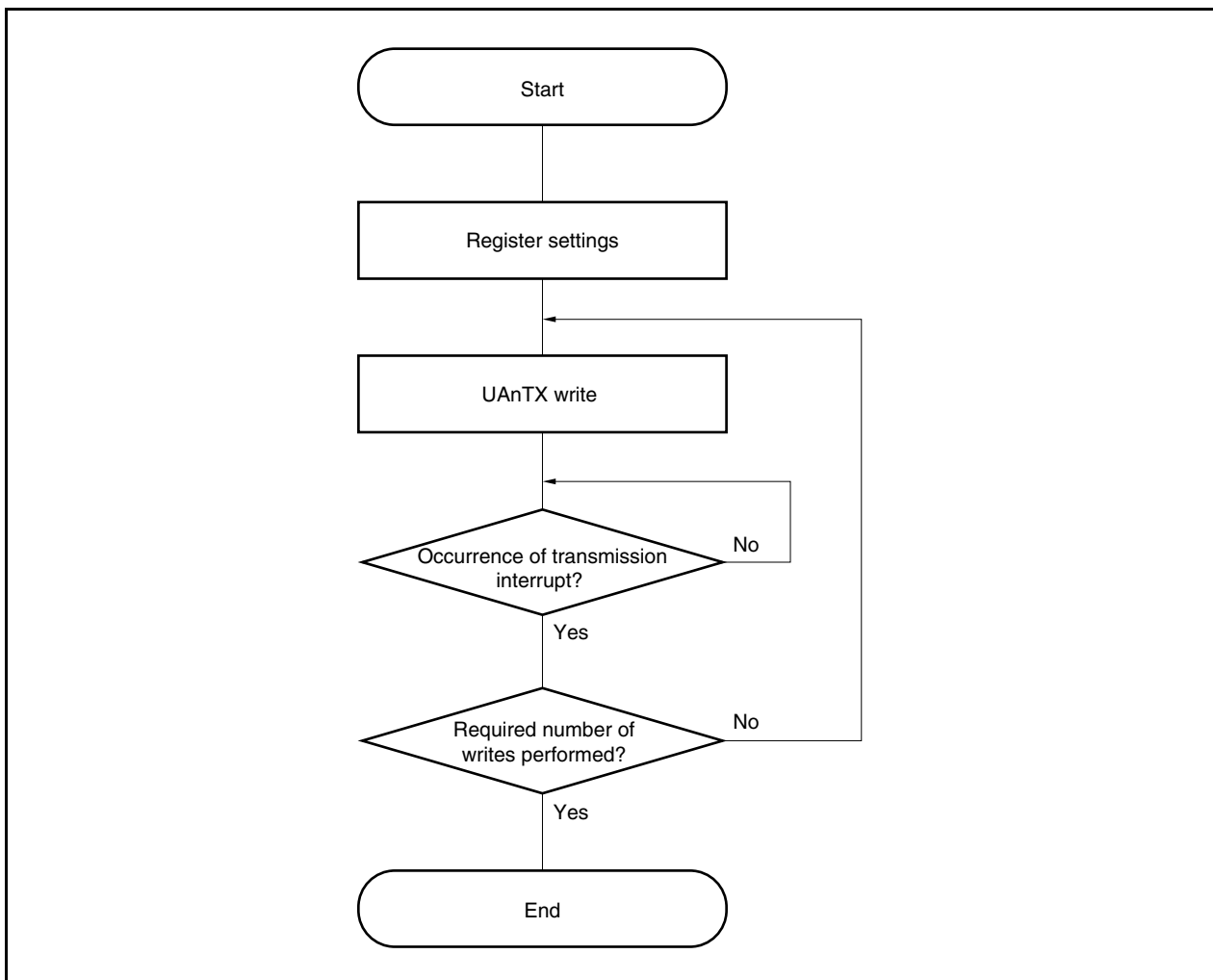
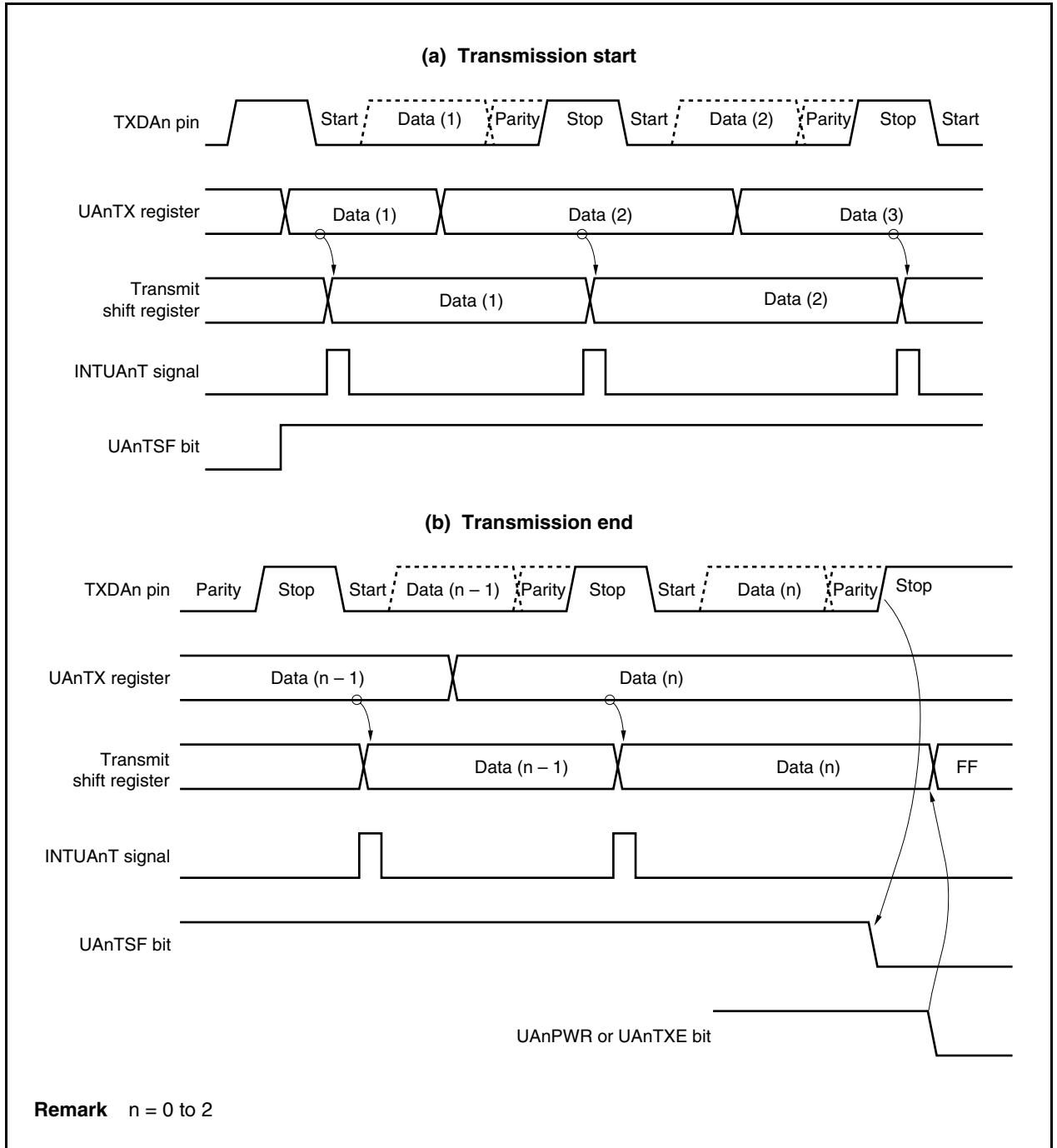


Figure 14-8. Continuous Transmission Operation Timing



**14.6.4 UART reception**

The reception wait status is set by setting the UAnCTL0.UAnPWR bit to 1 and then setting the UAnCTL0.UAnRXE bit to 1. In the reception wait status, the RXDAn pin is monitored and start bit detection is performed.

Start bit detection is performed using a two-step detection routine.

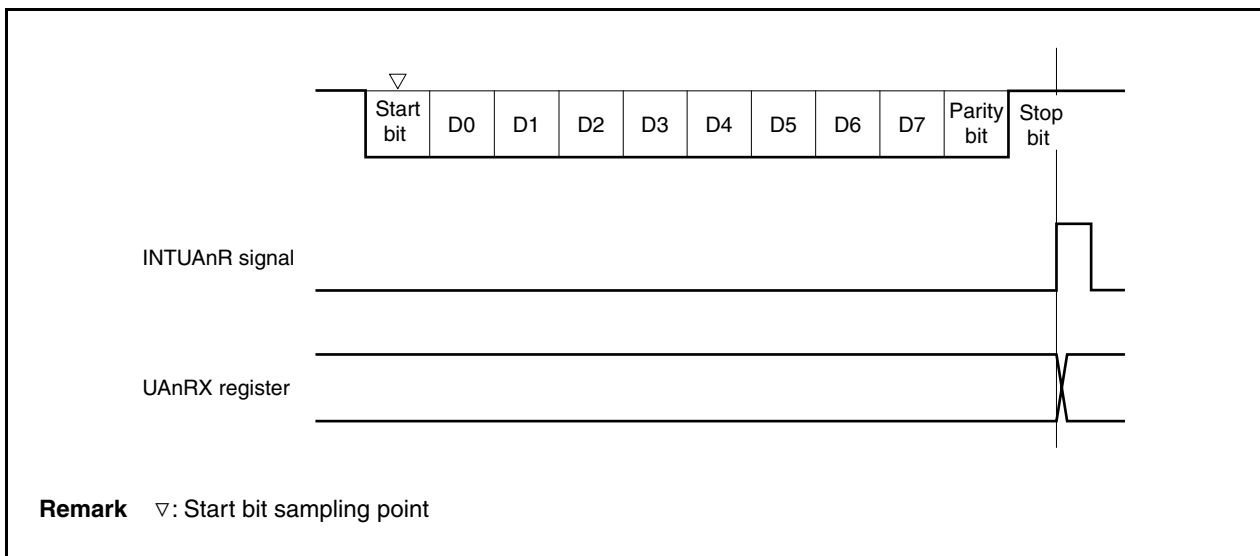
First the falling edge of the RXDAn pin is detected and sampling is started at the falling edge. The start bit is recognized if the RXDAn pin is low level at the start bit sampling point. After a start bit has been recognized, the receive operation starts, and serial data is saved to the UARTAn receive shift register according to the set baud rate.

When the reception end 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 occurs (UAnSTR.UAnOVE bit = 1), the receive data at this time is not written to the UAnRX register and is discarded.

Even if a parity error (UAnSTR.UAnPE bit = 1) or a framing error (UAnSTR.UAnFE bit = 1) occurs during reception, reception continues until the reception position of the first stop bit, and the INTUAnRE signal is output following reception end.

**Remark** n = 0 to 2

**Figure 14-9. 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 end 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 end 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 end reception without waiting INTUAnR signal generation, be sure to clear (0) the interrupt request flag (UAnRIC.UAnRIF), after setting (1) the interrupt mask flag (UAnRIC.UAnRMK) and then set (1) the UAnPWR bit = 0 or UAnRXE bit = 0.



### 14.6.5 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 error interrupt request signal (INTUAnRE) 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.

**Caution** The reception end interrupt request signal (INTUAnR) and reception error interrupt request signal (INTUAnRE) are not generated simultaneously. The INTUAnR signal is generated when a reception ends normally. The INTUAnRE signal is generated and the INTUAnR signal is not generated when a reception error occurs.

**Remark** n = 0 to 2

- 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 ended before data was read from UAnRX register |

**14.6.6 Parity types and operations**

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.

**14.6.7 Receive data noise filter**

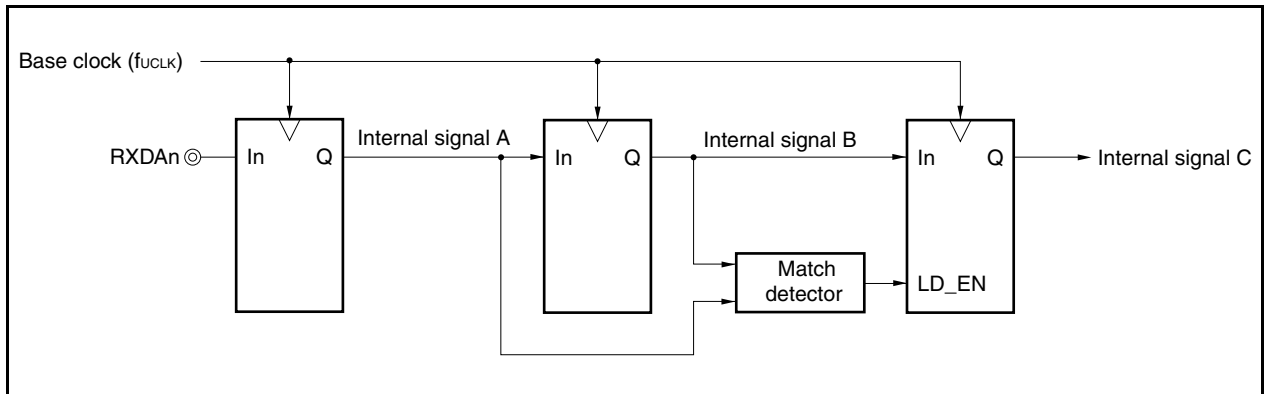
This filter samples signals received via the RXDAn pin using the base clock ( $f_{CLK}$ ) supplied by the dedicated baud rate generator.

<R> 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 1 clock cycle width is judged to be noise and is not delivered to the internal circuit (see **Figure 14-11**). See **14.7 (1) (a) Base clock** regarding the base clock.

Moreover, since the circuit is as shown in Figure 14-10, the processing that goes on within the receive operation is delayed by 3 clocks in relation to the external signal status.

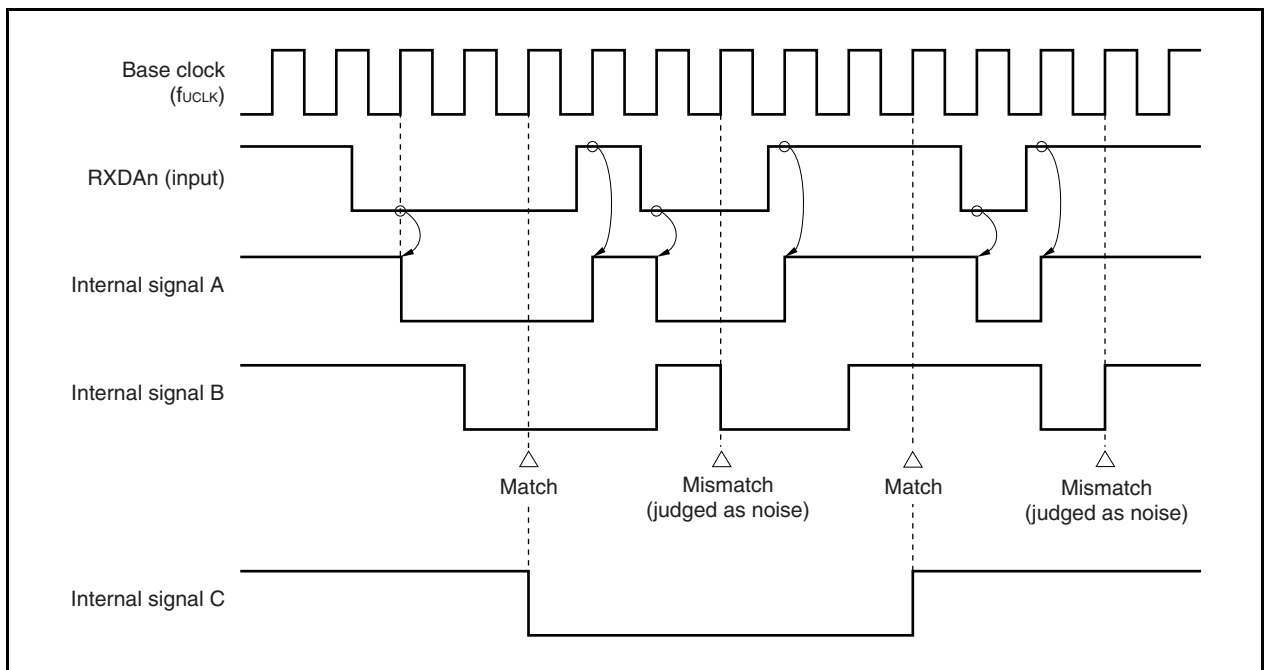
**Remark**  $n = 0$  to 2

**Figure 14-10. Noise Filter Circuit**



<R>

**Figure 14-11. Timing of RXDAn Signal Judged as Noise**



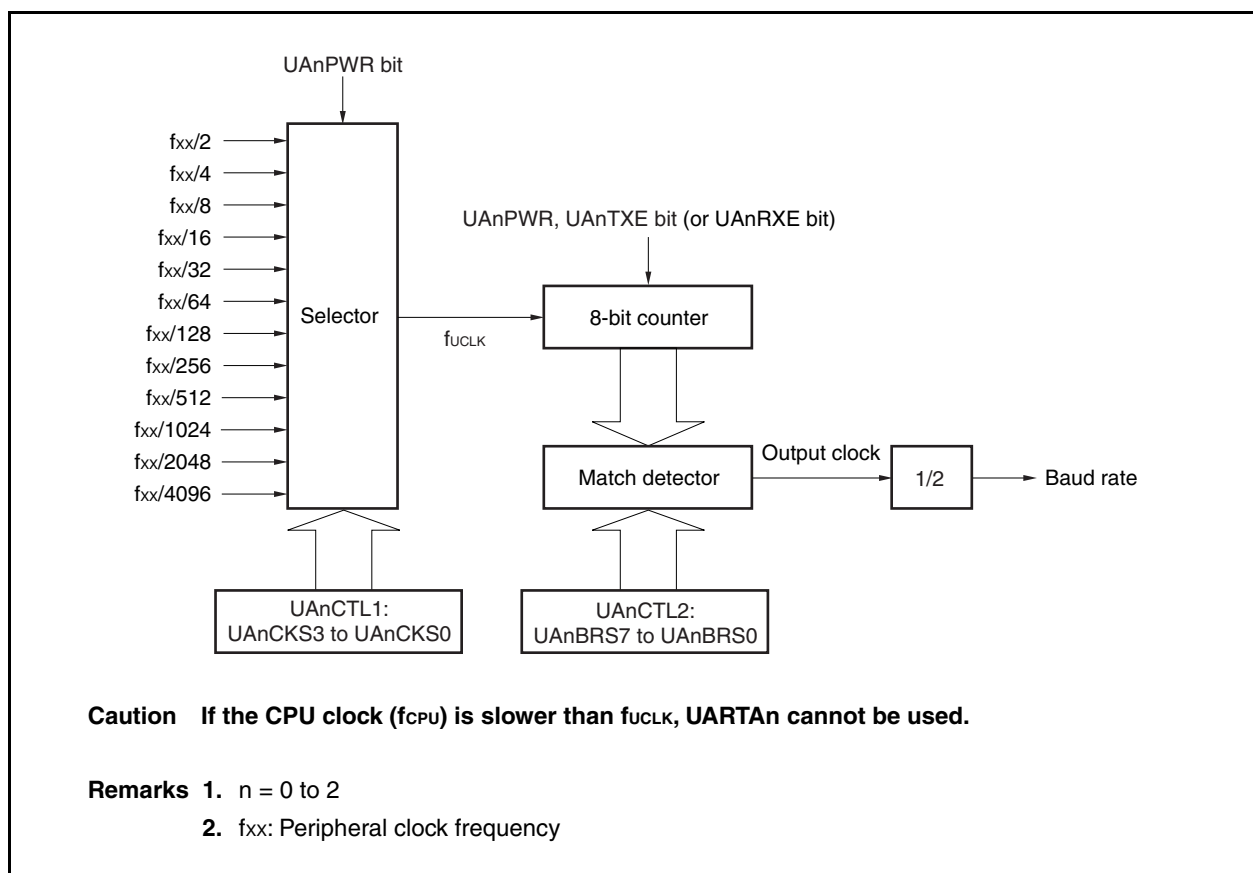
## 14.7 Dedicated Baud Rate Generator

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 14-12. 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 ( $f_{UCLK}$ ). When the UAnPWR bit = 0,  $f_{UCLK}$  is fixed to the low level.

#### (b) Serial clock generation

A serial clock can be generated by setting the UAnCTL1 register and the UAnCTL2 register.

The base clock ( $f_{UCLK}$ ) is selected by the 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 or 1-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 FFFFA01H, UA1CTL1 FFFFA11H,  
UA2CTL1 FFFFA21H

|                         |   |   |   |   |         |         |         |         |
|-------------------------|---|---|---|---|---------|---------|---------|---------|
|                         | 7 | 6 | 5 | 4 | 3       | 2       | 1       | 0       |
| UAnCTL1<br>(n = 0 to 2) | 0 | 0 | 0 | 0 | UAnCKS3 | UAnCKS2 | UAnCKS1 | UAnCKS0 |

| UAnCKS3          | UAnCKS2 | UAnCKS1 | UAnCKS0 | Base clock (f <sub>CLK</sub> ) selection |
|------------------|---------|---------|---------|--|
| 0                | 0       | 0       | 0       | fxx/2                                    |
| 0                | 0       | 0       | 1       | fxx/4                                    |
| 0                | 0       | 1       | 0       | fxx/8                                    |
| 0                | 0       | 1       | 1       | fxx/16                                   |
| 0                | 1       | 0       | 0       | fxx/32                                   |
| 0                | 1       | 0       | 1       | fxx/64                                   |
| 0                | 1       | 1       | 0       | fxx/128                                  |
| 0                | 1       | 1       | 1       | fxx/256                                  |
| 1                | 0       | 0       | 0       | fxx/512                                  |
| 1                | 0       | 0       | 1       | fxx/1,024                                |
| 1                | 0       | 1       | 0       | fxx/2,048                                |
| 1                | 0       | 1       | 1       | fxx/4,096                                |
| Other than above |         |         |         | Setting prohibited                       |

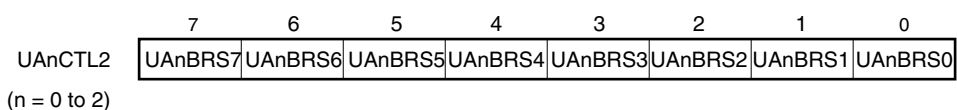
**Remark** fxx: Peripheral clock frequency

**(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 FFFFFFFA02H, UA1CTL2 FFFFFFFA12H,  
UA2CTL2 FFFFFFFA22H



| 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 <sub>UCLK</sub> /4   |
| 0        | 0        | 0        | 0        | 0        | 1        | 0        | 1        | 5           | f <sub>UCLK</sub> /5   |
| 0        | 0        | 0        | 0        | 0        | 1        | 1        | 0        | 6           | f <sub>UCLK</sub> /6   |
| :        | :        | :        | :        | :        | :        | :        | :        | :           | :                      |
| 1        | 1        | 1        | 1        | 1        | 1        | 0        | 0        | 252         | f <sub>UCLK</sub> /252 |
| 1        | 1        | 1        | 1        | 1        | 1        | 0        | 1        | 253         | f <sub>UCLK</sub> /253 |
| 1        | 1        | 1        | 1        | 1        | 1        | 1        | 0        | 254         | f <sub>UCLK</sub> /254 |
| 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 255         | f <sub>UCLK</sub> /255 |

**Remark** f<sub>UCLK</sub>: Frequency of base clock selected by the UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits

**(4) Baud rate**

The baud rate is obtained by the following equation.

$$\text{Baud rate} = \frac{f_{\text{CLK}}}{2 \times k} \text{ [bps]}$$

$f_{\text{CLK}}$ : Frequency of base clock selected by the UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits

$k$ : Value set using the UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits ( $k = 4, 5, 6, \dots, 255$ )

**(5) Baud rate error**

The baud rate error is obtained by the following equation.

$$\text{Error (\%)} = \left( \frac{\text{Actual baud rate (baud rate with error)}}{\text{Target baud rate (correct baud rate)}} - 1 \right) \times 100 \text{ [\%]}$$

**Cautions 1. The baud rate error during transmission must be within the error tolerance on the receiving side.**

**2. The baud rate error during reception must satisfy the range indicated in section (7) Allowable baud rate range during reception.**

**Example** Peripheral clock frequency = 32 MHz = 32,000,000 Hz

Set value of UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits = 0000B ( $f_{\text{CLK}} = 16,000,000$  Hz)

Set value of UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits = 00110100B ( $k = 52$ )

Target baud rate = 153,600

$$\text{Baud rate} = 16,000,000 / (2 \times 52) = 153,846 \text{ [bps]}$$

$$\begin{aligned} \text{Error} &= (153,846/153,600 - 1) \times 100 \\ &= 0.160 \text{ [\%]} \end{aligned}$$

(6) Baud rate setting example

Table 14-3. Baud Rate Generator Setting Data

| Baud Rate<br>(bps) | f <sub>xx</sub> = 64 MHz |         |         | f <sub>xx</sub> = 32 MHz |         |         |
|--------------------|--------------------------|---------|---------|--------------------------|---------|---------|
|                    | UAnCTL1                  | UAnCTL2 | ERR (%) | UAnCTL1                  | UAnCTL2 | ERR (%) |
| 300                | 08H                      | D0H     | 0.16    | 07H                      | D0H     | 0.16    |
| 600                | 07H                      | D0H     | 0.16    | 06H                      | D0H     | 0.16    |
| 1,200              | 06H                      | D0H     | 0.16    | 05H                      | D0H     | 0.16    |
| 2,400              | 05H                      | D0H     | 0.16    | 04H                      | D0H     | 0.16    |
| 4,800              | 04H                      | D0H     | 0.16    | 03H                      | D0H     | 0.16    |
| 9,600              | 03H                      | D0H     | 0.16    | 02H                      | D0H     | 0.16    |
| 19,200             | 02H                      | D0H     | 0.16    | 01H                      | D0H     | 0.16    |
| 31,250             | 02H                      | 80H     | 0       | 00H                      | 80H     | 0       |
| 38,400             | 01H                      | D0H     | 0.16    | 00H                      | D0H     | 0.16    |
| 76,800             | 00H                      | D0H     | 0.16    | 00H                      | 68H     | 0.16    |
| 153,600            | 00H                      | 68H     | 0.16    | 00H                      | 34H     | 0.16    |
| 312,500            | 00H                      | 33H     | 0.39    | 00H                      | 1AH     | -1.54   |
| 625,000            | 00H                      | 1AH     | -1.54   | 00H                      | 0DH     | -1.54   |
| 1,250,000          | 00H                      | 0DH     | -1.54   | 00H                      | 06H     | 6.67    |

**Remark** f<sub>xx</sub>: Peripheral clock frequency  
 ERR: Baud rate error (%)

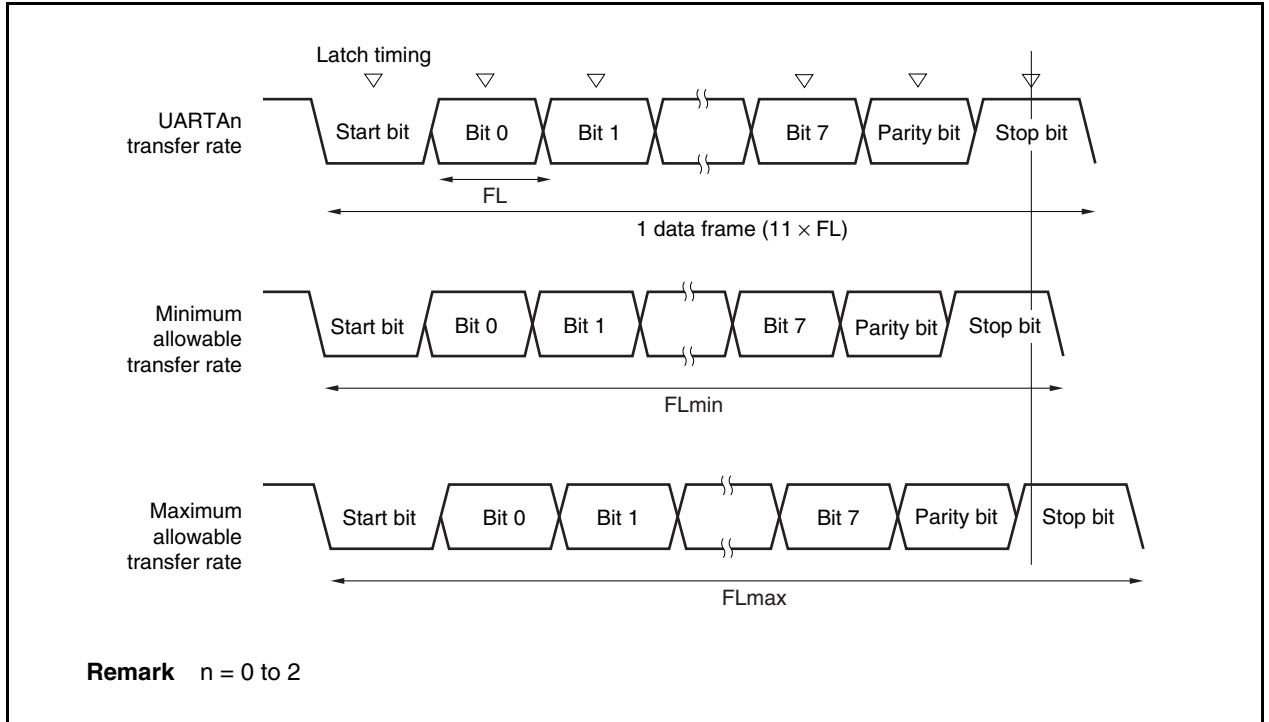


**(7) Allowable baud rate range during reception**

The baud rate error range at the destination that is allowable during reception is shown below.

**Caution** The baud rate error during reception must be set within the allowable error range using the following equation.

**Figure 14-13. Allowable Baud Rate Range During Reception**



As shown in Figure 14-13, 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: UARTA baud rate (n = 0 to 2)

k: Set value of UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits (n = 0 to 2)

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 14-4. Maximum/Minimum Allowable Baud Rate Error**

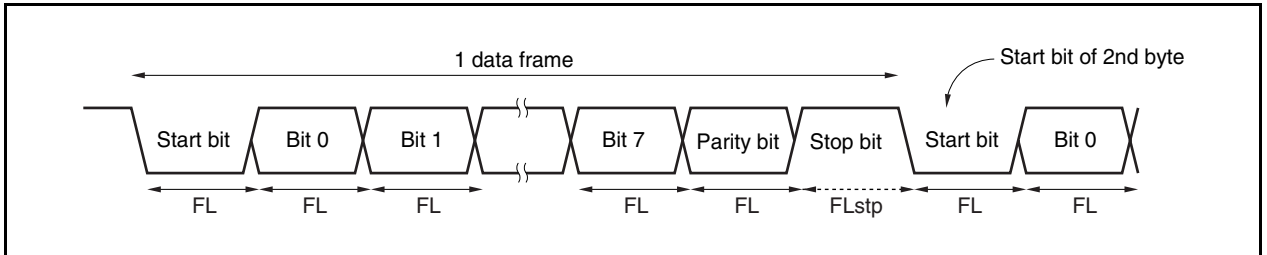
| Division Ratio (k) | Maximum Allowable Baud Rate Error | Minimum Allowable Baud Rate Error |
|--------------------|-----------------------------------|-----------------------------------|
| 4                  | +2.32%                            | -2.43%                            |
| 8                  | +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: Set value of UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits (n = 0 to 2)

**(8) Transfer rate during continuous transmission**

During continuous transmission, the transfer rate from the stop bit to the next start bit is usually 2 base clocks longer. However, timing initialization is performed via start bit detection by the receiving side, so this has no influence on the transfer result.

**Figure 14-14. Transfer Rate During Continuous Transmission**



Assuming 1 bit data length: FL; stop bit length: FLstp; and base clock frequency:  $f_{UCLK}$ , we obtain the following equation.

$$FL_{stp} = FL + 2/f_{UCLK}$$

Therefore, the transfer rate during continuous transmission is as follows.

$$\text{Transfer rate} = 11 \times FL + (2/f_{UCLK})$$

**14.8 Cautions**

When the clock supply to UARTAn is stopped (for example, in IDLE 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.UAnRXE, and UAnCTL0.UAnTXE bits to 000.

**Remark** n = 0 to 2

## CHAPTER 15 ASYNCHRONOUS SERIAL INTERFACE B (UARTB)

### 15.1 Mode Switching Between UARTB and CSIB2

In the V850E/IF3 and V850E/IG3, UARTB and CSIB2 function alternately, and these pins cannot be used at the same time. To switch between UARTB and CSIB2, the PMC3, PFC3, and PFCE3 registers must be set in advance.

**Caution** The operations related to transmission and reception of UARTB or CSIB2 are not guaranteed if the mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 15-1. Mode Switch Settings of UARTB and CSIB2

| After reset: 00H | R/W   | Address: FFFFF446H  |  |        |        |  |        |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
|------------------|---|---|--|--------|--------|--|--------|---|---|----------|--------|--------|--------|-------------|-------|--------|--------|--------------|---|---|---|-----------------------------|---|---|---|--------------------|
| PMC3             |   | <table border="1"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>PMC37</td><td>PMC36</td><td>PMC35</td><td>PMC34</td><td>PMC33</td><td>PMC32</td><td>PMC31</td><td>PMC30</td> </tr> </table>   | 7  | 6      | 5      | 4  | 3      | 2 | 1 | 0        | PMC37  | PMC36  | PMC35  | PMC34       | PMC33 | PMC32  | PMC31  | PMC30        |   |   |   |                             |   |   |   |                    |
| 7                | 6   | 5   | 4  | 3      | 2      | 1  | 0      |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
| PMC37            | PMC36                                       | PMC35   | PMC34  | PMC33  | PMC32  | PMC31  | PMC30  |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
| After reset: 00H | R/W   | Address: FFFFF466H  |  |        |        |  |        |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
| PFC3             |   | <table border="1"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>PFC37</td><td>PFC36</td><td>PFC35</td><td>PFC34</td><td>PFC33</td><td>PFC32</td><td>PFC31</td><td>PFC30</td> </tr> </table>   | 7  | 6      | 5      | 4  | 3      | 2 | 1 | 0        | PFC37  | PFC36  | PFC35  | PFC34       | PFC33 | PFC32  | PFC31  | PFC30        |   |   |   |                             |   |   |   |                    |
| 7                | 6   | 5   | 4  | 3      | 2      | 1  | 0      |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
| PFC37            | PFC36                                       | PFC35   | PFC34  | PFC33  | PFC32  | PFC31  | PFC30  |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
| After reset: 00H | R/W   | Address: FFFFF706H  |  |        |        |  |        |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
| PFCE3            |   | <table border="1"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>PFCE37</td><td>PFCE36</td><td>PFCE35</td><td>PFCE34</td><td>0</td><td>PFCE32</td><td>PFCE31</td><td>PFCE30</td> </tr> </table>  | 7  | 6      | 5      | 4  | 3      | 2 | 1 | 0        | PFCE37 | PFCE36 | PFCE35 | PFCE34      | 0     | PFCE32 | PFCE31 | PFCE30       |   |   |   |                             |   |   |   |                    |
| 7                | 6   | 5   | 4  | 3      | 2      | 1  | 0      |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
| PFCE37           | PFCE36                                      | PFCE35  | PFCE34   | 0      | PFCE32 | PFCE31   | PFCE30 |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
|                  |   | <table border="1"> <thead> <tr> <th>PMC37</th><th>PFCE37</th><th>PFC37</th><th>Specification of alternate function of P37 pin</th></tr> </thead> <tbody> <tr> <td>0</td><td>x</td><td>x</td><td>I/O port</td></tr> <tr> <td>1</td><td>0</td><td>0</td><td>SCKB2 I/O</td></tr> <tr> <td>1</td><td>0</td><td>1</td><td>INTP12 input</td></tr> <tr> <td>1</td><td>1</td><td>0</td><td>ASTB<sup>Note</sup> output</td></tr> <tr> <td>1</td><td>1</td><td>1</td><td>Setting prohibited</td></tr> </tbody> </table> | PMC37  | PFCE37 | PFC37  | Specification of alternate function of P37 pin | 0      | x | x | I/O port | 1      | 0      | 0      | SCKB2 I/O   | 1     | 0      | 1      | INTP12 input | 1 | 1 | 0 | ASTB <sup>Note</sup> output | 1 | 1 | 1 | Setting prohibited |
| PMC37            | PFCE37                                      | PFC37   | Specification of alternate function of P37 pin |        |        |  |        |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
| 0                | x   | x   | I/O port                                       |        |        |  |        |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
| 1                | 0   | 0   | SCKB2 I/O                                      |        |        |  |        |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
| 1                | 0   | 1   | INTP12 input                                   |        |        |  |        |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
| 1                | 1   | 0   | ASTB <sup>Note</sup> output                    |        |        |  |        |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
| 1                | 1   | 1   | Setting prohibited                             |        |        |  |        |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
|                  |   | <table border="1"> <thead> <tr> <th>PMC36</th><th>PFCE36</th><th>PFC36</th><th>Specification of alternate function of P36 pin</th></tr> </thead> <tbody> <tr> <td>0</td><td>x</td><td>x</td><td>I/O port</td></tr> <tr> <td>1</td><td>0</td><td>0</td><td>SOB2 output</td></tr> <tr> <td>1</td><td>0</td><td>1</td><td>TXDB output</td></tr> <tr> <td>1</td><td>1</td><td>0</td><td>Setting prohibited</td></tr> <tr> <td>1</td><td>1</td><td>1</td><td>Setting prohibited</td></tr> </tbody> </table>        | PMC36  | PFCE36 | PFC36  | Specification of alternate function of P36 pin | 0      | x | x | I/O port | 1      | 0      | 0      | SOB2 output | 1     | 0      | 1      | TXDB output  | 1 | 1 | 0 | Setting prohibited          | 1 | 1 | 1 | Setting prohibited |
| PMC36            | PFCE36                                      | PFC36   | Specification of alternate function of P36 pin |        |        |  |        |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
| 0                | x   | x   | I/O port                                       |        |        |  |        |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
| 1                | 0   | 0   | SOB2 output                                    |        |        |  |        |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
| 1                | 0   | 1   | TXDB output                                    |        |        |  |        |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
| 1                | 1   | 0   | Setting prohibited                             |        |        |  |        |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
| 1                | 1   | 1   | Setting prohibited                             |        |        |  |        |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
|                  |   | <table border="1"> <thead> <tr> <th>PMC35</th><th>PFCE35</th><th>PFC35</th><th>Specification of alternate function of P35 pin</th></tr> </thead> <tbody> <tr> <td>0</td><td>x</td><td>x</td><td>I/O port</td></tr> <tr> <td>1</td><td>0</td><td>0</td><td>SIB2 input</td></tr> <tr> <td>1</td><td>0</td><td>1</td><td>RXDB input</td></tr> <tr> <td>1</td><td>1</td><td>0</td><td>Setting prohibited</td></tr> <tr> <td>1</td><td>1</td><td>1</td><td>Setting prohibited</td></tr> </tbody> </table>          | PMC35  | PFCE35 | PFC35  | Specification of alternate function of P35 pin | 0      | x | x | I/O port | 1      | 0      | 0      | SIB2 input  | 1     | 0      | 1      | RXDB input   | 1 | 1 | 0 | Setting prohibited          | 1 | 1 | 1 | Setting prohibited |
| PMC35            | PFCE35                                      | PFC35   | Specification of alternate function of P35 pin |        |        |  |        |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
| 0                | x   | x   | I/O port                                       |        |        |  |        |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
| 1                | 0   | 0   | SIB2 input                                     |        |        |  |        |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
| 1                | 0   | 1   | RXDB input                                     |        |        |  |        |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
| 1                | 1   | 0   | Setting prohibited                             |        |        |  |        |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
| 1                | 1   | 1   | Setting prohibited                             |        |        |  |        |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
| <b>Note</b>      | μPD70F3454GC-8EA-A and 70F3454F1-DA9-A only |   |  |        |        |  |        |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |
| <b>Remark</b>    | x = don't care                              |   |  |        |        |  |        |   |   |          |        |        |        |             |       |        |        |              |   |   |   |                             |   |   |   |                    |

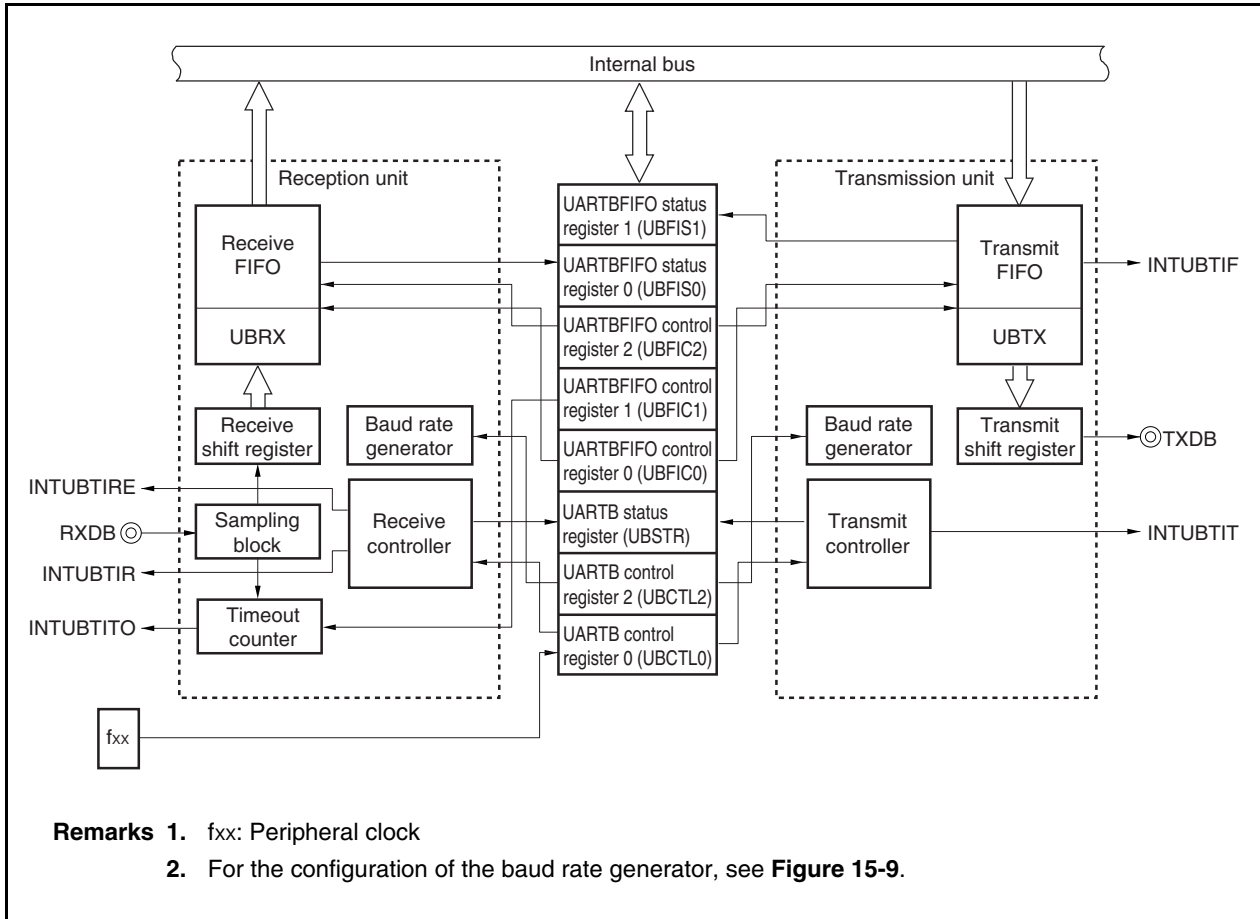
## 15.2 Features

- Transfer rate: Maximum 5.33 Mbps (using a dedicated baud rate generator)
- Full-duplex communications
- Single mode and FIFO mode selectable
  - Single mode: 8-bit × 1-stage data register (UBTX register or UBRX register) is used for each of transmission and reception.
  - FIFO mode
    - Transmit FIFO: UBTX register (8 bits × 16 stages).
    - Receive FIFO: UBRXAP register (16 bits × 16 stages)  
2 bits of the higher 8 bits of the UBRXAP register are for an error flag.
- Two-pin configuration
  - TXDB: Transmit data output pin
  - RXDB: Receive data input pin
- Reception error detection function
  - Overflow error (FIFO mode only)
  - Parity error
  - Framing error
  - Overrun error (single mode only)
- Interrupt sources: 5 types
  - Reception error interrupt request signal (INTUBTIRE)
  - Reception end interrupt request signal (INTUBTIR)
  - Transmission enable interrupt request signal (INTUBTIT)
  - FIFO transmission end interrupt request signal (INTUBTIF) (FIFO mode only)
  - Reception timeout interrupt request signal (INTUBTITO) (FIFO mode only)
- The character length of transmit/receive data is specified according to the UBCTL0 register
- Character length: 7 or 8 bits
- Parity functions: Odd, even, 0, or none
- Transmission stop bits: 1 or 2 bits
- MSB first/LSB first selectable for transfer data
- On-chip dedicated baud rate generator

### 15.3 Configuration

The block diagram of the UARTB is shown below.

Figure 15-2. Block Diagram of UARTB



UARTB consists of the following hardware units.

**Table 15-1. Configuration of UARTB**

| Item      | Configuration  |
|-----------|--|
| Registers | UARTB control register 0 (UBCTL0)<br>UARTB control register 2 (UBCTL2)<br>UARTB status register (UBSTR)<br>UARTB FIFO control register 0 (UBFIC0)<br>UARTB FIFO control register 1 (UBFIC1)<br>UARTB FIFO control register 2 (UBFIC2)<br>UARTB FIFO status register 0 (UBFIS0)<br>UARTB FIFO status register 1 (UBFIS1)<br>Receive shift register<br>UARTB receive data register AP (UBRXAP)<br>UARTB receive data register (UBRX)<br>Transmit shift register<br>UARTB transmit data register (UBTX) |

**(1) UARTB control register 0 (UBCTL0)**

This register controls the transfer operation of UARTB.

**(2) UARTB status register (UBSTR)**

This register indicates the transfer status during transmission and the contents of a reception error. The status flag of this register, which indicates the transfer status during transmission, indicates the data retention status of the transmit shift register and the transmit data register (the UBTX register in the single mode or transmit FIFO in the FIFO mode). Each reception error flag is set to 1 when a reception error occurs, and cleared to 0 when 0 is written to the UBSTR register.

**(3) UARTB control register 2 (UBCTL2)**

This register is used to specify the division ratio by which to control the baud rate (serial transfer speed) of UARTB.

**(4) UARTB FIFO control register 0 (UBFIC0)**

This register is used to select the operation mode of UARTB, clear the transmit FIFO/receive FIFO that becomes valid in the FIFO mode, and specify the timing mode in which the transmission enable interrupt request signal (INTUBTIT)/reception end interrupt request signal (INTUBTIR) occurs.

**(5) UARTB FIFO control register 1 (UBFIC1)**

This register is valid in the FIFO mode. It generates a reception timeout interrupt request signal (INTUBTITO) if data is stored in the receive FIFO when the next data does not come (start bit is not detected) even after the reception wait time of the next data has elapsed after the stop bit has been received.

**(6) UARTB FIFO control register 2 (UBFIC2)**

This register is valid in the FIFO mode. It is used to set the timing to generate the transmission enable interrupt request signal (INTUBTIT)/reception end interrupt request signal (INTUBTIR), using the number of data transmitted or received as a trigger.

**(7) UARTB FIFO status register 0 (UBFIS0)**

This register is valid in the FIFO mode. The number of bytes of data stored in the receive FIFO can be read from this register.

**(8) UARTB FIFO status register 1 (UBFIS1)**

This register is valid in the FIFO mode. The number of empty bytes of the transmit FIFO can be read from this register.

**(9) Receive shift register**

This is a shift register that converts the serial data that was input to the RXDB pin into parallel data. One byte of data is received, and if a stop bit is detected, the received data is transferred to the receive data register.

This register cannot be directly manipulated.

**(10) UARTB receive data register AP (UBRXAP), UARTB receive data register (UBRX)**

The receive data register holds receive data. In the single mode, the 8-bit  $\times$  1-stage UBRX register is used. The 16-bit  $\times$  16-stage receive FIFO (UBRXAP register) is used in the FIFO mode.

The receive data is stored in the lower 8 bits of the receive FIFO (UBRXAP register) and the error information of the received data is stored in the higher 8 bits (bit 8 and bit 9). If a reception error (such as a parity error or a framing error) occurs in the FIFO mode, the error data can be identified by reading the UBRXAP register in 16-bit (halfword) units (error information is appended as UBPEF bit = 1 or UBFEF bit = 1). When the lower 8 bits of the UBRXAP register are read in 8-bit (byte) units, the higher 8 bits are discarded. Therefore, if no error has occurred, only the receive data of the UBRXAP register can be read successively by being read in 8-bit (byte) units in the same way as the UBRX register.

When 7-bit length data is received with the LSB first, the received data is transferred to bits 6 to 0 of the receive data register from the LSB (bit 0), with the MSB (bit 7) always being 0. When data is received with the MSB first, the received data is transferred to bits 7 to 1 of the receive data register from the MSB (bit 7), with the LSB (bit 0) always being 0. If an overrun error occurs, the receive data at that time is not transferred to the receive data register.

While reception is enabled, the received data is transferred from the receive shift register to the receive data register, in synchronization with the shift-in processing of one frame.

A reception end interrupt request signal (INTUBTIR) is generated by transferring the data to the UBRX register in the single mode, or transferring the number of receive data set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits to receive FIFO in the FIFO mode. If data is stored in receive FIFO when the next data does not come (start bit is not detected) after the next data reception wait time specified by the UBFIC1.UBTC4 to UBFIC1.UBTC0 bits has elapsed in the FIFO mode, a reception timeout interrupt request signal (INTUBTITO) is generated.

**(11) Transmit shift register**

This is a shift register that converts the parallel data that was transferred from the transmit data register into serial data.

When one byte of data is transferred from the transmit data register, the transmit shift register data is output from the TXDB pin.

This register cannot be directly manipulated.



**(12) UARTB transmit data register (UBTX)**

The transmit data register is a buffer for transmit data. The 8-bit × 1-stage UBTX register is used as this buffer in the single mode. In the FIFO mode, the 8-bit × 16-stage transmit FIFO is used.

When 7-bit length data is transmitted with the LSB first, bits 6 to 0 of the transmit data register are transmitted as the transmit data from the LSB (bit 0) with the MSB (bit 7) always being 0. When data is transmitted with the MSB first, bits 7 to 1 of the transmit data register are transmitted as the transmit data from the MSB (bit 7) with the LSB (bit 0) always being 0.

In the single mode, transmission is started by writing transmit data to the UBTX register while transmission is enabled (UBCTL0.UBTXE bit = 1). When writing the transmit data to the UBTX register is enabled (when 1-byte data is transferred from the UBTX register to the transmit shift register), a transmission enable interrupt request signal (INTUBTIT) is generated.

In the FIFO mode, transmission is started by writing at least the number of transmit data set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits and 16 bytes or less to transmit FIFO and then enabling transmission (UBTXE bit = 1). When the number of transmit data set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits have been transferred from transmit FIFO to the transmit shift register (transmit data of the number set as the trigger can be written), a transmission enable interrupt request signal (INTUBTIT) is generated. In the FIFO mode, a FIFO transmission enable interrupt request signal (INTUBTIF) is generated when there is no more data in transmit FIFO and the transmit shift register (when FIFO and the register become empty).

**(13) Timeout counter**

This counter is used to recognize that data exists (remains) in receive FIFO when the number of received data does not reach the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits, and is valid only in the FIFO mode.

If data is stored in receive FIFO when the next data does not come (start bit is not detected) after the next data reception wait time specified by the UBFIC1.UBTC4 to UBFIC1.UBTC0 bits has elapsed after the stop bit has been received, a reception timeout interrupt request signal (INTUBTITO) is generated.

**(14) Sampling block**

This block samples the RXDB signal at the rising edge of the input clock ( $f_{xx}$ ). If the same sampling value is detected two times, output of the match detector changes, and the value is sampled as input data. Data of less than one clock width is judged as noise and is not transmitted to the internal circuitry.

## 15.4 Control Registers

### (1) UARTB control register 0 (UBCTL0)

The UBCTL0 register controls the transfer operations of UARTB.

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

Reset sets this register to 10H.

**Cautions** 1. When using UARTB, set the external pins related to the UARTB function in the alternate-function mode, set UARTB control register 2 (UBCTL2). Then set the UBPWR bit to 1 before setting the other bits.

2. Be sure to input a high level to the RXDB pin when setting the external pins related to the UARTB function in the alternate-function mode. If a low level is input, it is judged that a falling edge is input after the UBRXE bit has been set to 1, and reception may be started.

**Remark** When reception is disabled, the receive shift register does not detect a start bit. No shift-in processing or transfer processing to the receive data register is performed, and the contents of the receive data register are retained.

When reception is enabled, the receive shift operation starts, in synchronization with the detection of the start bit, and when the reception of one frame is completed, the contents of the receive shift register are transferred to the receive data register.

A reception end interrupt request signal (INTUBTIR) is also generated, in synchronization with the transfer to the receive data register (in FIFO mode, transfer triggered by reaching set number of receive data).

If data is stored in receive FIFO when the next data does not come (start bit is not detected) after the next data reception wait time specified by the UBFIC1.UBTC4 to UBFIC1.UBTC0 bits has elapsed in the FIFO mode, a reception timeout interrupt request signal (INTUBTITO) is generated.

After reset: 10H R/W Address: FFFFA40H

|        |       |       |       |       |       |       |      |      |
|--------|-------|-------|-------|-------|-------|-------|------|------|
|        | <7>   | <6>   | <5>   | <4>   | 3     | 2     | 1    | 0    |
| UBCTL0 | UBPWR | UBTXE | UBRXE | UBDIR | UBPS1 | UBPS0 | UBCL | UBSL |

|   |                                  |
|---|----------------------------------|
| UBPWR   | Operation clock control to UARTB |
| 0   | Stops supply of clocks to UARTB  |
| 1   | Supplies clocks to UARTB         |
| <ul style="list-style-type: none"> <li>• When the UBPWR bit is cleared to 0, the UARTB can be asynchronously reset.</li> <li>• When the UBPWR bit = 0, UARTB is in a reset state. Therefore, to operate UARTB, the UBPWR bit must be set to 1.</li> <li>• When the UBPWR bit is changed from 1 to 0, all registers of UARTB are initialized. When the UBPWR bit is set to 1 again, the UARTB registers must be set again.</li> <li>• The TXDB pin output is high level when the UBPWR bit is cleared to 0.</li> </ul> |                                  |

|   |                          |
|---|--------------------------|
| UBTXE   | Transmission enable      |
| 0   | Transmission is disabled |
| 1   | Transmission is enabled  |
| <ul style="list-style-type: none"> <li>• On startup, set the UBPWR bit to 1 and then set the UBTXE bit to 1. To stop transmission, clear the UBTXE bit to 0 and then the UBPWR bit to 0.</li> <li>• When the transmission unit status is to be initialized, the transmission status may not be able to be initialized unless the UBTXE bit is set to 1 again after an interval of two cycles of f<sub>xx</sub> has elapsed since the UBTXE bit was cleared to 0.</li> </ul> |                          |

|  |                       |
|--|-----------------------|
| UBRXE  | Reception enable      |
| 0  | Reception is disabled |
| 1  | Reception is enabled  |
| <ul style="list-style-type: none"> <li>• On startup, set the UBPWR bit to 1 and then set the UBRXE bit to 1. To stop reception, clear the UBRXE bit to 0 and then the UBPWR bit to 0.</li> <li>• When the reception unit status is to be initialized, the reception status may not be able to be initialized unless the UBRXE bit is set to 1 again after an interval of two cycles of f<sub>xx</sub> has elapsed since the UBRXE bit was cleared to 0.</li> </ul> |                       |

| UBDIR | Specification of transfer direction mode (MSB/LSB) |
|-------|--|
| 0     | MSB transfer first                                 |
| 1     | LSB transfer first                                 |

- Clear the UBPWR bit or UBTXE and UBRXE bits to 0 before changing the setting of the UBDIR bit.

| UBPS1 | UBPS0 | Parity selection during transmission | Parity selection during reception |
|-------|-------|--------------------------------------|-----------------------------------|
| 0     | 0     | Do not output a parity bit           | Receive with no parity            |
| 0     | 1     | Output 0 parity                      | Receive as 0 parity               |
| 1     | 0     | Output odd parity                    | Judge as odd parity               |
| 1     | 1     | Output even parity                   | Judge as even parity              |

- Clear the UBTXE and UBRXE bits to 0 before overwriting the UBPS1 and UBPS0 bits.
- If “0 parity” is selected for reception, no parity judgment is made. Therefore, no error interrupt is generated because the UBSTR.UBPE bit is not set to 1.

| UBCL | Specification of data character length of 1-frame transmit/receive data |
|------|---|
| 0    | 7 bits  |
| 1    | 8 bits  |

Clear the UBTXE and UBRXE bits to 0 before overwriting the UBCL bit.

| UBSL | Specification of stop bit length of transmit data |
|------|---|
| 0    | 1 bit   |
| 1    | 2 bits  |

- Clear the UBTXE bit to 0 before overwriting the UBSL bit.
- Since reception always operates by using a single stop bit length, the UBSL bit setting does not affect receive operations.

**Remark** For details of parity, see 15.7.6 Parity types and corresponding operation.

**(2) UARTB status register (UBSTR)**

The UBSTR register indicates the transfer status and reception error contents while UARTB is transmitting data.

The status flag that indicates the transfer status during transmission indicates the data retention status of the transmit shift register and transmit data register (the UBTX register in the single mode or transmit FIFO in the FIFO mode). The status flag that indicates a reception error holds its status until it is cleared to 0.

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

Reset sets this register to 00H.

**Caution** When the UBCTL0.UBPWR bit or UBCTL0.UBRXE bit is set to 0, or when 0 is written to the UBSTR register, the UBSTR.UBOVF, UBSTR.UBPE, UBSTR.UBFE, and UBSTR.UBOVE bits are cleared to 0.

(1/2)

|                  |       |     |                      |   |       |      |      |       |
|------------------|-------|-----|----------------------|---|-------|------|------|-------|
| After reset: 00H |       | R/W | Address: FFFFFFFA44H |   |       |      |      |       |
|                  | <7>   | 6   | 5                    | 4 | 3     | <2>  | <1>  | <0>   |
| UBSTR            | UBTSF | 0   | 0                    | 0 | UBOVF | UBPE | UBFE | UBOVE |

|       |  |
|-------|--|
| UBTSF | Transfer status flag   |
| 0     | <ul style="list-style-type: none"> <li>In single mode (UBFIC0.UBMOD bit = 0)<br/>Data to be transferred to the transmit shift register and UBTX register does not exist (cleared (0) when UBCTL0.UBPWR bit = 0 or UBCTL0.UBTXE bit = 0).</li> <li>In FIFO mode (UBFIC0.UBMOD bit = 1)<br/>Data to be transferred to the transmit shift register and transmit FIFO does not exist (cleared (0) when UBCTL0.UBPWR bit = 0 or UBCTL0.UBTXE bit = 0).</li> </ul> |
| 1     | <ul style="list-style-type: none"> <li>In single mode (UBFIC0.UBMOD bit = 0)<br/>Data to be transferred to the transmit shift register or UBTX register exists (transmission in progress).</li> <li>In FIFO mode (UBFIC0.UBMOD bit = 1)<br/>Data to be transferred to the transmit shift register and transmit FIFO exists (transmission in progress).</li> </ul>  |

The value of the UBTSF bit is reflected after two periods of  $f_{xx}$  have elapsed, after the transmit data is written to the UBTX register. Therefore, exercise care when referencing the UBTSF bit after transmit data has been written to the UBTX register.

|       |                                       |
|-------|---------------------------------------|
| UBOVF | Overflow flag                         |
| 0     | Overflow did not occur.               |
| 1     | Overflow occurred (during reception). |

- The UBOVF bit is valid only in the FIFO mode (when UBFIC0.UBMOD bit = 1), and invalid in the single mode (when UBFIC0.UBMOD bit = 0).
- If an overflow occurs, the received data is not written to receive FIFO but discarded.

| UBPE | Parity error flag                         |
|------|---|
| 0    | Parity error did not occur.               |
| 1    | Parity error occurred (during reception). |

- The UBPE bit is valid only in the single mode (when UBFIC0.UBMOD bit = 0), and invalid in the FIFO mode (when UBFIC0.UBMOD bit = 1).
- The operation of the UBPE bit differs according to the settings of the UBCTL0.UBPS1 and UBCTL0.UBPS0 bits.

| UBFE | Framing error flag                         |
|------|--|
| 0    | Framing error did not occur.               |
| 1    | Framing error occurred (during reception). |

- The UBFE bit is valid only in the single mode (when UBFIC0.UBMOD bit = 0), and invalid in the FIFO mode (when UBFIC0.UBMOD bit = 1).
- Only the first bit of the stop bits of the receive data is checked, regardless of the stop bit length.

| UBOVE | Overrun error flag                         |
|-------|--|
| 0     | Overrun error did not occur.               |
| 1     | Overrun error occurred (during reception). |

- The UBOVE bit is valid only in the single mode (when UBFIC0.UBMOD bit = 0), and invalid in the FIFO mode (when UBFIC0.UBMOD bit = 1).
- When an overrun error occurs, the next receive data value is not written to the UBRX register and the data is discarded.

**(3) UARTB control register 2 (UBCTL2)**

The UBCTL2 register is used to specify the division ratio by which to control the baud rate (serial transfer speed) of UARTB.

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

Reset sets this register to FFFFH.

**Caution** When rewriting the UBBRS15 to UBBRS0 bits of this register, set the UBCTL0.UBTXE and UBCTL0.UBRXE bits to 0 or clear the UBCTL0.UBPWR bit to 0.

|                    |     |     |                   |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|--------------------|-----|-----|-------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| After reset: FFFFH |     | R/W | Address: FFFFA42H |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|                    |     |     | 15                | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| UBCTL2             | UB  | UB  | UB                | UB  | UB  | UB  | UB  | UB  | UB  | UB  | UB  | UB  | UB  | UB  | UB  | UB  | UB  | UB  |
|                    | BRS | BRS | BRS               | BRS | BRS | BRS | BRS | BRS | BRS | BRS | BRS | BRS | BRS | BRS | BRS | BRS | BRS | BRS |
|                    | 15  | 14  | 13                | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |     |     |

**Remark** For the UBBRS15 to UBBRS0 bits, see **Table 15-2 Division Value of 16-bit Counter**.

**Table 15-2. Division Value of 16-bit Counter**

| UB BRS 15 | UB BRS 14 | UB BRS 13 | UB BRS 12 | UB BRS 11 | UB BRS 10 | UB BRS 9 | UB BRS 8 | UB BRS 7 | UB BRS 6 | UB BRS 5 | UB BRS 4 | UB BRS 3 | UB BRS 2 | UB BRS 1 | UB BRS 0 | k     | Output Clock Selected |
|-----------|-----------|-----------|-----------|-----------|-----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-------|-----------------------|
| 0         | 0         | 0         | 0         | 0         | 0         | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | x        | x        | 4     | fxx/k                 |
| 0         | 0         | 0         | 0         | 0         | 0         | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 1        | 0        | 0        | 4     | fxx/k                 |
| 0         | 0         | 0         | 0         | 0         | 0         | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 1        | 0        | 1        | 5     | fxx/k                 |
| 0         | 0         | 0         | 0         | 0         | 0         | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 1        | 1        | 0        | 6     | fxx/k                 |
| •         | •         | •         | •         | •         | •         | •        | •        | •        | •        | •        | •        | •        | •        | •        | •        | •     | •                     |
| •         | •         | •         | •         | •         | •         | •        | •        | •        | •        | •        | •        | •        | •        | •        | •        | •     | •                     |
| •         | •         | •         | •         | •         | •         | •        | •        | •        | •        | •        | •        | •        | •        | •        | •        | •     | •                     |
| 1         | 1         | 1         | 1         | 1         | 1         | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 0        | 0        | 65532 | fxx/k                 |
| 1         | 1         | 1         | 1         | 1         | 1         | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 0        | 1        | 65533 | fxx/k                 |
| 1         | 1         | 1         | 1         | 1         | 1         | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 0        | 65534 | fxx/k                 |
| 1         | 1         | 1         | 1         | 1         | 1         | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 65535 | fxx/k                 |

- Remarks**
1. fxx: Peripheral clock
  2. k: Value set by the UBCTL2.UBBRS15 to UBCTL2.UBBRS0 bits (k = 4, 5, 6, ..., 65535)
  3. x: Don't care

**(4) UARTB transmit data register (UBTX)**

The UBTX register is used to set transmit data. It functions as the 8-bit × 1-stage UBTX register, in the single mode (UBFIC0.UBMOD bit = 0), and as the 8-bit × 16-stage transmit FIFO in the FIFO mode (UBFIC0.UBMOD bit = 1).

In the single mode, transmission is started by writing transmit data to the UBTX register when transmission is enabled (UBCTL0.UBTXE bit = 1). When data can be written to the UBTX register (when 1 byte of data is transferred from the UBTX register to the transmit shift register), a transmission enable interrupt request signal (INTUBTIT) is generated.

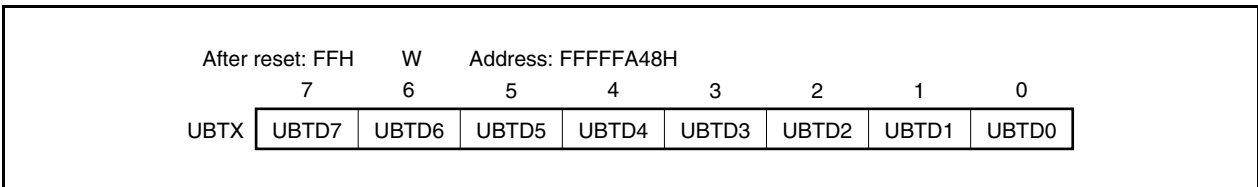
In the FIFO mode, transmission is started by enabling transmission (UBTXE bit = 1) after writing at least the number of transmit data set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits and 16 bytes or less to transmit FIFO. When the number of transmit data set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits have been transferred from transmit FIFO to the transmit shift register (transmit data of the number set as the trigger can be written to transmit FIFO), a transmission enable interrupt request signal (INTUBTIT) is generated. In the FIFO mode, a FIFO transmission enable interrupt request signal (INTUBTIF) is generated when there is no more data in transmit FIFO and the transmit shift register (when the FIFO and register become empty).

For the generation timing of the interrupt, see **15.5 Interrupt Request Signals**.

When 7-bit length data is transmitted with the LSB first, bits 6 to 0 of the transmit data register are transmitted as the transmit data from the LSB (bit 0) with the MSB (bit 7) always being 0. When data is transmitted with the MSB first, bits 7 to 1 of the transmit data register are transmitted as the transmit data from the MSB (bit 7) with the LSB (bit 0) always being 0.

This register is write-only in 8-bit units. Data is written to the transmit data register.

Reset sets this register to FFH.





**(5) UARTB receive data register AP (UBRXAP), UARTB receive data register (UBRX)**

These registers store parallel data converted by the receive shift register. They function as the 8-bit × 1-stage UBRX register, in the single mode (UBFIC0.UBMOD bit = 0), and as the 16-bit × 16-stage receive FIFO (UBRXAP register) in the FIFO mode (UBFIC0.UBMOD bit = 1).

The receive data is stored in the lower 8 bits of the receive FIFO (UBRXAP register) and the error information of the received data is stored in the higher 8 bits (bit 8 and bit 9). If a reception error (such as a parity error or a framing error) occurs in the FIFO mode, the UBRXAP register is read in 16-bit (halfword) units. In this way, the flag of the data stored in receive FIFO can be checked (error information is appended as UBPEF bit = 1 or UBFEF bit = 1), so that the error data can be recognized (when the lower 8 bits of the UBRXAP register are read in 8-bit (byte) units, the higher 8 bits are discarded. Therefore, if no error has occurred, the receive data of the UBRXAP register can be read successively by being read in 8-bit (byte) units in the same way as the UBRX register).

If reception is enabled (UBCTL0.UBRXE bit = 1), the receive data is transferred from the receive shift register to the receive data register, in synchronization with the completion of the shift-in processing of one frame.

By transferring the receive data to the UBRX register in the single mode or by transferring the number of receive data set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits to the receive FIFO in the FIFO mode, a reception end interrupt request signal (INTUBTIR) is generated. If data is stored in receive FIFO when the next data does not come (start bit is not detected) even after the next data reception wait time specified by the UBFIC1.UBTC4 to UBFIC1.UBTC0 bits has elapsed in the FIFO mode, a reception timeout interrupt request signal (INTUBTITO) is generated.

For information about the timing for generating these interrupt requests, see **15.5 Interrupt Request Signals**.

If data is received with the LSB first when the data length is specified as 7 bits, the received data is transferred to bits 6 to 0 of the receive data register from the LSB (bit 0), with the MSB (bit 7) always being 0. If data is received with the MSB first, it is transferred to bits 7 to 1 of the receive data register from the MSB (bit 7) with the LSB (bit 0) always being 0. However, if an overrun error occurs, the receive data at that time is not transferred to the receive data register.

The UBRXAP register is read-only in 16-bit units. However, the lower 8 bits of the UBRXAP register are read-only in 8-bit units.

The UBRX register is read-only in 8-bit units.

In addition to reset input, the value of these registers can be set to FFH in the single mode or to 00FFH in the FIFO mode, by clearing the UBCTL0.UBPWR bit to 0.

- Cautions**
1. **The UBPEF and UBFEF bits cannot be read because these registers serve as 8-bit registers in the single mode.**
  2. **When no reception error has occurred in the FIFO mode, the receive data of the UBRXAP register can be read successively by reading the lower 8 bits of the UBRXAP register in 8-bit (byte) units. An 8-bit access to the higher 8 bits is prohibited. If they are accessed, the operation is not guaranteed.**

**Cautions 3. Do not perform the following operations when debugging a system that uses the single mode.**

- **Setting a break for an instruction immediately after the UBRX register is read**
- **Setting a break before DMA transfer with the UBRX register specified as the transfer source is ended**
- **Setting a break before end of reception of the next data after reception of data and reading the UBRX register, and checking the UBRX register in the I/O register window of the debugger**

**If any of these operations is performed, an overrun error may occur during the subsequent reception.**

After reset: 00FFH R Address: FFFFA46H

|        |    |    |    |    |    |    |     |     |     |     |     |     |     |     |     |     |
|--------|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|        | 15 | 14 | 13 | 12 | 11 | 10 | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| UBRXAP | 0  | 0  | 0  | 0  | 0  | 0  | UB  | UB  | UB  | UB  | UB  | UB  | UB  | UB  | UB  | UB  |
|        |    |    |    |    |    |    | PEF | FEF | RD7 | RD6 | RD5 | RD4 | RD3 | RD2 | RD1 | RD0 |

After reset: FFH R Address: FFFFA46H

|      |       |       |       |       |       |       |       |       |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
|      | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| UBRX | UBRD7 | UBRD6 | UBRD5 | UBRD4 | UBRD3 | UBRD2 | UBRD1 | UBRD0 |

|   |   |
|---|---|
| UBPEF   | Parity error flag                       |
| 0   | No parity error                         |
| 1   | Parity error occurs (during reception). |
| <ul style="list-style-type: none"> <li>• The UBPEF bit is valid only in the FIFO mode (UBFIC0.UBMOD bit = 1), and is invalid in the single mode (UBFIC0.UBMOD bit = 0).</li> <li>• The operation of the UBPEF bit differs depending on the set values of the UBCTL0.UBPS1 and UBCTL0.UBPS0 bits.</li> </ul> |   |

|  |  |
|--|--|
| UBFEF  | Framing error flag                       |
| 0  | No framing error                         |
| 1  | Framing error occurs (during reception). |
| <ul style="list-style-type: none"> <li>• The UBFEF bit is valid only in the FIFO mode (UBFIC0.UBMOD bit = 1), and is invalid in the single mode (UBFIC0.UBMOD bit = 0).</li> <li>• Only the first bit of the stop bits of the receive data is checked, regardless of the stop bit length.</li> </ul> |  |

|                |                      |
|----------------|----------------------|
| UBRD7 to UBRD0 | Stores receive data. |
|----------------|----------------------|

**(6) UARTB FIFO control register 0 (UBFIC0)**

The UBFIC0 register is used to select the operation mode of UARTB and the functions that become valid in the FIFO mode (UBMOD bit = 1). In the FIFO mode, it clears transmit FIFO/receive FIFO and specifies the timing mode in which the transmission enable interrupt request signal (INTUBTIT)/reception end interrupt request signal (INTUBTIR) is generated.

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: FFFFFFFA4AH |   |   |       |       |       |       |
| UBFIC0           | 7     | 6                    | 5 | 4 | 3     | 2     | 1     | 0     |
|                  | UBMOD | 0                    | 0 | 0 | UBTFC | UBRFC | UBITM | UBIRM |

|       |                                       |
|-------|---------------------------------------|
| UBMOD | Specification of UARTB operation mode |
| 0     | Single mode                           |
| 1     | FIFO mode                             |

|       |   |
|-------|---|
| UBTFC | Transmit FIFO clear trigger bit   |
| 0     | Normal status   |
| 1     | Clear (This bit automatically returns to 0 after transmit FIFO is cleared.) |

- The UBTFC bit is valid only in the FIFO mode (UBMOD bit = 1), and is invalid in the single mode (UBMOD bit = 0).
- When 1 is written to the UBTFC bit, the pointer to transmit FIFO is cleared to 0. In the pending mode (UBITM bit = 0), the interrupt request signal (INTUBTIT) held pending is cleared<sup>Note</sup>. However, bit 7 (UTIF) of the interrupt control register (UTIC) is not cleared to 0. Clear this bit to 0 as necessary. When 0 is written to the UBTFC bit, the status is retained. No operation, such as clearing or setting, is executed.
- When writing 1 to the UBTFC bit, be sure to clear the UBCTL0.UBTXE bit to 0 (disabling transmission). If 1 is written to the UBTFC bit when the UBTXE bit is 1 (transmission enabled), the operation is not guaranteed.

**Note** After transmit FIFO is cleared (UBTFC bit = 1), accessing the registers related to UARTB is prohibited for the duration of four cycles of f<sub>xx</sub> or until clearing the UBTFC bit (automatic recovery) is confirmed by reading the UBFIC0 register. If these registers are accessed, the operation is not guaranteed.

**Remark** f<sub>xx</sub>: Peripheral clock

| UBRFC | Receive FIFO (UBRXAP) clear trigger bit                                    |
|-------|--|
| 0     | Normal status  |
| 1     | Clear (This bit automatically returns to 0 after receive FIFO is cleared.) |

- The UBRFC bit is valid only in the FIFO mode (UBMOD bit = 1), and is invalid in the single mode (UBMOD bit = 0).
- When 1 is written to the UBRFC bit, the pointer to receive FIFO is cleared to 0. In the pending mode (UBIRM bit = 0), the interrupt request signal (INTUBTIR) held pending is cleared<sup>Note</sup>. However, bit 7 (URIF) of the interrupt control register (URIC) is not cleared to 0. Clear this bit to 0 as necessary. When 0 is written to the UBRFC bit, the status is retained. No operation, such as clearing or setting, is executed.
- When writing 1 to the UBRFC bit, be sure to clear the UBCTL0.UBRXE bit to 0 (disabling reception). If 1 is written to the UBRFC bit when the UBRXE bit is 1 (reception enabled), the operation is not guaranteed.

| UBITM | Specification of INTUBTIT interrupt generation timing in FIFO mode |
|-------|--|
| 0     | Pending mode   |
| 1     | Pointer mode   |

In the FIFO mode, the INTUBTIT signal is generated as soon as transmit data of the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits have been transferred from transmit FIFO to the transmit shift register. After the INTUBTIT signal request has been generated, specify the timing of actually generating the INTUBTIT signal as the pending mode or pointer mode. For details, see **15.6 (2) Pending mode/pointer mode**.

| UBIRM | Specification of INTUBTIR interrupt generation timing in FIFO mode |
|-------|--|
| 0     | Pending mode   |
| 1     | Pointer mode   |

In the FIFO mode, the INTUBTIR signal is generated as soon as receive data of the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits have been transferred from the receive shift register to receive FIFO. After the INTUBTIR signal request has been generated, specify the timing of actually generating the INTUBTIR signal as the pending mode or pointer mode. For details, see **15.6 (2) Pending mode/pointer mode**.

**Note** After receive FIFO (UBRXAP) is cleared (UBRFC bit = 1), accessing the registers related to UARTB is prohibited for the duration of four cycles of f<sub>xx</sub> or until clearing the UBRFC bit (automatic recovery) is confirmed by reading the UBFIC0 register. If these registers are accessed, the operation is not guaranteed.

**Remark** f<sub>xx</sub>: Peripheral clock

**(7) UARTB FIFO control register 1 (UBFIC1)**

The UBFIC1 register is valid in the FIFO mode (UBFIC0.UBMOD bit = 1). It generates a reception timeout interrupt request signal (INTUBTITO) if data is stored in receive FIFO when the next data does not come (start bit is not detected) after the lapse of the time set by the UBTC4 to UBTC0 bits (next data reception wait time), after the stop bit has been received.

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: FFFFFFFA4BH |   |   |       |       |       |       |       |
|                  |     | 7                    | 6 | 5 | 4     | 3     | 2     | 1     | 0     |
| UBFIC1           |     | UBTCE                | 0 | 0 | UBTC4 | UBTC3 | UBTC2 | UBTC1 | UBTC0 |

|       |  |
|-------|--|
| UBTCE | Specification of timeout counter function disable/enable |
| 0     | Disable use of timeout counter function.                 |
| 1     | Enable use of timeout counter function.                  |

| UBTC4 | UBTC3 | UBTC2 | UBTC1 | UBTC0 | Next data reception wait time |
|-------|-------|-------|-------|-------|-------------------------------|
| 0     | 0     | 0     | 0     | 0     | 32 bytes (32 × 8/ baud rate)  |
| 0     | 0     | 0     | 0     | 1     | 31 bytes (31 × 8/ baud rate)  |
| 0     | 0     | 0     | 1     | 0     | 30 bytes (30 × 8/ baud rate)  |
| 0     | 0     | 0     | 1     | 1     | 29 bytes (29 × 8/ baud rate)  |
| •     | •     | •     | •     | •     | •                             |
| •     | •     | •     | •     | •     | •                             |
| •     | •     | •     | •     | •     | •                             |
| 1     | 1     | 1     | 0     | 0     | 4 bytes (4 × 8/ baud rate)    |
| 1     | 1     | 1     | 0     | 1     | 3 bytes (3 × 8/ baud rate)    |
| 1     | 1     | 1     | 1     | 0     | 2 bytes (2 × 8/ baud rate)    |
| 1     | 1     | 1     | 1     | 1     | 1 byte (1 × 8/ baud rate)     |

When counting up of the reception wait time, set by the UBTC4 to UBTC0 bits, is complete, the count value of the timeout counter is cleared to 0, regardless of the status of the data stored in receive FIFO. When the next start bit is later detected, counting is started again from the stop bit of that data.

**(8) UARTB FIFO control register 2 (UBFIC2)**

The UBFIC2 register is valid in the FIFO mode (UBFIC0.UBMOD bit = 1). It sets the timing of generating an interrupt, using the number of transmit/receive data as a trigger. When data is transmitted, the number of data transferred from transmit FIFO is specified as the condition of generating the interrupt. When data is received, the number of data stored in receive FIFO is specified as the interrupt generation condition.

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

When the higher 8 bits of the UBFIC2 register can be used as the UBFIC2H register and the lower 8 bits, as the UBFIC2L register, these registers can be read or written in 8-bit units.

Reset sets the UBFIC2 register to 0000H and the UBFIC2H and UBFIC2L registers to 00H.

**Caution** Be sure to set the UBCTL0.UBTXE bit (to disable transmission) and UBCTL0.UBRXE bit (to disable reception) to 0 before writing data to the UBFIC2 register. If data is written to the UBFIC2 register with the UBTXE or UBRXE bit set to 1, the operation is not guaranteed.

(1/2)

After reset: 0000H    R/W    Address: FFFFFA4CH

|        |    |    |    |    |     |     |     |     |   |   |   |   |     |     |     |     |
|--------|----|----|----|----|-----|-----|-----|-----|---|---|---|---|-----|-----|-----|-----|
|        | 15 | 14 | 13 | 12 | 11  | 10  | 9   | 8   | 7 | 6 | 5 | 4 | 3   | 2   | 1   | 0   |
| UBFIC2 | 0  | 0  | 0  | 0  | UB  | UB  | UB  | UB  | 0 | 0 | 0 | 0 | UB  | UB  | UB  | UB  |
|        |    |    |    |    | TT3 | TT2 | TT1 | TT0 |   |   |   |   | RT3 | RT2 | RT1 | RT0 |

| UBTT3 | UBTT2 | UBTT1 | UBTT0 | Number of data of transmit FIFO set as trigger | Pointer mode       | Pending mode |
|-------|-------|-------|-------|--|--------------------|--------------|
| 0     | 0     | 0     | 0     | 1 byte   | Settable           | Settable     |
| 0     | 0     | 0     | 1     | 2 bytes  | Setting prohibited |              |
| 0     | 0     | 1     | 0     | 3 bytes  |                    |              |
| 0     | 0     | 1     | 1     | 4 bytes  |                    |              |
| 0     | 1     | 0     | 0     | 5 bytes  |                    |              |
| 0     | 1     | 0     | 1     | 6 bytes  |                    |              |
| 0     | 1     | 1     | 0     | 7 bytes  |                    |              |
| 0     | 1     | 1     | 1     | 8 bytes  |                    |              |
| 1     | 0     | 0     | 0     | 9 bytes  |                    |              |
| 1     | 0     | 0     | 1     | 10 bytes                                       |                    |              |
| 1     | 0     | 1     | 0     | 11 bytes                                       |                    |              |
| 1     | 0     | 1     | 1     | 12 bytes                                       |                    |              |
| 1     | 1     | 0     | 0     | 13 bytes                                       |                    |              |
| 1     | 1     | 0     | 1     | 14 bytes                                       |                    |              |
| 1     | 1     | 1     | 0     | 15 bytes                                       |                    |              |
| 1     | 1     | 1     | 1     | 16 bytes                                       |                    |              |

- Set the number of transmit FIFO transmit data to be the trigger.
- Each time data of the specified number has shifted out from transmit FIFO to the transmit shift register, the INTUBTIT signal is generated.  
In the pending mode (UBFIC0.UBITM bit = 0), the INTUBTIT signal is generated under the conditions of the pending mode.
- In the pointer mode (UBFIC0.UBITM bit = 1), the number of transmit data set as the trigger can be only 1 byte (UBTT3 to UBTT0 bits = 0000), and other settings are prohibited. If a setting of other than 1 byte is made, the operation is not guaranteed.

| UBRT3 | UBRT2 | UBRT1 | UBRT0 | Number of data of transmit FIFO set as trigger | Pointer mode       | Pending mode |
|-------|-------|-------|-------|--|--------------------|--------------|
| 0     | 0     | 0     | 0     | 1 byte   | Settable           | Settable     |
| 0     | 0     | 0     | 1     | 2 bytes  | Setting prohibited |              |
| 0     | 0     | 1     | 0     | 3 bytes  |                    |              |
| 0     | 0     | 1     | 1     | 4 bytes  |                    |              |
| 0     | 1     | 0     | 0     | 5 bytes  |                    |              |
| 0     | 1     | 0     | 1     | 6 bytes  |                    |              |
| 0     | 1     | 1     | 0     | 7 bytes  |                    |              |
| 0     | 1     | 1     | 1     | 8 bytes  |                    |              |
| 1     | 0     | 0     | 0     | 9 bytes  |                    |              |
| 1     | 0     | 0     | 1     | 10 bytes                                       |                    |              |
| 1     | 0     | 1     | 0     | 11 bytes                                       |                    |              |
| 1     | 0     | 1     | 1     | 12 bytes                                       |                    |              |
| 1     | 1     | 0     | 0     | 13 bytes                                       |                    |              |
| 1     | 1     | 0     | 1     | 14 bytes                                       |                    |              |
| 1     | 1     | 1     | 0     | 15 bytes                                       |                    |              |
| 1     | 1     | 1     | 1     | 16 bytes                                       |                    |              |

- Set the number of receive FIFO receive data to be the trigger.
- Each time data of the specified number has been stored from the receive shift register to receive FIFO, the INTUBTIR interrupt is generated.  
In the pending mode (UBFIC0.UBIRM bit = 0), the INTUBTIR signal is generated under the conditions of the pending mode.
- In the pointer mode (UBFIC0.UBIRM bit = 1), the number of receive data set as the trigger can be only 1 byte (UBRT3 to UBRT0 bits = 0000), and other settings are prohibited. If a setting of other than 1 byte is made, the operation is not guaranteed.

**(9) UARTB FIFO status register 0 (UBFIS0)**

The UBFIS0 register is valid in the FIFO mode (UBFIC0.UBMOD bit = 1). It is used to read the number of bytes of the data stored in receive FIFO.

This register is read-only in 8-bit units.

Reset sets this register to 00H.

After reset: 00H    R    Address: FFFFFFFA4EH

|        |   |   |   |       |       |       |       |       |
|--------|---|---|---|-------|-------|-------|-------|-------|
|        | 7 | 6 | 5 | 4     | 3     | 2     | 1     | 0     |
| UBFIS0 | 0 | 0 | 0 | UBRB4 | UBRB3 | UBRB2 | UBRB1 | UBRB0 |

| UBRB4  | UBRB3 | UBRB2 | UBRB1 | UBRB0 | Receive FIFO pointer |
|--|-------|-------|-------|-------|----------------------|
| 0  | 0     | 0     | 0     | 0     | 0 bytes              |
| 0  | 0     | 0     | 0     | 1     | 1 byte               |
| 0  | 0     | 0     | 1     | 0     | 2 bytes              |
| 0  | 0     | 0     | 1     | 1     | 3 bytes              |
| 0  | 0     | 1     | 0     | 0     | 4 bytes              |
| 0  | 0     | 1     | 0     | 1     | 5 bytes              |
| 0  | 0     | 1     | 1     | 0     | 6 bytes              |
| 0  | 0     | 1     | 1     | 1     | 7 bytes              |
| 0  | 1     | 0     | 0     | 0     | 8 bytes              |
| 0  | 1     | 0     | 0     | 1     | 9 bytes              |
| 0  | 1     | 0     | 1     | 0     | 10 bytes             |
| 0  | 1     | 0     | 1     | 1     | 11 bytes             |
| 0  | 1     | 1     | 0     | 0     | 12 bytes             |
| 0  | 1     | 1     | 0     | 1     | 13 bytes             |
| 0  | 1     | 1     | 1     | 0     | 14 bytes             |
| 0  | 1     | 1     | 1     | 1     | 15 bytes             |
| 1  | 0     | 0     | 0     | 0     | 16 bytes             |
| Other than above   |       |       |       |       | Invalid              |
| Indicates the number of bytes (readable bytes) of the data stored in receive FIFO as a receive FIFO pointer. |       |       |       |       |                      |



**(10) UARTB FIFO status register 1 (UBFIS1)**

The UBFIS1 register is valid in the FIFO mode (UBFIC0.UBMOD bit = 1). This register can be used to read the number of empty bytes of transmit FIFO.

This register is read-only in 8-bit units.

Reset sets this register to 10H.

**Caution** The values of the UBTB4 to UBTB0 bits are reflected after transmit data has been written to the UBTX register and then time of two cycles of f<sub>xx</sub> has passed. Therefore, care must be exercised when referencing the UBFIS1 register after transmit data has been written to the UBTX register.

|                  |   |   |                   |       |       |       |       |       |
|------------------|---|---|-------------------|-------|-------|-------|-------|-------|
| After reset: 10H |   | R | Address: FFFFA4FH |       |       |       |       |       |
|                  | 7 | 6 | 5                 | 4     | 3     | 2     | 1     | 0     |
| UBFIS1           | 0 | 0 | 0                 | UBTB4 | UBTB3 | UBTB2 | UBTB1 | UBTB0 |

| UBTB4  | UBTB3 | UBTB2 | UBTB1 | UBTB0 | Transmit FIFO pointer |
|--|-------|-------|-------|-------|-----------------------|
| 0  | 0     | 0     | 0     | 0     | 0 bytes               |
| 0  | 0     | 0     | 0     | 1     | 1 byte                |
| 0  | 0     | 0     | 1     | 0     | 2 bytes               |
| 0  | 0     | 0     | 1     | 1     | 3 bytes               |
| 0  | 0     | 1     | 0     | 0     | 4 bytes               |
| 0  | 0     | 1     | 0     | 1     | 5 bytes               |
| 0  | 0     | 1     | 1     | 0     | 6 bytes               |
| 0  | 0     | 1     | 1     | 1     | 7 bytes               |
| 0  | 1     | 0     | 0     | 0     | 8 bytes               |
| 0  | 1     | 0     | 0     | 1     | 9 bytes               |
| 0  | 1     | 0     | 1     | 0     | 10 bytes              |
| 0  | 1     | 0     | 1     | 1     | 11 bytes              |
| 0  | 1     | 1     | 0     | 0     | 12 bytes              |
| 0  | 1     | 1     | 0     | 1     | 13 bytes              |
| 0  | 1     | 1     | 1     | 0     | 14 bytes              |
| 0  | 1     | 1     | 1     | 1     | 15 bytes              |
| 1  | 0     | 0     | 0     | 0     | 16 bytes              |
| Setting prohibited   |       |       |       |       | Invalid               |
| Indicates the number of empty bytes of transmit FIFO (bytes that can be written) as a transmit FIFO pointer. |       |       |       |       |                       |

## 15.5 Interrupt Request Signals

The following five types of interrupt requests are generated from UARTB.

- Reception error interrupt request signal (INTUBTIRE)
- Reception end interrupt request signal (INTUBTIR)
- Transmission enable interrupt request signal (INTUBTIT)
- FIFO transmission end interrupt request signal (INTUBTIF)
- Reception timeout interrupt request signal (INTUBTITO)

The default priorities among these five types of interrupt requests is, from high to low, reception error interrupt request signal, reception end interrupt request signal, transmission enable interrupt request signal, FIFO transmission end interrupt request signal, and reception timeout interrupt request signal.

**Table 15-3. Generated Interrupts and Default Priorities**

| Interrupt             | Priority |
|-----------------------|----------|
| Reception error       | 1        |
| Reception end         | 2        |
| Transmission enable   | 3        |
| FIFO transmission end | 4        |
| Reception timeout     | 5        |

### (1) Reception error interrupt request signal (INTUBTIRE)

#### (a) Single mode

When reception is enabled, a reception error interrupt request signal is generated according to the logical OR of the three types of reception errors (parity error, framing error, overrun error) explained for the UBSTR register.

When reception is disabled, no reception error interrupt request signal is generated.

#### (b) FIFO mode

When reception is enabled, a reception error interrupt request signal is generated according to the logical OR of the three types of reception errors (parity error, framing error, overflow error) explained for the UBSTR register.

When reception is disabled, no reception error interrupt request signal is generated.

**(2) Reception end interrupt request signal (INTUBTIR)****(a) Single mode**

When reception is enabled, a reception end interrupt request signal is generated if data is shifted into the receive shift register and stored in the UBRX register (if the receive data can be read).

When reception is disabled, no reception end interrupt request signal is generated.

**(b) FIFO mode**

When reception is enabled, a reception end interrupt request signal is generated if data is shifted into the receive shift register and receive data of the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits is transferred to receive FIFO (if receive data of the specified number can be read).

When reception is disabled, no reception end interrupt request signal is generated.

**(3) Transmission enable interrupt request signal (INTUBTIT)****(a) Single mode**

The transmission enable interrupt request signal is generated if transmit data of one frame, including 7 or 8 bits of characters, is shifted out from the transmit shift register and the UBTX register becomes empty (if transmit data can be written).

**(b) FIFO mode**

The transmission enable interrupt request signal is generated if transmit data of the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits is transferred to the transmit shift register from transmit FIFO (if transmit data of the specified number can be written).

**(4) FIFO transmission end interrupt request signal (INTUBTIF)****(a) Single mode**

Cannot be used.

**(b) FIFO mode**

The FIFO transmission end interrupt request signal is generated when no more data is in transmit FIFO and the transmit shift register (when the FIFO and register become empty). After the FIFO transmission end interrupt request signal has occurred, clear the interrupt request signal (INTUBTIF) held pending in the pending mode (UBFIC0.UBITM bit = 0) by clearing the FIFO (UBFIC0.UBTFC bit = 1).

**Caution** If the FIFO transmission end interrupt request signal is generated (all transmit data are not transmitted) because writing the next transmit data to transmit FIFO is delayed, do not clear the FIFO.

**(5) Reception timeout interrupt request signal (INTUBTITO)**

**(a) Single mode**

Cannot be used.

**(b) FIFO mode**

The reception timeout interrupt request signal is generated if data is stored in receive FIFO when the next data does not come (start bit is not detected) even after the next data reception wait time specified by the UBFIC1.UBTC4 to UBFIC1.UBTC0 bits has elapsed, when the timeout counter function is used (UBFIC1.UBTCE bit = 1).

The reception timeout interrupt request signal is not generated while reception is disabled.

If receive data of the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits is not received, the timing of reading the number of receive data less than the specified number can be set by the reception timeout interrupt request signal.

Since the timeout counter starts counting at start bit detection, a receive timeout interrupt request signal does not occur if data of 1 character has not been received.

## 15.6 Control Modes

### (1) Single mode/FIFO mode

The single mode or FIFO mode can be selected by using the UBFIC0.UBMOD bit.

#### (a) Single mode

- Each of the UBRX and UBTX registers consists of 8 bits × 1 stage.
- When 1 byte of data is received, the INTUBTIR signal is generated.
- If the next reception operation of UARTB is ended before the receive data of the UBRX register is read after the INTUBTIR signal has been generated, the INTUBTIRE signal is generated and an overrun error occurs.

#### (b) FIFO mode

- Receive FIFO (UBRXAP register) consists of 16 bits × 16 stages and transmit FIFO consists of 8 bits × 16 stages.
- Receive FIFO can recognize error data by reading the 16-bit UBRXAP register only when a reception error (parity error or framing error) occurs.
- Transmission is started when transmission is enabled (UBCTL0.UBTXE bit = 1) after transmit data of at least the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits and 16 bytes or less are written to transmit FIFO.
- The pending mode or pointer mode can be selected for the generation timing of the INTUBTIT and INTUBTIR signals.

**(2) Pending mode/pointer mode**

The pending mode or pointer mode can be selected by using the UBFIC0.UBITM and UBFIC0.UBIRM bits in the FIFO mode (UBFIC0.UBMOD bit = 1).

If transmission is started by writing data of more than double the amount set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits to transmit FIFO, the transmission enable interrupt request signal (INTUBTIT) may occur more than once. The reception end interrupt request signal (INTUBTIR) may also occur more than once if the number of receive data set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits is 8 bytes or less in receive FIFO. In the pending or pointer mode, it can be specified how an interrupt is handled after it has been held pending.

**(a) Pending mode****(i) During transmission (writing to transmit FIFO)**

- If the data of the first transmission enable interrupt request signal (INTUBTIT) is not written to transmit FIFO after the interrupt has occurred, the second INTUBTIT signal does not occur (is held pending) even if the generation condition of the second INTUBTIT signal is satisfied (when transmit data of the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits is transferred from transmit FIFO to the transmit shift register).

When data for the first INTUBTIT signal is later written to transmit FIFO, the pending INTUBTIT signal is generated<sup>Note</sup>.

**Note** The number of pending interrupts is as follows.

When trigger is set to 1 byte (UBFIC2.UBTT3 to UBFIC2.UBTT0 bits = 0000): 15 times max.

When trigger is set to 2 bytes (UBFIC2.UBTT3 to UBFIC2.UBTT0 bits = 0001): 7 times max.

:

When trigger is set to 6 bytes (UBFIC2.UBTT3 to UBFIC2.UBTT0 bits = 0101): 1 time max.

When trigger is set to 7 bytes (UBFIC2.UBTT3 to UBFIC2.UBTT0 bits = 0110): 1 time max.

When trigger is set to 8 bytes (UBFIC2.UBTT3 to UBFIC2.UBTT0 bits = 0111): 1 time max.

- In the pending mode, transmit data of the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits is always written to transmit FIFO when the transmission enable interrupt request signal (INTUBTIT) occurs. Writing data to transmit FIFO is prohibited if the data is more or less than the specified number. If data more or less than the specified number is written, the operation is not guaranteed.
- Fix the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits to 0000 (set number of transmit data: 1 byte) to write transmit data to transmit FIFO by DMA. If any other setting is made, the operation is not guaranteed.

**(ii) During reception (reading from receive FIFO)**

- If data for the first reception end interrupt request signal (INTUBTIR) is not read from receive FIFO, the second INTUBTIR signal does not occur (is held pending) even if the generation condition of the second INTUBTIR is satisfied (if receive data of the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits can be read from receive FIFO). When data for the first INTUBTIR signal is later read from the receive FIFO, the pending INTUBTIR signal is generated<sup>Note</sup>.

**Note** The number of pending interrupts is as follows.

When trigger is set to 1 byte (UBFIC2.UBRT3 to UBFIC2.UBRT0 bits = 0000): 15 times max.  
 When trigger is set to 2 bytes (UBFIC2.UBRT3 to UBFIC2.UBRT0 bits = 0001): 7 times max.  
 :  
 When trigger is set to 6 bytes (UBFIC2.UBRT3 to UBFIC2.UBRT0 bits = 0101): 1 time max.  
 When trigger is set to 7 bytes (UBFIC2.UBRT3 to UBFIC2.UBRT0 bits = 0110): 1 time max.  
 When trigger is set to 8 bytes (UBFIC2.UBRT3 to UBFIC2.UBRT0 bits = 0111): 1 time max.

- In the pending mode, receive data of the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits is always read from receive FIFO when the reception end interrupt request signal (INTUBTIR) occurs. Reading data from receive FIFO is prohibited if the data is more or less than the specified number. If data more or less than the specified number is read, the operation is not guaranteed.
- Fix the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits to 0000 (set number of receive data: 1 byte) to read receive data from receive FIFO by DMA. If any other setting is made, the operation is not guaranteed.

**(b) Pointer mode**

**(i) During transmission (writing to transmit FIFO)**

- Each time the data of 1 byte is transferred to the transmit shift register from transmit FIFO, a transmission enable interrupt request signal (INTUBTIT) occurs.
- In the pointer mode, be sure to fix the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits to 0000 (set number of transmit data: 1 byte) as the number of transmit data set as the trigger for transmit FIFO when the transmission enable interrupt request signal (INTUBTIT) occurs. If any other setting is made, the operation is not guaranteed.
- Writing transmit data to transmit FIFO by DMA is prohibited. The operation is not guaranteed if DMA control is used.
- After the transmission enable interrupt request signal (INTUBTIT) has been acknowledged, data of the number of empty bytes of transmit FIFO can be written to transmit FIFO by referencing the UBFIS1 register.

**(ii) During reception (reading from receive FIFO)**

- Each time the data of 1 byte is transferred to receive FIFO from the receive shift register, a reception end interrupt request signal (INTUBTIR) occurs.
- In the pointer mode, be sure to fix the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits to 0000 (set number of receive data: 1 byte) as the number of receive data set as the trigger for receive FIFO when the reception end interrupt request signal (INTUBTIR) occurs. If any other setting is made, the operation is not guaranteed.
- Reading receive data from receive FIFO by DMA is prohibited. The operation is not guaranteed if DMA control is used.
- After the reception end interrupt request signal (INTUBTIR) has been acknowledged, data of the number of bytes stored in receive FIFO can be read from receive FIFO by referencing the UBFIS0 register. In some cases, however, data is not stored in receive FIFO even though the INTUBTIR signal is generated (UBFIS0.UBRB4 to UBFIS0.UBRB0 bits = 00000). In these cases, do not read data from receive FIFO. Always read data from receive FIFO when the number of bytes stored in receive FIFO is 1 byte or more (UBRB4 to UBRB0 bits = other than 00000).



## 15.7 Operation

### 15.7.1 Data format

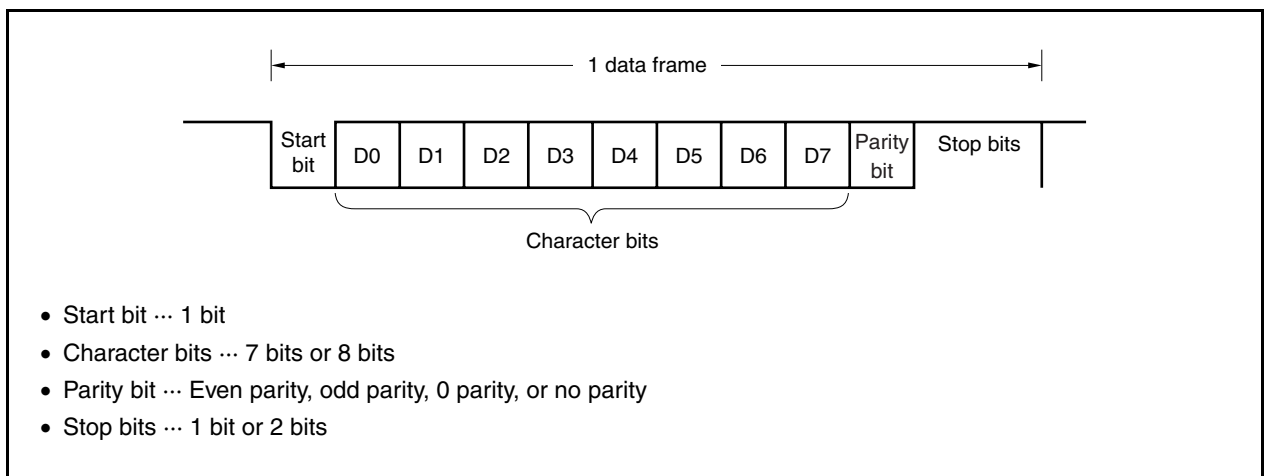
Full-duplex serial data transmission and reception can be performed.

The transmit/receive data format consists of one data frame containing a start bit, character bits, a parity bit, and stop bits as shown in Figure 15-3.

The character bit length within one data frame, the type of parity, and the stop bit length are specified by UARTB control register 0 (UBCTL0).

Also, data is transferred with LSB first/MSB first.

**Figure 15-3. Asynchronous Serial Interface Transmit/Receive Data Format (LSB-First Transfer)**



### 15.7.2 Transmit operation

In the single mode (UBFIC0.UBMOD bit = 0), transmission is enabled when the UBCTL0.UBTXE bit is set to 1, and transmission is started when transmit data is written to the UBTX register.

In the FIFO mode (UBFIC0.UBMOD bit = 1), transmission is started when transmit data of at least the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits and 16 bytes or less is written to transmit FIFO and then the UBTXE bit is set to 1.

**Caution** Setting the UBCTL0.UBTXE bit to 1 before writing transmit data to transmit FIFO in the FIFO mode is prohibited. The operation is not guaranteed if this setting is made.

#### (1) Transmission enabled state

This state is set by the UBCTL0.UBTXE bit.

- UBTXE = 1: Transmission enabled state
- UBTXE = 0: Transmission disabled state

However, because this bit is also used by CSIB2, enable transmission after setting the CB2CTL0.CB2PWR bit to 0.

Since UARTB does not have a CTS (transmission enabled signal) input pin, a port should be used to confirm whether the destination is in the reception enabled state.

#### (2) Starting a transmit operation

- **In single mode (UBFIC0.UBMOD bit = 0)**

In the single mode, transmission is started when transmit data is written to the UBTX register while transmission is enabled.

- **In FIFO mode (UBFIC0.UBMOD bit = 1)**

In the FIFO mode, transmission is started when transmit data of at least the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits and 16 bytes or less is written to transmit FIFO and then transmission is enabled (UBTXE bit = 1).

Data in the transmit data register (UBTX register in single mode or transmit FIFO in the FIFO mode) is transferred to the transmit shift register when transmission is started. Then, the transmit shift register outputs data to the TXDB pin sequentially beginning with the LSB (the transmit data is transferred sequentially starting with the start bit). The start bit, parity bit, and stop bits are added automatically.

**(3) Transmission interrupt request signal****(a) Transmission enable interrupt request signal (INTUBTIT)**

- **In single mode (UBFIC0.UBMOD bit = 0)**

In the single mode, the transmission enable interrupt request signal (INTUBTIT) occurs when transmit data can be written to the UBTX register (when 1 byte of data is transferred from the UBTX register to the transmit shift register).

- **In FIFO mode (UBFIC0.UBMOD bit = 1)**

In the FIFO mode, the INTUBTIT signal occurs when transmit data of the number set as the trigger specified by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits is transferred from transmit FIFO to the transmit shift register (if transmit data of the number set as the trigger can be written).

- **If pending mode is specified (UBFIC0.UBITM bit = 0) in FIFO mode**

If the pending mode is specified in the FIFO mode, the second INTUBTIT signal is held pending after the first INTUBTIT signal has occurred, until as many transmit data as the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits are written to transmit FIFO, even if the generation condition of the second INTUBTIT signal is satisfied. When as many transmit data as the number set as the trigger are written to transmit FIFO in response to the first INTUBTIT signal, the second pending INTUBTIT signal is generated.

- **If pointer mode is specified (UBFIC0.UBITM bit = 1) in FIFO mode**

If the pointer mode is specified in the FIFO mode, the second INTUBTIT signal occurs when the generation condition of the second INTUBTIT signal is satisfied even if as many transmit data as the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits are not written to transmit FIFO when the first INTUBTIT signal occurs.

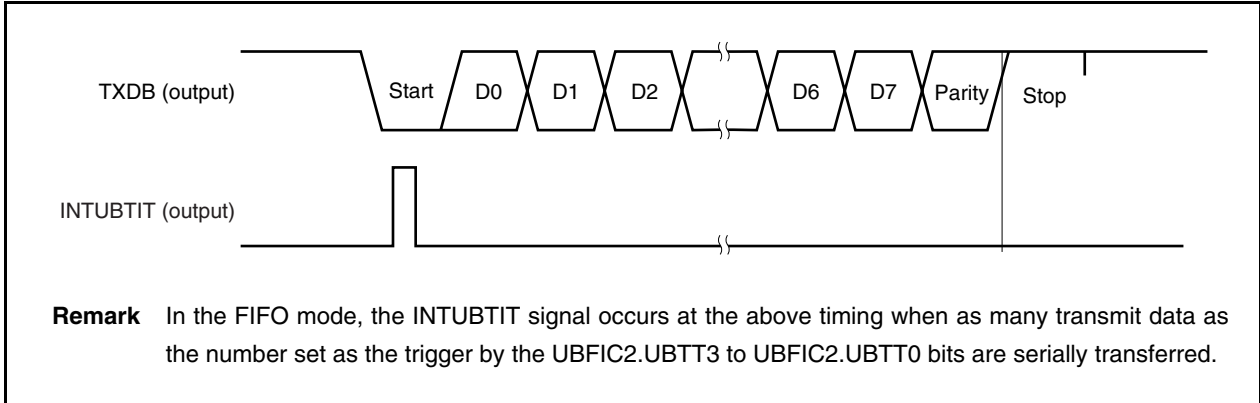
**(b) FIFO transmission end interrupt request signal (INTUBTIF)**

The FIFO transmission end interrupt request signal (INTUBTIF) occurs when no more data is in transmit FIFO and the transmit shift register in the FIFO mode (UBFIC0.UBMOD bit = 1). After the INTUBTIF signal has occurred, clear the pending INTUBTIT signal in the pending mode (UBFIC0.UBITM bit = 0) by clearing the FIFO (UBFIC0.UBTFC bit = 1). If the INTUBTIF signal occurs because writing the next transmit data to transmit FIFO is delayed (if all transmit data have not been transmitted), do not clear the FIFO.

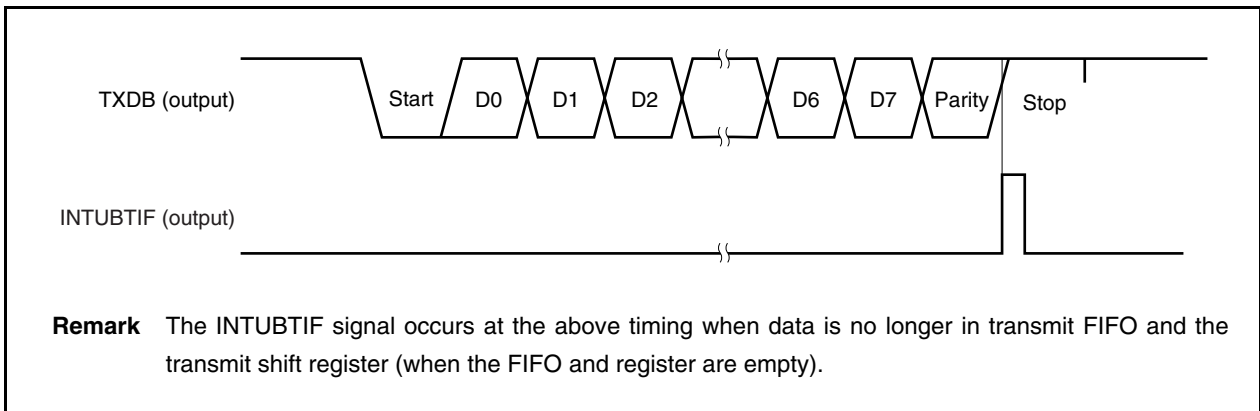
If the data to be transmitted next has not been written to the transmit data register, the transmit operation is suspended.

**Caution** In the single mode, the transmission enable interrupt request signal (INTUBTIT) occurs when the UBTX register becomes empty (when 1 byte of data is transferred from the UBTX register to the transmit shift register). In the FIFO mode, the FIFO transmission end interrupt request signal (INTUBTIF) occurs when data is no longer in transmit FIFO and the transmit shift register (when the FIFO and register are empty). However, the INTUBTIT signal or INTUBTIF signal is not generated if the transmit data register becomes empty due to RESET input.

**Figure 15-4. Timing of Asynchronous Serial Interface Transmission Enable Interrupt Request Signal (INTUBTIT)**



**Figure 15-5. Timing of Asynchronous Serial Interface FIFO Transmission End Interrupt Request Signal (INTUBTIF)**



### 15.7.3 Continuous transmission operation

- **In single mode (UBFIC0.UBMOD bit = 0)**

In the single mode, the next data can be written to the UBTX register as soon as the transmit shift register has started a shift operation. The timing of transfer can be identified by the transmission enable interrupt request signal (INTUBTIT). By writing the next transmit data to the UBTX register via the INTUBTIT signal within one data frame transmission period, data can be transmitted without an interval and an efficient communication rate can be realized.

**Caution** Confirm that the UBSTR.UBTSF bit is 0 before executing initialization during transmission processing. If initialization is executed while the UBTSF bit is 1, the transmit data is not guaranteed.

- **If pending mode is specified (UBFIC0.UBITM bit = 0) in FIFO mode**

If transmit data of at least the number set as the transmit trigger by UBFIC2.UBTT3 to UBFIC2.UBTT0 bits and 16 bytes or less is written to transmit FIFO, transmission starts.

If the pending mode is specified in the FIFO mode, as many of the next transmit data as the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits can be written to transmit FIFO as soon as the transmit shift register has started shifting the last data of the specified number of data. The timing of transfer can be identified by the INTUBTIT signal. By writing as many of the next transmit data as the number set as the trigger to transmit FIFO or writing the data to the FIFO within the transmission period of the data in transmit FIFO via the INTUBTIT signal, data can be transmitted without an interval and an efficient communication rate can be realized.

**Caution** Confirm that the UBSTR.UBTSF bit is 0 before executing initialization during transmission processing (this can also be done by the FIFO transmission end interrupt request signal (INTUBTIF)). If initialization is executed while the UBTSF bit is 1, the transmit data is not guaranteed. To write transmit data to transmit FIFO by DMA, set the number of transmit data specified as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits to 1 byte; otherwise the operation will not be guaranteed.

- **If pointer mode is specified (UBFIC0.UBITM bit = 1) in FIFO mode**

If the pointer mode is specified in the FIFO mode, a INTUBTIT signal occurs and the next data can be written to transmit FIFO as soon as the transmit shift register has started shifting the number of transmit data set as the trigger. At this time, as many data as the number of empty bytes of transmit FIFO can be written by referencing the UBFIS1 register. The timing of transfer can be identified by the INTUBTIT signal. By writing as many of the next transmit data as the number specified as the trigger to transmit FIFO or writing the data to the FIFO within the transmission period of the data in transmit FIFO via the INTUBTIT signal, data can be transmitted without an interval and an efficient communication rate can be realized.

**Caution** Confirm that the UBSTR.UBTSF bit is 0 before executing initialization during transmission processing (this can also be done by the FIFO transmission end interrupt request signal (INTUBTIF)). If initialization is executed while the UBTSF bit is 1, the transmit data is not guaranteed.

#### 15.7.4 Receive operation

The awaiting reception state is set by setting the UBCTL0.UBPWR bit to 1 and then setting the UBCTL0.UBRXE bit to 1. RXDB pin sampling begins and a start bit is detected. When the start bit is detected, the receive operation begins, and data is stored sequentially in the receive shift register according to the baud rate that was set.

In the single mode (UBFIC0.UBMOD bit = 0), a reception end interrupt request signal (INTUBTIR) is generated each time the reception of one frame of data is completed. Normally, the receive data is transferred from the UBRX register to memory by this interrupt servicing.

In the FIFO mode (UBFIC0.UBMOD bit = 1), the INTUBTIR signal occurs when as many receive data as the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits are transferred to receive FIFO.

If the pending mode is specified (UBFIC0.UBIRM bit = 0) in the FIFO mode, as many receive data as the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits can be read from receive FIFO.

If the pointer mode is specified (UBFIC0.UBIRM bit = 1) in the FIFO mode, as many data as the number of bytes stored in receive FIFO (0 bytes or more) can be read from receive FIFO by referencing the number of receive data specified as the trigger by the UBRT3 to UBRT0 bits (1 byte) or the UBFIS0 register.

**Caution** If the pointer mode is specified in the FIFO mode and if as many data as the number of bytes stored in receive FIFO are read by referencing the UBFIS0 register, no data may be stored in receive FIFO (UBFIS0.UBRB4 to UBFIS0.UBRB0 bits = 00000) even though the reception end interrupt request signal (INTUBTIR) has occurred. In this case, do not read data from receive FIFO. Be sure to read data from receive FIFO after confirming that the number of bytes stored in receive FIFO = 1 byte or more (UBRB4 to UBRB0 bits = other than 00000).

##### (1) Reception enabled state

This state is set by the UBCTL0.UBRXE bit.

- UBRXE = 1: Reception enabled state
- UBRXE = 0: Reception disabled state

However, because this bit is also used by CSIB2, enable reception after setting the CB2CTL0.CB2PWR bit to 0 and disabling the CSIB2 operation.

In the reception disabled state, the reception hardware stands by in the initial state. At this time, the reception end interrupt request signal or reception error interrupt request signal does not occur, and the contents of the receive data register (UBRX register in the single mode or receive FIFO in the FIFO mode (UBRXAP register)) are retained.

##### (2) Starting a receive operation

A receive operation is started by the detection of a start bit.

The RXDB pin is sampled using the serial clock from UARTB control register 2 (UBCTL2).

**(3) Reception interrupt request signal****(a) Reception end interrupt request signal (INTUBTIR)****• In single mode (UBFIC0.UBMOD bit = 0)**

When UBCTL0.UBRXE bit = 1 and the reception of one frame of data is ended (the stop bit is detected) in the single mode, a reception end interrupt request signal (INTUBTIR) is generated and the receive data in the receive shift register is transferred to the UBRX register at the same time.

Also, if an overrun error occurs, the receive data at that time is not transferred to the UBRX register, and a reception error interrupt request signal (INTUBTIRE) is generated.

If a parity error or framing error occurs during the reception operation, the reception operation continues up to the position at which the stop bit is received. After completion of reception, an INTUBTIRE signal occurs (the receive data in the receive shift register is transferred to the UBRX register).

If the UBRXE bit is reset (0) during a receive operation, the receive operation is immediately stopped. At this time, the contents of the UBRX register remain unchanged, the contents of the UARTB status register (UBSTR) are cleared, and the INTUBTIR and INTUBTIRE signals do not occur.

No INTUBTIR signal is generated when the UBRXE bit = 0 (reception is disabled).

**• In FIFO mode (UBFIC0.UBMOD bit = 1)**

In the FIFO mode, the reception end interrupt request signal (INTUBTIR) occurs when data of one frame has been received (stop bit is detected) and when as many receive data as the number specified as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits are transferred from the receive shift register to receive FIFO. If an overflow error occurs, the receive data is not transferred to receive FIFO and the reception error interrupt request signal (INTUBTIRE) occurs.

If a parity error or framing error occurs during reception, reception continues up to the reception position of the stop bit. After reception has been completed, the INTUBTIRE signal occurs and the receive data in the receive shift register is transferred to receive FIFO. At this time, error information is appended as the UBRXAP.UBPEF or UBRXAP.UBFEF bit = 1. If the INTUBTIRE signal occurs, the error data can be recognized by reading receive FIFO as a 16-bit register, UBRXAP.

**(b) Reception timeout interrupt request signal (INTUBTITO) (only in FIFO mode)**

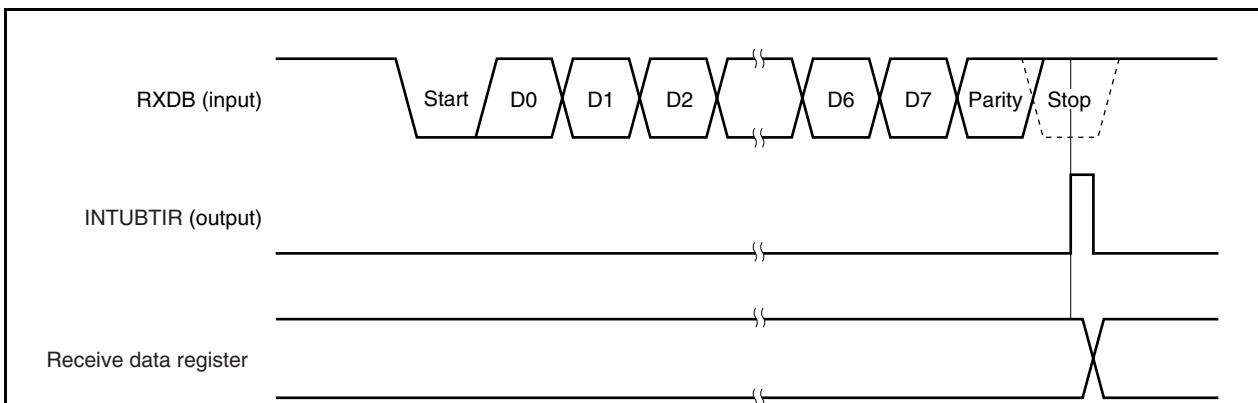
When the timeout counter function (UBFIC1.UBTCE bit = 1) is used in the FIFO mode, the reception timeout interrupt request signal (INTUBTITO) occurs if the next data does not come even after the next data reception wait time specified by the UBFIC1.UBTC4 to UBFIC1.UBTC0 bits has elapsed and if data is stored in receive FIFO.

The INTUBTITO signal does not occur while reception is disabled.

If as many receive data as the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits are not received, the timing of reading less receive data than the specified number can be set by the INTUBTITO signal.

Since the timeout counter starts counting at start bit detection, a receive timeout interrupt request signal does not occur if data of 1 character has not been received.

**Figure 15-6. Timing of Asynchronous Serial Interface Reception End Interrupt Request Signal (INTUBTIR)**



**Cautions 1. Be sure to read all the data (the number of data indicated by the UBFIS0.UBRB4 to UBFIS0.UBRB0 bits) stored in the receive data register (UBRX register in the single mode or receive FIFO in the FIFO mode (UBRXAP register)) even when a reception error occurs.**

Unless the receive data register is read, an overrun error occurs when the next data is received, causing the reception error status to persist.

If the pending mode is specified in the FIFO mode, however, be sure to clear the FIFO (UBFIC0.UBRFC bit = 1) after reading the data stored in receive FIFO.

In the FIFO mode, the FIFO can be cleared even without reading the data stored in receive FIFO.

If a parity error or framing error occurs in the FIFO mode, the UBRXAP register can be read in 16-bit (halfword) units.

2. Data is always received with one stop bit (1).  
A second stop bit is ignored.



**15.7.5 Reception error**

In the single mode (UBFIC0.UBMOD bit = 0), the three types of errors that can occur during a receive operation are a parity error, framing error, and overrun error. In the FIFO mode (UBFIC0.UBMOD bit = 1), the three types of errors that can occur during a receive operation are a parity error, framing error, and overflow error.

As a result of data reception, the UBSTR.UBPE, UBSTR.UBFE, or UBSTR.UBOVE bit is set to 1 if a parity error, framing error, or overrun error occurs in the single mode. The UBSTR.UBOVF bit is set to 1 if an overflow error occurs in the FIFO mode. The UBRXAP.UBPEF or UBRXAP.UBFEF bit is set to 1 if a parity error or framing error occurs in the FIFO mode. At the same time, a reception error interrupt request signal (INTUBTIRE) occurs. The contents of the error can be detected by reading the contents of the UBSTR or UBRXAP register.

The contents of the UBSTR register are reset when 0 is written to the UBOVF, UBPE, UBFE, or UBOVE bit, or the UBCTL0.UBPWR or UBCTL0.UBRXE bit. The contents of the UBRXAP register are reset when 0 is written to the UBCTL0.UBPWR bit.

**Table 15-4. Reception Error Causes**

| Error Flag | Valid Operation Mode | Error Flag | Reception Error | Cause  |
|------------|----------------------|------------|-----------------|--|
| UBPE       | Single mode          | UBPE       | Parity error    | The parity specification during transmission does not match the parity of the receive data         |
| UBFE       |                      | UBFE       | Framing error   | No stop bit detected   |
| UBOVE      |                      | UBOVE      | Overrun error   | The reception of the next data is ended before data is read from the UBRX register                 |
| UBOVF      | FIFO mode            | UBOVF      | Overflow error  | The reception of the next data is ended while receive FIFO is full and before data is read.        |
| UBPEF      |                      | UBPEF      | Parity error    | The parity specification during transmission does not match the parity of the data to be received. |
| UBFEF      |                      | UBFEF      | Framing error   | The stop bit is not detected when the target data is loaded.                                       |

### 15.7.6 Parity types and corresponding operation

A parity bit is used to detect a bit error in communication data. Normally, the same type of parity bit is used at the transmission and reception sides.

#### (1) Even parity

##### (a) During transmission

The parity bit is controlled so that the number of bits with the value "1" within the transmit data including the parity bit is even. The parity bit value is as follows.

- If the number of bits with the value "1" within the transmit data is odd: 1
- If the number of bits with the value "1" within the transmit data is even: 0

##### (b) During reception

The number of bits with the value "1" within the receive data including the parity bit is counted, and a parity error is generated if this number is odd.

#### (2) Odd parity

##### (a) During transmission

In contrast to even parity, the parity bit is controlled so that the number of bits with the value "1" within the transmit data including the parity bit is odd. The parity bit value is as follows.

- If the number of bits with the value "1" within the transmit data is odd: 0
- If the number of bits with the value "1" within the transmit data is even: 1

##### (b) During reception

The number of bits with the value "1" within the receive data including the parity bit is counted, and a parity error is generated if this number is even.

#### (3) 0 parity

During transmission the parity bit is set to "0" regardless of the transmit data.

During reception, no parity bit check is performed. Therefore, no parity error is generated regardless of whether the parity bit is "0" or "1".

#### (4) No parity

No parity bit is added to the transmit data.

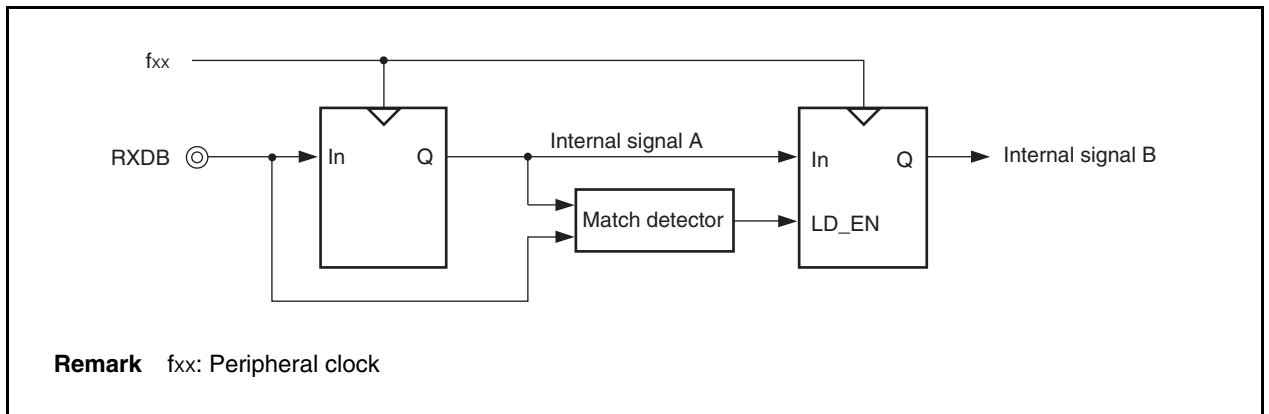
During reception, the receive operation is performed as if there were no parity bit. Since there is no parity bit, no parity error is generated.

**15.7.7 Receive data noise filter**

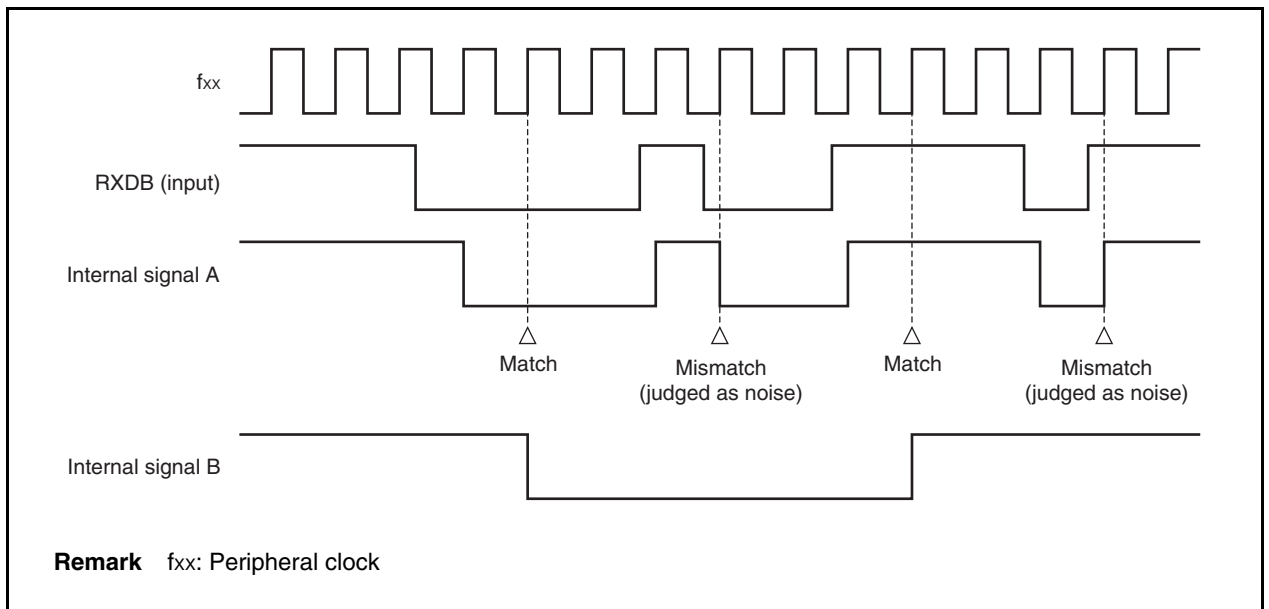
The RXDB signal is sampled at the rising edge of the input clock ( $f_{xx}$ ). If the same sampling value is obtained twice, the match detector output changes, and this output is sampled as input data. Therefore, data not exceeding one clock width is judged to be noise and is not delivered to the internal circuit (see **Figure 15-8**).

Also, since the circuit is configured as shown in Figure 15-7, internal processing during a receive operation is delayed by up to 2 clocks according to the external signal status.

**Figure 15-7. Noise Filter Circuit**



**Figure 15-8. Timing of RXDB Signal Judged as Noise**



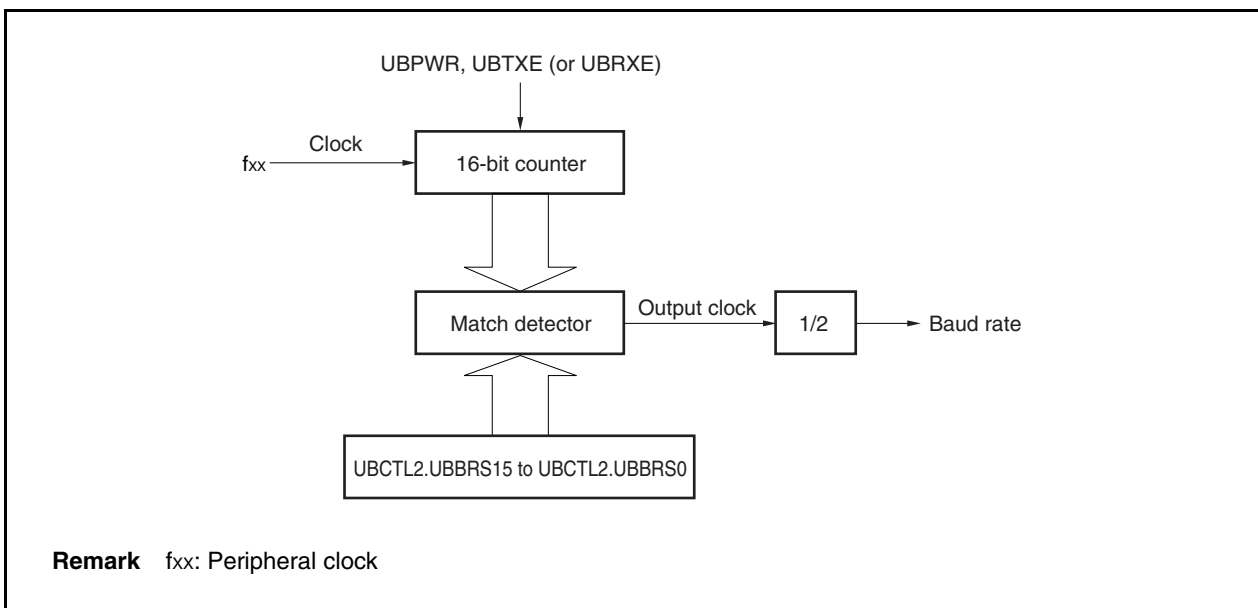
### 15.8 Dedicated Baud Rate Generator (BRG)

A dedicated baud rate generator, which consists of a 16-bit programmable counter, generates serial clocks during transmission/reception in UARTB. The dedicated baud rate generator output can be selected as the serial clock for each channel.

Separate 16-bit counters exist for transmission and for reception. The baud rate for transmission/reception is the same at the same channel.

#### (1) Baud rate generator configuration

Figure 15-9. Baud Rate Generator Configuration



#### (a) Base clock (Clock)

When  $UBCTL0.UBPWR$  bit = 1, the input clock ( $f_{xx}$ ) is supplied to the transmission/reception unit. This clock is called the base clock. When the  $UBPWR$  bit = 0, the clock signal is fixed at low level.

**(2) Serial clock generation**

A serial clock can be generated according to the settings of the UBCTL2 register.

The 16-bit counter divisor value can be selected according to the UBCTL2.UBBRS15 to UBCTL2.UBBRS0 bits.

**(a) Baud rate**

The baud rate is the value obtained according to the following formula.

$$\text{Baud rate} = \frac{\text{Base clock frequency}}{2 \times k} \text{ [bps]}$$

Base clock frequency = f<sub>xx</sub>

k = Value set according to UBCTL2.UBBRS15 to UBCTL2.UBBRS0 bits (k = 4, 5, 6, ..., 65535)

**(b) Baud rate error**

The baud rate error is obtained according to the following formula.

$$\text{Error (\%)} = \left( \frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (normal baud rate)}} - 1 \right) \times 100 \text{ [\%]}$$

**Cautions 1. Make sure that the baud rate error during transmission does not exceed the allowable error of the reception destination.**

**2. Make sure that the baud rate error during reception is within the allowable baud rate range during reception, which is described in paragraph (4).**

**Example:** Base clock (f<sub>xx</sub>) = 64 MHz = 64,000,000 Hz

Settings of UBCTL2.UBBRS15 to UBCTL2.UBBRS0 bits = 000000001100110B  
(k = 102)

Target baud rate = 312500 bps

Baud rate = 64 M/(2 × 102)  
= 64000000/(2 × 102) = 313725 [bps]

Error = (313725/312500 - 1) × 100  
= 0.392 [%]

When base clock (f<sub>xx</sub>) = 60 MHz and k = 96, the error is 0%.

(3) Baud rate setting example

Table 15-5. Baud Rate Generator Setting Data (1/2)

| Baud Rate<br>(bps) | f <sub>xx</sub> = 64 MHz |                    |        | f <sub>xx</sub> = 60 MHz |                    |        | f <sub>xx</sub> = 50 MHz |                    |        |
|--------------------|--------------------------|--------------------|--------|--------------------------|--------------------|--------|--------------------------|--------------------|--------|
|                    | k<br>(Decimal)           | k<br>(Hexadecimal) | ERR    | k<br>(Decimal)           | k<br>(Hexadecimal) | ERR    | k<br>(Decimal)           | k<br>(Hexadecimal) | ERR    |
| 300                | –                        | –                  | –      | –                        | –                  | –      | –                        | –                  | –      |
| 600                | 53333                    | D055H              | 0.001  | 50000                    | C350H              | 0.000  | 41667                    | A2C3H              | –0.001 |
| 1200               | 26667                    | 682BH              | –0.001 | 25000                    | 61A8H              | 0.000  | 20833                    | 5161H              | 0.002  |
| 2400               | 13333                    | 3415H              | 0.003  | 12500                    | 30D4H              | 0.000  | 10417                    | 28B1H              | –0.003 |
| 4800               | 6667                     | 1A0BH              | –0.005 | 6250                     | 186AH              | 0.000  | 5208                     | 1458H              | 0.006  |
| 9600               | 3333                     | 0D05H              | 0.010  | 3125                     | 0C35H              | 0.000  | 2604                     | 0A2CH              | 0.006  |
| 19200              | 1667                     | 0683H              | –0.020 | 1563                     | 061BH              | 0.000  | 1302                     | 0516H              | 0.006  |
| 31250              | 1024                     | 0400H              | 0.000  | 960                      | 03C0H              | 0.000  | 800                      | 0320H              | 0.000  |
| 38400              | 833                      | 0341H              | 0.040  | 781                      | 030DH              | 0.000  | 651                      | 028BH              | 0.006  |
| 76800              | 417                      | 01A1H              | –0.080 | 391                      | 0187H              | 0.000  | 326                      | 0146H              | –0.147 |
| 153600             | 208                      | 00D0H              | 0.160  | 195                      | 00C3H              | 0.000  | 163                      | 00A3H              | –0.147 |
| 312500             | 102                      | 0066H              | 0.392  | 96                       | 0060H              | 0.000  | 80                       | 0050H              | 0.000  |
| 500000             | 64                       | 0040H              | 0.000  | 60                       | 003CH              | 0.000  | 50                       | 0032H              | 0.000  |
| 1000000            | 32                       | 0020H              | 0.000  | 30                       | 001EH              | 0.000  | 25                       | 0019H              | 0.000  |
| 2000000            | 16                       | 0010H              | 0.000  | 15                       | 000FH              | 0.000  | 13                       | 000DH              | –3.846 |
| 3000000            | 11                       | 000BH              | –3.030 | 10                       | 000AH              | 0.000  | 8                        | 0008H              | 4.167  |
| 4000000            | 8                        | 0008H              | 0.000  | 8                        | 0008H              | –6.250 | –                        | –                  | –      |
| 5000000            | 6                        | 0006H              | 6.667  | 6                        | 0006H              | 0.000  | –                        | –                  | –      |
| 5333333            | 6                        | 0006H              | 0.000  | –                        | –                  | –      | –                        | –                  | –      |

**Caution** The maximum allowable frequency of the peripheral clock (f<sub>xx</sub>) is 64 MHz.  
The maximum transfer speed of the baud rate is 5.33 Mbps.

**Remark** f<sub>xx</sub>: Peripheral clock  
k: Settings of UBCTL2.UBBSR15 to UBCTL2.UBBSR0 bits  
ERR: Baud rate error [%]

Table 15-5. Baud Rate Generator Setting Data (2/2)

| Baud Rate<br>(bps) | f <sub>xx</sub> = 40 MHz |                    |        | f <sub>xx</sub> = 32 MHz |                    |        |
|--------------------|--------------------------|--------------------|--------|--------------------------|--------------------|--------|
|                    | k<br>(Decimal)           | k<br>(Hexadecimal) | ERR    | k<br>(Decimal)           | k<br>(Hexadecimal) | ERR    |
| 300                | –                        | –                  | –      | 53333                    | D055H              | 0.001  |
| 600                | 33333                    | 8235H              | 0.001  | 26667                    | 682BH              | –0.001 |
| 1200               | 16667                    | 411BH              | –0.002 | 13333                    | 3415H              | 0.003  |
| 2400               | 8333                     | 208DH              | 0.004  | 6667                     | 1A0BH              | –0.005 |
| 4800               | 4167                     | 1047H              | –0.008 | 3333                     | 0D05H              | 0.010  |
| 9600               | 2083                     | 0823H              | 0.016  | 1667                     | 0683H              | –0.020 |
| 19200              | 1042                     | 0412H              | –0.032 | 833                      | 0341H              | 0.040  |
| 31250              | 640                      | 0280H              | 0.000  | 512                      | 0200H              | 0.000  |
| 38400              | 521                      | 0209H              | –0.032 | 417                      | 01A1H              | –0.080 |
| 76800              | 260                      | 0104H              | 0.160  | 208                      | 00D0H              | 0.160  |
| 153600             | 130                      | 0082H              | 0.160  | 104                      | 0068H              | 0.160  |
| 312500             | 64                       | 0040H              | 0.000  | 51                       | 0033H              | 0.392  |
| 500000             | 40                       | 0028H              | 0.000  | 32                       | 0020H              | 0.000  |
| 1000000            | 20                       | 0014H              | 0.000  | 16                       | 0010H              | 0.000  |
| 2000000            | 10                       | 000AH              | 0.000  | 8                        | 0008H              | 0.000  |
| 3000000            | 7                        | 0007H              | –4.762 | –                        | –                  | –      |
| 4000000            | –                        | –                  | –      | –                        | –                  | –      |
| 5000000            | –                        | –                  | –      | –                        | –                  | –      |
| 5333333            | –                        | –                  | –      | –                        | –                  | –      |

**Caution** The maximum allowable frequency of the peripheral clock (f<sub>xx</sub>) is 64 MHz.  
The maximum transfer speed of the baud rate is 5.33 Mbps.

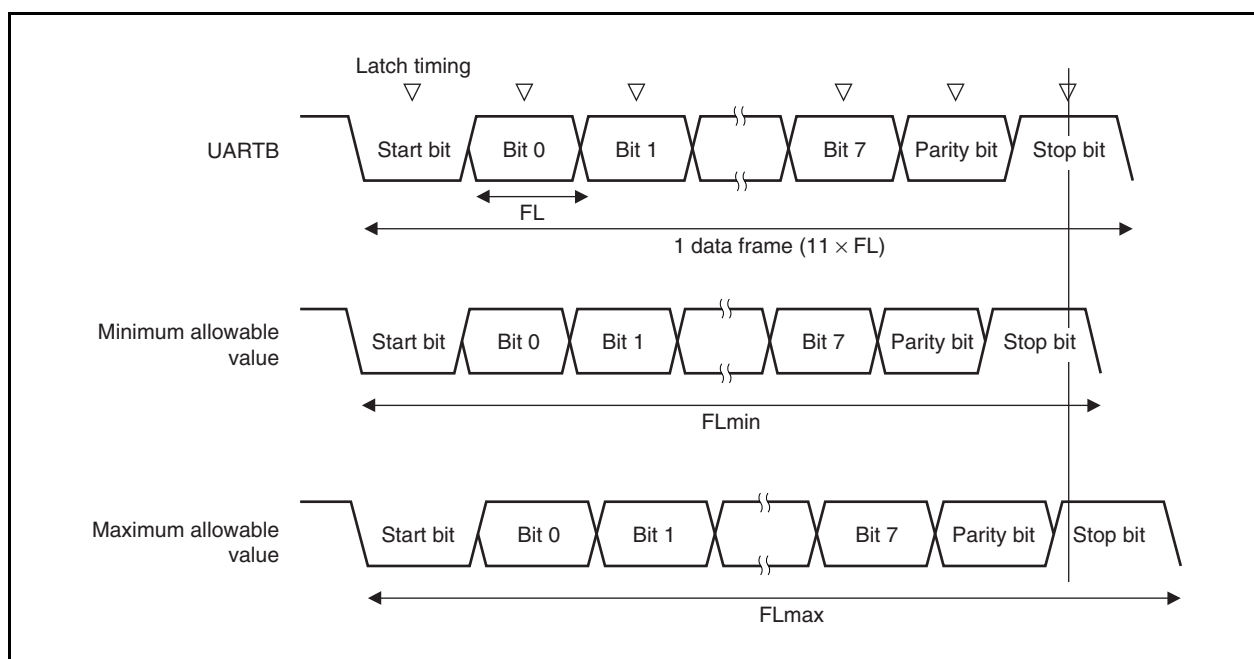
**Remark** f<sub>xx</sub>: Peripheral clock  
k: Settings of UBCTL2.UBBRS15 to UBCTL2.UBBRS0 bits  
ERR: Baud rate error [%]

**(4) Allowable baud rate range during reception**

The degree to which a discrepancy from the transmission destination's baud rate is allowed during reception is shown below.

**Caution** The equations described below should be used to set the baud rate error during reception so that it always is within the allowable error range.

**Figure 15-10. Allowable Baud Rate Range During Reception**



As shown in Figure 15-10, after the start bit is detected, the receive data latch timing is determined according to the counter that was set by the UBCTL2 register. If all data up to the final data (stop bit) is in time for this latch timing, the data can be received normally.

Applying this to 11-bit reception is, theoretically, as follows.

$$FL = (\text{Brate})^{-1}$$

Brate: UARTB baud rate

k: UBCTL2 set value

FL: 1-bit data length

Latch timing margin: 2 clocks

$$\text{Minimum allowable value: } FL_{\min} = 11 \times FL - \frac{k - 2}{2k} \times FL = \frac{21k + 2}{2k} FL$$



Therefore, the maximum baud rate that can be received at the transfer destination is as follows.

$$BR_{max} = (FL_{min}/11)^{-1} = \frac{22k}{21k + 2} \text{ Brate}$$

Similarly, the maximum allowable value can be obtained as follows.

$$\frac{10}{11} \times FL_{max} = 11 \times FL - \frac{k + 2}{2 \times k} \times FL = \frac{21k - 2}{2 \times k} FL$$

$$FL_{max} = \frac{21k - 2}{20k} FL \times 11$$

Therefore, the minimum baud rate that can be received at the transfer destination is as follows.

$$BR_{min} = (FL_{max}/11)^{-1} = \frac{20k}{21k - 2} \text{ Brate}$$

The allowable baud rate error of UARTB and the transfer destination can be obtained as follows from the expressions described above for computing the minimum and maximum baud rate values.

**Table 15-6. Maximum and Minimum Allowable Baud Rate Error**

| Division Ratio (k) | Maximum Allowable Baud Rate Error | Minimum Allowable Baud Rate Error |
|--------------------|-----------------------------------|-----------------------------------|
| 4                  | +2.33 %                           | -2.44                             |
| 8                  | +3.53 %                           | -3.61                             |
| 16                 | +4.14 %                           | -4.19                             |
| 32                 | +4.45 %                           | -4.48                             |
| 64                 | +4.61 %                           | -4.62                             |
| 128                | +4.68 %                           | -4.69                             |
| 256                | +4.72 %                           | -4.73                             |
| 512                | +4.74 %                           | -4.74                             |
| 1024               | +4.75 %                           | -4.75                             |
| 2048               | +4.76 %                           | -4.76                             |
| 4096               | +4.76 %                           | -4.76                             |
| 8192               | +4.76 %                           | -4.76                             |
| 16384              | +4.76 %                           | -4.76                             |
| 32768              | +4.76 %                           | -4.76                             |
| 65535              | +4.76 %                           | -4.76                             |

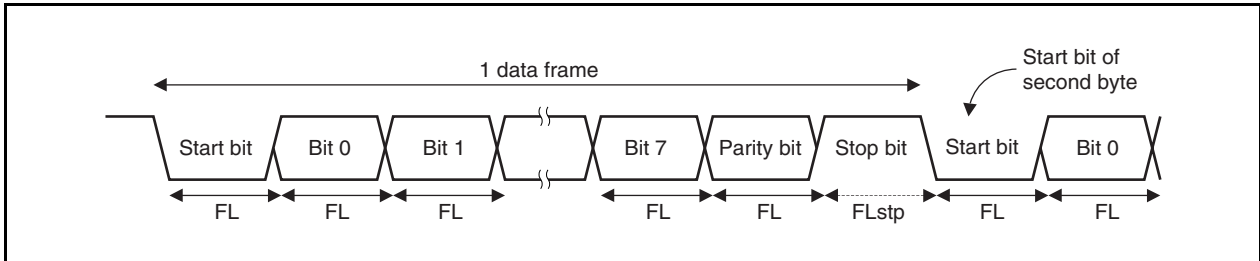
**Remarks 1.** The reception precision depends on the number of bits in one frame, the base clock frequency, and the division ratio (k). The higher the base clock frequency and the larger the division ratio (k), the higher the precision.

2. k: UBCTL2 set value

**(5) Transfer rate during continuous transmission**

During continuous transmission, the transfer rate from a stop bit to the next start bit is extended two clocks longer than normal. However, on the reception side, the transfer result is not affected since the timing is initialized by the detection of the start bit.

**Figure 15-11. Transfer Rate During Continuous Transmission**



Representing the 1-bit data length by FL, the stop bit length by FLstp, and the base clock frequency by fxx yields the following equation.

$$FLstp = FL + 2/(f_{xx})$$

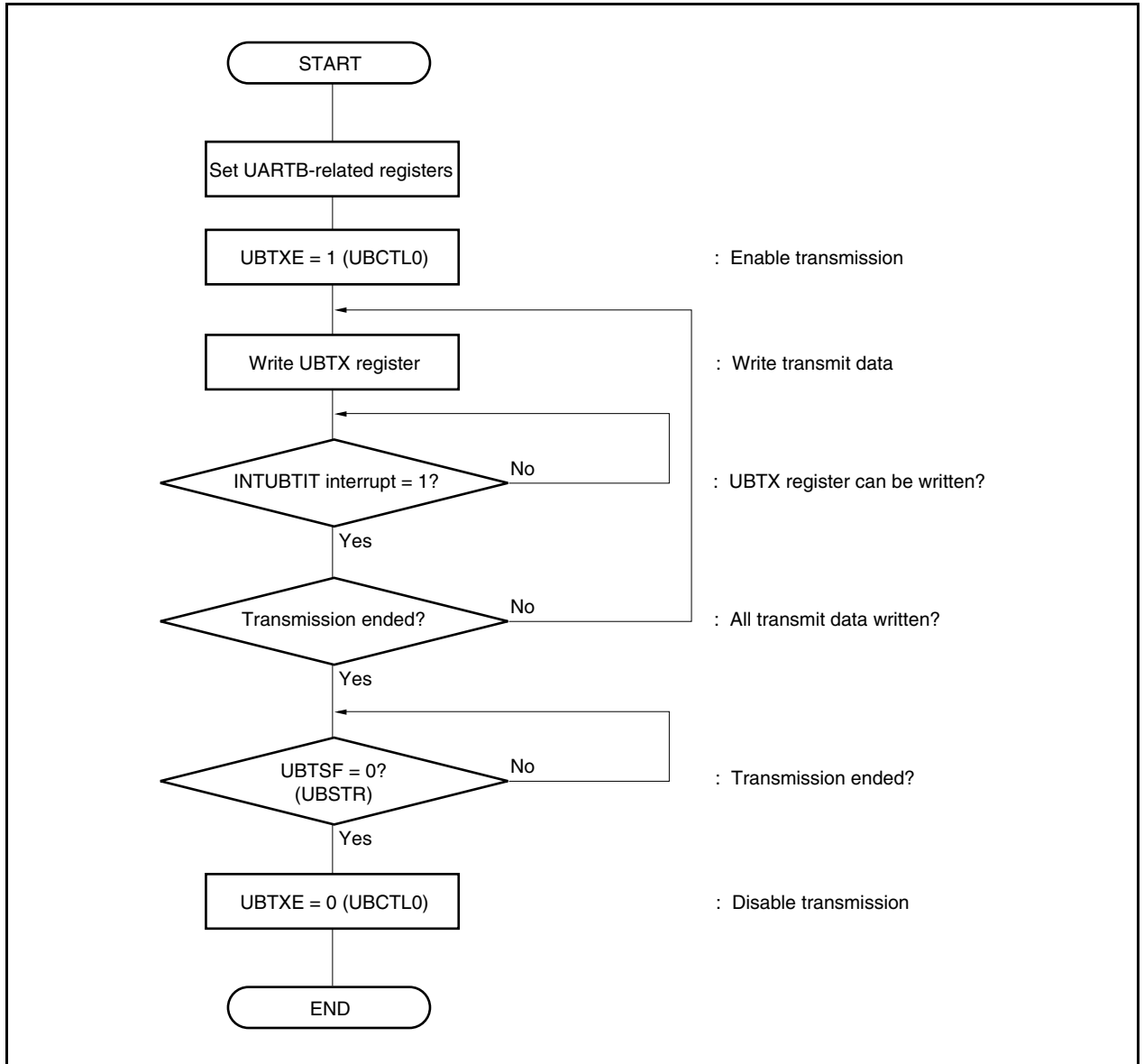
Therefore, the transfer rate during continuous transmission is as follows.

$$\text{Transfer rate} = 11 \times FL + 2/(f_{xx})$$

15.9 Control Flow

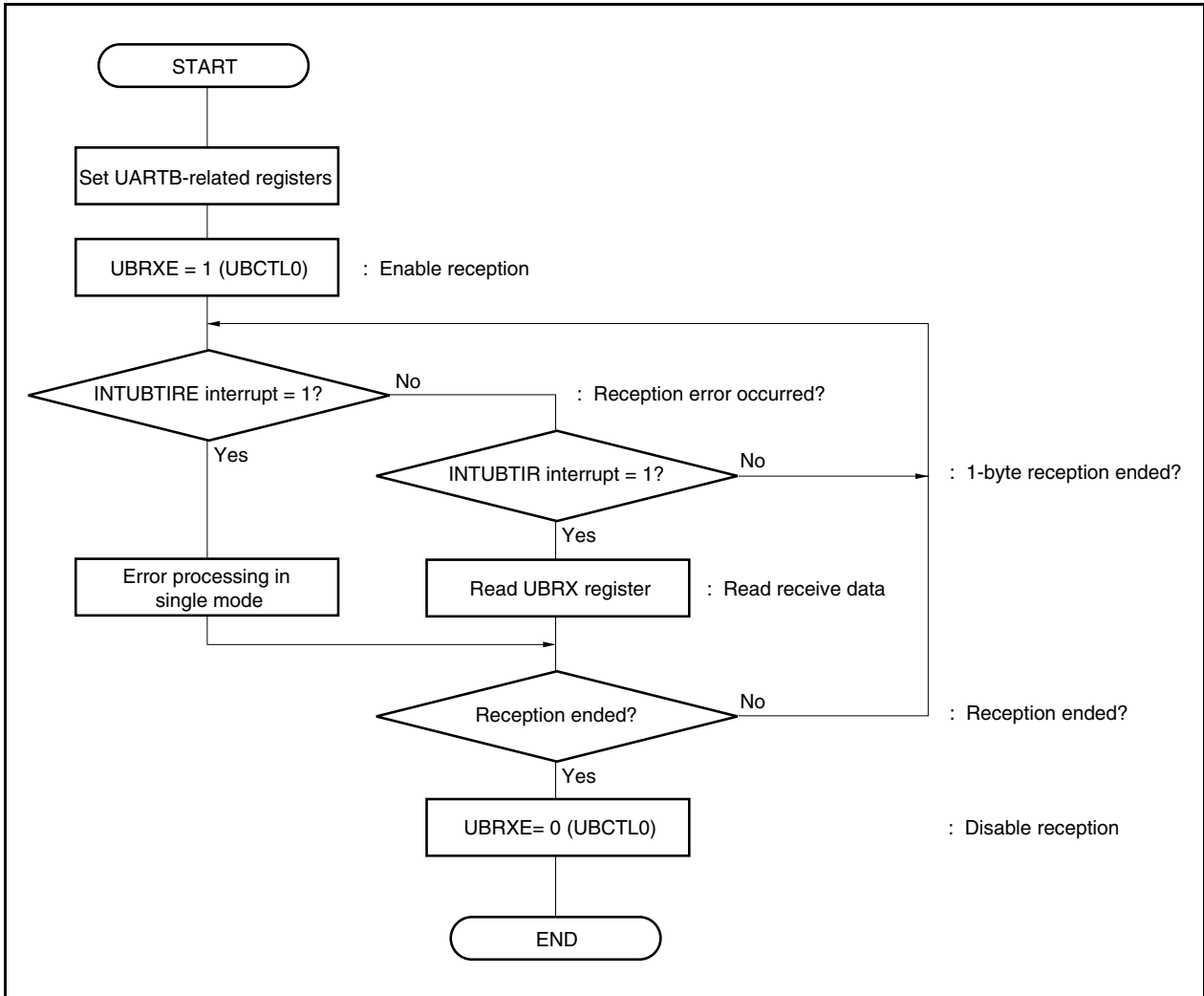
(1) Example of continuous transmission processing flow in single mode (CPU control)

Figure 15-12. Example of Continuous Transmission Processing Flow in Single Mode (CPU Control)



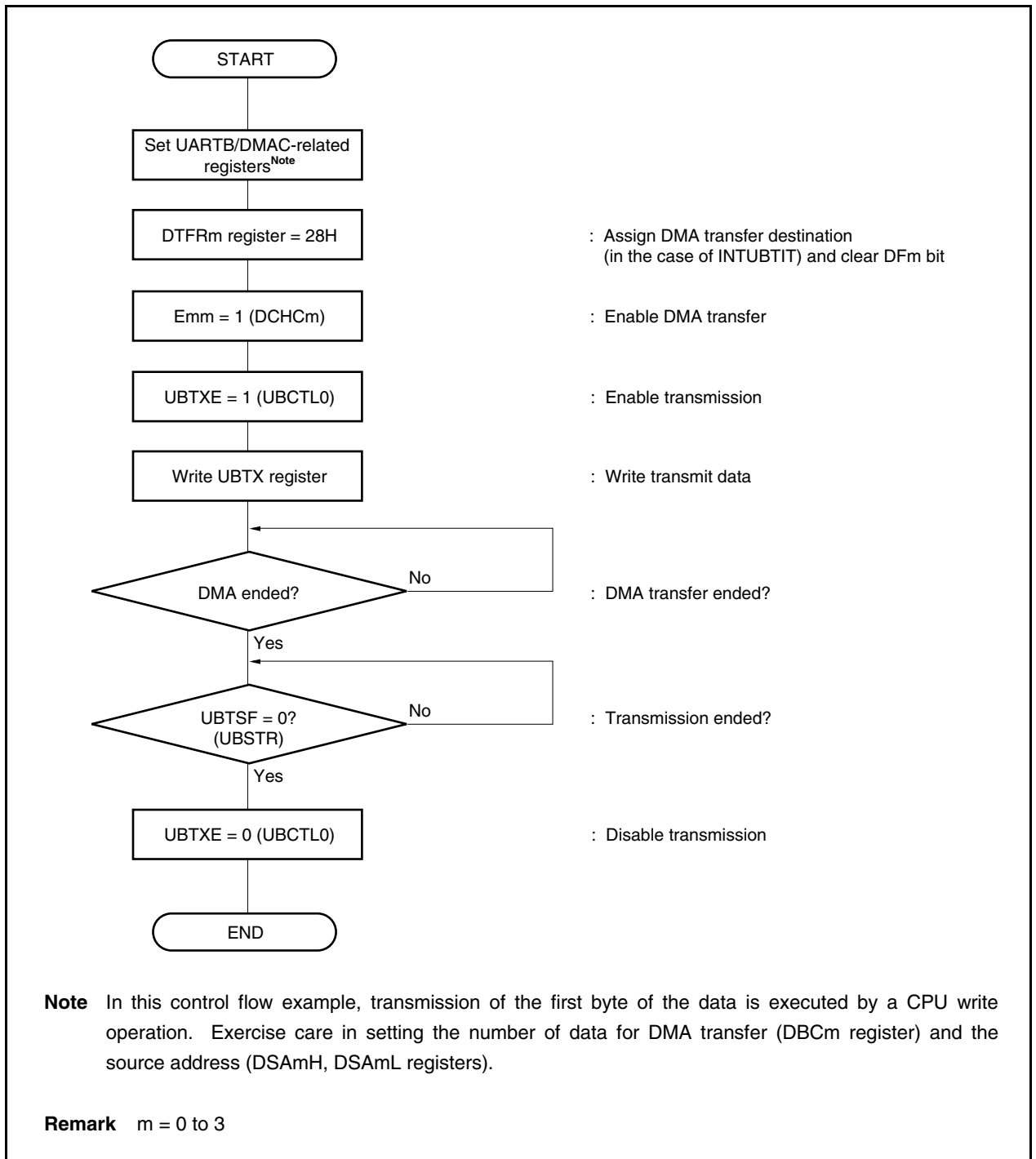
(2) Example of continuous reception processing flow in single mode (CPU control)

Figure 15-13. Example of Continuous Reception Processing Flow in Single Mode (CPU Control)



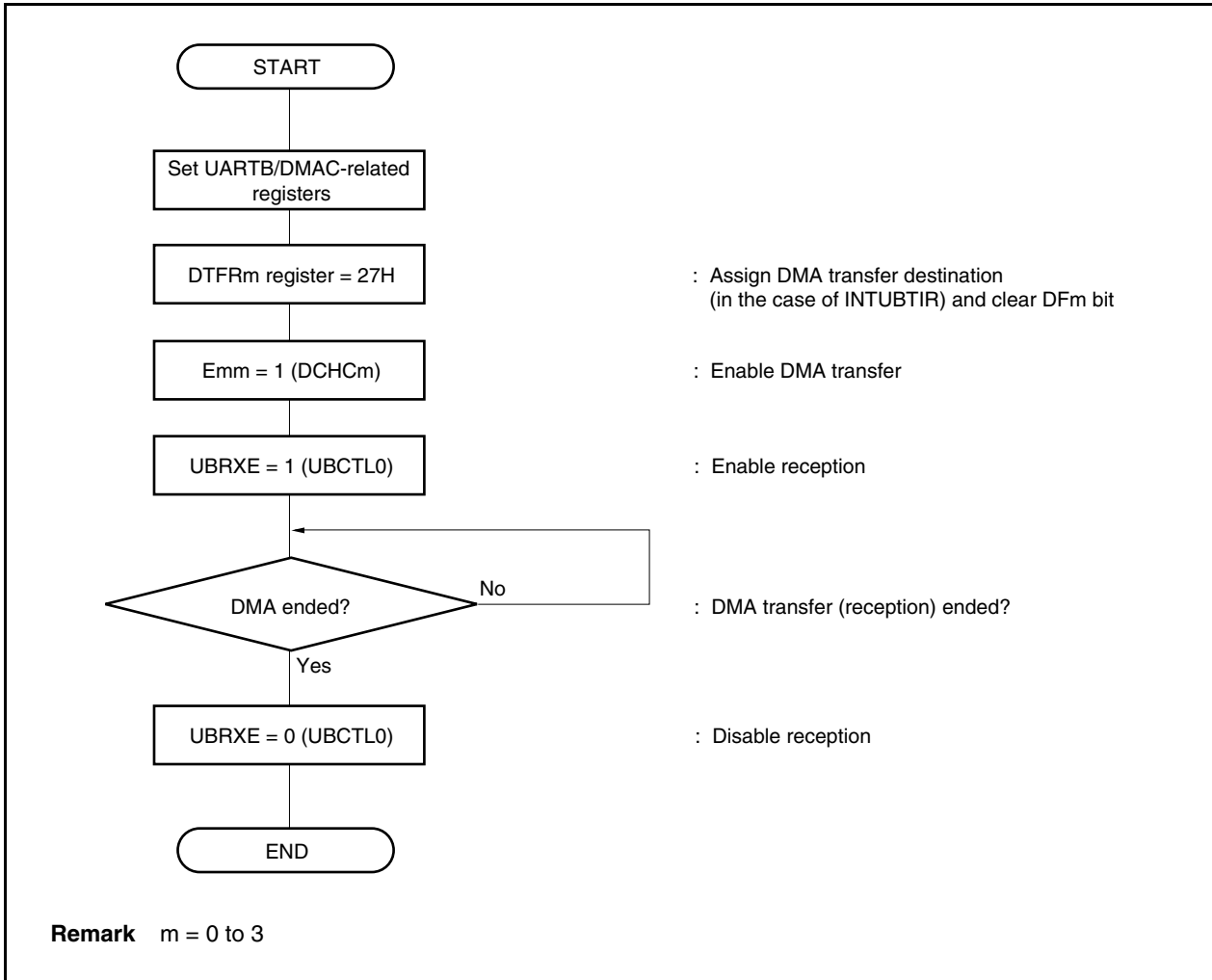
(3) Example of continuous transmission processing flow in single mode (DMA control)

Figure 15-14. Example of Continuous Transmission Processing Flow in Single Mode (DMA Control)



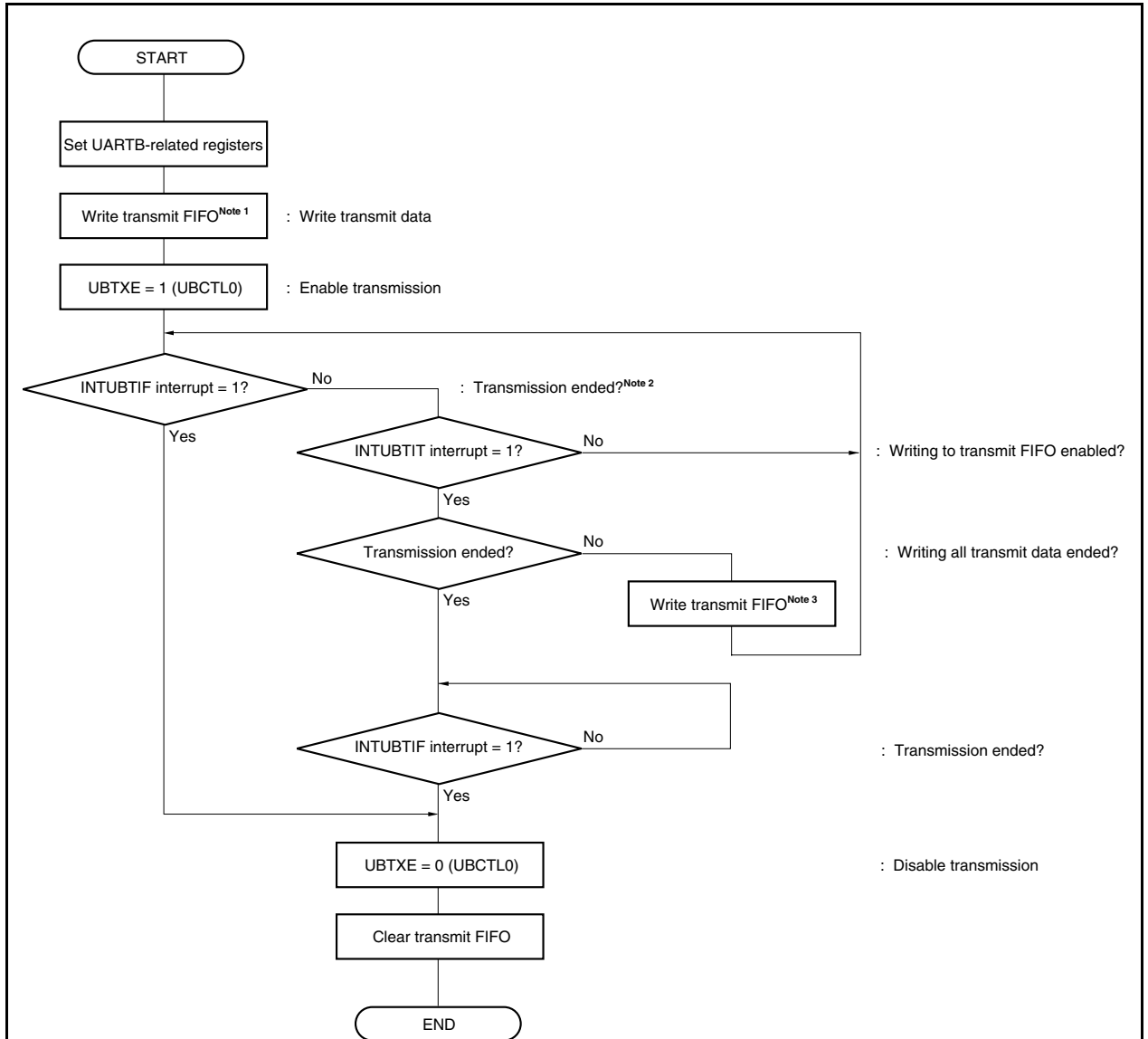
(4) Example of continuous reception processing flow in single mode (DMA control)

Figure 15-15. Example of Continuous Reception Processing Flow in Single Mode (DMA Control)



(5) Example of continuous transmission processing flow in FIFO mode (CPU control)

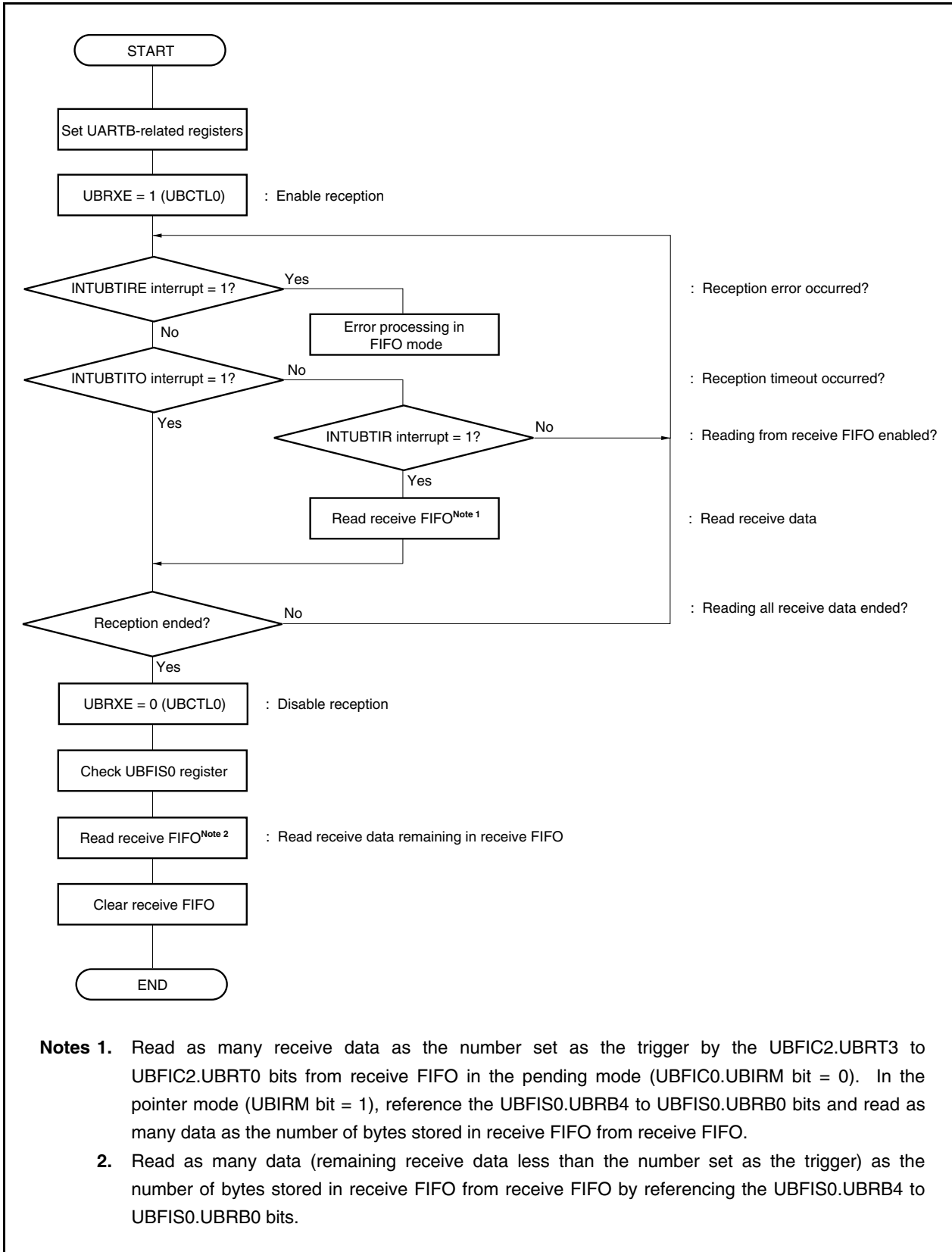
Figure 15-16. Example of Continuous Transmission Processing Flow in FIFO Mode (CPU Control)



- Notes**
1. Write more transmit data than the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits to transmit FIFO.
  2. This is the case where transmission is ended (transmit FIFO and the transmit shift register become empty) before the next transmit data is written. To continue data transmission, clear the INTUBTIF and INTUBTIT signals and write the next data to transmit FIFO.
  3. In the pending mode (UBFIC0.UBITM bit = 0), write as many transmit data as the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits of to transmit FIFO. In the pointer mode (UBITM bit = 1), reference the UBFIS1.UBTB4 to UBFIS1.UBTB0 bits and write as many data as the number of empty bytes in transmit FIFO to transmit FIFO. Write 16-byte data to fully use the 8-bit × 16-stage FIFO function.

(6) Example of continuous reception processing in FIFO mode (CPU control)

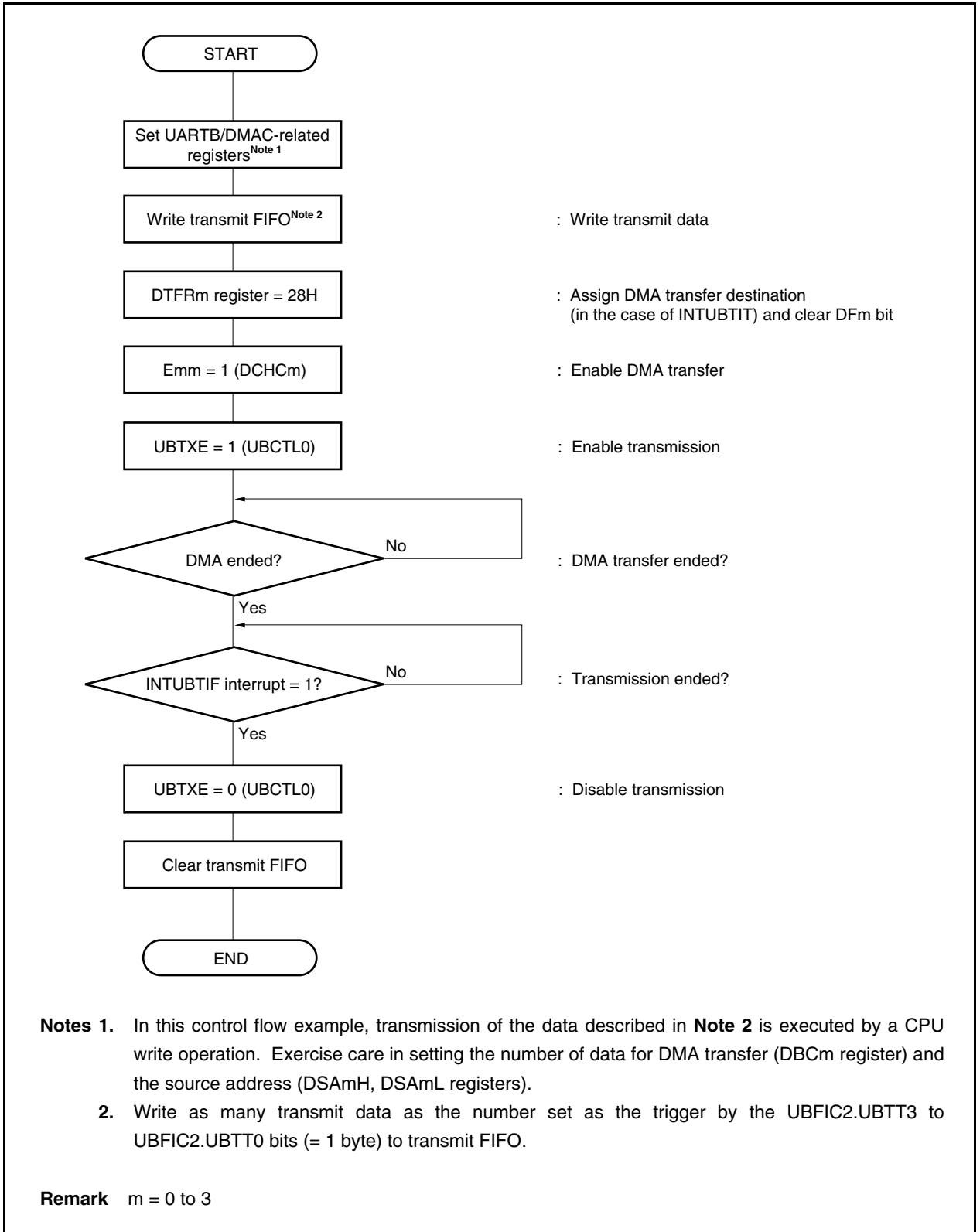
Figure 15-17. Example of Continuous Reception Processing in FIFO Mode (CPU Control)





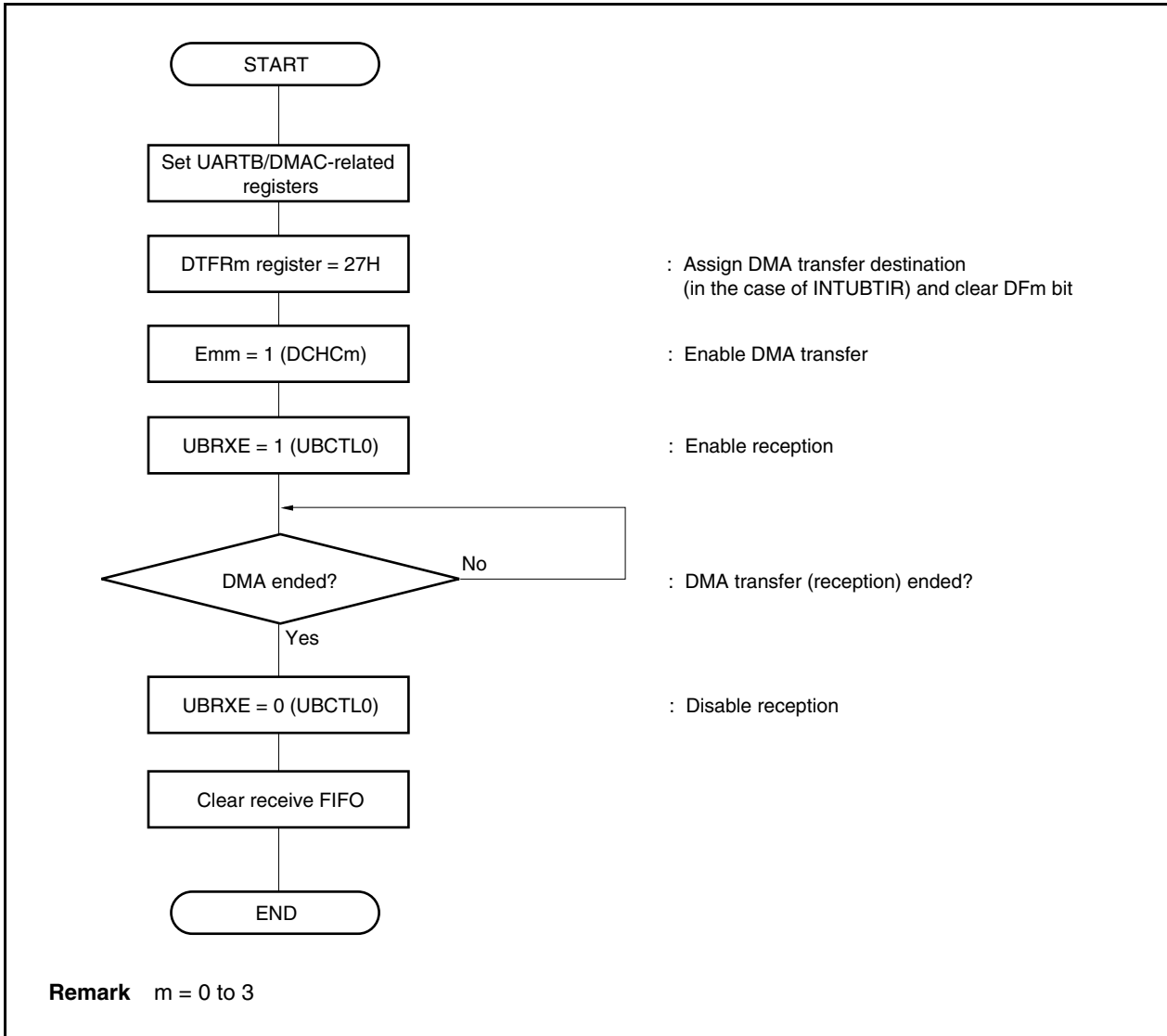
(7) Example of continuous transmission (pending mode) processing in FIFO mode (DMA control)

Figure 15-18. Example of Continuous Transmission (Pending Mode) Processing in FIFO Mode (DMA Control)



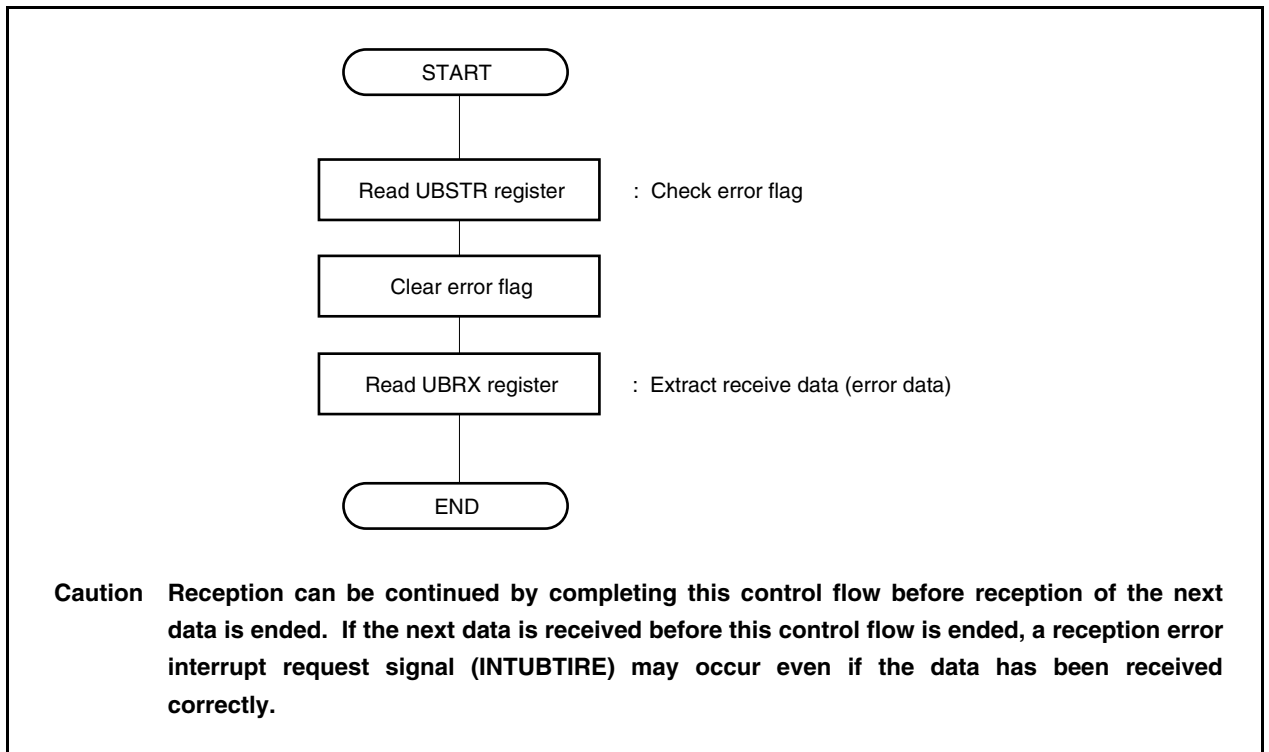
(8) Example of continuous reception (pending mode) processing flow in FIFO mode (DMA control)

Figure 15-19. Example of Continuous Reception (Pending Mode) Processing Flow in FIFO Mode (DMA Control)



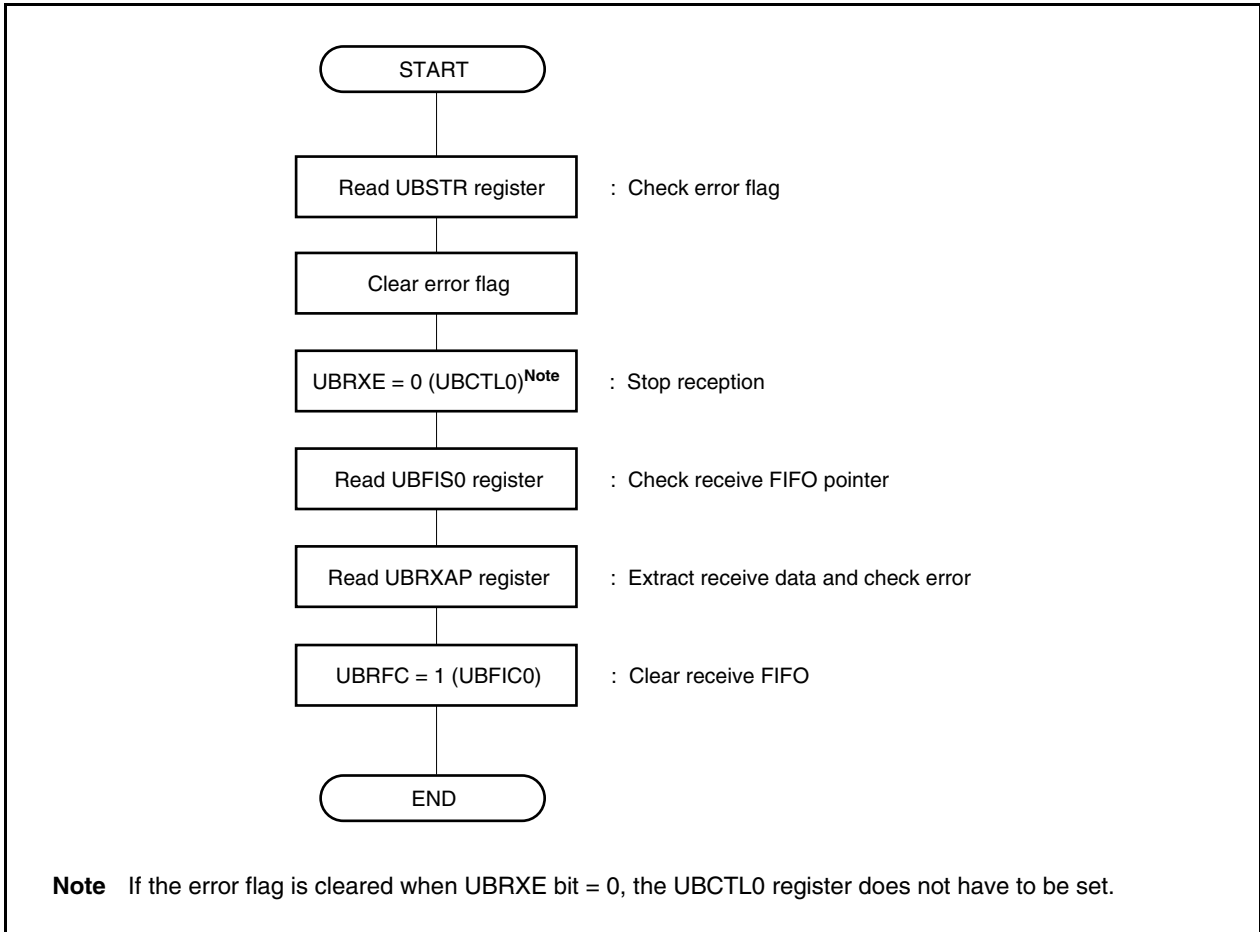
## (9) Example of reception error processing in single mode

Figure 15-20. Example of Reception Error Processing Flow in Single Mode



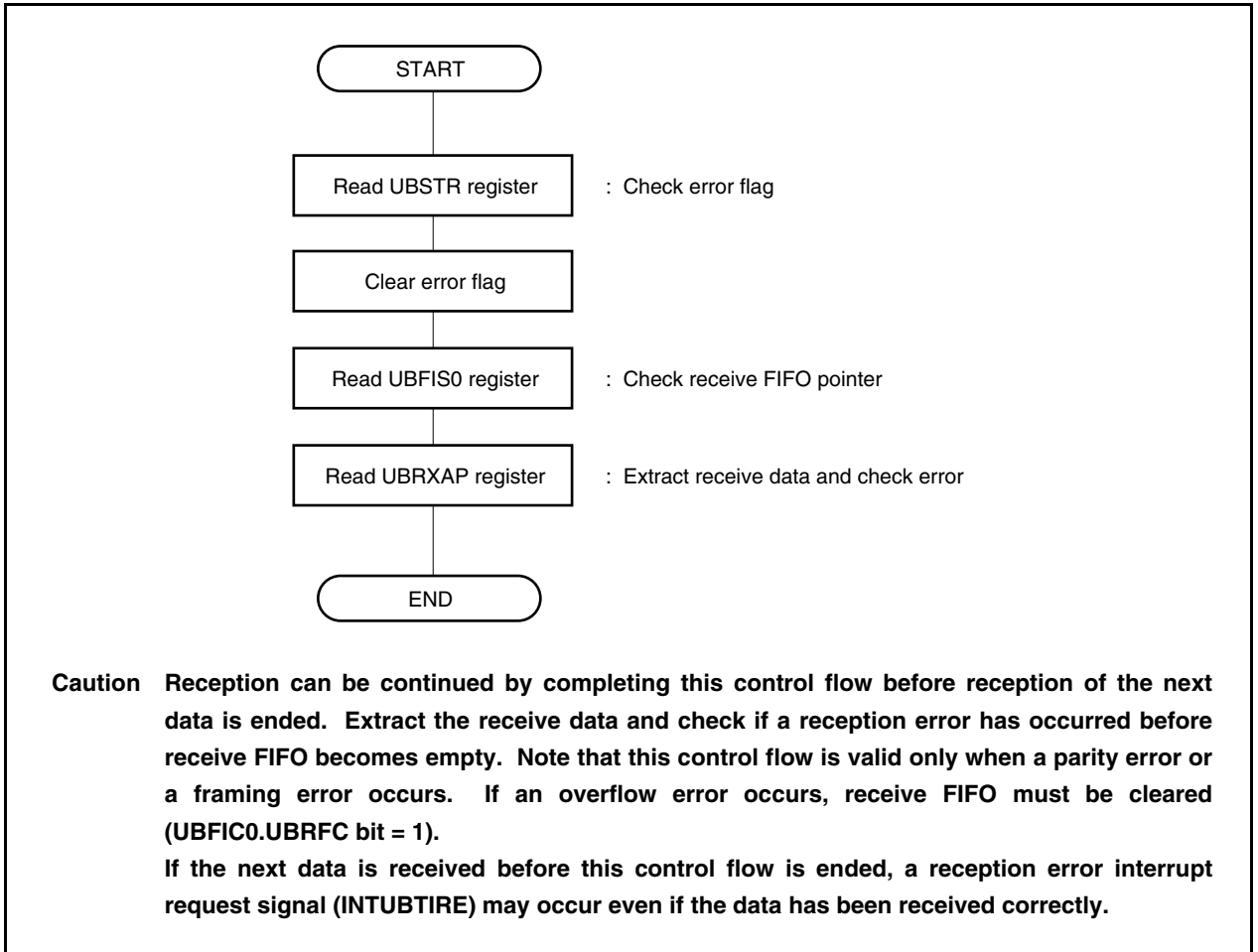
(10) Example of reception error processing flow in FIFO mode (1)

Figure 15-21. Example of Reception Error Processing Flow in FIFO Mode (1)



## (11) Example of reception error processing flow in FIFO mode (2)

Figure 15-22. Example of Reception Error Processing Flow in FIFO Mode (2)



## 15.10 Cautions

Cautions concerning UARTB are shown below.

### (1) When supply clock to UARTB is stopped

When the supply of clocks to UARTB is stopped (for example, IDLE and STOP modes), operation stops with each register retaining the value it had immediately before the supply of clocks was stopped. The TXDB pin output also holds and outputs the value it had immediately before the supply of clocks was stopped. However, operation is not guaranteed after the supply of clocks is restarted. Therefore, after the supply of clocks is restarted, the circuits should be initialized by setting the UBPWR bit = 0, UBRXE bit = 0, and UBTXE bit = 0.

### (2) Caution on setting UBCTL0 register

- When using UARTB, set the external pins related to the UARTB function to the alternate function and set the UBCTL2 register. Then set the UBCTL0.UBPWR bit to 1 before setting the other bits.
- Be sure to input a high level to the RXDB pin when setting the external pins related to the UARTB function to the alternate function. If a low level is input, it is judged that a falling edge is input after the UBCTL0.UBRXE bit has been set to 1, and reception may be started.

### (3) Caution on setting UBFIC2 register

Be sure to clear the UBCTL0.UBTXE bit (to disable transmission) and UBCTL0.UBRXE bit (to disable reception) to 0 before writing data to the UBFIC2 register. If data is written to the UBFIC2 register with the UBTXE or UBRXE bit set to 1, the operation is not guaranteed.

### (4) Transmission interrupt request signal

In the single mode, the transmission enable interrupt request signal (INTUBTIT) occurs when the UBTX register becomes empty (when 1 byte of data is transferred from the UBTX register to the transmit shift register). In the FIFO mode, the FIFO transmission end interrupt request signal (INTUBTIF) occurs when data is no longer in transmit FIFO and the transmit shift register (when the FIFO and register are empty). However, the INTUBTIT signal or INTUBTIF signal does not occur if the transmit data register becomes empty due to  $\overline{\text{RESET}}$  input.

### (5) Initialization during continuous transmission in single mode

Confirm that the UBSTR.UBTSF bit is 0 before executing initialization during transmission processing. If initialization is executed while the UBTSF bit is 1, the transmit data is not guaranteed.

### (6) Initialization during continuous transmission (pending mode) in FIFO mode

Confirm that the UBSTR.UBTSF bit is 0 before executing initialization during transmission processing (this can also be done by checking the FIFO transmission end interrupt request signal (INTUBTIF)). If initialization is executed while the UBTSF bit is 1, the transmit data is not guaranteed.

To write transmit data to transmit FIFO by DMA control, set the number of transmit data specified as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits to 1 byte; otherwise the operation will not be guaranteed.

### (7) Initialization during continuous transmission (pointer mode) in FIFO mode

Confirm that the UBSTR.UBTSF bit is 0 before executing initialization during transmission processing (this can also be done by checking the FIFO transmission end interrupt request signal (INTUBTIF)). If initialization is executed while the UBTSF bit is 1, the transmit data is not guaranteed.

**(8) Receive operation in FIFO mode (pointer mode specified)**

If the pointer mode is specified in the FIFO mode and if as many data as the number of bytes stored in receive FIFO are read by referencing the UBFIS0 register, no data may be stored in receive FIFO (UBFIS0.UBRB4 to UBFIS0.UBRB0 bits = 00000) even though the reception end interrupt request signal (INTUBTIR) has occurred. In this case, do not read data from receive FIFO. Be sure to read data from receive FIFO after confirming that the number of bytes stored in receive FIFO = 1 byte or more (UBRB4 to UBRB0 bits = other than 00000).

## CHAPTER 16 CLOCKED SERIAL INTERFACE B (CSIB)

### 16.1 Mode Switching Between CSIB and Other Serial Interface

#### 16.1.1 Mode switching between CSIB0 and UARTA0

In the V850E/IF3 and V850E/IG3, CSIB0 and UARTA0 function alternately, and these functions cannot be used at the same time. To use CSIB0 and UARTA0, the PMC4, PFC4, and PFCE4 registers must be set in advance.

**Caution** The operations related to transmission and reception of CSIB0 or UARTA0 are not guaranteed if the mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 16-1. Mode Switch Settings of CSIB0 and UARTA0

|                  |        |   |        |  |        |        |        |   |   |   |        |        |        |        |        |        |        |        |
|------------------|--------|---|--------|--|--------|--------|--------|---|---|---|--------|--------|--------|--------|--------|--------|--------|--------|
| After reset: 00H | R/W    | Address: FFFFF448H  |        |  |        |        |        |   |   |   |        |        |        |        |        |        |        |        |
| PMC4             |        | <table border="1"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>PMC47</td><td>PMC46</td><td>PMC45</td><td>PMC44</td><td>PMC43</td><td>PMC42</td><td>PMC41</td><td>PMC40</td> </tr> </table>         | 7      | 6  | 5      | 4      | 3      | 2 | 1 | 0 | PMC47  | PMC46  | PMC45  | PMC44  | PMC43  | PMC42  | PMC41  | PMC40  |
| 7                | 6      | 5   | 4      | 3  | 2      | 1      | 0      |   |   |   |        |        |        |        |        |        |        |        |
| PMC47            | PMC46  | PMC45   | PMC44  | PMC43  | PMC42  | PMC41  | PMC40  |   |   |   |        |        |        |        |        |        |        |        |
| After reset: 00H | R/W    | Address: FFFFF468H  |        |  |        |        |        |   |   |   |        |        |        |        |        |        |        |        |
| PFC4             |        | <table border="1"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>PFC47</td><td>PFC46</td><td>PFC45</td><td>PFC44</td><td>PFC43</td><td>PFC42</td><td>PFC41</td><td>PFC40</td> </tr> </table>         | 7      | 6  | 5      | 4      | 3      | 2 | 1 | 0 | PFC47  | PFC46  | PFC45  | PFC44  | PFC43  | PFC42  | PFC41  | PFC40  |
| 7                | 6      | 5   | 4      | 3  | 2      | 1      | 0      |   |   |   |        |        |        |        |        |        |        |        |
| PFC47            | PFC46  | PFC45   | PFC44  | PFC43  | PFC42  | PFC41  | PFC40  |   |   |   |        |        |        |        |        |        |        |        |
| After reset: 00H | R/W    | Address: FFFFF708H  |        |  |        |        |        |   |   |   |        |        |        |        |        |        |        |        |
| PFCE4            |        | <table border="1"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>PFCE47</td><td>PFCE46</td><td>PFCE45</td><td>PFCE44</td><td>PFCE43</td><td>PFCE42</td><td>PFCE41</td><td>PFCE40</td> </tr> </table> | 7      | 6  | 5      | 4      | 3      | 2 | 1 | 0 | PFCE47 | PFCE46 | PFCE45 | PFCE44 | PFCE43 | PFCE42 | PFCE41 | PFCE40 |
| 7                | 6      | 5   | 4      | 3  | 2      | 1      | 0      |   |   |   |        |        |        |        |        |        |        |        |
| PFCE47           | PFCE46 | PFCE45  | PFCE44 | PFCE43   | PFCE42 | PFCE41 | PFCE40 |   |   |   |        |        |        |        |        |        |        |        |
|                  | PMC42  | PFCE42  | PFC42  | Specification of alternate function of P42 pin |        |        |        |   |   |   |        |        |        |        |        |        |        |        |
|                  | 0      | ×   | ×      | I/O port                                       |        |        |        |   |   |   |        |        |        |        |        |        |        |        |
|                  | 1      | 0   | 0      | $\overline{\text{SCKB0}}$ I/O                  |        |        |        |   |   |   |        |        |        |        |        |        |        |        |
|                  | 1      | 0   | 1      | INTP13 input                                   |        |        |        |   |   |   |        |        |        |        |        |        |        |        |
|                  | 1      | 1   | 0      | Setting prohibited                             |        |        |        |   |   |   |        |        |        |        |        |        |        |        |
|                  | 1      | 1   | 1      | Setting prohibited                             |        |        |        |   |   |   |        |        |        |        |        |        |        |        |
|                  | PMC41  | PFCE41  | PFC41  | Specification of alternate function of P41 pin |        |        |        |   |   |   |        |        |        |        |        |        |        |        |
|                  | 0      | ×   | ×      | I/O port                                       |        |        |        |   |   |   |        |        |        |        |        |        |        |        |
|                  | 1      | 0   | 0      | SOB0 output                                    |        |        |        |   |   |   |        |        |        |        |        |        |        |        |
|                  | 1      | 0   | 1      | TXDA0 output                                   |        |        |        |   |   |   |        |        |        |        |        |        |        |        |
|                  | 1      | 1   | 0      | Setting prohibited                             |        |        |        |   |   |   |        |        |        |        |        |        |        |        |
|                  | 1      | 1   | 1      | Setting prohibited                             |        |        |        |   |   |   |        |        |        |        |        |        |        |        |
|                  | PMC40  | PFC40   |        | Specification of alternate function of P40 pin |        |        |        |   |   |   |        |        |        |        |        |        |        |        |
|                  | 0      | ×   |        | I/O port                                       |        |        |        |   |   |   |        |        |        |        |        |        |        |        |
|                  | 1      | 0   |        | SIB0 input                                     |        |        |        |   |   |   |        |        |        |        |        |        |        |        |
|                  | 1      | 1   |        | RXDA0 input                                    |        |        |        |   |   |   |        |        |        |        |        |        |        |        |

**Remark** × = don't care



16.1.2 Mode switching between CSIB1 and UARTA2

In the V850E/IF3 and V850E/IG3, CSIB1 and UARTA2 function alternately, and these functions cannot be used at the same time. To use CSIB1 and UARTA2, the PMC3, PFC3, and PFCE3 registers must be set in advance.

**Caution** The operations related to transmission and reception of CSIB1 or UARTA2 are not guaranteed if the mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 16-2. Mode Switch Settings of CSIB1 and UARTA2

After reset: 00H R/W Address: FFFFF446H

|      |       |       |       |       |       |       |       |       |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
|      | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| PMC3 | PMC37 | PMC36 | PMC35 | PMC34 | PMC33 | PMC32 | PMC31 | PMC30 |

After reset: 00H R/W Address: FFFFF466H

|      |       |       |       |       |       |       |       |       |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
|      | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| PFC3 | PFC37 | PFC36 | PFC35 | PFC34 | PFC33 | PFC32 | PFC31 | PFC30 |

After reset: 00H R/W Address: FFFFF706H

|       |        |        |        |        |   |        |        |        |
|-------|--------|--------|--------|--------|---|--------|--------|--------|
|       | 7      | 6      | 5      | 4      | 3 | 2      | 1      | 0      |
| PFCE3 | PFCE37 | PFCE36 | PFCE35 | PFCE34 | 0 | PFCE32 | PFCE31 | PFCE30 |

| PMC34 | PFCE34 | PFC34 | Specification of alternate function of P34 pin |
|-------|--------|-------|--|
| 0     | ×      | ×     | I/O port                                       |
| 1     | 0      | 0     | SCKB1 I/O                                      |
| 1     | 0      | 1     | INTP11 input                                   |
| 1     | 1      | 0     | CS0 <sup>Note</sup> output                     |
| 1     | 1      | 1     | Setting prohibited                             |

| PMC33 | PFC33 | Specification of alternate function of P33 pin |
|-------|-------|--|
| 0     | ×     | I/O port                                       |
| 1     | 0     | SOB1 output                                    |
| 1     | 1     | TXDA2 output                                   |

| PMC32 | PFCE32 | PFC32 | Specification of alternate function of P32 pin |
|-------|--------|-------|--|
| 0     | ×      | ×     | I/O port                                       |
| 1     | 0      | 0     | SIB1 input                                     |
| 1     | 0      | 1     | RXDA2 input                                    |
| 1     | 1      | 0     | CS1 <sup>Note</sup> output                     |
| 1     | 1      | 1     | Setting prohibited                             |

**Note**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**Remark** × = don't care

16.1.3 Mode switching between CSIB2 and UARTB

In the V850E/IF3 and V850E/IG3, CSIB2 and UARTB function alternately, and these functions cannot be used at the same time. To use CSIB2 and UARTB, the PMC3, PFC3, and PFCE3 registers must be set in advance.

**Caution** The operations related to transmission and reception of CSIB2 or UARTB are not guaranteed if the mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 16-3. Mode Switch Settings of CSIB2 and UARTB

After reset: 00H R/W Address: FFFFF446H

|      |       |       |       |       |       |       |       |       |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
|      | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| PMC3 | PMC37 | PMC36 | PMC35 | PMC34 | PMC33 | PMC32 | PMC31 | PMC30 |

After reset: 00H R/W Address: FFFFF466H

|      |       |       |       |       |       |       |       |       |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
|      | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| PFC3 | PFC37 | PFC36 | PFC35 | PFC34 | PFC33 | PFC32 | PFC31 | PFC30 |

After reset: 00H R/W Address: FFFFF706H

|       |        |        |        |        |   |        |        |        |
|-------|--------|--------|--------|--------|---|--------|--------|--------|
|       | 7      | 6      | 5      | 4      | 3 | 2      | 1      | 0      |
| PFCE3 | PFCE37 | PFCE36 | PFCE35 | PFCE34 | 0 | PFCE32 | PFCE31 | PFCE30 |

| PMC37 | PFCE37 | PFC37 | Specification of alternate function of P37 pin |
|-------|--------|-------|--|
| 0     | ×      | ×     | I/O port                                       |
| 1     | 0      | 0     | SCKB2 I/O                                      |
| 1     | 0      | 1     | INTP12 input                                   |
| 1     | 1      | 0     | ASTB <sup>Note</sup> output                    |
| 1     | 1      | 1     | Setting prohibited                             |

| PMC36 | PFCE36 | PFC36 | Specification of alternate function of P36 pin |
|-------|--------|-------|--|
| 0     | ×      | ×     | I/O port                                       |
| 1     | 0      | 0     | SOB2 output                                    |
| 1     | 0      | 1     | TXDB output                                    |
| 1     | 1      | 0     | Setting prohibited                             |
| 1     | 1      | 1     | Setting prohibited                             |

| PMC35 | PFCE35 | PFC35 | Specification of alternate function of P35 pin |
|-------|--------|-------|--|
| 0     | ×      | ×     | I/O port                                       |
| 1     | 0      | 0     | SIB2 input                                     |
| 1     | 0      | 1     | RXDB input                                     |
| 1     | 1      | 0     | Setting prohibited                             |
| 1     | 1      | 1     | Setting prohibited                             |

**Note**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**Remark** × = don't care

## 16.2 Features

- Transfer rate: 8 Mbps (using internal clock)
- Master mode and slave mode selectable
- Interrupt request signals: 3
  - Reception end interrupt request signal (INTCBnR): This signal is generated when reception is enabled and receive data is transferred from the shift register to the CSIBn receive data register (CBnRX) after completion of a serial transfer.
  - Transmission enable interrupt request signal (INTCBnT): This signal is generated when transmission is enabled in the continuous transmission or continuous transmission/reception mode and transmission data is transferred from the CSIBn transmit data register (CBnTX) to the shift register.
  - Reception error interrupt request signal (INTCBnRE): This signal is generated if an overrun error occurs (CBnSTR.CBnOVE bit = 1) when reception is enabled in the continuous transfer mode.
- Serial clock and data phase switchable
- 3-wire serial interface, 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
  - SCKBn: Serial clock I/O

Transmission mode, reception mode, and transmission/reception mode specifiable
- Double buffer for both transmission and reception
- Overrun error detection

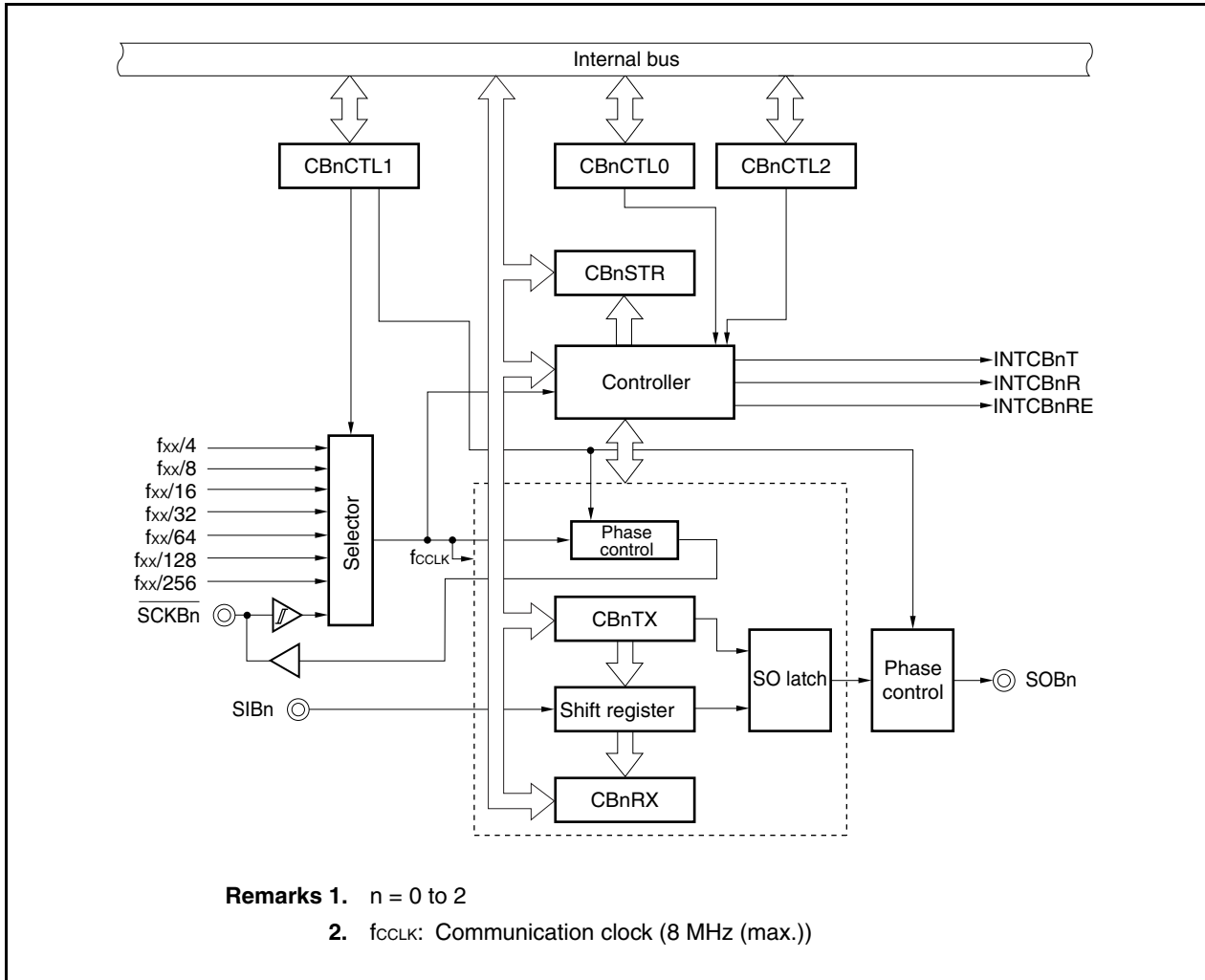
**Remark** n = 0 to 2

16.3 Configuration

The following shows the block diagram of CSIBn.

<R>

Figure 16-4. Block Diagram of CSIBn



CSIBn includes the following hardware.

Table 16-1. Configuration of CSIBn

| Item              | Configuration  |
|-------------------|--|
| Registers         | CSIBn receive data register (CBnRX)<br>CSIBn transmit data register (CBnTX)  |
| Control registers | CSIBn control register 0 (CBnCTL0)<br>CSIBn control register 1 (CBnCTL1)<br>CSIBn control register 2 (CBnCTL2)<br>CSIBn status register (CBnSTR) |



## 16.4 Control Registers

The following registers are used to control CSIBn.

- CSIBn control register 0 (CBnCTL0)
- CSIBn control register 1 (CBnCTL1)
- CSIBn control register 2 (CBnCTL2)
- CSIBn status register (CBnSTR)

### (1) CSIBn control register 0 (CBnCTL0)

CBnCTL0 is a register that controls the CSIBn serial transfer operation.

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

Reset sets this register to 01H.

(1/2)

After reset: 01H    R/W    Address: CB0CTL0 FFFFFFFD00H, CB1CTL0 FFFFFFFD10H,  
CB2CTL0 FFFFFFFD20H

|                         |        |                        |                        |                        |   |   |                        |        |
|-------------------------|--------|------------------------|------------------------|------------------------|---|---|------------------------|--------|
| CBnCTL0<br>(n = 0 to 2) | <7>    | <6>                    | <5>                    | <4>                    | 3 | 2 | 1                      | <0>    |
|                         | CBnPWR | CBnTXE <sup>Note</sup> | CBnRXE <sup>Note</sup> | CBnDIR <sup>Note</sup> | 0 | 0 | CBnTMS <sup>Note</sup> | CBnSCE |

|  |   |
|--|---|
| CBnPWR   | Specification of CSIBn operation disable/enable       |
| 0  | Disable CSIBn operation and reset the CBnSTR register |
| 1  | Enable CSIBn operation                                |
| • The CBnPWR bit controls the CSIBn operation and resets the internal circuit. |   |

|  |  |
|--|--|
| CBnTXE <sup>Note</sup>                                   | 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 <sup>Note</sup>  | Specification of receive operation disable/enable |
| 0   | Disable receive operation                         |
| 1   | Enable receive operation                          |
| • When the CBnRXE bit is cleared to 0, no reception end 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** Be sure to set bits 3 and 2 to “0”.

| CBnDIR <sup>Note 1</sup> | Specification of transfer direction mode (MSB/LSB) |
|--------------------------|--|
| 0                        | MSB first  |
| 1                        | LSB first  |

| CBnTMS <sup>Note 1</sup> | Transfer mode specification |
|--------------------------|-----------------------------|
| 0                        | Single transfer mode        |
| 1                        | Continuous transfer mode    |

- When using single transmission or transmission/reception mode with communication type 2 or 4 (CBnCTL1.CBnDAP bit = 1), write the transfer data to the CBnTX register after checking that the CBnSTR.CBnTSF bit is 0.
- When using DMA, use the continuous transfer mode.

| CBnSCE | Specification of start transfer disable/enable |
|--------|--|
| 0      | Communication start trigger invalid            |
| 1      | Communication start trigger valid              |

- In master mode  
This bit enables or disables the communication start trigger.
  - In single reception mode  
Set the CBnSCE bit to 0 before reading the receive data (CBnRX register)<sup>Note 2</sup>.
  - In continuous reception mode  
Set the CBnSCE bit to 0 one communication clock before reception of the last data is ended<sup>Note 3</sup>.
- In slave mode  
This bit enables or disables the communication start trigger.
  - In single reception mode or continuous reception mode  
Set the CBnSCE bit to 1<sup>Note 4</sup>.
- In single transmission or transmission/reception mode, or continuous transmission or transmission/reception mode  
The function of the CBnSCE bit is invalid. It is recommended to set this bit to 1.

- Notes**
1. These bits can only be rewritten when the CBnPWR bit = 0. However, the CBnPWR bit can be set to 1 at the same time as these bits are rewritten.
  2. If the CBnSCE bit is read while it is 1, the next communication operation is started.
  3. The CBnSCE bit is not set to 0 one communication clock before the end of the last data reception, the next communication operation is automatically started.  
To start communication operation again after reading the last data, set the CBnSCE bit to 1 and perform a dummy read of the CBnRX register.
  4. To start the reception, a dummy read is necessary.

**(a) How to use CBnSCE bit****(i) In single reception mode**

- <1> When the reception of the last data is ended with INTCBnR interrupt servicing, clear the CBnSCE bit to 0, and then read the CBnRX register.
- <2> When the reception is disabled after the reception of the last data has been ended, check that the CBnSTR.CBnTSF bit is 0, and then clear the CBnPWR and CBnRXE bits to 0. To continue reception, set the CBnSCE bit to 1 and start the next receive operation by performing a dummy read of the CBnRX register.

**(ii) In continuous reception mode**

- <1> Clear the CBnSCE bit to 0 during reception of the last data with INTCBnR interrupt servicing by the reception before the last reception, and then read the CBnRX register.
- <2> After receiving the INTCBnR signal of the last reception, read the last data from the CBnRX register.
- <3> When the reception is disabled after the reception of the last data has been ended, check that the CBnSTR.CBnTSF bit is 0, and then clear the CBnPWR and CBnRXE bits to 0. To continue reception, set the CBnSCE bit to 1 and start the next receive operation by performing a dummy read of the CBnRX register.

**Caution** In continuous reception mode, the serial clock is not stopped until the reception executed when the CBnSCE bit is cleared to 0 is ended after the reception is started by a dummy read.



**(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.

**Caution** The CBnCTL1 register can be rewritten only when the CBnCTL0.CBnPWR bit = 0.

After reset: 00H    R/W    Address: CB0CTL1 FFFFFFFD01H, CB1CTL1 FFFFFFFD11H,  
 CB2CTL1 FFFFFFFD21H

|         |  |   |   |   |        |        |         |         |         |
|---------|--|---|---|---|--------|--------|---------|---------|---------|
|         |  | 7 | 6 | 5 | 4      | 3      | 2       | 1       | 0       |
| CBnCTL1 |  | 0 | 0 | 0 | CBnCKP | CBnDAP | CBnCKS2 | CBnCKS1 | CBnCKS0 |

(n = 0 to 2)

|                      | CBnCKP | CBnDAP | Specification of data transmission/<br>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 <sub>CCLK</sub> ) | Mode        |
|---------|---------|---------|--|-------------|
| 0       | 0       | 0       | f <sub>xx</sub> /4                       | Master mode |
| 0       | 0       | 1       | f <sub>xx</sub> /8                       | Master mode |
| 0       | 1       | 0       | f <sub>xx</sub> /16                      | Master mode |
| 0       | 1       | 1       | f <sub>xx</sub> /32                      | Master mode |
| 1       | 0       | 0       | f <sub>xx</sub> /64                      | Master mode |
| 1       | 0       | 1       | f <sub>xx</sub> /128                     | Master mode |
| 1       | 1       | 0       | f <sub>xx</sub> /256                     | Master mode |
| 1       | 1       | 1       | External clock (SCKBn)                   | Slave mode  |

**Caution** Set f<sub>CCLK</sub> to 8 MHz or lower.

**(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 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,  
CB2CTL2 FFFFFFFD22H

|                         |   |   |   |   |        |        |        |        |
|-------------------------|---|---|---|---|--------|--------|--------|--------|
|                         | 7 | 6 | 5 | 4 | 3      | 2      | 1      | 0      |
| CBnCTL2<br>(n = 0 to 2) | 0 | 0 | 0 | 0 | CBnCL3 | CBnCL2 | CBnCL1 | CBnCL0 |

| CBnCL3 | CBnCL2 | CBnCL1 | CBnCL0 | Serial register bit length |
|--------|--------|--------|--------|----------------------------|
| 0      | 0      | 0      | 0      | 8 bits                     |
| 0      | 0      | 0      | 1      | 9 bits                     |
| 0      | 0      | 1      | 0      | 10 bits                    |
| 0      | 0      | 1      | 1      | 11 bits                    |
| 0      | 1      | 0      | 0      | 12 bits                    |
| 0      | 1      | 0      | 1      | 13 bits                    |
| 0      | 1      | 1      | 0      | 14 bits                    |
| 0      | 1      | 1      | 1      | 15 bits                    |
| 1      | ×      | ×      | ×      | 16 bits                    |

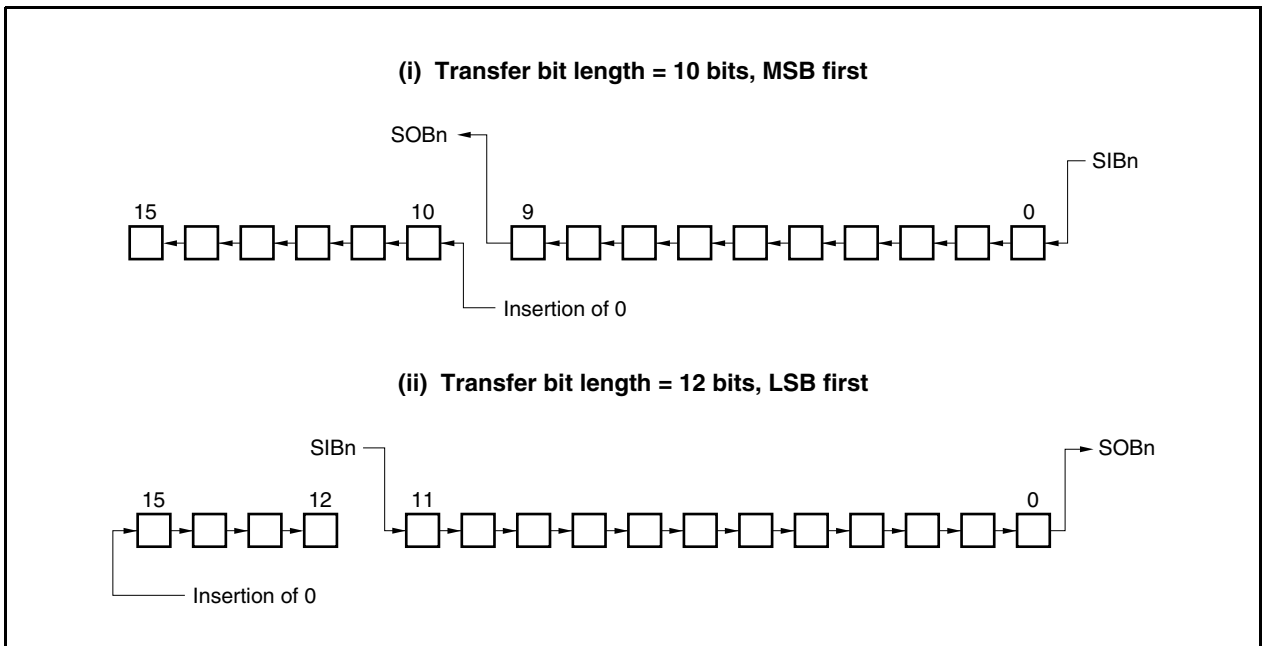
**Remark** If the number of transfer bits is other than 8 or 16, prepare and use data stuffed from the LSB of the CBnTX and CBnRX registers.

**(a) Transfer data length change function**

The CSIB<sub>n</sub> 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.

**Remark** n = 0 to 2



**(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, the CBnSTR register can be initialized by clearing (0) the CBnCTL0.CBnPWR bit.

After reset: 00H    R/W    Address: CB0STR FFFFFFFD03H, CB1STR FFFFFFFD13H,  
CB2STR FFFFFFFD23H

|                        |        |   |   |   |   |   |   |        |
|------------------------|--------|---|---|---|---|---|---|--------|
|                        | <7>    | 6 | 5 | 4 | 3 | 2 | 1 | <0>    |
| CBnSTR<br>(n = 0 to 2) | CBnTSF | 0 | 0 | 0 | 0 | 0 | 0 | CBnOVE |

|   |                           |
|---|---------------------------|
| CBnTSF  | Communication status flag |
| 0   | Communication stopped     |
| 1   | Communicating             |
| <ul style="list-style-type: none"> <li>• 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.</li> </ul> |                           |

|  |                    |
|--|--------------------|
| CBnOVE   | Overrun error flag |
| 0  | No overrun         |
| 1  | Overrun            |
| <ul style="list-style-type: none"> <li>• An overrun error occurs when the next reception starts without performing a CPU read of the value of the CBnRX register, upon end of the receive operation. The CBnOVE flag displays the overrun error occurrence status in this case.</li> <li>• The CBnOVE flag is cleared by writing 0 to it. It cannot be set even by writing 1 to it.</li> </ul> |                    |

**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.

For example, if the next data is written to the CBnTX register when DMA is started by generating the INTCBnR signal, the written data is not transferred because the CBnTSF bit is set to 1.

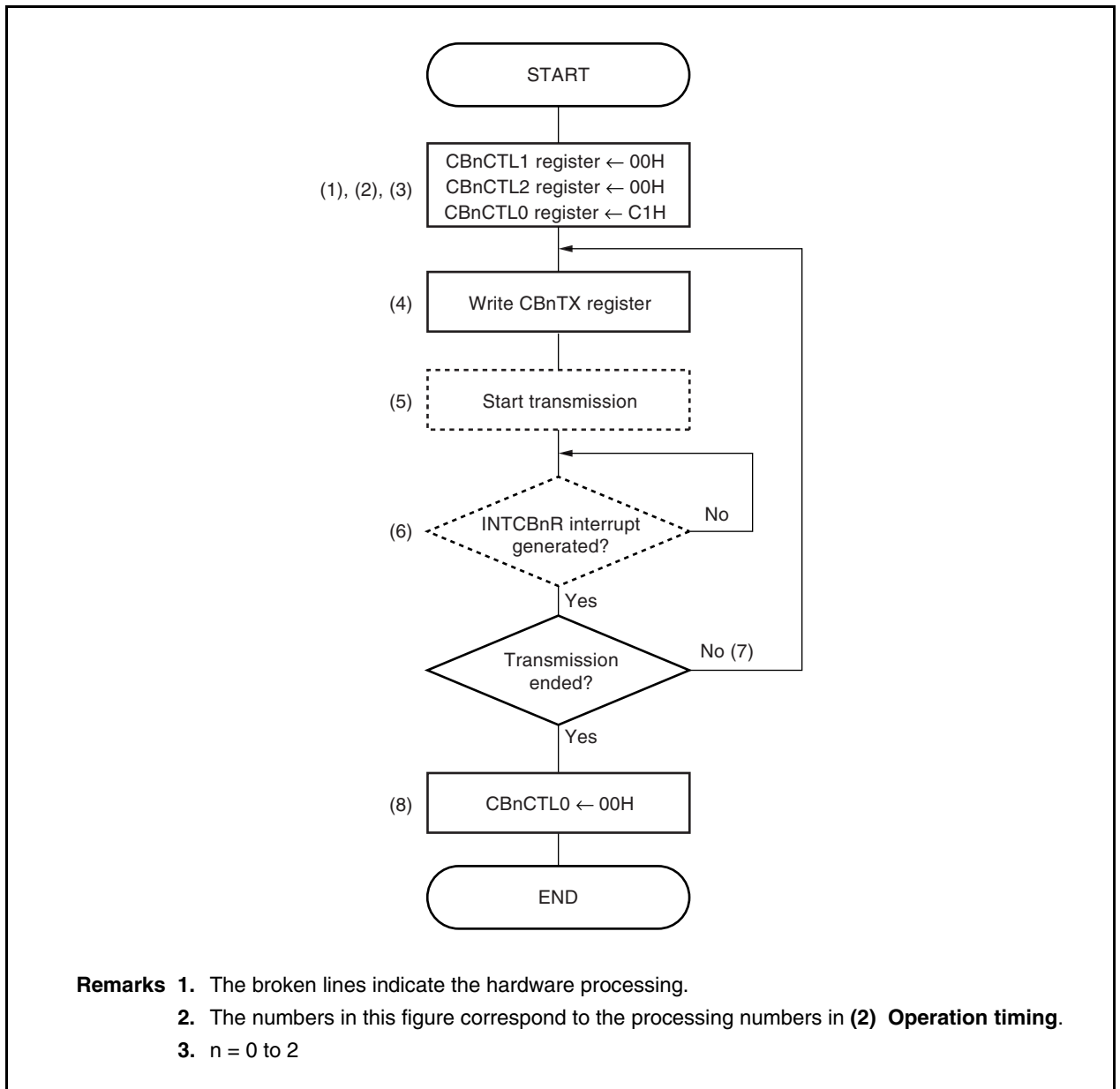
Use the continuous transfer mode, not the single transfer mode, for such applications.

## 16.5 Operation

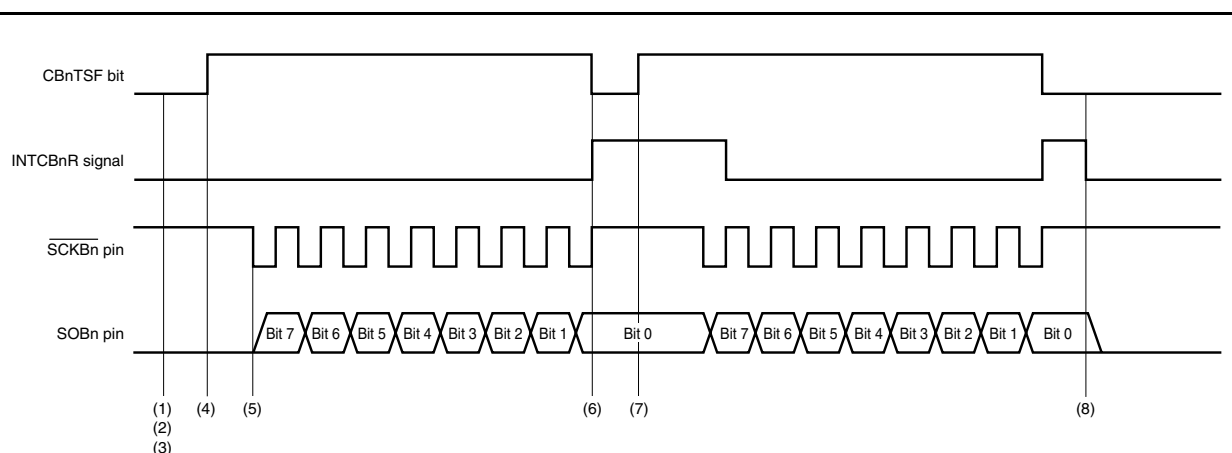
### 16.5.1 Single transfer mode (master mode, transmission mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock ( $f_{CCLK} = f_{xx}/4$ ) (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



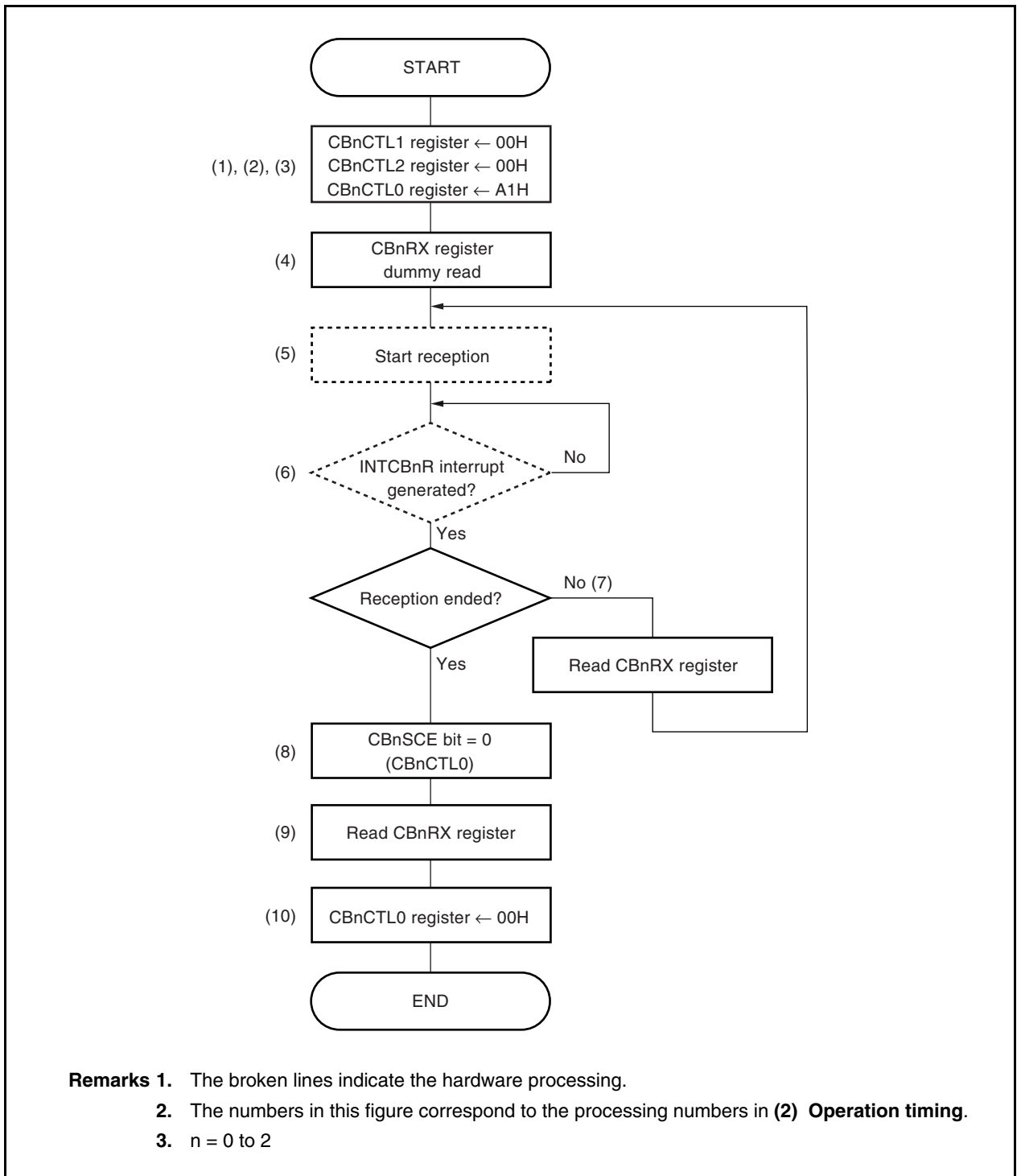
- (1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock ( $f_{\text{CLK}}$ ) =  $f_{\text{xx}}/4$ , and master mode.
- (2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.
- (3) Write C1H to the CBnCTL0 register, and select the transmission mode and MSB first at the same time as enabling the operation of the communication clock ( $f_{\text{CLK}}$ ).
- (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  $\overline{\text{SCKn}}$  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 end 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 to 2

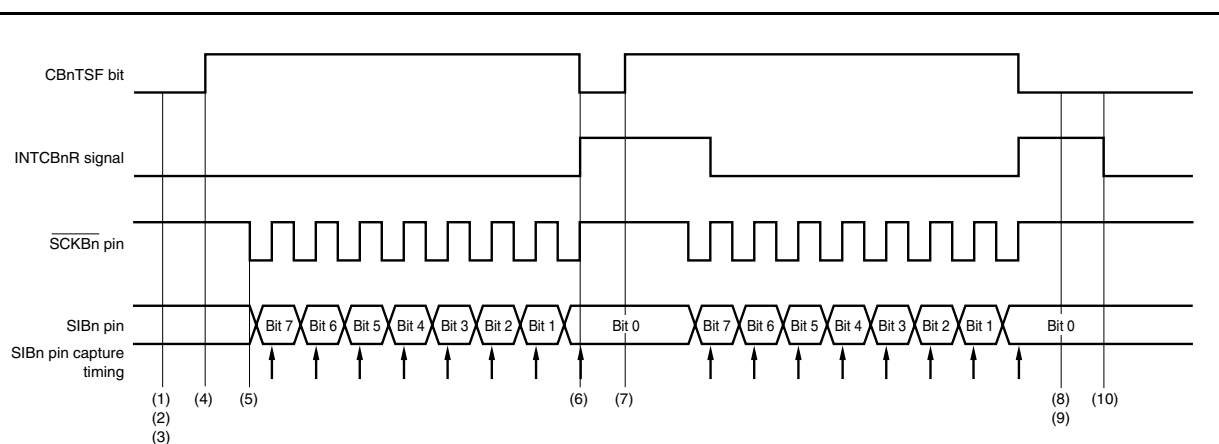
**16.5.2 Single transfer mode (master mode, reception mode)**

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock ( $f_{CCLK}$ ) =  $f_{XX}/4$  (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 to 2

**(2) Operation timing**

- (1) Write 00H to the CbNCTL1 register, and select communication type 1, communication clock ( $f_{CCLK}$ ) =  $f_{xx}/4$ , 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.CbNtSf 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  $\overline{SCKbN}$  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 end interrupt request signal (INTCbNR) at the last edge of the serial clock, and clear the CbNtSf 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.

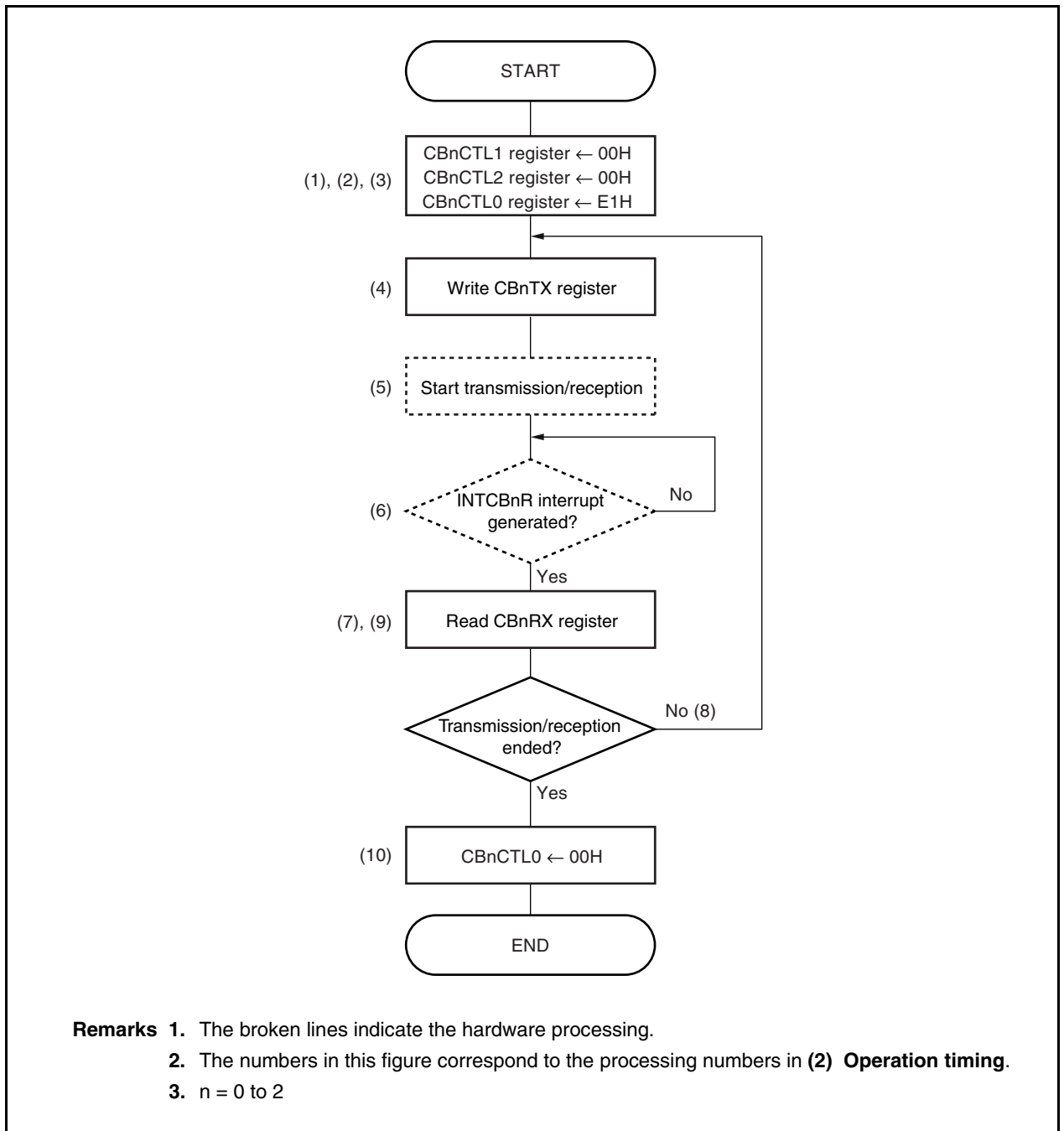
**Remark** n = 0 to 2



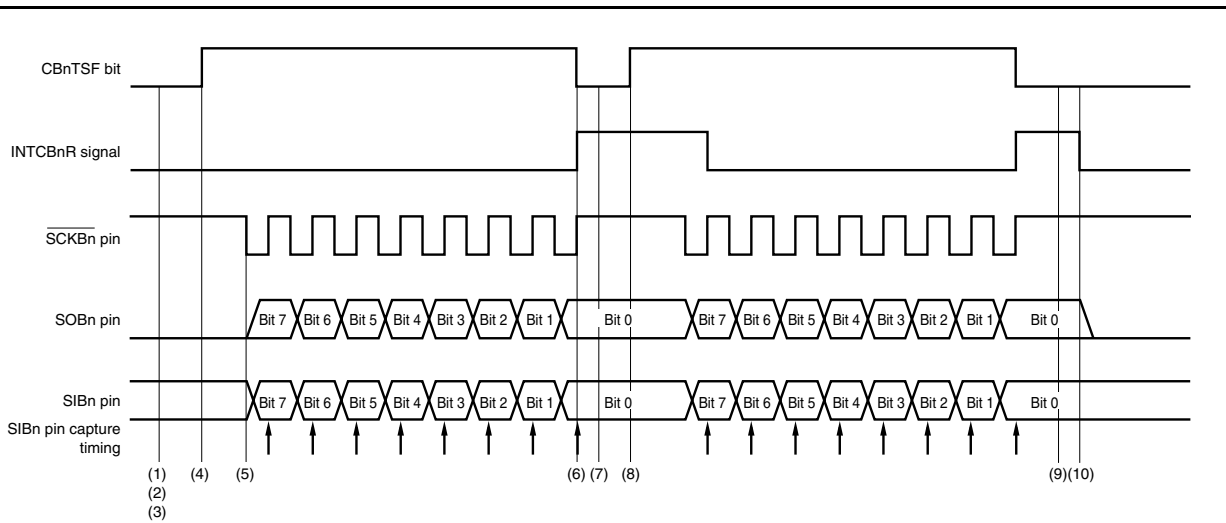
**16.5.3 Single transfer mode (master mode, transmission/reception mode)**

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock ( $f_{CCLK}$ ) =  $f_{XX}/4$  (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



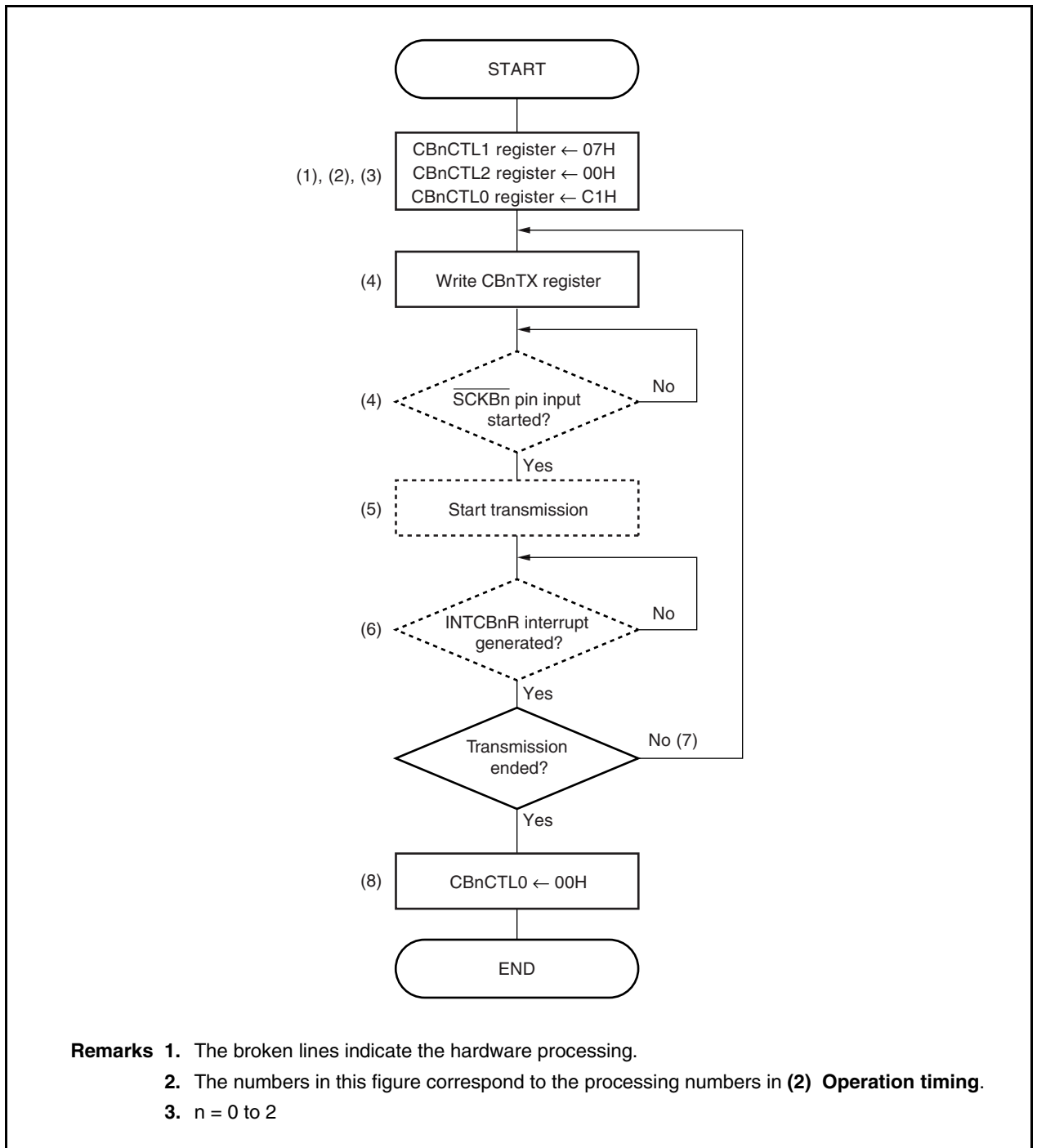
- (1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock ( $f_{CCLK}$ ) =  $f_x/4$ , and master mode.
- (2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.
- (3) Write E1H to the CBnCTL0 register, and select the transmission/reception mode and MSB first at the same time as enabling the operation of the communication clock ( $f_{CCLK}$ ).
- (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 end 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 to 2

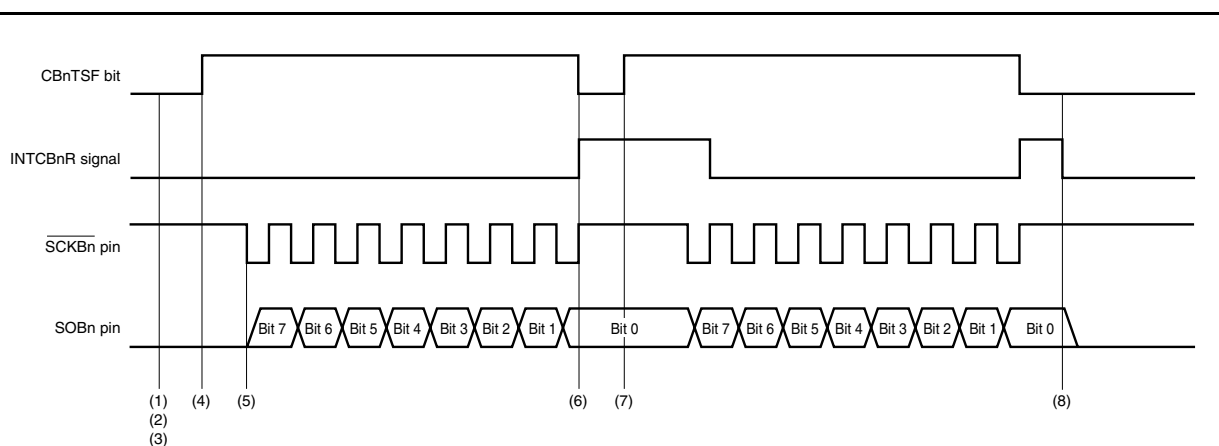
**16.5.4 Single transfer mode (slave mode, transmission mode)**

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f<sub>CCLK</sub>) = external clock ( $\overline{SCKBn}$ ) (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 to 2

**(2) Operation timing**

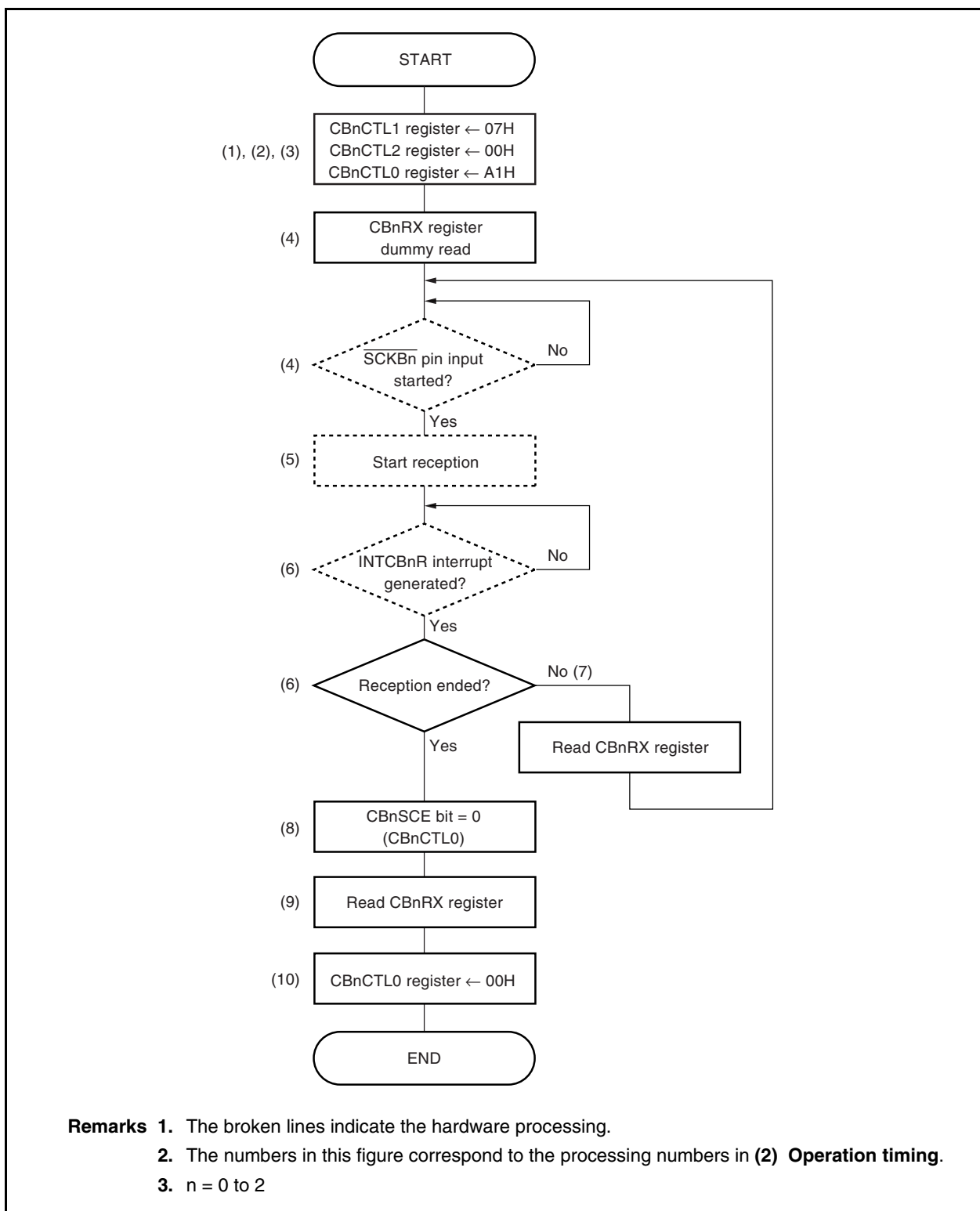
- (1) Write 07H to the CBNCTL1 register, and select communication type 1, communication clock ( $f_{\text{CLK}}$ ) = external clock ( $\overline{\text{SCKBn}}$ ), and slave mode.
- (2) Write 00H to the CBNCTL2 register, and set the transfer data length to 8 bits.
- (3) Write C1H to the CBNCTL0 register, and select the transmission mode and MSB first at the same time as enabling the operation of the communication clock ( $f_{\text{CLK}}$ ).
- (4) The CBNSTR.CBN\_TSF bit is set to 1 by writing the transmit data to the CBN\_TX 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 transmission of the transfer data length set with the CBNCTL2 register is completed, stop the serial clock input and transmit data output, generate the reception end interrupt request signal (INTCBnR) at the last edge of the serial clock, and clear the CBN\_TSF bit to 0.
- (7) To continue transmission, write the transmit data to the CBN\_TX register again after the INTCBnR signal is generated, and wait for a serial clock input.
- (8) To end transmission, write the CBNCTL0.CBN\_PWR bit = 0 and the CBNCTL0.CBN\_TXE bit = 0.

**Remark** n = 0 to 2

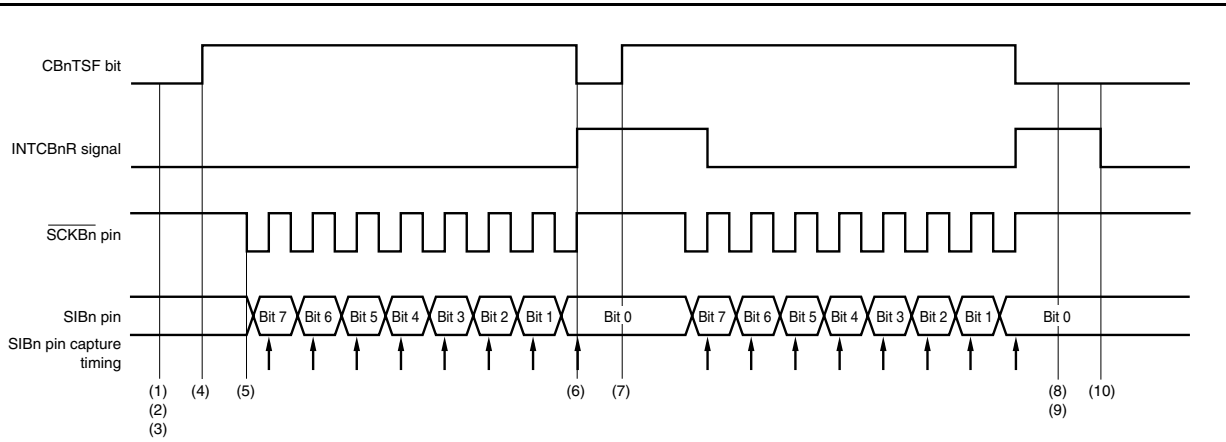
**16.5.5 Single transfer mode (slave mode, reception mode)**

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f<sub>CCLK</sub>) = external clock ( $\overline{SCKBn}$ ) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

**(1) Operation flow**



(2) Operation timing



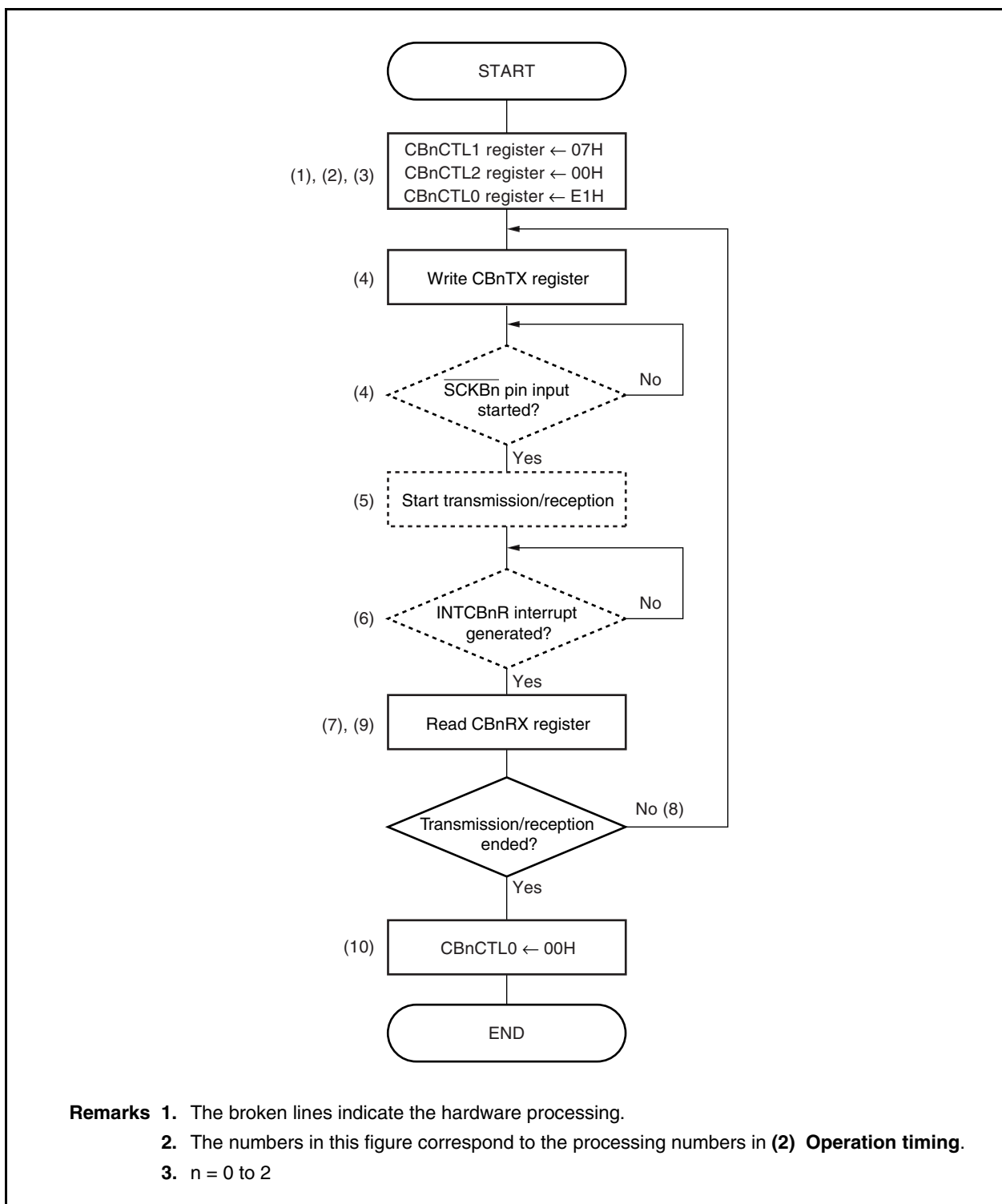
- (1) Write 07H to the CBnCTL1 register, and select communication type 1, communication clock ( $f_{CCLK}$ ) = external clock ( $\overline{SCKBn}$ ), and slave mode.
- (2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.
- (3) Write A1H to the CBnCTL0 register, and select the reception mode and MSB first at the same time as enabling the operation of the communication clock ( $f_{CCLK}$ ).
- (4) The CBnSTR.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 of the transfer data length set with the CBnCTL2 register is completed, stop the serial clock input and data capturing, generate the reception end interrupt request signal (INTCBnR) at the last edge of the serial clock, and clear the CBnTSF bit to 0.
- (7) To continue reception, read the CBnRX register with the CBnCTL0.CBnSCE bit = 1 remained after the INTCBnR signal is generated, and wait for a serial clock input.
- (8) To end 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 to 2

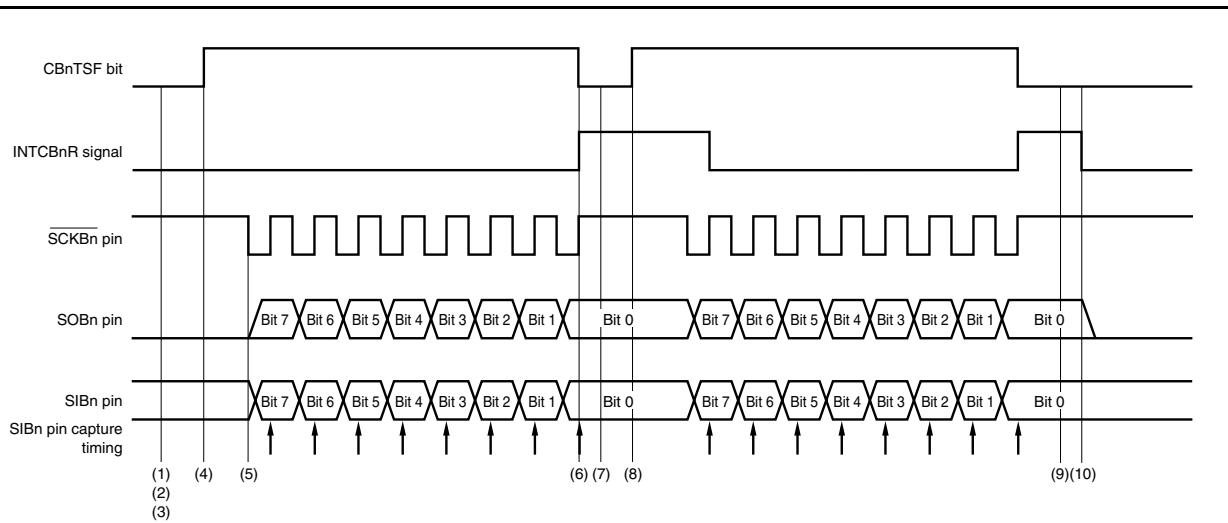
**16.5.6 Single transfer mode (slave mode, transmission/reception mode)**

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f<sub>CLK</sub>) = external clock ( $\overline{SCKBn}$ ) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

**(1) Operation flow**



(2) Operation timing



- (1) Write 07H to the CBnCTL1 register, and select communication type 1, communication clock ( $f_{\text{CLK}}$ ) = 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 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 ( $f_{\text{CLK}}$ ).
- (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 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 input, transmit data output, and data capturing, generate the reception end 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, and wait for a serial clock input.
- (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.

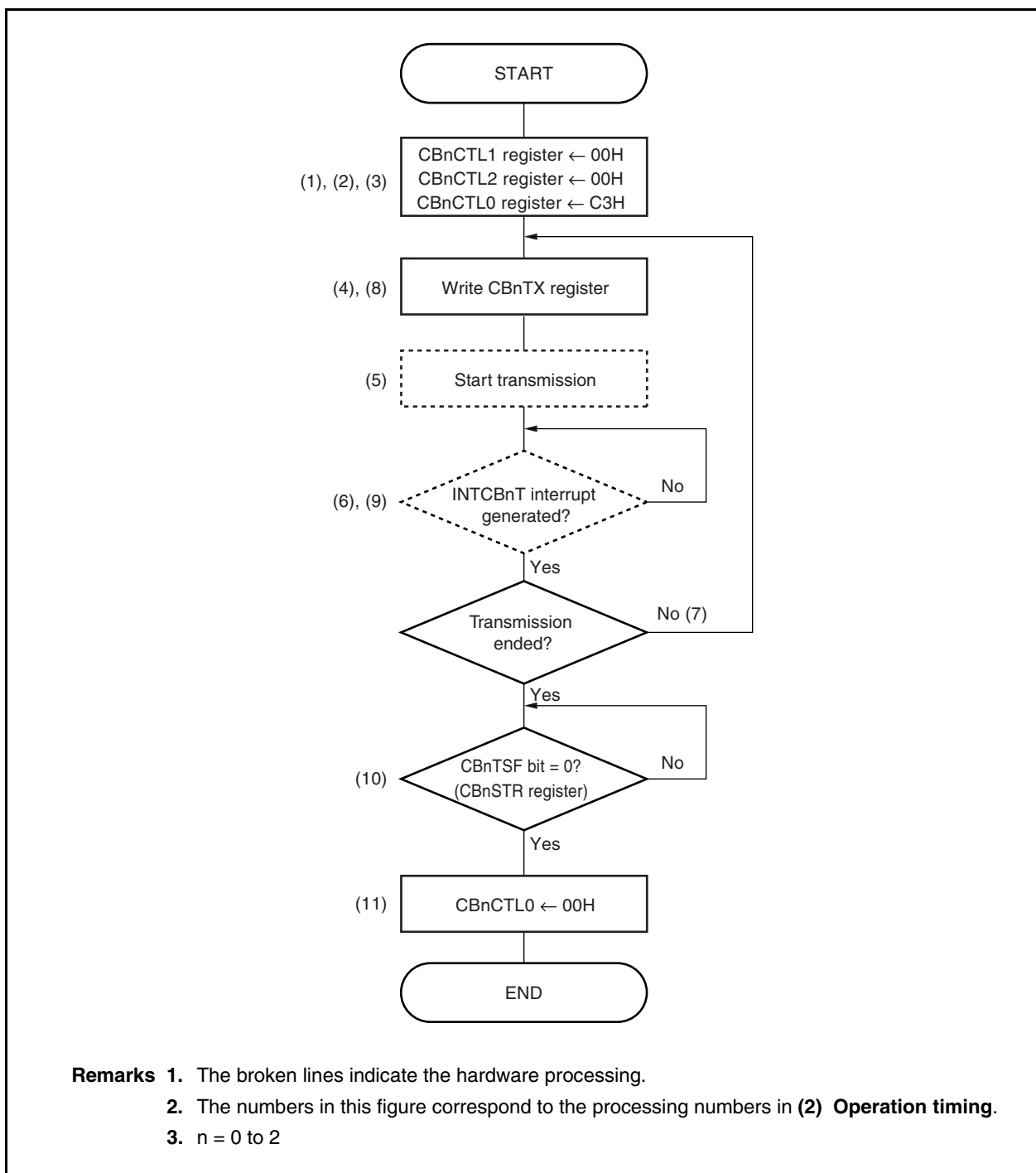
**Remark** n = 0 to 2



**16.5.7 Continuous transfer mode (master mode, transmission mode)**

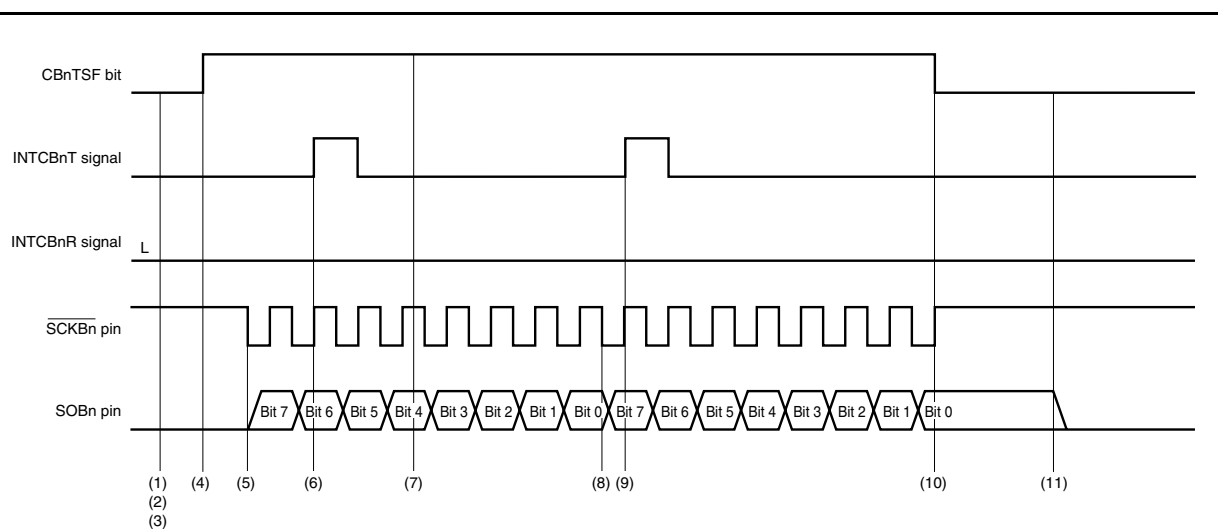
MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock ( $f_{CCLK} = f_{xx}/4$ ) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

**(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.  $n = 0$  to 2

## (2) Operation timing



- (1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock ( $f_{\text{CLK}}$ ) =  $f_{\text{xx}}/4$ , 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 ( $f_{\text{CLK}}$ ).
- (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  $\overline{\text{SCKBn}}$  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 ended 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 new transmit data is written to the CBnTX register before communication end, the next communication is started following communication end.
- (9) The transfer of the transmit data from the CBnTX register to the shift register is ended and the INTCBnT signal is generated. To end continuous transmission at 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 end, stop the serial clock output to the  $\overline{\text{SCKBn}}$  pin after transfer end, and clear the CBnTSF bit to 0.
- (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.

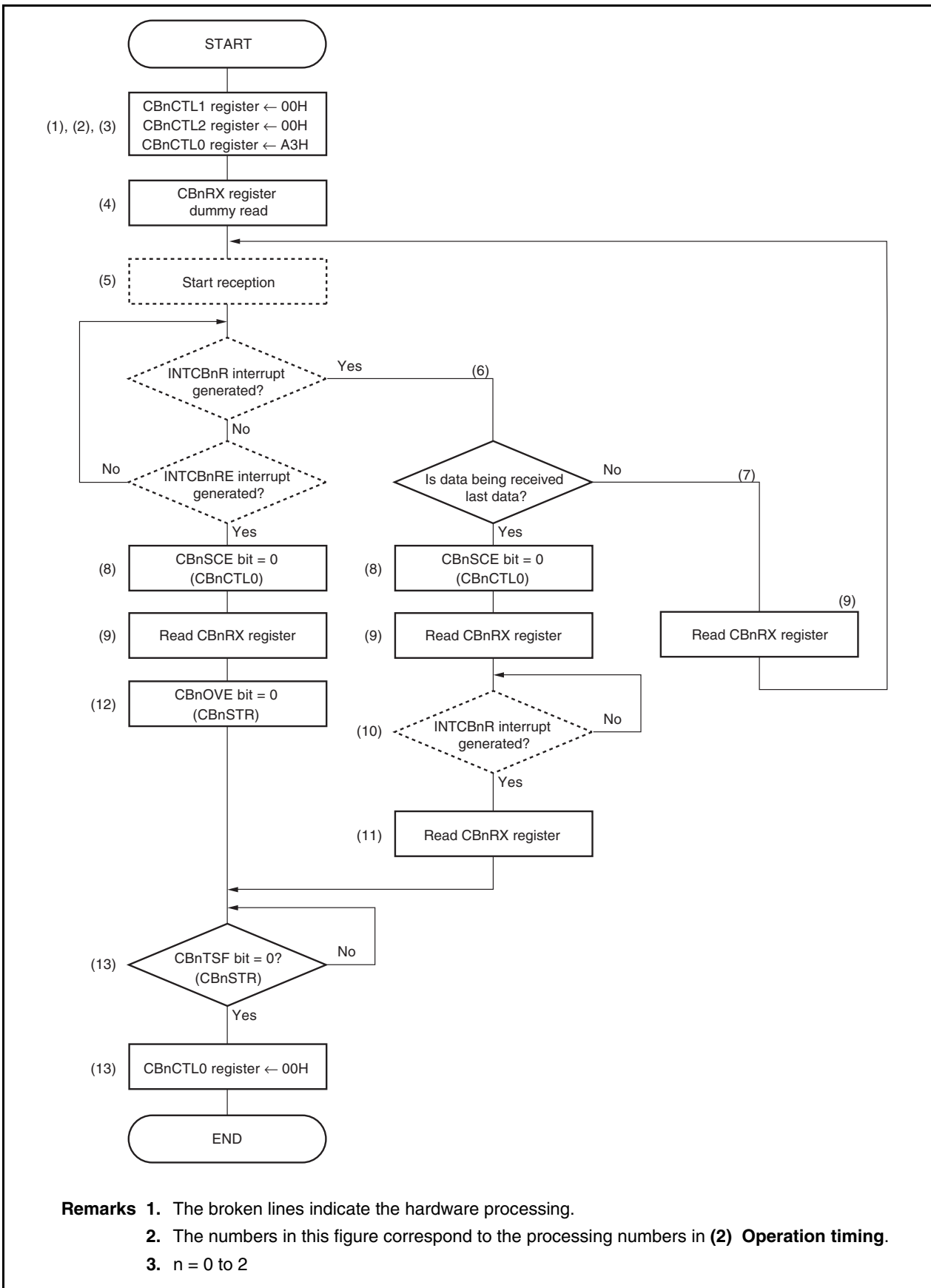
**Caution** In continuous transmission mode, the reception end interrupt request signal (INTCBnR) is not generated.

**Remark**  $n = 0$  to 2

**16.5.8 Continuous transfer mode (master mode, reception mode)**

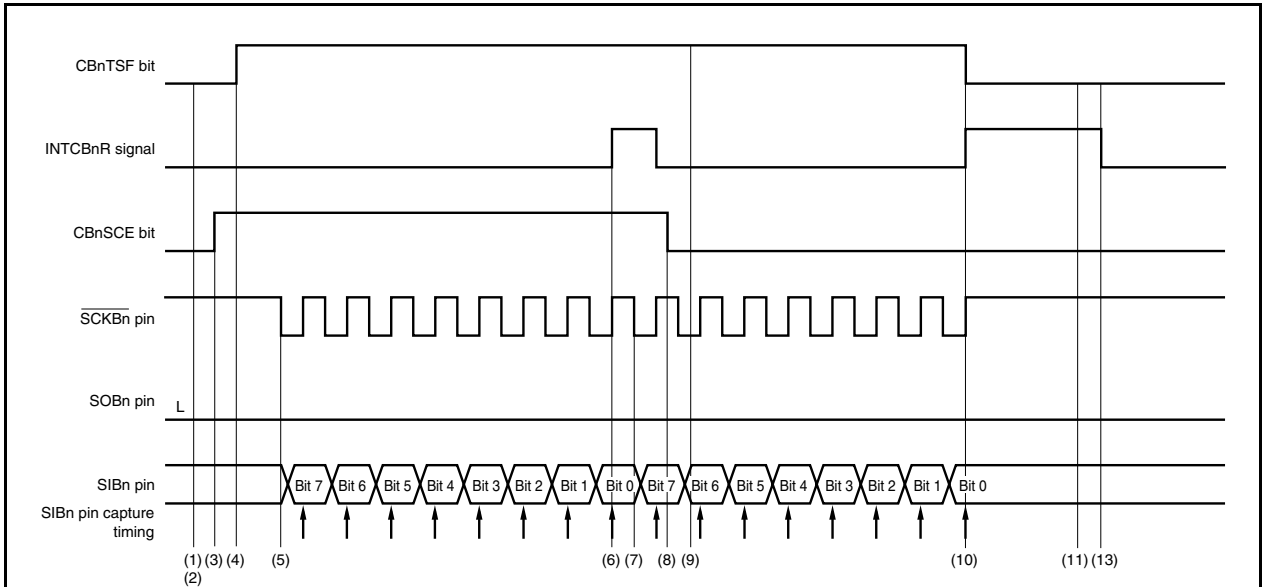
MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock ( $f_{\text{CLK}}$ ) =  $f_{\text{X}}/4$  (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(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 to 2

(2) Operation timing



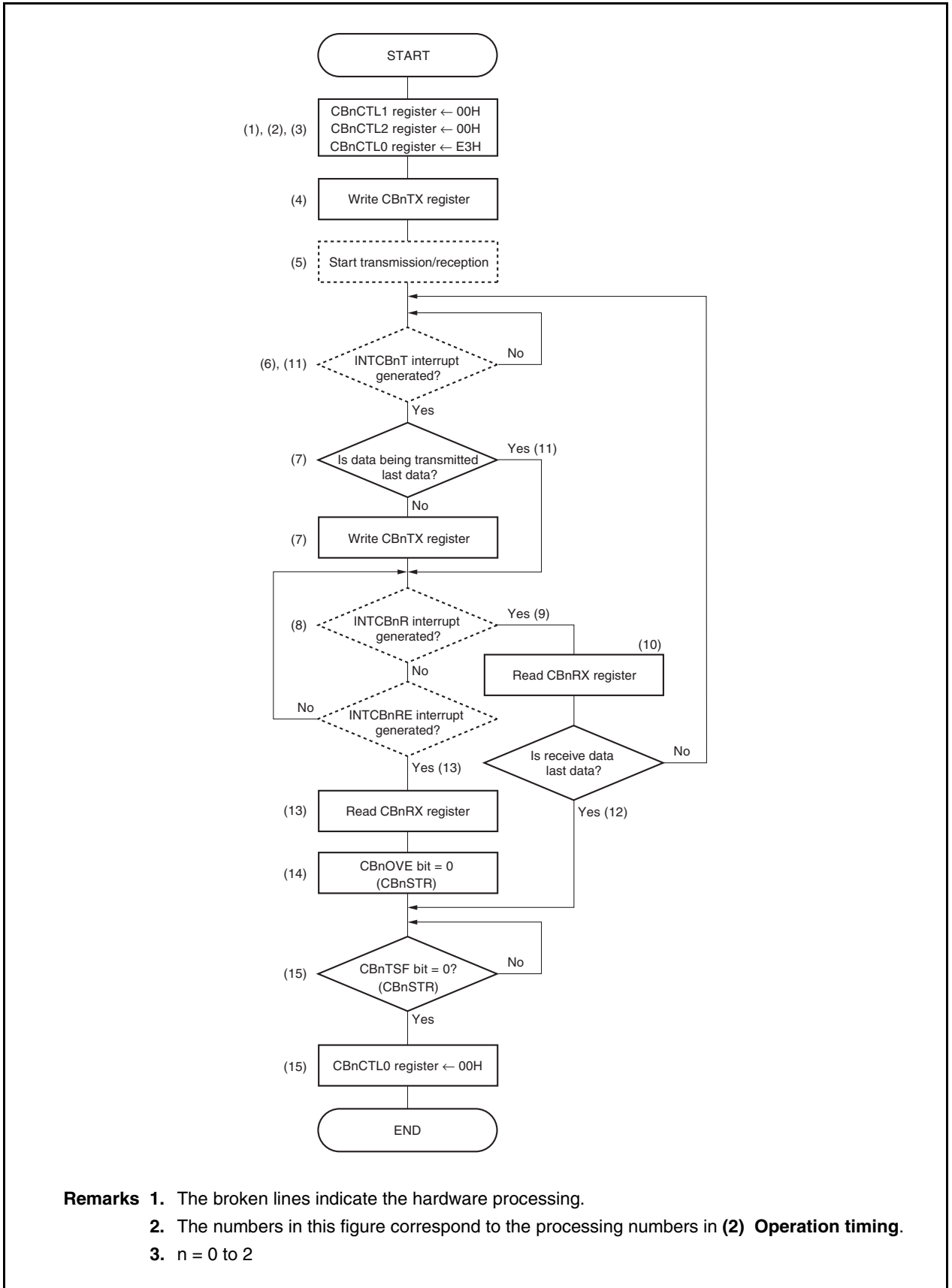
- (1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock ( $f_{CLK}$ ) =  $f_{xx}/4$ , and master mode.
- (2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.
- (3) Write A3H to the CBnCTL0 register, and select the reception mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock ( $f_{CLK}$ ).
- (4) The CBnSTR.CBnTSF 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  $\overline{SCKBn}$  pin, and capture the receive data of the SIBn pin in synchronization with the serial clock.
- (6) When reception is ended, the reception end interrupt request signal (INTCBnR) is generated, and reading of the CBnRX register is enabled.
- (7) When the CBnCTL0.CBnSCE bit = 1 upon communication end, the next communication is started following communication end.
- (8) To end continuous reception at the current reception, write the CBnSCE bit = 0.
- (9) Read the CBnRX register.
- (10) When reception is ended, the INTCBnR signal is generated, and reading of the CBnRX register is enabled. When the CBnSCE bit = 0 is set before communication end, stop the serial clock output to the  $\overline{SCKBn}$  pin, and 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 to 2

**16.5.9 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 ( $f_{\text{CLK}}$ ) =  $f_{\text{xx}}/4$  (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

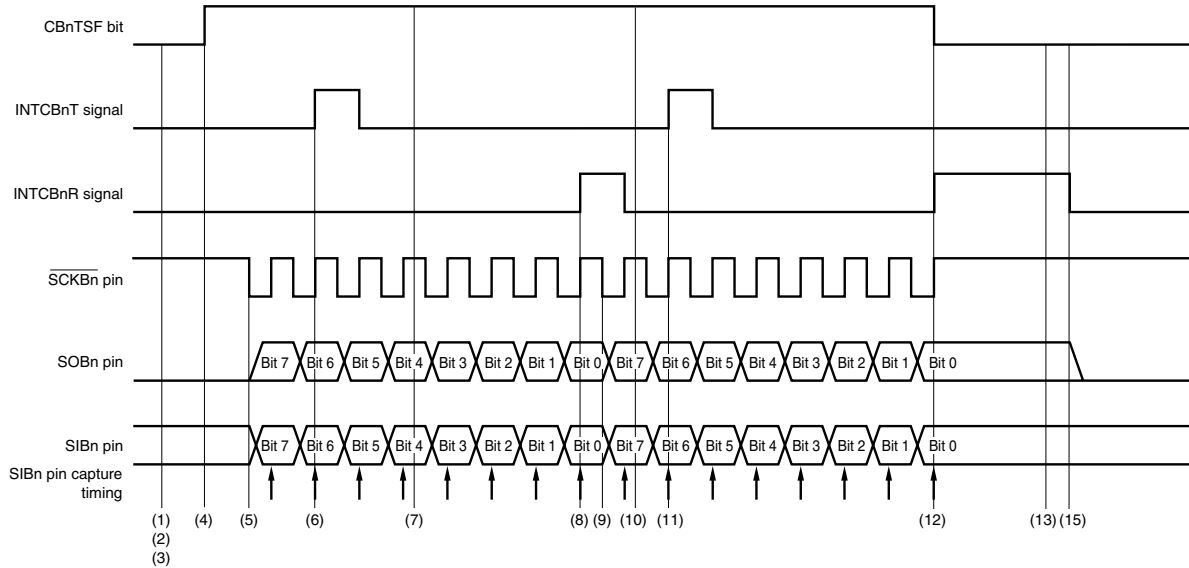
(1) Operation flow



- 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 to 2

(2) Operation timing

(1/2)



- (1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock ( $f_{CLK}$ ) =  $f_{xx}/4$ , 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.CBnTStF 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  $\overline{SCKBn}$  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 ended 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 ended, the reception end 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 end, the next communication is started following communication end.
- (10) Read the CBnRX register.

**Remark** n = 0 to 2



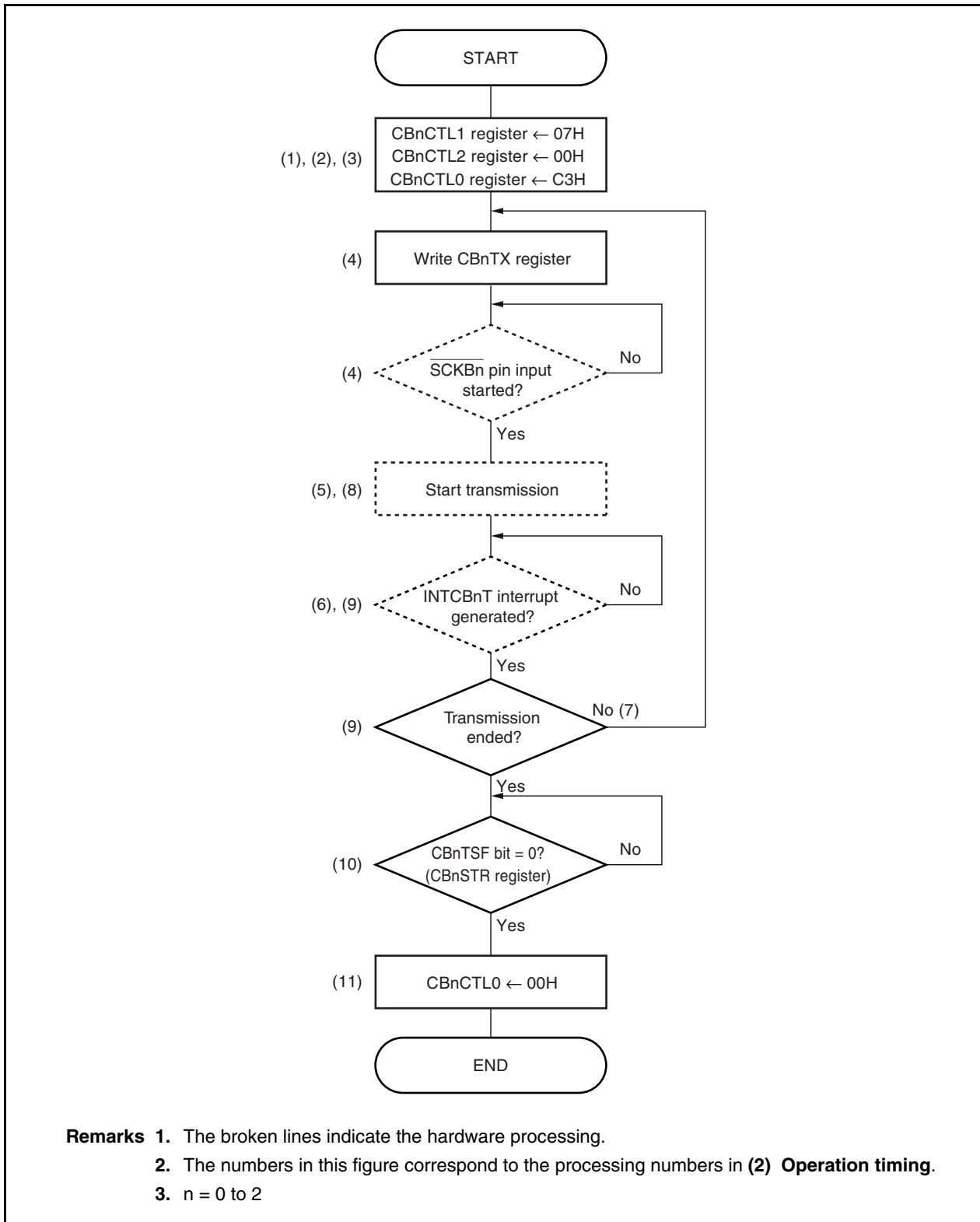
- (11) The transfer of the transmit data from the CBnTX register to the shift register is ended and the INTCBnT signal is generated. To end continuous transmission/reception at the current transmission/reception, do not write to the CBnTX register.
- (12) When the next transmit data is not written to the CBnTX register before transfer end, stop the serial clock output to the  $\overline{\text{SCKBn}}$  pin after transfer end, and clear the CBnTSF bit to 0.
- (13) When the reception error interrupt request signal (INTCBnRE) 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 to 2

**16.5.10 Continuous transfer mode (slave mode, transmission mode)**

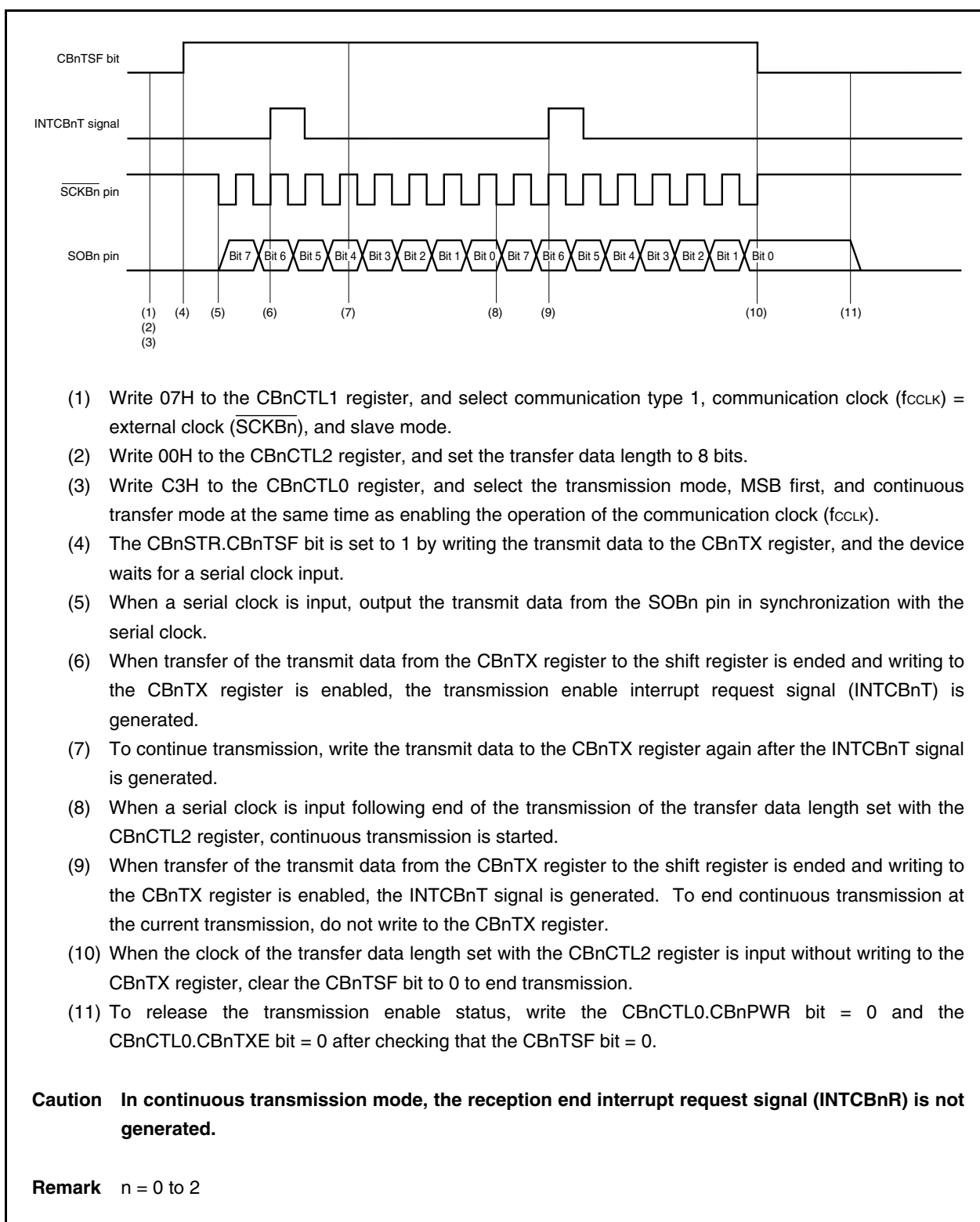
MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f<sub>CLK</sub>) = external clock (SCKBn) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

**(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 to 2

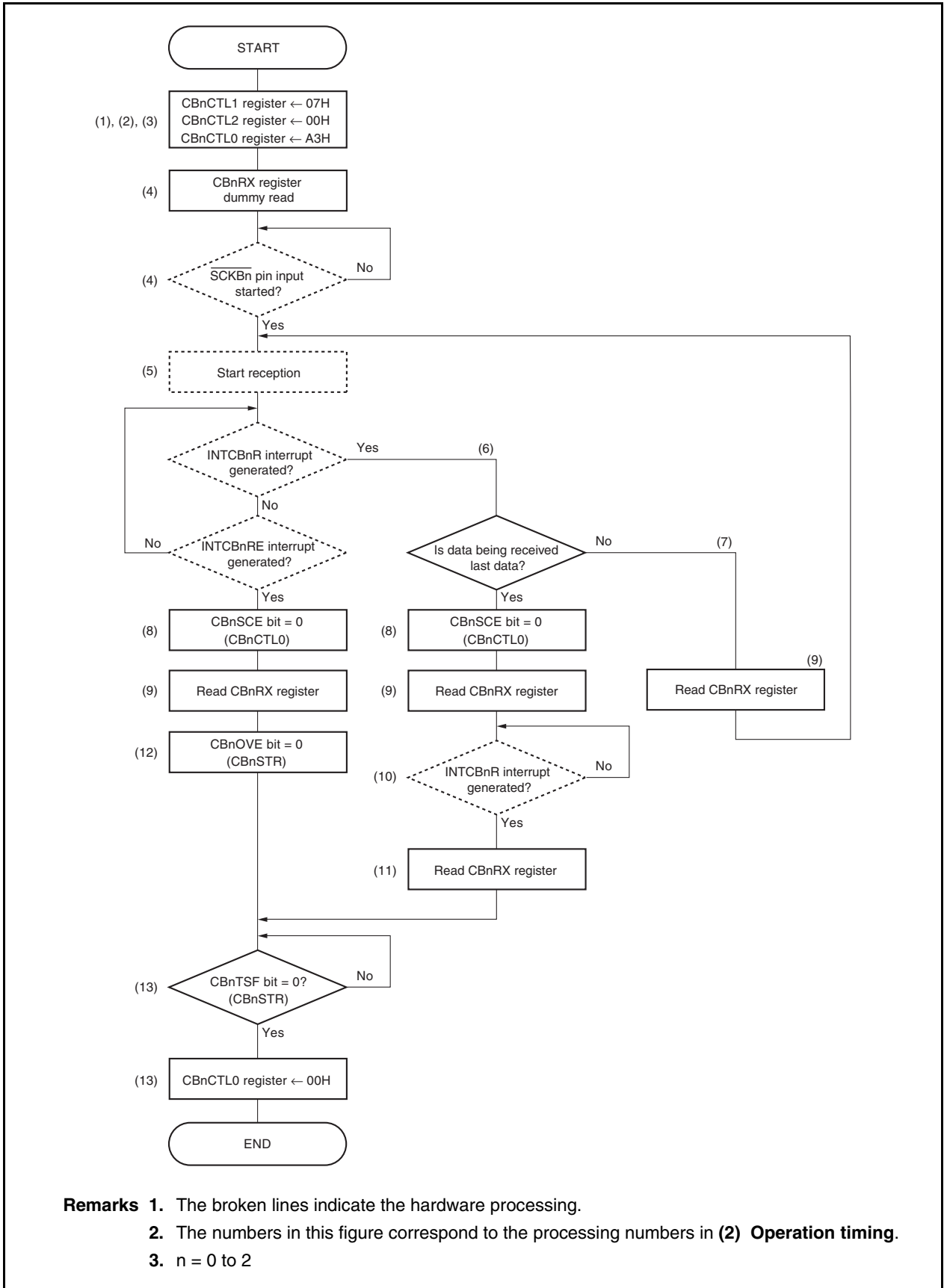
## (2) Operation timing



**16.5.11 Continuous transfer mode (slave mode, reception mode)**

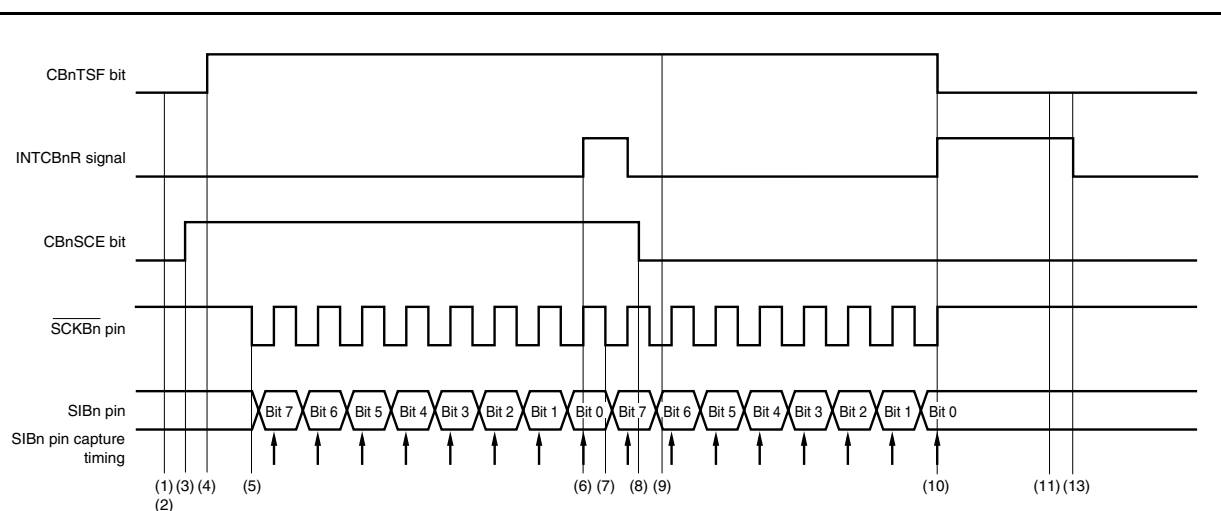
MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock ( $f_{\text{CCLK}}$ ) = external clock ( $\overline{\text{SCKBn}}$ ) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 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 to 2

## (2) Operation timing



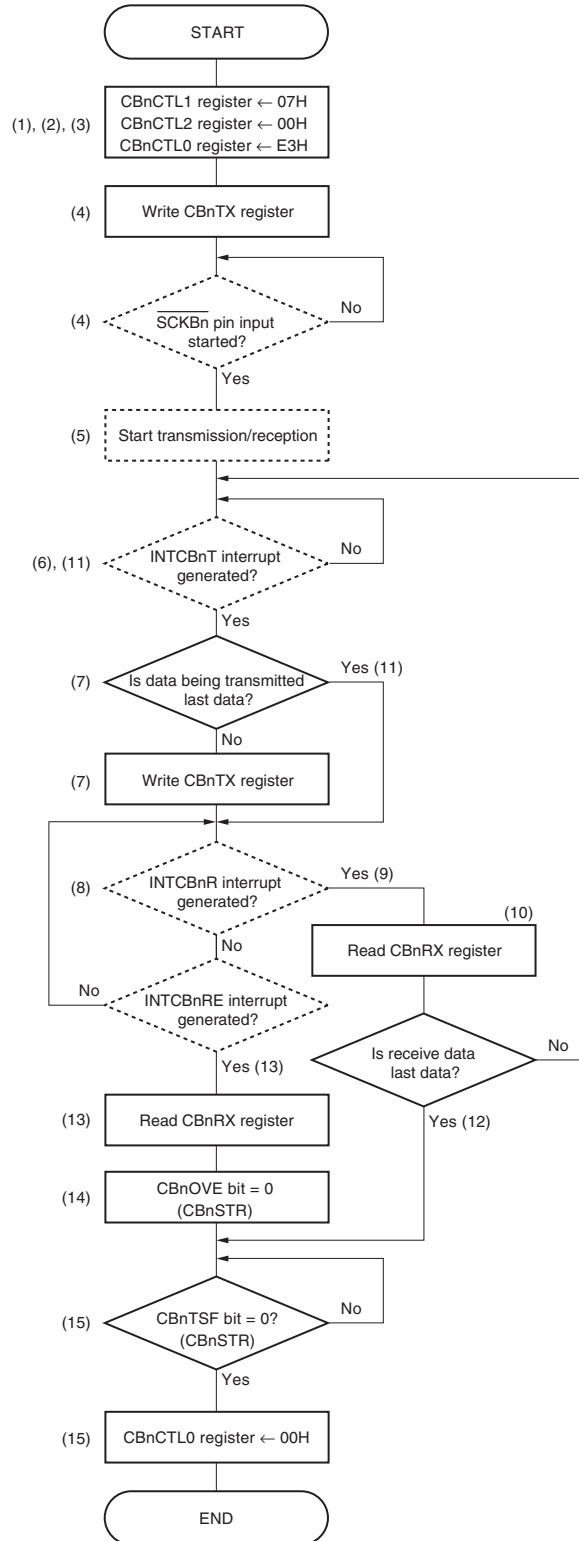
- (1) Write 07H to the CBnCTL1 register, and select communication type 1, communication clock ( $f_{\text{CLK}}$ ) = external clock ( $\overline{\text{SCKBn}}$ ), and slave mode.
- (2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.
- (3) Write A3H to the CBnCTL0 register, and select the reception mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock ( $f_{\text{CLK}}$ ).
- (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 ended, the reception end 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 at the current reception, write the CBnSCE bit = 0.
- (9) Read the CBnRX register.
- (10) When reception is ended, the INTCBnR signal is generated, and reading of the CBnRX register is enabled. When the CBnSCE bit = 0 is set before communication end, 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 to 2

**16.5.12 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 ( $f_{\text{CLK}}$ ) = 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

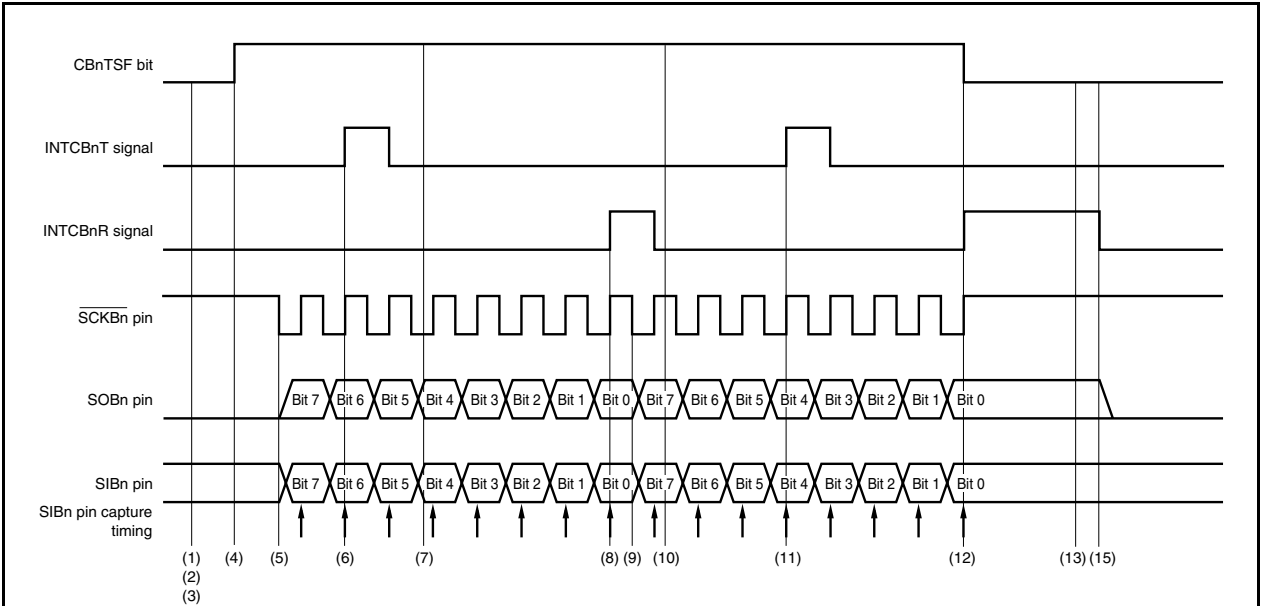


- Remarks**
1. The broken lines indicate the hardware processing.
  2. The numbers in this figure correspond to the processing numbers in (2) Operation timing.
  3. n = 0 to 2



(2) Operation timing

(1/2)



- (1) Write 07H to the CBnCTL1 register, and select communication type 1, communication clock (f<sub>CLK</sub>) = external clock (SCKBn), and slave 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<sub>CLK</sub>).
- (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 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 ended 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 reception of the transfer data length set with the CBnCTL2 register is completed, the reception end interrupt request signal (INTCBnR) is generated, and reading of the CBnRX register is enabled.
- (9) When a serial clock is input continuously, continuous transmission/reception is started.
- (10) Read the CBnRX register.
- (11) When transfer of the transmit data from the CBnTX register to the shift register is ended and writing to the CBnTX register is enabled, the INTCBnT signal is generated. To end continuous transmission/reception at the current transmission/reception, do not write to the CBnTX register.

**Remark** n = 0 to 2

- (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 reception error interrupt request signal (INTCBnRE) 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 to 2

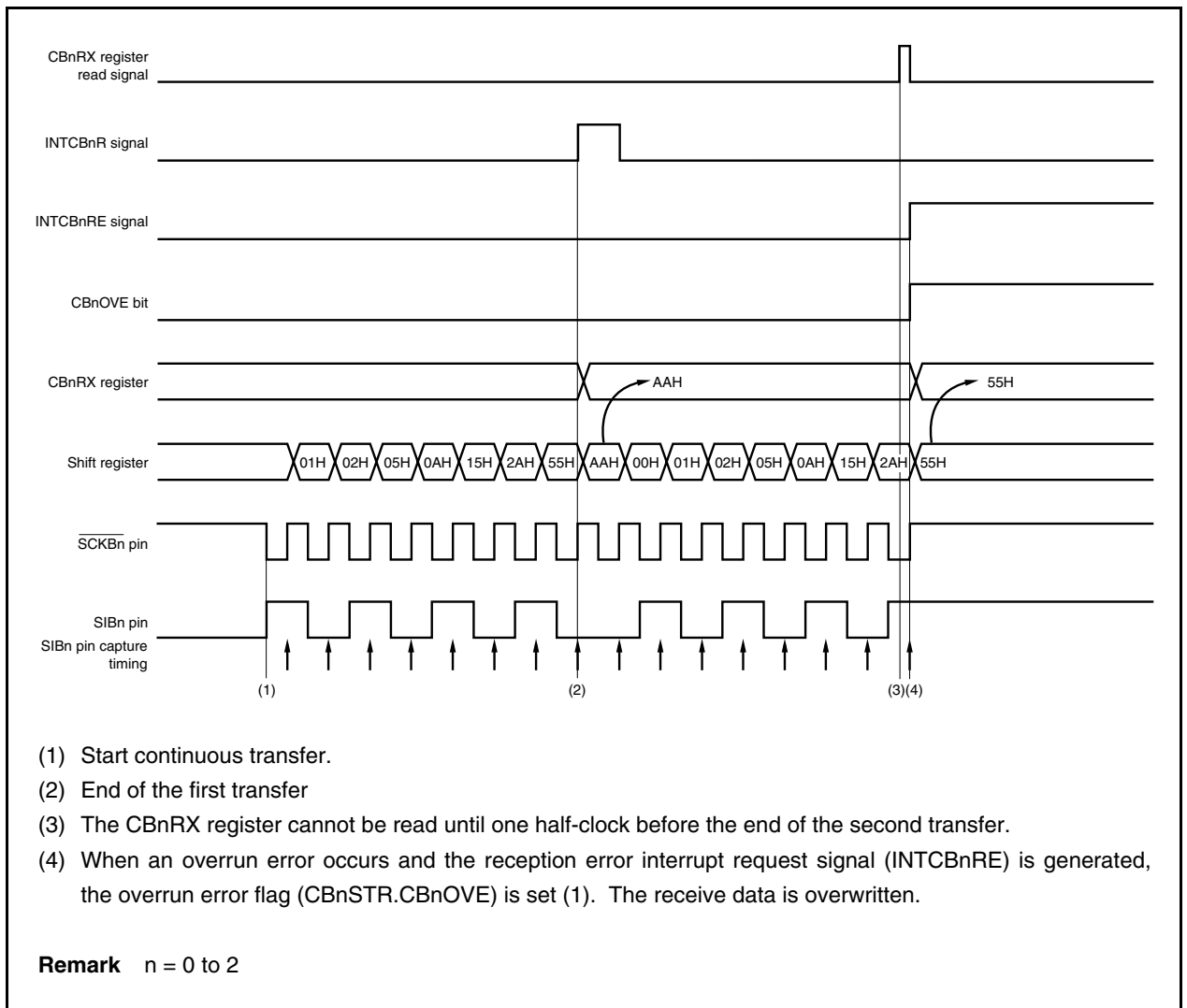
**16.5.13 Reception error**

When transfer is performed with reception enabled (CBnCTL0.CBnRXE bit = 1) in the continuous transfer mode, the reception error interrupt request signal (INTCBnRE) is generated when the next receive operation is ended before the CBnRX register is read after the reception end interrupt request signal (INTCBnR) 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 INTCBnRE signal is generated again upon the next reception end if the CBnRX register is not read.

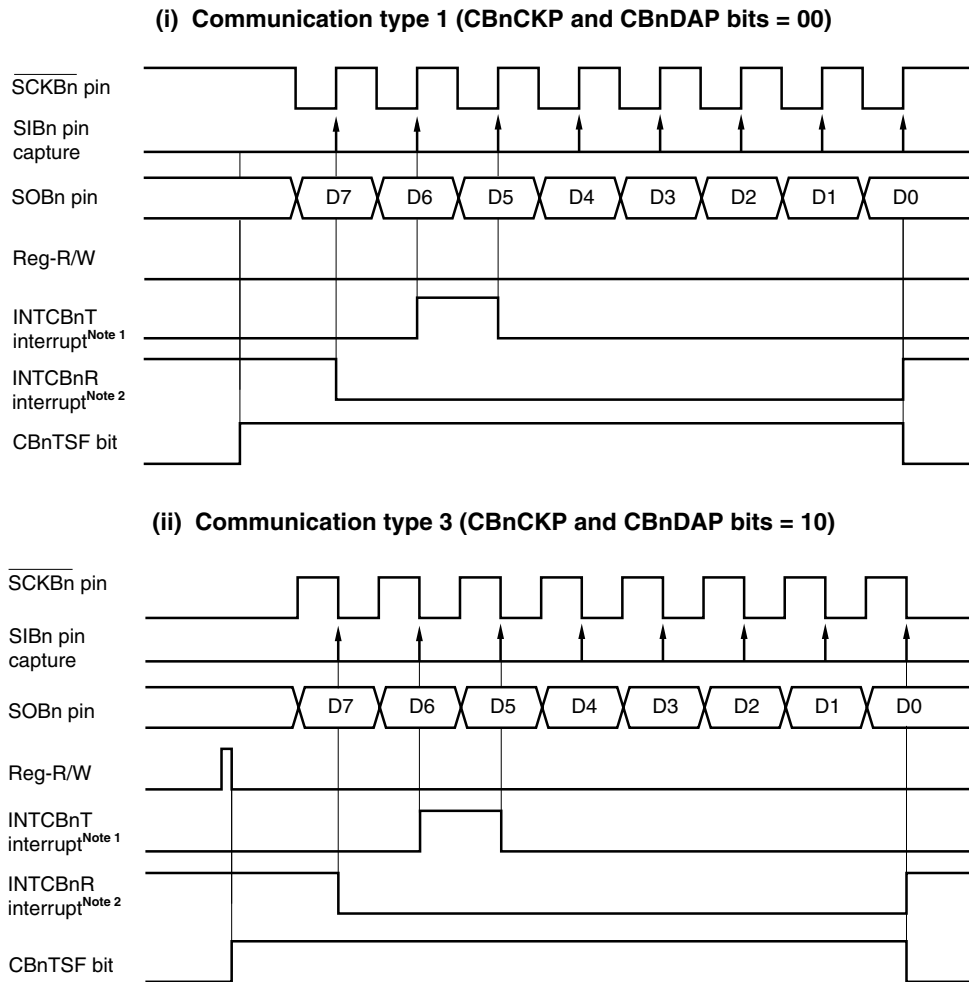
To avoid an overrun error, end reading the CBnRX register until one half clock before sampling the last bit of the next receive data from the INTCBnR signal generation.

**(1) Operation timing**



16.5.14 Clock timing

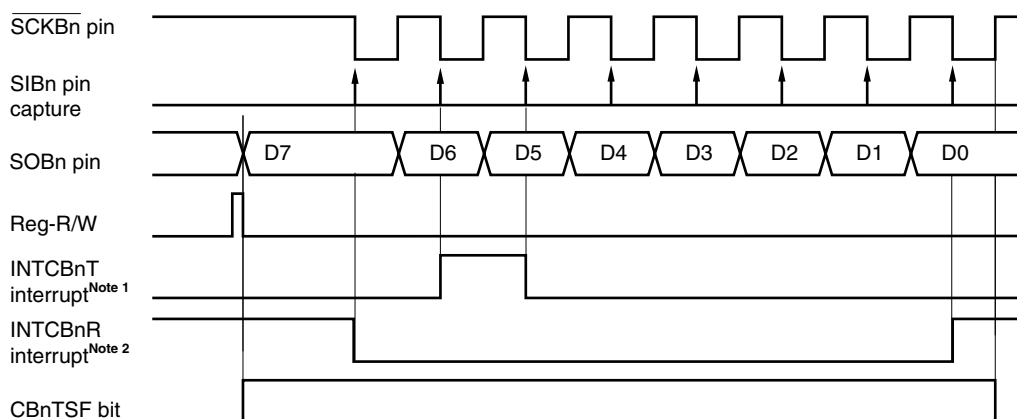
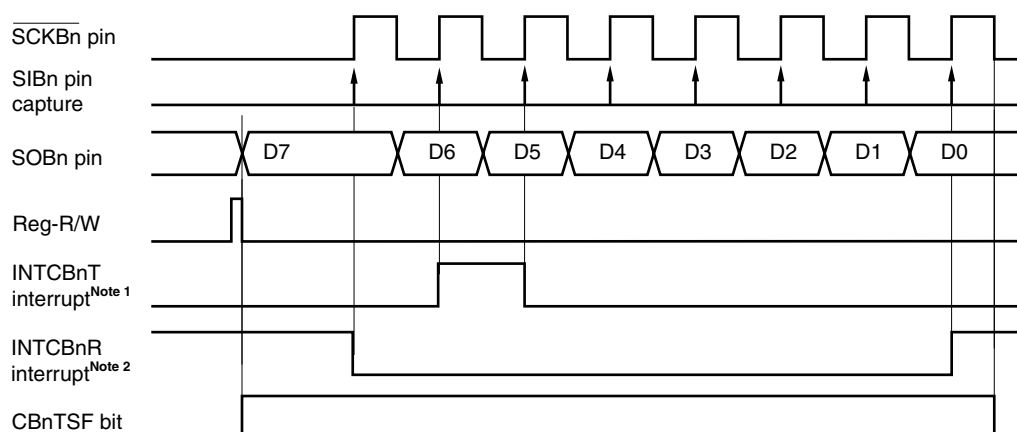
(1/2)



- 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 mode. In the single transmission or single transmission/reception mode, the INTCBnT interrupt request signal is not generated, but the INTCBnR interrupt request signal is generated upon 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. For example, if the next data is written to the CBnTX register when DMA is started by generating the INTCBnR signal, the written data is not transferred because the CBnTSF bit is set to 1. Use the continuous transfer mode, not the single transfer mode, for such applications.

**Remark** n = 0 to 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 mode. In the single transmission or single transmission/reception mode, the INTCBnT interrupt request signal is not generated, but the INTCBnR interrupt request signal is generated upon 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.

For example, if the next data is written to the CBnTX register when DMA is started by generating the INTCBnR signal, the written data is not transferred because the CBnTSF bit is set to 1.

Use the continuous transfer mode, not the single transfer mode, for such applications.

**Remark** n = 0 to 2

## 16.6 Output Pins

### (1) $\overline{\text{SCKBn}}$ pin

When CSIBn operation is disabled (CBnCTL0.CBnPWR bit = 0), the  $\overline{\text{SCKBn}}$  pin output status is as follows.

**Remark** n = 0 to 2

| CBnCKP | CBnCKS2          | CBnCKS1 | CBnCKS0 | $\overline{\text{SCKBn}}$ Pin Output |
|--------|------------------|---------|---------|--------------------------------------|
| 0      | 1                | 1       | 1       | High impedance                       |
|        | Other than above |         |         | Fixed to high level                  |
| 1      | 1                | 1       | 1       | High impedance                       |
|        | Other than above |         |         | Fixed to low level                   |

**Remark** The output level of the  $\overline{\text{SCKBn}}$  pin changes if any of the CBnCTL1.CBnCKP and CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits is rewritten.

### (2) SOBn pin

When CSIBn operation is disabled (CBnPWR bit = 0), the SOBn pin output status is as follows.

**Remark** n = 0 to 2

| CBnTXE | CBnDAP | CBnDIR | SOBn Pin Output              |
|--------|--------|--------|------------------------------|
| 0      | ×      | ×      | Fixed to low level           |
| 1      | 0      | ×      | SOBn latch value (low level) |
|        | 1      | 0      | CBnTX value (MSB)            |
|        |        | 1      | CBnTX value (LSB)            |

- Remarks**
- The SOBn pin output changes when any one of the CBnCTL0.CBnTXE, CBnCTL0.CBnDIR, or CBnCTL1.CBnDAP bits is rewritten.
  - ×: Don't care

## CHAPTER 17 I<sup>2</sup>C BUS

To use the I<sup>2</sup>C bus function, use the P30/SCL and P31/SDA pins as the serial transmit/receive data and set them to N-ch open-drain output.

In the V850E/IF3 and V850E/IG3, one channel of I<sup>2</sup>C bus is provided.

### 17.1 Mode Switching Between I<sup>2</sup>C and UARTA1

In the V850E/IF3 and V850E/IG3, I<sup>2</sup>C and UARTA1 function alternately, and these pins cannot be used at the same time. To switch between I<sup>2</sup>C and UARTA1, the PMC3, PFC3, and PFCE3 registers must be set in advance.

**Caution** The operations related to transmission and reception of I<sup>2</sup>C or UARTA1 are not guaranteed if the mode is switched during transmission or reception. Be sure to disable the unit that is not used.

**Figure 17-1. Mode Switch Settings of I<sup>2</sup>C and UARTA1**

|       |                  |        |                    |  |       |        |        |        |  |  |  |
|-------|------------------|--------|--------------------|--|-------|--------|--------|--------|--|--|--|
|       | After reset: 00H | R/W    | Address: FFFFF446H |  |       |        |        |        |  |  |  |
|       | 7                | 6      | 5                  | 4  | 3     | 2      | 1      | 0      |  |  |  |
| PMC3  | PMC37            | PMC36  | PMC35              | PMC34  | PMC33 | PMC32  | PMC31  | PMC30  |  |  |  |
|       |                  |        |                    |  |       |        |        |        |  |  |  |
|       | 7                | 6      | 5                  | 4  | 3     | 2      | 1      | 0      |  |  |  |
| PFC3  | PFC37            | PFC36  | PFC35              | PFC34  | PFC33 | PFC32  | PFC31  | PFC30  |  |  |  |
|       |                  |        |                    |  |       |        |        |        |  |  |  |
|       | 7                | 6      | 5                  | 4  | 3     | 2      | 1      | 0      |  |  |  |
| PFCE3 | PFCE37           | PFCE36 | PFCE35             | PFCE34   | 0     | PFCE32 | PFCE31 | PFCE30 |  |  |  |
|       |                  |        |                    |  |       |        |        |        |  |  |  |
|       | 7                | 6      | 5                  | 4  | 3     | 2      | 1      | 0      |  |  |  |
|       | PMC31            | PFCE31 | PFC31              | Specification of alternate function of P31 pin |       |        |        |        |  |  |  |
|       | 0                | ×      | ×                  | I/O port                                       |       |        |        |        |  |  |  |
|       | 1                | 0      | 0                  | TXDA1 output                                   |       |        |        |        |  |  |  |
|       | 1                | 0      | 1                  | SDA I/O  |       |        |        |        |  |  |  |
|       | 1                | 1      | 0                  | Setting prohibited                             |       |        |        |        |  |  |  |
|       | 1                | 1      | 1                  | Setting prohibited                             |       |        |        |        |  |  |  |
|       |                  |        |                    |  |       |        |        |        |  |  |  |
|       | 7                | 6      | 5                  | 4  | 3     | 2      | 1      | 0      |  |  |  |
|       | PMC30            | PFCE30 | PFC30              | Specification of alternate function of P30 pin |       |        |        |        |  |  |  |
|       | 0                | ×      | ×                  | I/O port                                       |       |        |        |        |  |  |  |
|       | 1                | 0      | 0                  | RXDA1 input                                    |       |        |        |        |  |  |  |
|       | 1                | 0      | 1                  | SCL I/O  |       |        |        |        |  |  |  |
|       | 1                | 1      | 0                  | Setting prohibited                             |       |        |        |        |  |  |  |
|       | 1                | 1      | 1                  | Setting prohibited                             |       |        |        |        |  |  |  |

**Remark** × = don't care

## 17.2 Features

The I<sup>2</sup>C has the following two modes.

- Operation stop mode
- I<sup>2</sup>C (Inter IC) bus mode (multimaster supported)

### (1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

### (2) I<sup>2</sup>C bus mode (multimaster supported)

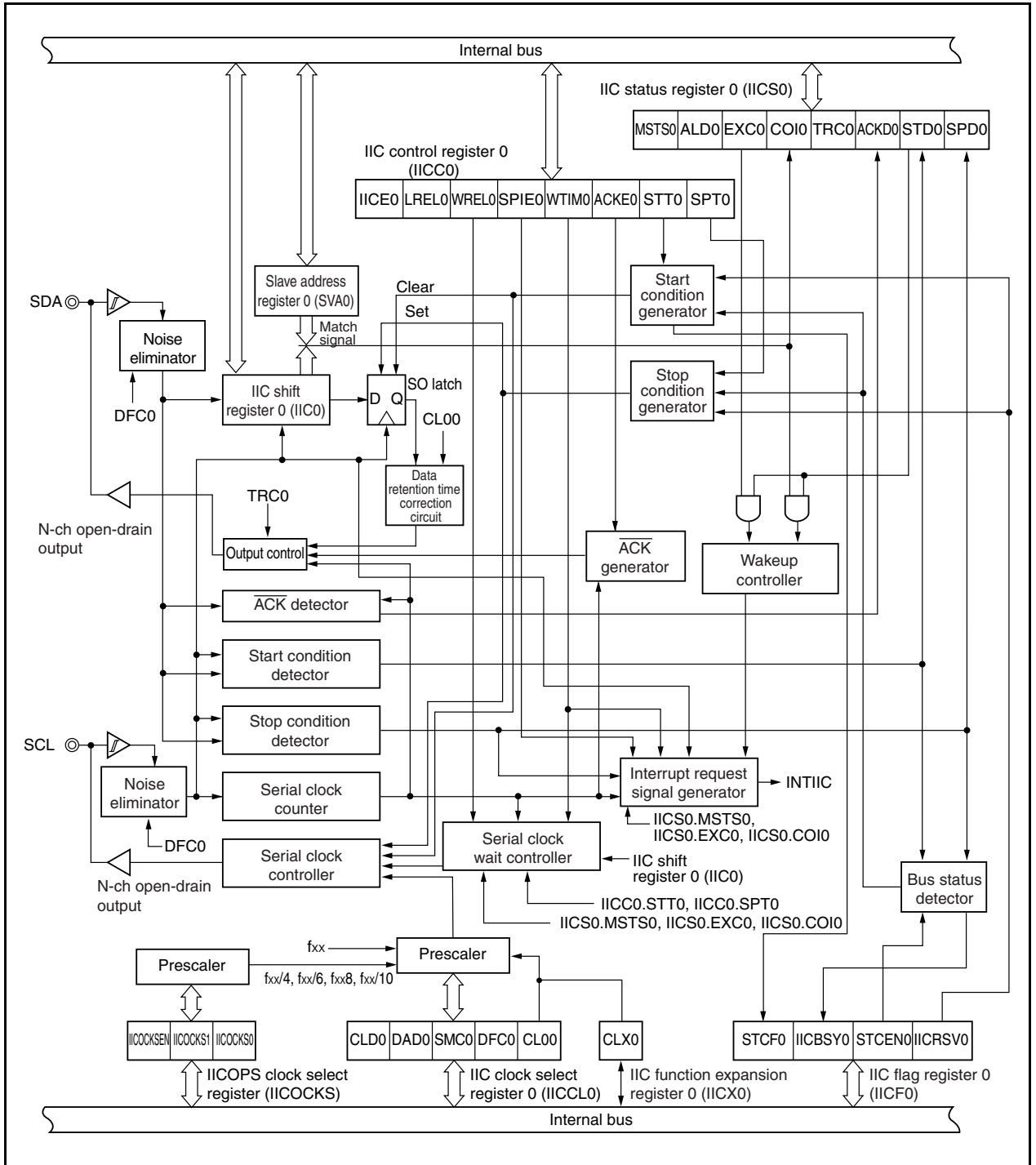
This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCL) line and a serial data bus (SDA) line.

This mode complies with the I<sup>2</sup>C bus format and the master device can generate “start condition”, “address”, “transfer direction specification”, “data”, and “stop condition” data to the slave device, via the serial data bus. The slave device automatically detects these received state and data by hardware. This function can simplify the part of application program that controls the I<sup>2</sup>C bus.

Since the SCL and SDA pins are used for N-ch open drain outputs, I<sup>2</sup>C requires pull-up resistors for the serial clock line and the serial data bus line.

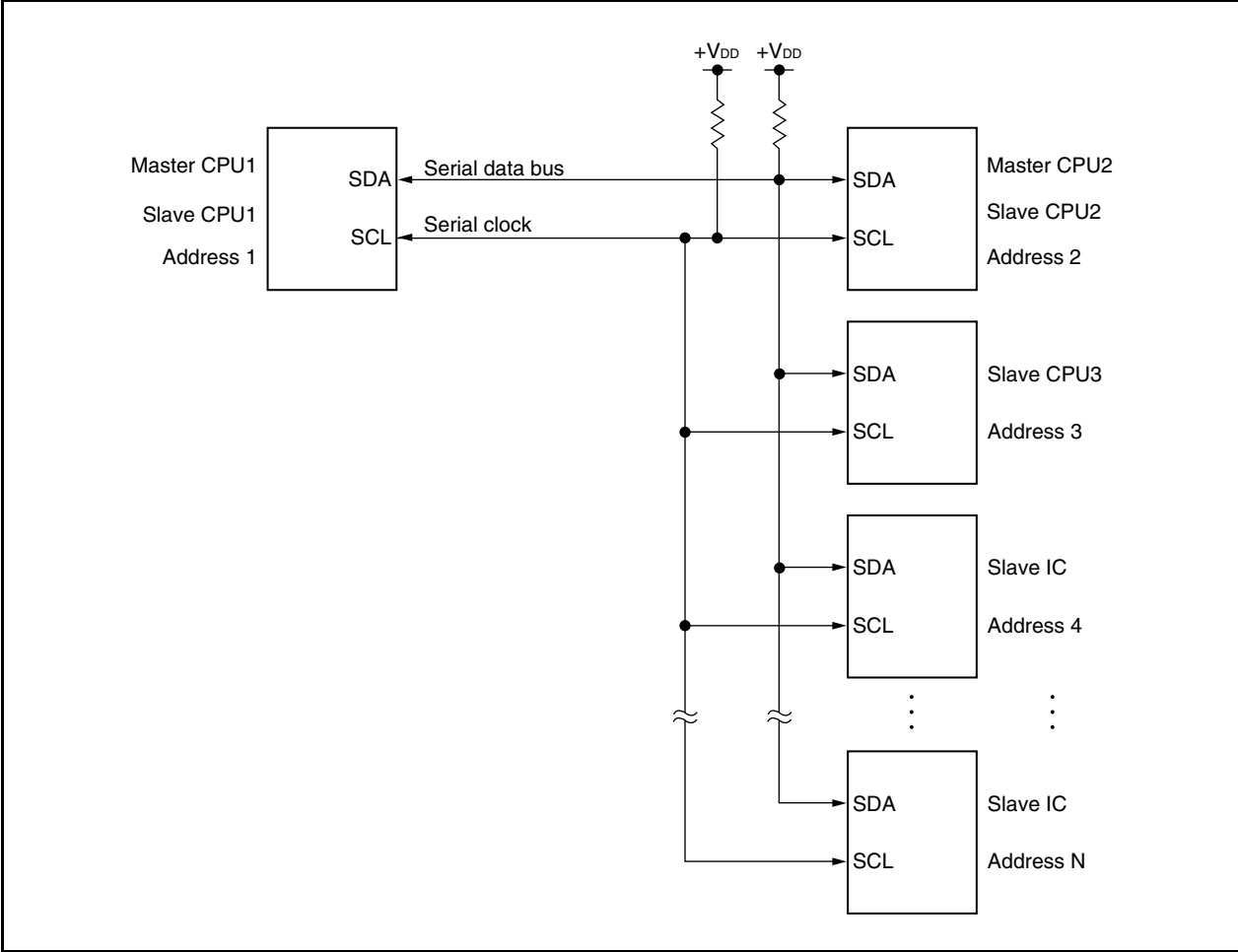


Figure 17-2. Block Diagram of I<sup>2</sup>C



A serial bus configuration example is shown below.

Figure 17-3. Serial Bus Configuration Example Using I<sup>2</sup>C Bus



### 17.3 Configuration

I<sup>2</sup>C includes the following hardware.

**Table 17-1. Configuration of I<sup>2</sup>C**

| Item              | Configuration   |
|-------------------|---|
| Registers         | IIC shift register 0 (IIC0)<br>Slave address register 0 (SVA0)  |
| Control registers | IIC control register 0 (IICC0)<br>IIC status register 0 (IICS0)<br>IIC flag register 0 (IICF0)<br>IIC clock select register 0 (IICCL0)<br>IIC function expansion register 0 (IICX0)<br>IICOPS clock select register (IICOCKS) |

#### (1) IIC shift register 0 (IIC0)

The IIC0 register is used to convert 8-bit serial data to 8-bit parallel data and to convert 8-bit parallel data to 8-bit serial data. The IIC0 register can be used for both transmission and reception.

Write and read operations to the IIC0 register are used to control the actual transmit and receive operations.

The IIC0 register can be read or written in 8-bit units.

Reset sets IIC0 to 00H.

#### (2) Slave address register 0 (SVA0)

The SVA0 register sets local addresses when in slave mode.

The SVA0 register can be read or written in 8-bit units.

Reset sets SVA0 to 00H.

#### (3) SO latch

The SO latch is used to retain the SDA pin's output level.

#### (4) Wakeup controller

This circuit generates an interrupt request signal (INTIIC) when the address received by this register matches the address value set to the SVA0 register or when an extension code is received.

#### (5) Prescaler

This selects the sampling clock to be used.

#### (6) Serial clock counter

This counter counts the serial clocks that are output and the serial clocks that are input during transmit/receive operations and is used to verify that 8-bit data was sent or received.

#### (7) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (INTIIC).

An I<sup>2</sup>C interrupt is generated following either of two triggers.

- Falling of the eighth or ninth clock of the serial clock (set by IICC0.WTIM0 bit)
- Interrupt request generated when a stop condition is detected (set by IICC0.SPIE0 bit)

**(8) Serial clock controller**

In master mode, this circuit generates the clock output via the SCL pin from a sampling clock.

**(9) Serial clock wait controller**

This circuit controls the wait timing.

**(10)  $\overline{\text{ACK}}$  generator, stop condition detector, start condition detector, and  $\overline{\text{ACK}}$  detector**

These circuits are used to generate and detect various statuses.

**(11) Data hold time correction circuit**

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

**(12) Start condition generator**

This circuit generates a start condition when the IICC0.STT0 bit is set.

However, in the communication reservation disabled status (IICF0.IICRSV0 bit = 1), when the bus is not released (IICF0.IICBSY0 bit = 1), start condition requests are ignored and the IICF0.STCF0 bit is set to 1.

**(13) Stop condition generator**

A stop condition is generated when the IIC0.SPT0 bit is set (1).

**(14) Bus status detector**

This circuit detects whether or not the bus is released by detecting start conditions and stop conditions.

However, as the bus status cannot be detected immediately following operation, the initial status is set by the IICF0.STCEN0 bit.

## 17.4 Registers

I<sup>2</sup>C is controlled by the following registers.

- IIC control register 0 (IICC0)
- IIC status register 0 (IICS0)
- IIC flag register 0 (IICF0)
- IIC clock select register 0 (IICCL0)
- IIC function expansion register 0 (IICX0)
- IICOPS clock select register (IICOCKS)

The following registers are also used.

- IIC shift register 0 (IIC0)
- Slave address register 0 (SVA0)

**Remark** For the alternate-function pin settings, see **Table 4-14 Settings When Port Pins Are Used for Alternate Functions**.

**(1) IIC control register 0 (IICC0)**

The IICC0 register is used to enable/stop I<sup>2</sup>C operations, set wait timing, and set other I<sup>2</sup>C operations.

The IICC0 register can be read or written in 8-bit or 1-bit units. However, set the SPIE0, WTIM0, and ACKE0 bits when the IICE0 bit is 0 or during the wait period. When setting the IICE0 bit from “0” to “1”, these bits can also be set at the same time.

Reset sets this register to 00H.

(1/4)

After reset: 00H      R/W      Address: FFFFFFFD82H

|       |                  |                  |                  |                  |                  |                  |                  |                  |
|-------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
|       | <b>&lt;7&gt;</b> | <b>&lt;6&gt;</b> | <b>&lt;5&gt;</b> | <b>&lt;4&gt;</b> | <b>&lt;3&gt;</b> | <b>&lt;2&gt;</b> | <b>&lt;1&gt;</b> | <b>&lt;0&gt;</b> |
| IICC0 | IICE0            | LRELO            | WRELO            | SPIE0            | WTIM0            | ACKE0            | STT0             | SPT0             |

|   |  |
|---|--|
| IICE0   | I <sup>2</sup> C operation enable/disable specification                              |
| 0   | Stop operation. Reset the IICS0 register <sup>Note1</sup> . Stop internal operation. |
| 1   | Enable operation.  |
| Be sure to set this bit to 1 when the SCL and SDA lines are high level.                     |  |
| Condition for clearing (IICE0 bit = 0)  | Condition for setting (IICE0 bit = 1)  |
| <ul style="list-style-type: none"> <li>• Cleared by instruction</li> <li>• Reset</li> </ul> | <ul style="list-style-type: none"> <li>• Set by instruction</li> </ul>               |

|  |   |
|--|---|
| LRELO <sup>Note2</sup>   | Exit from communications  |
| 0  | Normal operation  |
| 1  | This exits from the current communications and sets standby mode. This setting is automatically cleared to 0 after being executed.<br>Its uses include cases in which a locally irrelevant extension code has been received.<br>The SCL and SDA lines are set to high impedance.<br>The STT0, SPT0, IICS0.MSTS0, IICS0.EXC0, IICS0.COIO, IICS0.TRC0, IICS0.ACKD0, and IICS0.STD0 bits are cleared to 0. |
| The standby mode following exit from communications remains in effect until the following communications entry conditions are met.   |   |
| <ul style="list-style-type: none"> <li>• After a stop condition is detected, restart is in master mode.</li> <li>• An address match or extension code reception occurs after the start condition.</li> </ul> |   |
| Condition for clearing (LRELO bit = 0)   | Condition for setting (LRELO bit = 1)   |
| <ul style="list-style-type: none"> <li>• Automatically cleared after execution</li> <li>• Reset</li> </ul>   | <ul style="list-style-type: none"> <li>• Set by instruction</li> </ul>  |

**Notes**

1. The IICS0 register, and the IICF0.STCF0, IICF0.IICBSY0, IICCL0.CLD0, and IICCL0.DAD0 bits are reset.
2. This flag's signal is invalid when the IICE0 bit = 0.

**Caution** If the I<sup>2</sup>C operation is enabled (IICE0 bit = 1) when the SCL line is high level and the SDA line is low level, the start condition is detected immediately. To avoid this, after enabling the I<sup>2</sup>C operation, immediately set the LRELO bit to 1 with a bit manipulation instruction.

| WRELO <sup>Note</sup>  | Wait cancellation control   |
|--|---|
| 0  | Do not cancel wait  |
| 1  | Cancel wait. This setting is automatically cleared to 0 after wait is canceled. |
| Condition for clearing (WRELO bit = 0)   |   |
| <ul style="list-style-type: none"> <li>Automatically cleared after execution</li> <li>Reset</li> </ul> |   |
| Condition for setting (WRELO bit = 1)  |   |
| <ul style="list-style-type: none"> <li>Set by instruction</li> </ul>                                   |   |

| SPIE0 <sup>Note</sup>   | Enable/disable generation of interrupt request when stop condition is detected |
|---|--|
| 0   | Disable  |
| 1   | Enable   |
| Condition for clearing (SPIE0 bit = 0)  |  |
| <ul style="list-style-type: none"> <li>Cleared by instruction</li> <li>Reset</li> </ul> |  |
| Condition for setting (SPIE0 bit = 1)   |  |
| <ul style="list-style-type: none"> <li>Set by instruction</li> </ul>                    |  |

| WTIMO <sup>Note</sup>   | Control of wait and interrupt request generation  |
|---|---|
| 0   | Interrupt request is generated at the eighth clock's falling edge.<br>Master mode: After output of eight clocks, clock output is set to low level and wait is set.<br>Slave mode: After input of eight clocks, the clock is set to low level and wait is set for master device. |
| 1   | Interrupt request is generated at the ninth clock's falling edge.<br>Master mode: After output of nine clocks, clock output is set to low level and wait is set.<br>Slave mode: After input of nine clocks, the clock is set to low level and wait is set for master device.    |
| An interrupt is generated at the falling of the 9th clock during address transfer independently of the setting of this bit. The setting of this bit is valid when the address transfer is completed. When in master mode, a wait is inserted at the falling edge of the ninth clock during address transfers. For a slave device that has received a local address, a wait is inserted at the falling edge of the ninth clock after $\overline{\text{ACK}}$ is issued. However, when the slave device has received an extension code, a wait is inserted at the falling edge of the eighth clock. |   |
| Condition for clearing (WTIMO bit = 0)  |   |
| <ul style="list-style-type: none"> <li>Cleared by instruction</li> <li>Reset</li> </ul>   |   |
| Condition for setting (WTIMO bit = 1)   |   |
| <ul style="list-style-type: none"> <li>Set by instruction</li> </ul>  |   |

| ACKE0 <sup>Note</sup>   | Acknowledgment control  |
|---|---|
| 0   | Disable acknowledgment.   |
| 1   | Enable acknowledgment. During the ninth clock period, the SDA line is set to low level. |
| The ACKE0 bit setting is invalid for address reception. In this case, $\overline{\text{ACK}}$ is generated when the addresses match. However, the ACKE0 bit setting is valid for address reception of the extension code. |   |
| Condition for clearing (ACKE0 bit = 0)  |   |
| <ul style="list-style-type: none"> <li>Cleared by instruction</li> <li>Reset</li> </ul>   |   |
| Condition for setting (ACKE0 bit = 1)   |   |
| <ul style="list-style-type: none"> <li>Set by instruction</li> </ul>  |   |

**Note** This flag's signal is invalid when the IICE0 bit = 0.

| STT0  | Start condition trigger   |                                       |                                      |   |  |
|---|---|---------------------------------------|--------------------------------------|---|--|
| 0   | Do not generate a start condition.  |                                       |                                      |   |  |
| 1   | <p>When bus is released (in STOP mode):<br/>                     Generate a start condition (for starting as master). The SDA line is changed from high level to low level while the SCL line is high level and then the start condition is generated. Next, after the rated amount of time has elapsed, the SCL line is changed to low level (wait status).</p> <p>When a third party is communicating</p> <ul style="list-style-type: none"> <li>• When communication reservation function is enabled (IICF0.IICRSV0 bit = 0)<br/>                             Functions as the start condition reservation flag. When set to 1, automatically generates a start condition after the bus is released.</li> <li>• When communication reservation function is disabled (IICRSV0 bit = 1)<br/>                             The IICF0.STCF0 bit is set to 1 and the information set (1) to the STT0 bit is cleared. No start condition is generated.</li> </ul> <p>In the wait state (when master device):<br/>                     Generates a restart condition after releasing the wait.</p> |                                       |                                      |   |  |
| <p>Cautions concerning set timing</p> <p>For master reception: Cannot be set to 1 during transfer. Can be set to 1 only when the ACKE0 bit has been cleared to 0 and slave has been notified of final reception.</p> <p>For master transmission: A start condition may not be generated normally during the ACK period. Set to 1 during the wait period that follows output of the ninth clock.</p> <ul style="list-style-type: none"> <li>• Cannot be set to 1 at the same time as the SPT0 bit.</li> <li>• When the STT0 bit is set to 1, setting the STT0 bit to 1 again is disabled until the setting is cleared to 0.</li> </ul>                             |   |                                       |                                      |   |  |
| <table border="1"> <thead> <tr> <th>Condition for clearing (STT0 bit = 0)</th> <th>Condition for setting (STT0 bit = 1)</th> </tr> </thead> <tbody> <tr> <td> <ul style="list-style-type: none"> <li>• When the STT0 bit is set to 1 in the communication reservation disabled status</li> <li>• Cleared by loss in arbitration</li> <li>• Cleared when start condition is generated by master device</li> <li>• When the LREL0 bit = 1 (exit from communications)</li> <li>• When the IICE0 bit = 0 (operation stop)</li> <li>• Reset</li> </ul> </td> <td> <ul style="list-style-type: none"> <li>• Set by instruction</li> </ul> </td> </tr> </tbody> </table> |   | Condition for clearing (STT0 bit = 0) | Condition for setting (STT0 bit = 1) | <ul style="list-style-type: none"> <li>• When the STT0 bit is set to 1 in the communication reservation disabled status</li> <li>• Cleared by loss in arbitration</li> <li>• Cleared when start condition is generated by master device</li> <li>• When the LREL0 bit = 1 (exit from communications)</li> <li>• When the IICE0 bit = 0 (operation stop)</li> <li>• Reset</li> </ul> | <ul style="list-style-type: none"> <li>• Set by instruction</li> </ul> |
| Condition for clearing (STT0 bit = 0)   | Condition for setting (STT0 bit = 1)  |                                       |                                      |   |  |
| <ul style="list-style-type: none"> <li>• When the STT0 bit is set to 1 in the communication reservation disabled status</li> <li>• Cleared by loss in arbitration</li> <li>• Cleared when start condition is generated by master device</li> <li>• When the LREL0 bit = 1 (exit from communications)</li> <li>• When the IICE0 bit = 0 (operation stop)</li> <li>• Reset</li> </ul>   | <ul style="list-style-type: none"> <li>• Set by instruction</li> </ul>  |                                       |                                      |   |  |

**Remark** The STT0 bit is 0 if it is read after data setting.



| SPT0   | Stop condition trigger  |                                       |                                      |   |  |
|--|---|---------------------------------------|--------------------------------------|---|--|
| 0  | Stop condition is not generated.  |                                       |                                      |   |  |
| 1  | Stop condition is generated (termination of master device's transfer).<br>After the SDA line goes to low level, either set the SCL line to high level or wait until the SCL pin goes to high level. Next, after the rated amount of time has elapsed, the SDA line is changed from low level to high level and a stop condition is generated. |                                       |                                      |   |  |
| <p>Cautions concerning setting timing</p> <p>For master reception: Cannot be set to 1 during transfer. Can be set to 1 only when the ACKE0 bit has been cleared to 0 and during the wait period after slave has been notified of final reception.</p> <p>For master transmission: A stop condition may not be generated normally during the <math>\overline{\text{ACK}}</math> period. Set to 1 during the wait period that follows output of the ninth clock.</p> <ul style="list-style-type: none"> <li>• Cannot be set to 1 at the same time as the STT0 bit.</li> <li>• The SPT0 bit can be set to 1 only when in master mode<sup>Note</sup>.</li> <li>• When the WTIM0 bit has been cleared to 0, if the SPT0 bit is set to 1 during the wait period that follows output of eight clocks, note that a stop condition will be generated during the high-level period of the ninth clock. The WTIM0 bit should be changed from 0 to 1 during the wait period following output of eight clocks, and the SPT0 bit should be set to 1 during the wait period that follows output of the ninth clock.</li> <li>• When the SPT0 bit is set to 1, setting the SPT0 bit to 1 again is disabled until the setting is cleared to 0.</li> </ul> |   |                                       |                                      |   |  |
| <table border="1"> <thead> <tr> <th>Condition for clearing (SPT0 bit = 0)</th> <th>Condition for setting (SPT0 bit = 1)</th> </tr> </thead> <tbody> <tr> <td> <ul style="list-style-type: none"> <li>• Cleared by loss in arbitration</li> <li>• Automatically cleared after stop condition is detected</li> <li>• When the LREL0 bit = 1 (exit from communications)</li> <li>• When the IICE0 bit = 0 (operation stop)</li> <li>• Reset</li> </ul> </td> <td> <ul style="list-style-type: none"> <li>• Set by instruction</li> </ul> </td> </tr> </tbody> </table>  |   | Condition for clearing (SPT0 bit = 0) | Condition for setting (SPT0 bit = 1) | <ul style="list-style-type: none"> <li>• Cleared by loss in arbitration</li> <li>• Automatically cleared after stop condition is detected</li> <li>• When the LREL0 bit = 1 (exit from communications)</li> <li>• When the IICE0 bit = 0 (operation stop)</li> <li>• Reset</li> </ul> | <ul style="list-style-type: none"> <li>• Set by instruction</li> </ul> |
| Condition for clearing (SPT0 bit = 0)  | Condition for setting (SPT0 bit = 1)  |                                       |                                      |   |  |
| <ul style="list-style-type: none"> <li>• Cleared by loss in arbitration</li> <li>• Automatically cleared after stop condition is detected</li> <li>• When the LREL0 bit = 1 (exit from communications)</li> <li>• When the IICE0 bit = 0 (operation stop)</li> <li>• Reset</li> </ul>  | <ul style="list-style-type: none"> <li>• Set by instruction</li> </ul>  |                                       |                                      |   |  |

**Note** Set the SPT0 bit to 1 only in master mode. However, the SPT0 bit must be set to 1 and a stop condition generated before the first stop condition is detected following the switch to operation enable status. For details, see **17.15 Cautions**.

**Caution** When the IICS0.TRC0 bit is set to 1, the WREL0 bit is set to 1 during the ninth clock and wait is canceled, after which the TRC0 bit is cleared to 0 and the SDA line is set to high impedance.

**Remark** The SPT0 bit is 0 if it is read after data setting.

**(2) IIC status register 0 (IICS0)**

The IICS0 register indicates the status of the I<sup>2</sup>C bus.  
 The IICS0 register is read-only, in 8-bit or 1-bit units.  
 However, the IICS0 register can only be read when the IICC0.STT0 bit is 1 or during the wait period.  
 Reset sets this register to 00H.

(1/3)

After reset: 00H      R      Address: FFFFFFFD86H

|       |                  |                  |                  |                  |                  |                  |                  |                  |
|-------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
|       | <b>&lt;7&gt;</b> | <b>&lt;6&gt;</b> | <b>&lt;5&gt;</b> | <b>&lt;4&gt;</b> | <b>&lt;3&gt;</b> | <b>&lt;2&gt;</b> | <b>&lt;1&gt;</b> | <b>&lt;0&gt;</b> |
| IICS0 | MSTS0            | ALD0             | EXC0             | COI0             | TRC0             | ACKD0            | STD0             | SPD0             |

|  |   |
|--|---|
| <b>MSTS0</b>   | <b>Master device status</b>                         |
| 0  | Slave device status or communication standby status |
| 1  | Master device communication status                  |
| Condition for clearing (MSTS0 bit = 0)   |   |
| <ul style="list-style-type: none"> <li>• When a stop condition is detected</li> <li>• When the ALD0 bit = 1 (arbitration loss)</li> <li>• Cleared by the IICC0.LRELO bit = 1 (exit from communications)</li> <li>• When the IICC0.IICE0 bit changes from 1 to 0 (operation stop)</li> <li>• Reset</li> </ul> |   |
| Condition for setting (MSTS0 bit = 1)  |   |
| <ul style="list-style-type: none"> <li>• When a start condition is generated</li> </ul>  |   |

|   |  |
|---|--|
| <b>ALD0</b>   | <b>Detection of arbitration loss</b>   |
| 0   | This status means either that there was no arbitration or that the arbitration result was a “win”. |
| 1   | This status indicates the arbitration result was a “loss”. The MSTS0 bit is cleared to 0.          |
| Condition for clearing (ALD0 bit = 0)   |  |
| <ul style="list-style-type: none"> <li>• Automatically cleared after the IICS0 register is read<sup>Note</sup></li> <li>• When the IICE0 bit changes from 1 to 0 (operation stop)</li> <li>• Reset</li> </ul> |  |
| Condition for setting (ALD0 bit = 1)  |  |
| <ul style="list-style-type: none"> <li>• When the arbitration result is a “loss”.</li> </ul>  |  |

**Note** This bit is also cleared when a bit manipulation instruction is executed for another bit in the IICS0 register.

|  |                                       |   |
|--|---------------------------------------|---|
| EXC0   | Detection of extension code reception |   |
| 0  | Extension code was not received.      |   |
| 1  | Extension code was received.          |   |
| Condition for clearing (EXC0 bit = 0)  |                                       | Condition for setting (EXC0 bit = 1)  |
| <ul style="list-style-type: none"> <li>• When a start condition is detected</li> <li>• When a stop condition is detected</li> <li>• Cleared by the LREL0 bit = 1 (exit from communications)</li> <li>• When the IICE0 bit changes from 1 to 0 (operation stop)</li> <li>• Reset</li> </ul> |                                       | <ul style="list-style-type: none"> <li>• When the higher four bits of the received address data is either "0000" or "1111" (set at the rising edge of the eighth clock).</li> </ul> |

|   |                                 |   |
|---|---------------------------------|---|
| COI0  | Detection of matching addresses |   |
| 0   | Addresses do not match.         |   |
| 1   | Addresses match.                |   |
| Condition for clearing (COI0 bit = 0)   |                                 | Condition for setting (COI0 bit = 1)  |
| <ul style="list-style-type: none"> <li>• When a start condition is detected</li> <li>• When a stop condition is detected</li> <li>• Cleared by the LREL0 bit = 1 (exit from communications)</li> <li>• When the IICE0 bit changes from 1 to 0</li> <li>• Reset</li> </ul> |                                 | <ul style="list-style-type: none"> <li>• When the received address matches the local address (SVA0 register) (set at the rising edge of the eighth clock).</li> </ul> |

|   |   |  |
|---|---|--|
| TRC0  | Detection of transmit/receive status  |  |
| 0   | Receive status (other than transmit status). The SDA line is set for high impedance.  |  |
| 1   | Transmit status. The value in the SO latch is enabled for output to the SDA line (valid starting at the rising edge of the first byte's ninth clock). |  |
| Condition for clearing (TRC0 bit = 0)   |   | Condition for setting (TRC0 bit = 1)   |
| <ul style="list-style-type: none"> <li>• When a stop condition is detected</li> <li>• Cleared by the LREL0 bit = 1 (exit from communications)</li> <li>• When the IICE0 bit changes from 1 to 0 (operation stop)</li> <li>• Cleared by the IICC0.WREL0 bit = 1<sup>Note</sup> (wait release)</li> <li>• When the ALD0 bit changes from 0 to 1 (arbitration loss)</li> <li>• Reset</li> </ul> <p>Master</p> <ul style="list-style-type: none"> <li>• When "1" is output to the first byte's LSB (transfer direction specification bit)</li> </ul> <p>Slave</p> <ul style="list-style-type: none"> <li>• When a start condition is detected</li> </ul> <p>When not used for communication</p> |   | <p>Master</p> <ul style="list-style-type: none"> <li>• When a start condition is generated</li> <li>• When "0" is output to the first byte's LSB (transfer direction specification bit)</li> </ul> <p>Slave</p> <ul style="list-style-type: none"> <li>• When "1" is input in the first byte's LSB (transfer direction specification bit)</li> </ul> |

**Note** The IICS0.TR0 bit is cleared to 0 and the SDA line become high impedance when the IICC0.WREL0 bit is set to 1 and wait state is released at the ninth clock with the TRC0 bit = 1.

| ACKD0   | Detection of $\overline{\text{ACK}}$      |   |
|---|---|---|
| 0   | $\overline{\text{ACK}}$ was not detected. |   |
| 1   | $\overline{\text{ACK}}$ was detected.     |   |
| Condition for clearing (ACKD0 bit = 0)  |   | Condition for setting (ACKD0 bit = 1)   |
| <ul style="list-style-type: none"> <li>• When a stop condition is detected</li> <li>• At the rising edge of the next byte's first clock</li> <li>• Cleared by the LREL0 bit = 1 (exit from communications)</li> <li>• When the IICE0 bit changes from 1 to 0 (operation stop)</li> <li>• Reset</li> </ul> |   | <ul style="list-style-type: none"> <li>• After the SDA pin is set to low level at the rising edge of the SCL pin's ninth clock</li> </ul> |

| STD0   | Detection of start condition   |  |
|--|--|--|
| 0  | Start condition was not detected.  |  |
| 1  | Start condition was detected. This indicates that the address transfer period is in effect |  |
| Condition for clearing (STD0 bit = 0)  |  | Condition for setting (STD0 bit = 1)   |
| <ul style="list-style-type: none"> <li>• When a stop condition is detected</li> <li>• At the rising edge of the next byte's first clock following address transfer</li> <li>• Cleared by the LREL0 bit = 1 (exit from communications)</li> <li>• When the IICE0 bit changes from 1 to 0 (operation stop)</li> <li>• Reset</li> </ul> |  | <ul style="list-style-type: none"> <li>• When a start condition is detected</li> </ul> |

| SPD0   | Detection of stop condition   |   |
|--|---|---|
| 0  | Stop condition was not detected.  |   |
| 1  | Stop condition was detected. The master device's communication is terminated and the bus is released. |   |
| Condition for clearing (SPD0 bit = 0)  |   | Condition for setting (SPD0 bit = 1)  |
| <ul style="list-style-type: none"> <li>• At the rising edge of the address transfer byte's first clock following setting of this bit and detection of a start condition</li> <li>• When the IICE0 bit changes from 1 to 0 (operation stop)</li> <li>• Reset</li> </ul> |   | <ul style="list-style-type: none"> <li>• When a stop condition is detected</li> </ul> |

**(3) IIC flag register 0 (IICF0)**

IICF0 is a register that set the operation mode of I<sup>2</sup>C and indicate the status of the I<sup>2</sup>C bus.

These registers can be read or written in 8-bit or 1-bit units. However, the STCF0 and IICBSY0 bits are read-only.

The IICRSV0 bit can be used to enable/disable the communication reservation function (see **17.14 Communication Reservation**).

The STCEN0 bit can be used to set the initial value of the IICBSY0 bit (see **17.15 Cautions**).

The IICRSV0 and STCEN0 bits can be written only when the operation of I<sup>2</sup>C is disabled (IICC0.IICE0 bit = 0).

When operation is enabled, the IICF0 register can be read.

Reset sets this register to 00H.

After reset: 00H R/W<sup>Note</sup> Address: FFFFFFFD8AH

|       |       |         |   |   |   |   |        |         |
|-------|-------|---------|---|---|---|---|--------|---------|
|       | <7>   | <6>     | 5 | 4 | 3 | 2 | <1>    | <0>     |
| IICF0 | STCF0 | IICBSY0 | 0 | 0 | 0 | 0 | STCEN0 | IICRSV0 |

|   |  |
|---|--|
| STCF0   | IIC0.STT0 clear flag                                     |
| 0   | Generate start condition                                 |
| 1   | Start condition generation unsuccessful: clear STT0 flag |
| Condition for clearing (STCF0 bit = 0)  |  |
| <ul style="list-style-type: none"> <li>Clearing by setting the STT0 bit = 1</li> <li>When the IICE0 bit = 1 → 0 (operation stop)</li> <li>Reset</li> </ul>                            |  |
| Condition for setting (STCF0 bit = 1)   |  |
| <ul style="list-style-type: none"> <li>Generating start condition unsuccessful and the STT0 bit cleared to 0 when communication reservation is disabled (IICRSV0 bit = 1).</li> </ul> |  |

|   |   |
|---|---|
| IICBSY0   | I <sup>2</sup> C bus status flag  |
| 0   | Bus release status (initial communication status when STCEN0 bit = 1)       |
| 1   | Bus communication status (initial communication status when STCEN0 bit = 0) |
| Condition for clearing (IICBSY0 bit = 0)  |   |
| <ul style="list-style-type: none"> <li>Detection of stop condition</li> <li>When the IICE0 bit = 1 → 0 (operation stop)</li> <li>Reset</li> </ul> |   |
| Condition for setting (IICBSY0 bit = 1)   |   |
| <ul style="list-style-type: none"> <li>Detection of start condition</li> <li>Setting of the IICE0 bit when the STCEN0 bit = 0</li> </ul>          |   |

|   |  |
|---|--|
| STCEN0  | Initial start enable trigger   |
| 0   | After operation is enabled (IICE0 bit = 1), enable generation of a start condition upon detection of a stop condition. |
| 1   | After operation is enabled (IICE0 bit = 1), enable generation of a start condition without detecting a stop condition. |
| Condition for clearing (STCEN0 bit = 0)   |  |
| <ul style="list-style-type: none"> <li>Detection of start condition</li> <li>Reset</li> </ul> |  |
| Condition for setting (STCEN0 bit = 1)  |  |
| <ul style="list-style-type: none"> <li>Setting by instruction</li> </ul>                      |  |

|  |  |
|--|--|
| IICRSV0  | Communication reservation function disable bit |
| 0  | Enable communication reservation               |
| 1  | Disable communication reservation              |
| Condition for clearing (IICRSV0 bit = 0)   |  |
| <ul style="list-style-type: none"> <li>Clearing by instruction</li> <li>Reset</li> </ul> |  |
| Condition for setting (IICRSV0 bit = 1)  |  |
| <ul style="list-style-type: none"> <li>Setting by instruction</li> </ul>                 |  |

**Note** Bits 6 and 7 are read-only bits.

- Cautions**
1. Write to the STCEN0 bit only when the operation is stopped (IICE0 bit = 0).
  2. As the bus release status (IICBSY0 bit = 0) is recognized regardless of the actual bus status when the STCEN0 bit = 1, when generating the first start condition (STT0 bit = 1), it is necessary to verify that no third party communications are in progress in order to prevent such communications from being destroyed.
  3. Write to the IICRSV0 bit only when the operation is stopped (IICE0 bit = 0).

**(4) IIC clock select register 0 (IICCL0)**

The IICCL0 register is used to set the transfer clock for the I<sup>2</sup>C bus.

The IICCL0 register can be read or written in 8-bit or 1-bit units. However, the CLD0 and DAD0 bits are read-only. The SMC0 and CL00 bits are set in combination with the IICX0.CLX0, IICOCKS.IICOCKS1, and IICOCKS.IICOCKS0 bits (see **17.4 (7) I<sup>2</sup>C transfer clock setting method**).

Set the IICCL0 register when the IICC0.IICE0 bit = 0.

Reset sets this register to 00H.

|                  |  |                     |   |                    |      |      |      |   |      |
|------------------|--|---------------------|---|--------------------|------|------|------|---|------|
| After reset: 00H |  | R/W <sup>Note</sup> |   | Address: FFFFFD84H |      |      |      |   |      |
|                  |  | 7                   | 6 | <5>                | <4>  | 3    | 2    | 1 | 0    |
| IICCL0           |  | 0                   | 0 | CLD0               | DAD0 | SMC0 | DFC0 | 0 | CL00 |

|  |  |   |
|--|--|---|
| CLD0   | Detection of SCL pin level (valid only when IICC0.IICE0 bit = 1) |   |
| 0  | The SCL pin was detected at low level.                           |   |
| 1  | The SCL pin was detected at high level.                          |   |
| Condition for clearing (CLD0 bit = 0)  |  | Condition for setting (CLD0 bit = 1)  |
| <ul style="list-style-type: none"> <li>• When the SCL pin is at low level</li> <li>• When the IICE0 bit = 1 → 0 (operation stop)</li> <li>• Reset</li> </ul> |  | <ul style="list-style-type: none"> <li>• When the SCL pin is at high level</li> </ul> |

|  |  |   |
|--|--|---|
| DAD0   | Detection of SDA pin level (valid only when IICE0 bit = 1) |   |
| 0  | The SDA pin was detected at low level.                     |   |
| 1  | The SDA pin was detected at high level.                    |   |
| Condition for clearing (DAD0 bit = 0)  |  | Condition for setting (DAD0 bit = 1)  |
| <ul style="list-style-type: none"> <li>• When the SDA pin is at low level</li> <li>• When the IICE0 bit = 1 → 0 (operation stop)</li> <li>• Reset</li> </ul> |  | <ul style="list-style-type: none"> <li>• When the SDA pin is at high level</li> </ul> |

|      |                              |  |
|------|------------------------------|--|
| SMC0 | Operation mode switching     |  |
| 0    | Operates in standard mode.   |  |
| 1    | Operates in high-speed mode. |  |

|   |                                  |  |
|---|----------------------------------|--|
| DFC0  | Digital filter operation control |  |
| 0   | Digital filter off.              |  |
| 1   | Digital filter on.               |  |
| Digital filter can be used only in high-speed mode.<br>In high-speed mode, the transfer clock does not vary regardless of DFC0 bit set/clear.<br>The digital filter is used for noise elimination in high-speed mode. |                                  |  |

|      |                               |                     |
|------|-------------------------------|---------------------|
| CL00 | Communication clock selection |                     |
|      | Normal mode                   | High-speed mode     |
| 0    | F <sub>xx</sub> /44           | F <sub>xx</sub> /24 |
| 1    | F <sub>xx</sub> /86           | F <sub>xx</sub> /24 |

**Note** Bits 4 and 5 are read-only bits.

**Remark** F<sub>xx</sub>: Selection clock

**(5) IIC function expansion register 0 (IICX0)**

This register sets the function expansion of I<sup>2</sup>C (valid only in high-speed mode).

This register can be read or written in 8-bit or 1-bit units. The CLX0 bit is set in combination with the IICCL0.SMC0, IICCL0.CL00, IICOCKS.IICOCKS1, and IICOCKS.IICOCKS0 bits (see 17.4 (7) I<sup>2</sup>C transfer clock setting method).

Set the IICX0 register when the IICC0.IICE0 bit = 0.

Reset sets this register to 00H.

|                  |   |     |                      |   |   |   |   |      |
|------------------|---|-----|----------------------|---|---|---|---|------|
| After reset: 00H |   | R/W | Address: FFFFFFFD85H |   |   |   |   |      |
| IICX0            | 7   | 6   | 5                    | 4 | 3 | 2 | 1 | <0>  |
|                  | 0   | 0   | 0                    | 0 | 0 | 0 | 0 | CLX0 |
| CLX0             | Clock select expansion bit  |     |                      |   |   |   |   |      |
| 0                | Communicate at transfer rate set by the IICCL0.CL00 bit.                            |     |                      |   |   |   |   |      |
| 1                | Communicate at double transfer rate set by the IICCL0.CL00 bit in high-speed mode . |     |                      |   |   |   |   |      |

**(6) IICOPS clock select register (IICOCKS)**

This register controls the division clock of I<sup>2</sup>C.

This register can be read or written in 8-bit or 1-bit units. The IICOCKS1 and IICOCKS0 bits are set in combination with the IICCL0.SMC0, IICCL0.CL00, and IICX0.CLX0 bits (see 17.4 (7) I<sup>2</sup>C transfer clock setting method).

Reset sets this register to 00H.

|                  |  |   |                      |           |   |   |          |          |
|------------------|--|---|----------------------|-----------|---|---|----------|----------|
| After reset: 00H |  | R/W                                       | Address: FFFFFFFD90H |           |   |   |          |          |
| IICOCKS          | 7  | 6   | 5                    | 4         | 3 | 2 | 1        | 0        |
|                  | 0  | 0   | 0                    | IICOCKSEN | 0 | 0 | IICOCKS1 | IICOCKS0 |
| IICOCKSEN        | Specification of I <sup>2</sup> C division clock operation |   |                      |           |   |   |          |          |
| 0                | I <sup>2</sup> C division clock operation stop             |   |                      |           |   |   |          |          |
| 1                | I <sup>2</sup> C division clock operation enable           |   |                      |           |   |   |          |          |
| IICOCKS1         | IICOCKS0   | I <sup>2</sup> C division clock selection |                      |           |   |   |          |          |
| 0                | 0  | f <sub>xx</sub> /4                        |                      |           |   |   |          |          |
| 0                | 1  | f <sub>xx</sub> /6                        |                      |           |   |   |          |          |
| 1                | 0  | f <sub>xx</sub> /8                        |                      |           |   |   |          |          |
| 1                | 1  | f <sub>xx</sub> /10                       |                      |           |   |   |          |          |



**(7) I<sup>2</sup>C transfer clock setting method**

The I<sup>2</sup>C transfer clock frequency ( $f_{SCL}$ ) is calculated using the following expression.

$$f_{SCL} = 1/(m \times T + t_R + t_F)$$

$m = 96, 120, 144, 192, 240, 344, 352, 440, 516, 688, 860$  (see **Table 17-2 Selection Clock Setting.**)

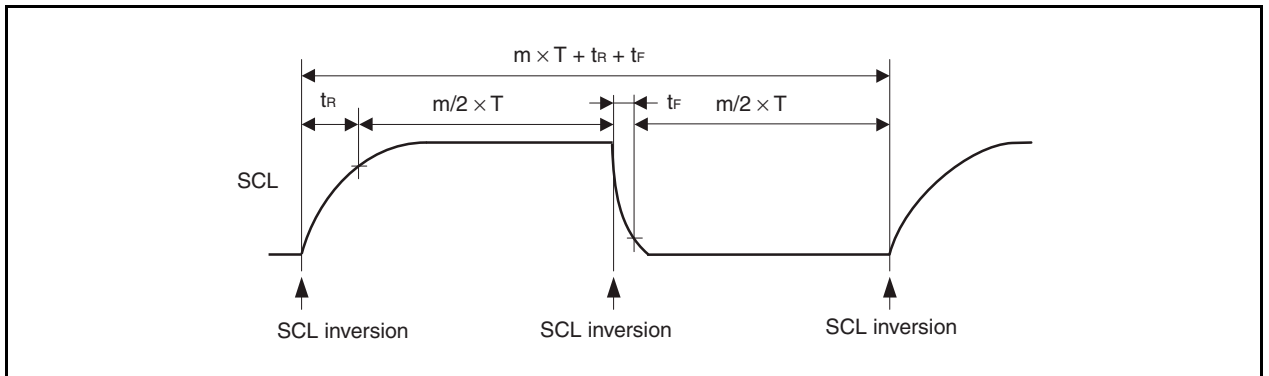
$T: 1/f_{XX}$

$t_R$ : SCL rise time

$t_F$ : SCL fall time

For example, the I<sup>2</sup>C transfer clock frequency ( $f_{SCL}$ ) when  $f_{XX} = 64$  MHz,  $m = 192$ ,  $t_R = 200$  ns, and  $t_F = 50$  ns is calculated using following expression.

$$f_{SCL} = 1/(192 \times 15.6 \text{ ns} + 200 \text{ ns} + 50 \text{ ns}) \cong 308 \text{ kHz}$$



The selection clock is set using a combination of the IICCL0.SMC0, IICCL0.CL00, IICX0.CLX0, IICOCKS.IICOCKS1, and IICOCKS.IICOCKS0 bits.

**Table 17-2. Selection Clock Setting**

| IICX0<br>Bit 0 | IICCL0 |       | Selection Clock                  | Transfer Clock<br>( $f_{XX}/m$ ) | Settable Internal System Clock<br>Frequency ( $f_{XX}$ ) Range | Operation Mode                    |
|----------------|--------|-------|----------------------------------|----------------------------------|--|-----------------------------------|
|                | Bit 3  | Bit 0 |                                  |                                  |  |                                   |
| CLX0           | SMC0   | CL00  |                                  |                                  |  |                                   |
| 0              | 0      | 0     | $f_{XX}/8$ (when IICOCKS = 12H)  | $f_{XX}/352$                     | 32.00 MHz to 33.52 MHz   | Normal mode<br>(SMC0 bit = 0)     |
|                |        |       | $f_{XX}/10$ (when IICOCKS = 13H) | $f_{XX}/440$                     | 32.00 MHz to 41.90 MHz   |                                   |
| 0              | 0      | 1     | $f_{XX}/4$ (when IICOCKS = 10H)  | $f_{XX}/344$                     | 32.00 MHz to 33.52 MHz   |                                   |
|                |        |       | $f_{XX}/6$ (when IICOCKS = 11H)  | $f_{XX}/516$                     | 32.00 MHz to 50.28 MHz   |                                   |
|                |        |       | $f_{XX}/8$ (when IICOCKS = 12H)  | $f_{XX}/688$                     | 33.52 MHz to 64.00 MHz   |                                   |
|                |        |       | $f_{XX}/10$ (when IICOCKS = 13H) | $f_{XX}/860$                     | 41.90 MHz to 64.00 MHz   |                                   |
| 0              | 1      | x     | $f_{XX}/4$ (when IICOCKS = 10H)  | $f_{XX}/96$                      | 32.00 MHz to 33.52 MHz   | High-speed mode<br>(SMC0 bit = 1) |
|                |        |       | $f_{XX}/6$ (when IICOCKS = 11H)  | $f_{XX}/144$                     | 32.00 MHz to 50.28 MHz   |                                   |
|                |        |       | $f_{XX}/8$ (when IICOCKS = 12H)  | $f_{XX}/192$                     | 32.00 MHz to 64.00 MHz   |                                   |
|                |        |       | $f_{XX}/10$ (when IICOCKS = 13H) | $f_{XX}/240$                     | 40.00 MHz to 64.00 MHz   |                                   |
| 1              | 0      | x     | Setting prohibited               |                                  |  |                                   |
| 1              | 1      | x     | $f_{XX}/8$ (when IICOCKS = 12H)  | $f_{XX}/96$                      | 32.00 MHz to 33.52 MHz   | High-speed mode<br>(SMC0 bit = 1) |
|                |        |       | $f_{XX}/10$ (when IICOCKS = 13H) | $f_{XX}/120$                     | 40.00 MHz to 41.90 MHz   |                                   |

**Remark** x: don't care

**(8) IIC shift register 0 (IIC0)**

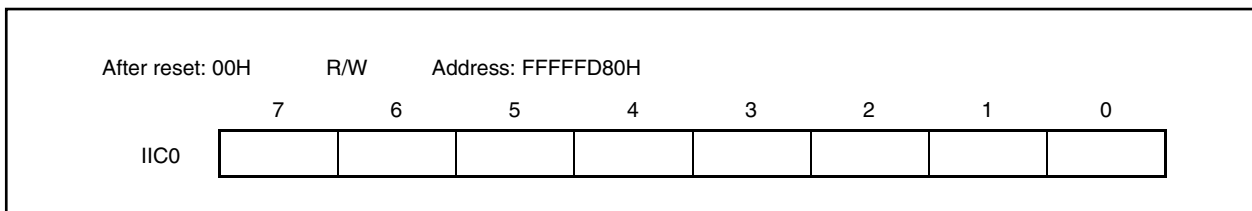
The IIC0 shift register is used for serial transmission/reception (shift operations) that is synchronized with the serial clock.

The IIC0 shift register can be read or written in 8-bit units, but data should not be written to the IIC0 shift register during a data transfer.

Access (read/write) the IIC0 shift register only during the wait period. Accessing this register in communication states other than the wait period is prohibited. However, for the master device, the IIC0 shift register can be written once only after the transmission trigger bit (IICC0.STT0 bit) has been set to 1.

When the IIC0 shift register is written during wait, the wait is cancelled and data transfer is started.

Reset sets this register to 00H.



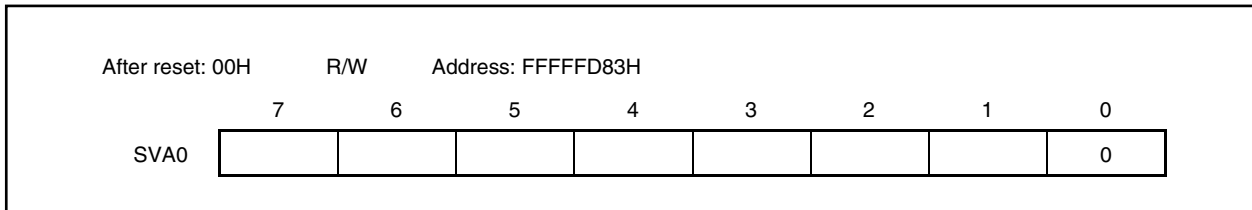
**(9) Slave address register 0 (SVA0)**

The SVA0 register holds the I<sup>2</sup>C bus's slave addresses.

However, rewriting this register is prohibited when the IICS0.STD0 bit = 1 (start condition detection).

The SVA0 register can be read or written in 8-bit units, but bit 0 is fixed to 0.

Reset sets this register to 00H.



**17.5 Functions**

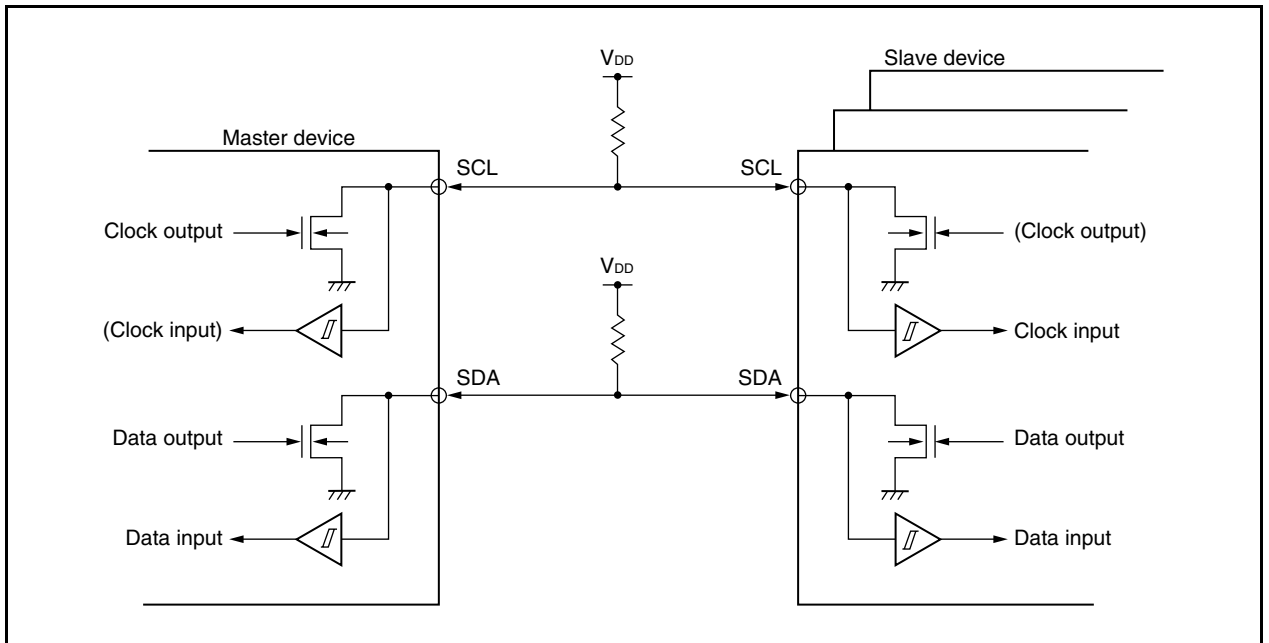
**17.5.1 Pin configuration**

The serial clock pin (SCL) and serial data bus pin (SDA) are configured as follows.

- SCL ..... This pin is used for serial clock input and output.  
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.
- SDA ..... This pin is used for serial data input and output.  
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required.

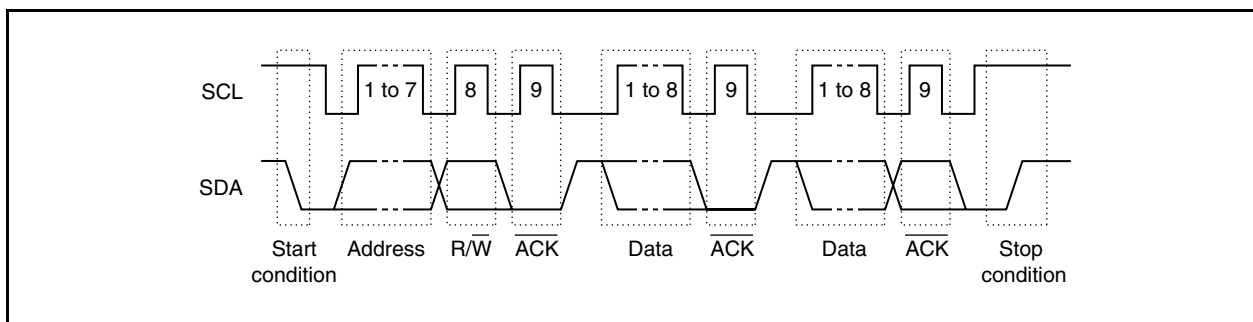
**Figure 17-4. Pin Configuration Diagram**



## 17.6 I<sup>2</sup>C Bus Definitions and Control Methods

The following section describes the I<sup>2</sup>C bus's serial data communication format and the status generated by the I<sup>2</sup>C bus. The transfer timing for the "start condition", "address", "transfer direction specification", "data", and "stop condition" generated via the I<sup>2</sup>C bus's serial data bus is shown below.

Figure 17-5. I<sup>2</sup>C Bus's Serial Data Transfer Timing



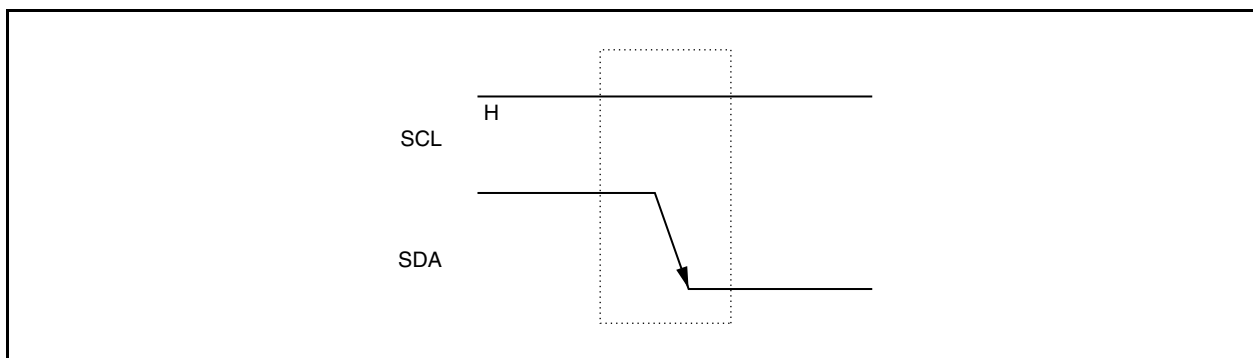
The master device generates the start condition, slave address, and stop condition.  $\overline{\text{ACK}}$  can be generated by either the master or slave device (normally, it is generated by the device that receives 8-bit data).

The serial clock (SCL) is continuously output by the master device. However, in the slave device, the SCL's low-level period can be extended and a wait can be inserted.

### 17.6.1 Start condition

A start condition is met when the SCL pin is at high level and the SDA pin changes from high level to low level. The start conditions for the SCL pin and SDA pin are generated when the master device starts a serial transfer to the slave device. Start conditions can be detected when the device is used as a slave.

Figure 17-6. Start Conditions



A start condition is generated when the IICCS0.STT0 bit is set to 1 after a stop condition has been detected (IICCS0.SP0 bit = 1). When a start condition is detected, IICCS0.STD0 bit is set to 1.

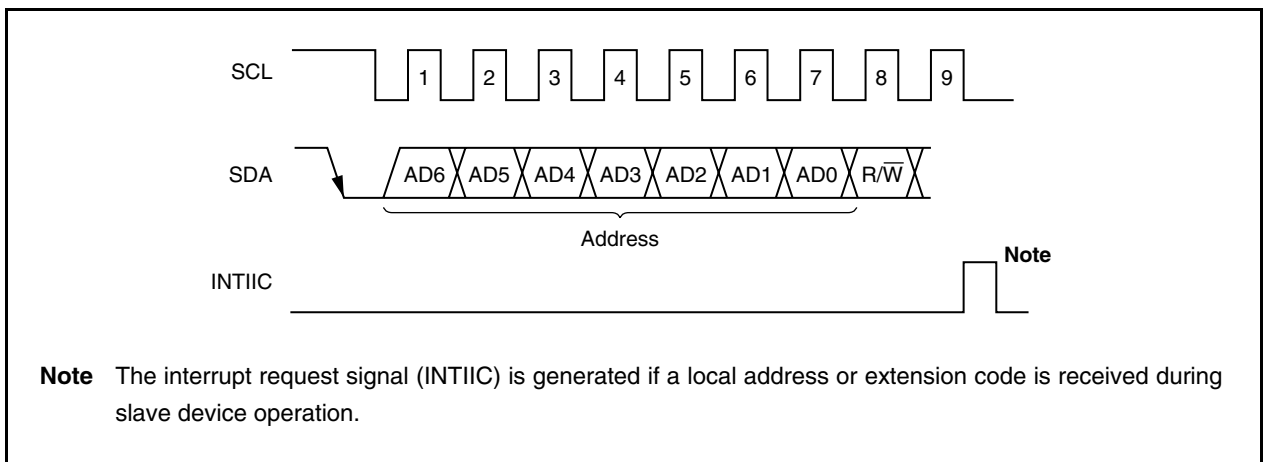
**17.6.2 Addresses**

The 7 bits of data that follow the start condition are defined as an address.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in the SVA0 register. If the address data matches the SVA0 values, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

**Figure 17-7. Address**



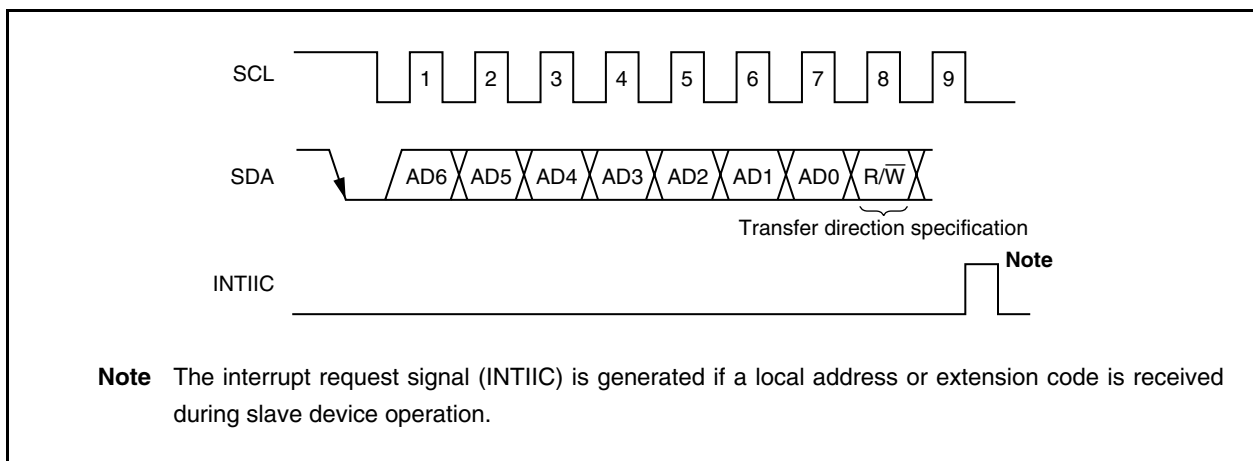
The slave address and the eighth bit, which specifies the transfer direction as described in **17.6.3 Transfer direction specification** below, are together written to the IIC0 register and are then output. Received addresses are written to the IIC0 register.

The slave address is assigned to the higher 7 bits of the IIC0 register.

### 17.6.3 Transfer direction specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction. When this transfer direction specification bit has a value of 0, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of 1, it indicates that the master device is receiving data from a slave device.

**Figure 17-8. Transfer Direction Specification**



### 17.6.4 $\overline{\text{ACK}}$

$\overline{\text{ACK}}$  is used to confirm the serial data status of the transmitting and receiving devices.

The receiving device returns  $\overline{\text{ACK}}$  for every 8 bits of data it receives.

The transmitting device normally receives  $\overline{\text{ACK}}$  after transmitting 8 bits of data. When  $\overline{\text{ACK}}$  is returned from the receiving device, the reception is judged as normal and processing continues. The detection of  $\overline{\text{ACK}}$  is confirmed with the IIC0.ACKD0 bit.

When the master device is the receiving device, after receiving the final data, it does not return  $\overline{\text{ACK}}$  and generates the stop condition. When the slave device is the receiving device and does not return  $\overline{\text{ACK}}$ , the master device generates either a stop condition or a restart condition, and then stops the current transmission. Failure to return  $\overline{\text{ACK}}$  may be caused by the following factors.

- (a) Reception was not performed normally.
- (b) The final data was received.
- (c) The receiving device (slave) does not exist for the specified address.

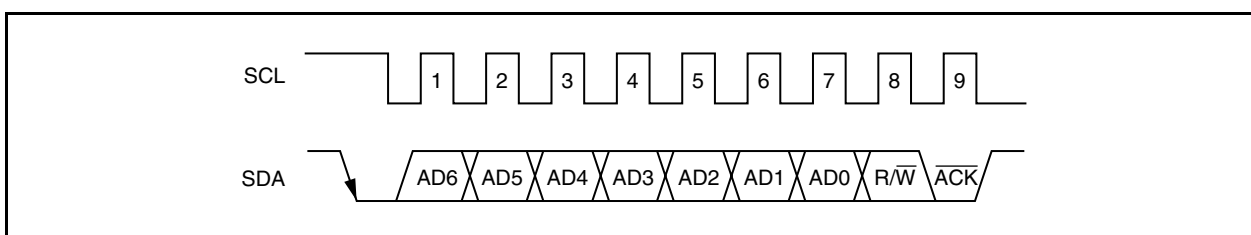
When the receiving device sets the SDA line to low level during the ninth clock,  $\overline{\text{ACK}}$  is generated (normal reception).

When the IICC0.ACKE0 bit is set to 1, automatic  $\overline{\text{ACK}}$  generation is enabled. Transmission of the eighth bit following the 7 address data bits causes the IIC0.TR0 bit to be set. Normally, set the ACKE0 bit to 1 for reception (TR0 bit = 0).

When the slave device is receiving (when TR0 bit = 0), if the slave device cannot receive data or does not need to receive any more data, clear the ACKE0 bit to 0 to indicate to the master that no more data can be received.

Similarly, when the master device is receiving (when TR0 bit = 0) and the subsequent data is not needed, clear the ACKE0 bit to 0 to prevent  $\overline{\text{ACK}}$  from being generated. This notifies the slave device (transmitting device) of the end of the data transmission (transmission stopped).

**Figure 17-9.  $\overline{\text{ACK}}$**



When the local address is received,  $\overline{\text{ACK}}$  is automatically generated regardless of the value of the ACKE0 bit. No  $\overline{\text{ACK}}$  is generated if the received address is not a local address (NACK).

When receiving the extension code, set the ACKE0 bit to 1 in advance to generate  $\overline{\text{ACK}}$ .

The  $\overline{\text{ACK}}$  generation method during data reception is based on the wait timing setting, as described by the following.

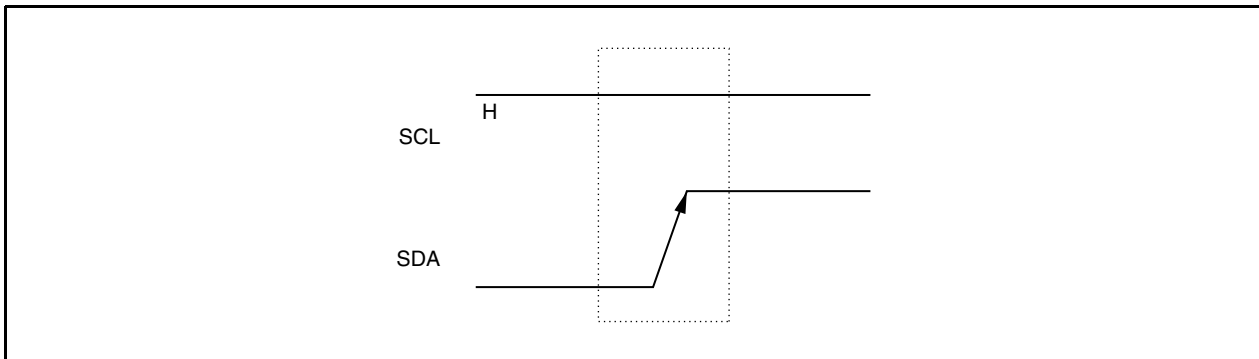
- When 8-clock wait is selected (IICC0.WTIM0 bit = 0):  
 $\overline{\text{ACK}}$  is generated at the falling edge of the SCL pin's eighth clock if the ACKE0 bit is set to 1 before the wait state cancellation.
- When 9-clock wait is selected (IICC0.WTIM0 bit = 1):  
 $\overline{\text{ACK}}$  is generated if the ACKE0 bit is set to 1 in advance.

### 17.6.5 Stop condition

When the SCL pin is at high level, changing the SDA pin from low level to high level generates a stop condition.

A stop condition is generated when serial transfer from the master device to the slave device has been completed. Stop conditions can be detected when the device is used as a slave.

**Figure 17-10. Stop Condition**



A stop condition is generated when the IICC0.SPT0 bit is set to 1. When the stop condition is detected, the IICS0.SP0 bit is set to 1 and the interrupt request signal (INTIIC) is generated when the IICC0.SPIE0 bit is set to 1.



**17.6.6 Wait state**

The wait state is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a wait state).

Setting the SCL pin to low level notifies the communication partner of the wait status. When wait status has been canceled for both the master and slave devices, the next data transfer can begin.

**Figure 17-11. Wait State (1/2)**

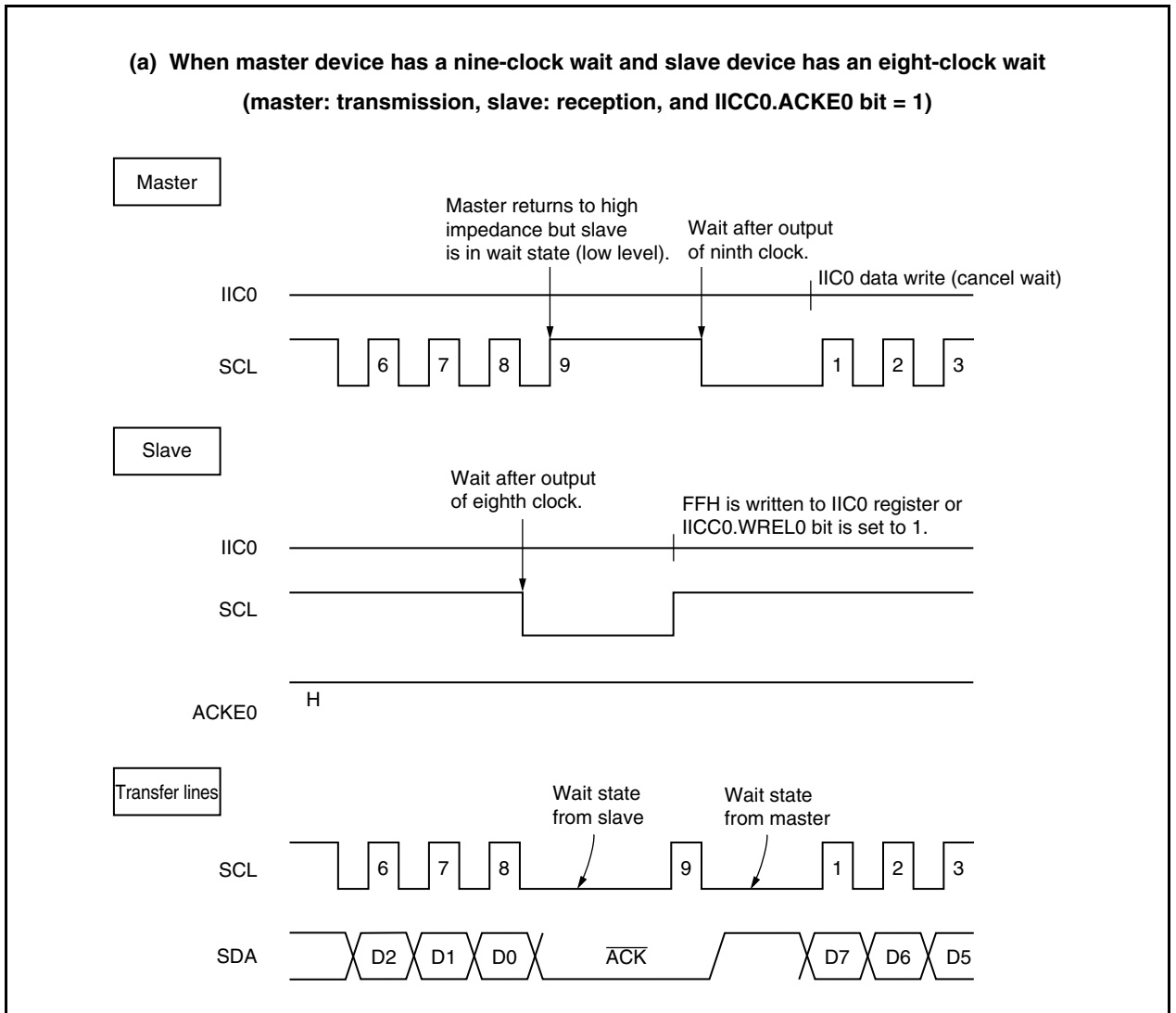
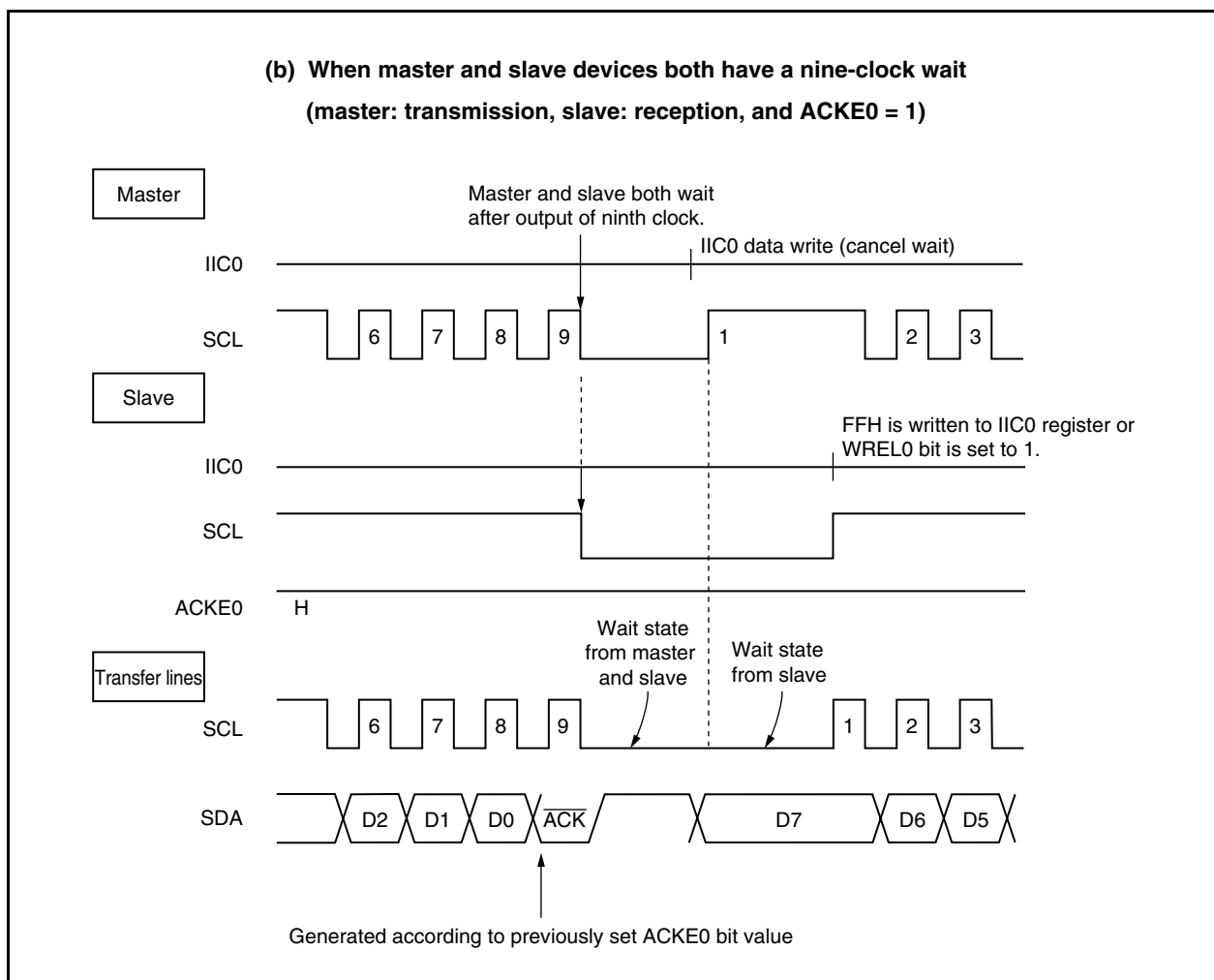


Figure 17-11. Wait State (2/2)



A wait state is automatically generated after a start condition is generated. Moreover, a wait state is automatically generated depending on the setting of the IICC0.WTIM0 bit.

Normally, when the IICC0.WREL0 bit is set to 1 or when FFH is written to the IIC0 register, the wait status is canceled and the transmitting side writes data to the IIC0 register to cancel the wait status.

The master device can also cancel the wait status via either of the following methods.

- By setting the IICC0.STT0 bit to 1
- By setting the IICC0.SPT0 bit to 1

### 17.6.7 Wait state cancellation method

In the case of I<sup>2</sup>C, wait state can be canceled normally in the following ways.

- By writing data to the IIC0 register
- By setting the IICC0.WRELO bit to 1 (wait state cancellation)
- By setting the IICC0.STT0 bit to 1 (start condition generation)<sup>Note</sup>
- By setting the IICC0.SPT0 bit to 1 (stop condition generation)<sup>Note</sup>

**Note** Master only

If any of these wait state cancellation actions is performed, I<sup>2</sup>C will cancel wait state and restart communication.

When canceling wait state and sending data (including address), write data to the IIC0 register.

To receive data after canceling wait state, or to end data transmission, set the WRELO bit to 1.

To generate a restart condition after canceling wait state, set the STT0 bit to 1.

To generate a stop condition after canceling wait state, set the SPT0 bit to 1.

Execute cancellation only once for each wait state.

For example, if data is written to the IIC0 register following wait state cancellation by setting the WRELO bit to 1, conflict between the SDA line change timing and IIC0 register write timing may result in the data output to the SDA line may be incorrect.

Even in other operations, if communication is stopped halfway, clearing the IICC0.IICE0 bit to 0 will stop communication, enabling wait state to be cancelled.

If the I<sup>2</sup>C bus dead-locks due to noise, etc., setting the IICC0.LRELO bit to 1 causes the communication operation to be exited, enabling wait state to be cancelled.

### 17.7 I<sup>2</sup>C Interrupt Request Signals (INTIIC)

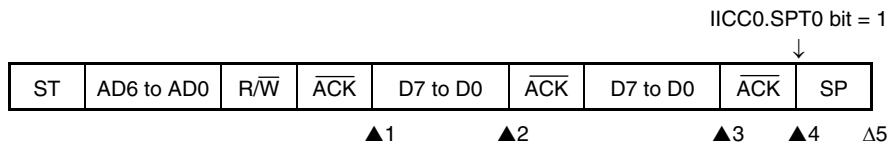
The following shows the value of the IICS0 register at the INTIIC interrupt request signal generation timing and at the INTIIC signal timing.

|               |                    |                                  |
|---------------|--------------------|----------------------------------|
| <b>Remark</b> | ST:                | Start condition                  |
|               | AD6 to AD0:        | Address                          |
|               | R/W:               | Transfer direction specification |
|               | $\overline{ACK}$ : | Acknowledge                      |
|               | D7 to D0:          | Data                             |
|               | SP:                | Stop condition                   |

17.7.1 Master device operation

(1) Start ~ Address ~ Data ~ Data ~ Stop (normal transmission/reception)

<1> When IICC0.WTIM0 bit = 0

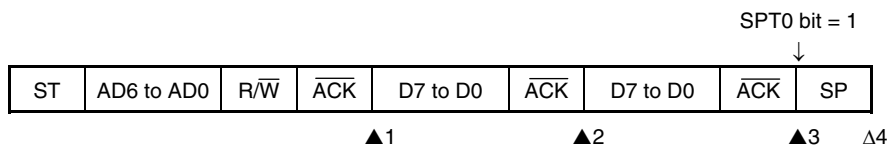


- ▲1: IICS0 register = 1000X110B
- ▲2: IICS0 register = 1000X000B
- ▲3: IICS0 register = 1000X000B (WTIM0 bit = 1<sup>Note</sup>)
- ▲4: IICS0 register = 1000XX00B
- Δ 5: IICS0 register = 00000001B

**Note** To generate a stop condition, set the WTIM0 bit to 1 and change the timing of the generation of the interrupt request signal (INTIIC).

**Remark** ▲: Always generated  
 Δ: Generated only when IICC0.SPIE0 bit = 1  
 X: don't care

<2> When WTIM0 bit = 1

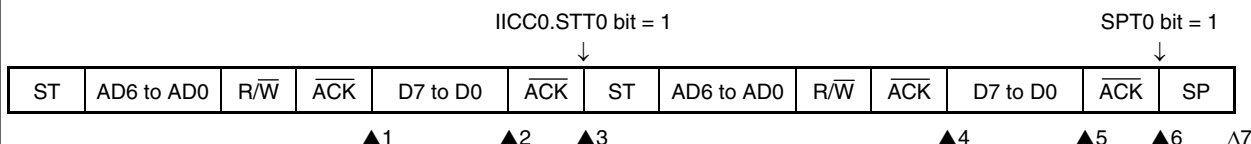


- ▲1: IICS0 register = 1000X110B
- ▲2: IICS0 register = 1000X100B
- ▲3: IICS0 register = 1000XX00B
- Δ 4: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

(2) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

<1> When WTIM0 bit = 0

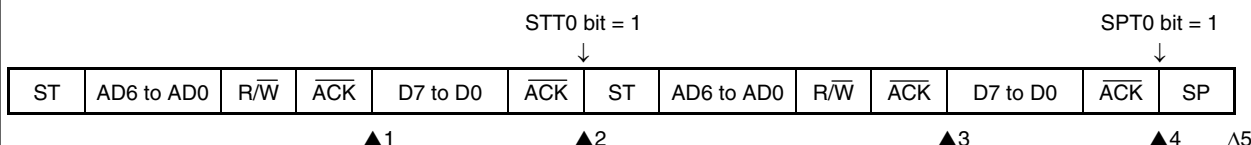


- ▲1: IICS0 register = 1000X110B
- ▲2: IICS0 register = 1000X000B (WTIM0 bit = 1<sup>Note 1</sup>)
- ▲3: IICS0 register = 1000XX00B (WTIM0 bit = 0<sup>Note 2</sup>)
- ▲4: IICS0 register = 1000X110B
- ▲5: IICS0 register = 1000X000B (WTIM0 bit = 1<sup>Note 3</sup>)
- ▲6: IICS0 register = 1000XX00B
- Δ 7: IICS0 register = 00000001B

- Notes**
1. To generate a start condition, set the WTIM0 bit to 1 and change the timing of the generation of the interrupt request signal (INTIIC).
  2. Clear the WTIM0 bit to 0 to make the settings original.
  3. To generate a stop condition, set the WTIM0 bit to 1 and change the timing of the generation of the interrupt request signal (INTIIC).

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

<2> When WTIM0 bit = 1

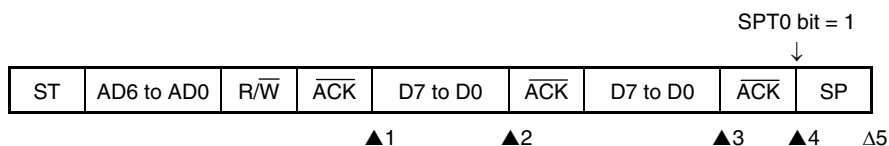


- ▲1: IICS0 register = 1000X110B
- ▲2: IICS0 register = 1000XX00B
- ▲3: IICS0 register = 1000X110B
- ▲4: IICS0 register = 1000XX00B
- Δ 5: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

(3) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

<1> When WTIM0 bit = 0

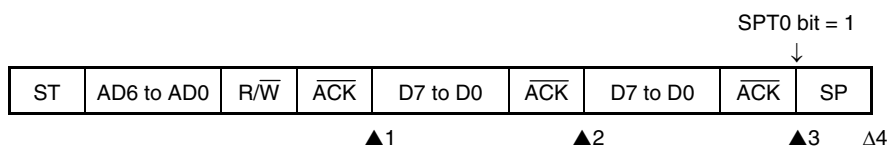


- ▲1: IICS0 register = 1010X110B
- ▲2: IICS0 register = 1010X000B
- ▲3: IICS0 register = 1010X000B (WTIM0 bit = 1<sup>Note</sup>)
- ▲4: IICS0 register = 1010XX00B
- Δ 5: IICS0 register = 00000001B

**Note** To generate a stop condition, set the WTIM0 bit to 1 and change the timing of the generation of the interrupt request signal (INTIIC).

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

<2> When WTIM0 bit = 1



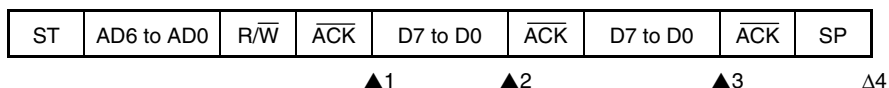
- ▲1: IICS0 register = 1010X110B
- ▲2: IICS0 register = 1010X100B
- ▲3: IICS0 register = 1010XX00B
- Δ 4: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

17.7.2 Slave device operation (when receiving slave address data (address match))

(1) Start ~ Address ~ Data ~ Data ~ Stop

<1> When IICC0.WTIM0 bit = 0



▲1: IICS0 register = 0001X110B

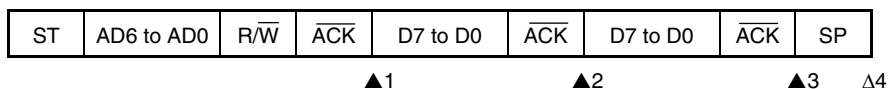
▲2: IICS0 register = 0001X000B

▲3: IICS0 register = 0001X000B

Δ 4: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when IICC0.SPIE0 bit = 1  
 X: don't care

<2> When WTIM0 bit = 1



▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001X100B

▲3: IICS0 register = 0001XX00B

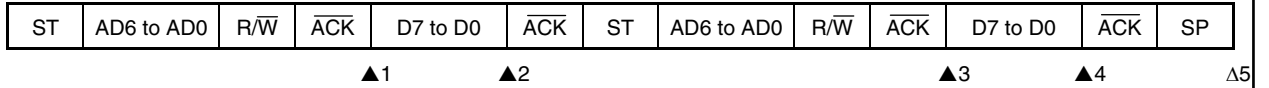
Δ 4: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care



(2) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

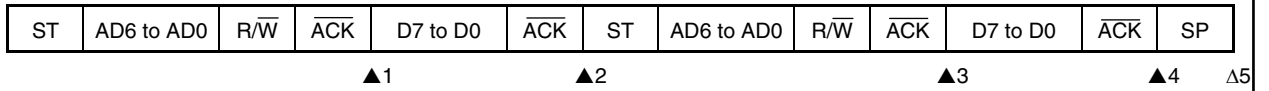
<1> When WTIM0 bit = 0 (after restart, address match)



- ▲1: IICS0 register = 0001X110B
- ▲2: IICS0 register = 0001X000B
- ▲3: IICS0 register = 0001X110B
- ▲4: IICS0 register = 0001X000B
- Δ 5: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

<2> When WTIM0 bit = 1 (after restart, address match)

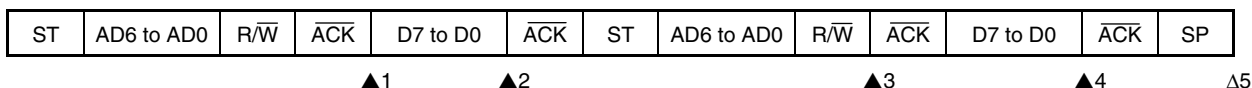


- ▲1: IICS0 register = 0001X110B
- ▲2: IICS0 register = 0001XX00B
- ▲3: IICS0 register = 0001X110B
- ▲4: IICS0 register = 0001XX00B
- Δ 5: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

(3) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

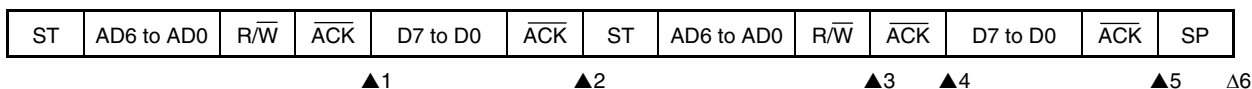
<1> When WTIM0 bit = 0 (after restart, address mismatch (extension code))



- ▲1: IICS0 register = 0001X110B
- ▲2: IICS0 register = 0001X000B
- ▲3: IICS0 register = 0010X010B
- ▲4: IICS0 register = 0010X000B
- Δ 5: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

<2> When WTIM0 bit = 1 (after restart, address mismatch (extension code))

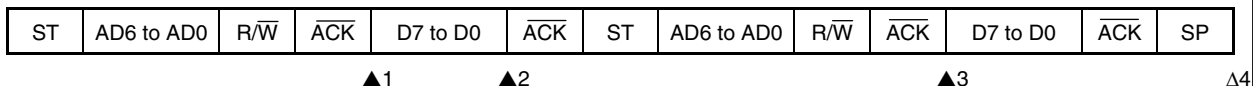


- ▲1: IICS0 register = 0001X110B
- ▲2: IICS0 register = 0001XX00B
- ▲3: IICS0 register = 0010X010B
- ▲4: IICS0 register = 0010X110B
- ▲5: IICS0 register = 0010XX00B
- Δ 6: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

(4) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

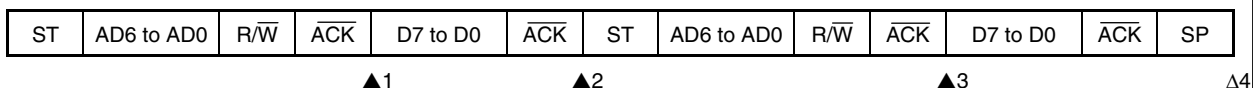
<1> When WTIM0 bit = 0 (after restart, address mismatch (= not extension code))



- ▲1: IICS0 register = 0001X110B
- ▲2: IICS0 register = 0001X000B
- ▲3: IICS0 register = 00000110B
- Δ 4: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

<2> When WTIM0 bit = 1 (after restart, address mismatch (= not extension code))



- ▲1: IICS0 register = 0001X110B
- ▲2: IICS0 register = 0001XX00B
- ▲3: IICS0 register = 00000110B
- Δ 4: IICS0 register = 00000001B

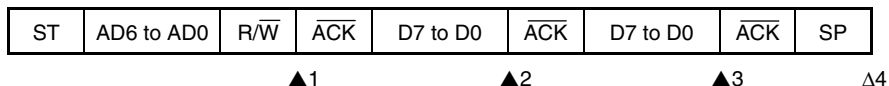
**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

17.7.3 Slave device operation (when receiving extension code)

Always under communication when receiving the extension code.

(1) Start ~ Code ~ Data ~ Data ~ Stop

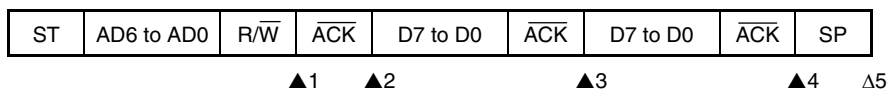
<1> When IICC0.WTIM0 bit = 0



- ▲1: IICS0 register = 0010X010B
- ▲2: IICS0 register = 0010X000B
- ▲3: IICS0 register = 0010X000B
- Δ 4: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when IICC0.SPIE0 bit = 1  
 X: don't care

<2> When WTIM0 bit = 1

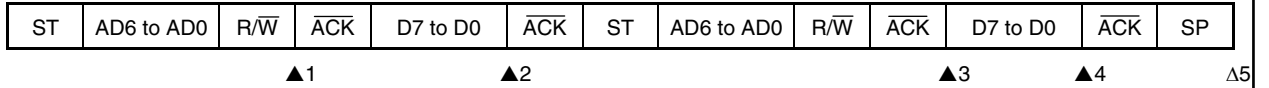


- ▲1: IICS0 register = 0010X010B
- ▲2: IICS0 register = 0010X110B
- ▲3: IICS0 register = 0010X100B
- ▲4: IICS0 register = 0010XX00B
- Δ 5: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

(2) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

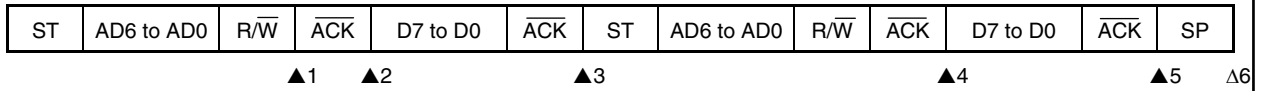
<1> When WTIM0 bit = 0 (after restart, address match)



- ▲1: IICS0 register = 0010X010B
- ▲2: IICS0 register = 0010X000B
- ▲3: IICS0 register = 0001X110B
- ▲4: IICS0 register = 0001X000B
- Δ 5: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

<2> When WTIM0 bit = 1 (after restart, address match)

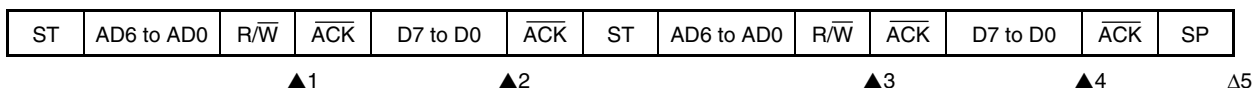


- ▲1: IICS0 register = 0010X010B
- ▲2: IICS0 register = 0010X110B
- ▲3: IICS0 register = 0010XX00B
- ▲4: IICS0 register = 0001X110B
- ▲5: IICS0 register = 0001XX00B
- Δ 6: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

(3) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

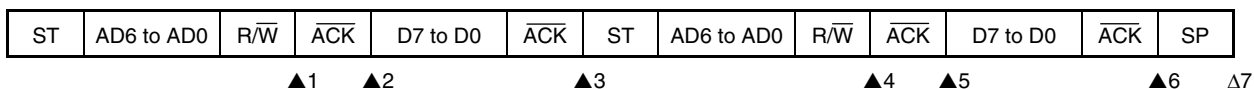
<1> When WTIM0 bit = 0 (after restart, extension code reception)



- ▲1: IICS0 register = 0010X010B
- ▲2: IICS0 register = 0010X000B
- ▲3: IICS0 register = 0010X010B
- ▲4: IICS0 register = 0010X000B
- Δ 5: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

<2> When WTIM0 bit = 1 (after restart, extension code reception)

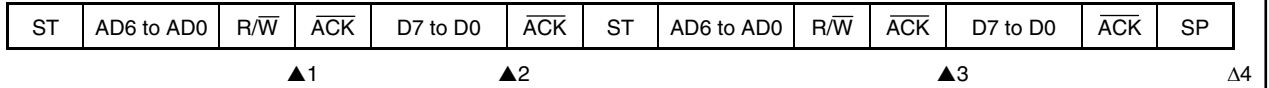


- ▲1: IICS0 register = 0010X010B
- ▲2: IICS0 register = 0010X110B
- ▲3: IICS0 register = 0010XX00B
- ▲4: IICS0 register = 0010X010B
- ▲5: IICS0 register = 0010X110B
- ▲6: IICS0 register = 0010XX00B
- Δ 7: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

(4) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

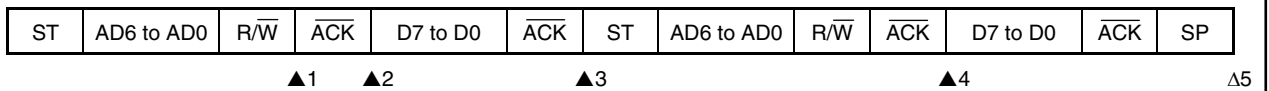
<1> When WTIM0 bit = 0 (after restart, address mismatch (= not extension code))



- ▲1: IICS0 register = 0010X010B
- ▲2: IICS0 register = 0010X000B
- ▲3: IICS0 register = 00000110B
- Δ 4: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

<2> When WTIM0 bit = 1 (after restart, address mismatch (= not extension code))

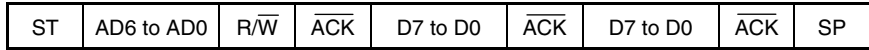


- ▲1: IICS0 register = 0010X010B
- ▲2: IICS0 register = 0010X110B
- ▲3: IICS0 register = 0010XX00B
- ▲4: IICS0 register = 00000110B
- Δ 5: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

17.7.4 Operation without communication

(1) Start ~ Code ~ Data ~ Data ~ Stop



Δ1

Δ 1: IICS0 register = 00000001B

**Remark** Δ: Generated only when IICC0.SPIE0 bit = 1

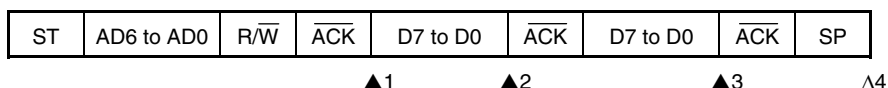


**17.7.5 Arbitration loss operation (operation as slave after arbitration loss)**

When used as master in the multi-master system, check the arbitration result by reading the IICS0.MSTS0 bit for checking arbitration result by each INTIIC interrupt occurrence.

**(1) When arbitration loss occurs during transmission of slave address data**

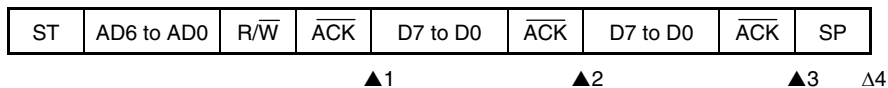
**<1> When IICC0.WTIM0 bit = 0**



- ▲1: IICS0 register = 0101X110B
- ▲2: IICS0 register = 0001X000B
- ▲3: IICS0 register = 0001X000B
- Δ 4: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when IICC0.SPIE0 bit = 1  
 X: don't care

**<2> When WTIM0 bit = 1**

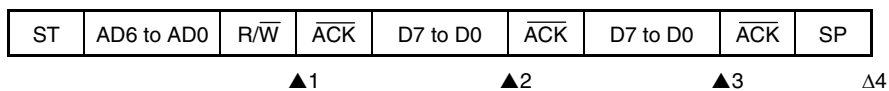


- ▲1: IICS0 register = 0101X110B
- ▲2: IICS0 register = 0001X100B
- ▲3: IICS0 register = 0001XX00B
- Δ 4: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

(2) When arbitration loss occurs during transmission of extension code

<1> When WTIM0 bit = 0



▲1: IICS0 register = 0110X010B

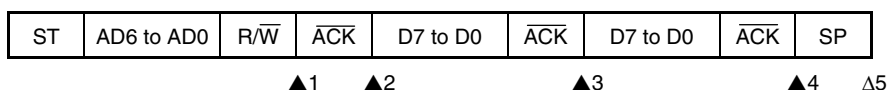
▲2: IICS0 register = 0010X000B

▲3: IICS0 register = 0010X000B

Δ 4: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

<2> When WTIM0 bit = 1



▲1: IICS0 register = 0110X010B

▲2: IICS0 register = 0010X110B

▲3: IICS0 register = 0010X100B

▲4: IICS0 register = 0010XX00B

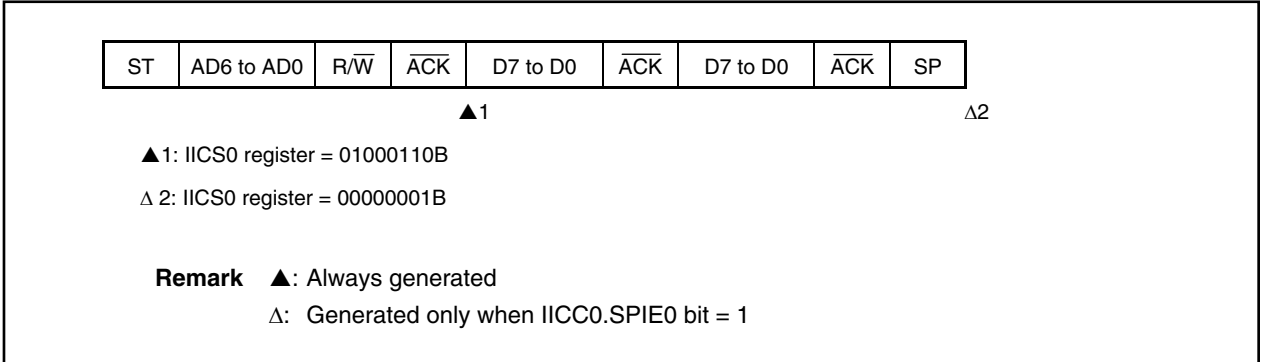
Δ 5: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

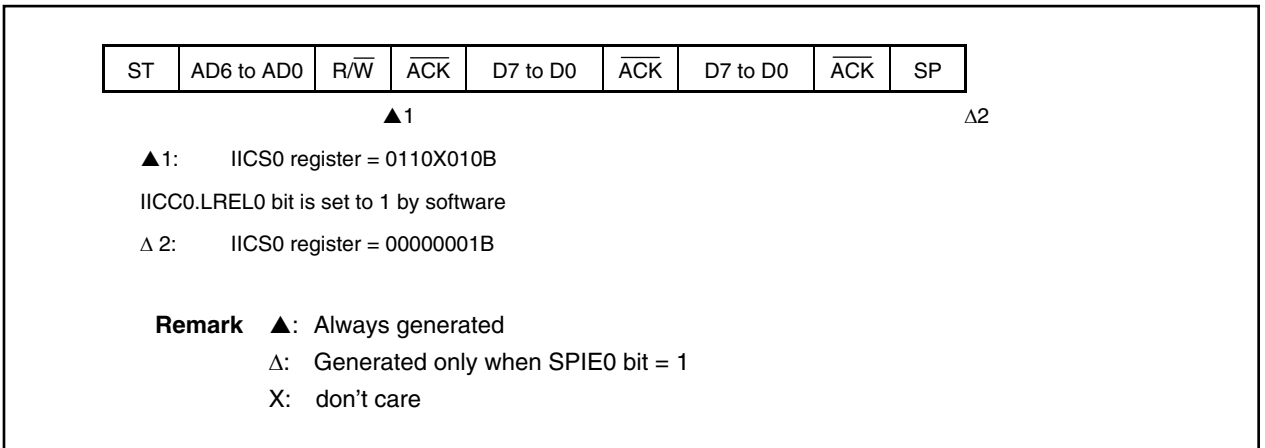
**17.7.6 Operation when arbitration loss occurs (no communication after arbitration loss)**

When used as master in the multi-master system, check the arbitration result by reading the IICS0.MSTS0 bit for checking arbitration result by each INTIIC interrupt occurrence.

**(1) When arbitration loss occurs during transmission of slave address data**

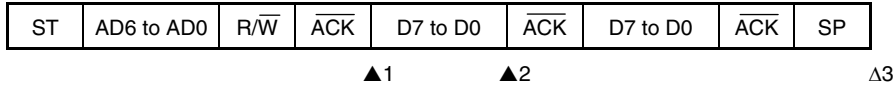


**(2) When arbitration loss occurs during transmission of extension code**



(3) When arbitration loss occurs during data transfer

<1> When IICC0.WTIM0 bit = 0



▲1: IICS0 register = 10001110B

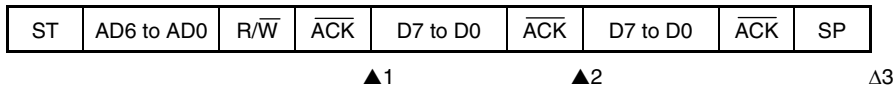
▲2: IICS0 register = 01000000B

Δ 3: IICS0 register = 00000001B

**Remark** ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

<2> When WTIM0 bit = 1



▲1: IICS0 register = 10001110B

▲2: IICS0 register = 01000100B

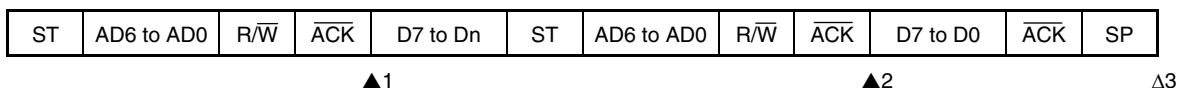
Δ 3: IICS0 register = 00000001B

**Remark** ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

(4) When arbitration loss occurs due to restart condition during data transfer

<1> Not extension code (Example: Address mismatch)



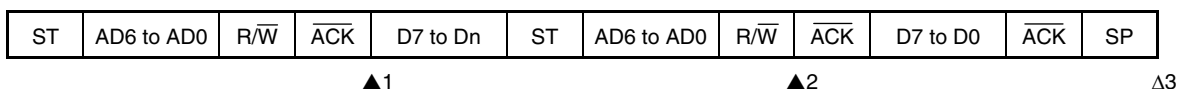
▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 01000110B

Δ 3: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care  
 Dn = D6 to D0

<2> Extension code



▲1: IICS0 register = 1000X110B

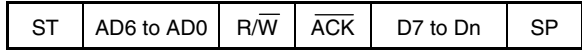
▲2: IICS0 register = 0110X010B

IICC0.LRELO bit is set to 1 by software

Δ 3: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care  
 Dn = D6 to D0

(5) When arbitration loss occurs due to stop condition during data transfer



▲1

Δ2

▲1: IICSO register = 1000X110B

Δ 2: IICSO register = 01000001B

**Remark** ▲: Always generated

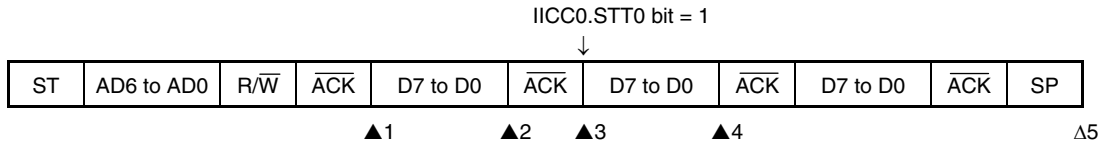
Δ: Generated only when SPIE0 bit = 1

X: don't care

Dn = D6 to D0

(6) When arbitration loss occurs due to low level of SDA<sub>n</sub> pin when attempting to generate a restart condition

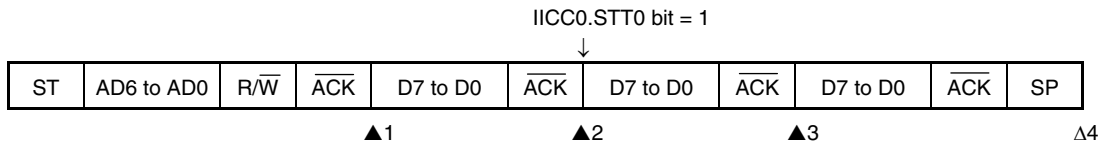
<1> When WTIM0 bit = 0



- ▲1: IICS0 register = 1000X110B
- ▲2: IICS0 register = 1000X000B (WTIM0 bit = 1)
- ▲3: IICS0 register = 1000X100B (WTIM0 bit = 0)
- ▲4: IICS0 register = 01000000B
- Δ 5: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

<2> When WTIM0 bit = 1

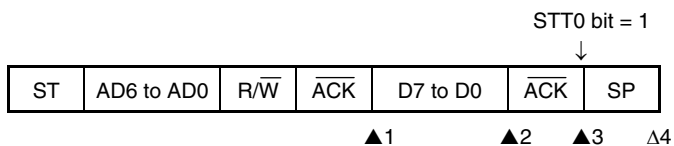


- ▲1: IICS0 register = 1000X110B
- ▲2: IICS0 register = 1000X100B
- ▲3: IICS0 register = 01000100B
- Δ 4: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

(7) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

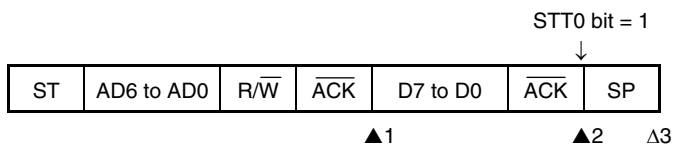
<1> When WTIM0 bit = 0



- ▲1: IICS0 register = 1000X110B
- ▲2: IICS0 register = 1000X000B (WTIM0 bit = 1)
- ▲3: IICS0 register = 1000XX00B
- Δ 4: IICS0 register = 01000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

<2> When WTIM0 bit = 1



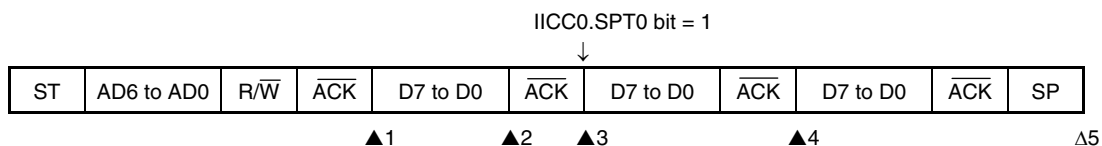
- ▲1: IICS0 register = 1000X110B
- ▲2: IICS0 register = 1000XX00B
- Δ 3: IICS0 register = 01000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care



**(8) When arbitration loss occurs due to low level of SDAn pin when attempting to generate a stop condition**

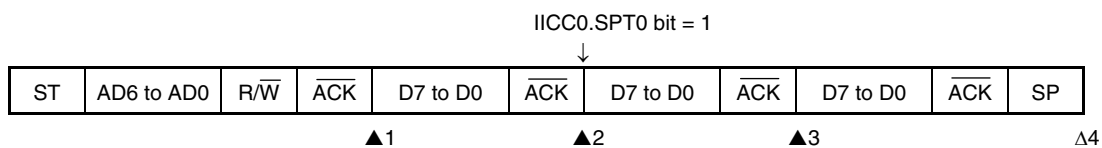
**<1> When WTIM0 bit = 0**



- ▲1: IICS0 register = 1000X110B
- ▲2: IICS0 register = 1000X000B (WTIM0 bit = 1)
- ▲3: IICS0 register = 1000X100B (WTIM0 bit = 0)
- ▲4: IICS0 register = 01000100B
- Δ 5: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

**<2> When WTIM0 bit = 1**



- ▲1: IICS0 register = 1000X110B
- ▲2: IICS0 register = 1000X100B
- ▲3: IICS0 register = 01000100B
- Δ 4: IICS0 register = 00000001B

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 bit = 1  
 X: don't care

## 17.8 Interrupt Request Signal (INTIIC) Generation Timing and Wait Control

The setting of the IICC0.WTIM0 bit determines the timing by which the INTIIC signal is generated and the corresponding wait control, as shown below.

**Table 17-3. INTIIC Signal Generation Timing and Wait Control**

| WTIM0 Bit | During Slave Device Operation |                     |                     | During Master Device Operation |                |                   |
|-----------|-------------------------------|---------------------|---------------------|--------------------------------|----------------|-------------------|
|           | Address                       | Data Reception      | Data Transmission   | Address                        | Data Reception | Data Transmission |
| 0         | g <sup>Notes 1, 2</sup>       | g <sup>Note 2</sup> | g <sup>Note 2</sup> | 9                              | 8              | 8                 |
| 1         | g <sup>Notes 1, 2</sup>       | g <sup>Note 2</sup> | g <sup>Note 2</sup> | 9                              | 9              | 9                 |

- Notes 1.** The slave device's INTIIC signal and wait period occurs at the falling edge of the ninth clock only when there is a match with the address set to the SVA0 register.  
At this point,  $\overline{ACK}$  is generated regardless of the value set to the IICC0.ACKE0 bit. For a slave device that has received an extension code, the INTIIC signal occurs at the falling edge of the eighth clock. When the address does not match after restart, the INTIIC signal is generated at the falling edge of the ninth clock, but no wait occurs.
- 2.** If the received address does not match the contents of the SVA0 register and extension codes have not been received, neither the INTIIC signal nor a wait occurs.

**Remark** The numbers in the table indicate the number of the serial clock's clock signals. Interrupt requests and wait control are both synchronized with the falling edge of these clock signals.

### (1) During address transmission/reception

- Slave device operation: Interrupt and wait timing are determined depending on the conditions in Notes 1 and 2 above regardless of the WTIM0 bit.
- Master device operation: Interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIM0 bit.

### (2) During data reception

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

### (3) During data transmission

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

### (4) Wait cancellation method

The four wait cancellation methods are as follows.

- By writing data to the IIC0 register
- By setting the IICC0.WREL0 bit (canceling wait state)
- By setting the IICC0.STT0 bit (generating start condition)<sup>Note</sup>
- By setting the IICC0.SPT0 bit (generating stop condition)<sup>Note</sup>

**Note** Master only

When an 8-clock wait has been selected (WTIM0 bit = 0), whether or not  $\overline{ACK}$  has been generated must be determined prior to wait cancellation.

### (5) Stop condition detection

The INTIIC signal is generated when a stop condition is detected.

## 17.9 Address Match Detection Method

When in I<sup>2</sup>C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

Address match detection is performed automatically by hardware. An INTIIC interrupt request signal occurs when a local address has been set to the SVA0 register and when the address set to the SVA0 register matches the slave address sent by the master device, or when an extension code has been received.

## 17.10 Error Detection

In I<sup>2</sup>C bus mode, the status of the serial data bus (SDA) during data transmission is captured by the IIC0 register of the transmitting device, so the IIC0 register data prior to transmission can be compared with the transmitted IIC0 register data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

## 17.11 Extension Code

- (1) When the higher 4 bits of the receive address are either 0000 or 1111, the extension code flag (EXC0) is set for extension code reception and an interrupt request signal (INTIIC) is issued at the falling edge of the eighth clock.

The local address stored in the SVA0 register is not affected.

- (2) If 11110xx0 is set to the SVA0 register by a 10-bit address transfer and 11110xx0 is transferred from the master device, the results are as follows. Note that the INTIIC signal occurs at the falling edge of the eighth clock.

- Higher 4 bits of data match: IICS0.EXC0 bit = 1
- 7 bits of data match: IICS0.COI0 bit = 1

- (3) Since the processing after the INTIIC signal occurs differs according to the data that follows the extension code, such processing is performed by software. The slave that has received an extension code is always under communication, even if the addresses mismatch.

For example, when operation as a slave is not desired after the extension code is received, set the IICC0.LREL0 bit to 1 and the CPU will enter the next communication wait state.

**Table 17-4. Bit Definitions for Major Extension Code**

| Slave Address | R/W Bit | Description   |
|---------------|---------|---|
| 0000 000      | 0       | General call address  |
| 1111 0xx      | 0       | 10-bit slave address specification (upon address authentication)                      |
| 1111 0xx      | 1       | 10-bit slave address specification (upon read command issuance after address matches) |

**Remark** For the extension codes other than above, see the I<sup>2</sup>C bus specifications issued by NXP Semiconductors.

### 17.12 Arbitration

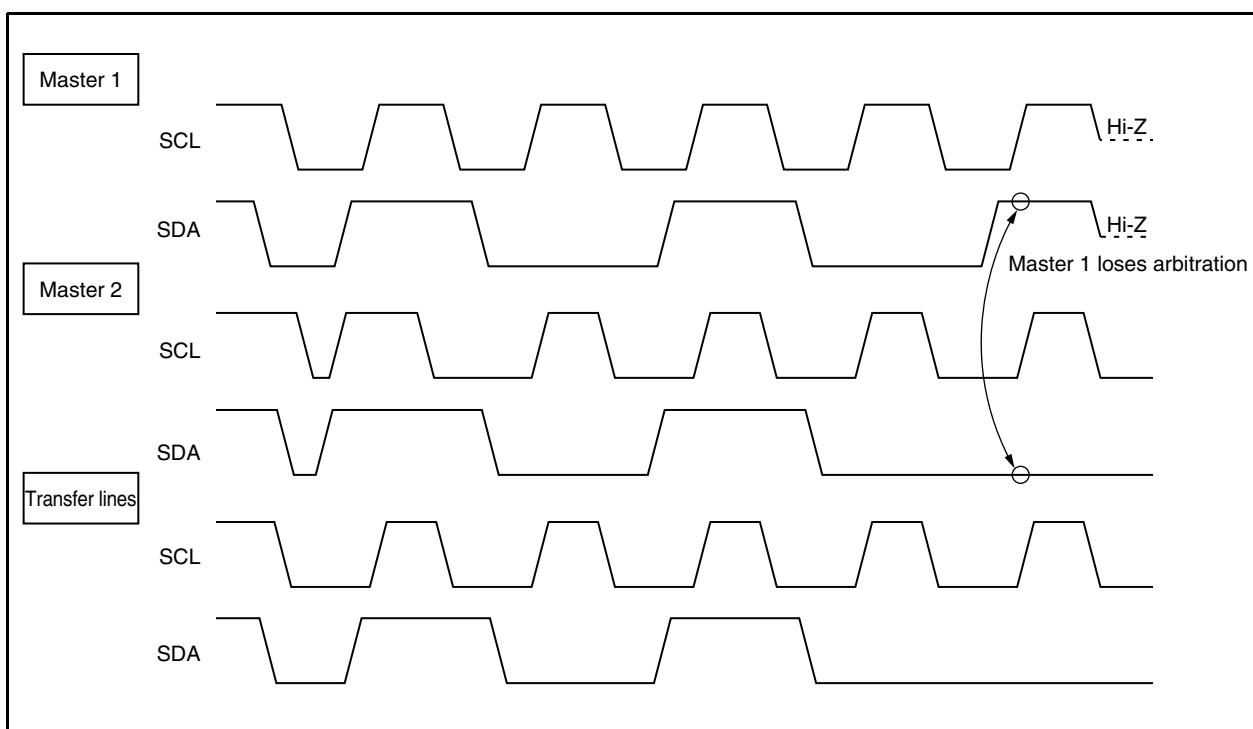
When several master devices simultaneously generate a start condition (when the IICC0.STT0 bit is set to 1 before the IICS0.STD0 bit is set to 1), communication among the master devices is performed as the number of clocks is adjusted until the data differs. This kind of operation is called arbitration.

When one of the master devices loses in arbitration, an arbitration loss flag (IICS0.ALD0 bit) is set (1) via the timing by which the arbitration loss occurred, and the SCL and SDA lines are both set for high impedance, which releases the bus.

The arbitration loss is detected based on the timing of the next interrupt request signal (INTIIC) (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALD0 bit = 1 setting that has been made by software.

For details of interrupt request timing, see 17.7 I<sup>2</sup>C Interrupt Request Signals (INTIIC).

Figure 17-12. Arbitration Timing Example



**Table 17-5. Status During Arbitration and Interrupt Request Generation Timing**

| Status During Arbitration   | Interrupt Request Generation Timing  |
|---|--|
| During address transmission   | At falling edge of eighth or ninth clock following byte transfer <sup>Note 1</sup> |
| Read/write data after address transmission  |  |
| During extension code transmission  |  |
| Read/write data after extension code transmission                                 |  |
| During data transmission  |  |
| During $\overline{\text{ACK}}$ transfer period after data reception               |  |
| When restart condition is detected during data transfer                           |  |
| When stop condition is detected during data transfer                              | When stop condition is generated (when IICC0.SPIE0 bit = 1) <sup>Note 2</sup>      |
| When the SDA pin is at low level while attempting to generate a restart condition | At falling edge of eighth or ninth clock following byte transfer <sup>Note 1</sup> |
| When stop condition is detected while attempting to generate a restart condition  | When stop condition is generated (when SPIE0 bit = 1) <sup>Note 2</sup>            |
| When the SDA pin is at low level while attempting to generate a stop condition    | At falling edge of eighth or ninth clock following byte transfer <sup>Note 1</sup> |
| When the SCL pin is at low level while attempting to generate a restart condition |  |

- Notes 1.** When the IICC0.WTIM0 bit = 1, an interrupt request occurs at the falling edge of the ninth clock. When the WTIM0 bit = 0 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock.
- 2.** When there is a possibility that arbitration will occur, set the SPIE0 bit = 1 for master device operation.

### 17.13 Wakeup Function

The I<sup>2</sup>C bus slave function is a function that generates an interrupt request signal (INTIIC) when a local address or extension code has been received.

This function makes processing more efficient by preventing unnecessary interrupt requests from occurring when addresses do not match.

When a start condition is detected, wakeup standby mode is set. This wakeup standby mode is in effect while addresses are transmitted due to the possibility that an arbitration loss may change the master device (which has generated a start condition) to a slave device.

However, when a stop condition is detected, the IICC0.SPIE0 bit is set regardless of the wake up function, and this determines whether interrupt requests are enabled or disabled.

## 17.14 Communication Reservation

### 17.14.1 When communication reservation function is enabled (IICF0.IICRSV0 bit = 0)

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ( $\overline{ACK}$  is not returned and the bus was released when the IICC0.LRELO bit was set to “1”).

If the IICC0.STT0 bit is set (1) while the bus is not used, a start condition is automatically generated and wait status is set after the bus is released (after a stop condition is detected).

A communication is automatically started as the master by setting the IICC0.SPIE0 bit to 1, detecting the bus release due to an interrupt request (INTIIC) occurrence (detecting a stop condition), and then writing the address to the IIC0 register. Before detecting a stop condition, data written to the IIC0 register is set to invalid.

When the STT0 bit has been set (1), the operation mode (as start condition or as communication reservation) is determined according to the bus status.

If the bus has been released .....a start condition is generated

If the bus has not been released (standby mode) .....communication reservation

To detect which operation mode has been determined for the STT0 bit, set the STT0 bit (1), wait for the wait period, then check the IICS0.MSTS0 bit.

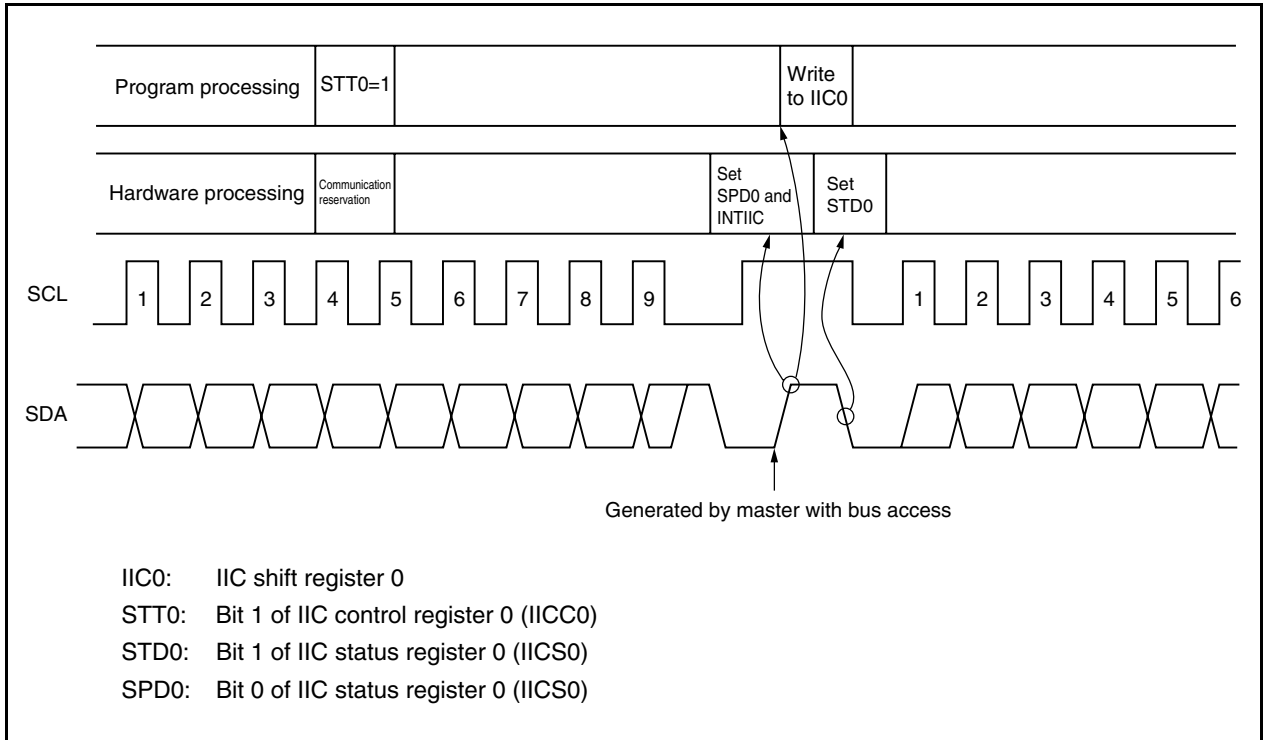
Wait periods, which should be set via software, are listed in Table 17-6. These wait periods can be set via the settings for the IICX0.CLX0, IICCL0.SMC0, and IICCL0.CL00 bits.

**Table 17-6. Wait Periods**

| Selection Clock             | CLX0 | SMC0 | CL00 | Wait Clock | Wait Time<br>When $f_{xx} = 64 \text{ MHz}$ |
|-----------------------------|------|------|------|------------|---|
| $f_{xx}/8$ (IICOCKS = 12H)  | 0    | 0    | 0    | 23 clocks  | 2.88 $\mu\text{s}$                          |
| $f_{xx}/10$ (IICOCKS = 13H) | 0    | 0    | 0    | 23 clocks  | 3.59 $\mu\text{s}$                          |
| $f_{xx}/4$ (IICOCKS = 10H)  | 0    | 0    | 1    | 43 clocks  | 2.69 $\mu\text{s}$                          |
| $f_{xx}/6$ (IICOCKS = 11H)  | 0    | 0    | 1    | 43 clocks  | 4.03 $\mu\text{s}$                          |
| $f_{xx}/8$ (IICOCKS = 12H)  | 0    | 0    | 1    | 43 clocks  | 5.38 $\mu\text{s}$                          |
| $f_{xx}/10$ (IICOCKS = 13H) | 0    | 0    | 1    | 43 clocks  | 6.72 $\mu\text{s}$                          |
| $f_{xx}/4$ (IICOCKS = 10H)  | 0    | 1    | x    | 15 clocks  | 0.94 $\mu\text{s}$                          |
| $f_{xx}/6$ (IICOCKS = 11H)  | 0    | 1    | x    | 15 clocks  | 1.41 $\mu\text{s}$                          |
| $f_{xx}/8$ (IICOCKS = 12H)  | 0    | 1    | x    | 15 clocks  | 1.88 $\mu\text{s}$                          |
| $f_{xx}/10$ (IICOCKS = 13H) | 0    | 1    | x    | 15 clocks  | 2.34 $\mu\text{s}$                          |
| $f_{xx}/8$ (IICOCKS = 12H)  | 1    | 1    | x    | 9 clocks   | 1.13 $\mu\text{s}$                          |
| $f_{xx}/10$ (IICOCKS = 13H) | 1    | 1    | x    | 9 clocks   | 1.41 $\mu\text{s}$                          |

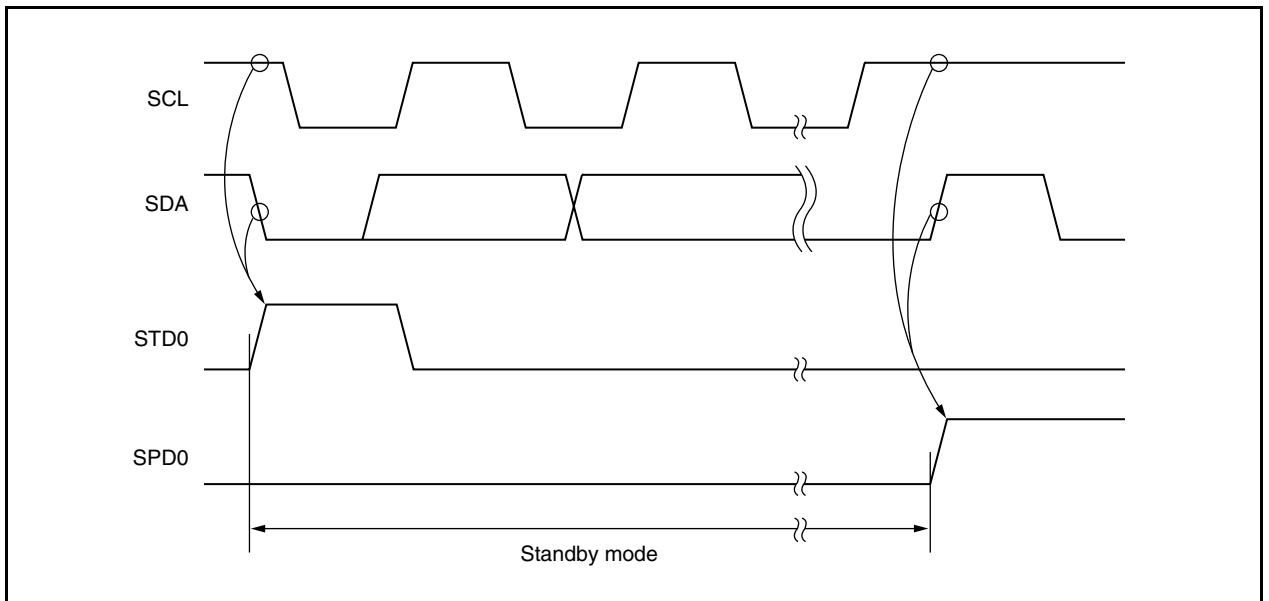
The communication reservation timing is shown below.

Figure 17-13. Communication Reservation Timing



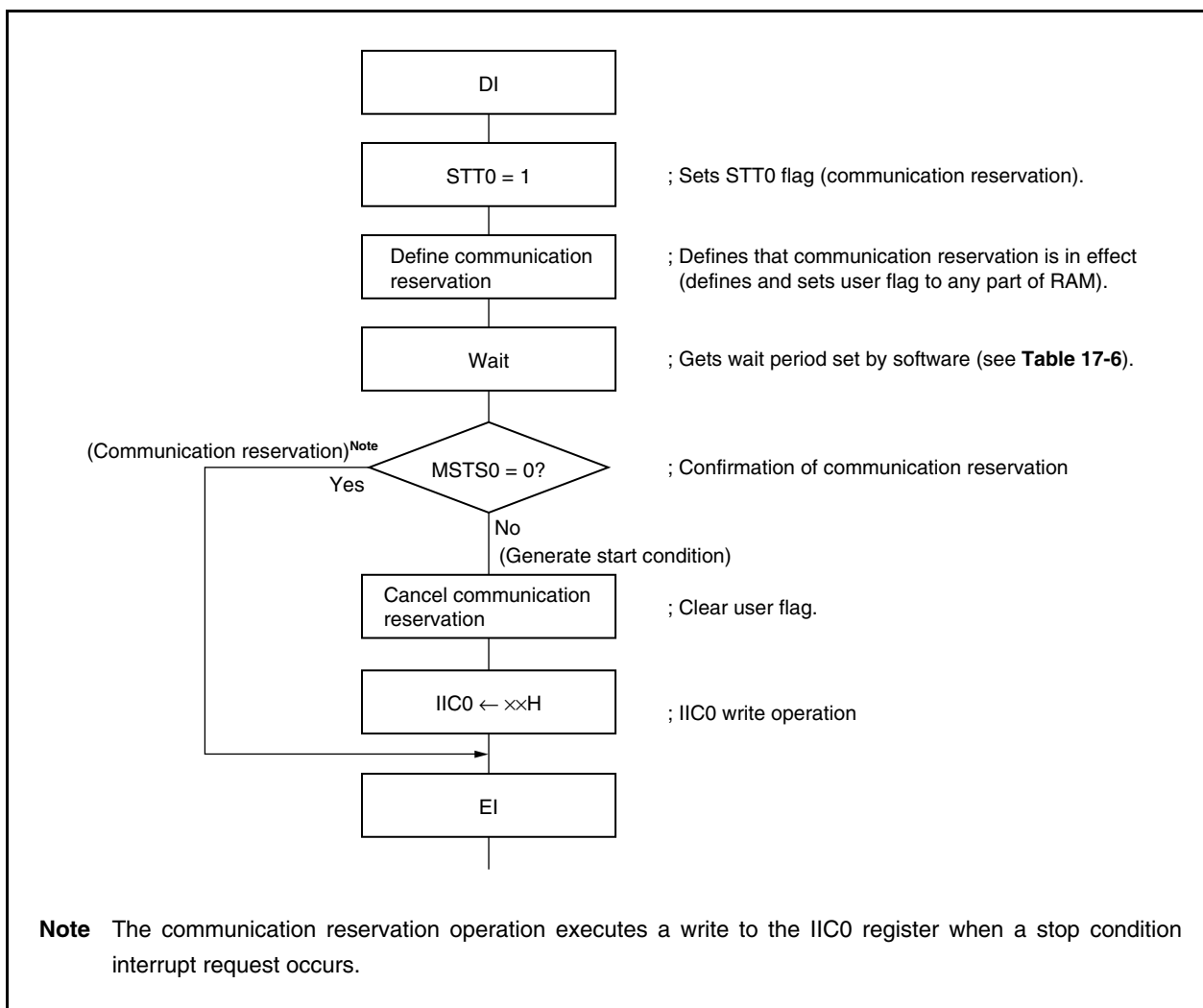
Communication reservations are accepted via the following timing. After the IICS0.STD0 bit is set to 1, a communication reservation can be made by setting the IICC0.STT0 bit to 1 before a stop condition is detected.

Figure 17-14. Timing for Accepting Communication Reservations



The communication reservation flowchart is illustrated below.

Figure 17-15. Communication Reservation Flowchart





**17.14.2 When communication reservation function is disabled (IICF0.IICRSV0 bit = 1)**

When the IICC0.STT0 bit is set when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. The following two statuses are included in the status where bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ( $\overline{ACK}$  is not returned and the bus was released when the IICC0.LREL0 bit was set to 1)

To confirm whether the start condition was generated or request was rejected, check the IICF0.STCF0 flag. The time shown in Table 17-7 is required until the STCF0 flag is set after setting the STT0 bit = 1. Therefore, secure the time by software.

**Table 17-7. Wait Periods**

| Selection Clock             | CLX0 | SMC0 | CL00 | Wait Clock | Wait Time<br>When $f_{xx} = 64$ MHz |
|-----------------------------|------|------|------|------------|-------------------------------------|
| $f_{xx}/8$ (IICOCKS = 12H)  | 0    | 0    | 0    | 5 clocks   | 0.63 $\mu$ s                        |
| $f_{xx}/10$ (IICOCKS = 13H) | 0    | 0    | 0    | 5 clocks   | 0.78 $\mu$ s                        |
| $f_{xx}/4$ (IICOCKS = 10H)  | 0    | 0    | 1    | 5 clocks   | 0.31 $\mu$ s                        |
| $f_{xx}/6$ (IICOCKS = 11H)  | 0    | 0    | 1    | 5 clocks   | 0.47 $\mu$ s                        |
| $f_{xx}/8$ (IICOCKS = 12H)  | 0    | 0    | 1    | 5 clocks   | 0.63 $\mu$ s                        |
| $f_{xx}/10$ (IICOCKS = 13H) | 0    | 0    | 1    | 5 clocks   | 0.78 $\mu$ s                        |

### 17.15 Cautions

- (1) When IICF0.STCEN0 bit = 0  
Immediately after I<sup>2</sup>C operation is enabled, the bus communication status (IICF0.IICBSY0 bit = 1) is recognized regardless of the actual bus status. To execute master communication in the status where a stop condition has not been detected, generate a stop condition and then release the bus before starting the master communication.

Use the following sequence for generating a stop condition.

<1> Set the IICCL0 register.

<2> Set the IICC0.IICE0 bit.

<3> Set the IICC0.SPT0 bit.

- (2) When IICF0.STCEN0 bit = 1  
Immediately after I<sup>2</sup>C operation is enabled, the bus released status (IICBSY0 bit = 0) is recognized regardless of the actual bus status. To generate the first start condition (IICC0.STT0 bit = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

- (3) When the IICC0.IICE0 bit of the V850E/IF3 and V850E/IG3 is set to 1 while communications with other devices are in progress, the start condition may be detected depending on the status of the communication line. Be sure to set the IICC0.IICE0 bit to 1 when the SCL and SDA lines are high level.

- <R> (4) Procedure for starting or stopping I<sup>2</sup>C operation

(a) Starting I<sup>2</sup>C operation

<1> Select the division clock by using the IICOCKS.IICOCKS1 and IICOCKS.IICOCKS0 bits and set the IICOCKS.IICOCKSEN bit to 1 (to enable I<sup>2</sup>C division clock operation).

<2> Specify the transfer speed by using the IICCL0 and IICX0 registers.

<3> Set the IICC0.IICE0 bit to 1 (to start I<sup>2</sup>C operation).

When changing the transfer speed for I<sup>2</sup>C, do so after clearing the IICC0.IICE0 bit to 0.

(b) Stopping I<sup>2</sup>C operation

<1> Clear the IICC0.IICE0 bit to 0 (to stop I<sup>2</sup>C operation).

<2> Clear the IICOCKS.IICOCKSEN bit to 0 (to disable I<sup>2</sup>C division clock operation).

- (5) After the IICC0.STT0 and IICC0.SPT0 bits have been set to 1, they must not be re-set without being cleared to 0 first.

- (6) If transmission has been reserved, set the IICC0.SPIE0 bit to 1 so that an interrupt request is generated by the detection of a stop condition. After an interrupt request has been generated, the wait state will be released by writing communication data to I<sup>2</sup>C, then transferring will begin. If an interrupt is not generated by the detection of a stop condition, transmission will halt in the wait state because an interrupt request was not generated. However, it is not necessary to set the SPIE0 bit to 1 for the software to detect the IICS0.MSTS0 bit.

## 17.16 Communication Operations

The following shows three operation procedures with the flowchart.

### (1) Master operation in single master system

The flowchart when using the V850E/IF3 and V850E/IG3 as the master in a single master system is shown below.

This flowchart is broadly divided into the initial settings and communication processing. Execute the initial settings at startup. If communication with the slave is required, prepare the communication and then execute communication processing.

### (2) Master operation in multimaster system

In the I<sup>2</sup>C bus multimaster system, whether the bus is released or used cannot be judged by the I<sup>2</sup>C bus specifications when the bus takes part in a communication. Here, when data and clock are at a high level for a certain period (1 frame), the V850E/IF3 and V850E/IG3 take part in a communication with bus released state.

This flowchart is broadly divided into the initial settings, communication waiting, and communication processing. The processing when the V850E/IF3 and V850E/IG3 lose in arbitration and are specified as the slave is omitted here, and only the processing as the master is shown. Execute the initial settings at startup to take part in a communication. Then, wait for the communication request as the master or wait for the specification as the slave. The actual communication is performed in the communication processing, and it supports the transmission/reception with the slave and the arbitration with other masters.

### (3) Slave operation

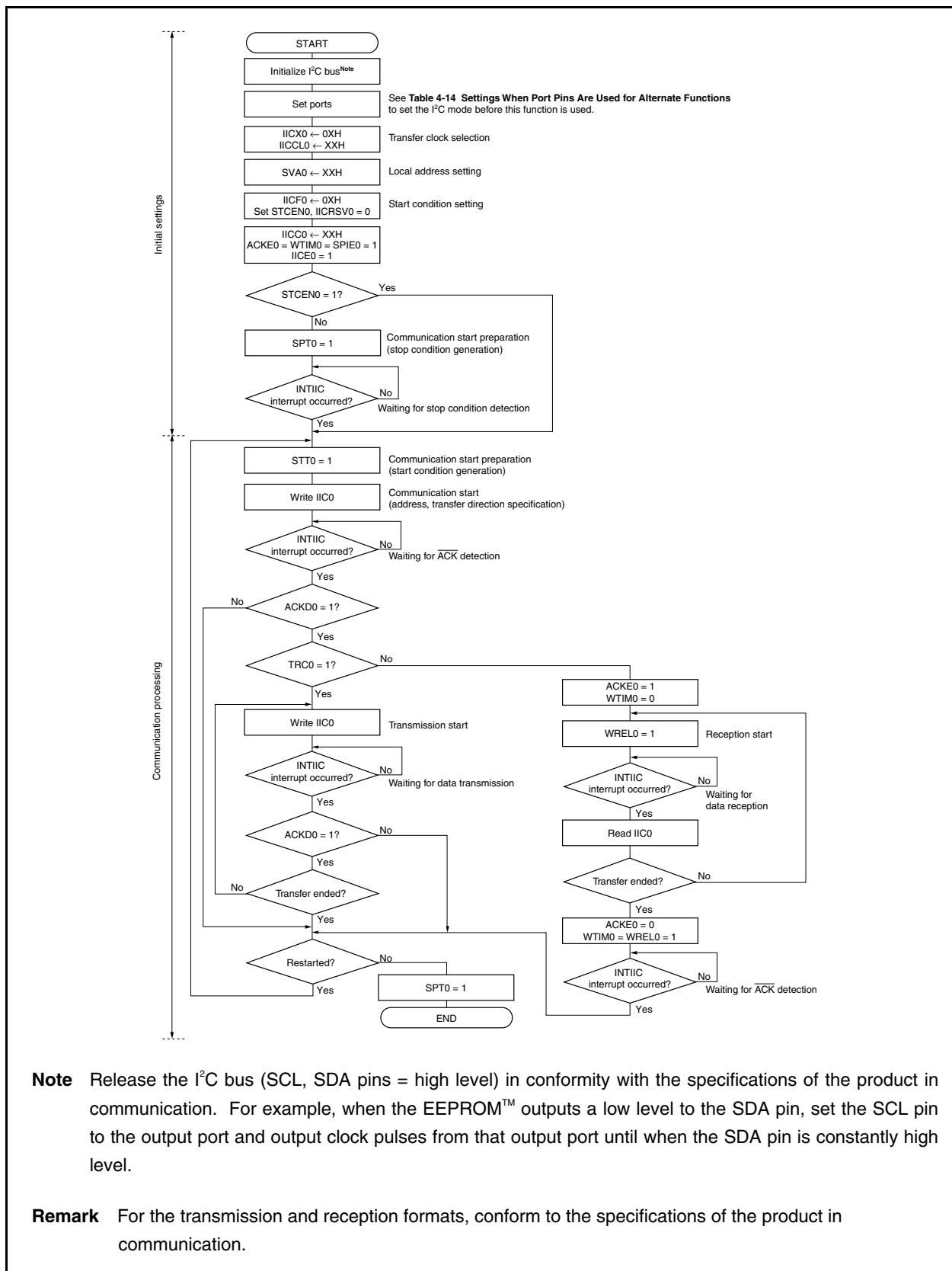
An example of when the V850E/IF3 and V850E/IG3 are used as the slave is shown below.

When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIIC interrupt occurrence (communication waiting). When the INTIIC interrupt occurs, the communication status is judged and its result is passed as a flag over to the main processing.

By checking the flags, necessary communication processing is performed.

17.16.1 Master operation in single master system

Figure 17-16. Master Operation in Single Master System



17.16.2 Master operation in multimaster system

Figure 17-17. Master Operation in Multimaster System (1/3)

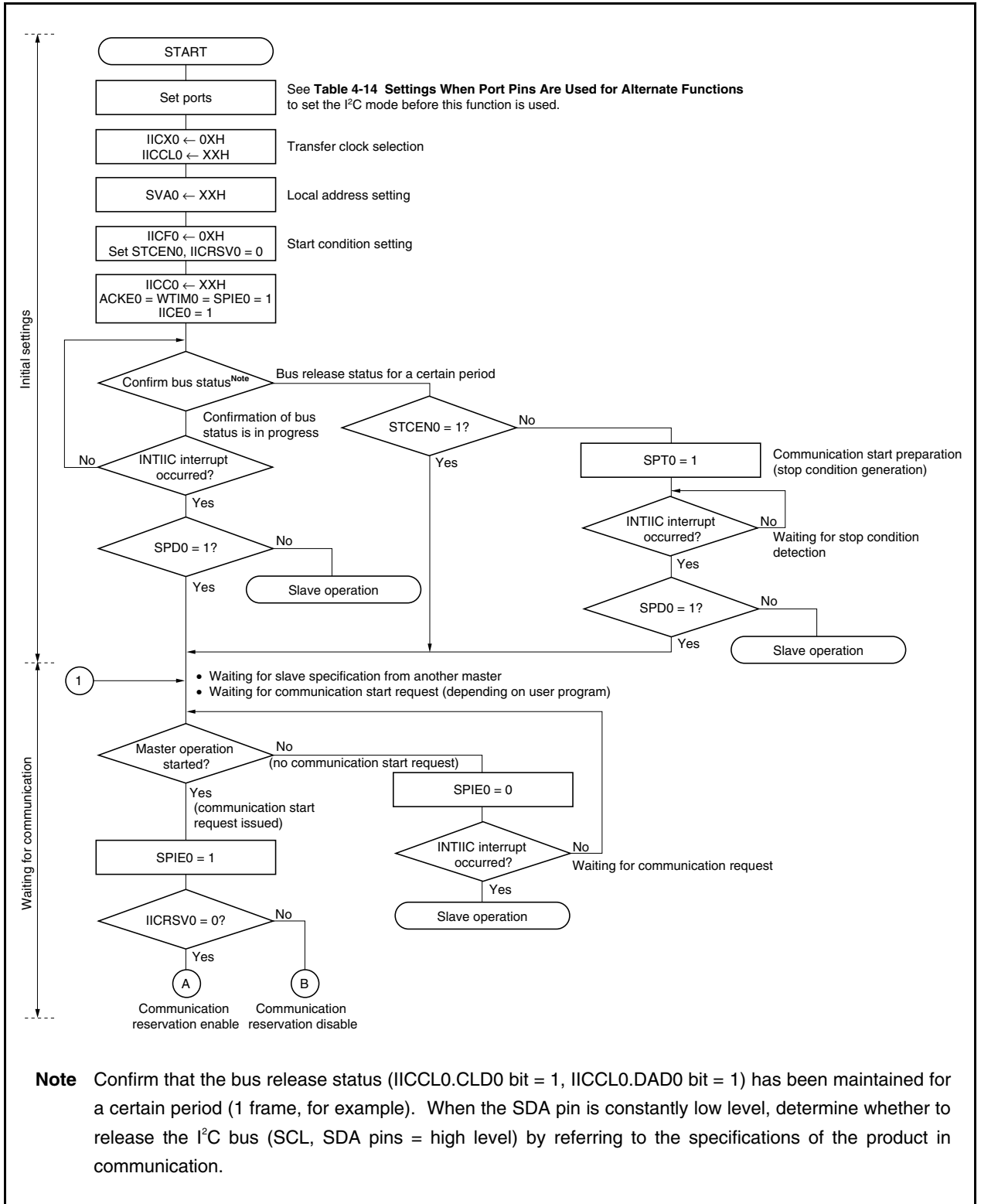


Figure 17-17. Master Operation in Multimaster System (2/3)

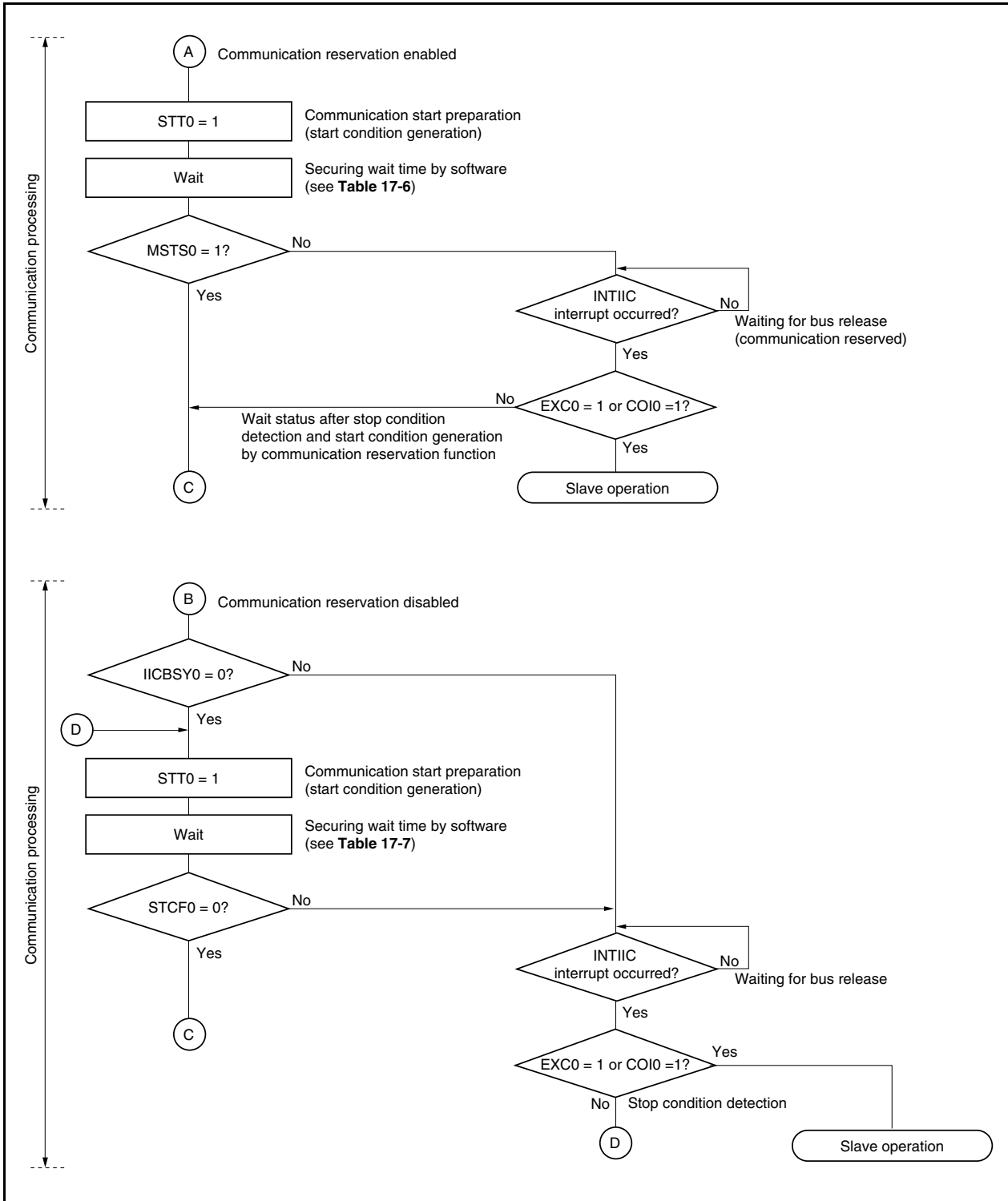
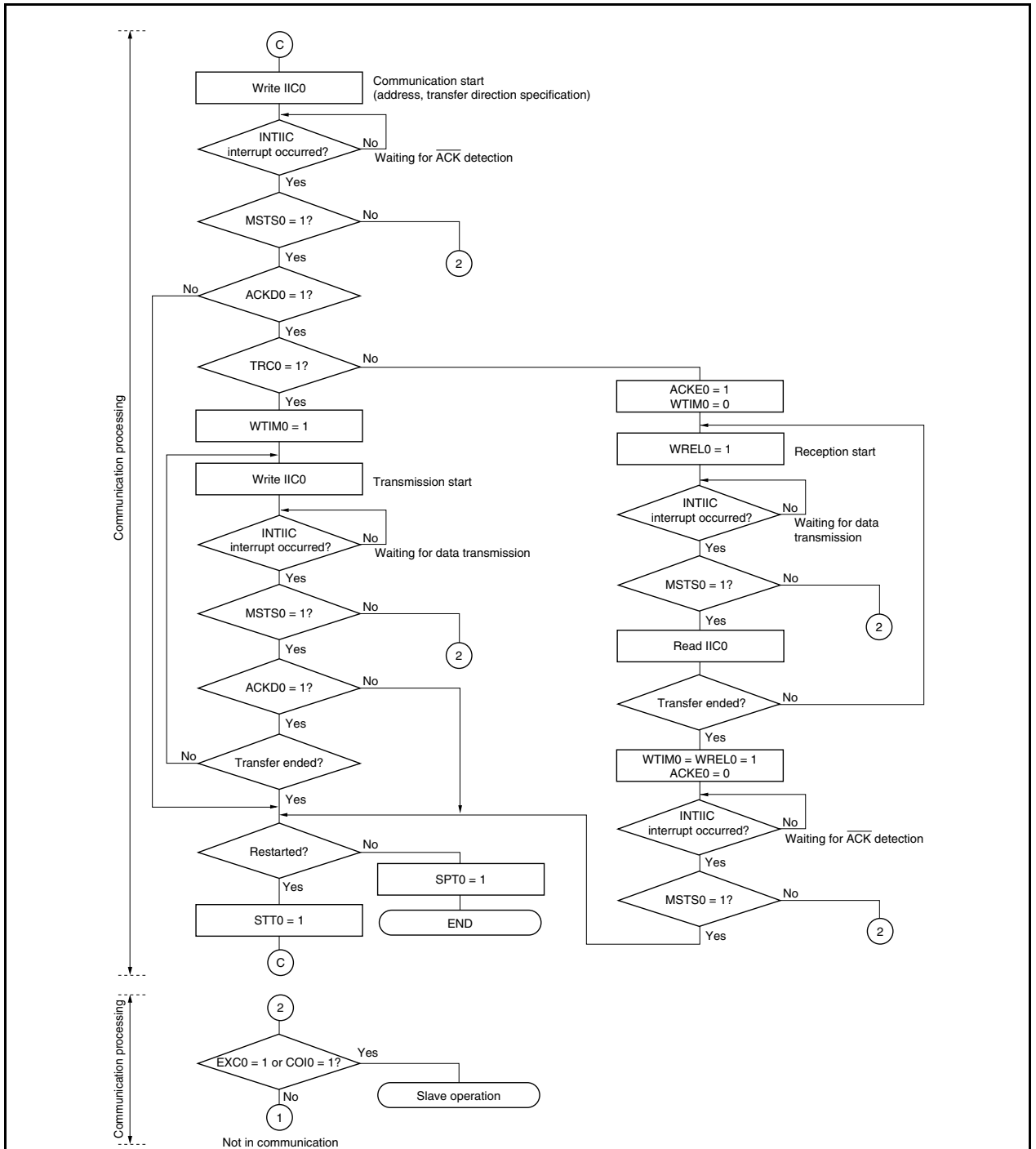


Figure 17-17. Master Operation in Multimaster System (3/3)



- Remarks**
1. Conform the transmission and reception formats to the specifications of the product in communication.
  2. When using the V850E/IF3 and V850E/IG3 as the master in the multimaster system, read the IICS0.MSTS0 bit for each INTIIC interrupt occurrence to confirm the arbitration result.
  3. When using the V850E/IF3 and V850E/IG3 as the slave in the multimaster system, confirm the status using the IICS0 and IICF0 registers for each INTIIC interrupt occurrence to determine the next processing.

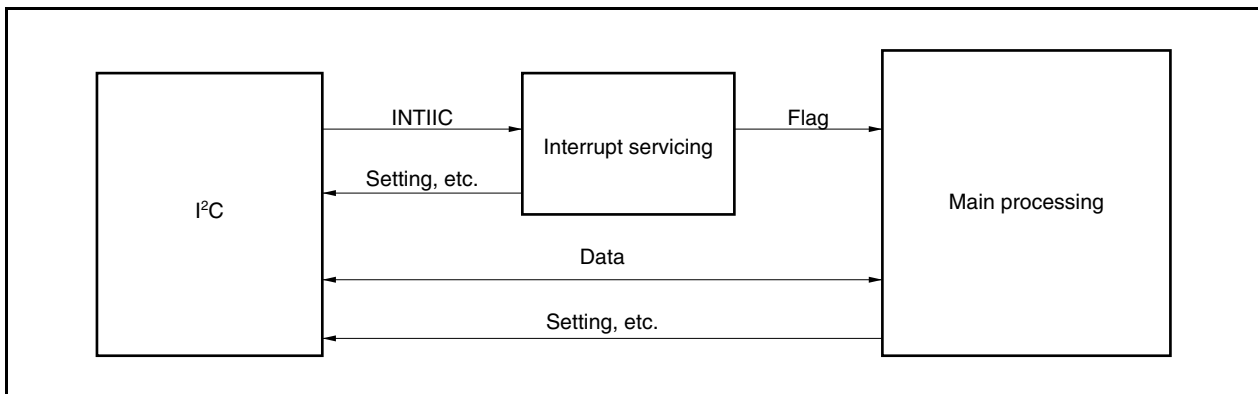
### 17.16.3 Slave operation

The following shows the processing procedure of the slave operation.

Basically, the operation of the slave device is event-driven. Therefore, processing by an INTIIC interrupt (processing requiring a significant change of the operation status, such as stop condition detection during communication) is necessary.

The following description assumes that data communication does not support extension codes. Also, it is assumed that the INTIIC interrupt servicing performs only status change processing and that the actual data communication is performed during the main processing.

**Figure 17-18. Software Outline During Slave Operation**



Therefore, the following three flags are prepared so that the data transfer processing can be performed by transmitting these flags to the main processing instead of the INTIIC signal.

#### (1) Communication mode flag

This flag indicates the following communication statuses.

Clear mode: Data communication not in progress

Communication mode: Data communication in progress (valid address detection stop condition detection,  $\overline{\text{ACK}}$  from master not detected, address mismatch)

#### (2) Ready flag

This flag indicates that data communication is enabled. This is the same status as an INTIIC interrupt during normal data transfer. This flag is set in the interrupt processing block and cleared in the main processing block. The ready flag for the first data for transmission is not set in the interrupt processing block, so the first data is transmitted without clearance processing (the address match is regarded as a request for the next data).

#### (3) Communication direction flag

This flag indicates the direction of communication and is the same as the value of the IICS0.TRC0 bit.

The following shows the operation of the main processing block during slave operation.

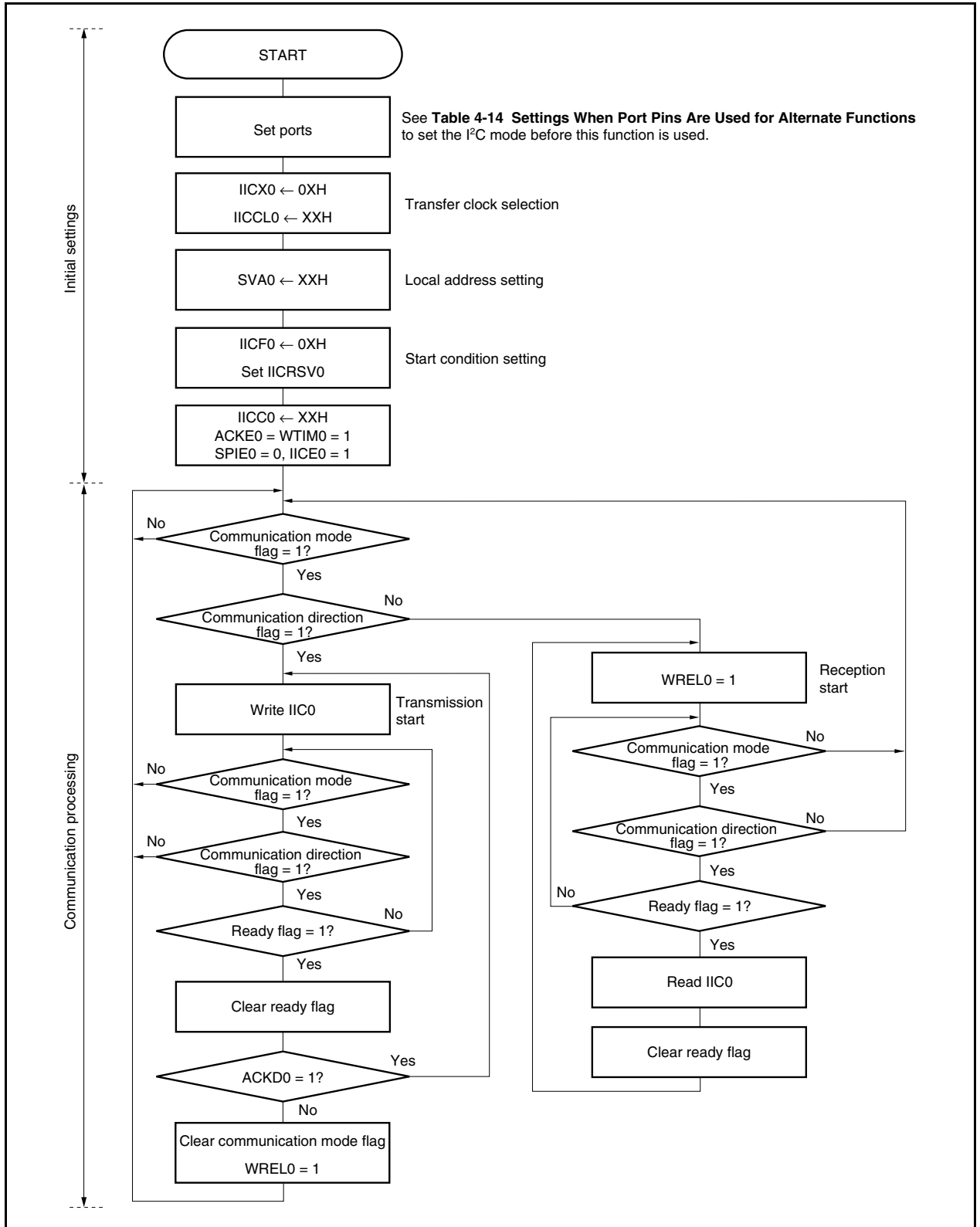
Start I<sup>2</sup>C and wait for the communication enabled status. When communication is enabled, perform transfer using the communication mode flag and ready flag (the processing of the stop condition and start condition is performed by interrupts, conditions are confirmed by flags).

For transmission, repeat the transmission operation until the master device stops returning  $\overline{\text{ACK}}$ . When the master device stops returning  $\overline{\text{ACK}}$ , transfer is end.



For reception, receive the required number of data and do not return  $\overline{\text{ACK}}$  for the next data immediately after transfer is end. After that, the master device generates the stop condition or restart condition. This causes exit from communications.

Figure 17-19. Slave Operation Flowchart (1)

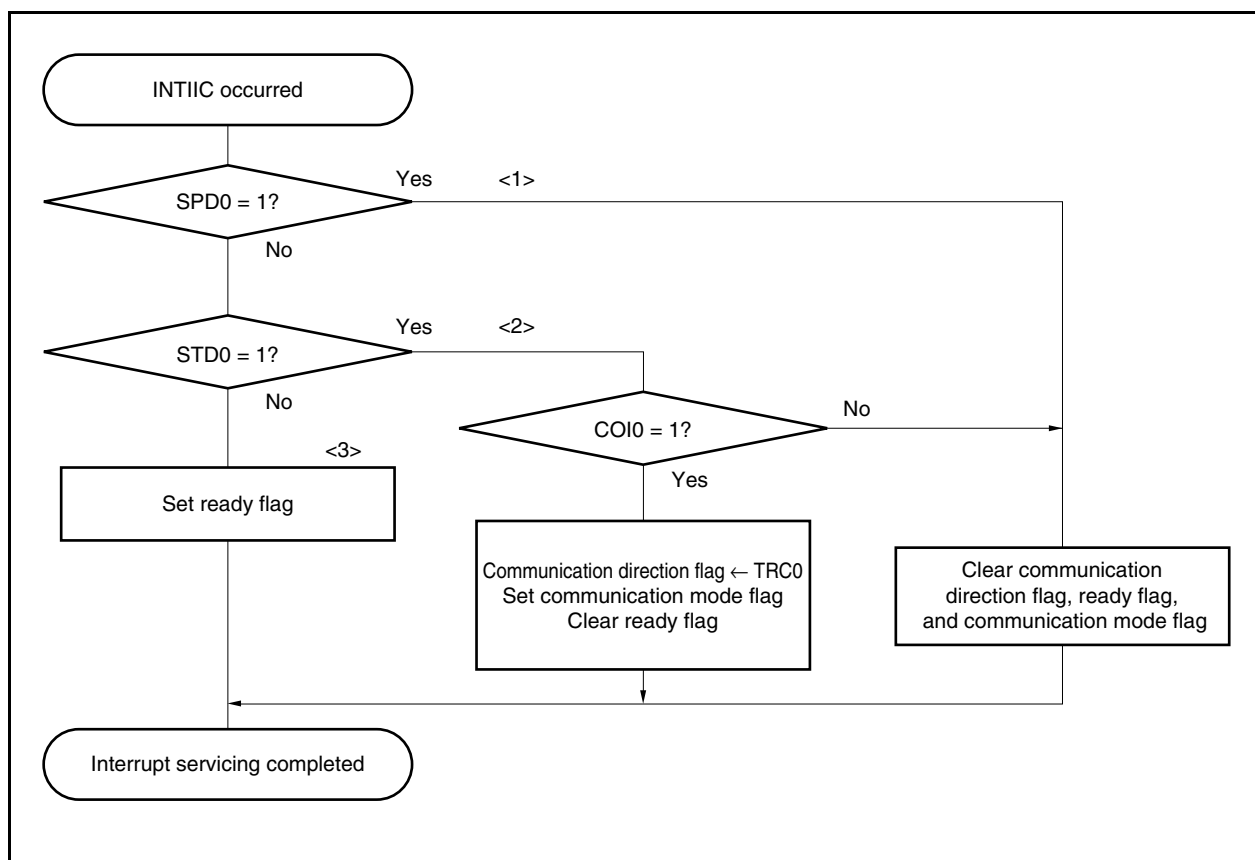


The following shows an example of the processing of the slave device by an INTIIC interrupt (it is assumed that no extension codes are used here). During an INTIIC interrupt, the status is confirmed and the following steps are executed.

- <1> When a stop condition is detected, communication is terminated.
- <2> When a start condition is detected, the address is confirmed. If the address does not match, communication is terminated. If the address matches, the communication mode is set and wait is released, and operation returns from the interrupt (the ready flag is cleared).
- <3> For data transmission/reception, when the ready flag is set, operation returns from the interrupt while the I<sup>2</sup>C bus remains in the wait status.

**Remark** <1> to <3> in the above correspond to <1> to <3> in **Figure 17-20 Slave Operation Flowchart (2)**.

**Figure 17-20. Slave Operation Flowchart (2)**



### 17.17 Timing of Data Communication

When using I<sup>2</sup>C bus mode, the master device generates an address via the serial bus to select one of several slave devices as its communication partner.

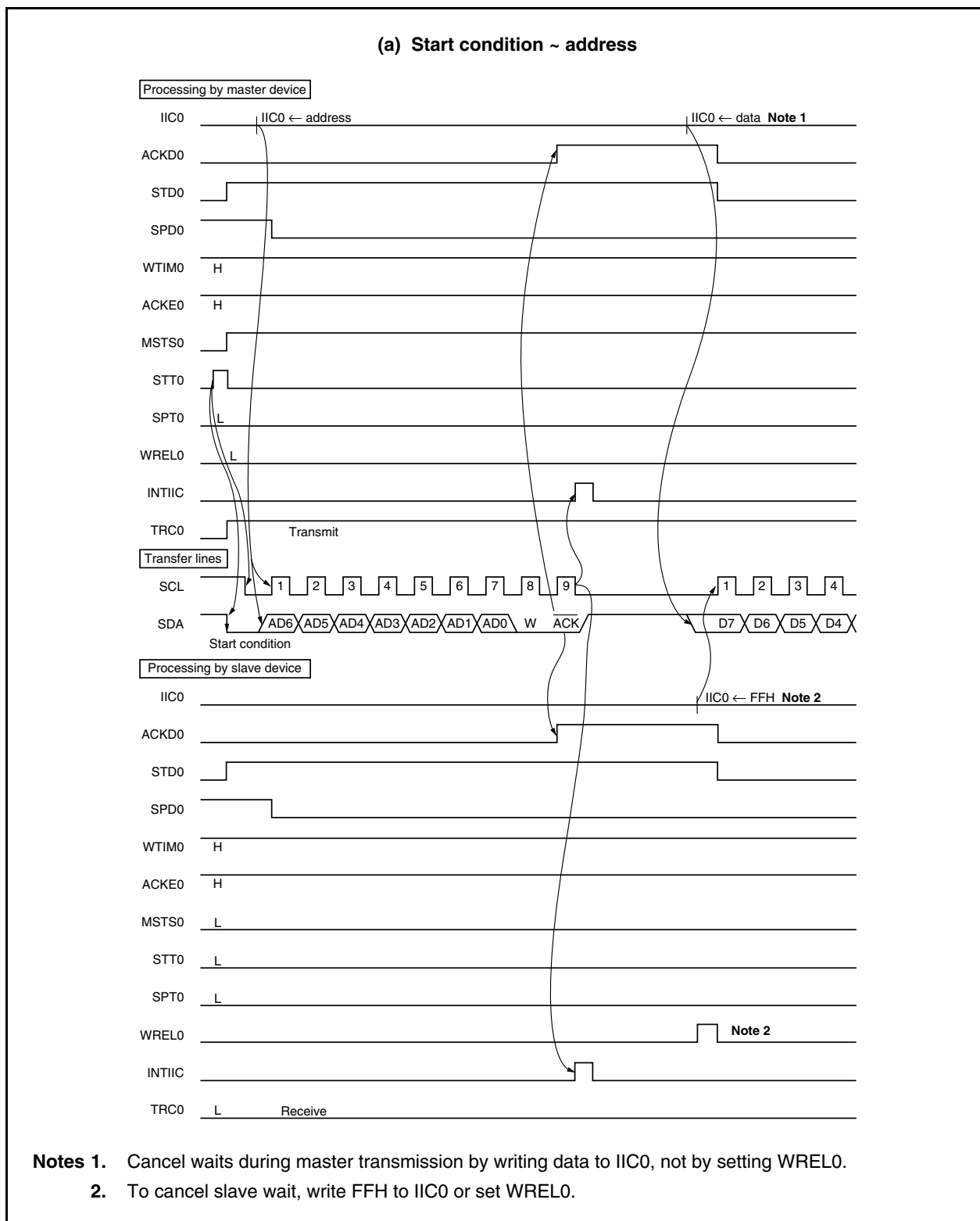
After outputting the slave address, the master device transmits the IICS0.TRC0 bit that specifies the data transfer direction and then starts serial communication with the slave device.

The IIC0 register's shift operation is synchronized with the falling edge of the serial clock (SCL pin). The transmit data is transferred to the SO latch and is output (MSB first) via the SDA pin.

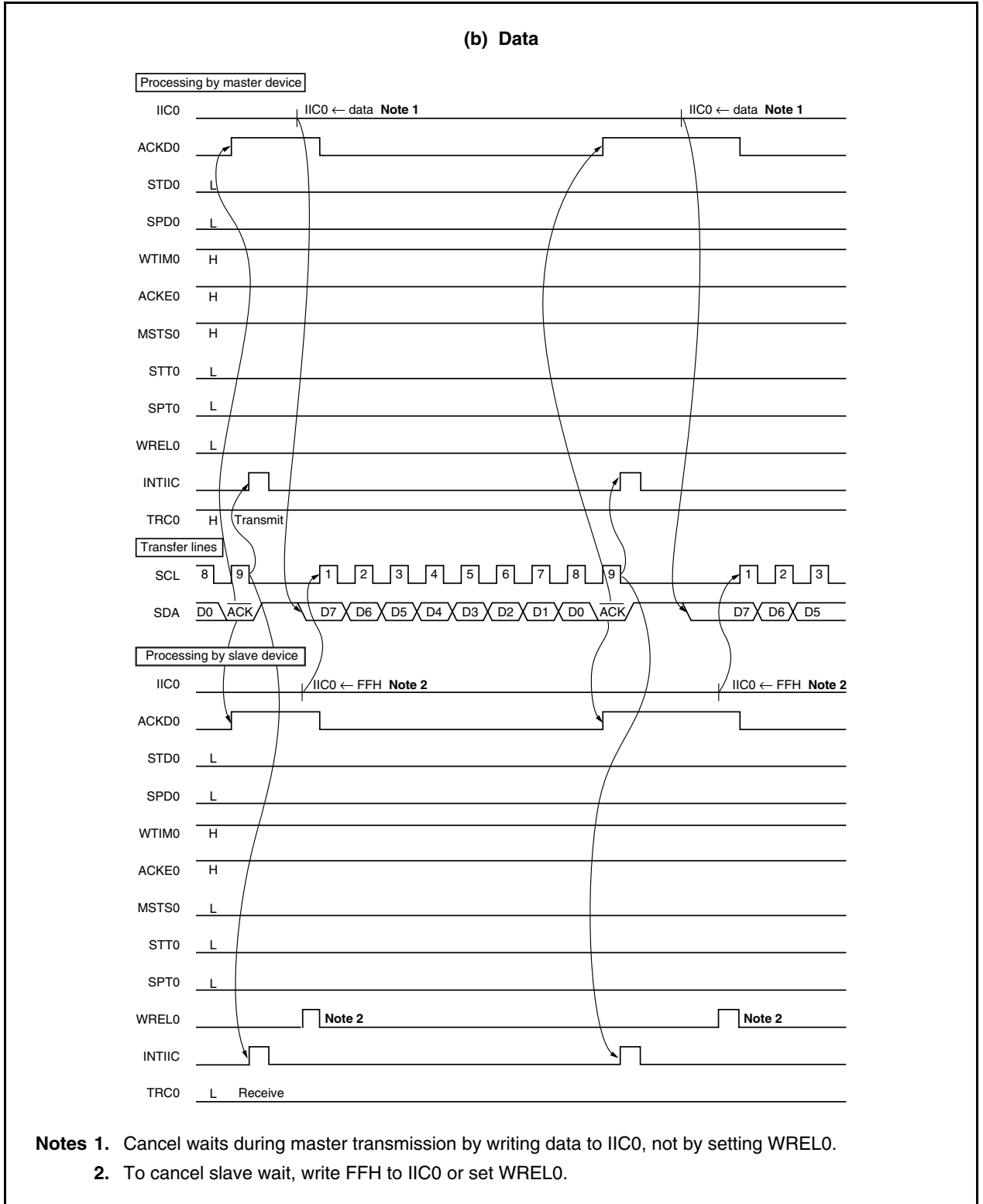
Data input via the SDA pin is captured by the IIC0 register at the rising edge of the SCL pin.

The data communication timing is shown below.

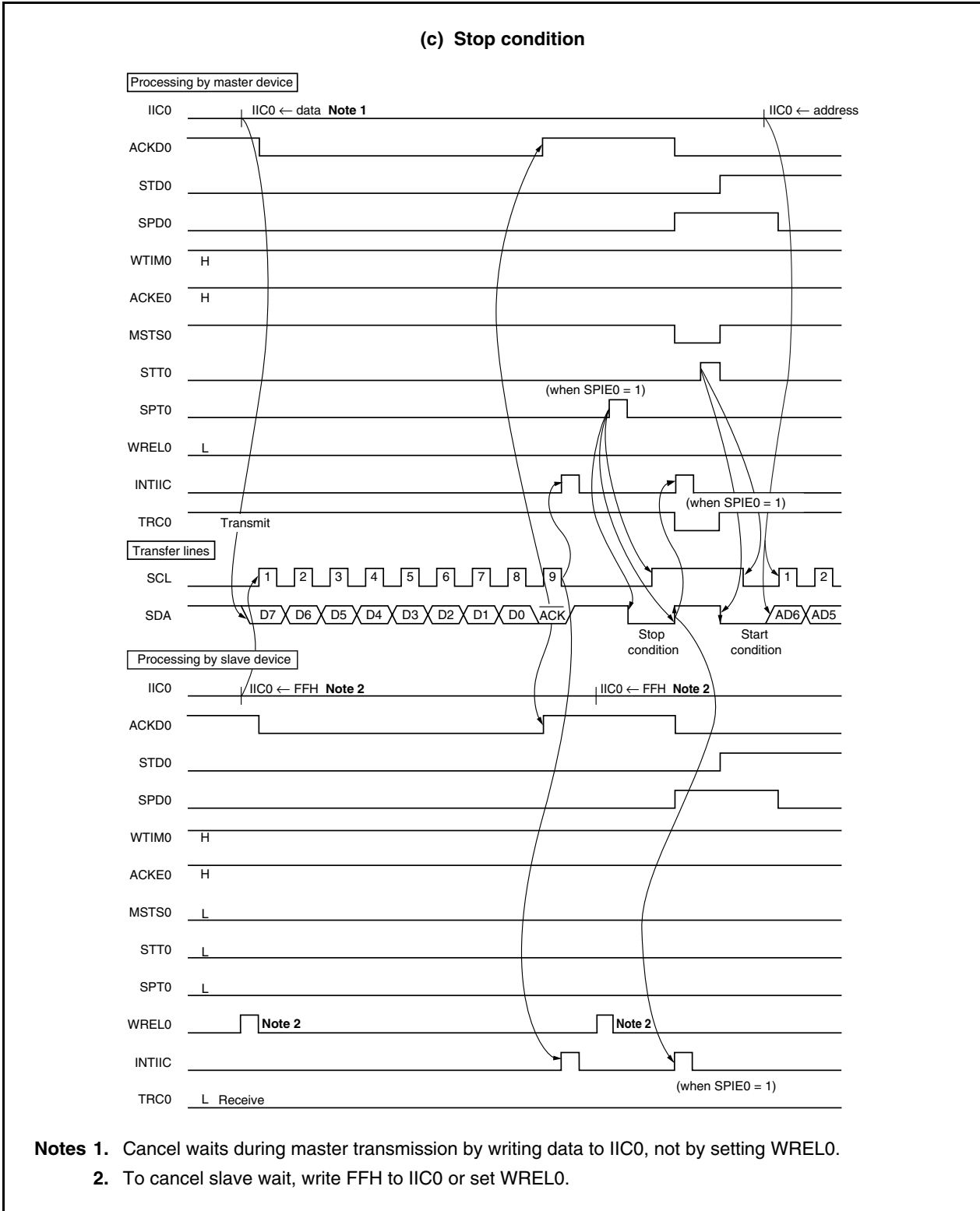
**Figure 17-21. Example of Master to Slave Communication  
(When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)**



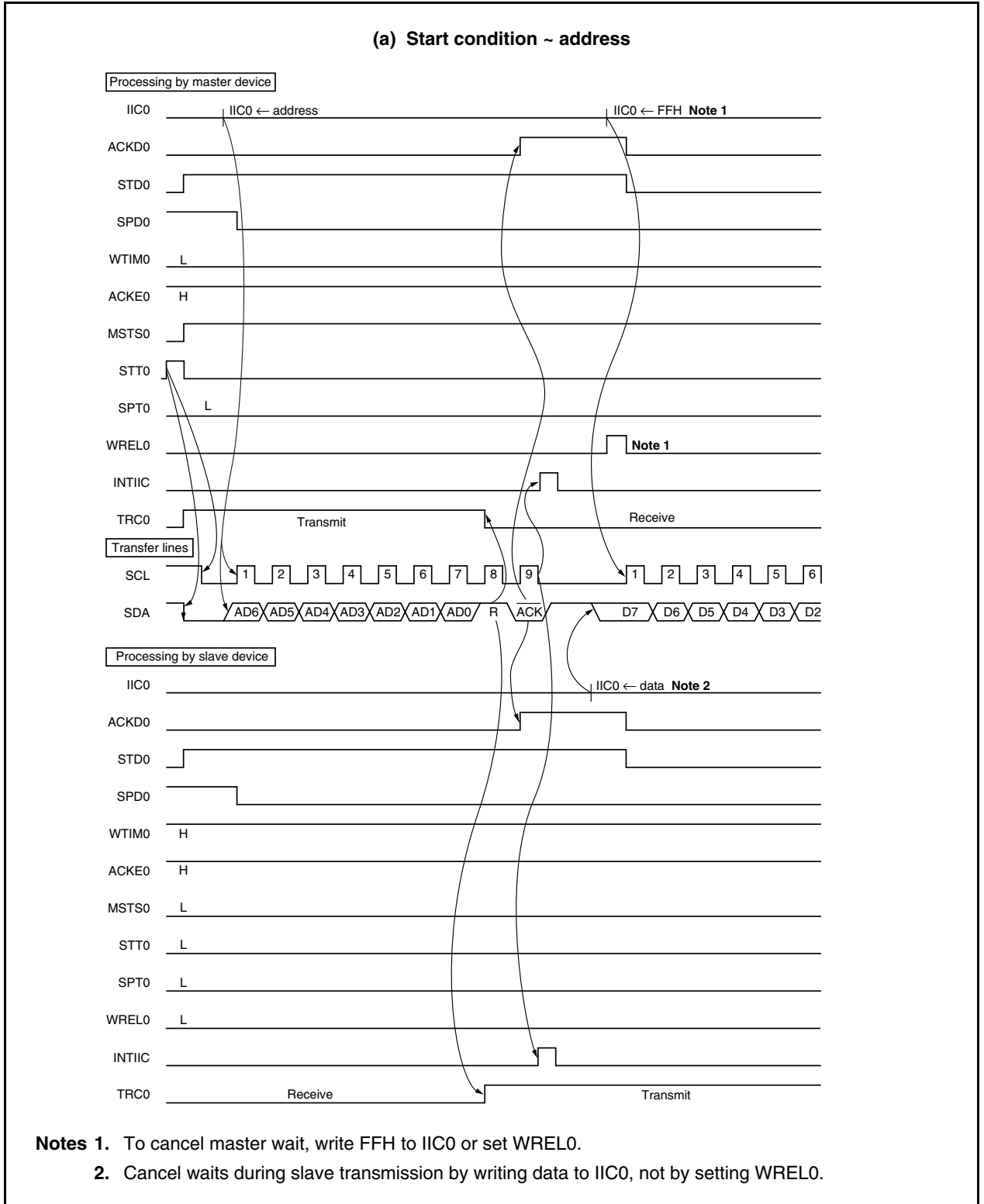
**Figure 17-21. Example of Master to Slave Communication  
(When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)**



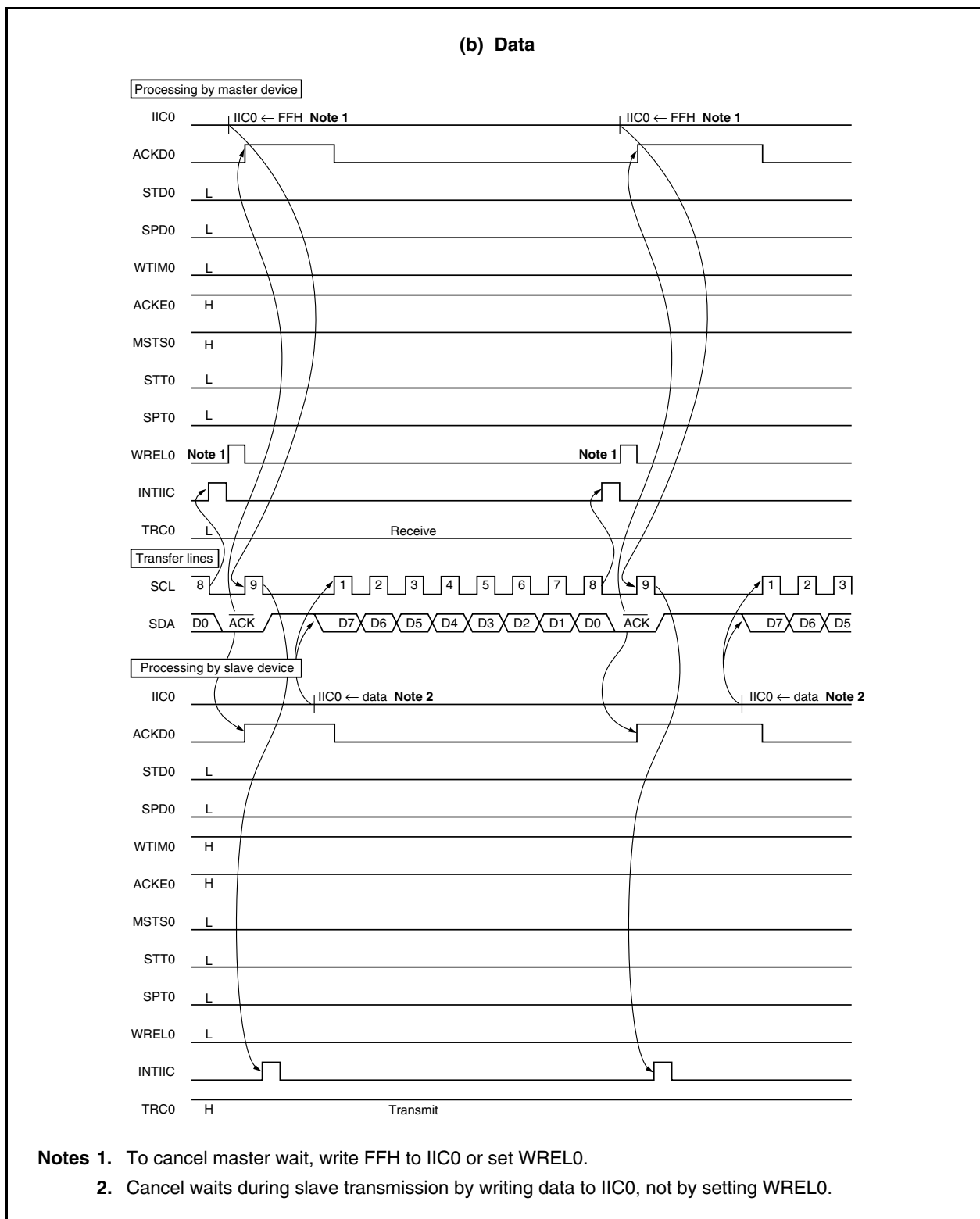
**Figure 17-21. Example of Master to Slave Communication  
(When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)**



**Figure 17-22. Example of Slave to Master Communication**  
**(When 8-Clock Wait for Master and 9-Clock Wait for Slave Are Selected) (1/3)**

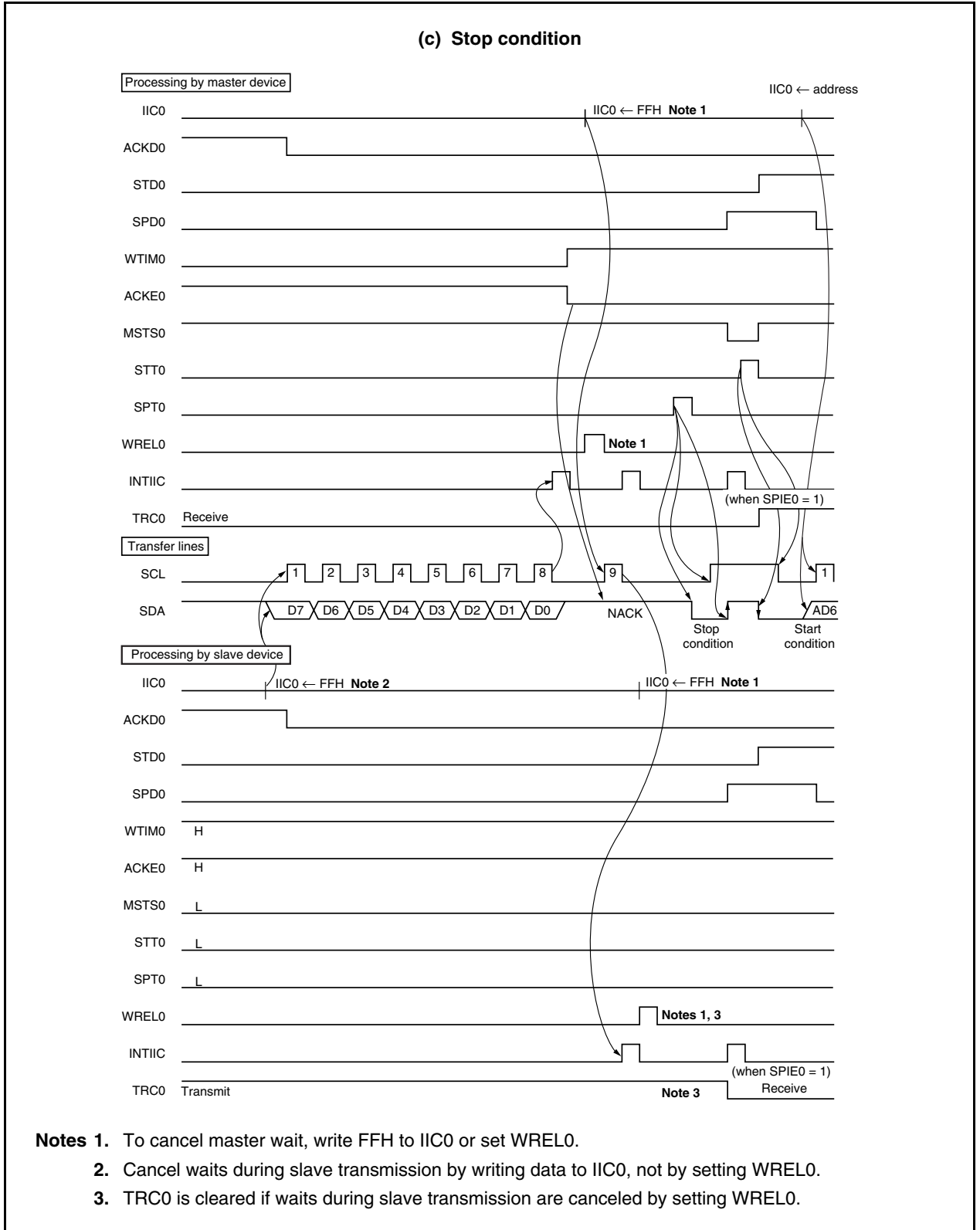


**Figure 17-22. Example of Slave to Master Communication**  
**(When 8-Clock Wait for Master and 9-Clock Wait for Slave Are Selected) (2/3)**





**Figure 17-22. Example of Slave to Master Communication**  
**(When 8-Clock → 9-Clock Wait for Master and 9-Clock Wait for Slave Are Selected) (3/3)**



## CHAPTER 18 BUS CONTROL FUNCTION

The  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A are provided with an external bus interface function via which external memories such as ROM and RAM, and I/O devices can be connected to areas other than the internal ROM, internal RAM, or on-chip peripheral I/O registers via ports 0, 1 (separate bus mode only), 3, 4, and DL. These ports control address/data I/O, the read/write strobe signal, waits, the clock output, and the address strobe signal.

The  $\mu$ PD70F3453GC-8EA-A, GF version of the V850E/IG3, and the V850E/IF3 are not provided with an external bus interface function.

This chapter describes the  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A as target microcontrollers.

### 18.1 Features

- 16-bit/8-bit data bus sizing function
- 2-space chip select function
- Wait function
  - Programmable wait function, through which up to 7 wait states can be inserted for each memory block
  - Address setup wait and address hold wait insertion functions, through which 1 wait state can be inserted for each memory block
  - External wait function via  $\overline{\text{WAIT}}$  pin
- Idle state insertion function
  - A low-speed device can be connected by inserting an idle state after a read cycle.
- External device connection enabled via bus control/port alternate function pins
- Separate bus mode (8-bit address bus, 8-bit/16-bit data bus)
- Multiplexed bus mode (16-bit address bus, 8-bit/16-bit data bus)
- Support for little endian
- External bus clock frequency ( $f_{\text{BUS}}$ ): 32 MHz/16 MHz selectable function
- Misaligned access is possible.
- Up to 4 MB of physical memory can be connected (1 MB are shared with the internal ROM area).

## 18.2 Bus Control Pins

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

**Table 18-1. Bus Control Pins (When Separate Bus Mode Selected)**

| Bus Control Pin                                | I/O    | Function                  | Alternate-Function Pin | Register to Switch Between Port Mode/<br>Alternate-Function Mode |
|--|--------|---------------------------|------------------------|--|
| AD0 to AD15                                    | I/O    | Data bus                  | PDL0 to PDL15          | PMCDL register   |
| A0 to A7                                       | Output | Address bus               | P10 to P17             | PMC1 register  |
| $\overline{\text{WAIT}}$                       | Input  | External wait control     | P44                    | PMC4 register  |
| CLKOUT   | Output | External bus clock output | P07                    | PMC0 register  |
| $\overline{\text{CS0}}, \overline{\text{CS1}}$ | Output | Chip select               | P34, P32               | PMC3 register  |
| $\overline{\text{WR0}}, \overline{\text{WR1}}$ | Output | Write strobe signal       | P46, P45               | PMC4 register  |
| $\overline{\text{RD}}$                         | Output | Read strobe signal        | P47                    | PMC4 register  |
| ASTB   | Output | Address strobe signal     | P37                    | PMC3 register  |

**Table 18-2. Bus Control Pins (When Multiplexed Bus Mode Selected)**

| Bus Control Pin                                | I/O    | Function                  | Alternate-Function Pin | Register to Switch Between Port Mode/<br>Alternate-Function Mode |
|--|--------|---------------------------|------------------------|--|
| AD0 to AD15                                    | I/O    | Address/data bus          | PDL0 to PDL15          | PMCDL register   |
| $\overline{\text{WAIT}}$                       | Input  | External wait control     | P44                    | PMC4 register  |
| CLKOUT   | Output | External bus clock output | P07                    | PMC0 register  |
| $\overline{\text{CS0}}, \overline{\text{CS1}}$ | Output | Chip select               | P34, P32               | PMC3 register  |
| $\overline{\text{WR0}}, \overline{\text{WR1}}$ | Output | Write strobe signal       | P46, P45               | PMC4 register  |
| $\overline{\text{RD}}$                         | Output | Read strobe signal        | P47                    | PMC4 register  |
| ASTB   | Output | Address strobe signal     | P37                    | PMC3 register  |

### 18.2.1 Pin status during internal ROM, internal RAM, and on-chip peripheral I/O access

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

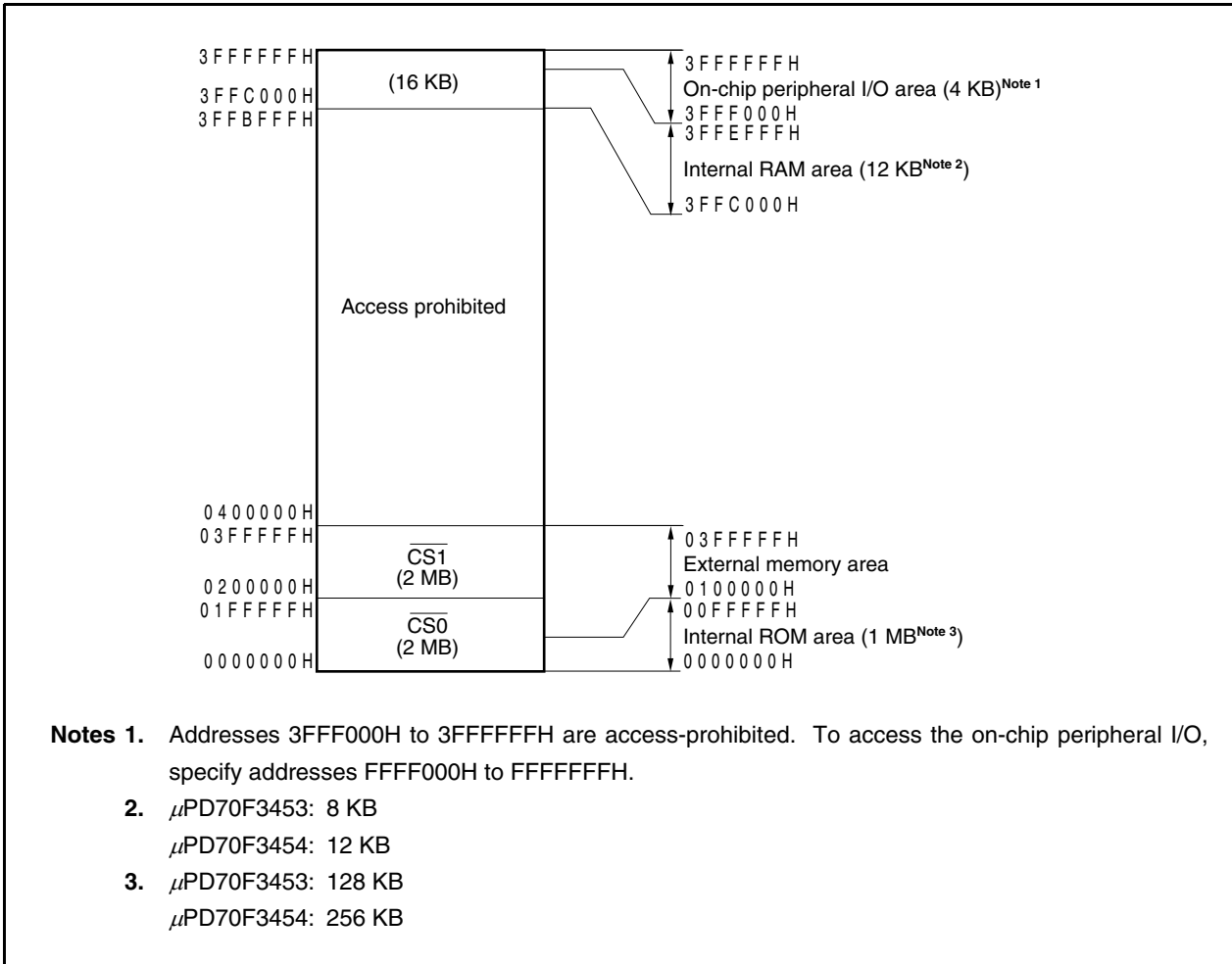
**Table 18-1. Pin Status List In Internal ROM, Internal RAM, and On-Chip Peripheral I/O Access**

| Access Destination          | Internal ROM               | Internal RAM               | On-Chip Peripheral I/O     |
|-----------------------------|----------------------------|----------------------------|----------------------------|
| Address bus                 | Undefined                  | Undefined                  | <b>Note 1</b>              |
| Data bus                    | Hi-Z                       | Hi-Z                       | Hi-Z                       |
| External bus control signal | Inactive <sup>Note 2</sup> | Inactive <sup>Note 2</sup> | Inactive <sup>Note 2</sup> |

- Notes 1.** While the on-chip peripheral I/O is accessed, the address the on-chip peripheral I/O accesses is also output to the external address bus.
- 2.** The  $\overline{\text{WAIT}}$  pin does not input any signal during this operation.

### 18.3 Memory Block Function

The lower 4 MB of the 64 MB memory space is reserved for external memory expansion and is divided into memory blocks of 2 MB and 2 MB. The bus width and programmable wait function can be independently specified for each block.



#### 18.3.1 Chip select control function

Of the 64 MB address space (linear), the lower 4 MB (0000000H to 03FFFFFFH) has two chip select functions,  $\overline{CS0}$  and  $\overline{CS1}$ . The areas selected by  $\overline{CS0}$  and  $\overline{CS1}$  are fixed.

The memory area can be effectively used by dividing it into memory blocks using the chip select control function. The allocation of memory blocks is described below.

| Chip Select Signal | Area                         |
|--------------------|------------------------------|
| $\overline{CS0}$   | 0000000H to 01FFFFFFH (2 MB) |
| $\overline{CS1}$   | 0200000H to 03FFFFFFH (2 MB) |

### 18.4 Bus Cycle Type Control Function

In the  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A, SRAM, external ROM, and external I/O can be connected directly.

**(1) Bus cycle type configuration register 0 (BCT0)**

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

Reset sets this register to CCCCH.

- Cautions**
1. Do not access an external memory area until the initial setting of the BCT0 register is complete. However, it is possible to access external memory areas whose initialization settings are complete.
  2. The set contents of each register are invalid for the CSn space where operations are prohibited.

|                                |                         |     |                    |                         |     |    |   |   |
|--------------------------------|-------------------------|-----|--------------------|-------------------------|-----|----|---|---|
| After reset: CCCCH             |                         | R/W | Address: FFFFF480H |                         |     |    |   |   |
|                                | 15                      | 14  | 13                 | 12                      | 11  | 10 | 9 | 8 |
| BCT0                           | 1                       | 1   | 0                  | 0                       | 1   | 1  | 0 | 0 |
| $\overline{\text{CSn}}$ signal | 7                       | 6   | 5                  | 4                       | 3   | 2  | 1 | 0 |
|                                | ME1                     | 1   | 0                  | 0                       | ME0 | 1  | 0 | 0 |
| $\overline{\text{CSn}}$ signal | $\overline{\text{CS1}}$ |     |                    | $\overline{\text{CS0}}$ |     |    |   |   |

|                 |  |
|-----------------|--|
| ME <sub>n</sub> | Memory controller operation enable for each CS <sub>n</sub> space (n = 0, 1) |
| 0               | Operation disabled   |
| 1               | Operation enabled  |

**Caution** Be sure to set bits 0, 1, 4, 5, 8, 9, 12, and 13 to “0”, and set bits 2, 6, 10, 11, 14, and 15 to “1”. If they are set other than above, the operation is not guaranteed.

## 18.5 Bus Access

### 18.5.1 Number of access clocks

The number of base clocks (MIN. value) necessary for accessing each resource is as follows.

| Bus Cycle Configuration          |                      | Instruction Fetch<br>(Normal Access) | Instruction Fetch<br>(Branch) | Operand Data Access |
|----------------------------------|----------------------|--------------------------------------|-------------------------------|---------------------|
| Resource (Bus Width)             |                      |                                      |                               |                     |
| Internal ROM (32 bits)           |                      | 1                                    | 2                             | 5                   |
| Internal RAM (32 bits)           |                      | 1 <sup>Note 1</sup>                  | 1 <sup>Note 1</sup>           | 1                   |
| On-chip peripheral I/O (16 bits) |                      | –                                    | –                             | 3 <sup>Note 2</sup> |
| External memory<br>(16 bits)     | Separate bus mode    | 3 + n                                | 3 + n                         | 3 + n               |
|                                  | Multiplexed bus mode | 3 + n                                | 3 + n                         | 3 + n               |

**Notes 1.** This value is 2 if there is conflict with data access.

**2.** Depending on the set value of the VSWC register.

**Remarks 1.** Unit: Clock/access

**2.** n: Number of wait state inserted

**18.5.2 Bus sizing function**

The bus sizing function controls the data bus width for each CS space. The data bus width is specified by using the BSC register.

If a 16-bit bus width is specified, the lower 8 bits are used for even addresses and the higher 8 bits are used for odd addresses.

**(1) Bus size configuration register (BSC)**

This register controls the bus width of each CS space.

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

Reset sets this register to 5555H.

**Caution** Write to the BSC register after reset, and then do not change the set value. Also, when changing the initial values of the BSC register, do not access an external memory area until the settings are complete. However, it is possible to access external memory areas whose initialization settings are complete.

After reset: 5555H    R/W Address: FFFFF066H

|     |    |    |    |    |    |    |   |   |
|-----|----|----|----|----|----|----|---|---|
|     | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BSC | 0  | 1  | 0  | 1  | 0  | 1  | 0 | 1 |

|                                |   |   |   |   |   |      |   |      |
|--------------------------------|---|---|---|---|---|------|---|------|
| $\overline{\text{CSn}}$ signal | 7 | 6 | 5 | 4 | 3 | 2    | 1 | 0    |
|                                | 0 | 1 | 0 | 1 | 0 | BS10 | 0 | BS00 |

|                                |     |  |     |  |
|--------------------------------|-----|--|-----|--|
| $\overline{\text{CSn}}$ signal | CS1 |  | CS0 |  |
|--------------------------------|-----|--|-----|--|

|      |  |
|------|--|
| BSn0 | Specification of data bus width of each CSn space (n = 0, 1) |
| 0    | 8 bits   |
| 1    | 16 bits  |

**Caution** Be sure to set bits 1, 3, 5, 7, 9, 11, 13, and 15 to “0”, and set bits 4, 6, 8, 10, 12, and 14 to “1”. If they are set other than above, the operation is not guaranteed.

**18.5.3 Endian function**

The  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A support little endian.

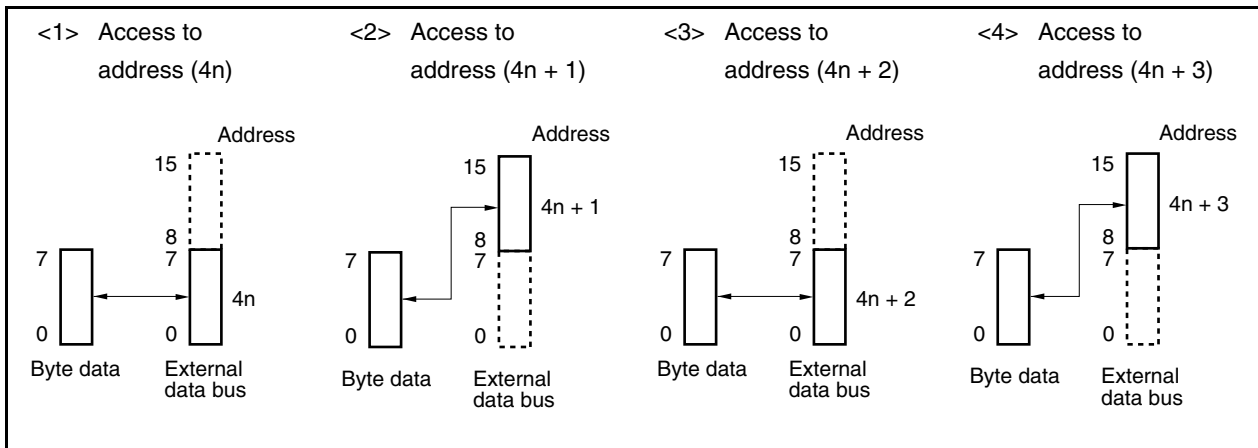
**18.5.4 Bus width**

The  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A access on-chip peripheral I/O and external memory in 8-bit, 16-bit, or 32-bit units. The following shows the operation for each type of access. All data is accessed in order starting from the lower order side.

**(1) Byte access (8 bits)**

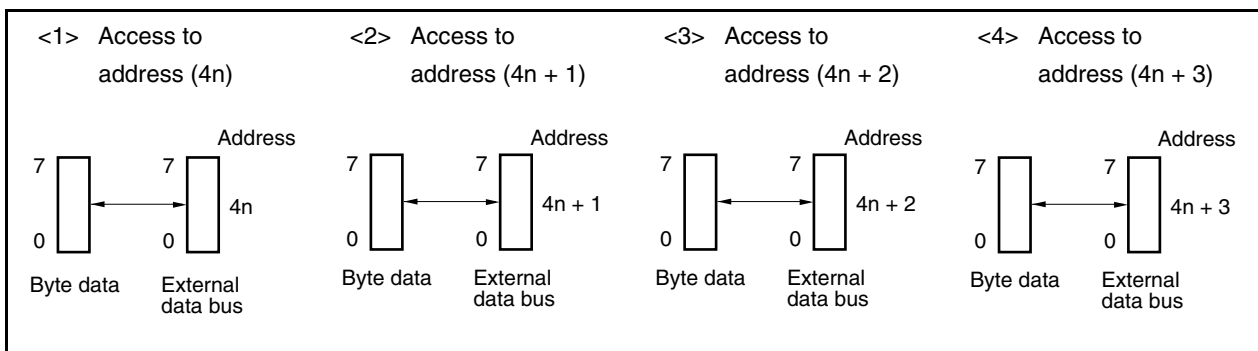
**(a) When the data bus width is 16 bits**

8-bit data is transmitted/received via a 16-bit bus. Therefore, if an even address is specified, the lower byte of the external data bus address is accessed. If an odd address is specified, the higher byte of the external data bus address is accessed.



**(b) When the data bus width is 8 bits**

8-bit data is transmitted/received via an 8-bit bus. Therefore, the specified even/odd address of the external data bus is accessed.

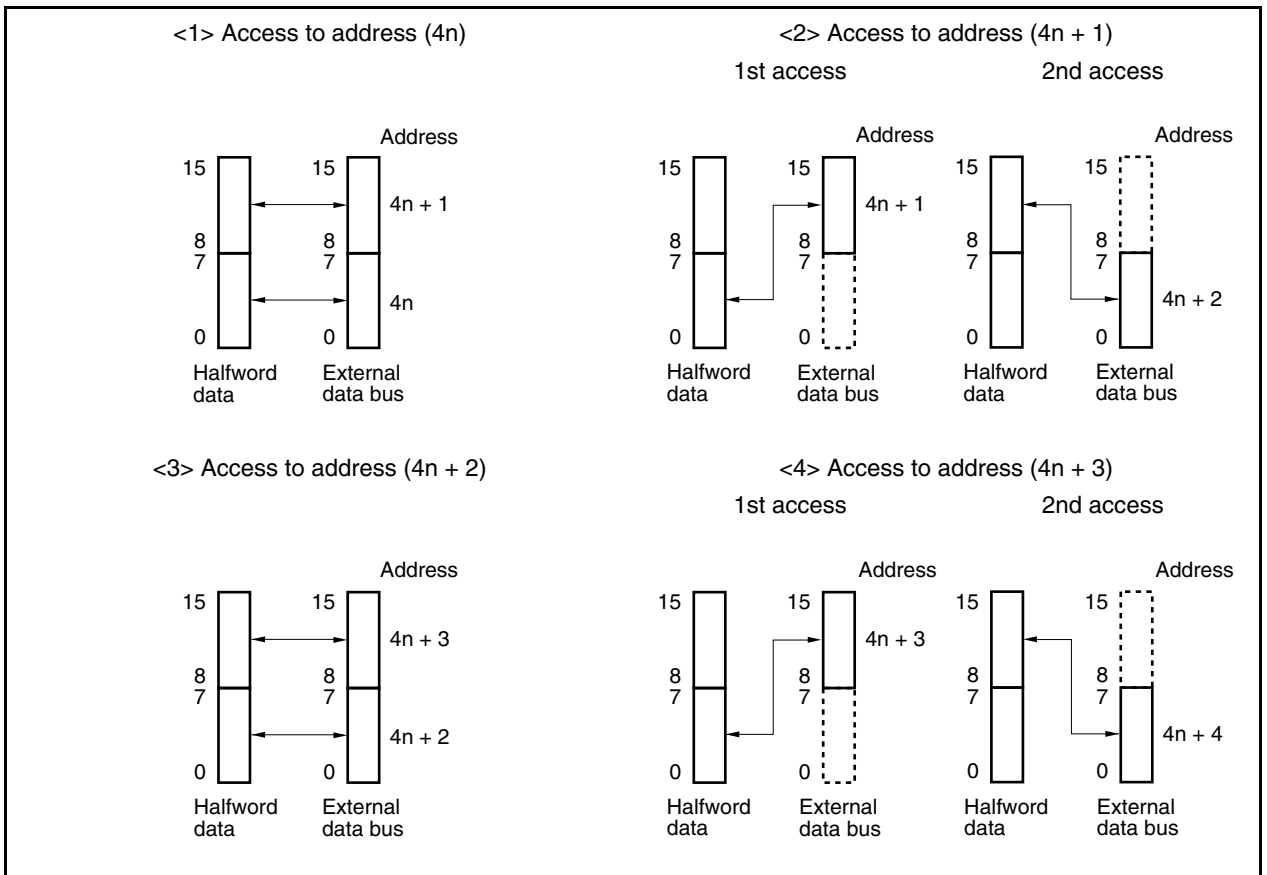




(2) Halfword access (16 bits)

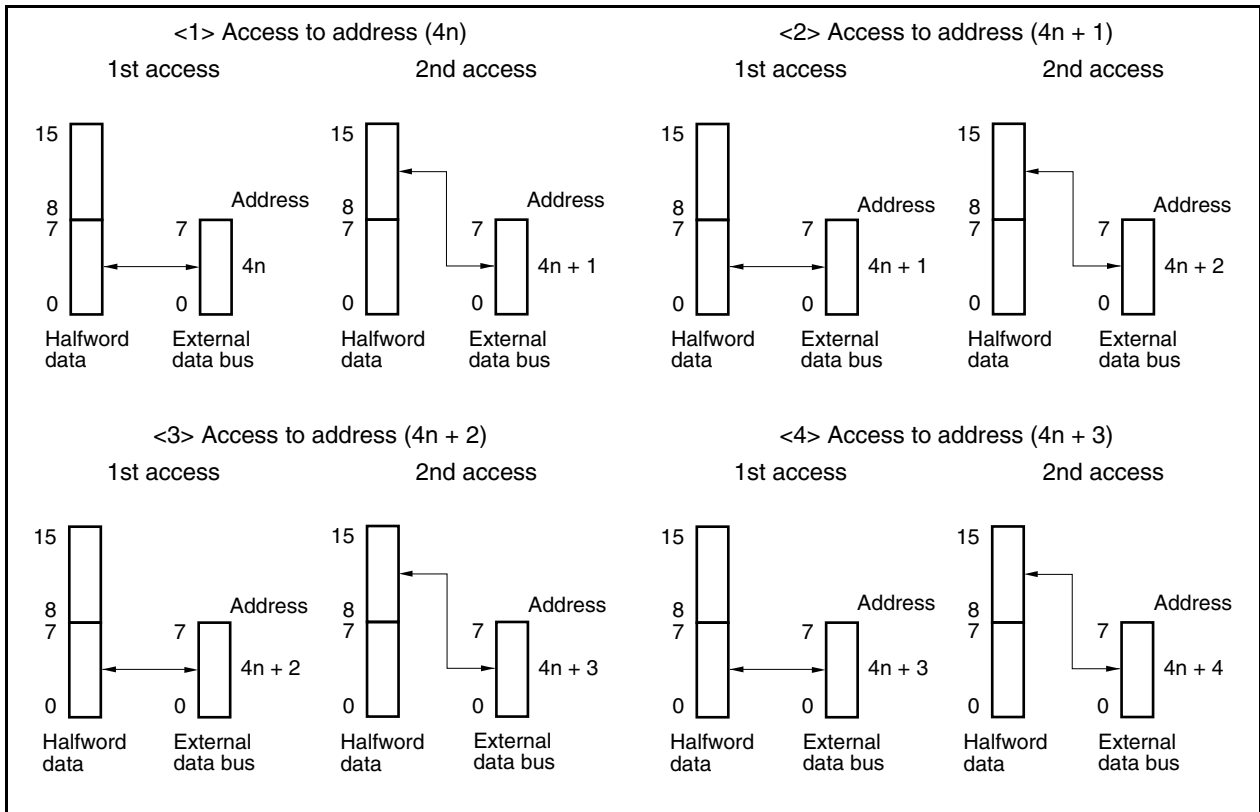
(a) When the data bus width is 16 bits

16-bit data is transmitted/received via a 16-bit bus. Therefore, if an even address is specified, the lower and higher bytes of the external data bus address are accessed at the same time. If an odd address is specified, the lower byte of the data is transmitted/received to/from an odd address via the higher byte of the external data bus address in the first access. In the second access, the higher byte of the data is transmitted/received to/from an even address via the lower 8 bits of the external data bus address.



**(b) When the data bus width is 8 bits**

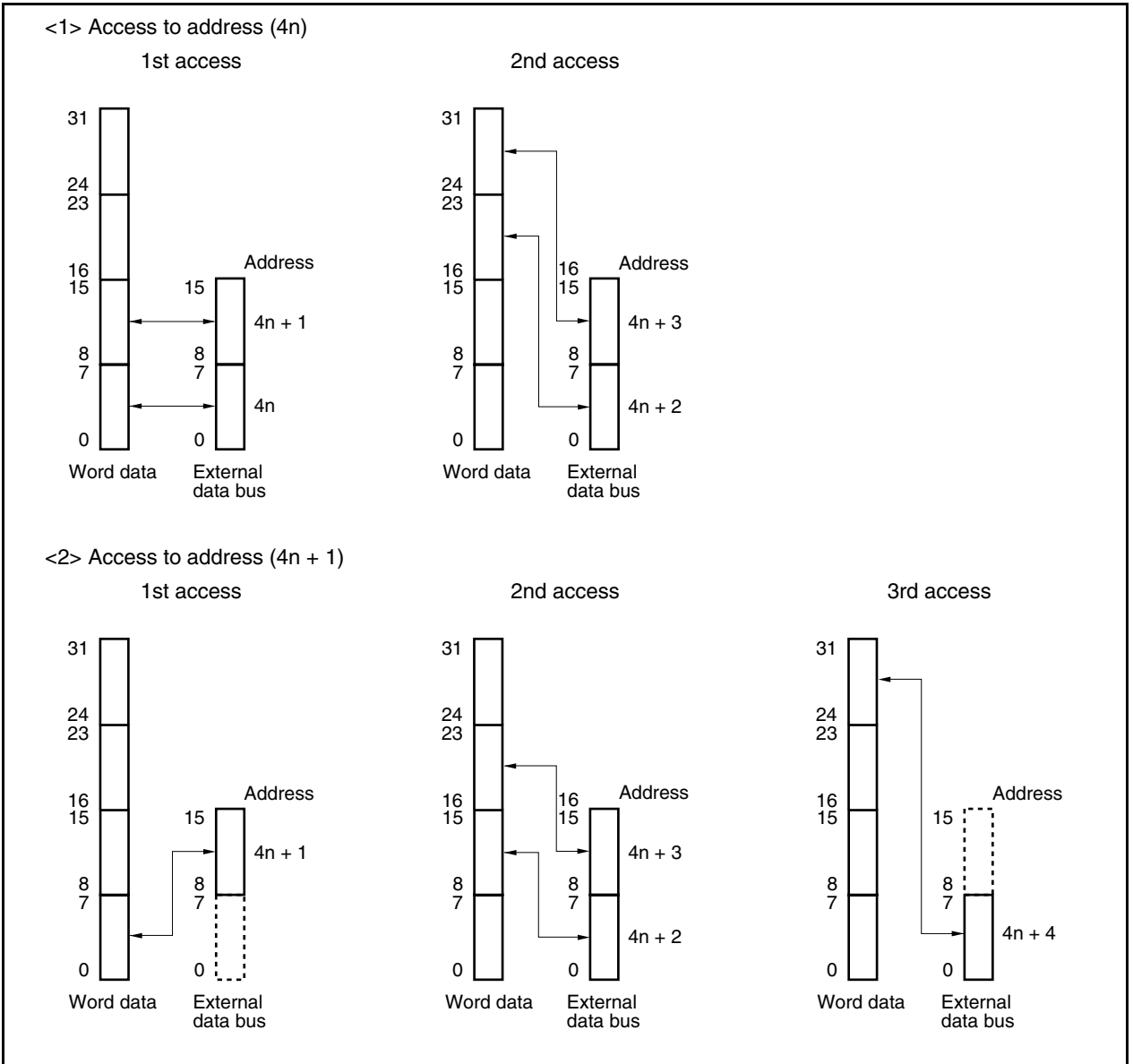
16-bit data is transmitted/received via an 8-bit bus. Therefore, the data is transmitted/received in two accesses. The lower/higher byte of the data is transmitted/received to/from the corresponding lower/higher byte of the external data bus address.



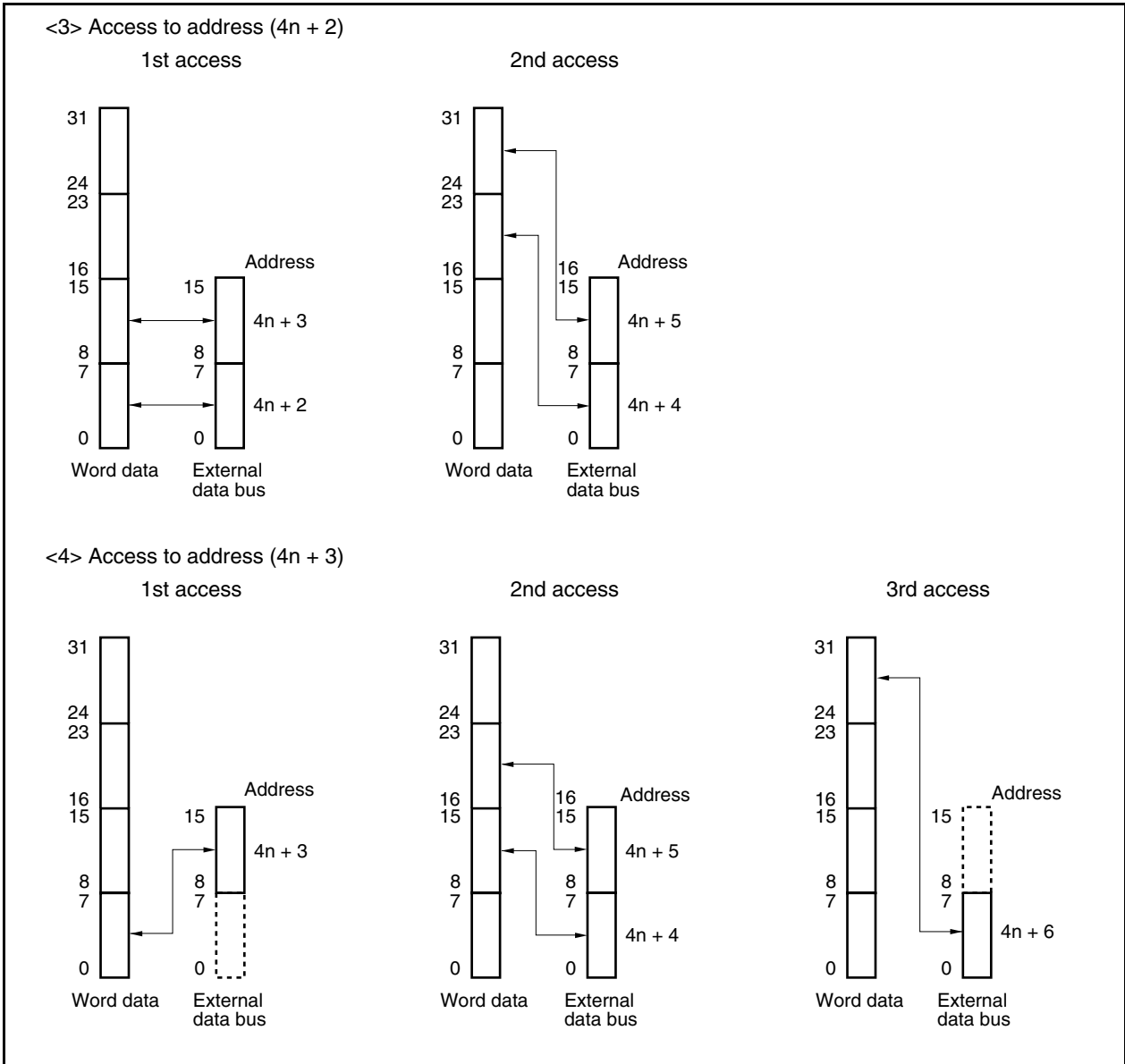
**(3) Word access (32 bits)**

**(a) When the data bus width is 16 bits (1/2)**

32-bit data is transmitted/received via a 16-bit bus. Therefore, if an even address is specified, the data is transmitted/received in two accesses in 16-bit units. If an odd address is specified, the lower quarter-word data is transmitted/received to/from the higher byte (first access), the middle halfword data is transmitted/received to/from the middle bytes (second access), and the upper quarter-word data is transmitted/received to/from the lower byte (third access), of the external data bus address.

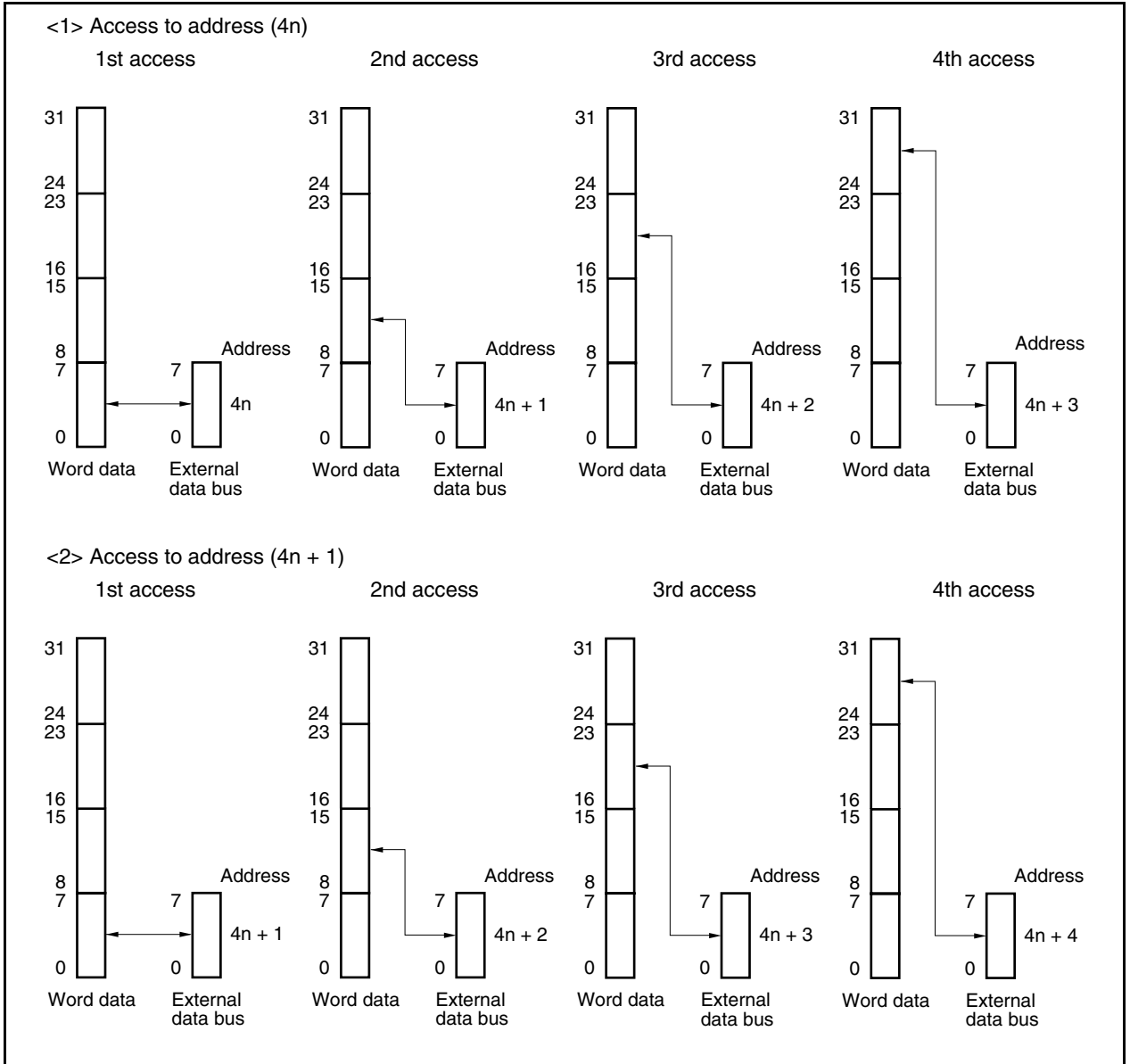


(a) When the data bus width is 16 bits (2/2)

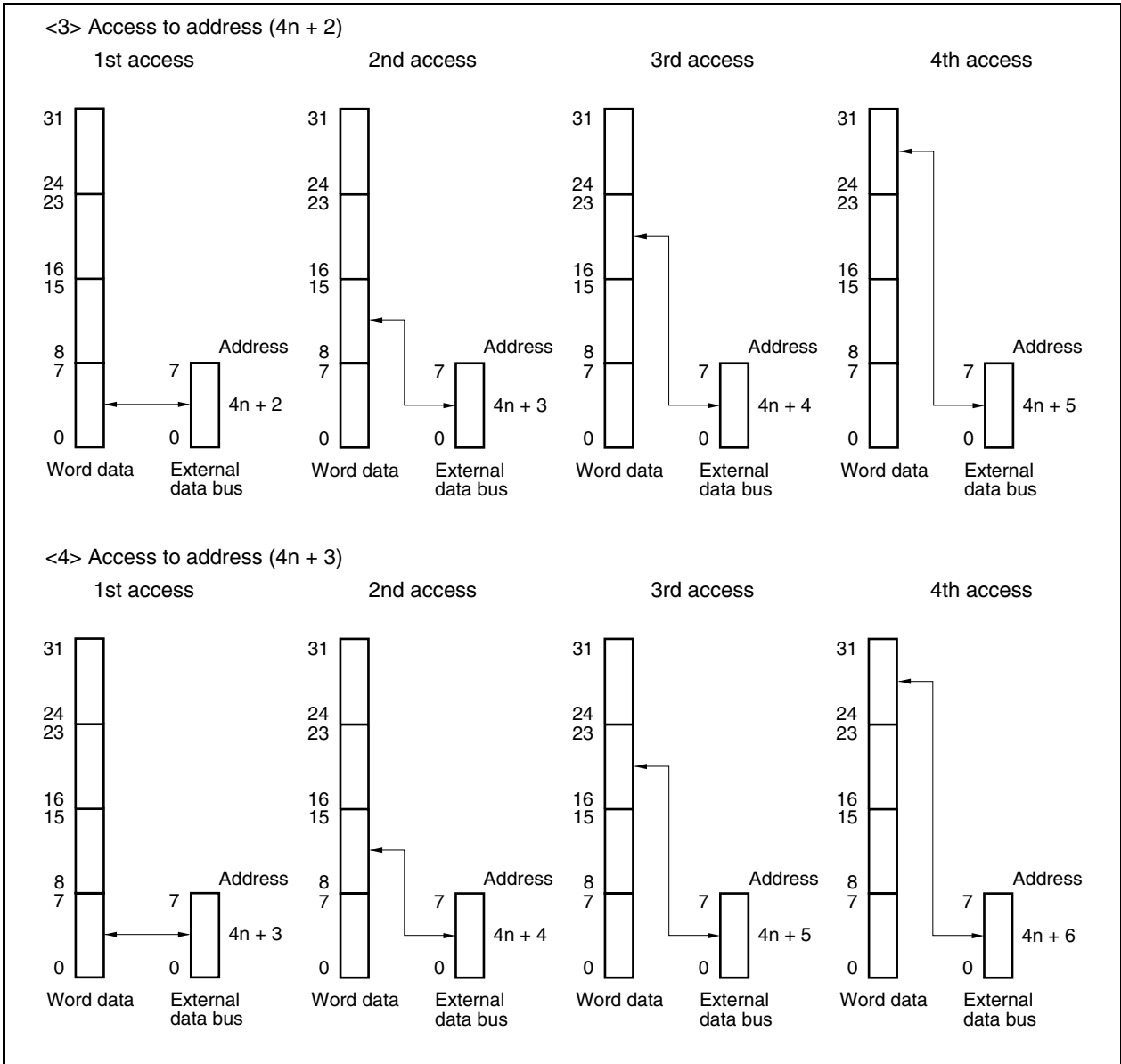


**(b) When the data bus width is 8 bits (1/2)**

32-bit data is transmitted/received via an 8-bit bus. Therefore, the data is transmitted/received in four accesses. The data is transmitted/received to/from the specified even/odd address of the external data bus.



(b) When the data bus width is 8 bits (2/2)



## 18.6 Wait Function

### 18.6.1 Programmable wait function

#### (1) Data wait control register 0 (DWC0)

To facilitate interfacing with a low-speed memory or I/O device and creating an interface circuit, it is possible to insert up to 7 data wait states in the starting bus cycle<sup>Note</sup> for each CS space.

The number of wait states can be specified by program using DWC0 register. Just after system reset, all blocks have 7 data wait states inserted.

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

Reset sets this register to 7777H.

**Note** SRAM read/write cycle

**Cautions** 1. The internal ROM and internal RAM areas are not subject to programmable waits and ordinarily no wait access is carried out.

The on-chip peripheral I/O area is not subject to programmable waits, with wait control performed by each on-chip peripheral function only.

2. Write to the DWC0 register after reset, and then do not change the set value. Also, when changing the initial values of the DWC0 register, do not access an external memory area until the settings are complete. However, it is possible to access external memory areas whose initialization settings are complete.

After reset: 7777H R/W Address: FFFFF484H



| DWn2 | DWn1 | DWn0 | Specification of number of wait states inserted in each CSn space (n = 0, 1) |
|------|------|------|--|
| 0    | 0    | 0    | Not inserted   |
| 0    | 0    | 1    | 1  |
| 0    | 1    | 0    | 2  |
| 0    | 1    | 1    | 3  |
| 1    | 0    | 0    | 4  |
| 1    | 0    | 1    | 5  |
| 1    | 1    | 0    | 6  |
| 1    | 1    | 1    | 7  |

**Caution** Be sure to set bits 3, 7, 11, and 15 to “0”, and set bits 8 to 10 and 12 to 14 to “1”. If they are set other than above, the operation is not guaranteed.



**(2) Address wait control register (AWC)**

This register is used to secure the setup and hold time for the address latch.

The AWC register can set an address setup wait state or address hold wait state that is to be inserted in each bus cycle. The address setup wait state is inserted before T1 state and the address hold wait state is inserted after T1 state.

Address setup wait state and address hold wait state insertion can be set with the AWC register for each CS space.

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

Reset sets this register to FFFFH.

- Cautions**
1. The internal ROM, internal RAM, and on-chip peripheral I/O areas are not subject to address setup wait state and address hold wait state insertion.
  2. During address setup wait state and address hold wait state, the  $\overline{\text{WAIT}}$  pin-based external wait function is disabled.
  3. Write to the AWC register after reset, and then do not change the set values. Also, when changing the initial values of the AWC register, do not access an external memory area until the settings are complete.

After reset: FFFFH    R/W    Address: FFFFF488H

|     |    |    |    |    |    |    |   |   |
|-----|----|----|----|----|----|----|---|---|
|     | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| AWC | 1  | 1  | 1  | 1  | 1  | 1  | 1 | 1 |

|                                |   |   |   |   |      |      |      |      |
|--------------------------------|---|---|---|---|------|------|------|------|
| $\overline{\text{CSn}}$ signal | 7 | 6 | 5 | 4 | 3    | 2    | 1    | 0    |
|                                | 1 | 1 | 1 | 1 | AHW1 | ASW1 | AHW0 | ASW0 |

$\overline{\text{CSn}}$  signal  
CS1

CS0

|      |  |
|------|--|
| AHWn | Specification of address hold wait state inserted in each CSn space (n = 0, 1) |
| 0    | Not inserted   |
| 1    | Inserted   |

|      |   |
|------|---|
| ASWn | Specification of address setup wait state inserted in each CSn space (n = 0, 1) |
| 0    | Not inserted  |
| 1    | Inserted  |

**Caution** Be sure to set bits 4 to 15 to “1”. If they are set to “0”, the operation is not guaranteed.

**18.6.2 External wait function**

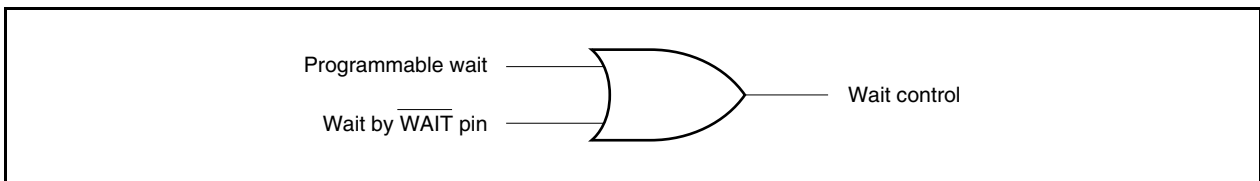
When a low-speed device or asynchronous system is connected, an arbitrary number of wait states can be inserted in the bus cycle by the external wait pin ( $\overline{\text{WAIT}}$ ) for synchronization with the external device.

Just as with programmable waits, accessing internal ROM, internal RAM, and on-chip peripheral I/O areas cannot be controlled by external waits.

The external  $\overline{\text{WAIT}}$  signal can be input asynchronously to the external bus clock frequency.

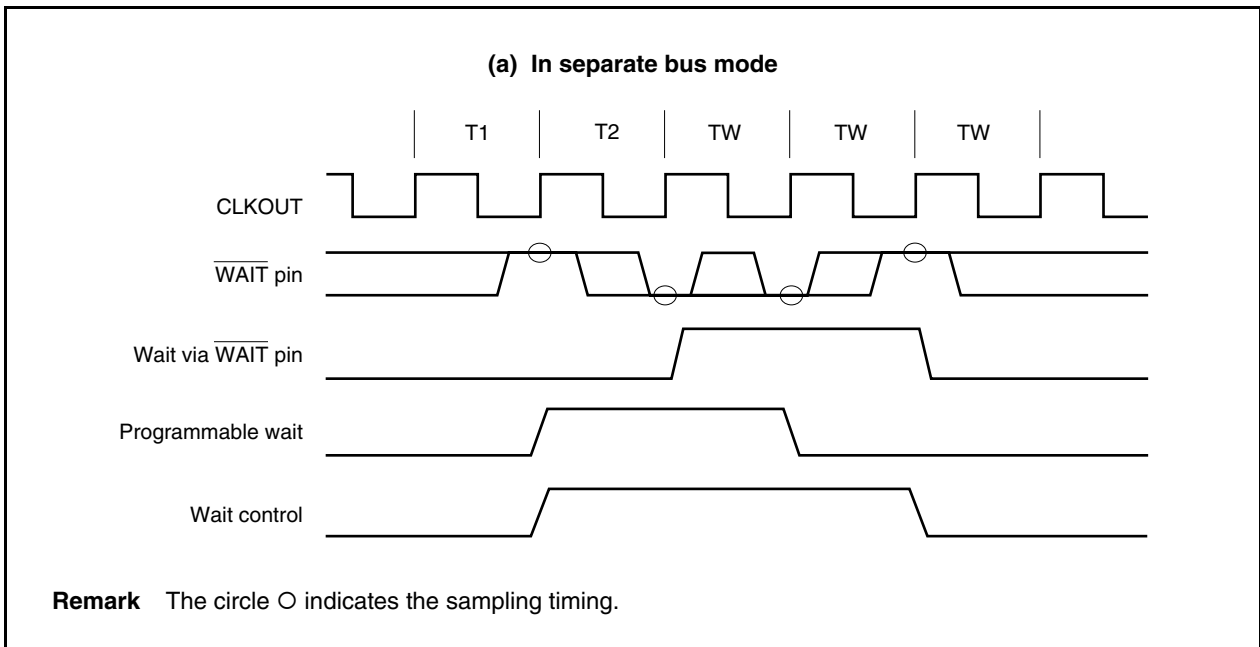
**18.6.3 Relationship between programmable wait and external wait**

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



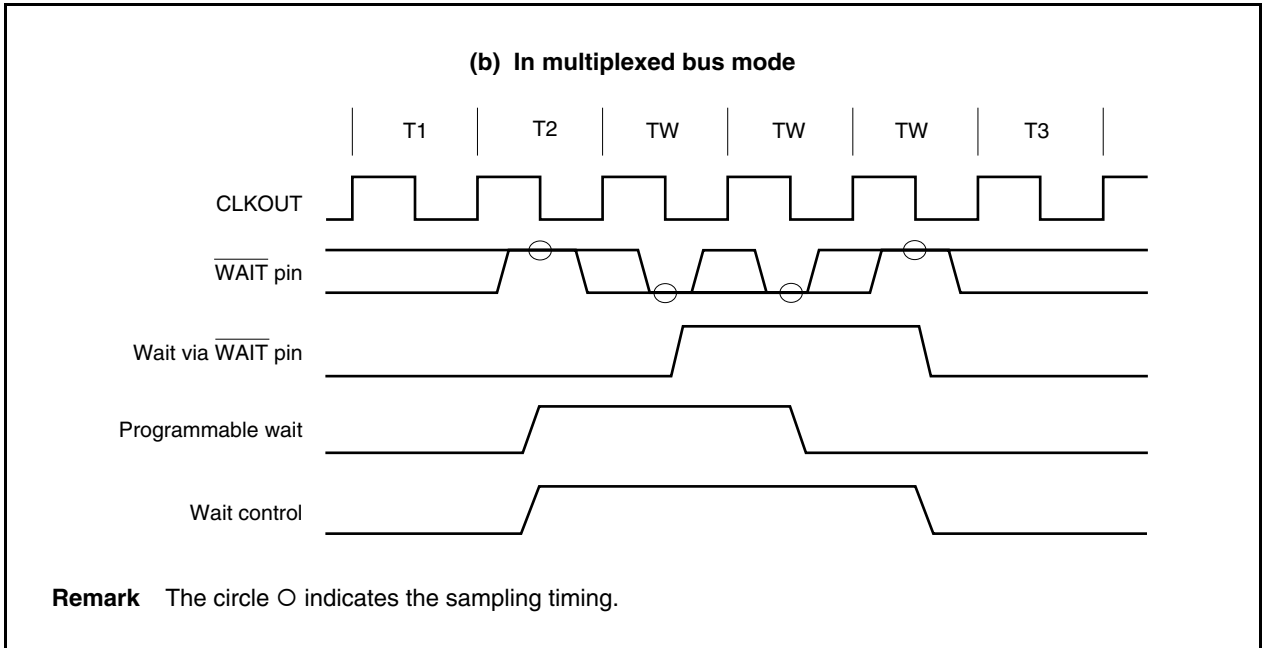
For example, if the timings of the programmable wait and the  $\overline{\text{WAIT}}$  pin signal are as illustrated below, three wait states will be inserted in the bus cycle. Wait states inserted via the  $\overline{\text{WAIT}}$  pin may be slower than the expected timing. In this case, use programmable waits to adjust the timing.

<R> **Figure 18-1. Example of Inserting Wait States (1/2)**



<R>

Figure 18-1. Example of Inserting Wait States (2/2)



**18.6.4 Bus cycles in which wait function is valid**

In the  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A, the number of waits can be specified for each memory block. The following shows the bus cycles in which the wait function is valid and the registers used for wait setting.

**Table 18-2. Bus Cycles in Which Wait Function Is Valid**

| Bus Cycle                               | Wait Type          | Programmable Wait Setting |              |                 | Wait by WAIT Pin |
|---|--------------------|---------------------------|--------------|-----------------|------------------|
|   |                    | Register                  | Bit          | Number of Waits |                  |
| SRAM, external ROM, external I/O cycles | Address setup wait | AWC                       | ASWn         | 0, 1            | × (invalid)      |
|   | Address hold wait  | AWC                       | AHWn         | 0, 1            | × (invalid)      |
|   | Data wait          | DWC0                      | DWn2 to DWn0 | 0 to 7          | √ (valid)        |

**Remark** n = 0, 1

### 18.7 Idle State Insertion Function

The idle state is inserted after a read cycle or a write cycle to the SRAM, external ROM, or external I/O.

#### (1) Bus cycle control register (BCC)

To facilitate interfacing with low-speed devices, an idle state (TI) can be inserted into the current bus cycle after the T2 state (after TW state if a data wait state is inserted) to secure the data output float delay time on memory read access for each CS space. The bus cycle following the T2 state (or TW state) starts after the idle state is inserted.

An idle state can be inserted after a write access by using the bus clock division control register (DVC).

The idle state insertion setting can be specified by program using the BCC register. Immediately after the system reset, idle state insertion is automatically programmed for all memory blocks. For the timing when an idle state is inserted, see **18.8 Bus Timing**.

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

Reset sets this register to AAAAH.

- Cautions**
1. The internal ROM, internal RAM, and on-chip peripheral I/O areas are not subject to idle state insertion.
  2. Write to the BCC register after reset, and then do not change the set values. Also, when changing the initial values of the BCC register, do not access an external memory area until the settings are complete. However, it is possible to access external memory areas whose initialization settings are complete.
  3. The chip select signal ( $\overline{CSn}$ ) does not become active in the idle state ( $n = 0, 1$ ).

After reset: AAAAH R/W Address: FFFFF48AH

|     |    |    |    |    |    |    |   |   |
|-----|----|----|----|----|----|----|---|---|
|     | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BCC | 1  | 0  | 1  | 0  | 1  | 0  | 1 | 0 |

|                         |   |   |   |   |      |   |      |   |
|-------------------------|---|---|---|---|------|---|------|---|
| $\overline{CSn}$ signal | 7 | 6 | 5 | 4 | 3    | 2 | 1    | 0 |
|                         | 1 | 0 | 1 | 0 | BC11 | 0 | BC01 | 0 |

$\overline{CSn}$  signal

$\overline{CS1}$

$\overline{CS0}$

| BCn1 | Specification of idle state inserted in each CSn space (n = 0, 1) |
|------|---|
| 0    | Not inserted  |
| 1    | Inserted  |

Insertion of an idle state can be specified for each CSn space after completion of a read cycle or a write cycle.  
 If the DVC.BCWI bit = 0, however, the idle state is inserted only after completion of a read cycle and not after completion of a write cycle.

**Caution** Be sure to set bits 0, 2, 4, 6, 8, 10, 12, and 14 to “0”, and set bits 5, 7, 9, 11, 13, and 15 to “1”. If they are set other than above, the operation is not guaranteed.

**(2) Bus clock division control register (DVC)**

The DVC register is used to specify insertion of an idle state (TI) after completion of a write cycle, and an external bus clock frequency.

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

Reset sets this register to 81H.

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

**2. Write to the DVC register after reset once (initial setting), and then do not change the set value. Also, do not access an external memory area until the initial setting of the DVC register is complete.**

**However, it is possible to access external memory areas whose initialization settings are complete.**

After reset: 81H R/W Address: FFFFF48EH

|     |      |   |   |   |   |   |      |      |
|-----|------|---|---|---|---|---|------|------|
|     | 7    | 6 | 5 | 4 | 3 | 2 | 1    | 0    |
| DVC | BCWI | 0 | 0 | 0 | 0 | 0 | DVC1 | DVC0 |

|      |   |
|------|---|
| BCWI | Specification of idle state inserted after write cycle ends |
| 0    | Not inserted  |
| 1    | Inserted (only when BCC.BCn1 bit = 1)                       |

|      |      |   |
|------|------|---|
| DVC1 | DVC0 | Specification of external bus clock frequency ( $f_{BUS}$ ) |
| 0    | 0    | $f_{CLK}/1$ <sup>Note</sup>                                 |
| 0    | 1    | $f_{CLK}/2$   |
| 1    | 0    | Setting prohibited  |
| 1    | 1    | $f_{CLK}/4$   |

**Note** Can be set only when  $f_{CLK} \leq 32$  MHz.  
Setting prohibited when  $32 \text{ MHz} < f_{CLK} \leq 64$  MHz

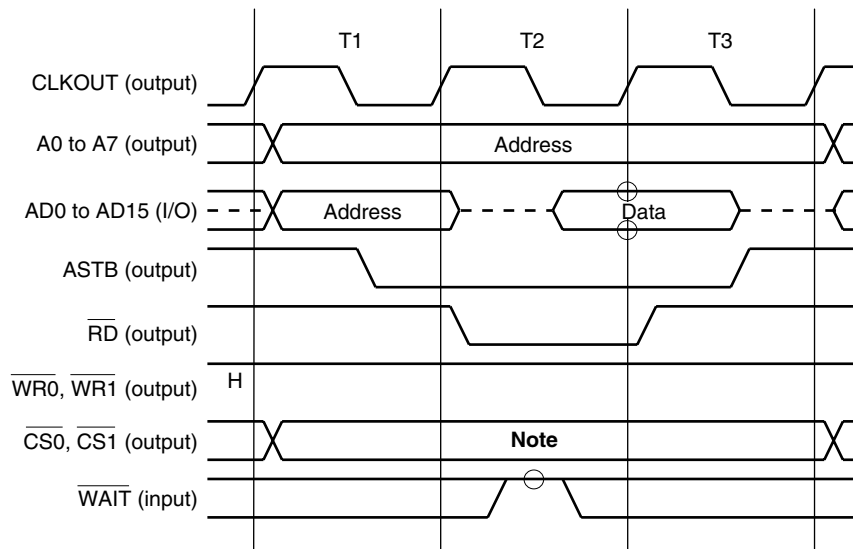
**Cautions 1. Be sure to set the CLKOUT pin in the port mode before changing the setting of the DVC1 and DVC0 bits. Changing the setting of the DVC1 and DVC0 bits while the alternate function (CLKOUT) is used is prohibited.**

**2. Set the external bus clock frequency ( $f_{BUS}$ ) in a range of  $16 \text{ MHz} \leq f_{BUS} \leq 32 \text{ MHz}$ .**

**3. Be sure to set bits 2 to 6 to "0". If they are set to "1", the operation is not guaranteed.**

## 18.8 Bus Timing

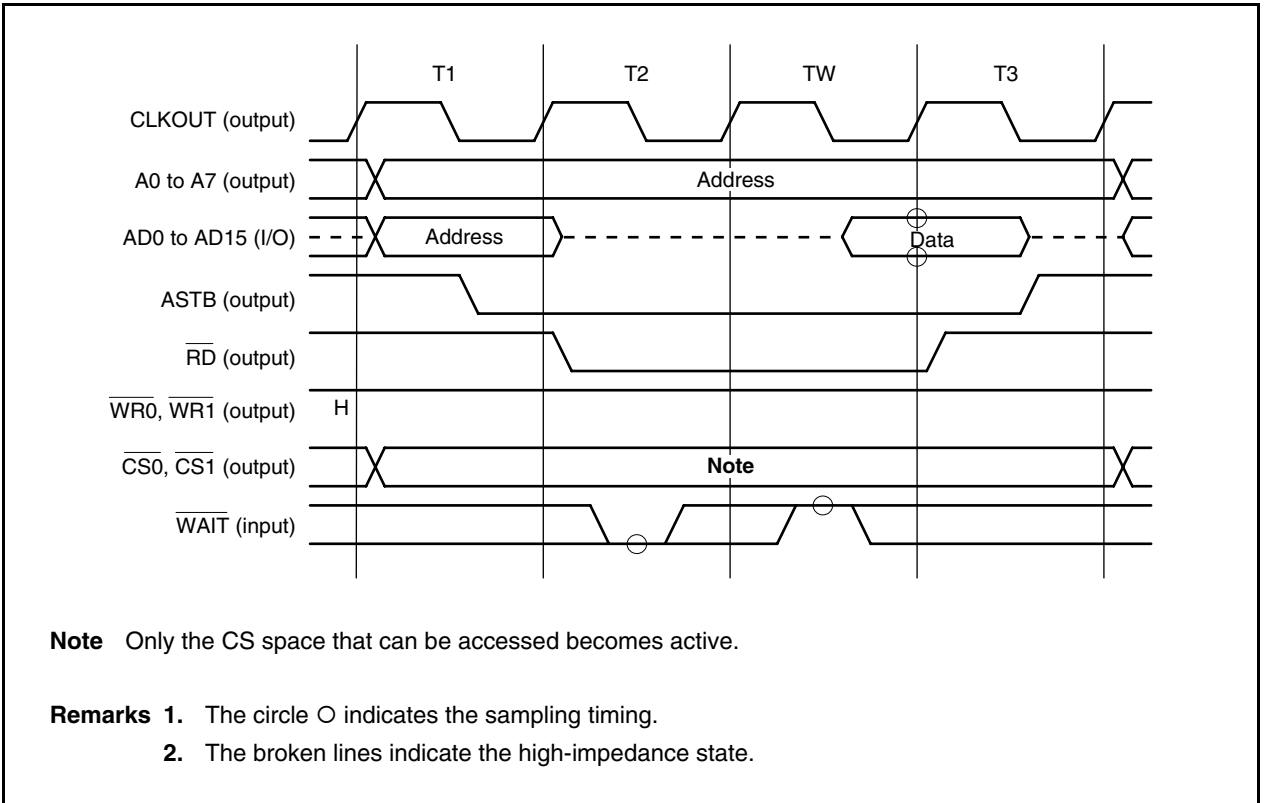
### (1) Read cycle (basic cycle)



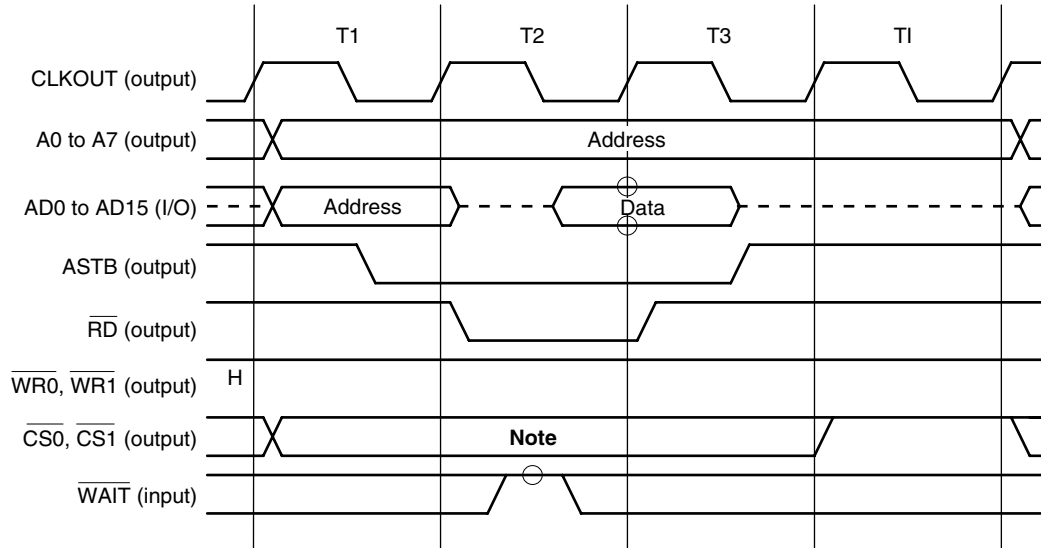
**Note** Only the CS space that can be accessed becomes active.

- Remarks**
1. The circle O indicates the sampling timing.
  2. The broken lines indicate the high-impedance state.

(2) Read cycle (when data wait state (1 wait) insertion)



(3) Read cycle (when idle state insertion)

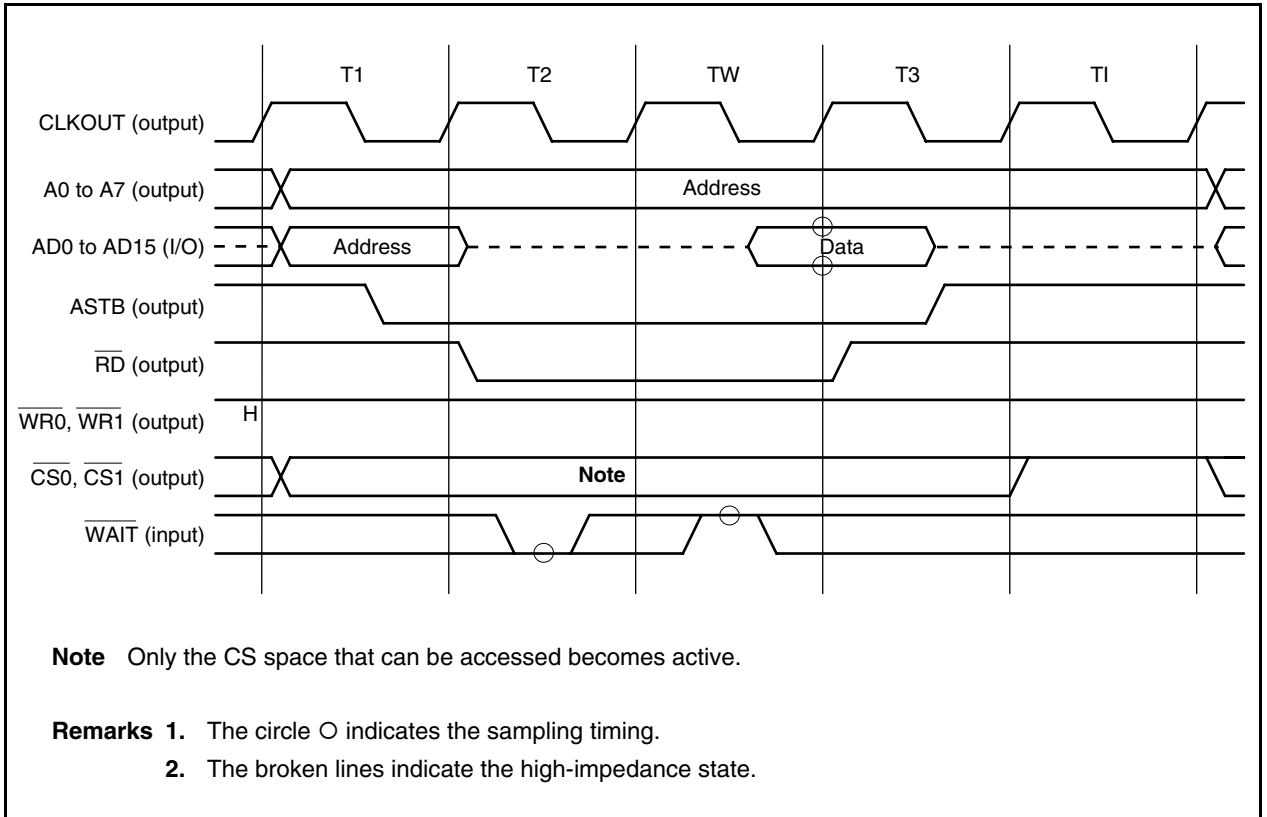


**Note** Only the CS space that can be accessed becomes active.

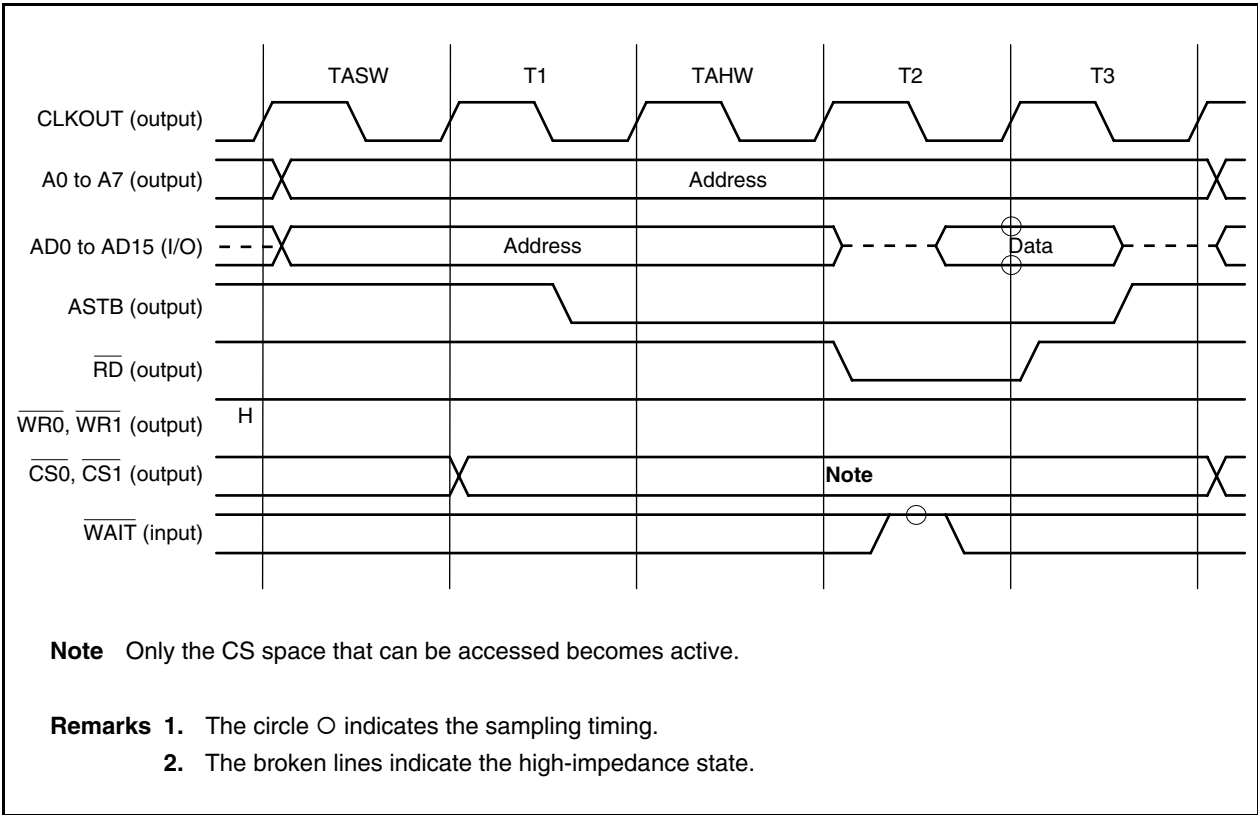
- Remarks**
1. The circle ○ indicates the sampling timing.
  2. The broken lines indicate the high-impedance state.



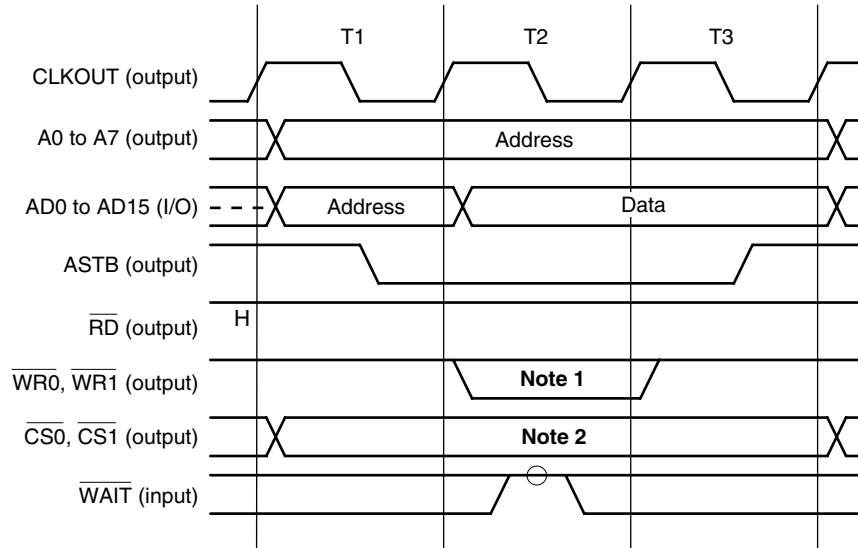
(4) Read cycle (when data wait state (1 wait), idle state insertion)



(5) Read cycle (when address setup wait state, address hold wait state insertion)



(6) Write cycle (basic cycle)



**Notes 1.** The levels of these signals are as follows, depending on the access data bus width.

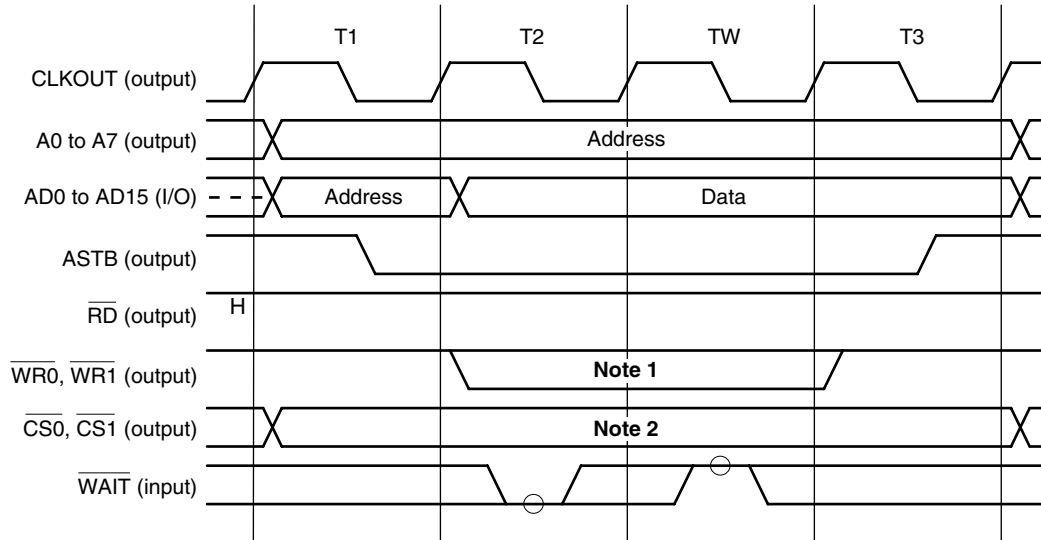
| Access Data Bus Width | $\overline{WR1}$ | $\overline{WR0}$ |
|-----------------------|------------------|------------------|
| 16 bits               | Low level        | Low level        |
| 8 bits                | High level       | Low level        |

**2.** Only the CS space that can be accessed becomes active.

**Remarks 1.** The circle O indicates the sampling timing.

**2.** The broken lines indicate the high-impedance state.

(7) Write cycle (when data wait state (1 wait) insertion)



**Notes 1.** The levels of these signals are as follows, depending on the access data bus width.

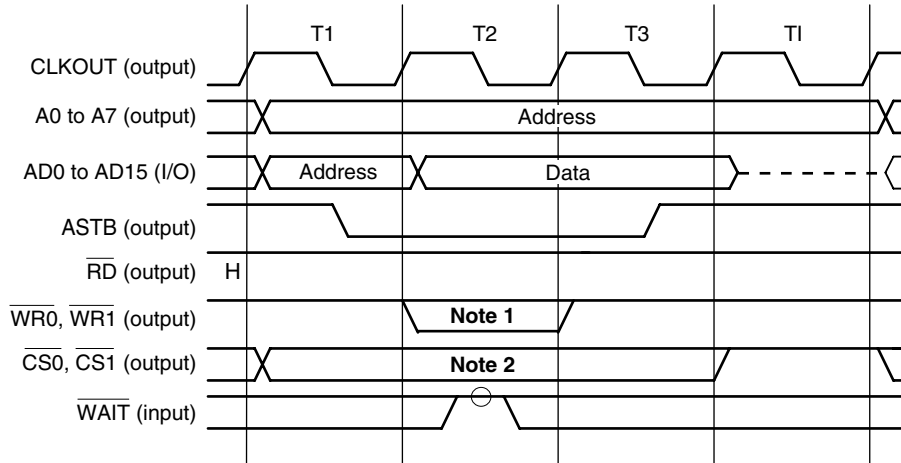
| Access Data Bus Width | $\overline{WR1}$ | $\overline{WR0}$ |
|-----------------------|------------------|------------------|
| 16 bits               | Low level        | Low level        |
| 8 bits                | High level       | Low level        |

**2.** Only the CS space that can be accessed becomes active.

**Remarks 1.** The circle ○ indicates the sampling timing.

**2.** The broken lines indicate the high-impedance state.

**(8) Write cycle (when idle state insertion)**



**Notes 1.** The levels of these signals are as follows, depending on the access data bus width.

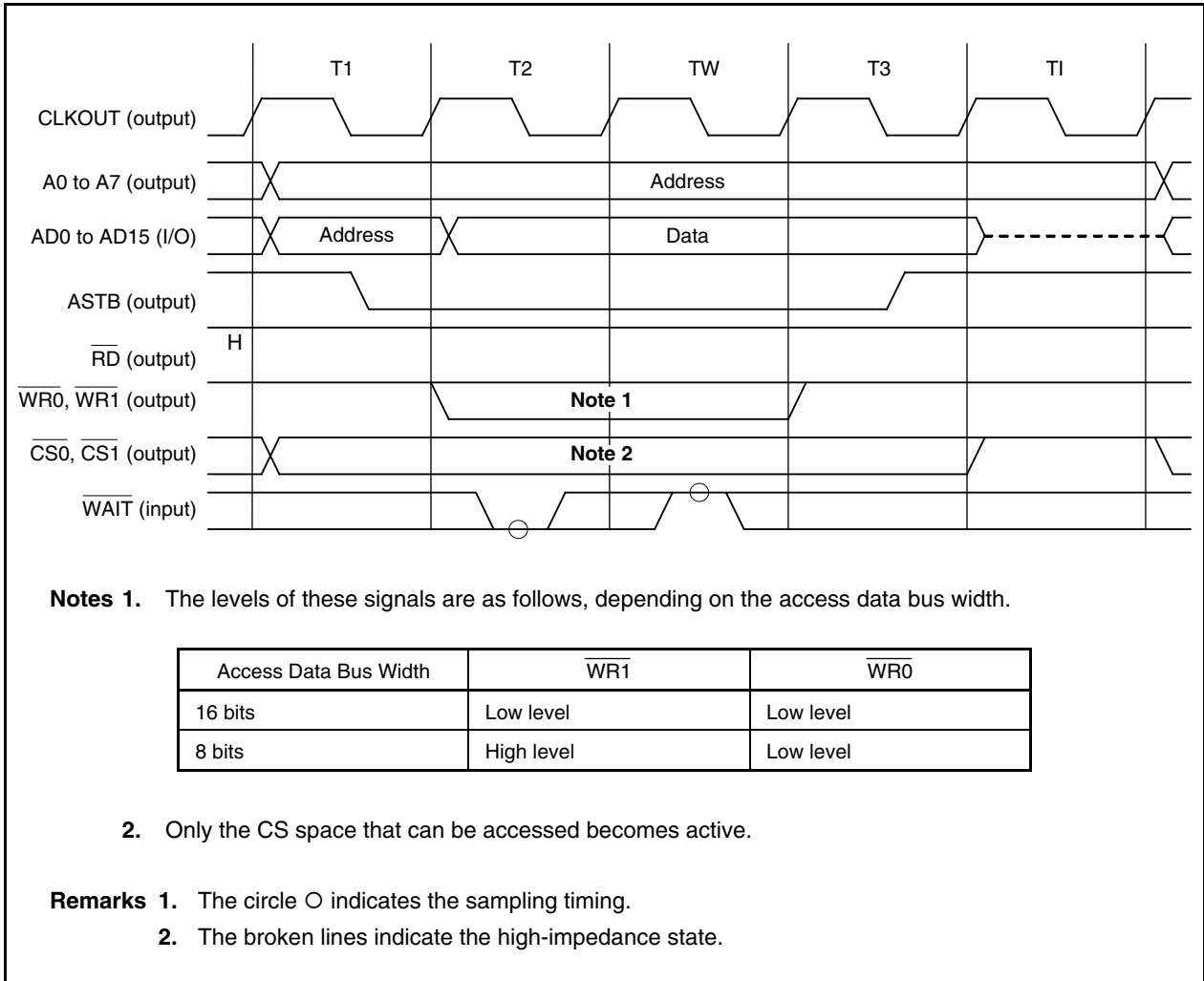
| Access Data Bus Width | $\overline{WR1}$ | $\overline{WR0}$ |
|-----------------------|------------------|------------------|
| 16 bits               | Low level        | Low level        |
| 8 bits                | High level       | Low level        |

**2.** Only the CS space that can be accessed becomes active.

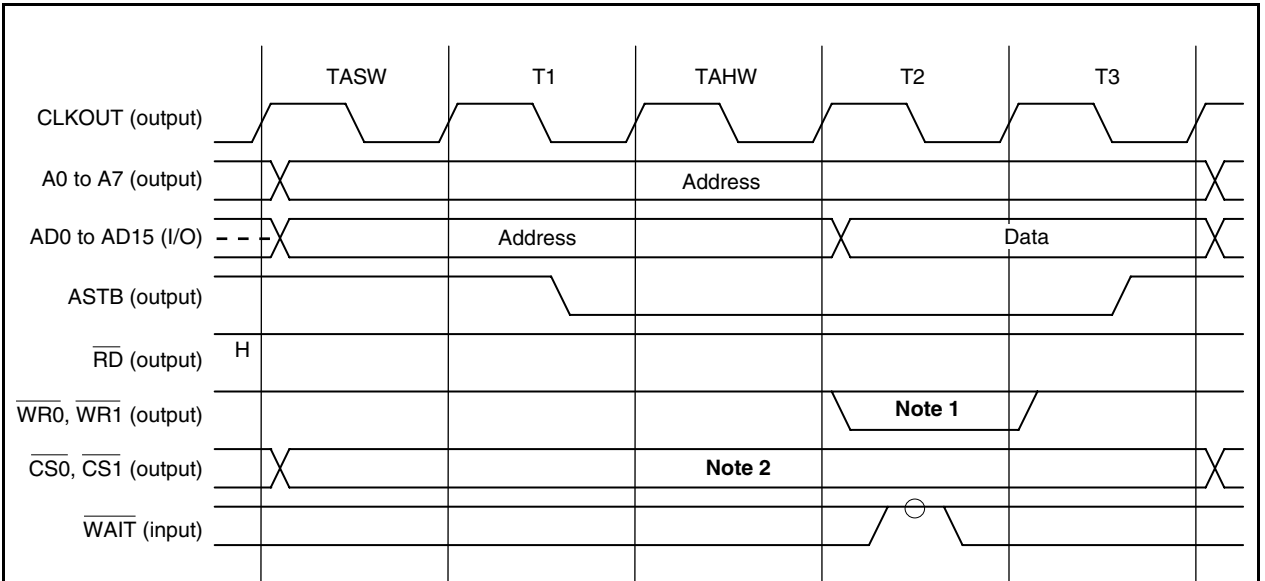
**Remarks 1.** The circle ○ indicates the sampling timing.

**2.** The broken lines indicate the high-impedance state.

(9) Write cycle (when data wait state (1 wait), idle state insertion)



(10) Write cycle (when address setup wait state, address hold wait state insertion)



**Notes 1.** The levels of these signals are as follows, depending on the access data bus width.

| Access Data Bus Width | $\overline{WR1}$ | $\overline{WR0}$ |
|-----------------------|------------------|------------------|
| 16 bits               | Low level        | Low level        |
| 8 bits                | High level       | Low level        |

**2.** Only the CS space that can be accessed becomes active.

**Remarks 1.** The circle ○ indicates the sampling timing.

**2.** The broken lines indicate the high-impedance state.

## 18.9 Bus Priority Order

There are two external bus cycles: instruction fetch and operand data access.

In order of priority, operand data access is the higher and instruction fetch is the lower.

However, an instruction fetch may be inserted between a read access and write access during a read modify write access.

**Table 18-3. Bus Priority Order**

| Priority Order | External Bus Cycle  | Bus Master |
|----------------|---------------------|------------|
| High           | Operand data access | CPU        |
| Low            | Instruction fetch   | CPU        |

## 18.10 Boundary Operation Conditions

### 18.10.1 Program space

Branching to the on-chip peripheral I/O area is prohibited. If the above is performed, undefined data is fetched, and fetching from the external memory is not performed.

### 18.10.2 Data space

The  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A are provided with an address misalign function.

Through this function, data can be allocated to all addresses, regardless of the data format (word or halfword). In the case of word data and halfword data, however, the bus cycle will be generated at least twice if data is not aligned to the boundary, which causes the bus efficiency to drop.

**(1) In the case of halfword-length data access**

When the address's LSB is 1, a byte-length bus cycle will be generated 2 times.

**(2) In the case of word-length data access**

(a) When the address's LSB is 1, bus cycles will be generated in the order of byte-length bus cycle, halfword-length bus cycle, and byte-length bus cycle.

(b) When the address's lower 2 bits are 10, a halfword-length bus cycle will be generated 2 times.



## CHAPTER 19 DMA FUNCTIONS (DMA CONTROLLER)

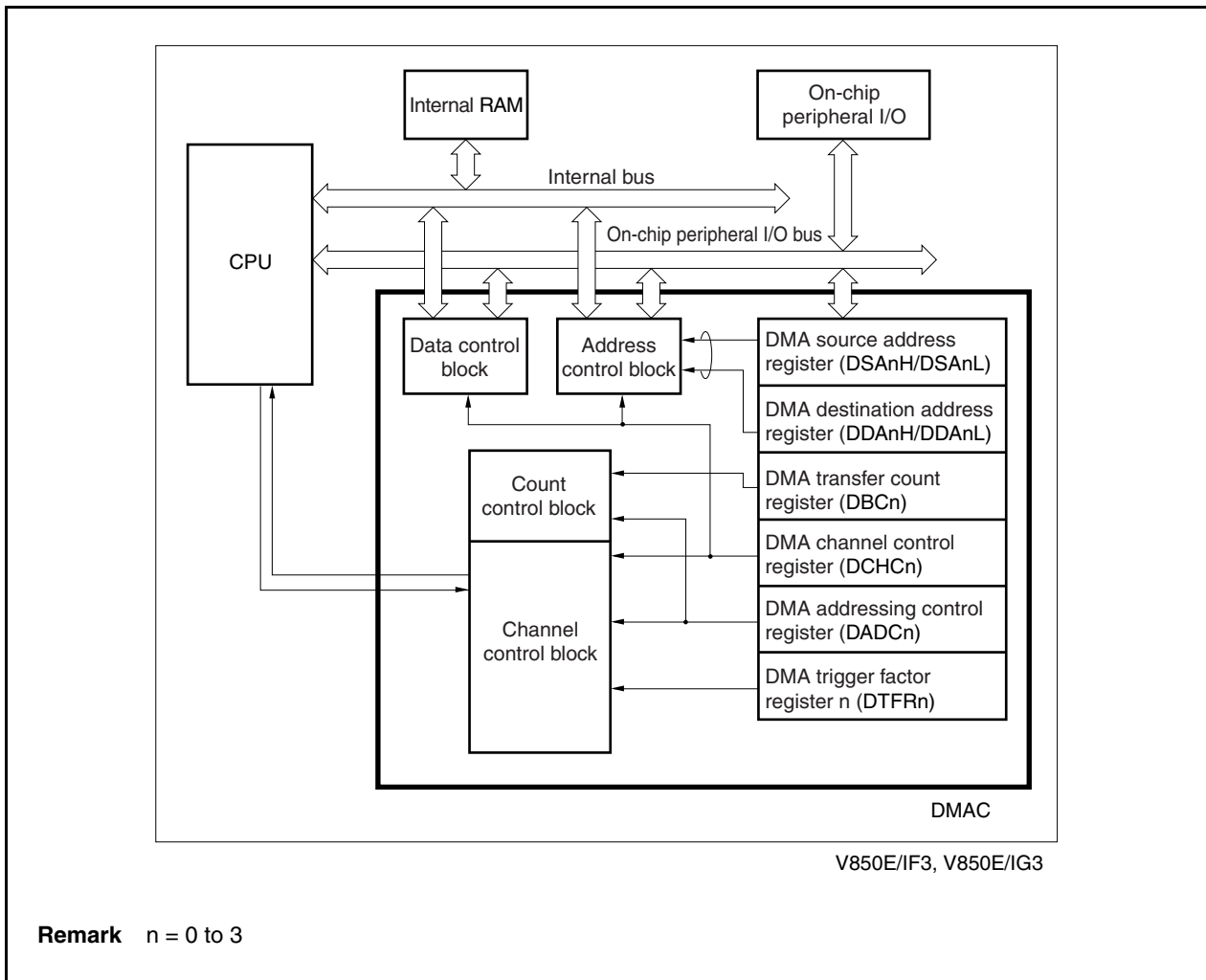
The V850E/IF3 and V850E/IG3 include a direct memory access (DMA) controller (DMAC) that executes and controls DMA transfer.

The DMAC controls data transfers between the internal memory and on-chip peripheral I/O, or between on-chip peripheral I/Os, based on requests by interrupts from the on-chip peripheral I/O (serial interface, timer, and A/D converter) or DMA requests issued by software triggers.

### 19.1 Features

- 4 independent DMA channels
- Transfer unit: 8/16 bits
- Maximum transfer count: 65536 ( $2^{16}$ )
- Transfer type: 2-cycle transfer
- Three transfer modes
  - Single transfer mode
  - Single-step transfer mode
  - Block transfer mode
- Transfer requests
  - Request by interrupts from on-chip peripheral I/O (serial interface, timer, A/D converter)
  - Requests by software trigger
- Transfer targets
  - Internal memory ↔ on-chip peripheral I/O
  - On-chip peripheral I/O ↔ on-chip peripheral I/O
- Next address setting function

19.2 Configuration



### 19.3 Control Registers

#### 19.3.1 DMA source address registers 0 to 3 (DSA0 to DSA3)

The DSA0 to DSA3 registers set the DMA transfer source address (28 bits) for DMA channel n (n = 0 to 3). These registers are divided into two 16-bit registers, DSAnH and DSAnL.

Since these registers are configured as 2-stage FIFO buffer registers consisting of the master register and slave register, a new transfer source address for DMA transfer can be specified during DMA transfer (see **19.8 Next Address Setting Function**). When setting the next address, the newly set value of the DSAn register is transferred to the slave register and becomes valid only when DMA transfer has been completed normally and the DCHCn.TCn bit is set to 1, or when the DCHCn.INITn bit is set to 1 (n = 0 to 3). However, the set value of the DSAn register is invalid even when the DCHCn.Enn bit is cleared to 0 to disable DMA transfer and then the DSAn register is set.

##### (1) DMA source address registers 0H to 3H (DSA0H to DSA3H)

The DSA0H to DSA3H registers can be read or written in 16-bit units.

Reset makes these registers undefined.

**Cautions 1. When setting an address of an on-chip peripheral I/O register for the source address, be sure to specify an address between FFFF000H and FFFFFFFH. An address of the on-chip peripheral I/O register image (3FFF000H to 3FFFFFFH) must not be specified.**

**2. Do not set the DSAnH register while DMA is suspended.**

|                        |                   |   |       |       |       |       |       |       |  |
|------------------------|-------------------|---|-------|-------|-------|-------|-------|-------|--|
| After reset: Undefined | R/W               | Address: DSA0H FFFFF082H, DSA1H FFFFF08AH,<br>DSA2H FFFFF092H, DSA3H FFFFF09AH  |       |       |       |       |       |       |  |
| DSAnH<br>(n = 0 to 3)  | 15                | 14  | 13    | 12    | 11    | 10    | 9     | 8     |  |
|                        | IRSn              | 0   | 0     | 0     | SAn27 | SAn26 | SAn25 | SAn24 |  |
|                        | 7                 | 6   | 5     | 4     | 3     | 2     | 1     | 0     |  |
|                        | SAn23             | SAn22   | SAn21 | SAn20 | SAn19 | SAn18 | SAn17 | SAn16 |  |
|                        | IRSn              | DMA transfer source specification   |       |       |       |       |       |       |  |
|                        | 0                 | On-chip peripheral I/O  |       |       |       |       |       |       |  |
|                        | 1                 | Internal RAM  |       |       |       |       |       |       |  |
|                        | SAn27 to<br>SAn16 | Set the DMA transfer source address (A27 to A16). During DMA transfer, these bits store the next DMA transfer source address. |       |       |       |       |       |       |  |

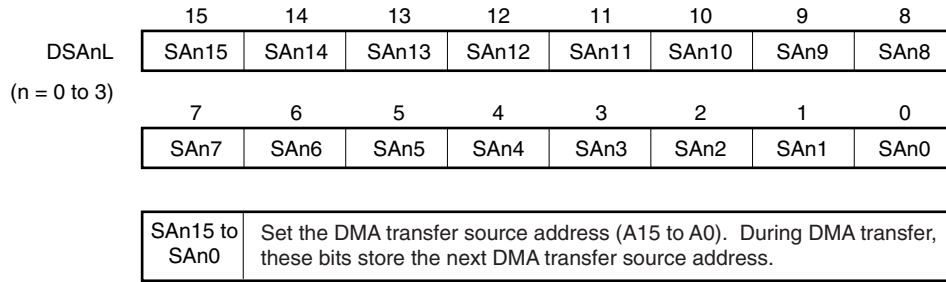
**Caution** Be sure to set bits 14 to 12 to “0”. If they are set to “1”, the operation is not guaranteed.

**(2) DMA source address registers 0L to 3L (DSA0L to DSA3L)**

The DSA0L to DSA3L registers can be read or written in 16-bit units.

Reset makes these registers undefined.

After reset: Undefined R/W Address: DSA0L FFFFF080H, DSA1L FFFFF088H,  
DSA2L FFFFF090H, DSA3L FFFFF098H



**19.3.2 DMA destination address registers 0 to 3 (DDA0 to DDA3)**

The DDA0 to DDA3 registers set the DMA transfer destination address (28 bits) for DMA channel n (n = 0 to 3). They are divided into two 16-bit registers, DDAnH and DDAnL.

Since these registers are configured as 2-stage FIFO buffer registers consisting of the master register and slave register, a new transfer destination address for DMA transfer can be specified during DMA transfer (see **19.8 Next Address Setting Function**). When setting the next address, the newly set value of the DDAn register is transferred to the slave register and becomes valid only when DMA transfer has been completed normally and the DCHCn.TCn bit is set to 1, or when the DCHCn.INITn bit is set to 1 (n = 0 to 3). However, the set value of the DDAn register is invalid even when the DCHCn.Enn bit is cleared to 0 to disable DMA transfer and then the DDAn register is set.

**(1) DMA destination address registers 0H to 3H (DDA0H to DDA3H)**

The DDA0H to DDA3H registers can be read or written in 16-bit units.

Reset makes these registers undefined.

- Cautions**
1. When setting an address of an on-chip peripheral I/O register for the destination address, be sure to specify an address between FFFF000H and FFFFFFFH. An address of the on-chip peripheral I/O register image (3FFF000H to 3FFFFFFH) must not be specified.
  2. Do not set the DDAnH register while DMA is suspended.

|                        |                   |   |       |       |       |       |       |       |  |
|------------------------|-------------------|---|-------|-------|-------|-------|-------|-------|--|
| After reset: Undefined | R/W               | Address: DDA0H FFFF086H, DDA1H FFFF08EH,<br>DDA2H FFFF096H, DDA3H FFFF09EH  |       |       |       |       |       |       |  |
| DDAnH<br>(n = 0 to 3)  | 15                | 14  | 13    | 12    | 11    | 10    | 9     | 8     |  |
|                        | IRAn              | 0   | 0     | 0     | DAn27 | DAn26 | DAn25 | DAn24 |  |
|                        | 7                 | 6   | 5     | 4     | 3     | 2     | 1     | 0     |  |
|                        | DAn23             | DAn22   | DAn21 | DAn20 | DAn19 | DAn18 | DAn17 | DAn16 |  |
|                        | IRAn              | DMA transfer destination specification  |       |       |       |       |       |       |  |
|                        | 0                 | On-chip peripheral I/O  |       |       |       |       |       |       |  |
|                        | 1                 | Internal RAM  |       |       |       |       |       |       |  |
|                        | DAn27 to<br>DAn16 | Set the DMA transfer destination address (A27 to A16). During DMA transfer, these bits store the next DMA transfer destination address. |       |       |       |       |       |       |  |

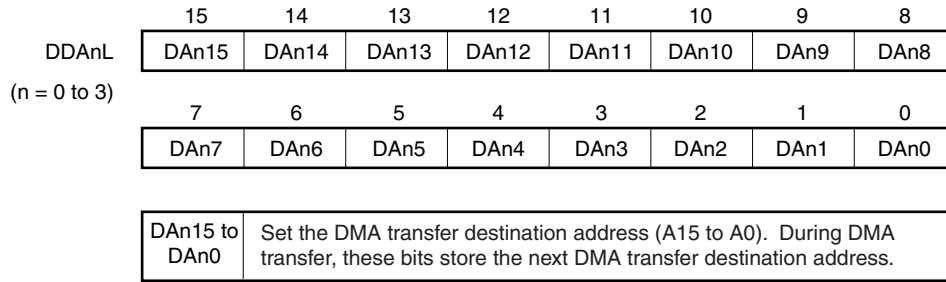
**Caution** Be sure to set bits 14 to 12 to “0”. If they are set to “1”, the operation is not guaranteed.

**(2) DMA destination address registers 0L to 3L (DDA0L to DDA3L)**

The DDA0L to DDA3L registers can be read or written in 16-bit units.

Reset makes these registers undefined.

After reset: Undefined    R/W    Address: DDA0L FFFFF084H, DDA1L FFFFF08CH,  
 DDA2L FFFFF094H, DDA3L FFFFF09CH



**19.3.3 DMA transfer count registers 0 to 3 (DBC0 to DBC3)**

The DBC0 to DBC3 registers are 16-bit registers that set the byte transfer count for DMA channel n (n = 0 to 3). These registers store the remaining transfer count during DMA transfer.

Since these registers are configured as 2-stage FIFO buffer registers consisting of the master register and slave register, a new DMA byte transfer count for DMA transfer can be specified during DMA transfer (see **19.8 Next Address Setting Function**). When setting the next address, the newly set value of the DBCn register is transferred to the slave register and becomes valid only when DMA transfer has been completed normally and the DCHCn.TCn bit is set to 1, or when the DCHCn.INITn bit is set to 1 (n = 0 to 3). However, the set value of the DBCn register is invalid even when the DCHCn.Enn bit is cleared to 0 to disable DMA transfer and then the DBCn register is set.

These registers are decremented by 1 for each transfer, and transfer ends when a borrow occurs.

These registers can be read or written in 16-bit units.

Reset makes these registers undefined.

**Caution** Do not set the DBCn register while DMA is suspended.

**Remark** If the DBCn register is read during DMA transfer after a terminal count has occurred without the register being overwritten, the value set immediately before the DMA transfer will be read out (0000H will not be read, even if DMA transfer has ended).

|                        |   |       |  |       |       |       |      |      |
|------------------------|---|-------|--|-------|-------|-------|------|------|
| After reset: Undefined |   | R/W   | Address: DBC0 FFFFF0C0H, DBC1 FFFFF0C2H,<br>DBC2 FFFFF0C4H, DBC3 FFFFF0C6H |       |       |       |      |      |
| DBCn<br>(n = 0 to 3)   | 15  | 14    | 13   | 12    | 11    | 10    | 9    | 8    |
|                        | BCn15   | BCn14 | BCn13  | BCn12 | BCn11 | BCn10 | BCn9 | BCn8 |
|                        | 7   | 6     | 5  | 4     | 3     | 2     | 1    | 0    |
|                        | BCn7  | BCn6  | BCn5   | BCn4  | BCn3  | BCn2  | BCn1 | BCn0 |
| BCn15 to BCn0          | Transfer count setting (store remaining transfer count during DMA transfer) |       |  |       |       |       |      |      |
| 0000H                  | Transfer count 1 or remaining transfer count                                |       |  |       |       |       |      |      |
| 0001H                  | Transfer count 2 or remaining transfer count                                |       |  |       |       |       |      |      |
| :                      | :   |       |  |       |       |       |      |      |
| FFFFH                  | Transfer count 65536 (2 <sup>16</sup> ) or remaining transfer count         |       |  |       |       |       |      |      |

### 19.3.4 DMA addressing control registers 0 to 3 (DADC0 to DADC3)

The DADC0 to DADC3 registers are 16-bit registers that control the DMA transfer mode for DMA channel n (n = 0 to 3). These registers cannot be accessed during a DMA operation.

These registers can be read or written in 16-bit units.

Reset sets these registers to 0000H.

**Cautions 1. The DSn0 bit sets how many bits of data are to be transferred.**

**If the transfer data size is set to 16 bits, transfer is always started from an address with the lowest bit of the address aligned to “0”. In this case, transfer cannot be started from an odd address.**

**2. Set the DADCn register when the target channel is in one of the following periods (the operation is not guaranteed if the register is set at any other time).**

- **Period from system reset to the generation of the first DMA transfer request**
- **Period from end of DMA transfer (after terminal count) to the generation of the next DMA transfer request**
- **Period from forced termination of DMA transfer (after the DCHCn.INITn bit was set to 1) to the generation of the next DMA transfer request**

After reset: 0000H    R/W    Address: DADC0 FFFF0D0H, DADC1 FFFF0D2H,  
DADC2 FFFF0D4H, DADC3 FFFF0D6H

|                       |       |       |       |       |      |      |   |   |
|-----------------------|-------|-------|-------|-------|------|------|---|---|
| DADCn<br>(n = 0 to 3) | 15    | 14    | 13    | 12    | 11   | 10   | 9 | 8 |
|                       | 0     | DSn0  | 0     | 0     | 0    | 0    | 0 | 0 |
|                       | 7     | 6     | 5     | 4     | 3    | 2    | 1 | 0 |
|                       | SADn1 | SADn0 | DADn1 | DADn0 | TMn1 | TMn0 | 0 | 0 |

|      |  |  |
|------|--|--|
| DSn0 | Setting of transfer data size for DMA transfer |  |
| 0    | 8 bits   |  |
| 1    | 16 bits  |  |

|       |       |   |
|-------|-------|---|
| SADn1 | SADn0 | Setting of count direction of transfer source address for DMA channel n |
| 0     | 0     | Increment   |
| 0     | 1     | Decrement   |
| 1     | 0     | Fixed   |
| 1     | 1     | Setting prohibited  |

|       |       |  |
|-------|-------|--|
| DADn1 | DADn0 | Setting of count direction of transfer destination address for DMA channel n |
| 0     | 0     | Increment  |
| 0     | 1     | Decrement  |
| 1     | 0     | Fixed  |
| 1     | 1     | Setting prohibited   |

|      |      |  |
|------|------|--|
| TMn1 | TMn0 | Setting of transfer mode during DMA transfer |
| 0    | 0    | Single transfer mode                         |
| 0    | 1    | Single-step transfer mode                    |
| 1    | 0    | Setting prohibited                           |
| 1    | 1    | Block transfer mode                          |

**Caution** Be sure to set bits 15, 13 to 8, 1, and 0 to “0”. If they are set to “1”, the operation is not guaranteed.



**19.3.5 DMA channel control registers 0 to 3 (DCHC0 to DCHC3)**

The DCHC0 to DCHC3 registers are 8-bit registers that control the DMA transfer operating mode for DMA channel  $n$  ( $n = 0$  to  $3$ ).

These registers can be read or written in 8-bit or 1-bit units. (However, bit 7 is read-only.)

Reset sets these registers to 00H.

- Cautions**
1. If transfer has been ended with the MLEn bit set to 1 and if the next transfer request is made by DMA transfer (hardware DMA) that is started by an interrupt from an on-chip peripheral I/O, the next transfer is executed with the TCn bit set to 1 (not automatically cleared to 0).
  2. Set the MLEn bit when the target channel is in one of the following periods (the operation is not guaranteed if the bit is set at any other time).
    - Period from system reset to the generation of the first DMA transfer request
    - Period from end of DMA transfer (after terminal count) to the generation of the next DMA transfer request
    - Period from forced termination of DMA transfer (after the INITn bit was set to 1) to the generation of the next DMA transfer request
  3. If DMA transfer is forcibly terminated in the last transfer cycle with the MLEn bit set to 1, the operation is performed in the same manner as when transfer is ended (the TCn bit is set to 1). (The Enn bit is cleared to 0 upon forced termination, regardless of the value of the MLEn bit.) In this case, the Enn bit must be set to 1 and the TCn bit must be read (cleared to 0) when the next DMA transfer request is made.
  4. Upon end of DMA transfer (during terminal count), each bit is updated with the Enn bit cleared to 0 and then the TCn bit set to 1. If the statuses of the TCn bit and Enn bit are polled and if the DCHCn register is read while each bit is updated, therefore, a value indicating the status “transfer not ended and prohibited” (TCn bit = 0 and Enn bit = 0) may be read (this is not abnormal).
  5. Be sure to read (clear to 0) the TCn bit after end of DMA transfer (after terminal count). The TCn bit does not have to be read (cleared to 0) only if the following two conditions are satisfied.
    - The MLEn bit is set to 1 upon end of DMA transfer (during terminal count).
    - The next DMA transfer (hardware DMA) start factor is an interrupt from the on-chip peripheral I/O (hardware DMA)

If even one of these conditions is not satisfied, be sure to read (clear to 0) the TCn bit before the next DMA transfer request is generated.

The operation cannot be guaranteed if the next DMA transfer request is generated while the TCn bit is set to 1.
  6. Do not set the Enn and STGn bits while DMA is suspended. Otherwise, the operation is not guaranteed.
  7. Do not end DMA transfer by clearing the Enn bit to 0.
  8. The relationship between the status of DMA transfer and the register value is as follows.
    - DMA transfer is in progress: TCn bit = 0, Enn bit = 1
    - DMA transfer is aborted: TCn bit = 0, Enn bit = 0
    - DMA transfer is stopped (ends): TCn bit = 1

After reset: 00H R/W Address: DCHC0 FFFF0E0H, DCHC1 FFFF0E2H,  
DCHC2 FFFF0E4H, DCHC3 FFFF0E6H

|                       |     |   |   |   |      |       |      |     |
|-----------------------|-----|---|---|---|------|-------|------|-----|
|                       | <7> | 6 | 5 | 4 | <3>  | <2>   | <1>  | <0> |
| DCHCn<br>(n = 0 to 3) | TCn | 0 | 0 | 0 | MLEn | INITn | STGn | Enn |

|  |   |
|--|---|
| TCn <sup>Note 1</sup>  | Status bit that indicates whether DMA transfer via DMA channel n has ended or not |
| 0  | DMA transfer has not ended.   |
| 1  | DMA transfer has ended.   |
| This bit is set (1) at the last DMA transfer and cleared (0) when it is read. If DMA transfer is executed to transfer data from the internal RAM, this bit is set (1) 4 clocks after end of the last transfer. |   |

|      |  |
|------|--|
| MLEn | When this bits is set (1) at DMA transfer end (at the terminal count output), the Enn bit is not cleared (0) and the DMA transfer enabled state is retained.<br>If the next DMA transfer start factor is input from an on-chip peripheral I/O (hardware DMA), the DMA transfer request is acknowledged even if the TCn bit is not read.<br>If the next DMA transfer start factor is input by setting the STGn bit to 1 (software DMA), the DMA transfer request is acknowledged if the TCn bit is read and cleared (0).<br>When this bit is cleared (0) at DMA transfer end (at the terminal count output), the Enn bit is cleared (0) and the DMA transfer disabled state is entered. At the next DMA transfer request, the TCn bit must be read and the Enn bit must be set (1). |
|------|--|

|                         |  |
|-------------------------|--|
| INITn <sup>Note 2</sup> | If this bit is set (1) during DMA transfer or while DMA is suspended, DMA transfer is forcibly terminated. |
|-------------------------|--|

|                        |   |
|------------------------|---|
| STGn <sup>Note 2</sup> | If this bit is set (1) in the DMA transfer enabled state (TCn bit = 0, Enn bit = 1), DMA transfer is started. |
|------------------------|---|

|   |   |
|---|---|
| Enn   | Setting whether DMA transfer via DMA channel n is to be enabled or disabled |
| 0   | DMA transfer disabled   |
| 1   | DMA transfer enabled  |
| <ul style="list-style-type: none"> <li>This bit is cleared (0) when DMA transfer ends. It is also cleared (0) when DMA transfer is forcibly terminated by setting (1) the INITn bit.</li> <li>If the Enn bit is set (1), do not set it until DMA transfer has been ended the number of times set by the DBCn register or DMA transfer is forcibly terminated by the INITn bit.</li> </ul> |   |

- Notes**
1. TCn bit is read-only.
  2. INITn and STGn bits are write-only. If these bits are read, 0 is read.

**Caution** Be sure to set bits 6 to 4 to “0”. If they are set to “1”, the operation is not guaranteed.

**19.3.6 DMA trigger factor registers 0 to 3 (DTFR0 to DTFR3)**

The DTFR0 to DTFR3 registers are 8-bit registers that control the DMA transfer start trigger via interrupt requests from on-chip peripheral I/O.

The interrupt requests set by these registers serve as DMA transfer start factors.

These registers can be read or written in 8-bit or 1-bit units. However, only bit 7 (DFn) can be read or written in 1-bit units; bits 5 to 0 (IFCn5 to IFCn0) can only be read or written in 8-bit units.

Reset sets these registers to 00H.

**Cautions 1. Be sure to follow the steps below when changing the DTFRn register settings.**

- When the values to be set to the IFCn5 to IFCn0 bits are not set to the IFCm5 to IFCm0 bits of another channel ( $n = 0$  to  $3$ ,  $m = 0$  to  $3$ ,  $n \neq m$ )
    - <1> Follow steps <3> to <5> when the DCHCn.Enn bit is cleared to 0, and follow steps <2> to <5> when the Enn bit is set to 1.
    - <2> Stop the DMA<sub>n</sub> operation of the channel to be rewritten (DCHCn.INITn bit = 1).
    - <3> Change the DTFRn register settings. (Be sure to set DFn bit = 0 and change the settings in the 8-bit manipulation.)
    - <4> To clear a DMA transfer request, clear the DMA transfer request flag (DTFRn.DFn) to 0.
    - <5> Enable the DMA<sub>n</sub> operation (Enn bit = 1).
  - When the values to be set to the IFCn5 to IFCn0 bits are set to the IFCm5 to IFCm0 bits of another channel ( $n = 0$  to  $3$ ,  $m = 0$  to  $3$ ,  $n \neq m$ )
    - <1> Follow steps <4> to <6> when the DCHCn.Enn bit is cleared to 0, and follow steps <2> to <6> when the Enn bit is set to 1.
    - <2> Stop the DMA<sub>n</sub> operation of the channel to be rewritten (DCHCn.INITn bit = 1).
    - <3> Stop the DMA<sub>m</sub> operation of the channel where the same values are set to the IFCm5 to IFCm0 bits as the values to be used to rewrite the IFCn5 to IFCn0 bits (DCHCm.INITm bit = 1).
    - <4> Change the DTFRn register settings. (Be sure to set the DFn bit = 0 and change the settings in the 8-bit manipulation.)
    - <5> To clear a DMA transfer request, clear the DMA transfer request flag (DTFRn.DFn) to 0.
    - <6> Enable the DMA<sub>n</sub> operation (Enn and Emm bits = 1).
2. An interrupt request from an on-chip peripheral I/O input in the standby mode (IDLE or STOP mode) is held pending as a DMA transfer start factor. The held DMA start factor is executed after restoring to the normal operation mode.
  3. If the start factor of DMA transfer is changed using the IFCn5 to IFCn0 bits, be sure to set (0) the DFn bit by instruction immediately after.

After reset: 00H R/W Address: DTFR0 FFFF810H, DTFR1 FFFF812H,  
DTFR2 FFFF814H, DTFR3 FFFF816H

|                       |     |   |       |       |       |       |       |       |
|-----------------------|-----|---|-------|-------|-------|-------|-------|-------|
|                       | <7> | 6 | 5     | 4     | 3     | 2     | 1     | 0     |
| DTFRn<br>(n = 0 to 3) | DFn | 0 | IFCn5 | IFCn4 | IFCn3 | IFCn2 | IFCn1 | IFCn0 |

|                     |                            |
|---------------------|----------------------------|
| DFn <sup>Note</sup> | DMA transfer request flag  |
| 0                   | DMA transfer not requested |
| 1                   | DMA transfer requested     |

**Note** Do not set the DFn bit to “1” by software.

If the interrupt specified as the DMA transfer start factor occurs and it is necessary to clear the DMA transfer request while DMA transfer is disabled (including when it is forcibly terminated by software), stop the operation of the source causing the interrupt, and then write 0 to the DFn bit (for example, disable reception in the case of serial reception). If it is clear that the interrupt will not occur until DMA transfer is resumed next, it is not necessary to stop the operation of the source causing the interrupt.

**Cautions 1.** For the IFCn5 to IFCn0 bits, see Table 19-1 DMA Transfer Start Factors.

**2.** Be sure to set bit 6 to “0”. If it is set to “1”, the operation is not guaranteed.

Table 19-1. DMA Transfer Start Factors (1/2)

| IFCn5 | IFCn4 | IFCn3 | IFCn2 | IFCn1 | IFCn0 | Interrupt Source                                 |
|-------|-------|-------|-------|-------|-------|--|
| 0     | 0     | 0     | 0     | 0     | 0     | DMA request from on-chip peripheral I/O disabled |
| 0     | 0     | 0     | 0     | 0     | 1     | INTLVIL  |
| 0     | 0     | 0     | 0     | 1     | 0     | INTLVIH  |
| 0     | 0     | 0     | 0     | 1     | 1     | INTP11   |
| 0     | 0     | 0     | 1     | 0     | 0     | INTP12   |
| 0     | 0     | 0     | 1     | 0     | 1     | INTP13   |
| 0     | 0     | 0     | 1     | 1     | 0     | INTP15   |
| 0     | 0     | 0     | 1     | 1     | 1     | INTTB0OV_BASE <sup>Note</sup>                    |
| 0     | 0     | 1     | 0     | 0     | 0     | INTTB1OV_BASE <sup>Note</sup>                    |
| 0     | 0     | 1     | 0     | 0     | 1     | INTCMP0L   |
| 0     | 0     | 1     | 0     | 1     | 0     | INTCMP0F   |
| 0     | 0     | 1     | 0     | 1     | 1     | INTCMP1L   |
| 0     | 0     | 1     | 1     | 0     | 0     | INTCMP1F   |
| 0     | 0     | 1     | 1     | 0     | 1     | INTTB0CC0  |
| 0     | 0     | 1     | 1     | 1     | 0     | INTTB0CC1  |
| 0     | 0     | 1     | 1     | 1     | 1     | INTTB0CC2  |
| 0     | 1     | 0     | 0     | 0     | 0     | INTTB0CC3  |
| 0     | 1     | 0     | 0     | 0     | 1     | INTTB1CC0  |
| 0     | 1     | 0     | 0     | 1     | 0     | INTTB1CC1  |
| 0     | 1     | 0     | 0     | 1     | 1     | INTTB1CC2  |
| 0     | 1     | 0     | 1     | 0     | 0     | INTTB1CC3  |
| 0     | 1     | 0     | 1     | 0     | 1     | INTTTEQC00                                       |
| 0     | 1     | 0     | 1     | 1     | 0     | INTTTEQC01                                       |
| 0     | 1     | 0     | 1     | 1     | 1     | INTTTEQC10                                       |
| 0     | 1     | 1     | 0     | 0     | 0     | INTTTEQC11                                       |
| 0     | 1     | 1     | 0     | 0     | 1     | INTTA0CC0  |
| 0     | 1     | 1     | 0     | 1     | 0     | INTTA0CC1  |
| 0     | 1     | 1     | 0     | 1     | 1     | INTTA1CC0  |
| 0     | 1     | 1     | 1     | 0     | 0     | INTTA1CC1  |
| 0     | 1     | 1     | 1     | 0     | 1     | INTTA2CC0  |
| 0     | 1     | 1     | 1     | 1     | 0     | INTTA2CC1  |
| 0     | 1     | 1     | 1     | 1     | 1     | INTTA3CC0  |
| 1     | 0     | 0     | 0     | 0     | 0     | INTTA3CC1  |
| 1     | 0     | 0     | 0     | 0     | 1     | INTTA4CC0  |
| 1     | 0     | 0     | 0     | 1     | 0     | INTTA4CC1  |
| 1     | 0     | 0     | 0     | 1     | 1     | INTDMA0  |
| 1     | 0     | 0     | 1     | 0     | 0     | INTDMA1  |
| 1     | 0     | 0     | 1     | 0     | 1     | INTDMA2  |
| 1     | 0     | 0     | 1     | 1     | 0     | INTDMA3  |

**Remark** n = 0 to 3

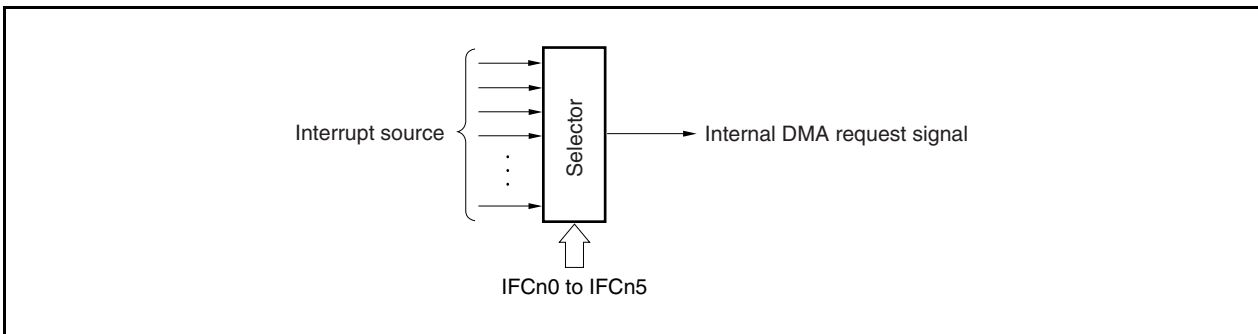
**Note** INTTBaOV\_BASE is an interrupt signal before INTTBaOV is called by the TMQa option in the 6-phase PWM output mode (a = 0, 1). The INTTBaCC0 is also an interrupt signal before culling. For details, see **Figure 10-2 TMQn Option**.

Table 19-1. DMA Transfer Start Factors (2/2)

| IFCn5            | IFCn4 | IFCn3 | IFCn2 | IFCn1 | IFCn0 | Interrupt Source   |
|------------------|-------|-------|-------|-------|-------|--------------------|
| 1                | 0     | 0     | 1     | 1     | 1     | INTUBTIR           |
| 1                | 0     | 1     | 0     | 0     | 0     | INTUBTIT           |
| 1                | 0     | 1     | 0     | 0     | 1     | INTUBTIF           |
| 1                | 0     | 1     | 0     | 1     | 0     | INTUA0R            |
| 1                | 0     | 1     | 0     | 1     | 1     | INTUA0T            |
| 1                | 0     | 1     | 1     | 0     | 0     | INTCB0R            |
| 1                | 0     | 1     | 1     | 0     | 1     | INTCB0T            |
| 1                | 0     | 1     | 1     | 1     | 0     | INTUA1R            |
| 1                | 0     | 1     | 1     | 1     | 1     | INTUA1T            |
| 1                | 1     | 0     | 0     | 0     | 0     | INTCB1R            |
| 1                | 1     | 0     | 0     | 0     | 1     | INTCB1T            |
| 1                | 1     | 0     | 0     | 1     | 0     | INTUA2R            |
| 1                | 1     | 0     | 0     | 1     | 1     | INTUA2T            |
| 1                | 1     | 0     | 1     | 0     | 0     | INTCB2R            |
| 1                | 1     | 0     | 1     | 0     | 1     | INTCB2T            |
| 1                | 1     | 0     | 1     | 1     | 0     | INTIIC             |
| 1                | 1     | 0     | 1     | 1     | 1     | INTAD0             |
| 1                | 1     | 1     | 0     | 0     | 0     | INTAD1             |
| 1                | 1     | 1     | 0     | 0     | 1     | INTAD2             |
| 1                | 1     | 1     | 0     | 1     | 0     | INTTM0EQ0          |
| 1                | 1     | 1     | 0     | 1     | 1     | INTTM1EQ0          |
| 1                | 1     | 1     | 1     | 0     | 0     | INTTM2EQ0          |
| 1                | 1     | 1     | 1     | 0     | 1     | INTTM3EQ0          |
| Other than above |       |       |       |       |       | Setting prohibited |

**Remark** n = 0 to 3

The relationship between the interrupt source and the DMA transfer trigger is as follows (n = 0 to 3).



- Cautions**
1. DMA transfer starts when an interrupt factor set by the IFCn5 to IFCn0 bits occur. To prevent interrupt servicing from being executed, mask the interrupt by setting the interrupt control register. DMA transfer starts even if an interrupt is masked.
  2. If the frequency of the CPU clock falls below the clock of each on-chip peripheral I/O because of the setting of prescaler 2 of the clock generator, the DMA transfer start factor may not be acknowledged.

## 19.4 Transfer Modes

### 19.4.1 Single transfer mode

In single transfer mode, the DMAC releases the bus at each byte/halfword transfer. If there is a subsequent DMA transfer request, transfer is performed again once. This operation continues until a terminal count occurs.

When the DMAC has released the bus, if another higher priority DMA transfer request is issued, the higher priority DMA request always takes precedence. If another DMA transfer request with a lower priority occurs one clock after single transfer has been completed, however, this request does not take precedence even if the previous DMA transfer request signal with a higher priority remains active. DMA transfer with the newly requested lower priority request is executed after the CPU bus has been released.

Figures 19-1 to 19-4 show examples of single transfer.

**Figure 19-1. Single Transfer Example 1**

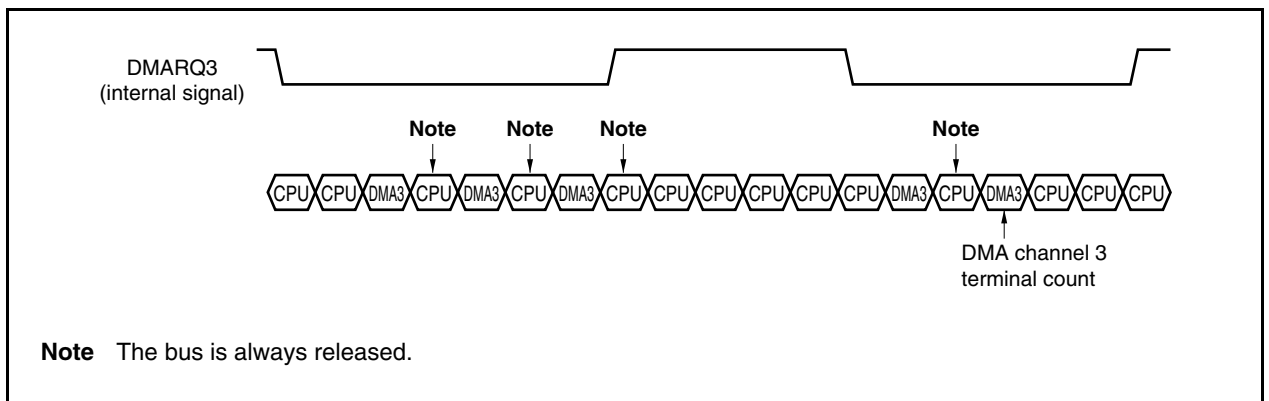


Figure 19-2 shows an example of a single transfer in which a higher priority DMA request is issued. DMA channels 0 to 2 are in the block transfer mode and channel 3 is in the single transfer mode.

**Figure 19-2. Single Transfer Example 2**

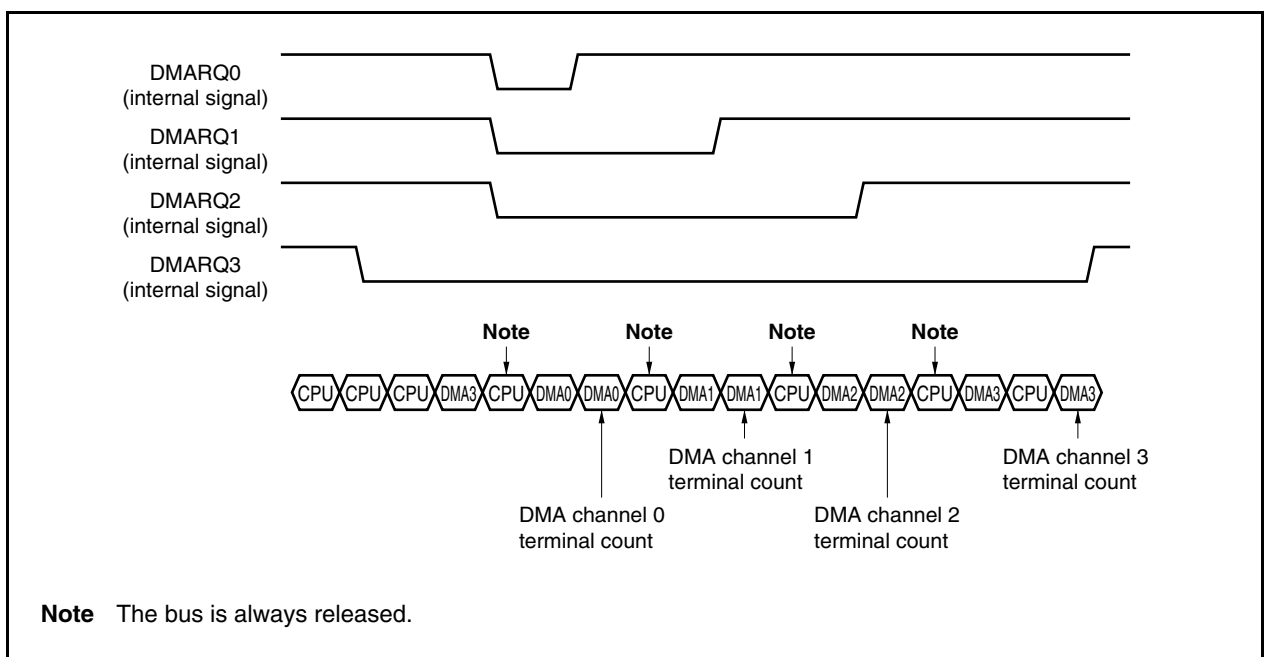


Figure 19-3 is an example of single transfer where a DMA transfer request with a lower priority is issued one clock after single transfer has been completed. DMA channels 0 and 3 are used for single transfer. If two DMA transfer request signals become active at the same time, two DMA transfer operations are alternately executed.

Figure 19-3. Single Transfer Example 3

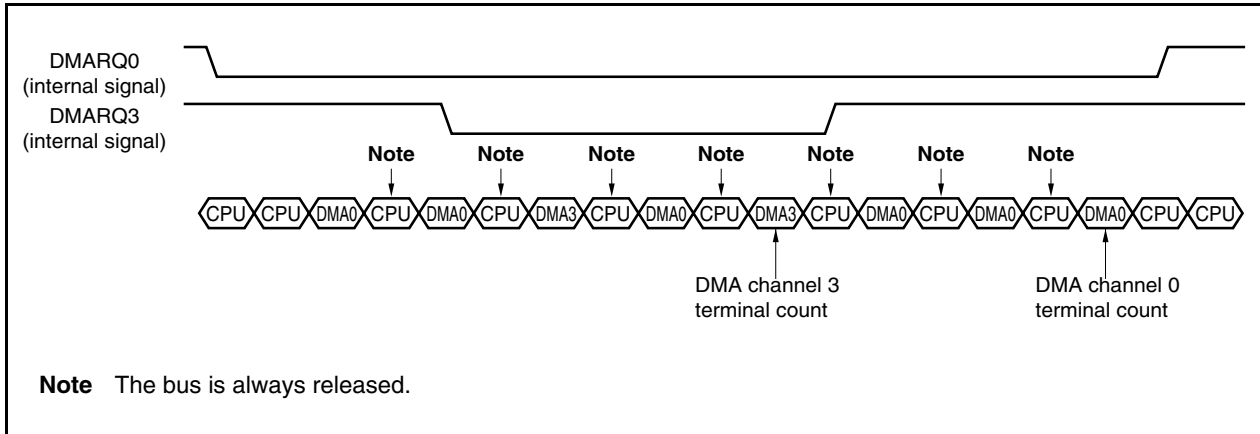
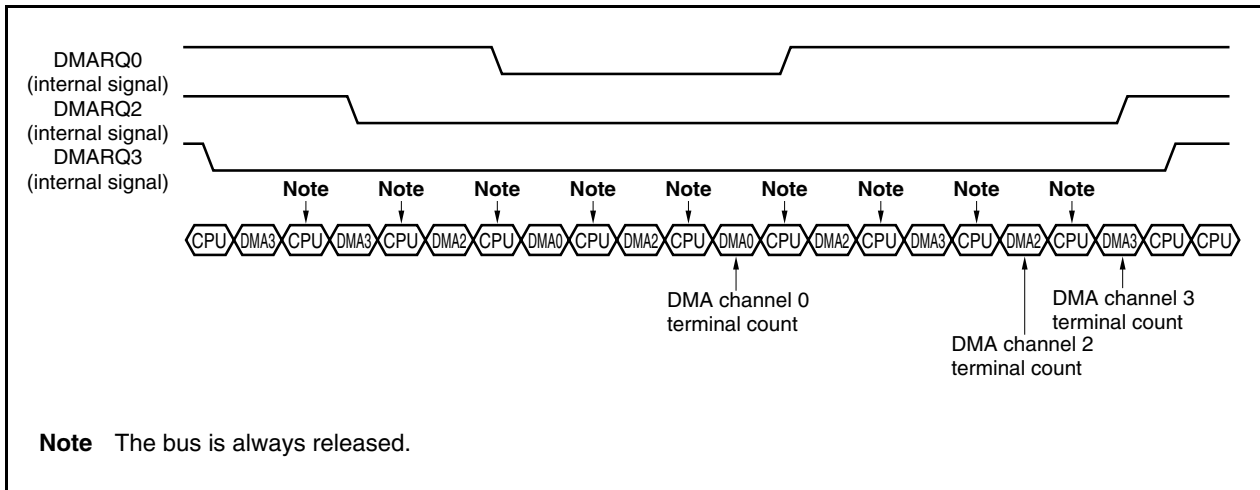


Figure 19-4 is an example of single transfer where two or more DMA transfer requests with a lower priority are issued one clock after single transfer has been completed. DMA channels 0, 2, and 3 are used for single transfer. If three or more DMA transfer request signals become active at the same time, two DMA transfer operations are alternately executed, starting from the one with the highest priority.

Figure 19-4. Single Transfer Example 4





19.4.2 Single-step transfer mode

In single-step transfer mode, the DMAC releases the bus at each byte/halfword transfer. If there is a subsequent DMA transfer request signal, transfer is performed again. This operation continues until a terminal count occurs.

When the DMAC has released the bus, if another higher priority DMA transfer request is issued, the higher priority DMA request always takes precedence.

The following shows an example of a single-step transfer. Figure 19-6 shows an example of single-step transfer made in which a higher priority DMA request is issued. DMA channels 0 and 1 are in the single-step transfer mode.

Figure 19-5. Single-Step Transfer Example 1

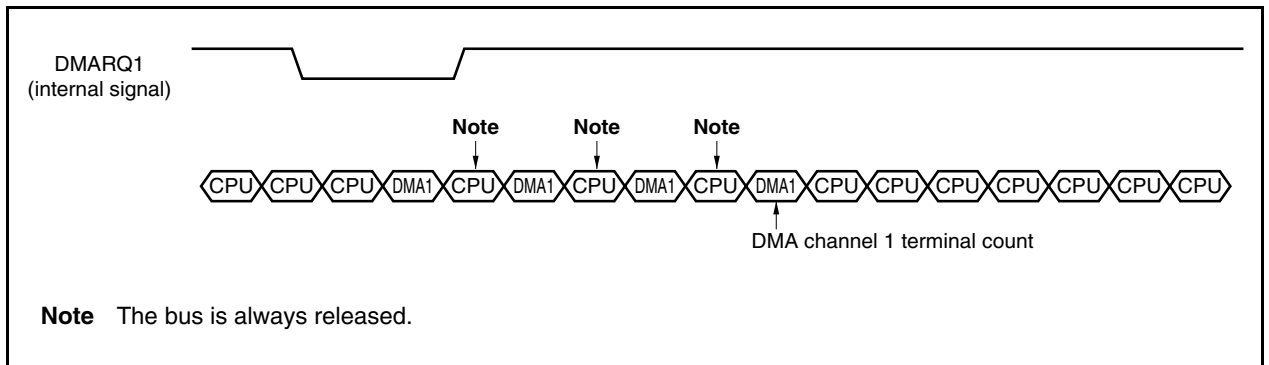
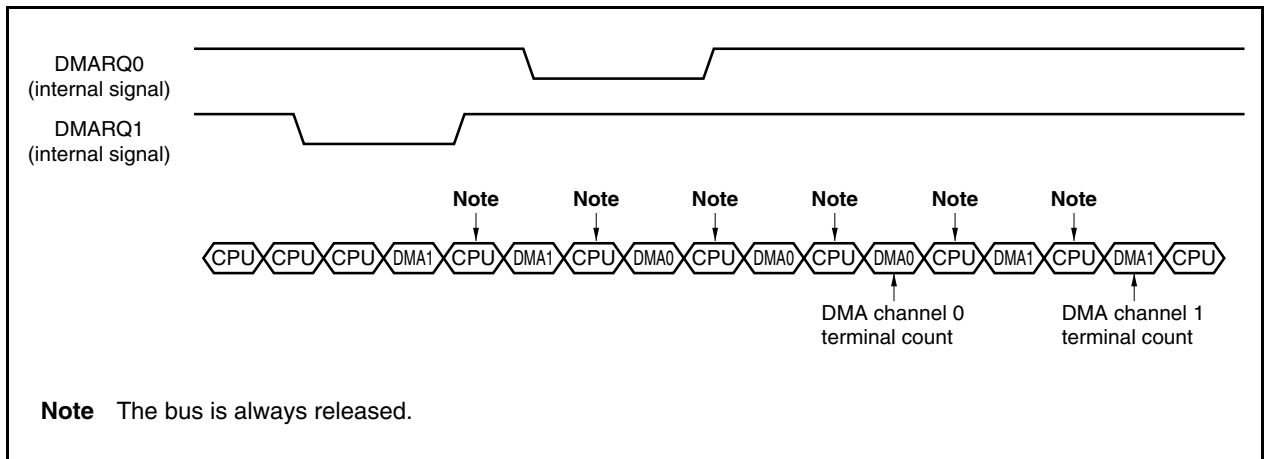


Figure 19-6. Single-Step Transfer Example 2



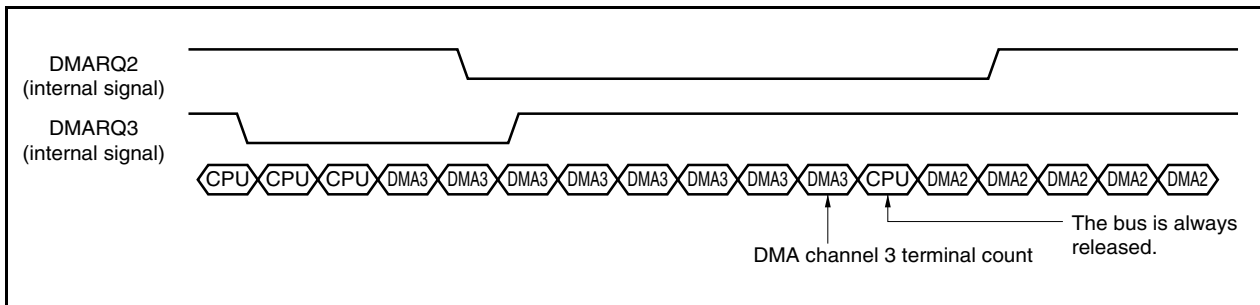
**19.4.3 Block transfer mode**

In the block transfer mode, once transfer starts, the DMAC continues the transfer operation without releasing the bus until a terminal count occurs. No other DMA requests are acknowledged during block transfer.

After the block transfer ends and the DMAC releases the bus, another DMA transfer can be acknowledged.

The following shows an example of block transfer in which a higher priority DMA request is issued. DMA channels 2 and 3 are in the block transfer mode.

**Figure 19-7. Block Transfer Example**



## 19.5 Transfer Types

### 19.5.1 2-cycle transfer

In 2-cycle transfer, data transfer is performed in two cycles, a read cycle (source to DMAC) and a write cycle (DMAC to destination).

In the first cycle, the source address is output and reading is performed from the source to the DMAC. In the second cycle, the destination address is output and writing is performed from the DMAC to the destination.

**Caution** An idle cycle of 1 to 2 clocks is always inserted between a read cycle and a write cycle.

## 19.6 Transfer Target

### 19.6.1 Transfer type and transfer target

Table 19-2 lists the relationship between the transfer type and transfer target. The mark “√” means “transfer possible”, and the mark “x” means “transfer impossible”.

**Table 19-2. Relationship Between Transfer Type and Transfer Target**

|        |  | Destination  |  |              |
|--------|--|--------------|--|--------------|
|        |  | Internal ROM | On-Chip Peripheral I/O <sup>Note</sup> | Internal RAM |
| Source | On-chip peripheral I/O <sup>Note</sup> | x            | √                                      | √            |
|        | Internal RAM                           | x            | √                                      | x            |
|        | Internal ROM                           | x            | x                                      | x            |

**Note** If the transfer target is the on-chip peripheral I/O, only the single transfer mode can be used.

**Cautions 1.** The operation is not guaranteed for combinations of transfer destination and source marked with “x” in Table 19-2.

**2.** Addresses between 3FFF000H and 3FFFFFFH cannot be specified for the source and destination address of DMA transfer.

Be sure to specify an address between FFFF000H and FFFFFFFH.

**Remark** If DMA transfer is executed to transfer data of an on-chip peripheral I/O register (as a transfer source or destination), be sure to specify the same transfer size as the register size. For example, to execute DMA transfer of an 8-bit register, be sure to specify byte (8-bit) transfer.

## 19.7 DMA Channel Priorities

The DMA channel priorities are fixed as follows.

DMA channel 0 > DMA channel 1 > DMA channel 2 > DMA channel 3

In the block transfer mode, the channel used for transfer is never switched.

In the single-step transfer mode, if a higher priority DMA transfer request is issued while the bus is released, the higher priority DMA transfer request is acknowledged.

## 19.8 Next Address Setting Function

The DSAnH, DSAnL, DDAnH, DDAnL, and DBCn registers are two-stage FIFO buffer registers consisting of a master register and a slave register (n = 0 to 3).

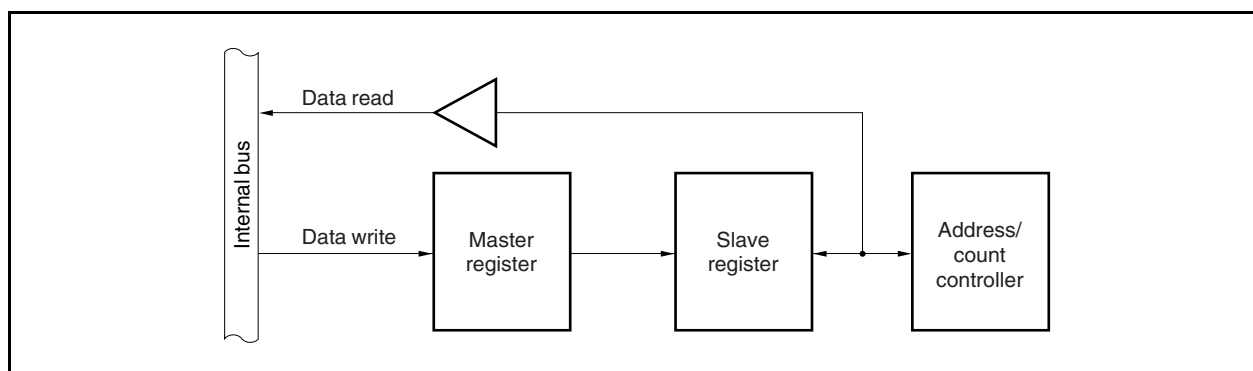
When the terminal count is issued, these registers are automatically rewritten with the value that was set immediately before.

If new DMA transfer setting is made to these registers during DMA transfer, therefore, the values of the registers are automatically updated to the new value after completion of transfer<sup>Note</sup>.

**Note** To make new DMA transfer setting, confirm that DMA transfer has been started. If a new setting is made before the start of DMA transfer, the set value is overwritten to both the master and slave registers.

Figure 19-8 shows the configuration of the buffer register.

**Figure 19-8. Buffer Register Configuration**



The actual DMA transfer is executed in accordance with the contents of the slave register.

The set value to be reflected upon the master register and slave register differs as follows, depending on the timing (period) of setting.

### (1) Period from system reset to the generation of the first DMA transfer request

The set values are reflected on both the master and slave registers.

### (2) During DMA transfer (period from the generation of DMA transfer request to completion of DMA transfer)

The set value is reflected only on the master register and not on the slave register (the slave register holds the set value for the next DMA transfer).

After completion of DMA transfer, however, the contents of the master register are automatically overwritten to the slave register.

If the value of a register is read during this period, the value of the slave register is read.

To check that DMA transfer has been started, confirm that the first transfer has been executed by reading the DBCn register (n = 0 to 3).

### (3) Period from completion of DMA transfer to start of next DMA transfer

The set value is reflected on both the master and slave registers.

**Remark** "Completion of DMA transfer" means either of the following cases.

- End of DMA transfer (terminal count)
- Forced termination of DMA transfer (setting DCHCn.INITn bit to 1).

## 19.9 DMA Transfer Start Factors

There are two types of DMA transfer start factors, as shown below.

- Cautions**
1. Do not use both start factors ((1) and (2)) in combination for the same channel (if both start factors are generated at the same time, only one of them is valid, but the valid start factor cannot be identified).

The operation is not guaranteed if both start factors are used in combination.

2. If DMA transfer is started via software and if the software does not correctly detect whether the expected DMA transfer operation has been ended through manipulation (setting to 1) of the DCHCn.STGn bit, it cannot be guaranteed whether the next (second) manipulation of the STGn bit corresponds to the start of “the next DMA transfer expected by software” (n = 0 to 3).

For example, suppose single transfer is started by manipulating the STGn bit. Even if the STGn bit is manipulated next (the second time) without checking by software whether the single transfer has actually been executed, the next (second) DMA transfer is not always executed. This is because the STGn bit may be manipulated the second time before the first DMA transfer is started or ended because, for example, DMA transfer with a higher priority had already been started when the STGn bit was manipulated for the first time.

It is therefore necessary to manipulate the STGn bit the next time (the second time) after checking whether DMA transfer started by the first manipulation of the STGn bit has been ended.

End of DMA transfer can be checked by checking the contents of the DBCn register.

### (1) Transfer start triggered by software

If the DCHCn.STGn, DCHCn.Enn, and DCHCn.TCn bits are set as follows, DMA transfer is started by software request (n = 0 to 3).

- STGn bit = 1
- Enn bit = 1
- TCn bit = 0

### (2) Transfer start triggered by request from on-chip peripheral I/O

If, when the DCHCn.Enn and DCHCn.TCn bits are set as shown below, an interrupt request is issued from the on-chip peripheral I/O that is set in the DTFRn register, DMA transfer starts (n = 0 to 3).

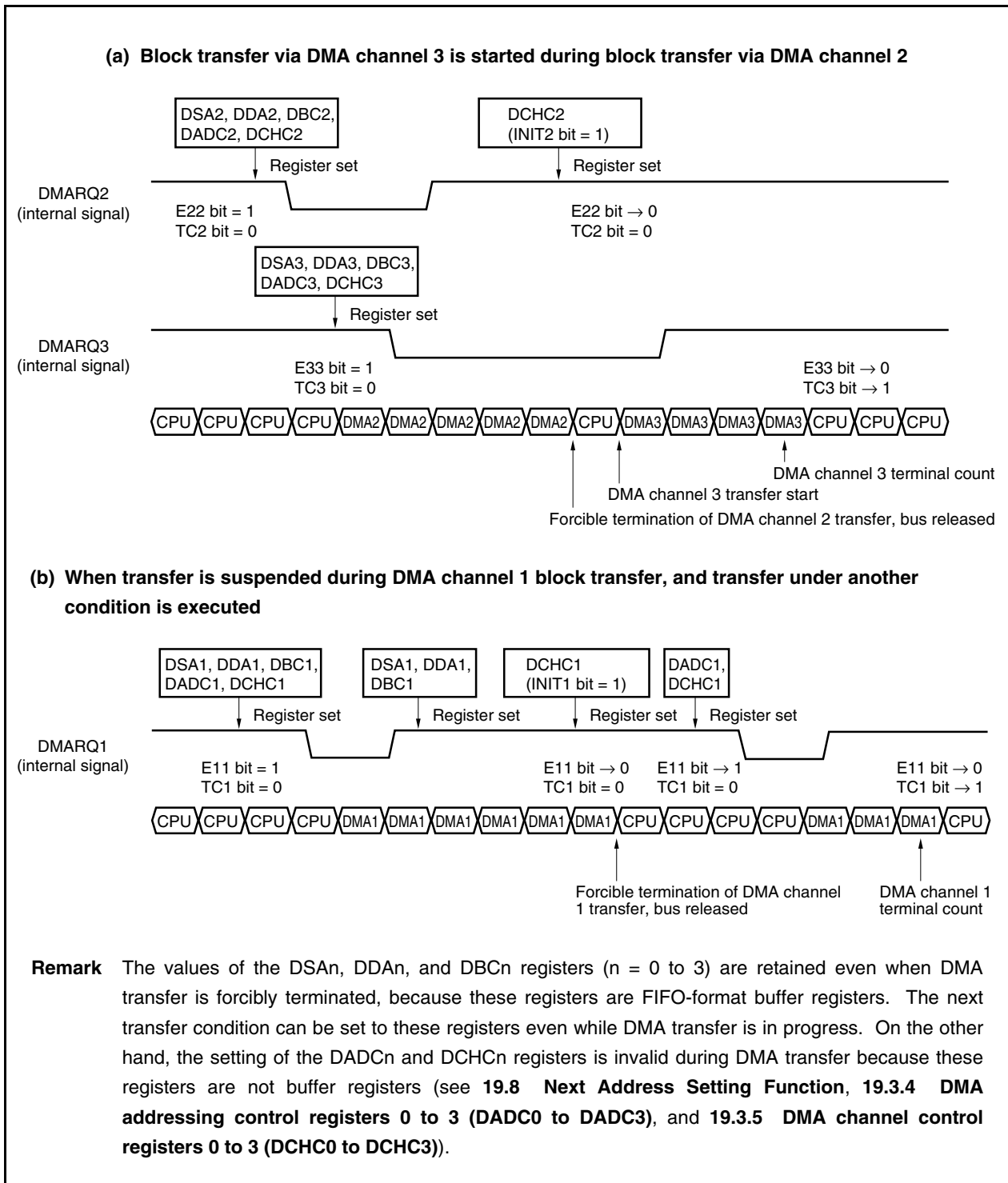
- Enn bit = 1
- TCn bit = 0

19.10 Forcible Termination

DMA transfer can be forcibly terminated by the DCHCn.INITn bit (n = 0 to 3).

An example of forcible termination by the DCHCn.INITn bit is illustrated below (n = 0 to 3).

Figure 19-9. Example of Forcible Termination of DMA Transfer



### 19.11 Times Related to DMA Transfer

The overhead before and after DMA transfer and minimum execution clock for DMA transfer are shown below.

**Table 19-3. Number of Minimum Execution Clocks in DMA Cycle**

| DMA Cycle                        |  | Minimum Number of Execution Clocks                          |
|----------------------------------|--|---|
| <1> Response time to DMA request |  | 4 clocks <sup>Note 1</sup>                                  |
| <2> Memory access                | Internal RAM access                    | 2 clocks <sup>Note 2</sup>                                  |
|                                  | On-chip peripheral I/O register access | 4 clocks + Number of wait cycles specified by VSWC register |

**Notes** 1. If an external interrupt (INTPn) is specified as the DMA transfer start factor, noise elimination time is added (n = 11 to 13, 15, 17, 18).

2. Two clocks for the DMA cycle

The minimum number of execution clocks during the DMA cycle in each mode is as follows.

Single transfer: DMA response time (<1>) + Transfer source memory access (<2>) + 1<sup>Note</sup> + Transfer destination memory access (<2>)

Block transfer: DMA response time (<1>) + Transfer source memory access (<2>) + 1<sup>Note</sup> + Transfer destination memory access (<2>) × Number of transfers

**Note** One clock is always inserted between the read cycle and write cycle of DMA transfer.

### 19.12 Cautions

#### (1) Memory boundary

The transfer operation is not guaranteed if the source or the destination address exceeds the area of DMA targets (internal RAM or on-chip peripheral I/O) during DMA transfer.

#### (2) Transfer of misaligned data

DMA transfer of 16-bit bus width misaligned data is not supported. If the source or the destination address is set to an odd address, the LSB of the address is forcibly handled as "0".

#### (3) Bus arbitration for CPU

Because the DMA controller has a higher priority bus mastership than the CPU, a CPU access that takes place during DMA transfer is held pending until the DMA transfer cycle is completed and the bus is released to the CPU.

However, the CPU can access the internal ROM and RAM to/from which DMA transfer is not being executed.

- The CPU can access the internal ROM when DMA transfer is being executed between the on-chip peripheral I/O and internal RAM.
- The CPU can access the internal ROM and internal RAM when DMA transfer is being executed between on-chip peripheral I/Os.

<R>

**(4) DMA start factors**

Note with caution when setting two or more DMA channels with the same factor.

If two or more DMA channels are started with the same factor, the DMA channel with a lower priority may be acknowledged before the DMA channel with a higher priority.

**(5) Program execution and DMA transfer with internal RAM**

Do not execute DMA transfer to/from the internal RAM and an instruction in the internal RAM simultaneously.

**(6) Timing of setting DCHCn.TCn bit**

The DCHCn.TCn bit is usually set to 1 at the end of DMA transfer. In the case of DMA transfer that is initiated from the internal RAM, however, it is set 4 clocks after end of the last transfer ( $n = 0$  to 3).

**(7) Read values of DSAn and DDAn registers**

If the values of the DSAn and DDAn registers are read during DMA transfer, values in the middle of being updated may be read ( $n = 0$  to 3).

For example, if the DSAnH register and the DSAnL register are read in that order when the value of the DMA transfer source address (DSAn register) is "0000FFFFH" and the counting direction is incremental (when the SADn1 and SADn0 bits of the DADCn register = 00), the value of the DSAnL register differs as follows depending on whether DMA transfer is executed immediately after the DSAnH register has been read.

**(a) If DMA transfer does not occur while the DSAn register is being read**

<1> Reading DSAnH register: DSAnH = 0000H

<2> Reading DSAnL register: DSAnL = FFFFH

**(b) If DMA transfer occurs while the DSAn register is being read**

<1> Reading DSAnH register: DSAnH = 0000H

<2> Occurrence of DMA transfer

<3> Incrementing DSAn register: DSAn = 00010000H

<4> Reading DSAnL register: DSAnL = 0000H

**(8) CLR1, NOT1, and SET1 instructions**

Write the CLR1, NOT1, and SET1 instructions after reading a register and then manipulating the target bit. To set the DCHCn.Enn bit to 1 by using the SET1 instruction, therefore, the TCn bit is cleared to 0 when the DCHCn.TCn bit = 1 ( $n = 0$  to 3).

**19.13 DMA Transfer End**

When DMA transfer ends and the DCHCn.TCn bit is set to 1, a DMA transfer end interrupt (INTDMA $n$ ) is issued to the interrupt controller (INTC) ( $n = 0$  to 3).



## CHAPTER 20 INTERRUPT/EXCEPTION PROCESSING FUNCTION

The V850E/IF3 and V850E/IG3 are provided with a dedicated interrupt controller (INTC) for interrupt servicing and can process a total of 89 to 96 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 V850E/IF3 and V850E/IG3 can process interrupt requests from the on-chip peripheral hardware and external sources. Moreover, exception processing can be started by the TRAP instruction (software exception) or by generation of an exception event (i.e. fetching of an illegal opcode) (exception trap).

### 20.1 Features

#### ○ Interrupts

- Non-maskable interrupts: 1 source (external: none, internal: 1 source)
- Maskable interrupts (the number of maskable interrupt sources differs depending on the product)
  - V850E/IF3: 88 sources (external: 15 sources, internal: 73 sources)
  - V850E/IG3: 95 sources (external: 21 sources, internal: 74 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 traps: 2 sources (illegal opcode exception and debug trap)

Interrupt sources are listed in Table 20-1.

Table 20-1. Interrupt Source List (1/4)

| Type               | Classification | Interrupt/Exception Source |  |  |                   | Default Priority | Exception Code | Handler Address | Restored PC |
|--------------------|----------------|----------------------------|--|--|-------------------|------------------|----------------|-----------------|-------------|
|                    |                | Name                       | Control Register                                 | Generating Source                                | Generating Unit   |                  |                |                 |             |
| Reset              | Interrupt      | RESET                      | –  | RESET pin input                                  | Pin               | –                | 0000H          | 00000000H       | Undefined   |
|                    |                |                            | –  | WDT overflow (WDTRES)                            | WDT               |                  |                |                 |             |
| Non-maskable       | Interrupt      | INTWDT                     | –  | WDT overflow                                     | WDT               | –                | 0010H          | 00000010H       | nextPC      |
| Software exception | Exception      | TRAP0n <sup>Note 1</sup>   | –  | TRAP instruction                                 | –                 | –                | 004nH          | 00000040H       | nextPC      |
|                    | Exception      | TRAP1n <sup>Note 1</sup>   | –  | TRAP instruction                                 | –                 | –                | 005nH          | 00000050H       | nextPC      |
| Exception trap     | Exception      | ILGOP/<br>DBG0             | –  | Illegal instruction code/<br>DBTRAP instruction  | –                 | –                | 0060H          | 00000060H       | nextPC      |
| Maskable           | Interrupt      | INTLVIL                    | LVILIC   | LVI low level voltage detection                  | LVI               | 0                | 0080H          | 00000080H       | nextPC      |
|                    | Interrupt      | INTLVIH                    | LVIHIC   | LVI high level voltage detection                 | LVI               | 1                | 0090H          | 00000090H       | nextPC      |
|                    | Interrupt      | INTP00                     | PIC00  | INTP00 pin valid edge input                      | Pin               | 2                | 00A0H          | 000000A0H       | nextPC      |
|                    | Interrupt      | INTP01                     | PIC01  | INTP01 pin valid edge input                      | Pin               | 3                | 00B0H          | 000000B0H       | nextPC      |
|                    | Interrupt      | INTP02 <sup>Note 2</sup>   | PIC02 <sup>Note 2</sup>                          | INTP02 pin valid edge input                      | Pin               | 4                | 00C0H          | 000000C0H       | nextPC      |
|                    | Interrupt      | INTP03 <sup>Note 2</sup>   | PIC03 <sup>Note 2</sup>                          | INTP03 pin valid edge input                      | Pin               | 5                | 00D0H          | 000000D0H       | nextPC      |
|                    | Interrupt      | INTP04 <sup>Note 2</sup>   | PIC04 <sup>Note 2</sup>                          | INTP04 pin valid edge input                      | Pin               | 6                | 00E0H          | 000000E0H       | nextPC      |
|                    | Interrupt      | INTP05 <sup>Note 2</sup>   | PIC05 <sup>Note 2</sup>                          | INTP05 pin valid edge input                      | Pin               | 7                | 00F0H          | 000000F0H       | nextPC      |
|                    | Interrupt      | INTP06 <sup>Note 2</sup>   | PIC06 <sup>Note 2</sup>                          | INTP06 pin valid edge input                      | Pin               | 8                | 0100H          | 00000100H       | nextPC      |
|                    | Interrupt      | INTP07 <sup>Note 2</sup>   | PIC07 <sup>Note 2</sup>                          | INTP07 pin valid edge input                      | Pin               | 9                | 0110H          | 00000110H       | nextPC      |
|                    | Interrupt      | INTP08                     | PIC08  | INTP08 pin valid edge input                      | Pin               | 10               | 0120H          | 00000120H       | nextPC      |
|                    | Interrupt      | INTP09                     | PIC09  | INTP09 pin valid edge input                      | Pin               | 11               | 0130H          | 00000130H       | nextPC      |
|                    | Interrupt      | INTP10                     | PIC10  | INTP10 pin valid edge input                      | Pin               | 12               | 0140H          | 00000140H       | nextPC      |
|                    | Interrupt      | INTP11                     | PIC11  | INTP11 pin valid edge input                      | Pin               | 13               | 0150H          | 00000150H       | nextPC      |
|                    | Interrupt      | INTP12                     | PIC12  | INTP12 pin valid edge input                      | Pin               | 14               | 0160H          | 00000160H       | nextPC      |
|                    | Interrupt      | INTP13                     | PIC13  | INTP13 pin valid edge input                      | Pin               | 15               | 0170H          | 00000170H       | nextPC      |
|                    | Interrupt      | INTP14                     | PIC14  | INTP14 pin valid edge input                      | Pin               | 16               | 0180H          | 00000180H       | nextPC      |
|                    | Interrupt      | INTP15                     | PIC15  | INTP15 pin valid edge input                      | Pin               | 17               | 0190H          | 00000190H       | nextPC      |
|                    | Interrupt      | INTP16                     | PIC16  | INTP16 pin valid edge input                      | Pin               | 18               | 01A0H          | 000001A0H       | nextPC      |
|                    | Interrupt      | INTP17                     | PIC17  | INTP17 pin valid edge input                      | Pin               | 19               | 01B0H          | 000001B0H       | nextPC      |
|                    | Interrupt      | INTP18                     | PIC18  | INTP18 pin valid edge input                      | Pin               | 20               | 01C0H          | 000001C0H       | nextPC      |
|                    | Interrupt      | INTCMP0L                   | COMPIC0L   | ADC0 overvoltage detection L (comparator output) | ADC0 (comparator) | 21               | 01D0H          | 000001D0H       | nextPC      |
|                    | Interrupt      | INTCMP0F                   | COMPIC0F   | ADC0 overvoltage detection F (comparator output) | ADC0 (comparator) | 22               | 01E0H          | 000001E0H       | nextPC      |
|                    | Interrupt      | INTCMP1L                   | COMPIC1L   | ADC1 overvoltage detection L (comparator output) | ADC1 (comparator) | 23               | 01F0H          | 000001F0H       | nextPC      |
| Interrupt          | INTCMP1F       | COMPIC1F                   | ADC1 overvoltage detection F (comparator output) | ADC1 (comparator)                                | 24                | 0200H            | 00000200H      | nextPC          |             |

- Notes 1. n = 0 to FH  
 2. V850E/IG3 only

Table 20-1. Interrupt Source List (2/4)

| Type     | Classification | Interrupt/Exception Source |                            |  |                 | Default Priority | Exception Code | Handler Address | Restored PC |
|----------|----------------|----------------------------|----------------------------|--|-----------------|------------------|----------------|-----------------|-------------|
|          |                | Name                       | Control Register           | Generating Source  | Generating Unit |                  |                |                 |             |
| Maskable | Interrupt      | INTTB0OV                   | TB0OVIC                    | TAB0 overflow <sup>Note 2</sup>                            | TAB0            | 25               | 0210H          | 00000210H       | nextPC      |
|          | Interrupt      | INTTB0CC0                  | TB0CCIC0                   | TAB0CCR0 capture input/<br>compare match <sup>Note 3</sup> | TAB0            | 26               | 0220H          | 00000220H       | nextPC      |
|          | Interrupt      | INTTB0CC1                  | TB0CCIC1                   | TAB0CCR1 capture input/<br>compare match                   | TAB0            | 27               | 0230H          | 00000230H       | nextPC      |
|          | Interrupt      | INTTB0CC2                  | TB0CCIC2                   | TAB0CCR2 capture input/<br>compare match                   | TAB0            | 28               | 0240H          | 00000240H       | nextPC      |
|          | Interrupt      | INTTB0CC3                  | TB0CCIC3                   | TAB0CCR3 capture input/<br>compare match                   | TAB0            | 29               | 0250H          | 00000250H       | nextPC      |
|          | Interrupt      | INTTB1OV                   | TB1OVIC                    | TAB1 overflow <sup>Note 2</sup>                            | TAB1            | 30               | 0260H          | 00000260H       | nextPC      |
|          | Interrupt      | INTTB1CC0                  | TB1CCIC0                   | TAB1CCR0 capture input/<br>compare match <sup>Note 3</sup> | TAB1            | 31               | 0270H          | 00000270H       | nextPC      |
|          | Interrupt      | INTTB1CC1                  | TB1CCIC1                   | TAB1CCR1 capture input/<br>compare match                   | TAB1            | 32               | 0280H          | 00000280H       | nextPC      |
|          | Interrupt      | INTTB1CC2                  | TB1CCIC2                   | TAB1CCR2 capture input/<br>compare match                   | TAB1            | 33               | 0290H          | 00000290H       | nextPC      |
|          | Interrupt      | INTTB1CC3                  | TB1CCIC3                   | TAB1CCR3 capture input/<br>compare match                   | TAB1            | 34               | 02A0H          | 000002A0H       | nextPC      |
|          | Interrupt      | INTTTIOV0                  | TT0OVIC                    | TMT0 overflow  | TMT0            | 35               | 02B0H          | 000002B0H       | nextPC      |
|          | Interrupt      | INTTTEQC00                 | TT0CCIC0                   | TT0CCR0 capture input <sup>Note 4</sup> /<br>compare match | TMT0            | 36               | 02C0H          | 000002C0H       | nextPC      |
|          | Interrupt      | INTTTEQC01                 | TT0CCIC1                   | TT0CCR1 capture input <sup>Note 4</sup> /<br>compare match | TMT0            | 37               | 02D0H          | 000002D0H       | nextPC      |
|          | Interrupt      | INTTIEC0 <sup>Note 1</sup> | TT0IECIC <sup>Note 1</sup> | Encoder input interrupt 0                                  | TMT0            | 38               | 02E0H          | 000002E0H       | nextPC      |
|          | Interrupt      | INTTTIOV1                  | TT1OVIC                    | TMT1 overflow  | TMT1            | 39               | 02F0H          | 000002F0H       | nextPC      |
|          | Interrupt      | INTTTEQC10                 | TT1CCIC0                   | TT1CCR0 capture input/<br>compare match                    | TMT1            | 40               | 0300H          | 00000300H       | nextPC      |
|          | Interrupt      | INTTTEQC11                 | TT1CCIC1                   | TT1CCR1 capture input/<br>compare match                    | TMT1            | 41               | 0310H          | 00000310H       | nextPC      |
|          | Interrupt      | INTTIEC1                   | TT1IECIC                   | Encoder input interrupt 1                                  | TMT1            | 42               | 0320H          | 00000320H       | nextPC      |
|          | Interrupt      | INTTA0OV                   | TA0OVIC                    | TAA0 overflow  | TAA0            | 43               | 0330H          | 00000330H       | nextPC      |

**Notes 1.** V850E/IG3 only

**2.** When TAB<sub>m</sub> is used in the 6-phase PWM output mode, it functions as INTTB<sub>m</sub>OV (trough interrupt) from the TMQ<sub>m</sub> option (TMQOP<sub>m</sub>) (m = 0, 1).

**3.** When TAB<sub>m</sub> is used in the 6-phase PWM output mode, it functions as INTTB<sub>m</sub>CC0 (peak interrupt) from the TMQ<sub>m</sub> option (TMQOP<sub>m</sub>) (m = 0, 1).

**4.** V850E/IG3 only

In the V850E/IF3, compare match only

Table 20-1. Interrupt Source List (3/4)

| Type      | Classification | Interrupt/Exception Source |                            |  |                 | Default Priority | Exception Code | Handler Address | Restored PC |
|-----------|----------------|----------------------------|----------------------------|--|-----------------|------------------|----------------|-----------------|-------------|
|           |                | Name                       | Control Register           | Generating Source  | Generating Unit |                  |                |                 |             |
| Maskable  | Interrupt      | INTTA0CC0                  | TA0CCIC0                   | TA0CCR0 compare match                                    | TAA0            | 44               | 0340H          | 00000340H       | nextPC      |
|           | Interrupt      | INTTA0CC1                  | TA0CCIC1                   | TA0CCR1 compare match                                    | TAA0            | 45               | 0350H          | 00000350H       | nextPC      |
|           | Interrupt      | INTTA1OV                   | TA1OVIC                    | TAA1 overflow  | TAA1            | 46               | 0360H          | 00000360H       | nextPC      |
|           | Interrupt      | INTTA1CC0                  | TA1CCIC0                   | TA1CCR0 compare match                                    | TAA1            | 47               | 0370H          | 00000370H       | nextPC      |
|           | Interrupt      | INTTA1CC1                  | TA1CCIC1                   | TA1CCR1 compare match                                    | TAA1            | 48               | 0380H          | 00000380H       | nextPC      |
|           | Interrupt      | INTTA2OV                   | TA2OVIC                    | TAA2 overflow  | TAA2            | 49               | 0390H          | 00000390H       | nextPC      |
|           | Interrupt      | INTTA2CC0                  | TA2CCIC0                   | TA2CCR0 capture input/<br>compare match                  | TAA2            | 50               | 03A0H          | 000003A0H       | nextPC      |
|           | Interrupt      | INTTA2CC1                  | TA2CCIC1                   | TA2CCR1 capture input/<br>compare match                  | TAA2            | 51               | 03B0H          | 000003B0H       | nextPC      |
|           | Interrupt      | INTTA3OV                   | TA3OVIC                    | TAA3 overflow  | TAA3            | 52               | 03C0H          | 000003C0H       | nextPC      |
|           | Interrupt      | INTTA3CC0                  | TA3CCIC0                   | TA3CCR0 capture input <sup>Note</sup> /<br>compare match | TAA3            | 53               | 03D0H          | 000003D0H       | nextPC      |
|           | Interrupt      | INTTA3CC1                  | TA3CCIC1                   | TA3CCR1 capture input <sup>Note</sup> /<br>compare match | TAA3            | 54               | 03E0H          | 000003E0H       | nextPC      |
|           | Interrupt      | INTTA4OV                   | TA4OVIC                    | TAA4 overflow  | TAA4            | 55               | 03F0H          | 000003F0H       | nextPC      |
|           | Interrupt      | INTTA4CC0                  | TA4CCIC0                   | TA4CCR0 capture input/<br>compare match                  | TAA4            | 56               | 0400H          | 00000400H       | nextPC      |
|           | Interrupt      | INTTA4CC1                  | TA4CCIC1                   | TA4CCR1 capture input/<br>compare match                  | TAA4            | 57               | 0410H          | 00000410H       | nextPC      |
|           | Interrupt      | INTDMA0                    | DMAIC0                     | DMA channel 0 transfer end                               | DMA0            | 58               | 0420H          | 00000420H       | nextPC      |
|           | Interrupt      | INTDMA1                    | DMAIC1                     | DMA channel 1 transfer end                               | DMA1            | 59               | 0430H          | 00000430H       | nextPC      |
|           | Interrupt      | INTDMA2                    | DMAIC2                     | DMA channel 2 transfer end                               | DMA2            | 60               | 0440H          | 00000440H       | nextPC      |
|           | Interrupt      | INTDMA3                    | DMAIC3                     | DMA channel 3 transfer end                               | DMA3            | 61               | 0450H          | 00000450H       | nextPC      |
|           | Interrupt      | INTUBTIRE                  | UREIC                      | UARTB reception error                                    | UARTB           | 62               | 0460H          | 00000460H       | nextPC      |
|           | Interrupt      | INTUBTIR                   | URIC                       | UARTB reception end                                      | UARTB           | 63               | 0470H          | 00000470H       | nextPC      |
|           | Interrupt      | INTUBTIT                   | UTIC                       | UARTB transmission enable                                | UARTB           | 64               | 0480H          | 00000480H       | nextPC      |
|           | Interrupt      | INTUBTIF                   | UIFIC                      | UARTB FIFO transmission<br>end                           | UARTB           | 65               | 0490H          | 00000490H       | nextPC      |
|           | Interrupt      | INTUBTITO                  | UTOIC                      | UARTB reception timeout                                  | UARTB           | 66               | 04A0H          | 000004A0H       | nextPC      |
|           | Interrupt      | INTUA0RE                   | UA0REIC                    | UARTA0 reception error                                   | UARTA0          | 67               | 04B0H          | 000004B0H       | nextPC      |
|           | Interrupt      | INTUA0R                    | UA0RIC                     | UARTA0 reception end                                     | UARTA0          | 68               | 04C0H          | 000004C0H       | nextPC      |
|           | Interrupt      | INTUA0T                    | UA0TIC                     | UARTA0 transmission enable                               | UARTA0          | 69               | 04D0H          | 000004D0H       | nextPC      |
|           | Interrupt      | INTCB0RE                   | CB0REIC                    | CSIB0 reception error                                    | CSIB0           | 70               | 04E0H          | 000004E0H       | nextPC      |
|           | Interrupt      | INTCB0R                    | CB0RIC                     | CSIB0 reception end                                      | CSIB0           | 71               | 04F0H          | 000004F0H       | nextPC      |
|           | Interrupt      | INTCB0T                    | CB0TIC                     | CSIB0 transmission enable                                | CSIB0           | 72               | 0500H          | 00000500H       | nextPC      |
|           | Interrupt      | INTUA1RE                   | UA1REIC                    | UARTA1 reception error                                   | UARTA1          | 73               | 0510H          | 00000510H       | nextPC      |
| Interrupt | INTUA1R        | UA1RIC                     | UARTA1 reception end       | UARTA1   | 74              | 0520H            | 00000520H      | nextPC          |             |
| Interrupt | INTUA1T        | UA1TIC                     | UARTA1 transmission enable | UARTA1   | 75              | 0530H            | 00000530H      | nextPC          |             |

**Note** V850E/IG3 only

In the V850E/IF3, compare match only

Table 20-1. Interrupt Source List (4/4)

| Type     | Classification | Interrupt/Exception Source |                  |                             |                 | Default Priority | Exception Code | Handler Address | Restored PC |
|----------|----------------|----------------------------|------------------|-----------------------------|-----------------|------------------|----------------|-----------------|-------------|
|          |                | Name                       | Control Register | Generating Source           | Generating Unit |                  |                |                 |             |
| Maskable | Interrupt      | INTCB1RE                   | CB1REIC          | CSIB1 reception error       | CSIB1           | 76               | 0540H          | 00000540H       | nextPC      |
|          | Interrupt      | INTCB1R                    | CB1RIC           | CSIB1 reception end         | CSIB1           | 77               | 0550H          | 00000550H       | nextPC      |
|          | Interrupt      | INTCB1T                    | CB1TIC           | CSIB1 transmission enable   | CSIB1           | 78               | 0560H          | 00000560H       | nextPC      |
|          | Interrupt      | INTUA2RE                   | UA2REIC          | UARTA2 reception error      | UARTA2          | 79               | 0570H          | 00000570H       | nextPC      |
|          | Interrupt      | INTUA2R                    | UA2RIC           | UARTA2 reception end        | UARTA2          | 80               | 0580H          | 00000580H       | nextPC      |
|          | Interrupt      | INTUA2T                    | UA2TIC           | UARTA2 transmission enable  | UARTA2          | 81               | 0590H          | 00000590H       | nextPC      |
|          | Interrupt      | INTCB2RE                   | CB2REIC          | CSIB2 reception error       | CSIB2           | 82               | 05A0H          | 000005A0H       | nextPC      |
|          | Interrupt      | INTCB2R                    | CB2RIC           | CSIB2 reception end         | CSIB2           | 83               | 05B0H          | 000005B0H       | nextPC      |
|          | Interrupt      | INTCB2T                    | CB2TIC           | CSIB2 transmission enable   | CSIB2           | 84               | 05C0H          | 000005C0H       | nextPC      |
|          | Interrupt      | INTIIC                     | IICIC            | IIC serial transfer end     | IIC             | 85               | 05D0H          | 000005D0H       | nextPC      |
|          | Interrupt      | INTAD0                     | AD0IC            | ADC0 conversion end         | ADC0            | 86               | 05E0H          | 000005E0H       | nextPC      |
|          | Interrupt      | INTAD1                     | AD1IC            | ADC1 conversion end         | ADC1            | 87               | 05F0H          | 000005F0H       | nextPC      |
|          | Interrupt      | INTAD2                     | AD2IC            | ADC2 conversion end         | ADC2            | 88               | 0600H          | 00000600H       | nextPC      |
|          | Interrupt      | INTTM0EQ0                  | TM0EQIC0         | TM0CMP0 compare match       | TMM0            | 89               | 0610H          | 00000610H       | nextPC      |
|          | Interrupt      | INTTM1EQ0                  | TM1EQIC0         | TM1CMP0 compare match       | TMM1            | 90               | 0620H          | 00000620H       | nextPC      |
|          | Interrupt      | INTTM2EQ0                  | TM2EQIC0         | TM2CMP0 compare match       | TMM2            | 91               | 0630H          | 00000630H       | nextPC      |
|          | Interrupt      | INTTM3EQ0                  | TM3EQIC0         | TM3CMP0 compare match       | TMM3            | 92               | 0640H          | 00000640H       | nextPC      |
|          | Interrupt      | INTADT0                    | ADT0IC           | ADTRG0 pin valid edge input | Pin             | 93               | 0650H          | 00000650H       | nextPC      |
|          | Interrupt      | INTADT1                    | ADT1IC           | ADTRG1 pin valid edge input | Pin             | 94               | 0660H          | 00000660H       | nextPC      |

**Remarks 1.** Default priority: The priority order when two or more maskable interrupt requests occur at the same time. The highest priority is 0.

Restored PC: The value of the program counter (PC) saved to EIPC, FEPC, or DBPC of CPU 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. In this case, the address of the aborted instruction is the restore PC.)

- 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).

## 20.2 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.

The non-maskable interrupt signals of the V850E/IF3 and V850E/IG3 are the non-maskable interrupt request signals generated by the overflow of the watchdog timer (INTWDT).

INTWDT functions when the WDTM.WDM1 and WDTM.WDM0 bits are set to "01".

**20.2.1 Operation**

If a non-maskable interrupt request signal (INTWDT) 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 the exception code (0010H) to the higher halfword (FECC) of ECR.
- (4) Sets the PSW.NP and PSW.ID bits (1) and clears the PSW.EP bit (0).
- (5) Loads the handler address (00000010H) of the non-maskable interrupt routine to the PC, and transfers control.

The following shows the non-maskable interrupt servicing.

**Figure 20-1. Non-Maskable Interrupt Servicing**

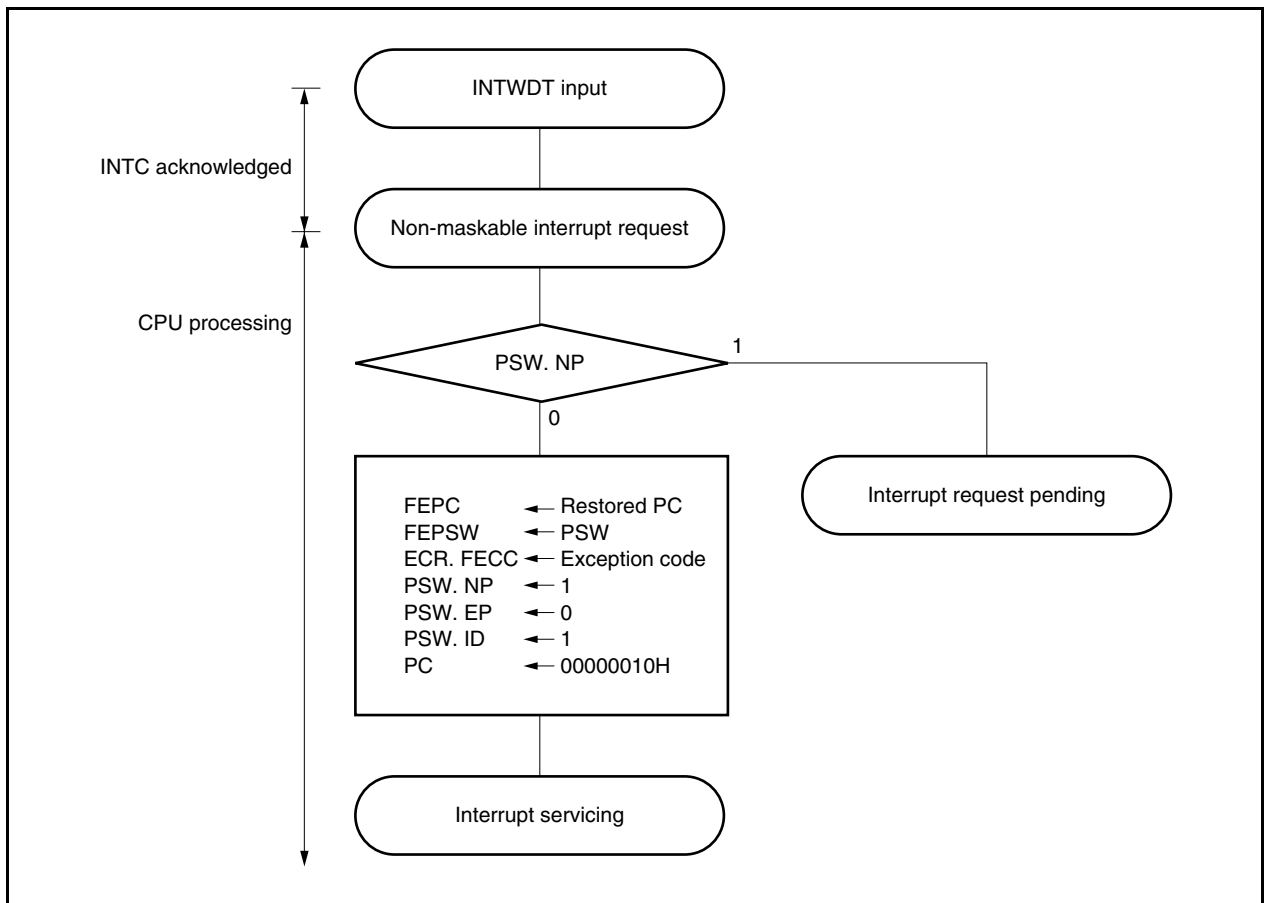
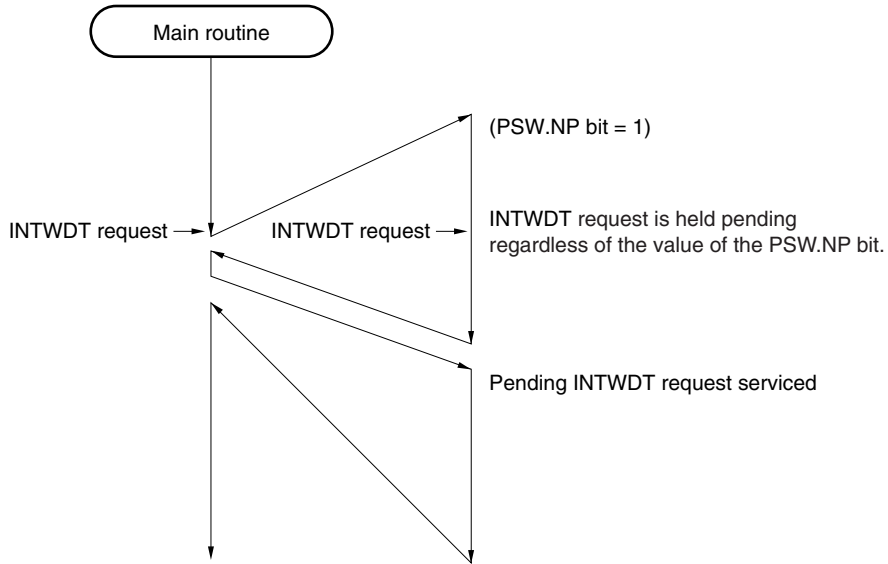
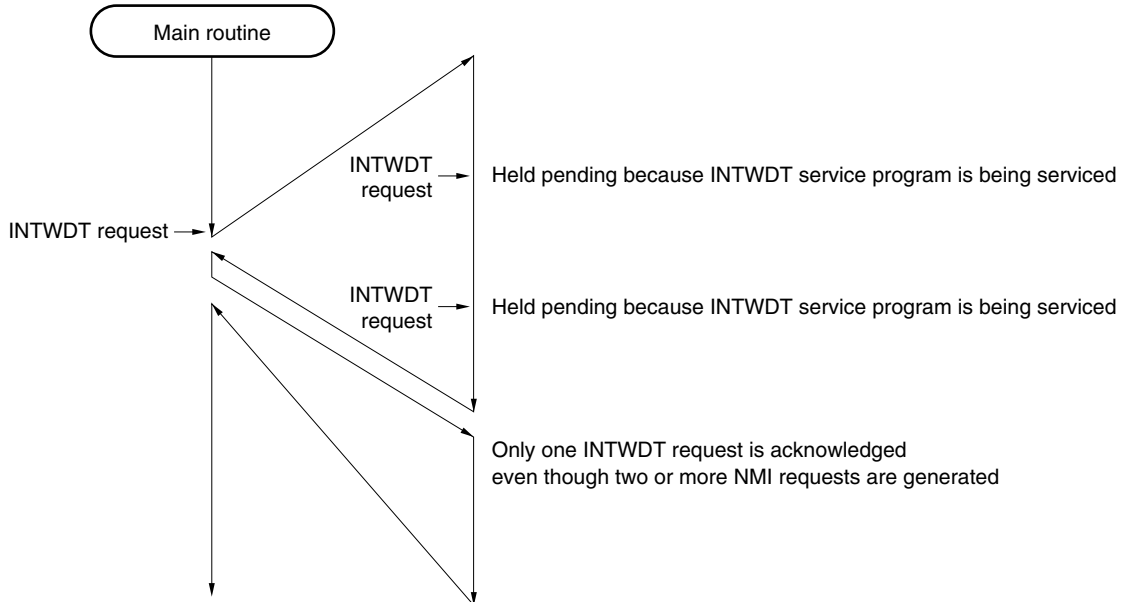


Figure 20-2. Acknowledging Non-Maskable Interrupt Request

(a) If a new INTWDT request is generated while an INTWDT service program is being executed



(b) If a new INTWDT request is generated twice while an INTWDT service program is being executed





**20.2.2 Restore**

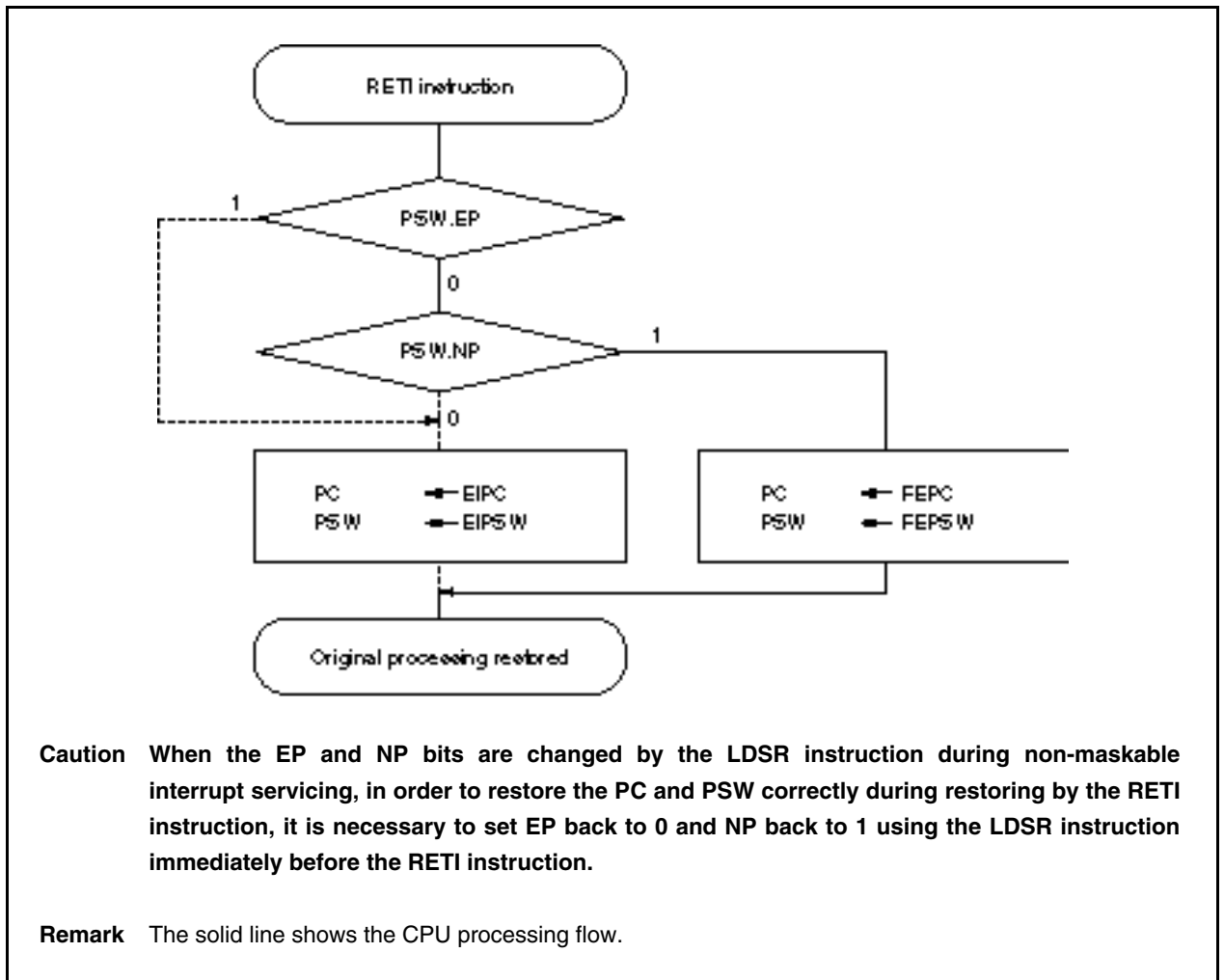
Execution is restored from non-maskable interrupt servicing by the RETI instruction.

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 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.

The following illustrates how the RETI instruction is processed.

**Figure 20-3. RETI Instruction Processing**

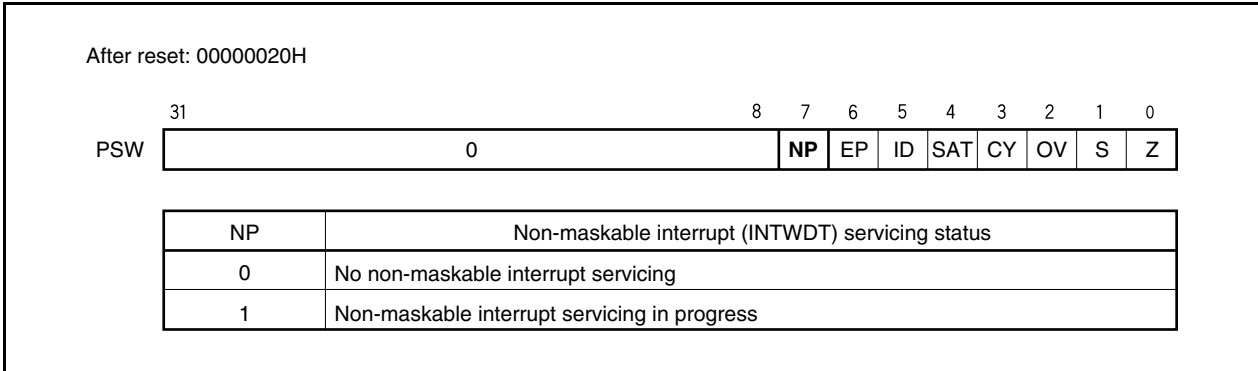


**20.2.3 Non-maskable interrupt status flag (NP)**

The NP flag is a status flag that indicates that non-maskable interrupt (INTWDT) servicing is in progress. The NP flag is allocated to the PSW.

This flag is set when an INTWDT interrupt request signal has been acknowledged, and masks all interrupt requests and exceptions to prohibit multiple interrupts from being acknowledged.

The flag is cleared to 00000020H after reset.



### 20.3 Maskable Interrupts

Maskable interrupt request signals can be masked by interrupt control registers. The V850E/IF3 and V850E/IG3 have 95 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 serviced as multiple interrupts.

To enable multiple interrupt servicing, however, save EIPC and EIPSW to memory or registers before executing the EI instruction, and execute the DI instruction before the RETI instruction to restore the original values of EIPC and EIPSW.

#### 20.3.1 Operation

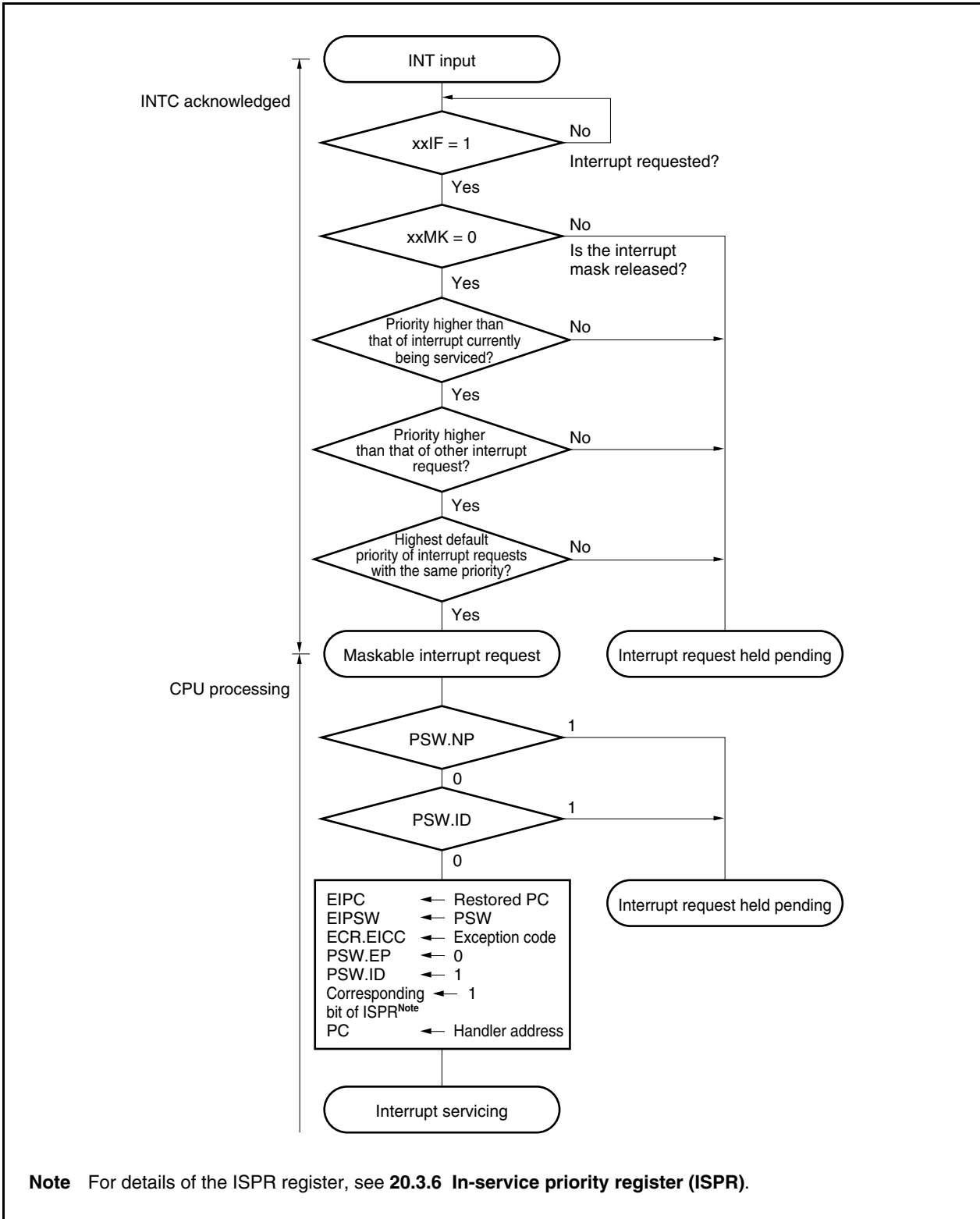
If a maskable interrupt occurs, the CPU performs the following processing, and transfers control to the handler routine.

- <1> Saves the restored PC to EIPC.
- <2> Saves the current PSW to EIPSW.
- <3> Writes an exception code to the lower 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 interrupt controller (INTC) and the maskable interrupt request signal generated while another interrupt is being serviced (while PSW.NP bit = 1 or ID bit = 1) are held pending inside the 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 NP and ID bits are cleared to 0 by using the RETI or LDSR instruction.

How maskable interrupts are serviced is illustrated below.

Figure 20-4. Maskable Interrupt Servicing



**20.3.2 Restore**

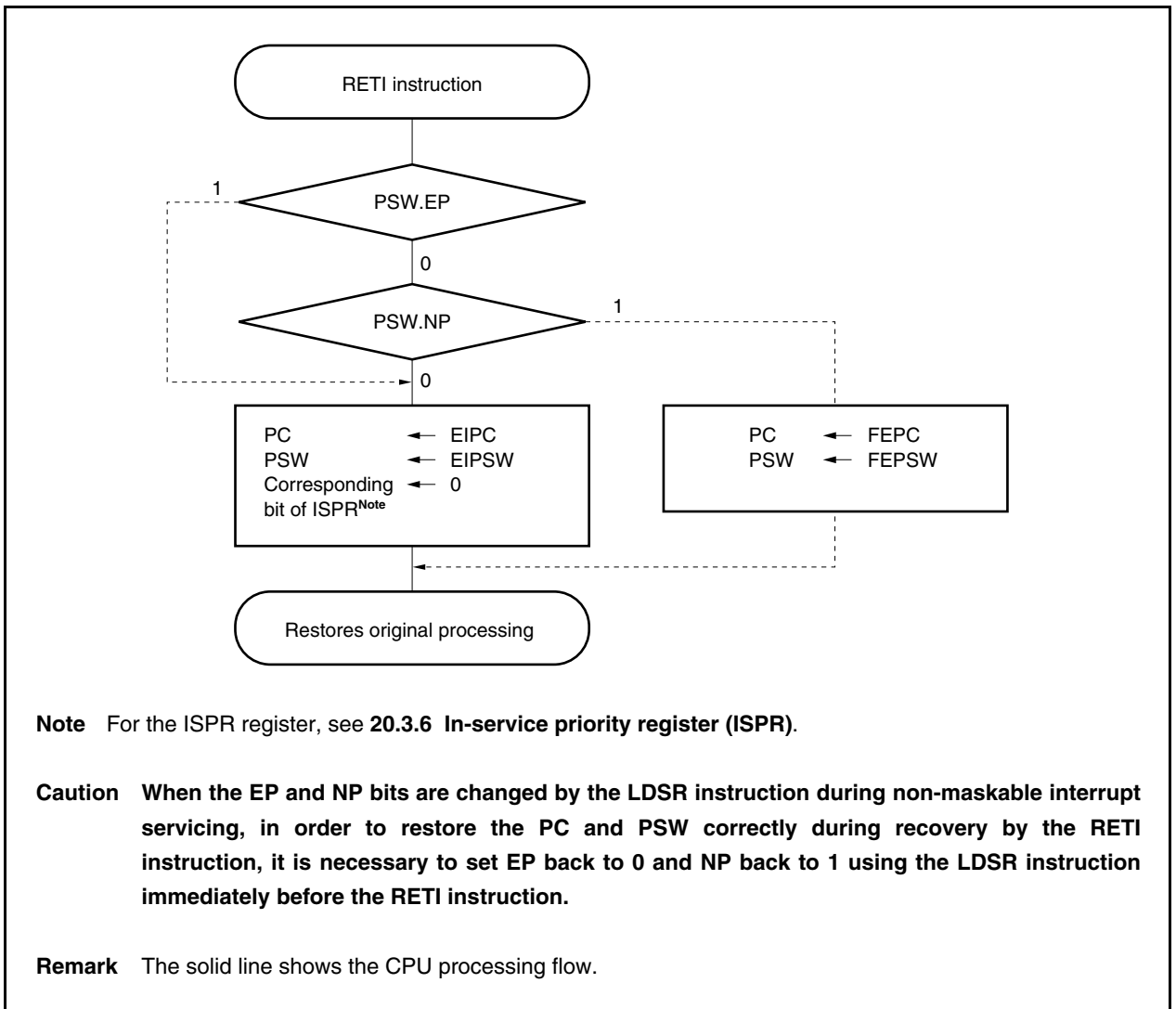
Recovery from maskable interrupt servicing is carried out by the RETI instruction.

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 values of the PC and the 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.

The processing of the RETI instruction is shown below.

**Figure 20-5. RETI Instruction Processing**



### 20.3.3 Priorities of maskable interrupts

The INTC provides 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, interrupts are serviced in order depending on the priority level allocated to each interrupt request signal type (default priority level) beforehand. For more information, see **Table 20-1 Interrupt Source List**. Programmable priority control customizes interrupt request signals into eight levels by the setting of 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 servicing program) to set the interrupt enabled mode.

**Remark** xx: Identification name of each peripheral unit (see **Table 20-2**)  
n: Peripheral unit number (see **Table 20-2**)

**Figure 20-6. Example of Processing in Which Another Interrupt Request Signal Is Issued While an Interrupt Is Being Serviced (1/2)**

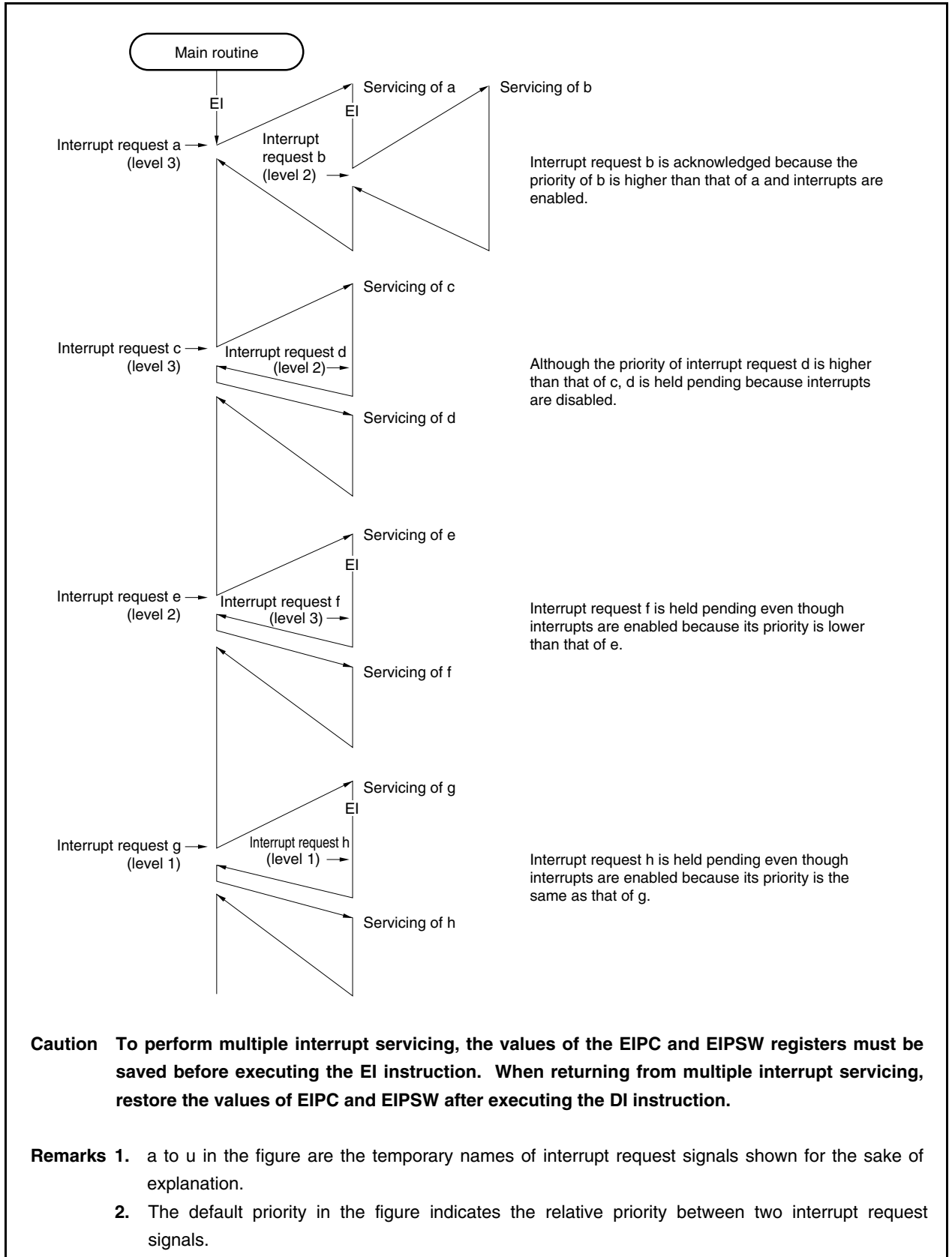


Figure 20-6. Example of Processing in Which Another Interrupt Request Signal Is Issued While an Interrupt Is Being Serviced (2/2)

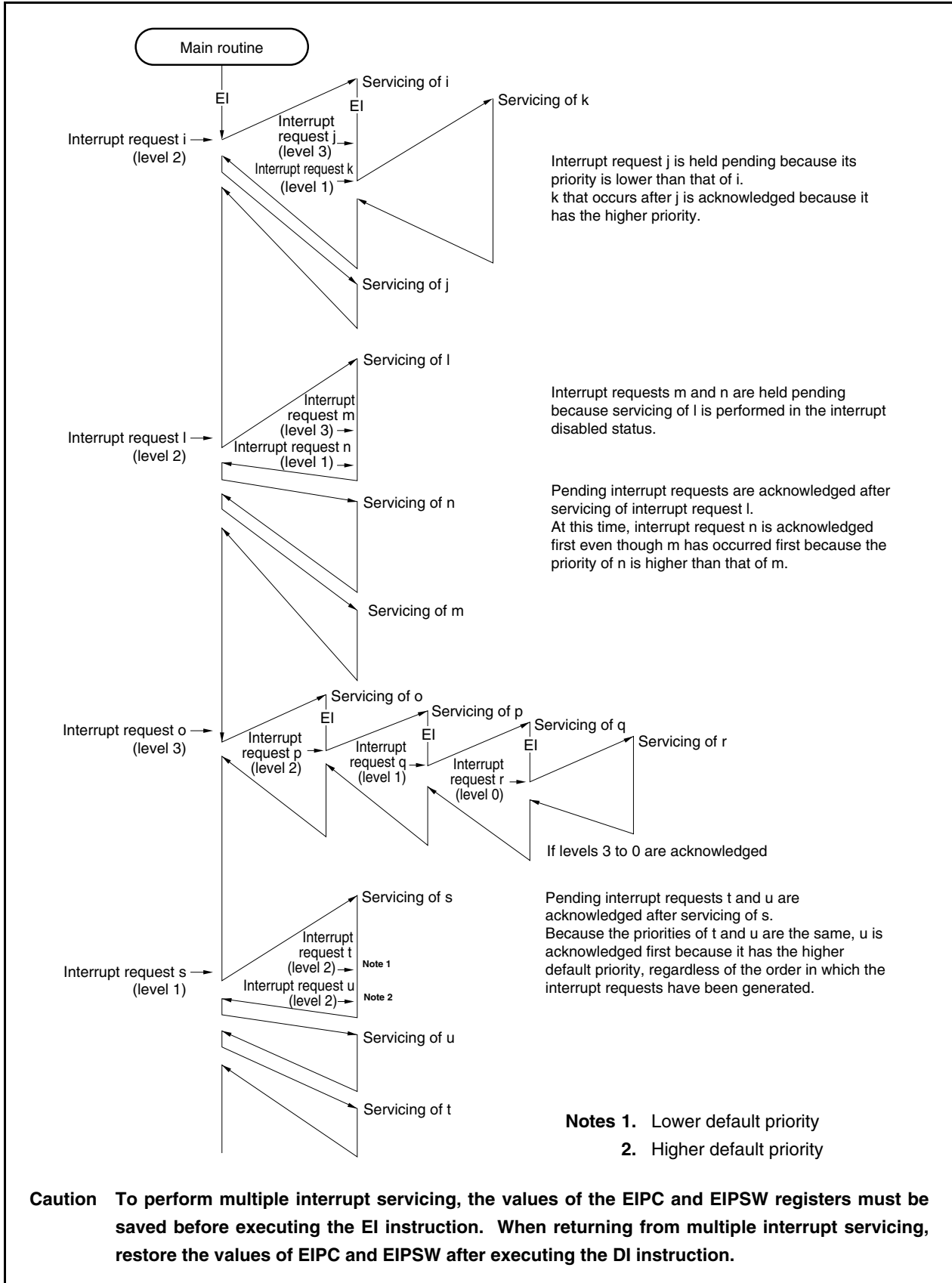
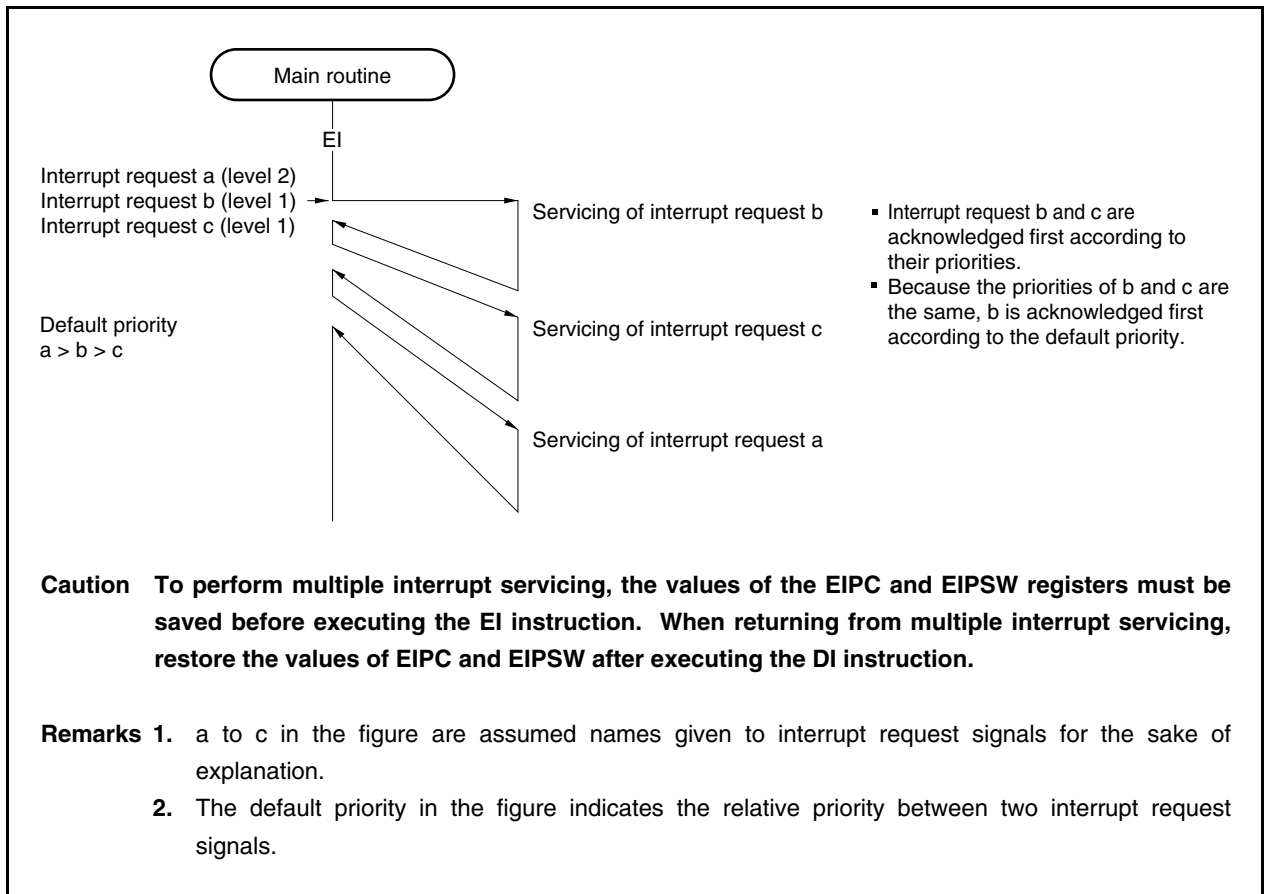




Figure 20-7. Example of Servicing Interrupt Request Signals Generated Simultaneously



#### 20.3.4 Interrupt control registers (xxICn)

An xxICn register is assigned to each interrupt request signal (maskable interrupt) and sets the control conditions for each maskable interrupt request.

These registers can be read or written in 8-bit or 1-bit units.

Reset sets these registers to 47H.

- Cautions**
1. **Disable interrupts (DI) to read the xxICn.xxIFn bit. If the xxIFn bit is read while interrupts are enabled (EI), the correct value may not be read when acknowledging an interrupt and reading the bit conflict.**
  2. **When manipulating the xxICn.xxMKn bit while interrupt requests may occur (including the state in which interrupts are disabled (DI)), be sure to use a bit manipulation instruction or use the IMRm.xxMKn bit (m = 0 to 5).**

After reset: 47H R/W Address: FFFFF110H to FFFFF1CCH

|       |       |       |   |   |   |        |        |        |
|-------|-------|-------|---|---|---|--------|--------|--------|
|       | <7>   | <6>   | 5 | 4 | 3 | 2      | 1      | 0      |
| xxICn | xxIFn | xxMKn | 0 | 0 | 0 | xxPRn2 | xxPRn1 | xxPRn0 |

|       |  |
|-------|--|
| xxIFn | Interrupt request flag <sup>Note</sup> |
| 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 20-2**)  
n: Peripheral unit number (see **Table 20-2**)

The addresses and bits of the interrupt control registers are as follows.

Table 20-2. Addresses and Bits of Interrupt Control Registers (1/3)

| Address   | Register                 | Bit      |          |   |   |   |           |           |           |
|-----------|--------------------------|----------|----------|---|---|---|-----------|-----------|-----------|
|           |                          | <7>      | <6>      | 5 | 4 | 3 | 2         | 1         | 0         |
| FFFFF110H | LVILIC                   | LVILIF   | LVILMK   | 0 | 0 | 0 | LVILPR2   | LVILPR1   | LVILPR0   |
| FFFFF112H | LVIHIC                   | LVIHIF   | LVIHMK   | 0 | 0 | 0 | LVIHPR2   | LVIHPR1   | LVIHPR0   |
| FFFFF114H | PIC00                    | PIF00    | PMK00    | 0 | 0 | 0 | PPR002    | PPR001    | PPR000    |
| FFFFF116H | PIC01                    | PIF01    | PMK01    | 0 | 0 | 0 | PPR012    | PPR011    | PPR010    |
| FFFFF118H | PIC02 <sup>Note</sup>    | PIF02    | PMK02    | 0 | 0 | 0 | PPR022    | PPR021    | PPR020    |
| FFFFF11AH | PIC03 <sup>Note</sup>    | PIF03    | PMK03    | 0 | 0 | 0 | PPR032    | PPR031    | PPR030    |
| FFFFF11CH | PIC04 <sup>Note</sup>    | PIF04    | PMK04    | 0 | 0 | 0 | PPR042    | PPR041    | PPR040    |
| FFFFF11EH | PIC05 <sup>Note</sup>    | PIF05    | PMK05    | 0 | 0 | 0 | PPR052    | PPR051    | PPR050    |
| FFFFF120H | PIC06 <sup>Note</sup>    | PIF06    | PMK06    | 0 | 0 | 0 | PPR062    | PPR061    | PPR060    |
| FFFFF122H | PIC07 <sup>Note</sup>    | PIF07    | PMK07    | 0 | 0 | 0 | PPR072    | PPR071    | PPR070    |
| FFFFF124H | PIC08                    | PIF08    | PMK08    | 0 | 0 | 0 | PPR082    | PPR081    | PPR080    |
| FFFFF126H | PIC09                    | PIF09    | PMK09    | 0 | 0 | 0 | PPR092    | PPR091    | PPR090    |
| FFFFF128H | PIC10                    | PIF10    | PMK10    | 0 | 0 | 0 | PPR102    | PPR101    | PPR100    |
| FFFFF12AH | PIC11                    | PIF11    | PMK11    | 0 | 0 | 0 | PPR112    | PPR111    | PPR110    |
| FFFFF12CH | PIC12                    | PIF12    | PMK12    | 0 | 0 | 0 | PPR122    | PPR121    | PPR120    |
| FFFFF12EH | PIC13                    | PIF13    | PMK13    | 0 | 0 | 0 | PPR132    | PPR131    | PPR130    |
| FFFFF130H | PIC14                    | PIF14    | PMK14    | 0 | 0 | 0 | PPR142    | PPR141    | PPR140    |
| FFFFF132H | PIC15                    | PIF15    | PMK15    | 0 | 0 | 0 | PPR152    | PPR151    | PPR150    |
| FFFFF134H | PIC16                    | PIF16    | PMK16    | 0 | 0 | 0 | PPR162    | PPR161    | PPR160    |
| FFFFF136H | PIC17                    | PIF17    | PMK17    | 0 | 0 | 0 | PPR172    | PPR171    | PPR170    |
| FFFFF138H | PIC18                    | PIF18    | PMK18    | 0 | 0 | 0 | PPR182    | PPR181    | PPR180    |
| FFFFF13AH | COMPIC0L                 | CMPIF0L  | CMPMK0L  | 0 | 0 | 0 | CMPPR0L2  | CMPPR0L1  | CMPPR0L0  |
| FFFFF13CH | COMPIC0F                 | CMPIF0F  | CMPMK0F  | 0 | 0 | 0 | CMPPR0F2  | CMPPR0F1  | CMPPR0F0  |
| FFFFF13EH | COMPIC1L                 | CMPIF1L  | CMPMK1L  | 0 | 0 | 0 | CMPPR1L2  | CMPPR1L1  | CMPPR1L0  |
| FFFFF140H | COMPIC1F                 | CMPIF1F  | CMPMK1F  | 0 | 0 | 0 | CMPPR1F2  | CMPPR1F1  | CMPPR1F0  |
| FFFFF142H | TB0OVIC                  | TB0OVIF  | TB0OVMK  | 0 | 0 | 0 | TB0OVPR2  | TB0OVPR1  | TB0OVPR0  |
| FFFFF144H | TB0CCIC0                 | TB0CCIF0 | TB0CCMK0 | 0 | 0 | 0 | TB0CCPR2  | TB0CCPR1  | TB0CCPR0  |
| FFFFF146H | TB0CCIC1                 | TB0CCIF1 | TB0CCMK1 | 0 | 0 | 0 | TB0CCPR12 | TB0CCPR11 | TB0CCPR10 |
| FFFFF148H | TB0CCIC2                 | TB0CCIF2 | TB0CCMK2 | 0 | 0 | 0 | TB0CCPR22 | TB0CCPR21 | TB0CCPR20 |
| FFFFF14AH | TB0CCIC3                 | TB0CCIF3 | TB0CCMK3 | 0 | 0 | 0 | TB0CCPR32 | TB0CCPR31 | TB0CCPR30 |
| FFFFF14CH | TB1OVIC                  | TB1OVIF  | TB1OVMK  | 0 | 0 | 0 | TB1OVPR2  | TB1OVPR1  | TB1OVPR0  |
| FFFFF14EH | TB1CCIC0                 | TB1CCIF0 | TB1CCMK0 | 0 | 0 | 0 | TB1CCPR2  | TB1CCPR1  | TB1CCPR0  |
| FFFFF150H | TB1CCIC1                 | TB1CCIF1 | TB1CCMK1 | 0 | 0 | 0 | TB1CCPR12 | TB1CCPR11 | TB1CCPR10 |
| FFFFF152H | TB1CCIC2                 | TB1CCIF2 | TB1CCMK2 | 0 | 0 | 0 | TB1CCPR22 | TB1CCPR21 | TB1CCPR20 |
| FFFFF154H | TB1CCIC3                 | TB1CCIF3 | TB1CCMK3 | 0 | 0 | 0 | TB1CCPR32 | TB1CCPR31 | TB1CCPR30 |
| FFFFF156H | TT0OVIC                  | TT0OVIF  | TT0OVMK  | 0 | 0 | 0 | TT0OVPR2  | TT0OVPR1  | TT0OVPR0  |
| FFFFF158H | TT0CCIC0                 | TT0CCIF0 | TT0CCMK0 | 0 | 0 | 0 | TT0CCPR2  | TT0CCPR1  | TT0CCPR0  |
| FFFFF15AH | TT0CCIC1                 | TT0CCIF1 | TT0CCMK1 | 0 | 0 | 0 | TT0CCPR12 | TT0CCPR11 | TT0CCPR10 |
| FFFFF15CH | TT0IECIC <sup>Note</sup> | TT0IECIF | TT0IECMK | 0 | 0 | 0 | TT0IECPR2 | TT0IECPR1 | TT0IECPR0 |
| FFFFF15EH | TT1OVIC                  | TT1OVIF  | TT1OVMK  | 0 | 0 | 0 | TT1OVPR2  | TT1OVPR1  | TT1OVPR0  |
| FFFFF160H | TT1CCIC0                 | TT1CCIF0 | TT1CCMK0 | 0 | 0 | 0 | TT1CCPR2  | TT1CCPR1  | TT1CCPR0  |
| FFFFF162H | TT1CCIC1                 | TT1CCIF1 | TT1CCMK1 | 0 | 0 | 0 | TT1CCPR12 | TT1CCPR11 | TT1CCPR10 |

Note V850E/IG3 only

Table 20-2. Addresses and Bits of Interrupt Control Registers (2/3)

| Address   | Register | Bit      |          |   |   |   |           |           |           |
|-----------|----------|----------|----------|---|---|---|-----------|-----------|-----------|
|           |          | <7>      | <6>      | 5 | 4 | 3 | 2         | 1         | 0         |
| FFFFF164H | TT1IECIC | TT1IECIF | TT1IECMK | 0 | 0 | 0 | TT1IECPR2 | TT1IECPR1 | TT1IECPR0 |
| FFFFF166H | TA0OVIC  | TA0OVIF  | TA0OVMK  | 0 | 0 | 0 | TA0OVPR2  | TA0OVPR1  | TA0OVPR0  |
| FFFFF168H | TA0CCIC0 | TA0CCIF0 | TA0CCMK0 | 0 | 0 | 0 | TA0CCPR2  | TA0CCPR1  | TA0CCPR0  |
| FFFFF16AH | TA0CCIC1 | TA0CCIF1 | TA0CCMK1 | 0 | 0 | 0 | TA0CCPR12 | TA0CCPR11 | TA0CCPR10 |
| FFFFF16CH | TA1OVIC  | TA1OVIF  | TA1OVMK  | 0 | 0 | 0 | TA1OVPR2  | TA1OVPR1  | TA1OVPR0  |
| FFFFF16EH | TA1CCIC0 | TA1CCIF0 | TA1CCMK0 | 0 | 0 | 0 | TA1CCPR2  | TA1CCPR1  | TA1CCPR0  |
| FFFFF170H | TA1CCIC1 | TA1CCIF1 | TA1CCMK1 | 0 | 0 | 0 | TA1CCPR12 | TA1CCPR11 | TA1CCPR10 |
| FFFFF172H | TA2OVIC  | TA2OVIF  | TA2OVMK  | 0 | 0 | 0 | TA2OVPR2  | TA2OVPR1  | TA2OVPR0  |
| FFFFF174H | TA2CCIC0 | TA2CCIF0 | TA2CCMK0 | 0 | 0 | 0 | TA2CCPR2  | TA2CCPR1  | TA2CCPR0  |
| FFFFF176H | TA2CCIC1 | TA2CCIF1 | TA2CCMK1 | 0 | 0 | 0 | TA2CCPR12 | TA2CCPR11 | TA2CCPR10 |
| FFFFF178H | TA3OVIC  | TA3OVIF  | TA3OVMK  | 0 | 0 | 0 | TA3OVPR2  | TA3OVPR1  | TA3OVPR0  |
| FFFFF17AH | TA3CCIC0 | TA3CCIF0 | TA3CCMK0 | 0 | 0 | 0 | TA3CCPR2  | TA3CCPR1  | TA3CCPR0  |
| FFFFF17CH | TA3CCIC1 | TA3CCIF1 | TA3CCMK1 | 0 | 0 | 0 | TA3CCPR12 | TA3CCPR11 | TA3CCPR10 |
| FFFFF17EH | TA4OVIC  | TA4OVIF  | TA4OVMK  | 0 | 0 | 0 | TA4OVPR2  | TA4OVPR1  | TA4OVPR0  |
| FFFFF180H | TA4CCIC0 | TA4CCIF0 | TA4CCMK0 | 0 | 0 | 0 | TA4CCPR2  | TA4CCPR1  | TA4CCPR0  |
| FFFFF182H | TA4CCIC1 | TA4CCIF1 | TA4CCMK1 | 0 | 0 | 0 | TA4CCPR12 | TA4CCPR11 | TA4CCPR10 |
| FFFFF184H | DMAIC0   | DMAIF0   | DMAMK0   | 0 | 0 | 0 | DMAPR2    | DMAPR1    | DMAPR0    |
| FFFFF186H | DMAIC1   | DMAIF1   | DMAMK1   | 0 | 0 | 0 | DMAPR12   | DMAPR11   | DMAPR10   |
| FFFFF188H | DMAIC2   | DMAIF2   | DMAMK2   | 0 | 0 | 0 | DMAPR22   | DMAPR21   | DMAPR20   |
| FFFFF18AH | DMAIC3   | DMAIF3   | DMAMK3   | 0 | 0 | 0 | DMAPR32   | DMAPR31   | DMAPR30   |
| FFFFF18CH | UREIC    | UREIF    | UREMK    | 0 | 0 | 0 | UREPR2    | UREPR1    | UREPR0    |
| FFFFF18EH | URIC     | URIF     | URMK     | 0 | 0 | 0 | URPR2     | URPR1     | URPR0     |
| FFFFF190H | UTIC     | UTIF     | UTMK     | 0 | 0 | 0 | UTPR2     | UTPR1     | UTPR0     |
| FFFFF192H | UIFIC    | UIFIF    | UIFMK    | 0 | 0 | 0 | UIFPR2    | UIFPR1    | UIFPR0    |
| FFFFF194H | UTOIC    | UTOIF    | UTOMK    | 0 | 0 | 0 | UTOPR2    | UTOPR1    | UTOPR0    |
| FFFFF196H | UA0REIC  | UA0REIF  | UA0REMK  | 0 | 0 | 0 | UA0REPR2  | UA0REPR1  | UA0REPR0  |
| FFFFF198H | UA0RIC   | UA0RIF   | UA0RMK   | 0 | 0 | 0 | UA0RPR2   | UA0RPR1   | UA0RPR0   |
| FFFFF19AH | UA0TIC   | UA0TIF   | UA0TMK   | 0 | 0 | 0 | UA0TPR2   | UA0TPR1   | UA0TPR0   |
| FFFFF19CH | CB0REIC  | CB0REIF  | CB0REMK  | 0 | 0 | 0 | CB0REPR2  | CB0REPR1  | CB0REPR0  |
| FFFFF19EH | CB0RIC   | CB0RIF   | CB0RMK   | 0 | 0 | 0 | CB0RPR2   | CB0RPR1   | CB0RPR0   |
| FFFFF1A0H | CB0TIC   | CB0TIF   | CB0TMK   | 0 | 0 | 0 | CB0TPR2   | CB0TPR1   | CB0TPR0   |
| FFFFF1A2H | UA1REIC  | UA1REIF  | UA1REMK  | 0 | 0 | 0 | UA1REPR2  | UA1REPR1  | UA1REPR0  |
| FFFFF1A4H | UA1RIC   | UA1RIF   | UA1RMK   | 0 | 0 | 0 | UA1RPR2   | UA1RPR1   | UA1RPR0   |
| FFFFF1A6H | UA1TIC   | UA1TIF   | UA1TMK   | 0 | 0 | 0 | UA1TPR2   | UA1TPR1   | UA1TPR0   |
| FFFFF1A8H | CB1REIC  | CB1REIF  | CB1REMK  | 0 | 0 | 0 | CB1REPR2  | CB1REPR1  | CB1REPR0  |
| FFFFF1AAH | CB1RIC   | CB1RIF   | CB1RMK   | 0 | 0 | 0 | CB1RPR2   | CB1RPR1   | CB1RPR0   |
| FFFFF1ACH | CB1TIC   | CB1TIF   | CB1TMK   | 0 | 0 | 0 | CB1TPR2   | CB1TPR1   | CB1TPR0   |
| FFFFF1AEH | UA2REIC  | UA2REIF  | UA2REMK  | 0 | 0 | 0 | UA2REPR2  | UA2REPR1  | UA2REPR0  |
| FFFFF1B0H | UA2RIC   | UA2RIF   | UA2RMK   | 0 | 0 | 0 | UA2RPR2   | UA2RPR1   | UA2RPR0   |
| FFFFF1B2H | UA2TIC   | UA2TIF   | UA2TMK   | 0 | 0 | 0 | UA2TPR2   | UA2TPR1   | UA2TPR0   |
| FFFFF1B4H | CB2REIC  | CB2REIF  | CB2REMK  | 0 | 0 | 0 | CB2REPR2  | CB2REPR1  | CB2REPR0  |
| FFFFF1B6H | CB2RIC   | CB2RIF   | CB2RMK   | 0 | 0 | 0 | CB2RPR2   | CB2RPR1   | CB2RPR0   |
| FFFFF1B8H | CB2TIC   | CB2TIF   | CB2TMK   | 0 | 0 | 0 | CB2TPR2   | CB2TPR1   | CB2TPR0   |

Table 20-2. Addresses and Bits of Interrupt Control Registers (3/3)

| Address    | Register | Bit      |          |   |   |   |           |           |           |
|------------|----------|----------|----------|---|---|---|-----------|-----------|-----------|
|            |          | <7>      | <6>      | 5 | 4 | 3 | 2         | 1         | 0         |
| FFFFFF1BAH | IICIC    | IICIF    | IICMK    | 0 | 0 | 0 | IICPR2    | IICPR1    | IICPR0    |
| FFFFFF1BCH | AD0IC    | AD0IF    | AD0MK    | 0 | 0 | 0 | AD0PR2    | AD0PR1    | AD0PR0    |
| FFFFFF1BEH | AD1IC    | AD1IF    | AD1MK    | 0 | 0 | 0 | AD1PR2    | AD1PR1    | AD1PR0    |
| FFFFFF1C0H | AD2IC    | AD2IF    | AD2MK    | 0 | 0 | 0 | AD2PR2    | AD2PR1    | AD2PR0    |
| FFFFFF1C2H | TM0EQIC0 | TM0EQIF0 | TM0EQMK0 | 0 | 0 | 0 | TM0EQPR02 | TM0EQPR01 | TM0EQPR00 |
| FFFFFF1C4H | TM1EQIC0 | TM1EQIF0 | TM1EQMK0 | 0 | 0 | 0 | TM1EQPR02 | TM1EQPR01 | TM1EQPR00 |
| FFFFFF1C6H | TM2EQIC0 | TM2EQIF0 | TM2EQMK0 | 0 | 0 | 0 | TM2EQPR02 | TM2EQPR01 | TM2EQPR00 |
| FFFFFF1C8H | TM3EQIC0 | TM3EQIF0 | TM3EQMK0 | 0 | 0 | 0 | TM3EQPR02 | TM3EQPR01 | TM3EQPR00 |
| FFFFFF1CAH | ADT0IC   | ADT0IF   | ADT0MK   | 0 | 0 | 0 | ADT0PR2   | ADT0PR1   | ADT0PR0   |
| FFFFFF1CCH | ADT1IC   | ADT1IF   | ADT1MK   | 0 | 0 | 0 | ADT1PR2   | ADT1PR1   | ADT1PR0   |

### 20.3.5 Interrupt mask registers 0 to 5 (IMR0 to IMR5)

The IMR0 to IMR5 registers set the interrupt mask state for the maskable interrupts. The IMR0.xxMKn to IMR3.xxMKn bits are equivalent to the xxICn.xxMKn bit.

The IMRm register can be read or written in 16-bit units (m = 0 to 5).

If the higher 8 bits of the IMRm register are used as the IMRmH register and the lower 8 bits as the IMRmL register, these registers can be read or written in 8-bit or 1-bit units.

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: IMR5 FFFF10AH  
IMR5L FFFF10AH, IMR5H FFFF10BH

|                                 |       |        |        |          |          |          |          |        |
|---------------------------------|-------|--------|--------|----------|----------|----------|----------|--------|
|                                 | 15    | 14     | 13     | 12       | 11       | 10       | 9        | 8      |
| IMR5 (IMR5H <sup>Note 1</sup> ) | 1     | ADT1MK | ADT0MK | TM3EQMK0 | TM2EQMK0 | TM1EQMK0 | TM0EQMK0 | AD2MK  |
|                                 | 7     | 6      | 5      | 4        | 3        | 2        | 1        | 0      |
| (IMR5L)                         | AD1MK | AD0MK  | IICMK  | CB2TMK   | CB2RMK   | CB2REMK  | UA2TMK   | UA2RMK |

After reset: FFFFH R/W Address: IMR4 FFFF108H  
IMR4L FFFF108H, IMR4H FFFF109H

|                                 |         |         |        |         |         |        |         |        |
|---------------------------------|---------|---------|--------|---------|---------|--------|---------|--------|
|                                 | 15      | 14      | 13     | 12      | 11      | 10     | 9       | 8      |
| IMR4 (IMR4H <sup>Note 1</sup> ) | UA2REMK | CB1TMK  | CB1RMK | CB1REMK | UA1TMK  | UA1RMK | UA1REMK | CB0TMK |
|                                 | 7       | 6       | 5      | 4       | 3       | 2      | 1       | 0      |
| (IMR4L)                         | CB0RMK  | CB0REMK | UA0TMK | UA0RMK  | UA0REMK | UTOMK  | UIFMK   | UTMK   |

After reset: FFFFH R/W Address: IMR3 FFFF106H  
IMR3L FFFF106H, IMR3H FFFF107H

|                                 |         |          |          |         |          |          |          |          |
|---------------------------------|---------|----------|----------|---------|----------|----------|----------|----------|
|                                 | 15      | 14       | 13       | 12      | 11       | 10       | 9        | 8        |
| IMR3 (IMR3H <sup>Note 1</sup> ) | URMK    | UREMK    | DMAMK3   | DMAMK2  | DMAMK1   | DMAMK0   | TA4CCMK1 | TA4CCMK0 |
|                                 | 7       | 6        | 5        | 4       | 3        | 2        | 1        | 0        |
| (IMR3L)                         | TA4OVMK | TA3CCMK1 | TA3CCMK0 | TA3OVMK | TA2CCMK1 | TA2CCMK0 | TA2OVMK  | TA1CCMK1 |

After reset: FFFFH R/W Address: IMR2 FFFF104H  
IMR2L FFFF104H, IMR2H FFFF105H

|                                 |          |                            |          |          |         |          |          |          |
|---------------------------------|----------|----------------------------|----------|----------|---------|----------|----------|----------|
|                                 | 15       | 14                         | 13       | 12       | 11      | 10       | 9        | 8        |
| IMR2 (IMR2H <sup>Note 1</sup> ) | TA1CCMK0 | TA1OVMK                    | TA0CCMK1 | TA0CCMK0 | TA0OVMK | TT1IECMK | TT1CCMK1 | TT1CCMK0 |
|                                 | 7        | 6                          | 5        | 4        | 3       | 2        | 1        | 0        |
| (IMR2L)                         | TT1OVMK  | TT0IECMK <sup>Note 2</sup> | TT0CCMK1 | TT0CCMK0 | TT0OVMK | TB1CCMK3 | TB1CCMK2 | TB1CCMK1 |

After reset: FFFFH R/W Address: IMR1 FFFF102H  
IMR1L FFFF102H, IMR1H FFFF103H

|                                 |          |         |          |          |          |          |         |         |
|---------------------------------|----------|---------|----------|----------|----------|----------|---------|---------|
|                                 | 15       | 14      | 13       | 12       | 11       | 10       | 9       | 8       |
| IMR1 (IMR1H <sup>Note 1</sup> ) | TB1CCMK0 | TB1OVMK | TB0CCMK3 | TB0CCMK2 | TB0CCMK1 | TB0CCMK0 | TB0OVMK | CMPMK1F |
|                                 | 7        | 6       | 5        | 4        | 3        | 2        | 1       | 0       |
| (IMR1L)                         | CMPMK1L  | CMPMK0F | CMPMK0L  | PMK18    | PMK17    | PMK16    | PMK15   | PMK14   |

After reset: FFFFH R/W Address: IMR0 FFFF100H  
IMR0L FFFF100H, IMR0H FFFF101H

|                                 |                         |                         |                         |                         |       |       |                         |                         |
|---------------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------|-------|-------------------------|-------------------------|
|                                 | 15                      | 14                      | 13                      | 12                      | 11    | 10    | 9                       | 8                       |
| IMR0 (IMR0H <sup>Note 1</sup> ) | PMK13                   | PMK12                   | PMK11                   | PMK10                   | PMK09 | PMK08 | PMK07 <sup>Note 2</sup> | PMK06 <sup>Note 2</sup> |
|                                 | 7                       | 6                       | 5                       | 4                       | 3     | 2     | 1                       | 0                       |
| (IMR0L)                         | PMK05 <sup>Note 2</sup> | PMK04 <sup>Note 2</sup> | PMK03 <sup>Note 2</sup> | PMK02 <sup>Note 2</sup> | PMK01 | PMK00 | LVIHMK                  | LVILMK                  |

- Notes 1.** When reading/writing bits 15 to 8 of the IMR0 to IMR5 registers in 8-bit or 1-bit units, specify these bits as bits 7 to 0 of the IMR0H to IMR5H registers.
- 2.** These bits are valid only in the V850E/IG3.  
Be sure to set these bits to 1 in the V850E/IF3.

**Caution** Set bit 15 of the IMR5 register (bit 7 of IMR5H register) to 1. The operation when these settings are changed is not guaranteed.



| xxMKn | Interrupt mask flag setting  |
|-------|------------------------------|
| 0     | Interrupt servicing enabled  |
| 1     | Interrupt servicing disabled |

**Remark** xx: Identification name of each peripheral unit (see **Table 20-2**)  
n: Peripheral unit number (see **Table 20-2**)

**20.3.6 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 signal request 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 cleared to 0 by hardware. However, it is not cleared 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** In the interrupt enabled (EI) state, if an interrupt is acknowledged during the reading of the ISPR register, the value of the ISPR register may be read after the bit is set (1) by this interrupt acknowledgment. To read the value of the ISPR register properly before interrupt acknowledgment, read it in the interrupt disabled (DI) state.

|                  |  |                  |                   |                  |                  |                  |                  |                  |
|------------------|--|------------------|-------------------|------------------|------------------|------------------|------------------|------------------|
| After reset: 00H |  | R                | Address: FFFF1FAH |                  |                  |                  |                  |                  |
| ISPR             | <b>&lt;7&gt;</b>   | <b>&lt;6&gt;</b> | <b>&lt;5&gt;</b>  | <b>&lt;4&gt;</b> | <b>&lt;3&gt;</b> | <b>&lt;2&gt;</b> | <b>&lt;1&gt;</b> | <b>&lt;0&gt;</b> |
|                  | ISPR7  | ISPR6            | ISPR5             | ISPR4            | ISPR3            | ISPR2            | ISPR1            | ISPR0            |
| ISPRn            | Priority of interrupt currently being acknowledged             |                  |                   |                  |                  |                  |                  |                  |
| 0                | Interrupt request signal with priority n is not acknowledged   |                  |                   |                  |                  |                  |                  |                  |
| 1                | Interrupt request signal with priority n is being acknowledged |                  |                   |                  |                  |                  |                  |                  |

**Remark** n: 0 to 7 (priority level)

**20.3.7 Maskable interrupt status flag (ID)**

The ID flag controls the maskable interrupt's operating state, and stores control information regarding enabling or disabling of interrupt requests. The ID flag is allocated to the PSW.

Reset sets this flag to 00000020H.

After rest: 00000020H

|     |    |  |   |   |   |   |    |    |    |     |    |    |   |   |
|-----|----|--|---|---|---|---|----|----|----|-----|----|----|---|---|
|     | 31 |  | 8 | 7 | 6 | 5 | 4  | 3  | 2  | 1   | 0  |    |   |   |
| PSW | 0  |  |   |   |   |   | NP | EP | ID | SAT | CY | OV | S | Z |

|    |   |
|----|---|
| ID | Maskable interrupt servicing specification <sup>Note</sup>          |
| 0  | Maskable interrupt request signal acknowledgment enabled            |
| 1  | Maskable interrupt request signal acknowledgment disabled (pending) |

**Note** Interrupt disable flag (ID) function  
 ID is set (1) by the DI instruction and cleared (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 (1) by hardware.  
 An interrupt request signal generated during the acknowledgment disabled period (ID flag = 1) can be acknowledged when the xxICn.xxIFn bit is set (1), and the ID flag is cleared (0).

## 20.4 External Interrupt Request Input Pins (INTP00 to INTP18, INTADT0, INTADT1)

### 20.4.1 Noise elimination

#### (1) Noise elimination of INTP00, INTP01, INTPa (V850E/IG3 only), INTP08 to INTP13, INTP17, INTP18, INTADT0, and INTADT1 pins

The INTP00, INTP01, INTPa (V850E/IG3 only), INTP08 to INTP13, INTP17, INTP18, INTADT0, and INTADT1 pins incorporate a noise eliminator that uses analog filter (a = 02 to 07). Unless, therefore, the input level of each pin is held for a certain time, an edge cannot be detected. An edge is detected after a certain time has elapsed.

#### (2) Noise elimination of INTP14 to INTP16 pins

The INTP14 to INTP16 pins incorporate a digital noise eliminator.

The sampling clock that performs digital sampling can be selected by the INTNFCm.INTNFCm2 to INTNFCm.INTNFCm0 bits (m = 14 to 16).

The system clock stops in the IDLE and STOP modes, so the INTP14 to INTP16 pins cannot be used to cancel the IDLE and STOP modes.

### 20.4.2 Edge detection

The valid edges of the INTn pin can be selected by program (V850E/IF3: n = P00, P01, P08 to P18, ADT0, ADT1, V850E/IG3: n = P00 to P18, ADT0, ADT1). The edge that can be selected as the valid edge is one of the following.

- Rising edge
- Falling edge
- Both the rising and falling edges

The edge-detected INTn signal becomes an interrupt source.

The valid edge is specified by the INTR0 to INTR2, ADTR, INTF0 to INTF2, and ADTF registers.

**(1) External interrupt rising edge specification register 0 (INTR0), external interrupt falling edge specification register 0 (INTF0)**

The INTR0 and INTF0 registers are used to specify the trigger mode of the INTP00, INTP01, and INTPa (V850E/IG3 only) pins and can specify the valid edge independently for each pin (rising edge, falling edge, or both rising and falling edges) (a = 02 to 07).

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 mode, an edge may be detected. Therefore, be sure to clear the INTF0n and INTR0n bits to 00, and then set the port mode (V850E/IF3: n = 0, 1, V850E/IG3: n = 0 to 7).

|                        |   |                        |                        |                        |                        |        |        |     |     |                        |                        |                        |                        |                        |                        |        |        |
|------------------------|---|------------------------|------------------------|------------------------|------------------------|--------|--------|-----|-----|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|--------|--------|
| After reset: 00H       | R/W   | Address: FFFFC20H      |                        |                        |                        |        |        |     |     |                        |                        |                        |                        |                        |                        |        |        |
| INTR0                  | <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">&lt;7&gt;</td> <td style="text-align: center;">&lt;6&gt;</td> <td style="text-align: center;">&lt;5&gt;</td> <td style="text-align: center;">&lt;4&gt;</td> <td style="text-align: center;">&lt;3&gt;</td> <td style="text-align: center;">&lt;2&gt;</td> <td style="text-align: center;">&lt;1&gt;</td> <td style="text-align: center;">&lt;0&gt;</td> </tr> <tr> <td style="text-align: center;">INTR07<sup>Note</sup></td> <td style="text-align: center;">INTR06<sup>Note</sup></td> <td style="text-align: center;">INTR05<sup>Note</sup></td> <td style="text-align: center;">INTR04<sup>Note</sup></td> <td style="text-align: center;">INTR03<sup>Note</sup></td> <td style="text-align: center;">INTR02<sup>Note</sup></td> <td style="text-align: center;">INTR01</td> <td style="text-align: center;">INTR00</td> </tr> </table> | <7>                    | <6>                    | <5>                    | <4>                    | <3>    | <2>    | <1> | <0> | INTR07 <sup>Note</sup> | INTR06 <sup>Note</sup> | INTR05 <sup>Note</sup> | INTR04 <sup>Note</sup> | INTR03 <sup>Note</sup> | INTR02 <sup>Note</sup> | INTR01 | INTR00 |
| <7>                    | <6>   | <5>                    | <4>                    | <3>                    | <2>                    | <1>    | <0>    |     |     |                        |                        |                        |                        |                        |                        |        |        |
| INTR07 <sup>Note</sup> | INTR06 <sup>Note</sup>  | INTR05 <sup>Note</sup> | INTR04 <sup>Note</sup> | INTR03 <sup>Note</sup> | INTR02 <sup>Note</sup> | INTR01 | INTR00 |     |     |                        |                        |                        |                        |                        |                        |        |        |
| After reset: 00H       | R/W   | Address: FFFFC00H      |                        |                        |                        |        |        |     |     |                        |                        |                        |                        |                        |                        |        |        |
| INTF0                  | <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">&lt;7&gt;</td> <td style="text-align: center;">&lt;6&gt;</td> <td style="text-align: center;">&lt;5&gt;</td> <td style="text-align: center;">&lt;4&gt;</td> <td style="text-align: center;">&lt;3&gt;</td> <td style="text-align: center;">&lt;2&gt;</td> <td style="text-align: center;">&lt;1&gt;</td> <td style="text-align: center;">&lt;0&gt;</td> </tr> <tr> <td style="text-align: center;">INTF07<sup>Note</sup></td> <td style="text-align: center;">INTF06<sup>Note</sup></td> <td style="text-align: center;">INTF05<sup>Note</sup></td> <td style="text-align: center;">INTF04<sup>Note</sup></td> <td style="text-align: center;">INTF03<sup>Note</sup></td> <td style="text-align: center;">INTF02<sup>Note</sup></td> <td style="text-align: center;">INTF01</td> <td style="text-align: center;">INTF00</td> </tr> </table> | <7>                    | <6>                    | <5>                    | <4>                    | <3>    | <2>    | <1> | <0> | INTF07 <sup>Note</sup> | INTF06 <sup>Note</sup> | INTF05 <sup>Note</sup> | INTF04 <sup>Note</sup> | INTF03 <sup>Note</sup> | INTF02 <sup>Note</sup> | INTF01 | INTF00 |
| <7>                    | <6>   | <5>                    | <4>                    | <3>                    | <2>                    | <1>    | <0>    |     |     |                        |                        |                        |                        |                        |                        |        |        |
| INTF07 <sup>Note</sup> | INTF06 <sup>Note</sup>  | INTF05 <sup>Note</sup> | INTF04 <sup>Note</sup> | INTF03 <sup>Note</sup> | INTF02 <sup>Note</sup> | INTF01 | INTF00 |     |     |                        |                        |                        |                        |                        |                        |        |        |

**Note** Valid only in the V850E/IG3.  
In the V850E/IF3, be sure to set these bits to 0.

**Remark** For the valid edge specification, see **Table 20-3**.

**Table 20-3. Valid Edge Specification of INTP00 to INTP07 Pins**

| INTF0n | INTR0n | Valid Edge Specification      |
|--------|--------|-------------------------------|
| 0      | 0      | No edge detected              |
| 0      | 1      | Rising edge                   |
| 1      | 0      | Falling edge                  |
| 1      | 1      | Both rising and falling edges |

**Caution** When not using these pins as the INTP0n pins, be sure to set the INTF0n and INTR0n bits to 00.

**Remark** V850E/IF3: n = 0, 1  
V850E/IG3: n = 0 to 7

**(2) External interrupt rising edge specification register 1 (INTR1), external interrupt falling edge specification register 1 (INTF1)**

The INTR1 and INTF1 registers are used to specify the trigger mode of the INTP08 to INTP13, INTP17, and INTP18 pins and can specify the valid edge independently for each pin (rising edge, falling edge, or both rising and falling edges).

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 mode, an edge may be detected. Therefore, be sure to clear the INTFn and INTRn bits to 00, and then set the port mode (n = 08 to 13, 17, 18).

|               |   |        |                   |        |        |        |        |        |
|---------------|---|--------|-------------------|--------|--------|--------|--------|--------|
|               | After reset: 00H  | R/W    | Address: FFFFC22H |        |        |        |        |        |
|               | <7>   | <6>    | <5>               | <4>    | <3>    | <2>    | <1>    | <0>    |
| INTR1         | INTR18  | INTR17 | INTR13            | INTR12 | INTR11 | INTR10 | INTR09 | INTR08 |
|               |   |        |                   |        |        |        |        |        |
|               | <7>   | <6>    | <5>               | <4>    | <3>    | <2>    | <1>    | <0>    |
| INTF1         | INTF18  | INTF17 | INTF13            | INTF12 | INTF11 | INTF10 | INTF09 | INTF08 |
|               |   |        |                   |        |        |        |        |        |
| <b>Remark</b> | For the valid edge specification, see <b>Table 20-4</b> . |        |                   |        |        |        |        |        |

**Table 20-4. Valid Edge Specification of INTP08 to INTP13, INTP17, and INTP18 Pins**

| INTFn | INTRn | Valid Edge Specification      |
|-------|-------|-------------------------------|
| 0     | 0     | No edge detected              |
| 0     | 1     | Rising edge                   |
| 1     | 0     | Falling edge                  |
| 1     | 1     | Both rising and falling edges |

**Caution** When not using these pins as the INTPn pins, be sure to set the INTFn and INTRn bits to 00.

**Remark** n = 08 to 13, 17, 18

**(3) External interrupt rising edge specification register 2 (INTR2), external interrupt falling edge specification register 2 (INTF2)**

The INTR2 and INTF2 registers are used to specify the trigger mode of the INTP14 to INTP16 pins and can specify the valid edge independently for each pin (rising edge, falling edge, or both rising and falling edges).

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 mode, an edge may be detected. Therefore, be sure to clear the INTF1n and INTR1n bits to 00, and then set the port mode (n = 4 to 6).

|       |                  |     |                      |   |   |        |        |        |
|-------|------------------|-----|----------------------|---|---|--------|--------|--------|
|       | After reset: 00H | R/W | Address: FFFFFFFC24H |   |   |        |        |        |
|       | 7                | 6   | 5                    | 4 | 3 | <2>    | <1>    | <0>    |
| INTR2 | 0                | 0   | 0                    | 0 | 0 | INTR16 | INTR15 | INTR14 |
|       |                  |     |                      |   |   |        |        |        |
|       | 7                | 6   | 5                    | 4 | 3 | <2>    | <1>    | <0>    |
| INTF2 | 0                | 0   | 0                    | 0 | 0 | INTF16 | INTF15 | INTF14 |

**Remark** For the valid edge specification, see **Table 20-5**.

**Table 20-5. Valid Edge Specification of INTP14 to INTP16 Pins**

| INTF1n | INTR1n | Valid Edge Specification      |
|--------|--------|-------------------------------|
| 0      | 0      | No edge detected              |
| 0      | 1      | Rising edge                   |
| 1      | 0      | Falling edge                  |
| 1      | 1      | Both rising and falling edges |

**Caution** When not using these pins as the INTP1n pins, be sure to set the INTF1n and INTR1n bits to 00.

**Remark** n = 4 to 6

**(4) A/D trigger rising edge, falling edge specification registers (ADTR, ADTF)**

The ADTR and ADTF registers are registers that specify the trigger mode of the ADTRG0/INTADT0 and ADTRG1/INTADT1 pins and can specify the valid edge independently for each pin (rising edge, falling edge, or both rising and falling edges).

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 trigger input of the A/D converter n (alternate function)/external interrupt function (alternate function) to the port mode, an edge may be detected. Therefore, be sure to set the ADTFn and ADTRn bits to 00, and then set the port mode.

|   |   |   |   |   |   |   |       |       |
|---|---|---|---|---|---|---|-------|-------|
| After reset: 00H R/W Address: FFFFF2F2H                                 |   |   |   |   |   |   |       |       |
| ADTR  | 7 | 6 | 5 | 4 | 3 | 2 | 1     | 0     |
|   | 0 | 0 | 0 | 0 | 0 | 0 | ADTR1 | ADTR0 |
| After reset: 00H R/W Address: FFFFF2F0H                                 |   |   |   |   |   |   |       |       |
| ADTF  | 7 | 6 | 5 | 4 | 3 | 2 | 1     | 0     |
|   | 0 | 0 | 0 | 0 | 0 | 0 | ADTF1 | ADTF0 |
| <b>Remark</b> For the valid edge specification, see <b>Table 20-6</b> . |   |   |   |   |   |   |       |       |

**Table 20-6. Valid Edge Specification of ADTRG0/INTADT0 and ADTRG1/INTADT1 Pins**

| ADTFn | ADTRn | Valid Edge Specification      |
|-------|-------|-------------------------------|
| 0     | 0     | No edge detected              |
| 0     | 1     | Rising edge                   |
| 1     | 0     | Falling edge                  |
| 1     | 1     | Both rising and falling edges |

**Caution** When not using these pins as the ADTRGn/INTADTn pins, be sure to set the ADTFn and ADTRn bits to 00.

**Remark** n = 0, 1



## 20.5 Software Exception

A software exception is generated when the CPU executes the TRAP instruction, and can always be acknowledged.

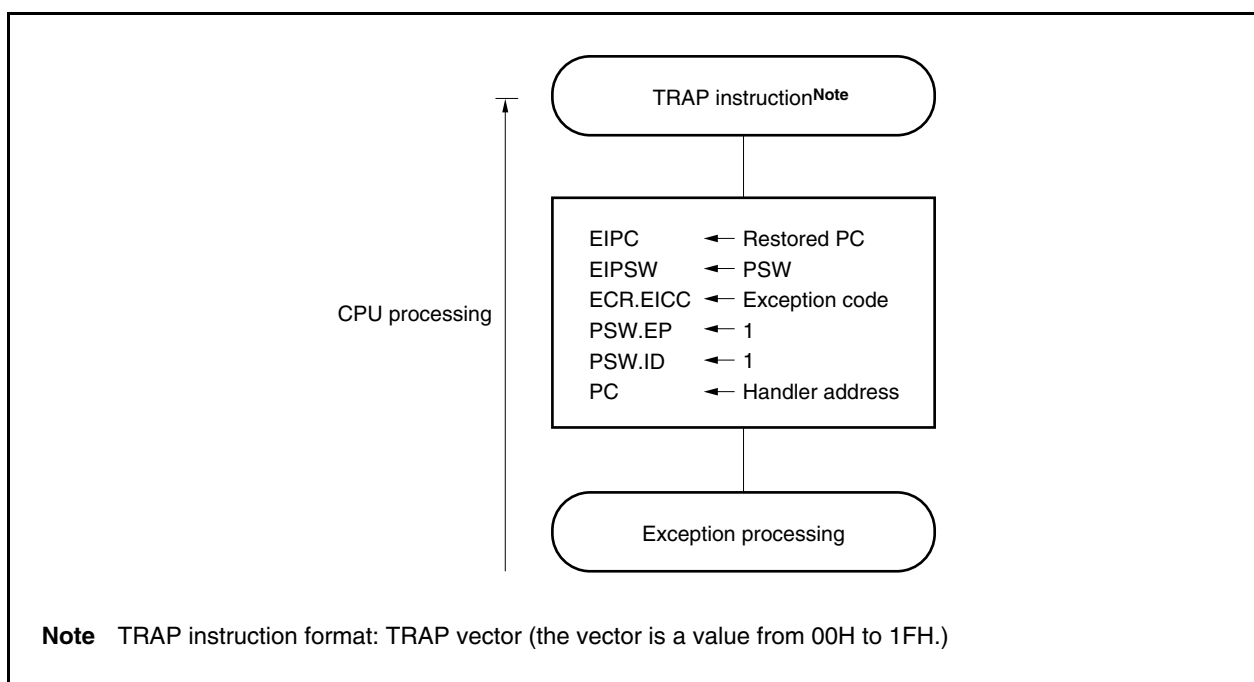
### 20.5.1 Operation

If a software exception occurs, the CPU performs the following processing, and transfers control to the handler routine.

- <1> Saves the restored PC to EIPC.
- <2> Saves the current PSW to EIPSW.
- <3> Writes an exception code to the lower 16 bits (EICC) of ECR (interrupt source).
- <4> Sets the PSW.EP and PSW.ID bits (1).
- <5> Sets the handler address (00000040H or 00000050H) corresponding to the software exception to the PC, and transfers control.

The processing of a software exception is shown below.

**Figure 20-8. 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.

**20.5.2 Restore**

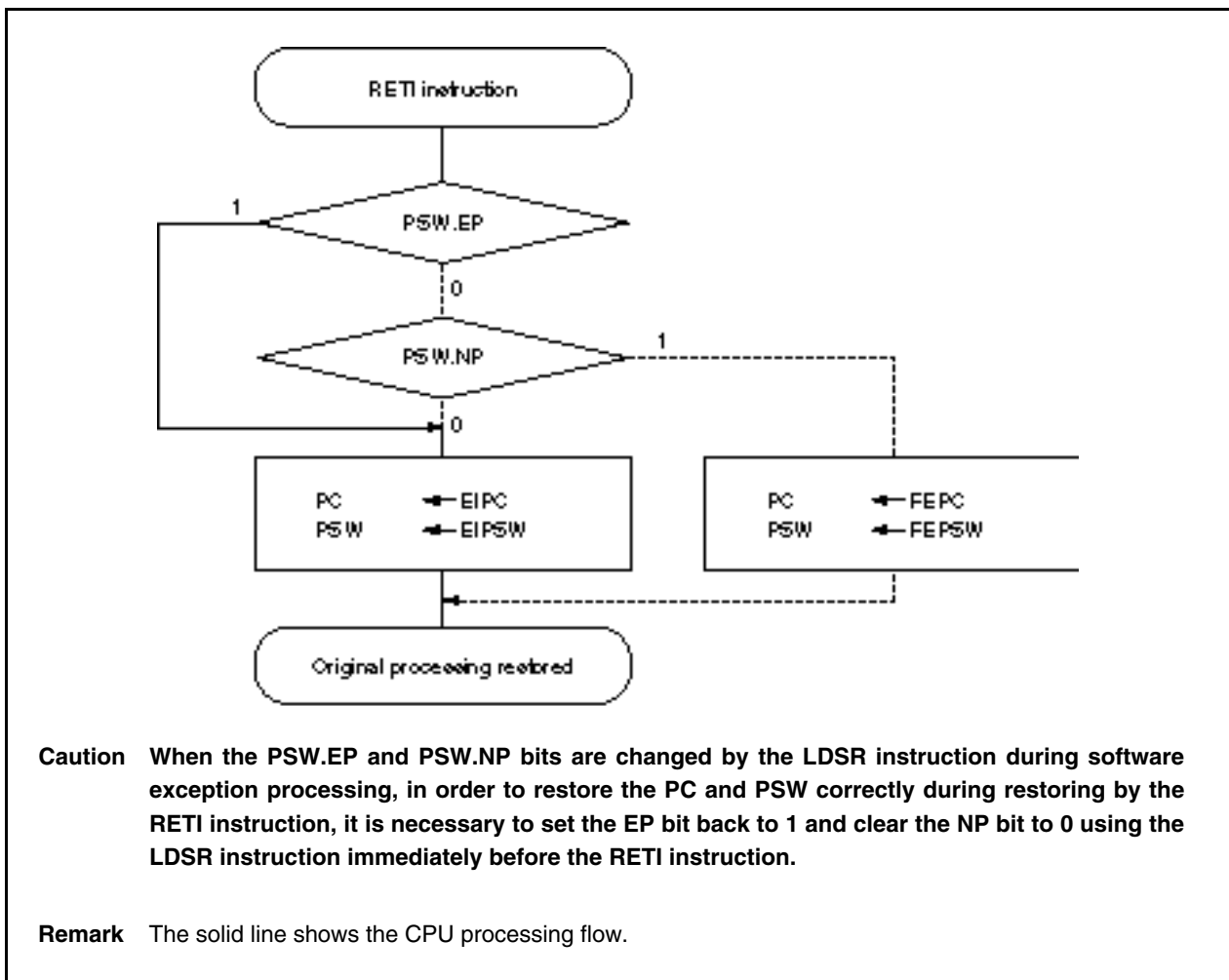
Execution is restored from software exception processing by the RETI instruction.

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 EIPC and EIPSW because the PSW.EP bit is 1.
- <2> Transfers control to the address of the restored PC and PSW.

The processing of the RETI instruction is shown below.

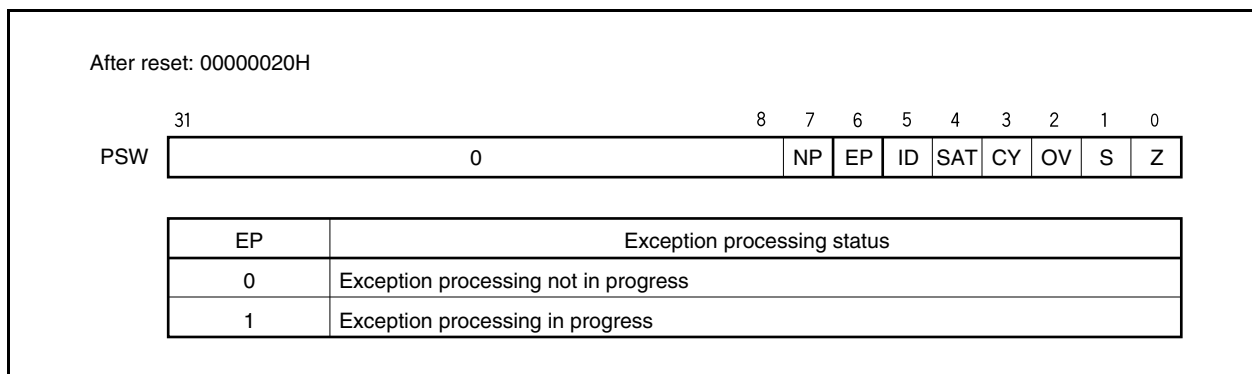
**Figure 20-9. RETI Instruction Processing**



### 20.5.3 Exception status flag (EP)

The EP flag is a status flag used to indicate that exception processing is in progress. This flag is set when an exception occurs. The EP flag is allocated to the PSW.

This flag is set to 00000020H after reset.

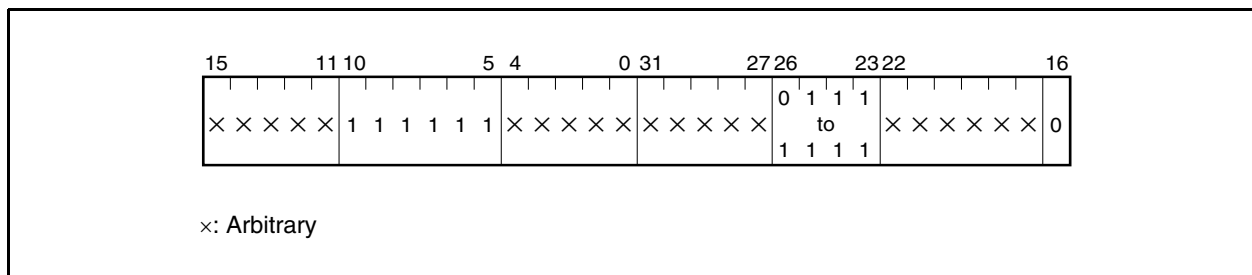


### 20.6 Exception Trap

An exception trap is an interrupt that is requested when the illegal execution of an instruction takes place. In the V850E/IF3 and V850E/IG3, an illegal opcode trap (ILGOP: Illegal Opcode Trap) is considered as an exception trap.

#### 20.6.1 Illegal opcode definition

The illegal instruction has an opcode (bits 10 to 5) of 111111B, a sub-opcode (bits 26 to 23) of 0111B to 1111B, and a sub-opcode (bit 16) of 0B. An exception trap is generated when an instruction applicable to this illegal instruction is executed.



**Caution** Since it is possible that this instruction may be assigned to an illegal opcode in the future, it is recommended that it not be used.

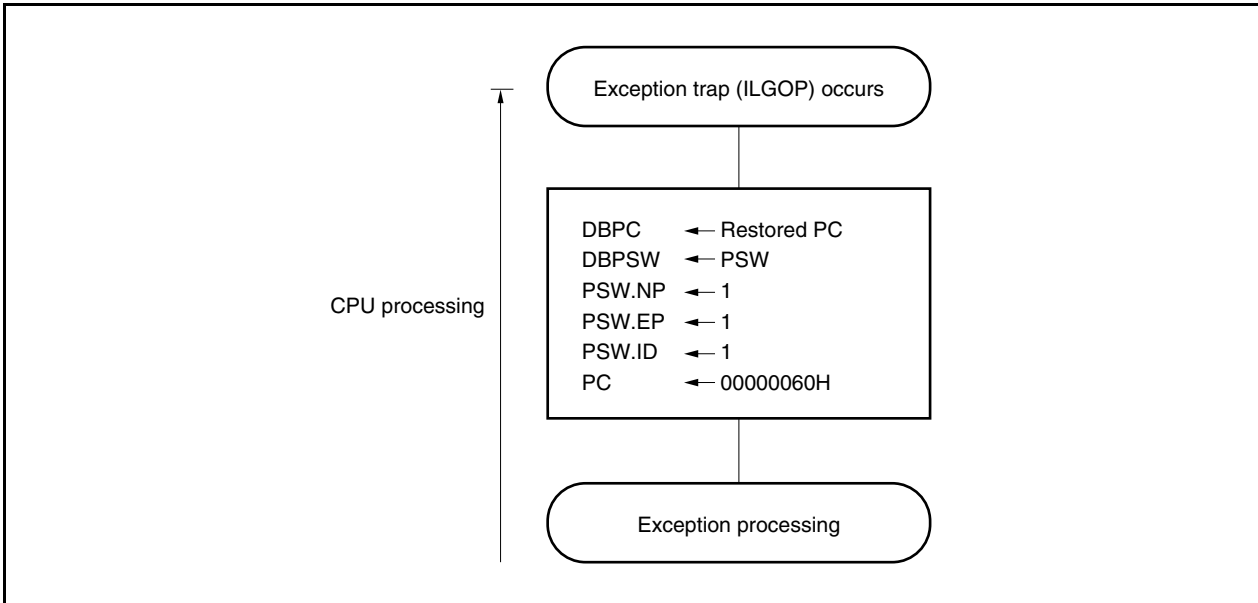
#### (1) Operation

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 (1).
- <4> Sets the handler address (00000060H) corresponding to the exception trap to the PC, and transfers control.

The processing of the exception trap is shown below.

Figure 20-10. Exception Trap Processing



**(2) Restore**

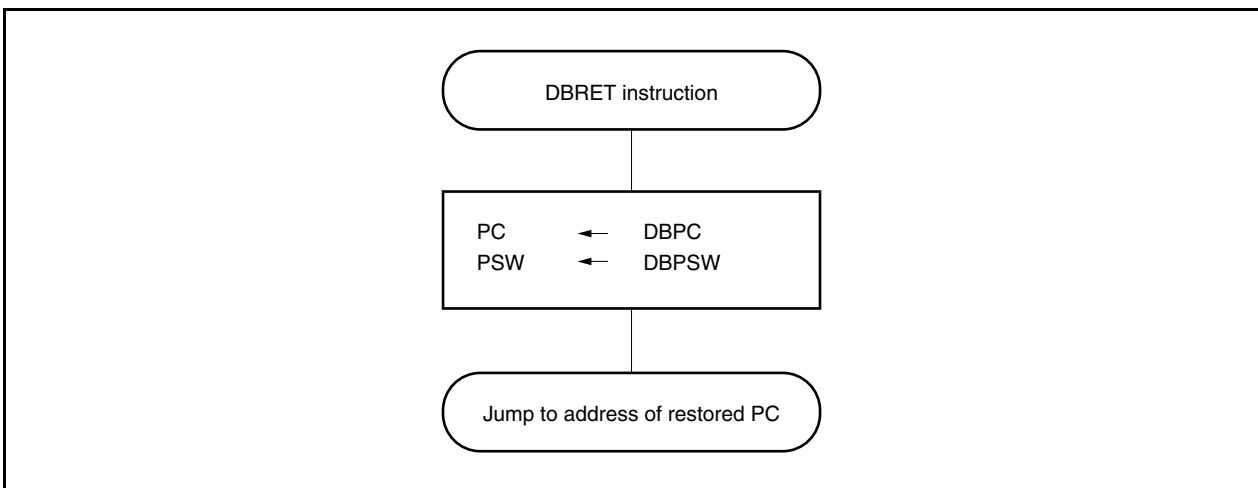
Execution is restored from an exception trap by the DBRET instruction. When the DBRET instruction is executed, the CPU performs the following processing, and transfers control to the address of the restored PC.

- <1> Loads the restored PC and PSW from DBPC and DBPSW.
- <2> Transfers control to the address of the restored PC and PSW.

**Caution** DBPC and DBPSW can be accessed only during the period between when the illegal opcode is executed and when the DBRET instruction is executed.

The restore processing from an exception trap is shown below.

Figure 20-11. Restore Processing from Exception Trap



**20.6.2 Debug trap**

The debug trap is an exception that can be acknowledged anytime and is generated by execution of the DBTRAP instruction.

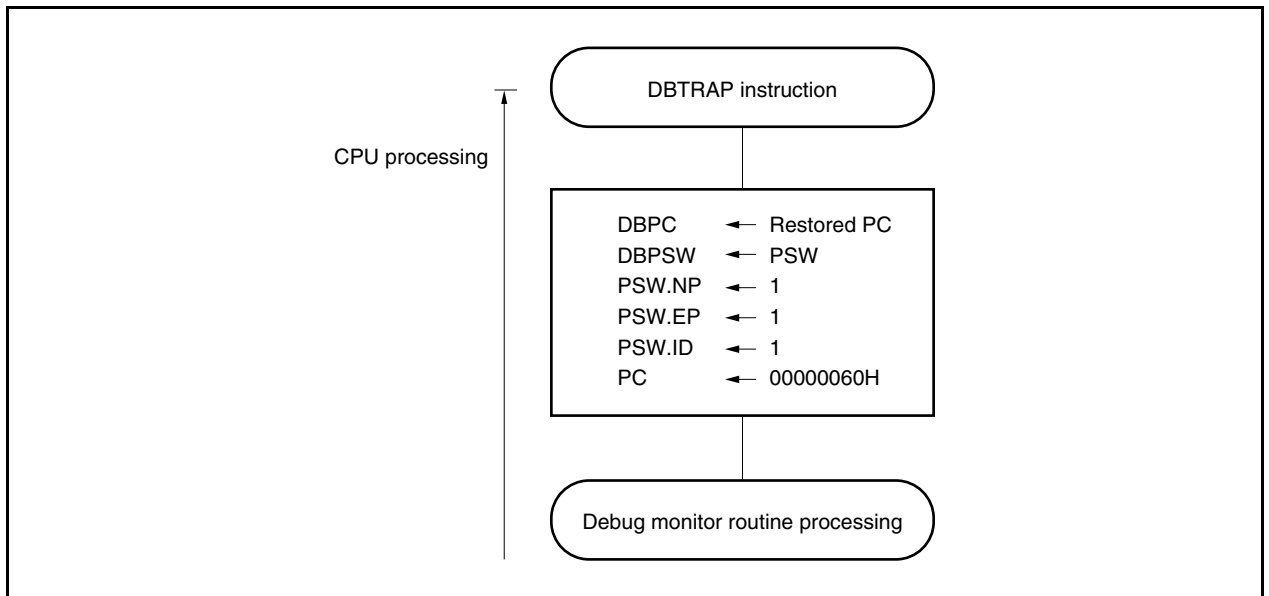
When the debug trap is generated, the CPU performs the following processing.

**(1) Operation**

- <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 (1).
- <4> Sets the handler address (00000060H) corresponding to the debug trap to the PC and transfers control.

The processing of the debug trap is shown below.

**Figure 20-12. Debug Trap Processing**



**(2) Restore**

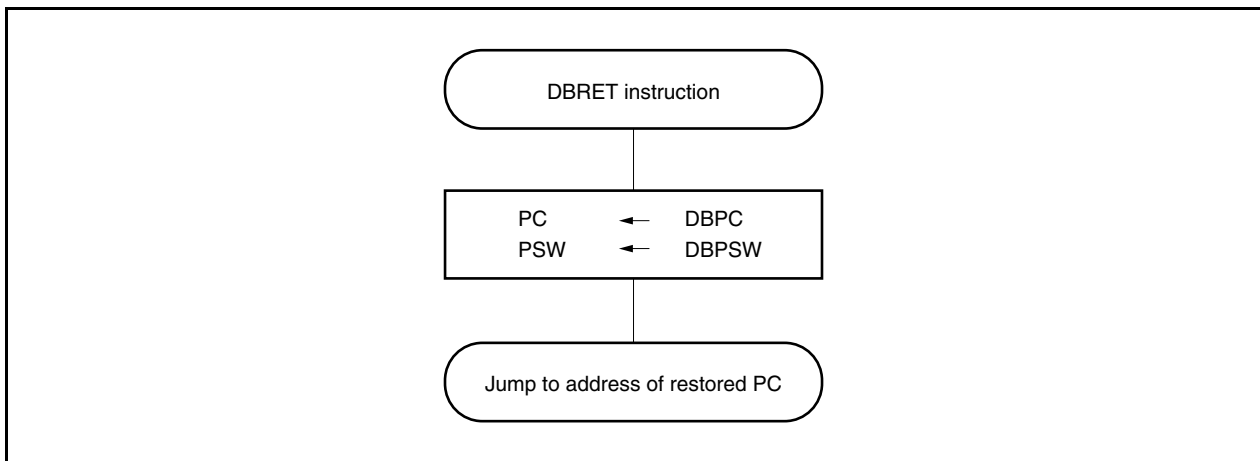
Execution is restored from a debug trap by the DBRET instruction. When the DBRET instruction is executed, the CPU performs the following processing and transfers control to the address of the restored PC.

- <1> Loads the restored PC and PSW from DBPC and DBPSW.
- <2> Transfers control to the address of the restored PC and PSW.

**Caution** DBPC and DBPSW can be accessed only during the period between when the DBTRAP is executed and when the DBRET instruction is executed.

The restore processing from a debug trap is shown below.

**Figure 20-13. Restore Processing from Debug Trap**



## 20.7 Multiple Interrupt Servicing Control

Multiple interrupt servicing control is a process by which an interrupt request that is currently being serviced can be interrupted during servicing if there is an interrupt request signal with a higher priority level, and the higher priority interrupt request signal is acknowledged and serviced first.

If there is an interrupt request signal with a lower priority level than the interrupt request currently being serviced, that interrupt request signal is held pending.

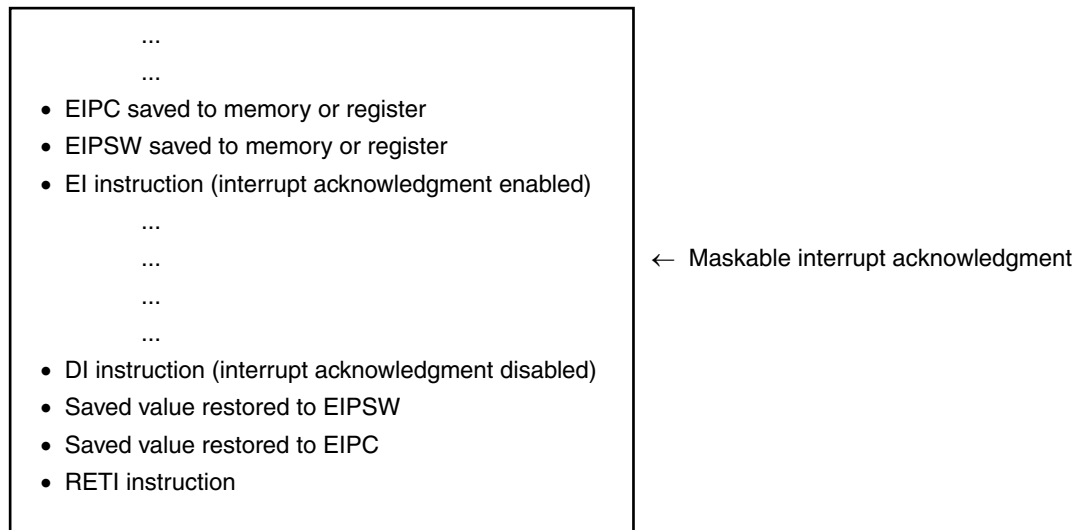
Multiple interrupt servicing control of maskable interrupts is executed when interrupts are enabled (PSW.ID bit = 0). Thus, to execute multiple interrupts, it is necessary to set the interrupt enabled state (PSW.ID bit = 0) even in an interrupt servicing routine.

If maskable interrupts are enabled or a software exception is generated in a maskable interrupt or software exception servicing program, it is necessary to save EIPC and EIPSW.

This is accomplished by the following procedure.

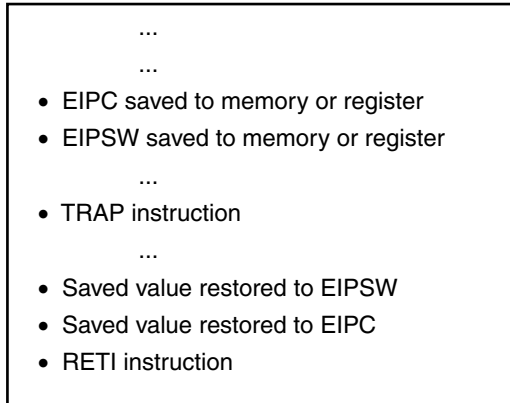
### (1) Acknowledgment of maskable interrupt signals in servicing program

Service program of maskable interrupt or exception



(2) Generation of exception in servicing program

Servicing program of maskable interrupt or exception



← Exception such as TRAP instruction acknowledged.

The priority order for multiple interrupt servicing control has 8 levels, from 0 to 7 for each maskable interrupt request signal (0 is the highest priority), but it can be set as desired via software. The priority order is set using the xxPRn0 to xxPRn2 bits of the interrupt control request register (xxICn), provided for each maskable interrupt request signal. After system reset, an interrupt request signal is masked by the xxMKn bit and the priority order is set to level 7 by the xxPRn0 to xxPRn2 bits.

The priority order of maskable interrupts is as follows.

(High) Level 0 > Level 1 > Level 2 > Level 3 > Level 4 > Level 5 > Level 6 > Level 7 (Low)

Interrupt servicing that has been suspended as a result of multiple servicing control is resumed after the servicing of the higher priority interrupt has been completed and the RETI instruction has been executed. A pending interrupt request signal is acknowledged after the current interrupt servicing has been completed and the RETI instruction has been executed.

**Caution** In a non-maskable interrupt servicing routine (time until the RETI instruction is executed), maskable interrupts are suspended and not acknowledged.

**Remark** xx: Identification name of each peripheral unit (see Table 20-2)  
n: Peripheral unit number (see Table 20-2)

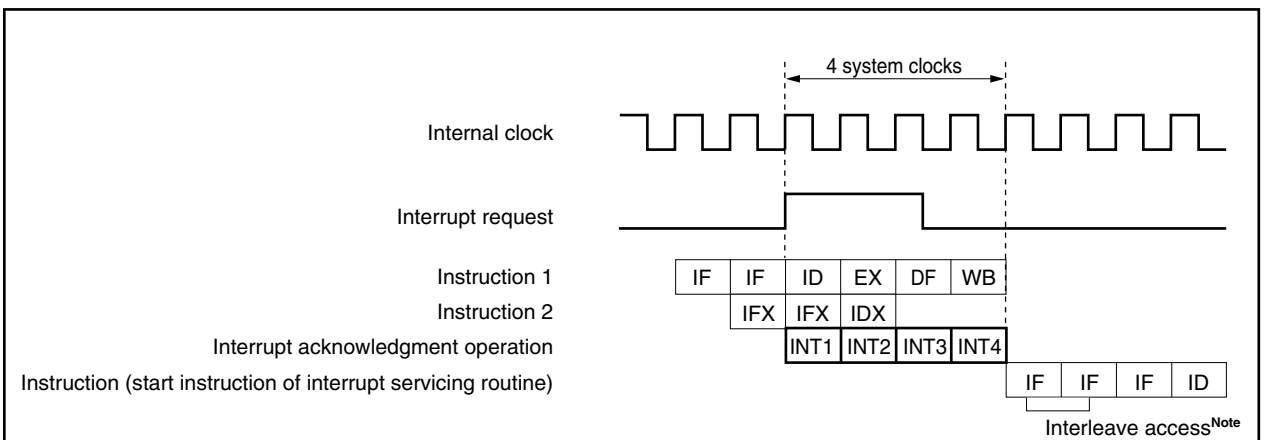


20.8 Interrupt Response Time of CPU

Except the following cases, the interrupt response time of the CPU is 4 clocks minimum. To input interrupt request signals successively, input the next interrupt request signal at least 4 clocks after the preceding interrupt.

- In IDLE/STOP mode
- When interrupt request non-sampling instructions are successively executed (see 20.9 Periods in Which CPU Does Not Acknowledge Interrupts.)
- When an on-chip peripheral I/O register is accessed

Figure 20-14. Pipeline Operation at Interrupt Request Acknowledgment (Outline)



**Note** For interleave accesses, refer to 8.1.2 2-clock branch in V850E1 Architecture User's Manual (U14559E).

**Remark** INT1 to INT4: Interrupt acknowledgment processing  
 IFX: Invalid instruction fetch  
 IDX: Invalid instruction decode

|         | Interrupt latency time (internal system clock) |  |  | Conditions   |
|---------|--|--|--|--|
|         | Internal interrupt                             | External interrupt   |  |  |
|         |  | INTP00, INTP01, INTPa <sup>Note 1</sup> , INTP08 to INTP13, INTP17, INTP18, INTADT0, INTADT1 | INTP14 to INTP16                         |  |
| Minimum | 4  | 4 + Analog filter time   | 4 + <b>Note 3</b> + Digital noise filter | The following cases are exceptions. <ul style="list-style-type: none"> <li>• In IDLE/STOP mode</li> <li>• Access to external bus</li> <li>• Two or more interrupt request non-sample instructions are executed in succession</li> <li>• Access to on-chip peripheral I/O register</li> </ul> |
| Maximum | 7 <sup>Note 2</sup>                            | 7 + Analog filter time   | 7 + <b>Note 3</b> + Digital noise filter |  |

- Notes**
1. V850E/IG3 only
  2. When LD instruction is executed to internal ROM (during align access)
  3. For the number of internal system clocks, see 4.6 (1) Digital noise elimination 0 control register n (INTNFCn).

**Remark** a = 02 to 07

## 20.9 Periods in Which CPU Does Not Acknowledge Interrupts

The CPU acknowledges an interrupt while an instruction is being executed. However, no interrupt will be acknowledged between an interrupt request non-sample instruction and the next instruction (interrupt is held pending).

The interrupt request non-sample instructions are as follows.

- EI instruction
- DI instruction
- LDSR reg2, 0x5 instruction (for PSW)
- Store instruction for the command register (PRCMD).
- Store instructions or bit manipulation instructions excluding tst1 instruction for the following registers.
  - Interrupt-related registers:  
Interrupt control register (xxICn) and interrupt mask registers 0 to 5 (IMR0 to IMR5)
  - Power save control register (PSC)

**Remark** xx: Identification name of each peripheral unit (see **Table 20-2**)

n: Peripheral unit number (see **Table 20-2**)

## 20.10 Caution

Note that if a port is set to external interrupt input (INTPn), the timer/counter-related interrupt, serial interface-related interrupt, and A/D converter-related interrupt, which are alternate functions, do not occur (V850E/IF3: n = 00, 01, 08 to 18, ADT0, ADT1, V850E/IG3: n = 00 to 18, ADT0, ADT1).

## CHAPTER 21 STANDBY FUNCTION

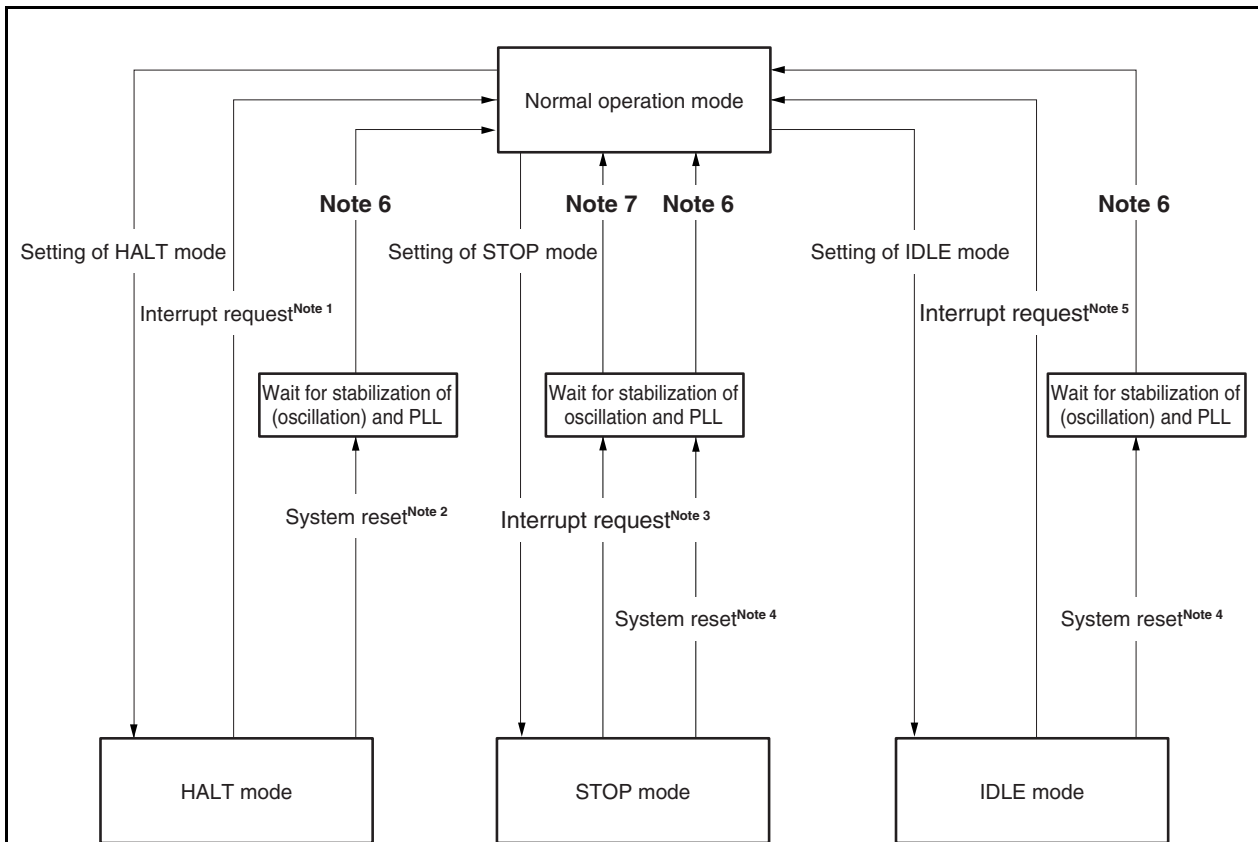
### 21.1 Overview

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 21-1.

**Table 21-1. Standby Modes**

| Mode      | Functional Outline  |
|-----------|---|
| HALT mode | Mode to stop only the operating clock of the CPU  |
| IDLE mode | Mode to stop all the operations of the internal circuit except the oscillator, PLL, CSIB in the slave mode, clock monitor, low-voltage detector (LVI), power-on-clear circuit (POC) |
| STOP mode | Mode to stop all the operations of the internal circuit except the CSIB in the slave mode, low-voltage detector (LVI), power-on-clear circuit (POC)                                 |

Figure 21-1. Status Transition



- Notes**
1. Non-maskable interrupt request signal (INTWDT) or unmasked maskable interrupt request signal
  2.  $\overline{\text{RESET}}$  pin input, reset signal (WDTRES) generation by watchdog timer overflow, reset signal (LVIREs) generation by low-voltage detector (LVI), or reset signal (POCREs) generation by power-on-clear circuit (POC)
  3. Unmasked external interrupt request signal (INTP00, INTP01, INTP02 to INTP07 (V850E/IG3 only), INTP08 to INTP13, INTP17, INTP18, INTADT0, or INTADT1) or unmasked internal interrupt request signal from (CSIB-related interrupt request signal in the slave mode) peripheral functions operable in STOP mode
  4.  $\overline{\text{RESET}}$  pin input, reset signal (LVIREs) generation by low-voltage detector (LVI), or reset signal (POCREs) generation by power-on-clear circuit (POC)
  5. Unmasked external interrupt request signal (INTP00, INTP01, INTP02 to INTP07 (V850E/IG3 only), INTP08 to INTP13, INTP17, INTP18, INTADT0, or INTADT1) or unmasked internal interrupt request signal (CSIB-related interrupt request signal in the slave mode) from peripheral functions operable in IDLE mode
  6. Oscillation stabilization time count by oscillation stabilization time wait control (OST)  
The oscillation stabilization time is necessary after release of reset because the PLL is initialized by a reset. The stabilization time is the time determined by default.
  7. Oscillation stabilization time count by oscillation stabilization time wait control (OST)  
The stabilization time is determined by the setting of the OSTs register.

21.2 Control Registers

(1) Power save control register (PSC)

The PSC register is an 8-bit register that controls the standby function. The STB bit of this register is used to specify the standby mode. This register is a special register (see 3.4.8 Special registers). This register can be written only by a combination of specific sequences.

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 | 0 | 0 | INTM | 0 | 0 | STB | 0 |

|      |   |
|------|---|
| INTM | Standby mode control <sup>Note 2</sup> by maskable interrupt request (INTxx <sup>Note 1</sup> ) |
| 0    | Standby mode release by INTxx request enabled   |
| 1    | Standby mode release by INTxx request disabled  |

|     |                     |
|-----|---------------------|
| STB | Sets operation mode |
| 0   | Normal mode         |
| 1   | Standby mode        |

**Notes 1.** For details, see **Table 20-1 Interrupt Source List**.

**2.** The setting is valid only in the IDLE mode and STOP mode.

**Cautions 1.** Be sure to set bits 0, 2, 3, and 5 to 7 to “0”.

**2.** Before setting a standby mode by setting the STB bit to 1, be sure to set the PCC register to 03H and then set the STB bit to 1. Otherwise, the standby mode may not be set or released. After releasing the standby mode, change the value of the PCC register to the desired value.

**3.** To set the IDLE mode or STOP mode, set the PCC register to 03H, and the PSMR.PSM0 bit in that order and then set the STB bit to 1.

**(2) Power save mode register (PSMR)**

The PSMR register is an 8-bit register that controls the operation in the software standby 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: FFFFF820H

|      |   |   |   |   |   |   |   |      |
|------|---|---|---|---|---|---|---|------|
|      | 7 | 6 | 5 | 4 | 3 | 2 | 1 | <0>  |
| PSMR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | PSM0 |

|      |  |
|------|--|
| PSM0 | Operation in software standby mode specification |
| 0    | IDLE mode  |
| 1    | STOP mode  |

**Cautions 1. Be sure to set bits 1 to 7 to “0”.**

**2. The PSM0 bit is valid only when the PSC.STB bit is 1.**

## 21.3 HALT Mode

### 21.3.1 Setting and operation status

The HALT mode is set when a dedicated instruction (HALT) is executed in the normal operation mode.

When HALT mode is set, clock supply is stopped to the CPU only. The clock generator and PLL continue operating. 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 21-3 shows the operation status in the HALT mode.

The average power consumption of the system can be reduced by using the HALT mode in combination with the normal operation mode for intermittent operation.

**Cautions 1. Insert five or more NOP instructions after the HALT instruction.**

**2. If the HALT instruction is executed while an interrupt request is being held pending, the HALT mode is set but is released immediately by the pending interrupt request.**

### 21.3.2 Releasing HALT mode

The HALT mode is released by a non-maskable interrupt request signal (INTWDT), an unmasked maskable interrupt request signal, and a reset signal ( $\overline{\text{RESET}}$  pin input, reset signal (WDTRES) generation by watchdog timer overflow, reset signal (LVIREs) generation by low-voltage detector (LVI), or reset signal (POCREs) generation by power-on-clear circuit (POC)).

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 (INTWDT) or an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request. If the HALT mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is serviced as follows.

- (a) If an interrupt request signal with a priority lower than or same as the interrupt currently being serviced is generated, the HALT mode is released, but the newly generated interrupt request signal is not acknowledged. The interrupt request signal itself is retained. Therefore, execution starts at the next instruction after the HALT instruction.
- (b) If an interrupt request signal with a priority higher than that of the interrupt currently being serviced is issued (including a non-maskable interrupt request signal), the HALT mode is released and that interrupt request signal is acknowledged. Therefore, execution branches to the handler address.

**Table 21-2. 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                                     |                                  |
| Unmasked 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 pin input or by WDTRES, LVIRES, or POCRES signal generation

The same operation as the normal reset operation is performed.

**Table 21-3. Operation Status in HALT Mode**

| Setting of HALT Mode                   |                  | Operation Status  |
|--|------------------|---|
| Item                                   |                  |   |
| Clock generator, PLL                   |                  | Operates  |
| System clock (f <sub>xx</sub> )        |                  | Supply  |
| CPU                                    |                  | Stops operation   |
| External bus interface <sup>Note</sup> |                  | See <b>Table 2-2 Pin Operation Status in Operation Modes.</b>   |
| DMA                                    |                  | Operable  |
| Interrupt controller                   |                  | Operable  |
| Timer                                  | TAA0 to TAA4     | Operable  |
|  | TAB0, TAB1       | Operable  |
|  | TMT0, TMT1       | Operable  |
|  | TMM0 to TMM3     | Operable  |
| Watchdog timer                         |                  | Operable  |
| Serial interface                       | CSIB0 to CSIB2   | Operable  |
|  | UARTA0 to UARTA2 | Operable  |
|  | UARTB            | Operable  |
|  | I <sup>2</sup> C | Operable  |
| A/D converters 0 to 2                  |                  | Operable  |
| Clock monitor                          |                  | Operable  |
| Low-voltage detector                   |                  | Operable  |
| Power-on-clear circuit                 |                  | 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. |

**Note**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only



## 21.4 IDLE Mode

### 21.4.1 Setting and operation status

The IDLE mode is set by clearing (0) the PSMR.PSM0 bit and setting (1) the PSC.STB bit in the normal operation mode.

In the IDLE mode, the clock generator and PLL continue operation 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 IDLE 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 an external clock continue operating.

Table 21-5 shows the operation status in the IDLE mode.

The IDLE mode can reduce the power consumption more than the HALT mode because it stops the operation of the on-chip peripheral functions. The clock generator and PLL do not stop, so the normal operation mode can be restored without waiting for the oscillation stabilization time after the IDLE mode has been released, in the same manner as when the HALT mode is released.

**Caution** Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the IDLE mode.

### 21.4.2 Releasing IDLE mode

The IDLE mode is released by an unmasked external interrupt request signal (INTP00, INTP01, INTP02 to INTP07 (V850E/IG3 only), INTP08 to INTP13, INTP17, INTP18, INTADT0, or INTADT1 pin input), an unmasked internal interrupt request signal (CSIB-related interrupt request signal in the slave mode) from the peripheral functions operable in the IDLE mode, or a reset signal (RESET pin input, reset signal (LVIRE) generation by low-voltage detector (LVI), or reset signal (POCRES) generation by power-on-clear circuit (POC)).

After the IDLE mode has been released, the normal operation mode is restored.

#### (1) Releasing IDLE mode by unmasked maskable interrupt request signal

The IDLE mode is released by an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request. If the IDLE mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is processed as follows.

**Caution** When PSC.INTM bit = 1, the IDLE mode cannot be released by the unmasked maskable interrupt request signal.

- (a) If an interrupt request with a priority lower than or same as the interrupt request signal currently being serviced is generated, the IDLE mode is released, but the newly generated interrupt is not acknowledged. The interrupt request signal itself is retained. Therefore, execution starts at the next instruction after the IDLE instruction.
- (b) If an interrupt request signal with a priority higher than that of the interrupt request signal currently being serviced is issued (including a non-maskable interrupt request signal), the IDLE mode is released and that interrupt request signal is acknowledged. Therefore, execution branches to the handler address.

**Table 21-4. Operation After Releasing IDLE Mode by Interrupt Request Signal**

| Release Source                      | Interrupt Enabled (EI) Status   | Interrupt Disabled (DI) Status   |
|-------------------------------------|---|----------------------------------|
| Unmasked maskable interrupt request | Execution branches to the handler address or the next instruction is executed | The next instruction is executed |

**(2) Releasing IDLE mode by  $\overline{\text{RESET}}$  pin input or by LVIRES or POCRES signal generation**

The same operation as the normal reset operation is performed.

**Table 21-5. Operation Status in IDLE Mode**

| Setting of IDLE Mode                   |                  | Operation Status  |
|--|------------------|---|
| Item                                   |                  |   |
| Clock generator, PLL                   |                  | Operates  |
| System clock (f <sub>xx</sub> )        |                  | Stops supply  |
| CPU                                    |                  | Stops operation   |
| External bus interface <sup>Note</sup> |                  | See <b>Table 2-2 Pin Operation Status in Operation Modes.</b>   |
| DMA                                    |                  | Stops operation   |
| Interrupt controller                   |                  | Stops operation   |
| Timer                                  | TAA0 to TAA4     | Stops operation   |
|  | TAB0, TAB1       | Stops operation   |
|  | TMT0, TMT1       | Stops operation   |
|  | TMM0 to TMM3     | Stops operation   |
| Watchdog timer                         |                  | Stops operation   |
| Serial interface                       | CSIB0 to CSIB2   | Operable when $\overline{\text{SCKBn}}$ input clock is selected as count clock (in slave mode) (n = 0 to 2)   |
|  | UARTA0 to UARTA2 | Stops operation   |
|  | UARTB            | Stops operation   |
|  | I <sup>2</sup> C | Stops operation   |
| A/D converters 0 to 2                  |                  | Stops operation   |
| Clock monitor                          |                  | Operable  |
| Low-voltage detector                   |                  | Operable  |
| Power-on-clear circuit                 |                  | Operable  |
| Port function                          |                  | Retains status before 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 IDLE mode was set. |

**Note**  $\mu\text{PD70F3454GC-8EA-A}$  and  $70\text{F3454F1-DA9-A}$  only

## 21.5 STOP Mode

### 21.5.1 Setting and operation status

The STOP mode is set by setting (1) the PSMR.PSM0 bit and setting (1) the PSC.STB bit in the normal operation mode.

In the STOP mode, the clock generator stops operation. Clock supply to the CPU and the on-chip peripheral functions is stopped.

As a result, program execution is stopped, and the contents of the internal RAM before the STOP mode was set are retained. The CPU and other on-chip peripheral functions stop operating. However, the on-chip peripheral functions that can operate with an external clock continue operating.

Table 21-7 shows the operation status in the STOP mode.

Because the STOP stops operation of the clock generator, it reduces the power consumption to a level lower than the IDLE mode. When the external clock is not used, the power consumption can be minimized with only leakage current flowing.

**Caution** Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the STOP mode.

### 21.5.2 Releasing STOP mode

The STOP mode is released by an unmasked external interrupt request signal (INTP00, INTP01, INTP02 to INTP07 (V850E/IG3 only), INTP08 to INTP13, INTP17, INTP18, INTADT0, or INTADT1 pin input), an unmasked internal interrupt request signal (CSIB-related interrupt signal in the slave mode) from the peripheral functions operable in the STOP mode, or a reset signal ( $\overline{\text{RESET}}$  pin input, reset signal (LVIREN) generation by low-voltage detector (LVI), or reset signal (POCREN) generation by power-on-clear circuit (POC)).

After the STOP mode has been released, the normal operation mode is restored after the oscillation stabilization time has been secured.

#### (1) Releasing STOP mode by unmasked maskable interrupt request signal

The STOP mode is released by an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request. If the STOP mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is serviced as follows.

**Caution** When PSC.INTM bit = 1, the STOP mode cannot be released by the unmasked maskable interrupt request signal.

- (a) If an interrupt request with a priority lower than or same as the interrupt request currently being serviced is generated, the STOP mode is released, but the newly generated interrupt is not acknowledged. The interrupt request itself is retained. Therefore, execution starts at the next instruction after the STOP instruction.
- (b) If an interrupt request with a priority higher than that of the interrupt request currently being serviced is issued, the STOP mode is released and that interrupt request is acknowledged. Therefore, execution branches to the handler address.

**Table 21-6. Operation After Releasing STOP Mode by Interrupt Request Signal**

| Release Source                      | Interrupt Enabled (EI) Status   | Interrupt Disabled (DI) Status   |
|-------------------------------------|---|--|
| Unmasked maskable interrupt request | Execution branches to the handler address or the next instruction is executed after securing oscillation stabilization time | The next instruction is executed after securing oscillation stabilization time |

**(2) Releasing STOP mode by  $\overline{\text{RESET}}$  pin input or by LVIRES or POCRES signal generation**

The same operation as the normal reset operation is performed.

**Table 21-7. Operation Status in STOP Mode**

| Setting of STOP Mode                   |                  | Operation Status  |
|--|------------------|---|
| Item                                   |                  |   |
| Clock generator, PLL                   |                  | Stops operation   |
| System clock (f <sub>xx</sub> )        |                  | Stops supply  |
| CPU                                    |                  | Stops operation   |
| External bus interface <sup>Note</sup> |                  | See <b>Table 2-2 Pin Operation Status in Operation Modes.</b>   |
| DMA                                    |                  | Stops operation   |
| Interrupt controller                   |                  | Stops operation   |
| Timer                                  | TAA0 to TAA4     | Stops operation   |
|  | TAB0, TAB1       | Stops operation   |
|  | TMT0, TMT1       | Stops operation   |
|  | TMM0 to TMM3     | Stops operation   |
| Watchdog timer                         |                  | Stops operation   |
| Serial interface                       | CSIB0 to CSIB2   | Operable when $\overline{\text{SCKBn}}$ input clock is selected as count clock (in slave mode) (n = 0 to 2)   |
|  | UARTA0 to UARTA2 | Stops operation   |
|  | UARTB            | Stops operation   |
|  | I <sup>2</sup> C | Stops operation   |
| A/D converters 0 to 2                  |                  | Stops operation   |
| Clock monitor                          |                  | Stops operation   |
| Low-voltage detector                   |                  | Operable  |
| Power-on-clear circuit                 |                  | 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. |

**Note**  $\mu\text{PD70F3454GC-8EA-A}$  and  $70\text{F3454F1-DA9-A}$  only

### 21.6 Securing Oscillation Stabilization Time

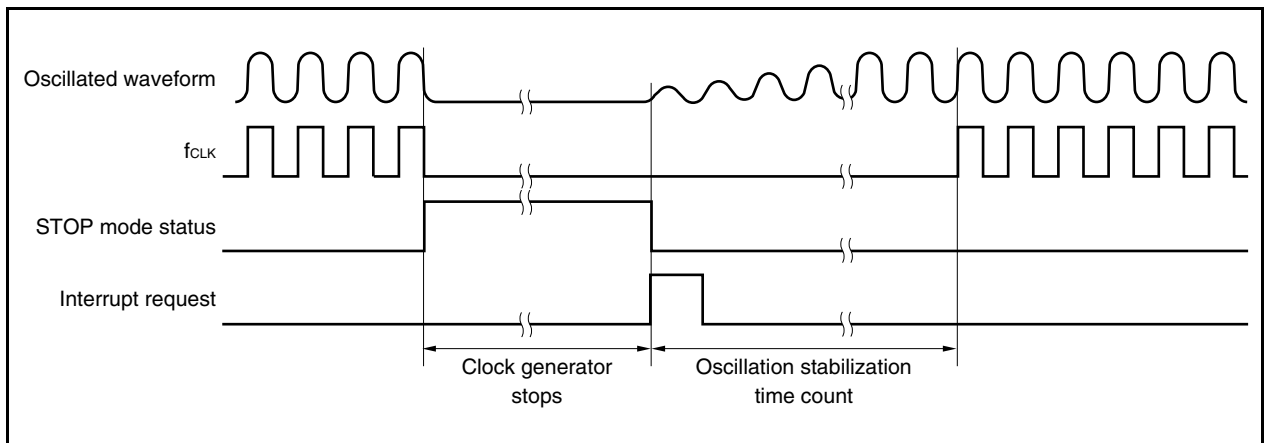
When the STOP mode is released, the oscillation stabilization time set by the OSTS register elapses. The oscillation stabilization time is the reset value of the OSTS register,  $2^{14}/f_x$  (2.048 ms at  $f_x = 8$  MHz), if the STOP mode is released by  $\overline{\text{RESET}}$  pin input.

However, the actual oscillation stabilization time is half this value (after reset:  $2^{13}/f_x$  (1.024 ms at  $f_x = 8$  MHz), and the other half is the stabilization time of the PLL. Set an oscillation stabilization time double that of the oscillation stabilization time of the oscillator used when the STOP mode is released. In addition, when releasing the mode by  $\overline{\text{RESET}}$  pin input, be sure to secure the oscillation stabilization time by outputting the  $\overline{\text{RESET}}$  signal at low level for the time longer than the oscillation stabilization time of the used resonator minus the fixed oscillation stabilization time.

The timer for counting the oscillation stabilization time secures oscillation stabilization time equal to the overflow time of the watchdog timer.

The operation performed when the STOP mode is released by an interrupt request signal is shown below.

**Figure 21-2. Oscillation Stabilization Time**



**Caution** For details of the OSTS register, see 5.3 (5) Oscillation stabilization time select register (OSTS).

## CHAPTER 22 RESET FUNCTIONS

### 22.1 Overview

- System reset by  $\overline{\text{RESET}}$  pin input
- System reset signal (WDTRES) generation by watchdog timer (WDT) overflow
- System reset signal (LVIRESET) generation by low-voltage detector (LVI)
- System reset signal (POCRES) generation by power-on-clear circuit (POC)
- Forced reset by on-chip debug function (DCU) and reset mask function (see **CHAPTER 26 ON-CHIP DEBUG FUNCTION**.)

### 22.2 Control Register

#### (1) Reset source flag register (RESF)

The RESF register is an 8-bit register that indicates occurrence of a reset request from the watchdog timer (WDT) or low-voltage detector (LVI).

<R> The WDTRF or LVIRF bit of this register is set to 1 when the internal reset source signal from WDT or LVI is asserted. The WDTRF or LVIRF bit is cleared by a reset signal (the one generated by inputting the  $\overline{\text{RESET}}$  pin, the POCRES signal generated by the power-on-clear circuit (POC), or the forced reset signal generated by the on-chip debug function), a bit manipulation instruction, or a store instruction (writing 0 to the WDTRF or LVIRF bit).

The RESF register is a special register and can be written only in a combination of specific sequences (see **3.4.8 Special registers**).

This register can be read or written in 8-bit or 1-bit units. However, bits 0 and 4 can only be cleared (0) by writing.

<R> This register is set to 00H by  $\overline{\text{RESET}}$  pin input, a reset by the power-on-clear circuit (POC), or a forced reset by the on-chip debug function. The reset value differs if the source of reset is other than those sources. For details on reset conflict, see **Cautions** on the next page.

After reset: 00H<sup>Note</sup> R/W Address: FFFF888H

|      |   |   |   |       |   |   |   |       |
|------|---|---|---|-------|---|---|---|-------|
|      | 7 | 6 | 5 | 4     | 3 | 2 | 1 | 0     |
| RESF | 0 | 0 | 0 | WDTRF | 0 | 0 | 0 | LIVRF |

|       |  |
|-------|--|
| WDTRF | Occurrence of reset signal from watchdog timer (WDT) |
| 0     | Read: No reset request, Write: Clear                 |
| 1     | Reset request  |

|       |  |
|-------|--|
| LIVRF | Occurrence of reset signal from low-voltage detector (LVI) |
| 0     | Read: No reset request, Write: Clear                       |
| 1     | Reset request  |

<R> **Note** After a reset by  $\overline{\text{RESET}}$  pin input or the power-on-clear circuit (POC), or after a forced reset by the on-chip debug function: 00H  
 After a reset due to a watchdog timer overflow: 10H  
 After a reset by the low-voltage detector (LVI): 01H

**Cautions 1. If setting (occurrence of reset of set source) and clearing (occurrence of system reset or writing 0 to the WDTRF or LVIRF bit) of the RESF register conflict, the priorities are as follows.**

- <R> <1> A reset by  $\overline{\text{RESET}}$  pin input or the power-on-clear circuit (POC), or a forced reset by the on-chip debug function (that clears the RESF register)
- <2> A reset by WDT or LVI (that sets the RESF register)
- <3> Writing 0 to the WDTRF or LVIRF bit by a bit manipulation or store instruction (that clears the RESF register)

<R> 2. Even if reset masking was specified when a flag setting source occurred, the flag is set. (Reset masking does not affect setting the flag.)

## 22.3 Operation

### (1) Reset operation by $\overline{\text{RESET}}$ pin input

When a low level is input to the  $\overline{\text{RESET}}$  pin, the V850E/IF3 and V850E/IG3 are reset, and each hardware unit is initialized to a specific status.

The oscillator continues oscillation even while a low level is input to the  $\overline{\text{RESET}}$  pin but the oscillation mode is initialized to the clock-through mode (PLLCTL register = 01H) and the CPU clock ( $f_{\text{CPU}}$ ) division to  $f_{\text{xx}}/8$  (PCC register = 03H).

The reset status is released when the  $\overline{\text{RESET}}$  pin input goes from low to high. After the reset status is released, the oscillation stabilization time of the oscillator and lockup time of PLL (default value of OSTS register for the total time:  $2^{14}/f_x$  (2.05 ms ( $f_x = 8$  MHz)) elapse, and then the CPU starts program execution. After release of reset, therefore, the operation is started in the clock-through mode and at  $f_{\text{xx}}/8$ .

The status of each hardware unit during the reset period and after the reset status is released is shown below.

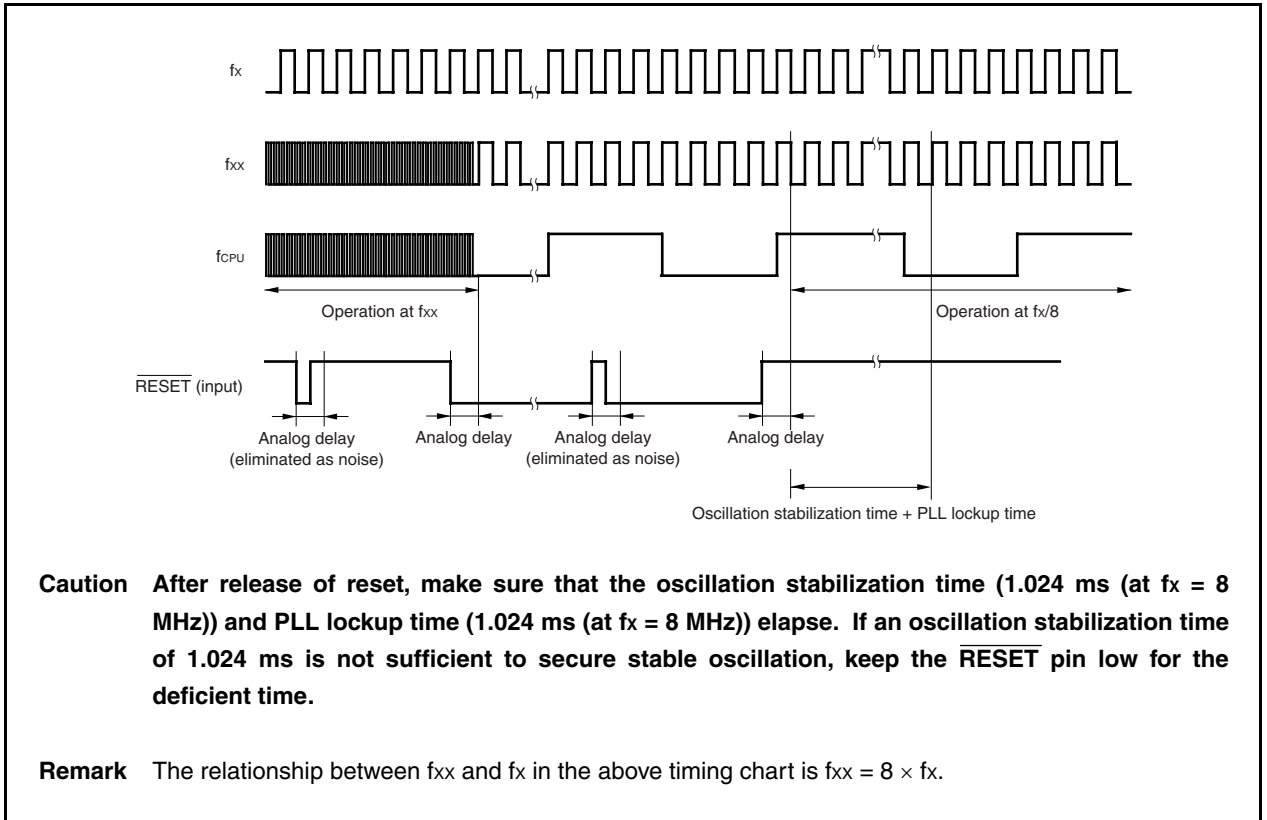
| Hardware  | During Reset Period   | After Reset Is Released  |
|---|---|--|
| Clock generator:<br>Oscillator ( $f_x$ )<br>Internal system clock ( $f_{\text{CLK}}$ )<br>CPU clock ( $f_{\text{CPU}}$ )<br>External bus clock ( $f_{\text{BUS}}$ ) <sup>Note</sup> | Oscillation/supply continues<br>However, the CPU clock ( $f_{\text{CPU}}$ ) is initialized to $f_{\text{xx}}/8$ . |  |
| Clock generator:<br>Peripheral clock ( $f_{\text{xx}}$ to $f_{\text{xx}}/4096$ )  | Oscillation/supply stops  | Oscillation/supply starts after securing of oscillation stabilization time |
| Clock generator:<br>Watchdog timer clock ( $f_{\text{xx}}/1024$ )   | Oscillation/supply stops  | Oscillation/supply starts  |
| CPU   | Initialized   | Program execution starts after securing of oscillation stabilization time  |
| Internal RAM  | Retains value immediately before reset input only in the STOP mode during reset input. Otherwise, undefined.      |  |
| Ports (including alternate-function pins)   | High impedance  |  |
| On-chip peripheral I/O registers (other than ports)   | Initialized to specific status  |  |
| On-chip peripheral functions other than above   | Stops operation   | Can start operation  |

**Note**  $\mu\text{PD70F3454GC-8EA-A}$  and  $70\text{F3454F1-DA9-A}$  only



The reset operation by  $\overline{\text{RESET}}$  pin input is illustrated below.

Figure 22-1. Reset Operation by  $\overline{\text{RESET}}$  Pin Input



The operation after release of reset is the same in both the PLL mode and clock-through mode and is started in the clock-through mode. Set the PLL mode by software control (setting PLLCTL.SELPLL bit to 1). To improve noise immunity, it is recommended to set the PLL mode and then speed up the CPU clock (example: PCC register = 00H (f<sub>xx</sub> operation)).

**(2) Reset operation (WDTRES) by overflow of watchdog timer (WDT)**

If the reset mode is set to reset upon overflow of the watchdog timer (WDT) (WDTM.WDM1 and WDTM.WDM0 bits = 10 or 11), the system is reset and each hardware is initialized to a specific state when WDT overflows (WDTRES).

If the WDTRES signal is generated, the RESF.WDTRF bit is set to 1, indicating that internal reset has occurred.

The operations during the reset period and after release of reset, other than the operation of the RESF register, are the same as the reset operation by  $\overline{\text{RESET}}$  pin input (see **(1) Reset operation by  $\overline{\text{RESET}}$  pin input**).

**<R> (3) Reset operation (LVIRE) by low-voltage detector (LVI)**

When LVI operation is enabled, the supply voltage ( $V_{DD0}$ ,  $V_{DD1}$ ) and detection voltage ( $V_{LVI}$ ) are compared and if the supply voltage drops below the detection voltage, the system is reset (when the LVIM.LVIMD bit is set to "1") and each hardware is initialized to a specific state.

The system is reset when the supply voltage drops below the detection voltage and the reset is released when the supply voltage is equal to or exceeds the detection voltage. After a reset is released, when the oscillation stabilization time (default value of the OSTS register:  $2^{14}/f_x$ ) of the oscillator has elapsed, the CPU starts executing the program.

The status of each hardware during the reset period and after reset release is the same as the reset operation by the  $\overline{\text{RESET}}$  pin (see **(1) Reset operation by  $\overline{\text{RESET}}$  pin input**).

For details of the reset operation by low-voltage detector (LVI), see **CHAPTER 23 LOW-VOLTAGE DETECTOR**.

**<R> (4) Reset operation (POCRES) by power-on-clear circuit (POC)**

When the supply voltage ( $V_{DD0}$ ,  $V_{DD1}$ ) and detection voltage ( $V_{POC0}$ ) are compared and if the supply voltage drops below the detection voltage (including at power application), the system is reset and each hardware is initialized to a specific state.

The system is reset when the supply voltage drops below the detection voltage and the reset is released when the supply voltage is equal to or exceeds the detection voltage. After a reset is released, when the oscillation stabilization time (default value of the OSTS register:  $2^{14}/f_x$ ) of the oscillator has elapsed, the CPU starts executing the program.

The status of each hardware during the reset period and after reset release is the same as the reset operation by the  $\overline{\text{RESET}}$  pin (see **(1) Reset operation by  $\overline{\text{RESET}}$  pin input**).

For details of the reset operation by power-on-clear circuit (POC), see **CHAPTER 24 POWER-ON-CLEAR CIRCUIT**.

## CHAPTER 23 LOW-VOLTAGE DETECTOR

### 23.1 Functions

The low-voltage detector (LVI) has the following functions.

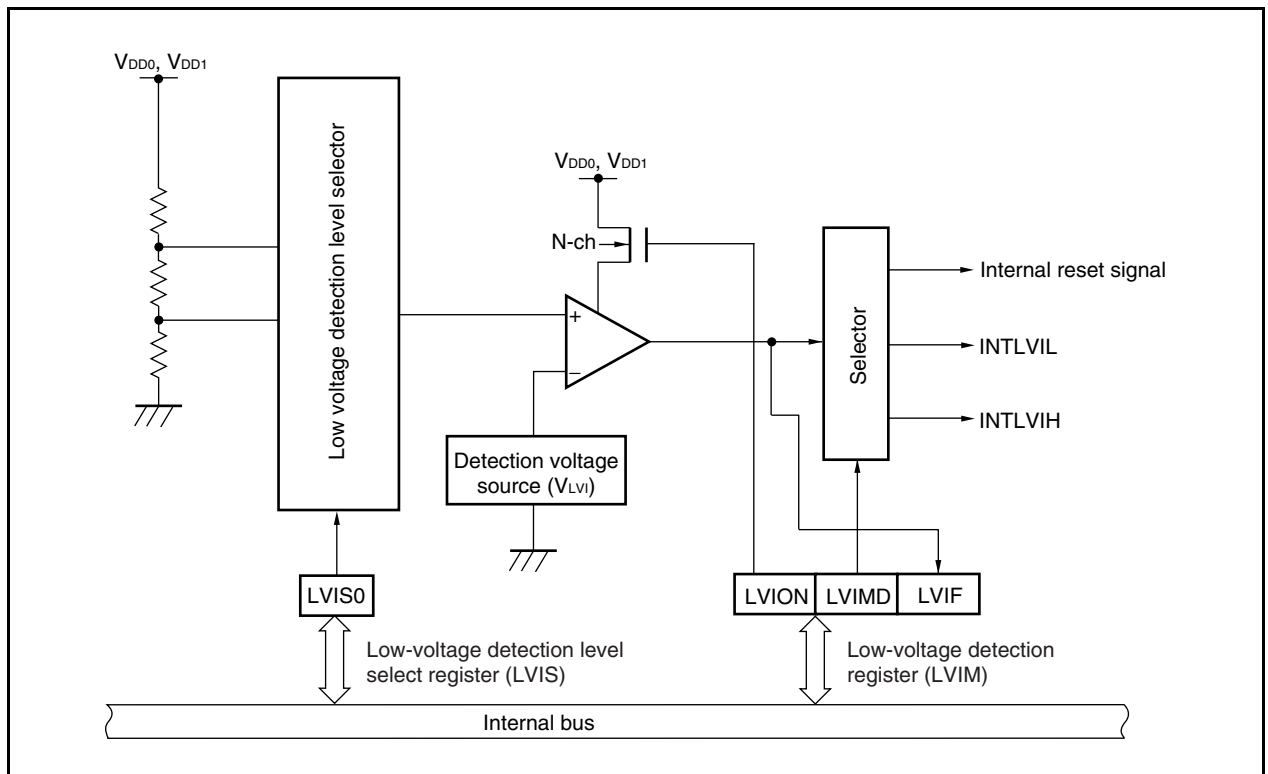
- Compares the supply voltage ( $V_{DD0}$ ,  $V_{DD1}$ ) and detection voltage ( $V_{LVI}$ ) and generates an interrupt request signal (INTLVIL, INTLVIH) or internal reset signal (LVIRES) when the supply voltage drops below the detection voltage.
- The level of the supply voltage to be detected can be changed by software (in two steps).
- An interrupt request signal (INTLVIL, INTLVIH) or internal reset signal (LVIRES) 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 RESF register, see **CHAPTER 22 RESET FUNCTIONS**.

### 23.2 Configuration

The block diagram is shown below.

Figure 23-1. Block Diagram of Low-Voltage Detector



23.3 Control 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 3.4.8 Special registers).

This register can be read or written in 8-bit or 1-bit units. However, bit 0 is read-only.

Reset other than reset by the low-voltage detector (LVI) sets this register to 00H.

After reset: 00H      R/W      Address: FFFFF890H

|      |       |   |   |   |   |   |       |      |
|------|-------|---|---|---|---|---|-------|------|
|      | <7>   | 6 | 5 | 4 | 3 | 2 | <1>   | <0>  |
| LVIM | LVION | 0 | 0 | 0 | 0 | 0 | LVIMD | LVIF |

|       |   |
|-------|---|
| LVION | 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 INTLVIL when supply voltage < detection voltage.<br>Generate interrupt request signal INTLVIH 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                                     |

<R>

- Cautions**
1. After setting the LVION bit to 1, wait for 0.1 ms or more before checking the voltage using the LVIF bit.
  2. The value of the LVIF flag is output as the output signals INTLVIL or INTLVIH when the LVION bit = 1 and LVIMD bit = 0.
  3. If the LVION bit = 1 and LVIMD bit = 1, the low-voltage detector (LVI) cannot be stopped until a reset request other than that of by the LVI is generated.
  4. Be sure to set bits 2 to 6 to "0".

**(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.

Reset other than reset by the low-voltage detector (LVI) sets this register to 00H.

|                  |     |                    |   |   |   |   |   |   |       |
|------------------|-----|--------------------|---|---|---|---|---|---|-------|
| After reset: 00H | R/W | Address: FFFFF891H |   |   |   |   |   |   |       |
|                  |     | 7                  | 6 | 5 | 4 | 3 | 2 | 1 | 0     |
| LVIS             |     | 0                  | 0 | 0 | 0 | 0 | 0 | 0 | LVIS0 |

| LVIS0 | Detection level   |
|-------|-------------------|
| 0     | 4.4 V $\pm$ 0.2 V |
| 1     | 4.2 V $\pm$ 0.2 V |

- Cautions**
- 1. The LVIS register cannot be written until a reset request due to something other than the low-voltage detector (LVI) 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”.**

## 23.4 Operation

Depending on the setting of the LVIM.LVIMD bit, an interrupt request signal (INTLVIL, INTLVIH) or an internal reset signal (LVIRESET) is generated.

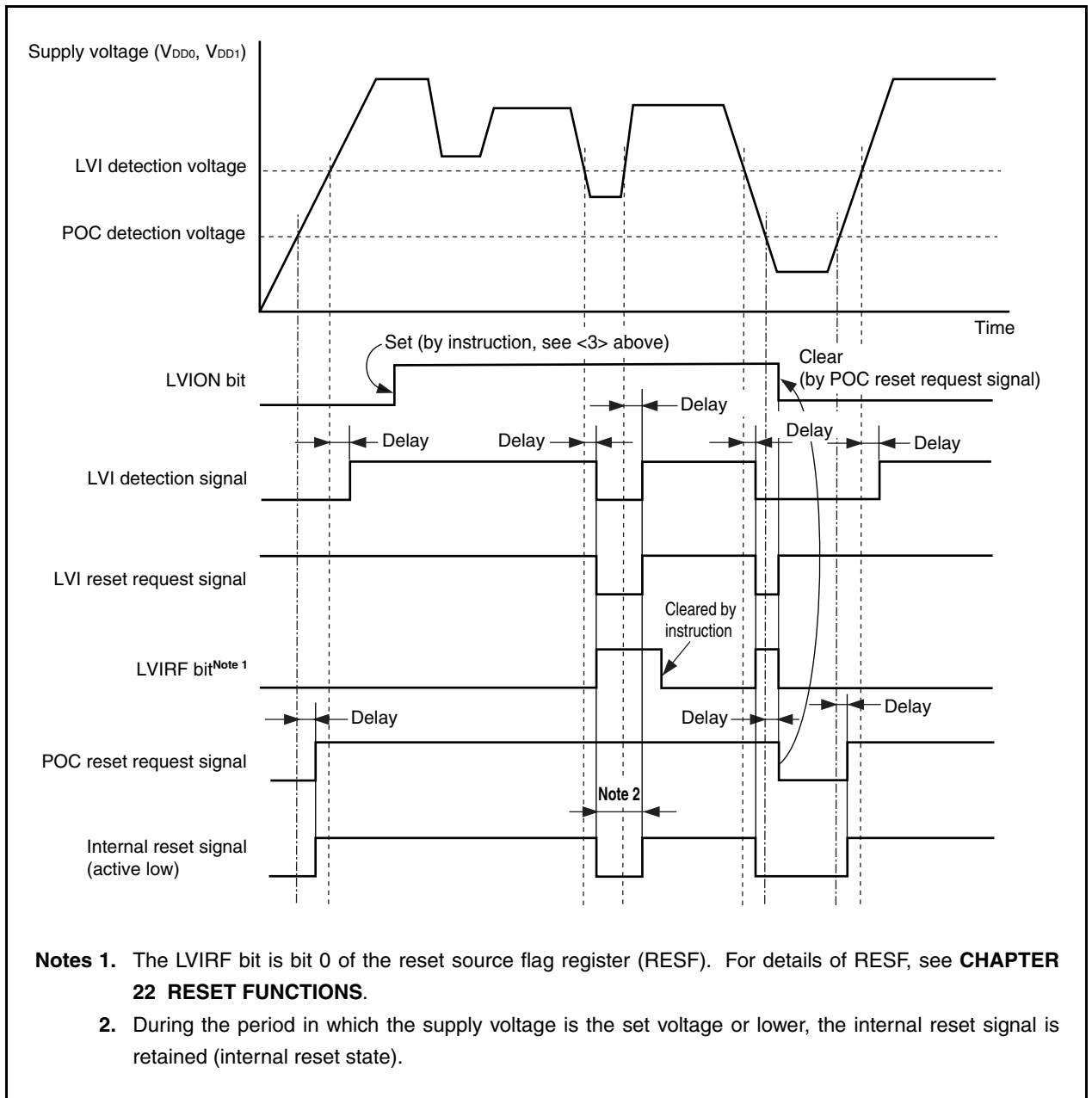
### 23.4.1 To use for internal reset signal

<To start operation>

- <1> Mask the interrupt of the low-voltage detector (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).
- <R> <4> Insert a wait cycle of 0.1 ms or more 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 the low-voltage detector (LVI) is generated.

Figure 23-2. Operation Timing of Low-Voltage Detector (LVIMD Bit = 1)



23.4.2 To use for interrupt

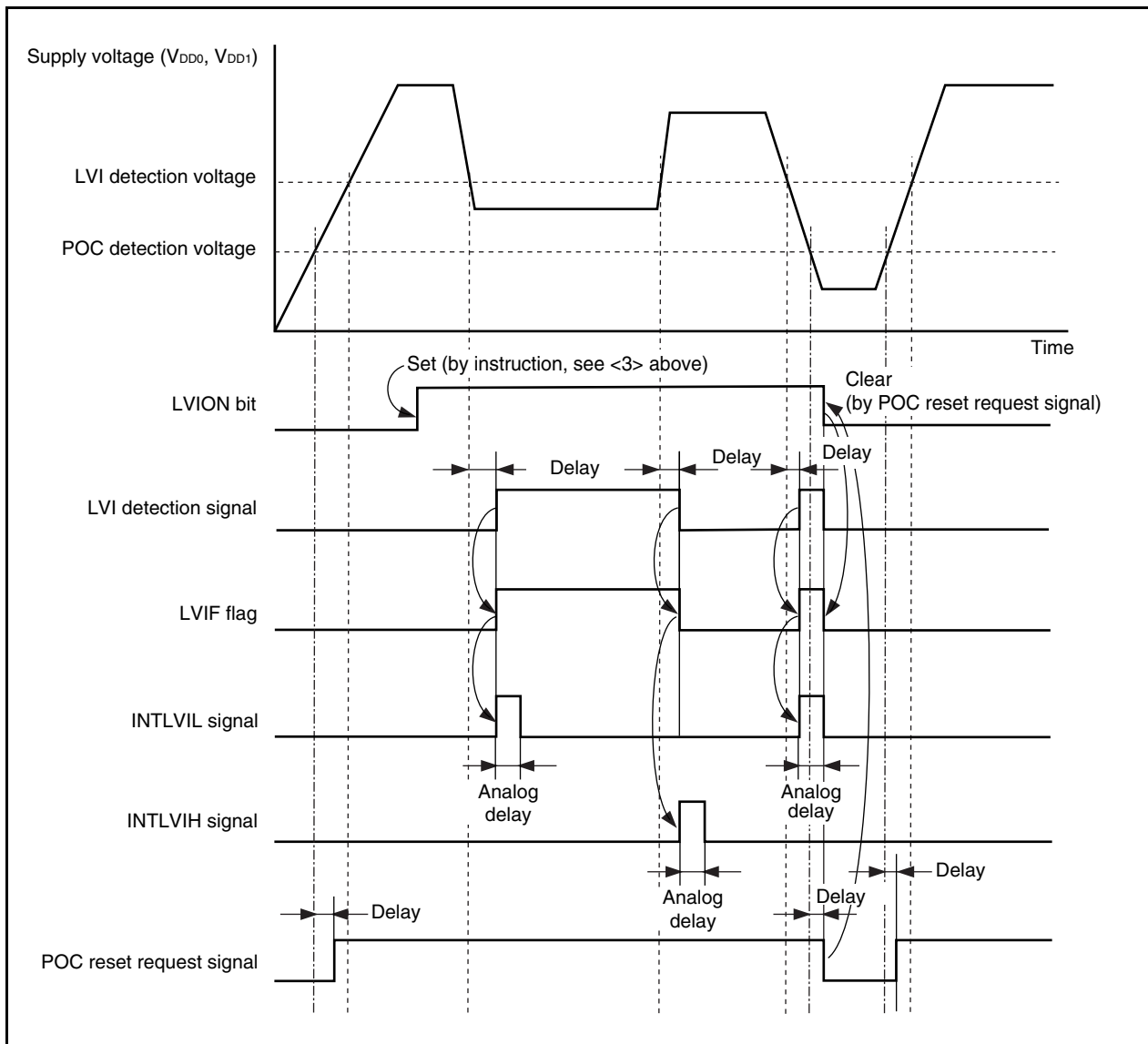
<To start operation>

- <1> Mask the interrupt of the low-voltage detector (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).
- <R> <4> Insert a wait cycle of 0.1 ms or more 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>

Set the LVION bit to 0.

Figure 23-3. Operation Timing of Low-Voltage Detector (LVIMD Bit = 0)





## CHAPTER 24 POWER-ON CLEAR CIRCUIT

### 24.1 Function

Functions of the power-on-clear circuit (POC) are shown below.

- Generates a reset signal (POCRES) upon power application.
- Compares the supply voltage ( $V_{DD0}$ ,  $V_{DD1}$ ) and detection voltage ( $V_{POC0}$ ), and generates a reset signal when the supply voltage drops below the detection voltage (detection voltage ( $V_{POC0}$ ):  $3.7\text{ V} \pm 0.2\text{ V}$ ).

**Remark** The V850E/IF3 and V850E/IG3 have the reset source flag register (RESF) that indicates generation of a reset signal (WDTRES) by watchdog timer overflow and a reset signal (LVIREs) by low-voltage detector (LVI).

The RESF register is not cleared to 00H when a reset signal (WDTRES or LVIREs) is generated, and its flag corresponding to the reset source is set to 1.

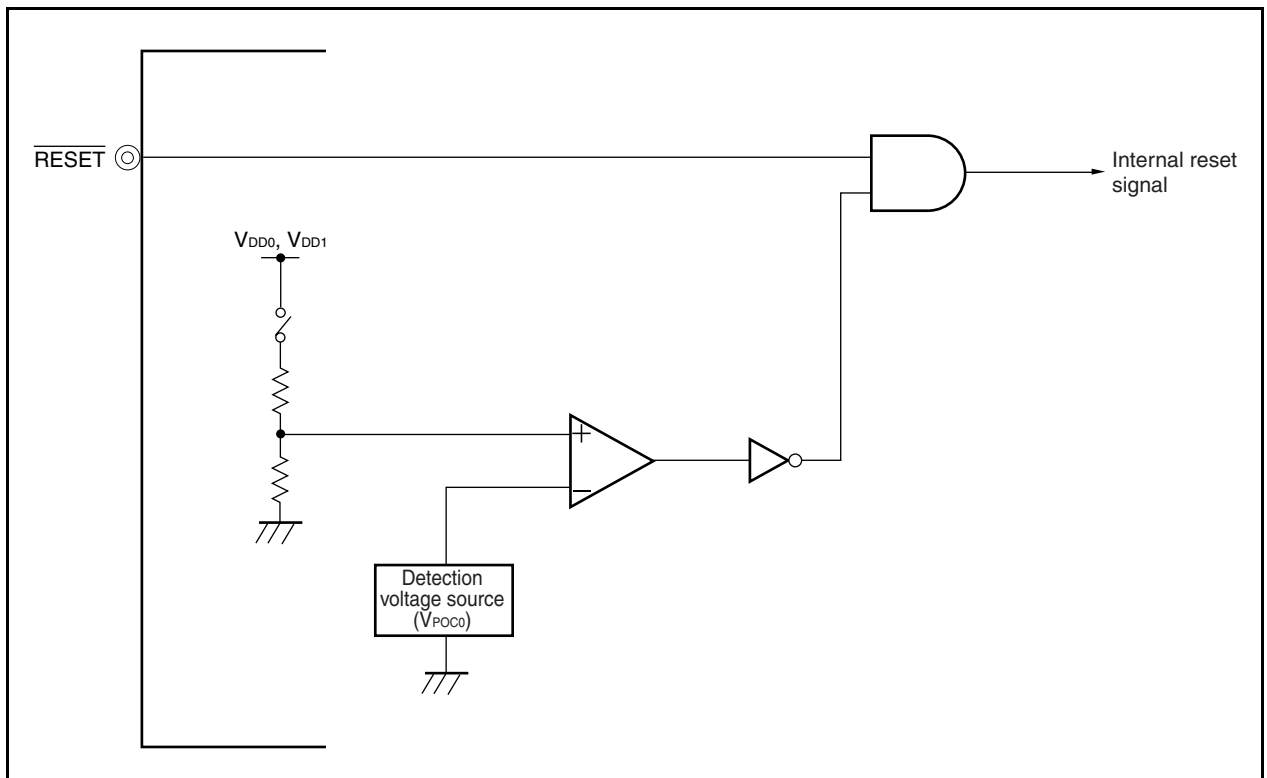
The RESF register is cleared (00H) when a reset signal (POCRES) by power-on-clear circuit (POC) is generated.

For details of the RESF register, see **CHAPTER 22 RESET FUNCTIONS**.

### 24.2 Configuration

The block diagram is shown below.

**Figure 24-1. Block Diagram of Power-on-Clear Circuit**



### 24.3 Operation

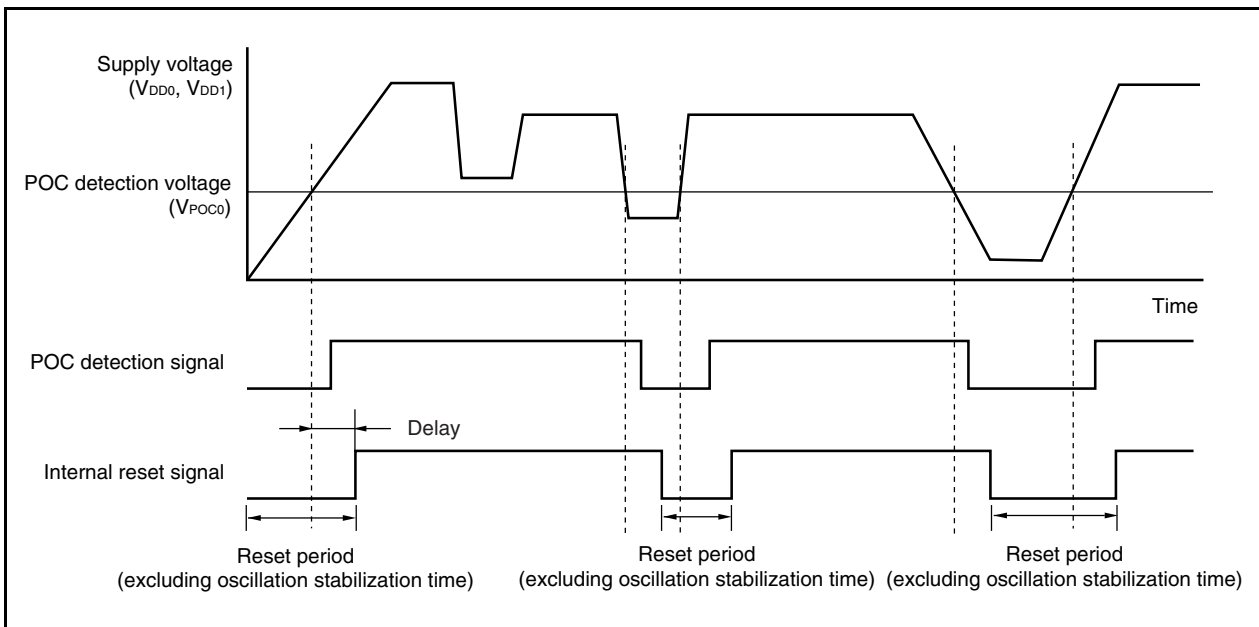
When the supply voltage and detection voltage are compared and if the supply voltage drops below the detection voltage (including at power application), the system is reset and each hardware is initialized to the specific status.

The system is reset from when low voltage is detected until the supply voltage becomes higher than the detection voltage. After a reset is released, when the oscillation stabilization time (default value of the OSTS register:  $2^{14}/f_x$ ) of the oscillator has elapsed, the CPU starts executing the program.

The status of each hardware during the reset period and after reset release is the same as the reset operation by the  $\overline{\text{RESET}}$  pin (see 22.3 (1) Reset operation by  $\overline{\text{RESET}}$  pin input).

The following shows the timing chart.

Figure 24-2. Timing of Reset Signal Generation by Power-on-Clear Circuit



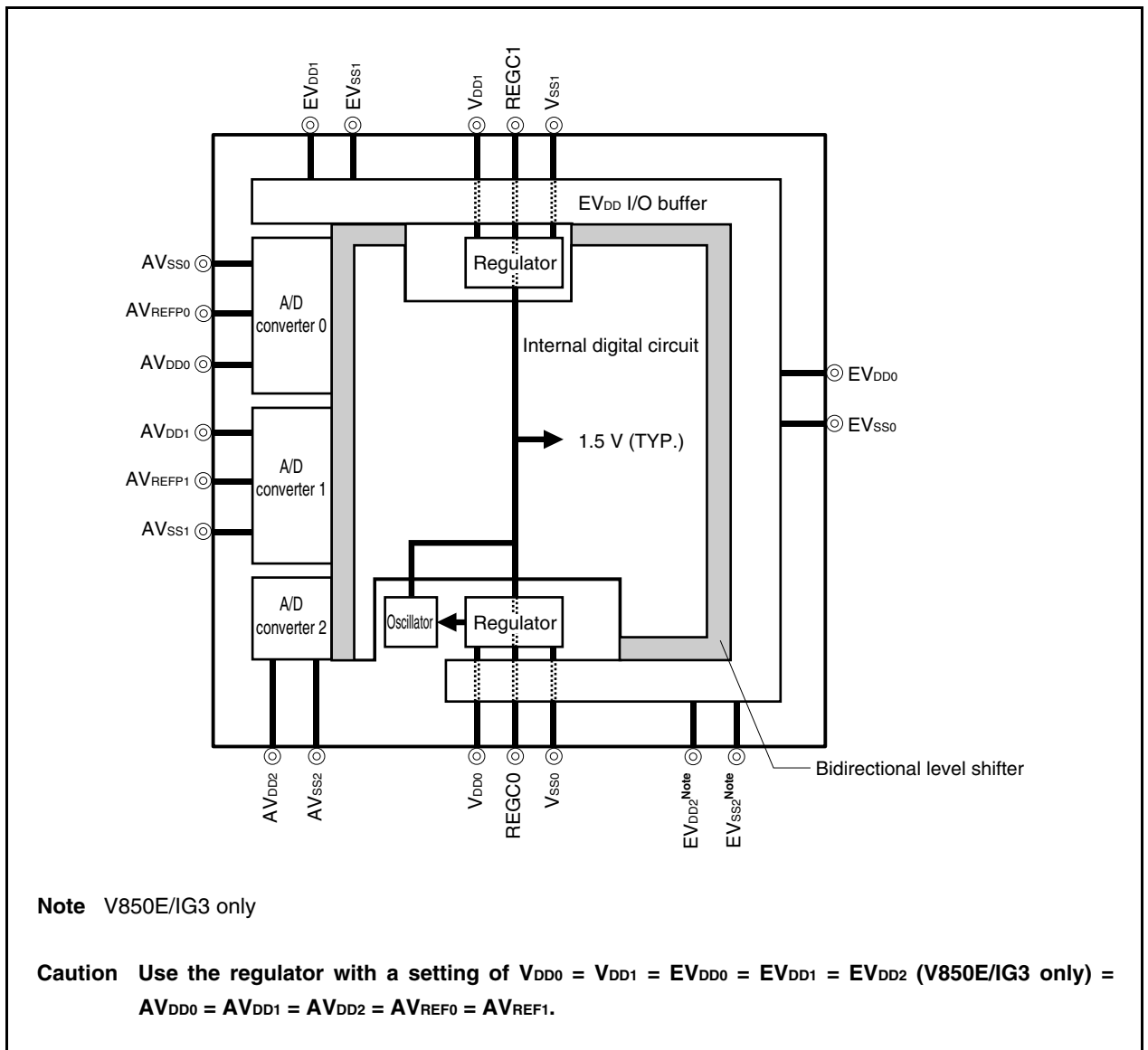
## CHAPTER 25 REGULATOR

### 25.1 Overview

The V850E/IF3 and V850E/IG3 have an internal regulator to realize a 5 V single power supply operation.

This regulator supplies a stepped-down  $V_{DD0}$  and  $V_{DD1}$  power supply voltage to the oscillation block and internal logic circuits (except the A/D converters 0 to 2 and I/O buffers). The regulator output voltage (REGC0, REGC1 pins) is set to 1.5 V (TYP.).

Figure 25-1. Regulator



## 25.2 Operation

The regulator of this product always operates in any mode (normal operation mode, HALT mode, IDLE mode, STOP mode, or during reset).

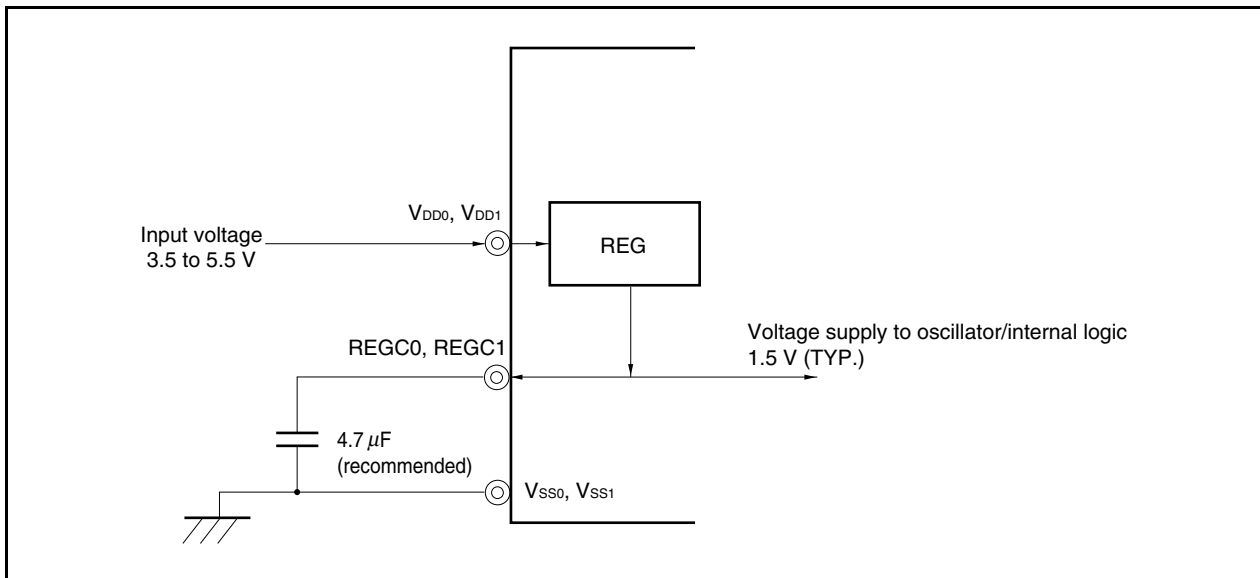
Be sure to connect a capacitor ( $4.7\ \mu\text{F}$  (recommended value))<sup>Note</sup> to the REGC0 and REGC1 pins to stabilize the regulator output.

A diagram of the regulator pin connection method is shown below.

**Note** Use the low-ESR ( $0.5\ \Omega$  or lower) of the series resistance component ESR.

**Caution** The V850E/IF3 and V850E/IG3 have two regulators each. Therefore, connect a capacitor to each of the REGC0 and REGC1 pins.

Figure 25-2. Connection of REGC0 and REGC1 Pins



## CHAPTER 26 ON-CHIP DEBUG FUNCTION

The on-chip debug function of the V850E/IF3 and V850E/IG3 can be realized in the following two ways.

- Debugging using DCU (debug control unit) (using MINICUBE)
 

By using the  $\overline{DRST}$ , DCK, DMS, DDI, and DDO pins as debug interface pins, on-chip debugging is realized by the internal DCU of the V850E/IG3<sup>Note</sup>.
- Debugging without using DCU (using MINICUBE2)
 

On-chip debugging is realized by MINICUBE2 without using the DCU but by using the user resources.

**Note** The V850E/IF3 does not have an internal 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  |
|--|-------------------|--|--|
| Target product   |                   | V850E/IG3  | V850E/IF3, V850E/IG3   |
| Debug interface pins   |                   | $\overline{DRST}$ , DCK, DMS, DDI, DDO             | <ul style="list-style-type: none"> <li>• When UARTA0 is used<br/>RXDA0, TXDA0</li> <li>• When CSIB0 is used<br/>SIB0, SOB0, SCKB0, HS (P43)</li> </ul> |
| Securing of user resources                                   |                   | Not required                                       | Required   |
| Hardware break function                                      |                   | 2 points   | 2 points (V850E/IG3 only)  |
| Software break function                                      | Internal ROM area | 4 points   | 4 points   |
|  | RAM area          | 2000 points  | 2000 points  |
| Real-time RAM monitor function <sup>Note 1</sup>             |                   | Available  | Available  |
| Dynamic memory modification (DMM) function <sup>Note 2</sup> |                   | Available  | Available  |
| Mask function  |                   | Reset, INTWDT, $\overline{WAIT}$ <sup>Note 3</sup> | $\overline{RESET}$ , $\overline{WAIT}$ <sup>Note 3</sup>   |
| ROM security function  |                   | 10-byte ID code authentication                     | 10-byte ID code authentication   |
| Hardware used  |                   | MINICUBE   | MINICUBE2  |
| 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.
  3.  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

## 26.1 Debugging Using DCU

The program can be debugged by using the debug interface pins ( $\overline{\text{DRST}}$ , DCK, DMS, DDI, and DDO) and connecting an on-chip debug simulator (MINICUBE).

**Caution** Only the V850E/IG3 has a DCU.

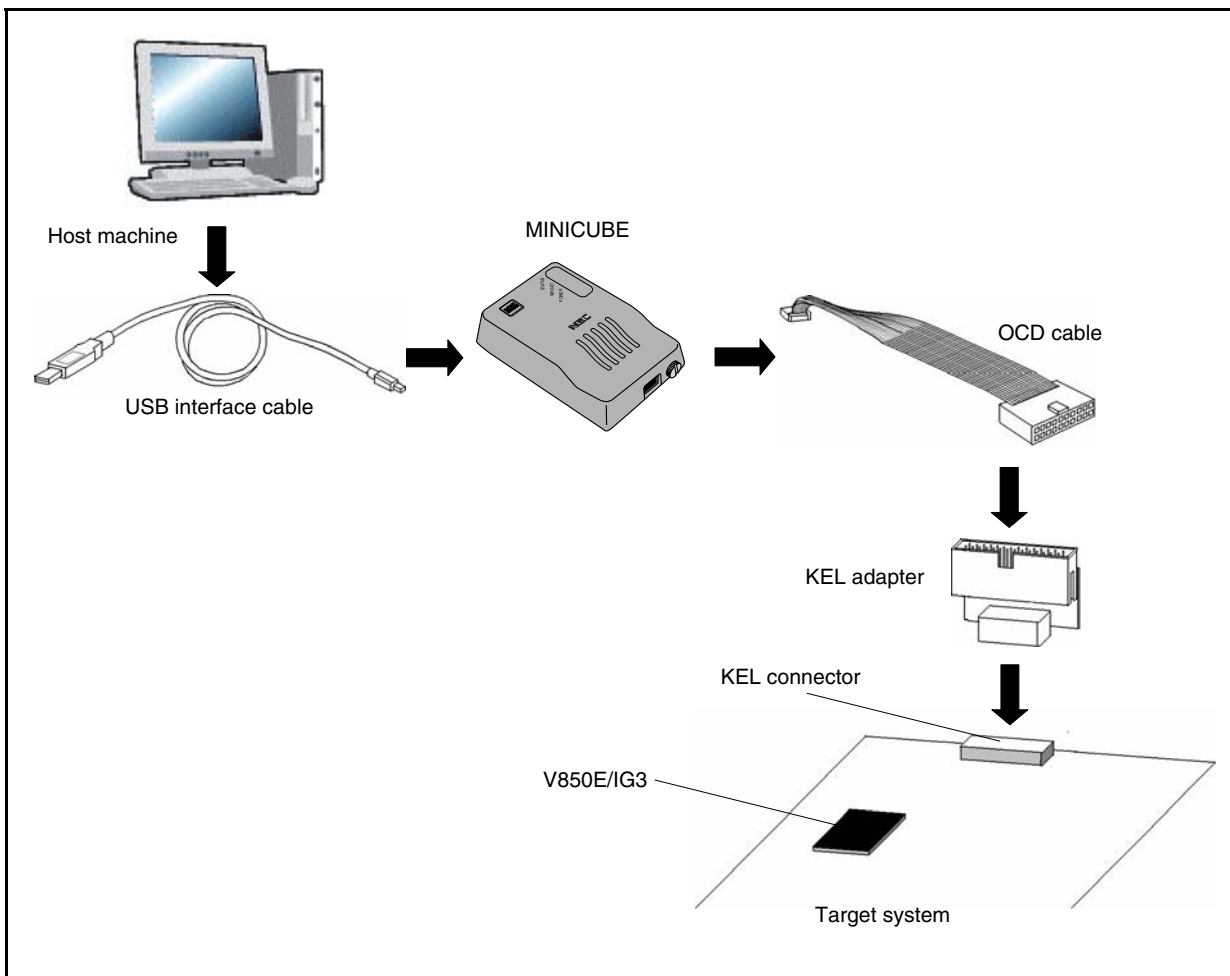
### 26.1.1 Circuit connection examples

When the MINICUBE is used, use of the following KEL connector is recommended.

- Part number
  - 8830E-026-170S: Straight type
  - 8830E-026-170L: Right-angle type

It is necessary to mount an emulator and circuit for connection on the target system.

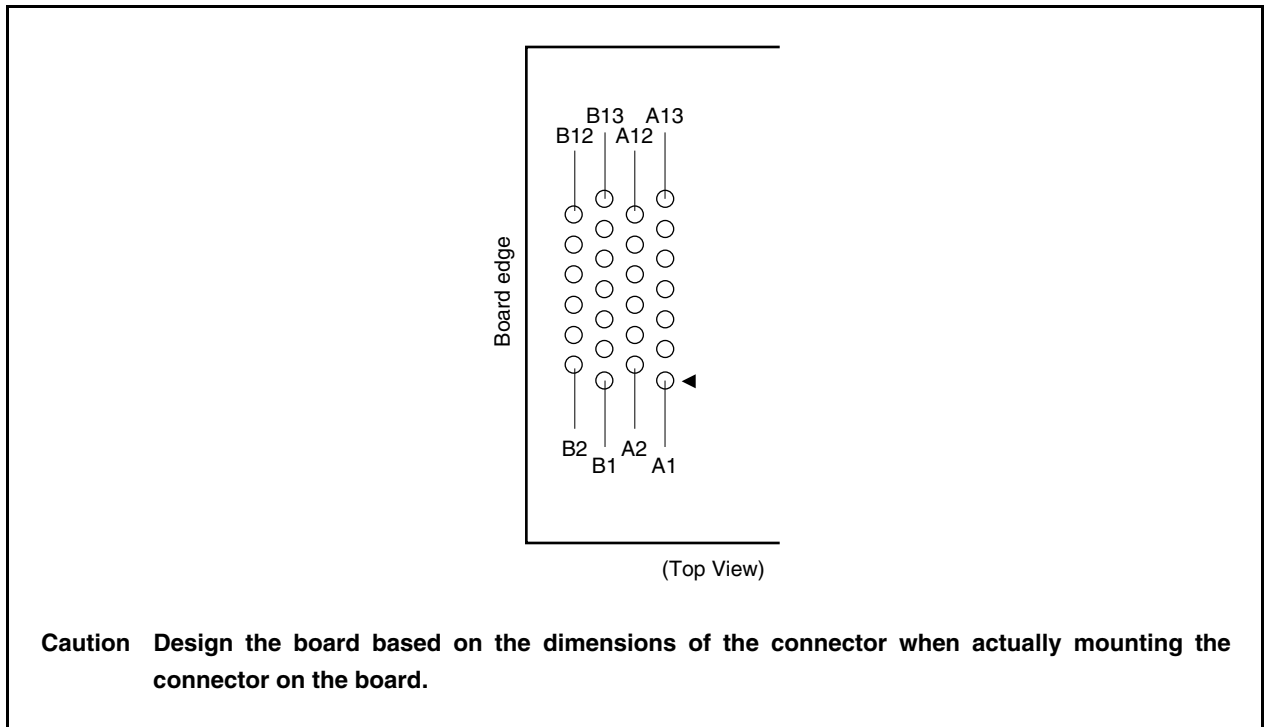
**Figure 26-1. Connection Example of On-Chip Debug Emulator (MINICUBE)**



(1) Pin configuration

The following figure shows the pin configuration of the emulator connector (on the target system side).

Figure 26-2. Pin Configuration of Emulator Connector (on Target System Side)



**(2) Pin functions**

The following table shows the pin functions of the emulator connector (on the target system side).

**Table 26-2. Pin Functions of Emulator Connector (on Target System Side)**

| Pin No. | Pin Name                 | I/O    | Pin Function   |
|---------|--------------------------|--------|--|
| A1      | (Reserved 1)             | –      | (Connect to GND)                                       |
| A2      | (Reserved 2)             | –      | (Connect to GND)                                       |
| A3      | (Reserved 3)             | –      | (Connect to GND)                                       |
| A4      | (Reserved 4)             | –      | (Connect to GND)                                       |
| A5      | (Reserved 5)             | –      | (Connect to GND)                                       |
| A6      | (Reserved 6)             | –      | (Connect to GND)                                       |
| A7      | DDI                      | Output | Data output for debug serial interface                 |
| A8      | DCK                      | Output | Clock output for debug serial interface                |
| A9      | DMS                      | Output | Transfer mode select output for debug serial interface |
| A10     | DDO                      | Input  | Data input for debug serial interface                  |
| A11     | $\overline{\text{DRST}}$ | Output | DCU reset output                                       |
| A12     | (Reserved 7)             | –      | (Leave open)   |
| A13     | FLMD0                    | Output | Control signal for flash memory downloading            |
| B1      | GND                      | –      | –  |
| B2      | GND                      | –      | –  |
| B3      | GND                      | –      | –  |
| B4      | GND                      | –      | –  |
| B5      | GND                      | –      | –  |
| B6      | GND                      | –      | –  |
| B7      | GND                      | –      | –  |
| B8      | GND                      | –      | –  |
| B9      | GND                      | –      | –  |
| B10     | GND                      | –      | –  |
| B11     | PORT0_IN                 | –      | (Connect to GND)                                       |
| B12     | PORT1_IN                 | –      | (Connect to GND)                                       |
| B13     | V <sub>DD</sub>          | –      | 5 V input (for monitoring power application to target) |

- Cautions**
1. The connection of the pins not supported in the V850E/IG3 depends on the emulator used.
  2. The pattern on the target board must satisfy the following conditions.

- Keep the pattern length to within 100 mm.
- Shield the clock signal with GND.

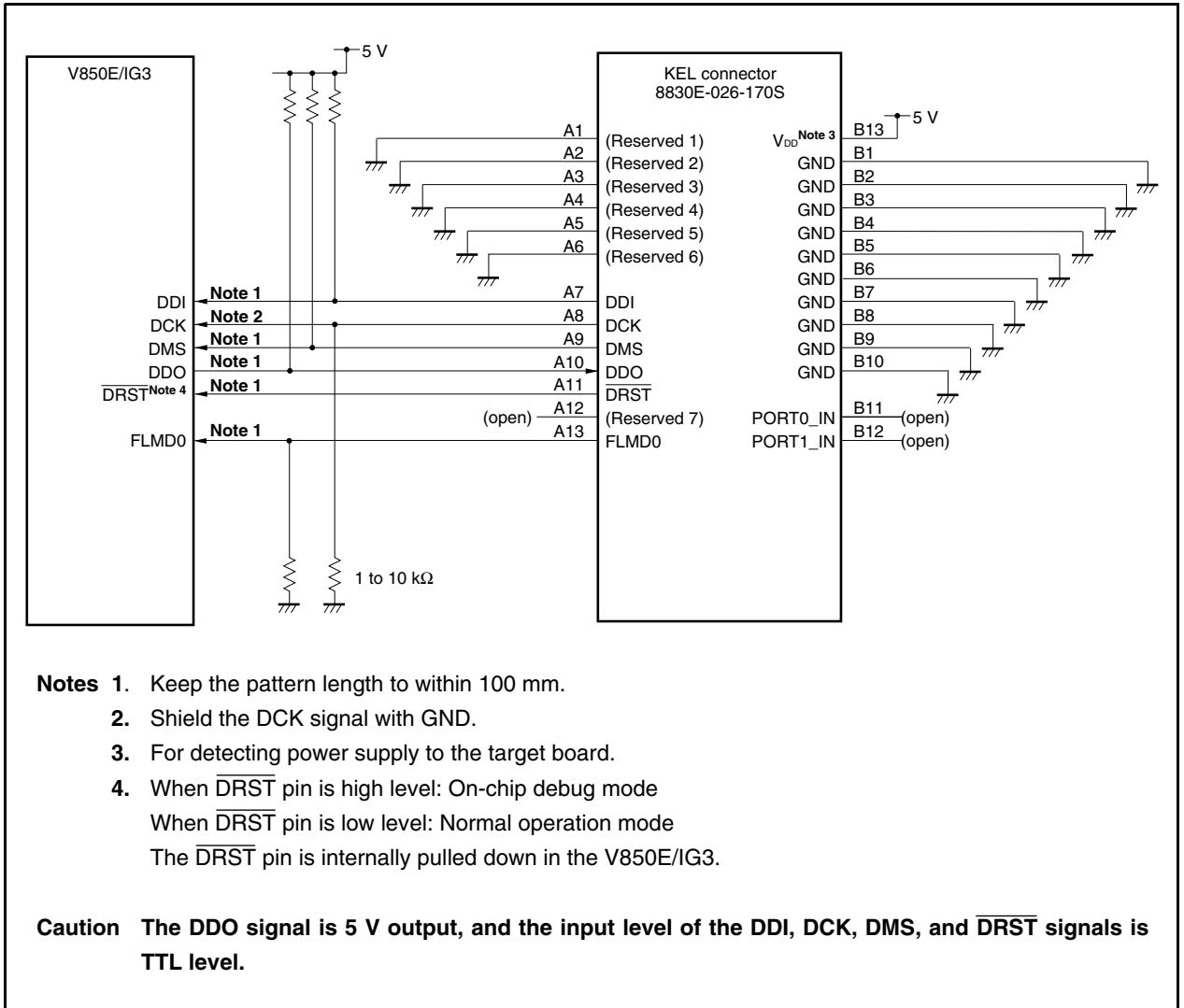
**Remark** Input/output is as viewed from the emulator side.



(3) Recommended circuit example

The following figure shows an example of the recommended circuit of the emulator connector (on the target system side).

Figure 26-3. Example of Recommended Connection of Emulator



### 26.1.2 Interface signals

The interface signals on the V850E/IG3 side 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 (DCU).

MINICUBE changes the level of the  $\overline{\text{DRST}}$  signal from low to high for output and starts the on-chip debug unit of the V850E/IG3 when it detects VDD of the target system after the integrated debugger is started. If VDD is not detected from the target system, the output signals ( $\overline{\text{DRST}}$ , DCK, DMS, DDI, and FLMD0 pins) from the MINICUBE go into a high-impedance state.

When the  $\overline{\text{DRST}}$  signal goes high, a reset signal is also generated in the V850E/IG3.

When starting debugging by starting the integrated debugger, a reset signal 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) 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 of the V850E/IG3.

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 of the V850E/IG3.

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 Integrated Debugger Operation User's Manual**.

### 26.1.3 Maskable functions

Reset, INTWDT, and  $\overline{\text{WAIT}}^{\text{Note}}$  signals can be masked.

The maskable functions with the debugger (ID850QB) and the corresponding functions are shown below.

**Note**  $\mu\text{PD70F3454GC-8EA-A}$  and  $70\text{F3454F1-DA9-A}$  only

**Table 26-3. Maskable Functions**

| Maskable Functions with Debugger (ID850QB) | Corresponding Function of V850E/IG3  |
|--|--|
| NMI0                                       | Non-maskable interrupt request signal (INTWDT) generation  |
| NMI1                                       | ×  |
| NMI2                                       | ×  |
| STOP                                       | ×  |
| HOLD                                       | ×  |
| RESET                                      | $\overline{\text{RESET}}$ pin input, reset signal (WDTRES) generation by watchdog timer overflow, reset signal (LVIRESET) generation by low-voltage detector (LVI), reset signal (POCRESET) generation by power-on-clear circuit (POC) |
| WAIT                                       | $\overline{\text{WAIT}}$ pin <sup>Note</sup> input   |

**Note**  $\mu\text{PD70F3454GC-8EA-A}$  and  $70\text{F3454F1-DA9-A}$  only

**<R> 26.1.4 Cautions**

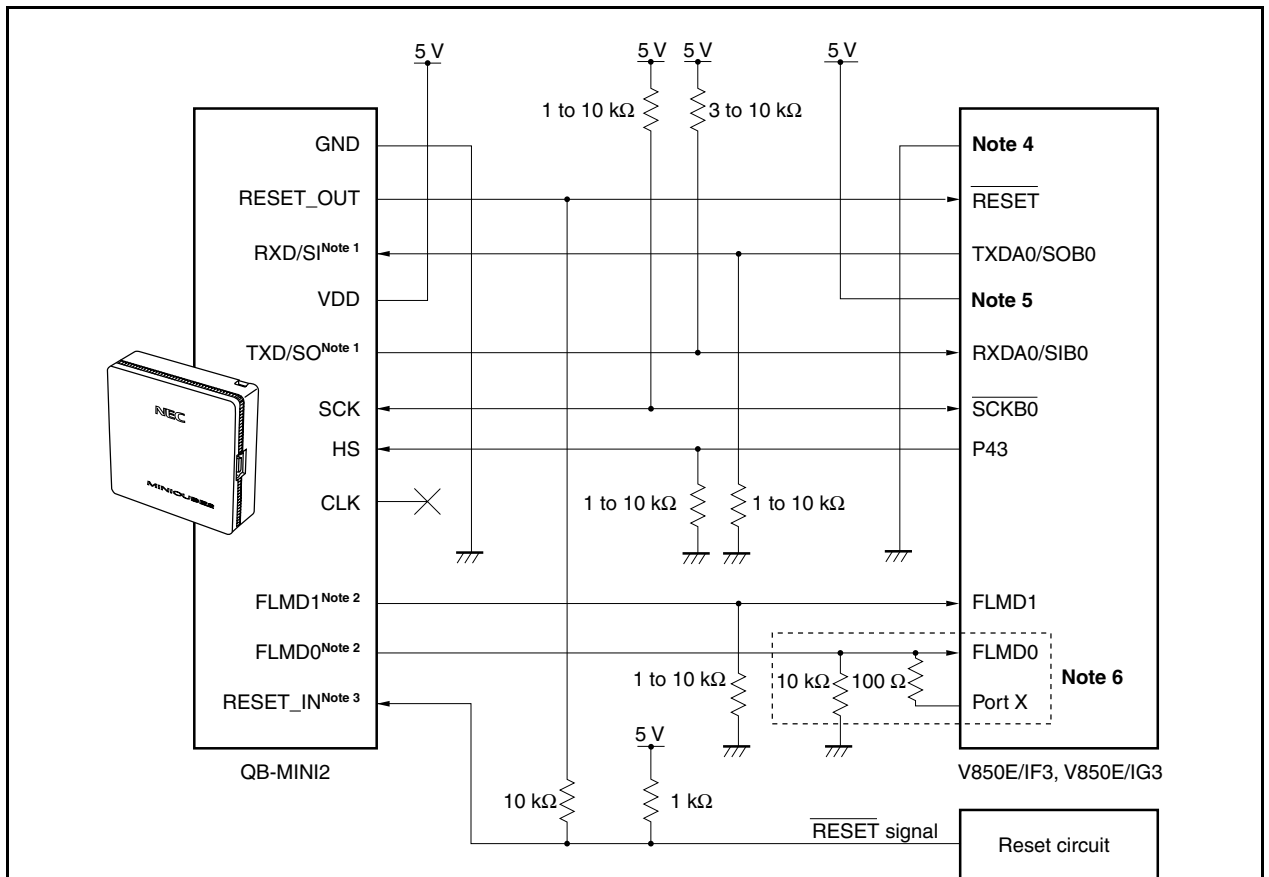
- (1) If a reset signal is input (from the target system or due to the execution of an internal reset) while the program is running, the software breaks specified for the on-chip flash memory area will no longer occur. Use hardware breaks to avoid this problem. The disabled software breaks can be enabled again by generating a forcible break or a hardware break.
- (2) Pin reset during a break is masked and the CPU and peripheral I/O are not reset. If pin reset or internal reset is generated as soon as the flash memory is rewritten by DMA or read by the RAM monitor function while the user program is being executed, the CPU and peripheral I/O may not be correctly reset.
- (3) In the on-chip debug mode, the DDO pin is forcibly set to the high-level output.
- (4) The flash memory of the device used in debugging is rewritten during debugging, so the number of flash memory rewrites cannot be guaranteed. Therefore, do not use the device used in debugging for a mass production product.
- (5) Because the DDI and DCK pins function alternately as the CSIB0 I/O pins (SOB0,  $\overline{\text{SCKB0}}$ ), UARTA0 output pin (TXDA0), and external interrupt pin (INTP13), CSIB0, UARTA0, and INTP13 cannot be used while the on-chip debug function is being used.
- (6) When the on-chip debug function is used, the clock generator and PLL continue operating even if the STOP mode is set.

## 26.2 Debugging Without Using DCU

The following describes how to implement an on-chip debug function using MINICUBE2 with the UARTA0 pins (RXDA0, TXDA0) or CSIB0 pins (SIB0, SOB0,  $\overline{SCKB0}$ , HS (P43)) as debug interfaces, without using the DCU.

### 26.2.1 Circuit connection examples

Figure 26-4. Circuit Connection Example When UARTA0/CSIB0 Is Used for Communication Interface



- Notes**
1. Connect TXDA0/SOB0 (transmit side) of the V850E/IF3 and V850E/IG3 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 V850E/IF3 and V850E/IG3.
  2. The V850E/IF3, V850E/IG3-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 a Hi-Z state.
  3. This connection is designed assuming that the  $\overline{RESET}$  signal is output from the N-ch open-drain buffer (output resistance: 100  $\Omega$  or less).
  4. EV<sub>SS0</sub>, EV<sub>SS1</sub>, EV<sub>SS2</sub> (V850E/IG3 only), V<sub>SS0</sub>, V<sub>SS1</sub>, AV<sub>SS0</sub>, AV<sub>SS1</sub>, AV<sub>SS2</sub>
  5. EV<sub>DD0</sub>, EV<sub>DD1</sub>, EV<sub>DD2</sub> (V850E/IG3 only), V<sub>DD0</sub>, V<sub>DD1</sub>, AV<sub>DD0</sub>, AV<sub>DD1</sub>, AV<sub>DD2</sub>
  6. The circuit enclosed by broken lines 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 $\Omega$ .

**Remark** See Table 26-4 for pins used when UARTA0 or CSIB0 is used for communication interface.

**Table 26-4. Wiring Between V850E/IF3, V850E/IG3, and MINICUBE2 (1/2)**

| Pin Configuration of MINICUBE2 (QB-MINI2) |        |   | When UAR2A0 Used          |            |            |            |                        |
|---|--------|---|---------------------------|------------|------------|------------|------------------------|
| Signal Name                               | I/O    | Pin Function  | Pin Name                  | Pin No.    |            |            |                        |
|   |        |   |                           | IF3        | IG3        |            |                        |
|   |        |   |                           | GC         | GC         | GF         | F1                     |
| SI/RxD                                    | Input  | Pin to receive commands and data from V850E/IF3 and V850E/IG3               | TXDA0                     | 39         | 48         | 76         | N12                    |
| SO/TxD                                    | Output | Pin to transmit commands and data to V850E/IF3 and V850E/IG3                | RXDA0                     | 38         | 47         | 75         | P12                    |
| SCK                                       | Output | Clock output pin for 3-wire serial communication                            | Not needed                | Not needed | Not needed | Not needed | Not needed             |
| CLK                                       | Output | Clock output pin to V850E/IF3 and V850E/IG3                                 | Not needed                | Not needed | Not needed | Not needed | Not needed             |
| RESET_OUT                                 | Output | Reset output pin to V850E/IF3 and V850E/IG3                                 | $\overline{\text{RESET}}$ | 35         | 40         | 68         | P9                     |
| FLMD0                                     | Output | Output pin to set V850E/IF3 and V850E/IG3 to debug mode or programming mode | FLMD0                     | 37         | 46         | 74         | N9                     |
| FLMD1                                     | Output | Output pin to set programming mode  | FLMD1                     | 60         | 76         | 4          | A12                    |
| HS  | Input  | Handshake signal for CS10 + HS communication                                | Not needed                | Not needed | Not needed | Not needed | Not needed             |
| GND                                       | -      | Ground  | V <sub>SS0</sub>          | 32         | 37         | 65         | N7, P7                 |
|   |        |   | V <sub>SS1</sub>          | 66         | 85         | 13         | A9, B9                 |
|   |        |   | AV <sub>SS0</sub>         | 6          | 7          | 35         | E1, E2                 |
|   |        |   | AV <sub>SS1</sub>         | 11         | 12         | 40         | H1, H2                 |
|   |        |   | AV <sub>SS2</sub>         | 22         | 27         | 55         | P1, N2, M3, L4         |
|   |        |   | EV <sub>SS0</sub>         | 54         | 64         | 92         | A14, G13, G14          |
|   |        |   | EV <sub>SS1</sub>         | 79         | 1          | 29         | A1, B1, C1, C2, D3, E4 |
|   |        |   | EV <sub>SS2</sub>         | Not needed | 42         | 70         | N11, P11, P14          |
| RESET_IN                                  | Input  | Reset input pin on the target system  |                           |            |            |            |                        |

Table 26-4. Wiring Between V850E/IF3, V850E/IG3, and MINICUBE2 (2/2)

| Pin Configuration of MINICUBE2 (QB-MINI2) |        |   | When CSIB0-HS Used        |            |            |            |                        |
|---|--------|---|---------------------------|------------|------------|------------|------------------------|
| Signal Name                               | I/O    | Pin Function  | Pin Name                  | Pin No.    |            |            |                        |
|   |        |   |                           | IF3        | IG3        |            |                        |
|   |        |   |                           | GC         | GC         | GF         | F1                     |
| SI/RxD                                    | Input  | Pin to receive commands and data from V850E/IF3 and V850E/IG3               | SOB0                      | 39         | 48         | 76         | N12                    |
| SO/TxD                                    | Output | Pin to transmit commands and data to V850E/IF3 and V850E/IG3                | SIB0                      | 38         | 47         | 75         | P12                    |
| SCK                                       | Output | Clock output pin for 3-wire serial communication                            | $\overline{\text{SCKB0}}$ | 40         | 49         | 77         | M14                    |
| CLK                                       | Output | Clock output pin to V850E/IF3 and V850E/IG3                                 | Not needed                | Not needed | Not needed | Not needed | Not needed             |
| RESET_OUT                                 | Output | Reset output pin to V850E/IF3 and V850E/IG3                                 | $\overline{\text{RESET}}$ | 35         | 40         | 68         | P9                     |
| FLMD0                                     | Output | Output pin to set V850E/IF3 and V850E/IG3 to debug mode or programming mode | FLMD0                     | 37         | 46         | 74         | N9                     |
| FLMD1                                     | Output | Output pin to set programming mode  | FLMD1                     | 60         | 76         | 4          | A12                    |
| HS  | Input  | Handshake signal for CSIO + HS communication                                | P43                       | 41         | 50         | 78         | M13                    |
| GND                                       | -      | Ground  | V <sub>SS0</sub>          | 32         | 37         | 65         | N7, P7                 |
|   |        |   | V <sub>SS1</sub>          | 66         | 85         | 13         | A9, B9                 |
|   |        |   | AV <sub>SS0</sub>         | 6          | 7          | 35         | E1, E2                 |
|   |        |   | AV <sub>SS1</sub>         | 11         | 12         | 40         | H1, H2                 |
|   |        |   | AV <sub>SS2</sub>         | 22         | 27         | 55         | P1, N2, M3, L4         |
|   |        |   | EV <sub>SS0</sub>         | 54         | 64         | 92         | A14, G13, G14          |
|   |        |   | EV <sub>SS1</sub>         | 79         | 1          | 29         | A1, B1, C1, C2, D3, E4 |
|   |        |   | EV <sub>SS2</sub>         | Not needed | 42         | 70         | N11, P11, P14          |
| RESET_IN                                  | Input  | Reset input pin on the target system  |                           |            |            |            |                        |

**26.2.2 Maskable functions**

Reset signal can only be masked.

The maskable functions with the debugger (ID850QB) and the corresponding functions are shown below.

**Table 26-5. Maskable Functions**

| Maskable Functions with Debugger (ID850QB) | Corresponding Function of V850E/IF3, V850E/IG3                 |
|--|--|
| NMI0                                       | ×  |
| NMI1                                       | ×  |
| NMI2                                       | ×  |
| STOP                                       | ×  |
| HOLD                                       | ×  |
| RESET                                      | Reset signal generation by $\overline{\text{RESET}}$ pin input |
| WAIT                                       | $\overline{\text{WAIT}}$ pin <sup>Note</sup> input             |

**Note**  $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only

**26.2.3 Securing of user resources**

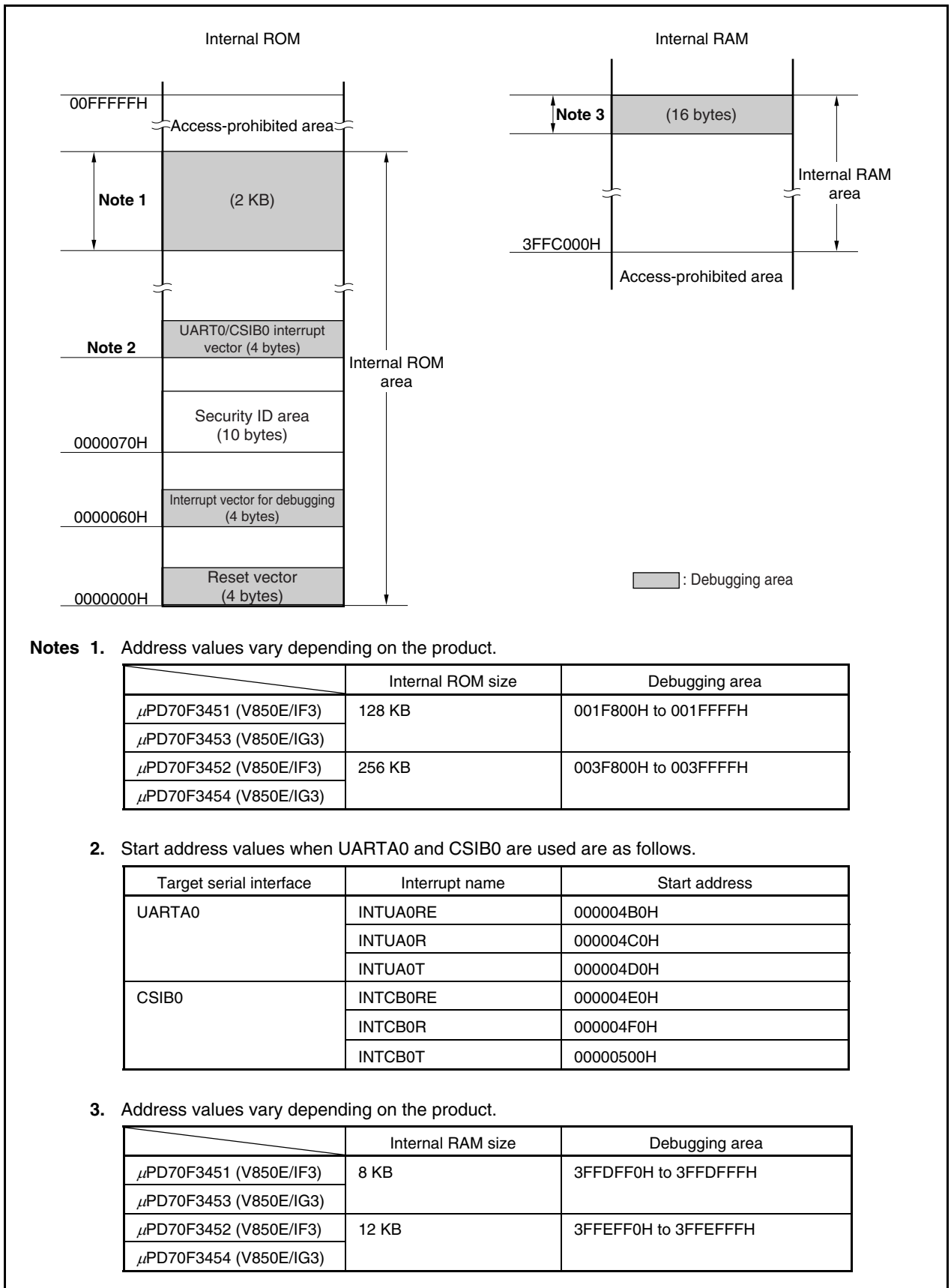
The user must prepare the following to perform communication between MINICUBE2 and the V850E/IF3, V850E/IG3 and implement each debug function. These items need to be set in the user program or using the compiler options.

**(1) Securement of memory space**

The shaded portions in Figure 26-5 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.



Figure 26-5. Memory Spaces Where Debug Monitor Programs Are Allocated



Notes 1. Address values vary depending on the product.

|                             | Internal ROM size | Debugging area       |
|-----------------------------|-------------------|----------------------|
| $\mu$ PD70F3451 (V850E/IF3) | 128 KB            | 001F800H to 001FFFFH |
| $\mu$ PD70F3453 (V850E/IG3) |                   |                      |
| $\mu$ PD70F3452 (V850E/IF3) | 256 KB            | 003F800H to 003FFFFH |
| $\mu$ PD70F3454 (V850E/IG3) |                   |                      |

2. Start address values when UARTA0 and CSIB0 are used are as follows.

| Target serial interface | Interrupt name | Start address |
|-------------------------|----------------|---------------|
| UARTA0                  | INTUA0RE       | 000004B0H     |
|                         | INTUA0R        | 000004C0H     |
|                         | INTUA0T        | 000004D0H     |
| CSIB0                   | INTCB0RE       | 000004E0H     |
|                         | INTCB0R        | 000004F0H     |
|                         | INTCB0T        | 00000500H     |

3. Address values vary depending on the product.

|                             | Internal RAM size | Debugging area       |
|-----------------------------|-------------------|----------------------|
| $\mu$ PD70F3451 (V850E/IF3) | 8 KB              | 3FFDFF0H to 3FFDFFFH |
| $\mu$ PD70F3453 (V850E/IG3) |                   |                      |
| $\mu$ PD70F3452 (V850E/IF3) | 12 KB             | 3FFEFF0H to 3FFEFFFH |
| $\mu$ PD70F3454 (V850E/IG3) |                   |                      |

- Security ID setting

The ID code must be embedded in the area between 0000070H and 0000079H in Figure 26-5, to prevent the memory from being read by an unauthorized person. For details, see **26.3 ROM Security Function**.

## (2) 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       |

**(3) Securement of area for debug monitor program**

The shaded portions in Figure 26-5 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 "INTCB0RE"
.space 4, 0xff
.section "INTCB0R"
.space 4, 0xff
.section "INTCB0T"
.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 12 KB (end address is 3FFEFFFH).

```

MROMSEG      : !LOAD ?R V0x03f800{
              MonitorROM = $PROGBITS ?A MonitorROM;
};

MRAMSEG      : !LOAD ?RW V0x03ffeff0{
              MonitorRAM = $NOBITS ?AW MonitorRAM;
};

```

**(4) Securement of communication serial interface**

UARTA0 or CSIB0 is used for communication between MINICUBE2 and the V850E/IF3, V850E/IG3. 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]

- Serial interface registers  
Do not set the registers related to UARTA0 and CSIB0 in the user program.
- Interrupt mask register  
When UARTA0 is used, do not mask the reception end interrupt (INTUA0R). When CSIB0 is used, do not mask the reception end interrupt (INTCB0R).

**(a) When UARTA0 is used**

|        |   |   |   |   |   |   |   |   |
|--------|---|---|---|---|---|---|---|---|
|        | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UA0RIC | × | 0 | × | × | × | × | × | × |

**(b) When CSIB0 is used**

|        |   |   |   |   |   |   |   |   |
|--------|---|---|---|---|---|---|---|---|
|        | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CB0RIC | × | 0 | × | × | × | × | × | × |

**Remark** ×: don't care

- Port registers when UARTA0 is used  
When UARTA0 is used for communication, 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 |
| PFCE4 | × | × | × | × | × | × | 0 | 0 |

|      |   |   |   |   |   |   |   |   |
|------|---|---|---|---|---|---|---|---|
|      | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PFC4 | × | × | × | × | × | × | 1 | 1 |

|      |   |   |   |   |   |   |   |   |
|------|---|---|---|---|---|---|---|---|
|      | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PMC4 | × | × | × | × | × | × | 1 | 1 |

**Remark** ×: 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 (P43) 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 |
| PFCE4 | x | x | x | x | x | 0 | 0 | 0 |

|      |   |   |   |   |   |   |   |   |
|------|---|---|---|---|---|---|---|---|
|      | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PFC4 | x | x | x | x | x | 0 | 0 | 0 |

|      |   |   |   |   |   |   |   |   |
|------|---|---|---|---|---|---|---|---|
|      | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PMC4 | x | x | x | x | x | 1 | 1 | 1 |

**(b) HS (P43 pin) settings**

|      |   |   |   |   |   |   |   |   |
|------|---|---|---|---|---|---|---|---|
|      | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PMC4 | x | x | x | x | 0 | x | x | x |

|     |   |   |   |   |   |   |   |   |
|-----|---|---|---|---|---|---|---|---|
|     | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PM4 | x | x | x | x | 0 | x | x | x |

|    |   |   |   |   |      |   |   |   |
|----|---|---|---|---|------|---|---|---|
|    | 7 | 6 | 5 | 4 | 3    | 2 | 1 | 0 |
| P4 | x | x | x | x | Note | x | x | x |

**Note** Writing to this bit is prohibited.

The 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

## 26.2.4 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 V850E/IF3, V850E/IG3, are masked
- Standby mode is entered while standby release by a maskable interrupt is prohibited
- Mode for communication between MINICUBE2 and the V850E/IF3, V850E/IG3 is UARTA0, and the peripheral 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 V850E/IF3, V850E/IG3, are masked
- Standby mode is entered while standby release by a maskable interrupt is prohibited
- Mode for communication between MINICUBE2 and the V850E/IF3, V850E/IG3 is UARTA0, and the peripheral clock has been stopped
- Mode for communication between MINICUBE2 and the V850E/IF3, V850E/IG3 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 V850E/IF3, V850E/IG3 is CSIB0
- Mode for communication between MINICUBE2 and the V850E/IF3, V850E/IG3 is UARTA0, and the peripheral clock has not stopped.

### (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) 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.

## 26.3 ROM Security Function

### 26.3.1 Security ID

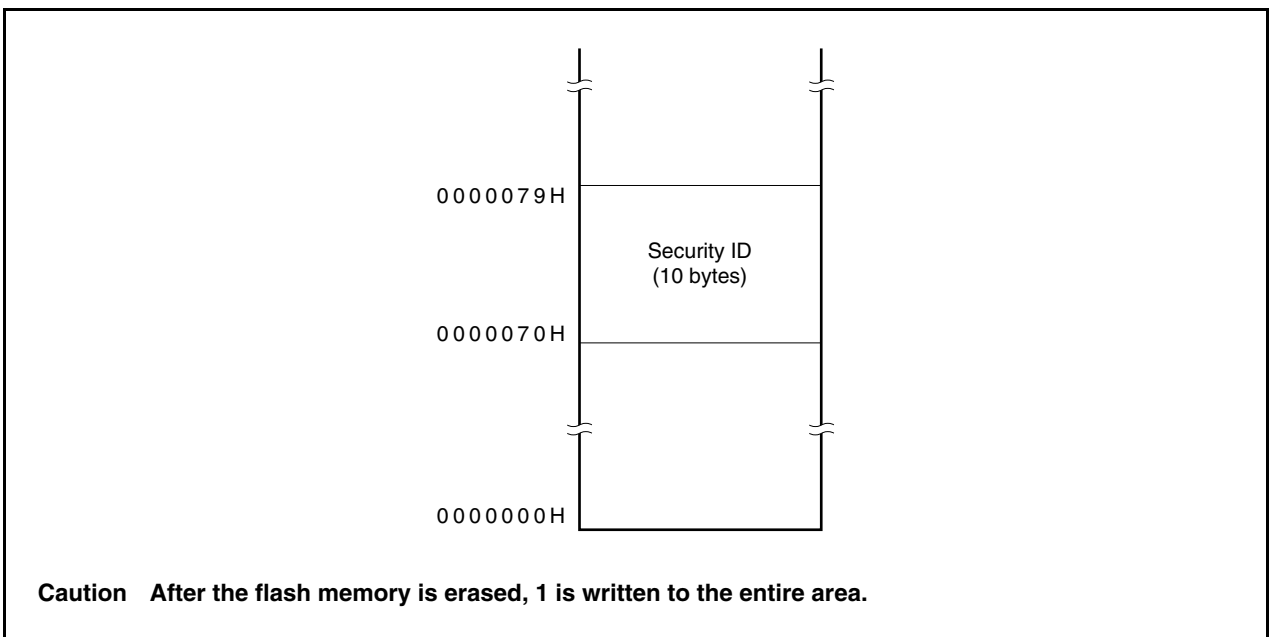
The flash memory versions of the V850E/IF3 and V850E/IG3 perform authentication using a 10-byte ID code to prevent the contents of the flash memory from being read by an unauthorized person during on-chip debugging by the on-chip debug emulator.

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 26-6. Security ID Area**



**26.3.2 Setting**

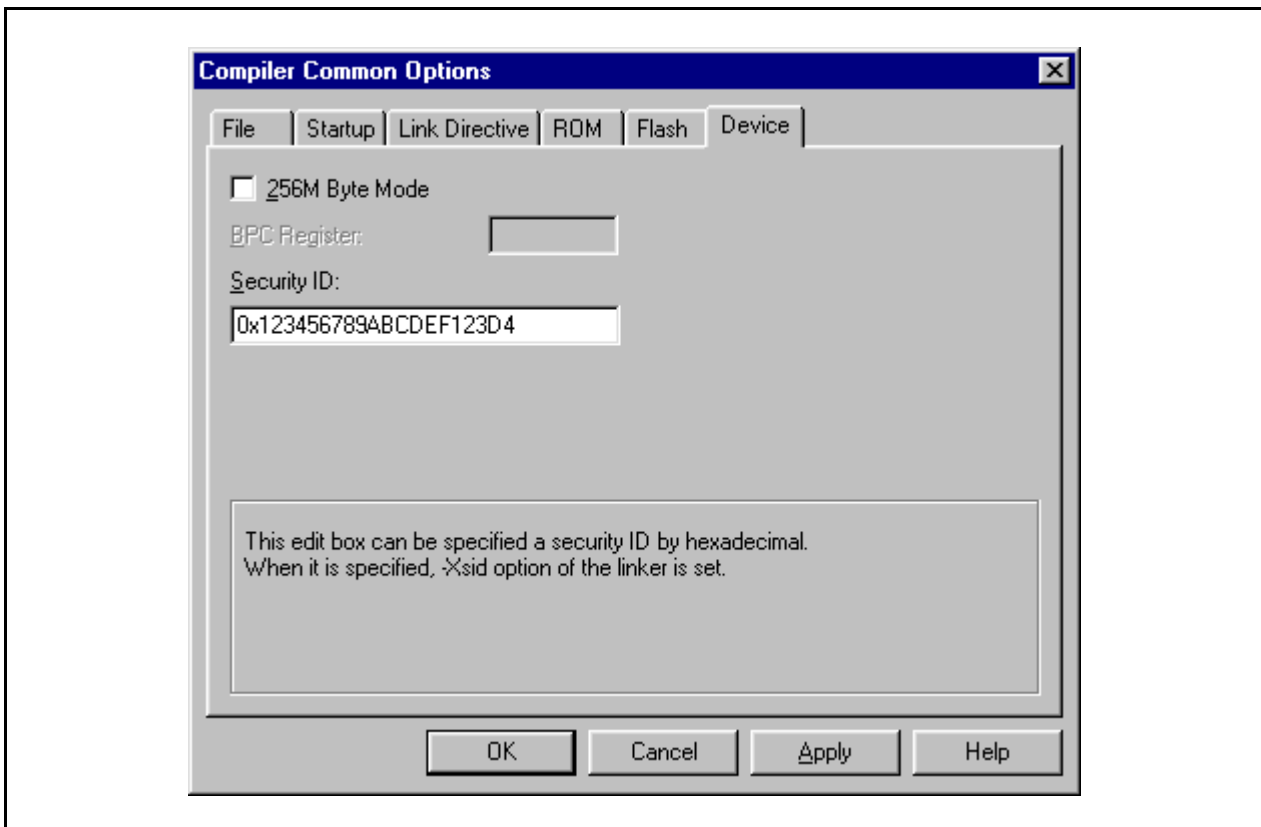
The following shows how to set the ID code as shown in Table 26-6.

When the ID code is set as shown in Table 26-6, the ID code input in the configuration dialog box of the ID850QB is "123456789ABCDEF123D4" (the ID code is case-insensitive).

**Table 26-6. ID Code**

| Address | Value |
|---------|-------|
| 0x70    | 0x12  |
| 0x71    | 0x34  |
| 0x72    | 0x56  |
| 0x73    | 0x78  |
| 0x74    | 0x9A  |
| 0x75    | 0xBC  |
| 0x76    | 0xDE  |
| 0x77    | 0XF1  |
| 0x78    | 0x23  |
| 0x79    | 0xD4  |

<R> The ID code can be specified in the Compiler Common Options dialog box in PM+ if a device file that supports CA850 Ver. 2.60 and later and the security ID is used.





## CHAPTER 27 FLASH MEMORY

The V850E/IF3 and V850E/IG3 have a 128 KB or 256 KB on-chip flash memory.

- $\mu$ PD70F3451 (V850E/IF3), 70F3453 (V850E/IG3): 128 KB on-chip flash memory version
- $\mu$ PD70F3452 (V850E/IF3), 70F3454 (V850E/IG3): 256 KB on-chip flash memory version

Flash memory can be rewritten with the flash memory programmer or using the self programming mode.

Writing to the flash memory programmer can be performed with the flash memory programmer that is connected to the target system.

Writing in the self programming mode can be performed with an application program, without using the flash memory programmer.

Flash memory versions are commonly used in the following development environments and mass production applications.

- For altering software after the V850E/IF3 and V850E/IG3 is soldered onto the target system.
- For differentiating software according to the specification in small scale production of various models.
- For data adjustment when starting mass production.

### 27.1 Features

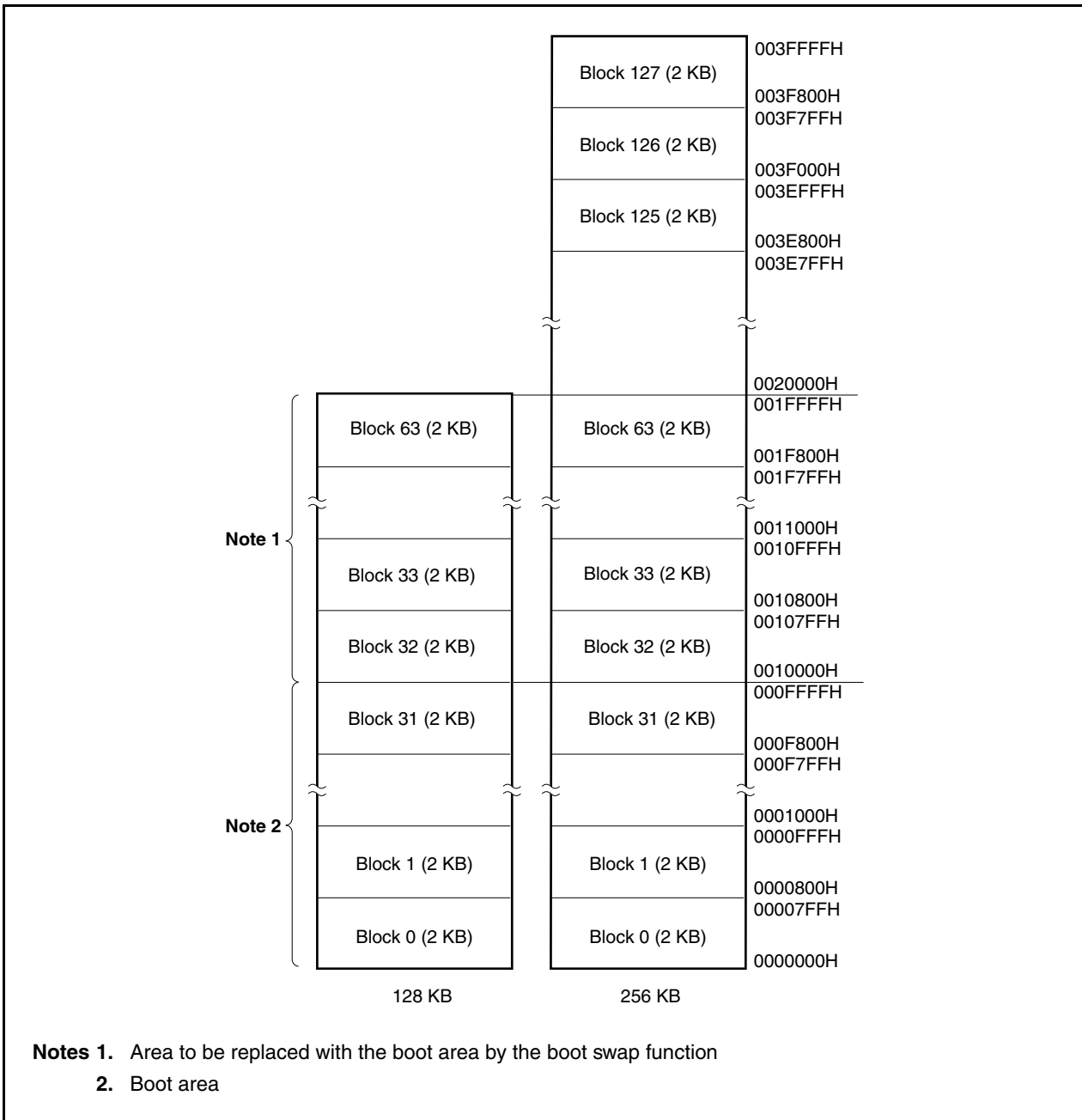
- All area batch erase or erase in block units (2 KB)
- Communication through serial interface from the flash memory programmer
- Erase/write voltage: Erase/write is possible with a single power supply
- On-board programming
- Flash memory self programming possible
- Secure rewriting of entire flash memory area by self programming using boot swap function
- Rewriting method
  - Rewriting by communication with flash memory programmer via serial interface (on-board/off-board programming)
  - Rewriting flash memory by user program (self programming)
- Rewriting flash memory and read disable function supported (security enforced)
- Interrupts can be acknowledged during self programming.

<R> 27.2 Memory Configuration

The internal flash memory area of the V850E/IF3 and V850E/IG3 is divided into 64 or 128 blocks and can be programmed/erased in block units. All the blocks can also be erased at once.

When the boot swap function is used, the physical memory located at the addresses of blocks 0 to 31 is replaced by the physical memory located at the addresses of blocks 32 to 63. For details of the boot swap function, see 27.9 **Rewriting by Self Programming**.

Figure 27-1. Flash Memory Mapping



&lt;R&gt;

**27.3 Functional Overview**

The internal flash memory of the V850E/IF3 and V850E/IG3 can be rewritten by using the rewrite function of the dedicated flash programmer, regardless of whether the V850E/IF3 and V850E/IG3 have already been mounted on the target system or not (off-board/on-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 27-1. 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 memory 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 memory 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 on-chip flash memory area. Therefore, the rewrite program must be transferred to the internal RAM or external memory ( $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only) in advance). | Normal operation mode         |

**Remark** The FA series is a product of Naito Densai Machida Mfg. Co., Ltd.

**Table 27-2. 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.   | √  | ×<br>(supported by specifying area for block erasure)                    |
| 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 memory programmer.      | √  | ×<br>(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 can be prohibited. | √  | ×<br>(Only values set by on-board/off-board programming can be retained) |

**Table 27-3. Security Functions**

| Function                     | Function Outline   | Support  |                  |
|------------------------------|--|--|------------------|
|                              |  | On-Board/Off-Board Programming                     | Self Programming |
| Block erase command prohibit | Execution of a block erase command on all blocks is prohibited. Setting of prohibition can be initialized by execution of a chip erase command.  | For details, see <b>27.3.2 Security function</b> . |                  |
| Chip erase command prohibit  | Execution of block erase and chip erase commands on all blocks is prohibited. Once prohibition is set, setting of prohibition cannot be initialized because the chip erase command cannot be executed. |  |                  |
| Program command prohibit     | Write and block erase commands on all blocks are prohibited. Setting of prohibition can be initialized by execution of the chip erase command.   |  |                  |
| Read command prohibit        | Read command on all blocks is prohibited. Setting of prohibition can be initialized by execution of a chip erase command.  |  |                  |

27.3.1 Erase units

(1) All area batch erase

Flash memory area 128 KB or 256 KB can be erased at the same time.

(2) Erase in block units

Can be erased in block units.

- $\mu$ PD70F3451 (V850E/IF3), 70F3453 (V850E/IG3): Block 0 to block 63: Each 2 KB
- $\mu$ PD70F3452 (V850E/IF3), 70F3454 (V850E/IG3): Block 0 to block 127: Each 2 KB

27.3.2 Security function

The commands and functions can be secured when the flash memory is rewritten.

As a factory-set condition in the V850E/IF3 and V850E/IG3, "All enabled" is selected and the flash memory to which nothing has been written is secured.

(1) In flash memory programming mode

| Security Setting (Flag) \ Command               | All Enabled | Reading Prohibited | Writing Prohibited | Chip Erase Prohibited | Block Erase Prohibited | Boot Block Cluster Rewriting Prohibited |
|---|-------------|--------------------|--------------------|-----------------------|------------------------|---|
| Read  | √           | ×                  | √                  | √                     | √                      | √                                       |
| Write   | √           | √                  | ×                  | √                     | √                      | △                                       |
| Chip erase                                      | √           | √                  | √                  | ×                     | √                      | ×                                       |
| Block erase                                     | √           | √                  | ×                  | ×                     | ×                      | △                                       |
| Changing of security setting                    | √           | <b>Note 1</b>      | <b>Note 1</b>      | <b>Note 1</b>         | <b>Note 1</b>          | <b>Notes 1, 2</b>                       |
| Other commands (such as blank check and verify) | √           | √                  | √                  | √                     | √                      | √                                       |

√: Command can be accepted.

△: Command can be accepted in areas other than boot block cluster.

×: Protect error

**Notes 1.** Enabled setting can be disabled. However, a protect error occurs if disabled setting is enabled. To enable security that has been disabled, the chip must be erased. If chip erase prohibition or boot block cluster rewriting prohibition is selected, the security setting cannot be enabled.

**2.** If boot block cluster rewriting prohibition is set, the boot block cluster last block number (128 KB version: 63, 256 KB version: 127) cannot be changed.

(2) In self programming mode

| Security Setting (Flag)<br>Command                                  | All Enabled | Reading Prohibited | Writing Prohibited | Chip Erase Prohibited | Block Erase Prohibited | Boot Block Cluster Rewriting Prohibited |
|---|-------------|--------------------|--------------------|-----------------------|------------------------|---|
| FlashWordRead   | √           | √                  | √                  | √                     | √                      | √                                       |
| FlashWordWrite  | √           | √                  | √                  | √                     | √                      | △                                       |
| FlashBlockErase   | √           | √                  | √                  | √                     | √                      | △                                       |
| FlashBootSwap   | √           | √                  | √                  | √                     | √                      | ×                                       |
| FlashSetInfo  | √           | <b>Note 1</b>      | <b>Note 1</b>      | <b>Note 1</b>         | <b>Note 1</b>          | <b>Notes 1, 2</b>                       |
| Other functions (such as FlashBlockBlankCheck and FlashBlockVerify) | √           | √                  | √                  | √                     | √                      | √                                       |

√: Command can be accepted.

△: Function can be executed in an area other than boot block cluster.

×: Protect error

- Notes 1.** Enabled setting can be disabled. However, a protect error occurs if disabled setting is enabled. To enable security that has been disabled, the chip must be erased in the flash self programming mode. If chip erase prohibition is selected, the security setting cannot be enabled.
- 2.** If boot block cluster rewriting prohibition is set, the boot block cluster last block number (128 KB version: 63, 256 KB version: 127) and boot swap cluster setting flag cannot be changed.

## 27.4 Writing with Flash Memory Programmer

Writing can be performed either on-board or off-board using a flash memory programmer (PG-FP4, PG-FP5, FL-PR4, or FL-PR5) and MINICUBE2.

### (1) On-board programming

The contents of the flash memory are rewritten after the V850E/IF3 or V850E/IG3 is mounted on the target system. Mount connectors, etc., on the target system to connect the flash memory programmer.

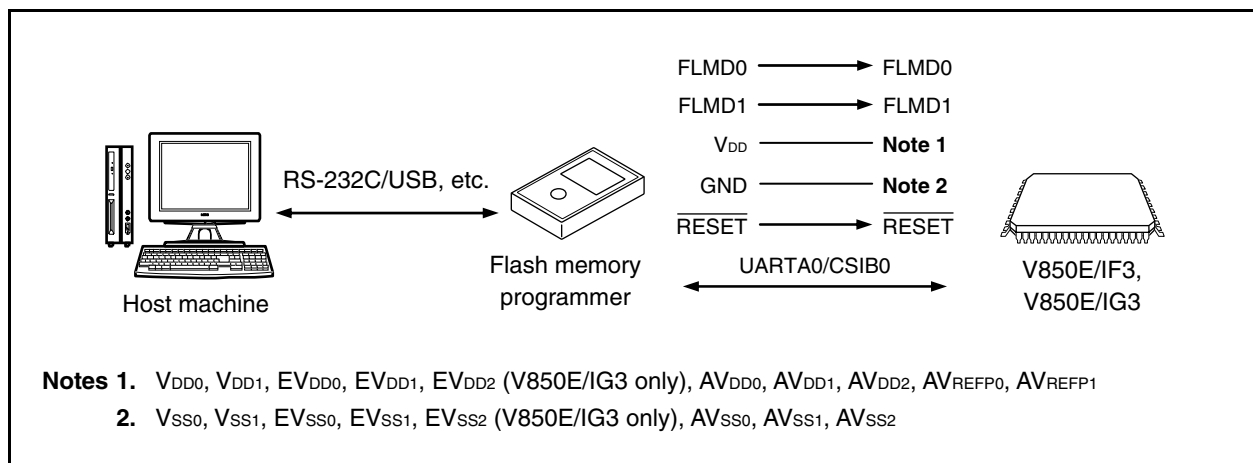
### (2) Off-board programming

Writing to a flash memory is performed before mounting the V850E/IF3 or V850E/IG3 on the target system.

**Remark** FL-PR4 and FL-PR5 are products of Naito Densetsu Machida Mfg. Co., Ltd.

### 27.5 Flash Memory Programming Environment

The following shows the environment required for writing programs to the flash memory of the V850E/IF3 and V850E/IG3.



A host machine is required for controlling the flash memory programmer.

UARTA0 or CSIB0 is used for the interface between the flash memory programmer and the V850E/IF3 or V850E/IG3 to perform writing, erasing, etc. Supply the operating clock of the V850E/IF3 or V850E/IG3 via the oscillator configured on the V850E/IF3 or V850E/IG3 board using a resonator and a capacitor.

**Table 27-4. Environment and Communication Mode**

| Environment   | Communication Mode |       |                     |
|---|--------------------|-------|---------------------|
|   | UARTA0             | CSIB0 | CSIB0 for Handshake |
| Flash memory programmer<br>(PG-FP4, PG-FP5, FL-PR4, and FL-PR5) | √                  | √     | √                   |
| MINICUBE2   | √                  | ×     | √                   |

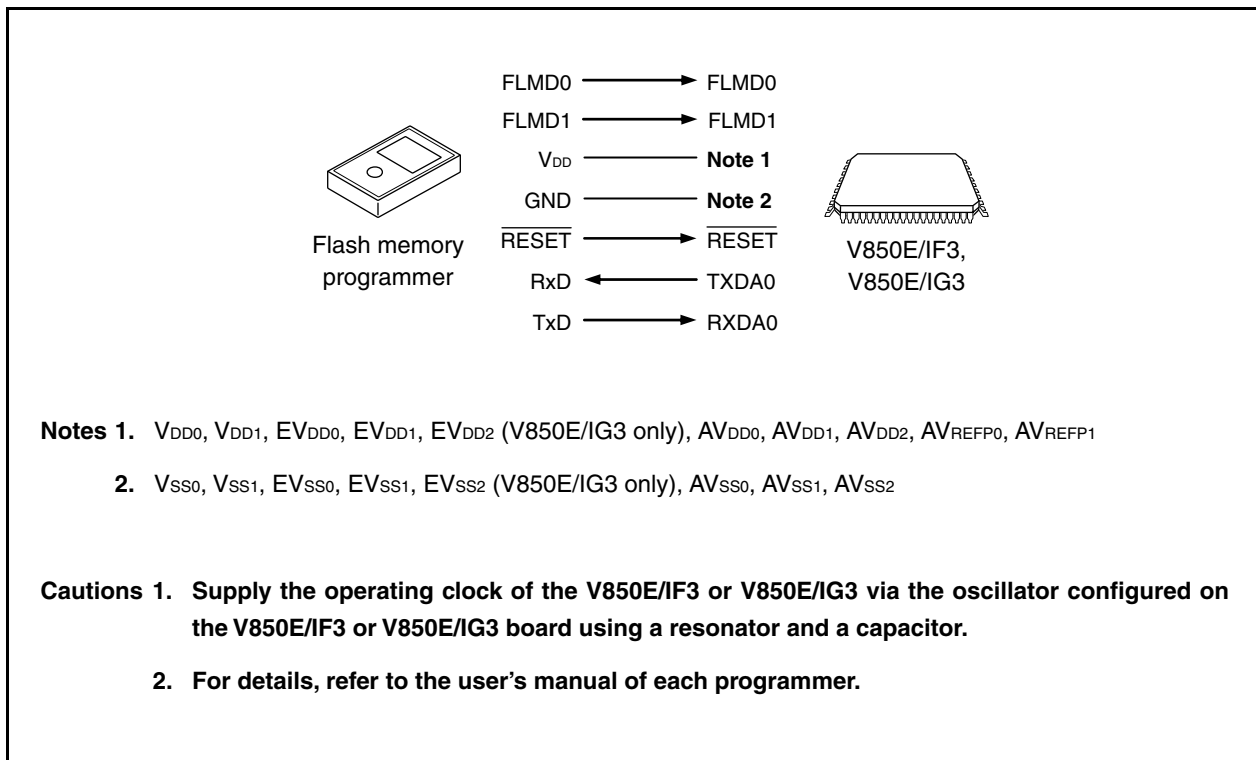
**Remark** √: Supported, ×: Not supported



## 27.6 Communication Method of Flash Memory Programming

### (1) UART0 communication method

Transfer rate: 9600 to 153,600 bps (LSB first)



**Table 27-5. Wiring Correspondence Between Dedicated Flash Memory Programmer and V850E/IF3, V850E/IG3**

| Pin No. | Dedicated Flash Memory Programmer (PG-FP4 or PG-FP5) | I/O (PG-FP4 or PG-FP5 Side) | Pin Name             | V850E/IF3, V850E/IG3 |           |    |                |
|---------|--|-----------------------------|----------------------|----------------------|-----------|----|----------------|
|         |  |                             |                      | Pin No.              |           |    |                |
|         |  |                             |                      | V850E/IF3            | V850E/IG3 |    |                |
|         |  |                             |                      | GC                   | GC        | GF | F1             |
| 1       | GND  | –                           | V <sub>SS0</sub>     | 32                   | 37        | 65 | N7, P7         |
|         |  |                             | V <sub>SS1</sub>     | 66                   | 85        | 13 | A9, B9         |
|         |  |                             | EV <sub>SS0</sub>    | 54                   | 64        | 92 | A14, G13, G14  |
|         |  |                             | EV <sub>SS1</sub>    | 79                   | 1         | 29 | <b>Note 1</b>  |
|         |  |                             | EV <sub>SS2</sub>    | –                    | 42        | 70 | N11, P11, P14  |
|         |  |                             | AV <sub>SS0</sub>    | 6                    | 7         | 35 | E1, E2         |
|         |  |                             | AV <sub>SS1</sub>    | 11                   | 12        | 40 | H1, H2         |
|         |  |                             | AV <sub>SS2</sub>    | 22                   | 27        | 55 | L4, M3, N2, P1 |
| 2       | RESET  | Output                      | RESET                | 35                   | 40        | 68 | P9             |
| 3       | SI/RxD   | Input                       | TXDA0                | 39                   | 48        | 76 | N12            |
| 4       | V <sub>DD</sub>                                      | –                           | V <sub>DD0</sub>     | 30                   | 35        | 63 | N5, P5         |
|         |  |                             | V <sub>DD1</sub>     | 68                   | 87        | 15 | A7, B7         |
|         |  |                             | EV <sub>DD0</sub>    | 55                   | 65        | 93 | F13, F14       |
|         |  |                             | EV <sub>DD1</sub>    | 80                   | 100       | 28 | A3, B2         |
|         |  |                             | EV <sub>DD2</sub>    | –                    | 41        | 69 | N10, P10       |
|         |  |                             | AV <sub>DD0</sub>    | 8                    | 9         | 37 | F1             |
|         |  |                             | AV <sub>DD1</sub>    | 9                    | 10        | 38 | G1             |
|         |  |                             | AV <sub>DD2</sub>    | 21                   | 26        | 54 | M1, M2         |
|         |  |                             | AV <sub>REFP0</sub>  | 7                    | 8         | 36 | F2             |
|         |  |                             | AV <sub>REFP1</sub>  | 10                   | 11        | 39 | G2             |
| 5       | SO/TxD   | Output                      | RXDA0                | 38                   | 47        | 75 | P12            |
| 6       | V <sub>PP</sub>                                      | ×                           | NC                   | –                    | –         | –  | –              |
| 7       | SCK  | ×                           | NC                   | –                    | –         | –  | –              |
| 8       | H/S  | ×                           | NC                   | –                    | –         | –  | –              |
| 9       | CLK <sup>Note 2</sup>                                | Output                      | X1 <sup>Note 2</sup> | 33                   | 38        | 66 | P8             |
| 10      | VDE  | ×                           | NC                   | –                    | –         | –  | –              |
| 11      | V <sub>DD2</sub>                                     | –                           | NC                   | –                    | –         | –  | –              |
| 12      | FLMD1  | Output                      | <b>Note 3</b>        | 60                   | 76        | 4  | A12            |
| 13      | RFU-1  | ×                           | NC                   | –                    | –         | –  | –              |
| 14      | FLMD0  | Output                      | FLMD0                | 37                   | 46        | 74 | N9             |
| 15      | Not used   | ×                           | NC                   | –                    | –         | –  | –              |
| 16      | Not used   | ×                           | NC                   | –                    | –         | –  | –              |

**Notes 1.** A1, B1, C1, C2, D3, and E4

**2.** In the V850E/IF3 and V850E/IG3, external clock input is prohibited. Mount the resonator on board.

**3.** Connect to FLMD1 or GND via a resistor.

**Remark** NC: No Connection

GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)

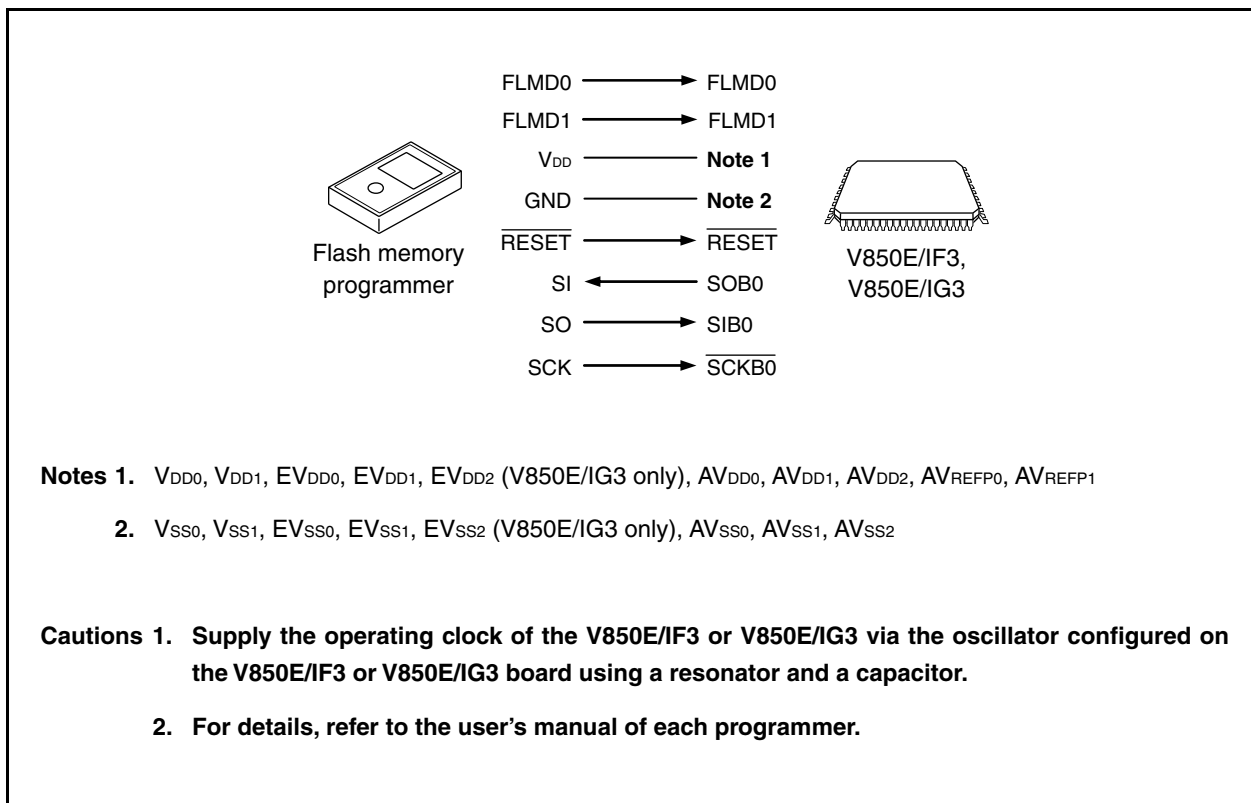
GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)

GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

**(2) CSIB0 communication method**

Transfer rate: Up to 2.5 Mbps (MSB first)



The flash memory programmer outputs (master) transfer clocks and the V850E/IF3 or V850E/IG3 operates as a slave.

**Table 27-6. Wiring Correspondence Between Dedicated Flash Memory Programmer and V850E/IF3, V850E/IG3**

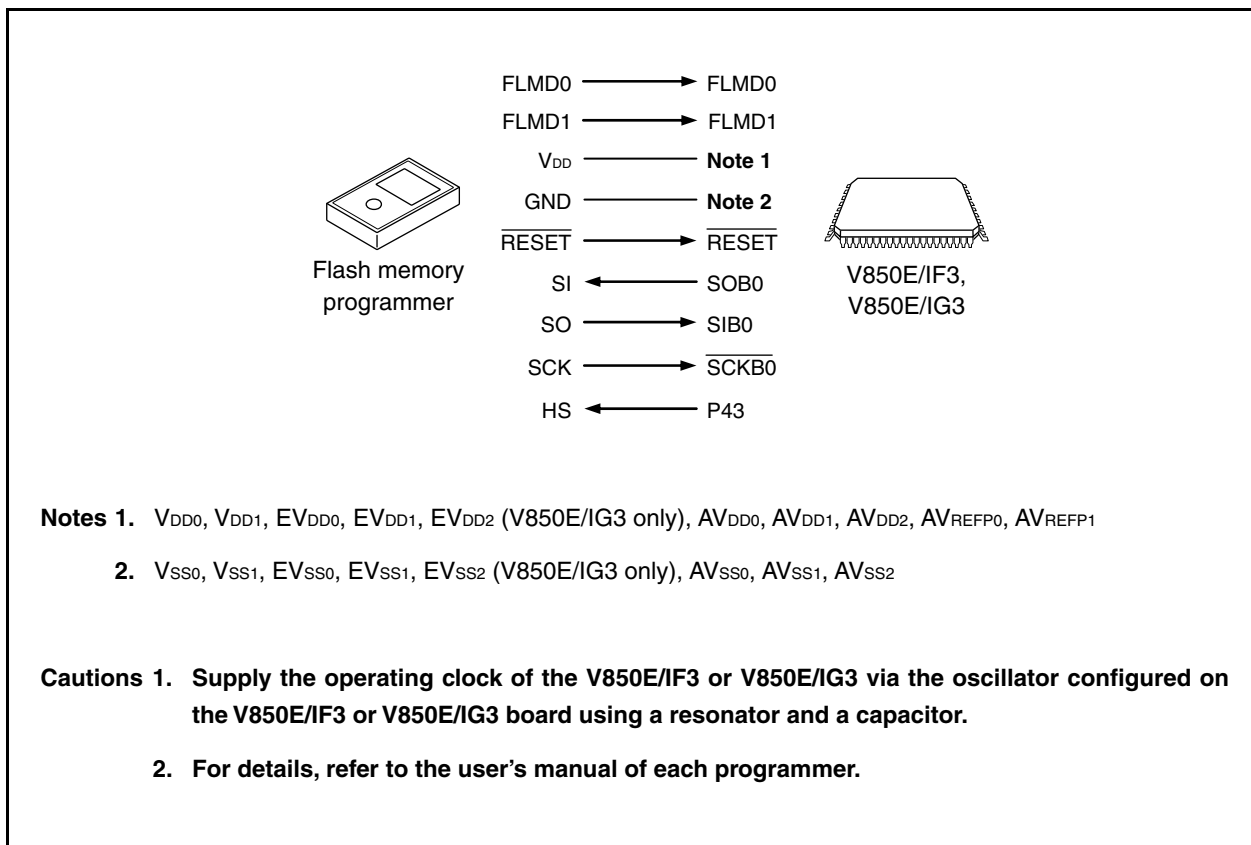
| Pin No. | Flash Memory Programmer<br>(PG-FP4 or PG-FP5) | I/O<br>(PG-FP4 or PG-FP5 Side) | Pin Name             | V850E/IF3, V850E/IG3 |           |    |                |
|---------|---|--------------------------------|----------------------|----------------------|-----------|----|----------------|
|         |   |                                |                      | Pin No.              |           |    |                |
|         |   |                                |                      | V850E/IF3            | V850E/IG3 |    |                |
|         |   |                                |                      | GC                   | GC        | GF | F1             |
| 1       | GND   | -                              | V <sub>SS0</sub>     | 32                   | 37        | 65 | N7, P7         |
|         |   |                                | V <sub>SS1</sub>     | 66                   | 85        | 13 | A9, B9         |
|         |   |                                | EV <sub>SS0</sub>    | 54                   | 64        | 92 | A14, G13, G14  |
|         |   |                                | EV <sub>SS1</sub>    | 79                   | 1         | 29 | <b>Note 1</b>  |
|         |   |                                | EV <sub>SS2</sub>    | -                    | 42        | 70 | N11, P11, P14  |
|         |   |                                | AV <sub>SS0</sub>    | 6                    | 7         | 35 | E1, E2         |
|         |   |                                | AV <sub>SS1</sub>    | 11                   | 12        | 40 | H1, H2         |
|         |   |                                | AV <sub>SS2</sub>    | 22                   | 27        | 55 | L4, M3, N2, P1 |
| 2       | RESET   | Output                         | RESET                | 35                   | 40        | 68 | P9             |
| 3       | SI/RxD  | Input                          | SOB0                 | 39                   | 48        | 76 | N12            |
| 4       | V <sub>DD</sub>                               | -                              | V <sub>DD0</sub>     | 30                   | 35        | 63 | N5, P5         |
|         |   |                                | V <sub>DD1</sub>     | 68                   | 87        | 15 | A7, B7         |
|         |   |                                | EV <sub>DD0</sub>    | 55                   | 65        | 93 | F13, F14       |
|         |   |                                | EV <sub>DD1</sub>    | 80                   | 100       | 28 | A3, B2         |
|         |   |                                | EV <sub>DD2</sub>    | -                    | 41        | 69 | N10, P10       |
|         |   |                                | AV <sub>DD0</sub>    | 8                    | 9         | 37 | F1             |
|         |   |                                | AV <sub>DD1</sub>    | 9                    | 10        | 38 | G1             |
|         |   |                                | AV <sub>DD2</sub>    | 21                   | 26        | 54 | M1, M2         |
|         |   |                                | AV <sub>REFP0</sub>  | 7                    | 8         | 36 | F2             |
|         |   |                                | AV <sub>REFP1</sub>  | 10                   | 11        | 39 | G2             |
| 5       | SO/TxD  | Output                         | SIB0                 | 38                   | 47        | 75 | P12            |
| 6       | V <sub>PP</sub>                               | x                              | NC                   | -                    | -         | -  | -              |
| 7       | SCK   | Output                         | SCKB0                | 40                   | 49        | 77 | M14            |
| 8       | H/S   | x                              | NC                   | -                    | -         | -  | -              |
| 9       | CLKZ <sup>Note 2</sup>                        | Output                         | X1 <sup>Note 2</sup> | 33                   | 38        | 66 | P8             |
| 10      | VDE   | x                              | NC                   | -                    | -         | -  | -              |
| 11      | V <sub>DD2</sub>                              | -                              | NC                   | -                    | -         | -  | -              |
| 12      | FLMD1   | Output                         | <b>Note 3</b>        | 60                   | 76        | 4  | A12            |
| 13      | RFU-1   | x                              | NC                   | -                    | -         | -  | -              |
| 14      | FLMD0   | Output                         | FLMD0                | 37                   | 46        | 74 | N9             |
| 15      | Not used                                      | x                              | NC                   | -                    | -         | -  | -              |
| 16      | Not used                                      | x                              | NC                   | -                    | -         | -  | -              |

- Notes 1.** A1, B1, C1, C2, D3, and E4  
**2.** In the V850E/IF3 and V850E/IG3, external clock input is prohibited. Mount the resonator on board.  
**3.** Connect to FLMD1 or GND via a resistor.

**Remark** NC: No Connection  
 GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)  
 GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)  
 GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)  
 F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

**(3) CSIB0 communication method supporting handshake**

Transfer rate: Up to 2.5 Mbps (MSB first)



The flash memory programmer outputs the transfer clock, and the V850E/IF3 or V850E/IG3 operates as a slave.

When the PG-FP4 or PG-FP5 is used, it sends the following signals to the V850E/IF3 or V850E/IG3. For details, refer to the **PG-FP4 User's Manual (U15260E)** or **PG-FP5 User's Manual (U18865E)**.

**Table 27-7. Wiring Correspondence Between Dedicated Flash Memory Programmer and V850E/IF3, V850E/IG3**

| Pin No. | Flash memory programmer<br>(PG-FP4 or PG-FP5) | I/O<br>(PG-FP4 or PG-FP5 Side) | Pin Name             | V850E/IF3, V850E/IG3 |           |    |                |
|---------|---|--------------------------------|----------------------|----------------------|-----------|----|----------------|
|         |   |                                |                      | Pin No.              |           |    |                |
|         |   |                                |                      | V850E/IF3            | V850E/IG3 |    |                |
|         |   |                                |                      | GC                   | GC        | GC | F1             |
| 1       | GND   | -                              | V <sub>SS0</sub>     | 32                   | 37        | 65 | N7, P7         |
|         |   |                                | V <sub>SS1</sub>     | 66                   | 85        | 13 | A9, B9         |
|         |   |                                | EV <sub>SS0</sub>    | 54                   | 64        | 92 | A14, G13, G14  |
|         |   |                                | EV <sub>SS1</sub>    | 79                   | 1         | 29 | <b>Note 1</b>  |
|         |   |                                | EV <sub>SS2</sub>    | -                    | 42        | 70 | N11, P11, P14  |
|         |   |                                | AV <sub>SS0</sub>    | 6                    | 7         | 35 | E1, E2         |
|         |   |                                | AV <sub>SS1</sub>    | 11                   | 12        | 40 | H1, H2         |
|         |   |                                | AV <sub>SS2</sub>    | 22                   | 27        | 55 | L4, M3, N2, P1 |
| 2       | RESET   | Output                         | RESET                | 35                   | 40        | 68 | P9             |
| 3       | SI/RxD  | Input                          | SOB0                 | 39                   | 48        | 76 | N12            |
| 4       | V <sub>DD</sub>                               | -                              | V <sub>DD0</sub>     | 30                   | 35        | 63 | N5, P5         |
|         |   |                                | V <sub>DD1</sub>     | 68                   | 87        | 15 | A7, B7         |
|         |   |                                | EV <sub>DD0</sub>    | 55                   | 65        | 93 | F13, F14       |
|         |   |                                | EV <sub>DD1</sub>    | 80                   | 100       | 28 | A3, B2         |
|         |   |                                | EV <sub>DD2</sub>    | -                    | 41        | 69 | N10, P10       |
|         |   |                                | AV <sub>DD0</sub>    | 8                    | 9         | 37 | F1             |
|         |   |                                | AV <sub>DD1</sub>    | 9                    | 10        | 38 | G1             |
|         |   |                                | AV <sub>DD2</sub>    | 21                   | 26        | 54 | M1, M2         |
|         |   |                                | AV <sub>REFP0</sub>  | 7                    | 8         | 36 | F2             |
|         |   |                                | AV <sub>REFP1</sub>  | 10                   | 11        | 39 | G2             |
| 5       | SO/TxD  | Output                         | SIB0                 | 38                   | 47        | 75 | P12            |
| 6       | V <sub>PP</sub>                               | x                              | NC                   | -                    | -         | -  | -              |
| 7       | SCK   | Output                         | SCKB0                | 40                   | 49        | 77 | M14            |
| 8       | H/S   | Input                          | P43                  | 41                   | 50        | 78 | M13            |
| 9       | CLK <sup>Note 2</sup>                         | Output                         | X1 <sup>Note 2</sup> | 33                   | 38        | 66 | P8             |
| 10      | VDE   | x                              | NC                   | -                    | -         | -  | -              |
| 11      | V <sub>DD2</sub>                              | -                              | NC                   | -                    | -         | -  | -              |
| 12      | FLMD1   | Output                         | <b>Note 3</b>        | 60                   | 76        | 4  | A12            |
| 13      | RFU-1   | x                              | NC                   | -                    | -         | -  | -              |
| 14      | FLMD0   | Output                         | FLMD0                | 37                   | 46        | 74 | N9             |
| 15      | Not used                                      | x                              | NC                   | -                    | -         | -  | -              |
| 16      | Not used                                      | x                              | NC                   | -                    | -         | -  | -              |

**Notes 1.** A1, B1, C1, C2, D3, and E4

**2.** In the V850E/IF3 and V850E/IG3, external clock input is prohibited. Mount the resonator on board.

**3.** Connect to FLMD1 or GND via a resistor.

**Remark** NC: No Connection

GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)

GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)

GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

## 27.7 Pin Processing During Flash Memory Programming

When performing on-board programming, mount a connector on the target system to connect to the flash memory programmer.

In the flash memory programming mode, all the pins not used for flash memory programming become the same status as that immediately after reset in the normal operation mode. Therefore, because all the ports become high-impedance status, pin processing is required when the external device does not acknowledge the high-impedance status.

### 27.7.1 Power supply

Supply the same power supplies ( $V_{DD0}$ ,  $V_{DD1}$ ,  $V_{SS0}$ ,  $V_{SS1}$ ,  $EV_{DD0}$ ,  $EV_{DD1}$ ,  $EV_{DD2}$  (V850E/IG3 only),  $EV_{SS0}$ ,  $EV_{SS1}$ ,  $EV_{SS2}$  (V850E/IG3 only),  $AV_{DD0}$ ,  $AV_{DD1}$ ,  $AV_{DD2}$ ,  $AV_{SS0}$ ,  $AV_{SS1}$ ,  $AV_{SS2}$ ,  $AV_{REFP0}$ , and  $AV_{REFP1}$ ) as in the normal operation mode. Connect  $V_{DD}$  and GND of the flash memory programmer to  $V_{DD0}$ ,  $V_{DD1}$ ,  $V_{SS0}$ ,  $V_{SS1}$ ,  $EV_{DD0}$ ,  $EV_{DD1}$ ,  $EV_{DD2}$  (V850E/IG3 only),  $EV_{SS0}$ ,  $EV_{SS1}$ ,  $EV_{SS2}$  (V850E/IG3 only),  $AV_{DD0}$ ,  $AV_{DD1}$ ,  $AV_{DD2}$ ,  $AV_{SS0}$ ,  $AV_{SS1}$ ,  $AV_{SS2}$ ,  $AV_{REFP0}$ , and  $AV_{REFP1}$ . ( $V_{DD}$  of the flash memory programmer is provided with a power supply monitoring function.)

In the flash memory programming mode (including flash memory self programming), insert capacitors between  $V_{DD0}$ ,  $V_{DD1}$  pins and  $V_{SS0}$ ,  $V_{SS1}$  pins, and between  $EV_{DD0}$ ,  $EV_{DD1}$ ,  $EV_{DD2}$  (V850E/IG3 only) pins and  $EV_{SS0}$ ,  $EV_{SS1}$ ,  $EV_{SS2}$  (V850E/IG3 only) pins to stabilize the power supply voltage.

### 27.7.2 Pins used

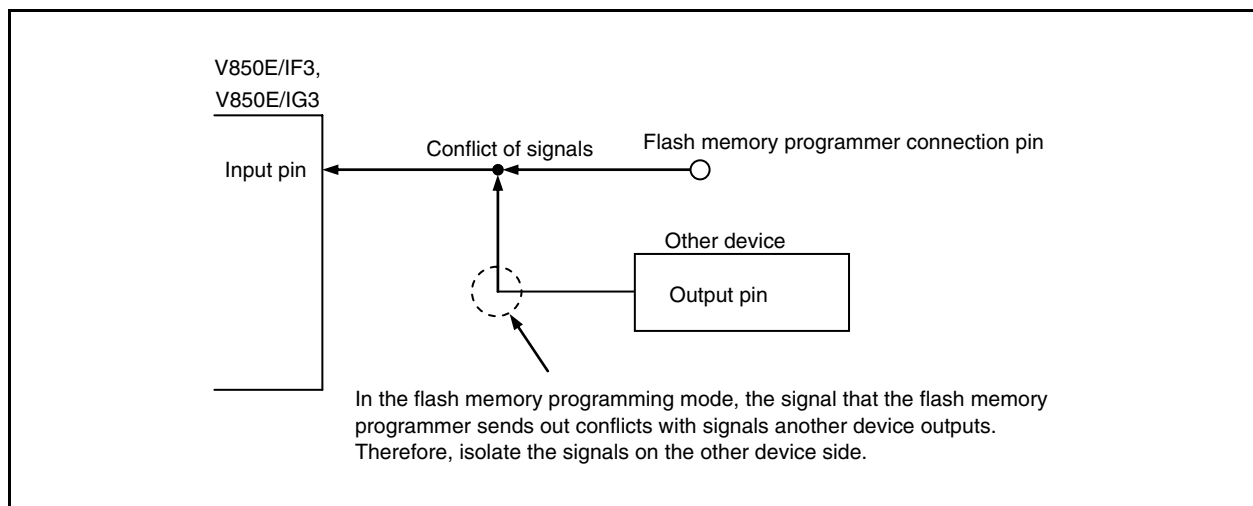
The following shows the pins used by each interface.

| Communication Mode  | Pins Used                            |
|---------------------|--------------------------------------|
| UARTA0              | TXD0, RXD0                           |
| CSIB0               | SOB0, SIB0, $\overline{SCKB0}$       |
| CSIB0 for handshake | SOB0, SIB0, $\overline{SCKB0}$ , P43 |

When connecting a flash memory programmer to an interface pin that is connected to other devices on-board, care should be taken to avoid a conflict of signals or the malfunction of other devices.

**(1) Conflict of signals**

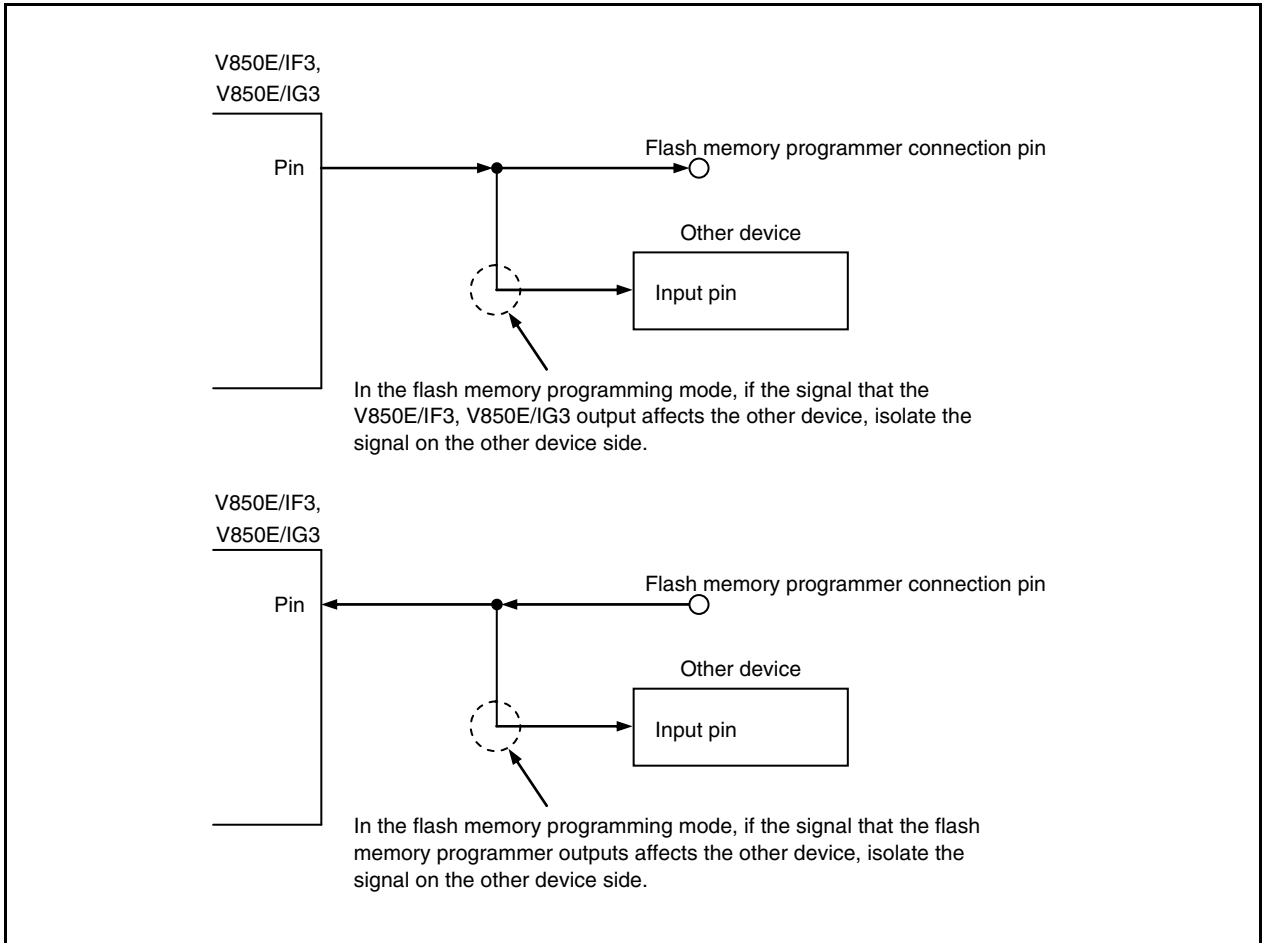
When the flash memory programmer (output) is connected to an interface pin (input) that is connected to another device (output), a conflict of signals occurs. To avoid the conflict of signals, isolate the connection to the other device or set the other device to the output high-impedance status.





**(2) Malfunction of other device**

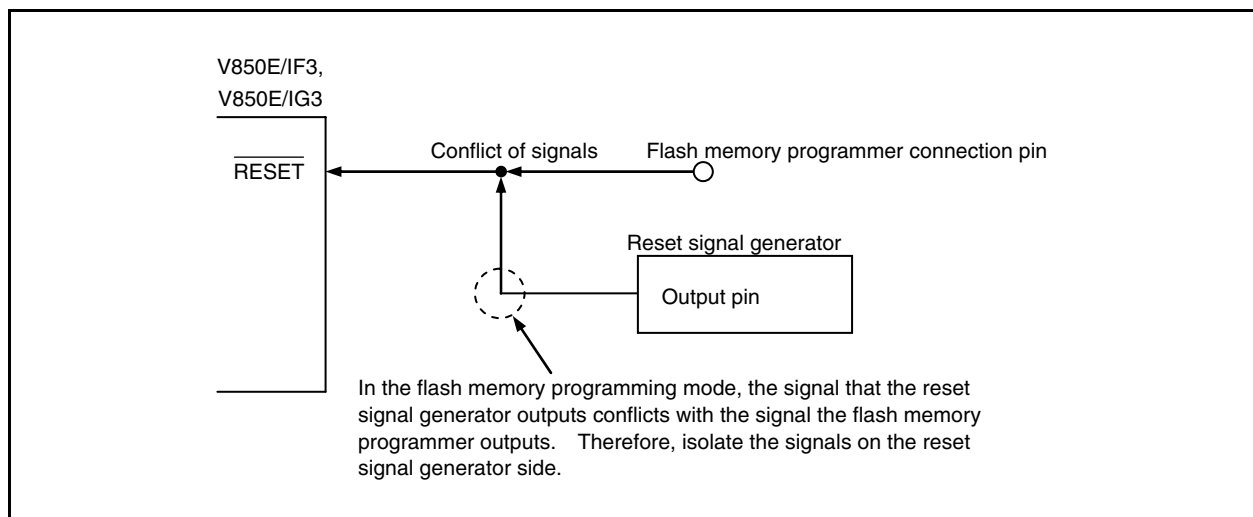
When the flash memory programmer (output or input) is connected to an interface pin (input or output) that is connected to another device (input), the signal is output to the other device, causing the device to malfunction. To avoid this, isolate the connection to the other device or make the setting so that the input signal to the other device is ignored.



### 27.7.3 $\overline{\text{RESET}}$ pin

When the reset signal of the flash memory programmer is 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 flash memory programmer.

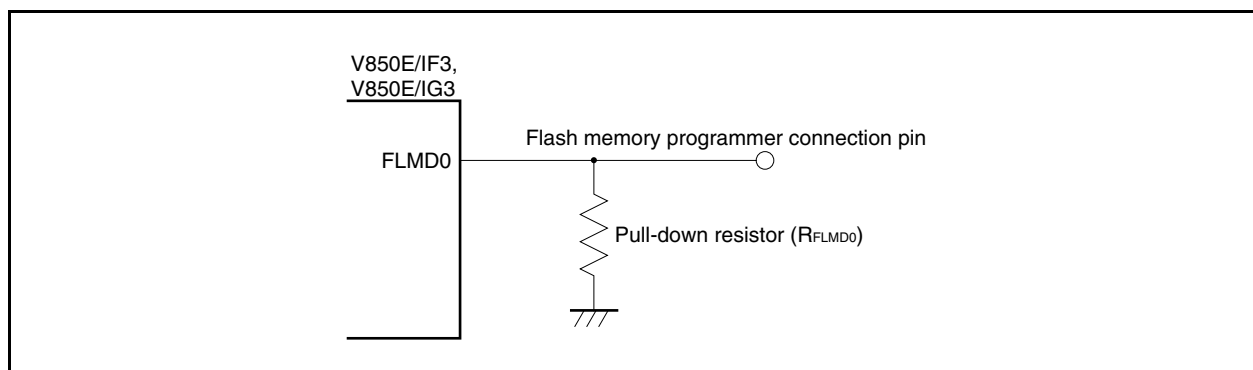


### 27.7.4 FLMD0 and FLMD1 pins

#### (1) FLMD0 pin

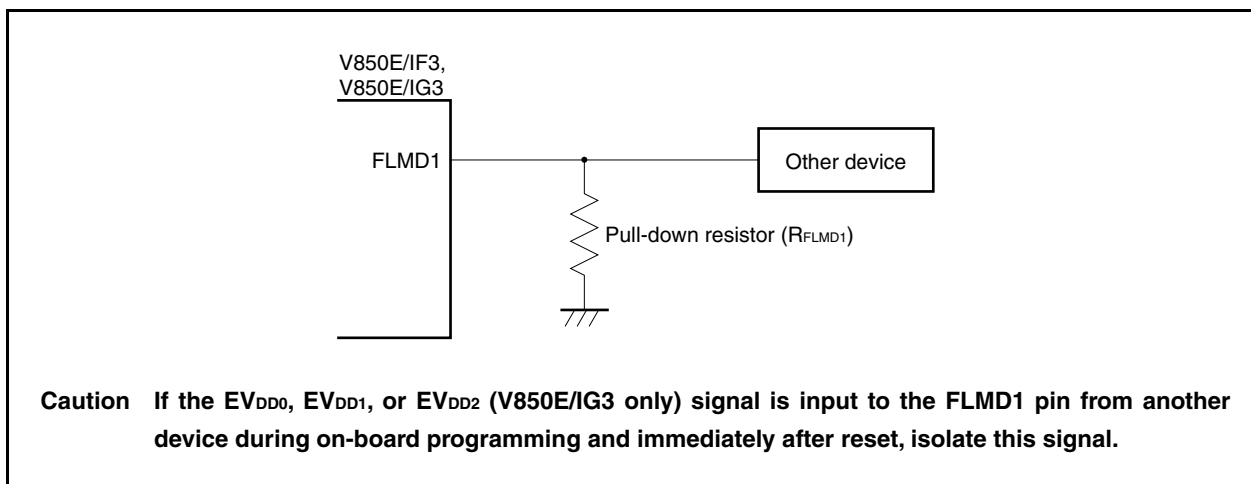
In the normal operation mode, input a voltage of  $\text{EV}_{\text{SS}0}$ ,  $\text{EV}_{\text{SS}1}$ , or  $\text{EV}_{\text{SS}2}$  (V850E/IG3) level to the FLMD0 pin. In the flash memory programming mode, supply a write voltage of  $\text{EV}_{\text{DD}0}$ ,  $\text{EV}_{\text{DD}1}$ ,  $\text{EV}_{\text{DD}2}$  (V850E/IG3 only) level to the FLMD0 pin.

Because the FLMD0 pin serves as a write protection pin in the self programming mode, a voltage of  $\text{EV}_{\text{DD}0}$ ,  $\text{EV}_{\text{DD}1}$ , or  $\text{EV}_{\text{DD}2}$  (V850E/IG3 only) level must be supplied to the FLMD0 pin via port control, etc., before writing to the flash memory. For details, see 27.9.5 (1) FLMD0 pin.



**(2) FLMD1 pin**

When 0 V is input to the FLMD0 pin, the FLMD1 pin does not function. When  $EV_{DD0}$ ,  $EV_{DD1}$ , or  $EV_{DD2}$  (V850E/IG3 only) 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.



**Table 27-8. Relationship Between FLMD0 and FLMD1 Pins and Operation Mode When Reset Is Released**

| FLMD0     | FLMD1      | Operation Mode                |
|-----------|------------|-------------------------------|
| 0         | Don't care | Normal operation mode         |
| $EV_{DD}$ | 0          | Flash memory programming mode |
| $EV_{DD}$ | $EV_{DD}$  | Setting prohibited            |

**Remark**  $EV_{DD}$ :  $EV_{DD0}$ ,  $EV_{DD1}$ ,  $EV_{DD2}$  (V850E/IG3 only)

**27.7.5 Port pins**

When the flash memory programming mode is set, all the port pins except the pin that communicates with the flash memory programmer change to the high-impedance status. These port pins need not be processed. If problems such as disabling of the high-impedance status should occur to the external devices connected to the ports, connect them to  $EV_{DD0}$ ,  $EV_{DD1}$ ,  $EV_{DD2}$  (V850E/IG3 only), or  $EV_{SS0}$ ,  $EV_{SS1}$ ,  $EV_{SS2}$  (V850E/IG3 only) via resistors.

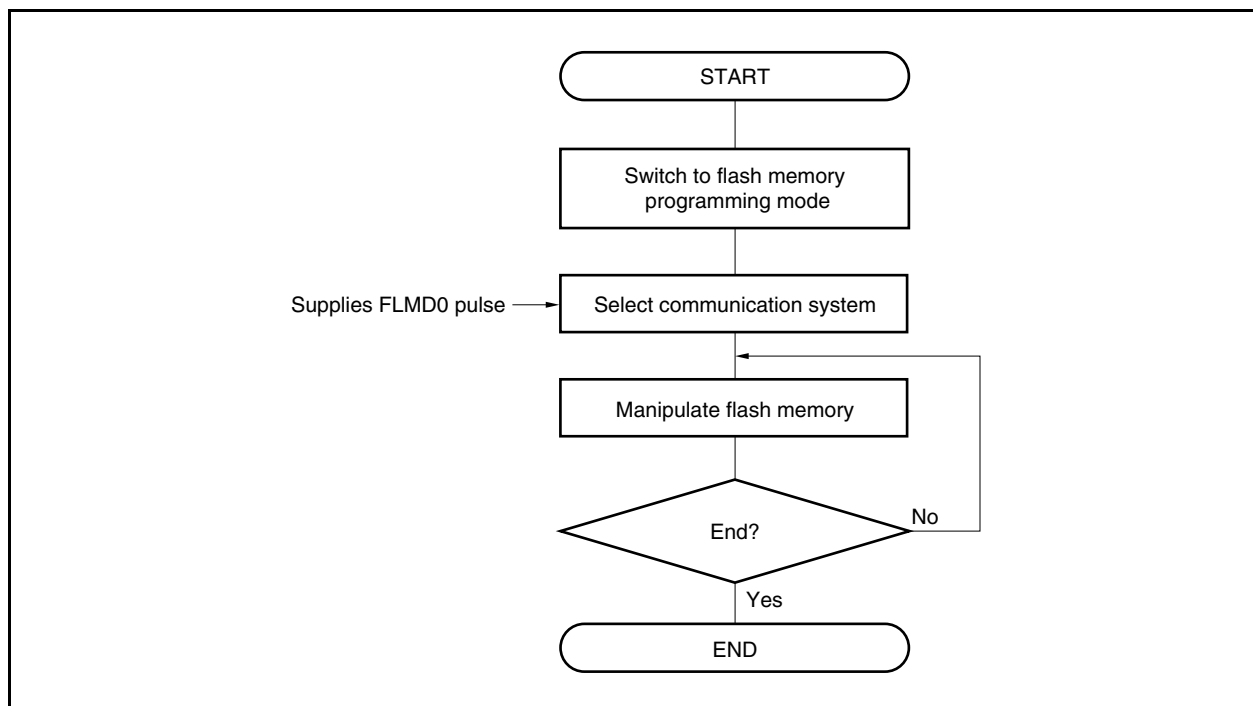
**27.7.6 Other signal pins**

Connect X1 and X2 in the same status as in the normal operation mode.

## 27.8 Flash Memory Programming Mode

### 27.8.1 Flash memory control

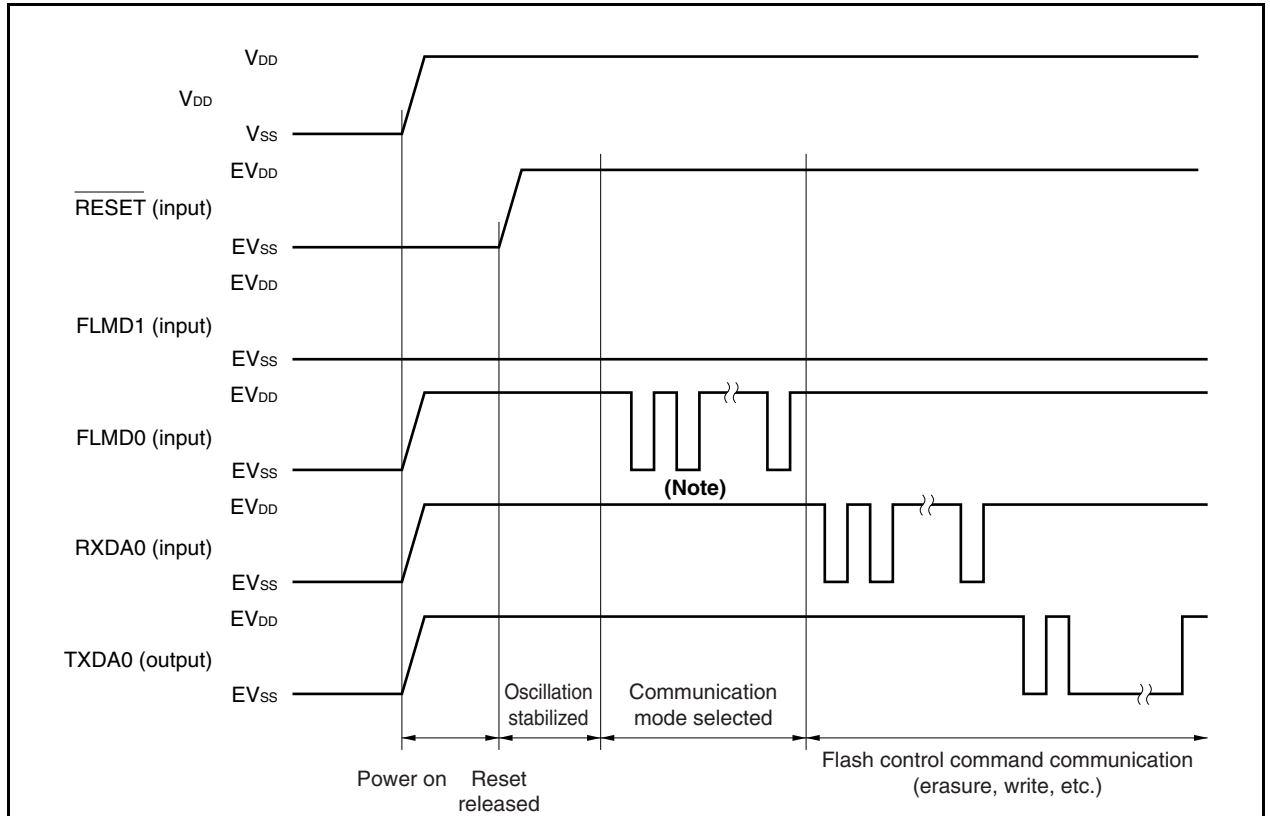
The following shows the procedure for manipulating the flash memory.



**27.8.2 Selection of communication mode**

In the V850E/IF3 and V850E/IG3, 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 flash memory programmer.

The following shows the relationship between the number of pulses and the communication mode.



**Note** The number of clocks is as follows depending on the communication mode.

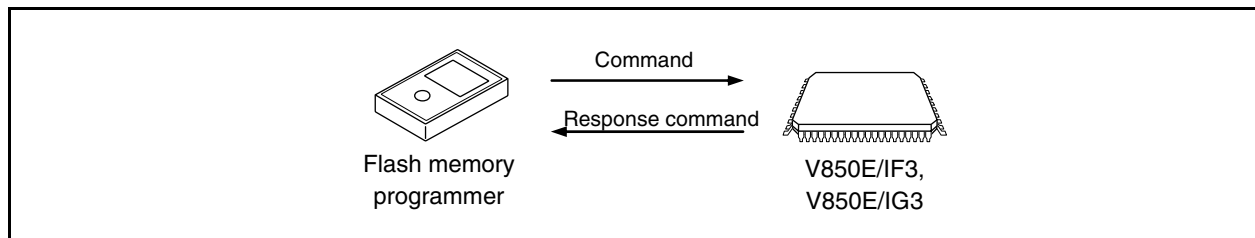
| FLMD0 pulse | Communication mode  | Remarks  |
|-------------|---------------------|--|
| 0           | UARTA0              | Communication rate: 9,600 bps (after reset), LSB first     |
| 8           | CSIB0               | V850E/IF3 and V850E/IG3 perform slave operation, MSB first |
| 11          | CSIB0 for handshake | V850E/IF3 and V850E/IG3 perform slave operation, MSB first |
| Other       | RFU                 | Setting prohibited   |

**Caution** When UARTA0 is selected, the receive clock is calculated based on the reset command sent from the flash memory programmer after receiving the FLMD0 pulse.

**Remark** V<sub>DD</sub>: V<sub>DD0</sub>, V<sub>DD1</sub>  
 EV<sub>DD</sub>: EV<sub>DD0</sub>, EV<sub>DD1</sub>, EV<sub>DD2</sub> (V850E/IG3 only)  
 EV<sub>SS</sub>: EV<sub>SS0</sub>, EV<sub>SS1</sub>, EV<sub>SS2</sub> (V850E/IG3 only)

**27.8.3 Communication commands**

The V850E/IF3 and V850E/IG3 communicate with a flash memory programmer by means of commands. The commands sent from the flash memory programmer to the V850E/IF3 or V850E/IG3 are called “commands”. The response signals sent from the V850E/IF3 or V850E/IG3 to the flash memory programmer are called “response commands”.



The following shows the commands for flash memory control in the V850E/IF3 and V850E/IG3. All of these commands are issued from the dedicated flash memory programmer, and the V850E/IF3 and V850E/IG3 perform the processing corresponding to the commands.

**Table 27-9. Flash Memory Control Commands**

| Classification             | Command Name                          | Support |       |      | Function  |
|----------------------------|---------------------------------------|---------|-------|------|---|
|                            |                                       | UARTA0  | CSIB0 | Note |   |
| Verify                     | Block verify command                  | √       | √     | √    | Compares the contents of the specified block and the input data |
| Erase                      | Chip erase command                    | √       | √     | √    | Erases the contents of the entire flash memory                  |
|                            | Block erase command                   | √       | √     | √    | Erases the specified block contents.                            |
| Blank check                | Block blank check command             | √       | √     | √    | Checks the erase state of the specified block.                  |
| Data write                 | Write command                         | √       | √     | √    | Writes data to the specified block.                             |
| Data read                  | Read command                          | √       | √     | √    | Reads out data of the specified block.                          |
| System setting and control | Status command                        | √       | ×     | √    | Obtains the status of operations.                               |
|                            | Oscillation frequency setting command | √       | √     | √    | Sets the oscillation frequency.                                 |
|                            | Baud rate setting command             | √       | ×     | ×    | Changes the baud rate when UART0 is selected.                   |
|                            | Silicon signature command             | √       | √     | √    | Reads out the silicon signature information.                    |
|                            | Version acquisition command           | √       | √     | √    | Reads out the device version and firmware version.              |
|                            | Security setting command              | √       | √     | √    | Sets the security information and the boot block size.          |
|                            | Checksum command                      | √       | √     | √    | Sends the checksum value of the specified block data.           |
|                            | Reset command                         | √       | √     | √    | Used for communication synchronization detecting.               |

**Note** CSIB0 for handshake

The V850E/IF3 and V850E/IG3 send back response commands for the commands issued from the flash memory programmer. The response commands sent from the V850E/IF3 and V850E/IG3 are listed below.

**Table 27-10. Response Commands**

| Response Command Name    | Function                                  |
|--------------------------|---|
| ACK (Acknowledge)        | Acknowledges command/data, etc.           |
| NAK (Not acknowledge)    | Acknowledges illegal frame, etc.          |
| Command number error     | Acknowledges illegal command/data, etc.   |
| Parameter error          | Acknowledges illegal parameter, etc.      |
| Checksum error           | Acknowledges checksum of frame            |
| Protect error            | Acknowledges when protection is in effect |
| During processing (BUSY) | Acknowledges during processing            |
| Other than above         | Error                                     |

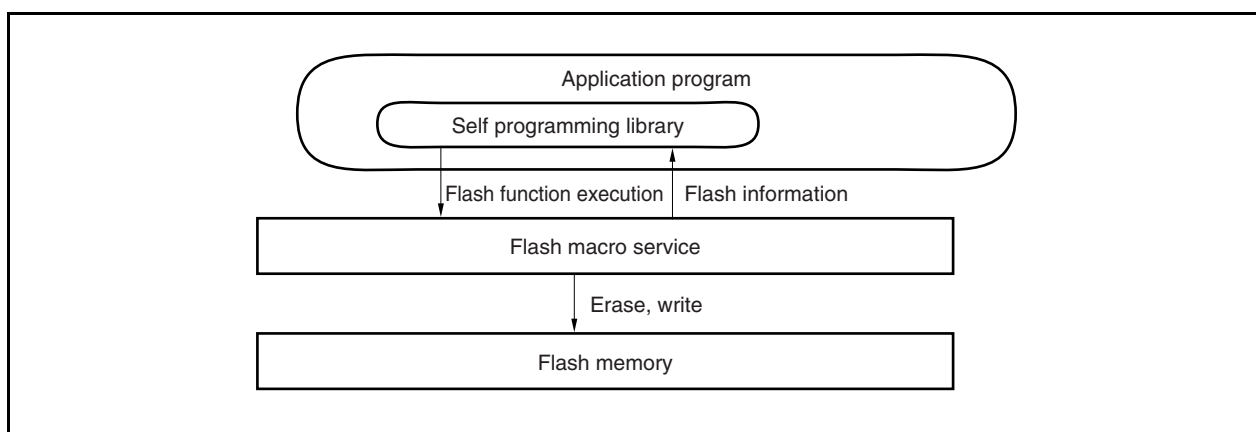
## <R> 27.9 Rewriting by Self Programming

### 27.9.1 Overview

The V850E/IF3 and V850E/IG3 support 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<sup>Note</sup> can be rewritten in the field. For details about self programming, see **Flash Memory Self Programming Library User's Manual**.

**Note** Be sure not to allocate the program code to the block where the constant data of rewriting target is allocated. See **27.2 Memory Configuration** for the block configuration.

**Figure 27-2. Concept of Self Programming**





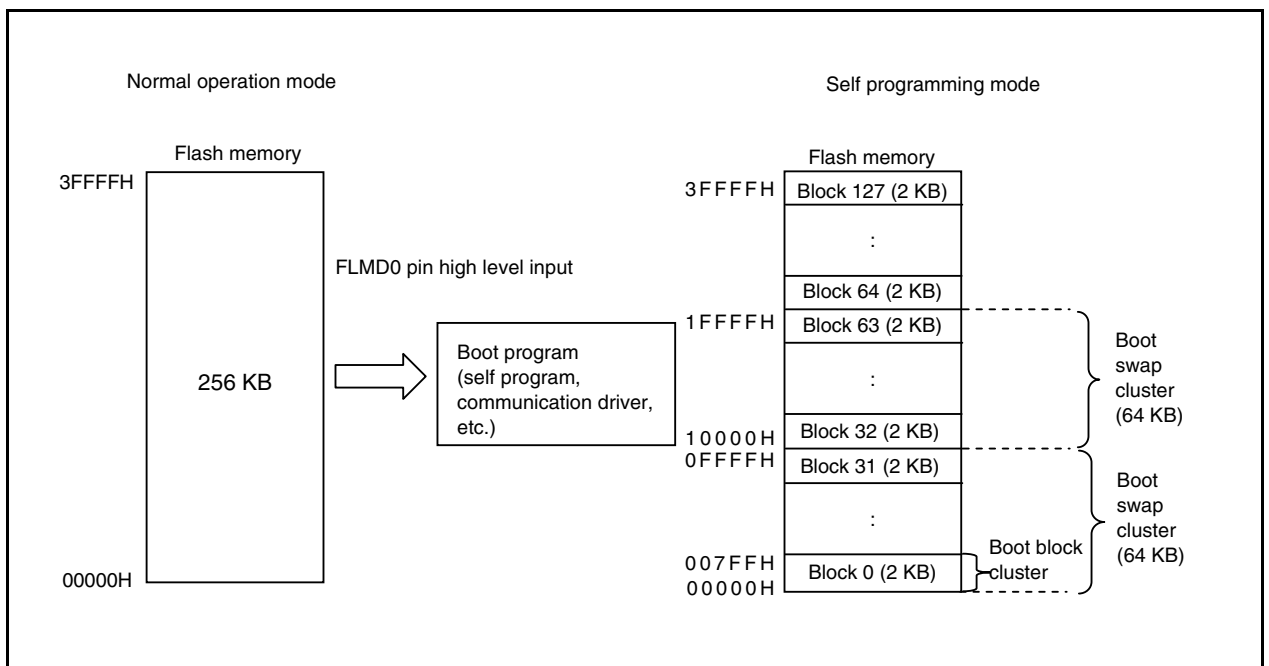
27.9.2 Features

(1) Flash memory self programming

Flash memory self programming is used to erase or write the flash memory by calling the flash function from a program stored in an area other than the flash memory area to be erased or written. To store the program that implements self programming in the area to be erased or written, copy the program to the internal RAM area, execute the program at the copy destination, and call the flash function.

To call the flash function, change the mode from the normal operation mode to the self programming mode by using the flash programming mode control register.

Figure 27-3. Self Programming



(a) Boot swap cluster

The contents of the boot swap cluster of the lower address side (00000H to 0FFFFH) and the boot swap cluster of the higher address side (10000H to 1FFFFH) can be interchanged while flash memory programming is performed.

(b) Boot block cluster

By specifying the boot block cluster from 00000H in 2 KB units, the contents of the boot block cluster can be protected from rewriting.

**(2) Interrupt support**

A user handler written to the flash memory cannot be used even if an interrupt occurred during flash function execution.

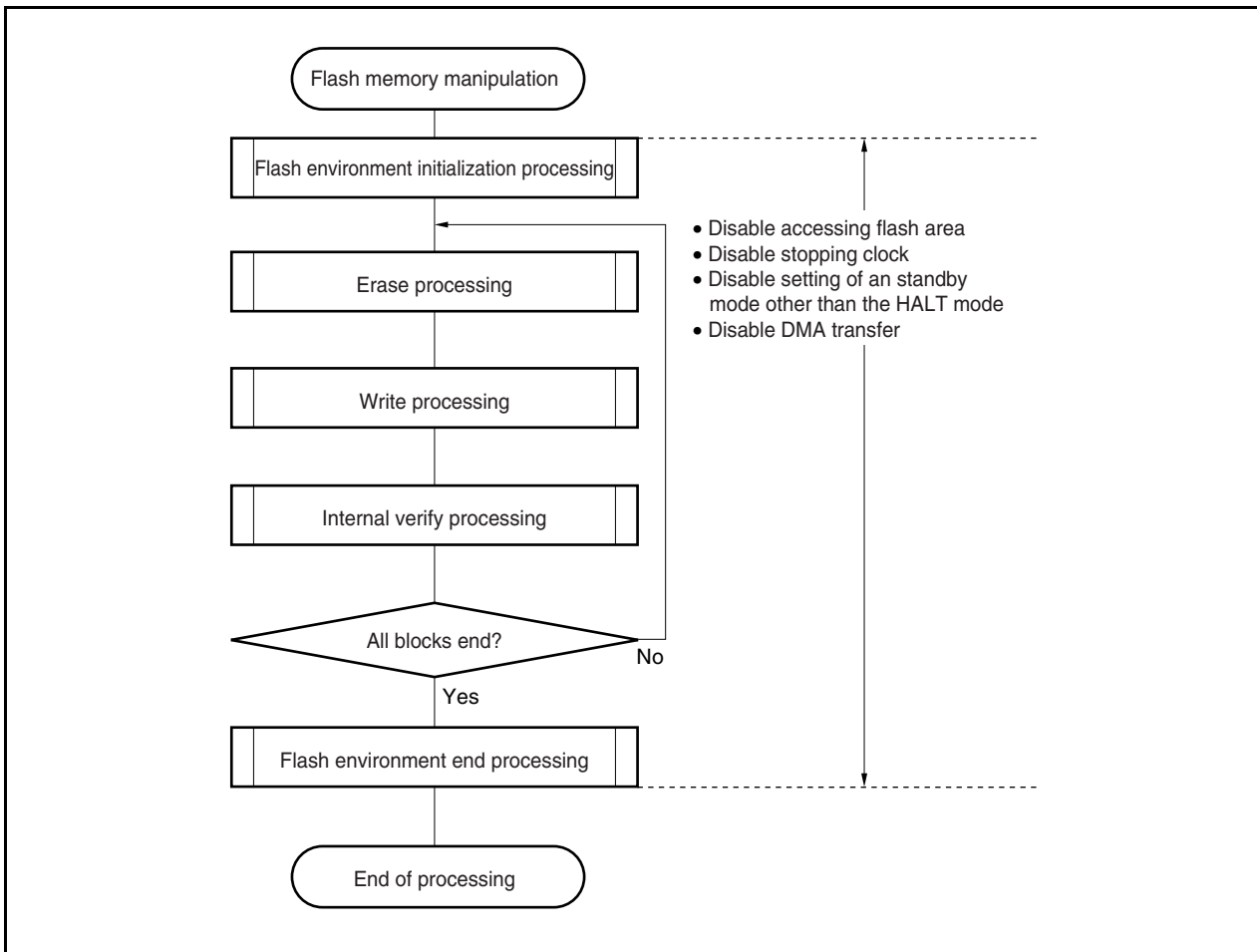
Therefore, in the V850E/IF3 and V850E/IG3, to use an interrupt during flash function execution, processing transits to the specific address<sup>Note</sup> in the internal RAM. Allocate the jump instruction that transits processing to the user interrupt servicing at the specific address<sup>Note</sup> in the internal RAM.

- Note** NMI interrupt: Start address of internal RAM  
 Maskable interrupt: Start address of internal RAM + 4 addresses

**27.9.3 Standard self programming flow**

The entire processing to rewrite the flash memory by flash self programming is illustrated below.

**Figure 27-4. Standard Self Programming Flow**



27.9.4 Flash functions

Table 27-11. Flash Function List

| Type               | Function Name        | Abbr. | Function  |
|--------------------|----------------------|-------|---|
| Initialize         | FlashEnv             | FLE   | Initializes flash control.  |
| Erase              | FlashBlockErase      | FLBE  | Erases the specified block.   |
| Write              | FlashWordWrite       | FLWW  | Successively writes the specified memory contents from the specified flash memory address, for the number of words specified in 4-byte units. |
| Check              | FlashBlockBlankCheck | FLBBC | Checks the erase status of the specified block.   |
|                    | FlashBlockIVerify    | FLBIV | Perform internal verify for the specified block.  |
|                    | FlashFLMDCheck       | FLFC  | Inputs FLMD0 pin and checks FLMD0 setting register value.   |
| Obtain information | FlashGetInfo         | FLGI  | Reads out information about the flash memory.   |
| Setting            | FlashSetInfo         | FLSI  | Sets the flash information.   |
|                    | FlashBootSwap        | FLBS  | Interchanges the contents of the boot swap cluster.   |
|                    | FlashWordRead        | FLWR  | Reads out the data from the specified address.  |

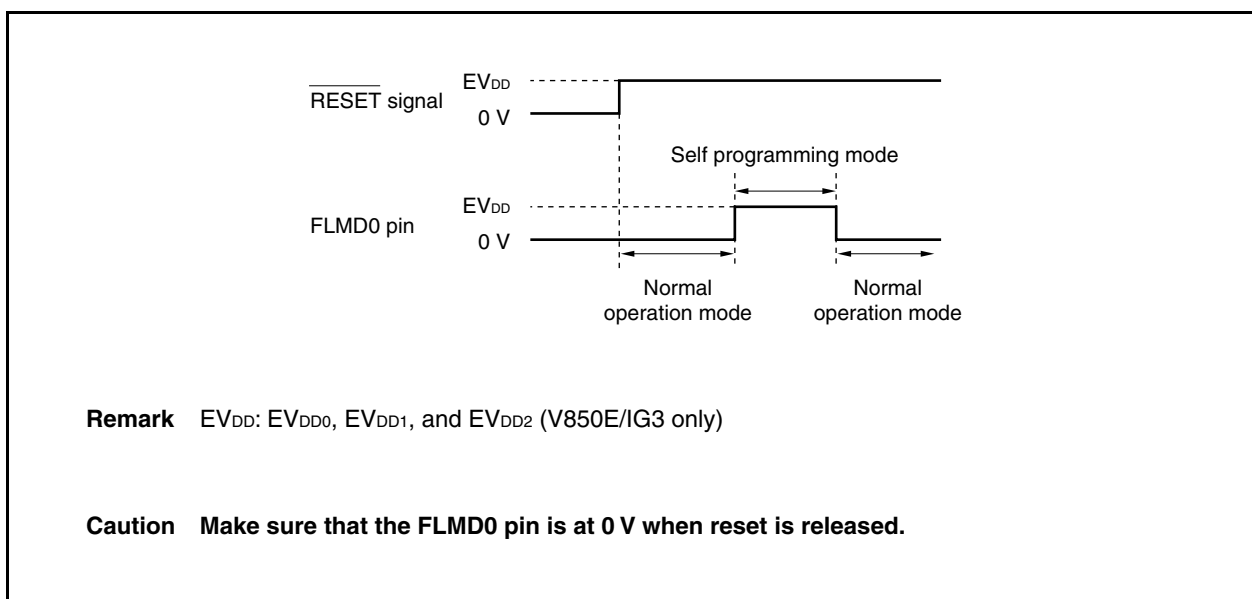
27.9.5 Pin processing

(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 EV<sub>DD0</sub>, EV<sub>DD1</sub>, and EV<sub>DD2</sub> (V850E/IG3 only) 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 27-5. Mode Change Timing



**27.9.6 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 27-12. Internal Resources Used**

| Resource Name                | Description  |
|------------------------------|--|
| Stack area                   | An extension of the stack used by the user is used by the library (can be used in both the internal RAM and external RAM).   |
| Library code <sup>Note</sup> | Program entity of library (can be used anywhere other than the flash memory block to be manipulated).  |
| Application program          | Executed as user application.<br>Calls flash functions.  |
| Maskable interrupt           | Can be used in the user application execution status or flash function execution status. To use this interrupt in the flash function execution 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 the user application execution status or flash function execution status. To use this interrupt in the flash function execution 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.  |

**Note** About resources used, refer to the **Flash Memory Self-Programming Library User's Manual**.

## CHAPTER 28 ELECTRICAL SPECIFICATIONS

### 28.1 V850E/IF3

#### 28.1.1 Absolute maximum ratings

(T<sub>A</sub> = 25°C)

| Parameter                          | Symbol            | Conditions  | Ratings  | Unit |    |
|------------------------------------|-------------------|---|--|------|----|
| Supply voltage                     | V <sub>DD</sub>   | V <sub>DDa</sub> = EV <sub>DDb</sub> = AV <sub>DDk</sub>  | -0.5 to +6.5                                     | V    |    |
|                                    | V <sub>SS</sub>   | V <sub>SSa</sub> = EV <sub>SSb</sub> = AV <sub>SSk</sub>  | -0.5 to +0.5                                     | V    |    |
|                                    | EV <sub>DD</sub>  | V <sub>DDa</sub> = EV <sub>DDb</sub> = AV <sub>DDk</sub>  | -0.5 to +6.5                                     | V    |    |
|                                    | EV <sub>SS</sub>  | V <sub>SSa</sub> = EV <sub>SSb</sub> = AV <sub>SSk</sub>  | -0.5 to +0.5                                     | V    |    |
|                                    | AV <sub>DD</sub>  | V <sub>DDa</sub> = EV <sub>DDb</sub> = AV <sub>DDk</sub>  | -0.5 to +6.5                                     | V    |    |
|                                    | AV <sub>SS</sub>  | V <sub>SSa</sub> = EV <sub>SSb</sub> = AV <sub>SSk</sub>  | -0.5 to +0.5                                     | V    |    |
| Input voltage                      | V <sub>I1</sub>   | <b>Note 1</b>   | -0.5 to EV <sub>DD</sub> + 0.5 <sup>Note 2</sup> | V    |    |
|                                    | V <sub>I2</sub>   | X1, X2  | -0.5 to V <sub>RO</sub> + 0.35                   | V    |    |
| Output current, low                | I <sub>OL</sub>   | All pins  | Per pin  | 4    | mA |
|                                    |                   |   | Total of all pins                                | 42   | mA |
| Output current, high               | I <sub>OH</sub>   | All pins  | Per pin  | -4   | mA |
|                                    |                   |   | Total of all pins                                | -42  | mA |
| Analog input voltage               | V <sub>IAN</sub>  | P70/ANI20 to P73/ANI23,<br>ANI00 to ANI05, ANI10 to ANI17 | -0.5 to AV <sub>DD</sub> + 0.5 <sup>Note 2</sup> | V    |    |
| Analog reference input voltage     | V <sub>IREF</sub> | AV <sub>REFP0</sub> , AV <sub>REFP1</sub>                 | -0.5 to AV <sub>DD</sub> + 0.5 <sup>Note 2</sup> | V    |    |
| Comparator reference input voltage | V <sub>CREF</sub> | CREF0L, CREF1L, CREF0F, CREF1F                            | -0.5 to AV <sub>DD</sub> + 0.5 <sup>Note 2</sup> | V    |    |
| Operating ambient temperature      | T <sub>A</sub>    | In normal operating mode                                  | -40 to +85                                       | °C   |    |
|                                    |                   | In flash memory programming mode                          | -40 to +85                                       | °C   |    |
| Operating ambient temperature      | T <sub>stg</sub>  |   | -40 to +125                                      | °C   |    |

**Notes** 1. P00, P01, P10 to P17, P20 to P27, P30 to P37, P40 to P47, PDL0 to PDL9,  $\overline{\text{RESET}}$ , FLMD0

2. Be sure not to exceed the absolute maximum ratings (MAX. value) of each supply voltage.

**Cautions** 1. Do not directly connect the output pins (or I/O pins in the output state) of IC products to other output pins (including I/O pins in the output state), power supply pins such as V<sub>DD</sub> and EV<sub>DD</sub>, or GND pin. Direct connection of the output pins between an IC product and an external circuit is possible, if the output pins can be set to the high-impedance state and the output timing of the external circuit is designed to avoid output conflict.

2. Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

The ratings and conditions indicated for DC characteristics and AC characteristics represent the quality assurance range during normal operation.

**Remark** a = 0, 1  
b = 0, 1  
k = 0 to 2

**28.1.2 Capacitance**

( $T_A = 25^\circ\text{C}$ ,  $V_{DD0} = V_{SS0} = V_{DD1} = V_{SS1} = EV_{DD0} = EV_{SS0} = EV_{DD1} = EV_{SS1} = AV_{DD0} = AV_{SS0} = AV_{DD1} = AV_{SS1} = AV_{DD2} = AV_{SS2} = 0\text{ V}$ )

| Parameter         | Symbol   | Conditions                      | MIN. | TYP. | MAX. | Unit |
|-------------------|----------|---------------------------------|------|------|------|------|
| Input capacitance | $C_i$    | $f_c = 1\text{ MHz}$            |      |      | 15   | pF   |
| I/O capacitance   | $C_{iO}$ | Unmeasured pins returned to 0 V |      |      | 15   | pF   |

- Notes**
1. ANI00 to ANI05, ANI10 to ANI17,  $\overline{\text{RESET}}$
  2. P00, P01, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P70 to P73, PDL0 to PDL9

- Cautions**
1. Excludes the FLMD0, X1, and X2 pins.
  2. In addition to input capacitance, sampling capacitance is added to the ANI00 to ANI05, ANI10 to ANI17, and ANI20 to ANI27 pins when sampling.

**28.1.3 Operating conditions**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0\text{ V}$ )

| Parameter                    | Symbol                  | Conditions  | MIN. | TYP. | MAX. | Unit |
|------------------------------|-------------------------|---|------|------|------|------|
| System clock frequency       | $f_{xx}$                | PLL mode  | 32   |      | 64   | MHz  |
|                              |                         | Clock through mode  | 4    |      | 8    | MHz  |
| CPU clock frequency          | $f_{CPU}$               | PLL mode  | 4    |      | 64   | MHz  |
|                              |                         | Clock through mode  | 0.5  |      | 8    | MHz  |
| $V_{DD}$ , $EV_{DD}$ voltage | $V_{DD}$ ,<br>$EV_{DD}$ | $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} =$<br>$AV_{DD0} = AV_{DD1} = AV_{DD2}$ | 3.5  |      | 5.5  | V    |
| $AV_{DD}$ voltage            | $AV_{DD}$               | When A/D converters 0 to 2 are operating  | 4.0  |      | 5.5  | V    |
|                              |                         | When A/D converters 0 to 2 are not operating                                    | 3.5  |      | 5.5  | V    |

28.1.4 Clock oscillator characteristics

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5$  to  $5.5$  V,  $V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V)

| Resonator                  | Recommended Circuit | Parameter                       | Conditions              | MIN. | TYP. | MAX.         | Unit |    |
|----------------------------|---------------------|---------------------------------|-------------------------|------|------|--------------|------|----|
| Ceramic /crystal resonator |                     | Oscillation frequency ( $f_x$ ) |                         | 4    |      | 8            | MHz  |    |
|                            |                     | Oscillation stabilization time  | After reset release     |      |      | $2^{14}/f_x$ |      | ms |
|                            |                     |                                 | After STOP mode release |      |      | <b>Note</b>  |      |    |

**Note** The value varies depending on the setting of the oscillation stabilization time select register (OSTS).

- Cautions**
1. Connect the oscillator as close to the X1 and X2 pins as possible.
  2. Do not cross the wiring with the other signal lines in the area enclosed by the broken lines in the above figure.
  3. For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.
  4. Inputting an external clock to the V850E/IF3 is prohibited.

(i) Murata Mfg. Co., Ltd.: Ceramic resonator ( $T_A = -40$  to  $+85^\circ\text{C}$ )

| Type             | Part Number     | Oscillation Frequency $f_x$ (MHz) | Recommended Circuit Constant |                 |                  | Oscillation Voltage Range |          |
|------------------|-----------------|-----------------------------------|------------------------------|-----------------|------------------|---------------------------|----------|
|                  |                 |                                   | C1                           | C2              | Rd (k $\Omega$ ) | MIN. (V)                  | MAX. (V) |
| Lead             | CSTLS4M00G56-B0 | 4                                 | On chip (47 pF)              | On chip (47 pF) | 0                | 3.5                       | 5.5      |
|                  | CSTLS5M00G56-B0 | 5                                 | On chip (47 pF)              | On chip (47 pF) | 0                | 3.5                       | 5.5      |
|                  | CSTLS8M00G56-B0 | 8                                 | On chip (47 pF)              | On chip (47 pF) | 0                | 3.5                       | 5.5      |
| Surface mounting | CSTCR4M00G55-R0 | 4                                 | On chip (39 pF)              | On chip (39 pF) | 0                | 3.5                       | 5.5      |
|                  | CSTCR5M00G55-R0 | 5                                 | On chip (39 pF)              | On chip (39 pF) | 0                | 3.5                       | 5.5      |
|                  | CSTCE8M00G55-R0 | 8                                 | On chip (33 pF)              | On chip (33 pF) | 0                | 3.5                       | 5.5      |

**Caution** These oscillator constants are reference values based on evaluation under a specific environment by the resonator manufacturer. When optimization of the oscillator characteristics on the actual application is necessary, request evaluation on the mounting circuit from the resonator manufacturer.

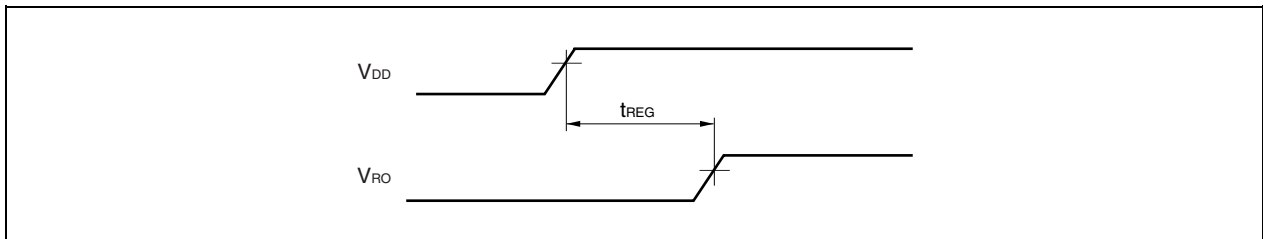
The oscillation voltage and oscillation frequency indicate only oscillator characteristics, therefore use the V850E/IF3 within the DC characteristics and AC characteristics for internal operation conditions.

28.1.5 Regulator characteristics

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = 3.5$  to  $5.5$  V,  $V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = 0$  V)

| Parameter                         | Symbol    | Conditions   | MIN. | TYP. | MAX.                | Unit |
|-----------------------------------|-----------|--|------|------|---------------------|------|
| Input voltage                     | $V_{DD}$  |  | 3.5  |      | 5.5                 | V    |
| Output voltage                    | $V_{RO}$  |  |      | 1.5  |                     | V    |
| Output voltage stabilization time | $t_{REG}$ | Stabilizing capacitor: $C = 4.7 \mu\text{F}^{\text{Note 1}}$ |      |      | $1^{\text{Note 2}}$ | ms   |

- Notes**
1. Connect a stabilizing capacitor between the REGC0 pin and  $V_{SS0}$  pin, and between the REGC1 pin and  $V_{SS1}$  pin.
  2. Internal reset signal is output until the power-on-clear circuit (POC) output voltage stabilizes during  $t_{REG}$  period.





**28.1.6 DC characteristics**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5$  to  $5.5$  V,  
 $V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V) (1/2)

| Parameter                    | Symbol     | Conditions            |                    | MIN.            | TYP. | MAX.         | Unit             |
|------------------------------|------------|-----------------------|--------------------|-----------------|------|--------------|------------------|
| Input voltage, high          | $V_{IH1}$  | <b>Note 1</b>         |                    | $0.7EV_{DD}$    |      | $EV_{DD}$    | V                |
|                              | $V_{IH2}$  | <b>Note 2</b>         |                    | $0.8EV_{DD}$    |      | $EV_{DD}$    | V                |
|                              | $V_{IH3}$  | <b>Note 3</b>         |                    | $0.7AV_{DD}$    |      | $AV_{DD}$    | V                |
| Input voltage, low           | $V_{IL1}$  | <b>Note 1</b>         |                    | $EV_{SS}$       |      | $0.3EV_{DD}$ | V                |
|                              | $V_{IL2}$  | <b>Note 2</b>         |                    | $EV_{SS}$       |      | $0.2EV_{DD}$ | V                |
|                              | $V_{IL3}$  | <b>Note 3</b>         |                    | $AV_{SS}$       |      | $0.3AV_{DD}$ | V                |
| Input leakage current, high  | $I_{LIH1}$ | $V_i = \text{Note 4}$ | Other than X1      |                 |      | 5            | $\mu\text{A}$    |
|                              | $I_{LIH2}$ |                       | X1                 |                 |      | 20           | $\mu\text{A}$    |
| Input leakage current, low   | $I_{LIL1}$ | $V_i = 0$ V           | Other than X1      |                 |      | -5           | $\mu\text{A}$    |
|                              | $I_{LIL2}$ |                       | X1                 |                 |      | -20          | $\mu\text{A}$    |
| Output leakage current, high | $I_{LOH}$  | $V_o = \text{Note 4}$ |                    |                 |      | 5            | $\mu\text{A}$    |
| Output leakage current, low  | $I_{LOL}$  | $V_o = 0$ V           |                    |                 |      | -5           | $\mu\text{A}$    |
| Output voltage, high         | $V_{OH1}$  | <b>Note 5</b>         | $I_{OH} = -1.0$ mA | $EV_{DD} - 1.0$ |      |              | V                |
| Output voltage, low          | $V_{OL1}$  | <b>Note 5</b>         | $I_{OL} = 1.0$ mA  |                 |      | 0.4          | V                |
| Pull to up resistor          | $R_{L1}$   |                       |                    | 10              | 30   | 100          | $\text{k}\Omega$ |

**Notes** 1. P33, P36, P41, and PDL0 to PDL9 pins

2. P00, P01, P10 to P17, P20 to P27, P30 to P32, P34, P35, P37, P40, P42 to P47,  $\overline{\text{RESET}}$ , and FLMD0 pins

3. P70 to P73 pins

4.  $AV_{DD0} = AV_{DD1} = AV_{DD2} = EV_{DD0} = EV_{DD1}$

5. P00, P01, P10 to P17, P20 to P27, P30 to P37, P40 to P47, and PDL0 to PDL9 pins

**Remark** The characteristics of alternate-function pins are the same as those of port pins.

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5$  to  $5.5$  V,  
 $V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V) (2/2)

| Parameter  | Symbol           | Conditions               | MIN.             | TYP. <sup>Note 1</sup> | MAX. | Unit |    |
|--|------------------|--------------------------|------------------|------------------------|------|------|----|
| V <sub>DD</sub> supply current <sup>Note 2</sup> | I <sub>DD1</sub> | f <sub>XX</sub> = 64 MHz | Normal operation |                        | 64   | 93   | mA |
|  | I <sub>DD2</sub> |                          | HALT mode        |                        | 42   | 60   | mA |
|  | I <sub>DD3</sub> |                          | IDLE mode        |                        | 5    | 10   | mA |
|  | I <sub>DD4</sub> | STOP mode                |                  | 40                     | 800  | μA   |    |

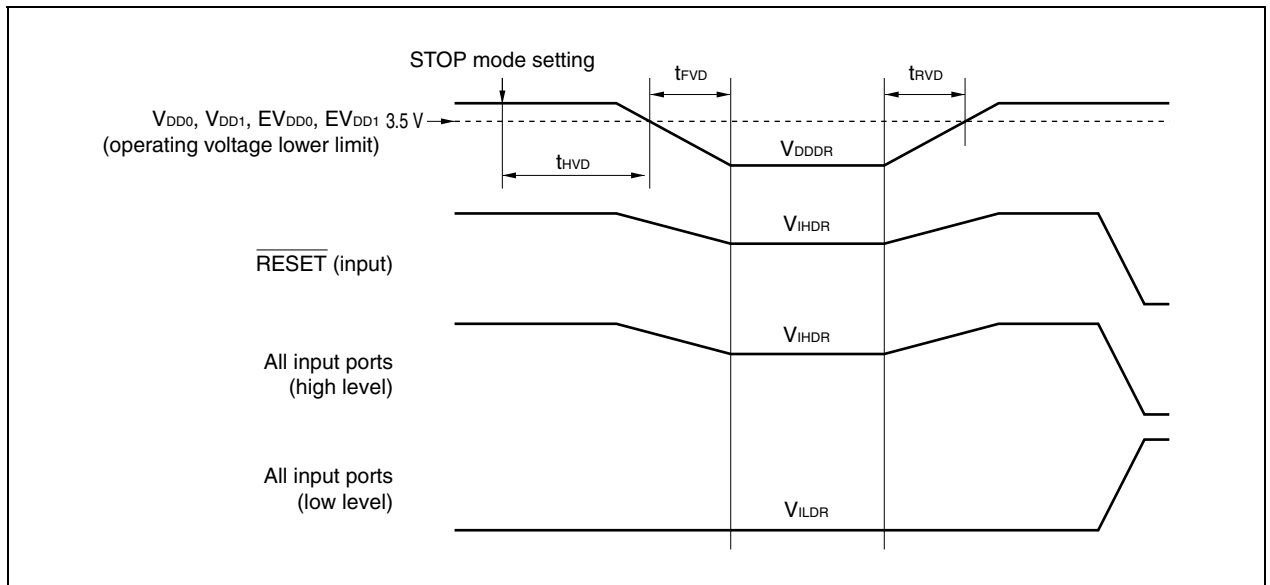
- Notes**
1. The TYP. value is a reference value when  $V_{DD0} = V_{DD1} = 5.0$  V and  $T_A = 25^\circ\text{C}$ .
  2. The current consumed by the EV<sub>DD</sub> system (output buffer and pull-up resistor) and the operating currents of A/D converters 0 to 2, the operational amplifier, and the comparator are not included.

28.1.7 Data retention characteristics

**STOP mode ( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V)**

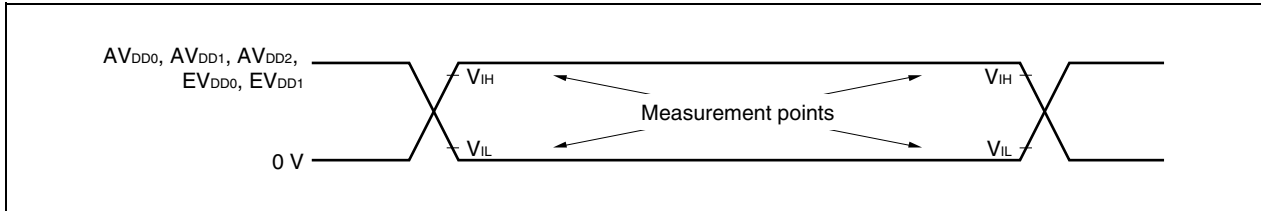
| Parameter  | Symbol     | Conditions                     | MIN.          | TYP. | MAX.          | Unit          |
|--|------------|--------------------------------|---------------|------|---------------|---------------|
| Data retention voltage                               | $V_{DDDR}$ | STOP mode                      | <b>Note</b>   |      | 5.5           | V             |
| Data retention current                               | $I_{DDDR}$ | $V_{DD0} = V_{DD1} = V_{DDDR}$ |               | 40   | 800           | $\mu\text{A}$ |
| Supply voltage rise time                             | $t_{rVD}$  |                                | 1             |      |               | $\mu\text{s}$ |
| Supply voltage fall time                             | $t_{fVD}$  |                                | 1             |      |               | $\mu\text{s}$ |
| Supply voltage hold time<br>(from STOP mode setting) | $t_{hVD}$  |                                | 0             |      |               | ms            |
| 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                | $EV_{SS}$     |      | $0.1V_{DDDR}$ | V             |

**Note** When the low-voltage detector (LVI) reset mode is not used (LVIM.LVIMD bit = 0): POC detection voltage ( $V_{POC0}$ )  
 When the low-voltage detector (LVI) reset mode is used (LVIM.LVIMD bit = 1): LVI detection voltage ( $V_{LV10}/V_{LV11}$ )

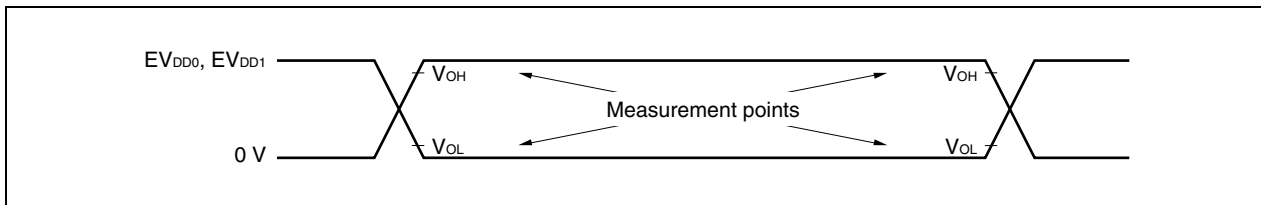


28.1.8 AC characteristics

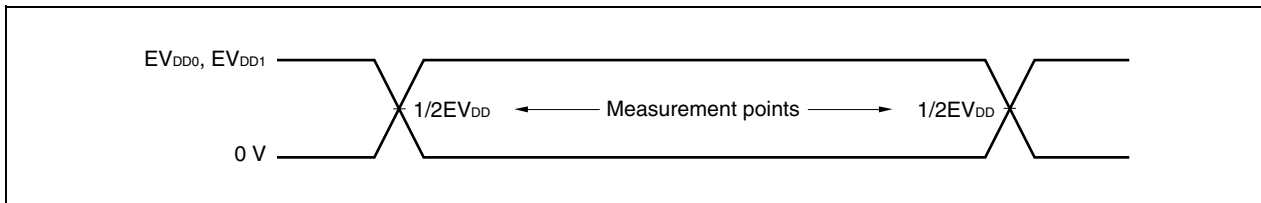
AC Test Input Measurement Points (Pins Other than CSIB0 to CSIB2)



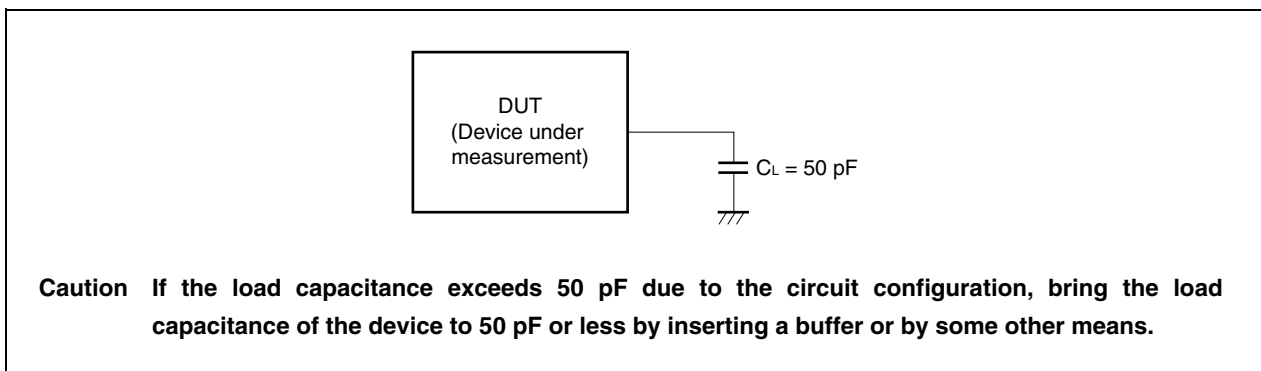
AC Test Output Measurement Points (Pins Other than CSIB0 to CSIB2)



AC Test I/O Measurement Points (CSIB0 to CSIB2 Pins)

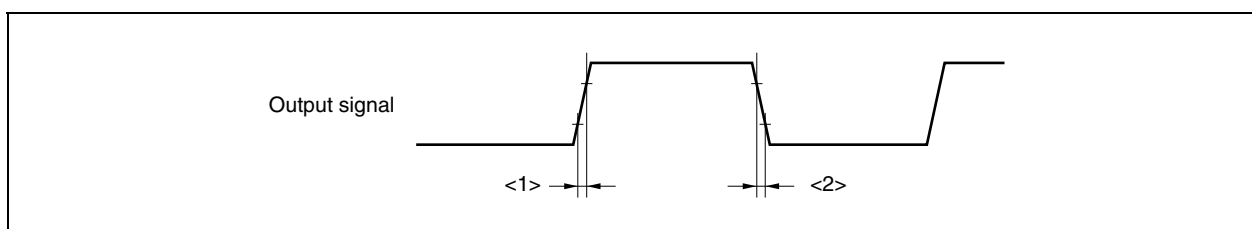


Load Conditions



**(1) Output signal timing**
 $(T_A = -40 \text{ to } +85^\circ\text{C}, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 \text{ to } 5.5 \text{ V},$ 
 $V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 \text{ V}, C_L = 50 \text{ pF})$ 

| Parameter        | Symbol   | Conditions       | MIN. | MAX. | Unit |
|------------------|----------|------------------|------|------|------|
| Output rise time | $t_{OR}$ | <1> PDL0 to PDL9 |      | 8    | ns   |
|                  |          | Other than above |      | 15   | ns   |
| Output fall time | $t_{OF}$ | <2> PDL0 to PDL9 |      | 8    | ns   |
|                  |          | Other than above |      | 15   | ns   |


**(2) Reset, external interrupt timing**
 $(T_A = -40 \text{ to } +85^\circ\text{C}, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 \text{ to } 5.5 \text{ V},$ 
 $V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 \text{ V}, C_L = 50 \text{ pF})$ 

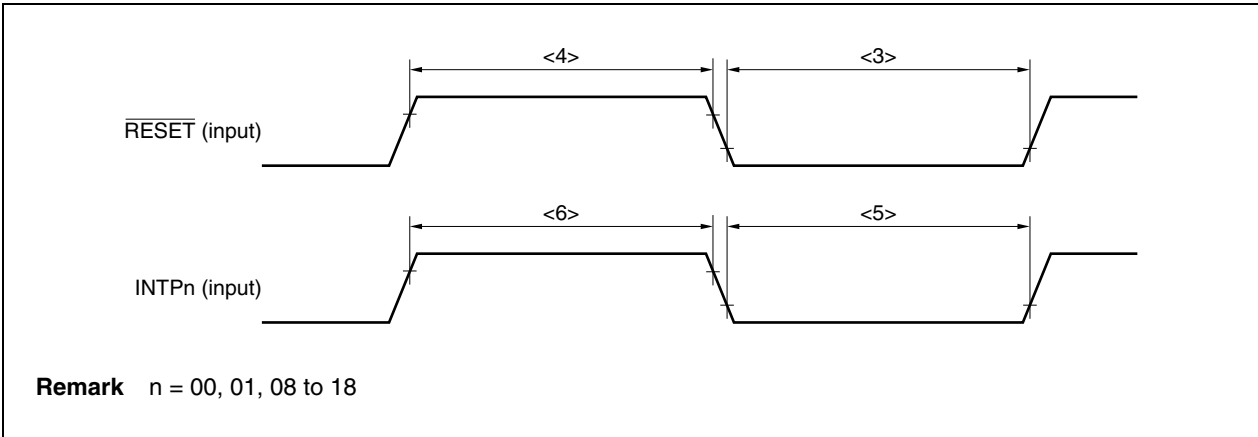
| Parameter              | Symbol     | Conditions   | MIN.           | MAX. | Unit |
|------------------------|------------|--|----------------|------|------|
| RESET low-level width  | $t_{WRSL}$ | <3> Power is on, STOP mode is released                         | $500 + T_{OS}$ |      | ns   |
|                        |            | Other than above   | 500            |      | ns   |
| RESET high-level width | $t_{WRSH}$ | <4>  | 500            |      | ns   |
| INTPn low-level width  | $t_{WITL}$ | <5> n = 00, 01, 08 to 13, 17, 18<br>(analog noise elimination) | 500            |      | ns   |
|                        |            | n = 14 to 16 (digital noise elimination)                       | $4T_{SMP}$     |      | ns   |
| INTPn high-level width | $t_{WITH}$ | <6> n = 00, 01, 08 to 13, 17, 18<br>(analog noise elimination) | 500            |      | ns   |
|                        |            | n = 14 to 16 (digital noise elimination)                       | $4T_{SMP}$     |      | ns   |

**Remarks 1.**  $T_{OS}$ : Oscillation stabilization time

$T_{SMP}$ : Noise elimination sampling clock cycle (set by INTNFCn register)

- After reset release, a 1 ms oscillation stabilization time is internally secured when the oscillation frequency ( $f_x$ ) = 8 MHz. The oscillation stabilization time is therefore  $(T_{OS} + 1)$  ms. After STOP mode release, an oscillation stabilization time half the value set to the OSTs register is internally secured. Therefore,  $T_{OS} = 0$  ns is acceptable if sufficient stabilization time can be secured by the OSTs register setting.

**Reset/Interrupt**



(3) Timer timing

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0$  to  $5.5$  V,

$V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V,  $C_L = 50$  pF)

| Parameter   | Symbol              | Conditions                     | MIN.                    | MAX. | Unit |
|---|---------------------|--------------------------------|-------------------------|------|------|
| T <sub>in</sub> high-level width <sup>Notes 1, 2</sup>  | t <sub>WTIHn</sub>  | <7> n = B00 to B03, B10 to B13 | 12T + 10                |      | ns   |
|   |                     | n = A20, A21, A40, A41         | 3T <sub>smp1</sub> + 10 |      | ns   |
| T <sub>in</sub> low-level width <sup>Notes 1, 2</sup>   | t <sub>WTILn</sub>  | n = B00 to B03, B10 to B13     | 12T + 10                |      | ns   |
|   |                     | n = A20, A21, A40, A41         | 3T <sub>smp1</sub> + 10 |      | ns   |
| EVTBm high-level width <sup>Note 1</sup>                | t <sub>WEVBHm</sub> | <9> m = 0, 1                   | 12T + 10                |      | ns   |
| EVTBm low-level width <sup>Note 1</sup>                 | t <sub>WEVBLm</sub> | <10> m = 0, 1                  | 12T + 10                |      | ns   |
| TRGBm high-level width <sup>Note 1</sup>                | t <sub>WTRHm</sub>  | <11> m = 0, 1                  | 12T + 10                |      | ns   |
| TRGBm low-level width <sup>Note 1</sup>                 | t <sub>WTRLm</sub>  | <12> m = 0, 1                  | 12T + 10                |      | ns   |
| TENC10/TENC11 high-level width <sup>Note 3</sup>        | t <sub>WENCH1</sub> | <13>                           | 3T <sub>smp2</sub> + 10 |      | ns   |
| TENC10/TENC11 low-level width <sup>Note 3</sup>         | t <sub>WENCL1</sub> | <14>                           | 3T <sub>smp2</sub> + 10 |      | ns   |
| TECR1 high-level width <sup>Note 3</sup>                | t <sub>WCRH1</sub>  | <15>                           | 3T <sub>smp2</sub> + 10 |      | ns   |
| TECR1 low-level width <sup>Note 3</sup>                 | t <sub>WCRL1</sub>  | <16>                           | 3T <sub>smp2</sub> + 10 |      | ns   |
| TIT10/TIT11 high-level width <sup>Note 3</sup>          | t <sub>WTITH1</sub> | <17>                           | 3T <sub>smp2</sub> + 10 |      | ns   |
| TIT10/TIT11 low-level width <sup>Note 3</sup>           | t <sub>WTITL1</sub> | <18>                           | 3T <sub>smp2</sub> + 10 |      | ns   |
| EVTT1 high-level width <sup>Note 3</sup>                | t <sub>WEVTH1</sub> | <19>                           | 3T <sub>smp2</sub> + 10 |      | ns   |
| EVTT1 low-level width <sup>Note 3</sup>                 | t <sub>WEVTL1</sub> | <20>                           | 3T <sub>smp2</sub> + 10 |      | ns   |
| TENC10/TENC11 input time differential <sup>Note 3</sup> | t <sub>PHUD1</sub>  | <21>                           | 3T <sub>smp2</sub> + 10 |      | ns   |

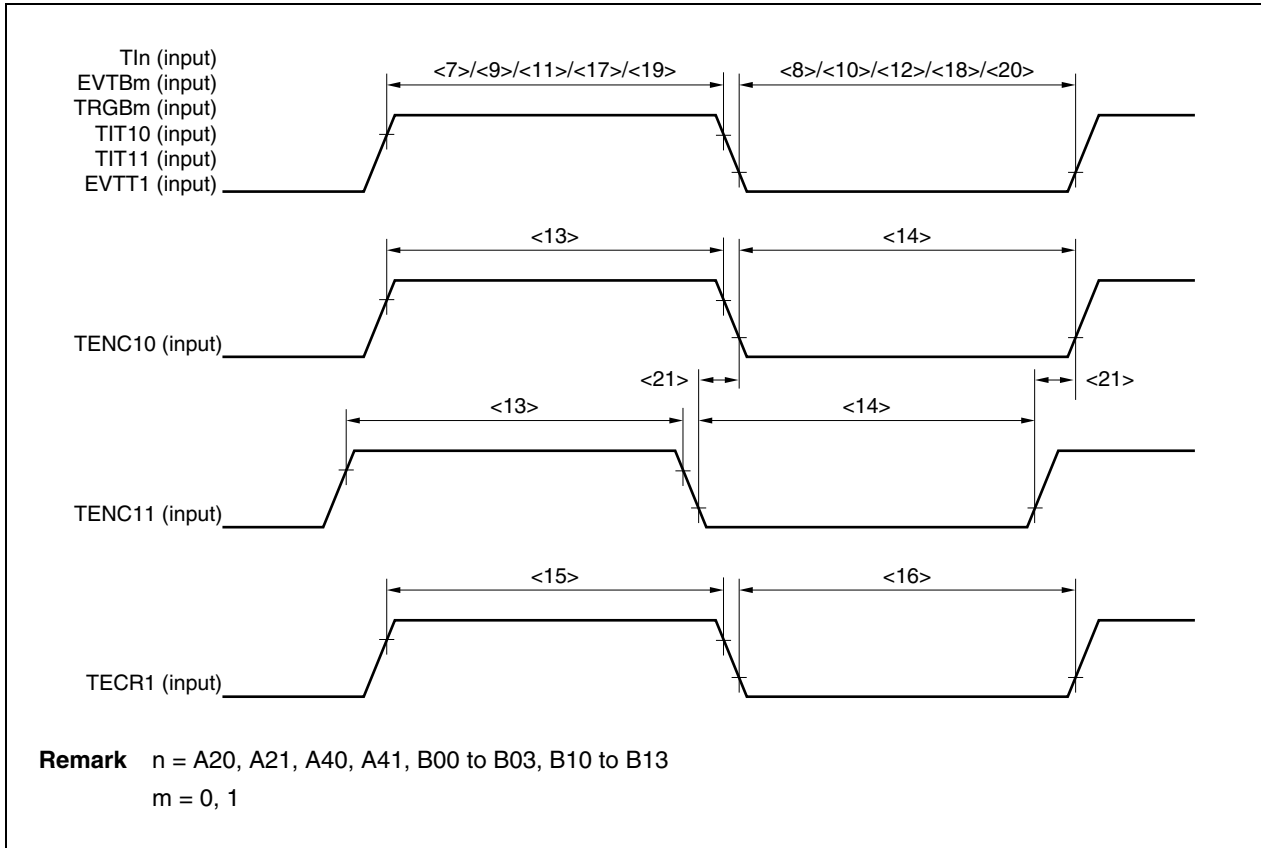
**Notes** 1.  $T = 1/f_{xx}$

2. T<sub>smp1</sub>: Noise elimination sampling clock cycle (set by TANFC2 and TANFC4 registers)

3. T<sub>smp2</sub>: Noise elimination sampling clock cycle (set by TTNFC1 register)

**Remark** The above specification shows a pulse width that is accurately detected as a valid edge. Even if a pulse narrower than the above specification is input, therefore, it may be detected as a valid edge.

Timer Input Timing





**(4) CSIB timing**
**(a) Master mode**
 $(T_A = -40 \text{ to } +85^\circ\text{C}, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 \text{ to } 5.5 \text{ V},$ 
 $V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 \text{ V}, C_L = 50 \text{ pF})$ 

| Parameter   | Symbol                                  | Conditions | MIN.                     | MAX. | Unit |
|---|---|------------|--------------------------|------|------|
| $\overline{\text{SCKBn}}$ cycle   | $t_{\text{KCYM}}$                       | <22>       | 125                      |      | ns   |
| $\overline{\text{SCKBn}}$ high-/low-level width   | $t_{\text{KWHM}},$<br>$t_{\text{KWLM}}$ | <23>       | $t_{\text{KCYM}}/2 - 10$ |      | ns   |
| SI $\overline{\text{Bn}}$ setup time (to $\overline{\text{SCKBn}}\uparrow$ )            | $t_{\text{SSIM}}$                       | <24>       | 30                       |      | ns   |
| SI $\overline{\text{Bn}}$ setup time (to $\overline{\text{SCKBn}}\downarrow$ )          |   |            | 30                       |      | ns   |
| SI $\overline{\text{Bn}}$ hold time (from $\overline{\text{SCKBn}}\uparrow$ )           | $t_{\text{HSIM}}$                       | <25>       | 30                       |      | ns   |
| SI $\overline{\text{Bn}}$ hold time (from $\overline{\text{SCKBn}}\downarrow$ )         |   |            | 30                       |      | ns   |
| SO $\overline{\text{Bn}}$ output delay time (from $\overline{\text{SCKBn}}\downarrow$ ) | $t_{\text{DSOM}}$                       | <26>       |                          | 30   | ns   |
| SO $\overline{\text{Bn}}$ output delay time (from $\overline{\text{SCKBn}}\uparrow$ )   |   |            |                          | 30   | ns   |
| SO $\overline{\text{Bn}}$ output hold time (from $\overline{\text{SCKBn}}\uparrow$ )    | $t_{\text{HSOM}}$                       | <27>       | $t_{\text{KCYM}}/2 - 10$ |      | ns   |
| SO $\overline{\text{Bn}}$ output hold time (from $\overline{\text{SCKBn}}\downarrow$ )  |   |            | $t_{\text{KCYM}}/2 - 10$ |      | ns   |

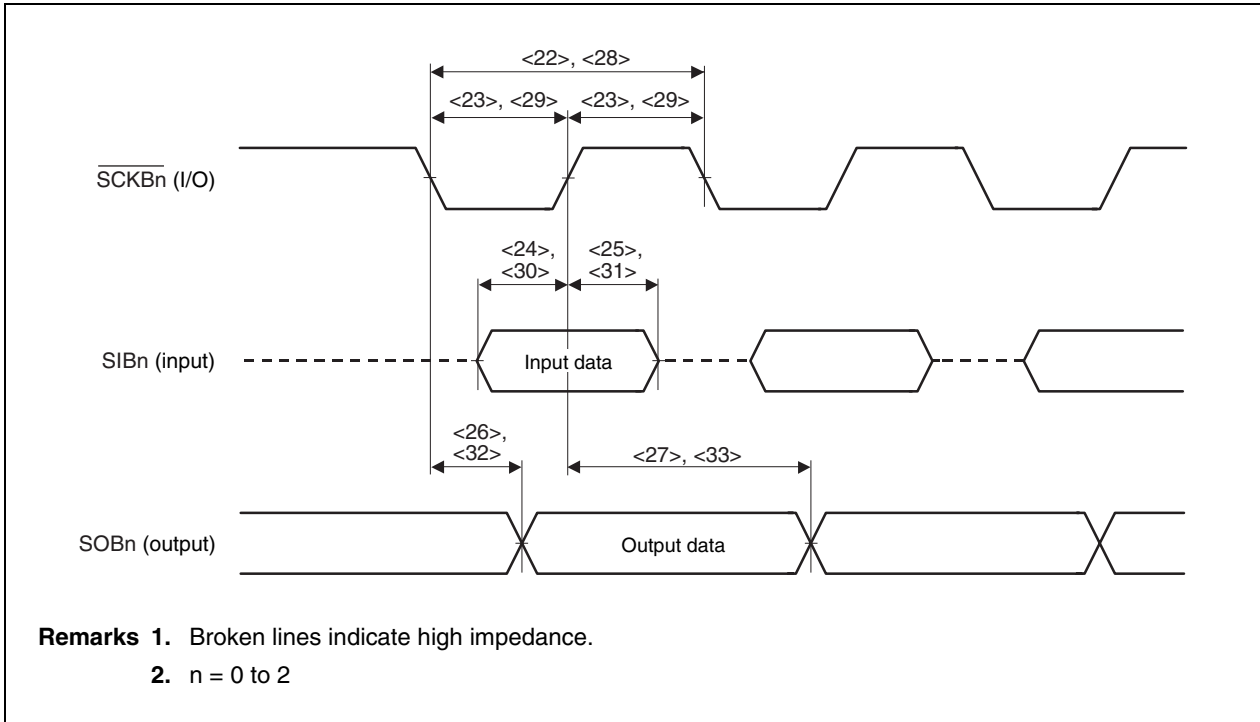
**Remark** n = 0 to 2

**(b) Slave mode**
 $(T_A = -40 \text{ to } +85^\circ\text{C}, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 \text{ to } 5.5 \text{ V},$ 
 $V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 \text{ V}, C_L = 50 \text{ pF})$ 

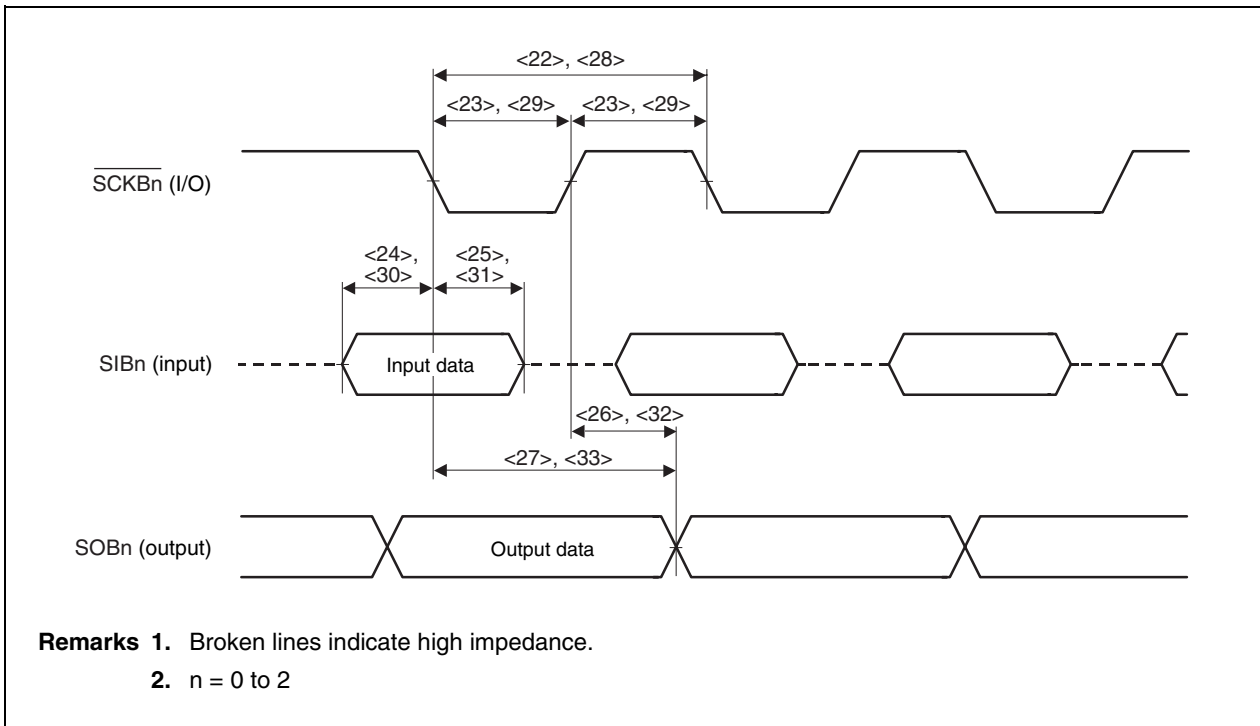
| Parameter   | Symbol                                  | Conditions | MIN.                     | MAX. | Unit |
|---|---|------------|--------------------------|------|------|
| $\overline{\text{SCKBn}}$ cycle   | $t_{\text{KCYS}}$                       | <28>       | 125                      |      | ns   |
| $\overline{\text{SCKBn}}$ high-/low-level width   | $t_{\text{KWHs}},$<br>$t_{\text{KWLS}}$ | <29>       | $t_{\text{KCYS}}/2 - 10$ |      | ns   |
| SI $\overline{\text{Bn}}$ setup time (to $\overline{\text{SCKBn}}\uparrow$ )            | $t_{\text{SSIS}}$                       | <30>       | 30                       |      | ns   |
| SI $\overline{\text{Bn}}$ setup time (to $\overline{\text{SCKBn}}\downarrow$ )          |   |            | 30                       |      | ns   |
| SI $\overline{\text{Bn}}$ hold time (from $\overline{\text{SCKBn}}\uparrow$ )           | $t_{\text{HSIS}}$                       | <31>       | 30                       |      | ns   |
| SI $\overline{\text{Bn}}$ hold time (from $\overline{\text{SCKBn}}\downarrow$ )         |   |            | 30                       |      | ns   |
| SO $\overline{\text{Bn}}$ output delay time (from $\overline{\text{SCKBn}}\downarrow$ ) | $t_{\text{DSOS}}$                       | <32>       |                          | 30   | ns   |
| SO $\overline{\text{Bn}}$ output delay time (from $\overline{\text{SCKBn}}\uparrow$ )   |   |            |                          | 30   | ns   |
| SO $\overline{\text{Bn}}$ output hold time (from $\overline{\text{SCKBn}}\uparrow$ )    | $t_{\text{HSOS}}$                       | <33>       | $t_{\text{KCYS}}/2 - 10$ |      | ns   |
| SO $\overline{\text{Bn}}$ output hold time (from $\overline{\text{SCKBn}}\downarrow$ )  |   |            | $t_{\text{KCYS}}/2 - 10$ |      | ns   |

**Remark** n = 0 to 2

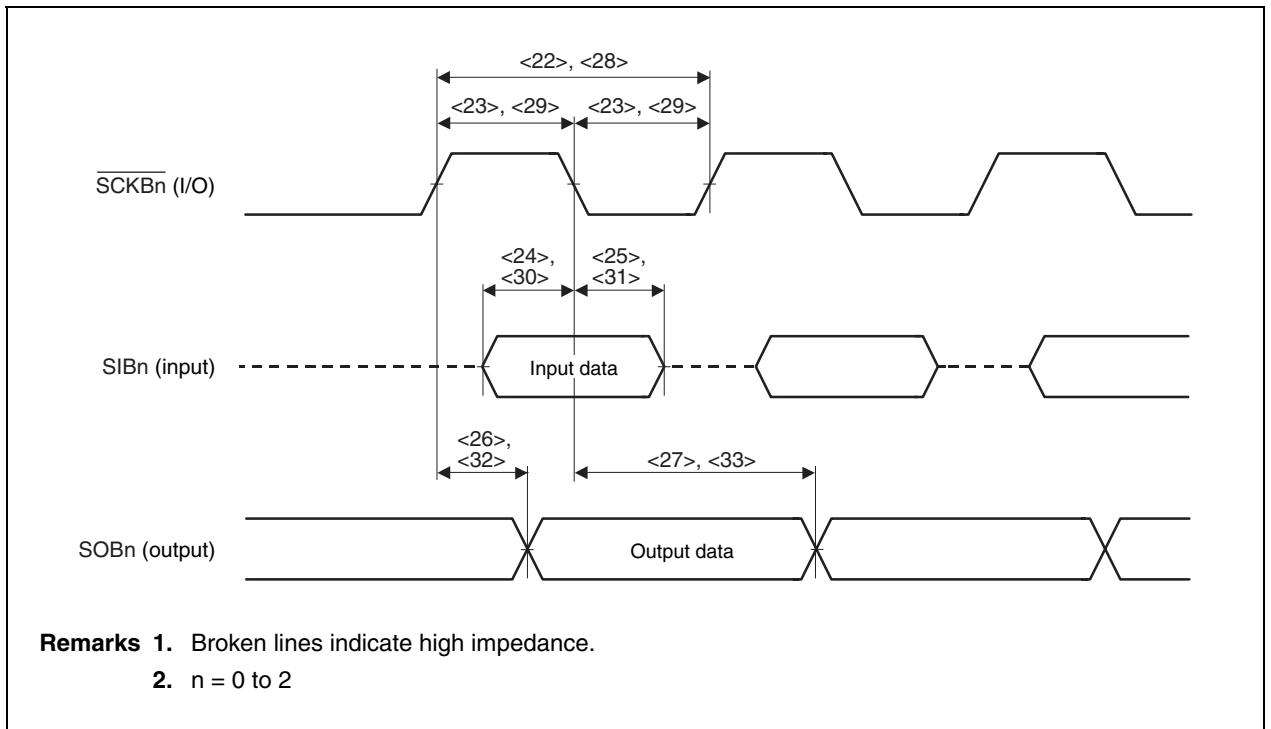
CSIB timing when CBnCKP and CBnDAP bits of CBnCTL1 register = 00



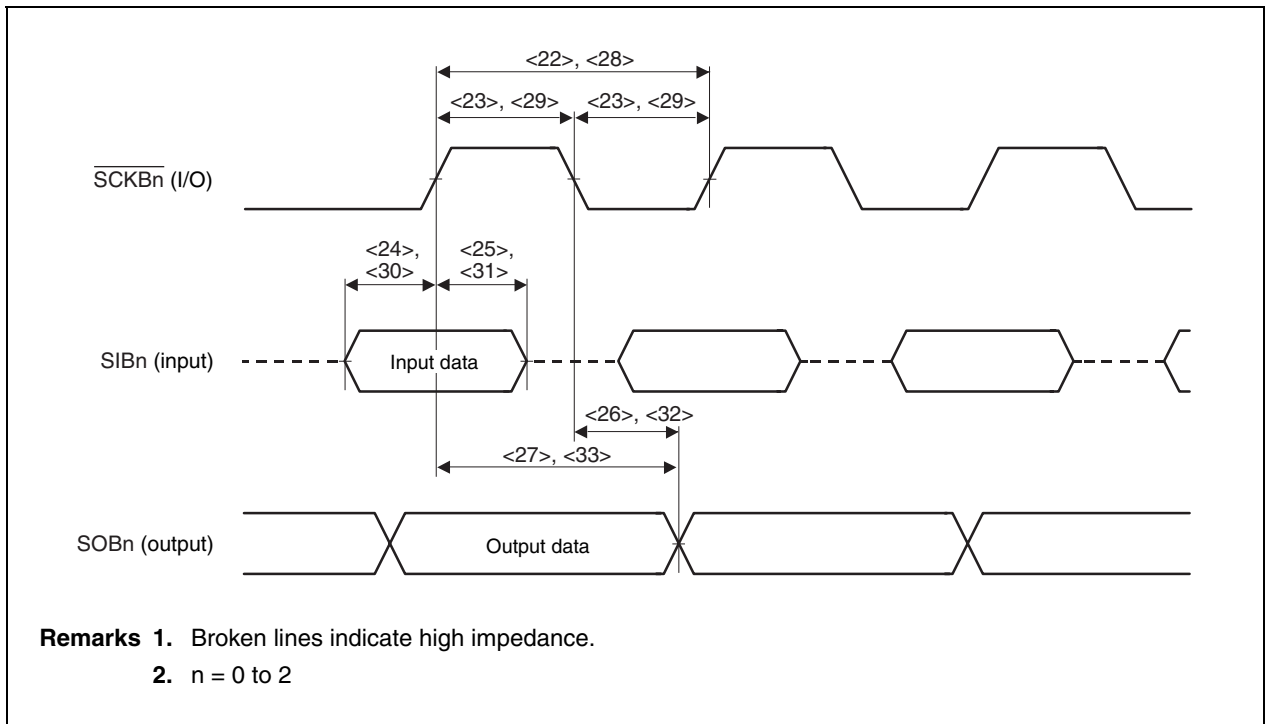
CSIB timing when CBnCKP and CBnDAP bits of CBnCTL1 register = 01



CSIB timing when CBNCKP and CBN DAP bits of CBNCTL1 register = 10



CSIB timing when CBNCKP and CBN DAP bits of CBNCTL1 register = 11



**(5) I<sup>2</sup>C bus timing**
**(T<sub>A</sub> = -40 to +85°C, V<sub>DD0</sub> = V<sub>DD1</sub> = EV<sub>DD0</sub> = EV<sub>DD1</sub> = AV<sub>DD0</sub> = AV<sub>DD1</sub> = AV<sub>DD2</sub> = 4.0 to 5.5 V,**
**V<sub>SS0</sub> = V<sub>SS1</sub> = EV<sub>SS0</sub> = EV<sub>SS1</sub> = AV<sub>SS0</sub> = AV<sub>SS1</sub> = AV<sub>SS2</sub> = 0 V, C<sub>L</sub> = 50 pF)**

| Parameter  |                        | Symbol              | Standard Mode |                     | High-Speed Mode |                              | Unit                  |     |
|--|------------------------|---------------------|---------------|---------------------|-----------------|------------------------------|-----------------------|-----|
|  |                        |                     | MIN.          | MAX.                | MIN.            | MAX.                         |                       |     |
| SCL clock frequency  |                        | f <sub>CLK</sub>    | –             | 0                   | 100             | 0                            | 400                   | kHz |
| Bus free time (between stop condition and start condition) |                        | t <sub>BUF</sub>    | <34>          | 4.7                 | –               | 1.3                          | –                     | μs  |
| Hold time <sup>Note 1</sup>                                |                        | t <sub>HD:STA</sub> | <35>          | 4.0                 | –               | 0.6                          | –                     | μs  |
| SCL clock low-level width                                  |                        | t <sub>LOW</sub>    | <36>          | 4.7                 | –               | 1.3                          | –                     | μs  |
| SCL clock high-level width                                 |                        | t <sub>HIGH</sub>   | <37>          | 4.0                 | –               | 0.6                          | –                     | μs  |
| Start/restart condition setup time                         |                        | t <sub>SU:STA</sub> | <38>          | 4.7                 | –               | 0.6                          | –                     | μs  |
| Data hold time   | CBUS-compatible master | t <sub>HD:DAT</sub> | <39>          | 5.0                 | –               | –                            | –                     | μs  |
|  | I <sup>2</sup> C mode  |                     |               | 0 <sup>Note 2</sup> | –               | 0 <sup>Note 2</sup>          | 0.9 <sup>Note 3</sup> | μs  |
| Data setup time  |                        | t <sub>SU:DAT</sub> | <40>          | 250                 | –               | 100 <sup>Note 4</sup>        | –                     | ns  |
| SDA, SCL signal rise time                                  |                        | t <sub>R</sub>      | <41>          | –                   | 1000            | 20 + 0.1Cb <sup>Note 5</sup> | 300                   | ns  |
| SDA, SCL signal fall time                                  |                        | t <sub>F</sub>      | <42>          | –                   | 300             | 20 + 0.1Cb <sup>Note 5</sup> | 300                   | ns  |
| Stop condition setup time                                  |                        | t <sub>SU:STO</sub> | <43>          | 4.0                 | –               | 0.6                          | –                     | μs  |
| Pulse width of spike suppressed by input filter            |                        | t <sub>SP</sub>     | <44>          | –                   | –               | 0                            | 50                    | ns  |
| Each bus line capacitive load                              |                        | C <sub>b</sub>      | –             | –                   | 400             | –                            | 400                   | pF  |

**Notes** 1. The first clock pulse is generated after a hold time during the start condition.

2. The system must internally supply a hold time of at least 300 ns for the SDA signal (at V<sub>IHmin.</sub> of SCL signal) to fill the undefined area at the falling edge of SCL.

3. If the system does not extend the low hold time (t<sub>LOW</sub>) of the SCL signal, the maximum data hold time (t<sub>HD:DAT</sub>) must be satisfied.

4. The high-speed mode I<sup>2</sup>C bus can be used in the standard mode I<sup>2</sup>C bus system. In this case, make sure that the following conditions are satisfied.

- If system does not extend the low status hold time of the SCL signal

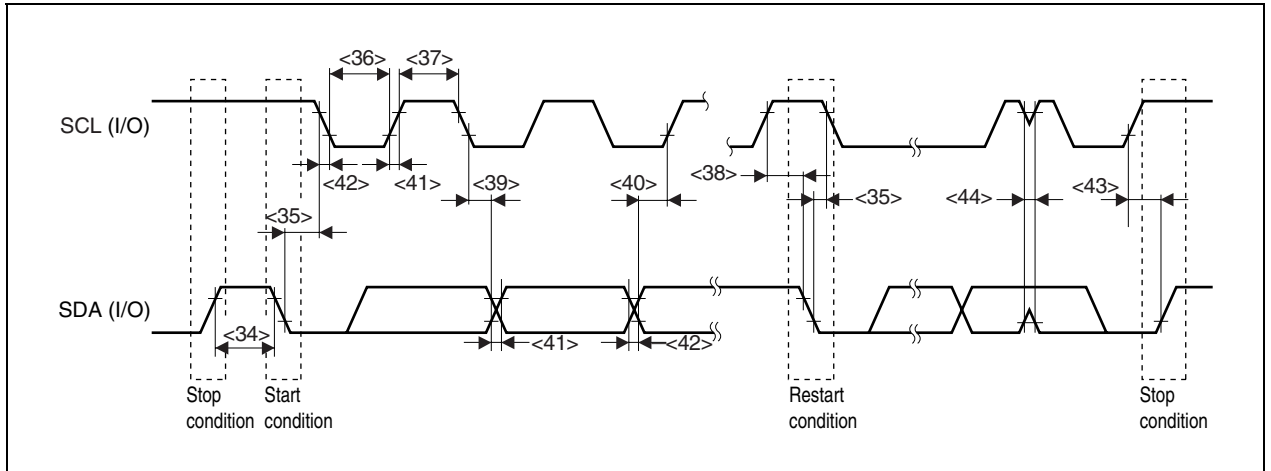
$$t_{SU:DAT} \geq 250 \text{ ns}$$

- If system extends the low status hold time of SCL signal

Sends the next data bit to the SDA line before the SCL line is released (t<sub>Rmax.</sub> + t<sub>SU:DAT</sub> = 1000 + 250 = 1250 ns: standard mode I<sup>2</sup>C bus specification).

5. C<sub>b</sub>: Total capacitance of one bus line (unit: pF)

I<sup>2</sup>C bus timing



(6) High-impedance control timing

(T<sub>A</sub> = -40 to +85°C, V<sub>DD0</sub> = V<sub>DD1</sub> = EV<sub>DD0</sub> = EV<sub>DD1</sub> = AV<sub>DD0</sub> = AV<sub>DD1</sub> = AV<sub>DD2</sub> = 4.0 to 5.5 V,

V<sub>SS0</sub> = V<sub>SS1</sub> = EV<sub>SS0</sub> = EV<sub>SS1</sub> = AV<sub>SS0</sub> = AV<sub>SS1</sub> = AV<sub>SS2</sub> = 0 V, C<sub>L</sub> = 50 pF)

| Parameter   | Symbol            | Conditions                      | MIN. | MAX. | Unit |
|---|-------------------|---------------------------------|------|------|------|
| Oscillation stop → timer output high impedance                    | t <sub>CLM</sub>  | When clock monitor is operating |      | 65   | μs   |
| Input to TOBnOFF → timer output high impedance                    | t <sub>HTQn</sub> |                                 |      | 300  | ns   |
| Input to TOA2OFF → timer output high impedance                    | t <sub>HTP2</sub> |                                 |      | 300  | ns   |
| Input to ANI00/ANI05 → timer output high impedance                | t <sub>ANI0</sub> |                                 |      | 10   | μs   |
| Input to ANI10/ANI15 to ANI12/ANI17 → timer output high impedance | t <sub>ANI1</sub> |                                 |      | 10   | μs   |

**Remark** n = 0, 1

**28.1.9 Characteristics of A/D converters 0 and 1**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = AV_{REFP0} = AV_{REFP1} = 4.0$  to  $5.5$  V,  $V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V,  $C_L = 50$  pF)

| Parameter                                      | Symbol     | Conditions   | MIN.      | TYP. | MAX.      | Unit          |
|--|------------|--|-----------|------|-----------|---------------|
| Resolution                                     |            |  | 12        | 12   | 12        | bit           |
| Overall error <sup>Note 1</sup>                |            |  |           |      | $\pm 10$  | LSB           |
| Conversion time                                | $t_{CONV}$ | $f_{AD01} = 16$ MHz, ADAnCTC register = 0BH or 0CH | 2.0       |      |           | $\mu\text{s}$ |
|  |            | $f_{AD01} = 12$ MHz, ADAnCTC register = 00H        |           |      | 7.42      | $\mu\text{s}$ |
| Zero scale error <sup>Note 1</sup>             |            |  |           |      | $\pm 10$  | LSB           |
| Full-scale error <sup>Note 1</sup>             |            |  |           |      | $\pm 10$  | LSB           |
| Integral linearity error <sup>Note 1</sup>     |            |  |           |      | $\pm 4$   | LSB           |
| Differential linearity error <sup>Note 1</sup> |            |  |           |      | $\pm 2.5$ | LSB           |
| Analog reference voltage                       | $AV_{DD}$  |  | 4.0       |      | 5.5       | V             |
| Analog input voltage                           | $V_{IAN}$  |  | $AV_{SS}$ |      | $AV_{DD}$ | V             |
| $AV_{DD}$ supply current <sup>Note 2</sup>     | $AI_{DD}$  | Operating  |           | 4.5  | 7.5       | mA            |
|  | $AI_{DSS}$ | In STOP mode <sup>Note 3</sup>                     |           | 3.5  | 17.5      | $\mu\text{s}$ |

- Notes**
1. Excludes quantization error ( $\pm 0.5$  LSB).
  2. This value is for only one A/D converter (A/D converter 0 or 1).
  3. Stop the operation of A/D converters 0 and 1 (ADnSCM.ADnCE bit = 0) before setting STOP mode.

- Remarks**
1. LSB: Least Significant Bit
  2.  $f_{AD01}$ : Base clock of A/D converters 0 and 1
  3.  $n = 0, 1$

**28.1.10 Characteristics of A/D converter 2**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0$  to  $5.5$  V,

$V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V,  $C_L = 50$  pF)

| Parameter                                      | Symbol     | Conditions                     | MIN.      | TYP. | MAX.      | Unit |
|--|------------|--------------------------------|-----------|------|-----------|------|
| Resolution                                     |            |                                | 10        | 10   | 10        | bit  |
| Overall error <sup>Note 1</sup>                |            |                                |           |      | ±4.0      | LSB  |
| Conversion time                                | $t_{CONV}$ |                                | 3.88      |      | 10        | μs   |
| Zero scale error <sup>Note 1</sup>             |            |                                |           |      | ±4.0      | LSB  |
| Full-scale error <sup>Note 1</sup>             |            |                                |           |      | ±4.0      | LSB  |
| Integral linearity error <sup>Note 1</sup>     |            |                                |           |      | ±4.0      | LSB  |
| Differential linearity error <sup>Note 1</sup> |            |                                |           |      | ±2.0      | LSB  |
| Analog reference voltage                       | $AV_{DD}$  |                                | 4.0       |      | 5.5       | V    |
| Analog input voltage                           | $V_{IAN}$  |                                | $AV_{SS}$ |      | $AV_{DD}$ | V    |
| $AV_{DD}$ supply current                       | $I_{DD}$   | During operation               |           | 3.5  | 7         | mA   |
|  | $I_{DDs}$  | In STOP mode <sup>Note 2</sup> |           | 1    | 10        | μA   |

**Notes** 1. Excludes quantization error ( $\pm 0.5$  LSB).

2. Stop the operation of A/D converter 2 (AD2M0.AD2CE bit = 0) before setting STOP mode.

**Remark** LSB: Least Significant Bit

**28.1.11 Operational amplifier characteristics**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0$  to  $5.5$  V,

$V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V,  $C_L = 50$  pF)

| Parameter                           | Symbol     | Conditions                     | MIN.                  | TYP.      | MAX.           | Unit             |   |
|-------------------------------------|------------|--------------------------------|-----------------------|-----------|----------------|------------------|---|
| Input offset voltage                | $V_{IO}$   |                                |                       | $\pm 9.0$ |                | mV               |   |
| Input voltage range                 | $V_I$      | Gain = 2.500                   | $0.04AV_{DD}$         |           | $0.36AV_{DD}$  | V                |   |
|                                     |            | Gain = 5.000                   | $0.02AV_{DD}$         |           | $0.18AV_{DD}$  | V                |   |
|                                     |            | Gain = 10.00                   | $0.01AV_{DD}$         |           | $0.085AV_{DD}$ | V                |   |
| Slew rate <sup>Note 1</sup>         | $S_R$      |                                | 10                    | 15        |                | V/ $\mu\text{s}$ |   |
| Gain error                          |            | <b>Note 2</b>                  | Gain = 2.500 to 4.444 |           | $\pm 1.0$      | $\pm 1.3$        | % |
|                                     |            |                                | Gain = 5.000 to 6.667 |           | $\pm 1.0$      | $\pm 1.5$        | % |
|                                     |            |                                | Gain = 8.000, 10.00   |           | $\pm 1.0$      | $\pm 1.7$        | % |
|                                     |            | <b>Note 3</b>                  | Gain = 2.500 to 4.444 |           | $\pm 1.0$      | $\pm 2.0$        | % |
|                                     |            |                                | Gain = 5.000 to 6.667 |           | $\pm 1.0$      | $\pm 2.1$        | % |
|                                     |            |                                | Gain = 8.000, 10.00   |           | $\pm 1.0$      | $\pm 2.2$        | % |
| Operating current <sup>Note 4</sup> | $I_{OPDD}$ | During operation               |                       | 1.8       | 2.6            | mA               |   |
|                                     | $A_{IDDS}$ | In STOP mode <sup>Note 5</sup> |                       | 1.0       | 10             | $\mu\text{A}$    |   |

**Notes** 1. Inclination characteristic of output voltage from 10% to 90%

2.  $AV_{DD0} = AV_{DD1} = 4.5$  to  $5.5$  V

3.  $AV_{DD0} = AV_{DD1} = 4.0$  to  $5.5$  V

4. Four operational amplifiers are provided in total. The value shows the operating current per operational amplifier.

5. Stop operational amplifier operation (OP0CTL0.OP0EN bit = 0, OP1CTL0.OP12EN, OP11EN, and OP10EN bits = 0) before setting STOP mode.

**Remark** Power supplies  $AV_{DD0}$  and  $AV_{DD1}$  are used for the operational amplifier.



**28.1.12 Comparator characteristics**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0$  to  $5.5$  V,

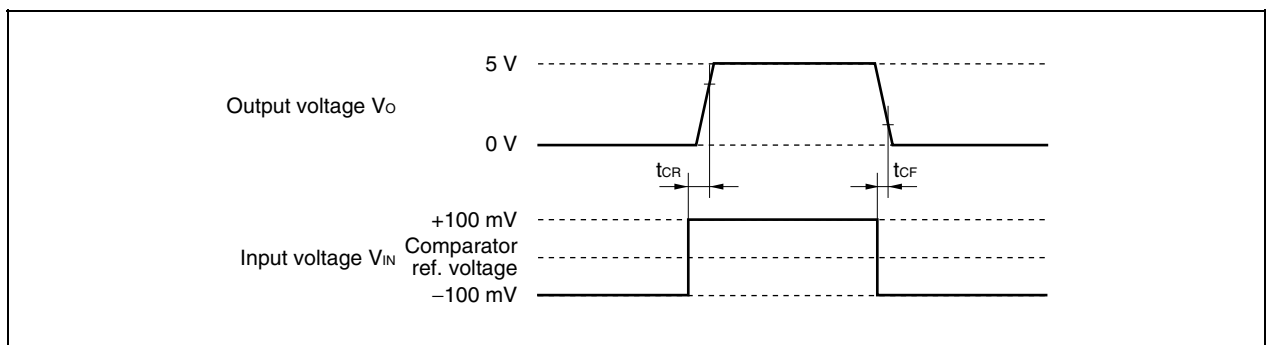
$V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V,  $C_L = 50$  pF)

| Parameter                                 | Symbol     | Conditions  | MIN.                | TYP.      | MAX.                | Unit          |
|---|------------|---|---------------------|-----------|---------------------|---------------|
| Input offset voltage                      | $V_{IO}$   |   |                     | $\pm 3.0$ |                     | mV            |
| Input voltage range                       | $V_I$      |   | $AV_{SS}$           |           | $AV_{DD}$           | V             |
| Comparator reference voltage (full range) | $CREFnF$   |   | $0.02AV_{DD} + 0.1$ |           | $0.92AV_{DD} - 0.1$ | V             |
| Comparator reference voltage (low range)  | $CREFnL$   |   | $0.02AV_{DD} + 0.1$ |           | $0.5AV_{DD} - 0.1$  | V             |
| Response time                             | $t_{CR}$   | Input amplitude = 100 mV, at rising edge <sup>Note 1</sup>  |                     | 1.0       |                     | $\mu\text{s}$ |
|   | $t_{CF}$   | Input amplitude = 100 mV, at falling edge <sup>Note 2</sup> |                     | 1.0       |                     | $\mu\text{s}$ |
| Operating current <sup>Note 3</sup>       | $I_{CPDD}$ | During operation  |                     |           | 250                 | $\mu\text{A}$ |
|   | $A_{IDDS}$ | In STOP mode <sup>Note 4</sup>                              |                     | 2.0       | 20                  | nA            |

- Notes**
1. Characteristics of pulse response when ANIm input changes from the comparator reference voltage – 100 mV to the comparator reference voltage + 100 mV
  2. Characteristics of pulse response when ANIm input changes from the comparator reference voltage + 100 mV to the comparator reference voltage – 100 mV
  3. Four comparators are provided in total. The value shows the operating current per comparator.
  4. Stop comparator operation (CMPnCTL0 register = 00H) before setting STOP mode.

- Remarks**
1. Power supplies for the comparators are  $AV_{DD0}$  and  $AV_{DD1}$ .
  2.  $m = 05, 15$  to  $17$   
 $n = 0, 1$

**Comparator Characteristics**



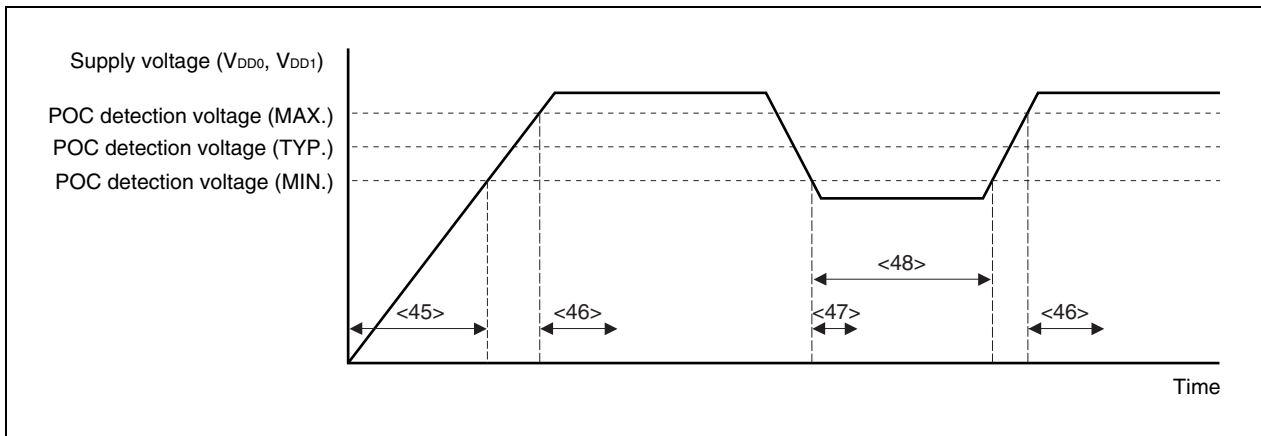
**28.1.13 Power-on-clear circuit (POC)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5$  to  $5.5$  V,

$V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V,  $C_L = 50$  pF)

| Parameter                           | Symbol     | Conditions  | MIN.              | TYP. | MAX.  | Unit |
|-------------------------------------|------------|---|-------------------|------|-------|------|
| POC detection voltage               | $V_{POC0}$ |   | 3.5               | 3.7  | 3.9   | V    |
| Supply voltage rise time            | $t_{PTH}$  | <45> $V_{DD0}, V_{DD1} = 0$ to $3.5$ V                                | $2.5 \mu\text{s}$ |      | 1.8 s |      |
| Response time 1 <sup>Note 1</sup>   | $t_{PTH0}$ | <46> After $V_{DD0}$ and $V_{DD1}$ reach $3.9$ V on power application |                   |      | 3.0   | ms   |
| Response time 2 <sup>Note 2</sup>   | $t_{PD}$   | <47> After $V_{DD0}$ and $V_{DD1}$ drop to $3.5$ V on power off       |                   |      | 1.0   | ms   |
| Minimum width of $V_{DD0}, V_{DD1}$ | $t_{PW}$   | <48>  | 0.2               |      |       | ms   |

- Notes**
1. The time required to release a reset signal (POCRES) after the POC detection voltage is detected.
  2. The time required to output a reset signal (POCRES) after the POC detection voltage is detected.



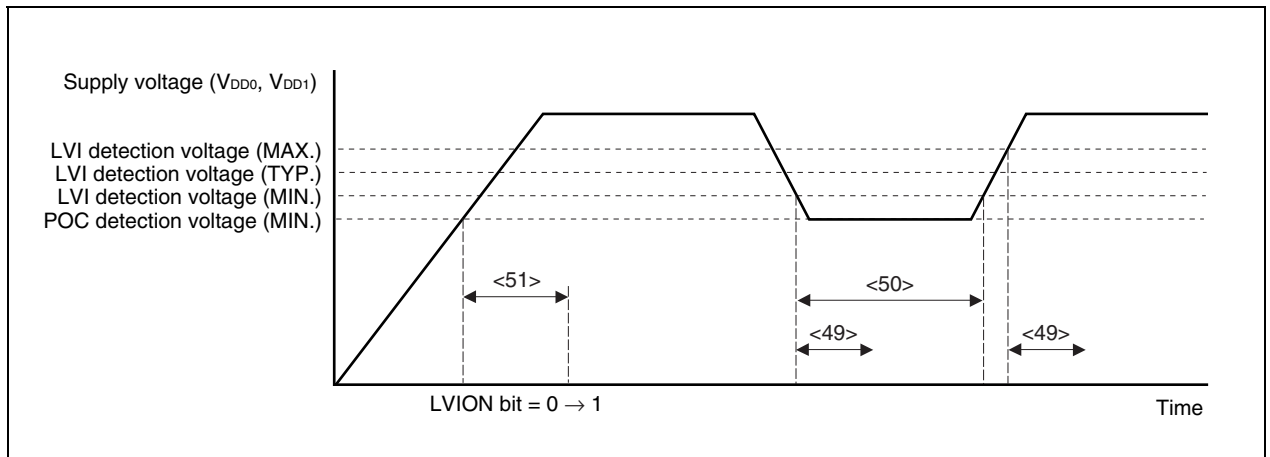
**28.1.14 Low-voltage detector (LVI)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5$  to  $5.5$  V,

$V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V,  $C_L = 50$  pF)

| Parameter                                 | Symbol      | Conditions  | MIN. | TYP. | MAX. | Unit |
|---|-------------|---|------|------|------|------|
| LVI detection voltage                     | $V_{LVI0}$  | LVIS.LVIS0 bit = 0  | 4.2  | 4.4  | 4.6  | V    |
|   | $V_{LVI1}$  | LVIS.LVIS0 bit = 1  | 4.0  | 4.2  | 4.4  | V    |
| Response time 1 <sup>Note</sup>           | $t_{LD}$    | <49> After $V_{DD0}$ and $V_{DD1}$ reach $V_{LVI0}/V_{LVI1}$ (MAX.) or drop to $V_{LVI0}/V_{LVI1}$ (MIN.)           |      | 0.2  | 2.0  | ms   |
| Minimum width of $V_{DD0}$ , $V_{DD1}$    | $t_{LW}$    | <50>  | 0.2  |      |      | ms   |
| Reference voltage stabilization wait time | $t_{LWAIT}$ | <51> After $V_{DD0}$ and $V_{DD1}$ reach POC detection voltage (MIN.) and the LVIM.LVION bit is changed from 0 to 1 |      | 0.1  |      | ms   |

**Note** The time required to output an interrupt request signal (INTLVIL, INTLVIH) or internal reset signal (LVIRES) after the LVI detection voltage is detected.



**28.1.15 Flash memory programming characteristics**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5$  to  $5.5$  V,

$V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V,  $C_L = 50$  pF)

| Parameter     | Symbol     | Conditions  | MIN. | TYP. | MAX. | Unit  |
|---------------|------------|-------------|------|------|------|-------|
| Rewrite count | $C_{ERWR}$ | <b>Note</b> |      | 100  |      | Times |

**Note** Rewrite as follows.

Example when three rewrites: Shipped product  $\rightarrow E \rightarrow P \rightarrow E \rightarrow P \rightarrow E \rightarrow P$  (P: Write, E: Erase)

## 28.2 V850E/IG3

### 28.2.1 Absolute maximum ratings

( $T_A = 25^\circ\text{C}$ )

| Parameter                          | Symbol     | Conditions  | Ratings                                   | Unit             |    |
|------------------------------------|------------|---|---|------------------|----|
| Supply voltage                     | $V_{DD}$   | $V_{DDa} = EV_{DDb} = AV_{DDk}$                           | -0.5 to +6.5                              | V                |    |
|                                    | $V_{SS}$   | $V_{SSa} = EV_{SSb} = AV_{SSk}$                           | -0.5 to +0.5                              | V                |    |
|                                    | $EV_{DD}$  | $V_{DDa} = EV_{DDb} = AV_{DDk}$                           | -0.5 to +6.5                              | V                |    |
|                                    | $EV_{SS}$  | $V_{SSa} = EV_{SSb} = AV_{SSk}$                           | -0.5 to +0.5                              | V                |    |
|                                    | $AV_{DD}$  | $V_{DDa} = EV_{DDb} = AV_{DDk}$                           | -0.5 to +6.5                              | V                |    |
|                                    | $AV_{SS}$  | $V_{SSa} = EV_{SSb} = AV_{SSk}$                           | -0.5 to +0.5                              | V                |    |
| Input voltage                      | $V_{I1}$   | <b>Note 1</b>   | -0.5 to $EV_{DD} + 0.5$ <sup>Note 2</sup> | V                |    |
|                                    | $V_{I2}$   | X1, X2  | -0.5 to $V_{RO} + 0.35$                   | V                |    |
| Output current, low                | $I_{OL}$   | All pins  | Per pin                                   | 4                | mA |
|                                    |            |   | Total of all pins                         | 63               | mA |
| Output current, high               | $I_{OH}$   | All pins  | Per pin                                   | -4               | mA |
|                                    |            |   | Total of all pins                         | -63              | mA |
| Analog input voltage               | $V_{IAN}$  | P70/ANI20 to P77/ANI27,<br>ANI00 to ANI05, ANI10 to ANI17 | -0.5 to $AV_{DD} + 0.5$ <sup>Note 2</sup> | V                |    |
| Analog reference input voltage     | $V_{IREF}$ | $AV_{REFP0}$ , $AV_{REFP1}$                               | -0.5 to $AV_{DD} + 0.5$ <sup>Note 2</sup> | V                |    |
| Comparator reference input voltage | $V_{CREF}$ | CREF0L, CREF1L, CREF0F, CREF1F                            | -0.5 to $AV_{DD} + 0.5$ <sup>Note 2</sup> | V                |    |
| Operating ambient temperature      | $T_A$      | In normal operating mode                                  | -40 to +85                                | $^\circ\text{C}$ |    |
|                                    |            | In flash memory programming mode                          | -40 to +85                                | $^\circ\text{C}$ |    |
| Operating ambient temperature      | $T_{stg}$  |   | -40 to +125                               | $^\circ\text{C}$ |    |

- Notes** 1. P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, PDL0 to PDL15,  $\overline{\text{RESET}}$ , FLMD0, DRST  
 2. Be sure not to exceed the absolute maximum ratings (MAX. value) of each supply voltage.

- Cautions** 1. Do not directly connect the output pins (or I/O pins in the output state) of IC products to other output pins (including I/O pins in the output state), power supply pins such as  $V_{DD}$  and  $EV_{DD}$ , or GND pin. Direct connection of the output pins between an IC product and an external circuit is possible, if the output pins can be set to the high-impedance state and the output timing of the external circuit is designed to avoid output conflict.
2. Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.
- The ratings and conditions indicated for DC characteristics and AC characteristics represent the quality assurance range during normal operation.

**Remark** a = 0, 1  
 b = 0 to 2  
 k = 0 to 2

### 28.2.2 Capacitance

( $T_A = 25^\circ\text{C}$ ,  $V_{DD0} = V_{SS0} = V_{DD1} = V_{SS1} = EV_{DD0} = EV_{SS0} = EV_{DD1} = EV_{SS1} = EV_{DD2} = EV_{SS2} = AV_{DD0} = AV_{SS0} = AV_{DD1} = AV_{SS1} = AV_{DD2} = AV_{SS2} = 0\text{ V}$ )

| Parameter          | Symbol   | Conditions                      | MIN. | TYP. | MAX. | Unit |
|--------------------|----------|---------------------------------|------|------|------|------|
| Input capacitance  | $C_i$    | $f_c = 1\text{ MHz}$            |      |      | 15   | pF   |
| I/O capacitance    | $C_{iO}$ | Unmeasured pins returned to 0 V |      |      | 15   | pF   |
| Output capacitance | $C_o$    |                                 |      |      | 15   | pF   |

- Notes**
1. ANI00 to ANI05, ANI10 to ANI17,  $\overline{\text{RESET}}$
  2. P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P70 to P77, PDL0 to PDL15
  3. DDO

- Cautions**
1. Excludes the FLMD0,  $\overline{\text{DRST}}$ , X1, and X2 pins.
  2. In addition to input capacitance, sampling capacitance is added to the ANI00 to ANI05, ANI10 to ANI17, and ANI20 to ANI27 pins when sampling.

### 28.2.3 Operating conditions

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0\text{ V}$ )

| Parameter                    | Symbol               | Conditions                                   | MIN. | TYP. | MAX. | Unit |
|------------------------------|----------------------|--|------|------|------|------|
| System clock frequency       | $f_{xx}$             | PLL mode                                     | 32   |      | 64   | MHz  |
|                              |                      | Clock through mode                           | 4    |      | 8    | MHz  |
| CPU clock frequency          | $f_{CPU}$            | PLL mode                                     | 4    |      | 64   | MHz  |
|                              |                      | Clock through mode                           | 0.5  |      | 8    | MHz  |
| $V_{DD}$ , $EV_{DD}$ voltage | $V_{DD}$ , $EV_{DD}$ | <b>Note 1</b> When external bus is not used  | 3.5  |      | 5.5  | V    |
|                              |                      | When external bus is used <sup>Note 2</sup>  | 4.0  |      | 5.5  | V    |
| $AV_{DD}$ voltage            | $AV_{DD}$            | When A/D converters 0 to 2 are operating     | 4.0  |      | 5.5  | V    |
|                              |                      | When A/D converters 0 to 2 are not operating | 3.5  |      | 5.5  | V    |

- Notes**
1.  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2}$
  2.  $\mu\text{PD70F3454GC-8EA-A}$  and  $70F3454F1-DA9-A$  only

**28.2.4 Clock oscillator characteristics**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5$  to  $5.5$  V,

$V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V)

| Resonator                  | Recommended Circuit | Parameter                       | Conditions              | MIN. | TYP.         | MAX. | Unit |
|----------------------------|---------------------|---------------------------------|-------------------------|------|--------------|------|------|
| Ceramic /crystal resonator |                     | Oscillation frequency ( $f_x$ ) |                         | 4    |              | 8    | MHz  |
|                            |                     | Oscillation stabilization time  | After reset release     |      | $2^{14}/f_x$ |      | ms   |
|                            |                     |                                 | After STOP mode release |      | <b>Note</b>  |      | ms   |

**Note** The value varies depending on the setting of the oscillation stabilization time select register (OSTS).

- Cautions**
1. Connect the oscillator as close to the X1 and X2 pins as possible.
  2. Do not cross the wiring with the other signal lines in the area enclosed by the broken lines in the above figure.
  3. For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.
  4. Inputting an external clock to the V850E/IG3 is prohibited.

(i) Murata Mfg. Co., Ltd.: Ceramic resonator ( $T_A = -40$  to  $+85^\circ\text{C}$ ) (Products other than  $\mu\text{PD70F3454F1-DA9-A}$ )

| Type             | Part Number     | Oscillation Frequency $f_x$ (MHz) | Recommended Circuit Constant |                 |                  | Oscillation Voltage Range |          |
|------------------|-----------------|-----------------------------------|------------------------------|-----------------|------------------|---------------------------|----------|
|                  |                 |                                   | C1                           | C2              | Rd (k $\Omega$ ) | MIN. (V)                  | MAX. (V) |
| Lead             | CSTLS4M00G56-B0 | 4                                 | On chip (47 pF)              | On chip (47 pF) | 0                | 3.5                       | 5.5      |
|                  | CSTLS5M00G56-B0 | 5                                 | On chip (47 pF)              | On chip (47 pF) | 0                | 3.5                       | 5.5      |
|                  | CSTLS8M00G56-B0 | 8                                 | On chip (47 pF)              | On chip (47 pF) | 0                | 3.5                       | 5.5      |
| Surface mounting | CSTCR4M00G55-R0 | 4                                 | On chip (39 pF)              | On chip (39 pF) | 0                | 3.5                       | 5.5      |
|                  | CSTCR5M00G55-R0 | 5                                 | On chip (39 pF)              | On chip (39 pF) | 0                | 3.5                       | 5.5      |
|                  | CSTCE8M00G55-R0 | 8                                 | On chip (33 pF)              | On chip (33 pF) | 0                | 3.5                       | 5.5      |

**Caution** These oscillator constants are reference values based on evaluation under a specific environment by the resonator manufacturer. When optimization of the oscillator characteristics on the actual application is necessary, request evaluation on the mounting circuit from the resonator manufacturer.

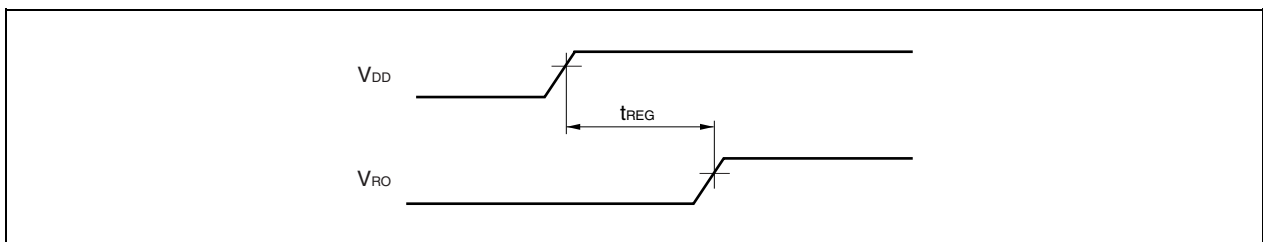
The oscillation voltage and oscillation frequency indicate only oscillator characteristics, therefore use the V850E/IG3 within the DC characteristics and AC characteristics for internal operation conditions.

28.2.5 Regulator characteristics

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = 3.5$  to  $5.5$  V,  $V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = 0$  V)

| Parameter                         | Symbol    | Conditions  | MIN. | TYP. | MAX.                | Unit |
|-----------------------------------|-----------|---|------|------|---------------------|------|
| Input voltage                     | $V_{DD}$  |   | 3.5  |      | 5.5                 | V    |
| Output voltage                    | $V_{RO}$  |   |      | 1.5  |                     | V    |
| Output voltage stabilization time | $t_{REG}$ | Stabilizing capacitor $C = 4.7 \mu\text{F}$ <sup>Note 1</sup> |      |      | 1 <sup>Note 2</sup> | ms   |

- Notes**
1. Connect a stabilizing capacitor between the REGC0 pin and  $V_{SS0}$  pin, and between the REGC1 pin and  $V_{SS1}$  pin.
  2. Internal reset signal is output until the power-on-clear circuit (POC) output voltage stabilizes during  $t_{REG}$  period.





**28.2.6 DC characteristics**

 (T<sub>A</sub> = -40 to +85°C, V<sub>DD0</sub> = V<sub>DD1</sub> = EV<sub>DD0</sub> = EV<sub>DD1</sub> = EV<sub>DD2</sub> = AV<sub>DD0</sub> = AV<sub>DD1</sub> = AV<sub>DD2</sub> = 3.5 to 5.5 V,

 V<sub>SS0</sub> = V<sub>SS1</sub> = EV<sub>SS0</sub> = EV<sub>SS1</sub> = EV<sub>SS2</sub> = AV<sub>SS0</sub> = AV<sub>SS1</sub> = AV<sub>SS2</sub> = 0 V) (1/2)

| Parameter                            | Symbol            | Conditions  |                           | MIN.                   | TYP.                   | MAX.                | Unit |
|--------------------------------------|-------------------|---|---------------------------|------------------------|------------------------|---------------------|------|
| Input voltage, high                  | V <sub>IH1</sub>  | <b>Note 1</b>                                     |                           | 0.7EV <sub>DD</sub>    |                        | EV <sub>DD</sub>    | V    |
|                                      | V <sub>IH2</sub>  | <b>Note 2</b>                                     |                           | 0.8EV <sub>DD</sub>    |                        | EV <sub>DD</sub>    | V    |
|                                      | V <sub>IH3</sub>  | <b>Note 3</b>                                     |                           | 2.2                    |                        | EV <sub>DD</sub>    | V    |
|                                      | V <sub>IH4</sub>  | <b>Note 4</b>                                     |                           | 0.7AV <sub>DD</sub>    |                        | AV <sub>DD</sub>    | V    |
| Input voltage, low                   | V <sub>IL1</sub>  | <b>Note 1</b>                                     |                           | EV <sub>SS</sub>       |                        | 0.3EV <sub>DD</sub> | V    |
|                                      | V <sub>IL2</sub>  | <b>Note 2</b>                                     |                           | EV <sub>SS</sub>       |                        | 0.2EV <sub>DD</sub> | V    |
|                                      | V <sub>IL3</sub>  | <b>Note 3</b>                                     |                           | EV <sub>SS</sub>       |                        | 0.8                 | V    |
|                                      | V <sub>IL4</sub>  | <b>Note 4</b>                                     |                           | AV <sub>SS</sub>       |                        | 0.3AV <sub>DD</sub> | V    |
| Input leakage current, high          | I <sub>LIH1</sub> | V <sub>I</sub> = <b>Note 5</b> ,<br><b>Note 6</b> | Other than X1             |                        |                        | 5                   | μA   |
|                                      | I <sub>LIH2</sub> |   | X1                        |                        |                        | 20                  | μA   |
| Input leakage current, low           | I <sub>LIL1</sub> | V <sub>I</sub> = 0 V                              | Other than X1             |                        |                        | -5                  | μA   |
|                                      | I <sub>LIL2</sub> |   | X1                        |                        |                        | -20                 | μA   |
| Output leakage current, high         | I <sub>LOH</sub>  | V <sub>O</sub> = <b>Note 5</b>                    |                           |                        |                        | 5                   | μA   |
| Output leakage current, low          | I <sub>LOL</sub>  | V <sub>O</sub> = 0 V                              |                           |                        |                        | -5                  | μA   |
| Output voltage, high                 | V <sub>OH1</sub>  | <b>Note 7</b>                                     | I <sub>OH</sub> = -1.0 mA | Total of pins = -57 mA | EV <sub>DD</sub> - 1.0 |                     | V    |
| Output voltage, low                  | V <sub>OL1</sub>  | <b>Note 7</b>                                     | I <sub>OL</sub> = 1.0 mA  | Total of pins = 57 mA  |                        | 0.4                 | V    |
| Pull-up resistor                     | R <sub>L1</sub>   |   |                           | 10                     | 30                     | 100                 | kΩ   |
| Pull-down resistor <sup>Note 8</sup> | R <sub>L2</sub>   |   |                           | 10                     | 30                     | 100                 | kΩ   |

**Notes 1.** P33, P36, P41, and PDL0 to PDL15 pins

**2.** P00 to P07, P10 to P17, P20 to P27, P30 to P32, P34, P35, P37, P40, P42 to P47,  $\overline{\text{RESET}}$ , and FLMD0 pins

**3.**  $\overline{\text{DRST}}$ , DDI, DCK, and DMS pins

**4.** P70 to P77 pins

**5.** AV<sub>DD0</sub> = AV<sub>DD1</sub> = AV<sub>DD2</sub> = EV<sub>DD0</sub> = EV<sub>DD1</sub> = EV<sub>DD2</sub>
**6.** Except for  $\overline{\text{DRST}}$  pin

**7.** P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, PDL0 to PDL15, and DDO pins

**8.**  $\overline{\text{DRST}}$  pin only

**Remark** The characteristics of alternate-function pins are the same as those of port pins.

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5$  to  $5.5$  V,  
 $V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V) (2/2)

| Parameter  | Symbol           | Conditions               | MIN.             | TYP. <sup>Note 1</sup> | MAX. | Unit |    |
|--|------------------|--------------------------|------------------|------------------------|------|------|----|
| V <sub>DD</sub> supply current <sup>Note 2</sup> | I <sub>DD1</sub> | f <sub>xx</sub> = 64 MHz | Normal operation |                        | 64   | 93   | mA |
|  | I <sub>DD2</sub> |                          | HALT mode        |                        | 42   | 60   | mA |
|  | I <sub>DD3</sub> |                          | IDLE mode        |                        | 5    | 10   | mA |
|  | I <sub>DD4</sub> | STOP mode                |                  | 40                     | 800  | μA   |    |

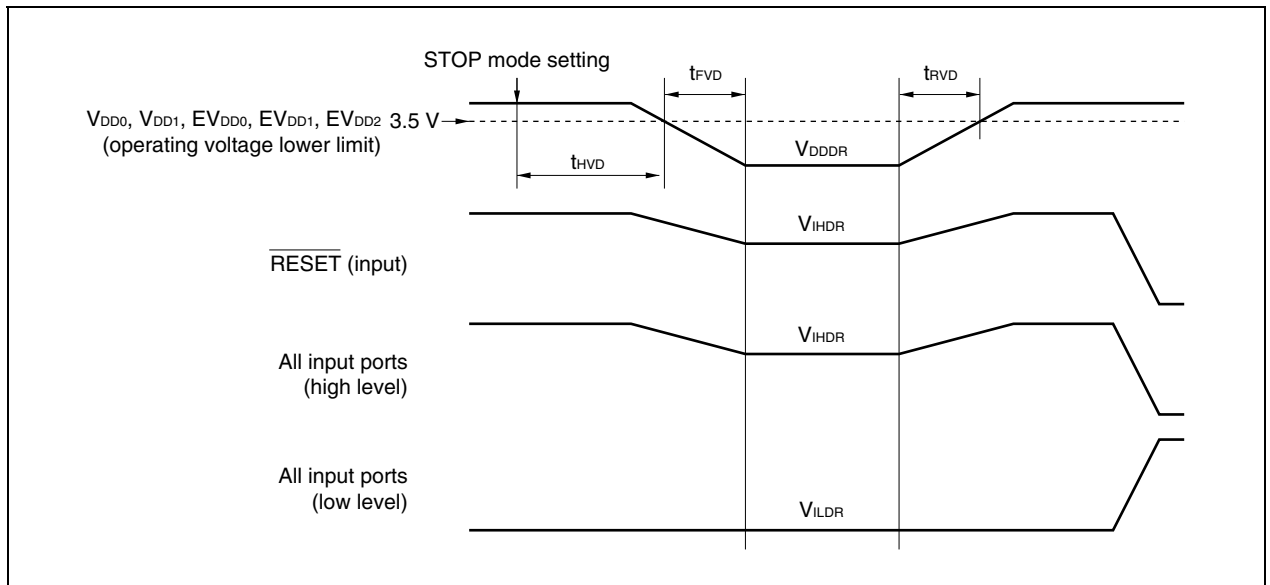
- Notes**
1. The TYP. value is a reference value when  $V_{DD0} = V_{DD1} = 5.0$  V and  $T_A = 25^\circ\text{C}$ .
  2. The current consumed by the EV<sub>DD</sub> system (output buffer and pull-up resistor) and the operating currents of A/D converters 0 to 2, the operational amplifier, and the comparator are not included.

28.2.7 Data retention characteristics

**STOP mode ( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V)**

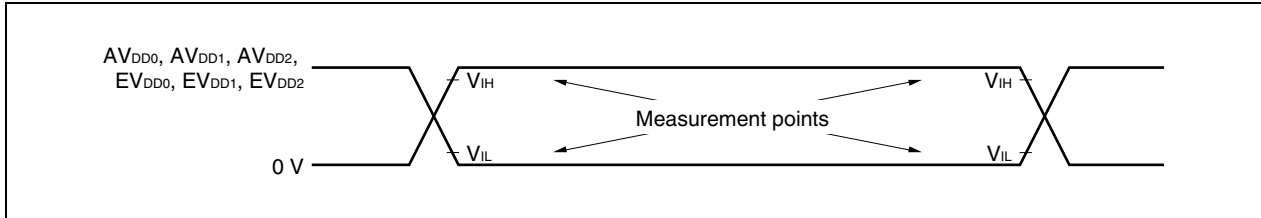
| Parameter   | Symbol     | Conditions                     | MIN.          | TYP. | MAX.          | Unit          |
|---|------------|--------------------------------|---------------|------|---------------|---------------|
| Data retention voltage                                    | $V_{DDDR}$ | STOP mode                      | <b>Note</b>   |      | 5.5           | V             |
| Data retention current                                    | $I_{DDDR}$ | $V_{DD0} = V_{DD1} = V_{DDDR}$ |               | 40   | 800           | $\mu\text{A}$ |
| Supply voltage rise time                                  | $t_{rVD}$  |                                | 1             |      |               | $\mu\text{s}$ |
| Supply voltage fall time                                  | $t_{fVD}$  |                                | 1             |      |               | $\mu\text{s}$ |
| Supply voltage retention time<br>(from STOP mode setting) | $t_{HVD}$  |                                | 0             |      |               | ms            |
| 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                | $EV_{SS}$     |      | $0.1V_{DDDR}$ | V             |

**Note** When the low-voltage detector (LVI) reset mode is not used (LVIM.LVIMD bit = 0): POC detection voltage ( $V_{POC0}$ )  
 When the low-voltage detector (LVI) reset mode is used (LVIM.LVIMD bit = 1): LVI detection voltage ( $V_{LV10}/V_{LV11}$ )

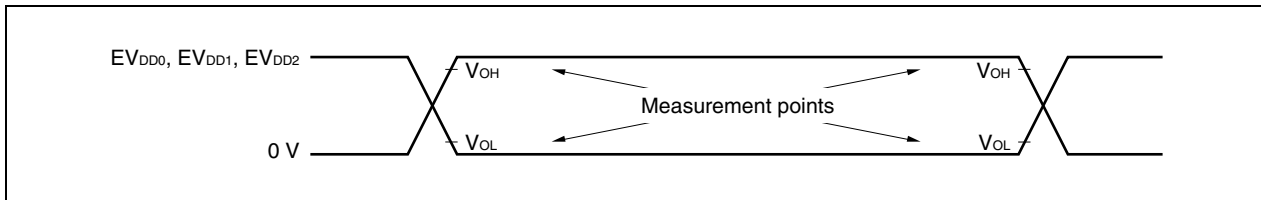


28.2.8 AC characteristics

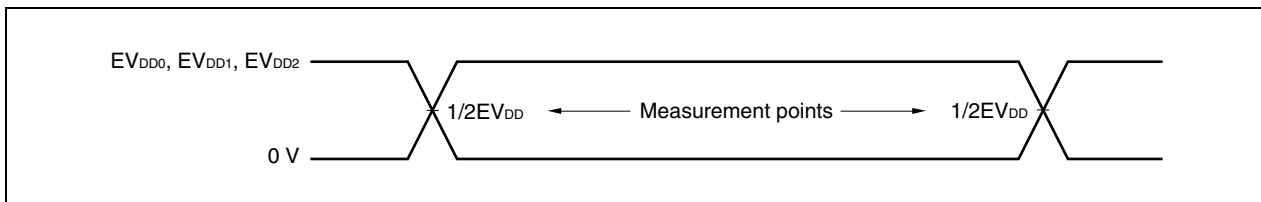
AC test input measurement points (External bus ( $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only), pins other than CSIB0 to CSIB2)



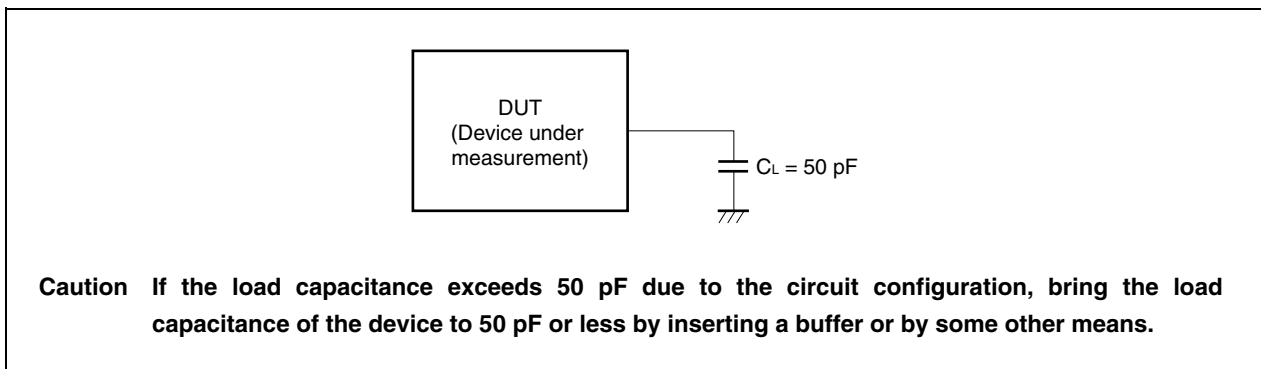
AC test output measurement points (External bus ( $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only), pins other than CSIB0 to CSIB2)



AC test I/O measurement points (External bus ( $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only), CSIB0 to CSIB2 pins)



Load conditions

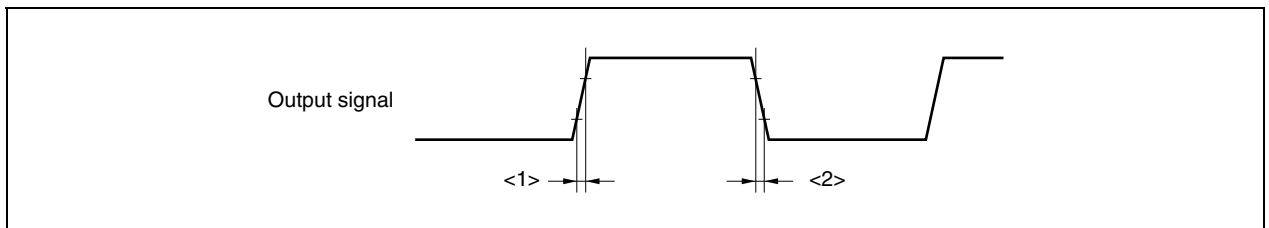


**(1) Output signal timing**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0$  to  $5.5$  V,

$V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V,  $C_L = 50$  pF)

| Parameter        | Symbol   | Conditions                  | MIN. | MAX. | Unit |
|------------------|----------|-----------------------------|------|------|------|
| Output rise time | $t_{OR}$ | <1> P07, PDL0 to PDL15, DDO |      | 8    | ns   |
|                  |          | Other than above            |      | 15   | ns   |
| Output fall time | $t_{OF}$ | <2> P07, PDL0 to PDL15, DDO |      | 8    | ns   |
|                  |          | Other than above            |      | 15   | ns   |



**(2) Reset, external interrupt timing**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0$  to  $5.5$  V,

$V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V,  $C_L = 50$  pF)

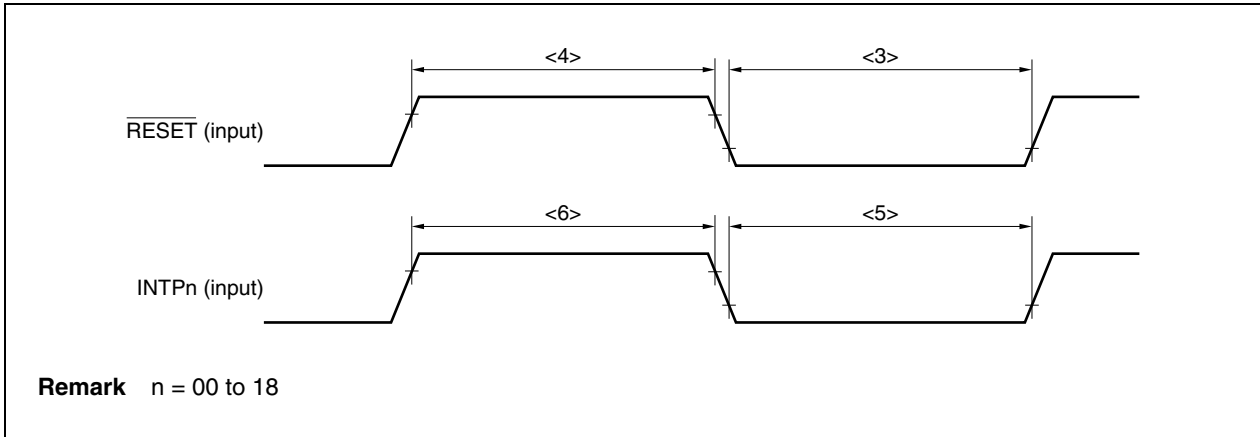
| Parameter              | Symbol     | Conditions  | MIN.           | MAX. | Unit |
|------------------------|------------|---|----------------|------|------|
| RESET low-level width  | $t_{WRSL}$ | <3> Power is on, STOP mode is released              | $500 + T_{OS}$ |      | ns   |
|                        |            | Other than above                                    | 500            |      | ns   |
| RESET high-level width | $t_{WRSH}$ | <4>   | 500            |      | ns   |
| INTPn low-level width  | $t_{WITL}$ | <5> n = 00 to 13, 17, 18 (analog noise elimination) | 500            |      | ns   |
|                        |            | n = 14 to 16 (digital noise elimination)            | $4T_{SMP}$     |      | ns   |
| INTPn high-level width | $t_{WITH}$ | <6> n = 00 to 13, 17, 18 (analog noise elimination) | 500            |      | ns   |
|                        |            | n = 14 to 16 (digital noise elimination)            | $4T_{SMP}$     |      | ns   |

**Remarks 1.**  $T_{OS}$ : Oscillation stabilization time

$T_{SMP}$ : Noise elimination sampling clock cycle (set by INTNFCn register)

- After reset release, a 1 ms oscillation stabilization time is internally secured when the oscillation frequency ( $f_x$ ) = 8 MHz. The oscillation stabilization time is therefore ( $T_{OS} + 1$ ) ms. After STOP mode release, an oscillation stabilization time half the value set to the OSTs register is internally secured. Therefore,  $T_{OS} = 0$  ns is acceptable if sufficient stabilization time can be secured by the OSTs register setting.

Reset/Interrupt

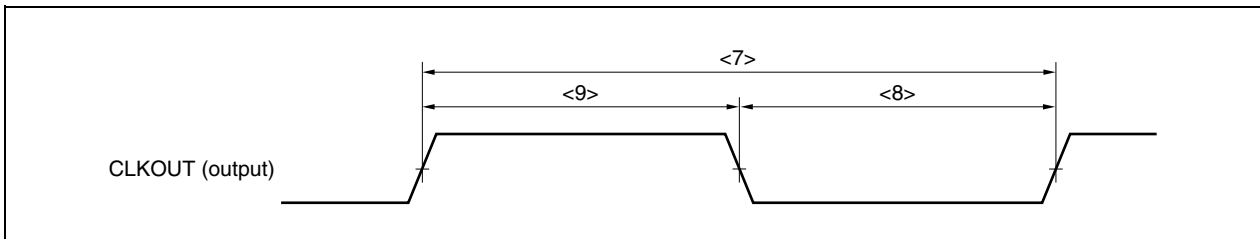


(3) CLKOUT output timing ( $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only)

( $T_A = -40$  to  $+85$  °C,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0$  to  $5.5$  V,

$V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V,  $C_L = 50$  pF)

| Parameter        | Symbol        | Conditions | MIN.              | MAX.      | Unit |
|------------------|---------------|------------|-------------------|-----------|------|
| Output cycle     | $t_{CYK}$ <7> |            | 31.25 ns          | 8 $\mu$ s |      |
| Low-level width  | $t_{WKH}$ <8> |            | $t_{CYK}/2 - 6.2$ |           | ns   |
| High-level width | $t_{WKL}$ <9> |            | $t_{CYK}/2 - 6.2$ |           | ns   |



(4) Bus timing ( $\mu$ PD70F3454GC-8EA-A and 70F3454F1-DA9-A only)

(a) Read cycle (CLKOUT asynchronous)

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0$  to  $5.5$  V,

$V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V,  $C_L = 50$  pF)

| Parameter   | Symbol       | Conditions | MIN.                               | MAX.                                | Unit |
|---|--------------|------------|------------------------------------|-------------------------------------|------|
| Delay time from address to ASTB $\downarrow$                          | $t_{DAST2}$  | <10>       | $(0.5 + WAS) T - 20$               |                                     | ns   |
| ASTB high-level width   | $t_{WSTH}$   | <11>       | $(1 + WAS + i) T - 17$             |                                     | ns   |
| Address hold time from ASTB $\downarrow$                              | $t_{HSTA}$   | <12>       | $(0.5 + WAH) T - 19$               |                                     | ns   |
| Address hold time from $\overline{RD}\uparrow$                        | $t_{HRDA2}$  | <13>       | $(1 + i) T - 29$                   |                                     | ns   |
| Delay time from address to $\overline{RD}\downarrow$                  | $t_{DARD2}$  | <14>       | $(1 + WAS + WAH) T - 36$           |                                     | ns   |
| Delay time from $\overline{RD}\downarrow$ to address float            | $t_{FRDA}$   | <15>       |                                    | 16                                  | ns   |
| Data input setup time from address                                    | $t_{DAID2}$  | <16>       |                                    | $(2 + WD + W + WAS + WAH) T - 37$   | ns   |
| Data input setup time from ASTB $\downarrow$                          | $t_{DSTID}$  | <17>       |                                    | $(1.5 + WD + W + WAH) T - 37$       | ns   |
| Data input setup time from $\overline{RD}\downarrow$                  | $t_{DRID2}$  | <18>       |                                    | $(1 + WD + w) T - 37$               | ns   |
| Delay time from ASTB $\downarrow$ to $\overline{RD}\downarrow$        | $t_{DSTRD3}$ | <19>       | $(0.5 + WAH) T - 16$               |                                     | ns   |
| Data input hold time (from $\overline{RD}\uparrow$ )                  | $t_{HRID2}$  | <20>       | 2                                  |                                     | ns   |
| Delay time from $\overline{RD}\uparrow$ to bus output                 | $t_{DRDOD2}$ | <21>       | $(1 + i) T - 19$                   |                                     | ns   |
| Delay time from $\overline{RD}\uparrow$ to ASTB $\uparrow$            | $t_{DRDST}$  | <22>       | $0.5T - 16$                        |                                     | ns   |
| $\overline{RD}$ low-level width                                       | $t_{WRDL2}$  | <23>       | $(1 + WD + w) T - 20$              |                                     | ns   |
| $\overline{RD}$ high-level width                                      | $t_{WRDH2}$  | <24>       | $(2 + i + WAS + WAH) T - 20$       |                                     | ns   |
| High-level hold time from $\overline{RD}\uparrow$ to $\overline{WRn}$ | $t_{HRDWR2}$ | <25>       | $(2 + i + WAS + WAH) T - 20$       |                                     | ns   |
| $\overline{WAIT}$ setup time (to address)                             | $t_{DAWT2}$  | <26>       |                                    | $(1.5 + WD + W + WAS + WAH) T - 45$ | ns   |
| $\overline{WAIT}$ hold time (from address)                            | $t_{HAWT2}$  | <27>       | $(1.5 + WD + W + WAS + WAH) T - 1$ |                                     | ns   |
| $\overline{WAIT}$ setup time (to ASTB $\downarrow$ )                  | $t_{DSTWT}$  | <28>       |                                    | $(1 + WD + W + WAH) T - 37$         | ns   |
| $\overline{WAIT}$ hold time (from ASTB $\downarrow$ )                 | $t_{HSTWT}$  | <29>       | $(1 + WD + W + WAH) T + 2$         |                                     | ns   |
| $\overline{WAIT}$ setup time (to $\overline{RD}\downarrow$ )          | $t_{DRDWT2}$ | <30>       |                                    | $(0.5 + WD + w) T - 37$             | ns   |
| $\overline{WAIT}$ hold time (from $\overline{RD}\downarrow$ )         | $t_{HRDWT2}$ | <31>       | $(0.5 + WD + w) T + 2$             |                                     | ns   |

**Cautions** 1. Set T in accordance with the following condition.

$$31.25 \text{ ns} \geq T$$

2. Be sure to insert the address setup waits and address hold waits.

**Remarks** 1. WAS: Number of address setup waits by the AWC register

WAH: Number of address hold waits by the AWC register

WD: Number of data waits by the DWCO register

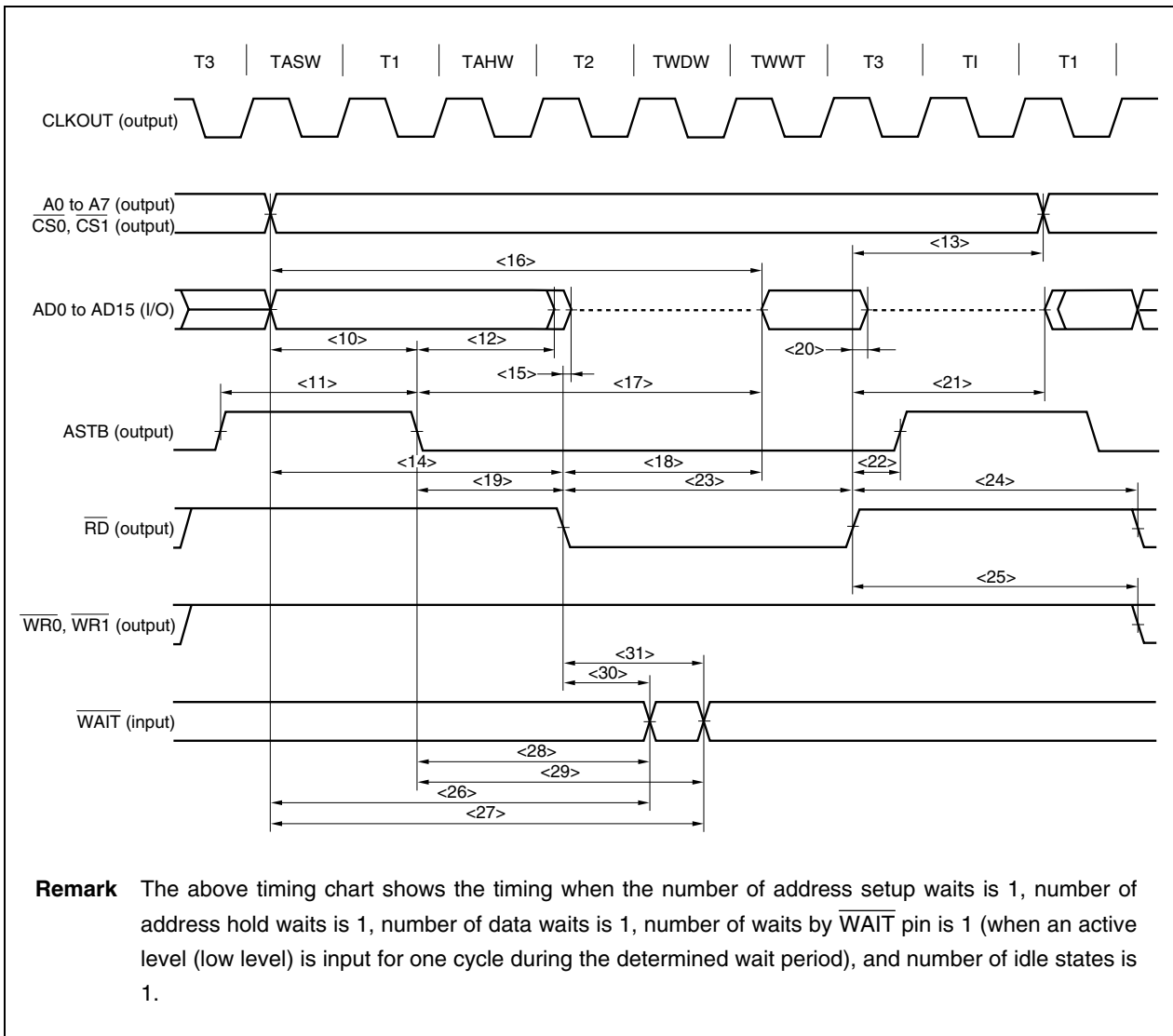
w: Number of external waits by the  $\overline{WAIT}$  pin

2.  $T = 1/f_{CPU}$  ( $f_{CPU}$ : CPU clock frequency)

3.  $n = 0, 1$

4. i: Number of idle states

Read cycle (CLKOUT asynchronous)





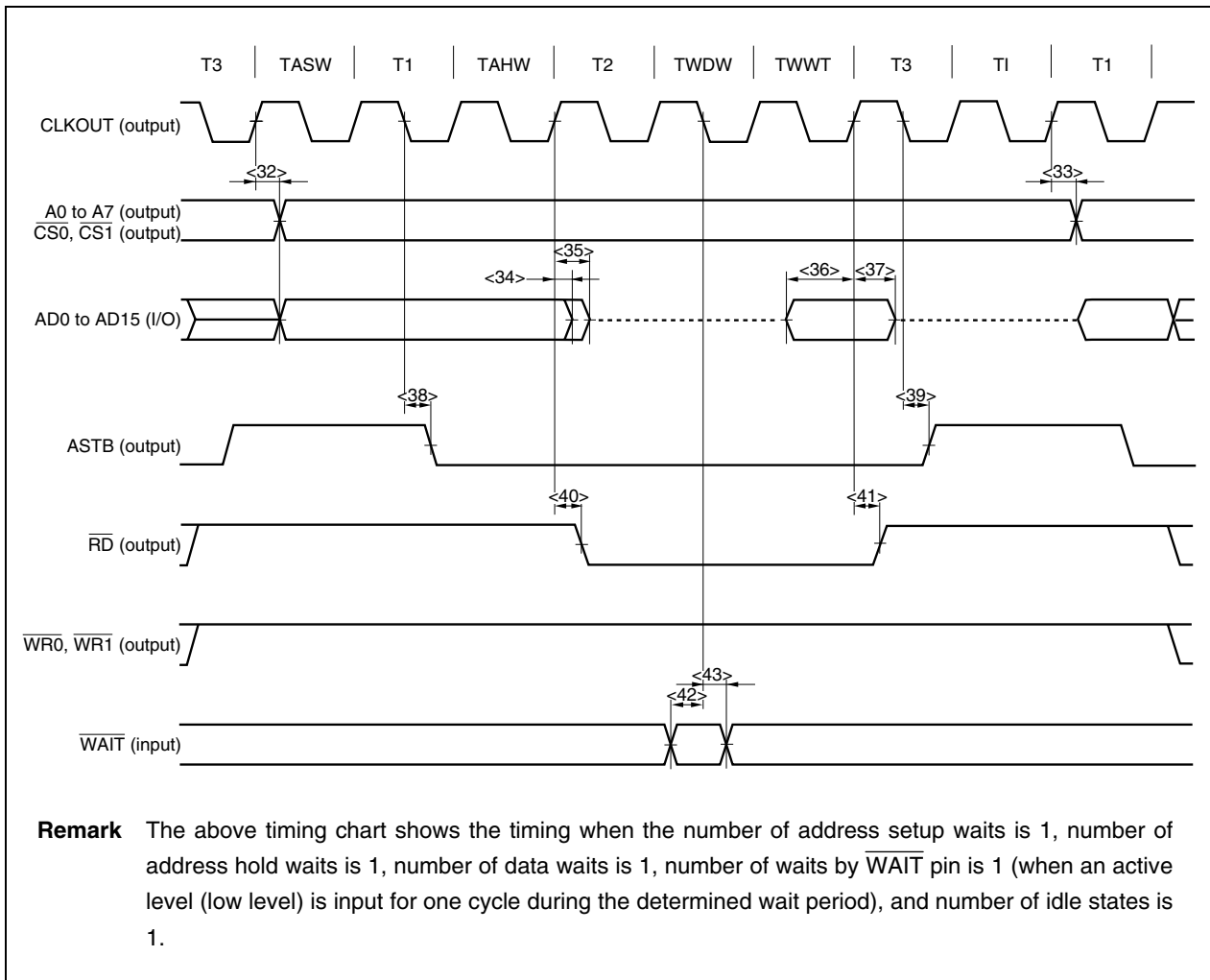
(b) Read cycle (CLKOUT synchronous)

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0$  to  $5.5$  V,  
 $V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V,  $C_L = 50$  pF)

| Parameter  | Symbol      | Conditions | MIN. | MAX. | Unit |
|--|-------------|------------|------|------|------|
| Delay time from CLKOUT $\uparrow$ to address                   | $t_{DKA2}$  | <32>       |      | 22   | ns   |
| Address hold time from CLKOUT $\uparrow$                       | $t_{HKA2}$  | <33>       | -5   |      | ns   |
| Address hold time from CLKOUT $\uparrow$                       | $t_{HKA3}$  | <34>       | -8   |      | ns   |
| Delay time from CLKOUT $\uparrow$ to address float             | $t_{FKA}$   | <35>       |      | 15   | ns   |
| Data input setup time (to CLKOUT $\uparrow$ )                  | $t_{SIDK2}$ | <36>       | 30   |      | ns   |
| Data input hold time (from CLKOUT $\uparrow$ )                 | $t_{HKID2}$ | <37>       | 9    |      | ns   |
| Delay time from CLKOUT $\downarrow$ to ASTB $\downarrow$       | $t_{DKST3}$ | <38>       | -8   | 18   | ns   |
| Delay time from CLKOUT $\downarrow$ to ASTB $\uparrow$         | $t_{DKST4}$ | <39>       | -8   | 18   | ns   |
| Delay time from CLKOUT $\uparrow$ to $\overline{RD}\downarrow$ | $t_{DKRD3}$ | <40>       | -10  | 17   | ns   |
| Delay time from CLKOUT $\uparrow$ to $\overline{RD}\uparrow$   | $t_{DKRD4}$ | <41>       | -10  | 17   | ns   |
| $\overline{WAIT}$ setup time (to CLKOUT $\downarrow$ )         | $t_{SWTK2}$ | <42>       | 30   |      | ns   |
| $\overline{WAIT}$ hold time (from CLKOUT $\downarrow$ )        | $t_{HKWT2}$ | <43>       | 9    |      | ns   |

**Caution** Be sure to insert the address setup waits and address hold waits.

Read cycle (CLKOUT synchronous)



**(c) Write cycle (CLKOUT asynchronous)**

$(T_A = -40 \text{ to } +85^\circ\text{C}, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 \text{ to } 5.5 \text{ V},$   
 $V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 \text{ V}, C_L = 50 \text{ pF})$

| Parameter   | Symbol       | Conditions | MIN.                                      | MAX.                                       | Unit |
|---|--------------|------------|---|--|------|
| Delay time from address to $\overline{\text{ASTB}}\downarrow$                           | $t_{DAST2}$  | <10>       | $(0.5 + W_{AS}) T - 20$                   |  | ns   |
| ASTB high-level width   | $t_{WSTH}$   | <11>       | $(1 + W_{AS} + i) T - 16$                 |  | ns   |
| Address hold time from $\overline{\text{ASTB}}\downarrow$                               | $t_{HSTA}$   | <12>       | $(0.5 + W_{AH}) T - 19$                   |  | ns   |
| Address hold time from $\overline{\text{WRn}}\uparrow$                                  | $t_{HWRA2}$  | <44>       | $T - 19$                                  |  | ns   |
| Delay time from address to $\overline{\text{WRn}}\downarrow$                            | $t_{DAWR2}$  | <45>       | $(1 + W_{AS} + W_{AH}) T - 36$            |  | ns   |
| Delay time from $\overline{\text{WRn}}\downarrow$ to data output                        | $t_{DWROD3}$ | <46>       |   | 15   | ns   |
| Delay time from $\overline{\text{ASTB}}\downarrow$ to $\overline{\text{WRn}}\downarrow$ | $t_{DSTWR3}$ | <47>       | $(0.5 + W_{AH}) T - 16$                   |  | ns   |
| Delay time from data output to $\overline{\text{WRn}}\uparrow$                          | $t_{DODWR2}$ | <48>       | $(1 + W_D + w) T - 25$                    |  | ns   |
| Data output hold time from $\overline{\text{WRn}}\uparrow$                              | $t_{HWROD2}$ | <49>       | $T - 19$                                  |  | ns   |
| Delay time from $\overline{\text{WRn}}\uparrow$ to $\overline{\text{ASTB}}\uparrow$     | $t_{DWRST}$  | <50>       | $0.5T - 16$                               |  | ns   |
| $\overline{\text{WRn}}$ low-level width   | $t_{WWRL2}$  | <51>       | $(1 + W_D + w) T - 20$                    |  | ns   |
| $\overline{\text{WRn}}$ high-level width  | $t_{WWRH2}$  | <52>       | $(2 + W_{AS} + W_{AH}) T - 20$            |  | ns   |
| High-level hold time from $\overline{\text{WRn}}\uparrow$ to $\overline{\text{RD}}$     | $t_{HWRRD2}$ | <53>       | $(2 + W_{AS} + W_{AH}) T - 20$            |  | ns   |
| $\overline{\text{WAIT}}$ setup time (to address)  | $t_{DAWT2}$  | <26>       |   | $(1.5 + W_D + w + W_{AS} + W_{AH}) T - 45$ | ns   |
| $\overline{\text{WAIT}}$ hold time (from address)                                       | $t_{HAWT2}$  | <27>       | $(1.5 + W_D + w + W_{AS} + W_{AH}) T - 1$ |  | ns   |
| $\overline{\text{WAIT}}$ setup time (to $\overline{\text{ASTB}}\downarrow$ )            | $t_{DSTWT}$  | <28>       |   | $(1 + W_D + w + W_{AH}) T - 37$            | ns   |
| $\overline{\text{WAIT}}$ hold time (from $\overline{\text{ASTB}}\downarrow$ )           | $t_{HSTWT}$  | <29>       | $(1 + W_D + w + W_{AH}) T + 2$            |  | ns   |
| $\overline{\text{WAIT}}$ setup time (to $\overline{\text{WRn}}\downarrow$ )             | $t_{DWRWT2}$ | <54>       |   | $(0.5 + W_D + w) T - 37$                   | ns   |
| $\overline{\text{WAIT}}$ hold time (from $\overline{\text{WRn}}\downarrow$ )            | $t_{HWRWT2}$ | <55>       | $(0.5 + W_D + w) T + 2$                   |  | ns   |

**Cautions** 1. Set T in accordance with the following condition.

$$31.25 \text{ ns} \geq T$$

2. Be sure to insert the address setup waits and address hold waits.

**Remarks** 1.  $W_{AS}$ : Number of address setup waits by the AWC register

$W_{AH}$ : Number of address hold waits by the AWC register

$W_D$ : Number of data waits by the DWC0 register

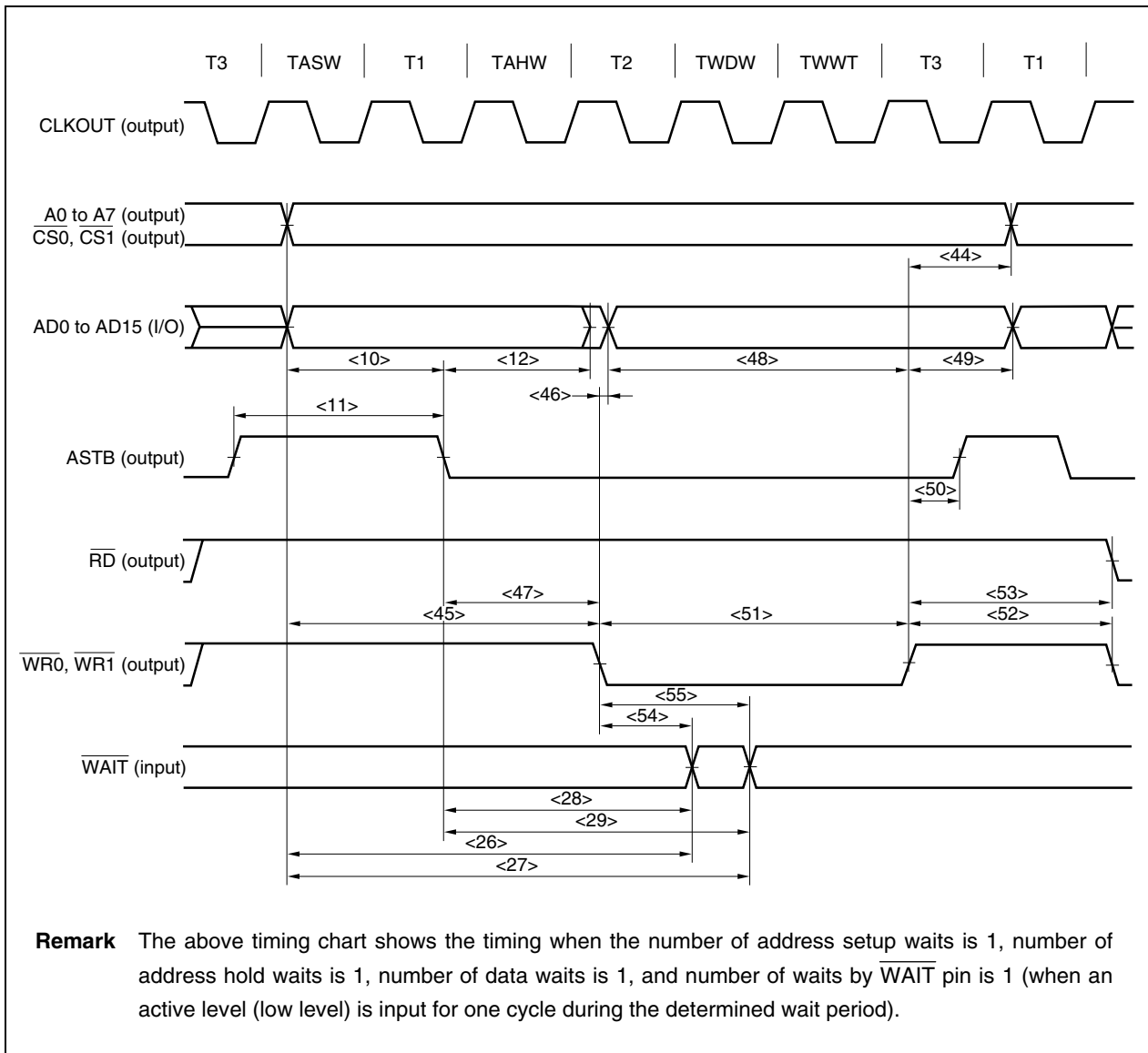
$w$ : Number of external waits by the  $\overline{\text{WAIT}}$  pin

2.  $T = 1/f_{\text{CPU}}$  ( $f_{\text{CPU}}$ : CPU operating clock frequency)

3.  $n = 0, 1$

4.  $i$ : Number of idle states

Write cycle (CLKOUT asynchronous)



(d) Write cycle (CLKOUT synchronous)

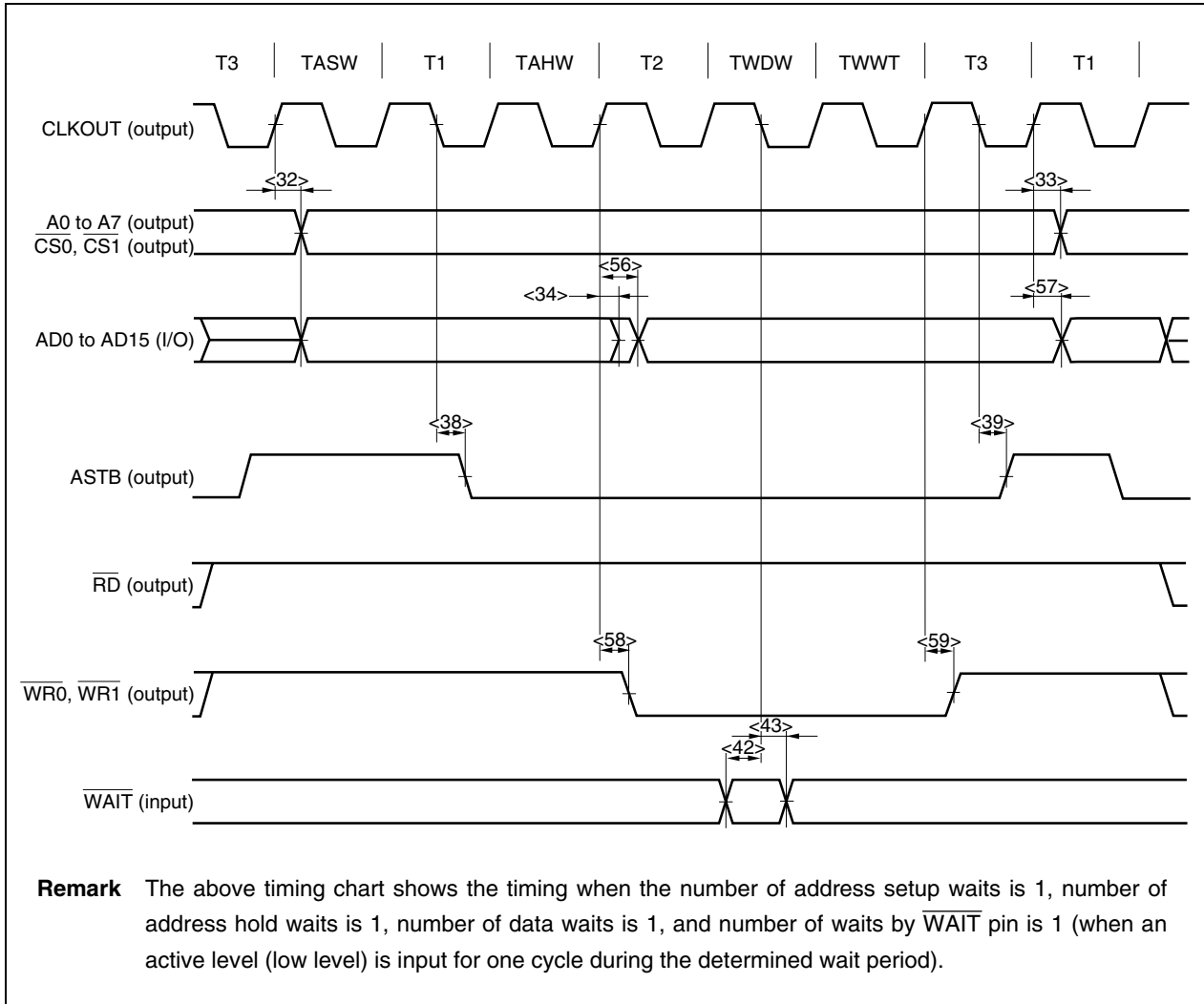
( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0$  to  $5.5$  V,  
 $V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V,  $C_L = 50$  pF)

| Parameter   | Symbol      | Conditions | MIN. | MAX. | Unit |
|---|-------------|------------|------|------|------|
| Delay time from CLKOUT $\uparrow$ to address                    | $t_{DKA2}$  | <32>       |      | 22   | ns   |
| Address hold time from CLKOUT $\uparrow$                        | $t_{HKA2}$  | <33>       | -5   |      | ns   |
| Address hold time from CLKOUT $\uparrow$                        | $t_{HKA3}$  | <34>       | -8   |      | ns   |
| Delay time from CLKOUT $\downarrow$ to ASTB $\downarrow$        | $t_{DKST3}$ | <38>       | -8   | 18   | ns   |
| Delay time from CLKOUT $\downarrow$ to ASTB $\uparrow$          | $t_{DKST4}$ | <39>       | -8   | 18   | ns   |
| Delay time from CLKOUT $\uparrow$ to data output                | $t_{DKOD3}$ | <56>       |      | 22   | ns   |
| Data output hold time from CLKOUT $\uparrow$                    | $t_{HKOD2}$ | <57>       | -9   |      | ns   |
| Delay time from CLKOUT $\uparrow$ to $\overline{WRn}\downarrow$ | $t_{DKWR3}$ | <58>       | -10  | 17   | ns   |
| Delay time from CLKOUT $\uparrow$ to $\overline{WRn}\uparrow$   | $t_{DKWR4}$ | <59>       | -10  | 17   | ns   |
| $\overline{WAIT}$ setup time (to CLKOUT $\downarrow$ )          | $t_{SWTK2}$ | <42>       | 30   |      | ns   |
| $\overline{WAIT}$ hold time (from CLKOUT $\downarrow$ )         | $t_{HKWT2}$ | <43>       | 9    |      | ns   |

**Caution** Be sure to insert the address setup waits and address hold waits.

**Remark** n = 0, 1

Write cycle (CLKOUT synchronous)



(5) Timer Timing

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0$  to  $5.5$  V,  $V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V,  $C_L = 50$  pF)

| Parameter   | Symbol              | Conditions                       | MIN.                    | MAX. | Unit |
|---|---------------------|----------------------------------|-------------------------|------|------|
| T <sub>In</sub> high-level width <sup>Notes 1, 2</sup>  | t <sub>WTIHn</sub>  | <60> n = B00 to B03, B10 to B13  | 12T + 10                |      | ns   |
|   |                     | n = A20, A21, A30, A31, A40, A41 | 3T <sub>smp1</sub> + 10 |      | ns   |
| T <sub>In</sub> low-level width <sup>Notes 1, 2</sup>   | t <sub>WTILn</sub>  | <61> n = B00 to B03, B10 to B13  | 12T + 10                |      | ns   |
|   |                     | n = A20, A21, A30, A31, A40, A41 | 3T <sub>smp1</sub> + 10 |      | ns   |
| EVTBm high-level width <sup>Note 1</sup>                | t <sub>WEVBHm</sub> | <62> m = 0, 1                    | 12T + 10                |      | ns   |
| EVTBm low-level width <sup>Note 1</sup>                 | t <sub>WEVBLm</sub> | <63> m = 0, 1                    | 12T + 10                |      | ns   |
| TRGBm high-level width <sup>Note 1</sup>                | t <sub>WTRHm</sub>  | <64> m = 0, 1                    | 12T + 10                |      | ns   |
| TRGBm low-level width <sup>Note 1</sup>                 | t <sub>WTRLm</sub>  | <65> m = 0, 1                    | 12T + 10                |      | ns   |
| TENCm0/TENCm1 high-level width <sup>Note 3</sup>        | t <sub>WENCHm</sub> | <66> m = 0, 1                    | 3T <sub>smp2</sub> + 10 |      | ns   |
| TENCm0/TENCm1 low-level width <sup>Note 3</sup>         | t <sub>WENCLm</sub> | <67> m = 0, 1                    | 3T <sub>smp2</sub> + 10 |      | ns   |
| TECRm high-level width <sup>Note 3</sup>                | t <sub>WCRHm</sub>  | <68> m = 0, 1                    | 3T <sub>smp2</sub> + 10 |      | ns   |
| TECRm low-level width <sup>Note 3</sup>                 | t <sub>WCRLm</sub>  | <69> m = 0, 1                    | 3T <sub>smp2</sub> + 10 |      | ns   |
| TITm0/TITm1 high-level width <sup>Note 3</sup>          | t <sub>WTITHm</sub> | <70> m = 0, 1                    | 3T <sub>smp2</sub> + 10 |      | ns   |
| TITm0/TITm1 low-level width <sup>Note 3</sup>           | t <sub>WTITLm</sub> | <71> m = 0, 1                    | 3T <sub>smp2</sub> + 10 |      | ns   |
| EVTm high-level width <sup>Note 3</sup>                 | t <sub>WEVTHm</sub> | <72> m = 0, 1                    | 3T <sub>smp2</sub> + 10 |      | ns   |
| EVTm low-level width <sup>Note 3</sup>                  | t <sub>WEVTLm</sub> | <73> m = 0, 1                    | 3T <sub>smp2</sub> + 10 |      | ns   |
| TENCm0/TENCm1 input time differential <sup>Note 3</sup> | t <sub>PHUDm</sub>  | <74> m = 0, 1                    | 3T <sub>smp2</sub> + 10 |      | ns   |

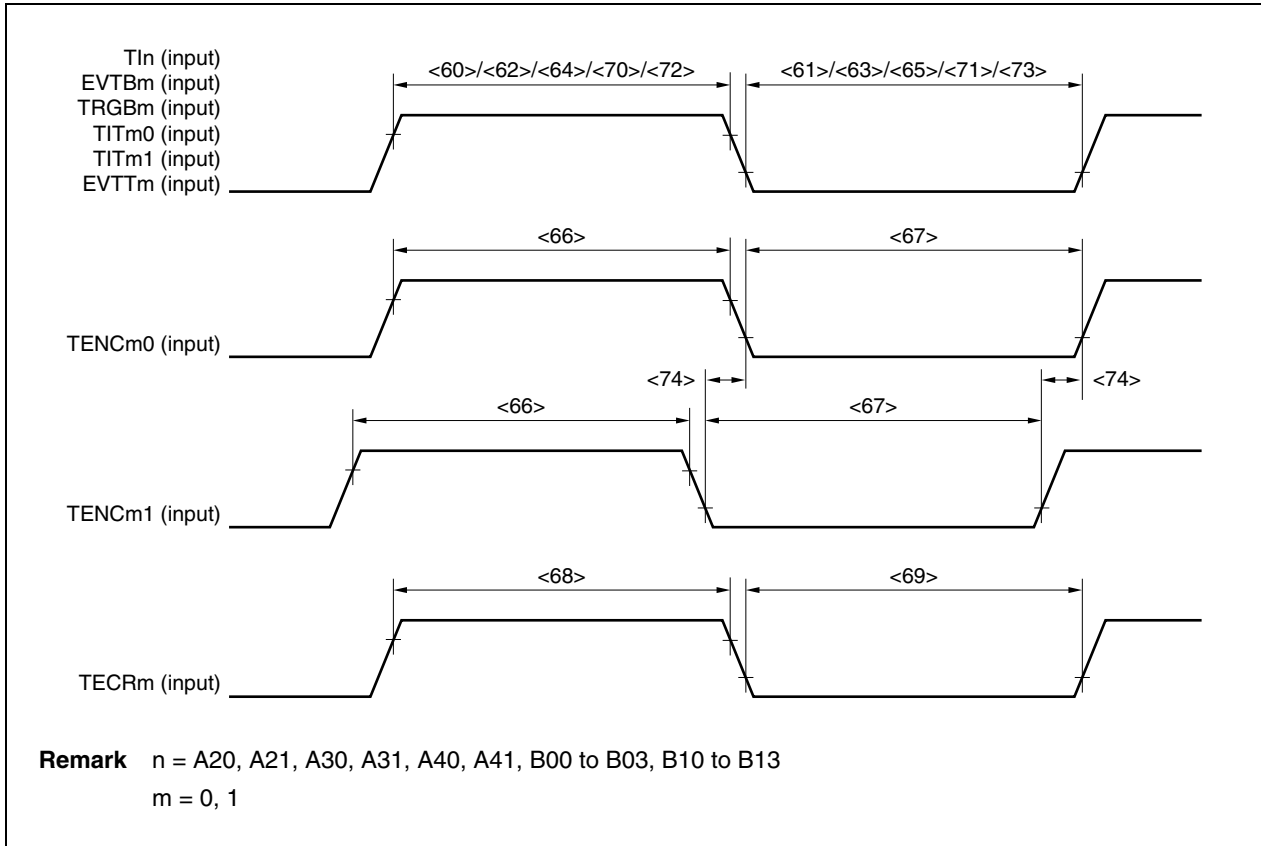
**Notes** 1.  $T = 1/f_{xx}$

2. T<sub>smp1</sub>: Noise elimination sampling clock cycle (set by TANFC2 to TANFC4 registers)

3. T<sub>smp2</sub>: Noise elimination sampling clock cycle (set by TTNFC0 and TTNFC1 registers)

**Remark** The above specification shows a pulse width that is accurately detected as a valid edge. Even if a pulse narrower than the above specification is input, therefore, it may be detected as a valid edge.

Timer Input Timing





**(6) CSIB Timing**
**(a) Master mode**

$(T_A = -40 \text{ to } +85^\circ\text{C}, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 \text{ to } 5.5 \text{ V},$   
 $V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 \text{ V}, C_L = 50 \text{ pF})$

| Parameter   | Symbol                                  | Conditions | MIN.                     | MAX.                     | Unit |
|---|---|------------|--------------------------|--------------------------|------|
| $\overline{\text{SCKBn}}$ cycle   | $t_{\text{KCYM}}$                       | <75>       | 125                      |                          | ns   |
| $\overline{\text{SCKBn}}$ high-/low-level width   | $t_{\text{KWHM}},$<br>$t_{\text{KWLM}}$ | <76>       | $t_{\text{KCYM}}/2 - 10$ |                          | ns   |
| SI $\overline{\text{Bn}}$ setup time (to $\overline{\text{SCKBn}}\uparrow$ )            | $t_{\text{SSIM}}$                       | <77>       | 30                       |                          | ns   |
| SI $\overline{\text{Bn}}$ setup time (to $\overline{\text{SCKBn}}\downarrow$ )          |   |            | 30                       |                          | ns   |
| SI $\overline{\text{Bn}}$ hold time (from $\overline{\text{SCKBn}}\uparrow$ )           | $t_{\text{HSIM}}$                       | <78>       | 30                       |                          | ns   |
| SI $\overline{\text{Bn}}$ hold time (from $\overline{\text{SCKBn}}\downarrow$ )         |   |            | 30                       |                          | ns   |
| SO $\overline{\text{Bn}}$ output delay time (from $\overline{\text{SCKBn}}\downarrow$ ) | $t_{\text{DSOM}}$                       | <79>       |                          | 30                       | ns   |
| SO $\overline{\text{Bn}}$ output delay time (from $\overline{\text{SCKBn}}\uparrow$ )   |   |            |                          | 30                       | ns   |
| SO $\overline{\text{Bn}}$ output hold time (from $\overline{\text{SCKBn}}\uparrow$ )    | $t_{\text{HSOM}}$                       | <80>       |                          | $t_{\text{KCYM}}/2 - 10$ | ns   |
| SO $\overline{\text{Bn}}$ output hold time (from $\overline{\text{SCKBn}}\downarrow$ )  |   |            |                          | $t_{\text{KCYM}}/2 - 10$ | ns   |

**Remark** n = 0 to 2

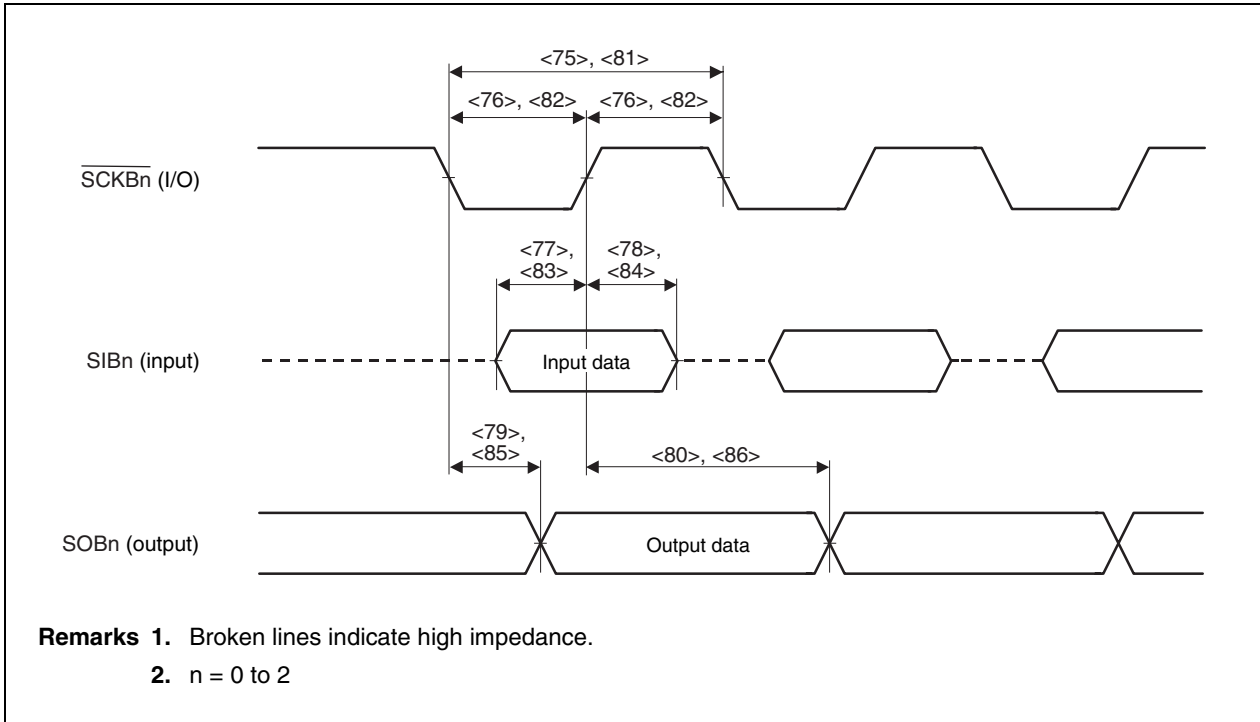
**(b) Slave mode**

$(T_A = -40 \text{ to } +85^\circ\text{C}, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 \text{ to } 5.5 \text{ V},$   
 $V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 \text{ V}, C_L = 50 \text{ pF})$

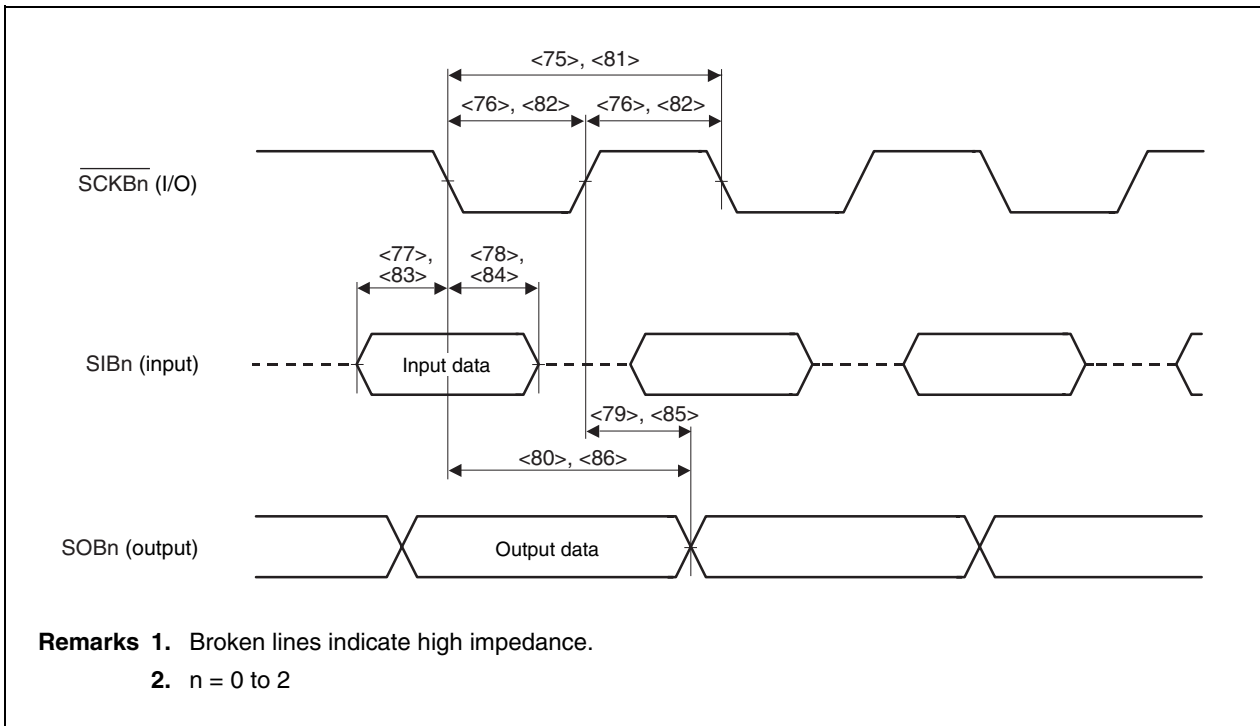
| Parameter   | Symbol                                  | Conditions | MIN.                     | MAX.                     | Unit |
|---|---|------------|--------------------------|--------------------------|------|
| $\overline{\text{SCKBn}}$ cycle   | $t_{\text{KCYs}}$                       | <81>       | 125                      |                          | ns   |
| $\overline{\text{SCKBn}}$ high-/low-level width   | $t_{\text{KWHs}},$<br>$t_{\text{KWLS}}$ | <82>       | $t_{\text{KCYs}}/2 - 10$ |                          | ns   |
| SI $\overline{\text{Bn}}$ setup time (to $\overline{\text{SCKBn}}\uparrow$ )            | $t_{\text{SSIS}}$                       | <83>       | 30                       |                          | ns   |
| SI $\overline{\text{Bn}}$ setup time (to $\overline{\text{SCKBn}}\downarrow$ )          |   |            | 30                       |                          | ns   |
| SI $\overline{\text{Bn}}$ hold time (from $\overline{\text{SCKBn}}\uparrow$ )           | $t_{\text{HSIS}}$                       | <84>       | 30                       |                          | ns   |
| SI $\overline{\text{Bn}}$ hold time (from $\overline{\text{SCKBn}}\downarrow$ )         |   |            | 30                       |                          | ns   |
| SO $\overline{\text{Bn}}$ output delay time (from $\overline{\text{SCKBn}}\downarrow$ ) | $t_{\text{DSOS}}$                       | <85>       |                          | 30                       | ns   |
| SO $\overline{\text{Bn}}$ output delay time (from $\overline{\text{SCKBn}}\uparrow$ )   |   |            |                          | 30                       | ns   |
| SO $\overline{\text{Bn}}$ output hold time (from $\overline{\text{SCKBn}}\uparrow$ )    | $t_{\text{HSOS}}$                       | <86>       |                          | $t_{\text{KCYs}}/2 - 10$ | ns   |
| SO $\overline{\text{Bn}}$ output hold time (from $\overline{\text{SCKBn}}\downarrow$ )  |   |            |                          | $t_{\text{KCYs}}/2 - 10$ | ns   |

**Remark** n = 0 to 2

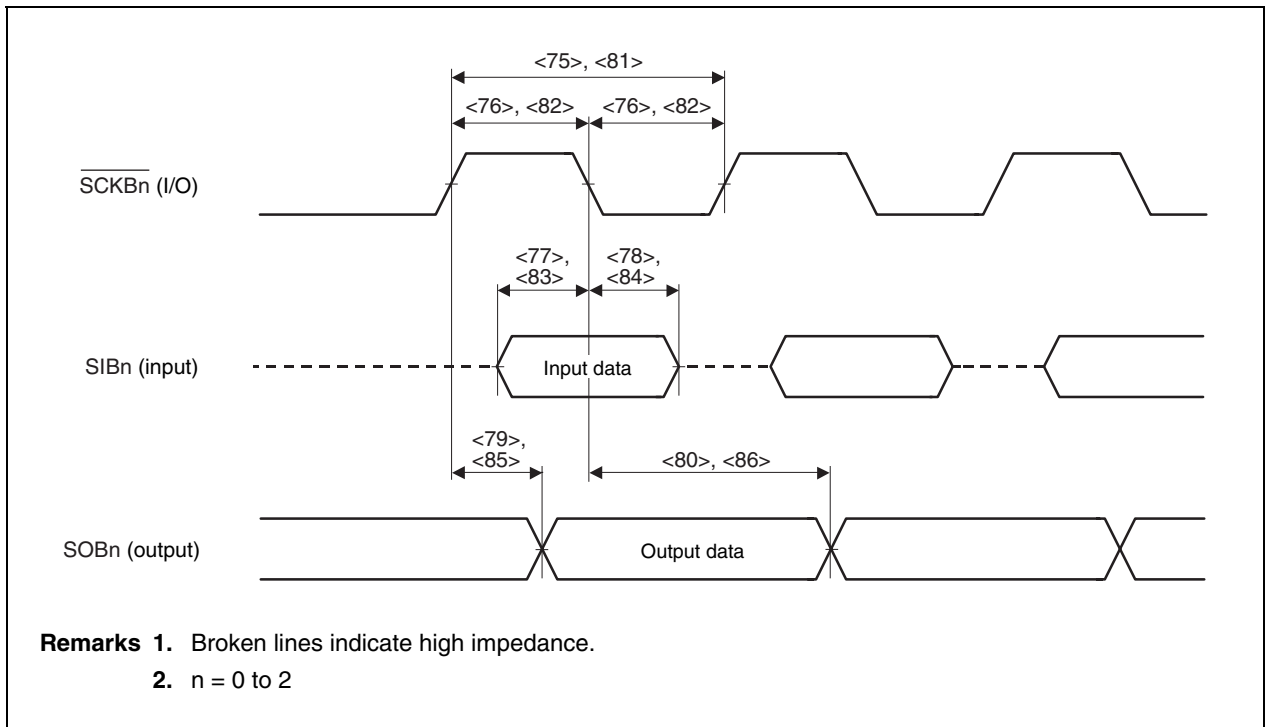
CSIB timing when CBnCKP and CBnDAP bits of CBnCTL1 register = 00



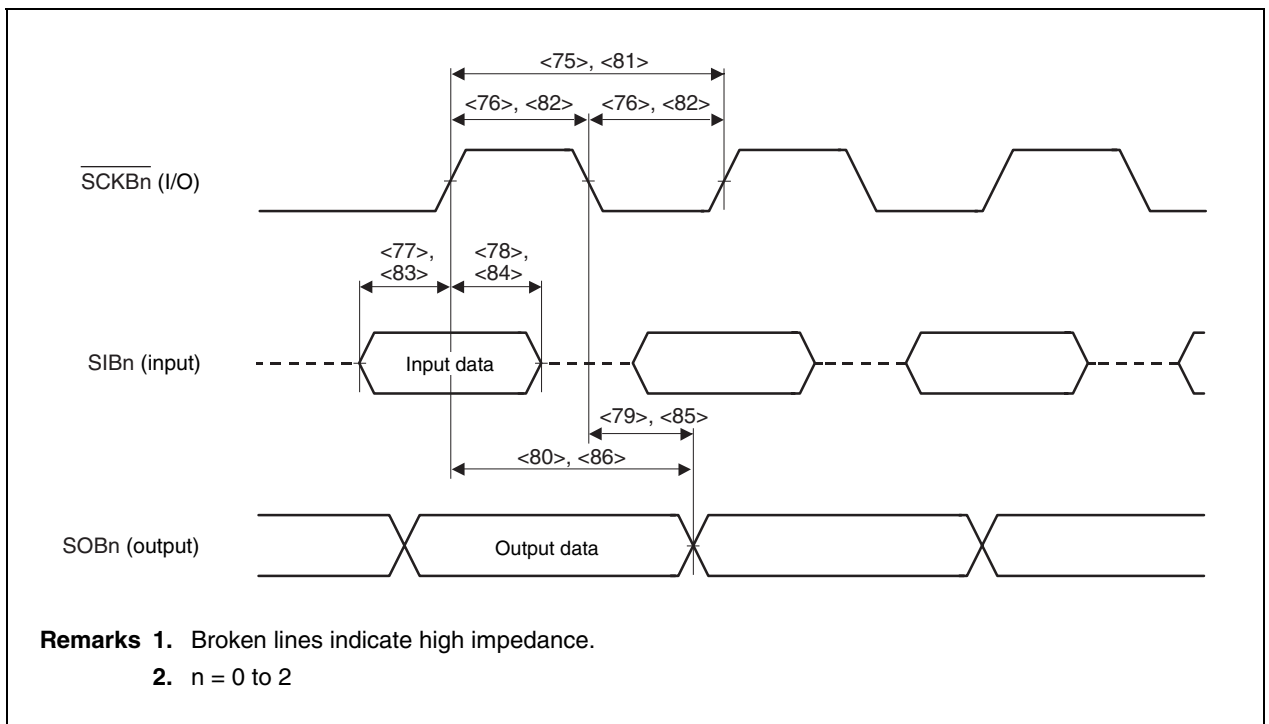
CSIB timing when CBnCKP and CBnDAP bits of CBnCTL1 register = 01



CSIB timing when CBnCKP and CBnDAP bits of CBnCTL1 register = 10



CSIB timing when CBnCKP and CBnDAP bits of CBnCTL1 register = 11



**(7) I<sup>2</sup>C bus timing**
**(T<sub>A</sub> = -40 to +85°C, V<sub>DD0</sub> = V<sub>DD1</sub> = EV<sub>DD0</sub> = EV<sub>DD1</sub> = EV<sub>DD2</sub> = AV<sub>DD0</sub> = AV<sub>DD1</sub> = AV<sub>DD2</sub> = 4.0 to 5.5 V,**
**V<sub>SS0</sub> = V<sub>SS1</sub> = EV<sub>SS0</sub> = EV<sub>SS1</sub> = EV<sub>SS2</sub> = AV<sub>SS0</sub> = AV<sub>SS1</sub> = AV<sub>SS2</sub> = 0 V, C<sub>L</sub> = 50 pF)**

| Parameter  | Symbol                 |                     | Standard Mode |                     | High-Speed Mode              |                     | Unit                  |
|--|------------------------|---------------------|---------------|---------------------|------------------------------|---------------------|-----------------------|
|  |                        |                     | MIN.          | MAX.                | MIN.                         | MAX.                |                       |
| SCL clock frequency  | f <sub>CLK</sub>       | –                   | 0             | 100                 | 0                            | 400                 | kHz                   |
| Bus free time (between stop condition and start condition) | t <sub>BUF</sub>       | <87>                | 4.7           | –                   | 1.3                          | –                   | μs                    |
| Hold time <sup>Note 1</sup>                                | t <sub>HD:STA</sub>    | <88>                | 4.0           | –                   | 0.6                          | –                   | μs                    |
| SCL clock low-level width                                  | t <sub>LOW</sub>       | <89>                | 4.7           | –                   | 1.3                          | –                   | μs                    |
| SCL clock high-level width                                 | t <sub>HIGH</sub>      | <90>                | 4.0           | –                   | 0.6                          | –                   | μs                    |
| Start/restart condition setup time                         | t <sub>SU:STA</sub>    | <91>                | 4.7           | –                   | 0.6                          | –                   | μs                    |
| Data hold time   | CBUS-compatible master | t <sub>HD:DAT</sub> | <92>          | 5.0                 | –                            | –                   | μs                    |
|  | I <sup>2</sup> C mode  |                     |               | 0 <sup>Note 2</sup> | –                            | 0 <sup>Note 2</sup> | 0.9 <sup>Note 3</sup> |
| Data setup time  | t <sub>SU:DAT</sub>    | <93>                | 250           | –                   | 100 <sup>Note 4</sup>        | –                   | ns                    |
| SDA, SCL signal rise time                                  | t <sub>R</sub>         | <94>                | –             | 1000                | 20 + 0.1Cb <sup>Note 5</sup> | 300                 | ns                    |
| SDA, SCL signal fall time                                  | t <sub>F</sub>         | <95>                | –             | 300                 | 20 + 0.1Cb <sup>Note 5</sup> | 300                 | ns                    |
| Stop condition setup time                                  | t <sub>SU:STO</sub>    | <96>                | 4.0           | –                   | 0.6                          | –                   | μs                    |
| Pulse width of spike suppressed by input filter            | t <sub>SP</sub>        | <97>                | –             | –                   | 0                            | 50                  | ns                    |
| Each bus line capacitive load                              | C <sub>b</sub>         | –                   | –             | 400                 | –                            | 400                 | pF                    |

**Notes** 1. The first clock pulse is generated after a hold time during the start condition.

 2. The system must internally supply a hold time of at least 300 ns for the SDA signal (at V<sub>IHmin.</sub> of SCL signal) to fill the undefined area at the falling edge of SCL.

 3. If the system does not extend the low hold time (t<sub>LOW</sub>) of the SCL signal, the maximum data hold time (t<sub>HD:DAT</sub>) must be satisfied.

 4. The high-speed mode I<sup>2</sup>C bus can be used in the standard mode I<sup>2</sup>C bus system. In this case, make sure that the following conditions are satisfied.

- If system does not extend the low status hold time of the SCL signal

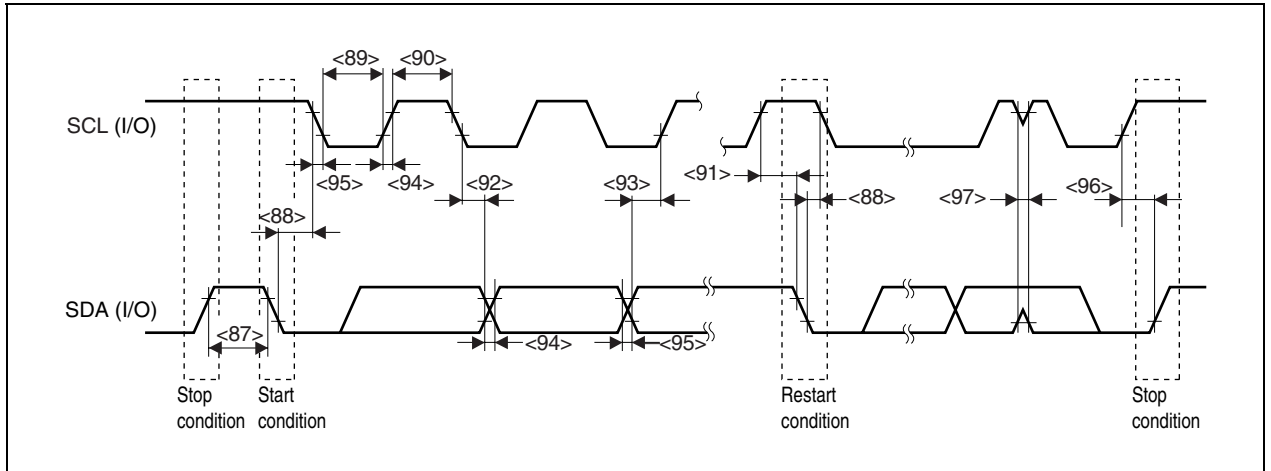
$$t_{SU:DAT} \geq 250 \text{ ns}$$

- If system extends the low status hold time of SCL signal

Sends the next data bit to the SDA line before the SCL line is released (t<sub>Rmax.</sub> + t<sub>SU:DAT</sub> = 1000 + 250 = 1250 ns: standard mode I<sup>2</sup>C bus specification).

 5. C<sub>b</sub>: Total capacitance of one bus line (unit: pF)

I<sup>2</sup>C bus timing



(8) High-impedance control timing

(T<sub>A</sub> = -40 to +85°C, V<sub>DD0</sub> = V<sub>DD1</sub> = EV<sub>DD0</sub> = EV<sub>DD1</sub> = EV<sub>DD2</sub> = AV<sub>DD0</sub> = AV<sub>DD1</sub> = AV<sub>DD2</sub> = 4.0 to 5.5 V,

V<sub>SS0</sub> = V<sub>SS1</sub> = EV<sub>SS0</sub> = EV<sub>SS1</sub> = EV<sub>SS2</sub> = AV<sub>SS0</sub> = AV<sub>SS1</sub> = AV<sub>SS2</sub> = 0 V, C<sub>L</sub> = 50 pF)

| Parameter   | Symbol            | Conditions                      | MIN. | MAX. | Unit |
|---|-------------------|---------------------------------|------|------|------|
| Oscillation stop → timer output high impedance                    | t <sub>CLM</sub>  | When clock monitor is operating |      | 65   | μs   |
| Input to TOBnOFF → timer output high impedance                    | t <sub>HTQn</sub> |                                 |      | 300  | ns   |
| Input to TOAmOFF → timer output high impedance                    | t <sub>HTPm</sub> |                                 |      | 300  | ns   |
| Input to ANI00/ANI05 → timer output high impedance                | t <sub>ANI0</sub> |                                 |      | 10   | μs   |
| Input to ANI10/ANI15 to ANI12/ANI17 → timer output high impedance | t <sub>ANI1</sub> |                                 |      | 10   | μs   |

**Remark** n = 0, 1  
m = 2, 3

**28.2.9 Characteristics of A/D converters 0, 1**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = AV_{REFP0} = AV_{REFP1} = 4.0$  to  $5.5$  V,  $V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V,  $C_L = 50$  pF)

| Parameter                                      | Symbol     | Conditions   | MIN.      | TYP. | MAX.      | Unit          |
|--|------------|--|-----------|------|-----------|---------------|
| Resolution                                     |            |  | 12        | 12   | 12        | bit           |
| Overall error <sup>Note 1</sup>                |            |  |           |      | $\pm 10$  | LSB           |
| Conversion time                                | $t_{CONV}$ | $f_{AD01} = 16$ MHz, ADAnCTC register = 0BH or 0CH | 2.0       |      |           | $\mu\text{s}$ |
|  |            | $f_{AD01} = 12$ MHz, ADAnCTC register = 00H        |           |      | 7.42      | $\mu\text{s}$ |
| Zero scale error <sup>Note 1</sup>             |            |  |           |      | $\pm 10$  | LSB           |
| Full-scale error <sup>Note 1</sup>             |            |  |           |      | $\pm 10$  | LSB           |
| Integral linearity error <sup>Note 1</sup>     |            |  |           |      | $\pm 4$   | LSB           |
| Differential linearity error <sup>Note 1</sup> |            |  |           |      | $\pm 2.5$ | LSB           |
| Analog reference voltage                       | $AV_{DD}$  |  | 4.0       |      | 5.5       | V             |
| Analog input voltage                           | $V_{IAN}$  |  | $AV_{SS}$ |      | $AV_{DD}$ | V             |
| $AV_{DD}$ supply current <sup>Note 2</sup>     | $AI_{DD}$  | During operation                                   |           | 4.5  | 7.5       | mA            |
|  | $AI_{DSS}$ | In STOP mode <sup>Note 3</sup>                     |           | 3.5  | 17.5      | $\mu\text{A}$ |

- Notes**
1. Excludes quantization error ( $\pm 0.5$  LSB).
  2. This value is for only one A/D converter (A/D converter 0 or 1).
  3. Stop the operation of A/D converters 0 and 1 (ADnSCM.ADnCE bit = 0) before setting STOP mode.

- Remarks**
1. LSB: Least Significant Bit
  2.  $f_{AD01}$ : Base clock of A/D converters 0 and 1
  3.  $n = 0, 1$

**28.2.10 Characteristics of A/D converter 2**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0$  to  $5.5$  V,  
 $V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V,  $C_L = 50$  pF)

| Parameter                                      | Symbol     | Conditions                     | MIN.      | TYP. | MAX.      | Unit          |
|--|------------|--------------------------------|-----------|------|-----------|---------------|
| Resolution                                     |            |                                | 10        | 10   | 10        | bit           |
| Overall error <sup>Note 1</sup>                |            |                                |           |      | $\pm 4.0$ | LSB           |
| Conversion time                                | $t_{CONV}$ |                                | 3.88      |      | 10        | $\mu\text{s}$ |
| Zero scale error <sup>Note 1</sup>             |            |                                |           |      | $\pm 4.0$ | LSB           |
| Full-scale error <sup>Note 1</sup>             |            |                                |           |      | $\pm 4.0$ | LSB           |
| Integral linearity error <sup>Note 1</sup>     |            |                                |           |      | $\pm 4.0$ | LSB           |
| Differential linearity error <sup>Note 1</sup> |            |                                |           |      | $\pm 2.0$ | LSB           |
| Analog reference voltage                       | $AV_{DD}$  |                                | 4.0       |      | 5.5       | V             |
| Analog input voltage                           | $V_{IAN}$  |                                | $AV_{SS}$ |      | $AV_{DD}$ | V             |
| $AV_{DD}$ supply current                       | $AI_{DD}$  | During operation               |           | 3.5  | 7         | mA            |
|  | $AI_{DDs}$ | In STOP mode <sup>Note 2</sup> |           | 1    | 10        | $\mu\text{A}$ |

- Notes**
1. Excludes quantization error ( $\pm 0.5\text{LSB}$ ).
  2. Stop the operation of A/D converter 2 (AD2M0.AD2CE bit = 0) before setting STOP mode.

**Remark** LSB: Least Significant Bit

**28.2.11 Operational amplifier characteristics**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0$  to  $5.5$  V,

$V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V,  $C_L = 50$  pF)

| Parameter                           | Symbol     | Conditions                     | MIN.                  | TYP.      | MAX.           | Unit             |   |
|-------------------------------------|------------|--------------------------------|-----------------------|-----------|----------------|------------------|---|
| Input offset voltage                | $V_{IO}$   |                                |                       | $\pm 9.0$ |                | mV               |   |
| Input voltage range                 | $V_I$      | Gain = 2.500                   | $0.04AV_{DD}$         |           | $0.36AV_{DD}$  | V                |   |
|                                     |            | Gain = 5.000                   | $0.02AV_{DD}$         |           | $0.18AV_{DD}$  | V                |   |
|                                     |            | Gain = 10.00                   | $0.01AV_{DD}$         |           | $0.085AV_{DD}$ | V                |   |
| Slew rate <sup>Note 1</sup>         | $S_R$      |                                | 10                    | 15        |                | V/ $\mu\text{s}$ |   |
| Gain error                          |            | <b>Note 2</b>                  | Gain = 2.500 to 4.444 |           | $\pm 1.0$      | $\pm 1.3$        | % |
|                                     |            |                                | Gain = 5.000 to 6.667 |           | $\pm 1.0$      | $\pm 1.5$        | % |
|                                     |            |                                | Gain = 8.000, 10.00   |           | $\pm 1.0$      | $\pm 1.7$        | % |
|                                     |            | <b>Note 3</b>                  | Gain = 2.500 to 4.444 |           | $\pm 1.0$      | $\pm 2.0$        | % |
|                                     |            |                                | Gain = 5.000 to 6.667 |           | $\pm 1.0$      | $\pm 2.1$        | % |
|                                     |            |                                | Gain = 8.000, 10.00   |           | $\pm 1.0$      | $\pm 2.2$        | % |
| Operating current <sup>Note 4</sup> | $I_{OPDD}$ | During operation               |                       | 1.8       | 2.6            | mA               |   |
|                                     | $A_{IDDS}$ | In STOP mode <sup>Note 5</sup> |                       | 1.0       | 10             | $\mu\text{A}$    |   |

- Notes**
1. Inclination characteristic of 10% to 90% of output voltage
  2.  $AV_{DD0} = AV_{DD1} = 4.5$  to  $5.5$  V
  3.  $AV_{DD0} = AV_{DD1} = 4.0$  to  $5.5$  V
  4. Four operational amplifiers are provided in total. The value shows the operating current per operational amplifier.
  5. Stop operational amplifier operation (OP0CTL0.OP0EN bit = 0, OP1CTL0.OP12EN, OP11EN, and OP10EN bits = 0) before setting STOP mode.

**Remark** Power supplies  $AV_{DD0}$  and  $AV_{DD1}$  are used for the operational amplifier.



**28.2.12 Comparator characteristics**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0$  to  $5.5$  V,

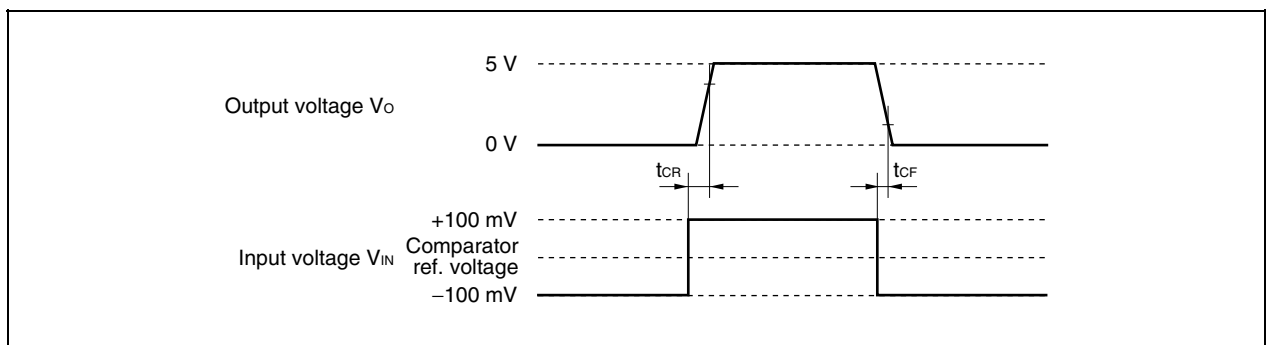
$V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V,  $C_L = 50$  pF)

| Parameter                                 | Symbol     | Conditions  | MIN.                | TYP.      | MAX.                | Unit          |
|---|------------|---|---------------------|-----------|---------------------|---------------|
| Input offset voltage                      | $V_{IO}$   |   |                     | $\pm 3.0$ |                     | mV            |
| Input voltage range                       | $V_I$      |   | $AV_{SS}$           |           | $AV_{DD}$           | V             |
| Comparator reference voltage (full range) | CREFnF     |   | $0.02AV_{DD} + 0.1$ |           | $0.92AV_{DD} - 0.1$ | V             |
| Comparator reference voltage (low range)  | CREFnL     |   | $0.02AV_{DD} + 0.1$ |           | $0.5AV_{DD} - 0.1$  | V             |
| Response time                             | $t_{CR}$   | Input amplitude = 100 mV, at rising edge <sup>Note 1</sup>  |                     | 1.0       |                     | $\mu\text{s}$ |
|   | $t_{CF}$   | Input amplitude = 100 mV, at falling edge <sup>Note 2</sup> |                     | 1.0       |                     | $\mu\text{s}$ |
| Operating current <sup>Note 3</sup>       | $I_{CPDD}$ | During operation  |                     |           | 250                 | $\mu\text{A}$ |
|   | $A_{IDDS}$ | In STOP mode <sup>Note 4</sup>                              |                     | 2.0       | 20                  | nA            |

- Notes**
1. Characteristics of pulse response when ANIm input changes from the comparator reference voltage – 100 mV to the comparator reference voltage + 100 mV
  2. Characteristics of pulse response when ANIm input changes from the comparator reference voltage + 100 mV to the comparator reference voltage – 100 mV
  3. Four comparators are provided in total. The value shows the operating current per comparator.
  4. Stop comparator operation (CMPnCTL0 register = 00H) before setting STOP mode.

- Remarks**
1. Power supplies for the comparators are  $AV_{DD0}$  and  $AV_{DD1}$ .
  2.  $m = 05, 15$  to  $17$   
 $n = 0, 1$

**Comparator Characteristics**



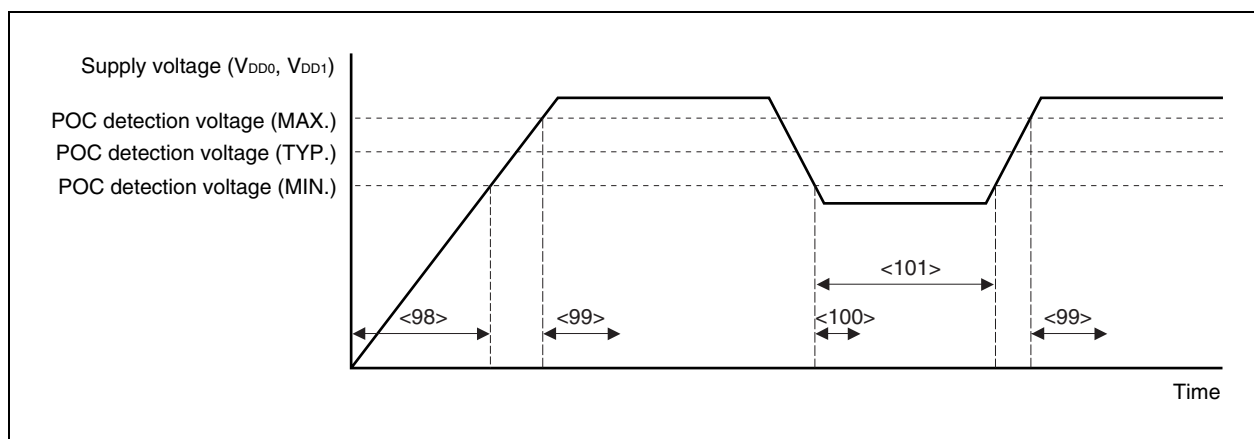
**28.2.13 Power-on-clear circuit (POC)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5$  to  $5.5$  V,

$V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V,  $C_L = 50$  pF)

| Parameter                           | Symbol     | Conditions  | MIN.              | TYP. | MAX.  | Unit |
|-------------------------------------|------------|---|-------------------|------|-------|------|
| POC detection voltage               | $V_{POC0}$ |   | 3.5               | 3.7  | 3.9   | V    |
| Supply voltage rise time            | $t_{PTH}$  | <98> $V_{DD0}, V_{DD1} = 0$ to $3.5$ V                                | $2.5 \mu\text{s}$ |      | 1.8 s |      |
| Response time 1 <sup>Note 1</sup>   | $t_{PTH0}$ | <99> After $V_{DD0}$ and $V_{DD1}$ reach $3.9$ V on power application |                   |      | 3.0   | ms   |
| Response time 2 <sup>Note 2</sup>   | $t_{PD}$   | <100> After $V_{DD0}$ and $V_{DD1}$ drop to $3.5$ V on power off      |                   |      | 1.0   | ms   |
| Minimum width of $V_{DD0}, V_{DD1}$ | $t_{PW}$   | <101>   | 0.2               |      |       | ms   |

- Notes**
1. The time required to release a reset signal (POCRES) after the POC detection voltage is detected.
  2. The time required to output a reset signal (POCRES) after the POC detection voltage is detected.



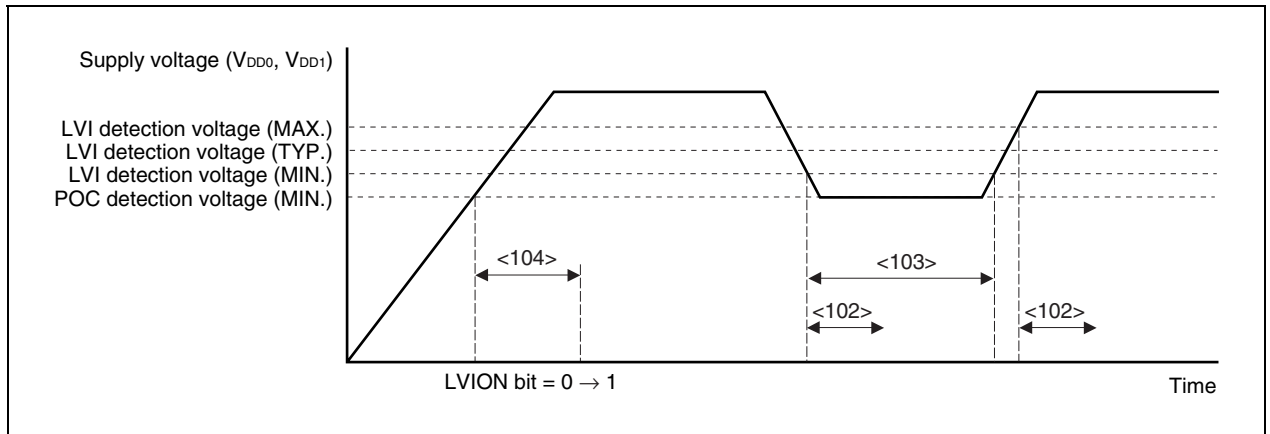
**28.2.14 Low-voltage detector (LVI)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5$  to  $5.5$  V,

$V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V,  $C_L = 50$  pF)

| Parameter                                 | Symbol      | Conditions   | MIN. | TYP. | MAX. | Unit |
|---|-------------|--|------|------|------|------|
| LVI detection voltage                     | $V_{LVI0}$  | LVIS.LVIS0 bit = 0   | 4.2  | 4.4  | 4.6  | V    |
|   | $V_{LVI1}$  | LVIS.LVIS0 bit = 1   | 4.0  | 4.2  | 4.4  | V    |
| Response time 1 <sup>Note</sup>           | $t_{LD}$    | <102> After $V_{DD0}$ and $V_{DD1}$ reach $V_{LVI0}/V_{LVI1}$ (MAX.) or drop to $V_{LVI0}/V_{LVI1}$ (MIN.)           |      | 0.2  | 2.0  | ms   |
| Minimum width of $V_{DD0}$ , $V_{DD1}$    | $t_{LW}$    | <103>  | 0.2  |      |      | ms   |
| Reference voltage stabilization wait time | $t_{LWAIT}$ | <104> After $V_{DD0}$ and $V_{DD1}$ reach POC detection voltage (MIN.) and the LVIM.LVION bit is changed from 0 to 1 |      | 0.1  |      | ms   |

**Note** The time required to output an interrupt request signal (INTLVIL, INTLVIH) or internal reset signal (LVIRE) after the LVI detection voltage is detected.



**28.2.15 Flash memory programming characteristics**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5$  to  $5.5$  V,

$V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0$  V,  $C_L = 50$  pF)

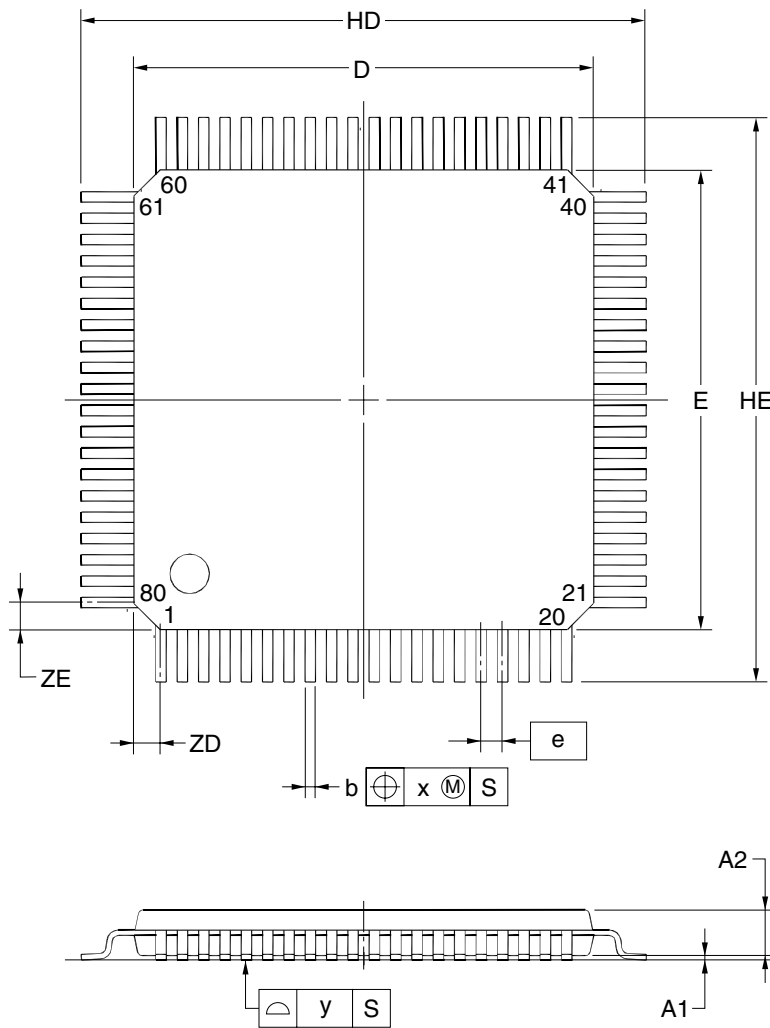
| Parameter     | Symbol     | Conditions  | MIN. | TYP. | MAX. | Unit  |
|---------------|------------|-------------|------|------|------|-------|
| Rewrite count | $C_{ERWR}$ | <b>Note</b> |      | 100  |      | Times |

**Note** Rewrite as follows.

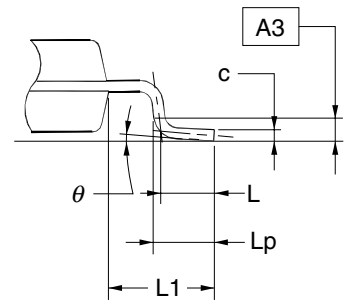
Example when three rewrites: Shipped product  $\rightarrow E \rightarrow P \rightarrow E \rightarrow P \rightarrow E \rightarrow P$  (P: Write, E: Erase)

CHAPTER 29 PACKAGE DRAWINGS

80-PIN PLASTIC LQFP(14x14)



detail of lead end



(UNIT:mm)

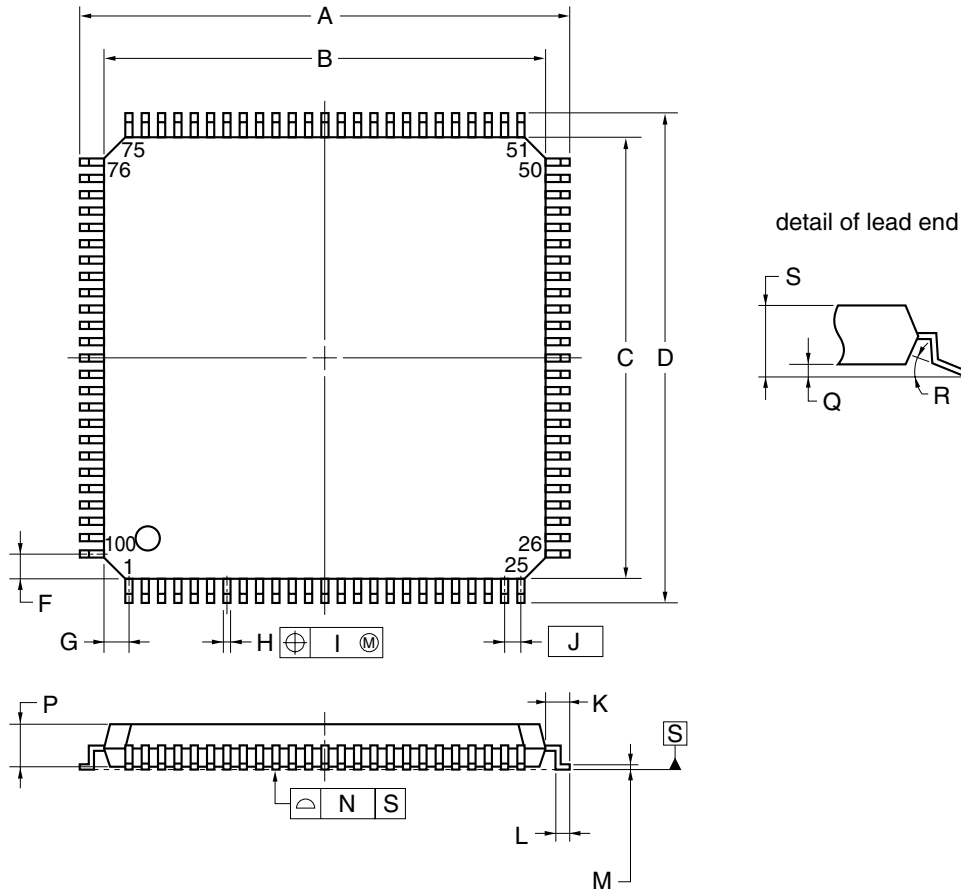
| ITEM     | DIMENSIONS                             |
|----------|--|
| D        | 14.00±0.20                             |
| E        | 14.00±0.20                             |
| HD       | 17.20±0.20                             |
| HE       | 17.20±0.20                             |
| A        | 1.70 MAX.                              |
| A1       | 0.125±0.075                            |
| A2       | 1.40±0.05                              |
| A3       | 0.25                                   |
| b        | 0.32±0.06                              |
| c        | 0.17 <sup>+0.03</sup> <sub>-0.06</sub> |
| L        | 0.80                                   |
| Lp       | 0.886±0.15                             |
| L1       | 1.60±0.20                              |
| $\theta$ | 3° <sup>+5°</sup> <sub>-3°</sub>       |
| e        | 0.65                                   |
| x        | 0.13                                   |
| y        | 0.10                                   |
| ZD       | 0.825                                  |
| ZE       | 0.825                                  |

P80GC-65-UBT

**NOTE**

Each lead centerline is located within 0.13 mm of its true position at maximum material condition.

100-PIN PLASTIC LQFP (FINE PITCH) (14x14)



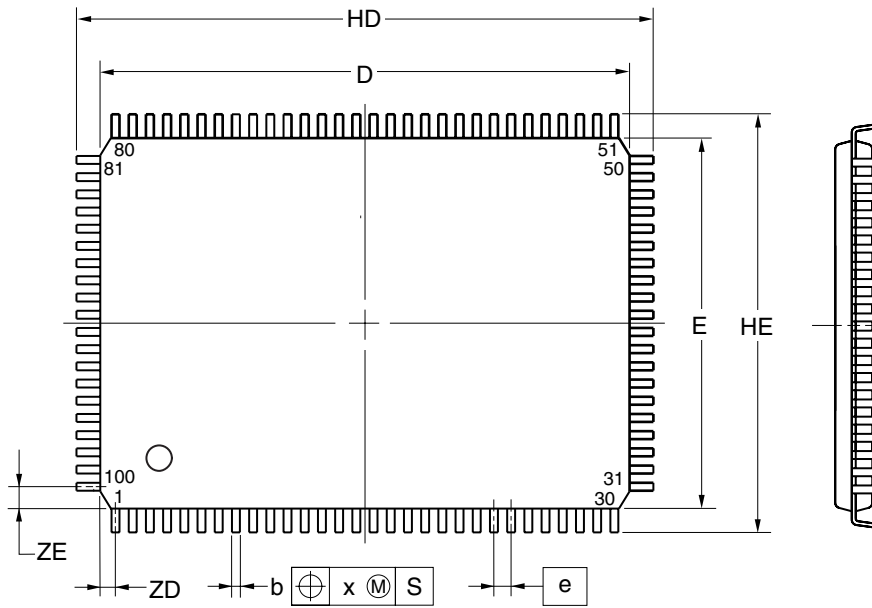
**NOTE**

Each lead centerline is located within 0.08 mm of its true position (T.P.) at maximum material condition.

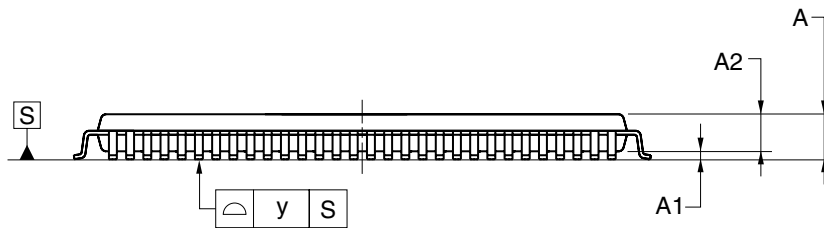
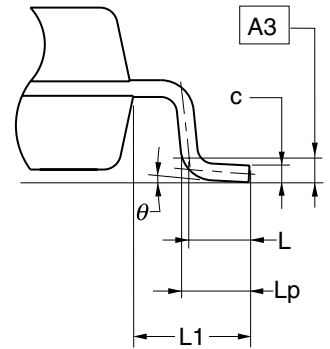
| ITEM | MILLIMETERS                            |
|------|--|
| A    | 16.00±0.20                             |
| B    | 14.00±0.20                             |
| C    | 14.00±0.20                             |
| D    | 16.00±0.20                             |
| F    | 1.00                                   |
| G    | 1.00                                   |
| H    | 0.22 <sup>+0.05</sup> <sub>-0.04</sub> |
| I    | 0.08                                   |
| J    | 0.50 (T.P.)                            |
| K    | 1.00±0.20                              |
| L    | 0.50±0.20                              |
| M    | 0.17 <sup>+0.03</sup> <sub>-0.07</sub> |
| N    | 0.08                                   |
| P    | 1.40±0.05                              |
| Q    | 0.10±0.05                              |
| R    | 3° <sup>+7°</sup> <sub>-3°</sub>       |
| S    | 1.60 MAX.                              |

**S100GC-50-8EU, 8EA-2**

100-PIN PLASTIC LQFP (14x20)



detail of lead end



(UNIT:mm)

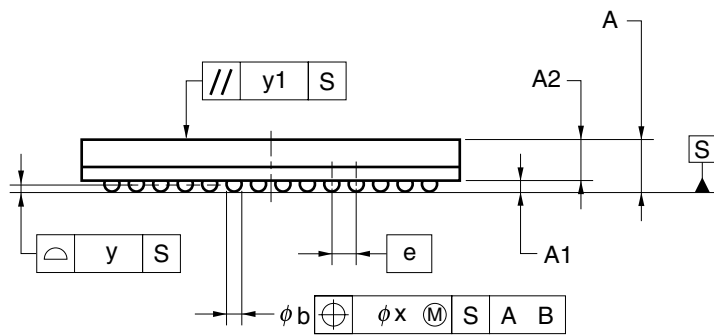
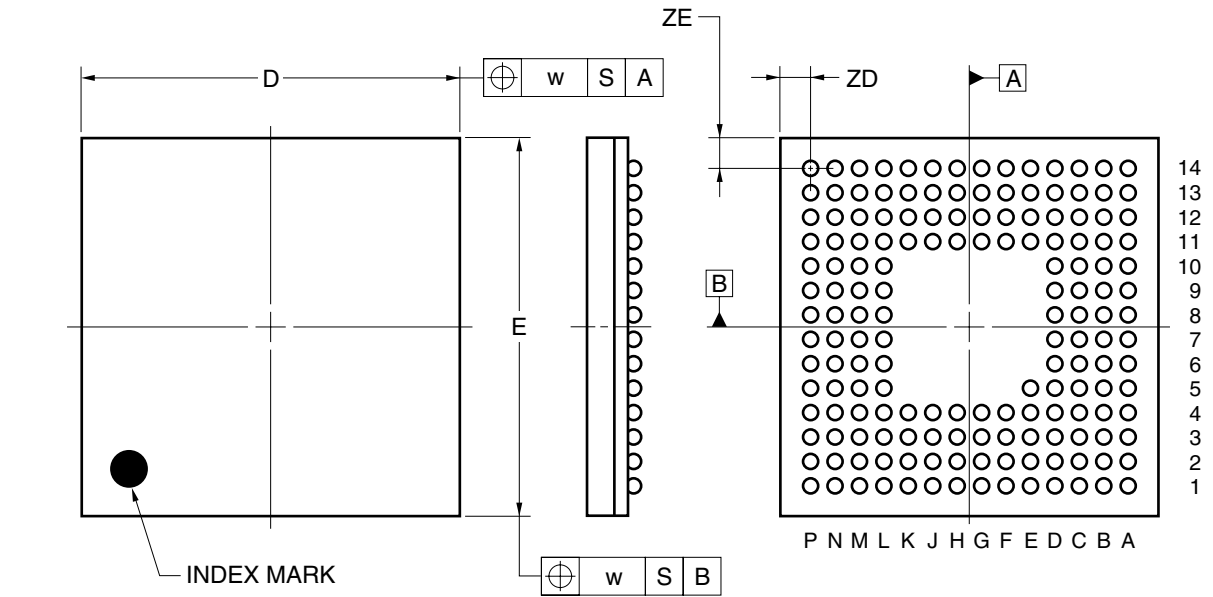
| ITEM | DIMENSIONS                                |
|------|---|
| D    | 20.00±0.20                                |
| E    | 14.00±0.20                                |
| HD   | 22.00±0.20                                |
| HE   | 16.00±0.20                                |
| A    | 1.60 MAX.                                 |
| A1   | 0.10±0.05                                 |
| A2   | 1.40±0.05                                 |
| A3   | 0.25                                      |
| b    | 0.30 <sup>+0.08</sup> <sub>-0.04</sub>    |
| c    | 0.125 <sup>+0.075</sup> <sub>-0.025</sub> |
| L    | 0.50                                      |
| Lp   | 0.60±0.15                                 |
| L1   | 1.00±0.20                                 |
| θ    | 3° <sup>+5°</sup> <sub>-3°</sub>          |
| e    | 0.65                                      |
| x    | 0.13                                      |
| y    | 0.10                                      |
| ZD   | 0.575                                     |
| ZE   | 0.825                                     |

P100GF-65-GAS

NOTE

Each lead centerline is located within 0.13 mm of its true position at maximum material condition.

<R> 161-PIN PLASTIC FBGA (10x10)



(UNIT:mm)

| ITEM | DIMENSIONS |
|------|------------|
| D    | 10.00±0.10 |
| E    | 10.00±0.10 |
| w    | 0.20       |
| A    | 1.41±0.10  |
| A1   | 0.30±0.05  |
| A2   | 1.11       |
| e    | 0.65       |
| b    | 0.40±0.05  |
| x    | 0.08       |
| y    | 0.10       |
| y1   | 0.20       |
| ZD   | 0.775      |
| ZE   | 0.775      |

P161F1-65-DA9

© NEC Electronics Corporation 2007



## CHAPTER 30 RECOMMENDED SOLDERING CONDITIONS

The V850E/IF3 and V850E/IG3 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 30-1. Surface Mounting Type Soldering Conditions (1/2)**

- (1) **μPD70F3451GC-UBT-A: 80-pin plastic LQFP (14 × 14)**  
**μPD70F3452GC-UBT-A: 80-pin plastic LQFP (14 × 14)**  
**μPD70F3453GC-8EA-A: 100-pin plastic LQFP (fine pitch) (14 × 14)**  
**μPD70F3454GC-8EA-A: 100-pin plastic LQFP (fine pitch) (14 × 14)**

| Soldering Method | Soldering Conditions  | Recommended Condition Symbol |
|------------------|---|------------------------------|
| Infrared reflow  | Package peak temperature: 260°C, Time: 60 seconds max. (at 220°C or higher),<br>Count: Three times or less,<br>Exposure limit: 7 days <sup>Note</sup> (after that, prebake at 125°C for 10 to 72 hours) | IR60-107-3                   |
| Wave soldering   | For details, contact an NEC Electronics sales representative.   | –                            |
| 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.

Table 30-1. Surface Mounting Type Soldering Conditions (2/2)

- (2)  $\mu$ PD70F3453GF-GAS-AX: 100-pin plastic LQFP (14 × 20)  
 $\mu$ PD70F3454GF-GAS-AX: 100-pin plastic LQFP (14 × 20)

| Soldering Method | Soldering Conditions  | Recommended Condition Symbol |
|------------------|---|------------------------------|
| Infrared reflow  | Package peak temperature: 260°C, Time: 60 seconds max. (at 220°C or higher),<br>Count: Three times or less,<br>Exposure limit: 7 days <sup>Note</sup> (after that, prebake at 125°C for 20 to 72 hours) | IR60-207-3                   |
| Wave soldering   | For details, contact an NEC Electronics sales representative.   | –                            |
| 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 -AX 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.

- (3)  $\mu$ PD70F3454F1-DA9-A: 161-pin plastic FBGA (10 × 10)

| Soldering Method | Soldering Conditions  | Recommended Condition Symbol |
|------------------|---|------------------------------|
| Infrared reflow  | Package peak temperature: 260°C, Time: 60 seconds max. (at 220°C or higher),<br>Count: Three times or less,<br>Exposure limit: 7 days <sup>Note</sup> (after that, prebake at 125°C for 20 to 72 hours) | IR60-207-3                   |
| Wave soldering   | For details, contact an NEC Electronics sales representative.   | –                            |

**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.

- 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 CAUTIONS

### A.1 Restriction on Conflict Between sld Instruction and Interrupt Request

#### A.1.1 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  
.  
.  
.  
<ii> mov r10, r28  
<iii> sld.w 0x28, 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.

#### A.1.2 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) For 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.

## APPENDIX B REGISTER INDEX

(1/13)

| Symbol   | Name  | Unit | Page |
|----------|---|------|------|
| AD0CH1   | A/D converter 0 channel specification register 1            | ADC0 | 634  |
| AD0CH2   | A/D converter 0 channel specification register 2            | ADC0 | 636  |
| AD0CHEN  | A/D converter 0 conversion channel specification register   | ADC0 | 626  |
| AD0CHENH | A/D converter 0 conversion channel specification register H | ADC0 | 626  |
| AD0CHENL | A/D converter 0 conversion channel specification register L | ADC0 | 626  |
| AD0CR0   | A/D0 conversion result register 0                           | ADC0 | 628  |
| AD0CR0H  | A/D0 conversion result register 0H                          | ADC0 | 628  |
| AD0CR1   | A/D0 conversion result register 1                           | ADC0 | 628  |
| AD0CR10  | A/D0 conversion result register 10                          | ADC0 | 628  |
| AD0CR10H | A/D0 conversion result register 10H                         | ADC0 | 628  |
| AD0CR11  | A/D0 conversion result register 11                          | ADC0 | 628  |
| AD0CR11H | A/D0 conversion result register 11H                         | ADC0 | 628  |
| AD0CR12  | A/D0 conversion result register 12                          | ADC0 | 628  |
| AD0CR12H | A/D0 conversion result register 12H                         | ADC0 | 628  |
| AD0CR13  | A/D0 conversion result register 13                          | ADC0 | 628  |
| AD0CR13H | A/D0 conversion result register 13H                         | ADC0 | 628  |
| AD0CR14  | A/D0 conversion result register 14                          | ADC0 | 628  |
| AD0CR14H | A/D0 conversion result register 14H                         | ADC0 | 628  |
| AD0CR15  | A/D0 conversion result register 15                          | ADC0 | 628  |
| AD0CR15H | A/D0 conversion result register 15H                         | ADC0 | 628  |
| AD0CR1H  | A/D0 conversion result register 1H                          | ADC0 | 628  |
| AD0CR2   | A/D0 conversion result register 2                           | ADC0 | 628  |
| AD0CR2H  | A/D0 conversion result register 2H                          | ADC0 | 628  |
| AD0CR3   | A/D0 conversion result register 3                           | ADC0 | 628  |
| AD0CR3H  | A/D0 conversion result register 3H                          | ADC0 | 628  |
| AD0CR4   | A/D0 conversion result register 4                           | ADC0 | 628  |
| AD0CR4H  | A/D0 conversion result register 4H                          | ADC0 | 628  |
| AD0CR5   | A/D0 conversion result register 5                           | ADC0 | 628  |
| AD0CR5H  | A/D0 conversion result register 5H                          | ADC0 | 628  |
| AD0CR6   | A/D0 conversion result register 6                           | ADC0 | 628  |
| AD0CR6H  | A/D0 conversion result register 6H                          | ADC0 | 628  |
| AD0CR7   | A/D0 conversion result register 7                           | ADC0 | 628  |
| AD0CR7H  | A/D0 conversion result register 7H                          | ADC0 | 628  |
| AD0CR8   | A/D0 conversion result register 8                           | ADC0 | 628  |
| AD0CR8H  | A/D0 conversion result register 8H                          | ADC0 | 628  |
| AD0CR9   | A/D0 conversion result register 9                           | ADC0 | 628  |
| AD0CR9H  | A/D0 conversion result register 9H                          | ADC0 | 628  |
| AD0CTC   | A/D converter 0 conversion time control register            | ADC0 | 625  |
| AD0CTL0  | A/D converter 0 control register                            | ADC0 | 632  |

**APPENDIX B REGISTER INDEX**

(2/13)

| Symbol   | Name  | Unit | Page |
|----------|---|------|------|
| AD0ECR0  | A/D0 conversion result extension register 0                 | ADC0 | 638  |
| AD0ECR0H | A/D0 conversion result extension register 0H                | ADC0 | 638  |
| AD0ECR1  | A/D0 conversion result extension register 1                 | ADC0 | 638  |
| AD0ECR1H | A/D0 conversion result extension register 1H                | ADC0 | 638  |
| AD0ECR2  | A/D0 conversion result extension register 2                 | ADC0 | 638  |
| AD0ECR2H | A/D0 conversion result extension register 2H                | ADC0 | 638  |
| AD0ECR3  | A/D0 conversion result extension register 3                 | ADC0 | 638  |
| AD0ECR3H | A/D0 conversion result extension register 3H                | ADC0 | 638  |
| AD0ECR4  | A/D0 conversion result extension register 4                 | ADC0 | 638  |
| AD0ECR4H | A/D0 conversion result extension register 4H                | ADC0 | 638  |
| AD0FLG   | A/D converter 0 flag register                               | ADC0 | 640  |
| AD0FLGB  | A/D converter 0 flag buffer register                        | ADC0 | 641  |
| AD0IC    | Interrupt control register                                  | INTC | 1000 |
| AD0OCKS  | A/D converter 0 clock select register                       | ADC0 | 643  |
| AD0SCM   | A/D converter 0 scan mode register                          | ADC0 | 623  |
| AD0SCMH  | A/D converter 0 scan mode register H                        | ADC0 | 623  |
| AD0SCML  | A/D converter 0 scan mode register L                        | ADC0 | 623  |
| AD0TSEL  | A/D converter 0 trigger select register                     | ADC0 | 633  |
| AD1CH1   | A/D converter 1 channel specification register 1            | ADC1 | 634  |
| AD1CH2   | A/D converter 1 channel specification register 2            | ADC1 | 636  |
| AD1CHEN  | A/D converter 1 conversion channel specification register   | ADC1 | 626  |
| AD1CHENH | A/D converter 1 conversion channel specification register H | ADC1 | 626  |
| AD1CHENL | A/D converter 1 conversion channel specification register L | ADC1 | 626  |
| AD1CR0   | A/D1 conversion result register 0                           | ADC1 | 628  |
| AD1CR0H  | A/D1 conversion result register 0H                          | ADC1 | 628  |
| AD1CR1   | A/D1 conversion result register 1                           | ADC1 | 628  |
| AD1CR10  | A/D1 conversion result register 10                          | ADC1 | 628  |
| AD1CR10H | A/D1 conversion result register 10H                         | ADC1 | 628  |
| AD1CR11  | A/D1 conversion result register 11                          | ADC1 | 628  |
| AD1CR11H | A/D1 conversion result register 11H                         | ADC1 | 628  |
| AD1CR12  | A/D1 conversion result register 12                          | ADC1 | 628  |
| AD1CR12H | A/D1 conversion result register 12H                         | ADC1 | 628  |
| AD1CR13  | A/D1 conversion result register 13                          | ADC1 | 628  |
| AD1CR13H | A/D1 conversion result register 13H                         | ADC1 | 628  |
| AD1CR14  | A/D1 conversion result register 14                          | ADC1 | 628  |
| AD1CR14H | A/D1 conversion result register 14H                         | ADC1 | 628  |
| AD1CR15  | A/D1 conversion result register 15                          | ADC1 | 628  |
| AD1CR15H | A/D1 conversion result register 15H                         | ADC1 | 628  |
| AD1CR1H  | A/D1 conversion result register 1H                          | ADC1 | 628  |
| AD1CR2   | A/D1 conversion result register 2                           | ADC1 | 628  |
| AD1CR2H  | A/D1 conversion result register 2H                          | ADC1 | 628  |
| AD1CR3   | A/D1 conversion result register 3                           | ADC1 | 628  |
| AD1CR3H  | A/D1 conversion result register 3H                          | ADC1 | 628  |
| AD1CR4   | A/D1 conversion result register 4                           | ADC1 | 628  |

**APPENDIX B REGISTER INDEX**

(3/13)

| Symbol   | Name   | Unit | Page |
|----------|--|------|------|
| AD1CR4H  | A/D1 conversion result register 4H               | ADC1 | 628  |
| AD1CR5   | A/D1 conversion result register 5                | ADC1 | 628  |
| AD1CR5H  | A/D1 conversion result register 5H               | ADC1 | 628  |
| AD1CR6   | A/D1 conversion result register 6                | ADC1 | 628  |
| AD1CR6H  | A/D1 conversion result register 6H               | ADC1 | 628  |
| AD1CR7   | A/D1 conversion result register 7                | ADC1 | 628  |
| AD1CR7H  | A/D1 conversion result register 7H               | ADC1 | 628  |
| AD1CR8   | A/D1 conversion result register 8                | ADC1 | 628  |
| AD1CR8H  | A/D1 conversion result register 8H               | ADC1 | 628  |
| AD1CR9   | A/D1 conversion result register 9                | ADC1 | 628  |
| AD1CR9H  | A/D1 conversion result register 9H               | ADC1 | 628  |
| AD1CTC   | A/D converter 1 conversion time control register | ADC1 | 625  |
| AD1CTL0  | A/D converter 1 control register                 | ADC1 | 632  |
| AD1ECR0  | A/D1 conversion result extension register 0      | ADC1 | 638  |
| AD1ECR0H | A/D1 conversion result extension register 0H     | ADC1 | 638  |
| AD1ECR1  | A/D1 conversion result extension register 1      | ADC1 | 638  |
| AD1ECR1H | A/D1 conversion result extension register 1H     | ADC1 | 638  |
| AD1ECR2  | A/D1 conversion result extension register 2      | ADC1 | 638  |
| AD1ECR2H | A/D1 conversion result extension register 2H     | ADC1 | 638  |
| AD1ECR3  | A/D1 conversion result extension register 3      | ADC1 | 638  |
| AD1ECR3H | A/D1 conversion result extension register 3H     | ADC1 | 638  |
| AD1ECR4  | A/D1 conversion result extension register 4      | ADC1 | 638  |
| AD1ECR4H | A/D1 conversion result extension register 4H     | ADC1 | 638  |
| AD1FLG   | A/D converter 1 flag register                    | ADC1 | 640  |
| AD1FLGB  | A/D converter 1 flag buffer register             | ADC1 | 641  |
| AD1IC    | Interrupt control register                       | INTC | 1000 |
| AD1OCKS  | A/D converter 1 clock select register            | ADC1 | 643  |
| AD1SCM   | A/D converter 1 scan mode register               | ADC1 | 623  |
| AD1SCMH  | A/D converter 1 scan mode register H             | ADC1 | 623  |
| AD1SCML  | A/D converter 1 scan mode register L             | ADC1 | 623  |
| AD1TSEL  | A/D converter 1 trigger select register          | ADC1 | 633  |
| AD2CR0   | A/D2 conversion result register 0                | ADC2 | 695  |
| AD2CR0H  | A/D2 conversion result register 0H               | ADC2 | 695  |
| AD2CR1   | A/D2 conversion result register 1                | ADC2 | 695  |
| AD2CR1H  | A/D2 conversion result register 1H               | ADC2 | 695  |
| AD2CR2   | A/D2 conversion result register 2                | ADC2 | 695  |
| AD2CR2H  | A/D2 conversion result register 2H               | ADC2 | 695  |
| AD2CR3   | A/D2 conversion result register 3                | ADC2 | 695  |
| AD2CR3H  | A/D2 conversion result register 3H               | ADC2 | 695  |
| AD2CR4   | A/D2 conversion result register 4                | ADC2 | 695  |
| AD2CR4H  | A/D2 conversion result register 4H               | ADC2 | 695  |
| AD2CR5   | A/D2 conversion result register 5                | ADC2 | 695  |
| AD2CR5H  | A/D2 conversion result register 5H               | ADC2 | 695  |
| AD2CR6   | A/D2 conversion result register 6                | ADC2 | 695  |

**APPENDIX B REGISTER INDEX**

(4/13)

| Symbol  | Name  | Unit       | Page |
|---------|---|------------|------|
| AD2CR6H | A/D2 conversion result register 6H              | ADC2       | 695  |
| AD2CR7  | A/D2 conversion result register 7               | ADC2       | 695  |
| AD2CR7H | A/D2 conversion result register 7H              | ADC2       | 695  |
| AD2IC   | Interrupt control register                      | INTC       | 1000 |
| AD2M0   | A/D converter 2 mode register 0                 | ADC2       | 692  |
| AD2M1   | A/D converter 2 mode register 1                 | ADC2       | 693  |
| AD2S    | A/D converter 2 channel specification register  | ADC2       | 694  |
| ADLTS1  | A/D LDTRG1 input select register                | ADC0, ADC1 | 642  |
| ADLTS2  | A/D LDTRG2 input select register                | ADC0, ADC1 | 642  |
| ADT0IC  | Interrupt control register                      | INTC       | 1000 |
| ADT1IC  | Interrupt control register                      | INTC       | 1000 |
| ADTF    | A/D trigger falling edge specification register | ADC0, ADC1 | 644  |
| ADTR    | A/D trigger rising edge specification register  | ADC0, ADC1 | 644  |
| AWC     | Address wait control register                   | BCU        | 943  |
| BCC     | Bus cycle control register                      | BCU        | 946  |
| BCT0    | Bus cycle type configuration register 0         | BCU        | 931  |
| BSC     | Bus size configuration register                 | BCU        | 933  |
| CB0CTL0 | CSIB0 control register 0                        | CSIB       | 812  |
| CB0CTL1 | CSIB0 control register 1                        | CSIB       | 815  |
| CB0CTL2 | CSIB0 control register 2                        | CSIB       | 816  |
| CB0REIC | Interrupt control register                      | INTC       | 1000 |
| CB0RIC  | Interrupt control register                      | INTC       | 1000 |
| CB0RX   | CSIB0 receive data register                     | CSIB       | 811  |
| CB0RXL  | CSIB0 receive data register L                   | CSIB       | 811  |
| CB0STR  | CSIB0 status register                           | CSIB       | 818  |
| CB0TIC  | Interrupt control register                      | INTC       | 992  |
| CB0TX   | CSIB0 transmit data register                    | CSIB       | 811  |
| CB0TXL  | CSIB0 transmit data register L                  | CSIB       | 811  |
| CB1CTL0 | CSIB1 control register 0                        | CSIB       | 812  |
| CB1CTL1 | CSIB1 control register 1                        | CSIB       | 815  |
| CB1CTL2 | CSIB1 control register 2                        | CSIB       | 816  |
| CB1REIC | Interrupt control register                      | INTC       | 1000 |
| CB1RIC  | Interrupt control register                      | INTC       | 1000 |
| CB1RX   | CSIB1 receive data register                     | CSIB       | 811  |
| CB1RXL  | CSIB1 receive data register L                   | CSIB       | 811  |
| CB1STR  | CSIB1 status register                           | CSIB       | 818  |
| CB1TIC  | Interrupt control register                      | INTC       | 1000 |
| CB1TX   | CSIB1 transmit data register                    | CSIB       | 811  |
| CB1TXL  | CSIB1 transmit data register L                  | CSIB       | 811  |
| CB2CTL0 | CSIB2 control register 0                        | CSIB       | 812  |
| CB2CTL1 | CSIB2 control register 1                        | CSIB       | 815  |
| CB2CTL2 | CSIB2 control register 2                        | CSIB       | 816  |
| CB2REIC | Interrupt control register                      | INTC       | 1000 |
| CB2RIC  | Interrupt control register                      | INTC       | 1000 |

**APPENDIX B REGISTER INDEX**

(5/13)

| Symbol   | Name  | Unit       | Page |
|----------|---|------------|------|
| CB2RX    | CSIB2 receive data register                                     | CSIB       | 811  |
| CB2RXL   | CSIB2 receive data register L                                   | CSIB       | 811  |
| CB2STR   | CSIB2 status register   | CSIB       | 818  |
| CB2TIC   | Interrupt control register                                      | INTC       | 1000 |
| CB2TX    | CSIB2 transmit data register                                    | CSIB       | 811  |
| CB2TXL   | CSIB2 transmit data register L                                  | CSIB       | 811  |
| CLM      | Clock monitor mode register                                     | CG         | 183  |
| CMPOCTL0 | Comparator 0 control register 0                                 | ADC0       | 647  |
| CMPOCTL1 | Comparator 0 control register 1                                 | ADC0       | 649  |
| CMPOCTL2 | Comparator 0 control register 2                                 | ADC0       | 651  |
| CMPOCTL3 | Comparator 0 control register 3                                 | ADC0       | 652  |
| CMP1CTL0 | Comparator 1 control register 0                                 | ADC1       | 647  |
| CMP1CTL1 | Comparator 1 control register 1                                 | ADC1       | 649  |
| CMP1CTL2 | Comparator 1 control register 2                                 | ADC1       | 651  |
| CMP1CTL3 | Comparator 1 control register 3                                 | ADC1       | 652  |
| CMPIC0F  | Interrupt control register                                      | INTC       | 1000 |
| CMPIC0L  | Interrupt control register                                      | INTC       | 1000 |
| CMPIC1F  | Interrupt control register                                      | INTC       | 1000 |
| CMPIC1L  | Interrupt control register                                      | INTC       | 1000 |
| CMPNFC0F | Comparator output digital noise elimination register 0F         | ADC0       | 654  |
| CMPNFC0L | Comparator output digital noise elimination register 0L         | ADC0       | 654  |
| CMPNFC1F | Comparator output digital noise elimination register 1F         | ADC1       | 654  |
| CMPNFC1L | Comparator output digital noise elimination register 1L         | ADC1       | 654  |
| CMPOF    | Comparator output interrupt falling edge specification register | ADC0, ADC1 | 655  |
| CMPOR    | Comparator output interrupt rising edge specification register  | ADC0, ADC1 | 655  |
| DADC0    | DMA addressing control register 0                               | DMAC       | 966  |
| DADC1    | DMA addressing control register 1                               | DMAC       | 966  |
| DADC2    | DMA addressing control register 2                               | DMAC       | 966  |
| DADC3    | DMA addressing control register 3                               | DMAC       | 966  |
| DBC0     | DMA transfer count register 0                                   | DMAC       | 965  |
| DBC1     | DMA transfer count register 1                                   | DMAC       | 965  |
| DBC2     | DMA transfer count register 2                                   | DMAC       | 965  |
| DBC3     | DMA transfer count register 3                                   | DMAC       | 965  |
| DCHC0    | DMA channel control register 0                                  | DMAC       | 967  |
| DCHC1    | DMA channel control register 1                                  | DMAC       | 967  |
| DCHC2    | DMA channel control register 2                                  | DMAC       | 967  |
| DCHC3    | DMA channel control register 3                                  | DMAC       | 967  |
| DDA0H    | DMA destination address register 0H                             | DMAC       | 963  |
| DDA0L    | DMA destination address register 0L                             | DMAC       | 964  |
| DDA1H    | DMA destination address register 1H                             | DMAC       | 963  |
| DDA1L    | DMA destination address register 1L                             | DMAC       | 964  |
| DDA2H    | DMA destination address register 2H                             | DMAC       | 963  |
| DDA2L    | DMA destination address register 2L                             | DMAC       | 964  |
| DDA3H    | DMA destination address register 3H                             | DMAC       | 963  |



**APPENDIX B REGISTER INDEX**

(6/13)

| Symbol   | Name                                      | Unit             | Page |
|----------|---|------------------|------|
| DDA3L    | DMA destination address register 3L       | DMAC             | 964  |
| DMAIC0   | Interrupt control register                | INTC             | 1000 |
| DMAIC1   | Interrupt control register                | INTC             | 1000 |
| DMAIC2   | Interrupt control register                | INTC             | 1000 |
| DMAIC3   | Interrupt control register                | INTC             | 1000 |
| DSA0H    | DMA source address register 0H            | DMAC             | 961  |
| DSA0L    | DMA source address register 0L            | DMAC             | 962  |
| DSA1H    | DMA source address register 1H            | DMAC             | 961  |
| DSA1L    | DMA source address register 1L            | DMAC             | 962  |
| DSA2H    | DMA source address register 2H            | DMAC             | 961  |
| DSA2L    | DMA source address register 2L            | DMAC             | 962  |
| DSA3H    | DMA source address register 3H            | DMAC             | 961  |
| DSA3L    | DMA source address register 3L            | DMAC             | 962  |
| DTFR0    | DMA trigger factor register 0             | DMAC             | 969  |
| DTFR1    | DMA trigger factor register 1             | DMAC             | 969  |
| DTFR2    | DMA trigger factor register 2             | DMAC             | 969  |
| DTFR3    | DMA trigger factor register 3             | DMAC             | 969  |
| DVC      | Bus clock division control register       | BCU              | 947  |
| DWC0     | Data wait control register 0              | BCU              | 941  |
| HZA0CTL0 | High-impedance output control register 00 | Timer            | 556  |
| HZA0CTL1 | High-impedance output control register 01 | Timer            | 556  |
| HZA1CTL0 | High-impedance output control register 10 | Timer            | 556  |
| HZA1CTL1 | High-impedance output control register 11 | Timer            | 556  |
| HZA2CTL0 | High-impedance output control register 20 | Timer            | 556  |
| HZA2CTL1 | High-impedance output control register 21 | Timer            | 556  |
| HZA3CTL0 | High-impedance output control register 30 | Timer            | 556  |
| HZA3CTL1 | High-impedance output control register 31 | Timer            | 556  |
| IIC0     | IIC shift register 0                      | I <sup>2</sup> C | 857  |
| IICC0    | IIC control register 0                    | I <sup>2</sup> C | 860  |
| IICCL0   | IIC clock select register 0               | I <sup>2</sup> C | 869  |
| IICF0    | IIC flag register 0                       | I <sup>2</sup> C | 867  |
| IICIC    | Interrupt control register                | INTC             | 1000 |
| IICOCKS  | IICOPS clock select register              | I <sup>2</sup> C | 870  |
| IICS0    | IIC status register 0                     | I <sup>2</sup> C | 864  |
| IICX0    | IIC function expansion register 0         | I <sup>2</sup> C | 870  |
| IMR0     | Interrupt mask register 0                 | INTC             | 1005 |
| IMR0H    | Interrupt mask register 0H                | INTC             | 1005 |
| IMR0L    | Interrupt mask register 0L                | INTC             | 1005 |
| IMR1     | Interrupt mask register 1                 | INTC             | 1005 |
| IMR1H    | Interrupt mask register 1H                | INTC             | 1005 |
| IMR1L    | Interrupt mask register 1L                | INTC             | 1005 |
| IMR2     | Interrupt mask register 2                 | INTC             | 1005 |
| IMR2H    | Interrupt mask register 2H                | INTC             | 1005 |
| IMR2L    | Interrupt mask register 2L                | INTC             | 1005 |

**APPENDIX B REGISTER INDEX**

(7/13)

| Symbol   | Name   | Unit | Page |
|----------|--|------|------|
| IMR3     | Interrupt mask register 3                                | INTC | 1005 |
| IMR3H    | Interrupt mask register 3H                               | INTC | 1005 |
| IMR3L    | Interrupt mask register 3L                               | INTC | 1005 |
| IMR4     | Interrupt mask register 4                                | INTC | 1005 |
| IMR4H    | Interrupt mask register 4H                               | INTC | 1005 |
| IMR4L    | Interrupt mask register 4L                               | INTC | 1005 |
| IMR5     | Interrupt mask register 5                                | INTC | 1005 |
| IMR5H    | Interrupt mask register 5H                               | INTC | 1005 |
| IMR5L    | Interrupt mask register 5L                               | INTC | 1005 |
| INTF0    | External interrupt falling edge specification register 0 | INTC | 1011 |
| INTF1    | External interrupt falling edge specification register 1 | INTC | 1012 |
| INTF2    | External interrupt falling edge specification register 2 | INTC | 1013 |
| INTNFC14 | Digital noise elimination 0 control register 14          | Port | 169  |
| INTNFC15 | Digital noise elimination 0 control register 15          | Port | 169  |
| INTNFC16 | Digital noise elimination 0 control register 16          | Port | 169  |
| INTR0    | External interrupt rising edge specification register 0  | INTC | 1011 |
| INTR1    | External interrupt rising edge specification register 1  | INTC | 1012 |
| INTR2    | External interrupt rising edge specification register 2  | INTC | 1013 |
| ISPR     | In-service priority register                             | INTC | 1008 |
| LVIHIC   | Interrupt control register                               | INTC | 1000 |
| LVILIC   | Interrupt control register                               | INTC | 1000 |
| LVIM     | Low-voltage detection register                           | LVI  | 1042 |
| LVIS     | Low-voltage detection level select register              | LVI  | 1043 |
| OPOCTL0  | Operational amplifier 0 control register 0               | ADC0 | 645  |
| OP1CTL0  | Operational amplifier 1 control register 0               | ADC0 | 645  |
| OSTS     | Oscillation stabilization time select register           | CG   | 182  |
| P0       | Port 0 register  | Port | 108  |
| P1       | Port 1 register  | Port | 114  |
| P2       | Port 2 register  | Port | 120  |
| P3       | Port 3 register  | Port | 126  |
| P4       | Port 4 register  | Port | 132  |
| P7       | Port 7 register  | Port | 138  |
| PCC      | Processor clock control register                         | CG   | 179  |
| PDL      | Port DL register   | Port | 140  |
| PDLH     | Port DLH register  | Port | 140  |
| PDLL     | Port DLL register  | Port | 140  |
| PF3      | Port 3 function register                                 | Port | 130  |
| PFC0     | Port 0 function control register                         | Port | 110  |
| PFC1     | Port 1 function control register                         | Port | 116  |
| PFC2     | Port 2 function control register                         | Port | 122  |
| PFC3     | Port 3 function control register                         | Port | 128  |
| PFC4     | Port 4 function control register                         | Port | 134  |
| PFCE0    | Port 0 function control expansion register               | Port | 110  |
| PFCE1    | Port 1 function control expansion register               | Port | 116  |

**APPENDIX B REGISTER INDEX**

(8/13)

| Symbol | Name                                       | Unit | Page      |
|--------|--|------|-----------|
| PFCE2  | Port 2 function control expansion register | Port | 122       |
| PFCE3  | Port 3 function control expansion register | Port | 128       |
| PFCE4  | Port 4 function control expansion register | Port | 134       |
| PIC00  | Interrupt control register                 | INTC | 1000      |
| PIC01  | Interrupt control register                 | INTC | 1000      |
| PIC02  | Interrupt control register                 | INTC | 1000      |
| PIC03  | Interrupt control register                 | INTC | 1000      |
| PIC04  | Interrupt control register                 | INTC | 1000      |
| PIC05  | Interrupt control register                 | INTC | 1000      |
| PIC06  | Interrupt control register                 | INTC | 1000      |
| PIC07  | Interrupt control register                 | INTC | 1000      |
| PIC08  | Interrupt control register                 | INTC | 1000      |
| PIC09  | Interrupt control register                 | INTC | 1000      |
| PIC10  | Interrupt control register                 | INTC | 1000      |
| PIC11  | Interrupt control register                 | INTC | 1000      |
| PIC12  | Interrupt control register                 | INTC | 1000      |
| PIC13  | Interrupt control register                 | INTC | 1000      |
| PIC14  | Interrupt control register                 | INTC | 1000      |
| PIC15  | Interrupt control register                 | INTC | 1000      |
| PIC16  | Interrupt control register                 | INTC | 1000      |
| PIC17  | Interrupt control register                 | INTC | 1000      |
| PIC18  | Interrupt control register                 | INTC | 1000      |
| PLLCTL | PLL control register                       | CG   | 178       |
| PM0    | Port 0 mode register                       | Port | 108       |
| PM1    | Port 1 mode register                       | Port | 114       |
| PM2    | Port 2 mode register                       | Port | 120       |
| PM3    | Port 3 mode register                       | Port | 126       |
| PM4    | Port 4 mode register                       | Port | 132       |
| PMC0   | Port 0 mode control register               | Port | 109       |
| PMC1   | Port 1 mode control register               | Port | 115       |
| PMC2   | Port 2 mode control register               | Port | 121       |
| PMC3   | Port 3 mode control register               | Port | 127       |
| PMC4   | Port 4 mode control register               | Port | 133       |
| PMC7   | Port 7 mode control register               | Port | 138       |
| PMCDL  | Port DL mode control register              | Port | 142       |
| PMCDLH | Port DL mode control register H            | Port | 142       |
| PMCDLL | Port DL mode control register L            | Port | 142       |
| PMDL   | Port DL mode register                      | Port | 141       |
| PMDLH  | Port DL mode register H                    | Port | 141       |
| PMDLL  | Port DL mode register L                    | Port | 141       |
| PRCMD  | Command register                           | CPU  | 95        |
| PSC    | Power save control register                | CPU  | 180, 1027 |
| PSMR   | Power save mode register                   | CPU  | 181, 1028 |
| PU0    | Pull-up resistor option register 0         | Port | 112       |

**APPENDIX B REGISTER INDEX**

(9/13)

| Symbol   | Name                                 | Unit             | Page |
|----------|--------------------------------------|------------------|------|
| PU1      | Pull-up resistor option register 1   | Port             | 118  |
| PU2      | Pull-up resistor option register 2   | Port             | 124  |
| PU3      | Pull-up resistor option register 3   | Port             | 130  |
| PU4      | Pull-up resistor option register 4   | Port             | 136  |
| PUDL     | Pull-up resistor option register DL  | Port             | 143  |
| PUDLH    | Pull-up resistor option register DLH | Port             | 143  |
| PUDLL    | Pull-up resistor option register DLL | Port             | 143  |
| RESF     | Reset source flag register           | Reset            | 1036 |
| SVA0     | Slave address register 0             | I <sup>2</sup> C | 857  |
| SYS      | System status register               | CPU              | 96   |
| TA0CCIC0 | Interrupt control register           | INTC             | 1000 |
| TA0CCIC1 | Interrupt control register           | INTC             | 1000 |
| TA0OVIC  | Interrupt control register           | INTC             | 1000 |
| TA1CCIC0 | Interrupt control register           | INTC             | 1000 |
| TA1CCIC1 | Interrupt control register           | INTC             | 1000 |
| TA1OVIC  | Interrupt control register           | INTC             | 1000 |
| TA2CCIC0 | Interrupt control register           | INTC             | 1000 |
| TA2CCIC1 | Interrupt control register           | INTC             | 1000 |
| TA2OVIC  | Interrupt control register           | INTC             | 1000 |
| TA3CCIC0 | Interrupt control register           | INTC             | 1000 |
| TA3CCIC1 | Interrupt control register           | INTC             | 1000 |
| TA3OVIC  | Interrupt control register           | INTC             | 1000 |
| TA4CCIC0 | Interrupt control register           | INTC             | 1000 |
| TA4CCIC1 | Interrupt control register           | INTC             | 1000 |
| TA4OVIC  | Interrupt control register           | INTC             | 1000 |
| TAA0CCR0 | TAA0 capture/compare register 0      | TAA              | 207  |
| TAA0CCR1 | TAA0 capture/compare register 1      | TAA              | 209  |
| TAA0CNT  | TAA0 counter read buffer register    | TAA              | 211  |
| TAA0CTL0 | TAA0 control register 0              | TAA              | 199  |
| TAA0CTL1 | TAA0 control register 1              | TAA              | 200  |
| TAA0OPT0 | TAA0 option register 0               | TAA              | 206  |
| TAA1CCR0 | TAA1 capture/compare register 0      | TAA              | 207  |
| TAA1CCR1 | TAA1 capture/compare register 1      | TAA              | 209  |
| TAA1CNT  | TAA1 counter read buffer register    | TAA              | 211  |
| TAA1CTL0 | TAA1 control register 0              | TAA              | 199  |
| TAA1CTL1 | TAA1 control register 1              | TAA              | 200  |
| TAA1OPT0 | TAA1 option register 0               | TAA              | 206  |
| TAA2CCR0 | TAA2 capture/compare register 0      | TAA              | 207  |
| TAA2CCR1 | TAA2 capture/compare register 1      | TAA              | 209  |
| TAA2CNT  | TAA2 counter read buffer register    | TAA              | 211  |
| TAA2CTL0 | TAA2 control register 0              | TAA              | 199  |
| TAA2CTL1 | TAA2 control register 1              | TAA              | 200  |
| TAA2IOC0 | TAA2 I/O control register 0          | TAA              | 202  |
| TAA2IOC1 | TAA2 I/O control register 1          | TAA              | 204  |

**APPENDIX B REGISTER INDEX**

(10/13)

| Symbol   | Name                              | Unit  | Page     |
|----------|-----------------------------------|-------|----------|
| TAA2IOC2 | TAA2 I/O control register 2       | TAA   | 205      |
| TAA2OPT0 | TAA2 option register 0            | TAA   | 206      |
| TAA3CCR0 | TAA3 capture/compare register 0   | TAA   | 207      |
| TAA3CCR1 | TAA3 capture/compare register 1   | TAA   | 209      |
| TAA3CNT  | TAA3 counter read buffer register | TAA   | 211      |
| TAA3CTL0 | TAA3 control register 0           | TAA   | 199      |
| TAA3CTL1 | TAA3 control register 1           | TAA   | 200      |
| TAA3IOC0 | TAA3 I/O control register 0       | TAA   | 202      |
| TAA3IOC1 | TAA3 I/O control register 1       | TAA   | 204      |
| TAA3IOC2 | TAA3 I/O control register 2       | TAA   | 205      |
| TAA3OPT0 | TAA3 option register 0            | TAA   | 206      |
| TAA4CCR0 | TAA4 capture/compare register 0   | TAA   | 207      |
| TAA4CCR1 | TAA4 capture/compare register 1   | TAA   | 209      |
| TAA4CNT  | TAA4 counter read buffer register | TAA   | 211      |
| TAA4CTL0 | TAA4 control register 0           | TAA   | 199      |
| TAA4CTL1 | TAA4 control register 1           | TAA   | 200      |
| TAA4IOC0 | TAA4 I/O control register 0       | TAA   | 202      |
| TAA4IOC1 | TAA4 I/O control register 1       | TAA   | 204      |
| TAA4IOC2 | TAA4 I/O control register 2       | TAA   | 205      |
| TAA4OPT0 | TAA4 option register 0            | TAA   | 206      |
| TAB0CCR0 | TAB0 capture/compare register 0   | TAB   | 303      |
| TAB0CCR1 | TAB0 capture/compare register 1   | TAB   | 305      |
| TAB0CCR2 | TAB0 capture/compare register 2   | TAB   | 307      |
| TAB0CCR3 | TAB0 capture/compare register 3   | TAB   | 309      |
| TAB0CNT  | TAB0 counter read buffer register | TAB   | 311      |
| TAB0CTL0 | TAB0 control register 0           | TAB   | 297      |
| TAB0CTL1 | TAB0 control register 1           | TAB   | 298      |
| TAB0DTC  | TAB0 dead-time compare register   | Timer | 547      |
| TAB0IOC0 | TAB0 I/O control register 0       | TAB   | 299      |
| TAB0IOC1 | TAB0 I/O control register 1       | TAB   | 300      |
| TAB0IOC2 | TAB0 I/O control register 2       | TAB   | 301      |
| TAB0IOC3 | TAB0 I/O control register 3       | Timer | 553      |
| TAB0OPT0 | TAB0 option register 0            | TAB   | 302, 549 |
| TAB0OPT1 | TAB0 option register 1            | Timer | 549      |
| TAB0OPT2 | TAB0 option register 2            | Timer | 550      |
| TAB0OPT3 | TAB0 option register 3            | Timer | 552      |
| TAB1CCR0 | TAB1 capture/compare register 0   | TAB   | 303      |
| TAB1CCR1 | TAB1 capture/compare register 1   | TAB   | 305      |
| TAB1CCR2 | TAB1 capture/compare register 2   | TAB   | 307      |
| TAB1CCR3 | TAB1 capture/compare register 3   | TAB   | 309      |
| TAB1CNT  | TAB1 counter read buffer register | TAB   | 311      |
| TAB1CTL0 | TAB1 control register 0           | TAB   | 297      |
| TAB1CTL1 | TAB1 control register 1           | TAB   | 298      |
| TAB1DTC  | TAB1 dead-time compare register   | TAB   | 547      |

**APPENDIX B REGISTER INDEX**

(11/13)

| Symbol   | Name   | Unit | Page     |
|----------|--|------|----------|
| TAB1IOC0 | TAB1 I/O control register 0                    | TAB  | 299      |
| TAB1IOC1 | TAB1 I/O control register 1                    | TAB  | 300      |
| TAB1IOC2 | TAB1 I/O control register 2                    | TAB  | 301      |
| TAB1IOC3 | TAB1 I/O control register 3                    | TAB  | 553      |
| TAB1OPT0 | TAB1 option register 0                         | TAB  | 302, 549 |
| TAB1OPT1 | TAB1 option register 1                         | TAB  | 549      |
| TAB1OPT2 | TAB1 option register 2                         | TAB  | 550      |
| TAB1OPT3 | TAB1 option register 3                         | TAB  | 552      |
| TANFC2   | Digital noise elimination 1 control register 2 | Port | 170      |
| TANFC3   | Digital noise elimination 1 control register 3 | Port | 170      |
| TANFC4   | Digital noise elimination 1 control register 4 | Port | 170      |
| TB0CCIC0 | Interrupt control register                     | INTC | 1000     |
| TB0CCIC1 | Interrupt control register                     | INTC | 1000     |
| TB0CCIC2 | Interrupt control register                     | INTC | 1000     |
| TB0CCIC3 | Interrupt control register                     | INTC | 1000     |
| TB0OVIC  | Interrupt control register                     | INTC | 1000     |
| TB1CCIC0 | Interrupt control register                     | INTC | 1000     |
| TB1CCIC1 | Interrupt control register                     | INTC | 1000     |
| TB1CCIC2 | Interrupt control register                     | INTC | 1000     |
| TB1CCIC3 | Interrupt control register                     | INTC | 1000     |
| TB1OVIC  | Interrupt control register                     | INTC | 1000     |
| TM0CMP0  | TMM0 compare register 0                        | TMM  | 536      |
| TM0CTL0  | TMM0 control register 0                        | TMM  | 537      |
| TM0EQIC0 | Interrupt control register                     | INTC | 1000     |
| TM1CMP0  | TMM1 compare register 0                        | TMM  | 536      |
| TM1CTL0  | TMM1 control register 0                        | TMM  | 537      |
| TM1EQIC0 | Interrupt control register                     | INTC | 1000     |
| TM2CMP0  | TMM2 compare register 0                        | TMM  | 536      |
| TM2CTL0  | TMM2 control register 0                        | TMM  | 537      |
| TM2EQIC0 | Interrupt control register                     | INTC | 1000     |
| TM3CMP0  | TMM3 compare register 0                        | TMM  | 536      |
| TM3CTL0  | TMM3 control register 0                        | TMM  | 537      |
| TM3EQIC0 | Interrupt control register                     | INTC | 1000     |
| TT0CCIC0 | Interrupt control register                     | INTC | 1000     |
| TT0CCIC1 | Interrupt control register                     | INTC | 1000     |
| TT0CCR0  | TMT0 capture/compare register 0                | TMT  | 423      |
| TT0CCR1  | TMT0 capture/compare register 1                | TMT  | 425      |
| TT0CNT   | TMT0 counter read buffer register              | TMT  | 427      |
| TT0CTL0  | TMT0 control register 0                        | TMT  | 408      |
| TT0CTL1  | TMT0 control register 1                        | TMT  | 409      |
| TT0CTL2  | TMT0 control register 2                        | TMT  | 411      |
| TT0IECIC | Interrupt control register                     | INTC | 1000     |
| TT0IOC0  | TMT0 I/O control register 0                    | TMT  | 413      |
| TT0IOC1  | TMT0 I/O control register 1                    | TMT  | 415      |

**APPENDIX B REGISTER INDEX**

(12/13)

| Symbol   | Name   | Unit  | Page |
|----------|--|-------|------|
| TT0IOC2  | TMT0 I/O control register 2                    | TMT   | 416  |
| TT0IOC3  | TMT0 I/O control register 3                    | TMT   | 417  |
| TT0OPT0  | TMT0 option register 0                         | TMT   | 419  |
| TT0OPT1  | TMT0 option register 1                         | TMT   | 420  |
| TT0OVIC  | Interrupt control register                     | INTC  | 1000 |
| TT0TCW   | TMT0 counter write register                    | TMT   | 427  |
| TT1CCIC0 | Interrupt control register                     | INTC  | 1000 |
| TT1CCIC1 | Interrupt control register                     | INTC  | 1000 |
| TT1CCR0  | TMT1 capture/compare register 0                | TMT   | 423  |
| TT1CCR1  | TMT1 capture/compare register 1                | TMT   | 425  |
| TT1CNT   | TMT1 counter read buffer register              | TMT   | 427  |
| TT1CTL0  | TMT1 control register 0                        | TMT   | 408  |
| TT1CTL1  | TMT1 control register 1                        | TMT   | 409  |
| TT1CTL2  | TMT1 control register 2                        | TMT   | 411  |
| TT1IECIC | Interrupt control register                     | INTC  | 1000 |
| TT1IOC0  | TMT1 I/O control register 0                    | TMT   | 413  |
| TT1IOC1  | TMT1 I/O control register 1                    | TMT   | 415  |
| TT1IOC2  | TMT1 I/O control register 2                    | TMT   | 416  |
| TT1IOC3  | TMT1 I/O control register 3                    | TMT   | 417  |
| TT1OPT0  | TMT1 option register 0                         | TMT   | 419  |
| TT1OPT1  | TMT1 option register 1                         | TMT   | 420  |
| TT1OVIC  | Interrupt control register                     | INTC  | 1000 |
| TT1TCW   | TMT1 counter write register                    | TMT   | 427  |
| TTISL0   | TMT0 capture input select register             | TMT   | 422  |
| TTISL1   | TMT1 capture input select register             | TMT   | 422  |
| TTNFC0   | Digital noise elimination 2 control register 0 | Port  | 171  |
| TTNFC1   | Digital noise elimination 2 control register 1 | Port  | 171  |
| UA0CTL0  | UARTA0 control register 0                      | UARTA | 723  |
| UA0CTL1  | UARTA0 control register 1                      | UARTA | 739  |
| UA0CTL2  | UARTA0 control register 2                      | UARTA | 740  |
| UA0OPT0  | UARTA0 option control register 0               | UARTA | 725  |
| UA0REIC  | Interrupt control register                     | INTC  | 1000 |
| UA0RIC   | Interrupt control register                     | INTC  | 1000 |
| UA0RX    | UARTA0 receive data register                   | UARTA | 727  |
| UA0STR   | UARTA0 status register                         | UARTA | 725  |
| UA0TIC   | Interrupt control register                     | INTC  | 1000 |
| UA0TX    | UARTA0 transmit data register                  | UARTA | 727  |
| UA1CTL0  | UARTA1 control register 0                      | UARTA | 723  |
| UA1CTL1  | UARTA1 control register 1                      | UARTA | 739  |
| UA1CTL2  | UARTA1 control register 2                      | UARTA | 740  |
| UA1OPT0  | UARTA1 option control register 0               | UARTA | 725  |
| UA1REIC  | Interrupt control register                     | INTC  | 1000 |
| UA1RIC   | Interrupt control register                     | INTC  | 1000 |
| UA1RX    | UARTA1 receive data register                   | UARTA | 727  |

**APPENDIX B REGISTER INDEX**

(13/13)

| Symbol  | Name                             | Unit  | Page |
|---------|----------------------------------|-------|------|
| UA1STR  | UARTA1 status register           | UARTA | 725  |
| UA1TIC  | Interrupt control register       | INTC  | 1000 |
| UA1TX   | UARTA1 transmit data register    | UARTA | 727  |
| UA2CTL0 | UARTA2 control register 0        | UARTA | 723  |
| UA2CTL1 | UARTA2 control register 1        | UARTA | 739  |
| UA2CTL2 | UARTA2 control register 2        | UARTA | 740  |
| UA2OPT0 | UARTA2 option control register 0 | UARTA | 725  |
| UA2REIC | Interrupt control register       | INTC  | 1000 |
| UA2RIC  | Interrupt control register       | INTC  | 1000 |
| UA2RX   | UARTA2 receive data register     | UARTA | 727  |
| UA2STR  | UARTA2 status register           | UARTA | 725  |
| UA2TIC  | Interrupt control register       | INTC  | 1000 |
| UA2TX   | UARTA2 transmit data register    | UARTA | 727  |
| UBCTL0  | UARTB control register 0         | UARTB | 749  |
| UBCTL2  | UARTB control register 2         | UARTB | 757  |
| UBFIC0  | UARTB FIFO control register 0    | UARTB | 761  |
| UBFIC1  | UARTB FIFO control register 1    | UARTB | 763  |
| UBFIC2  | UARTB FIFO control register 2    | UARTB | 764  |
| UBFIC2H | UARTB FIFO control register 2H   | UARTB | 764  |
| UBFIC2L | UARTB FIFO control register 2L   | UARTB | 764  |
| UBFIS0  | UARTB FIFO status register 0     | UARTB | 766  |
| UBFIS1  | UARTB FIFO status register 1     | UARTB | 767  |
| UBRX    | UARTB receive data register      | UARTB | 759  |
| UBRXAP  | UARTB receive data register AP   | UARTB | 759  |
| UBSTR   | UARTB status register            | UARTB | 755  |
| UBTX    | UARTB transmit data register     | UARTB | 758  |
| UIFIC   | Interrupt control register       | INTC  | 1000 |
| UREIC   | Interrupt control register       | INTC  | 1000 |
| URIC    | Interrupt control register       | INTC  | 1000 |
| UTIC    | Interrupt control register       | INTC  | 1000 |
| UTOIC   | Interrupt control register       | INTC  | 1000 |
| VSWC    | System wait control register     | BCU   | 97   |
| WDTE    | Watchdog timer enable register   | WDT   | 609  |
| WDTM    | Watchdog timer mode register     | WDT   | 608  |



## 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 order 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 (SP)   |
| 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  |
| S               | 1-bit data that specifies a system register in the register list   |

**APPENDIX C INSTRUCTION SET 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).<br>If, as a result of calculations,<br>n ≥ 7FFFFFFFH, let it be 7FFFFFFFH.<br>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). |

**APPENDIX C INSTRUCTION SET LIST**

**(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 Name<br>(cond) | Condition Code<br>(cccc) | Condition Formula                         | Explanation                                   |
|--------------------------|--------------------------|---|---|
| V                        | 0 0 0 0                  | $OV = 1$                                  | Overflow                                      |
| NV                       | 1 0 0 0                  | $OV = 0$                                  | No overflow                                   |
| C/L                      | 0 0 0 1                  | $CY = 1$                                  | Carry<br>Lower (Less than)                    |
| NC/NL                    | 1 0 0 1                  | $CY = 0$                                  | No carry<br>Not lower (Greater than or equal) |
| Z/E                      | 0 0 1 0                  | $Z = 1$                                   | Zero<br>Equal                                 |
| NZ/NE                    | 1 0 1 0                  | $Z = 0$                                   | Not zero<br>Not equal                         |
| NH                       | 0 0 1 1                  | $(CY \text{ or } Z) = 1$                  | Not higher (Less than or equal)               |
| H                        | 1 0 1 1                  | $(CY \text{ or } Z) = 0$                  | Higher (Greater than)                         |
| N                        | 0 1 0 0                  | $S = 1$                                   | Negative                                      |
| P                        | 1 1 0 0                  | $S = 0$                                   | Positive                                      |
| T                        | 0 1 0 1                  | –   | Always (Unconditional)                        |
| SA                       | 1 1 0 1                  | $SAT = 1$                                 | Saturated                                     |
| LT                       | 0 1 1 0                  | $(S \text{ xor } OV) = 1$                 | Less than signed                              |
| GE                       | 1 1 1 0                  | $(S \text{ xor } OV) = 0$                 | Greater than or equal signed                  |
| LE                       | 0 1 1 1                  | $((S \text{ xor } OV) \text{ or } Z) = 1$ | Less than or equal signed                     |
| GT                       | 1 1 1 1                  | $((S \text{ xor } OV) \text{ or } Z) = 0$ | Greater than signed                           |

APPENDIX C INSTRUCTION SET LIST

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<br>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<br>iiiiiiiiiiiiiiii | GR[reg2]←GR[reg1]AND zero-extend(imm16)   | 1                                    | 1 | 1 |       | 0  | 0 | x |     |  |
| Bcond    | disp9               | dddd1011ddcccc<br><b>Note 1</b>      | if conditions are satisfied<br>then PC←PC+sign-extend(disp9)  | When conditions<br>are satisfied     | 3 | 3 | 3     |    |   |   |     |  |
|          |                     |                                      |   | When conditions<br>are not satisfied | 1 | 1 | 1     |    |   |   |     |  |
| BSH      | reg2,reg3           | rrrrr11111100000<br>wwwww01101000010 | GR[reg3]←GR[reg2] (23:16)    GR[reg2] (31:24)   <br>GR[reg2] (7:0)    GR[reg2] (15:8)   | 1                                    | 1 | 1 | x     | 0  | x | x |     |  |
| BSW      | reg2,reg3           | rrrrr11111100000<br>wwwww01101000000 | GR[reg3]←GR[reg2] (7:0)    GR[reg2] (15:8)    GR<br>[reg2] (23:16)    GR[reg2] (31:24)  | 1                                    | 1 | 1 | x     | 0  | x | x |     |  |
| CALLT    | imm6                | 0000001000iiii                       | CTPC←PC+2(return PC)<br>CTPSW←PSW<br>adr←CTBP+zero-extend(imm6 logically shift left by 1)<br>PC←CTBP+zero-extend(Load-memory(adr,Halfword)) | 5                                    | 5 | 5 |       |    |   |   |     |  |
| CLR1     | bit#3, disp16[reg1] | 10bbb111110RRRRR<br>dddddddddddd     | adr←GR[reg1]+sign-extend(disp16)<br>Z flag←Not(Load-memory-bit(adr,bit#3))<br>Store-memory-bit(adr,bit#3,0)                                 | 3                                    | 3 | 3 |       |    |   |   | x   |  |
|          | reg2,[reg1]         | rrrrr111111RRRRR<br>0000000011100100 | adr←GR[reg1]<br>Z flag←Not(Load-memory-bit(adr,reg2))<br>Store-memory-bit(adr,reg2,0)   | 3                                    | 3 | 3 |       |    |   |   | x   |  |
| CMOV     | cccc,imm5,reg2,reg3 | rrrrr111111iiii<br>wwwww011000cccc0  | if conditions are satisfied<br>then GR[reg3]←sign-extended(imm5)<br>else GR[reg3]←GR[reg2]  | 1                                    | 1 | 1 |       |    |   |   |     |  |
|          | cccc,reg1,reg2,reg3 | rrrrr111111RRRRR<br>wwwww011001cccc0 | if conditions are satisfied<br>then GR[reg3]←GR[reg1]<br>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<br>0000000101000100  | PC←CTPC<br>PSW←CTPSW  | 4                                    | 4 | 4 | R     | R  | R | R | R   |  |
| DBRET    |                     | 000001111100000<br>0000000101000110  | PC←DBPC<br>PSW←DBPSW  | 4                                    | 4 | 4 | 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(return PC)<br>DBPSW←PSW<br>PSW.NP←1<br>PSW.EP←1<br>PSW.ID←1<br>PC←00000060H  | 4               | 4             | 4             |       |    |   |   |     |
| DI       |                    | 000001111100000<br>0000000101100000                        | PSW.ID←1   | 1               | 1             | 1             |       |    |   |   |     |
| DISPOSE  | imm5,list12        | 0000011001iiiiL<br>LLLLLLLLLLLL00000                       | sp←sp+zero-extend(imm5 logically shift left by 2)<br>GR[reg in list12]←Load-memory(sp,Word)<br>sp←sp+4<br>repeat 2 steps above until all regs in list12 is loaded                | n+1<br>Note 4   | n+1<br>Note 4 | n+1<br>Note 4 |       |    |   |   |     |
|          | imm5,list12,[reg1] | 0000011001iiiiL<br>LLLLLLLLLLLLRRRRR<br><b>Note 5</b>      | sp←sp+zero-extend(imm5 logically shift left by 2)<br>GR[reg in list12]←Load-memory(sp,Word)<br>sp←sp+4<br>repeat 2 steps above until all regs in list12 is loaded<br>PC←GR[reg1] | n+3<br>Note 4   | n+3<br>Note 4 | n+3<br>Note 4 |       |    |   |   |     |
| DIV      | reg1,reg2,reg3     | rrrrr11111RRRRR<br>wwwwww01011000000                       | GR[reg2]←GR[reg2]÷GR[reg1]<br>GR[reg3]←GR[reg2]%GR[reg1]   | 35              | 35            | 35            |       | x  | x | x |     |
| DIVH     | reg1,reg2          | rrrrr000010RRRRR   | GR[reg2]←GR[reg2]÷GR[reg1] <sup>Note 6</sup>   | 35              | 35            | 35            |       | x  | x | x |     |
|          | reg1,reg2,reg3     | rrrrr11111RRRRR<br>wwwwww01010000000                       | GR[reg2]←GR[reg2]÷GR[reg1] <sup>Note 6</sup><br>GR[reg3]←GR[reg2]%GR[reg1]   | 35              | 35            | 35            |       | x  | x | x |     |
| DIVHU    | reg1,reg2,reg3     | rrrrr11111RRRRR<br>wwwwww01010000010                       | GR[reg2]←GR[reg2]÷GR[reg1] <sup>Note 6</sup><br>GR[reg3]←GR[reg2]%GR[reg1]   | 34              | 34            | 34            |       | x  | x | x |     |
| DIVU     | reg1,reg2,reg3     | rrrrr11111RRRRR<br>wwwwww01011000010                       | GR[reg2]←GR[reg2]÷GR[reg1]<br>GR[reg3]←GR[reg2]%GR[reg1]   | 34              | 34            | 34            |       | x  | x | x |     |
| EI       |                    | 100001111100000<br>0000000101100000                        | PSW.ID←0   | 1               | 1             | 1             |       |    |   |   |     |
| HALT     |                    | 000001111100000<br>0000000100100000                        | Stop   | 1               | 1             | 1             |       |    |   |   |     |
| HSW      | reg2,reg3          | rrrrr11111100000<br>wwwwww01101000100                      | GR[reg3]←GR[reg2](15:0)    GR[reg2] (31:16)  | 1               | 1             | 1             | x     | 0  | x | x |     |
| JARL     | disp22,reg2        | rrrrr11110dddddd<br>ddddddddddddddd0<br><b>Note 7</b>      | GR[reg2]←PC+4<br>PC←PC+sign-extend(disp22)   | 3               | 3             | 3             |       |    |   |   |     |
| JMP      | [reg1]             | 0000000011RRRRR  | PC←GR[reg1]  | 4               | 4             | 4             |       |    |   |   |     |
| JR       | disp22             | 0000011110dddddd<br>ddddddddddddddd0<br><b>Note 7</b>      | PC←PC+sign-extend(disp22)  | 3               | 3             | 3             |       |    |   |   |     |
| LD.B     | disp16[reg1],reg2  | rrrrr111000RRRRR<br>ddddddddddddddd                        | adr←GR[reg1]+sign-extend(disp16)<br>GR[reg2]←sign-extend(Load-memory(adr,Byte))  | 1               | 1             | Note 11       |       |    |   |   |     |
| LD.BU    | disp16[reg1],reg2  | rrrrr11110bRRRRR<br>ddddddddddddddd1<br><b>Notes 8, 10</b> | adr←GR[reg1]+sign-extend(disp16)<br>GR[reg2]←zero-extend(Load-memory(adr,Byte))  | 1               | 1             | Note 11       |       |    |   |   |     |

APPENDIX C INSTRUCTION SET LIST

(3/6)

| Mnemonic                | Operand            | Opcode   | Operation  | Execution Clock |   |                | Flags |    |   |   |     |  |
|-------------------------|--------------------|--|--|-----------------|---|----------------|-------|----|---|---|-----|--|
|                         |                    |  |  | i               | r | l              | CY    | OV | S | Z | SAT |  |
| LD.H                    | disp16[reg1],reg2  | rrrrr111001RRRRR<br>dddddddddddddd0<br><b>Note 8</b>     | adr←GR[reg1]+sign-extend(disp16)<br>GR[reg2]←sign-extend(Load-memory(adr,Halfword))                              | 1               | 1 | <b>Note 11</b> |       |    |   |   |     |  |
| LDSR                    | reg2,regID         | rrrrr111111RRRRR<br>000000000100000<br><b>Note 12</b>    | SR[regID]←GR[reg2]   | 1               | 1 | 1              |       |    |   |   |     |  |
|                         |                    |  | Other than regID = PSW<br>regID = PSW  | 1               | 1 | 1              | ×     | ×  | × | × | ×   |  |
| LD.HU                   | disp16[reg1],reg2  | rrrrr111111RRRRR<br>dddddddddddddd1<br><b>Note 8</b>     | adr←GR[reg1]+sign-extend(disp16)<br>GR[reg2]←zero-extend(Load-memory(adr,Halfword))                              | 1               | 1 | <b>Note 11</b> |       |    |   |   |     |  |
| LD.W                    | disp16[reg1],reg2  | rrrrr111001RRRRR<br>dddddddddddddd1<br><b>Note 8</b>     | adr←GR[reg1]+sign-extend(disp16)<br>GR[reg2]←Load-memory(adr,Word)   | 1               | 1 | <b>Note 11</b> |       |    |   |   |     |  |
| MOV                     | reg1,reg2          | rrrrr000000RRRRR   | GR[reg2]←GR[reg1]  | 1               | 1 | 1              |       |    |   |   |     |  |
|                         | imm5,reg2          | rrrrr010000iiii  | GR[reg2]←sign-extend(imm5)   | 1               | 1 | 1              |       |    |   |   |     |  |
|                         | imm32,reg1         | 00000110001RRRRR<br>iiiiiiiiiiiiiiii<br>iiiiiiiiiiiiiiii | GR[reg1]←imm32   | 2               | 2 | 2              |       |    |   |   |     |  |
| MOVEA                   | imm16,reg1,reg2    | rrrrr110001RRRRR<br>iiiiiiiiiiiiiiii                     | GR[reg2]←GR[reg1]+sign-extend(imm16)   | 1               | 1 | 1              |       |    |   |   |     |  |
| MOVHI                   | imm16,reg1,reg2    | rrrrr110010RRRRR<br>iiiiiiiiiiiiiiii                     | GR[reg2]←GR[reg1]+(imm16 0 <sup>16</sup> )   | 1               | 1 | 1              |       |    |   |   |     |  |
| MUL <sup>Note 22</sup>  | reg1,reg2,reg3     | rrrrr111111RRRRR<br>wwwww01000100000                     | GR[reg3]    GR[reg2]←GR[reg2]xGR[reg1]   | 1               | 2 | 2              |       |    |   |   |     |  |
|                         | imm9,reg2,reg3     | rrrrr111111iiii<br>wwwww01001111100<br><b>Note 13</b>    | GR[reg3]    GR[reg2]←GR[reg2]xsign-extend(imm9)  | 1               | 2 | 2              |       |    |   |   |     |  |
| MULH                    | reg1,reg2          | rrrrr000111RRRRR   | GR[reg2]←GR[reg2] <sup>Note 6</sup> xGR[reg1] <sup>Note 6</sup>  | 1               | 1 | 2              |       |    |   |   |     |  |
|                         | imm5,reg2          | rrrrr010111iiii  | GR[reg2]←GR[reg2] <sup>Note 6</sup> xsign-extend(imm5)   | 1               | 1 | 2              |       |    |   |   |     |  |
| MULHI                   | imm16,reg1,reg2    | rrrrr110111RRRRR<br>iiiiiiiiiiiiiiii                     | GR[reg2]←GR[reg1] <sup>Note 6</sup> ximm16   | 1               | 1 | 2              |       |    |   |   |     |  |
| MULU <sup>Note 22</sup> | reg1,reg2,reg3     | rrrrr111111RRRRR<br>wwwww01000100010                     | GR[reg3]    GR[reg2]←GR[reg2]xGR[reg1]   | 1               | 2 | 2              |       |    |   |   |     |  |
|                         | imm9,reg2,reg3     | rrrrr111111iiii<br>wwwww0100111110<br><b>Note 13</b>     | GR[reg3]    GR[reg2]←GR[reg2]xzero-extend(imm9)  | 1               | 2 | 2              |       |    |   |   |     |  |
| 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] | 01bbb11110RRRRR<br>dddddddddddddd                        | adr←GR[reg1]+sign-extend(disp16)<br>Z flag←Not(Load-memory-bit(adr,bit#3))<br>Store-memory-bit(adr,bit#3,Z flag) | 3               | 3 | 3              |       |    |   |   | ×   |  |
|                         | reg2,[reg1]        | rrrrr111111RRRRR<br>0000000011100010                     | adr←GR[reg1]<br>Z flag←Not(Load-memory-bit(adr,reg2))<br>Store-memory-bit(adr,reg2,Z flag)                       | 3               | 3 | 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  | x | x |     |
| ORI      | imm16,reg1,reg2                           | rrrrr110100RRRRR<br>iiiiiiiiiiiiiiii                                  | GR[reg2]←GR[reg1]OR zero-extend(imm16)   | 1               | 1             | 1             |       | 0  | x | x |     |
| PREPARE  | list12,imm5                               | 0000011110iiiiL<br>LLLLLLLLLLLL00001                                  | Store-memory(sp-4,GR[reg in list12],Word)<br>sp←sp-4<br>repeat 1 step above until all regs in list12 is stored<br>sp←sp-zero-extend(imm5)                                | n+1<br>Note 4   | n+1<br>Note 4 | n+1<br>Note 4 |       |    |   |   |     |
|          | list12,imm5,<br>sp/imm <sup>Note 15</sup> | 0000011110iiiiL<br>LLLLLLLLLLLLff011<br>imm16/imm32<br><b>Note 16</b> | Store-memory(sp-4,GR[reg in list12],Word)<br>GR[reg in list 12]←Load-memory(sp,Word)<br>sp←sp+4<br>repeat 2 step above until all regs in list12 is loaded<br>PC←GR[reg1] | n+2<br>Note 4   | n+2<br>Note 4 | n+2<br>Note 4 |       |    |   |   |     |
| RETI     |   | 000001111100000<br>000000010100000                                    | if PSW.EP=1<br>then PC ←EIPC<br>PSW ←EIPSW<br>else if PSW.NP=1<br>then PC ←FEPC<br>PSW ←FEPSW<br>else PC ←EIPC<br>PSW ←EIPSW   | 4               | 4             | 4             | R     | R  | R | R | R   |
| SAR      | reg1,reg2                                 | rrrrr11111RRRRR<br>000000010100000                                    | GR[reg2]←GR[reg2]arithmetically shift right<br>by GR[reg1]   | 1               | 1             | 1             | x     | 0  | x | x |     |
|          | imm5,reg2                                 | rrrrr010101iiii   | GR[reg2]←GR[reg2]arithmetically shift right<br>by zero-extend(imm5)  | 1               | 1             | 1             | x     | 0  | x | x |     |
| SASF     | cccc,reg2                                 | rrrrr111110cccc<br>0000001000000000                                   | if conditions are satisfied<br>then GR[reg2]←(GR[reg2]Logically shift left by 1)<br>OR 00000001H<br>else GR[reg2]←(GR[reg2]Logically shift left by 1)<br>OR 00000000H    | 1               | 1             | 1             |       |    |   |   |     |
| SATADD   | reg1,reg2                                 | rrrrr000110RRRRR  | GR[reg2]←saturated(GR[reg2]+GR[reg1])  | 1               | 1             | 1             | x     | x  | x | x | x   |
|          | imm5,reg2                                 | rrrrr010001iiii   | GR[reg2]←saturated(GR[reg2]+sign-extend(imm5))   | 1               | 1             | 1             | x     | x  | x | x | x   |
| SATSUB   | reg1,reg2                                 | rrrrr000101RRRRR  | GR[reg2]←saturated(GR[reg2]-GR[reg1])  | 1               | 1             | 1             | x     | x  | x | x | x   |
| SATSUBI  | imm16,reg1,reg2                           | rrrrr110011RRRRR<br>iiiiiiiiiiiiiiii                                  | GR[reg2]←saturated(GR[reg1]-sign-extend(imm16))  | 1               | 1             | 1             | x     | x  | x | x | x   |
| SATSUBR  | reg1,reg2                                 | rrrrr000100RRRRR  | GR[reg2]←saturated(GR[reg1]-GR[reg2])  | 1               | 1             | 1             | x     | x  | x | x | x   |
| SETF     | cccc,reg2                                 | rrrrr111110cccc<br>0000000000000000                                   | If conditions are satisfied<br>then GR[reg2]←-00000001H<br>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<br>dddddddddddddd            | adr←GR[reg1]+sign-extend(disp16)<br>Z flag←Not(Load-memory-bit(adr,bit#3))<br>Store-memory-bit(adr,bit#3,1) | 3<br>Note 3     | 3<br>Note 3 | 3<br>Note 3 |       |    |   |   | ×   |  |
|          | reg2,[reg1]        | rrrrr111111RRRRR<br>0000000011100000          | adr←GR[reg1]<br>Z flag←Not(Load-memory-bit(adr,reg2))<br>Store-memory-bit(adr,reg2,1)                       | 3<br>Note 3     | 3<br>Note 3 | 3<br>Note 3 |       |    |   |   | ×   |  |
| SHL      | reg1,reg2          | rrrrr111111RRRRR<br>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<br>by zero-extend(imm5)  | 1               | 1           | 1           | ×     | 0  | × | × |     |  |
| SHR      | reg1,reg2          | rrrrr111111RRRRR<br>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<br>by zero-extend(imm5)   | 1               | 1           | 1           | ×     | 0  | × | × |     |  |
| SLD.B    | disp7[ep],reg2     | rrrrr0110dddddd                               | adr←ep+zero-extend(disp7)<br>GR[reg2]←sign-extend(Load-memory(adr,Byte))                                    | 1               | 1           | Note 9      |       |    |   |   |     |  |
| SLD.BU   | disp4[ep],reg2     | rrrrr0000110ddd<br>Note 18                    | adr←ep+zero-extend(disp4)<br>GR[reg2]←zero-extend(Load-memory(adr,Byte))                                    | 1               | 1           | Note 9      |       |    |   |   |     |  |
| SLD.H    | disp8[ep],reg2     | rrrrr1000dddddd<br>Note 19                    | adr←ep+zero-extend(disp8)<br>GR[reg2]←sign-extend(Load-memory(adr,Halfword))                                | 1               | 1           | Note 9      |       |    |   |   |     |  |
| SLD.HU   | disp5[ep],reg2     | rrrrr0000111ddd<br>Notes 18, 20               | adr←ep+zero-extend(disp5)<br>GR[reg2]←zero-extend(Load-memory(adr,Halfword))                                | 1               | 1           | Note 9      |       |    |   |   |     |  |
| SLD.W    | disp8[ep],reg2     | rrrrr1010dddddd0<br>Note 21                   | adr←ep+zero-extend(disp8)<br>GR[reg2]←Load-memory(adr,Word)   | 1               | 1           | Note 9      |       |    |   |   |     |  |
| SST.B    | reg2,disp7[ep]     | rrrrr0111dddddd                               | adr←ep+zero-extend(disp7)<br>Store-memory(adr,GR[reg2],Byte)  | 1               | 1           | 1           |       |    |   |   |     |  |
| SST.H    | reg2,disp8[ep]     | rrrrr1001dddddd<br>Note 19                    | adr←ep+zero-extend(disp8)<br>Store-memory(adr,GR[reg2],Halfword)  | 1               | 1           | 1           |       |    |   |   |     |  |
| SST.W    | reg2,disp8[ep]     | rrrrr1010dddddd1<br>Note 21                   | adr←ep+zero-extend(disp8)<br>Store-memory(adr,GR[reg2],Word)  | 1               | 1           | 1           |       |    |   |   |     |  |
| ST.B     | reg2,disp16[reg1]  | rrrrr111010RRRRR<br>dddddddddddddd            | adr←GR[reg1]+sign-extend(disp16)<br>Store-memory(adr,GR[reg2],Byte)   | 1               | 1           | 1           |       |    |   |   |     |  |
| ST.H     | reg2,disp16[reg1]  | rrrrr111011RRRRR<br>dddddddddddddd0<br>Note 8 | adr←GR[reg1]+sign-extend(disp16)<br>Store-memory(adr,GR[reg2],Halfword)                                     | 1               | 1           | 1           |       |    |   |   |     |  |
| ST.W     | reg2,disp16[reg1]  | rrrrr111011RRRRR<br>dddddddddddddd1<br>Note 8 | adr←GR[reg1]+sign-extend(disp16)<br>Store-memory(adr,GR[reg2],Word)   | 1               | 1           | 1           |       |    |   |   |     |  |
| STSR     | regID,reg2         | rrrrr111111RRRRR<br>0000000010000000          | GR[reg2]←SR[regID]  | 1               | 1           | 1           |       |    |   |   |     |  |



**APPENDIX C INSTRUCTION SET LIST**

(6/6)

| Mnemonic | Operand            | Opcode   | Operation   | Execution Clock |   |   | Flags  |        |        |   |     |
|----------|--------------------|--|---|-----------------|---|---|--------|--------|--------|---|-----|
|          |                    |  |   | i               | r | l | CY     | OV     | S      | Z | SAT |
| SUB      | reg1,reg2          | rrrrr001101RRRRR   | GR[reg2]←GR[reg2]-GR[reg1]  | 1               | 1 | 1 | x      | x      | x      | x |     |
| SUBR     | reg1,reg2          | rrrrr001100RRRRR   | GR[reg2]←GR[reg1]-GR[reg2]  | 1               | 1 | 1 | x      | x      | x      | x |     |
| SWITCH   | reg1               | 0000000010RRRRR  | adr←(PC+2) + (GR[reg1] logically shift left by 1)<br>PC←(PC+2) + (sign-extend<br>(Load-memory(adr,Halfword))<br>logically shift left by 1   | 5               | 5 | 5 |        |        |        |   |     |
| SXB      | reg1               | 0000000101RRRRR  | GR[reg1]←sign-extend<br>(GR[reg1] (7:0))  | 1               | 1 | 1 |        |        |        |   |     |
| SXH      | reg1               | 0000000111RRRRR  | GR[reg1]←sign-extend<br>(GR[reg1] (15:0))   | 1               | 1 | 1 |        |        |        |   |     |
| TRAP     | vector             | 0 0 0 0 0 1 1 1 1 1 1 i i i i<br>0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 | EIPC ←PC+4(return PC)<br>EIPSW ←PSW<br>ECR.EICC ←Exception code<br>(40H to 4FH, 50H to 5FH)<br>PSW.EP ←1<br>PSW.ID ←1<br>PC ←00000040H<br>(when vector is 00H to 0FH<br>(exception code: 40H to 4FH))<br>00000050H<br>(when vector is 10H to 1FH<br>(exception code: 50H to 5FH)) | 4               | 4 | 4 |        |        |        |   |     |
| TST      | reg1,reg2          | rrrrr001011RRRRR   | result←GR[reg2] AND GR[reg1]  | 1               | 1 | 1 |        | 0      | x      | x |     |
| TST1     | bit#3,disp16[reg1] | 11bbb111110RRRRR<br>ddddddddddddddd                              | adr←GR[reg1]+sign-extend(disp16)<br>Z flag←Not(Load-memory-bit(adr,bit#3))  | 3               | 3 | 3 | Note 3 | Note 3 | Note 3 |   | x   |
|          | reg2, [reg1]       | rrrrr111111RRRRR<br>0000000011100110                             | adr←GR[reg1]<br>Z flag←Not(Load-memory-bit(adr,reg2))   | 3               | 3 | 3 | Note 3 | Note 3 | Note 3 |   | x   |
| XOR      | reg1,reg2          | rrrrr001001RRRRR   | GR[reg2]←GR[reg2] XOR GR[reg1]  | 1               | 1 | 1 |        | 0      | x      | x |     |
| XORI     | imm16,reg1,reg2    | rrrrr110101RRRRR<br>i i i i i i i i i i i i i i i i              | GR[reg2]←GR[reg1] XOR zero-extend(imm16)  | 1               | 1 | 1 |        | 0      | x      | x |     |
| ZXB      | reg1               | 00000000100RRRRR   | GR[reg1]←zero-extend(GR[reg1] (7:0))  | 1               | 1 | 1 |        |        |        |   |     |
| ZXH      | reg1               | 00000000110RRRRR   | GR[reg1]←zero-extend(GR[reg1] (15:0))   | 1               | 1 | 1 |        |        |        |   |     |

- Notes**
1. dddddddd: Higher 8 bits of disp9.
  2. 4 if there is an instruction that rewrites the contents of the PSW immediately before.
  3. If there is no wait state (3 + the number of read access wait states).
  4. n is the total number of list12 load registers. (According to the number of wait states. Also, if there are no wait states, n is the total number of list12 registers. If n = 0, same operation as when n = 1)
  5. RRRRR: other than 00000.
  6. The lower halfword data only are valid.
  7. dddddddddddddddddddd: The higher 21 bits of disp22.
  8. dddddddddddddddd: The higher 15 bits of disp16.
  9. According to the number of wait states (1 if there are no wait states).
  10. b: bit 0 of disp16.
  11. According to the number of wait states (2 if there are no wait states).

**Notes 12.** In this instruction, for convenience of mnemonic description, the source register is made reg2, but the reg1 field is used in the opcode. Therefore, the meaning of register specification in the mnemonic description and in the opcode differs from other instructions.

rrrrr = reg1D specification

RRRRR = reg2 specification

13. iiii: Lower 5 bits of imm9.  
IIII: Higher 4 bits of imm9.
14. In the case of reg2 = reg3 (the lower 32 bits of the results are not written in the register) or reg3 = r0 (the higher 32 bits of the results are not written in the register), shortened by 1 clock.
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.
22. Do not make a combination that satisfies all the following conditions when using the "MUL reg1, reg2, reg3" instruction and "MULU reg1, reg2, reg3" instruction. Operation is not guaranteed when an instruction that satisfies the following conditions is executed.
  - Reg1 = reg3
  - Reg1 ≠ reg2
  - Reg1 ≠ r0
  - Reg3 ≠ r0

## APPENDIX D REVISION HISTORY

### D.1 Major Revisions in This Edition

| Page                   | Description  |
|------------------------|--|
| p. 183                 | Deletion of description in <b>5.3 (6) Clock monitor mode register (CLM)</b>  |
| p. 564                 | Modification of description in <b>Figure 10-6 Timing Chart of 6-Phase PWM Output Mode</b>  |
| p. 582                 | Modification of description in <b>Figure 10-21 Timing of Reflecting Rewritten Value</b>  |
| p. 600                 | Modification of description in <b>10.4.5 (3) When not tuning TAA<sub>n</sub></b>   |
| p. 608                 | Addition of description to <b>11.3 (1) Watchdog timer mode register (WDTM)</b>   |
| p. 609                 | Deletion of description in <b>11.4 Operation</b>   |
| p. 614                 | Modification of description in <b>Figure 12-3 Block Diagram of Operational Amplifier for Input Level Amplification and Overvoltage Detection Comparator in A/D Converter 0</b> |
| p. 615                 | Modification of description in <b>Figure 12-4 Block Diagram of Operational Amplifier for Input Level Amplification and Overvoltage Detection Comparator in A/D Converter 1</b> |
| p. 616                 | Addition of <b>Figure 12-5 CMPnCTL3 Register Selector Circuit Configuration</b>  |
| p. 624                 | Modification of description in <b>12.3 (1) A/D converter n scan mode register (ADnSCM)</b>   |
| pp. 645, 646           | Addition of description to <b>12.3 (16) Operational amplifier n control register 0 (OPnCTL0)</b>   |
| p. 648                 | Modification of description in <b>12.3 (17) Comparator n control register 0 (CMPnCTL0)</b>   |
| p. 679                 | Modification of description in <b>12.5 Internal Equivalent Circuit</b>   |
| p. 713                 | Modification of description in <b>Figure 13-12 Processing of Analog Input Pin</b>  |
| p. 810                 | Modification of figure in <b>Figure 16-4 Block Diagram of CSIB<sub>n</sub></b>   |
| p. 912                 | Modification of description in <b>17.15 (4) Procedure for starting or stopping I<sup>2</sup>C operation</b>  |
| p. 944                 | Addition of description to <b>18.6.3 Relationship between programmable wait and external wait</b>  |
| pp. 944, 945           | Modification of description in <b>Figure 18-1 Example of Inserting Wait States</b>   |
| p. 981                 | Addition of description to <b>19.12 (3) Bus arbitration for CPU</b>  |
| pp. 1036, 1037         | Modification of description in <b>22.2 (1) Reset source flag register (RESF)</b>   |
| p. 1040                | Deletion of description in <b>22.3 (3) Reset operation (LVIRE<sub>S</sub>) by low-voltage detector (LVI)</b>   |
| p. 1040                | Deletion of description in <b>22.3 (4) Reset operation (POCRE<sub>S</sub>) by power-on-clear circuit (POC)</b>   |
| p. 1042                | Modification of description in <b>23.3 (1) Low-voltage detection register (LVIM)</b>   |
| p. 1044                | Modification of description in <b>23.4.1 To use for internal reset signal</b>  |
| p. 1046                | Modification of description in <b>23.4.2 To use for interrupt</b>  |
| p. 1058                | Modification of description in <b>26.1.4 Cautions</b>  |
| p. 1070                | Modification of description in <b>26.3.2 Setting</b>   |
| p. 1072                | Addition of <b>27.2 Memory Configuration</b>   |
| p. 1073                | Addition of description to <b>27.3 Functional Overview</b>   |
| pp. 1094 to 1096, 1098 | Modification of description in <b>27.9 Rewriting by Self Programming</b>   |
| p. 1158                | Modification of description in <b>CHAPTER 29 PACKAGE DRAWINGS</b>  |

APPENDIX D REVISION HISTORY

## D.2 Revision History up to Previous Edition

The following table shows the revision history up to this edition. The “Applied to:” column indicates the chapters of each edition in which the revision was applied.

(1/3)

| Edition   | Description  | Applied to:  |
|---|--|--|
| 2nd edition   | <ul style="list-style-type: none"> <li>• Addition of product<br/>μPD70F3454F1-DA9-A</li> <li>• Addition of package<br/>161-pin plastic FBGA (10 × 10)</li> <li>• Addition of flash memory programmers<br/>PG-FP5 and FL-PR5</li> </ul> | Throughout   |
|   | Addition of <b>Note</b> to <b>1.3.4 Pin configuration (V850E/IG3)</b>  | <b>CHAPTER 1<br/>INTRODUCTION</b>                                    |
|   | Modification of description in <b>3.2.2 (2) NMI status saving registers (FEPC, FEPSW)</b>  | <b>CHAPTER 3<br/>CPU FUNCTION</b>                                    |
|   | Modification of <b>Cautions 1 and 2</b> in <b>Table 4-15 Noise Eliminator (1/2)</b>  | <b>CHAPTER 4<br/>PORT FUNCTIONS</b>                                  |
|   | Modification of <b>Cautions 1 and 2</b> in <b>Table 4-15 Noise Eliminator (2/2)</b>  |  |
|   | Addition of <b>Note</b> to <b>4.6 (1) Digital noise elimination 0 control register n (INTNFCn)</b>   |  |
|   | Modification of description in <b>6.6 (1) (a) Counter start operation</b>  | <b>CHAPTER 6<br/>16-BIT<br/>TIMER/EVENT<br/>COUNTER AA<br/>(TAA)</b> |
|   | Addition of <b>Caution 2</b> to <b>Figure 6-19 Register Setting for Operation in External Event Count Mode (2/2)</b>   |  |
|   | Modification of description in, addition of <b>Caution</b> to <b>Figure 6-24 Configuration in External Trigger Pulse Output Mode</b>   |  |
|   | Modification of description in <b>Figure 6-26 Setting of Registers in External Trigger Pulse Output Mode (1/2)</b>   |  |
|   | Modification of description in <b>Figure 6-26 Setting of Registers in External Trigger Pulse Output Mode (2/2)</b>   |  |
|   | Modification of description in, addition of <b>Caution</b> to <b>Figure 6-28 Configuration in One-Shot Pulse Output Mode</b>   |  |
|   | Modification of description in <b>Figure 6-30 Setting of Registers in One-Shot Pulse Output Mode (1/2)</b>   |  |
|   | Modification of description in, addition of <b>Caution</b> to <b>Figure 6-30 Setting of Registers in One-Shot Pulse Output Mode (2/2)</b>  |  |
|   | Addition of description to <b>Figure 6-31 Software Processing Flow in One-Shot Pulse Output Mode</b>   |  |
|   | Modification of description in, addition of <b>Caution</b> to <b>Figure 6-42 Configuration in Pulse Width Measurement Mode</b>   |  |
|   | Modification of description in <b>Figure 6-44 Register Setting in Pulse Width Measurement Mode (1/2)</b>   |  |
|   | Modification of description in <b>Figure 6-44 Register Setting in Pulse Width Measurement Mode (2/2)</b>   |  |
|   | Modification of description in <b>Figure 6-45 &lt;1&gt; Count operation start flow</b>   |  |
|   | Modification of description in <b>7.6 (1) (a) Counter start operation</b>  | <b>CHAPTER 7<br/>16-BIT<br/>TIMER/EVENT<br/>COUNTER AB<br/>(TAB)</b> |
| Addition of <b>Caution</b> to <b>Figure 7-26 Register Setting in One-Shot Pulse Output Mode (3/3)</b> |  |  |

APPENDIX D REVISION HISTORY

(2/3)

| Edition     | Description  | Applied to:  |
|-------------|--|--|
| 2nd edition | Modification of description in <b>8.6 (1) (a) Counter start operation</b>  | <b>CHAPTER 8<br/>16-BIT<br/>TIMER/EVENT<br/>COUNTER T (TMT)</b>            |
|             | Addition of <b>Caution</b> to <b>Figure 8-16 (f) TMTm capture/compare register 1 (TTmCCR1)</b>   |  |
|             | Addition of <b>Caution</b> to <b>8.6.2 (2) Operation timing in external event count mode</b>   |  |
|             | Addition of <b>Caution</b> to <b>Figure 8-27 Setting of Registers in One-Shot Pulse Output Mode (2/2)</b>  |  |
|             | Modification of figure in <b>Figure 8-28 Software Processing Flow in One-Shot Pulse Output Mode</b>  |  |
|             | Addition of <b>8.6.8 (1) PWM output of 0%/100%</b>   |  |
|             | Modification of description in, addition of <b>Note</b> to <b>10.3 (2) TABn option register 1 (TABnOPT1)</b>                                       | <b>CHAPTER 10<br/>MOTOR CONTROL<br/>FUNCTION</b>                           |
|             | Addition of <b>Note</b> to <b>12.3 (20) Comparator output digital noise elimination register nL, nF (CMPNFCnL, CMPNFCnF)</b>                       | <b>CHAPTER 12<br/>A/D CONVERTERS<br/>0 AND 1</b>                           |
|             | Modification of description in <b>12.6.8 A/D conversion result hysteresis characteristics</b>  |  |
|             | Modification of <b>Caution</b> in <b>13.3 (2) A/D converter 2 mode register 1 (AD2M1)</b>  | <b>CHAPTER 13<br/>A/D CONVERTER 2</b>                                      |
|             | Modification of description in <b>13.7 (11) A/D conversion result hysteresis characteristics</b>   |  |
|             | Modification of description in <b>Figure 14-4 Block Diagram of UARTAn</b>  | <b>CHAPTER 14<br/>ASYNCHRONOUS<br/>SERIAL INTERFACE<br/>A (UARTA)</b>      |
|             | Modification of description in <b>Figure 14-8 Continuous Transmission Operation Timing</b>   |  |
|             | Modification of figure in <b>Figure 17-21 (a) Start condition ~ address</b>  | <b>CHAPTER 17<br/>I<sup>2</sup>C BUS</b>                                   |
|             | Modification of figure in <b>Figure 17-21 (b) Data</b>   |  |
|             | Modification of figure in <b>Figure 17-21 (c) Stop condition</b>   |  |
|             | Modification of figure in <b>Figure 17-22 (a) Start condition ~ address</b>  |  |
|             | Modification of figure in <b>Figure 17-22 (b) Data</b>   |  |
|             | Modification of figure in <b>Figure 17-22 (c) Stop condition</b>   |  |
|             | Modification of <b>Caution 1</b> in <b>19.3.6 DMA trigger factor registers 0 to 3 (DTFR0 to DTFR3)</b>   | <b>CHAPTER 19<br/>DMA FUNCTIONS<br/>(DMA<br/>CONTROLLER)</b>               |
|             | Modification of <b>Note</b> in <b>Table 19-1 DMA Start Factors (1/2)</b>   |  |
|             | Modification of description in <b>19.12 (3) Bus arbitration for CPU</b>  |  |
|             | Addition of <b>19.12 (4) DMA start factors</b>   |  |
|             | Modification of <b>Caution 2</b> in <b>20.3.4 Interrupt control registers (xxICn)</b>  | <b>CHAPTER 20<br/>INTERRUPT/<br/>EXCEPTION<br/>PROCESSING<br/>FUNCTION</b> |
|             | Deletion of description in <b>26.3.2 Setting</b>   | <b>CHAPTER 26<br/>ON-CHIP DEBUG<br/>FUNCTION</b>                           |
|             | Addition of <b>28.1.4 (i) Murata Mfg. Co., Ltd.: Ceramic resonator (TA = -40 to +85°C)</b>   | <b>CHAPTER 28<br/>ELECTRICAL<br/>SPECIFICATIONS</b>                        |
|             | Addition of <b>28.2.4 (i) Murata Mfg. Co., Ltd.: Ceramic resonator (TA = -40 to +85°C) (Products other than <math>\mu</math>PD70F3454F1-DA9-A)</b> |  |
|             | Addition of <b>Caution</b> to <b>28.2.8 (4) (b) Read cycle (CLKOUT synchronous)</b>  |  |
|             | Addition of <b>Caution</b> to <b>28.2.8 (4) (d) Write cycle (CLKOUT synchronous)</b>   |  |

**APPENDIX D REVISION HISTORY**

---

(3/3)

| Edition     | Description   | Applied to:  |
|-------------|---|--|
| 2nd edition | Addition of description to <b>CHAPTER 30 RECOMMENDED SOLDERING CONDITIONS</b> | <b>CHAPTER 30<br/>RECOMMENDED<br/>SOLDERING<br/>CONDITIONS</b> |
|             | Addition of <b>APPENDIX D REVISION HISTORY</b>                                | <b>APPENDIX D<br/>REVISION HISTORY</b>                         |



## X-ON Electronics

Largest Supplier of Electrical and Electronic Components

*Click to view similar products for [8-bit Microcontrollers - MCU category](#):*

*Click to view products by [Renesas manufacturer](#):*

Other Similar products are found below :

[CY8C20524-12PVXIT](#) [CY8C28433-24PVXIT](#) [MB95F012KPFT-G-SNE2](#) [MB95F013KPMC-G-SNE2](#) [MB95F263KPF-G-SNE2](#)  
[MB95F264KPFT-G-SNE2](#) [MB95F398KPMC-G-SNE2](#) [MB95F478KPMC2-G-SNE2](#) [MB95F562KPF-G-SNE2](#) [MB95F564KPF-G-SNE2](#)  
[MB95F634KPMC-G-SNE2](#) [MB95F636KWQN-G-SNE1](#) [MB95F696KPMC-G-SNE2](#) [MB95F698KPMC1-G-SNE2](#) [MB95F698KPMC2-G-SNE2](#) [MB95F698KPMC-G-SNE2](#) [MB95F818KPMC1-G-SNE2](#) [MC908JK1ECDWER](#) [MC9S08PA32AVLD](#) [MC9S08PT60AVLD](#)  
[R5F1076CMSPV0](#) [R5F5631ECDFBV0](#) [C8051F389-B-GQ](#) [C8051F392-A-GMR](#) [ISD-ES1600\\_USB\\_PROG](#) [901015X](#) [SC705C8AE0VFBE](#)  
[STM8TL53G4U6](#) [PIC16F877-04/P-B](#) [R5F10Y17ASP#30](#) [CY8C3MFIDOCK-125](#) [403708R](#) [MB95F354EPF-G-SNE2](#) [MB95F564KPFT-G-SNE2](#) [MB95F564KWQN-G-SNE1](#) [MB95F636KP-G-SH-SNE2](#) [MB95F636KPMC-G-SNE2](#) [MB95F694KPMC-G-SNE2](#) [MB95F778JPMC1-G-SNE2](#) [MB95F818KPMC-G-SNE2](#) [MC908QY8CDWER](#) [MC9S08PT16AVLD](#) [MC9S08PT32AVLH](#) [MC9S08PT60AVLC](#)  
[MC9S08PT60AVLH](#) [C8051F500-IQR](#) [LC87F0G08AUJA-AH](#) [CP8361BT](#) [STM8S207C6T3](#) [CG8421AF](#)