

Hardware

# V850E/PHC3

32-bit Single-Chip Microcontroller

μPD70F3485

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (<http://www.renesas.com>).

## 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.

## General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

### 1. Handling of unused Pins

Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

- The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

### 2. Processing at power-on

The state of the product is undefined at the moment when power is supplied.

- The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

### 3. Prohibition of access to reserved addresses

Access to reserved addresses is prohibited.

- The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

### 4. Clock signals

After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

- When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

### 5. Differences between products

Before changing from one product to another, i.e. to one with a different part number, confirm that the change will not lead to problems.

- The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.



## Preface

<b>Readers</b>	This manual is intended for users who want to understand the functions of the concerned microcontrollers.
<b>Purpose</b>	This manual presents the hardware manual for the concerned microcontrollers.
<b>Organization</b>	This system specification describes the following sections: <ul style="list-style-type: none"><li>• Pin function</li><li>• CPU function</li><li>• Internal peripheral function</li></ul>
<b>Module instances</b>	These microcontrollers may contain several instances of a dedicated module. In general the different instances of such modules are identified by the index “n”, where “n” counts from 0 to the number of instances minus one.
<b>Legend</b>	Symbols and notation are used as follows: <ul style="list-style-type: none"><li>• Weight in data notation: Left is high order column, right is low order column</li><li>• Active low notation: <math>\overline{\text{xxx}}</math> (pin or signal name is over-scored) or /xxx (slash before signal name)</li><li>• Memory map address: High order at high stage and low order at low stage</li></ul>
<b>Note</b>	Additional remark or tip
<b>Caution</b>	Item deserving extra attention
<b>Numeric notation:</b>	<ul style="list-style-type: none"><li>• Binary: xxxx or <math>\text{xxx}_B</math></li><li>• Decimal: xxxx</li><li>• Hexadecimal: <math>\text{xxxx}_H</math> or 0x xxxx</li></ul>
<b>Prefixes</b>	representing powers of 2 (address space, memory capacity): <ul style="list-style-type: none"><li>• K (kilo): <math>2^{10} = 1024</math></li><li>• M (mega): <math>2^{20} = 1024^2 = 1,048,576</math></li><li>• G (giga): <math>2^{30} = 1024^3 = 1,073,741,824</math></li></ul>
<b>Register contents:</b>	X, x = don't care
<b>Diagrams</b>	Block diagrams do not necessarily show the exact wiring in hardware but the functional structure.  Timing diagrams are for functional explanation purposes only, without any relevance to the real hardware implementation.
<b>Further Information</b>	For further information see <a href="http://www.renesas.eu/">http://www.renesas.eu/</a> .

**SuperFlash<sup>(R)</sup>** SuperFlash<sup>(R)</sup> is a registered trademark of Silicon Storage Technology, Inc. in several countries including the United States and Japan.

This product uses SuperFlash<sup>(R)</sup> technology licensed from Silicon Technology, Inc.

# Table of Contents

<b>Table of Contents</b> .....	7
<b>Chapter 1 Introduction</b> .....	17
<b>1.1 General</b> .....	17
<b>1.2 Features Summary</b> .....	18
<b>1.3 Description</b> .....	20
1.3.1 Internal units.....	21
1.3.2 Structure of the manual.....	22
<b>Chapter 2 Pin Functions</b> .....	23
<b>2.1 Overview</b> .....	23
2.1.1 Description.....	24
2.1.2 Terms.....	26
2.1.3 Noise elimination.....	26
<b>2.2 Port Group Configuration Registers</b> .....	27
2.2.1 Overview.....	27
2.2.2 Pin function configuration.....	28
2.2.3 Pin data input/output.....	32
2.2.4 Emergency shut-off configuration.....	34
2.2.5 Write enable register.....	38
<b>2.3 Port Group Configuration</b> .....	40
2.3.1 Port group configuration lists.....	40
2.3.2 Non-port pins.....	42
2.3.3 Port Types Diagrams.....	44
2.3.4 Alphabetic pin function list.....	45
2.3.5 Port group 0.....	51
2.3.6 Port group 1.....	52
2.3.7 Port group 2.....	53
2.3.8 Port group 3.....	54
2.3.9 Port group 4.....	55
2.3.10 Port group 5.....	56
2.3.11 Port group 6.....	57
2.3.12 Port group 7.....	58
2.3.13 Port group 8.....	59
2.3.14 Port group 9.....	60
2.3.15 Port group 10.....	61
2.3.16 Port group 11.....	62
<b>2.4 Noise Elimination</b> .....	63
2.4.1 Analog filtered inputs.....	63
2.4.2 Digitally filtered inputs.....	64
<b>2.5 Pin Functions during and after Reset</b> .....	67
<b>2.6 Recommended Connection of unused Pins</b> .....	68
<b>2.7 Package Pins Assignment</b> .....	69
<b>Chapter 3 CPU System Functions</b> .....	71
<b>3.1 Overview</b> .....	71
3.1.1 Description.....	72

<b>3.2</b>	<b>CPU Register Set</b> .....	73
3.2.1	General purpose registers (r0 to r31) .....	74
3.2.2	System register set .....	75
3.2.3	Floating-point arithmetic registers .....	82
<b>3.3</b>	<b>Operation Modes</b> .....	84
3.3.1	Normal operation mode .....	84
3.3.2	Flash programming mode .....	84
<b>3.4</b>	<b>Address Space</b> .....	85
3.4.1	CPU address space and physical address space .....	85
3.4.2	Program and data space .....	87
<b>3.5</b>	<b>Memory</b> .....	89
3.5.1	Memory areas .....	89
<b>3.6</b>	<b>RAM ECC error detection</b> .....	91
3.6.1	RAM ECC initialization .....	91
3.6.2	RAM ECC registers .....	92
3.6.3	RAM ECC function check .....	95
3.6.4	Recommended use of data address space .....	96
<b>3.7</b>	<b>Write Protected Registers</b> .....	97

## **Chapter 4 Interrupt Controller (INTC)** .....

<b>4.1</b>	<b>Features</b> .....	99
<b>4.2</b>	<b>Non-Maskable Interrupts</b> .....	105
4.2.1	Operation .....	106
4.2.2	Restore .....	108
4.2.3	Non-maskable interrupt status flag (NP) .....	109
4.2.4	NMI control .....	109
<b>4.3</b>	<b>Maskable Interrupts</b> .....	110
4.3.1	Operation .....	110
4.3.2	Restore .....	112
4.3.3	Priorities of maskable interrupts .....	113
4.3.4	xxICn - Maskable interrupts control register .....	117
4.3.5	IMR0 to IMR7 - Interrupt mask registers .....	120
4.3.6	ISPR - In-service priority register .....	122
4.3.7	Maskable interrupt status flag (ID) .....	123
4.3.8	External maskable interrupts .....	123
<b>4.4</b>	<b>Edge and Level Detection Configuration</b> .....	124
<b>4.5</b>	<b>Interrupt Sharing</b> .....	127
<b>4.6</b>	<b>Software Exception</b> .....	131
4.6.1	Operation .....	131
4.6.2	Restore .....	132
4.6.3	Exception status flag (EP) .....	133
<b>4.7</b>	<b>Exception Trap</b> .....	134
4.7.1	Illegal opcode definition .....	134
4.7.2	Debug trap .....	136
<b>4.8</b>	<b>Multiple Interrupt Processing Control</b> .....	138
<b>4.9</b>	<b>Interrupt Response Time</b> .....	140
<b>4.10</b>	<b>Periods in Which Interrupts Are Not Acknowledged</b> .....	141

## **Chapter 5 CRC Function (CRC)** .....

<b>5.1</b>	<b>Features</b> .....	142
5.1.1	DATA-CRC .....	142
<b>5.2</b>	<b>Configuration</b> .....	143
<b>5.3</b>	<b>CRC Registers</b> .....	144
<b>5.4</b>	<b>Operation</b> .....	147
5.4.1	DATA-CRC .....	147
<b>Chapter 6 Clock Generator</b> .....		148
<b>6.1</b>	<b>Overview</b> .....	148
6.1.1	Features .....	148
6.1.2	Description .....	149
<b>6.2</b>	<b>Control Registers</b> .....	151
<b>6.3</b>	<b>Clock Monitor</b> .....	152
6.3.1	Clock Monitor registers .....	152
6.3.2	Operation of the Clock Monitor .....	154
<b>6.4</b>	<b>Power Save Control</b> .....	156
6.4.1	HALT mode .....	157
<b>Chapter 7 Flash Memory</b> .....		159
<b>7.1</b>	<b>Code Flash Memory Overview</b> .....	160
7.1.1	Code flash memory features .....	160
7.1.2	Code flash memory mapping .....	160
7.1.3	Code flash memory functional outline .....	162
7.1.4	Code flash memory erasure and rewrite .....	164
7.1.5	Code flash memory error detection .....	166
<b>7.2</b>	<b>Data Flash Memory</b> .....	168
7.2.1	Data flash memory features .....	168
7.2.2	Data flash memory map .....	168
7.2.3	Data flash control register .....	169
7.2.4	Data flash reading .....	169
7.2.5	Data flash writing .....	169
<b>7.3</b>	<b>Flash Programming with Flash Programmer</b> .....	170
7.3.1	Programming environment .....	170
7.3.2	Communication mode .....	171
7.3.3	Pin connection with flash programmer PG-FP5 .....	173
7.3.4	Flash memory programming control .....	175
<b>7.4</b>	<b>Code Flash Self-Programming</b> .....	181
7.4.1	Self-programming enable .....	182
7.4.2	Self-programming library functions .....	182
7.4.3	Secure self-programming (boot cluster swapping) .....	183
7.4.4	Interrupt handling during flash self-programming .....	186
<b>Chapter 8 Data Protection and Security</b> .....		187
<b>8.1</b>	<b>Overview</b> .....	187
<b>8.2</b>	<b>N-Wire Debug Interface Protection</b> .....	187
<b>8.3</b>	<b>Flash Programmer and Self-Programming Protection</b> .....	189
<b>Chapter 9 Bus and Memory Control (BCU, MEMC)</b> .....		192
<b>9.1</b>	<b>Overview</b> .....	192

9.1.1	Description	193
9.1.2	Memory areas/banks and chip select signals	194
9.1.3	Peripheral I/O areas	197
9.1.4	Data flash access	199
9.1.5	CRC module access	200
9.1.6	Boundary operation conditions	201
9.1.7	Initializing access to the Data flash memory and the CRC module	202
<b>9.2</b>	<b>Registers</b>	<b>203</b>
9.2.1	BCU registers	204
9.2.2	Memory Controller registers	206

## **Chapter 10 DMA Functions (DMA Controller)** . . . . . 211

<b>10.1</b>	<b>Overview</b>	<b>211</b>
10.1.1	Principle of operation	211
10.1.2	Forcible termination of DMA transfer	212
10.1.3	DMA interrupt function	213
<b>10.2</b>	<b>Registers</b>	<b>214</b>
<b>10.3</b>	<b>Transfer Details</b>	<b>223</b>
10.3.1	DMA transfer from A/D Converter	223
10.3.2	DMA transfer to Timer S	227
10.3.3	DMA transfer to/from serial interfaces	232
10.3.4	Example of forcible termination of DMA transfer	240

## **Chapter 11 Asynchronous Serial Interface (UARTC)** . . . . . 241

<b>11.1</b>	<b>Features</b>	<b>241</b>
<b>11.2</b>	<b>Configuration</b>	<b>242</b>
<b>11.3</b>	<b>UARTC Registers</b>	<b>245</b>
<b>11.4</b>	<b>Interrupt Request Signals</b>	<b>255</b>
<b>11.5</b>	<b>Operation</b>	<b>256</b>
11.5.1	Data format	256
11.5.2	SBF transmission/reception format	258
11.5.3	SBF transmission	260
11.5.4	SBF reception	260
11.5.5	UART transmission	262
11.5.6	Continuous transmission procedure	263
11.5.7	UART reception	265
11.5.8	Reception errors	267
11.5.9	Parity types and operations	267
11.5.10	Receive data noise filter	269
<b>11.6</b>	<b>Baud Rate Generator</b>	<b>270</b>
<b>11.7</b>	<b>Cautions</b>	<b>277</b>

## **Chapter 12 Clocked Serial Interface (CSIB)** . . . . . 278

<b>12.1</b>	<b>Features</b>	<b>278</b>
<b>12.2</b>	<b>Configuration</b>	<b>279</b>
<b>12.3</b>	<b>CSIB Control Registers</b>	<b>281</b>
<b>12.4</b>	<b>Operation</b>	<b>288</b>
12.4.1	Single transfer mode (master mode, transmission/reception mode)	288
12.4.2	Single transfer mode (master mode, reception mode)	290

12.4.3	Continuous mode (master mode, transmission/reception mode) . . . . .	291
12.4.4	Continuous mode (master mode, reception mode) . . . . .	293
12.4.5	Continuous reception mode (error) . . . . .	294
12.4.6	Continuous mode (slave mode, transmission/reception mode) . . . . .	295
12.4.7	Continuous mode (slave mode, reception mode) . . . . .	297
12.4.8	Clock timing . . . . .	298
<b>12.5</b>	<b>Output Pins</b> . . . . .	<b>300</b>
<b>12.6</b>	<b>Operation Flow</b> . . . . .	<b>301</b>
<b>12.7</b>	<b>Baud Rate Generator</b> . . . . .	<b>307</b>
12.7.1	Overview . . . . .	307
12.7.2	Baud rate generator registers . . . . .	308
12.7.3	Baud rate generation . . . . .	310
<b>12.8</b>	<b>Cautions</b> . . . . .	<b>311</b>
12.8.1	CSIB operation stop . . . . .	311

## Chapter 13 Enhanced Queued Clocked Serial Interface (CSIE) . .

312

<b>13.1</b>	<b>Features</b> . . . . .	<b>312</b>
13.1.1	Queued CSI block diagram . . . . .	313
13.1.2	Input/Output pins . . . . .	314
<b>13.2</b>	<b>Queued CSI Control Registers</b> . . . . .	<b>315</b>
<b>13.3</b>	<b>Explanation of Queued CSI Functions</b> . . . . .	<b>335</b>
13.3.1	Transmit buffer . . . . .	335
13.3.2	Serial data direction select function . . . . .	336
13.3.3	Data length select function . . . . .	337
13.3.4	Slave mode . . . . .	337
13.3.5	Master mode . . . . .	338
13.3.6	Transmission clock select function . . . . .	338
13.3.7	Description of the single transfer mode . . . . .	339
13.3.8	Description of the block transfer mode . . . . .	341
13.3.9	Description of the operation modes . . . . .	342
13.3.10	Additional timing and delay selections . . . . .	343
13.3.11	Default pin levels . . . . .	346
13.3.12	Transmit buffer overflow interrupt signal (INTCEnOF) . . . . .	347
13.3.13	Enhanced chip select timing . . . . .	348
<b>13.4</b>	<b>Operating Procedure</b> . . . . .	<b>355</b>
13.4.1	Single transfer mode (master mode, transmit only mode) . . . . .	355
13.4.2	Single transfer mode (master mode, receive only mode) . . . . .	356
13.4.3	Single transfer mode (master mode, transmit/receive mode) . . . . .	357
13.4.4	Single transfer mode (slave mode, transmit only mode) . . . . .	358
13.4.5	Single transfer mode (slave mode, receive only mode) . . . . .	359
13.4.6	Single transfer mode (slave mode, transmit/receive mode) . . . . .	360
13.4.7	Block transfer mode (master mode, transmit only mode) . . . . .	361
13.4.8	Block transfer mode (master mode, receive only mode) . . . . .	362
13.4.9	Block transfer mode (master mode, transmit/receive mode) . . . . .	363
13.4.10	Block transfer mode (slave mode, transmit only mode) . . . . .	364
13.4.11	Block transfer mode (slave mode, receive only mode) . . . . .	365
13.4.12	Block transfer mode (slave mode, transmit/receive mode) . . . . .	366
<b>13.5</b>	<b>Cautions</b> . . . . .	<b>367</b>

13.5.1	Inconsistent read data in master mode . . . . .	367
13.5.2	Transmit/receive enable after CEnTX0 write . . . . .	367
<b>Chapter 14 CAN Controller (CAN)</b> . . . . .		368
<b>14.1</b>	<b>Features</b> . . . . .	368
14.1.1	Overview of functions . . . . .	369
14.1.2	Configuration . . . . .	370
<b>14.2</b>	<b>CAN Protocol</b> . . . . .	371
14.2.1	Frame format . . . . .	371
14.2.2	Frame types . . . . .	372
14.2.3	Data frame and remote frame . . . . .	372
14.2.4	Error frame . . . . .	379
14.2.5	Overload frame . . . . .	380
<b>14.3</b>	<b>Functions</b> . . . . .	381
14.3.1	Determining bus priority . . . . .	381
14.3.2	Bit stuffing . . . . .	381
14.3.3	Multi masters . . . . .	382
14.3.4	Multi cast . . . . .	382
14.3.5	CAN sleep mode/CAN stop mode function . . . . .	382
14.3.6	Error control function . . . . .	382
14.3.7	Baud rate control function . . . . .	389
<b>14.4</b>	<b>Connection with Target System</b> . . . . .	392
<b>14.5</b>	<b>Internal Registers of CAN Controller</b> . . . . .	393
14.5.1	CAN module register and message buffer addresses . . . . .	393
14.5.2	CAN Controller configuration . . . . .	394
14.5.3	CAN registers overview . . . . .	395
14.5.4	Register bit configuration . . . . .	397
<b>14.6</b>	<b>Bit Set/Clear Function</b> . . . . .	400
<b>14.7</b>	<b>Control Registers</b> . . . . .	402
<b>14.8</b>	<b>CAN Controller Initialization</b> . . . . .	438
14.8.1	Initialization of CAN module . . . . .	438
14.8.2	Initialization of message buffer . . . . .	438
14.8.3	Redefinition of message buffer . . . . .	438
14.8.4	Transition from initialization mode to operation mode . . . . .	440
14.8.5	Resetting error counter CnERC of CAN module . . . . .	441
<b>14.9</b>	<b>Message Reception</b> . . . . .	442
14.9.1	Message reception . . . . .	442
14.9.2	Receive data read . . . . .	443
14.9.3	Receive history list function . . . . .	444
14.9.4	Mask function . . . . .	446
14.9.5	Multi buffer receive block function . . . . .	448
14.9.6	Remote frame reception . . . . .	449
<b>14.10</b>	<b>Message Transmission</b> . . . . .	450
14.10.1	Message transmission . . . . .	450
14.10.2	Transmit history list function . . . . .	452
14.10.3	Automatic block transmission (ABT) . . . . .	454
14.10.4	Transmission abort process . . . . .	456
14.10.5	Remote frame transmission . . . . .	457
<b>14.11</b>	<b>Power Saving Modes</b> . . . . .	458
14.11.1	CAN sleep mode . . . . .	458



14.11.2	CAN stop mode . . . . .	461
14.11.3	Example of using power saving modes . . . . .	462
<b>14.12</b>	<b>Interrupt Function . . . . .</b>	<b>463</b>
<b>14.13</b>	<b>Diagnosis Functions and Special Operational Modes . . . . .</b>	<b>464</b>
14.13.1	Receive-only mode . . . . .	464
14.13.2	Single-shot mode . . . . .	465
14.13.3	Self-test mode . . . . .	466
14.13.4	Receive/transmit operation in each operation mode . . . . .	467
<b>14.14</b>	<b>Time Stamp Function . . . . .</b>	<b>468</b>
14.14.1	Time stamp function . . . . .	468
<b>14.15</b>	<b>Baud Rate Settings . . . . .</b>	<b>469</b>
14.15.1	Baud rate setting conditions . . . . .	469
14.15.2	Representative examples of baud rate settings . . . . .	473
<b>14.16</b>	<b>Operation of CAN Controller . . . . .</b>	<b>477</b>
 <b>Chapter 15 A/D Converter (ADC) . . . . .</b>		<b>506</b>
<b>15.1</b>	<b>Features . . . . .</b>	<b>506</b>
<b>15.2</b>	<b>Configuration . . . . .</b>	<b>507</b>
<b>15.3</b>	<b>Control Registers . . . . .</b>	<b>510</b>
<b>15.4</b>	<b>Operation . . . . .</b>	<b>519</b>
15.4.1	Basic operation . . . . .	519
15.4.2	Operation mode and trigger mode . . . . .	520
<b>15.5</b>	<b>Operation in A/D Trigger Mode . . . . .</b>	<b>525</b>
15.5.1	Select mode operation . . . . .	525
15.5.2	Scan mode operation . . . . .	527
<b>15.6</b>	<b>Operation in Timer Trigger Mode . . . . .</b>	<b>529</b>
15.6.1	Select mode operation . . . . .	529
15.6.2	Scan mode operation . . . . .	532
<b>15.7</b>	<b>Operation in External Trigger Mode . . . . .</b>	<b>534</b>
15.7.1	Select mode operation . . . . .	534
15.7.2	Scan mode operation . . . . .	536
<b>15.8</b>	<b>Precautions . . . . .</b>	<b>538</b>
 <b>Chapter 16 16-Bit Timer/Event Counter AA (TAA) . . . . .</b>		<b>541</b>
<b>16.1</b>	<b>Features . . . . .</b>	<b>541</b>
<b>16.2</b>	<b>Function Outline . . . . .</b>	<b>541</b>
<b>16.3</b>	<b>Configuration . . . . .</b>	<b>542</b>
<b>16.4</b>	<b>Input Selection Registers . . . . .</b>	<b>549</b>
<b>16.5</b>	<b>Control Registers . . . . .</b>	<b>552</b>
<b>16.6</b>	<b>Operation . . . . .</b>	<b>564</b>
16.6.1	Anytime write and reload . . . . .	565
16.6.2	Interval timer mode (TAA <sub>n</sub> MD2 to TAA <sub>n</sub> MD0 = 000 <sub>B</sub> ) . . . . .	569
16.6.3	External event counter mode (TAA <sub>n</sub> MD2 to TAA <sub>n</sub> MD0 = 001 <sub>B</sub> ) . . . . .	572
16.6.4	External trigger pulse mode (TAA <sub>n</sub> MD2 to TAA <sub>n</sub> MD0 = 010 <sub>B</sub> ) . . . . .	576
16.6.5	One-shot pulse mode (TAA <sub>n</sub> MD2 to TAA <sub>n</sub> MD0 = 011 <sub>B</sub> ) . . . . .	579
16.6.6	PWM mode (TAA <sub>n</sub> MD2 to TAA <sub>n</sub> MD0 = 100 <sub>B</sub> ) . . . . .	582
16.6.7	Free-running mode (TAA <sub>n</sub> MD2 to TAA <sub>n</sub> MD0 = 101 <sub>B</sub> ) . . . . .	587
16.6.8	Pulse width measurement mode (TAA <sub>n</sub> MD2 to TAA <sub>n</sub> MD0 = 110 <sub>B</sub> )	593

16.6.9	32-bit Capture in free-running cascade mode . . . . .	600
16.6.10	Specific TAAAnCCRm overflow value write . . . . .	605
<b>16.7</b>	<b>Timer Synchronization Operation Function . . . . .</b>	<b>611</b>
<b>Chapter 17 16-Bit Timer/Event Counter S (TMS) . . . . .</b>		<b>613</b>
<b>17.1</b>	<b>Features . . . . .</b>	<b>613</b>
<b>17.2</b>	<b>Configuration . . . . .</b>	<b>615</b>
<b>17.3</b>	<b>Control Register . . . . .</b>	<b>627</b>
<b>17.4</b>	<b>Basic Operation . . . . .</b>	<b>689</b>
17.4.1	16-bit counter basic operation . . . . .	689
17.4.2	Compare register rewriting . . . . .	691
17.4.3	Output list in each mode . . . . .	705
<b>17.5</b>	<b>Match Interrupt . . . . .</b>	<b>713</b>
17.5.1	Caution on compare match interrupt . . . . .	715
<b>17.6</b>	<b>Flags . . . . .</b>	<b>716</b>
17.6.1	Count-up flags (TSnCUF, TSnSUF) . . . . .	717
17.6.2	Positive/negative-phase simultaneous active detection flag (TSnTBF) . . . . .	720
17.6.3	Reload request flag (TSnRSF) . . . . .	721
17.6.4	Noise detection flag (TSnNDF) . . . . .	722
17.6.5	Pattern order detection flag (TSnTSF) . . . . .	723
17.6.6	Pattern error detection flag (TSnPEF) . . . . .	725
17.6.7	Pattern reverse detection flag (TSnPRF) . . . . .	726
17.6.8	TAPTSn2 to TAPTSn0 pin abnormal toggle detection flag (TSnPTF) . . . . .	728
17.6.9	TSnSTCI0, TSnSTCI1 signal simultaneous trigger detection flag (TSnTDF) . . . . .	730
17.6.10	Pattern phase difference detection flag (TSnPPF) . . . . .	731
17.6.11	Overflow flag (TSnOVF) . . . . .	732
17.6.12	Timer output pattern flags (TSnOPF2 to TSnOPF0) . . . . .	733
17.6.13	Pattern switch detection flag (TSnESG) . . . . .	734
<b>17.7</b>	<b>Interrupt Thinning out Function . . . . .</b>	<b>735</b>
17.7.1	Interrupt thinning out function operation . . . . .	736
17.7.2	Operation example when peak and trough interrupts occur alternatively (in high-accuracy T-PWM mode)738	
17.7.3	Operation example when only peak interrupt occurs (in high-accuracy T-PWM mode)740	
17.7.4	Operation example when only trough interrupt occurs (in high-accuracy T-PWM mode)742	
17.7.5	Operation example when only peak interrupt occurs (in PWM mode) . . . . .	744
<b>17.8</b>	<b>A/D Conversion Trigger Function . . . . .</b>	<b>746</b>
17.8.1	A/D conversion trigger operation . . . . .	747
<b>17.9</b>	<b>Error/warning Interrupt . . . . .</b>	<b>752</b>
17.9.1	Error interrupt function . . . . .	752
17.9.2	Warning interrupt function . . . . .	755
<b>17.10</b>	<b>Operation in Each Mode . . . . .</b>	<b>756</b>
17.10.1	Interval timer mode . . . . .	757
17.10.2	External event count mode . . . . .	761
17.10.3	External trigger pulse output mode . . . . .	768
17.10.4	One-shot pulse mode . . . . .	772
17.10.5	PWM mode . . . . .	776
17.10.6	Free-running mode . . . . .	786

17.10.7	Triangular-wave PWM mode	790
17.10.8	High-accuracy T-PWM mode	795
17.10.9	PWM mode with dead time	840
17.10.10	120° excitation mode	852
17.10.11	Special 120° excitation mode	891
17.10.12	Special pattern output mode	921
17.10.13	Software output function	936
17.10.14	180° excitation control function	938

## **Chapter 18 16-bit Timer/Event Counter T (TMT)** 940

<b>18.1</b>	<b>Features</b>	940
<b>18.2</b>	<b>Function Outline</b>	940
<b>18.3</b>	<b>Configuration</b>	941
<b>18.4</b>	<b>Control Registers</b>	947
<b>18.5</b>	<b>Basic Operation</b>	962
18.5.1	Basic counter operation	962
18.5.2	Method for writing to compare register	966
<b>18.6</b>	<b>Operation in Each Mode</b>	970
18.6.1	Interval timer mode	970
18.6.2	External event count mode	973
18.6.3	External trigger pulse output mode	978
18.6.4	One-shot pulse mode	981
18.6.5	PWM mode	984
18.6.6	Free-running mode	988
18.6.7	Pulse width measurement mode	993
18.6.8	Triangular wave PWM mode	994
18.6.9	Encoder count function	995
18.6.10	Offset trigger generation mode	1018
<b>18.7</b>	<b>Cautions</b>	1019
18.7.1	TMT 'Encoder load enable'-mode limitation	1019

## **Chapter 19 Timer Interconnection** 1020

<b>19.1</b>	<b>Control register</b>	1023
<b>19.2</b>	<b>2-Phase Encoder Control Function</b>	1025
19.2.1	Function overview	1025
19.2.2	Configuration	1026
<b>19.3</b>	<b>3-Phase Pulse Input Control Function</b>	1028
19.3.1	Function overview	1028
19.3.2	Configuration	1029
<b>19.4</b>	<b>3-Phase Encoder Function</b>	1031
19.4.1	Function overview	1031
19.4.2	Configuration	1031
<b>19.5</b>	<b>TMT as Filter for TMAA</b>	1033

## **Chapter 20 On-Chip Debug Unit** 1034

<b>20.1</b>	<b>Functional Outline</b>	1034
20.1.1	Debug functions	1034
20.1.2	Security function	1036
<b>20.2</b>	<b>Connection to N-Wire Emulator</b>	1037

---

20.2.1	KEL connector	1037
20.3	Restrictions and Cautions on On-Chip Debug Function	1041
<b>Chapter 21 Auxiliary Frequency Output Function (AFO)</b>		1042
21.1	Features	1042
21.2	Configuration	1042
21.3	Control Registers	1043
21.4	Operation	1045
21.4.1	Auxiliary frequency generation	1045
21.4.2	Interval timer function	1045
<b>Chapter 22 Reset Function</b>		1046
22.1	Features	1046
22.2	Configuration	1046
22.3	Operation	1048
<b>Appendix A Special Function Registers</b>		1049
A.1	CAN Registers	1049
A.2	Other Special Function Registers	1052
<b>Revision History</b>		1067
<b>Index</b>		1069

# Chapter 1 Introduction

The V850E/PHC3 is a product in NEC Electronics' V850 family of single-chip microcontrollers designed for automotive applications.

## 1.1 General

With its high performance V850E CPU core and its compact feature set the V850E/PHC3 is especially suited for embedded control applications.

The V850E/PHC3 devices provide an excellent combination of general purpose peripheral functions like serial communication interfaces, timers/counters, measurement and control functions, with dedicated motor control timers and full CAN network support.

Thus equipped, the V850E/PHC3 product is ideally suited for automotive control applications, such as electric power steering (EPS). It is also an excellent choice for other embedded applications where a combination of sophisticated peripheral functions and CAN network support is required.

### (1) V850E CPU

The V850E CPU core is a 32-bit RISC processor. Through the use of basic instructions that can be executed in one clock period combined with an optimized pipeline architecture, it achieves marked improvements in instruction execution speed.

In addition, to make it ideal for use in digital control applications, a 32-bit hardware multiplier supports multiply instructions, saturated multiply instructions, bit operation instructions, etc.

Through two-byte basic instructions and instructions compatible with high level languages, the object code efficiency in a C compiler is increased, and program size can be reduced.

Further, because the on-chip interrupt controller provides high-speed interrupt response and processing, the device is well suited for high level real-time control applications.

### (2) On-chip flash memory

The V850E/PHC3 microcontroller has on-chip flash memory. It is possible to program the controller directly in the target environment where it is mounted.

With this feature, system development time can be reduced and system maintainability after shipping can be markedly improved.

**(3) Reliability**

Focus of the design was to achieve utmost reliability and availability. Flash memory and RAM are equipped with error correction code (ECC). ECC makes it possible to correct single bit errors automatically and to detect and flag double bit errors.

Writing to the flash is protected by various measures.

**(4) A full range of software development tools**

A development system is available that includes an optimized C compiler, debugger and other elements.

## 1.2 Features Summary

The CPU core provides:

- 96 instructions
- 32 general registers (32 bits each)
- Comprehensive instruction set:
  - V850E (compatible with V850, added powerful instructions for reducing code and increasing execution speed)
  - Signed multiplication (16 bits  $\times$  16 bits  $\rightarrow$  32 bits or 32 bits  $\times$  32 bits  $\rightarrow$  64 bits) in 1 to 2 clocks
  - Saturated operation instructions (with overflow/underflow detection)
  - 32-bit shift instructions in 1 clock cycle
  - Bit manipulation instructions
  - Load/store instructions with long/short format
  - Signed load instructions
- Linear address space 256 MB
- Floating Point Unit: IEEE754 Single-precision

The following table gives an overview of the most outstanding controller features.

**Table 1-1 V850E/PHC3 features**

Features		V850E/PHC3
CPU		V850E (32-bit RISC) with single-precision FPU
Internal memory	Code ROM	480KB
	Mask ROM	-
	RAM	32 KB
	Data Flash	32 KB
Operating clocks	CPU frequency	80 MHz
	MainOSC	operates on 16 MHz crystal
	PLL ratio	×2, ×5
	Internal oscillator	6.8 MHz typ.
Interrupts	Non-maskable	1 ch
	Maskable int.	98 ch
	Maskable ext.	13 ch
I/O lines	I/O ports	76
	Input ports	5
Timers	TAA	10 ch (16-bit general purpose timer/counter, cascadable for 32-bit operations)
	TMS	2 ch (16-bit timer/counter with Motor Control Functions)
	TMT	2 ch (16-bit general purpose timer/counter with PWM functions)
A/D converters	Analog inputs	20
	Resolution	10 bit
	Type	Successive approximation
Serial interfaces	UARTC	2 ch
	CSIB	2 ch Clocked Serial Interface
	CSIE	2 ch queued CSI
	CAN	2 ch CAN (32 message buffers for each channel)
DMA		8 ch
Other functions	Power save modes	HALT
	Clock Monitor	Main oscillator monitor
	Aux. frequency output	Programmable baud rate generator
	On-chip debug	Connection of an external N-Wire emulator
Power supply		<ul style="list-style-type: none"> <li>3.3 V ± 0.3 V</li> <li>1.5V ± 0.15 V</li> </ul> (refer to Data Sheet)
Package		144 pin LQFP

## 1.3 Description

The following figure provides a functional block diagram of the V850E/PHC3 microcontroller.

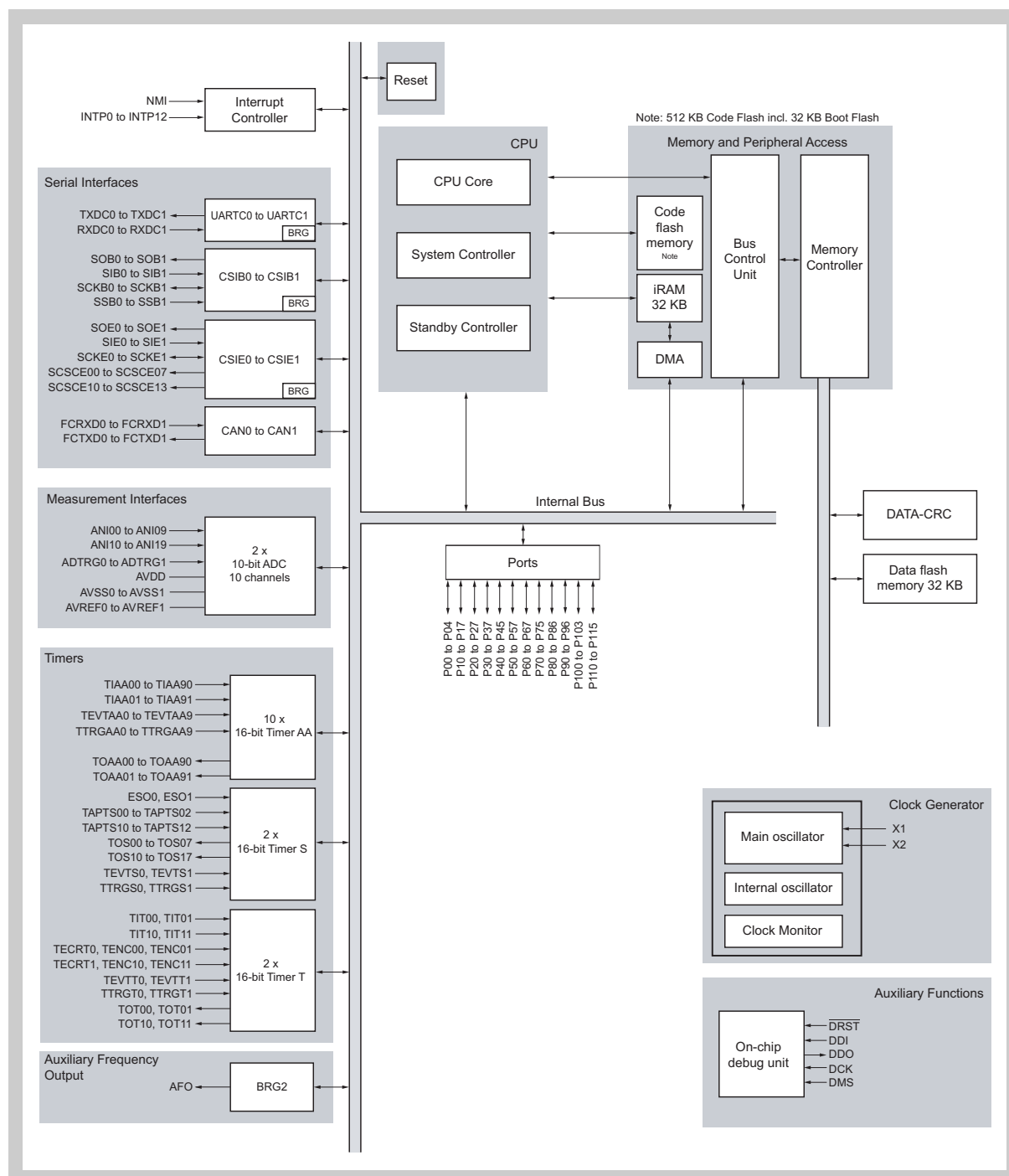


Figure 1-1 V850E/PHC3 block diagram



### 1.3.1 Internal units

<b>CPU</b>	The CPU can execute almost all instruction processing, such as address calculation, arithmetic and logic operations, and data transfer, in one clock cycle under control of a five-stage pipeline. Dedicated hardware units such as a multiplier and a 32-bit barrel shifter are provided to speed up program execution. The integrated Single-precision Floating Point Unit simplifies the execution of complex control algorithms.
<b>Code flash</b>	This is a built-in flash memory of 480 KB mapped to addresses 000 0000 <sub>H</sub> to 007 7FFF <sub>H</sub> . The CPU can access this memory in one clock when it fetches an instruction.
<b>Data flash</b>	A 32 KB data flash is provided, accessible via the Memory Controller.
<b>RAM</b>	This is a RAM of 32KB mapped to addresses 3FF 7000 <sub>H</sub> to 3FF EFFF <sub>H</sub> . The CPU can access this RAM in one clock when it accesses data.
<b>DMA Controller</b>	The V850E/PHC3 has an eighth-channel DMA Controller that transfers data between the internal RAM and serial interfaces, Timer S and the A/D converter.
<b>BCU / MEMC</b>	The Bus Control Unit (BCU) controls peripheral bus access. The Memory Controller (MEMC) controls the access to the on-chip Data-CRC unit and the data flash memory.
<b>Clock Generator</b>	The Clock Generator generates the system clocks. It includes the crystal-controlled 16 MHz main oscillator and an internal 6.8 MHz internal oscillator.
<b>Clock Monitor</b>	The Clock Monitor monitors the main oscillator. In case of failure, it can switch the CPU system and the Memory Controller to the internal oscillator.
<b>On-chip Debug Unit</b>	An on-chip debug function that uses an N-Wire interface is provided.
<b>INTC</b>	The Interrupt Controller (INTC) processes non-maskable and maskable interrupt requests from the on-chip peripheral hardware and external sources. Eight levels of priorities can be specified for these interrupt requests, and multiple servicing control can be performed on interrupt sources.
<b>DATA-CRC</b>	The DATA-CRC (DATA Cyclic Redundancy Check) can be used verify or generate CRC protected data streams of arbitrary length and different bit widths.
<b>UARTC</b>	The UARTs provide 2-wire asynchronous serial interfaces.
<b>CSIB</b>	The Clocked Serial Interfaces are 3-wire variable-length serial interfaces.
<b>CSIE</b>	The Enhanced Queued Clocked Serial Interfaces CSIE provide an internal FIFO buffer for queued operation.
<b>CAN</b>	The CAN Controller of this device fulfills the requirements according ISO 11898. Additionally, the CAN Controller was tested according to the test procedures required by ISO 16845. The CAN Controller has successfully passed all test patterns. Beyond these test patterns, other tests like robustness tests and processor interface tests as recommended by C&S/FH Wolfenbuettel have been performed with success.
<b>A/D Converter</b>	These are two high-speed, high-resolution 10-bit A/D Converters, each with 10 analog input pins. They are of successive approximation type.
<b>Timers/counters</b>	Ten 16-bit timers/event counters TAA, two 16-bit timers/event counters TMS with motor control capability, and two general purpose 16-bit timers TMT are provided.
<b>AFO</b>	The Auxiliary Frequency Output (AFO) provides a clock signal generated by a separate baud rate generator (BRG2).

### 1.3.2 Structure of the manual

This manual explains how to use the V850E/PHC3 microcontroller devices. It provides comprehensive information about the building blocks, their features, and how to set registers in order to enable or disable specific functions.

The manual provides individual chapters for the building blocks. These chapters are organized according to the grouping in the diagram.

- Core functions
  - “Pin Functions” on page 23*
  - “CPU System Functions” on page 71*
  - “Interrupt Controller (INTC)” on page 99*
  - “CRC Function (CRC)” on page 142*
  - “Clock Generator” on page 148*
- Memory access
  - “Flash Memory” on page 159*
  - “Bus and Memory Control (BCU, MEMC)” on page 192*
  - “DMA Functions (DMA Controller)” on page 211*
- Communication interfaces
  - “Introduction” on page 17*
  - “Clocked Serial Interface (CSIB)” on page 278*
  - “Enhanced Queued Clocked Serial Interface (CSIE)” on page 312*
  - “CAN Controller (CAN)” on page 368*
- Measurement interfaces
  - “A/D Converter (ADC)” on page 506*
- Timers
  - “16-Bit Timer/Event Counter AA (TAA)” on page 541*
  - “16-Bit Timer/Event Counter S (TMS)” on page 613*
  - “16-bit Timer/Event Counter T (TMT)” on page 940*
  - “Timer Interconnection” on page 1020*
- Auxiliary functions
  - “On-Chip Debug Unit” on page 1034*
  - “Auxiliary Frequency Output Function (AFO)” on page 1042*
- Power and reset
  - “Reset Function” on page 1046*

## Chapter 2 Pin Functions

This chapter lists the ports of the microcontroller. It presents the configuration of the ports for control functions. Noise elimination on input signals is explained and a recommendation for the connection of unused pins is given at the end of the chapter.

### 2.1 Overview

The microcontroller offers various pins for input/output functions, so-called ports. The ports are organized in port groups.

To allocate other than general purpose input/output functions to the pins, several control registers are provided.

For a description of the terms pin, port or port group, see *“Terms” on page 26*.

#### Features summary

- Number of ports and port groups:
  - Input/Output ports: 76
  - Input ports: 5
  - Port groups: 12
- 3.3 V I/O:  
Refer to the Data Sheet.
- Configuration possible for individual pins.
- Emergency shut-off configuration (only for dedicated ports 5 and 6)

### 2.1.1 Description

This microcontroller has the port groups shown below.

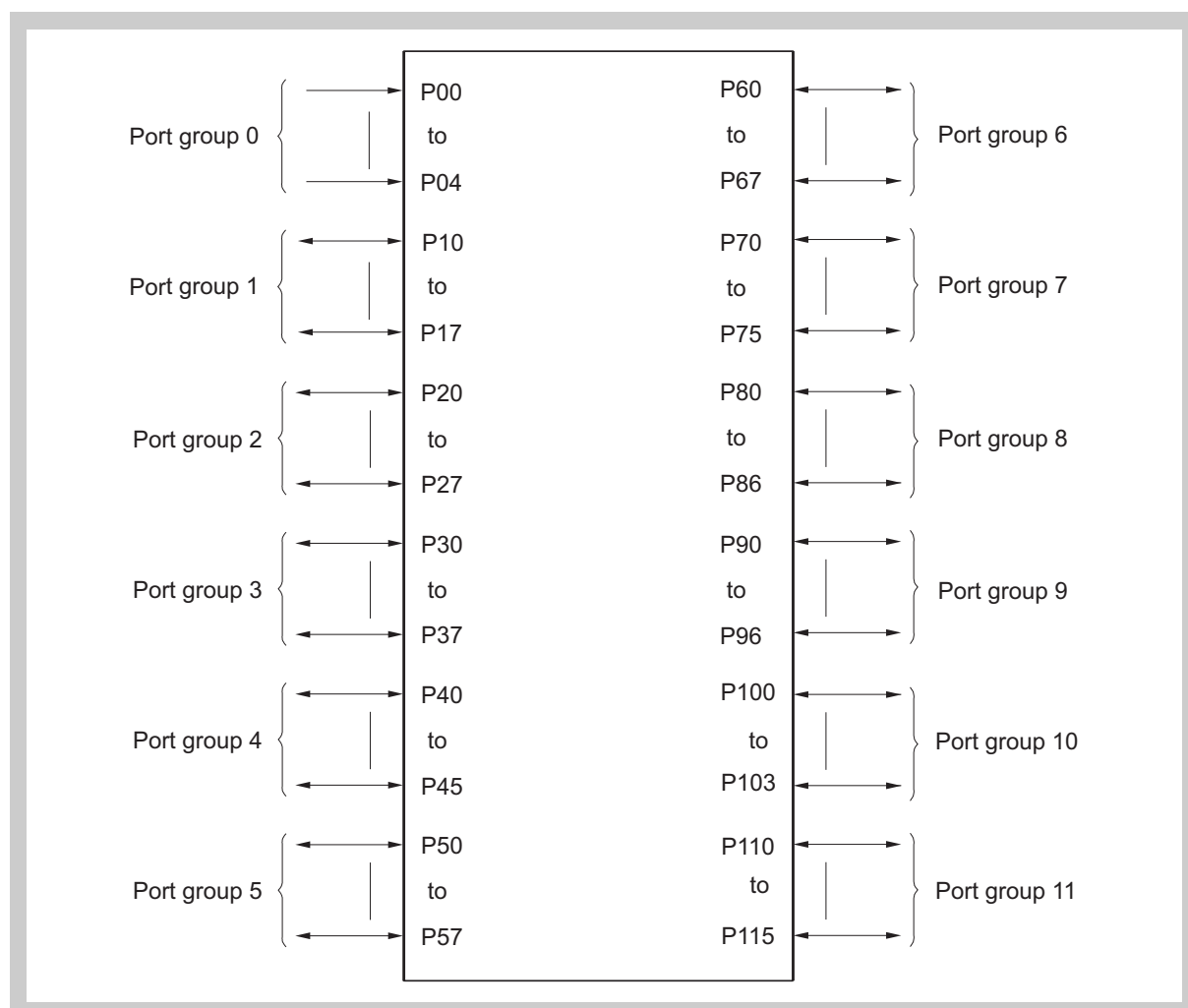


Figure 2-1 Port groups

**Port group overview** Table 2-1 gives an overview of the port groups. For each port group it shows the supported functions in port mode and in alternative mode. Any port group can operate in 8-bit or 1-bit units.

**Table 2-1 Functions of each port group**

Port group name	Function	
	Port mode	Alternative mode
0	5-bit input	<ul style="list-style-type: none"> <li>External interrupt 0 to 3</li> <li>Non maskable interrupt</li> <li>Emergency shut-off control of Timers TMS0/TMS1</li> <li>ADC trigger input</li> </ul>
1	8-bit input/output	<ul style="list-style-type: none"> <li>Timer TAA0 to TAA3 channels</li> </ul>
2	8-bit input/output	<ul style="list-style-type: none"> <li>Timer TAA4 to TAA7 channels</li> </ul>
3	8-bit input/output	<ul style="list-style-type: none"> <li>External interrupt 4 and 5</li> <li>UARTC0 receive/transmit data,</li> <li>UARTC1 receive/transmit data</li> <li>CAN0 receive/transmit data,</li> <li>CAN1 receive/transmit data</li> </ul>
4	6-bit input/output	<ul style="list-style-type: none"> <li>CSIB0 data/clock line</li> <li>CSIB1 data/clock line</li> </ul>
5	8-bit input/output	<ul style="list-style-type: none"> <li>Timer TMS0 pulse signal output</li> </ul>
6	8-bit input/output	<ul style="list-style-type: none"> <li>Timer TMS1 pulse signal output</li> </ul>
7	6-bit input/output	<ul style="list-style-type: none"> <li>Timer TMT0 channels</li> <li>Timer TMT1 channels</li> <li>External interrupt 12</li> <li>Auxiliary frequency output</li> </ul>
8	7-bit input/output	<ul style="list-style-type: none"> <li>CSIE0 data/clock line</li> <li>CSIE0 chip select areas 0 to 3</li> <li>CSIB0 serial slave select input</li> <li>External interrupts 6 to 8</li> </ul>
9	7-bit input/output	<ul style="list-style-type: none"> <li>CSIE0 chip select areas 4 to 7</li> <li>CSIE1 data/clock line</li> <li>CSIE1 chip select areas</li> <li>CSIB1 serial slave select input</li> <li>External interrupts 9 to 11</li> </ul>
10	4-bit input/output	<ul style="list-style-type: none"> <li>Timer TAA8 channels</li> <li>Timer TAA9 channels</li> </ul>
11	6-bit input/output	<ul style="list-style-type: none"> <li>Timer TMS0 input channels</li> <li>Timer TMS1 input channels</li> </ul>

**Pin configuration** To define the function of a pin, several control registers are provided.

- For a general description of the registers, see “Port Group Configuration Registers” on page 27.
- For every port, detailed information on the configuration registers is given in “Port Group Configuration” on page 40.

There are several types of control circuits, defined as port types. For a description of the port types, see “Port Types Diagrams” on page 44.

### 2.1.2 Terms

In this section, the following terms are used:

- **Pin**

Denotes the physical pin. Every pin is uniquely denoted by its pin number.

A pin can be used in several modes. Depending on the selected mode, a pin name is allocated to the pin.

- **Port group**

Denotes a group of pins. The pins of a port group have a common set of port mode control registers.

- **Port mode / Port**

A pin in port mode works as a general purpose input/output pin. It is then called “port”.

The corresponding name is Pnm. For example, P07 denotes port 7 of port group 0. It is referenced as “port P07”.

- **Alternative mode**

In alternative mode, a pin can work in various non-general purpose input/output functions, for example, as the input/output pin of on-chip peripherals.

The corresponding pin name depends on the selected function. For example, pin INTP0 denotes the pin for one of the external interrupt inputs.

Note that for example P01 and INTP0 denote the same physical pin. The different names indicate the function in which the pin is being operated.

- **Port type**

A control circuit evaluates the settings of the configuration registers. There are different types of control circuits, called “port types”.

### 2.1.3 Noise elimination

The input signals at some pins are passing a filter to remove noise and glitches. The microcontroller supports both analog and digital filters.

See “*Noise Elimination*” on page 63 for a detailed description.

## 2.2 Port Group Configuration Registers

This section starts with an overview of all configuration registers and then presents all registers in detail. The configuration registers are classified in the following groups:

- “Pin function configuration” on page 28
- “Pin data input/output” on page 32
- “Emergency shut-off configuration” on page 34
- “Write enable register” on page 38

### 2.2.1 Overview

For the configuration of the individual pins of the port groups, the following registers are used:

**Table 2-2 Registers for port group configuration**

Register name	Shortcut	Function
Port mode control register	PMCn	Pin function configuration
Port mode register	PMn	
Port function control register	PFCn	
Port function control expansion register	PFCEn	
Port register	Pn	Pin data input/output
Emergency shut-off control register	PESCn	Emergency shut-off configuration
Emergency shut-off status register	ESOSTn	
Emergency shut-off mask register	PESMKn	
Write enable register	PRCMD	Port configuration registers write enable

n = 0 to 11

## 2.2.2 Pin function configuration

The registers for pin function configuration define the general function of a pin:

- port mode or alternative mode
- in port mode: input mode or output mode
- in alternative mode: selection of one of the alternative functions in alternative mode

An overview of the register settings is given in the table below.

**Table 2-3 Pin function configuration (overview)**

Function	Registers				I/O
	PMC	PM	PFCE	PFC	
Port mode (output)	0	0	X	X	O
Port mode (input)		1	X	X	I
Alternative mode (alternative function 1)	1	X	0	0	I/O <sup>a</sup>
Alternative mode (alternative function 2)				1	
Alternative mode (alternative function 3)			1	0	
Alternative mode (alternative function 4)				1	

<sup>a)</sup> In alternative mode, the corresponding port type defines whether a pin is in input mode or output mode.



**(1) PMcN - Port mode control register**

The PMcN register specifies whether the individual pins of port group n are in port mode or in alternative mode.

This is an 8-bit register.

**Access** This register can be read/written in 8-bit and 1-bit units.

**Address** see “Port Group Configuration” on page 40

**Initial Value** 00<sub>H</sub> or 0000<sub>H</sub>. This register is initialized by  $\overline{\text{RESET}}$ .

7	6	5	4	3	2	1	0
PMcN7	PMcN6	PMcN5	PMcN4	PMcN3	PMcN2	PMcN1	PMcN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 2-4 PMcN register contents**

Bit position	Bit name	Function
7 to 0	PMcN[7:0]	Specifies the operation mode of the corresponding pin 0: Port mode 1: Alternative mode

**Caution** When changing the function of a port from port mode (PCMnm = 0) to external interrupt input (PCMnm = 1) an advertent interrupt may occur.

Therefore, it is recommended to follow the below procedure:

1. To select the alternative input function INTPn (I), set PFCE.PFCEnm and PFC.PFCnm accordingly.
2. Set PMcNm = 1 to change to the alternative mode.
3. Wait until the delay of the noise rejection filter has passed.
4. Set INTnIC.INTnIF = 0 to clear the interrupt request.
5. Clear INTnIC.INTnMK (or clear INTMR.INTnMK) to enable the interrupt.

In step 3 you must wait for a certain time span because the external interrupt pins are equipped with noise rejection filters. The filters cause a delay in which the interrupt request flag INTnIC.INTnIF is set. This flag must be cleared (step 4).

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

If a pin is in port mode ( $PMCn.PMCnm = 0$ ), the PMn register specifies whether the individual pins of the port group n are in input mode or in output mode.

This is an 8-bit register.

**Note** If a pin is in alternative mode ( $PMCn.PMCnm = 1$ ) and the corresponding PMn bit is set ( $PMn.PMnm = 1$ ), then the pin behaves as in input port mode: Reading Pn.Pmn reads the pin status.

**Access** This register can be read/written in 8-bit and 1-bit units.

**Address** see “Port Group Configuration” on page 40

**Initial Value** FF<sub>H</sub> or FFFF<sub>H</sub>. This register is initialized by  $\overline{RESET}$ .

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

**Table 2-5 PMn register contents**

Bit position	Bit name	Function
7 to 0	PMn[7:0]	Specifies input/output mode of the corresponding pin in port mode 0: Output mode (output enabled) 1: Input mode (output disabled)

**(3) PFCn - Port function control register**

If a pin is in alternative mode ( $PMCn.PMCnm = 1$ ) some pins offer up to four alternative functions.

The 8-bit PFCn register together with the 8-bit PFCEn register specifies which function of a pin is to be used. The corresponding port type defines whether a pin is in input or output mode.

**Access** This register can be read/written in 8-bit and 1-bit units.

**Address** see “Port Group Configuration” on page 40

**Initial Value** 00<sub>H</sub>

This register is initialized by  $\overline{RESET}$ .

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

**Table 2-6 PFCn register contents**

Bit position	Bit name	Function
7 to 0	PFCn[7:0]	See Table 2-3 on page 28 for details

**(4) PFCEn - Port function control expansion register**

If a pin is in alternative mode ( $PMCn.PMCnm = 1$ ) some pins offer up to four alternative functions.

The 8-bit PFCEn together with the PFCn register specifies which function of a pin is to be used. The corresponding port type defines whether a pin is in input or output mode.

**Access** This register can be read/written in 8-bit and 1-bit units.

**Address** see “Port Group Configuration” on page 40

**Initial Value** 00<sub>H</sub>

This register is initialized by  $\overline{RESET}$ .

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

**Table 2-7 PFCEn register contents**

Bit position	Bit name	Function
7 to 0	PFCEn[7:0]	See Table 2-3 on page 28 for details

## 2.2.3 Pin data input/output

If a pin is in port mode, the registers for pin data input/output specify the input and output data.

### (1) Pn - Port register

If a pin is in port mode ( $PMCn.PMCnm = 0$ ), data is input from or output to an external device by writing or reading the Pn register.

This is an 8-bit register.

**Access** This register can be read/written in 8-bit and 1-bit units.

**Address** see “Port Group Configuration” on page 40

**Initial Value** Undefined.

**Note** After reset, the ports are in input mode ( $PMn.PMnm = 1$ ). The read input value is determined by the port pins.

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

**Table 2-8 Pn register contents**

Bit position	Bit name	Function
7 to 0	Pn[7:0]	Data, see Table 2-9 on page 32 and Table 2-10 on page 33 for details.

**Note** The value written to register Pn is retained until a new value is written to register Pn.

**Port mode** In port mode ( $PMCn.PMCnm = 0$ ), register PMn specifies whether a pin is in input or in output mode. Data is written to or read from the Pn register as follows:

**Table 2-9 Writing/reading register Pn in port mode ( $PMCn.PMCnm = 0$ )**

Function	PM	I/O
Write to Pn...		
...and output contents of Pn to pins	0	O
...without affecting the pin status	1	I
Read from Pn...		
...and thus read the pin status	1	I
...read the data in the register Pn, and disregard the actual pin status	0	O

**Alternative mode** In alternative mode ( $PMCn.PMCnm = 1$ ), the corresponding port type defines whether a pin is in input or output mode. However, register  $PMn$  influences the writing/reading of register  $Pn$ .

In alternative mode, data is written to or read from the  $Pn$  register as follows:

**Table 2-10 Writing/reading register  $Pn$  in alternative mode ( $PMCn.PMCnm = 1$ )**

Function	PM	I/O
Write to $Pn$ without affecting the pin status	X	—
Read from $Pn$ ...		
...and read the value of the alternative output function (for pins in alternative output function)	0	—
...read the data in the register $Pn$ , and disregard the actual pin status (for pins in alternative input function)		
...and thus read the pin status	1	I

**Caution** Although 1-bit operations (read-modify-write operations) on  $Pn$  registers are intended to modify only a single bit, the entire  $Pn$  register is read. After the single bit has been modified, the contents of the complete register is written back.

If the ports of the register  $Pn$  contain both input and output ports  $Pnm$ , the read of  $Pn$  returns

- the contents of the register  $Pn$  for output ports
- the pin status of input ports, but not the  $Pn$  register bits

That means the read value of  $Pn$  may be different to the contents of the  $Pn$  register at bit positions, which are assigned to input ports.

Thus the contents of  $Pn$  may differ to the previous value not just in the bit that was to be modified, but also in other bits.

Example:

- Register  $P1$  has the contents  $00_H$ .
- Port  $P10$  is configured as an output port, all other ports of port group 1 (ports  $P11$  to  $P17$ ) are configured as input ports.
- The port pins of ports  $P11$  to  $P17$  all have the level “1”.
- Bit  $P1.P10$  is set to 1 by a 1-bit operation.

Afterwards, register  $P1$  holds the value  $FF_H$  instead of the expected value  $01_H$ , since bits  $P11$  to  $P17$  have been overwritten with the corresponding pin levels “1”.

## 2.2.4 Emergency shut-off configuration

The emergency shut-off function sets a port immediately into high impedance state upon occurrence of two events:

- assertion of an external signal ESOm
- interrupt of the Clock Monitor INTOSD

Both events can be separately permitted respectively prohibited to trigger an emergency shut-off.

The registers for controlling the emergency shut-off are described in the following.

### (1) PESCN - Emergency shut-off control register

The 8-bit PESCN register specifies the condition of the concerned external control pin to trigger an emergency shut-off. An emergency shut-off can be generated upon edges or levels.

Writing to this register is protected by a special sequence of instructions. To enable write access to PESCN, first write to PRCMD. Please refer to “Write Protected Registers” on page 97 for details.

**Access** This register can be read/written in 8-bit units.

**Address** see “Port Group Configuration” on page 40

**Initial Value** 00<sub>H</sub>. This register is initialized by  $\overline{\text{RESET}}$ .

7	6	5	4	3	2	1	0
0	0	0	0	ESOmEN	0	ESOmED1	ESOmED0
R	R	R	R	R/W	R	R/W	R/W

**Table 2-11 PESCN register contents**

Bit Position	Bit Name	Function															
3	ESOmEN	Enables/disables the emergency shut-off control by ESOm input 0: disabled 1: enabled															
1 to 0	ESOmED [1:0]	Sets the edge or level of input signal ESOm for triggering an emergency shut-off. <table border="1"> <thead> <tr> <th>ESOmED1</th><th>ESOmED0</th><th>Selected edge/level</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>falling edge</td></tr> <tr> <td>0</td><td>1</td><td>rising edge</td></tr> <tr> <td>1</td><td>0</td><td>low level</td></tr> <tr> <td>1</td><td>1</td><td>high level</td></tr> </tbody> </table>	ESOmED1	ESOmED0	Selected edge/level	0	0	falling edge	0	1	rising edge	1	0	low level	1	1	high level
ESOmED1	ESOmED0	Selected edge/level															
0	0	falling edge															
0	1	rising edge															
1	0	low level															
1	1	high level															

**Caution** State of the edge detection control bits ESOmED1 and ESOmED0 must not be changed while ESOmEN is set to 1. Otherwise the output shut-off function may be unintentionally triggered or a trigger event may be lost.

- Note**
1. The emergency shut-off function is not dedicated to a single port of a port group, but acts on the *entire* port group. However, note that not all ports of a port group may be subject to the emergency shut-off function. Refer to the description of the concerned port group.
  2. If the emergency shut-off condition is fulfilled, the concerned pin is not switched to high impedance state if the port is configured as port output (PM<sub>Cnm</sub> = 0, PM<sub>nm</sub> = 0).
  3. The output buffer of the concerned ports P<sub>nm</sub> are forcibly disabled (high impedance output) as long as PES<sub>Cn</sub>.ES<sub>Om</sub>EN and ES<sub>OSTn</sub>.ES<sub>Om</sub>ST are set to 1.

**Setting up an emergency shut-off**

The setup of the emergency shut-off function must be performed in the following sequence. Otherwise the output shut-off function may be unintentionally triggered or a trigger event may be lost.

1. Power on (all registers are reset)
2. PRCMD write (write protect released)
3. Clear ES<sub>Om</sub>EN bit to 0
4. PRCMD write (write protect released)
5. Set ES<sub>Om</sub>ED0, ES<sub>Om</sub>ED1 bits
6. PRCMD write (write protect released)
7. Clear ES<sub>Om</sub>ST bit of ES<sub>OSTn</sub> register to 0
8. PRCMD write (write protect released)
9. Set PES<sub>MKn</sub>m bits to 0 respectively 1
10. PRCMD write (write protect released)
11. Set ES<sub>Om</sub>EN bit to 1

**(2) ESOSTn - Emergency shut-off status register**

The 8-bit ESOSTn register indicates the status of an emergency shut-off condition.

Writing to this register is protected by a special sequence of instructions. To enable write access to ESOSTn, first write to PRCMD. Please refer to “*Write Protected Registers*” on page 97 for details.

**Access** This register can be read/written in 8-bit units.

**Address** see “*Port Group Configuration*” on page 40

**Initial Value** 00<sub>H</sub>. This register is initialized by  $\overline{\text{RESET}}$ .

7	6	5	4	3	2	1	0
ESOmST	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

**Table 2-12 ESOSTn register contents**

Bit Position	Bit Name	Function
7	ESOmST	Specifies the level at the corresponding pin after emergency shut-off 0: high (no emergency shut-off was triggered: pin output is active) 1: low (emergency shut-off was triggered: pin is in high impedance state)

- Note**
1. The emergency shut-off function is not dedicated to a single port of a port group, but acts on the *entire* port group. However, note that not all ports of a port group may be subject to the emergency shut-off function. Refer to the description of the concerned port group.
  2. Writing the emergency shut-off status flag (ESOmST) is only possible if PESCN.ESmEN is cleared (0).
  3. The ESOmST flag can only be cleared by CPU to 0. Setting the ESOmST flag to 1 is not possible.
  4. The output buffers of concerned ports Pnm are forcibly disabled (high impedance output) as long as PESCN.ESOmEN and PESCN.ESOmST are set to 1.



**(3) PESMKn - Emergency shut-off mask register**

The 8-bit PESMKn register allows to mask and unmask events to perform an emergency shut-off.

Writing to this register is protected by a special sequence of instructions. To enable write access to PESMKn, first write to PRCMD. Please refer to “*Write Protected Registers*” on page 97 for details.

**Access** This register can be read/written in 8-bit and 1-bit units.

**Address** see “*Port Group Configuration*” on page 40

**Initial Value** 00<sub>H</sub>. This register is initialized by  $\overline{\text{RESET}}$ .

7	6	5	4	3	2	1	0
0	0	0	0	0	0	PESMKn1	PESMKn0
R	R	R	R	R	R	R	R/W

**Table 2-13 PESMKn register contents**

Bit Position	Bit Name	Function
1	PESMKn1	Masks function of INTOSD 0: unmasked (INTOSD can trigger an emergency shut-off) 1: masked (INTOSD can not trigger an emergency shut-off)
0	PESMKn0	Masks function of ESOM input signal 0: unmasked (ESOM can trigger an emergency shut-off) 1: masked (ESOM can not trigger an emergency shut-off)

## 2.2.5 Write enable register

Some port group configuration registers are write protected. Write access to a write protected register is only given immediately after writing to a corresponding write enable register.

### (1) PRCMD - Command register

The 8-bit PRCMD register protects port group configuration registers from inadvertent write access, so that the system does not stop in case of a program hang-up.

After writing to the PRCMD register, you are permitted to write once to one of the protected registers. This must be done immediately after writing to the PRCMD register. After the second write action, or if the second write action does not follow immediately, all protected registers are write-locked again.

The following registers are protected by PRCMD (n = 5, 6):

PMCn	Port mode control register
PMn	Port mode register
Pn	Port register
PESCn	Emergency shut-off control register
PESMKn	Emergency shut-off mask register
ESOSTn	Emergency shut-off status register

An invalid write attempt to one of the above registers sets the error flag PHS.PRERR. PHS.PRERR is also set, if a write access to PRCMD is not immediately followed by an access to one of the protected registers.

**Access** This register can only be written in 8-bit units.

**Address** FFFF F1FC<sub>H</sub>.

**Initial Value** The contents of this register is undefined.

7	6	5	4	3	2	1	0
X	X	X	X	X	X	X	X
W	W	W	W	W	W	W	W

**(2) PHS - Peripheral status register**

The 8-bit PHS register indicates the status of a write attempt to a register protected by PRCMD (see also “*PRCMD - Command register*” on page 38).

**Access** This register can be read/written in 8-bit units.

**Address** FFFF F802<sub>H</sub>.

**Initial Value** 00<sub>H</sub>. The register is cleared by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	PRERR
R <sup>a</sup>	R <sup>a</sup>	R <sup>a</sup>	R <sup>a</sup>	R <sup>a</sup>	R <sup>a</sup>	R <sup>a</sup>	R/W

a) These bits may be written, but write is ignored.

**Table 2-14 PHS register contents**

Bit position	Bit name	Function
0	PRERR	Write error status: 0: Write access was successful. 1: Write access failed. You can clear this bit by writing 0 to it. Setting this bit to 1 by software is not possible.

**Note** PHS.PRERR is set, if a write access to register PRCMD is not directly followed by a write access to one of the write-protected registers.

## 2.3 Port Group Configuration

This section provides an overview of the port groups (Table 2-15) and of the pin functions (Table 2-17 on page 45). In Table 2-47 on page 67 it is listed how the pin functions change if the microcontroller is reset or if it is in one of the standby modes.

In the subsections, for every port group the settings of the configuration registers is listed. Further, the addresses and initial values of the configuration registers are given. See “Port group 0” on page 51 to “Port group 11” on page 62.

For the port type diagrams refer to “Port Types Diagrams” on page 44.

### 2.3.1 Port group configuration lists

Table 2-15 provides an overview of the functions available at each port pin.

**Table 2-15 Port group list (1/3)**

Port group name	Port name	Alternative outputs	Alternative inputs	Port type	Pin number
0	P00	—	NMI	2	5
	P01	—	ESO0/INTP0	2	6
	P02	—	ESO1/INTP1	2	7
	P03	—	ADTRG0/INTP2	2	8
	P04	—	ADTRG1/INTP3	2	9
1	P10	TOAA00	TIAA00/TEVTAA1	5-K	51
	P11	TOAA01	TIAA01/TTRGAA1	5-K	52
	P12	TOAA10	TIAA10/TTRGAA0	5-K	53
	P13	TOAA11	TIAA11/TEVTAA0	5-K	54
	P14	TOAA20	TIAA20/TEVTAA3	5-K	55
	P15	TOAA21	TIAA21/TTRGAA3	5-K	60
	P16	TOAA30	TIAA30/TTRGAA2	5-K	61
	P17	TOAA31	TIAA31/TEVTAA2	5-K	62
2	P20	TOAA40	TIAA40/TEVTAA5	5-K	63
	P21	TOAA41	TIAA41/TTRGAA5	5-K	64
	P22	TOAA50	TIAA50/TTRGAA4	5-K	65
	P23	TOAA51	TIAA51/TEVTAA4	5-K	66
	P24	TOAA60	TIAA60/TEVTAA7	5-K	67
	P25	TOAA61	TIAA61/TTRGAA7	5-K	68
	P26	TOAA70	TIAA70/TTRGAA6	5-K	69
	P27	TOAA71	TIAA71/TEVTAA6	5-K	70

Table 2-15 Port group list (2/3)

Port group name	Port name	Alternative outputs	Alternative inputs	Port type	Pin number
3	P30	–	RXDC0/INTP4	5-K	31
	P31	TXDC0	–	5-K	32
	P32	–	RXDC1/INTP5	5-K	33
	P33	TXDC1	–	5-K	39
	P34	–	FCRXD0	5-K	40
	P35	FCTXD0	–	5-K	41
	P36	–	FCRXD1	5-K	42
	P37	FCTXD1	–	5-K	43
4	P40	–	SIB0	5-K	71
	P41	SOB0	–	5-K	72
	P42	SCKB0	SCKB0	5-K	73
	P43	–	SIB1	5-K	21
	P44	SOB1	–	5-K	22
	P45	SCKB1	SCKB1	5-K	23
5	P50	TOS00	–	5-K	94
	P51	TOS01	–	5-K	95
	P52	TOS02	–	5-K	96
	P53	TOS03	–	5-K	97
	P54	TOS04	–	5-K	98
	P55	TOS05	–	5-K	99
	P56	TOS06	–	5-K	100
	P57	TOS07	–	5-K	101
6	P60	TOS10	–	5-K	102
	P61	TOS11	–	5-K	103
	P62	TOS12	–	5-K	104
	P63	TOS13	–	5-K	105
	P64	TOS14	–	5-K	106
	P65	TOS15	–	5-K	107
	P66	TOS16	–	5-K	108
	P67	TOS17	–	5-K	111
7	P70	TOT00	TIT00/TEVTT1 TENCT00	5-K	112
	P71	TOT01	TIT01/TTRGT1 TENCT01	5-K	113
	P72	–	TECRT0/INTP12	5-K	114
	P73	TOT10	TIT10/TTRGT0 TENCT11	5-K	144
	P74	TOT11	TIT11/TEVTT0 TENCT10	5-K	3
	P75	AFO	TECRT1	5-K	4

**Table 2-15 Port group list (3/3)**

Port group name	Port name	Alternative outputs	Alternative inputs	Port type	Pin number
8	P80	–	SIE0	5-K	24
	P81	SOE0	–	5-K	25
	P82	SCKE0	SCKE0	5-K	26
	P83	SCSE00	INTP6	5-K	27
	P84	SCSE01	INTP7	5-K	28
	P85	SCSE02	INTP8	5-K	29
	P86	SCSE03	SSB0	5-K	30
9	P90	–	SIE1	5-K	44
	P91	SOE1	–	5-K	45
	P92	SCKE1	SCKE1	5-K	46
	P93	SCSE10/SCSE04	INTP9	5-K	47
	P94	SCSE11/SCSE05	INTP10	5-K	48
	P95	SCSE12/SCSE06	INTP11	5-K	49
	P96	SCSE13/SCSE07	SSB1	5-K	50
10	P100	TOAA80	TIAA80/TTRGAA8	5-K	88
	P101	TOAA81	TIAA81/TEVTAA8	5-K	91
	P102	TOAA90	TIAA90/TTRGAA9	5-K	92
	P103	TOAA91	TIAA91/TEVTAA9	5-K	93
11	P110	–	TAPTS00/TEVTS0	5-K	87
	P111	–	TAPTS01/TTRGS0	5-K	86
	P112	–	TAPTS02	5-K	85
	P113	–	TAPTS10/TEVTS1	5-K	76
	P114	–	TAPTS11/TTRGS1	5-K	75
	P115	–	TAPTS12	5-K	74

## 2.3.2 Non-port pins

Voltage and clock supply pins do not have general purpose I/O port functionality. Furthermore *Table 2-16* lists all other pins which do not have a port function.

**Table 2-16 Non-port pins**

Name	Port type	Pin location
ANI00	7	116
ANI01	7	117
ANI02	7	118
ANI03	7	119
ANI04	7	120
ANI05	7	121
ANI06	7	122
ANI07	7	123

**Table 2-16 Non-port pins**

Name	Port type	Pin location
ANI08	7	124
ANI09	7	125
ANI10	7	138
ANI11	7	137
ANI12	7	136
ANI13	7	135
ANI14	7	134
ANI15	7	133
ANI16	7	132
ANI17	7	131
ANI18	7	130
ANI19	7	129
AVREF0	-	126
AVREF1	-	128
DCK	2	16
DDI	2	18
DDO	5	15
DMS	2	17
$\overline{\text{DRST}}$	2-I	14
MODE0/FLMD1	2	81
MODE1/FLMD0	2	82
MODE2	2	80
MODE3	2	77
$\overline{\text{RESET}}$	2	34

### 2.3.3 Port Types Diagrams

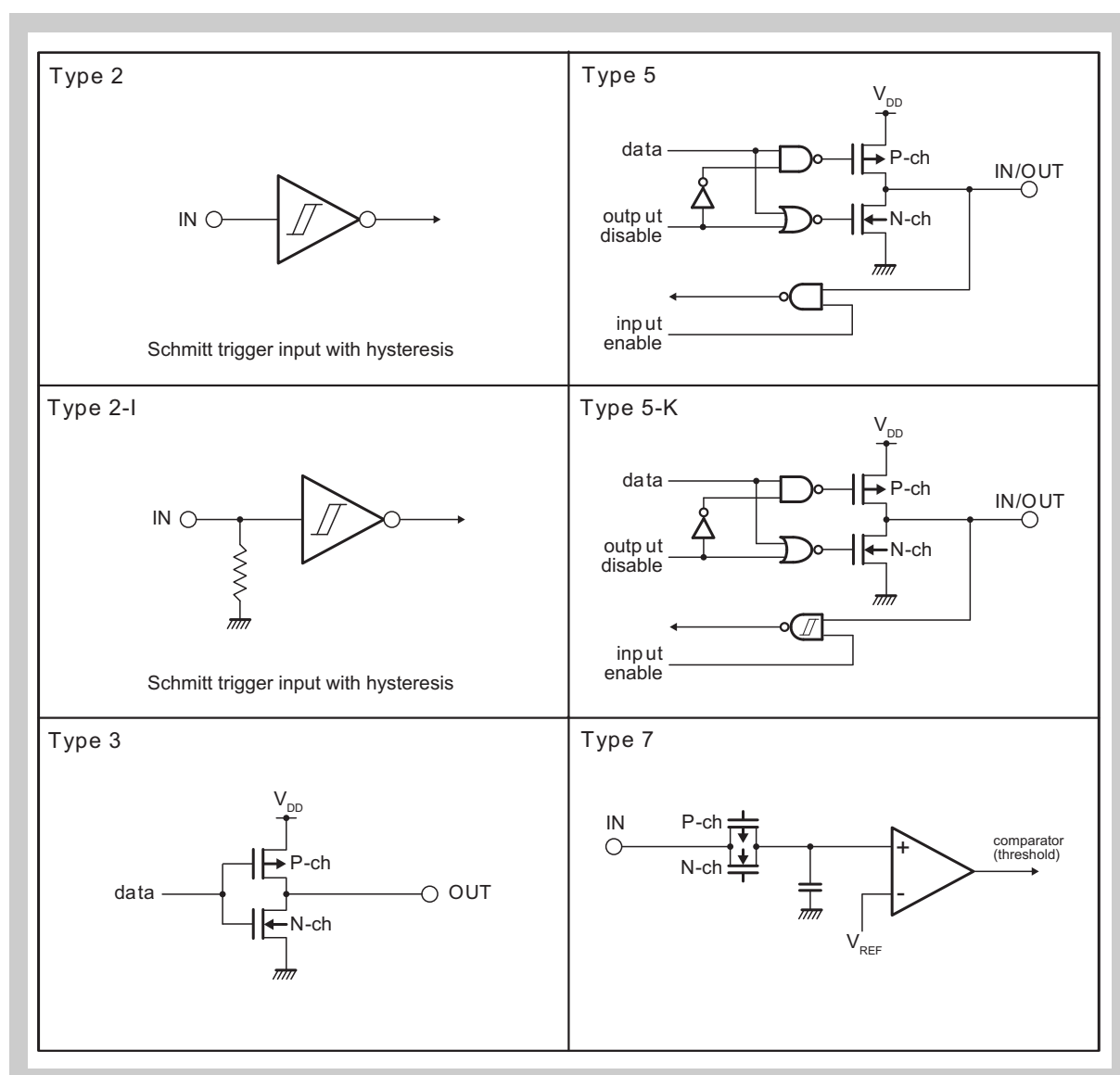


Figure 2-2 Port type diagrams



## 2.3.4 Alphabetic pin function list

Table 2-17 provides a list of all pin function names in alphabetic order.

**Table 2-17 Alphabetic pin functions list (1/6)**

Pin name	I/O	Pin function	Port	Pin location
ADTRG0	I	ADC0 trigger input	P03	8
ADTRG1	I	ADC1 trigger input	P04	9
AFO	O	Auxiliary frequency output	P75	4
ANI00	I	ADC inputs	—	116
ANI01			—	117
ANI02			—	118
ANI03			—	119
ANI04			—	120
ANI05			—	121
ANI06			—	122
ANI07			—	123
ANI08			—	124
ANI09			—	125
ANI10			—	138
ANI11			—	137
ANI12			—	136
ANI13			—	135
ANI14			—	134
ANI15			—	133
ANI16			—	132
ANI17			—	131
ANI18			—	130
ANI19			—	129
AVDD	—	ADC supply voltage (3.3 V)	—	127
AVREF0	—	ADC0 reference voltage input	—	126
AVREF1	—	ADC1 reference voltage input	—	128
AVSS0	—	ADC0 ground	—	115
AVSS1	—	ADC1 ground	—	139
CVDD15	—	Oscillator supply voltage (1.5 V)	—	11
CVSS15		Oscillator supply ground	—	10
DCK	I	N-Wire interface clock	—	16
DDI	I	N-Wire interface debug data input	—	18
DDO	O	N-Wire interface debug data output	—	15
DMS	I	N-Wire interface debug mode select input	—	17
$\overline{\text{DRST}}$	I	N-Wire debug interface reset	—	14
ESO0	I	Emergency shut-off control of Timer TMS0	P01	6
ESO1	I	Emergency shut-off control of Timer TMS1	P02	7
FCRXD0	I	CAN0 receive data	P34	40

Table 2-17 Alphabetic pin functions list (2/6)

Pin name	I/O	Pin function	Port	Pin location
FCRXD1	I	CAN1 receive data	P36	42
FCTXD0	O	CAN0 transmit data	P35	41
FCTXD1	O	CAN1 transmit data	P37	43
FLMD0 (= MODE1)	I	Flash writing control	–	82
FLMD1 (= MODE0)	I	Flash writing control	–	81
INTP0	I	External interrupts	P01	6
INTP1			P02	7
INTP2			P03	8
INTP3			P04	9
INTP4			P30	31
INTP5			P32	33
INTP6			P83	27
INTP7			P84	28
INTP8			P85	29
INTP9			P93	47
INTP10			P94	48
INTP11			P95	49
INTP12			P72	114
MODE0	I	Operating mode select pins	–	81
MODE1			–	82
MODE2			–	80
MODE3			–	77
NMI	I	Non-maskable interrupt	P00	5
RESET	I	Reset input	–	34
RXDC0	I	UARTC0 receive data	P30	31
RXDC1	I	UARTC1 receive data	P32	33
SCKB0	I/O	Clocked Serial Interface CSIB0 clock line	P42	73
SCKB1	I/O	Clocked Serial Interface CSIB1 clock line	P45	23
SCKE0	I/O	CSIE0 clock line	P82	26
SCKE1	I/O	CSIE1 clock line	P92	46
SCSE00	O	CSIE0 chip selects	P83	27
SCSE01			P84	28
SCSE02			P85	29
SCSE03			P86	30
SCSE04			P93	47
SCSE05			P94	48
SCSE06			P95	49
SCSE07			P96	50

Table 2-17 Alphabetic pin functions list (3/6)

Pin name	I/O	Pin function	Port	Pin location
SCSE10	O	CSIE1 chip selects	P93	47
SCSE11			P94	48
SCSE12			P95	49
SCSE13			P96	50
SIB0	I	CSIB0 data input	P40	71
SIB1	I	CSIB1 data input	P43	21
SIE0	I	CSIE0 data input	P80	24
SIE1	I	CSIE1 data input	P90	44
SOB0	O	CSIB0 data output	P41	72
SOB1	O	CSIB1 data output	P44	22
SOE0	O	CSIE0 data output	P81	25
SOE1	O	CSIE1 data output	P91	45
SSB0	I	CSIB0 serial slave select input	P86	30
SSB1	I	CSIB1 serial slave select input	P96	50
TAPTS00	I	Timer TMS0 trigger signals	P110	87
TAPTS01			P111	86
TAPTS02			P112	85
TAPTS10	I	Timer TMS1 trigger signals	P113	76
TAPTS11			P114	75
TAPTS12			P115	74
TECRT0	I	Timer TMT0 clear input signal	P72	114
TECRT1	I	Timer TMT1 clear input signal	P75	4
TENCT00	I	Timer TMT0 encoder input	P70	112
TENCT01			P71	113
TENCT10	I	Timer TMT1 encoder input	P74	3
TENCT11			P73	144
TEVTAA0	I	Timer TAA0 event input	P13	54
TEVTAA1	I	Timer TAA1 event input	P10	51
TEVTAA2	I	Timer TAA2 event input	P17	62
TEVTAA3	I	Timer TAA3 event input	P14	55
TEVTAA4	I	Timer TAA4 event input	P23	66
TEVTAA5	I	Timer TAA5 event input	P20	63
TEVTAA6	I	Timer TAA6 event input	P27	70
TEVTAA7	I	Timer TAA7 event input	P24	67
TEVTAA8	I	Timer TAA8 event input	P101	91
TEVTAA9	I	Timer TAA9 event input	P103	93
TEVTS0	I	Timer TMS0 event input	P110	87
TEVTS1	I	Timer TMS1 event input	P113	76
TEVTT0	I	Timer TMT0 event input	P74	3
TEVTT1	I	Timer TMT1 event input	P70	112

Table 2-17 Alphabetic pin functions list (4/6)

Pin name	I/O	Pin function	Port	Pin location
TIAA00	I	Timer TAA0 capture trigger input	P10	51
TIAA01			P11	52
TIAA10	I	Timer TAA1 capture trigger input	P12	53
TIAA11			P13	54
TIAA20	I	Timer TAA2 capture trigger input	P14	55
TIAA21			P15	60
TIAA30	I	Timer TAA3 capture trigger input	P16	61
TIAA31			P17	62
TIAA40	I	Timer TAA4 capture trigger input	P20	63
TIAA41			P21	64
TIAA50	I	Timer TAA5 capture trigger input	P22	65
TIAA51			P23	66
TIAA60	I	Timer TAA6 capture trigger input	P24	67
TIAA61			P25	68
TIAA70	I	Timer TAA7 capture trigger input	P26	69
TIAA71			P27	70
TIAA80	I	Timer TAA8 capture trigger input	P100	88
TIAA81			P101	91
TIAA90	I	Timer TAA9 capture trigger input	P102	92
TIAA91			P103	93
TIT00	I	Timer TMT0 capture trigger input	P70	112
TIT01			P71	113
TIT10	I	Timer TMT1 capture trigger input	P73	144
TIT11			P74	3
TOAA00	O	Timer TAA0 pulse signal output	P10	51
TOAA01			P11	52
TOAA10	O	Timer TAA1 pulse signal output	P12	53
TOAA11			P13	54
TOAA20	O	Timer TAA2 pulse signal output	P14	55
TOAA21			P15	60
TOAA30	O	Timer TAA3 pulse signal output	P16	61
TOAA31			P17	62
TOAA40	O	Timer TAA4 pulse signal output	P20	63
TOAA41			P21	64
TOAA50	O	Timer TAA5 pulse signal output	P22	65
TOAA51			P23	66
TOAA60	O	Timer TAA6 pulse signal output	P24	67
TOAA61			P25	68
TOAA70	O	Timer TAA7 pulse signal output	P26	69
TOAA71			P27	70

Table 2-17 Alphabetic pin functions list (5/6)

Pin name	I/O	Pin function	Port	Pin location
TOAA80	O	Timer TAA8 pulse signal output	P100	88
TOAA81			P101	91
TOAA90	O	Timer TAA9 pulse signal output	P102	92
TOAA91			P103	93
TOS00	O	Timer TMS0 pulse signal outputs Outputs TOS01 to TOS06 can be controlled by an emergency shut-off. See “Emergency shut-off configuration” on page 34.	P50	94
TOS01			P51	95
TOS02			P52	96
TOS03			P53	97
TOS04			P54	98
TOS05			P55	99
TOS06			P56	100
TOS07			P57	101
TOS10	O	Timer TMS1 pulse signal outputs Outputs TOS11 to TOS16 can be controlled by an emergency shut-off. See “Emergency shut-off configuration” on page 34.	P60	102
TOS11			P61	103
TOS12			P62	104
TOS13			P63	105
TOS14			P64	106
TOS15			P65	107
TOS16			P66	108
TOS17			P67	111
TOT00	O	Timer TMT0 pulse signal output	P70	112
TOT01			P71	113
TOT10	O	Timer TMT1 pulse signal output	P73	144
TOT11			P74	3
TTRGAA0	I	Timer TAA0 trigger input	P12	53
TTRGAA1	I	Timer TAA1 trigger input	P11	52
TTRGAA2	I	Timer TAA2 trigger input	P16	61
TTRGAA3	I	Timer TAA3 trigger input	P15	60
TTRGAA4	I	Timer TAA4 trigger input	P22	65
TTRGAA5	I	Timer TAA5 trigger input	P21	64
TTRGAA6	I	Timer TAA6 trigger input	P26	69
TTRGAA7	I	Timer TAA7 trigger input	P25	68
TTRGAA8	I	Timer TAA8 trigger input	P100	88
TTRGAA9	I	Timer TAA9 trigger input	P102	92
TTRGS0	I	Timer TMS0 trigger input	P111	86
TTRGS1	I	Timer TMS1 trigger input	P114	75
TTRGT0	I	Timer TMT0 trigger input	P73	144
TTRGT1	I	Timer TMT1 trigger input	P71	113
TXDC0	O	UARTC0 transmit data	P31	32
TXDC1	O	UARTC1 transmit data	P33	39

**Table 2-17 Alphabetic pin functions list (6/6)**

Pin name	I/O	Pin function	Port	Pin location
VDD15	–	Core supply voltage	–	1, 19, 37, 56, 79, 90, 109, 140
VDD30	–	I/O supply voltage	–	36, 59, 84, 142
VSS15	–	Core supply ground	–	2, 20, 38, 57, 78, 89, 110, 141
VSS30	–	I/O supply ground	–	35, 58, 83, 143
X1	–	Main oscillator terminals	–	12
X2		Crystal connection/external clock input for system clock oscillation (X2 opens when the external clock is input)	–	13

### 2.3.5 Port group 0

Port group 0 is an 5-bit port group. This port group is only for input alternative mode. It comprises pins for the following functions:

- Non-maskable interrupt (NMI)
- External interrupt (INTP0 to INTP3)
- Emergency shut-off control of Timer TMS (ESO0, ESO1)
- Trigger input of A/D Converter (ADTRG0, ADTRG1)

Port group 0 includes the following pins:

**Table 2-18 Port group 0: pin functions and port types**

Pin functions in different modes		Pin function after reset	Port type	Noise filter <sup>a</sup>	Input charact. <sup>b</sup>
Port mode	Alternative mode <sup>c</sup>				
P00 (I)	NMI (I)	P00 (I)	2	A	S
P01 (I)	ESO0 (I)/ INTP0 (I)	P01 (I)	2	A	S
P02 (I)	ESO1 (I)/ INTP1 (I)	P02 (I)	2	A	S
P03 (I)	ADTRG0 (I)/ INTP2 (I)	P03 (I)	2	D	S
P04 (I)	ADTRG1 (I)/ INTP3 (I)	P04 (I)	2	D	S

a) A: analog noise filter; D: digital noise filter; —: no noise filter

The filters are only effective for the alternative input functions.

b) S: Schmitt trigger; nS: non-Schmitt trigger

c) Since all alternative functions are pure input functions, no PMCN register is required.

**Table 2-19 Port group 0: configuration registers**

Register	Address	Initial value	Used bits							
P0 <sup>a</sup>	FFFF F400 <sub>H</sub>	—	0	0	0	P04	P03	P02	P01	P00

a) Note that this register is read-only since port 0 is for input only.

**Access** All 8-bit registers can be accessed in 8-bit or 1-bit units.

### 2.3.6 Port group 1

Port group 1 is an 8-bit port group. In alternative mode, it comprises pins for the following functions:

- Timer TAA0 channels  
(TOAA00, TOAA01, TIAA00, TIAA01, TEVTAA0, TTRGAA0)
- Timer TAA1 channels  
(TOAA10, TOAA11, TIAA10, TIAA11, TEVTAA1, TTRGAA1)
- Timer TAA2 channels  
(TOAA20, TOAA21, TIAA20, TIAA21, TEVTAA2, TTRGAA2)
- Timer TAA3 channels  
(TOAA30, TOAA31, TIAA30, TIAA31, TEVTAA3, TTRGAA3)

Port group 1 includes the following pins:

**Table 2-20 Port group 1: pin functions and port types**

Pin functions in different modes			Pin function after reset	Port type	Noise filter <sup>a</sup>	Input charact. <sup>b</sup>
Port mode (PMCnm = 0)	Alternative mode (PMCnm = 1)					
	Function 1 PFC = 0	Function 2 PFC = 1				
P10 (I/O)	TIAA00 (I)/ TEVTAA1 (I)	TOAA00 (O)	P10 (I)	5-K	D	S
P11 (I/O)	TIAA01 (I)/ TTRGAA1 (I)	TOAA01 (O)	P11 (I)	5-K	D	S
P12 (I/O)	TIAA10 (I)/ TTRGAA0 (I)	TOAA10 (O)	P12 (I)	5-K	D	S
P13 (I/O)	TIAA11 (I)/ TEVTAA0 (I)	TOAA11 (O)	P13 (I)	5-K	D	S
P14 (I/O)	TIAA20 (I)/ TEVTAA3 (I)	TOAA20 (O)	P14 (I)	5-K	D	S
P15 (I/O)	TIAA21 (I)/ TTRGAA3 (I)	TOAA21 (O)	P15 (I)	5-K	D	S
P16 (I/O)	TIAA30 (I)/ TTRGAA2 (I)	TOAA30 (O)	P16 (I)	5-K	D	S
P17 (I/O)	TIAA31 (I)/ TEVTAA2 (I)	TOAA31 (O)	P17 (I)	5-K	D	S

a) A: analog noise filter; D: digital noise filter; —: no noise filter  
The filters are only effective for the alternative input functions.

b) S: Schmitt trigger; nS: non-Schmitt trigger

**Table 2-21 Port group 1: configuration registers**

Register	Address	Initial value	Used bits							
PMC1	FFFF F442 <sub>H</sub>	00 <sub>H</sub>	PMC17	PMC16	PMC15	PMC14	PMC13	PMC12	PMC11	PMC10
PM1	FFFF F422 <sub>H</sub>	FF <sub>H</sub>	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10
PFC1	FFFF F462 <sub>H</sub>	00 <sub>H</sub>	PFC17	PFC16	PFC15	PFC14	PFC13	PFC12	PFC11	PFC10
P1	FFFF F402 <sub>H</sub>	00 <sub>H</sub>	P17	P16	P15	P14	P13	P12	P11	P10

**Access** All 8-bit registers can be accessed in 8-bit or 1-bit units.



### 2.3.7 Port group 2

Port group 2 is an 8-bit port group. In alternative mode, it comprises pins for the following functions:

- Timer TAA4 channels  
(TOAA40, TOAA41, TIAA40, TIAA41, TEVTAA4, TTRGAA4)
- Timer TAA5 channels  
(TOAA50, TOAA51, TIAA50, TIAA51, TEVTAA5, TTRGAA5)
- Timer TAA6 channels  
(TOAA60, TOAA61, TIAA60, TIAA61, TEVTAA6, TTRGAA6)
- Timer TAA7 channels  
(TOAA70, TOAA71, TIAA70, TIAA71, TEVTAA7, TTRGAA7)

Port group 2 includes the following pins:

**Table 2-22 Port group 2: pin functions and port types**

Pin functions in different modes			Pin function after reset	Port type	Noise filter <sup>a</sup>	Input charact. <sup>b</sup>
Port mode (PMCnm = 0)	Alternative mode (PMCnm = 1)					
	Function 1 PFC = 0	Function 2 PFC = 1				
P20 (I/O)	TIAA40 (I)/ TEVTAA5 (I)	TOAA40 (O)	P20 (I)	5-K	D	S
P21 (I/O)	TIAA41 (I)/ TTRGAA5 (I)	TOAA41 (O)	P21 (I)	5-K	D	S
P22 (I/O)	TIAA50 (I)/ TTRGAA4 (I)	TOAA50 (O)	P22 (I)	5-K	D	S
P23 (I/O)	TIAA51 (I)/ TEVTAA4 (I)	TOAA51 (O)	P23 (I)	5-K	D	S
P24 (I/O)	TIAA60 (I)/ TEVTAA7 (I)	TOAA60 (O)	P24 (I)	5-K	D	S
P25 (I/O)	TIAA61 (I)/ TTRGAA7 (I)	TOAA61 (O)	P25 (I)	5-K	D	S
P26 (I/O)	TIAA70 (I)/ TTRGAA6 (I)	TOAA70 (O)	P26 (I)	5-K	D	S
P27 (I/O)	TIAA71 (I)/ TEVTAA6 (I)	TOAA71 (O)	P27 (I)	5-K	D	S

a) A: analog noise filter; D: digital noise filter; —: no noise filter  
The filters are only effective for the alternative input functions.

b) S: Schmitt trigger; nS: non-Schmitt trigger

**Table 2-23 Port group 2: configuration registers**

Register	Address	Initial value	Used bits							
PMC2	FFFF F444 <sub>H</sub>	00 <sub>H</sub>	PMC27	PMC26	PMC25	PMC24	PMC23	PMC22	PMC21	PMC20
PM2	FFFF F424 <sub>H</sub>	FF <sub>H</sub>	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20
PFC2	FFFF F464 <sub>H</sub>	00 <sub>H</sub>	PFC27	PFC26	PFC25	PFC24	PFC23	PFC22	PFC21	PFC20
P2	FFFF F404 <sub>H</sub>	00 <sub>H</sub>	P27	P26	P25	P24	P23	P22	P21	P20

**Access** All 8-bit registers can be accessed in 8-bit or 1-bit units.

### 2.3.8 Port group 3

Port group 3 is an 8-bit port group. In alternative mode, it comprises pins for the following functions:

- External interrupts (INTP4, INTP5)
- UARTC0 receive/transmit data (RXDC0, TXDC0)
- UARTC1 receive/transmit data (RXDC1, TXDC1)
- CAN0 receive/transmit data (FCRXD0, FCTXD0)
- CAN1 receive/transmit data (FCRXD1, FCTXD1)

Port group 3 includes the following pins:

**Table 2-24 Port group 3: pin functions and port types**

Pin functions in different modes		Pin function after reset	Port type	Noise filter <sup>a</sup>	Input charact. <sup>b</sup>
Port mode (PMCnm = 0)	Alternative mode (PMCnm = 1)				
P30 (I/O)	RXDC0 (I)/INTP4 (I)	P30 (I)	5-K	D	S
P31 (I/O)	TXDC0 (O)	P31 (I)	5-K	—	S
P32 (I/O)	RXDC1 (I)/INTP5 (I)	P32 (I)	5-K	D	S
P33 (I/O)	TXDC1 (O)	P33 (I)	5-K	—	S
P34 (I/O)	FCRXD0 (I)	P34 (I)	5-K	—	S
P35 (I/O)	FCTXD0 (O)	P35 (I)	5-K	—	S
P36 (I/O)	FCRXD1 (I)	P36 (I)	5-K	—	S
P37 (I/O)	FCTXD1 (O)	P37 (I)	5-K	—	S

a) A: analog noise filter; D: digital noise filter; —: no noise filter  
The filters are only effective for the alternative input functions.

b) S: Schmitt trigger; nS: non-Schmitt trigger

**Table 2-25 Port group 3: configuration registers**

Register	Address	Initial value	Used bits							
PMC3	FFFF F446 <sub>H</sub>	00 <sub>H</sub>	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30
PM3	FFFF F426 <sub>H</sub>	FF <sub>H</sub>	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30
P3	FFFF F406 <sub>H</sub>	00 <sub>H</sub>	P37	P36	P35	P34	P33	P32	P31	P30

**Access** All 8-bit registers can be accessed in 8-bit or 1-bit units.

### 2.3.9 Port group 4

Port group 4 is a 6-bit port group. In alternative mode, it comprises pins for the following functions:

- Clocked Serial Interface CSIB0 data/clock line (SIB0, SOB0, SCKB0)
- Clocked Serial Interface CSIB1 data/clock line (SIB1, SOB1, SCKB1)

Port group 4 includes the following pins:

**Table 2-26 Port group 4: pin functions and port types**

Pin functions in different modes		Pin function after reset	Port type	Noise filter <sup>a</sup>	Input charact. <sup>b</sup>
Port mode (PMCnm = 0)	Alternative mode (PMCnm = 1)				
P40 (I/O)	SIB0 (I)	P40 (I)	5-K	—	S
P41 (I/O)	SOB0 (O)	P41 (I)	5-K	—	S
P42 (I/O)	SCKB0 (I/O)	P42 (I)	5-K	—	S
P43 (I/O)	SIB1 (I)	P43 (I)	5-K	—	S
P44 (I/O)	SOB1 (O)	P44 (I)	5-K	—	S
P45 (I/O)	SCKB1 (I/O)	P45 (I)	5-K	—	S

a) A: analog noise filter; D: digital noise filter; —: no noise filter  
The filters are only effective for the alternative input functions.

b) S: Schmitt trigger; nS: non-Schmitt trigger

**Table 2-27 Port group 4: configuration registers**

Register	Address	Initial value	Used bits							
PMC4	FFFF F448 <sub>H</sub>	00 <sub>H</sub>	0	0	PMC45	PMC44	PMC43	PMC42	PMC41	PMC40
PM4	FFFF F428 <sub>H</sub>	FF <sub>H</sub>	1	1	PM45	PM44	PM43	PM42	PM41	PM40
P4	FFFF F408 <sub>H</sub>	00 <sub>H</sub>	0	0	P45	P44	P43	P42	P41	P40

**Access** All 8-bit registers can be accessed in 8-bit or 1-bit units.

### 2.3.10 Port group 5

Port group 5 is an 8-bit port group. In alternative mode, it comprises pins for the following functions:

- Timer TMS0 pulse signal output (TOS00 to TOS07)

**Note** The outputs TOS01 to TOS06 of timer TMS0 can be controlled by an emergency shut-off. See “Emergency shut-off configuration” on page 34 for details.

Port group 5 includes the following pins:

**Table 2-28 Port group 5: pin functions and port types**

Pin functions in different modes		Pin function after reset	Port type	Noise filter <sup>a</sup>	Input charact. <sup>b</sup>
Port mode (PMCnm = 0)	Alternative mode (PMCnm = 1)				
P50 (I/O)	TOS00 (O)	P50 (I)	5-K	—	S
P51 (I/O)	TOS01 (O) <sup>c</sup>	P51 (I)	5-K	—	S
P52 (I/O)	TOS02 (O) <sup>c</sup>	P52 (I)	5-K	—	S
P53 (I/O)	TOS03 (O) <sup>c</sup>	P53 (I)	5-K	—	S
P54 (I/O)	TOS04 (O) <sup>c</sup>	P54 (I)	5-K	—	S
P55 (I/O)	TOS05 (O) <sup>c</sup>	P55 (I)	5-K	—	S
P56 (I/O)	TOS06 (O) <sup>c</sup>	P56 (I)	5-K	—	S
P57 (I/O)	TOS07 (O)	P57 (I)	5-K	—	S

- a) A: analog noise filter; D: digital noise filter; —: no noise filter  
The filters are only effective for the alternative input functions.
- b) S: Schmitt trigger; nS: non-Schmitt trigger
- c) Subject to emergency shut-off function

**Table 2-29 Port group 5: configuration registers**

Register	Address	Initial value	Used bits							
PMC5	FFFF F44A <sub>H</sub>	00 <sub>H</sub>	PMC57	PMC56	PMC55	PMC54	PMC53	PMC52	PMC51	PMC50
PM5	FFFF F42A <sub>H</sub>	FF <sub>H</sub>	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50
P5	FFFF F40A <sub>H</sub>	00 <sub>H</sub>	P57	P56	P55	P54	P53	P52	P51	P50
PESC5	FFFF F888 <sub>H</sub>	00 <sub>H</sub>	0	0	0	0	ESO0EN	0	ESO0ED1	ESO0ED0
ESOST5	FFFF F88A <sub>H</sub>	00 <sub>H</sub>	ESO0ST	0	0	0	0	0	0	0
PESMK5	FFFF F890 <sub>H</sub>	00 <sub>H</sub>	0	0	0	0	0	0	PESMK51	PESMK50

**Access** All 8-bit registers can be accessed in 8-bit or 1-bit units.

**Note** Writing to registers PMC5, PM5, P5, PESC5, ESOST5 and RESMK5 is protected by a special sequence of instructions. Please refer to “Write enable register” on page 38 for details.

### 2.3.11 Port group 6

Port group 6 is an 8-bit port group. In alternative mode, it comprises pins for the following functions:

- Timer TMS1 pulse signal output (TOS10 to TOS17)

**Note** The outputs TOS11 to TOS16 of timer TMS1 can be controlled by an emergency shut-off. See “Emergency shut-off configuration” on page 34 for details.

Port group 6 includes the following pins:

**Table 2-30 Port group 6: pin functions and port types**

Pin functions in different modes		Pin function after reset	Port type	Noise filter <sup>a</sup>	Input charact. <sup>b</sup>
Port mode (PMCnm = 0)	Alternative mode (PMCnm = 1)				
P60 (I/O)	TOS10 (O)	P60 (I)	5-K	—	S
P61 (I/O)	TOS11 (O) <sup>c</sup>	P61 (I)	5-K	—	S
P62 (I/O)	TOS12 (O) <sup>c</sup>	P62 (I)	5-K	—	S
P63 (I/O)	TOS13 (O) <sup>c</sup>	P63 (I)	5-K	—	S
P64 (I/O)	TOS14 (O) <sup>c</sup>	P64 (I)	5-K	—	S
P65 (I/O)	TOS15 (O) <sup>c</sup>	P65 (I)	5-K	—	S
P66 (I/O)	TOS16 (O) <sup>c</sup>	P66 (I)	5-K	—	S
P67 (I/O)	TOS17 (O)	P67 (I)	5-K	—	S

- a) A: analog noise filter; D: digital noise filter; —: no noise filter  
The filters are only effective for the alternative input functions.
- b) S: Schmitt trigger; nS: non-Schmitt trigger
- c) Subject to emergency shut-off function

**Table 2-31 Port group 6: configuration registers**

Register	Address	Initial value	Used bits							
PMC6	FFFF F44C <sub>H</sub>	00 <sub>H</sub>	PMC67	PMC66	PMC65	PMC64	PMC63	PMC62	PMC61	PMC60
PM6	FFFF F42C <sub>H</sub>	FF <sub>H</sub>	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60
P6	FFFF F40C <sub>H</sub>	00 <sub>H</sub>	P67	P66	P65	P64	P63	P62	P61	P60
PESC6	FFFF F88C <sub>H</sub>	00 <sub>H</sub>	0	0	0	0	ESO1EN	0	ESO1ED1	ESO1ED0
ESOST6	FFFF F88E <sub>H</sub>	00 <sub>H</sub>	ESO1ST	0	0	0	0	0	0	0
PESMK6	FFFF F892 <sub>H</sub>	00 <sub>H</sub>	0	0	0	0	0	0	PESMK61	PESMK60

**Access** All 8-bit registers can be accessed in 8-bit or 1-bit units.

**Note** Writing to registers PMC6, PM6, P6, PESC6, ESOST6 and PESMK6 is protected by a special sequence of instructions. Please refer to “Write enable register” on page 38 for details.

### 2.3.12 Port group 7

Port group 7 is a 6-bit port group. In alternative mode, it comprises pins for the following functions:

- Timer TMT0 channels  
(TIT00, TIT01, TOT00, TOT01, TEVTT0, TTRGT0, TECRT0, TENCT00, TENCT01)
- Timer TMT1 channels  
(TIT10, TIT11, TOT10, TOT11, TEVTT1, TTRGT1, TECRT1, TENCT10, TENCT11)
- External interrupt (INTP12)
- Auxiliary frequency output (AFO)

Port group 7 includes the following pins:

**Table 2-32 Port group 7: pin functions and port types**

Pin functions in different modes			Pin function after reset	Port type	Noise filter <sup>a</sup>	Input charact. <sup>b</sup>
Port mode (PMCnm = 0)	Alternative mode (PMCnm = 1)					
	Function 1 PFC = 0	Function 2 PFC = 1				
P70 (I/O)	TIT00 (I)/ TEVTT1 (I)/ TENCT00 (I)	TOT00 (O)	P70 (I)	5-K	D	S
P71 (I/O)	TIT01 (I)/ TTRGT1 (I)/ TENCT01 (I)	TOT01 (O)	P71 (I)	5-K	D	S
P72 (I/O)	TECRT0 (I)	INTP12 (I)	P72 (I)	5-K	D	S
P73 (I/O)	TIT10 (I)/ TTRGT0 (I)/ TENCT11 (I)	TOT10 (O)	P73 (I)	5-K	D	S
P74 (I/O)	TIT11 (I)/ TEVTT0 (I)/ TENCT10 (I)	TOT11 (O)	P74 (I)	5-K	D	S
P75 (I/O)	TECRT1 (I)	AFO (O)	P75 (I)	5-K	D	S

a) A: analog noise filter; D: digital noise filter; -: no noise filter  
The filters are only effective for the alternative input functions.

b) S: Schmitt trigger; nS: non-Schmitt trigger

**Table 2-33 Port group 7: configuration registers**

Register	Address	Initial value	Used bits							
PMC7	FFFF F44E <sub>H</sub>	00 <sub>H</sub>	0	0	PMC75	PMC74	PMC73	PMC72	PMC71	PMC70
PM7	FFFF F42E <sub>H</sub>	FF <sub>H</sub>	1	1	PM75	PM74	PM73	PM72	PM71	PM70
PFC7	FFFF F46E <sub>H</sub>	00 <sub>H</sub>	0	0	PFC75	PFC74	PFC73	PFC72	PFC71	PFC70
P7	FFFF F40E <sub>H</sub>	00 <sub>H</sub>	0	0	P75	P74	P73	P72	P71	P70

**Access** All 8-bit registers can be accessed in 8-bit or 1-bit units.

### 2.3.13 Port group 8

Port group 8 is a 7-bit port group. In alternative mode, it comprises pins for the following functions:

- Clocked Serial Interface CSIE0 data/clock line (SIE0, SOE0, SCKE0)
- Clocked Serial Interface CSIE0 chip select signal (SSCSE00 to SCSE03)
- Clocked Serial Interface CSIB0 serial slave select input (SSB0)
- External interrupts (INTP6 to INTP8)

Port group 8 includes the following pins:

**Table 2-34 Port group 8: pin functions and port types**

Pin functions in different modes			Pin function after reset	Port type	Noise filter <sup>a</sup>	Input charact. <sup>b</sup>
Port mode (PMCnm = 0)	Alternative mode (PMCnm = 1)					
	Function 1 PFC = 0	Function 2 PFC = 1				
P80 (I/O)	SIE0 (I)	prohibited	P80 (I)	5-K	–	S
P81 (I/O)	SOE0 (O)	prohibited	P81 (I)	5-K	–	S
P82 (I/O)	SCKE0 (I/O)	prohibited	P82 (I)	5-K	–	S
P83 (I/O)	SCSE00 (O)	INTP6 (I)	P83 (I)	5-K	D	S
P84 (I/O)	SCSE01 (O)	INTP7 (I)	P84 (I)	5-K	D	S
P85 (I/O)	SCSE02 (O)	INTP8 (I)	P85 (I)	5-K	D	S
P86 (I/O)	SCSE03 (O)	SSB0 (I)	P86 (I)	5-K	–	S

a) A: analog noise filter; D: digital noise filter; –: no noise filter  
The filters are only effective for the alternative input functions.

b) S: Schmitt trigger; nS: non-Schmitt trigger

**Table 2-35 Port group 8: configuration registers**

Register	Address	Initial value	Used bits							
PMC8	FFFF F450 <sub>H</sub>	00 <sub>H</sub>	0	PMC86	PMC85	PMC84	PMC83	PMC82	PMC81	PMC80
PM8	FFFF F430 <sub>H</sub>	FF <sub>H</sub>	1	PM86	PM85	PM84	PM83	PM82	PM81	PM80
PFC8	FFFF F470 <sub>H</sub>	00 <sub>H</sub>	0	PFC86	PFC85	PFC84	PFC83	0	0	0
P8	FFFF F410 <sub>H</sub>	00 <sub>H</sub>	0	P86	P85	P84	P83	P82	P81	P80

**Access** All 8-bit registers can be accessed in 8-bit or 1-bit units.

### 2.3.14 Port group 9

Port group 9 is a 7-bit port group. In alternative mode, it comprises pins for the following functions:

- Clocked Serial Interface CSIE0 chip select signal (SCSE04 to SCSE07)
- Clocked Serial Interface CSIE1 data/clock line (SIE1, SOE1, SCKE1)
- Clocked Serial Interface CSIE1 chip select signal (SCSE10 to SCSE13)
- Clocked Serial Interface CSIB1 serial slave select input (SSB1)
- External interrupt (INTP9 to INTP11)

Port group 9 includes the following pins:

**Table 2-36 Port group 9: pin functions and port types**

Pin functions in different modes					Pin function after reset	Port type	Noise filter <sup>a</sup>	Input charact. <sup>b</sup>
Port mode (PMCnm = 0)	Alternative mode (PMCnm = 1)							
	PFCE = 0		PFCE = 1					
	Function 1 PFC = 0	Function 2 PFC = 1	Function 3 PFC = 0	Function 4 PFC = 1				
P90 (I/O)	SIE1 (I)	prohibited	prohibited	prohibited	P90 (I)	5-K	–	S
P91 (I/O)	SOE1 (O)	prohibited	prohibited	prohibited	P91 (I)	5-K	–	S
P92 (I/O)	SCKE1 (I/O)	prohibited	prohibited	prohibited	P92 (I)	5-K	–	S
P93 (I/O)	SCSE10 (O)	INTP9 (I)	SCSE04 (O)	prohibited	P93 (I)	5-K	D	S
P94 (I/O)	SCSE11 (O)	INTP10 (I)	SCSE05 (O)	prohibited	P94 (I)	5-K	D	S
P95 (I/O)	SCSE12 (O)	INTP11 (I)	SCSE06 (O)	prohibited	P95 (I)	5-K	D	S
P96 (I/O)	SCSE13 (O)	SSB1 (I)	SCSE07 (O)	prohibited	P96 (I)	5-K	–	S

a) A: analog noise filter; D: digital noise filter; –: no noise filter  
The filters are only effective for the alternative input functions.

b) S: Schmitt trigger; nS: non-Schmitt trigger

**Table 2-37 Port group 9: configuration registers**

Register	Address	Initial value	Used bits							
PMC9	FFFF F452 <sub>H</sub>	00 <sub>H</sub>	0	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90
PM9	FFFF F432 <sub>H</sub>	FF <sub>H</sub>	1	PM96	PM95	PM94	PM93	PM92	PM91	PM90
PFC9	FFFF F472 <sub>H</sub>	00 <sub>H</sub>	0	PFC96	PFC95	PFC94	PFC93	0	0	0
PFCE9	FFFF FC32 <sub>H</sub>	00 <sub>H</sub>	0	PFCE96	PFCE95	PFCE94	PFCE93	0	0	0
P9	FFFF F412 <sub>H</sub>	00 <sub>H</sub>	0	P96	P95	P94	P93	P92	P91	P90

**Access** All 8-bit registers can be accessed in 8-bit or 1-bit units.



### 2.3.15 Port group 10

Port group 10 is a 4-bit port group. In alternative mode, it comprises pins for the following functions:

- Timer TAA8 channels  
(TOAA80, TOAA81, TIAA80, TIAA81, TEVTAA8, TTRGAA8)
- Timer TAA9 channels  
(TOAA90, TOAA91, TIAA90, TIAA91, TEVTAA9, TTRGAA9)

Port group 10 includes the following pins:

**Table 2-38 Port group 10: pin functions and port types**

Pin functions in different modes			Pin function after reset	Port type	Noise filter <sup>a</sup>	Input charact. <sup>b</sup>
Port mode (PMCnm = 0)	Alternative mode (PMCnm = 1)					
	Function 1 PFC = 0	Function 2 PFC = 1				
P100 (I/O)	TIAA80 (I)/ TTRGAA8 (I)	TOAA80 (O)	P100 (I)	5-K	D	S
P101 (I/O)	TIAA81 (I)/ TEVTAA8 (I)	TOAA81 (O)	P101 (I)	5-K	D	S
P102 (I/O)	TIAA90 (I)/ TTRGAA9 (I)	TOAA90 (O)	P102 (I)	5-K	D	S
P103 (I/O)	TIAA91 (I)/ TEVTAA9 (I)	TOAA91 (O)	P103 (I)	5-K	D	S

a) A: analog noise filter; D: digital noise filter; –: no noise filter  
The filters are only effective for the alternative input functions.

b) S: Schmitt trigger; nS: non-Schmitt trigger.

**Table 2-39 Port group 10: configuration registers**

Register	Address	Initial value	Used bits							
PMC10	FFFF F454 <sub>H</sub>	00 <sub>H</sub>	0	0	0	0	PMC103	PMC102	PMC101	PMC100
PM10	FFFF F434 <sub>H</sub>	FF <sub>H</sub>	1	1	1	1	PM103	PM102	PM101	PM100
PFC10	FFFF F474 <sub>H</sub>	00 <sub>H</sub>	0	0	0	0	PFC103	PFC102	PFC101	PFC100
P10	FFFF F414 <sub>H</sub>	00 <sub>H</sub>	0	0	0	0	P103	P102	P101	P100

**Access** All 8-bit registers can be accessed in 8-bit or 1-bit units.

### 2.3.16 Port group 11

Port group 11 is a 6-bit port group. In alternative mode, it comprises pins for the following functions:

- Timer TMS0 input channels  
(TAPTS00 to TAPTS02, TEVTS0, TTRGS0)
- Timer TMS1 input channels  
(TAPTS10 to TAPTS12, TEVTS1, TTRGS1)

Port group 11 includes the following pins:

**Table 2-40 Port group 11: pin functions and port types**

Pin functions in different modes		Pin function after reset	Port type	Noise filter <sup>a</sup>	Input charact. <sup>b</sup>
Port mode (PMCnm = 0)	Alternative mode (PMCnm = 1)				
P110 (I/O)	TAPTS00 (I)/ TEVTS0 (I)	P110 (I)	5-K	D	S
P111 (I/O)	TAPTS01 (I)/ TTRGS0 (I)	P111 (I)	5-K	D	S
P112 (I/O)	TAPTS02 (I)	P112 (I)	5-K	D	S
P113 (I/O)	TAPTS10 (I)/ TEVTS1 (I)	P113 (I)	5-K	D	S
P114 (I/O)	TAPTS11 (I)/ TTRGS1 (I)	P114 (I)	5-K	D	S
P115 (I/O)	TAPTS12 (I)	P115 (I)	5-K	D	S

- a) A: analog noise filter; D: digital noise filter; —: no noise filter  
The filters are only effective for the alternative input functions.
- b) S: Schmitt trigger; nS: non-Schmitt trigger

**Table 2-41 Port group 11: configuration registers**

Register	Address	Initial value	Used bits							
PMC11	FFFF F456 <sub>H</sub>	00 <sub>H</sub>	0	0	PMC115	PMC114	PMC113	PMC112	PMC111	PMC110
PM11	FFFF F436 <sub>H</sub>	FF <sub>H</sub>	1	1	PM115	PM114	PM113	PM112	PM111	PM110
P11	FFFF F416 <sub>H</sub>	00 <sub>H</sub>	0	0	P115	P114	P113	P112	P111	P110

**Access** All 8-bit registers can be accessed in 8-bit or 1-bit units.

## 2.4 Noise Elimination

The input signals at some pins are passing a filter to remove noise and glitches. The microcontroller supports both analog and digital filters.

If the pins are used as general purpose I/O ports Pmn, the signals are not filtered.

Following alternative function signals are not filtered at the ports:

- UARTCn receive data signals RXDCn, but UARTCn has built-in noise rejection filters
- CANn receive data signals FCRXDn
- CSIBn signals: SCKBn, SIBn, SSBn
- CSIE n signals: SCKEn, SIEn

In the following all signals are listed, which are filtered.

The port group functions tables in section “Port group 0” on page 51 and all following also indicate filtered signals.

### 2.4.1 Analog filtered inputs

The analog filter suppresses input pulses that are shorter than a specified pulse width (refer to the Data Sheet). This assures the hold time for the external interrupt signals.

The analog filter is not effective if the corresponding pin works as a general purpose I/O port.

Table 2-42 lists all signals which are passed through an analog filter to remove noise and glitches.

**Table 2-42 Input signals with analog filters**

Function	Pin name	Inputs with analog filter
Reset	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$
	$\overline{\text{DRST}}$	$\overline{\text{DRST}}$
Interrupts	P00/NMI	NMI
	P01/ESO0/INTP0	ESO0/INTP0
	P02/ESO1/INTP1	ESO1/INTP1

## 2.4.2 Digitally filtered inputs

The inputs of the peripherals listed in *Table 2-43* are passed through a digital filter to remove noise and glitches.

The digital filter is not effective if the corresponding pin works as a general purpose I/O port.

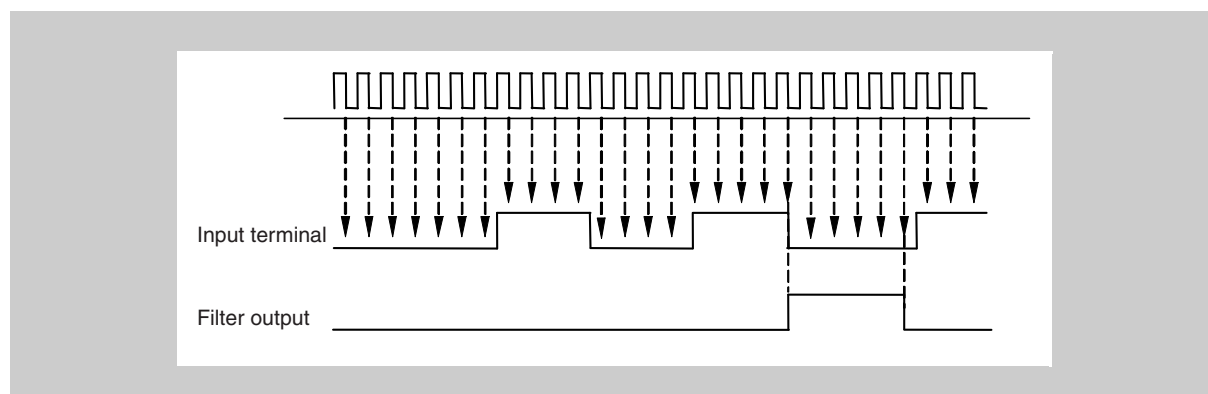
**Table 2-43 Input signals with digital filters (1/2)**

Function	Pin name	Inputs with digital filter
Interrupts	P03/ADTRG0/INTP2	ADTRG0/INTP2
	P04/ADTRG1/INTP3	ADTRG1/INTP3
	P30/RXDC0/INTP4	INTP4
	P32/RXDC1/INTP5	INTP5
	P83/SCSE00/INTP6	INTP6
	P84/SCSE01/INTP7	INTP7
	P85/SCSE02/INTP8	INTP8
	P93/SCSE10/INTP9/SCSE04	INTP9
	P94/SCSE11/INTP10/SCSE05	INTP10
	P95/SCSE12/INTP11/SCSE06	INTP11
	P72/TECRT0/INTP12	TECRT0/INTP12
Timers A	P10/TIAA00/TEVTAA1/TOAA00	TIAA00/TEVTAA1
	P11/TIAA01/TTRGAA1/TOAA01	TIAA01/TTRGAA1
	P12/TIAA10/TTRGAA0/TOAA10	TIAA10/TTRGAA0
	P13/TIAA11/TEVTAA0/TOAA11	TIAA11/TEVTAA0
	P14/TIAA20/TEVTAA3/TOAA20	TIAA11/TEVTAA0
	P15/TIAA21/TTRGAA3/TOAA21	TIAA21/TTRGAA3
	P16/TIAA30/TTRGAA2/TOAA30	TIAA30/TTRGAA2
	P17/TIAA31/TEVTAA2/TOAA31	TIAA31/TEVTAA2
	P20/TIAA40/TEVTAA5/TOAA40	TIAA40/TEVTAA5
	P21/TIAA41/TTRGAA5/TOAA41	TIAA41/TTRGAA5
	P22/TIAA50/TTRGAA4/TOAA50	TIAA41/TTRGAA5
	P23/TIAA51/TEVTAA4/TOAA51	TIAA51/TEVTAA4
	P24/TIAA60/TEVTAA7/TOAA60	TIAA60/TEVTAA7
	P25/TIAA61/TTRGAA7/TOAA61	TIAA61/TTRGAA7
	P26/TIAA70/TTRGAA6/TOAA70	TIAA70/TTRGAA6
	P27/TIAA71/TEVTAA6/TOAA71	TIAA71/TEVTAA6
	P100/TIAA80/TTRGAA8/TOAA80	TIAA80/TTRGAA8
	P101/TIAA81/TEVTAA8/TOAA81	TIAA81/TEVTAA8
	P102/TIAA90/TTRGAA9/TOAA90	TIAA90/TTRGAA9
	P103/TIAA91/TEVTAA9/TOAA91	TIAA91/TEVTAA9

**Table 2-43 Input signals with digital filters (2/2)**

Function	Pin name	Inputs with digital filter
Timers S	P110/TAPTS00/TEVTS0	TAPTS00/TEVTS0
	P111/TAPTS01/TTRGS0	TAPTS01/TTRGS0
	P112/TAPTS02	TAPTS02
	P113/TAPTS10/TEVTS1	TAPTS10/TEVTS1
	P114/TAPTS11/TTRGS1	TAPTS11/TTRGS1
	P115/TAPTS12	TAPTS12
Timers T	P70/TIT00/TEVTT1/TOT00/TENCT00	TIT00/TEVTT1/TENCT00
	P71/TIT01/TTRGT1/TOT01/TENCT01	TIT01/TTRGT1/TENCT01
	P72/TECRT0/INTP12	TECRT0/INTP12
	P73/TIT10/TTRGT0/TOT10/TENCT11	TIT10/TTRGT0/TENCT11
	P74/TIT11/TEVTT0/TOT11/TENCT10	TIT11/TEVTT0/TENCT10
	P75/TECRT1/AFO	TECRT1

**Filter operation** The input terminal signal is sampled with the sampling frequency  $f_s$ . Spikes shorter than 4 sampling cycles are suppressed and no internal signal is generated. Pulses longer than 5 sampling cycles are recognized as valid pulses and an internal signal is generated. For pulses between 4 and 5 sampling cycles, the behaviour is not defined. The filter operation is illustrated in *Figure 2-3*.

**Figure 2-3 Digital noise removal example**

The minimum input terminal pulse width to be validated is defined by the sampling frequency  $f_s$ . The sampling frequency  $f_s$  can be chosen as 32 MHz or 4 MHz by the noise rejection time control register NCR.

**Table 2-44 Digital noise removal features**

Sampling frequency $f_s$	Minimum pulse width to generate an internal signal
32 MHz	125 – 156.25 ns
4 MHz	1 – 1.25 $\mu$ sec

The sampling clock for the noise elimination circuit can be individually adjusted for each of the aforementioned external input signals. The clock is specified by the 8-bit register NRC.

**(1) NRC - Noise rejection time control register**

The 8-bit NRC register specifies the sampling clock for the noise elimination circuit.

**Access** This register can be read/written in 8-bit and 1-bit units.

**Address** FFFF F7A0<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by  $\overline{\text{RESET}}$ .

7	6	5	4	3	2	1	0
NRC7	NRC6	NRC5	NRC4	NRC3	NRC2	NRC1	NRC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 2-45 NRC register contents**

Bit position	Bit name	Function
7 to 0	NRC[7:0]	Sets the sampling clock for the noise elimination circuit for the corresponding input signal: 0: 32 MHz 1: 4 MHz For an assignment of bit positions to input signals see <i>Table 2-46 on page 66</i> .

**Table 2-46 Assignment of input signals to bit positions for register NRC**

Bit position	Bit name	Input signals	Description
7	NRC7	<ul style="list-style-type: none"> <li>TAPTS10 to TAPTS12, TEVTS1, TTRGS1</li> <li>TAPTS00 to TAPTS02, TEVTS0, TTRGS0</li> </ul>	<ul style="list-style-type: none"> <li>Timer TMS1 input channels</li> <li>Timer TMS0 input channels</li> </ul>
6	NRC6	<ul style="list-style-type: none"> <li>TIAA90, TIAA91, TEVTAA9, TTRGAA9</li> <li>TIAA80, TIAA81, TEVTAA8, TTRGAA8</li> </ul>	<ul style="list-style-type: none"> <li>Timer TAA9 input channels</li> <li>Timer TAA8 input channels</li> </ul>
5	NRC5	<ul style="list-style-type: none"> <li>TIAA70, TIAA71, TEVTAA7, TTRGAA7</li> <li>TIAA60, TIAA61, TEVTAA6, TTRGAA6</li> </ul>	<ul style="list-style-type: none"> <li>Timer TAA7 input channels</li> <li>Timer TAA6 input channels</li> </ul>
4	NRC4	<ul style="list-style-type: none"> <li>TIAA50, TIAA51, TEVTAA5, TTRGAA5</li> <li>TIAA40, TIAA41, TEVTAA4, TTRGAA4</li> </ul>	<ul style="list-style-type: none"> <li>Timer TAA5 input channels</li> <li>Timer TAA4 input channels</li> </ul>
3	NRC3	<ul style="list-style-type: none"> <li>TIAA30, TIAA31, TEVTAA3, TTRGAA3</li> <li>TIAA20, TIAA21, TEVTAA2, TTRGAA2</li> </ul>	<ul style="list-style-type: none"> <li>Timer TAA3 input channels</li> <li>Timer TAA2 input channels</li> </ul>
2	NRC2	<ul style="list-style-type: none"> <li>TIAA10, TIAA11, TEVTAA1, TTRGAA1</li> <li>TIAA00, TIAA01, TEVTAA0, TTRGAA0</li> </ul>	<ul style="list-style-type: none"> <li>Timer TAA1 input channels</li> <li>Timer TAA0 input channels</li> </ul>
1	NRC1	<ul style="list-style-type: none"> <li>TIT10, TIT11, TEVTT1, TTRGT1, TECRT1, TENCT11, TENCT10</li> <li>TIT00, TIT01, TEVTT0, TTRGT0, TECRT0, TENCT01, TENCT00</li> </ul>	<ul style="list-style-type: none"> <li>Timer TMT1 input channels</li> <li>Timer TMT0 input channels</li> </ul>
0	NRC0	<ul style="list-style-type: none"> <li>ADTRG0, ADTRG1</li> <li>INTP2 to INTP12</li> </ul>	<ul style="list-style-type: none"> <li>ADC0 and ADC1 trigger input</li> <li>External interrupts 2 to 12</li> </ul>

## 2.5 Pin Functions during and after Reset

The following table summarizes the status of the pins during reset and after reset release in normal operation mode, i.e. MODE1 = 0.

**Table 2-47 Pin functions during and after  $\overline{\text{RESET}}$**

Pin name		$\overline{\text{RESET}} = 0$	$\overline{\text{RESET}} = 1$	
			Normal operation	CPU in HALT mode, DMA operating
N-Wire interface				
DCK	$\overline{\text{DRST}} = \text{L}$	Hi-Z	Hi-Z	Hi-Z
	$\overline{\text{DRST}} = \text{H}$	operating	operating	operating
DDI	$\overline{\text{DRST}} = \text{L}$	Hi-Z	Hi-Z	Hi-Z
	$\overline{\text{DRST}} = \text{H}$	operating	operating	operating
DDO	$\overline{\text{DRST}} = \text{L}$	Hi-Z	L	L
	$\overline{\text{DRST}} = \text{H}$	operating	operating	operating
DMS	$\overline{\text{DRST}} = \text{L}$	Hi-Z	Hi-Z	Hi-Z
	$\overline{\text{DRST}} = \text{H}$	operating	operating	operating
Pins in alternative mode, which are not mentioned above				
Alternative inputs		–	operating	operating
Alternative outputs		–	operating	operating
Pins in port mode, which are not mentioned above				
Port inputs		Hi-Z	operating	–
Port outputs		–	operating	retains the last status

## 2.6 Recommended Connection of unused Pins

If a pin is not used, it is recommended to connect it as follows:

**Table 2-48 Recommended connection of unused pins**

Pin	Recommended connection
<b>Port pins</b>	
Pins of port group 0: P0[4:0]	connect to VSS3x
Pins of all other port groups: P1[7:0], P2[7:0], P3[7:0], P4[5:0], P5[7:0], P6[7:0], P7[5:0], P8[6:0], P9[6:0], P10[3:0], P11[5:0]	<ul style="list-style-type: none"> <li>output ports: leave open</li> <li>input ports: connect to VDD3x or VSS3x via resistors</li> </ul>
<b>Non-port pins</b>	
ANI00 to ANI09	connect to AVDD or AVSS0
ANI10 to ANI19	connect to AVDD or AVSS1
DCK, DDI, DMS	connect to VDD3x via resistors
DDO	leave open
$\overline{\text{DRST}}$	connect to VSS3x

**Caution** Pins to be connected via a resistor to VSS3x respectively VDD3x must be connected via separate resistors.

- Note**
1. For pins requiring a pull-up/-down resistor a resistance of 1 to 10 K $\Omega$  is recommended.
  2. If the overall maximum output current exceeds its maximum value the output buffer can be damaged. We recommend the placement of a series resistor to prevent damage in case of accidentally enabled outputs. Refer to the absolute maximum rating parameter in the Data Sheet.



## 2.7 Package Pins Assignment

The following figure shows the pin location of a packaged device. Every pin is labelled with its pin name. This diagram is a schematic top view only, for detailed package drawing please regard the electrical specification document.

**Note** When connecting the pins, consider the following:

- VDD15 / VSS15: 1.5V supply (for details please regard the electrical specification document)
- CVDD / CVSS: 1.5V supply (for details please regard the electrical specification document)
- VDD30 / VSS30: 3.3V supply (for details please regard the electrical specification document)
- AVDD / AVSS: 3.3V supply (for details please regard the electrical specification document)
- AVSS: Avoid injecting noise from VSS30 or VSS15 by providing a low impedance connection of AVSS to a location with low noise ground potential.

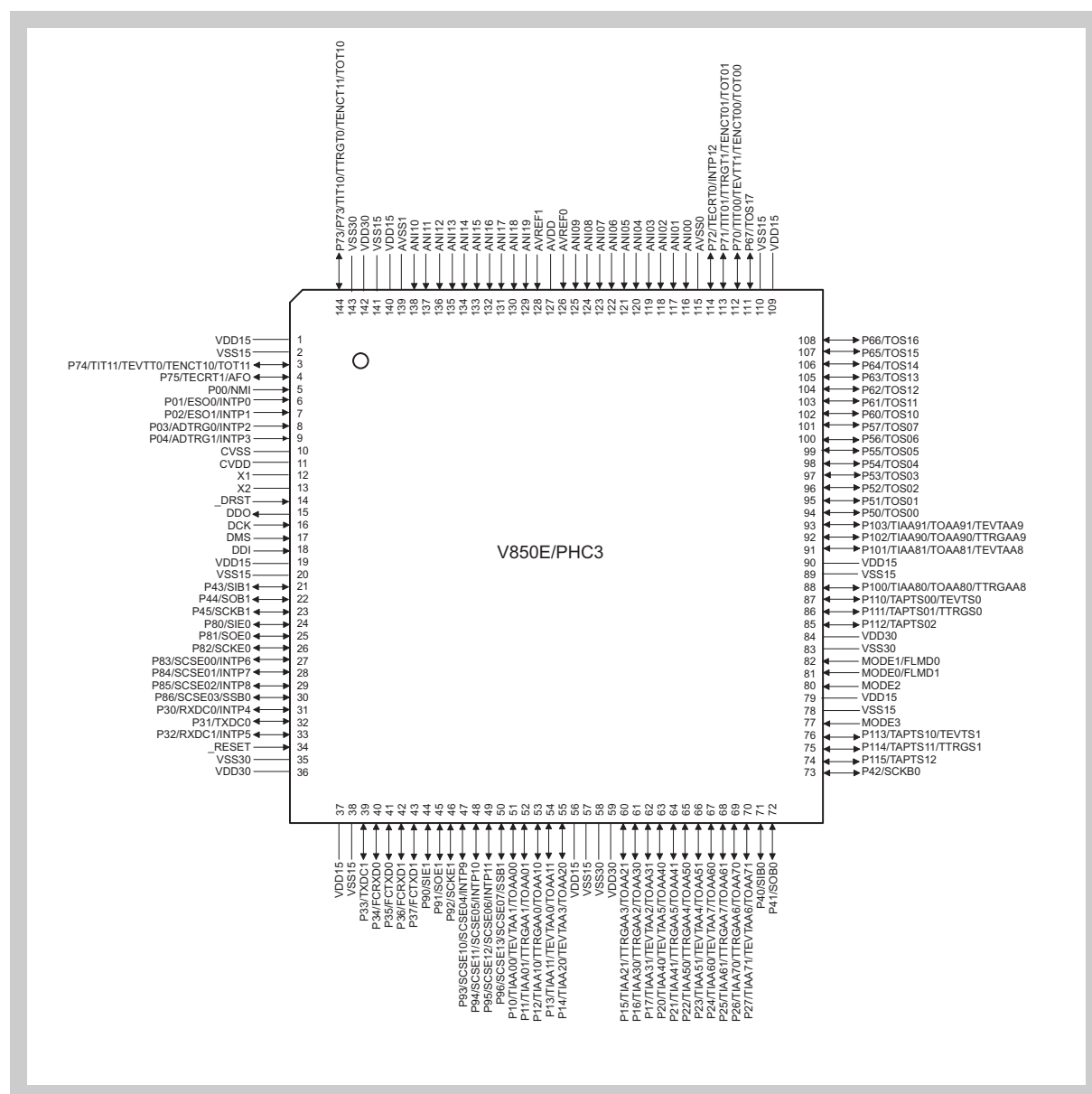


Figure 2-4 Pin overview

## Chapter 3 CPU System Functions

This chapter describes the registers of the CPU, the operation modes, the address space and the memory areas.

### 3.1 Overview

The CPU is founded on Harvard architecture and it supports a RISC instruction set. Basic instructions can be executed in one clock period. Optimized five-stage pipelining is supported. This improves instruction execution speed.

The floating-point unit (FPU) supports online diagnostics as well as applications demanding floating-point representation such as physical sensor data or automatically generated code.

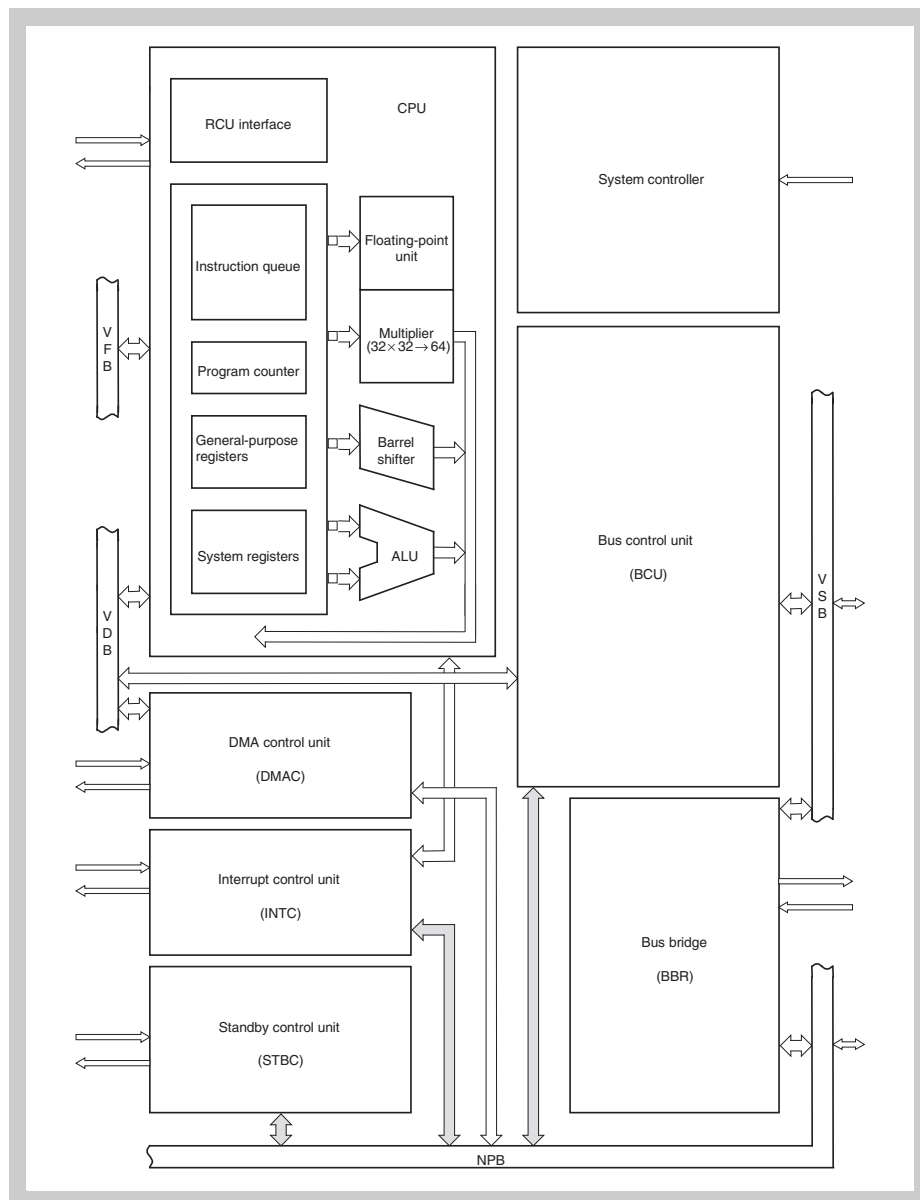
In order to make the microcontroller ideal for use in digital control applications, a 32-bit hardware multiplier enables this CPU to support multiply instructions, saturated multiply instructions, bit operation instructions, etc.

**Features summary** The CPU has the following special features:

- Memory space:
  - 64 MB linear program space
  - 4 GB linear data space
- 32 general purpose registers
- Internal 32-bit architecture
- Five-stage pipeline
- Efficient multiplication and division instructions
- Saturation logic (saturated operation instructions)
- Floating-point arithmetic unit  
(single precision, 32 bits, according to IEEE 754-85 standard)
- Barrel shifter (32-bit shift in one clock cycle)
- Instruction formats: long and short
- Four types of bit manipulation instructions: set, clear, not, test

### 3.1.1 Description

The figure below shows a block diagram of the microcontroller, focusing on the CPU and modules that interact with the CPU directly. *Table 3-1* lists the bus types.



### Figure 3-1 CPU system

The shaded busses are used for accessing the configuration registers of the concerned modules.

### Table 3-1 Bus types

Bus type	Function
NPB – Peripheral bus	Bus interface to the peripherals (internal bus).
VSB – System bus	Bus interface to the Memory Controller for access to external memory and to the NPB bus bridge BBR.
VFB – Fetch bus	Interface to the internal ROM (mask ROM or flash ROM).
VDB – Data bus	Interface to the internal RAM.

## 3.2 CPU Register Set

There are three categories of registers:

- General purpose registers
- System registers
- Floating-point arithmetic registers

All registers are 32-bit registers. An overview is given in the figure below. For details, refer to V850E1 User's Manual Architecture.

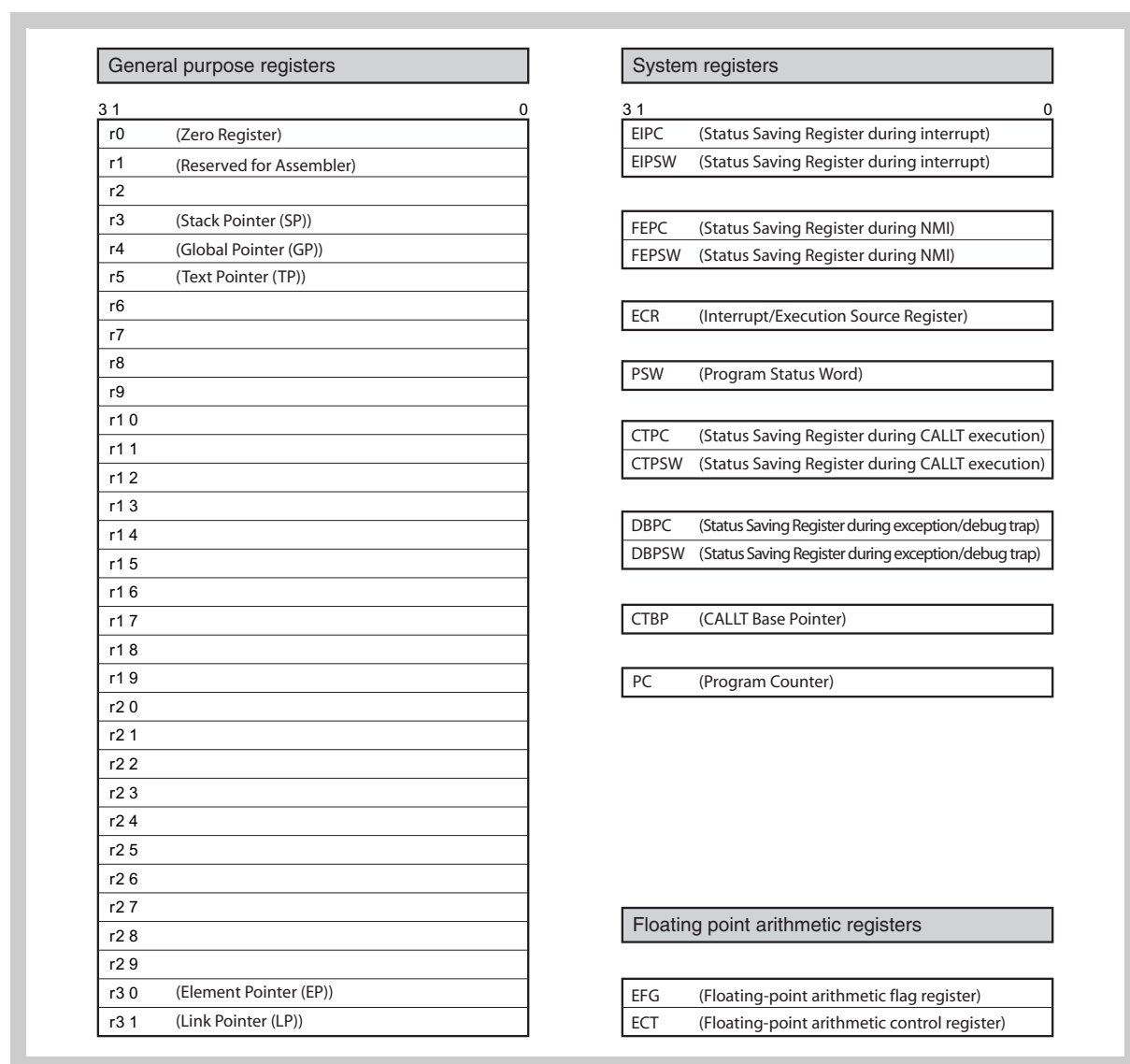


Figure 3-2 CPU register set

Some registers are write protected. That means, writing to those registers is protected by a special sequence of instructions. Refer to “*Write Protected Registers*” on page 97 for more details.

### 3.2.1 General purpose registers (r0 to r31)

Each of the 32 general purpose registers can be used as a data variable or address variable.

However, the registers r0, r1, r3 to r5, r30, and r31 may implicitly be used by the assembler/compiler (see *Table 3-2*). For details refer to the documentation of your assembler/compiler.

**Table 3-2 General purpose registers**

Register name	Usage	Operation
r0	Zero register	Always holds 0. It is used for operations using 0 and offset 0 addressing. <sup>a</sup>
r1	Assembler-reserved register	Used for 32-bit direct addressing. <sup>b</sup>
r2	User address/data variable register	
r3	Stack pointer	Used to generate stack frame when function is called. <sup>b</sup>
r4	Global pointer	Used to access global variable in data area. <sup>b</sup>
r5	Text pointer	Used to indicate the start of the text area (where program code is located). <sup>b</sup>
r6 to r29	User address/data variable registers	
r30	Element pointer	Base pointer when memory is accessed by means of instructions SLD (short format load) and SST (short format store). <sup>a</sup>
r31	Link pointer	Used when calling a function. <sup>b</sup>

a) Registers r0 and r30 are used by dedicated instructions.

b) Registers r1, r3, r4, r5, and r31 may be used by the assembler/compiler.

**Caution** Before using registers r1, r3 to r5, r30, and r31, 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.

### 3.2.2 System register set

System registers control the status of the CPU and hold interrupt information. Additionally, the program counter holds the instruction address during program execution.

To read/write the system registers, use instructions LDSR (load to system register) or STSR (store contents of system register), respectively, with a specific system register number (RegID) indicated below.

The program counter states an exception. It cannot be accessed via LDSR or STSR instructions. No regID is allocated to the program counter.

**Example** STSR 0, r2

Stores the contents of system register 0 (EIPC) in general purpose register r2.

**System register numbers** The table below gives an overview of all system registers and their system register number (regID). It shows whether a load/store instruction is allowed (x) for the register or not (–).

**Table 3-3 System register numbers**

regID	System register name	Shortcut	Operand specification	
			LDSR	STSR
0	Status saving register during interrupt (stores contents of PC)	EIPC	x	x
1	Status saving register during interrupt (stores contents of PSW)	EIPSW	x	x
2	Status saving register during non-maskable interrupts (stores contents of PC)	FEPC	x	x
3	Status saving register during non-maskable interrupts (stores contents of PSW)	FEPSW	x	x
4	Interrupt source register	ECR	–	x
5	Program status word	PSW	x	x
6 to 15	Reserved (operations that access these register numbers cannot be guaranteed).		–	–
16	Status saving register during CALLT execution (stores contents of PC)	CTPC	x	x
17	Status saving register during CALLT execution (stores contents of PSW)	CTPSW	x	x
18	Status saving register during exception/debug trap (stores contents of PC)	DBPC	x <sup>a</sup>	x
19	Status saving register during exception/debug trap (stores contents of PSW)	DBPSW	x <sup>a</sup>	x
20	CALLT base pointer	CTBP	x	x
21 to 31	Reserved (operations that access these register numbers cannot be guaranteed).		–	–

<sup>a)</sup> Reading from this register is only enabled between a DBTRAP exception (exception handler address 0000 0060<sub>H</sub>) and the exception handler terminating DBRET instruction. DBTRAP exceptions are generated upon ILGOP detections (refer to “Interrupt Controller (INTC)” on page 99).

**(1) PC - Program counter**

The program counter holds the instruction address during program execution. The lower 26 bits are valid, and bits 31 to 26 are fixed to 0. If a carry occurs from bit 25 to 26, it is ignored. Branching to an odd address cannot be performed. Bit 0 is fixed to 0.

**Access** This register can not be accessed by any instruction.

**Initial Value** 0000 0000<sub>H</sub>. The program counter is cleared by  $\overline{\text{RESET}}$ .

31	26	25	1	0
fixed to 0			instruction address during execution	
				0

**(2) EIPC, FEPC, DBPC, CTPC - PC saving registers**

The PC saving registers save the contents of the program counter for different occasions, see *Table 3-4*.

When one of the occasions listed in *Table 3-4* occurs, except for some instructions, the address of the instruction following the one being executed is saved to the saving registers.

For more details refer to *Table 3-9 on page 80* and to the “*Interrupt Controller (INTC)*” on page 99.

All PC saving registers are built up as the PC, with the initial value 0xxx xxxx<sub>H</sub> (x = undefined).

**Table 3-4 PC saving registers**

Register	Shortcut	Saves contents of PC in case of
Status saving register during interrupt	EIPC	<ul style="list-style-type: none"> <li>software exception</li> <li>maskable interrupt</li> </ul>
Status saving register during non-maskable interrupts	FEPC	<ul style="list-style-type: none"> <li>non-maskable interrupt</li> </ul>
Status saving register during exception/debug trap	DBPC <sup>a</sup>	<ul style="list-style-type: none"> <li>exception trap</li> <li>debug trap</li> <li>debug break</li> <li>during a single-step operation</li> </ul>
Status saving register during CALLT execution	CTPC	<ul style="list-style-type: none"> <li>execution of CALLT instruction</li> </ul>

<sup>a)</sup> Reading from this register is only enabled in debug mode. Otherwise, the read value is undefined.

**Note** When multiple interrupt servicing is enabled, the contents of EIPC or FEPC must be saved by program because only one PC saving register for maskable interrupts and non-maskable interrupts is provided, respectively.

**Caution** When setting the value of any of the PC saving registers, use even values (bit 0 = 0). If bit 0 is set to 1, the setting of this bit is ignored. This is because bit 0 of the program counter is fixed to 0.



**(3) PSW - Program status word**

The 32-bit program status word is a collection of flags that indicates the status of the program (result of instruction execution) and the status of the CPU.

If the bits in the register are modified by the LDSR instruction, the PSW will take on the new value immediately after the LDSR instruction has been executed.

**Initial Value** 0000 0020<sub>H</sub>. The program status is initialized by  $\overline{\text{RESET}}$ .

31	8	7	6	5	4	3	2	1	0
fixed to 0		NP	EP	ID	SAT	CY	OV	S	Z
R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 3-5 PSW register contents (1/2)**

Bit position	Flag	Function
7	NP	Indicates that non-maskable interrupt (NMI) servicing is in progress. This flag is set when NMI request is acknowledged, and multiple interrupt servicing is disabled. 0: NMI servicing is not in progress. 1: NMI servicing is in progress.
6	EP	Indicates that exception processing is in progress. This flag is set when an exception occurs. Even when this bit is set, interrupt requests can be acknowledged. 0: Exception processing is not in progress. 1: Exception processing is in progress.
5	ID	Indicates whether a maskable interrupt request can be acknowledged. 0: Interrupts enabled. 1: Interrupts disabled. <b>Note:</b> Setting this flag will disable interrupt requests even while the LDSR instruction is being executed.
4	SAT <sup>a</sup>	For saturated operation processing instructions only: Indicates that the operation result is saturated due to overflow. 0: Not saturated. 1: Saturated. <b>Note:</b> 1. This is a cumulative flag: The bit is not automatically cleared if subsequent instructions lead to not saturated results. To clear this bit, use the LDSR instruction to set PSW.SAT = 0. 2. In a general arithmetic operation this bit is neglected. It is neither set nor cleared.
3	CY	Carry/borrow flag. Indicates whether a carry or borrow occurred as a result of the operation. 0: Carry or borrow did not occur 1: Carry or borrow occurred.

**Table 3-5 PSW register contents (2/2)**

Bit position	Flag	Function
2	OV <sup>a</sup>	Overflow flag. Indicates whether an overflow occurred as a result of the operation. 0: Overflow did not occur. 1: Overflow occurred.
1	S <sup>a</sup>	Sign flag. Indicates whether the result of the operation is negative. 0: Result is positive or zero. 1: Result is negative.
0	Z	Zero flag. Indicates whether the result of the operation is zero. 0: Result is not zero. 1: Result is zero.

<sup>a)</sup> In the case of saturate instructions, the SAT, S, and OV flags will be set according to the result of the operation as shown in the table below. Note that the SAT flag is set only when the OV flag has been set during a saturated operation.

**Saturated operation instructions** The following table shows the setting of flags PWS.SAT, PWS.OV, and PWS.S, depending on the status of the operation result.

**Table 3-6 Saturation-processed operation result**

Status of operation result	Flag status			Saturation-processed operation result
	SAT	OV	S	
Maximum positive value exceeded	1	1	0	7FFF FFFF <sub>H</sub>
Maximum negative value exceeded	1	1	1	8000 0000 <sub>H</sub>
Positive (maximum not exceeded)	x <sup>a</sup>	0	0	Operation result itself
Negative (maximum not exceeded)			1	

<sup>a)</sup> Retains the value before operation.

**(4) EIPSW, FEPSW, DBPSW, CTPSW saving registers**

The PSW saving registers save the contents of the program status word for different occasions, see *Table 3-4*.

When one of the occasions listed in *Table 3-4* occurs, the current value of the PSW is saved to the saving registers.

All PSW saving registers are built up as the PSW, with the initial value 0000 0xxx<sub>H</sub> (x = undefined).

**Table 3-7 PSW saving registers**

Register	Shortcut	Saves contents of PSW in case of
Status saving register during interrupt	EIPSW	<ul style="list-style-type: none"> <li>software exception</li> <li>maskable interrupt</li> </ul>
Status saving register during non-maskable interrupts	FEPSW	<ul style="list-style-type: none"> <li>non-maskable interrupt</li> </ul>
Status saving register during exception/debug trap	DBPSW <sup>a</sup>	<ul style="list-style-type: none"> <li>exception trap</li> <li>debug trap</li> <li>debug break</li> <li>during a single-step operation</li> </ul>
Status saving register during CALLT execution	CTPSW	<ul style="list-style-type: none"> <li>execution of CALLT instruction</li> </ul>

a) Reading from this register is only enabled in debug mode. Otherwise, the read value is undefined.

**Note** When multiple interrupt servicing is enabled, the contents of EIPSW or FEPSW must be saved by program because only one PSW saving register for maskable interrupts and non-maskable interrupts is provided, respectively.

**Caution** Bits 31 to 26 of EIPC and bits 31 to 12 and 10 to 8 of EIPSW are reserved for future function expansion (fixed to 0). When setting the value of EIPC, FEPC, or CTPC, use even values (bit 0 = 0). If bit 0 is set to 1, the setting of this bit is ignored. This is because bit 0 of the program counter is fixed to 0.

**(5) ECR - Interrupt/exception source register**

The 32-bit ECR register displays the exception codes if an exception or an interrupt has occurred. With the exception code, the interrupt/exception source can be identified.

For a list of interrupts/exceptions and corresponding exception codes, see *Table 3-9 on page 80*.

**Initial Value** 0000 0000<sub>H</sub>. This register is cleared by  $\overline{\text{RESET}}$ .

31	26 25	0
FECC		EICC

**Table 3-8 ECR register contents**

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

The following table lists the exception codes.

**Table 3-9 Interrupt/execution codes**

Interrupt/Exception Source			Classification	Exception Code	Handler Address	Value restored to EIPC/FEPC
Name	Trigger					
Non-maskable interrupt (NMI)	refer to "Interrupt Controller (INTC)" on page 99		Interrupt	0010 <sub>H</sub>	0000 0010 <sub>H</sub>	next PC (see Note)
Maskable interrupt			Interrupt	refer to "Interrupt Controller (INTC)" on page 99	<ul style="list-style-type: none"> <li>higher 16 bits: 0000<sub>H</sub></li> <li>lower 16 bits: exception code</li> </ul>	next PC (see Note)
Software exception	TRAP0n (n = 0 to F <sub>H</sub> )	TRAP instruction	Exception	004n <sub>H</sub>	0000 0040 <sub>H</sub>	next PC
	TRAP1n (n = 0 to F <sub>H</sub> )	TRAP instruction	Exception	005n <sub>H</sub>	0000 0050 <sub>H</sub>	next PC
Exception trap (ILGOP)	Illegal instruction code		Exception	0060 <sub>H</sub>	0000 0060 <sub>H</sub>	next PC
Debug trap	DBTRAP instruction		Exception	0060 <sub>H</sub>	0000 0060 <sub>H</sub>	next PC

If an interrupt (maskable or non-maskable) is acknowledged during instruction execution, generally, the address of the instruction *following* the one being executed is saved to the saving registers, except when an interrupt is acknowledged during execution of one of the following instructions:

- load instructions (SLD.B, SLD.BU, SLD.H, SLD.HU, SLD.W)
- divide instructions (DIV, DIVH, DIVU, DIVHU)
- PREPARE, DISPOSE instruction (only if an interrupt is generated before the stack pointer is updated)

In this case, the address of the *interrupted* instruction is restored to the EIPC or FEPC, respectively. Execution is stopped, and after the completion of interrupt servicing the execution is resumed.

**(6) CTBP - CALLT base pointer**

The 32-bit CALLT base pointer is used with the CALLT instruction. The register content is used as a base address to generate both a 32-bit table entry address and a 32-bit target address.

Initial Value Undefined

[illegible]

a) These bits may be written, but write is ignored.

### 3.2.3 Floating-point arithmetic registers

Floating-point arithmetic unit (FPU) is controlled by means of the following registers:

**Table 3-10 FPU control registers**

Register name	Shortcut
Floating-point arithmetic flag register	EFG
Floating-point arithmetic control register	ECT

#### (1) EFG - Floating-point arithmetic flag register

The 32-bit EFG register holds the status of the FPU.

**Initial Value** 0000 0000<sub>H</sub>.

31	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	RO	IV	ZD	VF	UD	PR	0	0	0	0	TR	0	OV	S	Z

**Table 3-11 EFG register contents (1/2)**

Bit position	Bit name	Function
13	RO	Running Operation: Indicates whether the floating-point arithmetic unit is running. 0: Operation in progress. 1: FPU idle.
12	IV	InValid operation: Indicates that an invalid operation has been requested. 0: Normal operation. 1: Invalid operation detected.
11	ZD	Zero Divide: Indicates whether a division by 0 has been detected. 0: Normal operation. 1: Division by 0 detected.
10	VF	oVerFlow: Indicates that the result of executing a floating-point operation has overflowed. 0: No overflow generated. 1: Overflow generated.
9	UD	UnDervalue: Indicates that the result of executing a floating-point operation has underflowed. 0: No underflow generated. 1: Underflow generated.
8	PR	PRecision error: indicates that an accuracy failure occurred. 0: No accuracy failure occurred. 1: Accuracy failure occurred.
4	TR	This flag summarizes the state of the FPU: 0: Normal state. 1: Abnormal condition detected: one of the bits 12 to 8 of the ECT register is set. The setting conditions of this flag depend on the ECT register value.
2	OV	Overflow: Indicates whether an overflow occurred during floating-point to integer conversion 0: No overflow generated. 1: Overflow generated.
1	S	Sign: Indicates whether a floating-point operation result is negative. 0: Operation result is not negative. 1: Operation result is negative.

**Table 3-11 EFG register contents (2/2)**

Bit position	Bit name	Function
0	Z	Zero: Indicates whether a floating-point operation result is 0. 0: Operation result is not 0. 1: Operation result is 0.

**(2) ECT - Floating-point arithmetic control register**

The 32-bit ECT register controls the setting conditions of the TR flag in the EFG register. The TR flag is a logical OR between all the invalid operations the FPU can detect. Each bit of ECT is a mask bit for one condition.

**Initial Value** 0000 0000<sub>H</sub>.

31	13	12	11	10	9	8	7	0
0	IT	ZT	VT	UT	PT	0		

**Table 3-12 ECT register contents**

Bit position	Bit name	Function
12	IT	Enables invalid operation detection in the TR value calculation 0: IV is set when an invalid operation is detected 1: IV and TR are set when an invalid operation is detected
11	ZT	Enables zero divide operation detection in the TR value calculation 0: ZD is set when a zero divide operation is detected. 1: ZD and TR are set when a zero divide operation is detected.
10	VT	Enables overflow detection in the TR value calculation 0: VF is set when an overflow is detected. 1: VF and TR are set when an overflow is detected.
9	UT	Enables underflow detection in the TR value calculation 0: UD is set when an underflow is detected. 1: UD and TR are set when an underflow is detected.
8	PT	Enables accuracy fail detection in the TR value calculation 0: PR is set when an accuracy fail is detected. 1: PR and TR are set when an accuracy fail is detected.

### 3.3 Operation Modes

This section describes the operation modes of the CPU and how the modes are specified.

The following operation modes are available for the flash memory devices:

- Normal operation mode
- Flash programming mode

After reset release, the microcontroller starts to fetch instructions from an internal boot ROM which contains the internal firmware. The firmware checks the pins MODE0 to MODE3 to set the operation mode after reset release according to *Table 3-13*.

**Table 3-13 Selection of operation modes**

Pins				Operation Mode
MODE3	MODE2	MODE1/ FLMD0	MODE0/ FLMD1	
0	0	0	0	Normal operation mode (fetch from flash)
		1		Flash programming mode
all other				Reserved

- Note**
1. The MODE1 pin function is shared with the FLMD0 pin.
  2. Pins MODE0 and MODE3 must be connected to VSS3x.

#### 3.3.1 Normal operation mode

In normal operation mode, the internal flash memory is not re-programmed.

After reset release program execution is started at address 0000 0000<sub>H</sub>.

The lower 1 MB of the memory area is mapped to internal resources, for instance to the internal flash memory.

#### 3.3.2 Flash programming mode

In flash programming mode, the internal flash memory is erased and re-programmed.

After reset release, the firmware initiates loading of the user's program code from the external flash programmer and programs the flash memory.

After detaching the external flash programmer, the microcontroller can be started up with the new user's program in normal operation mode.

For more information see section “Flash Memory” on page 159.



## 3.4 Address Space

In the following sections, the address space of the CPU is explained. Size and addresses of CPU address space and physical address space are explained. The address range of data space and program space together with their wrap-around properties are presented.

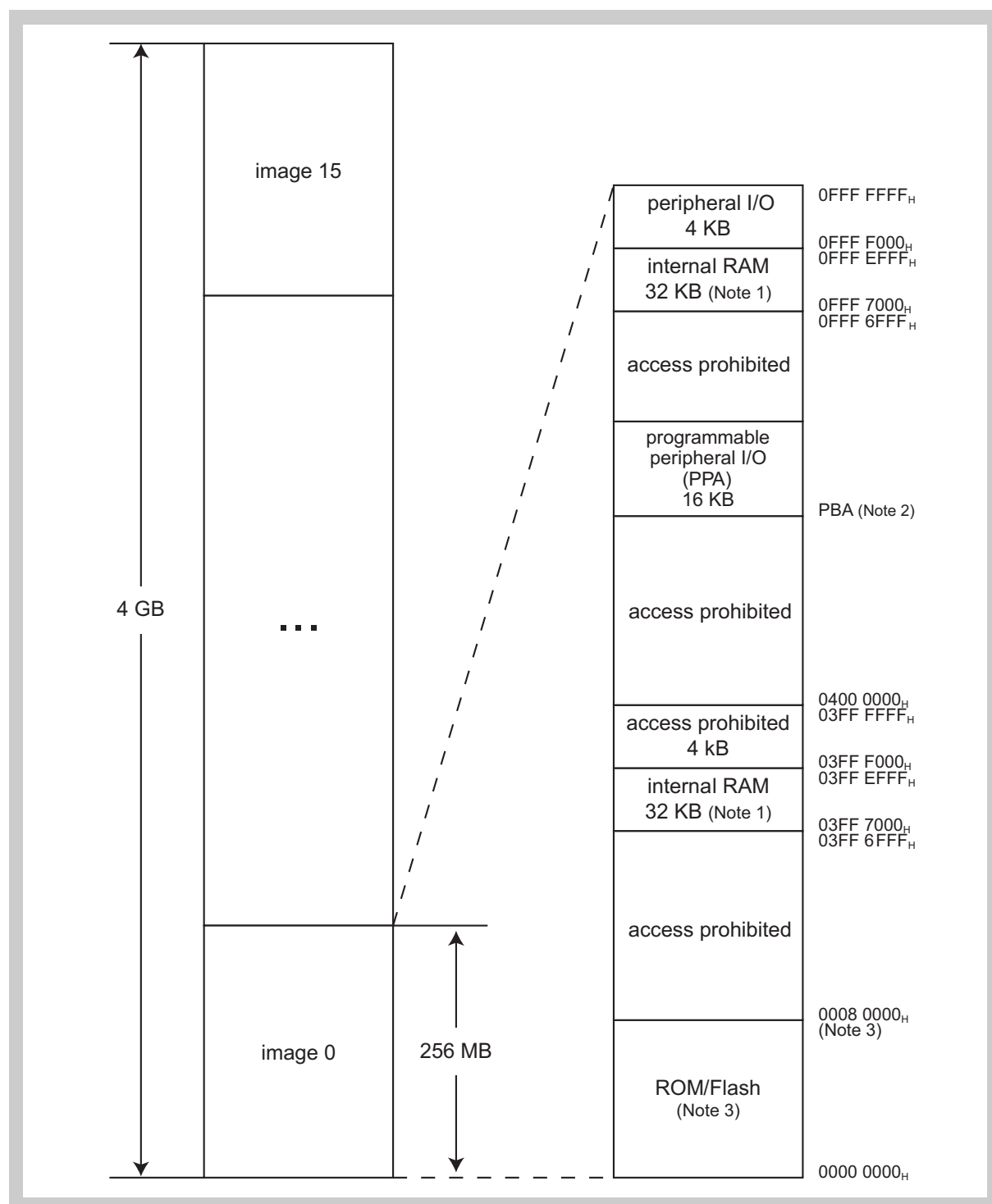
### 3.4.1 CPU address space and physical address space

The CPU supports the following address space:

- 4 GB CPU address space  
With the 32-bit general purpose registers, addresses for a 4 GB memory can be generated. This is the maximum address space supported by the CPU.
- 256 MB physical address space  
The CPU provides 256 MB physical address space. That means that a maximum of 256 MB internal or external memory can be accessed.

Any 32-bit address is translated to its corresponding physical address by ignoring bits 31 to 28 of the address. Thus, 16 addresses point to the same physical memory address. In other words, data at the physical address 0000 0000<sub>H</sub> can additionally be accessed by addresses 1000 0000<sub>H</sub>, 2000 0000<sub>H</sub>, ..., E000 0000<sub>H</sub>, or F000 0000<sub>H</sub>.

The 256 MB physical address space is seen as 16 images in the 4 GB CPU address space as shown in *Figure 3-3*.



**Figure 3-3** Images in the CPU address space

- Note**
1. The internal RAM area (03FF 7000<sub>H</sub> to 03FF EFFF<sub>H</sub>) is mirrored to the area 0FFF 7000<sub>H</sub> to 0FFF EFFF<sub>H</sub>. If data is written in one area, it appears also in the other area.
  2. The programmable peripheral base address is defined by the BPC register.
  3.  $\mu$ PD70F3485: 480 KB up to 0007 7FFF<sub>H</sub>

### 3.4.2 Program and data space

The CPU allows the following assignment of data and instructions to the CPU address space:

- 4 GB as data space  
The entire CPU address space can be used for operand addresses.
- 64 MB as program space  
Only the lower 64 MB of the CPU address space can be used for instruction addresses. When an instruction address for a branch instruction is calculated and moved to the program counter (PC), then bits 31 to 26 are set to zero.

Figure 3-4 shows the assignment of the CPU address space to data and program space.

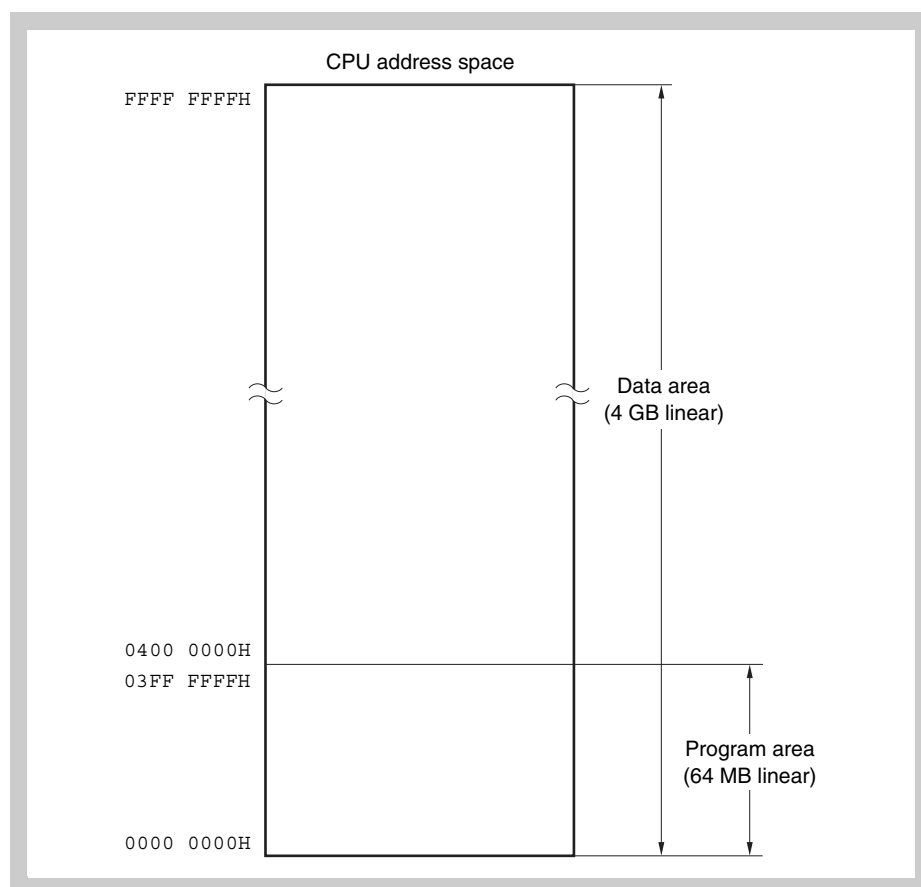


Figure 3-4 CPU address space

**(1) Wrap-around of data space**

If an operand address calculation exceeds 32 bits, only the lower 32 bits of the result are considered. Therefore, the addresses 0000 0000<sub>H</sub> and FFFF FFFF<sub>H</sub> are contiguous addresses. This results in a wrap-around of the data space:

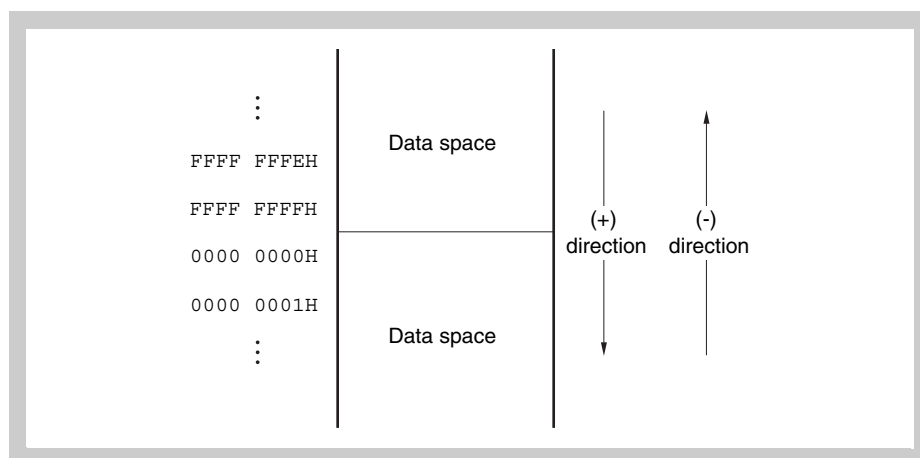


Figure 3-5 Wrap-around of data space

**(2) Wrap-around of program space**

If an instruction address calculation exceeds 26 bits, only the lower 26 bits of the result are considered. Therefore, the addresses 0000 0000<sub>H</sub> and 03FF FFFF<sub>H</sub> are contiguous addresses. This results in a wrap-around of the program space:

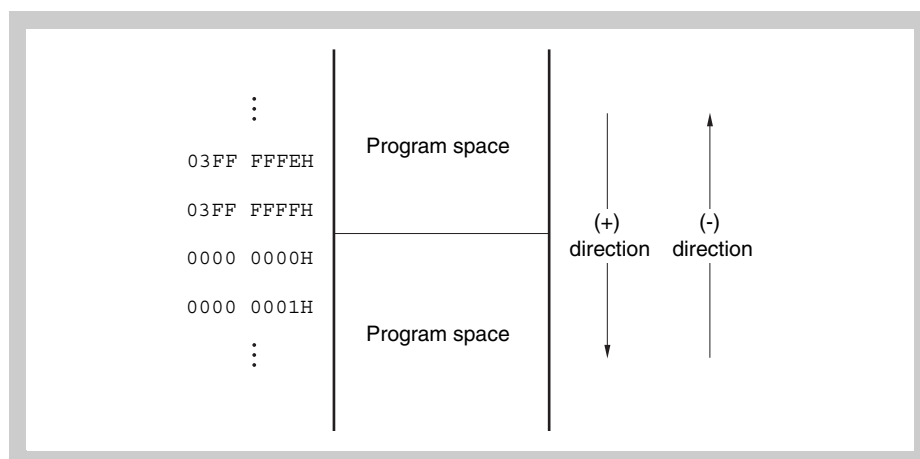


Figure 3-6 Wrap-around of program space

**Caution** No instruction can be fetched from the 4 KB area of 0FFF F000<sub>H</sub> to 0FFF FFFF<sub>H</sub> because this area is defined as peripheral I/O area. Therefore, do not execute any branch to this area.

## 3.5 Memory

In the following sections, the memory of the CPU is introduced. Specific memory areas are described and a recommendation for the usage of the address space is given.

### 3.5.1 Memory areas

The internal memory of the CPU provides several areas:

- Internal flash area for flash memory devices
- Internal RAM area
- Internal fixed peripheral I/O area
- Programmable peripheral I/O area

The areas are briefly described below.

#### (1) Internal flash area

summarizes the size and addresses of the internal flash memories. These cannot be used for access to external memory.

**Table 3-14 Internal flash areas**

Device	Internal flash size	Address range
μPD70F3485	480 KB	0000 0000 <sub>H</sub> to 0007 7FFF <sub>H</sub>

#### (2) Internal RAM area

The 32 KB area between addresses 03FF 7000<sub>H</sub> and 03FF EFFF<sub>H</sub> is provided as the internal RAM area.

**Note** The internal RAM area is mirrored to the area 0FFF 7000<sub>H</sub> to 0FFF EFFF<sub>H</sub>. If data is written in one area, it appears also in the other area.

#### (3) Fixed peripheral I/O area

The 4 KB area between addresses 0FFF F000<sub>H</sub> and 0FFF FFFF<sub>H</sub> is provided as the fixed peripheral I/O area. Accesses to these addresses are passed over to the NPB bus (internal bus).

The following registers are mapped to the peripheral I/O area:

- All registers of peripheral functions
- Registers of timers
- Configuration registers of Interrupt, DMA, Bus and Memory Controller Units
- Configuration registers of the Clock Generator

For a list of all peripheral I/O registers, see “*Special Function Registers*” on page 1049.

- Note**
1. Because the physical address space covers 256 MB, the address bits A[31:28] are not considered. Thus, this address space can also be addressed via the area FFFF F000<sub>H</sub> to FFFF FFFF<sub>H</sub>. This has the advantage that the area can be indirectly addressed by an offset and the zero base r0.  
Therefore, in this manual, all addresses of peripheral I/O registers in the 4 KB peripheral I/O area are given in the range FFFF F000<sub>H</sub> to FFFF FFFF<sub>H</sub> instead of 0FFF F000<sub>H</sub> to 0FFF FFFF<sub>H</sub>.
  2. The *fixed* peripheral I/O area is mirrored to the upper 4 KB of the *programmable* peripheral I/O area PPA—regardless of the base address of the PPA. If data is written to one area, it appears also in the other area.
  3. Program fetches cannot be executed from any peripheral I/O area.
  4. Word registers, that means 32-bit registers, are accessed in two half word accesses. The lower two address bits are ignored.
  5. For registers in which byte access is possible, if half word access is executed:
    - During read operation: The higher 8 bits become undefined.
    - During write operation: The lower 8 bits of data are written to the register.

- Caution**
1. Addresses that are not defined as registers are reserved for future expansion. If these addresses are accessed, the operation is undefined and not guaranteed.
  2. For DMA transfer, the fixed peripheral I/O area (0FFF F000H to 0FFF FFFFH) cannot be specified as the source/destination address in the MARn - DMA transfer memory start address registers. Be sure to use the RAM area (0FFF 7000H to 0FFF EFFFH or 03FF 7000H to 03FF EFFFH) for this source/destination address of DMA transfer.

#### (4) Programmable peripheral I/O area

A 16 KB area is provided as a programmable peripheral I/O area (PPA). The PPA can be freely located. The base address of the programmable peripheral I/O area is specified by the initialization of the peripheral area selection control register (BPC).

See “*Bus and Memory Control (BCU, MEMC)*” on page 192 for details.

## 3.6 RAM ECC error detection

The RAM control circuit is equipped with an error detection function. A 5-bit ECC (Error Correction Coding) code is dedicated to each 8-bit word in the RAM.

The ECC can detect and correct single bit errors and detect double bit errors. In the latter case the double error detection interrupt INTDEDR is generated.

INTDEDR can generate two kinds of interrupts:

- an NMI via the NMI sharing function
- a maskable interrupt (exception code 0080<sub>H</sub>), that is shared with double error detections of the flash memory

Besides generating an non-maskable or maskable interrupt, the address of the erroneous data and the location of the erroneous data byte of the 32-bit data are saved in registers.

**Note** A RAM access is always performed with 32-bit, i.e. 4-byte, data. Thus a double error detection interrupt INTDEDR is also generated upon an 1-byte or 2-byte access, even if the read target byte(s) are not erroneous, but any other data byte that is read concurrently.

### 3.6.1 RAM ECC initialization

The RAM content and the 5-bit ECC values are undefined after device power-up, the 5-bit ECC values do not fit to their respective 8-bit data. Thus the entire RAM must be initialized, i.e. written, in order to fit the ECC values to the data. Since a double error detection interrupt INTDEDR may have been issued due to undefined data and ECC values after power-up, follow below procedure to start up the RAM ECC:

1. After device power-up NMI sharing is disabled (INTSEL.ISR = 0) and the maskable interrupt is masked (ERRIC.ERRMK = 1). Don't enable NMI sharing respectively unmask the maskable interrupt before the RAM has been completely initialized.
2. Initialize the entire RAM by writing to all RAM locations. The data written is not of concern.
3. Enable NMI sharing by INTSEL.ISR = 1 or unmask the maskable interrupt by ERRIC.ERRMK = 0, whatever is desired.

Also when program code shall be executed from RAM, it is recommended to fill the entire RAM, preferably all data bytes with the NOP instruction code, before fetching the first instruction from RAM. This avoids ECC errors generated by instruction prefetches by the execution pipeline.

### 3.6.2 RAM ECC registers

The RAM ECC is operated by means of the following registers:

**Table 3-15 RAM ECC registers overview**

Register name	Shortcut	Address
RAM ECC error address register	RAMEAD	FFFF F8B0 <sub>H</sub>
RAM ECC error data location register	RAMEDLR	FFFF F8B2 <sub>H</sub>
RAM ECC control register	RAMECC	FFFF F8B4 <sub>H</sub>

#### (1) RAMEAD - RAM ECC error address register

The 16-bit RAMEAD register holds the lower 16 bit of the address of the RAM location, where an error was detected first.

**Access** This register can be read in 16-bit units.

**Address** FFFF F8B0<sub>H</sub>

**Initial Value** 0000<sub>H</sub>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAMEADDR														0	0

**Table 3-16 RAMEAD register contents**

Bit Position	Bit Name	Function
15 to 2	RAMEADDR	16-bit address of first detection of a RAM error.

The address of the first error detection is stored in RAMEAD. At the same time the interrupt flag INTERRF.INTERR1 is set to 1. Additionally the erroneous byte location is indicated in the RAMEDLR register.

This register is not overwritten by a new error detection until

- the INTERRF.INTERR1 is cleared to 0. Thus no new error is signaled by interrupt INTDEDR until the INTERRF.INTERR1 is cleared to 0.
- RAMEAD is cleared to 0000<sub>H</sub> by any reset.

Afterwards the lower 16 bit of a new ECC detection address can be stored and a new INTDEDR interrupt can be generated.



**(2) RAMEDLR - RAM ECC error data location register**

RAMEDLR indicates which byte of the 32-bit data byte at address RAMEAD has generated an error.

**Access** This register can be read in 8-bit units.

**Address** FFFF F8B2<sub>H</sub>

**Initial Value** 00<sub>H</sub>

7	6	5	4	3	2	1	0
0	0	0	0	REBEN3	REBEN2	REBEN1	REBEN0
R	R	R	R	R	R	R	R

**Table 3-17 RAMEDLR register contents**

Bit Position	Bit Name	Function
3	REBEN3	ECC error status in data bits 31 to 24: 0: ECC error did not occur in bits 31 to 24 1: ECC error occurred in bits 31 to 24
2	REBEN2	ECC error status in data bits 23 to 16: 0: ECC error did not occur in bits 23 to 16 1: ECC error occurred in bits 23 to 16
1	REBEN1	ECC error status in data bits 15 to 8: 0: ECC error did not occur in bits 15 to 8 1: ECC error occurred in bits 15 to 8
0	REBEN0	ECC error status in data bits 7 to 0: 0: ECC error did not occur in bits 7 to 0 1: ECC error occurred in bits 7 to 0

The content of RAMEDLR is related to the error detection address in RAMEAD, thus only valid after an error detection interrupt INTDEDR, i.e. if INTERRF.INTERR1 = 1.

**(3) RAMECC - RAM ECC control register**

RAMECC control the operation of the RAM ECC function.

Writing to this register is only possible immediately after writing to the associated write protection register.

First write to the PRCMD register. The contents is ignored. Then, you are permitted to write once to the RAMECC register. This must be done immediately after writing to the PRCMD register. After the second write action, or if the second write action does not follow immediately, all protected registers are write-locked again.

**Access** This register can be read/written in 8-bit and 1-bit units.

**Address** FFFF F8B4<sub>H</sub>

**Initial Value** 00<sub>H</sub>

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	RAEENB
R	R	R	R	R	R	R	R

**Table 3-18 RAMECC register contents**

Bit Position	Bit Name	Function
0	RAEENB	ECC function enable: 0: ECC function enabled 1: ECC function disabled

- Caution**
1. Before modifying RAEENB make sure to stop any DMA transfers.
  2. Confirm correct setting of RAEENB by re-reading this bit after its modification.

### 3.6.3 RAM ECC function check

In order to check the functionality of the RAM ECC error detection proceed as described below:

1. enable ECC function by `RAMECC.RAEENB = 0`
2. write some data word (e.g. `data_1`) to the RAM: correct ECC will be generated
3. disable ECC function by `RAMECC.RAEENB = 1`
4. write another data word (e.g. `data_2`) to the RAM address of the first data word `data_1`: no new ECC will be generated, thus ECC does not fit to `data_2`
5. enable ECC function by `RAMECC.RAEENB = 0`
6. read `data_2`: a maskable or non-maskable ECC error interrupt should be generated

### 3.6.4 Recommended use of data address space

When accessing operand data in the data space, one register has to be used for address generation. This register is called pointer register. With relative addressing, an instruction can access operand data at all addresses that lie in the range of  $\pm 32$  KB relative to the address in the pointer register.

By this offset addressing method load/store instructions can be accommodated in a single 32-bit instruction word, resulting in faster program execution and smaller code size.

To enhance the efficiency of using the pointer in consideration of the memory map, the following is recommended:

For efficient use of the relative addressing feature, the data segments should be located in the address range FFFF 8000<sub>H</sub> to FFFF EFFF<sub>H</sub> and 0000 0000<sub>H</sub> to 0000 7FFF<sub>H</sub>. The peripheral I/O registers and the internal RAM is aligned to the upper bound, thus the registers and a part of the RAM can be addressed via relative addressing, with base address 0 (r0).

It is recommended to locate flash memory data segments in the area up to 0000 7FFF<sub>H</sub>, so access to these constant data can utilize also relative addressing.

Use the r0 register as pointer register for operand addressing. Since the r0 register is fixed to zero by hardware, it can be used as a pointer register and, at the same time, for any other purposes, where a zero register is required. Thus, no other general purpose register has to be reserved as pointer register.

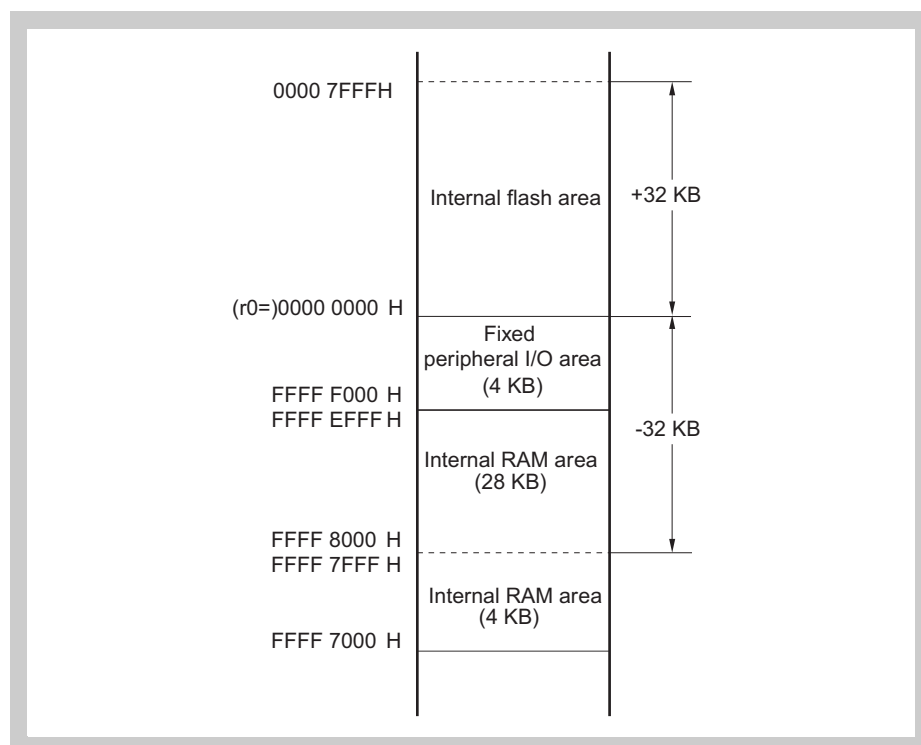


Figure 3-7 Example application of wrap-around

## 3.7 Write Protected Registers

Write protected registers are protected from inadvertent write access due to erroneous program execution, etc. Write access to a write protected register is only given immediately after writing to a corresponding write enable register. For a write access to the write protected registers you have to use the following instructions:

1. Store instruction (ST/SST instruction)
2. Bit operation instruction (SET1/CLR1/NOT1 instruction)

When *reading* write protected registers, no special sequence is required.

The following table gives an overview of the write protected registers and their corresponding write enable registers.

For some registers, incorrect store operations can be checked by a flag of the corresponding status register. This is also marked in the table below.

**Table 3-19 Overview of write protected registers**

Write protected register	Shortcut	Corresponding write enable register	Shortcut	For details see
Port group control registers of port group 5	P5, PM5, PMC5, PESC5, ESOST5, PESMK5	Command register	PRCMD	<i>"Pin Functions" on page 23</i>
Port group control registers of port group 6	P6, PM6, PMC6, PESC6, ESOST6, PESMK6			
RAM ECC control register	RAMECC	Command register	PRCMD	<i>"RAM ECC error detection" on page 91</i>
Clock Monitor mode register	CLM	Command register	PRCMD	<i>"Clock Generator" on page 148</i>
Data flash control register	DFLCTL	Command register	PRCMD	<i>"Flash Memory" on page 159</i>

**Example** Write to port register P5:

```
PRCMD = 0x5A;
P5 = 0x80;
```

**Note** Make sure that the compiler generates two consecutive assembler "store" instructions to PRCMD and the protected register (P5 in the above example) from the associated C statements. Ensure to check the bit PHS.PRERR to be "0", as this indicates a successful write to the protected register.

Since any action between writing to a write enable register and writing to a protected register destroys this sequence, the effects of interrupts and DMA transfers have to be considered:

- **Interrupts:**  
After writing to PRCMD no maskable interrupts will be acknowledged, until the subsequent instruction has been executed. Thus a maskable interrupt can not destroy the sequence.  
However, any non-maskable interrupt can still be acknowledged.

- **DMA:**  
In the above example, DMA transfers can still take place. They may destroy the sequence.  
  
If appropriate, you may disable DMA transfers in advance. Otherwise you must check whether writing to the protected register was successful. To do so, check the status via the status register, if available, or by reading back the protected register.

## Chapter 4 Interrupt Controller (INTC)

This controller is provided with a dedicated Interrupt Controller (INTC) for interrupt servicing and can process a large amount of maskable and up to five non-maskable 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. Generally, an exception takes precedence over an interrupt.

This controller 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).

Eight levels of software-programmable priorities can be specified for each interrupt request. Starting of interrupt servicing takes no fewer than 4 system clocks after the generation of an interrupt request.

### 4.1 Features

- Interrupts
  - Non-maskable interrupts: up to 4 sources  
(depending on whether interrupt sharing is enabled or not)
  - Maskable interrupts:
    - internal peripherals: 98 sources
    - external: 13 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, level detection for external interrupt request signals
- Exceptions
  - Software exceptions: 2 channels with each 16 sources
  - Exception traps: 1 source (illegal opcode exception)

Table 4-1 Interrupt/exception source list (1/5)

Type	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
	Name	Control Register	Generating Source	Generating Unit				
Reset	RESET	–	RESET input	Pin	–	0000H	00000000H	undef.
Non-maskable	NMI	–	P00 NMI Input	Pin	–	0010H	00000010H	nextPC
	INTDEDF <sup>a</sup>	–	Double error detection	Flash memory				
	INTDEDR <sup>a</sup>	–	Double error detection	RAM				
	INTOSD <sup>a</sup>	–	Oscillator stop detection	Clock Monitor				
Software exception	TRAP0n (n = 0 to F <sub>H</sub> )	–	TRAP instruction	–	–	004nH (n = 0 to F <sub>H</sub> )	00000040H	nextPC
	TRAP1n (n = 0 to F <sub>H</sub> )	–	TRAP instruction	–	–	005nH (n = 0 to F <sub>H</sub> )	00000050H	nextPC
Exception trap	ILGOP	–	Illegal opcode	–	–	0060H	00000060H	nextPC
	DBTRAP	–	DBTRAP instruction	–				
Maskable	INTDEDF <sup>a</sup>	ERRIC	Double error detection	Flash memory	0	0080H	00000080H	nextPC
	INTDEDR <sup>a</sup>		Double error detection	RAM				
	INTOSD <sup>a</sup>	OSDIC	Oscillator stop detection	Clock Monitor	1	0090H	00000090H	nextPC
	INTP0	PIC0	External interrupt 0	Pin	2	00A0H	000000A0H	next PC
	INTP1	PIC1	External interrupt 1	Pin	3	00B0H	000000B0H	next PC
	INTP2	PIC2	External interrupt 2	Pin	4	00C0H	000000C0H	next PC
	INTP3	PIC3	External interrupt 3	Pin	5	00D0H	000000D0H	next PC
	INTUC0R <sup>b</sup>	PIC4	UARTC0 receive complete interrupt	UARTC0 (DMAC) <sup>c</sup>	6	00E0H	000000E0H	next PC
	INTP4 <sup>b</sup>		External interrupt 4	Pin				
	INTUC1R <sup>b</sup>	PIC5	UARTC1 receive complete interrupt	UARTC1 (DMAC) <sup>c</sup>	7	00F0H	000000F0H	next PC
	INTP5 <sup>b</sup>		External interrupt 5	Pin				
	INTP6	PIC6	External interrupt 6	Pin	8	0100H	00000100H	next PC
	INTP7	PIC7	External interrupt 7	Pin	9	0110H	00000110H	next PC
	INTP8	PIC8	External interrupt 8	Pin	10	0120H	00000120H	next PC
	INTP9	PIC9	External interrupt 9	Pin	11	0130H	00000130H	next PC
	INTP10	PIC10	External interrupt 10	Pin	12	0140H	00000140H	next PC
	INTP11	PIC11	External interrupt 11	Pin	13	0150H	00000150H	next PC
	INTP12	PIC12	External interrupt 12	Pin	14	0160H	00000160H	next PC



Table 4-1 Interrupt/exception source list (2/5)

Type	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
	Name	Control Register	Generating Source	Generating Unit				
Maskable	INTTS0OV	TS0OVIC	TMS0 overflow interrupt	TMS0	15	0170H	00000170H	next PC
	INTTS0CC0	TS0CCIC0	TMS0 capture compare channel 0	TMS0	16	0180H	00000180H	next PC
	INTTS0CC1	TS0CCIC1	TMS0 capture compare channel 1	TMS0	17	0190H	00000190H	next PC
	INTTS0CC2	TS0CCIC2	TMS0 capture compare channel 2	TMS0	18	01A0H	000001A0H	next PC
	INTTS0CC3	TS0CCIC3	TMS0 capture compare channel 3	TMS0	19	01B0H	000001B0H	next PC
	INTTS0CC4	TS0CCIC4	TMS0 capture compare channel 4	TMS0	20	01C0H	000001C0H	next PC
	INTTS0CC5	TS0CCIC5	TMS0 capture compare channel 5	TMS0	21	01D0H	000001D0H	next PC
	INTTS0CD0	TS0CDIC0	TS0CNT top reversal	TMS0	22	01E0H	000001E0H	next PC
	INTTS0OD	TS0ODIC	TS0CNT bottom reversal	TMS0	23	01F0H	000001F0H	next PC
	INTTS0ER	TS0ERIC	TMS0 error detection	TMS0	24	0200H	00000200H	next PC
	INTTS0WN	TS0WNIC	TMS0 warning interrupt	TMS0	25	0210H	00000210H	next PC
	INTTS1OV	TS1OVIC	TMS1 overflow interrupt	TMS1	26	0220H	00000220H	next PC
	INTTS1CC0	TS1CCIC0	TMS1 capture compare channel 0	TMS1	27	0230H	00000230H	next PC
	INTTS1CC1	TS1CCIC1	TMS1 capture compare channel 1	TMS1	28	0240H	00000240H	next PC
	INTTS1CC2	TS1CCIC2	TMS1 capture compare channel 2	TMS1	29	0250H	00000250H	next PC
	INTTS1CC3	TS1CCIC3	TMS1 capture compare channel 3	TMS1	30	0260H	00000260H	next PC
	INTTS1CC4	TS1CCIC4	TMS1 capture compare channel 4	TMS1	31	0270H	00000270H	next PC
	INTTS1CC5	TS1CCIC5	TMS1 capture compare channel 5	TMS1	32	0280H	00000280H	next PC
	INTTS1CD0	TS1CDIC0	TS1CNT top reversal	TMS1	33	0290H	00000290H	next PC
	INTTS1OD	TS1ODIC	TS1CNT bottom reversal	TMS1	34	02A0H	000002A0H	next PC
	INTTS1ER	TS1ERIC	TMS1 error detection	TMS1	35	02B0H	000002B0H	next PC
	INTTS1WN	TS1WNIC	TMS1 warning interrupt	TMS1	36	02C0H	000002C0H	next PC
	INTTT0OV	TT0OVIC	TMT0 overflow interrupt	TMT0	37	02D0H	000002D0H	nextPC
	INTTT0CC0	TT0CCIC0	TMT0 capture compare channel 0	TMT0	38	02E0H	000002E0H	nextPC
	INTTT0CC1	TT0CCIC1	TMT0 capture compare channel 1	TMT0	39	02F0H	000002F0H	nextPC
	INTTT0EC	TT0ECIC	TMT0 encoder clear interrupt	TMT0	40	0300H	00000300H	nextPC

Table 4-1 Interrupt/exception source list (3/5)

Type	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
	Name	Control Register	Generating Source	Generating Unit				
Maskable	INTTT1OV	TT1OVIC	TMT1 overflow interrupt	TMT1	41	0310H	00000310H	nextPC
	INTTT1CC0	TT1CCIC0	TMT1 capture compare channel 0	TMT1	42	0320H	00000320H	nextPC
	INTTT1CC1	TT1CCIC1	TMT1 capture compare channel 1	TMT1	43	0330H	00000330H	nextPC
	INTTT1EC	TT1ECIC	TMT1 encoder clear interrupt	TMT1	44	0340H	00000340H	nextPC
	INTTAA0OV	TAA0OVIC	TAA0 overflow interrupt	TAA0	45	0350H	00000350H	nextPC
	INTTAA0CC0	TAA0CCIC0	TAA0 capture compare channel 0	TAA0	46	0360H	00000360H	nextPC
	INTTAA0CC1	TAA0CCIC1	TAA0 capture compare channel 1	TAA0	47	0370H	00000370H	nextPC
	INTTAA1OV	TAA1OVIC	TAA1 overflow interrupt	TAA1	48	0380H	00000380H	nextPC
	INTTAA1CC0	TAA1CCIC0	TAA1 capture compare channel 0	TAA1	49	0390H	00000390H	nextPC
	INTTAA1CC1	TAA1CCIC1	TAA1 capture compare channel 1	TAA1	50	03A0H	000003A0H	nextPC
	INTTAA2OV	TAA2OVIC	TAA2 overflow interrupt	TAA2	51	03B0H	000003B0H	nextPC
	INTTAA2CC0	TAA2CCIC0	TAA2 capture compare channel 0	TAA2	52	03C0H	000003C0H	nextPC
	INTTAA2CC1	TAA2CCIC1	TAA2 capture compare channel 1	TAA2	53	03D0H	000003D0H	nextPC
	INTTAA3OV	TAA3OVIC	TAA3 overflow interrupt	TAA3	54	03E0H	000003E0H	nextPC
	INTTAA3CC0	TAA3CCIC0	TAA3 capture compare channel 0	TAA3	55	03F0H	000003F0H	nextPC
	INTTAA3CC1	TAA3CCIC1	TAA3 capture compare channel 1	TAA3	56	0400H	00000400H	nextPC
	INTTAA4OV	TAA4OVIC	TAA4 overflow interrupt	TAA4	57	0410H	00000410H	nextPC
	INTTAA4CC0	TAA4CCIC0	TAA4 capture compare channel 0	TAA4	58	0420H	00000420H	nextPC
	INTTAA4CC1	TAA4CCIC1	TAA4 capture compare channel 1	TAA4	59	0430H	00000430H	nextPC
	INTTAA5OV	TAA5OVIC	TAA5 overflow interrupt	TAA5	60	0440H	00000440H	nextPC
	INTTAA5CC0	TAA5CCIC0	TAA5 capture compare channel 0	TAA5	61	0450H	00000450H	nextPC
	INTTAA5CC1	TAA5CCIC1	TAA5 capture compare channel 1	TAA5	62	0460H	00000460H	nextPC
	INTTAA6OV	TAA6OVIC	TAA6 overflow interrupt	TAA6	63	0470H	00000470H	nextPC
	INTTAA6CC0	TAA6CCIC0	TAA6 capture compare channel 0	TAA6	64	0480H	00000480H	nextPC
	INTTAA6CC1	TAA6CCIC1	TAA6 capture compare channel 1	TAA6	65	0490H	00000490H	nextPC

Table 4-1 Interrupt/exception source list (4/5)

Type	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
	Name	Control Register	Generating Source	Generating Unit				
Maskable	INTTAA7OV	TAA7OVIC	TAA7 overflow interrupt	TAA7	66	04A0H	000004A0H	nextPC
	INTTAA7CC0	TAA7CCIC0	TAA7 capture compare channel 0	TAA7	67	04B0H	000004B0H	nextPC
	INTTAA7CC1	TAA7CCIC1	TAA7 capture compare channel 1	TAA7	68	04C0H	000004C0H	nextPC
	INTTAA8OV	TAA8OVIC	TAA8 overflow interrupt	TAA8	69	04D0H	000004D0H	nextPC
	INTTAA8CC0	TAA8CCIC0	TAA8 capture compare channel 0	TAA8	70	04E0H	000004E0H	nextPC
	INTTAA8CC1	TAA8CCIC1	TAA8 capture compare channel 1	TAA8	71	04F0H	000004F0H	nextPC
	INTTAA9OV	TAA9OVIC	TAA9 overflow interrupt	TAA9	72	0500H	00000500H	nextPC
	INTTAA9CC0	TAA9CCIC0	TAA9 capture compare channel 0	TAA9	73	0510H	00000510H	nextPC
	INTTAA9CC1	TAA9CCIC1	TAA9 capture compare channel 1	TAA9	74	0520H	00000520H	nextPC
	INTBRG0 <sup>b</sup>	BRGIC	BRG0 match	BRG0	75	0530H	00000530H	nextPC
	INTBRG1 <sup>b</sup>		BRG1 match	BRG1				
	INTBRG2	BRGIC2	BRG2 match	BRG2	76	0540H	00000540H	nextPC
	INTC0ERR	C0ERRIC	CAN0 error interrupt	CAN0	83	05B0H	000005B0H	nextPC
	INTC0WUP	C0WUPIC	CAN0 wake up interrupt	CAN0	84	05C0H	000005C0H	nextPC
	INTC0REC	C0RECIC	CAN0 receive interrupt	CAN0	85	05D0H	000005D0H	nextPC
	INTC0TRX	C0TRXIC	CAN0 transmit interrupt	CAN0	86	05E0H	000005E0H	nextPC
	INTC1ERR	C1ERRIC	CAN1 error interrupt	CAN1	87	05F0H	000005F0H	nextPC
	INTC1WUP	C1WUPIC	CAN1 wake up interrupt	CAN1	88	0600H	00000600H	nextPC
	INTC1REC	C1RECIC	CAN1 receive interrupt	CAN1	89	0610H	00000610H	nextPC
	INTC1TRX	C1TRXIC	CAN1 transmit interrupt	CAN1	90	0620H	00000620H	nextPC
	Reserved	–	Reserved	–	91	0630H	00000630H	nextPC
	Reserved	–	Reserved	–	92	0640H	00000640H	nextPC
	Reserved	–	Reserved	–	93	0650H	00000650H	nextPC
	Reserved	–	Reserved	–	94	0660H	00000660H	nextPC
	Reserved	–	Reserved	–	95	0670H	00000670H	nextPC
	Reserved	–	Reserved	–	96	0680H	00000680H	nextPC
	Reserved	–	Reserved	–	97	0690H	00000690H	nextPC
	Reserved	–	Reserved	–	98	06A0H	000006A0H	nextPC
	INTCB0T	CB0TIC	CSIB0 transmit interrupt	CSIB0 (DMAC) <sup>c</sup>	99	06B0H	000006B0H	nextPC
	INTCB0R	CB0RIC	CSIB0 receive complete interrupt	CSIB0 (DMAC) <sup>c</sup>	100	06C0H	000006C0H	nextPC
	INTCB0RE	CB0REIC	CSIB0 receive error interrupt	CSIB0	101	06D0H	000006D0H	nextPC

**Table 4-1 Interrupt/exception source list (5/5)**

Type	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
	Name	Control Register	Generating Source	Generating Unit				
Maskable	INTCB1T	CB1TIC	CSIB1 transmit interrupt	CSIB1 (DMAC) <sup>c</sup>	102	06E0H	000006E0H	nextPC
	INTCB1R	CB1RIC	CSIB1 receive complete interrupt	CSIB1 (DMAC) <sup>c</sup>	103	06F0H	000006F0H	nextPC
	INTCB1RE	CB1REIC	CSIB1 receive error interrupt	CSIB1	104	0700H	00000700H	nextPC
	INTCE0OF	CE0OFIC	CSIE0 overflow interrupt	CSIE0	105	0710H	00000710H	nextPC
	INTCE0C	CE0CIC	CSIE0 transfer end interrupt	CSIE0 (DMAC) <sup>c</sup>	106	0720H	00000720H	nextPC
	INTCE1OF	CE1OFIC	CSIE1 overflow interrupt	CSIE1	107	0730H	00000730H	nextPC
	INTCE1C	CE1CIC	CSIE1 transfer end interrupt	CSIE1 (DMAC) <sup>c</sup>	108	0740H	00000740H	nextPC
	INTUC0RE	UC0REIC	UARTC0 receive error interrupt	UARTC0	109	0750H	00000750H	nextPC
	INTUC0T	UC0TIC	UARTC0 transmit interrupt	UARTC0 (DMAC) <sup>c</sup>	110	0760H	00000760H	nextPC
	INTUC1RE	UC1REIC	UARTC1 receive error interrupt	UARTC1	111	0770H	00000770H	nextPC
	INTUC1T	UC1TIC	UARTC1 transmit interrupt	UARTC1 (DMAC) <sup>c</sup>	112	0780H	00000780H	nextPC
	INTAD0	ADIC0	ADC0 end of conversion	ADC0 (DMAC) <sup>c</sup>	113	0790H	00000790H	nextPC
	INTAD1	ADIC1	ADC1 end of conversion	ADC1 (DMAC) <sup>c</sup>	114	07A0H	000007A0H	nextPC
	INTDMA2	DMAIC2	DMA2 transmission end	DMAC	115	07B0H	000007B0H	nextPC
	INTDMA3	DMAIC3	DMA3 transmission end	DMAC	116	07C0H	000007C0H	nextPC

- a) Depending on the setting of INTSEL.ISR, this interrupt source generates an NMI or a maskable interrupt. For more information see *“Interrupt Sharing” on page 127*.
- b) This is a shared interrupt. For details see *“Interrupt Sharing” on page 127*.
- c) If this interrupt is defined as a DMA trigger, it is generated by the corresponding DMA completion interrupt INTDMA<sub>n</sub> instead of the original interrupt source. For further details refer to *“DMA Functions (DMA Controller)” on page 211*

- Note**
1. Default priority: The priority order when two or more maskable interrupt requests are generated at the same time. The highest priority is 0.
  2. Restored PC: The value of the PC saved to EIPC or FEPC when interrupt/exception processing is started. However, the value of the PC saved when an interrupt is acknowledged during division (DIV, DIVH, DIVU, DIVHU) instruction execution is the value of the PC of the current instruction (DIV, DIVH, DIVU, DIVHU).
  3. nextPC: The PC value that starts the processing following interrupt/exception processing.
  4. The execution address of the illegal instruction when an illegal opcode exception occurs is calculated by (Restored PC – 4).

## 4.2 Non-Maskable Interrupts

A non-maskable interrupt request is acknowledged unconditionally, even when interrupts are in the interrupt disabled (DI) status.

Non-maskable interrupts of this microcontroller are available for the following requests:

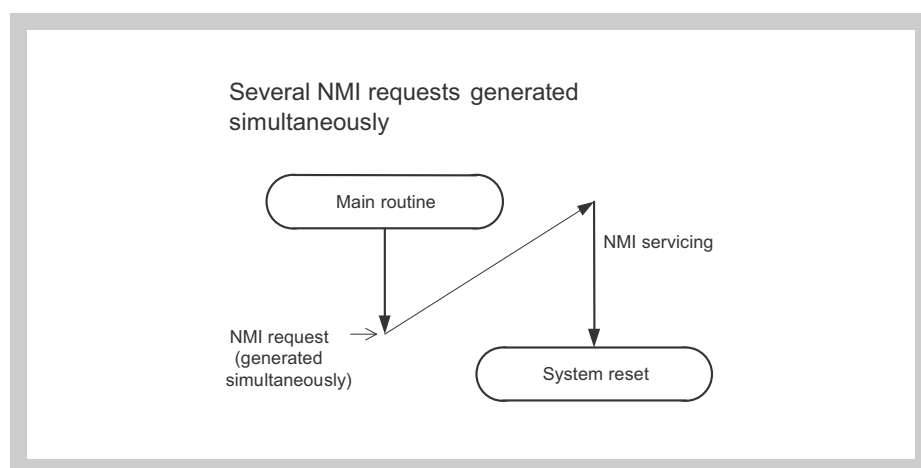
**Table 4-2 Non-maskable interrupts**

Interrupt Source	Generating Unit	Comment
NMI	Pin	When the valid edge specified by bits INTM0.ESN[1:0] is detected on the NMI pin, the interrupt occurs.
INTDEDF	Flash memory	These interrupt sources only generate a <i>non-maskable</i> interrupt if interrupt sharing is enabled for non-maskable interrupts. See “Interrupt Sharing” on page 127.
INTDEDR	RAM	
INTOSD	Clock Monitor	

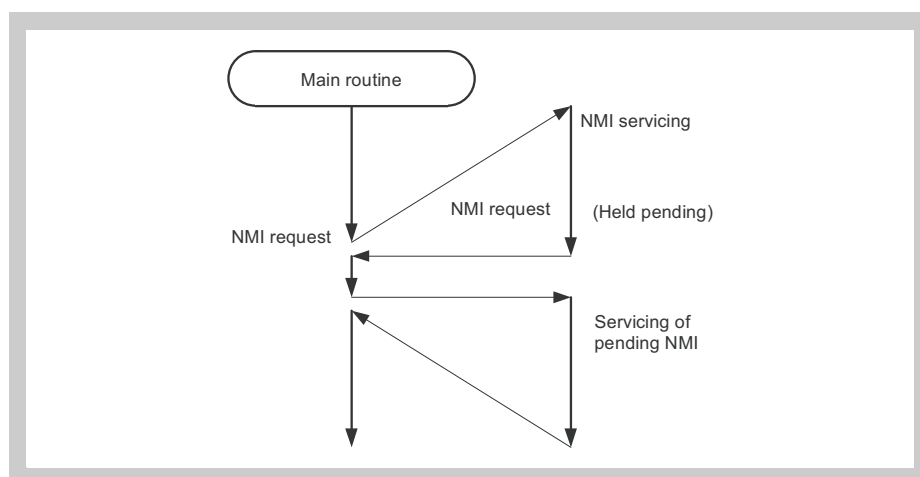
**Note** If a NMI is generated while NMI is being serviced, the service is executed as follows:

The new NMI request is held pending regardless of the value of the PSW.NP bit. The pending NMIVC request is acknowledged after servicing of the current NMI request has finished (after execution of the RETI instruction).

**Caution** If PSW.NP is cleared by the LDSR instruction while a non-maskable interrupt is being serviced, a following NMI interrupt cannot be acknowledged correctly.



**Figure 4-1 Example of non-maskable interrupt request acknowledgement operation: multiple NMI requests generated at the same time**



**Figure 4-2** Example of non-maskable interrupt request acknowledgement operation: NMI request generated during NMI servicing

### 4.2.1 Operation

If a non-maskable interrupt is generated, the CPU performs the following processing, and transfers control to the handler routine:

1. Saves the restored PC to FEPC.
2. Saves the current PSW to FEPSW.
3. Writes exception code 0010H to the higher halfword (FECC) of ECR.
4. Sets the NP and ID bits of the PSW and clears the EP bit.
5. Sets the handler address corresponding to the non-maskable interrupt to the PC, and transfers control.

The processing configuration of a non-maskable interrupt is shown in *Figure 4-3*.

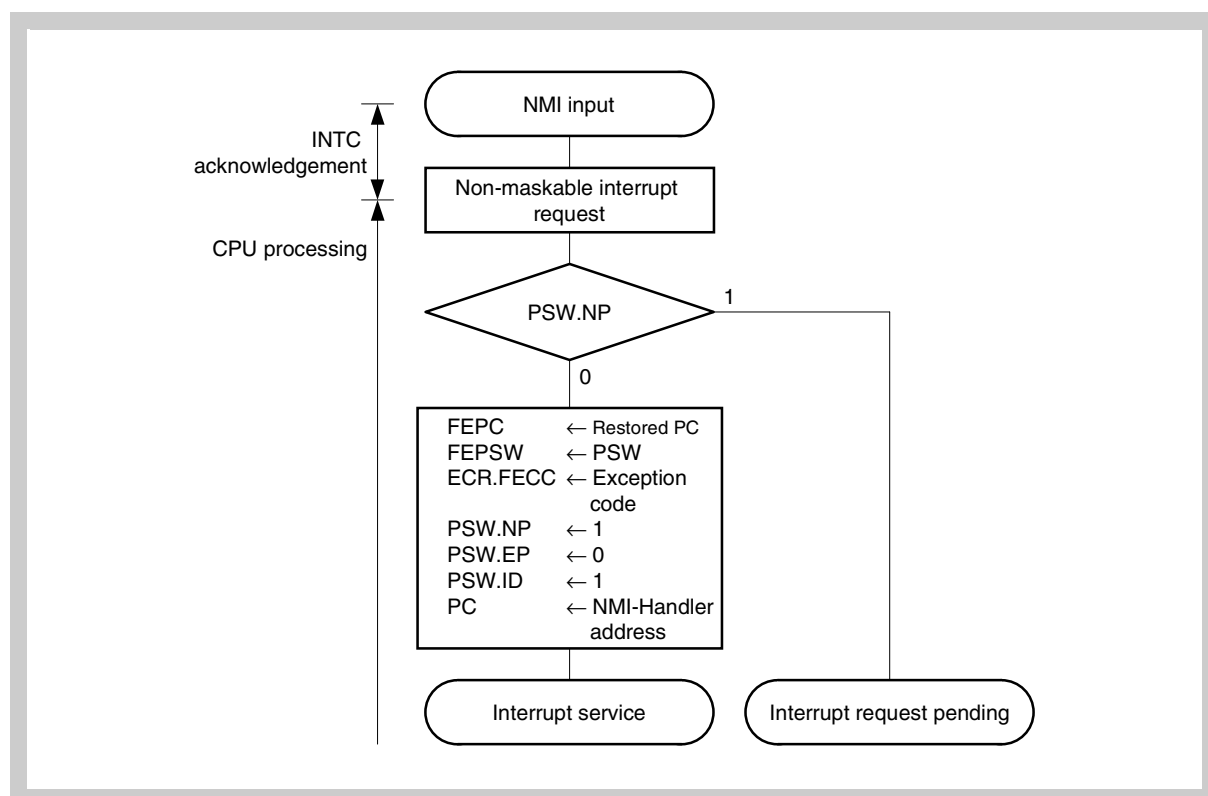


Figure 4-3 Processing configuration of non-maskable interrupt

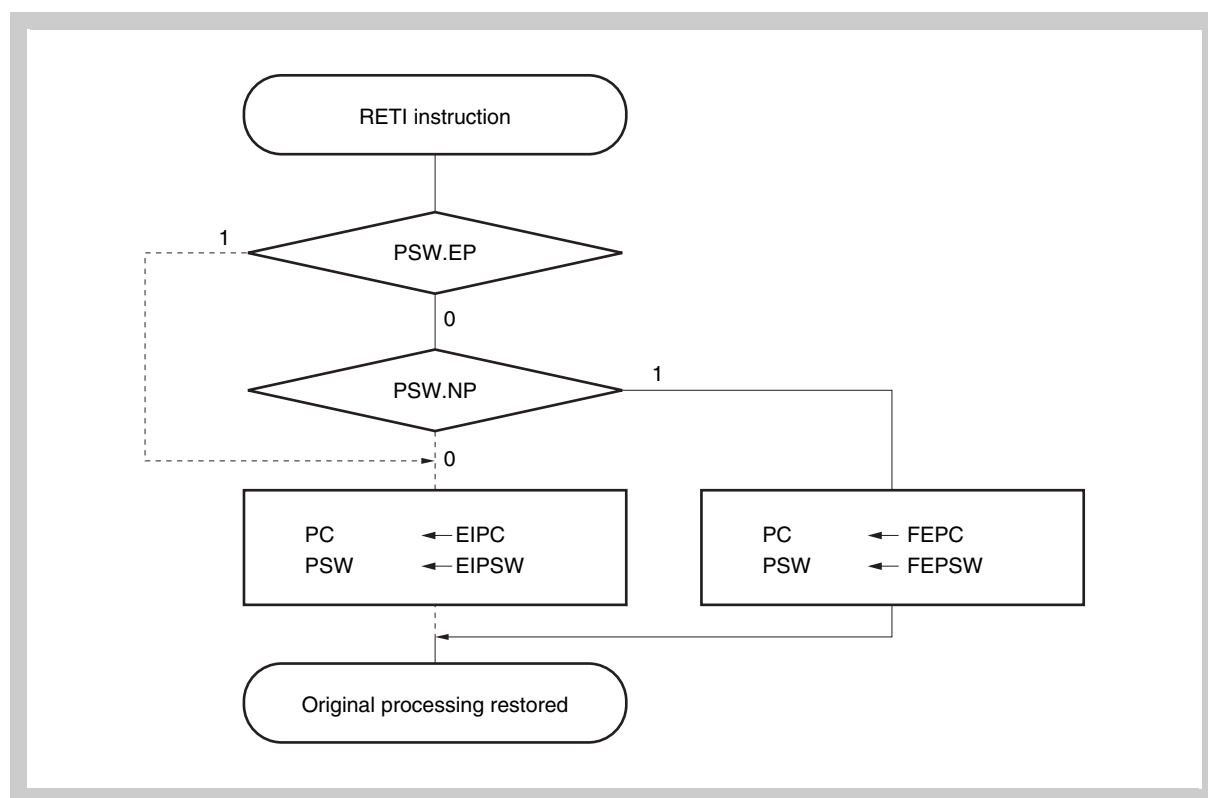
### 4.2.2 Restore

Execution is restored from the non-maskable interrupt (NMI) 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. Restores the values of the PC and the PSW from FEPC and FEPSW, respectively, because the EP bit of the PSW is 0 and the NP bit of the PSW is 1.
2. Transfers control back to the address of the restored PC and PSW.

Figure 4-4 illustrates how the RETI instruction is processed.



**Figure 4-4** RETI instruction processing

**Caution** When the PSW.EP bit and PSW.NP bit are changed by the LDSR instruction during non-maskable interrupt processing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 0 and PSW.NP back to 1 using the LDSR instruction immediately before the RETI instruction.

**Note** The solid line indicates the CPU processing flow.





## 4.3 Maskable Interrupts

Maskable interrupt requests can be masked by interrupt control registers.

If two or more maskable interrupt requests 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 has been acknowledged, the acknowledgement of other maskable interrupt requests is disabled and the interrupt disabled (DI) status is set.

When the EI instruction is executed in an interrupt processing routine, the interrupt enabled (EI) status is set, which enables servicing of interrupts having a higher priority than the interrupt request in progress (specified by the interrupt control register). Note that only interrupts with a higher priority will have this capability; interrupts with the same priority level cannot be nested.

However, if multiple interrupts are executed, the following processing is necessary.

1. Save EIPC and EIPSW in memory or a general-purpose register before executing the EI instruction.
2. Execute the DI instruction before executing the RETI instruction, then reset EIPC and EIPSW with the values saved in (1).

### 4.3.1 Operation

If a maskable interrupt occurs by INT input, the CPU performs the following processing, and transfers control to a handler routine:

1. Saves the restored PC to EIPC.
2. Saves the current PSW to EIPSW.
3. Writes an exception code to the lower halfword of ECR (EICC).
4. Sets the ID bit of the PSW and clears the EP bit.
5. Sets the handler address corresponding to each interrupt to the PC, and transfers control.

The processing configuration of a maskable interrupt is shown in *Figure 4-5*.

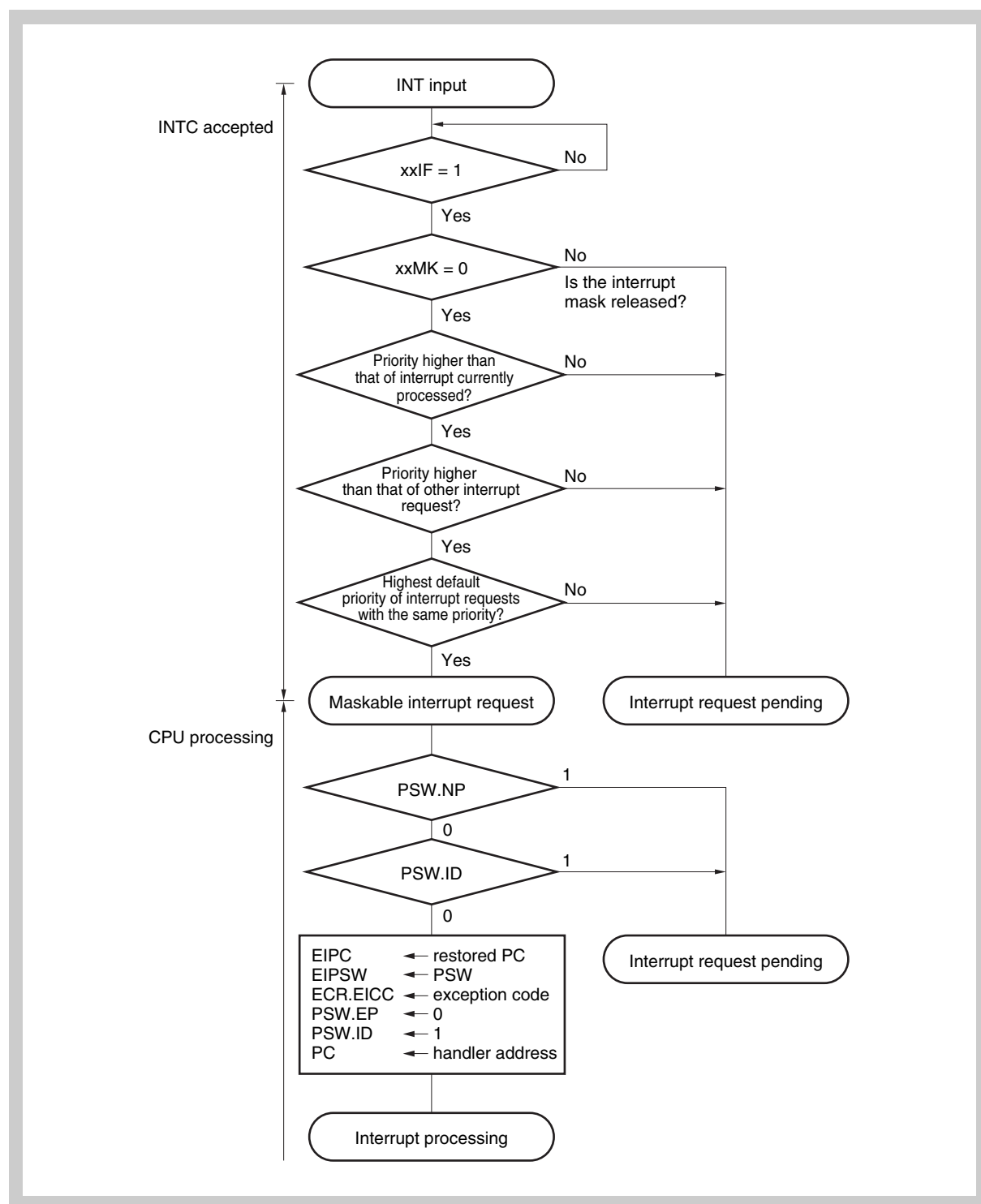


Figure 4-5 Maskable interrupt processing

**Note** For the ISPR register, see “ISPR - In-service priority register” on page 122.

An INT input masked by the Interrupt Controllers and an INT input that occurs while another interrupt is being processed (when PSW.NP = 1 or PSW.ID = 1) are held pending internally by the Interrupt Controller. In such case, if the interrupts are unmasked, or when PSW.NP = 0 and PSW.ID = 0 as set by the RETI and LDSR instructions, input of the pending INT starts the new maskable interrupt processing.

### 4.3.2 Restore

Recovery from maskable interrupt processing 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. Restores the values of the PC and the PSW from EIPC and EIPSW because the EP bit of the PSW is 0 and the NP bit of the PSW is 0.
2. Transfers control to the address of the restored PC and PSW.

Figure 4-6 illustrates the processing of the RETI instruction.

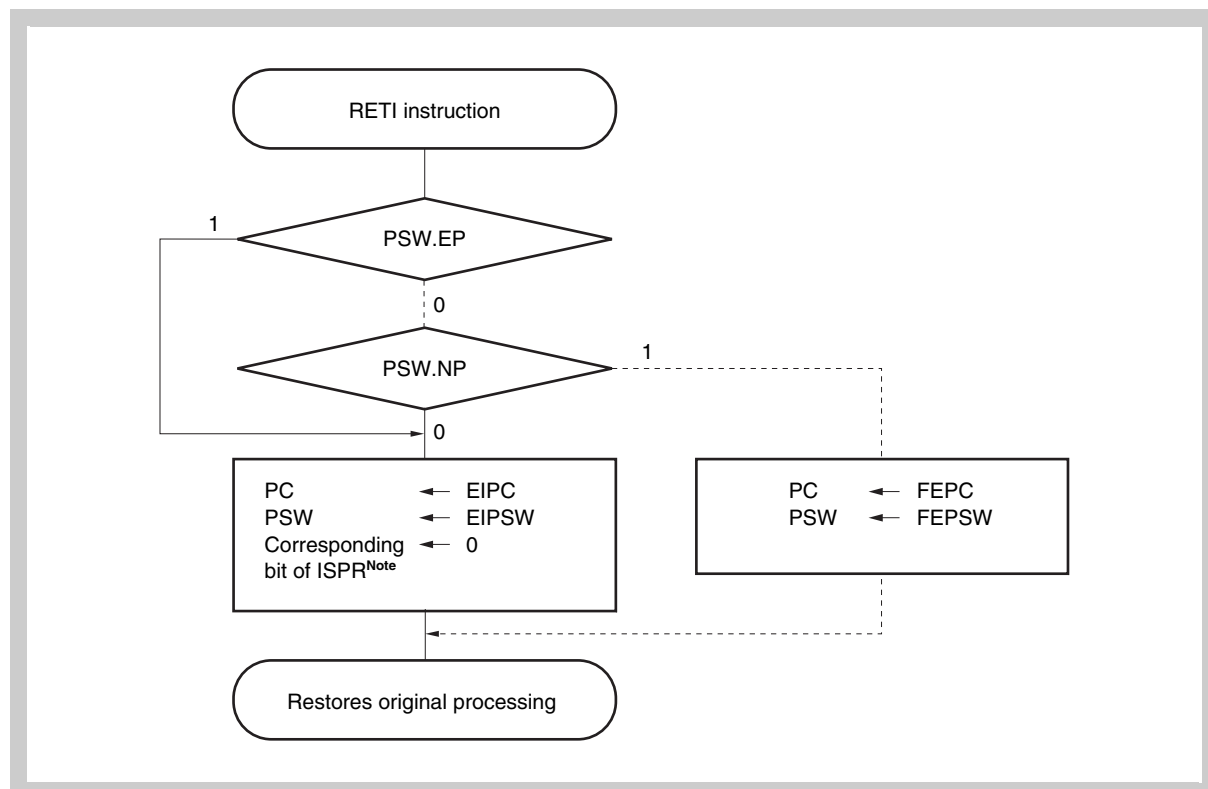


Figure 4-6 RETI instruction processing

- Note**
1. For the ISPR register, see “ISPR - In-service priority register” on page 122.
  2. The solid lines show the CPU processing flow.

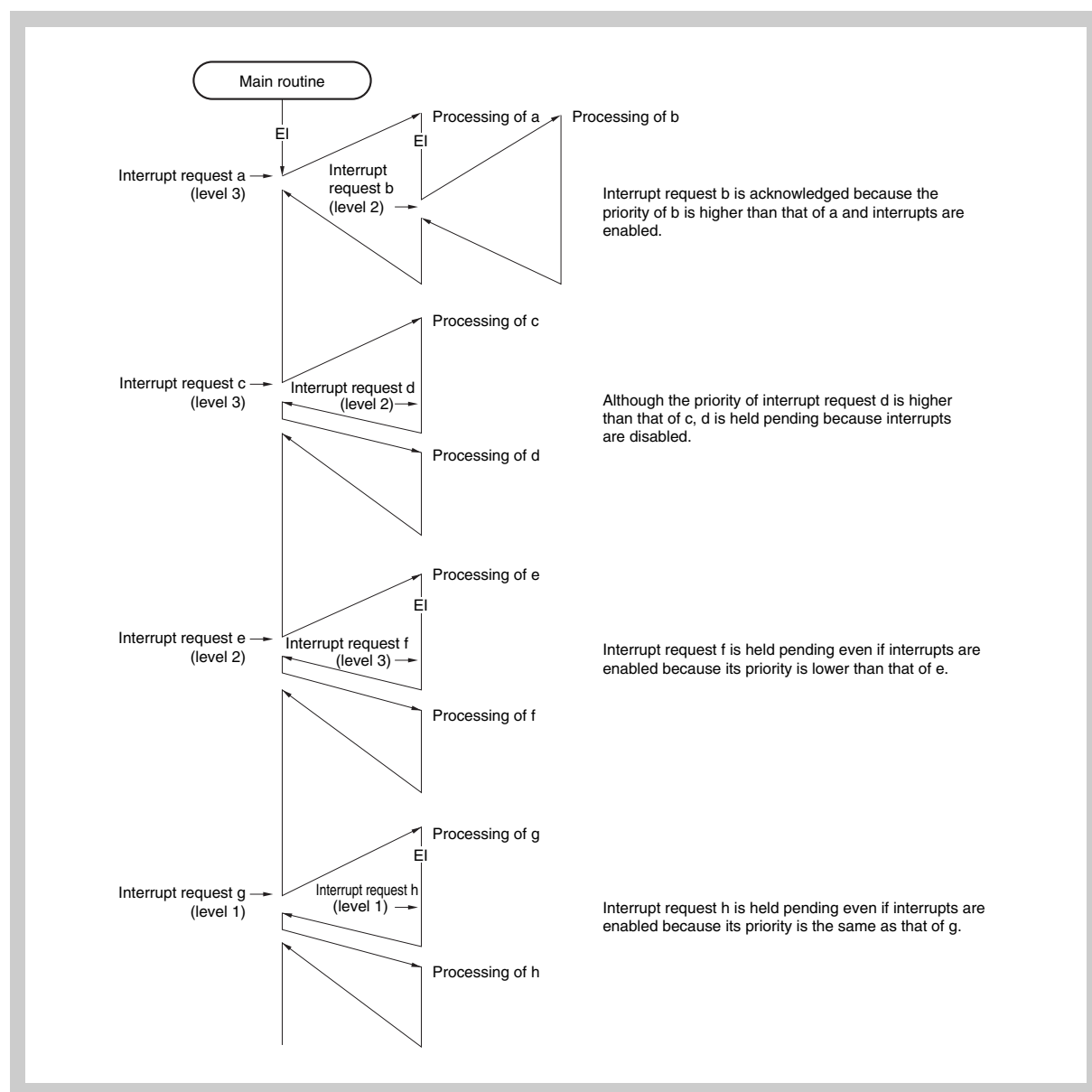
**Caution** When the PSW.EP bit and the PSW.NP bit are changed by the LDSR instruction during maskable interrupt processing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 0 and PSW.NP back to 0 using the LDSR instruction immediately before the RETI instruction.

### 4.3.3 Priorities of maskable interrupts

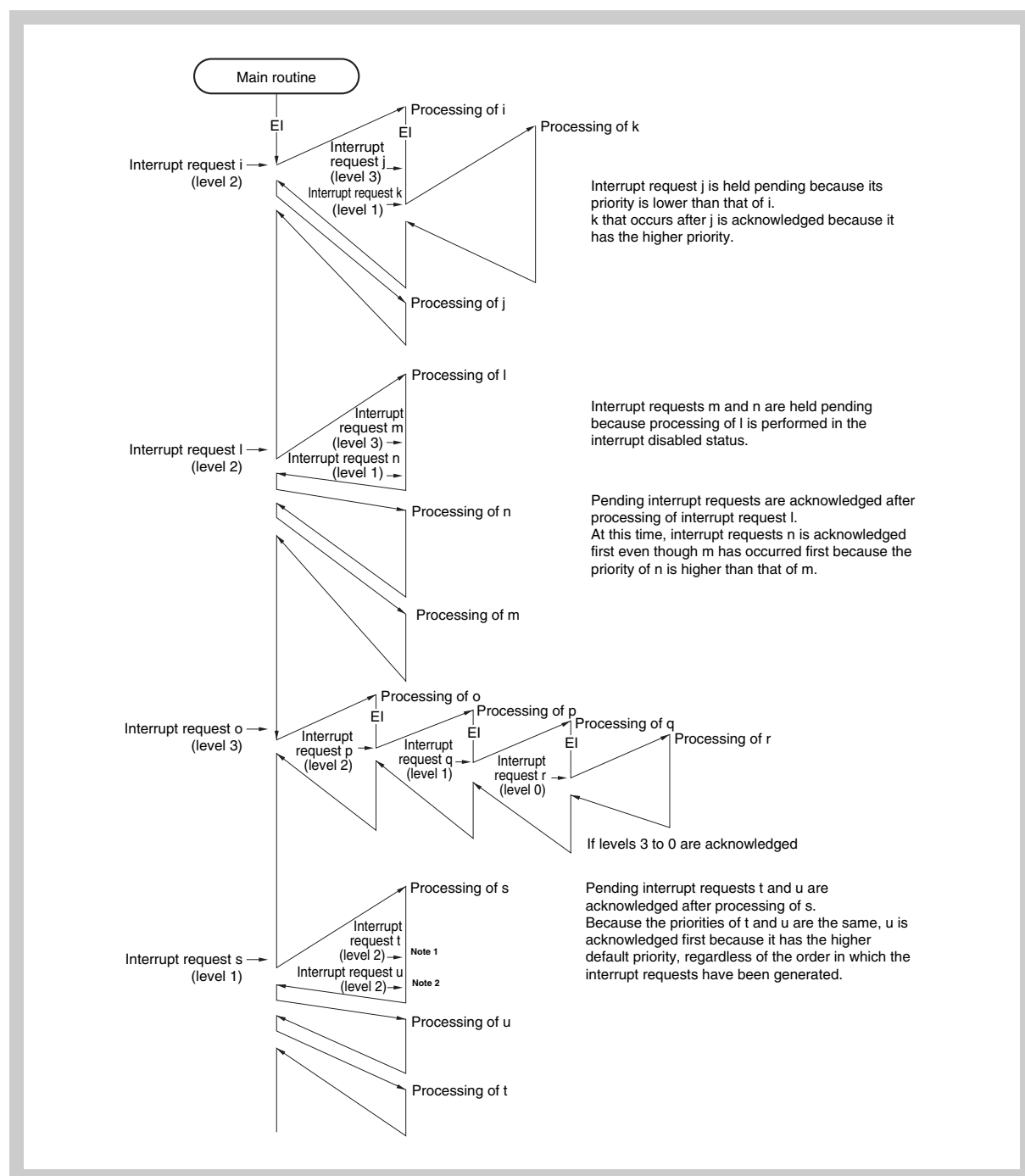
This microcontroller provides multiple interrupt servicing in which an interrupt is acknowledged while another interrupt is being serviced. Multiple interrupts can be controlled by priority levels.

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 type (default priority level) beforehand. For more information, refer to the interrupt/exception source list table. The programmable priority control customizes interrupt requests into eight levels by setting the priority level specification flag.

Note that when an interrupt request is acknowledged, the ID flag of PSW is automatically set to 1. Therefore, when multiple interrupts are to be used, clear the ID flag to 0 beforehand (for example, by placing the EI instruction in the interrupt service program) to set the interrupt enable mode.



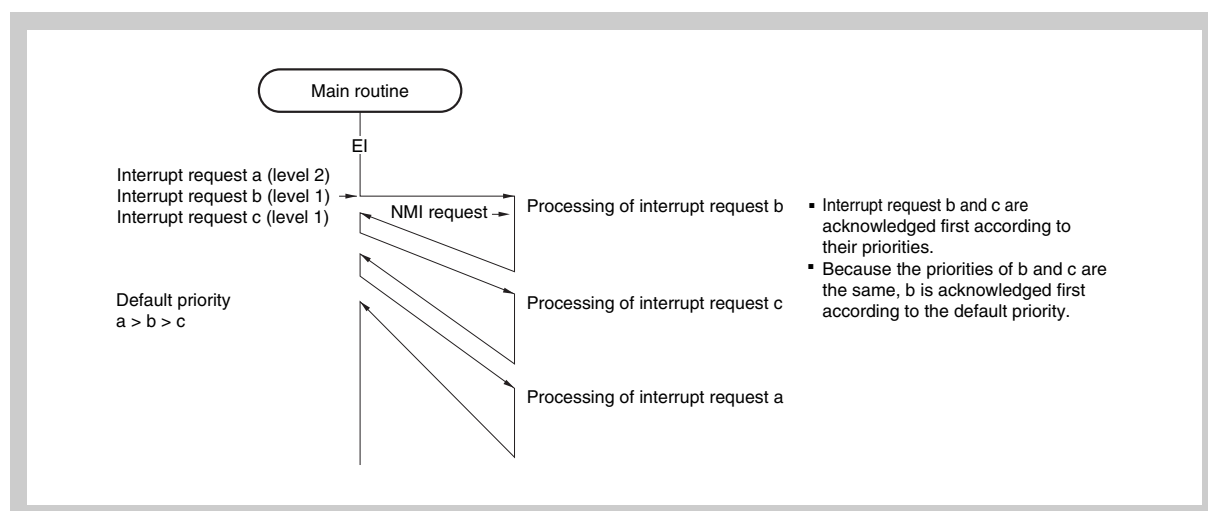
**Figure 4-7** Example of processing in which another interrupt request is issued while an interrupt is being processed (1/2)



**Figure 4-8** Example of processing in which another interrupt request is issued while an interrupt is being processed (2/2)

**Caution** The values of the EIPC and EIPSW registers must be saved before executing multiple interrupts. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

- Note**
1. <a> to <u> in the figures are the temporary names of interrupt requests shown for the sake of explanation.
  2. The default priority in the figure indicates the relative priority between two interrupt requests.



**Figure 4-9** Example of processing interrupt requests simultaneously generated

**Caution** The values of the EIPC and EIPSW registers must be saved before executing multiple interrupts. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

**Note** <a> to <c> in the figure are the temporary names of interrupt requests shown for the sake of explanation.



### 4.3.4 xxICn - Maskable interrupts control register

An interrupt control register is assigned to each interrupt request (maskable interrupt) and sets the control conditions for each maskable interrupt request.

**Access** These registers can be read/written in 8-bit or 1-bit units.

**Address** refer to *Table 4-4 on page 118*

**Initial Value** 47<sub>H</sub>. These registers are cleared by any reset.

	7	6	5	4	3	2	1	0
<b>xxICn</b>	xxIFn	xxMKn	0	0	0	xxPR2	xxPR1	xxPR0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>TSnCCR1/ TSnCCR1A</b>	Compare value 1															
	R/W															

**Table 4-3 xxICn register contents**

Bit position	Bit name	Function																																				
7	xxIFn	This is an interrupt request flag. 0: Interrupt request not issued 1: Interrupt request issued The flag xxIFn is reset automatically by the hardware if an interrupt request is acknowledged.																																				
6	xxMKn	This is an interrupt mask flag. 0: Enables interrupt processing 1: Disables interrupt processing (pending)																																				
2 to 0	xxPR2 to xxPR0	8 levels of priority order are specified for each interrupt.																																				
		<table><tr><th>xxPR2</th><th>xxPR1</th><th>xxPR0</th><th>Interrupt priority specification bit</th></tr><tr><td>0</td><td>0</td><td>0</td><td>Specifies level 0 (highest)</td></tr><tr><td>0</td><td>0</td><td>1</td><td>Specifies level 1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>Specifies level 2</td></tr><tr><td>0</td><td>1</td><td>1</td><td>Specifies level 3</td></tr><tr><td>1</td><td>0</td><td>0</td><td>Specifies level 4</td></tr><tr><td>1</td><td>0</td><td>1</td><td>Specifies level 5</td></tr><tr><td>1</td><td>1</td><td>0</td><td>Specifies level 6</td></tr><tr><td>1</td><td>1</td><td>1</td><td>Specifies level 7 (lowest)</td></tr></table>	xxPR2	xxPR1	xxPR0	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)
		xxPR2	xxPR1	xxPR0	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** xx: identification name of each peripheral unit. For a complete list of all interrupt control registers see *Table 4-4 on page 118*.

The address and bit of each interrupt control register is shown in the following table.

Table 4-4 Addresses and bits of interrupt control registers (1/3)

Address	Register	Bit						Associated interrupt
		7	6	5 - 3	2	1	0	
FFFFF110H	ERRIC	ERRIF	ERRMK	0	ERRPR2	ERRPR1	ERRPR0	INTDEDF INTDEDR INTDEDFR
FFFFF112H	OSDIC	OSDIF	OSDMK	0	OSDPR2	OSDPR1	OSDPR0	INTOSD
FFFFF114H	PIC0	PIF0	PMK0	0	PPR02	PPR01	PPR00	INTP0
FFFFF116H	PIC1	PIF1	PMK1	0	PPR12	PPR11	PPR10	INTP1
FFFFF118H	PIC2	PIF2	PMK2	0	PPR22	PPR21	PPR20	INTP2
FFFFF11AH	PIC3	PIF3	PMK3	0	PPR32	PPR31	PPR30	INTP3
FFFFF11CH	PIC4	PIF4	PMK4	0	PPR42	PPR41	PPR40	INTUC0R INTP4
FFFFF11EH	PIC5	PIF5	PMK5	0	PPR52	PPR51	PPR50	INTUC1R INTP5
FFFFF120H	PIC6	PIF6	PMK6	0	PPR62	PPR61	PPR60	INTP6
FFFFF122H	PIC7	PIF7	PMK7	0	PPR72	PPR71	PPR70	INTP7
FFFFF124H	PIC8	PIF8	PMK8	0	PPR82	PPR81	PPR80	INTP8
FFFFF126H	PIC9	PIF9	PMK9	0	PPR92	PPR91	PPR90	INTP9
FFFFF128H	PIC10	PIF10	PMK10	0	PPR102	PPR101	PPR100	INTP10
FFFFF12AH	PIC11	PIF11	PMK11	0	PPR112	PPR111	PPR110	INTP11
FFFFF12CH	PIC12	PIF12	PMK12	0	PPR122	PPR121	PPR120	INTP12
FFFFF12EH	TS0OVIC	TS0OVIF	TS0OVMK	0	TS0OVPR2	TS0OVPR1	TS0OVPR0	INTTS0OV
FFFFF130H	TS0CCIC0	TS0CCIF0	TS0CCMK0	0	TS0CCPR02	TS0CCPR01	TS0CCPR00	INTTS0CC0
FFFFF132H	TS0CCIC1	TS0CCIF1	TS0CCMK1	0	TS0CCPR12	TS0CCPR11	TS0CCPR10	INTTS0CC1
FFFFF134H	TS0CCIC2	TS0CCIF2	TS0CCMK2	0	TS0CCPR22	TS0CCPR21	TS0CCPR20	INTTS0CC2
FFFFF136H	TS0CCIC3	TS0CCIF3	TS0CCMK3	0	TS0CCPR32	TS0CCPR31	TS0CCPR30	INTTS0CC3
FFFFF138H	TS0CCIC4	TS0CCIF4	TS0CCMK4	0	TS0CCPR42	TS0CCPR41	TS0CCPR40	INTTS0CC4
FFFFF13AH	TS0CCIC5	TS0CCIF5	TS0CCMK5	0	TS0CCPR52	TS0CCPR51	TS0CCPR50	INTTS0CC5
FFFFF13CH	TS0CDIC0	TS0CDIF0	TS0CDMK0	0	TS0CDPR02	TS0CDPR01	TS0CDPR00	INTTS0CD0
FFFFF13EH	TS0ODIC	TS0ODIF	TS0ODMK	0	TS0ODPR2	TS0ODPR1	TS0ODPR0	INTTS0OD
FFFFF140H	TS0ERIC	TS0ERIF	TS0ERMK	0	TS0ERPR2	TS0ERPR1	TS0ERPR0	INTTS0ER
FFFFF142H	TS0WNIC	TS0WNIF	TS0WNMK	0	TS0WNPR2	TS0WNPR1	TS0WNPR0	INTTS0WN
FFFFF144H	TS1OVIC	TS1OVIF	TS1OVMK	0	TS1OVPR2	TS1OVPR1	TS1OVPR0	INTTS1OV
FFFFF146H	TS1CCIC0	TS1CCIF0	TS1CCMK0	0	TS1CCPR02	TS1CCPR01	TS1CCPR00	INTTS1CC0
FFFFF148H	TS1CCIC1	TS1CCIF1	TS1CCMK1	0	TS1CCPR12	TS1CCPR11	TS1CCPR10	INTTS1CC1
FFFFF14AH	TS1CCIC2	TS1CCIF2	TS1CCMK2	0	TS1CCPR22	TS1CCPR21	TS1CCPR20	INTTS1CC2
FFFFF14CH	TS1CCIC3	TS1CCIF3	TS1CCMK3	0	TS1CCPR32	TS1CCPR31	TS1CCPR30	INTTS1CC3
FFFFF14EH	TS1CCIC4	TS1CCIF4	TS1CCMK4	0	TS1CCPR42	TS1CCPR41	TS1CCPR40	INTTS1CC4
FFFFF150H	TS1CCIC5	TS1CCIF5	TS1CCMK5	0	TS1CCPR52	TS1CCPR51	TS1CCPR50	INTTS1CC5
FFFFF152H	TS1CDIC0	TS1CDIF0	TS1CDMK0	0	TS1CDPR02	TS1CDPR01	TS1CDPR00	INTTS1CD0
FFFFF154H	TS1ODIC	TS1ODIF	TS1ODMK	0	TS1ODPR2	TS1ODPR1	TS1ODPR0	INTTS1OD
FFFFF156H	TS1ERIC	TS1ERIF	TS1ERMK	0	TS1ERPR2	TS1ERPR1	TS1ERPR0	INTTS1ER
FFFFF158H	TS1WNIC	TS1WNIF	TS1WNMK	0	TS1WNPR2	TS1WNPR1	TS1WNPR0	INTTS1WN
FFFFF15AH	TT0OVIC	TT0OVIF	TT0OVMK	0	TT0OVPR2	TT0OVPR1	TT0OVPR0	INTTT0OV

Table 4-4 Addresses and bits of interrupt control registers (2/3)

Address	Register	Bit						Associated interrupt
		7	6	5 - 3	2	1	0	
FFFFF15CH	TT0CCIC0	TT0CCIF0	TT0CCMK0	0	TT0CCPR02	TT0CCPR01	TT0CCPR00	INTTT0CC0
FFFFF15EH	TT0CCIC1	TT0CCIF1	TT0CCMK1	0	TT0CCPR12	TT0CCPR11	TT0CCPR10	INTTT0CC1
FFFFF160H	TT0ECIC	TT0ECIF	TT0ECMK	0	TT0ECPR2	TT0ECPR1	TT0ECPR0	INTTT0EC
FFFFF162H	TT1OVIC	TT1OVIF	TT1OVMK	0	TT1OVPR2	TT1OVPR1	TT1OVPR0	INTTT1OV
FFFFF164H	TT1CCIC0	TT1CCIF0	TT1CCMK0	0	TT1CCPR02	TT1CCPR01	TT1CCPR00	INTTT1CC0
FFFFF166H	TT1CCIC1	TT1CCIF1	TT1CCMK1	0	TT1CCPR12	TT1CCPR11	TT1CCPR10	INTTT1CC1
FFFFF168H	TT1ECIC	TT1ECIF	TT1ECMK	0	TT1ECPR2	TT1ECPR1	TT1ECPR0	INTTT1EC
FFFFF16AH	TAA0OVIC	TAA0OVIF	TAA0OVMK	0	TAA0OVPR2	TAA0OVPR1	TAA0OVPR0	INTTAA0OV
FFFFF16CH	TAA0CCIC0	TAA0CCIF0	TAA0CCMK0	0	TAA0CCPR02	TAA0CCPR01	TAA0CCPR00	INTTAA0CC0
FFFFF16EH	TAA0CCIC1	TAA0CCIF1	TAA0CCMK1	0	TAA0CCPR12	TAA0CCPR11	TAA0CCPR10	INTTAA0CC1
FFFFF170H	TAA1OVIC	TAA1OVIF	TAA1OVMK	0	TAA1OVPR2	TAA1OVPR1	TAA1OVPR0	INTTAA1OV
FFFFF172H	TAA1CCIC0	TAA1CCIF0	TAA1CCMK0	0	TAA1CCPR02	TAA1CCPR01	TAA1CCPR00	INTTAA1CC0
FFFFF174H	TAA1CCIC1	TAA1CCIF1	TAA1CCMK1	0	TAA1CCPR12	TAA1CCPR11	TAA1CCPR10	INTTAA1CC1
FFFFF176H	TAA2OVIC	TAA2OVIF	TAA2OVMK	0	TAA2OVPR2	TAA2OVPR1	TAA2OVPR0	INTTAA2OV
FFFFF178H	TAA2CCIC0	TAA2CCIF0	TAA2CCMK0	0	TAA2CCPR02	TAA2CCPR01	TAA2CCPR00	INTTAA2CC0
FFFFF17AH	TAA2CCIC1	TAA2CCIF1	TAA2CCMK1	0	TAA2CCPR12	TAA2CCPR11	TAA2CCPR10	INTTAA2CC1
FFFFF17CH	TAA3OVIC	TAA3OVIF	TAA3OVMK	0	TAA3OVPR2	TAA3OVPR1	TAA3OVPR0	INTTAA3OV
FFFFF17EH	TAA3CCIC0	TAA3CCIF0	TAA3CCMK0	0	TAA3CCPR02	TAA3CCPR01	TAA3CCPR00	INTTAA3CC0
FFFFF180H	TAA3CCIC1	TAA3CCIF1	TAA3CCMK1	0	TAA3CCPR12	TAA3CCPR11	TAA3CCPR10	INTTAA3CC1
FFFFF182H	TAA4OVIC	TAA4OVIF	TAA4OVMK	0	TAA4OVPR2	TAA4OVPR1	TAA4OVPR0	INTTAA4OV
FFFFF184H	TAA4CCIC0	TAA4CCIF0	TAA4CCMK0	0	TAA4CCPR02	TAA4CCPR01	TAA4CCPR00	INTTAA4CC0
FFFFF186H	TAA4CCIC1	TAA4CCIF1	TAA4CCMK1	0	TAA4CCPR12	TAA4CCPR11	TAA4CCPR10	INTTAA4CC1
FFFFF188H	TAA5OVIC	TAA5OVIF	TAA5OVMK	0	TAA5OVPR2	TAA5OVPR1	TAA5OVPR0	INTTAA5OV
FFFFF18AH	TAA5CCIC0	TAA5CCIF0	TAA5CCMK0	0	TAA5CCPR02	TAA5CCPR01	TAA5CCPR00	INTTAA5CC0
FFFFF18CH	TAA5CCIC1	TAA5CCIF1	TAA5CCMK1	0	TAA5CCPR12	TAA5CCPR11	TAA5CCPR10	INTTAA5CC1
FFFFF18EH	TAA6OVIC	TAA6OVIF	TAA6OVMK	0	TAA6OVPR2	TAA6OVPR1	TAA6OVPR0	INTTAA6OV
FFFFF190H	TAA6CCIC0	TAA6CCIF0	TAA6CCMK0	0	TAA6CCPR02	TAA6CCPR01	TAA6CCPR00	INTTAA6CC0
FFFFF192H	TAA6CCIC1	TAA6CCIF1	TAA6CCMK1	0	TAA6CCPR12	TAA6CCPR11	TAA6CCPR10	INTTAA6CC1
FFFFF194H	TAA7OVIC	TAA7OVIF	TAA7OVMK	0	TAA7OVPR2	TAA7OVPR1	TAA7OVPR0	INTTAA7OV
FFFFF196H	TAA7CCIC0	TAA7CCIF0	TAA7CCMK0	0	TAA7CCPR02	TAA7CCPR01	TAA7CCPR00	INTTAA7CC0
FFFFF198H	TAA7CCIC1	TAA7CCIF1	TAA7CCMK1	0	TAA7CCPR12	TAA7CCPR11	TAA7CCPR10	INTTAA7CC1
FFFFF19AH	TAA8OVIC	TAA8OVIF	TAA8OVMK	0	TAA8OVPR2	TAA8OVPR1	TAA8OVPR0	INTTAA8OV
FFFFF19CH	TAA8CCIC0	TAA8CCIF0	TAA8CCMK0	0	TAA8CCPR02	TAA8CCPR01	TAA8CCPR00	INTTAA8CC0
FFFFF19EH	TAA8CCIC1	TAA8CCIF1	TAA8CCMK1	0	TAA8CCPR12	TAA8CCPR11	TAA8CCPR10	INTTAA8CC1
FFFFF1A0H	TAA9OVIC	TAA9OVIF	TAA9OVMK	0	TAA9OVPR2	TAA9OVPR1	TAA9OVPR0	INTTAA9OV
FFFFF1A2H	TAA9CCIC0	TAA9CCIF0	TAA9CCMK0	0	TAA9CCPR02	TAA9CCPR01	TAA9CCPR00	INTTAA9CC0
FFFFF1A4H	TAA9CCIC1	TAA9CCIF1	TAA9CCMK1	0	TAA9CCPR12	TAA9CCPR11	TAA9CCPR10	INTTAA9CC1
FFFFF1A6H	BRGIC	BRGIF	BRGMK	0	BRGPR2	BRGPR1	BRGPR0	INTBRG0 INTBRG1
FFFFF1A8H	BRGIC2	BRGIF2	BRGMK2	0	BRGPR22	BRGPR21	BRGPR20	INTBRG2
FFFFF1B6H	C0ERRIC	C0ERRIF	C0ERRMK	0	C0ERRPR2	C0ERRPR1	C0ERRPR0	INTC0ERR

**Table 4-4 Addresses and bits of interrupt control registers (3/3)**

Address	Register	Bit						Associated interrupt
		7	6	5 - 3	2	1	0	
FFFFF1B8H	C0WUPIC	C0WUPIF	C0WUPMK	0	C0WUPPR2	C0WUPPR1	C0WUPPR0	INTC0WUP
FFFFF1BAH	C0RECIC	C0RECIF	C0RECMK	0	C0RECPR2	C0RECPR1	C0RECPR0	INTC0REC
FFFFF1BCH	C0TRXIC	C0TRXIF	C0TRXMK	0	C0TRXPR2	C0TRXPR1	C0TRXPR0	INTC0TRX
FFFFF1BEH	C1ERRIC	C1ERRIF	C1ERRMK	0	C1ERRPR2	C1ERRPR1	C1ERRPR0	INTC1ERR
FFFFF1C0H	C1WUPIC	C1WUPIF	C1WUPMK	0	C1WUPPR2	C1WUPPR1	C1WUPPR0	INTC1WUP
FFFFF1C2H	C1RECIC	C1RECIF	C1RECMK	0	C1RECPR2	C1RECPR1	C1RECPR0	INTC1REC
FFFFF1C4H	C1TRXIC	C1TRXIF	C1TRXMK	0	C1TRXPR2	C1TRXPR1	C1TRXPR0	INTC1TRX
FFFFF1D6H	CB0TIC	CB0TIF	CB0TMK	0	CB0TPR2	CB0TPR1	CB0TPR0	INTCB0T
FFFFF1D8H	CB0RIC	CB0RIF	CB0RMK	0	CB0RPR2	CB0RPR1	CB0RPR0	INTCB0R
FFFFF1DAH	CB0REIC	CB0REIF	CB0REMK	0	CB0REPR2	CB0REPR1	CB0REPR0	INTCB0RE
FFFFF1DCH	CB1TIC	CB1TIF	CB1TMK	0	CB1TPR2	CB1TPR1	CB1TPR0	INTCB1T
FFFFF1DEH	CB1RIC	CB1RIF	CB1RMK	0	CB1RPR2	CB1RPR1	CB1RPR0	INTCB1R
FFFFF1E0H	CB1REIC	CB1REIF	CB1REMK	0	CB1REPR2	CB1REPR1	CB1REPR0	INTCB1RE
FFFFF1E2H	CE0OFIC	CE0OFIF	CE0OFMK	0	CE0OFPR2	CE0OFPR1	CE0OFPR0	INTCE0OF
FFFFF1E4H	CE0CIC	CE0CIF	CE0CMK	0	CE0CPR2	CE0CPR1	CE0CPR0	INTCE0C
FFFFF1E6H	CE1OFIC	CE1OFIF	CE1OFMK	0	CE1OFPR2	CE1OFPR1	CE1OFPR0	INTCE1OF
FFFFF1E8H	CE1CIC	CE1CIF	CE1CMK	0	CE1CPR2	CE1CPR1	CE1CPR0	INTCE1C
FFFFF1EAH	UC0REIC	UC0REIF	UC0REMK	0	UC0REPR2	UC0REPR1	UC0REPR0	INTUC0RE
FFFFF1ECH	UC0TIC	UC0TIF	UC0TMK	0	UC0TPR2	UC0TPR1	UC0TPR0	INTUC0T
FFFFF1EEH	UC1REIC	UC1REIF	UC1REMK	0	UC1REPR2	UC1REPR1	UC1REPR0	INTUC1RE
FFFFF1F0H	UC1TIC	UC1TIF	UC1TMK	0	UC1TPR2	UC1TPR1	UC1TPR0	INTUC1T
FFFFF1F2H	ADIC0	ADIF0	ADMK0	0	ADPR02	ADPR01	ADPR00	INTAD0
FFFFF1F4H	ADIC1	ADIF1	ADMK1	0	ADPR12	ADPR11	ADPR10	INTAD1
FFFFF1F6H	DMAIC2	DMAIF2	DMAMK2	0	DMAPR22	DMAPR21	DMAPR20	INTDMA2
FFFFF1F8H	DMAIC3	DMAIF3	DMAMK3	0	DMAPR32	DMAPR31	DMAPR30	INTDMA3 INTFL

### 4.3.5 IMR0 to IMR7 - Interrupt mask registers

These registers set the interrupt mask state for the maskable interrupts.

The IMKn bit of the IMRm (m = 0 to 7) registers is equivalent to the xxMK bit of the xxIC register.

IMRm registers can be read/written in 16-bit, 8-bit, and 1-bit units.

The address of the lower 8-bit register IMRmL is equal to that of the 16-bit IMRm register, and the higher 8-bit register IMRmH can be accessed on the following address (address (IMRm) + 1).

- Caution**
1. Mask bits, indicated with “0”, must not be altered. Make sure to set them “0”
  2. Mask bits, indicated with “1”, must not be altered. Make sure to set them “1” when writing to the register.

IMR0	15	14	13	12	11	10	9	8	Address	Initial value
	TS0OVMK	PMK12	PMK11	PMK10	PMK9	PMK8	PMK7	PMK6		
	7	6	5	4	3	2	1	0	FFFFF100H	FFFFH
	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	OSDMK	ERRMK		
IMR1	15	14	13	12	11	10	9	8	Address	Initial value
	TS1CCMK4	TS1CCMK3	TS1CCMK2	TS1CCMK1	TS1CCMK0	TS1OVMK	TS0WNMK	TS0ERMK		
	7	6	5	4	3	2	1	0	FFFFF102H	FFFFH
	TS0ODMK	TS0CDMK0	TS0CCMK5	TS0CCMK4	TS0CCMK3	TS0CCMK2	TS0CCMK1	TS0CCMK0		
IMR2	15	14	13	12	11	10	9	8	Address	Initial value
	TTA0CCMK1	TTA0CCMK0	TTA0OVMK	TT1ECMK	TT1CCMK1	TT1CCMK0	TT1OVMK	TT0ECMK		
	7	6	5	4	3	2	1	0	FFFFF104H	FFFFH
	TT0CCMK1	TT0CCMK0	TT0OVMK	TS1WNMK	TS1ERMK	TS1ODMK	TS1CDMK0	TS1CCMK5		
IMR3	15	14	13	12	11	10	9	8	Address	Initial value
	TTA6OVMK	TTA5CCMK1	TTA5CCMK0	TTA5OVMK	TTA4CCMK1	TTA4CCMK0	TTA4OVMK	TTA3CCMK1		
	7	6	5	4	3	2	1	0	FFFFF106H	FFFFH
	TTA3CCMK0	TTA3OVMK	TTA2CCMK1	TTA2CCMK0	TTA2OVMK	TTA1CCMK1	TTA1CCMK0	TTA1OVMK		
IMR4	15	14	13	12	11	10	9	8	Address	Initial value
	1	1	1	BRGMK2	BRGMK	TTA9CCMK1	TTA9CCMK0	TTA9OVMK		
	7	6	5	4	3	2	1	0	FFFFF108H	FFFFH
	TTA8CCMK1	TTA8CCMK0	TTA8OVMK	TTA7CCMK1	TTA7CCMK0	TTA7OVMK	TTA6CCMK1	TTA6CCMK0		
IMR5	15	14	13	12	11	10	9	8	Address	Initial value
	1	1	1	1	1	C1TRXMK	C1RECMK	C1WUPMK		
	7	6	5	4	3	2	1	0	FFFFF10AH	FFFFH
	C1ERRMK	C0TRXMK	C0RECMK	C0WUPMK	C0ERRMK	1	1	1		
IMR6	15	14	13	12	11	10	9	8	Address	Initial value
	UC1REMK	UC0TMK	UC0REMK	CE1CMK	CE1OFMK	CE0CMK	CE0OFMK	CB1REMK		
	7	6	5	4	3	2	1	0	FFFFF10CH	FFFFH
	CB1RMK	CB1TMK	CB0REMK	CB0RMK	CB0TMK	1	1	1		
IMR7	15	14	13	12	11	10	9	8	Address	Initial value
	0	0	0	0	0	0	0	0		
	7	6	5	4	3	2	1	0	FFFFF10EH	001FH
	0	0	0	DMAMK3	DMAMK2	ADMK1	ADMK0	UC1TMK		

Bit position	Bit name	Function
15 to 0	xxMKn	Interrupt mask flag. 0: Interrupt servicing enabled 1: Interrupt servicing disabled (pending)

### 4.3.6 ISPR - In-service priority register

This register holds the priority level of the maskable interrupt currently acknowledged. When an interrupt request is acknowledged, the bit of this register corresponding to the priority level of that interrupt 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 having the highest priority is automatically reset to 0 by hardware. However, it is not reset to 0 when execution is returned from non-maskable interrupt servicing or exception processing.

This register is read-only in 8-bit or 1-bit units.

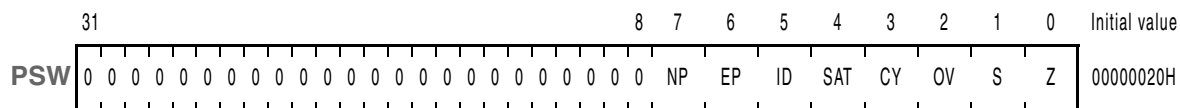
	7	6	5	4	3	2	1	0	Address	Initial value
ISPR	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0	FFFFF1FAH	00H

Bit position	Bit name	Function
7 to 0	ISPR7 to ISPR0	Indicates priority of interrupt currently acknowledged 0: Interrupt request with priority n not acknowledged 1: Interrupt request with priority n acknowledged

**Note** n = 0 to 7 (priority level)

### 4.3.7 Maskable interrupt status flag (ID)

The ID flag is bit 5 of the PSW and this controls the maskable interrupt's operating state, and stores control information regarding enabling or disabling of interrupt requests.



Bit position	Bit name	Function
5	ID	<p>Indicates whether maskable interrupt processing is enabled or disabled.</p> <p>0: Maskable interrupt request acknowledgement enabled</p> <p>1: Maskable interrupt request acknowledgement disabled (pending)</p> <p>This bit is set to 1 by the DI instruction and reset to 0 by the EI instruction. Its value is also modified by the RETI instruction or LDSR instruction when referencing to PSW.</p> <p>Non-maskable interrupt requests and exceptions are acknowledged regardless of this flag. when a maskable interrupt is acknowledged, the ID flag is automatically set to 1 by hardware.</p> <p>The interrupt request generated during the acknowledgement disabled period (ID = 1) is acknowledged when the PIFn bit of PICn register is set to 1, and the ID flag is reset to 0.</p>

### 4.3.8 External maskable interrupts

This microcontroller provides 13 maskable external interrupts INTP0 to INTP12 with the following features:

- Analog input filter (refer to “Analog filtered inputs” on page 63)
- Interrupt detection selectable for each interrupt input:
  - Rising edge
  - Falling edge
  - Both edges: rising and falling edge
- Wakeup capability from stand-by mode of INTPn upon
  - Rising edge
  - Falling edge
  - Both edges: rising and falling edge

For configuration of the external interrupt events refer to “Edge and Level Detection Configuration” on page 124.

## 4.4 Edge and Level Detection Configuration

The registers for the configuration of interrupts specify, whether an interrupt is detected at the falling, rising, or on both falling and rising edge of the signal. For every maskable or non-maskable interrupt two bits define the type of edge:

**Table 4-5 Edge specification of interrupts**

ESN1	ESN0	Function	
		INTP0 to INTP12	NMI
0	0	falling edge	falling edge
0	1	rising edge	rising edge
1	0	disabled (no INT generated)	Setting prohibited
1	1	falling and rising edge	falling and rising edge

### (1) INTM0 - External interrupt edge specification register

The 8-bit INTM0 register specifies the edge at which an interrupt is detected. This register concerns the pins NMI and INTP0 to INTP2.

**Access** This register can be read/written in 8-bit units.

**Address** FFFF F880<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is initialized by  $\overline{\text{RESET}}$ .

7	6	5	4	3	2	1	0
ES21	ES20	ES11	ES10	ES01	ES00	ESN1	ESN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
INTP2		INTP1		INTP0		NMI	

**Table 4-6 INTM0 register contents**

Bit position	Bit name	Function
7	ES21	Specifies the edge detection at terminal INTP2. See Table 4-5 on page 124.
6	ES20	
5	ES11	Specifies the edge detection at terminal INTP1. See Table 4-5 on page 124.
4	ES10	
3	ES01	Specifies the edge detection at terminal INTP0. See Table 4-5 on page 124.
2	ES00	
1	ESN1	Specifies the edge detection at terminal NMI. See Table 4-5 on page 124.
0	ESN0	



**(2) INTM1 - External interrupt edge specification register**

The 8-bit INTM1 register specifies the edge at which an interrupt is detected. This register concerns the pins INTP3 to INTP6.

**Access** This register can be read/written in 8-bit units.

**Address** FFFF F882<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is initialized by  $\overline{\text{RESET}}$ .

7	6	5	4	3	2	1	0
ES61	ES60	ES51	ES50	ES41	ES40	ES31	ES30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
INTP6		INTP5		INTP4		INTP3	

**Table 4-7 INTM1 register contents**

Bit position	Bit name	Function
7	ES61	Specifies the edge detection at terminal INTP6. See Table 4-5 on page 124.
6	ES60	
5	ES51	Specifies the edge detection at terminal INTP5. See Table 4-5 on page 124.
4	ES50	
3	ES41	Specifies the edge detection at terminal INTP4. See Table 4-5 on page 124.
2	ES40	
1	ES31	Specifies the edge detection at terminal INTP3. See Table 4-5 on page 124.
0	ES30	

**(3) INTM2 - External interrupt edge specification register**

The 8-bit INTM2 register specifies the edge at which an interrupt is detected. This register concerns the pins INTP7 to INTP10.

**Access** This register can be read/written in 8-bit units.

**Address** FFFF F884<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is initialized by  $\overline{\text{RESET}}$ .

7	6	5	4	3	2	1	0
ES101	ES100	ES91	ES90	ES81	ES80	ES71	ES70
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
INTP10		INTP9		INTP8		INTP7	

**Table 4-8 INTM2 register contents**

Bit position	Bit name	Function
7	ES101	Specifies the edge detection at terminal INTP10. See Table 4-5 on page 124.
6	ES100	
5	ES91	Specifies the edge detection at terminal INTP9. See Table 4-5 on page 124.
4	ES90	
3	ES81	Specifies the edge detection at terminal INTP8. See Table 4-5 on page 124.
2	ES80	
1	ES71	Specifies the edge detection at terminal INTP7. See Table 4-5 on page 124.
0	ES70	

**(4) INTM3 - External interrupt edge specification register**

The 8-bit INTM3 register specifies the edge at which an interrupt is detected. This register concerns the pins INTP11 to INTP12.

**Access** This register can be read/written in 8-bit units.

**Address** FFFF F886<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is initialized by  $\overline{\text{RESET}}$ .

7	6	5	4	3	2	1	0
0	0	0	0	ES121	ES120	ES111	ES110
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
				INTP12		INTP11	

**Table 4-9 INTM3 register contents**

Bit position	Bit name	Function
3	ES121	Specifies the edge detection at terminal INTP12. See Table 4-5 on page 124.
2	ES120	
1	ES111	Specifies the edge detection at terminal INTP11. See Table 4-5 on page 124.
0	ES110	

## 4.5 Interrupt Sharing

Interrupt sharing is supported for both non-maskable and maskable interrupts. The INTSEL register configures interrupt sharing.

**Non-maskable interrupts** Interrupt sharing for non-maskable interrupts can be enabled by setting bit INTSEL.ISR = 1. If interrupt sharing is enabled, the interrupt signals INTDEDF, INTDEDR, and INTOSD generate an NMI instead of a maskable interrupt.

Figure 4-10 shows the logic operations of interrupt sharing for NMI.

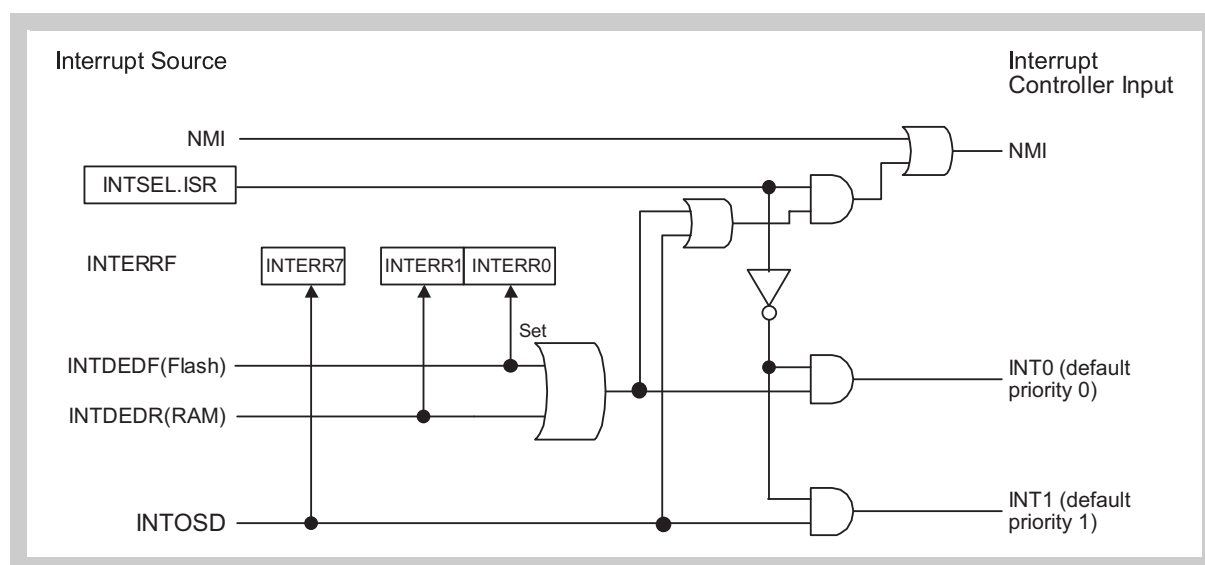


Figure 4-10 Interrupt sharing for NMI

**Maskable interrupts** Some interrupt sources share the same maskable interrupt. There are the following options for interrupt sharing:

- Any of the interrupt sources can generate the corresponding interrupt request. All these interrupt sources are combined with logical “or”.
- An interrupt selection bit defines which interrupt source can generate the corresponding interrupt request.

Table 4-10 on page 128 lists the shared maskable interrupts and, if available, their corresponding interrupt selection bit. For more details on the interrupt sources see the interrupt/exception source list in Table 4-1 on page 100.

Table 4-10 Shared maskable interrupts

Shared interrupts	Interrupt selection		Valid interrupt sources	Interrupt source identification	Default priority
INTDEDF / INTDEDR	logical “or” combination		INTDEDF / INTDEDR	status flags in the interrupt source module	0
INTP4 / INTUC0R	logical “or” combination		INTP4 / INTUC0R	INTUC0R identified by status flags in UARTC0.  If only INTUC0R interrupts shall be served, INP4 interrupts can be suppressed by setting INTM1.ES4[1:0] = 10 <sub>B</sub> .	6
INTP5 / INTUC1R	logical “or” combination		INTP5 / INTUC1R	INTUC1R identified by status flags in UARTC1.  If only INTUC1R interrupts shall be served, INP5 interrupts can be suppressed by setting INTM1.ES5[1:0] = 10 <sub>B</sub> .	7
INTBRG0 / INTBRG1	INTSEL.BRGSSR	0	INTBRG0	determined by INTSEL.BRGSSR	75
		1	INTBRG1		
INTDMA3	INTSEL.INTSEL4	0	INTDMA3	—	116

**(1) INTSEL - Interrupt share enable register**

The 8-bit INTSEL register specifies

- which interrupt source generates a maskable interrupt with priorities 75, 116
- whether or not the interrupt source for the NMI is shared

**Access** This register can be read/written in 8-bit and 1-bit units.

**Address** FFFF F6D0<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is initialized by  $\overline{\text{RESET}}$ .

7	6	5	4	3	2	1	0
ISR	0	0	INTSEL4	0	0	0	BRGSSR
R/W	R	R	R/W	R	R	R	R/W

**Table 4-11 INTSEL register contents**

Bit position	Bit name	Function
7	ISR	Enables/disables sharing of NMI 0: NMI sharing disabled 1: NMI sharing enabled <hr/> <b>Caution:</b> ISR can only be changed once after reset. Thus the NMI can be enabled after reset, but not disabled afterwards.
4	INTSEL4	Maskable interrupt with priority 116 is generated by 0: INTDMA3 1: reserved
0	BRGSSR	Maskable interrupt with priority 75 is generated by 0: INTBRG0 1: INTBRG1

**Caution** Before changing any interrupt selection bit in this register the related interrupt sources must be disabled. Otherwise unrelated interrupt requests may occur.

**(2) INTERRF - Interrupt source flag register**

The 8-bit INTERRF register identifies the interrupt source of NMI and INT0 (default priority 0) interrupts.

**Access** This register can be read/written in 8-bit and 1-bit units.

**Address** FFFF F6D2<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is initialized by  $\overline{\text{RESET}}$ .

7	6	5	4	3	2	1	0
INTERR7	0	0	0	0	0	INTERR1	INTERR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 4-12 INTTERF register contents**

Bit position	Bit name	Function
7	INTERR7	Displays the status of interrupt INTOSD (Clock Monitor) 0: Interrupt not generated 1: Interrupt generated
1	INTERR1	Displays the status of interrupt INTDEDR (RAM) 0: Interrupt not generated 1: Interrupt generated
0	INTERR0	Displays the status of interrupt INTDEDF (Flash) 0: Interrupt not generated 1: Interrupt generated

- Note**
1. After the CPU has processed the associated interrupt routine, the related flag in the INTERRF register must be reset to '0' by a bit manipulation instruction. If a further interrupt occurs while the associated bit in the register is reset, the new interrupt request will also be discarded.
  2. Writing to ROMEAD register is prohibited while INTERR0 bit is "1".  
Writing to RAMEAD register is prohibited while INTERR1 bit is "1".

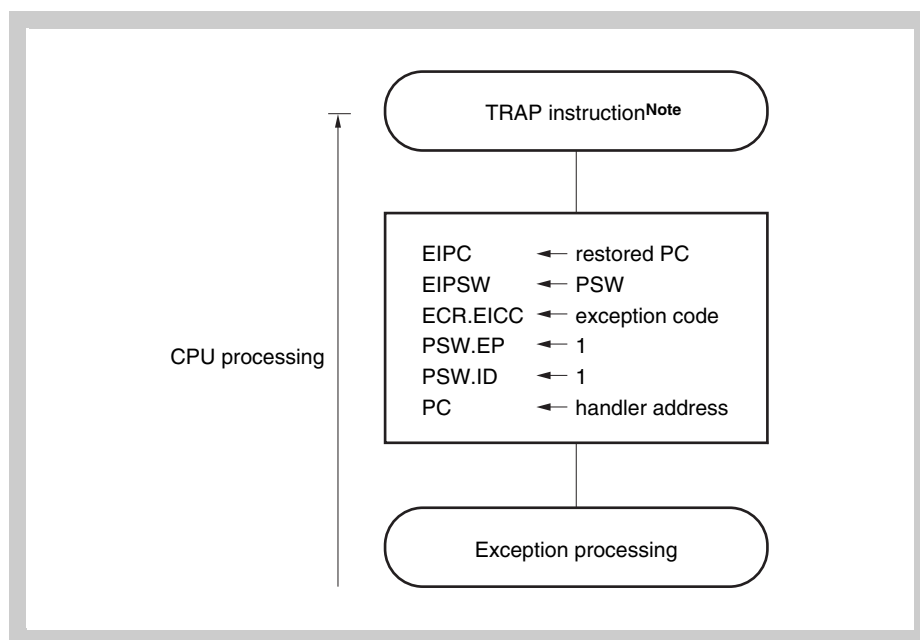
## 4.6 Software Exception

### 4.6.1 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 EP and ID bits of the PSW.
5. Sets the handler address (00000040H or 00000050H) corresponding to the software exception to the PC, and transfers control.

Figure 4-11 illustrates the processing of a software exception.



**Figure 4-11** Software exception processing

**Note** TRAP Instruction Format: TRAP vector (the vector is a value from 0 to 1FH.)

The handler address is determined by the TRAP instruction's operand (vector). If the vector is 0 to 0FH, it becomes 00000040H, and if the vector is 10H to 1FH, it becomes 00000050H.

### 4.6.2 Restore

Recovery from software exception processing is carried out by the RETI instruction.

By executing the RETI instruction, the CPU carries out the following processing and shifts control to the restored PC's address.

1. Loads the restored PC and PSW from EIPC and EIPSW because the EP bit of the PSW is 1.
2. Transfers control to the address of the restored PC and PSW.

Figure 4-12 illustrates the processing of the RETI instruction.

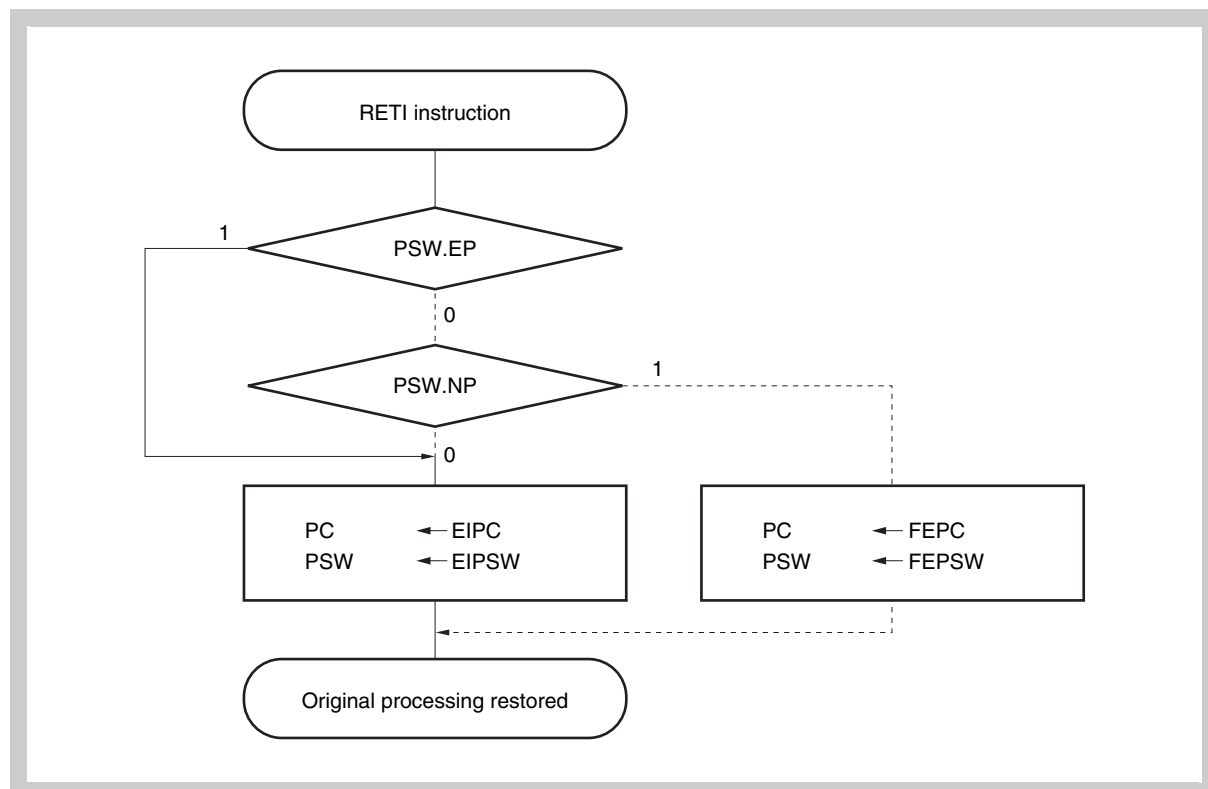


Figure 4-12 RETI instruction processing

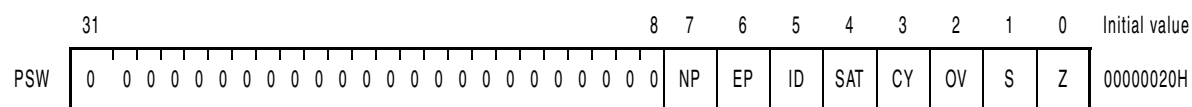
**Caution** When the PSW.EP bit and the PSW.NP bit are changed by the LDSR instruction during the software exception processing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 1 using the LDSR instruction immediately before the RETI instruction.

**Note** The solid lines show the CPU processing flow.



### 4.6.3 Exception status flag (EP)

The EP flag is bit 6 of PSW, and is a status flag used to indicate that exception processing is in progress. It is set when an exception occurs.



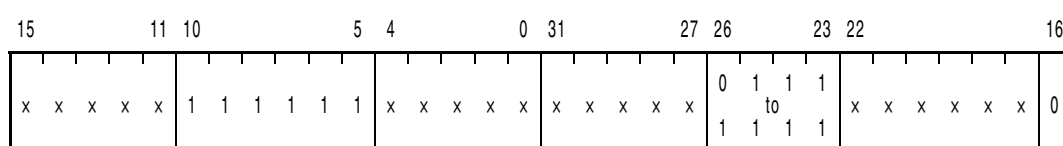
Bit position	Bit name	Function
6	EP	Shows that exception processing is in progress. 0: Exception processing not in progress. 1: Exception processing in progress.

## 4.7 Exception Trap

An exception trap is an interrupt that is requested when an illegal execution of an instruction takes place. For this microcontroller, an illegal opcode exception (ILGOP: Illegal Opcode Trap) is considered as an exception trap.

### 4.7.1 Illegal opcode definition

The illegal instruction has an opcode (bits 10 to 5) of 111111B, a sub-opcode (bits 23 to 26) 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.



**Note** x: Arbitrary

#### (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 NP, EP, and ID bits of the PSW.
4. Sets the handler address (00000060H) corresponding to the exception trap to the PC, and transfers control.

Figure 4-13 illustrates the processing of the exception trap.

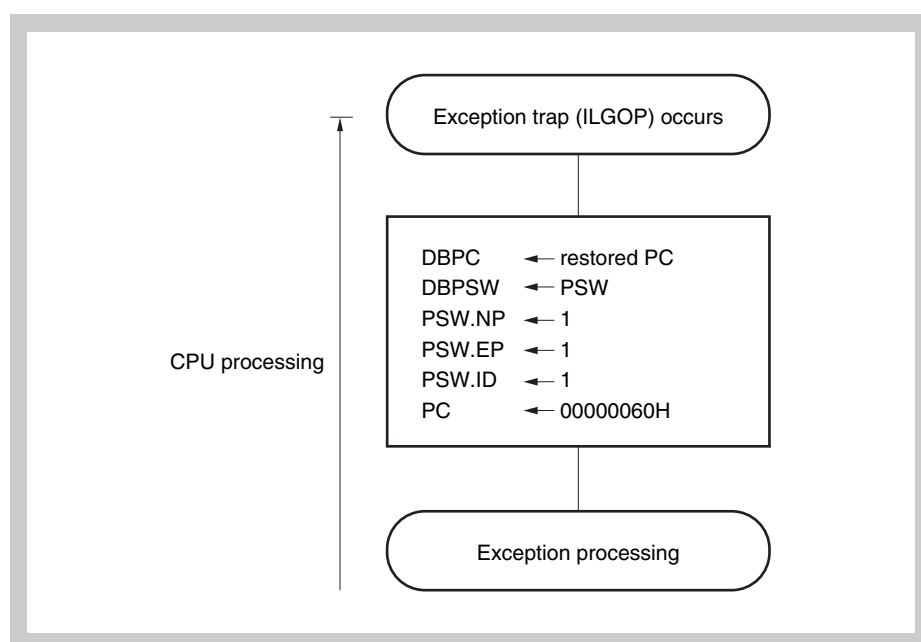


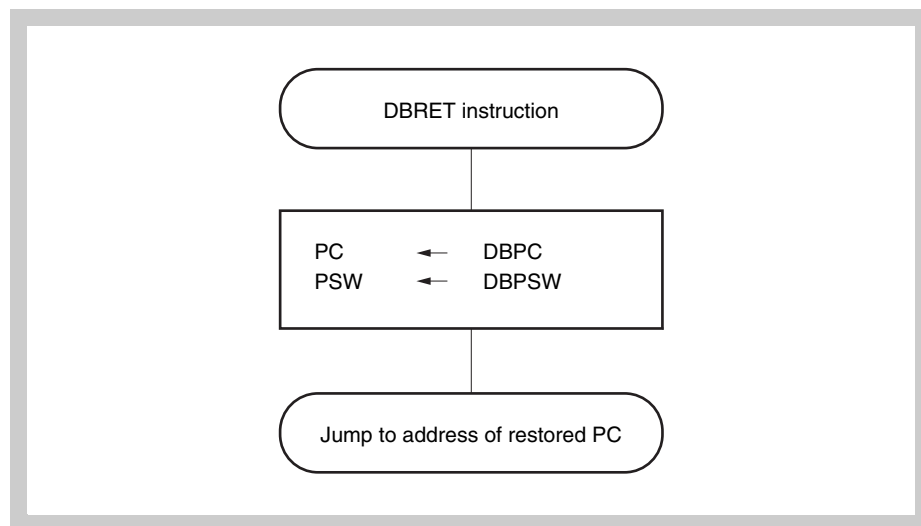
Figure 4-13 Exception trap processing

**(2) Restore**

Recovery from an exception trap is carried out by the DBRET instruction. By executing the DBRET instruction, the CPU carries out the following processing and controls the address of the restored PC.

1. Loads the restored PC and PSW from DBPC and DBPSW.
2. Transfers control to the address indicated by the restored PC and PSW.

Figure 4-14 illustrates the restore processing from an exception trap.



**Figure 4-14** Restore processing from exception trap

### 4.7.2 Debug trap

The debug trap is an exception that can be acknowledged every time and is generated by execution of the DBTRAP instruction.

When the debug trap is generated, the CPU performs the following processing.

#### (1) Operation

When the debug trap is generated, the CPU performs the following processing, transfers control to the debug monitor routine, and shifts to debug mode.

1. Saves the restored PC to DBPC.
2. Saves the current PSW to DBPSW.
3. Sets the NP, EP and ID bits of the PSW.
4. Sets the handler address (00000060H) corresponding to the debug trap to the PC and transfers control.

Figure 4-15 illustrates the processing of the debug trap.

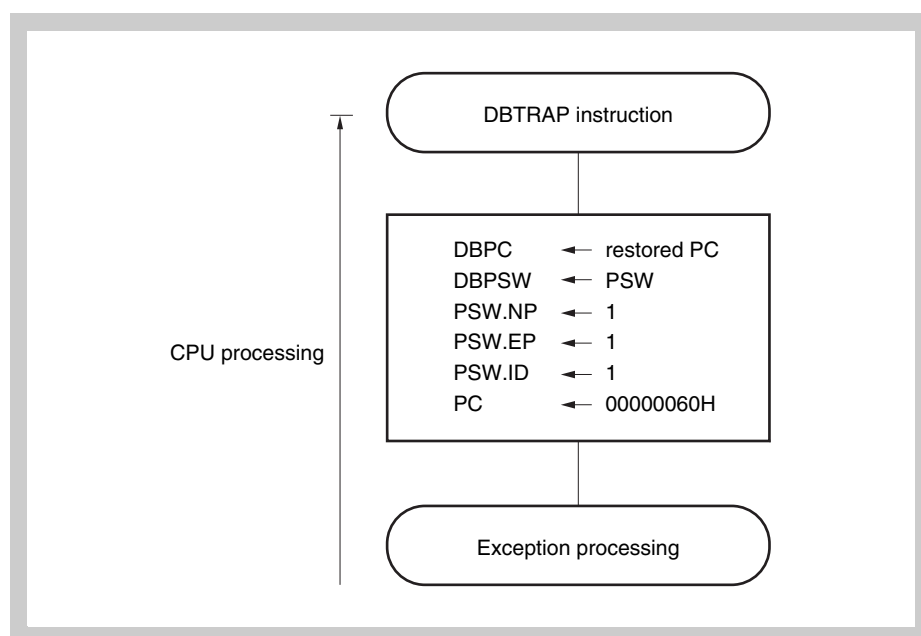


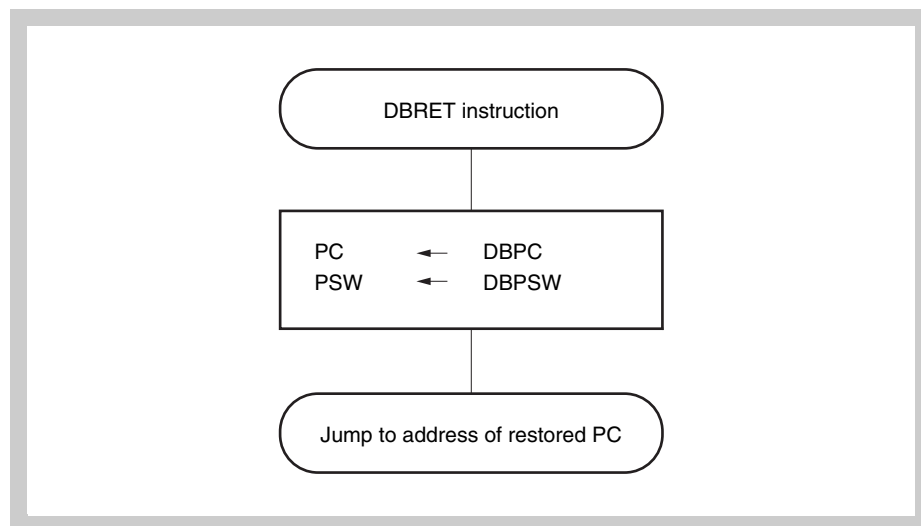
Figure 4-15 Debug trap processing

**(2) Restore**

Recovery from a debug trap is carried out by the DBRET instruction. By executing the DBRET instruction, the CPU carries out the following processing and controls the address of the restored PC.

1. Loads the restored PC and PSW from DBPC and DBPSW.
2. Transfers control to the address indicated by the restored PC and PSW.

Figure 4-16 illustrates the restore processing from a debug trap.



**Figure 4-16** Restore processing from debug trap

## 4.8 Multiple Interrupt Processing Control

Multiple interrupt processing control is a process by which an interrupt request that is currently being processed can be interrupted during processing if there is an interrupt request with a higher priority level, and the higher priority interrupt request is received and processed first.

If there is an interrupt request with a lower priority level than the interrupt request currently being processed, that interrupt request is held pending.

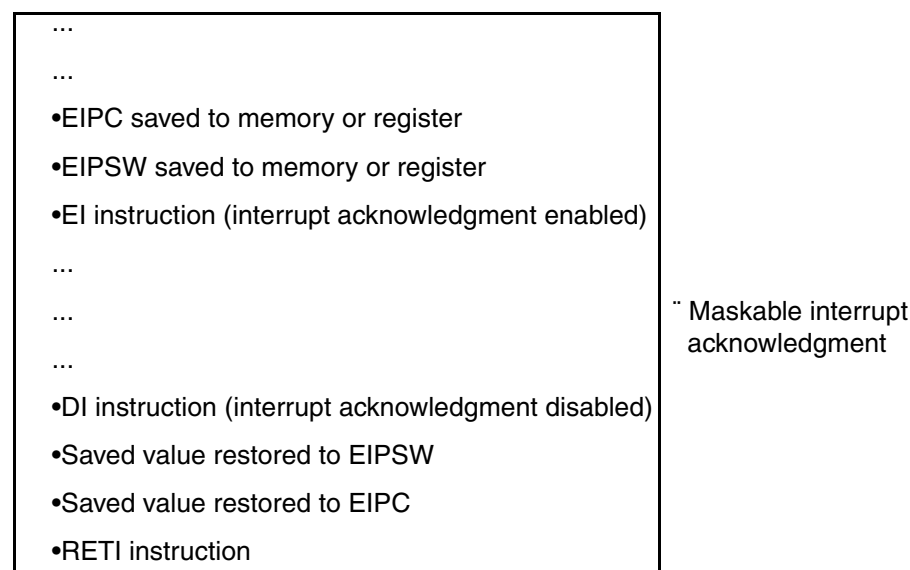
Maskable interrupt multiple processing control is executed when an interrupt has an enable status (ID = 0). Thus, if multiple interrupts are executed, it is necessary to have an interrupt enable status (ID = 0) even for an interrupt processing routine.

If a maskable interrupt enable or a software exception is generated in a maskable interrupt or software exception service program, it is necessary to save EIPC and EIPSW.

This is accomplished by the following procedure.

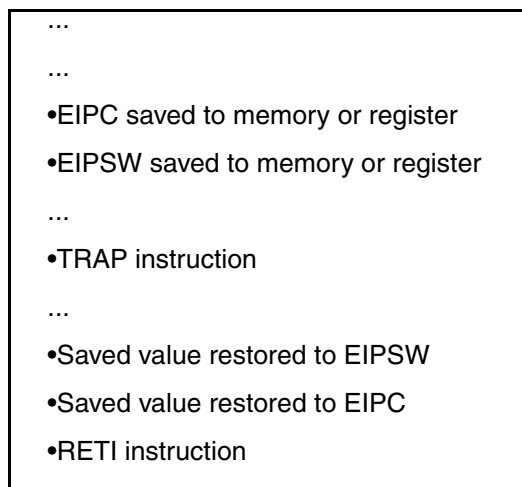
### (1) Acknowledgment of maskable interrupts in service program

Service program of maskable interrupt or exception



**(2) Generation of exception in service program**

Service program of maskable interrupt or exception



“ Exception such as TRAP instruction acknowledged.

The priority order for multiple interrupt processing control has 8 levels, from 0 to 7 for each maskable interrupt request (0 is the highest priority), but it can be set as desired via software. Setting of the priority order level is done using the PPRn0 to PPRn2 bits of the interrupt control request register (PICn), which is provided for each maskable interrupt request. After system reset, an interrupt request is masked by the PMKn bit and the priority order is set to level 7 by the PPRn0 to PPRn2 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 processing that has been suspended as a result of multiple processing control is resumed after the processing of the higher priority interrupt has been completed and the RETI instruction has been executed.

A pending interrupt request is acknowledged after the current interrupt processing has been completed and the RETI instruction has been executed.

---

**Caution** In a non-maskable interrupt processing routine (time until the RETI instruction is executed), maskable interrupts are suspended and not acknowledged.

---

## 4.9 Interrupt Response Time

The following table describes the interrupt response time (from interrupt generation to start of interrupt processing).

Except in the following cases, the interrupt response time is a minimum of 4 clocks:

- During software or hardware STOP mode
- When an external bus is accessed
- When there are two or more successive interrupt request non-sampling instructions (see “Periods in Which Interrupts Are Not Acknowledged” on page 141).
- When the interrupt control register is accessed

To input interrupt requests continuously, leave a space of at least 4 clocks between interrupt request inputs.

If a peripheral register via the NPB is accessed, the “MEM” cycle of the interrupted instruction may induce additional delay cycles, thus delaying also the fetch of the 1st instruction of the interrupt service routine.

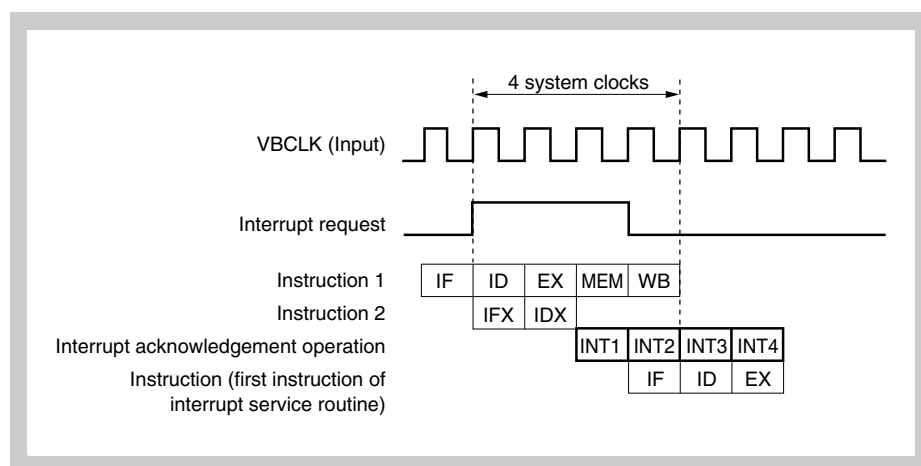


Figure 4-17 Pipeline operation at interrupt request acknowledgment (outline)

**Note** INT1 to INT4: Interrupt acknowledgment processing  
 IFx: Invalid instruction fetch  
 IDx: Invalid instruction decode

Table 4-13 Interrupt response time

Interrupt response time (internal system clocks)	Condition	
	Internal interrupt	External interrupt
Minimum	4	4 + digital noise filter delay
Maximum	8	8 + digital noise filter delay

The following cases are exceptions:

- In IDLE/software STOP mode
- External bit access
- Two or more interrupt request non-sample instructions are executed
- Access to interrupt control register



## 4.10 Periods in Which Interrupts Are Not Acknowledged

An interrupt is acknowledged while an instruction is being executed. However, no interrupt will be acknowledged between an interrupt non-sample instruction and the next instruction.

The interrupt request non-sampling instructions are as follows:

- EI instruction
- DI instruction
- LDSR reg2, 0x5 instruction (for PSW)
- The store instruction to the following registers:
  - interrupt control registers xxICn
  - interrupt in-service priority register ISPR
  - command register PRCMD
  - Timer T counter write buffer registers TTnTCW

## Chapter 5 CRC Function (CRC)

The microcontroller offers hardware support for Cyclic Redundancy Check (CRC) by its DATA-CRC unit.

### 5.1 Features

#### 5.1.1 DATA-CRC

The DATA-CRC can be used to verify or generate CRC protected data streams of arbitrary length and different bit widths.

- 32-bit Ethernet CRC (04C11DB7<sub>H</sub>)  
 $(X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X^1+1)$   
Input data reflected  
Output data reflected  
Final XOR with FFFF FFFF<sub>H</sub>
- 16-bit CCITT CRC (1021<sub>H</sub>)  
 $(X^{16}+X^{12}+X^5+1)$   
Input data not reflected  
Output data not reflected  
Final XOR with 0000<sub>H</sub>
- CRC generation to an arbitrary data block length
- One channel for DATA-CRC generation
- After initialization of the DATA-CRC input register every write access to the DATA-CRC input register generates a new CRC according to the chosen polynomial and the result is stored in the DATA-CRC data register.

## 5.2 Configuration

The following picture shows the block diagram of the CRC module.

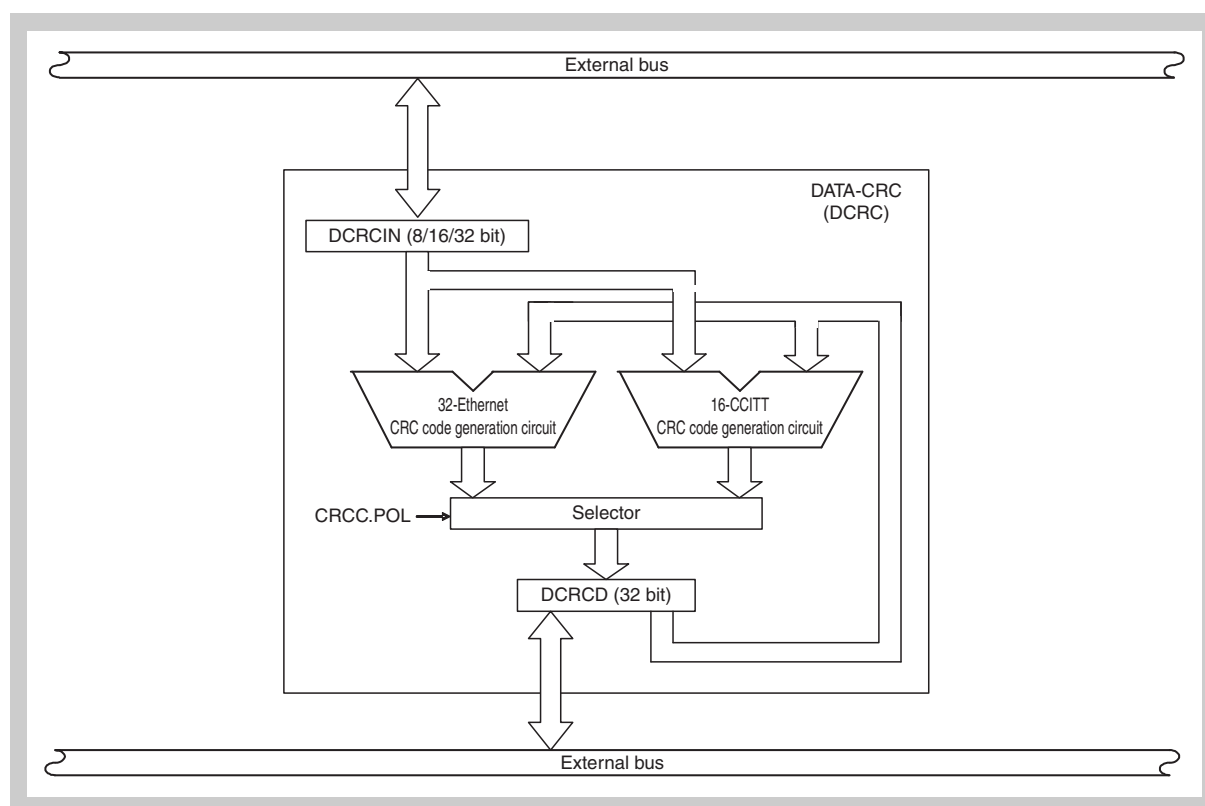


Figure 5-1 Block diagram of DATA-CRC

## 5.3 CRC Registers

**Caution** The CRC module is addressed via memory controller of the external bus, hence the chip select CS7 of the memory controller (MEMC) must be configured in advance.

The CRC is controlled and operated by means of the following registers:

**Table 5-1 CRC registers overview**

Register name	Shortcut	Address
DATA-CRC input register	DCRCIN	<base> + 30 <sub>H</sub>
DATA-CRC data register	DCRCD	<base> + 34 <sub>H</sub>
CRC control register	CRCC	<base> + 40 <sub>H</sub>

**Base address** The base address of the CRC module is 0FE0 0000<sub>H</sub>.

**(1) DCRCIN - DATA-CRC input register**

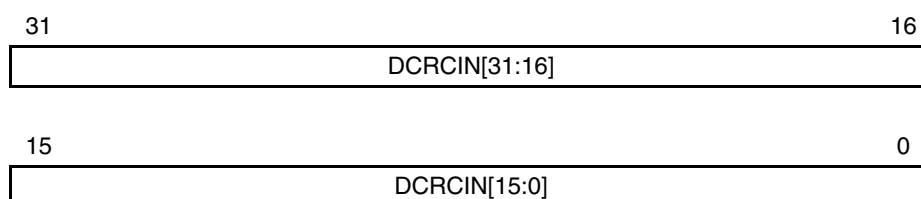
The DCRCIN register is a 32-bit data input register that holds the input data for DATA-CRC calculation. The effective bit width used for CRC calculation must be set at CRCC.ISZn bits of the CRC control register.

The CRC calculation is immediately started after the DCRCIN register is written. The DCRCD register must be initialized, with the initial starting value, before the first data of the data block is written to DCRCIN register.

**Access** This register can be read/written in 32-bit units.

**Address** <base> + 30<sub>H</sub>

**Initial Value** 0000 0000<sub>H</sub>. The register is cleared by any reset.

**(2) DCRCD - DATA-CRC data register**

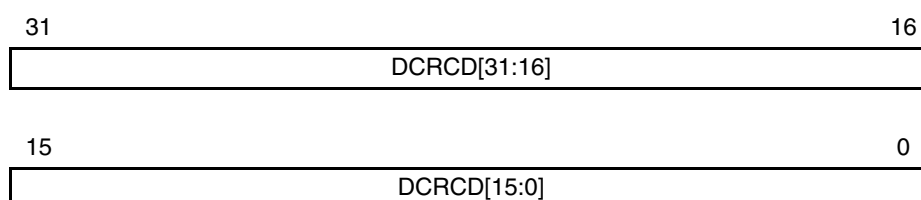
This register is a 32-bit register for the result storage of the DATA-CRC generated by the 32bit Ethernet or 16-bit CCITT polynomial.

When the 16-bit CCITT polynomial is enabled, the bits 15-0 of the DCRCD register show the CRC result. The bits 31-16 are undefined.

**Access** This register can be read/written in 32-bit units.

**Address** <base> + 34<sub>H</sub>

**Initial Value** 0000 0000<sub>H</sub>. The register is cleared by any reset.



**Caution** 1. This register must be initialized, with the initial starting value, before the first data of the data block is written to DCRCIN register.

**(3) CRCC - CRC control register**

CCRC is a register that controls the DATA-CRC generation process.

**Access** This register can be read/written in 32-bit units. The upper 16-bit are fixed to "0".

**Address** <base> + 40<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. The register is cleared by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	0	0	0	ISZ1	ISZ0	POL

ISZ1 ISZ0	Select DATA-CRC bit width
00	32-bit
01	16-bit
10	8-bit
11	setting prohibited

POL	Select DATA-CRC generating function
0	32-bit Ethernet CRC polynomial generation. The byte order of the DCRCIN register is LSB (Least Significant Byte) first, means LSB at bit position 7...0 of DCRCIN register.
1	16-bit CCITT CRC polynomial generation. The byte order of the DCRCIN register is MSB (Most Significant Byte) first, means MSB at bit position 7...0 of DCRCIN register.

**Note** After changing the DATA-CRC generating function (CRCC.POL) the DCRCD register must be initialized.

**Caution** The DATA-CRC bit width (CRCC.ISZn) must be set according to the data block bit width. Switching the DATA-CRC bit width is not allowed during processing of a data block (a data block consists of n bytes, half words or words). After the final DATA-CRC result is read from DCRCD register, the bit width can be changed and the DCRCD register must be initialized with the initial value afterwards.

## 5.4 Operation

### 5.4.1 DATA-CRC

The DATA-CRC module offers the possibility to generate a CRC of an arbitrary data block length with a data width of 8, 16 or 32bit. The CRC polynomial can either be selected for 32-bit Ethernet or 16-bit CCITT, the initial starting value must be set at the DCRCD register before the first write access to the CRC input register (DCRCIN) is done.

The flow chart below shows the DATA-CRC generating procedure.

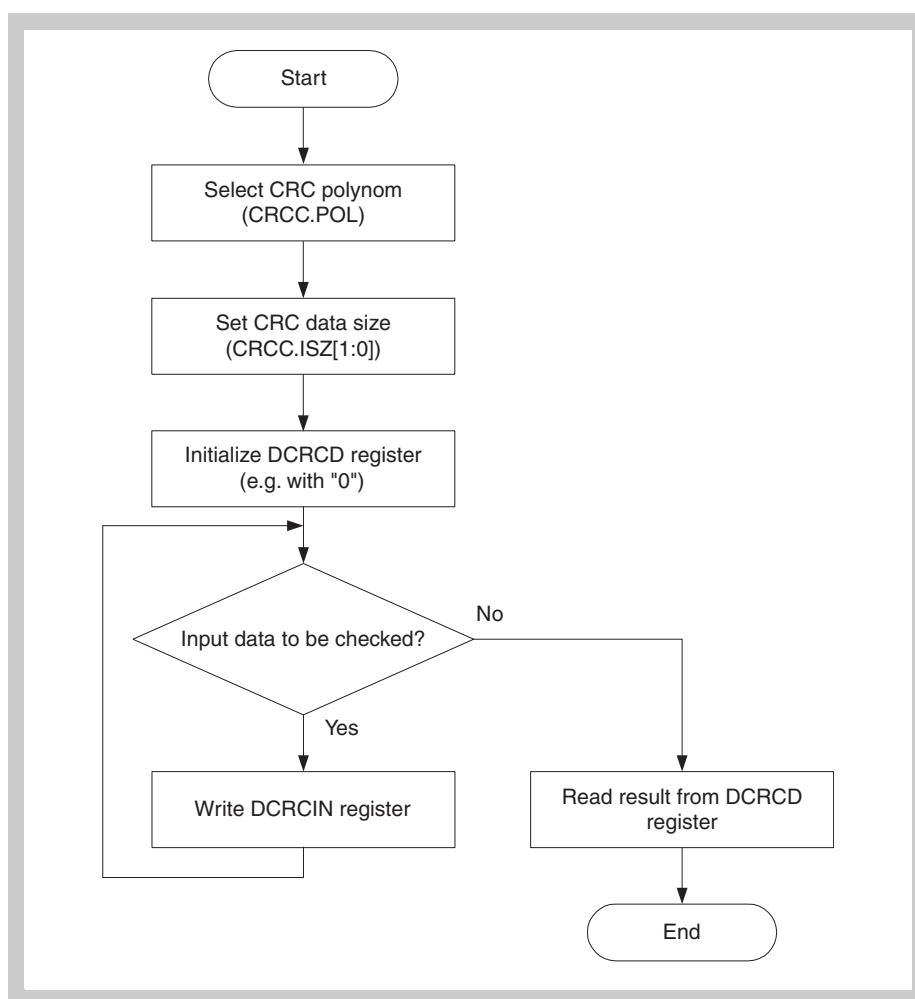


Figure 5-2 DATA-CRC flow diagram

- Note**
1. After the first data write to the input register DCRCIN, the DATA-CRC operational result is not taken into the CRC data register (DCRCD). But on read access of the DCRCD register the actual CRC value is forwarded to the register and the actual CRC value is read.
  2. The CRC data register DCRCD must be initialized if the polynomial is changed by changing bit CRCC.POL.

# Chapter 6 Clock Generator

The Clock Generator generates and monitors the internal system clock that is supplied to the CPU and the peripherals.

## 6.1 Overview

The Clock Generator has two oscillators:

- The main oscillator generates the system clock from an external 16 MHz crystal or resonator.
- An internal oscillator generates a clock that is used to monitor the main oscillator. In case of a main oscillator malfunction detection, an interrupt or a reset can be generated, while the microcontroller is operating with the free-running PLL output clock.

### 6.1.1 Features

- Crystal frequency:  $f_X = 16 \text{ MHz}$
- Multiplier function using two phase locked loop (PLL) synthesizers:
  - A times 5 multiplier generates  $f_{XX'} = 80 \text{ MHz}$ , the high-speed internal system clock for the CPU supply.
  - A times 2 multiplier generates  $f_{XX} = 32 \text{ MHz}$  for the peripheral clock supply.
- 6.8 MHz internal oscillator
- Built-in Clock Monitor
- Automatic change of CPU clock source if main oscillator fails



## 6.1.2 Description

The following figure shows a simplified block diagram of the Clock Generator:

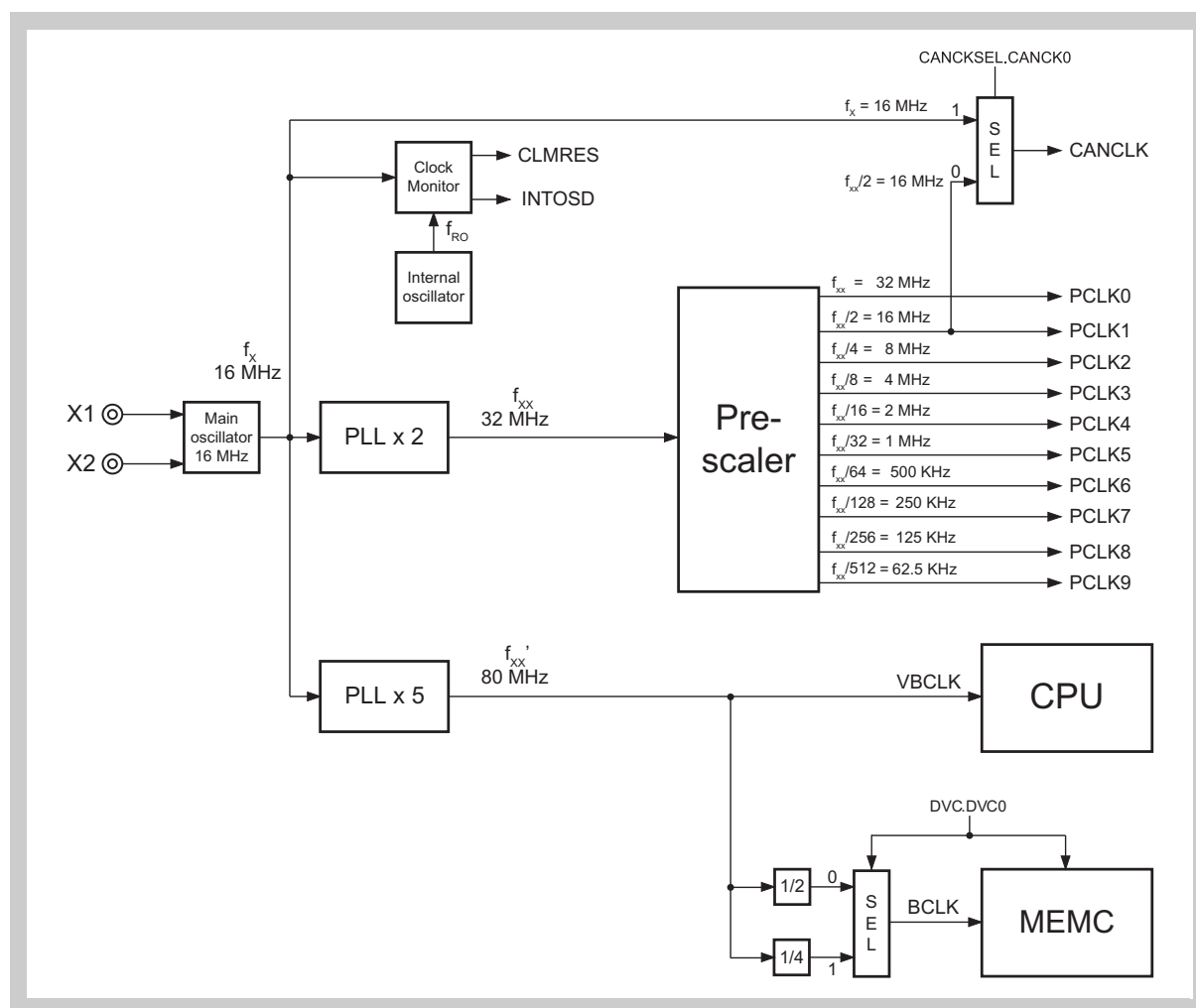


Figure 6-1 Clock Generator block diagram

<b>Note</b>	$f_x$ :	Main oscillator output clock (16 MHz)
	$f_{xx}$ :	Internal system clock (PLL $\times$ 2 output, 32 MHz)
	$f'_{xx}$ :	Internal system clock (PLL $\times$ 5 output, 80 MHz)
	$f_{RO}$ :	Internal oscillator clock
	VBCLK:	Clock to CPU (80 MHz)
	BCLK:	Clock to Memory Controller

**$f_{xx}$  PLL** An external crystal has to be connected to the X1 and X2 pins. Its frequency is multiplied by the PLL synthesizer. By this an internal system clock  $f_{xx}$  is generated that has 2 times the frequency ( $f_x$ ) of the external crystal.

The clock controller enables the PLL automatically and starts clock supply to the system after oscillation stabilization time has passed.

**$f_{xx}$ ' PLL** An additional PLL, fed with the main oscillator clock  $f_x$ , with a multiplication factor of 5 generates the 80 MHz clock  $f_{xx}$ , which is used to supply the CPU.

**VBCLK** The CPU clock VBCLK is 80MHz.

**CANCLK** The clock CANCLK for the CAN controllers can be chosen from the peripheral clock PCLK1 or the main oscillator clock  $f_x$ , both having a frequency of 16 MHz. The CANCLK choice is made by CANCKSEL.CANCK0.

**PCLK** The peripheral modules are supplied with a set of 10 peripheral clock PCLK0 to PCLK9, derived from the PLL output clock  $f_{xx}$  by a prescaler. The maximum frequency delivers PCLK0 with 32 MHz, the other PCLKn are the half of its predecessor PCLK(n-1).

## 6.2 Control Registers

The Clock Generator is controlled and operated by means of the following register:

**Table 6-1 Clock Generator registers overview**

Register name	Shortcut	Address
CAN clock selection register	CANCKSEL	FFFF F860 <sub>H</sub>

### (1) CANCKSEL - CAN clock selection register

The CANCKSEL register determines the clock source of the CAN clock CANCLK.

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

**Address** FFFF F860<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	CANCK0
R	R	R	R	R	R	R	R/W

**Table 6-2 CANCKSEL register contents**

Bit position	Bit name	Function
0	CANCK0	16 MHz CAN clock CANCLK supply selection 0: peripheral clock PCLK1 1: main oscillator clock $f_x$

## 6.3 Clock Monitor

The Clock Monitor monitors the operation of the main oscillator.

In case of malfunction of the main oscillator, it can generate the reset CLMRES or the interrupt INTOSD.

If the main oscillator has stopped to operate, the PLLs output their free-running clock frequency at  $f_{XX}$  respectively  $f_{XX}'$ . Thus the CPU system clock VBCLK and the base clock for all peripheral clocks PCLKn is derived from the free-running PLL output clocks.

---

**Caution** In case of a main oscillator stop, be aware of the following:

1. The CPU can operate on the free-running PLL output clock.
  2. DMA transfers may fail to operate.
  3. Peripheral modules, supplied by PCLKn, may fail to operate.
  4. In on-chip debug mode the Clock Monitor is automatically stopped.  
Read/write of the Clock Monitor's registers is still possible.
- 

**Emergency shut-off** The Clock Monitor interrupt INTOSD is capable to initiate an emergency shut-off (immediate change into high impedance mode) of certain output pins. For further information refer to “Pin Functions” on page 23.

### 6.3.1 Clock Monitor registers

The Clock Monitor is controlled and operated by means of the following register:

**Table 6-3 Clock Monitor registers overview**

Register name	Shortcut	Address
Clock Monitor mode register	CLM	FFFF FCA0 <sub>H</sub>

**(1) CLM - Clock Monitor mode register**

The 8-bit CLM register controls the operation of the Clock Monitor.

Writing to this register is only possible immediately after writing to the associated write protection register.

First write to the PRCMD register. The contents is ignored. Then, you are permitted to write once to the CLM register. This must be done immediately after writing to the PRCMD register. After the second write action, or if the second write action does not follow immediately, all protected registers are write-locked again.

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

**Address** FFFF FCA0<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is initialized by  $\overline{\text{RESET}}$ .

7	6	5	4	3	2	1	0
0	0	0	0	0	0	CLMD	CLME
R	R	R	R	R	R	R	R/W

**Table 6-4 CLM register contents**

Bit position	Bit name	Function
1	CLMD	Defines the action if the main oscillator frequency is too low: 0: Generate internal reset CLMRES. 1: Generate interrupt INTOSD.
0	CLME	Clock Monitor enable: 0: Disable the Clock Monitor. 1: Enable the Clock Monitor.

**Note** Once CLM.CLME has been set, it cannot be cleared by writing to the register. This bit can only be cleared by  $\overline{\text{RESET}}$ .

**Caution** CLMD must not be changed if the Clock Monitor is enabled (CLME = 1).

Since disabling of the Clock Monitor (CLME = 0) is only carried out by a  $\overline{\text{RESET}}$ , it is not possible to change CLMD after the Clock Monitor has been started.

### 6.3.2 Operation of the Clock Monitor

The Clock Monitor counts the falling edges of the main oscillator clock  $f_X$  during a period of 3 clock cycles of the half of the internal oscillator clock  $f_{RO}$  ( $= 1/T_{RO}$ ), i.e. within  $3 \times 2T_{RO}$ .

In case the number of counted  $f_X$  is less than two, the Clock Monitor issues the reset CLMRES or the interrupt INTOSD, depending on CLM.CLMD.

As the frequency of the internal oscillator is heavily temperature depend,  $f_{RO}$  varies between its minimum  $f_{ROmin}$  ( $= 1/T_{ROmin}$ ) and maximum  $f_{ROmax}$  ( $= 1/T_{ROmax}$ ) values.

**Main oscillator fail detection** *Figure 6-2 illustrates the calculation of the minimum frequency  $f_{Xmin1}$ , that the Clock Monitor always detects as a main oscillator fail condition.*

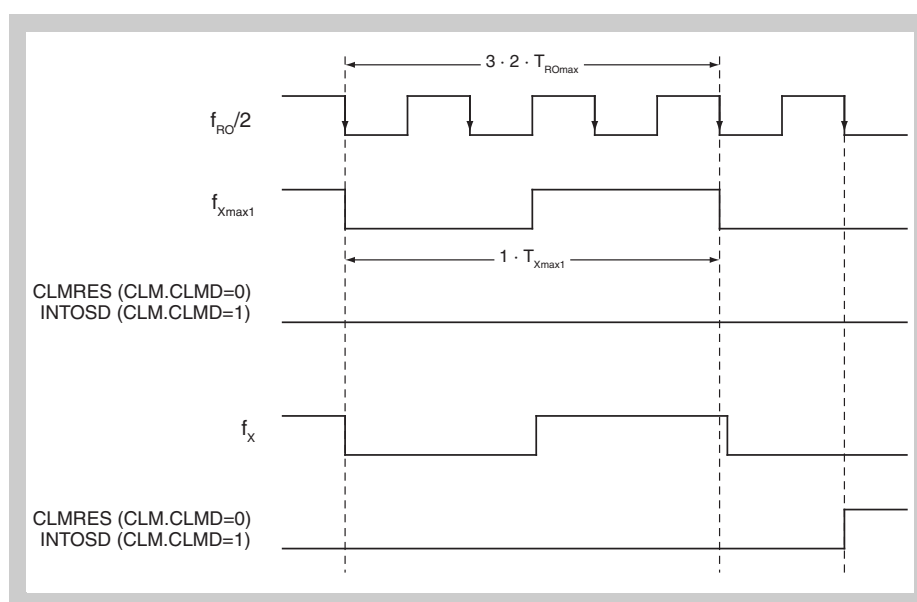
$$T_{Xmax1} = 3 \cdot 2 \cdot T_{ROmax}$$

and thus

$$f_{Xmin1} = f_{ROmin} / 6$$

All main oscillator frequencies below  $f_{Xmin1}$  are reported as main oscillator failure.

The main oscillator frequency  $f_X$  ( $= 1/T_X$ ) in *Figure 6-2* falls below this limit  $f_{Xmin1}$  and the Clock Monitor reports oscillator failure by generating CLMRES respectively INTOSD.



**Figure 6-2 Main oscillator fail detection**

**Main oscillator no-fail detection** *Figure 6-3 illustrates the calculation of the minimum frequency  $f_{Xmin2}$ , that the Clock Monitor always detects as a main oscillator no-fail condition.*

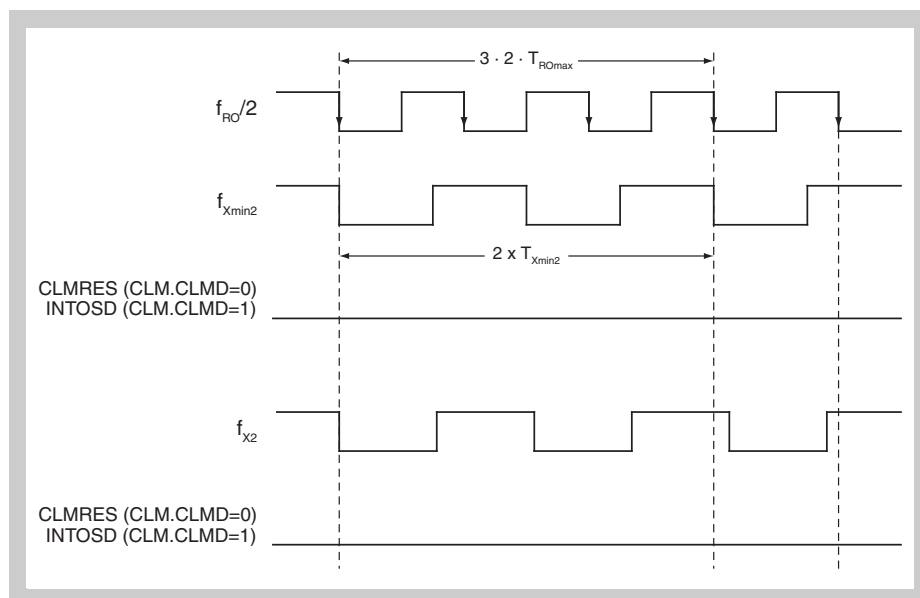
$$2 \cdot T_{Xmax2} = 3 \cdot 2 \cdot T_{ROmin}$$

and thus

$$f_{Xmin2} = f_{ROmax} / 3$$

All main oscillator frequencies above  $f_{Xmin2}$  are judged as main oscillator no-fail.

The main oscillator frequency  $f_X (= 1/T_X)$  in *Figure 6-3* falls below this limit  $f_{Xmin2}$  and the Clock Monitor reports oscillator failure by generating CLMRES respectively INTOSD.



**Figure 6-3** Main oscillator “no fail” detection

All main oscillator frequencies between  $f_{Xmin1}$  and  $f_{Xmin2}$  may be detected as oscillator failure.

*Table 6-5* summarizes the main oscillator fail/no-fail detections and gives an example with following internal oscillator frequencies:

- $f_{ROmin} = 5,88 \text{ MHz} \Rightarrow T_{ROmax} = 170 \text{ ns}$
- $f_{ROmax} = 8,00 \text{ MHz} \Rightarrow T_{ROmin} = 125 \text{ ns}$

For information concerning the real internal oscillator frequencies refer to the Data Sheet.

**Table 6-5** Main oscillator fail detection

$f_X$	Clock Monitor detection	Example
$f_X < f_{Xmin1} = f_{ROmin} / 6$	fail	$f_X < 0,98 \text{ MHz}$
$f_{Xmin1} < f_X < f_{Xmin2}$	fail or no-fail	$0,98 \text{ MHz} < f_X < 2,67 \text{ MHz}$
$f_X > f_{Xmin2} = f_{ROmax} / 3$	no-fail	$f_X > 2,67 \text{ MHz}$

**Note** Main oscillator frequencies above its nominal value  $f_X$  will not be detected as failure.

## 6.4 Power Save Control

The power save function of V850E/PHC3 supports the HALT mode. In this mode, the Clock Generator (oscillator and PLL synthesizers) continues to operate, but the CPU clock and hence program execution stops. The built-in RAM keeps the state it had before the HALT mode was set.

The clock supply to the on-chip peripherals continues. Peripheral devices that do not require instruction processing remain operating.

The HALT mode can reduce the average power consumption of the entire system by intermittent operation with the normal operation mode.

The system is switched to HALT mode by a specific instruction (the HALT instruction).

Figure 6-4 shows the transitions between normal operation mode and HALT mode.

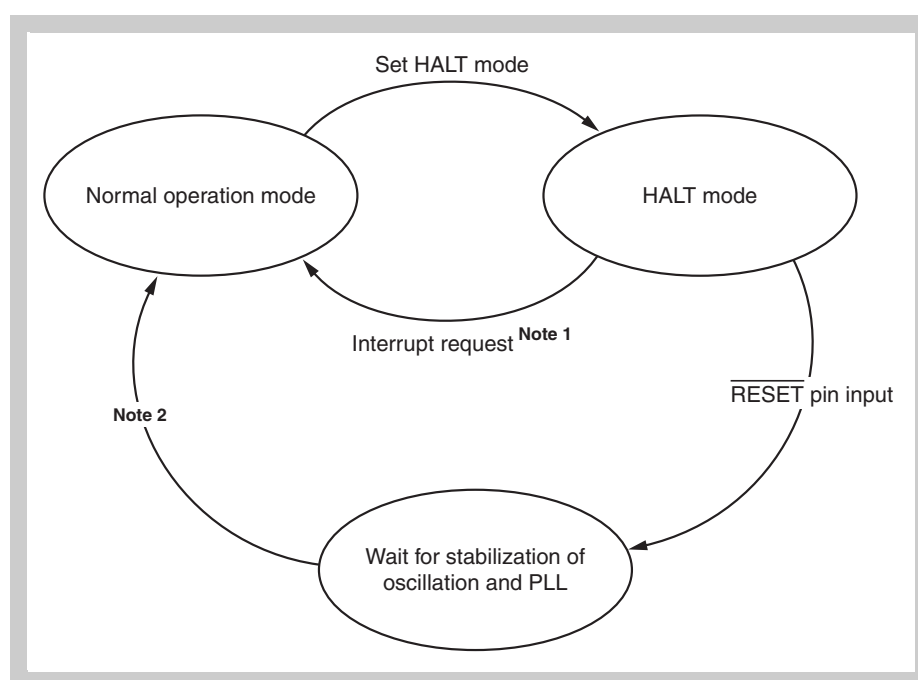


Figure 6-4 Power Save Mode State Transition Diagram

- Note**
1. Non-maskable interrupt request signal (NMI) or unmasked maskable interrupt request signal.
  2. The oscillation stabilization time is necessary after release of reset because the PLL is initialized by a reset. The stabilization time is determined by hardware.



### 6.4.1 HALT mode

#### (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 6-6 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.

- 
- Caution**
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.
- 

The following table shows the operation status in the HALT mode.

**Table 6-6 Operation Status in HALT Mode**

Function	Operation Status
Clock Generator	Operating
Internal system clock ( $f_{XX}$ )	Supplied
CPU	Stopped
DMA	Operating
Interrupt controller	Operating
Ports	Maintained
On-chip peripheral I/O (excluding ports)	Operating
Internal data	All internal data such as CPU registers, states, data, and the contents of internal RAM are retained in the state they were before HALT mode was set.

#### (2) Releasing HALT mode

The HALT mode is released by a non-maskable interrupt request signal (NMI), an unmasked maskable interrupt request signal, or any reset.

After the HALT mode has been released, the normal operation mode is restored.

### (a) Releasing HALT mode by non-maskable interrupt request signal or unmasked maskable interrupt request signal

The HALT mode is released by a non-maskable interrupt request signal or an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request. If the HALT mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is serviced as follows.

- 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.
- 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.

**Table 6-7 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

### (b) Releasing HALT mode by reset

The same operation as the normal reset operation is performed.

## Chapter 7 Flash Memory

The V850E/PHC3 microcontroller is equipped with following internal flash memory:

- Code flash:
  - $\mu$ PD70F3485: 480 KB
- Data flash: 32 KB

The code flash memory is attached to the dedicated fetch bus interface of the V850 CPU core. It is used for non-volatile storage of program code and constant data.

The data flash memory is accessible via the memory interface bus. It holds nonvolatile user's data, which are subject to be altered during normal program operation.

Flash memory is commonly used in the following development environments and applications:

- For altering software after solder-mounting of the V850 microcontroller on the target system.
- For differentiating software in small-scale production of various models.
- For data adjustment when starting mass production.
- For facilitating inventory management.
- For updating software after shipment.

The flash memory can be written in different ways:

- by a flash programmer equipped with a suitable adapter (off-board write)
- mounted on the target board by connecting a dedicated flash programmer to the target system (on-board write)
- by the V850 microcontroller's application software (self-programming)

## 7.1 Code Flash Memory Overview

### 7.1.1 Code flash memory features

- Internal code flash memory:
  - : 480 KB
- Operation speed: 80 MHz by 2-way interleaved access
  - 4-byte/1 CPU clock cycle access for consecutive instruction fetches
  - 4-byte/5 CPU clock cycles access for random instruction and data fetches
- All-blocks batch erase or single block erase
- Erase/write with single power supply
- Communication with dedicated flash programmer via two serial interfaces
- On-board and off-board programming
- Flash memory programming by self-programming
- 7-bit ECC (error correction code) for every 4 bytes
  - 1 bit error: automatic correction
  - 2 bit errors: detection
  - Specific interrupt generation in case of bit errors

### 7.1.2 Code flash memory mapping

The V850 microcontroller's internal code flash memory area is divided into blocks of 4 KB blocks and can be programmed/erased in block units. All or some of the blocks can also be erased at once.

Following figures list the block structures and address assignments of the code flash memory.

Additional information comprise:

- Boot swap cluster size  
Configurable size of boot cluster for secure self-programming, refer to *"Secure self-programming (boot cluster swapping)" on page 183.*
- Interleave  
Interleave configuration of the flash memory blocks.
- CPU branch latency  
Number of additional CPU clock cycles during instruction fetches of non-linear code.

Block 119 (4 KB)	0007 7FFFH	Address
	0007 7000H	
...	...	
	...	
Block 96 (4 KB)	0006 0FFFH	
	0006 0000H	
Block 95 (4 KB)	0005 FFFFH	
	0005 F000H	
...	...	
	...	
Block 1 (4 KB)	0000 1FFFH	
	0000 1000H	
Block 0 (4 KB)	0000 0FFFH	
	0000 0000H	
480 KB	Code flash size	
64 KB	Boot swap cluster sizes	
2-way	Interleave	
4 cycles	CPU branch latency	
	Products	

Figure 7-1 Code flash memory configuration

### 7.1.3 Code flash memory functional outline

- Serial programming** The internal flash memory of the V850 microcontroller can be rewritten by using the rewrite function of a dedicated flash programmer, regardless of whether the V850 microcontroller has already been mounted on the target system or the device is not mounted (off-board/on-board programming).
- Since there is no functional difference between on-board and off-board programming by an external flash programmer, both will be gathered as “serial programming” - in contrast “to self-programming”.
- Self-programming** The self-programming facility, which facilitates rewriting of the flash memory by the user program, is ideal for program updates after production and shipment, since no additional programming equipment is required. During self-programming some software services as well as interrupt serving can still be in operation, e.g. to sustain communication with other devices.
- While the self-programming mode can be initiated from the normal operation mode the external flash programmer mode is entered immediately after release of a system reset.
- Refer to “Flash memory programming control” on page 175 for details on how to enter normal operation or serial flash programming mode.
- Extra area** The flash memory contains an extra area, used to store the settings of security and protection functions and other flash relevant information.
- The extra area is not mapped into the CPU’s address space, thus is not directly accessible by the user’s program. The extra area’s settings can only be read and modified by an external programmer or by the self-programming library.
- Boot swap** A boot swap function makes safe re-programming of the flash memory possible and is used to maintain an operable software version, even if re-programming fails for any reason, e.g. in a power fail situation.
- For further information concerning boot swapping refer to “Secure self-programming (boot cluster swapping)” on page 183.
- Protection** A set of protection flags can be specified during flash memory programming to prohibit access the flash memory in different ways, implying read-out, rewrite and erase protections. By these means the code flash memory can be protected against read-out and rewrite of the flash memory content by unauthorized persons.
- For further information concerning data protection refer to “Data Protection and Security” on page 187.

**Table 7-1 Flash memory write methods**

Environment	Interface	Outline	Operation Mode
Serial programming	Serial I/F (UART, CSI)	Flash memory programming is done by an external flash programmer. The device may be mounted on the target system (on-board) or unmounted (off-board) by using a suitable programming adapter board. In either case the communication between the device and the flash programmer is using a serial interface. For details refer to “Flash Programming with Flash Programmer” on page 170.	Flash memory programming mode

**Table 7-1 Flash memory write methods**

Environment	Interface	Outline	Operation Mode
Self-programming	Self-programming library	Flash memory can be rewritten by executing a user program that has been written to the flash memory in advance by means of off-board/on-board programming. The self-programming library provides all necessary functions to be called by the user's software. For details refer to "Code Flash Self-Programming" on page 181.	Normal operation mode

Table 7-2 on page 164 summarizes the functions used to modify flash memory content.

**Table 7-2 Basic functions for flash memory modifications**

Function	Functional outline	Support (√: Supported, ×: Not supported)	
		Serial programming	Self-programming
Block erasure	The contents of specified memory blocks are erased.	√	√
Multiple block erasure	The contents of the specified successive multiple blocks are erased.	√	√
Chip erasure	<p>The contents of the entire memory area is erased all at once. The extra area - except the boot block cluster protection flag - is also erased.</p> <hr/> <p><b>Caution:</b> The chip erase function erases also the data flash memory.</p> <hr/>	√	× <sup>a</sup>
Write	Writing to specified addresses, and a verify check to see if write level is secured are performed.	√	√
Verify	Data read from the flash memory is compared with data transferred from the flash programmer.	√	× <sup>b</sup>
Checksum	Microcontroller internally calculated checksum over the entire flash memory content is compared with the checksum calculated by the external programmer	√	×
Blank check	The erasure status of the entire memory is checked.	√	√
Protection settings	<p>Following functions can be prohibited:</p> <ul style="list-style-type: none"> <li>• chip erase</li> <li>• block erase</li> <li>• write</li> <li>• read</li> <li>• rewriting of the boot block cluster</li> </ul>	√	√ <sup>c</sup>

- a) In self-programming mode all blocks can be specified to be erased at once by block erasure. Note that the extra area is not erased in this case.
- b) Can be carried out by the user's program.
- c) Except protection against rewriting of the boot block cluster all other protections have no effect in self-programming mode.  
Protection settings can be activated in self-programming mode. Already activate protection settings can not be deactivated.

The following table lists the available flash memory protection functions.

For details refer to "Data Protection and Security" on page 187.

### 7.1.4 Code flash memory erasure and rewrite

**Erasure** According to its block structure the flash memory can be erased in two different modes.

- All-blocks batch erasure  
All blocks are erased all together.
- Block erasure  
Each 4 KB flash memory block can be erased separately.  
In self-programming mode any number of contiguous flash memory blocks can be erased all together.



**Table 7-3 Protection functions**

Function	Functional outline	Applicable (√: applies, ×: doesn't apply)	
		Serial programming	Self-programming
Chip erase command prohibit	Erasure of the entire flash (including the extra area <sup>a</sup> and the data flash) or single blocks impossible.	√	×
Block erase command prohibit	Erasure of single blocks impossible.	√	×
Program command prohibit	Erasure and rewrite of single blocks impossible.	√	×
Read command prohibit	Read-out of any flash content impossible.	√	×
Rewriting boot area prohibit	Erasure (by block or chip erase) or writing of the boot block cluster impossible.	√	√

a) The boot block cluster protection flag is not erased.

**Rewrite** In self-programming and serial programming mode it is possible to rewrite the flash memory in smaller units than one block. Once a complete block has been erased it can be rewritten in units of 8 byte, starting at 8 byte aligned addresses, i.e. at 000<sub>B</sub> as the lower 3 address bits. Each unit can be rewritten only once after erasure of the complete block.

### 7.1.5 Code flash memory error detection

The code flash memory control circuit is equipped with an error detection function. A 7-bit ECC (Error Correction Coding) code is dedicated to each 32-bit word in the flash memory.

The ECC can detect and correct single bit errors and detect double bit errors. In the latter case the double error detection interrupt INTDEDF is generated.

INTDEDF can generate two kinds of interrupts:

- an NMI via the NMI sharing function
- a maskable interrupt (exception code 0080<sub>H</sub>), that is shared with double error detections of the RAM error detection functions

Besides generating an non-maskable or maskable interrupt, the address of the erroneous data is saved a register.

**Note** As error detection is also performed on instructions, which are prefetched by the CPU's execution pipeline, an error may be detected and reported, even if the instruction is not attempted to be executed because of a program branch.

One register is provided for the error correction function.

#### (1) ROMEAD - Flash ROM ECC error register

The 32-bit ROMEAD register holds the address of the flash memory location, where an error was detected first.

**Access** This register can be read in 32-bit units.

**Address** FFFF F8A0<sub>H</sub>

**Initial Value** 0000 0000<sub>H</sub>

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	ROMEADDR		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROMEADDR														0	0

**Table 7-4 ROMEAD register contents**

Bit Position	Bit Name	Function
18 - 2	ROMEADDR	19-bit address of first detection of a flash ROM error.

The address of the first error detection is stored in ROMEAD. At the same time the interrupt flag INTERRF.INTERR0 is set to 1.

This register is not overwritten by a new error detection until

- the INTERRF.INTERR0 is cleared to 0. Thus no new error is signaled by interrupt INTDEDF until the INTERRF.INTERR0 is cleared to 0.
- ROMEAD is cleared to 0000 0000<sub>H</sub> by any reset.

Afterwards a new ECC detection address can be stored.

---

**(2) Flash memory error detection check**

For checking the functionality of the flash memory error detection, a dedicated address in the flash memory needs to be reserved for generating a mismatch between the flash memory data word and its associated ECC.

This flash memory address needs to be programmed twice:

1. The first data written to the address generates the correct ECC
2. The second data written to the same address must differ to the first data. Consider that during re-programming only “1” bits can be changed to “0”, but not vice versa. Further the correct ECC of the second data must differ to the correct ECC of the first data.

Afterwards any read of the data (either as an instruction fetch or as a read of a constant) should generate a maskable or non-maskable flash memory error interrupt INTDEDF.

## 7.2 Data Flash Memory

The V850E/PHC3 microcontroller contains a 32 KB data flash in addition to the code flash. The on-chip data flash is connected to the memory bus controller (MEMC).

### 7.2.1 Data flash memory features

The data flash has the following features:

- 32 KB of data flash memory in 2 KB blocks
- Write access in 32-bit steps
- Erase in 2 KB blocks
- Write, erase operations to the data flash while user's code can be executed from code flash

**Note** Renesas Electronics provides sample code for a library enabling the flexible emulation of an EEPROM. By means of this library the emulated number of write/erase cycles can be extended far beyond the number of maximum write/erase cycles specified in the Data Sheet.

### 7.2.2 Data flash memory map

The on-chip data flash is connected to the memory bus controller (MEMC). The data flash is mapped to the CS1 area to address range 3E0 0000<sub>H</sub> to 3E0 7FFF<sub>H</sub>.

The memory interface has to be set up as follows:

**Table 7-5 BCU/MEMC register settings for data flash access**

Control bit	Required setting	Comment
BCT0.ME1	1	<ul style="list-style-type: none"> <li>• enable <math>\overline{CS1}</math></li> <li>• no default, must be changed</li> </ul>
AWC.AHW1 AWC.ASW1	00 <sub>B</sub>	<ul style="list-style-type: none"> <li>• no address setup/hold waits</li> <li>• no default, must be changed</li> </ul>
DWC0.DWC1[2:0]	001 <sub>B</sub>	<ul style="list-style-type: none"> <li>• 1 data wait state</li> <li>• no default, must be changed</li> </ul>
BCC.BC11	0	<ul style="list-style-type: none"> <li>• no idle states</li> <li>• no default, must be changed</li> </ul>

For further information about the memory interface configuration refer to “Bus and Memory Control (BCU, MEMC)” on page 192.

## 7.2.3 Data flash control register

### (1) DFLCTL - Data flash control register

The data flash is controlled with the data flash control register DFLCTL to enable the access to the data flash and to define the memory address location.

Writing to this register is protected by a special sequence of instructions. Please refer to “*Write Protected Registers*” on page 97.

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

**Address** FFFF FCF8<sub>H</sub>

**Initial value** 00<sub>H</sub>. This register is cleared by any reset.

7	6	5	4	3	2	1	0
DFLEN	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 7-6 DFLCTL register contents**

Bit position	Bit name	Function
7	DFLEN	Read access control of data flash: 0: Disable 1: Enable

## 7.2.4 Data flash reading

The data flash can be read via the external memory bus.

Reading of the data flash is performed with the following procedure:

1. Enable the read access to the data flash by setting the DFLCTL.DFLEN = 1.
2. Execute read operation

## 7.2.5 Data flash writing

The data flash can be written by using the data flash library or serial programming with an external flash programmer tool.

Programming during normal operation is achieved by using the data flash access layer software library. The data flash access layer is described in a separate User's Manual.

**Note** The chip erase command of an external programmer erases also the data flash.

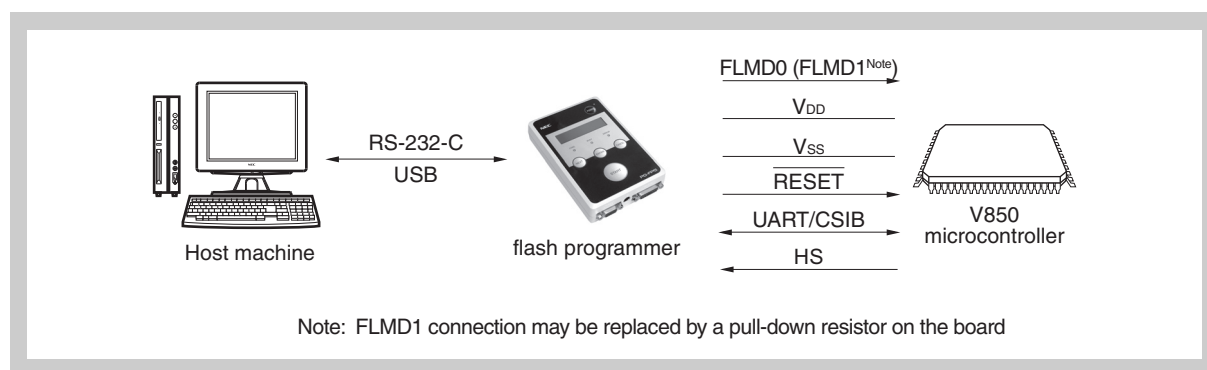
## 7.3 Flash Programming with Flash Programmer

A dedicated flash programmer can be used for external writing of the flash memory.

- On-board programming  
The contents of the flash memory can be rewritten with the V850 microcontroller mounted on the target system. Mount a connector that connects the flash programmer on the target system.
- Off-board programming  
The flash memory of the V850 microcontroller can be written before the device is mounted on the target system, by using a dedicated programming adapter.

### 7.3.1 Programming environment

The necessary environment to write a program to the flash memory of the V850 microcontroller is shown below.



**Figure 7-2 Environment to write program to flash memory**

A host machine is required for controlling the flash programmer.

Following V850 microcontroller serial interfaces can be used as the interface between the flash programmer and the V850 microcontroller:

- asynchronous serial interface - UART
- clocked serial interface - CSI

If a CSI interface is used with handshake, the flash programmer's HS signal is connected to a certain V850 port, in the following generally named as HSPORT. The port used as HSPORT for this product is given in *Table 7-8*.

Flash memory programming off-board requires a dedicated programming adapter.

In this chapter the terms UART and CSI may be used generically for the dedicated interface types and channels the V850 microcontroller provides. UART and CSI signal names are used accordingly.

### 7.3.2 Communication mode

The communication between the flash programmer and the V850 microcontroller utilizes the asynchronous serial interface UART or optionally the synchronous serial interface CSI.

For programming via the synchronous serial interface CSI without handshake and with handshake modes are supported. In the latter mode the port pin HSPORT is used for the programmer's handshake signal HS.

#### (1) UART

The external flash programmer offers various choices of available baud rates.

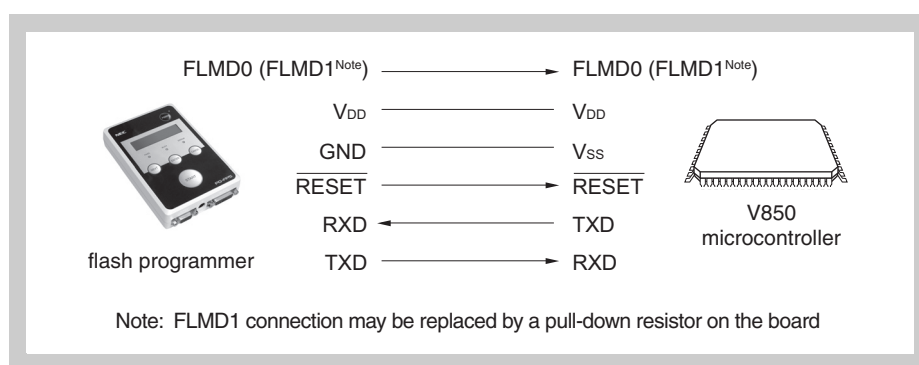


Figure 7-3 Communication with flash programmer via UART

#### (2) CSI without handshake

The external flash programmer offers various choices of available clock rates.

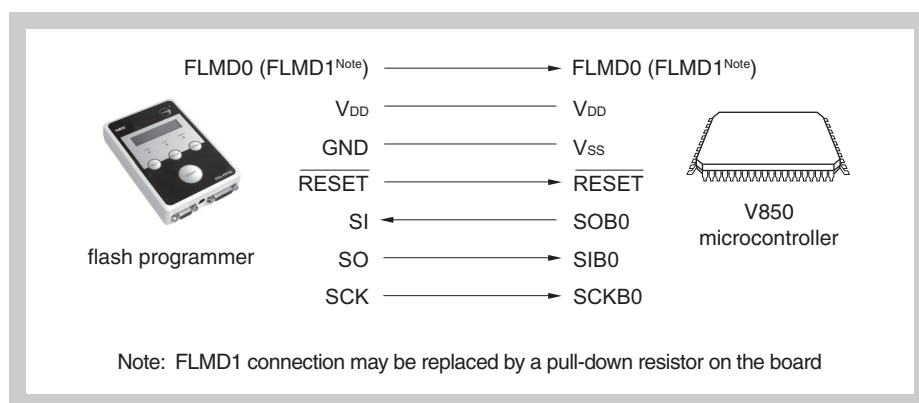
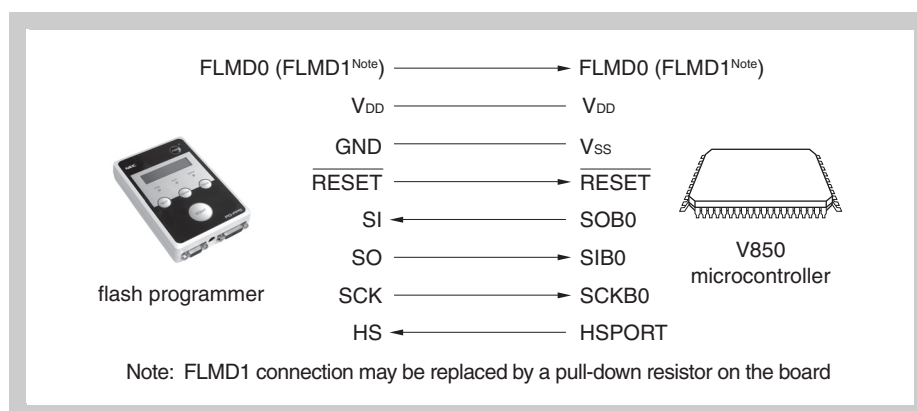


Figure 7-4 Communication with flash programmer via CSI without handshake

The flash programmer outputs a transfer clock and the V850 microcontroller operates as a slave.

**(3) CSI with handshake (CSI + HS)**

The external flash programmer offers various choices of available clock rates.



**Figure 7-5** Communication with flash programmer via CSI with handshake

The flash programmer outputs a transfer clock and the V850 microcontroller operates as a slave.

**HSPORT** The microcontroller port used for the handshake signal HS is specified in *Table 7-8 on page 174*.



### 7.3.3 Pin connection with flash programmer PG-FP5

A connector must be mounted on the target system to connect the flash programmer for on-board writing. In addition, functions to switch between the normal operation mode and flash memory programming mode and to control the V850 microcontroller's reset pin must be provided on the board.

When the flash memory programming mode is set, all the pins not used for flash memory programming are in the same status as immediately after reset.

If the PG-FP5 is used as the flash programmer, it generates the signals listed in Table 7-7 for the V850 microcontroller. For details, refer to the *PG-FP5 User's Manual (R20UT0008E)*.

**Table 7-7 Signals generated by flash programmer PG-FP5**

PG-FP5			Controller	Connection		
Signal name	I/O	Pin function	Pin name	UART	CSI	CSI + HS
FLMD0	Output	Write enable/disable	FLMD0	√	√	√
FLMD1	Output	Write enable/disable	FLMD1	×	×	×
V <sub>DD</sub>	I/O	V <sub>DD</sub> voltage generation/voltage monitor	V <sub>DD</sub>	√	√	√
GND	–	Ground	V <sub>SS</sub>	√	√	√
CLK	Output	Clock output to the controller	X1	×	×	×
$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$	√	√	√
SI/RXD	Input	Receive signal	SO/TXD	√	√	√
SO/TXD	Output	Transmit signal	SI/RXD	√	√	√
SCK	Output	Transfer clock	SCK	×	√	√
HS	Input	Handshake signal for CSI + HS communication	HSPORT <sup>a</sup>	×	×	√

<sup>a)</sup> The microcontroller port used for the handshake signal HS is specified in Table 7-8 on page 174.

**Note**    √:    Must be connected.  
               ×:    Does not have to be connected.

**Table 7-8 Wiring of V850E/PHC3 flash writing adapters for UARTC and CSIB**

Flash programmer (FG-FP5) connection pin			Name of FA board pin	UARTC0	CSIB0 + HS	CSIB0
Signal name	I/O	Pin function		Pin name	Pin name	Pin name
SI/RxD	I	Receive signal	SI	TXDC0	SOB0	
SO/TxD	O	Transmit signal	SO	RXDC0	SIB0	
SCK	O	Transfer clock	SCK	Not needed	SCKB0	
CLK	O	Clock to V850 microcontroller	X1	Leave open		
			X2	Leave open		
RESET	O	Reset signal	RESET	RESET		
FLMD0	I	Write voltage	FLMD0	FLMD0		
FLMD1	I	Write voltage	FLMD1	FLMD1		
HS	I	Handshake signal for CSI + HS	HS	Not needed	HSPORT = P110	Not needed
VDD	–	VDD voltage generation/ voltage monitor	VDD	V <sub>DD</sub>		
				V <sub>DD1</sub>		
				BV <sub>DD</sub>		
				EV <sub>DD</sub>		
				AV <sub>REF0</sub>		
				AV <sub>REF1</sub>		
GND	–	Ground	GND	V <sub>SS</sub>		
				V <sub>SS1</sub>		
				BV <sub>SS</sub>		
				EV <sub>SS</sub>		
				AV <sub>SS</sub>		
				AV <sub>SS1</sub>		

**Table 7-9 V850E/PHC3 pin numbers for serial programming**

Pin name	Port	V850E/PHC3 pin no.
TXDC0	P31	32
RXDC0	P30	31
SIB0	P40	71
SOB0	P41	72
SCKB0	P42	73
$\overline{\text{RESET}}$	–	34
FLMD0	–	82
FLMD1	–	81
HS/P110	P110	87

### 7.3.4 Flash memory programming control

The procedure to program the flash memory is illustrated below.

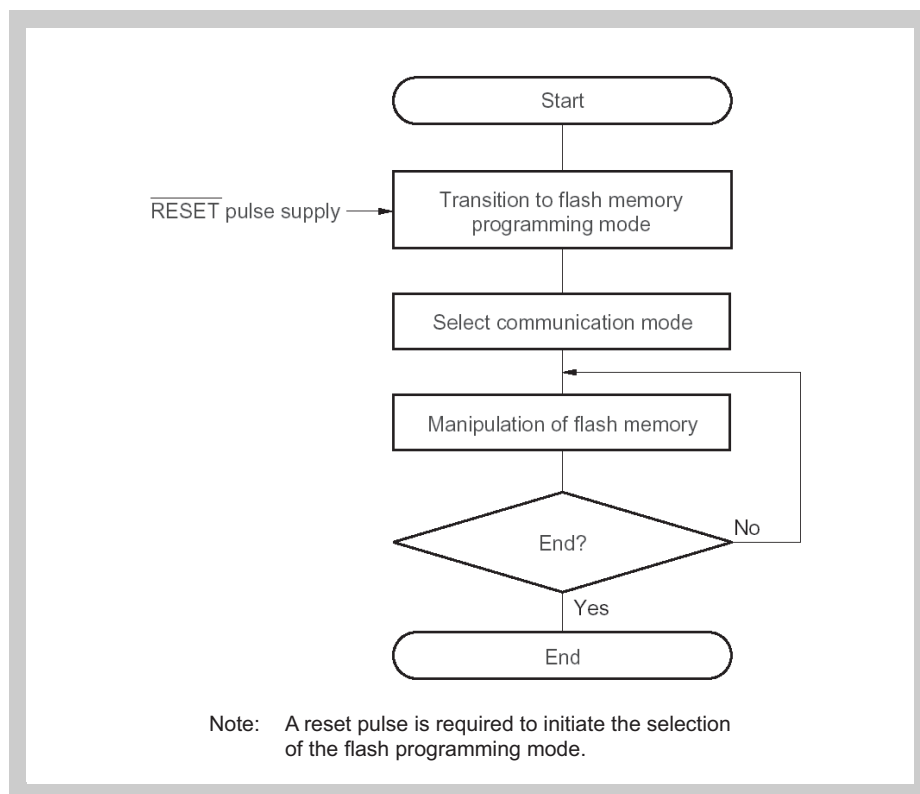


Figure 7-6 Flash memory programming procedure

**(1) Operation mode control**

To rewrite the contents of the flash memory by using the flash programmer, set the V850 microcontroller in the flash memory programming mode.

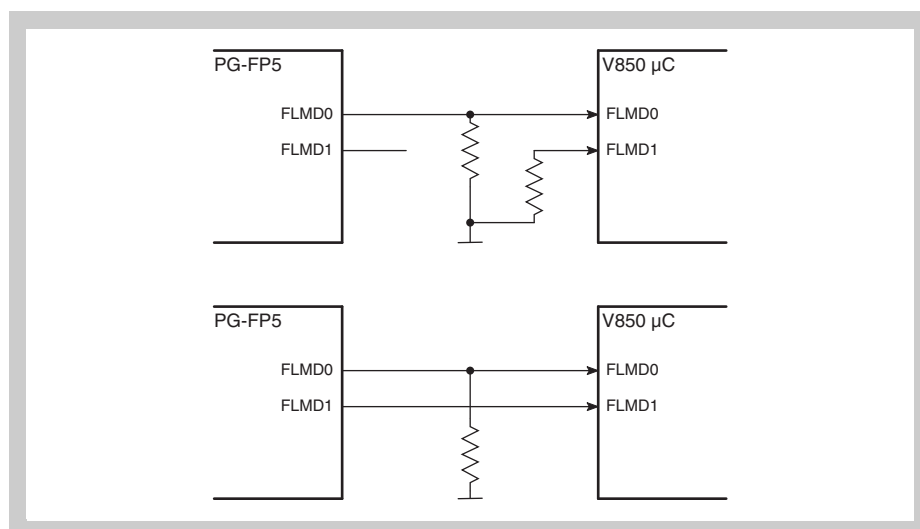
To set this mode, set the FLMD0 and FLMD1 pins as shown in *Table 7-10 on page 176* and release  $\overline{\text{RESET}}$ .

In the normal operation mode, 0 V is input to the FLMD0 pin. A pull-down resistor at the FLMD0 pin ensures normal operation mode if no flash programmer is connected. In the flash memory programming mode, the  $V_{DD}$  write voltage is supplied to the FLMD0 pin. Additionally the FLMD1 pin has to hold 0 V level.

**Table 7-10 Operation mode setting**

Pins		Operation mode
FLMD0	FLMD1	
$V_{SS}$	$V_{SS}$	Normal operation mode
$V_{DD}$	$V_{SS}$	Flash programming mode
	$V_{DD}$	Setting prohibited

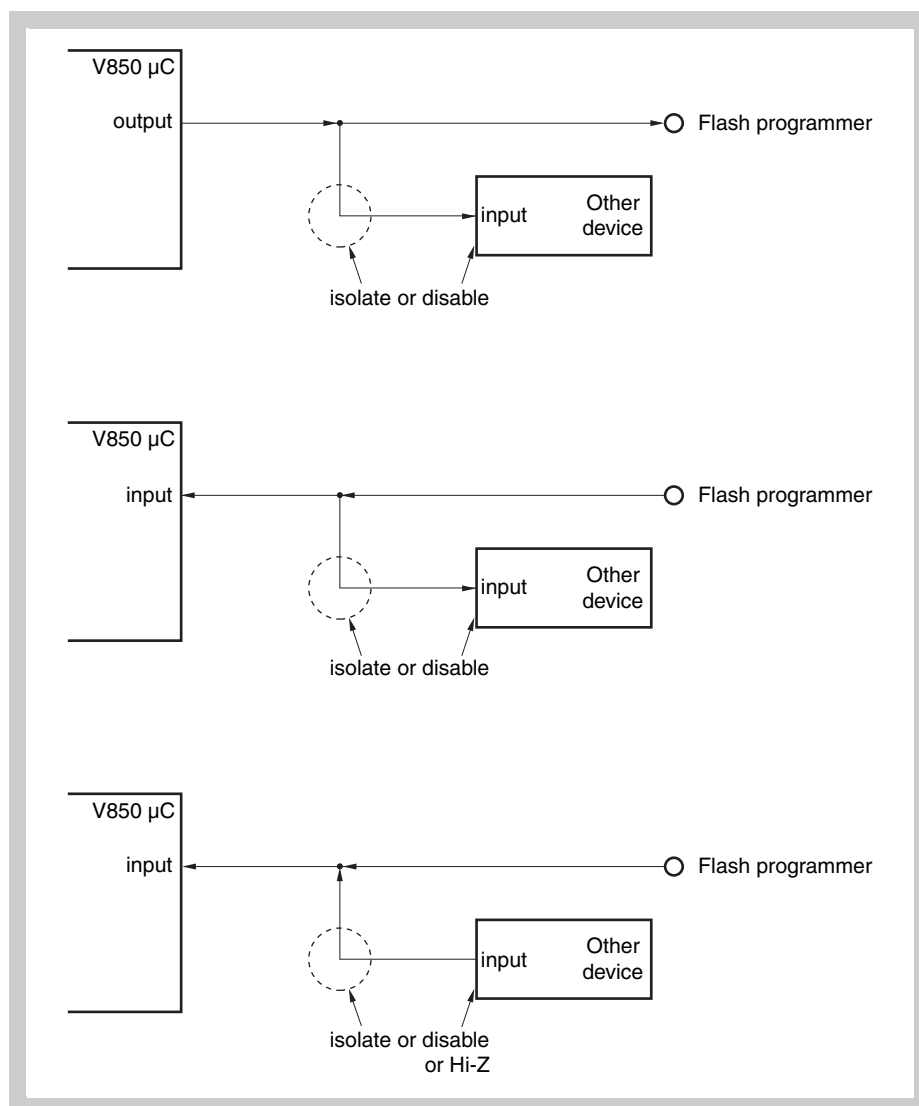
An example of connection of the FLMD0 and FLMD1 pins is shown below. FLMD1 can be connected to ground via a resistor. Alternatively the FLMD1 pin may also be connected directly to the FLMD1 signal of the flash programmer.

**Figure 7-7 Example of connection to flash programmer PG-FP5**

Once started in normal operation mode (FLMD0 = 0), FLMD0 pin is used for enabling self-programming. Refer also to “Code Flash Self-Programming” on page 181.

**(2) Potential conflicts with on-board signal connections****Serial I/O signals**

If other devices are connected to the serial interface pins in use for flash memory programming in on-board programming mode take care that the concerned signals do not conflict with the signals of the flash programmer and the V850 microcontroller. Output pins of the other devices must be isolated or set in high impedance state. Ensure that the other devices do not malfunction because of flash programmer signals.



**Figure 7-8** Potential conflicts with serial interfaces signals

**RESET** Pay attention in particular if the flash programmer's  $\overline{\text{RESET}}$  signal is connected also to an on-board reset generation circuit. The reset output of the reset generator may ruin the flash programming process and may need to be isolated or disabled.

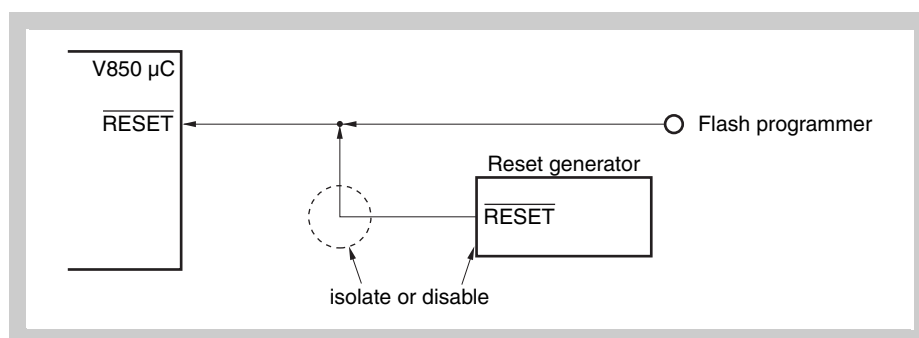


Figure 7-9 Potential conflict with  $\overline{\text{RESET}}$

**Ports** The V850 port pins adopts following status during serial programming:  
Ports used for programming are configured as UART respectively CSI pins.

All other pins remain in their default state after reset release.

In case the default state after reset of the pins not used for programming is inport port or high -impedance output port, pay attention to other devices connected to these pins. If these devices require defined levels at the pins, the ports may have to be connected to  $V_{DD}$  or  $V_{SS}$  via a resistors.

**Oscillators** Connect all oscillator pins in the same way as in the normal operation mode.

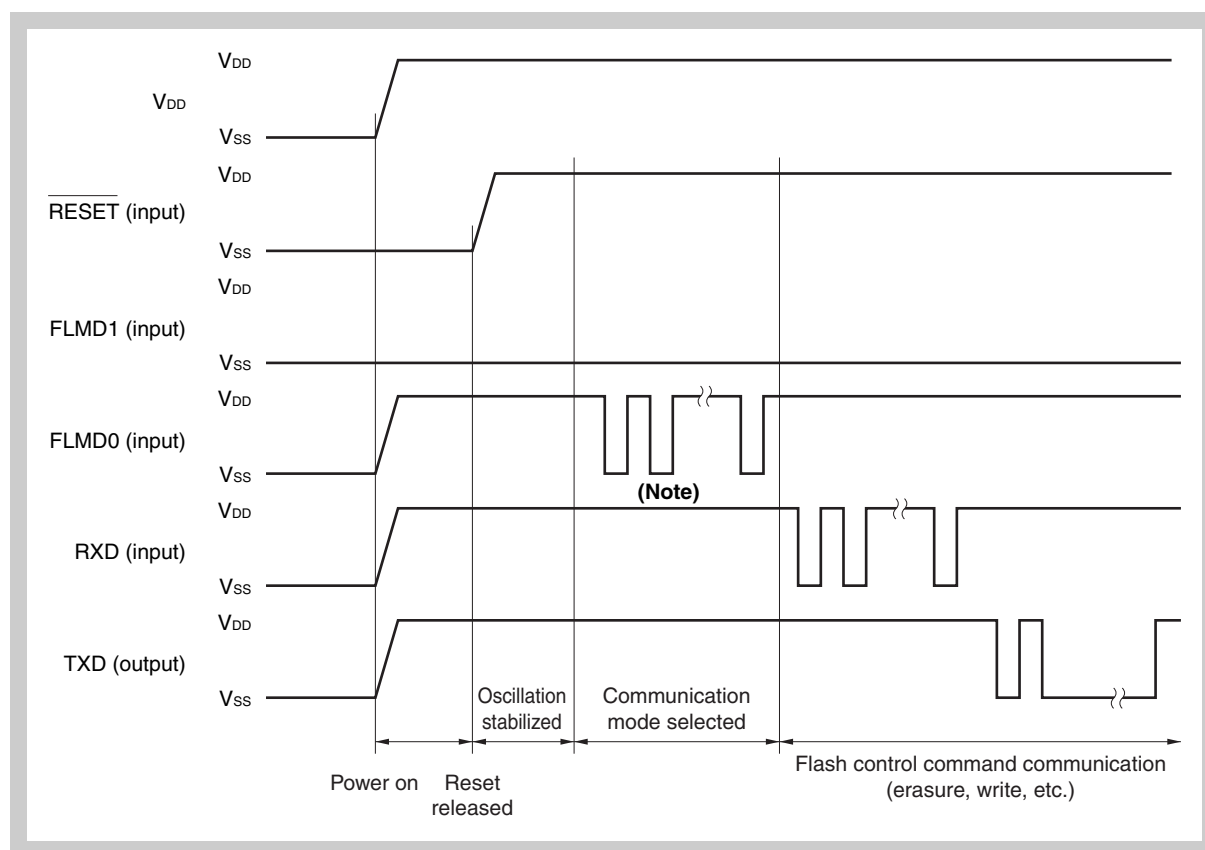
**DRST** During flash memory programming, input a low level to  $\overline{\text{DRST}}$  or leave it open. Do not input a high level.

**Power supply** Supply the same power to all power supply pins, including reference voltages, power regulator pins, etc., as in the normal operation mode.

### (3) Selection of the communication mode

The communication interface is chosen by applying a specified number of pulses to the FLMD0 pin after reset release. Note that this is handled by the flash programmer.

Figure 7-10 on page 179 gives an example how the UART is established for the communication between the flash programmer and the V850 microcontroller.



**Figure 7-10** Selection of communication mode

**Note** The number of clocks to be inserted differs depending on the chosen communication mode. For details, refer to *Table 7-11 on page 179*.

**Table 7-11** FLMD0 pulses for communication mode setting

FLMD0 pulses	Communication Mode	Remarks
0	UART	Communication rate: 9600 bps (after reset), LSB first
8	CSI	V850E/PHC3 performs slave operation, MSB first
11	CSI + HS	V850E/PHC3 performs slave operation, MSB first
Other	—	Setting prohibited

When UART has been selected after reception of the FLMD0 pulses with 9600 bps, the flash programmer changes the baud rate according to the user's choice via the flash programmer's user interface.

At first the programmer sends two 00<sub>H</sub> bytes, which are used by the microcontroller to measure the baud rate and to set up its own baud rate accordingly.

#### (4) Communication commands

The flash programmer sends commands to the V850 microcontroller. Depending on the commands, the V850 microcontroller returns status information or the requested data.

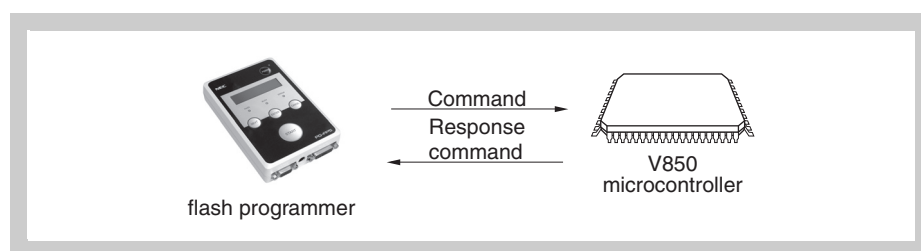


Figure 7-11 Communication commands exchange

The following table lists the flash memory control commands of the V850 microcontroller. All these commands are issued by the flash programmer, and the V850 microcontroller performs the corresponding processing.

Table 7-12 Flash memory control commands

Classification	Command name	Support			Function
		CSIB	CSIB + HS	UARTA	
Blank check	Block blank check command	√	√	√	Checks erasure status of entire memory.
Erase	Chip erase command	√	√	√	Erases all memory contents.
	Block erase command	√	√	√	Erases memory contents of specified block.
Write	Write command	√	√	√	Writes data by specifying write address and number of bytes to be written, and executes verify check.
Verify	Verify command	√	√	√	Compares input data with all memory contents.
System setting and control	Reset command	√	√	√	Escapes from each status.
	Oscillation frequency setting command	√	√	√	Sets oscillation frequency.
	Baud rate setting command	—	—	√	Sets baud rate when UART is used.
	Silicon signature command	√	√	√	Reads silicon signature information.
	Version acquisition command	√	√	√	Reads version information of device.
	Status command	√	√	—	Acquires operation status.
	Protection setting command	√	√	√	Sets protection against chip erasure, block erasure, and writing.



## 7.4 Code Flash Self-Programming

This V850 microcontroller supports a flash macro service that allows the user program to rewrite the internal flash memory by itself.

By using this flash macro service and a self-programming library, provided by Renesas Electronics, the user's program is able to rewrite the flash memory with data, transferred in advance to the internal RAM.

Thus the user program can be upgraded and constant data can be rewritten in the field.

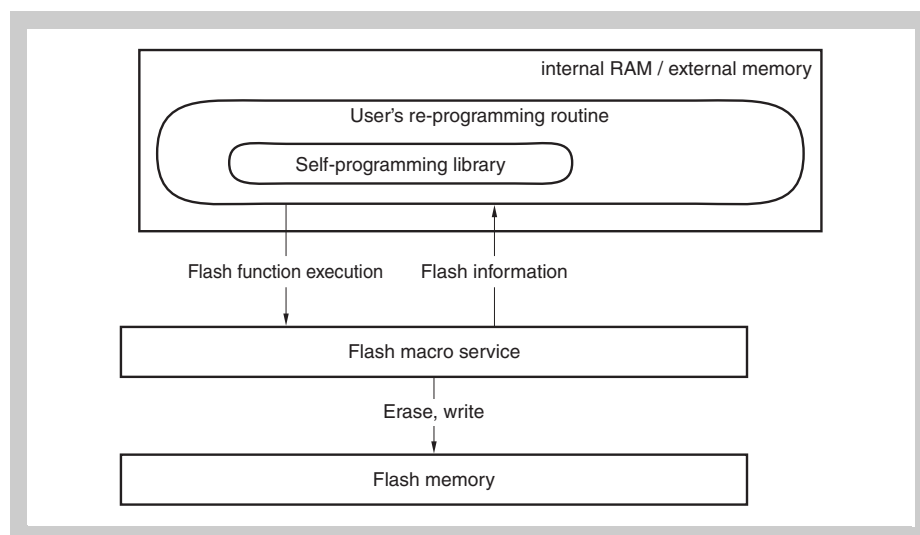


Figure 7-12 Concept of self-programming

During self-programming access to the flash memory is not possible. Thus program execution is only possible by instruction fetching from internal RAM.

Consequently the instructions of user re-programming software routines, which shall remain in operation during the self-programming procedure, must be copied from the flash memory to the internal RAM prior to activating the self-programming. Since interrupt processing by using the interrupt vectors in the flash memory is also impossible during self-programming, a special feature is provided to re-route interrupt acknowledges to the internal RAM (refer to *"Interrupt handling during flash self-programming"* on page 186).

It is recommended to refer to the *Application Note "Self-Programming"* (document no. U16929EE) for comprehensive information concerning flash self-programming. This document explains also the functions of the self-programming library. The latest version of this document and the library can be loaded via the URL

<http://www.renesas.eu>

**Caution** The self-programming operation employs also the internal Firmware, which makes use of 9 KB of the internal RAM in the address range FFFF CC00<sub>H</sub> to FFFF EFFF<sub>H</sub>. Thus this RAM area is overwritten and must be re-initialized after completion of the self-programming. Consider also that the stack occupies additionally up to 1 KB of the RAM.

### 7.4.1 Self-programming enable

The self-programming functions can be started out of the normal user mode of the V850 microcontroller.

The V850 microcontroller must be set into self-programming mode via the self-programming library.

For security reasons writing and erasing of the flash memory must be additionally permitted by setting the external FLMD0 pin to high level. Note that FLMD0 holds low level in normal operation mode after reset release.

This requires some external components or wiring, e.g. connecting an output port to FLMD0.

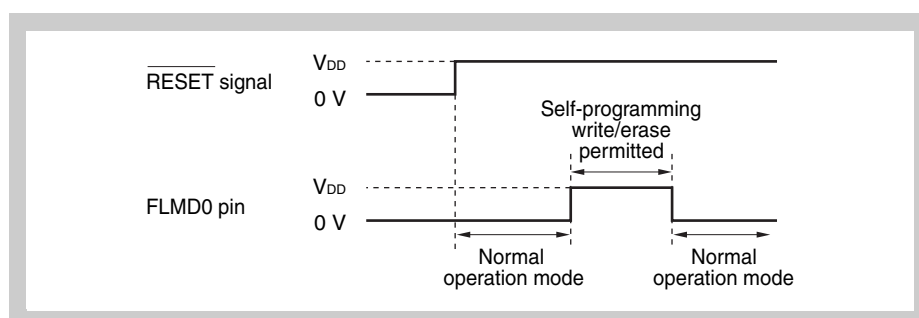


Figure 7-13 Self-programming enable

When self-programming has been completed, the voltage on the FLMD0 pin must be returned to 0 V.

### 7.4.2 Self-programming library functions

Code flash memory self-programming by the user's program is supported by the self-programming library.

This library provides a set of C function calls to carry out basic functions like

- blank-check/erase/rewrite/verify of the flash
- boot cluster swapping, including definition of boot block clusters
- setting of protection flags
- obtain various information concerning the code flash memory

Detailed information how to use the library functions is given in the *Application Note "Self-Programming Library for embedded Single Voltage FLASH"* (document no. U16929EE).

The up-to-date version of the self-programming library and the above mentioned Application Note can be obtained from

<http://www2.renesas.eu/products/micro/download/>.

### 7.4.3 Secure self-programming (boot cluster swapping)

The V850 flash microcontrollers support a mechanism to swap a cluster of code flash memory blocks, starting from address 0000 0000<sub>H</sub>, with another cluster of the same size, located immediately above the first one.

**Boot swap cluster** The cluster of blocks starting at address 0000 0000<sub>H</sub> is named active boot swap cluster, since it contains the entry point of the user's program at the default reset vector 0000 0000<sub>H</sub>.

**Boot swap flag** Which of the two clusters is the active boot block cluster is controlled by the boot swap flag, that can be defined during flash programming via the self-programming library.  
The boot swap flag is stored in the flash memory extra area.

Figure 7-14 on page 183 shows an example of the boot block swapping function with a cluster size of 4 flash memory blocks. After inverting the boot\_flag - it becomes not(boot\_flag) - blocks 4 to 7 become the active boot block cluster. Thus after next reset release the user's program starts from the new boot swap cluster.

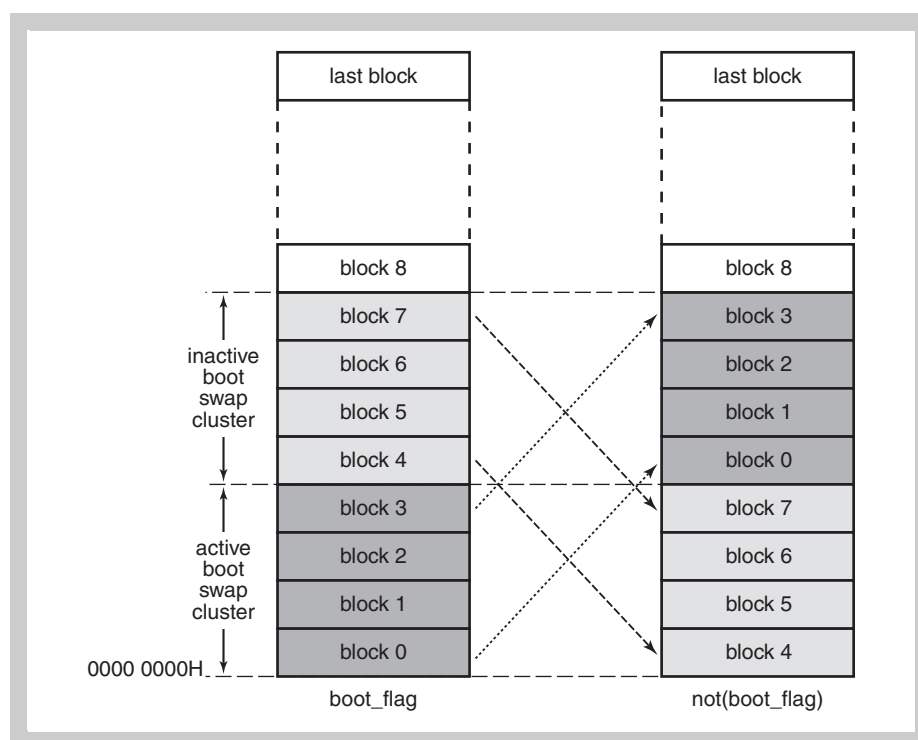


Figure 7-14 Boot swap cluster swapping function

**Secure self-programming** The boot cluster swapping function enables secure self-programming. In case the boot code shall be rewritten, the new code can be written to the inactive boot block cluster, while the boot\_flag remains in its previous state. If rewriting of the boot block cluster has been completed successfully, the boot\_flag can be inverted, making the new boot code active. If rewriting of the new boot code fails for any reason, e.g. power fail or unintended reset, the old boot code still remains active and rewriting can be started again.

**Boot block cluster** The boot code size itself may be smaller than the boot swap cluster size. The number of flash memory blocks, which are part of the boot code, are named boot block cluster. The number of boot blocks, which are member of the cluster, can be defined during self-programming via the self-programming library. The boot block cluster size of the V850E/PHC3 is fixed to 64 KB, i.e. it comprises 16 blocks.

Table 7-13 on page 184 shows the relation between the number of boot blocks, the boot block cluster size and the boot swap cluster.

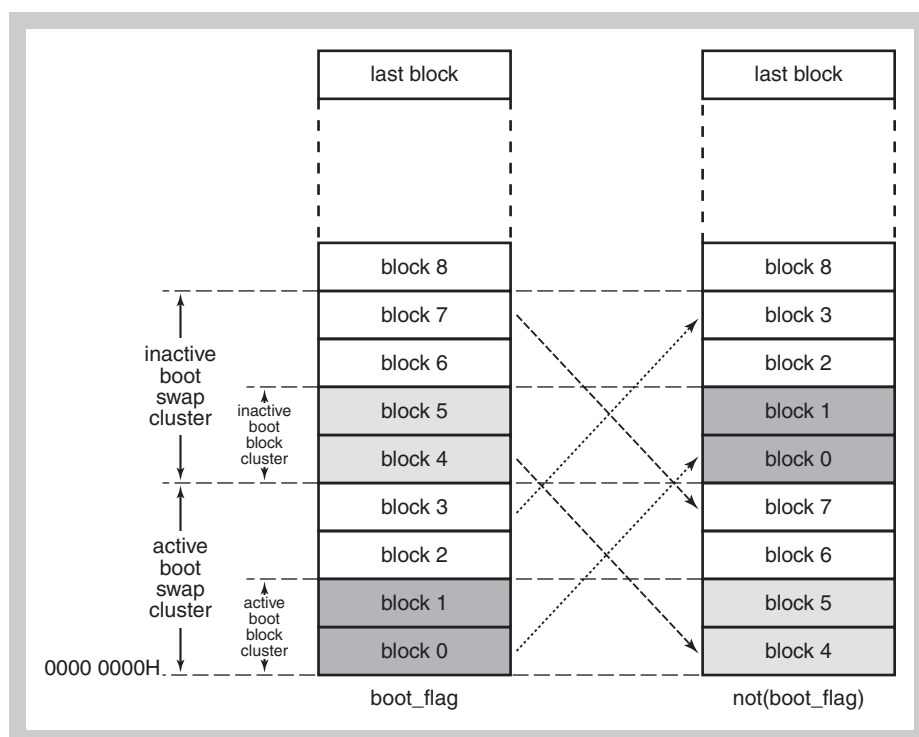
**Table 7-13 Relation between boot block and boot swap cluster**

Number of boot blocks	Boot block cluster	Boot swap cluster
00 <sub>H</sub>	0000 0000 <sub>H</sub> - 0000 0FFF <sub>H</sub> (4 KB)	0000 0000 <sub>H</sub> - 0000 FFFF <sub>H</sub> (64 KB)
01 <sub>H</sub>	0000 0000 <sub>H</sub> - 0000 1FFF <sub>H</sub> (8 KB)	
02 <sub>H</sub>	0000 0000 <sub>H</sub> - 0000 2FFF <sub>H</sub> (12 KB)	
03 <sub>H</sub>	0000 0000 <sub>H</sub> - 0000 3FFF <sub>H</sub> (16 KB)	
04 <sub>H</sub>	0000 0000 <sub>H</sub> - 0000 4FFF <sub>H</sub> (20 KB)	
...	...	
07 <sub>H</sub>	0000 0000 <sub>H</sub> - 0000 7FFF <sub>H</sub> (32 KB)	
08 <sub>H</sub>	0000 0000 <sub>H</sub> - 0000 8FFF <sub>H</sub> (36 KB)	
...	...	
0F <sub>H</sub>	0000 0000 <sub>H</sub> - 0000 FFFF <sub>H</sub> (64 KB)	
10 <sub>H</sub>	Setting prohibited	
...		
7F <sub>H</sub>		

**Number of boot blocks** The number of boot blocks has to be defined by the user during self-programming. It determines the blocks, which are subject to the boot block cluster protection, that allows to protect the boot blocks from any erase or write process.

Figure 7-15 on page 185 illustrates an example with following settings:

- number of boot blocks: 2 (boot block cluster contains 2 blocks), thus the active boot block cluster comprises
  - if boot\_flag: blocks 0 and 1
  - if not(boot\_flag): blocks 4 and 5
- active boot swap clusters comprises
  - if boot\_flag: blocks 0 to 3
  - if not(boot\_flag): blocks 4 to 7



**Figure 7-15** Boot cluster swapping function

**Boot block protection** To prohibit rewriting of the boot blocks, the boot block cluster protection flag can be set during flash memory programming. When this flag is set, the blocks of the active boot block cluster can neither be erased nor written. Boot cluster swapping is impossible as well.

Note that only the blocks of the active boot block cluster are protected. In the example according to Figure 7-15 on page 185, for instance, blocks 0 and 1 would be prohibited, while blocks 2 and 3 could still be erased and written.

- Caution**
1. Once the boot block cluster protection has been activated, it can never be deactivated again.
  2. When using boot block cluster protection, do not apply the block erase function to blocks starting with or including address 00080000H. Otherwise additional flash blocks not being part of the boot block cluster itself may be protected unintentionally.

For further information concerning flash memory protection flags refer to “Data Protection and Security” on page 187.

### 7.4.4 Interrupt handling during flash self-programming

This V850 microcontroller provides functions to maintain interrupt servicing during the self-programming procedure.

Since neither the interrupt vector table nor the interrupt handler routines, which are normally located in the code flash memory, are accessible while self-programming is active, interrupt acknowledges have to be re-routed to non-flash memory, i.e. to the internal RAM.

Therefore two prerequisites are necessary to enable interrupt servicing during self-programming:

- The concerned interrupt handler routine needs to be copied to the internal RAM. The user has to initiate this copy process.
- The concerned interrupt acknowledge has to be re-routed to that handler. Re-routing to the handler is done by the internal firmware. Thus the user doesn't have to care about.

The internal firmware and the self-programming library provide functions to initialize and process such interrupts.

The interrupt handler routines can be copied from flash to the internal RAM by use of self-programming library functions.

The addresses of the interrupt handler routines are set up via the self-programming library as well.

- Note**
1. Note that this special interrupt handling adds some interrupt latency time.
  2. Special interrupt handling is done only during the flash programming environment is activated. If self-programming is deactivated, the normal interrupt vector table in the flash memory is used.

All interrupt vectors are relocated to one entry point in the internal RAM:

- New entry point of *all* maskable interrupts is the 1st address of the internal RAM. A handler routine must check the interrupt source. The interrupt request source can be identified via the interrupt/exception source register ECR.EICC (refer to “System register set” on page 75)
- New entry point of *all* non maskable interrupts is the word address following the maskable interrupt entry, i.e. the second address of the internal RAM. The interrupt request source can be identified via the interrupt/exception source register ECR.FECC (refer to “System register set” on page 75).

In general a jump to a special handler routine will be placed at the 1st and 2nd internal RAM address, which identifies the interrupt sources and branches to the correct interrupt service routine.

The function serving the interrupt needs to be compiled as an interrupt function (i.e. terminate with a RETI instruction, save/restore all used registers, etc.).

It is recommended to read the Application Note “Self-Programming” (document no. U16929EE) for comprehensive information concerning flash self-programming. This document explains also the functions of the self-programming library. The latest version of this document can be loaded via the URL

<http://www2.renesas.eu/products/micro/download>

# Chapter 8 Data Protection and Security

## 8.1 Overview

The microcontroller supports various methods for securing safe (re-)programming of the internal flash memory and protection of the flash memory data against undesired access, such as illegal read-out or illegal reprogramming.

**Security functions** Security functions enhance the reliability of the flash operation during normal operation and support countermeasures against unexpected failures during reprogramming processes. These are basically:

- Secure self-programming
- Secure bootloader update
- Boot block cluster protection

These functions are described in detail in *“Flash Memory” on page 159*.

**Protection functions** Protection functions provide a set of mechanisms to protect the internal flash memory data from being read, erased or altered by unauthorized persons. These are basically:

- On-chip debug interface (N-Wire) protection
- Flash memory erase/write/read protection via the serial programming interface

Certain interfaces offer access to the internal flash memory: N-Wire debug interface, external flash programmer interfaces and self-programming facilities. All of these interfaces need to be considered for a proper protection concept.

The following sections give an overview about supported protection methods.

## 8.2 N-Wire Debug Interface Protection

In general, read-out of the flash memory contents is possible via the N-Wire debug interface, but protection against illegal read-out can be enabled. For protection of the flash memory, the usage of the debug interface can be protected and it can be disabled. The debug interface is protected via a 10-byte ID code and an internal flag (N-Wire use enable flag).

When the debugger is started, the status of a flag is queried (N-Wire use enable flag). Set this flag to zero to disable the use of the N-Wire in-circuit emulator.

When debugging is enabled (N-Wire use enable flag is set), you have to enter a 10-byte ID code via the debugger. The code is compared with the ID code stored in the internal flash memory. If the codes do not match, debugging is not possible.

The N-Wire use enable flag can be set or reset while reprogramming the flash by an external flash writer or with the self-programming feature. The flag is located at bit 7 at address 0000 0079<sub>H</sub>.

You can specify your own 10-byte ID code and program it to the internal flash memory by an external flash writer or with the self-programming feature. The ID code is located in the address range 0000 0070<sub>H</sub> to 0000 0079<sub>H</sub>.

The protection levels are summarized in *Table 8-1*

**Table 8-1 Possible results of ID code comparison**

N-Wire use enable flag	ID code	Protection Level
0	X <sup>a</sup>	Level 2: Full protection N-Wire debug interface cannot be used. <sup>b</sup>
1	user-specific ID code	Level 1: ID code protection user ID code N-Wire debug interface can only be used if the user enters the correct ID code.
	ID code is all ones <sup>c</sup>	Level 0: ID code protection with default ID code N-Wire debug interface can be used if the user enters the default ID code FF <sub>H</sub> for all ID bytes.

a) Codes are not compared

b) Once the N-Wire debug interface has been set as “use-prohibited”, it cannot be used until the flash memory is re-programmed.

c) This is the default state after the flash memory has been erased.

- Note**
- After you have set protection levels 1 or 2, set the “block erase disable flag” in the flash extra area. Otherwise, an unauthorized person could erase the block that contains the ID code or the “N-Wire use enable flag”, respectively, and thus suspend the protection.
  - If an unauthorized user tries to find out the 10-byte ID by comparing all possible ID codes, this will take up to  $3.83 \times 10^8$  years at 100 MHz.



## 8.3 Flash Programmer and Self-Programming Protection

In general, illegal read-out and re-programming of the flash memory contents is possible via the flash writer interface and the self-programming feature. For protection of the flash memory, the set of flags which can be set by the user, provide various protection levels.

- Serial programming** It is possible to prohibit any access from external via the serial programming interface, e.g. by an external flash programmer. With maximum protection the internal flash memory can not be erased, read-out or written at all, neither in block units nor the entire flash memory.
- Self-programming** During self-programming all operations to erase, read or program the flash memory is under control of the user's program. Thus no further protection functions in self-programming mode are considered. One exception is the boot block protection, which applies also in self-programming mode.
- Protection flags** The protection flags can be set respectively reset by an external flash programmer, provided the effective protection level allows to do so. In self-programming mode the effective protection flags can not be reset, but other ones can be set to enhance the protection level. The protection flags are stored in the flash memory extra area.
- Each protection function can be used in combination with the others at the same time.

### (1) Write protection flag

Set this flag to disable the write function via external flash programmer interfaces.

No flash memory content can be written from external, if this flag is set. Erasure of single blocks is prohibited as well.

This flag does not affect the self-programming interface.

In self-programming mode writing of the flash memory is further on possible.

### (2) Chip erase protection flag

Set this flag to disable the chip erase function via external flash programmer interfaces.

No flash memory content can be erased - neither in single blocks nor the entire flash memory - from external, if this flag is set.

Chip erase is not available in self-programming mode, though it is possible to erase the entire flash memory content by block erase of all blocks all together. Note that the contents of the extra area is not erased by this means. I.e. protection flags, etc. are still valid.

### (3) Block erase protection flag

Set this flag to disable the feature to erase single blocks via external flash programmer interfaces.

Single blocks can not be erased. Chip erase is still possible, provided the chip erase protection flag is not set.

This flag does not affect the self-programming interface.

In self-programming mode erasure of single blocks or sets of contiguous blocks of the flash memory is further on possible.

**(4) Read-out protection flag**

Set this flag to disable the feature that allows reading back the flash memory via the serial programming interfaces.  
No flash content can be read out.

This flag does not affect the self-programming interface.  
In self-programming mode read-out of flash memory content is further on possible.

**(5) Boot block cluster protection flag**

Set this flag to disable erasure and rewrite of the boot block cluster.  
The boot block cluster can not be manipulated in any way (no erase/write).

This applies in serial and self-programming mode.

Once this flag is set, it is impossible to reset this flag. Thus the boot block cluster content can not be changed any more.

For the explanation of the boot block cluster refer to “*Secure self-programming (boot cluster swapping)*” on page 183.

All protection flags are reset after shipment of the device, thus no protection is enabled at all.

Once a protection flag has been set, i.e. the protection is effective, it can not be reset by any means, except after a chip erase, which erases the entire flash memory including the extra area.

Consequently without prior chip erase the protection level can only be increased, but not decreased.

**Table 8-2 Protection functions overview**

Function	Functional outline	Applicable (√: applies, ×: doesn't apply)	
		Serial programming	Self-programming
Block erase command prohibit	Erasure of single blocks impossible. Once block erase protection is enabled, disable is only possible after chip erase.	√	×
Chip erase command prohibit	Erasure of the entire flash (including the extra area) or single blocks impossible. Once chip erase protection is enabled, all protection flag settings can not be changed any more.	√	×
Program command prohibit	Erasure and rewrite of single blocks impossible. Once write protection is enabled, disable is only possible after chip erase.	√	×
Read command prohibit	Read-out of any flash content impossible. Once read protection is enabled, disable is only possible after chip erase.	√	×
Rewriting boot block cluster prohibit	Erasure (by block or chip erase) or writing of the boot block cluster impossible. Once rewrite protection of the boot block cluster is enabled, it can not be disabled any more.	√	√

**Table 8-3 Rewriting operation when erasing/writing is enabled/prohibited**

Prohibition state		Programming mode	Block erasure		Chip erasure	Write	
			Boot area	None boot area		Boot area	None boot area
Rewriting boot area enabled	All enabled	Self-programming	yes		—	yes	
		Serial programming	yes		yes	yes	
	Block erase command prohibited	Self-programming	yes		—	yes	
		Serial programming	no		yes	yes	
	Chip erase command prohibited	Self-programming	yes		—	yes	
		Serial programming	no		no	yes	
	Write command prohibited	Self-programming	yes		—	yes	
		Serial programming	no		yes	no	
Rewriting boot area prohibited	All enabled	Self-programming	no	yes	—	no	yes
		Serial programming		yes	no		yes
	Block erase command prohibited	Self-programming		yes	—		yes
		Serial programming		no	yes		yes
	Chip erase command prohibited	Self-programming		yes	—		yes
		Serial programming		no	no		yes
	Write command prohibited	Self-programming		yes	—		yes
		Serial programming		no	yes		no

**Note** —: not supported

**Table 8-4 Read operation when reading is enabled/prohibited**

Prohibition State	Programming mode	Read
Read command enabled	Self-programming	√
	Serial programming	√
Read command prohibited	Self-programming	√
	Serial programming	×

**Note** √: execution enabled, ×: execution disabled, —: not supported

## Chapter 9 Bus and Memory Control (BCU, MEMC)

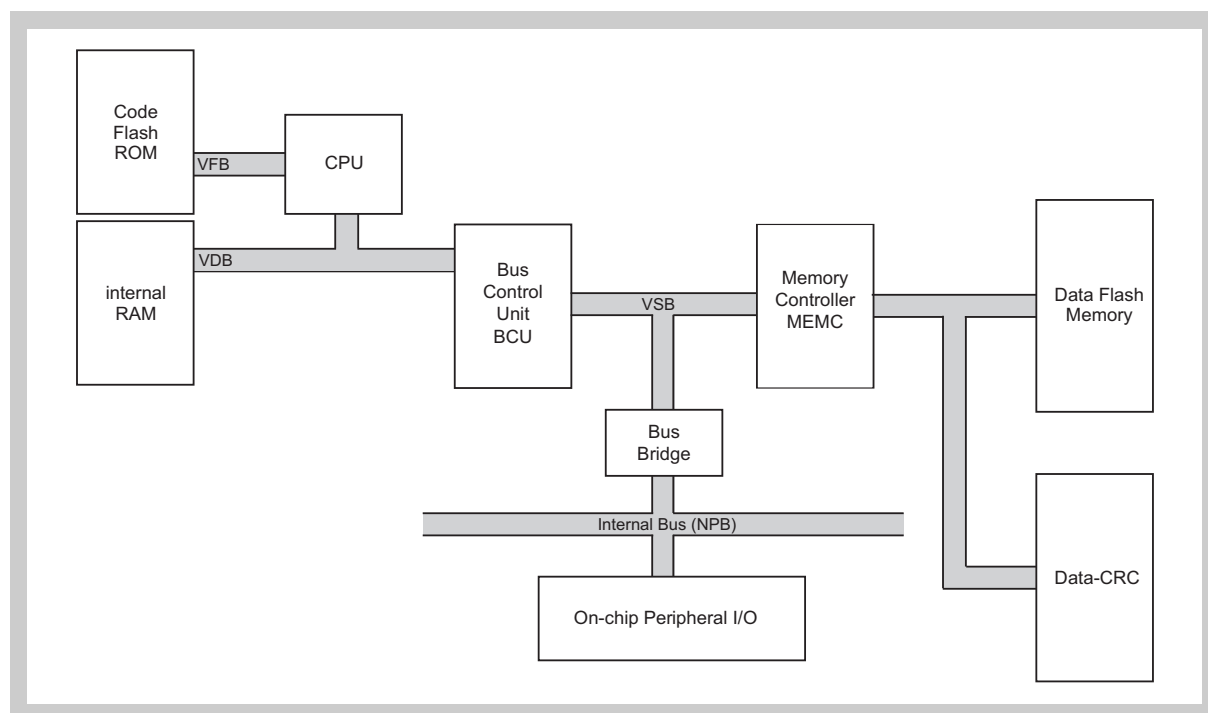
The Bus Control Unit BCU and Memory Controller MEMC within V850E/PHC3 control the access to the on-chip peripherals as well as to the data flash memory.

### 9.1 Overview

- Features summary** The bus and memory control of the microcontroller device provides:
- Access to the on-chip peripherals via the peripheral bus (NPB)
  - Access to the DATA-CRC unit and the data flash memory via the memory controller

### 9.1.1 Description

The figure below shows a block diagram of the modules that are necessary for accessing on-chip peripherals.



**Figure 9-1 Bus and Memory Control block diagram**

**Busses** The busses are abbreviated as follows:

- NPB: Peripheral bus
- VSB: V850 system bus
- VDB: V850 data bus
- VFB: V850 fetch bus

**BCU** The Bus Control Unit (BCU) controls the access to on-chip peripherals.

For access to the data flash memory and the Data CRC unit, the BCU generates the necessary control signals (chip select signals) for the Memory Controller.

**Memory Controller** The Memory Controller generates the control signals for access to the DATA-CRC unit and the data flash. For example, it generates the read strobe ( $\overline{RD}$ ) and the write strobe ( $\overline{WR}$ ).

This product does not provide any signals of the Memory Controller externally.

### 9.1.2 Memory areas/banks and chip select signals

This product does not provide any signals of the Memory Controller externally. Therefore MEMC chip select, address, data, read/write or wait signals are not available.

This section is describing the memory area set-up and the chip select implementation. Further configuration is needed for the dedicated chip select areas to make use of the data flash and the CRC unit which are attached to the memory controller.

The 256 MB address range is divided into the 64 MB areas 0 to 3 (see *Figure 9-2 on page 195*).

The chip select signals  $\overline{CS1}$ ,  $\overline{CS3}$ ,  $\overline{CS4}$ ,  $\overline{CS6}$  are fixed assigned to these areas.

Memory areas 0 and 3 are further subdivided into

- four 2 MB memory banks (banks 0 to 3, respectively 12 to 15), for which the chip select signals  $\overline{CS0}$ ,  $\overline{CS2}$ ,  $\overline{CS5}$ ,  $\overline{CS7}$  are fixed assigned.
- the remaining 56 MB

:

**Chip select area** The memory area that activates the same chip select signal is called chip select area.

**Note** Throughout this chapter, the individual chip select signals are identified by “k” (k = 1, 7) .

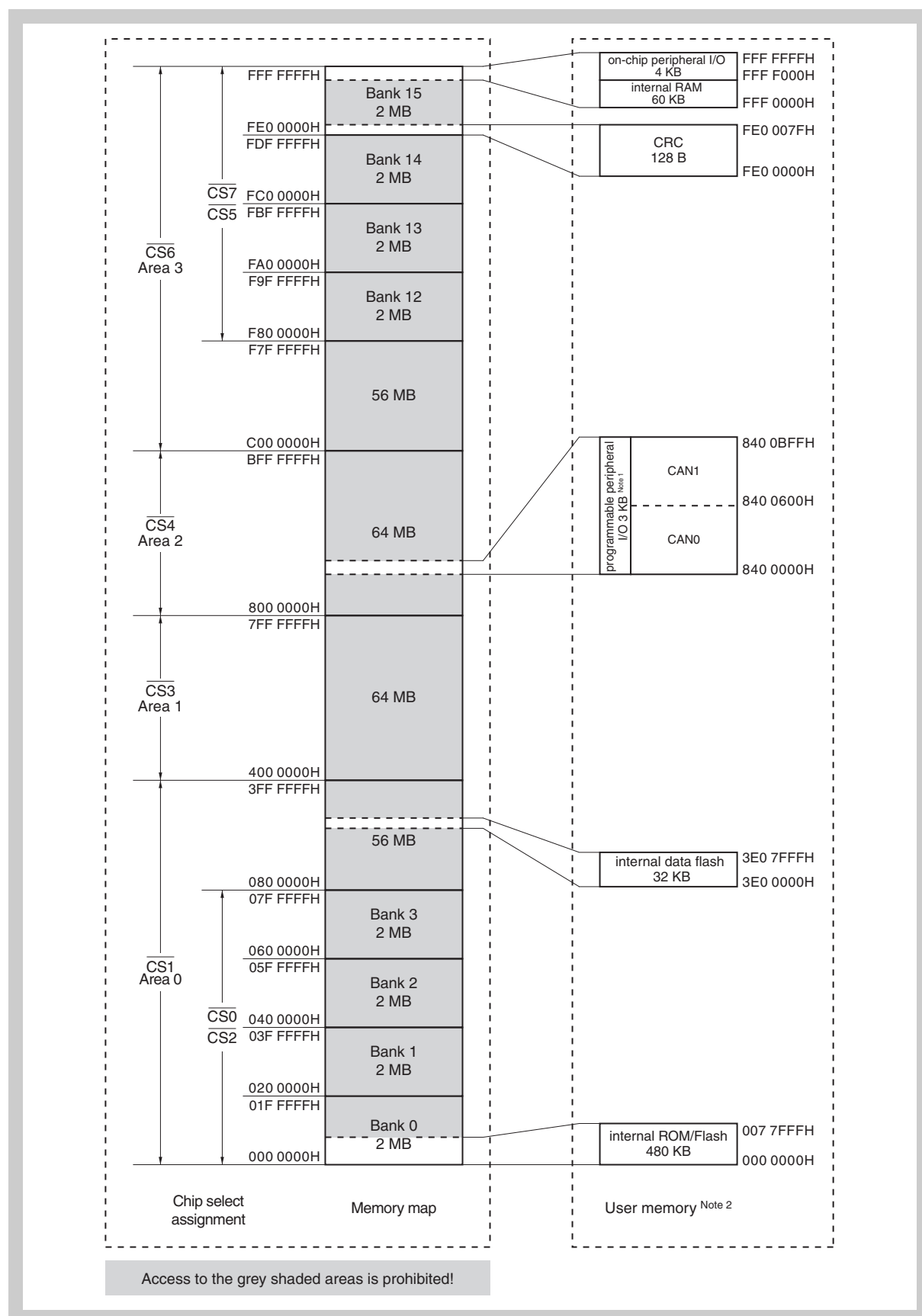


Figure 9-2 Memory areas/banks and chip select signals

- Note**
1. The shown base address 0840 0000H of the programmable peripheral area assumes the Peripheral area selection control register BPC = A100H.
  2. Access to memory areas others than the shown user memory areas is forbidden.



### 9.1.3 Peripheral I/O areas

Two areas of the address range are reserved for the registers of the on-chip peripheral functions. These areas are called “peripheral I/O areas”:

**Table 9-1 Peripheral I/O areas**

Name	Address range	Size
Fixed peripheral I/O area	0FFF F000 <sub>H</sub> to 0FFF FFFF <sub>H</sub>	4 KB
Programmable peripheral I/O area (PPA)	Can be allocated at arbitrary addresses. Base address is defined in the BPC register.	16 KB

#### (1) Fixed peripheral I/O area

The fixed peripheral I/O area holds the registers of the on-chip peripheral I/O functions.

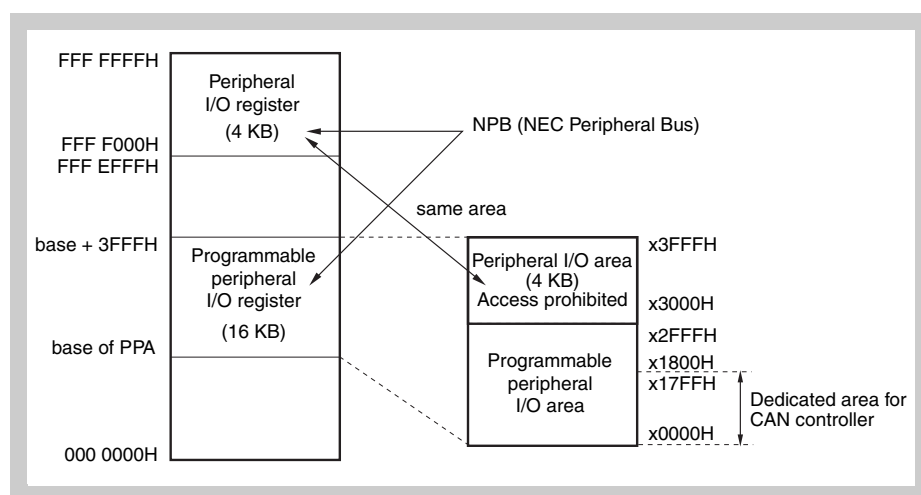
**Access timing** During a read or write access the CPU operation stops until the access via the NPB is completed.

**Note** Because the address space covers 256 MB, the address bits A[31:28] are not considered. Therefore, in this manual, all addresses of peripheral I/O registers in the 4 KB peripheral I/O area are given in the range FFFF F000<sub>H</sub> to FFFF FFFF<sub>H</sub> instead of 0FFF F000<sub>H</sub> to 0FFF FFFF<sub>H</sub>.

#### (2) Programmable peripheral I/O area (PPA)

The usage and the address range of the PPA is configurable. The PPA extends the fixed peripheral I/O area and assigns an additional 12 KB address space for accessing on-chip peripherals.

The figure below illustrates the programmable peripheral I/O area (PPA).



**Figure 9-3 Programmable peripheral I/O area**

The CAN modules registers and message buffers are allocated to the PPA. Refer to “CAN module register and message buffer addresses” on page 393 for information how to calculate the register and message buffer addresses of the CAN modules.

**Access timing** During a read access the CPU operation stops until the read access via the NPB is completed.

---

During a write access the CPU operation continues operation, provided any preceded NPB access is already finished. If a preceded NPB access is still ongoing the CPU stops until this access is finished and the NPB is cleared.

- 
- Caution**
1. If the programmable peripheral I/O area overlaps one of the following areas, the programmable peripheral I/O area becomes ineffective:
    - Peripheral I/O area
    - ROM area
    - RAM area
  2. The *fixed* peripheral I/O area is mirrored to the upper 4 KB of the *programmable* peripheral I/O area – regardless of the base address of the PPA. Access to this mirror of the *fixed* peripheral I/O area is prohibited.
- 

**Note** All address definitions in this manual that refer to the programmable peripheral area assume that the base address of the PPA is 840 0000<sub>H</sub>, that means BPC = A100<sub>H</sub>.

### 9.1.4 Data flash access

The on-chip data flash is accessed by the memory controller, which requires certain settings. The data flash is fixed assigned to  $\overline{CS1}$ , the memory controller options must be configured accordingly.

Table 9-2 shows all required register settings for the data flash's chip select signal  $\overline{CS1}$ . Some settings are appropriate for the data flash per default, others must be changed before the first access to the data flash.

For details about the control settings refer to the description of the registers.

**Table 9-2 Register settings for data flash access**

Control bit	Required setting	Comment
BCT0.ME1	1	<ul style="list-style-type: none"> <li>enable <math>\overline{CS1}</math></li> <li>no default, must be changed</li> </ul>
AWC.AHW1 AWC.ASW1	00 <sub>B</sub>	<ul style="list-style-type: none"> <li>no address setup/hold waits</li> <li>no default, must be changed</li> </ul>
DWC0.DWC1[2:0]	001 <sub>B</sub>	<ul style="list-style-type: none"> <li>1 data wait state</li> <li>no default, must be changed</li> </ul>
BCC.BC11	0	<ul style="list-style-type: none"> <li>no idle states</li> <li>no default, must be changed</li> </ul>

**Base address** The base address of the data flash is 03E0 0000<sub>H</sub>.

### 9.1.5 CRC module access

The on-chip Data-CRC module is accessed by the memory controller, which requires certain settings. The DATA-CRC unit is fixed assigned to  $\overline{CS7}$ , the memory controller options must be configured accordingly.

Table 9-3 shows all required register settings for the CRC module's chip select signal  $\overline{CS7}$ . Some settings are appropriate for the CRC module per default, others must be changed before the first access to the CRC module.

For details about the control settings refer to the description of the registers.

**Table 9-3 Register settings for CRC module access**

Control bit	Required setting	Comment
BCT1.ME7	1	<ul style="list-style-type: none"> <li>enable <math>\overline{CS7}</math></li> <li>no default, must be changed</li> </ul>
AWC.AHW7 AWC.ASW7	00 <sub>B</sub>	<ul style="list-style-type: none"> <li>no address setup/hold waits</li> <li>no default, must be changed</li> </ul>
DWC1.DWC7[2:0]	000 <sub>B</sub>	<ul style="list-style-type: none"> <li>no data wait states</li> <li>no default, must be changed</li> </ul>
BCC.BC71	0	<ul style="list-style-type: none"> <li>no idle states</li> <li>no default, must be changed</li> </ul>

**Base address** The base address of the CRC module is 0FE0 0000<sub>H</sub>.

### 9.1.6 Boundary operation conditions

The microcontroller device has the following boundary operation conditions:

#### (1) Program space

Instruction fetches from the internal peripheral I/O area are inhibited and yield NOP operations.

If a branch instruction exists at the upper limit of the internal RAM area, a pre-fetch operation (invalid fetch) that straddles over the internal peripheral I/O area does not occur.

#### (2) Data space

The microcontroller device is provided with an address misalign function.

By this function, data of any format (word: 32 bit, halfword: 16 bit, byte: 8 bit) can be placed to any address in memory, even though the address is not aligned to the data format (that means address  $4n$  for words, address  $2n$  for halfwords).

- Unaligned halfword data access  
When the LSB of the address is  $A_0 = 1$ , two byte accesses are performed.
- Unaligned word data access  
When the LSB of the address is  $A_0 = 1$ , two byte and one halfword accesses are performed. In total it takes 3 bus cycles.
  - When the LSBs of the address are  $A[1:0] = 10_B$ , two halfword accesses are performed.

**Note** Accessing data on misaligned addresses takes more than one bus cycle to complete data read/write. Consequently, the bus efficiency will drop.

### 9.1.7 Initializing access to the Data flash memory and the CRC module

To enable access to the on-chip Data flash memory and the on-chip Data-CRC module, initialize the following registers after any reset.

1. Bus cycle configuration registers BCTn  
Set BCT0 to 44C4<sub>H</sub> and BCT1 to C444<sub>H</sub>.
2. Data wait control registers DWCN  
Set DWC0 to 7717<sub>H</sub> and DWC1 to 0777<sub>H</sub>.
3. Bus cycle configuration register BCC  
Set BCC to 2AA2<sub>H</sub>.
4. Address wait control register AWC  
Set AWC to 3FF3<sub>H</sub>.
5. Control register DVC  
Specify the frequency of the bus clock BCLK (e.g. to 00<sub>H</sub> for BCLK = f<sub>xx</sub>/2).

- 
- Caution**
1. Do not change these registers after initialization.
  2. Do not access the memory controller controlled areas before initialization is finished.
-

## 9.2 Registers

Access to on-chip peripherals is controlled and operated by registers of the Bus Control Unit (BCU) and the Memory Controller:

**Table 9-4 Bus and memory control register overview**

Module	Register name	Shortcut	Address
Bus Control Unit (BCU)	Peripheral area selection control register	BPC	FFFF F064 <sub>H</sub>
	Internal peripheral function wait control register	VSWC	FFFF F06E <sub>H</sub>
Memory Controller (MEMC)	Bus cycle configuration registers	BCT0	FFFF F480 <sub>H</sub>
		BCT1	FFFF F482 <sub>H</sub>
	Address wait control register	AWC	FFFF F488 <sub>H</sub>
	Data wait control registers	DWC0	FFFF F484 <sub>H</sub>
		DWC1	FFFF F486 <sub>H</sub>
	Bus cycle control register	BCC	FFFF F48A <sub>H</sub>
	Bus clock dividing frequency control register	DVC	FFFF F48E <sub>H</sub>

## 9.2.1 BCU registers

The following registers are part of the BCU. They define the data bus width, the endian format of word data, and they control access to external devices.

### (1) BPC - Peripheral area selection control register

The 16-bit BPC register defines whether the programmable peripheral I/O area (PPA) is used or not and determines the starting address of the PPA.

**Access** This register can be read/written in 16-bit units.

**Address** FFFF F064<sub>H</sub>

**Initial Value** 0000<sub>H</sub>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PA15	0	PA13	PA12	PA11	PA10	PA9	PA8	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0

**Table 9-5 BPC register contents**

Bit Position	Bit Name	Function
15	PA15	Select usage of programmable peripheral I/O area (PPA). 0: PPA disabled 1: PPA enabled
13 to 0	PA[13:0]	Bits PA[13:0] specify bits 27 to 14 of the starting address of the PPA. The other bits of the address are fixed to 0.

**Caution** The bits marked with 0 must always be 0.  
The base address PBA of the programmable peripheral area sets the start address of the 16 KB PPA in a range of 256 MB. The 256 MB page is mirrored 16 times to the entire 32-bit address range.

The base address PBA is calculated by

$$PBA = BPC.PA[13:0] \times 2^{14}$$

Table 9-6 shows how the addresses of the programmable peripheral area are assembled. The base address PBA is highlighted.

**Table 9-6 Address range of programmable peripheral area (16 KB)**

31	...	28	27	...	14	13	...	1	0	bit
0	...	0		BPC.PA[13:0]		1	...	1	1	
...				...						
0	...	0		BPC.PA[13:0]		0	...	0	1	
0	...	0		BPC.PA[13:0]		0	...	0	0	PBA

**Note** The recommended setting for the BPC register is A100<sub>H</sub>. With this configuration the programmable peripheral area is mapped to the address range 0840 0000<sub>H</sub> to 0840 3FFF<sub>H</sub>. With this setting the CAN message buffer registers are accessible via the addresses given in “CAN Controller (CAN)” on page 368.



**(2) VSWC - Internal peripheral function wait control register**

The 8-bit VSWC register controls the bus access wait for the on-chip peripheral I/O registers.

Access to on-chip peripheral I/O registers is made in 3 clocks (without wait), however, waits may be required depending on the operation frequency. Set the values described below to the VSWC register in accordance with the operation frequency used.

For DMA transfers, the bus access wait and signal timing is controlled by registers DMAWC0 and DMAWC1. For a description of these registers see “DMA Functions (DMA Controller)” on page 211.

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

**Address** FFFF F06E<sub>H</sub>

**Initial Value** 77<sub>H</sub>

VSWC must be initialized after  $\overline{\text{RESET}}$  according to below setting:

VSWC = 24<sub>H</sub>

## 9.2.2 Memory Controller registers

The following registers are part of the Memory Controller. They specify the number of data wait states, the number of address wait states, and the number of idle states.

### (1) BCTn - Bus cycle configuration registers

The 16-bit BCTn registers enable the operation of the Memory Controller for each chip select signal.

**Access** These registers can be read/written in 16-bit units.

**Address** BCT0: FFFF F480<sub>H</sub>  
BCT1: FFFF F482<sub>H</sub>

**Initial Value** 4444<sub>H</sub>

These registers must be initialized as follows:

Set BCT0 to 44C4<sub>H</sub> and BCT1 to C444<sub>H</sub>..

BCT0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	ME1	1	0	0	0	1	0	0
$\overline{CS3}$				$\overline{CS2}$				$\overline{CS1}$				$\overline{CS0}$			

BCT1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ME7	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
$\overline{CS7}$				$\overline{CS6}$				$\overline{CS5}$				$\overline{CS4}$			

**Table 9-7 BCTn register contents**

Bit Position	Bit Name	Function
BCT0: 7 BCT1: 11	ME <sub>k</sub>	Enables/disables Memory Controller operation for chip select area k. 0: Operation disabled 1: Operation enabled

- Caution**
1. The bits marked with 0 must always be 0.
  2. The bits marked with 1 must always be 1.
  3. To initialize the memory controller after a reset, registers BCTn have to be set. Do not change this register after initialization. Do not access the memory controller controlled areas before initialization is finished.

**(2) AWC - Address wait control register**

The 16-bit AWC register controls the insertion of an address setup wait and address hold wait state before and after the T1 cycle. The address setup wait and address hold wait state can be enabled for each chip select area.

**Access** This register can be read/written in 16-bit units.

**Address** FFFF F488<sub>H</sub>

**Initial Value** FFFF<sub>H</sub>: After system setup, by default, address hold and wait states insertion is enabled for each chip select area.

This register must be initialized as follows:

Set AWC to 3FF3<sub>H</sub>.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AHW7	ASW7	1	1	1	1	1	1	1	1	1	1	AHW1	ASW1	1	1
CS7		CS6		CS5		CS4		CS3		CS2		CS1		CS0	

**Table 9-8 AWC register contents**

Bit position	Bit name	Function
3, 15	AHWk	Enables/disables the address hold wait insertion for chip select areas 0: No wait state inserted 1: Address hold wait state inserted after T1 bus cycle
2, 14	ASWk	Enables/disables the address setup wait insertion for chip select areas 0: No wait state inserted 1: Address setup wait state inserted before T1 bus cycle

**Caution** To initialize the memory controller after a reset, register AWC has to be set. Do not change this register after initialization. Do not access the memory controller controlled areas before initialization is finished.

**Note** For access to Code Flash, iRAM and peripheral I/O areas, which are not controlled by the MEMC, programmable waits are *not* carried out.

**(3) DWcN - Data wait control registers**

The 16-bit DWcN registers control the number of data wait states after the first access cycle (T1). Each chip select area is controlled separately. A maximum of seven data wait states is possible.

**Access** This register can be read/written in 16-bit units.

**Address** DWc0: FFFF F484<sub>H</sub>  
DWc1: FFFF F486<sub>H</sub>

**Initial Value** 7777<sub>H</sub>: After system setup, by default, seven data wait states are inserted for each chip select area.

These registers must be initialized as follows:

Set DWc0 to 7717<sub>H</sub> and DWc1 to 0777<sub>H</sub>.

DWc0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	0	1	1	1	0	DW12	DW11	DW10	0	1	1	1
CS3				CS2				CS1				CS0			

DWc1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	DW72	DW71	DW70	0	1	1	1	0	1	1	1	0	1	1	1
CS7				CS6				CS5				CS4			

**Table 9-9 DWcN register contents**

Bit position	Bit name	Function													
DWC0: 4, 5, 6 DWC1: 12, 13, 14	DWk[2:0]	Sets the number of wait states after the first access cycle (T1) for chip select areas.													
		DWk[2:0]	Number of inserted wait states	000 <sub>B</sub>	No wait state inserted	001 <sub>B</sub>	1 wait state	010 <sub>B</sub>	2 wait states	011 <sub>B</sub>	3 wait states	...	...	111 <sub>B</sub>	7 wait states
		DWk[2:0]	Number of inserted wait states												
		000 <sub>B</sub>	No wait state inserted												
		001 <sub>B</sub>	1 wait state												
		010 <sub>B</sub>	2 wait states												
		011 <sub>B</sub>	3 wait states												
		...	...												
111 <sub>B</sub>	7 wait states														

**Caution**

1. The bits marked with 0 must always be 0.
2. To initialize the memory controller after a reset, registers DWcN have to be set. Do not change these registers after initialization. Do not access the memory controller controlled areas before initialization is finished.

**Note** For access to Code Flash, iRAM and peripheral I/O areas, which are not controlled by the MEMC, programmable waits are *not* carried out.

**(4) BCC - Bus cycle control register**

The 16-bit BCC register controls for each chip select area

- insertion of an idle state after any *read* access

**Access** This register can be read/written in 16-bit units.

**Address** FFFF F48A<sub>H</sub>

**Initial Value** AAAA<sub>H</sub>: After system reset, an idle state is inserted.

These registers must be initialized as follows:

Set BCC to 2AA2<sub>H</sub>.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BC71	0	1	0	1	0	1	0	1	0	1	0	BC11	0	1	0
$\overline{CS7}$		$\overline{CS6}$		$\overline{CS5}$		$\overline{CS4}$		$\overline{CS3}$		$\overline{CS2}$		$\overline{CS1}$		$\overline{CS0}$	

**Note** The default value 0 of bits 14, 12, 10, 8, 6, 4, 0 and 0 must not be changed.

**Table 9-10 BCC register contents**

Bit position	Bit name	Function
3, 15	BCK1	Enables/disables the idle state insertion after any read access for chip select areas. 0: No idle state inserted 1: Idle state inserted after any read access

**Caution** To initialize the memory controller area after a reset, register BCC has to be set. Do not change this register after initialization. Do not access the memory controller controlled areas before initialization is finished.

**Note** For access to Code Flash, iRAM and peripheral I/O areas, which are not controlled by the MEMC, *no* idle states are inserted.

**(5) DVC - Bus clock dividing frequency control register**

The 8-bit DVC register controls

- idle state inserted after any write access
- the frequency of the bus clock BCLK

**Access** This register can be read/written in 8-bit units.

**Address** FFFF F48E<sub>H</sub>

**Initial Value** 00<sub>H</sub>.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	DVC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 9-11 DVC register contents**

Bit position	Bit name	Function
0	DVC0	Specifies the frequency of the bus clock BCLK. 0: BCLK = $f_{XX}/2$ 1: BCLK = $f_{XX}/4$

- Caution**
1. The bits marked with 0 must always be 0.
  2. To initialize the memory controller after a reset, register DVC has to be set. Do not change this register after initialization. Do not access the memory controller controlled areas before initialization is finished.

**Note** For access to Code Flash, iRAM and peripheral I/O areas, which are not controlled by the MEMC, no idle states are inserted.

# Chapter 10 DMA Functions (DMA Controller)

The microcontroller includes a Direct Memory Access Controller (DMAC) that executes and controls DMA transfers.

**Note** Throughout this chapter, the individual channels of the DMA Controller are identified by “n” (n = 0 to 7)

## 10.1 Overview

The DMAC controls data transfer between internal RAM and I/O, based on DMA requests issued by the on-chip peripheral I/O.

**Features summary** The DMAC has the following features:

- 8 independently configurable channels
- Programmable access wait and signal timing
- Maximum DMA transfer count: 256 ( $2^8$ )

---

**Caution** If the RAM error correction function is disabled, DMA operation is prohibited. Thus disable all DMA channels in that case.

---

### 10.1.1 Principle of operation

A DMA transfer essentially transfers a predefined number of data units between the internal RAM (iRAM) and peripheral I/O registers. The CPU initiates the transfer and the transfer itself is performed by the DMAC.

**DMA channels** Each DMA channel has a fixed transfer direction to transfer data between the internal RAM and a peripheral I/O register:

- DMA channels 0, 1, 4, 5: I/O → iRAM
- DMA channels 2, 3, 6, 7: iRAM → I/O

**DMA priorities** The DMA channel priorities are fixed as follows:  
ch.0 > ch.1 > ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7

**Transfer data units** The size of a data unit to be transferred is either 8 bit or 16 bit:

- DMA channels 0 to 3: 16 bit
- DMA channels 4 to 7: 8 bit or 16 bit selectable

**Transfer addresses** For a DMA transfer, the following addresses have to be defined:

- Source address: address of data to be copied

The source address depends on the DMA channel:

- DMA channels  $n = 0, 1$ : implicitly defined fixed ADC registers
- DMA channels  $n = 4, 5$ : implicitly defined fixed CSIB, CSIE, UARTC data receive register
- DMA channels  $n = 2, 3, 6, 7$ : iRAM start address, defined by MAR $n$  registers
- Destination address: address to store the data to

The destination address depends on the DMA channel:

- DMA channels  $n = 0, 1, 4, 5$ : iRAM start address, defined by MAR $n$  register
- DMA channels  $n = 2, 3$ : Timer S start register address, defined by SAR $n$
- DMA channels  $n = 6, 7$ : implicitly defined fixed CSIB, CSIE, UARTC data transmit registers

Depending on the transfer data unit, source start addresses or destination start addresses are automatically incremented after each data transfer:

- 8-bit data: address increment by 1
- 16-bit data: address increment by 2

**Transfer start/stop** The DMA transfer is initiated by defined interrupts of the on-chip peripheral I/Os. These interrupts are called DMA triggers.

To start a DMA transfer, the following conditions have to be fulfilled:

- DMA transfer is enabled (DMAMCL.DEn = 1).
- The number of transfer actions that has to be performed (transfer count) is written to register DTRC $n$ .
- The transfer status is set to “idle” (DMASL.DMAS $n$  = 0).
- DMA trigger is generated.

When the defined number of transfer actions is completed, a DMA completion interrupt (INTDMA $n$ ) is generated and the DMA transfer is stopped.

**Transfer status** The transfer status can be monitored:

- Register DMASL displays the transfer status (idle/in progress/completed).
- Register DTCR $n$  displays the current transfer count (it is decreased with every transfer action).

### 10.1.2 Forcible termination of DMA transfer

A once started DMA transfer can be forcible terminated by clearing bit DMAMC.DEn = 0.

If the DMAMC.DEn bit is cleared during a DMA transfer action, the current transfer action is finished before the DMA transfer is stopped (see “*Example of forcible termination of DMA transfer*” on page 240).



### 10.1.3 DMA interrupt function

The DMA triggers are interrupt signals of

- A/D Converters
- serial interfaces
- Timer S

These interrupts (except Timer S) are shared with their corresponding DMA transfer completion interrupts (INTDMA<sub>n</sub>). If a DMA channel is enabled, the peripheral I/O interrupt corresponding to the DMA trigger can no longer generate an interrupt request. Instead, the DMA transfer completion interrupt can generate an interrupt with the corresponding interrupt handler address.

The DMA triggers of Timers S (INTTS<sub>n</sub>OD and INTTS<sub>n</sub>CD<sub>0</sub>) are *not* shared with their DMA transfer completion interrupts (INTDMA<sub>n</sub>). These DMA completion interrupts have dedicated entries in the interrupt source list (refer to “Interrupt Controller (INTC)” on page 99).

Table 10-1 shows the relations between DMA triggers and DMA completion interrupts.

**Table 10-1 Relations between DMA triggers and DMA completion interrupts**

DMA channel	DMA trigger	DMA completion interrupt		
		Name	Entry in interrupt list <sup>a</sup>	Handler address
0	ADDMARQ0	INTDMA0	INTAD0	0000 0790 <sub>H</sub>
1	ADDMARQ1	INTDMA1	INTAD1	0000 07A0 <sub>H</sub>
2	INTTS0CD0 or INTTS0OD	INTDMA2	INTDMA2	0000 07B0 <sub>H</sub>
3	INTTS1CD0 or INTTS1OD	INTDMA3	INTDMA3	0000 07C0 <sub>H</sub>
4, 5	INTCE0C	INTDMA4, INTDMA5	INTCE0C	0000 0720 <sub>H</sub>
	INTCE1C		INTCE1C	0000 0740 <sub>H</sub>
	INTCB0R		INTCB0R	0000 06C0 <sub>H</sub>
	INTCB1R		INTCB1R	0000 06F0 <sub>H</sub>
	INTUC0R		INTUC0R	0000 00E0 <sub>H</sub>
	INTUC1R		INTUC1R	0000 00F0 <sub>H</sub>
6, 7	INTCE0C	INTDMA6, INTDMA7	INTCE0C	0000 0720 <sub>H</sub>
	INTCE1C		INTCE1C	0000 0740 <sub>H</sub>
	INTCB0T		INTCB0T	0000 06B0 <sub>H</sub>
	INTCB1T		INTCB1T	0000 06E0 <sub>H</sub>
	INTUC0T		INTUC0T	0000 0760 <sub>H</sub>
	INTUC1T		INTUC1T	0000 0780 <sub>H</sub>

a) The interrupt/exception list is given in “Interrupt Controller (INTC)” on page 99

## 10.2 Registers

The DMA Controller is controlled by means of the following registers:

**Table 10-2 DMA Controller register overview**

Register name	Shortcut	Address
DMA wait control registers	DMAWC0	FFFF FE00 <sub>H</sub>
	DMAWC1	FFFF FE02 <sub>H</sub>
DMA transfer SFR start address registers	SAR2	<base> + 24 <sub>H</sub>
	SAR3	<base> + 26 <sub>H</sub>
DMA transfer memory start address registers	MARn (n = 0 to 7)	<base> + n x 2 <sub>H</sub>
DMA trigger factor registers	DTFR4	<base> + 88 <sub>H</sub>
	DTFR5	<base> + 8A <sub>H</sub>
	DTFR6	<base> + 8C <sub>H</sub>
	DTFR7	<base> + 8E <sub>H</sub>
DMA transfer count registers	DTCRn (n = 0 to 7)	<base> + 40 <sub>H</sub> + n x 2 <sub>H</sub>
DMA status register	DMASL	<base> + 62 <sub>H</sub>
DMA mode control register	DMAMCL	<base> + 60 <sub>H</sub>
DMA data size control register	DMADSCL	<base> + 64 <sub>H</sub>

**Table 10-3 DMA Controller register base address**

Unit	Base address
DMA Controller	FFFF F300 <sub>H</sub>

---

**(1) DMAWCm - DMA wait control registers**

The 8-bit DMAWCm registers control the internal bus timing for DMA transfers (m = 0, 1).

**Access** These registers can be read/written in 8-bit units.

**Address** DMAWC0: FFFF FE00<sub>H</sub>  
DMAWC1: FFFF FE02<sub>H</sub>

**Initial Value** DMAWC0: 37<sub>H</sub>  
DMAWC1: 07<sub>H</sub>

**DMAWC0** The DMAWC0 register must have the same setting as the internal peripheral function wait control register VSWC:

DMAWC0 = VSWC.

Note that the contents of the VSWC register depend on the system clock (see “Bus and Memory Control (BCU, MEMC)” on page 192 for details).

**DMAWC1** The DMAWC1 register must be set as follows:

DMAWC1 = 00<sub>H</sub>.

---

**Caution** The DMAWCm registers must be set up correctly before the first DMA transfer is started. Do not change these registers afterwards.

---

**(2) SARn - DMA transfer SFR start address registers**

The 8-bit SARn register specifies the start address of the TMS register for which the DMA transfer is started on the corresponding DMA channel n (n = 2, 3).

This register is only valid for DMA transfers to timers S.

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

**Address** SAR2: <base> + 24<sub>H</sub>

SAR3: <base> + 26<sub>H</sub>

**Initial Value** 0x00.

7	6	5	4	3	2	1	0
0	0	0	0	SARn3	SARn2	SARn1	SARn0
R	R	R	R	R/W	R/W	R/W	R/W

Writing to SARn.bit[7:4] is ignored.

**Table 10-4 SARn register contents**

Bit position	Bit name	Function																																																																																																																								
3 to 0	SARn[3:0] (n = 2, 3)	Sets the start address of the TMS register for which the DMA transfer is started on the corresponding DMA channel n (n = 2, 3).																																																																																																																								
		<table><tr><th rowspan="3">SARn3</th><th rowspan="3">SARn2</th><th rowspan="3">SARn1</th><th rowspan="3">SARn0</th><th colspan="4">DMA transfer start address of TMS reload register</th></tr><tr><th colspan="2">n = 2</th><th colspan="2">n = 3</th></tr><tr><th>Register</th><th>Address</th><th>Register</th><th>Address</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>TS0OPT1A</td><td>FFFF FB40<sub>H</sub></td><td>TS1OPT1A</td><td>FFFF FBC0<sub>H</sub></td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>TS0DTC1A</td><td>FFFF FB42<sub>H</sub></td><td>TS1DTC1A</td><td>FFFF FBC2<sub>H</sub></td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>TS0DTC0A</td><td>FFFF FB44<sub>H</sub></td><td>TS1DTC0A</td><td>FFFF FBC4<sub>H</sub></td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>TS0CCR5A</td><td>FFFF FB46<sub>H</sub></td><td>TS1CCR5A</td><td>0FFFF FBC6<sub>H</sub></td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>TS0CCR4A</td><td>FFFF FB48<sub>H</sub></td><td>TS1CCR4A</td><td>FFFF FBC8<sub>H</sub></td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>TS0CCR3B</td><td>FFFF FB4A<sub>H</sub></td><td>TS1CCR3B</td><td>FFFF FBCA<sub>H</sub></td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>TS0CCR2B/ TS0PAT1<sup>a</sup></td><td>FFFF FB4C<sub>H</sub></td><td>TS1CCR2B/ TS1PAT1<sup>b</sup></td><td>FFFF FBCC<sub>H</sub></td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>TS0CCR1B/ TS0PAT0<sup>c</sup></td><td>FFFF FB4E<sub>H</sub></td><td>TS1CCR1B/ TS1PAT0<sup>d</sup></td><td>FFFF FBCE<sub>H</sub></td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>TS0CCR0A</td><td>FFFF FB50<sub>H</sub></td><td>TS1CCR0A</td><td>FFFF FBD0<sub>H</sub></td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>TS0CCR3A</td><td>FFFF FB52<sub>H</sub></td><td>TS1CCR3A</td><td>FFFF FBD2<sub>H</sub></td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>TS0CCR2A</td><td>FFFF FB54<sub>H</sub></td><td>TS1CCR2A</td><td>FFFF FBD4<sub>H</sub></td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td><td>TS0CCR1A</td><td>FFFF FB56<sub>H</sub></td><td>TS1CCR1A</td><td>FFFF FBD6<sub>H</sub></td></tr><tr><td colspan="4">other settings than above</td><td colspan="4">prohibited</td></tr></table>	SARn3	SARn2	SARn1	SARn0	DMA transfer start address of TMS reload register				n = 2		n = 3		Register	Address	Register	Address	0	0	0	0	TS0OPT1A	FFFF FB40 <sub>H</sub>	TS1OPT1A	FFFF FBC0 <sub>H</sub>	0	0	0	1	TS0DTC1A	FFFF FB42 <sub>H</sub>	TS1DTC1A	FFFF FBC2 <sub>H</sub>	0	0	1	0	TS0DTC0A	FFFF FB44 <sub>H</sub>	TS1DTC0A	FFFF FBC4 <sub>H</sub>	0	0	1	1	TS0CCR5A	FFFF FB46 <sub>H</sub>	TS1CCR5A	0FFFF FBC6 <sub>H</sub>	0	1	0	0	TS0CCR4A	FFFF FB48 <sub>H</sub>	TS1CCR4A	FFFF FBC8 <sub>H</sub>	0	1	0	1	TS0CCR3B	FFFF FB4A <sub>H</sub>	TS1CCR3B	FFFF FBCA <sub>H</sub>	0	1	1	0	TS0CCR2B/ TS0PAT1 <sup>a</sup>	FFFF FB4C <sub>H</sub>	TS1CCR2B/ TS1PAT1 <sup>b</sup>	FFFF FBCC <sub>H</sub>	0	1	1	1	TS0CCR1B/ TS0PAT0 <sup>c</sup>	FFFF FB4E <sub>H</sub>	TS1CCR1B/ TS1PAT0 <sup>d</sup>	FFFF FBCE <sub>H</sub>	1	0	0	0	TS0CCR0A	FFFF FB50 <sub>H</sub>	TS1CCR0A	FFFF FBD0 <sub>H</sub>	1	0	0	1	TS0CCR3A	FFFF FB52 <sub>H</sub>	TS1CCR3A	FFFF FBD2 <sub>H</sub>	1	0	1	0	TS0CCR2A	FFFF FB54 <sub>H</sub>	TS1CCR2A	FFFF FBD4 <sub>H</sub>	1	0	1	1	TS0CCR1A	FFFF FB56 <sub>H</sub>	TS1CCR1A	FFFF FBD6 <sub>H</sub>	other settings than above				prohibited			
		SARn3					SARn2	SARn1	SARn0	DMA transfer start address of TMS reload register																																																																																																																
										n = 2		n = 3																																																																																																														
			Register	Address	Register	Address																																																																																																																				
		0	0	0	0	TS0OPT1A	FFFF FB40 <sub>H</sub>	TS1OPT1A	FFFF FBC0 <sub>H</sub>																																																																																																																	
		0	0	0	1	TS0DTC1A	FFFF FB42 <sub>H</sub>	TS1DTC1A	FFFF FBC2 <sub>H</sub>																																																																																																																	
		0	0	1	0	TS0DTC0A	FFFF FB44 <sub>H</sub>	TS1DTC0A	FFFF FBC4 <sub>H</sub>																																																																																																																	
		0	0	1	1	TS0CCR5A	FFFF FB46 <sub>H</sub>	TS1CCR5A	0FFFF FBC6 <sub>H</sub>																																																																																																																	
		0	1	0	0	TS0CCR4A	FFFF FB48 <sub>H</sub>	TS1CCR4A	FFFF FBC8 <sub>H</sub>																																																																																																																	
		0	1	0	1	TS0CCR3B	FFFF FB4A <sub>H</sub>	TS1CCR3B	FFFF FBCA <sub>H</sub>																																																																																																																	
		0	1	1	0	TS0CCR2B/ TS0PAT1 <sup>a</sup>	FFFF FB4C <sub>H</sub>	TS1CCR2B/ TS1PAT1 <sup>b</sup>	FFFF FBCC <sub>H</sub>																																																																																																																	
		0	1	1	1	TS0CCR1B/ TS0PAT0 <sup>c</sup>	FFFF FB4E <sub>H</sub>	TS1CCR1B/ TS1PAT0 <sup>d</sup>	FFFF FBCE <sub>H</sub>																																																																																																																	
		1	0	0	0	TS0CCR0A	FFFF FB50 <sub>H</sub>	TS1CCR0A	FFFF FBD0 <sub>H</sub>																																																																																																																	
		1	0	0	1	TS0CCR3A	FFFF FB52 <sub>H</sub>	TS1CCR3A	FFFF FBD2 <sub>H</sub>																																																																																																																	
		1	0	1	0	TS0CCR2A	FFFF FB54 <sub>H</sub>	TS1CCR2A	FFFF FBD4 <sub>H</sub>																																																																																																																	
		1	0	1	1	TS0CCR1A	FFFF FB56 <sub>H</sub>	TS1CCR1A	FFFF FBD6 <sub>H</sub>																																																																																																																	
other settings than above				prohibited																																																																																																																						

a) TS0CCR2B in T-PWM mode, TSOPAT1 in all other modes

b) TS1CCR2B in T-PWM mode, TS1PAT1 in all other modes

c) TS0CCR1B in T-PWM mode, TSOPAT0 in all other modes

d) TS1CCR1B in T-PWM mode, TS1PAT0 in all other modes

**(3) MARn - DMA transfer memory start address registers**

The 16-bit MARn register specifies the subordinated 16 bits of

- the source address (transfer direction iRAM → I/O)
- the destination start address (transfer direction I/O → iRAM)

within the internal RAM area for the DMA channel n.

**Access** This register can be read/written in 16-bit units.

**Address** <base> + n x 2<sub>H</sub>

**Initial Value** 0x0000.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MARn15	MARn14	MARn13	MARn12	MARn11	MARn10	MARn9	MARn8	MARn7	MARn6	MARn5	MARn4	MARn3	MARn2	MARn1	MARn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 10-5 MARn register contents**

Bit position	Bit name	Function
15 to 0	MARn15 to MARn0	Sets the subordinated 16 bits of the DMA transfer start address (A15 to A0).

- Caution**
1. Due to the internal RAM size, the value written to the MARn register has to be in the range from FFFF\_7000<sub>H</sub> to FFFF\_EFFF<sub>H</sub>.
  2. For DMA channels that are configured for 16-bit transfer data units, the RAM source or destination area must be 16 bit aligned. Thus the memory address must be even, that is MARn.MARn0 = 0.

**(4) DTFRn - DMA trigger factor registers**

The 8-bit DTFRn register controls the DMA transfer start trigger of DMA channel n via interrupt requests from on-chip peripheral I/O (n = 4 to 7).

The interrupt request set by this register serves as DMA transfer start factor.

This register is only valid for DMA transfers to serial interfaces.

**Access** This register can be read/written in 8-bit units.

**Address** DTFR4: <base> + 88<sub>H</sub>  
 DTFR5: <base> + 8A<sub>H</sub>  
 DTFR6: <base> + 8C<sub>H</sub>  
 DTFR7: <base> + 8E<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by  $\overline{\text{RESET}}$ .

7	6	5	4	3	2	1	0
0	0	0	0	0	IFCn2	IFCn1	IFCn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Caution**
1. Do not set the same DMA trigger by different DTFRn registers.
  2. Do not rewrite the DTFRn register until a started DMA transfer ends (corresponding DTCRn register value is 00<sub>H</sub>).
  3. Write the DTFRn register before setting the corresponding DTCRn register. Otherwise, if the DMA trigger is present, the DMA transfer might be started by writing the DTFRn register.

**Table 10-6 DTFRn register contents**

Bit position	Bit name	Function				
2 to 0	IFCn[2:0] (n = 4 to 7)	Sets the DMA transfer start trigger of DMA channel n via interrupt requests from on-chip peripheral I/O (n = 4 to 7).				
		IFCn2	IFCn1	IFCn0	DMA transfer start factor	
					when n = 4, 5	when n = 6, 7
		0	0	0	DMA request from on-chip peripheral I/O disabled	
		0	0	1	INTUC0R	INTUC0T
		0	1	0	INTUC1R	INTUC1T
		0	1	1	INTCB0R	INTCB0T
		1	0	0	INTCB1R	INTCB1T
		1	0	1	INTCE0C	INTCE0C
		1	1	0	INTCE1C	INTCE1C
		1	1	1	setting prohibited	setting prohibited

**(5) DTCRn - DMA transfer count registers**

The 8-bit DTCRn register has two functions:

- Before DMA transfer, it sets the transfer count for DMA channel n.
- During DMA transfer, it stores the remaining transfer count.

**Access** This register can be read/written in 8-bit units.

**Address** <base> + 40<sub>H</sub> + n x 2<sub>H</sub>

**Initial Value** 0x00.

7	6	5	4	3	2	1	0
DTCRn7	DTCRn6	DTCRn5	DTCRn4	DTCRn3	DTCRn2	DTCRn1	DTCRn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 10-7 DTCRn register contents**

Bit position	Bit name	Function						
7 to 0	DTCRn[7:0]	Sets the transfer count for DMA channel n and stores the remaining transfer count during DMA transfer.						
		DTCRn7	DTCRn6	...	DTCRn2	DTCRn1	DTCRn0	Remaining DMA transfer counts
		0	0	...	0	0	0	256
		0	0	...	0	0	1	1
		0	0	...	0	1	0	2
		0	0	...	0	1	1	3
		...	...	...	...	...	...	...
		1	1	...	1	0	1	253
		1	1	...	1	1	0	254
		1	1	...	1	1	1	255

- Caution**
1. The value set to the DTCRn register is decreased by each DMA transfer of channel n. It does not keep the initial value after the DMA transfer ends. Therefore, after DMA transfer end the DTCRn register value becomes 00<sub>H</sub>.
  2. A DMA request becomes only effective after the DTCRn register was written. Even if 00<sub>H</sub> (means a transfer count of 256) is the initial value, the DTRCn register must be rewritten in order to enable a new DMA transfer.
  3. Writing to the DTCRn register during DMA transmission (DMAMC.DEn=1) is prohibited.
  4. The DMA transfer ends if DTCRn becomes 00<sub>H</sub>. Thus even if a new DMA trigger occurs no transfer takes place and no DMA interrupt is generated.

**(6) DMASL - DMA status register**

The 8-bit DMASL register displays the transfer status of the DMA channels.

**Access** This register can be read/written in 8-bit and 1-bit units.

**Address** <base> + 62<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by  $\overline{\text{RESET}}$ .

7	6	5	4	3	2	1	0
DMAS7	DMAS6	DMAS5	DMAS4	DMAS3	DMAS2	DMAS1	DMAS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 10-8 DMASL register contents**

Bit position	Bit name	Function
7 to 0	DMASn	Displays the transfer status of DMA channel n: 0: Transfer is idle or on progress 1: Transfer is completed

- Note**
1. The DMASn 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.
  2. Since the DMASn bit is not cleared by the DMAC, it has to be cleared by software before DMA transfer is started.



**(7) DMAMCL - DMA mode control register**

The 8-bit DMAMCL register enables/disables the operation of the DMA channels.

**Access** This register can be read/written in 8-bit and 1-bit units.

**Address** <base> + 60<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by  $\overline{\text{RESET}}$ .

7	6	5	4	3	2	1	0
DE7	DE6	DE5	DE4	DE3	DE2	DE1	DE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 10-9 DMAMCL register contents**

Bit position	Bit name	Function
7 to 0	DEn	Enables/disables DMA transfer of channel n: 0: Operation disabled 1: Operation enabled

**Caution** Writing of a DEn bit is prohibited if the corresponding peripheral function, that generates the DMA trigger, is operating.

**(8) DMADSCl - DMA data size control register**

The 8-bit DMADSCl register controls the transfer data size of DMA channels 4 to 7. The transfer data size of DMA channels 0 to 3 is fixed to 16 bits.

This register is only valid for DMA transfers to serial interfaces.

**Access** This register can be read/written in 8-bit and 1-bit units.

**Address** <base> + 64<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by  $\overline{\text{RESET}}$ .

7	6	5	4	3	2	1	0
DMADSC7	DMADSC6	DMADSC5	DMADSC4	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 10-10 DMADSCl register contents**

Bit position	Bit name	Function
7 to 4	DMADSC <sub>n</sub> (n = 4 to 7)	Sets the transfer data size of DMA channel n: 0: 8 bits 1: 16 bits

## 10.3 Transfer Details

This section presents details and examples of all possible DMA transfers:

- “DMA transfer from A/D Converter” on page 223
- “DMA transfer to Timer S” on page 227
- “DMA transfer to/from serial interfaces” on page 232

Further, an “Example of forcible termination of DMA transfer” on page 240 is given.

### 10.3.1 DMA transfer from A/D Converter

**Caution** For transferring A/D Converter results from the A/D Converter to the internal RAM via DMA the A/D Converter has to be operated in timer triggered scan mode. Refer to “A/D Converter (ADC)” on page 506 for information how to set up this mode.

The following table summarizes the DMA transfer  $ADC_m \rightarrow iRAM$  ( $m = 0, 1$ ).

**Table 10-11 Overview of DMA transfer from A/D Converter**

Channel n	Transfer unit	DMA trigger		Transfer			
				Size	Max. count	Source register	Destination start address MARn
0	ADC0	ADDMARQ0	End of conversion interrupt signal of A/D Converter	16 bit	256	ADDMA0 (fixed)	any even address in iRAM
1	ADC1	ADDMARQ1				ADDMA1 (fixed)	

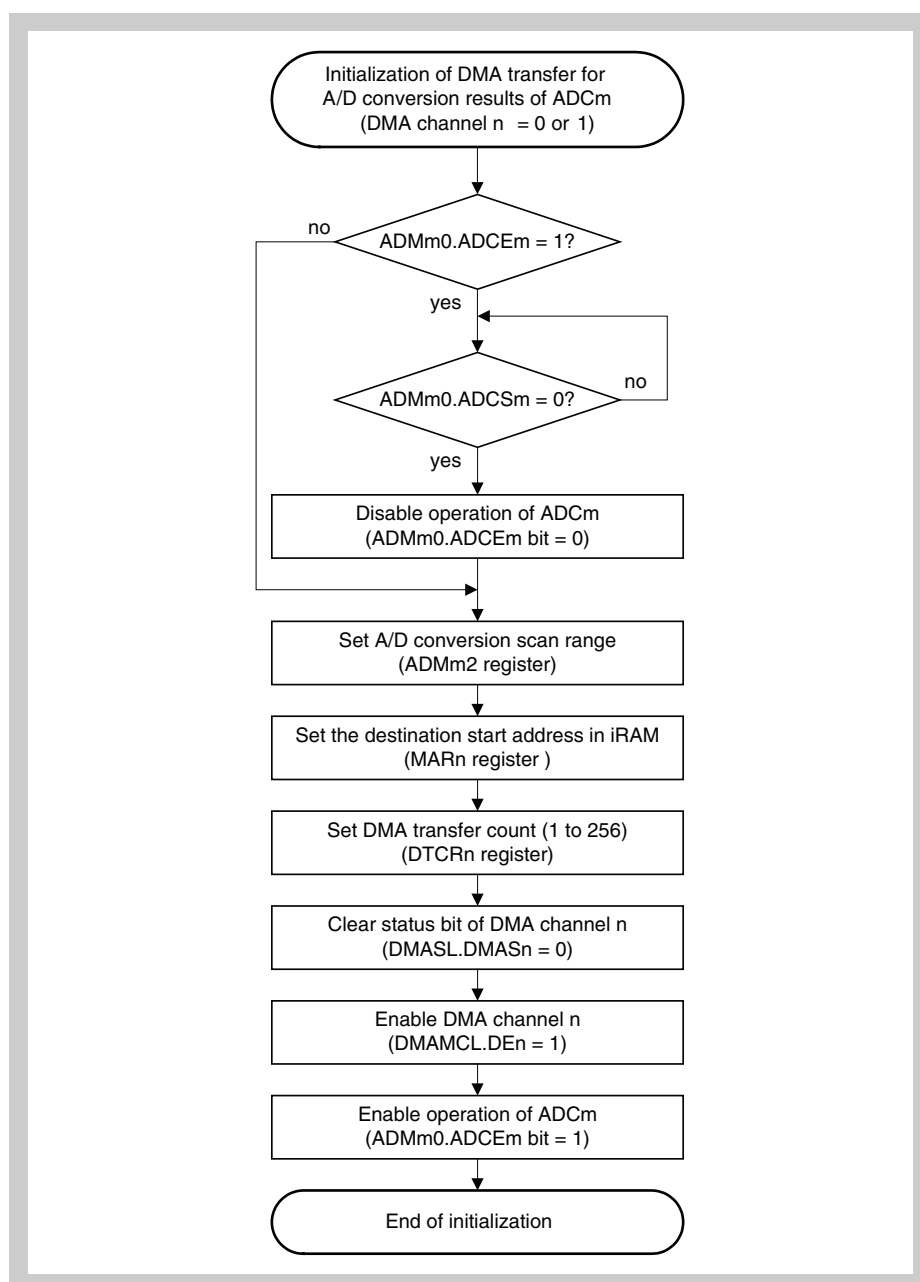
For each DMA trigger, the data will be transferred from the A/D conversion result register for DMA (ADDMA<sub>m</sub>) to the internal RAM area specified by the destination start address in MAR<sub>n</sub>.

**Transferred data** The data that is to be transferred is defined by the following:

- A/D Converter scan area  
The number of ADC channels that are to be scanned can be defined in the A/D Converter.  
The DMA transfer is performed for every finished A/D conversion within the A/D Converter scan area. Thus, for every ADC channel, a separate DMA transfer is required.
- DMA transfer count  
The number of DMA transfers has to be defined in the DTCR<sub>n</sub> register. The total number of DMA transfers can be calculated as  
scan area size × number of DMA triggers

**Note** The user has to take care that the number of DMA transfer counts complies with the A/D Converter scan area size and the number of A/D Converter start triggers.

For an example see “Example of DMA transfer of A/D Converter results” on page 226.

**(1) Principle of DMA transfer of A/D conversion results****Figure 10-1 Initialization of DMA Channel 0/1 for Transfer of A/D Conversion Results**

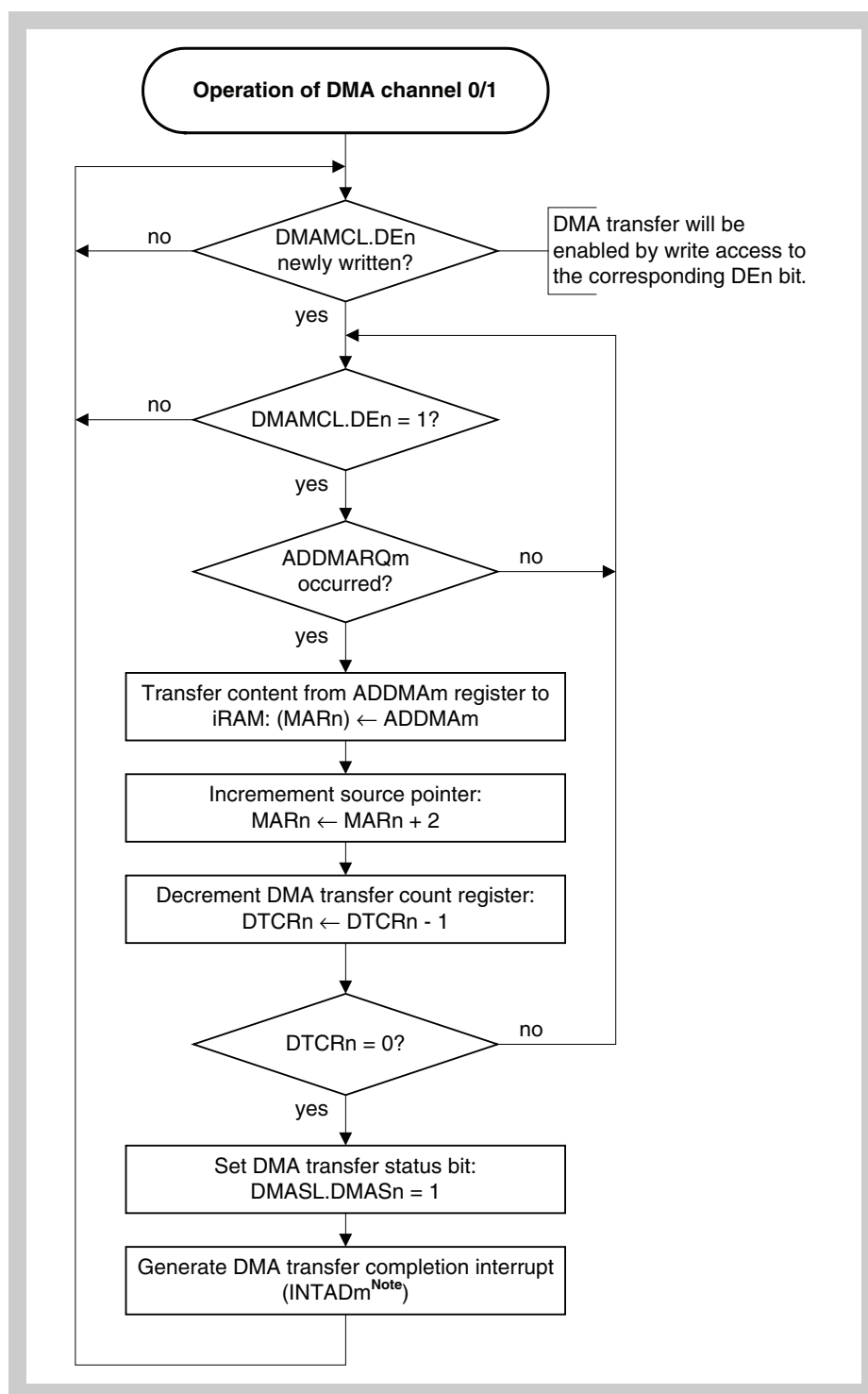


Figure 10-2 Operation of DMA Channel 0/1

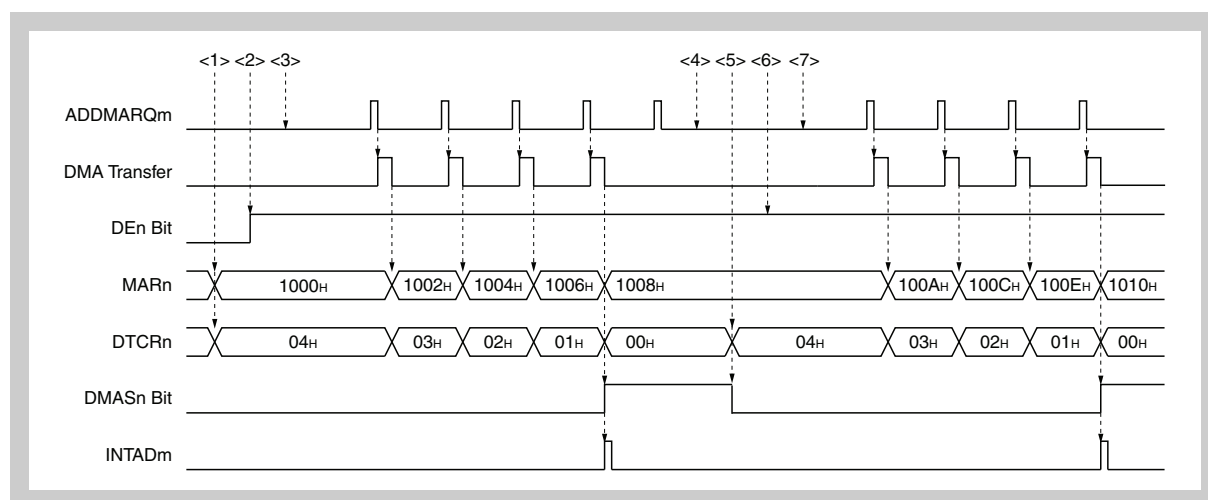
**Note** The DMA completion interrupt has the same interrupt vector address as the A/D conversion completion interrupt (INTADm) of the corresponding ADCm, and replaces that interrupt.

**(2) Example of DMA transfer of A/D Converter results**

Figure 10-3 exemplifies a DMA transfer from the A/D conversion result register of ADC0 to the internal RAM.

**Example** The following steps are performed by the exemplary application program.

- <1> Setup of DMA transfer:  
     MARn = 1000<sub>H</sub> (start address of FFFF1000<sub>H</sub> in iRAM)  
     DTCRn = 04<sub>H</sub> (DMA transfer count is 4)  
     DMASL.DMASn = 0
- <2> Start DMA transfer:  
     DMAMCL.DEn = 1
- <3> Enable ADCm:  
     ADMm0.ADCEm = 1
- <4> Disable ADCm:  
     ADMm0.ADCEm = 0
- <5> Re-setup and restart of the DMA transfer:  
     DTCRn = 04<sub>H</sub>  
     DMASL.DMASn = 0
- <6> Restart of the DMA transfer:  
     DMAMCL.DEn = 1
- <7> Enable ADCm:  
     ADMm0.ADCEm = 1



**Figure 10-3 Exemplary Timing Diagram of a DMA Transfer of A/D Conversion Result**

### 10.3.2 DMA transfer to Timer S

The following table summarizes the DMA transfer iRAM → TMSm (m = 0, 1):

**Table 10-12 Overview of DMA transfer to Timer S**

Channel n	Transfer unit	DMA trigger		Transfer				
				Size	Max. count	Source address (MARn)	Destination address	
							Start (SARn)	End
2	TMS0	INTTS0OD, INTTS0CD0	TSMCNT top / bottom reversal	16 bit	01 <sub>H</sub> to 03 <sub>H</sub>	any even address in iRAM	refer to below table	TSMCCR1A
3	TMS1	INTTS1OD, INTTS1CD0						

For each DMA trigger, the data will be transferred from internal RAM to the capture/compare registers of timer TMSm.

**Transferred data** The data that is to be transferred is defined by the following destination area:

- The *destination start address*  
It is defined by the SARn.SARn[3:0] register bits. The address offset of the Timer S DMA start register is evaluated from SARn.SARn[3:0] (refer to below table).
- The *destination end address*  
It is fixed to the TSMCCR1A register of Timer TMSm.
- The timer TMS address mapping  
It defines the order in which registers are written to. The table below lists all possible destination registers together with their address offset and related setting of SARn.SARn[3:0]:

DMA transfer destination address			
TMSm registers	SARn.SARn[3:0]	Address Offset	
TSMOPT1A	00 <sub>H</sub>	00 <sub>H</sub>	Selectable as destination start address
TSMBTC1	01 <sub>H</sub>	02 <sub>H</sub>	
TSMBTC0	02 <sub>H</sub>	04 <sub>H</sub>	
TSMCCR5A	03 <sub>H</sub>	06 <sub>H</sub>	
TSMCCR4A	04 <sub>H</sub>	08 <sub>H</sub>	
TSMCCR3B	05 <sub>H</sub>	0A <sub>H</sub>	
TSMCCR2B/ TSMPAT1	06 <sub>H</sub>	0C <sub>H</sub>	
TSMCCR1B/ TSMPAT0	07 <sub>H</sub>	0E <sub>H</sub>	
TSMCCR0A	08 <sub>H</sub>	10 <sub>H</sub>	
TSMCCR3A	09 <sub>H</sub>	12 <sub>H</sub>	
TSMCCR2A	0A <sub>H</sub>	14 <sub>H</sub>	
TSMCCR1A	0B <sub>H</sub>	16 <sub>H</sub>	Always destination end address

The DMA transfer is performed for every reload of the internal buffer compare registers by the contents of the capture/compare registers TSmCCR0A to TSmCCR5A.

With every DMA transfer, data is written to all registers within the destination area.

For an example see *“Example of DMA transfer for timer S reload” on page 231.*



## (1) Principle of DMA transfer for timer S reload

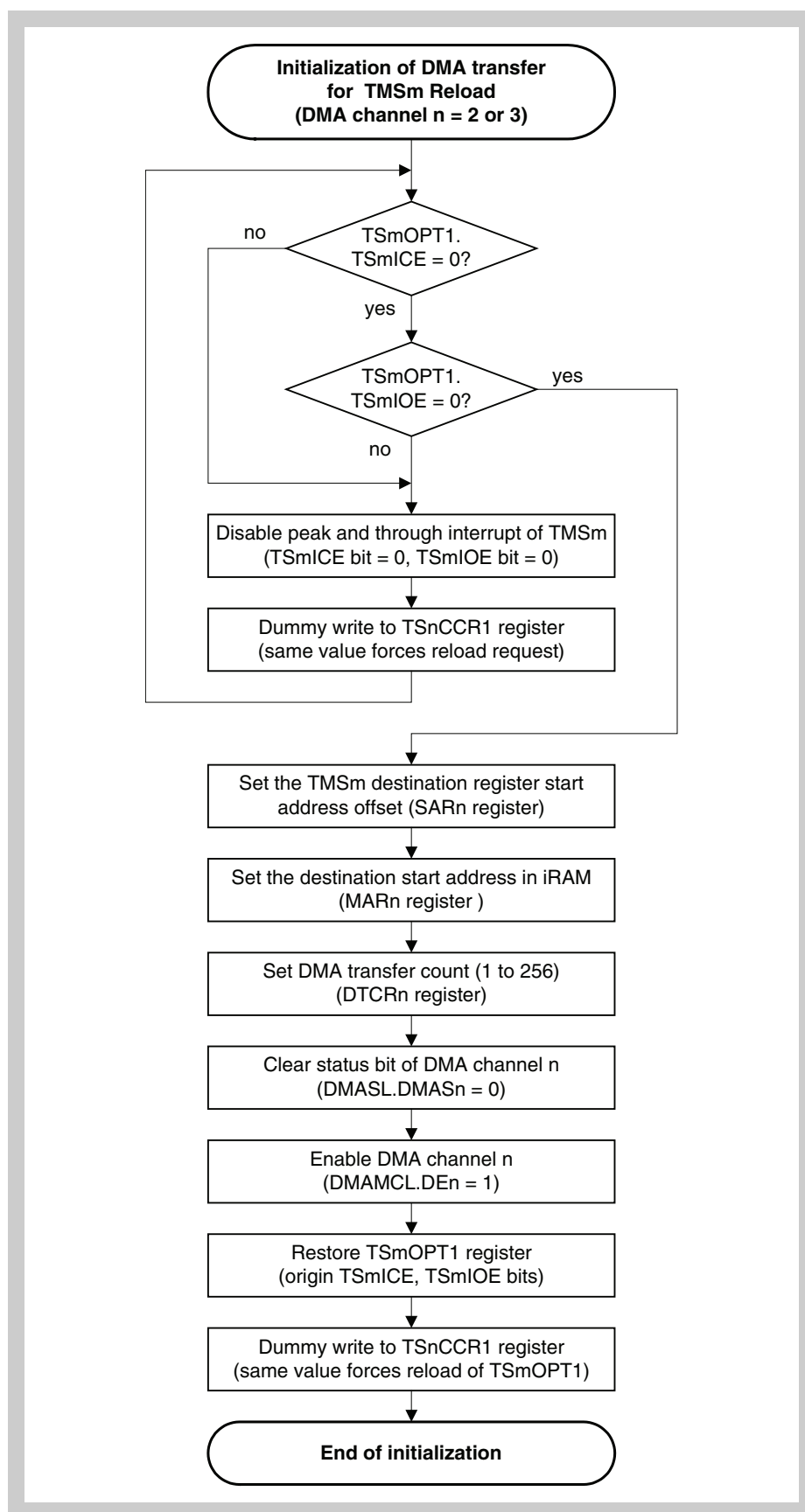


Figure 10-4 Initialization of DMA Channel 2/3 for Timer S Reload

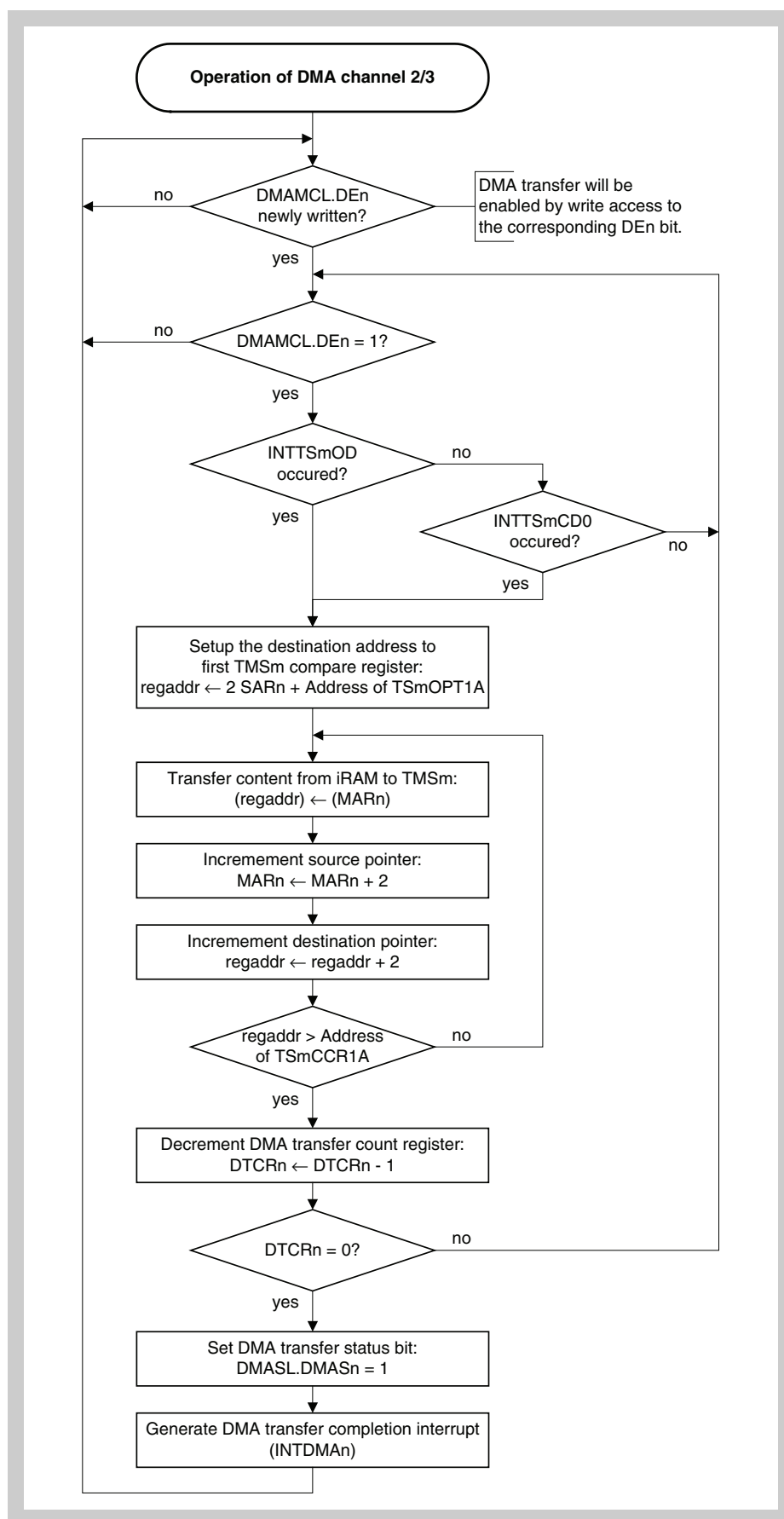


Figure 10-5 Operation of DMA Channel 2/3

**(2) Example of DMA transfer for timer S reload**

Figure 10-3 exemplifies a DMA transfer from the internal RAM to timer TMS0 compare registers for reload.

**Example** The following steps are performed by the exemplary application program.

- <1> Setup of DMA transfer:  
 SAR2 = 08<sub>H</sub> (means start register is TSmCCR0A)  
 MAR2 = C000<sub>H</sub> (means start address of FFFFC000<sub>H</sub> in iRAM)  
 DTCR2 = 02<sub>H</sub> (means a transfer count of 2)  
 DMASL.DMAS2 = 0
- <2> Ensure that DMA trigger factor (INTTS0OD or INTTS0CD0) does not occur when DE2 is set, otherwise disable TMS0:  
 TS0CTL0.TS0CEm = 0
- <3> Start DMA transfer:  
 DMAMCL.DE2 = 1
- <4> If disabled, enable TMS0:  
 TS0CTL0.TS0CE = 1

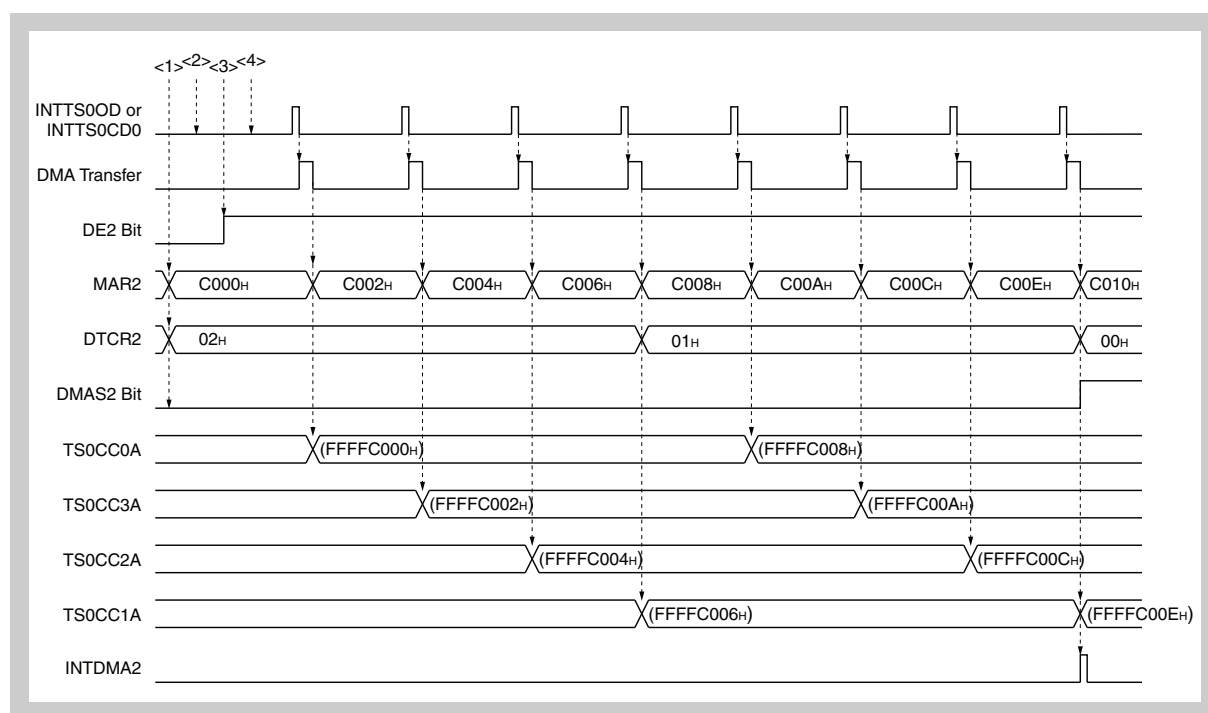


Figure 10-6 Exemplary Timing Diagram of DMA Transfer for TMS0 Reload

### 10.3.3 DMA transfer to/from serial interfaces

#### (1) Serial data reception with DMA transfer

The DMA channels 4 and 5 are dedicated to transfer data from serial interfaces CSIE0, CSIE1, CSIB0, CSIB1, UARTC0, UARTC1 to the iRAM.

The following table summarizes the DMA transfer serial interface → iRAM:

**Table 10-13 Overview of DMA transfer from serial interfaces**

Channel n	Transfer unit	DMA trigger		Transfer			
				Size	Max. count	Source address	Destination start address (MARn)
4	CSIE0	INTCE0C	Receive complete interrupt	8 bit	255	CE0RX0L	Any iRAM address
				16 bit		CE0RX0	Any even iRAM address
	CSIE1	INTCE1C		8 bit		CE1RX0L	Any iRAM address
				16 bit		CE1RX0	Any even iRAM address
	UARTC0	INTUC0R		8 bit		UC0RX	Any iRAM address
				16 bit		Setting prohibited	
	UARTC1	INTUC1R		8 bit		UC1RX	Any iRAM address
				16 bit		Setting prohibited	
	CSIB0	INTCB0R		8 bit		CB0RXL	Any iRAM address
				16 bit		CB0RX	Any even iRAM address
	CSIB1	INTCB1R		8 bit		CB1RXL	Any iRAM address
				16 bit		CB1RX	Any even iRAM address
5	CSIE0	INTCE0C		8 bit	CE0RX0L	Any iRAM address	
				16 bit	CE0RX0	Any even iRAM address	
	CSIE1	INTCE1C		8 bit	CE1RX0L	Any iRAM address	
				16 bit	CE1RX0	Any even iRAM address	
	UARTC0	INTUC0R		8 bit	UC0RX	Any iRAM address	
				16 bit	Setting prohibited		
	UARTC1	INTUC1R		8 bit	UC1RX	Any iRAM address	
				16 bit	Setting prohibited		
	CSIB0	INTCB0R		8 bit	CB0RXL	Any iRAM address	
				16 bit	CB0RX	Any even iRAM address	
	CSIB1	INTCB1R		8 bit	CB1RXL	Any iRAM address	
				16 bit	CB1RX	Any even iRAM address	

---

<b>Source address</b>	The source address is any of the listed serial interfaces of ' data receive registers. It is implicitly defined by the channel number $n$ and the selection made by the DMA trigger factor register DTFR $n$ register.
<b>Destination address</b>	The destination address is any iRAM address (for 16-bit transfers an even iRAM address), specified by the DMA memory start address MAR $n$ .
<b>DMA trigger</b>	<p>Upon each DMA trigger 8-bit or 16-bit data is transferred from the corresponding serial reception register to the internal RAM.</p> <p>The DMA trigger is the end-of-reception interrupt signal of the selected serial interface. Like the source address, It is defined by the DMA channel number and the DMA trigger factor register DTFR<math>n</math> register.</p>
<b>Transfer size</b>	<p>The transfer data size can be set to 8- or 16-bit via the register DMADSCL.</p> <ul style="list-style-type: none"><li>• In case of 8-bit transfer data size, the destination address is incremented by 1 for each DMA trigger.</li><li>• In case of 16-bit transfer data size, the destination address MAR<math>n</math> must be even, and is incremented by 2 for each DMA trigger.</li></ul>

## (2) Principle of DMA transfer from serial interfaces (reception)

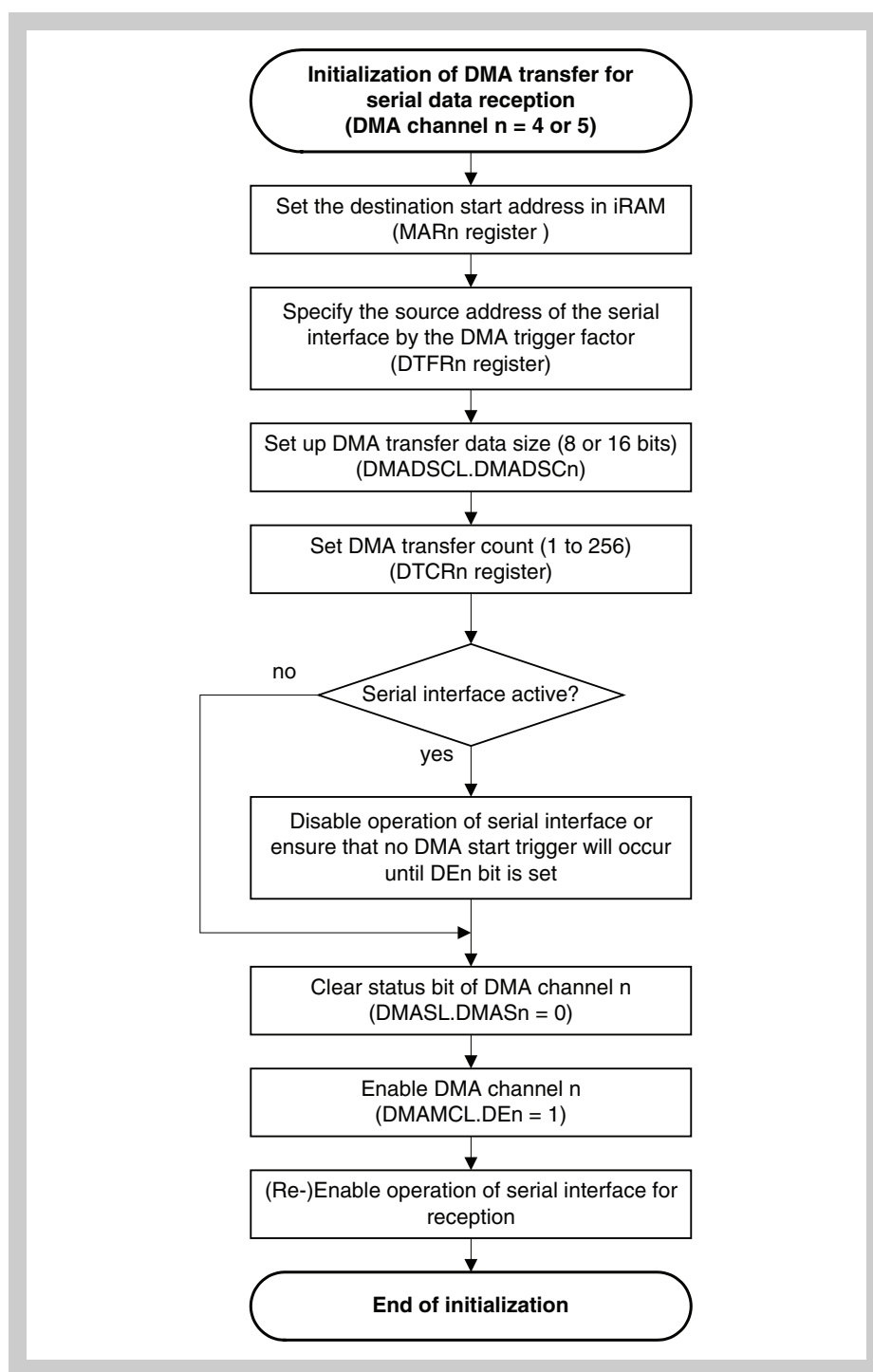


Figure 10-7 Initialization of DMA Channel 4/5 for Serial Data Reception

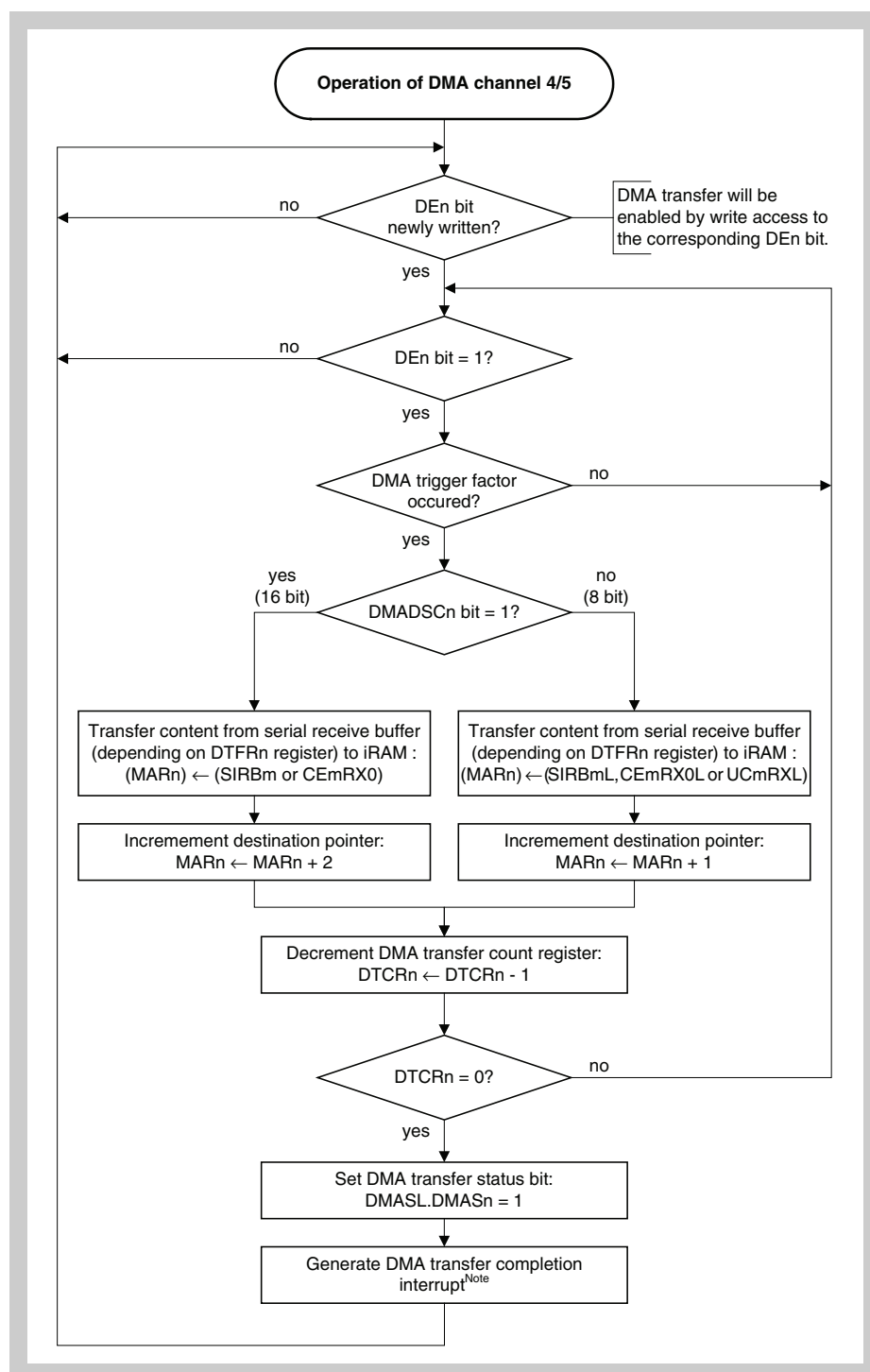


Figure 10-8 Operation of DMA Channel 4/5

**Note** The interrupt vector address of the DMA transfer completion interrupt overlays and replaces the corresponding reception completion interrupt of the serial interface source specified by the DTFRn register.

**(3) Serial data transmission with DMA transfer**

The DMA channels 6 and 7 are dedicated to transfer data from the iRAM to the serial interfaces CSIE0, CSIE1, CSIB0, CSIB1, UARTC0, UARTC1.

The following table summarizes the DMA transfer iRAM → serial interface:

**Table 10-14 Overview of DMA transfer to serial interfaces**

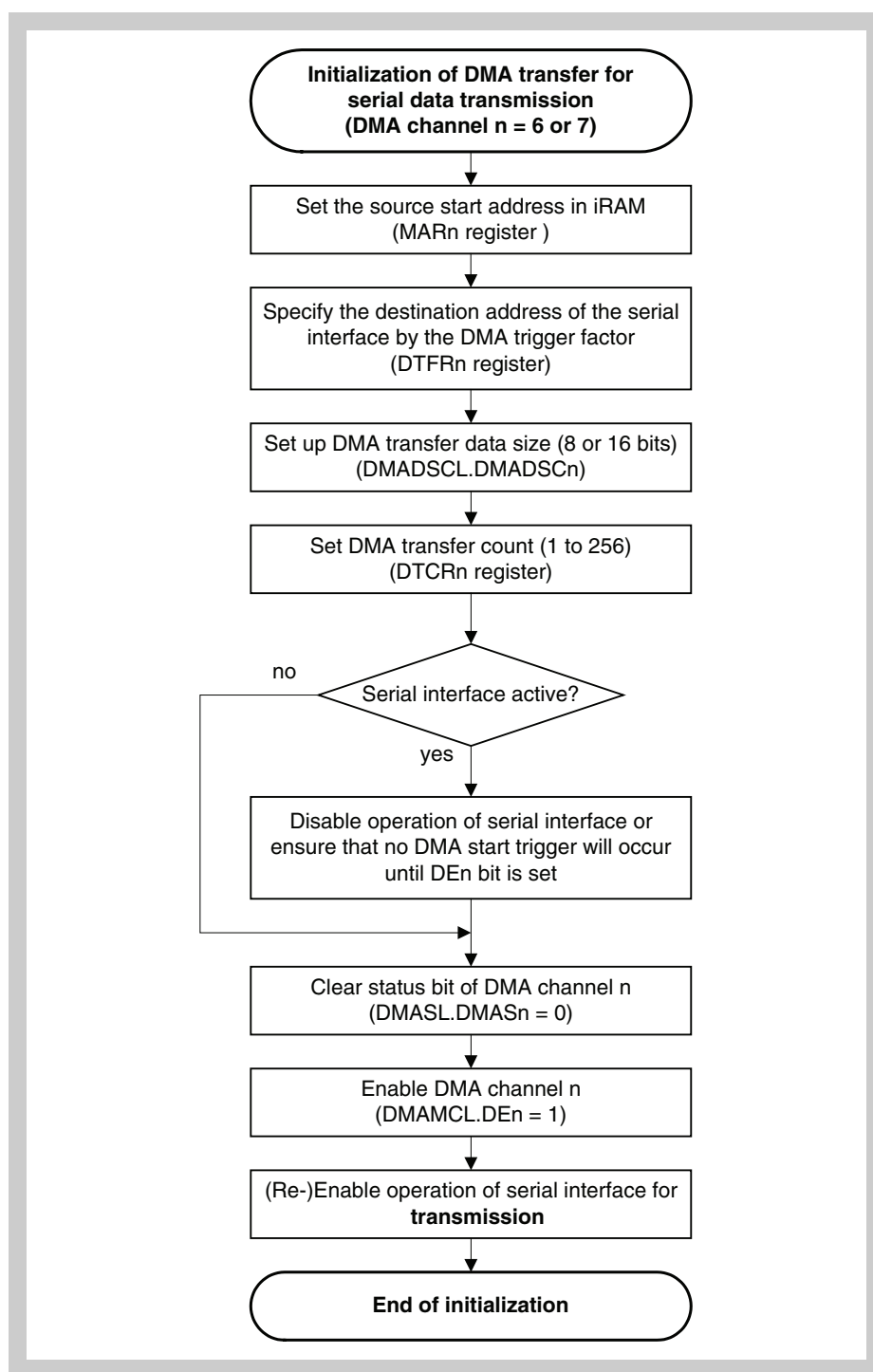
Channel n	Transfer unit	DMA trigger		Transfer			
				Size	Max. count	Source address (MARn)	Destination start address
6	CSIE0 <sup>a</sup>	INTCE0C	Transmit interrupt	8 bit	255	Any iRAM address	CE0TX0L
				16 bit		Any even iRAM address	CE0TX0
	CSIE1 <sup>a</sup>	INTCE1C		8 bit		Any iRAM address	CE1TX0L
				16 bit		Any even iRAM address	CE1TX0
	UARTC0	INTUC0T		8 bit		Any iRAM address	UC0TX
				16 bit		Setting prohibited	
	UARTC1	INTUC1T		8 bit		Any iRAM address	UC1TX
				16 bit		Setting prohibited	
	CSIB0	INTCB0T		8 bit		Any iRAM address	CB0TXL
				16 bit		Any even iRAM address	CB0TX
	CSIB1	INTCB1T		8 bit		Any iRAM address	CB1TXL
				16 bit		Any even iRAM address	CB1TX
7	CSIE0 <sup>a</sup>	INTCE0C		8 bit	255	Any iRAM address	CE0TX0L
				16 bit		Any even iRAM address	CE0TX0
	CSIE1 <sup>a</sup>	INTCE1C		8 bit		Any iRAM address	CE1TX0L
				16 bit		Any even iRAM address	CE1TX0
	UARTC0	INTUC0T		8 bit		Any iRAM address	UC0TX
				16 bit		Setting prohibited	
	UARTC1	INTUC1T		8 bit		Any iRAM address	UC1TX
				16 bit		Setting prohibited	
	CSIB0	INTCB0T		8 bit		Any iRAM address	CB0TXL
				16 bit		Any even iRAM address	CB0TX
	CSIB1	INTCB1T		8 bit		Any iRAM address	CB1TXL
				16 bit		Any even iRAM address	CB1TX

a) Writing to the chip select data buffer registers CSnCS is not supported by DMA.



---

<b>Source address</b>	The source address is any iRAM address (for 16-bit transfers an even iRAM address), specified by the DMA memory start address MARn.
<b>Destination address</b>	The destination address is any of the listed serial interfaces' data transmit registers. It is implicitly defined by the channel number n and the selection made by the DMA trigger factor register DTFRn register.
<b>DMA trigger</b>	<p>Upon each DMA trigger 8-bit or 16-bit is transferred from the internal RAM to the corresponding serial transmit register.</p> <p>The DMA trigger is the transmit-enable interrupt signal of the selected serial interface. Like the destination address, It is defined by the DMA channel number and the DMA trigger factor register DTFRn register.</p>
<b>Transfer size</b>	<p>The transfer data size can be set to 8- or 16-bit via the register DMADSCL.</p> <ul style="list-style-type: none"><li>• In case of 8-bit transfer data size, the source address is incremented by 1 for each DMA trigger.</li><li>• In case of 16-bit transfer data size, the source address MARn must be even, and is incremented by 2 for each DMA trigger.</li></ul>

**(4) Principle of DMA transfer to serial interfaces (transmission)****Figure 10-9 Initialization of DMA Channel 6/7 for Serial Data Transmission**

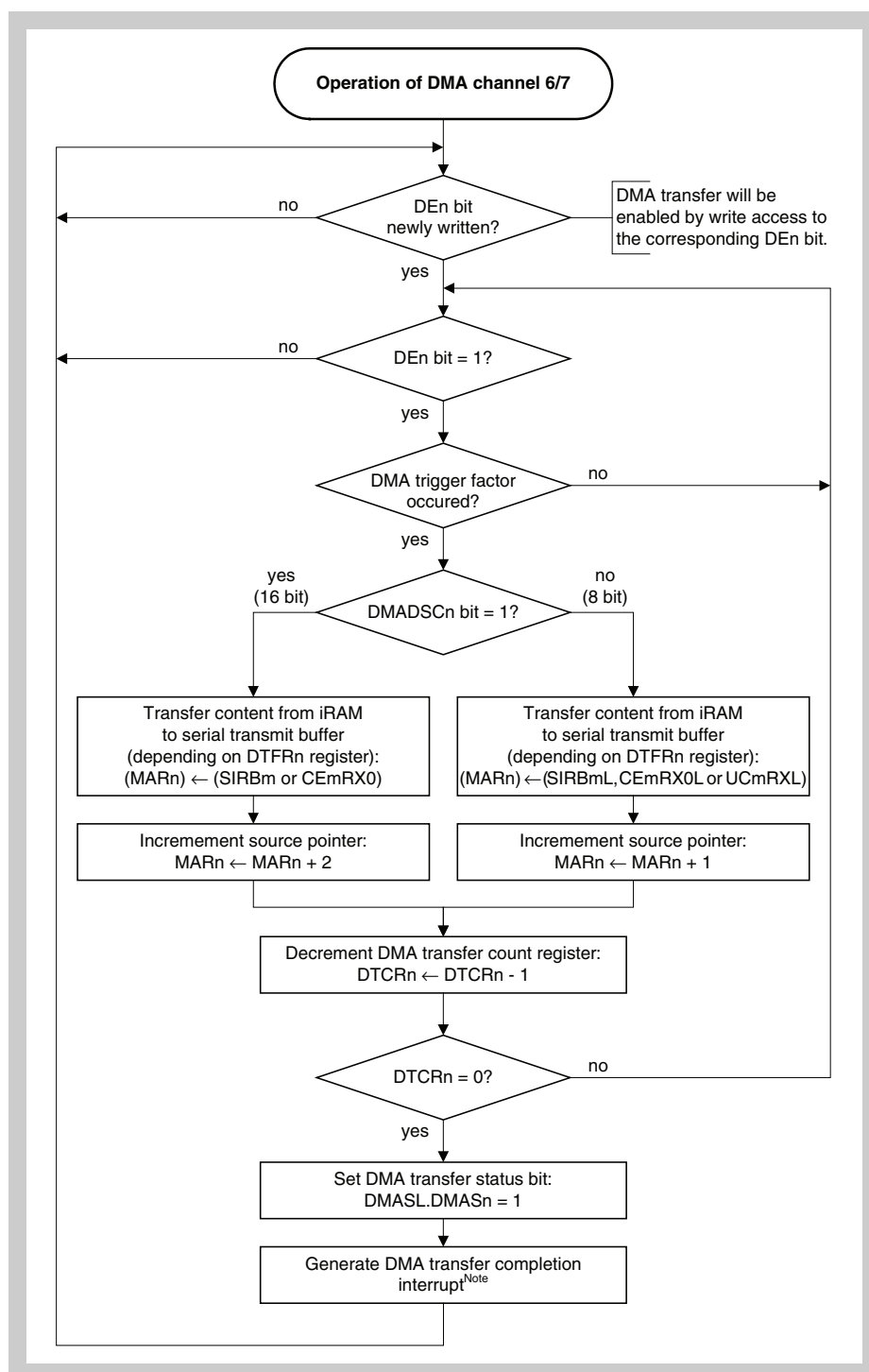


Figure 10-10 Operation of DMA Channel 6/7

**Note** The interrupt vector address of the DMA transfer completion interrupt overlays and replaces the corresponding transmission completion interrupt of the serial interface source specified by the DTFRn register.

---

### 10.3.4 Example of forcible termination of DMA transfer

- CPU**
1. Set DMA transfer count (e.g. DTCRn = 10<sub>H</sub>)
  2. Start DMA transfer (DMAMCL.DEn = 1)
- DMA**
3. Waiting for DMA trigger
  4. Start first data transfer
  5. Decrement DTCRn register by 1 (e.g. DTCRn = 09<sub>H</sub>)
  6. Waiting for DMA trigger
  7. Start next data transfer
- CPU**
8. Stop DMA transfer forcibly (DMAMCL.DEn = 0)
- DMA**
9. Decrement DTCRn register by 1 (e.g. DTCRn = 08<sub>H</sub>)
  10. DMA transfer is stopped completely

# Chapter 11 Asynchronous Serial Interface (UARTC)

This microcontroller has two instances of the universal Asynchronous Serial Interface UARTC.

**Note** Throughout this chapter, the individual instances of UARTC<sub>n</sub> are identified by “n” (n = 0 to 1), for example UCnCTL0 for the UARTC<sub>n</sub> control register 0.

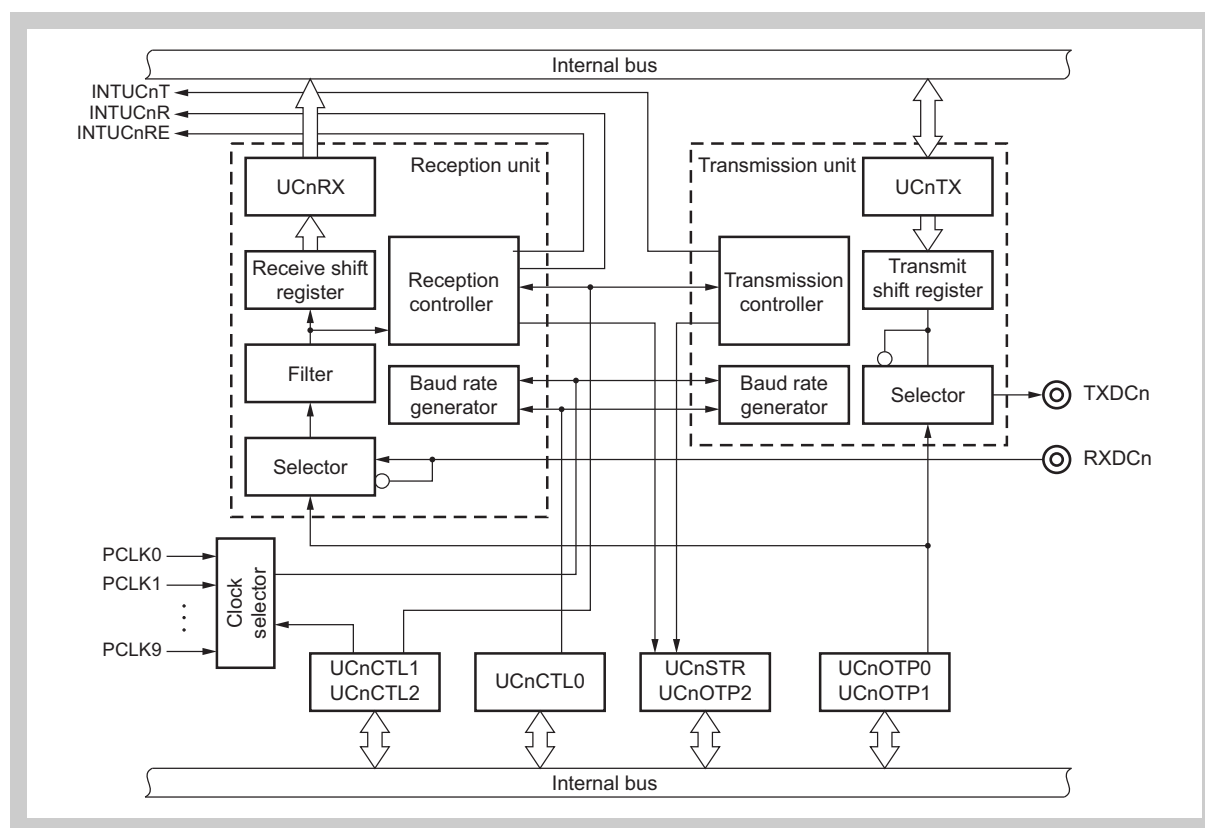
## 11.1 Features

- Transfer rate: 300 bps to 4 Mbps
- Full-duplex communication:
  - Internal UARTC receive data register n (UCnRX)
  - Internal UARTC transmit data register n (UCnTX)
- 2-pin configuration:
  - TXDCn: Transmit data output pin
  - RXDCn: Receive data input pin
- Reception error output function
  - Parity error
  - Framing error
  - Overrun error
- Interrupt sources: 3
  - Reception complete interrupt (INTUCnR):  
This interrupt occurs upon transfer of receive data from the shift register to receive buffer register n after serial transfer completion, in the reception enabled status.
  - Transmission enable interrupt (INTUCnT):  
This interrupt occurs upon transfer of transmit data from the transmit buffer register to the shift register in the transmission enabled status.
  - Receive error interrupt (INTUCnRE):  
This interrupt occurs upon transfer of erroneous receive data.
- 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 level inversion possible

- 13 to 20 bits selectable for the SBF (Sync Break Field) in the LIN (Local Interconnect Network) communication format
  - Recognition of 11 bits or more possible for SBF reception in LIN communication format
  - SBF reception flag provided
- Extension bit operation possible (uses parity bit as 9th data bit)
- Transfer and reception status flags

## 11.2 Configuration

The block diagram of the UARTCn is shown below.



**Figure 11-1** Block diagram of Asynchronous Serial Interface UARTCn

**Note** PCLK0 to PCLK9: peripheral clocks (refer to “Clock Generator” on page 148). For the configuration of the baud rate generator, see Figure 11-11 on page 270.

UARTCn consists of the following hardware units.

**Table 11-1** Configuration of UARTCn

Item	Configuration
Registers	UARTCn control register 0 (UCnCTL0) UARTCn control register 1 (UCnCTL1) UARTCn control register 2 (UCnCTL2) UARTCn option control register 0 (UCnOPT0) UARTCn option control register 1 (UCnOPT1) UARTCn option control register 2 (UCnOPT2) UARTCn status register (UCnSTR) UARTCn receive shift register UARTCn receive data register (UCnRX) UARTCn transmit shift register UARTCn transmit data register (UCnTX)

**(1) UARTCn control register 0 (UCnCTL0)**

The UCnCTL0 register is an 8-bit register used to specify the UARTCn operation.

**(2) UARTCn control register 1 (UCnCTL1)**

The UCnCTL1 register is an 8-bit register used to select the input clock for the UARTCn.

**(3) UARTCn control register 2 (UCnCTL2)**

The UCnCTL2 register is an 8-bit register used to control the baud rate for the UARTCn.

**(4) UARTCn option control register 0 (UCnOPT0)**

The UCnOPT0 register is an 8-bit register used to control serial transfer for the UARTCn.

**(5) UARTCn option control register 1 (UCnOPT1)**

The UCnOPT1 register is an 8-bit register used to control the extension bit operation.

**(6) UARTCn option control register 2 (UCnOPT2)**

The UCnOPT2 register is an 8-bit register indicates the operating status during a reception.

**(7) UARTCn status register (UCnSTR)**

The UCnSTRn register consists of flags indicating the error contents when a reception error occurs. Each one of the reception error flags is set (to 1) upon occurrence of a reception error and is reset (to 0) by reading the UCnSTR register.

**(8) UARTCn receive shift register**

This is a shift register used to convert the serial data input to the RXDCn pin into parallel data. Upon reception of 1 byte of data and detection of the stop bit, the receive data is transferred to the UCnRX register.

This register cannot be manipulated directly.

**(9) UARTCn receive data register (UCnRX)**

The UCnRX 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 UARTCn receive shift register to the UCnRX register in synchronization with the completion of shift-in processing of 1 frame.

Transfer to the UCnRX register also causes the reception complete interrupt request signal (INTUCnR) to be output.

**(10) UARTCn transmit shift register**

The transmit shift register is a shift register used to convert the parallel data transferred from the UCnTX register into serial data.

When 1 byte of data is transferred from the UCnTX register, the shift register data is output from the TXDCn pin.

This register cannot be manipulated directly.

**(11) UARTCn transmit data register (UCnTX)**

The UCnTX register is an 8-bit transmit data buffer. Transmission starts when transmit data is written to the UCnTX register. When data can be written to the UCnTX register (when data of one frame is transferred from the UCnTX register to the UARTCn transmit shift register), the transmission enable interrupt request signal (INTUCnT) is generated.



## 11.3 UARTC Registers

### (1) UCnCTL0 - UARTCn control register 0

The UCnCTL0 register is an 8-bit register that controls the UARTCn serial transfer operation.

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

**Address** UC0CTL0: FFFFA00<sub>H</sub>  
 UC1CTL0: FFFFA20<sub>H</sub>  
 UC2CTL0: FFFFA40<sub>H</sub>

**Initial Value** 10<sub>H</sub>. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
<b>UCnCTL0</b>	UCnPWR	UCnTXE	UCnRXE	UCnDIR	UCnPS1	UCnPS0	UCnCL	UCnSL
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Caution** Be sure to set the UCnPWR bit = 1 and the UCnRXE bit = 1 while the RXDCn pin is high level (when UCnRDL bit of UCnOP0 register = 0).  
 If the UCnPWR bit = 1 and the UCnRXE bit = 1 are set while the RXDCn pin is low level, reception will inadvertently start.

**Table 11-2 UCnCTL0 register contents (1/2)**

Bit position	Bit name	Function
7	UCnPWR	UARTCn Operation Control 0: Stops clock operation (UARTCn reset asynchronously) 1: LSB-first transfer Operating clock control and UARTCn asynchronous reset are performed with the UCnPWR bit. The TXDCn pin output is fixed to high level by setting the UCnPWR bit to 0.
6	UCnTXE	Transmission Operation Enable 0: Stops transmission operation 1: Enables transmission operation The TXDCn pin output is fixed to high level by setting the UCnTXE bit to 0. Since the UCnTXE bit is initialized by the operating clock, to initialize the transmission unit, set UCnTXE from 0 to 1, and 2 clocks later, the transmission enabled status is entered. <b>Note:</b> When UCnPWR bit = 0, the value written to the UCnTXE bit is ignored.
5	UCnRXE	Reception Operation Enable 0: Stops reception operation 1: Enables reception operation The receive operation is stopped by setting the UCnRXE bit to 0. Therefore, even if the prescribed data is transferred, no reception completion interrupt is output and the UARTCn reception data register (UCnRX) is not updated. Since the UCnRXE bit is synchronized using the operating clock, to initialize the reception unit, set UCnRXE from 0 to 1, and 2 clocks later, the reception enabled status is entered. <b>Note:</b> When UCnPWR bit = 0, the value written to the UCnRXE bit is ignored.
4	UCnDIR	Transfer Direction Selection 0: MSB-first transfer 1: Data is sent/received with LSB first <b>Note:</b> This bit can be rewritten only when UCnPWR = 0 or UCnTXE = UCnRXE = 0.

Table 11-2 UCnCTL0 register contents (2/2)

Bit position	Bit name	Function																						
3 2	UCnPS1 UCnPS0	<div>Parity Selection</div> <table><tr><th rowspan="2">UCnPS1</th><th rowspan="2">UCnPS0</th><th colspan="2">Parity Selection</th></tr><tr><th>During Transmission</th><th>During Reception</th></tr><tr><td>0</td><td>0</td><td>No parity output</td><td>Reception with no parity</td></tr><tr><td>0</td><td>1</td><td>0 parity output</td><td>Reception with 0 parity</td></tr><tr><td>1</td><td>0</td><td>Odd parity output</td><td>Odd parity check</td></tr><tr><td>1</td><td>1</td><td>Even parity output</td><td>Even parity check</td></tr></table> <div>For details of parity, see “Parity types and operations” on page 267.</div> <div><b>Note:</b> 1. If “Reception with 0 parity” is selected during reception, a parity check is not performed. Therefore, since the UCnSTR.UCnPE bit is not set, no error interrupt is output.</div> <div>2. When transmission and reception are performed in the LIN format, set the UCnPS[1:0] bits to 00B.</div> <div>3. This bit can be rewritten only when UCnPWR = 0 or UCnTXE = UCnRXE = 0.</div>	UCnPS1	UCnPS0	Parity Selection		During Transmission	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
UCnPS1	UCnPS0	Parity Selection																						
		During Transmission	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																					
1	UCnDL	<div>Data Character Length Specification</div> <div>0: 7 bits</div> <div>1: 8 bits</div> <div><b>Note:</b> This bit can be rewritten only when UCnPWR = 0 or UCnTXE = UCnRXE = 0.</div>																						
0	UCnSL	<div>Stop Bit Length Specification</div> <div>0: 1 bit</div> <div>1: 2 bits</div> <div><b>Note:</b> This bit can be rewritten only when UCnPWR = 0 or UCnTXE = UCnRXE = 0.</div>																						

**(2) UCnCTL1 - UARTCn control register 1**

This register controls the Baud Rate Generator. For details see “UCnCTL1 - UARTCn control register 1” on page 271.

**(3) UCnCTL2 - UARTCn control register 2**

This register controls the Baud Rate Generator. For details see “UCnCTL2 - UARTCn control register 2” on page 272.

**(4) UCnOPT0 - UARTCn option control register 0**

The UCnOPT0 register is an 8-bit register that controls the serial transfer operation of the UCRTCn register.

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

**Address** UC0OPT0: FFFFA03<sub>H</sub>  
 UC1OPT0: FFFFA23<sub>H</sub>  
 UC2OPT0: FFFFA43<sub>H</sub>

**Initial Value** 14<sub>H</sub>. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
<b>UCnOPT0</b>	UCnSRF	UCnSRT	UCnSTT	UCnSLS2	UCnSLS1	UCnSLS0	UCnTDL	UCnRDL
	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 11-3 UCnOPT0 register contents (1/2)**

Bit position	Bit name	Function
7	UCnSRF	SBF Reception Flag 0: When UCnCTL0.UCnPWR = 0 and UCnCTL0.UCnRXE = 0 are set to 1. Also upon normal end of SBF reception. 1: During SBF reception <ul style="list-style-type: none"> <li>SBF (Sync Brake Field) reception is judged during LIN communication.</li> <li>The UCnSRF bit is held high when a SBF reception error occurs, and then SBF reception is started again.</li> </ul>
6	UCnSRT	SBF Reception Trigger 0: – 1: SBF reception trigger <ul style="list-style-type: none"> <li>For SBF reception, set the UCnSRT bit (to 1) to enable reception.</li> </ul> <b>Note:</b> 1. When this bit is read, always “0” is returned. 2. Set the UCnSRT bit when UCnCTL0.UCnPWR = 1 and UCnCTL0.UCnRXE = 1.
5	UCnSTT	SBF Transmission Trigger 0: – 1: SBF transmission trigger <ul style="list-style-type: none"> <li>This bit triggers the SBF transmission during LIN communication.</li> </ul> <b>Note:</b> 1. When this bit is read, always “0” is returned. 2. Set the UCnSTT bit when UCnCTL0.UCnPWR = 1 and UCnCTL0.UCnTXE = 1. 3. Before starting the SBF transmission by UCnSTT = 1 it has to be ensured that no data transfer is ongoing: UCnSTR.UCnTSF = 0.

Table 11-3 UCnOPT0 register contents (2/2)

Bit position	Bit name	Function																																				
4 to 2	UCnSLS[2:0]	SBF Length Selection																																				
		<table><tr><th>UCnSLS2</th><th>UCnSLS1</th><th>UCnSLS0</th><th>SBF Length Selection</th></tr><tr><td>1</td><td>0</td><td>1</td><td>13-bit output (reset value)</td></tr><tr><td>1</td><td>1</td><td>0</td><td>14-bit output</td></tr><tr><td>1</td><td>1</td><td>1</td><td>15-bit output</td></tr><tr><td>0</td><td>1</td><td>0</td><td>16-bit output</td></tr><tr><td>0</td><td>0</td><td>1</td><td>17-bit output</td></tr><tr><td>0</td><td>0</td><td>0</td><td>18-bit output</td></tr><tr><td>0</td><td>1</td><td>1</td><td>19-bit output</td></tr><tr><td>1</td><td>0</td><td>0</td><td>20-bit output</td></tr></table>	UCnSLS2	UCnSLS1	UCnSLS0	SBF Length Selection	1	0	1	13-bit output (reset value)	1	1	0	14-bit output	1	1	1	15-bit output	0	1	0	16-bit output	0	0	1	17-bit output	0	0	0	18-bit output	0	1	1	19-bit output	1	0	0	20-bit output
		UCnSLS2	UCnSLS1	UCnSLS0	SBF Length Selection																																	
		1	0	1	13-bit output (reset value)																																	
		1	1	0	14-bit output																																	
		1	1	1	15-bit output																																	
		0	1	0	16-bit output																																	
		0	0	1	17-bit output																																	
		0	0	0	18-bit output																																	
		0	1	1	19-bit output																																	
1	0	0	20-bit output																																			
<b>Note:</b> Setting of the UCnSLS[2:0] bits is permitted only when UCnCTL0.UCnPWR = 0, or UCnCTL0.UCnTXE = 0.																																						
1	UCnTDL	Transmit Data Level 0: Normal output of transfer data 1: Inverted output of transfer data																																				
		<ul style="list-style-type: none"><li>The value of the TXDCn pin can be inverted using the UCnTDL bit.</li></ul> <b>Note:</b> Setting of the UCnTDL bit is permitted only when UCnCTL0.UCnPWR = 0, or UCnCTL0.UCnTXE = 0.																																				
0	UCnRDL	Receive Data Level 0: Normal input of transfer data 1: Inverted input of transfer data																																				
		<ul style="list-style-type: none"><li>The value of the RXDCn pin can be inverted using the UCnRDL bit.</li></ul> <b>Note:</b> Setting of the UCnRDL bit is permitted only when UCnCTL0.UCnPWR = 0 ,or UCnCTL0.UCnRXE = 0.																																				

**(5) UCnOPT1 - UARTCn option control register 1**

The UCnOPT1 register is an 8-bit register that controls the extension bit operation of the UARTCn.

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

**Address** UC0OPT1: FFFFFFFA0<sub>H</sub>  
 UC1OPT1: FFFFFFFA2<sub>H</sub>  
 UC2OPT1: FFFFFFFA4<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>UCnOPT1</b>	0	0	0	0	0	0	0	UCnEBE
	R	R	R	R	R	R	R	R/W

**Table 11-4 UCnOPT1 register contents**

Bit position	Bit name	Function
0	UCnEBE	Extension Bit Operation Enable 0: Extension bit operation disabled. Transfer data length set by UCnCTL0.UCnCL. 1: Extension bit operation enabled. <ul style="list-style-type: none"> <li>During extension bit operation a 9-th data bit is sent or received instead of the parity bit.</li> </ul> <b>Note:</b> Extension bit operation is only effective when the parity selection is set to no parity (UCnCTL0.UCnPS[1:0] = 00B), and the character length is set to 8 bits (UCnCTL0.UCnCL = 1). In all other cases the setting of UCnEBE bit is ignored.

**Table 11-5 Relation between UARTCn Register Settings and Data Format**

Register Bit Settings					Data Format				
UCnEBE	UCnPS1	UCnPS0	UCnCL	UCnSL	D0 - D6	D7	D8	D9	D10
0	0	0	0	0	Data	Stop			
			0	1	Data	Stop	Stop		
			1	0	Data	Data	Stop		
			1	1	Data	Data	Stop	Stop	
	other than 00B		0	0	Data	Parity	Stop		
			0	1	Data	Parity	Stop	Stop	
			1	0	Data	Data	Parity	Stop	
			1	1	Data	Data	Parity	Stop	Stop
1	0	0	0	0	Data	Stop			
			0	1	Data	Stop	Stop		
			1	0	Data	Data	Data <sup>a</sup>	Stop	
			1	1	Data	Data	Data <sup>a</sup>	Stop	Stop
	other than 00B		0	0	Data	Parity	Stop		
			0	1	Data	Parity	Stop	Stop	
			1	0	Data	Data	Parity	Stop	
			1	1	Data	Data	Parity	Stop	Stop

a) Insertion of extension bit

**(6) UARTCn option control register 2 (UCnOPT2)**

The UCnOPT2 register is an 8-bit register that displays the UARTCn reception status.

**Access** This register can be read only in 8-bit or 1-bit units.

**Address** UC0OPT1: FFFFA0B<sub>H</sub>  
 UC1OPT1: FFFFA2B<sub>H</sub>  
 UC2OPT1: FFFFA4B<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>UCnOPT2</b>	0	0	0	0	0	0	0	UCnRSF
	R	R	R	R	R	R	R	R

**Table 11-6 UCnOPT2 register contents**

Bit position	Bit name	Function
0	UCnRSF	<p>Receive Status Flag</p> <p>0: When UCnCTL0.UCnPWR = 0 or UCnCTL0.UCnRXE = 0 has been set, or when the stop bit has been detected.</p> <p>1: During reception, when the start bit has been detected.</p> <ul style="list-style-type: none"> <li>The UCnRSF flag is set (1) by the start bit detection, and it is cleared (0) by detection of the first stop bit condition.</li> </ul> <p><b>Note:</b> In case of a two stop bit setting (UCnCTL0.UCnSL = 1), the UCnRSF flag is cleared during the first stop bit timing, simultaneously with the reception complete interrupt timing (INTUCnR).</p>

**(7) UCnSTR - UARTCn status register**

The UCnSTR register is an 8-bit register that displays the UARTCn transfer status and reception error contents.

**Access** This register can be read/written in 8-bit or 1-bit units.  
Though the UCnTSF bit is a read-only bit, the UCnPE, UCnFE, and UCnOVE bits can be read and written. However, these bits can only be cleared by writing 0 to it; but cannot be set by writing 1 to it (even if 1 is written to them, the value is retained).

**Address** UC0STR: FFFFA04<sub>H</sub>  
UC1STR: FFFFA24<sub>H</sub>  
UC2STR: FFFFA44<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset, and when UCnCTL0.PWR = 0 is set.

	7	6	5	4	3	2	1	0
<b>UCnSTR</b>	UCnTSF	0	0	0	0	UCnPE	UCnFE	UCnOVE
	R	R	R	R	R	R/W	R/W	R/W

**Table 11-7 UCnSTR register contents (1/2)**

Bit position	Bit name	Function
7	UCnTSF	<p>Transfer Status Flag</p> <p>0: When UCnCTL0.UCnPWR = 0, or when UCnCTL0.UCnTXE = 0 has been set (transfer disabled), or when the current transfer is completed and no next data was written to be transferred from UCnTX, or when the SBF has been finished after SBF transmission trigger was set.</p> <p>1: When data to be transferred is written to UCnTX register, or when SBF transmission trigger bit is set (UCnSTT = 1).</p> <p><b>Note:</b> The UCnTSF bit is always 1 when performing continuous transmission. When initializing the transmission unit, check that UCnTSF = 0 before performing initialization. The transmit data is not guaranteed when initialization is performed while UCnTSF = 1.</p>
2	UCnPE	<p>Parity Error Flag</p> <p>0: When UCnCTL0.UCnPWR = 0, or when UCnCTL0.UCnRXE = 0 has been set (reception disabled), or when 0 has been written</p> <p>1: When parity of data and parity bit do not match during reception.</p> <p><b>Note:</b> 1. The operation of the UCnPE bit is controlled by the settings of the UCnCTL0.UCnPS[1:0] bits.</p> <p>2. The UCnPE bit can be read and written, but it can only be cleared by writing 0 to it, but it cannot be set by writing 1 to it. When 1 is written to this bit, the hold status is entered.</p>

**Table 11-7 UCnSTR register contents (2/2)**

Bit position	Bit name	Function
1	UCnFE	<p>Framing Error Flag</p> <p>0: When UCnCTL0.UCnPWR = 0, or when UCnCTL0.UCnRXE = 0 has been set (reception disabled), or when 0 has been written</p> <p>1: When no stop bit is detected during reception.</p> <p><b>Note:</b> 1. Only the first bit of the receive data stop bits is checked, regardless of the value of the UCnCTL0.UCnSL bit.</p> <p>2. The UCnFE bit can be read and written, but it can only be cleared by writing 0 to it, but it cannot be set by writing 1 to it. When 1 is written to this bit, the hold status is entered.</p>
0	UCnOVE	<p>Overrun Error Flag</p> <p>0: When UCnCTL0.UCnPWR = 0, or when UCnCTL0.UCnRXE = 0 has been set (reception disabled), or when 0 has been written</p> <p>1: When data has been received into the UCnRX register and the next receive operation is completed before that receive data has been read.</p> <p><b>Note:</b> 1. When an overrun error occurs, the data is discarded without the next receive data being written to the receive buffer.</p> <p>2. The UCnOVE bit can be read and written, but it can only be cleared by writing 0 to it, but it cannot be set by writing 1 to it. When 1 is written to this bit, the hold status is entered.</p>



**(8) UCnRX- UARTCn receive data register**

The UCnRX register is a 16-bit buffer register that stores parallel data converted by receive shift register. It is overlaid by an 8-bit register UCnRXL on the lower 8 bits, which stores the lower byte of the received data.

The data stored in the receive shift register is transferred to the UCnRX register upon completion of reception of one data frame.

When extension bit operation is enabled (UCnOPT1.UCnEBE = 1) the 9th data bit is received in bit 8 of the UCnRX register. When the extension bit operation is disabled (UCnEBE bit = 0) the data bits are received in the lower byte of the UCnRX register. The lower byte can be read also by 8-bit access of the UCnRXL register.

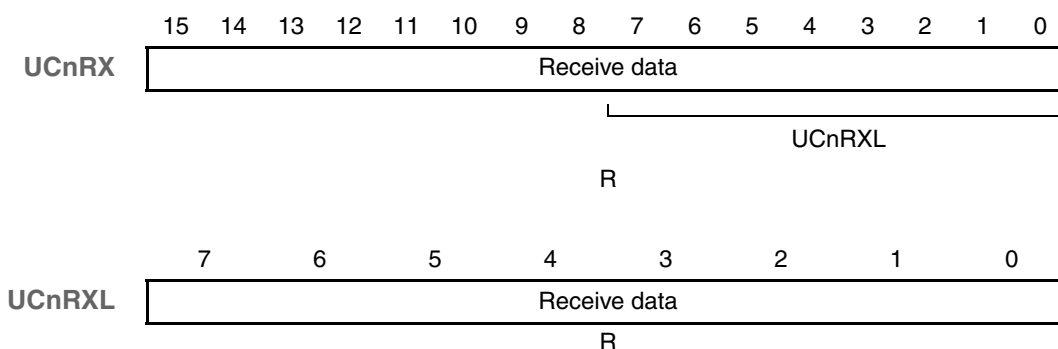
During LSB-first reception when the data length has been specified as 7 bits and the extension bit operation is disabled, the receive data is transferred to bits 6 to 0 of the UCnRXL register and the MSB always becomes 0. During MSB-first reception, the receive data is transferred to bits 7 to 1 of the UCnRXL register and the LSB always becomes 0.

When an overrun error (UCnOVE bit = 1) occurs, the receive data at this time is not transferred to the UCnRX and UCnRXL register respectively.

**Access** This register can be read-only in 16-bit units as UCnRX register, or in 8-bit units as UCnRXL register for lower 8 bits.

**Address** UC0RX, UC0RXL: FFFFA06<sub>H</sub>  
 UC1RX, UC1RXL: FFFFA26<sub>H</sub>  
 UC2RX, UC2RXL: FFFFA46<sub>H</sub>

**Initial Value** 1FF<sub>H</sub> in case of UCnRX register, and  
 FF<sub>H</sub> in case of UCnRXL register, respectively,  
 by any reset, and when UCnCTL0.PWR = 0 is set.



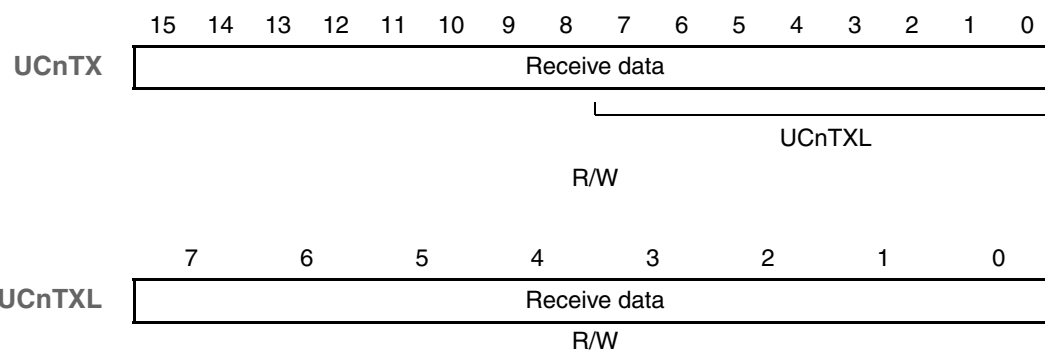
**(9) UCnTX - UARTCn transmit data register**

The UCnTX register is a 16-bit buffer register used to set transmit data. It is overlaid by an 8-bit register UCnTXL on the lower 8 bits. The UCnTXL register is used for setting the transmit data when 7-bit or 8-bit data character length is specified (UCnEBE bit = 0).

**Access** This register can be read or written in 16-bit units as UCnTX register, or in 8-bit units as UCnRXL register for lower 8 bits.

**Address** UC0TX, UC0TXL: FFFFA08<sub>H</sub>  
 UC1TX, UC1TXL: FFFFA28<sub>H</sub>  
 UC2TX, UC2TXL: FFFFA48<sub>H</sub>

**Initial Value** 1FF<sub>H</sub> in case of UCnTX register, and  
 FF<sub>H</sub> in case of UCnTXL register, respectively,  
 by any reset.



## 11.4 Interrupt Request Signals

The following three interrupt request signals are generated from UARTCn:

- Reception complete interrupt request signal (INTUCnR)
- Receive error interrupt request signal (INTUCnRE)
- Transmission enable interrupt request signal (INTUCnT)

### (1) Reception complete interrupt request signal (INTUCnR)

A reception complete interrupt request signal is output when data is shifted into the receive shift register and transferred to the UCnRX register in the reception enabled status.

In case of erroneous reception, the reception error interrupt INTUanRE is generated instead of INTUCnR.

No reception complete interrupt request signal is generated in the reception disabled status.

### (2) Receive error interrupt request signal (INTUCnRE)

A receive error interrupt request is generated if an error condition occurred during reception, as reflected by UCnSTR.UCnPE (parity error flag), UCnSTR.UCnFE (framing error flag), UCnSTR.UCnOVE (overrun error flag).

Note that INTUCnR and INTUCnRE do exclude each other: upon correct reception of data only INTUCnR is generated. In case of a reception error INTUCnRE is generated only.

### (3) Transmission enable interrupt request signal (INTUCnT)

If transmit data is transferred from the UCnTX register to the UCRTCn transmit shift register with transmission enabled, the transmission enable interrupt request signal is generated.

## 11.5 Operation

### 11.5.1 Data format

Full-duplex serial data reception and transmission is performed.

As shown in the figures below, 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 UCnCTL0 register. UARTCn features additionally the extension bit operation for a ninth transfer data bit, which can be specified in the UCnOPT1 register.

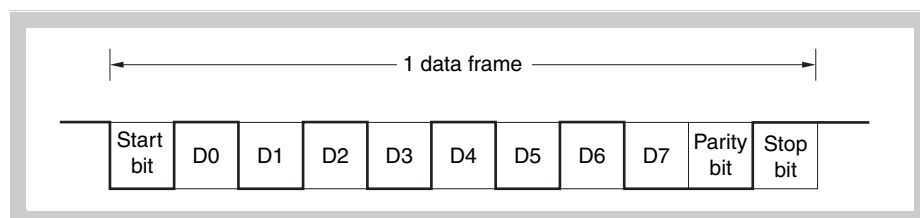
Moreover, control of UART output/inverted output for the TXDCn bit is performed using the UCnOPT0.UCnTDL bit.

- Start bit..... 1 bit
- Character bits..... 7 bits/8 bits/9 bits
- Parity bit ..... Even parity/odd parity/0 parity/no parity
- Stop bit..... 1 bit/2 bits

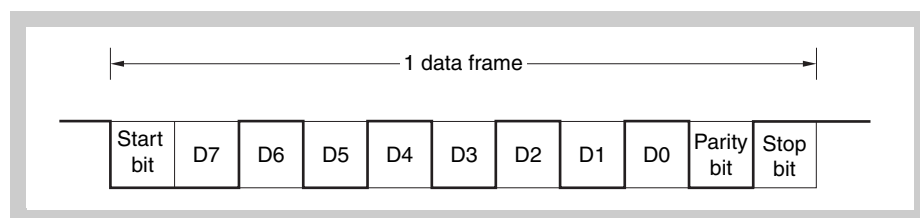
**Note** Extension bit operation presumes no parity setting.

#### (1) UARTC transmit/receive data format

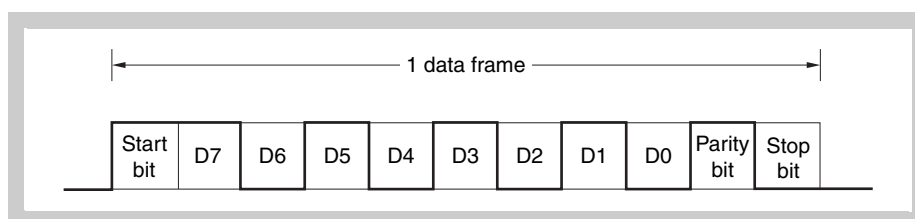
##### (a) 8-bit data length, LSB first, even parity, 1 stop bit, transfer data: 55H



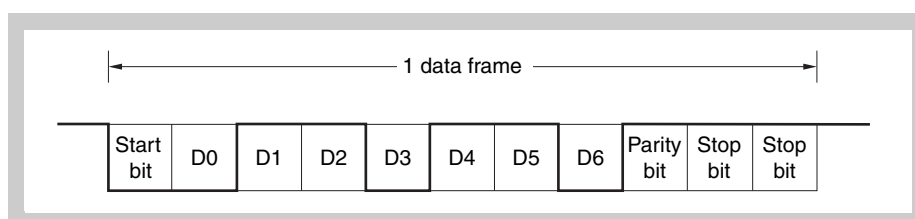
##### (b) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H



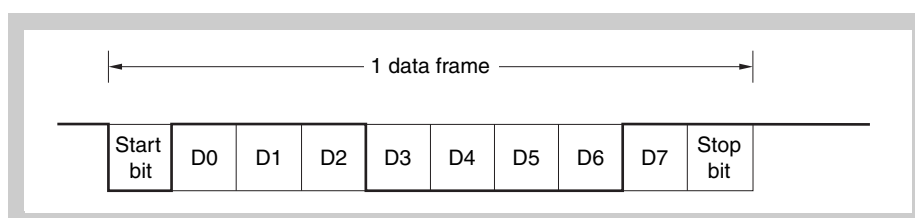
- (c) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H, TXDCn inversion



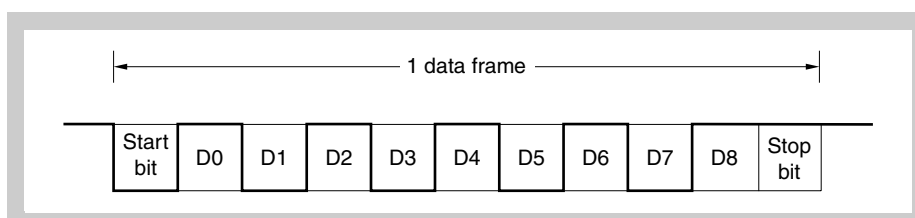
- (d) 7-bit data length, LSB first, odd parity, 2 stop bits, transfer data: 36H



- (e) 8-bit data length, LSB first, no parity, 1 stop bit, transfer data: 87H



- (f) 9-bit data length, LSB first, no parity, 1 stop bit, transfer data: 155H



### 11.5.2 SBF transmission/reception format

The UARTC has an SBF (Sync Break Field) transmission/reception control function to enable use of the LIN function.

**About LIN** LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol intended to aid the cost reduction of an automotive network.

LIN communication is single-master communication, and up to 15 slaves can be connected to one master.

The LIN slaves are used to control the switches, actuators, and sensors, and these are connected to the LIN master via the LIN network.

Normally, the LIN master is connected to a network such as CAN (Controller Area Network).

In addition, the LIN bus uses a single-wire method and is connected to the nodes via a transceiver that complies with ISO9141.

In the LIN protocol, the master transmits a frame with baud rate information and the slave receives it and corrects the baud rate error. Therefore, communication is possible when the baud rate error in the slave is  $\pm 15\%$  or less.

Figure 11-2 and Figure 11-3 outline the transmission and reception manipulations of LIN.

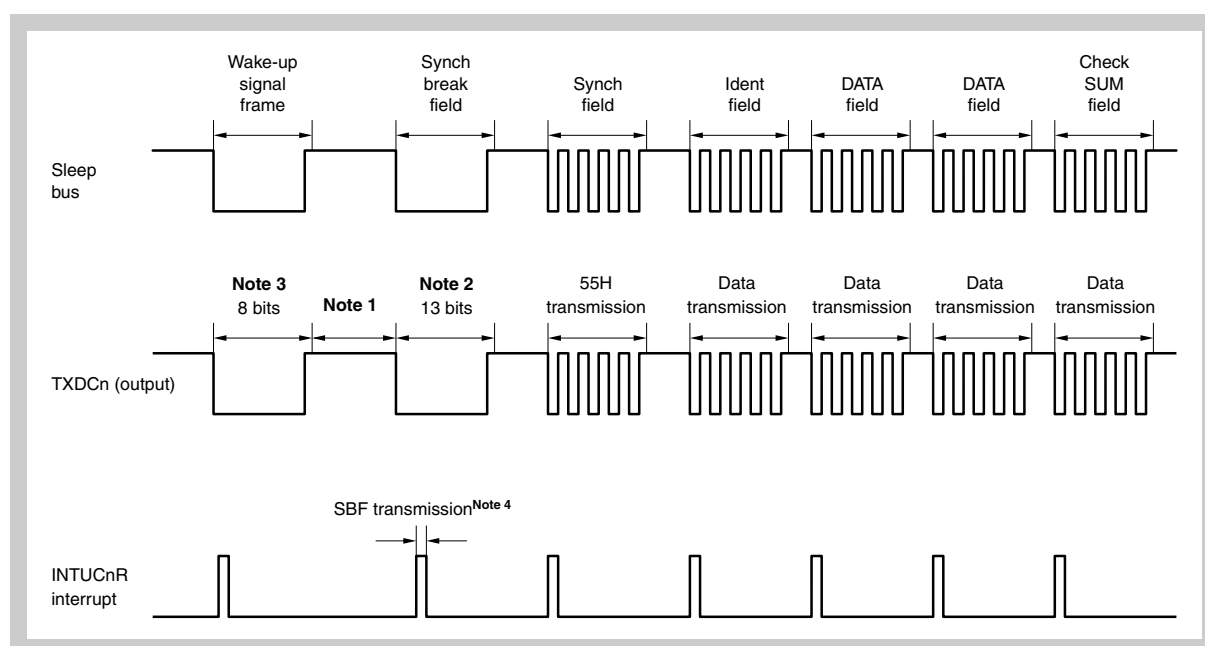
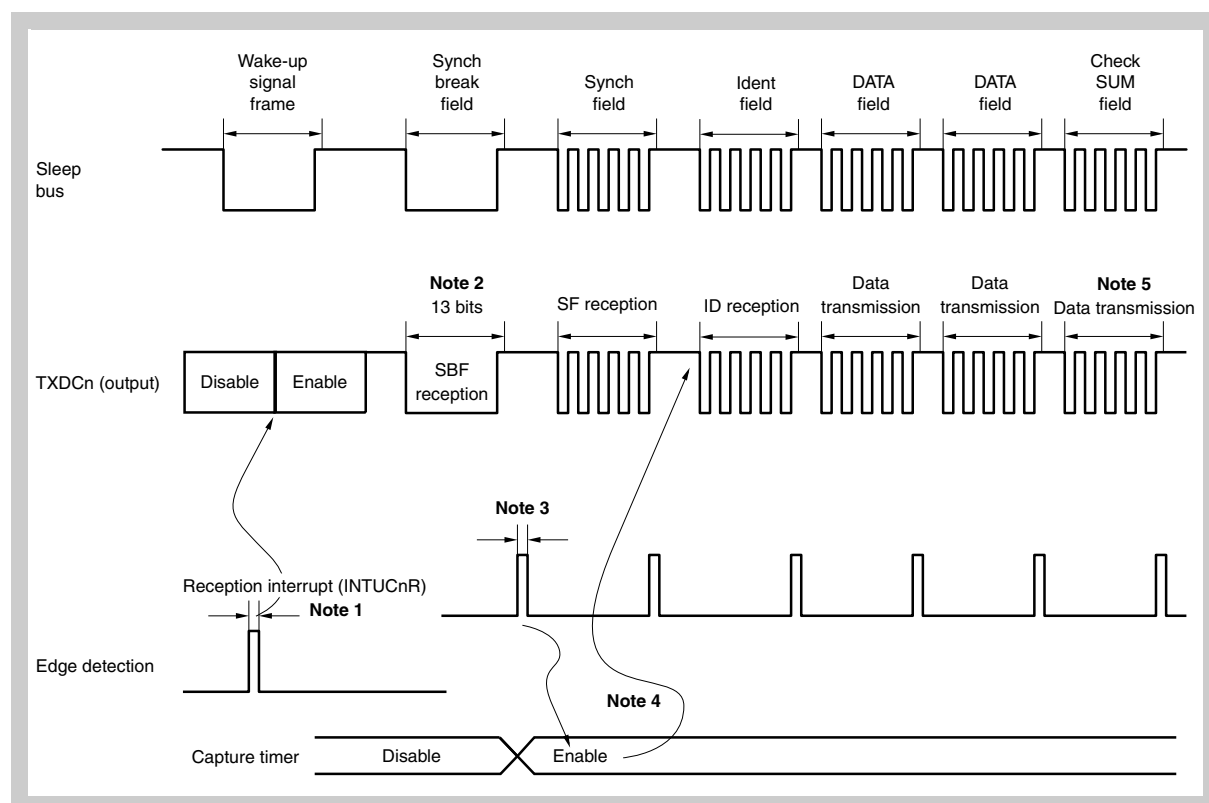


Figure 11-2 LIN transmission manipulation outline

- Note**
1. The interval between each field is controlled by software.
  2. SBF output is performed by hardware. The output width is the bit length set by the UCnOPT0.UCnSLS2 to UCnOPT0.UCnSLS0 bits. If even finer output width adjustments are required, such adjustments can be performed using the UCnCTLn.UCnBRS7 to UCnCTLn.UCnBRS0 bits.
  3. 80H transfer in the 8-bit mode is substituted for the wakeup signal frame.
  4. A transmission enable interrupt request signal (INTUCnT) is output at the start of each transmission. The INTUCnT signal is also output at the start of each SBF transmission.



**Figure 11-3** LIN reception manipulation outline

- Note**
1. The wakeup signal is sent by the pin edge detector, UARTCn is enabled, and the SBF reception mode is set.
  2. The receive operation is performed until detection of the stop bit. Upon detection of SBF reception of 11 or more bits, normal SBF reception end is judged, and an interrupt signal is output. Upon detection of SBF reception of less than 11 bits, an SBF reception error is judged, no interrupt signal is output, and the mode returns to the SBF reception mode.
  3. If SBF reception ends normally, an interrupt request signal is output. The timer is enabled by an SBF reception complete interrupt. Moreover, error detection for the UCnSTR.UCnOVE, UCnSTR.UCnPE, and UCnSTR.UCnFE bits is suppressed and UART communication error detection processing and UARTCn receive shift register and data transfer of the UCnRX register are not performed. The UARTCn receive shift register holds the initial value, FFH.
  4. The RXDCn pin is connected to TI (capture input) of the timer, the transfer rate is calculated, and the baud rate error is calculated. The value of the UCnCTL2 register obtained by correcting the baud rate error after dropping UARTC enable is set again, causing the status to become the reception status.
  5. Check-sum field distinctions are made by software. UARTCn is initialized following CSF reception, and the processing for setting the SBF reception mode again is performed by software.

### 11.5.3 SBF transmission

When the UCnCTL0.UCnPWR bit = UCnCTL0.UCnTXE bit = 1, the transmission enabled status is entered, and SBF transmission is started by setting (to 1) the SBF transmission trigger (UCnOPT0.UCnSTT bit).

Thereafter, a low level width of bits 13 to 20 specified by the UCnOPT0.UCnSLS2 to UCnOPT0.UCnSLS0 bits is output. A transmission enable interrupt request signal (INTUCnT) is generated upon SBF transmission start. Following the end of SBF transmission, the UCnSTT bit is automatically cleared. Thereafter, the UART transmission mode is restored.

Transmission is suspended until the data to be transmitted next is written to the UCnTX register, or until the SBF transmission trigger (UCnSTT bit) is set.

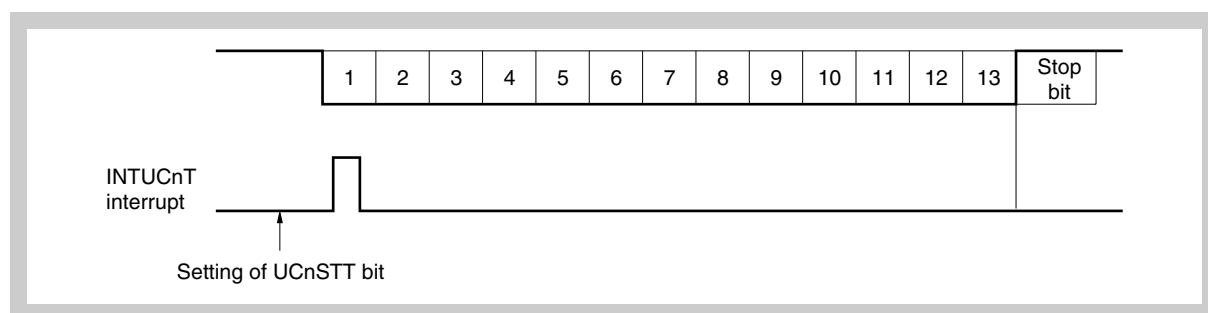


Figure 11-4 SBF transmission

### 11.5.4 SBF reception

The reception enabled status is achieved by setting the UCnCTL0.UCnPWR bit to 1 and then setting the UCnCTL0.UCnRX bit to 1.

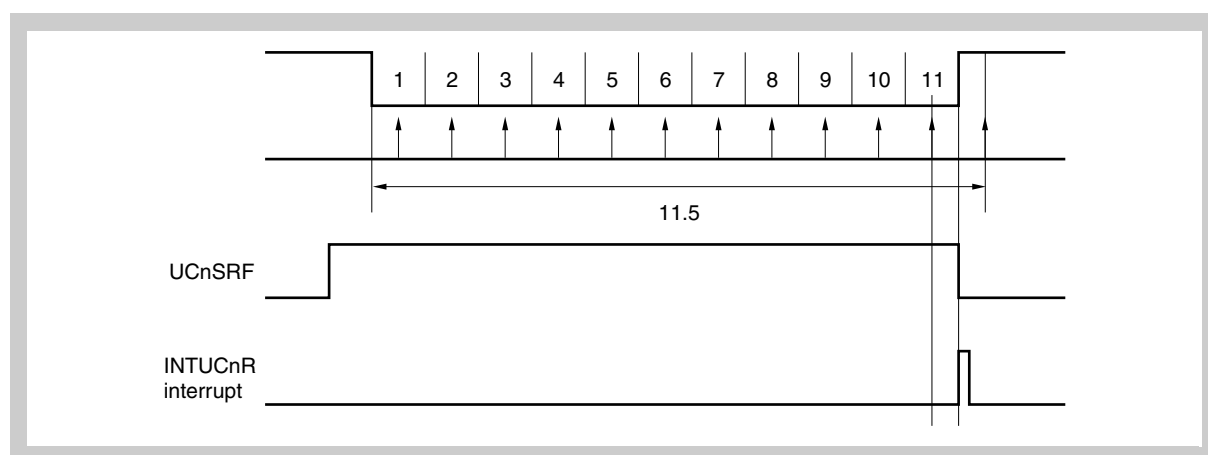
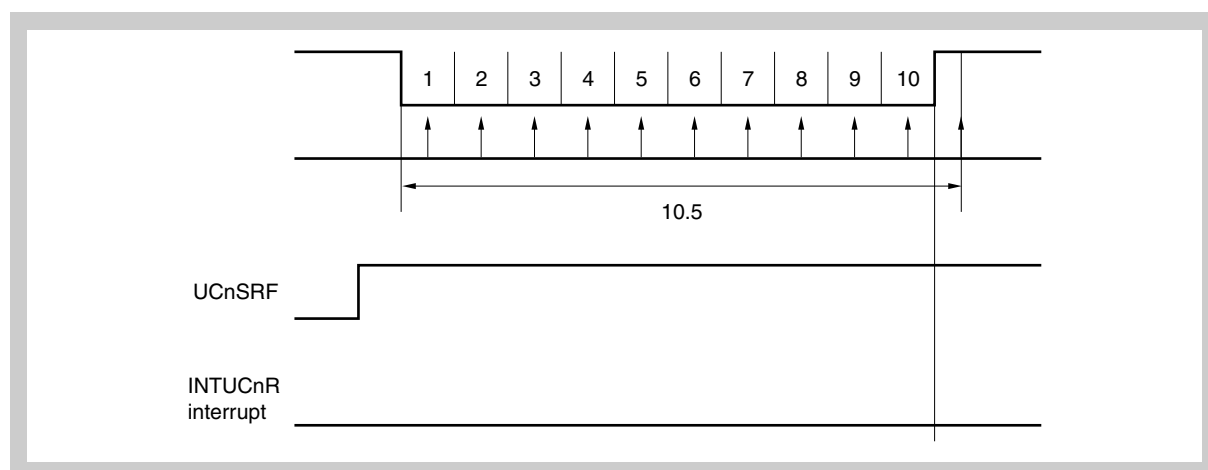
The SBF reception wait status is set by setting the SBF reception trigger (UCnOPT0.UCnSTR bit) to 1.

In the SBF reception wait status, similarly to the UART reception wait status, the RXDCn pin is monitored and start bit detection is performed.

Following detection of the start bit, reception is started and the internal counter counts up according to the set baud rate.

When a stop bit is received, if the SBF width is 11 or more bits, normal processing is judged and a reception complete interrupt request signal (INTUCnR) is output. The UCnOPT0.UCnSRF bit is automatically cleared and SBF reception ends. Error detection for the UCnSTR.UCnOVE, UCnSTR.UCnPE, and UCnSTR.UCnFE bits is suppressed and UART communication error detection processing is not performed. Moreover, data transfer of the UARTCn reception shift register and UCnRX register is not performed and FFH, the initial value, is held. If the SBF width is 10 or fewer bits, reception is terminated as error processing without outputting an interrupt, and the SBF reception mode is returned to. The UCnSRF bit is not cleared at this time.



**(a) Normal SBF reception (detection of stop bit in more than 10.5 bits)****(b) SBF reception error (detection of stop bit in 10.5 or fewer bits)**

### 11.5.5 UART transmission

The transmission enabled status is set by setting UCnCTL0.UCnTXE to 1, after UCnCTL0.UCnPWR was set to 1, and transmission is started by writing transmit data to the UCnTX register. The start bit, parity bit, and stop bit are automatically added.

The data in the UCnTX register is transferred to the UARTCn transmit shift register upon the start of the transmit operation.

A transmission enable interrupt (INTUCnT) is generated upon completion of transmission of the data of the UCnTX register to the UARTCn transmit shift register, and thereafter the contents of the UARTCn transmit shift register are output to the TXDCn pin LSB first.

Write of the next transmit data to the UCnTX register is enabled by generating the INTUCnT signal.

Continuous transmission is enabled by writing the data to be transmitted next to the UCnTX register during transfer.

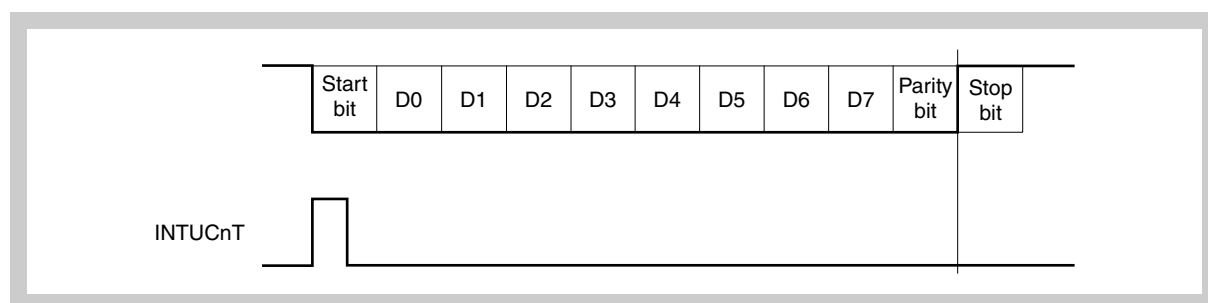


Figure 11-5 UART transmission

**Note** LSB first

### 11.5.6 Continuous transmission procedure

UARTCn can write the next transmit data to the UCnTX register when the UARTCn transmit shift register starts the shift operation. The transfer timing of the UARTCn transmit shift register can be judged from the transmission enable interrupt (INTUCnT). Transmission can be performed without interruption even during interrupt processing following the transmission of 1 data frame via the INTUCnT signal, and an efficient communication rate can thus be achieved.

During continuous transmission, overrun (the completion of the next transmission before the first transmission completion processing has been executed) may occur.

An overrun can be detected by incorporating a program that can count the number of transmit data and by referencing transfer status flag (UCnSTR.UCnTSF).

**Caution** During continuous transmission execution, perform initialization after checking that the UCnSTR.UCnTSF bit is 0. The transmit data cannot be guaranteed when initialization is performed while the UCnTSF bit is 1.

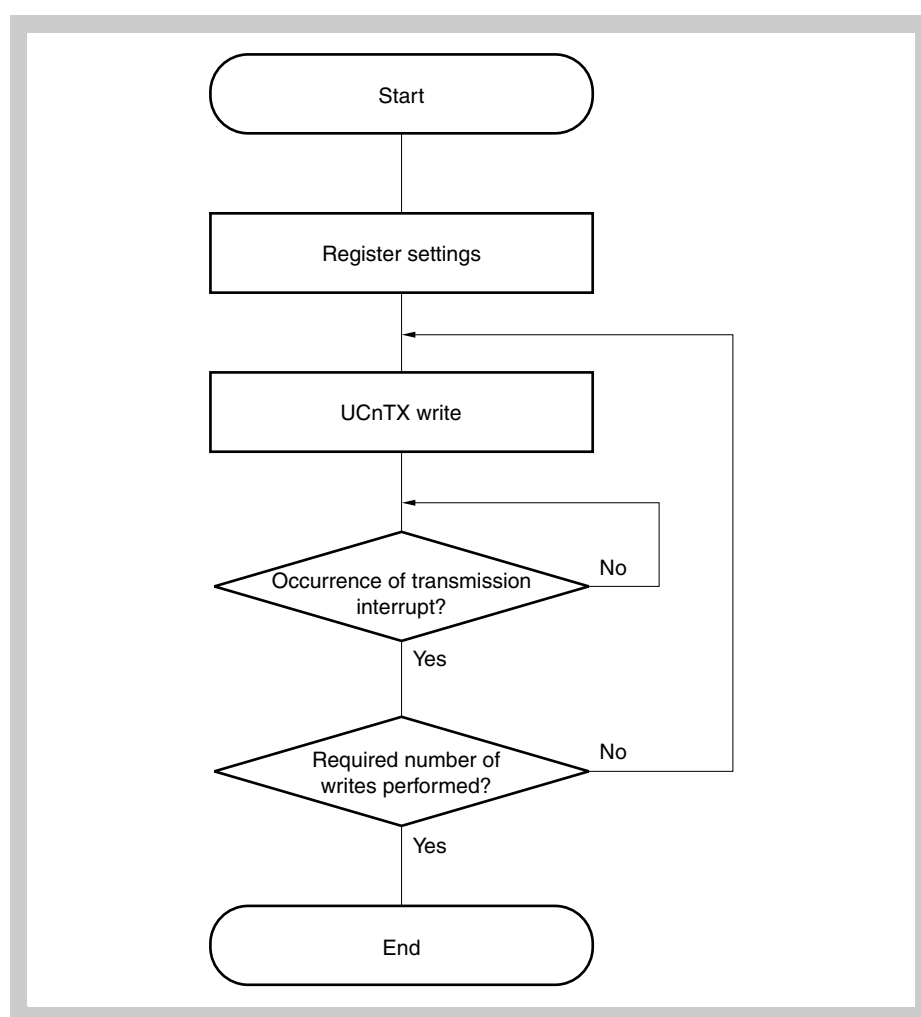


Figure 11-6 Continuous transmission processing flow

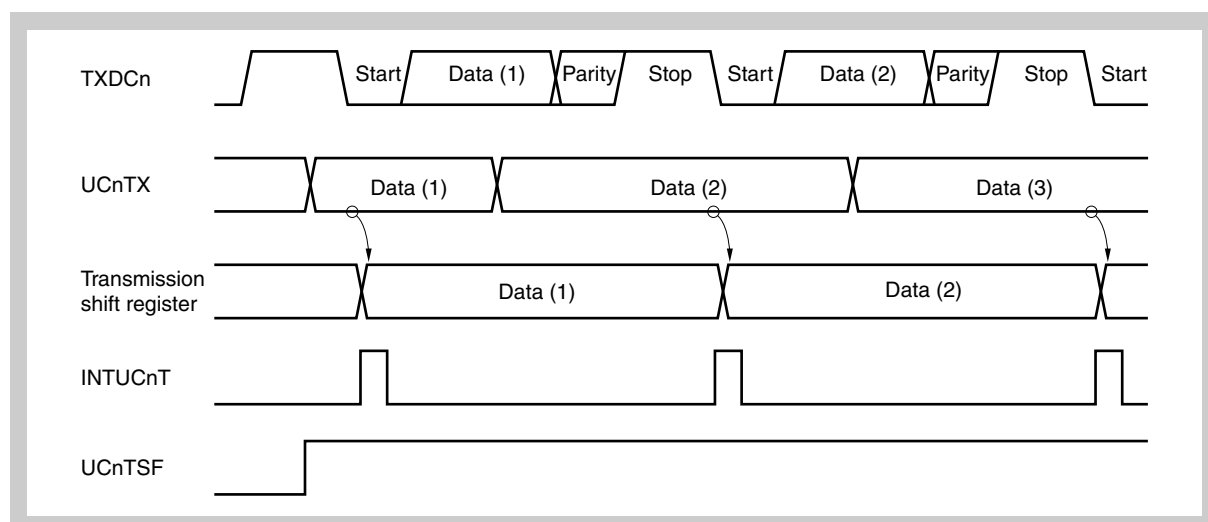


Figure 11-7 Continuous transmission operation timing —transmission start

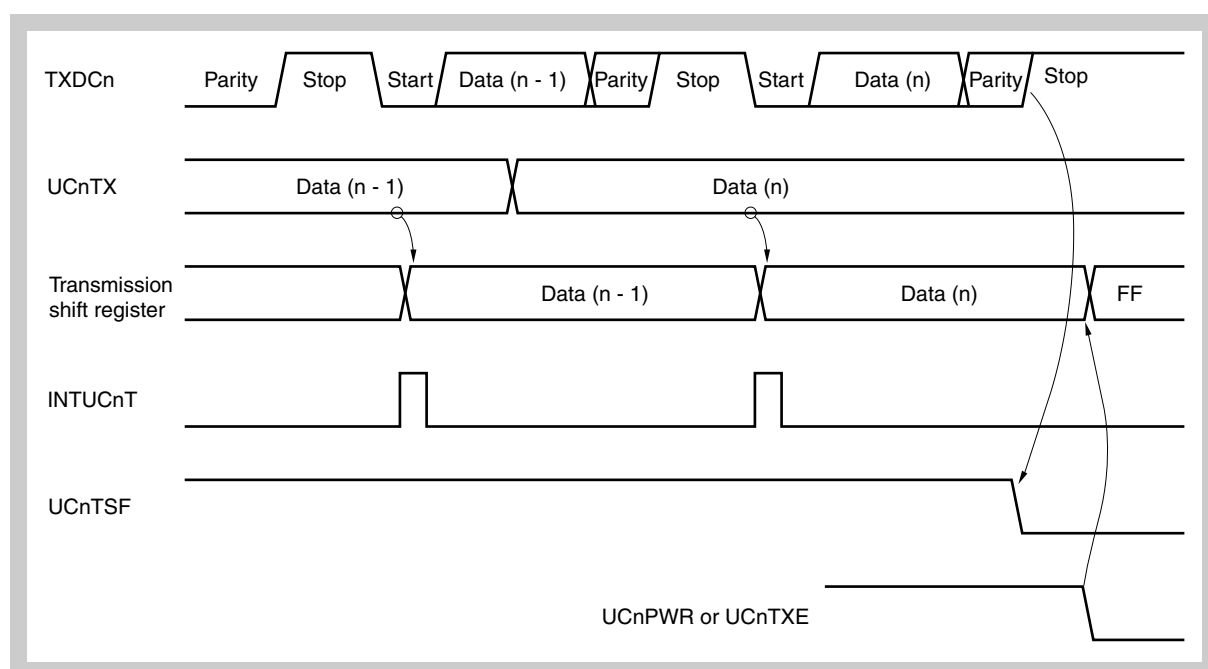


Figure 11-8 Continuous transmission operation timing—transmission end

### 11.5.7 UART reception

The reception wait status is set by setting UCnCTL0.UCnPWR to 1 and then setting UCnCTL0.UCnRXE to 1. In the reception wait status, the RXDCn pin is monitored and start bit detection is performed.

Start bit detection is performed using a two-step detection routine.

First, an 8-bit counter starts upon detection of the falling edge of the RXDCn pin. When the 8-bit counter has counted the UCnCTL2 register setting value, the level of the RXDCn pin is monitored again (corresponds to the — mark in Figure 15-19). If the RXDCn pin is low level at this time too, a start bit is recognized. After a start bit has been recognized, the receive operation starts, and serial data is saved to the UARTCn receive shift register according to the set baud rate. Additionally the flag UCnOPT2.UCnRSF is set (1) to indicate the receive operation status.

When the reception complete interrupt (INTUCnR) is output upon reception of the stop bit, the data of the UARTCn receive shift register is written to the UCnRX register, and the UCnRSF flag is cleared (0) simultaneously. However, if an overrun error occurs (UCnOVE bit = 1), the receive data at this time is not written to the UCnRX register, and a reception error interrupt (INTUCnRE) is output.

Even if a parity error (UCnPE bit = 1) or a framing error (UCnFE bit = 1) occurs during reception, reception continues until the stop bit reception position, but a reception error interrupt (INTUCnRE) is output following reception completion.

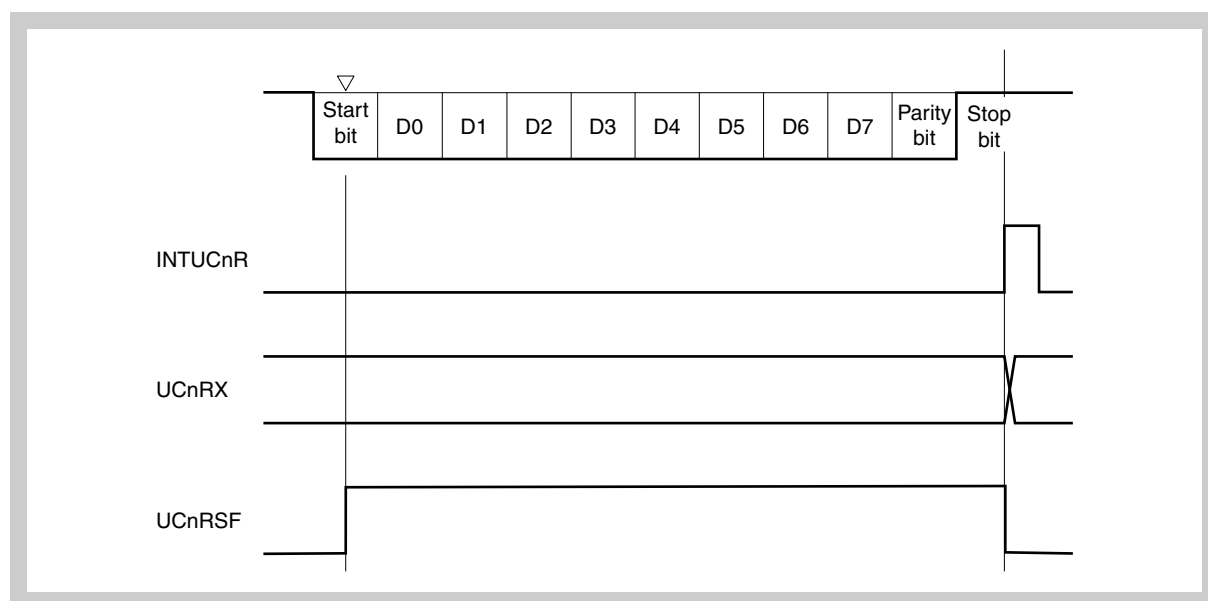


Figure 11-9 UART reception

- Caution**
1. Be sure to read the UCnRX register even when a reception error occurs. If the UCnRX 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 UCnRX register after the reception complete interrupt request signal (INTUCnR) has been generated, and clear the UCnPWR or UCnRXE bit to 0. If the UCnPWR or UCnRXE bit is cleared to 0 before the INTUCnR signal is generated, the read value of the UCnRX register cannot be guaranteed.
  4. If receive completion processing (INTUCnR signal generation) of UARTCn and the UCnPWR bit = 0 or UCnRXE bit = 0 conflict, the INTUCnR signal may be generated in spite of these being no data stored in the UCnRX register.  
To complete reception without waiting INTUCnR signal generation, be sure to clear (0) the interrupt request flag (UCnRIF) of the UCnRIC register, after setting (1) the interrupt mask flag (UCnRMK) of the interrupt control register (UCnRIC) and then set (1) the UCnPWR bit = 0 or UCnRXE bit = 0.
-

### 11.5.8 Reception errors

Errors during a receive operation are of three types: parity errors, framing errors, and overrun errors. Data reception result error flags are set in the UCnSTR register and a reception error interrupt request signal (INTUCnRE) is output when an error occurs.

It is possible to ascertain which error occurred during reception by reading the contents of the UCnSTR register.

Clear the reception error flag by writing 0 to it after reading it.

**Table 11-8 Reception error causes**

Error flag	Reception error	Cause
UCnPE	Parity error	Received parity bit does not match the setting
UCnFE	Framing error	Stop bit not detected
UCnOVE	Overrun error	Reception of next data completed before data was read from receive buffer

- Caution**
1. Be sure to read the UCnRX register even when a reception error occurs. If the UCnRX 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.

**Note** Note that even in case of a parity or framing error, data is transferred from the receive shift register to the receive data register UCnRX. Consequently the data from UCnRX must be read. Otherwise an overrun error UCnSTR.UCnOVE will occur at reception of the next data.

In case of an overrun error, the receive shift register data is not transferred to UCnRX, thus the previous data is not overwritten.

### 11.5.9 Parity types and operations

- Caution** When using the LIN function, fix the UCnPS1 and UCnPS0 bits of the UCnCTL0 register to 00.

The parity bit is used to detect bit errors in the communication data. Normally the same parity is used on the transmission side and the reception side.

In the case of even parity and odd parity, it is possible to detect odd-count bit errors. In the case of 0 parity and no parity, errors cannot be detected.

**(1) Even parity**

- 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
- 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.

**(2) Odd parity**

- 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
- 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.

**(3) 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.

**(4) 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.



### 11.5.10 Receive data noise filter

This filter samples the RXDCn pin using the selected clock ( $f_{XCLK}$ ).

When the same sampling value is read twice, the match detector output changes and the RXDCn signal is sampled as the input data.

Furthermore the processing that goes on within the receive operation is delayed by 2 clocks in relation to the external signal status (refer to the circuit as shown in *Figure 11-10*).

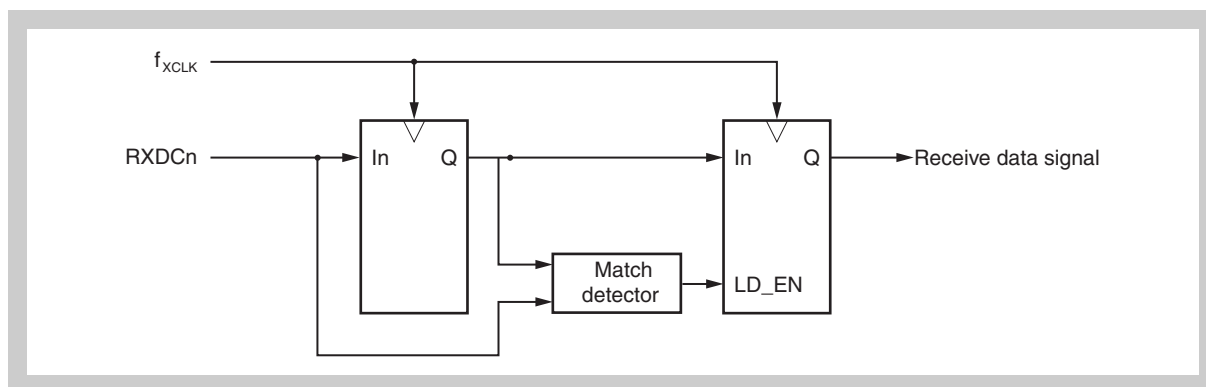


Figure 11-10 Noise filter circuit

## 11.6 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 UARTCn. 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

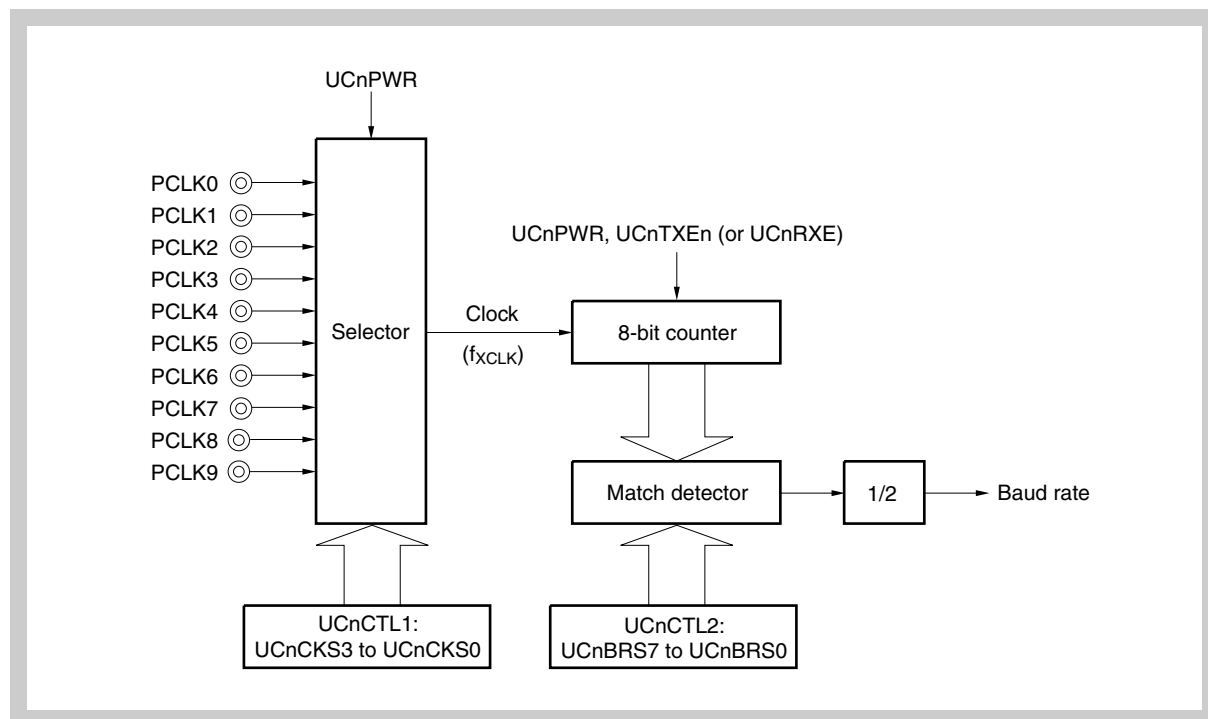


Figure 11-11 Configuration of baud rate generator

**Note** PCLK0 to PCLK9: peripheral clocks (refer to “Clock Generator” on page 148):

PCLK	0	1	2	3	4	5	6	7	8	9
f [MHz]	32	16	8	4	2	1	0.5	0.25	0.125	0.0625

#### (a) Base clock (Clock)

When the UCNCTL0.UCNPWR bit is 1, the clock selected by the UCNCTL1.UCnCKS[3:0] bits is supplied to the 8-bit counter. This clock is called the base clock. When the UCNPWR bit = 0, the clock is fixed to the low level.

#### (b) Serial clock generation

A serial clock can be generated by setting the UCNCTL1 register and the UCNCTL2 register.

The base clock is selected by bits UCNCTL1.UCnCKS[3:0] bits.

The frequency division value for the 8-bit counter can be set using the UCNCTL2.UCnBRS[7:0] bits.

**(2) UCnCTL1 - UARTCn control register 1**

The UCnCTL1 register is an 8-bit register that selects the UARTCn base clock ( $f_{XCLK}$ ).

**Access** This register can be read/written in 8-bit units.

**Address** UC0CTL1: FFFFA01<sub>H</sub>  
 UC1CTL1: FFFFA21<sub>H</sub>  
 UC2CTL1: FFFFA41<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>UCnCTL1</b>	0	0	0	0	UCnCKS3	UCnCKS2	UCnCKS1	UCnCKS0
	R	R	R	R	R/W	R/W	R/W	R/W

**Caution** Clear the UCnCTL0.UCnPWR bit to 0 before rewriting the UCnCTL1 register.

**Table 11-9 UCnCTL1 register contents**

Bit position	Bit name	Function																																																												
3 to 0	UCnCKS[3:0]	Base clock selection																																																												
		<table><tr><th>UCnCKS3</th><th>UCnCKS2</th><th>UCnCKS1</th><th>UCnCKS0</th><th>Base clock (<math>f_{XCLK}</math>) selection</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>PCLK0 (32 MHz)</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>PCLK1 (16 MHz)</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>PCLK2 (8 MHz)</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>PCLK3 (4 MHz)</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>PCLK4 (2 MHz)</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>PCLK5 (1 MHz)</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>PCLK6 (500 KHz)</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>PCLK7 (250 KHz)</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>PCLK8 (125 KHz)</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>PCLK9 (62.5 KHz)</td></tr><tr><td colspan="4">all others</td><td>setting prohibited</td></tr></table>	UCnCKS3	UCnCKS2	UCnCKS1	UCnCKS0	Base clock ( $f_{XCLK}$ ) selection	0	0	0	0	PCLK0 (32 MHz)	0	0	0	1	PCLK1 (16 MHz)	0	0	1	0	PCLK2 (8 MHz)	0	0	1	1	PCLK3 (4 MHz)	0	1	0	0	PCLK4 (2 MHz)	0	1	0	1	PCLK5 (1 MHz)	0	1	1	0	PCLK6 (500 KHz)	0	1	1	1	PCLK7 (250 KHz)	1	0	0	0	PCLK8 (125 KHz)	1	0	0	1	PCLK9 (62.5 KHz)	all others				setting prohibited
		UCnCKS3	UCnCKS2	UCnCKS1	UCnCKS0	Base clock ( $f_{XCLK}$ ) selection																																																								
		0	0	0	0	PCLK0 (32 MHz)																																																								
		0	0	0	1	PCLK1 (16 MHz)																																																								
		0	0	1	0	PCLK2 (8 MHz)																																																								
		0	0	1	1	PCLK3 (4 MHz)																																																								
		0	1	0	0	PCLK4 (2 MHz)																																																								
		0	1	0	1	PCLK5 (1 MHz)																																																								
		0	1	1	0	PCLK6 (500 KHz)																																																								
		0	1	1	1	PCLK7 (250 KHz)																																																								
		1	0	0	0	PCLK8 (125 KHz)																																																								
		1	0	0	1	PCLK9 (62.5 KHz)																																																								
		all others				setting prohibited																																																								
<ul style="list-style-type: none"><li>For details of peripheral clocks (PCLK0 to PCLK9) refer to chapter “Clock Generator” on page 148.</li></ul>																																																														

**(3) UCnCTL2 - UARTCn control register 2**

The UCnCTL2 register is an 8-bit register that specifies the divisor to control the baud rate (serial transfer speed) clock of UARTCn.

**Access** This register can be read/written in 8-bit units.

**Address** UC0CTL2: FFFFA02<sub>H</sub>  
 UC1CTL2: FFFFA22<sub>H</sub>  
 UC2CTL2: FFFFA42<sub>H</sub>

**Initial Value** FF<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>UCnCTL2</b>	UCnBRS7	UCnBRS6	UCnBRS5	UCnBRS4	UCnBRS3	UCnBRS2	UCnBRS1	UCnBRS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Caution** Clear the UCnCTL0.UCnPWR bit to 0, or clear the UCnCTL0.UCnTXE and UCnCTL0.UCnRXE bits to 0 before rewriting the UCnCTL2 register.

**Table 11-10 UCnCTL2 register contents**

Bit position	Bit name	Function									
7 to 0	UCnBRS[7:0]	Baud rate selection									
		UCn BRS7	UCn BRS6	UCn BRS5	UCn BRS4	UCn BRS3	UCn BRS2	UCn BRS1	UCn BRS0	Default (k)	Serial clock <sup>a</sup>
		0	0	0	0	0	0	-	-	-	Setting prohibited
		0	0	0	0	0	1	0	0	4	f <sub>XCLK</sub> /4
		0	0	0	0	0	1	0	1	5	f <sub>XCLK</sub> /5
		0	0	0	0	0	1	1	0	6	f <sub>XCLK</sub> /6
		:	:	:	:	:	:	:	:	:	:
		1	1	1	1	1	1	0	0	252	f <sub>XCLK</sub> /252
		1	1	1	1	1	1	0	1	253	f <sub>XCLK</sub> /253
		1	1	1	1	1	1	1	0	254	f <sub>XCLK</sub> /254
		1	1	1	1	1	1	1	1	255	f <sub>XCLK</sub> /255
a) f <sub>XCLK</sub> : Clock frequency selected by UCnCTL1.UCnCKS[3:0]											

**(4) Baud rate**

The baud rate is obtained by the following equation.

$$\text{Baud rate} = \frac{f_{\text{XCLK}}}{2 \times k} [\text{bps}]$$

f<sub>XCLK</sub> = Clock frequency selected by UCnCTL1.UCnCKS[3:0].

k = Value set using the UCnCTL2.UCnBRS[7:0] bits  
 (k = 4, 5, 6, ..., 255)

**(5) Baud rate error**

The baud rate error is obtained by the following equation.

$$\text{Error (\%)} = \left( \frac{\text{Actual baud rate (baud rate with error)}}{\text{Target baud rate (correct baud rate)}} - 1 \right) \times 100 [\%]$$

- Caution**
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 “Allowable baud rate range during reception” on page 274.

- Example**
- Setting value of
    - UCnCTL1.UDnCKS[3:0] = 0001<sub>B</sub>:  $f_{\text{CLK}} = \text{PCLK1} = 16 \text{ MHz}$
    - UCnCTL2.UDnBRS[7:0] = 0011 0100<sub>B</sub>:  $k = 52$
  - Target baud rate = 153,600 bps
  - Actual Baud rate =  $16 \text{ MHz} / (2 \times 52) = 153,846 \text{ [bps]}$
  - Baud rate error =  $(153,846/153,600 - 1) \times 100 = 0.160 [\%]$

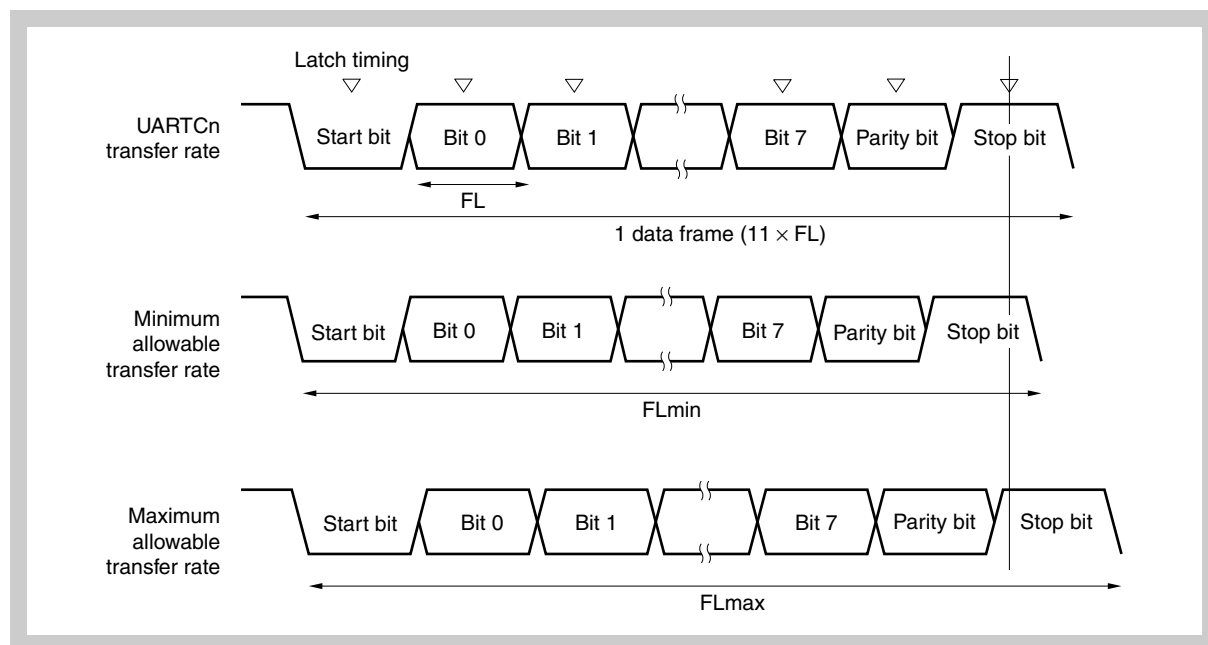
**(6) Baud rate setting example****Table 11-11 Baud rate generator setting data**

Target baud rate [bps]	UCnCTL1	$f_{\text{CLK}}$	UCnCTL2	k	Baud rate error [%]
600	08H	125 KHz	68H	104	0.16
1200	07H	250 KHz	68H	104	0.16
2400	06H	500 KHz	68H	104	0.16
4800	05H	1 MHz	68H	104	0.16
9600	04H	2 MHz	68H	104	0.16
10400	04H	2 MHz	60H	96	0.16
19200	03H	4 MHz	68H	104	0.16
31250	02H	8 MHz	80H	128	0.00
38400	02H	8 MHz	68H	104	0.16
56000	01H	16 MHz	8FH	143	-0.1
76800	02H	8 MHz	1AH	52	0.16
125000	02H	8 MHz	20H	32	0.00
153600	02H	16 MHz	1AH	52	0.16
250000	02H	8 MHz	10H	16	0.00
312500	00H	32 MHz	33H	51	0.39
1000000	00H	32 MHz	10H	16	0.00
2000000	00H	32 MHz	08H	8	0.00

**(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 11-12 Allowable baud rate range during reception**

As shown in *Figure 11-12*, the receive data latch timing is determined by the counter set using the UCnCTL2 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: UARTCn baud rate

k: Setting value of UCnCTL2.UCnBRS[7:0]

FL: 1-bit data length

Latch timing margin: 2 clocks

Minimum allowable transfer rate:

$$FL_{\min} = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} \times FL$$

Therefore, the maximum baud rate that can be received by the destination is as follows.

$$BR_{\max} = (FL_{\min}/11)^{-1} = \frac{22k}{21k+2} \times \text{Brate}$$

Similarly, obtaining the following maximum allowable transfer rate yields the following.

$$\frac{10}{11} \times FL_{\max} = 11 \times FL - \frac{k+2}{2k} \times FL = \frac{21k-2}{2k} \times FL$$

$$FL_{\max} = \frac{21k-2}{20k} \times FL \times 11$$

Therefore, the minimum baud rate that can be received by the destination is as follows.

$$BR_{\min} = (FL_{\max}/11)^{-1} = \frac{20k}{21k-2} \times \text{Brate}$$

Obtaining the allowable baud rate error for UARTCn and the destination from the above-described equations for obtaining the minimum and maximum baud rate values yields the following.

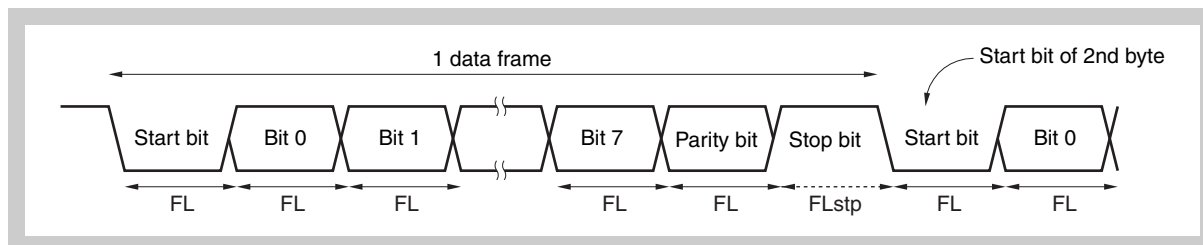
**Table 11-12 Maximum/Minimum allowable baud rate error**

Division ratio (k)	Maximum allowable baud rate error	Minimum allowable baud rate error
8	+3.53%	-3.61%
20	+4.26%	-4.31%
50	+4.56%	-4.58%
100	+4.66%	-4.67%
255	+4.72%	-4.73%

- Note**
1. The reception accuracy depends on the bit count in 1 frame, the input clock frequency, and the division ratio (k). The higher the input clock frequency and the larger the division ratio (k), the higher the accuracy.
  2. k: Setting value of UCnCTL2.UCnBRS[7:0]

**(8) Baud rate during continuous transmission**

During continuous transmission, the transfer rate from the stop bit to the next start bit is usually 2 base clocks longer. However, timing initialization is performed via start bit detection by the receiving side, so this has no influence on the transfer result.



**Figure 11-13** Transfer rate during continuous transfer

Assuming 1 bit data length: FL; stop bit length: FLstp; and base clock frequency:  $f_{XCLK}$ , we obtain the following equation.

$$FL_{stp} = FL + 2/f_{XCLK}$$

Therefore, the transfer rate during continuous transmission is as follows.

$$\text{Transfer rate} = 11 \times FL + (2/f_{XCLK})$$



## 11.7 Cautions

### (1) UARTCn operation stop

If both of the following actions in UARTCn happen at the same time the INTUCnR signal may be generated inadvertently and no data is stored in the UCnRX register:

- INTUCnR is generated due to completion of a serial receive operation,
- UCnPWR bit or UCnRXE bit of the UCnCTL0 register is cleared (set to 0).

**Workaround** To avoid the generation of the INTUCnR signal when UCnPWR bit or UCnRXE bit is cleared (set to 0) do the following:

1. Set (set to 1) the interrupt mask flag (UCnRMK) of the interrupt control register (UCnRIC),
2. Clear (set to 0) the UCnPWR bit or UCnRXE bit of the UCnCTL0 register,
3. Clear (set to 0) the interrupt request flag (UCnRIF) of the UCnRIC register.

### (2) UARTCn receive error interrupt

If both of the following actions in UARTCn happen at the same time the INTUCnRE may not be generated and the INTUCnR signal may be generated inadvertently:

- INTUCnRE is generated due to erroneous frame reception.
- UCnPWR bit or UCnRXE bit of control register UCnCTL0 is cleared to 0.

**Workaround**

1. Set the interrupt mask flag UCnRMK of the interrupt control register UCnRIC to 1 and set the interrupt mask flag UCnREMK of the interrupt control register UCnREIC.
2. Clear the UCnPWR bit or UCnRXE bit of the UCnCTL0 register to 0.
3. Clear the interrupt request flag UCnRIF of the UCnRIC register and clear the interrupt request flag UCnREIF of the UCnREIC register to 0.

## Chapter 12 Clocked Serial Interface (CSIB)

The microcontroller has two instances of the Clocked Serial Interface CSIB: CSIB0 to CSIB1.

**Note** Throughout this chapter, the individual instances of CSIB are identified by “n” (n = 0, 1).

### 12.1 Features

- Transfer rate: Maximum 8 Mbps
- Master mode and slave mode selectable
- 8-bit to 16-bit transfer, 3-wire serial interface
- 3 interrupt request signals (INTCBnT, INTCBnR, INTCBnRE)
- Serial clock and data phase switchable
- Transfer data length selectable in 1-bit units between 8 and 16 bits
- Transfer data MSB-first/LSB-first switchable
- 3-wire transfer
  - SOBn: Serial data output
  - SIBn: Serial data input
  - SCKBn: Serial clock input/output

Transmission mode, reception mode, and transmission/reception mode specifiable

- Slave select function supported
  - $\overline{\text{SSBn}}$ : Serial slave select input
- DMA support
- Dedicated baud rate generator for each interface instance

## 12.2 Configuration

The following shows the block diagram of CSIBn.

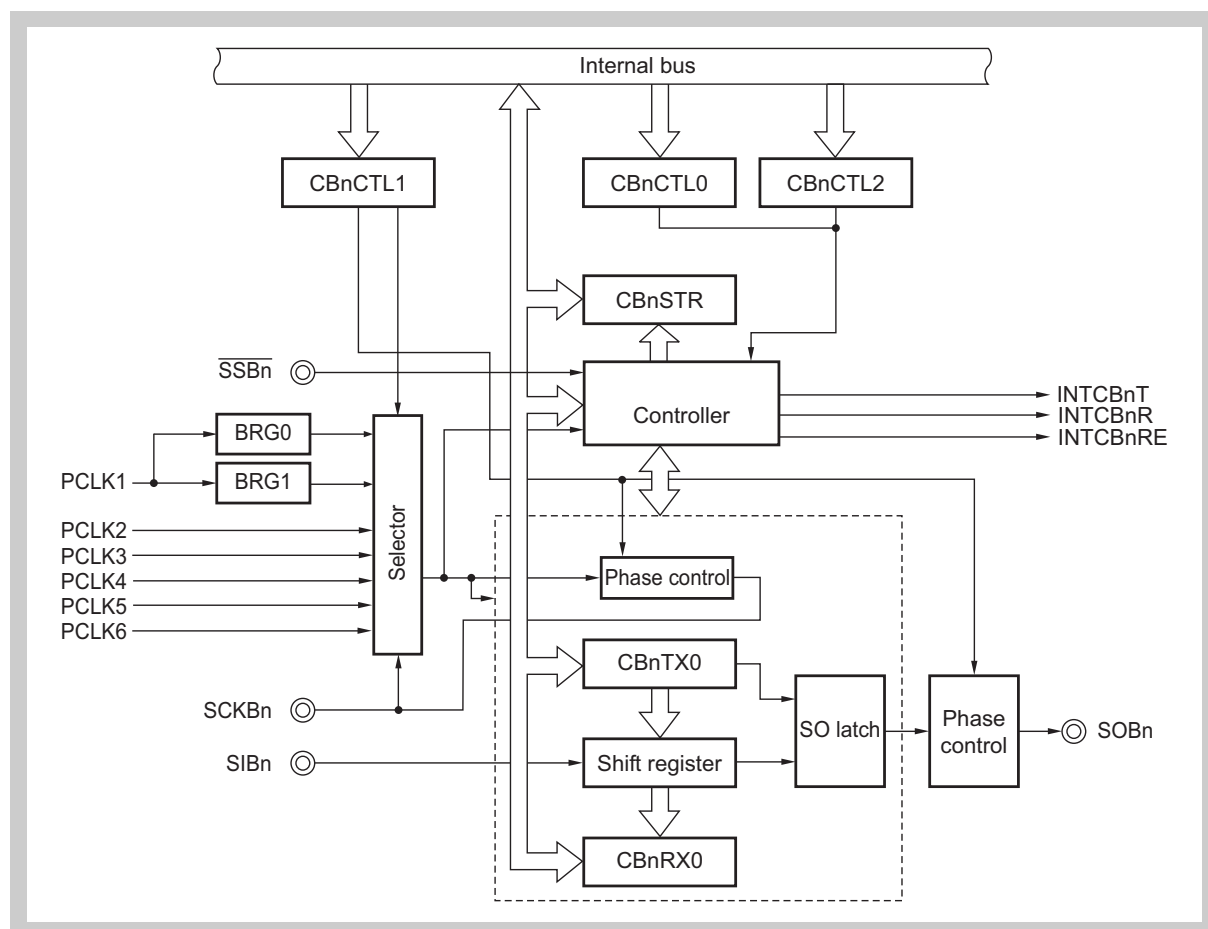


Figure 12-1 Block diagram of CSIBn

**Note** PCLK1 to PCLK6: peripheral clocks (refer to “Clock Generator” on page 148):

PCLK	1	2	3	4	5	6
f [MHz]	16	8	4	2	1	0.5

Additional clocks are generated by dedicated baud rate generators BRG0/BRG1.

CSIBn includes the following hardware.

Table 12-1 Configuration of CSIBn

Item	Configuration
Registers	CSIBn receive data register (CBnRX0) CSIBn transmit data register (CBnTX0)
Control registers	CSIBn control register 0 (CBnCTL0) CSIBn control register 1 (CBnCTL1) CSIBn control register 2 (CBnCTL2) CSIBn status register (CBnSTR)

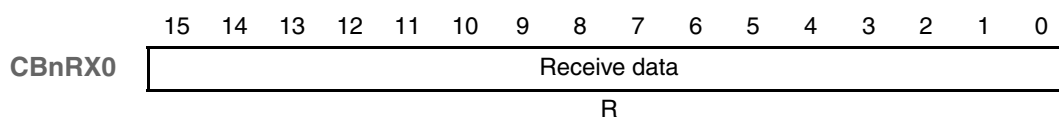
**(1) CBnRX0 - CSIBn receive data register**

The CBnRX0 register is a 16-bit buffer register that holds receive data. The receive operation is started by reading the CBnRX0 register in the reception enabled status.

**Access** This register can be read-only in 16-bit units.  
If the transfer data length is 8 bits, the lower 8 bits of this register are read-only in 8-bit units as the CBnRX0L register.

**Address** CB0RX0: FFFFFFFD04<sub>H</sub>  
CB1RX0: FFFFFFFD24<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. This register is cleared by any reset.  
In addition to reset input, the CBnRX0 register can be initialized by clearing (to 0) the CBnPWR bit of the CBnCTL0 register.

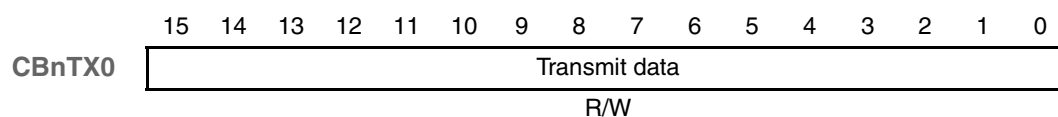
**(2) CBnTX0 - CSIBn transmit data register**

The CBnTX0 register is a 16-bit buffer register used to write the CSIBn transfer data. The transmit operation is started by writing data to the CBnTX register in the transmission enabled status.

**Access** This register can be read/written in 16-bit units.  
If the transfer data length is 8 bits, the lower 8 bits of this register are read/write in 8-bit units as the CBnTX0L register.

**Address** CB0TX0: FFFFFFFD06<sub>H</sub>  
CB1TX0: FFFFFFFD26<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. This register is cleared by any reset.  
In addition to reset input, the CBnTX0 register can be initialized by clearing (to 0) the CBnPWR bit of the CBnCTL0 register.



**Note** The communication start conditions are shown below:

- Transmission mode (CBnTXE bit = 1, CBnRXE bit = 0):  
Write to CBnTX0 register
- Transmission/reception mode (CBnTXE bit = 1, CBnRXE bit = 1):  
Write to CBnTX0 register
- Reception mode (CBnTXE bit = 0, CBnRXE bit = 1):  
Read from CBnRX0 register

## 12.3 CSIB 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) CBnCTL0 - CSIBn control register 0

CBnCTL0 is a register that controls the CSIBn serial transfer operation.

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

**Address** CB0CTL0: FFFFFFFD00<sub>H</sub>  
CB1CTL0: FFFFFFFD20<sub>H</sub>

**Initial Value** 01<sub>H</sub>

	7	6	5	4	3	2	1	0
<b>CBnCTL0</b>	CBnPWR	CBnTXE <sup>a</sup>	CBnRXE <sup>a</sup>	CBnDIR <sup>a</sup>	0	0	CBnTMS <sup>a</sup>	CBnSCE
	R/W	R/W	R/W	R/W	R	R	R/W	R/W

- a) 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.

**Table 12-2 CBnCTL0 register contents (1/2)**

Bit position	Bit name	Function
7	CBnPWR	CSIBn operation disable/enable: 0: Disable CSIBn operation and reset the CSIBn registers 1: Enable CSIBn operation The CBnPWR bit controls the CSIBn operation and resets the internal circuit. <b>Note:</b> To abort reception/transmission forcibly, clear the CBnPWR bit (not the CBnTXE bit or CBnRXE bit) to 0. The clock output stops at this time.
6	CBnTXE	Transmit operation disable/enable: 0: Disable transmit operation 1: Enable transmit operation The SOBn output is low level and transmission is disabled when the CBnTXE bit is 0.
5	CBnRXE	Receive operation disable/enable: 0: Disable receive operation 1: Enable receive operation When the CBnRXE bit is cleared to 0, no reception complete interrupt is output even when the prescribed data is transferred in order to disable the receive operation, and the receive data (CBnRX0 register) is not updated.
4	CBnDIR	Transfer direction mode specification (MSB/LSB): 0: MSB first transfer 1: LSB first transfer
1	CBnTMS	Transfer mode specification (MSB/LSB): 0: Single transfer mode 1: Continuous transfer mode

**Table 12-2 CBnCTL0 register contents (2/2)**

Bit position	Bit name	Function
0	CBnSCE	<p>Specification of start transfer disable/enable:            0: Communication start trigger invalid            1: Communication start trigger valid</p> <p>This bit controls the behaviour upon a communication start trigger in master/slave single/continuous reception mode.</p> <p>To start the reception operation set the bit to 1 before performing a dummy read to the CBnRX0 register.</p> <p>To stop the reception operation in</p> <ul style="list-style-type: none"> <li>• Single reception mode clear the CBnSCE bit before reading the final data from the CBnRX0 register.</li> <li>• Continuous reception mode clear the CBnSCE bit at least one communication clock before the completion of the last data reception. Otherwise the next communication operation is automatically started</li> <li>• For details refer to chapter 12.4“Operation” on page 288.</li> </ul>

**(2) CBnCTL1 - CSIBn control register 1**

CBnCTL1 is an 8-bit register that controls the CSIBn serial transfer operation.

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

**Address** CB0CTL1: FFFFFFFD01<sub>H</sub>  
CB1CTL1: FFFFFFFD21<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>CBnCTL1</b>	0	0	0	CBnCKP	CBnDAP	CBnCKS2	CBnCKS1	CBnCKS0
	R	R	R	R/W	R/W	R/W	R/W	R/W

**Caution** The CBnCTL1 register can be rewritten only when the CBnCTL0.CBnPWR bit = 0.

**Table 12-3 CBnCTL1 register contents (1/2)**

Bit position	Bit name	Function															
4 3	CBnCKP CBnDAP	<p>Specification of data transmission/reception timing in relation to SCKBn.</p> <table> <tr> <th>CBnCKP</th><th>CBnDAP</th><th>SIBn/SOBn timing in relation to SCKBn</th></tr> <tr> <td>0</td><td>0</td><td> <p>Communication type 1</p> </td></tr> <tr> <td>0</td><td>1</td><td> <p>Communication type 2</p> </td></tr> <tr> <td>1</td><td>0</td><td> <p>Communication type 3</p> </td></tr> <tr> <td>1</td><td>1</td><td> <p>Communication type 4</p> </td></tr> </table>	CBnCKP	CBnDAP	SIBn/SOBn timing in relation to SCKBn	0	0	<p>Communication type 1</p>	0	1	<p>Communication type 2</p>	1	0	<p>Communication type 3</p>	1	1	<p>Communication type 4</p>
CBnCKP	CBnDAP	SIBn/SOBn timing in relation to SCKBn															
0	0	<p>Communication type 1</p>															
0	1	<p>Communication type 2</p>															
1	0	<p>Communication type 3</p>															
1	1	<p>Communication type 4</p>															

**Table 12-3 CBnCTL1 register contents (2/2)**

Bit position	Bit name	Function				
2 to 0	CBnCKS[2:0]	Communication clock setting.				
		CBnCKS2	CBnCKS1	CBnCKS0	Communication clock	Mode
		0	0	0	BRG0	Master mode
		0	0	1	BRG1	Master mode
		0	1	0	PCLK2	Master mode
		0	1	1	PCLK3	Master mode
		1	0	0	PCLK4	Master mode
		1	0	1	PCLK5	Master mode
		1	1	0	PCLK6	Master mode
		1	1	1	External clock (SCKBn)	Slave mode

**Note** PCLK2 to PCLK6: peripheral clocks (refer to “Clock Generator” on page 148):

PCLK	2	3	4	5	6
f [MHz]	8	4	2	1	0.5

Additional clocks are generated by dedicated baud rate generators BRG0/BRG1.



**(3) CBnCTL2 - CSIBn control register 2**

CBnCTL2 is an 8-bit register that controls the number of CSIBn serial transfer bits.

**Access** This register can be read/written in 8-bit units.

**Address** CB0CTL2: FFFFFFFD02<sub>H</sub>  
CB1CTL2: FFFFFFFD22<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>CBnCTL2</b>	0	0	0	0	CBnCL3	CBnCL2	CBnCL1	CBnCL0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Caution** The CBnCTL2 register can be rewritten only when the CBnCTL0.CBnPWR bit = 0 or when both the CBnTXE and CBnRXE bits = 0.

**Table 12-4 CBnCTL2 register contents**

Bit position	Bit name	Function				
3 to 0	CBnCL[3:0]	Number of serial transfer bits (character length)				
		CBnCL3	CBnCL2	CBnCL1	CBnCL0	Number of serial transfer bits
		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	x	x	x	16 bits

**Note** If the number of transfer bits is other than 8 or 16, prepare and use data stuffed from the LSB of the CBnTX0 and CBnRX0 registers.

**Transfer data length change function** The CSIBn transfer data length can be set in 1-bit units between 8 and 16 bits using the CBnCTL2.CBnCL[3:0] bits.

When the transfer bit length is set to a value other than 16 bits, set the data to the CBnTX0 or CBnRX0 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.

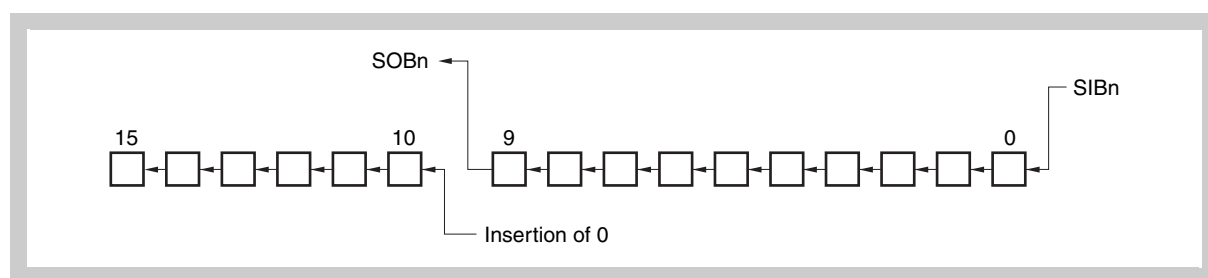


Figure 12-2 (i) Transfer bit length = 10 bits, MSB first

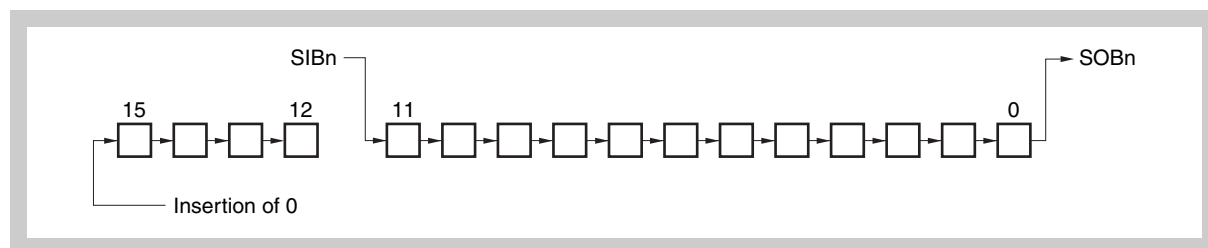


Figure 12-3 (ii) Transfer bit length = 12 bits, LSB first

**(4) CBnSTR - CSIBn status register**

CBnSTR is an 8-bit register that displays the CSIBn status.

**Access** This register can be read/written in 8-bit or 1-bit units.  
Bit CBnTSF is read-only.

**Address** CB0CTL2: FFFFFFFD03<sub>H</sub>  
CB1CTL2: FFFFFFFD23<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.  
In addition to reset input, the CBnSTR register can be initialized by clearing the CBnCTL0.CBnPWR bit to 0.

	7	6	5	4	3	2	1	0
<b>CBnSTR</b>	CBnTSF	0	0	0	0	0	0	CBnOVE
	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 12-5 CBnSTR register contents**

Bit position	Bit name	Function
7	CBnTSF	Communication status flag 0: Communication stopped 1: Communicating During transmission, this register is set when data is prepared in the CBnTX register, and during reception, it is set when a dummy read of the CBnRX register is performed. When transfer ends, this flag is cleared to 0 at the last edge of the clock.
0	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 receive buffer, upon completion 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>

**Note** In case of an overrun error, the reception error interrupt INTCBnRE behaves different, depending on the transfer mode:

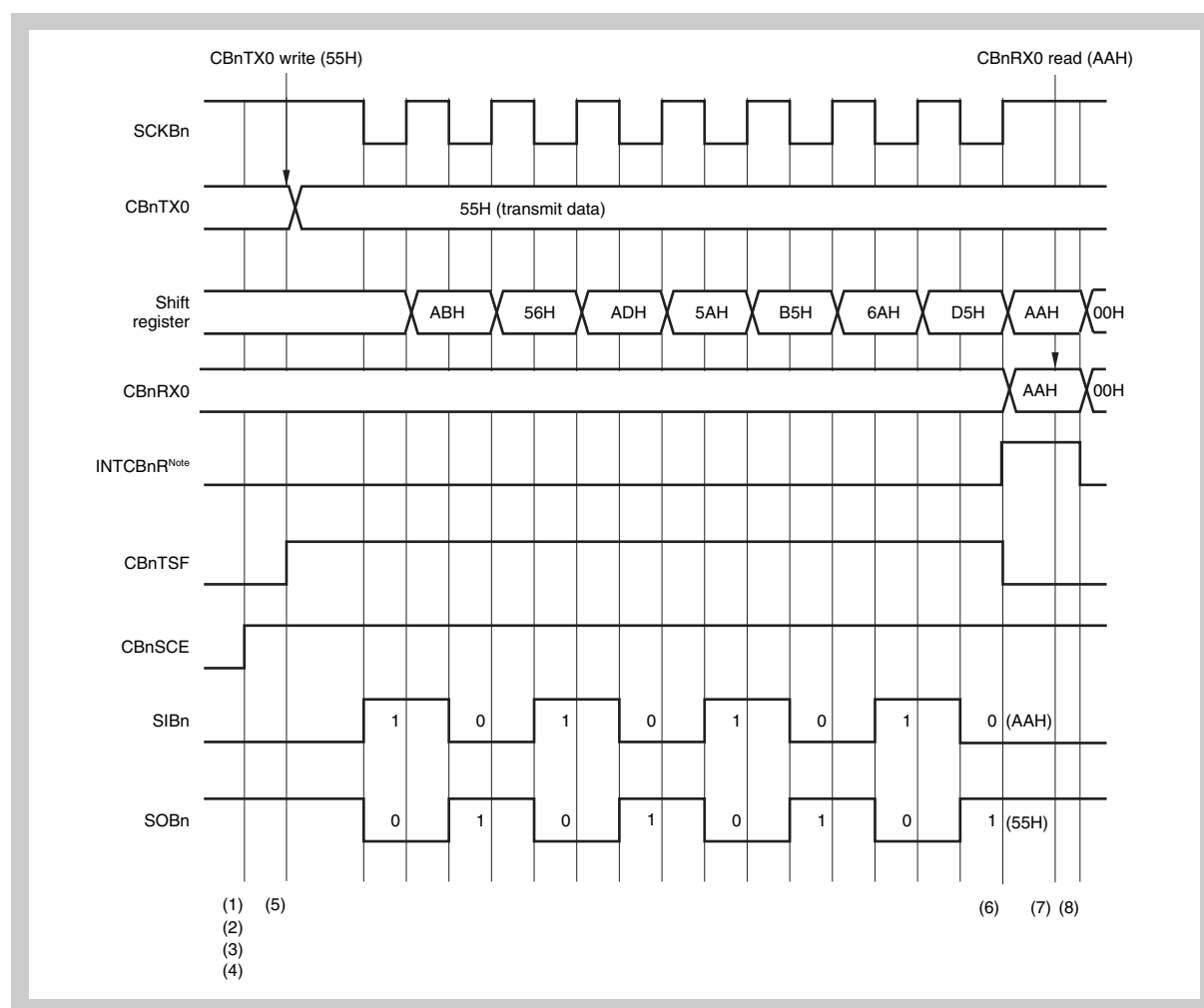
- Continuous transfer mode  
The reception error interrupt INTCBnRE is generated instead of the reception completion interrupt INTCBnR.
- Single transfer mode  
No interrupt is generated.

In either case the overflow flag CBnSTR.CBnOVE is set to 1 and the previous data in CBnRX0 will be overwritten with the new data.

## 12.4 Operation

### 12.4.1 Single transfer mode (master mode, transmission/reception mode)

MSB first (CBnCTL0.CBnDIR = 0),  
communication type 1 (CBnCTL1.CBnCKP = 0, CBnCTL1.CBnDAP = 0),  
transfer data length = 8 bits (CBnCTL2.CBnCL[3:0] = 0000B)



1. Clear the CBnCTL0.CBnPWR bit to 0.
2. Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
3. Set the CBnTXE, CBnRXE, and CBnSCE bits of the CBnCTL0 register to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the transmission/reception enabled status.
4. Set the CBnPWR bit to 1 to enable the CSIBn operation.
5. Write transfer data to the CBnTX0 register (transmission/reception start).
6. The reception complete interrupt request signal (INTCBnR) is output.
7. Read the CBnRX0 register before clearing the CBnPWR bit to 0.
8. Check that the CBnSTR.CBnTSF bit = 0 and set the CBnPWR bit to 0 to stop operation of CSIBn (end of transmission/reception).

To continue transfer, repeat steps (5) to (7) before (8).

- Note**
1. In transmission mode or transmission/reception mode, communication is not started by reading the CBnRX0 register.
  2. In single transmission or single transmission/reception mode, the INTCBnT signal is not generated. When communication is complete, the INTCBnR signal is generated.
  3. The processing of steps (3) and (4) can be set simultaneously.

---

**Caution** In case the CSIB interface is operating in

- single transmit/reception mode (CBnCTL0.CBnTMS = 0)
- communication type 2 respectively type 4 (CBnCTL1.CBnDAP = 1)

pay attention to following effect:

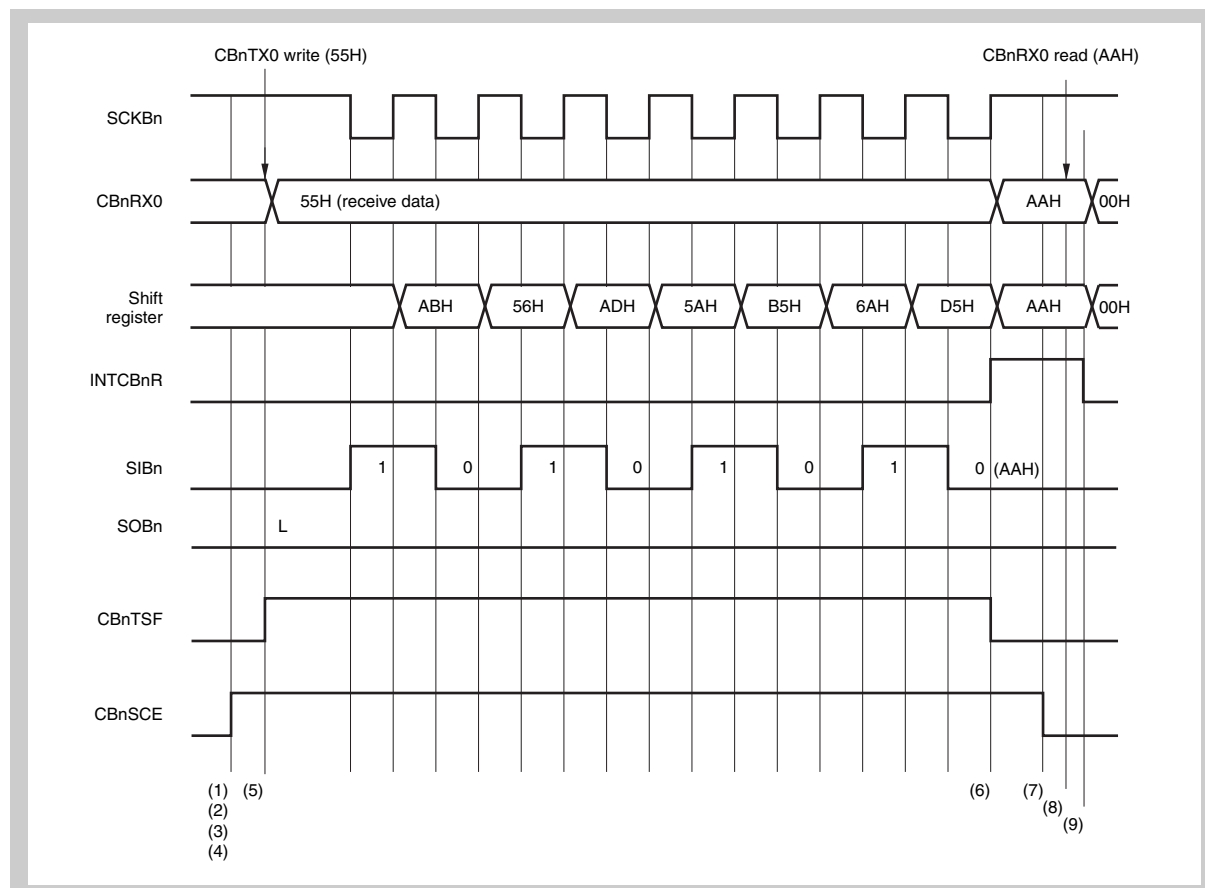
In case the next transmit should be initiated immediately after the occurrence of the reception completion interrupt INTCBnR any write to the CBnTX0 register is ignored as long as the communication status flag is still reflecting an ongoing communication (CBnTSF = 1). Thus the new transmission will not be started.

For transmitting data continuously use one of the following options:

- Use continuous transfer mode (CBnCTL0.CBnTMS = 1). This is the only usable mode for automatic transmission of data by the DMA Controller.
  - If single transfer mode (CBnCTL0.CBnTMS = 0) should be used, CBnSTR.CBnTSF = 0 needs to be verified before writing data to the CBnTX0 register.
-

### 12.4.2 Single transfer mode (master mode, reception mode)

MSB first (CBnCTL0.CBnDIR = 0),  
communication type 1 (CBnCTL1.CBnCKP = 0, CBnCTL1.CBnDAP = 0),  
transfer data length = 8 bits (CBnCTL2.CBnCL[3:0] = 0000B)



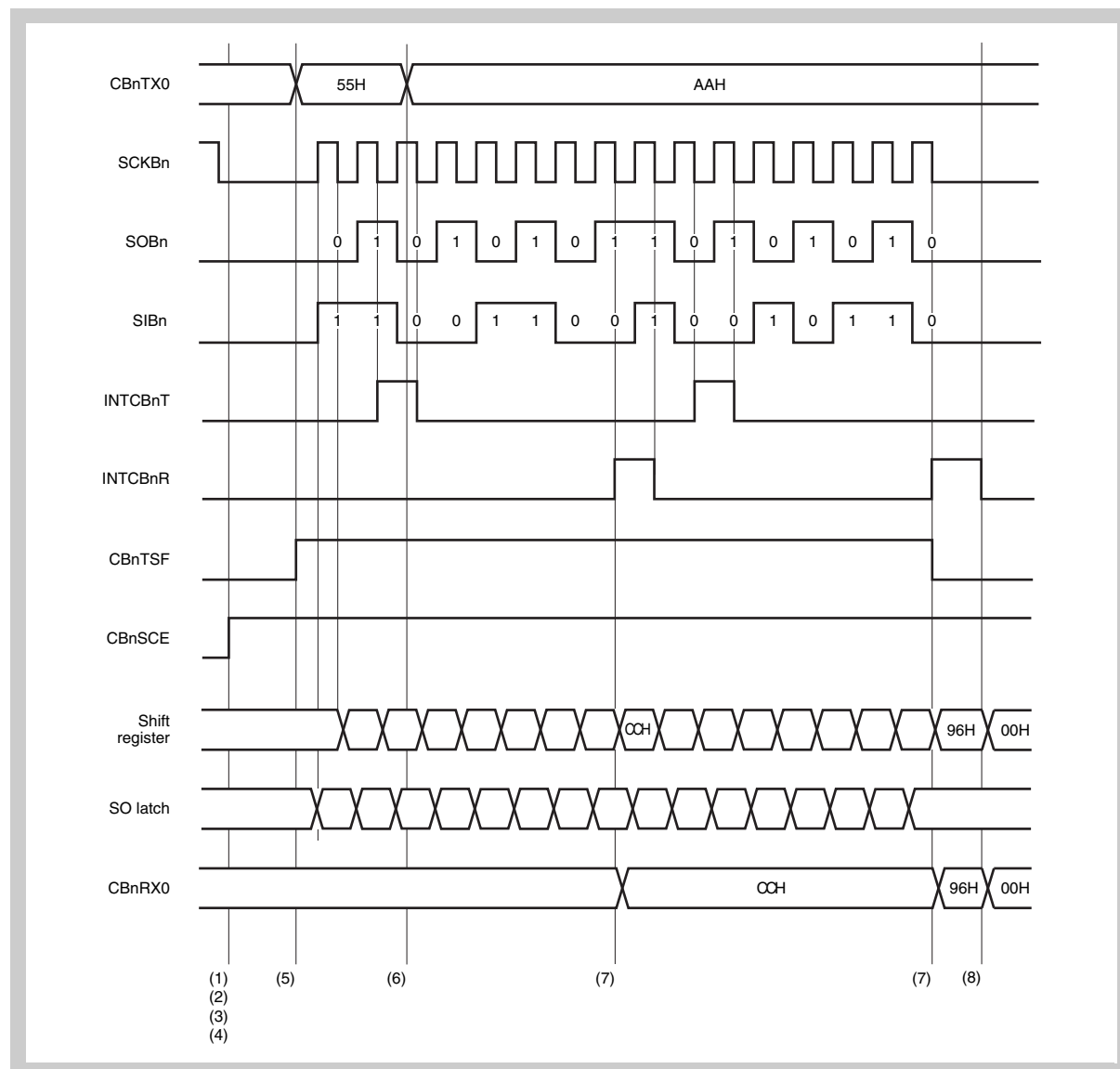
1. Clear the CBnCTL0.CBnPWR bit to 0.
2. Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
3. Set the CBnCTL0.CBnRXE and CBnCTL0.CBnSCE bits to 1, CBnCTL0.TXE to 0, at the same time as specifying the transfer mode using the CBnDIR bit, to set the reception enabled status.
4. Set the CBnPWR bit to 1 to enable the CSIBn operation.
5. Perform a dummy read of the CBnRX0 register (reception start trigger).
6. The reception complete interrupt request signal (INTCBnR) is output.
7. Set the CBnSCE bit to 0 to set the final receive data status.
8. Read the CBnRX0 register.
9. Check that the CBnSTR.CBnTSF bit = 0 and set the CBnPWR bit to 0 to stop the CSIBn operation (end of reception).

To continue transfer, repeat steps (5) and (6) before (7). (At this time, (5) is not a dummy read, but a receive data read combined with the reception trigger.)

**Note** The processing of steps (3) and (4) can be set simultaneously.

### 12.4.3 Continuous mode (master mode, transmission/reception mode)

MSB first (CBnCTL0.CBnDIR = 0),  
communication type 3 (CBnCTL1.CBnCKP = 1, CBnCTL1.CBnDAP = 0),  
transfer data length = 8 bits (CBnCTL2.CBnCL[3:0] = 0000B)



1. Clear the CBnCTL0.CBnPWR bit to 0.
2. Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
3. Set the CBnTXE, CBnRXE, and CBnSCE bits of the CBnCTL0 register to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the transmission/reception enabled status.
4. Set the CBnPWR bit to 1 to enable the CSIBn operation.
5. Write transfer data to the CBnTX0 register (transmission/reception start).
6. The transmission enable interrupt request signal (INTCBnT) is received and transfer data is written to the CBnTX0 register.
7. The reception complete interrupt request signal (INTCBnR) is output. Read the CBnRX0 register before the next receive data arrives or before the CBnPWR bit is cleared to 0.
8. Check that the CBnSTR.CBnTSF bit = 0 and set the CBnPWR bit to 0 to stop the operation of CSIBn (end of transmission/reception).

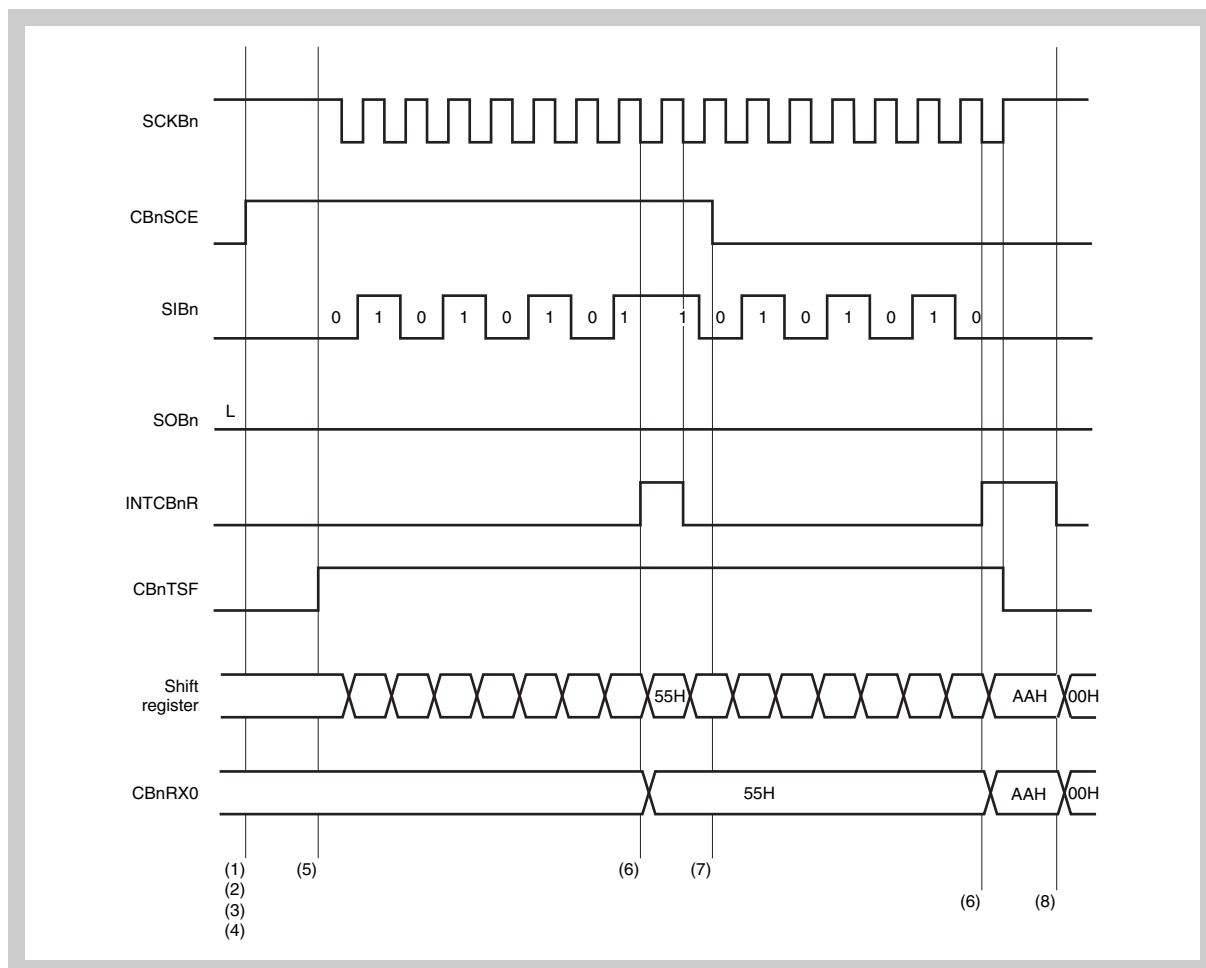
To continue transfer, repeat steps (5) to (7) before (8).

In transmission mode or transmission/reception mode, the communication is not started by reading the CBnRX0 register.



### 12.4.4 Continuous mode (master mode, reception mode)

MSB first (CBnCTL0.CBnDIR = 0),  
communication type 2 (CBnCTL1.CBnCKP = 0, CBnCTL1.CBnDAP = 1),  
transfer data length = 8 bits (CBnCTL2.CBnCL[3:0] = 0000B)

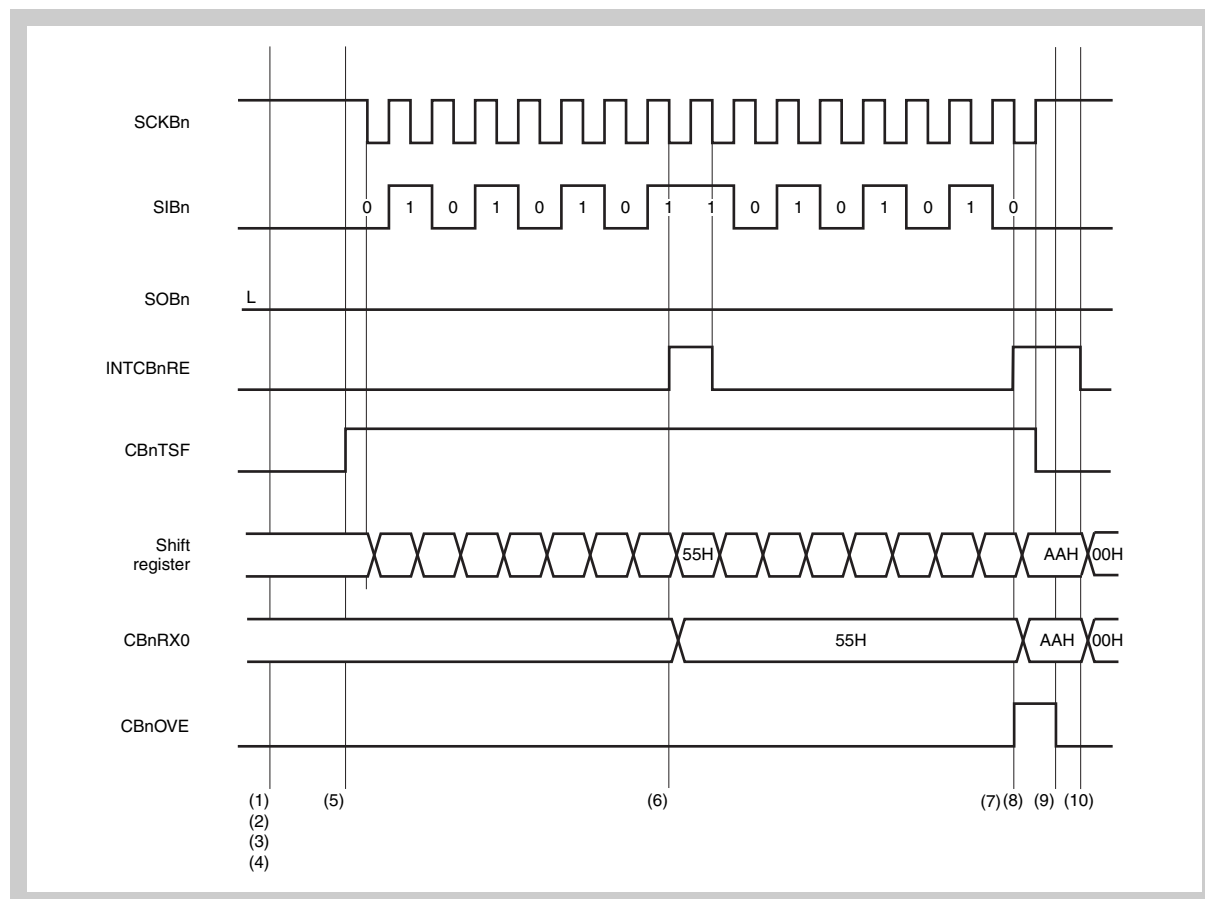


1. Clear the CBnCTL0.CBnPWR bit to 0.
2. Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
3. Set the CBnCTL0.CBnRXE and CBnCTL0.CBnSCE bits to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the reception enabled status.
4. Set the CBnPWR bit to 1 to enable the CSIBn operation.
5. Perform a dummy read of the CBnRX0 register (reception start trigger).
6. The reception complete interrupt request signal (INTCBnR) is output. Read the CBnRX0 register before the next receive data arrives or before the CBnPWR bit is cleared to 0.
7. Set the CBnCTL0.CBnSCE bit = 0 while the last data being received to set the final receive data status.
8. Check that the CBnSTR.CBnTSF bit = 0 and set the CBnPWR bit to 0 to stop the operation of CSIBn (end of reception).

To continue transfer, repeat steps (5) and (6) before (7).

### 12.4.5 Continuous reception mode (error)

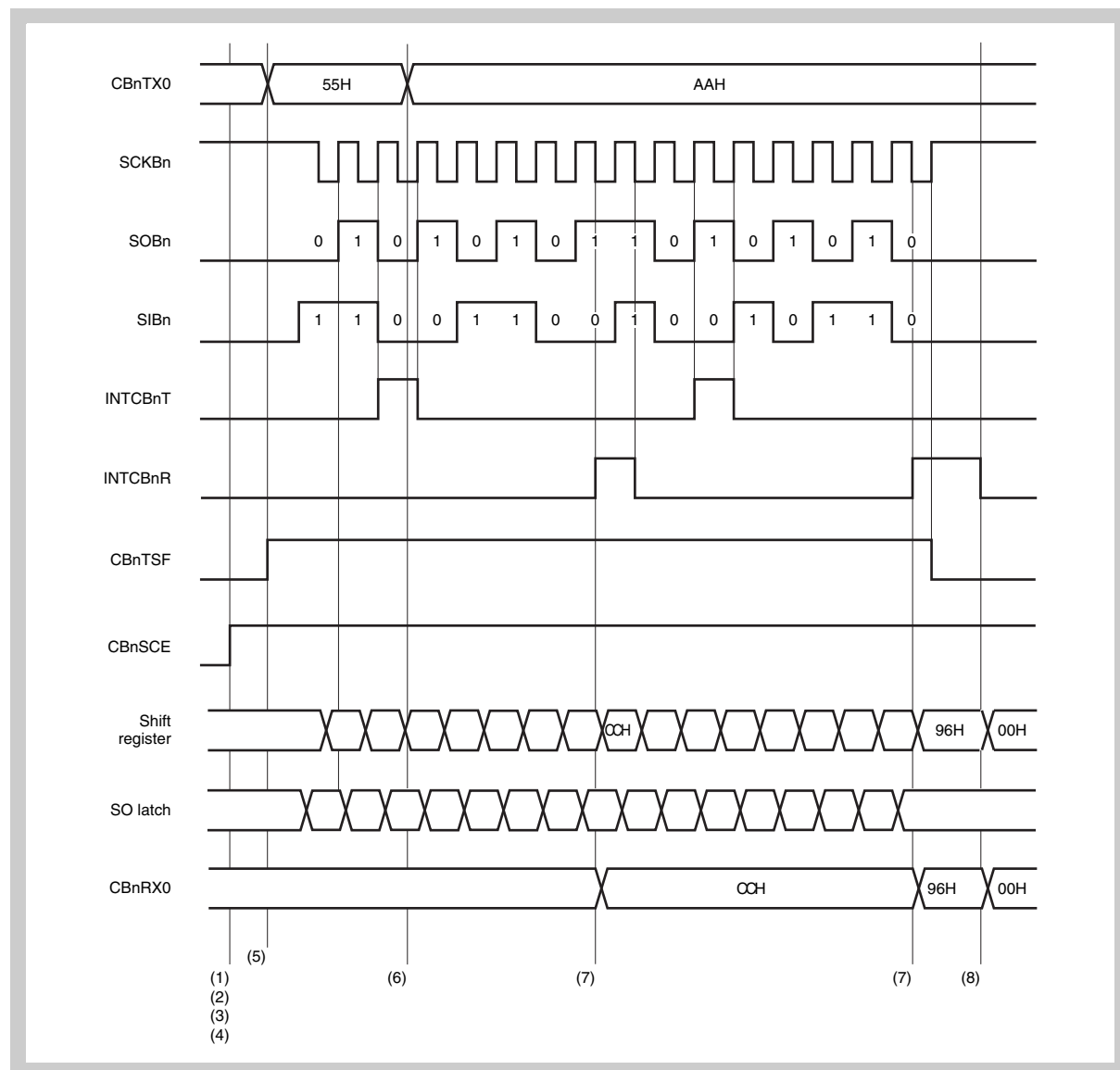
MSB first (CBnCTL0.CBnDIR = 0),  
communication type 2 (CBnCTL1.CBnCKP = 0, CBnCTL1.CBnDAP = 1),  
transfer data length = 8 bits (CBnCTL2.CBnCL[3:0] = 0000B)



1. Clear the CBnCTL0.CBnPWR bit to 0.
2. Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
3. Set the CBnCTL0.CBnRXE and CBnCTL0.CBnSCE bits to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the reception enabled status.
4. Set the CBnPWR bit = 1 to enable CSIBn operation.
5. Perform a dummy read of the CBnRX0 register (reception start trigger).
6. The reception complete interrupt request signal (INTCBnR) is output.
7. If the data could not be read before the end of the next transfer, the CBnSTR.CBnOVE flag is set to 1 upon the end of reception and the INTCBnR signal is output.
8. Overrun error processing is performed after checking that the CBnOVE bit = 1 in the INTCBnRE interrupt servicing.
9. Clear CBnOVE bit to 0.
10. Check that the CBnSTR.CBnTSF bit = 0 and set the CBnPWR bit to 0 to stop the operation CSIBn (end of reception).

### 12.4.6 Continuous mode (slave mode, transmission/reception mode)

MSB first (CBnCTL0.CBnDIR = 0),  
communication type 2 (CBnCTL1.CBnCKP = 0, CBnCTL1.CBnDAP = 1),  
transfer data length = 8 bits (CBnCTL2.CBnCL[3:0] = 0000B)

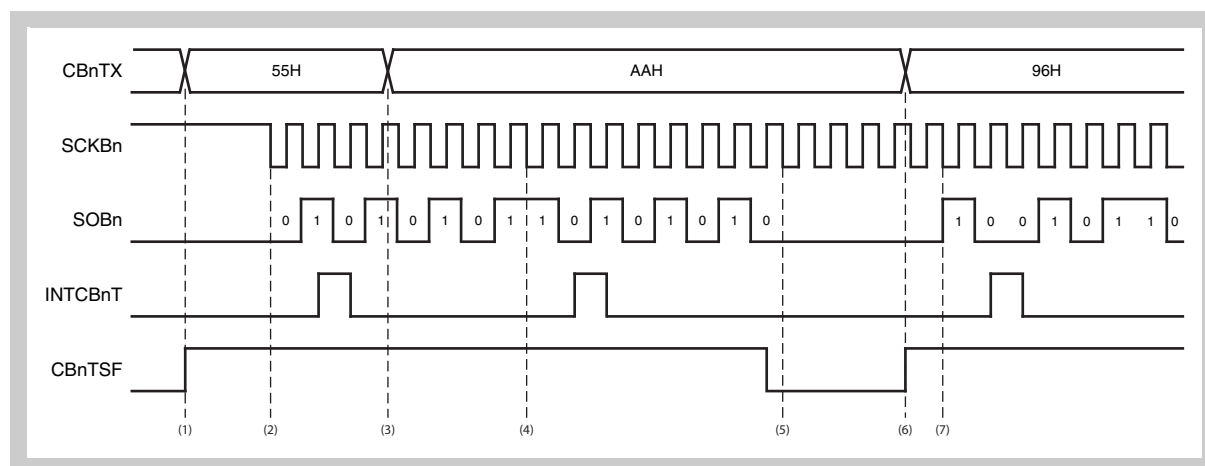


1. Clear the CBnCTL0.CBnPWR bit to 0.
2. Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
3. Set the CBnTXE, CBnRXE and CBnSCE bits of the CBnCTL0 register to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the transmission/reception enabled status.
4. Set the CBnPWR bit to 1 to enable supply of the CSIBn operation.
5. Write the transfer data to the CBnTX0 register.
6. The transmission enable interrupt request signal (INTCBnT) is received and the transfer data is written to the CBnTX0 register.
7. The reception complete interrupt request signal (INTCBnR) is output. Read the CBnRX0 register.
8. Check that the CBnSTR.CBnTSF bit = 0 and set the CBnPWR bit to 0 to stop the operation of CSIBn (end of transmission/reception).

To continue transfer, repeat steps (5) to (7) before (8).

**Discontinued transmission** In case the CSIB is operating in continuous slave transmission mode ( $\text{CBnCTL0.CBnTMS} = 1$ ,  $\text{CBnCTL1.CBnCKS}[2:0] = 111_{\text{B}}$ ) and new data is not written to the  $\text{CBnTX0}$  register the  $\text{SOBn}$  pin outputs the level of the last bit.

Figure 12-4 outlines this behaviour.



**Figure 12-4 Discontinued slave transmission**

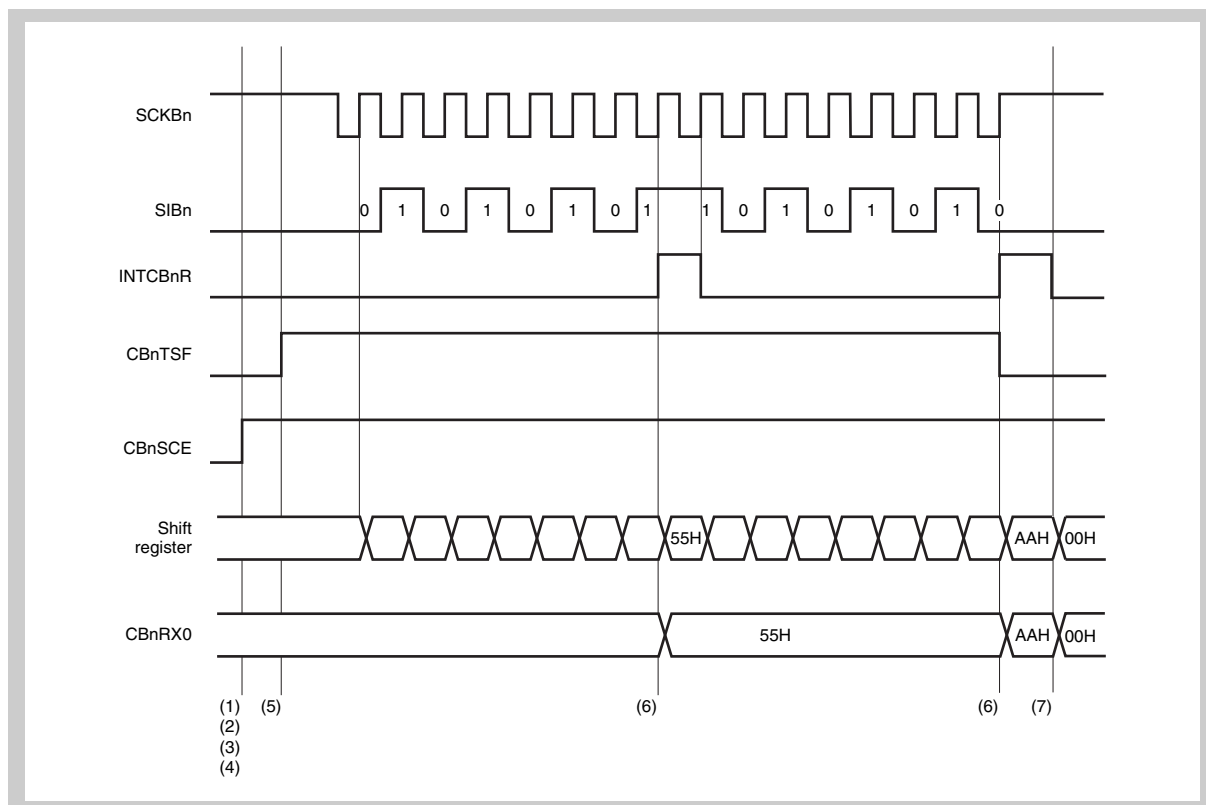
The example shows the situation that two data bytes (55<sub>H</sub>, AA<sub>H</sub>) are transmitted correctly, but the third (96<sub>H</sub>) fails.

- (1) Data 55<sub>H</sub> is written (by the CPU or DMA) to  $\text{CBnTX0}$ .
- (2) The master issues the clock  $\text{SCKBn}$  and transmission of 55<sub>H</sub> starts.
- (3)  $\text{INTCBnT}$  is generated and the next data AA<sub>H</sub> is written to  $\text{CBnTX0}$  promptly, i.e. before the first data has been transmitted completely.
- (4) Transmission of the second data AA<sub>H</sub> continues correctly and  $\text{INTCBnT}$  is generated. But this time the next data is not written to  $\text{CBnTX0}$  in time.
- (5) Since there is no new data available in  $\text{CBnTX0}$ , but the master continues to apply  $\text{SCKBn}$  clocks,  $\text{SOBn}$  remains at the level of the transmitted last bit.
- (6) New data (96<sub>H</sub>) is written to  $\text{CBnTX0}$ .
- (7) With the next  $\text{SCKBn}$  cycle transmission of the new data (96<sub>H</sub>) starts.

As a consequence the master receives a corrupted data byte from (5) onwards, which is made up of a random number of the repeated last bit of the former data and some first bits of the new data.

### 12.4.7 Continuous mode (slave mode, reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0),  
communication type 1 (CBnCTL1.CBnCKP = 0, CBnCTL1.CBnDAP = 0),  
transfer data length = 8 bits (CBnCTL2.CBnCL[3:0] = 0000B)



1. Clear the CBnCTL0.CBnPWR bit to 0.
2. Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
3. Set the CBnCTL0.CBnRXE and CBnCTL0.CBnSCE bits to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the reception enabled status.
4. Set the CBnPWR bit = 1 to enable CSIBn operation.
5. Perform a dummy read of the CBnRX0 register (reception start trigger).
6. The reception complete interrupt request signal (INTCBnR) is output. Read the CBnRX0 register.
7. Check that the CBnSTR.CBnTSF bit = 0 and set the CBnPWR bit to 0 to stop the operation of CSIBn (end of reception).

To continue transfer, repeat steps (5) and (6) before (7).

### 12.4.8 Clock timing

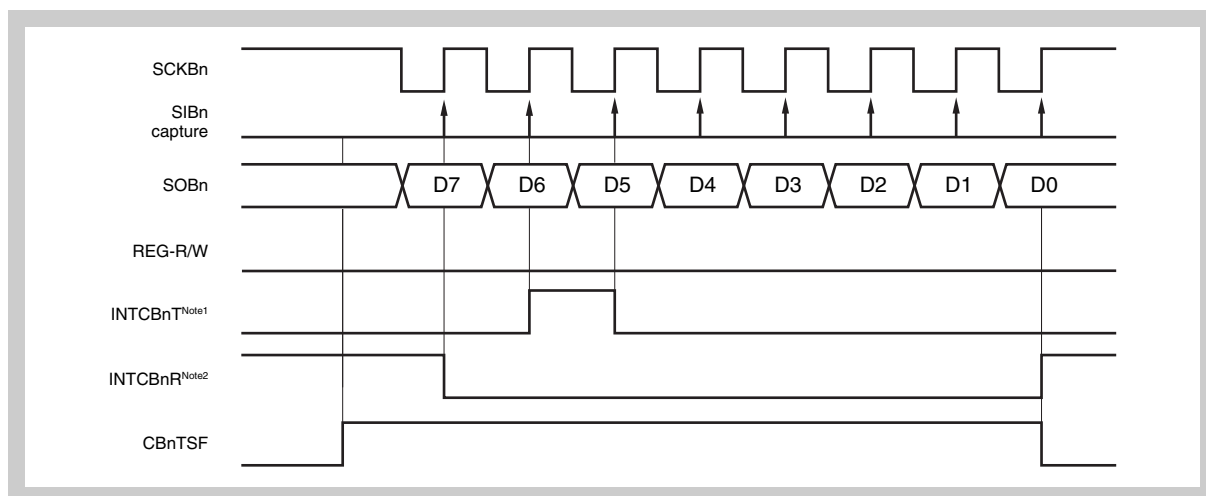


Figure 12-5 (i) Communication type 1 (CBnCKP = 0, CBnDAP = 0)

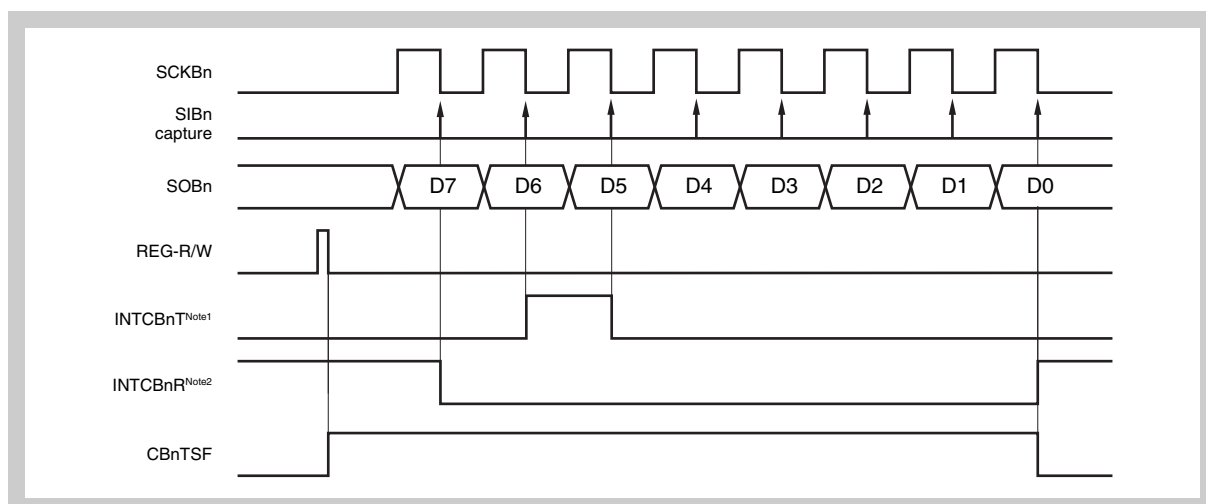


Figure 12-6 (ii) Communication type 3 (CBnCKP = 1, CBnDAP = 0)

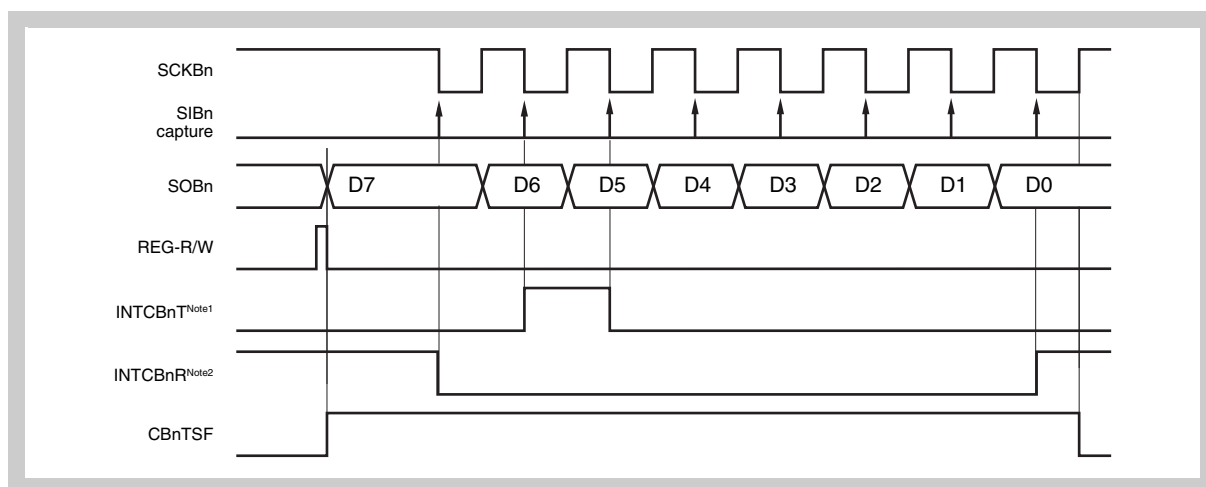
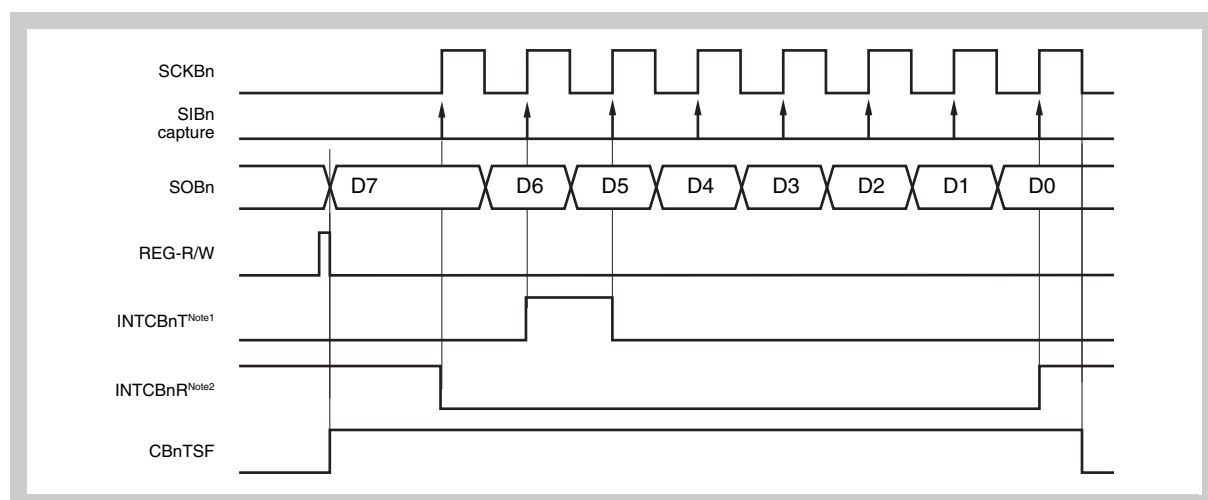


Figure 12-7 (iii) Communication type 2 (CBnCKP = 0, CBnDAP = 1)



**Figure 12-8 (iv) Communication type 4 (CBnCKP = 1, CBnDAP = 1)**

- Note**
1. The INTCBnT interrupt is set when the data written to the transmit buffer is transferred to the data shift register in the continuous transmission or continuous transmission/reception modes. In the single transmission or single transmission/reception modes, the INTCBnT interrupt request signal is not generated, but the INTCBnR interrupt request signal is generated upon completion of communication.
  2. The INTCBnR interrupt occurs if reception is correctly completed and receive data is ready in the CBnRX0 register while reception is enabled, and if an overrun error occurs. In the single mode, the INTCBnR interrupt request signal is generated even in the transmission mode, upon completion of communication.

## 12.5 Output Pins

### (1) SCKBn pin

When CSIBn operation is disabled (CBnCTL0.CBnPWR bit = 0), the SCKBn pin output status is as follows.

CBnCKP	CBnCKS2	CBnCKS1	CBnCKS0	SCKBn pin output
0	Don't care	Don't care	Don't care	Fixed to high level
1	1	1	1	High impedance
	Other than above			Fixed to low level

**Note** The output level of the SCKBn pin changes if any of the CBnCTL1.CBnCKP and CBnCKS2 to CBnCKS0 bits is rewritten.

### (2) SOBn pin

When CSIBn operation is disabled (CBnPWR bit = 0), the SOBn pin output status is as follows.

CBnTXE	CBnDAP	CBnDIR	SOBn pin output
0	×	×	Fixed to low level
1	0	×	SOBn latch value (low level)
	1	0	CBnTX0 value (MSB)
		1	CBnTX0 value (LSB)

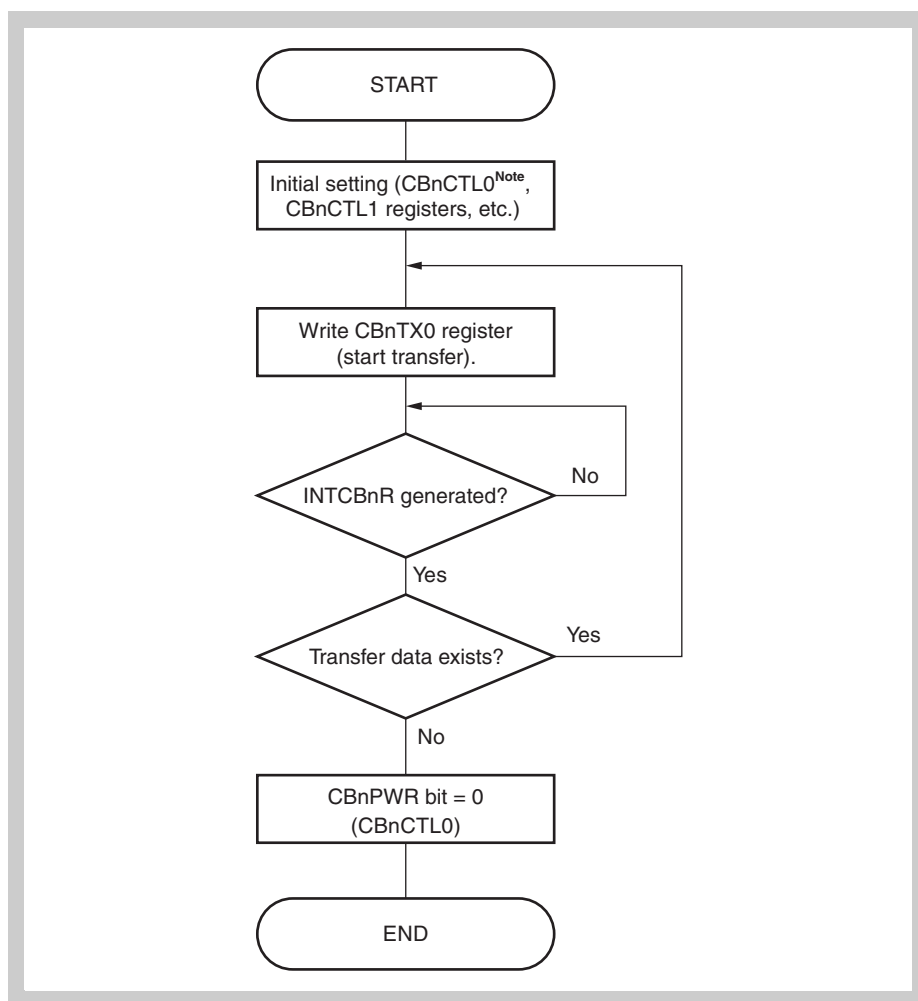
**Note** 1. The SOBn pin output changes when any one of the CBnCTL0.CBnTXE, CBnCTL0.CBnDIR bits, and CBnCTL1.CBnDAP bit is rewritten.

2. ×: don't care



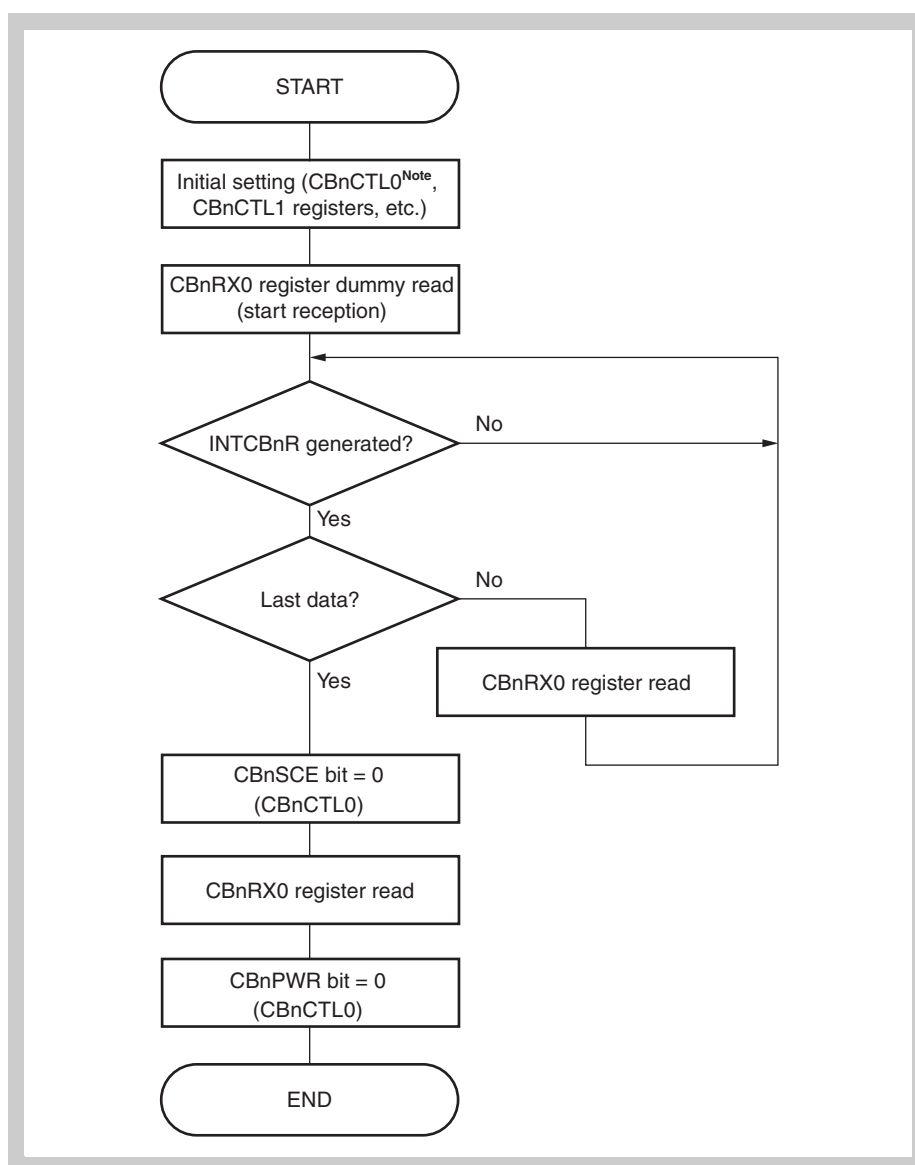
## 12.6 Operation Flow

### (1) Single transmission



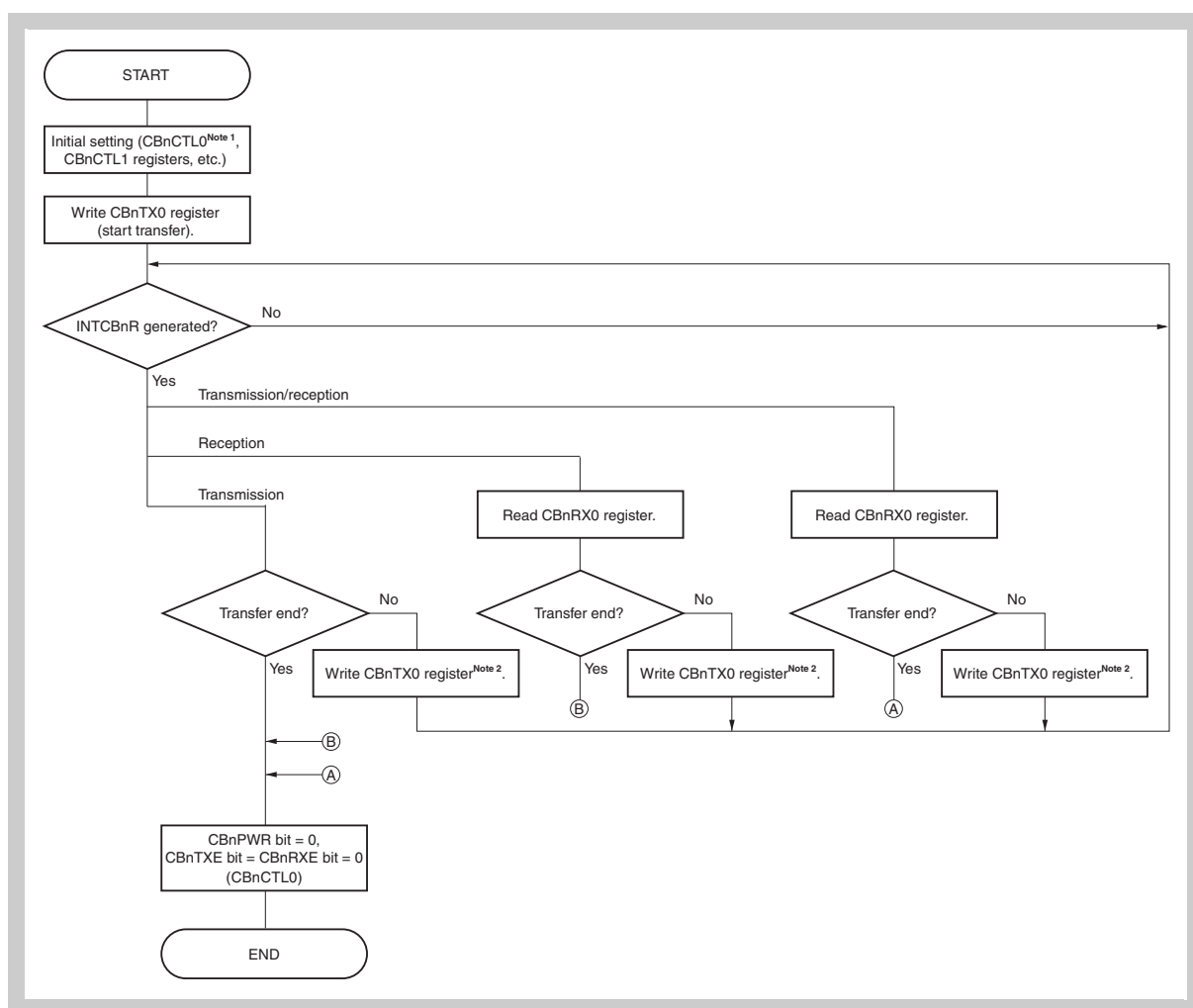
**Note** Set the CBnSCE bit to 1 in the initial setting.

**Caution** In the slave mode, data cannot be correctly transmitted if the next transfer clock is input earlier than the CBnTX0 register is written.

**(2) Single reception**

**Note** Set the CBnSCE bit to 1 in the initial setting.

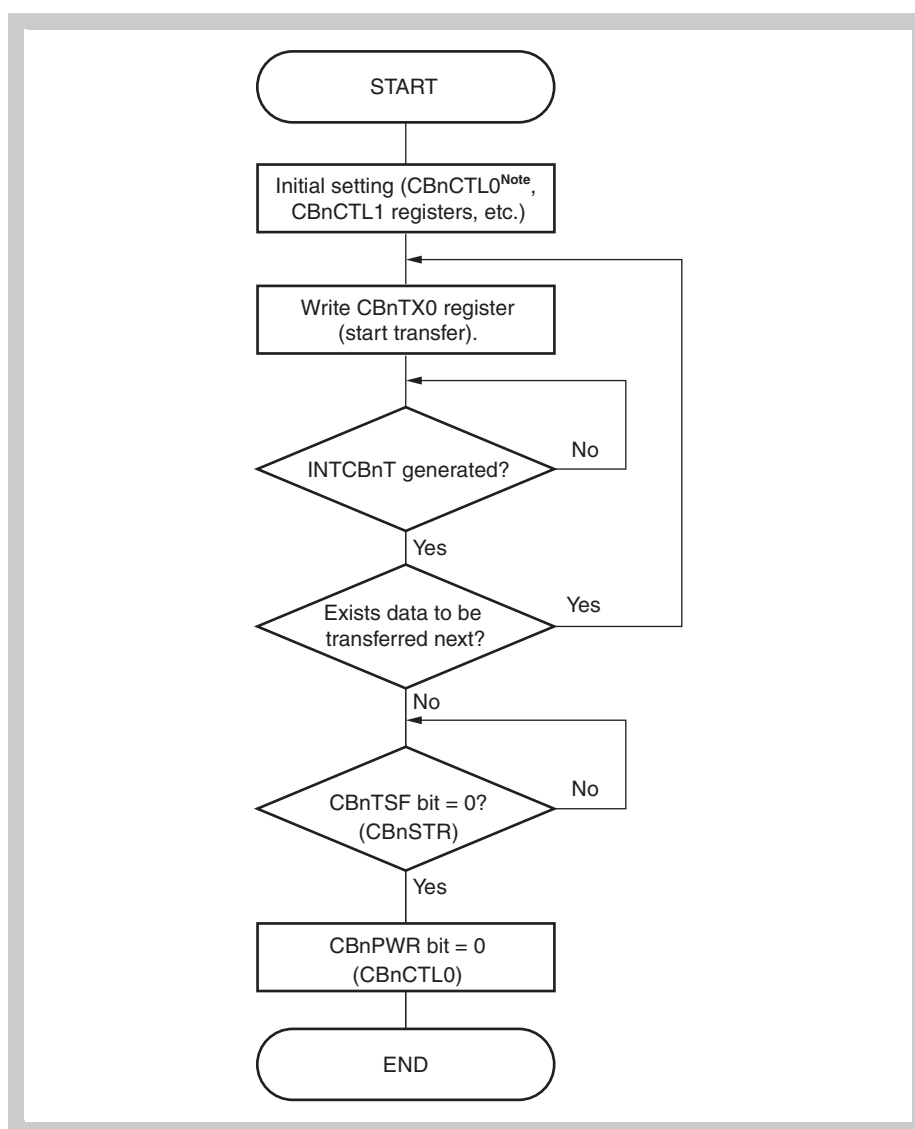
**Caution** In the single mode, data cannot be correctly received if the next transfer clock is input earlier than the CBnRX0 register is read.

**(3) Single transmission/reception**

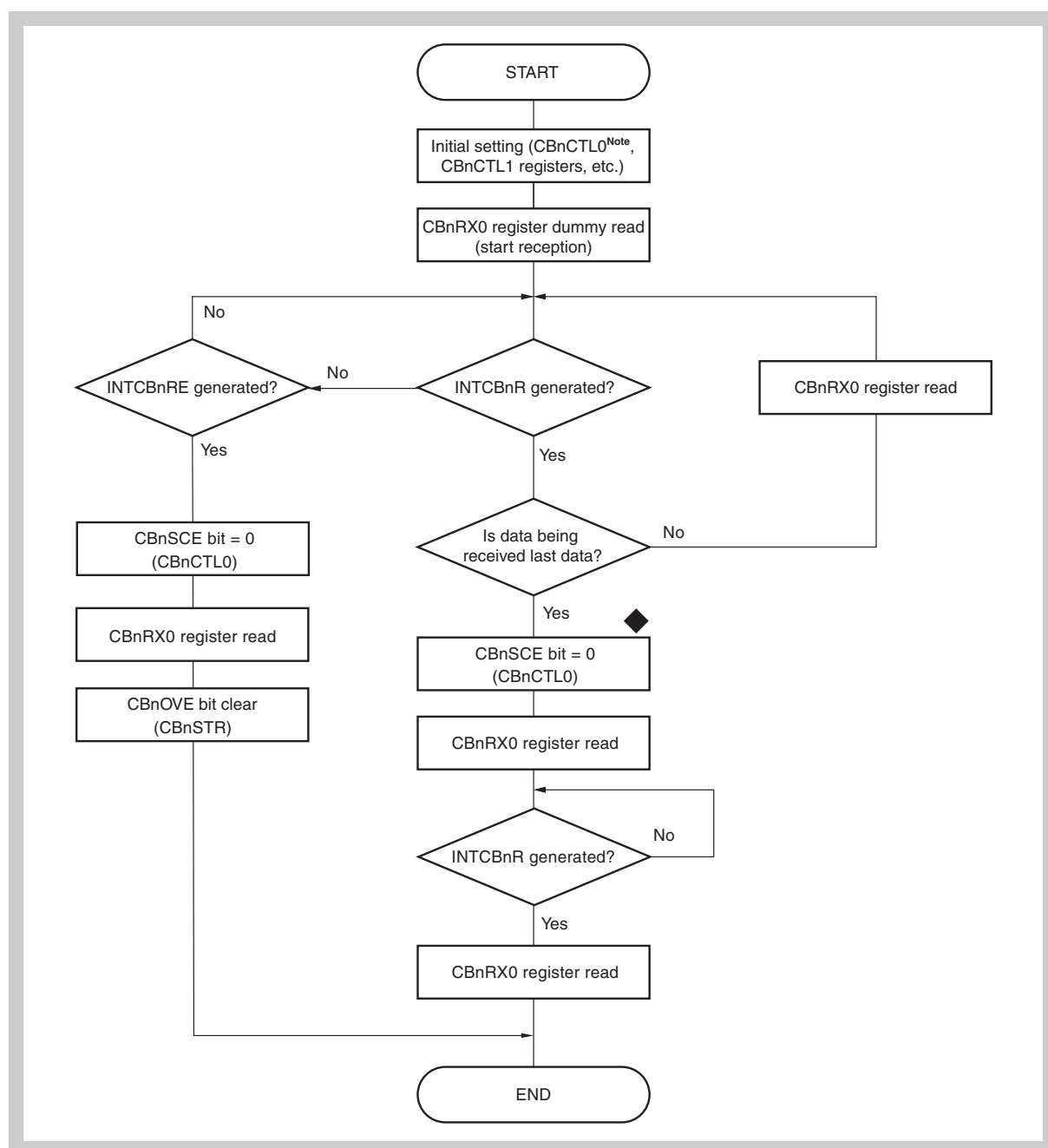
**Note** 1. Set the CBnSCE bit to 1 in the initial setting.

2. If the next transfer is reception only, dummy data is written to the CBnTX0 register.

**Caution** Even in the single mode, the CBnSTR.CBnOVE flag is set to 1. If only transmission is used in the transmission/reception mode, therefore, programming without checking the CBnOVE flag is recommended.

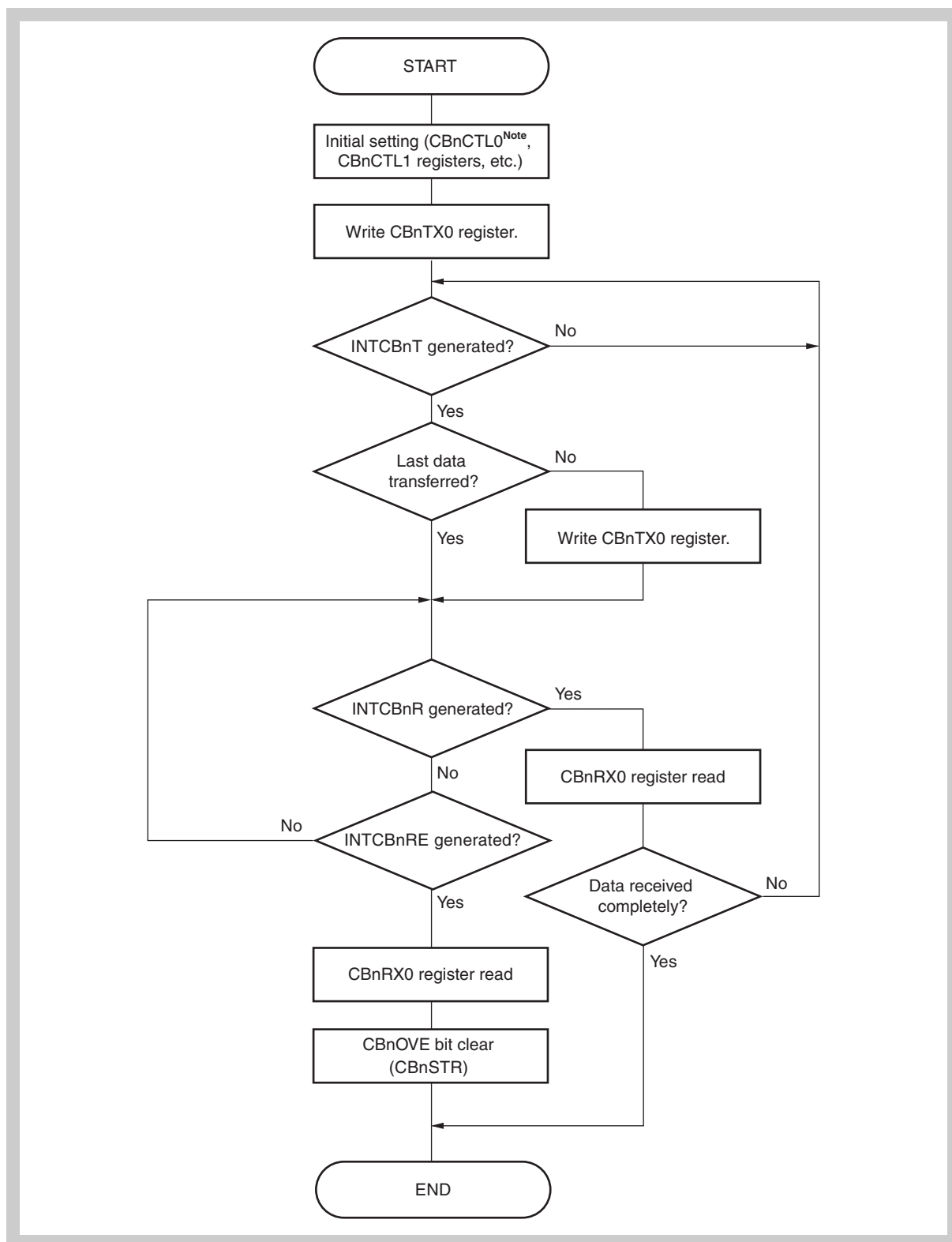
**(4) Continuous transmission**

**Note** Set the CBnSCE bit to 1 in the initial setting.

**(5) Continuous reception**

**Note** Set the CBnSCE bit to 1 in the initial setting.

**Caution** In the master mode, the clock is output without limit when dummy data is read from the CBnRX0 register. To stop the clock, execute the flow marked **◆** in the above flowchart.  
In the slave mode, malfunction due to noise during communication can be prevented by executing the flow marked **◆** in the above flowchart.  
Before resuming communication, set the CBnCTL0.CBnSCE bit to 1, and read dummy data from the CBnRX0 register.

**(6) Continuous transmission/reception**

**Note** Set the CBnSCE bit to 1 in the initial setting.

## 12.7 Baud Rate Generator

### 12.7.1 Overview

Each CSIBn interface is equipped with a dedicated baud rate generator.

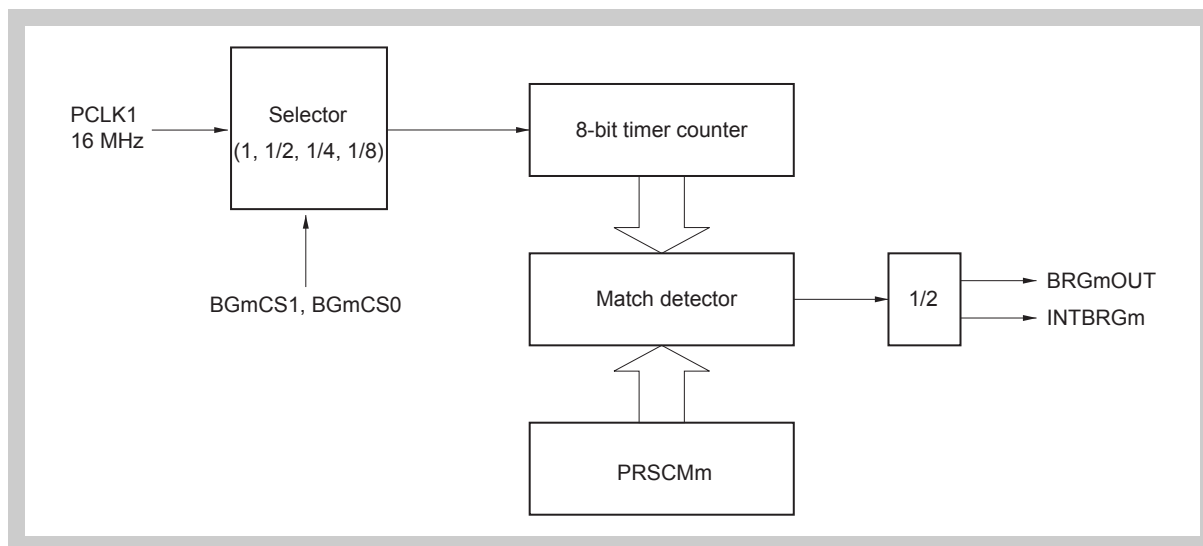


Figure 12-9 Block diagram of baudrate generator (BRGm)

The baud rate generators 0 and 1 (BRG0, BRG1) and CSIB0 and CSIB1 are connected as shown in the following block diagram.

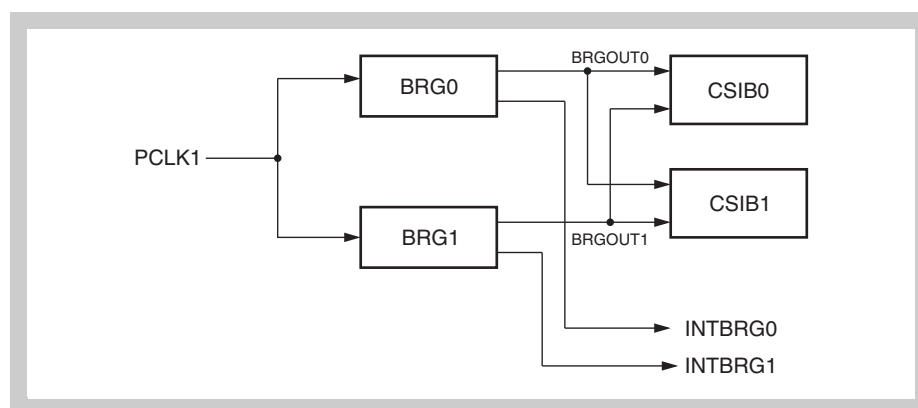


Figure 12-10 Connection between CSIBn and baud rate generators

**Note** An unused baud rate generator (BRGm) can be employed as interval timer generating a dedicated interrupt request (INTBRGm).

## 12.7.2 Baud rate generator registers

### (1) Prescaler mode registers (PRSMm)

The PRSMm register controls generation of the baud rate signal for CSIB.

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

**Address** PRSM0: FFFFFDC0<sub>H</sub>  
PRSM1: FFFFFDD0<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>PRSMm</b>	0	0	0	BGCEm	0	TODISm	BGCSm1	BGCSm0
	R	R	R	R/W	R/W	R/W	R/W	R/W

- Caution**
1. Do not rewrite the PRSMm register during operation.
  2. Set bits BGCSm0, BGCSm1 and TODISm before setting the BGCEn bit to 1.

**Table 12-6 PRSMm register contents**

Bit position	Bit name	Function				
4 2	BGCEm TODISm	Specification of baudrate generator output.				
		BGCEm	TODISm	BRGOUT	INTBRG	Counter
		0	0	Fixed to 0	Fixed to 0	Fixed to 01H
		0	1	Fixed to 0	Fixed to 0	Fixed to 01H
		1	0	Active	Active	Active
		1	1	Fixed to 0	Active	Active
1, 0	BGCSm[1:0]	Baudrate generator input clock setting.				
		BGCSm1	BGCSm0	Input clock selection (f <sub>BGCSm</sub> )	Setting value (k)	
		0	0	PCLK1 (16 Mhz)	0	
		0	1	PCLK1 :2 (8 Mhz)	1	
		1	0	PCLK1 :4 (4 Mhz)	2	
		1	1	PCLK1 :8 (2Mhz)	3	



**(2) Prescaler compare registers (PRSCMm)**

The PRSCMm registers are 8-bit compare registers.

**Access** This register can be read/written in 8-bit units.

**Address** PRSCM0: FFFFDC0<sub>H</sub>  
PRSCM1: FFFFDD0<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>PRSCMm</b>	PRSCMm7	PRSCMm6	PRSCMm5	PRSCMm4	PRSCMm3	PRSCMm2	PRSCMm1	PRSCMm0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Caution**
1. Do not rewrite the PRSCMm register during operation.
  2. Set the PRSCMm register before setting the PRSMm.BGCEn bit to 1.

**Table 12-7 PRSMm register contents**

Bit position	Bit name	Function
7 to 0	PRSCMm[7:0]	Specification of prescaler compare value (N). <b>Note:</b> If PRSCMm[7:0] = 0, then N = 256

### 12.7.3 Baud rate generation

The transmission/reception clock is generated by dividing the main clock. The baud rate generated from the main clock is obtained by the following equation.

$$f_{\text{BRGn}} = \frac{\text{PCLK1}}{2^k \times N \times 2}$$

$f_{\text{BRGn}}$ : BRGm output clock

PCLK1: 16 MHz peripheral clock (refer to “Clock Generator” on page 148)

k: PRSMm.BGCSm[1:0] register setting value ( $0 \leq k \leq 3$ )

N: PRSCMm.PRSCMn[7:0] register value  
if PRSCMm = 00H: N = 256

## 12.8 Cautions

### 12.8.1 CSIB operation stop

**Description** If any channel of CSIBn is operated in slave mode and an external clock signal is input via the SCKBn pin while no transmission or reception sequence is in progress the CSIB may stop operating.

Depending on the CSIB operating configuration the CSIB behaves as described below.

- Transmit mode or transmit/receive mode:  
In transmit mode (CBnCTL0.CBnTXE = 1, CBnCTL0.CBnRXE = 0) or transmit/receive mode (CBnCTL0.CBnTXE = 1, CBnCTL0.CBnRXE = 1) a write operation to the CBnTX0 register may trigger the aforementioned behaviour.  
Any write to the related CBnTX0 register will no longer start a transmission sequence. Furthermore the related transmission interrupt request will not be generated.
- Receive mode:  
In receive mode (CBnCTL0.CBnTXE = 0, CBnCTL0.CBnRXE = 1) a read operation from the CBnRX0 register may trigger the aforementioned behaviour.  
Any read from the related CBnRX0 register will no longer start a receive sequence. Furthermore the related receive interrupt request will not be generated.

The described CSIBn stop condition can be escaped by initiating a system reset or by a sequential clear and set of the CBnCTL0.CBnPWR bit.

**Workaround** In order to avoid the CSIBn stop condition in slave mode take the following precautions.

- Transmit mode or transmit/receive mode:  
Make sure the external clock via the SCKBn pin is not input while writing to the CBnTX0 register after a transmission sequence is finished.
- Receive mode:  
Make sure the external clock via the SCKBn pin is not input While reading from the CBnRX0 register after a reception sequence is finished.

# Chapter 13 Enhanced Queued Clocked Serial Interface (CSIE)

This microcontroller has two instances of the Enhanced Queued Clocked Serial Interface CSIE: CSIE0 and CSIE1.

**Note** Throughout this chapter, the individual instances of CSIE are identified by “n” (n = 0 to 1), for example CEnCTL2 for the Queued CSIn control register 2.

The individual chip select signals are identified by “m” (m = 0 to 7 for CSIE0 and m = 0 to 3 for CSIE1), for example SCSE06 for the chip select pin 6 of CSIE0.

## 13.1 Features

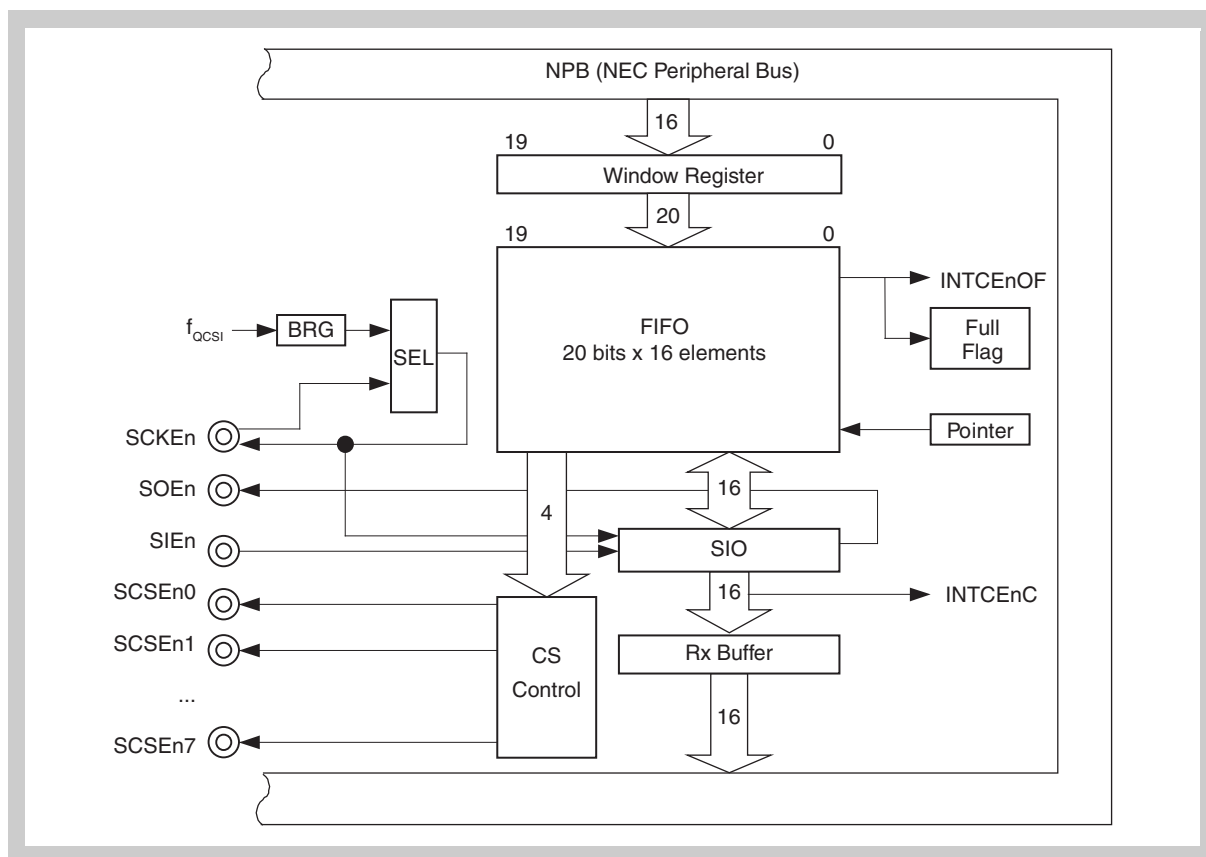
- 3-wire serial synchronous transfers
- The following pins are provided to enable a 3-wire serial interface:

Pin Function	CSIE0	CSIE1
Serial data output	SOE0	SOE1
Serial data input	SIE0	SIE1
Serial clock I/O	SCKE0	SCKE1
Chip select	SCSE00 to SCSE07	SCSE10 to SCSE13

- Master mode and slave mode selectable
- Serial clock and data phase selectable
- Transfer data length selectable from 8 to 16 bits in 1-bit units
- Data transfer with MSB- for LSB-first selectable
- Three selectable transfer modes:
  - transmit only mode
  - receive only mode
  - transmit/receive mode
- Transmit and receive FIFO (16 elements)
- Selection between single transfer mode and block transfer mode
- Internal baud rate generator
- Programmable baud rate through BRG output (master) or slave clock
- DMA transfer of received data to memory available
- Maximum SCKE frequency: 8 MHz

**Note** In all figures of this chapter, except where explicitly mentioned, it is assumed that the chip select signals SCSEn0 to SCSEn7 are set as “active low”, although it is possible to define the active level for each chip select also to “active high”.

### 13.1.1 Queued CSI block diagram



### Figure 13-1 Queued CSI Block Diagram

**Note**

1. CSIE1 offers only four chip select pins: SCSE10 to SCSE13.
2.  $f_{QCSI} = f_{PCLK0} = 32 \text{ MHz}$  is the clock supply of the Queued-CSI macro.

### 13.1.2 Input/Output pins

The table below shows the input/output pins of the CSIE.

**Table 13-1 Input/Output Pins of the CSIE**

Signal name	I/O	Active level	Disabled level	Function
SCKEn	I/O	–	H	Serial clock signal
SIEn	I	–	–	Input serial data signal
SOEn	O	–	–	Output serial data signal
SCSEn0	O	L <sup>a</sup>	H <sup>a</sup>	Serial peripheral chip select signal 0
SCSEn1	O	L <sup>a</sup>	H <sup>a</sup>	Serial peripheral chip select signal 1
SCSEn2	O	L <sup>a</sup>	H <sup>a</sup>	Serial peripheral chip select signal 2
SCSEn3	O	L <sup>a</sup>	H <sup>a</sup>	Serial peripheral chip select signal 3
SCSEn4 <sup>b</sup>	O	L <sup>a</sup>	H <sup>a</sup>	Serial peripheral chip select signal 4
SCSEn5 <sup>b</sup>	O	L <sup>a</sup>	H <sup>a</sup>	Serial peripheral chip select signal 5
SCSEn6 <sup>b</sup>	O	L <sup>a</sup>	H <sup>a</sup>	Serial peripheral chip select signal 6
SCSEn7 <sup>b</sup>	O	L <sup>a</sup>	H <sup>a</sup>	Serial peripheral chip select signal 7

a) The active level is programmable for each chip select.

b) Not available for CSIE1.

## 13.2 Queued CSI Control Registers

The Enhanced Queued CSIEs are controlled and operated by means of the following registers:

**Table 13-2 CSIEn register overview**

Register name	Shortcut	Address
Queued CSI control register 0	CEnCTL0	<base>
Queued CSI control register 1	CEnCTL1	<base> + 01 <sub>H</sub>
Receive data buffer	CEnRX0	<base> + 02 <sub>H</sub>
Chip Select FIFO buffer	CEnCS	<base> + 04 <sub>H</sub>
Transmit data FIFO buffer	CEnTX0	<base> + 06 <sub>H</sub>
CSI Buffer Status Register	CEnSTR	<base> + 08 <sub>H</sub>
Queued CSI control register 2	CEnCTL2	<base> + 09 <sub>H</sub>
Queued CSI control register 3	CEnCTL3	<base> + 0C <sub>H</sub>
Queued CSI control register 4	CEnCTL4	<base> + 0D <sub>H</sub>
CSEn0 enhanced timing	CEnOPT0	<base> + 10 <sub>H</sub>
CSEn1 enhanced timing	CEnOPT1	<base> + 12 <sub>H</sub>
CSEn2 enhanced timing	CEnOPT2	<base> + 14 <sub>H</sub>
CSEn3 enhanced timing	CEnOPT3	<base> + 16 <sub>H</sub>
CSEn4 enhanced timing <sup>a</sup>	CEnOPT4	<base> + 18 <sub>H</sub>
CSEn5 enhanced timing <sup>a</sup>	CEnOPT5	<base> + 1A <sub>H</sub>
CSEn6 enhanced timing <sup>a</sup>	CEnOPT6	<base> + 1C <sub>H</sub>
CSEn7 enhanced timing <sup>a</sup>	CEnOPT7	<base> + 1E <sub>H</sub>

a) only valid for CSIE0

**Table 13-3 CSIEn register base address**

CSIE module	Base address
CSIE0	FFFF FD40 <sub>H</sub>
CSIE1	FFFF FD80 <sub>H</sub>

**(1) CEnCTL0 - Queued CSI control register 0**

The CEnCTL0 register controls the Queued CSI macro's operations.

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

**Address** CE0CTL0: FFFFFFFD40<sub>H</sub>  
CE1CTL0: FFFFFFFD80<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>CEnCTL0</b>	CEnPWR	CEnTXE	CEnRXE	CEnTMS	CEnDIR	CEnSIT	CEnWE	CEnCSM
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Caution**
1. When changing the CEnPWR bit, do not change any other bit at the same time.  
While CEnPWR="0", the only registers that can be accessed are CEnCTL0, CEnTX0, CEnTX0L, and CEnSTR.  
Set the CEnPWR bit before writing any of the other bits of CEnCTL0.
  2. Write of the CEnTMS, CEnDIR, CEnSIT, CEnWE, and CEnCSM bit is only permitted when CEnTXE = 0 and CEnRXE = 0.

**Table 13-4 CEnCTL0 register contents (1/2)**

Bit position	Bit name	Function
7	CEnPWR	CSIE <sub>n</sub> operation disable/enable: 0: Stop CSIE <sub>n</sub> operation clock (reset internal control circuits) 1: Enable CSIE <sub>n</sub> operation <b>Note:</b> To abort reception/transmission forcibly, clear the CEnPWR bit (not the CEnTXE bit or CEnRXE bit) to 0. The clock output stops at this time.
6	CEnTXE	Transmission operation disable/enable: 0: Disable transmission operation 1: Enable transmission operation The SOE <sub>n</sub> output is low level and transmission is disabled when the CEnTXE bit is 0.
5	CEnRXE	Receive operation disable/enable: 0: Disable receive operation 1: Enable receive operation
4	CEnTMS	Transfer mode select: 0: Single transfer mode 1: Block transfer mode
3	CEnDIR	Serial data direction selection: 0: Data is sent/received with MSB first 1: Data is sent/received with LSB first Refer to "Serial data direction select function" on page 336 for further details on the CEnDIR bit setting.
2	CEnSIT	Interrupt delay mode select (INTCE <sub>n</sub> C signal): 0: No delay 1: Half clock delay Refer to "Additional timing and delay selections" on page 343 for further details on the CEnSIT bit setting. <b>Note:</b> This bit is only valid in master mode. In slave mode, no delay is generated.



Table 13-4 CEnCTL0 register contents (2/2)

Bit position	Bit name	Function
1	CEnWE	<p>Transmission wait enable/disable selection:            0: Transmission wait disable. Not insert half clock wait at transmission start.            1: Transmission wait enable. Insert 1 clock (SCKE) wait at transmission start.            Refer to “Additional timing and delay selections” on page 343 for further details on the CEnWE bit setting.</p> <p><b>Note:</b> 1. This bit is only valid in master mode. In slave mode, no wait is generated.            2. This bit has no effect when CEnCTL4.CEnOPE = 1.</p>
0	CEnCSM	<p>Chip select mode specification:            0: Chip select inactive level output disable. Do not force chip select inactive state after each transfer of a data element.            1: Chip select inactive level output enable. Hold all chip selects inactive for half-length SCKE after each transfer of a data element.            Refer to “Additional timing and delay selections” on page 343 for further details on the CEnCSM bit setting.</p> <p><b>Note:</b> 1. This bit is only valid for CEnWE=1.            2. This bit is only valid in master mode. In slave mode, CS signals are always held at inactive level.            3. This bit has no effect when CEnCTL4.CEnOPE = 1.</p>

Table 13-5 Combination of CEnWE and CEnCSM Bits

CEnWE	CEnCSM	Transmission wait	Chip Select inactive level
0	0	None	Not output
0	1	None	Not output
1	0	One SCKE length clock wait	Not output
1	1	One SCKE length clock wait	Output inactive level of half-length SCKE at first half of transmission wait

See “Additional timing and delay selections” on page 343 for further details on the timing selections by CEnSIT, CEnWE, CEnCSM bits.

**(2) CEnCTL1 - Queued CSI control register 1**

The CEnCTL1 register is an 8-bit register that is used to control the serial transfer operations.

It can be read or written in 1-bit and 8-bit units.

Initial value is 07H by reset.

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

**Address** CE0CTL1: FFFFFFFD41<sub>H</sub>  
CE1CTL1: FFFFFFFD81<sub>H</sub>

**Initial Value** 07<sub>H</sub>. T

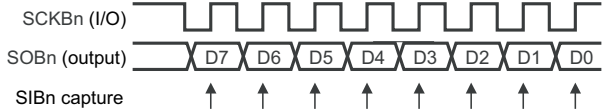
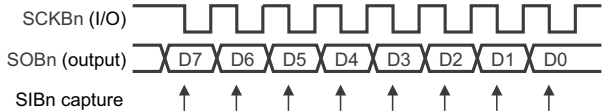
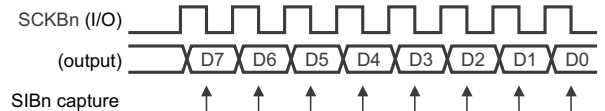
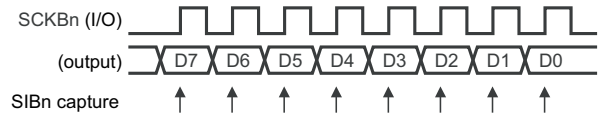
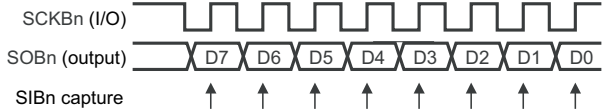
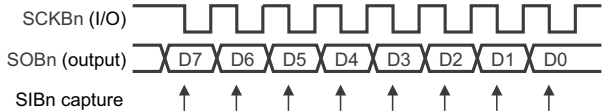
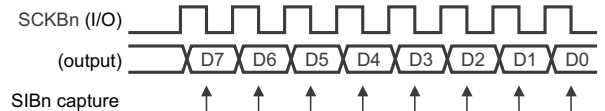
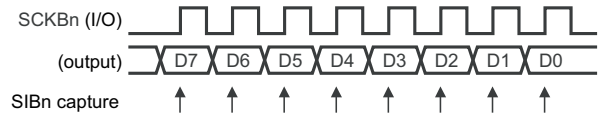
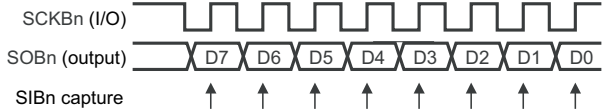
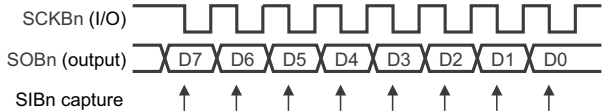
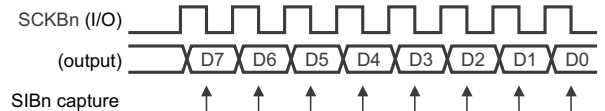
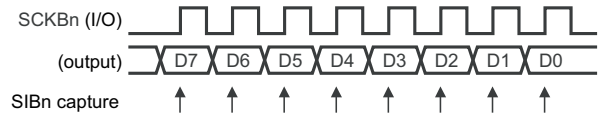
	7	6	5	4	3	2	1	0
<b>CEnCTL1</b>	CEnMDL2	CEnMDL1	CEnMDL0	CEnCKP	CEnDAP	CEnCKS2	CEnCKS1	CEnCKS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Caution** Write to the CEnCTL1 register is permitted only while CEnTXE = 0 and CEnRXE = 0.

**Table 13-6 CBnCTL1 register contents (1/2)**

Bit position	Bit name	Function					
7 to 5	CEnMDL [2:0]	These bits are used to modulate the selected clock.					
		CEnMDL2	CEnMDL1	CEnMDL0	Prescaler output (PRSOUT)	Mode	K
		0	0	0	$f_{QCSI}$	Master Mode	0
		0	0	1	$f_{QCSI}/2$	Master Mode	1
		0	1	0	$f_{QCSI}/4$	Master Mode	2
		0	1	1	$f_{QCSI}/8$	Master Mode	3
		1	0	0	$f_{QCSI}/16$	Master Mode	4
		1	0	1	$f_{QCSI}/32$	Master Mode	5
		1	1	0	$f_{QCSI}/64$	Master Mode	6
		1	1	1	SCKE (input) <sup>a</sup>	Slave Mode	—
a) SCKE pin is configured as input mode for serial transfer clock. $f_{QCSI}$ is the clock supply of the Queued-CSI macro.							

Table 13-6 CBnCTL1 register contents (2/2)

Bit position	Bit name	Function																																																						
4 3	CEnCKP CEnDAP	Specification of data transmission/reception timing in relation to SCKEn.																																																						
		<table><tr><th>CEnCKP</th><th>CEnDAP</th><th>SIEn/SOEn timing in relation to SCKEn</th></tr><tr><td rowspan="4">0   1   1</td><td rowspan="4">0   0   1</td><td>Communication type 1  </td></tr><tr><td>Communication type 2  </td></tr><tr><td>Communication type 3  </td></tr><tr><td>Communication type 4  </td></tr></table>	CEnCKP	CEnDAP	SIEn/SOEn timing in relation to SCKEn	0   1   1	0   0   1	Communication type 1  	Communication type 2  	Communication type 3  	Communication type 4  																																													
		CEnCKP	CEnDAP	SIEn/SOEn timing in relation to SCKEn																																																				
		0   1   1	0   0   1	Communication type 1  																																																				
				Communication type 2  																																																				
Communication type 3  																																																								
Communication type 4  																																																								
2 to 0	CEnCKS[2:0]	Communication clock setting.																																																						
<table><tr><th>CEnCKS2</th><th>CEnCKS1</th><th>CEnCKS0</th><th>Operation clock</th><th>N</th><th>Comment</th></tr><tr><td>0</td><td>0</td><td>0</td><td>BRG disable</td><td>—</td><td>BRG stop for power saving</td></tr><tr><td>0</td><td>0</td><td>1</td><td>PRSOUT/2</td><td>1</td><td>See caution below</td></tr><tr><td>0</td><td>1</td><td>0</td><td>PRSOUT/4</td><td>2</td><td></td></tr><tr><td>0</td><td>1</td><td>1</td><td>PRSOUT/6</td><td>3</td><td></td></tr><tr><td>1</td><td>0</td><td>0</td><td>PRSOUT/8</td><td>4</td><td></td></tr><tr><td>1</td><td>0</td><td>1</td><td>PRSOUT/10</td><td>5</td><td></td></tr><tr><td>1</td><td>1</td><td>0</td><td>PRSOUT/12</td><td>6</td><td></td></tr><tr><td>1</td><td>1</td><td>1</td><td>PRSOUT/14</td><td>7</td><td></td></tr></table>			CEnCKS2	CEnCKS1	CEnCKS0	Operation clock	N	Comment	0	0	0	BRG disable	—	BRG stop for power saving	0	0	1	PRSOUT/2	1	See caution below	0	1	0	PRSOUT/4	2		0	1	1	PRSOUT/6	3		1	0	0	PRSOUT/8	4		1	0	1	PRSOUT/10	5		1	1	0	PRSOUT/12	6		1	1	1	PRSOUT/14	7	
CEnCKS2	CEnCKS1	CEnCKS0	Operation clock	N	Comment																																																			
0	0	0	BRG disable	—	BRG stop for power saving																																																			
0	0	1	PRSOUT/2	1	See caution below																																																			
0	1	0	PRSOUT/4	2																																																				
0	1	1	PRSOUT/6	3																																																				
1	0	0	PRSOUT/8	4																																																				
1	0	1	PRSOUT/10	5																																																				
1	1	0	PRSOUT/12	6																																																				
1	1	1	PRSOUT/14	7																																																				
<div>Caution: CEnMDL[2:0] = [0,0,1] is prohibited when CEnCKS[2:0] = [0,0,0].</div>																																																								

Refer to “Transmission clock select function” on page 338 for further explanation on the transfer clock selection.

The baud rate for the transmission is calculated with the following formula:

$$\text{baudrate} = \frac{f_{\text{QCSI}}}{(2 \times N) \times 2^K} = \frac{f_{\text{QCSI}}}{N \times 2^{(K+1)}} = \frac{f_{\text{QCSI}}}{\text{CEnMDL} \times 2^{(\text{CEnCKS} + 1)}}$$

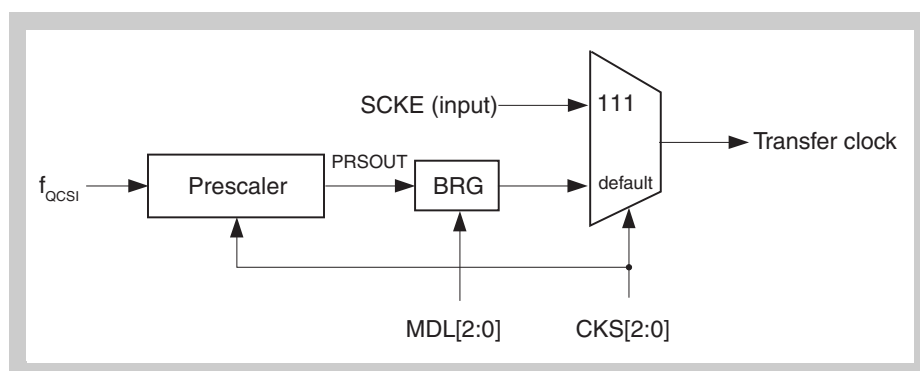


Figure 13-2 Queued CSI Baud Rate Block Diagram

**(3) CEnCTL2 - Queued CSI control register 2**

The CEnCTL2 register is an 8-bit register that specifies the active voltage level of the chip select pins SCSEn0 to SCSEn3 and the Queued CSI data length.

**Access** This register can be read/written in 8-bit units.

**Address** CE0CTL2: FFFFFFFD49<sub>H</sub>  
CE1CTL2: FFFFFFFD89<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>CEnCTL2</b>	CEnCSL3	CEnCSL2	CEnCSL1	CEnCSL0	CEnDL3	CEnDL2	CEnDL1	CEnDL0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Caution** Write to the CEnCTL2 register is permitted only while CEnTXE = 0 and CEnRXE = 0.

**Note** To select the active level of the remaining pins, see “CEnCTL4 - Queued CSI control register 4” on page 323

**Table 13-7 CEnCTL2 register contents**

Bit position	Bit name	Function																																																															
7 to 4	CEnCSL[3:0]	Chip Select active level selection of SCSEnm pin: 0: Chip Select signal SCSEnm is active low 1: Chip Select signal SCSEnm is active high <b>Note:</b> The chip select settings will be reflected at the output pins immediately after the register is set.																																																															
3 to 0	CEnDL[3:0]	Transfer data length specification <table><tr><th>CEnDL3</th><th>CEnDL2</th><th>CEnDL1</th><th>CEnDL0</th><th>Transfer data length</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Data length is 16 bits</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td rowspan="3">Setting prohibited</td></tr><tr><td>0</td><td>0</td><td>1</td><td>x</td></tr><tr><td>0</td><td>1</td><td>x</td><td>x</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>Data length is 8 bits</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>Data length is 9 bits</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>Data length is 10 bits</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td><td>Data length is 11 bits</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td><td>Data length is 12 bits</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>Data length is 13 bits</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td><td>Data length is 14 bits</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>Data length is 15 bits</td></tr></table> <b>Note:</b> Refer to “Data length select function” on page 337 for further explanation on the data length selection.	CEnDL3	CEnDL2	CEnDL1	CEnDL0	Transfer data length	0	0	0	0	Data length is 16 bits	0	0	0	1	Setting prohibited	0	0	1	x	0	1	x	x	1	0	0	0	Data length is 8 bits	1	0	0	1	Data length is 9 bits	1	0	1	0	Data length is 10 bits	1	0	1	1	Data length is 11 bits	1	1	0	0	Data length is 12 bits	1	1	0	1	Data length is 13 bits	1	1	1	0	Data length is 14 bits	1	1	1	1	Data length is 15 bits
CEnDL3	CEnDL2	CEnDL1	CEnDL0	Transfer data length																																																													
0	0	0	0	Data length is 16 bits																																																													
0	0	0	1	Setting prohibited																																																													
0	0	1	x																																																														
0	1	x	x																																																														
1	0	0	0	Data length is 8 bits																																																													
1	0	0	1	Data length is 9 bits																																																													
1	0	1	0	Data length is 10 bits																																																													
1	0	1	1	Data length is 11 bits																																																													
1	1	0	0	Data length is 12 bits																																																													
1	1	0	1	Data length is 13 bits																																																													
1	1	1	0	Data length is 14 bits																																																													
1	1	1	1	Data length is 15 bits																																																													

**(4) CEnCTL3 - Queued CSI Control Register 3**

The CEnCTL3 register is an 8-bit register that specifies the number of data elements to be transferred in block transfer mode.

**Access** This register can be read/written in 8-bit units.

**Address** CE0CTL3: FFFFFFFD4C<sub>H</sub>  
CE1CTL3: FFFFFFFD8C<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>CEnCTL3</b>	0 <sup>Note</sup>	0 <sup>Note</sup>	0 <sup>Note</sup>	0 <sup>Note</sup>	CEnSFN3	CEnSFN2	CEnSFN1	CEnSFN0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Note** Unused bits must be written as 0.

**Table 13-8 CEnCTL3 register contents**

Bit position	Bit name	Function				
3 to 0	CEnSFN[3:0]	Transfer data number specification				
		CEnSFN3	CEnSFN2	CEnSFN1	CEnSFN0	Transfer data number
		0	0	0	0	Transfer 16 data elements
		0	0	0	1	Transfer 1 data element
		0	0	1	0	Transfer 2 data elements
		0	0	1	1	Transfer 3 data elements
		0	1	0	0	Transfer 4 data elements
		0	1	0	1	Transfer 5 data elements
		0	1	1	0	Transfer 6 data elements
		0	1	1	1	Transfer 7 data elements
		1	0	0	0	Transfer 8 data elements
		1	0	0	1	Transfer 9 data elements
		1	0	1	0	Transfer 10 data elements
		1	0	1	1	Transfer 11 data elements
		1	1	0	0	Transfer 12 data elements
		1	1	0	1	Transfer 13 data elements
		1	1	1	0	Transfer 14 data elements
		1	1	1	1	Transfer 15 data elements

**(5) CEnCTL4 - Queued CSI control register 4**

The CEnCTL4 register is an 8-bit register that specifies the active voltage level of the chip select pins SCSEn4 to SCSEn7, the alternative clock and data phase setting and that controls the enhanced chip select timing.

**Access** This register can be read/written in 8-bit units.

**Address** CE0CTL3: FFFFFFFD4C<sub>H</sub>  
CE1CTL3: FFFFFFFD8C<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>CE0CTL4</b>	CE0CSL7	CE0CSL6	CE0CSL5	CE0CSL4	CE0CPA	CE0DPA	CE0OPE	CE0MD
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>CE1CTL4</b>	0	0	0	0	CE1CPA	CE1DPA	CE1OPE	CE1MD
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Caution** Write to the CEnCTL4 register is permitted only while CEnTXE = 0 and CEnRXE = 0.

**Table 13-9 CEnCTL4 register contents (1/2)**

Bit position	Bit name	Function															
7 to 4	CE0CSL[7:4]	<p>Chip Select active level selection of SCSE0m pin:            0: Chip Select signal SCSE0m is active low            1: Chip Select signal SCSE0m is active high</p> <p><b>Note:</b> The chip select settings will be reflected at the output pins immediately after the register is set.</p>															
3 2	CEnCPA CEnDPA	<p>Alternative clock and data phase selection</p> <table border="1"> <thead> <tr> <th>CEnCPA</th><th>CEnDPA</th><th>Alternative Clock and Data phase settings</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>Clock and data phase as CKP=0, DAP=0 (communication type 1)</td></tr> <tr> <td>0</td><td>1</td><td>Clock and data phase as CKP=0, DAP=1 (communication type 2)</td></tr> <tr> <td>1</td><td>0</td><td>Clock and data phase as CKP=1, DAP=0 (communication type 3)</td></tr> <tr> <td>1</td><td>1</td><td>Clock and data phase as CKP=1, DAP=1 (communication type 4)</td></tr> </tbody> </table> <p>The CPA/DPA bits define an alternative setting for the serial communication within the same Queued-CSI macro. Each chip select can be assigned to either the standard (CKP/DAP) or alternative (CPA/DPA) phase setting when the enhanced timing is active (CEnOPE = 1). The CPA/DPA bits have no effect when CEnOPE = 0.</p> <p><b>Note:</b> Please refer to the description “CEnCTL1 - Queued CSI control register 1” on page 318 for details on the timings for each clock and data phase.</p>	CEnCPA	CEnDPA	Alternative Clock and Data phase settings	0	0	Clock and data phase as CKP=0, DAP=0 (communication type 1)	0	1	Clock and data phase as CKP=0, DAP=1 (communication type 2)	1	0	Clock and data phase as CKP=1, DAP=0 (communication type 3)	1	1	Clock and data phase as CKP=1, DAP=1 (communication type 4)
CEnCPA	CEnDPA	Alternative Clock and Data phase settings															
0	0	Clock and data phase as CKP=0, DAP=0 (communication type 1)															
0	1	Clock and data phase as CKP=0, DAP=1 (communication type 2)															
1	0	Clock and data phase as CKP=1, DAP=0 (communication type 3)															
1	1	Clock and data phase as CKP=1, DAP=1 (communication type 4)															

**Table 13-9 CEnCTL4 register contents (2/2)**

Bit position	Bit name	Function
1	CE0OPE	<p>Enhanced timing enable/disable:            0: Enhanced timing is disabled            1: Enhanced timing is enabled.</p> <p>When the enhanced timing is enabled (CE0OPE = 1), the setup time, inter-data time, hold time and idle time for each chip select signal can be defined using the CEnOPTm registers. For CE0OPE = 0, the standard timing applies to all chip select signals.</p>
0	CE0MD	<p>Chip select mode specification:            0: 4 chip select mode. Each of the 4 bits of the CEnCS register (CEnCS[3:0]) represents a chip select line.            1: 8 chip select mode. The combination of the lower 3 bits of the CEnCS register (CEnCS[2:0]) select one out of 8 available chip select lines.</p> <ul style="list-style-type: none"> <li>4 chip select mode              In 4 chip select mode, only 4 chip selects can be used for the communication, but multiple chip selects can be active for the same communication ("broadcast"), as each bit represents a chip select signal.</li> <li>8 chip select mode              In 8 chip select mode, all 8 chip selects are accessible, although only one out of the 8 chip selects can be selected as active for a communication.</li> </ul> <hr/> <p><b>Caution:</b> The 8 chip select mode is prohibited for CSIE1. Therefore keep CE1MD = 0.</p> <hr/>



**(6) CEnRX0 - Receive data buffer registers**

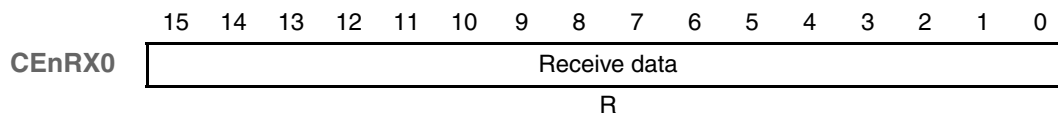
The CEnRX0 register is a 16-bit register or separated as upper 8 bits (CEnRX0H) and lower 8 bits (CEnRX0L), that is used to store receive data.

**Access** This register can be read-only in 16-bit units, or in 8-bit units as CEnRX0L register for lower 8 bits or as CEnRX0H register for upper 8 bits.

**Address** CE0RX0, CE0RX0L: FFFFD42<sub>H</sub>      CE0RX0H: FFFFD43<sub>H</sub>  
CE1RX0, CE1RX0L: FFFFD82<sub>H</sub>      CE1RX0H: FFFFD83<sub>H</sub>

**Initial Value**

- in single transfer mode (CEnCTL0.CEnTMS = 0): 0000H
- in block transfer mode (CEnCTL0.CEnTMS = 1): undefined.



**Caution** The receive data buffer register is considered as emptied by the application software whenever the lower 8 bits of the register are read. It is therefore necessary to read CEnRX0H before CEnRX0L when 8-bit access is used.

**(7) CEnCS - Chip select data buffer registers**

The CEnCS register is a 16-bit register than can be also accessed as 8 bit register (CEnCSL) that stores Chip Select data.

Following the FIFO write pointer, the value written to CEnCS is stored in the FIFO data buffer as Chip Select bits. The value is stored to these bits when the transmit data is written to its register (CEnTX0 or CEnTX0L)

**Access** This register can be read/written in 8-bit or 16-bit units.

**Address** CE0CS, CE0CSL: FFFFFFFD44<sub>H</sub>  
CE1CS, CE1CSL: FFFFFFFD84<sub>H</sub>

**Initial Value** FFFF<sub>H</sub>.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>CEnCS</b>	-	-	-	-	-	-	-	-	-	-	-	-	CEn CS3	CEn CS2	CEn CS1	CEn CS0
	R															

- Caution**
1. Write of CEnCS register is prohibited when CEnCTL0.CEnPWR = 0.
  2. The Chip Select register is stored in the FIFO buffer when the transmit data is written to its register. It is therefore necessary to write the CEnCS register before the CEnTX0 (CEnTX0L) register.
  3. In 8 chip select mode (CEnCTL4.CEnMD = 1) CEnCS3 must be set to 0.

Depending on the chip select mode (defined by the CEnMD flag - see "CEnCTL4 - Queued CSI control register 4" on page 323 for details) the CEnCS[3:0] bits have a different meaning:

**(a) CSIE0**

- 4 chip select mode (CE0MD = 0)

CE0CS bits				Chip select signals			
CE0CS3	CE0CS2	CE0CS1	CE0CS0	SCSE03	SCSE02	SCSE01	SCSE00
0	0	0	0	selected	selected	selected	selected
0	0	0	1	selected	selected	selected	–
0	0	1	0	selected	selected	–	selected
0	0	1	1	selected	selected	–	–
0	1	0	0	selected	–	selected	selected
0	1	0	1	selected	–	selected	–
0	1	1	0	selected	–	–	selected
0	1	1	1	selected	–	–	–
1	0	0	0	–	selected	selected	selected
1	0	0	1	–	selected	selected	–
1	0	1	0	–	selected	–	selected
1	0	1	1	–	selected	–	–
1	1	0	0	–	–	selected	selected
1	1	0	1	–	–	selected	–
1	1	1	0	–	–	–	selected
1	1	1	1	–	–	–	–

**Caution** "When the enhanced timing mode is enabled by setting CEnCTL4.CEnOPE = 1, setting of CEnCS.CEnCS[3:0] = F<sub>H</sub> is prohibited."

• 8 chip select mode (CE0MD = 1)

CE0CS3	CE0CS2	CE0CS1	CE0CS0	Active chip select signal
0 <sup>a</sup>	0	0	0	SCSE00 selected
	0	0	1	SCSE01 selected
	0	1	0	SCSE02 selected
	0	1	1	SCSE03 selected
	1	0	0	SCSE04 selected
	1	0	1	SCSE05 selected
	1	1	0	SCSE06 selected
	1	1	1	SCSE07 selected

a) In 8 chip select mode CE0CS3 must be set to 0.

(b) CSIE1: 4 chip select mode (CE1MD = 0)

- Note**
1. The active level for each chip select is defined in the CEnCTL2 and CEnCTL4 register (CEnCSL[7:0] bits).
  2. In 4 chip select mode, multiple chip selects can be active at the same time. When enhanced timing is enabled in 4 chip select mode, the Queued-CSI macro will use the timing of the enabled chip select with the lowest number. But to ensure proper communication, it is mandatory that the timings of all chip selects that can be active at a same time are set to the same values.

CE1CS bits				Chip select signals			
CE1CS3	CE1CS2	CE1CS1	CE1CS0	SCSE13	SCSE12	SCSE11	SCSE10
0	0	0	0	selected	selected	selected	selected
0	0	0	1	selected	selected	selected	–
0	0	1	0	selected	selected	–	selected
0	0	1	1	selected	selected	–	–
0	1	0	0	selected	–	selected	selected
0	1	0	1	selected	–	selected	–
0	1	1	0	selected	–	–	selected
0	1	1	1	selected	–	–	–
1	0	0	0	–	selected	selected	selected
1	0	0	1	–	selected	selected	–
1	0	1	0	–	selected	–	selected
1	0	1	1	–	selected	–	–
1	1	0	0	–	–	selected	selected
1	1	0	1	–	–	selected	–
1	1	1	0	–	–	–	selected
1	1	1	1	–	–	–	–

**Caution** "When the enhanced timing mode is enabled by setting CEnCTL4.CEnOPE = 1, setting of CEnCS.CEnCS[3:0] = F<sub>H</sub> is prohibited."

**(8) CEnTX0 - Transmission data buffer registers**

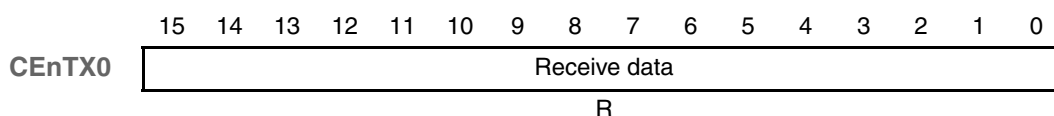
The CEnTX0 register is a 16-bit buffer register, or separated as upper 8 bits (CEnTX0H) and lower 8 bits (CEnTX0L), that stores transmission data.

Incrementing and following the FIFO buffer write pointer, the value written to CEnTX0 is stored to the FIFO buffer as transmission data.

**Access** This register can be read/written in 16-bit units, or in 8-bit units as CEnTX0L register for lower 8 bits or as CEnTX0H register for upper 8 bits.

**Address** CE0TX0, CE0TX0L: FFFFFFFD46<sub>H</sub>      CE0TX0H: FFFFFFFD47<sub>H</sub>  
CE1TX0, CE1TX0L: FFFFFFFD86<sub>H</sub>      CE1TX0H: FFFFFFFD87<sub>H</sub>

**Initial Value** 0000<sub>H</sub>



- Caution**
1. Write to CEnTX0 register is prohibited when CEnPWR = 0.
  2. The transmit data buffer register is considered as written whenever the lower 8 bits of the register are written. It is therefore necessary to write CEnTX0H before CEnTX0L when 8-bit access is used.

**(9) CEnSTR - Queued-CSI status registers**

The CEnSTR register is an 8-bit register that shows the status of the Queued CSI.

**Access** This register can be read/written in 1-bit and 8-bit units.  
The bits CEnFLF, CEnEMF, CEnTSF and CEnSFP[3:0] are read only.

**Address** CE0STR: FFFFFFFD48<sub>H</sub>  
CE1STR: FFFFFFFD88<sub>H</sub>

**Initial Value** 20<sub>H</sub>

	7	6	5	4	3	2	1	0
<b>CEnSTR</b>	CEnPCT	CEnFLF	CEnEMF	CEnTSF	CEnSFP3	CEnSFP2	CEnSFP1	CEnSFP0
	R/W	R	R	R	R	R	R	R

- Caution**
1. CEnFLF, CEnEMF, CEnTSF and CEnSFP[3:0] are continuously updated with the current status of the Queued CSI. This means that a value read might be outdated shortly after the read was executed.
  2. When writing accidentally a 17th data element into the FIFO, an overflow interrupt (INTCEnOFF) will occur to indicate the error.

**Table 13-10 CEnSTR register contents (1/2)**

Bit position	Bit name	Function
7	CEnPCT	FIFO buffer pointer clear command: 0: No operation 1: Clear all FIFO pointers to "0" <b>Note:</b> Reading of the CEnPCT bit always returns "0".
6	CEnFLF	FIFO buffer full status flag: 0: FIFO buffer is not full 1: FIFO buffer is full <b>Note:</b> The CEnFLFbit can only be read.
5	CEnEMF	FIFO buffer empty status flag: 0: FIFO buffer is not empty 1: FIFO buffer is empty <b>Note:</b> The CEnEMFbit can only be read.
4	CEnTSF	Transmission status flag: 0: Idle state 1: Transmission in on going or preparing In single transfer mode, this bit holds "1" from transmission start until FIFO empty. In block transfer mode, this bit holds "1" from transmission start until finish transferring all data to be sent <b>Note:</b> 1. The CEnTSFbit can only be read. 2. This bit is cleared to "0" by CEnCTL0.CEnPWR = 0 or (CEnCTL0.CEnTXE = 0 and CEnCTL0.CEnRXE = 0).

**Table 13-10 CEnSTR register contents (2/2)**

Bit position	Bit name	Function
3 to 0	CEnSFP[3:0]	<p>Transmission data count:</p> <ul style="list-style-type: none"> <li>• In Single transfer mode, CEnSFP[3:0] indicates the number of remaining transfers in the FIFO. This value can be understood as (Write FIFO pointer) - (SIO load pointer)</li> <li>• In FIFO transfer mode, CEnSFP[3:0] indicates the number of data transfers completed.</li> </ul> <p>In case of CEnSFP[3:0] = 0<sub>H</sub>:</p> <ul style="list-style-type: none"> <li>- CEnEMF = 0 and CEnSFP[3:0] = 0<sub>H</sub>: Number of receptions completed = 0</li> <li>- CEnEMF = 1 and CEnSFP[3:0] = 0<sub>H</sub>: Number of receptions completed = 16</li> </ul> <p><b>Note:</b> The CEnSFP[3:0] bits hold its value until reset or CEnPCT = 1 is written.</p>

**(10) CEnOPTm - Queued CSI enhanced timing registers**

The CEnOPTm registers are 16-bit register, that are used to define the timing for the chip select signal CEnm when enhanced timing is active (CEnOPE = 1). For a detailed description of the enhanced chip select timing refer to “Enhanced chip select timing” on page 348.

**Access** This register can be read/written in 16-bit units.

**Address** CE0OPT0: FFFFFFFD50<sub>H</sub>                      CE1OPT0: FFFFFFFD90<sub>H</sub>  
 CE0OPT1: FFFFFFFD52<sub>H</sub>                      CE1OPT1: FFFFFFFD92<sub>H</sub>  
 CE0OPT2: FFFFFFFD54<sub>H</sub>                      CE1OPT2: FFFFFFFD94<sub>H</sub>  
 CE0OPT3: FFFFFFFD56<sub>H</sub>                      CE1OPT3: FFFFFFFD96<sub>H</sub>  
 CE0OPT4: FFFFFFFD58<sub>H</sub>  
 CE0OPT5: FFFFFFFD5A<sub>H</sub>  
 CE0OPT6: FFFFFFFD5C<sub>H</sub>  
 CE0OPT7: FFFFFFFD5E<sub>H</sub>

**Initial Value** 0002<sub>H</sub>.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>CEnOPTm</b>	CEn PHSm	CEn IDLm	CEn IDm2	CEn IDm1	CEn IDm0	CEn HDm2	CEn HDm1	CEn HDm0	CEn DLm1	CEn DLm0	CEn INm2	CEn INm1	CEn INm0	CEn SPm2	CEn SPm1	CEn SPm0
	R/W															

- Caution**
1. The CEnOPTm registers can be written only while CEnCTL0.CEnTXE = 0 and CEnCTL0.CEnRXE = 0.
  2. CEnOPT4 to CEnOPT7 registers are not available for CSIE1.

**Table 13-11 CEnOPTm register contents (1/3)**

Bit position	Bit name	Function
15	CEnPHSm	Clock/data phase selection: 0: Clock/data phase selection defined by CKP/DAP are used for chip select m. 1: Alternative clock/data phase selection defined by CKA/DAP are used for chip select m.
14	CEnIDLm	Chip select idle enforcement: 0: Chip select m becomes idle whenever a different chip select value is defined the CEnCS[3:0] bits. If the chip select value remains the same or the buffer becomes empty, the chip select stays active. 1: An idle state is always inserted after each transfer. <b>Note:</b> When the multiple data is transferred using the same chip select setting, the chip select line remains active. The CEnIDLm bit forced the chip select line to become idle after each transfer, even if the same chip select setting is used for the next transfer. For more details refer to “Enforced chip select idle setting” on page 348.



Table 13-11 CEnOPTm register contents (2/3)

Bit position	Bit name	Function																																															
13 to 11	CEnIDm[2:0]	Chip select m idle timing specification																																															
		<table><tr><th>CEnIDm2</th><th>CEnIDm1</th><th>CEnIDm0</th><th>Chip select m idle timing <sup>Note</sup></th></tr><tr><td>0</td><td>0</td><td>0</td><td>chip select m is idle for 0 SCKE</td></tr><tr><td>0</td><td>0</td><td>1</td><td>chip select m is idle for 0.5 SCKE</td></tr><tr><td>0</td><td>1</td><td>0</td><td>chip select m is idle for 1 SCKE</td></tr><tr><td>0</td><td>1</td><td>1</td><td>chip select m is idle for 2 SCKE</td></tr><tr><td>1</td><td>0</td><td>0</td><td>chip select m is idle for 3 SCKE</td></tr><tr><td>1</td><td>0</td><td>1</td><td>chip select m is idle for 4 SCKE</td></tr><tr><td>1</td><td>1</td><td>0</td><td>chip select m is idle for 6 SCKE</td></tr><tr><td>1</td><td>1</td><td>1</td><td>chip select m is idle for 8 SCKE</td></tr></table>	CEnIDm2	CEnIDm1	CEnIDm0	Chip select m idle timing <sup>Note</sup>	0	0	0	chip select m is idle for 0 SCKE	0	0	1	chip select m is idle for 0.5 SCKE	0	1	0	chip select m is idle for 1 SCKE	0	1	1	chip select m is idle for 2 SCKE	1	0	0	chip select m is idle for 3 SCKE	1	0	1	chip select m is idle for 4 SCKE	1	1	0	chip select m is idle for 6 SCKE	1	1	1	chip select m is idle for 8 SCKE											
		CEnIDm2	CEnIDm1	CEnIDm0	Chip select m idle timing <sup>Note</sup>																																												
		0	0	0	chip select m is idle for 0 SCKE																																												
		0	0	1	chip select m is idle for 0.5 SCKE																																												
		0	1	0	chip select m is idle for 1 SCKE																																												
		0	1	1	chip select m is idle for 2 SCKE																																												
		1	0	0	chip select m is idle for 3 SCKE																																												
		1	0	1	chip select m is idle for 4 SCKE																																												
		1	1	0	chip select m is idle for 6 SCKE																																												
1	1	1	chip select m is idle for 8 SCKE																																														
<b>Note:</b> A minimum idle period of around 2-3 macro clock cycles ( $f_{QCSl}$ ) is applied whenever a CS is set from active to inactive. This time is an addition to the time set by the CEnIDm[2:0] bits.																																																	
10 to 8	CEnHDm[2:0]	Chip select m hold timing specification																																															
		<table><tr><th rowspan="2">CEnHDm2</th><th rowspan="2">CEnHDm1</th><th rowspan="2">CEnHDm0</th><th colspan="2">Chip select m hold timing</th></tr><tr><th>CEnSIT=0</th><th>CEnSIT=1</th></tr><tr><td>0</td><td>0</td><td>0</td><td>no hold time</td><td>0.5 SCKE</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0.5 SCKE</td><td>1 SCKE</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1 SCKE</td><td>1.5 SCKE</td></tr><tr><td>0</td><td>1</td><td>1</td><td>2 SCKE</td><td>2.5 SCKE</td></tr><tr><td>1</td><td>0</td><td>0</td><td>3 SCKE</td><td>3.5 SCKE</td></tr><tr><td>1</td><td>0</td><td>1</td><td>4 SCKE</td><td>4.5 SCKE</td></tr><tr><td>1</td><td>1</td><td>0</td><td>6 SCKE</td><td>6.5 SCKE</td></tr><tr><td>1</td><td>1</td><td>1</td><td>8 SCKE</td><td>8.5 SCKE</td></tr></table>	CEnHDm2	CEnHDm1	CEnHDm0	Chip select m hold timing		CEnSIT=0	CEnSIT=1	0	0	0	no hold time	0.5 SCKE	0	0	1	0.5 SCKE	1 SCKE	0	1	0	1 SCKE	1.5 SCKE	0	1	1	2 SCKE	2.5 SCKE	1	0	0	3 SCKE	3.5 SCKE	1	0	1	4 SCKE	4.5 SCKE	1	1	0	6 SCKE	6.5 SCKE	1	1	1	8 SCKE	8.5 SCKE
		CEnHDm2				CEnHDm1	CEnHDm0	Chip select m hold timing																																									
			CEnSIT=0	CEnSIT=1																																													
		0	0	0	no hold time	0.5 SCKE																																											
		0	0	1	0.5 SCKE	1 SCKE																																											
		0	1	0	1 SCKE	1.5 SCKE																																											
		0	1	1	2 SCKE	2.5 SCKE																																											
		1	0	0	3 SCKE	3.5 SCKE																																											
		1	0	1	4 SCKE	4.5 SCKE																																											
1	1	0	6 SCKE	6.5 SCKE																																													
1	1	1	8 SCKE	8.5 SCKE																																													
<b>Note:</b> The hold times given above assume that a new transfer element is waiting in the transfer buffer. When the transfer queue becomes empty, the hold time will be longer than the times given above.																																																	
7, 6	CEnDL[1:0]	Chip select m data length specification																																															
		<table><tr><th>CEnDLm1</th><th>CEnDLm0</th><th>Chip select m data length</th></tr><tr><td>0</td><td>×</td><td>The data length defined by CEnCTL2.CEnDL[3:0] is used as data length for chip select m</td></tr><tr><td>1</td><td>0</td><td>The data length for chip select m is 8 bits (regardless of CEnCTL2.CEnDL[3:0] setting)</td></tr><tr><td>1</td><td>1</td><td>The data length for chip select m is 16 bits (regardless of CEnCTL2.CEnDL[3:0] setting)</td></tr></table>	CEnDLm1	CEnDLm0	Chip select m data length	0	×	The data length defined by CEnCTL2.CEnDL[3:0] is used as data length for chip select m	1	0	The data length for chip select m is 8 bits (regardless of CEnCTL2.CEnDL[3:0] setting)	1	1	The data length for chip select m is 16 bits (regardless of CEnCTL2.CEnDL[3:0] setting)																																			
		CEnDLm1	CEnDLm0	Chip select m data length																																													
		0	×	The data length defined by CEnCTL2.CEnDL[3:0] is used as data length for chip select m																																													
1	0	The data length for chip select m is 8 bits (regardless of CEnCTL2.CEnDL[3:0] setting)																																															
1	1	The data length for chip select m is 16 bits (regardless of CEnCTL2.CEnDL[3:0] setting)																																															

Table 13-11 CEnOPTm register contents (3/3)

Bit position	Bit name	Function				
5 to 3	CEnINm[2:0]	Chip select m inter-data delay specification				
		CEnINm2	CEnINm1	CEnINm0	Chip select m inter-data delay timing	
					CEnSIT=0	CEnSIT=1
		0	0	0	no inter-data delay	0.5 SCKE
		0	0	1	0.5 SCKE	
		0	1	0	1 SCKE	
		0	1	1	2 SCKE	
		1	0	0	3 SCKE	
		1	0	1	4 SCKE	
		1	1	0	6 SCKE	
		1	1	1	8 SCKE	
		<b>Note:</b> 1. Inter-data delay only applies when no idle state is enforced between data (CEnIDLm=0).				
		2. The inter-data times given above assume that a new transfer element is waiting in the transfer buffer. When the transfer queue becomes empty, the hold time will be longer than the times given above.				
2 to 0	CEnSPm[2:0]	Chip select m setup timing specification				
		CEnSPm2	CEnSPm1	CEnSPm0	Chip select m setup timing	
		0	0	0	No setup delay	
		0	0	1	Setup delay is 0.5 SCKE	
		0	1	0	Setup delay is 1 SCKE	
		0	1	1	Setup delay is 2 SCKE	
		1	0	0	Setup delay is 3 SCKE	
		1	0	1	Setup delay is 4 SCKE	
		1	1	0	Setup delay is 6 SCKE	
		1	1	1	Setup delay is 8 SCKE	

## 13.3 Explanation of Queued CSI Functions

### 13.3.1 Transmit buffer

Chip select data and transmission data can be stored to the transmit FIFO buffer continuously by writing to the CEnCS register and CEnTX0 register. The Writing FIFO pointer is automatically incremented when data is written to CEnTX0. The size of the transmit FIFO buffer is 20 bits × 16 entries.

In slave mode, it is not necessary to set the chip select data.

The transfer start condition (assuming that CEnEMF = 0) is to write to the lower bits of CEnTX0 register. If the transmission data length is 9 bits or more, data should be written by a 16-bit write to CEnTX0 or by two 8-bit writes with first CEnTX0H, then CEnTX0L (in that order). When transmission data length is 8 bits, data should be set by one 8-bit write to CEnTX0L or by one 16-bit write to CEnTX0. For the 16-bit write the upper 8 bits are ignored in the 8-bit transmission.

The CEnFLF bit in the CEnSTR status register is set “1” after 16 writes have been made to the transmit buffer, assuming the Writing FIFO pointer was reset previously.

When a transmission write is attempted while the FIFO is full (CEnFLF=1), the interrupt INTCEnOFF is generated to indicate an overflow. In that case, the transmission and chip select data are discarded and not stored.

When a transfer cycle is finished and the SIO Loading FIFO pointer is incremented, the FIFO buffer of the previous location is considered empty in case of single transfer mode. Refer to “Description of the block transfer mode” on page 341 for more information on block transfer mode.

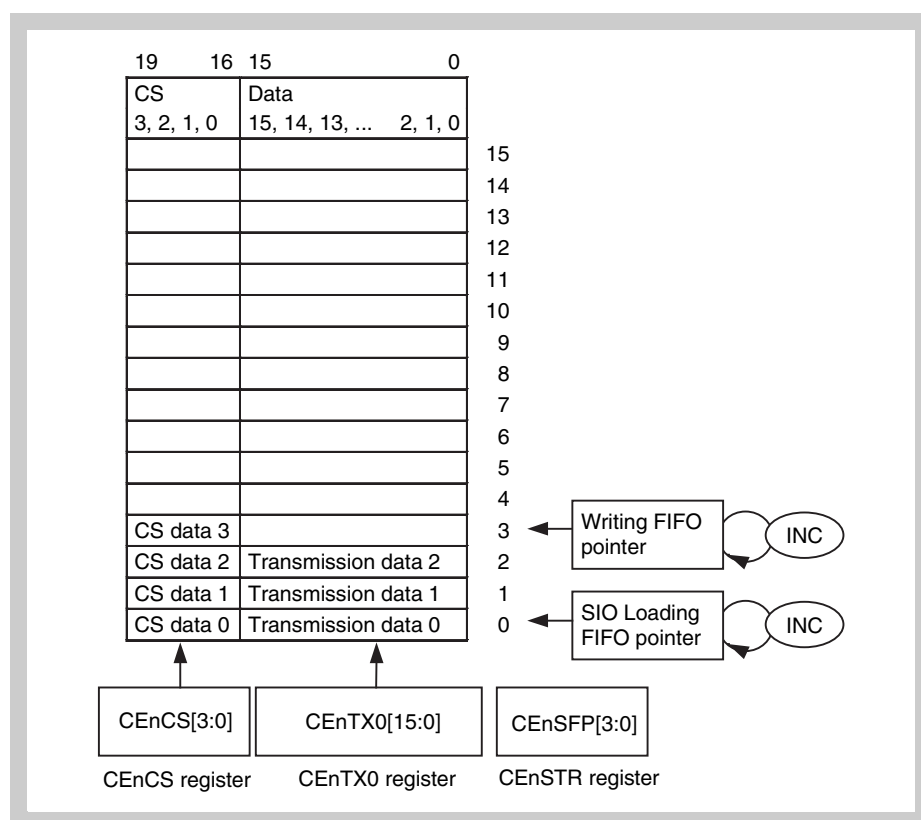


Figure 13-3 Transmit buffer

### 13.3.2 Serial data direction select function

The serial data direction is selectable using the CEnDIR bit in the CEnCTL0 register. The examples below show the communication for data length of 8 bit (CEnDL[3:0] = [1,0,0,0]):

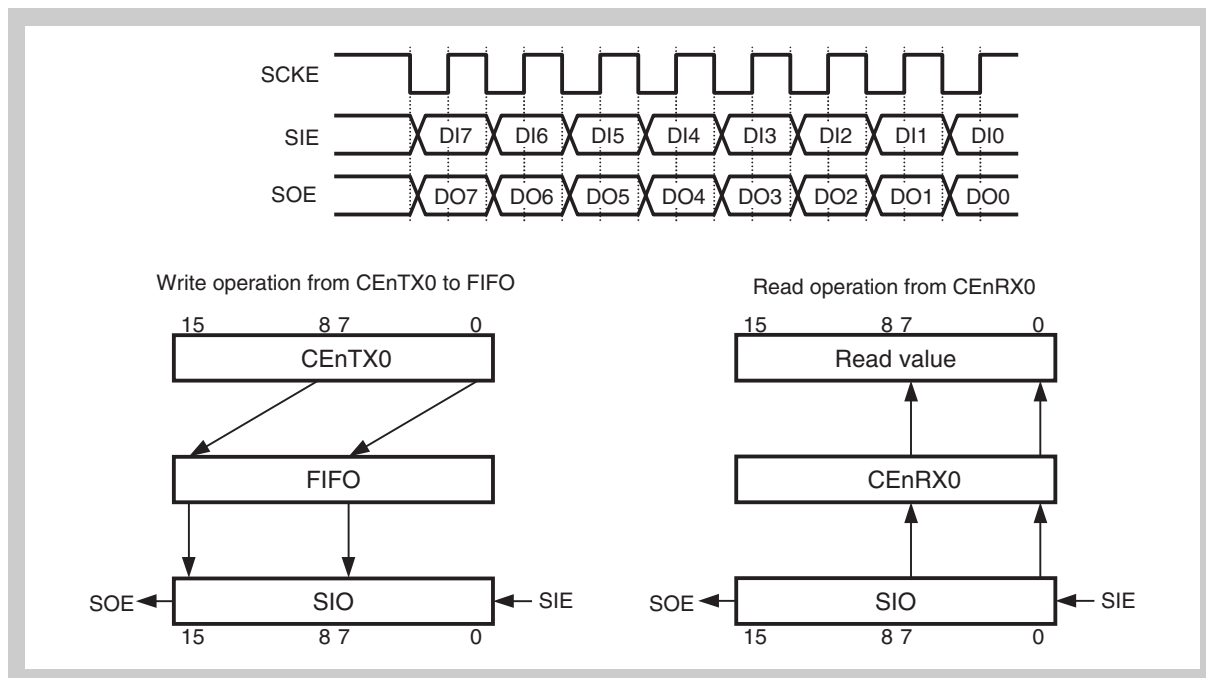


Figure 13-4 Serial data direction select function - msb first (CEnDIR = 0)

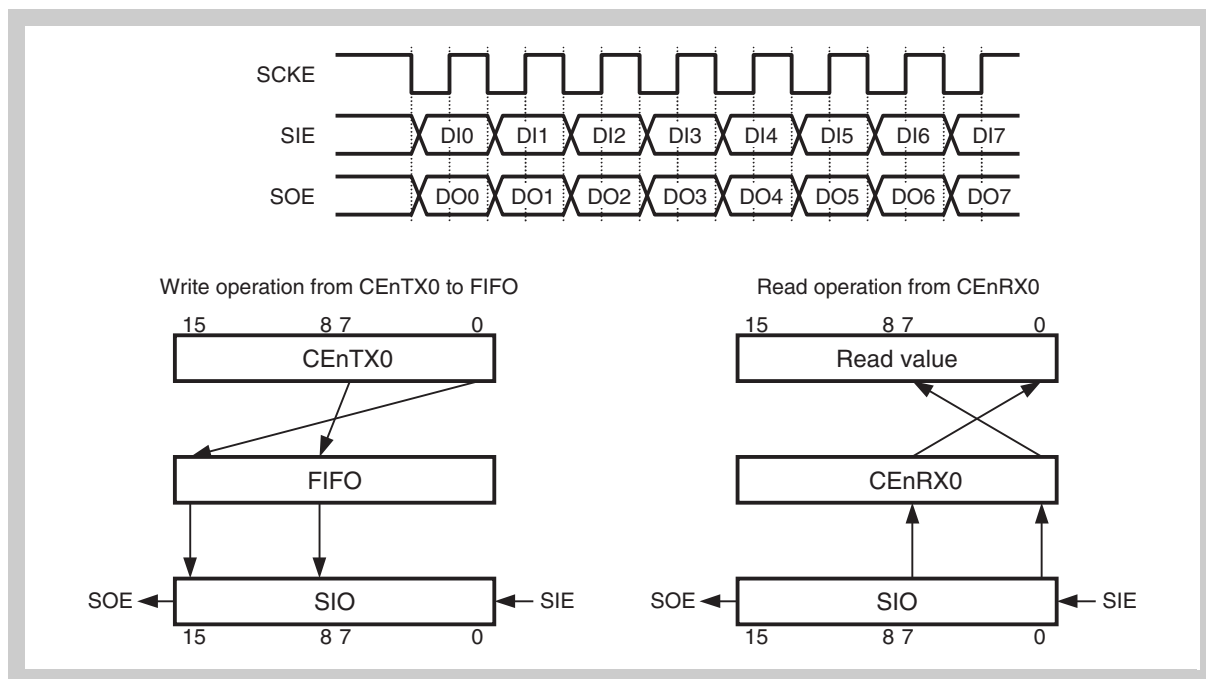


Figure 13-5 Serial data direction select function - LSB first (CEnDIR = 0)

### 13.3.3 Data length select function

Transmission data length is selectable from 8 bits to 16 bits using the CEnDL[3:0] bits in CEnCTL2 register. The examples below show the communication with MSB first (CEnDIR = 1):

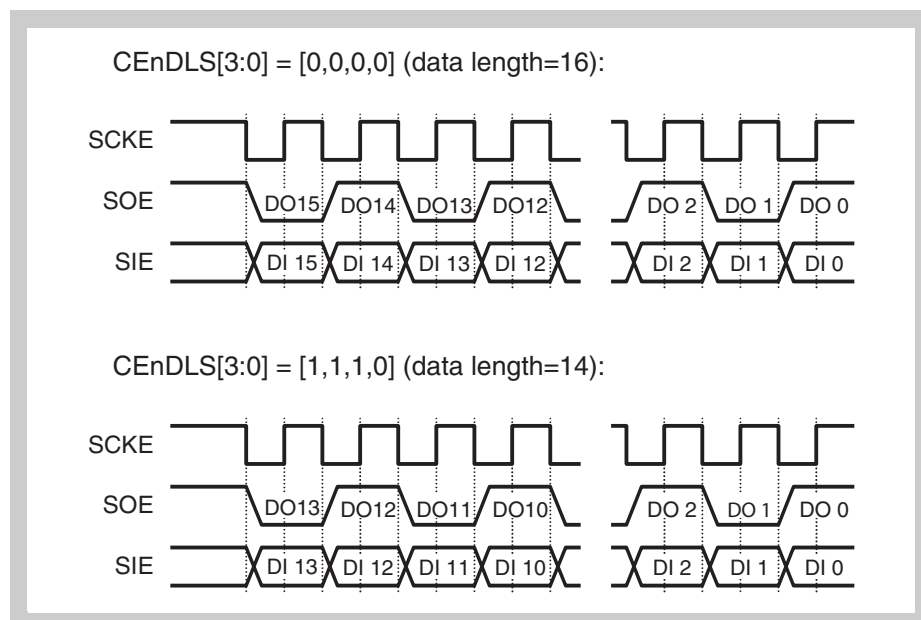


Figure 13-6 Data length select function

### 13.3.4 Slave mode

When the CEnCKS[2:0] bits in CEnCTL1 are set to [1,1,1], the Queued CSI operates in slave mode. In slave mode, the SCKE serial clock pin becomes input and another device is the CSI communication master. The baud rate generator "BRG" is recommended to be disabled by setting bits CEnMDL[2:0] to [0,0,0] when using slave mode. Also, the chip select pin SCSEn[3:0] outputs are not available in slave mode, as they are only available in master mode.

The example below shows the communication in slave mode for 8 data bits, CEnCKP=0, CEnDAP=0 and MSB first:

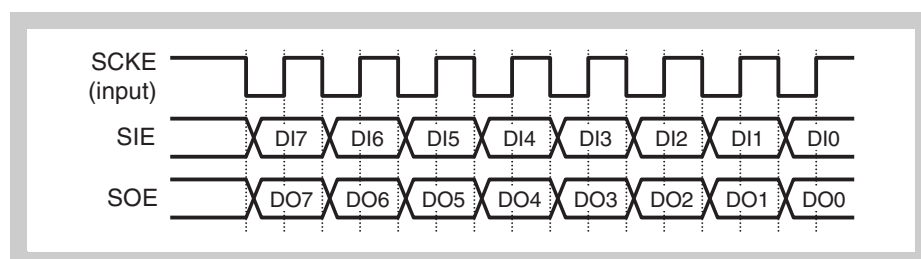


Figure 13-7 Slave mode

### 13.3.5 Master mode

When the CEnCKS[2:0] bits in CEnCTL1 are not set to [1,1,1], the Queued CSI operates in master mode. In master mode, the SCKE pin is configured as output and the serial communication clock is generated by the Queued CSI module. The SCKE pin's default value is "1" when CEnCKP = 1, and is default "0" when CEnCKP = 0. The SCSEn[3:0] pin outputs are available in master mode.

The example below shows the communication in master mode for 8 data bits, CEnCKP=0, CEnDAP=0 and MSB first:

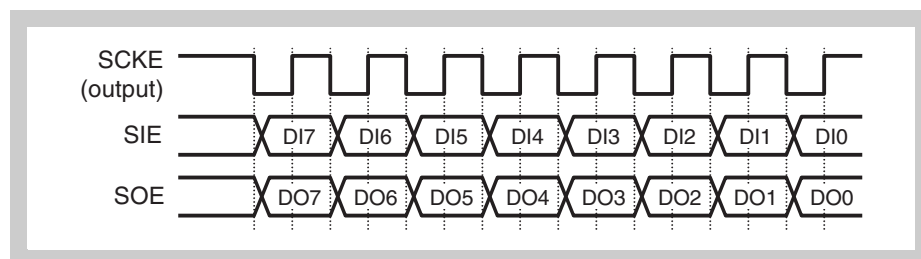


Figure 13-8 Master mode

### 13.3.6 Transmission clock select function

In Master Mode, the transfer baud rate is selectable using CEnCKS[2:0] bits and CEnMDL[2:0] bits in CEnCTL1 register. The baud rate generator "BRG" counts up at each rising edge of  $f_{QCSI}$ .

The example below illustrates the baud rate generation for CEnMDL[2:0] = [0,1,0].

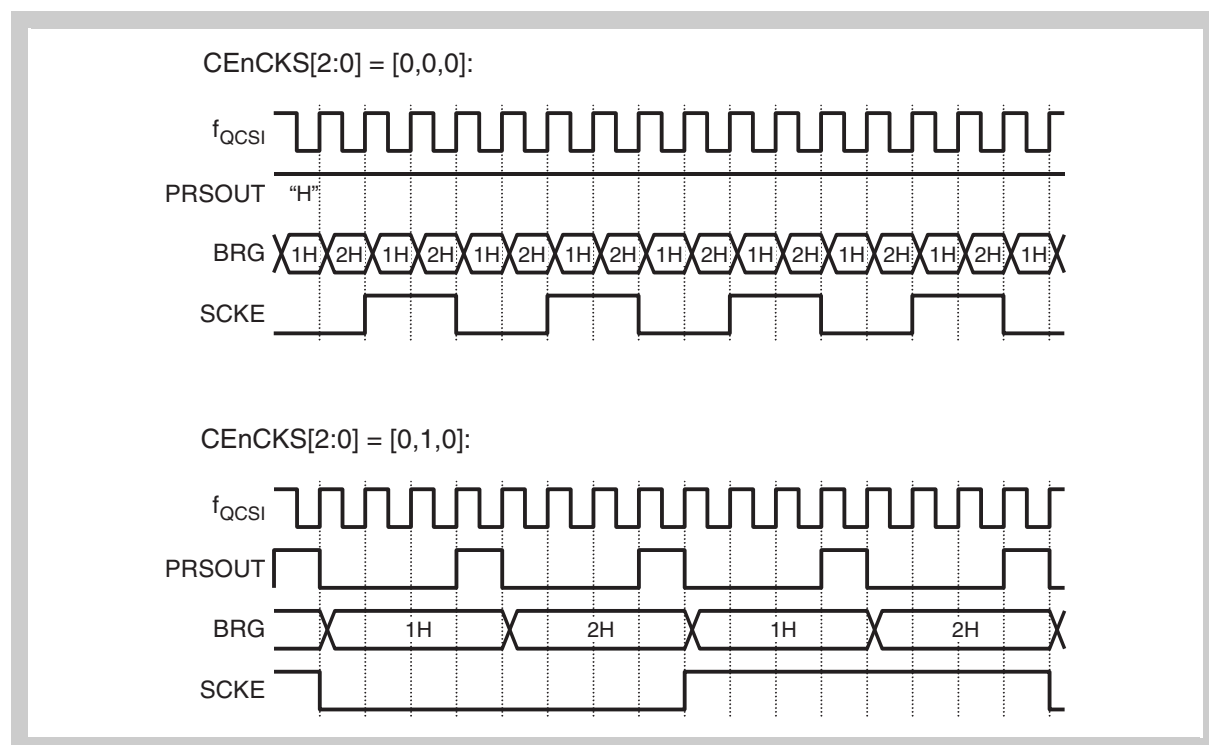
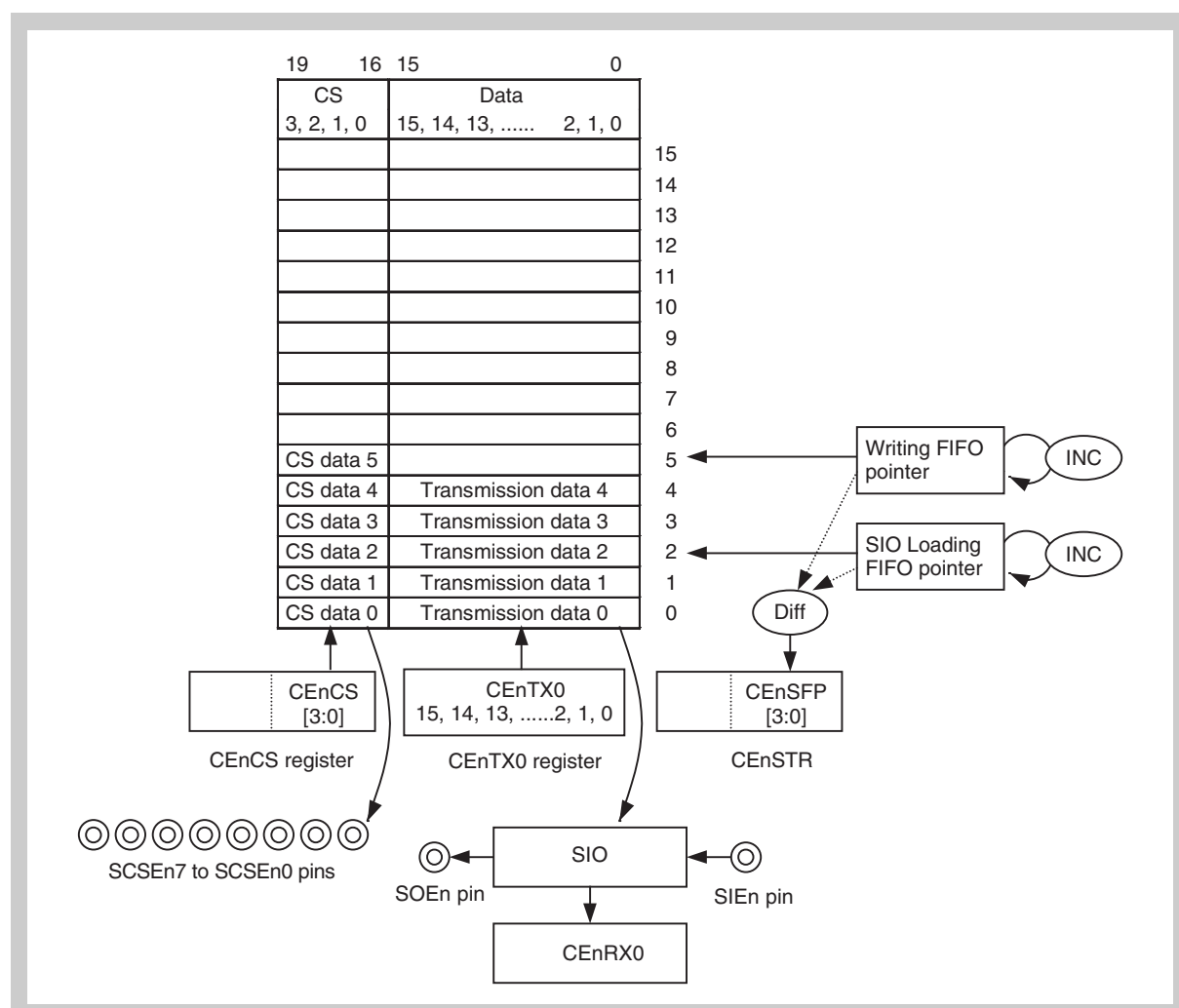


Figure 13-9 Transfer clock select function

### 13.3.7 Description of the single transfer mode

When the CEnTMS bit in the CEnCTL0 register is set "0", the Queued CSI operates in single transfer mode.



**Figure 13-10 Single transfer mode data handling**

Transfer start condition in single transfer mode:

[CEnTXE = 1 or CEnRXE = 1] and  
[Data exists in FIFO (CEnEMF = 0)]

A transfer starts once the transmission data (pointed to by the SIO Loading FIFO pointer) is transferred from the FIFO buffer to the serial shift register SIO. At that time, the transfer status flag CEnTSF turns to "1". The SCSEn[3:0] pins output the chip select data from the FIFO buffer.

At the end of the transfer:

1. If CEnRX0 is empty, the received data is stored from SIO to CEnRX0, and transfer end interrupt signal INTCEnC is generated (in transmit only mode, INTCEnC will be generated only when the FIFO buffer becomes empty). Finally, the SIO Loading FIFO pointer is incremented.
2. If CEnRX0 is not empty, the storing of receive data, INTCEnC generation and SIO Loading FIFO pointer incrementing wait for the CEnRX0 to be emptied by a software read operation.
3. In transmit only mode, if transmission data is available in the FIFO buffer,

the next transfer will start immediately, regardless of the CEnRX0 buffer condition.

When a transfer finishes and the FIFO buffer is empty (Writing FIFO pointer = SIO Loading FIFO pointer), CEnTSF is cleared "0".

CEnSFP[3:0] always show the current value of: (Writing FIFO pointer) - (SIO Loading FIFO pointer).

It is recommended to check that CEnFLF = 0 just before data is written to the CEnTX0 register. If CEnFLF = 1 when a write to CEnTX0 is attempted, the overflow interrupt INTCEnc is generated and the written data is ignored.

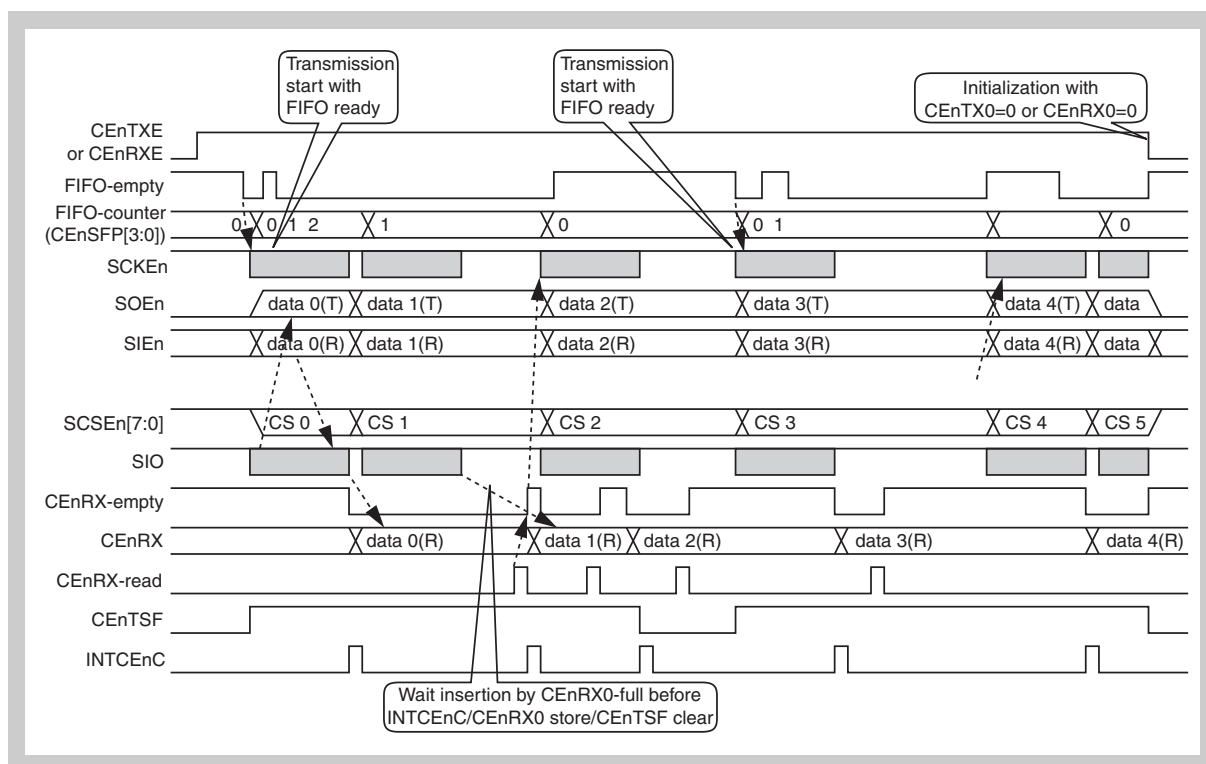
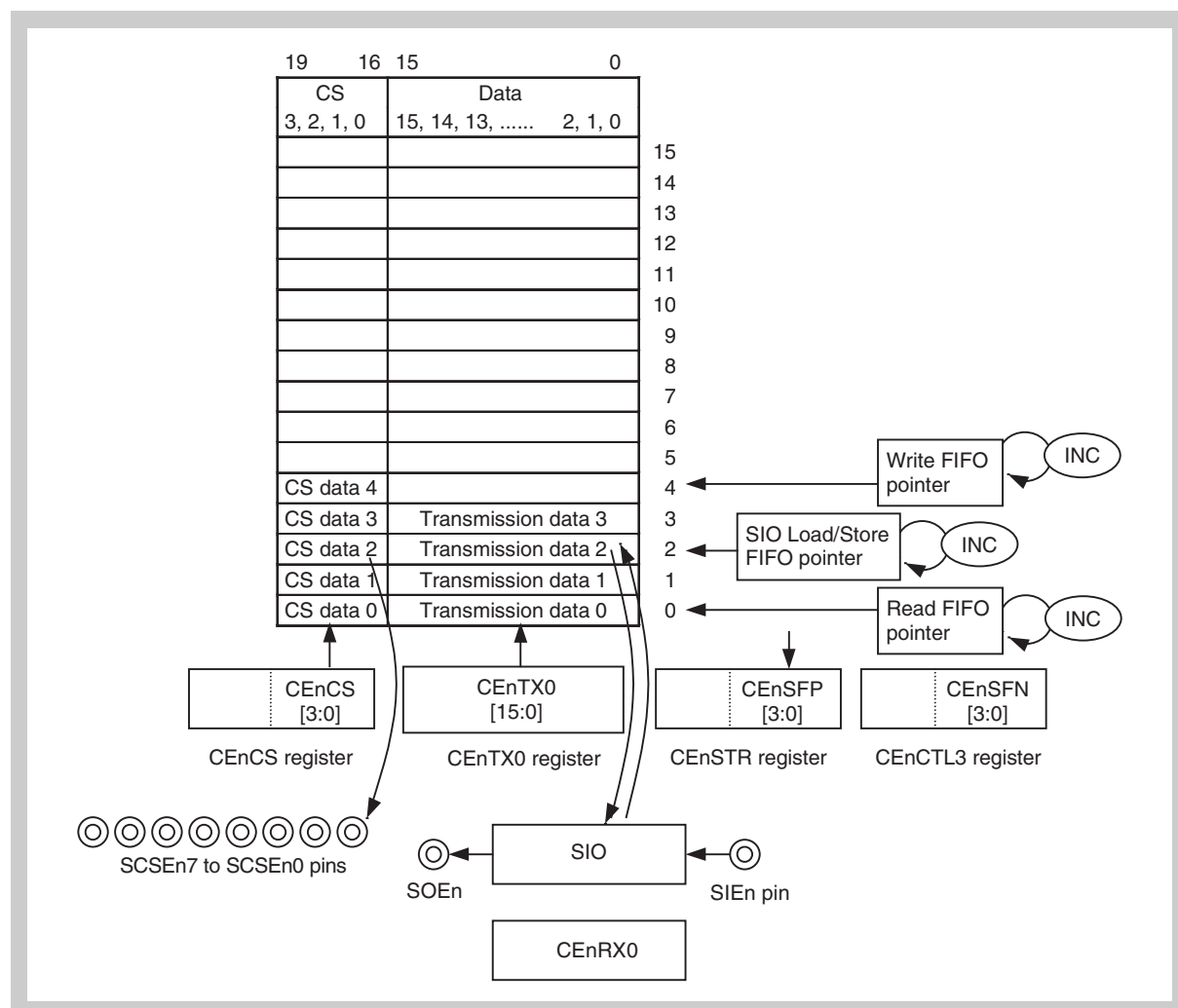


Figure 13-11 Single transfer mode (master, transmit/receive) timing



### 13.3.8 Description of the block transfer mode

When the CEnTMS bit in the CEnCTL0 register is set “1”, the Queued CSI operates in block transfer mode.



**Figure 13-12** Block transfer mode data handling

Transfer start condition in block transfer mode:

[CEnTXE = 1 or CEnRXE = 1] and  
[Data exists in FIFO (CEnEMF = 0)]

The transmission data number must be set in CEnSFN[3:0]. Note that writing a value greater than 16 to the CEnCTL3 register is prohibited, as the FIFO buffer design can hold up to 16 elements only.

The transfer starts by copying the first data element - pointed to by SIO Load/Store FIFO pointer - to the SIO shift register. At that time the transmission status flag CEnTSF is set to “1”, and the SCSEn[3:0] pins output the CS value from the FIFO.

When the transfer of the data element is finished, the received data overwrites the location in the FIFO using the SIO Load/Store FIFO pointer, and the SIO Load/Store FIFO pointer is then incremented.

When the transmission/reception counter reaches the value set by CEnSFN[3:0], then CEnTSF is cleared “0” and the transmission/reception end interrupt signal INTCEnC is generated.

After the interrupt occurred, the received data can be read from CEnRX0. The Read FIFO pointer is automatically incremented by the CEnRX0 read operation.

All FIFO pointers must be cleared by setting CEnPCT = 1 before the next transmit/receive cycle can start.

SFP[3:0] represents the [SIO Load/Store FIFO pointer] and shows the number of transmission/receptions completed. In case of SFP[3:0]=0H, the numbers of transmissions/receptions depends on the setting of the SFEMP bit:

SFEMP=0: 0 transmissions/receptions completed

SFEMP=1: 16 transmissions/receptions completed

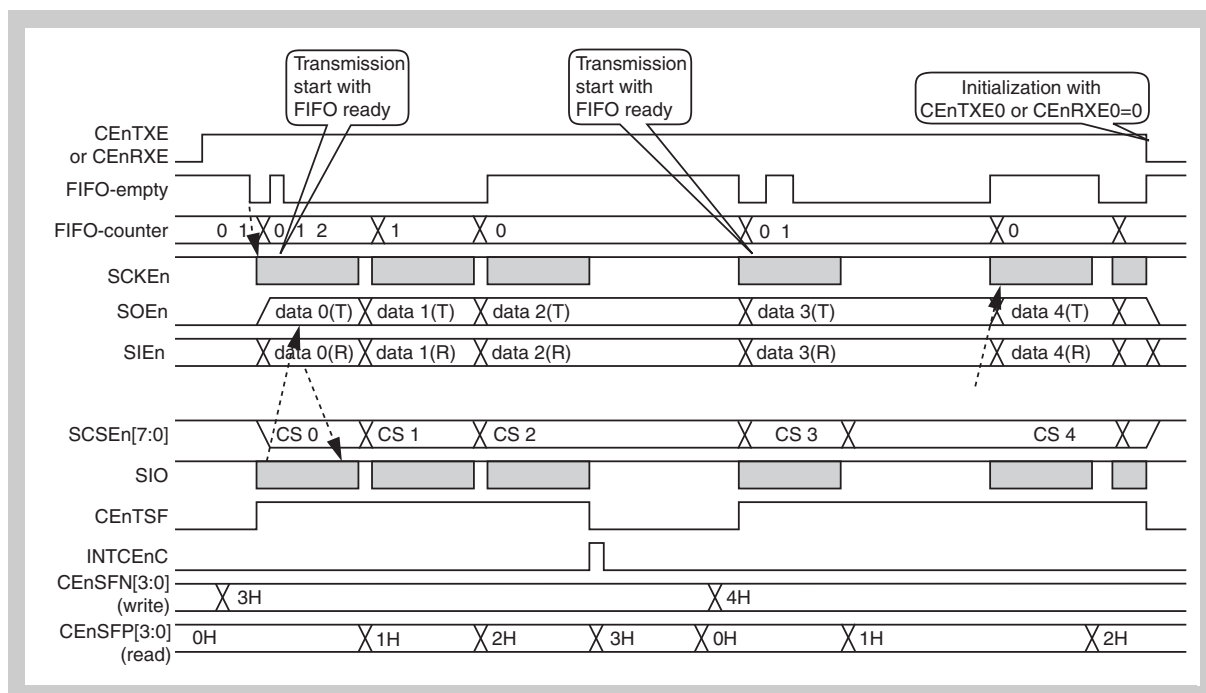


Figure 13-13 Block transfer mode (master. transmit/receive) timing

### 13.3.9 Description of the operation modes

#### (1) Transmit only mode

Setting the CEnCTL0 register's CEnTXE = 1 and CEnRXE = 0 places the Queued CSI in transmit only mode. A transmission starts when transmit data is written in the CEnTX0 register. The current condition of the CEnRX0 buffer and SIO register no effect. The data in the CEnRX0 and the SIO buffer is undefined after completion of the transmission.

#### (2) Receive only mode

Setting the CEnCTL0 register's CEnTXE = 0 and CEnRXE = 1 places the Queued CSI in receive only mode. A reception starts when dummy data is written in the CEnTX0 register. It is mandatory, though, that the CEnRX0 and SIO are empty. If a receive operation is terminated while the previous receive data remains unread in CEnRX0, the Queued CSI is placed on wait status until the previous data is completely read and CEnRX0 becomes empty.

**(3) Transmit/receive mode**

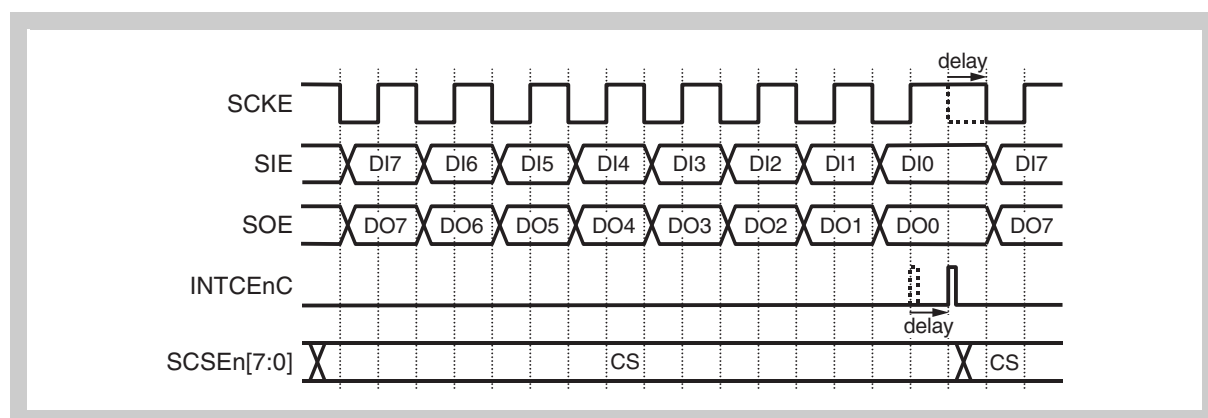
Setting the CEnCTL0 register's CEnTXE = 1 and CEnRXE = 1 places the Queued CSI in transmit/receive mode. A transfer (meaning transmission and reception) starts when transmit data is written in the CEnTX0 register. Note that an empty CEnRX0 or SIO is mandatory. If a receive operation is terminated while the previous receive data remains unread in CEnRX0, the Queued CSI is placed on wait status until the previous data is completely read and CEnRX0 becomes empty.

In block transfer mode with slave mode, only the first dummy data write operation is required. There is no need to write chip-select data, as these bits are ignored.

**13.3.10 Additional timing and delay selections****(1) Delay selection of receive termination interrupt signal (INTCEnC)**

In master mode, the CEnSIT bit of the CEnCTL0 register can be used to delay the generation of the receive termination interrupt signal (INTCEnC) by a half serial clock cycle (SCKE). The CEnSIT bit takes effect only in the master mode and is ignored in slave mode.

Figure 13-14 below illustrates the CEnSIT function, assuming a setting of CEnSIT=1, CEnWE=0, CEnCKP=0, CEnDAP=0 and CEnDL[3:0] = [1,0,0,0].



**Figure 13-14 Delay selection of receive termination interrupt (INTCEnC)**

**(2) Selection of transmit wait enable/disable**

In master mode, while the enhanced timing is not enabled, the CEnCTL0 register's CEnWE bit setting can be used to delay the start of transmission by one SCKE clock cycle. The CEnWE bit takes effect only in the master mode and is ignored in slave mode.

Figure 13-15 below illustrates the CEnWE function, assuming a setting of CEnWE=1, CEnCSM=0, CEnOPE=0, CEnCKP=0, CEnDAP=0 and CEnDL[3:0] = [1,0,0,0].

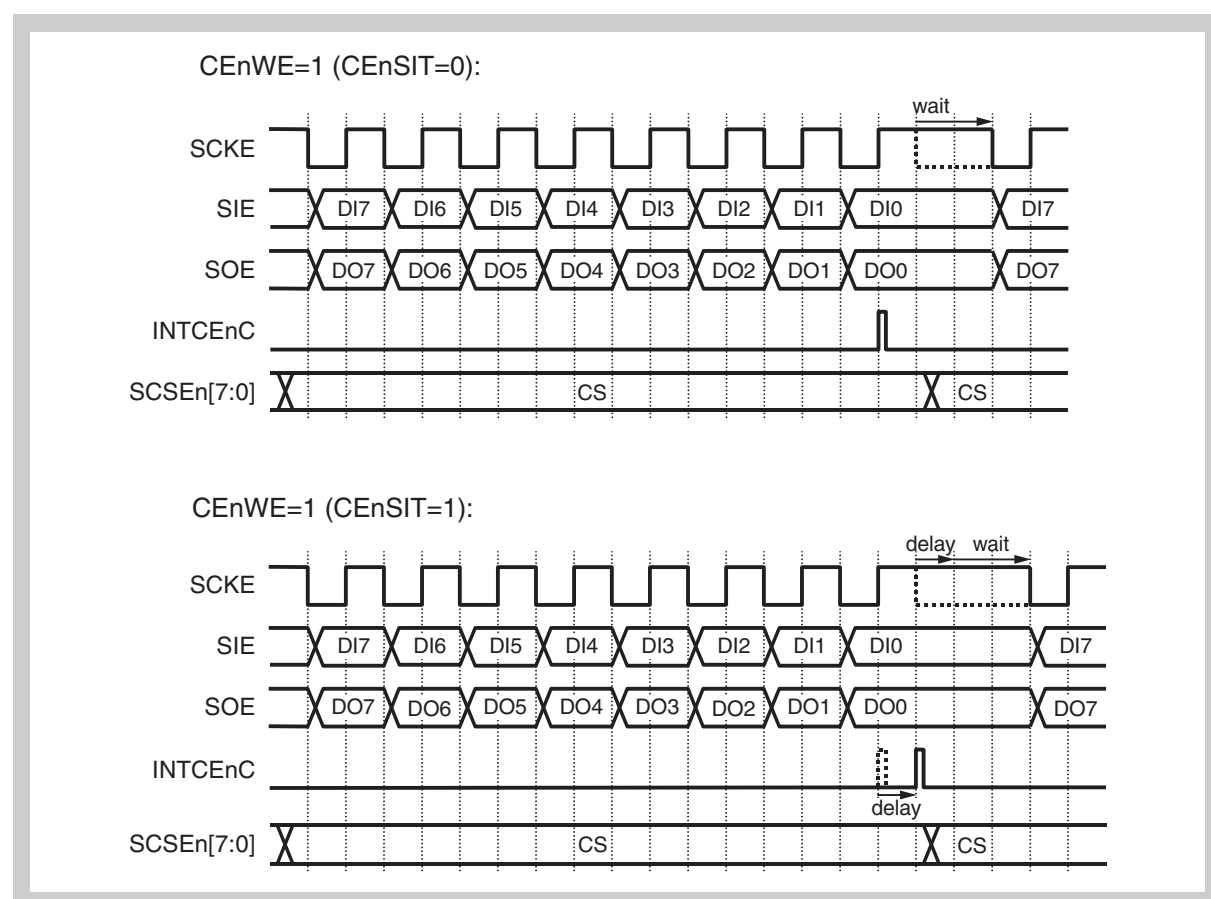


Figure 13-15 Selection of transmit wait enable/disable

**(3) Selection of chip-select mode**

In master mode with  $CEnWE = 1$  and  $CEnOPE=0$ , the  $CEnCSM$  bit setting can be used to output an inactive level at the chip select pins  $SCSEn[3:0]$  during the delay between data transmissions. The  $CEnCSM$  bit takes effect only in the master mode and is ignored in slave mode.

The  $CEnCSM$  bit has no effect while  $CEnWE$  is set to 0.

Figure 13-16 below illustrates the  $CEnCSM$  function, assuming a setting of  $CEnCKP=0$ ,  $CEnDAP=0$  and  $CEnDL[3:0] = [1,0,0,0]$ .

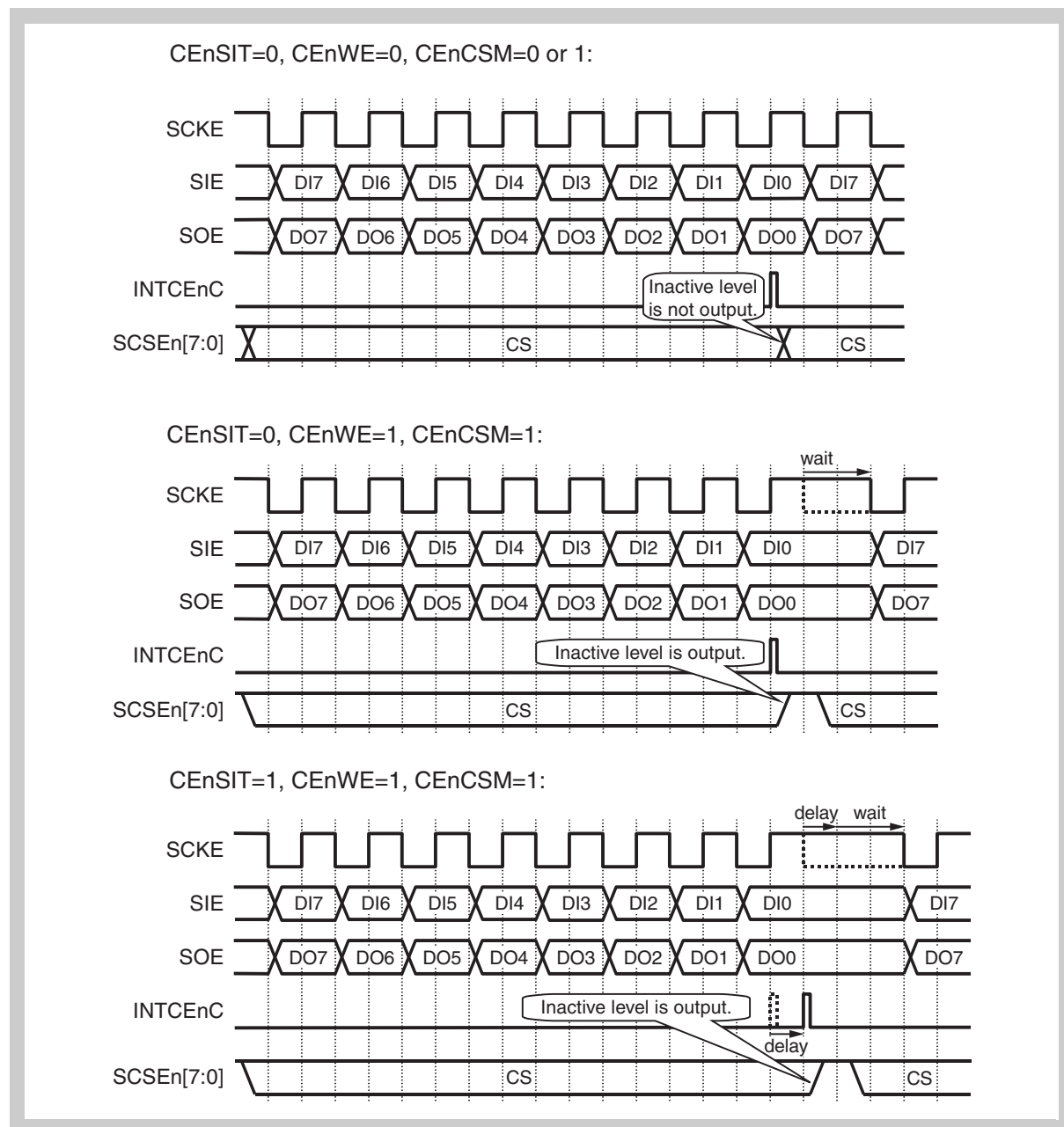


Figure 13-16 Selection of chip-select mode

### 13.3.11 Default pin levels

#### (1) SCKE pin's default level

The SCKE pin's default level with the CEnCTL0 register settings CEnPWR=0 or (CEnTXE=0 and CEnRXE = 0) is as follows:

CEnCKP	CEnCKS[2:0]	SCKE default level
0	1, 1, 1 (slave mode)	1
	Other than 1, 1, 1 (master mode)	1
1	1, 1, 1 (slave mode)	1
	Other than 1, 1, 1 (master mode)	0

← Initialization after reset

**Note** In slave mode, the SCKE pin is always set to "1".

#### (2) SOE pin's default level

The SOE pin's default level with the CEnCTL0 register settings CEnPWR=0 or (CEnTXE=0 and CEnRXE=0) is as follows:

SOE pin's default level
0

← Initialization after reset

#### (3) SCSEn0 to SCSEn7 pins' default level

The SCSEn0 to SCSEn7 pins' default level with the CEnCTL0 register settings CEnPWR=0 or (CEnTXE=0 and CEnRXE=0) is as follows:

SCSEn0 to SCSEn7 pins' default level
inactive

← Initialization after reset

### 13.3.12 Transmit buffer overflow interrupt signal (INTCEnOF)

When the transmit FIFO buffer contains 16 elements, writing a 17th chip-select data (CEnCS write) or transfer data (CEnTX0 write) results in the generation of the overflow interrupt INTCEnOF. For the 17th item, both chip-select and transfer data values are discarded.

The transmit FIFO buffer contains 16 elements if the FIFO pointer value for the write operation equals the FIFO pointer value for the SIO load operation plus 15. When the transfer is completed and the FIFO buffer pointer for the SIO load operation is incremented, space for one element is available again in transmit buffer.

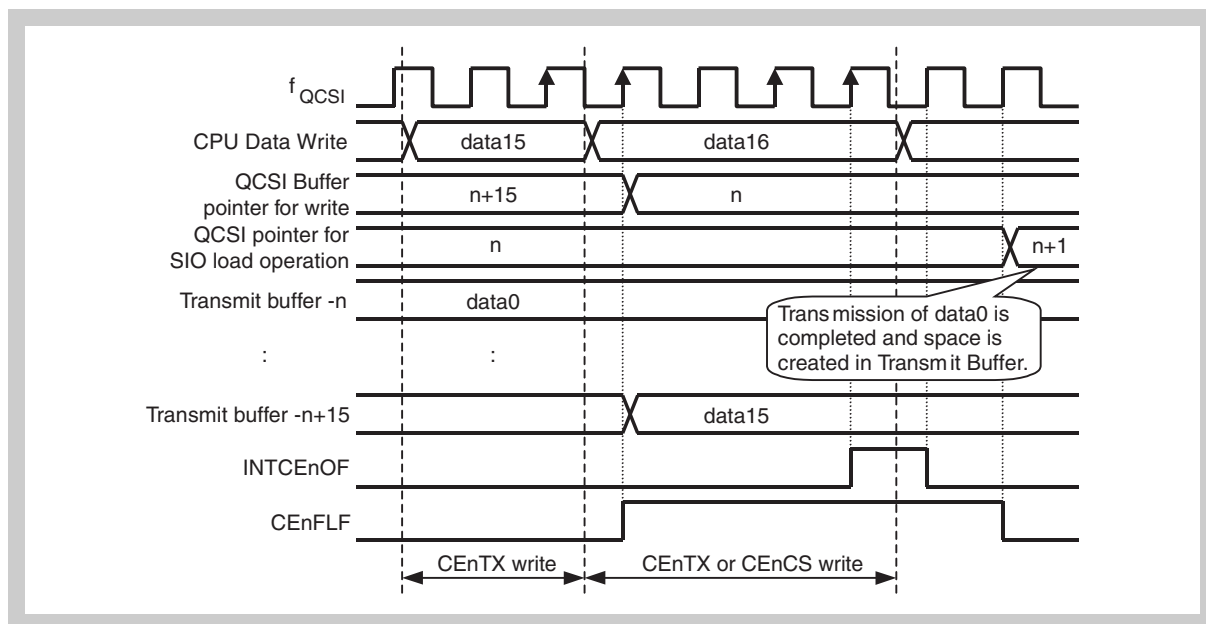


Figure 13-17 Transmit buffer overflow interrupt signal (INTCEnOF)

### 13.3.13 Enhanced chip select timing

#### (1) General

The Enhanced Queued-CSI CSIE allows specific timing settings for each chip select line. The figure below shows the general timings of the chip select lines:

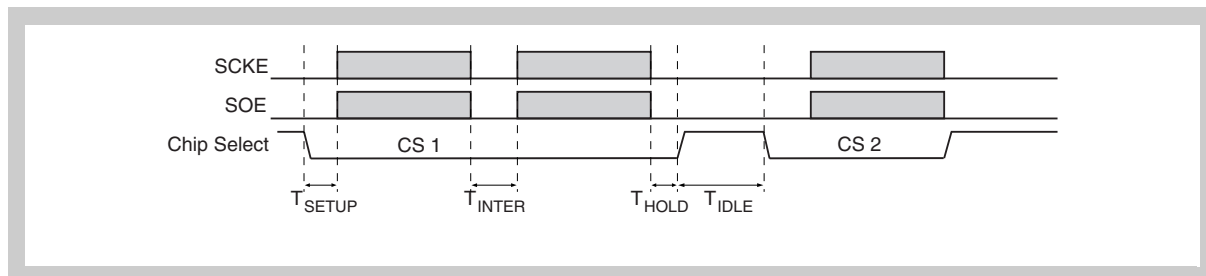


Figure 13-18 General Overview on Enhanced Chip Select Timing

$T_{\text{SETUP}}$  is the time between the chip select line becoming active and the first data bit to be transferred.  $T_{\text{INTER}}$  defines the delay between two transfers while the same chip select is chosen and active. After the last transfer, the chip select line remains active for a period set by  $T_{\text{HOLD}}$ . Last not least,  $T_{\text{IDLE}}$  can be used to delay the start of a new transfer with a different chip select setting.

Each delay time is set relative to the serial communication clock SCKE and can range from 0 (no delay) up to 8 serial clocks.

#### (2) Chip select deactivation

A chip select line is selected by writing the corresponding value to the CEnCS[3:0] bits. The chip select line remains active until a value is written to the CEnCS[3:0] bits that is different from the previous value.

The chip select line remains active even when the transfer queue becomes empty. After a new element is written in the transfer queue, either the inter-data or the hold timing will be applied before the new element is transmitted. In other words, the queue idle time will be seen in addition to the defined chip select timing.

#### (3) Multiple chip select activation

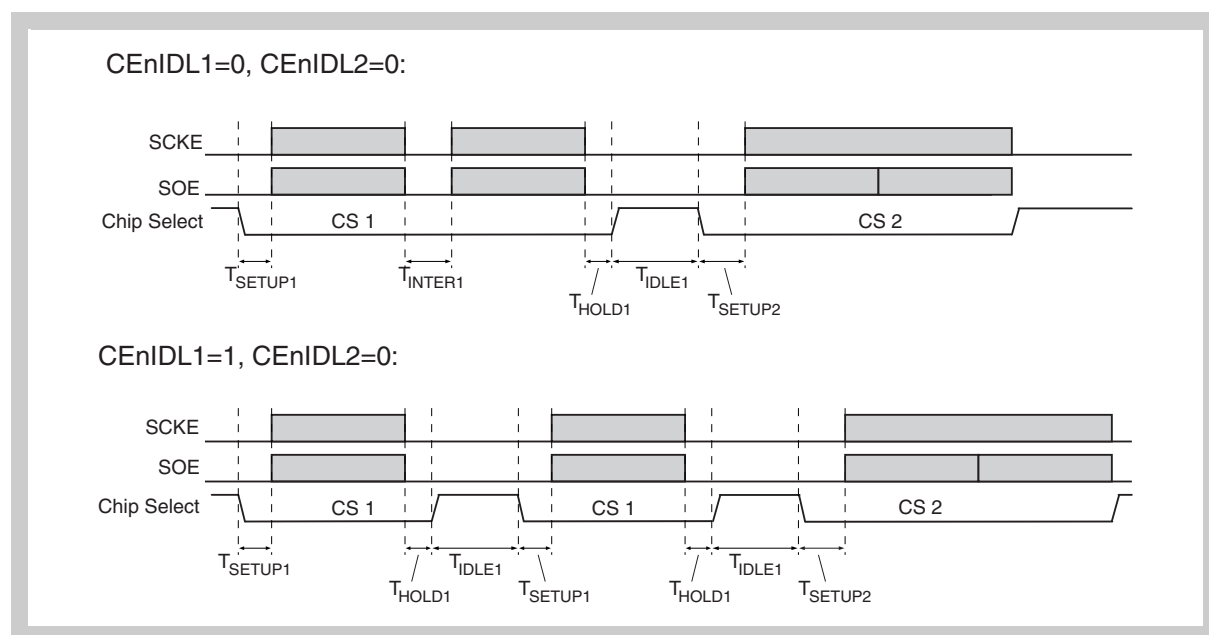
In 4 chip select mode, multiple chip selects can be active at the same time. When enhanced timing is enabled, the Queued-CSI macro will use the timing of the enabled chip select with the lowest number. But to ensure proper communication, it is mandatory that the timings of all chip selects that can be active at a same time are set to the same values.

#### (4) Enforced chip select idle setting

The chip select line is hold active until the chip select setting of a transfer element changes. While this is suitable for most peripheral devices connected to the serial interface, some devices require an inactive state of the chip select line after each element transmitted.

The “Force Chip Select Idle” (CEnIDLm) bit is used to force this behaviour. Figure 13-19 illustrates the difference between a standard communication and a communication with the CEnIDL1 bit set. It is assumed that the delay timings of CS2 are set as  $T_{\text{INTER}2}=0$  and  $T_{\text{HOLD}2}=0$ , and that two elements are transferred to CS1 one before a transfer of two elements to CS2.



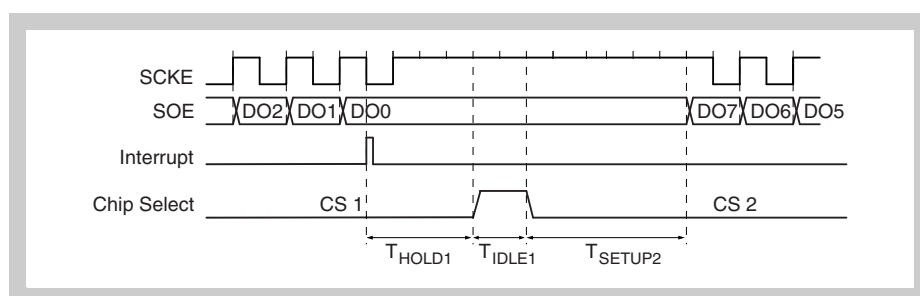


**Figure 13-19 Enforced chip select idle example**

While the timing for CEnIDL1=0 uses  $T_{INTER1}$  as between the two elements for CS1, the setting of CEnIDL1=1 forces the  $T_{HOLD1}$ ,  $T_{IDLE1}$  and  $T_{SETUP1}$  delays for the chip select line between the elements.

#### (5) Hold, idle and setup timing relative to $\overline{SCKE}$

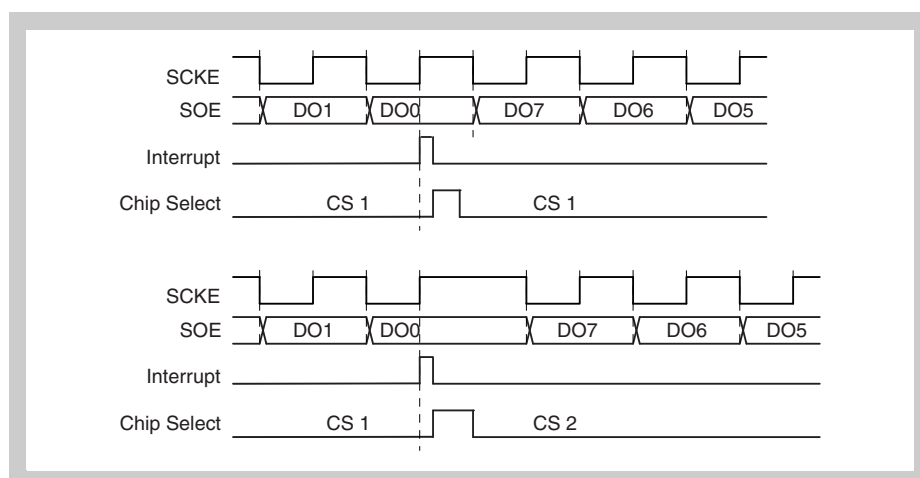
The start of any timing is at the last valid data signal of the SCKE, not at the end of the SCKE cycle. *Figure 13-20* illustrates the timing. It is assumed that  $CKP=0$ ,  $DAP=1$ ,  $T_{HOLD1}=2$  SCKE,  $T_{IDLE1}=1$  SCKE,  $T_{SETUP2}=3$  SCKE, and that the data bits per element is 8:



**Figure 13-20 Enhanced chip select timing extension relative to SCKE**

With the setting of  $CKP=0$  and  $DAP=1$ , data is output on the rising edge of SCKE and the data is valid at the falling edge. When the last data bit is sent, the interrupt is generated at the falling edge of the SCKE.

The next figure shows the timing with  $T_{\text{HOLD}1}=T_{\text{IDLE}1}=T_{\text{SETUP}2}=0$ , but  $\text{CEnIDL1}=1$ ,  $\text{CKP/DAP}=0/0$ :



A small chip select idle pulse is generated after the “data valid” edge of SCKE, which is the rising edge in this example due to  $\text{CKP}=\text{DAP}=0$ .

#### (6) Impact of interrupt delay on enhanced timing

The  $\text{CEnSIT}$  bit allows to set a delay of half serial communication clock before the transfer complete interrupt is generated. This delay is inserted before any delay defined by the enhanced timing.

Figure 13-21 shows as example a communication with  $T_{\text{INTER}1}=0$ ,  $T_{\text{HOLD}1}=0.5$ ,  $T_{\text{IDLE}1}=1$ ,  $T_{\text{SETUP}2}=0$  and  $T_{\text{INTER}2}=1$ , using single buffer transfer mode and 8-bit data length.

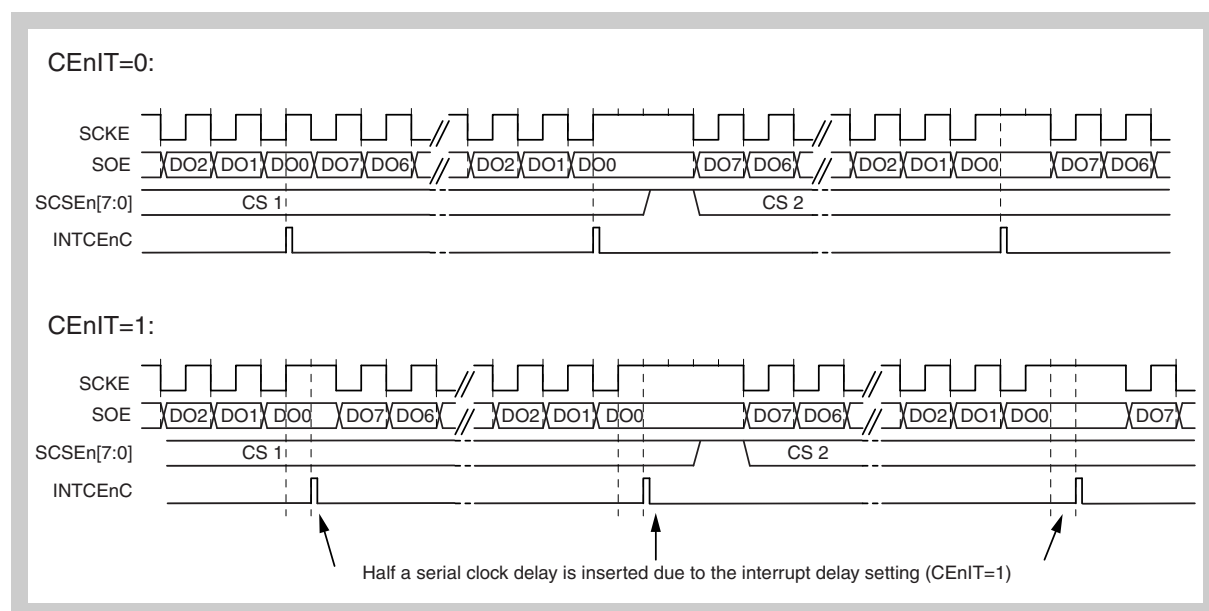


Figure 13-21 Enforced chip select idle timing extension

#### (7) Data phase details (dap=1) with enhanced timing

The  $\text{DAP}$  bit defines the phase of the data line relative to the serial clock. While the timing diagram for  $\text{DAP}=0$  is straight forward, for  $\text{DAP}=1$  there would be two options for the rising edge of the serial clock signal at the last data bit ( $\text{DO}0$ ) when hold time is applied. Please use the figure below as reference.

The rising edge could be either done at the end of the DO0 bit, or it could be done at the end of the hold time (dashed line in the timing diagram). The Queued-CSI module places the rising edge at the end of the data bit.

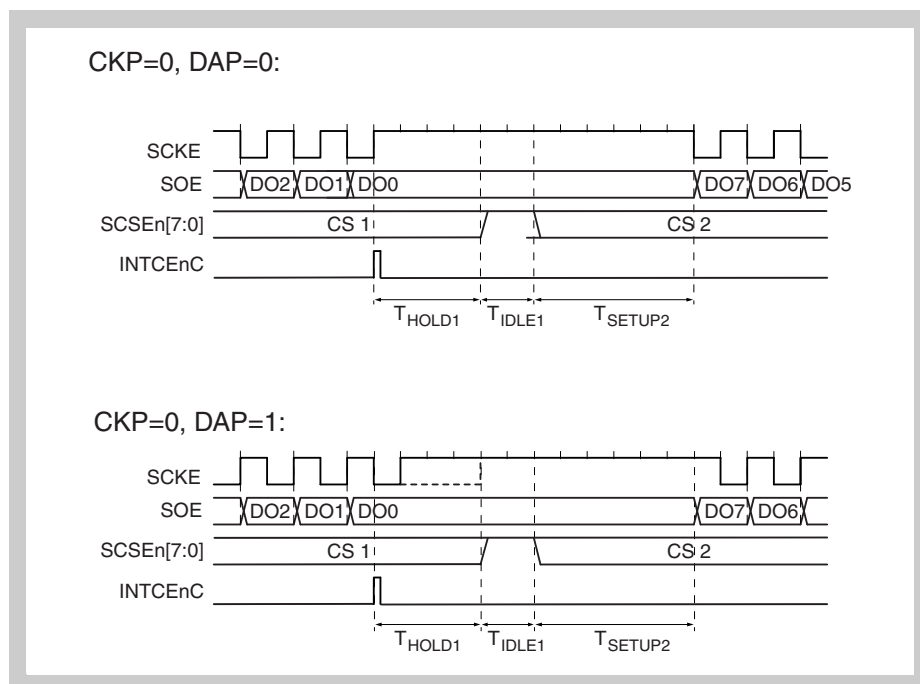


Figure 13-22 Data phase details (DAP=1) with enhanced timing

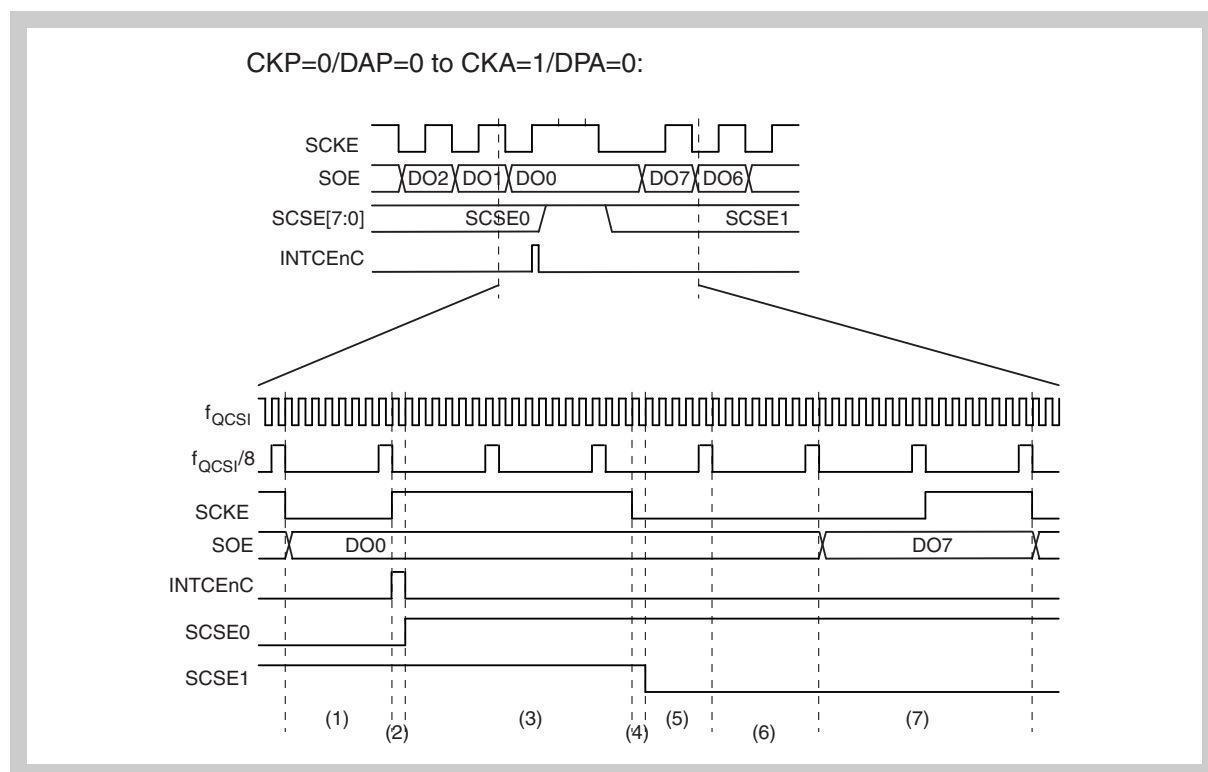
#### (8) Change of clock phase and data phase

The enhanced timing also allows the definition of two clock polarity/data phase pairs. The following figures (Figure 13-23 to Figure 13-25) illustrate the transition in polarity down to internal Queued-CSI operating clock level. In a nutshell, there is one macro clock delay between the change in polarity and the activation of the chip select signal.

Except when noted otherwise, it is assumed in the following examples that there is no setup delay defined for CS2. If a delay is defined, the standard rules would apply after the activation of the chip select signal.

**Caution** Figure 13-23 shows the synchronisation delay after the new CS signal is activated, while in the other figures the synchronisation delay is ignored to simplify the drawings. The synchronization delay would apply, though!

Figure 13-23 shows the transition from SCSE0 with (CKP=0, DAP=0) to SCSE1 with (CKA=1, DPA=0), using the following settings:  $T_{HOLD0}=0$ ,  $T_{IDLE0}=1$ ,  $T_{SETUP1}=0.5$ ,  $f_{SCKE}=f_{QCSI}/16$ :



**Figure 13-23 CKP=0, DAP=0 to CKA=1, DPA=0 with enhanced timing**

1. Is the last first half of the last data bit sent with SCSE0 active
2. At the rising edge of SCKE, the interrupt signal is generated.
3. Represents the idle time ( $T_{IDLE0}=1$ ). One  $f_{QCSI}$  clock after the generation of the interrupt signal, SCSE0 is set to inactive. It is hold inactive for serial clock cycle plus one internal clock  $f_{QCSI}$ .
4. After the idle time, the serial clock polarity is changed due to the CKA=1 transition.
5. One internal clock  $f_{QCSI}$  after the change of the clock polarity SCSE1 is set to active. Then the timing is synchronized with the internal baud rate generator that operates continuously at  $f_{QCSI}/8$ .
6. Is the setup delay of SCSE1 ( $T_{SETUP1}=0.5$ ).
7. Is the output of the first data bit with SCSEn1 active.

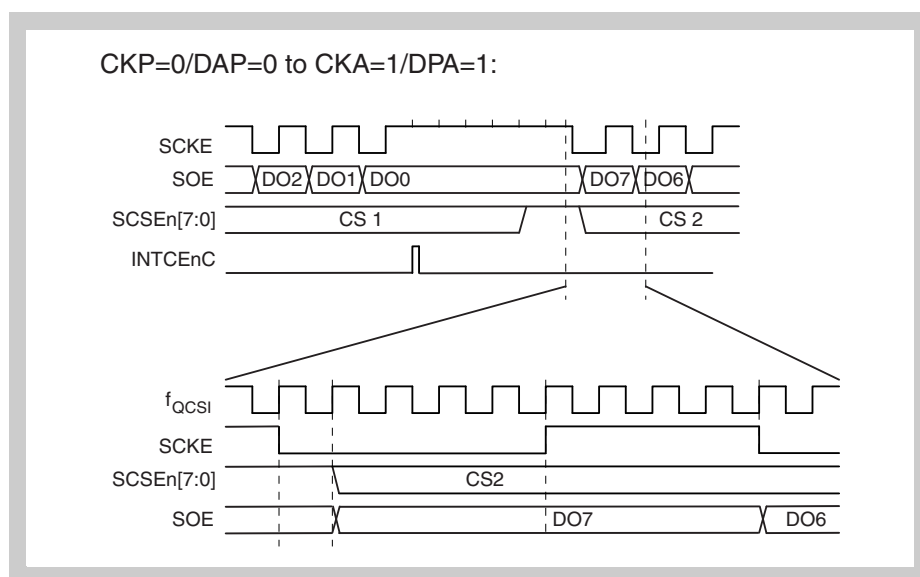


Figure 13-24 CKP=0, DAP=0 to CKA=1, DPA=1 with enhanced timing

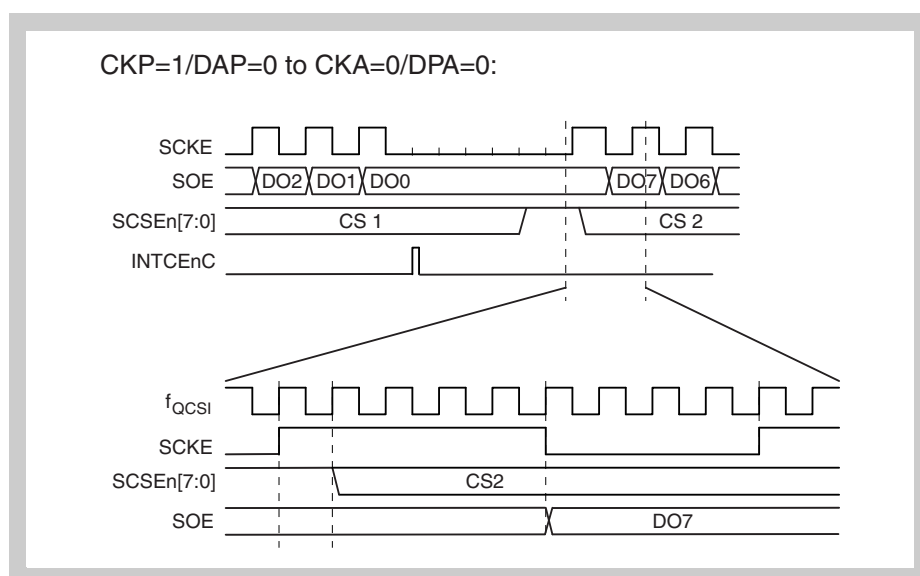


Figure 13-25 CKP=1, DAP=0 to CKA=0, DPA=1 with enhanced timing

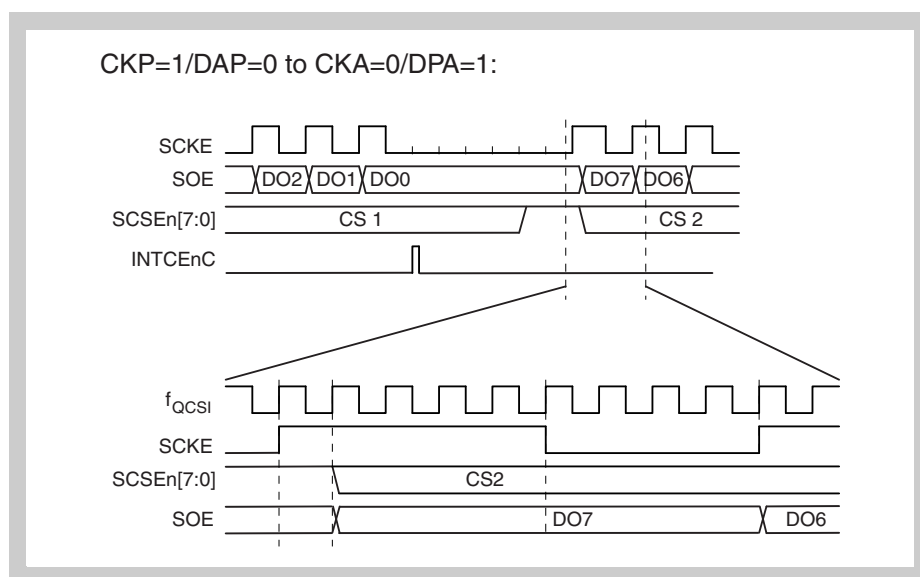


Figure 13-26 CKP=1, DAP=0 to CKA=0, DPA=0 with enhanced timing

## 13.4 Operating Procedure

### 13.4.1 Single transfer mode (master mode, transmit only mode)

MSB first (CEnDIR = 0), no INTCEnC delay (CEnSIT=0), transmission wait disabled (CEnWE = 0),

CS inactive disabled (CEnCSM = 0), CEnCKP = 0, CEnDAP = 0, transmission data length of 8 bits (CEnDL[3:0] = [1,0,0,0]):

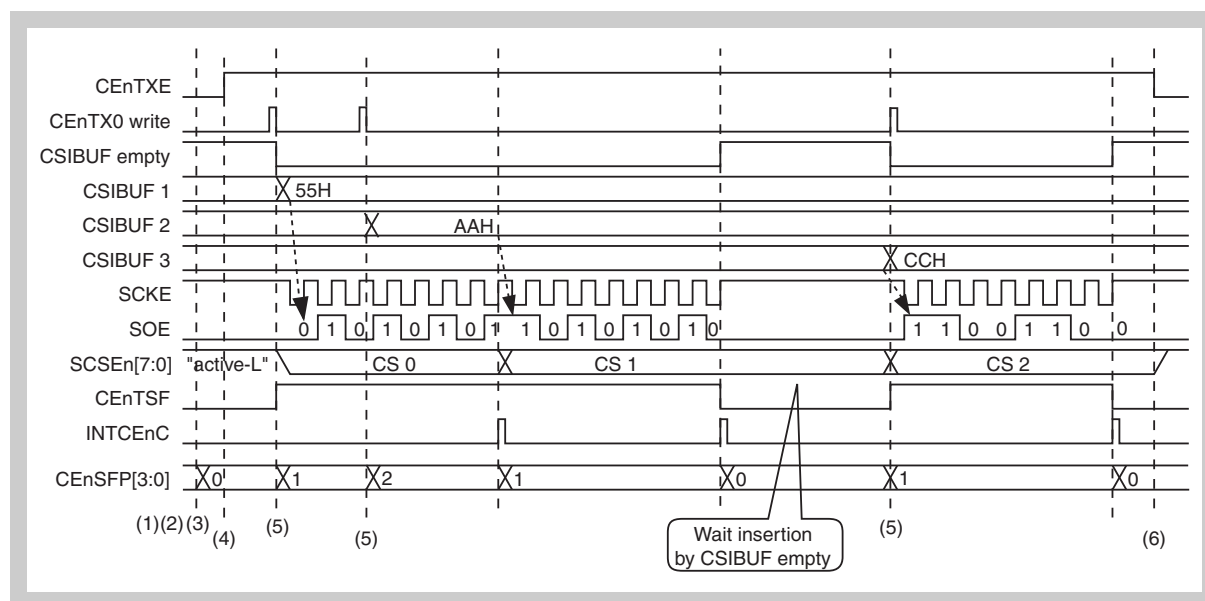


Figure 13-27 Single transfer mode (master, transmit only) timing

1. Set the CEnCTL0 register's CEnPWR bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CEnCTL1 and CEnCTL2 registers to specify the transfer mode.
3. Write "1" in the CEnSTR register's CEnPCT bit to clear all FIFO pointers.
4. Specify the transfer mode using the CEnCTL0 register's CEnTMS, CEnDIR, and CEnSIT bits; at the same time, set the CEnTXE bit to 1 to enable transmission.
5. Make sure that the CEnSTR register's CEnFLF bit is set to 0, then write chip-select data and transmission data in the CEnCS and CEnTX0 registers in this order.

Repeat step (5) until the last element to be transmitted is written in the CEnCS/CEnTX0 registers.

6. Verify that CEnSTR.CEnTSF = 0 (CSIE in idle state) and set the CEnCTL0 register's CEnTXE bit to 0 to disable transmission (end of transmission).

### 13.4.2 Single transfer mode (master mode, receive only mode)

MSB first (CEnDIR = 0), no INTCEnC delay (CEnSIT = 0), transmission wait disabled (CEnWE = 0), CS inactive disabled (CEnCSM = 0), CEnCKP = 0, CEnDAP = 1, transmission data length of 8 bits (CEnDL[3:0] = [1,0,0,0]):

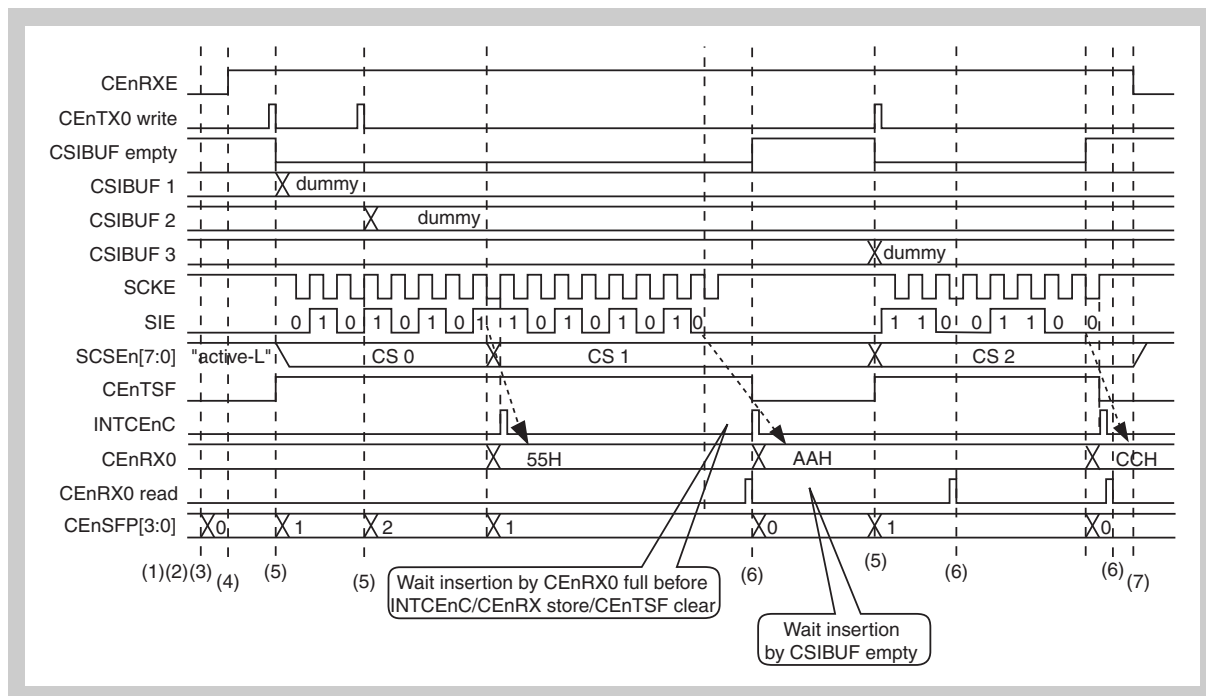


Figure 13-28 Single transfer mode (master, receive only) timing

1. Set the CEnCTL0 register's CEnPWR bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CEnCTL1 and CEnCTL2 registers to specify the transfer mode.
3. Write "1" in the CEnSTR register's CEnPCT bit to clear all FIFO pointers.
4. Specify the transfer mode using the CEnCTL0 register's CEnTMS, CEnDIR, and CEnSIT bits; at the same time, set the CEnRXE bit to 1 to enable the receive operation.
5. Make sure that the CEnSTR register's CEnFLF bit is set to 0, then write chip-select data and dummy transmission data in the CEnCS and CEnTX0 registers in this order (start-of-receive trigger).
6. Check for a reception to be completed by monitoring the INTCEnC interrupt. If so, read the CEnRX0 register.

Repeat steps (5) and (6) until the last element is received and read from the CEnRX0 register.

7. Verify that CEnSTR.CEnTSF = 0 (CSIE in idle state) and set the CEnCTL0 register's CEnRXE bit to 0 to disable the receive operation (end of receive operation).

**Note** The SOE pin is invalid and maintains its signal level, as the output latch is disabled.



### 13.4.3 Single transfer mode (master mode, transmit/receive mode)

MSB first (CEnDIR = 0), no INTCEnC delay (CEnSIT = 0), transmission wait disabled (CEnWE = 0), CS inactive disabled (CEnCSM = 0), CEnCKP = 1, CEnDAP = 0, transmission data length of 8 bits (CEnDL[3:0] = [1,0,0,0]):

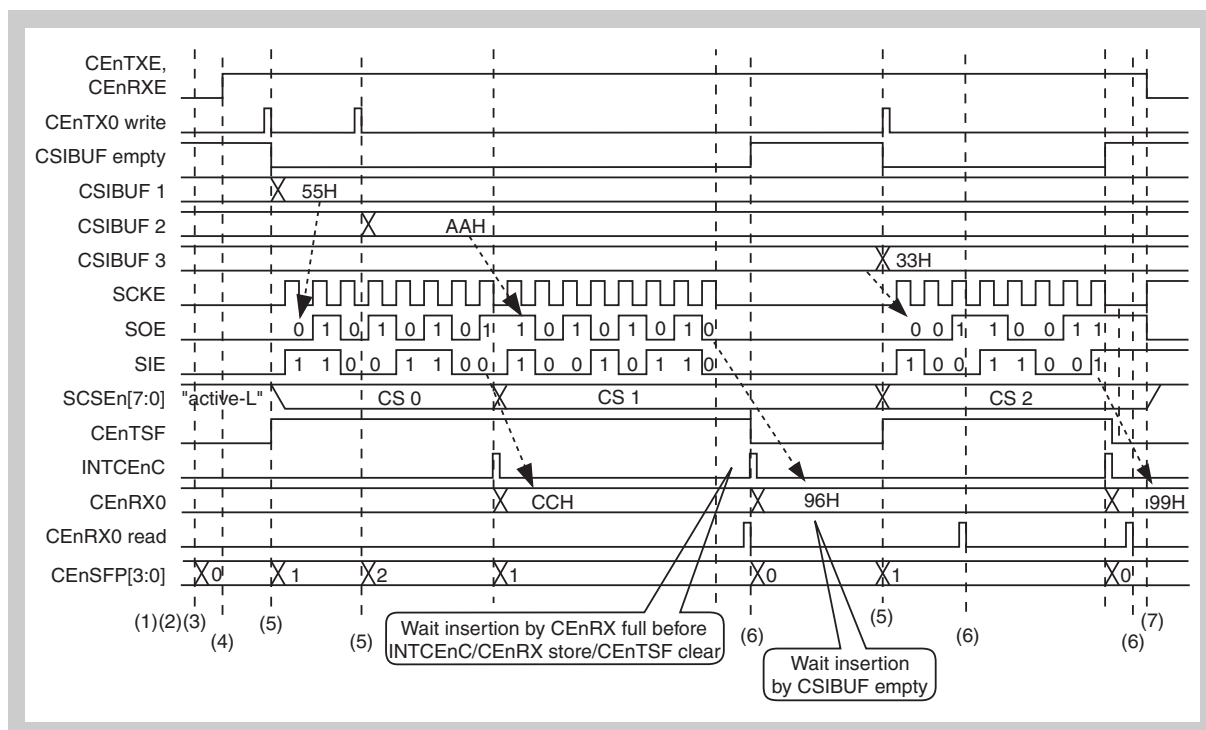


Figure 13-29 Single transfer mode (master, transmit/receive) timing

1. Set the CEnCTL0 register's CEnPWR bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CEnCTL1 and CEnCTL2 registers to specify the transfer mode.
3. Write "1" in the CEnSTR register's CEnPCT bit to clear all FIFO pointers.
4. Specify the transfer mode using the CEnCTL0 register's CEnTMS, CEnDIR, and CEnSIT bits; at the same time, set the CEnTXE and CEnRXE bits to 1 to enable the transmit/receive operation.
5. Make sure that the CEnSTR register's CEnFLF bit is set to 0, then write chip-select data and transmission data in the CEnCS and CEnTX0 registers in this order.
6. Check for a transmission to be finished by monitoring the INTCEnC interrupt. If so, read the CEnRX0 register.

Repeat steps (5) and (6) until the last element is send/received and read from the CEnRX0 register.

7. Verify that CEnSTR.CEnTSF = 0 (CSIE in idle state) and set the CEnCTL0 register's CEnTXE and CEnRXE bits to 0 to disable the transmit/receive operation (end of transmit/receive operation).

### 13.4.4 Single transfer mode (slave mode, transmit only mode)

MSB first (CEnDIR = 0), no INTCEnC delay (CEnSIT = 0), transmission wait disabled (CEnWE = 0), CS inactive disabled (CEnCSM = 0), CEnCKP = 1, CEnDAP = 1, transmission data length of 8 bits (CEnDL[3:0] = [1,0,0,0]):

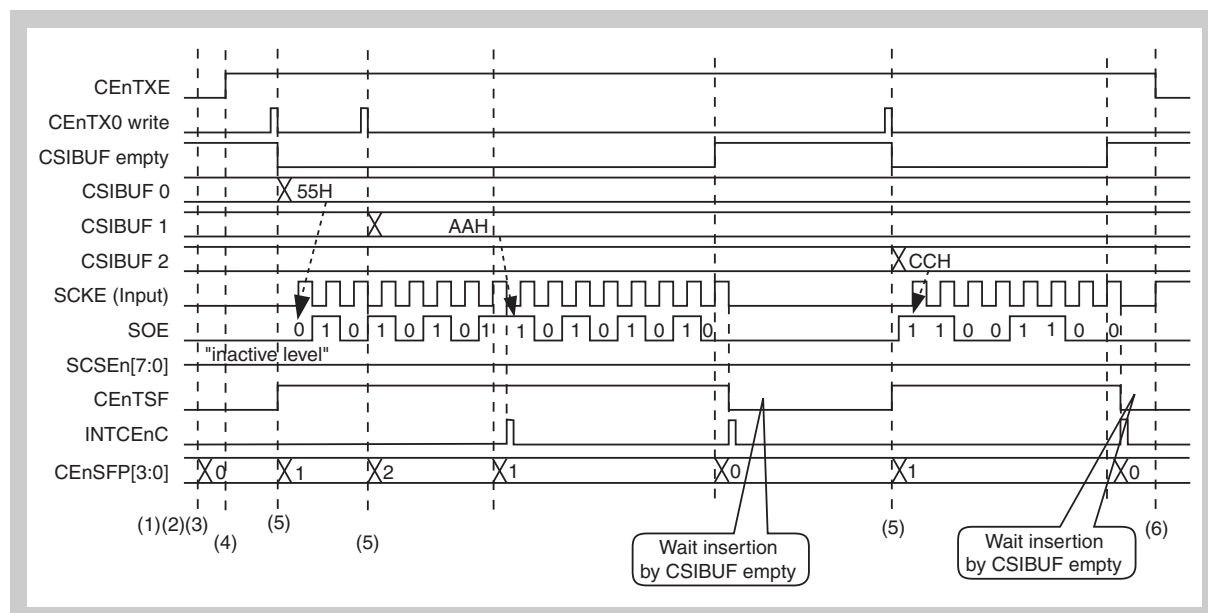


Figure 13-30 Single transfer mode (slave, transmit only) timing

1. Set the CEnCTL0 register's CEnPWR bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CEnCTL1 and CEnCTL2 registers to specify the transfer mode.
3. Write "1" in the CEnSTR register's CEnPCT bit to clear all FIFO pointers.
4. Specify the transfer mode using the CEnCTL0 register's CEnTMS, CEnDIR, and CEnSIT bits; at the same time, set the CEnTXE bit to 1 to enable transmission.
5. Make sure that the CEnSTR register's CEnFLF bit is set to 0, then write transmission data in the CEnTX0 register. (In the slave mode, there is no need to set data in the CEnCS register, as the chip select pins SCSEn[3:0] are not used.)

Repeat step (5) until the last element to be transmitted is written in the CEnTX0 register.

6. Verify that CEnSTR.CEnTSF = 0 (CSIE in idle state) and set the CEnCTL0 register's CEnTXE bit to 0 to disable transmission (end of transmission).

To continue transmission, repeat step (5) before executing step (6).

### 13.4.5 Single transfer mode (slave mode, receive only mode)

MSB first (CEnDIR = 0), no INTCEnC delay (CEnSIT = 0), transmission wait disabled (CEnWE = 0), CS inactive disabled (CEnCSM = 0), CEnCKP = 0, CEnDAP = 0, transmission data length of 8 bits (CEnDL[3:0] = [1,0,0,0]):

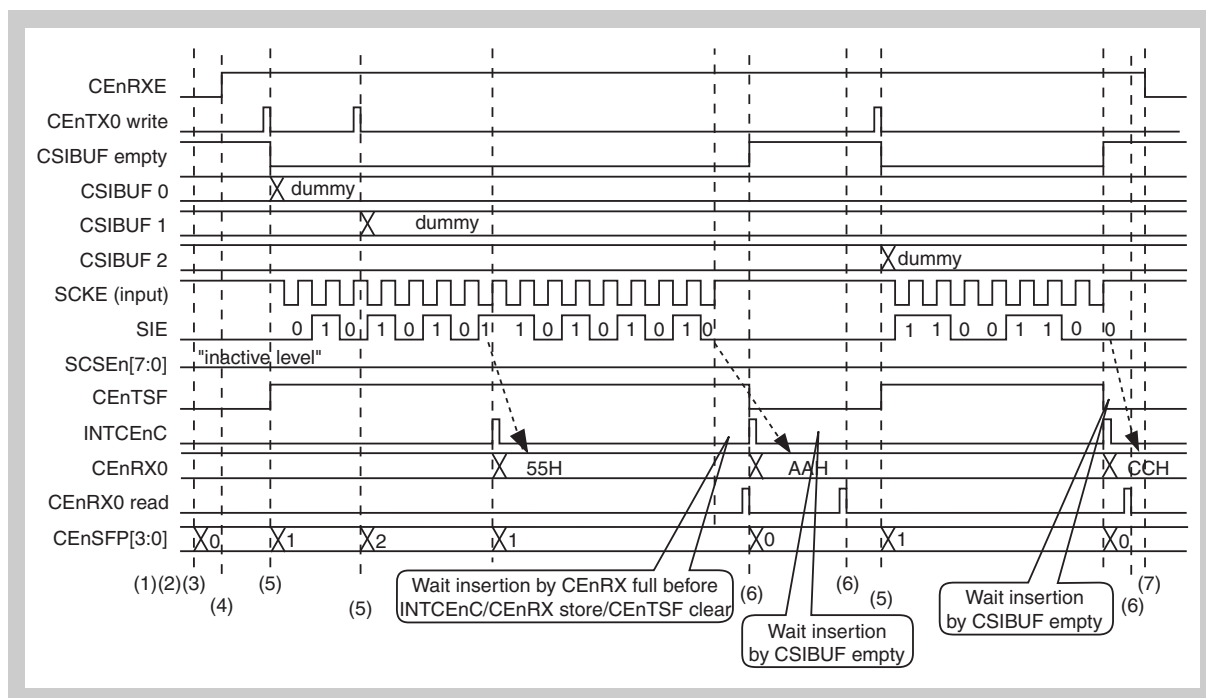


Figure 13-31 Single transfer mode (slave, receive only) timing

1. Set the CEnCTL0 register's CEnPWR bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CEnCTL1 and CEnCTL2 registers to specify the transfer mode.
3. Write "1" in the CEnSTR register's CEnPCT bit to clear all FIFO pointers.
4. Specify the transfer mode using the CEnCTL0 register's CEnTMS, CEnDIR, and CEnSIT bits; at the same time, set the CEnRXE bit to 1 to enable the receive operation.
5. Make sure that the CEnSTR register's CEnFLF bit is set to 0, then write dummy transmission data in the CEnTX0 register (start-of-receive trigger). (In the slave mode, there is no need to set data in the CEnCS register, as the chip select pins SCSEn[3:0] are not used.)
6. Check for a reception to be completed by monitoring the INTCEnC interrupt. If so, read the CEnRX0 register.

Repeat steps (5) and (6) until the last element is received and read from the CEnRX0 register.

7. Verify that CEnSTR.CEnTSF = 0 (CSIE in idle state) and set the CEnCTL0 register's CEnRXE bit to 0 to disable the receive operation (end of receive operation).

**Note** The SOEn pin is invalid and maintaining its signal level, as the output latch is disabled.

### 13.4.6 Single transfer mode (slave mode, transmit/receive mode)

MSB first (CEnDIR = 0), no INTCEnC delay (CEnSIT = 0), transmission wait disabled (CEnWE = 0), CS inactive disabled (CEnCSM = 0), CEnCKP = 0, CEnDAP = 1, transmission data length of 8 bits (CEnDL[3:0] = [1,0,0,0]):

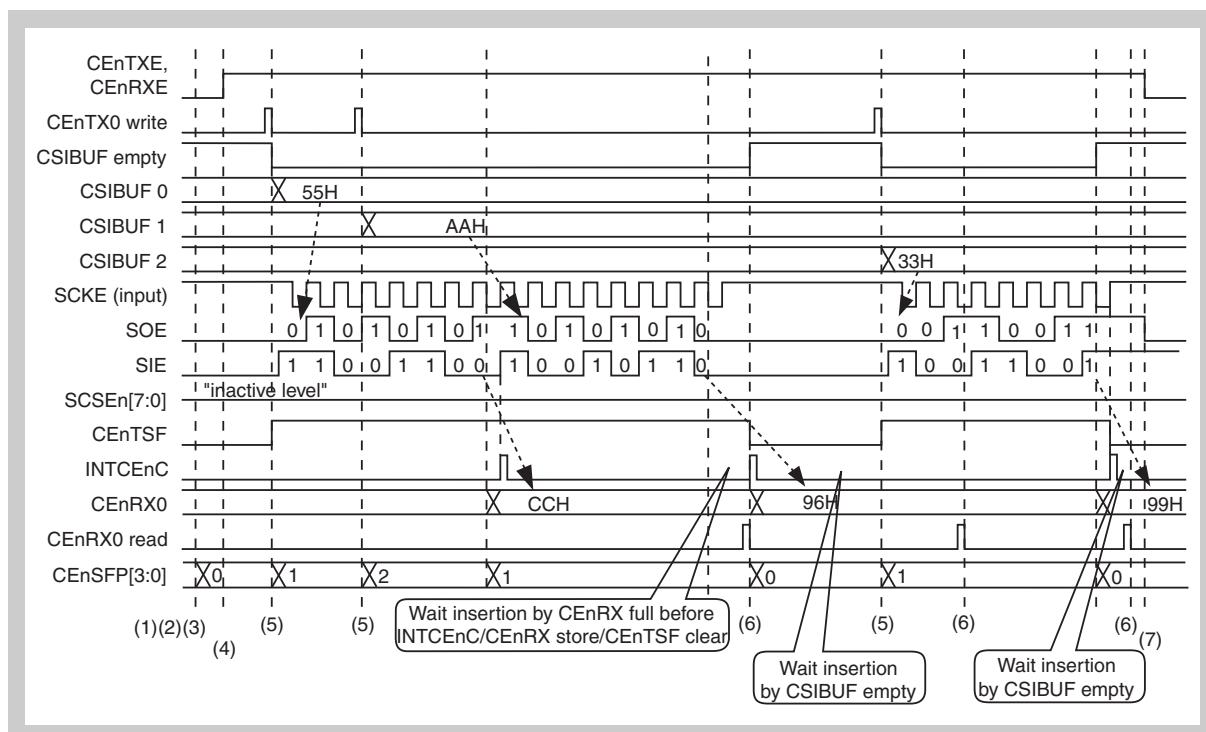


Figure 13-32 Single transfer mode (slave, transmit/receive) timing

1. Set the CEnCTL0 register's CEnPWR bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CEnCTL1 and CEnCTL2 registers to specify the transfer mode.
3. Write "1" in the CEnSTR register's CEnPCT bit to clear all FIFO pointers.
4. Specify the transmit mode using the CEnCTL0 register's CEnTMS, CEnDIR, and CEnSIT bits; at the same time, set the CEnTXE and CEnRXE bits to 1 to enable the transmit/receive operation.
5. Make sure that the CEnSTR register's CEnFLF bit is set to 0, then write transmission data in the CEnTX0 register. (In the slave mode, there is no need to set data in the CEnCS register, as the chip select pins SCSEn[3:0] are not used.)
6. Check for a reception to be completed by monitoring the INTCEnC interrupt. If so, read the CEnRX0 register.

Repeat steps (5) and (6) until the last element is send/received and read from the CEnRX0 register.

7. Verify that CEnSTR.CEnTSF = 0 (CSIE in idle state) and set the CEnCTL0 register's CEnTXE and CEnRXE bits to 0 to disable the transmit/receive operation (end of transmit/receive operation).

### 13.4.7 Block transfer mode (master mode, transmit only mode)

MSB first (CEnDIR = 0), no INTCEnC delay (CEnSIT = 0), transmission wait disabled (CEnWE = 0), CS inactive disabled (CEnCSM = 0), CEnCKP = 0, CEnDAP = 0, transmission data length of 8 bits (CEnDL[3:0] = [1,0,0,0]):

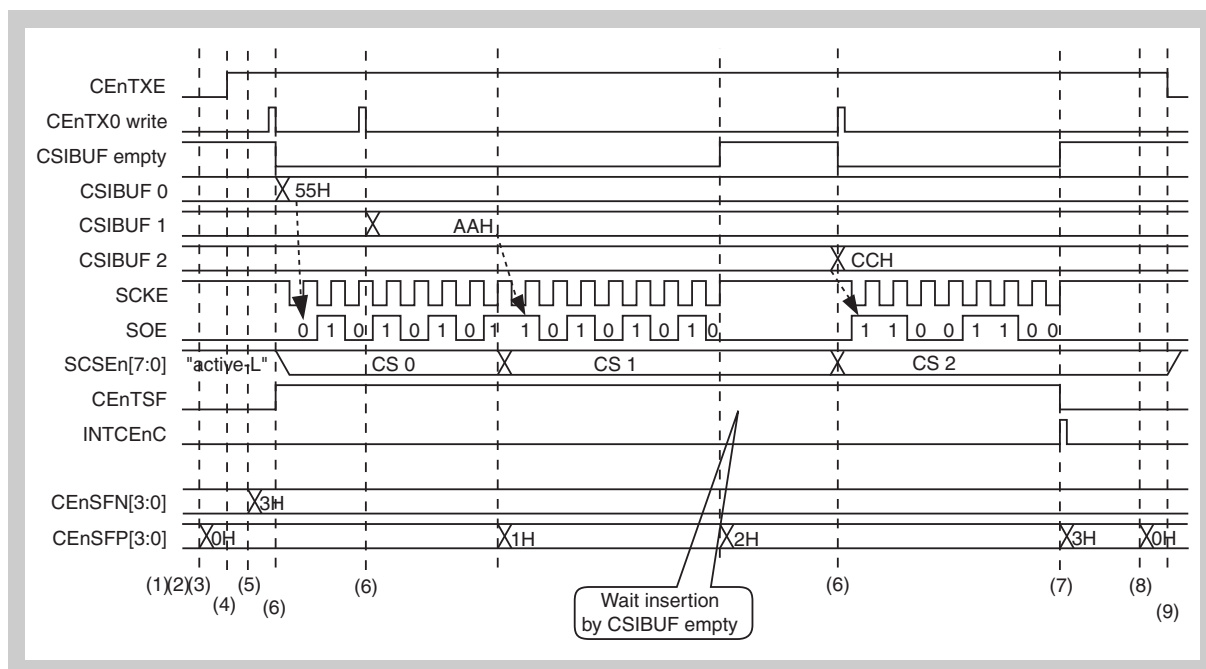


Figure 13-33 Block transfer mode (master, transmit only) timing

1. Set the CEnCTL0 register's CEnPWR bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CEnCTL1 and CEnCTL2 registers to specify the transfer mode.
3. Write "1" in the CEnSTR register's CEnPCT bit to clear all FIFO pointers.
4. Specify the transfer mode using the CEnCTL0 register's CEnTMS, CEnDIR, and CEnSIT bits; at the same time, set the CEnTXE bit to 1 to enable transmission.
5. Set the number of send-data items in the CEnCTL3 register's CEnSFN[3:0] bits.
6. Make sure that the CEnSTR register's CEnFLF bit is set to 0, then write chip-select data and transmission data in the CEnCS and CEnTX0 registers in this order.
7. Wait for the transmissions to be completed by monitoring the INTCEnC interrupt.
8. Write "1" in the CEnSTR register's CEnPCT bit and clear all FIFO pointers for the next transmission.

To continue transmission, repeat steps (5) - (8).

9. Verify that CEnSTR.CEnTSF = 0 (CSIE in idle state) and set the CEnCTL0 register's CEnTXE bit to 0 to disable transmission (end of transmission).

### 13.4.8 Block transfer mode (master mode, receive only mode)

MSB first (CEnDIR = 0), no INTCEnC delay (CEnSIT = 0), transmission wait disabled (CEnWE = 0), CS inactive disabled (CEnCSM = 0), CEnCKP = 0, CEnDAP = 1, transmission data length of 8 bits (CEnDL[3:0] = [1,0,0,0]):

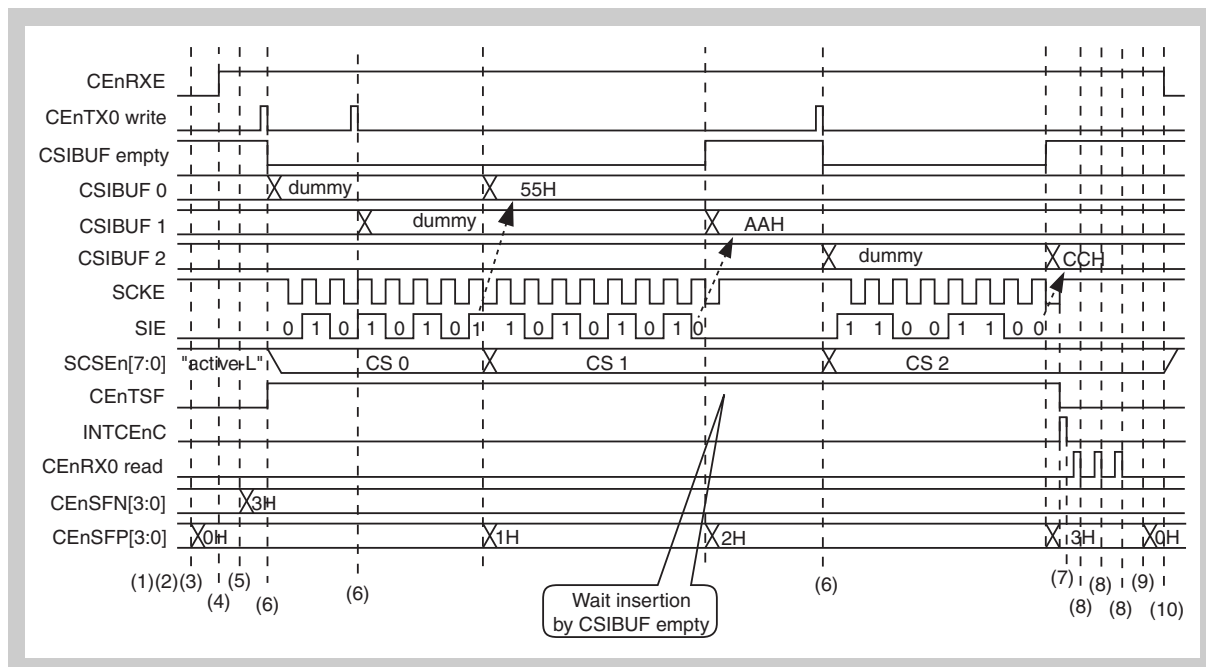


Figure 13-34 Block transfer mode (master, receive only) timing

1. Set the CEnCTL0 register's CEnPWR bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CEnCTL1 and CEnCTL2 registers to specify the transfer mode.
3. Write "1" in the CEnSTR register's CEnPCT bit to clear all FIFO pointers.
4. Specify the transfer mode using the CEnCTL0 register's CEnTMS, CEnDIR, and CEnSIT bits; at the same time, set the CEnRXE bit to 1 to enable the receive operation.
5. Set the number of receive-data items in the CEnCTL3 register's CEnSFN[3:0] bits.
6. Make sure that the CEnSTR register's CEnFLF bit is set to 0, then write chip-select data and dummy transmission data in the CEnCS and CEnTX0 registers in this order (start-of-receive trigger).
7. Wait for the receptions to be completed by monitoring the INTCEnC interrupt.
8. Read the received data by multiple read of the CEnRX0 register (= sequential read from the FIFO).
9. Write "1" in the CEnSTR register's CEnPCT bit and clear all FIFO pointers for the next transmission.

To continue reception, repeat steps (5) - (9).

10. Verify that CEnSTR.CEnTSF = 0 (CSIE in idle state) and set the CEnCTL0 register's CEnRXE bit to 0 to disable the receive operation (end of receive operation).

**Note** The SOEn pin is invalid and maintains its signal level, as the output latch is disabled.

### 13.4.9 Block transfer mode (master mode, transmit/receive mode)

MSB first (CEnDIR = 0), no INTCEnC delay (CEnSIT = 0), transmission wait disabled (CEnWE = 0), CS inactive disabled (CEnCSM = 0), CEnCKP = 1, CEnDAP = 0, transmission data length of 8 bits (CEnDL[3:0] = [1,0,0,0]):

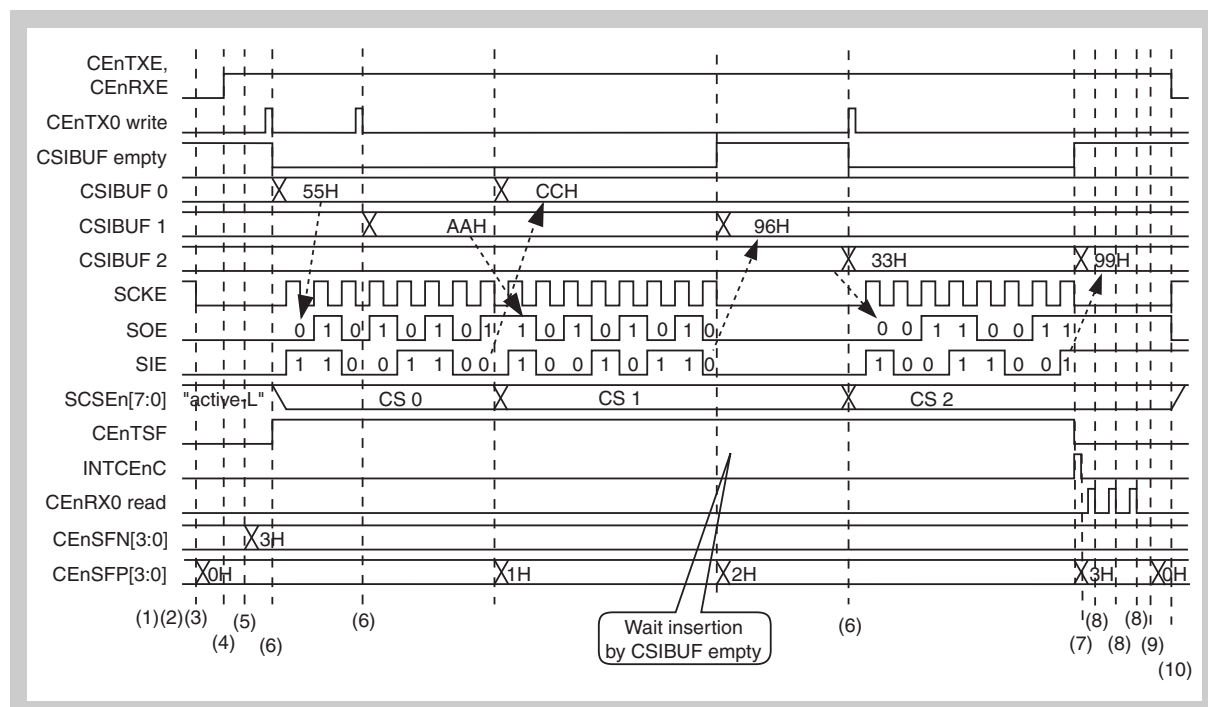


Figure 13-35 Block transfer mode (master, transmit/receive) timing

1. Set the CEnCTL0 register's CEnPWR bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CEnCTL1 and CEnCTL2 registers to specify the transfer mode.
3. Write "1" in the CEnSTR register's CEnPCT bit to clear all FIFO pointers.
4. Specify the transfer mode using the CEnCTL0 register's CEnTMS, CEnDIR, and CEnSIT bits; at the same time, set the CEnTXE and CEnRXE bits to 1 to enable the transmit/receive operation.
5. Set the number of transmit/receive data items in the CEnCTL3 register's CEnSFN[3:0] bits.
6. Make sure that the CEnSTR register's CEnFLF bit is set to 0, then write chip-select data and transmission data in the CEnCS and CEnTX0 registers in this order.
7. Wait for the transmissions/receptions to be completed by monitoring the INTCEnC interrupt.
8. Read the received data by multiple read of the CEnRX0 register (= sequential read from the FIFO).
9. Write "1" in the CEnSTR register's CEnPCT bit and clear all FIFO pointers for the next transmission.

To continue transmission/reception, repeat steps (5) - (9).

10. Verify that CEnSTR.CEnTSE = 0 (CSIE in idle state) and set the CEnCTL0 register's CEnTXE and CEnRXE bits to 0 to disable the transmit/receive operation (end of transmit/receive operation).

### 13.4.10 Block transfer mode (slave mode, transmit only mode)

MSB first (CEnDIR = 0), no INTCEnC delay (CEnSIT = 0), transmission wait disabled (CEnWE = 0), CS inactive disabled (CEnCSM = 0), CEnCKP = 1, CEnDAP = 1, transmission data length of 8 bits (CEnDL[3:0] = [1,0,0,0]):

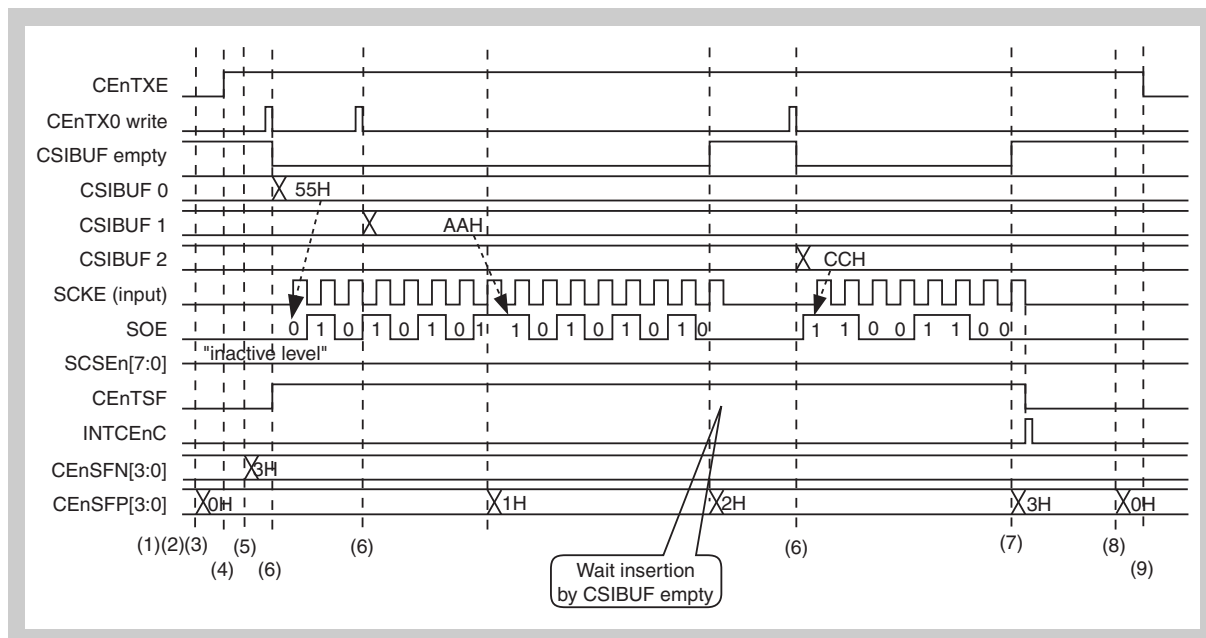


Figure 13-36 Block transfer mode (slave, transmit only) timing

1. Set the CEnCTL0 register's CEnPWR bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CEnCTL1 and CEnCTL2 registers to specify the transfer mode.
3. Write "1" in the CEnSTR register's CEnPCT bit to clear all FIFO pointers.
4. Specify the transfer mode using the CEnCTL0 register's CEnTMS, CEnDIR, and CEnSIT bits; at the same time, set the CEnTXE bit to 1 to enable transmission.
5. Set the number of send-data items in the CEnCTL3 register's CEnSFN[3:0] bits.
6. Make sure that the CEnSTR register's CEnFLF bit is set to 0, then write transmission data in the CEnTX0 register. (In the slave mode, there is no need to set data in the CEnCS register, as the chip select pins SCSEn[3:0] are not used.)
7. Wait for the transmissions to be completed by monitoring the INTCEnC interrupt.
8. Write "1" in the CEnSTR register's CEnPCT bit and clear all FIFO pointers for the next transmission.

To continue transmission, repeat steps (5) - (8).

9. Verify that CEnSTR.CEnTSF = 0 (CSIE in idle state) and set the CEnCTL0 register's CEnTXE bit to 0 to disable transmission (end of transmission).



### 13.4.11 Block transfer mode (slave mode, receive only mode)

MSB first (CEnDIR = 0), no INTCEnC delay (CEnSIT = 0), transmission wait disabled (CEnWE = 0), CS inactive disabled (CEnCSM = 0), CEnCKP = 0, CEnDAP = 0, transmission data length of 8 bits (CEnDL[3:0] = [1,0,0,0]):

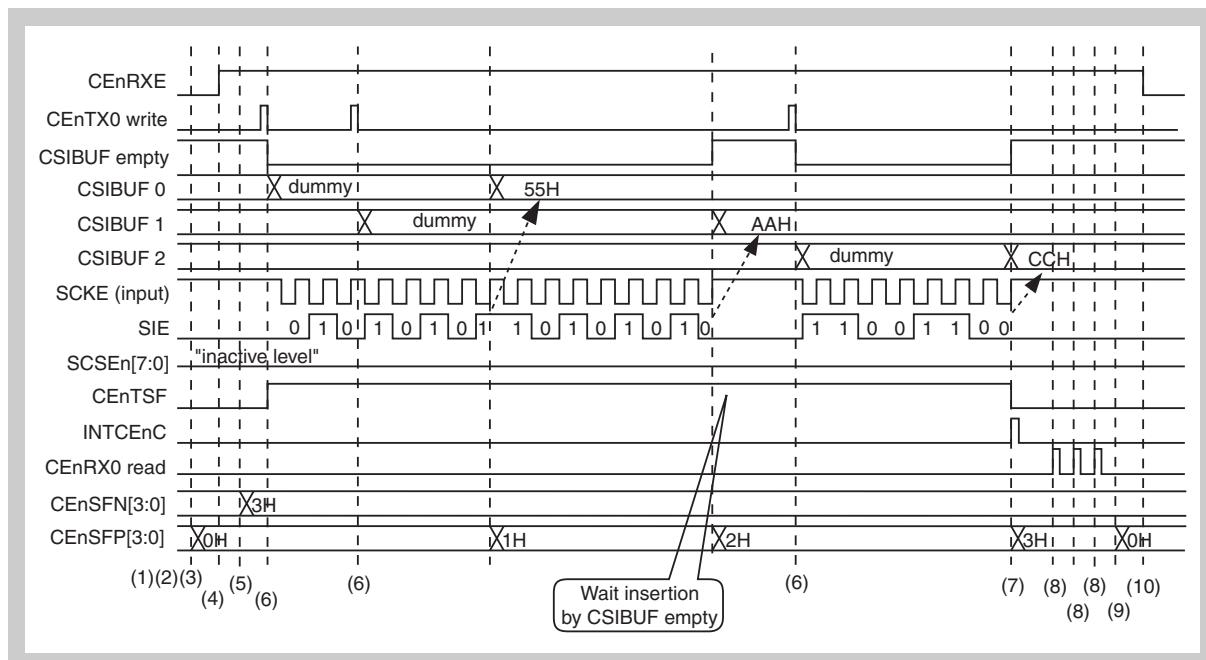


Figure 13-37 Block transfer mode (slave, receive only) timing

1. Set the CEnCTL0 register's CEnPWR bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CEnCTL1 and CEnCTL2 registers to specify the transfer mode.
3. Write "1" in the CEnSTR register's CEnPCT bit to clear all FIFO pointers.
4. Specify the transfer mode using the CEnCTL0 register's CEnTMS, CEnDIR, and CEnSIT bits; at the same time, set the CEnRXE bit to 1 to enable the receive operation.
5. Set the number of receive-data items in the CEnCTL3 register's CEnSFN[3:0] bits.
6. Make sure that the CEnSTR register's CEnFLF bit is set to 0, then write dummy transmission data in the CEnTX0 register (start-of-receive trigger). (In the slave mode, there is no need to set data in the CEnCS register, as the chip select pins SCSEn[3:0] are not used.)
7. Wait for the receptions to be completed by monitoring the INTCEnC interrupt.
8. Read the received data by multiple read of the CEnRX0 register (= sequential read from the FIFO).
9. Write "1" in the CEnSTR register's CEnPCT bit and clear all FIFO pointers for the next transmission.

To continue reception, repeat steps (5) - (9).

10. Verify that CEnSTR.CEnTSF = 0 (CSIE in idle state) and set the CEnCTL0 register's CEnRXE bit to 0 to disable the receive operation (end of receive operation).

**Note** The SOEn pin is invalid and maintains its signal level, as the output latch is disabled.

### 13.4.12 Block transfer mode (slave mode, transmit/receive mode)

MSB first (CEnDIR = 0), no INTCEnC delay (CEnSIT = 0), transmission wait disabled (CEnWE = 0), CS inactive disabled (CEnCSM = 0), CEnCKP = 0, CEnDAP = 1, transmission data length of 8 bits (CEnDL[3:0] = [1,0,0,0]):

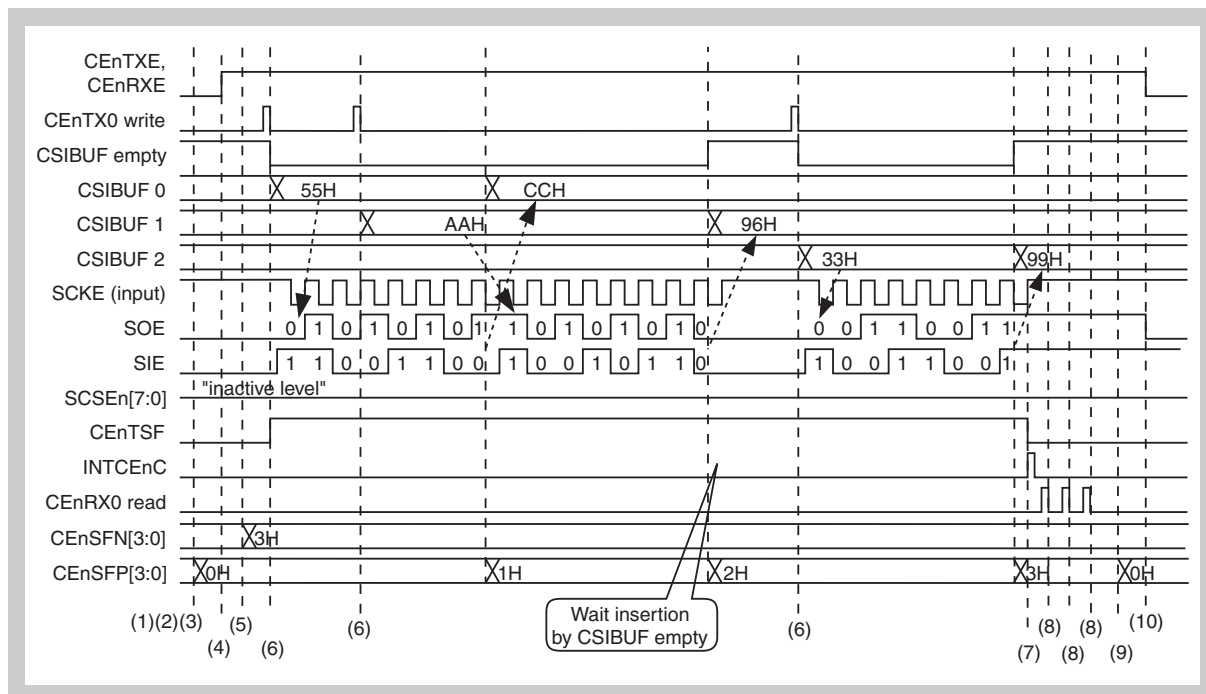


Figure 13-38 Block transfer mode (slave, transmit/receive) timing

1. Set the CEnCTL0 register's CEnPWR bit to 1 to enable the supply of the Queued CSI operation clock.
2. Set the CEnCTL1 and CEnCTL2 registers to specify the transfer mode.
3. Write "1" in the CEnSTR register's CEnPCT bit to clear all FIFO pointers.
4. Specify the transfer mode using the CEnCTL0 register's CEnTMS, CEnDIR, and CEnSIT bits; at the same time, set the CEnTXE and CEnRXE bits to 1 to enable the transmit/receive operation.
5. Set the number of transmit/receive data items in the CEnCTL3 register's CEnSFN[3:0] bits.
6. Make sure that the CEnSTR register's CEnFLF bit is set to 0, then write transmission data in the CEnTX0 register. (In the slave mode, there is no need to set data in the CEnCS register, as the chip select pins SCSEn[3:0] are not used.)
7. Wait for the transmissions/receptions to be completed by monitoring the INTCEnC interrupt.
8. Read the received data by multiple read of the CEnRX0 register (= sequential read from the FIFO).
9. Write "1" in the CEnSTR register's CEnPCT bit and clear all FIFO pointers for the next transmission.

To continue transmission, repeat steps (5) - (9).

10. Verify that CEnSTR.CEnTSF = 0 (CSIE in idle state) and set the CEnCTL0 register's CEnTXE and CEnRXE bits to 0 to disable the transmit/receive operation (end of transmit/receive operation).

## 13.5 Cautions

### 13.5.1 Inconsistent read data in master mode

**Description** Data read from the CEnRx0 register may be incorrect due to a wrong data width being applied to the received data. The behaviour occurs if all of the following conditions are met:

- Enhanced timing mode is used (CEnCTL4.CEnOPE = 1)
- Different data width configured for each chip select SCSEnm (different CEnOPTm.CEnDLm[1:0])
- Master mode is enabled (CEnCTL1.CEnCKS[2:0]  $\neq$  111<sub>B</sub>)
- Receive or Transmit-Receive is enabled (CEnCTL0.CEnRXE = 1)
- Serial data direction is LSB first (CEnCTL0.CEnDIR = 1)

**Workaround** Configure the CSIE in such a way that not all of above mentioned conditions apply, e.g. use MSB first or configure the same data length for each chip select SCSEnm.

### 13.5.2 Transmit/receive enable after CEnTX0 write

**Description** If transmission (CEnCTL0.CEnTXE = 1) or reception (CEnCTL0.CEnRXE = 1) is enabled after writing data to the transmission data buffer register CEnTX0 the receive/transmission operation may fail.

**Workaround** Enable transmission/reception before writing to CEnTX0.

# Chapter 14 CAN Controller (CAN)

The microcontroller features an on-chip 2-channel CAN (Controller Area Network) controller that complies with the CAN protocol as standardized in ISO 11898.

- Note**
1. Throughout this chapter, the individual CAN channels are identified by “n” (n = 0, 1), for example CANn, or CnGMCTRL for the CANn global control register.
  2. Throughout this chapter, the CAN message buffer registers are identified by “m” (m = 0 to 31), for example COMDATA4m for CAN0 message data byte 4 of message buffer register m.

## 14.1 Features

- Compliant with ISO 11898 and tested according to ISO/DIS 16845 (CAN conformance test)
- Standard frame and extended frame transmission/reception enabled
- Transfer rate: 1 Mbps max. (if CAN clock input  $\geq$  8 MHz, for 32 channels)
- 32 message buffers per channel
- Receive/transmit history list function
- Automatic block transmission function
- Multi-buffer receive block function
- Mask setting of four patterns is possible for each channel
- Data bit time, communication baud rate and sample point can be controlled by CAN module bit-rate prescaler register (CnBRP) and bit rate register (CnBTR)
  - As an example the following sample-point configurations can be configured:
    - 66.7%, 70.0%, 75.0%, 80.0%, 81.3%, 85.0%, 87.5%
    - Baud rates in the range of 10 kbps up to 1000 kbps can be configured
- Enhanced features:
  - Each message buffer can be configured to operate as a transmit or a receive message buffer
  - Transmission priority is controlled by the identifier or by mailbox number (selectable)
  - A transmission request can be aborted by clearing the dedicated Transmit-Request flag of the concerned message buffer.
  - Automatic block transmission operation mode (ABT)
  - Time stamp function for CAN channels 0 to n in collaboration with timers capture channels

### 14.1.1 Overview of functions

Table 14-1 presents an overview of the CAN Controller functions.

**Table 14-1 Overview of functions**

Function	Details
Protocol	CAN protocol ISO 11898 (standard and extended frame transmission/reception)
Baud rate	Maximum 1 Mbps (CAN clock input $\geq 8$ MHz)
Data storage	Storing messages in the CAN RAM
Number of messages	<ul style="list-style-type: none"> <li>32 message buffers per channel</li> <li>Each message buffer can be set to be either a transmit message buffer or a receive message buffer.</li> </ul>
Message reception	<ul style="list-style-type: none"> <li>Unique ID can be set to each message buffer.</li> <li>Mask setting of four patterns is possible for each channel.</li> <li>A receive completion interrupt is generated each time a message is received and stored in a message buffer.</li> <li>Two or more receive message buffers can be used as a FIFO receive buffer (multi-buffer receive block function).</li> <li>Receive history list function</li> </ul>
Message transmission	<ul style="list-style-type: none"> <li>Unique ID can be set to each message buffer.</li> <li>Transmit completion interrupt for each message buffer</li> <li>Message buffer number 0 to 7 specified as the transmit message buffer can be set for automatic block transfer. Message transmission interval is programmable (automatic block transmission function (hereafter referred to as "ABT")).</li> <li>Transmission history list function</li> </ul>
Remote frame processing	Remote frame processing by transmit message buffer
Time stamp function	<ul style="list-style-type: none"> <li>The time stamp function can be set for a message reception when a 16-bit timer is used in combination.</li> <li>Time stamp capture trigger can be selected (SOF or EOF in a CAN message frame can be detected.).</li> <li>The time stamp function can be set for a transmit message.</li> </ul>
Diagnostic function	<ul style="list-style-type: none"> <li>Readable error counters</li> <li>"Valid protocol operation flag" for verification of bus connections</li> <li>Receive-only mode</li> <li>Single-shot mode</li> <li>CAN protocol error type decoding</li> <li>Self-test mode</li> </ul>
Release from bus-off state	<ul style="list-style-type: none"> <li>Forced release from bus-off (by ignoring timing constraint) possible by software.</li> <li>No automatic release from bus-off (software must re-enable).</li> </ul>
Power save mode	<ul style="list-style-type: none"> <li>CAN Sleep mode (can be woken up by CAN bus)</li> <li>CAN Stop mode (cannot be woken up by CAN bus)</li> </ul>

### 14.1.2 Configuration

The CAN Controller is composed of the following four blocks.

- **NPB interface**  
This functional block provides an NPB (Peripheral I/O Bus) interface and means of transmitting and receiving signals between the CAN module and the host CPU.
- **MCM (Memory Control Module)**  
This functional block controls access to the CAN protocol layer and to the CAN RAM within the CAN module.
- **CAN protocol layer**  
This functional block is involved in the operation of the CAN protocol and its related settings.
- **CAN RAM**  
This is the CAN memory functional block, which is used to store message IDs, message data, etc.

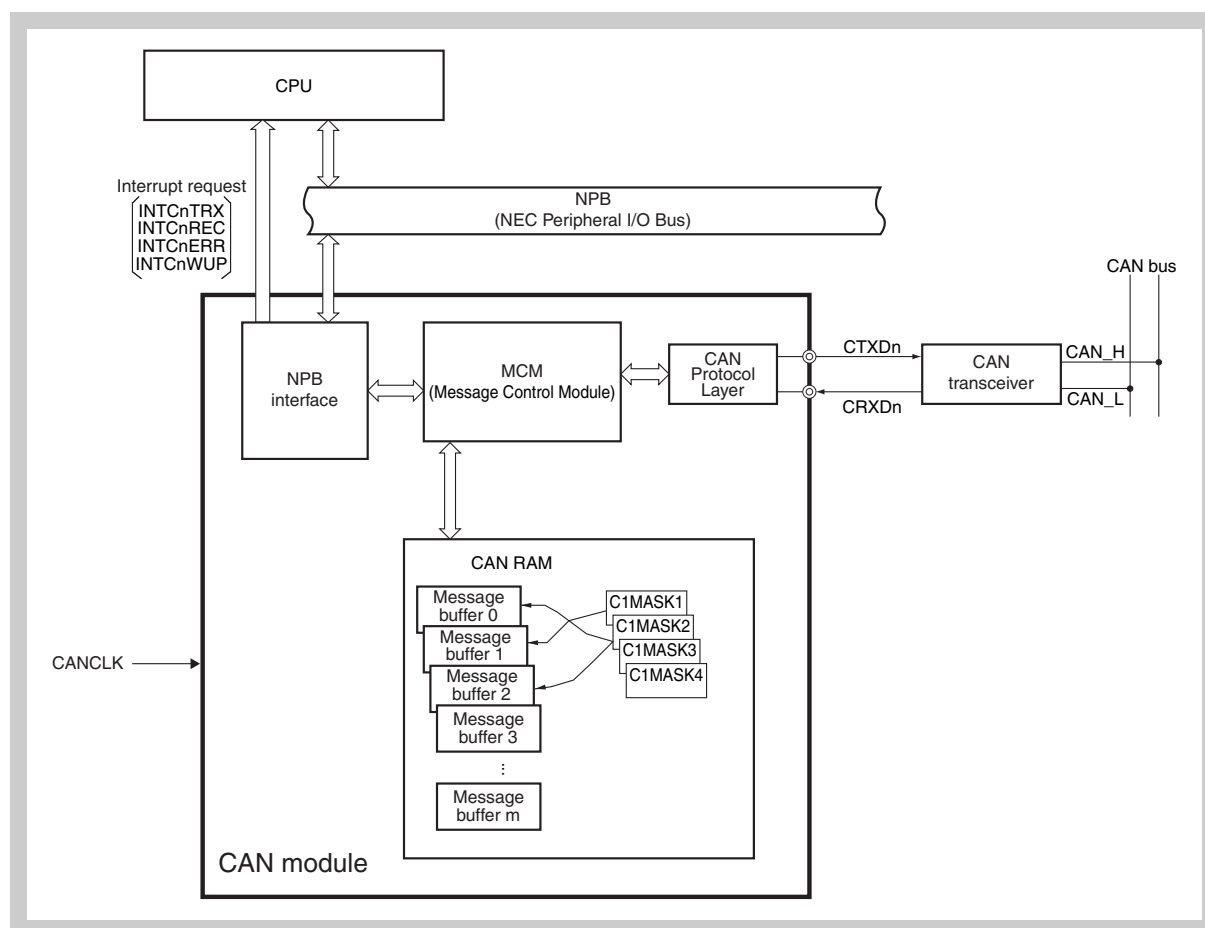


Figure 14-1 Block diagram of CAN module

## 14.2 CAN Protocol

CAN (Controller Area Network) is a high-speed multiplex communication protocol for real-time communication in automotive applications (class C). CAN is prescribed by ISO 11898. For details, refer to the ISO 11898 specifications.

The CAN specification is generally divided into two layers: a physical layer and a data link layer. In turn, the data link layer includes logical link and medium access control. The composition of these layers is illustrated below.

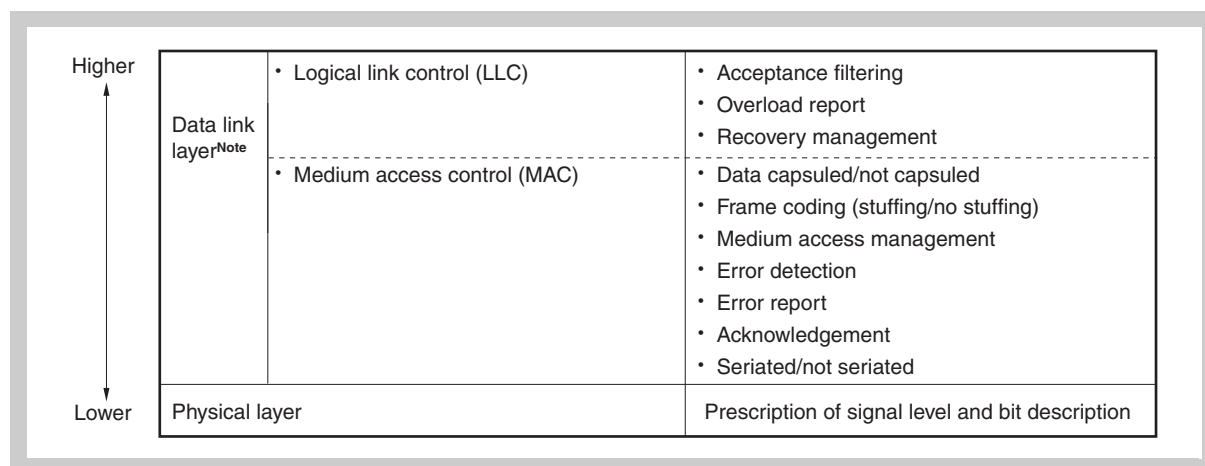


Figure 14-2 Composition of layers

**Note** CAN Controller specification

### 14.2.1 Frame format

#### (1) Standard format frame

- The standard format frame uses 11-bit identifiers, which means that it can handle up to 2,048 messages.

#### (2) Extended format frame

- The extended format frame uses 29-bit (11 bits + 18 bits) identifiers, which increases the number of messages that can be handled to  $2,048 \times 2^{18}$  messages.
- An extended format frame is set when “recessive level” (CMOS level of “1”) is set for both the SRR and IDE bits in the arbitration field.

## 14.2.2 Frame types

The following four types of frames are used in the CAN protocol.

**Table 14-2 Frame types**

Frame Type	Description
Data frame	Frame used to transmit data
Remote frame	Frame used to request a data frame
Error frame	Frame used to report error detection
Overload frame	Frame used to delay the next data frame or remote frame

### (1) Bus value

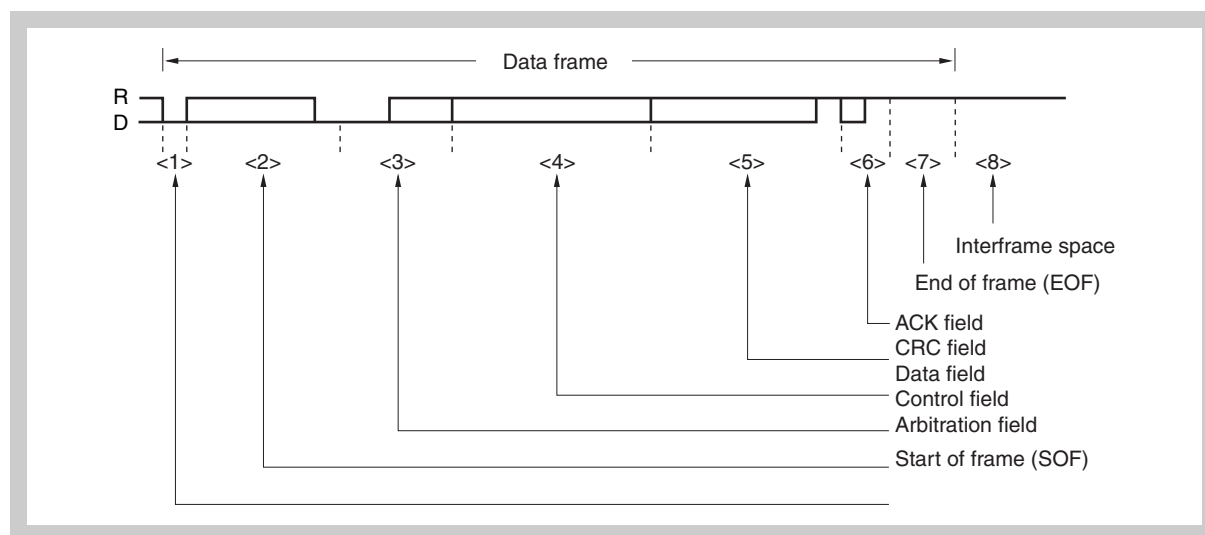
The bus values are divided into dominant and recessive.

- Dominant level is indicated by logical 0.
- Recessive level is indicated by logical 1.
- When a dominant level and a recessive level are transmitted simultaneously, the bus value becomes dominant level.

## 14.2.3 Data frame and remote frame

### (1) Data frame

A data frame is composed of seven fields.



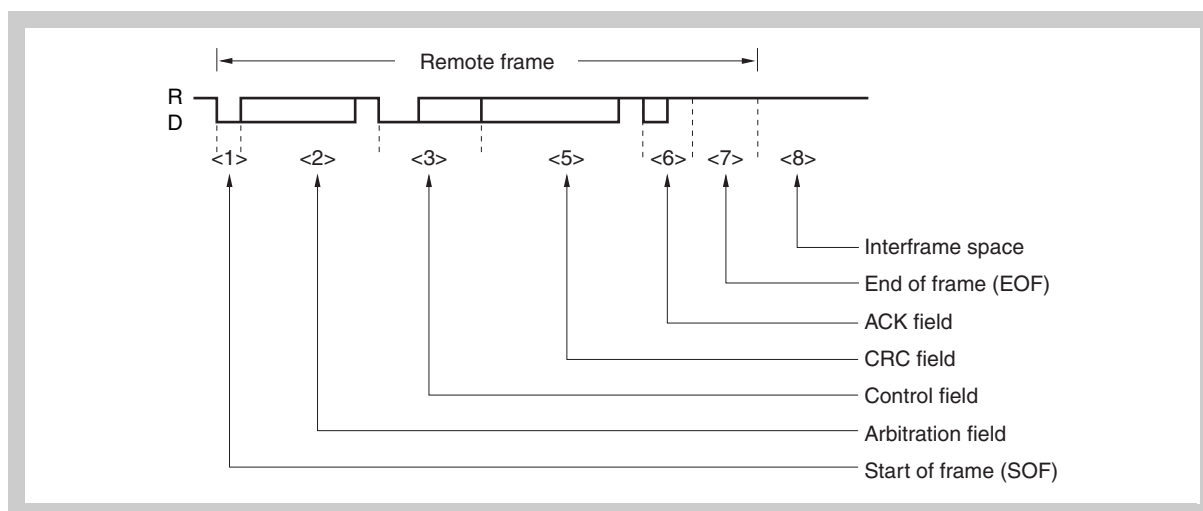
**Figure 14-3 Data frame**

**Note** D: Dominant = 0  
R: Recessive = 1



**(2) Remote frame**

A remote frame is composed of six fields.

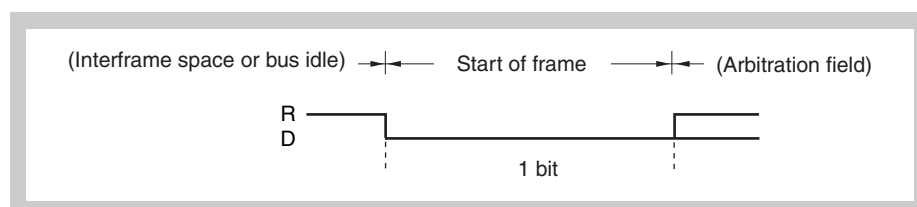


**Figure 14-4 Remote frame**

- Note**
1. The data field is not transferred even if the control field's data length code is not "0000<sub>B</sub>".
  2. D: Dominant = 0  
R: Recessive = 1

**(3) Description of fields****(a) Start of frame (SOF)**

The start of frame field is located at the start of a data frame or remote frame.



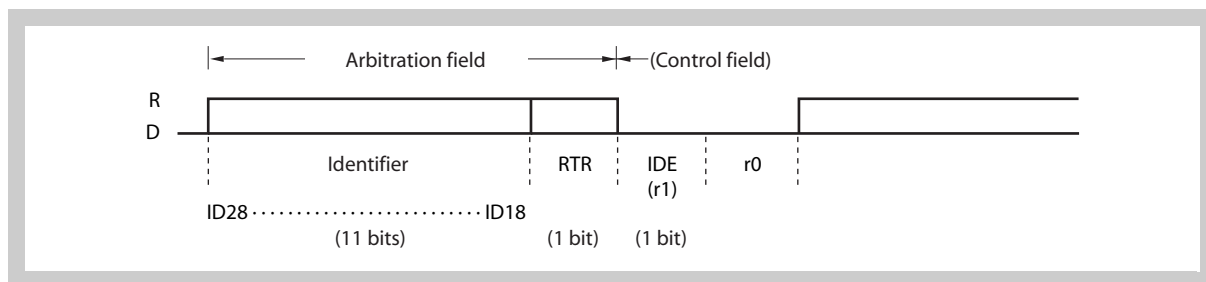
**Figure 14-5 Start of frame (SOF)**

- Note**
- D: Dominant = 0  
R: Recessive = 1

- If dominant level is detected in the bus idle state, a hard-synchronization is performed (the current TQ is assigned to be the SYNC segment).
- If dominant level is sampled at the sample point following such a hard-synchronization, the bit is assigned to be a SOF. If recessive level is detected, the protocol layer returns to the bus idle state and regards the preceding dominant pulse as a disturbance only. No error frame is generated in such case.

**(b) Arbitration field**

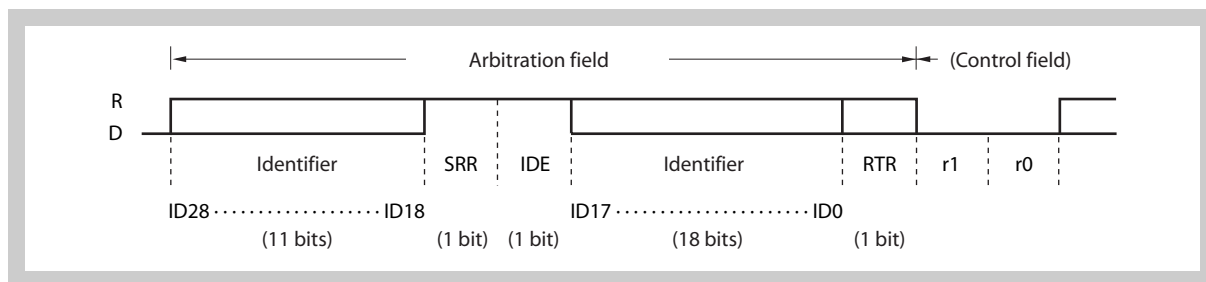
The arbitration field is used to set the priority, data frame/remote frame, and frame format.



**Figure 14-6** Arbitration field (in standard format mode)

- Caution**
1. ID28 to ID18 are identifiers.
  2. An identifier is transmitted MSB first.

**Note** D: Dominant = 0  
R: Recessive = 1



**Figure 14-7** Arbitration field (in extended format mode)

- Caution**
1. ID28 to ID18 are identifiers.
  2. An identifier is transmitted MSB first.

**Note** D: Dominant = 0  
R: Recessive = 1

**Table 14-3** RTR frame settings

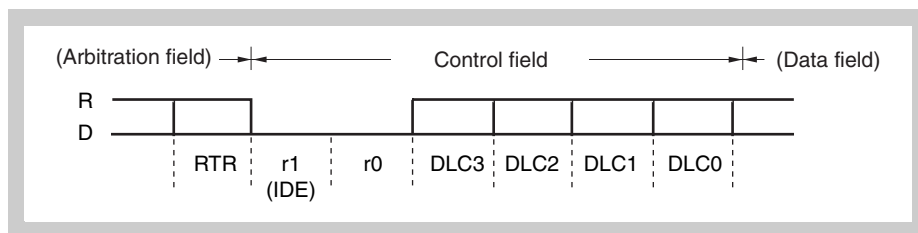
Frame type	RTR bit
Data frame	0 (D)
Remote frame	1 (R)

**Table 14-4** Frame format setting (IDE bit) and number of identifier (ID) bits

Frame format	SRR bit	IDE bit	Number of bits
Standard format mode	None	0 (D)	11 bits
Extended format mode	1 (R)	1 (R)	29 bits

**(c) Control field**

The control field sets “DLC” as the number of data bytes in the data field (DLC = 0 to 8).



**Figure 14-8** Control field

**Note** D: Dominant = 0  
R: Recessive = 1

In a standard format frame, the control field's IDE bit is the same as the r1 bit.

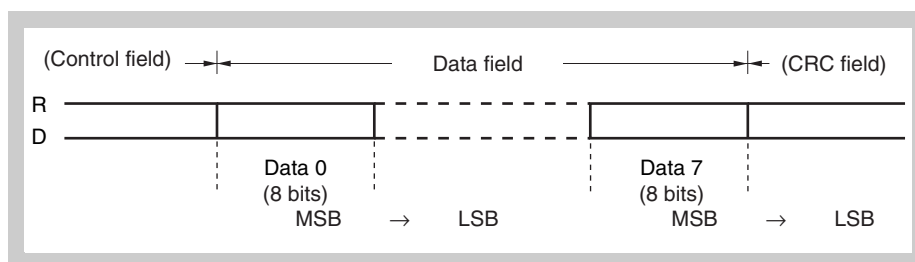
**Table 14-5** Data length setting

Data length code				Data byte count
DLC3	DLC2	DLC1	DLC0	
0	0	0	0	0 bytes
0	0	0	1	1 byte
0	0	1	0	2 bytes
0	0	1	1	3 bytes
0	1	0	0	4 bytes
0	1	0	1	5 bytes
0	1	1	0	6 bytes
0	1	1	1	7 bytes
1	0	0	0	8 bytes
Other than above				8 bytes regardless of the value of DLC3 to DLC0

**Caution** In the remote frame, there is no data field even if the data length code is not 0000<sub>B</sub>.

**(d) Data field**

The data field contains the amount of data (byte units) set by the control field. Up to 8 units of data can be set.

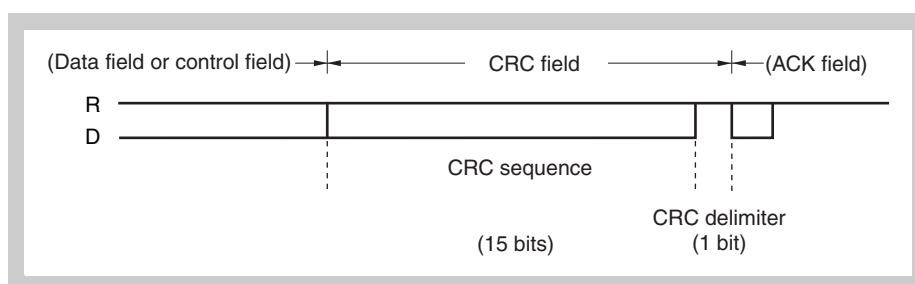


**Figure 14-9 Data field**

**Note** D: Dominant = 0  
R: Recessive = 1

**(e) CRC field**

The CRC field is a 16-bit field that is used to check for errors in transmit data.



**Figure 14-10 CRC field**

**Note** D: Dominant = 0  
R: Recessive = 1

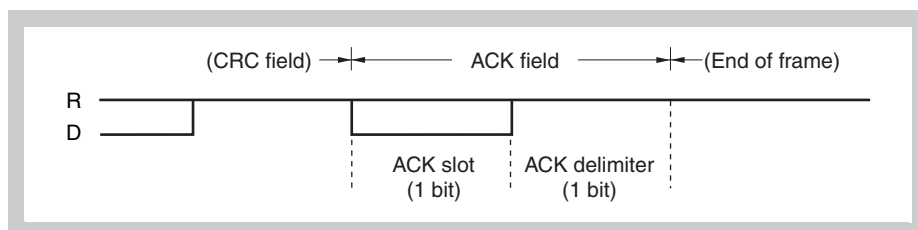
- The polynomial  $P(X)$  used to generate the 15-bit CRC sequence is expressed as follows.

$$P(X) = X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1$$

- Transmitting node:** Transmits the CRC sequence calculated from the data (before bit stuffing) in the start of frame, arbitration field, control field, and data field.
- Receiving node:** Compares the CRC sequence calculated using data bits that exclude the stuffing bits in the receive data with the CRC sequence in the CRC field. If the two CRC sequences do not match, the node issues an error frame.

**(f) ACK field**

The ACK field is used to acknowledge normal reception.



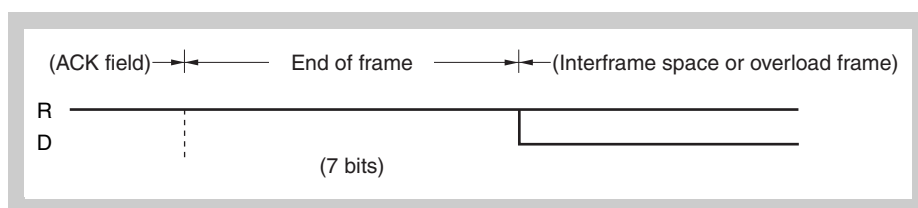
**Figure 14-11 ACK field**

**Note** D: Dominant = 0  
R: Recessive = 1

- If no CRC error is detected, the receiving node sets the ACK slot to the dominant level.
- The transmitting node outputs two recessive-level bits.

**(g) End of frame (EOF)**

The end of frame field indicates the end of data frame/remote frame.



**Figure 14-12 End of frame (EOF)**

**Note** D: Dominant = 0  
R: Recessive = 1

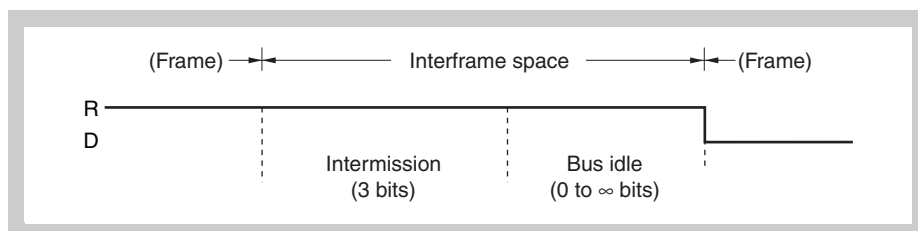
**(h) Interframe space**

The interframe space is inserted after a data frame, remote frame, error frame, or overload frame to separate one frame from the next.

- The bus state differs depending on the error status.

- **Error active node**

The interframe space consists of a 3-bit intermission field and a bus idle field.

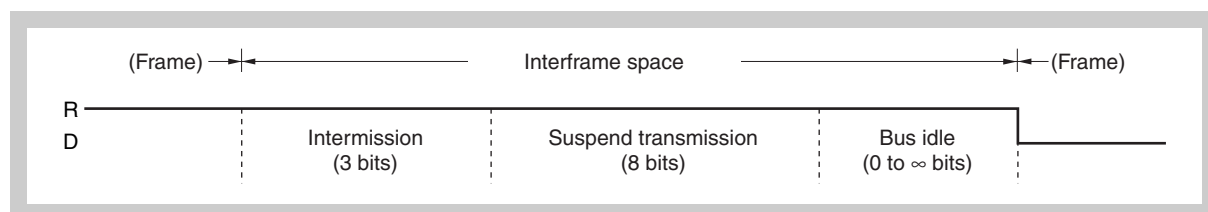


**Figure 14-13 Interframe space (error active node)**

- Note**
1. Bus idle: State in which the bus is not used by any node.
  2. D: Dominant = 0  
R: Recessive = 1

- **Error passive node**

The interframe space consists of an intermission field, a suspend transmission field, and a bus idle field.



**Figure 14-14 Interframe space (error passive node)**

- Note**
1. Bus idle: State in which the bus is not used by any node.  
Suspend transmission: Sequence of 8 recessive-level bits transmitted from the node in the error passive status.
  2. D: Dominant = 0  
R: Recessive = 1

Usually, the intermission field is 3 bits. If the transmitting node detects a dominant level at the third bit of the intermission field, however, it executes transmission.

- Operation in error status

**Table 14-6 Operation in error status**

Error status	Operation
Error active	A node in this status can transmit immediately after a 3-bit intermission.
Error passive	A node in this status can transmit 8 bits after the intermission.

### 14.2.4 Error frame

An error frame is output by a node that has detected an error.

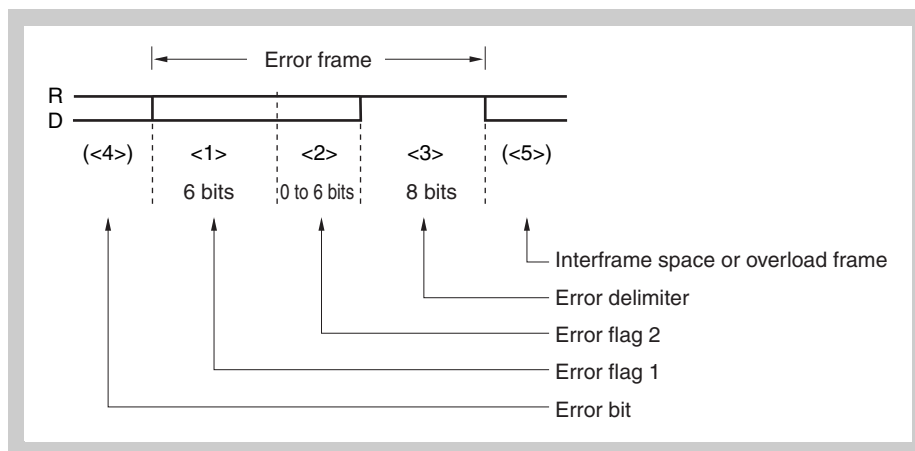


Figure 14-15 Error frame

**Note** D: Dominant = 0  
R: Recessive = 1

Table 14-7 Definition of error frame fields

No.	Name	Bit count	Definition
<1>	Error flag 1	6	Error active node: Outputs 6 dominant-level bits consecutively. Error passive node: Outputs 6 recessive-level bits consecutively.  If another node outputs a dominant level while one node is outputting a passive error flag, the passive error flag is not cleared until the same level is detected 6 bits in a row.
<2>	Error flag 2	0 to 6	Nodes receiving error flag 1 detect bit stuff errors and issues this error flag.
<3>	Error delimiter	8	Outputs 8 recessive-level bits consecutively. If a dominant level is detected at the 8th bit, an overload frame is transmitted from the next bit.
<4>	Error bit	–	The bit at which the error was detected. The error flag is output from the bit next to the error bit. In the case of a CRC error, this bit is output following the ACK delimiter.
<5>	Interframe space/overload frame	–	An interframe space or overload frame starts from here.

### 14.2.5 Overload frame

An overload frame is transmitted under the following conditions.

- When the receiving node has not completed the reception operation
- If a dominant level is detected at the first two bits during intermission
- If a dominant level is detected at the last bit (7th bit) of the end of frame or at the last bit (8th bit) of the error delimiter/overload delimiter

**Note** The CAN is internally fast enough to process all received frames not generating overload frames.

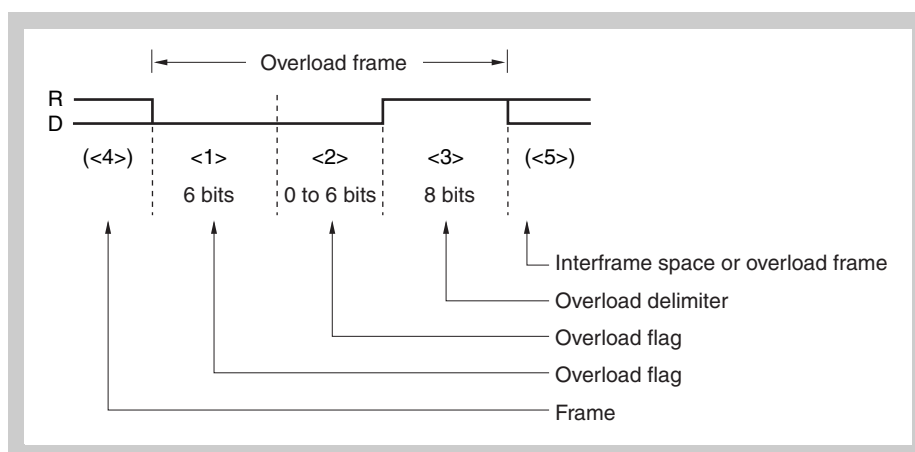


Figure 14-16 Overload frame

**Note** D: Dominant = 0  
R: Recessive = 1

Table 14-8 Definition of overload frame fields

No	Name	Bit count	Definition
<1>	Overload flag	6	Outputs 6 dominant-level bits consecutively.
<2>	Overload flag from other node	0 to 6	The node that received an overload flag in the interframe space outputs an overload flag.
<3>	Overload delimiter	8	Outputs 8 recessive-level bits consecutively. If a dominant level is detected at the 8th bit, an overload frame is transmitted from the next bit.
<4>	Frame	—	Output following an end of frame, error delimiter, or overload delimiter.
<5>	Interframe space/overload frame	—	An interframe space or overload frame starts from here.



## 14.3 Functions

### 14.3.1 Determining bus priority

**(1) When a node starts transmission:**

- During bus idle, the node that output data first transmits the data.

**(2) When more than one node starts transmission:**

- The node that consecutively outputs the dominant level for the longest from the first bit of the arbitration field has the bus priority (if a dominant level and a recessive level are simultaneously transmitted, the dominant level is taken as the bus value).
- The transmitting node compares its output arbitration field and the data level on the bus.

**Table 14-9 Determining bus priority**

Level match	Continuous transmission
Level mismatch	Stops transmission at the bit where mismatch is detected and starts reception at the following bit

**(3) Priority of data frame and remote frame**

- When a data frame and a remote frame are on the bus, the data frame has priority because its RTR bit, the last bit in the arbitration field, carries a dominant level.

**Note** If the extended-format data frame and the standard-format remote frame conflict on the bus (if ID28 to ID18 of both of them are the same), the standard-format remote frame takes priority.

### 14.3.2 Bit stuffing

Bit stuffing is used to establish synchronization by appending 1 bit of inverted-level data if the same level continues for 5 bits, in order to prevent a burst error.

**Table 14-10 Bit stuffing**

Transmission	During the transmission of a data frame or remote frame, when the same level continues for 5 bits in the data between the start of frame and the ACK field, 1 inverted-level bit of data is inserted before the following bit.
Reception	During the reception of a data frame or remote frame, when the same level continues for 5 bits in the data between the start of frame and the ACK field, reception is continued after deleting the next bit.

### 14.3.3 Multi masters

As the bus priority (a node acquiring transmit functions) is determined by the identifier, any node can be the bus master.

### 14.3.4 Multi cast

Although there is one transmitting node, two or more nodes can receive the same data at the same time because the same identifier can be set to two or more nodes.

### 14.3.5 CAN sleep mode/CAN stop mode function

The CAN sleep mode/CAN stop mode function puts the CAN Controller in waiting mode to achieve low power consumption.

The controller is woken up from the CAN sleep mode by bus operation but it is not woken up from the CAN stop mode by bus operation (the CAN stop mode is controlled by CPU access).

### 14.3.6 Error control function

#### (1) Error types

Table 14-11 Error types

Type	Description of error		Detection state	
	Detection method	Detection condition	Transmission/reception	Field/frame
Bit error	Comparison of the output level and level on the bus (except stuff bit)	Mismatch of levels	Transmitting/receiving node	Bit that is outputting data on the bus at the start of frame to end of frame, error frame and overload frame.
Stuff error	Check of the receive data at the stuff bit	6 consecutive bits of the same output level	Receiving node	Start of frame to CRC sequence
CRC error	Comparison of the CRC sequence generated from the receive data and the received CRC sequence	Mismatch of CRC	Receiving node	CRC field
Form error	Field/frame check of the fixed format	Detection of fixed format violation	Receiving node	CRC delimiter ACK field End of frame Error frame Overload frame
ACK error	Check of the ACK slot by the transmitting node	Detection of recessive level in ACK slot	Transmitting node	ACK slot

**(2) Output timing of error frame****Table 14-12 Output timing of error frame**

Type	Output timing
Bit error, stuff error, form error, ACK error	Error frame output is started at the timing of the bit following the detected error.
CEC error	Error frame output is started at the timing of the bit following the ACK delimiter.

**(3) Processing in case of error**

The transmission node re-transmits the data frame or remote frame after the error frame. (However, it does not re-transmit the frame in the single-shot mode.)

**(4) Error state****(a) Types of error states**

The following three types of error states are defined by the CAN specification:

- Error active
- Error passive
- Bus-off

These types of error states are classified by the values of the TEC7 to TEC0 bits (transmission error counter bits) and the REC6 to REC0 bits (reception error counter bits) as shown in *Table 14-13*.

The present error state is indicated by the CAN module information register (CnINFO).

When each error counter value becomes equal to or greater than the error warning level (96), the TECS0 or RECS0 bit of the CnINFO register is set to 1. In this case, the bus state must be tested because it is considered that the bus has a serious fault. An error counter value of 128 or more indicates an error passive state and the TECS1 or RECS1 bit of the CnINFO register is set to 1.

- If the value of the transmission error counter is greater than or equal to 256 (actually, the transmission error counter does not indicate a value greater than or equal to 256), the bus-off state is reached and the BOFF bit of the CnINFO register is set to 1.
- If only one node is active on the bus at startup (i.e., a particular case such as when the bus is connected only to the local station), ACK is not returned even if data is transmitted. Consequently, re-transmission of the error frame and data is repeated. In the error passive state, however, the transmission error counter is not incremented and the bus-off state is not reached.

**Table 14-13 Types of error states**

**Note** The value of the transmission error counter (TEC) is invalid when the BOFF bit is set to 1. If an error that increments the value of the transmission error counter by +8 while the counter value is in a range of 248 to 255, the counter is not incremented and the bus-off state is assumed.

Type	Operation	Value of error counter	Indication of CnINFO register	Operation specific to error state
Error active	Transmission	0 to 95	TECS1, TECS0 = 00	Outputs an active error flag (6 consecutive dominant-level bits) on detection of the error.
	Reception	0 to 95	RECS1, RECS0 = 00	
	Transmission	96 to 127	TECS1, TECS0 = 01	
	Reception	96 to 127	RECS1, RECS0 = 01	
Error passive	Transmission	128 to 255	TECS1, TECS0 = 11	Outputs a passive error flag (6 consecutive recessive-level bits) on detection of the error. Transmits 8 recessive-level bits, in between transmissions, following an intermission (suspend transmission).
	Reception	128 or more	RECS1, RECS0 = 11	
Bus-off	Transmission	256 or more (not indicated) <sup>Note</sup>	BOFF = 1, TECS1, TECS0 = 11	Communication is not possible. Messages are not stored when receiving frames, however, the following operations of <1>, <2>, and <3> are done. <1> TSOUT toggles. <2> REC is incremented/decremented. <3> VALID bit is set. If the CAN module is entered to the initialization mode and then transition request to any operation mode is made, and when 11 consecutive recessive-level bits are detected 128 times, the error counter is reset to 0 and the error active state can be restored.

**(b) Error counter**

The error counter counts up when an error has occurred, and counts down upon successful transmission and reception. The error counter is updated immediately after error detection.

**(c) Occurrence of bit error in intermission**

An overload frame is generated.

---

**Caution** If an error occurs, it is controlled according to the contents of the transmission error counter and reception error counter before the error occurred. The value of the error counter is incremented after the error flag has been output.

---

**(5) Recovery from bus-off state**

When the CAN module is in the bus-off state, the CAN module permanently sets its output signals (CTXDn) to recessive level.

The CAN module recovers from the bus-off state in the following bus-off recovery sequence.

1. **A request to enter the CAN initialization mode**
2. **A request to enter a CAN operation mode**
  - (a) Recovery operation through normal recovery sequence
  - (b) Forced recovery operation that skips recovery sequence

**Table 14-14 Error counter**

State	Transmission error counter (TEC7 to TEC0 bits)	Reception error counter (REC6 to REC0 bits)
Receiving node detects an error (except bit error in the active error flag or overload flag).	No change	+1 (when REPS = 0)
Receiving node detects dominant level following error flag of error frame.	No change	+8 (when REPS = 0)
Transmitting node transmits an error flag. [As exceptions, the error counter does not change in the following cases.] <1> ACK error is detected in error passive state and dominant level is not detected while the passive error flag is being output. <2> A stuff error is detected in an arbitration field that transmitted a recessive level as a stuff bit, but a dominant level is detected.	+8	No change
Bit error detection while active error flag or overload flag is being output (error-active transmitting node)	+8	No change
Bit error detection while active error flag or overload flag is being output (error-active receiving node)	No change	+8 (REPS bit = 0)
When the node detects 14 consecutive dominant-level bits from the beginning of the active error flag or overload flag, and then subsequently detects 8 consecutive dominant-level bits. When the node detects 8 consecutive dominant levels after a passive error flag	+8 (transmitting)	+8 (during reception, when REPS = 0)
When the transmitting node has completed transmission without error ( $\pm 0$ if error counter = 0)	-1	No change
When the receiving node has completed reception without error	No change	<ul style="list-style-type: none"> <li>-1 (<math>1 \leq \text{REC6 to REC0} \leq 127</math>, when REPS = 0)</li> <li><math>\pm 0</math> (REC6 to REC0 = 0, when REPS = 0)</li> <li>Value of 119 to 127 is set (when REPS = 1)</li> </ul>

**(a) Recovery from bus-off state through normal recovery sequence**

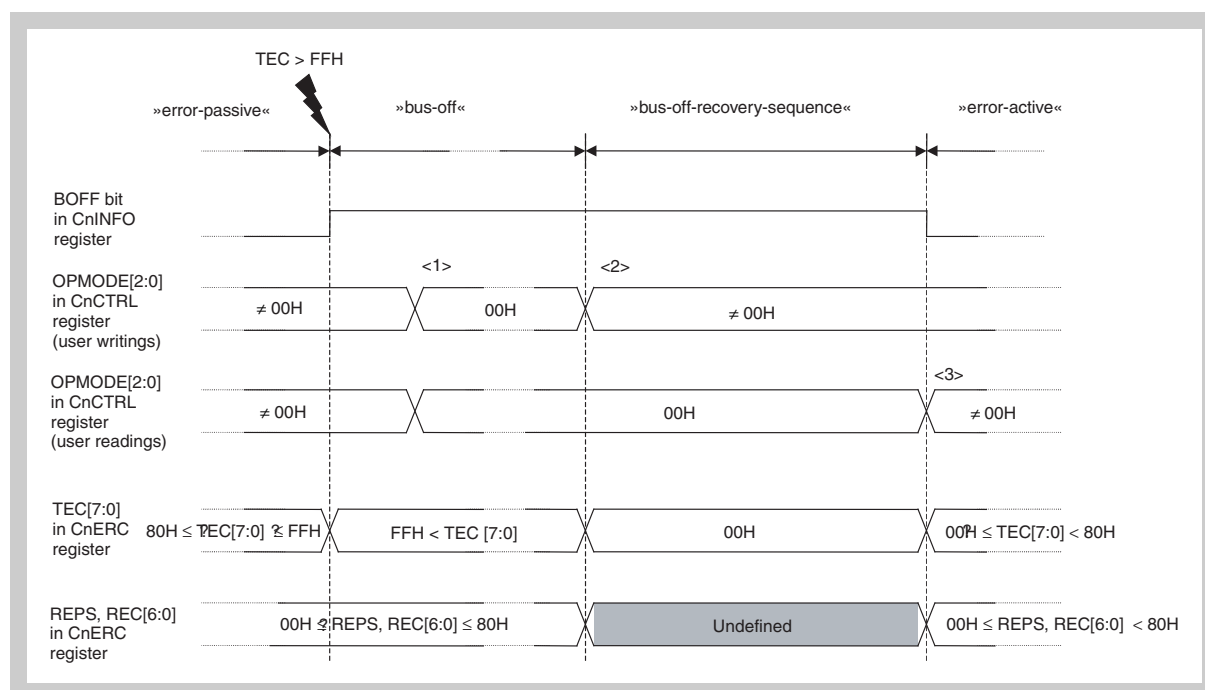
The CAN module first issues a request to enter the initialization mode (refer too timing <1> in *Figure 14-17 on page 387*). This request will be immediately acknowledged, and the OPMODE bits of the CnCTRL register are cleared to 000<sub>B</sub>. Processing such as analyzing the fault that has caused the bus-off state, re-defining the CAN module and message buffer using application software, or stopping the operation of the CAN module can be performed by clearing the GOM bit to 0.

Next, the module requests to change the mode from the initialization mode to an operation mode (refer to timing <2> in *Figure 14-17 on page 387*). This starts an operation to recover the CAN module from the bus-off state. The conditions under which the module can recover from the bus-off state are defined by the CAN protocol ISO 11898, and it is necessary to detect 11 consecutive recessive-level bits 128 times. At this time, the request to change the mode to an operation mode is held pending until the recovery conditions

are satisfied. When the recovery conditions are satisfied (refer to timing <3> in *Figure 14-17 on page 387*), the CAN module can enter the operation mode it has requested. Until the CAN module enters this operation mode, it stays in the initialization mode. Completion to be requested operation mode can be confirmed by reading the OPMODE bits of the CnCTRL register.

During the bus-off period and bus-off recovery sequence, the BOFF bit of the CnINFO register stays set (to 1). In the bus-off recovery sequence, the reception error counter (REC[6:0]) counts the number of times 11 consecutive recessive-level bits have been detected on the bus. Therefore, the recovery state can be checked by reading REC[6:0].

**Caution** In the bus-off recovery sequence, REC[6:0] counts up (+1) each time 11 consecutive recessive-level bits have been detected. Even during the bus-off period, the CAN module can enter the CAN sleep mode or CAN stop mode. To start the bus-off recovery sequence, it is necessary to transit to the initialization mode once. However, when the CAN module is in either CAN sleep mode or CAN stop mode, transition request to the initialization mode is not accepted, thus you have to release the CAN sleep mode first. In this case, as soon as the CAN sleep mode is released, the bus-off recovery sequence starts and no transition to initialization mode is necessary. If the can module detects a dominant edge on the CAN bus while in sleep mode even during bus-off, the sleep mode will be left and the bus-off recovery sequence will start.



**Figure 14-17** Recovery from bus-off state through normal recovery sequence

#### (b) Forced recovery operation that skips bus-off recovery sequence

The CAN module can be forcibly released from the bus-off state, regardless of the bus state, by skipping the bus-off recovery sequence. Here is the procedure.

First, the CAN module requests to enter the initialization mode. For the operation and points to be noted at this time, “*Recovery from bus-off state through normal recovery sequence*” on page 385.

Next, the module requests to enter an operation mode. At the same time, the CCERC bit of the CnCTRL register must be set to 1.

As a result, the bus-off recovery sequence defined by the CAN protocol ISO 11898 is skipped, and the module immediately enters the operation mode. In this case, the module is connected to the CAN bus after it has monitored 11 consecutive recessive-level bits. For details, refer to the processing in *Figure 14-55* on page 500.

---

**Caution** This function is not defined by the CAN protocol ISO 11898. When using this function, thoroughly evaluate its effect on the network system.

---

**(6) Initializing CAN module error counter register (CnERC) in initialization mode**

If it is necessary to initialize the CAN module error counter register (CnERC) and CAN module information register (CnINFO) for debugging or evaluating a program, they can be initialized to the default value by setting the CCERC bit of the CnCTRL register in the initialization mode. When initialization has been completed, the CCERC bit is automatically cleared to 0.

- 
- Caution**
1. This function is enabled only in the initialization mode. Even if the CCERC bit is set to 1 in a CAN operation mode, the CnERC and CnINFO registers are not initialized.
  2. The CCERC bit can be set at the same time as the request to enter a CAN operation mode.
-



### 14.3.7 Baud rate control function

#### (1) Prescaler

The CAN controller has a prescaler that divides the clock ( $f_{CAN}$ ) supplied to CAN. This prescaler generates a CAN protocol layer basic system clock ( $f_{TQ}$ ) derived from the CAN module system clock ( $f_{CANMOD}$ ), and divided by 1 to 256 (“CnBRP - CANn module bit rate prescaler register” on page 420).

#### (2) Data bit time (8 to 25 time quanta)

One data bit time is defined as shown in *Figure 14-18 on page 389*.

The CAN Controller sets time segment 1, time segment 2, and reSynchronization Jump Width (SJW) of data bit time, as shown in *Figure 14-18*. Time segment 1 is equivalent to the total of the propagation (prop) segment and phase segment 1 that are defined by the CAN protocol specification. Time segment 2 is equivalent to phase segment 2.

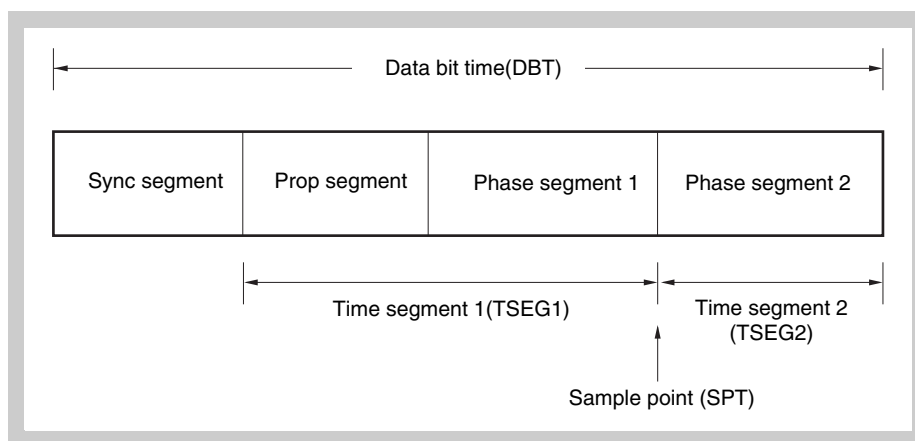


Figure 14-18 Segment setting

Table 14-15 Segment setting

Segment name	Settable range	Notes on setting to conform to CAN specification
Time segment 1 (TSEG1)	2TQ to 15TQ	-
Time segment 2 (TSEG2)	1TQ to 8TQ	IPT of the CAN controller is 0TQ. To conform to the CAN protocol specification, therefore, a length less or equal to phase segment 1 must be set here. This means that the length of time segment 1 minus 1TQ is the settable upper limit of time segment 2.
Resynchronization Jump Width (SJW)	1TQ to 4TQ	The length of time segment 1 minus 1TQ or 4 TQ, whichever is smaller.

- Note**
1. IPT: Information Processing Time
  2. TQ: Time Quanta

Reference: The CAN protocol specification defines the segments constituting the data bit time as shown in *Figure 14-19*.

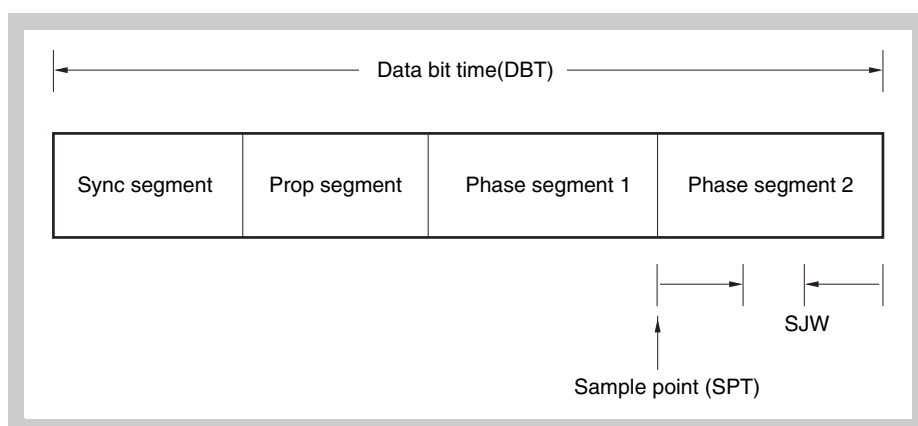


Figure 14-19 Configuration of data bit time defined by CAN specification

Table 14-16 Configuration of data bit time defined by CAN specification

Segment name	Settable range	Notes on setting to conform to CAN specification
Sync segment (Synchronization segment)	1	This segment starts at the edge where the level changes from recessive to dominant when hardware synchronization is established.
Prop segment	Programmable to 1 to 8 or more	This segment absorbs the delay of the output buffer, CAN bus, and input buffer.
Phase segment 1	Programmable to 1 to 8	The length of this segment is set so that ACK is returned before the start of phase segment 1. Time of prop segment $\geq$ (Delay of output buffer) + $2 \times$ (Delay of CAN bus) + (Delay of input buffer) This segment compensates for an error of data bit time. The longer this segment, the wider the permissible range but the slower the communication speed.
Phase segment 2	Phase segment 1 or IPT, whichever greater	
SJW	Programmable from 1TQ to length of segment 1 or 4TQ, whichever is smaller	This width sets the upper limit of expansion or contraction of the phase segment during resynchronization.

**Note** IPT: Information Processing Time

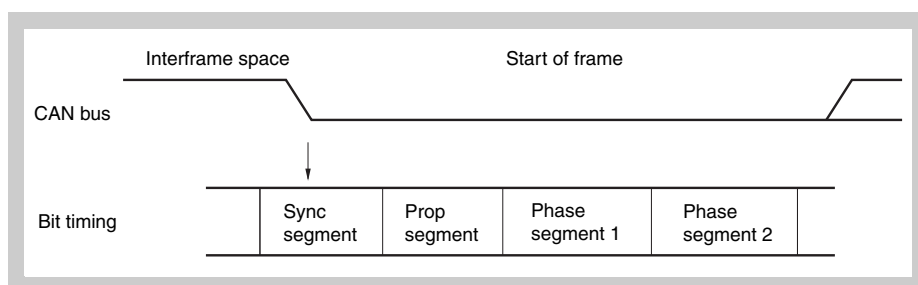
**(3) Synchronizing data bit**

- The receiving node establishes synchronization by a level change on the bus because it does not have a sync signal.
- The transmitting node transmits data in synchronization with the bit timing of the transmitting node.

**(a) Hardware synchronization**

This synchronization is established when the receiving node detects the start of frame in the interframe space.

- When a falling edge is detected on the bus, that TQ means the sync segment and the next segment is the prop segment. In this case, synchronization is established regardless of SJW.



**Figure 14-20 Adjusting synchronization of data bit**

**(b) Resynchronization**

Synchronization is established again if a level change is detected on the bus during reception (only if a recessive level was sampled previously).

- The phase error of the edge is given by the relative position of the detected edge and sync segment.

<Sign of phase error>

0: If the edge is within the sync segment

Positive: If the edge is before the sample point (phase error)

Negative: If the edge is after the sample point (phase error)

If phase error is positive: Phase segment 1 is lengthened by specified SJW.

If phase error is negative: Phase segment 2 is shortened by specified SJW.

- The sample point of the data of the receiving node moves relatively due to the “discrepancy” in the baud rate between the transmitting node and receiving node.

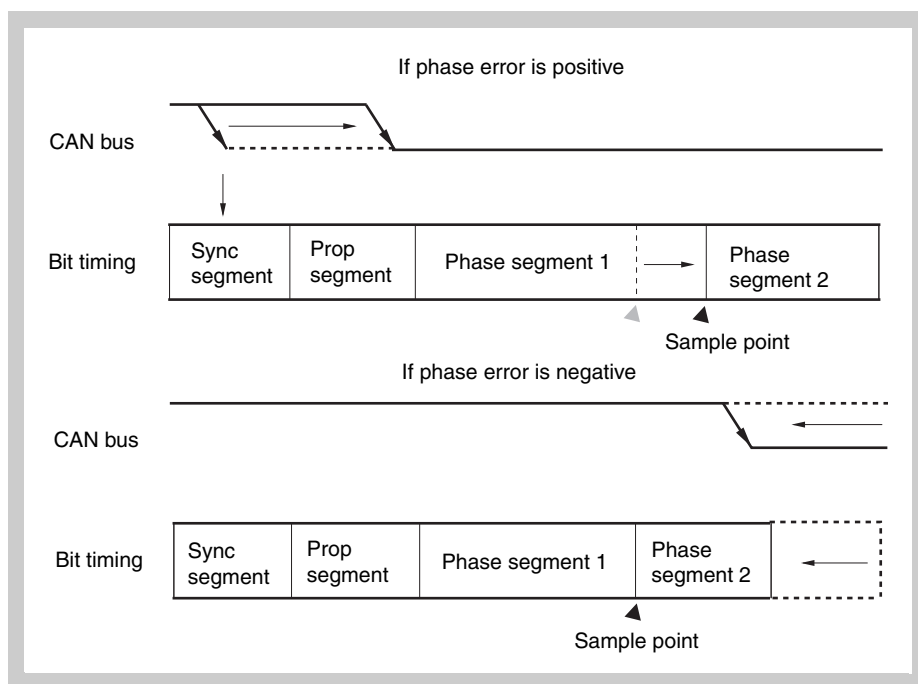


Figure 14-21 Resynchronization

## 14.4 Connection with Target System

The CAN module has to be connected to the CAN bus using an external transceiver.

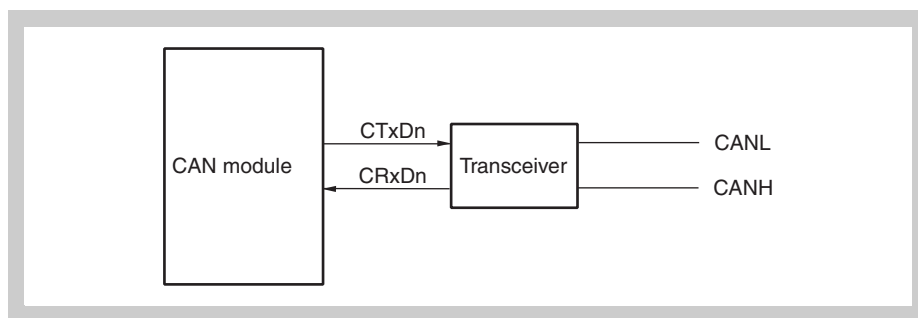


Figure 14-22 Connection to CAN bus

## 14.5 Internal Registers of CAN Controller

### 14.5.1 CAN module register and message buffer addresses

In this chapter all register and message buffer addresses are defined as address offsets to different base addresses.

Since all registers are accessed via the programmable peripheral area the bottom address is defined by the BPC register (refer to “*Programmable peripheral I/O area*” on page 90 or to “*Programmable peripheral I/O area (PPA)*” on page 197).

The addresses given in the following tables are offsets to the programmable peripheral area base address PBA.

The recommended setting of BPC is A100<sub>H</sub>. This setting would define the programmable peripheral area base address

$$PBA = 0840\ 0000_H$$

Table 14-17 lists all base addresses used throughout this chapter.

**Table 14-17 CAN module base addresses**

Base address name	Base address of	Address	Address for BPC = A100 <sub>H</sub>
C0RBaseAddr	CAN0 registers	PBA + 000 <sub>H</sub>	0840 0000 <sub>H</sub>
C0MBaseAddr	CAN0 message buffers	PBA + 100 <sub>H</sub>	0840 0100 <sub>H</sub>
C1RBaseAddr	CAN1 registers	PBA + 600 <sub>H</sub>	0840 0600 <sub>H</sub>
C1MBaseAddr	CAN1 message buffers	PBA + 700 <sub>H</sub>	0840 0700 <sub>H</sub>

In the following <CnRBaseAddr> respectively <CnMBaseAddr> are used for the base address names for CAN channel n.

## 14.5.2 CAN Controller configuration

**Table 14-18 List of CAN Controller registers**

Item	Register Name
CANn global registers	CANn global control register (CnGMCTRL)
	CANn global clock selection register (CnGMCS)
	CANn global automatic block transmission control register (CnGMABT)
	CANn global automatic block transmission delay setting register (CnGMABTD)
CANn module registers	CANn module mask 1 register (CnMASK1L, CnMASK1H)
	CANn module mask 2 register (CnMASK2L, CnMASK2H)
	CANn module mask3 register (CnMASK3L, CnMASK3H)
	CANn module mask 4 registers (CnMASK4L, CnMASK4H)
	CANn module control register (CnCTRL)
	CANn module last error information register (CnLEC)
	CANn module information register (CnINFO)
	CANn module error counter register (CnERC)
	CANn module interrupt enable register (CnIE)
	CANn module interrupt status register (CnINTS)
	CANn module bit rate prescaler register (CnBRP)
	CANn module bit rate register (CnBTR)
	CANn module last in-pointer register (CnLIPT)
	CANn module receive history list register (CnRGPT)
	CANn module last out-pointer register (CnLOPT)
	CANn module transmit history list register (CnTGPT)
	CANn module time stamp register (CnTS)
CANn message buffer registers	CANn message data byte 01 register m (CnMDATA01m)
	CANn message data byte 0 register m (CnMDATA0m)
	CANn message data byte 1 register m (CnMDATA1m)
	CANn message data byte 23 register m (CnMDATA23m)
	CANn message data byte 2 register m (CnMDATA2m)
	CANn message data byte 3 register m (CnMDATA3m)
	CANn message data byte 45 register m (CnMDATA45m)
	CANn message data byte 4 register m (CnMDATA4m)
	CANn message data byte 5 register m (CnMDATA5m)
	CANn message data byte 67 register m (CnMDATA67m)
	CANn message data byte 6 register m (CnMDATA6m)
	CANn message data byte 7 register m (CnMDATA7m)
	CANn message data length register m (CnMDLCm)
	CANn message configuration register m (CnMCONFm)
	CANn message ID register m (CnMIDLm, CnMIDHm)
	CANn message control register m (CnMCTRLm)

### 14.5.3 CAN registers overview

#### (1) CANn global and module registers

The following table lists the address offsets to the CANn register base address:

CnRBaseAddr.

**Table 14-19 CANn global and module registers**

Address offset	Register name	Symbol	R/W	Access			After reset
				1-bit	8-bit	16-bit	
000 <sub>H</sub>	CANn global control register	CnGMCTRL	R/W	–	–	√	0000 <sub>H</sub>
002 <sub>H</sub>	CANn global clock selection register	CnGMCS		–	√	–	0F <sub>H</sub>
006 <sub>H</sub>	CANn global automatic block transmission register	CnGMABT		–	–	√	0000 <sub>H</sub>
008 <sub>H</sub>	CANn global automatic block transmission delay register	CnGMABTD		–	√	–	00 <sub>H</sub>
040 <sub>H</sub>	CANn module mask 1 register	CnMASK1L		–	–	√	Undefined
042 <sub>H</sub>		CnMASK1H		–	–	√	Undefined
044 <sub>H</sub>	CANn module mask 2 register	CnMASK2L		–	–	√	Undefined
046 <sub>H</sub>		CnMASK2H		–	–	√	Undefined
048 <sub>H</sub>	CANn module mask 3 register	CnMASK3L		–	–	√	Undefined
04A <sub>H</sub>		CnMASK3H		–	–	√	Undefined
04C <sub>H</sub>	CANn module mask 4 register	CnMASK4L		–	–	√	Undefined
04E <sub>H</sub>		CnMASK4H		–	–	√	Undefined
050 <sub>H</sub>	CANn module control register	CnCTRL		–	–	√	0000 <sub>H</sub>
052 <sub>H</sub>	CANn module last error code register	CnLEC		–	√	–	00 <sub>H</sub>
053 <sub>H</sub>	CANn module information register	CnINFO	R	–	√	–	00 <sub>H</sub>
054 <sub>H</sub>	CANn module error counter register	CnERC		–	–	√	0000 <sub>H</sub>
056 <sub>H</sub>	CANn module interrupt enable register	CnIE	R/W	–	–	√	0000 <sub>H</sub>
058 <sub>H</sub>	CANn module interrupt status register	CnINTS		–	–	√	0000 <sub>H</sub>
05A <sub>H</sub>	CANn module bit-rate prescaler register	CnBRP		–	√	–	FF <sub>H</sub>
05C <sub>H</sub>	CANn module bit-rate register	CnBTR		–	–	√	370F <sub>H</sub>
05E <sub>H</sub>	CANn module last in-pointer register	CnLIPT	R	–	√	–	Undefined
060 <sub>H</sub>	CANn module receive history list register	CnRGPT	R/W	–	–	√	xx02 <sub>H</sub>
062 <sub>H</sub>	CANn module last out-pointer register	CnLOPT	R	–	√	–	Undefined
064 <sub>H</sub>	CANn module transmit history list register	CnTGPT	R/W	–	–	√	xx02 <sub>H</sub>
066 <sub>H</sub>	CANn module time stamp register	CnTS		–	–	√	0000 <sub>H</sub>

**(2) CANn message buffer registers**

The addresses in the following table denote the address offsets to the CANn message buffer base address:

CnMBaseAddr.

**Example** CAN0, message buffer register  $m = 14 = E_H$ , byte 6 COMDATA614 has the address  $E_H \times 20_H + 6_H + \text{COMBaseAddr}$

**Note** The message buffer register number  $m$  in the register symbols has 2 digits, for example,  
COMDATA01m = COMDATA0100 for  $m = 0$ .

**Table 14-20 CANn message buffer registers**

Address offset	Register name	Symbol	R/W	Access			After reset
				1-bit	8-bit	16-bit	
$mx20_H + 0_H$	CANn message data byte 01 register m	CnMDATA01m	R/W	–	–	✓	Undefined
$mx20_H + 0_H$	CANn message data byte 0 register m	CnMDATA0m		–	✓	–	Undefined
$mx20_H + 1_H$	CANn message data byte 1 register m	CnMDATA1m		–	✓	–	Undefined
$mx20_H + 2_H$	CANn message data byte 23 register m	CnMDATA23m		–	–	✓	Undefined
$mx20_H + 2_H$	CANn message data byte 2 register m	CnMDATA2m		–	✓	–	Undefined
$mx20_H + 3_H$	CANn message data byte 3 register m	CnMDATA3m		–	✓	–	Undefined
$mx20_H + 4_H$	CANn message data byte 45 register m	CnMDATA45m		–	–	✓	Undefined
$mx20_H + 4_H$	CANn message data byte 4 register m	CnMDATA4m		–	✓	–	Undefined
$mx20_H + 5_H$	CANn message data byte 5 register m	CnMDATA5m		–	✓	–	Undefined
$mx20_H + 6_H$	CANn message data byte 67 register m	CnMDATA67m		–	–	✓	Undefined
$mx20_H + 6_H$	CANn message data byte 6 register m	CnMDATA6m		–	✓	–	Undefined
$mx20_H + 7_H$	CANn message data byte 7 register m	CnMDATA7m		–	✓	–	Undefined
$mx20_H + 8_H$	CANn message data length register m	CnMDLCm		–	✓	–	0000 xxxx <sub>B</sub>
$mx20_H + 9_H$	CANn message configuration register m	CnMCONFm		–	✓	–	Undefined
$mx20_H + A_H$	CANn message identifier register m	CnMIDLm		–	–	✓	Undefined
$mx20_H + C_H$		CnMIDHm		–	–	✓	Undefined
$mx20_H + E_H$	CANn message control register m	CnMCTRLm		–	–	✓	0x00 0000 0000 0000 <sub>B</sub>



## 14.5.4 Register bit configuration

**Table 14-21 CAN global register bit configuration**

Address offset <sup>a</sup>	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
00 <sub>H</sub>	CnGMCTRL (W)	0	0	0	0	0	0	0	Clear GOM
01 <sub>H</sub>		0	0	0	0	0	0	Set EFSD	Set GOM
00 <sub>H</sub>	CnGMCTRL (R)	0	0	0	0	0	0	EFSD	GOM
01 <sub>H</sub>		MBON	0	0	0	0	0	0	0
02 <sub>H</sub>	CnGMCS	0	0	0	0	CCP3	CCP2	CCP1	CCP0
06 <sub>H</sub>	CnGMABT (W)	0	0	0	0	0	0	0	Clear ABTTRG
07 <sub>H</sub>		0	0	0	0	0	0	Set ABTCLR	Set ABTTRG
06 <sub>H</sub>	CnGMABT (R)	0	0	0	0	0	0	ABTCLR	ABTTRG
07 <sub>H</sub>		0	0	0	0	0	0	0	0
08 <sub>H</sub>	CnGMABTD	0	0	0	0	ABTD3	ABTD2	ABTD1	ABTD0

a) Base address: <CnRBaseAddr>

**Table 14-22 CAN module register bit configuration (1/2)**

Address offset <sup>a</sup>	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
40 <sub>H</sub>	CnMASK1L	CMID7 to CMID0							
41 <sub>H</sub>		CMID15 to CMID8							
42 <sub>H</sub>	CnMASK1H	CMID23 to CMID16							
43 <sub>H</sub>		0	0	0	CMID28 to CMID24				
44 <sub>H</sub>	CnMASK2L	CMID7 to CMID0							
45 <sub>H</sub>		CMID15 to CMID8							
46 <sub>H</sub>	CnMASK2H	CMID23 to CMID16							
47 <sub>H</sub>		0	0	0	CMID28 to CMID24				
48 <sub>H</sub>	CnMASK3L	CMID7 to CMID0							
49 <sub>H</sub>		CMID15 to CMID8							
4A <sub>H</sub>	CnMASK3H	CMID23 to CMID16							
4B <sub>H</sub>		0	0	0	CMID28 to CMID24				
4C <sub>H</sub>	CnMASK4L	CMID7 to CMID0							
4D <sub>H</sub>		CMID15 to CMID8							
4E <sub>H</sub>	CnMASK4H	CMID23 to CMID16							
4F <sub>H</sub>		0	0	0	CMID28 to CMID24				
50 <sub>H</sub>	CnCTRL (W)	0	Clear AL	Clear VALID	Clear PSMODE1	Clear PSMODE0	Clear OPMODE2	Clear OPMODE1	Clear OPMODE0
51 <sub>H</sub>		Set CCERC	Set AL	0	Set PSMODE1	Set PSMODE0	Set OPMODE2	Set OPMODE1	Set OPMODE0
50 <sub>H</sub>	CnCTRL (R)	CCERC	AL	VALID	PS MODE1	PS MODE0	OP MODE2	OP MODE1	OP MODE0
51 <sub>H</sub>		0	0	0	0	0	0	RSTAT	TSTAT

Table 14-22 CAN module register bit configuration (2/2)

Address offset <sup>a</sup>	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
52 <sub>H</sub>	CnLEC (W)	0	0	0	0	0	0	0	0
52 <sub>H</sub>	CnLEC (R)	0	0	0	0	0	LEC2	LEC1	LEC0
53 <sub>H</sub>	CnINFO	0	0	0	BOFF	TECS1	TECS0	RECS1	RECS0
54 <sub>H</sub>	CnERC	TEC7 to TEC0							
55 <sub>H</sub>		REC7 to REC0							
56 <sub>H</sub>	CnIE (W)	0	0	Clear CIE5	Clear CIE4	Clear CIE3	Clear CIE2	Clear CIE1	Clear CIE0
57 <sub>H</sub>		0	0	Set CIE5	Set CIE4	Set CIE3	Set CIE2	Set CIE1	Set CIE0
56 <sub>H</sub>	CnIE (R)	0	0	CIE5	CIE4	CIE3	CIE2	CIE1	CIE0
57 <sub>H</sub>		0	0	0	0	0	0	0	0
58 <sub>H</sub>	CnINTS (W)	0	0	Clear CINTS5	Clear CINTS4	Clear CINTS3	Clear CINTS2	Clear CINTS1	Clear CINTS0
59 <sub>H</sub>		0	0	0	0	0	0	0	0
58 <sub>H</sub>	CnINTS (R)	0	0	CINTS5	CINTS4	CINTS3	CINTS2	CINTS1	CINTS0
59 <sub>H</sub>		0	0	0	0	0	0	0	0
5A <sub>H</sub>	CnBRP	TQPRS7 to TQPRS0							
5C <sub>H</sub>	CnBTR	0	0	0	0	TSEG13 to TSEG10			
5D <sub>H</sub>		0	0	SJW1, SJW0		0	TSEG22 to TSEG20		
5E <sub>H</sub>	CnLIPT	LIPT7 to LIPT0							
60 <sub>H</sub>	CnRGPT (W)	0	0	0	0	0	0	0	Clear ROVF
61 <sub>H</sub>		0	0	0	0	0	0	0	0
60 <sub>H</sub>	CnRGPT (R)	0	0	0	0	0	0	RHPM	ROVF
61 <sub>H</sub>		RGPT7 to RGPT0							
F62 <sub>H</sub>	CnLOPT	LOPT7 to LOPT0							
64 <sub>H</sub>	CnTGPT (W)	0	0	0	0	0	0	0	Clear TOVF
65 <sub>H</sub>		0	0	0	0	0	0	0	0
64 <sub>H</sub>	CnTGPT (R)	0	0	0	0	0	0	THPM	TOVF
65 <sub>H</sub>		TGPT7 to TGPT0							
66 <sub>H</sub>	CnTS (W)	0	0	0	0	0	Clear TSLOCK	Clear TSSEL	Clear TSEN
67 <sub>H</sub>		0	0	0	0	0	Set TSLOCK	Set TSSEL	Set TSEN
66 <sub>H</sub>	CnTS (R)	0	0	0	0	0	TSLOCK	TSSEL	TSEN
67 <sub>H</sub>		0	0	0	0	0	0	0	0
68 <sub>H</sub> to FF <sub>H</sub>	-	Access prohibited (reserved for future use)							

a) Base address: &lt;CnRBaseAddr&gt;

**Table 14-23 Message buffer register bit configuration**

Address offset <sup>a</sup>	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
0 <sub>H</sub>	CnMDATA01m	Message data (byte 0)							
1 <sub>H</sub>		Message data (byte 1)							
0 <sub>H</sub>	CnMDATA0m	Message data (byte 0)							
1 <sub>H</sub>	CnMDATA1m	Message data (byte 1)							
2 <sub>H</sub>	CnMDATA23m	Message data (byte 2)							
3 <sub>H</sub>		Message data (byte 3)							
2 <sub>H</sub>	CnMDATA2m	Message data (byte 2)							
3 <sub>H</sub>	CnMDATA3m	Message data (byte 3)							
4 <sub>H</sub>	CnMDATA45m	Message data (byte 4)							
5 <sub>H</sub>		Message data (byte 5)							
4 <sub>H</sub>	CnMDATA4m	Message data (byte 4)							
5 <sub>H</sub>	CnMDATA5m	Message data (byte 5)							
6 <sub>H</sub>	CnMDATA67m	Message data (byte 6)							
7 <sub>H</sub>		Message data (byte 7)							
6 <sub>H</sub>	CnMDATA6m	Message data (byte 6)							
7 <sub>H</sub>	CnMDATA7m	Message data (byte 7)							
8 <sub>H</sub>	CnMDLCm	0				MDLC3	MDLC2	MDLC1	MDLC0
9 <sub>H</sub>	CnMCONFm	OWS	RTR	MT2	MT1	MT0	0	0	MA0
A <sub>H</sub>	CnMIDLm	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
B <sub>H</sub>		ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
C <sub>H</sub>	CnMIDHm	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16
D <sub>H</sub>		IDE	0	0	ID28	ID27	ID26	ID25	ID24
E <sub>H</sub>	CnMCTRLm (W)	0	0	0	Clear MOW	Clear IE	Clear DN	Clear TRQ	Clear RDY
F <sub>H</sub>		0	0	0	0	Set IE	0	Set TRQ	Set RDY
E <sub>H</sub>	CnMCTRLm (R)	0	0	0	MOW	IE	DN	TRQ	RDY
F <sub>H</sub>		0	0	MUC	0	0	0	0	0

a) Base address: <CnMBaseAddr>

**Note** For calculation of the complete message buffer register addresses refer to “CAN registers overview” on page 395.

## 14.6 Bit Set/Clear Function

The CAN control registers include registers whose bits can be set or cleared via the CPU and via the CAN interface. An operation error occurs if the following registers are written directly. Do not write any values directly via bit manipulation, read/modify/write, or direct writing of target values.

- CANn global control register (CnGMCTRL)
- CANn global automatic block transmission control register (CnGMABT)
- CANn module control register (CnCTRL)
- CANn module interrupt enable register (CnIE)
- CANn module interrupt status register (CnINTS)
- CANn module receive history list register (CnRGPT)
- CANn module transmit history list register (CnTGPT)
- CANn module time stamp register (CnTS)
- CANn message control register (CnMCTRLm)

All the 16 bits in the above registers can be read via the usual method. Use the procedure described in *Figure 14-23* below to set or clear the lower 8 bits in these registers.

Setting or clearing of lower 8 bits in the above registers is performed in combination with the higher 8 bits (refer to the bit status after set/clear operation is specified in *Figure 14-26*). *Figure 14-23* shows how the values of set bits or clear bits relate to set/clear/no change operations in the corresponding register.

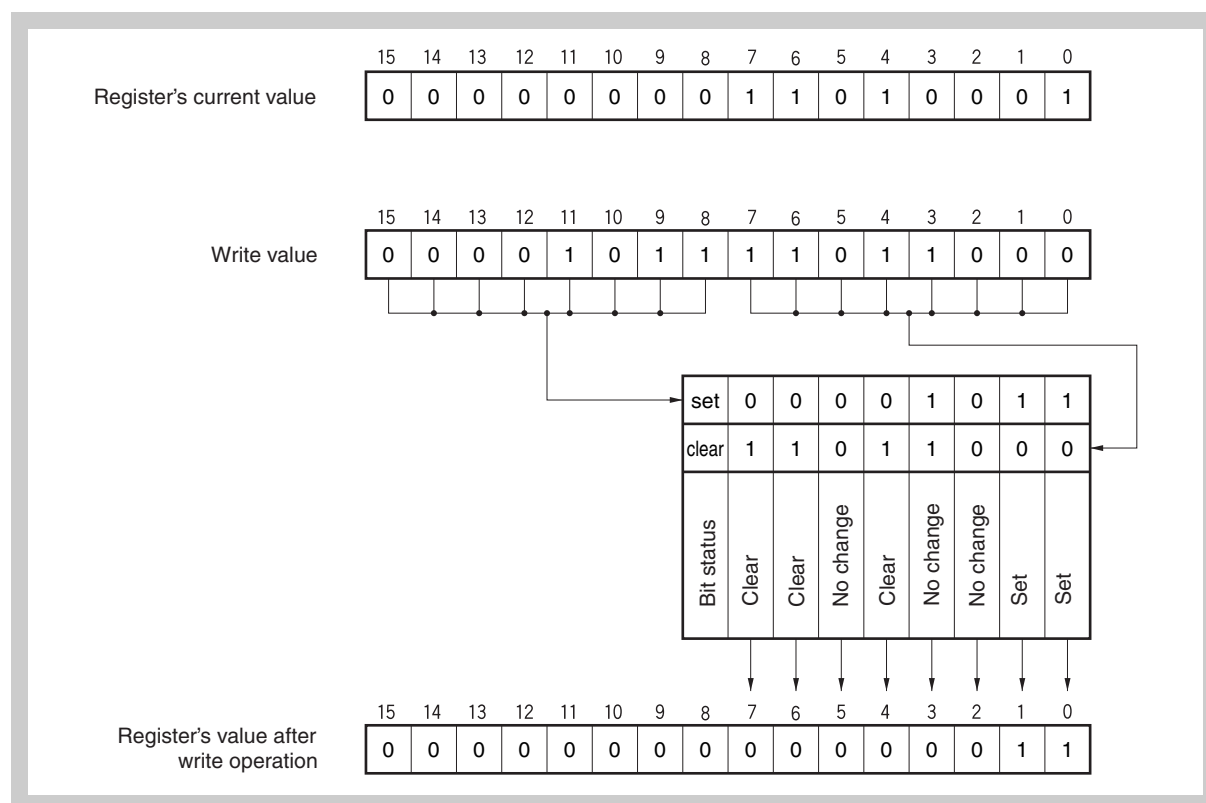


Figure 14-23 Example of bit setting/clearing operations

**(1) Bit status after bit setting/clearing operations**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Set 7	Set 6	Set 5	Set 4	Set 3	Set 2	Set 1	Set 0	Clear 7	Clear 6	Clear 5	Clear 4	Clear 3	Clear 2	Clear 1	Clear 0

Set 0 ... 7	Clear 0 ... 7	Status of bit n after bit set/clear operation
0	0	No change
0	1	0
1	0	1
1	1	No change

## 14.7 Control Registers

### (1) CnGMCTRL - CANn global control register

The CnGMCTRL register is used to control the operation of the CAN module.

**Access** This register can be read/written in 16-bit units.

**Address** <CnRBaseAddr> + 000<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. The register is initialized by any reset.

#### (a) CnGMCTRL read

15	14	13	12	11	10	9	8
MBON	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	0	0	0	0	EFSD	GOM

MBON	Bit enabling access to message buffer register, transmit/receive history registers
0	Write access and read access to the message buffer register and the transmit/receive history list registers is disabled.
1	Write access and read access to the message buffer register and the transmit/receive history list registers is enabled.

- Caution**
1. While the MBON bit is cleared (to 0), software access to the message buffers (CnMDATA0m, CnMDATA1m, CnMDATA01m, CnMDATA2m, CnMDATA3m, CnMDATA23m, CnMDATA4m, CnMDATA5m, CnMDATA45m, CnMDATA6m, CnMDATA7m, CnMDATA67m, CnMDLcM, CnMCONFm, CnMIDLm, CnMIDHm, and CnMCTRLm), or registers related to transmit history or receive history (CnLOPT, CnTGPT, CnLIPT, and CnRGPT) is disabled.
  2. This bit is read-only. Even if 1 is written to the MBON bit while it is 0, the value of the MBON bit does not change, and access to the message buffer registers, or registers related to transmit history or receive history remains disabled.

**Note** The MBON bit is cleared (to 0) when the CAN module enters CAN sleep mode/CAN stop mode, or when the GOM bit is cleared (to 0). The MBON bit is set (to 1) when the CAN sleep mode/CAN stop mode is released, or when the GOM bit is set (to 1).

EFSD	Bit enabling forced shut down
0	Forced shut down disabled.
1	Forced shut down enabled by subsequent clearing of GOM bit to 0.

- Caution**
1. To request forced shut down, the GOM bit must be cleared to 0 in a subsequent, immediately following access after the EFSD bit has been set to 1. If access to another register (including reading the CnGMCTRL register) is executed (even during NMI processing or DMAC operation) without clearing the GOM bit immediately after the EFSD bit has been set to 1, the EFSD bit is forcibly cleared to 0, and the forced shut down request is invalid.
  2. EFSD only works, if no continuous DMA transfer is performed.

GOM	Global operation mode bit
0	CAN module is disabled from operating.
1	CAN module is enabled to operate.

- Caution** The GOM can be cleared only in the initialization mode or immediately after EFSD bit is set (to 1).

#### (b) CnGMCTRL write

15	14	13	12	11	10	9	8
0	0	0	0	0	0	Set EFSD	Set GOM
7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	Clear GOM

Set EFSD	EFSD bit setting
0	No change in EFSD bit.
1	EFSD bit set to 1.

Set GOM	Clear GOM	GOM bit setting
0	1	GOM bit cleared to 0.
1	0	GOM bit set to 1.
Other than above		No change in GOM bit.

- Caution** Set the GOM bit and EFSD bit always separately.

**(2) CnGMCS - CANn global clock selection register**

The CnGMCS register is used to select the CAN module system clock.

**Access** This register can be read/written in 8-bit units.

**Address** <CnRBaseAddr> + 002<sub>H</sub>

**Initial Value** 0F<sub>H</sub>. The register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	CCP3	CCP2	CCP1	CCP0

CCP3	CCP2	CCP1	CCP1	CAN module system clock (f <sub>CANMOD</sub> )
0	0	0	0	f <sub>CAN</sub> /1
0	0	0	1	f <sub>CAN</sub> /2
0	0	1	0	f <sub>CAN</sub> /3
0	0	1	1	f <sub>CAN</sub> /4
0	1	0	0	f <sub>CAN</sub> /5
0	1	0	1	f <sub>CAN</sub> /6
0	1	1	0	f <sub>CAN</sub> /7
0	1	1	1	f <sub>CAN</sub> /8
1	0	0	0	f <sub>CAN</sub> /9
1	0	0	1	f <sub>CAN</sub> /10
1	0	1	0	f <sub>CAN</sub> /11
1	0	1	1	f <sub>CAN</sub> /12
1	1	0	0	f <sub>CAN</sub> /13
1	1	0	1	f <sub>CAN</sub> /14
1	1	1	0	f <sub>CAN</sub> /15
1	1	1	1	f <sub>CAN</sub> /16 (default value)

**Note** f<sub>CAN</sub> = clock supplied to CAN



**(3) CnGMABT - CANn global automatic block transmission control register**

The CnGMABT register is used to control the automatic block transmission (ABT) operation.

**Access** This register can be read/written in 16-bit units.

**Address** <CnRBaseAddr> + 006<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. The register is initialized by any reset.

**(a) CnGMABT read**

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	0	0	0	0	ABTCLR	ABTTRG

ABTCLR	Automatic block transmission engine clear status bit
0	Clearing the automatic transmission engine is completed.
1	The automatic transmission engine is being cleared.

- Note**
1. Set the ABTCLR bit to 1 while the ABTTRG bit is cleared to 0. The operation is not guaranteed if the ABTCLR bit is set to 1 while the ABTTRG bit is set to 1.
  2. When the automatic block transmission engine is cleared by setting the ABTCLR bit to 1, the ABTCLR bit is automatically cleared to 0 as soon as the requested clearing processing is complete.

ABTTRG	Automatic block transmission status bit
0	Automatic block transmission is stopped.
1	Automatic block transmission is under execution.

- Caution**
1. Do not set the ABTTRG bit (1) in the initialization mode. If the ABTTRG bit is set in the initialization mode, the operation is not guaranteed after the CAN module has entered the normal operation mode with ABT.
  2. Do not set the ABTTRG bit (1) while the CnCTRL.TSTAT bit is set (1). Confirm TSTAT = 0 directly in advance before setting ABTTRG bit.

**(b) CnGMABT write**

15	14	13	12	11	10	9	8
0	0	0	0	0	0	Set ABTCLR	Set ABTTRG
7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	Clear ABTTRG

**Caution** Before changing the normal operation mode with ABT to the initialization mode, be sure to set the CnGMABT register to the default value (0000<sub>H</sub>) and confirm the CnGMABT register is surely initialized to the default value (0000<sub>H</sub>).

Set ABTCLR	Automatic block transmission engine clear request bit
0	The automatic block transmission engine is in idle status or under operation.
1	Request to clear the automatic block transmission engine. After the automatic block transmission engine has been cleared, automatic block transmission is started from message buffer 0 by setting the ABTTRG bit to 1.

Set ABTTRG	Clear ABTTRG	Automatic block transmission start bit
0	1	Request to stop automatic block transmission.
1	0	Request to start automatic block transmission.
Other than above		No change in ABTTRG bit.

**(4) CnGMABTD - CANn global automatic block transmission delay register**

The CnGMABTD register is used to set the interval at which the data of the message buffer assigned to ABT is to be transmitted in the normal operation mode with ABT.

**Access** This register can be read/written in 8-bit units.

**Address** <CnRBaseAddr> + 008<sub>H</sub>

**Initial Value** 00<sub>H</sub>. The register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	ABTD3	ABTD2	ABTD1	ABTD0

ABTD3	ABTD2	ABTD1	ABTD0	Data frame interval during automatic block transmission in DBT <sup>a</sup>
0	0	0	0	0 DBT (default value)
0	0	0	1	2 <sup>5</sup> DBT
0	0	1	0	2 <sup>6</sup> DBT
0	0	1	1	2 <sup>7</sup> DBT
0	1	0	0	2 <sup>8</sup> DBT
0	1	0	1	2 <sup>9</sup> DBT
0	1	1	0	2 <sup>10</sup> DBT
0	1	1	1	2 <sup>11</sup> DBT
1	0	0	0	2 <sup>12</sup> DBT
Other than above				Setting prohibited

<sup>a)</sup> Unit: Data bit time (DBT)

- Caution**
1. Do not change the contents of the CnGMABTD register while the ABTTRG bit is set to 1.
  2. The timing at which the ABT message is actually transmitted onto the CAN bus differs depending on the status of transmission from the other station or how a request to transmit a message other than an ABT message (message buffers 8 to 31) is made.

**(5) CnMASKaL, CnMASKaH - CANn module mask control register (a = 1 to 4)**

The CnMASKaL and CnMASKaH registers are used to extend the number of receivable messages into the same message buffer by masking part of the identifier (ID) comparison of a message and invalidating the ID of the masked part.

**(a) CANn module mask 1 register (CnMASK1L, CnMASK1H)**

**Access** These registers can be read/written in 16-bit units.

**Address** CnMASK1L: <CnRBaseAddr> + 040<sub>H</sub>  
CnMASK1H: <CnRBaseAddr> + 042<sub>H</sub>

**Initial Value** Undefined.

CnMASK1L

15	14	13	12	11	10	9	8
CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
7	6	5	4	3	2	1	0
CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0

CnMASK1H

15	14	13	12	11	10	9	8
0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
7	6	5	4	3	2	1	0
CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

**(b) CANn module mask 2 register (CnMASK2L, CnMASK2H)**

**Access** These registers can be read/written in 16-bit units.

**Address** CnMASK2L: <CnRBaseAddr> + 044<sub>H</sub>  
CnMASK2H: <CnRBaseAddr> + 046<sub>H</sub>

**Initial Value** Undefined.

CnMASK2L

15	14	13	12	11	10	9	8
CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
7	6	5	4	3	2	1	0
CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0

CnMASK2H

15	14	13	12	11	10	9	8
0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
7	6	5	4	3	2	1	0
CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

**(c) CANn module mask 3 register (CnMASK3L, CnMASK3H)**

**Access** These registers can be read/written in 16-bit units.

**Address** CnMASK3L: <CnRBaseAddr> + 048<sub>H</sub>

CnMASK3H: <CnRBaseAddr> + 04A<sub>H</sub>

**Initial Value** Undefined.

CnMASK3L

15	14	13	12	11	10	9	8
CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
7	6	5	4	3	2	1	0
CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0

CnMASK3H

15	14	13	12	11	10	9	8
0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
7	6	5	4	3	2	1	0
CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

**(d) CANn module mask 4 register (CnMASK4L, CnMASK4H)**

**Access** These registers can be read/written in 16-bit units.

**Address** CnMASK4L: <CnRBaseAddr> + 04C<sub>H</sub>

CnMASK4H: <CnRBaseAddr> + 04E<sub>H</sub>

**Initial Value** Undefined.

CnMASK4L

15	14	13	12	11	10	9	8
CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8
7	6	5	4	3	2	1	0
CMID7	CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0

CnMASK4H

15	14	13	12	11	10	9	8
0	0	0	CMID28	CMID27	CMID26	CMID25	CMID24
7	6	5	4	3	2	1	0
CMID23	CMID22	CMID21	CMID20	CMID19	CMID18	CMID17	CMID16

CMID28 to CMID0	Mask pattern setting of ID bit
0	The ID bits of the message buffer set by the CMID28 to CMID0 bits are compared with the ID bits of the received message frame.
1	The ID bits of the message buffer set by the CMID28 to CMID0 bits are not compared with the ID bits of the received message frame (they are masked).

**Note** Masking is always defined by an ID length of 29 bits. If a mask is assigned to a message with a standard ID, the CMID17 to CMID0 bits are ignored. Therefore, only the CMID28 to CMID18 bits of the received ID are masked. The same mask can be used for both the standard and extended IDs.

**(6) CnCTRL - CANn module control register**

The CnCTRL register is used to control the operation mode of the CAN module.

**Access** This register can be read/written in 16-bit units.

**Address** <CnRBaseAddr> + 050<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. The register is initialized by any reset.

**(a) CnCTRL read**

15	14	13	12	11	10	9	8
0	0	0	0	0	0	RSTAT	TSTAT
7	6	5	4	3	2	1	0
CCERC	AL	VALID	PSMODE1	PSMODE0	OPMODE2	OPMODE1	OPMODE0

RSTAT	Reception status bit
0	Reception is stopped.
1	Reception is in progress.

- Note**
- The RSTAT bit is set to 1 under the following conditions (timing)
    - The SOF bit of a receive frame is detected
    - On occurrence of arbitration loss during a transmit frame
  - The RSTAT bit is cleared to 0 under the following conditions (timing)
    - When a recessive level is detected at the second bit of the interframe space
    - On transition to the initialization mode at the first bit of the interframe space

TSTAT	Transmission status bit
0	Transmission is stopped.
1	Transmission is in progress.

- Note**
- The TSTAT bit is set to 1 under the following conditions (timing)
    - The SOF bit of a transmit frame is detected
  - The TSTAT bit is cleared to 0 under the following conditions (timing)
    - During transition to bus-off state
    - On occurrence of arbitration loss in transmit frame
    - On detection of recessive level at the second bit of the interframe space
    - On transition to the initialization mode at the first bit of the interframe space

CCERC	Error counter clear bit
0	The CnERC and CnINFO registers are not cleared in the initialization mode.
1	The CnERC and CnINFO registers are cleared in the initialization mode.

- Note**
1. The CCERC bit is used to clear the CnERC and CnINFO registers for re-initialization or forced recovery from the bus-off state. This bit can be set to 1 only in the initialization mode.
  2. When the CnERC and CnINFO registers have been cleared, the CCERC bit is also cleared to 0 automatically.
  3. The CCERC bit can be set to 1 at the same time as a request to change the initialization mode to an operation mode is made.
  4. The CCERC bit is read-only in the CAN sleep mode or CAN stop mode.
  5. The receive data may be corrupted in case of setting the CCERC bit to (1) immediately after entering the INIT mode from self-test mode.

AL	Bit to set operation in case of arbitration loss
0	Re-transmission is not executed in case of an arbitration loss in the single-shot mode.
1	Re-transmission is executed in case of an arbitration loss in the single-shot mode.

- Note** The AL bit is valid only in the single-shot mode.

VALID	Valid receive message frame detection bit
0	A valid message frame has not been received since the VALID bit was last cleared to 0.
1	A valid message frame has been received since the VALID bit was last cleared to 0.

- Note**
1. Detection of a valid receive message frame is not dependent upon storage in the receive message buffer (data frame) or transmit message buffer (remote frame).
  2. Clear the VALID bit (0) before changing the initialization mode to an operation mode.
  3. If only two CAN nodes are connected to the CAN bus with one transmitting a message frame in the normal mode and the other in the receive-only mode, the VALID bit is not set to 1 before the transmitting node enters the error passive state, because in receive-only mode no acknowledge is generated.
  4. To clear the VALID bit, set the Clear VALID bit to 1 first and confirm that the VALID bit is cleared. If it is not cleared, perform clearing processing again.

PSMODE1	PSMODE0	Power save mode
0	0	No power save mode is selected.
0	1	CAN sleep mode
1	0	Setting prohibited
1	1	CAN stop mode

- Caution**
1. Transition to and from the CAN stop mode must be made via CAN sleep mode. A request for direct transition to and from the CAN stop mode is ignored.
  2. The MBON flag of CnGMCTRL must be checked after releasing a power save mode, prior to access the message buffers again.
  3. CAN sleep mode requests are kept pending, until cancelled by software or entered on appropriate bus condition (bus idle). Software can check the actual status by reading PSMODE.

OPMODE2	OPMODE1	OPMODE0	Operation mode
0	0	0	No operation mode is selected (CAN module is in the initialization mode).
0	0	1	Normal operation mode
0	1	0	Normal operation mode with automatic block transmission function (normal operation mode with ABT)
0	1	1	Receive-only mode
1	0	0	Single-shot mode
1	0	1	Self-test mode
Other than above			Setting prohibited

- Caution** Transit to initialization mode or power saving modes may take some time. Be sure to verify the success of mode change by reading the values, before proceeding.

**Note** The OPMODE0 to OPMODE2 bits are read-only in the CAN sleep mode or CAN stop mode.

**(b) CnCTRL write**

15	14	13	12	11	10	9	8
Set CCERC	Set AL	0	Set PSMODE1	Set PSMODE0	Set OPMODE2	Set OPMODE1	Set OPMODE0
7	6	5	4	3	2	1	0
0	Clear AL	Clear VALID	Clear PSMODE1	Clear PSMODE0	Clear OPMODE2	Clear OPMODE1	Clear OPMODE0



Set CCERC	Setting of CCERC bit
1	CCERC bit is set to 1.
Other than above	CCERC bit is not changed.

Set AL	Clear AL	Setting of AL bit
0	1	AL bit is cleared to 0.
1	0	AL bit is set to 1.
Other than above		AL bit is not changed.

Clear VALID	Setting of VALID bit
0	VALID bit is not changed.
1	VALID bit is cleared to 0.

Set PSMODE0	Clear PSMODE0	Setting of PSMODE0 bit
0	1	PSMODE0 bit is cleared to 0.
1	0	PSMODE0 bit is set to 1.
Other than above		PSMODE0 bit is not changed.

Set PSMODE1	Clear PSMODE1	Setting of PSMODE1 bit
0	1	PSMODE1 bit is cleared to 0.
1	0	PSMODE1 bit is set to 1.
Other than above		PSMODE1 bit is not changed.

Set OPMODE0	Clear OPMODE0	Setting of OPMODE0 bit
0	1	OPMODE0 bit is cleared to 0.
1	0	OPMODE0 bit is set to 1.
Other than above		OPMODE0 bit is not changed.

Set OPMODE1	Clear OPMODE1	Setting of OPMODE1 bit
0	1	OPMODE1 bit is cleared to 0.
1	0	OPMODE1 bit is set to 1.
Other than above		OPMODE1 bit is not changed.

Set OPMODE2	Clear OPMODE2	Setting of OPMODE2 bit
0	1	OPMODE2 bit is cleared to 0.
1	0	OPMODE2 bit is set to 1.
Other than above		OPMODE2 bit is not changed.

#### (7) CnLEC - CANn module last error information register

The CnLEC register provides the error information of the CAN protocol.

**Access** This register can be read/written in 8-bit units.

**Address** <CnRBaseAddr> + 052<sub>H</sub>

**Initial Value** 00<sub>H</sub>. The register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	LEC2	LEC1	LEC0

- Note**
1. The contents of the CnLEC register are not cleared when the CAN module changes from an operation mode to the initialization mode.
  2. If an attempt is made to write a value other than 00<sub>H</sub> to the CnLEC register by software, the access is ignored.

LEC2	LEC1	LEC0	Last CAN protocol error information
0	0	0	No error
0	0	1	Stuff error
0	1	0	Form error
0	1	1	ACK error
1	0	0	Bit error. (The CAN module tried to transmit a recessive-level bit as part of a transmit message (except the arbitration field), but the value on the CAN bus is a dominant-level bit.)
1	0	1	Bit error. (The CAN module tried to transmit a dominant-level bit as part of a transmit message, ACK bit, error frame, or overload frame, but the value on the CAN bus is a recessive-level bit.)
1	1	0	CRC error
1	1	1	Undefined

**(8) CnINFO - CANn module information register**

The CnINFO register indicates the status of the CAN module.

**Access** This register is read-only in 8-bit units.

**Address** <CnRBaseAddr> + 053<sub>H</sub>

**Initial Value** 00<sub>H</sub>. The register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	BOFF	TECS1	TECS0	RECS1	RECS0

BOFF	Bus-off state bit
0	Not bus-off state (transmit error counter ≤ 255). (The value of the transmit counter is less than 256.)
1	Bus-off state (transmit error counter > 255). (The value of the transmit counter is 256 or more.)

TECS1	TECS0	Transmission error counter status bit
0	0	The value of the transmission error counter is less than that of the warning level (< 96).
0	1	The value of the transmission error counter is in the range of the warning level (96 to 127).
1	0	Undefined
1	1	The value of the transmission error counter is in the range of the error passive or bus-off status (≥ 128).

RECS1	RECS0	Reception error counter status bit
0	0	The value of the reception error counter is less than that of the warning level (< 96).
0	1	The value of the reception error counter is in the range of the warning level (96 to 127).
1	0	Undefined
1	1	The value of the reception error counter is in the error passive range (≥ 128).

**(9) CnERC - CANn module error counter register**

The CnERC register indicates the count value of the transmission/reception error counter.

**Access** This register is read-only in 16-bit units.

**Address** <CnRBaseAddr> + 054<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. The register is initialized by any reset.

15	14	13	12	11	10	9	8
REPS	REC6	REC5	REC4	REC3	REC2	REC1	REC0
7	6	5	4	3	2	1	0
TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0

REPS	Reception error passive status bit
0	The reception error counter is not in the error passive range (< 128)
1	The reception error counter is in the error passive range (≥ 128)

REC6 to REC0	Reception error counter bit
0 to 127	Number of reception errors. These bits reflect the status of the reception error counter. The number of errors is defined by the CAN protocol.

**Note** REC6 to REC0 of the reception error counter are invalid in the reception error passive state (CnINFO.RECS[1:0] = 11<sub>B</sub>).

TEC7 to TEC0	Transmission error counter bit
0 to 255	Number of transmission errors. These bits reflect the status of the transmission error counter. The number of errors is defined by the CAN protocol.

**Note** The TEC7 to TEC0 bits of the transmission error counter are invalid in the bus-off state (CnINFO.BOFF = 1).

**(10) CnIE - CANn module interrupt enable register**

The CnIE register is used to enable or disable the interrupts of the CAN module.

**Access** This register can be read/written in 16-bit units.

**Address** <CnRBaseAddr> + 056<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. The register is initialized by any reset.

**(a) CnIE read**

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	CIE5	CIE4	CIE3	CIE2	CIE1	CIE0

CIE5 to CIE0	CAN module interrupt enable bit
0	Output of the interrupt corresponding to interrupt status register CINTSx is disabled.
1	Output of the interrupt corresponding to interrupt status register CINTSx is enabled.

**(b) CnIE write**

15	14	13	12	11	10	9	8
0	0	Set CIE5	Set CIE4	Set CIE3	Set CIE2	Set CIE1	Set CIE0
7	6	5	4	3	2	1	0
0	0	Clear CIE5	Clear CIE4	Clear CIE3	Clear CIE2	Clear CIE1	Clear CIE0

Set CIE5	Clear CIE5	Setting of CIE5 bit
0	1	CIE5 bit is cleared to 0.
1	0	CIE5 bit is set to 1.
Other than above		CIE5 bit is not changed.

Set CIE4	Clear CIE4	Setting of CIE4 bit
0	1	CIE4 bit is cleared to 0.
1	0	CIE4 bit is set to 1.
Other than above		CIE4 bit is not changed.

Set CIE3	Clear CIE3	Setting of CIE3 bit
0	1	CIE3 bit is cleared to 0.
1	0	CIE3 bit is set to 1.
Other than above		CIE3 bit is not changed.

Set CIE2	Clear CIE2	Setting of CIE2 bit
0	1	CIE2 bit is cleared to 0.
1	0	CIE2 bit is set to 1.
Other than above		CIE2 bit is not changed.

Set CIE1	Clear CIE1	Setting of CIE1 bit
0	1	CIE1 bit is cleared to 0.
1	0	CIE1 bit is set to 1.
Other than above		CIE1 bit is not changed.

Set CIE0	Clear CIE0	Setting of CIE0 bit
0	1	CIE0 bit is cleared to 0.
1	0	CIE0 bit is set to 1.
Other than above		CIE0 bit is not changed.

**(11) CnINTS - CANn module interrupt status register**

The CnINTS register indicates the interrupt status of the CAN module.

**Access** This register can be read/written in 16-bit units.

**Address** <CnRBaseAddr> + 058<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. The register is initialized by any reset.

**(a) CnINTS read**

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	CINTS5	CINTS4	CINTS3	CINTS2	CINTS1	CINTS0

CINTS5 to CINTS0	CAN interrupt status bit
0	No related interrupt source event is pending.
1	A related interrupt source event is pending.

Interrupt status bit	Related interrupt source event
CINTS5	Wakeup interrupt from CAN sleep mode <sup>a</sup>
CINTS4	Arbitration loss interrupt
CINTS3	CAN protocol error interrupt
CINTS2	CAN error status interrupt
CINTS1	Interrupt on completion of reception of valid message frame to message buffer m
CINTS0	Interrupt on normal completion of transmission of message frame from message buffer m

a) The CINTS5 bit is set only when the CAN module is woken up from the CAN sleep mode by a CAN bus operation. The CINTS5 bit is not set when the CAN sleep mode has been released by software.

**(b) CnINTS write**

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	Clear CINTS5	Clear CINTS4	Clear CINTS3	Clear CINTS2	Clear CINTS1	Clear CINTS0

Clear CINTS5 to CINTS0	Setting of CINTS5 to CINTS0 bits
0	CINTS5 to CINTS0 bits are not changed.
1	CINTS5 to CINTS0 bits are cleared to 0.

**Caution** Please clear the status bit of this register with software when the confirmation of each status is necessary in the interrupt processing, because these bits are not cleared automatically.

**(12) CnBRP - CANn module bit rate prescaler register**

The CnBRP register is used to select the CAN protocol layer basic system clock ( $f_{TQ}$ ). The communication baud rate is set to the CnBTR register.

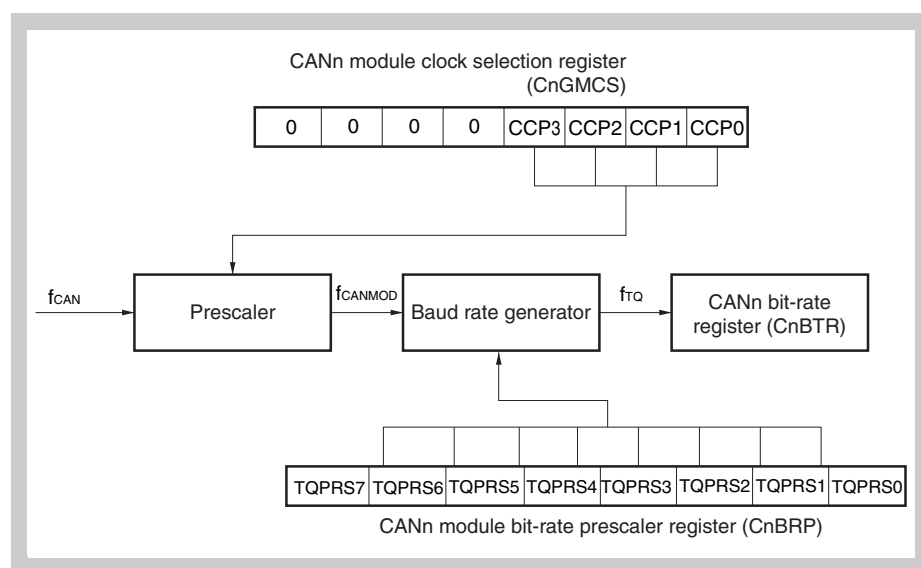
**Access** This register can be read/written in 8-bit units.

**Address** <CnRBaseAddr> + 05A<sub>H</sub>

**Initial Value** FF<sub>H</sub>. The register is initialized by any reset.

7	6	5	4	3	2	1	0
TQPRS7	TQPRS6	TQPRS5	TQPRS4	TQPRS3	TQPRS2	TQPRS1	TQPRS0

TQPRS7 to TQPRS0	CAN protocol layer base system clock ( $f_{TQ}$ )
0	$f_{CANMOD}/1$
1	$f_{CANMOD}/2$
n	$f_{CANMOD}/(n+1)$
:	:
255	$f_{CANMOD}/256$ (default value)



**Figure 14-24** CAN module clock

**Note**  $f_{CAN}$ : clock supplied to CAN  
 $f_{CANMOD}$ : CAN module system clock  
 $f_{TQ}$ : CAN protocol layer basic system clock

**Caution** The CnBRP register can be write-accessed only in the initialization mode.



**(13) CnBTR - CANn module bit rate register**

The CnBTR register is used to control the data bit time of the communication baud rate.

**Access** This register can be read/written in 16-bit units.

**Address** <CnRBaseAddr> + 05C<sub>H</sub>

**Initial Value** 370F<sub>H</sub>. The register is initialized by any reset.

15	14	13	12	11	10	9	8
0	0	SJW1	SJW0	0	TSEG22	TSEG21	TSEG20
7	6	5	4	3	2	1	0
0	0	0	0	TSEG13	TSEG12	TSEG11	TSEG10

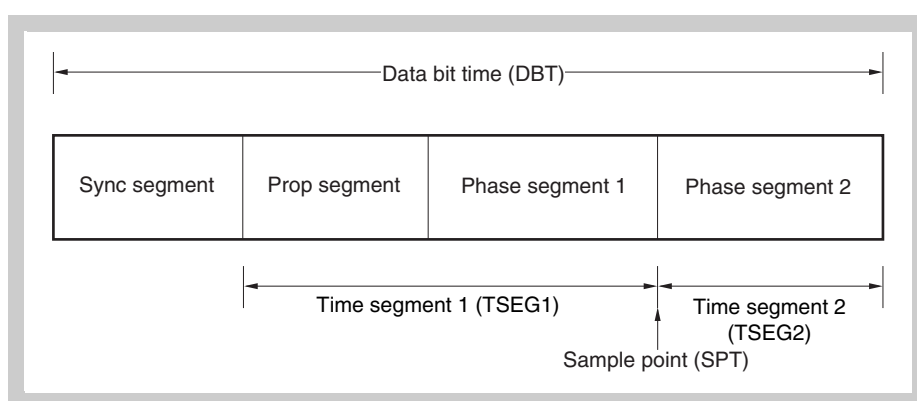


Figure 14-25 Data bit time

SJW1	SJW0	Length of synchronization jump width
0	0	1T <sub>Q</sub>
0	1	2T <sub>Q</sub>
1	0	3T <sub>Q</sub>
1	1	4T <sub>Q</sub> (default value)

TSEG22	TSEG21	TSEG20	Length of time segment 2
0	0	0	1T <sub>Q</sub>
0	0	1	2T <sub>Q</sub>
0	1	0	3T <sub>Q</sub>
0	1	1	4T <sub>Q</sub>
1	0	0	5T <sub>Q</sub>
1	0	1	6T <sub>Q</sub>
1	1	0	7T <sub>Q</sub>
1	1	1	8T <sub>Q</sub> (default value)

TSEG13	TSEG12	TSEG11	TSEG10	Length of time segment 1
0	0	0	0	Setting prohibited
0	0	0	1	$2T_Q^a$
0	0	1	0	$3T_Q^a$
0	0	1	1	$4T_Q$
0	1	0	0	$5T_Q$
0	1	0	1	$6T_Q$
0	1	1	0	$7T_Q$
0	1	1	1	$8T_Q$
1	0	0	0	$9T_Q$
1	0	0	1	$10T_Q$
1	0	1	0	$11T_Q$
1	0	1	1	$12T_Q$
1	1	0	0	$13T_Q$
1	1	0	1	$14T_Q$
1	1	1	0	$15T_Q$
1	1	1	1	$16T_Q$ (default value)

a) This setting must not be made when the CnBRP register =  $00_H$

**Note**  $T_Q = 1/f_{TQ}$  ( $f_{TQ}$ : CAN protocol layer basic system clock)

#### (14) CnLIPT - CANn module last in-pointer register

The CnLIPT register indicates the number of the message buffer in which a data frame or a remote frame was last stored.

**Access** This register is read-only in 8-bit units.

**Address** <CnRBaseAddr> +  $05E_H$

**Initial Value** Undefined.

7	6	5	4	3	2	1	0
LIPT7	LIPT6	LIPT5	LIPT4	LIPT3	LIPT2	LIPT1	LIPT0

LIPT7 to LIPT0	Last in-pointer register (CnLIPT)
0 to 31	When the CnLIPT register is read, the contents of the element indexed by the last in-pointer (LIPT) of the receive history list are read. These contents indicate the number of the message buffer in which a data frame or a remote frame was last stored.

**Note** The read value of the CnLIPT register is undefined if a data frame or a remote frame has never been stored in the message buffer. If the RHPPM bit of the CnRGPT register is set to 1 after the CAN module has changed from the initialization mode to an operation mode, therefore, the read value of the CnLIPT register is undefined.

**(15) CnRGPT - CANn module receive history list register**

The CnRGPT register is used to read the receive history list.

**Access** This register can be read/written in 16-bit units.

**Address** <CnRBaseAddr> + 060<sub>H</sub>

**Initial Value** xx02<sub>H</sub>. The register is initialized by any reset.

**(a) CnRGPT read**

15	14	13	12	11	10	9	8
RGPT7	RGPT6	RGPT5	RGPT4	RGPT3	RGPT2	RGPT1	RGPT0
7	6	5	4	3	2	1	0
0	0	0	0	0	0	RHPM	ROVF

RGPT7 to RGPT0	Receive history list read pointer
0 to 31	When the CnRGPT register is read, the contents of the element indexed by the receive history list get pointer (RGPT) of the receive history list are read. These contents indicate the number of the message buffer in which a data frame or a remote frame has been stored.

RHPM <sup>a</sup>	Receive history list pointer match
0	The receive history list has at least one message buffer number that has not been read.
1	The receive history list has no message buffer numbers that have not been read.

a) The read value of the RGPT0 to RGPT7 bits is invalid when the RHPM bit = 1.

ROVF <sup>a</sup>	Receive history list overflow bit
0	All the message buffer numbers that have not been read are preserved. All the numbers of the message buffers in which a new data frame or remote frame has been received and stored are recorded to the receive history list (the receive history list has a vacant element).
1	At least 23 entries have been stored since the host processor has serviced the RHL last time (i.e. read CnRGPT). The first 22 entries are sequentially stored while the last entry can have been overwritten whenever newly received message is stored because all buffer numbers are stored at position LIPT-1 when ROVF bit is set. Thus the sequence of receptions can not be recovered completely now.

a) If ROVF is set, RHPM is no longer cleared on message storage, but RHPM is still set, if all entries of CnRGPT are read by software.

**(b) CnRGPT write**

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	Clear ROVF

Clear ROVF	Setting of ROVF bit
0	ROVF bit is not changed.
1	ROVF bit is cleared to 0.

**(16) CnLOPT - CANn module last out-pointer register**

The CnLOPT register indicates the number of the message buffer to which a data frame or a remote frame was transmitted last.

**Access** This register is read-only in 8-bit units.

**Address** <CnRBaseAddr> + 062<sub>H</sub>

**Initial Value** Undefined

7	6	5	4	3	2	1	0
LOPT7	LOPT6	LOPT5	LOPT4	LOPT3	LOPT2	LOPT1	LOPT0

LOPT7 to LOPT0	Last out-pointer of transmit history list (LOPT)
0 to 31	When the CnLOPT register is read, the contents of the element indexed by the last out-pointer (LOPT) of the receive history list are read. These contents indicate the number of the message buffer to which a data frame or a remote frame was transmitted last.

**Note** The value read from the CnLOPT register is undefined if a data frame or remote frame has never been transmitted from a message buffer. If the CnTGPT.THPM bit is set to 1 after the CAN module has changed from the initialization mode to an operation mode, therefore, the read value of the CnLOPT register is undefined.

**(17) CnTGPT - CANn module transmit history list register**

The CnTGPT register is used to read the transmit history list.

**Access** This register can be read/written in 16-bit units.

**Address** <CnRBaseAddr> + 064<sub>H</sub>

**Initial Value** xx02<sub>H</sub>. The register is initialized by any reset.

**(a) CnTGPT read**

15	14	13	12	11	10	9	8
TGPT7	TGPT6	TGPT5	TGPT4	TGPT3	TGPT2	TGPT1	TGPT0
7	6	5	4	3	2	1	0
0	0	0	0	0	0	THPM	TOVF

TGPT7 to TGPT0	Transmit history list read pointer
0 to 31	When the CnTGPT register is read, the contents of the element indexed by the read pointer (TGPT) of the transmit history list are read. These contents indicate the number of the message buffer to which a data frame or a remote frame was transmitted last.

THPM <sup>a</sup>	Transmit history pointer match
0	The transmit history list has at least one message buffer number that has not been read.
1	The transmit history list has no message buffer numbers that have not been read.

a) The read value of the TGPT0 to TGPT7 bits is invalid when the THPM bit = 1.

TOVF <sup>a</sup>	Transmit history list overflow bit
0	All the message buffer numbers that have not been read are preserved. All the numbers of the message buffers to which a new data frame or remote frame has been transmitted are recorded to the transmit history list (the transmit history list has a vacant element).
1	At least 7 entries have been stored since the host processor has serviced the THL last time (i.e. read CnTGPT). The first 6 entries are sequentially stored while the last entry can have been overwritten whenever a message is newly transmitted because all buffer numbers are stored at position LOPT-1 when TOVF bit is set. Thus the sequence of transmissions can not be recovered completely now.

a) If TOVF is set, THPM is no longer cleared on message transmission, but THPM is still set, if all entries of CnTGPT are read by software.

**Note** Transmission from message buffers 0 to 7 is not recorded to the transmit history list in the normal operation mode with ABT.

**(b) CnTGPT write**

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	Clear TOVF

Clear TOVF	Setting of TOVF bit
0	TOVF bit is not changed.
1	TOVF bit is cleared to 0.

**(18) CnTS - CANn module time stamp register**

The CnTS register is used to control the time stamp function.

**Access** This register can be read/written in 16-bit units.

**Address** <CnRBaseAddr> + 066<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. The register is initialized by any reset.

**(a) CnTS read**

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	0	0	0	TSLOCK	TSSEL	TSEN

**Note** The lock function of the time stamp function must not be used when the CAN module is in the normal operation mode with ABT.

TSLOCK	Time stamp lock function enable bit
0	Time stamp lock function stopped. The TSOUT signal is toggled each time the selected time stamp capture event occurs.
1	Time stamp lock function enabled. The TSOUT signal is toggled each time the selected time stamp capture event occurs. However, the TSOUT output signal is locked when a data frame has been correctly received to message buffer 0 <sup>a</sup> .

a) The TSEN bit is automatically cleared to 0.

TSSEL	Time stamp capture event selection bit
0	The time capture event is SOF.
1	The time stamp capture event is the last bit of EOF.

TSEN	TSOUT operation setting bit
0	TSOUT toggle operation is disabled.
1	TSOUT toggle operation is enabled.

**(b) CnTS write**

15	14	13	12	11	10	9	8
0	0	0	0	0	Set TSLOCK	Set TSSEL	Set TSEN
7	6	5	4	3	2	1	0
0	0	0	0	0	Clear TSLOCK	Clear TSSEL	Clear TSEN

Set TSLOCK	Clear TSLOCK	Setting of TSLOCK bit
0	1	TSLOCK bit is cleared to 0.
1	0	TSLOCK bit is set to 1.
Other than above		TSLOCK bit is not changed.

Set TSSEL	Clear TSSEL	Setting of TSSEL bit
0	1	TSSEL bit is cleared to 0.
1	0	TSSEL bit is set to 1.
Other than above		TSSEL bit is not changed.

Set TSEN	Clear TSEN	Setting of TSEN bit
0	1	TSEN bit is cleared to 0.
1	0	TSEN bit is set to 1.
Other than above		TSEN bit is not changed.



**(19) CnMDATAxm, CnMDATAzm - CANn message data byte register (x = 0 to 7, z = 01, 23, 45, 67)**

The CnMDATAxm, CnMDATAzm registers are used to store the data of a transmit/receive message.

**Access** The CnMDATAzm registers can be read/written in 16-bit units.  
The CnMDATAxm registers can be read/written in 8-bit units.

**Address** Refer to “CAN registers overview” on page 395.

**Initial Value** Undefined.

**CnMDATA01m**

15	14	13	12	11	10	9	8
MDATA0115	MDATA0114	MDATA0113	MDATA0112	MDATA0111	MDATA0110	MDATA0109	MDATA0108
7	6	5	4	3	2	1	0
MDATA0107	MDATA0106	MDATA0105	MDATA0104	MDATA0103	MDATA0102	MDATA0101	MDATA0100

**CnMDATA0m**

7	6	5	4	3	2	1	0
MDATA007	MDATA006	MDATA005	MDATA004	MDATA003	MDATA002	MDATA001	MDATA000

**CnMDATA1m**

7	6	5	4	3	2	1	0
MDATA107	MDATA106	MDATA105	MDATA104	MDATA103	MDATA102	MDATA101	MDATA100

**CnMDATA23m**

15	14	13	12	11	10	9	8
MDATA2315	MDATA2314	MDATA2313	MDATA2312	MDATA2311	MDATA2310	MDATA2309	MDATA2308
7	6	5	4	3	2	1	0
MDATA2307	MDATA2306	MDATA2305	MDATA2304	MDATA2303	MDATA2302	MDATA2301	MDATA2300

**CnMDATA2m**

7	6	5	4	3	2	1	0
MDATA207	MDATA206	MDATA205	MDATA204	MDATA203	MDATA202	MDATA201	MDATA200

**CnMDATA3m**

7	6	5	4	3	2	1	0
MDATA307	MDATA306	MDATA305	MDATA304	MDATA303	MDATA302	MDATA301	MDATA300

## CnMDATA45m

15	14	13	12	11	10	9	8
MDATA4515	MDATA4514	MDATA4513	MDATA4512	MDATA4511	MDATA4510	MDATA4509	MDATA4508
7	6	5	4	3	2	1	0
MDATA4507	MDATA4506	MDATA4505	MDATA4504	MDATA4503	MDATA4502	MDATA4501	MDATA4500

## CnMDATA4m

7	6	5	4	3	2	1	0
MDATA407	MDATA406	MDATA405	MDATA404	MDATA403	MDATA402	MDATA401	MDATA400

## CnMDATA5m

7	6	5	4	3	2	1	0
MDATA507	MDATA506	MDATA505	MDATA504	MDATA503	MDATA502	MDATA501	MDATA500

## CnMDATA67m

15	14	13	12	11	10	9	8
MDATA6715	MDATA6714	MDATA6713	MDATA6712	MDATA6711	MDATA6710	MDATA6709	MDATA6708
7	6	5	4	3	2	1	0
MDATA6707	MDATA6706	MDATA6705	MDATA6704	MDATA6703	MDATA6702	MDATA6701	MDATA6700

## CnMDATA6m

7	6	5	4	3	2	1	0
MDATA607	MDATA606	MDATA605	MDATA604	MDATA603	MDATA602	MDATA601	MDATA600

## CnMDATA7m

7	6	5	4	3	2	1	0
MDATA707	MDATA706	MDATA705	MDATA704	MDATA703	MDATA702	MDATA701	MDATA700

**(20) CnMDLCm - CANn message data length register m**

The CnMDLCm register is used to set the number of bytes of the data field of a message buffer.

**Access** This register can be read/written in 8-bit units.

**Address** Refer to “CAN registers overview” on page 395.

**Initial Value** 0000xxxx<sub>B</sub>. The register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	MDLC3	MDLC2	MDLC1	MDLC0

MDLC3	MDLC2	MDLC1	MDLC0	Data length of transmit/receive message
0	0	0	0	0 bytes
0	0	0	1	1 byte
0	0	1	0	2 bytes
0	0	1	1	3 bytes
0	1	0	0	4 bytes
0	1	0	1	5 bytes
0	1	1	0	6 bytes
0	1	1	1	7 bytes
1	0	0	0	8 bytes
1	0	0	1	Setting prohibited (If these bits are set during transmission, 8-byte data is transmitted regardless of the set DLC value when a data frame is transmitted. However, the DLC actually transmitted to the CAN bus is the DLC value set to this register.) <sup>Note</sup>
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

**Note** The data and DLC value actually transmitted to CAN bus are as follows.

Type of transmit frame	Length of transmit data	DLC transmitted
Data frame	Number of bytes specified by DLC (However, 8 bytes if $DLC \geq 8$ )	MDLC3 to MDLC0 bits
Remote frame	0 bytes	

- Caution**
1. Be sure to set bits 7 to 4 to 0000<sub>B</sub>.
  2. Receive data is stored in as many CnMDATAxm register as the number of bytes (however, the upper limit is 8) corresponding to DLC of the received frame. The CnMDATAxm register in which no data is stored is undefined.

**(21) CnMCONFm - CANn message configuration register m**

The CnMCONFm register is used to specify the type of the message buffer and to set a mask.

**Access** This register can be read/written in 8-bit units.

**Address** Refer to “CAN registers overview” on page 395.

**Initial Value** Undefined.

7	6	5	4	3	2	1	0
OVS	RTR	MT2	MT1	MT0	0	0	MA0

OVS	Overwrite control bit
0	The message buffer that has already received a data frame <sup>a</sup> is not overwritten by a newly received data frame. The newly received data frame is discarded.
1	The message buffer that has already received a data frame <sup>a</sup> is overwritten by a newly received data frame.

a) The “message buffer that has already received a data frame” is a receive message buffer whose the CnMCTRLm.DN bit has been set to 1.

**Note** A remote frame is received and stored, regardless of the setting of OVS and DN. A remote frame that satisfies the other conditions (ID matches, RTR = 0, TRQ = 0) is always received and stored in the corresponding message buffer (interrupt generated, DN flag set, MDLC[3:0] updated, and recorded to the receive history list).

RTR	Remote frame request bit <sup>a</sup>
0	Transmit a data frame.
1	Transmit a remote frame.

a) The RTR bit specifies the type of message frame that is transmitted from a message buffer defined as a transmit message buffer. Even if a valid remote frame has been received, the RTR bit of the transmit message buffer that has received the frame remains cleared to 0. Even if a remote frame whose ID matches has been received from the CAN bus with the RTR bit of the transmit message buffer set to 1 to transmit a remote frame, that remote frame is not received or stored (interrupt generated, DN flag set, the MDLC0 to MDLC3 bits updated, and recorded to the receive history list).

MT2	MT1	MT0	Message buffer type setting bit
0	0	0	Transmit message buffer
0	0	1	Receive message buffer (no mask setting)
0	1	0	Receive message buffer (mask 1 set)
0	1	1	Receive message buffer (mask 2 set)
1	0	0	Receive message buffer (mask 3 set)
1	0	1	Receive message buffer (mask 4 set)
Other than above			Setting prohibited

MA0	Message buffer assignment bit
0	Message buffer not used.
1	Message buffer used.

---

**Caution** Be sure to write 0 to bits 2 and 1.

---

**(22) CnMIDLm, CnMIDHm - CANn message ID register m**

The CnMIDLm and CnMIDHm registers are used to set an identifier (ID).

**Access** These registers can be read/written in 16-bit units.

**Address** Refer to “CAN registers overview” on page 395.

**Initial Value** Undefined.

CnMIDLm

15	14	13	12	11	10	9	8
ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
7	6	5	4	3	2	1	0
ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0

CnMIDHm

15	14	13	12	11	10	9	8
IDE	0	0	ID28	ID27	ID26	ID25	ID24
7	6	5	4	3	2	1	0
ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16

IDE	Format mode specification bit
0	Standard format mode (ID28 to ID18: 11 bits) <sup>a</sup>
1	Extended format mode (ID28 to ID0: 29 bits)

<sup>a)</sup> The ID17 to ID0 bits are not used.

ID28 to ID0	Message ID
ID28 to ID18	Standard ID value of 11 bits (when IDE = 0)
ID28 to ID0	Extended ID value of 29 bits (when IDE = 1)

- Caution**
1. Be sure to write 0 to bits 14 and 13 of the CnMIDHm register.
  2. Be sure to align the ID value according to the given bit positions into this registers. Note that for standard ID, the ID value must be shifted to fit into ID28 to ID11 bit positions.

**(23) CnMCTRLm - CANn message control register m**

The CnMCTRLm register is used to control the operation of the message buffer.

**Access** This register can be read/written in 16-bit units.

**Address** Refer to “CAN registers overview” on page 395.

**Initial Value** 00x0 0000 0000 0000<sub>B</sub>. The register is initialized by any reset.

**(a) CnMCTRLm read**

15	14	13	12	11	10	9	8
0	0	MUC	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	0	MOW	IE	DN	TRQ	RDY

MUC <sup>a</sup>	Bit indicating that message buffer data is being updated
0	The CAN module is not updating the message buffer (reception and storage).
1	The CAN module is updating the message buffer (reception and storage).

a) The MUC bit is undefined until the first reception and storage is performed.

MOW <sup>a</sup>	Message buffer overwrite status bit
0	The message buffer is not overwritten by a newly received data frame.
1	The message buffer is overwritten by a newly received data frame.

a) The MOW bit is not set to 1 even if a remote frame is received and stored in the transmit message buffer with the DN bit = 1.

IE	Message buffer interrupt request enable bit
0	Receive message buffer: Valid message reception completion interrupt disabled. Transmit message buffer: Normal message transmission completion interrupt disabled.
1	Receive message buffer: Valid message reception completion interrupt enabled. Transmit message buffer: Normal message transmission completion interrupt enabled.

DN	Message buffer data update bit
0	A data frame or remote frame is not stored in the message buffer.
1	A data frame or remote frame is stored in the message buffer.

TRQ	Message buffer transmission request bit
0	No message frame transmitting request that is pending or being transmitted is in the message buffer.
1	The message buffer is holding transmission of a message frame pending or is transmitting a message frame.

RDY	Message buffer ready bit
0	The message buffer can be written by software. The CAN module cannot write to the message buffer.
1	Writing the message buffer by software is ignored (except a write access to the RDY, TRQ, DN, and MOW bits). The CAN module can write to the message buffer.

**(b) CnMCTRLm write**

15	14	13	12	11	10	9	8
0	0	0	0	Set IE	0	Set TRQ	Set RDY
7	6	5	4	3	2	1	0
0	0	0	Clear MOW	Clear IE	Clear DN	Clear TRQ	Clear RDY

Clear MOW	Setting of MOW bit
0	MOW bit is not changed.
1	MOW bit is cleared to 0.

Set IE	Clear IE	Setting of IE bit
0	1	IE bit is cleared to 0.
1	0	IE bit is set to 1.
Other than above		IE bit is not changed.

Clear DN	Setting of DN bit
1	DN bit is cleared to 0.
0	DN bit is not changed.

Set TRQ	Clear TRQ	Setting of TRQ bit
0	1	TRQ bit is cleared to 0.
1	0	TRQ bit is set to 1.
Other than above		TRQ bit is not changed.



Set RDY	Clear RDY	Setting of RDY bit
0	1	RDY bit is cleared to 0.
1	0	RDY bit is set to 1.
Other than above		RDY bit is not changed.

- 
- Caution**
1. Set IE bit and RDY bit always separately.
  2. Do not set the DN bit to 1 by software. Be sure to write 0 to bit 10.
  3. Do not set the TRQ bit and the RDY bit (1) at the same time. Set the RDY bit (1) before setting the TRQ bit.
  4. Do not clear the RDY bit (0) during message transmission. Follow the transmission abort process about clearing the RDY bit (0) for redefinition of the message buffer.
  5. Clear again when RDY bit is not cleared even if this bit is cleared.
  6. Be sure that RDY is cleared before writing to the other message buffer registers, by checking the status of the RDY bit.
-

## 14.8 CAN Controller Initialization

### 14.8.1 Initialization of CAN module

Before CAN module operation is enabled, the CAN module system clock needs to be determined by setting the CCP[3:0] bits of the CnGMCS register by software. Do not change the setting of the CAN module system clock after CAN module operation is enabled.

The CAN module is enabled by setting the GOM bit of the CnGMCTRL register.

For the procedure of initializing the CAN module, refer to “*Operation of CAN Controller*” on page 477.

### 14.8.2 Initialization of message buffer

After the CAN module is enabled, the message buffers contain undefined values. A minimum initialization for all the message buffers, even for those not used in the application, is necessary before switching the CAN module from the initialization mode to one of the operation modes.

- Clear the RDY, TRQ, and DN bits of all CnMCTRLm registers to 0.
- Clear the MA0 bit of all CnMCONFm registers to 0.

### 14.8.3 Redefinition of message buffer

Redefining a message buffer means changing the ID and control information of the message buffer while a message is being received or transmitted, without affecting other transmission/reception operations.

#### (1) To redefine message buffer in initialization mode

Place the CAN module in the initialization mode once and then change the ID and control information of the message buffer in the initialization mode. After changing the ID and control information, set the CAN module to an operation mode.

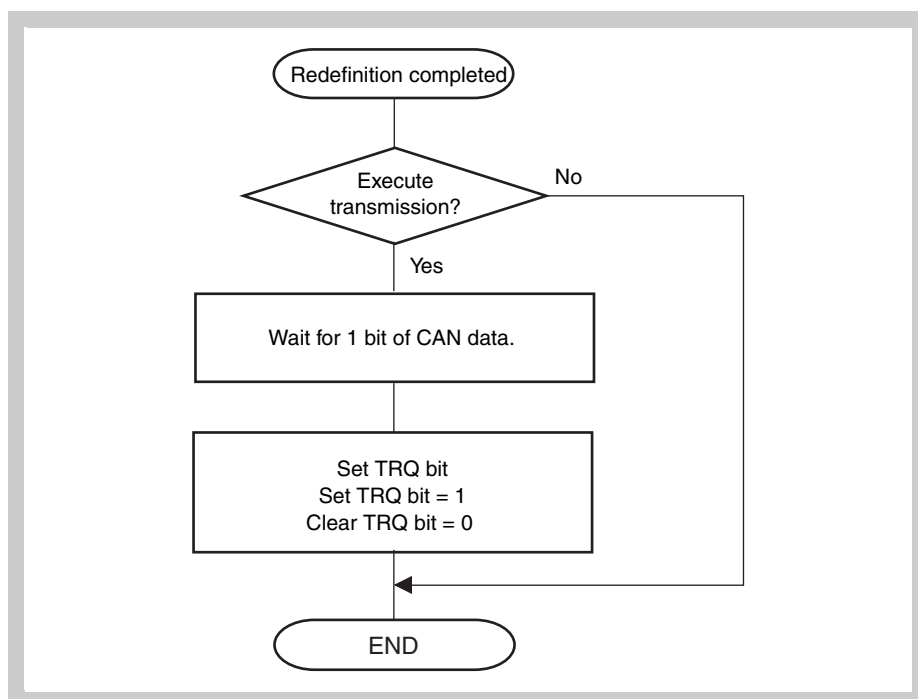
#### (2) To redefine message buffer during reception

Perform redefinition as shown in *Figure 14-38*.

#### (3) To redefine message buffer during transmission

To rewrite the contents of a transmit message buffer to which a transmission request has been set, perform transmission abort processing (see “*Transmission abort process except for in normal operation mode with automatic block transmission (ABT)*” on page 456 and “*Transmission abort process except for ABT transmission in normal operation mode with automatic block transmission (ABT)*” on page 456). Confirm that transmission has been aborted or completed, and then redefine the message buffer. After redefining

the transmit message buffer, set a transmission request using the procedure described below. When setting a transmission request to a message buffer that has been redefined without aborting the transmission in progress, however, the 1-bit wait time is not necessary.



**Figure 14-26** Setting transmission request (TRQ) to transmit message buffer after redefinition

- Caution**
1. When a message is received, reception filtering is performed in accordance with the ID and mask set to each receive message buffer. If the procedure in *Figure 14-38 on page 480* is not observed, the contents of the message buffer after it has been redefined may contradict the result of reception (result of reception filtering). If this happens, check that the ID and IDE received first and stored in the message buffer following redefinition are those stored after the message buffer has been redefined. If no ID and IDE are stored after redefinition, redefine the message buffer again.
  2. When a message is transmitted, the transmission priority is checked in accordance with the ID, IDE, and RTR bits set to each transmit message buffer to which a transmission request was set. The transmit message buffer having the highest priority is selected for transmission. If the procedure in *Figure 14-26 on page 439* is not observed, a message with an ID not having the highest priority may be transmitted after redefinition.

### 14.8.4 Transition from initialization mode to operation mode

The CAN module can be switched to the following operation modes.

- Normal operation mode
- Normal operation mode with ABT
- Receive-only mode
- Single-shot mode
- Self-test mode

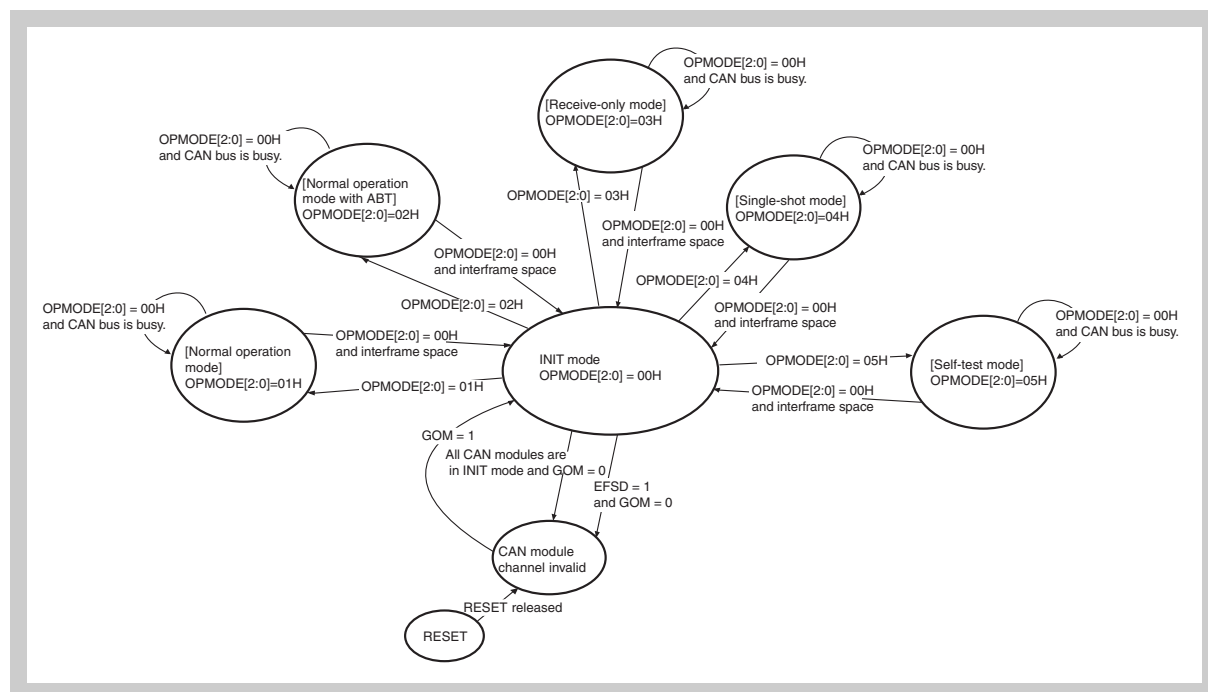


Figure 14-27 Transition to operation modes

The transition from the initialization mode to an operation mode is controlled by the bit string OPMODE[2:0] in the CnCTRL register.

Changing from one operation mode into another requires shifting to the initialization mode in between. Do not change one operation mode to another directly; otherwise the operation will not be guaranteed.

Requests for transition from an operation mode to the initialization mode are held pending when the CAN bus is not in the interframe space (i.e., frame reception or transmission is in progress), and the CAN module enters the initialization mode at the first bit in the interframe space (the values of the OPMODE[2:0] bits are changed to 000<sub>B</sub>). After issuing a request to change the mode to the initialization mode, read the OPMODE[2:0] bits until their value becomes 000<sub>B</sub> to confirm that the module has entered the initialization mode (see Figure 14-36 on page 478).

---

### 14.8.5 Resetting error counter CnERC of CAN module

If it is necessary to reset the CAN module error counter CnERC and CAN module information register CnINFO when re-initialization or forced recovery from the bus-off status is made, set the CCERC bit of the CnCTRL register to 1 in the initialization mode. When this bit is set to 1, the CnERC and CnINFO registers are cleared to their default values.

## 14.9 Message Reception

### 14.9.1 Message reception

In all the operation modes, the complete message buffer area is analyzed to find a suitable buffer to store a newly received message. All message buffers satisfying the following conditions are included in that evaluation (RX-search process).

- Used as a message buffer  
(MA0 bit of CnMCONFm register set to 1.)
- Set as a receive message buffer  
(MT[2:0] bits of CnMCONFm register are set to 001<sub>B</sub>, 010<sub>B</sub>, 011<sub>B</sub>, 100<sub>B</sub>, or 101<sub>B</sub>.)
- Ready for reception  
(RDY bit of CnMCTRLm register is set to 1.)

When two or more message buffers of the CAN module receive a message, the message is stored according to the priority explained below. The message is always stored in the message buffer with the highest priority, not in a message buffer with a low priority. For example, when an unmasked receive message buffer and a receive message buffer linked to mask 1 have the same ID, the received message is not stored in the message buffer linked to mask 1, even if that message buffer has not received a message and a message has already been received in the unmasked receive message buffer. In other words, when a condition has been set in two or more message buffers with different priorities, the message buffer with the highest priority always stores the message; the message is not stored in message buffers with a lower priority. This also applies when the message buffer with the highest priority is unable to store a message (i.e., when DN = 1 indicating that a message has already been received, but rewriting is disabled because OWS = 0). In this case, the message is not actually stored in the candidate message buffer with the highest priority, but neither is it stored in a message buffer with a lower priority.

**Table 14-24 MBRB priorities**

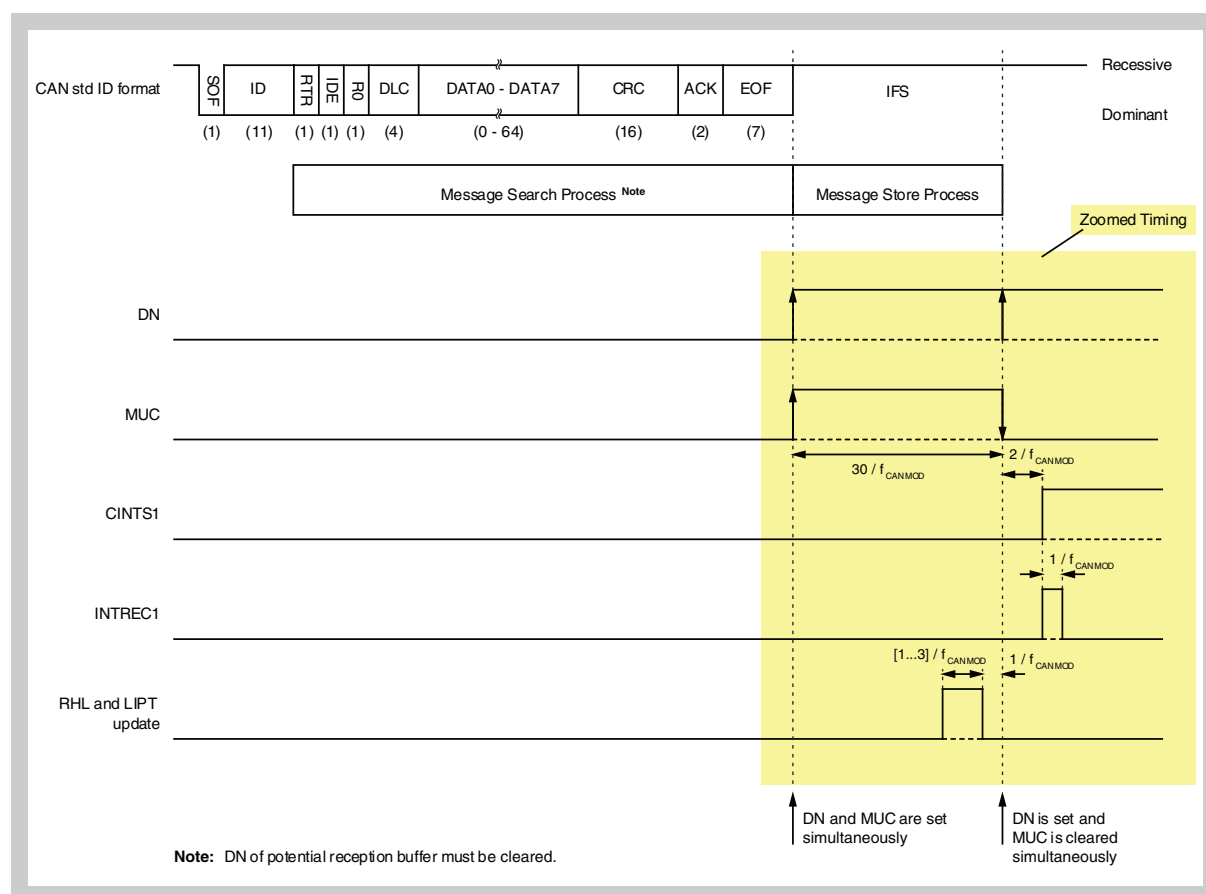
Priority	Storing condition if same ID is set	
1 (high)	Unmasked message buffer	DN bit = 0
		DN bit = 1 and OWS bit = 1
2	Message buffer linked to mask 1	DN bit = 0
		DN bit = 1 and OWS bit = 1
3	Message buffer linked to mask 2	DN bit = 0
		DN bit = 1 and OWS bit = 1
4	Message buffer linked to mask 3	DN bit = 0
		DN bit = 1 and OWS bit = 1
5 (low)	Message buffer linked to mask 4	DN bit = 0
		DN bit = 1 and OWS bit = 1

## 14.9.2 Receive data read

To keep data consistency when reading CAN message buffers, perform the data reading according to *Figure 14-49 on page 493* to *Figure 14-52 on page 497*.

During message reception, the CAN module sets DN of the CnMCTRLm register two times: at the beginning of the storage process of data to the message buffer, and again at the end of this storage process. During this storage process, the MUC bit of the CnMCTRLm register of the message buffer is set. (Refer to *Figure 14-28 on page 443*.)

The receive history list is also updated just before the storage process. In addition, during storage process (MUC = 1), the RDY bit of the CnMCTRL register of the message buffer is locked to avoid the coincidental data WR by CPU. Note the storage process may be disturbed (delayed) when the CPU accesses the message buffer.



**Figure 14-28** DN and MUC bit setting period (for standard ID format)

**Note** If a message shall be stored in a message buffer, the DN bit of this buffer must be cleared before the Message Search Process is started, i.e., right after the ID of the frame is on the bus. In worst case, this happens 15 CAN bits after EOF of the previous frame. Consider to use more than one Message Buffer for reception of a frame, if CAN frames are appearing back-to-back on the bus and none shall be lost.

### 14.9.3 Receive history list function

The receive history list (RHL) function records in the receive history list the number of the receive message buffer in which each data frame or remote frame was received and stored. The RHL consists of storage elements equivalent to up to 23 messages, the last in-message pointer (LIPT) with the corresponding CnLIPT register and the receive history list get pointer (RGPT) with the corresponding CnRGPT register.

The RHL is undefined immediately after the transition of the CAN module from the initialization mode to one of the operation modes.

The CnLIPT register holds the contents of the RHL element indicated by the value of the LIPT pointer minus 1. By reading the CnLIPT register, therefore, the number of the message buffer that received and stored a data frame or remote frame first can be checked. The LIPT pointer is utilized as a write pointer that indicates to what part of the RHL a message buffer number is recorded. Any time a data frame or remote frame is received and stored, the corresponding message buffer number is recorded to the RHL element indicated by the LIPT pointer. Each time recording to the RHL has been completed, the LIPT pointer is automatically incremented. In this way, the number of the message buffer that has received and stored a frame will be recorded chronologically.

The RGPT pointer is utilized as a read pointer that reads a recorded message buffer number from the RHL. This pointer indicates the first RHL element that the CPU has not read yet. By reading the CnRGPT register by software, the number of a message buffer that has received and stored a data frame or remote frame can be read. Each time a message buffer number is read from the CnRGPT register, the RGPT pointer is automatically incremented.

If the value of the RGPT pointer matches the value of the LIPT pointer, the RHPM bit (receive history list pointer match) of the CnRGPT register is set to 1. This indicates that no message buffer number that has not been read remains in the RHL. If a new message buffer number is recorded, the LIPT pointer is incremented and because its value no longer matches the value of the RGPT pointer, the RHPM bit is cleared. In other words, the numbers of the unread message buffers exist in the RHL.

If the LIPT pointer is incremented and matches the value of the RGPT pointer minus 1, the ROVF bit (receive history list overflow) of the CnRGPT register is set to 1. This indicates that the RHL is full of numbers of message buffers that have not been read. When further message reception and storing occur, the last recorded message buffer number is overwritten by the number of the message buffer that received and stored the newly received message. In this case, after the ROVF bit has been set (1), the recorded message buffer numbers in the RHL do not completely reflect the chronological order. However messages itself are not lost and can be located by CPU search in message buffer memory with the help of the DN-bit.

---

**Caution** If the history list is in the overflow condition (ROVF is set), reading the history list contents is still possible, until the history list is empty (indicated by RHPM flag set). Nevertheless, the history list remains in the overflow condition, until ROVF is cleared by software. If ROVF is not cleared, the RHPM flag will also not be updated (cleared) upon a message storage of newly received frame. This may lead to the situation, that RHPM indicates an empty history list, although a reception has taken place, while the history list is in the overflow state (ROVF and RHPM are set).

---



As long as the RHL contains 23 or less entries the sequence of occurrence is maintained. If more receptions occur without reading the RHL by the host processor, complete sequence of receptions can not be recovered.

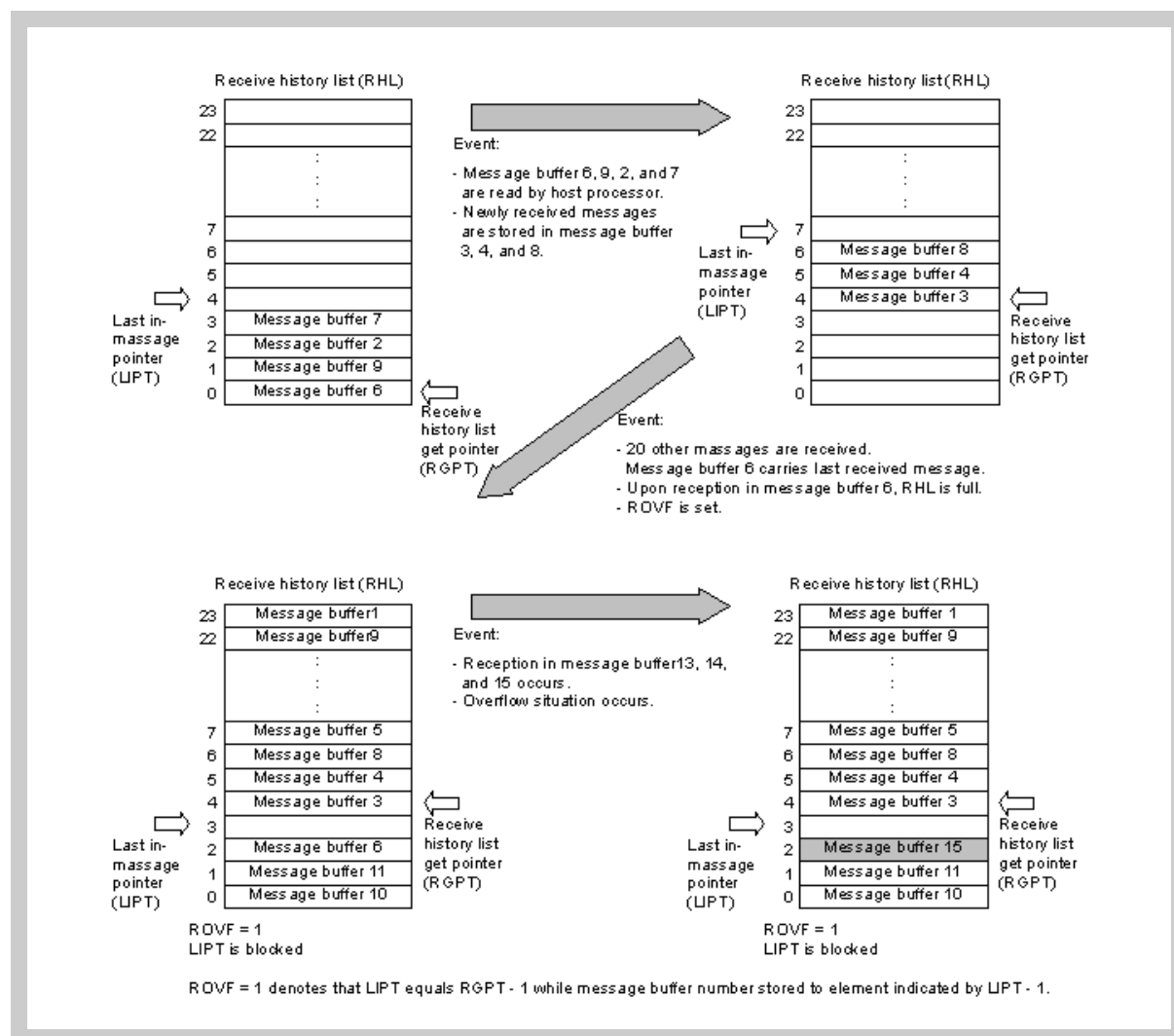


Figure 14-29 Receive history list

### 14.9.4 Mask function

For any message buffer, which is used for reception, the assignment to one of four global reception masks (or no mask) can be selected.

By using the mask function, the message ID comparison can be reduced by masked bits, herewith allowing the reception of several different IDs into one buffer.

While the mask function is in effect, an identifier bit that is defined to be 1 by a mask in the received message is not compared with the corresponding identifier bit in the message buffer.

However, this comparison is performed for any bit whose value is defined as 0 by the mask.

For example, let us assume that all messages that have a standard-format ID, in which bits ID27 to ID25 are 0 and bits ID24 and ID22 are 1, are to be stored in message buffer 14. The procedure for this example is shown below.

#### 1. Identifier to be stored in message buffer

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
x	0	0	0	1	x	1	x	x	x	x

#### 2. Identifier to be configured in message buffer 14 (example) (Using CnMIDL14 and CnMIDH14 registers)

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
x	0	0	0	1	x	1	x	x	x	x
ID17	ID16	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
x	x	x	x	x	x	x	x	x	x	x
ID6	ID5	ID4	ID3	ID2	ID1	ID0				
x	x	x	x	x	x	x				

- Note**
1. ID with the ID27 to ID25 bits cleared to 0 and the ID24 and ID22 bits set to 1 is registered (initialized) to message buffer 14.
  2. Message buffer 14 is set as a standard format identifier that is linked to mask 1 (MT[2:0] of CnMCONF14 register are set to 010<sub>B</sub>).

#### Mask setting for CAN module 1 (mask 1) (example)

(Using CAN1 address mask 1 registers L and H (C1MASKL1 and C1MASKH1))

CMID28	CMID27	CMID26	CMID25	CMID24	CMID23	CMID22	CMID21	CMID20	CMID19	CMID18
1	0	0	0	0	1	0	1	1	1	1
CMID17	CMID16	CMID15	CMID14	CMID13	CMID12	CMID11	CMID10	CMID9	CMID8	CMID7
1	1	1	1	1	1	1	1	1	1	1
CMID6	CMID5	CMID4	CMID3	CMID2	CMID1	CMID0				
1	1	1	1	1	1	1				

1: Not compared (masked)

0: Compared

The CMID27 to CMID24 and CMID22 bits are cleared to 0, and the CMID28, CMID23, and CMID21 to CMID0 bits are set to 1.

### 14.9.5 Multi buffer receive block function

The multi buffer receive block (MBRB) function is used to store a block of data in two or more message buffers sequentially with no CPU interaction, by setting the same ID to two or more message buffers with the same message buffer type. These message buffers can be allocated anywhere in the message buffer memory, they do not even have to follow each other adjacently.

Suppose, for example, the same message buffer type is set to 10 message buffers, message buffers 10 to 19, and the same ID is set to each message buffer. If the first message whose ID matches an ID of the message buffers is received, it is stored in message buffer 10. At this point, the DN bit of message buffer 10 is set, prohibiting overwriting the message buffer when subsequent messages are received.

When the next message with a matching ID is received, it is received and stored in message buffer 11. Each time a message with a matching ID is received, it is sequentially (in the ascending order) stored in message buffers 12, 13, and so on. Even when a data block consisting of multiple messages is received, the messages can be stored and received without overwriting the previously received matching-ID data.

Whether a data block has been received and stored can be checked by setting the IE bit of the CnMCTRLm register of each message buffer. For example, if a data block consists of k messages, k message buffers are initialized for reception of the data block. The IE bit in message buffers 0 to (k-2) is cleared to 0 (interrupts disabled), and the IE bit in message buffer k-1 is set to 1 (interrupts enabled). In this case, a reception completion interrupt occurs when a message has been received and stored in message buffer k-1, indicating that MBRB has become full. Alternatively, by clearing the IE bit of message buffers 0 to (k-3) and setting the IE bit of message buffer k-2, a warning that MBRB is about to overflow can be issued.

The basic conditions of storing receive data in each message buffer for the MBRB are the same as the conditions of storing data in a single message buffer.

- 
- Caution**
1. MBRB can be configured for each of the same message buffer types. Therefore, even if a message buffer of another MBRB whose ID matches but whose message buffer type is different has a vacancy, the received message is not stored in that message buffer, but instead discarded.
  2. MBRB does not have a ring buffer structure. Therefore, after a message is stored in the message buffer having the highest number in the MBRB configuration, a newly received message will not be stored in the message buffer having the lowest message buffer number.
  3. MBRB operates based on the reception and storage conditions; there are no settings dedicated to MBRB, such as function enable bits. By setting the same message buffer type and ID to two or more message buffers, MBRB is automatically configured.
  4. With MBRB, “matching ID” means “matching ID after mask”. Even if the ID set to each message buffer is not the same, if the ID that is masked by the mask register matches, it is considered a matching ID and the buffer that has this ID is treated as the storage destination of a message.
  5. The priority between MBRBs is mentioned in the table *Table 14-24*.
-

### 14.9.6 Remote frame reception

In all the operation modes, when a remote frame is received, the message buffer that is to store the remote frame is searched from all the message buffers satisfying the following conditions.

- Used as a message buffer  
(MA0 bit of CnMCONFm register set to 1.)
- Set as a transmit message buffer  
(MT[2:0] bits in CnMCONFm register set to 000<sub>B</sub>)
- Ready for reception  
(RDY bit of CnMCTRLm register set to 1.)
- Set to transmit message  
(RTR bit of CnMCONFm register is cleared to 0.)
- Transmission request is not set.  
(TRQ bit of CnMCTRLm register is cleared to 0.)

Upon acceptance of a remote frame, the following actions are executed if the ID of the received remote frame matches the ID of a message buffer that satisfies the above conditions.

- The DLC[3:0] bit string in the CnMDLCLm register store the received DLC value.
- The CnMDATA0m to CnMDATA7m registers in the data area are not updated (data before reception is saved).
- The DN bit of the CnMCTRLm register is set to 1.
- The CINTS1 bit of the CnINTS register is set to 1 (if the IE bit in the CnMCTRLm register of the message buffer that receives and stores the frame is set to 1).
- The receive completion interrupt (INTCnREC) is output (if the IE bit of the message buffer that receives and stores the frame is set to 1 and if the CIE1 bit of the CnIE register is set to 1).
- The message buffer number is recorded in the receive history list.

---

**Caution** When a message buffer is searched for receiving and storing a remote frame, overwrite control by the OWS bit of the CnMCONFm register of the message buffer and the DN bit of the CnMCTRLm register are not checked. The setting of OWS is ignored, and DN is set in any case.  
If more than one transmit message buffer has the same ID and the ID of the received remote frame matches that ID, the remote frame is stored in the transmit message buffer with the lowest message buffer number.

---

## 14.10 Message Transmission

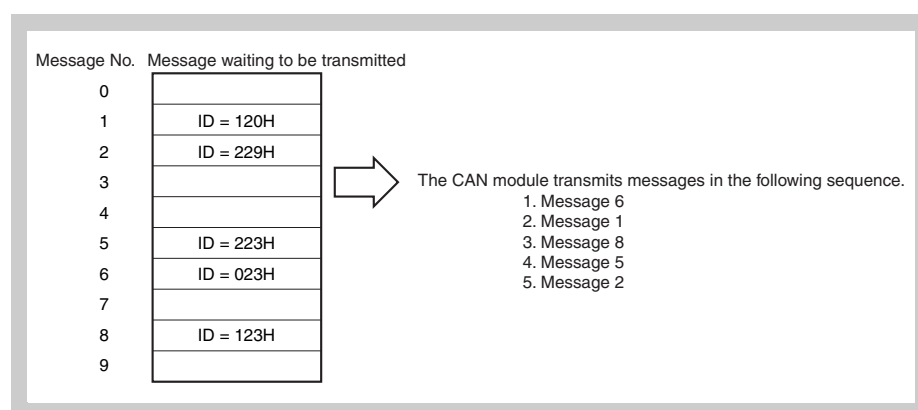
### 14.10.1 Message transmission

A message buffer with its TRQ bit set to 1 participates in the search for the most high-prioritized message when the following conditions are fulfilled. This behavior is valid for all operational modes.

- Used as a message buffer  
(MA0 bit of CnMCONFm register set to 1.)
- Set as a transmit message buffer  
(MT[2:0] bits of CnMCONFm register set to 000<sub>B</sub>.)
- Ready for transmission  
(RDY bit of CnMCTRLm register set to 1.)

The CAN system is a multi-master communication system. In a system like this, the priority of message transmission is determined based on message identifiers (IDs). To facilitate transmission processing by software when there are several messages awaiting transmission, the CAN module uses hardware to check the ID of the message with the highest priority and automatically identifies that message. This eliminates the need for software-based priority control.

Transmission priority is controlled by the identifier (ID).



**Figure 14-30** Message processing example

After the transmit message search, the transmit message with the highest priority of the transmit message buffers that have a pending transmission request (message buffers with the TRQ bit set to 1 in advance) is transmitted.

If a new transmission request is set, the transmit message buffer with the new transmission request is compared with the transmit message buffer with a pending transmission request. If the new transmission request has a higher priority, it is transmitted, unless transmission of a message with a low priority has already started. If transmission of a message with a low priority has already started, however, the new transmission request is transmitted later. To solve this priority inversion effect, the software can perform a transmission abort request for the lower priority message. The highest priority is determined according to the following rules.

Priority	Conditions	Description
1 (high)	Value of first 11 bits of ID [ID28 to ID18]:	The message frame with the lowest value represented by the first 11 bits of the ID is transmitted first. If the value of an 11-bit standard ID is equal to or smaller than the first 11 bits of a 29-bit extended ID, the 11-bit standard ID has a higher priority than a message frame with a 29-bit extended ID.
2	Frame type	A data frame with an 11-bit standard ID (RTR bit is cleared to 0) has a higher priority than a remote frame with a standard ID and a message frame with an extended ID.
3	ID type	A message frame with a standard ID (IDE bit is cleared to 0) has a higher priority than a message frame with an extended ID.
4	Value of lower 18 bits of ID [ID17 to ID0]:	If one or more transmission-pending extended ID message frame has equal values in the first 11 bits of the ID and the same frame type (equal RTR bit values), the message frame with the lowest value in the lower 18 bits of its extended ID is transmitted first.
5 (low)	Message buffer number	If two or more message buffers request transmission of message frames with the same ID, the message from the message buffer with the lowest message buffer number is transmitted first.

- Note** 1. If the automatic block transmission request bit ABTTRG is set to 1 in the normal operation mode with ABT, the TRQ bit is set to 1 only for one message buffer in the ABT message buffer group.

If the ABT mode was triggered by ABTTRG bit (1), one TRQ bit is set to 1 in the ABT area (buffer 0 through 7). Beyond this TRQ bit, the application can request transmissions (set TRQ bit to 1) for other TX-message buffers that do not belong to the ABT area. In that case an interval arbitration process (TX-search) evaluates all TX-message buffers with TRQ bit set to 1 and chooses the message buffer that contains the highest prioritized identifier for the next transmission. If there are 2 or more identifiers that have the highest priority (i.e. identical identifiers), the message located at the lowest message buffer number is transmitted at first.

Upon successful transmission of a message frame, the following operations are performed.

- The TRQ flag of the corresponding transmit message buffer is automatically cleared to 0.
  - The transmission completion status bit CINTS0 of the CnINTS register is set to 1 (if the interrupt enable bit (IE) of the corresponding transmit message buffer is set to 1).
  - An interrupt request signal INTCnTRX is output (if the CIE0 bit of the CnIE register is set to 1 and if the interrupt enable bit (IE) of the corresponding transmit message buffer is set to 1).
2. When changing the contents of a transmit buffer, the RDY flag of this buffer must be cleared before updating the buffer contents. As during internal transfer actions, the RDY flag may be locked temporarily, the status of RDY must be checked by software, after changing it.

### 14.10.2 Transmit history list function

The transmit history list (THL) function records in the transmit history list the number of the transmit message buffer from which data or remote frames have been sent. The THL consists of storage elements equivalent to up to seven messages, the last out-message pointer (LOPT) with the corresponding CnLOPT register, and the transmit history list get pointer (TGPT) with the corresponding CnTGPT register.

The THL is undefined immediately after the transition of the CAN module from the initialization mode to one of the operation modes.

The CnLOPT register holds the contents of the THL element indicated by the value of the LOPT pointer minus 1. By reading the CnLOPT register, therefore, the number of the message buffer that transmitted a data frame or remote frame first can be checked. The LOPT pointer is utilized as a write pointer that indicates to what part of the THL a message buffer number is recorded. Any time a data frame or remote frame is transmitted, the corresponding message buffer number is recorded to the THL element indicated by the LOPT pointer. Each time recording to the THL has been completed, the LOPT pointer is automatically incremented. In this way, the number of the message buffer that has received and stored a frame will be recorded chronologically.

The TGPT pointer is utilized as a read pointer that reads a recorded message buffer number from the THL. This pointer indicates the first THL element that the CPU has not yet read. By reading the CnTGPT register by software, the number of a message buffer that has completed transmission can be read. Each time a message buffer number is read from the CnTGPT register, the TGPT pointer is automatically incremented.

If the value of the TGPT pointer matches the value of the LOPT pointer, the THPM bit (transmit history list pointer match) of the CnTGPT register is set to 1. This indicates that no message buffer numbers that have not been read remain in the THL. If a new message buffer number is recorded, the LOPT pointer is incremented and because its value no longer matches the value of the TGPT pointer, the THPM bit is cleared. In other words, the numbers of the unread message buffers exist in the THL.

If the LOPT pointer is incremented and matches the value of the TGPT pointer minus 1, the TOVF bit (transmit history list overflow) of the CnTGPT register is set to 1. This indicates that the THL is full of message buffer numbers that have not been read. If a new message is received and stored, the message buffer number recorded last is overwritten by the message buffer number that transmitted its message afterwards. In this case, after the TOVF bit has been set (1), therefore, the recorded message buffer numbers in the THL do not completely reflect the chronological order. However the other transmitted messages can be found by a CPU search applied to all transmit message buffers unless the CPU has not overwritten a transmit object in one of these buffers beforehand. In total up to six transmission completions can occur without overflowing the THL.



**Caution** If the history list is in the overflow condition (TOVF is set), reading the history list contents is still possible, until the history list is empty (indicated by THPM flag set). Nevertheless, the history list remains in the overflow condition, until TOVF is cleared by software. If TOVF is not cleared, the THPM flag will also not be updated (cleared) upon successful transmission of a new message. This may lead to the situation, that THPM indicates an empty history list, although a successful transmission has taken place, while the history list is in the overflow state (TOVF and THPM are set).

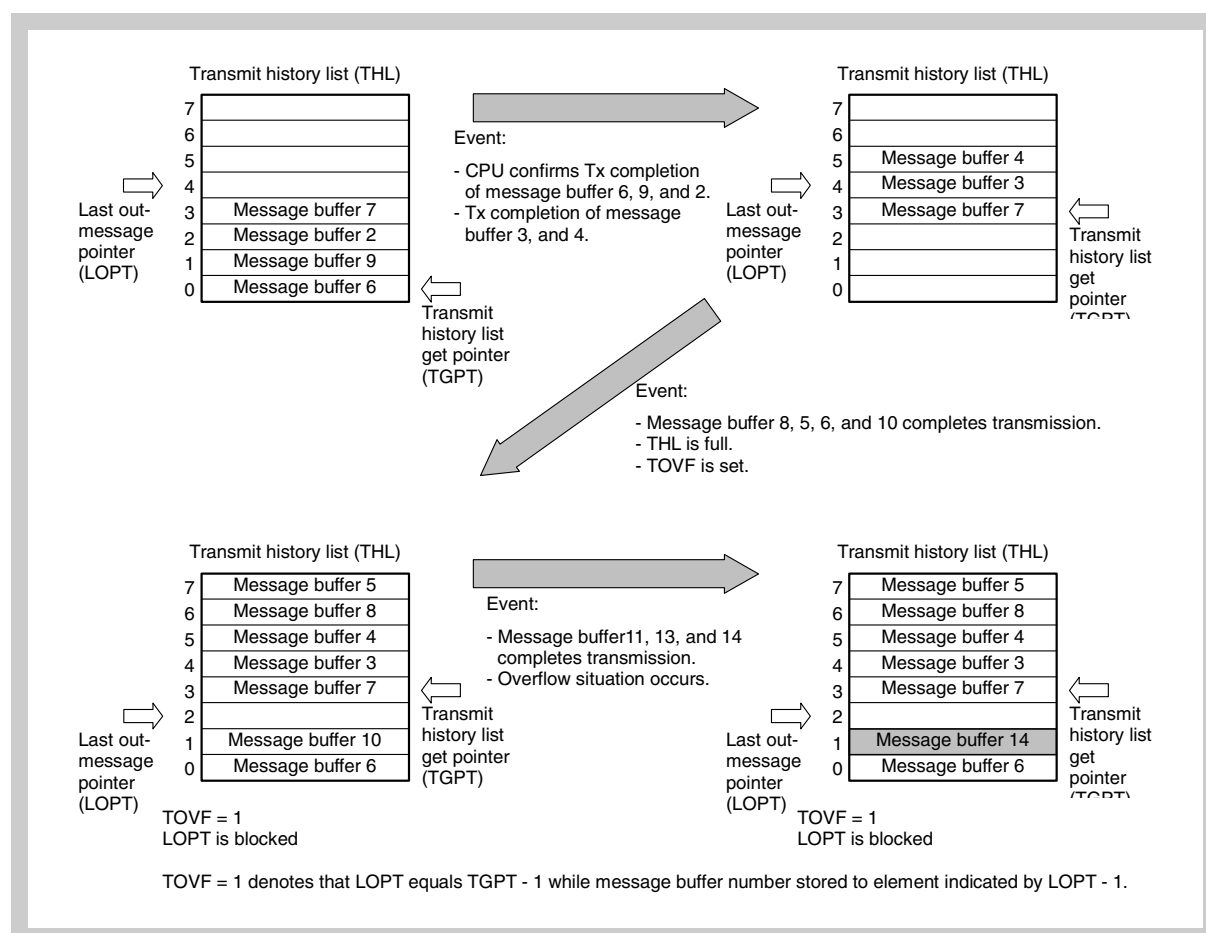


Figure 14-31 Transmit history list

### 14.10.3 Automatic block transmission (ABT)

The automatic block transmission (ABT) function is used to transmit two or more data frames successively with no CPU interaction. The maximum number of transmit message buffers assigned to the ABT function is eight (message buffer numbers 0 to 7).

By setting the OPMODE[2:0] bits of the CnCTRL register to 010<sub>B</sub>, “normal operation mode with automatic block transmission function” (hereafter referred to as ABT mode) can be selected.

To issue an ABT transmission request, define the message buffers by software first. Set the MA0 bit (1) in all the message buffers used for ABT, and define all the buffers as transmit message buffers by setting the MA[2:0] bits to 000<sub>B</sub>. Be sure to set the same ID for the message buffers for ABT even when that ID is being used for all the message buffers. To use two or more IDs, set the ID of each message buffer by using the CnMIDLm and CnMIDHm registers. Set the CnMDLCm and CnMDATA0m to CnMDATA7m registers before issuing a transmission request for the ABT function.

After initialization of message buffers for ABT is finished, the RDY bit needs to be set (1). In the ABT mode, the TRQ bit does not have to be manipulated by software.

After the data for the ABT message buffers has been prepared, set the ABTTRG bit to 1. Automatic block transmission is then started. When ABT is started, the TRQ bit in the first message buffer (message buffer 0) is automatically set to 1. After transmission of the data of message buffer 0 is finished, the TRQ bit of the next message buffer, message buffer 1, is set automatically. In this way, transmission is executed successively.

A delay time can be inserted by program in the interval in which the transmission request (TRQ) is automatically set while successive transmission is being executed. The delay time to be inserted is defined by the CnGMABTD register. The unit of the delay time is DBT (data bit time). DBT depends on the setting of the CnBRP and CnBTR registers.

Among transmit objects within the ABT-area, the priority of the transmission ID is not evaluated. The data of message buffers 0 to 7 are sequentially transmitted. When transmission of the data frame from message buffer 7 has been completed, the ABTTRG bit is automatically cleared to 0 and the ABT operation is finished.

If the RDY bit of an ABT message buffer is cleared during ABT, no data frame is transmitted from that buffer, ABT is stopped, and the ABTTRG bit is cleared. After that, transmission can be resumed from the message buffer where ABT stopped, by setting the RDY and ABTTRG bits to 1 by software. To not resume transmission from the message buffer where ABT stopped, the internal ABT engine can be reset by setting the ABTCLR bit to 1 while ABT mode is stopped and the ABTTRG bit is cleared to 0. In this case, transmission is started from message buffer 0 if the ABTCLR bit is cleared to 0 and then the ABTTRG bit is set to 1.

An interrupt can be used to check if data frames have been transmitted from all the message buffers for ABT. To do so, the IE bit of the CnMCTRLm register of each message buffer except the last message buffer needs to be cleared (0).

If a transmit message buffer other than those used by the ABT function (message buffers 8 to 31) is assigned to a transmit message buffer, the message to be transmitted next is determined by the priority of the transmission ID of the ABT message buffer whose transmission is currently

held pending and the transmission ID of the message buffers other than those used by the ABT function.

Transmission of a data frame from an ABT message buffer is not recorded in the transmit history list (THL).

- 
- Caution**
1. Set the ABTCLR bit to 1 while the ABTTRG bit is cleared to 0 in order to resume ABT operation at buffer No.0. If the ABTCLR bit is set to 1 while the ABTTRG bit is set to 1, the subsequent operation is not guaranteed.
  2. If the automatic block transmission engine is cleared by setting the ABTCLR bit to 1, the ABTCLR bit is automatically cleared immediately after the processing of the clearing request is completed.
  3. Do not set the ABTTRG bit in the initialization mode. If the ABTTRG bit is set in the initialization mode, the proper operation is not guaranteed after the mode is changed from the initialization mode to the ABT mode.
  4. Do not set the TRQ bit of the ABT message buffers to 1 by software in the normal operation mode with ABT. Otherwise, the operation is not guaranteed.
  5. The CnGMABTD register is used to set the delay time that is inserted in the period from completion of the preceding ABT message to setting of the TRQ bit for the next ABT message when the transmission requests are set in the order of message numbers for each message for ABT that is successively transmitted in the ABT mode. The timing at which the messages are actually transmitted onto the CAN bus varies depending on the status of transmission from other stations and the status of the setting of the transmission request for messages other than the ABT messages (message buffers 8 to 31).
  6. If a transmission request is made for a message other than an ABT message and if no delay time is inserted in the interval in which transmission requests for ABT are automatically set (CnGMABTD register = 00<sub>H</sub>), messages other than ABT messages may be transmitted not depending on their priority compared to the priority of the ABT message.
  7. Do not clear the RDY bit to 0 when the ABTTRG bit = 1.
  8. If a message is received from another node while normal operation mode with ABT is active, the TX-message from the ABT-area may be transmitted with delay of one frame although CnGMABTD register was set up with 00<sub>H</sub>.
-

### 14.10.4 Transmission abort process

**(1) Transmission abort process except for in normal operation mode with automatic block transmission (ABT)**

The user can clear the TRQ bit of the CnMCTRLm register to 0 to abort a transmission request. The TRQ bit will be cleared immediately if the abort was successful. Whether the transmission was successfully aborted or not can be checked using the TSTAT bit of the CnCTRL register and the CnTGPT register, which indicate the transmission status on the CAN bus (for details, refer to the processing in *Figure 14-45 on page 489*).

**(2) Transmission abort process except for ABT transmission in normal operation mode with automatic block transmission (ABT)**

The user can clear the ABTTRG bit of the CnGMABT register to 0 to abort a transmission request. After checking the ABTTRG bit of the CnGMABT register = 0, clear the TRQ bit of the CnMCTRLm register to 0. The TRQ bit will be cleared immediately if the abort was successful. Whether the transmission was successfully aborted or not can be checked using the TSTAT bit of the CnCTRL register and the CnTGPT register, which indicate the transmission status on the CAN bus (for details, refer to the processing in *Figure 14-46 on page 490*).

**(3) Transmission abort process for ABT transmission in normal operation mode with automatic block transmission (ABT)**

To abort ABT that is already started, clear the ABTTRG bit of the CnGMABT register to 0. In this case, the ABTTRG bit remains 1 if an ABT message is currently being transmitted and until the transmission is completed (successfully or not), and is cleared to 0 as soon as transmission is finished. This aborts ABT.

If the last transmission (before ABT) was successful, the normal operation mode with ABT is left with the internal ABT pointer pointing to the next message buffer to be transmitted.

In the case of an erroneous transmission, the position of the internal ABT pointer depends on the status of the TRQ bit in the last transmitted message buffer. If the TRQ bit is set to 1 when clearing the ABTTRG bit is requested, the internal ABT pointer points to the last transmitted message buffer (for details, refer to the process in *Figure 14-47 on page 491*). If the TRQ bit is cleared to 0 when clearing the ABTTRG bit is requested, the internal ABT pointer is incremented (+1) and points to the next message buffer in the ABT area (for details, refer to the process in *Figure 14-48 on page 492*).

---

**Caution** Be sure to abort ABT by clearing ABTTRG bit to 0. The operation is not guaranteed if aborting transmission is requested by clearing RDY.

---

When the normal operation mode with ABT is resumed after ABT has been aborted and the ABTTRG bit is set to 1, the next ABT message buffer to be transmitted can be determined from the following table.

Status of TRQ of ABT message buffer	Abort after successful transmission	Abort after erroneous transmission
Set (1)	Next message buffer in the ABT area <sup>a</sup>	Same message buffer in the ABT area
Cleared (0)	Next message buffer in the ABT area <sup>a</sup>	Next message buffer in the ABT area <sup>a</sup>

- <sup>a)</sup> The above resumption operation can be performed only if a message buffer ready for ABT exists in the ABT area. For example, an abort request that is issued while ABT of message buffer 7 is in progress is regarded as completion of ABT, rather than abort, if transmission of message buffer 7 has been successfully completed, even if the ABTTRG bit is cleared to 0. If the RDY bit in the next message buffer in the ABT area is cleared to 0, the internal ABT pointer is retained, but the resumption operation is not performed even if the ABTTRG bit is set to 1, and ABT ends immediately.

### 14.10.5 Remote frame transmission

Remote frames can be transmitted only from transmit message buffers. Set whether a data frame or remote frame is transmitted via the RTR bit of the CnMCONFm register. Setting (1) the RTR bit sets remote frame transmission.

## 14.11 Power Saving Modes

### 14.11.1 CAN sleep mode

The CAN sleep mode can be used to set the CAN Controller to stand-by mode in order to reduce power consumption. The CAN module can enter the CAN sleep mode from all operation modes. Release of the CAN sleep mode returns the CAN module to exactly the same operation mode from which the CAN sleep mode was entered.

In the CAN sleep mode, the CAN module does not transmit messages, even when transmission requests are issued or pending.

#### (1) Entering CAN sleep mode

The CPU issues a CAN sleep mode transition request by writing 01<sub>B</sub> to the PSMODE[1:0] bits of the CnCTRL register.

This transition request is acknowledged only under the following conditions.

1. The CAN module is already in one of the following operation modes
  - Normal operation mode
  - Normal operation mode with ABT
  - Receive-only mode
  - Single-shot mode
  - Self-test mode
  - CAN stop mode in all the above operation modes
2. The CAN bus state is bus idle (the 4th bit in the interframe space is recessive).  
If the CAN bus is fixed to dominant, the request for transition to the CAN sleep mode is held pending. Also the transition from CAN stop mode to CAN sleep mode is independent of the CAN bus state.
3. No transmission request is pending

**Note** If a sleep mode request is pending, and at the same time a message is received in a message box, the sleep mode request is not cancelled, but is executed right after message storage has been finished. This may result in AFCAN being in sleep mode, while the CPU would execute the RX interrupt routine. Therefore, the interrupt routine must check the access to the message buffers as well as reception history list registers by using the MBON flag, if sleep mode is used.

If any one of the conditions mentioned above is not met, the CAN module will operate as follows.

- If the CAN sleep mode is requested from the initialization mode, the CAN sleep mode transition request is ignored and the CAN module remains in the initialization mode.
- If the CAN bus state is not bus idle (i.e., the CAN bus state is either transmitting or receiving) when the CAN sleep mode is requested in one of the operation modes, immediate transition to the CAN sleep mode is not possible. In this case, the CAN sleep mode transition request has to be held pending until the CAN bus state becomes bus idle (the 4th bit in the interframe space is recessive). In the time from the CAN sleep mode request to successful transition, the PSMODE[1:0] bits remain 00<sub>B</sub>. When the module has entered the CAN sleep mode, the PSMODE[1:0] bits are set

to 01<sub>B</sub>.

- If a request for transition to the initialization mode and a request for transition to the CAN sleep mode are made at the same time while the CAN module is in one of the operation modes, the request for the initialization mode is enabled. The CAN module enters the initialization mode at a predetermined timing. At this time, the CAN sleep mode request is not held pending and is ignored.
- Even when initialization mode and sleep mode are not requested simultaneously (i.e the first request has not been granted while the second request is made), the request for initialization has priority over the sleep mode request. The sleep mode request is cancelled when the initialization mode is requested. When a pending request for initialization mode is present, a subsequent request for Sleep mode request is cancelled right at the point in time where it was submitted.

## (2) Status in CAN sleep mode

The CAN module is in the following state after it enters the CAN sleep mode:

- The internal operating clock is stopped and the power consumption is minimized.
- The function to detect the falling edge of the CAN reception pin (CRXDn) remains in effect to wake up the CAN module from the CAN bus.
- To wake up the CAN module from the CPU, data can be written to the PSMODE[1:0] bits of the CAN module control register (CnCTRL), but nothing can be written to other CAN module registers or bits.
- The CAN module registers can be read, except for the CnLIPT, CnRGPT, CnLOPT, and CnTGPT registers.
- The CAN message buffer registers cannot be written or read.
- MBON bit of the CAN Global Control register (CnGMCTRL) is cleared.
- A request for transition to the initialization mode is not acknowledged and is ignored.

---

**(3) Releasing CAN sleep mode**

The CAN sleep mode is released by the following events:

- When the CPU writes 00<sub>B</sub> to the PSMODE[1:0] bits of the CnCTRL register
- A falling edge at the CAN reception pin (CRXDn) (i.e. the CAN bus level shifts from recessive to dominant)

---

**Caution** Even if the falling edge belongs to the SOF of a receive message, this message will not be received and stored. If the CPU has turned off the clock supply to the CAN module while the CAN module was in sleep mode, even subsequently the CAN sleep mode will not be released and PSMODE[1:0] will remain 01<sub>B</sub> unless the clock to the CAN module is supplied again. In addition to this, the receive message will not be received after that.

---

After releasing the sleep mode, the CAN module returns to the operation mode from which the CAN sleep mode was requested and the PSMODE[1:0] bits of the CnCTRL register must be reset by software to 00<sub>B</sub>. If the CAN sleep mode is released by a change in the CAN bus state, the CINTS5 bit of the CnINTS register is set to 1, regardless of the CIE bit of the CnIE register. After the CAN module is released from the CAN sleep mode, it participates in the CAN bus again by automatically detecting 11 consecutive recessive-level bits on the CAN bus. The user application has to wait until MBON = 1, before accessing message buffers again.

When a request for transition to the initialization mode is made while the CAN module is in the CAN sleep mode, that request is ignored; the CAN module has to be released from sleep mode by software first before entering the initialization mode.

---

**Caution**

1. Be aware that the release of CAN sleep mode by CAN bus event, and thus the wake up interrupt may happen at any time, even right after requesting sleep mode, if a CAN bus event occurs.
2. Always reset the PSMODE[1:0] bits to 00<sub>B</sub>, when waking up from CAN sleep mode, before accessing any other registers of the CAN module.
3. Always clear the interrupt flag CINTS5, when waking up from CAN sleep mode.

---



### 14.11.2 CAN stop mode

The CAN stop mode can be used to set the CAN Controller to stand-by mode to reduce power consumption. The CAN module can enter the CAN stop mode only from the CAN sleep mode. Release of the CAN stop mode puts the CAN module in the CAN sleep mode.

The CAN stop mode can only be released (entering CAN sleep mode) by writing 01<sub>B</sub> to the PSMODE[1:0] bits of the CnCTRL register and not by a change in the CAN bus state. No message is transmitted even when transmission requests are issued or pending.

#### (1) Entering CAN stop mode

A CAN stop mode transition request is issued by writing 11<sub>B</sub> to the PSMODE[1:0] bits of the CnCTRL register.

A CAN stop mode request is only acknowledged when the CAN module is in the CAN sleep mode. In all other modes, the request is ignored.

---

**Caution** To set the CAN module to the CAN stop mode, the module must be in the CAN sleep mode. To confirm that the module is in the sleep mode, check that the PSMODE[1:0] bits = 01<sub>B</sub>, and then request the CAN stop mode. If a bus change occurs at the CAN reception pin (CRXDn) while this process is being performed, the CAN sleep mode is automatically released. In this case, the CAN stop mode transition request cannot be acknowledged.

---

#### (2) Status in CAN stop mode

The CAN module is in the following state after it enters the CAN stop mode.

- The internal operating clock is stopped and the power consumption is minimized.
- To wake up the CAN module from the CPU, data can be written to the PSMODE[1:0] bits of the CAN module control register (CnCTRL), but nothing can be written to other CAN module registers or bits.
- The CAN module registers can be read, except for the CnLIPT, CnRGPT, CnLOPT, and CnTGPT registers.
- The CAN message buffer registers cannot be written or read.
- MBON bit of the CAN Global Control register (CnGMCTRL) is cleared.
- An initialization mode transition request is not acknowledged and is ignored.

#### (3) Releasing CAN stop mode

The CAN stop mode can only be released by writing 01<sub>B</sub> to the PSMODE[1:0] bits of the CnCTRL register. After releasing the CAN stop mode, the CAN module enters the CAN sleep mode.

When the initialization mode is requested while the CAN module is in the CAN stop mode, that request is ignored; the CPU has to release the stop mode and subsequently CAN sleep mode before entering the initialization mode. It is impossible to enter the other operation mode directly from the CAN stop mode not entering the CAN sleep mode, that request is ignored.

### 14.11.3 Example of using power saving modes

In some application systems, it may be necessary to place the CPU in a power saving mode to reduce the power consumption. By using the power saving mode specific to the CAN module and the power saving mode specific to the CPU in combination, the CPU can be woken up from the power saving status by the CAN bus.

Here is an example for using the power saving modes.

- First, put the CAN module in the CAN sleep mode (PSMODE[1:0] = 01<sub>B</sub>). Next, put the CPU in the power saving mode. If an edge transition from recessive to dominant is detected at the CAN reception pin (CRXDn) in this status, the CINTS5 bit in the CAN module is set to 1. If the CIE5 bit of the CnCTRL register is set to 1, a wakeup interrupt (INTWUPn) is generated.
- The CAN module is automatically released from CAN sleep mode (PSMODE = 00<sub>B</sub>) and returns to normal operation mode.
- The CPU, in response to INTWUPn, can release its own power saving mode and return to normal operation mode.

To further reduce the power consumption of the CPU, the internal clock - including that of the CAN module - may be stopped. In this case, the operating clock supplied to the CAN module is stopped after the CAN module has been put in CAN sleep mode. Then the CPU enters a power saving mode in which the clock supplied to the CPU is stopped.

- If an edge transition from recessive to dominant is detected at the CAN reception pin (CRXDn) in this status, the CAN module can set the CINTS5 bit to 1 and generate the wakeup interrupt (INTWUPn) even if it is not supplied with the clock.
- The other functions, however, do not operate, because clock supply to the CAN module is stopped, and the module remains in CAN sleep mode.
- The CPU, in response to INTWUPn
  - releases its power saving mode,
  - resumes supply of the internal clocks - including the clock to the CAN module - after the oscillation stabilization time has elapsed, and
  - starts instruction execution.
- The CAN module is immediately released from the CAN sleep mode when clock supply is resumed, and returns to the normal operation mode (PSMODE = 00<sub>B</sub>).

## 14.12 Interrupt Function

The CAN module provides 6 different interrupt sources.

The occurrence of these interrupt sources is stored in interrupt status registers. Four separate interrupt request signals are generated from the six interrupt sources. When an interrupt request signal that corresponds to two or more interrupt sources is generated, the interrupt sources can be identified by using an interrupt status register. After an interrupt source has occurred, the corresponding interrupt status bit must be cleared to 0 by software.

**Table 14-25 List of CAN module interrupt sources**

No.	Interrupt status bit		Interrupt enable bit		Interrupt request signal	Interrupt source description
	Name	Register	Name	Register		
1	CINTS0	CnINTS	CIE0 <sup>a</sup>	CnIE	INTCnTRX	Message frame successfully transmitted from message buffer m
2	CINTS1	CnINTS	CIE1 <sup>a</sup>	CnIE	INTCnREC	Valid message frame reception in message buffer m
3	CINTS2	CnINTS	CIE2	CnIE	INTCnERR	CAN module error state interrupt (Supplement 1)
4	CINTS3	CnINTS	CIE3	CnIE		CAN module protocol error interrupt (Supplement 2)
5	CINTS4	CnINTS	CIE4	CnIE		CAN module arbitration loss interrupt
6	CINTS5	CnINTS	CIE5	CnIE	INTCnWUP	CAN module wakeup interrupt from CAN sleep mode (Supplement 3)

<sup>a)</sup> The IE bit (message buffer interrupt enable bit) in the CnMCTRL register of the corresponding message buffer has to be set to 1 for that message buffer to participate in the interrupt generation process.

- Supplements**
1. This interrupt is generated when the transmission/reception error counter is at the warning level, or in the error passive or bus-off state.
  2. This interrupt is generated when a stuff error, form error, ACK error, bit error, or CRC error occurs.
  3. This interrupt is generated when the CAN module is woken up from the CAN sleep mode because a falling edge is detected at the CAN reception pin (CAN bus transition from recessive to dominant).

## 14.13 Diagnosis Functions and Special Operational Modes

The CAN module provides a receive-only mode, single-shot mode, and self-test mode to support CAN bus diagnosis functions or the operation of special CAN communication methods.

### 14.13.1 Receive-only mode

The receive-only mode is used to monitor receive messages without causing any interference on the CAN bus and can be used for CAN bus analysis nodes.

For example, this mode can be used for automatic baud-rate detection. The baud rate in the CAN module is changed until “valid reception” is detected, so that the baud rates in the module match (“valid reception” means a message frame has been received in the CAN protocol layer without occurrence of an error and with an appropriate ACK between nodes connected to the CAN bus). A valid reception does not require message frames to be stored in a receive message buffer (data frames) or transmit message buffer (remote frames). The event of valid reception is indicated by setting the VALID bit of the CnCTRL register (1).

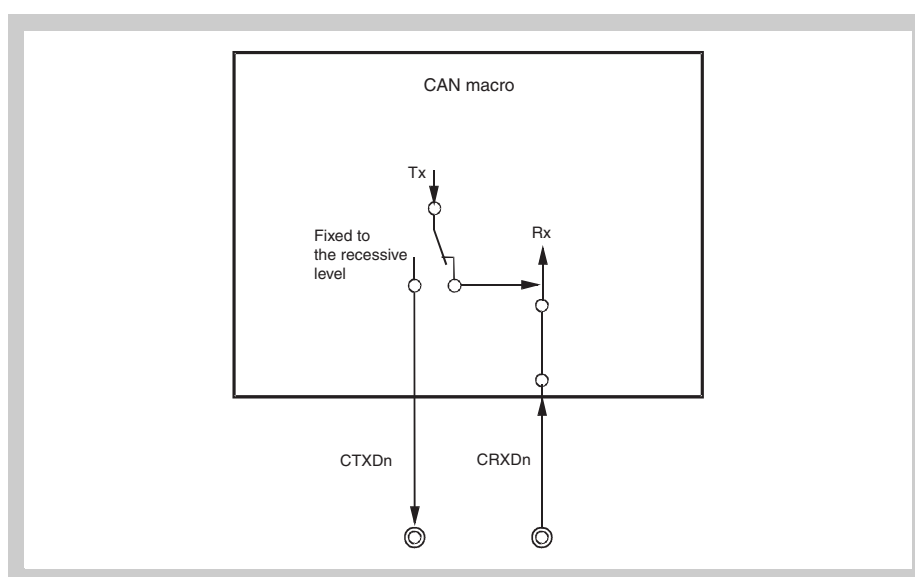


Figure 14-32 CAN module terminal connection in receive-only mode

In the receive-only mode, no message frames can be transmitted from the CAN module to the CAN bus. Transmit requests issued for message buffers defined as transmit message buffers are held pending.

In the receive-only mode, the CAN transmission pin (CTXDn) in the CAN module is fixed to the recessive level. Therefore, no active error flag can be transmitted from the CAN module to the CAN bus even when a CAN bus error is detected while receiving a message frame. Since no transmission can be issued from the CAN module, the transmission error counter the CnERC.TEC7 to CnERC.TEC0 bits are never updated. Therefore, a CAN module in the receive-only mode does not enter the bus-off state.

Furthermore, in the receive-only mode ACK is not returned to the CAN bus in this mode upon the valid reception of a message frame. Internally, the local

node recognizes that it has transmitted ACK. An overload frame cannot be transmitted to the CAN bus.

---

**Caution** If only two CAN nodes are connected to the CAN bus and one of them is operating in the receive-only mode, there is no ACK on the CAN bus. Due to the missing ACK, the transmitting node will transmit an active error flag, and repeat transmitting a message frame. The transmitting node becomes error passive after transmitting the message frame 16 times (assuming that the error counter was 0 in the beginning and no other errors have occurred). After the message frame for the 17th time is transmitted, the transmitting node generates a passive error flag. The receiving node in the receive-only mode detects the first valid message frame at this point, and the VALID bit is set to 1 for the first time.

---

### 14.13.2 Single-shot mode

In the single-shot mode, automatic re-transmission as defined in the CAN protocol is switched off. (According to the CAN protocol, a message frame transmission that has been aborted by either arbitration loss or error occurrence has to be repeated without control by software.) All other behavior of single shot mode is identical to normal operation mode. Features of single shot mode can not be used in combination with normal mode with ABT.

The single-shot mode disables the re-transmission of an aborted message frame transmission according to the setting of the AL bit of the CnCTRL register. When the AL bit is cleared to 0, re-transmission upon arbitration loss and upon error occurrence is disabled. If the AL bit is set to 1, re-transmission upon error occurrence is disabled, but re-transmission upon arbitration loss is enabled. As a consequence, the TRQ bit in a message buffer defined as a transmit message buffer is cleared to 0 by the following events:

- Successful transmission of the message frame
- Arbitration loss while sending the message frame
- Error occurrence while sending the message frame

The events arbitration loss and error occurrence can be distinguished by checking the CINTS4 and CINTS3 bits of the CnINTS register respectively, and the type of the error can be identified by reading the LEC[2:0] bits of the CnLEC register.

Upon successful transmission of the message frame, the transmit completion interrupt bit CINTS0 of the CnINTS register is set to 1. If the CIE0 bit of the CnIE register is set to 1 at this time, an interrupt request signal is output.

The single-shot mode can be used when emulating time-triggered communication methods (e.g., TTCAN level 1).

---

**Caution** The AL bit is only valid in single-shot mode. It does not influence the operation of re-transmission upon arbitration loss in the other operation modes.

---

### 14.13.3 Self-test mode

In the self-test mode, message frame transmission and message frame reception can be tested without connecting the CAN node to the CAN bus or without affecting the CAN bus.

In the self-test mode, the CAN module is completely disconnected from the CAN bus, but transmission and reception are internally looped back. The CAN transmission pin (CTXDn) is fixed to the recessive level.

If the falling edge on the CAN reception pin (CRXDn) is detected after the CAN module has entered the CAN sleep mode from the self-test mode, however, the module is released from the CAN sleep mode in the same manner as the other operation modes. To keep the module in the CAN sleep mode, use the CAN reception pin (CRXDn) as a port pin.

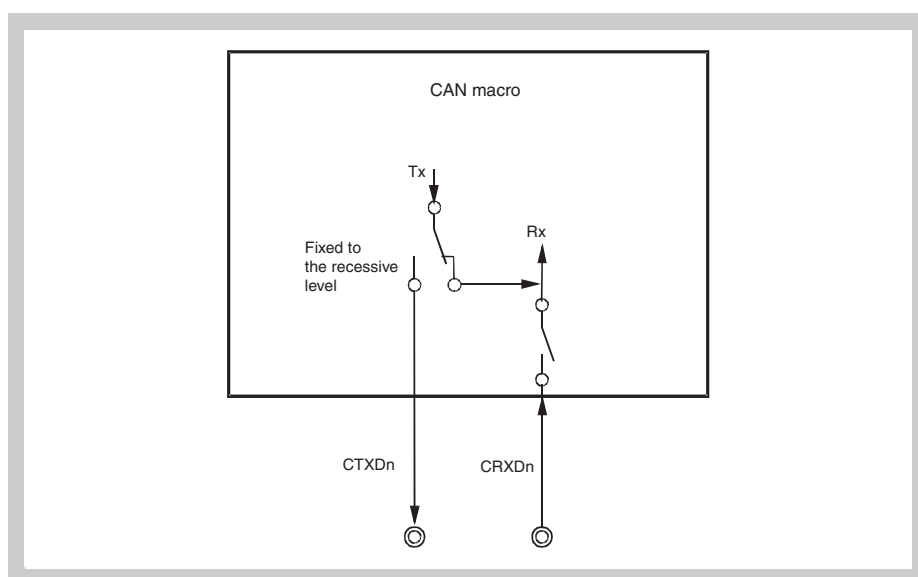


Figure 14-33 CAN module terminal connection in self-test mode

### 14.13.4 Receive/transmit operation in each operation mode

The following table shows outline of the receive/transmit operation in each operation mode.

**Table 14-26 Outline of the receive/transmit in each operation mode**

Operation mode	Transmission of data/remote frame	Transmission of ACK	Transmission of error/over load frame	Transmission retry	Automatic block transmission (ABT)	Set of VALID bit	Store data to message buffer
Initialization mode	No	No	No	No	No	No	No
Normal operation mode	Yes	Yes	Yes	Yes	No	Yes	Yes
Normal operation mode with ABT	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Receive only mode	No	No	No	No	No	Yes	Yes
Single-shot mode	Yes	Yes	Yes	No <sup>a</sup>	No	Yes	Yes
Self-test mode	Yes <sup>b</sup>	Yes <sup>b</sup>	Yes <sup>b</sup>	Yes <sup>b</sup>	No	Yes <sup>b</sup>	Yes <sup>b</sup>

a) When the arbitration lost occurs, control of re-transmission is possible by the AL bit of CnCTRL register.

b) Each signals are not generated to outside, but generated into the CAN module.

## 14.14 Time Stamp Function

CAN is an asynchronous, serial protocol. All nodes connected to the CAN bus have a local, autonomous clock. As a consequence, the clocks of the nodes have no relation (i.e., the clocks are asynchronous and may have different frequencies).

In some applications, however, a common time base over the network (= global time base) is needed. In order to build up a global time base, a time stamp function is used. The essential mechanism of a time stamp function is the capture of timer values triggered by signals on the CAN bus.

### 14.14.1 Time stamp function

The CAN Controller supports the capturing of timer values triggered by a specific frame. An on-chip 16-bit capture timer unit in a microcontroller system is used in addition to the CAN Controller. The 16-bit capture timer unit captures the timer value according to a trigger signal (TSOUT) for capturing that is output when a data frame is received from the CAN Controller. The CPU can retrieve the time of occurrence of the capture event, i.e., the time stamp of the message received from the CAN bus, by reading the captured value. The TSOUT signal can be selected from the following two event sources and is specified by the TSSEL bit of the CnTS register.

- SOF event (start of frame) (TSSEL = 0)
- EOF event (last bit of end of frame) (TSSEL = 1)

The TSOUT signal is enabled by setting the TSEN bit of the CnTS register to 1.

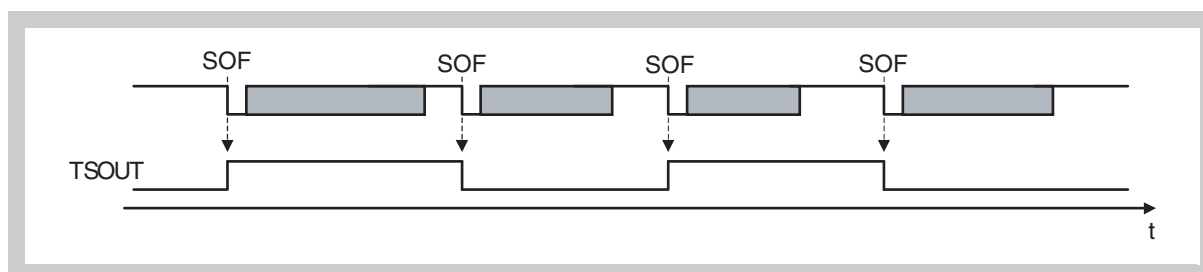


Figure 14-34 Timing diagram of capture signal TSOUT

The TSOUT signal toggles its level upon occurrence of the selected event during data frame reception (in *Figure 14-34*, the SOF is used as the trigger event source). To capture a timer value by using the TSOUT signal, the capture timer unit must detect the capture signal at both the rising edge and falling edge.

This time stamp function is controlled by the TSLOCK bit of the CnTS register. When TSLOCK is cleared to 0, the TSOUT signal toggles upon occurrence of the selected event. If TSLOCK is set to 1, the TSOUT signal toggles upon occurrence of the selected event, but the toggle is stopped as the TSEN bit is automatically cleared to 0 as soon as the message storing to the message buffer 0 starts. This suppresses the subsequent toggle occurrence by the TSOUT signal, so that the time stamp value toggled last (= captured last) can be saved as the time stamp value of the time at which the data frame was received in message buffer 0.



---

**Caution** The time stamp function using the TSLOCK bit stops toggle of the TSOUT signal by receiving a data frame in message buffer 0. Therefore, message buffer 0 must be set as a receive message buffer. Since a receive message buffer cannot receive a remote frame, toggle of the TSOUT signal cannot be stopped by reception of a remote frame. Toggle of the TSOUT signal does not stop when a data frame is received in a message buffer other than message buffer 0.

For these reasons, a data frame cannot be received in message buffer 0 when the CAN module is in the normal operation mode with ABT, because message buffer 0 must be set as a transmit message buffer. In this operation mode, therefore, the function to stop toggle of the TSOUT signal by the TSLOCK bit cannot be used.

---

## 14.15 Baud Rate Settings

### 14.15.1 Baud rate setting conditions

Make sure that the settings are within the range of limit values for ensuring correct operation of the CAN Controller, as follows.

- $5TQ \leq SPT$  (sampling point)  $\leq 17 TQ$   
 $SPT = TSEG1 + 1$
- $8 TQ \leq DBT$  (data bit time)  $\leq 25 TQ$   
 $DBT = TSEG1 + TSEG2 + 1TQ = TSEG2 + SPT$
- $1 TQ \leq SJW$  (synchronization jump width)  $\leq 4TQ$   
 $SJW \leq DBT - SPT$
- $4 \leq TSEG1 \leq 16$  [ $3 \leq$  Setting value of TSEG1[3:0]  $\leq 15$ ]
- $1 \leq TSEG2 \leq 8$  [ $0 \leq$  Setting value of TSEG2[2:0]  $\leq 7$ ]

- Note**
1.  $TQ = 1/f_{TQ}$  ( $f_{TQ}$ : CAN protocol layer basic system clock)
  2. TSEG1[3:0] (Bits 3 to 0 of CAN bit rate register (CnBTR))
  3. TSEG2[2:0] (Bits 10 to 8 of CAN bit rate register (CnBTR))

Table 14-27 shows the combinations of bit rates that satisfy the above conditions.

**Table 14-27 Settable bit rate combinations (1/3)**

Valid bit rate setting					CnBTR register setting value		Sampling point (unit %)
DBT length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1 [3:0]	TSEG2 [2:0]	
25	1	8	8	8	1111	111	68.0
24	1	7	8	8	1110	111	66.7
24	1	9	7	7	1111	110	70.8
23	1	6	8	8	1101	111	65.2
23	1	8	7	7	1110	110	69.6
23	1	10	6	6	1111	101	73.9
22	1	5	8	8	1100	111	63.6
22	1	7	7	7	1101	110	68.2
22	1	9	6	6	1110	101	72.7
22	1	11	5	5	1111	100	77.3
21	1	4	8	8	1011	111	61.9
21	1	6	7	7	1100	110	66.7
21	1	8	6	6	1101	101	71.4
21	1	10	5	5	1110	100	76.2
21	1	12	4	4	1111	011	81.0
20	1	3	8	8	1010	111	60.0
20	1	5	7	7	1011	110	65.0
20	1	7	6	6	1100	101	70.0
20	1	9	5	5	1101	100	75.0
20	1	11	4	4	1110	011	80.0
20	1	13	3	3	1111	010	85.0
19	1	2	8	8	1001	111	57.9
19	1	4	7	7	1010	110	63.2
19	1	6	6	6	1011	101	68.4
19	1	8	5	5	1100	100	73.7
19	1	10	4	4	1101	011	78.9
19	1	12	3	3	1110	010	84.2
19	1	14	2	2	1111	001	89.5
18	1	1	8	8	1000	111	55.6
18	1	3	7	7	1001	110	61.1
18	1	5	6	6	1010	101	66.7
18	1	7	5	5	1011	100	72.2
18	1	9	4	4	1100	011	77.8
18	1	11	3	3	1101	010	83.3
18	1	13	2	2	1110	001	88.9
18	1	15	1	1	1111	000	94.4
17	1	2	7	7	1000	110	58.8

Table 14-27 Settable bit rate combinations (2/3)

Valid bit rate setting					CnBTR register setting value		Sampling point (unit %)
DBT length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1 [3:0]	TSEG2 [2:0]	
17	1	4	6	6	1001	101	64.7
17	1	6	5	5	1010	100	70.6
17	1	8	4	4	1011	011	76.5
17	1	10	3	3	1100	010	82.4
17	1	12	2	2	1101	001	88.2
17	1	14	1	1	1110	000	94.1
16	1	1	7	7	0111	110	56.3
16	1	3	6	6	1000	101	62.5
16	1	5	5	5	1001	100	68.8
16	1	7	4	4	1010	011	75.0
16	1	9	3	3	1011	010	81.3
16	1	11	2	2	1100	001	87.5
16	1	13	1	1	1101	000	93.8
15	1	2	6	6	0111	101	60.0
15	1	4	5	5	1000	100	66.7
15	1	6	4	4	1001	011	73.3
15	1	8	3	3	1010	010	80.0
15	1	10	2	2	1011	001	86.7
15	1	12	1	1	1100	000	93.3
14	1	1	6	6	0110	101	57.1
14	1	3	5	5	0111	100	64.3
14	1	5	4	4	1000	011	71.4
14	1	7	3	3	1001	010	78.6
14	1	9	2	2	1010	001	85.7
14	1	11	1	1	1011	000	92.9
13	1	2	5	5	0110	100	61.5
13	1	4	4	4	0111	011	69.2
13	1	6	3	3	1000	010	76.9
13	1	8	2	2	1001	001	84.6
13	1	10	1	1	1010	000	92.3
12	1	1	5	5	0101	100	58.3
12	1	3	4	4	0110	011	66.7
12	1	5	3	3	0111	010	75.0

Table 14-27 Settable bit rate combinations (3/3)

Valid bit rate setting					CnBTR register setting value		Sampling point (unit %)
DBT length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	TSEG1 [3:0]	TSEG2 [2:0]	
12	1	7	2	2	1000	001	83.3
12	1	9	1	1	1001	000	91.7
11	1	2	4	4	0101	011	63.6
11	1	4	3	3	0110	010	72.7
11	1	6	2	2	0111	001	81.8
11	1	8	1	1	1000	000	90.9
10	1	1	4	4	0100	011	60.0
10	1	3	3	3	0101	010	70.0
10	1	5	2	2	0110	001	80.0
10	1	7	1	1	0111	000	90.0
9	1	2	3	3	0100	010	66.7
9	1	4	2	2	0101	001	77.8
9	1	6	1	1	0110	000	88.9
8	1	1	3	3	0011	010	62.5
8	1	3	2	2	0100	001	75.0
8	1	5	1	1	0101	000	87.5
7 <sup>a</sup>	1	2	2	2	0011	001	71.4
7 <sup>a</sup>	1	4	1	1	0100	000	85.7
6 <sup>a</sup>	1	1	2	2	0010	001	66.7
6 <sup>a</sup>	1	3	1	1	0011	000	83.3
5 <sup>a</sup>	1	2	1	1	0010	000	80.0
4 <sup>a</sup>	1	1	1	1	0001	000	75.0

a) Setting with a DBT value of 7 or less is valid only when the value of the CnBRP register is other than 00<sub>H</sub>.

**Caution** The values in *Table 14-27* do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

### 14.15.2 Representative examples of baud rate settings

Table 14-28 and Table 14-29 show representative examples of baud rate settings.

**Table 14-28 Representative examples of baud rate settings**  
(f<sub>CANMOD</sub> = 8 MHz) (1/2)

Set baud rate value (unit: kbps)	Division ratio of CnBRP register	CnBRP register set value	Valid bit rate setting (unit: kbps)					CnBTR register setting value		Sampling point (unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	TSEG1 [3:0]	TSEG2 [2:0]	
1000	1	00000000	8	1	1	3	3	0011	010	62.5
1000	1	00000000	8	1	3	2	2	0100	001	75.0
1000	1	00000000	8	1	5	1	1	0101	000	87.5
500	1	00000000	16	1	1	7	7	0111	110	56.3
500	1	00000000	16	1	3	6	6	1000	101	62.5
500	1	00000000	16	1	5	5	5	1001	100	68.8
500	1	00000000	16	1	7	4	4	1010	011	75.0
500	1	00000000	16	1	9	3	3	1011	010	81.3
500	1	00000000	16	1	11	2	2	1100	001	87.5
500	1	00000000	16	1	13	1	1	1101	000	93.8
500	2	00000001	8	1	1	3	3	0011	010	62.5
500	2	00000001	8	1	3	2	2	0100	001	75.0
500	2	00000001	8	1	5	1	1	0101	000	87.5
250	2	00000001	16	1	1	7	7	0111	110	56.3
250	2	00000001	16	1	3	6	6	1000	101	62.5
250	2	00000001	16	1	5	5	5	1001	100	68.8
250	2	00000001	16	1	7	4	4	1010	011	75.0
250	2	00000001	16	1	9	3	3	1011	010	81.3
250	2	00000001	16	1	11	2	2	1100	001	87.5
250	2	00000001	16	1	13	1	1	1101	000	93.8
250	4	00000011	8	1	3	2	2	0100	001	75.0
250	4	00000011	8	1	5	1	1	0101	000	87.5
125	4	00000011	16	1	1	7	7	0111	110	56.3
125	4	00000011	16	1	3	6	6	1000	101	62.5
125	4	00000011	16	1	5	5	5	1001	100	68.8
125	4	00000011	16	1	7	4	4	1010	011	75.0
125	4	00000011	16	1	9	3	3	1011	010	81.3
125	4	00000011	16	1	11	2	2	1100	001	87.5
125	4	00000011	16	1	13	1	1	1101	000	93.8
125	8	00000111	8	1	3	2	2	0100	001	75.0
125	8	00000111	8	1	5	1	1	0101	000	87.5
100	4	00000011	20	1	7	6	6	1100	101	70.0
100	4	00000011	20	1	9	5	5	1101	100	75.0
100	5	00000100	16	1	7	4	4	1010	011	75.0

**Table 14-28 Representative examples of baud rate settings**  
**( $f_{CANMOD} = 8 \text{ MHz}$ ) (2/2)**

Set baud rate value (unit: kbps)	Division ratio of CnBRP register	CnBRP register set value	Valid bit rate setting (unit: kbps)					CnBTR register setting value		Sampling point (unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	TSEG1 [3:0]	TSEG2 [2:0]	
100	5	00000100	16	1	9	3	3	1011	010	81.3
100	8	00000111	10	1	3	3	3	0101	010	70.0
100	8	00000111	10	1	5	2	2	0110	001	80.0
100	10	00001001	8	1	3	2	2	0100	001	75.0
100	10	00001001	8	1	5	1	1	0101	000	87.5
83.3	4	00000011	24	1	7	8	8	1110	111	66.7
83.3	4	00000011	24	1	9	7	7	1111	110	70.8
83.3	6	00000101	16	1	5	5	5	1001	100	68.8
83.3	6	00000101	16	1	7	4	4	1010	011	75.0
83.3	6	00000101	16	1	9	3	3	1011	010	81.3
83.3	6	00000101	16	1	11	2	2	1100	001	87.5
83.3	8	00000111	12	1	5	3	3	0111	010	75.0
83.3	8	00000111	12	1	7	2	2	1000	001	83.3
83.3	12	00001011	8	1	3	2	2	0100	001	75.0
83.3	12	00001011	8	1	5	1	1	0101	000	87.5
33.3	10	00001001	24	1	7	8	8	1110	111	66.7
33.3	10	00001001	24	1	9	7	7	1111	110	70.8
33.3	12	00001011	20	1	7	6	6	1100	101	70.0
33.3	12	00001011	20	1	9	5	5	1101	100	75.0
33.3	15	00001110	16	1	7	4	4	1010	011	75.0
33.3	15	00001110	16	1	9	3	3	1011	010	81.3
33.3	16	00001111	15	1	6	4	4	1001	011	73.3
33.3	16	00001111	15	1	8	3	3	1010	010	80.0
33.3	20	00010011	12	1	5	3	3	0111	010	75.0
33.3	20	00010011	12	1	7	2	2	1000	001	83.3
33.3	24	00010111	10	1	3	3	3	0101	010	70.0
33.3	24	00010111	10	1	5	2	2	0110	001	80.0
33.3	30	00011101	8	1	3	2	2	0100	001	75.0
33.3	30	00011101	8	1	5	1	1	0101	000	87.5

**Caution** The values in *Table 14-28* do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

**Table 14-29 Representative examples of baud rate settings**  
(f<sub>CANMOD</sub> = 16 MHz) (1/2)

Set baud rate value (unit: kbps)	Division ratio of CnBRP register	CnBRP register set value	Valid bit rate setting (unit: kbps)					CnBTR register setting value		Sampling point (unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	TSEG1 [3:0]	TSEG2 [2:0]	
1000	1	00000000	16	1	1	7	7	0111	110	56.3
1000	1	00000000	16	1	3	6	6	1000	101	62.5
1000	1	00000000	16	1	5	5	5	1001	100	68.8
1000	1	00000000	16	1	7	4	4	1010	011	75.0
1000	1	00000000	16	1	9	3	3	1011	010	81.3
1000	1	00000000	16	1	11	2	2	1100	001	87.5
1000	1	00000000	16	1	13	1	1	1101	000	93.8
1000	2	00000001	8	1	3	2	2	0100	001	75.0
1000	2	00000001	8	1	5	1	1	0101	000	87.5
500	2	00000001	16	1	1	7	7	0111	110	56.3
500	2	00000001	16	1	3	6	6	1000	101	62.5
500	2	00000001	16	1	5	5	5	1001	100	68.8
500	2	00000001	16	1	7	4	4	1010	011	75.0
500	2	00000001	16	1	9	3	3	1011	010	81.3
500	2	00000001	16	1	11	2	2	1100	001	87.5
500	2	00000001	16	1	13	1	1	1101	000	93.8
500	4	00000011	8	1	3	2	2	0100	001	75.0
500	4	00000011	8	1	5	1	1	0101	000	87.5
250	4	00000011	16	1	3	6	6	1000	101	62.5
250	4	00000011	16	1	5	5	5	1001	100	68.8
250	4	00000011	16	1	7	4	4	1010	011	75.0
250	4	00000011	16	1	9	3	3	1011	010	81.3
250	4	00000011	16	1	11	2	2	1100	001	87.5
250	8	00000111	8	1	3	2	2	0100	001	75.0
250	8	00000111	8	1	5	1	1	0101	000	87.5
125	8	00000111	16	1	3	6	6	1000	101	62.5
125	8	00000111	16	1	7	4	4	1010	011	75.0
125	8	00000111	16	1	9	3	3	1011	010	81.3
125	8	00000111	16	1	11	2	2	1100	001	87.5
125	16	00001111	8	1	3	2	2	0100	001	75.0
125	16	00001111	8	1	5	1	1	0101	000	87.5
100	8	00000111	20	1	9	5	5	1101	100	75.0
100	8	00000111	20	1	11	4	4	1110	011	80.0
100	10	00001001	16	1	7	4	4	1010	011	75.0
100	10	00001001	16	1	9	3	3	1011	010	81.3
100	16	00001111	10	1	3	3	3	0101	010	70.0
100	16	00001111	10	1	5	2	2	0110	001	80.0
100	20	00010011	8	1	3	2	2	0100	001	75.0

**Table 14-29 Representative examples of baud rate settings**  
**( $f_{CANMOD} = 16 \text{ MHz}$ ) (2/2)**

Set baud rate value (unit: kbps)	Division ratio of CnBRP register	CnBRP register set value	Valid bit rate setting (unit: kbps)					CnBTR register setting value		Sampling point (unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT 1	PHASE SEGMENT 2	TSEG1 [3:0]	TSEG2 [2:0]	
83.3	8	00000111	24	1	7	8	8	1110	111	66.7
83.3	8	00000111	24	1	9	7	7	1111	110	70.8
83.3	12	00001011	16	1	7	4	4	1010	011	75.0
83.3	12	00001011	16	1	9	3	3	1011	010	81.3
83.3	12	00001011	16	1	11	2	2	1100	001	87.5
83.3	16	00001111	12	1	5	3	3	0111	010	75.0
83.3	16	00001111	12	1	7	2	2	1000	001	83.3
83.3	24	00010111	8	1	3	2	2	0100	001	75.0
83.3	24	00010111	8	1	5	1	1	0101	000	87.5
33.3	30	00011101	24	1	7	8	8	1110	111	66.7
33.3	30	00011101	24	1	9	7	7	1111	110	70.8
33.3	24	00010111	20	1	9	5	5	1101	100	75.0
33.3	24	00010111	20	1	11	4	4	1110	011	80.0
33.3	30	00011101	16	1	7	4	4	1010	011	75.0
33.3	30	00011101	16	1	9	3	3	1011	010	81.3
33.3	32	00011111	15	1	8	3	3	1010	010	80.0
33.3	32	00011111	15	1	10	2	2	1011	001	86.7
33.3	37	00100100	13	1	6	3	3	1000	010	76.9
33.3	37	00100100	13	1	8	2	2	1001	001	84.6
33.3	40	00100111	12	1	5	3	3	0111	010	75.0
33.3	40	00100111	12	1	7	2	2	1000	001	83.3
33.3	48	00101111	10	1	3	3	3	0101	010	70.0
33.3	48	00101111	10	1	5	2	2	0110	001	80.0
33.3	60	00111011	8	1	3	2	2	0100	001	75.0
33.3	60	00111011	8	1	5	1	1	0101	000	87.5

**Caution** The values in *Table 14-29* do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.



## 14.16 Operation of CAN Controller

The processing procedure for showing in this chapter is recommended processing procedure to operate CAN controller.

Develop the program referring to recommended processing procedure in this chapter.

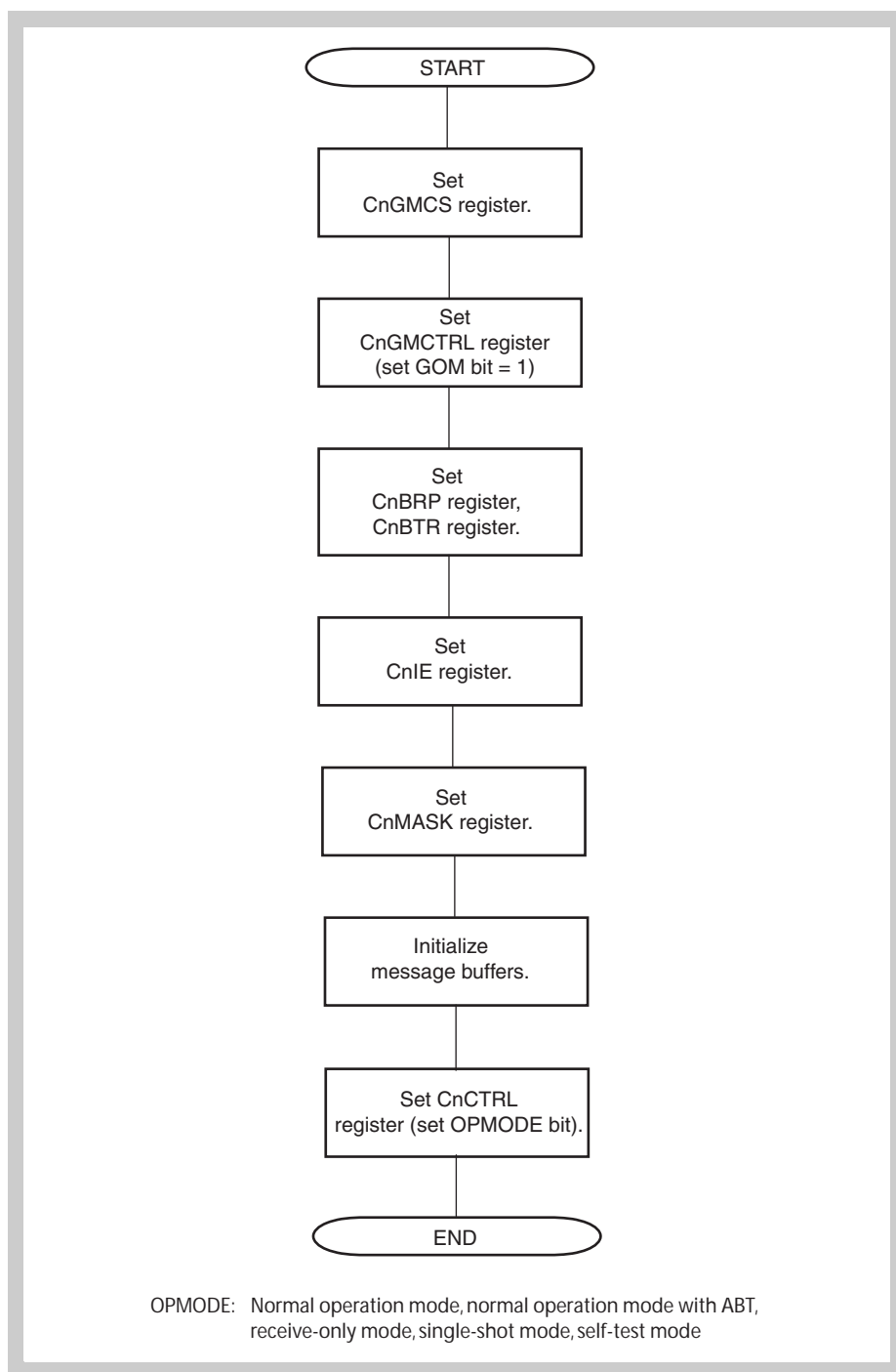


Figure 14-35 Initialization

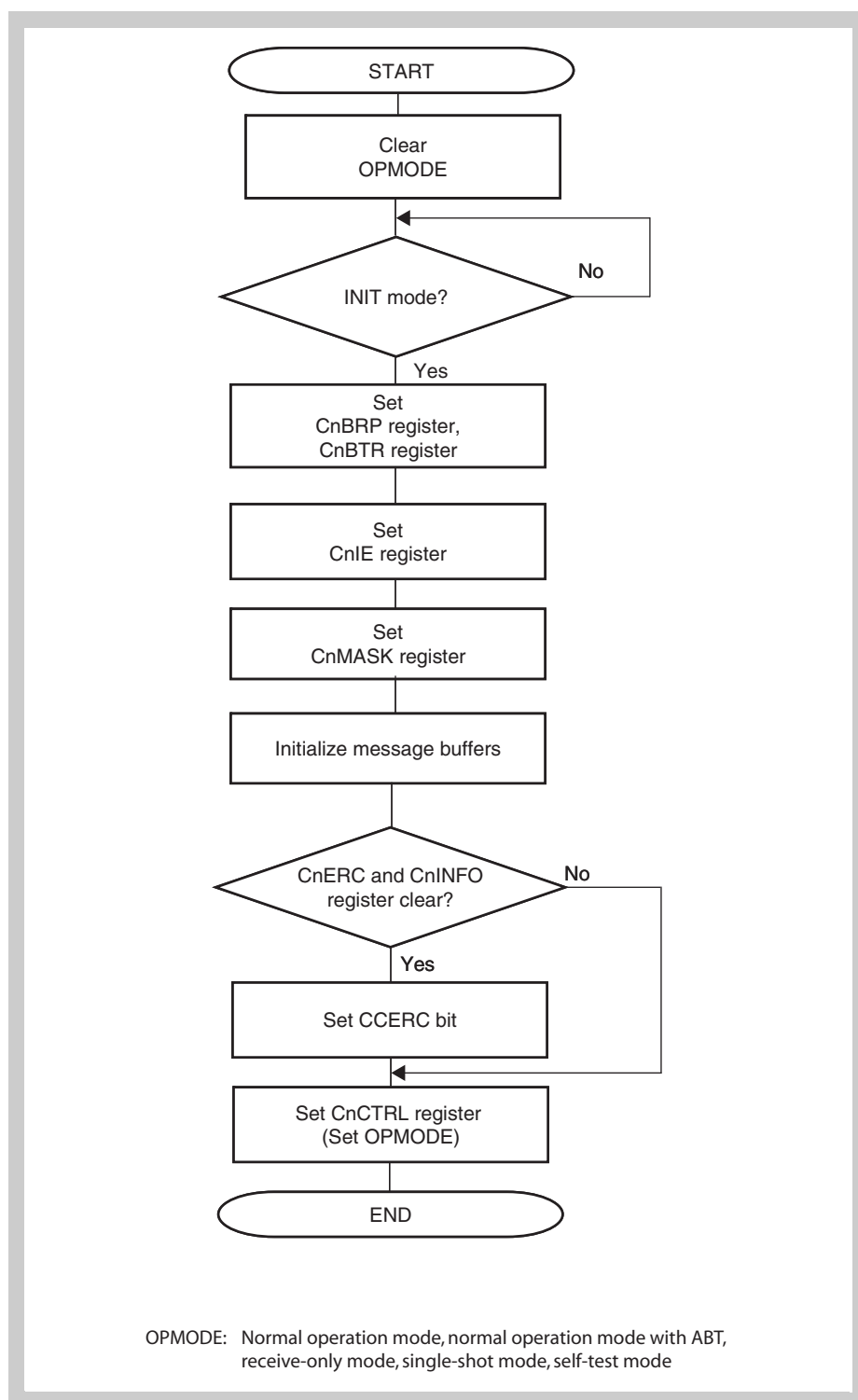
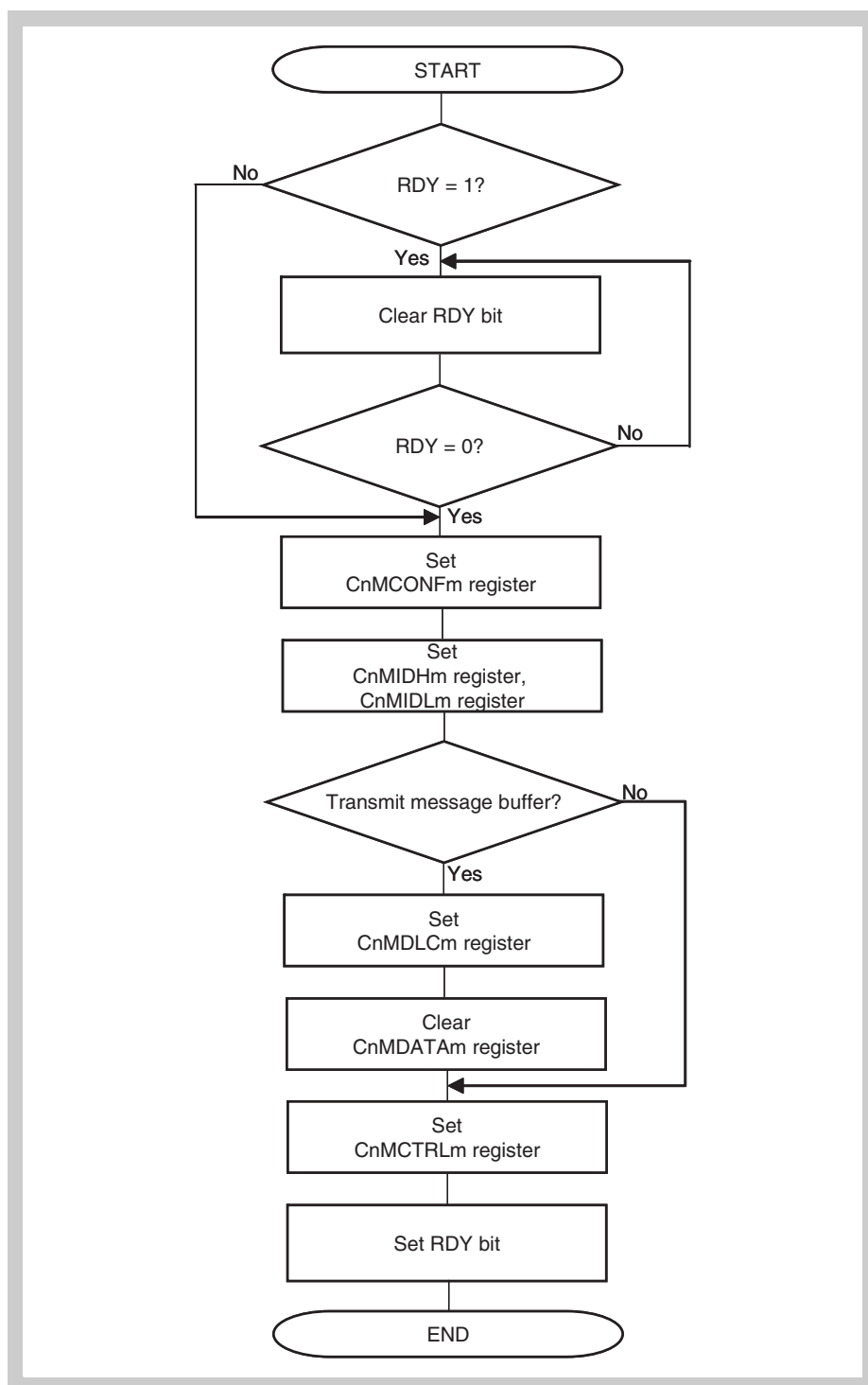


Figure 14-36 Re-initialization

**Caution** After setting the CAN module to the initialization mode, avoid setting the module to another operation mode immediately after. If it is necessary to immediately set the module to another operation mode, be sure to access registers other than the CnCTRL and CnGMCTRL registers (e.g., set a message buffer).



**Figure 14-37** Message buffer initialization

- Caution**
1. Before a message buffer is initialized, the RDY bit must be cleared.
  2. Make the following settings for message buffers not used by the application.
    - Clear the RDY, TRQ, and DN bits of the CnMCTRLm register to 0.
    - Clear the MA0 bit of the CnMCONFm register to 0.

Figure 14-38 shows the processing for a receive message buffer (MT[2:0] bits of CnMCONFm register = 001<sub>B</sub> to 101<sub>B</sub>).

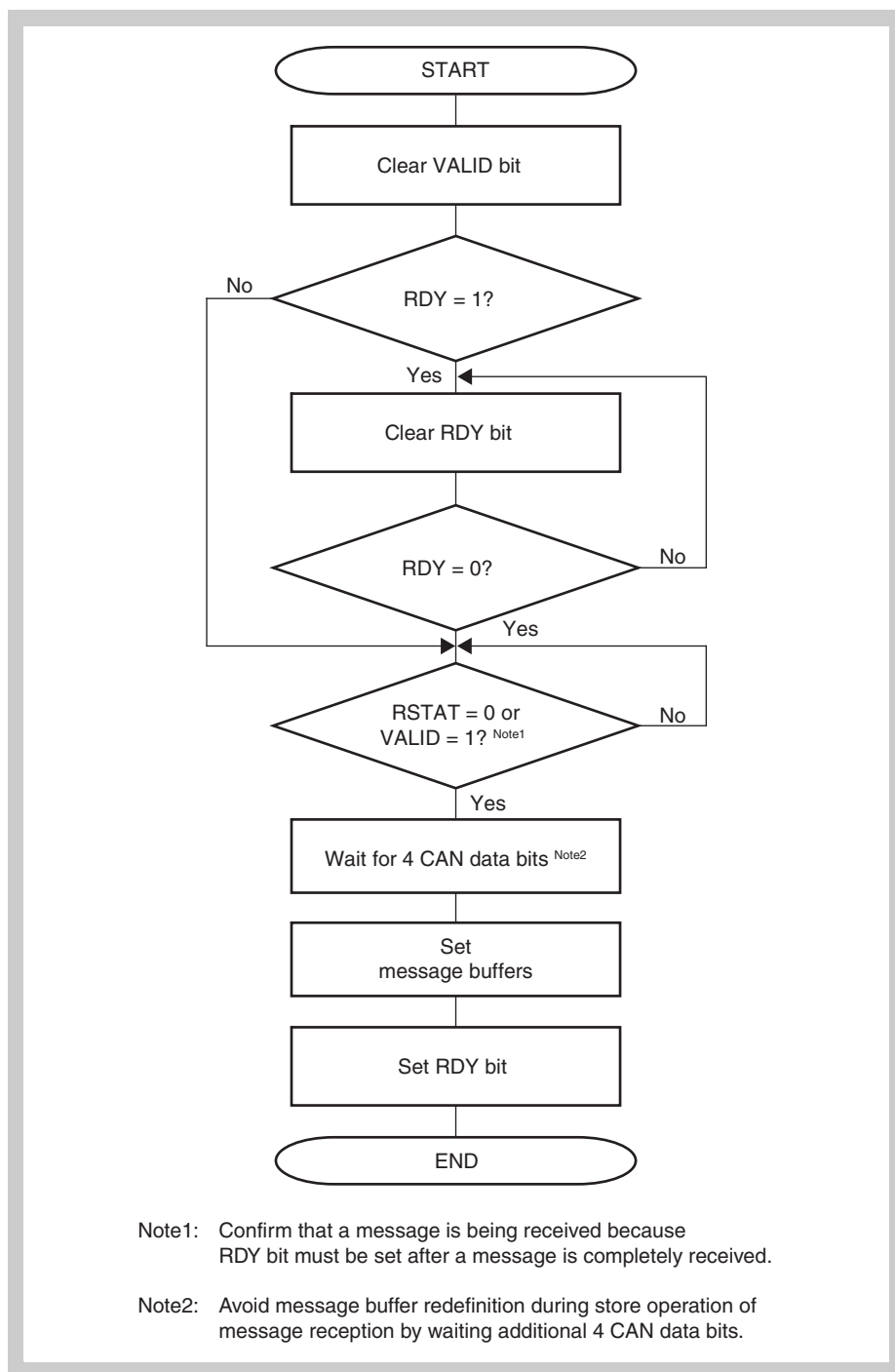


Figure 14-38 Message buffer redefinition

Figure 14-39 shows the processing for a transmit message buffer during transmission (MT[2:0] bits of CnMCONFm register = 000<sub>B</sub>).

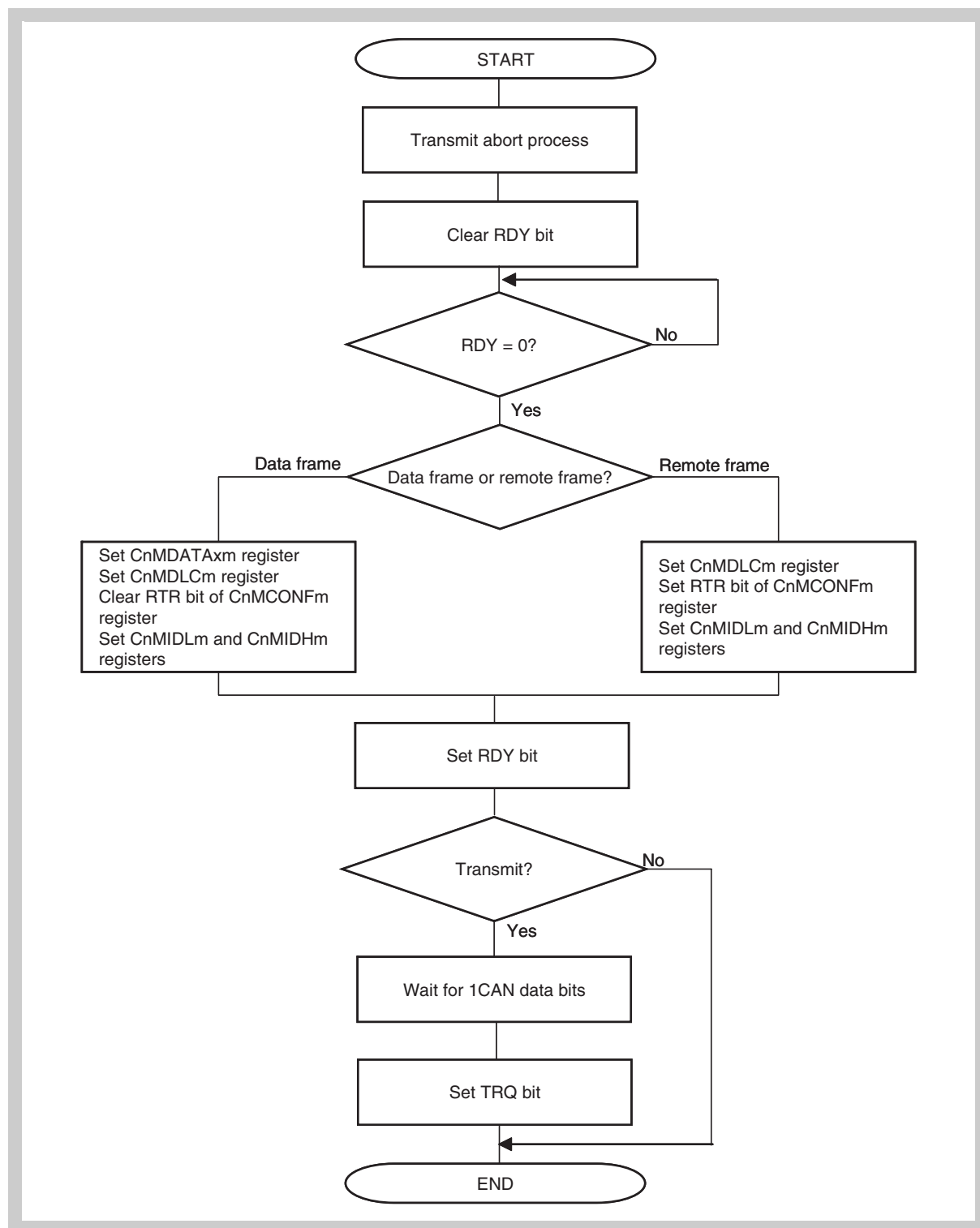


Figure 14-39 Message buffer redefinition during transmission

Figure 14-40 shows the processing for a transmit message buffer (MT[2:0] bits of CnMCONFm register = 000<sub>B</sub>).

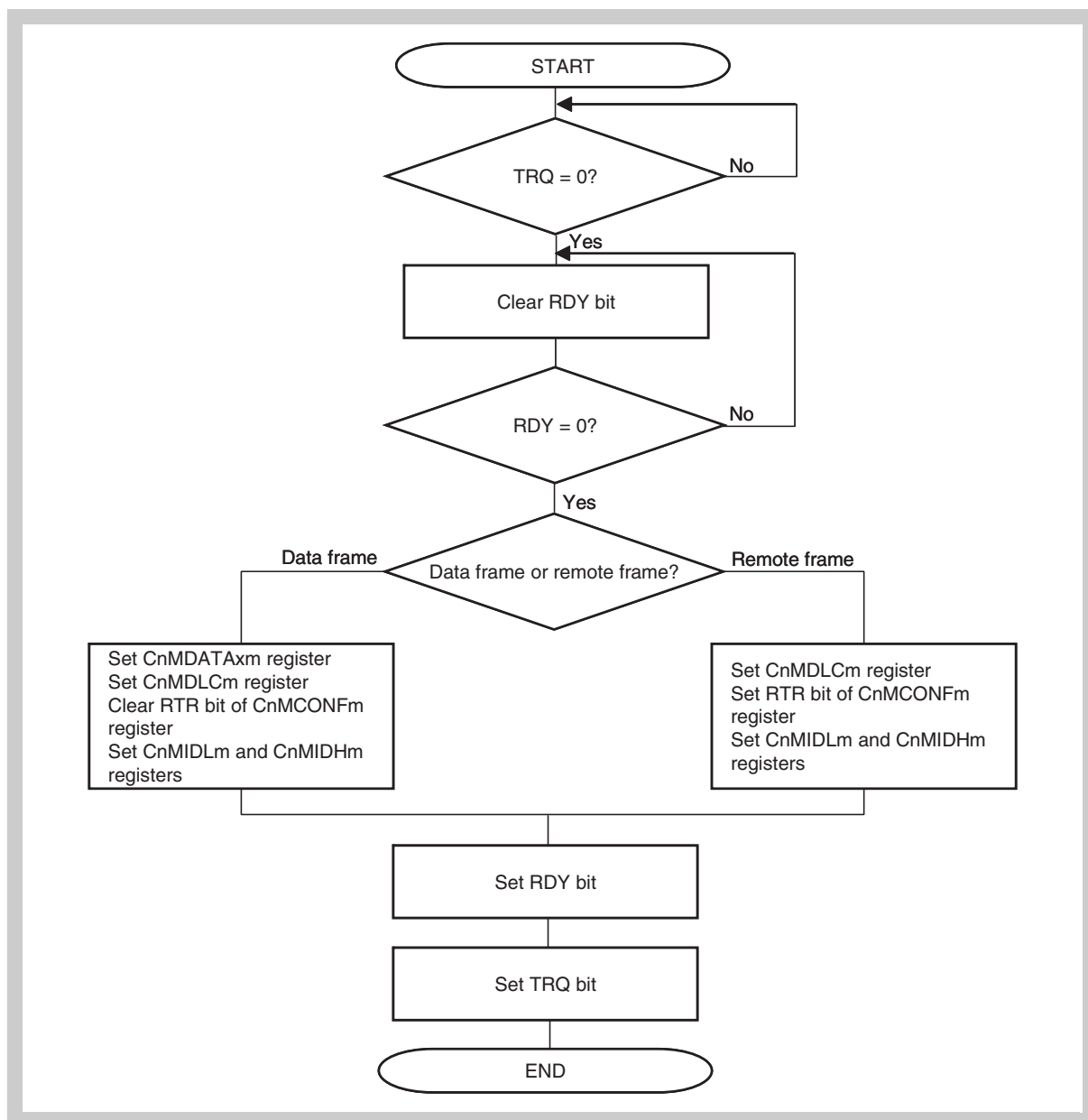
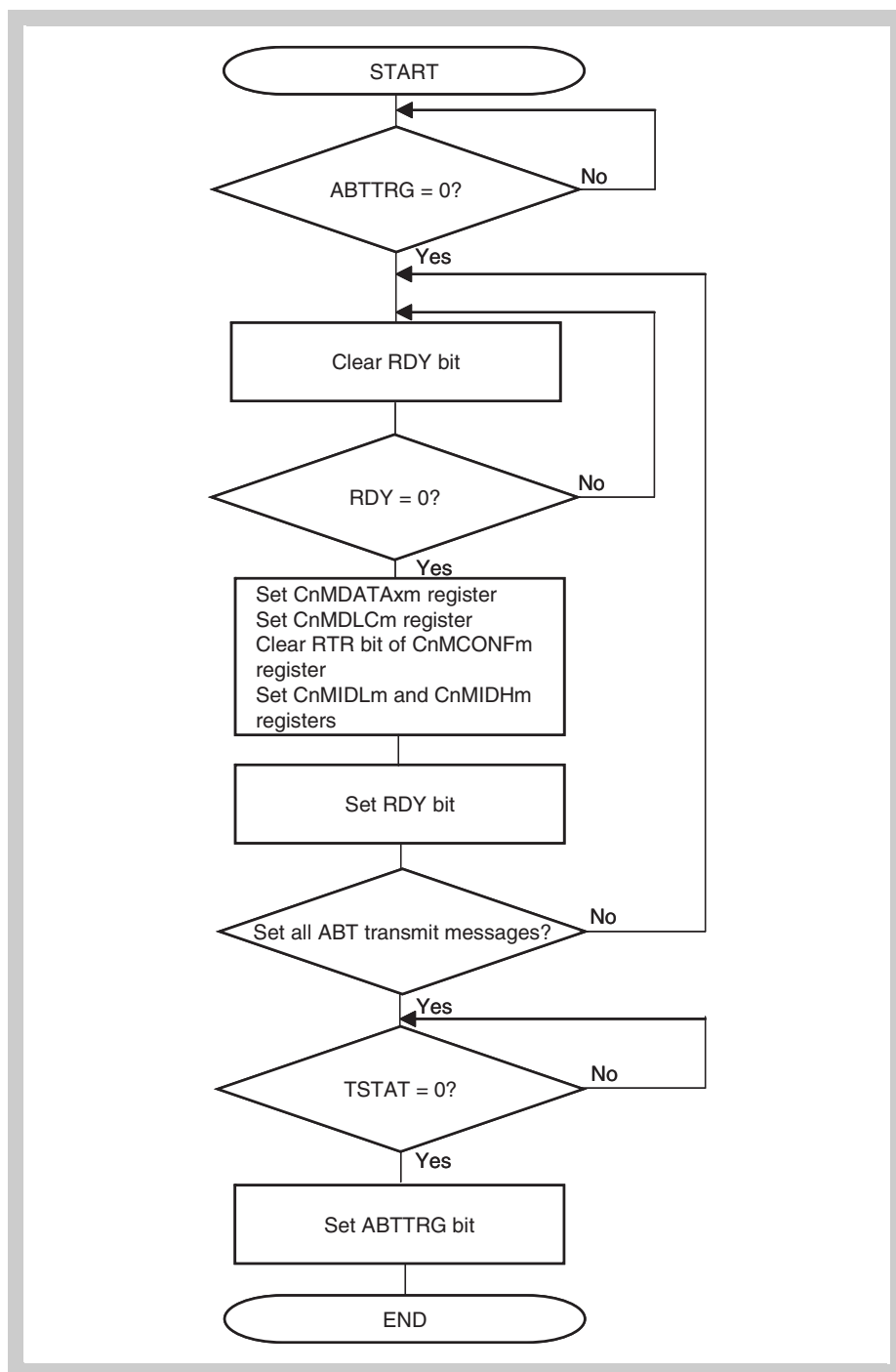


Figure 14-40 Message transmit processing

- Caution**
1. The TRQ bit should be set after the RDY bit is set.
  2. The RDY bit and TRQ bit should not be set at the same time.

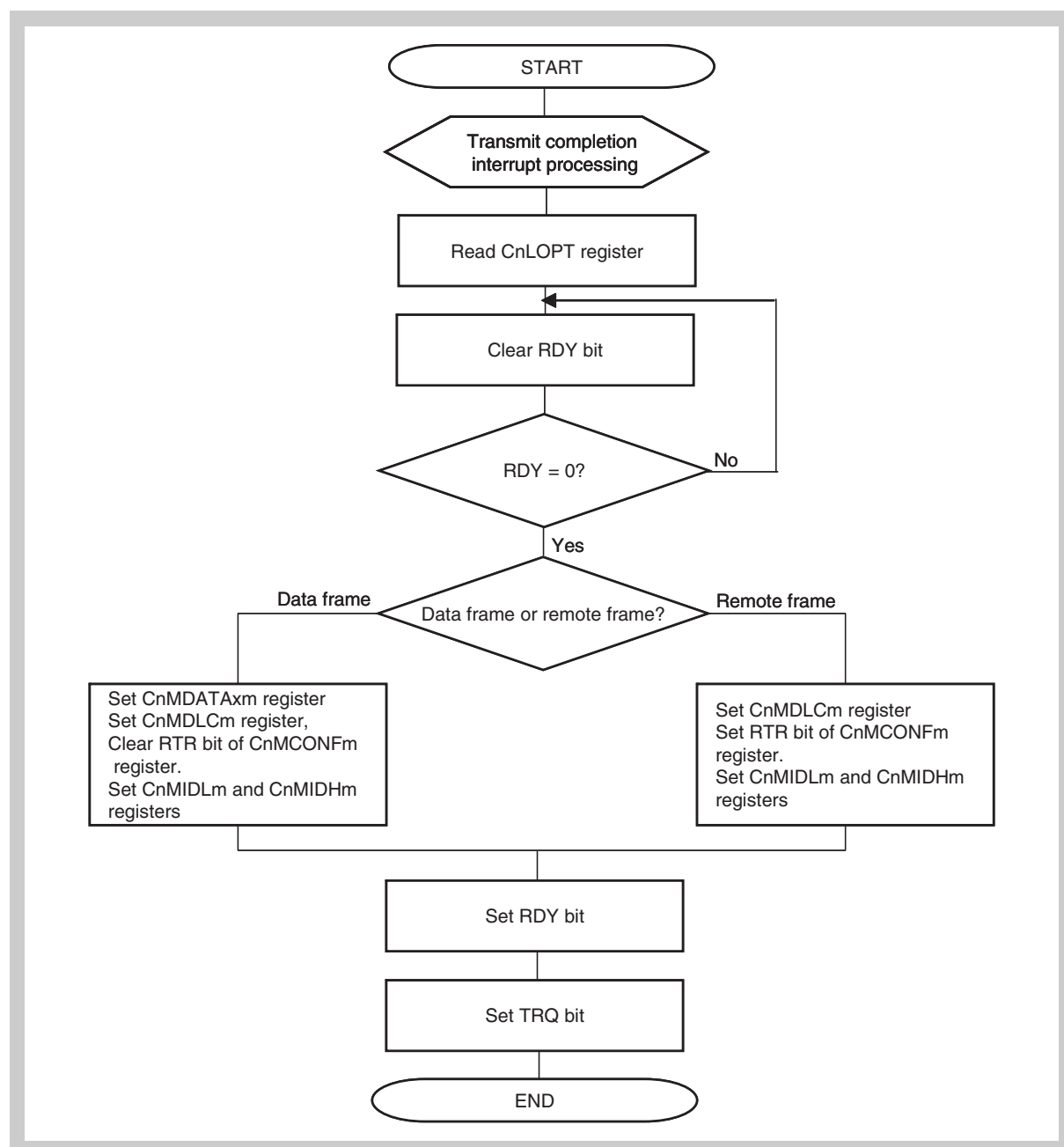
Figure 14-41 shows the processing for a transmit message buffer (MT[2:0] bits of CnMCONFm register = 000<sub>B</sub>)



**Figure 14-41** ABT message transmit processing

**Note** This processing (normal operation mode with ABT) can only be applied to message buffers 0 to 7. For message buffers other than the ABT message buffers, see Figure 14-40 on page 482.

**Caution** The ABTTRG bit should be set to 1 after the TSTAT bit is cleared to 0. Checking the TSTAT bit and setting the ABTTRG bit to 1 must be processed consecutively.



**Figure 14-42** Transmission via interrupt (using CnLOPT register)

- Caution**
1. The TRQ bit should be set after the RDY bit is set.
  2. The RDY bit and TRQ bit should not be set at the same time.

**Note** Also check the MBON flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as TX history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.  
It is recommended to cancel any sleep mode requests, before processing TX interrupts.



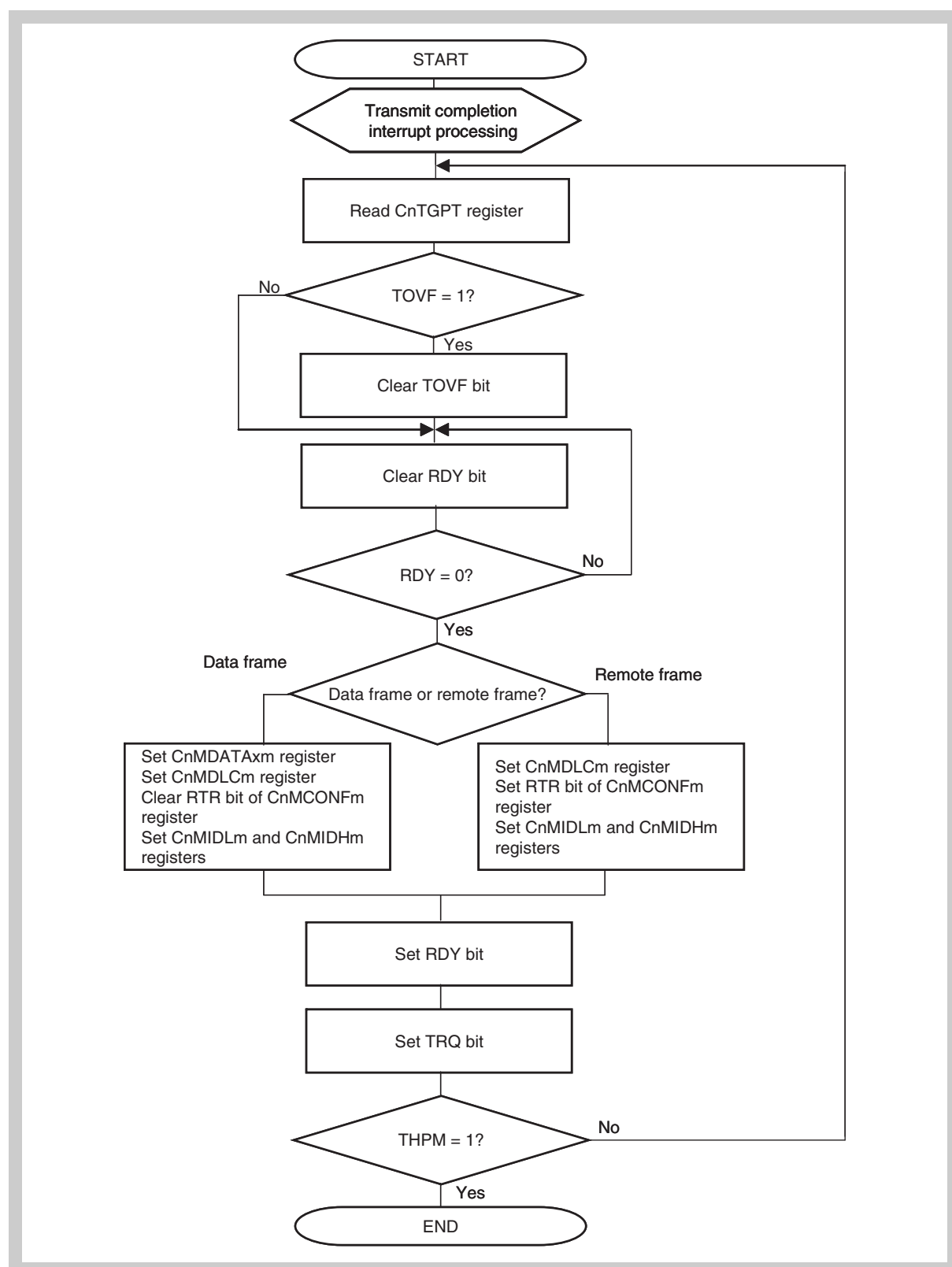


Figure 14-43 Transmission via interrupt (using CnTGPT register)

- Caution**
1. The TRQ bit should be set after the RDY bit is set.
  2. The RDY bit and TRQ bit should not be set at the same time.

- Note**
1. Also check the MBON flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as TX history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.  
It is recommended to cancel any sleep mode requests, before processing TX interrupts.
  2. If TOVF was set once, the transmit history list is inconsistent. Consider to scan all configured transmit buffers for completed transmissions.

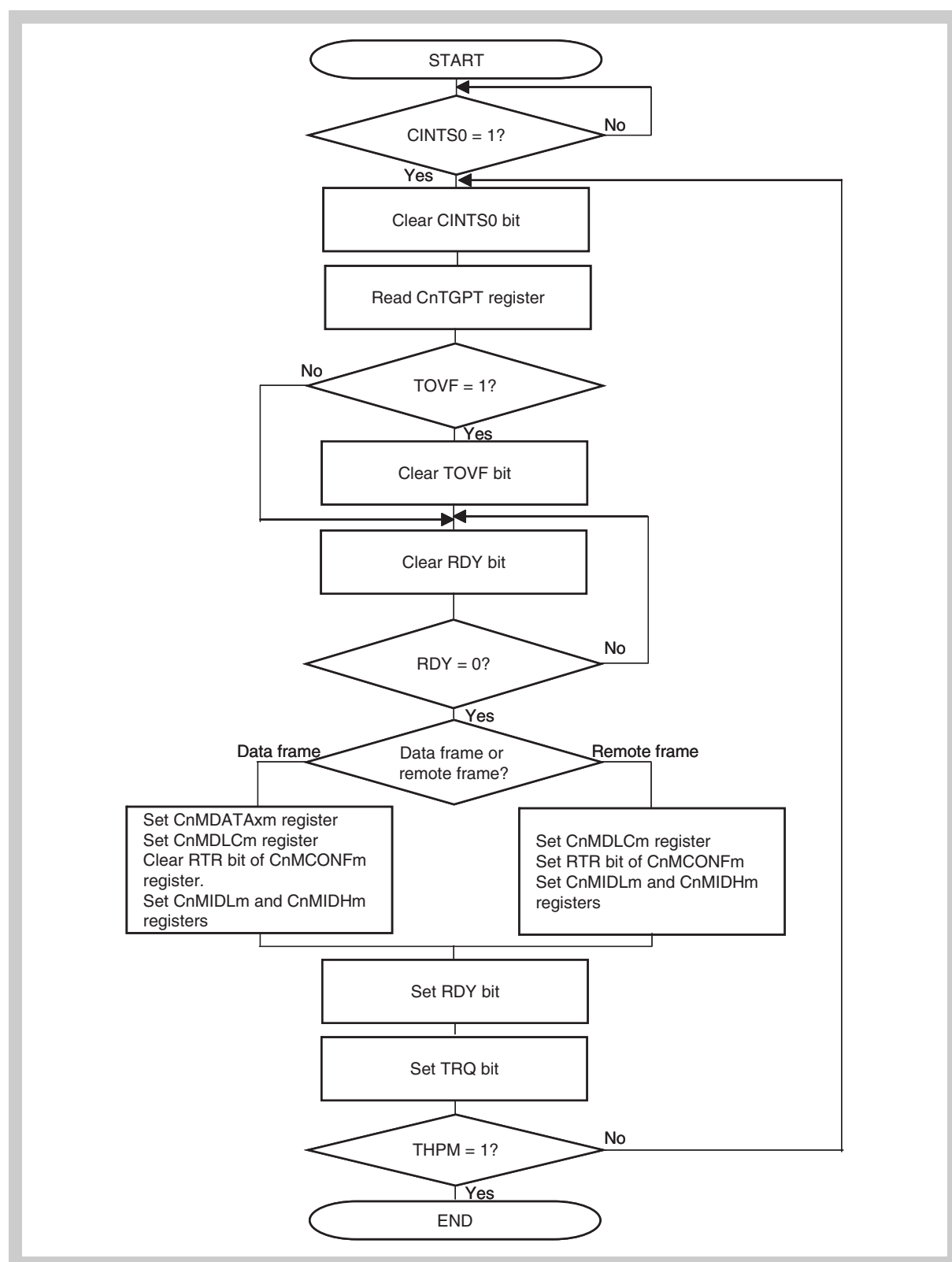
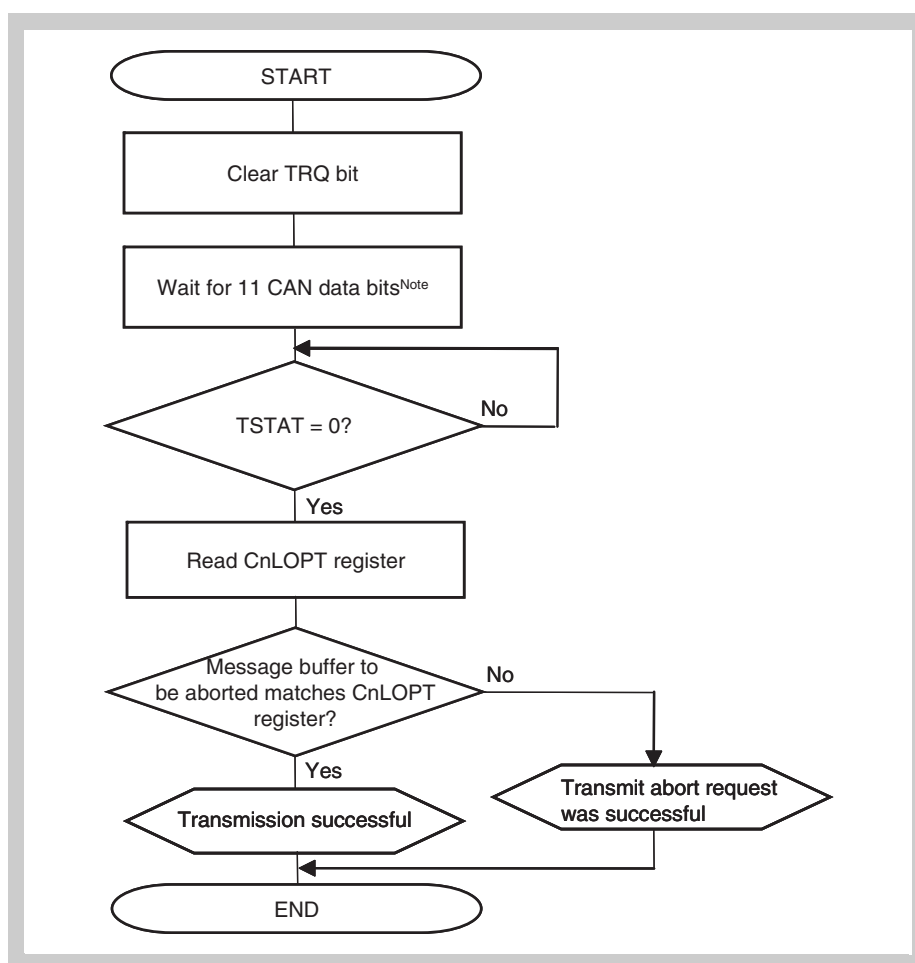


Figure 14-44 Transmission via software polling

- Caution**
1. The TRQ bit should be set after the RDY bit is set.
  2. The RDY bit and TRQ bit should not be set at the same time.

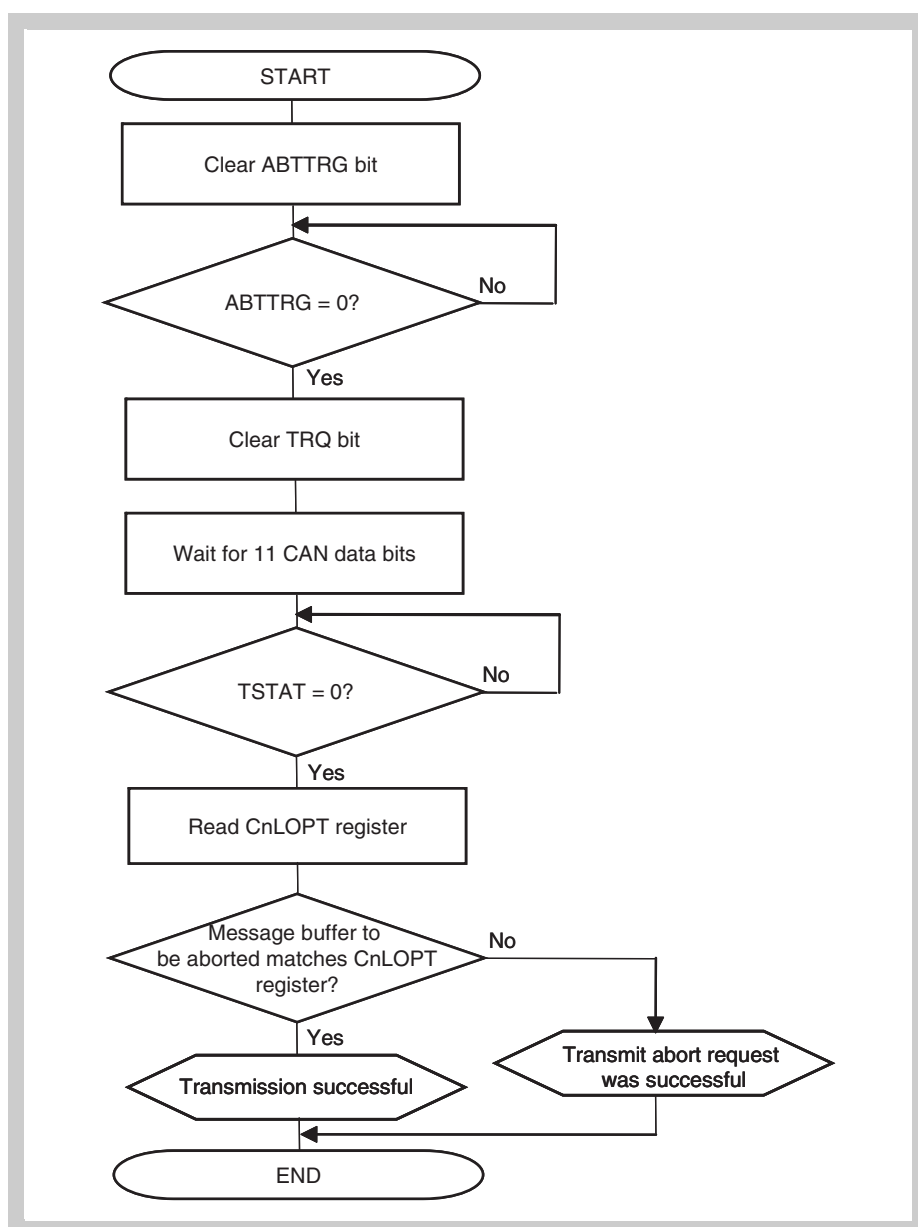
- 
- Note**
1. Also check the MBON flag at the beginning and at the end of the polling routine, in order to check the access to the message buffers as well as TX history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.
  2. If TOVF was set once, the transmit history list is inconsistent. Consider to scan all configured transmit buffers for completed transmissions.



**Figure 14-45** Transmission abort processing (except normal operation mode with ABT)

**Note** There is a possibility of starting the transmission without being aborted even if TRQ bit is cleared, because the transmission request to protocol layer might already been accepted between 11 bits, total of interframe space (3 bits) and suspend transmission (8 bits).

- Caution**
1. Clear the TRQ bit for aborting transmission request, not the RDY bit.
  2. Before making a sleep mode transition request, confirm that there is no transmission request left using this processing.
  3. The TSTAT bit can be periodically checked by a user application or can be checked after the transmit completion interrupt.
  4. Do not execute any new transmission request including in the other message buffers while transmission abort processing is in progress.
  5. If a new transmission request is executed for a message buffer within 500 clocks of the AFCAN macro, after transmission abort process and before IFS (Inter-Frame Space), that message might be transmitted in the next following transmission, although its ID priority was low.



**Figure 14-46** Transmission abort processing except for ABT transmission (normal operation mode with ABT)

- Caution**
1. Clear the TRQ bit for aborting transmission request, not the RDY bit.
  2. Before making a sleep mode transition request, confirm that there is no transmission request left using this processing.
  3. The TSTAT bit can be periodically checked by a user application or can be checked after the transmit completion interrupt.
  4. Do not execute any new transmission request including in the other message buffers while transmission abort processing is in progress.

Figure 14-47 shows the processing to skip resumption of transmitting a message that was stopped when transmission of an ABT message buffer was aborted.

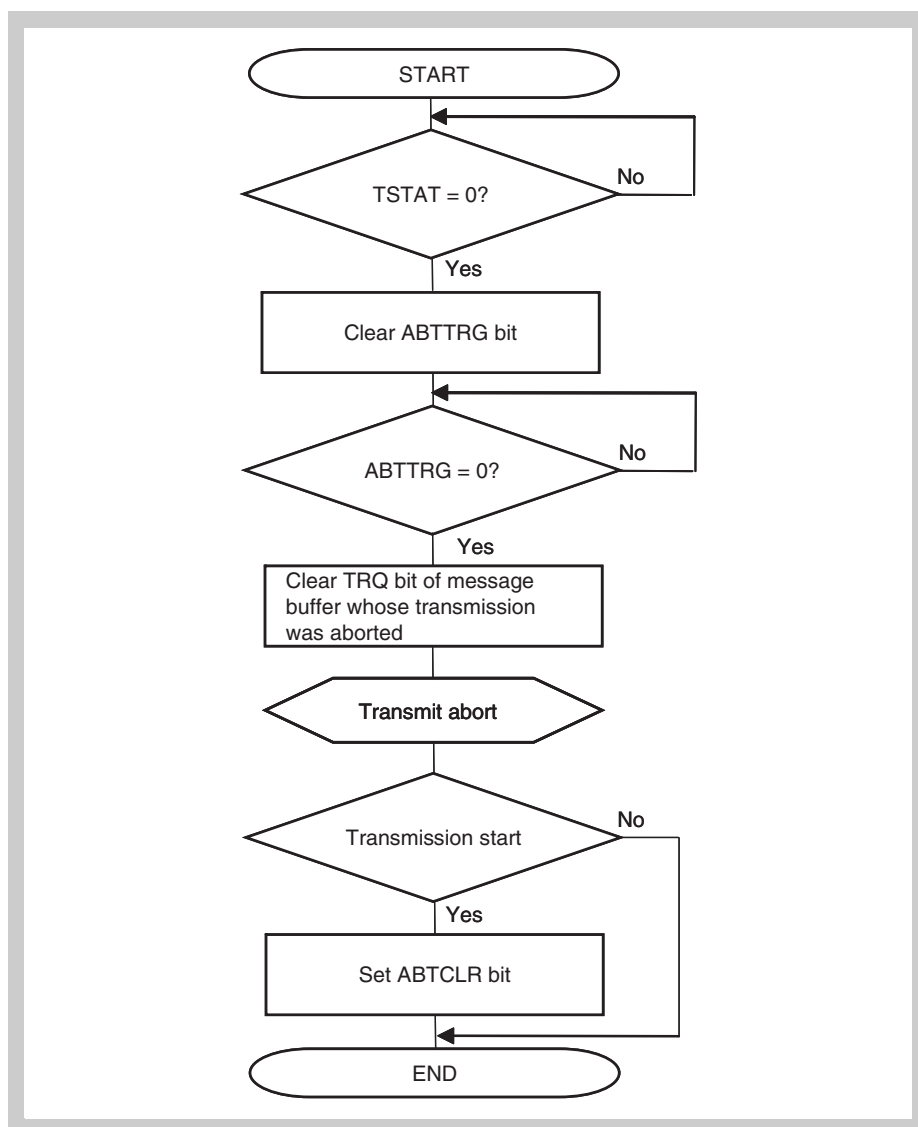
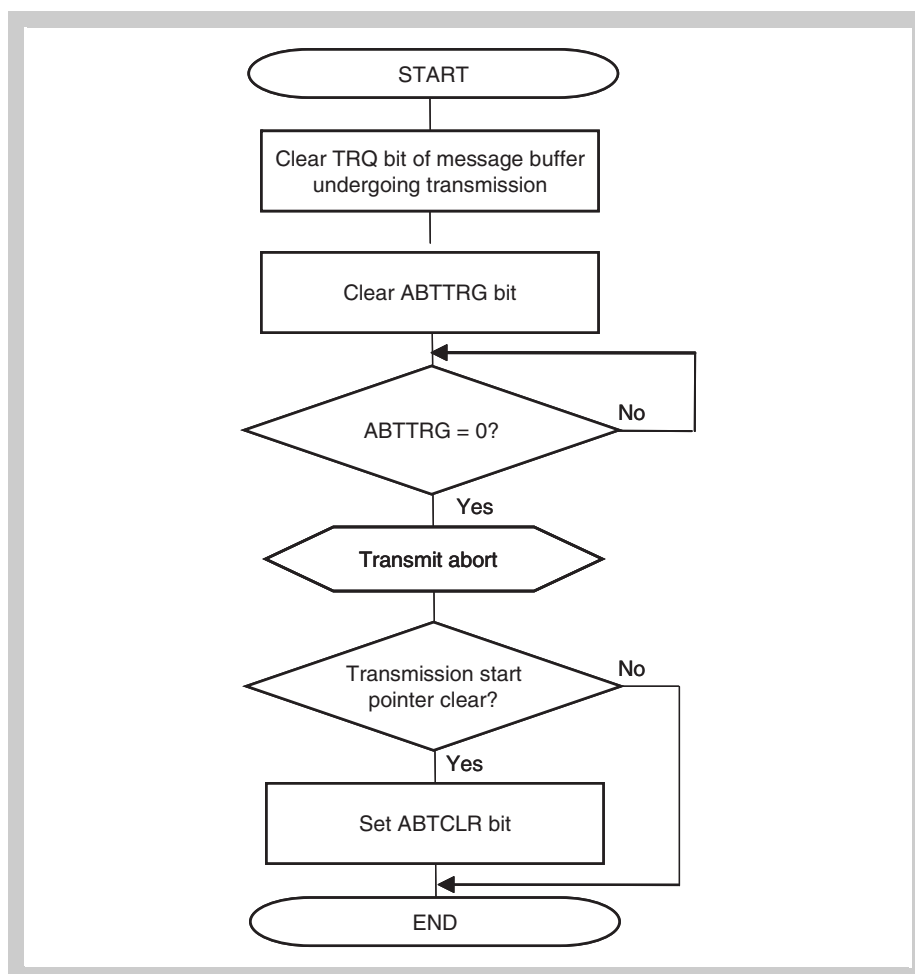


Figure 14-47 Transmission abort processing (normal operation mode with ABT)

- Caution**
1. Do not set any transmission requests while ABT transmission abort processing is in progress.
  2. Make a CAN sleep mode/CAN stop mode transition request after the ABTTRG bit is cleared (after ABT mode is aborted) following the procedure shown in *Figure 14-47* or *Figure 14-48*. When clearing a transmission request in an area other than the ABT area, follow the procedure shown in *Figure 14-45* on page 489.

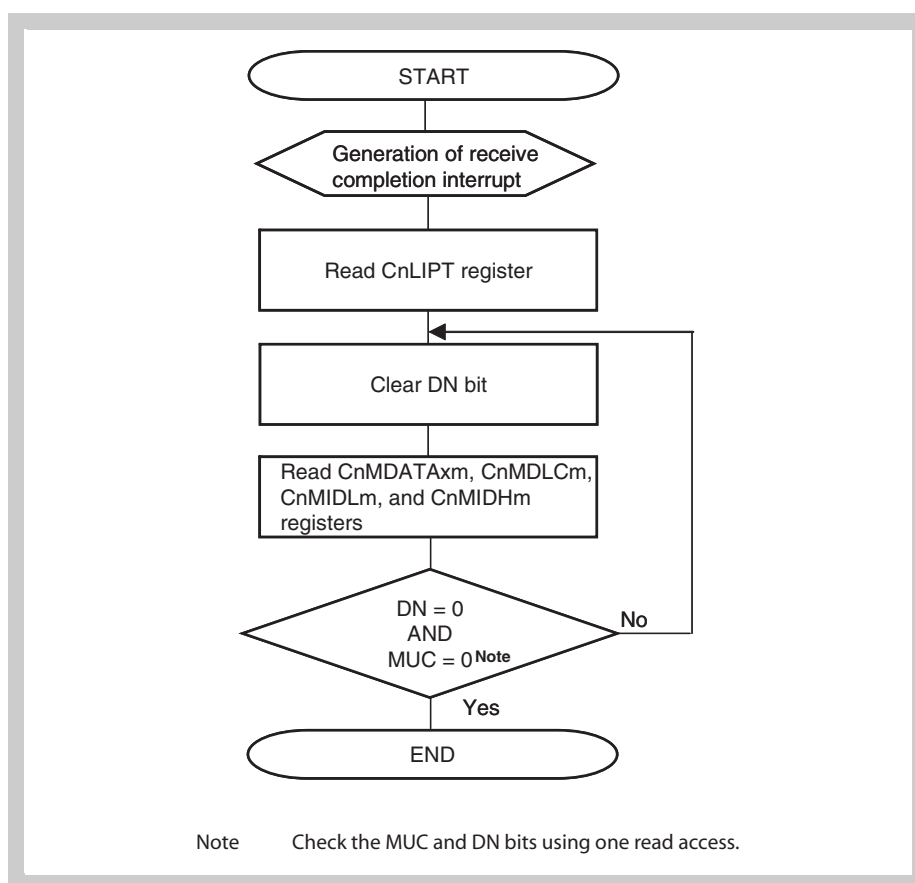
Figure 14-48 shows the processing to not skip resumption of transmitting a message that was stopped when transmission of an ABT message buffer was aborted.



**Figure 14-48** ABT transmission request abort processing (normal operation mode with ABT)

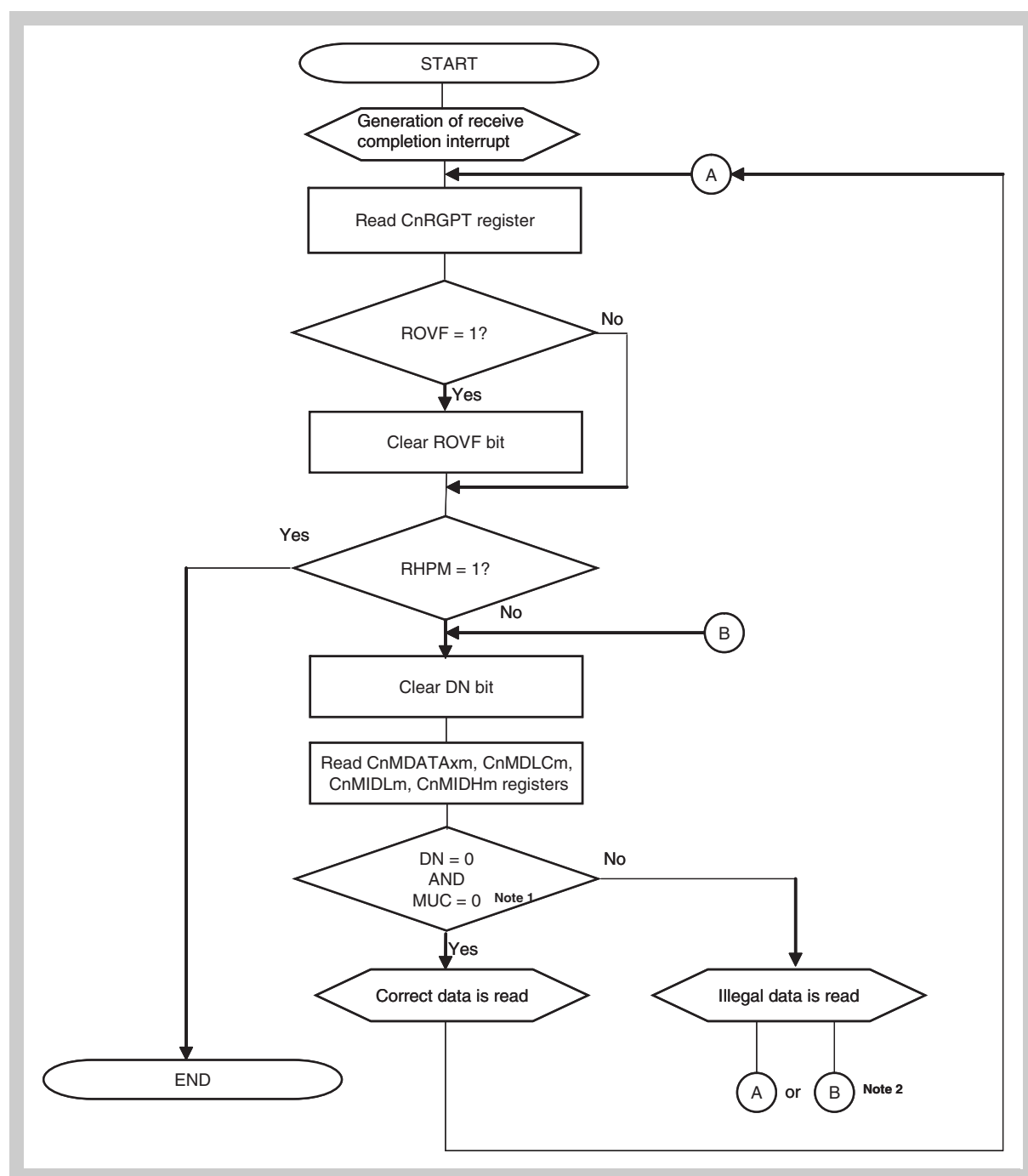
- Caution**
1. Do not set any transmission requests while ABT transmission abort processing is in progress.
  2. Make a CAN sleep mode/CAN stop mode request after the ABTTRG bit is cleared (after ABT mode is stopped) following the procedure shown in *Figure 14-47* or *Figure 14-48*. When clearing a transmission request in an area other than the ABT area, follow the procedure shown in *Figure 14-45* on page 489.





**Figure 14-49 Reception via interrupt (using CnLIPT register)**

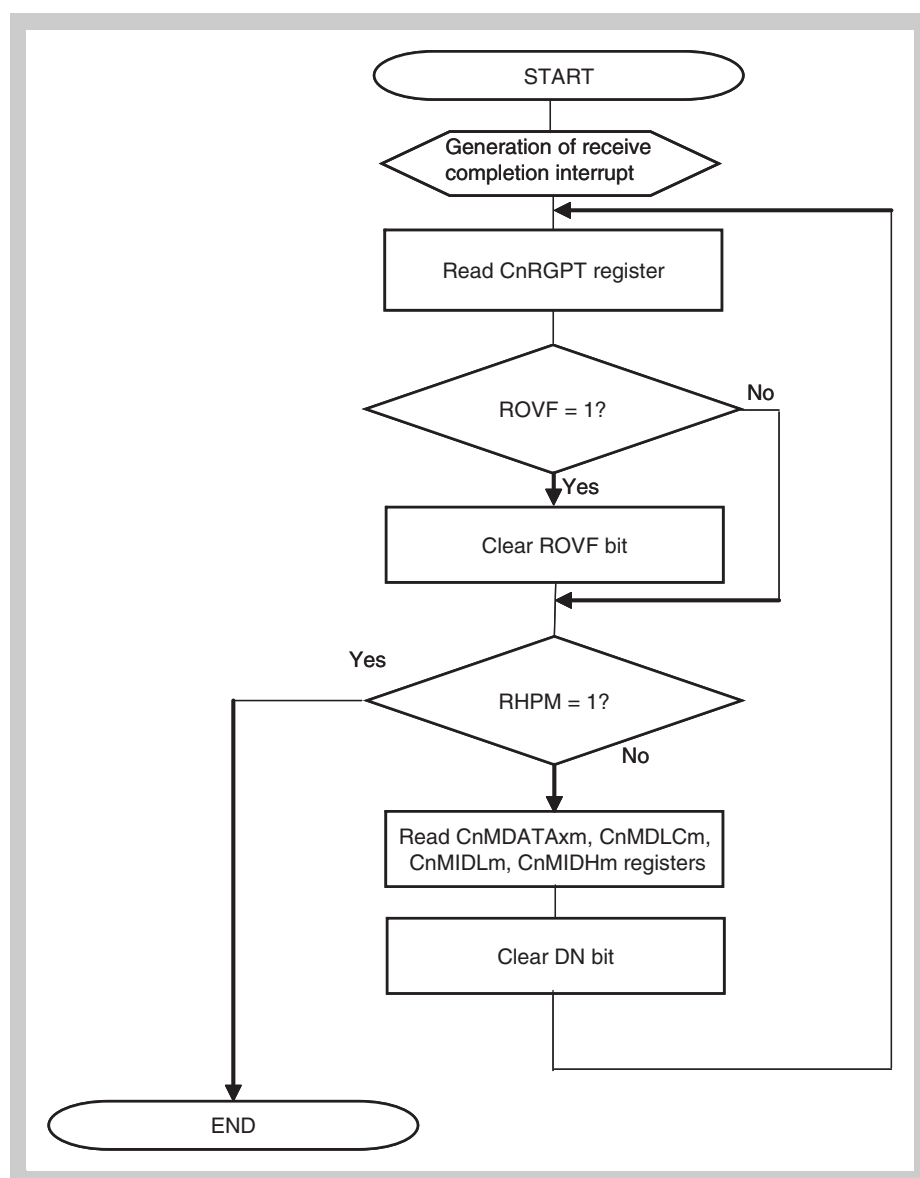
**Note** Also check the MBON flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as reception history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.  
It is recommended to cancel any sleep mode requests, before processing RX interrupts.



**Figure 14-50 Reception via interrupt (using CnRGPT register)**

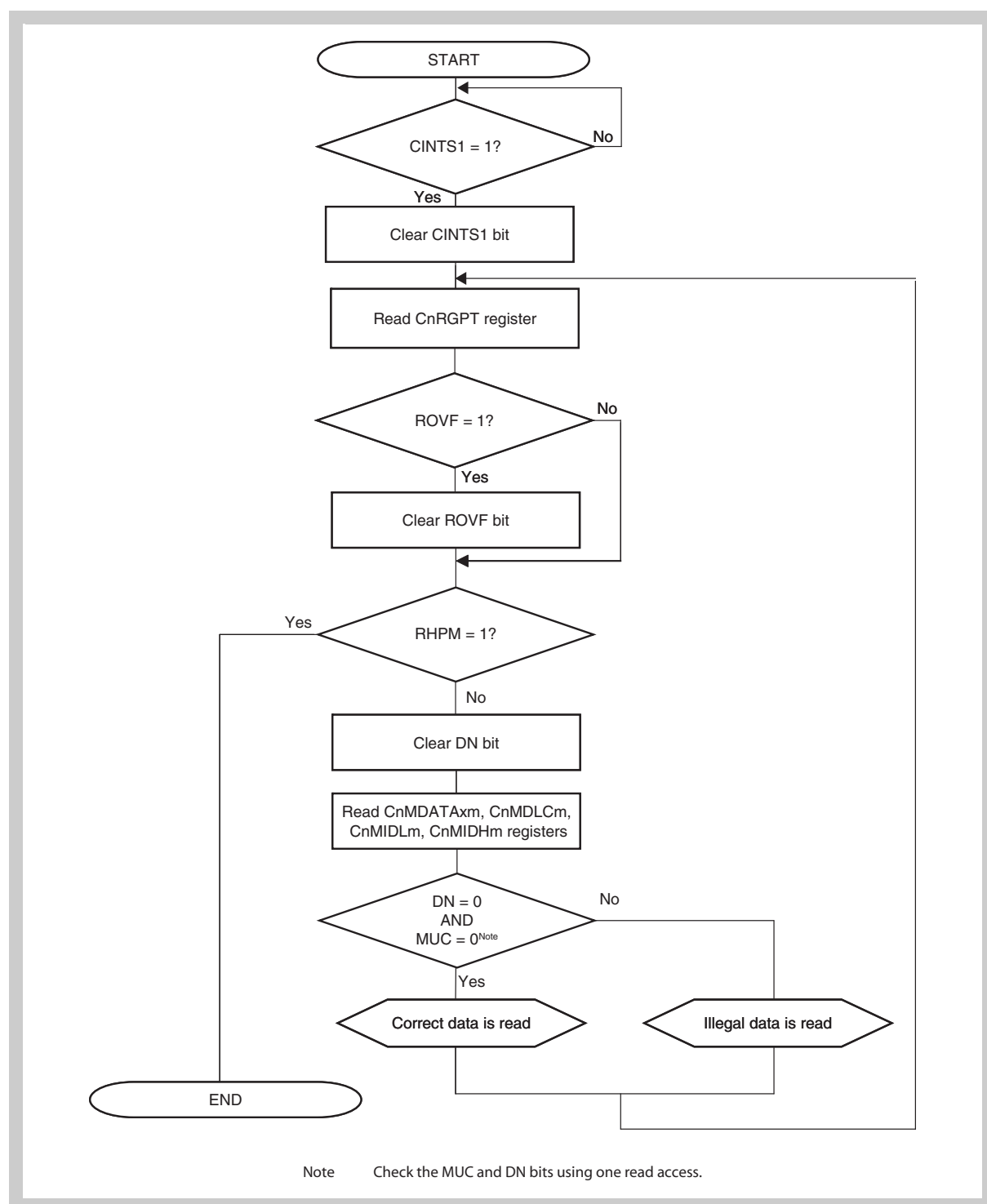
- Note**
1. Check the MUC and DN bits using one read access.
  2. Depending of the processing target of the application, two ways are possible:
    - Way A: The message is not processed within this pass, but with the next pass, depending on the timing this can happen latest with the next Receive Interrupt. Other messages will be processed earlier.
    - Way B: The message is processed within this pass, the loop waits on this message. Other messages will be processed later.

3. Also check the MBON flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as reception history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.  
It is recommended to cancel any sleep mode requests, before processing RX interrupts.
4. If ROVF was set once, the receive history list is inconsistent. Consider to scan all configured receive buffers for receptions.



**Figure 14-51 Reception via interrupt (using CnRGPT register), alternative way**

- Note**
1. Also check the MBON flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as reception history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.  
It is recommended to cancel any sleep mode requests, before processing RX interrupts.
  2. If ROVF was set once, the receive history list is inconsistent. Consider to scan all configured receive buffers for receptions.
  3. This flow will not provide most recently received data for the application. However, due to less effort on processing, it reduces interrupt load.
  4. The overwrite function (CnMCONFm.OWS=1) must not be used with this flow - data inconsistency could occur.
  5. It can be used alternatively to *Figure 14-50 on page 494*.



**Figure 14-52 Reception via software polling**

- Note**
1. Also check the MBON flag at the beginning and at the end of the polling routine, in order to check the access to the message buffers as well as reception history list registers, in case a pending sleep mode had been executed. If MBON is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after MBON is set again.
  2. If ROVF was set once, the receive history list is inconsistent. Consider to scan all configured receive buffers for receptions.

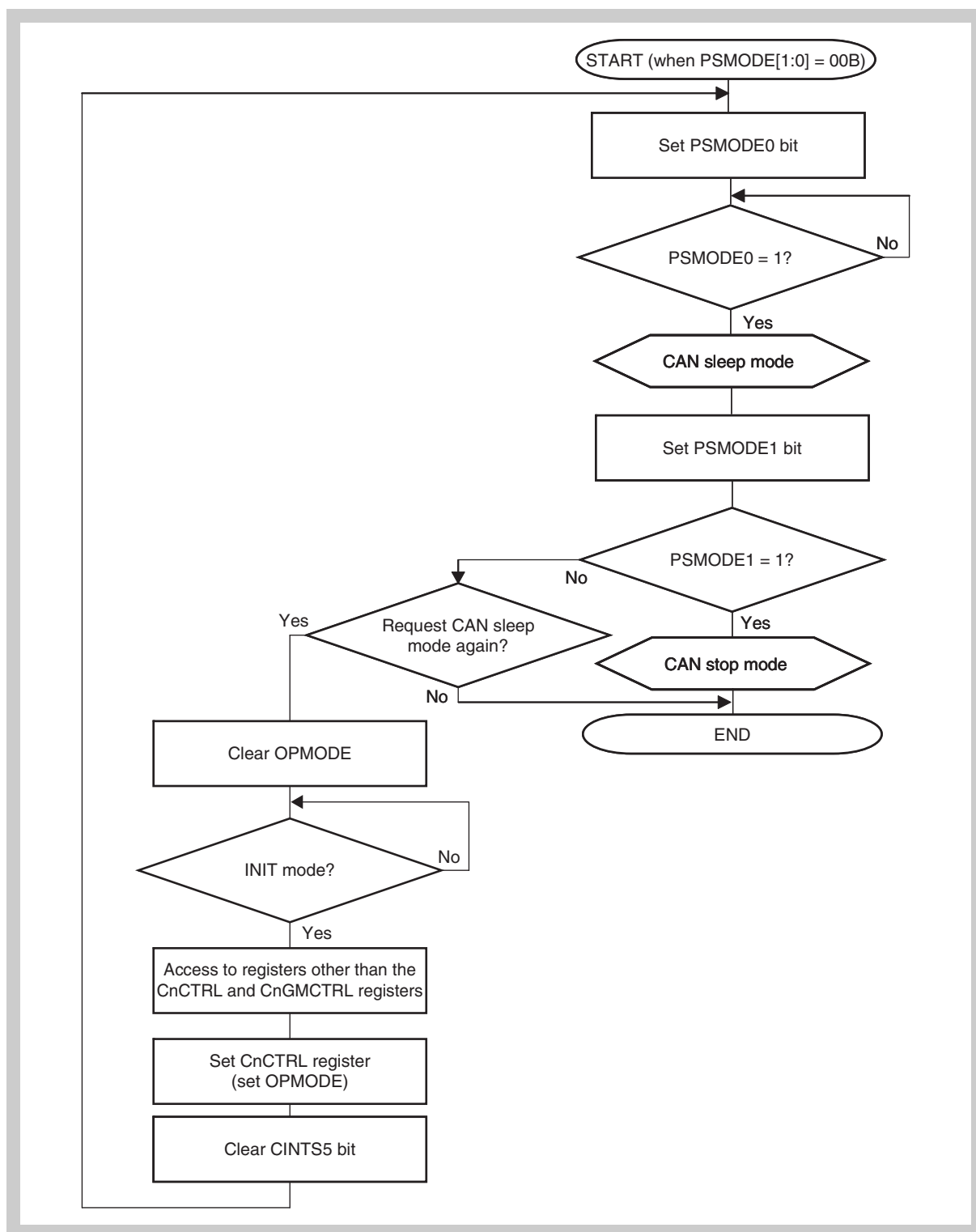
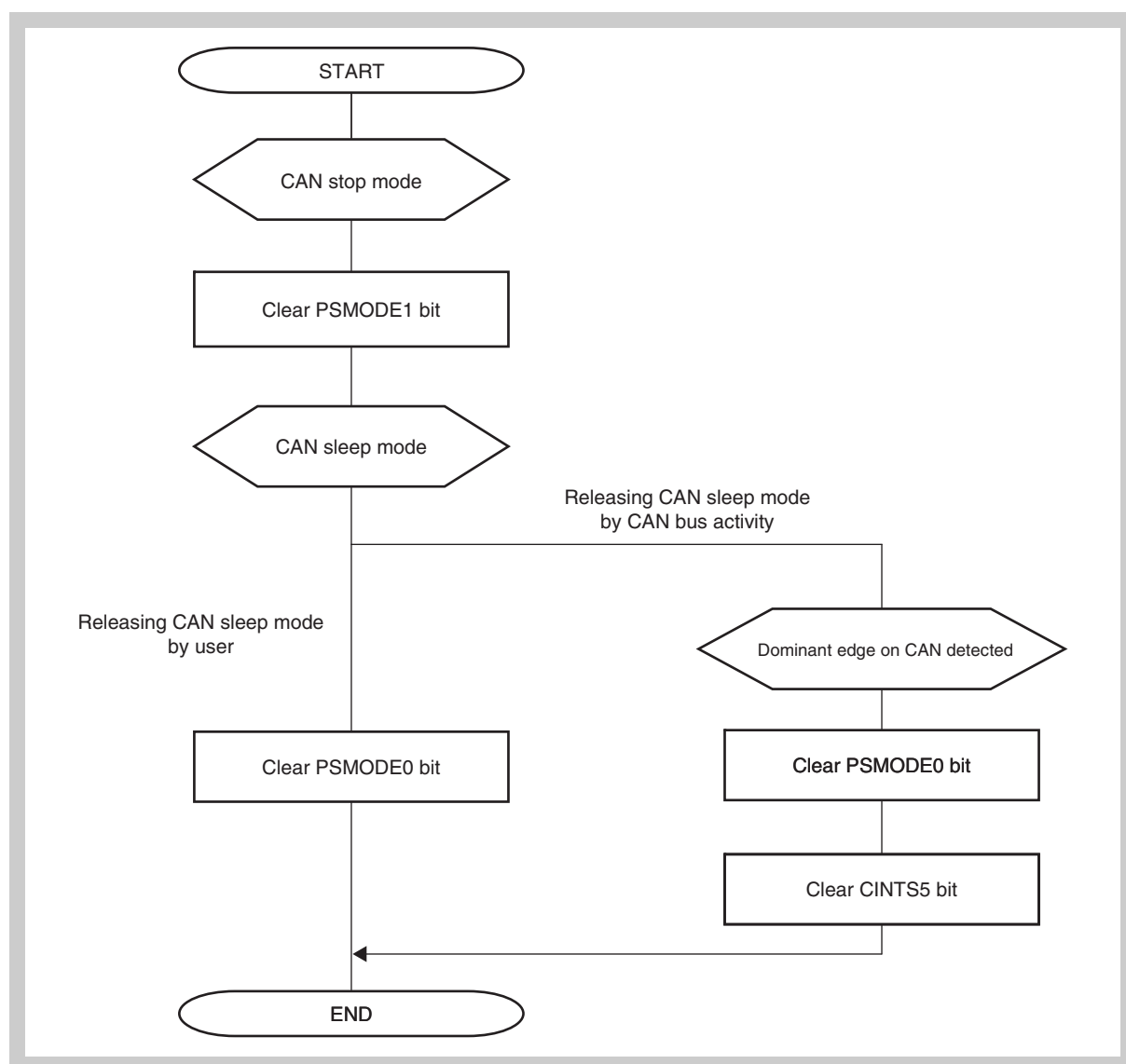
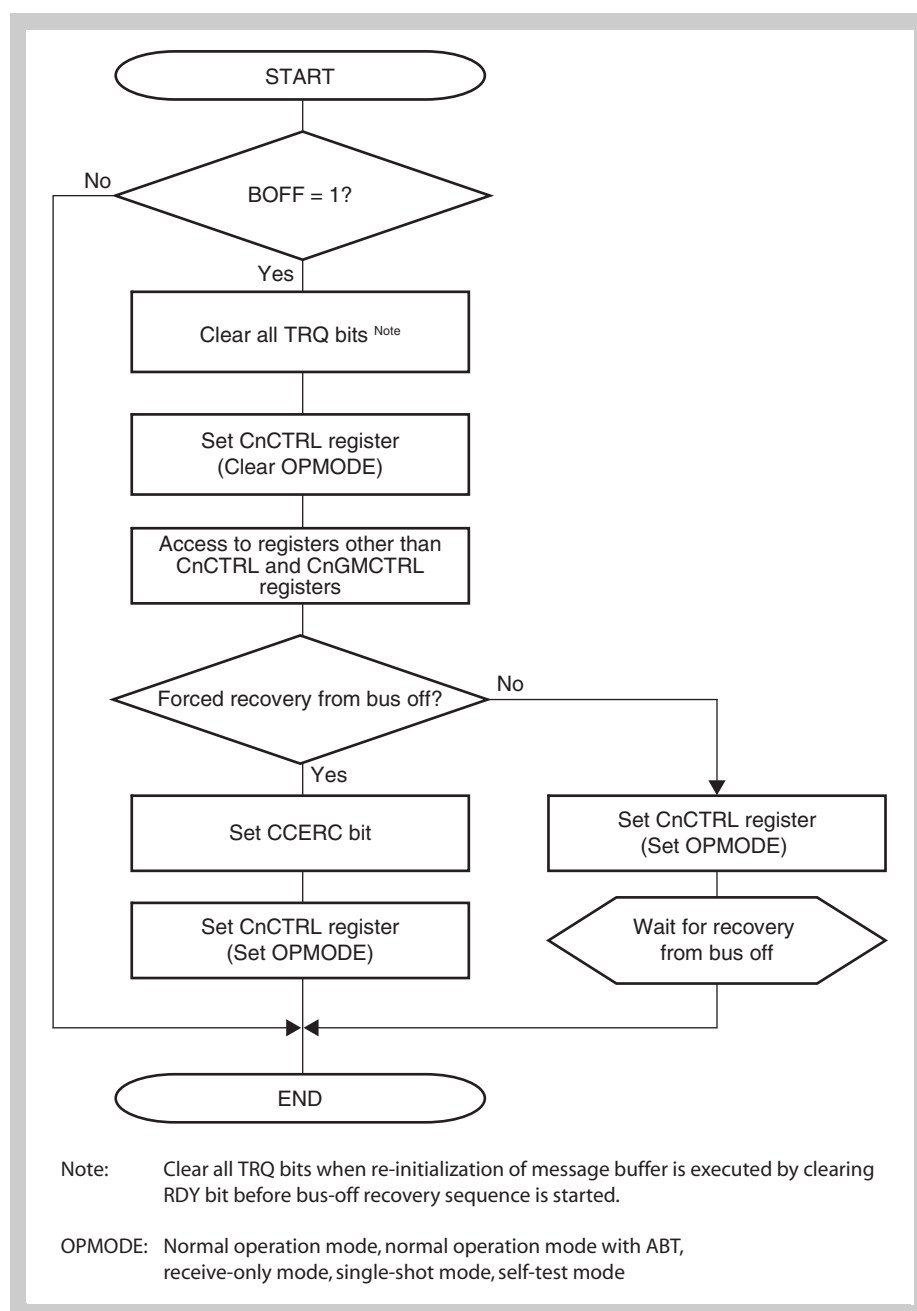


Figure 14-53 Setting CAN sleep mode/stop mode

**Caution** To abort transmission before making a request for the CAN sleep mode, perform processing according to Figure 14-45 on page 489 and Figure 14-47 on page 491.

**Figure 14-54** Clear CAN sleep/stop mode



**Figure 14-55** Bus-off recovery (except normal operation mode with ABT)

**Caution** When the transmission from the initialization mode to any operation modes is requested to execute bus-off recovery sequence again in the bus-off recovery sequence, reception error counter is cleared. Therefore it is necessary to detect 11 consecutive recessive-level bits 128 times on the bus again.



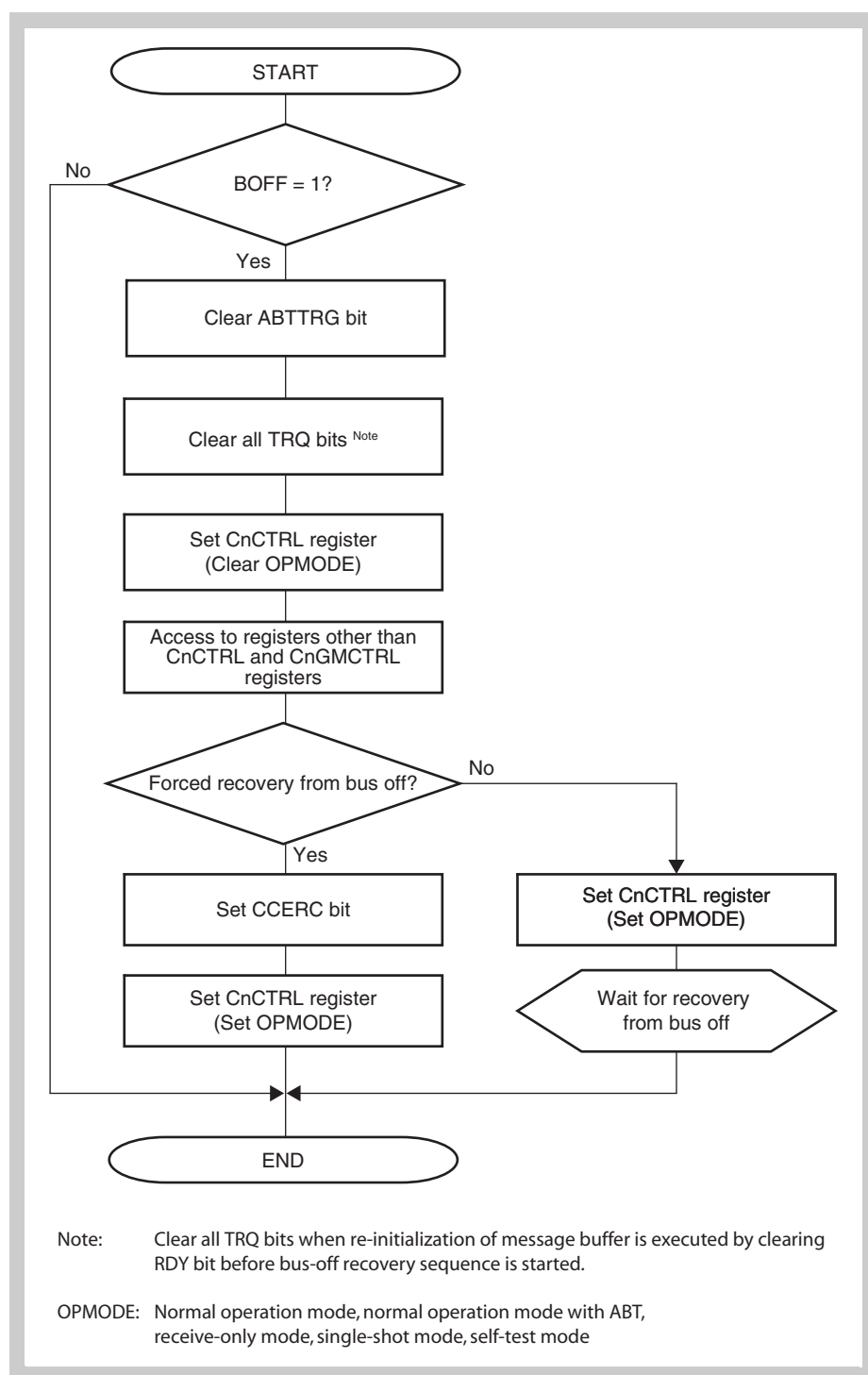


Figure 14-56 Bus-off recovery (Normal Operation Mode with ABT)

**Caution** When the transmission from the initialization mode to any operation modes is requested to execute bus-off recovery sequence again in the bus-off recovery sequence, reception error counter is cleared. Therefore it is necessary to detect 11 consecutive recessive-level bits 128 times on the bus again.

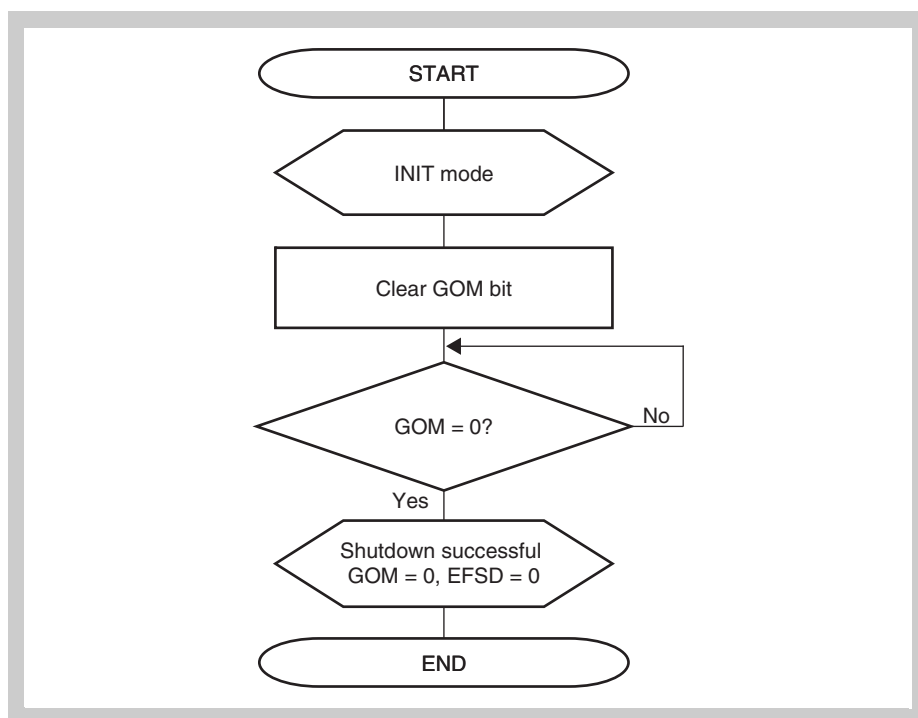


Figure 14-57 Normal shutdown process

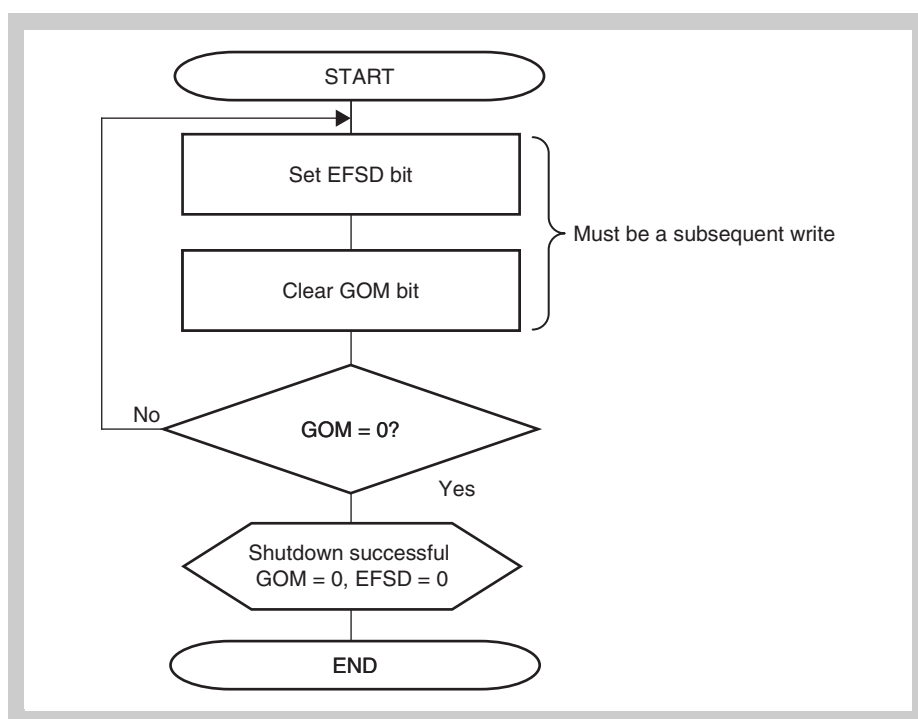


Figure 14-58 Forced shutdown process

**Caution** If a read/write access to any other register is executed by software (include EFSD bit setting becomes invalid and the GOM bit is not cleared.  
(refer to 14.7 (1) "CnGMCTRL - CANn global control register")

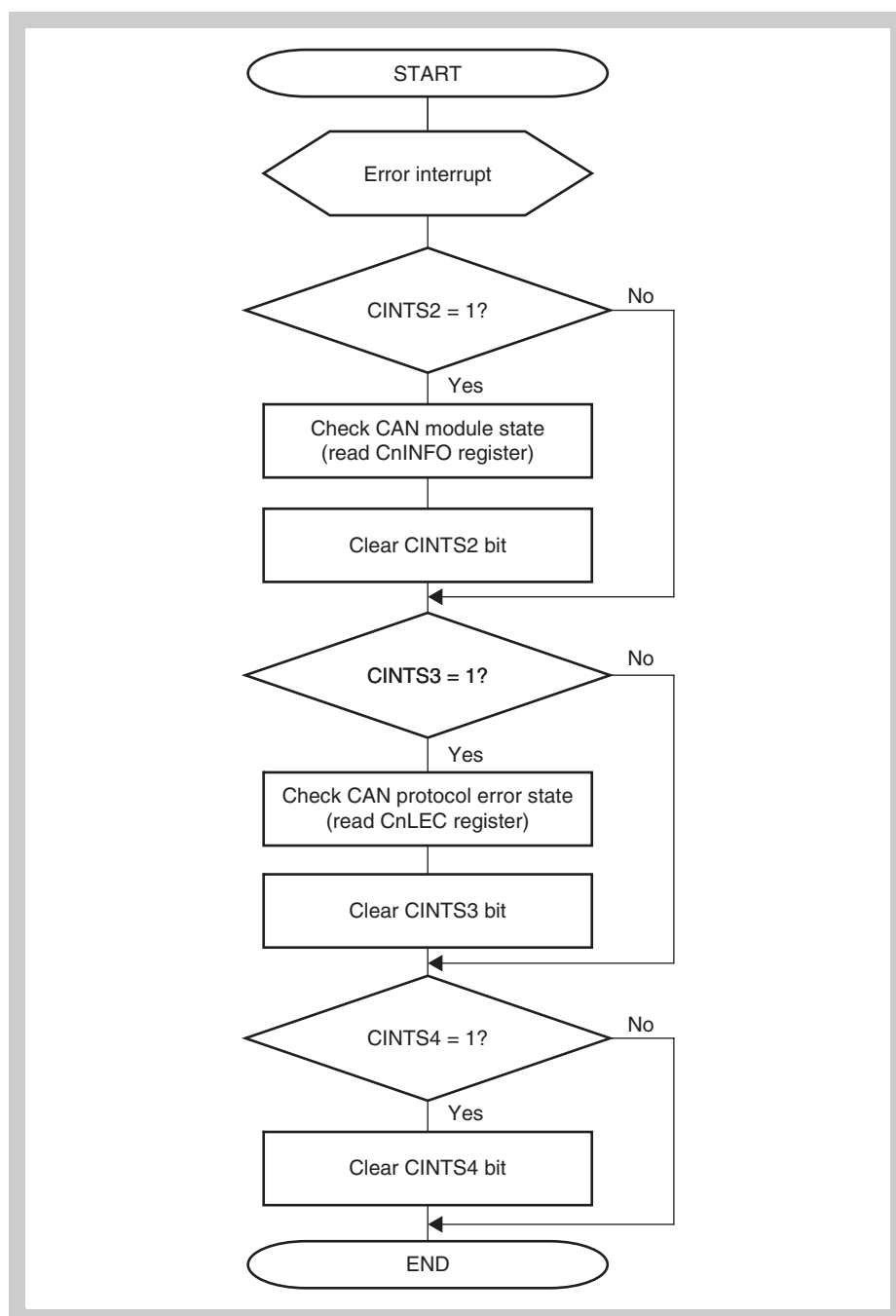
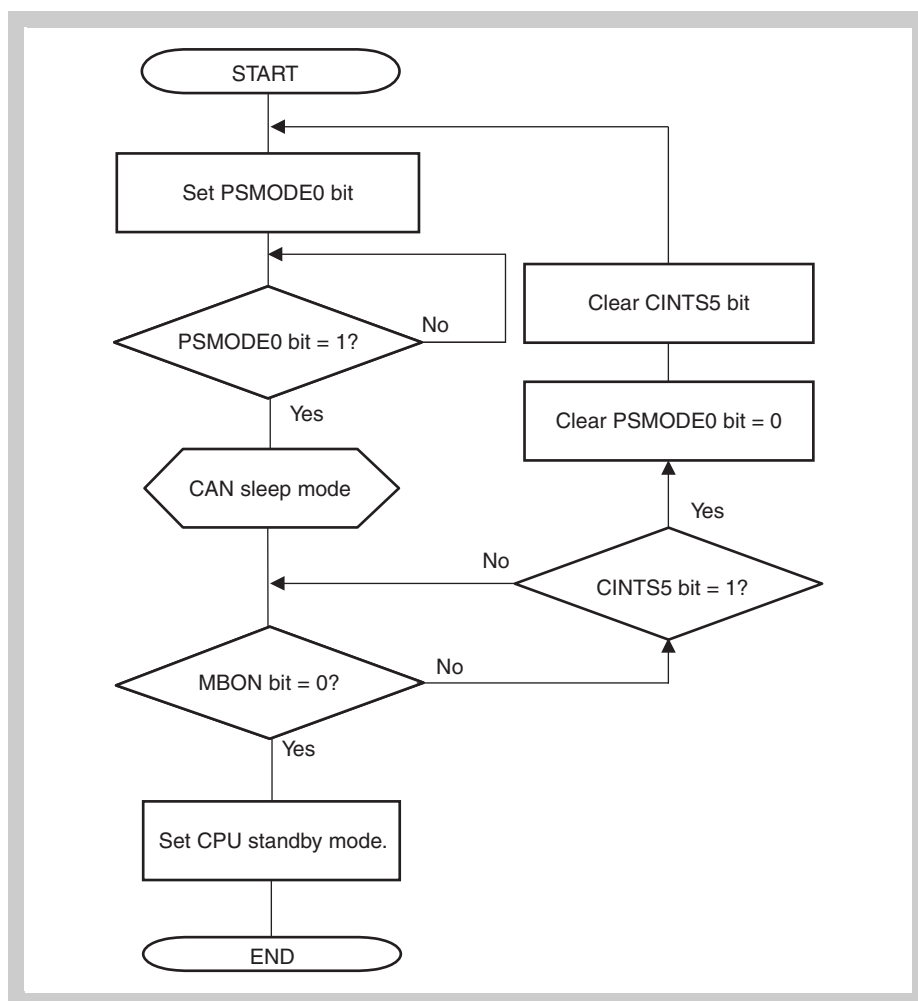


Figure 14-59 Error handling



**Figure 14-60** Setting CPU stand-by (from CAN sleep mode)

**Caution** Before the CPU is set in the CPU standby mode, please check if the CAN sleep mode has been reached. However, after check of the CAN sleep mode, until the CPU is set in the CPU standby mode, the CAN sleep mode may be cancelled by wakeup from CAN bus.

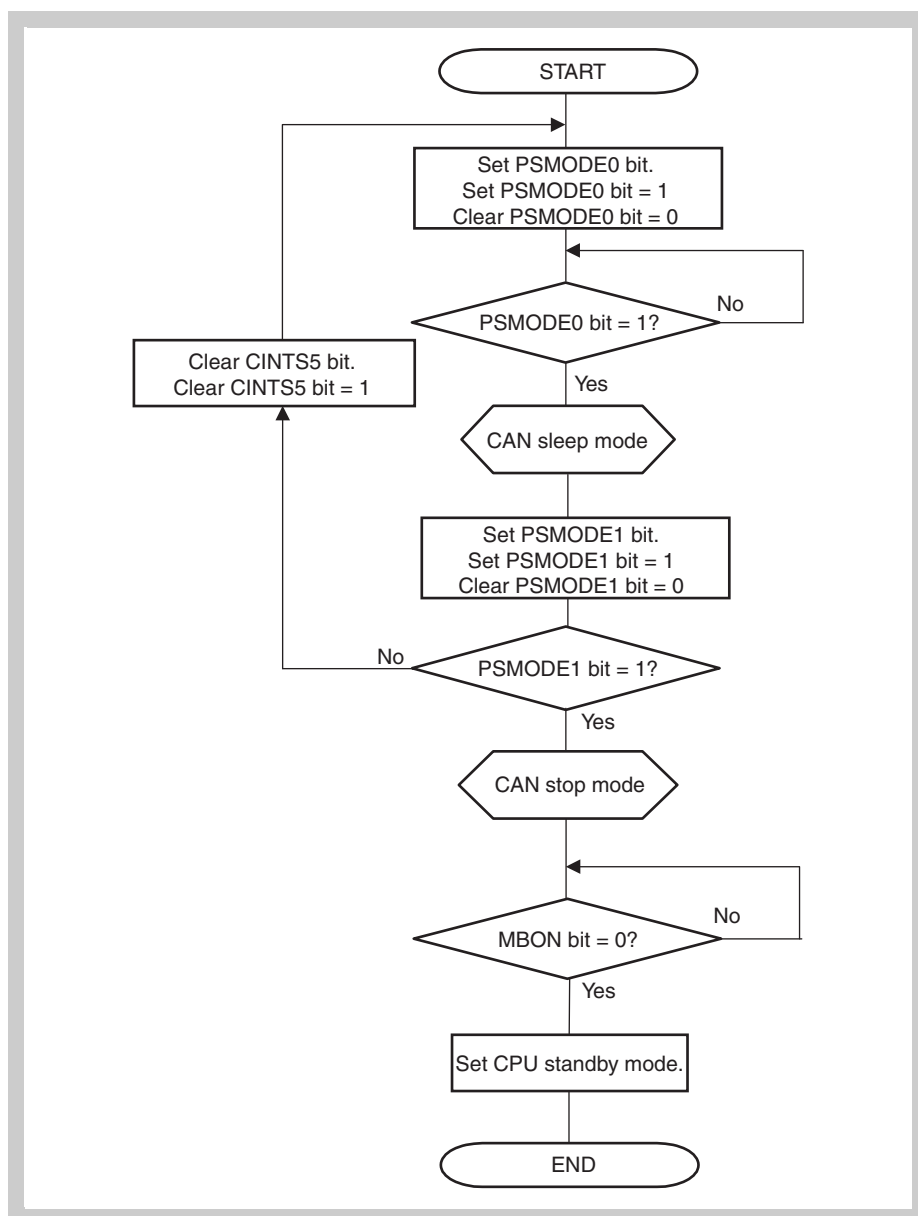


Figure 14-61 Setting CPU stand-by (from CAN stop mode)

**Caution** The CAN stop mode can only be released by writing 01<sub>B</sub> to the PSMODE[1:0] bit of the CnCTRL register and not by a change in the CAN bus state.

# Chapter 15 A/D Converter (ADC)

This microcontroller has two instances of the A/D Converter.

**Note** Throughout this chapter, the individual instances of the A/D Converter are identified by “n” (n = 0, 1). Each A/D Converter provides 10 channels, which are identified by “m” (m = 0 to 9).

## 15.1 Features

- Analog input: 2 × 10 channels (ANI00 to ANI09, ANI10 to ANI19)
- 10-bit resolution
- On-chip A/D conversion result register (ADCRn0 to ADCRn9):  
10 bits × 10
- A/D conversion trigger mode
  - A/D trigger mode
  - Timer trigger mode
  - External trigger mode
- Successive approximation method
- DMA transfer support of A/D conversion result to internal RAM

## 15.2 Configuration

The A/D Converter adopts the successive approximation method.

### (1) Input circuit

The input circuit selects the analog input (ANIn0 to ANIn9) according to the mode set by the ADMn0, ADMn1, and ADMn2 registers.

### (2) C-Array

Holds the charge of the differential voltage between the voltage input from the analog input pins (ANIn0 to ANIn9) and the reference voltage ( $1/2 AV_{DD}$ ), and redistributes the sampled charges.

### (3) C-Dummy

This block holds the reference voltage ( $1/2 AV_{DD}$ ) and assigns the reference of the comparator input.

### (4) Voltage Comparator

The Voltage Comparator compares the C-Array comparison potential with the C-Dummy reference potential.

### (5) A/D conversion result register (ADCRnm), A/D conversion result register nH (ADCRnmH)

ADCRnm is a 10-bit register that holds A/D conversion results. Each time A/D conversion is completed, the conversion results are loaded from the successive approximation register (SAR).

$\overline{\text{RESET}}$  input makes this register undefined.

### (6) A/D conversion result register for DMA transfer (ADDMAAn)

ADDMAAn is a 16-bit register that holds the last 10-bit A/D conversion result and an overrun flag for indicating a DMA transfer failure.

### (7) ANIn0 to ANIn9 pins

These are 10-channel analog input pins for the A/D Converter n. They input the analog signals to be A/D converted.

---

**Caution** Make sure that the voltages input to ANIn0 to ANIn9 do not exceed the rated values. If a voltage higher than  $AV_{DD}$  or lower than  $AV_{SSn}$  (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.

---

**(8)  $AV_{REFn}$  pins**

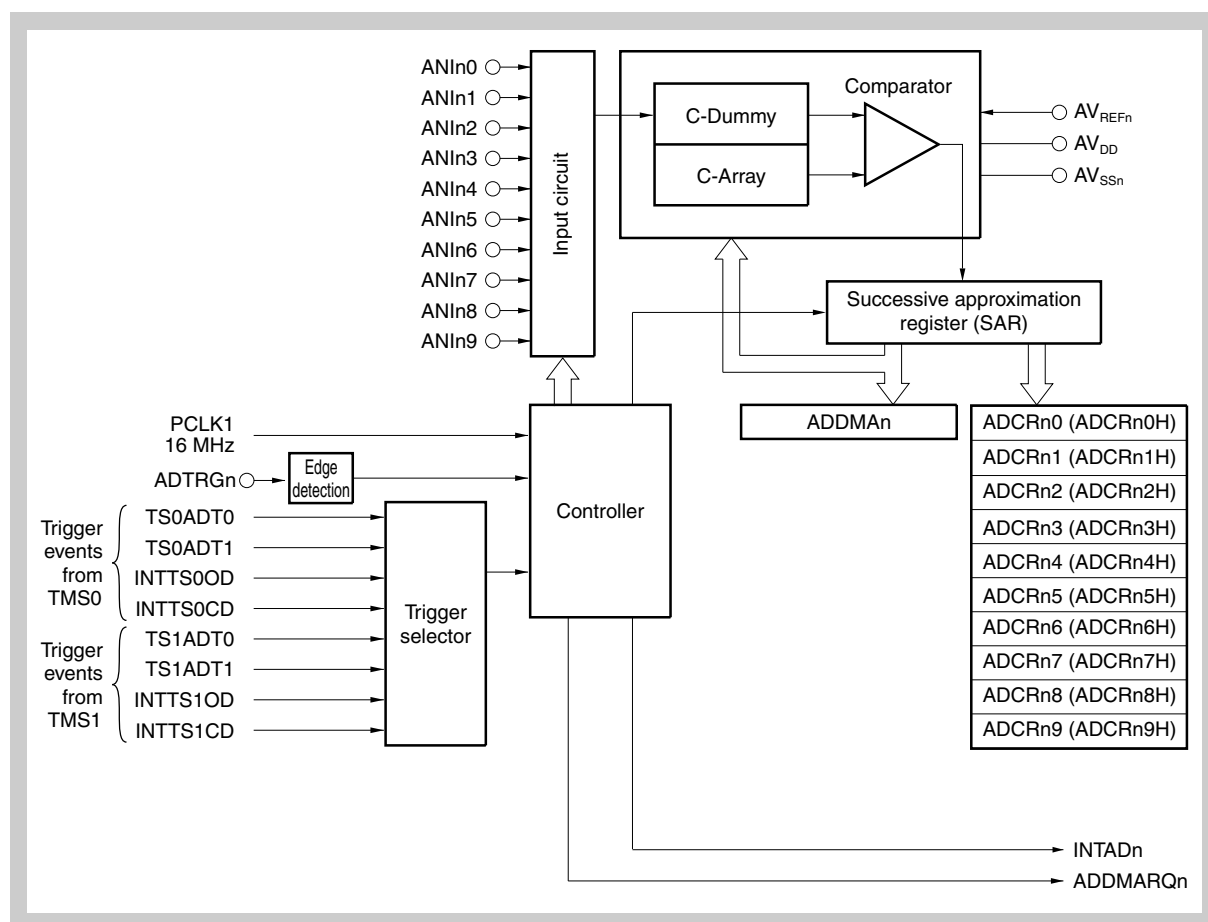
This is the pin for inputting the reference voltage of the A/D Converter. It converts signals input to the ANIn0 to ANIn9 pins to digital signals based on the voltage applied between  $AV_{SSn}$  and  $AV_{REFn}$ .

**(9)  $AV_{SSn}$  pin**

This is the ground pin of the A/D Converter. Always use this pin at the same potential as that of the  $EV_{SS}$  pin even when the A/D Converter is not used.

**(10)  $AV_{DD}$  pin**

This is the analog power supply pin of both A/D Converters (ADC0, ADC1).



**Figure 15-1** Block diagram of A/D Converter (ADCn)



- 
- Caution**
1. If there is noise at the analog input pins (ANIn0 to ANIn9) or at the reference voltage input pin ( $AV_{REFn}$ ), 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.
  2. Do not apply a voltage outside the  $AV_{SSn}$  to  $AV_{REFn}$  range to the pins that are used as A/D Converter input pins.
-

## 15.3 Control Registers

### (1) ADMn0 - A/D Converter mode register 0

The ADMn0 register is an 8-bit register that specifies the operation mode, and executes conversion operations.

**Access** This register can be read/written in 8-bit or 1-bit units. However, bit 6 can only be read. Writing this bit is ignored.

**Address** ADM00: FFFFF200<sub>H</sub>  
ADM10: FFFFF240<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
ADMn0	ADCEn	ADCSn	BSn	MSn	0	02	0	0
	R/W	R	R/W	R/W	R	R	R	R

- Caution**
- When the ADCEn bit is 1 in the timer trigger mode and external trigger mode, the trigger signal standby state is set. To clear the ADCEn bit, write 0 or reset.  
In A/D trigger mode (ADMn1.TRGn[1:0] = 00<sub>B</sub>), the conversion trigger is set by writing 1 to the ADCEn bit. After the operation, when the mode is changed to the timer trigger mode or external trigger mode without clearing the ADCEn bit, the trigger input standby state is set immediately after changing the register.
  - The A/D conversion can be restarted by writing to ADMn0 while A/D conversion is enabled (ADMn0.ADCEn = 1).  
In that case the conversion operation is initialized and conversion is executed from the beginning after the A/D converter stabilization time has elapsed.  
Note that the BSn and MSn bits must not be changed while A/D conversion is enabled, thus the same values must be re-written to restart the conversion.

**Table 15-1 ADMn0 register contents**

Bit position	Bit name	Function
7	ADCEn	Controls the A/D conversion operation of ADCn. 0: A/D conversion operation disabled. 1: A/D conversion operation enabled.
6	ADCSn	Indicates the A/D conversion status of ADCn. 0: A/D conversion of ADCn is stopped. 1: A/D conversion of ADCn is operating.
5	BSn	Specifies the buffer mode of ADCn. 0: 1-buffer mode 1: 4-buffer mode
4	MSn	Specifies the operation mode of ADCn. 0: Scan mode 1: Scan mode

**(2) ADMn1 - A/D Converter mode register 1**

The ADMn1 register is an 8-bit register that specifies the conversion operation time and trigger mode.

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

**Address** ADM01: FFFFF201<sub>H</sub>  
ADM11: FFFFF241<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>ADMn1</b>	EGAn1	EGAn0	TRGn1	TRGn0	0	02	FRn1	FRn0
	R/W	R/W	R/W	R/W	R	R	R/W	R/W

- Caution**
1. Changing the setting of ADMn1 is prohibited while A/D conversion is enabled (ADMn0.ADCEn = 1).
  2. The A/D conversion can be restarted by writing to ADMn1 while A/D conversion is enabled (ADMn0.ADCEn = 1).  
In that case the conversion operation is initialized and conversion is executed from the beginning after the A/D converter stabilization time has elapsed.  
Note that the ADnM1 value must not be changed while A/D conversion is enabled, thus the same value must be re-written to restart the conversion.

**Table 15-2 ADMn1 register contents (1/2)**

Bit position	Bit name	Function		
7, 6	EGAn1 EGAn0	Specifies the valid edge of the external trigger input (ADTRGn).		
		EGAn1	EGAn0	Valid Edge Specification of External Trigger Input (ADTRGn)
		0	0	No edge detected (does not operate as external trigger)
		0	1	Falling edge detected
		1	0	Rising edge detected
		1	1	Both edges, falling and rising edge detected
5, 4	TRGn1 TRGn0	Specifies the trigger mode of ADCn.		
		TRGn1	TRGn0	ADCn Trigger Mode Specification
		0	0	A/D trigger mode (conversion starts with ADMn0.ADCEn = 1)
		0	1	Timer trigger mode
		1	0	External trigger mode
		1	1	Setting prohibited

Table 15-2 ADMn1 register contents (2/2)

Bit position	Bit name	Function				
1, 0	FRn1 FRn0	Specifies the conversion time of ADCn.				
		FRn1	FRn0	Number of conversion clocks	Conversion Operation Time	A/D Stabilization Time <sup>Note</sup>
		0	0	32	2.0 μs	1.0 μs
		0	1	64	4.0 μs	2.0 μs
		1	0	96	6.0 μs	2.5 μs
		1	1	128	8.0 μs	2.5 μs
		Others than above		Setting prohibited		
		<b>Note:</b> After the ADMn0.ADCEn bit is set from 0 to 1, to secure the stabilization time of the A/D Converter, conversion is started after the A/D stabilization time has elapsed only before the first A/D conversion is executed.				

**(3) ADMn2 - A/D Converter mode register 2**

The ADMn2 register is an 8-bit register that specifies the analog input pin of the A/D Converter n.

**Access** This register can be read/written in 8-bit or 1-bit units

**Address** ADM02: FFFFF202<sub>H</sub>  
ADM12: FFFFF242<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>ADMn2</b>	0	0	0	0	ANISn3	ANISn2	ANISn1	ANISn0
	R	R	R	R	R/W	R/W	R/W	R/W

- Caution**
1. If a channel for which no analog input pin exists is specified, the result of A/D conversion is undefined.
  2. Changing the setting of the ANISn3 to ANISn0 bits is prohibited while A/D conversion is enabled (ADMn0.ADCEn = 1).
  3. The A/D conversion can be restarted by writing to ADMn1 while A/D conversion is enabled (ADMn0.ADCEn = 1).  
In that case the conversion operation is initialized and conversion is executed from the beginning after the A/D converter stabilization time has elapsed.  
Note that the ADMn2 value must not be changed while A/D conversion is enabled, thus the same value must be re-written to restart the conversion.

**Table 15-3 ADMn2 register contents**

Bit position	Bit name	Function					
3 to 0	ANIn[3:0]	Specifies the analog input pins for A/D conversion of ADCn.					
		ANIn3	ANIn2	ANIn2	ANIn0	Specification of Analog Input Pins for A/D Conversion	
						Select Mode	Scan Mode
		0	0	0	0	ANIn0	ANIn0
		0	0	0	1	ANIn1	ANIn0, ANIn1
		0	0	1	0	ANIn2	ANIn0 to ANIn2
		0	0	1	1	ANIn3	ANIn0 to ANIn3
		0	1	0	0	ANIn4	ANIn0 to ANIn4
		0	1	0	1	ANIn5	ANIn0 to ANIn5
		0	1	1	0	ANIn6	ANIn0 to ANIn6
		0	1	1	1	ANIn7	ANIn0 to ANIn7
		1	0	0	0	ANIn8	ANIn0 to ANIn8
		1	0	0	1	ANIn9	ANIn0 to ANIn9
		Others than above				Setting prohibited	

**(4) ADTRSELn - A/D Converter trigger source select register**

The ADTRSELn register is an 8-bit register that specifies the timer trigger signal in the timer trigger mode (TRGn1, TRGn0 bits of ADMn1 register = 01<sub>B</sub>).

**Access** This register can be read/written in 8-bit or 1-bit units

**Address** ADTRSEL0: FFFFF270<sub>H</sub>  
ADTRSEL1: FFFFF272<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>ADTRSELn</b>	0	0	0	0	TSELn3	TSELn2	TSELn1	TSELn0
	R	R	R	R	R/W	R/W	R/W	R/W

**Caution** Before changing the setting of the ADTRSELn register, stop the A/D conversion operation (by clearing the ADMn0.ADCEn bit to 0). The operation is not guaranteed if the setting of the ADTRSELn register is changed while A/D conversion is enabled (ADMn0.ADCEn = 1).

**Table 15-4 ADMn2 register contents**

Bit position	Bit name	Function				
3 to 0	TSELn[3:0]	Specifies the analog input pins for A/D conversion of ADCn.				
		TSELn3	TSELn2	TSELn1	TSELn0	Trigger Source Selection in Timer Trigger Mode
		0	0	0	1	TS0ADT0 signal (from TMS0)
		0	0	1	0	TS0ADT1 signal (from TMS0)
		0	0	1	1	TS1ADT0 signal (from TMS1)
		0	1	0	0	TS1ADT1 signal (from TMS1)
		0	1	0	1	INTTS0OD interrupt (from TMS0)
		0	1	1	0	INTTS0CD interrupt (from TMS0)
		0	1	1	1	INTTS1OD interrupt (from TMS1)
		1	0	0	0	INTTS1CD interrupt (from TMS1)
		Others than above				Setting prohibited

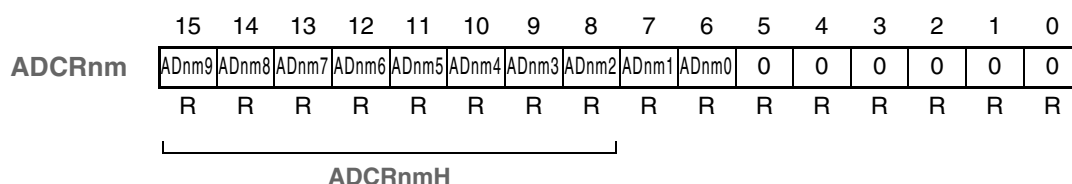
**(5) ADCRn0 to ADCRn9, ADCRn0H to ADCRn9H - A/D conversion result registers**

The ADCRnm register is a 10-bit register holding the A/D conversion results.

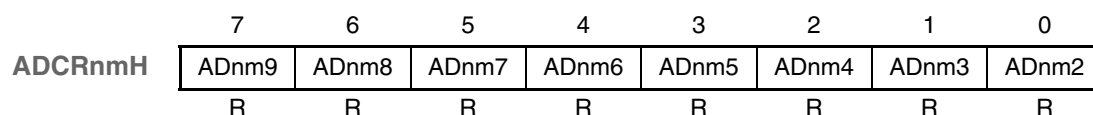
**Access** These registers are read-only in 16-bit or 8-bit units. When 16-bit access is performed, the ADCRnm register is specified, and when 8-bit access is performed, the ADCRnmH register holding the higher 8 bits of the conversion result is specified

<b>Address</b>	ADCR00:	FFFFFF210 <sub>H</sub>	ADCR00H:	FFFFFF211 <sub>H</sub>
	ADCR01:	FFFFFF212 <sub>H</sub>	ADCR01H:	FFFFFF213 <sub>H</sub>
	ADCR02:	FFFFFF214 <sub>H</sub>	ADCR02H:	FFFFFF215 <sub>H</sub>
	ADCR03:	FFFFFF216 <sub>H</sub>	ADCR03H:	FFFFFF217 <sub>H</sub>
	ADCR04:	FFFFFF218 <sub>H</sub>	ADCR04H:	FFFFFF219 <sub>H</sub>
	ADCR05:	FFFFFF21A <sub>H</sub>	ADCR05H:	FFFFFF21B <sub>H</sub>
	ADCR06:	FFFFFF21C <sub>H</sub>	ADCR06H:	FFFFFF21D <sub>H</sub>
	ADCR07:	FFFFFF21E <sub>H</sub>	ADCR07H:	FFFFFF21F <sub>H</sub>
	ADCR08:	FFFFFF220 <sub>H</sub>	ADCR08H:	FFFFFF221 <sub>H</sub>
	ADCR09:	FFFFFF222 <sub>H</sub>	ADCR09H:	FFFFFF223 <sub>H</sub>
	ADCR10:	FFFFFF250 <sub>H</sub>	ADCR10H:	FFFFFF251 <sub>H</sub>
	ADCR11:	FFFFFF252 <sub>H</sub>	ADCR11H:	FFFFFF253 <sub>H</sub>
	ADCR12:	FFFFFF254 <sub>H</sub>	ADCR12H:	FFFFFF255 <sub>H</sub>
	ADCR13:	FFFFFF256 <sub>H</sub>	ADCR13H:	FFFFFF257 <sub>H</sub>
	ADCR14:	FFFFFF258 <sub>H</sub>	ADCR14H:	FFFFFF259 <sub>H</sub>
	ADCR15:	FFFFFF25A <sub>H</sub>	ADCR15H:	FFFFFF25B <sub>H</sub>
	ADCR16:	FFFFFF25C <sub>H</sub>	ADCR16H:	FFFFFF25D <sub>H</sub>
	ADCR17:	FFFFFF25E <sub>H</sub>	ADCR17H:	FFFFFF25F <sub>H</sub>
	ADCR18:	FFFFFF260 <sub>H</sub>	ADCR18H:	FFFFFF261 <sub>H</sub>
	ADCR19:	FFFFFF262 <sub>H</sub>	ADCR19H:	FFFFFF263 <sub>H</sub>

**Initial Value** Undefined



**Note** When reading the 10-bit data of the A/D conversion results from the ADCRnm register, only the higher 10 bits are valid and the lower 6 bits are always read as 0.



The correspondence between each analog input pin and the ADCRn register is shown in Table 15-5 below.

**Table 15-5 Assignment of A/D conversion result registers to analog input pins**

Analog Input Pin	Assignment of A/D Conversion Result Registers	
	Select 1 Buffer Mode/ Scan Mode	Select 4 Buffer Mode
ANIn0	ADCRn0, ADCRn0H	ADCRn0 to ADCRn3, ADCRn0H to ADCRn3H
ANIn1	ADCRn1, ADCRn1H	
ANIn2	ADCRn2, ADCRn2H	
ANIn3	ADCRn3, ADCRn3H	
ANIn4	ADCRn4, ADCRn4H	ADCRn4 to ADCRn7, ADCRn4H to ADCRn7H
ANIn5	ADCRn5, ADCRn5H	
ANIn6	ADCRn6, ADCRn6H	
ANIn7	ADCRn7, ADCRn7H	
ANIn8	ADCRn8, ADCRn8H	ADCRn0 to ADCRn3, ADCRn0H to ADCRn3H
ANIn9	ADCRn9, ADCRn9H	

The relationship between the analog voltage input to the analog input pins (ANIn0 to ANIn9) and the A/D conversion result (of the A/D conversion result register (ADCRn)) is as follows:

$$\text{ADCR} = \text{INT}\left(\frac{V_{\text{IN}}}{AV_{\text{REF}}} \times 1024 + 0,5\right)$$

or,

$$(\text{ADCR} - 0,5) \times \frac{AV_{\text{REF}}}{1024} \leq V_{\text{IN}} < (\text{ADCR} + 0,5) \times \frac{AV_{\text{REF}}}{1024}$$

INT(): Function that returns the integer value

V<sub>IN</sub>: Analog input voltage

AV<sub>REF</sub>: AV<sub>REF</sub> pin voltage

ADCR: Value of the upper 10 bit of the A/D conversion result register, i.e. ADCRn[15:6]

Figure 15-2 on page 517 shows the relationship between the analog input voltage and the A/D conversion results.



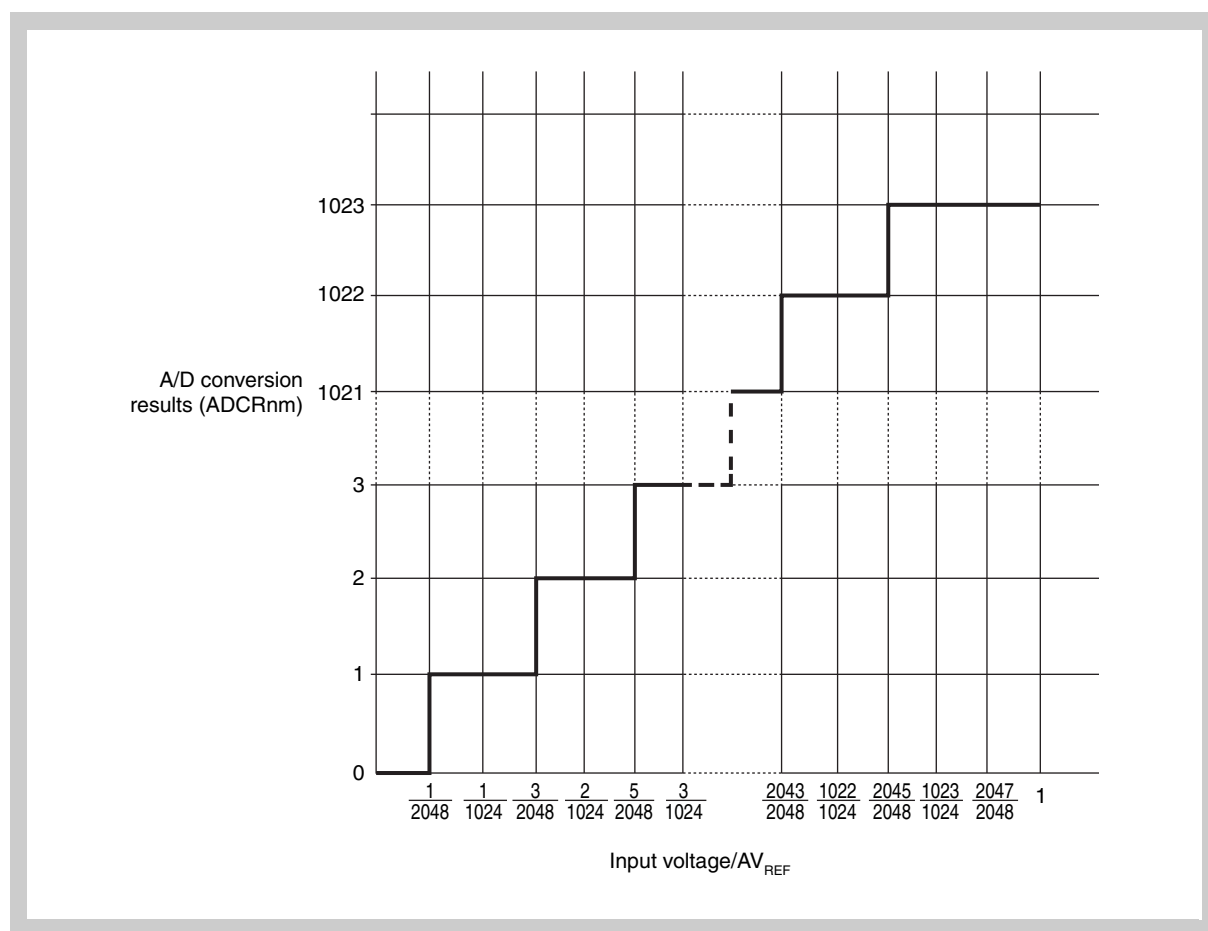


Figure 15-2 Relationship between analog input voltage and A/D conversion results

**(6) ADDMA<sub>n</sub> - A/D conversion result register for DMA**

The ADDMA<sub>n</sub> register is a 16-bit register holding the result of the latest A/D conversion operation, and is used for DMA transfer of ADC<sub>n</sub> results into the internal RAM. It has an overrun detection flag indicating an overrun situation of the DMA transfer mechanism.

**Access** This register is read-only in 16-bit units.

**Address** ADDMA0: FFFFF22E<sub>H</sub>  
ADDMA1: FFFFF26E<sub>H</sub>

**Initial Value** Undefined

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>ADDMA<sub>n</sub></b>	ADDMA <sub>n</sub> 9	ADDMA <sub>n</sub> 8	ADDMA <sub>n</sub> 7	ADDMA <sub>n</sub> 6	ADDMA <sub>n</sub> 5	ADDMA <sub>n</sub> 4	ADDMA <sub>n</sub> 3	ADDMA <sub>n</sub> 2	ADDMA <sub>n</sub> 1	ADDMA <sub>n</sub> 0	0	0	0	0	0	ODF <sub>n</sub>
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

**Caution** Do not read the ADDMA<sub>n</sub> register by CPU during DMA transfer activities. If this register is read by CPU, overflow detection cannot be ensured.

**Table 15-6 ADDMA<sub>n</sub> register contents**

Bit position	Bit name	Function				
15 to 6	ADDMA <sub>n</sub> [9:0]	Returns the recent A/D conversion result of ADC <sub>n</sub> required for DMA transfer..				
		<table><tr><th>ADDMA<sub>n</sub>[9:0]</th><th>A/D Conversion Result for DMA Transfer</th></tr><tr><td>000<sub>H</sub> to 3FF<sub>H</sub></td><td>Value of the recent A/D conversion result</td></tr></table>	ADDMA <sub>n</sub> [9:0]	A/D Conversion Result for DMA Transfer	000 <sub>H</sub> to 3FF <sub>H</sub>	Value of the recent A/D conversion result
		ADDMA <sub>n</sub> [9:0]	A/D Conversion Result for DMA Transfer			
000 <sub>H</sub> to 3FF <sub>H</sub>	Value of the recent A/D conversion result					
0	ODF <sub>n</sub>	<p>Indicates an overrun of the A/D conversion result in ADDMA<sub>n</sub> register.</p> <p>0: No A/D conversion result overrun was detected.</p> <p>1: At least one A/D conversion result was overrun since the last read of the ADDMA<sub>n</sub> register.</p> <ul style="list-style-type: none"><li>The ODF<sub>n</sub> flag is used for indicating a DMA transfer failure of the A/D conversion results.</li><li>The ODF<sub>n</sub> flag is cleared (0), when the A/D conversion is stopped (ADM<sub>n</sub>0.ADCEn is cleared to 0).</li></ul>				

## 15.4 Operation

### 15.4.1 Basic operation

A/D conversion is executed by the following procedure.

1. The selection of the analog input and specification of the operation mode, trigger mode, etc. should be specified using the ADMn0, ADMn1 or ADMn2 registers<sup>Note 1</sup>.  
When the ADCEn bit of the ADMn0 register is set to 1, A/D conversion starts in the A/D trigger mode. In the timer trigger mode and external trigger mode, the trigger standby state<sup>Note 2</sup> is set.
2. When A/D conversion is started, the C-array voltage on the analog input side and the C-array voltage on the reference side are compared by the comparator.
3. When the comparison of the 10 bits ends, the conversion results are stored in the ADCRnm register. When A/D conversion has been performed the specified number of times, the A/D conversion end interrupt (INTADn) is generated.

- Note**
1. If the setting of the ADMn0, ADMn1 or ADMn2 registers is changed during A/D conversion, the operation immediately before is stopped, and the result of the conversion is not stored in the ADCRnm register. The A/D conversion operation is then initialized, and conversion is executed from the beginning again.
  2. During the timer trigger mode and external trigger mode, if the ADCEn bit of the ADMn0 register is set to 1, the mode changes to the trigger standby state. The A/D conversion operation is started by the trigger signal (ADCSn bit in the ADMn0 register = 1), and the trigger standby state (ADCSn bit = 0) is returned when the A/D conversion operation ends.

### 15.4.2 Operation mode and trigger mode

Various conversion operations can be specified for the A/D Converter by specifying the operation mode and trigger mode. The operation mode and trigger mode are set by the ADMn0 and ADMn1 registers.

The following table shows the relationship between the operation mode and trigger mode.

**Table 15-7 Relationship between operation mode and trigger mode**

Trigger Mode	Operation Mode		Register Set Value	
			ADMn0	ADMn1
A/D trigger	Select	1 buffer	xx010000B	xx0000xxB
		4 buffers	xx110000B	
	Scan		xx000000B	
Timer trigger	Select	1 buffer	xx010000B	xx0100xxB
		4 buffers	xx110000B	
	Scan		xx000000B	
External trigger	Select	1 buffer	xx010000B	xx1000xxB
		4 buffers	xx110000B	
	Scan		xx000000B	

#### (1) Trigger mode

There are three types of trigger modes that serve as the start timing of A/D conversion processing: A/D trigger mode, timer trigger mode, and external trigger mode. These trigger modes are set by the TRGn1 and TRGn0 bits of the ADMn1 register.

- **A/D trigger mode**

This mode starts the conversion timing of the analog input set to the ANIn0 to ANIn9 pins, and by setting the ADMn0.ADCEn bit to 1, starts A/D conversion. Unless the ADCEn bit is cleared to 0 after conversion, the next conversion operation is repeated. If data is written to the ADMn0 to ADMn2 registers during conversion, conversion is stopped and then executed from the beginning again.

- **Timer trigger mode**

This mode specifies the conversion timing of the analog input set for the ANIn0 to ANIn9 pins using signals from the inverter timer S (TMS0, TMS1). The ADTRSELn register specifies the analog input conversion timing by selecting one of the A/D Converter trigger signals connected to the 16-bit inverter timer S (TMS0, TMS1).

If the ADMn0.ADCEn bit is set to 1, the A/D Converter waits for an event input, and starts conversion when the event occurs (ADMn0.ADCSn = 1). When conversion has finished, the converter waits for an event input again (ADMn0.ADCSn bit = 0). If data is written to the ADMn0 to ADMn2 registers during conversion, conversion is stopped and then executed from the beginning again.

- **External trigger mode**

This mode specifies the conversion timing of the analog input to the ANIn0 to ANIn9 pins using the ADTRGn pin.

The EGAn1 and EGAn0 bits of the ADMn1 register are used to specify the valid edge to be input to the ADTRGn pin.

When the ADMn0.ADCEn bit is set to 1, the A/D Converter waits for an external trigger (ADTRGn), and starts conversion when the valid edge of ADTRGn is detected (ADMn0.ADCSn bit = 1). When the converter has finished its conversion operation, it waits for an external trigger again (ADMn0.ADCSn bit = 0).

If the valid edge is detected at the ADTRGn pin during conversion, conversion is executed from the beginning again.

If data is written to the ADMn0 to ADMn2 registers during conversion, conversion is stopped and then executed from the beginning again.

**(2) Operation mode**

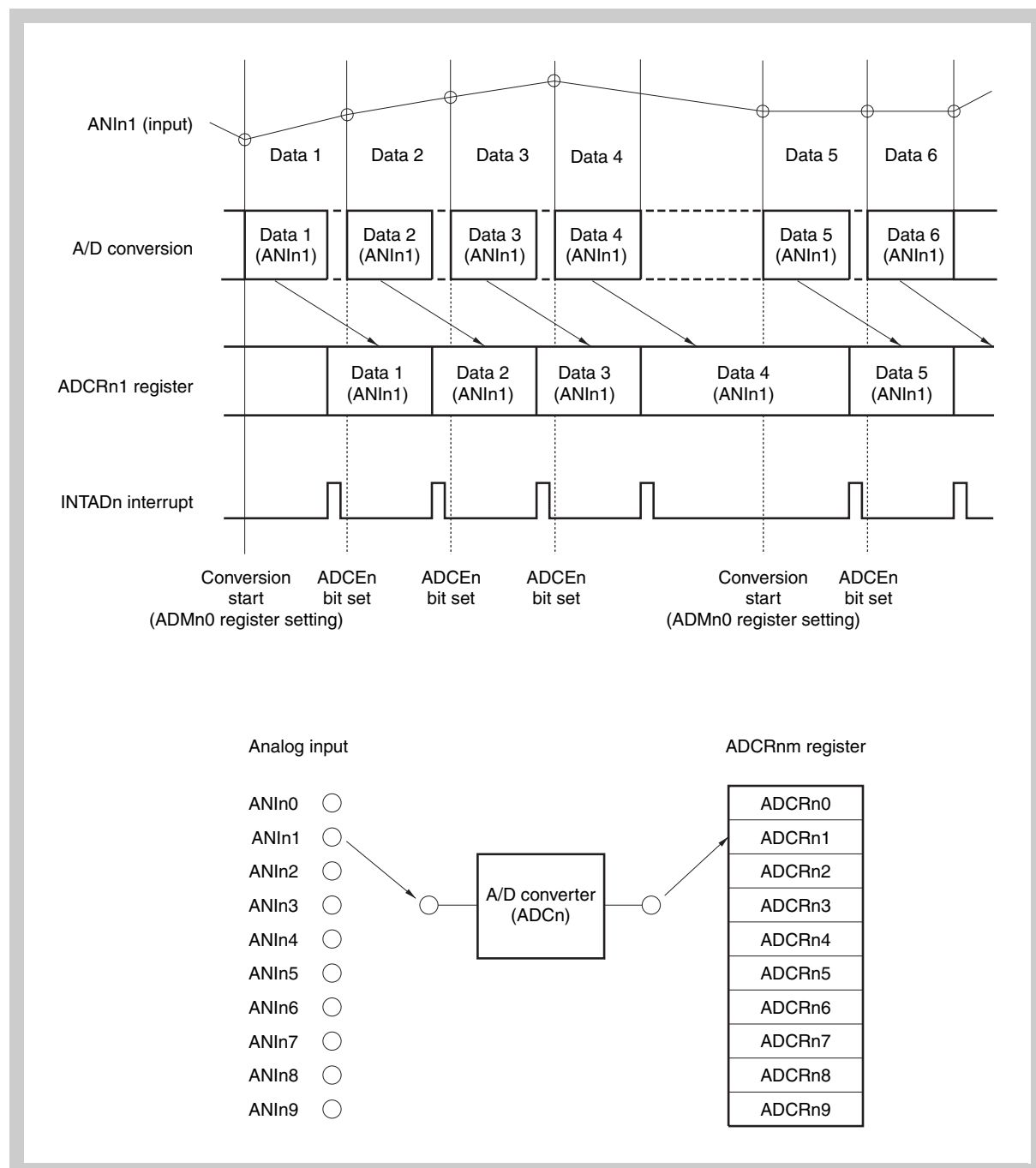
There are two operation modes that set the ANIn0 to ANIn9 pins: select mode and scan mode. The select mode has sub-modes that consist of 1-buffer mode and 4-buffer mode. These modes are set by the BSn and MSn bits of the ADMn0 register.

- **Select mode**

In this mode, one analog input specified by the ADMn2 register is A/D converted. The conversion results are stored in the ADCRnm register corresponding to the analog input (ANInm). For this mode, the 1-buffer mode and 4-buffer mode are provided for storing the A/D conversion results.

### – 1-buffer mode

In this mode, one analog input specified by the ADM2 register is A/D converted. The conversion results are stored in the ADCRnm register corresponding to the analog input (ANInm). The ANInm and ADCRnm register correspond one to one, and an A/D conversion end interrupt (INTADn) is generated each time one A/D conversion ends. After conversion has finished, the next conversion operation is repeated, unless the ADMn0.ADCEn bit is cleared to 0.

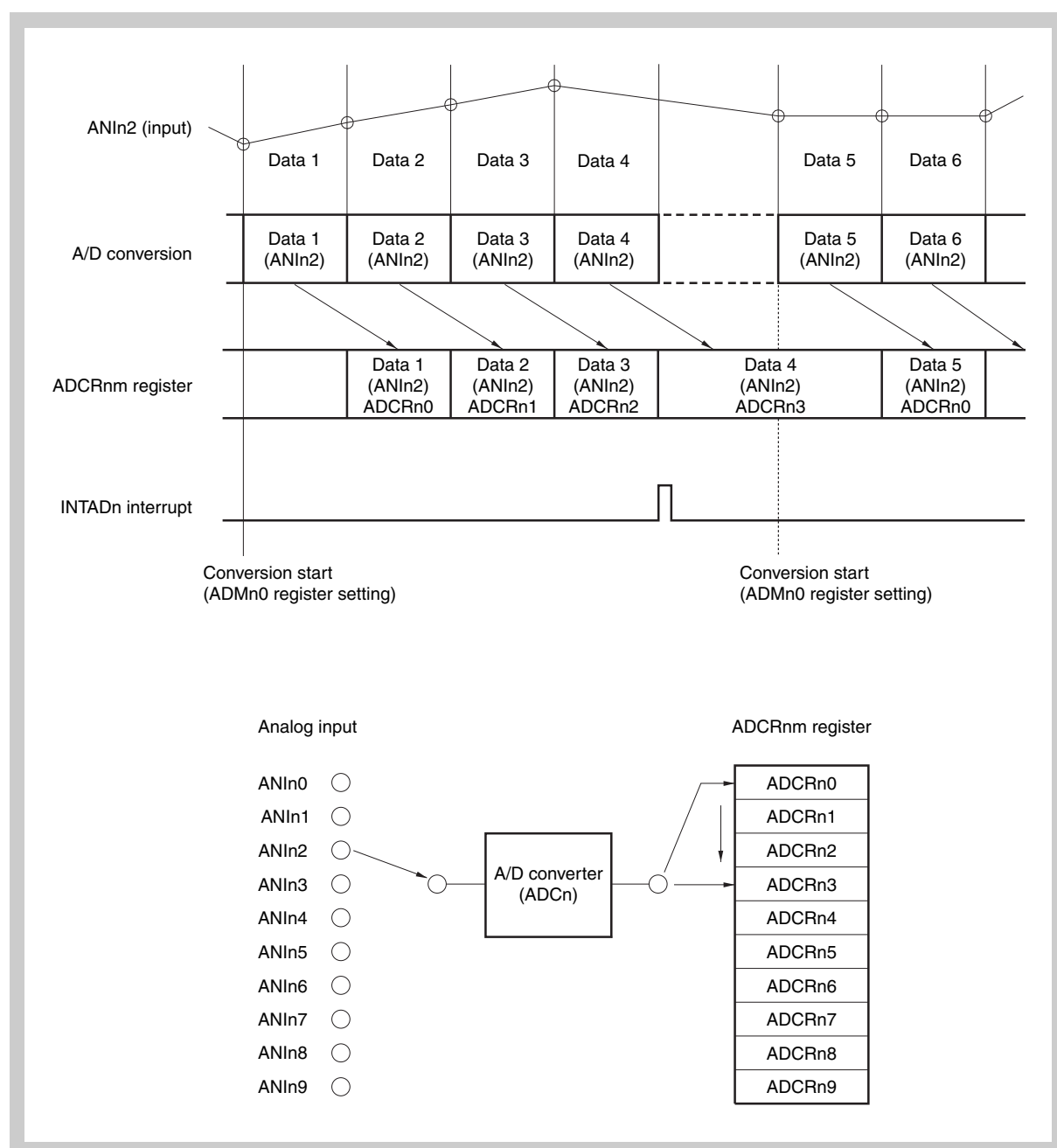


**Figure 15-3** Select mode operation timing: 1-buffer mode (conversion of ANIn1 in A/D trigger mode)

#### – 4-buffer mode

In this mode, one analog input is A/D converted and the results are stored in the ADCRn<sub>m</sub> registers. The A/D conversion end interrupt (INTAD<sub>n</sub>) is generated when the four A/D conversions end ( $m = 0$  to 3 when one of the analog input channels ANIn<sub>0</sub> to ANIn<sub>3</sub> is specified,  $m = 4$  to 7 when one of analog input channels ANIn<sub>4</sub> to ANIn<sub>7</sub> is specified, and  $m = 0$  to 3 when one of the analog input channels ANIn<sub>8</sub> or ANIn<sub>9</sub> is specified).

After conversion has finished, the next conversion operation is repeated, unless the ADMn<sub>0</sub>.ADCE<sub>n</sub> bit is cleared to 0.



**Figure 15-4** Select mode operation timing: 4-buffer mode (conversion of ANIn<sub>2</sub> in A/D trigger mode)

### • Scan mode

In this mode, the analog inputs specified by the ADMn2 register are selected sequentially from the ANIn0 pin, and A/D conversion is executed. The A/D conversion results are stored in the ADCRnm register corresponding to the analog input. When the conversion of the specified analog input ends, the A/D conversion end interrupt (INTADn) is generated. After conversion has finished, the next conversion operation is repeated, unless the ADMn0.ADCEn bit is cleared to 0.

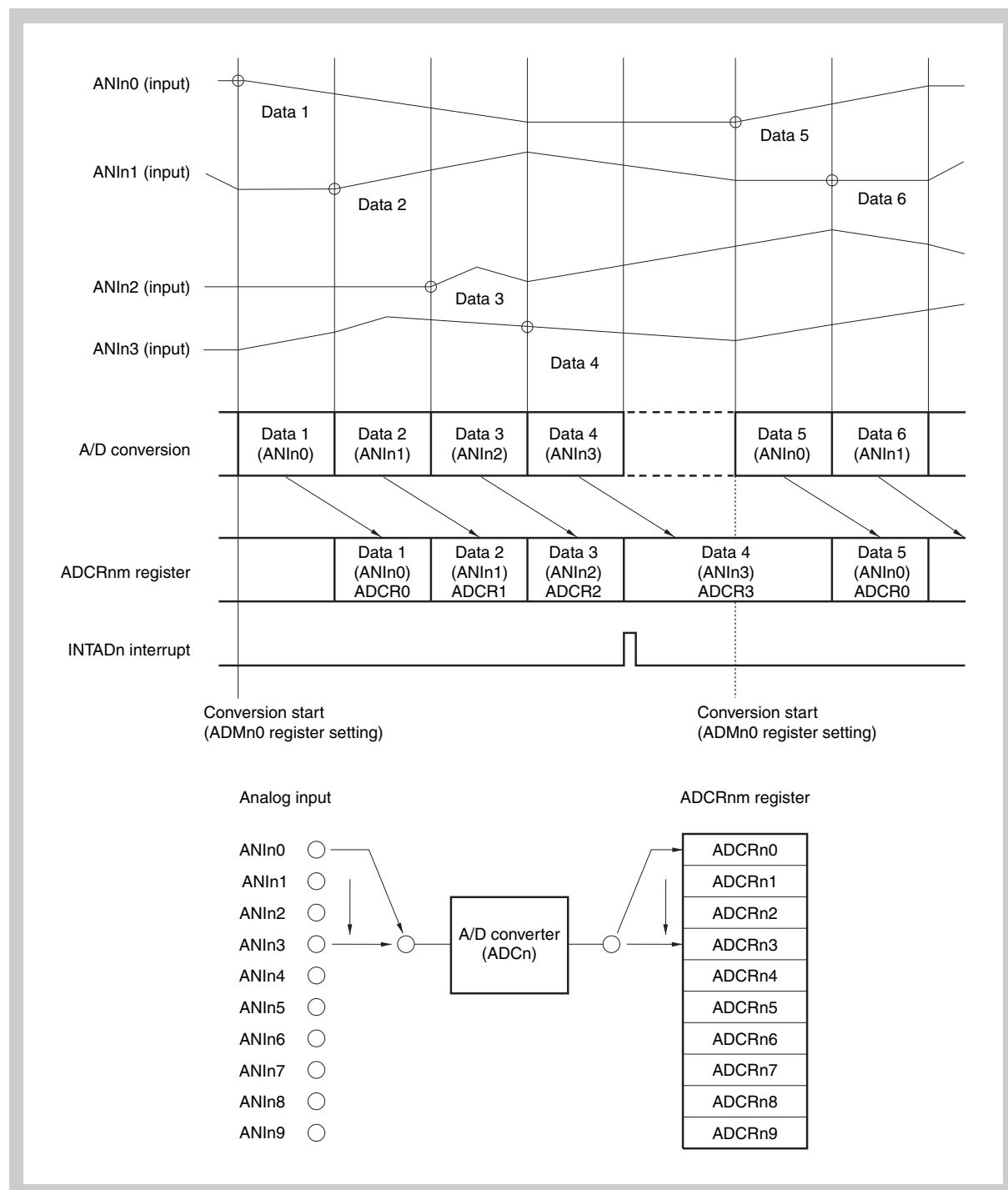


Figure 15-5 Scan mode operation timing: 4-channel scan (conversion of ANI0 to ANI3 in A/D trigger mode)



## 15.5 Operation in A/D Trigger Mode

When the ADMn0.ADCEn bit is set to 1, A/D conversion is started.

### 15.5.1 Select mode operation

In this mode, the analog input specified by the ADMn2 register is A/D converted. The conversion results are stored in the ADCRnm register corresponding to the analog input. In the select mode, the 1-buffer mode and 4-buffer mode are supported according to the storing method of the A/D conversion results.

#### (1) 1-buffer mode (A/D trigger select: 1 buffer)

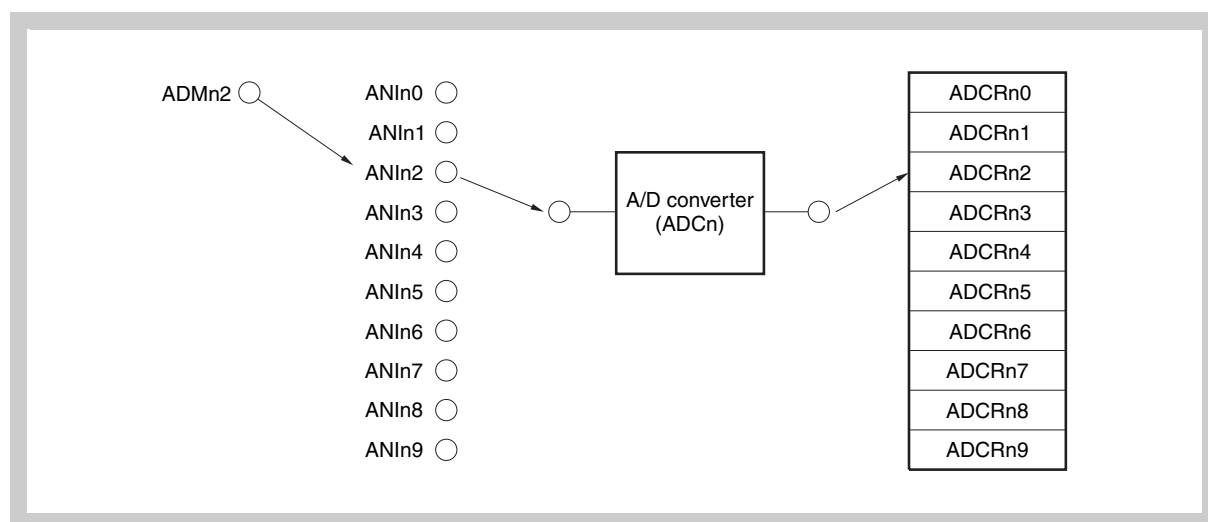
In this mode, one analog input is A/D converted once. The conversion results are stored in one ADCRn register. The analog input and ADCRn register correspond one to one.

Each time an A/D conversion is executed, an A/D conversion end interrupt (INTAD) is generated and A/D conversion ends. The next conversion operation is repeated, unless the ADMn0.ADCE bit is cleared to 0.

**Table 15-8 Correspondence between analog input pins and ADCRnm register (A/D trigger select: 1 buffer)**

Analog Input	A/D Conversion Result Register
ANInm	ADCRnm

This mode is most appropriate for applications in which the results of each first-time A/D conversion are read.



**Figure 15-6 Example of 1-buffer mode operation (A/D trigger select: 1 buffer)**

1. The ADMn0.ADCEn bit is set to 1 (enable)
2. ANIn2 is A/D converted
3. The conversion result is stored in ADCRn2 register
4. The INTAD interrupt is generated

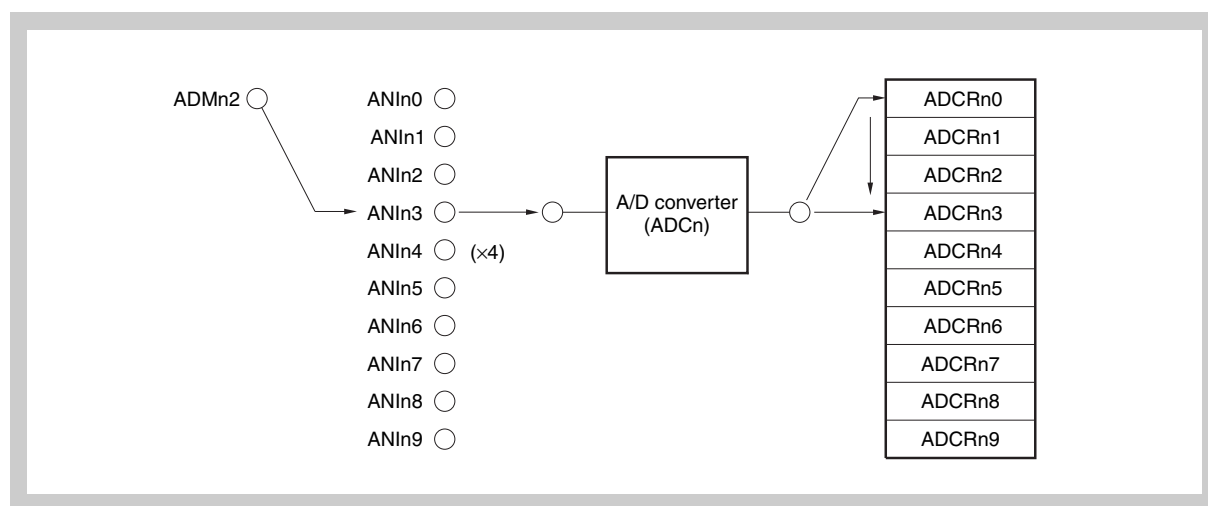
**(2) 4-buffer mode (A/D trigger select: 4 buffers)**

In this mode, one analog input is A/D converted four times and the results are stored in the ADCRn*m* register. When the 4th A/D conversion ends, an A/D conversion end interrupt (INTAD*n*) is generated and the A/D conversion is stopped. The next conversion operation is repeated, unless the ADM*n*0.ADCEn bit is cleared to 0.

**Table 15-9 Correspondence between analog input pins and ADCRn*m* register (A/D trigger select: 4 buffers)**

Analog Input	A/D Conversion Result Register
ANI0 to ANI3	ADCRn0 (1st time)
	ADCRn1 (2nd time)
	ADCRn2 (3rd time)
	ADCRn3 (4th time)
ANI4 to ANI7	ADCRn4 (1st time)
	ADCRn5 (2nd time)
	ADCRn6 (3rd time)
	ADCRn7 (4th time)
ANIn8, ANIn9	ADCRn0 (1st time)
	ADCRn1 (2nd time)
	ADCRn2 (3rd time)
	ADCRn3 (4th time)

This mode is suitable for applications in which the average of the A/D conversion results is calculated.



**Figure 15-7 Example of 4-buffer mode operation (A/D trigger select: 4 buffers)**

1. The ADM*n*0.ADCEn bit is set to 1 (enable)
2. ANIn3 is A/D converted
3. The conversion result is stored in ADCRn0 register
4. ANIn3 is A/D converted
5. The conversion result is stored in ADCRn1 register
6. ANIn3 is A/D converted
7. The conversion result is stored in ADCRn2 register
8. ANIn3 is A/D converted

9. The conversion result is stored in ADCRn3 register
10. The INTAD interrupt is generated

### 15.5.2 Scan mode operation

In this mode, the analog inputs specified by the ADMn2 register are selected sequentially from the ANIn0 pin, and A/D conversion is executed. The A/D conversion results are stored in the ADCRnm register corresponding to the analog input.

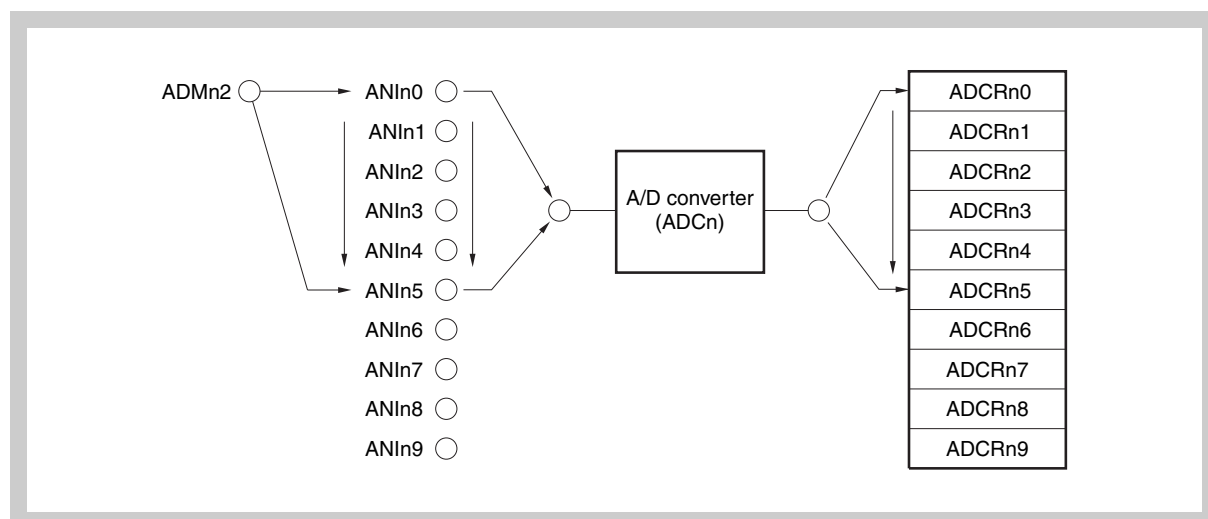
When conversion of all the specified analog input ends, the A/D conversion end interrupt (INTADn) is generated, and A/D conversion is stopped. The next conversion operation is repeated, unless the ADMn0.ADCEn bit is cleared to 0.

**Table 15-10 Correspondence between analog input pins and ADCRnm register (A/D trigger scan)**

Analog Input	A/D Conversion Result Register
ANIn0	ADCRn0
⋮	⋮
ANInm <sup>Note</sup>	ADCRnm

**Note** Set by the ADMn2.ANISn[3:0] bits.

This mode is most appropriate for applications in which multiple analog inputs are constantly monitored.



**Figure 15-8 Example of scan mode operation (A/D trigger scan)**

1. The ADMn0.ADCEn bit is set to 1 (enable)
2. ANIn0 is A/D converted
3. The conversion result is stored in ADCRn0
4. ANIn1 is A/D converted
5. The conversion result is stored in ADCRn1
6. ANIn2 is A/D converted
7. The conversion result is stored in ADCRn2
8. ANIn3 is A/D converted
9. The conversion result is stored in ADCRn3

10. ANIn4 is A/D converted
11. The conversion result is stored in ADCRn4
12. ANIn5 is A/D converted
13. The conversion result is stored in ADCRn5
14. The INTAD interrupt is generated

## 15.6 Operation in Timer Trigger Mode

In this mode, the conversion timing of the analog input signal set by the ANIn0 to ANIn9 pins is defined by a timer event signal.

The analog input conversion timing is generated when an A/D Converter trigger signal from the timers is generated.

When the ADMn0.ADCEn bit is set to 1, the A/D Converter waits for the timer signal and starts conversion when the timer event occurs (ADMn0.ADCSn = 1). When conversion is finished, the converter waits for a timer event signal again (ADMn0.ADCSn = 0).

If the timer event signal occurs during conversion, the conversion operation is executed from the beginning again.

If data is written to the ADMn0 to ADMn2 registers during conversion, the conversion operation is stopped and executed from the beginning again.

### 15.6.1 Select mode operation

In this mode, an analog input (ANIn0 to ANIn9) specified by the ADMn2 register is A/D converted. The conversion results are stored in the ADCRnm register corresponding to the analog input. In the select mode, the 1-buffer mode and 4-buffer mode are provided according to the storing method of the A/D conversion results.

#### (1) 1-buffer mode operation (timer trigger select: 1 buffer)

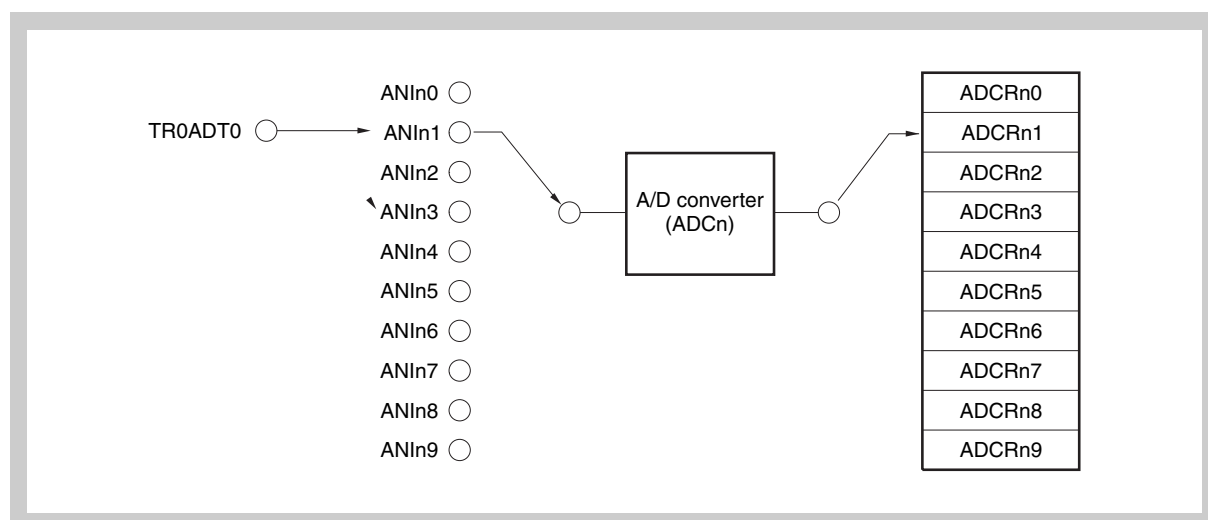
In this mode, one analog input is A/D converted once and the conversion results are stored in one ADCRnm register.

One analog input is A/D converted once using the trigger of the timer event signals and the results are stored in one ADCRnm register. An A/D conversion end interrupt (INTADn) is generated for each A/D conversion.

Unless the ADMn0.ADCEn bit is cleared to 0, A/D conversion is repeated each time a timer event signal is generated.

**Table 15-11 Correspondence between analog input pins and ADCRnm register (1-buffer mode (timer trigger select: 1 buffer))**

Trigger	Analog Input	A/D Conversion Result Register
Timer event signal	ANIn0	ADCRn0
	ANIn1	ADCRn1
	ANIn2	ADCRn2
	ANIn3	ADCRn3
	ANIn4	ADCRn4
	ANIn5	ADCRn5
	ANIn6	ADCRn6
	ANIn7	ADCRn7
	ANIn8	ADCRn8
	ANIn9	ADCRn9



**Figure 15-9 Example of 1-buffer mode operation (timer trigger select: 1 buffer) (ANIn1)**

1. The ADMn0.ADCEn is set to 1 (enable)
2. The TS0ADT0 signal is generated
3. ANIn1 is A/D converted
4. The conversion result is stored in ADCRn1
5. The INTADn interrupt is generated

**(2) 4-buffer mode operation (timer trigger select: 4 buffers)**

In this mode, A/D conversion of one analog input is executed four times, and the results are stored in the ADCRn<sub>m</sub> register.

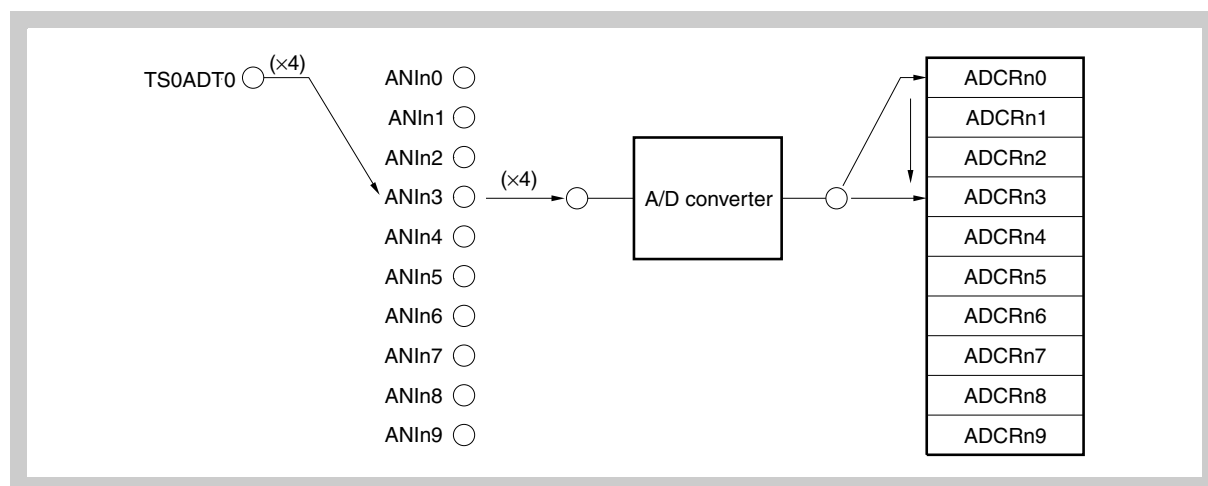
One analog input is A/D converted four times using the timer event signals as a trigger, and the results are stored in four ADCRn<sub>m</sub> registers. The A/D conversion end interrupt (INTADn) is generated when the four A/D conversions end.

After conversion has finished, the next conversion is repeated when a timer event signal is generated, unless the ADMn0.ADCEn bit is cleared to 0.

This mode is suitable for applications in which the average of the A/D conversion results is calculated.

**Table 15-12 Correspondence between analog input pins and ADCRn<sub>m</sub> register (4-buffer mode (timer trigger select: 4 buffers))**

Trigger	Analog Input	A/D Conversion Result Register
Timer event signal	ANI0 to ANI3	ADCRn0 (1st time)
		ADCRn1 (2nd time)
		ADCRn2 (3rd time)
		ADCRn3 (4th time)
	ANI4 to ANI7	ADCRn4 (1st time)
		ADCRn5 (2nd time)
		ADCRn6 (3rd time)
		ADCRn7 (4th time)
	ANIn8, ANIn9	ADCRn0 (1st time)
		ADCRn1 (2nd time)
		ADCRn2 (3rd time)
		ADCRn3 (4th time)



**Figure 15-10 Example of 4-buffer mode operation (timer trigger select: 4 buffers) (ANIn3)**

1. The ADMn0.ADCEn bit is set to 1 (enable)
2. The TS0ADT0 signal is generated
3. ANIn3 is A/D converted
4. The conversion result is stored in ADCR0

5. ANIn3 is A/D converted
6. The conversion result is stored in ADCR1
7. ANIn3 is A/D converted
8. The conversion result is stored in ADCR2
9. ANIn3 is A/D converted
10. The conversion result is stored in ADCR3
11. The INTADn interrupt is generated

### 15.6.2 Scan mode operation

In this mode, the analog inputs specified by the ADMn2 register are selected sequentially from the ANIn0 pin and are A/D converted the specified number of times using the timer event signal as a trigger.

The result of conversion is stored in the ADCRnm register corresponding to the analog input. When all the specified analog input signals have been converted, an A/D conversion end interrupt (INTADn) occurs.

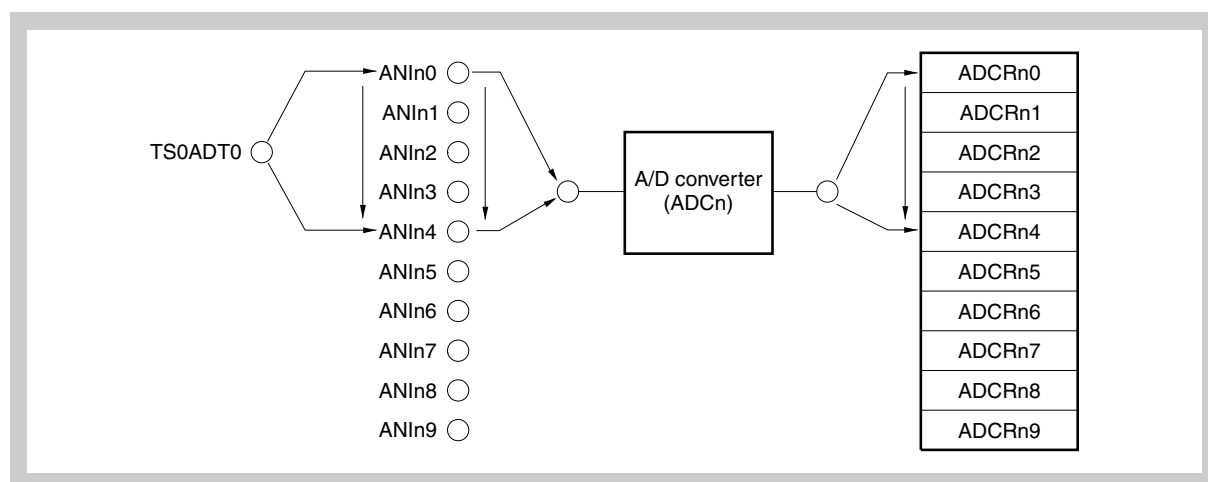
After conversion has finished, the A/D Converter waits for a trigger unless the ADMn0.ADCEn bit is cleared to 0. When a timer event occurs again, the converter starts A/D conversion again, starting from the ANIn0 input.

This mode is most appropriate for applications in which multiple analog inputs are constantly monitored.

**Table 15-13 Correspondence between analog input pins and ADCRnm register (scan mode (timer trigger scan))**

Trigger	Analog Input	A/D Conversion Result Register
Timer event signal	ANIn0	ADCRn0
	ANIn1	ADCRn1
	ANIn2	ADCRn2
	ANIn3	ADCRn3
	ANIn4	ADCRn4
	ANIn5	ADCRn5
	ANIn6	ADCRn6
	ANIn7	ADCRn7
	ANIn8	ADCRn8
	ANIn9	ADCRn9





**Figure 15-11 Example of scan mode operation (timer trigger scan) (ANIn0 to ANIn4)**

1. The ADMn0.ADCEn bit is set to 1 (enable)
2. The TS0ADT0 signal is generated
3. ANIn0 is A/D converted
4. The conversion result is stored in ADCRn0
5. ANIn1 is A/D converted
6. The conversion result is stored in ADCRn1
7. ANIn2 is A/D converted
8. The conversion result is stored in ADCRn2
9. ANIn3 is A/D converted
10. The conversion result is stored in ADCRn3
11. ANIn4 is A/D converted
12. The conversion result is stored in ADCRn4
13. The INTADn interrupt is generated

## 15.7 Operation in External Trigger Mode

In this mode, the conversion timing of the analog signals input to the ANIn0 to ANIn9 pins is specified by the ADTRGn pin.

Detection of the valid edge at the ADTRGn input pin is specified by using the EGAn1 and EGAn0 bits of the ADMn1 register.

When the ADMn0.ADCEn bit is set to 1, the A/D Converter waits for an external trigger (ADTRGn), and starts conversion when the valid edge of ADTRGn is detected (ADMn0.ADCSn = 1). When the converter has ended conversion, it waits for the external trigger again (ADMn0.ADCSn = 0).

If the valid edge is detected at the ADTRGn pin during conversion, conversion is executed from the beginning again.

If data is written to the ADMn0 to ADMn2 registers during conversion, conversion is stopped and executed from the beginning again.

### 15.7.1 Select mode operation

In this mode, one analog input (ANIn0 to ANIn9) specified by the ADMn2 register is A/D converted. The conversion results are stored in the ADCRnm register corresponding to the analog input. In the select mode, there are two select modes: 1-buffer mode and 4-buffer mode, according to the storing method of the conversion results.

#### (1) 1-buffer mode (external trigger select: 1 buffer)

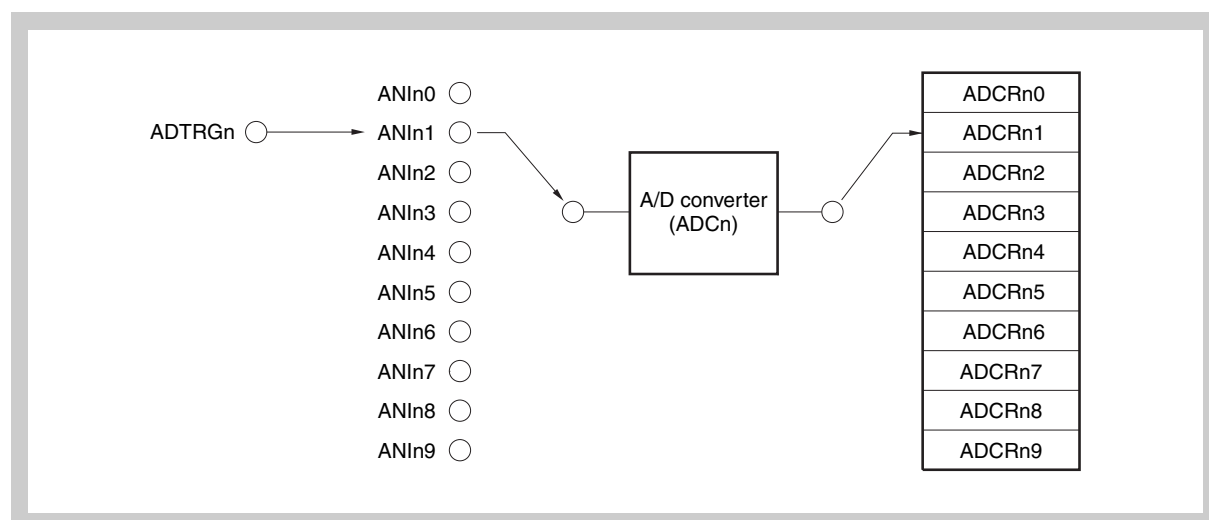
In this mode, one analog input is A/D converted using the ADTRGn signal as a trigger. The conversion results are stored in one ADCRnm register. The analog input and the A/D conversion results register correspond one to one. The A/D conversion end interrupt (INTADn) is generated for each A/D conversion, and A/D conversion is stopped.

**Table 15-14 Correspondence between analog input pins and ADCRnm register (external trigger select: 1 buffer)**

Trigger	Analog Input	A/D Conversion Result Register
ADTRGn signal	ANInm	ADCRnm

While the ADMn0.ADCEn = 1, A/D conversion is repeated every time a trigger is input from the ADTRGn pin.

This mode is most appropriate for applications in which the results are read after each A/D conversion.



**Figure 15-12 Example of 1-buffer mode operation (external trigger select: 1 buffer) (ANIn1)**

1. The ADMn0.ADCEn bit is set to 1 (enable)
2. The external trigger is generated
3. ANIn1 is A/D converted
4. The conversion result is stored in ADCRn1
5. The INTADn interrupt is generated

**(2) 4-buffer mode (external trigger select: 4 buffers)**

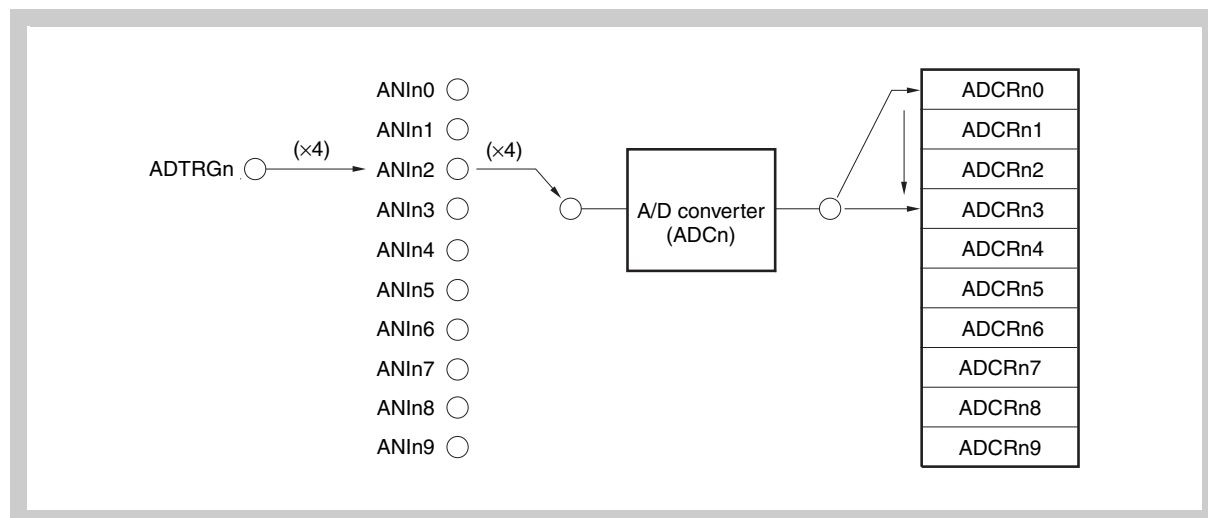
In this mode, one analog input is A/D converted four times using the ADTRGn signal as a trigger and the results are stored in the ADCRnm register. The A/D conversion end interrupt (INTADn) is generated and A/D conversion is stopped after the 4th A/D conversion.

**Table 15-15 Correspondence between analog input pins and ADCRnm register (external trigger select: 4 buffers)**

Trigger	Analog Input	A/D Conversion Result Register
ADTRGn signal	ANI0 to ANI3	ADCRn0 (1st time)
		ADCRn1 (2nd time)
		ADCRn2 (3rd time)
		ADCRn3 (4th time)
	ANI4 to ANI7	ADCRn4 (1st time)
		ADCRn5 (2nd time)
		ADCRn6 (3rd time)
		ADCRn7 (4th time)
	ANIn8, ANIn9	ADCRn0 (1st time)
		ADCRn1 (2nd time)
		ADCRn2 (3rd time)
		ADCRn3 (4th time)

While the ADCEn bit of the ADMn0 register is 1, A/D conversion is repeated every time a trigger is input from the ADTRGn pin.

This mode is suitable for applications in which the average of the A/D conversion results is calculated.



**Figure 15-13 Example of 4-buffer mode operation (external trigger select: 4 buffers) (ANIn2)**

1. The ADMn0.ADCEn bit is set to 1 (enable)
2. The external trigger is generated
3. ANIn2 is A/D converted
4. The conversion result is stored in ADCRn0
5. ANIn2 is A/D converted
6. The conversion result is stored in ADCRn1
7. ANIn2 is A/D converted
8. The conversion result is stored in ADCRn2
9. ANIn2 is A/D converted
10. The conversion result is stored in ADCRn3
11. The INTAD interrupt is generated

## 15.7.2 Scan mode operation

In this mode, the analog inputs specified by the ADMn2 register are selected sequentially from the ANIn0 pin using the ADTRGn signal as a trigger, and A/D converted. The A/D conversion results are stored in the ADCRnm register corresponding to the analog input ANInm.

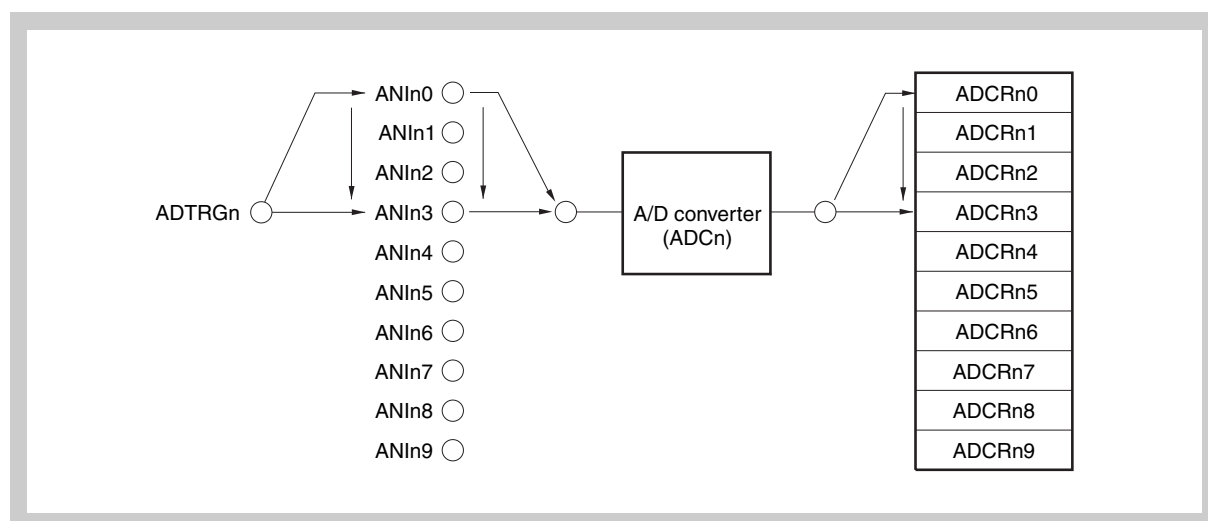
When conversion of all the specified analog inputs has ended, the A/D conversion end interrupt (INTADn) is generated. Unless the ADMn0.ADCE bit is cleared to 0 after end of conversion, the A/D Converter waits for a trigger. The converter starts A/D conversion from the ANIn0 input when a trigger is input to the ADTRGn pin again.

When a trigger is input to the ADTRGn pin while ADMn0.ADCEn = 1, A/D conversion is started again.

This is most appropriate for applications in which multiple analog inputs are constantly monitored.

**Table 15-16 Correspondence between analog input pins and ADCRnm register (external trigger scan)**

Trigger	Analog Input	A/D Conversion Result Register
ADTRGn signal	ANIn0	ADCRn0
	ANIn1	ADCRn1
	ANIn2	ADCRn2
	ANIn3	ADCRn3
	ANIn4	ADCRn4
	ANIn5	ADCRn5
	ANIn6	ADCRn6
	ANIn7	ADCRn7
	ANIn8	ADCRn8
	ANIn9	ADCRn9

**Figure 15-14 Example of scan mode operation (external trigger scan) (ANIn0 to ANIn3)**

1. The ADMn0.ADCEn bit is set to 1 (enable)
2. The external trigger is generated
3. ANIn0 is A/D converted
4. The conversion result is stored in ADCRn0
5. ANIn1 is A/D converted
6. The conversion result is stored in ADCRn1
7. ANIn2 is A/D converted
8. The conversion result is stored in ADCRn2
9. ANIn3 is A/D converted
10. The conversion result is stored in ADCRn3
11. The INTADn interrupt is generated

## 15.8 Precautions

### (1) Stopping conversion operation

When the ADMn0.ADCEn bit is cleared to 0 during a conversion operation, the conversion operation stops and the conversion results are not stored in the ADCRnm register.

### (2) External/timer trigger interval

Set the interval (input time interval) of the trigger in the external or timer trigger mode longer than the conversion time specified by the ADMn1.FRnm bits and the number of conversion channels specified by the ADMn2.ANISn[3:0] bits:

$$0 < \text{interval} \leq (\text{ANISn}[3:0] + 1) * \text{Conversion Operation Time (FRn1, FRn0)}$$

When the following external trigger or timer trigger is input during a conversion operation, the conversion operation is aborted and the conversion starts according to the last external trigger input or timer trigger input.

When conversion operations are aborted, the conversion results are not stored in the ADCRnm register. However, the number of times the trigger has been input is counted. When an interrupt occurs, the values that have been converted are stored in the ADCRnm register.

### (3) Operation in HALT mode

A/D conversion continues in the HALT mode. When this mode is released by NMI input or unmasked maskable interrupt input, the ADMn0, ADMn1, and ADMn2 registers as well as the ADCRnm register hold the value.

### (4) Input range of ANIn0 to ANIn9

Use the input voltage at ANIn0 to ANIn9 within the specified range. If a voltage outside the range of  $AV_{REF}$  is input to any of these pins (even within the absolute maximum rating range), the converted value of the channel is undefined. In addition, the converted value of the other channels may also be affected.

### (5) Conflicts

- **Conflict between writing A/D conversion result registers (ADCRnm, ADCRnmH) at end of conversion and reading ADCRnm and ADCRnmH registers by instruction**

Reading the ADCRnm and ADCRnmH registers takes precedence. After these registers have been read, the new conversion result is written to the ADCRnm and ADCRnmH registers.

- **Conflict between writing ADCRnm and ADCRnmH at end of conversion and input of external trigger signal**

The external trigger signal is not accepted during A/D conversion. Therefore, it is not accepted while ADCRnm and ADCRnmH are being written.

- **Conflict between writing ADCRnm and ADCRnmH at end of conversion and writing ADMn1 or ADMn2 register**

If ADMn1 or ADMn2 register is written immediately after ADCRnm and ADCRnmH have been written on completion of A/D conversion, the conversion result is written to the ADCRnm and ADCRnmH registers, but the A/D conversion end interrupt (INTADn) may not occur depending on the timing.

#### (6) Unexpected AD conversion start in timer trigger mode/external trigger mode

**Description** If the ADC is operated in timer trigger mode (ADMn1.TRGn[1:0] = 01<sub>B</sub>) or in external trigger mode (ADMn1.TRGn[1:0] = 10<sub>B</sub>) the ADC may start conversion autonomously on changing the AD conversion enable bit (ADMn0.ADCEn) from 0 to 1 before occurrence of the first configured trigger event (timer trigger or external trigger).

**Workaround** Whenever changing the ADC enable bit (ADMn0.ADCEn) from 0 to 1 write two times in sequence 1 to the ADC enable bit.

Make sure that no maskable or non-maskable interrupt may occur between the 1st and 2nd write instruction. Use the processor status word to temporarily disable non-maskable interrupts (PSW.NP = 1) or non-maskable interrupts (PSW.ID = 1) if interrupt occurrence cannot be excluded.

- Save PSW (optional)
- Disable INT and/or NMI (optional)
- 1st write 1 to ADMn0.ADCEn (byte or bit access)
- 2nd write 1 to ADMn0.ADCEn (byte or bit access)
- Restore PSW (optional)

#### Example (C - GHS Compiler):

```
unsigned int tmp;

tmp=__GETSR();           // Save PSW
__SETSR(0x000000a0|tmp); // Disable NMI and INT in PSW
ADM00 = 0x80;           // 1st write to ADMn0
ADM00 = 0x80;           // 2nd write to ADMn0
__SETSR(tmp);           // Restore PSW
```

#### Example (ASM):

```
stsr PSW, r14           -- Save PSW
mov 0x000000a0, r13
or r14, r13
ldsr r13, PSW           -- Disable NMI and INT in PSW
setl 7, ADM00[r0]       -- 1st write to ADMn0
setl 7, ADM00[r0]       -- 2nd write to ADMn0
ldsr r14, PSW           -- Restore PSW
```

**Note** This workaround is effective only in case code is executed from internal flash memory. While debugging do not set a breakpoint on the 2nd write instruction to ADMn0.

**(7) Restart of conversion operation**

When A/D conversion is stopped (aborted) and restarted by

- clearing and setting the ADCEn bit (1 → 0 → 1)
- rewriting one of the ADMn0, ADMn1, or ADMn2 registers (with same value)

a ADCRnm register should be read not before the corresponding channel has been converted the first time.

The end of conversion of all channels is indicated by the occurrence of INTADn interrupt. Therefore read the ADCRnm registers within an interrupt service routine (ISR), or if no interrupt is serviced, check the conversion end by checking the indicating IF flag of the corresponding interrupt control register (ADICn).

A DMA transfer of the conversion results can be executed at any time and does not need to observe the end of conversion.

**Example (when no ISR is applied):**

- Initialization/ Reconfiguration

```
...
// stop ADC
ADMn0.ADCEn = 0;
// reconfigure ADC
...
// clear request flag & start ADC
ADICn.IF = 0;
ADMn0.ADCEn = 1;
...
```

- Asynchronous read

```
...
// check request flag
// (or wait until it is set)
if (ADICn.IF == 1)
{
    // read ADCRnm registers
    ...
}
...
```



## Chapter 16 16-Bit Timer/Event Counter AA (TAA)

The microcontroller has ten instances of this 16-bit timer/event counter AA, TAA0 to TAA9. The timer is upward compatible to Timer P used in various other devices of the V850E and the V850ES family, including V850E/PHC3. It offers new additional features for enhanced output control and for 32-bit capture.

Throughout this chapter, the individual instances of Timer AA are identified by “n” (n = 0 to 9), for example, TAA<sub>n</sub>CTL0 for the TAA<sub>n</sub> control register 0.

### 16.1 Features

Timer AA (TAA) is a 16-bit timer/event counter provided with general-purpose functions.

TAA can perform the following operations:

- 16-bit-accuracy PWM output timer
- Interval timer
- External event counter function
- Timer synchronised operation function
- One-shot pulse output
- Pulse interval and frequency measurement counter
- Free running function
- External trigger pulse output function

### 16.2 Function Outline

- Capture trigger input signal × 2
- External trigger input signal × 1
- Clock select × 8
- External event count input × 1
- Readable counter × 1
- Capture/compare reload register × 2
- Capture/compare match interrupt × 2
- Timer output (TOAA<sub>n</sub>0, TOAA<sub>n</sub>1) × 2
- 32-bit capture by cascading two timer AA (TAA0+TAA<sub>x</sub>, TAA4+TAA<sub>y</sub>, TAA8+TAA9).

**Note** x = 1 to 3 and y = 5 to 7

## 16.3 Configuration

TAA includes the following hardware.

**Table 16-1** Timer TAA registers and external connections

Item	Configuration
Timer register	16-bit counter
Registers	<ul style="list-style-type: none"> <li>TAA<sub>n</sub> timer capture/compare registers 0, 1 (TAA<sub>n</sub>CCR0, TAA<sub>n</sub>CCR1)</li> <li>TAA<sub>n</sub> timer read buffer register (TAA<sub>n</sub>CNT)</li> <li>CCR0 buffer register, CCR1 buffer register</li> </ul>
Input selection registers	<ul style="list-style-type: none"> <li>TAA<sub>n</sub> input selection registers (TAAIC0 to TAAIC2)</li> </ul>
Timer output	TOAA <sub>n</sub> 0, TOAA <sub>n</sub> 1
Timer input	TIAA <sub>n</sub> 0, TIAA <sub>n</sub> 1, TTRGAAn, TEVTAA <sub>n</sub>
Control registers	<ul style="list-style-type: none"> <li>TAA<sub>n</sub> control registers 0, 1 (TAA<sub>n</sub>CTL0, TAA<sub>n</sub>CTL1)</li> <li>TAA<sub>n</sub> I/O control registers 0 to 3 (TAA<sub>n</sub>IOC0, TAA<sub>n</sub>IOC2, TAA<sub>n</sub>IOC4)</li> <li>TAA<sub>n</sub> option registers 0, 1 (TAA<sub>n</sub>OPT0, TAA<sub>n</sub>OPT1)</li> <li>TAA<sub>n</sub> Specific overflow value control register (TAA<sub>n</sub>SVC)</li> </ul>

**Note** n = 0 to 9

Timer AA (TAA) pins are alternate function of port pins. For how to set the alternate function, refer to the description of the registers in “Pin Functions” on page 23.

The block diagram of the timer TAA is shown below. *Figure 16-2* and *Figure 16-3* show the block diagrams of the input circuits of the different timers TAA<sub>n</sub>.

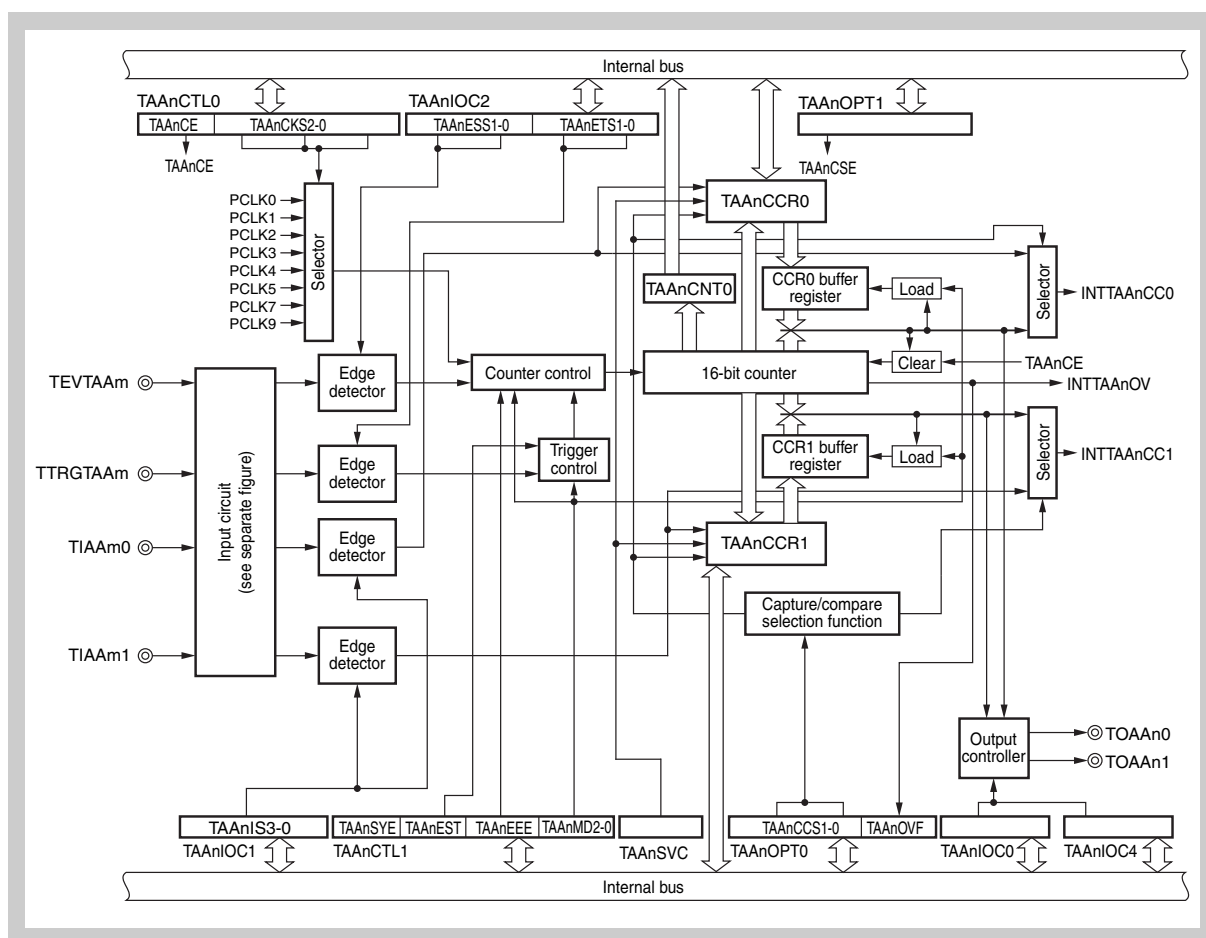


Figure 16-1 Block Diagram of Timer AA

- Note**
1.  $n = 0$  to 9
  2. PCLKn: peripheral clocks (refer to "Clock Generator" on page 148):

PCLK	0	1	2	3	4	5	7	9
f [MHz]	32	16	8	4	2	1	0.25	0.0625

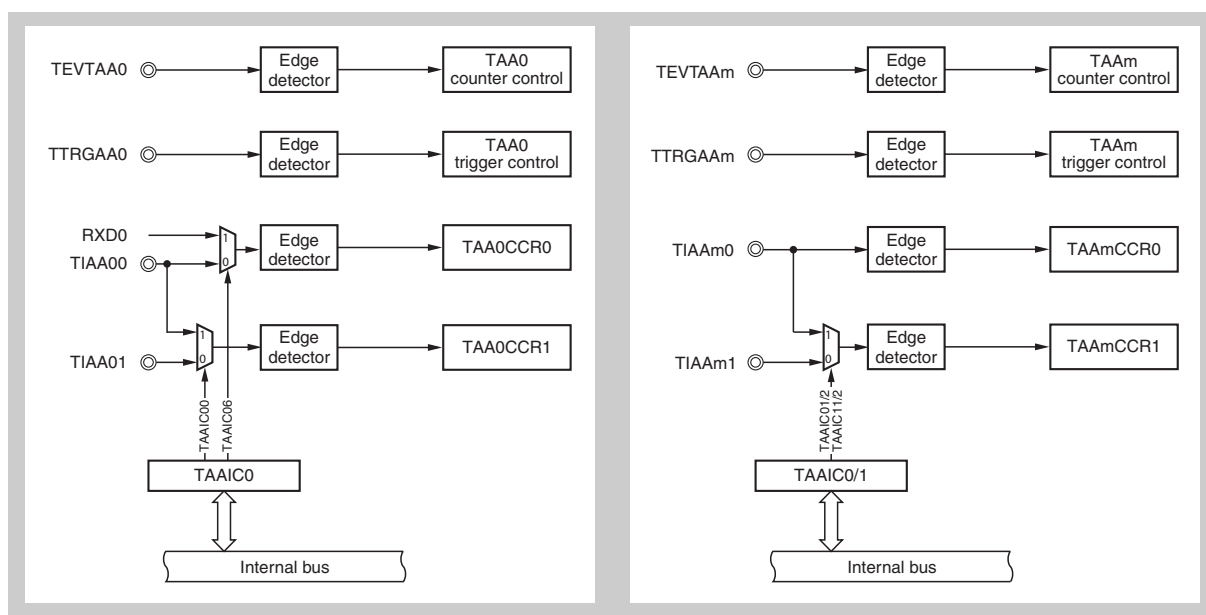


Figure 16-2 Input circuit of TAA0 (left) and TAAm with m = 1, 2, 5, 6 (right)

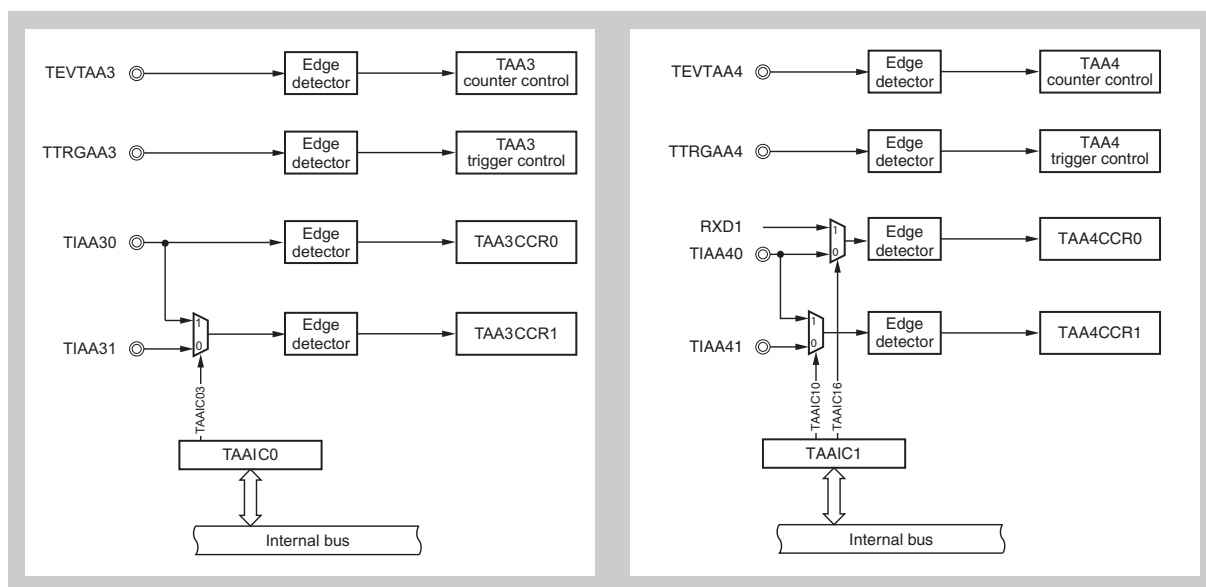


Figure 16-3 Input circuit of TAA3 (left) and TAA4 (right)

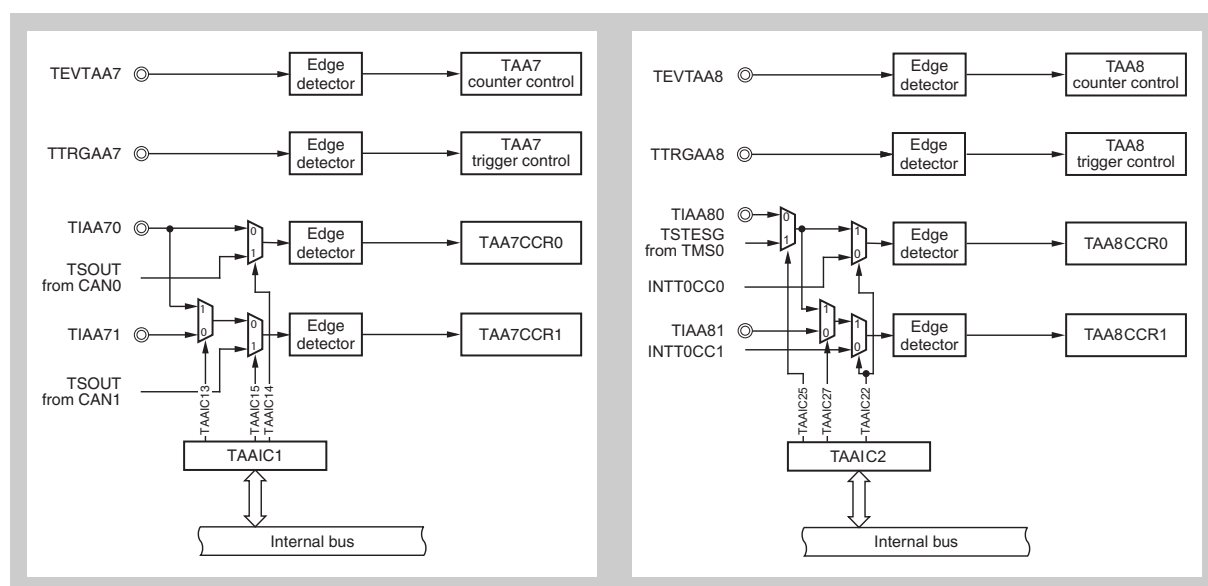


Figure 16-4 Input circuit of TAA7 (left) and TAA8 (right)

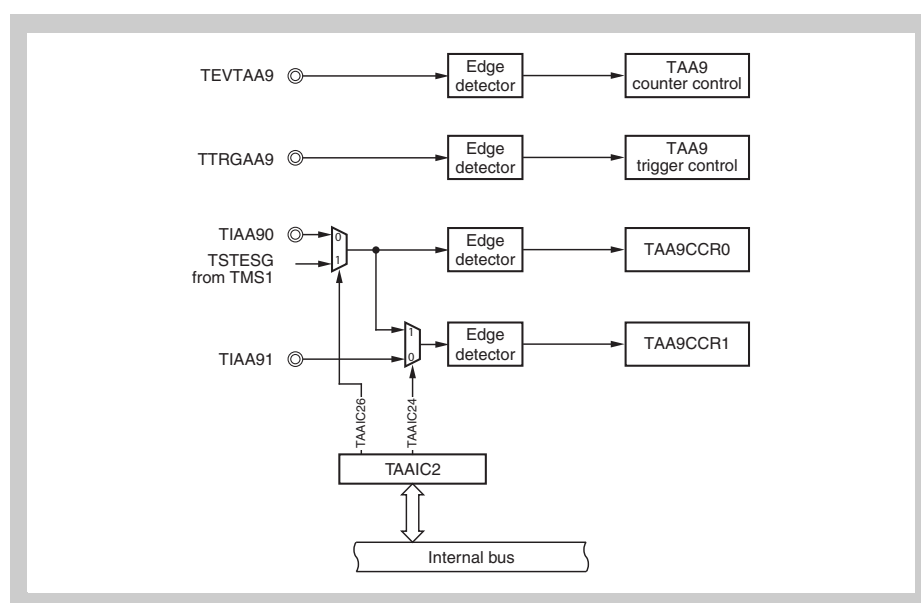


Figure 16-5 Input circuit of TAA9

**(1) TAAAnCCR0 - TAA capture/compare register 0**

The TAAAnCCR0 register is a 16-bit register that operates either as capture register or as a compare register.

In free-running mode, this register can be used as a capture register or as a compare register specified by bit TAAAnOPT0.TAAAnCCS0.

In the pulse width measurement mode, this register can be used only as a capture register (the compare function cannot be used.)

In all modes other than free-running mode and pulse width measurement mode, this register is used as a compare register.

After reset the TAAAnCCR0 register default status is compare register.

**Access** This register can be read/written in 16-bit units.

**Address**

TAA0CCR0: FFFFF606 <sub>H</sub>	TAA1CCR0: FFFFF616 <sub>H</sub>
TAA2CCR0: FFFFF626 <sub>H</sub>	TAA3CCR0: FFFFF636 <sub>H</sub>
TAA4CCR0: FFFFF646 <sub>H</sub>	TAA5CCR0: FFFFF656 <sub>H</sub>
TAA6CCR0: FFFFF666 <sub>H</sub>	TAA7CCR0: FFFFF676 <sub>H</sub>
TAA8CCR0: FFFFF686 <sub>H</sub>	TAA9CCR0: FFFFF6B6 <sub>H</sub>

**Initial Value** 0000<sub>H</sub>. This registers is cleared by any reset, or if the internal operation clock is disabled by TAAAnCTL0.TAAAnCE = 0.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>TAAAnCCR0</b>	Compare value 0															
	R/W															

**Caution** When external event counter mode is used, do not set TAAAnCCR0 register to 0000<sub>H</sub>.

**Use as compare register** When used as a compare register, TAAAnCCR0 can be rewritten when TAAAnCE = 1, as shown below:

TAA operation mode	TAAAnCCR0 register writing method
PWM mode, external trigger pulse output mode	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Any time write
Pulse width measurement mode	Not applicable (used as capture register)

**Use as capture register** When used as capture register, the count value is stored in TAAAnCCR0 upon capture trigger (TIAAn0) input edge detection.

**Note** The value of TAAAnCCR0 register is read when TAAAnCE bit of TAAAn control register 0 (TAAAnCTL0) equals 1.

**(2) TAAAnCCR1 - TAA capture/compare register 1**

The TAAAnCCR1 register is a 16-bit register that operates either both as a capture register or as a compare register.

In free-running mode, this register can be used as a capture register or as a compare register specified by bit TAAAnOPT0.TAAAnCCS1.

In the pulse width measurement mode, this register can be used only as a capture register (the compare function cannot be used.)

In all modes other than free-running mode and pulse width measurement mode, this register is used as a compare register.

After reset the TAAAnCCR1 register default status is compare register.

**Access** This register can be read/written in 16-bit units.

**Address**

TAA0CCR1: FFFFF608 <sub>H</sub>	TAA1CCR1: FFFFF618 <sub>H</sub>
TAA2CCR1: FFFFF628 <sub>H</sub>	TAA3CCR1: FFFFF638 <sub>H</sub>
TAA4CCR1: FFFFF648 <sub>H</sub>	TAA5CCR1: FFFFF658 <sub>H</sub>
TAA6CCR1: FFFFF668 <sub>H</sub>	TAA7CCR1: FFFFF678 <sub>H</sub>
TAA8CCR1: FFFFF688 <sub>H</sub>	TAA9CCR1: FFFFF6B8 <sub>H</sub>

**Initial Value** 0000<sub>H</sub>. This register is cleared by any reset, or if the internal operation clock is disabled by TAAAnCTL0.TAAAnCE = 0.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>TAAAnCCR1</b>	Compare value 0															
	R/W															

**Caution** When external event counter mode is used, do not set TAAAnCCR1 register to 0000<sub>H</sub>.

**Use as compare register** When used as a compare register TAAAnCCR1 can be rewritten when TAAAnCE = 1, as below mentioned.

TAA operation mode	TAAAnCCR1 register writing method
PWM mode, external trigger pulse output mode	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Any time write
Pulse width measurement mode	Not applicable (used as capture register)

**Use as capture register** When used as a capture register the count value is stored in TAAAnCCR1 upon capture trigger (TIAAn1) input edge detection.

**Note** The value of TAAAnCCR1 register is read when TAAAnCE bit of TAAAn control register 0 (TAAAnCTL0) equals 1.

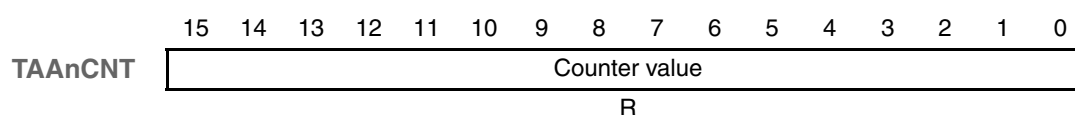
**(3) TAA<sub>n</sub>CNT - TAA counter read buffer register**

TAA<sub>n</sub>CNT register is a read buffer register that can read 16-bit counter values.

**Access** This register can be read only in 16-bit units.

<b>Address</b>	TAA0CNT: FFFFF60A <sub>H</sub>	TAA1CNT: FFFFF61A <sub>H</sub>
	TAA2CNT: FFFFF62A <sub>H</sub>	TAA3CNT: FFFFF63A <sub>H</sub>
	TAA4CNT: FFFFF64A <sub>H</sub>	TAA5CNT: FFFFF65A <sub>H</sub>
	TAA6CNT: FFFFF66A <sub>H</sub>	TAA7CNT: FFFFF67A <sub>H</sub>
	TAA8CNT: FFFFF68A <sub>H</sub>	TAA9CNT: FFFFF6BA <sub>H</sub>

**Initial Value** 0000<sub>H</sub>. This register is cleared by any reset.



**Note** 0000H is read from this register, when TAA<sub>n</sub>CTL0.TAA<sub>n</sub>CE = 0. The current counter value is read when TAA<sub>n</sub>CE bit = 1.



## 16.4 Input Selection Registers

### (1) TAAIC0 - Timer input control register 0

The TAAIC0 register is an 8-bit register that controls the external input pin source of the capture register 1 of TAA0 to TAA3.

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

**Address** FFFF6F0<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>TAAIC0</b>	0	TAAIC06	0	0	TAAIC03	TAAIC02	TAAIC01	TAAIC00
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Caution** Do not change the timer input when the timer is in operation.

**Table 16-2 TAAIC0 register contents**

Bit position	Bit name	Function
6	TAAIC06	Selects the TAA0CCR0 register capture source input. 0: Capture source input is pin P10/TIAA00/TEVTAA1 1: Capture source input is pin P16/TIAA30/TTRGAA2
3	TAAIC03	Selects the TAA3CCR1 register capture source input. 0: Capture source input is pin P17/TIAA31/TEVTAA2 1: Capture source input is pin P16/TIAA30/TTRGAA2
2	TAAIC02	Selects the TAA1CCR1 register capture source input. 0: Capture source input is pin P15/TIAA21/TTRGAA3 1: Capture source input is pin P14/TIAA20/TEVTAA3
1	TAAIC01	Selects the TAA1CCR1 register capture source input. 0: Capture source input is pin P13/TIAA11/TEVTAA0 1: Capture source input is pin P12/TIAA10/TTRGAA0
0	TAAIC00	Selects the TAA0CCR1 register capture source input. 0: Capture source input is pin P11/TIAA01/TTRGAA1 1: Capture source input is pin P10/TIAA00/TEVTAA1

**(2) TAAIC1 - Timer input control register 1**

The TAAIC1 register is an 8-bit register that controls the external input pin source of the capture register 1 of TAA4 to TAA7, as well as the internal time trigger source from the CAN Controllers to both capture registers 0 and 1 of TAA7.

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

**Address** FFFFF6F1<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>TAAIC1</b>	0	TAAIC16	TAAIC15	TAAIC14	TAAIC13	TAAIC12	TAAIC11	TAAIC10
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Caution** Do not change the timer input when the timer is in operation.

**Table 16-3 TAAIC0 register contents**

Bit position	Bit name	Function															
6	TAAIC16	Selects the TAA4CCR0 register capture source input. 0: Capture source input is pin P20/TIAA40/TEVTAA5 1: Capture source input is RXDC															
5, 3	TAAIC15 TAAIC13	Selects the TAA7CCR1 register capture source input <table border="1"> <tr> <th>TAAIC15</th><th>TAAIC13</th><th>TAA7CCR1 Register Capture Source Input Selection</th></tr> <tr> <td>0</td><td>0</td><td>Capture source input is pin P27/TIAA71/ TEVTAA6</td></tr> <tr> <td>0</td><td>1</td><td>Capture source input is pin P26/TIAA70/ TTRGAA6</td></tr> <tr> <td>1</td><td>0</td><td>Capture source input is CAN1 time trigger (TSOUT1)</td></tr> <tr> <td>1</td><td>1</td><td></td></tr> </table>	TAAIC15	TAAIC13	TAA7CCR1 Register Capture Source Input Selection	0	0	Capture source input is pin P27/TIAA71/ TEVTAA6	0	1	Capture source input is pin P26/TIAA70/ TTRGAA6	1	0	Capture source input is CAN1 time trigger (TSOUT1)	1	1	
TAAIC15	TAAIC13	TAA7CCR1 Register Capture Source Input Selection															
0	0	Capture source input is pin P27/TIAA71/ TEVTAA6															
0	1	Capture source input is pin P26/TIAA70/ TTRGAA6															
1	0	Capture source input is CAN1 time trigger (TSOUT1)															
1	1																
4	TAAIC14	Selects the TAA7CCR0 register capture source input. 0: Capture source input is pin P26/TIAA70/ TTRGAA6 1: Capture source input is CAN0 time trigger (TSOUT0)															
2	TAAIC12	Selects the TAA6CCR1 register capture source input. 0: Capture source input is pin P25/TIAA61/TTRGAA7 1: Capture source input is pin P24/TIAA60/TEVTAA7															
1	TAAIC11	Selects the TAA5CCR1 register capture source input. 0: Capture source input is pin P23/TIAA51/TEVTAA4 1: Capture source input is pin P22/TIAA50/TTRGAA4															
0	TAAIC10	Selects the TAA4CCR1 register capture source input. 0: Capture source input is pin P21/TIAA41/TTRGAA5 1: Capture source input is pin P20/TIAA40/TEVTAA5															

**(3) TAAIC2 - Timer input control register 2**

The TAAIC2 register is an 8-bit register that controls the external input pin source of the capture register 1 of TMT0 and TMT1, as well as the internal source of both capture registers 0 and 1 of TAA8 and TAA9.

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

**Address** FFFFF6F2<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>TAAIC2</b>	TAAIC27	TAAIC26	TAAIC25	TAAIC24	0	TAAIC22	TAAIC21	TAAIC20
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Caution** Do not change the timer input when the timer is in operation.

**Table 16-4 TAAIC0 register contents**

Bit position	Bit name	Function				
7, 5, 2	TAAIC27 TAAIC25 TAAIC22	Selects the TAA8CCR0 and TAA8CCR1 registers capture source input				
		TAAIC22	TAAIC25	TAAIC27	Capture Source Input Selection of	
					TAA8CCR0	TAA8CCR1
		0	X	X	INTT0CC0 signal from TMT0	INTT0CC1 signal from TMT0
		1	0	0	Pin P100/TIAA80/TTRGAA8	Pin P101/TIAA81/TEVTAA8
			0	1	Pin P100/TIAA80/T TRGAA8	
			1	0	TSTESG signal from TMS0	Pin P101/TIAA81/TEVTAA8
1	1		TSTESG signal from TMS0			
6, 4	TAAIC26 TAAIC24	Selects the TAA9CCR0 and TAA9CCR1 registers capture source input				
		TAAIC26	TAAIC24	Capture Source Input Selection of		
				TAA9CCR0	TAA9CCR1	
		0	0	Pin P102/TIAA90/TTRGAA9	Pin P103/TIAA91/TEVTAA9	
		0	1		Pin P102/TIAA90/TTRGAA9	
		1	0	TSTESG signal from TMS1	Pin P103/TIAA91/TEVTAA9	
		1	1		TSTESG signal from TMS1	
1	TAAIC21	Selects the TT1CCR1 register capture source input. 0: Capture source input is pin P74/TIT11/TEVTT0 1: Capture source input is pin P73/TIT10/TTRGT0				
0	TAAIC20	Selects the TT0CCR1 register capture source input. 0: Capture source input is pin P71/TIT01/TTRGT1 1: Capture source input is pin P70/TIT00/TEVTT1				

## 16.5 Control Registers

### (1) TAA<sub>n</sub>CTL0 - TAA control register 0

TAA<sub>n</sub> control register 0 is an 8-bit register that controls the operation of timer AA.

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

**Address**

TAA0CTL0: FFFFF600 <sub>H</sub>	TAA1CTL0: FFFFF610 <sub>H</sub>
TAA2CTL0: FFFFF620 <sub>H</sub>	TAA3CTL0: FFFFF630 <sub>H</sub>
TAA4CTL0: FFFFF640 <sub>H</sub>	TAA5CTL0: FFFFF650 <sub>H</sub>
TAA6CTL0: FFFFF660 <sub>H</sub>	TAA6CTL0: FFFFF670 <sub>H</sub>
TAA8CTL0: FFFFF680 <sub>H</sub>	TAA7CTL0: FFFFF6B0 <sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>TAA<sub>n</sub>CTL0</b>	TAA <sub>n</sub> CE	0	0	0	0	TAA <sub>n</sub> CKS2	TAA <sub>n</sub> CKS1	TAA <sub>n</sub> CKS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Caution** Set bits TAA<sub>n</sub>CKS2 to TAA<sub>n</sub>CKS0 only when TAA<sub>n</sub>CE = 0.  
 When TAA<sub>n</sub>CE bit setting is changed from 0 to 1, TAA<sub>n</sub>CKS2 to TAA<sub>n</sub>CKS0 bits can be set simultaneously.  
 The same value can always be written to the TAA<sub>n</sub>CTL0 register by software.

**Table 16-5 TAA<sub>n</sub>CTL0 register contents**

Bit position	Bit name	Function																																				
7	TAAAnCE	<p>Controls the timer TAAAn operation.</p> <p>0: Internal operating clock operation disabled (TAAAn is asynchronously reseted)</p> <p>1: Internal operating clock operation enabled</p> <p>Internal operating clock control and TAAAn asynchronous reset are performed with the TAAAnCE bit. When TAAAnCE bit is cleared to 0, the internal operating clock of TAAAn stops (fixed to low level) and TAAAn is reset asynchronously.</p> <p>When the TAAAnCE bit is set to 1, the internal operating clock is enabled within 2 input clocks, and TAAAn counts up.</p>																																				
2 to 0	TAAAnCKS [2:0]	<p>Selects the count clock of timer TAAAn.</p> <table><tr><th>TAAAnCKS2</th><th>TAAAnCKS1</th><th>TAAAnCKS0</th><th>Internal count clock selection</th></tr><tr><td>0</td><td>0</td><td>0</td><td>PCLK0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>PCLK1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>PCLK2</td></tr><tr><td>0</td><td>1</td><td>1</td><td>PCLK3</td></tr><tr><td>1</td><td>0</td><td>0</td><td>PCLK4</td></tr><tr><td>1</td><td>0</td><td>1</td><td>PCLK5</td></tr><tr><td>1</td><td>1</td><td>0</td><td>PCLK7</td></tr><tr><td>1</td><td>1</td><td>1</td><td>PCLK9</td></tr></table>	TAAAnCKS2	TAAAnCKS1	TAAAnCKS0	Internal count clock selection	0	0	0	PCLK0	0	0	1	PCLK1	0	1	0	PCLK2	0	1	1	PCLK3	1	0	0	PCLK4	1	0	1	PCLK5	1	1	0	PCLK7	1	1	1	PCLK9
TAAAnCKS2	TAAAnCKS1	TAAAnCKS0	Internal count clock selection																																			
0	0	0	PCLK0																																			
0	0	1	PCLK1																																			
0	1	0	PCLK2																																			
0	1	1	PCLK3																																			
1	0	0	PCLK4																																			
1	0	1	PCLK5																																			
1	1	0	PCLK7																																			
1	1	1	PCLK9																																			

**(2) TAACTL1 - TAA timer control register 1**

TAA control register 1 is an 8-bit register that controls the operation of timer AA.

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

**Address**

TAA0CTL1: FFFFF601 <sub>H</sub>	TAA1CTL1: FFFFF611 <sub>H</sub>
TAA2CTL1: FFFFF621 <sub>H</sub>	TAA3CTL1: FFFFF631 <sub>H</sub>
TAA4CTL1: FFFFF641 <sub>H</sub>	TAA5CTL1: FFFFF651 <sub>H</sub>
TAA6CTL1: FFFFF661 <sub>H</sub>	TAA6CTL1: FFFFF671 <sub>H</sub>
TAA8CTL1: FFFFF681 <sub>H</sub>	TAA7CTL1: FFFFF6B1 <sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>TAACTL1</b>	TAA <sub>SYE</sub>	TAA <sub>EST</sub>	TAA <sub>EEE</sub>	0	0	TAA <sub>MD2</sub>	TAA <sub>MD1</sub>	TAA <sub>MD0</sub>
	R/W	R/W	R/W	R	R	R/W	R/W	R/W

**Caution** Set bits TAA<sub>EEE</sub> and TAA<sub>MD</sub>[2:0] when TAACTL0.TAA<sub>CE</sub> = 0. The operation is not guaranteed if rewriting is performed when TAA<sub>CE</sub> = 1. (Despite of this the same value can be written when TAA<sub>CE</sub> = 1.) If rewriting was mistakenly performed, clear TAA<sub>CE</sub> to 0 and then set the bits again.

**Table 16-6 TAACTL1 register contents (1/2)**

Bit position	Bit name	Function												
7	TAA <sub>n</sub> SYE	<p>Controls the tuned operation mode of timer TAA<sub>n</sub>.</p> <p>0: Independent operation mode (asynchronous operation mode)</p> <p>1: Tuned operation mode (specification of slave operation)</p> <p>In this mode, timer TAA<sub>n</sub> can operate in synchronization with a master timer.</p> <table><tr><th>Master timer</th><th colspan="3">Slave timer</th></tr><tr><td>TAA0</td><td>TAA1</td><td>TAA2</td><td>TAA3</td></tr><tr><td>TAA4</td><td>TAA5</td><td>TAA6</td><td>TAA7</td></tr></table> <p>For the tuned operation mode, refer to “Timer Synchronization Operation Function” on page 611.</p> <hr/> <p><b>Caution:</b></p> <ol style="list-style-type: none"><li>1. Be sure to clear the TAA0SYE and TAA4SYE bits to 0 (master timer) and set the TAA<sub>m</sub>SYE bits (m = 1 to 3, 5 to 7) to 1 respectively when used as slave timers.</li><li>2. In the synchronous operation mode, the master timer can be used only in the PWM mode, external trigger pulse output mode, pulse output mode, and free-running mode. The slave timers can be used in the free-running mode only. Setting the external event count mode, one-shot pulse mode, and pulse width measurement mode is prohibited.</li></ol>	Master timer	Slave timer			TAA0	TAA1	TAA2	TAA3	TAA4	TAA5	TAA6	TAA7
Master timer	Slave timer													
TAA0	TAA1	TAA2	TAA3											
TAA4	TAA5	TAA6	TAA7											

**Table 16-6 TAAAnCTL1 register contents (2/2)**

Bit position	Bit name	Function																																				
6	TAAAnEST	<p>Controls the software trigger of timer TAAAn.</p> <p>0: No operation</p> <p>1: In one-shot pulse mode: One-shot pulse software trigger</p> <p>In external trigger pulse output mode: Pulse output software trigger</p> <p>The TAAAnEST bit functions as a software trigger in the one-shot pulse mode or external trigger pulse output mode (this bit is invalid in any other mode). By setting TAAAnEST to 1 when TAAAnCE = 1, a software trigger is issued. Therefore, be sure to set TAAAnEST to 1 after setting TAAAnCE = 1. The TTRGAAn pin is used for an external trigger.</p> <p><b>Note:</b> The read value of the TAAAnEST bit is always 0.</p>																																				
5	TAAAnEEE	<p>Selects the count clock input of timer TAAAn.</p> <p>0: Use the internal clock (clock selected by TAAAnCTL0.TAAAnCKS[2:0] bits)</p> <p>1: Use external clock (TEVTAAAn input edge)</p> <p>The valid edge is specified with TAAAnEES1 and TAAAnEES0 bits when TAAAnEEE bit = 1 (external clock TEVTAAAn).</p> <p><b>Note:</b> The read value of the TAAAnEST bit is always 0.</p>																																				
2 to 0	TAAAnMD [2:0]	<p>Selects the operation mode of timer TAAAn.</p> <table><tr><th>TAAAnMD2</th><th>TAAAnMD1</th><th>TAAAnMD0</th><th>Timer mode selection</th></tr><tr><td>0</td><td>0</td><td>0</td><td>Interval timer mode</td></tr><tr><td>0</td><td>0</td><td>1</td><td>External event counter mode</td></tr><tr><td>0</td><td>1</td><td>0</td><td>External trigger pulse output mode</td></tr><tr><td>0</td><td>1</td><td>1</td><td>One-shot pulse mode</td></tr><tr><td>1</td><td>0</td><td>0</td><td>PWM mode</td></tr><tr><td>1</td><td>0</td><td>1</td><td>Free-running mode</td></tr><tr><td>1</td><td>1</td><td>0</td><td>Pulse width measurement mode</td></tr><tr><td>1</td><td>1</td><td>1</td><td>Setting prohibited</td></tr></table>	TAAAnMD2	TAAAnMD1	TAAAnMD0	Timer mode selection	0	0	0	Interval timer mode	0	0	1	External event counter mode	0	1	0	External trigger pulse output mode	0	1	1	One-shot pulse mode	1	0	0	PWM mode	1	0	1	Free-running mode	1	1	0	Pulse width measurement mode	1	1	1	Setting prohibited
TAAAnMD2	TAAAnMD1	TAAAnMD0	Timer mode selection																																			
0	0	0	Interval timer mode																																			
0	0	1	External event counter mode																																			
0	1	0	External trigger pulse output mode																																			
0	1	1	One-shot pulse mode																																			
1	0	0	PWM mode																																			
1	0	1	Free-running mode																																			
1	1	0	Pulse width measurement mode																																			
1	1	1	Setting prohibited																																			

**Note** n = 0 to 9

**(3) TAAAnIOC0 - TAA dedicated I/O control register 0**

The TAAAnIOC0 register is an 8-bit register that controls the timer output.

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

**Address**

TAA0IOC0: FFFFF602 <sub>H</sub>	TAA1IOC0: FFFFF612 <sub>H</sub>
TAA2IOC0: FFFFF622 <sub>H</sub>	TAA3IOC0: FFFFF632 <sub>H</sub>
TAA4IOC0: FFFFF642 <sub>H</sub>	TAA5IOC0: FFFFF652 <sub>H</sub>
TAA6IOC0: FFFFF662 <sub>H</sub>	TAA7IOC0: FFFFF672 <sub>H</sub>
TAA8IOC0: FFFFF682 <sub>H</sub>	TAA9IOC0: FFFFF6B2 <sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>TAAAnIOC0</b>	0	0	0	0	TAAAnOL1	TAAAnOE1	TAAAnOL0	TAAAnOE0
	R	R	R	R	R/W	R/W	R/W	R/W

- Caution**
1. Rewrite bits TAAAnOLm and TAAAnOEm when TAAAnCTL0.TAAAnCE = 0 (the same value can be written when TAAAnCE = 1.). If rewriting was mistakenly performed, clear TAAAnCE to 0 and then set the bits again.
  2. To enable the timer output, be sure to set the corresponding alternate-function pins TAAAnIOC1.TAAAnIS[3:0] to "No edge detection" and invalidate the capture operation. Then set the corresponding alternate-function port to output mode.

**Table 16-7 TAAAnIOC0 register contents**

Bit position	Bit name	Function
3, 1	TAAAnOL1 TAAAnOL0	Specifies the TOAAAnm output level. 0: Normal output (inactive level = L, active level = H) 1: Inverted output (inactive level = H, active level = L)  This bit can be used to invert the timer output
2, 0	TAAAnOE1 TAAAnOE0	Controls the TOAAAnm output. 0: Timer output is disabled (inactive level is output depending on the TAAAnOLm bit) 1: Timer output is enabled (TOAAAnm pin outputs pulses.)

**Note** m = 0, 1

**(4) TAAAnIOC1 - TAA dedicated I/O control register 1**

The TAAAnIOC1 register is an 8-bit register that controls the valid edge for the external input signals (TIAAn0 and TIAAn1).

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

**Address**

TAA0IOC1: FFFFF603 <sub>H</sub>	TAA1IOC1: FFFFF613 <sub>H</sub>
TAA2IOC1: FFFFF623 <sub>H</sub>	TAA3IOC1: FFFFF633 <sub>H</sub>
TAA4IOC1: FFFFF643 <sub>H</sub>	TAA5IOC1: FFFFF653 <sub>H</sub>
TAA6IOC1: FFFFF663 <sub>H</sub>	TAA7IOC1: FFFFF673 <sub>H</sub>
TAA8IOC1: FFFFF683 <sub>H</sub>	TAA9IOC1: FFFFF6B3 <sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>TAAAnIOC1</b>	0	0	0	0	TAAAnIS3	TAAAnIS2	TAAAnIS1	TAAAnIS0
	R	R	R	R	R/W	R/W	R/W	R/W

- Caution**
1. Rewrite bits TAAAnIS[3:0] when TAAAnCTL0.TAAAnCE = 0 (the same value can be written when TAAAnCE = 1.). If rewriting was mistakenly performed, clear TAAAnCE to 0 and then set the bits again.
  2. Bits TAAAnIS[3:0] are valid only in the free-running capture mode and pulse width measurement mode. In all the other modes, capture operation is not performed.
  3. If used as the capture input, be sure to set the corresponding alternate-function pins TAAAnOE1 and TAAAnOE0 of the TAAAnIOC0 register to "Timer output is disabled" and set the capture input valid edge. Then set the corresponding alternate-function port to input mode.

**Table 16-8 TAAAnIOC1 register contents**

Bit position	Bit name	Function															
3, 2	TAAIS3 TAAIS2	Specifies the capture input (TIAAn1) valid edge.															
		<table><tr><th>TAAIS3</th><th>TAAIS2</th><th>Capture input (TIAAn1) valid edge setting</th></tr><tr><td>0</td><td>0</td><td>No edge detection (capture operation invalid)</td></tr><tr><td>0</td><td>1</td><td>Rising edge detection</td></tr><tr><td>1</td><td>0</td><td>Falling edge detection</td></tr><tr><td>1</td><td>1</td><td>Both, rising and falling edge detection</td></tr></table>	TAAIS3	TAAIS2	Capture input (TIAAn1) valid edge setting	0	0	No edge detection (capture operation invalid)	0	1	Rising edge detection	1	0	Falling edge detection	1	1	Both, rising and falling edge detection
		TAAIS3	TAAIS2	Capture input (TIAAn1) valid edge setting													
		0	0	No edge detection (capture operation invalid)													
		0	1	Rising edge detection													
		1	0	Falling edge detection													
		1	1	Both, rising and falling edge detection													
Capture operation is performed and capture interrupt (INTTAAAnCC1) is output upon edge detection.																	
1, 0	TAAIS1 TAAIS0	Specifies the capture input (TIAAn0) valid edge.															
		<table><tr><th>TAAIS1</th><th>TAAIS0</th><th>Capture input (TIAAn0) valid edge setting</th></tr><tr><td>0</td><td>0</td><td>No edge detection (capture operation invalid)</td></tr><tr><td>0</td><td>1</td><td>Rising edge detection</td></tr><tr><td>1</td><td>0</td><td>Falling edge detection</td></tr><tr><td>1</td><td>1</td><td>Both, rising and falling edge detection</td></tr></table>	TAAIS1	TAAIS0	Capture input (TIAAn0) valid edge setting	0	0	No edge detection (capture operation invalid)	0	1	Rising edge detection	1	0	Falling edge detection	1	1	Both, rising and falling edge detection
		TAAIS1	TAAIS0	Capture input (TIAAn0) valid edge setting													
		0	0	No edge detection (capture operation invalid)													
		0	1	Rising edge detection													
		1	0	Falling edge detection													
		1	1	Both, rising and falling edge detection													
Capture operation is performed and capture interrupt (INTTAAAnCC0) is output upon edge detection.																	



**(5) TAA<sub>n</sub>IOC2 - TAA I/O control register 2**

The TAA<sub>n</sub>IOC2 register is an 8-bit register that controls the valid edge for external event count input signals (TEVTAA<sub>n</sub>) and external trigger input signal (TTRGAA<sub>n</sub>).

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

<b>Address</b>	TAA0IOC2:	FFFFF604 <sub>H</sub>	TAA1IOC2:	FFFFF614 <sub>H</sub>
	TAA2IOC2:	FFFFF624 <sub>H</sub>	TAA3IOC2:	FFFFF634 <sub>H</sub>
	TAA4IOC2:	FFFFF644 <sub>H</sub>	TAA5IOC2:	FFFFF654 <sub>H</sub>
	TAA6IOC2:	FFFFF664 <sub>H</sub>	TAA7IOC2:	FFFFF674 <sub>H</sub>
	TAA8IOC2:	FFFFF684 <sub>H</sub>	TAA9IOC2:	FFFFF6B4 <sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>TAA<sub>n</sub>IOC2</b>	0	0	0	0	TAA <sub>n</sub> EES1	TAA <sub>n</sub> EES0	TAA <sub>n</sub> ETS1	TAA <sub>n</sub> ETS0
	R	R	R	R	R/W	R/W	R/W	R/W

**Caution** Rewrite TAA<sub>n</sub>EES[1:0] and TAA<sub>n</sub>ETS[1:0] bits when TAA<sub>n</sub>CTL0.TAA<sub>n</sub>CE = 0 (the same value can be written when TAA<sub>n</sub>CE = 1). If rewriting was mistakenly performed, clear TAA<sub>n</sub>CE to 0 and then set the bits again.

**Table 16-9 TAA<sub>n</sub>IOC2 register contents**

Bit position	Bit name	Function															
3, 2	TAA <sub>n</sub> EES1 TAA <sub>n</sub> EES0	Specifies the external event counter input (TEVTAA <sub>n</sub> ) valid edge.															
		<table><tr><th>TAA<sub>n</sub>EES1</th><th>TAA<sub>n</sub>EES0</th><th>External event counter input (TEVTAA<sub>n</sub>) valid edge setting</th></tr><tr><td>0</td><td>0</td><td>No edge detection (capture operation invalid)</td></tr><tr><td>0</td><td>1</td><td>Rising edge detection</td></tr><tr><td>1</td><td>0</td><td>Falling edge detection</td></tr><tr><td>1</td><td>1</td><td>Both, rising and falling edge detection</td></tr></table>	TAA <sub>n</sub> EES1	TAA <sub>n</sub> EES0	External event counter input (TEVTAA <sub>n</sub> ) valid edge setting	0	0	No edge detection (capture operation invalid)	0	1	Rising edge detection	1	0	Falling edge detection	1	1	Both, rising and falling edge detection
		TAA <sub>n</sub> EES1	TAA <sub>n</sub> EES0	External event counter input (TEVTAA <sub>n</sub> ) valid edge setting													
		0	0	No edge detection (capture operation invalid)													
		0	1	Rising edge detection													
		1	0	Falling edge detection													
		1	1	Both, rising and falling edge detection													
<b>Note:</b> TAA <sub>n</sub> EES[1:0] bits are valid only when TAA <sub>n</sub> EEE = 1 or when the external event count mode has been set (TAA <sub>n</sub> CTL1.TAA <sub>n</sub> MD[2:0] = 001B).																	
1, 0	TAA <sub>n</sub> ETS1 TAA <sub>n</sub> ETS0	Specifies the external trigger input (TTRGAA <sub>n</sub> ) valid edge.															
		<table><tr><th>TAA<sub>n</sub>ETS1</th><th>TAA<sub>n</sub>ETS0</th><th>External trigger input (TTRGAA<sub>n</sub>) valid edge setting</th></tr><tr><td>0</td><td>0</td><td>No edge detection (capture operation invalid)</td></tr><tr><td>0</td><td>1</td><td>Rising edge detection</td></tr><tr><td>1</td><td>0</td><td>Falling edge detection</td></tr><tr><td>1</td><td>1</td><td>Both, rising and falling edge detection</td></tr></table>	TAA <sub>n</sub> ETS1	TAA <sub>n</sub> ETS0	External trigger input (TTRGAA <sub>n</sub> ) valid edge setting	0	0	No edge detection (capture operation invalid)	0	1	Rising edge detection	1	0	Falling edge detection	1	1	Both, rising and falling edge detection
		TAA <sub>n</sub> ETS1	TAA <sub>n</sub> ETS0	External trigger input (TTRGAA <sub>n</sub> ) valid edge setting													
		0	0	No edge detection (capture operation invalid)													
		0	1	Rising edge detection													
		1	0	Falling edge detection													
		1	1	Both, rising and falling edge detection													
<b>Note:</b> TAA <sub>n</sub> ETS[1:0] bits are only valid when the external trigger pulse output mode or one-shot pulse mode is set (TAA <sub>n</sub> MD[2:0] = 010B or 011B).																	

**Note** n = 0 to 9

**(6) TAAAnIOC4 - TAA I/O control register 4**

The TAAAnIOC4 register is an 8-bit register that controls the output function of Timer AA.

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

<b>Address</b>	TAA0IOC4:	FFFFF60C <sub>H</sub>	TAA1IOC4:	FFFFF61C <sub>H</sub>
	TAA2IOC4:	FFFFF62C <sub>H</sub>	TAA3IOC4:	FFFFF63C <sub>H</sub>
	TAA4IOC4:	FFFFF64C <sub>H</sub>	TAA5IOC4:	FFFFF65C <sub>H</sub>
	TAA6IOC4:	FFFFF66C <sub>H</sub>	TAA7IOC4:	FFFFF67C <sub>H</sub>
	TAA8IOC4:	FFFFF68C <sub>H</sub>	TAA9IOC4:	FFFFF6BC <sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>TAAAnIOC4</b>	0	0	0	0	TAAAnOS1	TAAAnOR1	TAAAnOS0	TAAAnOR0
	R	R	R	R	R/W	R/W	R/W	R/W

**Caution** The TAAAnIOC4 register can only be used when interval mode or free-running compare mode is selected. In all other modes the TAAAnIOC4 register has to be set to 00<sub>H</sub>.

**Table 16-10 TAAAnIOC4 register contents**

Bit position	Bit name	Function															
3, 2	TAAAnOS1 TAAAnOR1	Controls toggling of the timer output TOAAAn1.															
		<table><tr><th>TAAAnOS1</th><th>TAAAnOR1</th><th>Toggle Control of TOAAAn1</th></tr><tr><td>0</td><td>0</td><td>Standard operation.</td></tr><tr><td>0</td><td>1</td><td>Force output level to inactive at next toggle event</td></tr><tr><td>1</td><td>0</td><td>Force output level to active at next toggle event</td></tr><tr><td>1</td><td>1</td><td>Freeze current output level.</td></tr></table>	TAAAnOS1	TAAAnOR1	Toggle Control of TOAAAn1	0	0	Standard operation.	0	1	Force output level to inactive at next toggle event	1	0	Force output level to active at next toggle event	1	1	Freeze current output level.
		TAAAnOS1	TAAAnOR1	Toggle Control of TOAAAn1													
		0	0	Standard operation.													
		0	1	Force output level to inactive at next toggle event													
		1	0	Force output level to active at next toggle event													
		1	1	Freeze current output level.													
<b>Note:</b> 1. After forcing the output level to either active or inactive, the TOAAAn1 output maintains this level (= no toggling afterwards) until the TAAAnOS1 and TAAAnOR1 are cleared to standard operation.																	
2. The forcing of an output level is executed at the time of the next upcoming toggle event, while the freeze becomes effective immediately.																	
1, 0	TAAAnOS0 TAAAnOR0	Controls toggling of the timer output TOAAAn0.															
		<table><tr><th>TAAAnOS0</th><th>TAAAnOR0</th><th>Toggle Control of TOAAAn0</th></tr><tr><td>0</td><td>0</td><td>Standard operation.</td></tr><tr><td>0</td><td>1</td><td>Force output level to inactive at next toggle event</td></tr><tr><td>1</td><td>0</td><td>Force output level to active at next toggle event</td></tr><tr><td>1</td><td>1</td><td>Freeze current output level.</td></tr></table>	TAAAnOS0	TAAAnOR0	Toggle Control of TOAAAn0	0	0	Standard operation.	0	1	Force output level to inactive at next toggle event	1	0	Force output level to active at next toggle event	1	1	Freeze current output level.
		TAAAnOS0	TAAAnOR0	Toggle Control of TOAAAn0													
		0	0	Standard operation.													
		0	1	Force output level to inactive at next toggle event													
		1	0	Force output level to active at next toggle event													
		1	1	Freeze current output level.													
<b>Note:</b> 1. After forcing the output level to either active or inactive, the TOAAAn0 output maintains this level (= no toggling afterwards) until the TAAAnOS0 and TAAAnOR0 are cleared to standard operation.																	
2. The forcing of an output level is executed at the time of the next upcoming toggle event, while the freeze becomes effective immediately.																	

**(7) TAAOPT0 - TAA option register 0**

The TAAOPT0 register is an 8-bit register used to set the capture/compare operation and detect overflow.

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

**Address**

TAA0OPT0: FFFFF605 <sub>H</sub>	TAA1OPT0: FFFFF615 <sub>H</sub>
TAA2OPT0: FFFFF625 <sub>H</sub>	TAA3OPT0: FFFFF635 <sub>H</sub>
TAA4OPT0: FFFFF645 <sub>H</sub>	TAA5OPT0: FFFFF655 <sub>H</sub>
TAA6OPT0: FFFFF665 <sub>H</sub>	TAA7OPT0: FFFFF675 <sub>H</sub>
TAA8OPT0: FFFFF685 <sub>H</sub>	TAA9OPT0: FFFFF6B5 <sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>TAAOPT0</b>	0	0	TAA <sub>CCS</sub> 1	TAA <sub>CCS</sub> 0	0	0	0	TAA <sub>OVF</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Caution** Rewrite TAA<sub>CCS</sub>1 and TAA<sub>CCS</sub>0 bits when TAA<sub>CTL</sub>0.TAA<sub>CE</sub> = 0 (the same value can be written when TAA<sub>CE</sub> = 1.). If rewriting was mistakenly performed, clear TAA<sub>CE</sub> to 0 and then set the bits again.

**Table 16-11 TAAOPT0 register contents**

Bit position	Bit name	Function
5	TAA <sub>CCS</sub> 1	Specifies the operation mode of register TAA <sub>CCR</sub> 1 0: Operation as compare register 1: Operation as capture register <b>Note:</b> The setting of bit TAA <sub>CCS</sub> 1 is valid in the free-running mode only.
4	TAA <sub>CCS</sub> 0	Specifies the operation mode of register TAA <sub>CCR</sub> 0 0: Operation as compare register 1: Operation as capture register <b>Note:</b> The setting of bit TAA <sub>CCS</sub> 0 is valid in the free-running mode only.
0	TAA <sub>OVF</sub>	Indicates timer TAA overflow 0: No overflow occurrence after timer restart or flag reset 1: Overflow occurrence  <ul style="list-style-type: none"> <li>The TAA<sub>OVF</sub> bit is set when the 16-bit counter value overflows from FFFFH to 0000H in the free-running mode and the pulse width measurement mode.</li> <li>An interrupt request signal (INTTAA<sub>OV</sub>) is generated as soon as TAA<sub>OVF</sub> bit is set (1). The INTTAA<sub>OV</sub> signal is not generated in any mode other than free-running mode and the pulse width measurement mode.</li> <li>The TAA<sub>OVF</sub> bit is cleared by writing 0 to it, or if TAA<sub>CTL</sub>0.TAA<sub>CE</sub> is set to 0.</li> </ul> <b>Caution:</b> 1. When TAA <sub>OVF</sub> = 1, the TAA <sub>OVF</sub> flag is not cleared even if the TAA <sub>OVF</sub> flag and TAAOPT0 register are read. 2. The TAA <sub>OVF</sub> flag can be read and written, but even if 1 is written to the TAA <sub>OVF</sub> flag from the CPU, this is ignored and has no influence on the timer operation.

**(8) TAA<sub>n</sub>OPT1 - TAA option register 1**

The TAA<sub>n</sub>OPT1 register is an 8-bit register used to set the 32-bit capture mode by cascading two Timer AA.

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

**Address** TAA1OPT1: FFFFF61D<sub>H</sub>                      TAA2OPT1=FFFFF62D<sub>H</sub>  
 TAA3OPT1: FFFFF63D<sub>H</sub>                      TAA5OPT1=FFFFF65D<sub>H</sub>  
 TAA6OPT1: FFFFF66D<sub>H</sub>                      TAA7OPT1=FFFFF67D<sub>H</sub>  
 TAA9OPT1: FFFFF6BD<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>TAA<sub>n</sub>OPT1</b>	TAA <sub>n</sub> CSE	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 16-12 TAA<sub>n</sub>OPT0 register contents**

Bit position	Bit name	Function
7	TAA <sub>n</sub> CSE	<p>Controls the cascade mode of timers TAA<sub>n</sub>/TAA<sub>m</sub>.            0: 16-bit non-cascaded mode            1: 32-bit cascaded capture mode.            Timer AA<sub>n</sub> becomes the upper 16-bit and slave. The master timer is TAA<sub>m</sub> (where m = 0, when n = 1 to 3; m = 4, when n = 5 to 7; m = 8, when n = 9).</p> <p><b>Note:</b> 1. When setting TAA<sub>n</sub>CSE, the timer becomes the upper 16-bit of a 32-bit timer.            2. Cascading is only available for capture with free-running counter.            3. The following pairs of timers can be cascaded:</p> <ul style="list-style-type: none"> <li>• TAA0 and one of TAA1 to TAA3              (TAA0 will become master and will hold the lower 16-bit value)</li> <li>• TAA4 and one of TAA5 to TAA7              (TAA4 will become master and will hold the lower 16-bit value)</li> <li>• TAA8 and TAA9              (TAA8 will become master and will hold the lower 16-bit value)</li> </ul>

The table below shows the effects of the TAA<sub>n</sub>CSE flag on the timer operation:

**Table 16-13 Timer Operation in Non-cascaded/Cascaded Capture Mode**

	TAA <sub>n</sub> CSE=0	TAA <sub>n</sub> CSE=1
Operating clock	macro clock from clock tree	macro clock of TAA <sub>m</sub>
Count Enable	TAA <sub>n</sub> CE bit of TAA <sub>n</sub> CTL0	TAA <sub>m</sub> CE bit of TAA <sub>m</sub>
Count Clock	selected by TAA <sub>n</sub> CKS[2:0]	Counter overflow from TAA <sub>m</sub>
Capture Signal 0	TIAA <sub>n</sub> 0 input with edge filter as selected by TAA <sub>n</sub> IS[1:0]	TIAA <sub>m</sub> 0 with edge filter selected for TAA <sub>m</sub>
Capture Signal 1	TIAA <sub>n</sub> 1 input with edge filter as selected by TAA <sub>n</sub> IS[3:2]	TIAA <sub>m</sub> 1 with edge filter selected for TAA <sub>m</sub>
Capture Interrupt	INTTAA <sub>n</sub> CC0 or INTTAA <sub>n</sub> CC1	INTTAA <sub>m</sub> CC0 or INTTAA <sub>m</sub> CC1

n=1 or 3; m= (n-1).

For details on the 32-bit capture mode, refer to “32-bit Capture in free-running cascade mode” on page 600.

**(9) TAAAnSVC - TAA specific overflow value control register**

The TAAAnSVC controls the setting of defined overflow values OVfV to the TAAAn capture/compare registers in case of overflows.

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

**Address**

TAA0SVC: FFFFF60E <sub>H</sub>	TAA1SVC: FFFFF61E <sub>H</sub>
TAA2SVC: FFFFF62E <sub>H</sub>	TAA3SVC: FFFFF63E <sub>H</sub>
TAA4SVC: FFFFF64E <sub>H</sub>	TAA5SVC: FFFFF65E <sub>H</sub>
TAA6SVC: FFFFF66E <sub>H</sub>	TAA7SVC: FFFFF67E <sub>H</sub>
TAA8SVC: FFFFF68E <sub>H</sub>	TAA9SVC: FFFFF6BE <sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>TAAAnSVC</b>	TAAAnENC1	TAAAnCSE1	TAAAnCSD1	TAAAnCSR1	TAAAnENC0	TAAAnCSE0	TAAAnCSD0	TAAAnCSR0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Caution**
1. Write to the TAAAnSVC register is only permitted, when TAAAnCTL0.TAAAnCE = 0.
  2. The capture inputs TIAAn0 and TIAAn1 of TAAAn must be connected externally, so that both inputs get the same signal.

**Table 16-14 TAAAnSVC register contents (1/2)**

Bit position	Bit name	Function
7, 3	TAAAnENC1 TAAAnENC0	Controls setting of TAAAnCCRm register overflow value. 0: Overflow value writing disabled. 1: Overflow value writing enabled. TAAAnCCRm is overwritten upon INTAAAnOV generation.
6, 2	TAAAnCSE1 TAAAnCSE0	Specifies the TIAAnm overflow value (OVfV). 0: If TIAAnm = L (0) during overflow: overflow value OVfV = 0xFFFF If TIAAnm = H (1) during overflow: overflow value OVfV = 0x0000 1: If TIAAnm = H (1) during overflow: overflow value OVfV = 0xFFFF If TIAAnm = L (0) during overflow: overflow value OVfV = 0x0000  TAAAnCSEm determines also the selection of the valid capture input trigger edge of the input signals TIAAnm in combination with TAAAnIOC1.TAAAnIS[3:2] and TAAAnIOC1.TAAAnIS[1:0] respectively. Upon an overflow event a comparison is made between the level of TIAAnm input signal and the TAAAnCSEm bit. The result of the comparison determines the overflow value: <ul style="list-style-type: none"> <li>• if TIAAnm = TAAAnCSEm: overflow value OVfV = 0xFFFF (match)</li> <li>• if TIAAnm = not TAAAnCSEm: overflow value OVfV = 0x0000 (no match)</li> </ul> The overflow value finally set to the TAAAnCCRm register depends also on TAAAnCSDm and TAAAnCSRm.

**Table 16-14 TAAAnSVC register contents (2/2)**

Bit position	Bit name	Function
5, 1	TAAAnCSD1 TAAAnCSD0	<p>Specified the TAAAnCCRm overwrite operation.</p> <p>0: Overflow value is written to TAAAnCCRm in case of an overflow event, independent of TIAAnm level and TAAAnCSEn.</p> <p>1: Overflow value, written to TAAAnCCRm in case of an overflow event, depends on level of TIAAnm and TAAAnCSEn:</p> <ul style="list-style-type: none"> <li>if TIAAnm = TAAAnCSEm: overflow value OVfV set to TAAAnCCRm</li> <li>if TIAAnm = not(TAAAnCSEm): TAAAnCCRm remains unchanged</li> </ul> <p>By use of the TAAAnCSDm bit, overwriting of TAAAnCCRm by an overflow value OVfV can be prohibited, if TIAAnm = not(TAAAnCSEm).</p>
4, 0	TAAAnCSR1 TAAAnCSR0	<p>Overflow value OVfV inversion control</p> <p>0: Overflow value OVfV is not inverted.</p> <p>1: Overflow value OVfV is inverted.</p> <p>By use of the TAAAnCSRm bit, the overflow value OVfV written to TAAAnCCRm can be inverted.</p>

Overwriting of the TAAAnCCRm register coincide with the generation of the overflow interrupt INTTAAAnCCm and setting of the overflow flag TAAAnOPT0.TAAAnOVf.

The following table summarizes the values written to the TAAAnCCRm register under the various control settings of TAAAnSVC, in case the overflow setting function is enabled for capture/compare channel m by TAAAnSVC.TAAAnENCm = 1.

**Valid edge specification** The valid edges of the capture trigger inputs TIAAnm are determined by TAAAnSVC.TAAAnCSE0 and TAAAnIOC1.TAAAnIS[3:2], respectively TAAAnIOC1.TAAAnIS[1:0]. These bits must be set according to *Table 16-15*.

**Table 16-15 Overflow value setting conditions (1/2)**

TAAAnCSE0	TAAAnCSD0	TAAAnCSR0	TIAAn0	TAAAnIS1	TAAAnIS0	TAAAnCCR0	Remark
TAAAnCSE1	TAAAnCSD1	TAAAnCSR1	TIAAn1	TAAAnIS3	TAAAnIS2	TAAAnCCR1	
0	0	0	L	X	1	0xFFFF	TIAAnm = TAAAnCSEm
			H	X	1	0x0000	$\overline{\text{TIAAnm}} = \overline{\text{TAAAnCSEm}}$
		1	L	X	1	0x0000	reverse of above
			H	X	1	0xFFFF	
	1	0	L	X	1	0xFFFF	TIAAnm = TAAAnCSEm
			H	X	1	no change	$\overline{\text{TIAAnm}} = \overline{\text{TAAAnCSEm}}$
		1	L	X	1	no change	reverse of above
			H	X	1	0xFFFF	

Table 16-15 Overflow value setting conditions (2/2)

TAAncSE0	TAAncSD0	TAAncSR0	TIAnc0	TAAncIS1	TAAncIS0	TAAncCCR0	Remark
TAAncSE1	TAAncSD1	TAAncSR1	TIAnc1	TAAncIS3	TAAncIS2	TAAncCCR1	
1	0	0	L	1	X	0x0000	$TIAncm = \overline{TAAncSEm}$
			H	1	X	0xFFFF	$TIAncm = TAAncSEm$
		1	L	1	X	0xFFFF	reverse of above
			H	1	X	0x0000	
	1	0	L	1	X	no change	$TIAncm = \overline{TAAncSEm}$
			H	1	X	0xFFFF	$TIAncm = TAAncSEm$
		1	L	1	X	0xFFFF	reverse of above
			H	1	X	no change	

The following flowchart illustrates how the final content of the TAAncCCRm is evaluated.

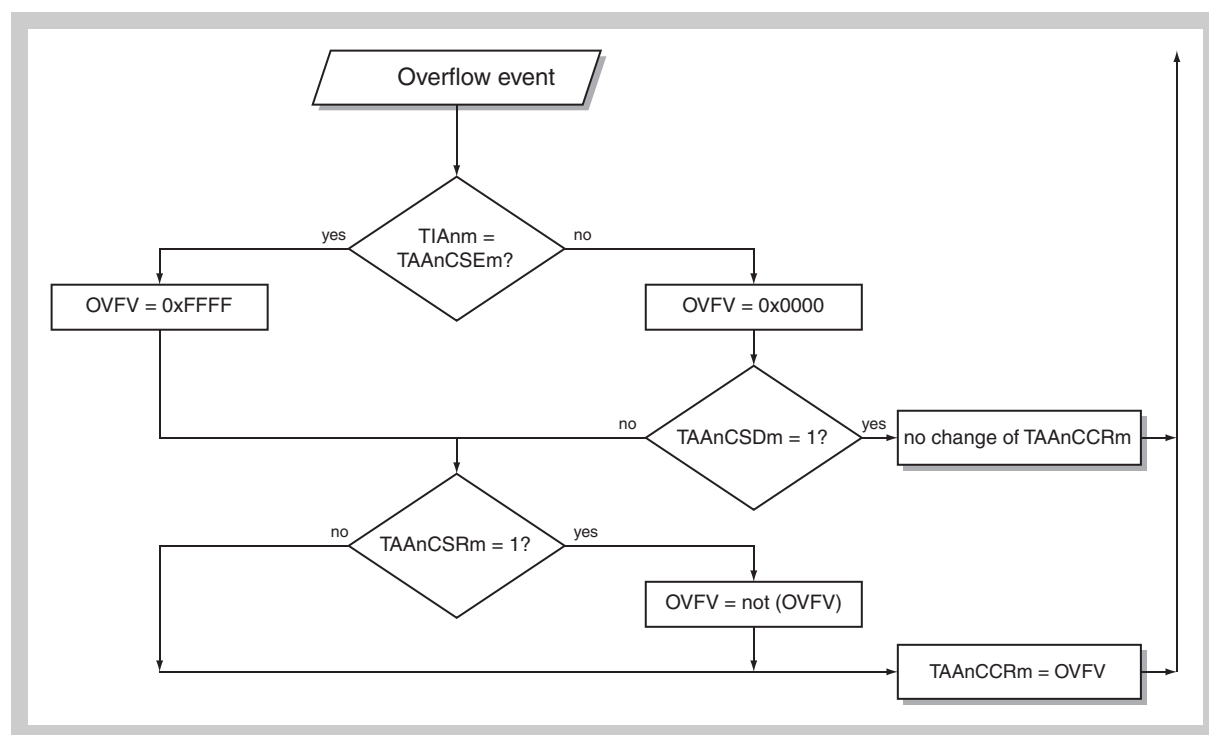


Figure 16-6 Evaluation of overflow value OVfV

For further information concerning the overflow value function refer to “Specific TAAncCCRm overflow value write” on page 605.

## 16.6 Operation

Timer AA can perform the following operations when not in cascade mode:

Operation	TAAAnEST Software trigger bit	TIAAn0 External trigger input	TAAAnEEE Count clock selection	Capture/Compare Selection	Compare Write
Interval timer mode	Invalid	Invalid	Internal/TIAAn0 pin	Compare only	Any time write
External event counter mode	Invalid	Invalid	External only	Compare only	Any time write
External trigger pulse output mode <sup>Note</sup>	Valid	Valid	Internal only	Compare only	Reload
One-shot pulse output mode <sup>Note</sup>	Valid	Valid	Internal only	Compare only	Any time write
PWM mode	Invalid	Invalid	Internal/TIAAn0 pin	Compare only	Reload
Free-running mode	Invalid	Invalid	Internal/TIAAn0 pin	Capture/compare selectable	Any time write
Pulse width measurement mode <sup>Note</sup>	Invalid	Invalid	Internal only	Capture only	Not applicable

**Note** To use the external trigger pulse output mode, one-shot pulse mode, or pulse width measurement mode, select a count clock by clearing the TAAAnEEE bit of the TAAAnCTL1 register to 0.



### 16.6.1 Anytime write and reload

TAAAnCCR0 and TAAAnCCR1 register rewrite is possible for timer AA during timer operation (TAAAnCE = 1), but the write method (any time write, reload) differs depending on the mode.

#### (1) Anytime write

When data is written to the TAAAnCCRm register during timer operation, it is transferred at any time to CCRm buffer register and used as the 16-bit counter comparison value.

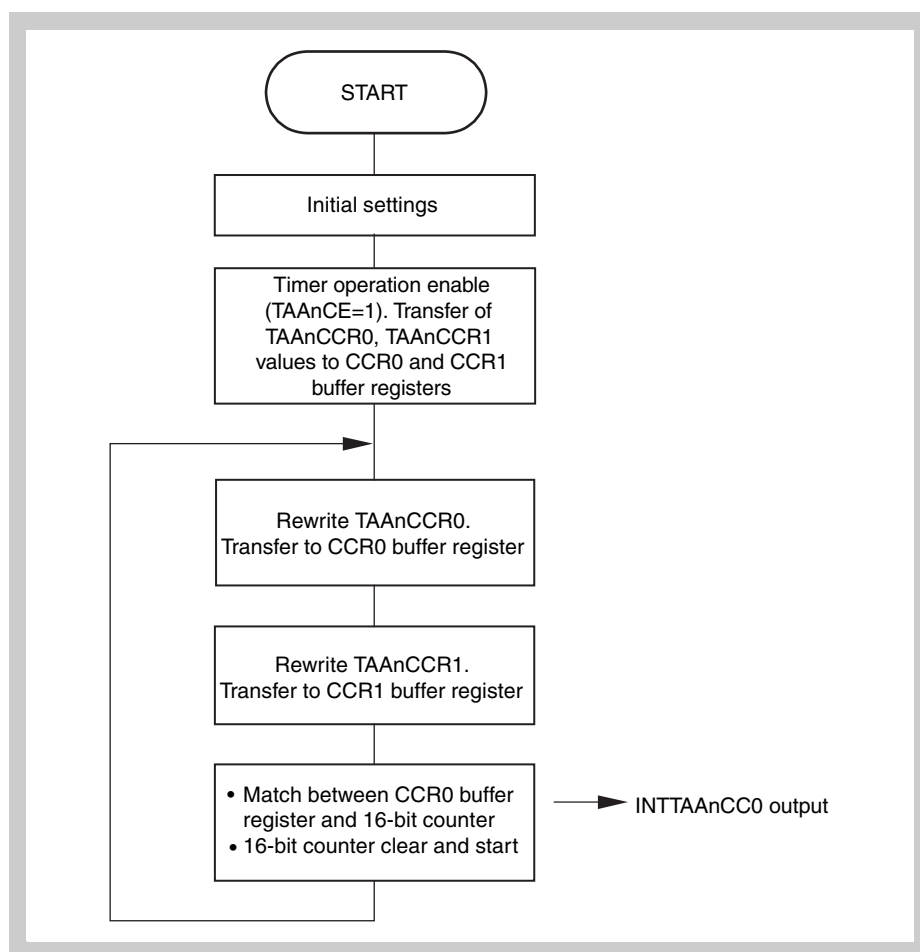
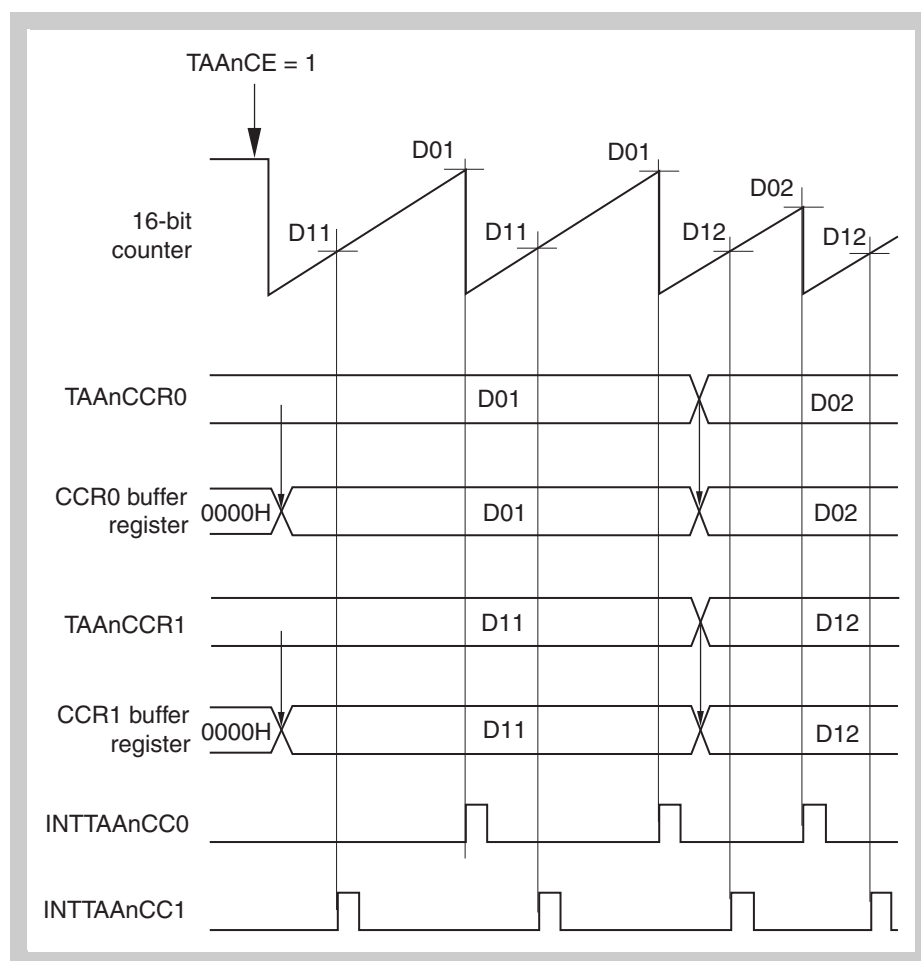


Figure 16-7 Flowchart of basic operation for anytime write

- Note**
1. The above flowchart illustrates an example of the operation in the interval timer mode.
  2.  $n = 0$  to  $9$ ;  $m = 0, 1$



**Figure 16-8** Timing diagram for anytime write

D01, D02: Setting values of TAAAnCCR0 register (0000H to FFFFH)

D11, D12: Setting values of TAAAnCCR1 register (0000H to FFFFH)

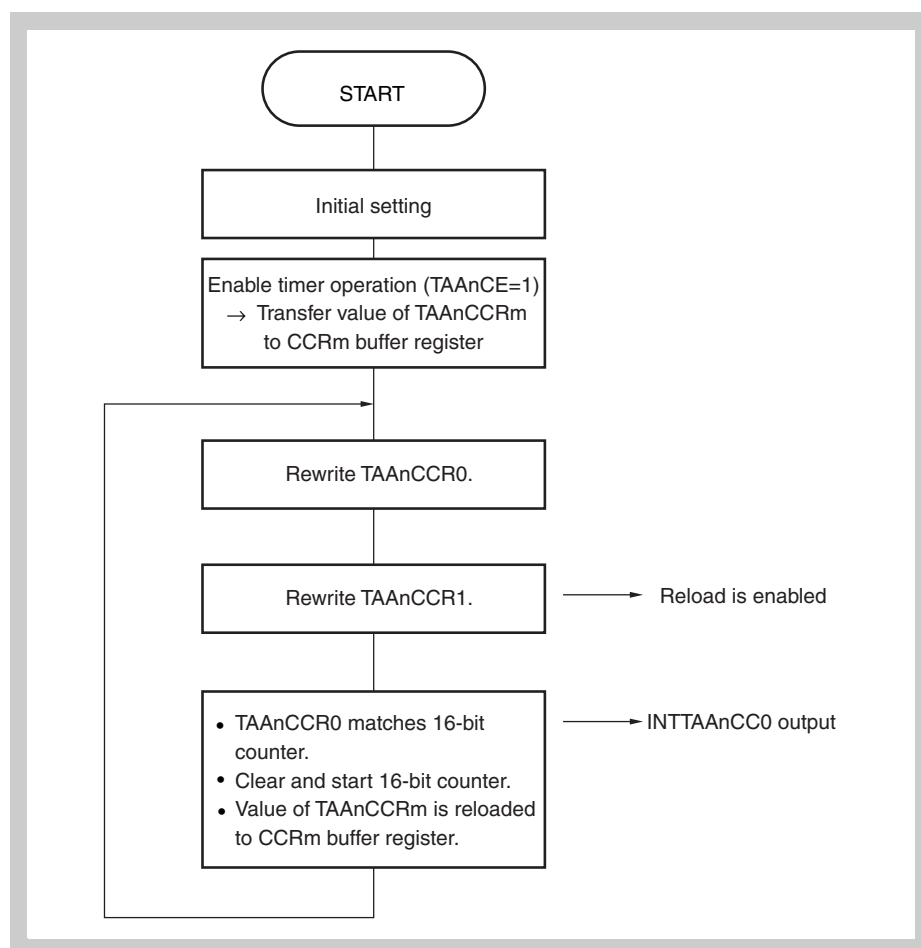
The above timing chart illustrates an example of the operation in the interval timer mode.

**(2) Reload**

When data is written to the TAA<sub>n</sub>CCR0 and TAA<sub>n</sub>CCR1 registers during timer operation, it is compared with the value of the 16-bit counter via the CCR<sub>m</sub> buffer register. The values of the TAA<sub>n</sub>CCR0 and TAA<sub>n</sub>CCR1 registers can be rewritten when TAA<sub>n</sub>CE = 1.

So that the set values of the TAA<sub>n</sub>CCR0 and TAA<sub>n</sub>CCR1 registers are compared with the value of the 16-bit counter (the set values are reloaded to the CCR<sub>m</sub> buffer register), the value of the TAA<sub>n</sub>CCR0 register must be rewritten and then a value must be written to the TAA<sub>n</sub>CCR1 register before the value of the 16-bit counter matches the value of TAA<sub>n</sub>CCR0. When the value of the TAA<sub>n</sub>CCR0 register matches the value of the 16-bit counter, the values of the TAA<sub>n</sub>CCR0 and TAA<sub>n</sub>CCR1 registers are reloaded.

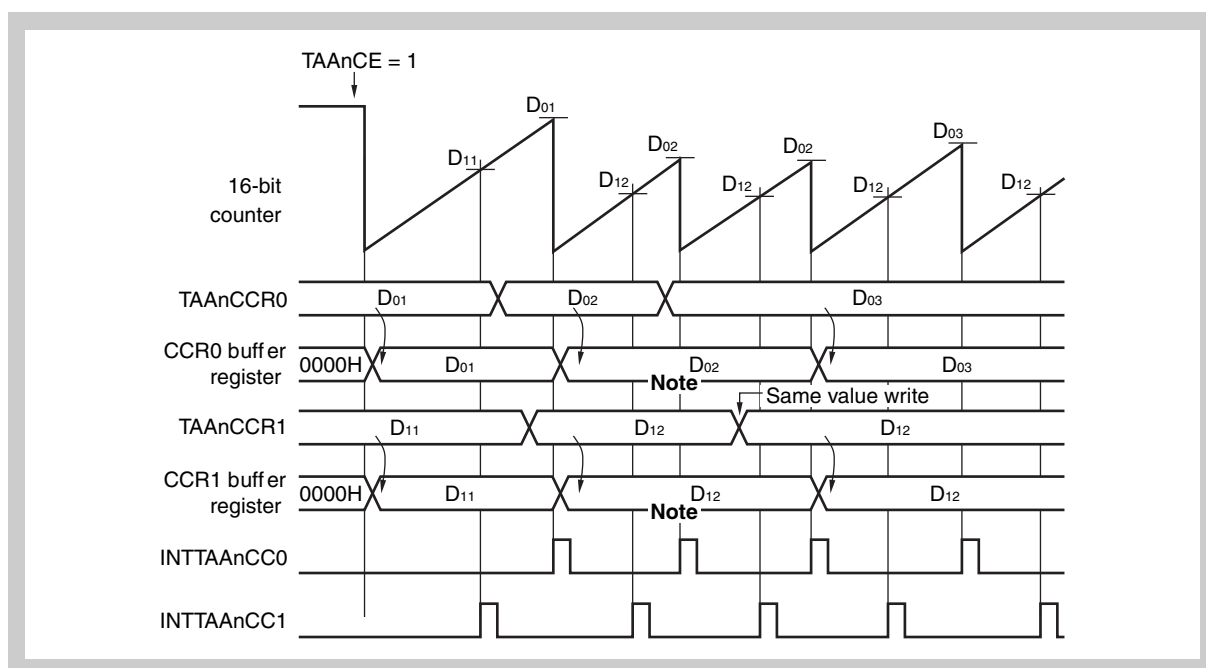
Whether the next reload timing is made valid or not is controlled by writing to the TAA<sub>n</sub>CCR1 register. Therefore, write the same value to the TAA<sub>n</sub>CCR1 register when it is necessary to rewrite the value of only the TAA<sub>n</sub>CCR0 register.



**Figure 16-9** Flowchart of basic operation for reload

**Caution** Writing to the TAA<sub>n</sub>CCR1 register includes an operation to enable reload. Therefore, rewrite the TAA<sub>n</sub>CCR1 register after rewriting the TAA<sub>n</sub>CCR0 register.

- Note**
1. Above flowchart illustrates an example of the PWM mode operation.
  2.  $n = 0$  to  $9$ ;  $m = 0, 1$



**Figure 16-10** Timing chart for reload

**Note** Reload is not performed because TAAAnCCR1 register is not written.

D01, D02, D03: Setting value of TAAAnCCR0 register (0000H to FFFFH)

D11, D12: Setting value of TAAAnCCR1 register (0000H to FFFFH)

Above flowchart illustrates PWM mode operation.

### 16.6.2 Interval timer mode (TAA<sub>n</sub>MD2 to TAA<sub>n</sub>MD0 = 000<sub>B</sub>)

In the interval timer mode, an interrupt request signal (INTTAA<sub>n</sub>CC0) is generated upon a match between the setting value of the TAA<sub>n</sub>CCR0 register and the value of the 16-bit counter, and the 16-bit counter is cleared. The TAA<sub>n</sub>CCR0 register can be rewritten when TAA<sub>n</sub>CE = 1, and when a value is set to TAA<sub>n</sub>CCR0 with a write instruction from the CPU, it is transferred to the CCR0 buffer register through any time write mode, and is compared with the 16-bit counter value.

In the interval timer mode, the 16-bit counter is cleared only upon a match between the value of the 16-bit counter and the value of the CCR0 buffer register.

16-bit counter clearing using the TAA<sub>n</sub>CCR1 register is not performed. However, the setting value of the TAA<sub>n</sub>CCR1 register is transferred to the CCR1 buffer register and compared with the value of the 16-bit counter, and an interrupt request (INTTAA<sub>n</sub>CC1) is output if these values match. In addition, TOAA<sub>n</sub> pin output is also possible by setting the TAA<sub>n</sub>OE1 bit to 1.

When the TAA<sub>n</sub>CCR1 register is not used, it is recommended to set FFFFH as the setting value for the TAA<sub>n</sub>CCR1 register.

When performing timer output with the TOAA<sub>n</sub> pin, set the same values to the TAA<sub>n</sub>CCR0 register and the TAA<sub>n</sub>CCR1 register since the 16-bit timer counter cannot be cleared with the TAA<sub>n</sub>CCR1 register.

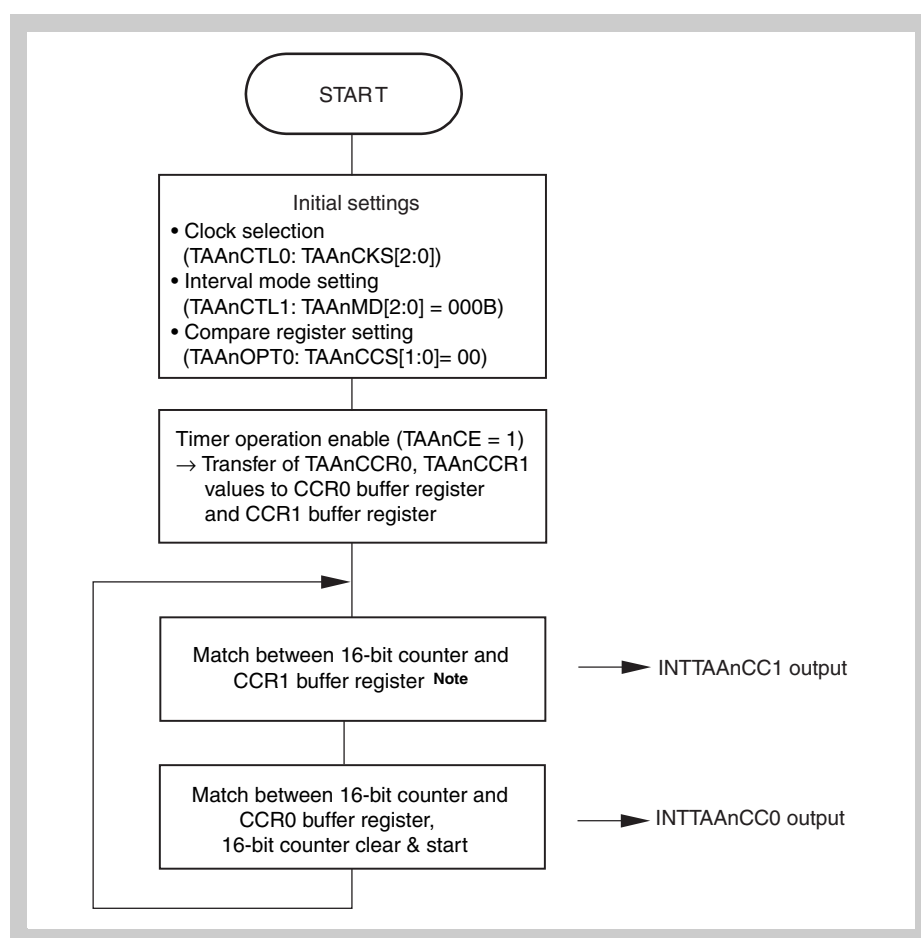
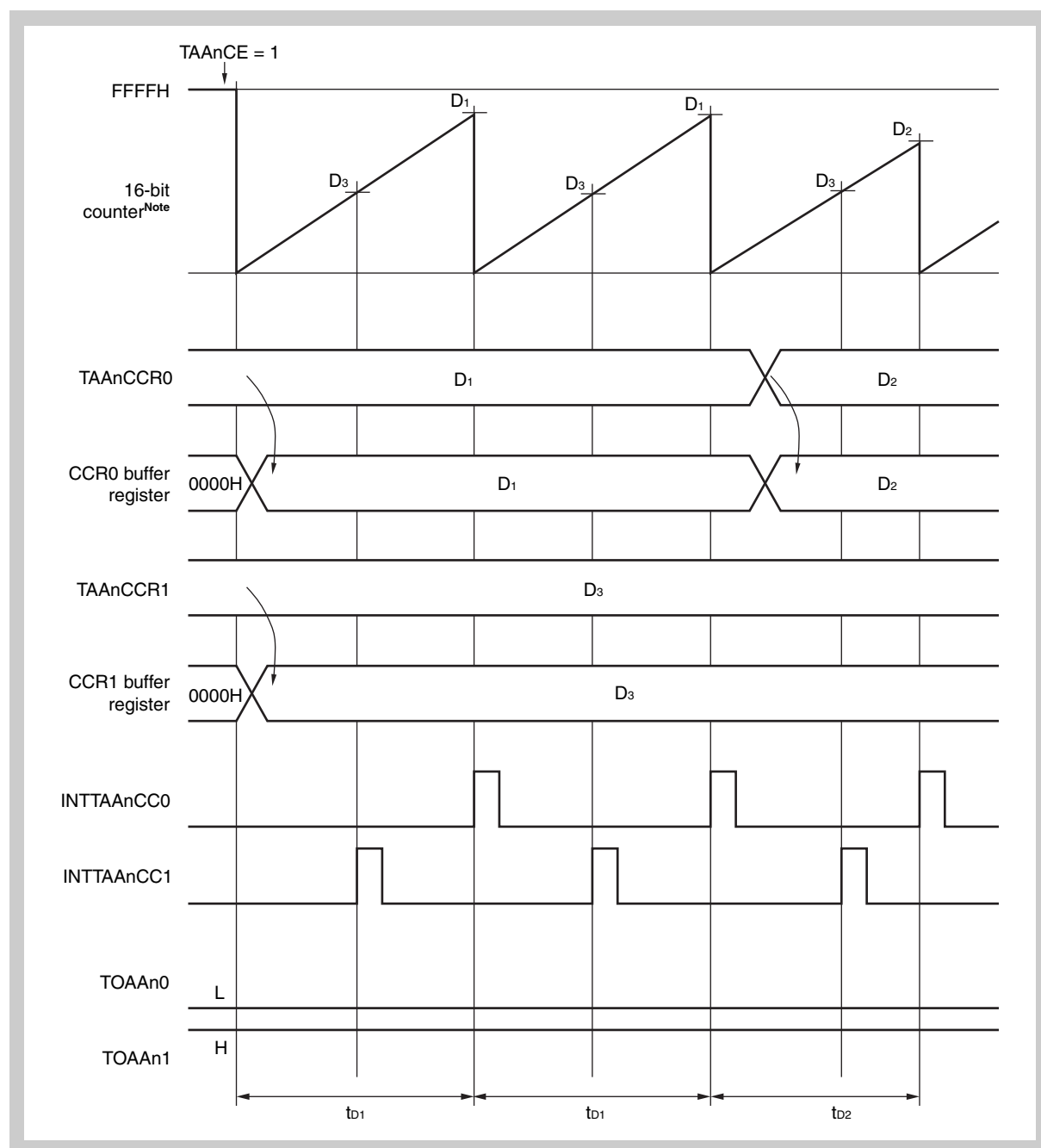


Figure 16-11 Flowchart of basic operation in interval timer mode

**Note** The 16-bit counter is not cleared when its value matches the value of TAA<sub>n</sub>CCR1.

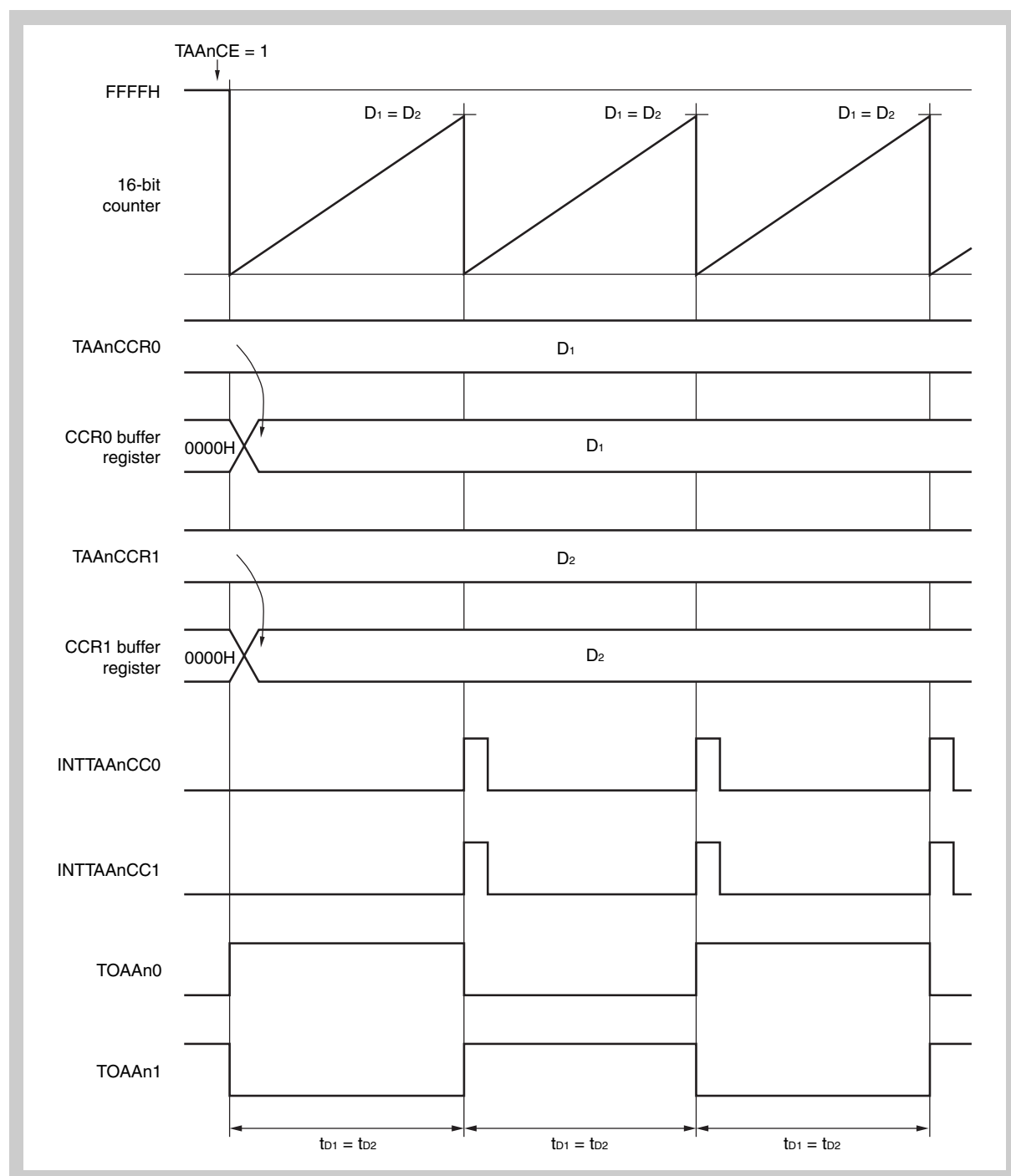


**Figure 16-12 Basic operation timing in interval timer mode**  
 When  $D1 > D2 > D3$ ; only TAAAnCCR0 register value is written, and TOAAAn0 and TOAAAn1 are not output  
 (TAAAnOE0 = 0, TAAAnOE1 = 0, TAAAnOL0 = 0, TAAAnOL1 = 1)

**Note** The 16-bit counter is not cleared when its value matches the value of TAAAnCCR1.

D1, D2: Setting values of TAAAnCCR0 register (0000H to FFFFH)  
 D3: Setting value of TAAAnCCR1 register (0000H to FFFFH)

Interval time ( $t_{Dn}$ ) =  $(Dn + 1) \times (\text{count clock cycle})$



**Figure 16-13 Basic operation timing in interval timer mode**  
 When  $D_1 = D_2$ ; TAAAnCCR0 and TAAAnCCR1 are not rewritten, and TOAAAn0 and TOAAAn1 are output  
 (TAAAnOE0 = 1, TAAAnOE1 = 1, TAAAnOL0 = 0, TAAAnOL1 = 1)

$D_1$ : Setting value of TAAAnCCR0 register (0000H to FFFFH)  
 $D_2$ : Setting value of TAAAnCCR1 register (0000H to FFFFH)

Interval time ( $t_{Dn}$ ) =  $(Dn + 1) \times (\text{count clock cycle})$

### 16.6.3 External event counter mode (TAA<sub>n</sub>MD2 to TAA<sub>n</sub>MD0 = 001<sub>B</sub>)

In the external event count mode, the external event count input (TEVTAA<sub>n</sub> pin input) is used as a count-up signal. Regardless of the setting of the TAA<sub>n</sub>EEE bit of the TAA<sub>n</sub>CTL0 register, 16-bit timer/event counter AA counts up the external event count input (TEVTAA<sub>n</sub> pin input) when it is set in the external event count mode. In the external event count mode, an interrupt request (INTTAA<sub>n</sub>CC0) is generated when the set value of the TAA<sub>n</sub>CCR0 register matches the value of the 16-bit counter, and the value of the 16-bit counter is cleared.

When a value is set to the TAA<sub>n</sub>CCR0 register with a write instruction from the CPU, it is transferred to the CCR0 buffer register through any time write, and is compared with the 16-bit counter value.

In the external event counter mode, the 16-bit counter is cleared only upon a match between the value of the 16-bit counter and the value of the CCR0 buffer register.

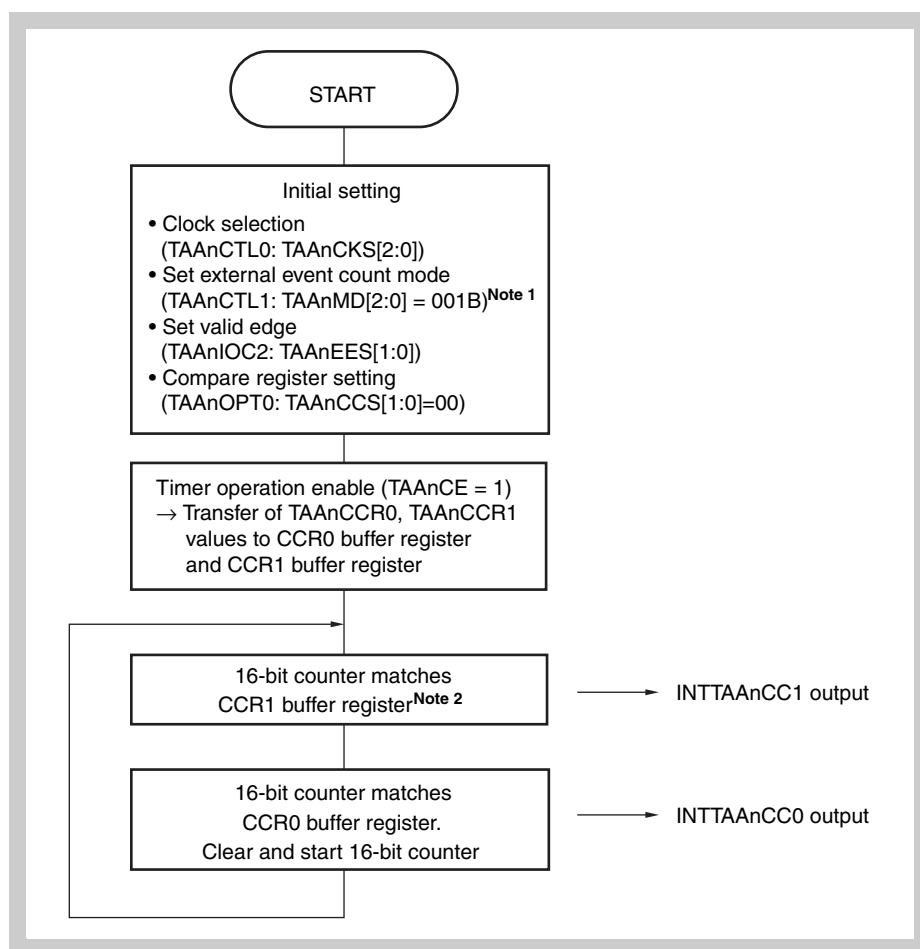
The 16-bit counter can not be cleared using TAA<sub>n</sub>CCR1 register. However, the setting value of the TAA<sub>n</sub>CCR1 register is transferred to the CCR1 buffer register and compared with the value of the 16-bit counter, and an interrupt request (INTTAA<sub>n</sub>CC1) is output if these values match.

Moreover, TOAA<sub>n</sub>m pin output is also possible by setting the TAA<sub>n</sub>OEm bit to 1.

When performing timer output with the TOAA<sub>n</sub>1 pin, set the same values to TAA<sub>n</sub>CCR0 register and TAA<sub>n</sub>CCR1 register since the 16-bit counter cannot be cleared with CCR1 buffer register.

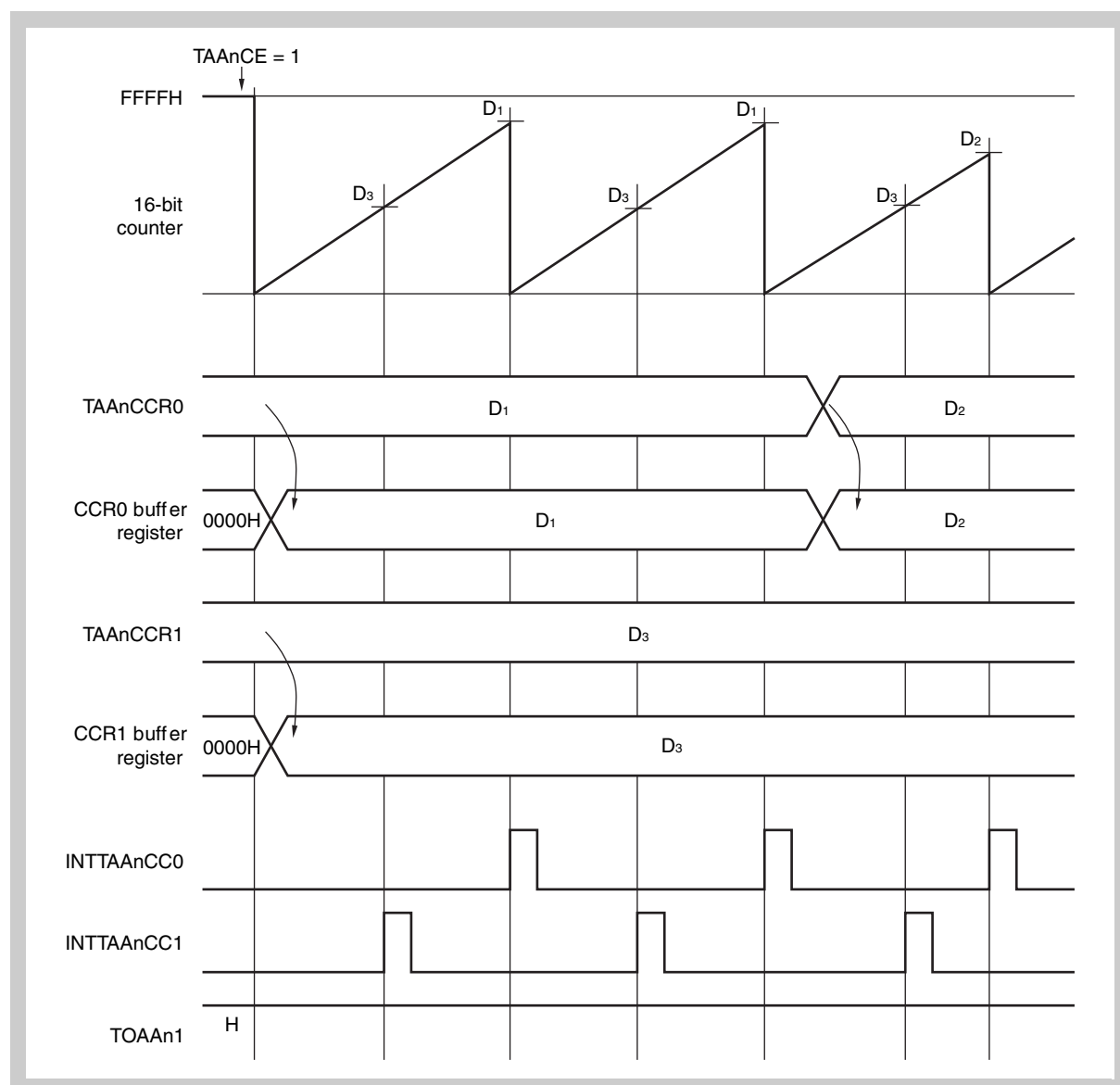
The TAA<sub>n</sub>CCR0 register can be rewritten when TAA<sub>n</sub>CE = 1. When TAA<sub>n</sub>CCR1 register is not used, it is recommended to set TAA<sub>n</sub>CCR1 register to FFFFH.





**Figure 16-14** Flowchart of basic operation in external event counter mode

- Note**
1. Selection of the TAAAnEEE bit has no influence.
  2. The 16-bit counter is not cleared upon a match between the 16-bit counter and the CCR1 buffer register.
  3.  $n = 0$  to  $9$ ;  $m = 0, 1$

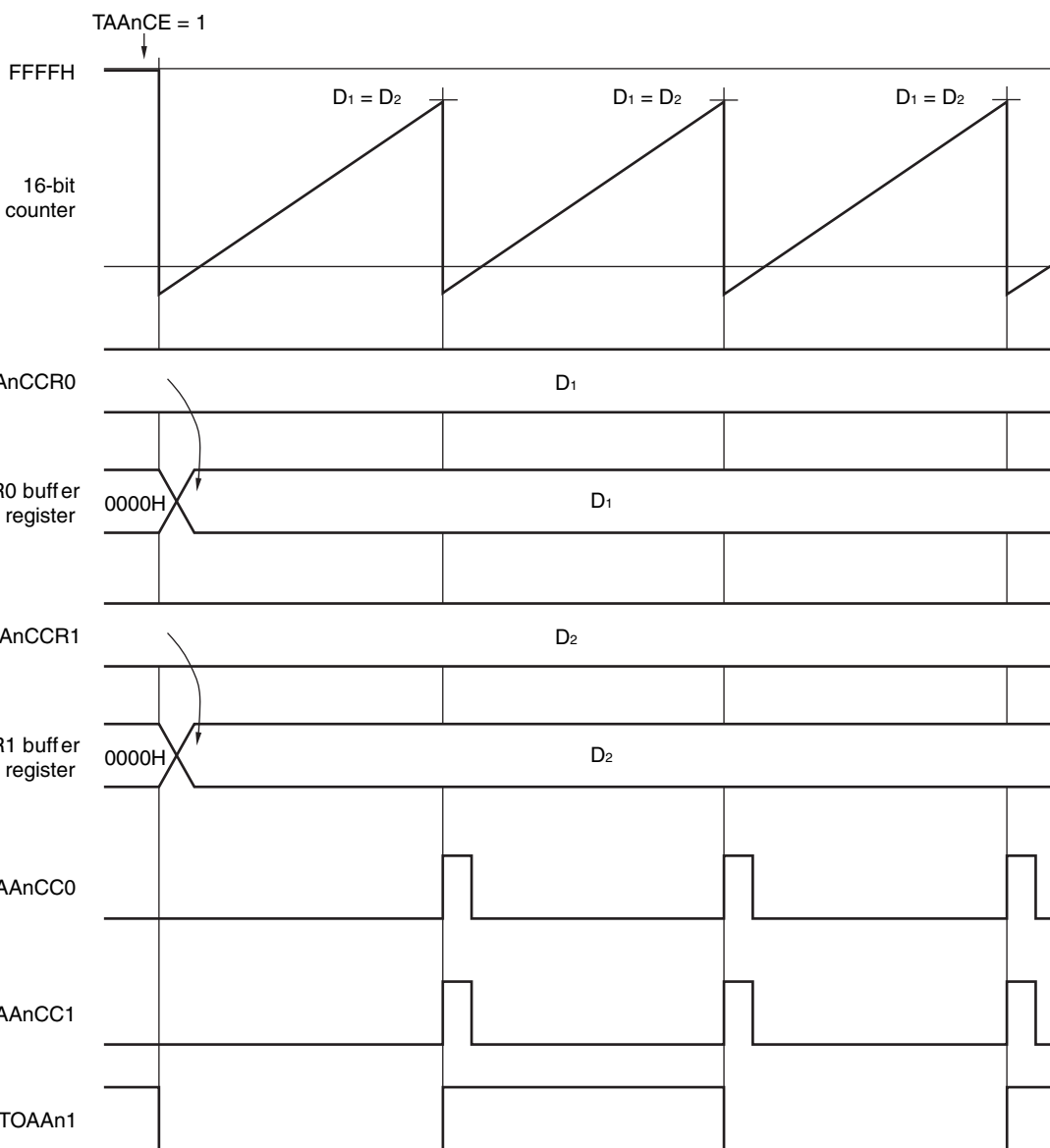


**Figure 16-15 Basic operation timing in external event counter mode**  
 When  $D1 > D2 > D3$ ; rewrite TAAAnCCR0 only; TOAAAn1 is not output  
 (TAAAnOE0 = 0, TAAAnOE1 = 0, TAAAnOL0 = 0, TAAAnOL1 = 1)

D1, D2: Setting values of TAAAnCCR0 register (0000H to FFFFH)

D3: Setting value of TAAAnCCR1 register (0000H to FFFFH)

Number of event counts =  $(Dn + 1)$



**Figure 16-16 Basic operation timing in external event counter mode**  
When D1 = D2; TAAAnCCR0 and TAAAnCCR1 are not rewritten, TOAAAn1 is output  
(TAAAnOE0 = 1, TAAAnOE1 = 1, TAAAnOL0 = 0, TAAAnOL1 = 1)

### D1: Setting value of TAAAnCCR0 register (0000H to FFFFH)

D2: Setting value of TAAAnCCR1 register (0000H to FFFFH)

Number of event count =  $(D_n + 1)$

### 16.6.4 External trigger pulse mode (TAA<sub>n</sub>MD2 to TAA<sub>n</sub>MD0 = 010<sub>B</sub>)

When TAA<sub>n</sub>CE = 1 in the external trigger pulse mode, the 16-bit counter stops at FFFFH and waits for input of an external trigger (TTRGAAn pin input). When the counter detects the edge of the external trigger (TTRGAAn pin input), it starts counting up. The duty factor of the signal output from the TOAAn1 pin is set by a reload register (TAA<sub>n</sub>CCR1) and the period is set by a compare register (TAA<sub>n</sub>CCR0).

Rewriting the TAA<sub>n</sub>CCR0 and TAA<sub>n</sub>CCR1 registers is enabled when TAA<sub>n</sub>CE = 1.

To ensure that the selected values of the TAA<sub>n</sub>CCR0 and TAA<sub>n</sub>CCR1 registers after rewriting are compared with the value of the 16-bit counter (reloaded to the CCRm buffer register), the TAA<sub>n</sub>CCR0 register and then the TAA<sub>n</sub>CCR1 register must be written before the value of the 16-bit counter matches the value of the TAA<sub>n</sub>CCR0 register.

When the value of the TAA<sub>n</sub>CCR0 register later matches the value of the 16-bit counter, the values of the TAA<sub>n</sub>CCR0 and TAA<sub>n</sub>CCR1 registers are reloaded to the CCRm buffer register.

Whether the next reload timing is made valid or not is controlled by writing to the TAA<sub>n</sub>CCR1 register. Therefore, write the same value to the TAA<sub>n</sub>CCR1 register when it is necessary to rewrite the value of only the TAA<sub>n</sub>CCR0 register.

Reload is invalid when only the TAA<sub>n</sub>CCR0 register is rewritten. To stop timer AA, clear TAA<sub>n</sub>CE to 0. If the edge of the external trigger (TTRGAAn pin input) is detected more than once in the external trigger pulse mode, the 16-bit counter is cleared at the point of edge detection, and resumes counting up. To realize the same function as the external trigger pulse mode by using a software trigger instead of the external trigger input (TTRGAAn pin input) (software trigger pulse mode), a software trigger is generated by setting the TAA<sub>n</sub>EST bit of the TAA<sub>n</sub>CTL1 register to 1.

When using a software trigger, a square wave that has one cycle of the PWM waveform as half of its own cycle can also be outputted from the TOAAn0 pin.

The waveform of the external trigger pulse is output from TOAAn1. A toggle output is produced from the TOAAn0 pin when the value of the TAA<sub>n</sub>CCR0 register matches the value of the 16-bit counter.

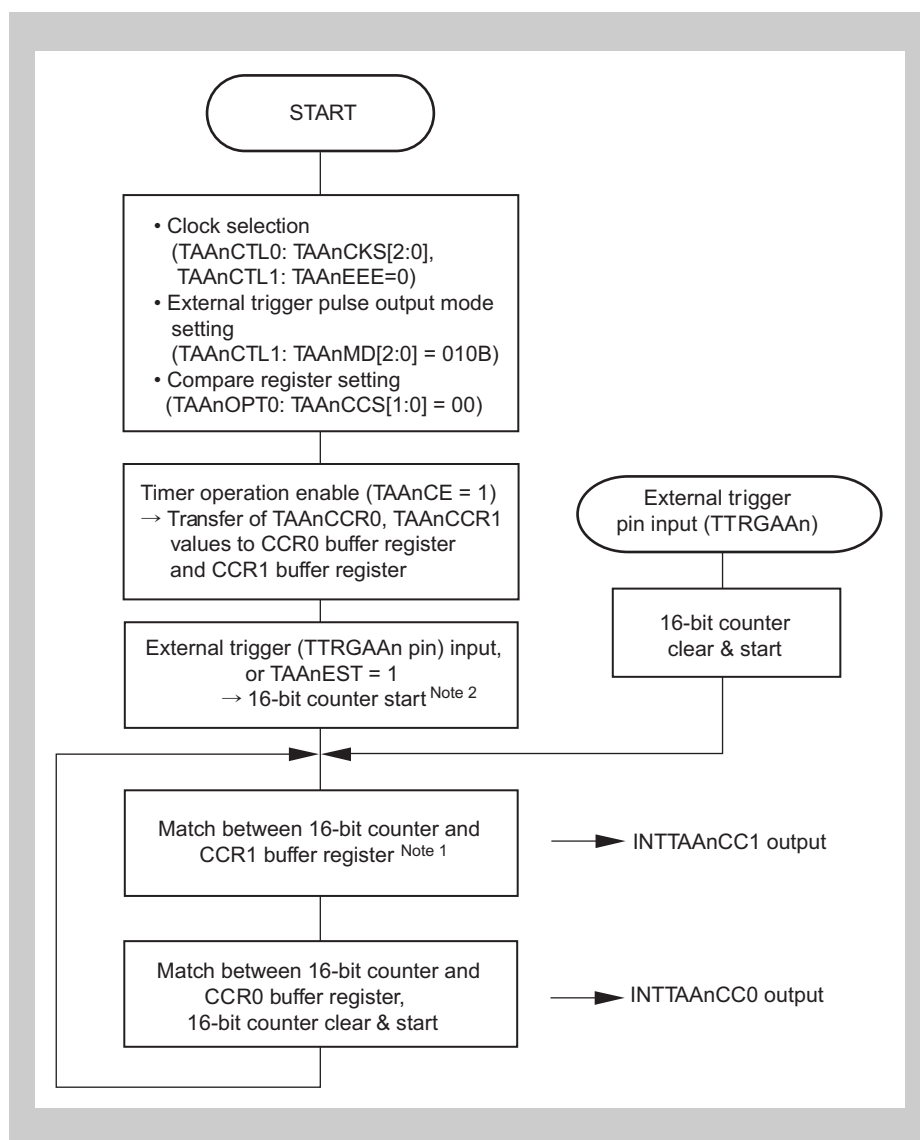
In the external trigger pulse mode, the capture function of the TAA<sub>n</sub>CCR0 and TAA<sub>n</sub>CCR1 registers cannot be used because these registers can be used only as compare registers.

---

**Caution** In the external trigger pulse mode, select the internal clock (TAA<sub>n</sub>EEE bit of TAA<sub>n</sub>CTL1 register = 0) for the count clock.

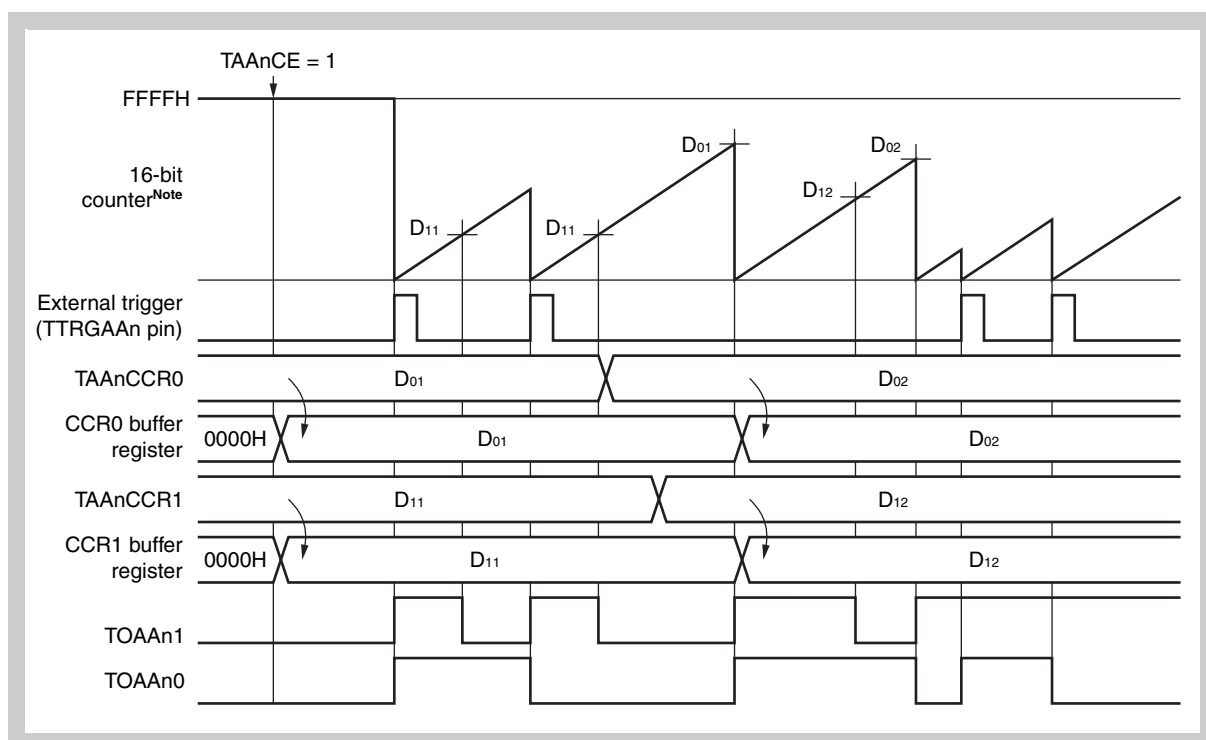
---

- Note**
1. For the reload operation when TAA<sub>n</sub>CCR0 and TAA<sub>n</sub>CCR1 are rewritten during timer operation, refer to “PWM mode (TAA<sub>n</sub>MD2 to TAA<sub>n</sub>MD0 = 100<sub>B</sub>)” on page 582.
  2. n = 0 to 9; m = 0, 1



**Figure 16-17** Flowchart of basic operation in external trigger pulse output mode

**Note** 1. The 16-bit counter is not cleared upon a match between the 16-bit counter and the CCR1 buffer register.



**Figure 16-18 Basic operation timing in external trigger pulse output mode**  
( $TAAAnOE0 = 1$ ,  $TAAAnOE1 = 1$ ,  $TAAAnOL0 = 0$ ,  $TAAAnOL1 = 0$ )

**Note** The 16-bit counter is not cleared when it matches the CCR1 buffer register.

$D_{01}$ ,  $D_{02}$ : Setting value of TAAAnCCR0 register (0000H to FFFFH)

$D_{11}$ ,  $D_{12}$ : Setting value of TAAAnCCR1 register (0000H to FFFFH)

Duty of TOAAAn1 output =

$(\text{Set value of TAAAnCCR1 register}) / (\text{Set value of TAA0CCR0 register})$

Cycle of TOAAAn1 output =

$(\text{Set value of TAAAnCCR0 register}) \times (\text{Count clock cycle})$

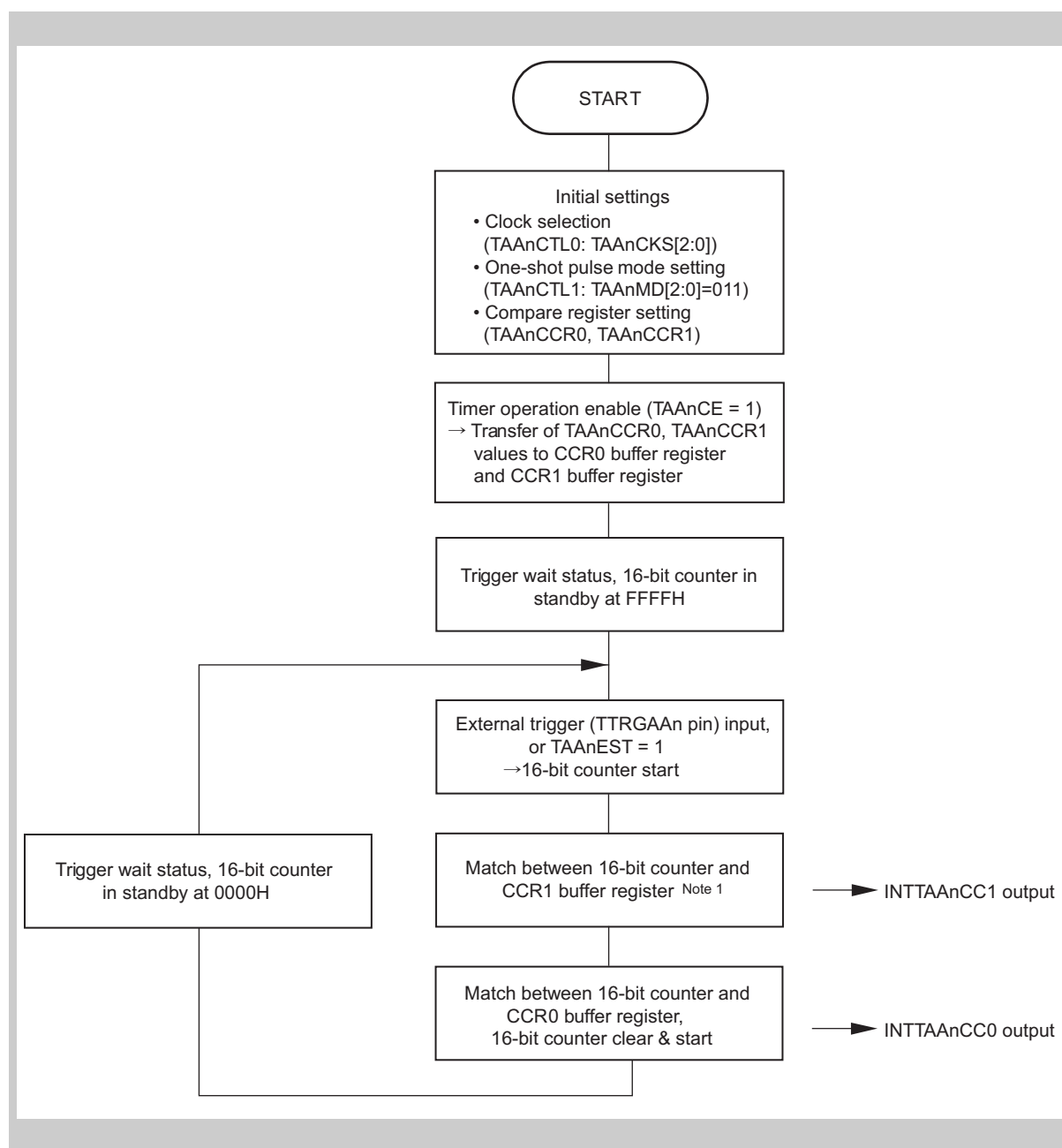
### 16.6.5 One-shot pulse mode (TAA<sub>n</sub>MD2 to TAA<sub>n</sub>MD0 = 011<sub>B</sub>)

When TAA<sub>n</sub>CE is set to 1 in the one-shot pulse mode, the 16-bit counter waits for the setting of the TAA<sub>n</sub>EST bit (to 1) or a trigger that is input when the edge of the TTRGA<sub>n</sub> pin is detected, while holding FFFFH. When the trigger is input, the 16-bit counter starts counting up.

When the value of the 16-bit counter matches the value of the CCR1 buffer register that has been transferred from the TAA<sub>n</sub>CCR1 register, TOAA<sub>n</sub>1 goes high. When the value of the 16-bit counter

matches the value of the CCR0 buffer register that has been transferred from the TAA<sub>n</sub>CCR0 register, TOAA<sub>n</sub>1 goes low, and the 16-bit counter is cleared to 0000H and stops. Input of a second or subsequent trigger is ignored while the 16-bit counter is operating. Be sure to input a second trigger while the 16-bit counter is stopped at 0000H. In the one shot pulse mode, rewriting the TAA<sub>n</sub>CCR0 and TAA<sub>n</sub>CCR1 registers is enabled when TAA<sub>n</sub>CE = 1. The set values of the TAA<sub>n</sub>CCR0 and TAA<sub>n</sub>CCR1 registers become valid after a write instruction from the CPU is executed. They are then transferred to the CCR0 and CCR1 buffer registers, and compared with the value of the 16-bit counter. The waveform of the one-shot pulse is output from the TOAA<sub>n</sub>1 pin. The TOAA<sub>n</sub>0 pin produces a toggle output when the value of the 16-bit counter matches the value of the TAA<sub>n</sub>CCR0 register. In the one-shot pulse mode, the TAA<sub>n</sub>CCR0 and TAA<sub>n</sub>CCR1 registers function only as compare registers. They cannot be used as capture registers.

- 
- Caution**
1. In the one-shot pulse mode, select the internal clock (TAA<sub>n</sub>EEE bit of TAA<sub>n</sub>CTL1 register = 0) as the count clock.
  2. In the one-shot pulse mode, it is prohibited to set the TAA<sub>n</sub>CCR1 register to 0000H.
-

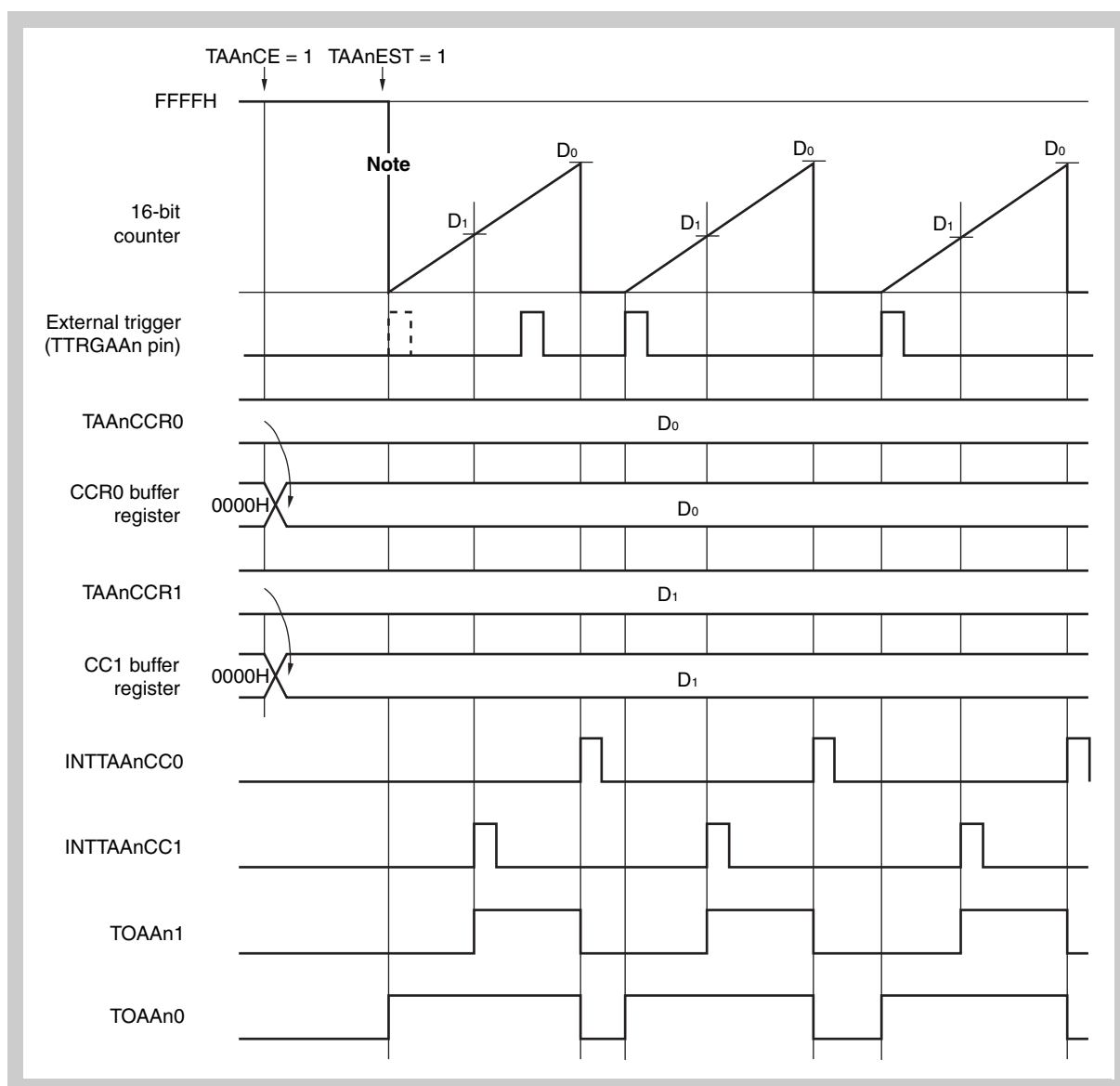


**Figure 16-19** Flowchart of basic operation in one-shot pulse mode

**Note** 1. The 16-bit counter is not cleared upon a match between the 16-bit counter and the CCR1 buffer register.

**Caution** The 16-bit counter is not cleared when a trigger input is performed during the count-up operation of the 16-bit counter.





**Figure 16-20** Timing of basic operation in one-shot pulse mode  
(TAAAnOE0 = 1, TAAAnOE1 = 1, TAAAnOL0 = 0, TAAAnOL1 = 0)

**Note** The 16-bit counter starts counting up when either TAAAnEST = 1 is set or the external trigger (TTRGAAn) is input.

D0: Setting value of TAAAnCCR0 register (0000H to FFFFH)

D1: Setting value of TAAAnCCR1 register (0000H to FFFFH)

### 16.6.6 PWM mode (TAA<sub>n</sub>MD2 to TAA<sub>n</sub>MD0 = 100<sub>B</sub>)

In the PWM mode, TAA<sub>n</sub> capture/compare register 1 (TAA<sub>n</sub>CCR1) is used to set the duty factor and TAA<sub>n</sub> capture/compare register 0 (TAA<sub>n</sub>CCR0) is used to set the cycle. By using these two registers and operating the timer, variable-duty PWM is output.

Rewriting the TAA<sub>n</sub>CCR0 and TAA<sub>n</sub>CCR1 registers is enabled when TAA<sub>n</sub>CE = 1.

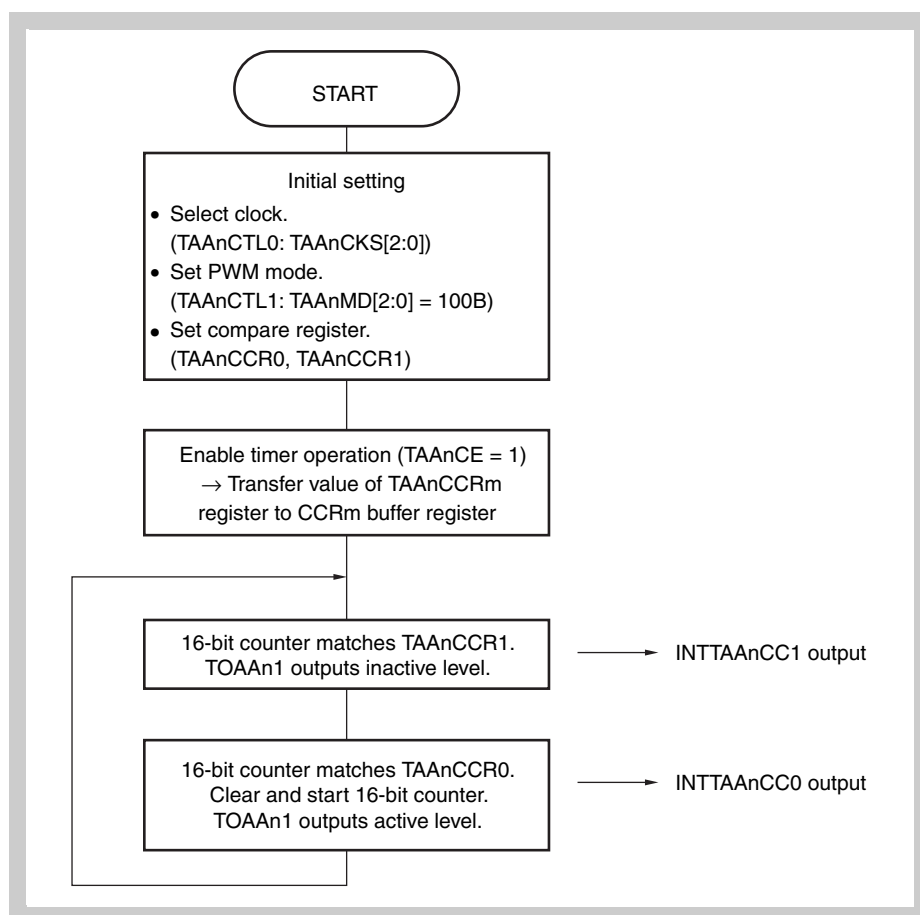
So that the set values of the TAA<sub>n</sub>CCR0 and TAA<sub>n</sub>CCR1 registers are compared with the value of the 16-bit counter (reloaded to the CCR0 and CCR1 buffer registers), the TAA<sub>n</sub>CCR0 register must be rewritten and then a value must be written to the TAA<sub>n</sub>CCR1 register before the value of the 16-bit counter matches the value of the TAA<sub>n</sub>CCR0 register.

The values of the TAA<sub>n</sub>CCR0 and TAA<sub>n</sub>CCR1 registers are reloaded when the value of the TAA<sub>n</sub>CCR0 register later matches the value of the 16-bit counter. Whether the next reload timing is made valid or not is controlled by writing to the TAA<sub>n</sub>CCR1 register. Therefore, write the same value to the TAA<sub>n</sub>CCR1 register even when only the value of the TAA<sub>n</sub>CCR0 register needs to be rewritten. Reload is invalid when only the value of the TAA<sub>n</sub>CCR0 register is rewritten.

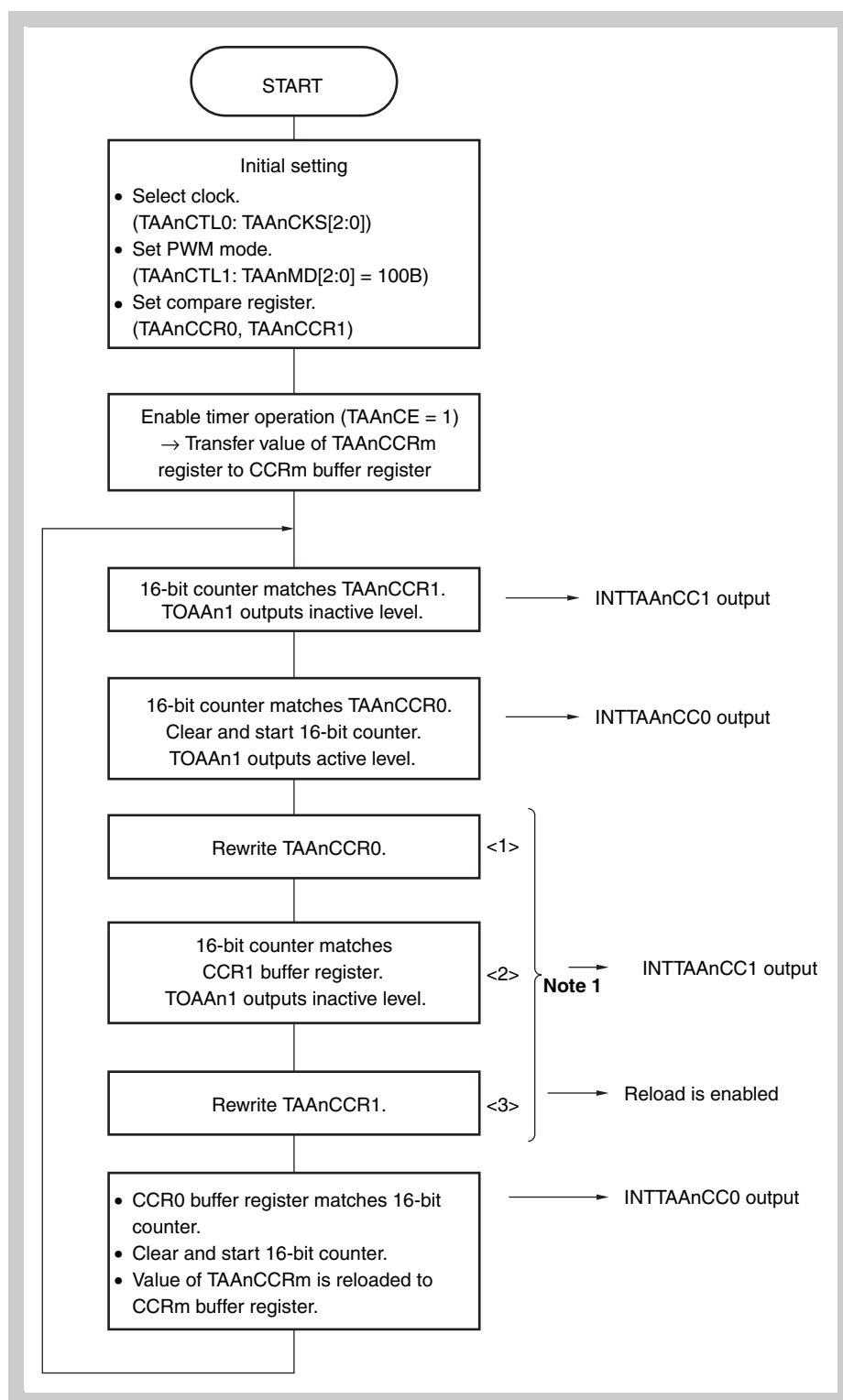
To stop timer AA, clear TAA<sub>n</sub>CE to 0.

The waveform of PWM is output from the TOAA<sub>n</sub>1 pin. The TOAA<sub>n</sub>0 pin produces a toggle output when the 16-bit counter matches the TAA<sub>n</sub>CCR0 register.

In the PWM mode, the TAA<sub>n</sub>CCR0 and TAA<sub>n</sub>CCR1 registers are used only as compare registers. They cannot be used as capture registers.

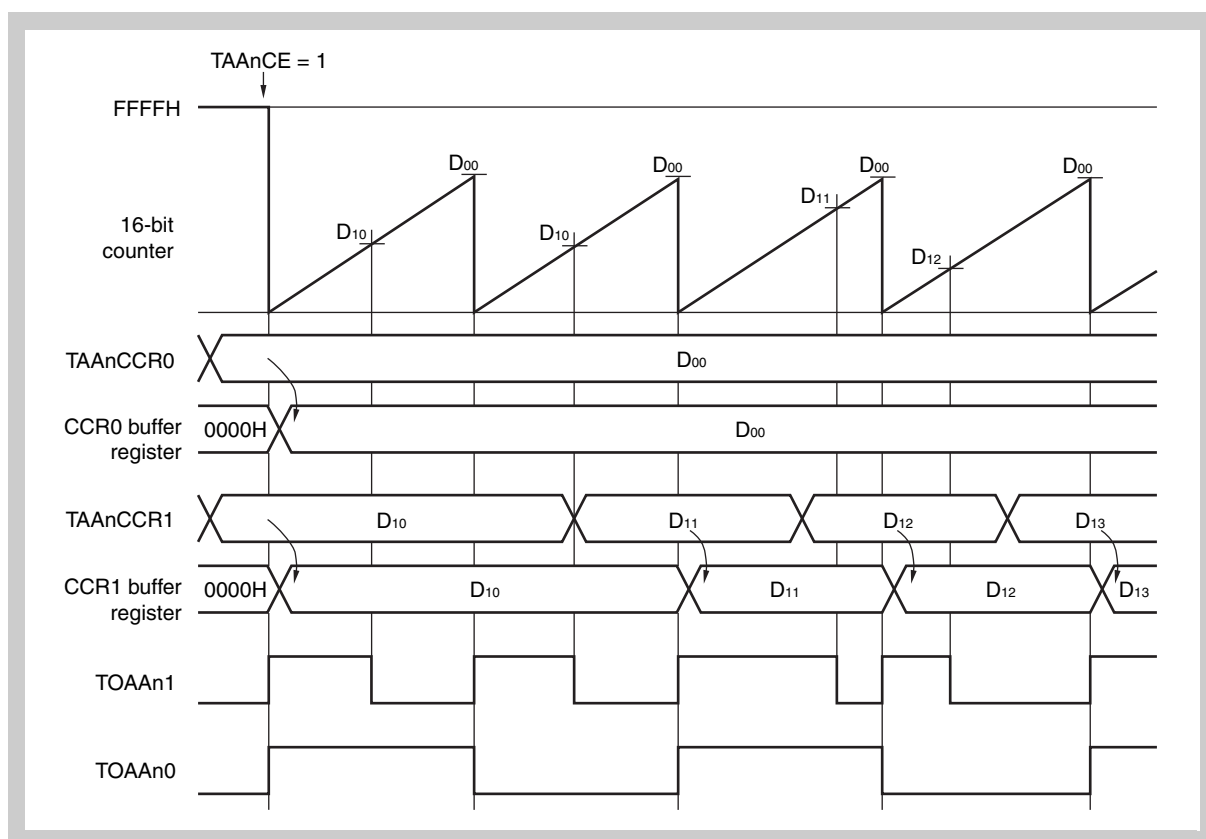


**Figure 16-21** Flowchart of basic operation in PWM mode  
When values of TAAAnCCR0, TAAAnCCR1 registers are not rewritten during timer operation



**Figure 16-22** Flowchart of basic operation in PWM mode  
When values of TAAAnCCR0, TAAAnCCR1 registers are rewritten during timer operation

- Note**
1. The timing of <2> in the above flowchart may differ depending on the rewrite timing of steps <1> and <3> and the value of TAAAnCCR1, but make sure that step <3> comes after step <1>.
  2.  $n = 0$  to 9;  $m = 0, 1$



**Figure 16-23 Basic Operation Timing in PWM mode**  
**When rewriting TAAAnCCR1 value**  
 (TAAAnOE0 = 1, TAAAnOE1 = 1, TAAAnOL0 = 0, TAAAnOL1 = 0)

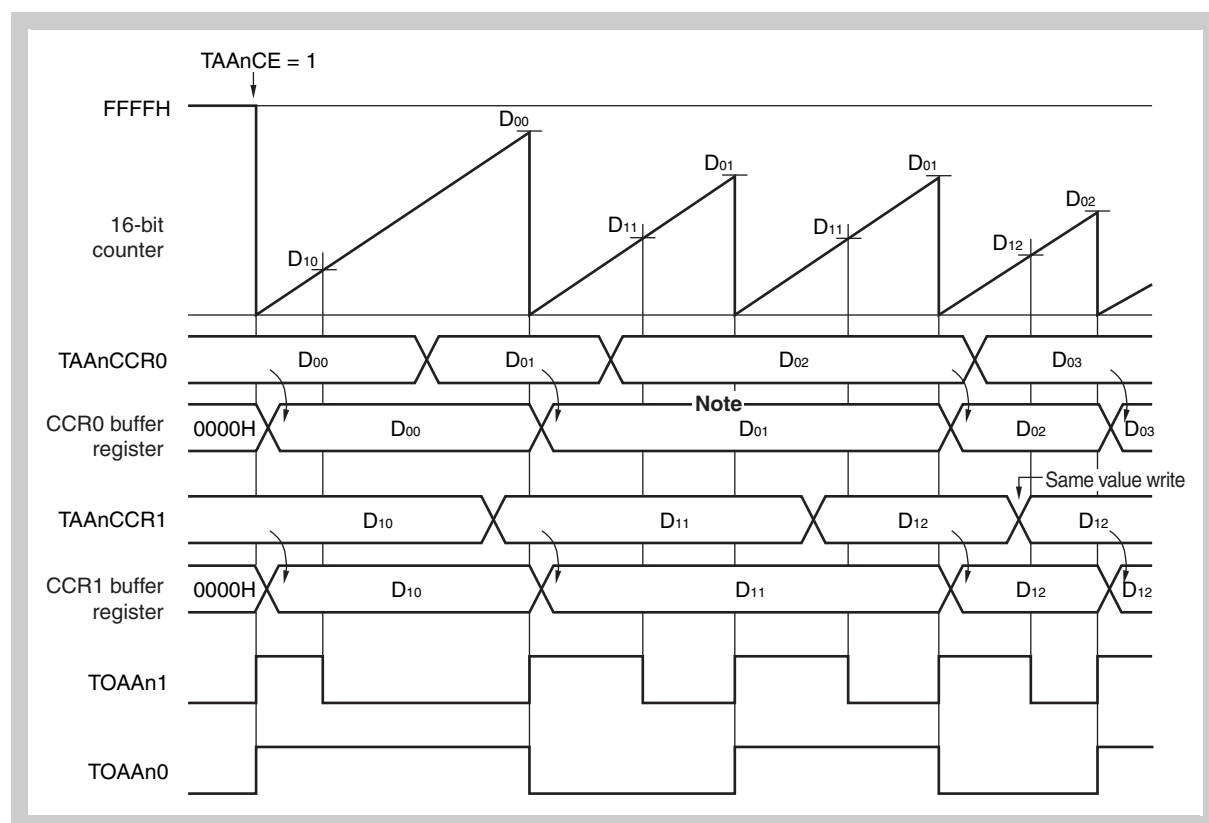
D00: Set value of TAAAnCCR0 register (0000H to FFFFH)  
 D10, D11, D12, D13: Set value of TAAAnCCR1 register (0000H to FFFFH)

Duty factor of TOAAAn1 output =  

$$\frac{\text{Set value of TAAAnCCR1 register}}{\text{Set value of TAAAnCCR0 register} + 1}$$
  
 Cycle of TOAAAn1 output =  

$$(\text{Set value of TAAAnCCR0 register} + 1) \times (\text{Count clock cycle})$$
  
 Toggle width of TOAAAn0 output =  

$$(\text{Set value of TAAAnCCR0 register} + 1) \times (\text{Count clock cycle})$$



**Figure 16-24 Basic operation timing in PWM mode**  
 When TAAAnCCR0, TAAAnCCR1 values are rewritten  
 (TAAAnOE0 = 1, TAAAnOE1 = 1, TAAAnOL0 = 0, TAAAnOL1 = 0)

**Note** Reload is not performed because the TAAAnCCR1 register was not rewritten.

D00, D01, D02, D03: Setting values of TAAAnCCR0 register (0000H to FFFFH)  
 D10, D11, D12, D13: Setting values of TAAAnCCR1 register (0000H to FFFFH)

Duty factor of TOAAAn1 output =  

$$\frac{\text{Set value of TAAAnCCR1 register}}{\text{Set value of TAAAnCCR0 register} + 1}$$
  
 Cycle of TOAAAn1 output =  

$$(\text{Set value of TAAAnCCR0 register} + 1) \times (\text{Count clock cycle})$$
  
 Toggle width of TOAAAn0 output =  

$$(\text{Set value of TAAAnCCR0 register} + 1) \times (\text{Count clock cycle})$$

### 16.6.7 Free-running mode (TAA<sub>n</sub>MD2 to TAA<sub>n</sub>MD0 = 101<sub>B</sub>)

In the free-running mode, both the interval function and the compare function can be realized by operating the 16-bit counter as a free-running counter and selecting capture/compare operation with the TAA<sub>n</sub>CCS1 and TAA<sub>n</sub>CCS0 bits.

The settings of the TAA<sub>n</sub>CCS1 and TAA<sub>n</sub>CCS0 bits of the TAA<sub>n</sub>OPT0 register are valid only in the free-running mode.

TAA <sub>n</sub> CCS1	Operation
0	Use TAA <sub>n</sub> CCR1 register as compare register
1	Use TAA <sub>n</sub> CCR1 register as capture register

TAA <sub>n</sub> CCS0	Operation
0	Use TAA <sub>n</sub> CCR0 register as compare register
1	Use TAA <sub>n</sub> CCR0 register as capture register

- Using TAA<sub>n</sub>CCR1 register as compare register  
An interrupt is output upon a match between the 16-bit counter and the CCR1 buffer register in the free-running mode (interval function).  
Rewrite during compare timer operation is enabled and performed with any time write mode. (Once the compare value has been written, synchronization with the internal clock is done and this value is used as the 16-bit counter comparison value.)  
When timer output (TOAA<sub>n</sub>1) has been enabled, TOAA<sub>n</sub>1 performs toggle output upon a match between the 16-bit counter and the CCR1 buffer register.
- Using TAA<sub>n</sub>CCR1 register as capture register  
The value of the 16-bit counter is saved to the TAA<sub>n</sub>CCR1 register upon TIAA<sub>n</sub>1 pin edge detection.
- Using TAA<sub>n</sub>CCR0 register as compare register  
An interrupt is output upon a match between the 16-bit counter and the CCR0 buffer register in the free-running mode (interval function).  
Rewrite during compare timer operation is enabled and performed with any time write mode. (Once the compare value has been written, synchronization with the internal clock is done and this value is used as the 16-bit counter comparison value.)  
When timer output (TOAA<sub>n</sub>0) has been enabled, TOAA<sub>n</sub>0 performs toggle output upon a match between the 16-bit counter and the CCR0 buffer register.
- Using TAA<sub>n</sub>CCR0 register as capture register  
The value of the 16-bit counter is saved to the TAA<sub>n</sub>CCR0 register upon TIAA<sub>n</sub>0 pin edge detection.

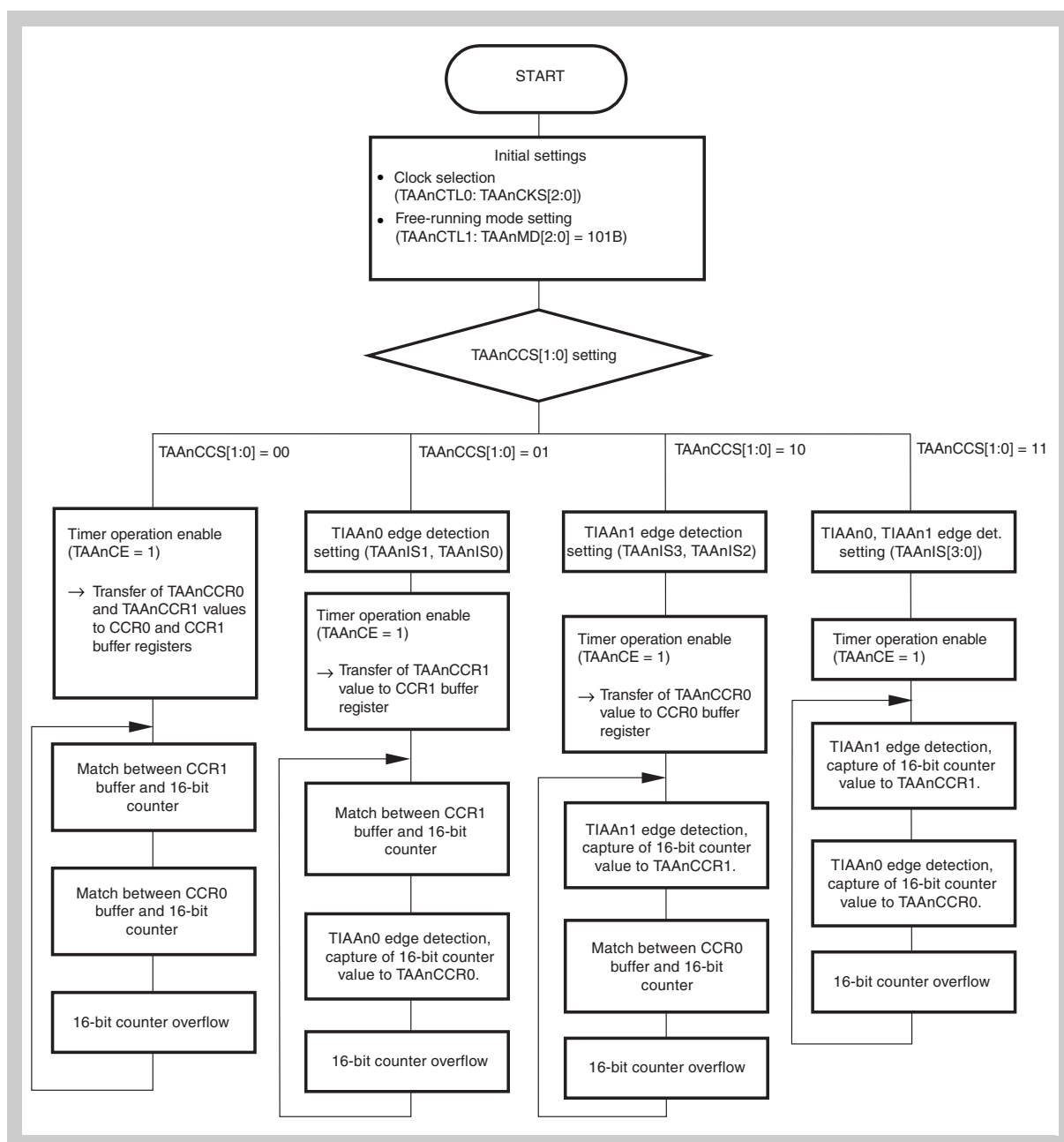


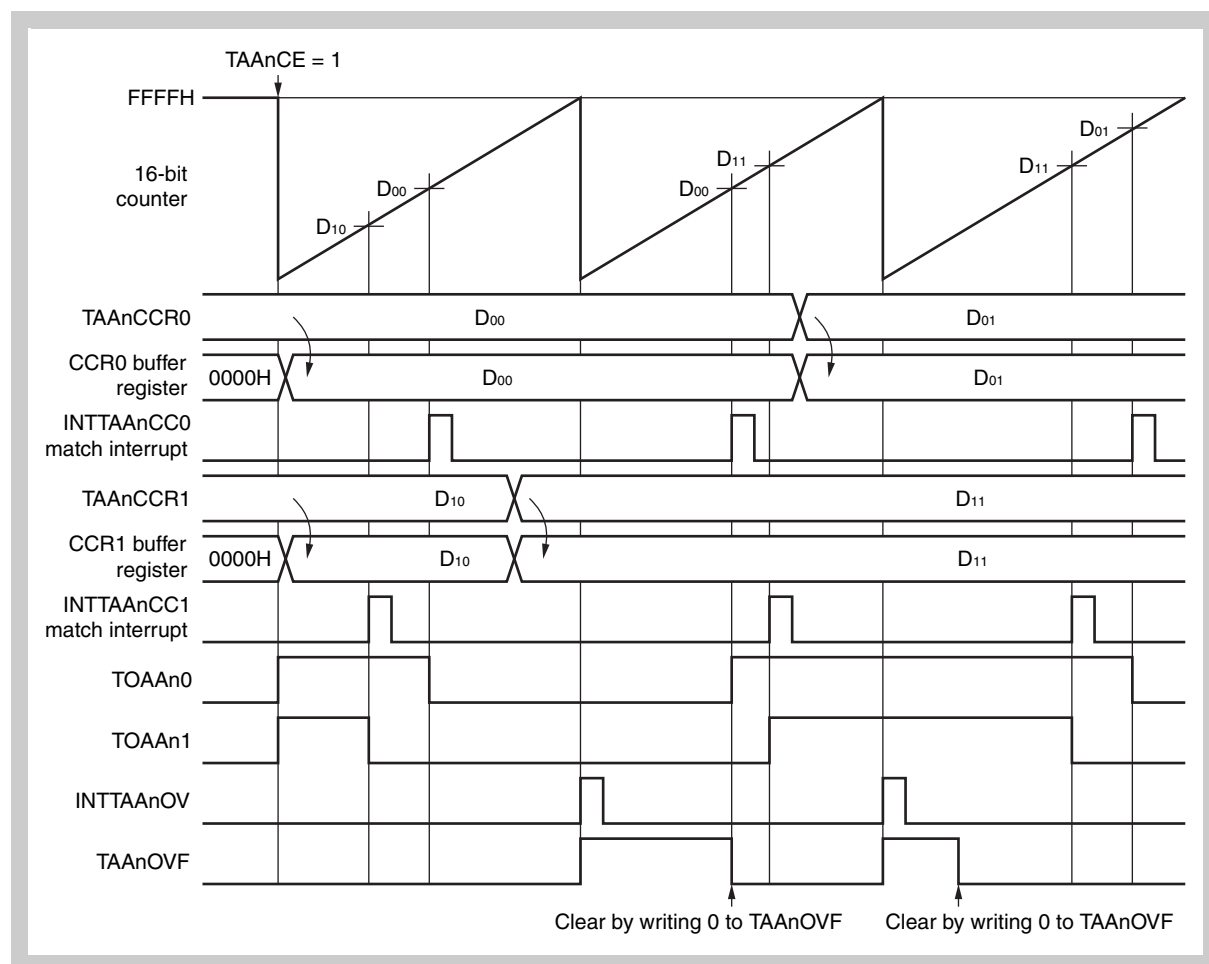
Figure 16-25 Flowchart of basic operation in free-running mode



**(1) When TAA<sub>n</sub>CCS1 = 0, and TAA<sub>n</sub>CCS0 = 0 settings (interval function description, compare function)**

When TAA<sub>n</sub>CE = 1 is set, the 16-bit counter counts from 0000H to FFFFH and the free-running count-up operation continues until TAA<sub>n</sub>CE = 0 is set.

In this mode, when a value is written to the TAA<sub>n</sub>CCR0 and TAA<sub>n</sub>CCR1 registers, they are transferred to the CCR0 buffer register and the CCR1 buffer register (any time write mode). In this mode, no one-shot pulse is output even when an one-shot pulse trigger is input. Moreover, when TAA<sub>n</sub>OEm = 1 is set, TOAA<sub>n</sub>m performs toggle output upon a match between the 16-bit counter and the CCR<sub>m</sub> buffer register.



**Figure 16-26 Basic Operation Timing In Free-running Mode**  
 (TAA<sub>n</sub>CCS1 = 0, TAA<sub>n</sub>CCS0 = 0)  
 (TAA<sub>n</sub>OEm = 1, TAA<sub>n</sub>OL0 = 0, TAA<sub>n</sub>OL1 = 0)

D00, D01: Setting values of TAA<sub>n</sub>CCR0 register (0000H to FFFFH)

D10, D11: Setting values of TAA<sub>n</sub>CCR1 register (0000H to FFFFH)

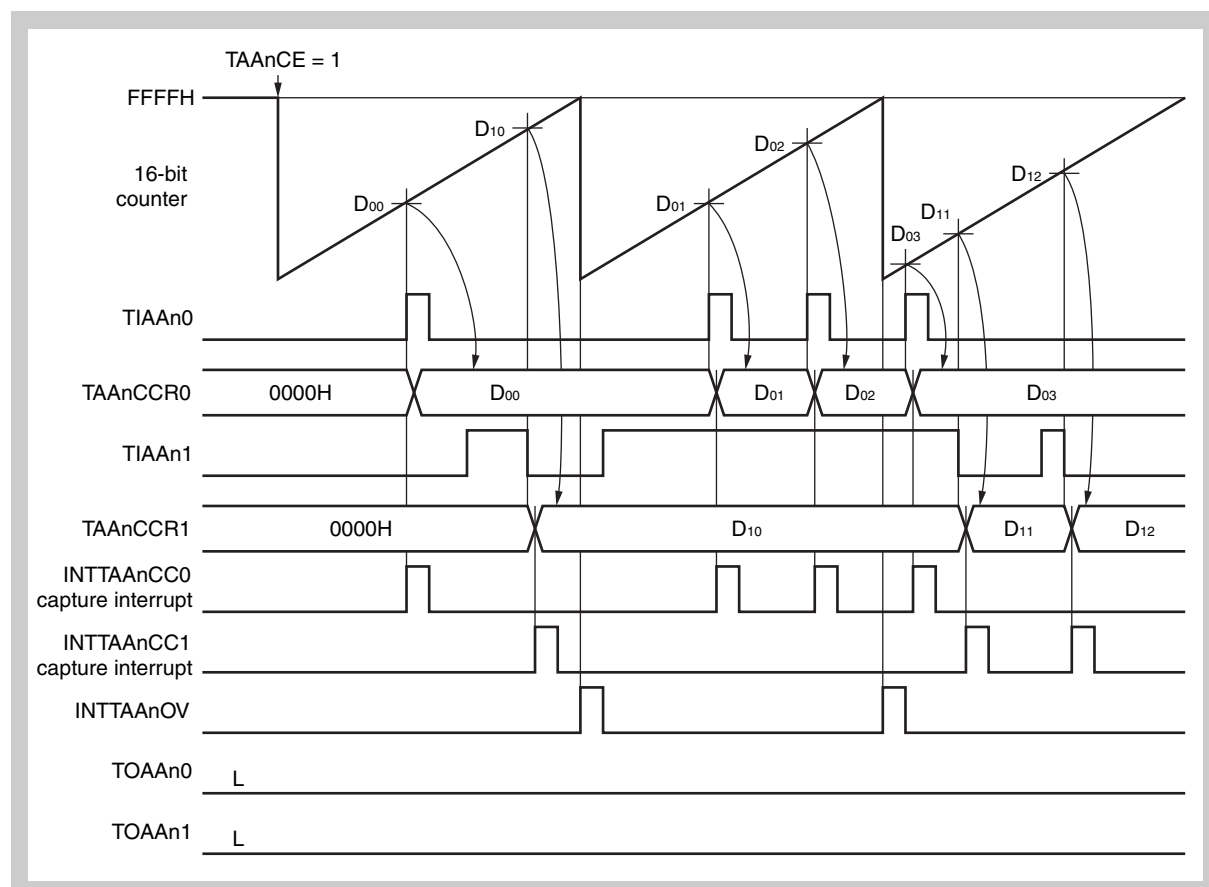
TOAA<sub>n</sub>m output goes high when counting is started.

**Note** n = 0 to 9; m = 0, 1

## (2) When TAA<sub>n</sub>CCS1 = 1 and TAA<sub>n</sub>CCS0 = 1 settings (capture function description)

When TAA<sub>n</sub>CE = 1, the 16-bit counter counts from 0000H to FFFFH and free-running count-up operation continues until TAA<sub>n</sub>CE = 0 is set. During this time, values are captured by capture trigger operation and are written to the TAA<sub>n</sub>CCR0 and TAA<sub>n</sub>CCR1 registers.

Regarding capture close to the overflow (FFFFH), judgment is made using the overflow flag (TAA<sub>n</sub>OVF). However, if overflow occurs twice (two or more free-running cycles), the capture trigger interval cannot be judged with the TAA<sub>n</sub>OVF flag. In this case, the system should be revised.



**Figure 16-27 Basic operation timing in free-running mode**  
 (TAA<sub>n</sub>CCS1 = 1, TAA<sub>n</sub>CCS0 = 1)  
 (TAA<sub>n</sub>OE0 = 1, TAA<sub>n</sub>OE1 = 1, TAA<sub>n</sub>OL0 = 0, TAA<sub>n</sub>OL1 = 0)

D00, D01, D02, D03:

Values captured to TAA<sub>n</sub>CCR0 register (0000H to FFFFH)

D10, D11, D12:

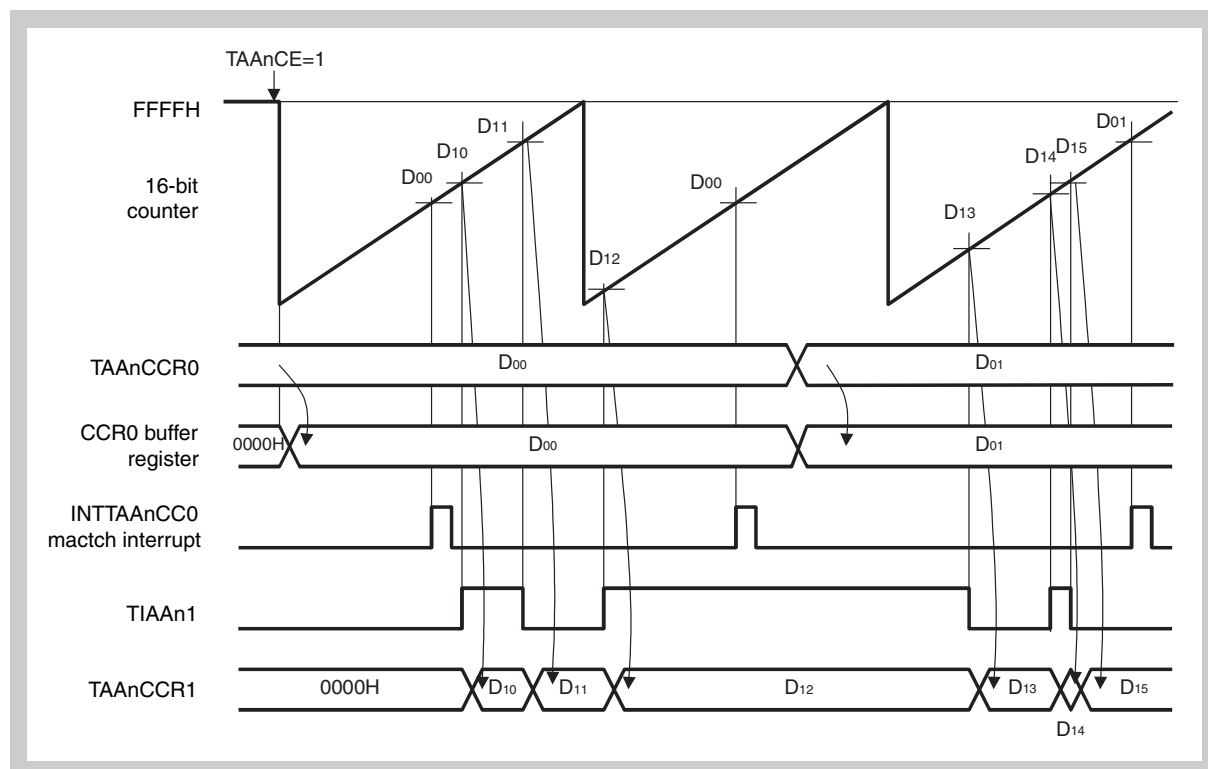
Values captured to TAA<sub>n</sub>CCR1 register (0000H to FFFFH)

TIAAn0: Set to rising edge detection (TAA<sub>n</sub>IS1, TAA<sub>n</sub>IS0 = 01)

TIAAn1: Set to falling edge detection (TAA<sub>n</sub>IS3, TAA<sub>n</sub>IS2 = 10)

**(3) When TAA<sub>n</sub>CCS1 = 1 and TAA<sub>n</sub>CCS0 = 0**

When TAA<sub>n</sub>CE = 1 is set, the counter counts from 0000H to FFFFH and free-running count-up operation continues until TAA<sub>n</sub>CE = 0 is set. The TAA<sub>n</sub>CCR0 register is used as a compare register. An interrupt signal is output upon a match between the value of the 16-bit counter and the setting value transferred to the CCR0 buffer register from the TAA<sub>n</sub>CCR0 register as an interval function. Even if TAA<sub>n</sub>OE1 = 1 to realize the output function, TAA<sub>n</sub>CCR1 register cannot control TOAA<sub>n</sub>1 because it is used as capture register.



**Figure 16-28 Basic operation timing in free-running mode**  
 (TAA<sub>n</sub>CCS1 = 1, TAA<sub>n</sub>CCS0 = 0)  
 (TAA<sub>n</sub>OE0 = 1, TAA<sub>n</sub>OE1 = 1, TAA<sub>n</sub>OL0 = 0, TAA<sub>n</sub>OL1 = 0)

D00, D01:

Setting compare values of TAA<sub>n</sub>CCR0 register (0000H to FFFFH)

D10, D11, D12, D13, D14, D15:

Values captured to TAA<sub>n</sub>CCR1 register (0000H to FFFFH)

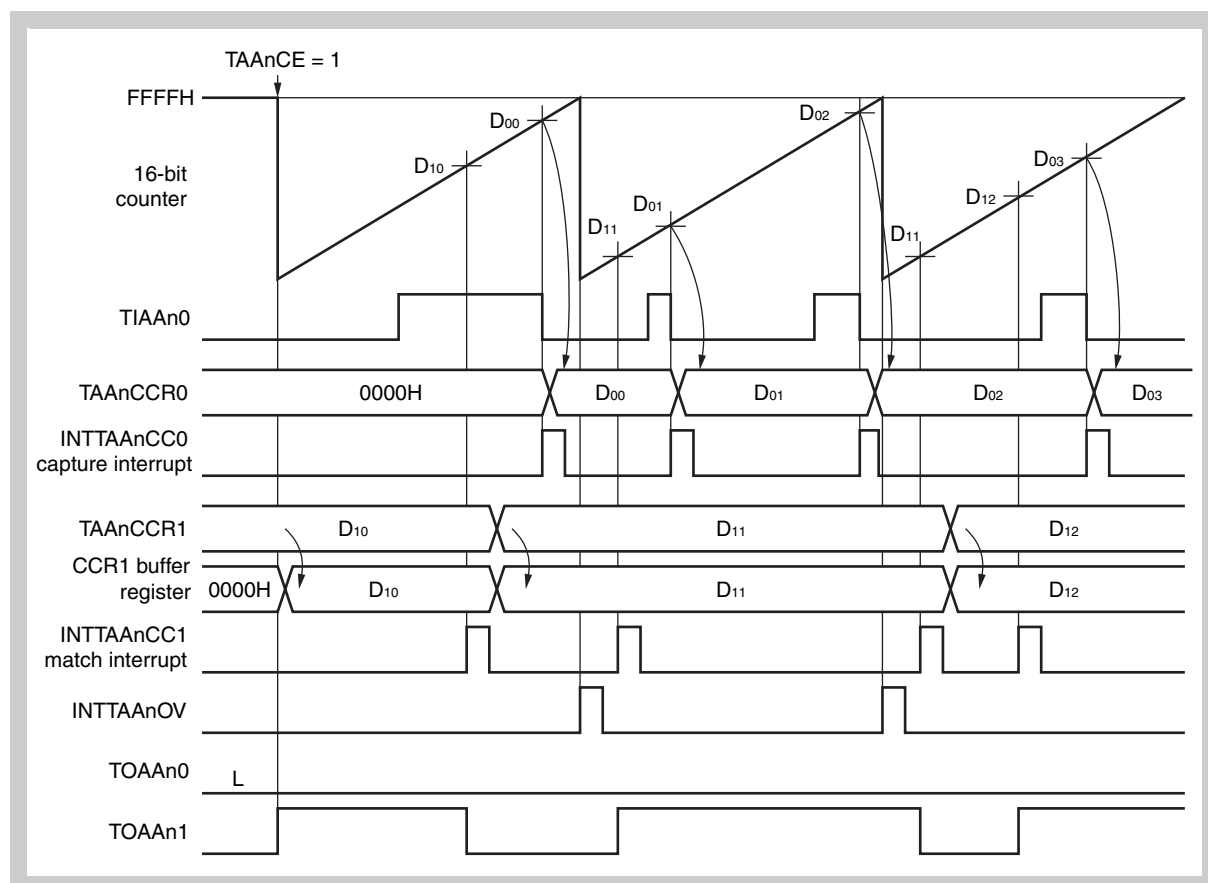
TIAAn1:

Set to detection of both rising and falling edges (TAA<sub>n</sub>IS3, TAA<sub>n</sub>IS2 = 11)

**(4) When TAA<sub>n</sub>CCS1 = 0 and TAA<sub>n</sub>CS0 = 1**

When TAA<sub>n</sub>CE is set to 1, the 16-bit counter counts from 0000H to FFFFH and free-running count-up operation continues until TAA<sub>n</sub>CE = 0 is set. The TAA<sub>n</sub>CCR1 register is used as a compare register. An interrupt signal is output upon a match between the value of the 16-bit counter and the setting value of the TAA<sub>n</sub>CCR1 register as an interval function. When TAA<sub>n</sub>OE1 = 1 is set, TOAA<sub>n</sub>1 performs toggle output upon match between the value of the 16-bit counter and the setting value of the TAA<sub>n</sub>CCR1 register.

Even if TAA<sub>n</sub>OE0 = 1 to realize the output function, TAA<sub>n</sub>CCR0 register cannot control TOAA<sub>n</sub>0 because it is used as capture register.



**Figure 16-29 Basic operation timing in free-running mode**  
 (TAA<sub>n</sub>CCS1 = 0, TAA<sub>n</sub>CCS0 = 1)  
 (TAA<sub>n</sub>OE0 = 1, TAA<sub>n</sub>OE1 = 1, TAA<sub>n</sub>OL0 = 0, TAA<sub>n</sub>OL1 = 0)

D00, D01, D02, D03:

Values captured to TAA<sub>n</sub>CCR0 register (0000H to FFFFH)

D10, D11, D12:

Setting compare value of TAA<sub>n</sub>CCR1 register (0000H to FFFFH)

TIAA<sub>n</sub>0: Set to falling edge detection (TAA<sub>n</sub>IS1, TAA<sub>n</sub>IS0 = 10)

**(5) Overflow flag**

When the counter overflows from FFFFH to 0000H in the free-running mode, the overflow flag (TAA<sub>n</sub>OVF) is set to 1 and an overflow interrupt (INTTAA<sub>n</sub>OV) is output.

The overflow flag is cleared by the CPU when writing 0 to it.

### 16.6.8 Pulse width measurement mode (TAAAnMD2 to TAAAnMD0 = 110B)

In the pulse width measurement mode, free-running count is performed. The value of the 16-bit counter is saved to capture register 0 (TAAAnCCR0), or capture register 1 (TAAAnCCR1) respectively, and the 16-bit counter is cleared upon edge detection of the TIAAn0 pin, or TIAAn1 respectively. The external input pulse width can be measured as a result.

However, when measuring a large pulse width that exceeds 16-bit counter overflow, perform judgment with the overflow flag. Since measurement of pulses for which overflow occurs twice or more is not possible, adjust the operating frequency of the 16-bit counter.

Depending on the selected capture input sources and specified edge detection three different measurement methods can be applied.

1. Pulse period measurement
2. Alternating pulse width and pulse space measurement:  
This requires a fast interrupt handling, in order to measure pulse width and pulse space correctly.
3. Simultaneous pulse width and pulse space measurement:  
Both capture inputs are required to measure pulse width and pulse space simultaneously.

The measurements methods are explained in the following sub-chapters.

---

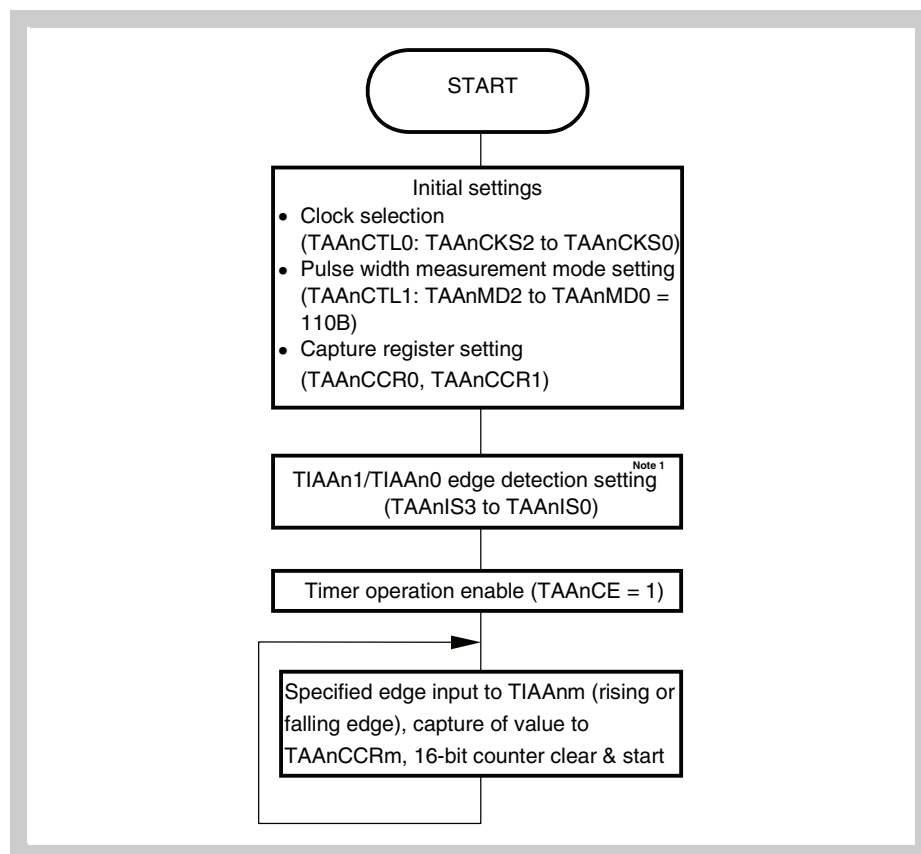
**Caution** In the pulse width measurement mode, select the internal clock (TAAAnEEE of TAAAnCTL1 register = 0).

---

**(1) Pulse period measurement**

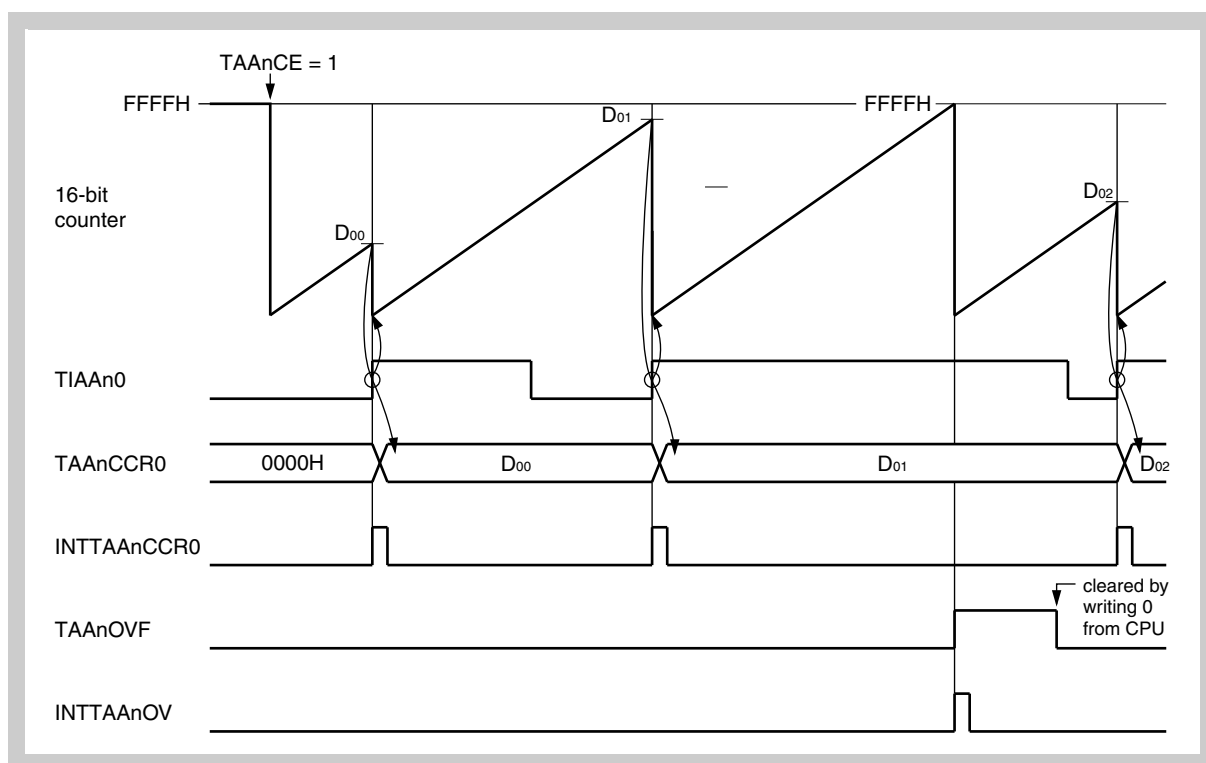
The pulse period of a signal can be measured in the pulse width measurement mode, when the edge detection of one of the inputs TIAAn0 and TIAAn1 is set either to “rising edge” or “falling edge”. The detection of the other input should be set to “no edge detection”.

By detection of the specified edge the resulting value is captured in the corresponding capture register (TAAAnCCR0 or TAAAnCCR1), and the timer is cleared and restarts counting.



**Figure 16-30** Flowchart of pulse period measurement

- Note**
1. External pulse input is possible for both TIAAn0 and TIAAn1, but only one should be selected for the pulse period measurement. Specify either “rising edge” or “falling edge” for edge detection. Specify the edge of the external input pulse that is not used as “no edge detection”.
  2.  $n = 0$  to  $9$ ;  $m = 0, 1$



**Figure 16-31 Basic operation timing of pulse period measurement**

D<sub>00</sub>, D<sub>01</sub>, D<sub>02</sub>: Values captured to TAAAnCCR0 register (0000H to FFFFH)

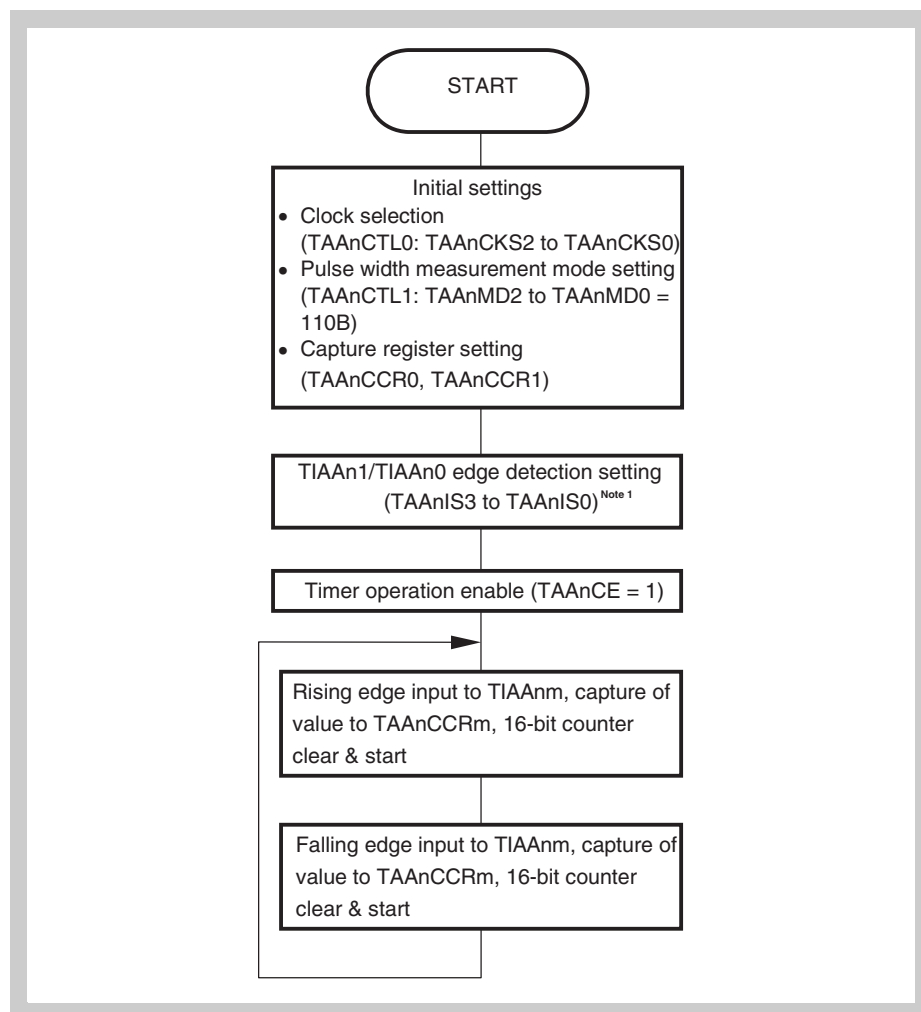
TIAAn0: Set to detection of rising edge (TAAAnIS1, TAAAnIS0 = 01B)

TIAAn1: Set to no edge detection (TAAAnIS3, TAAAnIS2 = 00B)

**(2) Alternating pulse width and pulse space measurement**

The pulse period of a signal can be measured in the pulse width measurement mode alternating in one capture register, when the edge detection of one of the inputs TIAAn0 and TIAAn1 is set to “both rising and falling edges”. The detection of the other input should be set to “no edge detection”.

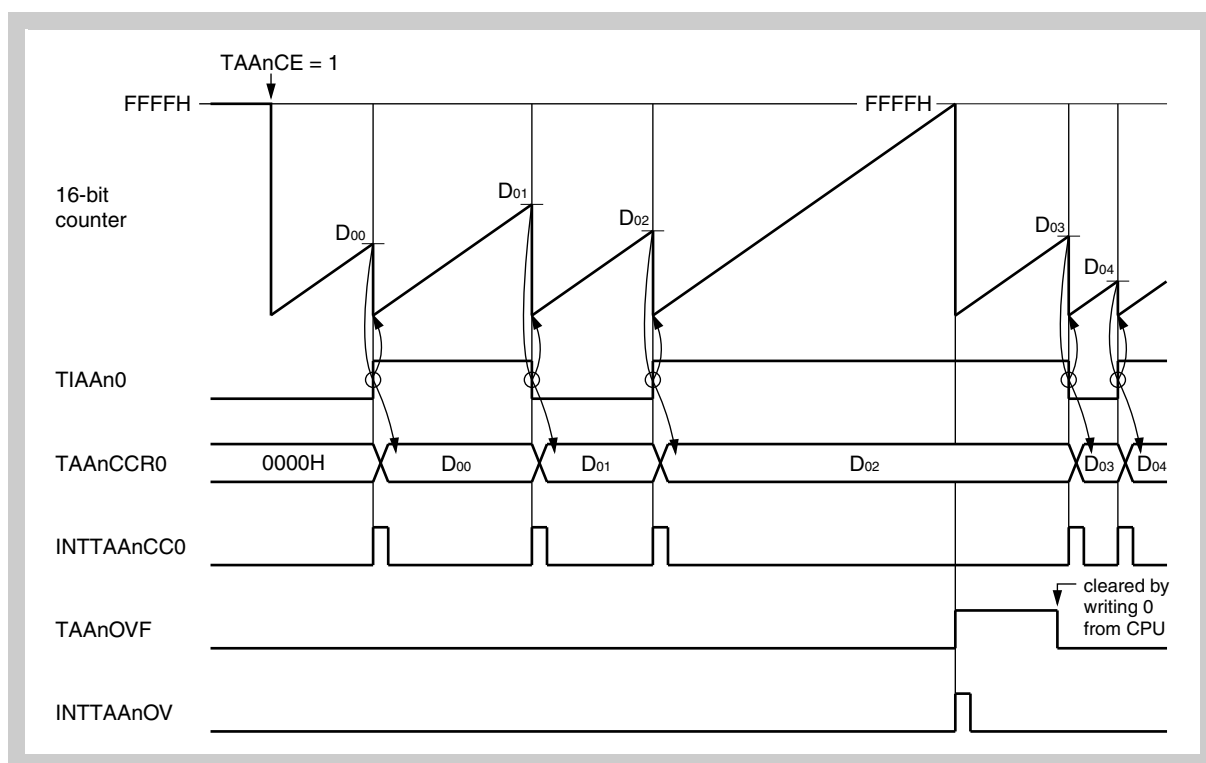
By detection of a falling or rising edge the resulting value is captured in the corresponding capture register (TAAAnCCR0 or TAAAnCCR1), and the timer is cleared and restarts counting.



**Figure 16-32** Flowchart of alternating pulse width and pulse space measurement

- Note**
1. External pulse input is possible for both TIAAn0 and TIAAn1, but only one should be selected for the alternating pulse width and pulse space measurement.  
Specify “both rising and the falling edges” for edge detection. Specify the edge of the external input pulse that is not used as “no edge detection”.
  2.  $n = 0$  to 9;  $m = 0, 1$





**Figure 16-33 Basic operation timing of alternating pulse width and pulse space measurement**

D<sub>00</sub>, D<sub>01</sub>, D<sub>02</sub>, D<sub>03</sub>, D<sub>04</sub>:

Values captured to TAAAnCCR0 register (0000H to FFFFH)

TIAAn0:

Set to detection of both rising and falling edges (TAAAnIS1, TAAAnIS0 = 11B)

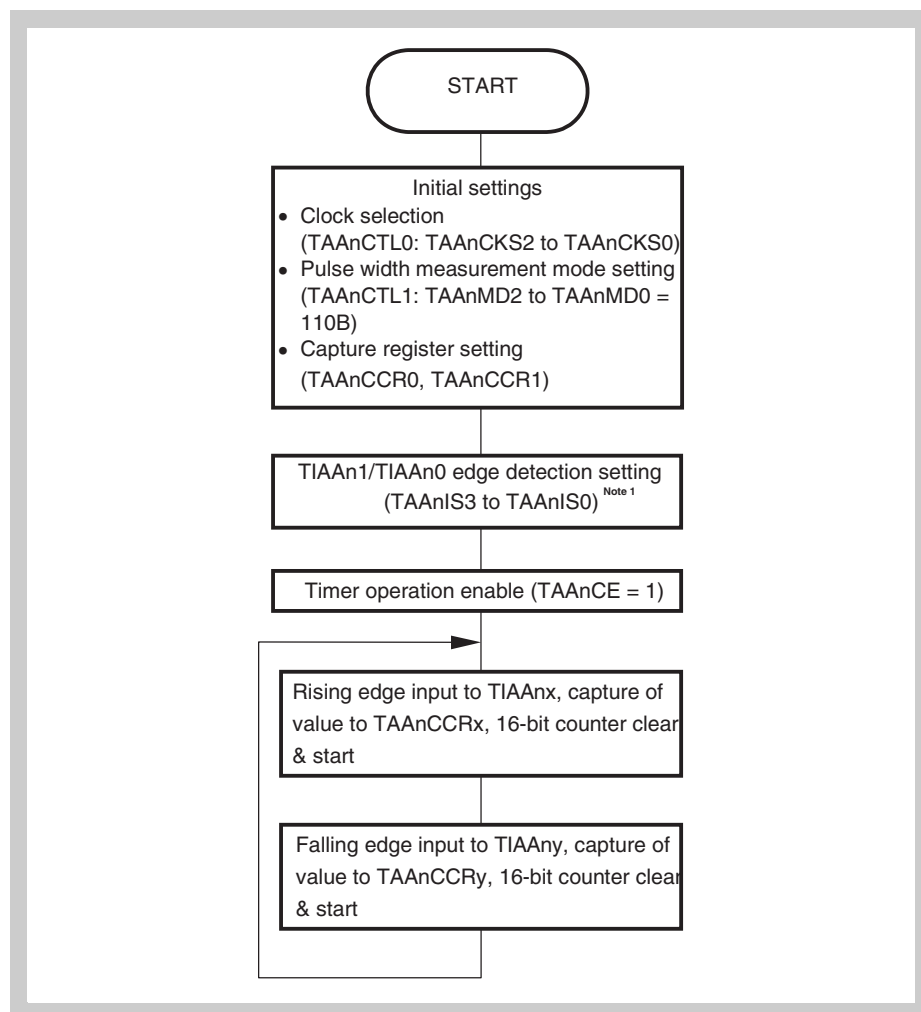
TIAAn1:

Set to no edge detection (TAAAnIS3, TAAAnIS2 = 00B)

**(3) Simultaneous pulse width and pulse space measurement**

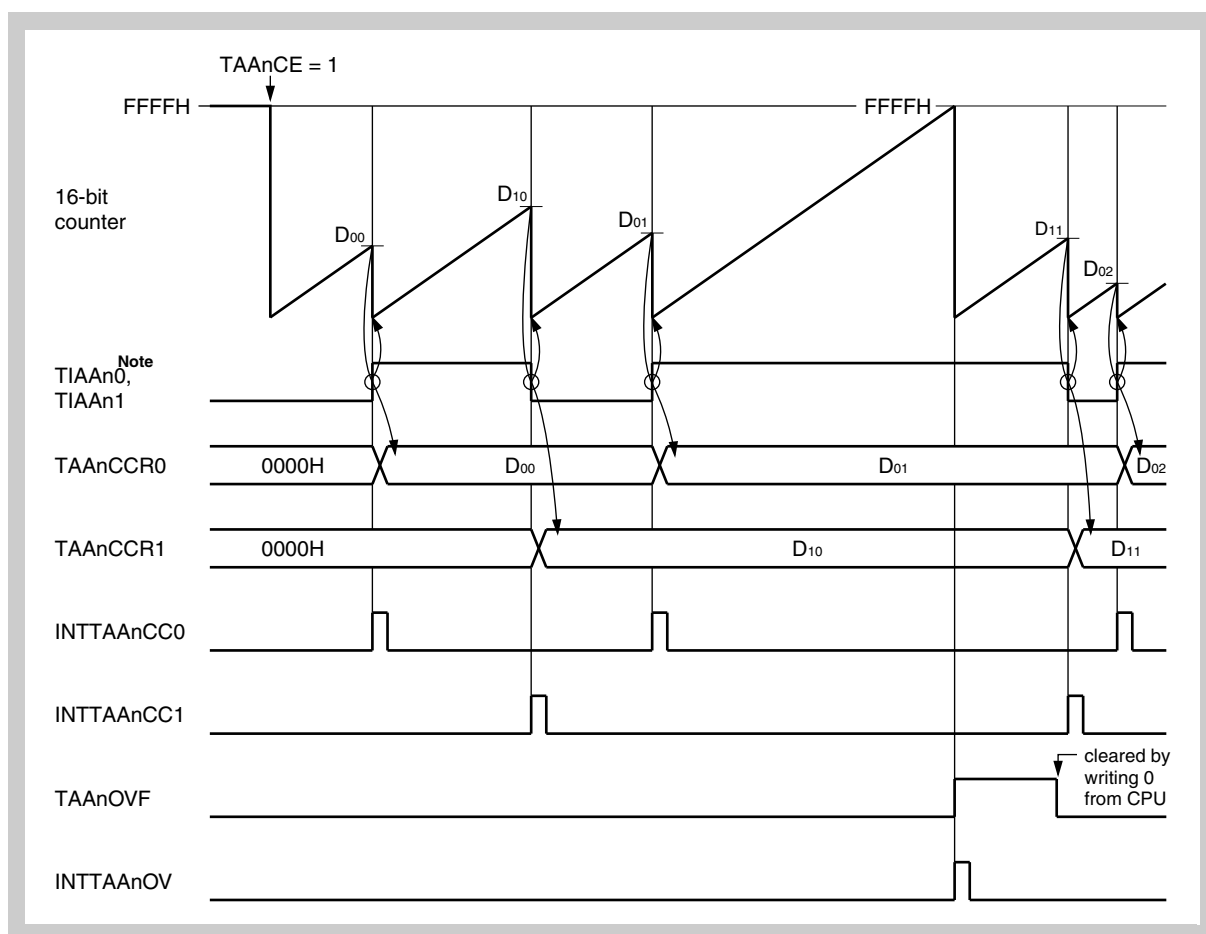
Pulse width and pulse space can be measured simultaneously in the pulse width measurement mode, when the signal is input to both inputs TIAAn0 and TIAAn1, where both inputs detect opposite edges. Alternatively the signal can be input to TIAAn0 only, when the capture source input selection for capture register 1 is used (refer to “TAAIC0 - Timer input control register 0” on page 549 and “TAAIC1 - Timer input control register 1” on page 550 and “TAAIC2 - Timer input control register 2” on page 551).

By detection of the specified edge the resulting values of pulse width or pulse space are captured in the corresponding capture registers (TAAAnCCR0, TAAAnCCR1), and the timer is cleared and restarts counting.



**Figure 16-34** Flowchart of simultaneous pulse width and pulse space measurement

- Note**
1. External pulse input must be input to both TIAAn0 and TIAAn1, or to TIAAn0 only, if the internal connection between both inputs is selected. Specify “rising edge” for edge detection of first input, and “falling edge” for the second input, or vice versa.
  2.  $n = 0$  to  $9$ ;  $x = 0, 1$   
 $y = 0$  when  $x = 1$ ;  $y = 1$  when  $x = 0$



**Figure 16-35 Basic operation timing of simultaneous pulse width and pulse space measurement**

**Note** The signal to measure has to be assigned to both inputs, TIAAn0 and TIAAn1. This can be done either by external pin connection, or internally when selecting TIAAn1 input on TIAAn0 pin. In case of internal connection the signal has to be input on TIAAn0 pin.

D<sub>00</sub>, D<sub>01</sub>, D<sub>02</sub>: Values captured to TAAAnCCR0 register (0000H to FFFFH)

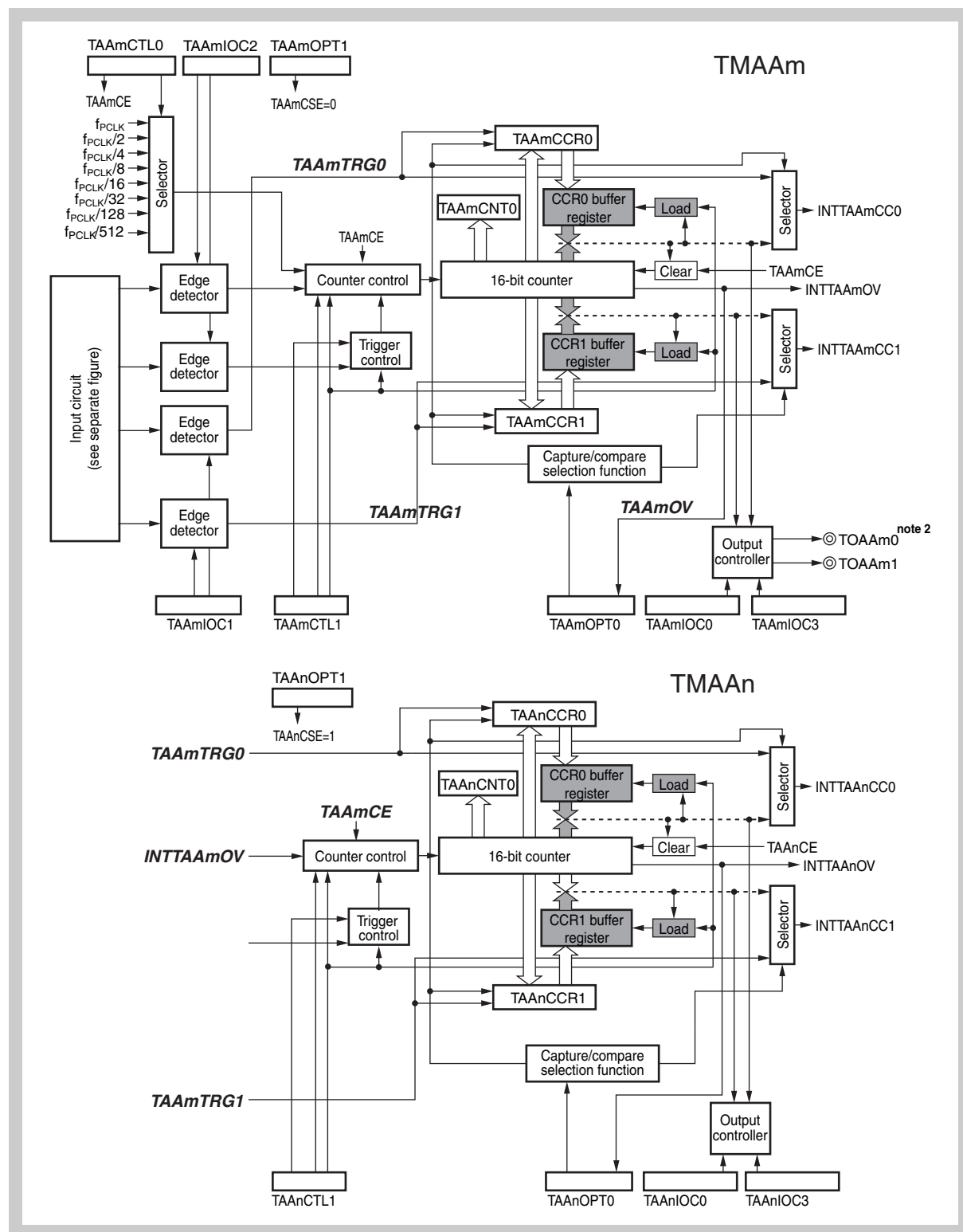
D<sub>10</sub>, D<sub>11</sub>: Values captured to TAAAnCCR1 register (0000H to FFFFH)

TIAAn0: Set detection to rising edge (TAAAnIS1, TAAAnIS0 = 01B)

TIAAn1: Set detection to falling edge (TAAAnIS3, TAAAnIS2 = 10B)

### 16.6.9 32-bit Capture in free-running cascade mode

Two Timer AA (TAA0 in combination with one of TAA1 to TAA3, or TAA4 in combination with one of TAA5 to TAA7, or TAA8 and TAA9) can be cascaded to operate as a 32-bit capture timer. In cascade mode, the timer with the lower number (TAA0, TAA4 or TAA8) is used to control the operation (master timer). Both cascaded timers have to be initialized as free-running timers.



**Figure 16-36** Block diagram of TAAm and TAAm in 32-bit capture mode

- Note**
1.  $m = 0, 4, \text{ or } 8$   
 $n = 1 \text{ to } 3, \text{ when } m = 0$   
 $n = 5 \text{ to } 7, \text{ when } m = 4$   
 $n = 9, \text{ when } m = 8$
  2. Explanation of signals can be found in *Figure 16-1 on page 543*.
  3. Block diagrams of the input circuits can be found in *Figure 16-2 on page 544* and *Figure 16-3 on page 544*,

*Figure 16-36* shows the block diagram of TAAm and TAA<sub>n</sub> in cascade mode. Signals that are irrelevant in cascade mode are not shown, the connections to the internal bus are also hidden for better readability of the image, as *Figure 16-1 on page 543* can be used for a in-depth look of each timer.

- Note** Cascading of two TAA is only allowed for free-running mode with both capture/compare registers set to capture mode. Proper operation of TAAm and TAA<sub>n</sub> is not guaranteed for any other setting.

*Figure 16-37* shows the recommended flow for setting up TAAm and TAA<sub>n</sub> in cascade mode. As TAAm is used for general control, TAA<sub>n</sub> is set up first and set in cascaded operation by setting the TAA<sub>n</sub>CSE bit to 1. Then TAAm is initialized by selecting the proper clock setting and capture trigger input. Only TIAAm0 and TIAAm1 can be used as external capture trigger.

- Note** When cascading TAAm and TAA<sub>n</sub>, set TAA<sub>n</sub>CSE=1 and TAAmCSE=0.

Operation starts when the count enable flag of TAAm (TAAmCE) is set to 1. The counter of TAAm is used for the lower 16-bit of the 32-bit count value, while the upper 16-bit are handled by TAA<sub>n</sub>.

Whenever the counter of TAAm overflows, the counter is cleared to 0, interrupt INTTAAmOV is generated and the counter of TAA<sub>n</sub> is incremented by 1. When the counter of TAA<sub>n</sub> overflows, the counter is also cleared to 0 and interrupt INTTAA<sub>n</sub>OV is generated.

When a capture trigger 0/1 is detected by TAAm, a capture of the lower 16-bit counter value to TAAmCCR0/1 and of the upper 16-bit counter value to TAA<sub>n</sub>CCR0/1 at the same time. The interrupts of the TAAm will indicate the capture (INTTAAmCC0/1).

*Figure 16-38 on page 603* shows an example of a 32-bit capture timing.

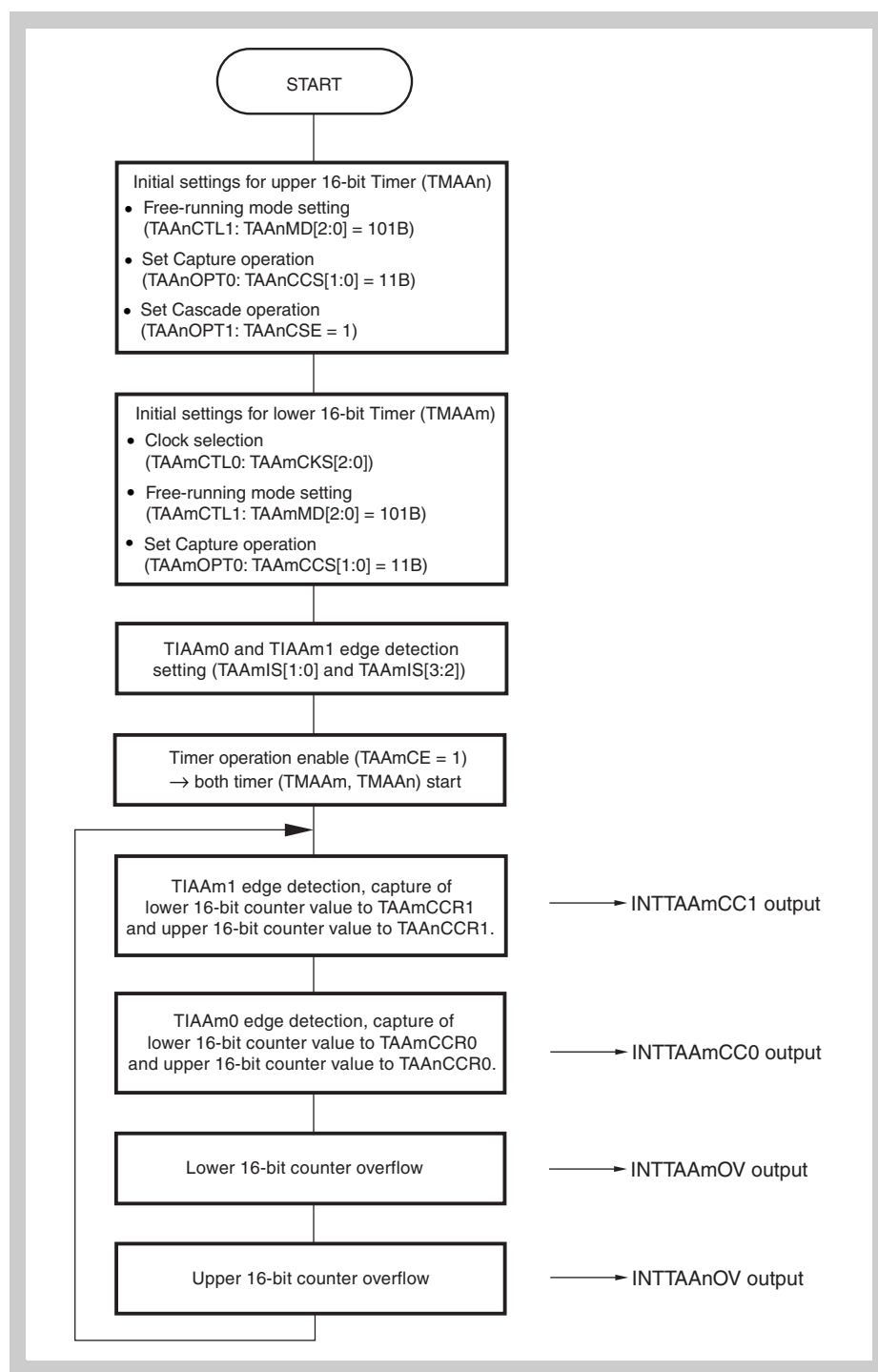
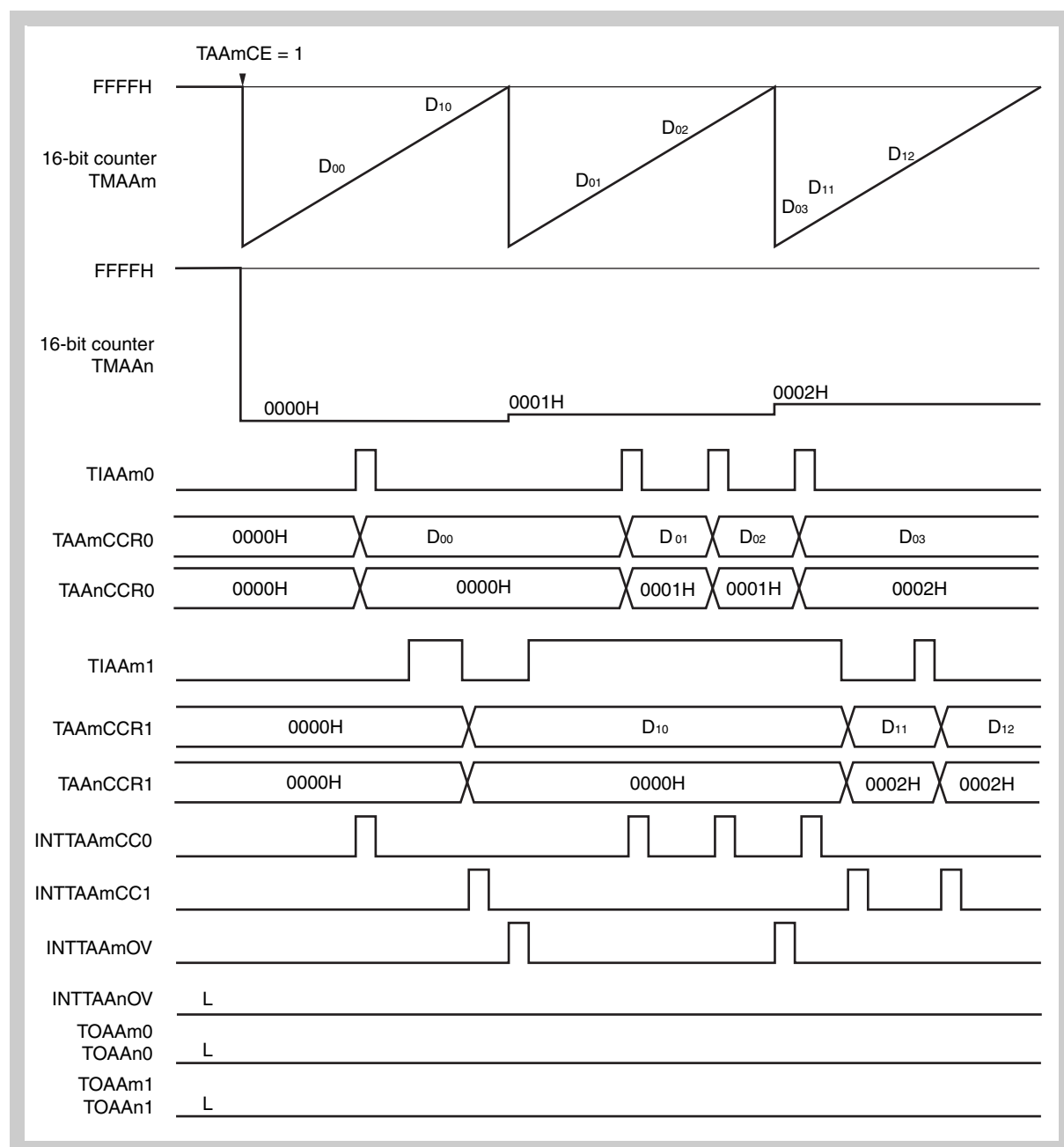


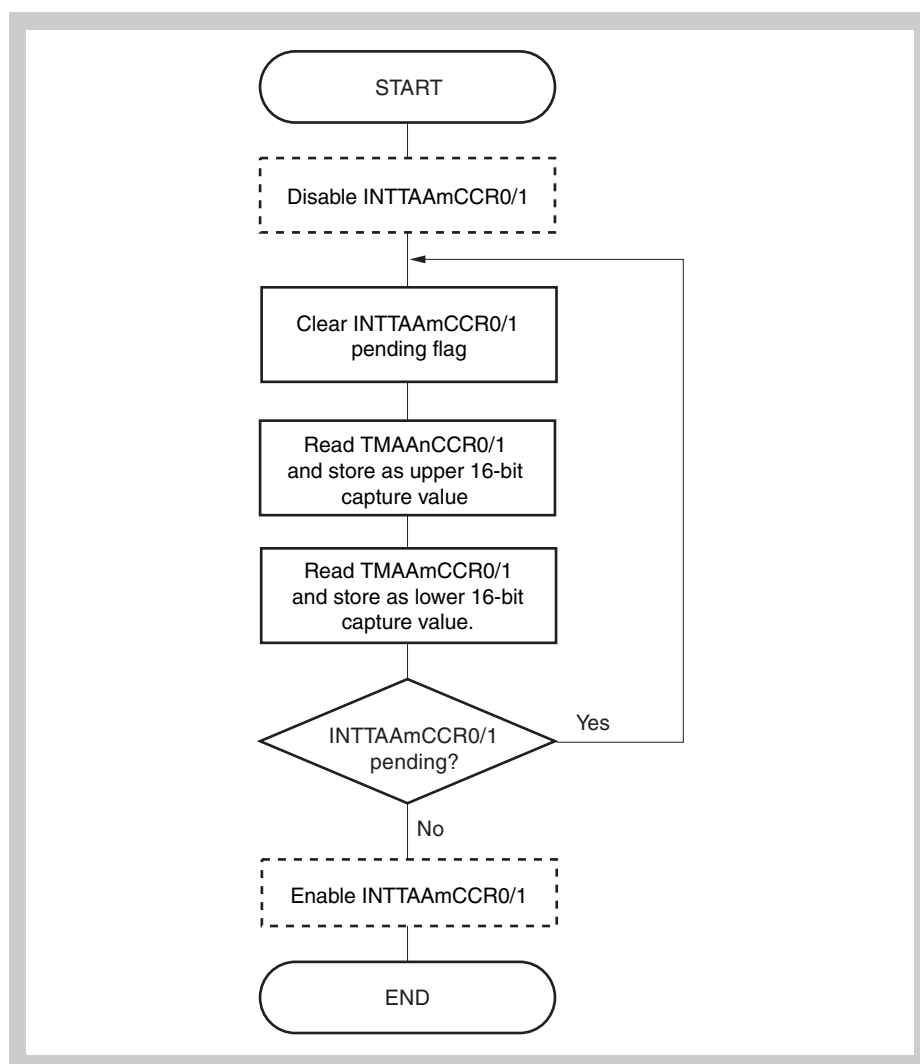
Figure 16-37 Basic flow of 32-bit capture mode



**Figure 16-38 Basic timing of 32-bit capture mode**

**Note**  $m = 0, 4, \text{ or } 8$   
 $n = 1 \text{ to } 3, \text{ when } m = 0$   
 $n = 5 \text{ to } 7, \text{ when } m = 4$   
 $n = 9, \text{ when } m = 8$

As the 32-bit resolution is achieved by cascading two individual TAA, a direct read of the 32-bit capture value is not possible. To ensure that the data is not corrupted during read operation, the following procedure for reading needs to be followed:



**Figure 16-39** Flow of 32-bit read (capture or counter value)

Disabling the capture interrupt (INTTAAmCCR0/1) is not required if the read sequence is done in the interrupt service routine, as nesting of the same interrupt is not possible. However, if the read operation is done in a normal routine while the interrupt signal is also assigned to an interrupt service routine, disabling the interrupt is mandatory, otherwise corrupted data might be read.

The same flow can be used for reading the timer counter value. In this case the relevant interrupt which pending flags needs to be cleared and checked is INTTAAmOV. Please note that you can either read the upper 16-bit counter (TAAmCNT) and then the lower 16-bit counter (TAAmCNT) or vice versa. While both methods work, the read values can be slightly different, as the count operation of the lower 16-bit counter continues while the upper 16-bit timer is read:

- When reading the upper 16-bit first, the lower 16-bit might be incremented during that read.
- When reading the lower 16-bit first, the value might be already “old” after reading the upper 16-bit.

The software programmer needs to decide which method is considered better for the application.



### 16.6.10 Specific TAAAnCCRm overflow value write

This section describes some examples of the specific overflow value write function.

#### (1) General operation

- Conditions**
- TAAAnIOC1.TAAAnIS[1:0] = 01<sub>B</sub>: TIAAn0 rising edge detection
  - TAAAnIOC1.TAAAnIS[3:2] = 10<sub>B</sub>: TIAAn1 falling edge detection
  - TAAAnSVC.TAAAnCSD0 = 0: TAAAnCCR0 always overwritten upon overflow
  - TAAAnSVC.TAAAnCSD1 = 0: TAAAnCCR1 always overwritten upon overflow
  - TAAAnSVC.TAAAnCSR0 = 0: OVfV for TAAAnCCR0 not reversed
  - TAAAnSVC.TAAAnCSR1 = 0: OVfV for TAAAnCCR1 not reversed
  - TAAAnSVC.TAAAnENCM = 1: overflow value write function enabled
  - TAAAnSVC.TAAAnCSE0 = 0: OVfV = 0x0000, if TIAAnm = L at overflow
  - TAAAnSVC.TAAAnCSE1 = 1: OVfV = 0xFFFF, if TIAAnm = H at overflow
  - TAAAnCTL0.TAAAnCE = 1: enable TAAAn operation

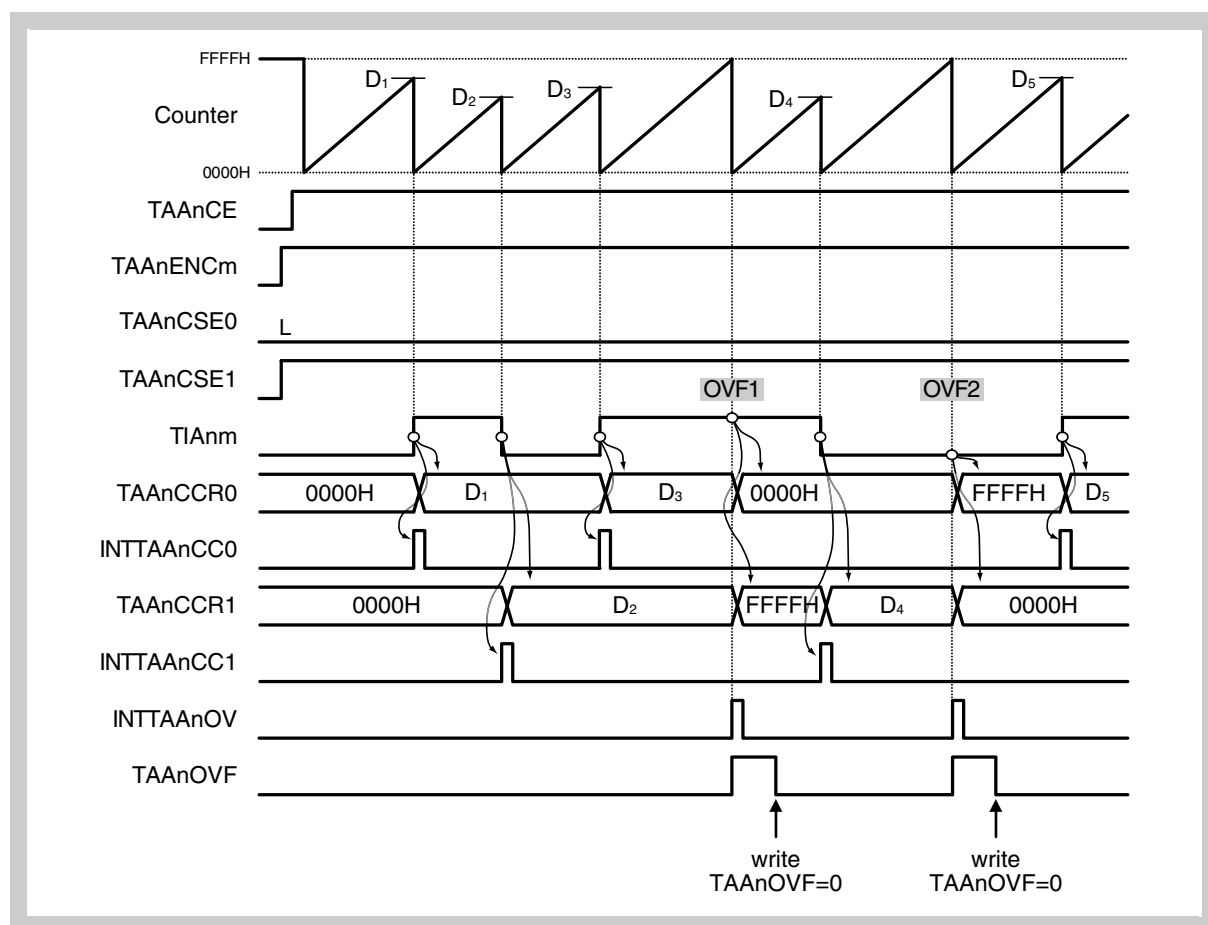


Figure 16-40 Specific overflow value write: general operation

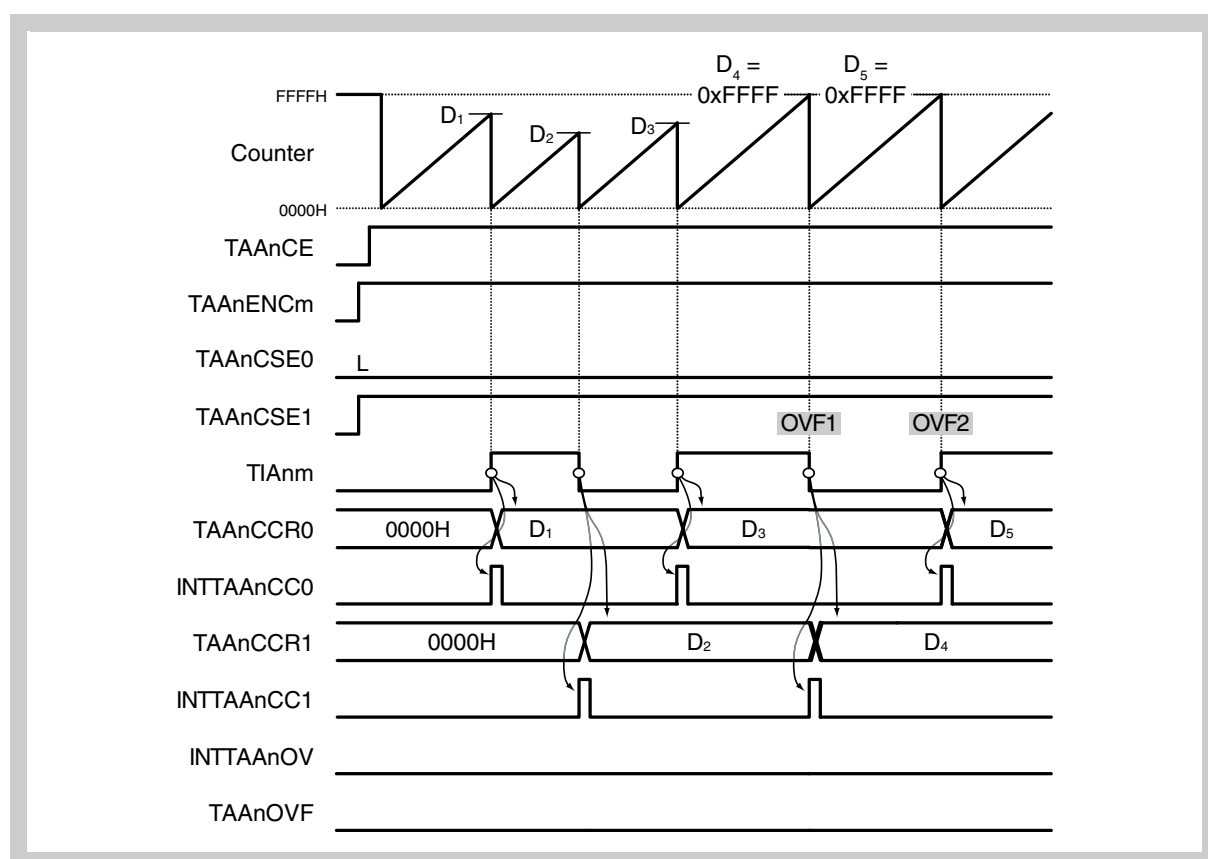
**OVf1** TAAAnCCR0 = 0x0000, because TAAAnCSE0 = L and TIAAnm = H

**OVf2** TAAAnCCR1 = 0xFFFF, because TAAAnCSE1 = L and TIAAnm = H

**(2) Concurrent capture edge and overflow**

When an overflow and a valid edge at the TIA<sub>n</sub>m input is detected concurrently, the capture operation has priority and the overflow value is not written to TAA<sub>n</sub>CCR<sub>m</sub>. Thus no overflow interrupt INTTAA<sub>n</sub>OV is generated.

- Conditions**
- TAA<sub>n</sub>IOC1.TAA<sub>n</sub>IS[1:0] = 01<sub>B</sub>: TIAA<sub>n</sub>0 rising edge detection
  - TAA<sub>n</sub>IOC1.TAA<sub>n</sub>IS[3:2] = 10<sub>B</sub>: TIAA<sub>n</sub>1 falling edge detection
  - TAA<sub>n</sub>SVC.TAA<sub>n</sub>CSD0 = 0: TAA<sub>n</sub>CCR0 always overwritten upon overflow
  - TAA<sub>n</sub>SVC.TAA<sub>n</sub>CSD1 = 0: TAA<sub>n</sub>CCR1 always overwritten upon overflow
  - TAA<sub>n</sub>SVC.TAA<sub>n</sub>CSR0 = 0: OV<sub>FV</sub> for TAA<sub>n</sub>CCR0 not reversed
  - TAA<sub>n</sub>SVC.TAA<sub>n</sub>CSR1 = 0: OV<sub>FV</sub> for TAA<sub>n</sub>CCR1 not reversed
  - TAA<sub>n</sub>SVC.TAA<sub>n</sub>ENC<sub>m</sub> = 1: overflow value write function enabled
  - TAA<sub>n</sub>SVC.TAA<sub>n</sub>CSE0 = 0: OV<sub>FV</sub> = 0x0000, if TIA<sub>n</sub>m = L at overflow
  - TAA<sub>n</sub>SVC.TAA<sub>n</sub>CSE1 = 1: OV<sub>FV</sub> = 0xFFFF, if TIA<sub>n</sub>m = H at overflow
  - TAA<sub>n</sub>CTL0.TAA<sub>n</sub>CE = 1: enable TAA<sub>n</sub> operation



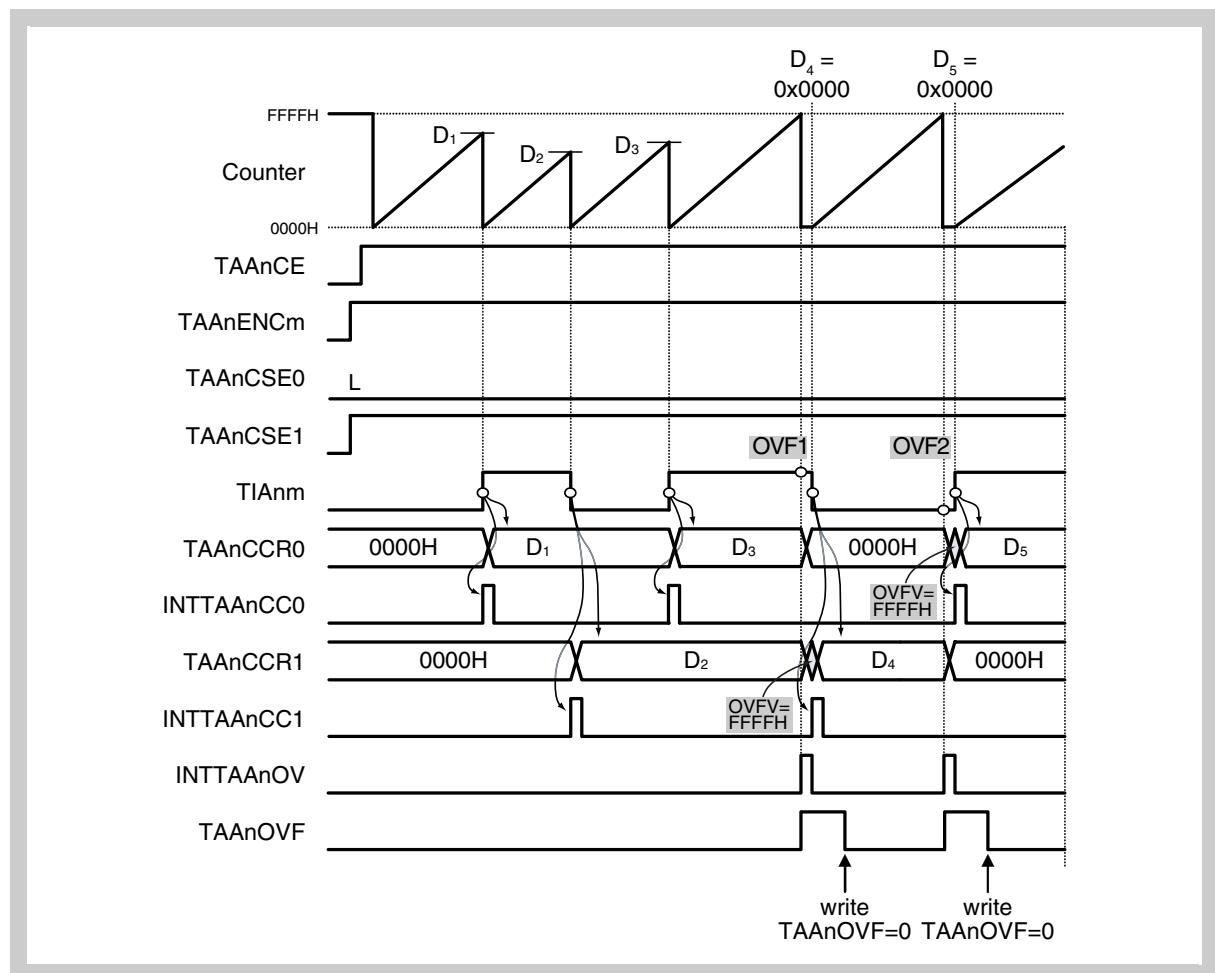
**Figure 16-41** Specific overflow value write: concurrent capture edge and overflow

- OV1** Though the counter overflows, D<sub>4</sub> = 0xFFFF is captured in TAA<sub>n</sub>CCR1 because of a valid falling edge at TIA<sub>n</sub>1. Thus no overflow interrupt INTTAA<sub>n</sub>OV is generated.
- OV2** Though the counter overflows, D<sub>5</sub> = 0xFFFF is captured in TAA<sub>n</sub>CCR0 because of a valid rising edge at TIA<sub>n</sub>0.

**(3) Capture edge immediately after overflow**

In case an overflow event is immediately followed by a valid capture edge, i.e. with the next count clock, the specified overflow value remains in the concerned TAA<sub>n</sub>CCR<sub>m</sub> register for a single count clock, as the valid capture edge captures stores the counter value to the TAA<sub>n</sub>CCR<sub>m</sub>.

- Conditions**
- TAA<sub>n</sub>IOC1.TAA<sub>n</sub>IS[1:0] = 01<sub>B</sub>: TIAA<sub>n</sub>0 rising edge detection
  - TAA<sub>n</sub>IOC1.TAA<sub>n</sub>IS[3:2] = 10<sub>B</sub>: TIAA<sub>n</sub>1 falling edge detection
  - TAA<sub>n</sub>SVC.TAA<sub>n</sub>CSD0 = 0: TAA<sub>n</sub>CCR0 always overwritten upon overflow
  - TAA<sub>n</sub>SVC.TAA<sub>n</sub>CSD1 = 0: TAA<sub>n</sub>CCR1 always overwritten upon overflow
  - TAA<sub>n</sub>SVC.TAA<sub>n</sub>CSR0 = 0: OV<sub>FV</sub> for TAA<sub>n</sub>CCR0 not reversed
  - TAA<sub>n</sub>SVC.TAA<sub>n</sub>CSR1 = 0: OV<sub>FV</sub> for TAA<sub>n</sub>CCR1 not reversed
  - TAA<sub>n</sub>SVC.TAA<sub>n</sub>ENCM = 1: overflow value write enabled
  - TAA<sub>n</sub>SVC.TAA<sub>n</sub>CSE0 = 0: rising edge
  - TAA<sub>n</sub>SVC.TAA<sub>n</sub>CSE0 = 1: falling edge
  - TAA<sub>n</sub>CTL0.TAA<sub>n</sub>CE = 1: enable TAA<sub>n</sub> operation



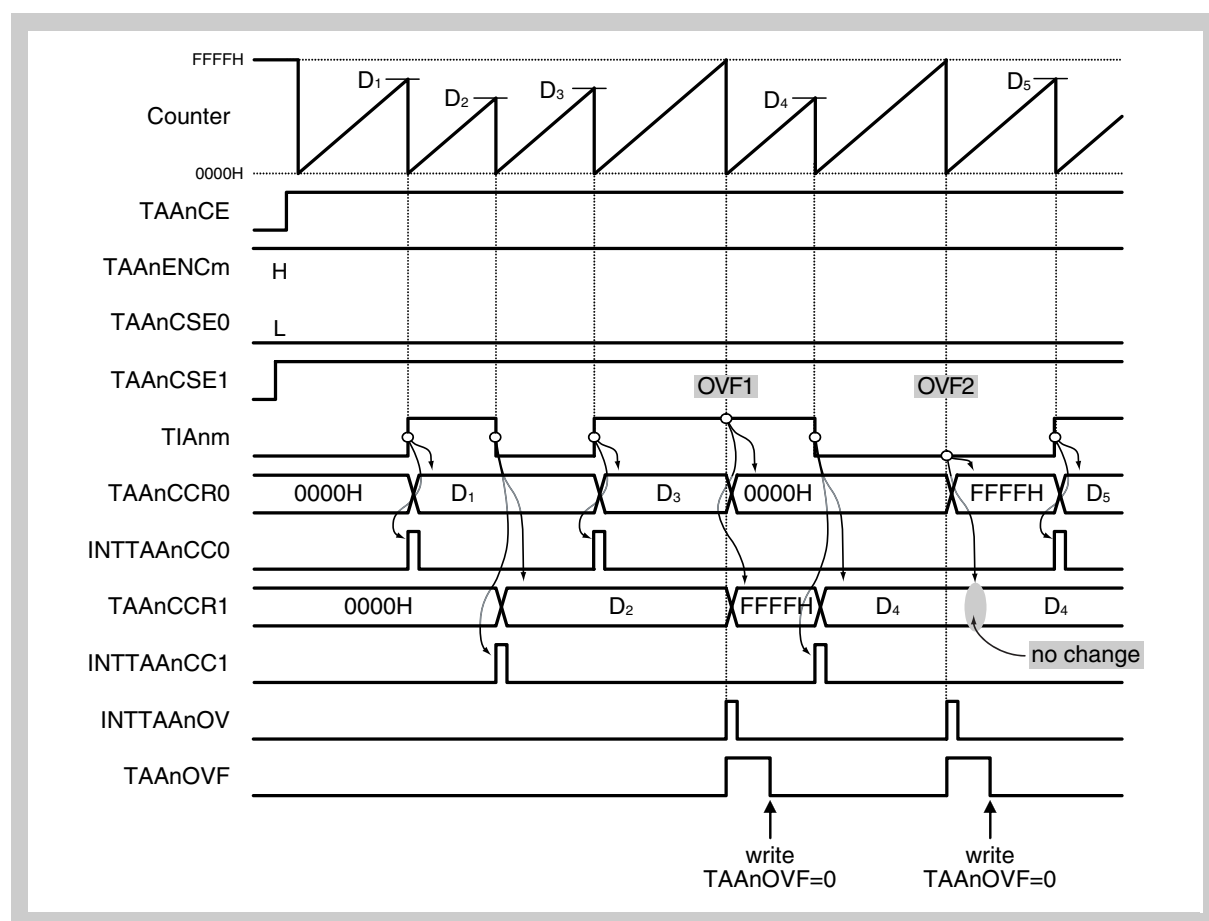
**Figure 16-42 Specific overflow value write: capture edge immediately after overflow**

- OVF1** TAA<sub>n</sub>CCR0 is set to overflow value OV<sub>FV</sub> = 0x0000 (TIA<sub>n</sub>m = H and TAA<sub>n</sub>CSE0 = 0) and TAA<sub>n</sub>CCR1 is set to overflow value OV<sub>FV</sub> = 0xFFFF (TIA<sub>n</sub>m = H and TAA<sub>n</sub>CSE1 = 1). The immediately following falling edge of TIA<sub>n</sub>m, valid for capture channel 1, captures D<sub>4</sub> = 0x0000 in TAA<sub>n</sub>CCR1, while TAA<sub>n</sub>CCR0 remains with the overflow value 0x0000.

- OVF2** TAAAnCCR0 is set to overflow value OVFV = 0xFFFF (TIAAnm = L and TAAAnCSE0 = 0) and TAAAnCCR1 is set to overflow value OVFV = 0x0000 (TIAAnm = L and TAAAnCSE1 = 1).  
The immediately following rising edge of TIAAn, valid for capture channel 0, captures D<sub>5</sub> = 0x0000 in TAAAnCCR0, while TAAAnCCR1 remains with the overflow value 0x0000.

**(4) TAAAnCCRm overwrite prohibition**

- Conditions**
- TAAAnIOC1.TAAAnIS[1:0] = 01<sub>B</sub>: TIAAn0 rising edge detection
  - TAAAnIOC1.TAAAnIS[3:2] = 10<sub>B</sub>: TIAAn1 falling edge detection
  - TAAAnSVC.TAAAnCSD0 = 0: TAAAnCCR0 always overwritten upon overflow
  - TAAAnSVC.TAAAnCSD1 = 1: TAAAnCCR1 overwritten upon overflow and if TIAAnm = not(TAAAnCSE1)
  - TAAAnSVC.TAAAnCSR0 = 0: OVfV for TAAAnCCR0 not reversed
  - TAAAnSVC.TAAAnCSR1 = 0: OVfV for TAAAnCCR1 not reversed
  - TAAAnSVC.TAAAnENCM = 1: overflow value write enabled
  - TAAAnSVC.TAAAnCSE0 = 0: rising edge
  - TAAAnSVC.TAAAnCSE0 = 1: falling edge
  - TAAAnCTL0.TAAAnCE = 1: enable TAAAn operation



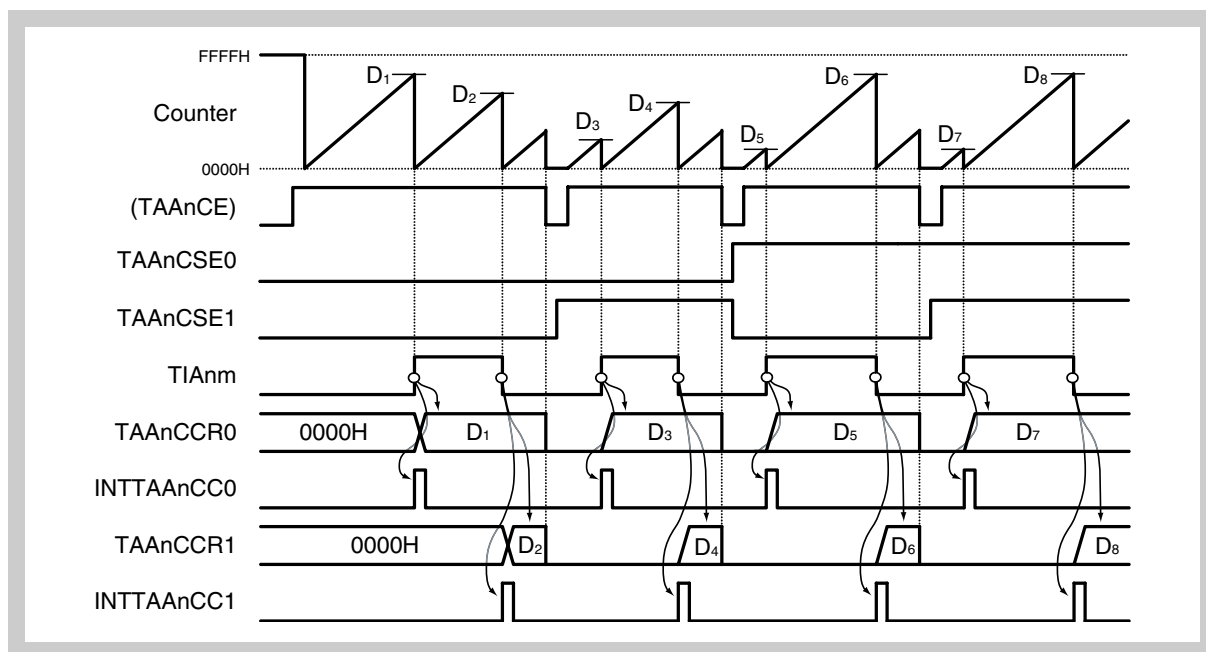
**Figure 16-43 Specific overflow value write: TAAAnCCRm overwrite prohibition**

- OVf1** TAAAnCCR0 is set to overflow value OVfV = 0x0000 (TIAAnm = H and TAAAnCSE0 = 0) and TAAAnCCR1 is set to overflow value OVfV = 0xFFFF (TIAAnm = H and TAAAnCSE1 = 1).
- OVf2** TAAAnCCR0 is set to overflow value OVfV = 0xFFFF (TIAAnm = L and TAAAnCSE0 = 0). TAAAnCCR1 is not overwritten since TAAAnCSD1 = 1, TIAAnm = L while TAAAnCSE1 = 1.

**(5) Overflow value write without overflow event**

The setting of TAAAnSVC.TAAAnCSEm does not impact normal capture behaviour, if no overflow occurs.

- Conditions**
- TAAAnIOC1.TAAAnIS[1:0] = 01<sub>B</sub>: TIAAn0 rising edge detection
  - TAAAnIOC1.TAAAnIS[3:2] = 10<sub>B</sub>: TIAAn1 falling edge detection
  - TAAAnSVC.TAAAnENCm = 1: overflow value write enabled



**Figure 16-44** Specific overflow value write:

- Changing the value of TAAAnSVC.TAAAnCSElm has no influence to the normal capture operation, if no overflow event occurs.

## 16.7 Timer Synchronization Operation Function

Timers AA have a timer synchronized operation function (tuned operation mode). Master timer and incorporated slave timers of the corresponding timer group (listed in *Table 16-16*) start and clock synchronously. When the master timer is cleared, the slave timers are cleared synchronously, too.

**Table 16-16 Tuned Operation Mode of Timer**

Master Timer	Slave Timer		
TAA0	TAA1	TAA2	TAA3
TAA4	TAA5	TAA6	TAA7

- Caution**
1. The tuned operation mode is enabled or disabled by the TAA<sub>n</sub>SYE bit of the TAA<sub>n</sub>CTL1 register.
  2. Set the tuned operation mode using the following procedure.
    - Set the TAA<sub>n</sub>SYE bit of the TAA<sub>n</sub>CTL1 register of the slave timer to enable the tuned operation. Set the TAA<sub>n</sub>MD2 to TAA<sub>n</sub>MD0 bits of the TAA<sub>n</sub>CTL1 register of the slave timer to the free-running mode (101B).
    - Set the timer mode of the master timer by using the TAA<sub>n</sub>MD2 to TAA<sub>n</sub>MD0 bits of the TAA<sub>n</sub>CTL1 register to PWM mode (110B), external trigger pulse output mode (010B), or free-running mode (101B). At this time, do not set the TAA<sub>n</sub>SYE bit of the TAA<sub>n</sub>CTL1 register of the master timer.
    - Set the compare register value of the master and slave timers.
    - Set the TAA<sub>n</sub>CE bit of the TAA<sub>n</sub>CTL0 register of the slave timer to enable operation on the internal operating clock.
    - Set the TAA<sub>n</sub>CE bit of the TAA<sub>n</sub>CTL0 register of the master timer to enable operation on the internal operating clock.

*Table 16-17* and *Table 16-18* show the timer modes that can be used in the tuned operation mode

(√: settable, ×: not settable).

**Table 16-17 Timer modes usable in tuned operation mode**

Master Timer	Free-Running Mode	PWM Mode	External Trigger Pulse Output Mode
TAA0	√	√	√
TAA4	√	√	√

- Note** The timing of transmitting data from the compare register of the master timer to the compare register of the slave timer is as follows.
- |              |   |
|--------------|---|
| PPG:         | CPU write timing (anytime write mode)   |
| Toggle, PWM: | Timing at which timer counter and compare register match TOAA <sub>n</sub> 0 (n = 0 to 9) (reload mode) |

**Table 16-18** Timer output functions

Tuned Timer Group	Timer	Pin	Free-Running Mode		PWM Mode	
			Tuning OFF	Tuning ON	Tuning OFF	Tuning ON
0	TAA0 (master)	TOAA00	PPG	←	Toggle	←
		TOAA01	PPG	←	PWM	←
	TAA1 (slave)	TOAA10	PPG	←	Toggle	PWM
		TOAA11	PPG	←	PWM	←
	TAA2 (slave)	TOAA20	PPG	←	Toggle	PWM
		TOAA21	PPG	←	PWM	←
	TAA3 (slave)	TOAA30	PPG	←	Toggle	PWM
		TOAA31	PPG	←	PWM	←
1	TAA4 (master)	TOAA40	PPG	←	Toggle	←
		TOAA41	PPG	←	PWM	←
	TAA5 (slave)	TOAA50	PPG	←	Toggle	PWM
		TOAA51	PPG	←	PWM	←
	TAA6 (slave)	TOAA60	PPG	←	Toggle	PWM
		TOAA61	PPG	←	PWM	←
	TAA7 (slave)	TOAA70	PPG	←	Toggle	PWM
		TOAA71	PPG	←	PWM	←



## Chapter 17 16-Bit Timer/Event Counter S (TMS)

This microcontroller has two instances of the Timer S (TMS), TMS0 to TMS1.

**Note** Throughout this chapter, the individual instances of Timer S are identified by “n” (n = 0 to 1), for example TSnCTL1 for the Timer Sn control register 1.

### 17.1 Features

Timer S is a 16-bit timer/counter with Motor Control Functions

- Count clock resolution: Minimum 31.25 ns (using 32 MHz count clock)
- Operation modes supporting general-purpose timer and each motor control method
- Reload buffer-equipped compare register
- 10-bit dead time counter
  - Dead time counter with reload buffer
  - Dead time value independent of positive phase to negative phase or negative phase to positive phase can be set.
- A/D conversion trigger signal generation
  - A/D conversion trigger generated by two compare registers, TSnCCR4 and TSnCCR5
- A/D conversion trigger
  - Individual thinning out function for TSnADTRG0/TSnADTRG1 signal: Thinning out ratio settable to 1/1, 1/2, 1/4, 1/8
  - Dedicated output pins (TOSn7), set with TSnADTRG0 signal and reset with TSnADTRG1 signal
- Interrupt thinning out function
  - Thinning out ratios: 1/1 to 1/32
- Enforced output stop function: ESO
  - Output high impedance of TOSn0 to TOSn7 pins can be executed during ESO<sub>n</sub> input.
- Compare value setting
  - Reload (batch rewrite function)/anytime rewrite function can be selected<sup>Note</sup>.
- Reload mode
  - Reload is enabled by writing to the TSnCCR1 register (the reload request flag TSnRSF is set) to enable simultaneous transfer of multiple registers
  - Enables transfer at the following reload timings: Peak, trough, and peak and trough<sup>Note</sup>
  - Provides reload request flag TSnRSF
  - Allocates register address enabling DMA transfer
  - Reload thinning out function

- High-accuracy T-PWM mode
  - Enables 0-100% duty PWM output including dead time reduction
  - Improves output resolution without software load because whether or not to use additional pulse(s) in PWM output of count-up operation can be controlled with compare register LSB.
  - Extended operation modes (center method, symmetric interchange method) available for special PWM output behavior, where the duty cycle is split into two phases.
- 180° excitation control
  - Semi-automatic driving function (Enables trigger input from 2-phase encoder and 3-phase encoder)
  - Enables switching from high-accuracy T-PWM mode, 120° excitation mode, or special 120° excitation mode during operation.
- 120° excitation control/Special 120° excitation control
  - Semi-automatic driving function (Enables trigger generation by the offset coordinated with 2-phase encoder, 3-phase encoder, and timer T)
- 3-phase encoder function (Enables signal input such as hall sensor)
- Count clock can be selected from the following 8:  $f_{PCLK}$ ,  $f_{PCLK}/2$ ,  $f_{PCLK}/4$ ,  $f_{PCLK}/8$ ,  $f_{PCLK}/16$ ,  $f_{PCLK}/32$ ,  $f_{PCLK}/128$  and  $f_{PCLK}/512$
- The active level of output pins TOSn0 to TOSn7 can be set individually for each pin.
- Fail safe function (Warning/error interrupt occurrence enabled)
  - Simultaneous active output detection function in positive/negative phase
  - Abnormal input detection function for 3-phase encoder

**Note** High-accuracy T-PWM mode

## 17.2 Configuration

Timer Sn consists of the following hardware.

**Table 17-1 Timer Sn configuration**

Item	Configuration
Counter	16-bit counter × 1 16-bit sub-counter × 1 10-bit dead time counter × 3
Register	Timer Sn counter read register (TSnCNT) Timer Sn sub-counter read register (TSnSBC) Timer Sn dead time setting registers 0, 0A, 1, 1A (TSnDTC0, TSnDTC0A, TSnDTC1, TSnDTC1A <sup>a</sup> ) Timer Sn compare registers 0 to 5 (TSnCCR0 to TSnCCR5, TSnCCR0A to TSnCCR5A <sup>a</sup> ) Timer Sn compare registers 0A to 5A (TSnCCR0 to TSnCCR5, TSnCCR0A to TSnCCR5A <sup>a</sup> ) Timer Sn compare registers 1B to 3B (TSnCCR1B to TSnCCR3B) Timer Sn pattern registers 0, 1 (TSnPAT0, TSnPAT1)
Timer input pin	6 pins (TTRGSn, TEVTSn, TAPTSn0 to TAPTSn2, ESON) <sup>b</sup>
Timer output pin	8 pins (TOSn0 to TOSn7) <sup>b</sup>
Timer input signal	TSnSTCI0, TSnSTCI1, TSnARD
Timer output signal	TSnADTRG0, TSnADTRG1, TSnAEDO, TSnESG, TSnTSF
Control register	Timer Sn control registers 0, 1 (TSnCTL0, TSnCTL1) Timer Sn I/O control registers 0, 2, 4 (TSnIOC0, TSnIOC2, TSnIOC4) Timer Sn option registers 0 to 7 (TSnOPT0 to TSnOPT7) Timer Sn option register 1A (TSnOPT1, TSnOPT1A <sup>a</sup> )
Interrupt	Compare match interrupt (INTTSnCC0 to INTTSnCC5) Peak interrupt (INTTSnCD0) Trough interrupt (INTTSnOD) Overflow interrupt (INTTSnOV) Error interrupt (INTTSnER) Warning interrupt (INTTSnWN)

a) Throughout this manual the register names with an “A” suffix are functional identical to their counterparts without the “A” suffix. Thus all references to register names without suffix are also valid to the registers with suffix.

b) Alternate-function pin

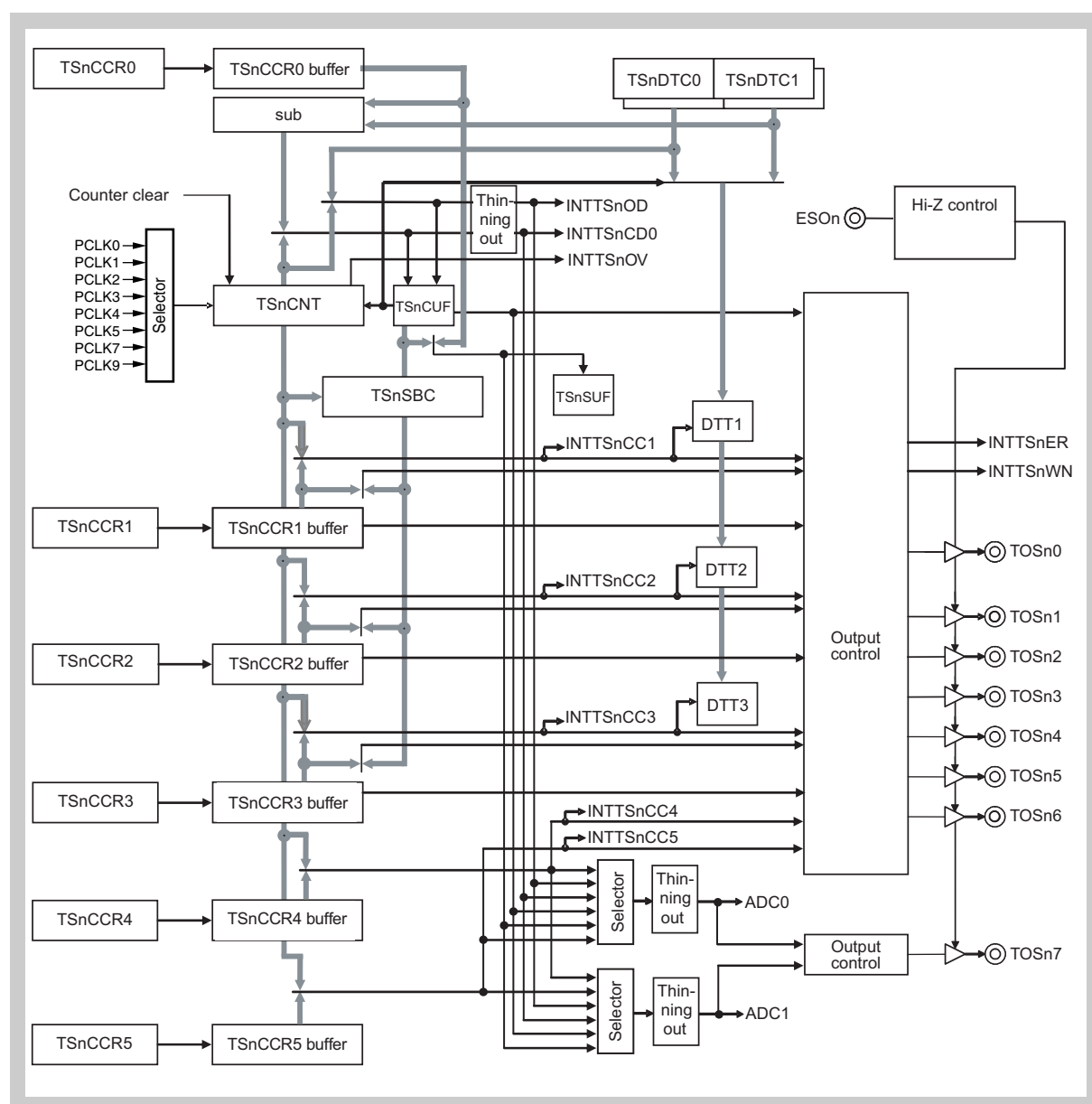


Figure 17-1 Timer Sn block diagram

TSnCCR0 to TSnCCR5:	Compare registers 0 to 5
TSnCCR0 buffer to TSnCCR5 buffer:	Compare buffer registers 0 to 5
TSnDTC0, TSnDTC1:	Timer Sn dead time setting registers 0, 1
TSnSBC:	Timer Sn sub-counter read register
TSnCUF:	Bit 1 of timer Sn option register 0
TSnSUF:	Bit 1 of timer Sn option register 6
INTTSnOD:	Trough interrupt
INTTSnCD0:	Peak interrupt
INTTSnCC1 to INTTSnCC5:	Compare match interrupts 1 to 5
ESOn:	Timer Sn output stop signal input pin
TOSn0 to TOSn7:	Timer output pins 0 to 7
ADC0, ADC1:	A/D Converters 0, 1
DTT1 to DTT3:	Dead time counters

Thinning out:

Thinning out circuit

**Note** PCLKn: peripheral clocks (refer to “Clock Generator” on page 148):

PCLK	0	1	2	3	4	5	7	9
f [MHz]	32	16	8	4	2	1	0.25	0.0625

**(1) Timer Sn compare register 0, 0A (TSnCCR0, TSnCCR0A)**

The TSnCCR0 and TSnCCR0A registers are 16-bit compare registers. The TSnCCR0A register is a mirror of the TSnCCR0 register for optimized DMA support.

**Access** These registers can be read/written in 16-bit units.

**Address** TS0CCR0: FFFFF598<sub>H</sub>  
 TS1CCR0: FFFFF5D8<sub>H</sub>  
 TS0CCR0A: FFFFFB50<sub>H</sub>  
 TS1CCR0A: FFFFFBD0<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. These registers are cleared by any reset.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>TSnCCR0/ TSnCCR0A</b>	Compare value 0															
	R/W															

**Caution** Write to bit 0 of the TSnCCR0 register, or TSnCCR0A respectively, is ignored in high-accuracy T-PWM mode.  
 Bit 0 reads 0.

The write access method for the TSnCCR0 register during the timer Sn operation (when TSnCE of TSnCTL0 register is set to 1) is as follows.

Timer Sn operation mode	TSnCCR0 register write access method
PWM mode, external trigger pulse output mode, triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, special pattern output mode	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Anytime rewrite
High-accuracy T-PWM mode	Reload/anytime rewrite switching enabled

**Caution** The TSnCCR0 register, or TSnCCR0A respectively, setting must be as follows to set the carrier wave cycle in the high-accuracy T-PWM mode.

TSnCCR0 or TSnCCR0A register value =  
 Count clock number of carrier wave cycle  
 + TSnDTC0 or TSnDTC0A register value  
 + TSnDTC1 or TSnDTC1A register value

Refer to “16-bit counter operation in high-accuracy T-PWM mode” on page 801 for the carrier wave and dead time setting.

- Note**
1. Refer to “Compare register rewriting” on page 691 for details on the compare register rewriting operation.
  2. Throughout this manual the register name TSnCCR0 means also the register TSnCCR0A.

**(2) Timer Sn compare register 1, 1A (TSnCCR1, TSnCCR1A)**

The TSnCCR1 and TSnCCR1A registers are 16-bit compare registers. The TSnCCR1A register is a mirror of the TSnCCR1 register for optimized DMA support.

When a compare register is rewritten in the reload mode, the reload request flag (TSnRSF) of TSnOPT6 is set to 1 by write access to the TSnCCR1 register or TSnCCR1A register, and then all of the TSnCCR0 to TSnCCR5, TSnCCR0A to TSnCCR5A, TSnCCR1B to TSnCCR3B, TSnDTC0, TSnDTC1, TSnOPT1, TSnPAT0, and TSnPAT1 registers are rewritten at the next reload timing.

**Access** These registers can be read/written in 16-bit units.

**Address** TS0CCR1: FFFFF59E<sub>H</sub>                      TS0CCR1A: FFFFFB56<sub>H</sub>  
TS1CCR1: FFFFF5DE<sub>H</sub>                      TS1CCR1A: FFFFFBD6<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. These registers are cleared by any reset.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>TSnCCR1/ TSnCCR1A</b>	Compare value 1															
	R/W															

**Note** Additional pulse control function operates if 1 is set to bit 0 of the TSnCCR1 register or TSnCCR1A register in the high-accuracy T-PWM mode. Refer to “Additional pulse control in high-accuracy T-PWM mode” on page 805 for the additional pulse control function.

The write access method for the TSnCCR1/TSnCCR1A register during the timer Sn operation (when TSnCE of TSnCTL0 register is set to 1) is as follows.

Timer Sn operation mode	TSnCCR1/TSnCCR1A register write access method
PWM mode, external trigger pulse output mode, triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, special pattern output mode	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Anytime rewrite
High-accuracy T-PWM mode	Reload/anytime rewrite switching enabled

- Note**
1. Refer to “Compare register rewriting” on page 691 for details on the compare register rewriting operation.
  2. Throughout this manual the register name TSnCCR1 means also the register TSnCCR1A.

**(3) Timer Sn compare register 1B (TSnCCR1B)**

The TSnCCR1B register is a 16-bit compare register, which is not used in the current product. Reading or writing to this register does not have any effect.

**Access** This register can be read/written in 16-bit units.

**Address** TS0CCR1B: FFFFFB4E<sub>H</sub>  
TS1CCR1B: FFFFFBCE<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. This register is cleared by any reset.

**(4) Timer Sn compare register 2, 2A (TSnCCR2, TSnCCR2A)**

The TSnCCR2 and TSnCCR2A registers are 16-bit compare registers. The TSnCCR2A register is a mirror of the TSnCCR2 register for optimized DMA support.

**Access** These registers can be read/written in 16-bit units.

**Address** TS0CCR2: FFFFF59C<sub>H</sub>                      TS0CCR2A: FFFFFB54<sub>H</sub>  
 TS1CCR2: FFFFF5DC<sub>H</sub>                      TS1CCR2A: FFFFFBD4<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. These registers are cleared by any reset.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>TSnCCR2/ TSnCCR2A</b>	Compare value 2															
	R/W															

**Note** The additional pulse control function operates if 1 is set to bit 0 of the TSnCCR2 register or TSnCCR2A register in the high-accuracy T-PWM mode. Refer to “Additional pulse control in high-accuracy T-PWM mode” on page 805 for the additional pulse control function.

The write access method for the TSnCCR2/TSnCCR2A register during the timer Sn operation (when TSnCE of TSnCTL0 register is set to 1) is as follows.

Timer Sn operation mode	TSnCCR2/TSnCCR2A register write access method
PWM mode, external trigger pulse output mode, triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, special pattern output mode	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Anytime rewrite
High-accuracy T-PWM mode	Reload/anytime rewrite switching enabled

- Note**
1. Refer to “Compare register rewriting” on page 691 for details on the compare register rewriting operation.
  2. Throughout this manual the register name TSnCCR2 means also the register TSnCCR2A.

**(5) Timer Sn compare register 2B (TSnCCR2B)**

The TSnCCR2B register is a 16-bit compare register, which is not used in the current product. Reading or writing to this register does not have any effect.

**Access** This register can be read/written in 16-bit units.

**Address** TS0CCR2B: FFFFFB4C<sub>H</sub>  
 TS1CCR2B: FFFFFBCC<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. This register is cleared by any reset.



**(6) Timer Sn compare register 3, 3A (TSnCCR3, TSnCCR3A)**

The TSnCCR3 and TSnCCR3A registers are 16-bit compare registers. The TSnCCR3A register is a mirror of the TSnCCR3 register for optimized DMA support.

**Access** These registers can be read/written in 16-bit units.

**Address** TS0CCR3: FFFFF59A<sub>H</sub>                      TS0CCR3A: FFFFFB52<sub>H</sub>  
 TS1CCR3: FFFFF5DA<sub>H</sub>                      TS1CCR3A: FFFFFBD2<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. These registers are cleared by any reset.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>TSnCCR3/ TSnCCR3A</b>	Compare value 3															
	R/W															

**Note** Additional pulse control function operates if 1 is set to bit 0 of the TSnCCR3 register or TSnCCR3A register in the high-accuracy T-PWM mode. Refer to “Additional pulse control in high-accuracy T-PWM mode” on page 805 for the additional pulse control function.

The write access method for the TSnCCR3/TSnCCR3A register during the timer Sn operation (when TSnCE of TSnCTL0 register is set to 1) is as follows.

Timer Sn operation mode	TSnCCR3B register write access method
PWM mode, external trigger pulse output mode, triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, special pattern output mode	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Anytime rewrite
High-accuracy T-PWM mode	Reload/anytime rewrite switching enabled

- Note**
1. Refer to “Compare register rewriting” on page 691 for details on the compare register rewriting operation.
  2. Throughout this manual the register name TSnCCR3 means also the register TSnCCR3A.

**(7) Timer Sn compare register 3B (TSnCCR3B)**

The TSnCCR3B register is a 16-bit compare register, which is not used in the current product. Reading or writing to this register does not have any effect.

**Access** This register can be read/written in 16-bit units.

**Address** TS0CCR3B: FFFFFB4A<sub>H</sub>  
 TS1CCR3B: FFFFFBCA<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. This register is cleared by any reset.

**(8) Timer Sn compare register 4, 4A (TSnCCR4, TSnCCR4A)**

The TSnCCR4 and TSnCCR4A registers are 16-bit compare registers. The TSnCCR4A register is a mirror of the TSnCCR4 register for optimized DMA support.

The match interrupt signal of the 16-bit counter and the TSnCCR4 or TSnCCR4A register can be selected as the A/D conversion trigger in the high-accuracy T-PWM mode, PWM mode with dead time, special 120° excitation mode, and special pattern output mode.

**Access** These registers can be read/written in 16-bit units.

**Address** TS0CCR4: FFFFF592<sub>H</sub>                      TS0CCR4A: FFFFFB48<sub>H</sub>  
TS1CCR4: FFFFF5D2<sub>H</sub>                      TS1CCR4A: FFFFFBC8<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. These registers are cleared by any reset.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>TSnCCR4/ TSnCCR4A</b>	Compare value 4															
	R/W															

**Note** Bit 0 of the TSnCCR4 and TSnCCR4A registers is ignored in the high-accuracy T-PWM mode.

The write access method for the TSnCCR4/TSnCCR4A register during the timer Sn operation (when TSnCE of TSnCTL0 register is set to 1) is as follows.

Timer Sn operation mode	TSnCCR4/TSnCCR4A register write access method
PWM mode, external trigger pulse output mode, triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, special pattern output mode	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Anytime rewrite
High-accuracy T-PWM mode	Reload/anytime rewrite switching enabled

- Note**
1. Refer to “Compare register rewriting” on page 691 for details on the compare register rewriting operation.
  2. Throughout this manual the register name TSnCCR4 means also the register TSnCCR4A.

**(9) Timer Sn compare register 5, 5A (TSnCCR5, TSnCCR5A)**

The TSnCCR5 and TSnCCR5A registers are 16-bit compare register. The TSnCCR5A register is a mirror of the TSnCCR5 register for optimized DMA support.

The match interrupt signal of the 16-bit counter and the TSnCCR5 register, or TSnCCR5A register respectively, can be selected as the A/D conversion trigger in the high-accuracy T-PWM mode, PWM mode with dead time, special 120° excitation mode, and special pattern output mode.

**Access** These registers can be read/written in 16-bit units.

**Address** TS0CCR5: FFFFF590<sub>H</sub>                      TS0CCR5A: FFFFFB46<sub>H</sub>  
 TS1CCR5: FFFFF5D0<sub>H</sub>                      TS1CCR5A: FFFFFBC6<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. These registers are cleared by any reset.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>TSnCCR5/ TSnCCR5A</b>	Compare value 5															
	R/W															

**Note** Bit 0 of the TSnCCR5 or TsnCCR5A register is ignored in the high-accuracy T-PWM mode.

The write access method for the TSnCCR5/TSnCCR5A register during the timer Sn operation (when TSnCE of TSnCTL0 register is set to 1) is as follows.

Timer Sn operation mode	TSnCCR5/TSnCCR5A register write access method
PWM mode, external trigger pulse output mode, triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, special pattern output mode	Reload
Free-running mode, external event count mode, one-shot pulse mode, interval timer mode	Anytime rewrite
High-accuracy T-PWM mode	Reload/anytime rewrite switching enabled

- Note**
1. Refer to “Compare register rewriting” on page 691 for details on the compare register rewriting operation.
  2. Throughout this manual the register name TSnCCR5 means also the register TSnCCR5A.

**(10) Timer Sn counter read register (TSnCNT)**

The TSnCNT register is a timer read register that can read the 16-bit main counter value.

**Access** This register can be read only in 16-bit units.

**Address** TS0CNT: FFFFF5A6<sub>H</sub>  
TS1CNT: FFFFF5E6<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. This register is cleared by any reset, or when TSnCTL0.TSnCE is set to 0.  
The TSnCNT register is set to FFFF<sub>H</sub> (or FFFE<sub>H</sub> in the high-accuracy T-PWM mode) in the interval from when from TSnCTL0.TSnCE is set to 1 until the count-up operation starts.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
<b>TSnCNT</b>	Counter value																Note
	R/W																

**Note** Bit 0 reads 0 in the high-accuracy T-PWM mode.

**(11) Timer Sn sub-counter read register (TSnSBC)**

The TSnSBC register can read the 16-bit sub counter value.

**Note** This register is operable only in the high-accuracy T-PWM mode or PWM mode with dead-time.

**Access** This register can be read only in 16-bit units.

**Address** TS0SBC: FFFFF5A8<sub>H</sub>  
TS1SBC: FFFFF5E8<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. This register is cleared by any reset, or when TSnCTL0.TSnCE is set to 0.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
<b>TSnSBC</b>	Sub-counter value																Note
	R/W																

**Note** Bit 0 reads 0 in the high-accuracy T-PWM mode.

**(12) Timer Sn dead time setting register 0, 0A (TSnDTC0, TSnDTC0A)**

The TSnDTC0 and TSnDTC0A registers are 10-bit register which specify the dead time value. The TSnDTC0A register is a mirror of the TSnDTC0 register for optimized DMA support.

**Access** These registers can be read/written in 16-bit units.

**Address** TS0DTC0: FFFFF58E<sub>H</sub>                      TS0DTC0A: FFFFFB44<sub>H</sub>  
 TS1DTC0: FFFFF5CE<sub>H</sub>                      TS1DTC0A: FFFFFBC4<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. These registers are cleared by any reset.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSnDTC0/ TSnDTC0A	0	0	0	0	0	0	Dead-time value 0									0
	R/W															

**Note** Throughout this manual the register name TSnDTC0 means also the register TSnDTC0A.

- Caution**
1. The dead time counter is enabled only in the PWM mode (TSnDSE of the TSnOPT0 register is set to 1), high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode (TSnDSE of the TSnOPT0 register is set to 1). In the other modes, be sure to set the TSnDTC0 register, or TSnDTC0A register respectively, to 0000H.
  2. Refer to “*Rewriting of TSnDTC0 and TSnDTC1 registers*” on page 827 regardless of any operation mode of timer Sn when TSnDTC0, or TSnDTC0A register respectively, is rewritten to another value during operation of timer Sn (when TSnCE of the TSnCTL0 register is set to 1).
  3. Dead time is not inserted when the TSnDTC0 register, or TSnDTC0A register respectively, is set to 0000H.
  4. Bits 15 to 10 and 0 are fixed to 0.

**(13) Timer Sn dead time setting register 1, 1A (TSnDTC1, TSnDTC1A)**

The TSnDTC1 and TSnDTC1A registers are 10-bit registers which specify the dead time value. The TSnDTC1A register is a mirror of the TSnDTC1 register for optimized DMA support.

These registers can be read/written with a 16-bit manipulation instruction. Reset input sets these registers to 0000H.

**Access** These registers can be read/written in 16-bit units.

**Address** TS0DTC1: FFFFF58C<sub>H</sub>                      TS0DTC1A: FFFFFB42<sub>H</sub>  
 TS1DTC1: FFFFF5CC<sub>H</sub>                      TS1DTC1A: FFFFFBC2<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. These registers are cleared by any reset.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSnDTC1/ TSnDTC1A	0	0	0	0	0	0	Dead-time value 0									0
	R/W															

**Note** Throughout this manual the register name TSnDTC1 means also the register TSnDTC1A.

- Caution**
1. The dead time counter is enabled only in the PWM mode (TSnDSE of the TSnOPT0 register is set to 1), high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode (TSnDSE of the TSnOPT0 register is set to 1). In the other modes, be sure to set the TSnDTC1, or TSnDTC1A register respectively, register to 0000H.
  2. Refer to “*Rewriting of TSnDTC0 and TSnDTC1 registers*” on page 827 regardless of any operation mode of timer Sn when TSnDTC1, or TSnDTC1A register respectively, is rewritten to another value during operation of timer Sn (when TSnCE of TSnCTL0 register is set to 1).
  3. Dead time is not inserted when the TSnDTC1 register, or TSnDTC1A register respectively, is set to 0000H.
  4. Bits 15 to 10 and 0 are fixed to 0.

## 17.3 Control Register

### (1) Timer Sn control register 0 (TSnCTL0)

The TSnCTL0 register is an 8-bit register that controls the timer Sn operation.

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

**Address** TS0CTL0: FFFFF580<sub>H</sub>  
TS1CTL0: FFFFF5C0<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>TSnCTL0</b>	TSnCE	0	0	0	0	TSnCKS2	TSnCKS1	TSnCKS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Caution** Only the TSnCTL0.TSnCE bit may be changed during operation of timer Sn (when TSnCE is 1). Do not rewrite any other bit than the TSnCE bit.

**Table 17-2 TSnCTL0 register contents (1/2)**

Bit position	Bit name	Function
7	TSnCE	<p>Specifies the timer Sn operation.</p> <p>0: Count disabled (Reset timer Sn to asynchronous)</p> <p>1: Count clock operation enabled</p> <p>When setting the TSnCE bit to 0, the internal operation clock of timer Sn stops and resets timer Sn to asynchronous.</p> <p>When the TSnCE bit is set to 1, the internal operation of timer Sn is enabled and starts count-up operation. See <i>Table 17-3 on page 628</i>.</p> <p><b>Note:</b> The functions initialized by TSnCE = 0 are as follows.</p> <ul style="list-style-type: none"> <li>Internal register and internal latch circuit except for register writable from CPU</li> <li>TSnOVF flag, flags of the TSnOPT6 register</li> <li>16-bit counter, 16-bit sub-counter, dead time counter, the TSnCNT register and TSnSBC register</li> <li>TSnCCR0 to TSnCCR5 buffer registers, TSnPAT0 buffer register, TSnPAT1 buffer register, TSnDTC0 buffer register and TSnDTC1 buffer register</li> </ul> <p>Timer output (inactive level output)</p>

Table 17-2 TSnCTL0 register contents (2/2)

Bit position	Bit name	Function																																				
2 to 0	TSnCKS2 TSnCKS1 TSnCKS0	Selects the count clock of timer TMTn.																																				
		<table><tr><th>TSnCKS2</th><th>TSnCKS1</th><th>TSnCKS0</th><th>Count Clock</th></tr><tr><td>0</td><td>0</td><td>0</td><td>PCLK0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>PCLK1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>PCLK2</td></tr><tr><td>0</td><td>1</td><td>1</td><td>PCLK3</td></tr><tr><td>1</td><td>0</td><td>0</td><td>PCLK4</td></tr><tr><td>1</td><td>0</td><td>1</td><td>PCLK5</td></tr><tr><td>1</td><td>1</td><td>0</td><td>PCLK7</td></tr><tr><td>1</td><td>1</td><td>1</td><td>PCLK9</td></tr></table>	TSnCKS2	TSnCKS1	TSnCKS0	Count Clock	0	0	0	PCLK0	0	0	1	PCLK1	0	1	0	PCLK2	0	1	1	PCLK3	1	0	0	PCLK4	1	0	1	PCLK5	1	1	0	PCLK7	1	1	1	PCLK9
		TSnCKS2	TSnCKS1	TSnCKS0	Count Clock																																	
		0	0	0	PCLK0																																	
		0	0	1	PCLK1																																	
		0	1	0	PCLK2																																	
		0	1	1	PCLK3																																	
		1	0	0	PCLK4																																	
		1	0	1	PCLK5																																	
		1	1	0	PCLK7																																	
1	1	1	PCLK9																																			
<b>Note:</b> 1. Set the TSnCKS[2:0] bits while TSnCE = 0. The TSnCKS[2:0] bits can be changed simultaneously when the setting of TSnCE is changed from 0 to 1.																																						
2. The TSnCKS[2:0] bits are valid only when TSnCTL1.TSnEEE is set to 0.																																						
3. $f_{PCLKn}$ : Peripheral I/O system clock $f_{TMSn}$ : Timer Sn base clock ( $f_{TMSn} = f_{PCLK0}$ )																																						

Table 17-3 Timer Sn Count Clock and Time from TSnCE = 1 Setting to Count-up Operation

Count clock	TSnCKS2	TSnCKS1	TSnCKS0	Min.	Max.
PCLK0	0	0	0	4 base clocks	5 base clocks
PCLK1	0	0	1		
PCLK2	0	1	0		
PCLK3	0	1	1	4 base clocks	5 base clocks + 1 count clock
PCLK4	1	0	0		
PCLK5	1	0	1		
PCLK7	1	1	0		
PCLK9	1	1	1		

**Note** Base clock:  $f_{TMSn}$



**(2) Timer Sn control register 1 (TSnCTL1)**

The TSnCTL1 register is an 8-bit register that controls the timer Sn operation.

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

**Address** TS0CTL1: FFFFF581<sub>H</sub>  
TS1CTL1: FFFFF5C1<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>TSnCTL1</b>	0	TSnEST	TSnEEE	0	TSnMD3	TSnMD2	TSnMD1	TSnMD0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Caution**
1. Only TSnEST bit can be changed during the timer Sn operation (when TSnCTL0.TSnCE is 1). Do not rewrite any other bit than the TSnEST bit. However, write access of the same value is possible.
  2. The default value 0 of bits 4 and 7 must not be changed.

**Table 17-4 TSnCTL1 register contents (1/2)**

Bit position	Bit name	Function
6	TSnEST	<p>Sets to enable/disable software trigger control.</p> <p>0: Disables software trigger control.</p> <p>1: Enables software trigger control.</p> <ul style="list-style-type: none"> <li>• In one-shot pulse mode. (One-shot pulse software trigger)</li> <li>• In external trigger pulse output mode (Pulse output software trigger)</li> </ul> <hr/> <p><b>Caution:</b> The function operates as a software trigger by setting TSnEST to 1 during the timer Sn operation (when the value of TSnCE of TSnCTL0 register is 1). The function operates as a software trigger by setting TSnCE = 1 when the value of TSnEST is 1 while TSnCE = 0.</p> <hr/> <p><b>Note:</b> The TSnEST bit is always read as "0".</p>
5	TSnEEE	<p>Specifies count clock.</p> <p>0: Use the clock selected with the TSnCKS2 to TSnCKS0 bits.</p> <p>1: Use the external clock (TEVTSn pin input edge).</p> <p>The valid edge during TSnEEE = 1 (external clock: TEVTSn pin) is specified with the TSnEES1 and TSnEES0 bits of the TSnIOC2 register.</p> <p><b>Note:</b> The TSnEEE bit is ignored in the external event count mode.</p>

**Table 17-4 TSnCTL1 register contents (2/2)**

Bit position	Bit name	Function				
3 to 0	TSnMD3 TSnMD2 TSnMD1 TSnMD0	Selects the timer mode..				
		TSnMD3	TSnMD2	TSnMD1	TSnMD0	Timer Mode
		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 mode
		0	1	0	0	PWM mode
		0	1	0	1	Free-running mode
		0	1	1	1	Triangular-wave PWM mode
		1	0	0	0	High-accuracy T-PWM mode
		1	0	0	1	PWM mode with dead time
		1	0	1	0	120° excitation mode
		1	0	1	1	Special 120° excitation mode
		1	1	0	0	Special pattern output mode
		Other than above				Setting prohibited

**(3) Timer Sn I/O control register 0 (TSnIOC0)**

The TSnIOC0 register is a 16-bit register that controls the timer output (TOSn0 to TOSn7 pins).

**Access** This register can be read/written in 16-bit or 8-bit units. However, use the TSnIOC0 register for 16-bit access only.

TSnIOC0H enables access to the higher 8 bits of the TSnIOC0 register and TSnIOC0L enables access to the lower 8 bits of the TSnIOC0 register.

**Address** TS0IOC0: FFFFF582<sub>H</sub> TS0IOC0H: FFFFF583<sub>H</sub> TS0IOC0L: FFFFF582<sub>H</sub>  
TS1IOC0: FFFFF5C2<sub>H</sub> TS1IOC0H: FFFFF5C3<sub>H</sub> TS1IOC0L: FFFFF5C2<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. This register is cleared by any reset.

<b>TSnIOC0</b>	15	14	13	12	11	10	9	8
	TSnOL7	TSnOE7	TSnOL6	TSnOE6	TSnOL5	TSnOE5	TSnOL4	TSnOE4
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	7	6	5	4	3	2	1	0
	TSnOL3	TSnOE3	TSnOL2	TSnOE2	TSnOL1	TSnOE1	TSnOL0	TSnOE0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>TSnIOC0H</b>	7	6	5	4	3	2	1	0
	TSnOL7	TSnOE7	TSnOL6	TSnOE6	TSnOL5	TSnOE5	TSnOL4	TSnOE4
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>TSnIOC0L</b>	7	6	5	4	3	2	1	0
	TSnOL3	TSnOE3	TSnOL2	TSnOE2	TSnOL1	TSnOE1	TSnOL0	TSnOE0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Caution**
1. Set the TSnIOC0 register when TSnCTL0.TSnCE is set to 0, in order to prevent the case that dead time cannot be secured, or short pulse is generated on output pins. Write access of the same value is possible for the TSnIOC0 register during timer Sn operation (when TSnCTL0.TSnCE is set to 1).
  2. If the TSnOL1 to TSnOL6 bits are changed while the TSnIOC4.TSnTBA[2:0] bits and the TSnIOC4.TSnEOC bit is set to 1, an error interrupt may be generated accidentally.  
To stop the timer output during timer operation or to change the output level forcibly, it is recommended to set the operating mode of timer output pins TOSn0 to TOSn7 to port mode, rather than manipulating the TSnIOC0 register.

**Table 17-5 TSnIOC0 register contents**

Bit position	Bit name	Function
15	TSnOL7	Sets the TOSn7 pin output level. 0: Active level is high level. 1: Active level is low level.
14	TSnOE7	Enables/disables the TOSn7 pin output. 0: Timer output disabled (Output is inactive level of TSnOL7.) 1: Timer output enabled
13	TSnOL6	Sets the TOSn6 pin output level. 0: Active level is high level. 1: Active level is low level.
12	TSnOE6	Enables/disables the TOSn6 pin output. 0: Timer output disabled (Output is inactive level of TSnOL6 bit). 1: Timer output enabled
11	TSnOL5	Sets the TOSn5 pin output level. 0: Active level is high level. 1: Active level is low level.
10	TSnOE5	Enables/disables the TOSn5 pin output. 0: Timer output disabled (Output is inactive level of TSnOL5 bit). 1: Timer output enabled
9	TSnOL4	Sets the TOSn4 pin output level. 0: Active level is high level. 1: Active level is low level.
8	TSnOE4	Enables/disables the TOSn4 pin output. 0: Timer output disabled (Output is inactive level of TSnOL4 bit). 1: Timer output enabled
7	TSnOL3	Sets the TOSn3 pin output level. 0: Active level is high level. 1: Active level is low level.
6	TSnOE3	Enables/disables the TOSn3 pin output. 0: Timer output disabled (Output is inactive level of TSnOL3 bit). 1: Timer output enabled
5	TSnOL2	Sets the TOSn2 pin output level. 0: Active level is high level. 1: Active level is low level.
4	TSnOE2	Enables/disables the TOSn2 pin output. 0: Timer output disabled (Output is inactive level of the TSnOL2 bit) 1: Timer output enabled
3	TSnOL1	Sets the TOSn1 output level. 0: Active level is high level. 1: Active level is low level.
2	TSnOE1	Enables/disables the TOSn1 pin output. 0: Timer output disabled (Output is inactive level of the TSnOL1 bit). 1: Timer output enabled
1	TSnOL0	Sets the TOSn0 output level. 0: Active level is high level. 1: Active level is low level.
0	TSnOE0	Enables/disables the TOSn0 pin output. 0: Timer output disabled (Output is inactive level of the TSnOL0 bit). 1: Timer output enabled

**(4) Timer Sn I/O control register 2 (TSnIOC2)**

The TSnIOC2 register is an 8-bit register that controls the valid edge for the external event count input (TEVTSn pin) and external trigger input (TTRGSn pin).

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

**Address** TS0IOC2: FFFFF584<sub>H</sub>  
TS1IOC2: FFFFF5C4<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>TSnIOC2</b>	0	0	0	0	TSnEES1	TSnEES0	TSnETS1	TSnETS0
	R	R	R	R	R/W	R/W	R/W	R/W

**Caution** Set the TSnIOC2 register while operation of timer Sn is stopped (when the value of TSnCTL0.TSnCE is 0). Write access of the same value is possible for the TSnIOC2 register during the timer Sn operation (TSnCTL0.TSnCE is set to 1).

**Table 17-6 TSnIOC2 register contents**

Bit position	Bit name	Function															
3, 2	TSnEES1 TSnEES0	Specifies the external event counter input (TEVTSn) valid edge.															
		<table><tr><th>TSnEES1</th><th>TTnEES0</th><th>External event counter input (TEVTTn) valid edge setting</th></tr><tr><td>0</td><td>0</td><td>No edge detection (Trigger is masked.)</td></tr><tr><td>0</td><td>1</td><td>Rising edge detection</td></tr><tr><td>1</td><td>0</td><td>Falling edge detection</td></tr><tr><td>1</td><td>1</td><td>Both, rising and falling edge detection</td></tr></table>	TSnEES1	TTnEES0	External event counter input (TEVTTn) valid edge setting	0	0	No edge detection (Trigger is masked.)	0	1	Rising edge detection	1	0	Falling edge detection	1	1	Both, rising and falling edge detection
		TSnEES1	TTnEES0	External event counter input (TEVTTn) valid edge setting													
		0	0	No edge detection (Trigger is masked.)													
		0	1	Rising edge detection													
		1	0	Falling edge detection													
		1	1	Both, rising and falling edge detection													
<b>Note:</b> The TSnEES1 and TSnEES0 bits are valid only when TSnCTL1.TSnEEE bit is set to 1, or when the external event count mode is selected (TSnCTL1.TSnMD[3:0] = 0001B)																	
1, 0	TSnETS1 TSnETS0	Specifies the external trigger input (TTRGSn) valid edge.															
		<table><tr><th>TSnETS1</th><th>TSnETS0</th><th>External trigger input (TTRGSn) valid edge setting</th></tr><tr><td>0</td><td>0</td><td>No edge detection (Trigger is masked.)</td></tr><tr><td>0</td><td>1</td><td>Rising edge detection</td></tr><tr><td>1</td><td>0</td><td>Falling edge detection</td></tr><tr><td>1</td><td>1</td><td>Both, rising and falling edge detection</td></tr></table>	TSnETS1	TSnETS0	External trigger input (TTRGSn) valid edge setting	0	0	No edge detection (Trigger is masked.)	0	1	Rising edge detection	1	0	Falling edge detection	1	1	Both, rising and falling edge detection
		TSnETS1	TSnETS0	External trigger input (TTRGSn) valid edge setting													
		0	0	No edge detection (Trigger is masked.)													
		0	1	Rising edge detection													
		1	0	Falling edge detection													
		1	1	Both, rising and falling edge detection													
<b>Note:</b> The TSnETS1 and TSnETS0 bits are valid only when the external trigger pulse output mode, or one-shot pulse mode is selected (TSnCTL1.TSnMD[3:0] = 0010B, or 0011B)																	

**(5) Timer Sn I/O control register 4 (TSnIOC4)**

The TSnIOC4 register is an 8-bit register that controls timer output (TOSn1 to TOSn6 pins).

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

**Address** TS0IOC4: FFFFF585<sub>H</sub>  
TS1IOC4: FFFFF5C5<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>TSnIOC4</b>	0	TSnTBA2	TSnTBA1	TSnTBA0	0	TSnWOC	0	TSnEOC
	R	R/W	R/W	R/W	R	R/W	R	R/W

**Caution** Set the TSnIOC4 register while operation of timer Sn is stopped (TSnCTL0.TSnCE is set to 0).

**Table 17-7 TSnIOC4 register contents (1/2)**

Bit position	Bit name	Function
6	TSnTBA2	<p>Selects whether to detect the simultaneous active states of the TOSn5 and TOSn6 pins.</p> <p>0: Don't detect the simultaneous active states of the TOSn5 and TOSn6 pins. 1: Detect the simultaneous active states of the TOSn5 and TOSn6 pins.</p> <p><b>Note:</b> If simultaneous active states are detected while TSnEOC = 1 or TSnTBA2 = 1, the positive/negative phase simultaneous active state detection flag (TSnTBF) is set to 1 and an error interrupt (INTTSnER) occurs.</p>
5	TSnTBA1	<p>Selects whether to detect the simultaneous active states of the TOSn3 and TOSn4 pins.</p> <p>0: Don't detect the simultaneous active states of the TOSn3 and TOSn4 pins. 1: Detect the simultaneous active states of the TOSn3 and TOSn4 pins.</p> <p><b>Note:</b> If simultaneous active states are detected while TSnEOC = 1 or TSnTBA1 = 1, the positive/negative phase simultaneous active state detection flag (TSnTBF) is set to 1 and an error interrupt (INTTSnER) occurs.</p>
4	TSnTBA0	<p>Selects whether to detect the simultaneous active states of the TOSn1 and TOSn2 pins.</p> <p>0: Don't detect the simultaneous active states of the TOSn1 and TOSn2 pins. 1: Detect the simultaneous active states of the TOSn1 and TOSn2 pins.</p> <p><b>Note:</b> If simultaneous active states are detected while TSnEOC = 1 or TSnTBA0 = 1, the positive/negative phase simultaneous active state detection flag (TSnTBF) is set to 1 and an error interrupt (INTTSnER) occurs.</p>
2	TSnWOC	<p>Selects the warning detection during motor control.</p> <p>0: Disable the warning interrupt (INTTSnWN) 1: Enable the warning interrupt (INTTSnWN)</p> <p><b>Note:</b> Refer to "Warning interrupt function" on page 755 for details on warning interrupt control.</p>

**Table 17-7 TSnIOC4 register contents (2/2)**

Bit position	Bit name	Function
0	TSnEOC	Selects the error status detection during motor control. 0: Disable the error interrupt (INTTSnER). 1: Enable the error interrupt (INTTSnER).  <b>Note:</b> Refer to “Error interrupt function” on page 752 for details on error interrupt control.

**(6) Timer Sn option register 0 (TSnOPT0)**

The TSnOPT0 register is an 8-bit register that detects overflow.  
This register can be read/written with an 8-bit or 1-bit manipulation instruction.  
Reset input sets this register to 00H.

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

**Address** TS0OPT0: FFFFF586<sub>H</sub>  
TS1IOPT0: FFFFF5C6<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>TSnOPT0</b>	0	0	0	0	TSnDSE	TSnCMS	TSnCUF	TSnOVF
	R	R	R	R	R/W	R/W	R/W	R/W

**Caution** Do not rewrite the TSnDSE bit of the TSnOPT0 register during the timer Sn operation (when TSnCTL0.TSnCE is set to 1). However, write access of the same value is possible.

**Table 17-8 TSnOPT0 register contents (1/2)**

Bit position	Bit name	Function
3	TSnDSE	Enables/disables dead time control in the PWM mode and in the special pattern output mode. 0: Disables dead time control 1: Enables dead time control  <b>Note:</b> The TSnDSE bit is valid only in PWM mode and in special pattern output mode, but is invalid in the other modes.
2	TSnCMS	Specifies the transfer timing of the compare register in the high-accuracy T-PWM mode. 0: Reload mode (batch rewrite function) When writing to the TSnCCR1 register, reload is enabled and then the value is updated (reloaded) simultaneously at the next reload timing. Reload is not enabled even if written to a register other than TSnCCR1. 1: Anytime rewrite mode Writing to each compare register under update operation by each independent compare register enables write access value updating anytime. 4 clocks are needed from register write until the value is transferred. In the case of the TSnCCR1 register, 5 clocks are needed.  <b>Caution:</b> Set TSnCMS to 0 in any mode other than the high-accuracy T-PWM mode because the transfer timing is fixed.
1	TSnCUF	This is a flag indicating the 16-bit counter count status. 0: 16-bit counter is in count up status. 1: 16-bit counter is in count down status.  <b>Note:</b> The TSnCUF bit is valid only in the high-accuracy T-PWM mode and in the triangular-wave PWM mode, but is invalid (TSnCUF = 0) in any other mode.



**Table 17-8 TSnOPT0 register contents (2/2)**

Bit position	Bit name	Function
0	TSnOVF	<p>This is a flag indicating the timer Sn overflow. It is set to 1 when the 16-bit counter value overflows from FFFFH to 0000H. This flag is cleared by writing 0 or setting TSnCE to 0. An overflow interrupt (INTTSnOV) occurs at the same time that the TSnOVF bit is set to 1.</p> <hr/> <p><b>Caution:</b></p> <ol style="list-style-type: none"> <li>1. Overflow occurs only in the free-running mode or high-accuracy T-PWM mode. In the high-accuracy T-PWM mode, the TSnOVF bit may be set to 1 if the setting value condition of the TSnDTC0 and TSnDTC1 registers is incorrect.</li> <li>2. The TSnOVF bit is not cleared even if the TSnOVF bit and the TSnOPT0 register are read while TSnOVF = 1.</li> <li>3. Reading or writing is possible for the TSnOVF bit, but writing 1 to the TSnOVF bit is ignored.</li> </ol> <hr/>

**(7) Timer Sn option registers 1, 1H, 1L (TSnOPT1, TSnOPT1H, TSnOPT1L)  
Timer Sn option registers 1A, 1AH, 1AL (TSnOPT1A, TSnOPT1AH,  
TSnOPT1AL)**

The TSnOPT1 and TSnOPT1A registers are 16-bit registers which enable/disable peak/trough interrupt and set the interrupt thinning out ratio. The TSnOPT1A register is a mirror of the TSnOPT1 register for optimized DMA support.

**Access** These registers can be read/written with a 16-bit, 8-bit, or 1-bit manipulation instruction.

TSnOPT1H, and TSnOPT1AH respectively, is an 8-bit register, which can access the higher 8 bits of the TSnOPT1 register, and TSnOPT1A register respectively. TSnOPT1L, and TSnOPT1AL respectively, is an 8-bit register, which can access the lower 8 bits of the TSnOPT1 register, and TSnOPT1AL respectively

**Address** TS0OPT1: FFFFF58A<sub>H</sub> TS0OPT1H: FFFFF58B<sub>H</sub> TS0OPT1L: FFFFF58A<sub>H</sub>  
TS1OPT1: FFFFF5CA<sub>H</sub> TS1OPT1H: FFFFF5CB<sub>H</sub> TS1OPT1L: FFFFF5CA<sub>H</sub>  
TS0OPT1A: FFFFFB40<sub>H</sub> TS0OPT1AH: FFFFFB41<sub>H</sub> TS0OPT1AL: FFFFF58A<sub>H</sub>  
TS1OPT1A: FFFFFB40<sub>H</sub> TS1OPT1AH: FFFFF5CB<sub>H</sub> TS1OPT1AL: FFFFF5CA<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. These registers are cleared by any reset.

TSnOPT1/ TSnOPT1A	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	TSnRTE	TSnRBE
	R	R	R	R	R	R	R/W	R/W
	7	6	5	4	3	2	1	0
	TSnICE	TSnIOE	TSnRDE	TSnID4	TSnID3	TSnID2	TSnID1	TSnID0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
TSnOPT1H/ TSnOPT1AH	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	TSnRTE	TSnRBE
	R	R	R	R	R	R	R/W	R/W
TSnOPT1L/ TSnOPT1AL	7	6	5	4	3	2	1	0
	TSnICE	TSnIOE	TSnRDE	TSnID4	TSnID3	TSnID2	TSnID1	TSnID0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Caution** The register rewriting methods for the TSnOPT1, TSnOPT1H, TSnOPT1L, TSnOPT1A, TSnOPT1AH, and TSnOPT1AL registers during count operation in the high-accuracy T-PWM mode are as follows based on the setting of the TSnCMS bit of the TSnOPT0 register.

- TSnCMS = 0: Reload rewriting
- TSnCMS = 1: Anytime rewriting

The reload rewriting method is used in the external trigger pulse output mode, PWM mode, triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode.

Table 17-9 TSnOPT1/TSnOPT1A register contents (1/2)

Bit position	Bit name	Function
9	TSnRTE	<p>Enables/disables reload at the peak reload timing.            0: Disables the reload operation at the peak timing of 16-bit counter.            1: Enables the reload operation at the peak timing of 16-bit counter.</p> <p><b>Note:</b> The TSnRTE bit is valid only in the external trigger pulse output mode, PWM mode, high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode.</p>
8	TSnRBE	<p>Enables/disables reload at the trough reload timing.            0: Disables reload operation at the trough timing for 16-bit counter.            1: Enables reload operation at the trough timing for 16-bit counter.</p> <p><b>Note:</b> The TSnRBE bit is valid only in the high-accuracy T-PWM mode and triangular-wave PWM mode.</p>
7	TSnICE	<p>Enables/disables occurrence of the peak interrupt (INTTSnCD0).            0: Disables the peak interrupt (INTTSnCD0) at the peak timing for 16-bit counter. Not subject to interrupt thinning out.            1: Enables the peak interrupt (INTTSnCD0) at the peak timing for 16-bit counter. Subject to interrupt thinning out.</p> <p><b>Note:</b> The TSnICE bit is valid only in the PWM mode, high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode.</p>
6	TSnIOE	<p>Enables/disables occurrence of the trough interrupt (INTTSnOD).            0: Disables the trough interrupt. Not subject to interrupt thinning out.            1: Enables the trough interrupt. Subject to interrupt thinning out.</p> <p>The TSnIOE bit is valid only in high-accuracy T-PWM mode and triangular-wave PWM output mode.</p>
5	TSnRDE	<p>Enables/disables thinning out at the reload timing.            0: No reload thinning out            Reload timing is generated based on the setting of the TSnRTE and TSnRBE bits.            1: Reload thinning out provided            Reload timing is generated in the same interval as interrupt thinning out.</p> <p><b>Note:</b> The TSnRDE bit is valid only in the PWM mode, high-accuracy T-PWM mode, triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode.            The TSnRDE bit must be set to 1 in any mode other than the above.</p>

Table 17-9 TSnOPT1/TSnOPT1A register contents (2/2)

Bit position	Bit name	Function																																																																		
4 to 0	TSnID[4:0]	Specifies the thinning out ratio of interrupts (INTTSnCD0, INTTSnOD).																																																																		
		<table><tr><th>TSnID4</th><th>TSnID3</th><th>TSnID2</th><th>TSnID1</th><th>TSnID0</th><th>Thinning out Ratio</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>No thinning out</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1/2</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1/3</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1/4</td></tr><tr><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td></tr><tr><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td></tr><tr><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1/30</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1/31</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1/32</td></tr></table>	TSnID4	TSnID3	TSnID2	TSnID1	TSnID0	Thinning out Ratio	0	0	0	0	0	No thinning out	0	0	0	0	1	1/2	0	0	0	1	0	1/3	0	0	0	1	1	1/4	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	1	1	0	1	1/30	1	1	1	1	0	1/31	1	1	1	1	1	1/32
		TSnID4	TSnID3	TSnID2	TSnID1	TSnID0	Thinning out Ratio																																																													
		0	0	0	0	0	No thinning out																																																													
		0	0	0	0	1	1/2																																																													
		0	0	0	1	0	1/3																																																													
		0	0	0	1	1	1/4																																																													
		.	.	.	.	.	.																																																													
		.	.	.	.	.	.																																																													
		.	.	.	.	.	.																																																													
		1	1	1	0	1	1/30																																																													
		1	1	1	1	0	1/31																																																													
		1	1	1	1	1	1/32																																																													
		<b>Note:</b> The TSnID4 to TSnID0 bits are valid only in the PWM mode, high-accuracy T-PWM mode, triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode.																																																																		
<b>Caution:</b> During the timer Sn operation (when TSnCTL0.TSnCE = 1), the interrupt thinning out counter is cleared if write access is executed to the TSnOPT1 and TSnOPT1L registers, TSnOPT1A and TSnOPT1AL registers respectively (including when the same value is written to the TSnID[4:0] bits). The interrupt thinning out counter is not cleared by write access of the TSnOPT1H register, and TSnOPT1AH register respectively.																																																																				

Table 17-10 Reload, peak interrupt and trough interrupt (1/3)

Mode	TSnC MS	TSnR DE	TSnI CE	TSnR TE	TSnI OE	TSnR BE	Reload	INTTSnCD0	INTTSnOD
PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, special pattern output mode	1	x	x	x	x	x	Setting prohibited		
	0	0	x	0	x	x	No		
	0	0	x	1	x	x	Peak	Occurrence Note 1	
	0	1	0	0	x	x	No		
	0	1	0	1	x	x	No		
	0	1	1	0	x	x	No		
	0	1	1	1	x	x	Peak <sup>Note 3</sup>	Occurrence	-

Table 17-10 Reload, peak interrupt and trough interrupt (2/3)

Mode	TSnC MS	TSnR DE	TSnI CE	TSnR TE	TSnI OE	TSnR BE	Reload	INTTSnCD0	INTTSnOD
Triangular-wave PWM mode	1	x	x	x	x	x	Setting prohibited		
	0	0	x	x	x	0	No		
	0	0	x	x	x	1	Trough	-	Occurrence Note 2
	0	1	x	x	0	0	No		
	0	1	x	x	0	1	No		
	0	1	x	x	1	0	No		
	0	1	x	x	1	1	Trough <sup>Note 3</sup>	-	Occurrence
High-accuracy T- PWM mode	1	x	x	x	x	x	Anytime	Occurrence Note 1	Occurrence Note 2
	0	0	x	0	x	0	No		
	0	0	x	0	x	1	Trough	Occurrence Note 1	Occurrence Note 2
	0	0	x	1	x	0	Peak	Occurrence Note 1	Occurrence Note 2
	0	0	x	1	x	1	Peak/trough	Occurrence Note 1	Occurrence Note 2
	0	1	0	0	0	0	No		
	0	1	0	0	0	1	No		
	0	1	0	0	1	0	No		
	0	1	0	0	1	1	Trough <sup>Note 3</sup>	-	Occurrence
	0	1	0	1	0	0	No		
	0	1	0	1	0	1	No		
	0	1	0	1	1	0	No		
	0	1	0	1	1	1	Trough <sup>Note 3</sup>	-	Occurrence
	0	1	1	0	0	0	No		
	0	1	1	0	0	1	No		
	0	1	1	0	1	0	No		
	0	1	1	0	1	1	Trough <sup>Note 3</sup>	Occurrence	Occurrence
	0	1	1	1	0	0	Peak <sup>Note 3</sup>	Occurrence	-
	0	1	1	1	0	1	Peak <sup>Note 3</sup>	Occurrence	-
	0	1	1	1	1	0	Peak <sup>Note 3</sup>	Occurrence	Occurrence
	0	1	1	1	1	1	Peak <sup>Note 3</sup> 3/Trough <sup>Note 3</sup>	Occurrence	Occurrence

**Table 17-10 Reload, peak interrupt and trough interrupt (3/3)**

Mode	TSnC MS	TSnR DE	TSnI CE	TSnR TE	TSnI OE	TSnR BE	Reload	INTTSnCD0	INTTSnOD
External trigger pulse output mode	1	x	x	x	x	x	Setting prohibited		
	0	0	x	0	x	x	No		
	0	0	x	1	x	x	Peak	-	-
	0	1	0	0	x	x	Setting prohibited		
	0	1	0	1	x	x	Setting prohibited		
	0	1	1	0	x	x	Setting prohibited		
	0	1	1	1	x	x	Setting prohibited		

- Note**
1. Only when TSnICE = 1
  2. Only when TSnIOE = 1
  3. At the timing at which thinning out of the peak interrupt or the trough interrupt does not occur.
  4. X: 0 or 1
  5. Throughout this manual the register name TSnOPT1(H/L) means also the register TSnOPT1A(H/L).

**(8) Timer Sn option registers 2, 2H, 2L (TSnOPT2, TSnOPT2H, TSnOPT2L)**

These TSnOPT2 registers are 16-bit registers that control A/D conversion trigger output (TSnADTRG0 signal).

**Access** These registers can be read/written with a 16-bit, 8-bit, or 1-bit manipulation instruction.

TSnOPT2H can access the higher 8 bits of the TSnOPT2 register. TSnOPT2L can access the lower 8 bits of the TSnOPT2 register.

**Address** TS0OPT2: FFFFF5A0<sub>H</sub> TS0OPT2H: FFFFF5A1<sub>H</sub> TS0OPT2L: FFFFF5A0<sub>H</sub>  
TS1OPT2: FFFFF5E0<sub>H</sub> TS1OPT2H: FFFFF5E1<sub>H</sub> TS1OPT2L: FFFFF5E0<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. These registers are cleared by any reset.

	15	14	13	12	11	10	9	8
TSnOPT2	0	0	0	0	0	0	TSnRTE	TSnRBE
	R	R	R	R	R	R	R/W	R/W
	7	6	5	4	3	2	1	0
	TSnICE	TSnIOE	TSnRDE	TSnID4	TSnID3	TSnID2	TSnID1	TSnID0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	7	6	5	4	3	2	1	0
TSnOPT2H	0	0	0	0	0	0	TSnRTE	TSnRBE
	R	R	R	R	R	R	R/W	R/W

	7	6	5	4	3	2	1	0
TSnOPT2L	TSnICE	TSnIOE	TSnRDE	TSnID4	TSnID3	TSnID2	TSnID1	TSnID0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Caution** The write timing is delayed due to wait insertion during register access under the following conditions if serial write to the TSnOPT2 register is executed during the timer Sn operation (when TSnCTL0.TSnCE = 1)

Write address	Next write address	Write delay
TSnOPT2L	TSnOPT2, TSnOPT2L	Delay
	TSnOPT2H	No delay
TSnOPT2H	TSnOPT2, TSnOPT2H	Delay
	TSnOPT2L	No delay
TSnOPT2	TSnOPT2, TSnOPT2H, TSnOPT2L	Delay

Table 17-11 TSnOPT2 register contents (1/3)

Bit position	Bit name	Function															
9,8	TSnACC1, TSnACC0	<p>Specifies the thinning out ratio of the A/D conversion trigger (TSnADTRG0 signal).</p> <table> <tr> <th>TSnACC1</th><th>TSnACC0</th><th>Thinning out Ratio</th></tr> <tr> <td>0</td><td>0</td><td>No thinning out</td></tr> <tr> <td>0</td><td>1</td><td>1/2</td></tr> <tr> <td>1</td><td>0</td><td>1/3</td></tr> <tr> <td>1</td><td>1</td><td>1/4</td></tr> </table> <p><b>Note:</b> During the timer Sn operation (when TSnCTL0.TSnCE = 1), the TSnADTRG0 signal thinning out counter is not cleared by write access to the TSnOPT2L register.</p> <p><b>Caution:</b> During the timer Sn operation (when TSnCTL0.TSnCE = 1), the TSnADTRG0 signal thinning out counter is cleared if write access to the TSnOPT2 and TSnOPT2H registers is executed (including when the same value is written to the TSnACC1 and TSnACC0 bits).</p>	TSnACC1	TSnACC0	Thinning out Ratio	0	0	No thinning out	0	1	1/2	1	0	1/3	1	1	1/4
TSnACC1	TSnACC0	Thinning out Ratio															
0	0	No thinning out															
0	1	1/2															
1	0	1/3															
1	1	1/4															
7	TSnAT07	<p>Enables generation of the A/D conversion trigger (TSnADTRG0 signal) at the (peak) timing when the 16-bit sub-counter switches from count-up operation to count-down operation.</p> <p>0: Don't use 16-bit sub-counter peak timing as A/D conversion trigger. 1: Use 16-bit sub-counter peak timing as A/D conversion trigger.</p> <p><b>Caution:</b></p> <ol style="list-style-type: none"> <li>1. The TSnAT07 bit can be set to 1 only in the high-accuracy T-PWM mode. Be sure to set the TSnAT07 bit to 0 in any other mode.</li> <li>2. Do not set the TSnAT07 bit to 1 when TSnDTC0 ≠ 0000H, TSnDTC1 = 0000H. The A/D conversion trigger is not generated at the peak timing of the 16-bit sub-counter.</li> </ol>															
6	TSnAT06	<p>Enables generation of the A/D conversion trigger (TSnADTRG0 signal) at the (trough) timing when the 16-bit sub-counter switches from count-down operation to count-up operation.</p> <p>0: Don't use 16-bit sub-counter trough timing as A/D conversion trigger. 1: Use 16-bit sub-counter trough timing as A/D conversion trigger.</p> <p><b>Caution:</b></p> <ol style="list-style-type: none"> <li>1. The TSnAT06 bit can be set to 1 only in the high-accuracy T-PWM mode. Be sure to set the TSnAT06 bit to 0 in any other mode.</li> <li>2. Do not set the TSnAT06 bit to 1 when TSnDTC0 = 0000H, TSnDTC1 ≠ 0000H. The A/D conversion trigger is not generated at the trough timing of the 16-bit sub-counter.</li> </ol>															



Table 17-11 TSnOPT2 register contents (2/3)

Bit position	Bit name	Function
5	TSnAT05	<p>Enables generation of the A/D conversion trigger (TSnADTRG0 signal) when the INTTSnCC5 interrupt occurs during count-down operation of the 16-bit counter.</p> <p>0: Don't use occurrence of the INTTSnCC5 interrupt during the 16-bit counter count-down operation as A/D conversion trigger.</p> <p>1: Use occurrence of the INTTSnCC5 interrupt during the 16-bit counter count-down operation as A/D conversion trigger.</p> <hr/> <p><b>Caution:</b> The TSnAT05 bit can be set to 1 only in the triangular-wave PWM mode and high-accuracy T-PWM mode. Be sure to set TSnAT05 to 0 in any other mode.</p> <hr/>
4	TSnAT04	<p>Enables generation of the A/D conversion trigger (TSnADTRG0 signal) when the INTTSnCC5 interrupt occurs during count-up operation of the 16-bit counter.</p> <p>0: Don't use occurrence of the INTTSnCC5 interrupt during the 16-bit counter count-up operation as A/D conversion trigger.</p> <p>1: Use occurrence of the INTTSnCC5 interrupt during the 16-bit counter count-up operation as A/D conversion trigger.</p> <hr/> <p><b>Caution:</b> The TSnAT04 bit can be set to 1 only in the PWM mode, triangular-wave PWM mode, high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode. Be sure to set TSnAT04 to 0 in any other mode.</p> <hr/>
3	TSnAT03	<p>Enables generation of the A/D conversion trigger (TSnADTRG0 signal) when the INTTSnCC4 interrupt occurs during count-down operation of the 16-bit counter.</p> <p>0: Don't use occurrence of the INTTSnCC4 interrupt during the 16-bit counter count-down operation as A/D conversion trigger.</p> <p>1: Use occurrence of the INTTSnCC4 interrupt during the 16-bit counter count-down operation as A/D conversion trigger.</p> <hr/> <p><b>Caution:</b> The TSnAT03 bit can be set to 1 only in the triangular-wave PWM mode and high-accuracy T-PWM mode. Be sure to set TSnAT03 to 0 in any other mode.</p> <hr/>
2	TSnAT02	<p>Enables generation of the A/D conversion trigger (TSnADTRG0 signal) when the INTTSnCC4 interrupt occurs during count-up operation of the 16-bit counter.</p> <p>0: Don't use occurrence of the INTTSnCC4 interrupt during the 16-bit counter count-up operation as A/D conversion trigger.</p> <p>1: Use occurrence of the INTTSnCC4 interrupt during the 16-bit counter count-up operation as A/D conversion trigger.</p> <hr/> <p><b>Caution:</b> The TSnAT02 bit can be set to 1 only in the PWM mode, high-accuracy T-PWM mode, triangular-wave PWM mode, PWM mode with dead-time, 120° excitation mode, special 120° excitation mode, and special pattern output mode. Be sure to set TSnAT02 to 0 in any other mode.</p> <hr/>

**Table 17-11 TSnOPT2 register contents (3/3)**

Bit position	Bit name	Function
1	TSnAT01	<p>Enables the A/D conversion trigger (TSnADTRG0 signal) generation at the switch timing from count-up operation to count-down operation of the 16-bit counter (peak interrupt).</p> <p>0: Don't use the peak interrupt (INTTSnCD0) after thinning out as the A/D conversion trigger.</p> <p>1: Use the peak interrupt (INTTSnCD0) after thinning out as the A/D conversion trigger.</p> <hr/> <p><b>Caution:</b> The TSnAT01 bit can be set to 1 only in the PWM mode, high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode. Be sure to set TSnAT01 to 0 in any other mode.</p> <hr/>
0	TSnAT00	<p>Enables the A/D conversion trigger (TSnADTRG0 signal) generation at the switch timing from count-up operation to count-down operation of the 16-bit counter (trough interrupt).</p> <p>0: Don't use the trough interrupt (INTTSnOD) after thinning out as the A/D conversion trigger.</p> <p>1: Use the trough interrupt (INTTSnOD) after thinning out as the A/D conversion trigger.</p> <hr/> <p><b>Caution:</b> The TSnAT00 bit can be set to 1 only in the triangular-wave PWM mode and high-accuracy T-PWM mode. Be sure to set TSnAT00 to 0 in any other mode.</p> <hr/>

**(9) Timer Sn option registers 3, 3H, 3L (TSnOPT3, TSnOPT3H, TSnOPT3L)**

The TSnOPT3 registers are 16-bit registers that control A/D conversion trigger output (TSnADTRG1 signal).

**Access** These registers can be read or written with a 16-bit, 8-bit, or 1-bit manipulation instruction.

TSnOPT3H enables access to the higher 8 bits of the TSnOPT3 register and TSnOPT3L enables access to the lower 8 bits of the TSnOPT3 register.

**Address** TS0OPT3: FFFFF5A2<sub>H</sub> TS0OPT3H: FFFFF5A3<sub>H</sub> TS0OPT3L: FFFFF5A2<sub>H</sub>  
TS1OPT3: FFFFF5E2<sub>H</sub> TS1OPT3H: FFFFF5E3<sub>H</sub> TS1OPT3L: FFFFF5E2<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. These registers are cleared by any reset.

	15	14	13	12	11	10	9	8
<b>TSnOPT3</b>	0	0	0	0	0	0	TSnACC3	TSnACC2
	R	R	R	R	R	R	R/W	R/W
	7	6	5	4	3	2	1	0
	TSnAT17	TSnAT16	TSnAT15	TSnAT14	TSnAT13	TSnAT12	TSnAT11	TSnAT10
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	7	6	5	4	3	2	1	0
<b>TSnOPT3H</b>	0	0	0	0	0	0	TSnACC3	TSnACC2
	R	R	R	R	R	R	R/W	R/W

	7	6	5	4	3	2	1	0
<b>TSnOPT3L</b>	TSnAT17	TSnAT16	TSnAT15	TSnAT14	TSnAT13	TSnAT12	TSnAT11	TSnAT10
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Caution** The write timing is delayed due to wait insertion during register access under the following conditions if serial write to the TSnOPT3 register is executed during the timer Sn operation (when the value of TSnCE of the TSnCTL0 register is 1).

Write address	Next write address	Write delay
TSnOPT3L	TSnOPT3, TSnOPT3L	Delay
	TSnOPT3H	No delay
TSnOPT3H	TSnOPT3, TSnOPT3H	Delay
	TSnOPT3L	No delay
TSnOPT3	TSnOPT3, TSnOPT3H, TSnOPT3L	Delay

Table 17-12 TSnOPT3 register contents (1/3)

Bit position	Bit name	Function															
9,8	TSnACC3, TSnACC2	Specifies the thinning out ratio of the A/D conversion trigger (TSnADTRG1 signal).															
		<table><tr><th>TSnACC3</th><th>TSnACC2</th><th>Thinning out Ratio</th></tr><tr><td>0</td><td>0</td><td>No thinning out</td></tr><tr><td>0</td><td>1</td><td>1/2</td></tr><tr><td>1</td><td>0</td><td>1/3</td></tr><tr><td>1</td><td>1</td><td>1/4</td></tr></table>	TSnACC3	TSnACC2	Thinning out Ratio	0	0	No thinning out	0	1	1/2	1	0	1/3	1	1	1/4
		TSnACC3	TSnACC2	Thinning out Ratio													
		0	0	No thinning out													
		0	1	1/2													
		1	0	1/3													
		1	1	1/4													
		<b>Note:</b> During the timer Sn operation (when TSnCTL0.TSnCE = 1), the TSnADTRG1 signal thinning out counter is not cleared by write access to the TSnOPT3L register.															
		<b>Caution:</b> During the timer Sn operation (when TSnCTL0.TSnCE = 1), the TSnADTRG1 signal thinning out counter is cleared if write access to the TSnOPT3 and TSnOPT3H registers is executed (including when the same value is written to the TSnACC3 and TSnACC2 bits).															
7	TSnAT17	<p>Enables generation of the A/D conversion trigger (TSnADTRG1 signal) at the (peak) timing when the 16-bit sub-counter switches from count-up operation to count-down operation.</p> <p>0: Don't use 16-bit sub-counter peak timing as A/D conversion trigger. 1: Use 16-bit sub-counter peak timing as A/D conversion trigger.</p>															
		<p><b>Caution:</b></p> <ol style="list-style-type: none"><li>1. The TSnAT17 bit can be set to 1 only in the high-accuracy T-PWM mode. Be sure to set the TSnAT17 bit to 0 in any other mode.</li><li>2. Do not set the TSnAT17 bit to 1 when TSnDTC0 ≠ 0000H, TSnDTC1 = 0000H. The A/D conversion trigger is not generated at the peak timing of the 16-bit sub-counter.</li></ol>															
6	TSnAT16	<p>Enables generation of the A/D conversion trigger (TSnADTRG1 signal) at the (trough) timing when the 16-bit sub-counter switches from count-down operation to count-up operation.</p> <p>0: Don't use 16-bit sub-counter trough timing as A/D conversion trigger. 1: Use 16-bit sub-counter trough timing as A/D conversion trigger.</p>															
		<p><b>Caution:</b></p> <ol style="list-style-type: none"><li>1. The TSnAT16 bit can be set to 1 only in the high-accuracy T-PWM mode. Be sure to set the TSnAT16 bit to 0 in any other mode.</li><li>2. Do not set the TSnAT16 bit to 1 when TSnDTC0 = 0000H, TSnDTC1 ≠ 0000H. The A/D conversion trigger is not generated at the trough timing of the 16-bit sub-counter.</li></ol>															

Table 17-12 TSnOPT3 register contents (2/3)

Bit position	Bit name	Function
5	TSnAT15	<p>Enables generation of the A/D conversion trigger (TSnADTRG1 signal) when the INTTSnCC5 interrupt occurs during count-down operation of the 16-bit counter.</p> <p>0: Don't use occurrence of the INTTSnCC5 interrupt during the 16-bit counter count-down operation as A/D conversion trigger.</p> <p>1: Use occurrence of the INTTSnCC5 interrupt during the 16-bit counter count-down operation as A/D conversion trigger.</p> <hr/> <p><b>Caution:</b> The TSnAT15 bit can be set to 1 only in the triangular-wave PWM mode and high-accuracy T-PWM mode. Be sure to set TSnAT15 to 0 in any other mode.</p> <hr/>
4	TSnAT14	<p>Enables generation of the A/D conversion trigger (TSnADTRG1 signal) when the INTTSnCC5 interrupt occurred during count-up operation of the 16-bit counter.</p> <p>0: Don't use occurrence of the INTTSnCC5 interrupt during the 16-bit counter count-up operation as A/D conversion trigger.</p> <p>1: Use occurrence of the INTTSnCC5 interrupt during the 16-bit counter count-up operation as A/D conversion trigger.</p> <hr/> <p><b>Caution:</b> The TSnAT14 bit can be set to 1 only in the PWM mode, triangular-wave PWM mode, high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode. Be sure to set TSnAT14 to 0 in any other mode.</p> <hr/>
3	TSnAT13	<p>Enables generation of the A/D conversion trigger (TSnADTRG1 signal) when the INTTSnCC4 interrupt occurs during count-down operation of the 16-bit counter.</p> <p>0: Don't use occurrence of the INTTSnCC4 interrupt during the 16-bit counter count-down operation as A/D conversion trigger.</p> <p>1: Use occurrence of the INTTSnCC4 interrupt during the 16-bit counter count-down operation as A/D conversion trigger.</p> <hr/> <p><b>Caution:</b> The TSnAT13 bit can be set to 1 only in the triangular-wave PWM mode and high-accuracy T-PWM mode. Be sure to set TSnAT13 to 0 in any other mode.</p> <hr/>
2	TSnAT12	<p>Enables generation of the A/D conversion trigger (TSnADTRG1 signal) when the INTTSnCC4 interrupt occurs during count-up operation of the 16-bit counter.</p> <p>0: Don't use occurrence of the INTTSnCC4 interrupt during the 16-bit counter count-up operation as A/D conversion trigger.</p> <p>1: Use occurrence of the INTTSnCC4 interrupt during the 16-bit counter count-up operation as A/D conversion trigger.</p> <hr/> <p><b>Caution:</b> The TSnAT12 bit can be set to 1 only in the PWM mode, high-accuracy T-PWM mode, triangular-wave PWM mode, PWM mode with dead-time, 120° excitation mode, special 120° excitation mode, and special pattern output mode. Be sure to set TSnAT12 to 0 in any other mode.</p> <hr/>

**Table 17-12 TSnOPT3 register contents (3/3)**

Bit position	Bit name	Function
1	TSnAT11	<p>Enables the A/D conversion trigger (TSnADTRG1 signal) generation at the switch timing from count-up operation to count-down operation of the 16-bit counter (peak interrupt).</p> <p>0: Don't use occurrence of the peak interrupt (INTTSnCD0) after thinning out as the A/D conversion trigger.</p> <p>1: Use occurrence of the peak interrupt (INTTSnCD0) after thinning out as the A/D conversion trigger.</p> <hr/> <p><b>Caution:</b> The TSnAT11 bit can be set to 1 only in the PWM mode, high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode. Be sure to set TSnAT11 to 0 in any other mode.</p> <hr/>
0	TSnAT10	<p>Enables the A/D conversion trigger (TSnADTRG1 signal) generation at the switch timing from count-up operation to count-down operation of the 16-bit counter (trough interrupt).</p> <p>0: Don't use occurrence of the trough interrupt (INTTSnOD) after thinning out as the A/D conversion trigger.</p> <p>1: Use occurrence of the trough interrupt (INTTSnOD) after thinning out as the A/D conversion trigger.</p> <hr/> <p><b>Caution:</b> The TSnAT10 bit can be set to 1 only in the triangular-wave PWM mode and high-accuracy T-PWM mode. Be sure to set TSnAT10 to 0 in any other mode.</p> <hr/>

**(10) Timer Sn option register 4 (TSnOPT4)**

The TSnOPT4 register is an 8-bit register that controls timer output (TOSn1 to TOSn6) by software and pattern output order.

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

**Address** TS0OPT4: FFFFF588<sub>H</sub>  
TS1OPT4: FFFFF5C8<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>TSnOPT4</b>	TSnSOC	0	0	0	TSnPSC	TSnIPC2	TSnIPC1	TSnIPC0
	R/W	R	R	R	R/W	R/W	R/W	R/W

**Table 17-13 TSnOPT4 register contents (1/2)**

Bit position	Bit name	Function
7	TSnSOC	<p>Selects the timer output (TOSn1 to TOSn6) control by software</p> <p>0: Software control is disabled. 1: Software control is enabled.</p> <p>When the TSnSOC bit is set to 1, timer output is switched to the output pattern of the software control function set by the TSnIPC2 to TSnIPC0 bits. Dead time is ensured by using the dead time 16-bit counter. Refer to “<i>Timer output initial pattern setting in each mode and function</i>” on page 663 for details.</p> <p><b>Note:</b> The setting of the TSnSOC bit is valid in the high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, and special excitation mode.</p> <hr/> <p><b>Caution:</b> The setting of TSnSOC = 1 and TSnADC of the TSnOPT5 register = 1 is prohibited.</p> <hr/>
3	TSnPSC	<p>Selects the pattern output order during semi-automatic driving.</p> <p>0: Switch the timer output (TOSn1 to TOSn6) into the normal rotation order. 1: Switch the timer output (TOSn1 to TOSn6) into the reverse rotation order.</p> <p>The TSnSPC bit specifies the timer output pattern order with output pattern set with the TSnIPC2 to TSnIPC0 bits as the initial pattern in the 120° excitation mode, special 120° excitation mode, and 180° excitation control function. The TSnPSC bit is valid when TSnSOC of the TSnOPT5 register is set to 1, or when TSnSOC is set to 0, TSnPOT of the TSnOPT5 register is set to 1, and TSnPSS is set to 1.</p> <p><b>Note:</b> Refer to “<i>Overview of 120° excitation mode</i>” on page 852 for the normal/reverse rotation output order.</p> <hr/> <p><b>Caution:</b></p> <ol style="list-style-type: none"> <li>1. The TSnPSC bit must be changed when TSnCE is set to 0 or TSnPOT is set to 0. The timer output pattern may differ from the expected one if rewritten when TSnPOT is set to 1.</li> <li>2. TSnCE = 1 setting must be executed after matching the PSC bit order with the input signal variation logic when the signal input to the TAPTSn0 to TAPTSn2 pins varies while the timer Sn operation is stopped (when the TSnCE of the TSnCTL0 register is set to 0).</li> </ol> <hr/>

**Table 17-13 TSnOPT4 register contents (2/2)**

Bit position	Bit name	Function			
2 to 0	TSnIPC [2:0]	Set the initial pattern of the timer output in the 120° excitation mode, special 120° excitation mode and 180° excitation control function. Refer to “ <i>Timer output initial pattern setting in each mode and function</i> ” on page 663 for details.			
		<b>TSnIPC2</b>	<b>TSnIPC1</b>	<b>TSnIPC0</b>	<b>Operation</b>
		0	0	0	Inactive output if there is no change from the initial value (000B). Output level remains the level before the change if there is a change from any value to the initial value (000B).
		1	1	1	The output level remains the level before the change.
		Other than above			Timer output based on the pattern.



(a) Function of the TSnIPC2 to TSnIPC0 bits and output of TOSn1 to TOSn6 pins by setting the TSnSOC and TSnADC bits

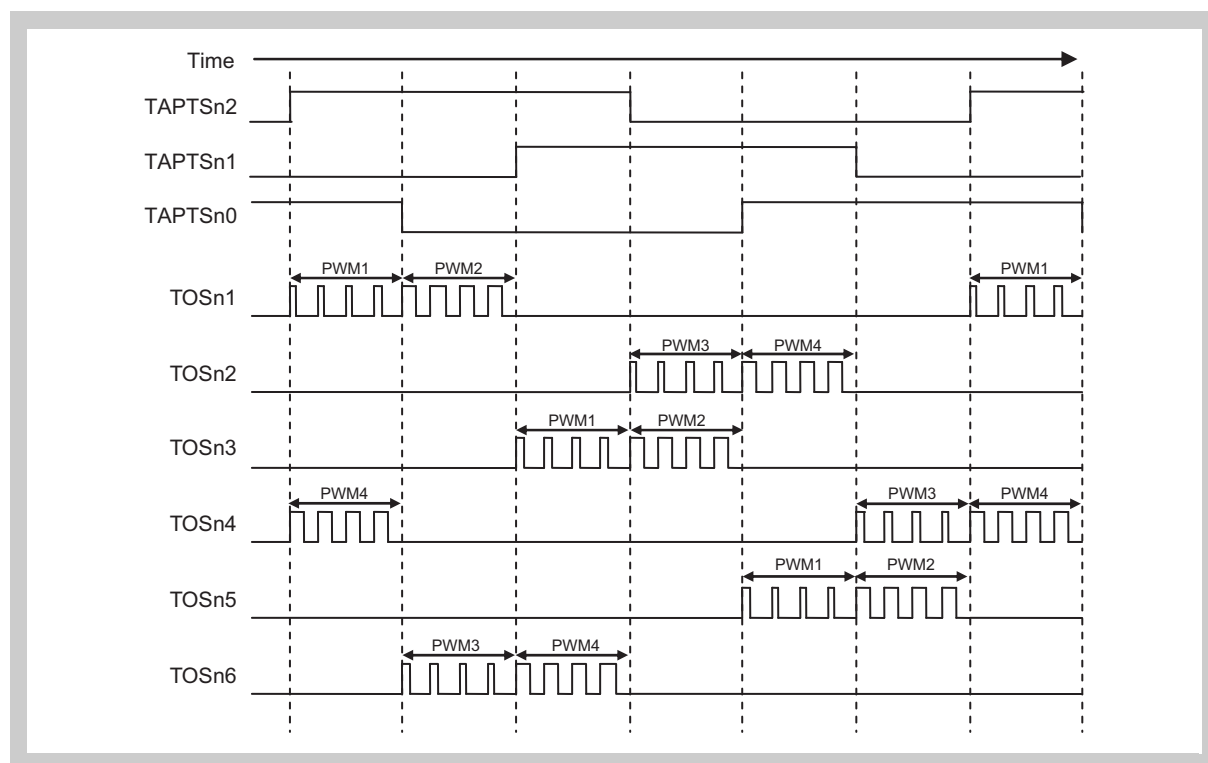
Operation mode	TSnOPT4 . TSnSOC	TSnOPT5. TSnADC <sup>Note</sup>	Function of TSnOPT4.TSnIPC[2:0]	Output of TOSn1 to TOSn6 pins
High-accuracy T-PWM mode	1	0	Pattern setting	Software 180° excitation
	0	1	Initial pattern setting	Semi-automatic driving 180° excitation
	0	0	Not used	High-accuracy T-PWM
	1	1	Setting prohibited	
PWM mode with dead time	1	0	Pattern setting	Software 180° excitation
	0	1	Setting prohibited	
	0	0	Not used	PWM with dead time
	1	1	Setting prohibited	
120° excitation mode	1	0	Pattern setting	Software 180° excitation
	0	1	Initial pattern setting	Semi-automatic driving 180° excitation
	0	0	Pattern setting	120° excitation
	1	1	Setting prohibited	
Special 120° excitation mode	1	0	Pattern setting for software output controlling	Software 180° excitation
	0	1	Initial pattern setting	Semi-automatic driving 180° excitation
	0	0	Initial pattern setting	Special 120° excitation mode
	1	1	Setting prohibited	

**Caution** The TSnOPT5.TSnPOT bit must be set to 1 if setting the TSnIPC[2:0] bits while using the 180° excitation control function.

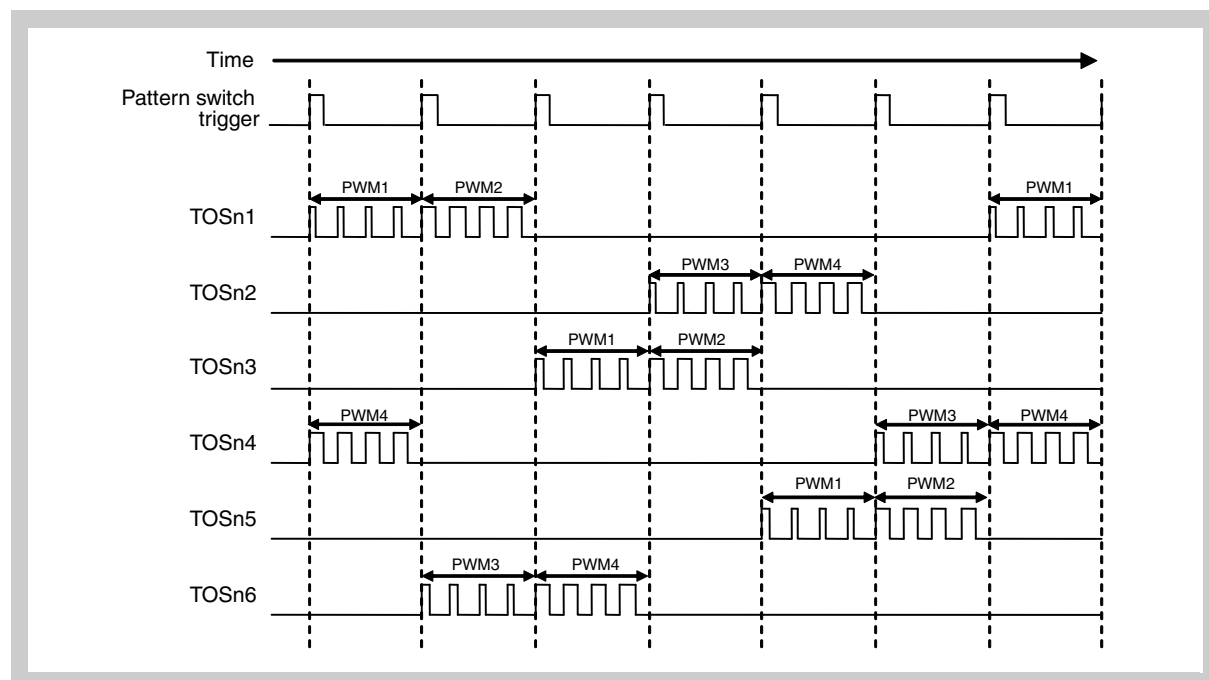
**(b) Timer output order in each mode and function**

- In 120° excitation mode  
The TSnOPT4.TSnIPC[2:0] bits are set when TSnOPT5.TSnPOT bit is set to 1 while the timer Sn operation is stopped (when TSnCTL0.TSnCE is set to 0), and then timer output starts with the set pattern when the TSnCTL0.TSnCE bit is set to 1.

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,  
TSnOPT5 register: TSnPOT = 0, TSnPSS = 1,  
TSnOPT7 register: TSnIDC = 0

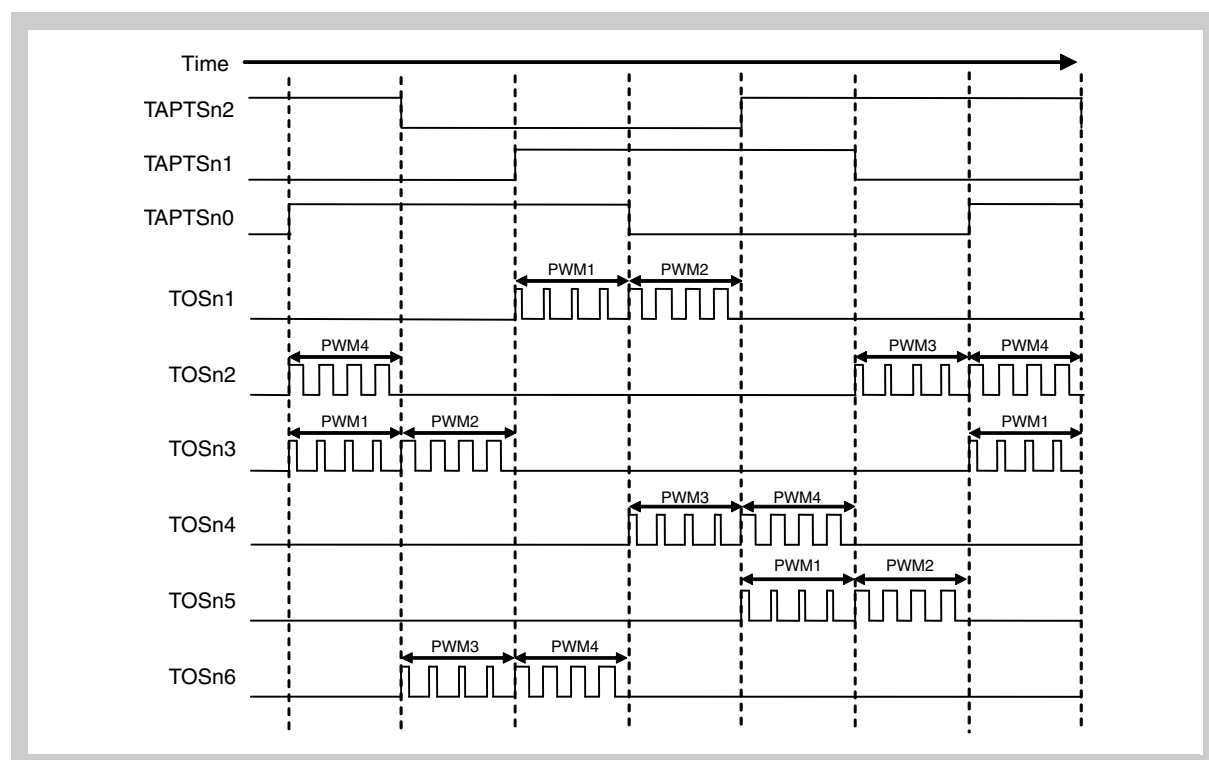


Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,  
 TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
 TSnOPT7 register: TSnIDC = 0

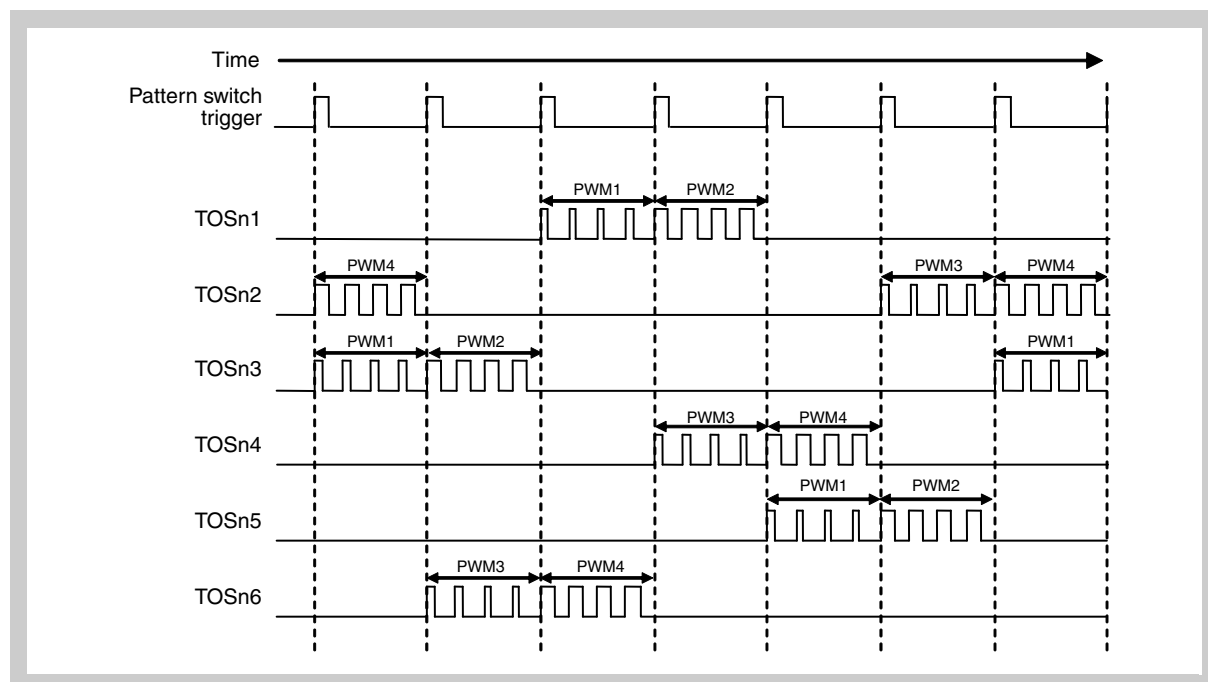


**Note** PWM1 to PWM4 indicate PWM output using the 16-bit counter and the TSnCCR1 to TSnCCR4 registers.

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,  
 TSnOPT5 register: TSnPOT = 0, TSnPSS = 1,  
 TSnOPT7 register: TSnIDC = 1



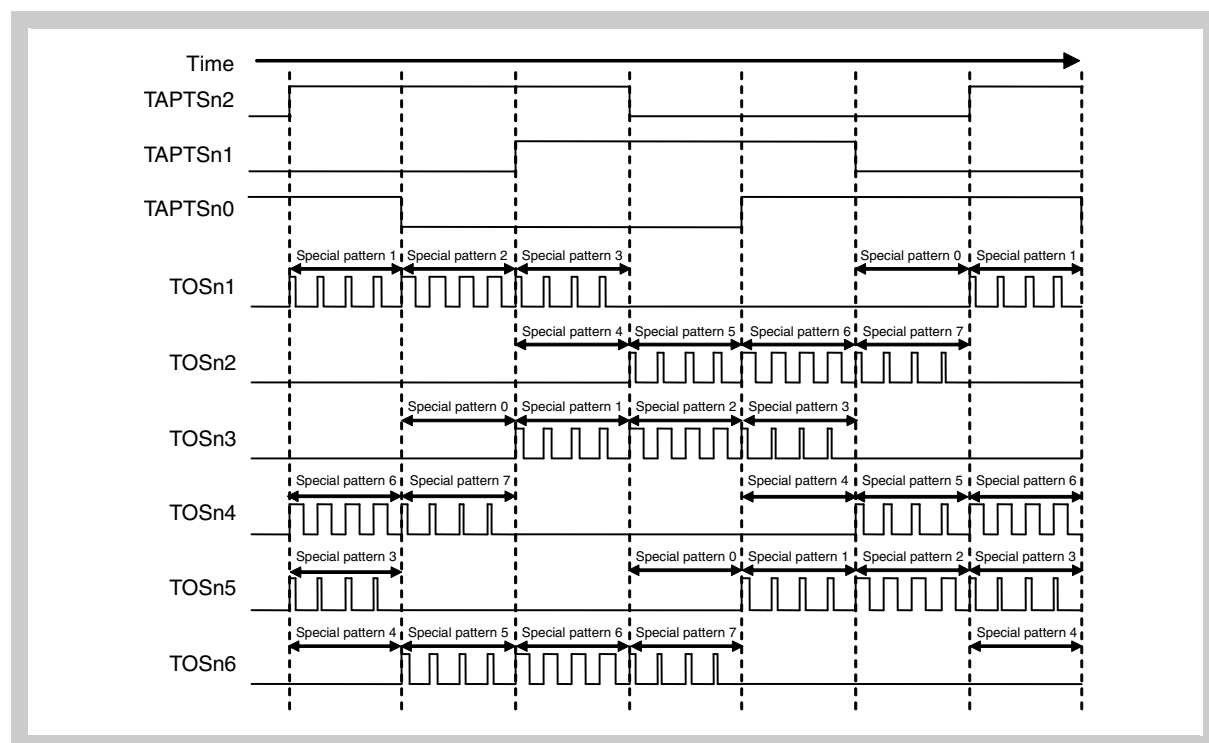
Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,  
 TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
 TSnOPT7 register: TSnIDC = 1



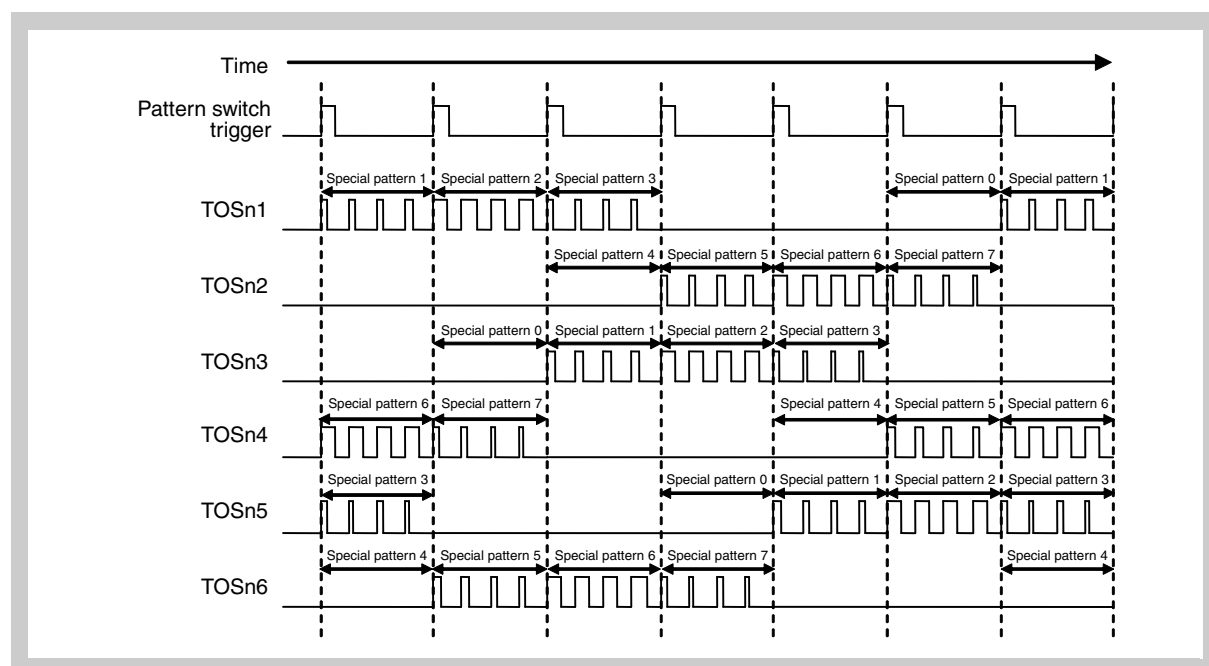
**Note** PWM1 to PWM4 indicate PWM output using the 16-bit counter and the TSnCCR1 to TSnCCR4 registers.

- In special 120° excitation mode

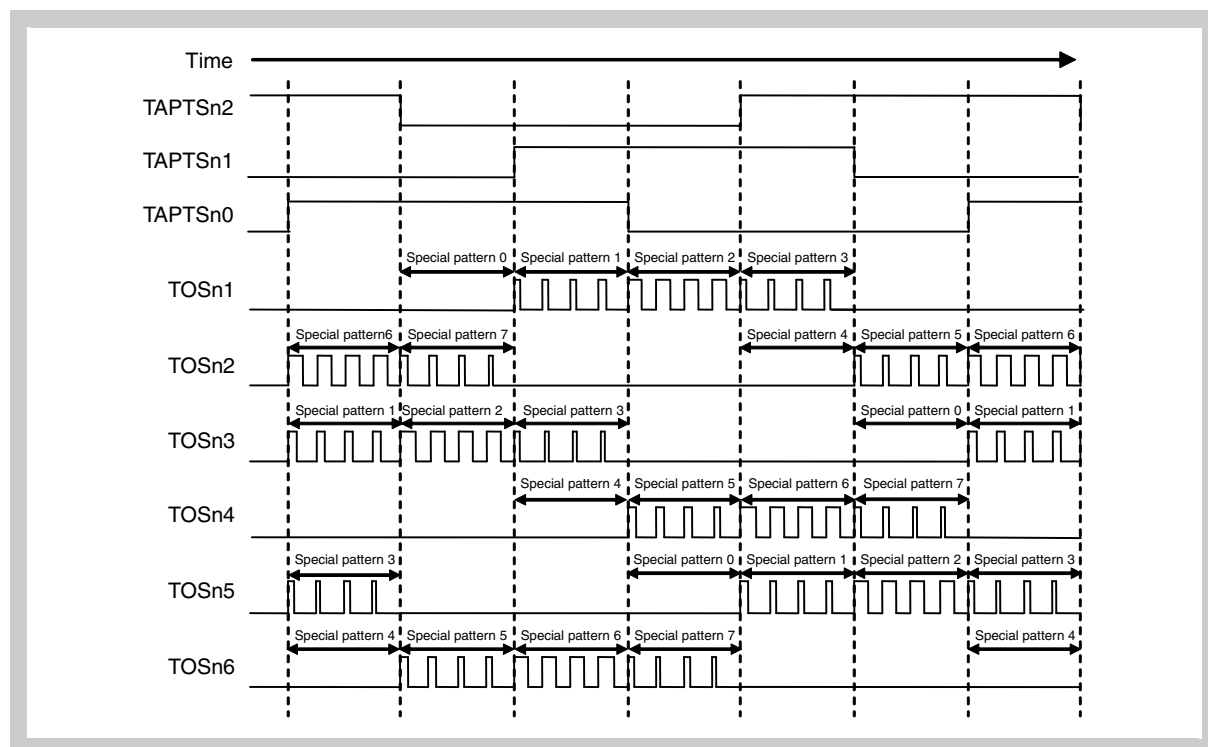
Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,  
 TSnOPT5 register: TSnPOT = 0, TSnPSS = 1,  
 TSnOPT7 register: TSnIDC = 0



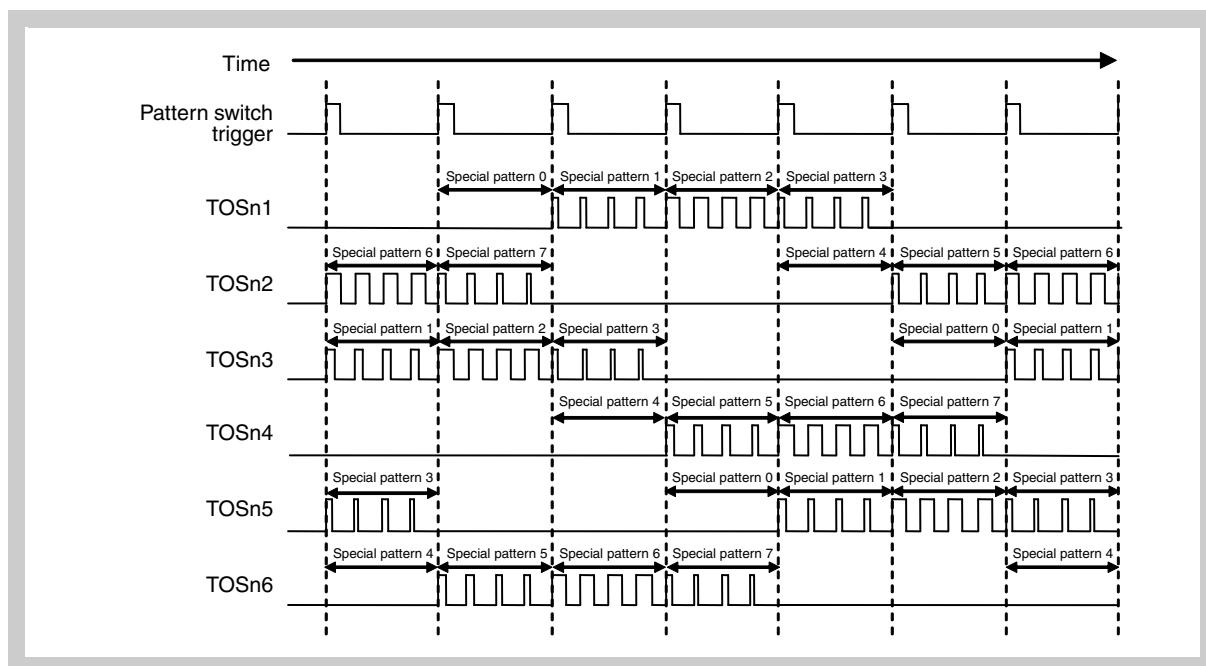
Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,  
 TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
 TSnOPT7 register: TSnIDC = 0



Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,  
 TSnOPT5 register: TSnPOT = 0, TSnPSS = 1,  
 TSnOPT7 register: TSnIDC = 1



Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,  
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
TSnOPT7 register: TSnIDC = 1



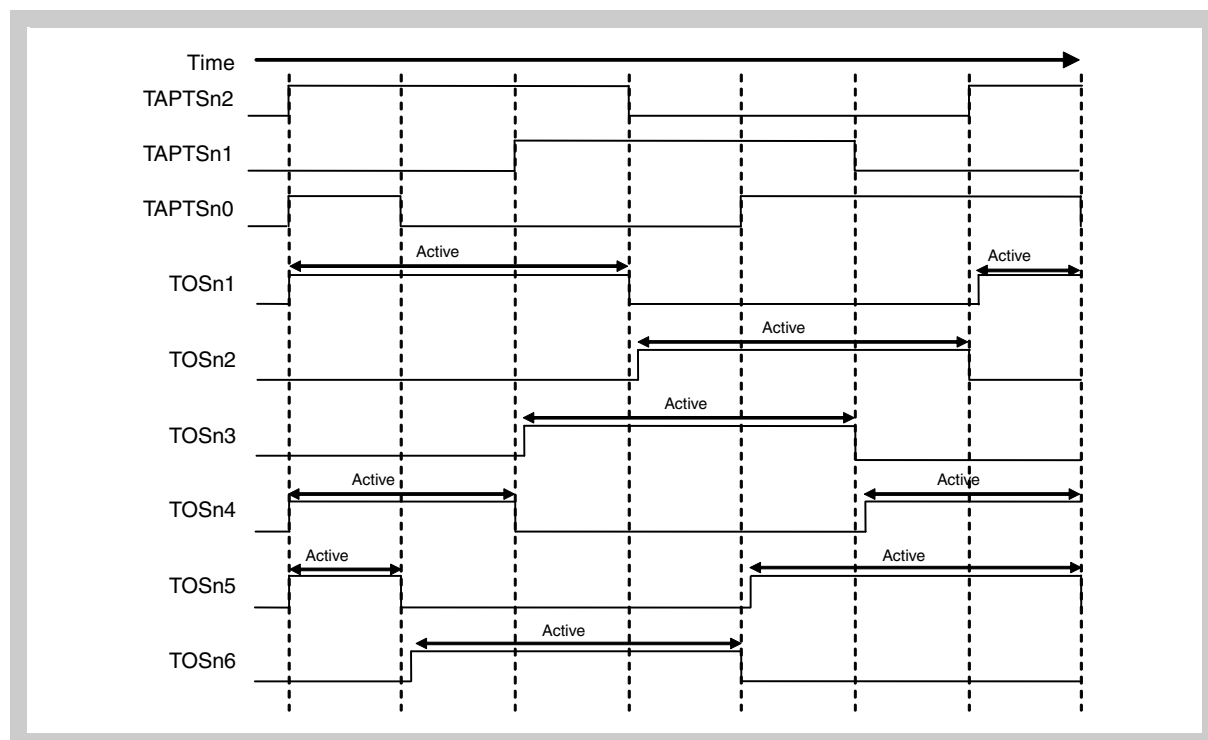
**Caution** Special pattern 3 and special pattern 4 cannot be active simultaneously. Special pattern 0 and special pattern 7 cannot be active simultaneously either. The simultaneous active setting results in the following operations.

1. Only TOSn1, TOSn3, and TOSn5 are active during the simultaneous active period if TOSn1, TOSn3, and TOSn5 become active prior to TOSn2, TOSn4, and TOSn6.
2. Only TOSn2, TOSn4, and TOSn6 are active during the simultaneous active period if TOSn2, TOSn4, and TOSn6 become active prior to TOSn1, TOSn3, and TOSn5.
3. Only TOSn1, TOSn3, and TOSn5 are active during the simultaneous active period if TOSn1, TOSn3, TOSn5, and TOSn2, TOSn4, TOSn6 become active simultaneously.

**Note** Special patterns 0 to 3 are output using the TSnPAT0 register.  
Special pattern 0: Bits 3 to 0  
Special pattern 1: Bits 7 to 4  
Special pattern 2: Bits 11 to 8  
Special pattern 3: Bits 15 to 12  
Special patterns 4 to 7 are output using the TSnPAT1 register.  
Special pattern 4: Bits 3 to 0  
Special pattern 5: Bits 7 to 4  
Special pattern 6: Bits 11 to 8  
Special pattern 7: Bits 15 to 12

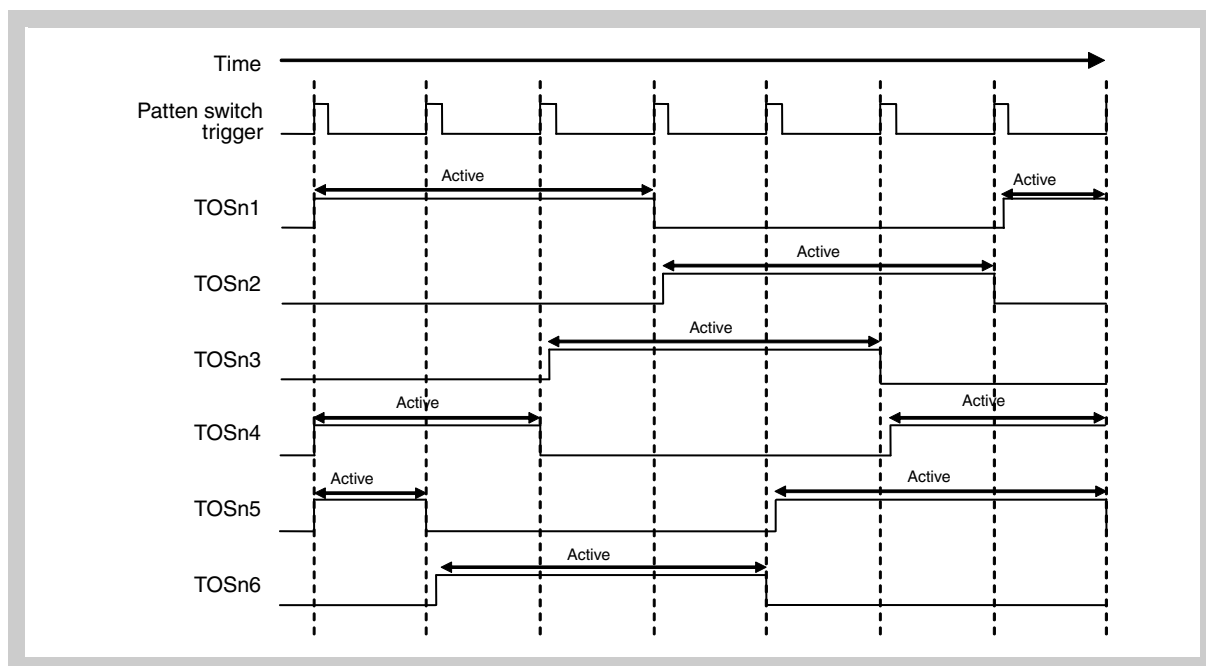
- When using 180° excitation control function  
The TSnIPC[2:0] bits start timer output with the setting pattern when TSnADC is set to 1.

Normal rotation: TSnOPT4 register: TSnSOC = 0,  
TSnOPT5 register: TSnADC = 1, TSnPOT = 0,  
TSnOPT7 register: TSnIDC = 0

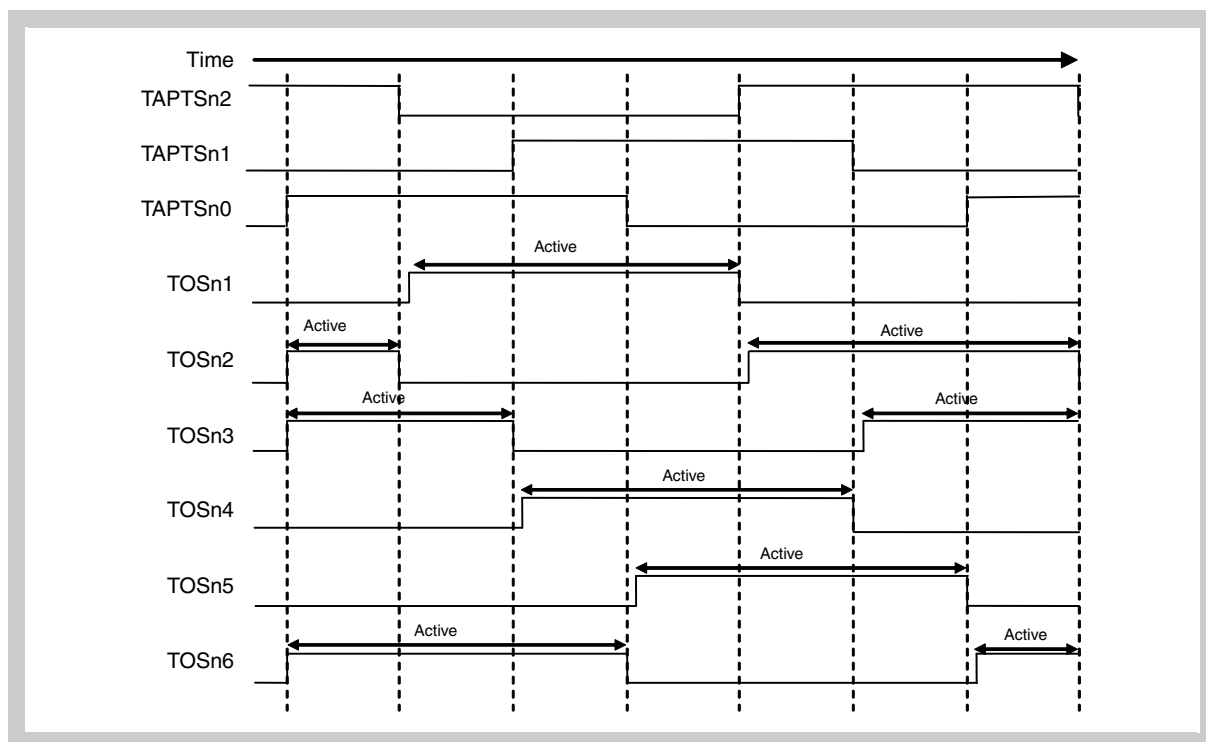




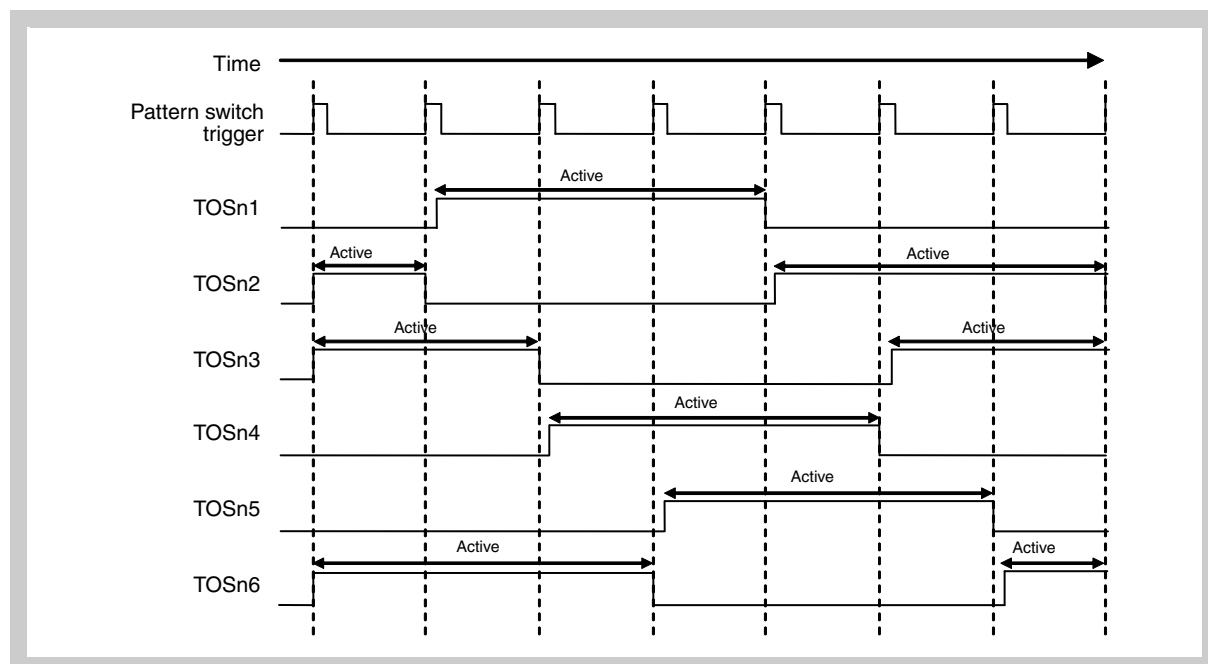
Normal rotation: TSnOPT4 register: TSnSOC = 0,  
 TSnOPT5 register: TSnADC = 1, TSnPOT = 1,  
 TSnOPT7 register: TSnIDC = 0



Reverse rotation: TSnOPT4 register: TSnSOC = 0,  
 TSnOPT5 register: TSnADC = 1, TSnPOT = 0,  
 TSnOPT7 register: TSnIDC = 1



Reverse rotation: TSnOPT4 register: TSnSOC = 0,  
 TSnOPT5 register: TSnADC = 1, TSnPOT = 1,  
 TSnOPT7 register: TSnIDC = 1



Each phase (TOSn1 to TOSn6) outputs the active level for the period of electric angle 180° and then outputs the inactive level for the period of electric angle 180°.

**(c) Timer output initial pattern setting in each mode and function**

- In 120° excitation mode

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,  
 TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
 TSnOPT7 register: TSnIDC = 0

→ Pattern switch order

Output pin	TSnIPC2 to TSnIPC0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	PWM1	PWM2	INACT	INACT	INACT	INACT	Note 1	Note 2
TOSn2	INACT	INACT	INACT	PWM3	PWM4	INACT	Note 1	Note 2
TOSn3	INACT	INACT	PWM1	PWM2	INACT	INACT	Note 1	Note 2
TOSn4	PWM4	INACT	INACT	INACT	INACT	PWM3	Note 1	Note 2
TOSn5	INACT	INACT	INACT	INACT	PWM1	PWM2	Note 1	Note 2
TOSn6	INACT	PWM3	PWM4	INACT	INACT	INACT	Note 1	Note 2

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,  
 TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
 TSnOPT7 register: TSnIDC = 1

→ Pattern switch order

Output pin	TSnIPC2 to TSnIPC0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	INACT	PWM1	PWM2	INACT	Note 1	Note 2
TOSn2	PWM3	PWM4	INACT	INACT	INACT	INACT	Note 1	Note 2
TOSn3	PWM2	INACT	INACT	INACT	INACT	PWM1	Note 1	Note 2
TOSn4	INACT	INACT	PWM3	PWM4	INACT	INACT	Note 1	Note 2
TOSn5	INACT	PWM1	PWM2	INACT	INACT	INACT	Note 1	Note 2
TOSn6	INACT	INACT	INACT	INACT	PWM3	PWM4	Note 1	Note 2

- Note**
1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], the output remains the level before the change.
  2. Output remains the level before the change.
  3. PWM1 to PWM4: PWM output from TSnCCR1 to TSnCCR4 registers
  4. INACT: Inactive level output

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,  
 TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
 TSnOPT7 register: TSnIDC = 0

← Pattern switch order

Output pin	TSnIPC2 to TSnIPC0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	PWM2	PWM1	INACT	INACT	INACT	INACT	Note 1	Note 2
TOSn2	INACT	INACT	INACT	PWM4	PWM3	INACT	Note 1	Note 2
TOSn3	INACT	INACT	PWM2	PWM1	INACT	INACT	Note 1	Note 2
TOSn4	PWM3	INACT	INACT	INACT	INACT	PWM4	Note 1	Note 2
TOSn5	INACT	INACT	INACT	INACT	PWM2	PWM1	Note 1	Note 2
TOSn6	INACT	PWM4	PWM3	INACT	INACT	INACT	Note 1	Note 2

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,  
 TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
 TSnOPT7 register: TSnIDC = 1

← Pattern switch order

Output pin	TSnIPC2 to TSnIPC0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	INACT	PWM2	PWM1	INACT	Note 1	Note 2
TOSn2	PWM4	PWM3	INACT	INACT	INACT	INACT	Note 1	Note 2
TOSn3	PWM1	INACT	INACT	INACT	INACT	PWM2	Note 1	Note 2
TOSn4	INACT	INACT	PWM4	PWM3	INACT	PWM3	Note 1	Note 2
TOSn5	INACT	PWM2	PWM1	INACT	INACT	INACT	Note 1	Note 2
TOSn6	INACT	INACT	INACT	INACT	PWM4	PWM3	Note 1	Note 2

- Note**
1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], the output remains the level before the change.
  2. Output remains the level before the change.
  3. PWM1 to PWM4: PWM output from TSnCCR1 to TSnCCR4 registers
  4. INACT: Inactive level output

- In special 120° excitation mode

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,  
 TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
 TSnOPT7 register: TSnIDC = 0

→ Pattern switch order

Output pin	TSnIPC2 to TSnIPC0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	Special pattern 1	Special pattern 2	Special pattern 3	INACT	INACT	Special pattern 0	Note 1	Note 2
TOSn2	INACT	INACT	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	Note 1	Note 2
TOSn3	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Special pattern 3	INACT	Note 1	Note 2
TOSn4	Special pattern 6	Special pattern 7	INACT	INACT	Special pattern 4	Special pattern 5	Note 1	Note 2
TOSn5	Special pattern 3	INACT	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Note 1	Note 2
TOSn6	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	INACT	INACT	Note 1	Note 2

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,  
 TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
 TSnOPT7 register: TSnIDC = 1

→ Pattern switch order

Output pin	TSnIPC2 to TSnIPC0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Special pattern 3	Note 1	Note 2
TOSn2	Special pattern 5	Special pattern 6	Special pattern 7	INACT	INACT	Special pattern 4	Note 1	Note 2
TOSn3	Special pattern 2	Special pattern 3	INACT	INACT	Special pattern 0	Special pattern 1	Note 1	Note 2
TOSn4	INACT	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	INACT	Note 1	Note 2
TOSn5	Special pattern 0	Special pattern 1	Special pattern 2	Special pattern 3	INACT	INACT	Note 1	Note 2
TOSn6	Special pattern 7	INACT	INACT	Special pattern 4	Special pattern 5	Special pattern 6	Note 1	Note 2

- Note**
1. INACT if no change from the initial value. If changed from any value to [0, 0, 0], output remains the level before the change.
  2. Output remains the level before the change.

**Caution** Special pattern 3 and special pattern 4 cannot be active simultaneously. Special pattern 0 and special pattern 7 cannot be active simultaneously either. The simultaneous active setting results in the following operations.

1. Only TOSn1, TOSn3, and TOSn5 are active during the simultaneous active period if TOSn1, TOSn3, and TOSn5 become active prior to TOSn2, TOSn4, and TOSn6.
2. Only TOSn2, TOSn4, and TOSn6 are active during the simultaneous active period if TOSn2, TOSn4 and TOSn6 become active prior to TOSn1, TOSn3, and TOSn5.
3. Only TOSn1, TOSn3, and TOSn5 are active during the simultaneous active period if TOSn1, TOSn3, TOSn5, and TOSn2, TOSn4, TOSn6 become active simultaneously.

- Note**
1. Special patterns 0 to 7: Special pattern output by TSnCCR0 to TSnCCR3, TSnPAT0, and TSnPAT1 registers
  2. INACT: Inactive level output

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,  
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
TSnOPT7 register: TSnIDC = 0

← Pattern switch order

Output pin	TSnIPC2 to TSnIPC0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	Special pattern 2	Special pattern 1	Special pattern 0	INACT	INACT	Special pattern 3	Note 1	Note 2
TOSn2	INACT	INACT	Special pattern 7	Special pattern 6	Special pattern 5	Special pattern 4	Note 1	Note 2
TOSn3	INACT	Special pattern 3	Special pattern 2	Special pattern 1	Special pattern 0	INACT	Note 1	Note 2
TOSn4	Special pattern 5	Special pattern 4	INACT	INACT	Special pattern 7	Special pattern 6	Note 1	Note 2
TOSn5	Special pattern 0	INACT	INACT	Special pattern 3	Special pattern 2	Special pattern 1	Note 1	Note 2
TOSn6	Special pattern 7	Special pattern 6	Special pattern 5	Special pattern 4	INACT	INACT	Note 1	Note 2

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,  
 TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
 TSnOPT7 register: TSnIDC = 1

← Pattern switch order

Output pin	TSnIPC2 to TSnIPC0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	Special pattern 3	Special pattern 2	Special pattern 1	Special pattern 0	Note 1	Note 2
TOSn2	Special pattern 6	Special pattern 5	Special pattern 4	INACT	INACT	Special pattern 7	Note 1	Note 2
TOSn3	Special pattern 1	Special pattern 0	INACT	INACT	Special pattern 3	Special pattern 2	Note 1	Note 2
TOSn4	INACT	Special pattern 7	Special pattern 6	Special pattern 5	Special pattern 4	INACT	Note 1	Note 2
TOSn5	Special pattern 3	Special pattern 2	Special pattern 1	Special pattern 0	INACT	INACT	Note 1	Note 2
TOSn6	Special pattern 4	INACT	INACT	Special pattern 7	Special pattern 6	Special pattern 5	Note 1	Note 2

- Note**
1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], output remains the level before the change.
  2. Output remains the level before the change.

**Caution** Special pattern 3 and special pattern 4 cannot be active simultaneously. Special pattern 0 and special pattern 7 cannot be active simultaneously either. The simultaneous active setting leads the following operations.

1. Only TOSn1, TOSn3, and TOSn5 are active during the simultaneous active period if TOSn1, TOSn3, and TOSn5 become active prior to TOSn2, TOSn4, and TOSn6.
2. Only TOSn2, TOSn4, and TOSn6 are active during the simultaneous active period if TOSn2, TOSn4, and TOSn6 become active prior to TOSn1, TOSn3, and TOSn5.
3. Only TOSn1, TOSn3, and TOSn5 are active during the simultaneous active period if TOSn1, TOSn3, TOSn5, and TOSn2, TOSn4, TOSn6 become active simultaneously.

- Note**
1. Special patterns 0 to 7: Special pattern output by TSnCCR0 to TSnCCR3, TSnPAT0, and TSnPAT1 registers
  2. INACT: Inactive level output

- When using 180° excitation function

TSnOPT4 register: TSnSOC = 0,

TSnOPT5 register: TSnADC = 1, TSnPOT = 1,

TSnOPT7 register: TSnIDC = 0

Output pin	TSnIPC2 to TSnIPC0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	ACT	ACT	ACT	INACT	INACT	INACT	Note 1	Note 2
TOSn2	INACT	INACT	INACT	ACT	ACT	ACT	Note 1	Note 2
TOSn3	INACT	INACT	ACT	ACT	ACT	INACT	Note 1	Note 2
TOSn4	ACT	ACT	INACT	INACT	INACT	ACT	Note 1	Note 2
TOSn5	ACT	INACT	INACT	INACT	ACT	ACT	Note 1	Note 2
TOSn6	INACT	ACT	ACT	ACT	INACT	INACT	Note 1	Note 2

TSnOPT4 register: TSnSOC = 0,

TSnOPT5 register: TSnADC = 1, TSnPOT = 1,

TSnOPT7 register: TSnIDC = 1

Output pin	TSnIPC2 to TSnIPC0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	INACT	ACT	ACT	ACT	Note 1	Note 2
TOSn2	ACT	ACT	ACT	INACT	INACT	INACT	Note 1	Note 2
TOSn3	ACT	ACT	INACT	INACT	INACT	ACT	Note 1	Note 2
TOSn4	INACT	INACT	ACT	ACT	ACT	INACT	Note 1	Note 2
TOSn5	INACT	ACT	ACT	ACT	INACT	INACT	Note 1	Note 2
TOSn6	ACT	INACT	INACT	INACT	ACT	ACT	Note 1	Note 2

- Note**
1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], output remains the level before the change.
  2. Output remains the level before the change.
  3. ACT: Active level output
  4. INACT: Inactive level output



**(11) Timer Sn option register 5 (TSnOPT5)**

The TSnOPT5 register is an 8-bit register that controls 180° excitation control function and timer output (TOSn1 to TOSn6) pattern.

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

**Address** TS0OPT5: FFFFF589<sub>H</sub>  
TS1OPT5: FFFFF5C9<sub>H</sub>







**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

7	6	5	4	3	2	1	0
TSnADC	TSnPOT	TSnPSS	TSnPTS	TSnTSF	TSnOPF2	TSnOPF1	TSnOPF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 17-14 TSnOPT5 register contents (1/2)**

Bit position	Bit name	Function
7	TSnADC	<p>Selects the 180° excitation control function of semi-automatic driving.</p> <p>0: Semi-automatic driving 180° excitation control function disabled 1: Semi-automatic driving 180° excitation control function enabled</p> <p>When the TSnADC bit is set to 1, the timer output pattern is switched to the pattern preset by the TSnOPT4.TSnIPC[2:0] bits. Upon input of the next pattern output trigger, the next pattern is output.</p> <p><b>Note:</b> The TSnADC bit is valid in the high-accuracy T-PWM mode, 120° excitation mode, and special 120° excitation mode.</p> <hr/> <p><b>Caution:</b> 1. Do not set the TSnOPT4.TSnSOC bit to 1.</p> <p>2. Set the output pattern to TSnIPC[2:0] bits in advance before setting TSnADC to 1.</p> <hr/>
6	TSnPOT	<p>Selects the pattern output trigger.</p> <p>0: Output pattern switch by using external pattern input pins (TAPTSn0 to TAPTSn2 pins) (pattern switch method). 1: Output pattern switch by rising edge of the TSnSTCI0 and TSnSTCI1 signals (trigger switch method).</p> <p><b>Note:</b> The TSnPOT bit setting is valid in the 120° excitation mode, special 120° excitation mode and during 180° excitation control in each mode.</p>
5	TSnPSS	<p>Selects the pattern output switch factor.</p> <p>0: Switch the pattern output order using input signal to the TSnARD signal. 1: Switch the pattern output order using the TSnOPT4.TSnPSC bit.</p> <hr/> <p><b>Caution:</b> The TSnPSS bit is valid only when the TSnPOT bit is set to 1.</p> <hr/>

Table 17-14 TSnOPT5 register contents (2/2)

Bit position	Bit name	Function						
4	TSnPTS	<p>Selects enabling/disabling toggle output of the TSnAEDO and TSnESG signals.</p> <p>0: Disable toggle output to the TSnAEDO and TSnESG<sup>Note</sup> signals at the edge detection timing of TAPTSn[2:0] pins.</p> <p>1: Enable toggle output to the TSnAEDO and TSnESG<sup>Note</sup> signals at the edge detection timing of the TAPTSn[2:0] pins.</p> <p>The level of the TSnAEDO and TSnESG signals is retained if the TSnPTS bit is changed from 1 to 0 during the timer Sn operation (when TSnCTL0.TSnCE = 1).</p> <p><b>Note:</b> The TSnESG signal is provided only in the 120° excitation mode and special 120° excitation mode.</p>						
3	TSnTSF	<p>This is a flag that indicates the order of pattern change detected with the TAPTSn[2:0] pins.</p> <p>0: TAPTSn[2:0] pins are input in the normal rotation order.</p> <p>1: TAPTSn[2:0] pins are input in the reverse rotation order.</p> <table><tr><td>Normal rotation</td><td></td></tr><tr><td>Reverse rotation</td><td></td></tr><tr><td>TAPTSn[2:0] pins</td><td>[1, 0, 1] [1, 0, 0] [1, 1, 0] [0, 1, 0] [0, 1, 1] [0, 0, 1]</td></tr></table> <p><b>Note:</b> The normal rotation or reverse rotation can be detected after the first change of the TAPTSn[2:0] pins after TSnCTL0.TSnCE = 1 is set.</p> <p>For details, refer to “Input pattern change order detection” on page 679.</p>	Normal rotation		Reverse rotation		TAPTSn[2:0] pins	[1, 0, 1] [1, 0, 0] [1, 1, 0] [0, 1, 0] [0, 1, 1] [0, 0, 1]
Normal rotation								
Reverse rotation								
TAPTSn[2:0] pins	[1, 0, 1] [1, 0, 0] [1, 1, 0] [0, 1, 0] [0, 1, 1] [0, 0, 1]							
2 to 0	TSnOPF [2:0]	<p>These are flags that indicate the pattern output of the timer output (TOSn1 to TOSn6). Refer to “Timer output pattern in each mode and function” on page 671 for details.</p> <p>The TSnOPF[2:0] bits are displayed in the 120° excitation mode, special 120° excitation mode, and when using 180° excitation control function.</p>						

**(a) Timer output pattern in each mode and function**

- Software control function

TSnOPT4 register: TSnSOC = 1, TSnOPT7 register: TSnIDC = 0

Output pin	TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	ACT	ACT	ACT	INACT	INACT	INACT	Note 1	Note 2
TOSn2	INACT	INACT	INACT	ACT	ACT	ACT	Note 1	Note 2
TOSn3	INACT	INACT	ACT	ACT	ACT	INACT	Note 1	Note 2
TOSn4	ACT	ACT	INACT	INACT	INACT	ACT	Note 1	Note 2
TOSn5	ACT	INACT	INACT	INACT	ACT	ACT	Note 1	Note 2
TOSn6	INACT	ACT	ACT	ACT	INACT	INACT	Note 1	Note 2

TSnOPT4 register: TSnSOC = 1, TSnOPT7 register: TSnIDC = 1

Output pin	TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	INACT	ACT	ACT	ACT	Note 1	Note 2
TOSn2	ACT	ACT	ACT	INACT	INACT	INACT	Note 1	Note 2
TOSn3	ACT	ACT	INACT	INACT	INACT	ACT	Note 1	Note 2
TOSn4	INACT	INACT	ACT	ACT	ACT	INACT	Note 1	Note 2
TOSn5	INACT	ACT	ACT	ACT	INACT	INACT	Note 1	Note 2
TOSn6	ACT	INACT	INACT	INACT	ACT	ACT	Note 1	Note 2

- Note**
1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], output remains the output level before the change.
  2. Output level remains the level before the change.
  3. ACT: Active level output
  4. INACT: Inactive level output

- In 120° excitation mode

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnARD signal or  
TSnPSC = 0,  
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
TSnOPT7 register: TSnIDC = 0

→ Pattern switch order

Output pin	TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	PWM1	PWM2	INACT	INACT	INACT	INACT	Note 1	Note 2
TOSn2	INACT	INACT	INACT	PWM3	PWM4	INACT	Note 1	Note 2
TOSn3	INACT	INACT	PWM1	PWM2	INACT	INACT	Note 1	Note 2
TOSn4	PWM4	INACT	INACT	INACT	INACT	PWM3	Note 1	Note 2
TOSn5	INACT	INACT	INACT	INACT	PWM1	PWM2	Note 1	Note 2
TOSn6	INACT	PWM3	PWM4	INACT	INACT	INACT	Note 1	Note 2

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnARD signal or  
TSnPSC = 0,  
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
TSnOPT7 register: TSnIDC = 1

→ Pattern switch order

Output pin	TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	INACT	PWM1	PWM2	INACT	Note 1	Note 2
TOSn2	PWM3	PWM4	INACT	INACT	INACT	INACT	Note 1	Note 2
TOSn3	PWM2	INACT	INACT	INACT	INACT	PWM1	Note 1	Note 2
TOSn4	INACT	INACT	PWM3	PWM4	INACT	INACT	Note 1	Note 2
TOSn5	INACT	PWM1	PWM2	INACT	INACT	INACT	Note 1	Note 2
TOSn6	INACT	INACT	INACT	INACT	PWM3	PWM4	Note 1	Note 2

- Note**
1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], output remains the output level before the change.
  2. Output remains the level before the change.
  3. PWM1 to PWM4: PWM output from TSnCCR1 to TSnCCR4 registers
  4. INACT: Inactive level output

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnARD signal or  
TSnPSC = 1,  
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
TSnOPT7 register: TSnIDC = 0

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnARD signal or  
TSnPSC = 1,  
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
TSnOPT7 register: TSnIDC = 1

← Pattern switch order

Output pin	TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	PWM2	PWM1	INACT	INACT	INACT	INACT	Note 1	Note 2
TOSn2	INACT	INACT	INACT	PWM4	PWM3	INACT	Note 1	Note 2
TOSn3	INACT	INACT	PWM 2	PWM 1	INACT	INACT	Note 1	Note 2
TOSn4	PWM3	INACT	INACT	INACT	INACT	PWM4	Note 1	Note 2
TOSn5	INACT	INACT	INACT	INACT	PWM2	PWM1	Note 1	Note 2
TOSn6	INACT	PWM4	PWM3	INACT	INACT	INACT	Note 1	Note 2

← Pattern switch order

Output pin	TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	INACT	PWM 2	PWM 1	INACT	Note 1	Note 2
TOSn2	PWM4	PWM3	INACT	INACT	INACT	INACT	Note 1	Note 2
TOSn3	PWM1	INACT	INACT	INACT	INACT	PWM2	Note 1	Note 2
TOSn4	INACT	INACT	PWM4	PWM3	INACT	INACT	Note 1	Note 2
TOSn5	INACT	PWM 2	PWM1	INACT	INACT	INACT	Note 1	Note 2
TOSn6	INACT	INACT	INACT	INACT	PWM4	PWM3	Note 1	Note 2

- Note**
1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], output remains the level before the change.
  2. Output remains the level before the change.
  3. PWM1 to PWM4: PWM output from TSnCCR1 to TSnCCR4 registers
  4. INACT: Inactive level output

- In special 120° excitation mode

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnARD signal or  
 TSnPSC = 0,  
 TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
 TSnOPT7 register: TSnIDC = 0

→ Pattern switch order

Output pin	TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	Special pattern 1	Special pattern 2	Special pattern 3	INACT	INACT	Special pattern 0	Note 1	Note 2
TOSn2	INACT	INACT	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	Note 1	Note 2
TOSn3	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Special pattern 3	INACT	Note 1	Note 2
TOSn4	Special pattern 6	Special pattern 7	INACT	INACT	Special pattern 4	Special pattern 5	Note 1	Note 2
TOSn5	Special pattern 3	INACT	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Note 1	Note 2
TOSn6	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	INACT	INACT	Note 1	Note 2

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnARD signal or  
 TSnPSC = 0,  
 TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
 TSnOPT7 register: TSnIDC = 1

→ Pattern switch order

Output pin	TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Special pattern 3	Note 1	Note 2
TOSn2	Special pattern 5	Special pattern 6	Special pattern 7	INACT	INACT	Special pattern 4	Note 1	Note 2
TOSn3	Special pattern 2	Special pattern 3	INACT	INACT	Special pattern 0	Special pattern 1	Note 1	Note 2
TOSn4	INACT	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	INACT	Note 1	Note 2
TOSn5	Special pattern 0	Special pattern 1	Special pattern 2	Special pattern 3	INACT	INACT	Note 1	Note 2
TOSn6	Special pattern 7	INACT	INACT	Special pattern 4	Special pattern 5	Special pattern 6	Note 1	Note 2

- Note**
1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], output remains the level before the change.
  2. Output remains the level before the change.

- Caution** Special pattern 3 and special pattern 4 cannot be active simultaneously. Special pattern 0 and special pattern 7 cannot be active simultaneously either. The simultaneous active setting results in the following operations.
1. Only TOSn1, TOSn3, and TOSn5 are active during the simultaneous active period if TOSn1, TOSn3, and TOSn5 become active prior to TOSn2, TOSn4, and TOSn6.
  2. Only TOSn2, TOSn4, and TOSn6 are active during the simultaneous active period if TOSn2, TOSn4, and TOSn6 become active prior to TOSn1, TOSn3, and TOSn5.
  3. Only TOSn1, TOSn3, and TOSn5 are active during the simultaneous active period if TOSn1, TOSn3, TOSn5, and TOSn2, TOSn4, TOSn6 become active simultaneously.

- Note**
1. Special patterns 0 to 7: Special pattern output by TSnCCR0 to TSnCCR3, TSnPAT0, and TSnPAT1 registers
  2. INACT: Inactive level output

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnARD signal or  
 TSnPSC = 1,  
 TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
 TSnOPT7 register: TSnIDC = 0

← Pattern switch order

Output pin	TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	Special pattern 2	Special pattern 1	Special pattern 0	INACT	INACT	Special pattern 3	Note 1	Note 2
TOSn2	INACT	INACT	Special pattern 7	Special pattern 6	Special pattern 5	Special pattern 4	Note 1	Note 2
TOSn3	INACT	Special pattern 3	Special pattern 2	Special pattern 1	Special pattern 0	INACT	Note 1	Note 2
TOSn4	Special pattern 5	Special pattern 4	INACT	INACT	Special pattern 7	Special pattern 6	Note 1	Note 2
TOSn5	Special pattern 0	INACT	INACT	Special pattern 3	Special pattern 2	Special pattern 1	Note 1	Note 2
TOSn6	Special pattern 7	Special pattern 6	Special pattern 5	Special pattern 4	INACT	INACT	Note 1	Note 2



Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnARD signal or  
 TSnPSC = 1,  
 TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
 TSnOPT7 register: TSnIDC = 1

← Pattern switch order

Output pin	TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	Special pattern 3	Special pattern 2	Special pattern 1	Special pattern 0	Note 1	Note 2
TOSn2	Special pattern 6	Special pattern 5	Special pattern 4	INACT	INACT	Special pattern 7	Note 1	Note 2
TOSn3	Special pattern 1	Special pattern 0	INACT	INACT	Special pattern 3	Special pattern 2	Note 1	Note 2
TOSn4	INACT	Special pattern 7	Special pattern 6	Special pattern 5	Special pattern 4	INACT	Note 1	Note 2
TOSn5	Special pattern 3	Special pattern 2	Special pattern 1	Special pattern 0	INACT	INACT	Note 1	Note 2
TOSn6	Special pattern 4	INACT	INACT	Special pattern 7	Special pattern 6	Special pattern 5	Note 1	Note 2

- Note**
1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], output remains the level before the change.
  2. Output remains the level before the change.

- Caution** Special pattern 3 and special pattern 4 cannot be active simultaneously. Special pattern 0 and special pattern 7 cannot be active simultaneously either. The simultaneous active setting results in the following operations.
1. Only TOSn1, TOSn3, and TOSn5 are active during the simultaneous active period if TOSn1, TOSn3, and TOSn5 become active prior to TOSn2, TOSn4, and TOSn6.
  2. Only TOSn2, TOSn4, and TOSn6 are active during the simultaneous active period if TOSn2, TOSn4, and TOSn6 become active prior to TOSn1, TOSn3, and TOSn5.
  3. Only TOSn1, TOSn3, and TOSn5 are active during the simultaneous active period if TOSn1, TOSn3, TOSn5, and TOSn2, TOSn4, TOSn6 become active simultaneously.

- Note**
1. Special patterns 0 to 7: Special pattern output by TSnCCR0 to TSnCCR3, TSnPAT0, and TSnPAT1 registers
  2. INACT: Inactive level output

- When using 180° excitation function

TSnOPT4 register: TSnSOC = 0,

TSnOPT5 register: TSnADC = 1, TSnPOT = 1,

TSnOPT7 register: TSnIDC = 0

Output pin	TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	ACT	ACT	ACT	INACT	INACT	INACT	Note 1	Note 2
TOSn2	INACT	INACT	INACT	ACT	ACT	ACT	Note 1	Note 2
TOSn3	INACT	INACT	ACT	ACT	ACT	INACT	Note 1	Note 2
TOSn4	ACT	ACT	INACT	INACT	INACT	ACT	Note 1	Note 2
TOSn5	ACT	INACT	INACT	INACT	ACT	ACT	Note 1	Note 2
TOSn6	INACT	ACT	ACT	ACT	INACT	INACT	Note 1	Note 2

TSnOPT4 register: TSnSOC = 0,

TSnOPT5 register: TSnADC = 1, TSnPOT = 1,

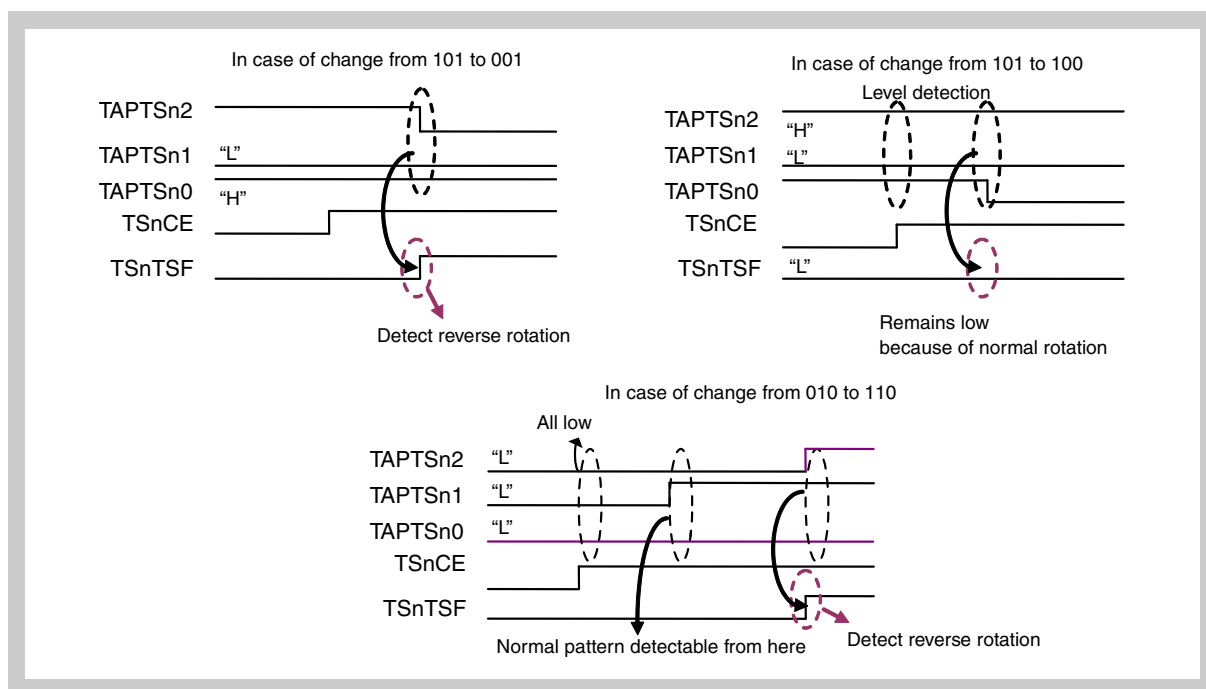
TSnOPT7 register: TSnIDC = 1

Output pin	TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	INACT	ACT	ACT	ACT	Note 1	Note 2
TOSn2	ACT	ACT	ACT	INACT	INACT	INACT	Note 1	Note 2
TOSn3	ACT	ACT	INACT	INACT	INACT	ACT	Note 1	Note 2
TOSn4	INACT	INACT	ACT	ACT	ACT	INACT	Note 1	Note 2
TOSn5	INACT	ACT	ACT	ACT	INACT	INACT	Note 1	Note 2
TOSn6	ACT	INACT	INACT	INACT	ACT	ACT	Note 1	Note 2

- Note**
1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], output remains the level before the change.
  2. Output remains the level before the change.
  3. ACT: Active level output
  4. INACT: Inactive level output

**(b) Input pattern change order detection**

Changes in the pattern (normal rotation/reverse rotation) input to the TAPTSn[2:0] pins are detected by the TSnTSF flag from the first change of the TAPTSn[2:0] pins from the time when TSnCE = 1 is set.



**Figure 17-2** Detection example of change in pattern (normal rotation/reverse rotation) Input to TAPTSn[2:0] Pins

**(12) Timer Sn option register (TSnOPT6)**

The TSnOPT6 register is a 16-bit register that controls each flag of timer Sn.

**Access** These registers can be read or written with a 16-bit, 8-bit, or 1-bit manipulation instruction.

TSnOPT6H enables access to the higher 8 bits of the TSnOPT6 register and TSnOPT6L enables access to the lower 8 bits of the TSnOPT6 register.

**Address** TS0OPT6: FFFFF5A4<sub>H</sub> TS0OPT6H: FFFFF5A5<sub>H</sub> TS0OPT6L: FFFFF5A4<sub>H</sub>  
TS1OPT6: FFFFF5E4<sub>H</sub> TS1OPT6H: FFFFF5E5<sub>H</sub> TS1OPT6L: FFFFF5E4<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. This registers is cleared by any reset, or when TSnCTL0.TSnCE is set to 0.

TSnOPT3	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	TSnPPF
	R	R	R	R	R	R	R	R/W
TSnOPT6H	7	6	5	4	3	2	1	0
	TSnPTF	TSnTDF	TSnNDF	TSnPRF	TSnPEF	TSnTBF	TSnSUF	TSnRSF
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
TSnOPT6L	7	6	5	4	3	2	1	0
	TSnPTF	TSnTDF	TSnNDF	TSnPRF	TSnPEF	TSnTBF	TSnSUF	TSnRSF
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Note** Refer to “Flags” on page 716 for detailed information on each flag function.

**Table 17-15 TSnOPT6 register contents (1/3)**

Bit position	Bit name	Function
8	TSnPPF	<p>This is a flag that compares input pattern (TAPTSn[2:0] pins) with output pattern (TOSn[6:1] pins) while the TSnOPT7.TSnPPC bit is set to 1, and detects the differences.</p> <p>0: Did not detect any phase difference between the TAPTSn[2:0] pins and TOSn[6:1] pins. 1: Detected phase difference between the TAPTSn[2:0] pins and TOSn[6:1] pins.</p> <p>The TSnPPF flag is set to 1 when the input/output pattern difference is detected, and the warning interrupt (INTTSnWn) occurs. This flag can be cleared by writing 0.</p> <p><b>Note:</b> 1. The TSnPPF flag is valid when the TSnOPT4.TSnSOC bit is set to 0, the TSnOPT5.TSnPOT bit is set to 1 or the TSnOPT7.TSnPPC bit is set to 1.</p> <p>2. The TSnPPF flag is valid in the 120° excitation mode and special 120° excitation mode.</p>

Table 17-15 TSnOPT6 register contents (2/3)

Bit position	Bit name	Function
7	TSnPTF	<p>This is a flag that detects abnormal toggle of the TAPTSn[2:0] pins when the value of the TSnOPT7.TSnPTC1 bit is 1.</p> <p>0: No detection of any abnormal toggle of the TAPTSn[2:0] pins. 1: Detected abnormal toggle of the TAPTSn[2:0] pins.</p> <p>The TSnPTF flag is set to 1 when the TAPTSn[2:0] pins (TSnAEDO signal toggle) are changed three times or more during the trigger of the TSnSTCI0 signal, or when the TAPTSn[2:0] pins (TSnAEDO signal toggles) are changed three times or more during the trigger of the TSnSTCI1 signal. Then a warning interrupt (INTTSnWN) occurs. The flag can be cleared by writing 0.</p> <p><b>Note:</b> 1. The TSnPTF flag is valid when the TSnOPT4.TSnSOC bit is set to 0, the TSnOPT5.TSnPOT bit is set to 1, or TSnOPT7.TSnPTC is set to 1.</p> <p>2. The TSnPTF flag is valid when the 180° excitation control function is used in the high-accuracy T-PWM mode, 120° excitation mode, or special 120° excitation mode.</p>
6	TSnTDF	<p>This is a flag that detects the simultaneous trigger generation of the TSnSTCI0 signal and the TSnSTCI1 signal when the value of the TSnOPT7.TSnTDC bit is 1.</p> <p>0: The triggers of the TSnSTCI0 signal and the TSnSTCI1 signal do not simultaneously occur 1: Detects the simultaneous occurrence of the TSnSTCI0 signal and TSnSTCI1 signal triggers.</p> <p>The TSnTDF is set to 1 when the simultaneously generated triggers of the TSnSTCI0 signal and the TSnSTCI1 signal are detected. Then the warning interrupt (INTTSnWN) is generated. The flag can be cleared by writing 0.</p> <p><b>Note:</b> 1. The TSnTDF flag is valid when the TSnOPT4.TSnSOC bit is set to 0, the TSnOPT5.TSnPOT bit is set to 1 or the TSnOPT7.TSnTDC bit is set to 1.</p> <p>2. The TSnTDF flag is valid when the 180° excitation control function is used in the high-accuracy T-PWM mode, 120° excitation mode, or special 120° excitation mode.</p>
5	TSnNDF	<p>This is a flag that detects noise generation at the TAPTSn[2:0] pins.</p> <p>0: No noise generated due to simultaneous change of 2 or more pins of the TAPTSn[2:0] pins. 1: Detected noise generation due to simultaneous change of 2 or more pins of the TAPTSn[2:0] pins.</p> <p>The TSnNDF flag is set to 1 when simultaneous value changes of 2 or more pins of the TAPTSn[2:0] pins are detected. Then the warning interrupt (INTTSnWN) occurs. The flag can be cleared by writing 0.</p> <p><b>Note:</b> The TSnNDF flag is valid when the 180° excitation control function is used in the high-accuracy T-PWM mode, 120° excitation mode, or special 120° excitation mode.</p>
4	TSnPRF	<p>This is a flag that detects the inversion of the input order of the TAPTSn[2:0] pins.</p> <p>0: The input order of TAPTSn[2:0] pins is not inverted. 1: The input order of TAPTSn[2:0] pins is inverted.</p> <p>The TSnPRF flag is set to 1 at the change timing of the TSnOPT5.TSnTSF bit and a warning interrupt (INTTSnWN) then occurs. The flag can be cleared by writing 0. Detection is possible from the timing of the second TAPTSn[2:0] pins change after the TSnCTL0.TSnCE bit is set to 1.</p> <p><b>Note:</b> The TSnPRF flag is valid when using the 180° excitation control function in the high-accuracy T-PWM mode, 120° excitation mode, or 120° excitation mode</p>

Table 17-15 TSnOPT6 register contents (3/3)

Bit position	Bit name	Function
3	TSnPEF	<p>This is a flag that detects [0, 0, 0] or [1, 1, 1] input to the TAPTSn[2:0] pins.  0: No input of [0, 0, 0] or [1, 1, 1] to the TAPTSn2 to TAPTSn0 pins.  1: [0, 0, 0] or [1, 1, 1] input to the TAPTSn2 to TAPTSn0 pins.</p> <p>The TSnPEF flag is set to 1 when [0, 0, 0] or [1, 1, 1] input to TAPTSn[2:0] pins is detected and a warning interrupt (INTTSnWN) then occurs. The flag can be cleared by writing 0.</p> <p><b>Note:</b> The TSnPEF flag is valid when the 180° excitation control function is used in the high-accuracy T-PWM mode, 120° excitation mode, or special 120° excitation mode.</p>
2	TSnTBF	<p>This is a flag that detects both of the positive phase and negative phase turns to be active simultaneously when any bit from the TSnIOC4.TSnTBA[2:0] bits is set to 1.  0: Positive phase and negative phase are not active simultaneously.  1: Positive phase and negative phase become active simultaneously.</p> <p>The TSnTBF flag is set to 1 when simultaneous active of any of positive phase (TOSn1, TOSn3, TOSn5) and any of negative phase (TOSn2, TOSn4, TOSn6) is detected. Then the error interrupt (INTTSnER) occurs. The flag can be cleared by writing 0.  Simultaneous active is not detected when TSnTBA[2:0] = [0, 0, 0].</p>
1	TSnSUF	<p>This is a flag that detects count up/down of 16-bit sub-counter.  0: Sub-counter performing count-up operation.  1: Sub-counter performing count-down operation.</p> <p>For the TSnSUF flag detection, count-up is the count from 0000H to (TSnCCR0 register value – 0002H), and count-down is the count from the TSnCCR0 register value to 0002H of sub-counter count operation.</p> <p><b>Note:</b> 1. TSnSUF flag is read only.  2. TSnSUF flag is valid only in the high-accuracy T-PWM mode.</p>
0	TSnRSF	<p>This is a flag that shows whether or not there is a reload request.  0: No reload request, or reload completed  Enable write access to the TSnCCR0 to TSnCCR5, TSnOPT1, TSnDTC0, TSnDTC1, TSnPAT0, and TSnPAT1 registers.  1: Reload request issued  (Disable write access to the TSnCCR0 to TSnCCR5, TSnOPT1, TSnDTC0, TSnDTC1, TSnPAT0, and TSnPAT1 registers)</p> <p>The TSnRSF flag indicates that the next transfer data is held in registers TSnCCR0 to TSnCCR5, TSnOPT1, TSnDTC0, TSnDTC1, TSnPAT0, and TSnPAT1.  The TSnRSF flag is set to 1 by writing to the TSnCCR1 register and cleared to 0 upon reload completion.</p> <hr/> <p><b>Caution:</b> The written value of the register may be undefined if writing to the TSnCCR0 to TSnCCR5, TSnOPT1, TSnDTC0, TSnDTC1, TSnPAT0, and TSnPAT1 registers and the reload timing conflict while TSnRSF = 1.</p> <hr/>

**(13) Timer Sn option register 7 (TSnOPT7)**

The TSnOPT7 register is an 8-bit register that detects the abnormal timer output and controls switching of the timer output.

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

**Address** TS0OPT7: FFFFF587<sub>H</sub>  
TS1OPT7: FFFFF5C7<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

7	6	5	4	3	2	1	0
0	0	TSnIDC	TSnPPC	TSnTDC	TSnPTC1	TSnPTC0	TSnTOS
R	R	R/W	R/W	R/W	R/W	R/W	R/W

**Table 17-16 TSnOPT7 register contents (1/2)**

Bit position	Bit name	Function																				
5	TSnIDC	Sets the output pattern of TOSn1 to TOSn6 by combination of the TSnOPT5.TSnTSF bit , the TSnOPT4.TSnPSC bit and the TSnARD signal. Refer to “Timer output order in each mode and function” on page 654 and “Timer output pattern in each mode and function” on page 671 for the output pattern.																				
		<table><tr><th>TSnOPT4.TSnSOC</th><th>TSnOPT5.TSnPOT</th><th>TSnOPT5.TSnPSS</th><th>Rotation Direction</th></tr><tr><td>1</td><td>-</td><td>-</td><td>-</td></tr><tr><td>0</td><td>0</td><td>-</td><td>TSnTSF flag</td></tr><tr><td>0</td><td>1</td><td>0</td><td>Input signal to TSnARD</td></tr><tr><td>0</td><td>1</td><td>1</td><td>TSnPSC bit</td></tr></table>	TSnOPT4.TSnSOC	TSnOPT5.TSnPOT	TSnOPT5.TSnPSS	Rotation Direction	1	-	-	-	0	0	-	TSnTSF flag	0	1	0	Input signal to TSnARD	0	1	1	TSnPSC bit
		TSnOPT4.TSnSOC	TSnOPT5.TSnPOT	TSnOPT5.TSnPSS	Rotation Direction																	
		1	-	-	-																	
		0	0	-	TSnTSF flag																	
		0	1	0	Input signal to TSnARD																	
		0	1	1	TSnPSC bit																	
<b>Caution:</b> The TSnIDC bit is valid in the 120° excitation mode, special 120° excitation mode, 180° excitation control, and software output control. 0 must be set in any other mode.																						
4	TSnPPC	Enables/disables input/output pattern difference detection (TSnOPT6.TSnPPF flag, warning interrupt (INTTSnWN)). 0: Disable input/output pattern difference detection 1: Enable input/output pattern difference detection																				
		<b>Note:</b> TSnPPC = 0 disables the TSnPPF bit detection and the warning interrupt (INTTSnWN). Also, input/output pattern difference detection flag (TSnOPT6.TSnPPF bit ) is not controlled.																				
3	TSnTDC	Controls the simultaneous trigger detection of the TSnSTCI0 and TSnSTCI1 signals (TSnOPT6.TSnTDF flag, warning interrupt (INTTSnWN)). 0: Disables simultaneous trigger detection of the TSnSTCI0 and TSnSTCI1 signals. 1: Enables simultaneous trigger detection of the TSnSTCI0 and TSnSTCI1 signals.																				
		<b>Note:</b> TSnTDC = 0 disables the warning interrupt (INTTSnWN). Also, simultaneous trigger detection flag (TSnOPT6.TSnTDF bit) is not controlled.																				

Table 17-16 TSnOPT7 register contents (2/2)

Bit position	Bit name	Function												
2, 1	TSnPTC1, TSnPTC0	Controls detection of abnormal toggle of the TAPTSn[2:0] pins (TSnOPT6.TSnPTF flag, warning interrupt (INTTSnWN)).												
		<table><tr><th>TSnPTC1</th><th>TSnPTC0</th><th>Operations</th></tr><tr><td>0</td><td>0/1</td><td><ul style="list-style-type: none"><li>Don't detect abnormal toggle of the TAPTSn[2:0] pins.</li></ul></td></tr><tr><td>1</td><td>0</td><td><ul style="list-style-type: none"><li>Detect abnormal toggle of the TAPTSn[2:0] pins.</li><li>Don't change timer output switch method by abnormal toggle detection (change of the TSnOPT5.TSnPOT bit from 1 to 0).</li></ul></td></tr><tr><td>1</td><td>1</td><td><ul style="list-style-type: none"><li>Detect abnormal toggle of the TAPTSn[2:0] pins.</li><li>Change the timer output switch method through abnormal toggle detection (change of the TSnOPT5.TSnPOT bit from 1 to 0).</li></ul></td></tr></table>	TSnPTC1	TSnPTC0	Operations	0	0/1	<ul style="list-style-type: none"><li>Don't detect abnormal toggle of the TAPTSn[2:0] pins.</li></ul>	1	0	<ul style="list-style-type: none"><li>Detect abnormal toggle of the TAPTSn[2:0] pins.</li><li>Don't change timer output switch method by abnormal toggle detection (change of the TSnOPT5.TSnPOT bit from 1 to 0).</li></ul>	1	1	<ul style="list-style-type: none"><li>Detect abnormal toggle of the TAPTSn[2:0] pins.</li><li>Change the timer output switch method through abnormal toggle detection (change of the TSnOPT5.TSnPOT bit from 1 to 0).</li></ul>
		TSnPTC1	TSnPTC0	Operations										
		0	0/1	<ul style="list-style-type: none"><li>Don't detect abnormal toggle of the TAPTSn[2:0] pins.</li></ul>										
		1	0	<ul style="list-style-type: none"><li>Detect abnormal toggle of the TAPTSn[2:0] pins.</li><li>Don't change timer output switch method by abnormal toggle detection (change of the TSnOPT5.TSnPOT bit from 1 to 0).</li></ul>										
		1	1	<ul style="list-style-type: none"><li>Detect abnormal toggle of the TAPTSn[2:0] pins.</li><li>Change the timer output switch method through abnormal toggle detection (change of the TSnOPT5.TSnPOT bit from 1 to 0).</li></ul>										
When the TSnPTC1 bit is set to 1, a warning interrupt (INTTSnWN) occurs when three or more toggles (abnormal toggles) are detected by the TAPTSn[2:0] pin inputs during the trigger of the TSnSTCI1 or TSnSTCI0 signal input. Then the TAPTSn[2:0] pin abnormal toggle detection flag (TSnPTF) is set to 1. The normal rotation or reverse rotation can be detected after the first change of the TAPTSn[2:0] pins after TSnCE = 1 is set.														
For details, refer to “Input pattern change order detection” on page 679.														
0	TSnTOS	<p>Controls the TOSn0 pin output switch.</p> <p>0: Outputs count up/down flag of 16-bit counter to TOSn0 pin.</p> <p>1: Outputs count up/down flag of 16-bit sub-counter to TOSn0 pin.</p> <p>When the TSnTOS bit is 0, the TSnOPT0 register is output to the TOSn0 pin. When the TSnTOS bit is 1, the TSnOPT6.TSnSUF bit status is output to the TOSn0 pin.</p> <p><b>Note:</b> The TSnTOS bit is valid only in the high-accuracy T-PWM mode.</p>												



**(14) Timer Sn pattern register 0 (TSnPAT0)**

The TSnPAT0 register is a 16-bit register that controls output patterns in the special 120° excitation mode and special pattern output mode..

**Access** This register can be read/written in 16-bit units only. Do not read/write this register in 8-bit or 1-bit units.

**Address** TS0PAT0: FFFFF596<sub>H</sub>  
TS1PAT0: FFFFF5D6<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. This register is cleared by any reset.

15	14	13	12	11	10	9	8
Special pattern 3				Special pattern 2			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
Special pattern 1				Special pattern 0			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Caution** The TSnPAT0 register setting must be executed while the timer Sn operation is stopped (when the TSnCE bit of the TSnCTL0 register is set to 0). In order to change the value of the TSnPAT0 register during the timer Sn operation (when the value of the TSnCE bit of the TSnCTL0 register is 1), do so after setting the TSnCE bit to 1 and waiting for 1 cycle of the 16-bit counter to be completed.

**Table 17-17 TSnPAT0 register contents**

Bit position	Bit name	Function
15 to 12	-	Controls the output pattern of special pattern 3 of the TOSn1, TOSn3, and TOSn5 pins in the special 120° excitation mode. The output pattern of the TOSn3 pin is controlled in the special pattern output mode. <b>Note:</b> Refer to <i>Table 17-18 on page 686</i> for the output pattern.
11 to 8	-	Controls the output pattern of special pattern 2 of the TOSn1, TOSn3, and TOSn5 pins in the special 120° excitation mode. The output pattern of the TOSn2 pin is controlled in the special pattern output mode. <b>Note:</b> Refer to <i>Table 17-18 on page 686</i> for the output pattern.
7 to 4	-	Controls the output pattern of special pattern 1 of the TOSn1, TOSn3, and TOSn5 pins in the special 120° excitation mode. The output pattern of the TOSn1 pin is controlled in the special pattern output mode. <b>Note:</b> Refer to <i>Table 17-18 on page 686</i> for the output pattern.
3 to 0	-	Controls output pattern of special pattern 0 of the TOSn1, TOSn3, and TOSn5 pins in the special 120° excitation mode. The output pattern of the TOSn0 pin is controlled in the special pattern output mode. <b>Note:</b> Refer to <i>Table 17-18 on page 686</i> for the output pattern.

**Table 17-18 Output Pattern Setting for TSnPAT0 Register**

TSnPAT0 register bit setting: bits 15 to 12 (special pattern 3), 11 to 8 (special pattern 2), 7 to 4 (special pattern 1), 3 to 0 (special pattern 0)	Output level by match between TSnCCRm register and 16-bit counter			
	Match with TSnCCR3 register	Match with TSnCCR2 register	Match with TSnCCR1 register	Match with TSnCCR0 register
0, 0, 0, 0	INACT <sup>Note 1</sup>	INACT <sup>Note 1</sup>	INACT <sup>Note 1</sup>	INACT <sup>Note 1</sup>
0, 0, 0, 1	INACT	INACT	INACT	ACT
0, 0, 1, 0	INACT	INACT	ACT	INACT
0, 0, 1, 1	INACT	INACT	ACT	ACT
0, 1, 0, 0	INACT	ACT	INACT	INACT
0, 1, 0, 1	INACT	ACT	INACT	ACT
0, 1, 1, 0	INACT	ACT	ACT	INACT
0, 1, 1, 1	INACT	ACT	ACT	ACT
1, 0, 0, 0	ACT	INACT	INACT	INACT
1, 0, 0, 1	ACT	INACT	INACT	ACT
1, 0, 1, 0	ACT	INACT	ACT	INACT
1, 0, 1, 1	ACT	INACT	ACT	ACT
1, 1, 0, 0	ACT	ACT	INACT	INACT
1, 1, 0, 1	ACT	ACT	INACT	ACT
1, 1, 1, 0	ACT	ACT	ACT	INACT
1, 1, 1, 1	ACT <sup>Note 2</sup>	ACT <sup>Note 2</sup>	ACT <sup>Note 2</sup>	ACT <sup>Note 2</sup>

- Note**
1. Inactive level is output regardless of match with the TSnCCRm register.
  2. Active level is output regardless of match with the TSnCCRm register.
  3. ACT: Active level output
  4. INACT: Inactive level output
  5. m = 1 to 3

**(15) Timer Sn pattern register 1 (TSnPAT1)**

The TSnPAT1 register is a 16-bit register that controls output patterns in the special 120° excitation mode and special pattern output mode.

**Access** This register can be read/written in 16-bit units only. Do not read/write this register in 8-bit or 1-bit units.

**Address** TS0PAT1: FFFFF594<sub>H</sub>  
TS1PAT1: FFFFF5D4<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. This register is cleared by any reset.

15	14	13	12	11	10	9	8
Special pattern 7				Special pattern 6			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
Special pattern 5				Special pattern 4			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Caution** The TSnPAT1 register setting must be executed while the timer Sn operation is stopped (when the TSnCE bit of the TSnCTL0 register is set to 0). In order to change the value of the TSnPAT0 register during the timer Sn operation (when the value of the TSnCE bit of the TSnCTL0 register is 1), do so after setting the TSnCE bit to 1 and waiting for 1 cycle of the 16-bit counter to be completed.

**Table 17-19 TSnPAT1 register contents**

Bit position	Bit name	Function
15 to 12	-	Controls the output pattern of special pattern 7 of the TOSn2, TOSn4, and TOSn6 pins in the special 120° excitation mode. Output pattern of the TOSn7 pin is controlled in the special pattern output mode. <b>Note:</b> Refer to <i>Table on page 688</i> for the output pattern.
11 to 8	-	Controls the output pattern of special pattern 6 of the TOSn2, TOSn4, and TOSn6 pins in the special 120° excitation mode. The output pattern of the TOSn6 pin is controlled in the special pattern output mode. <b>Note:</b> Refer to <i>Table on page 688</i> for the output pattern.
7 to 4	-	Controls the output pattern of special pattern 5 of the TOSn2, TOSn4, and TOSn6 pins in the special 120° excitation mode. The output pattern of the TOSn5 pin is controlled in the special pattern output mode. <b>Note:</b> Refer to <i>Table on page 688</i> for the output pattern.
3 to 0	-	Controls the output pattern of special pattern 4 of the TOSn2, TOSn4, and TOSn6 pins in the special 120° excitation mode. The output pattern of TOSn4 pin is controlled in the special pattern output mode. <b>Note:</b> Refer to <i>Table on page 688</i> for the output pattern.

**Table 17-20 Output pattern setting for TSnPAT1 register**

TSnPAT1 register bit setting: bits 15 to 12 (special pattern 7), 11 to 8 (special pattern 6), 7 to 4 (special pattern 5), 3 to 0 (special pattern 4)	Output level based on match between the tsnccrm register and 16-bit counter			
	Match with TSnCCR3 register	Match with TSnCCR2 register	Match with TSnCCR1 register	Match with TSnCCR0 register
0, 0, 0, 0	INACT <sup>Note 1</sup>	INACT <sup>Note 1</sup>	INACT <sup>Note 1</sup>	INACT <sup>Note 1</sup>
0, 0, 0, 1	INACT	INACT	INACT	ACT
0, 0, 1, 0	INACT	INACT	ACT	INACT
0, 0, 1, 1	INACT	INACT	ACT	ACT
0, 1, 0, 0	INACT	ACT	INACT	INACT
0, 1, 0, 1	INACT	ACT	INACT	ACT
0, 1, 1, 0	INACT	ACT	ACT	INACT
0, 1, 1, 1	INACT	ACT	ACT	ACT
1, 0, 0, 0	ACT	INACT	INACT	INACT
1, 0, 0, 1	ACT	INACT	INACT	ACT
1, 0, 1, 0	ACT	INACT	ACT	INACT
1, 0, 1, 1	ACT	INACT	ACT	ACT
1, 1, 0, 0	ACT	ACT	INACT	INACT
1, 1, 0, 1	ACT	ACT	INACT	ACT
1, 1, 1, 0	ACT	ACT	ACT	INACT
1, 1, 1, 1	ACT <sup>Note 2</sup>	ACT <sup>Note 2</sup>	ACT <sup>Note 2</sup>	ACT <sup>Note 2</sup>

- Note**
1. Inactive level is output regardless of match with the TSnCCRM register.
  2. Active level is output regardless of match with the TSnCCRM register.
  3. ACT: Active level output
  4. INACT: Inactive level output
  5. m = 1 to 3

## 17.4 Basic Operation

### 17.4.1 16-bit counter basic operation

The 16-bit counter basic operation is described below. Refer to “*Operation in Each Mode*” on page 756 for details.

#### (1) Count start operation

The 16-bit counter of timer Sn starts counting from the initial value FFFFH<sup>Note</sup> in every mode except the high-accuracy T-PWM mode.

Count-up operation is performed such as FFFFH, 0000H, 0001H, 0002H, 0003H ...

For the high-accuracy T-PWM mode, refer to “*16-bit counter operation in high-accuracy T-PWM mode*” on page 801.

**Note** FFFE<sub>H</sub> in high-accuracy T-PWM mode.

#### (2) Clear operation

The 16-bit counter is cleared to 0000H upon a match between the values of the 16-bit counter and compare register. Count operation from FFFFH to 0000H is not detected as clear operation immediately after count start or when overflow.

#### (3) Overflow operation

Overflow of the 16-bit counter occurs at the timing of the switch from FFFFH to 0000H in the free-running mode. If overflow occurs, the TSnOVF bit of the TSnOPT0 register is set to 1 and an overflow interrupt (INTTSnOV) occurs. The overflow interrupt (INTTSnOV) does not occur immediately after count operation start or when the compare value is cleared by detecting match at FFFFH.

#### (4) Counter read operation during count operation

Timer Sn can read the 16-bit counter value with the TSnCNT register during count operation.

**(5) Interrupt operation**

The following interrupt signals are generated in timer Sn.

- INTTSnCC0: Functions as a TSnCCRn0 buffer register match interrupt.
- INTTSnCC1: Functions as a TSnCCRn1 buffer register match interrupt.
- INTTSnCC2: Functions as a TSnCCRn2 buffer register match interrupt.
- INTTSnCC3: Functions as a TSnCCRn3 buffer register match interrupt.
- INTTSnCC4: Functions as a TSnCCRn4 buffer register match interrupt.
- INTTSnCC5: Functions as a TSnCCRn5 buffer register match interrupt.
- INTTSnOD0: Functions as a peak interrupt at the S/W timing from count-up operation to count-down operation of the 16-bit counter.
- INTTSnOD: Functions as a trough interrupt at the S/W timing from count-down operation to count-up operation of the 16-bit counter.
- INTTSnOV: Functions as an overflow interrupt.
- INTTSnER: Functions as a positive/negative phase simultaneous active detection interrupt.
- INTTSnWN: Functions as each type warning detection interrupt.

### 17.4.2 Compare register rewriting

The reload function is valid in the PWM mode, triangular-wave PWM mode, high-accuracy T-PWM mode (when the TSnCMS bit is 0), PWM mode with dead time, external trigger pulse output mode, 120° excitation mode, special 120° excitation mode, and special pattern output mode. Reload setting in any other mode than above is invalid.

The compare registers and control registers with reload function are as follows.

- TSnCCR0 to TSnCCR5
- TSnOPT1
- TSnPAT0, TSnPAT1
- TSnDTC0, TSnDTC1

There are the following two modes for rewriting compare registers with the reload function.

- Anytime rewrite mode  
In the anytime rewrite mode, writing to each compare register during update operation by each independent compare register enables write access value updating anytime.
- Reload mode (batch rewrite function)  
Reload is enabled by writing to the TSnCCR1 register (reload request flag TSnRSF is set) and then the values of the target registers are updated simultaneously at the next reload timing. The reload timing is the peak/trough timing of a 16-bit counter. Control is executed by the TSnRTE and TSnRBE bits of the TSnOPT1 register, respectively. Reload is not enabled even if writing to any other register than TSnCCR1.

Rewriting of compare registers is possible with DMA transfer.

The DMA transfer can start at each register. The address is incremented after each DMA transfer and always stops after the DMA transfer to the TSnCCR1 register.

Table 17-21 shows the order of the DMA transfers.

**Table 17-21 Timer S DMA transfer (1/2)**

Timer	Address	Register name	DMA transfer order	
TMS0	FFFFFB40H	TS0OPT1A		Start registers
	FFFFFB42H	TS0DTC1A		
	FFFFFB44H	TS0DTC0A		
	FFFFFB46H	TS0CCR5A		
	FFFFFB48H	TS0CCR4A		
	FFFFFB4AH	TS0CCR3B		
	FFFFFB4CH/ FFFFF594H <sup>a</sup>	TS0CCR2B/ TS0PAT1 <sup>a</sup>		
	FFFFFB4EH/ FFFFF596 <sup>a</sup>	TS0CCR1B/ TS0PAT0 <sup>a</sup>		
	FFFFFB50H	TS0CCR0A		
	FFFFFB52H	TS0CCR3A		
	FFFFFB54H	TS0CCR2A		
	FFFFFB56H	TS0CCR1A	↓	End register

**Table 17-21 Timer S DMA transfer (2/2)**

Timer	Address	Register name	DMA transfer order
TMS1	FFFFFBC0H	TS1OPT1A	
	FFFFFBC2H	TS1DTC1A	
	FFFFFBC4H	TS1DTC0A	
	FFFFFBC6H	TS1CCR5A	
	FFFFFBC8H	TS1CCR4A	
	FFFFFBCAH	TS1CCR3B	
	FFFFFBCCH/ FFFFF5D4 <sup>a</sup>	TS1CCR2B/ TS1PAT1 <sup>a</sup>	
	FFFFFBCEH/ FFFFF5D6 <sup>a</sup>	TS1CCR1B/ TS1PAT0 <sup>a</sup>	
	FFFFFBD0H	TS1CCR0A	
	FFFFFBD2H	TS1CCR3A	
	FFFFFBD4H	TS1CCR2A	
	FFFFFBD6H	TS1CCR1A	↓
			End register

a) If Timer Sn is operated in high-accuracy T-PWM mode, TSnCCR[2:1]B is used, otherwise TSnPAT[1:0]

Refer to “Interrupt Thinning out Function” on page 735 for details on the interrupt thinning out function through the TSnOPT1 register setting.

**Table 17-22 Rewrite timing by mode**

Mode	Rewrite timing
Interval timer mode	Anytime rewrite
External event count mode	Anytime rewrite
External trigger pulse output mode	Reload <sup>Note 3</sup>
One-shot pulse mode	Anytime rewrite
PWM mode	Reload
Free-running mode	Anytime rewrite
Triangular-wave PWM mode	Reload <sup>Note 1</sup>
High-accuracy T-PWM mode	Anytime rewrite/Reload <sup>Note 2</sup>
PWM mode with dead time	Reload
120° excitation mode	Reload
Special 120° excitation mode	Reload
Special pattern output mode	Reload

- Note**
1. Rewriting only in trough interrupt
  2. Setting with the TSnCMS bit of the TSnOPT0 register
  3. Reload thinning out function disabled

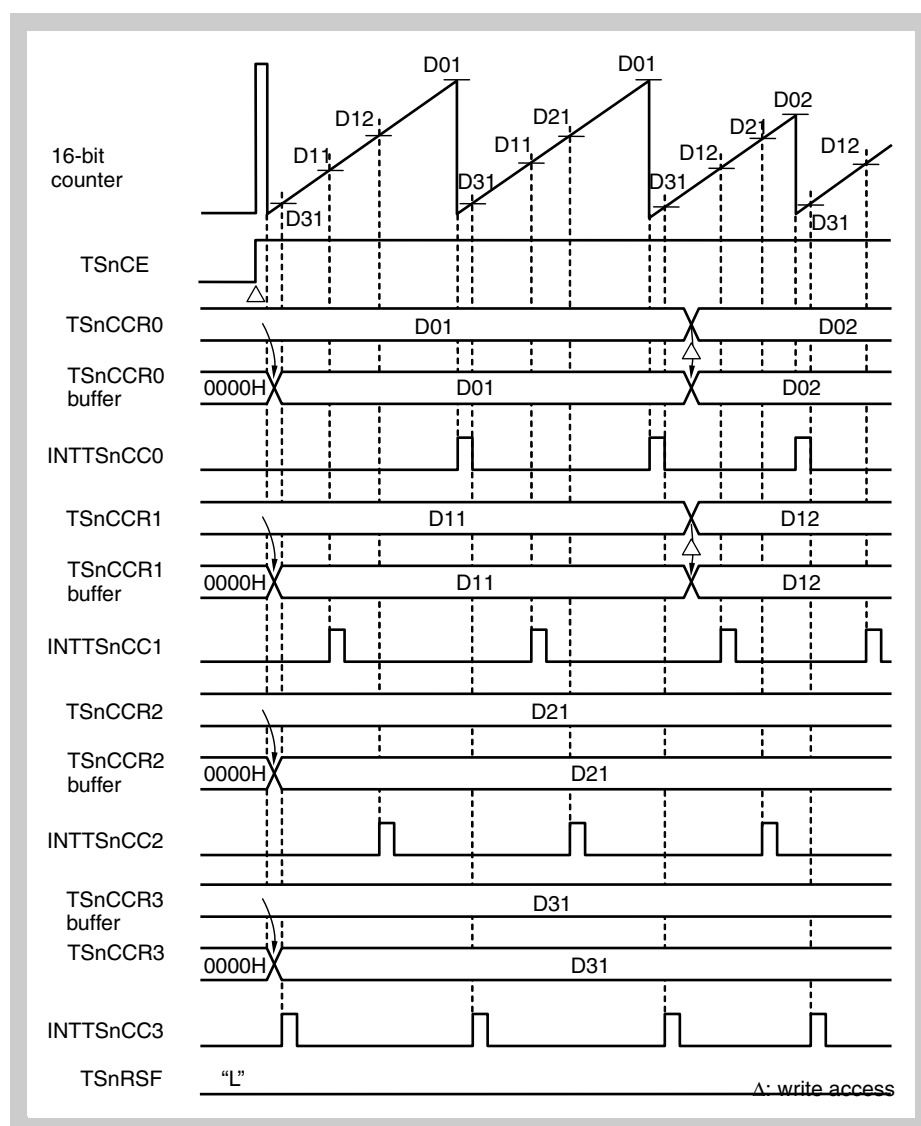


**(1) Anytime rewrite mode**

In the anytime rewrite mode, the value written to each compare register is immediately transferred to the internal buffer register as the comparison target for the counter value.

The value is written to the compare registers (such as TSnCCR0 to TSnCCR5) and then transferred to the internal buffer register in 4 clocks ( $f_{TMSn}$ ). However, only for the TSnCCR1 register, transfer timing is in 5 clocks ( $f_{TMSn}$ ) because of its 3-step buffer composition.

**Note** In the high-accuracy T-PWM mode, the mode is the anytime rewrite mode when the TSnCMS bit of the TSnOPT0 register is set to 1. The settings of the TSnIOE, TSnICE, TSnRDE, and TSnID4 to TSnID0 bits of the TSnOPT1 register are ignored.

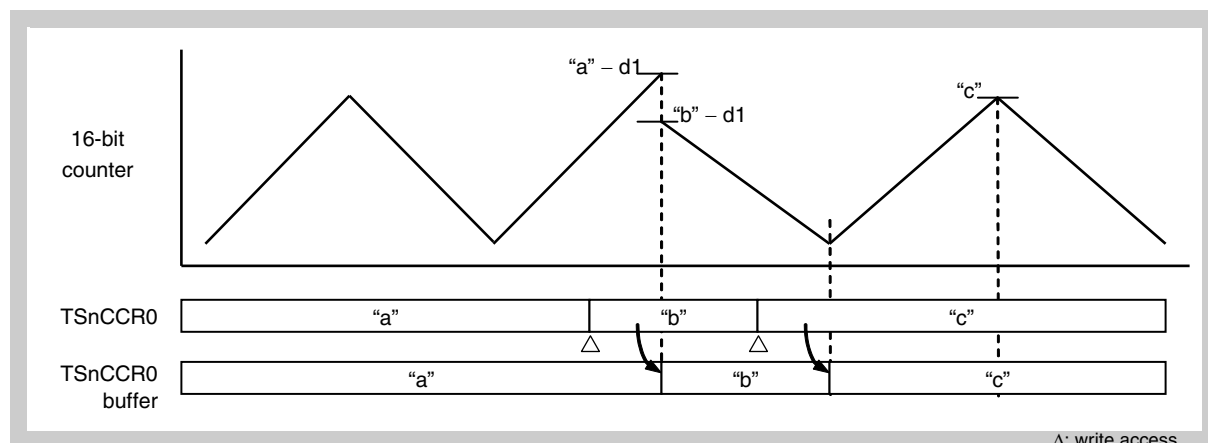


**Figure 17-3** Anytime rewrite timing (example in PWM mode)

- Note**
1. D<sub>01</sub>, D<sub>02</sub>: TSnCCR0 register setting values (0000H to FFFFH)  
D<sub>11</sub>, D<sub>12</sub>: TSnCCR1 register setting values (0000H to FFFFH)  
D<sub>21</sub>: TSnCCR2 register setting value (0000H to FFFFH)  
D<sub>31</sub>: TSnCCR3 register setting value (0000H to FFFFH)
  2. Timing chart where interval timer mode is provided as an example
  3. Δ: CPU write access

### (a) Caution on TSnCCR0 register rewriting in high-accuracy T-PWM mode

In the high-accuracy T-PWM mode, transfer to the TSnCCR0 buffer register is not executed if the TSnCCR0 register is rewritten in the anytime rewrite mode. The timing is shown below.

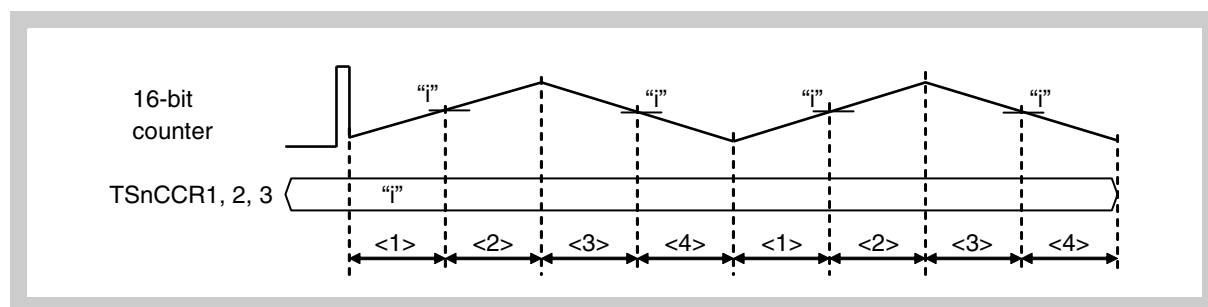


**Figure 17-4** TSnCCR0 register rewrite timing in anytime rewrite mode (high-accuracy T-PWM mode)

**Note** d1: TSnDTC1 setting value

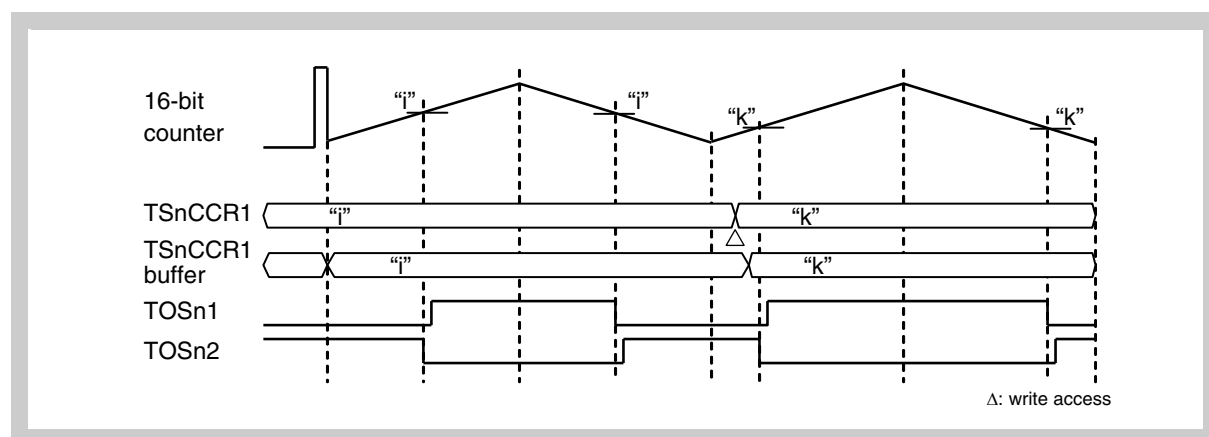
After rewriting of the TSnCCR0 register, the TSnCCR0 register value is transferred to the TSnCCR0 buffer register at the next peak/trough timing. The settings of the TSnIOE, TSnICE, TSnRDE, and TSnID4 to TSnID0 bits of the TSnOPT1 register are ignored because TSnCMS of TSnOPT0 is 1 (anytime rewrite mode).

### (b) Caution on rewriting in high-accuracy T-PWM mode for TSnCCR1 to TSnCCR3 registers

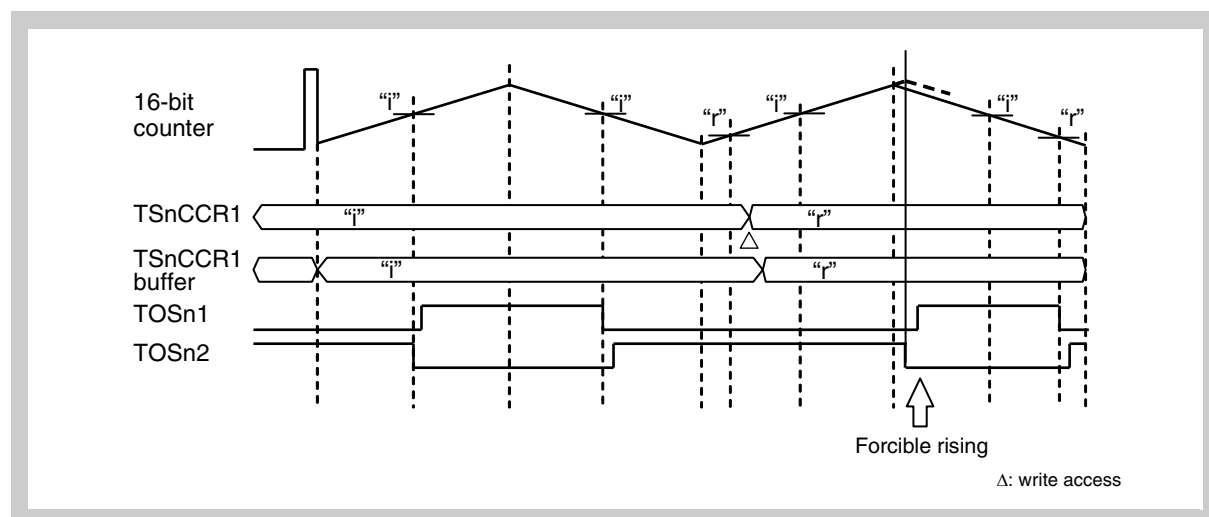


Rewriting during period <1> (rewriting before match detection)

When rewriting before match of the 16-bit counter value with the TSnCCR1 to TSnCCR3 values, the rewritten value is reflected immediately by detection of the match with the 16-bit counter value after rewriting.

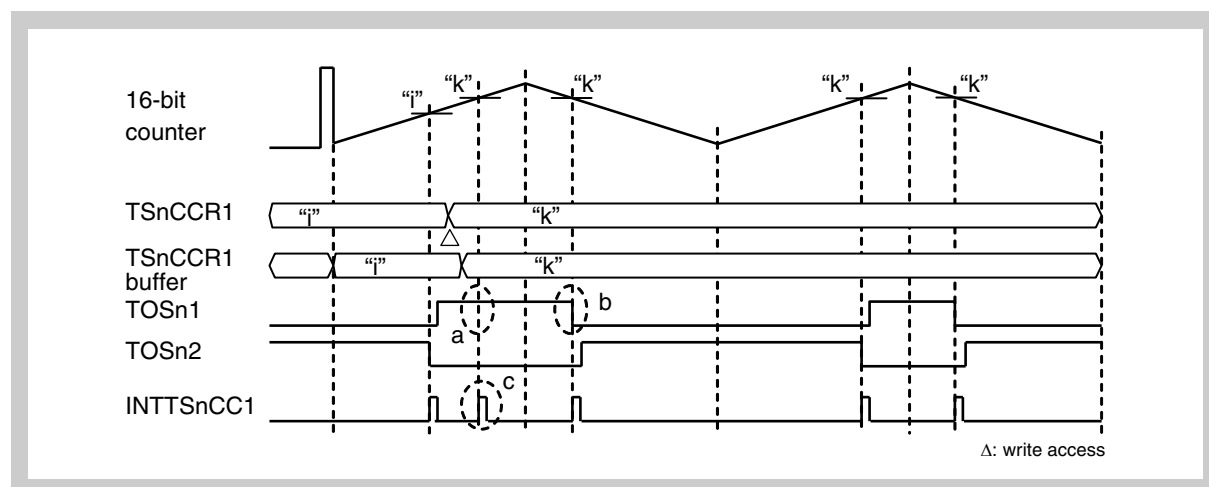


If a value smaller than the 16-bit counter value is rewritten before a match, the match is not detected, so that the output waveform is as follows. TOSn1, TOSn3, and TOSn5 forcibly change to active and TOSn2, TOSn4, and TOSn6 to inactive at the peak even without detection.



### Rewriting at period <2> (rewriting after match detection)

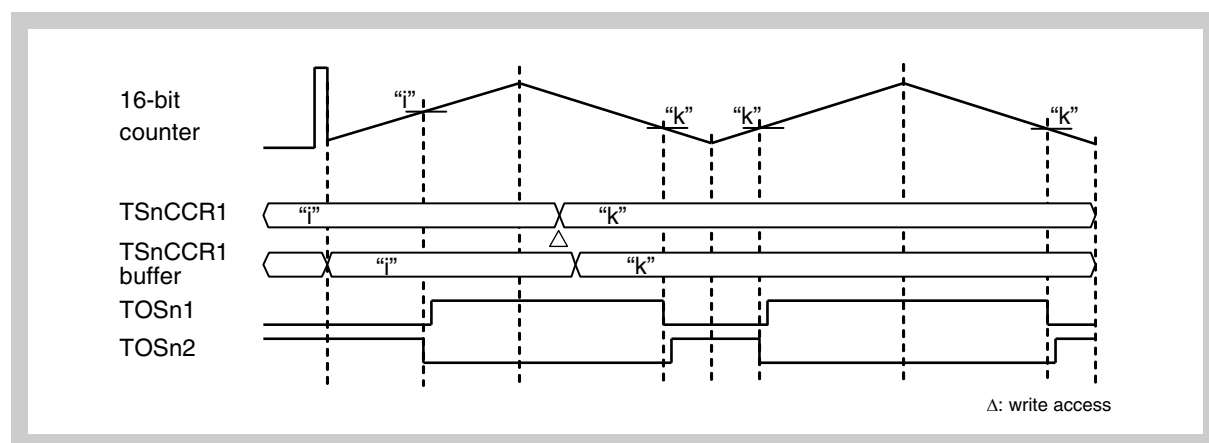
The timer output level does not change upon re-match after match detection if rewritten after match of the 16-bit counter value with the TSnCCR1 to TSnCCR3 register values.



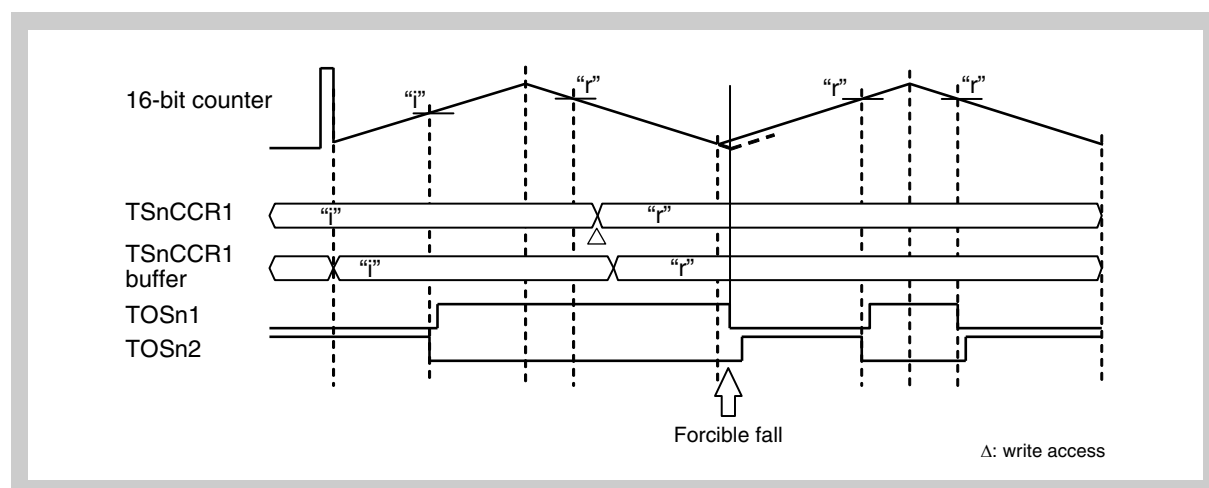
- Note**
1. The timer output level does not change upon rematch through rewriting after match detection.
  2. The timer output level changes from the first match after up/down switch.
  3. However, the interrupt output of INTTSnCC1 is match interrupt output.

### Rewriting at period <3> (rewriting before match detection)

When rewriting before match of the 16-bit counter value with the TSnCCR1 to TSnCCR3 values, the rewritten value is reflected immediately by detection of the match with the 16-bit counter value after rewriting.

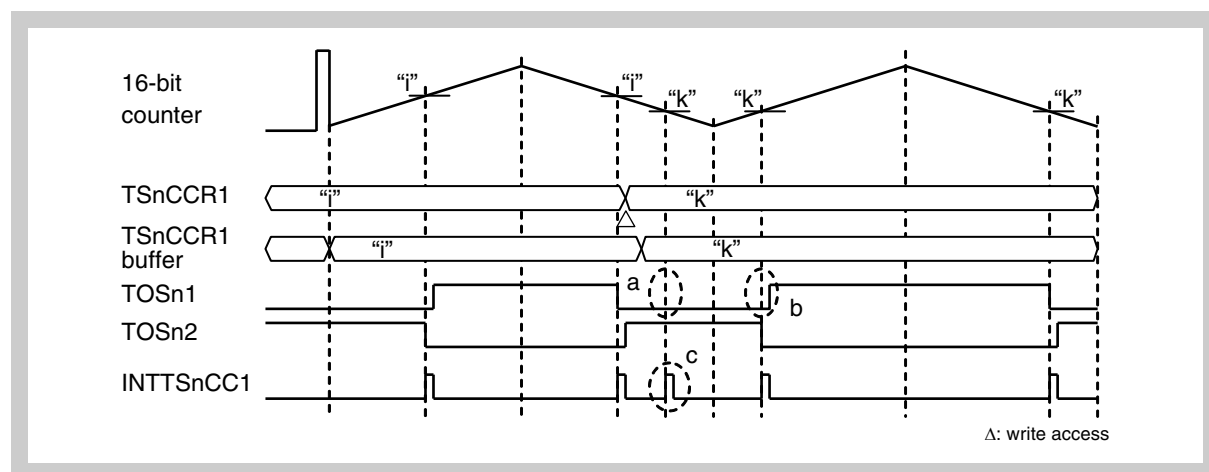


If a value larger than the 16-bit counter value is rewritten before match, match is not detected, so that the output waveform is as shown below. TOSn1, TOSn3, and TOSn5 forcibly change to inactive and TOSn2, TOSn4, and TOSn6 to active at the trough even without match detection.



#### Rewriting at period <4> (rewriting after match detection)

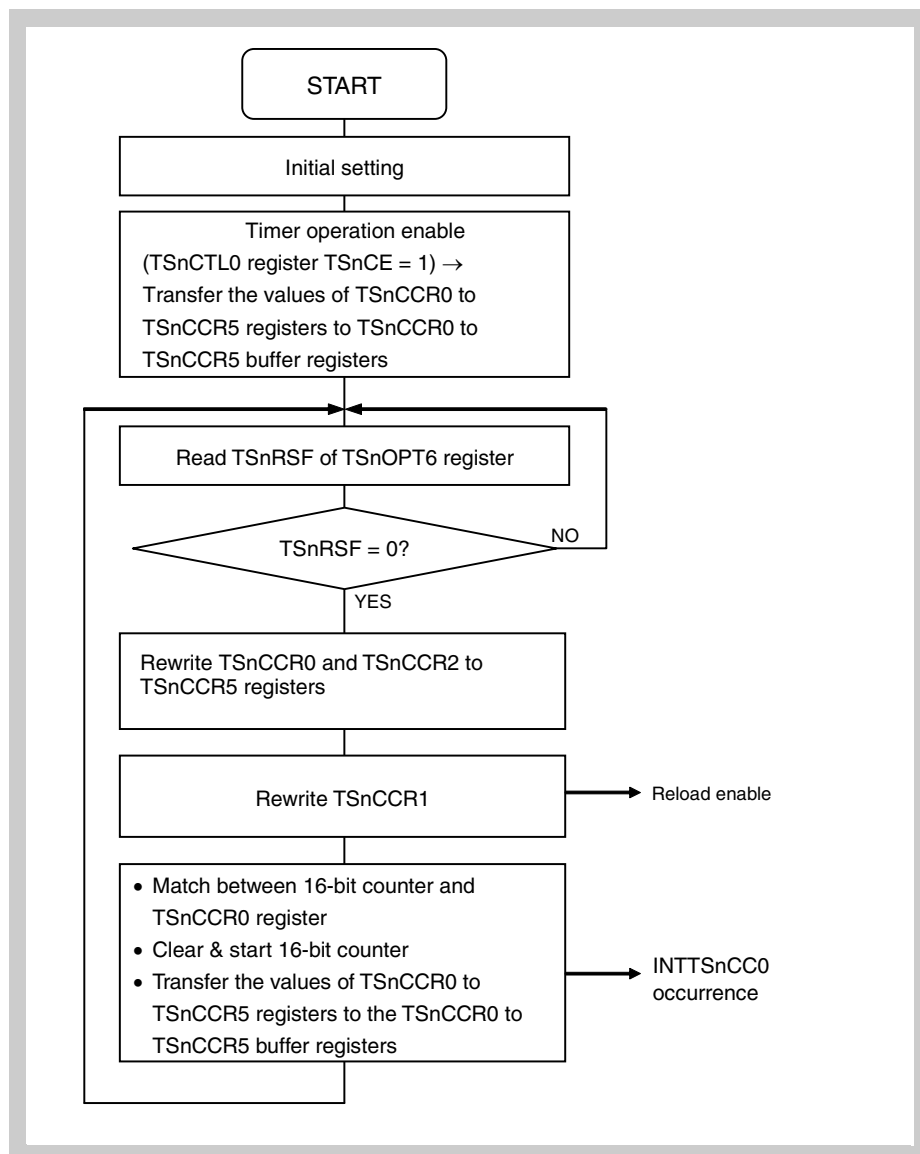
The timer output level does not change upon re-match after match detection if rewritten after match of the 16-bit counter value with the values of the TSnCCR1 to TSnCCR3 registers.



- Note**
1. The timer output level does not change even upon re-match through rewriting after match detection.
  2. The timer output level changes from the first match after up/down switch.
  3. However, the interrupt output of INTTSnCC1 is match interrupt output.

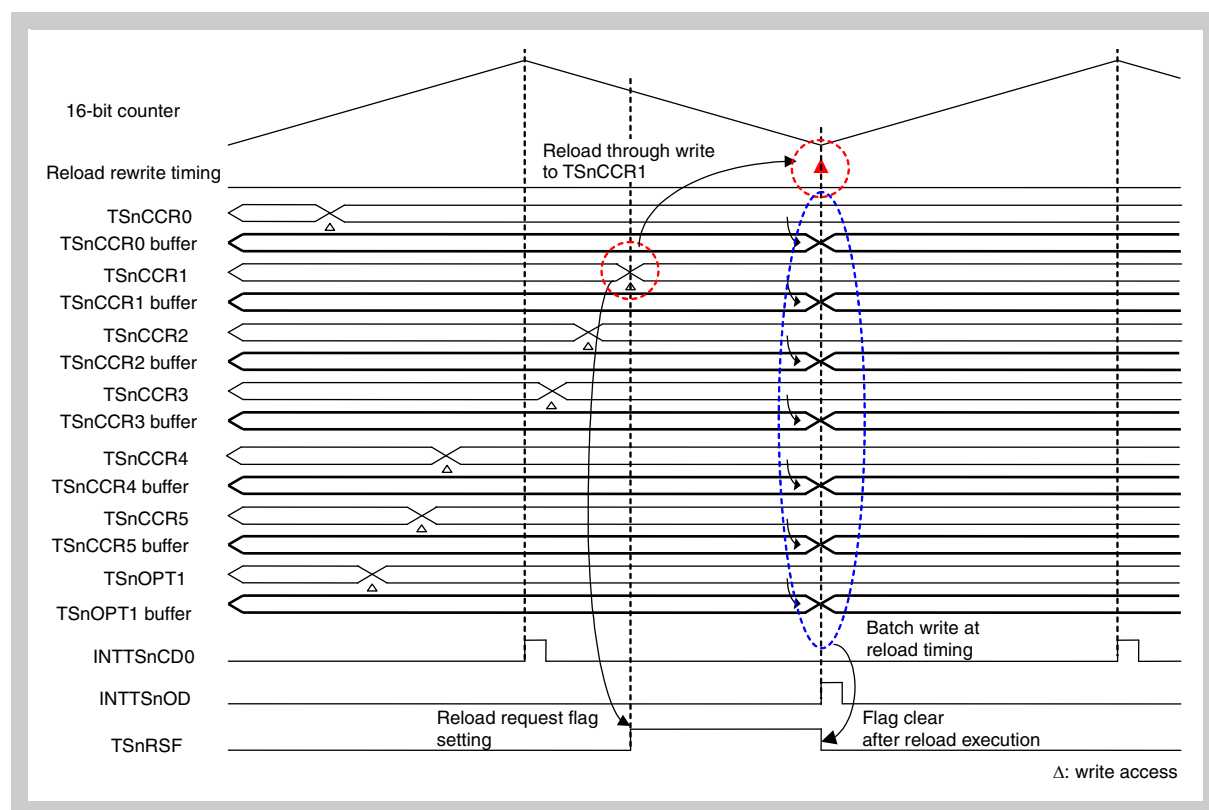
**(2) Reload mode (batch rewrite function)**

In the reload mode, the values written to the reload target registers (TSnCCR0 to TSnCCR5, TSnOPT1, TSnDTC0, TSnDTC1, TSnPAT0, and TSnPAT1) are transferred to each buffer register simultaneously at the reload timing. Reload target register value must be rewritten while the reload request flag (TSnRSF of the TSnOPT6 register) is 0.

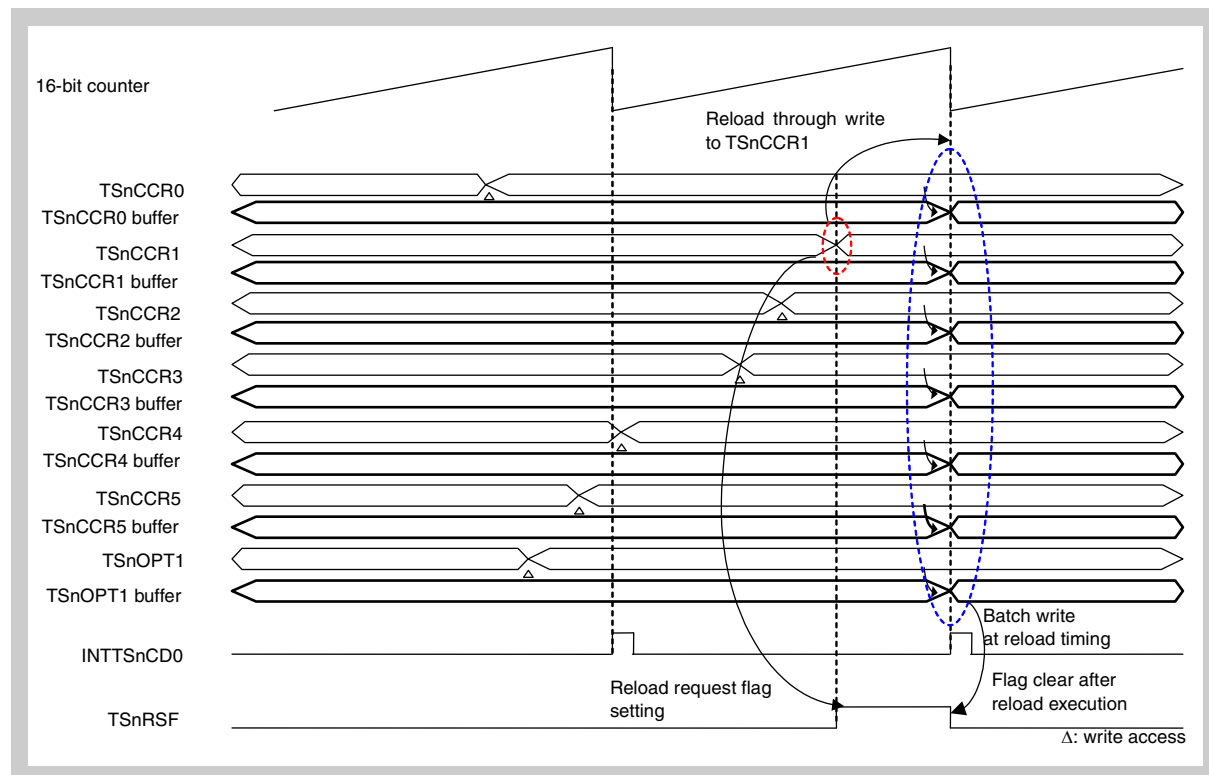


**Figure 17-5 Basic operation in reload mode (batch rewrite function) <example in PWM mode>**

**Caution** The write access to the TSnCCR1 register includes the operation that enables reload. Therefore, do not rewrite any TSnCCR1 register value before rewriting the other TSnCCR registers.



**Figure 17-6** Batch rewrite timing (1/2)  
(high-accuracy T-PWM mode)



**Figure 17-7** Batch rewrite timing (2/2)  
(PWM mode)

**(a) Example of reload rewriting setting in each mode**

The following is a setting example of the reload enable mode. Performing the settings listed in *Table 17-23 on page 700* results in reload. Reload does not occur in any other setting than this combination. Refer to *Table 17-10 on page 640* for details.

**Table 17-23 Reload rewriting setting by mode**

Setting bit mode	TSnCMS of TSnOPT0 register	TSnRDE of TSnOPT1 register	TSnICE of TSnOPT1 register	TS0nRTE of TSnOPT1 register	TSnIOE of TSnOPT1 register	TSnRBE of TSnOPT1 register
External trigger pulse output	0	0	×	1	×	×
PWM mode	0	0	×	1	×	×
	0	1	1	1	×	×
Triangular-wave PWM	0	0	×	×	×	1
	0	1	×	×	1	1
High-accuracy T-PWM	0	0	×	×	×	×
	0	1	×	×	1	1
	0	1	1	1	×	×
	0	1	1	1	1	1
PWM with dead time	0	0	×	1	×	×
	0	1	1	1	×	×
120° excitation	0	0	×	1	×	×
	0	1	1	1	×	×
Special 120° excitation	0	0	×	1	×	×
	0	1	1	1	×	×
Special pattern output	0	0	×	1	×	×
	0	1	1	1	×	×

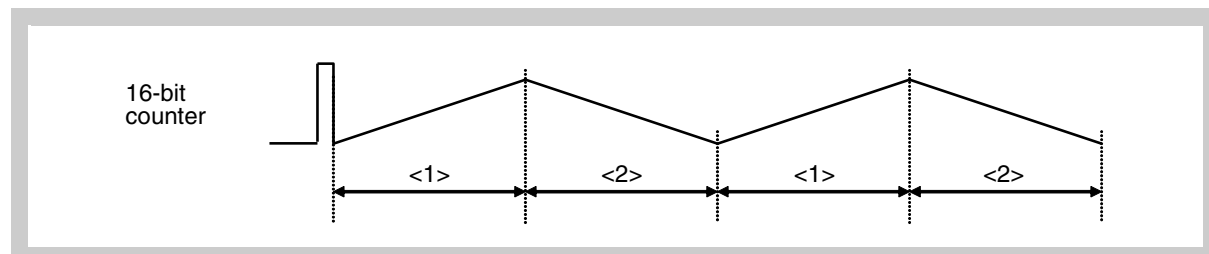
**Note** 1. Except for TS0nRTE = 0 and TSnRBE = 0

2. × indicates 1 or 0.



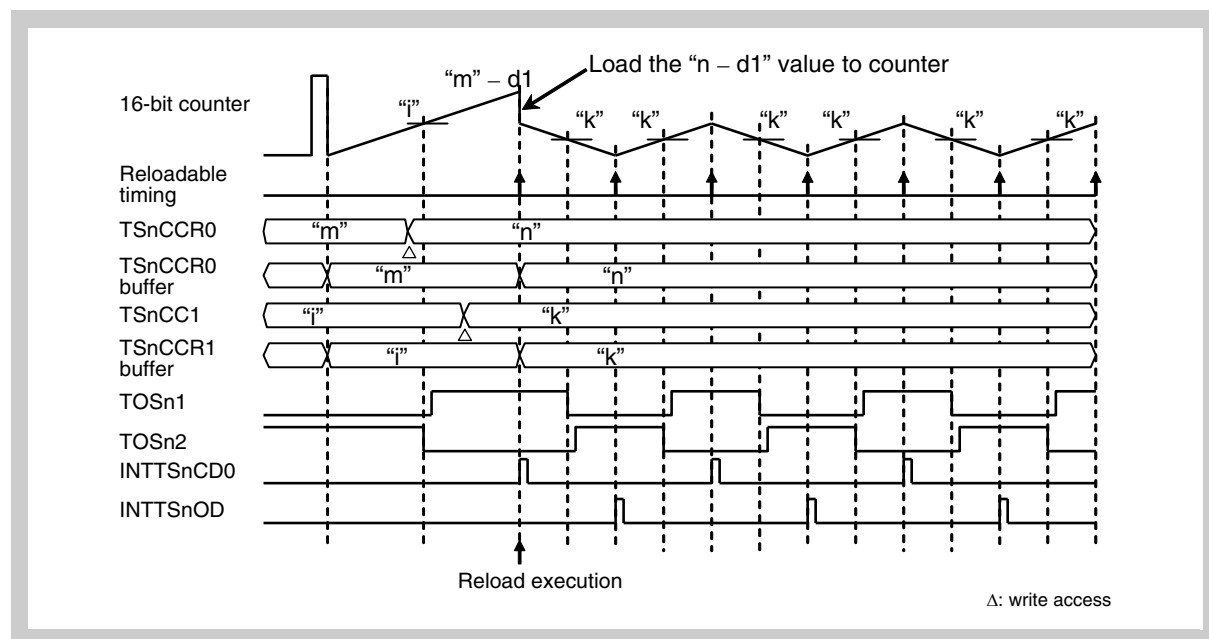
**(b) Caution for TSnCCR0 register in high-accuracy T-PWM mode**

The output waveform changes according to reload occurrence at peak/trough when rewriting the TSnCCR0 register in the reload mode (batch rewrite function) (when TSnCMS of the TSnOPT0 register is 0, TSnRTE is 1, TSnRBE is 1, TSnRDE is 0, TSnICE is 1, and TSnIOE is 1 in the TSnOPT1 register).

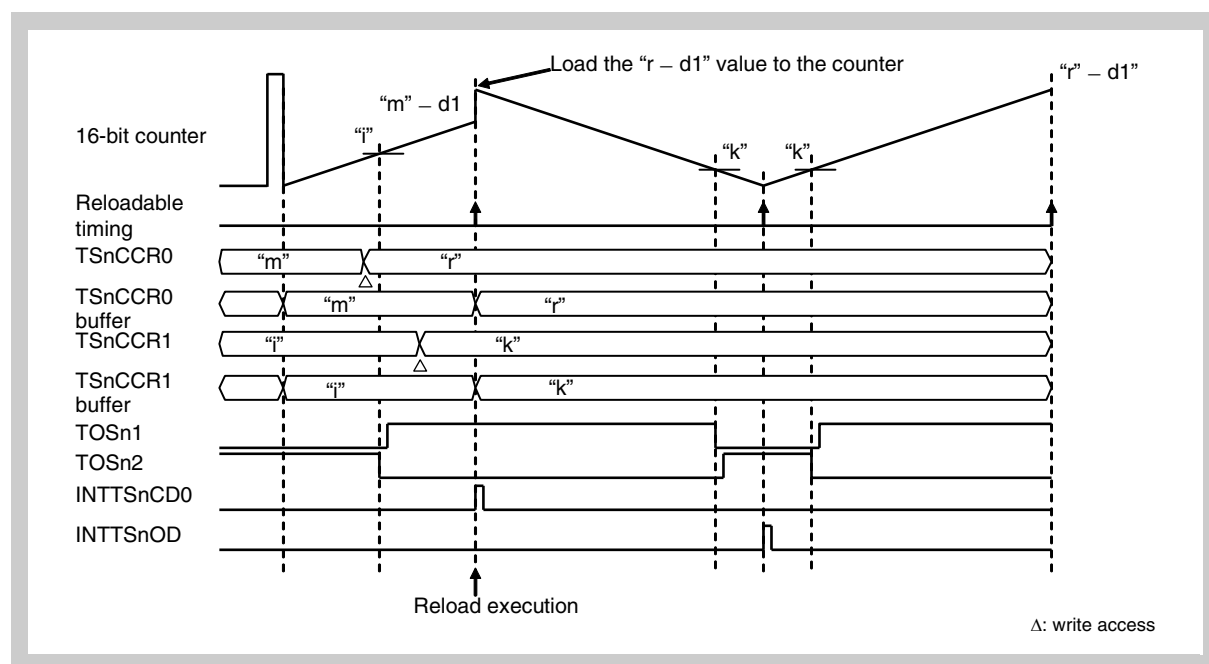


Rewriting in the period of <1> (rewriting during count-up operation)

The next reload timing causes cycle change and asymmetric triangular waveform output on the count-down side because of the peak point reload timing. The duty value must be set again due to cycle change.



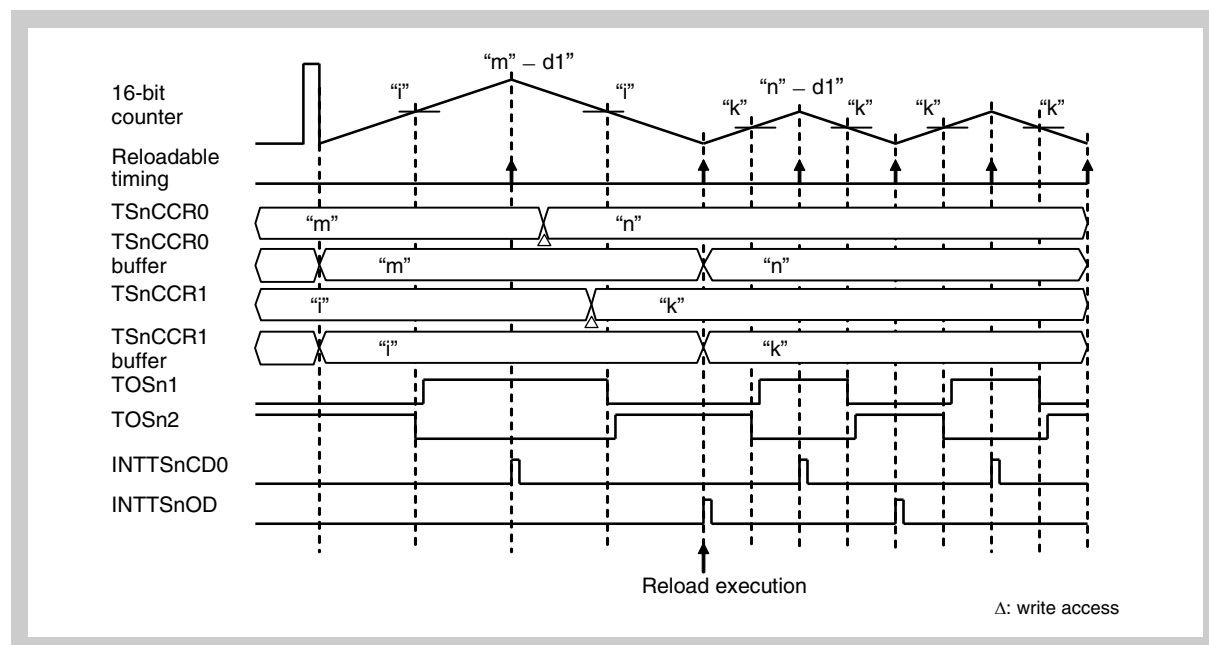
**Note** d1: TSnDTC1 setting value



**Note** d1: TSnDTC1 setting value

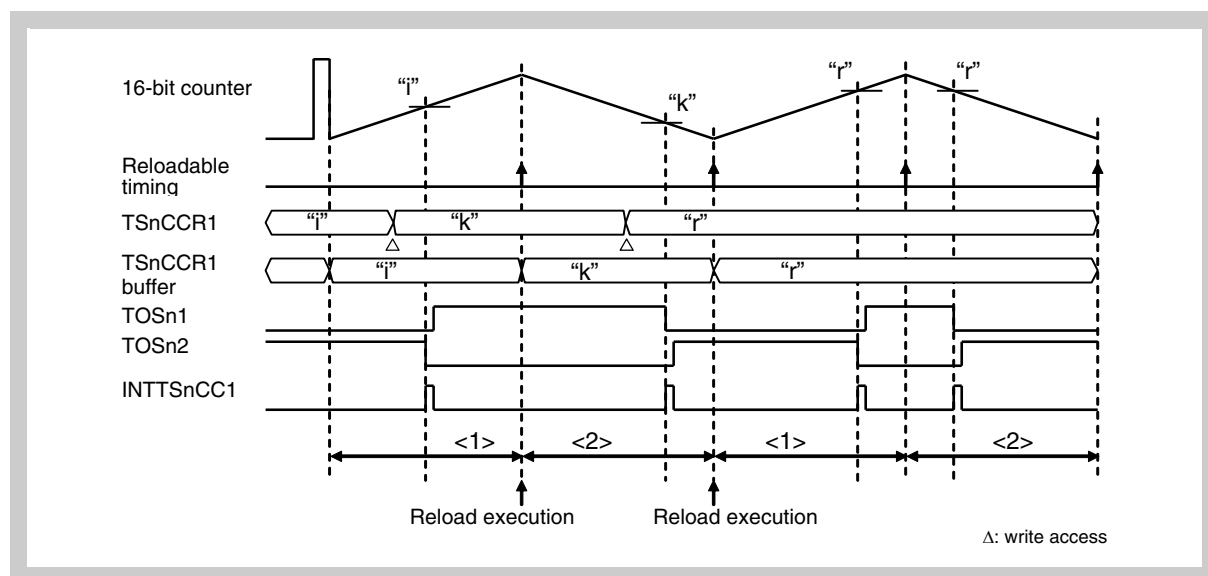
Rewriting at period <2> (Rewriting during count-down operation)

Because the next reload timing is at the trough point, the cycle value changes from the next cycle and the output remains a triangular waveform. The duty value must be set again due to cycle change.



**Note** d1: TSnDTC1 setting value

### (c) Cautions for TSnCCR1 to TSnCCR3 registers in high-accuracy T-PWM mode



Rewriting in the period <1> (rewriting during count-up operation)

An asymmetric triangular waveform is output because of reload at the peak interrupt timing.

Rewriting in the period <2> (rewriting during count-down operation)

A symmetric triangular waveform is output because of reload at the trough interrupt timing.

#### (d) Caution on rewriting TSnOPT1

Writing to the TSnOPT1 or TSnOPT1L registers clears the internal interrupt thinning out counter. Then, the interrupt interval may be longer temporarily when the interrupt thinning out function is used.

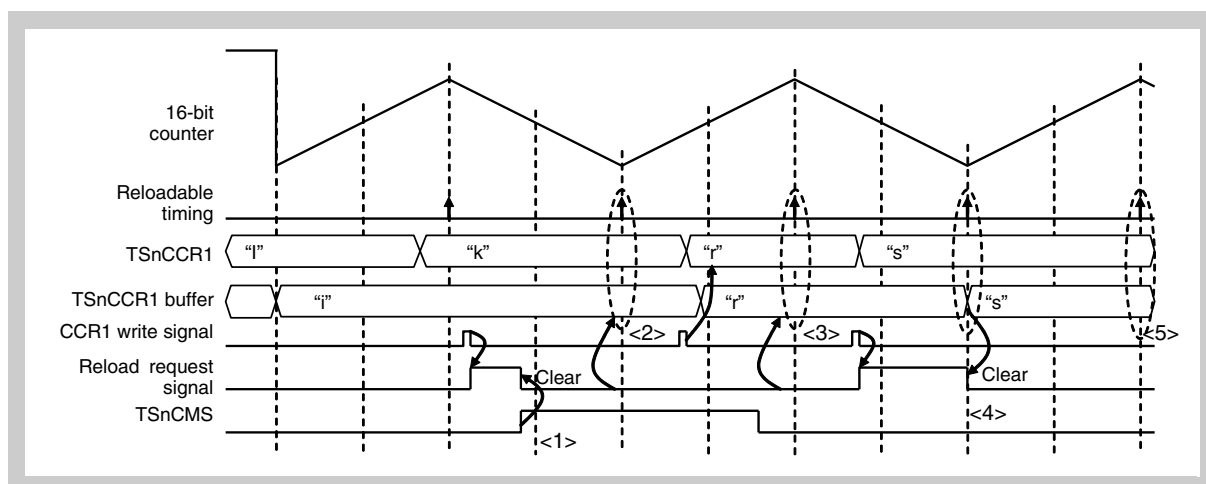
To avoid this situation, rewriting must be executed by performing a setting such that reloading is done in synchronization with interrupt thinning out (TSnCMS of the TSnOPT0 register is 0, or TSnRDE of the TSnOPT1 register is 1).

**(e) Caution on rewriting TSnCMS bit in high-accuracy T-PWM mode**

Rewriting the TSnCMS bit of the TSnOPT0 register is possible during operation ( $\text{TSnCE} = 1$ ), but care must be exercised in the TSnCCR1 register writing order.

Writing to the TSnCCR1 register under  $\text{TSnCMS} = 0$  enables setting of the reload request signal (internal signal).

Setting of the reload request signal causes reload at the next reload timing, and the reload request signal is then cleared. The reload request signal is cleared by  $\text{TSnCMS} = 1$  as well.



- <1> Writing to the TSnCCR1 register while  $\text{TSnCMS} = 0$  enables setting of the reload request signal (internal signal). The reload request signal is cleared by setting  $\text{TSnCMS}$  to 1 under the above status.
- <2> Reload is not executed because the reload request signal is cleared by  $\text{TSnCMS} = 1$ .
- <3> Writing to the TSnCCR1 register while  $\text{TSnCMS} = 1$  does not set the reload request signal. If  $\text{TSnCMS}$  is set to 0 after that, reload is not executed at the next reload timing.
- <4> Writing to the TSnCCR1 register while  $\text{TSnCMS} = 0$  sets the reload request signal. Reload is executed at the next reload timing to clear the reload request signal.
- <5> Once reload is executed, the reload request signal is cleared. Therefore, reload is not executed at the next reload timing until the TSnCCR1 register is written to.

### 17.4.3 Output list in each mode

#### (1) Timer output by each mode

The following list shows the timer outputs (TOSn0 to TOSn7 pins) in each mode.

**Table 17-24 Timer Outputs in Each Mode (1/2)**

Operation mode	TOSn0	TOSn1	TOSn2	TOSn3
Interval timer mode	Toggle output by TSnCCR0 compare match	Toggle output by TSnCCR1 compare match	Toggle output by TSnCCR2 compare match	Toggle output by TSnCCR3 compare match
External event count mode	Toggle output by TSnCCR0 compare match	Toggle output by TSnCCR1 compare match	Toggle output by TSnCCR2 compare match	Toggle output by TSnCCR3 compare match
External trigger pulse output mode	Toggle output by TSnCCR0 compare match or external trigger input	External trigger pulse waveform output	External trigger pulse waveform output	External trigger pulse waveform output
One-shot pulse mode	Active at count start Inactive by TSnCCR0 match	Active by TSnCCR1 match Inactive by TSnCCR0 match	Active by TSnCCR2 match Inactive by TSnCCR0 match	Active by TSnCCR3 match Inactive by TSnCCR0 match
PWM mode	Toggle output by TSnCCR0 compare match	PWM output by TSnCCR1 compare match	PWM output by TSnCCR2 compare match	PWM output by TSnCCR3 compare match
Free-running mode	Toggle output by TSnCCR0 compare match	Toggle output by TSnCCR1 compare match	Toggle output by TSnCCR2 compare match	Toggle output by TSnCCR3 compare match
Triangular-wave PWM mode	Up/down status output of 16-bit counter	PWM output by TSnCCR1 compare match	PWM output by TSnCCR2 compare match	PWM output by TSnCCR3 compare match
High-accuracy T-PWM mode	Up/down status output of 16-bit counter or 16-bit sub-counter	PWM output by TSnCCR1 compare match (with dead time)	Negative phase output against TOSn1	PWM output by TSnCCR2 compare match (with dead time)
PWM mode with dead time	Toggle output by TSnCCR0 compare match	PWM output by TSnCCR1 compare match (with dead time)	Negative phase output against TOSn1	PWM output by TSnCCR2 compare match (with dead time)
120° excitation mode	Toggle output by TSnCCR0 compare match	PWM output using TSnCCR1, TSnCCR2	PWM output using TSnCCR3, TSnCCR4	PWM output using TSnCCR1, TSnCCR2
Special 120° excitation mode	Toggle output by TSnCCR0 compare match	Pattern output by TSnPAT0	Pattern output by TSnPAT1	Pattern output by TSnPAT0
Special pattern output mode	Pattern output by TSnPAT0[3:0]	Pattern output by TSnPAT0[7:4]	Pattern output by TSnPAT0[11:8]	Pattern output by TSnPAT0[15:12]

**Table 17-25 Timer Outputs in Each Mode (2/2)**

Operation mode	TOSn4	TOSn5	TOSn6	TOSn7
Interval timer mode	Toggle output by TSnCCR4 compare match	Toggle output by TSnCCR5 compare match	-	-

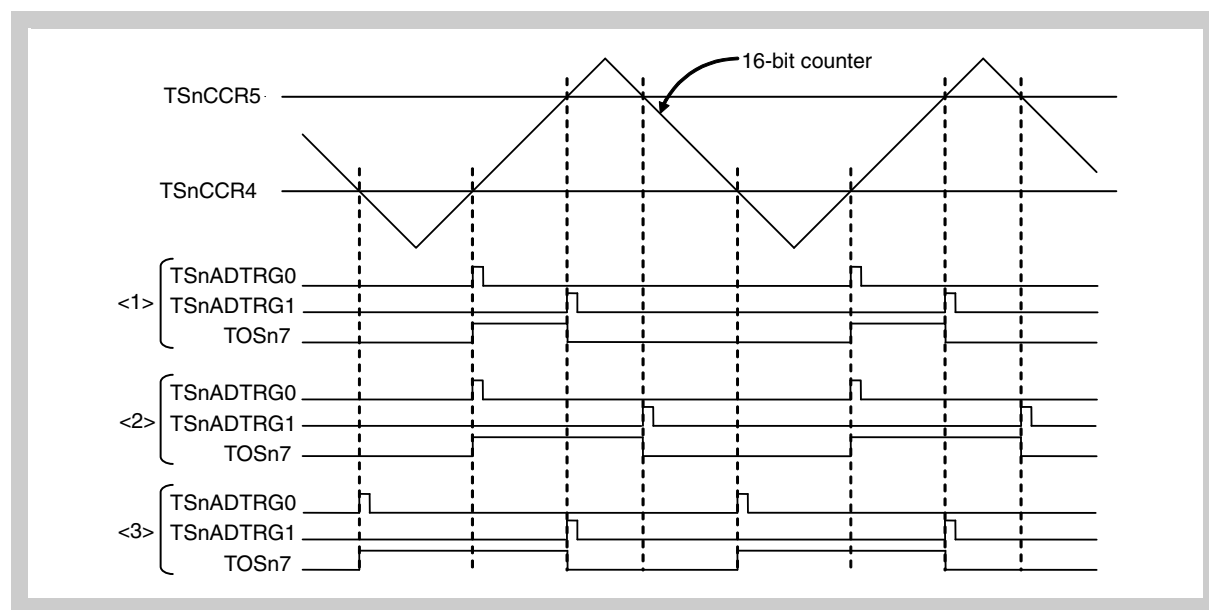
Operation mode	TOSn4	TOSn5	TOSn6	TOSn7
External event count mode	Toggle output by TSnCCR4 compare match	Toggle output by TSnCCR5 compare match	-	-
External trigger pulse output mode	External trigger pulse waveform output	External trigger pulse waveform output	-	-
One-shot pulse mode	Active by TSnCCR4 match Inactive by TSnCCR0 match	Active by TSnCCR5 match Inactive by TSnCCR0 match	-	-
PWM mode	PWM output by TSnCCR4 compare match	PWM output by TSnCCR5 compare match	-	Pulse output by A/D conversion trigger <sup>Note 1</sup>
Free-running mode	Toggle output by TSnCCR4 compare match	Toggle output by TSnCCR5 compare match	-	-
Triangular-wave PWM mode	PWM output by TSnCCR4 compare match	PWM output by TSnCCR5 compare match	-	Pulse output by A/D conversion trigger <sup>Note 1</sup>
High-accuracy T-PWM mode	Negative phase output against TOSn3	PWM output by TSnCCR3 compare match (with dead time)	Negative phase output against TOSn5	Pulse output by A/D conversion trigger <sup>Note 1</sup>
PWM mode with dead time	Negative phase output against TOSn3	PWM output by TSnCCR3 compare match (with dead time)	Negative phase output against TOSn5	Pulse output by A/D conversion trigger <sup>Note 1</sup>
120° excitation mode	PWM output using TSnCCR3, TSnCCR4	PWM output using TSnCCR1, TSnCCR2	PWM output using TSnCCR3, TSnCCR4	Pulse output by A/D conversion trigger <sup>Note 1</sup>
Special 120° excitation mode	Pattern output by TSnPAT1	Pattern output by TSnPAT0	Pattern output by TSnPAT1	Pulse output by A/D conversion trigger <sup>Note 1</sup>
Special pattern output mode	Pattern output by TSnPAT1[3:0]	Pattern output by TSnPAT1[7:4]	Pattern output by TSnPAT1[11:8]	Pattern output by TSnPAT1[15:12]

**Note 1.** Refer to “TOSn7 pin output control” on page 707.

**2.** – indicates inactive level output.

**(a) TOSn7 pin output control**

The A/D conversion trigger signal can be output to the TOSn7 pin. The TOSn7 pin becomes active at the rising edge of the TSnADTRG0 signal, and inactive at the rising edge of the TSnADTRG1 signal. If the TSnADTRG0 signal is detected during the active status of the TOSn7 pin, the TOSn7 pin remains active level. If the TSnADTRG1 signal is detected during the inactive status of the TOSn7 pin, the TOSn7 pin remains inactive level. The inactive status of the TOSn7 pin has priority if the TSnADTRG0 and TSnADTRG1 signal triggers are simultaneously generated.



**Figure 17-8 TOSn7 pin output timing**

**Note** Case 1: When TSnCCR4 < TSnCCR5, TSnOPT2 = 04H, TSnOPT3 = 10H  
 Case 2: When TSnCCR4 < TSnCCR5, TSnOPT2 = 04H, TSnOPT3 = 20H  
 Case 3: When TSnCCR4 < TSnCCR5, TSnOPT2 = 08H, TSnOPT3 = 10H

**(2) Interrupt in each mode**

The following list shows the interrupts (INTTSnCC0 to INTTSnCC5, INTTSnOV, INTTSnER, INTTSnWN) in each mode.

**Table 17-26 Interrupts in Each Mode (1/3)**

Operation mode	INTTSnCC0	INTTSnCC1	INTTSnCC2	INTTSnCC3
Interval timer mode	TSnCCR0 compare match interrupt	TSnCCR1 compare match interrupt <sup>Note 2</sup>	TSnCCR2 compare match interrupt <sup>Note 2</sup>	TSnCCR3 compare match interrupt <sup>Note 2</sup>
External event count mode	TSnCCR0 compare match interrupt	TSnCCR1 compare match interrupt <sup>Note 2</sup>	TSnCCR2 compare match interrupt <sup>Note 2</sup>	TSnCCR3 compare match interrupt <sup>Note 2</sup>
External trigger pulse output mode	TSnCCR0 compare match interrupt	TSnCCR1 compare match interrupt <sup>Note 2</sup>	TSnCCR2 compare match interrupt <sup>Note 2</sup>	TSnCCR3 compare match interrupt <sup>Note 2</sup>
One-shot pulse mode	TSnCCR0 compare match interrupt	TSnCCR1 compare match interrupt <sup>Note 2</sup>	TSnCCR2 compare match interrupt <sup>Note 2</sup>	TSnCCR3 compare match interrupt <sup>Note 2</sup>
PWM mode	TSnCCR0 compare match interrupt	TSnCCR1 compare match interrupt <sup>Note 2</sup>	TSnCCR2 compare match interrupt <sup>Note 2</sup>	TSnCCR3 compare match interrupt <sup>Note 2</sup>
Free-running mode	TSnCCR0 compare match interrupt	TSnCCR1 compare match interrupt	TSnCCR2 compare match interrupt	TSnCCR3 compare match interrupt
Triangular-wave PWM mode	TSnCCR0 compare match interrupt	TSnCCR1 compare match interrupt <sup>Note 2</sup>	TSnCCR2 compare match interrupt <sup>Note 2</sup>	TSnCCR3 compare match interrupt <sup>Note 2</sup>
High-accuracy T-PWM mode	TSnCCR0 compare match interrupt <sup>Note 1</sup>	TSnCCR1 compare match interrupt <sup>Note 3</sup>	TSnCCR2 compare match interrupt <sup>Note 3</sup>	TSnCCR3 compare match interrupt <sup>Note 3</sup>
PWM mode with dead time	TSnCCR0 compare match interrupt	TSnCCR1 compare match interrupt <sup>Note 2</sup>	TSnCCR2 compare match interrupt <sup>Note 2</sup>	TSnCCR3 compare match interrupt <sup>Note 2</sup>
120° excitation mode	TSnCCR0 compare match interrupt	TSnCCR1 compare match interrupt <sup>Note 2</sup>	TSnCCR2 compare match interrupt <sup>Note 2</sup>	TSnCCR3 compare match interrupt <sup>Note 2</sup>
Special 120° excitation mode	TSnCCR0 compare match interrupt	TSnCCR1 compare match interrupt <sup>Note 2</sup>	TSnCCR2 compare match interrupt <sup>Note 2</sup>	TSnCCR3 compare match interrupt <sup>Note 2</sup>
Special pattern output mode	TSnCCR0 compare match interrupt	TSnCCR1 compare match interrupt <sup>Note 2</sup>	TSnCCR2 compare match interrupt <sup>Note 2</sup>	TSnCCR3 compare match interrupt <sup>Note 2</sup>

- Note**
1. A compare match interrupt occurs only when 000H is set to the TSnDTC1 register. INTTSnCD0 can be used for peak interrupt.
  2. Compare match interrupt does not occur when setting  $TSnCCR0 < TSnCCRM$ .
  3. Compare match interrupt does not occur when setting is performed within the range of  $0000H \leq TSnCCRM < TSnDTC0$ ,  $(TSnCCR0 - TSnDTC1) < TSnCCRM$ .
  4.  $m = 1$  to 5



**Table 17-27 Interrupts in Each Mode (2/3)**

Operation mode	INTTSnCC4	INTTSnCC5
Interval timer mode	TSnCCR4 compare match interrupt <sup>Note 1</sup>	TSnCCR5 compare match interrupt <sup>Note 1</sup>
External event count mode	TSnCCR4 compare match interrupt <sup>Note 1</sup>	TSnCCR5 compare match interrupt <sup>Note 1</sup>
External trigger pulse output mode	TSnCCR4 compare match interrupt <sup>Note 1</sup>	TSnCCR5 compare match interrupt <sup>Note 1</sup>
One-shot pulse mode	TSnCCR4 compare match interrupt <sup>Note 1</sup>	TSnCCR5 compare match interrupt <sup>Note 1</sup>
PWM mode	TSnCCR4 compare match interrupt <sup>Note 1</sup>	TSnCCR5 compare match interrupt <sup>Note 1</sup>
Free-running mode	TSnCCR4 compare match interrupt	TSnCCR5 compare match interrupt
Triangular-wave PWM mode	TSnCCR4 compare match interrupt <sup>Note 1</sup>	TSnCCR5 compare match interrupt <sup>Note 1</sup>
High-accuracy T-PWM mode	TSnCCR4 compare match interrupt <sup>Note 2</sup>	TSnCCR5 compare match interrupt <sup>Note 2</sup>
PWM mode with dead time	TSnCCR4 compare match interrupt <sup>Note 1</sup>	TSnCCR5 compare match interrupt <sup>Note 1</sup>
120° excitation mode	TSnCCR4 compare match interrupt <sup>Note 1</sup>	TSnCCR5 compare match interrupt <sup>Note 1</sup>
Special 120° excitation mode	TSnCCR4 compare match interrupt <sup>Note 1</sup>	TSnCCR5 compare match interrupt <sup>Note 1</sup>
Special pattern output mode	TSnCCR4 compare match interrupt <sup>Note 1</sup>	TSnCCR5 compare match interrupt <sup>Note 1</sup>

- Note**
1. Compare match interrupt does not occur when setting  $TSnCCR0 < TSnCCRm$ .
  2. Compare match interrupt is not generated when setting within the range of  $0000H \leq TSnCCRm < TSnDTC0$ ,  $(TSnCCR0 - TSnDTC1) < TSnCCRm$ .
  3.  $m = 1$  to 5

**Table 17-28** Interrupts in Each Mode (3/3)

Operation mode	INTTSnOV	INTTSnER	INTTSnWN
Interval timer mode	-	-	-
External event count mode	-	-	-
External trigger pulse output mode	-	-	-
One-shot pulse mode	-	-	-
PWM mode	-	Error interrupt	-
Free-running mode	Overflow interrupt	-	-
Triangular-wave PWM mode	-	Error interrupt	-
High-accuracy T-PWM mode	Overflow interrupt <sup>Note 1</sup>	Error interrupt	Warning interrupt <sup>Note 2</sup>
PWM mode with dead time	-	Error interrupt	-
120° excitation mode	-	Error interrupt	Warning interrupt
Special 120° excitation mode	-	Error interrupt	Warning interrupt
Special pattern output mode	-	Error interrupt	-

- Note**
1. An overflow interrupt (INTTSnOV) occurs when setting TSnCCR0, TSnDTC0, and TSnDTC1 incorrectly.
  2. Only when the 180° excitation control function is used

**(3) A/D conversion trigger, peak interrupt, trough interrupt in each mode**

The A/D conversion triggers and peak/trough interrupts in each mode are listed below.

**Table 17-29 A/D Conversion Trigger, Peak Interrupt and Trough Interrupt**

Operation mode	TSnADTRG0	TSnADTRG1	INTTSnCD0	INTTSnOD
Interval timer mode	-	-	-	-
External event count mode	-	-	-	-
External trigger pulse output mode	-	-	-	-
One-shot pulse mode	-	-	-	-
PWM mode	Select from INTTSnCD0, INTTSnCC4, INTTSnCC5 interrupts <b>Note 1</b>	Select from INTTSnCD0, INTTSnCC4, INTTSnCC5 interrupts <b>Note 1</b>	Peak interrupt at the same timing with INTTSnCC0 interrupt	-
Free-running mode	-	-	-	-
Triangular-wave PWM mode	Select from INTTSnOD, INTTSnCC4, INTTSnCC5 interrupts <b>Note 1</b>	Select from INTTSnOD, INTTSnCC4, INTTSnCC5 interrupts <b>Note 1</b>	-	Trough interrupt
High-accuracy T-PWM mode	Select from INTTSnCD0, INTTSnOD, INTTSnCC4, INTTSnCC5 interrupts <b>Note 1</b>	Select from INTTSnCD0, INTTSnOD, INTTSnCC4, INTTSnCC5 interrupts <b>Note 1</b>	Peak interrupt	Trough interrupt
PWM mode with dead time	Select from INTTSnCD0, INTTSnCC4, INTTSnCC5 interrupts <b>Note 1</b>	Select from INTTSnCD0, INTTSnCC4, INTTSnCC5 interrupts <b>Note 1</b>	Peak interrupt at the same timing with INTTSnCC0 interrupt	-
120° excitation mode	Select from INTTSnCD0, INTTSnCC4, INTTSnCC5 interrupts <b>Note 1</b>	Select from INTTSnCD0, INTTSnCC4, INTTSnCC5 interrupts <b>Note 1</b>	Peak interrupt at the same timing with INTTSnCC0 interrupt	-
Special 120° excitation mode	Select from INTTSnCD0, INTTSnCC4, INTTSnCC5 interrupts <b>Note 1</b>	Select from INTTSnCD0, INTTSnCC4, INTTSnCC5 interrupts <b>Note 1</b>	Peak interrupt at the same timing with INTTSnCC0 interrupt	-
Special pattern output mode	Select from INTTSnCD0, INTTSnCC4, INTTSnCC5 interrupts <b>Note 1</b>	Select from INTTSnCD0, INTTSnCC4, INTTSnCC5 interrupts <b>Note 1</b>	Peak interrupt at the same timing with INTTSnCC0 interrupt	-

- 
- Note**
1. The TSnCCR4 and TSnCCR5 registers execute output control of the TOSn4 and TOSn5 pins (TOSn2, TOSn4, and TOSn6 pins in the 120° excitation mode) as well as determine the INTTSnCC4 and INTTSnCC5 timings. Therefore, the output timings of the TOSn4 and TOSn5 pins must be considered when using INTTSnCC4 and INTTSnCC5.
  2. INTTSnCD0 interrupt and INTTSnOD interrupt are the generation condition after interrupt thinning out.

## 17.5 Match Interrupt

Interrupt types include compare match interrupts (INTTSnCC0 to INTTSnCC5), peak interrupt (INTTSnCD0) and trough interrupt (INTTSnOD). Refer to “Error/warning Interrupt” on page 752 for error/warning interrupts (INTTSNER, INTTSnWN).

Compare match interrupts (INTTSnCC0 to INTTSnCC5) occur regardless of the operation mode.

The peak interrupt (INTTSnCD0) occurs in the PWM mode, high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode. This interrupt is output at the switch timing from count-up operation to count-down operation of the 16-bit counter in the high-accuracy T-PWM mode. In the modes with 16-bit counter sawtooth wave operation (PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode), the peak interrupt occurs after match of the 16-bit counter and the TSnCCR0 register values (at the same timing as an INTTSnCC0 interrupt). The trough interrupt occurs at the switch timing from count-down operation to count-up operation of the 16-bit counter in the triangular-wave PWM output mode and high-accuracy T-PWM mode.

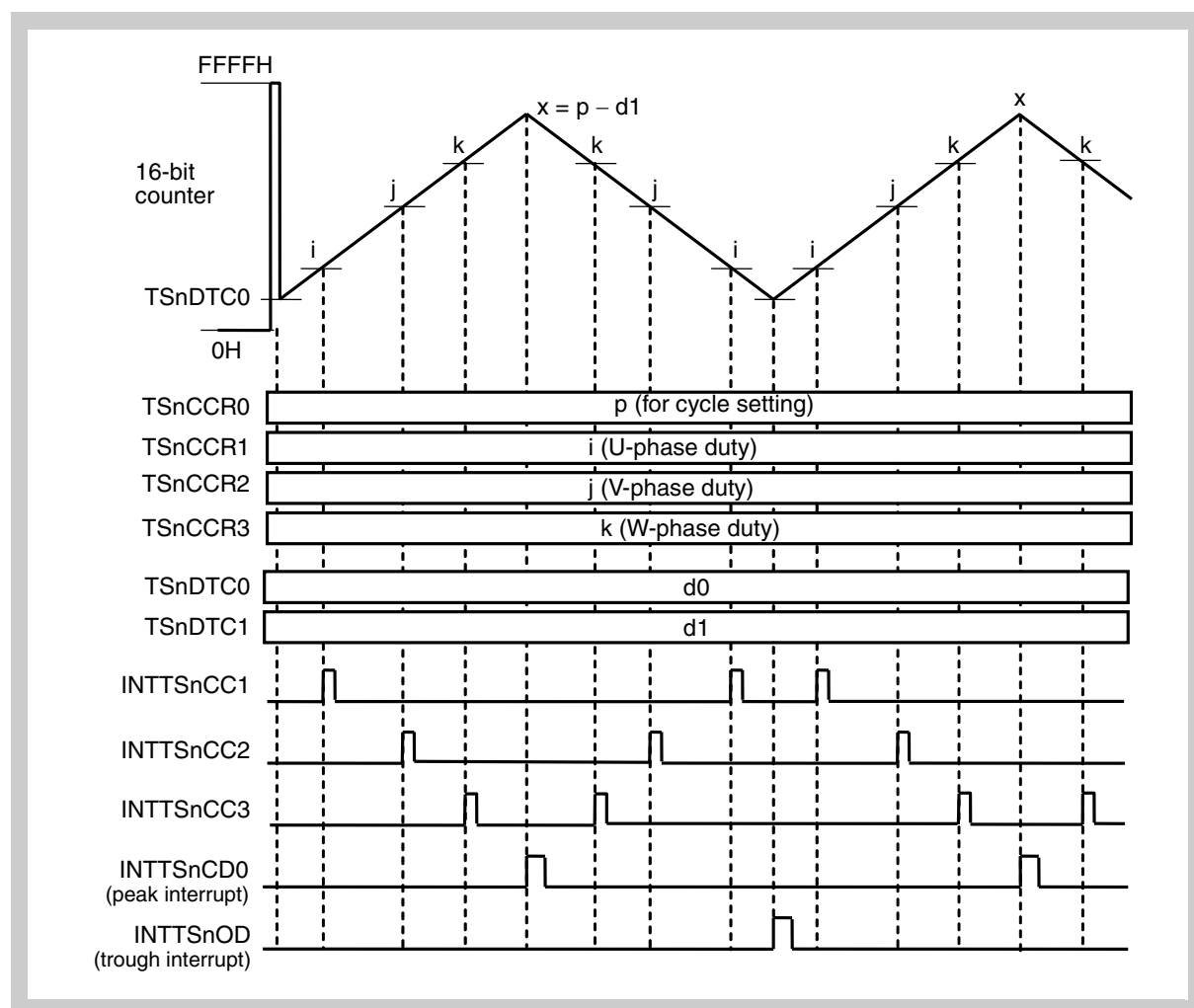
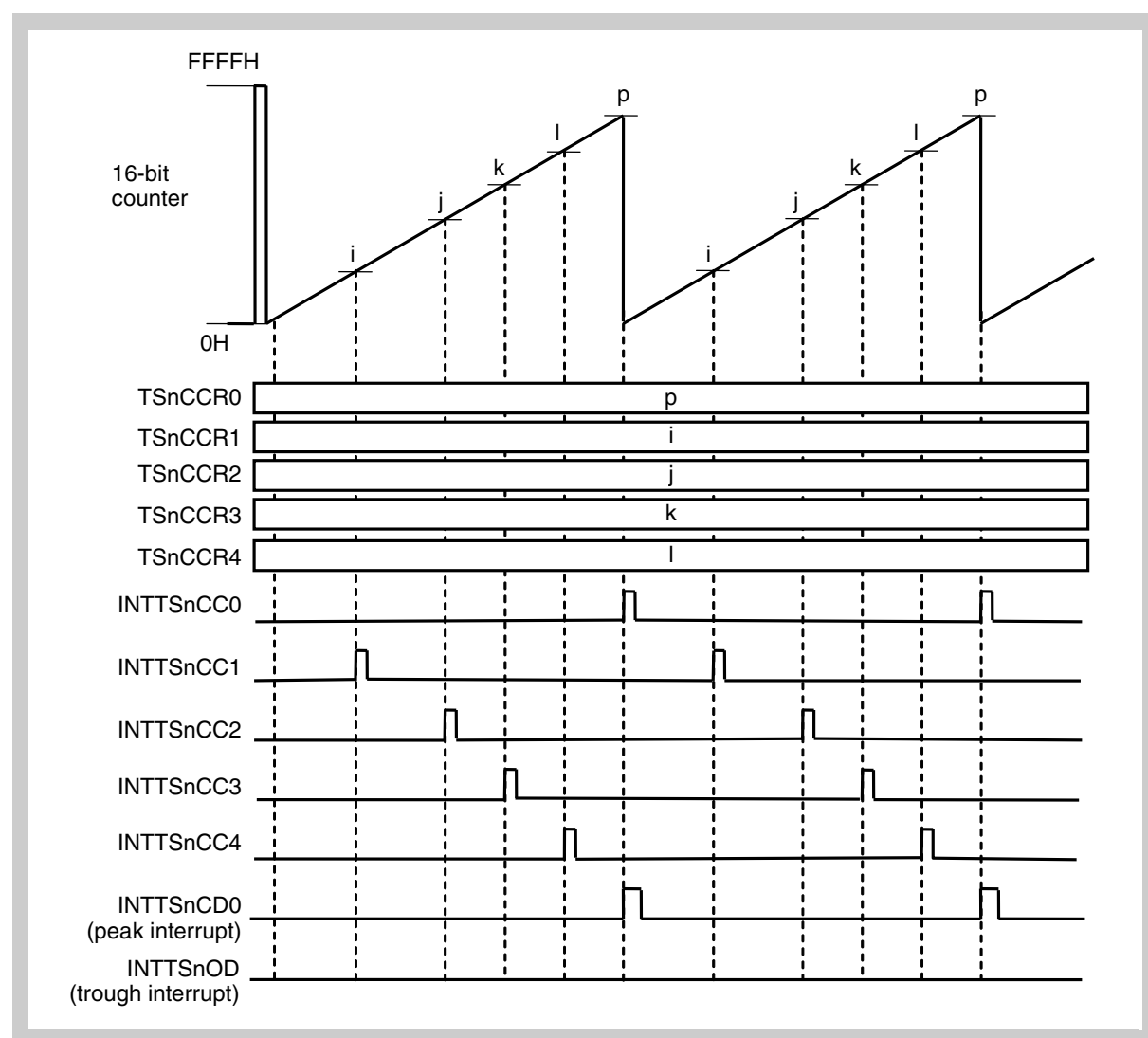


Figure 17-9 Interrupt signal output example (1/2)  
(high-accuracy T-PWM mode)



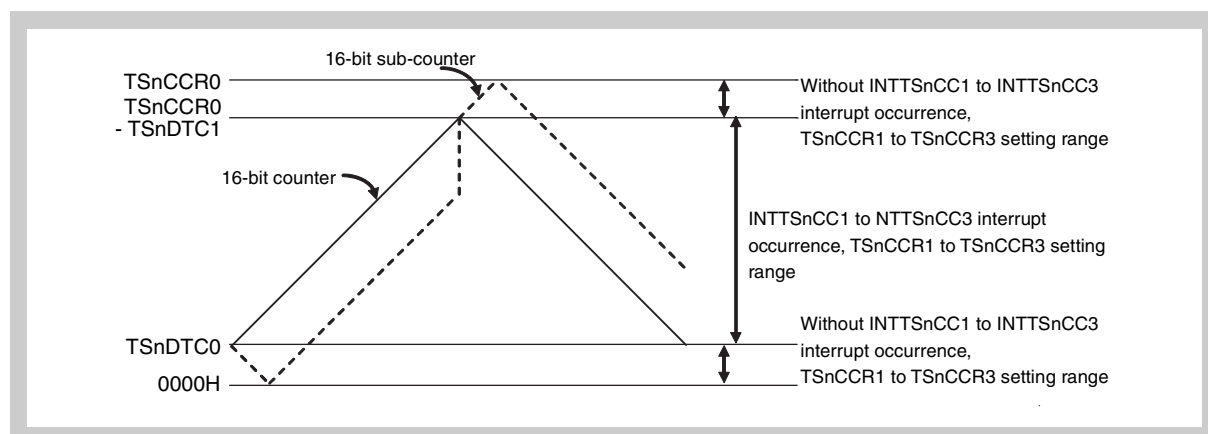
**Figure 17-10** Interrupt signal output example (2/2)  
(example in PWM mode)

## 17.5.1 Caution on compare match interrupt

### (1) Caution on high-accuracy T-PWM mode

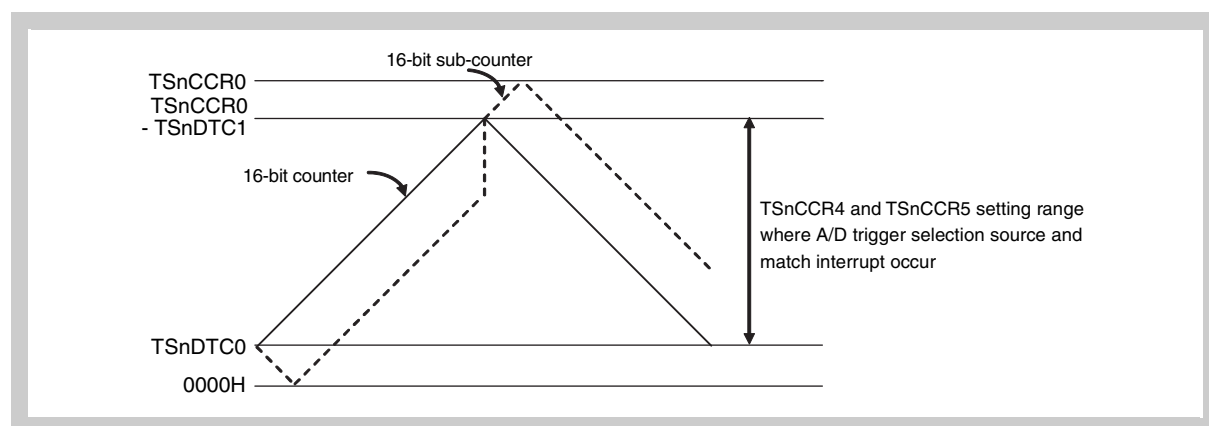
A compare match interrupt occurs upon a match between the 16-bit counter value and compare register (TSnCCR0 to TSnCCR5) values. Compare register setting beyond the 16-bit counter's count operation range is possible in the high-accuracy T-PWM mode. Therefore, compare match interrupts do not occur under the following conditions.

- Restriction concerning compare match interrupt of TSnCCR0 and the INTTSnCC0 register  
In the high-accuracy T-PWM mode, compare match interrupts (INTTSnCC0) does not occur while TSnDTC1  $\neq$  0000H.  
Use INTTSnOD (trough interrupt)/INTTSnCD0 (peak interrupt) for the interrupt as a cycle.
- Restriction concerning the TSnCCR1 to TSnCCR3 registers  
A compare match interrupt of match between the compare value and the 16-bit counter value is not generated in the high-accuracy T-PWM mode if set within the following range.  
 $0000H \leq \text{TSnCCR1 to TSnCCR3} < \text{TSnDTC0}$ ,  
 $\text{TSnCCR0 to TSnDTC1} + 1 < \text{TSnCCR1 to TSnCCR3} \leq \text{TSnCCR0}$



- Restriction concerning the TSnCCR4 and TSnCCR5 registers  
A compare match interrupt does not occur since a match between compare value and the 16-bit counter value is not generated in the high-accuracy T-PWM mode if set within the following range.  
 $0000H \leq \text{TSnCCR4, TSnCCR5} < \text{TSnDTC0}$ ,  
 $\text{TSnCCR0 to TSnDTC1} + 1 < \text{TSnCCR4, TSnCCR5} \leq \text{TSnCCR0}$

The setting must be within the range of  $\text{TSnDTC0} \leq \text{TSnCCR4}$ ,  $\text{TSnCCR5} \leq \text{TSnCCR0}$  to TSnDTC1 because the TSnCCR4 and TSnCCR5 registers are used as a trigger source for A/D conversion triggers.



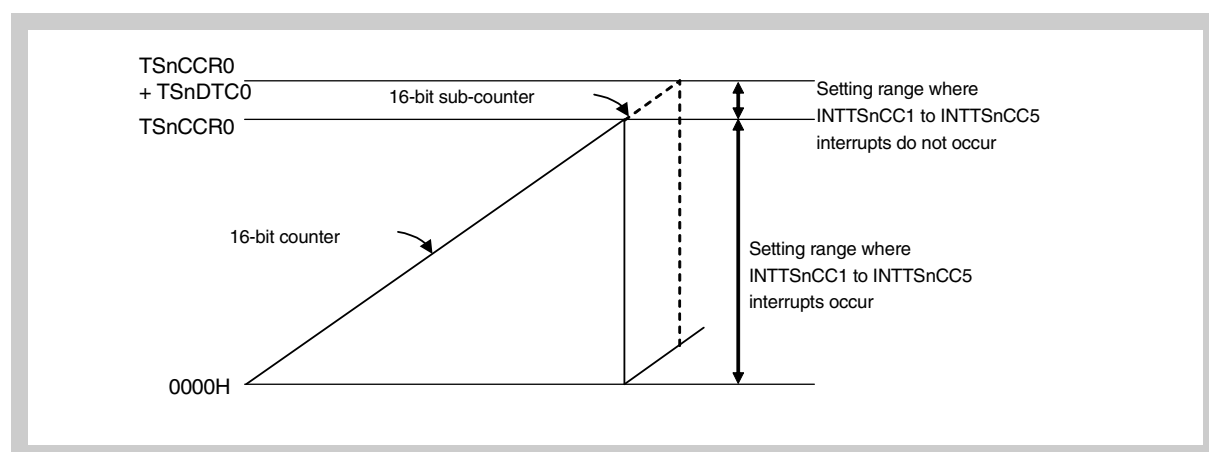
## (2) Caution on PWM mode with dead time

A compare match interrupt occurs upon a match of the 16-bit counter and compare register (TSnCCR0 to TSnCCR5). Compare register setting beyond the count operation range of the 16-bit counter is possible in the T-PWM mode with dead time. Therefore, compare match interrupts do not occur under the following conditions.

- Restrictions concerning the TSnCCR1 to TSnCCR5 registers  
A compare match interrupt does not occur if set within the following range in the PWM mode with dead time because no match occurs between the compare value and the 16-bit counter value.

$$TSnCCR0 < TSnCCR1 \text{ to } TSnCCR5 \leq TSnCCR0 + TSnDTC0$$

The setting must be within the range of TSnCCR4,  $TSnCCR5 \leq TSnCCR0$  when using the TSnCCR4 and TSnCCR5 registers because the TSnCCR4 and TSnCCR5 registers are used as trigger sources for A/D conversion triggers.



## 17.6 Flags

Table 17-30 Flag list

**Note** 1. Only in reload mode (when TSnCMS of the TSnOPT0 register is 0)



No.	Flag name	Symbol	Register	Operation mode
(1)	Count-up flag	TSnCUF	TSnOPT0	Triangular-wave PWM mode, high-accuracy T-PWM mode
		TSnSUF	TSnOPT6	High-accuracy T-PWM mode
(2)	Positive/negative simultaneous active detection flag	TSnTBF	TSnOPT6	PWM mode, triangular-wave PWM mode, high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode
(3)	Reload request flag	TSnRSF	TSnOPT6	External trigger pulse output mode, PWM mode, high-accuracy T-PWM mode <sup>Note 1</sup> , triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode
(4)	Noise detection flag	TSnNDF	TSnOPT6	High-accuracy T-PWM mode <sup>Note 2</sup> , PWM mode with dead time <sup>Note 2</sup> , 120° excitation mode, and special 120° excitation mode
(5)	Pattern order detection flag	TSnTSF	TSnOPT5	High-accuracy T-PWM mode <sup>Note 2</sup> , PWM mode with dead time <sup>Note 2</sup> , 120° excitation mode, and special 120° excitation mode
(6)	Pattern error detection flag	TSnPEF	TSnOPT6	High-accuracy T-PWM mode <sup>Note 2</sup> , PWM mode with dead time <sup>Note 2</sup> , 120° excitation mode, and special 120° excitation mode
(7)	Pattern reverse detection flag	TSnPRF	TSnOPT6	High-accuracy T-PWM mode <sup>Note 2</sup> , PWM mode with dead time <sup>Note 2</sup> , 120° excitation mode, and special 120° excitation mode
(8)	TAPTSn2 to TAPTSn0 pins abnormal toggle detection flag	TSnPTF	TSnOPT6	High-accuracy T-PWM mode <sup>Note 2</sup> , PWM mode with dead time <sup>Note 2</sup> , 120° excitation mode, and special 120° excitation mode
(9)	TSnSTCI0, TSnSTCI1 signal simultaneous trigger detection flag	TSnTDF	TSnOPT6	High-accuracy T-PWM mode <sup>Note 2</sup> , PWM mode with dead time <sup>Note 2</sup> , 120° excitation mode, and special 120° excitation mode
(10)	Pattern phase difference detection flag	TSnPPF	TSnOPT6	High-accuracy T-PWM mode <sup>Note 2</sup> , PWM mode with dead time <sup>Note 2</sup> , 120° excitation mode, and special 120° excitation mode
(11)	Overflow flag	TSnOVF	TSnOPT0	Free-running mode, and high-accuracy T-PWM mode <sup>Note 3</sup>
(12)	Timer output pattern flag	TSnOPF0 to TSnOPF2	TSnOPT5	High-accuracy T-PWM mode <sup>Note 4</sup> , PWM mode with dead time <sup>Note 4</sup> , 120° excitation mode, and special 120° excitation mode
(13)	Pattern switch detection flag (internal signal)	TSnESG	-	High-accuracy T-PWM mode <sup>Note 2</sup> , PWM mode with dead time <sup>Note 2</sup> , 120° excitation mode, and special 120° excitation mode

2. Selectable only when the software output control function (TSnSOC of the TSnOPT4 register is 1) or 180° excitation function (TSnADC of the TSnOPT5 register is 1) is used.
3. Selectable only when setting conditions for the TSnDTC0 and TSnDTC1 registers are incorrectly set.
4. Selectable only when the 180° excitation function (when TSnADC of the TSnOPT5 register is 1) is used.

### 17.6.1 Count-up flags (TSnCUF, TSnSUF)

[Name]

Count-up flag (TSnCUF) - TSnOPT0 register  
(TSnSUF) - TSnOPT6 register

[Description]

There are two kinds of count-up flags as follows.  
TSnCUF is a count-up/down flag of 16-bit counter.  
TSnSUF is a count-up/down flag of 16-bit sub-counter.  
For both TSnCUF and TSnSUF, 0 indicates count-up operation and 1 indicates count-down operation.

[Example]

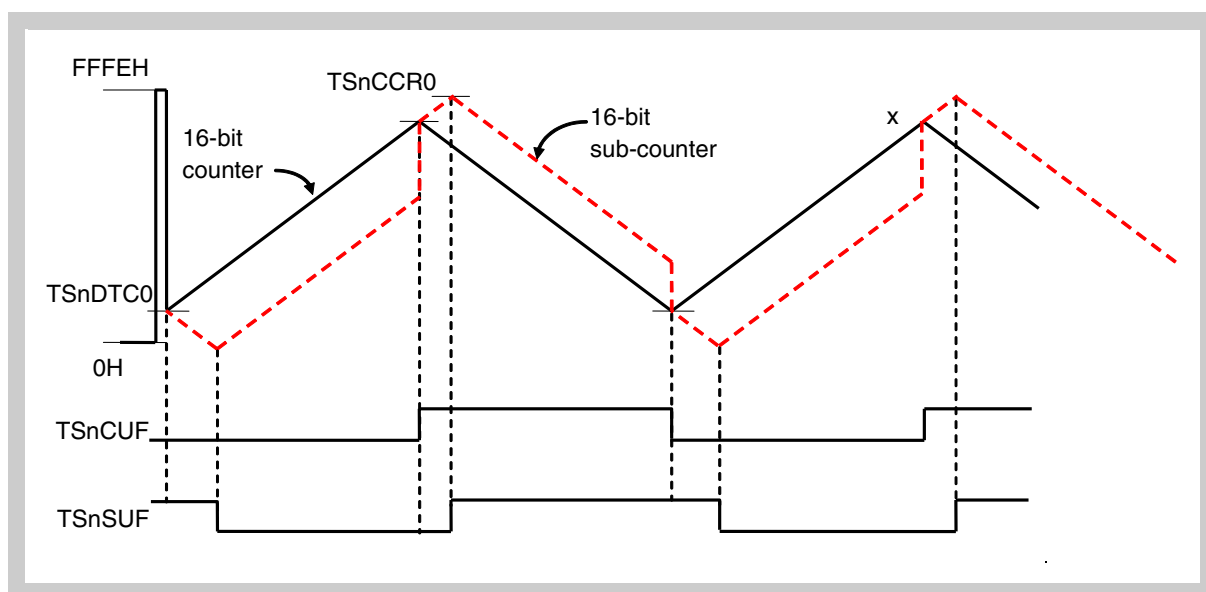


Figure 17-11 Count-up flag operation example

The TSnCUF value is as follows.

[In triangular-wave PWM mode]

$0 \leq 16\text{-bit counter} < \text{TSnCCR0} + 1 \dots\dots 0$  (count-up)  
 $\text{TSnCCR0} + 1 \geq 16\text{-bit counter} > 0 \dots\dots 1$  (count-down)

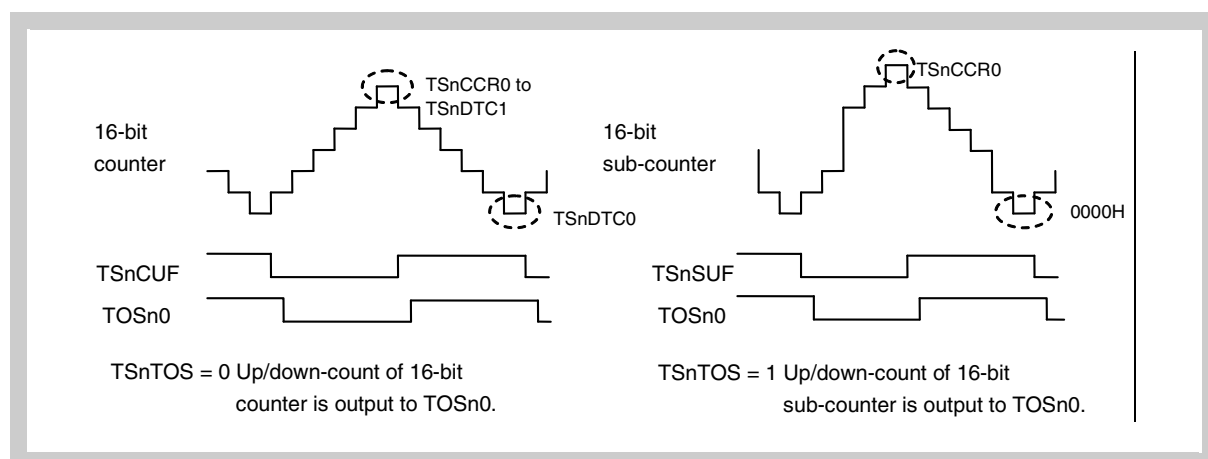
[In high-accuracy T-PWM mode]

$\text{TSnDTC0} + 2 \leq 16\text{-bit counter} < (\text{TSnCCR0} - \text{TSnDTC1}) \dots\dots 0$  (count-up)  
 $(\text{TSnCCR0} - \text{TSnDTC1} - 2) \geq 16\text{-bit counter} > \text{TSnDTC0} \dots\dots 1$  (count-down)

The TSnSUF value is as follows.

[In high-accuracy T-PWM mode]

$0 \leq 16\text{-bit sub-counter} < \text{TSnCCR0} \dots\dots 0$  (count-up)  
 $\text{TSnCCR0} \geq 16\text{-bit sub-counter} > 0 \dots\dots 1$  (count-down)



**Figure 17-12** TOSn0 Output at switch of TSnTOS bit of TSnOPT7 register

[Operation mode]

TSnCUF Triangular-wave PWM mode, high-accuracy T-PWM mode

TSnSUF High-accuracy T-PWM mode

### 17.6.2 Positive/negative-phase simultaneous active detection flag (TSnTBF)

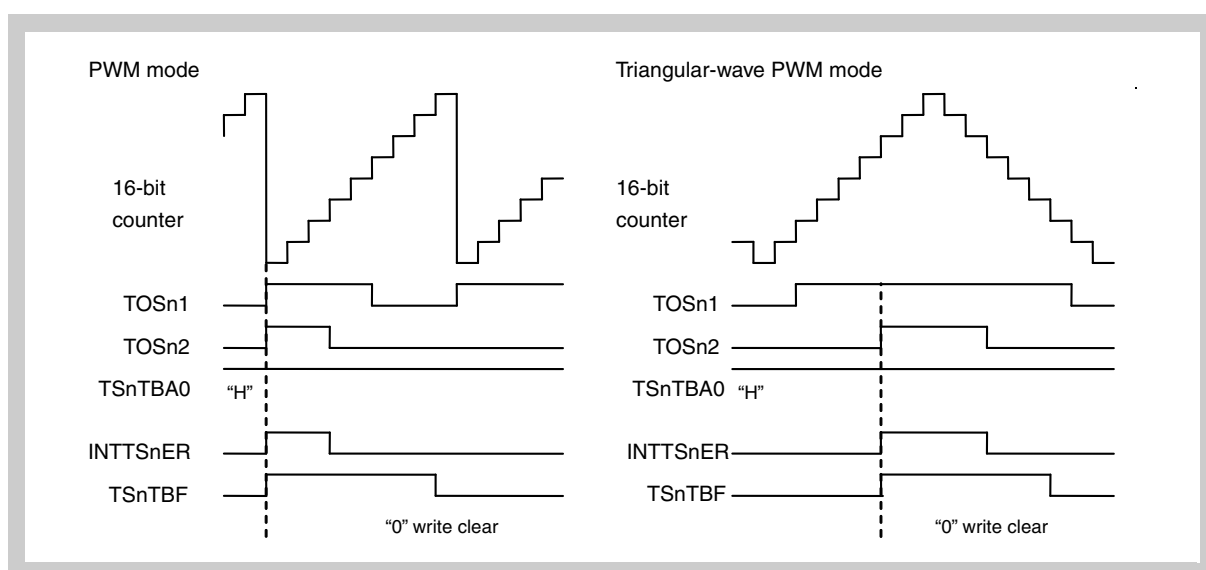
[Name]

Positive/negative phase simultaneous active detection flag (TSnTBF) -  
TSnOPT6 register

[Description]

Positive/negative phase simultaneous active of timer Sn can be detected by  
TSnTBF.

[Example]



**Figure 17-13** Positive/negative phase simultaneous active detection flag operation example

[Operation mode]

PWM mode, triangular-wave PWM mode, high-accuracy T-PWM mode,  
PWM mode with dead time, 120° excitation mode, special 120° excitation  
mode, special pattern output mode

### 17.6.3 Reload request flag (TSnRSF)

[Name]

Reload request flag (TSnRSF) - TSnOPT6 register

[Description]

TSnRSF is set to 1 when a reload request is generated (when the TSnCCR1 register is written). The reload request flag is cleared to 0 when reload is generated and the value is transferred to all the buffer registers.

[Example]

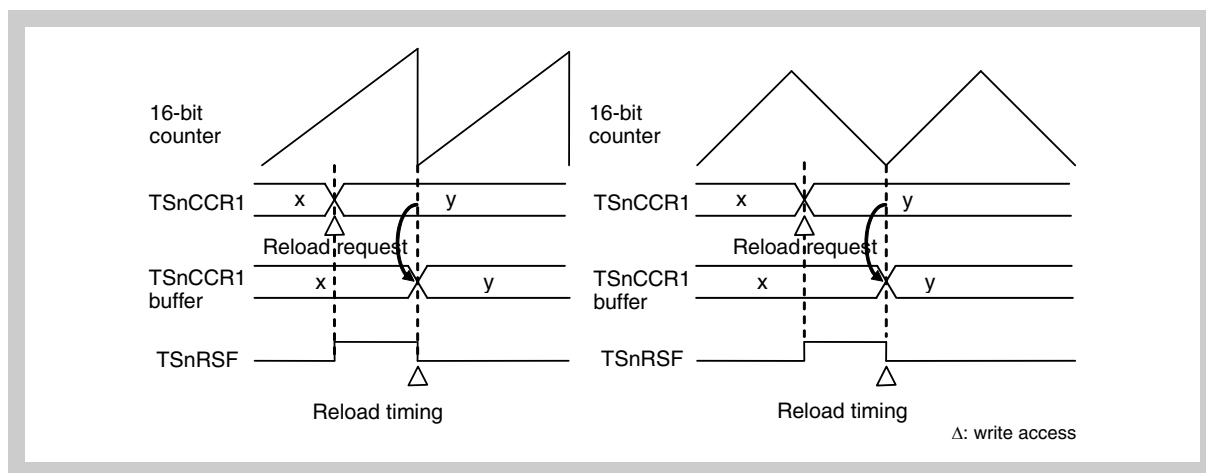


Figure 17-14 Reload request flag operation example

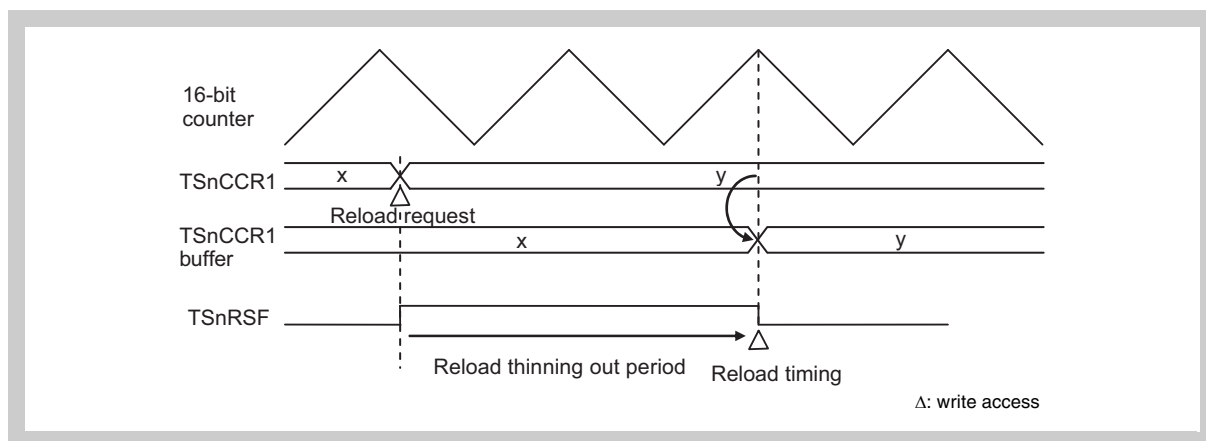


Figure 17-15 Reload request flag and reload thinning out period

## [Operation mode]

External trigger pulse output mode, PWM mode, high-accuracy T-PWM mode<sup>Note</sup>, triangular-wave PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, special pattern output mode

**Note** Only in reload mode (only when TSnCMS of the TSnOPT0 register is 0)

## 17.6.4 Noise detection flag (TSnNDF)

## [Name]

Noise detection flag (TSnNDF) - TSnOPT6 register

## [Description]

TSnNDF can detect noise generation on the TAPTSn2 to TAPTSn0 pins. "Noise" means simultaneous change on two or more pins among the TAPTSn2 to TAPTSn0 pins.

The noise detection flag (TSnNDF) is set to 1 when simultaneous change is detected on two or more pins among TAPTSn2 to TAPTSn0 pins. Clear to 0 is executed when 0 is written to the TSnNDF flag, or TSnCE of the TSnCTL0 register is set to 0.

## [Example]

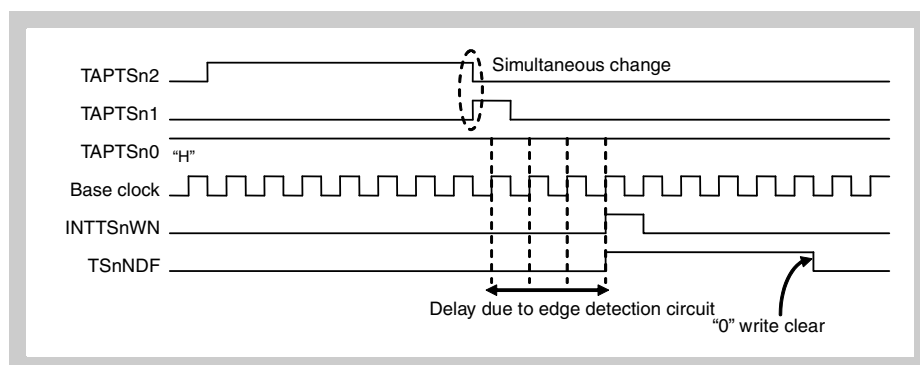


Figure 17-16 Noise detection flag operation example

## [Operation mode]

High-accuracy T-PWM mode<sup>Note</sup> and PWM mode with dead time<sup>Note</sup>

**Note** Selectable only when the software output control function (when TSnSOC of the TSnOPT4 register is 1) or 180° excitation function (when TSnADC of the TSnOPT5 register is 1) is used.

### 17.6.5 Pattern order detection flag (TSnTSF)

#### [Name]

TSnTSF detects the order of the pattern input to the TAPTSn2 to TAPTSn0 pins.

#### Pattern order detection flag (TSnTSF) - TSnOPT5 register

#### [Description]

The inputs the TAPTSn2 to TAPTSn0 pins result in the following operation.

- 0 is set when input to the TAPTSn2 to TAPTSn0 pins changes as follows.  
TAPTSn2 to TAPTSn0 =  
[1, 0, 1] → [1, 0, 0] → [1, 1, 0] → [0, 1, 0] → [0, 1, 1] → [0, 0, 1]
- 1 is set when input to TAPTSn2 to TAPTSn0 pins changes as follows.  
TAPTSn2 to TAPTSn0 =  
[1, 0, 1] ← [1, 0, 0] ← [1, 1, 0] ← [0, 1, 0] ← [0, 1, 1] ← [0, 0, 1]

#### [Example]

- When normal input is provided to the TAPTSn2 to TAPTSn0 pins  
As shown in *Figure 17-17 on page 723*, when the TAPTSn2 to TAPTSn0 pins are change in sequence, 0 or 1 is set according to the change order at the change timing.

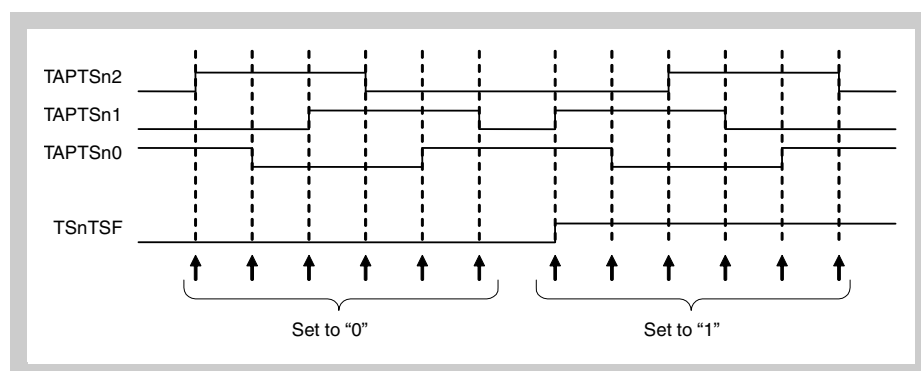
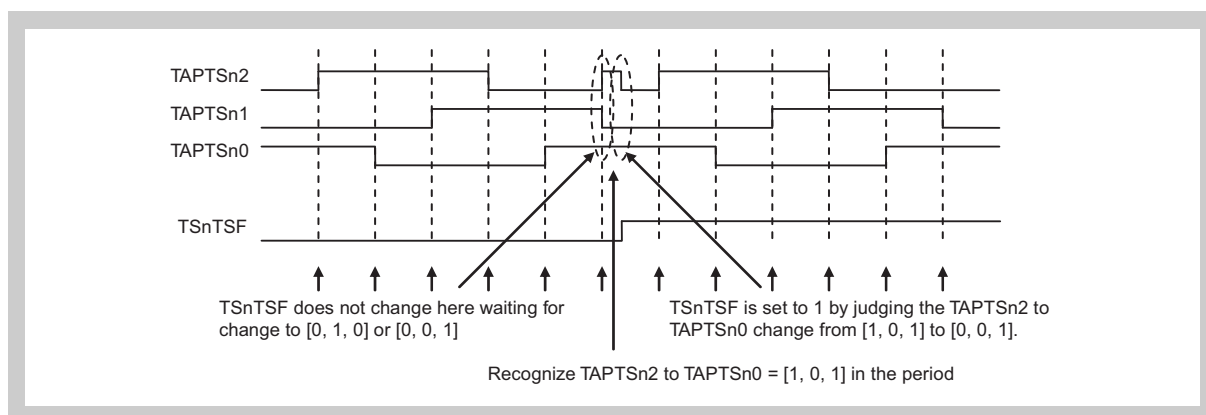
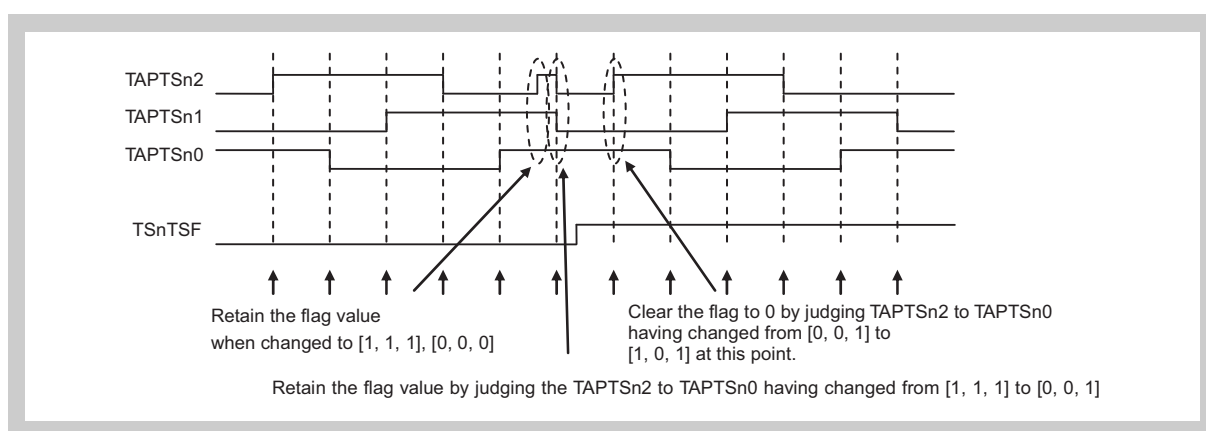


Figure 17-17 Pattern order detection flag operation example (in normal case)

- In exceptions occurrence



**Figure 17-18** When inputs to pins TAPTSn2 to TAPTSn0 are changed by 2 bits



**Figure 17-19** When inputs to pins TAPTSn2 to TAPTSn0 are changed to [0, 0, 0] or [1, 1, 1].

[Operation mode]

High-accuracy T-PWM mode<sup>Note</sup>, PWM mode with dead time<sup>Note</sup>, 120° excitation mode, and special 120° excitation mode

**Note** Selectable only when the software output control function (TSnSOC of the TSnOPT4 register = 1) or 180° excitation function (TSnADC of the TSnOPT5 register is 1) is used.



## 17.6.6 Pattern error detection flag (TSnPEF)

[Name]

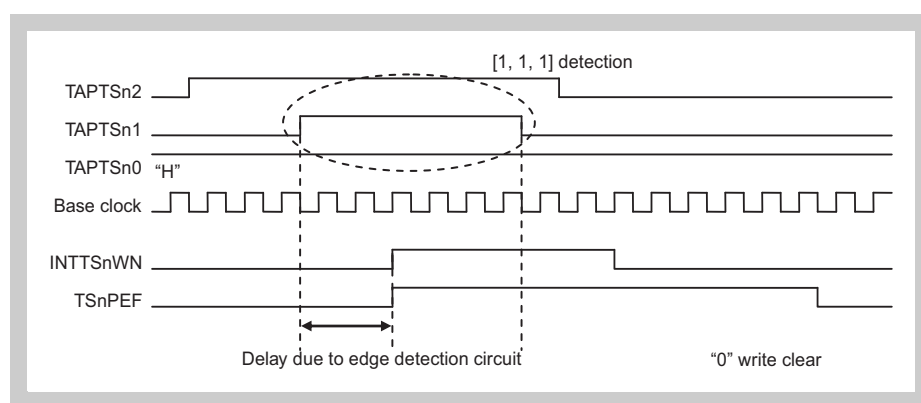
Pattern error detection flag (TSnPEF) - TSnOPT6 register

[Description]

TSnPEF can detect [0, 0, 0] or [1, 1, 1] input to the TAPTSn2 to TAPTSn0 pins.

TSnPEF is set to 1 when the pin level of the TAPTSn2 to TAPTSn0 pins is [0, 0, 0] or [1, 1, 1] level, and cleared when 0 is written to the TSnPEF flag, or while timer Sn operation is stopped (TSnCE of the TSnCTL0 register is set to 0).

[Example]



**Figure 17-20** Pattern error detection flag operation example  
(TAPTSn2 to TAPTSn0 = [1, 1, 1])

[Operation mode]

High-accuracy T-PWM mode<sup>Note</sup>, PWM mode with dead time<sup>Note</sup>, 120° excitation mode and special 120° excitation mode

**Note** Selectable only when the software output control function (TSnSOC of the TSnOPT4 register is 1) or 180° excitation function (TSnADC of the TSnOPT5 register is 1) is used.

### 17.6.7 Pattern reverse detection flag (TSnPRF)

[Name]

Pattern reverse detection flag (TSnPRF) - TSnOPT6 register

[Description]

TSnPRF can detect reversal of change order of the TAPTSn2 to TAPTSn0 pins.

TSnPRF switches at the change timing of pattern order detection flag (TSnTSF). The TSnPRF flag is valid with the 2nd or later change of the TAPTSn2 to TAPTSn0 pins immediately after TSnCE of the TSnCTL0 register is set to 1.

[Example]

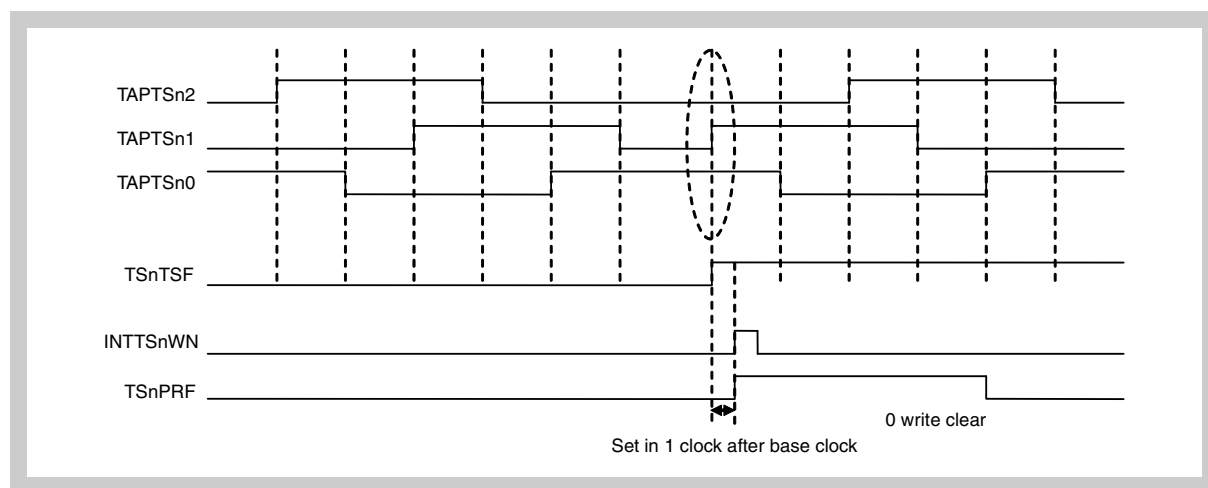


Figure 17-21 Pattern reverse rotation detection flag operation example

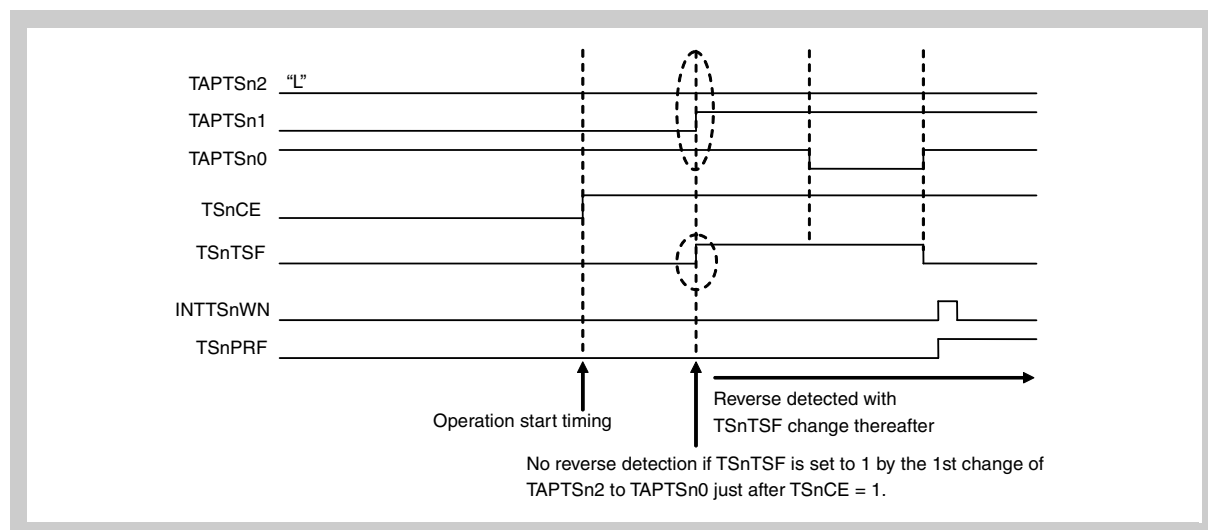


Figure 17-22 Operation example immediately after TSnCE of TSnCTL0 register is set to 1

---

[Operation mode]

High-accuracy T-PWM mode<sup>Note</sup>, PWM mode with dead time<sup>Note</sup>, 120° excitation mode, and special 120° excitation mode

**Note** Selectable only when the software output control function (TSnSOC of the TSnOPT4 register is 1) or 180° excitation function (TSnADC of the TSnOPT5 register is 1) is used.

### 17.6.8 TAPTSn2 to TAPTSn0 pin abnormal toggle detection flag (TSnPTF)

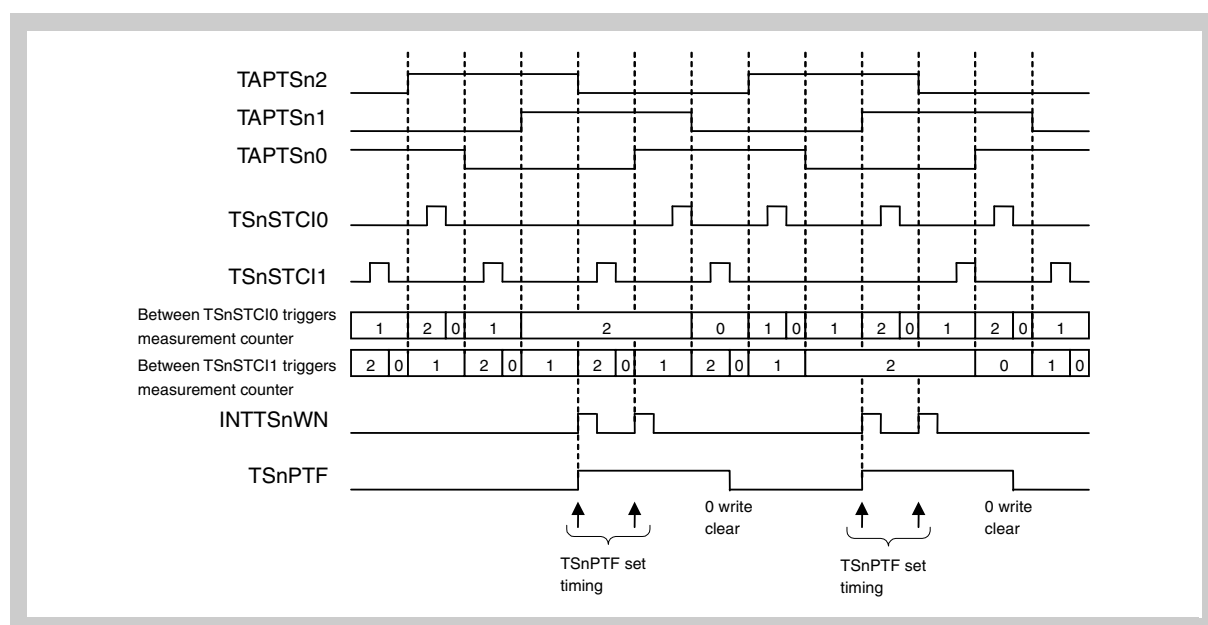
[Name]

TAPTSn2 to TAPTSn0 pins abnormal toggle detection flag (TSnPTF) -  
TSnOPT6 register

[Description]

TSnPTF can detect that three or more changes of the TAPTSn2 to TAPTSn0 pins are input between the TSnSTCI0 signal triggers, or between the TSnSTCI1 signal triggers. If the third trigger of the TSnSTCI0 and TSnSTCI1 signals and change of the TAPTSn2 to TAPTSn0 pins are generated simultaneously, the TSnPTF flag is set to 1 and a warning interrupt (INTTSnWN) occurs.

[Example]



**Figure 17-23** TAPTSn2 to TAPTSn0 pins abnormal toggle detection flag operation example

[Operation mode]

High-accuracy T-PWM mode<sup>Note</sup>, PWM mode with dead time<sup>Note</sup>, 120°  
excitation mode and special 120° excitation mode

**Note** Selectable only when the software output control function (TSnSOC of the TSnOPT4 register is 1) or 180° excitation function (TSnADC of the TSnOPT5 register is 1) is used.

- Caution**
1. The TSnPOT flag is valid only when TSnPOT of the TSnOPT5 register = 1, TSnSOC of the TSnOPT4 register = 0, and TSnPOTC1 of the TSnOPT7 register = 1.
  2. When TSnPOT = 1, TSnPOTC0 = 1 and TSnPOTC1 = 1, abnormal toggle detection leads automatic switch to the pattern switch system (TSnPOT = 0) by output switch control of TOSn1 to TOSn6 pins.
- 

**Note** If the third trigger of the TSnSTCI0 and TSnSTCI1 signals and change of the TAPTSn2 to TAPTSn0 pins are generated simultaneously, a warning interrupt (INTTSnWN) occurs and the TSnPOT flag is set.

### 17.6.9 TSnSTCI0, TSnSTCI1 signal simultaneous trigger detection flag (TSnTDF)

[Name]

TSnSTCI0, TSnSTCI1 signal simultaneous trigger detection flag (TSnTDF) - TSnOPT6 register

[Description]

TSnTDF detects that the TSnSTCI0 and TSnSTCI1 signals become triggers simultaneously.

[Example]

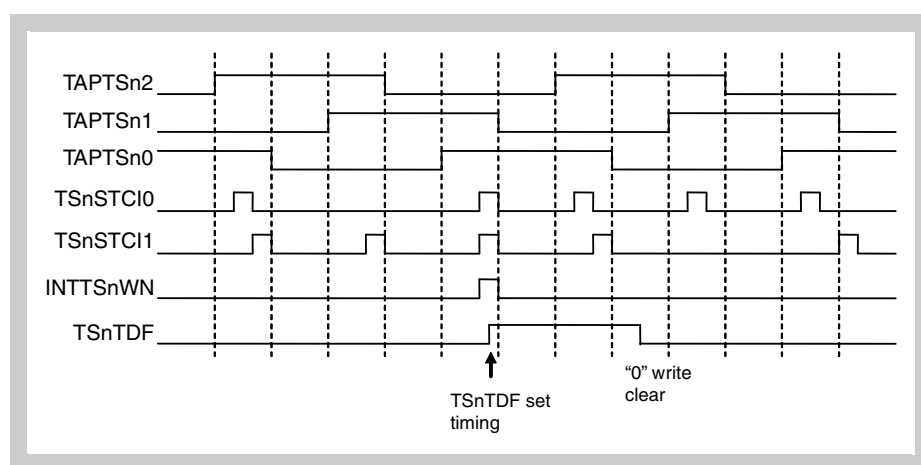


Figure 17-24 TSnSTCI0, TSnSTCI1 signal simultaneous trigger detection flag operation example

[Operation mode]

High-accuracy T-PWM mode<sup>Note</sup>, PWM mode with dead time<sup>Note</sup>, 120° excitation mode and special 120° excitation mode

**Note** Selectable only when the software output control function (TSnSOC of the TSnOPT4 register = 1) or 180° excitation function (TSnADC of the TSnOPT5 register = 1) is used

**Caution** The TSnTDF flag is valid only when TSnPOT of the TSnOPT5 register is 1, TSnSOC of the TSnOPT4 register is 0, and TSnPTC1 of the TSnOPT7 register is 1.

## 17.6.10 Pattern phase difference detection flag (TSnPPF)

[Name]

Pattern phase difference detection flag (TSnPPF) - TSnOPT6 register

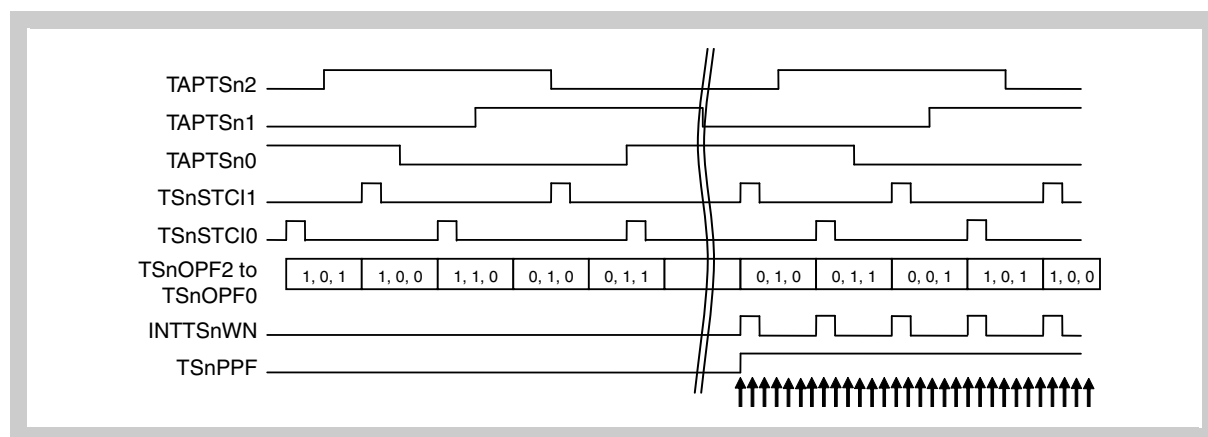
[Description]

TSnPPF can detect the phase difference between input pattern (TAPTSn2 to TAPTSn0 pins) and output pattern (TSnOPF2 to TSnOPF0 bits of TSnOPT5 register) when the semi-automatic driving system is used. The pattern phase difference is detected when the TSnSTCI0 or TSnSTCI1 signal is input, and the TSnPPF bit is then set. The TSnPPF bit remains 1 until it is cleared by software. When a phase difference is detected, the TSnPPF bit is set for each base clock ( $f_{TMSn}$ ) of timer Sn. Clear the TSnPPF bit (0) at a timing at which no phase differences may occur.

**Table 17-31 Normal input/output pattern**

TAPTSn2 to TAPTSn0 (Input)	"1, 0, 1"	"1, 0, 0"	"1, 1, 0"	"0, 1, 0"	"0, 1, 1"	"0, 0, 1"
TSnOPF2 to TSnOPF0 (Output)	"0, 0, 1"	"1, 0, 1"	"1, 0, 0"	"1, 1, 0"	"0, 1, 0"	"0, 1, 1"
	"1, 0, 1"	"1, 0, 0"	"1, 1, 0"	"0, 1, 0"	"0, 1, 1"	"0, 0, 1"
	"1, 0, 0"	"1, 1, 0"	"0, 1, 0"	"0, 1, 1"	"0, 0, 1"	"1, 0, 1"

[Example]



**Figure 17-25 Pattern phase difference detection flag operation example**

[Operation mode]

High-accuracy T-PWM mode<sup>Note</sup>, PWM mode with dead time<sup>Note</sup>,  
120° excitation mode and special 120° excitation mode

**Note** Selectable only when the software output control function (TSnSOC of TSnOPT4 register is 1) or 180° excitation function (TSnADC of TSnOPT5 register is 1) is used

**Caution** The TSnPPF flag is valid only when TSnPOT of the TSnOPT5 register is 1, TSnSOC of the TSnOPT4 register is 0, and TSnPPC of the TSnOPT7 register is 1.

### 17.6.11 Overflow flag (TSnOVF)

[Name]

Overflow flag (TSnOVF) - TSnOPT0 register

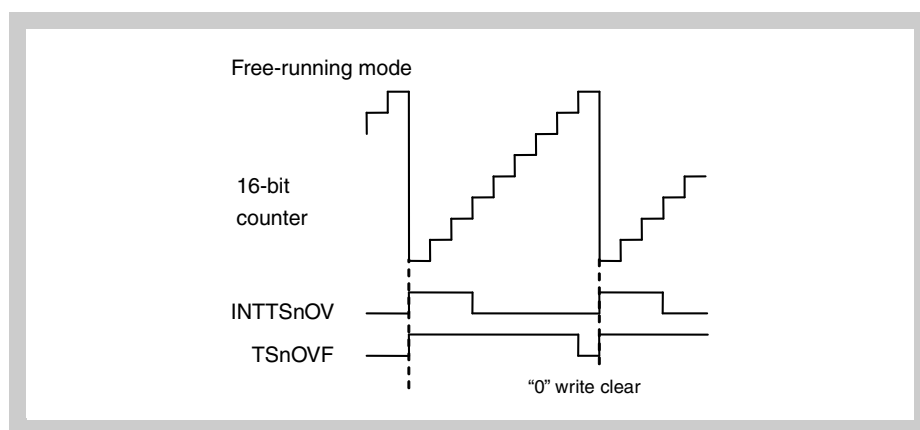
[Description]

TSnOVF is set to 1 when the 16-bit counter value overflows from FFFFH to 0000H.

This flag is cleared to 0 by writing 0 to it or the operation is stopped (TSnCE of the TSnCTL0 register = 0). If TSnOVF is set, an overflow interrupt (INTTSnOV) is generated simultaneously.

[Example]

Figure 17-26 Overflow flag operation example



[Operation mode]

Free-running mode and high-accuracy T-PWM mode<sup>Note</sup>

**Note** Only when the TSnDTC0, TSnDTC1 register values are incorrectly set



---

### 17.6.12 Timer output pattern flags (TSnOPF2 to TSnOPF0)

[Name]

Timer output pattern flags (TSnOPF2 to TSnOPF0) - TSnOPT5 register

[Description]

TSnOPF2 to TSnOPF0 are flags that indicate timer output patterns.  
Refer to “*Timer output pattern in each mode and function*” on page 671 for details.

[Operation mode]

High-accuracy T-PWM mode<sup>Note</sup>, PWM mode with dead time<sup>Note</sup>, 120° excitation mode and special 120° excitation mode

**Note** Selectable only when the 180° excitation function (TSnADC of TSnOPT5 register = 1) is used

### 17.6.13 Pattern switch detection flag (TSnESG)

[Name]

Pattern switch detection flag (TSnESG) - Internal signal

[Description]

TSnESG is toggled at the change timing of input pattern (TAPTSn2 to TAPTSn0).

TAPTSn2 to TAPTSn0 After Change									
		[0, 0, 0]	[1, 1, 1]	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]
TAPTSn2 to TAPTSn0 at present	[0, 0, 0]	-	-	-	-	-	-	-	-
	[1, 1, 1]	-	-	-	-	-	-	-	-
	[1, 0, 1]	-	-	-	Toggle	-	-	-	Toggle
	[1, 0, 0]	-	-	Toggle	-	Toggle	-	-	-
	[1, 1, 0]	-	-	-	Toggle	-	Toggle	-	-
	[0, 1, 0]	-	-	-	-	Toggle	-	Toggle	-
	[0, 1, 1]	-	-	-	-	-	Toggle	-	Toggle
	[0, 0, 1]	-	-	Toggle	-	-	-	Toggle	-

[Example]

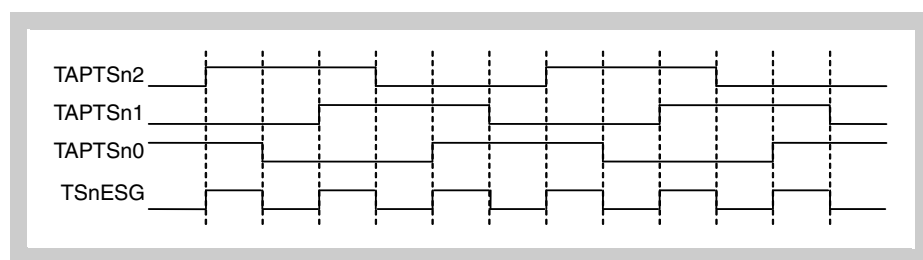


Figure 17-27 Pattern switch detection flag operation example

[Operation mode]

High-accuracy T-PWM mode<sup>Note</sup>, PWM mode with dead time<sup>Note</sup>, 120° excitation mode and special 120° excitation mode

**Note** Selectable only when the 180° excitation function (TSnADC of TSnOPT5 register = 1) is used

## 17.7 Interrupt Thinning out Function

The operations related to the interrupt thinning out function are as follows.

- The interrupts to be culled are peak interrupt (INTTSnCD0) and trough interrupt (INTTSnOD).
- The TSnICE bit of the TSnOPT1 register is used to enable the output of INTTSnCD0 interrupts and specify the interrupt thinning out count.
- The TSnIOE bit of the TSnOPT1 register is used to enable the output of INTTSnOD interrupts and specify the interrupt thinning out count.
- Whether or not reload thinning out is to be performed can be specified with the TSnRDE bit of the TSnOPT2 register.

If the use of reload thinning out is specified, reloading is executed at the same time as an interrupt output after the thinning out.

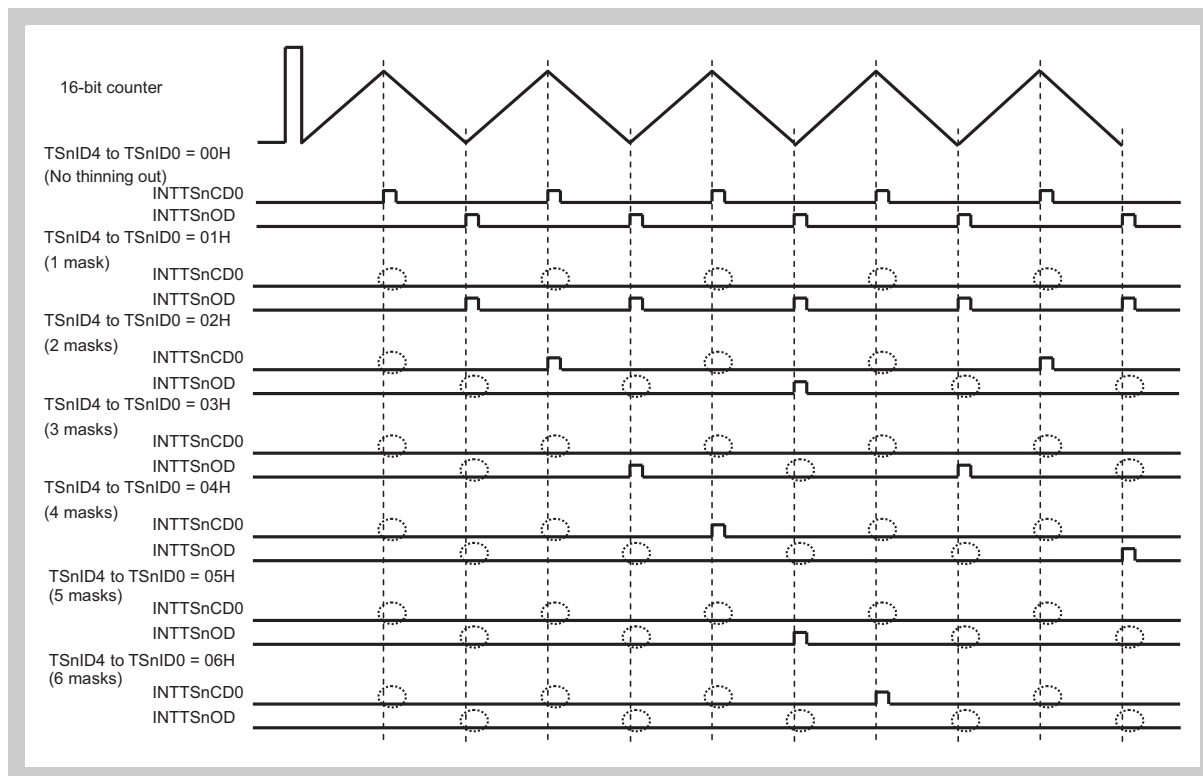
If the use of reload thinning out is not specified, reloading is executed at the reload timing after a write access is performed on the TSnCCR1 register.

- 
- Caution**
1. The interrupt thinning out function is enabled in the PWM mode, triangular-wave PWM mode, high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, or special pattern output mode.
  2. If a write access is performed on the TSnOPT1 or TSnOPT1L register by using the anytime rewrite mode or reload mode (batch rewrite function), the internal thinning out counter is cleared. Therefore, the interrupt interval may be extended temporarily when the interrupt thinning out function is used. To avoid this, change the interrupt thinning out count so that the values are reloaded to the counter in synchronization with the interrupt thinning out (TSnCMS of the TSnOPT0 register = 0, or TSnRDE of the TSnOPT1 register = 1).
-

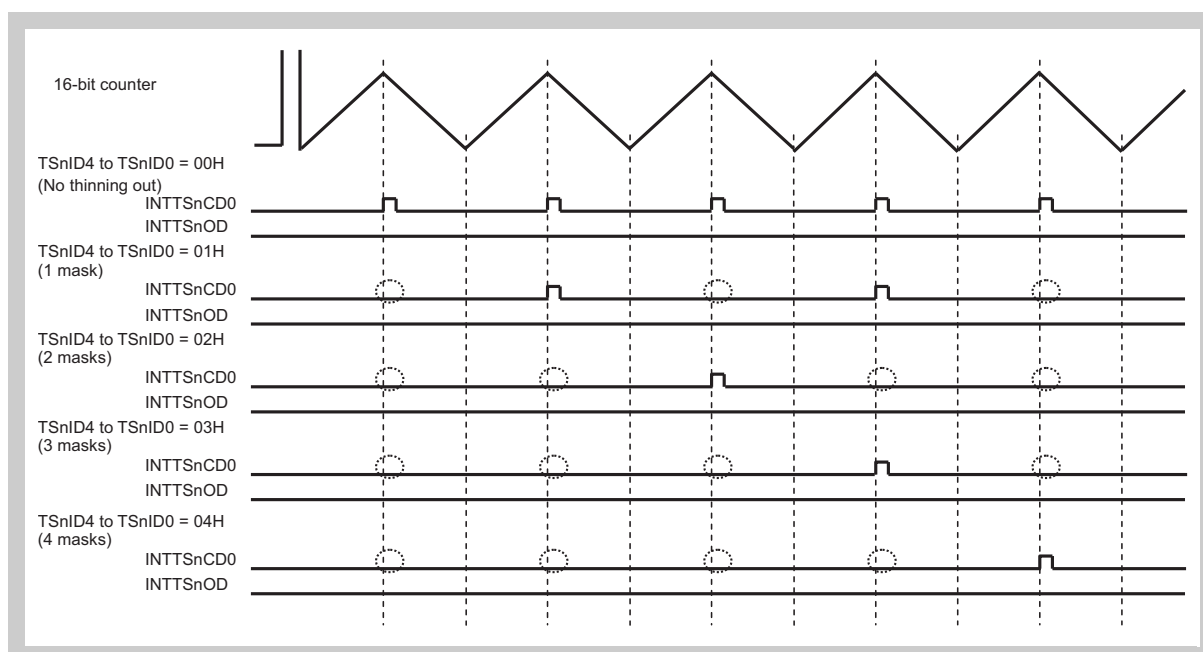
### 17.7.1 Interrupt thinning out function operation

rec

4. Interrupt thinning out operation when TSnICE and TSnIOE of the TSnOPT1 register are 1 (when a peak/trough interrupt occurs in high-accuracy T-PWM mode)

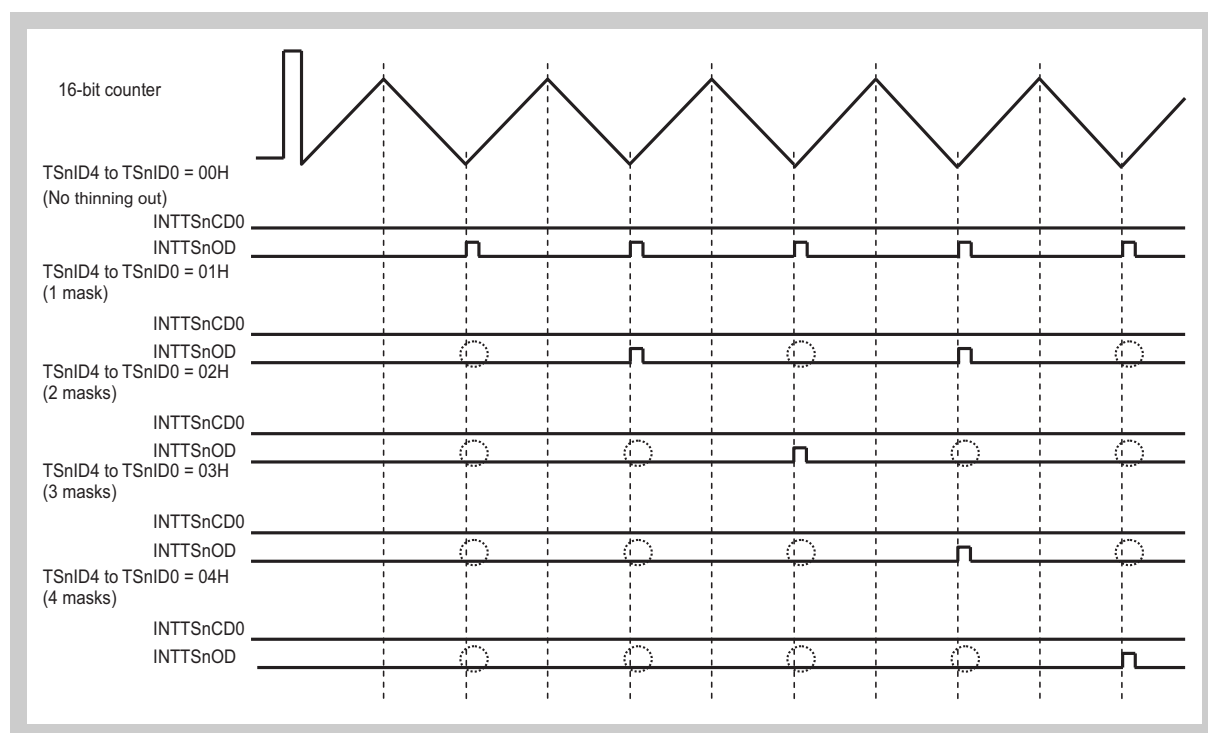


5. Interrupt thinning out operation when the TSnICE and TSnIOE of the TSnOPT1 register are 1 (when only a peak interrupt occurs in high-accuracy T-PWM mode)



6. Interrupt thinning out operation when the TSnOPT1 register's TSnICE is 0

and TSnIOE is 1 (when only a trough interrupt occurs in high-accuracy T-PWM mode)

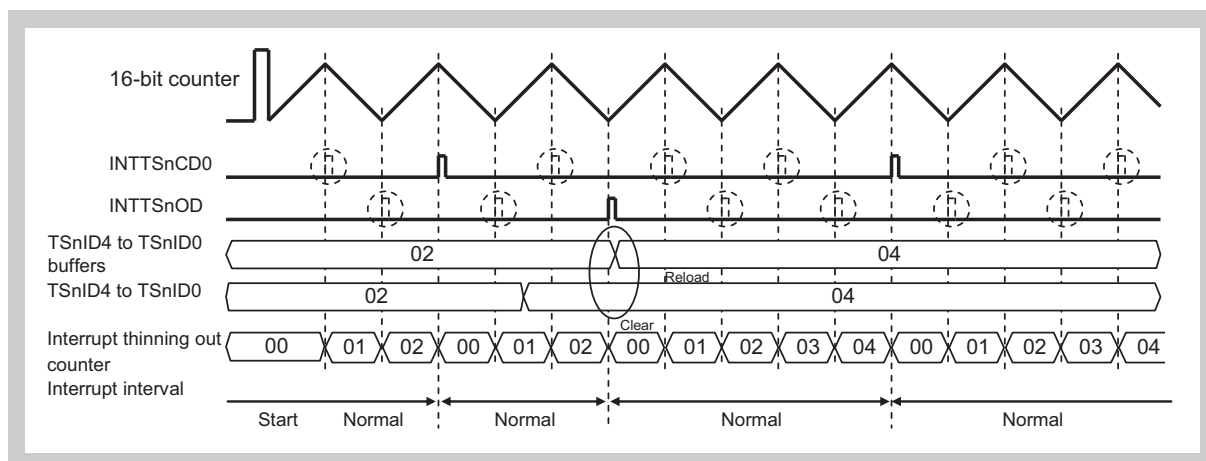


### 17.7.2 Operation example when peak and trough interrupts occur alternatively (in high-accuracy T-PWM mode)

#### (1) Register setting

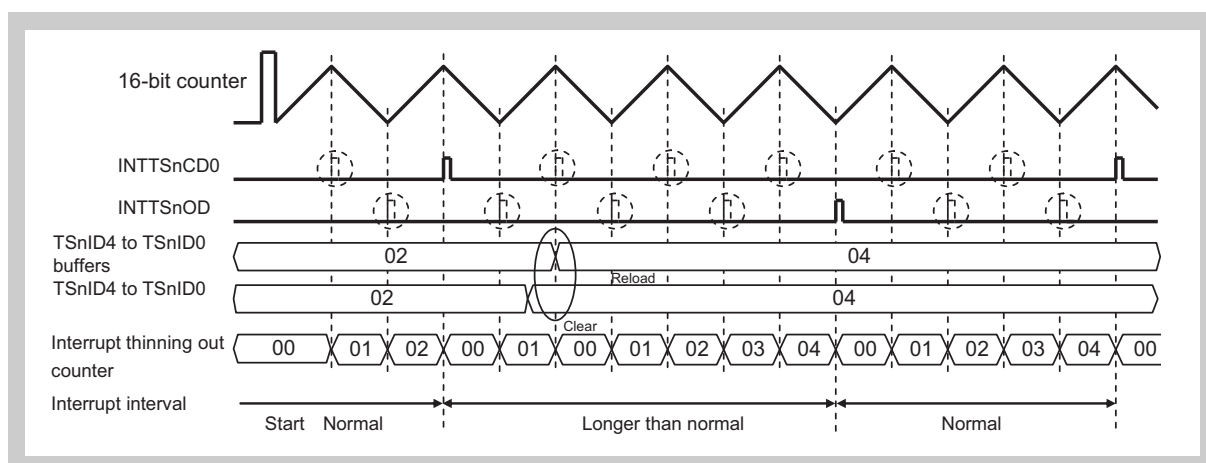
TSnICE = 1, TSnIOE = 1, TSnRBE = 1, and TSnRTE = 0 for the TSnOPT1 register

#### (2) Operation example



**Figure 17-28** When TSnCMS of TSnOPT0 Register = 0 and TSnRDE of TSnOPT1 Register = 1 (Reload Thinning out Function Selected) (Recommended Setting)

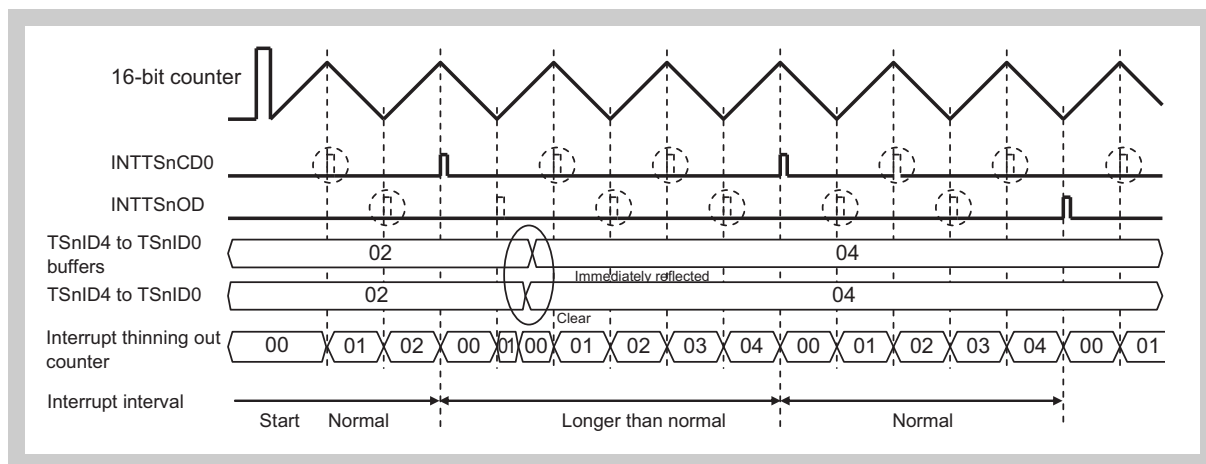
**Note** Values are reloaded at the thinning out interrupt output timing, and other reload timings are ignored.



**Figure 17-29** When TSnCMS of TSnOPT0 Register = 0 and TSnRDE of TSnOPT1 Register = 0 (Reload Function Selected)

**Note** Values are reloaded at the reload timing after rewriting.

**Caution** The interrupt interval may be extended.



**Figure 17-30** When TSnCMS of TSnOPT0 register = 1 and TSnRDE of TSnOPT1 register is x (in anytime rewrite mode)

**Note** The reload timing is ignored after rewriting, and the value is immediately reflected.  
The counter is cleared when the value is transferred to the buffer, not when the register is rewritten.

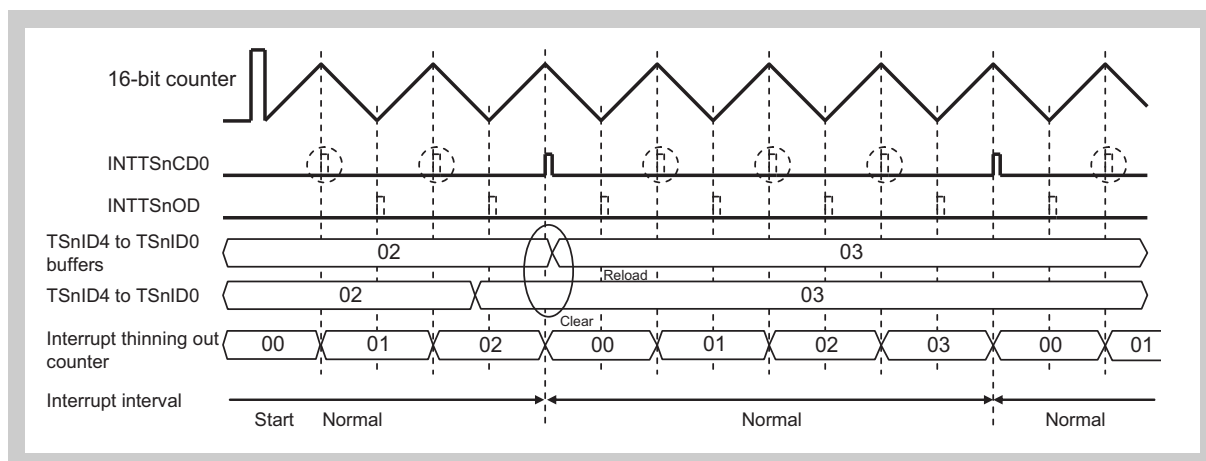
**Caution** The interrupt interval may be extended.

### 17.7.3 Operation example when only peak interrupt occurs (in high-accuracy T-PWM mode)

#### (1) Register setting

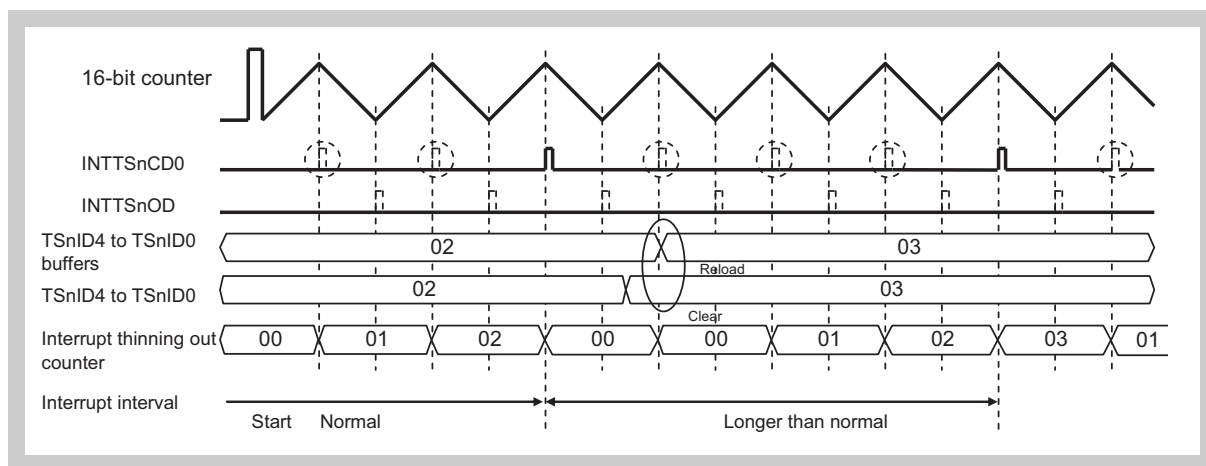
TSnICE = 1, TSnIOE = 0, TSnRBE = 0, and TSnRTE = 1 for the TSnOPT1 register

#### (2) Operation example



**Figure 17-31** When TSnCMS of TSnOPT0 register = 0 and TSnRDE of TSnOPT1 register = 1 (reload function selected) (recommended setting)

**Note** Values are reloaded at the thinning out interrupt output timing, and other reload timings are ignored.

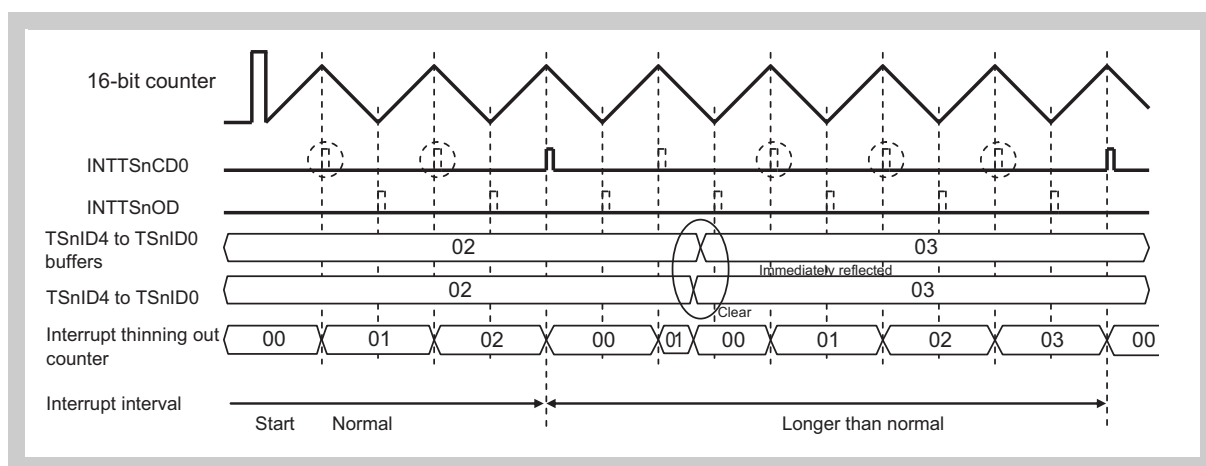


**Figure 17-32** When TSnCMS of TSnOPT0 register = 0 and TSnRDE of TSnOPT1 register = 0 (reload function selected)

**Note** Values are reloaded at the reload timing after rewriting.

**Caution** The interrupt interval may be extended.





**Figure 17-33** When TSnCMS of TSnOPT0 register = 1 and TSnRDE of TSnOPT1 register = x (in anytime rewrite mode)

- Note**
1. The reload timing is ignored after rewriting, and then the value is immediately reflected.
  2. The counter is cleared when the value is transferred to the buffer, not when the register is rewritten.

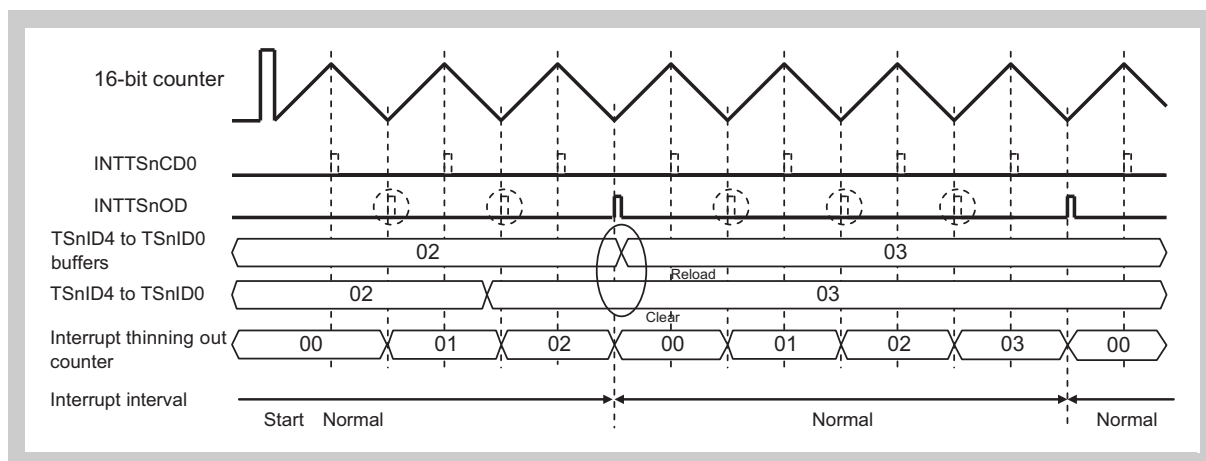
**Caution** The interrupt interval may be extended.

### 17.7.4 Operation example when only trough interrupt occurs (in high-accuracy T-PWM mode)

#### (1) Register setting

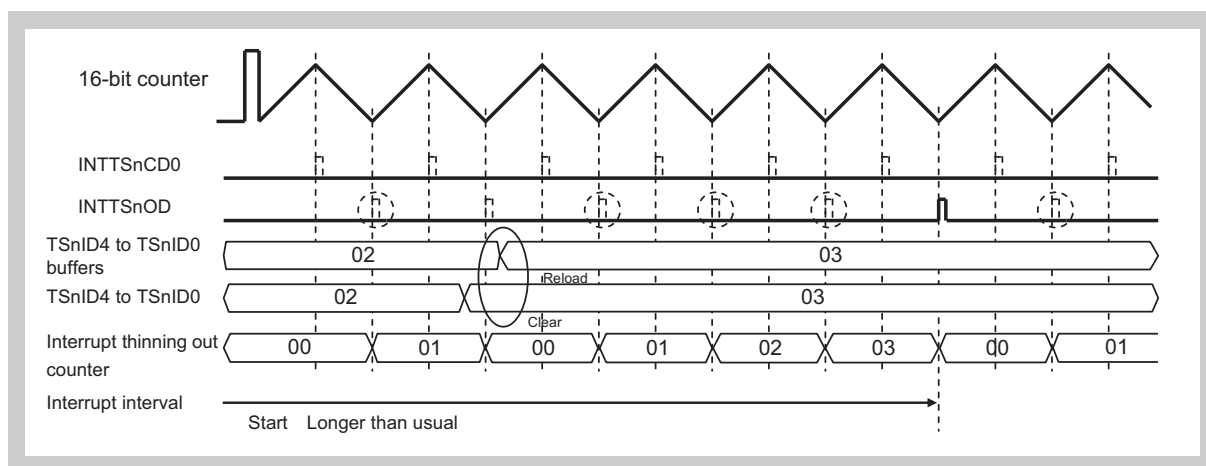
TSnICE = 0, TSnIOE = 1, TSnRBE = 1, and TSnRTE = 0 in the TSnOPT1 register

#### (2) Operation example



**Figure 17-34** When TSnCMS of TSnOPT0 register = 0 and TSnRDE of TSnOPT1 register = 1 (reload thinning out function) (recommended setting)

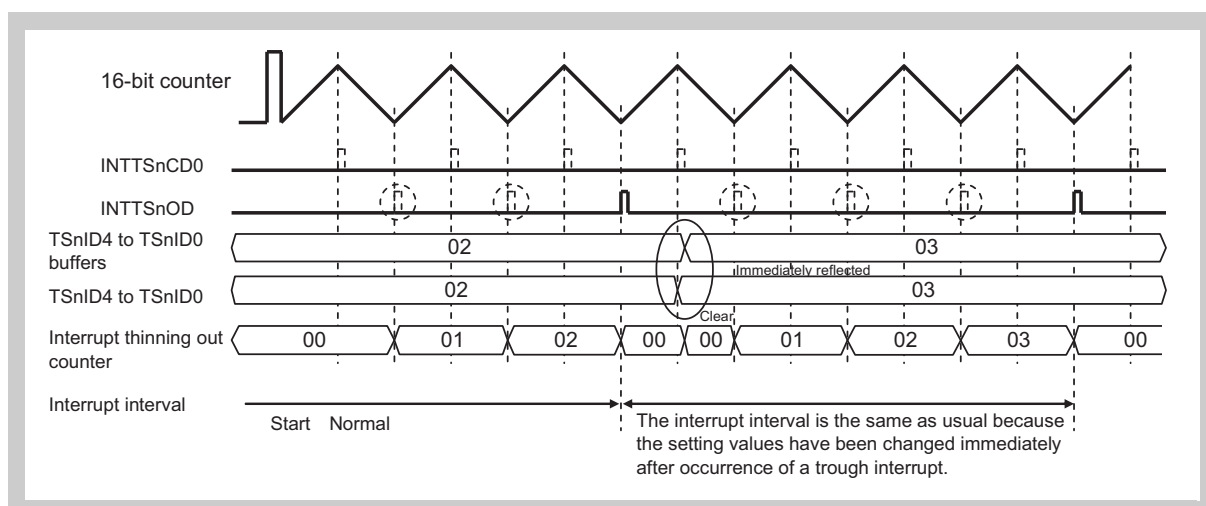
**Note** Values are reloaded at the thinning out interrupt output timing and other reload timings are ignored.



**Figure 17-35** TSnOPT0 register TSnCMS = 0 and TSnOPT1 register TSnRDE = 0 (reload function)

**Note** Values are reloaded at the reload timing after rewriting.

**Caution** The interrupt interval may be extended.



**Figure 17-36** When TSnCMS of TSnOPT0 register = 1 and TSnRDE of TSnOPT1 register = x (in anytime rewrite mode)

- Note**
1. The reload timing is ignored and the value is immediately reflected after rewriting.
  2. The counter is cleared when the value is transferred to the buffer, not when the register is rewritten.

**Caution** The interrupt interval may be extended.

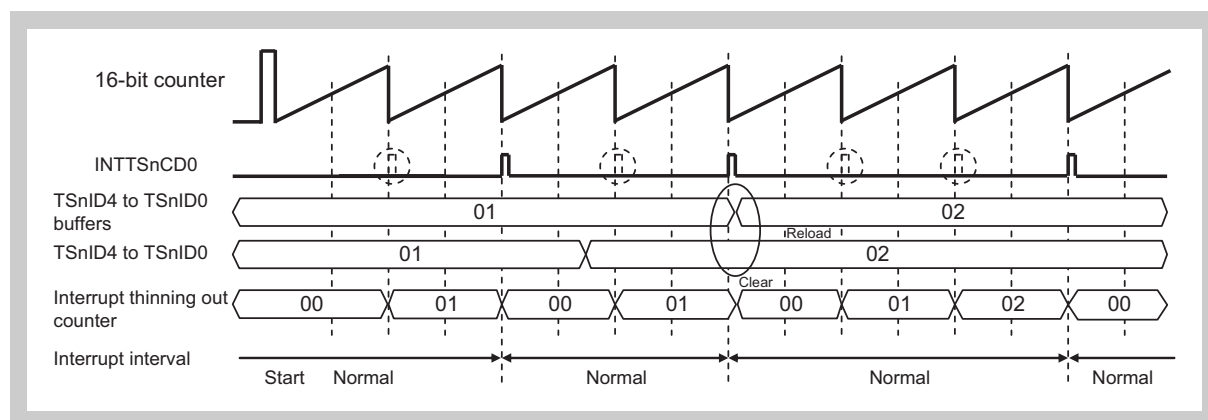
### 17.7.5 Operation example when only peak interrupt occurs (in PWM mode)

The operations related to the interrupt thinning out function in the PWM mode are as follows.

- The interrupt to be culled is the peak interrupt (INTTSnCD0). This interrupt is generated upon a match between the TSnCCR0 register and 16-bit counter during saw tooth wave operation.
- The TSnICE bit of the TSnOPT1 register is used to enable output of the INTTSnCD0 interrupt and specify the interrupt thinning out count.
- TSnIOE bit setting in the TSnOPT1 register is invalid. In this case, generation of INTTSnOD interrupts is disabled.
- Whether or not the reload thinning out is to be performed can be specified with the TSnRDE bit of the TSnOPT1 register.  
If the use of the reload thinning out is specified, reloading is executed at the same time as an interrupt output after the thinning out.  
If the use of the reload thinning out is not specified, reloading is executed at the peak timing after a write access is performed on the TSnCCR1 register.

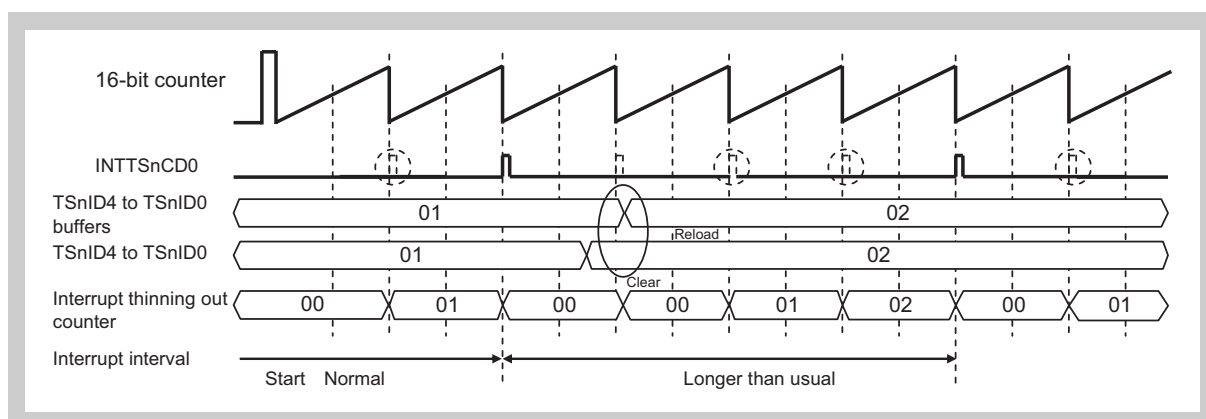
**Caution** If a write access is performed on the TSnOPT1 or TSnOPT1L register by using the anytime rewrite mode or reload mode (batch rewrite function), the internal thinning out counter is cleared. Therefore, the interrupt interval may be extended temporarily when the interrupt thinning out function is used. To avoid this, change the interrupt thinning out count so that the values are reloaded to the counter in synchronization with the interrupt thinning out (TSnCMS of the TSnOPT0 register = 0, or TSnRDE of the TSnOPT1 register = 1).

#### (1) Operation example



**Figure 17-37** When TSnCMS of TSnOPT0 register = 0 and TSnRDE of TSnOPT1 register = 1 (reload thinning out function) (recommended setting)

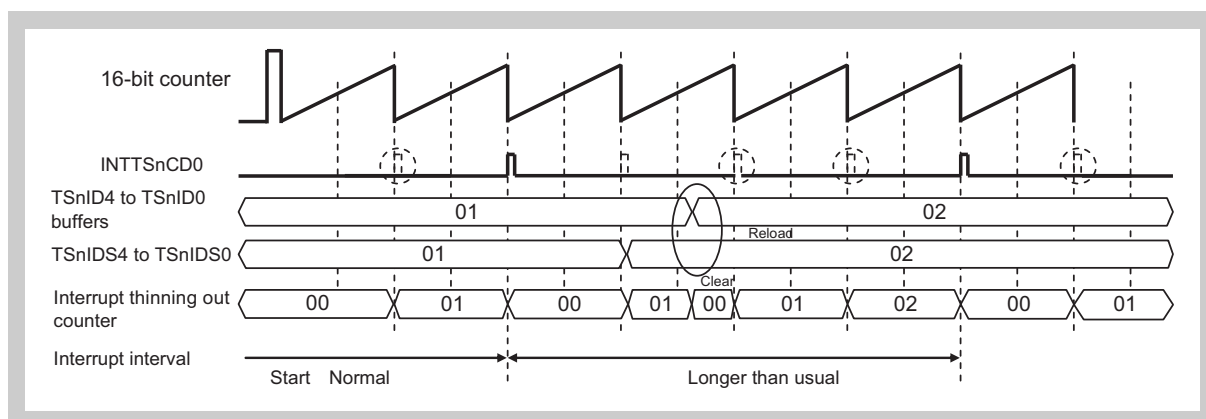
**Note** Values are reloaded at the thinning out interrupt output timing and other reload timings are ignored.



**Figure 17-38** When TSnCMS of TSnOPT0 register = 0 and TSnRDE of TSnOPT1 register = 0 (reload function)

**Note** Values are reloaded at the reload timing after rewriting.

**Caution** The interrupt interval may be extended.



**Figure 17-39** When TSnCMS of TSnOPT0 register = 1 and TSnRDE of TSnOPT1 register = x (in anytime rewrite mode)

**Note** The reload timing is ignored and the value is immediately reflected after rewriting.  
The counter is cleared when the value is transferred to the buffer, not when the register is rewritten.

**Caution** The interrupt interval may be extended.

## 17.8 A/D Conversion Trigger Function

The following describes the A/D conversion trigger output in PWM mode, triangular-wave PWM mode, high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode and special pattern output mode.

In the high-accuracy T-PWM mode, PWM mode with dead time, special 120° excitation mode, or special pattern output mode, the TSnCCR4 and TSnCCR5 registers do not affect the timer output as compare operation but are used for controlling match interrupts and A/D conversion triggers.

In the PWM mode, triangular-wave PWM mode, or 120° excitation mode, note that the setting of the TSnCCR4 and TSnCCR5 registers may affect the timer output.

Refer to “A/D conversion trigger, peak interrupt, trough interrupt in each mode” on page 711 for details on A/D conversion triggers that can be output in each mode.

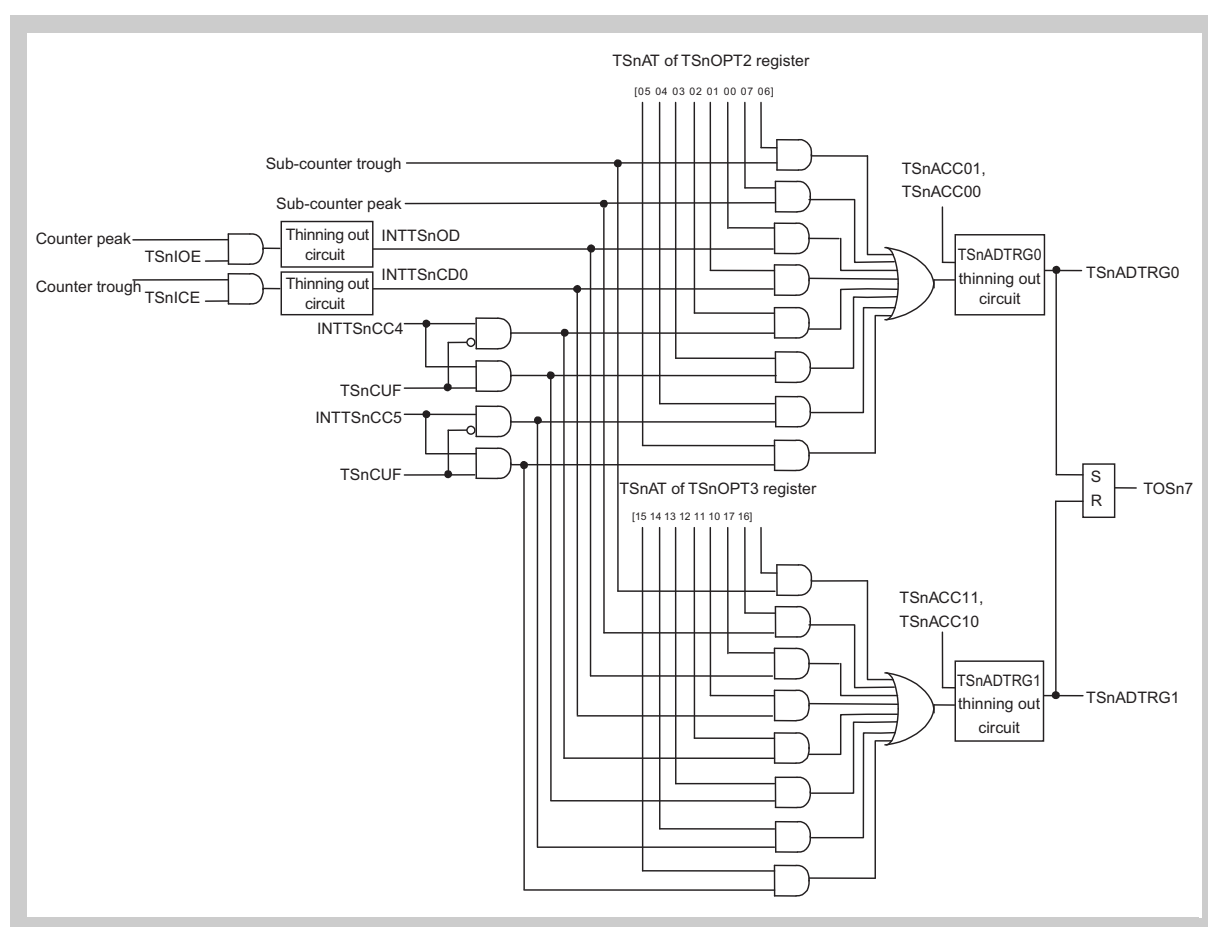


Figure 17-40 A/D conversion trigger output control circuit

The interrupt signals for compare match interrupts (INTTSnCC5, INTTSnCC4), peak interrupt (INTTSnCD0), trough interrupt (INTTSnOD), the 16-bit sub-counter peak timing, and the 16-bit sub-counter trough timing can be selected to provide OR output as shown in Figure 17-40 on page 746.

Timer Sn has two channels of the same A/D trigger control circuits. Each can be controlled independently.

The A/D trigger thinning out functions are equipped to enable setting with 1/2, 1/4, or 1/8 thinning out ratio, or without thinning out.

### 17.8.1 A/D conversion trigger operation

Timer Sn has a function that generates an A/D conversion start trigger (TSnADTRG0 or TSnADTRG1 signal) by arbitrarily selecting eight trigger sources.

The following eight trigger sources can be selected with the TSnAT07 to TSnAT00 bits of the TSnOPT2 register or the TSnAT17 to TSnAT10 bits of the TSnOPT3 register.

The following describes TSnADTRG0/TSnADTRG1 signal control using control bits TSnAT07 to TSnAT00/TSnAT17 to TSnAT10.

#### (1) Output control of TSnADTRG0/TSnADTRG1 (TSnOPT2 and TSnOPT3 registers)

<Trigger source>

- TSnAT00/TSnADT10 = 1  
An A/D conversion trigger is generated when a trough interrupt (INTTSnOD) occurs
- TSnAT01/TSnADT11 = 1  
An A/D conversion trigger is generated when a peak interrupt (INTTSnOD0) occurs.
- TSnAT02/TSnADT12 = 1  
An A/D conversion trigger generation is enabled when a compare match interrupt (INTTSnCC4) occurs during 16-bit counter count-up operation.
- TSnAT03/TSnADT13 = 1  
An A/D conversion trigger is enabled when a compare match interrupt (INTTSnCC4) occurs during 16-bit counter count-down operation.
- TSnAT04/TSnADT14 = 1  
An A/D conversion trigger generation is enabled when a compare match interrupt (INTTSnCC5) occurs 16-bit counter during count-up operation.
- TSnAT05/TSnADT15 = 1  
An A/D conversion trigger generation is enabled when a compare match interrupt (INTTSnCC5) occurs during 16-bit counter count-down operation.
- TSnAT06/TSnADT16 = 1  
An A/D conversion trigger generation is enabled at a peak of the 16-bit sub-counter (when the timing is switched from up to down).
- TSnAT07/TSnADT17 = 1  
A/D conversion trigger generation is enabled at a trough of the 16-bit sub-counter (when the timing is switched from down to up).

<Thinning out setting>

- TSnACC01, TSnACC00 bits/TSnACC11, TSnACC10 bits  
TSnADTRG0/TSnADTRG1 signal thinning out setting

A/D conversion start trigger signals are undergo ORing, and then culled as specified with the TSnACC01, TSnACC00/TSnACC11, TSnACC10 bits, and are output to the TSnADTRG0 or TSnADTRG1 signal.

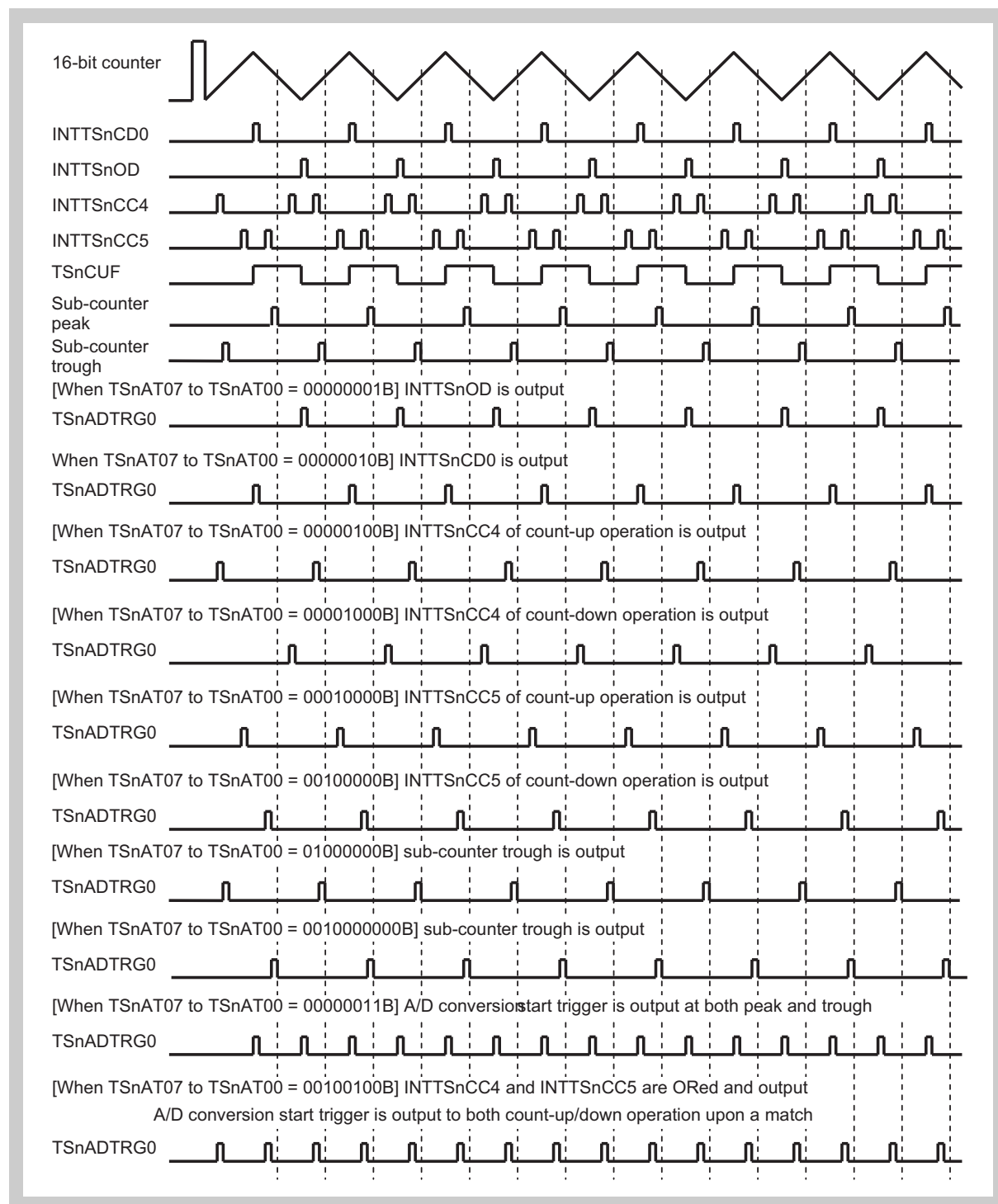
The peak interrupt (INTTSnOD) and trough interrupt (INTTSnOD) selected by the TSnAT00 and TSnAT01/TSnAT10, TSnAT11 bits are signals that are output after interrupt thinning out.

Therefore, output is provided at the timing when it is subject to interrupt thinning out control. If interrupt output (TSnICE and TSnIOE bits of TSnOPT1 register) is not enabled, neither can the A/D conversion start trigger be output.

The TSnACC01, TSnACC00, TSnAT07 to TSnAT00/TSnACC11, TSnACC10, TSnAT17 to TSnAT10 bits can be rewritten during timer operation.

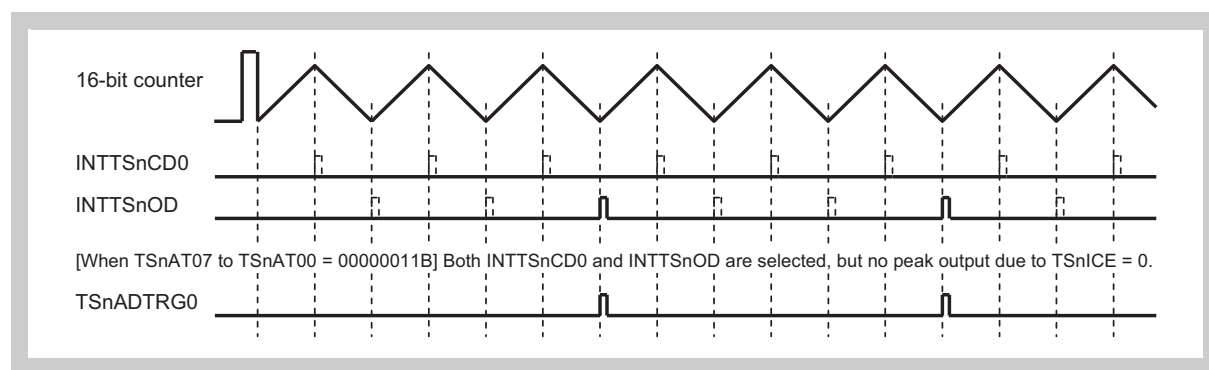
If a bit that sets an A/D conversion start trigger is rewritten during timer operation, the output status of the A/D conversion start trigger is reflected immediately.

These control bits can be written anytime regardless of the operation mode. When TSnACC01, TSnACC00/TSnACC11, TSnACC10 are written, the A/D conversion trigger thinning out counter is cleared and starts counting from 0. (The thinning out counter is also cleared when the same value is rewritten.)

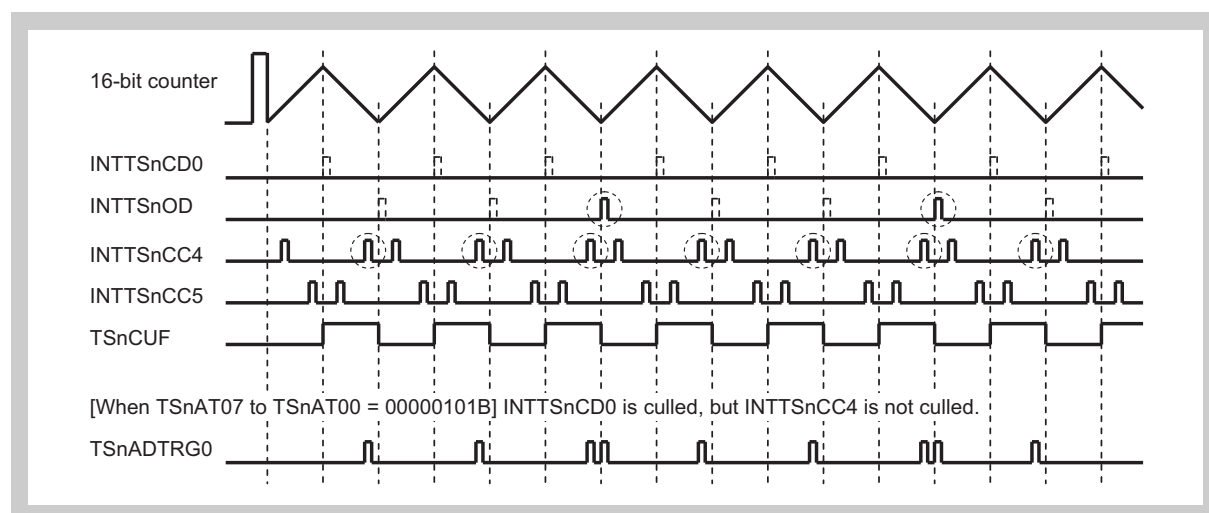


**Figure 17-41** When TSnICE = 1, TSnIOE = 1, TSnID4 to TSnID0 = 00H for TSnOPT1 register and TSnACC01, TSnACC00 bits = 00H for TSnOPT2 register (in high-accuracy T-PWM mode)





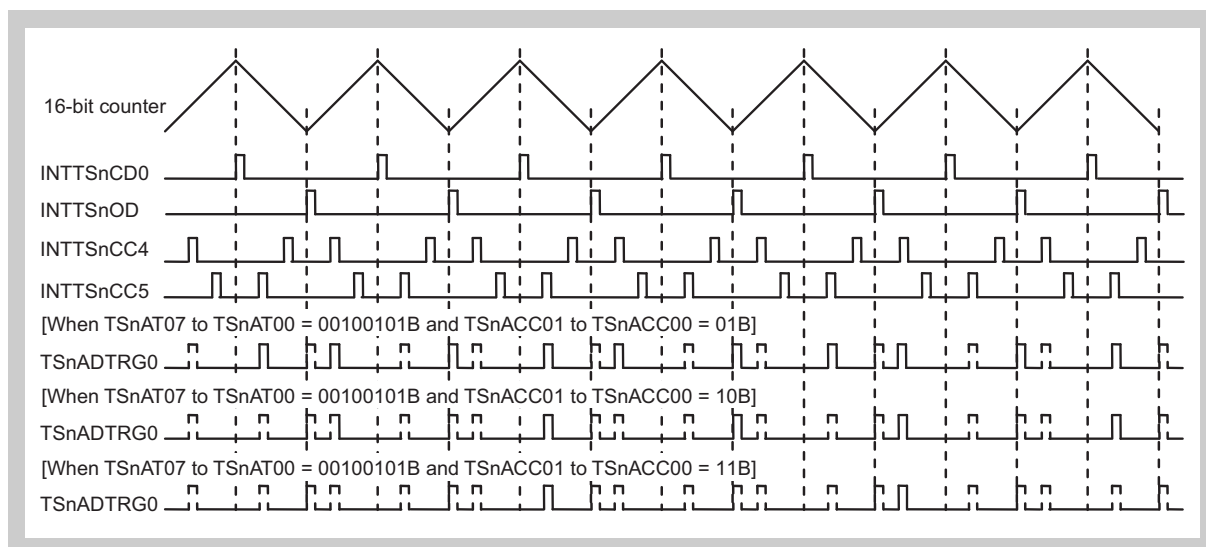
**Figure 17-42** When TSnICE = 0, TSnIOE = 1, TSnID4/TSnID0 = 02H for TSnOPT1 Register and TSnACC01, TSnACC00 Bits = 00H for TSnOPT2 Register (in High-Accuracy T-PWM Mode)



**Figure 17-43** When TSnICE = 0, TSnIOE = 1, TSnID4 to TSnID0 = 02H for TSnOPT1 register and TSnACC01, TSnACC00 Bits = 00H for TSnOPT2 register (in high-accuracy T-PWM mode)

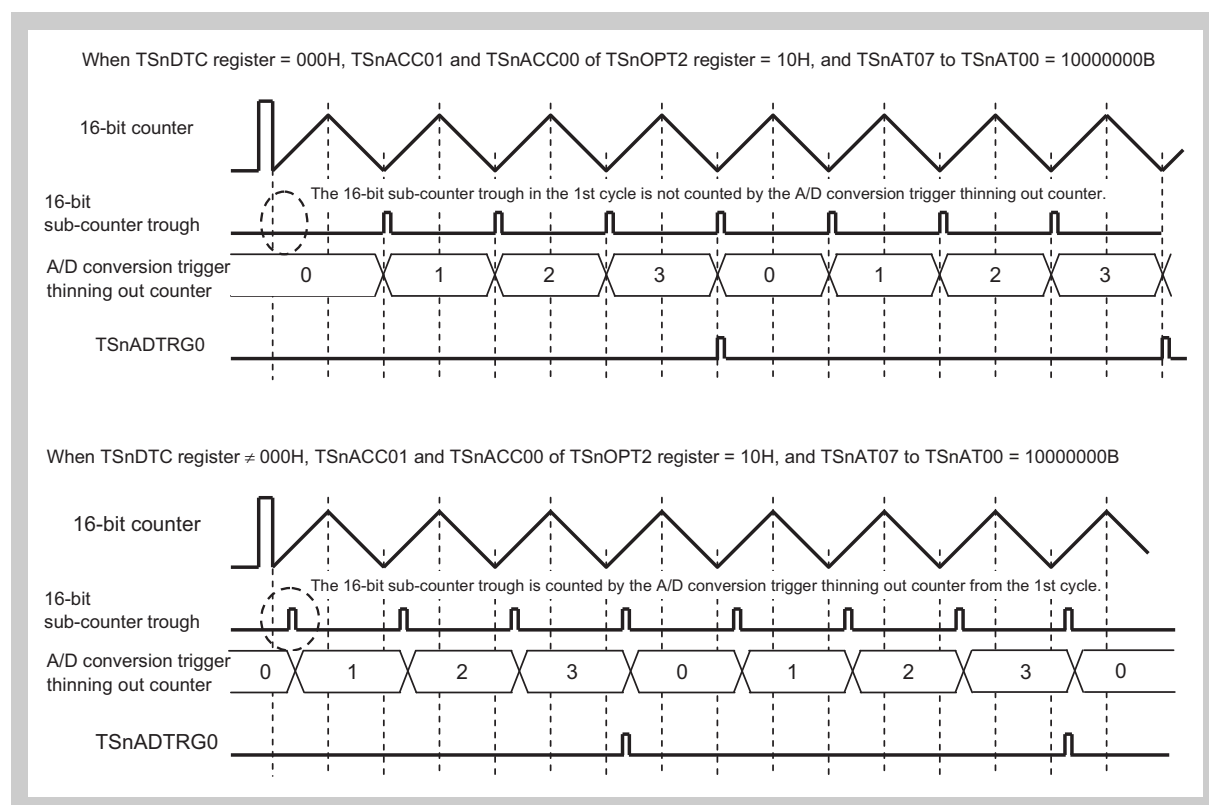
**(2) A/D conversion trigger thinning out function**

An operation example of the A/D conversion trigger thinning out function is shown below.



**Figure 17-44** Operation example of A/D conversion trigger thinning out function (except TSnDTC = 000H)

**Caution** The AD trigger thinning out counter operation differs according to whether the dead time register (TSnDTC0) is "000H" or "other than 000H". When the TSnDTC0 register is "000H", the 16-bit sub-counter trough of the first cycle from timer count start is not counted so that there is a difference in the AD trigger thinning out counter start timing as shown in *Figure 17-44* on page 750.



**Figure 17-45** Operation example of A/D Conversion trigger thinning out function (TSnDTC = 000H)

### (3) Cautions for A/D conversion trigger

- The count value of the A/D conversion trigger thinning out counter is incremented by 1 each time the same value is written (write access) to the TSnCCR4 register and TSnCCR5 register, and a valid A/D trigger is set under the same condition during count-up/down operation of the 16-bit counter. Output trigger is 1 pulse as well.
- The trough interrupt (INTTSnOD) does not occur in the PWM mode, PWM mode with dead time, 120° excitation mode, special 120° excitation mode, and special pattern output mode. Only the peak interrupt (INTTSnCD0) is valid.
- If the A/D conversion trigger is enabled when the INTTSnCC4 or INTTSnCC5 interrupt occurs during count-up/down operation, the period of interrupt occurrence is different from the count state indicated by the TSnOPT0.TSnCUF bit (refer to “Count-up flags (TSnCUF, TSnSUF)” on page 717).
  - Count-up:  

$$\text{TSnDTC0} < \text{TSnCCR4}, \text{TSnCCR5} < \text{TSnCCR0} - \text{TSnDTC1} - 2$$
  - Count-down:  

$$\text{TSnDTC0} + 2 < \text{TSnCCR4}, \text{TSnCCR5} < \text{TSnCCR0} - \text{TSnDTC1}$$

## 17.9 Error/warning Interrupt

### 17.9.1 Error interrupt function

When a positive/negative phase simultaneous active is detected, the TSnTBF flag of the TSnOPT6 register is set and an error interrupt (INTTSnER) of timer Sn is generated. Switching ON/OFF of error detection in each phase (TOSn1/TOSn2, TOSn3/TOSn4, TOSn5/TOSn6) is possible with the TSnTBA2 to TSnTBA0 bits of the TSnIOC4 register after error interrupts are enabled (TSnEOC of TSnIOC4 register = 1).

The following table lists whether or not positive/negative simultaneous active error detection is possible in each mode.

Mode	Positive/negative phase simultaneous active error detection
Interval timer mode	-
External event count mode	-
External trigger pulse output mode	-
One-shot pulse mode	-
PWM mode	√
Free-running mode	-
Triangular-wave PWM mode	√
High-accuracy T-PWM mode	√
PWM mode with dead time	√
120° excitation mode	√
Special pattern output mode	√

**Note** √: Detection possible  
 -: Detection not possible

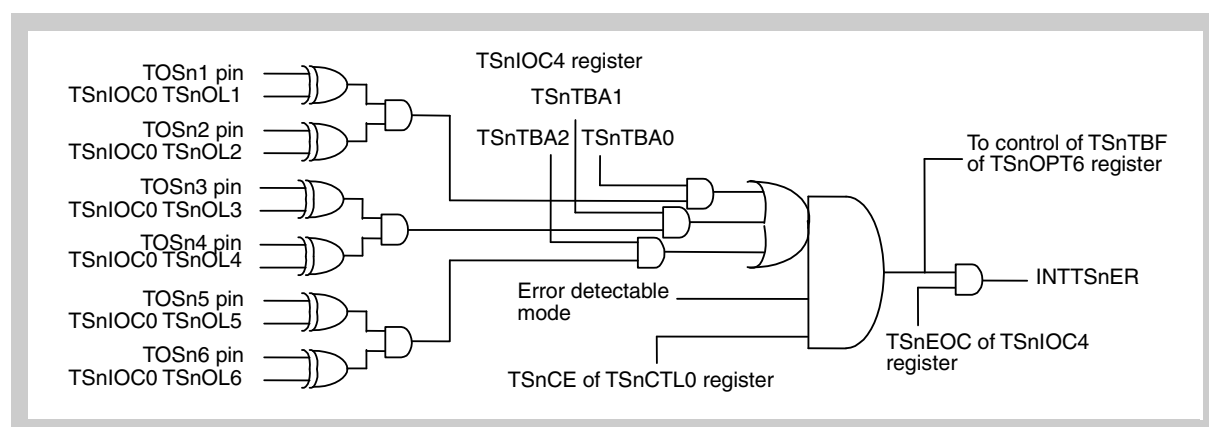


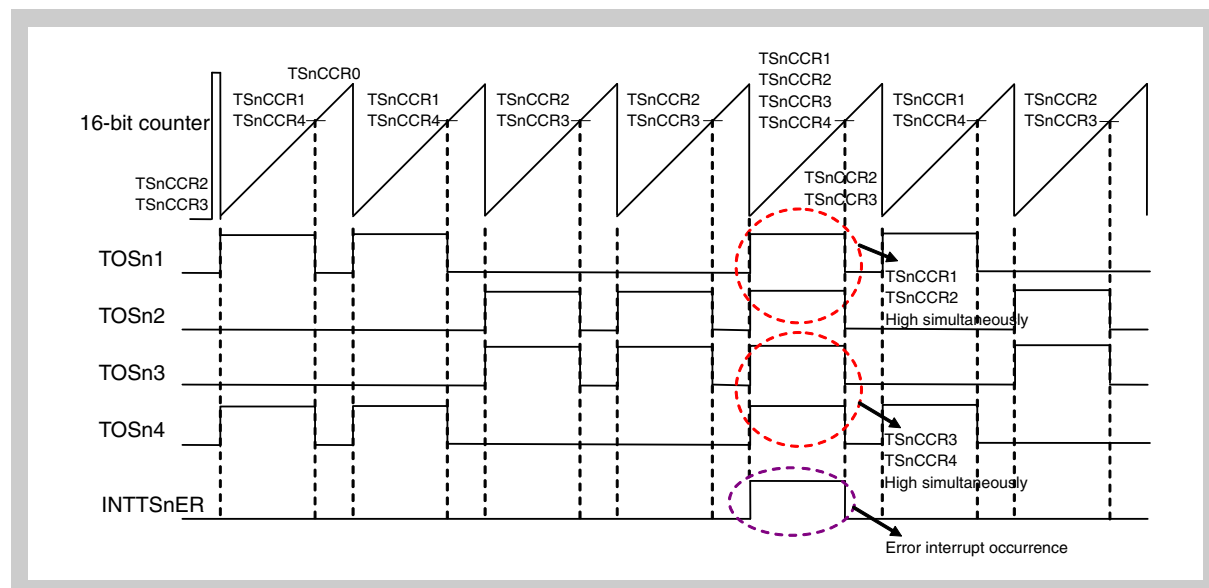
Figure 17-46 Error interrupt (INTTSnER) output control circuit

**Caution** If an error interrupt occurs, release the error status during error interrupt servicing. An error interrupt will not occur thereafter until the error status is released.

**(1) In PWM/special pattern output mode**

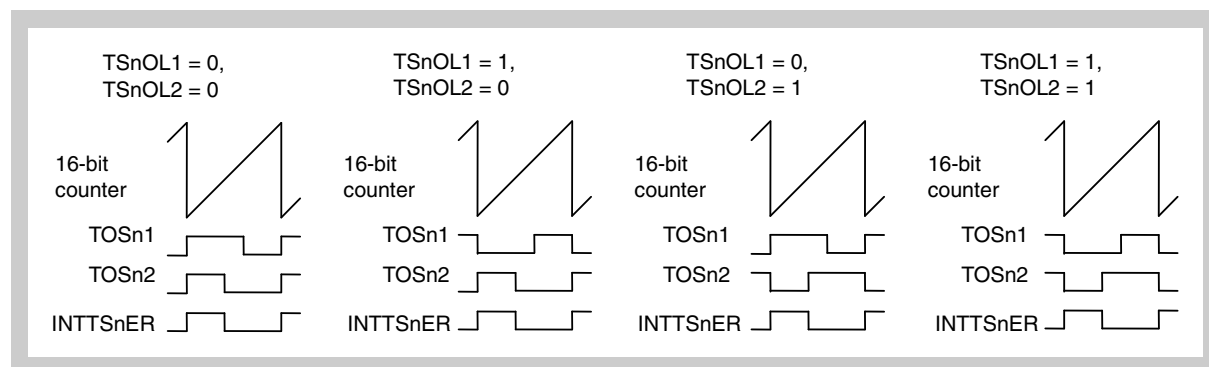
1. When the dead time control function is not used (TSnDSE of TSnOPT0 register = 0 in PWM mode)

As shown in Figure 17-47 on page 753, an error interrupt (INTTSnER) occurs when the TSnCCR1 and TSnCCR2 registers are set so that the active level is output from the TOSn1 and TOSn2 pins simultaneously. An error interrupt (INTTSnER) also occurs when the TSnCCR3 and TSnCCR4 registers are set so that the active level is output from the TOSn3 and TOSn4 pins simultaneously.



**Figure 17-47** Generation of error interrupt (INTTSnER) (in PWM mode)

The following is the case where the output active level is switched by using the TSnOL1 and TSnOL2 bits of the TSnIOC0 register.



**Figure 17-48** Generation of error interrupt in each active level (INTTSnER)

2. When the dead time control function is used (TSnDSE of the TSnOPT0 register = 1 in PWM mode, or in special pattern output mode)  
Even if the timer output is inactive, an error interrupt occurs when the set conditions occurs simultaneously. For details refer to
  - “Dead time control in high-accuracy T-PWM mode” on page 826
  - “Dead time control in PWM mode with dead time” on page 847
  - “Dead time control in 120° excitation mode” on page 874
  - “Dead time control in special 120° excitation mode” on page 916
  - “Dead time control in special pattern output mode” on page 929

## (2) Triangular-wave PWM mode

As shown in Figure 17-49 on page 754, an error interrupt (INTTSnER) occurs when the TSnCCR0 and TSnCCR1 registers are set so that the active level is output from the TOSn1 and TOSn2 pins simultaneously. An error interrupt (INTTSnER) also occurs when the TSnCCR3 and TSnCCR4 registers are set so that the active level is output from the TOSn3 and TOSn4 pins simultaneously.

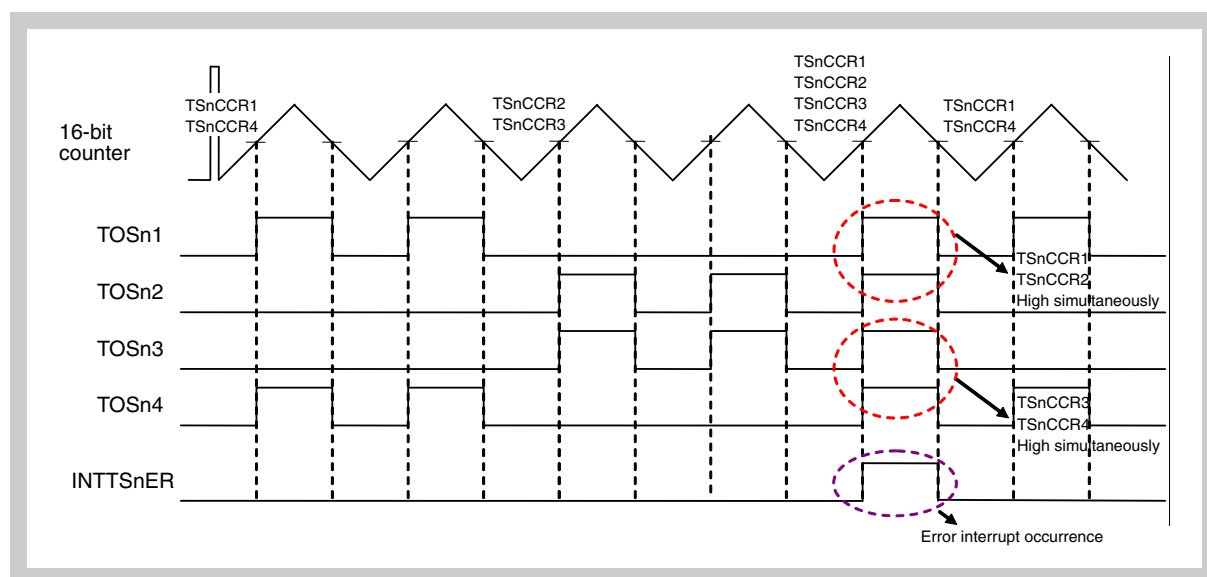


Figure 17-49 Generation of error interrupt (INTTSnER) (in triangular PWM mode)

## (3) In high-accuracy T-PWM mode, PWM mode with dead time, or 120° excitation mode

Errors caused by setting values of the compare register, the dead time setting register in the high-accuracy T-PWM mode, PWM mode with dead time, or 120° excitation mode do not occur.

**(4) In special 120° excitation mode**

An error interrupt (INTTSnER) occurs when timer Sn pattern registers 0 and 1 (TSnPAT0, TSnPAT1) are set so that the active level is output simultaneously. Refer to the caution on setting of the TSnPAT0 and TSnPAT1 registers in the special 120° excitation mode in “*Difference between 120° excitation mode and special 120° excitation mode*” on page 919 for details.

**17.9.2 Warning interrupt function**

Timer S has the warning interrupt (INTTSnWN) function.

A warning interrupt (INTTSnWN) is generated when any of the following conditions is detected.

Refer to “*Flags*” on page 716 for details.

- When two or more simultaneous pin changes are detected at the TAPTSn2 to TAPTSn0 pins  
(Refer to “*Noise detection flag (TSnNDF)*” on page 722)
- When a reverse is detected at the TAPTSn2 to TAPTSn0 pins  
(Refer to “*Pattern reverse detection flag (TSnPRF)*” on page 726)
- When [0, 0, 0] or [1, 1, 1] is detected at the TAPTSn2 to TAPTSn0 pins  
(Refer to “*Pattern error detection flag (TSnPPEF)*” on page 725)
- When the TSnPAT2 to TSnPAT0 pins are toggled three times or more during TSnSTCI0 and TSnSTCI1 signal triggers  
(Refer to “*TAPTSn2 to TAPTSn0 pin abnormal toggle detection flag (TSnPTF)*” on page 728)
- When the TSnSTCI0 and TSnSTCI1 signals are detected simultaneously  
(Refer to “*TSnSTCI0, TSnSTCI1 signal simultaneous trigger detection flag (TSnTDF)*” on page 730)
- When the phase relation between the input pattern (TAPTSn2 to TAPTSn0) and the output pattern (TSnOPF2 to TSnOPF0) is displaced  
(Refer to “*Pattern phase difference detection flag (TSnPPF)*” on page 731)

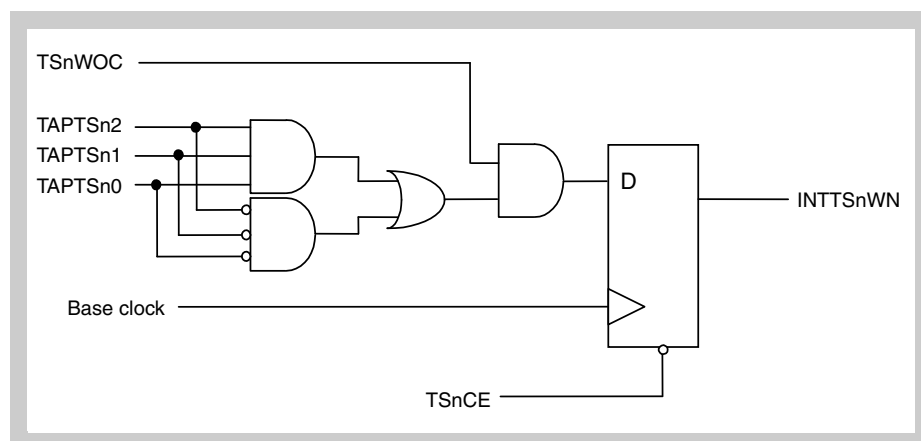


Figure 17-50 Error detection at TAPTSn2 to TAPTSn0 pins

## 17.10 Operation in Each Mode

Table 17-32 Mode list

TSnCTL1 Register				Timer Mode
TSnMD3	TSnMD2	TSnMD1	TSnMD0	
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 mode
0	1	0	1	Free-running mode
0	1	1	1	Triangular-wave PWM mode
1	0	0	0	High-accuracy T-PWM mode <sup>Notes 1, 2</sup>
1	0	0	1	PWM mode with dead time <sup>Note 1</sup>
1	0	1	0	120° excitation mode <sup>Notes 1, 2</sup>
1	0	1	1	Special 120° excitation mode <sup>Notes 1, 2</sup>
1	1	0	0	Special pattern output mode
Other than above				Setting prohibited

- Note**
1. The software output function can be used by setting TSnSOC of the TSnOPT4 register to 1.
  2. 180° excitation control can be used by setting TSnADC of the TSnOPT5 register to 1.



### 17.10.1 Interval timer mode

In the interval timer mode, a compare match interrupt (INTTSnCC0) is generated upon a match between the TSnCCR0 register and 16-bit counter, and the 16-bit counter is then cleared. The interval time means the INTTSnCC0 generation interval.

In the interval timer mode, the 16-bit counter is cleared only upon a match between the 16-bit counter and the TSnCCR0 register.

The 16-bit counter is not cleared by using the TSnCCR1 to TSnCCR5 registers.

Setting values of the TSnCCR1 to TSnCCR5 registers are transferred to the TSnCCR1 to TSnCCR5 buffer registers to be compared with the 16-bit counter value, and a compare match interrupt (INTTSnCC1 to INTTSnCC5) then occurs.

The TSnCCR0 to TSnCCR5 registers can be rewritten with the anytime rewrite method regardless of the timer operation (TSnCE bit values of TSnCTL0 register).

Toggle output is performed from the TOSn0 to TOSn7 pins by setting the TSnOE0 to TSnOE7 bits of the TSnLOC0 register to 1.

**Note** The interval timer mode will be valid when TSnMD3 to TSnMD0 of the TSnCTL1 register are set to [0, 0, 0, 0].

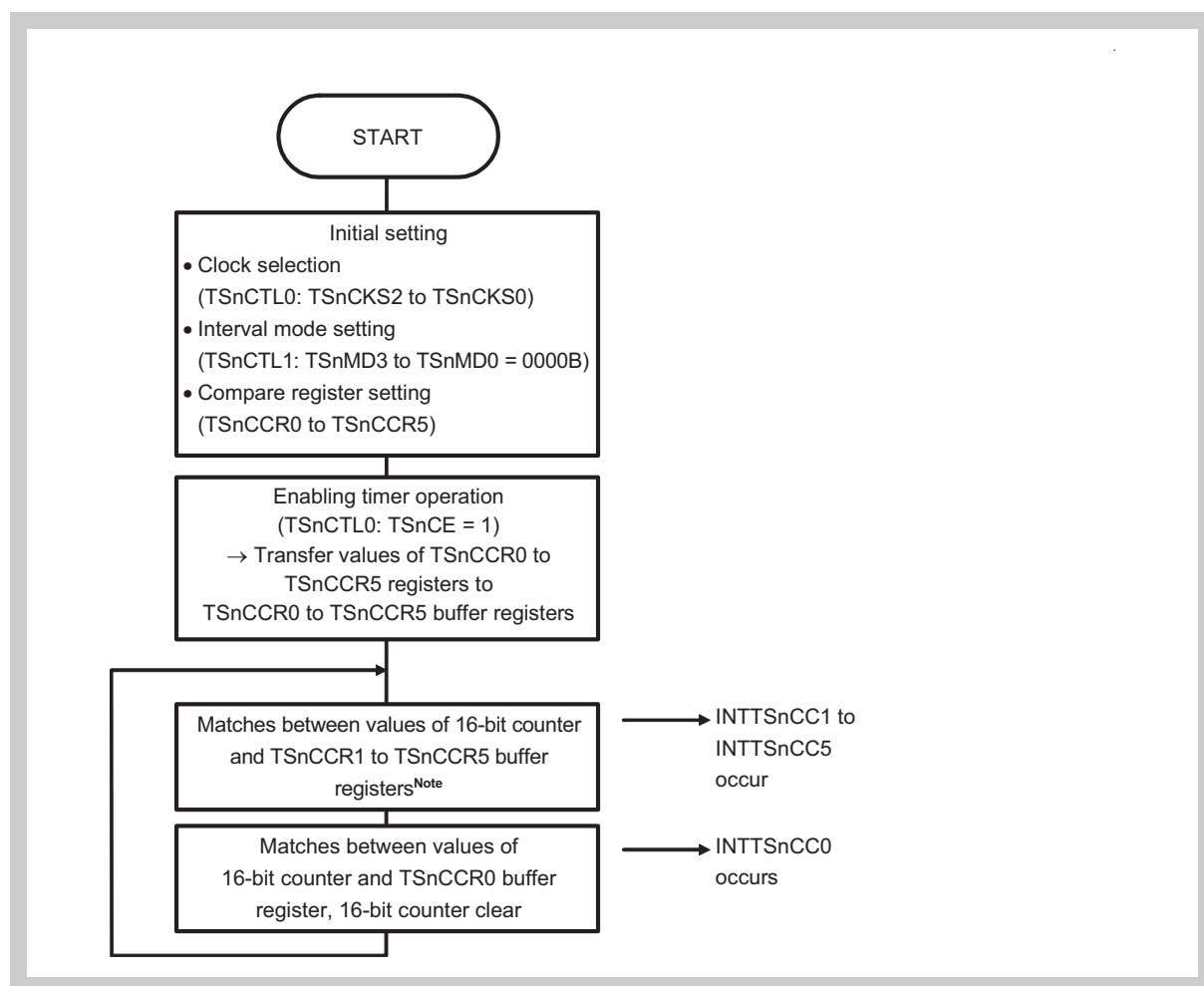


Figure 17-51 Basic operation flow in interval timer mode

**Note** The 16-bit counter is not cleared upon a match with the TSnCCR1 to TSnCCR5 registers.

**(1) Interval timer mode operation list****(a) Register rewriting**

Register	Rewriting method	Rewriting during operation	Function
TSnCCR0	Anytime rewrite	Possible	Compare value
TSnCCR1 to TSnCCR3	Anytime rewrite	Possible	Compare value
TSnCCR4, TSnCCR5	Anytime rewrite	Possible	Compare value

**(b) Input pins**

Pin	Function
TTRGSn	-
TEVTSn	-

**(c) Output pins**

Pin	Function
TOSn0 to TOSn5	Toggle output upon a compare match with the TSnCCR0 to TSnCCR5 registers
TOSn6, TOSn7	-

**(d) Interrupt**

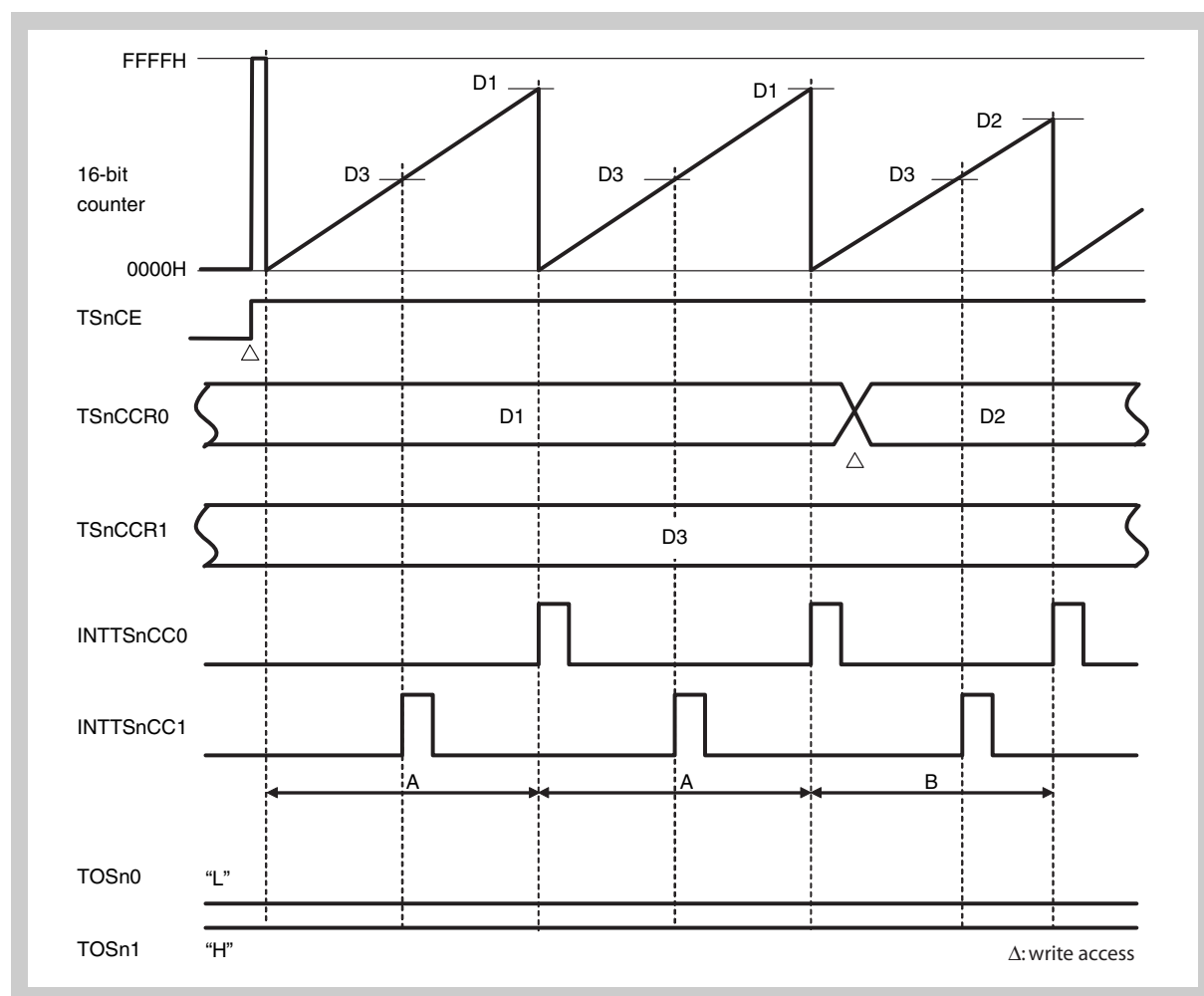
Interrupt	Function
INTTSnCC0 to INTTSnCC5	Compare match with the TSnCCR0 to TSnCCR5 registers
INTTSnOV	-
INTTSnER	-
INTTSnOD	-
INTTSnCD0	-
INTTSnWN	-

**(e) Compare match timing**

Compare match	Timing
TSnCCR0	Timing of switching 16-bit counter from TSnCCR0 to 0000H
TSnCCR1 to TSnCCR5	Timing of switching 16-bit counter from TSnCCR1 to TSnCCR5 to TSnCCR1 to TSnCCR5 + 1

**Note** “-” indicates an unused function in the interval timer mode.

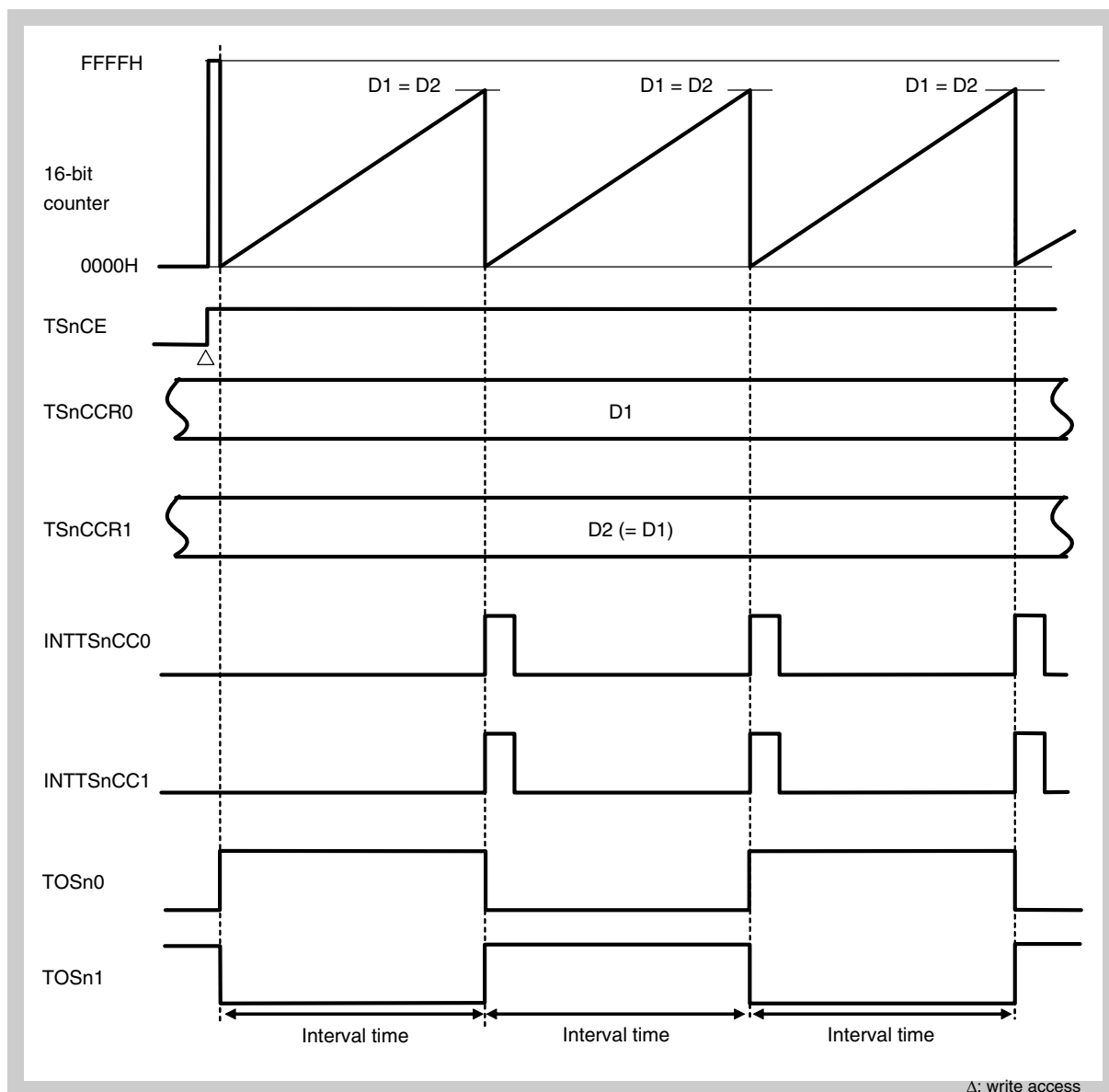
In case of  $D1 > D2 > D3$ , only value of the TSnCCR0 register is rewritten, without output from TOSn0 and TOSn1  
(TSnOE0 = 0, TSnOE1 = 0, TSnOL0 = 0, and TSnOL1 = 1 for TSnIOC0 register)



**Figure 17-52 Basic timing in interval timer mode (1/2)**

- Note**
1. D1, D2: TSnCCR0 register setting values (0000H to FFFFH)  
D3: TSnCCR1 register setting value (0000H to FFFFH)
  2. Interval time =A: interval time  $(D1 + 1) \times \text{count clock cycle}$   
B: Interval time  $(D2 + 1) \times \text{count clock cycle}$
  3. TSnCCR2 to TSnCCR5 registers have the same functions as those of TSnCCR1.

In case of  $D1 = D2$ , without rewriting of TSnCCR0 and TSnCCR1 registers, with output from TOSn0 and TOSn1 (TSnOE0 = 1, TSnOE1 = 1, TSnOL0 = 0, and TSnOL1 = 1 for TSnIOC0 register)



**Figure 17-53 Basic timing in interval timer mode (2/2)**

- Note**
1. D1: TSnCCR1 register setting value (0000H to FFFFH)  
D2: TSnCCR1 register setting value (0000H to FFFFH)
  2. Interval time = TOSn0, TOSn1 toggle time =  $(D1 + 1) \times (\text{count clock cycle})$
  3. The TSnCCR2 to TSnCCR5 registers have the same functions as those of TSnCCR1.

### 17.10.2 External event count mode

In the external event count mode, the count-up operation starts with external event input (TEVTSn pin). At that time, the input signal from the TEVTSn pin is used as the count clock, regardless of the TSnEEE bit of the TSnCTL1 register. In the external event count mode, the 16-bit counter is cleared only upon a match between the 16-bit counter and the TSnCCR0 register. The 16-bit counter cannot be cleared by using the TSnCCR1 to TSnCCR5 registers. The setting values of the TSnCCR1 to TSnCCR5 registers are transferred to the TSnCCR1 to TSnCCR5 buffer registers to be compared with the 16-bit counter value, and a compare match interrupt (INTTSnCC1 to INTTSnCC5) then occurs.

The TSnCCR0 to TSnCCR5 registers can be rewritten with the anytime rewrite method regardless of the timer Sn operation (TSnCE bit value of TSnCTL0 register).

Toggle output is performed from the TOSn0 to TOSn7 pins by setting the TSnOE0 to TSnOE7 bits of the TSnIOC0 register to 1.

- Note**
1. In the external event count mode, the edge detection count of the TEVTSn pin input is  $m + 1$ , where the TSnCCR0 register setting value is  $m$ .
  2. The external event count mode will be valid when TSnMD3 to TSnMD0 of the TSnCTL1 register are set to [0, 0, 0, 1].

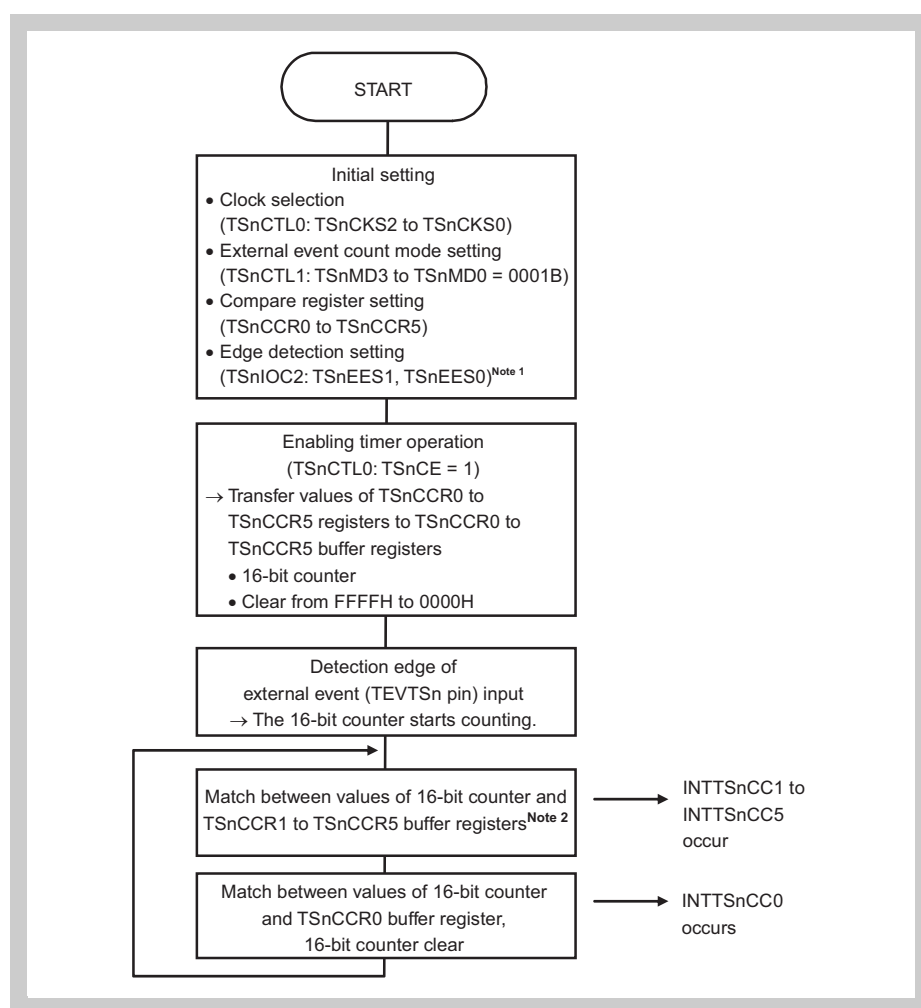


Figure 17-54 Basic operation flow in external event count mode

- Note**
1. Set other than TSnEES1 = 0 and TSnEES0 = 0.
  2. The 16-bit counter is not cleared upon a match between the values of the 16-bit counter and the TSnCCR1 to TSnCCR5 buffer registers.

### (1) External event count mode operation list

#### (a) Register rewriting

Register	Rewriting method	Rewriting during operation	Function
TSnCCR0	Anytime rewrite	Possible	Compare value
TSnCCR1 to TSnCCR3	Anytime rewrite	Possible	Compare value
TSnCCR4, TSnCCR5	Anytime rewrite	Possible	Compare value

#### (b) Input pin

Pin	Function
TTRGSn	-
TEVTSn	Timer count through external event count signal

#### (c) Output pin

Pin	Function
TOSn0 to TOSn5	Toggle output upon a compare match with TSnCCR0 to TSnCCR5 registers
TOSn6, TOSn7	-

#### (d) Interrupt

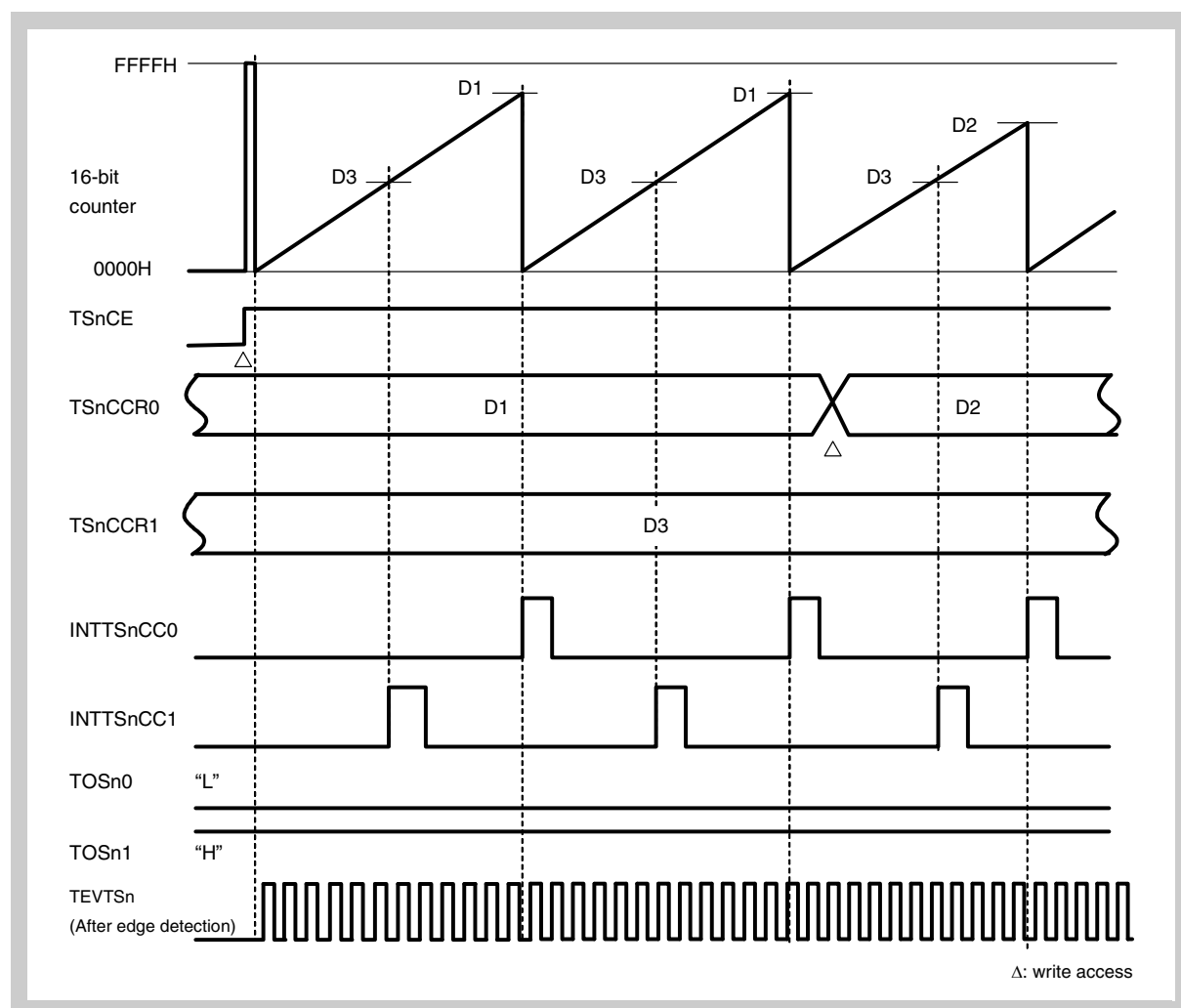
Interrupt	Function
INTTSnCC0 to INTTSnCC5	Compare match with TSnCCR0 to TSnCCR5 registers
INTTSnOV	-
INTTSnER	-
INTTSnOD	-
INTTSnCD0	-
INTTSnWN	-

#### (e) Compare match timing

Compare match	Timing
TSnCCR0	Timing of switching 16-bit counter from TSnCCR0 to 0000H
TSnCCR1 to TSnCCR5	Timing of switching 16-bit counter from <TSnCCR1 to TSnCCR5> to TSnCCR1 to TSnCCR5 + 1

**Note** “-” indicates an unused function in the external event count mode.

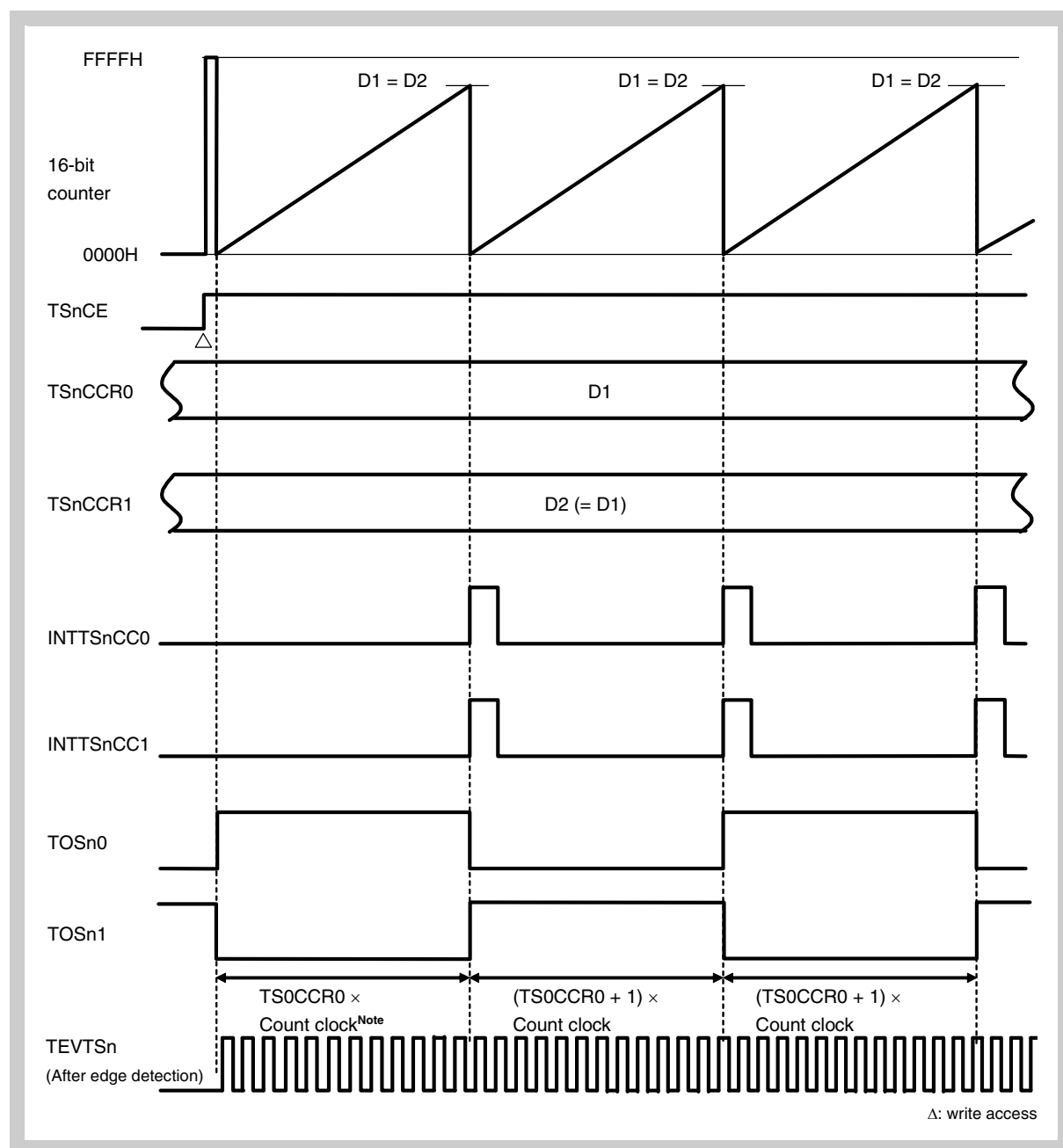
In case of  $D1 > D2 > D3$ , rewriting of only TSnCCR0 register value, without output from TOSn0 and TOSn1  
 (TSnOE0 = 0, TSnOE1 = 0, TSnOL0 = 0, TSnOL1 = 1 for TSnIOC0 register  
 TSnEES1 = 1, TSnEES0 = 1 for TSnIOC register, both edges detected)



**Figure 17-55 Basic operation timing in external event count mode (1/4)**

- Note**
1. After the timer Sn operation starts (TSnCE = 1), the 1st cycle count value will be 1 less than the value after the 2nd cycle.
  2. D1, D2: TSnCCR0 register setting values (0000H to FFFFH)  
D3: TSnCCR1 register setting value (0000H to FFFFH)
  3. Event count = (Dk + 1)
  4. The TSnCCR2 to TSnCCR5 registers have the same functions as those of TSnCCR1.
  5. k = 1, 2

In case of  $D1 = D2$ ,  $TSnCCR0$ , without rewriting of  $TSnCCR1$  register value, with output from  $TOSn0$  and  $TOSn1$  ( $TSnOE0 = 1$ ,  $TSnOE1 = 1$ ,  $TSnOL0 = 0$ , and  $TSnOL1 = 1$  for  $TSnIOC0$  register)

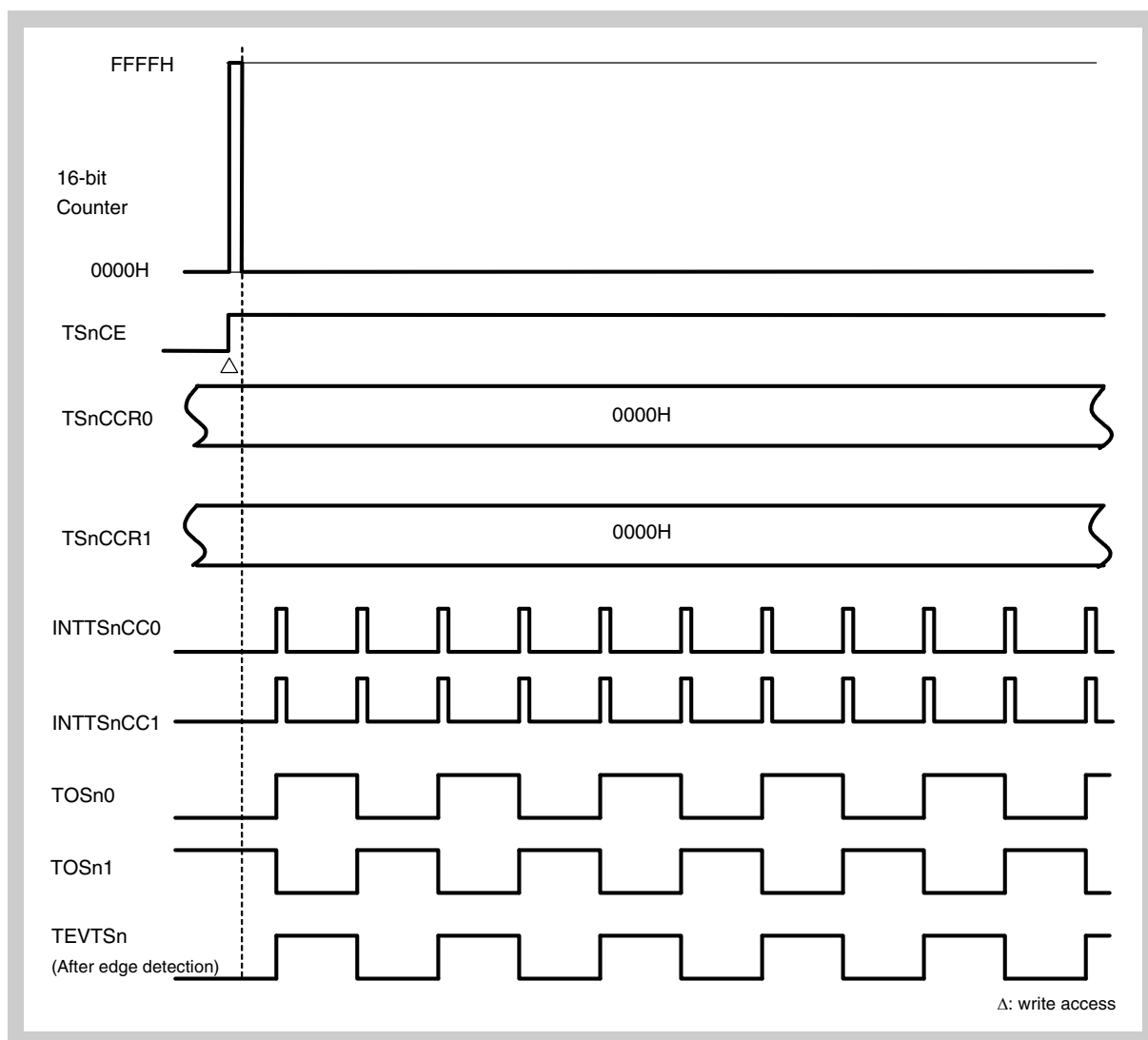


**Figure 17-56 Basic operation timing in external event count mode (2/4)**

- Note**
1. After the timer Sn operation starts ( $TSnCE = 1$ ), the 1st cycle count value will be 1 less than the value after the 2nd cycle.
  2. D1:  $TSnCCR0$  register setting value (0000H to FFFFH)  
D2:  $TSnCCR1$  register setting value (0000H to FFFFH)
  3. Event count =  $(D1 + 1)$
  4. The  $TSnCCR2$  to  $TSnCCR5$  registers have the same functions as those of  $TSnCCR1$ .



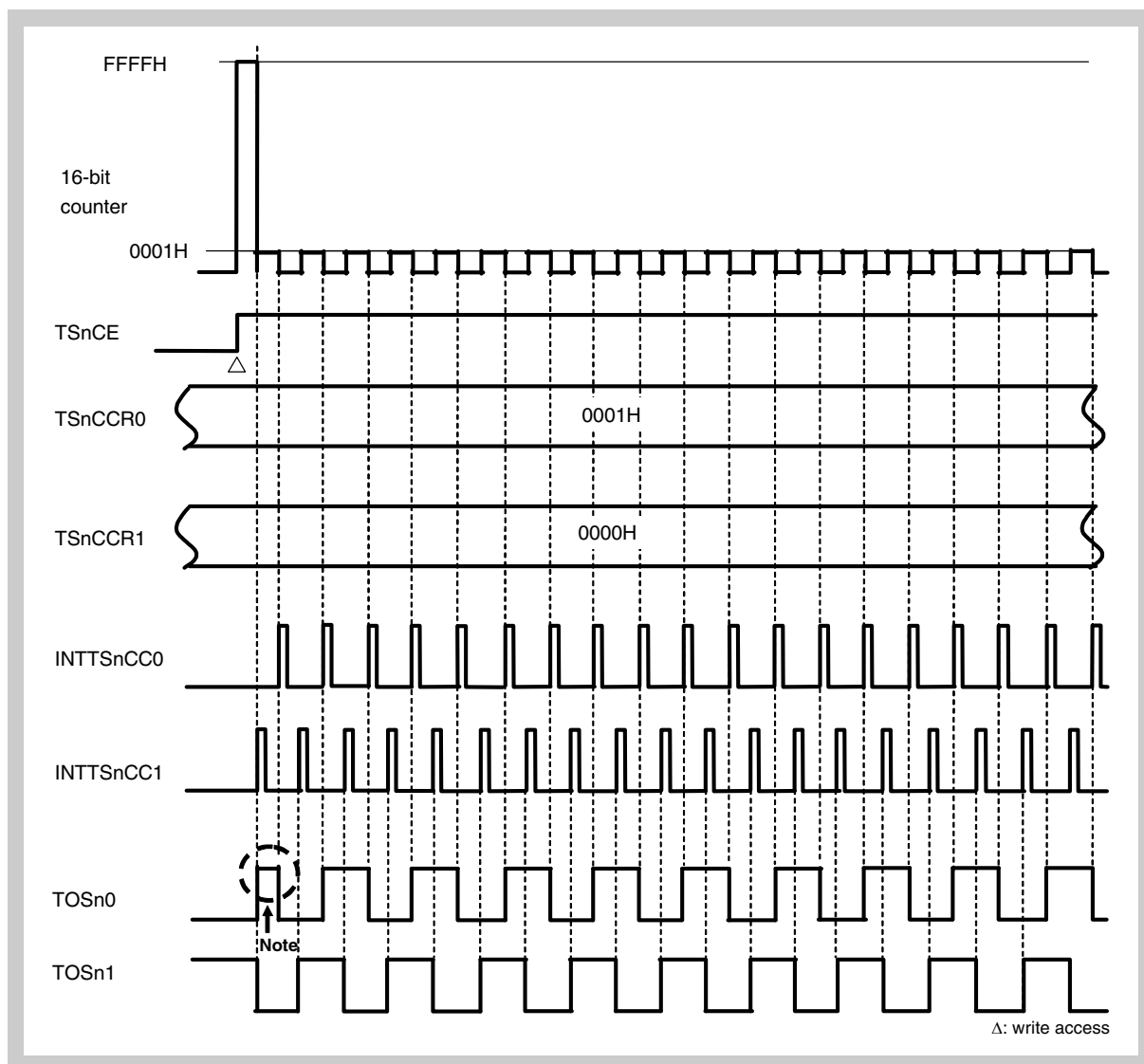
In case of D1 = D2, TSnCCR0, without rewriting of TSnCCR1 register value,  
 with output from TOSn0 and TOSn1  
 (TSnOE0 = 1, TSnOE1 = 1, TSnOL0 = 0, and TSnOL1 = 1 for TSnIOC0  
 register)  
 TSnEES1 = 1, TSnEES0 = 1 for TSnIOC register, both edges detected



**Figure 17-57 Basic operation timing in external event count mode (3/4)**

- Note**
1. Event count = 1
  2. The TSnCCR2 to TSnCCR5 registers have the same functions as those of TSnCCR1.

In case of  $D1 = D2$ , without rewriting of TSnCCR0, TSnCCR1 register value,  
with output from TOSn0 and TOSn1  
(TSnOE0 = 1, TSnOE1 = 1, TSnOL0 = 0, TSnOL1 = 1 for TSnIOC0 register)



**Figure 17-58 Basic operation timing in external event count mode (4/4)**

- Note**
1. After the timer Sn operation starts (TSnCE = 1), the 1st cycle count value will be 1 less than the value after the 2nd cycle.
  2. D1: TSnCCR0 register setting value (0001H)  
D2: TSnCCR1 register setting value (0001H)
  3. Event count = (Dk + 1)
  4. TOSn0 to TOSn5 are toggled when the 16-bit counter is set to 0001H.
  5. The TSnCCR2 to TSnCCR5 registers have the same functions as those of TSnCCR1.
  6. k = 1, 2

**(2) Caution on timer output**

In the external event count mode, the 16-bit counter is set to 0000H by setting the TSnCE bit to 1, regardless of TEVTSn pin edge detection. Output of the TOSn0 to TOSn5 pins will be active when the value is switched from 0000H to 0001H in the 1st cycle, and when the TSnCCR0 value is switched to 0000H in the 2nd and subsequent cycles. Therefore, the toggle width of the TOSn0 to TOSn5 pins differs between the 1st, 2nd and subsequent cycles.

**Differences between 1st cycle and 2nd cycle toggle width of TOSn0 to TOSn5**

- TOSn0

Output is toggled at the TSnCCR0 count in the 1st counter cycle. In the 2nd and subsequent cycles, output is toggled at every TSnCCR0 + 1 count.

- TOSn1 to TOSn5

- TSnCCR1 to TSnCCR5 = TSnCCR0

Output is toggled at the TSnCCR0 count in the 1st counter cycle. In the 2nd and subsequent cycles, output is toggled at every TSnCCR0 + 1 count.

- TSnCCR1 to TSnCCR5 = 0000H

Output is toggled at every TSnCCR0 + 1 count in all counter cycles.

- 0000H < TSnCCR1 to TSnCCR5 < TSnCCR0

Output is toggled at TSnCCR1 to TSnCCR5 counts in the 1st counter cycle. In the 2nd and subsequent cycles, output is toggled at every TSnCCR0 + 1 count.

### 17.10.3 External trigger pulse output mode

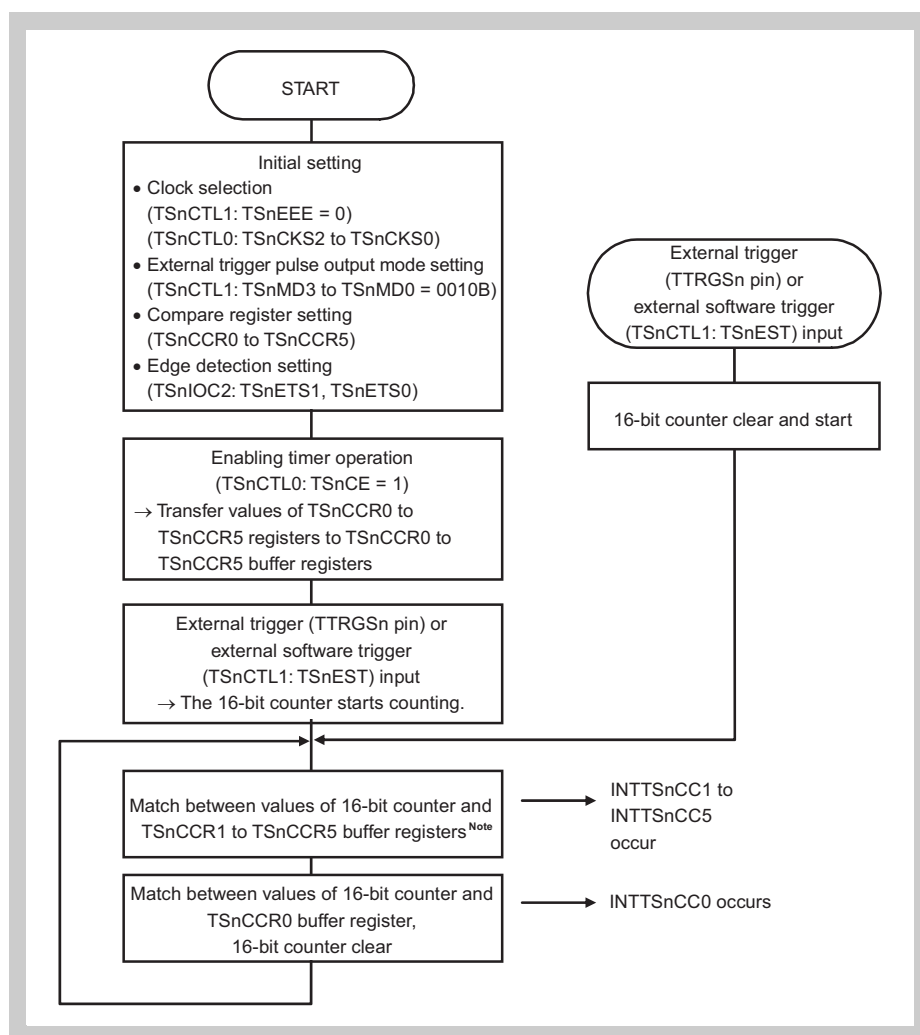
In the external trigger pulse output mode, when a duty is set with the TSnCCR1 to TSnCCR5 registers and a cycle is set with the TSnCCR0 register, and TSnCE is then set to 1, the 16-bit counter remains stopped with FFFFH and waits for external trigger input (TTRGSn pin). Count-up operation starts when a TTRGSn pin valid edge is detected or when the TSnEST bit of the TSnCTL1 register is set to 1. An external trigger pulse is output from the TOSn1 to TOSn5 pins. Toggle output is performed from the TOSn0 pin upon a match with the TSnCCR0 register value. A compare match interrupt (INTTSnCC0 to INTTSnCC5) occurs upon a match between the values of the 16-bit counter and the TSnCCR0 to TSnCCR5 registers during count operation.

The TSnCCR0 to TSnCCR5 registers can be rewritten by reload operation during count operation. Compare register values are reloaded upon a match between the values of the 16-bit counter and the TSnCCR0 register.

If a TTRGSn pin valid edge is detected or if the TSnEST bit of the TSnCTL1 register is set to 1 during operation in the external trigger pulse output mode, the 16-bit counter is cleared and starts the count-up operation again. In this case, if the TOSn1 to TOSn5 pins are inactive, the output from the TOSn1 to TOSn5 pins is active when an external trigger is input. If the TOSn1 to TOSn5 pins are inactive, the output remains active even if an external trigger is input. In the external trigger pulse output mode, the capture function cannot be used because the function of the TSnCCR0 to TSnCCR5 registers is fixed as a compare register.

- 
- Caution**
1. Clear the TSnEEE bit of the TSnCTL1 register to 0 in the external trigger pulse output mode.
  2. When the TSnRTE bit of the TSnOPT1 register is set to 1, reloading is executed by write to the TSnCCR1 register. In case of rewriting only the TSnCCR0 register value, the same value must also be written to the TSnCCR1 register. In this case, reloading is not performed if only the TSnCCR0 register is rewritten.
- 

**Note** The external trigger pulse output mode will be valid when TSnMD3 to TSnMD0 of the TSnCTL1 register are set to [0, 0, 1, 0].



**Figure 17-59 Basic operation flow in external trigger pulse output mode**

**Note** The 16-bit counter is not cleared upon a match between the values of the 16-bit counter and the TSnCCR1 to TSnCCR5 buffer registers.

**(1) External trigger pulse output mode operation list****(a) Register rewriting**

Register	Rewriting method	Rewriting during operation	Function
TSnCCR0	Reload	Possible	Cycle
TSnCCR1 to TSnCCR3	Reload	Possible	Duty
TSnCCR4, TSnCCR5	Reload	Possible	Duty

**(b) Input pin**

Pin	Function
TTRGSn	16-bit counter clear and start by trigger input
TEVTSn	-

**(c) Output pin**

Pin	Function
TOSn0	Toggle output by TSnCCR0 register compare match or external trigger input
TOSn1 to TOSn5	External trigger pulse waveform output
TOSn6, TOSn7	-

**(d) Interrupt**

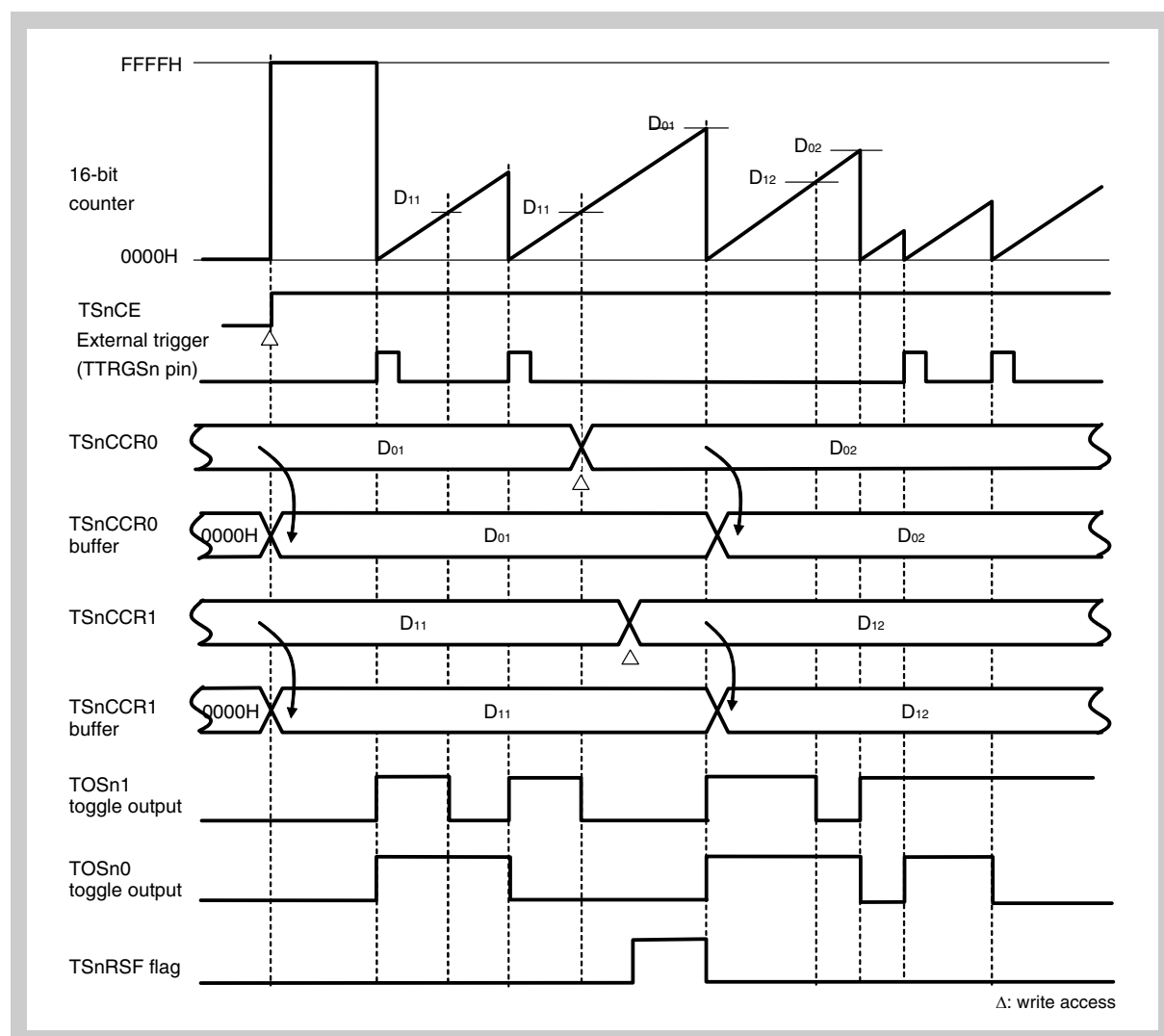
Interrupt	Function
INTTSnCC0 to INTTSnCC5	Compare match with TSnCCR0 to TSnCCR5 registers
INTTSnOV	-
INTTSnER	-
INTTSnOD	-
INTTSnCD0	-
INTTSnWN	-

**(e) Compare match timing**

Compare match	Timing
TSnCCR0	Timing of switching 16-bit counter from TSnCCR0 to 0000H
TSnCCR1 to TSnCCR5	After match detection between 16-bit counter and TSnCCR1 to TSnCCR5 values

**Note** “-” indicates an unused function in the external trigger pulse output mode.

In case of rewriting of TSnCCR0, TSnCCR1 register values, with output from TOSn0 and TOSn1  
(TSnOE0 = 1, TSnOE1 = 1, TSnOL0 = 0, and TSnOL1 = 0 for TSnIOC0 register)



**Figure 17-60 Basic operation timing in external trigger pulse output mode**

- Note**
1. D<sub>01</sub>, D<sub>02</sub>: TSnCCR0 register setting values (0000H to FFFFH)  
D<sub>11</sub>, D<sub>12</sub>: TSnCCR1 register setting values (0000H to FFFFH)
  2. TOSn1 (PWM) Duty = (TSnCCR1 register setting value) × (count clock cycle)  
Cycle = (TSnCCR0 register setting value + 1) × (count clock cycle)
  3. The TOSn0 pin is toggled immediately after the count start and when the 16-bit counter is cleared.
  4. The TSnCCR2 to TSnCCR5 registers have the same functions as those of TSnCCR1.

### 17.10.4 One-shot pulse mode

In the one-shot pulse mode, a cycle is set with the TSnCCR0 register and the output compare value is set with the TSnCCR1 to TSnCCR5 registers. When the TSnCE bit of the TSnCTL0 register is set to 1, the 16-bit counter remains stopped with FFFFH and waits for external trigger input (TTRGSn pin).

Count-up operation starts when the valid edge of an external trigger input (TTRGSn pin) is detected or when the TSnEST bit of the TSnCTL0 register is set to 1.

The TOSn1 to TOSn5 pins are active upon a match between the values of the 16-bit counter and the TSnCCR1 to TSnCCR5 registers. The TOSn1 to TOSn5 pins are inactive upon a match between the values of the 16-bit counter and the TSnCCR0 register, and the 16-bit counter is cleared to 0000H and then stops.

TOSn0 provides toggle output upon a match between the values of the 16-bit counter and the TSnCCR0 buffer register during count operation. A compare match interrupt (INTTSnCC0) occurs upon a match between the values of the 16-bit counter and the TSnCCR0 register, and a compare match interrupt (INTTSnCC1 to INTTSnCC5) occurs upon a match between the values of the 16-bit counter and the TSnCCR1 to TSnCCR5 buffer registers during count operation.

The TSnCCR0 to TSnCCR5 registers can be rewritten with the anytime rewrite method regardless of the timer Sn operation (TSnCE bit value of TSnCTL0 register).

Trigger input is ignored during 16-bit counter operation. The second trigger must be input while the 16-bit counter is stopped with 0000H.

In the one-shot pulse mode, the capture function cannot be used because the function of the TSnCCR0 to TSnCCR5 registers is fixed as a compare register.

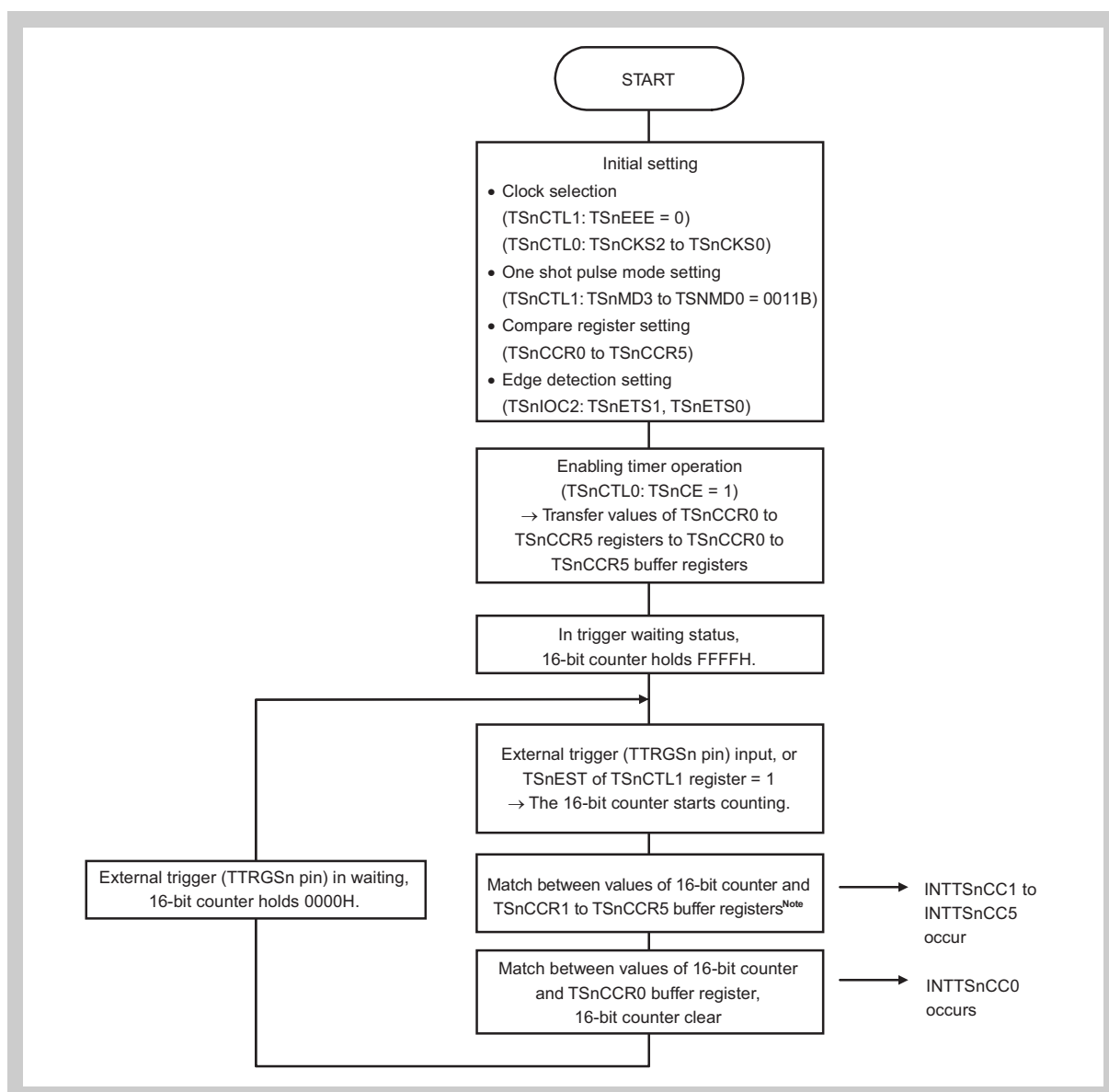
---

**Caution** Clear the TSnEEE bit of the TSnCTL1 register to 0 in the one-shot pulse mode.

---

**Note** The one-shot pulse mode will be valid when TSnMD3 to TSnMD0 of the TSnCTL1 register are set to [0, 0, 1, 1].





**Figure 17-61 Basic operation flow in one-shot pulse mode**

**Note** The 16-bit counter is not cleared upon a match between the values of the 16-bit counter and the TSnCCR1 to TSnCCR5 buffer registers.

**Caution** The 16-bit counter is not be cleared and trigger input is ignored even if the trigger is input during count-up operation by the 16-bit counter.

**(1) One-shot pulse mode operation list****(a) Register rewriting**

Register	Rewriting method	Rewriting During Operation	Function
TSnCCR0	Anytime rewrite	Possible	Cycle
TSnCCR1 to TSnCCR3	Anytime rewrite	Possible	Output delay value
TSnCCR4, TSnCCR5	Anytime rewrite	Possible	Output delay value

**(b) Input pin**

Pin	Function
TTRGSn	16-bit counter start by trigger input
TEVTSn	-

**(c) Output pin**

Pin	Function
TOSn0	Active at count start, inactive upon match of TSnCCR0 register
TOSn1 to TOSn5	Active upon match of TSnCCR1 to TSnCCR5 registers, inactive upon match of TSnCCR0 register
TOSn6, TOSn7	-

**(d) Interrupt**

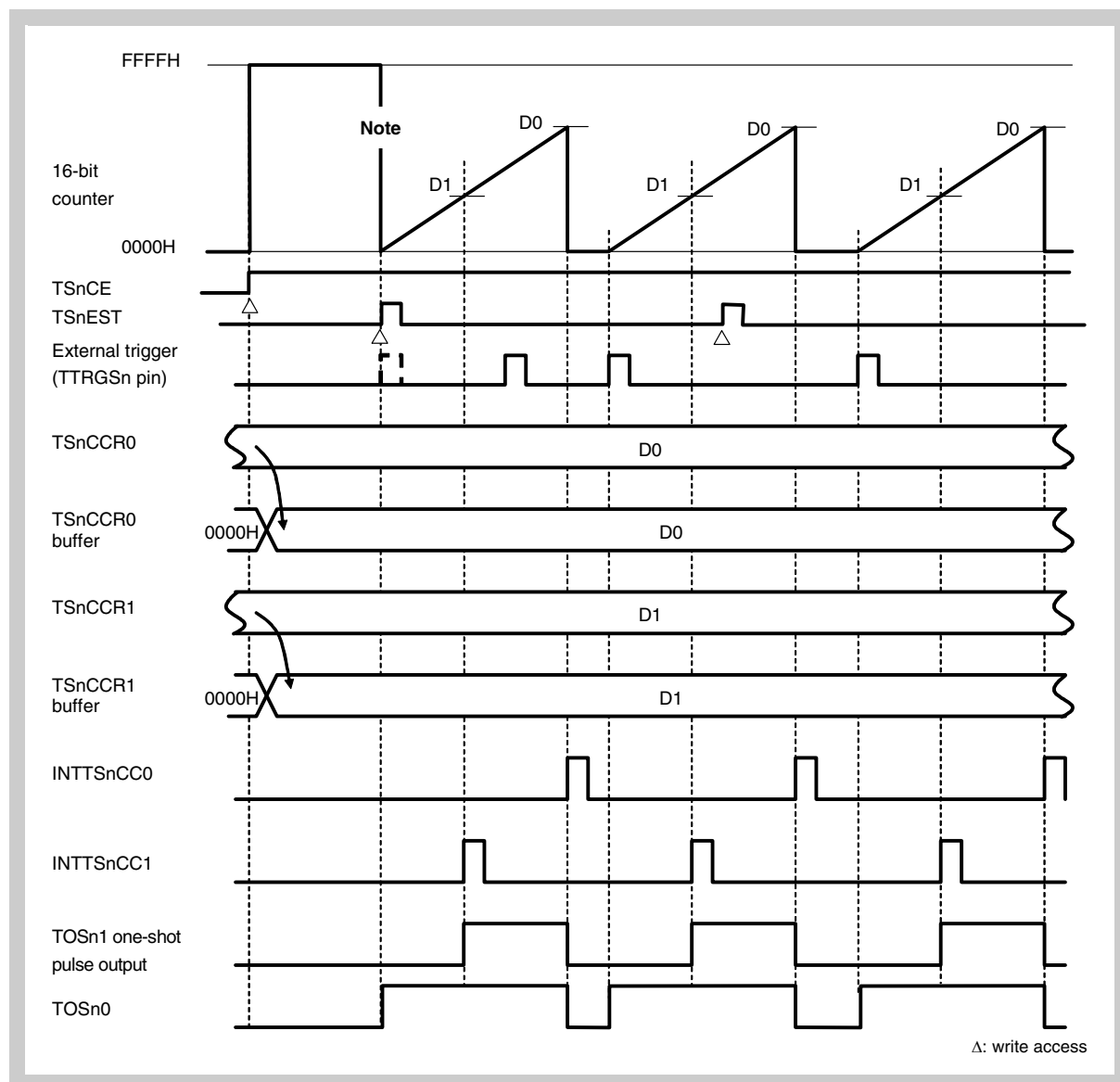
Interrupt	Function
INTTSnCC1 to INTTSnCC5	Compare match of TSnCCR1 to TSnCCR5 registers
INTTSnOV	-
INTTSnER	-
INTTSnOD	-
INTTSnCD0	-
INTTSnWN	-

**(e) Compare match timing**

Compare match	Timing
TSnCCR0	Timing of switching 16-bit counter from TSnCCR0 to 0000H
TSnCCR1 to TSnCCR5	After detection of match between values of 16-bit counter and TSnCCR1 to TSnCCR5 registers

- Note**
1. “-” indicates an unused function in the one-shot pulse mode.
  2. m = 1 to 5

(TSnOE0 = 1, TSnOE1 = 1, TSnOL0 = 1, and TSnOL1 = 1 for TSnIOC0 register,  
 TSnEES1 = 0 and TSnEES0 = 1 for TSnIOC2 register, rising edge detected)



**Figure 17-62 Basic operation timing in one-shot pulse mode**

- Note**
1. The 16-bit counter starts count-up operation when the TSnEST bit of the TSnCTL1 register is set to 1 or when a pulse is input from the TTRGSn pin.
  2. D0: TSnCCR0 register setting value (0000H to FFFFH)  
 D1: TSnCCR1 register setting value (0000H to FFFFH)
  3.  $TOSn1 \text{ (output delay)} = (TSnCCR1 \text{ register setting value}) \times (\text{count clock cycle})$   
 $TOSn1 \text{ (output pulse width)} = \{(TSnCCR0 \text{ register setting value} + 1) - (TSnCCR1 \text{ register setting value})\} \times (\text{count clock cycle})$
  4. The TOSn2 to TOSn5 pins have the same functions as those of the TOSn1 pin.

### 17.10.5 PWM mode

In the PWM mode, when a duty is set with the TSnCCR1 to TSnCCR5 registers and a cycle is set with the TSnCCR0 register, and the TSnCE bit of the TSnCTL0 register is then set to 1, duty variable type PWM is output from the TOSn1 to TOSn5 pins.

The TOSn1 to TOSn5 pins are active when count-up operation starts, and inactive upon a match between the values of the 16-bit counter and the TSnCCR1 to TSnCCR5 registers. Then, the TOSn1 to TOSn5 pins are active upon a match with the TSnCCR0 register value. The TOSn0 pin provides toggle output upon a match with the TSnCCR0 buffer register.

A compare match interrupt (INTTSnCC0 to INTTSnCC5) occurs upon a match between the values of the 16-bit counter and the TSnCCR0 to TSnCCR5 registers during count operation.

The TSnCCR0 to TSnCCR5 registers can be rewritten during count operation. Compare register values are reloaded upon a match between the 16-bit counter and TSnCCR0 register values.

The TSnCCR0 to TSnCCR5 registers can be rewritten with the reload write method, regardless of the timer Sn operation (TSnCE bit value of TSnCTL0 register).

In the PWM mode, the capture function cannot be used because the function of the TSnCCR0 to TSnCCR5 registers is fixed as a compare register.

---

**Caution** When the TSnRTE bit of the TSnOPT1 register is set to 1, reloading is executed by write to the TSnCCR1 register. In case of rewriting only the TSnCCR0 register value, the same value must also be written to the TSnCCR1 register. In this case, reloading is not performed if only the TSnCCR0 register is rewritten.

---

**Note** The PWM mode will be valid when TSnMD3 to TSnMD0 of the TSnCTL1 register are set to [0, 1, 0, 0].

In case of TSnCCR0, TSnCCR1 to TSnCCR5 values not rewritten during timer operation

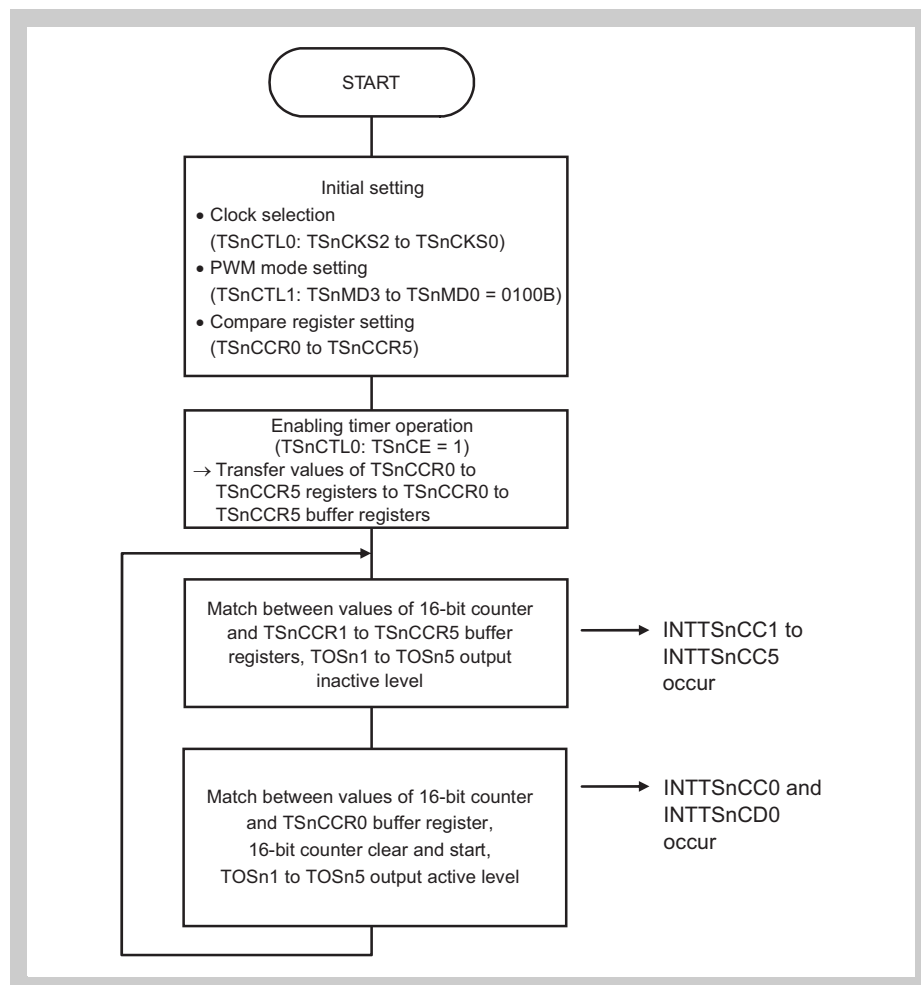


Figure 17-63 Basic operation flow in PWM mode (1/2)

In case of TSnCCR0, TSnCCR1 to TSnCCR5 values rewritten during timer operation

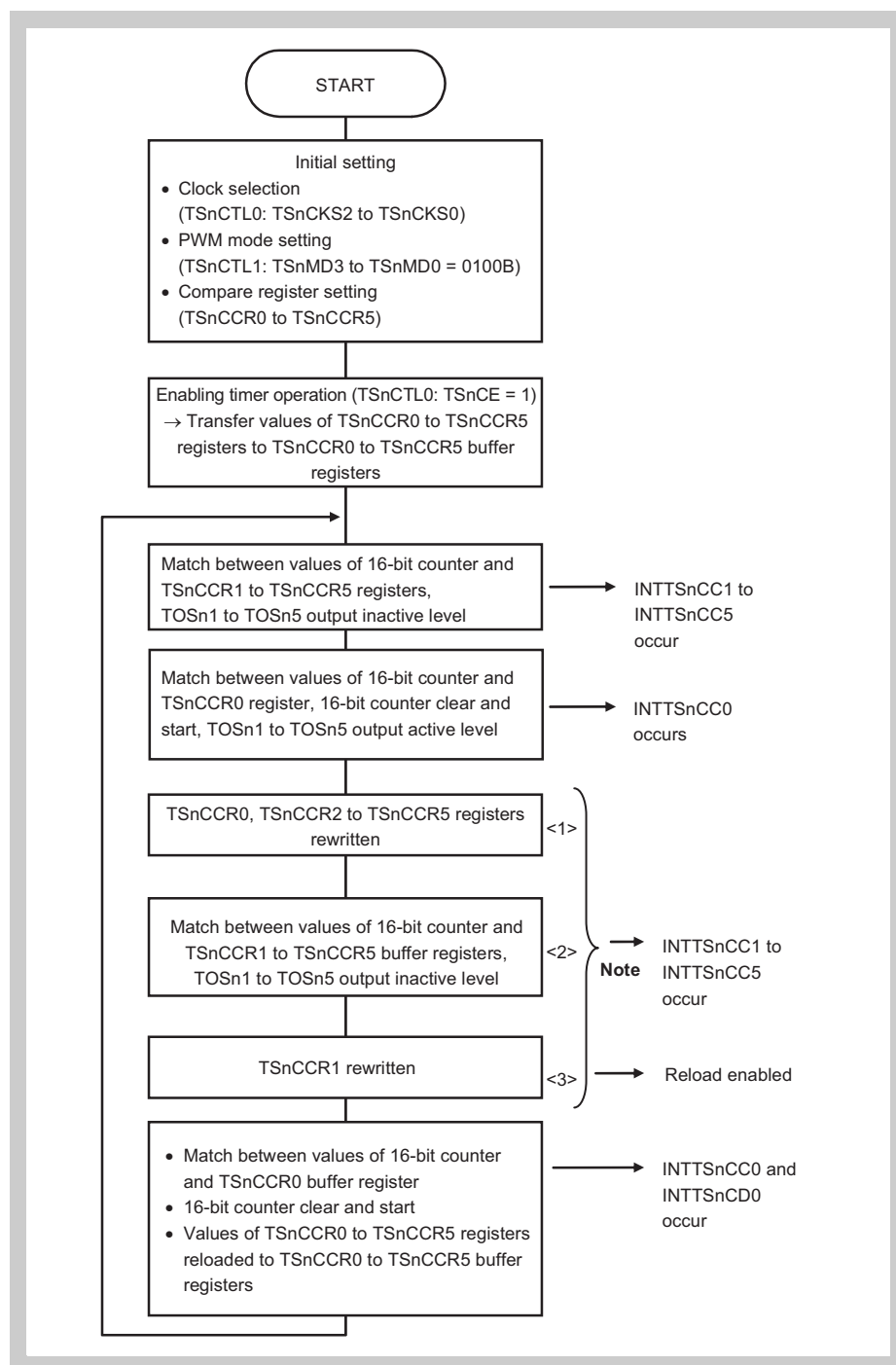


Figure 17-64 Basic operation flow in PWM mode (2/2)

**Note** Although the timing of <2> may differ according to the TSnCCR register value or the timing of rewriting <1> and <3>, be sure to perform <3> last.

**(1) PWM mode operation list****(a) Register rewriting**

Register	Rewriting Method	Rewriting During Operation	Function
TSnCCR0	Reload	Possible	Cycle
TSnCCR1 to TSnCCR3	Reload	Possible	Duty
TSnCCR4, TSnCCR5	Reload	Possible	Duty
TSnDTC0, TSnDTC1	Reload	Conditionally possible <sup>Note</sup>	Cycle, dead time

**(b) Input pin**

Pin	Function
TTRGSn	-
TEVTSn	-

**(c) Output pin**

Pin	Function
TOSn0	Toggle output by compare match of TSnCCR0 register
TOSn1 to TOSn5	PWM output by compare match of TSnCCR1 to TSnCCR5 registers
TOSn6	-
TOSn7	Pulse output by A/D conversion trigger

**(d) Interrupt**

Interrupt	Function
INTTSnCC0 to INTTSnCC5	Compare match of TSnCCR0 to TSnCCR5 registers
INTTSnOV	-
INTTSnER	Error
INTTSnOD	-
INTTSnCD0	Peak interrupt (simultaneous occurrence with INTTSnCC0)
INTTSnWN	-

**(e) Compare match timing**

Compare Match	Timing
TSnCCR0	Timing of switching 16-bit counter from TSnCCR0 to 0000H
TSnCCR1 to TSnCCR5	After detection of match between values of 16-bit counter and TSnCCR1 to TSnCCR5 registers

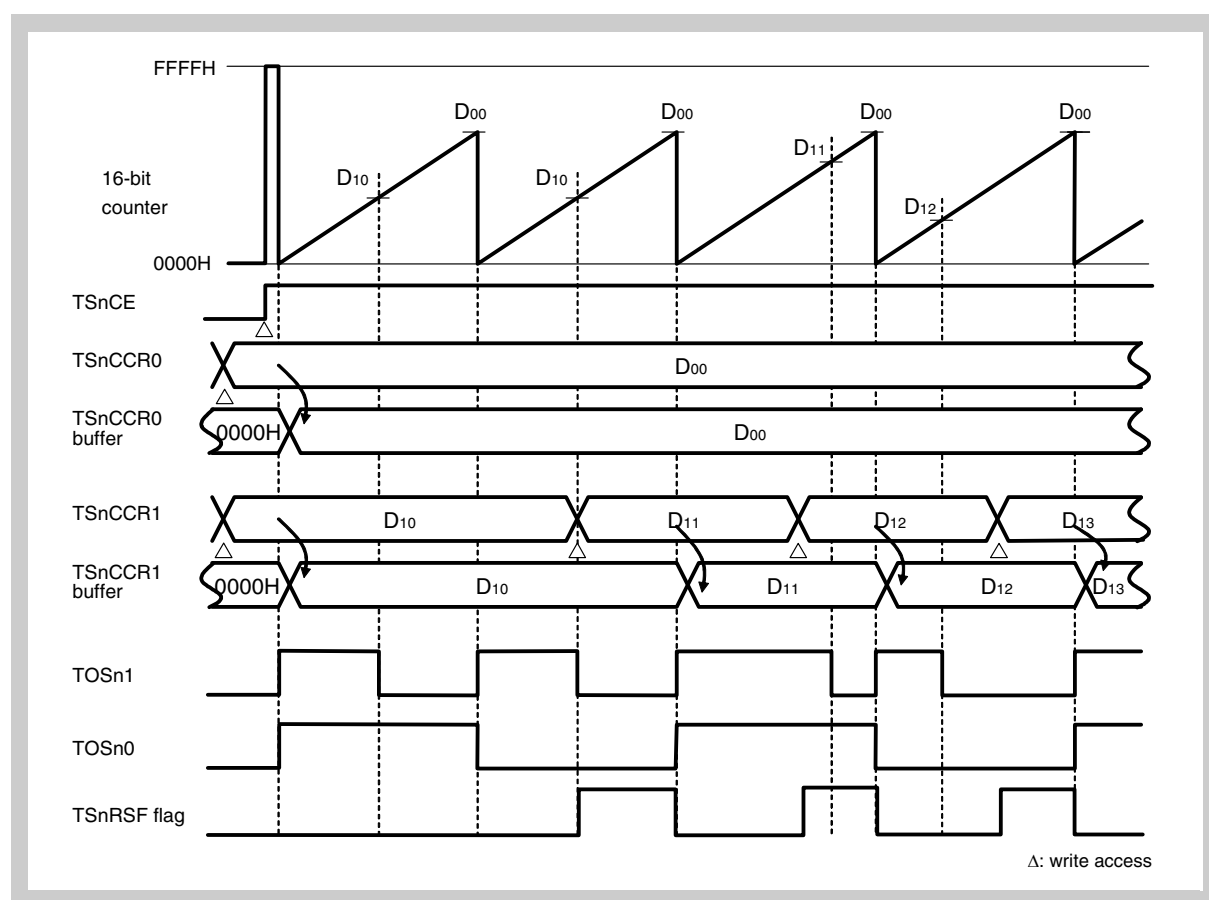
**Note** Refer to “Rewriting of TSnDTC0 and TSnDTC1 registers” on page 827 for details.

**(f) Output condition**

Output	Condition
Duty 0% output	TSnCCRm = 0000H
Duty 100% output	TSnCCRm = TSnCCR0 + 1
Dead time check	TSnDSE of TSnOPT0 register = 1

- Note**
1. “–” indicates an unused function in the PWM mode.
  2. m = 1 to 5

Only TSnCCR1 value rewritten, with output from TOSn0 and TOSn1  
(TSnOE0 = 1, TSnOE1 = 1, TSnOL0 = 0, and TSnOL1 = 0 for TSnIOC0 register)



**Figure 17-65 Basic operation timing in PWM mode (1/2)**

- Note**
1. D<sub>00</sub>: TSnCCR0 register setting value (0000H to FFFFH)  
D<sub>10</sub>, D<sub>11</sub>, D<sub>12</sub>, D<sub>13</sub>: TSnCCR1 register setting values (0000H to FFFFH)
  2. TOSn1 (PWM) duty = (TSnCCR1 register setting value) × (count clock cycle)  
TOSn1 (PWM) cycle = (TSnCCR0 register setting value + 1) × (count clock cycle)
  3. TOSn0 is toggled immediately after count start and with (TSnCCR0 register setting value + 1) × (count clock cycle)
  4. The TOSn2 to TOSn5 pins have the same functions as those of the TOSn1 pin.



Only TSnCCR1 value rewritten, with output from TOSn0 and TOSn1  
(TSnOE0 = 1, TSnOE1 = 1, TSnOL0 = 0, and TSnOL1 = 0 for TSnIOC0 register)

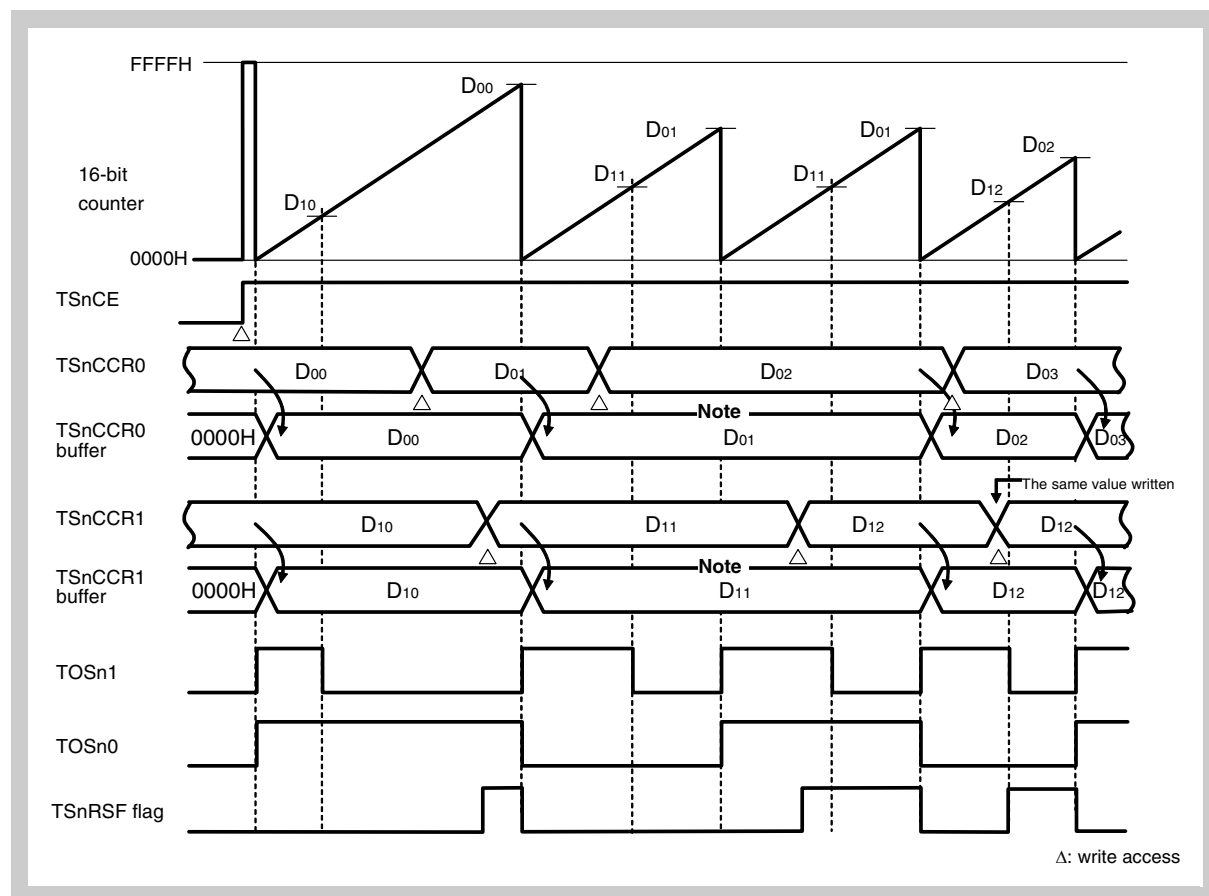


Figure 17-66 Basic operation timing in PWM mode (2/2)

- Note**
1. Since the TSnCCR1 register has not been written, this value is not reloaded to the TSnCCR0 buffer register.
  2. D<sub>00</sub>, D<sub>01</sub>, D<sub>02</sub>, D<sub>03</sub>: TSnCCR0 register setting values (0000H to FFFFH)  
D<sub>10</sub>, D<sub>11</sub>, D<sub>12</sub>, D<sub>13</sub>: TSnCCR1 register setting values (0000H to FFFFH)
  3. The TOSn0 and TOSn1 pins become active level when the timer starts counting.
  4. The TOSn2 to TOSn5 pins have the same functions as those of the TOSn1 pin.

**(2) Reload/interrupt thinning out functions in PWM mode**

The reload thinning out function and interrupt thinning out function are enabled by setting  $TSnRTE = 1$ ,  $TSnICE = 1$ , and  $TSnRDE = 1$  for the  $TSnOPT1$  register, and  $TSnID4$  to  $TSnID0$ .

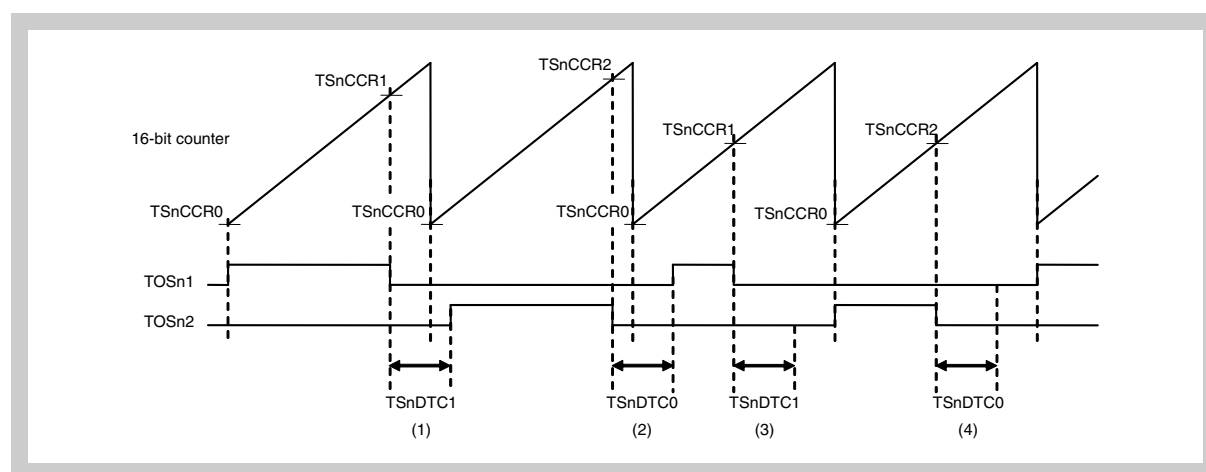
The interrupt thinning out function is enabled by setting  $TSnRTE = 1$ ,  $TSnRDE = 0$ , and  $TSnID4$  to  $TSnID0$ .

**(3) Dead time control in PWM mode**

In the PWM mode, dead time control is enabled by setting the dead time value to the  $TSnDTC0$  and  $TSnDTC1$  registers and setting  $TSnDSE$  of the  $TSnOPT0$  register to 1. The dead time can be controlled at the timing of switching the  $TOSn1$ ,  $TOSn2$ ,  $TOSn3$  and  $TOSn4$  pins.

**Table 17-33 Dead time in PWM mode**

Switch timing	Dead time
$TOSn1$ : High level to low level $TOSn2$ : Low level to high level	$TSnDTC1$ register value
$TOSn2$ : High level to low level $TOSn1$ : Low level to high level	$TSnDTC0$ register value
$TOSn3$ : High level to low level $TOSn4$ : Low level to high level	$TSnDTC1$ register value
$TOSn4$ : High level to low level $TOSn3$ : Low level to high level	$TSnDTC0$ register value

**Figure 17-67 Dead time control between  $TOSn1$  and  $TOSn2$  pins (1/2)**

The dead time counter starts at the falling edge of  $TOSn1$  during the period of (1). Then,  $TOSn2$  normally becomes active when the 16-bit counter becomes 0000H, but since the dead time counter is operating, it becomes active when the dead time count operation is finished.

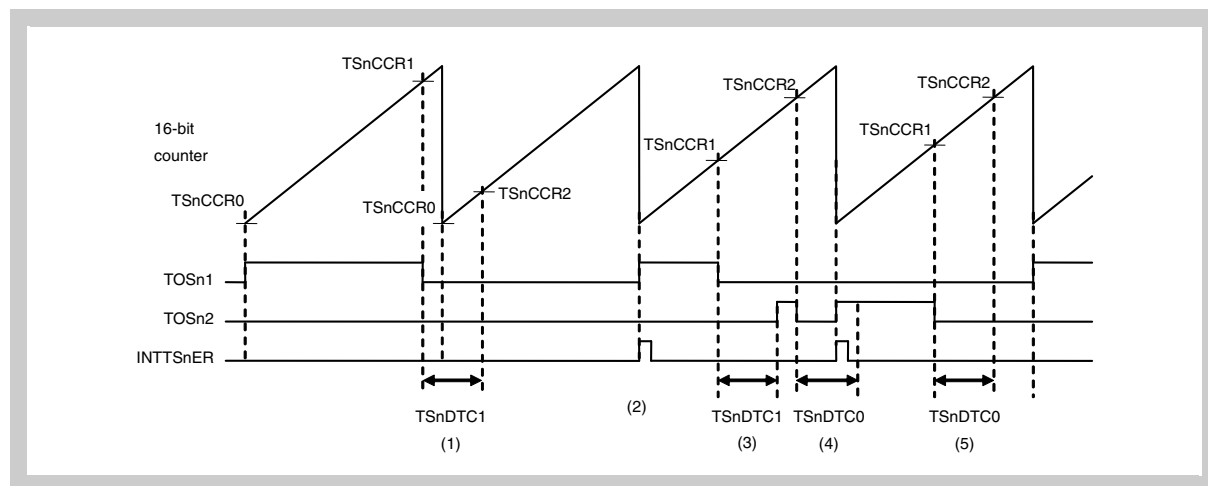
The dead time counter starts at the falling edge of  $TOSn2$  during the period of (2). Then,  $TOSn1$  normally becomes active when the 16-bit counter becomes 0000H, but since the dead time counter is operating, it becomes active when the dead time count operation is finished.

The dead time counter starts at the falling edge of  $TOSn1$  during the period of (3). Then,  $TOSn2$  becomes active when the 16-bit counter becomes 0000H after the dead time counter finishes the count operation.

The dead time counter starts at the falling edge of  $TOSn2$  during the period of

(4). Then, TOSn1 becomes active when the 16-bit counter becomes 0000H after the dead time counter finishes the count operation.

- Note**
1. The active level of TOSn1 and TOSn2 is high level.
  2. The active level of TOSn3 and TOSn4 is high level.



**Figure 17-68** Dead time control between TOSn1 and TOSn2 pins (2/2)

The dead time counter starts at the falling edge of TOSn1 during the period of (1). Then, TOSn2 normally becomes active when the 16-bit counter becomes 0000H, but since the dead time counter is operating, it becomes active when the dead time count operation is finished. In this case, however, TOSn2 becomes inactive upon a compare match, because a TSnCCR2 compare match occurs before the dead time counter operation finishes.

$TSnCCR1 + TSnDTC1 \geq TSnCCR0 + TSnCCR2$  (TOSn2 remains inactive).  
 $TSnCCR2 + TSnDTC0 \geq TSnCCR0 + TSnCCR1$  (TOSn1 remains inactive).

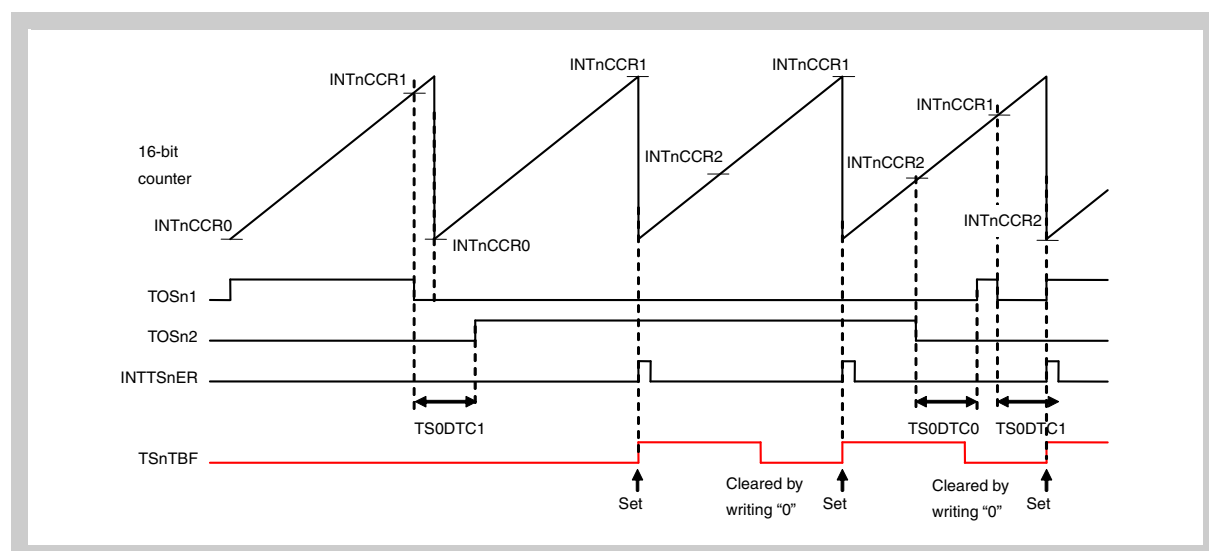
TOSn1 becomes active during the period of (2) because dead time control is not in operation. In this case, however, an INTTSnER interrupt occurs because TSnCCR1 and TSnCCR2 are set so that TOSn1 and TOSn2 rise simultaneously.

The dead time counter starts at the falling edge of TOSn1 during the period of (3). TOSn2 becomes active when the dead time counter operation is finished. TOSn2 becomes inactive upon a match of TSnCCR2.

The dead time counter starts at the falling edge of TOSn2 during the period of (4). Then, the 16-bit counter is cleared and the next PWM cycle is output. TOSn2 becomes active because TOSn1 is in the dead time period. In this case, however, an INTTSnER interrupt occurs because TSnCCR1 and TSnCCR2 are set so that TOSn1 and TOSn2 rise simultaneously.

The dead time counter starts at the falling edge of TOSn2 during the period of (5). TOSn1 does not become active even after the dead time counter operation is finished, because TSnCCR1 is smaller than TOSn2 and is already inactive.

- Note**
1. The active level of TOSn1 and TOSn2 is high level.
  2. The active level of TOSn3 and TOSn4 is high level.



**Figure 17-69 Duty 100% output**

As shown in *Figure 17-69 on page 784*, when TOSn2 is set to make duty 100% output ( $TSnCCR2 \geq TSnCCR0 + 1$ ), the TOSn1 output is fixed at the low level. This means that the control masks the TOSn1 active condition, because TOSn2 is already active when TOSn1 becomes active. In this case, an INTTSnER interrupt occurs because of the setting to make TOSn1 and TOSn2 high level simultaneously.

- Note**
1. The active level of TOSn1 and TOSn2 is high level.
  2. The active level of TOSn3 and TOSn4 is high level.

#### (4) Dead time rewriting by PWM mode operation

In the PWM mode, timer Sn dead time setting registers (TSnDTC0, TSnDTC1) can be rewritten during count operation. The new setting for the dead time becomes valid when the value is reloaded. The value cannot be changed with the anytime rewrite method.

Change of the dead time will be valid at the culled reload timing when the reload thinning out function (TS0RDE bit of TSnOPT1 register = 1) is used in the PWM mode.

TS0CCR1 must be written while the TS0RDE bit of the TSnOPT1 register is 1 in order to enable the reload timing.

**(5) Other timer outputs in PWM mode****(a) TOSn5 pin**

PWM output can be performed in the same manner as the TOSn1 to TOSn4 pins by setting the TSnCCR5 register.  
The dead time cannot be set.

**(b) TOSn0 and TOSn6 pins**

The TOSn0 pin repeats toggle operation in every PWM cycle.  
The TOSn6 pin outputs the inactive level.

**(c) INTTSnER interrupt**

The INTTSnER interrupt occurrence conditions vary depending on the TSnDSE bit setting in the TSnOPT0 register. TSnTBF is set to 1 when an interrupt occurs.

TSnDSE of TSnOPT0 register	TSnTBF of TSnOPT6 register	INTTSnER
0 (Dead time control disabled)	Set when timer outputs become active simultaneously	An error interrupt is output while timer outputs become active simultaneously and output active simultaneously.
1 (Dead time control enabled)	Set when timer outputs become active simultaneously	An error interrupt is output when a condition that sets timer outputs simultaneously occurs.

When the TSnDSE bit of the TSnOPT0 register is 1, or the TSnTBA0 and TSnTBA1 bits of the TSnIOC4 register are 1, occurrence of a condition that sets positive/negative phase simultaneously for the TOSn1 and TOSn3 pins and the TOSn2 and TOSn4 pins is the error interrupt occurrence condition. Therefore, an error interrupt occurs when a positive/negative phase simultaneous setting condition occurs even when TSnOE1 and TSnOE2 of the TSnIOC0 register are set to 0 and the TOSn1 and TOSn2 pins are fixed at the inactive level. Moreover, the positive/negative phase simultaneous active detection flag is also set.  
Clear TSnTBA0 and TSnTBA1 to 0 to prevent the above situation.

### 17.10.6 Free-running mode

In the free-running mode, the 16-bit counter counts from 0000H to FFFFH when the TSnCE bit of the TSnCTL0 register is set to 1. When the 16-bit counter overflows from FFFFH to 0000H, the overflow flag (TSnOVF) is set to 1, an overflow interrupt (INTTSnOV) occurs, and the count is then cleared. Count operation in the free-running mode continues until TSnCE is cleared to 0. A compare match interrupt (INTTSnCC0 to INTTSnCC5) occurs upon a match between the values of the 16-bit counter and the TSnCCR0 to TSnCCR5 registers during the count operation.

The TSnCCR0 to TSnCCR5 registers can be rewritten with the anytime rewrite method regardless of the timer Sn operation (TSnCE bit value).

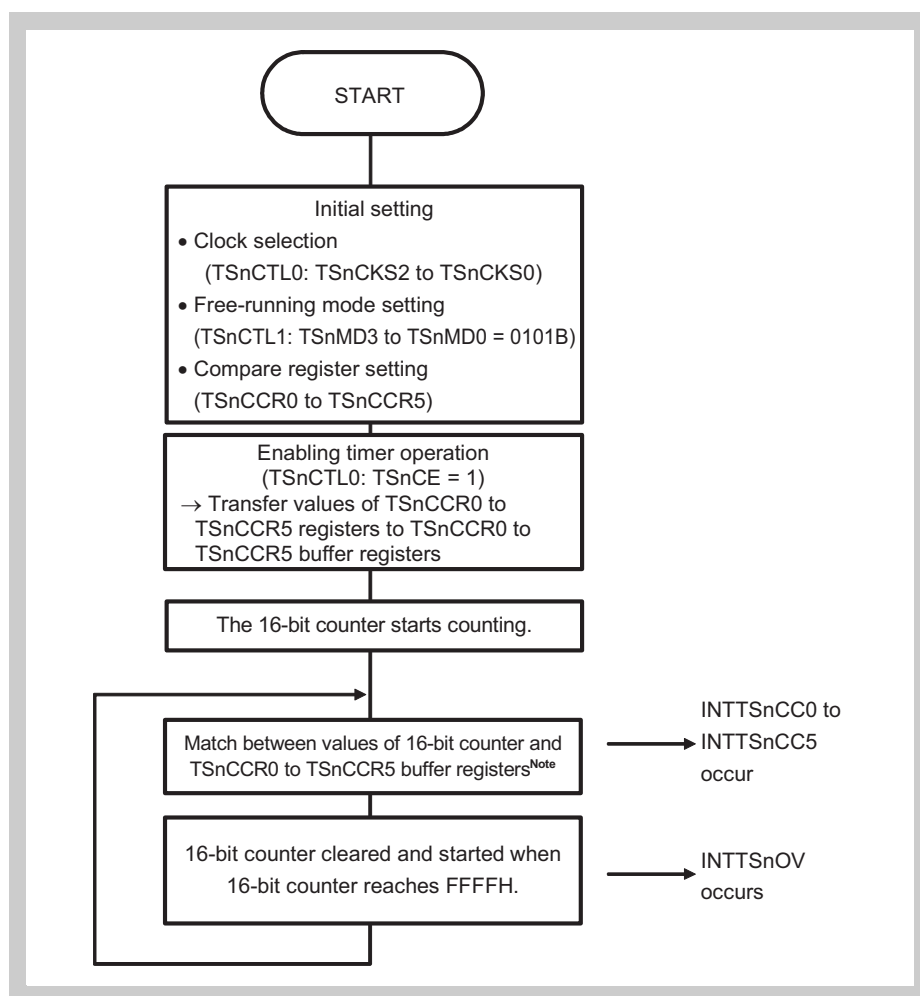
TOSn0 to TOSn5 become active immediately after the operation starts and perform toggle output when a match interrupt with the compare register occurs.

---

**Caution** The overflow flag (TSnOVF) is not cleared only by being read, but cleared when 0 is written or the operation stops (TSnCE = 0).

---

**Note** The free-running mode will be valid when TSnMD3 to TSnMD0 of the TSnCTL1 register are set to [0, 1, 0, 1].



**Figure 17-70 Basic operation flow in free-running mode**

**Note** The 16-bit counter is not cleared upon a match between the values of the 16-bit counter and the TSnCCR0 to TSnCCR5 buffer registers.

**(1) Free-running mode operation list****(a) Register rewriting**

Register	Rewriting method	Rewriting during operation	Function
TSnCCR0	Anytime rewrite	Possible	Compare value
TSnCCR1 to TSnCCR3	Anytime rewrite	Possible	Compare value
TSnCCR4, TSnCCR5	Anytime rewrite	Possible	Compare value

**(b) Input pin**

Pin	Function
TTRGSn	-
TEVTSn	-

**(c) Output pin**

Pin	Function
TOSn0 to TOSn5	Toggle output by compare match of TSnCCR0 to TSnCCR5 registers
TOSn6, TOSn7	-

**(d) Interrupt**

Interrupt	Function
INTTSnCC0 to INTTSnCC5	Compare match of TSnCCR0 to TSnCCR5 register
INTTSnOV	Overflow
INTTSnER	-
INTTSnOD	-
INTTSnCD0	-
INTTSnWN	-

**(e) Compare match timing**

Compare match	Timing
TSnCCR0	Timing of switching 16-bit counter from TSnCCR0 to 0000H
TSnCCR1 to TSnCCR5	Timing of switching 16-bit counter from "TSnCCR1 to TSnCCR5" to "TSnCCR1 to TSnCCR5 + 1"

**Note** “-” indicates an unused function in the free-running mode.



TSnCCR0, TSnCCR1 register values rewritten, with output from TOSn0 and TOSn1  
(TSnOE0 = 1, TSnOE1 = 1, TSnOL0 = 0, and TSnOL1 = 0 for TSnIOC0 register)

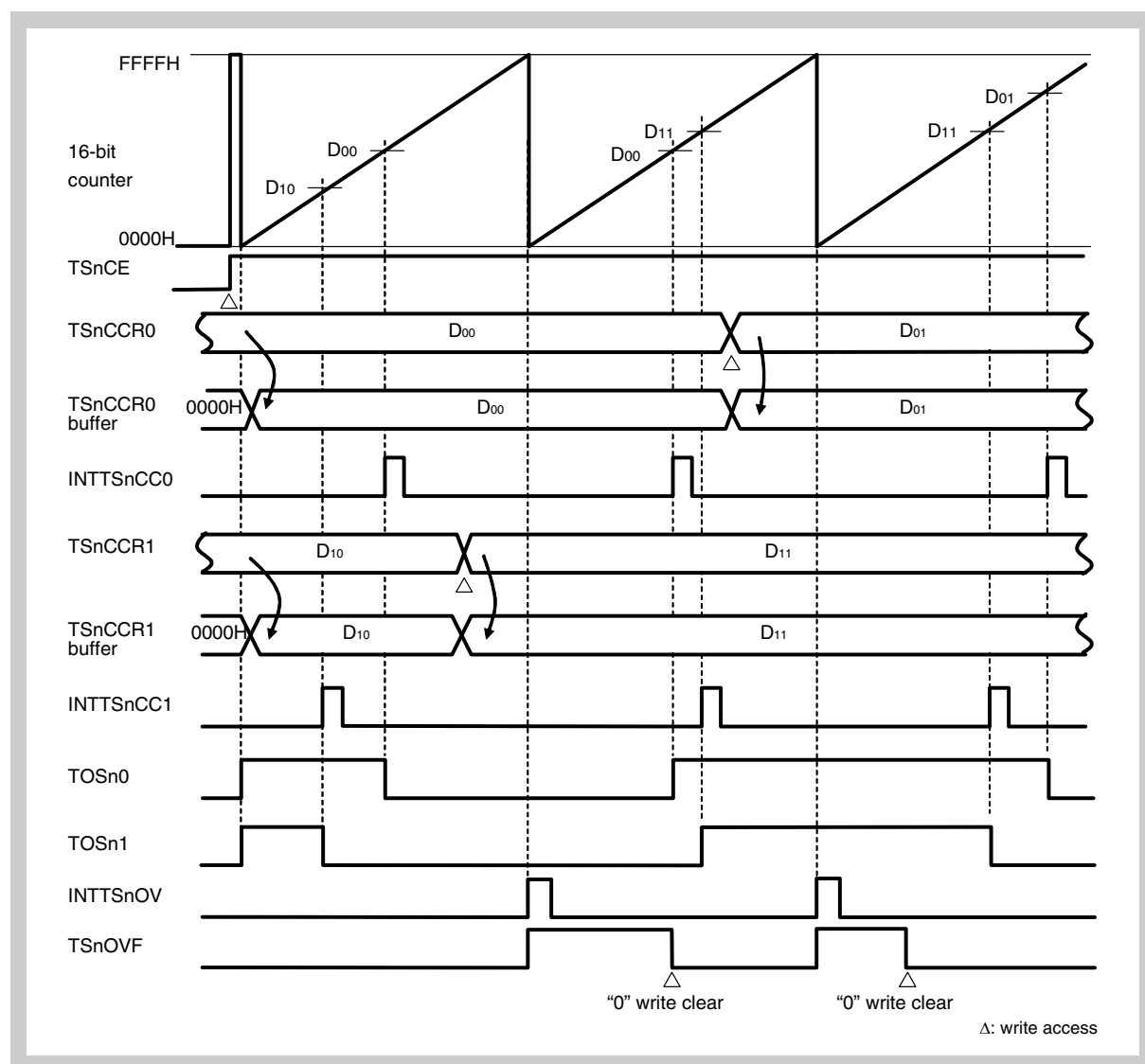


Figure 17-71 Basic operation flow in free-running mode

- Note**
1. D<sub>00</sub>, D<sub>01</sub>: TSnCCR0 register setting values (0000H to FFFFH)  
D<sub>10</sub>, D<sub>11</sub>: TSnCCR1 register setting values (0000H to FFFFH)
  2. The TOSn0 and TOSn1 pins become active level when the timer starts counting.
  3. The TOSn2 to TOSn5 pins have the same functions as those of the TOSn1 pin.

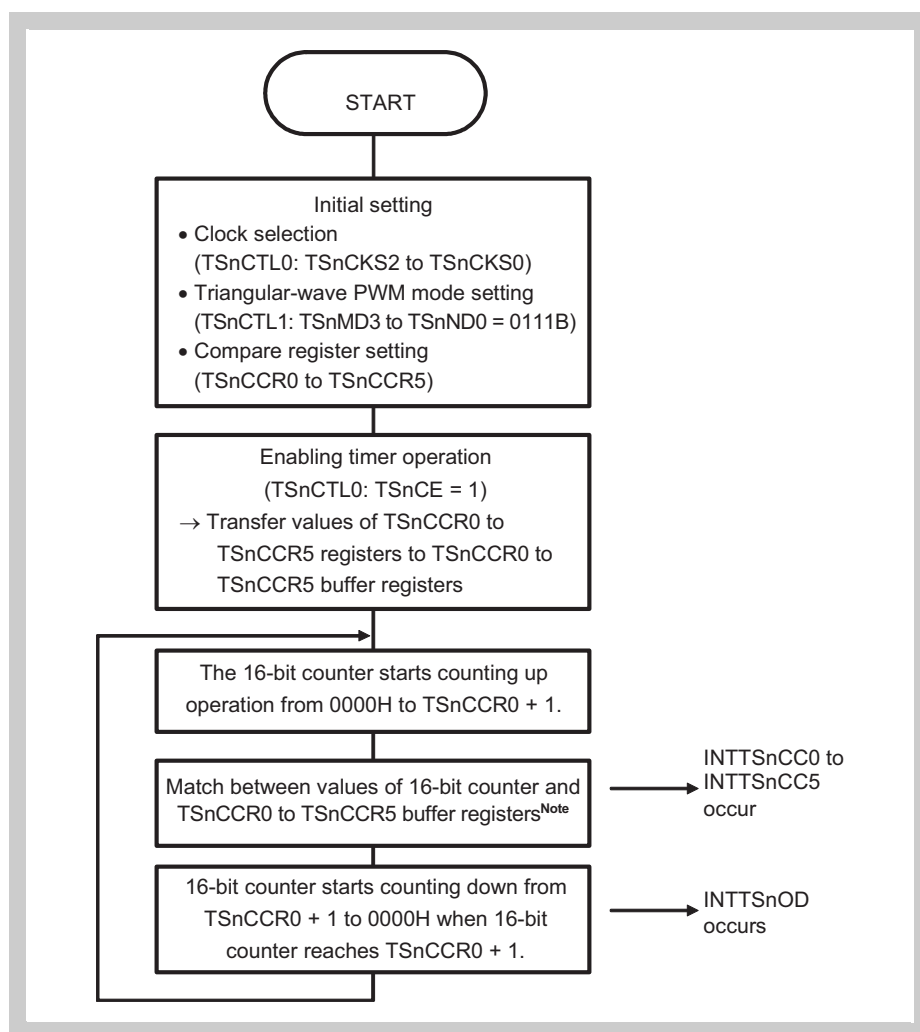
### 17.10.7 Triangular-wave PWM mode

In the triangular-wave PWM mode, when a duty is set with the TSnCCR1 to TSnCCR5 registers, a cycle is set with the TSnCCR0 register and the TSnCE bit of the TSnCTL0 register is then set to 1, duty variable type triangular-wave PWM output is provided from the TOSn1 to TOSn5 pins. TOSn0 performs toggle output upon a match of the TSnCCR0 buffer register and when a 16-bit counter underflow occurs. A compare match interrupt (INTTSnCC0 to INTTSnCC5) occurs upon a match between the values of the 16-bit counter and the TSnCCR0 to TSnCCR5 registers during count operation. Also, a trough interrupt (INTTSnOD) occurs when a 16-bit counter underflow occurs. The TSnCCR0 to TSnCCR5 registers can be rewritten during count operation. Compare register values are reloaded when a 16-bit counter underflow occurs.

Writing to the TSnCCR1 register enables the next reload timing.

In the triangular-wave PWM mode, the capture function cannot be used because the function of the TSnCCR0 to TSnCCR5 registers is fixed as a compare register.

- Note**
1. Set a value of  $0 \leq \text{TSnCCR0} \leq \text{FFFEH}$  to the TSnCCR0 register in the triangular-wave PWM mode.
  2. Set the TSnRBE bit of the TSnOPT1 register to 1 in the triangular-wave PWM mode. Reload does not occur by clearing TSnRBE to 0.
  3. The triangular-wave PWM mode will be valid when TSnMD3 to TSnMD0 of the TSnCTL1 register are set to [0, 1, 1, 1].



**Figure 17-72 Basic operation flow in triangular-wave PWM mode**

**Note** The 16-bit counter is not cleared upon a match between the values of the 16-bit counter and the TSnCCR0 to TSnCCR5 buffer registers.

**(1) Triangular-wave PWM mode operation list****(a) Register rewriting**

Register	Rewriting method	Rewriting during operation	Function
TSnCCR0	Reload	Possible	1/2 of cycle
TSnCCR1 to TSnCCR3	Reload	Possible	1/2 of duty
TSnCCR4, TSnCCR5	Reload	Possible	1/2 of duty

**(b) Input pin**

Pin	Function
TTRGSn	-
TEVTSn	-

**(c) Output pin**

Pin	Function
TOSn0	Inactive during 16-bit counter count-up operation Active during count-down operation.
TOSn1 to TOSn5	PWM output by compare match of TSnCCR1 to TSnCCR5 registers
TOSn6	-
TOSn7	Pulse output by A/D conversion trigger

**(d) Interrupt**

Interrupt	Function
INTTSnCC0 to INTTSnCC5	Compare match of TSnCCR0 to TSnCCR5 registers
INTTSnOV	-
INTTSnER	Error
INTTSnOD	Trough interrupt
INTTSnCD0	-
INTTSnWN	-

**(e) Compare match timing**

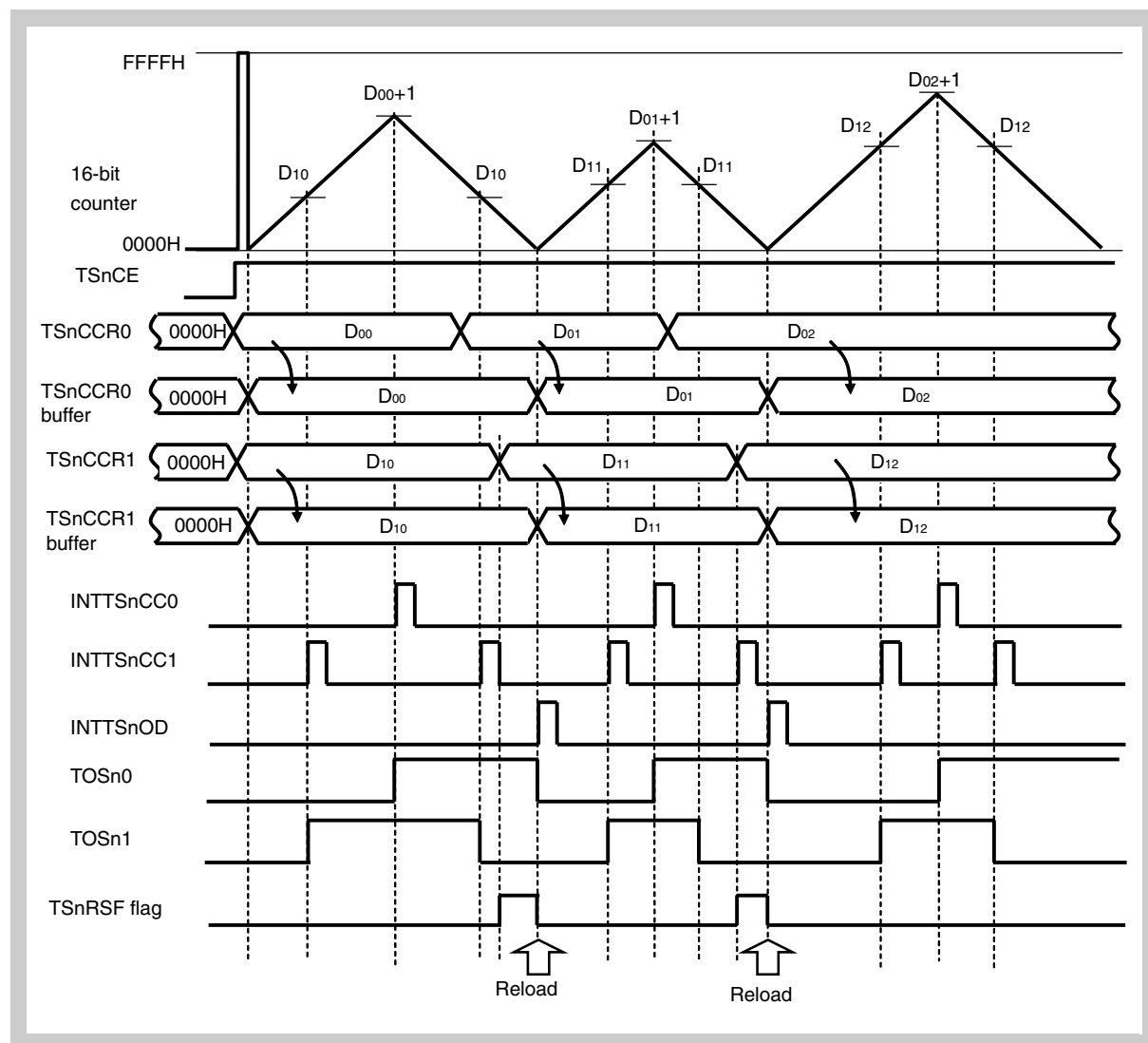
Compare match	Timing
TSnCCR0	Timing of switching 16-bit counter from TSnCCR0 to TSnCCR0 – 1
TSnCCR1 to TSnCCR5	Timing of switching 16-bit counter from “TSnCCR1 to TSnCCR5” to “TSnCCR1 to TSnCCR5 + 1”

**(f) Output condition**

Output	Condition
Duty 0% output	$TSnCCRM = TSnCCR0 + 1$
Duty 100% output	$TSnCCRM = 0000H$
Toggle at peak/trough	$TSnCCRM \geq TSnCCR0 + 2$

- Note**
1. “-” indicates an unused function in the triangular-wave PWM mode.
  2.  $m = 1$  to 5

With output from TOSn0 and TOSn1  
(TSnOE0 = 1, TSnOE1 = 1, TSnOL0 = 0, and TSnOL1 = 0 for TSnIOC0 register)



**Figure 17-73 Basic operation timing in triangular-wave PWM mode**

- Note**
1.  $D_{00}, D_{01}, D_{02}$ : Triangular-wave cycle  $(TSnCCR0 + 1) \times 2$   
 $D_{10}, D_{11}, D_{12}$ : PWM duty  $(TSnCCR0 - TSnCCR1 + 1) \times 2$   
 Count-up: 0000H to  $D_{00}$   
 Count-down:  $(D_{00} - 0001H)$
  2. The TOSn2 to TOSn5 pins have the same functions as those of the TOSn1 pin.

**(2) Reload/interrupt thinning out function in triangular-wave PWM mode**

The reload thinning out function and interrupt thinning out function are enabled by setting TSnRTE = 1, TSnICE = 1, and TSnRDE = 1 in the TSnOPT1 register, and TSnID4 to TSnID0.

The interrupt thinning out function is enabled by setting TSnRTE = 1, TSnRDE = 0, and TSnID4 to TSnID0.

### 17.10.8 High-accuracy T-PWM mode

In the high-accuracy T-PWM mode, 6-phase PWM is generated by using the 16-bit counter (up/down,  $\pm 2$  counts, real 15-bit) and four 16-bit compare registers (LSB = additional pulse control bit).

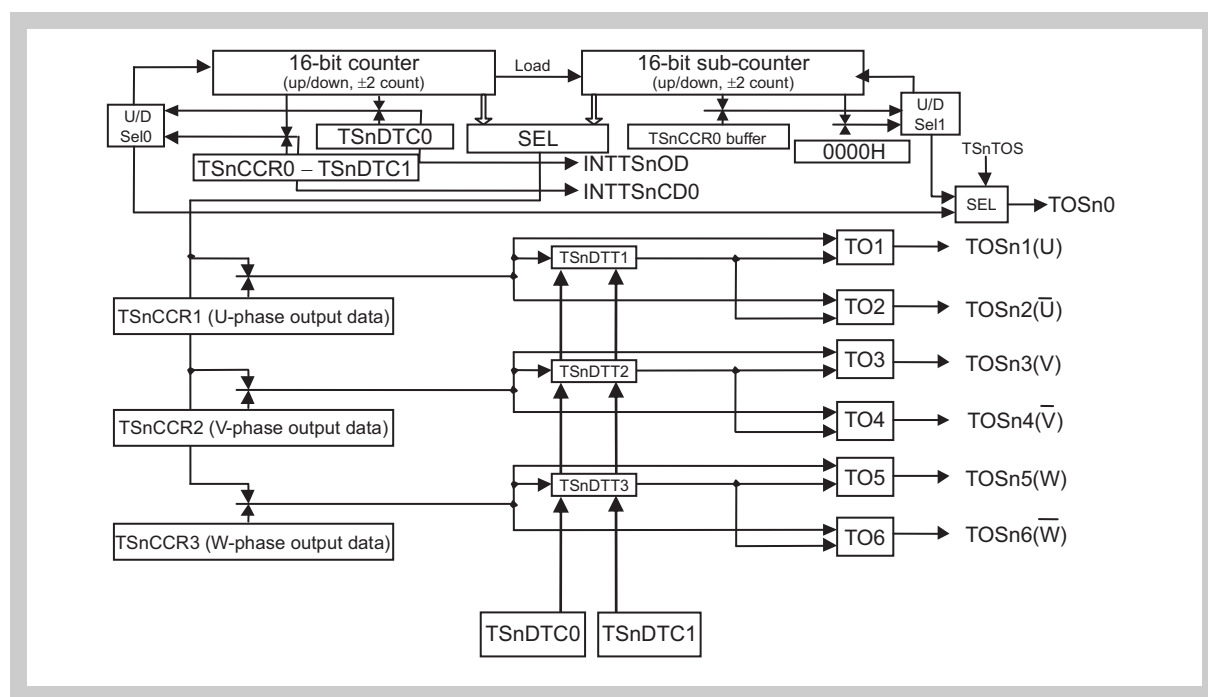
The carrier wave value calculated with “TSnCCR0 – TSnDTC0 – TSnDTC1” is set to the TSnCCR0 register. A duty of U, V, and W-phase voltage data signal is set with the TSnCCR1 to TSnCCR3 registers. A dead time is set with the TSnDTC0 and TSnDTC1 registers. The TSnDTC0 register can set the dead time of negative phase (OFF) to positive phase (ON). TSnDTC1 can sets the dead time of positive phase (OFF) to negative phase (ON).

The 16-bit counter performs count-up operation using the TSnDTC0 register value as the minimum value, and count-down operation upon a match with the maximum value indicated by TSnCCR0 – TSnDTC1.

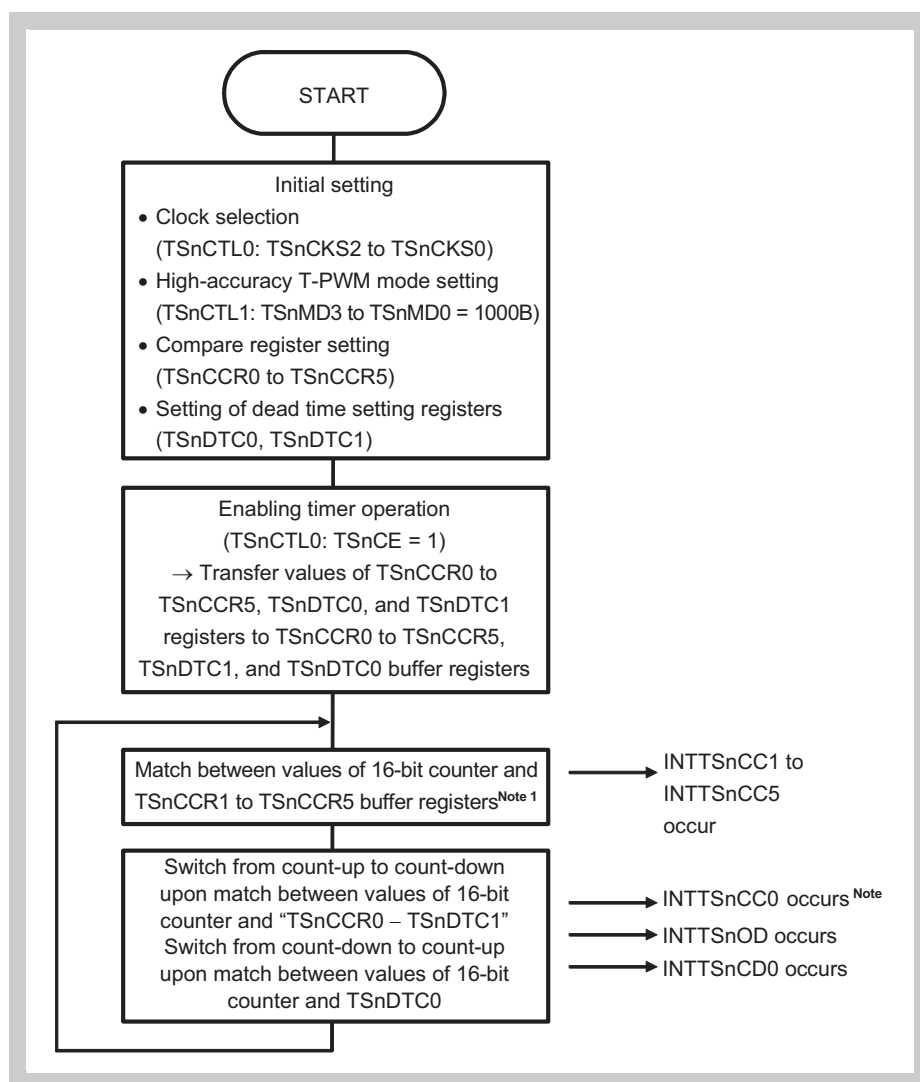
The 10-bit dead time counters (TSnDTC0 to TSnDTC3) reloads the value set to the TSnDTC0 and TSnDTC1 registers upon a match between the values of the TSnDTC0 to TSnDTC3 and TSnCCR0 to TSnCCR3 registers to perform count-down operation.

A compare match interrupt signal (INTTSnCC1 to INTTSnCC5) occurs upon a match between the values of the 16-bit counter and the TSnCCR1 to TSnCCR5 registers.

**Note** The high-accuracy T-PWM mode will be valid when TSnMD3 to TSnMD0 of the TSnCTL1 register are set to [1, 0, 0, 0].



**Figure 17-74** Block diagram of high-accuracy T-PWM mode



**Figure 17-75 Basic operation flow in high-accuracy T-PWM mode**

- Note**
1. The 16-bit counter is not cleared upon a match between the values of the 16-bit counter and the TSnCCR0 to TSnCCR5 buffer registers.
  2. Only when TSnDTC = 0000H



**(1) High accuracy T-PWM mode operation list****(a) Register rewriting**

Register	Rewriting method	Rewriting During Operation	Function
TSnCCR0	Reload/anytime rewrite	Possible	Cycle
TSnCCR1 to TSnCCR3	Reload/anytime rewrite	Possible	PWM duty
TSnCCR4, TSnCCR5	Reload/anytime rewrite	Possible	PWM duty (selectable as A/D conversion trigger)
TSnDTC0, TSnDTC1	Reload	Conditionally possible <sup>Note 1</sup>	Cycle, dead time

**(b) Input pin**

Pin	Function
TTRGSn	-
TEVTSn	-

**(c) Output pin**

Pin	Function
TOSn0	Inactive during counting up, active during counting down by 16-bit counter or 16-bit sub-counter
TOSn1	PWM output (with dead time) upon a TSnCCR1 compare match
TOSn2	Negative phase output for TOSn1 <sup>Note 2</sup>
TOSn3	PWM output (with dead time) upon a TSnCCR2 compare match
TOSn4	Negative phase output for TOSn3 <sup>Note 2</sup>
TOSn5	PWM output (with dead time) upon a TSnCCR3 compare match
TOSn6	Negative phase output for TOSn5 <sup>Note 2</sup>
TOSn7	Pulse output by A/D conversion trigger

- Note**
1. Refer to “Rewriting of TSnDTC0 and TSnDTC1 registers” on page 827 for details.
  2. The output level of TOSn2, TOSn4, and TOSn6 will be switched to active level by setting TSnCE of the TSnCTL0 register to 1 while TSnOE2, TSnOE4, and TSnOE6 are 1, and TSnOL2, TSnOL4, and TSnOL6 are 0 in the TSnIOC0 register.
  3. “-” indicates an unused function in the high-accuracy T-PWM mode.

**(d) Interrupt**

Interrupt	Function
INTTSnCC0	INTTSnCC0 compare match <sup>Note 1</sup>
INTTSnCC1 to INTTSnCC5	TSnCCR1 to TSnCCR5 compare match
INTTSnOV	Overflow <sup>Note 2</sup>
INTTSnER	Error
INTTSnOD	Trough interrupt
INTTSnCD0	Peak interrupt
INTTSnWN	Warning interrupt

**(e) Compare match timing**

Compare match	Timing
TSnCCR0	Timing of switching 16-bit counter from TSnCCR0 to TSnCCR0 – 2
TSnCCR1 to TSnCCR5	Timing of switching 16-bit counter from “TSnCCR1 to TSnCCR5” to “TSnCCR1 to TSnCCR5 + 1”

**(f) Output condition**

Output	Condition
Duty 0% output	TSnCCRm = TSnCCR0
Duty 100% output	TSnCCRm = 0000H

- Note**
1. Only when TSnDTC1 = 0000H
  2. When the TSnCCR0, TSnDTC0, and TSnDTC1 registers are incorrectly set
  3. “–” indicates an unused function in the high-accuracy T-PWM mode.
  4. m = 1 to 3

## (2) Various setting in high-accuracy T-PWM mode

### (a) Mode setting

The high-accuracy T-PWM mode is set by setting TSnMD3 to TSnMD0 of the TSnCTL1 register to [1, 0, 0, 0].

### (b) Output level/output enable setting

Output level/output is enabled by setting the TSnOL0 to TSnOL7 and TSnOE0 to TSnOE7 bits of the TSnIOC0 register.

The TOSn0 pin outputs the status of count-up/down operation by the 16-bit counter and 16-bit sub-counter. The 16-bit counter and 16-bit sub-counter is switched with the TSnTOS bit of the TSnOPT7 register.

The TOSn7 pin outputs pulses by A/D conversion trigger. Set this as necessary.

### (c) Enabling error interrupt occurrence

Occurrence of an error interrupt (INTTSnER) when positive/negative phase simultaneous active is detected is enabled by setting the TSnEOC bit of the TSnIOC4 register to 1.

In the high-accuracy T-PWM mode, the positive/negative phases do not become active simultaneously by setting any value to the TSnCCR0 to TSnCCR3 registers.

### (d) Register rewrite timing setting with reload function

Reload (batch rewrite) or anytime rewrite (default "0" reload) will be executed for registers with the reload function by using the TSnCMS bit of the TSnOPT0 register. Be sure to set the TSnRTE or TSnRBE bit of the TSnOPT1 register to 1 to execute the reload operation.

No reload timings will occur when both the TSnRTE and TSnRBE bits are 0. In case of anytime rewrite, unexpected output may occur depending on the rewrite timing.

Refer to (a) to (c) in *"Anytime rewrite mode"* on page 693 when using the anytime rewrite function.

### (e) Interrupt and thinning out function setting

The interrupt and thinning out functions can be set with the TSnOPT1 register. Set the TSnICE bit to 1 to generate peak interrupts (INTTSnCD0), and set the TSnIOE bit to 1 to generate trough interrupts (INTTSnOD).

To use the thinning out function for peak/trough interrupts, set the TSnID4 to TSnID0 bits.

### (f) Reload thinning out function setting

Reload timings can be set to the same timing as interrupt timings by setting the TSnRDE bit of the TSnOPT1 register to 1.

### (g) A/D conversion trigger output setting

A/D conversion trigger 0 (TSnADTRG0 signal) can be set using the TSnAT07 to TSnAT00 bits of the TSnOPT2 register.

Enabling/disabling of A/D conversion trigger output upon a match with the TSnCCR5 register (16-bit counter count-up/down operation), a match with the TSnCCR4 register (16-bit counter count-up/down operation), at a peak interrupt (INTTSnCD0), or at a trough interrupt of the 16-bit counter or 16-bit sub-counter, is set with the TSnAT07 to TSnAT00 bits.

A/D conversion trigger 1 (TSnADTRG1 signal) can be set using the TSnAT17 to TSnAT10 bits of the TSnOPT3 register.

Enabling/disabling of A/D conversion trigger output upon a match with the

TSnCCR5 register (16-bit counter count-up/down operation), a match with TSnCCR4 register (16-bit counter count-up/down operation), at a peak interrupt (INTTSnCD0), or at a trough interrupt of the 16-bit counter or 16-bit sub-counter, is set with the TSnAT17 to TSnAT10 bits.

To set match timings for the TSnCCR4 and TSnCCR5 register, set the compare values to each register.

The thinning out function can be used for the TSnADTRG0 and TSnADTRG1 signals. No thinning out, 1-thinning out, 3-thinning out and 7-thinning out can be set with the TSnACC01 and TSnACC00 bits of the TSnOPT2 register, and the TSnACC11 and TSnACC10 bits of the TSnOPT3 register.

---

**Caution** Set the TSnOPT2, TSnOPT3, TSnCCR4, and TSnCCR5 registers correctly in order to output A/D conversion trigger timing pulse to the TOSn7 pin.

---

#### (h) Dead time setting

A dead time is set with the TSnDTC0 and TSnDTC1 registers.

A dead time can be calculated with the following expressions.

16-bit counter operation clock cycle  $\times$  TSnDTC0

16-bit counter operation clock cycle  $\times$  TSnDTC1

The time from inactive change of the TOSn2, TOSn4, and TOSn6 pins to active change of the TOSn1, TOSn3, and TOSn5 pins can be set with the TSnDTC0 register.

The time from inactive change of the TOSn1, TOSn3, and TOSn5 pins to active change of the TOSn2, TOSn4, and TOSn6 pins can be set with TSnDTC1 register.

#### (i) Carrier wave cycle

Set the carrier wave cycle with the TSnCCR0 register based on the following expressions.

$$\text{TSnCCR0} = (\text{Carrier-wave cycle}/16\text{-bit counter operation clock cycle}) + \text{TSnDTC1} + \text{TSnDTC0}$$

The value set to the TSnCCR0 register must satisfy the following conditions in view of the dead time.

$$\text{TSnCCR0} > 3 \times \text{MAX}(\text{TSnDTC0}, \text{TSnDTC1}) + \text{MIN}(\text{TSnDTC0}, \text{TSnDTC1})$$

$$0002\text{H} \leq \text{TSnCCR0} \leq \text{FFFEH}$$

TSnCCR0 must be an even number.

**Note** MAX (A, B) indicates the greater value of A and B, and MIN (A, B) indicates the smaller value of A and B.

**(j) Duty (PWM width) setting**

U, V and W-phase duties can be set with the TSnCCR1 to TSnCCR3 registers, respectively. The setting range of the TSnCCR1 to TSnCCR3 registers are as follows.

$$0000H \leq \text{TSnCCR1 to TSnCCR3} \leq \text{TSnCCR0} + 1$$

LSB (Least Significant Bit) of the TSnCCR1 to TSnCCR3 registers means the additional pulse setting.

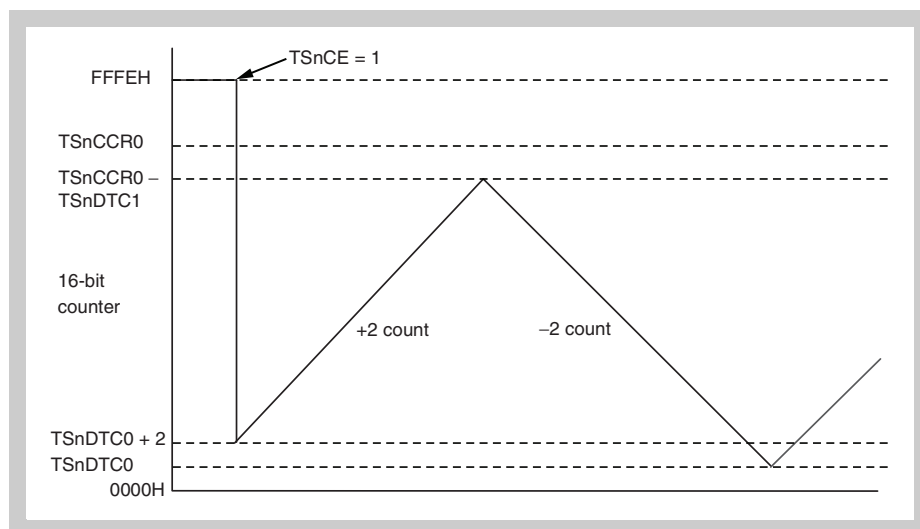
When TSnCCR1 = 0003H, a negative phase (TOSn2 pin) change delays by one count clock compared with that when TSnCCR1 = 0002H (during count-up operation by the 16-bit counter).

**Caution** Do not perform setting such that  $\text{TSnCCR0} + 2 < \text{TSnCCR1 to TSnCCR3}$ .

**(3) 16-bit counter operation in high-accuracy T-PWM mode**

The initial value of the 16-bit counter is FFFE<sub>H</sub>, the value “TSnDTC0 value + 2” is loaded immediately after the timer Sn operation is set to start (TSnCE of TSnCTL0 register = 1), and the 16-bit counter counts up in units of +2. Then, the 16-bit counter counts down in units of –2 at the match timing with “TSnCCR0 – TSnDTC1”.

The 16-bit counter operates as follows.



**Figure 17-76** 16-Bit counter operation in high-accuracy T-PWM mode

**Note** 16-bit counter minimum value: TSnDTC0  
 16-bit counter maximum value: TSnCCR0 – TSnDTC1  
 Carrier-wave cycle: (TSnCCR0 – TSnDTC0 – TSnDTC1) × count clock cycle

The initial value of the 16-bit sub-counter is FFFE<sub>H</sub>, and the value “TSnDTC0 value – 2” is loaded immediately after the timer Sn operation is set to start (TSnCE of TSnCTL0 register = 1). Then, the 16-bit counter counts down in units of –2 until the value matches with 0000<sub>H</sub> and starts counting up in units of +2 at the match timing.

The 16-bit counter value is loaded to the 16-bit sub-counter when the 16-bit counter operation is switched from counting up to counting down. The 16-bit sub-counter continues the count-up operation and starts counting down in units of –2 at the match timing with the TSnCCR0 register. The 16-bit counter value is loaded at the match timing with the TSnDTC0 register, and counting down is continued.

The 16-bit sub-counter operates as follows.

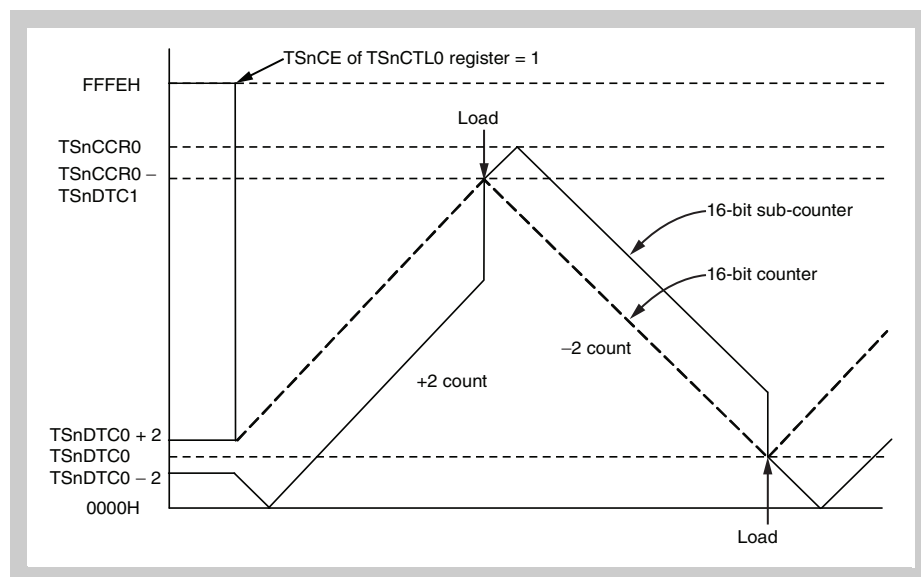


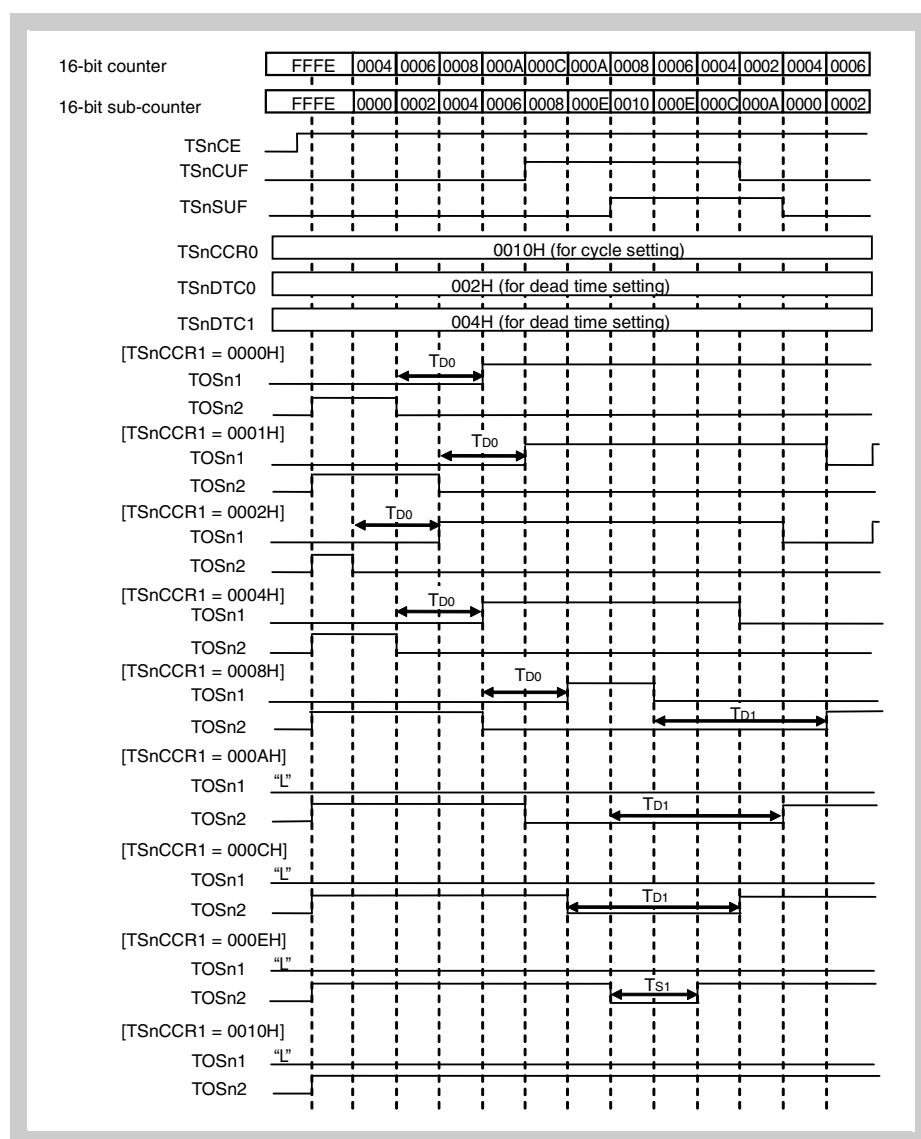
Figure 17-77 16-Bit sub-counter operation in high-accuracy T-PWM mode

**Note** 16-bit sub-counter minimum value: 0000<sub>H</sub>  
16-bit sub-counter maximum value: TSnCCR0

**(4) Basic operation in high-accuracy T-PWM mode****(a) Timer output example immediately after timer Sn operation starts**

The timing diagram when  $\text{TSnCCR0} = 0010\text{H}$ ,  $\text{TSnDTC0} = 0002\text{H}$ ,  $\text{TSnDTC1} = 0004\text{H}$ , and the  $\text{TSnCCR1}$  register =  $0000\text{H}$  to  $0010\text{H}$  (partly extracted) is shown below. In this example,  $\text{TSnOL1}$  to  $\text{TSnOL6}$  of the  $\text{TSnIOC0}$  register are set to  $000000\text{B}$ .

When  $\text{TSnCCR1} = \text{TSnDTC0}$ , the  $\text{TOSn2}$  pin output level changes after compare match. In case of  $\text{TSnCCR1} \leq \text{TSnDTC0} - 0002\text{H}$ , the  $\text{TOSn2}$  pin output level varies to the active level when the initial value of the count is loaded after the timer Sn operation starts ( $\text{TSnCE}$  of  $\text{TSnCTL0}$  register = 1). Change of the  $\text{TOSn2}$  pin output level delays by one count clock compared with the case of  $\text{TSnCCR1} = \text{TSnDTC0} - 0002\text{H}$ , because  $\text{TSnCCR1} = \text{TSnDTC0} - 0001\text{H}$  is an additional pulse.



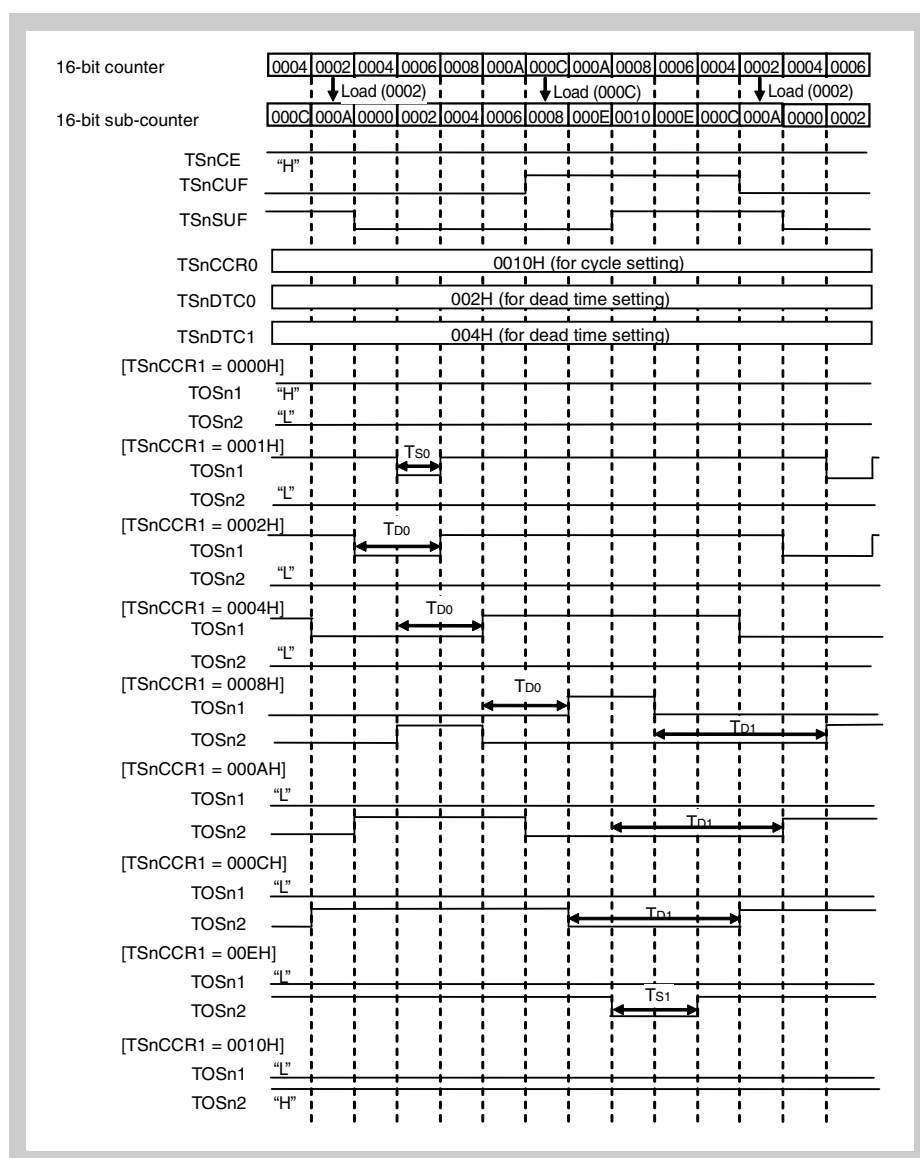
**Figure 17-78** Timer output example when  $\text{TSnCE} = 1$  (in high-accuracy T-PWM mode)

- Note**
1.  $\text{TSnCCR0} = 0010\text{H}$ ,  $\text{TSnDTC0} = 0002\text{H}$ ,  $\text{TSnDTC1} = 0004\text{H}$
  2.  $T_{D0}$ : Time depending on  $\text{TSnDTC0}$  register dead time setting  
 $T_{D1}$ : Time depending on  $\text{TSnDTC1}$  register dead time setting  
 $T_{S1}$ : Time determined by result of compare between 16-bit sub-counter and  $\text{TSnCCR1}$  when  $\text{TSnCCR1} > 16\text{-bit counter maximum value}$

**(b) Timer output example during timer Sn operation**

The timing diagram when  $TSnCCR0 = 0010H$ ,  $TSnDTC0 = 0002H$ ,  $TSnDTC1 = 0004H$ , and the  $TSnCCR1$  register =  $0000H$  to  $0010H$  (partly extracted) is shown below. In this example,  $TSnOL1$  to  $TSnOL6$  of the  $TSnIOC0$  register are set to  $000000B$ .

The active (high level) range output by the positive phase ( $TOSn1$  pin) is  $0000H \leq TSnCCR1 < TSnCCR0 - TSnDTC1 - TSnDTC0$  (additional pulse).  
The active (high level) range output by the negative phase ( $TOSn0$  pin) is  $TSnDTC0 + TSnDTC1 < TSnCCR1 \leq TSnCCR0$ .



**Figure 17-79** Timer output example during operation (in high-accuracy T-PWM mode)

- Note**
1.  $TSnCCR0 = 0010H$ ,  $TSnDTC0 = 0002H$ ,  $TSnDTC1 = 0004H$
  2.  $T_{D0}$ : Time depends on  $TSnDTC0$  register dead time setting  
 $T_{D1}$ : Time depends on  $TSnDTC1$  register dead time setting  
 $T_{S0}$ : Time determined by compare between 16-bit sub-counter and  $TSnCCR1$  when  $TSnCCR1 < 16$ -bit counter minimum value.  
 $T_{S1}$ : Time determined by result of compare between 16-bit sub-counter and  $TSnCCR1$  when  $TSnCCR1 > 16$ -bit counter maximum value.



**(5) Additional pulse control in high-accuracy T-PWM mode**

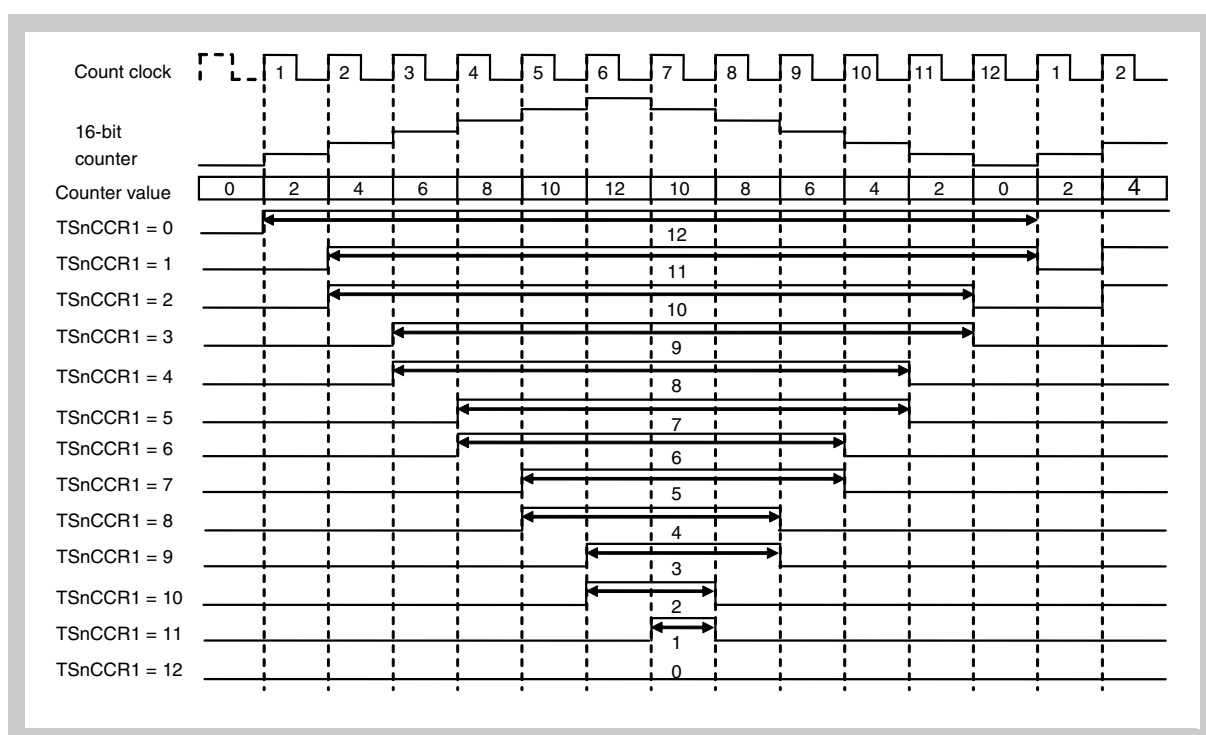
In the high-accuracy T-PWM mode, the additional pulse can be set by setting LSB of the duty setting registers (TSnCCR1 to TSnCCR3) to 1. The additional pulse control function enables finer duty control (with higher accuracy).

The examples of the TOSn1 pin output with/without additional pulse control are shown below, with the setting of TSnCCR0 = 12, and TSnDTC0, TSnDTC1 = 0.

**(a) Pulse output with additional pulse control**

In *Figure 17-80 on page 805*, additional pulse control is performed when an odd value is set to the TSnCCR1 register. Arrows and figures indicate the duty range of the TOSn1 pin output in one cycle.

As shown in *Figure 17-80 on page 805*, the TOSn1 pin output range (duty ratio) can be controlled by 1 count clock from 12-clock to 0-clock range when additional pulse control is performed.



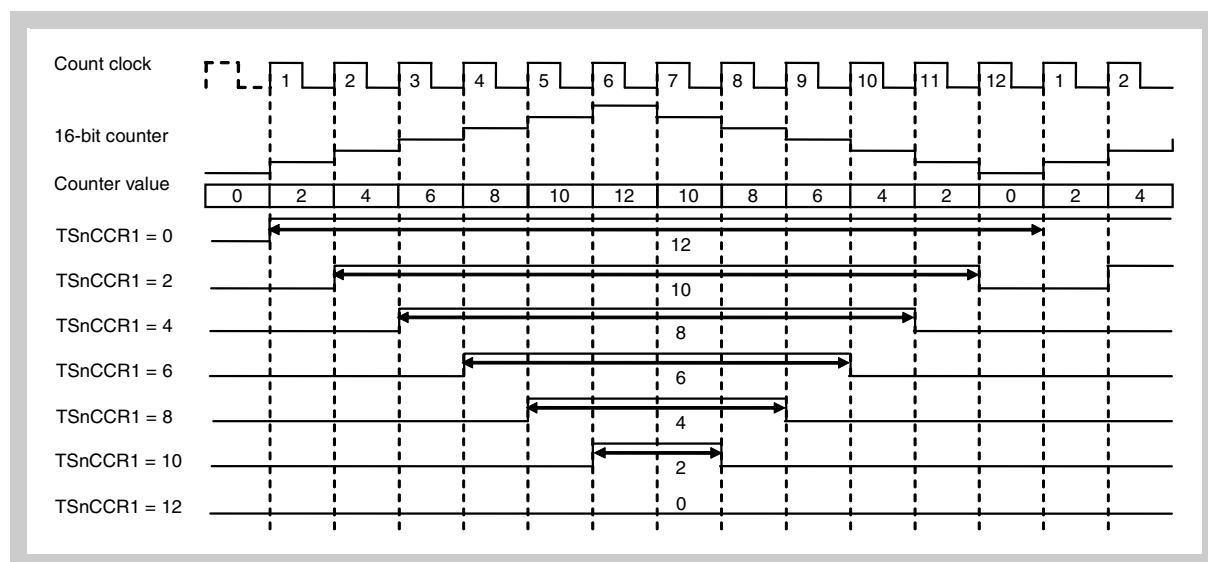
**Figure 17-80** TOSn1 pin output example with additional pulse control

**Note** TSnCCR0 = 12, TSnDTC0 = 0, TSnDTC1 = 0

**(b) Pulse output without additional pulse control**

In Figure 17-81 on page 806, arrows and figures indicate the duty range of the TOSn1 pin output in one cycle.

The TOSn1 pin output range is controlled by 2 count clocks from 12-clock to 0-clock range if additional pulse control is not performed. In this case, duty variation volume becomes larger than that when additional pulse control is performed.



**Figure 17-81** TOSn1 pin output example without additional pulse control

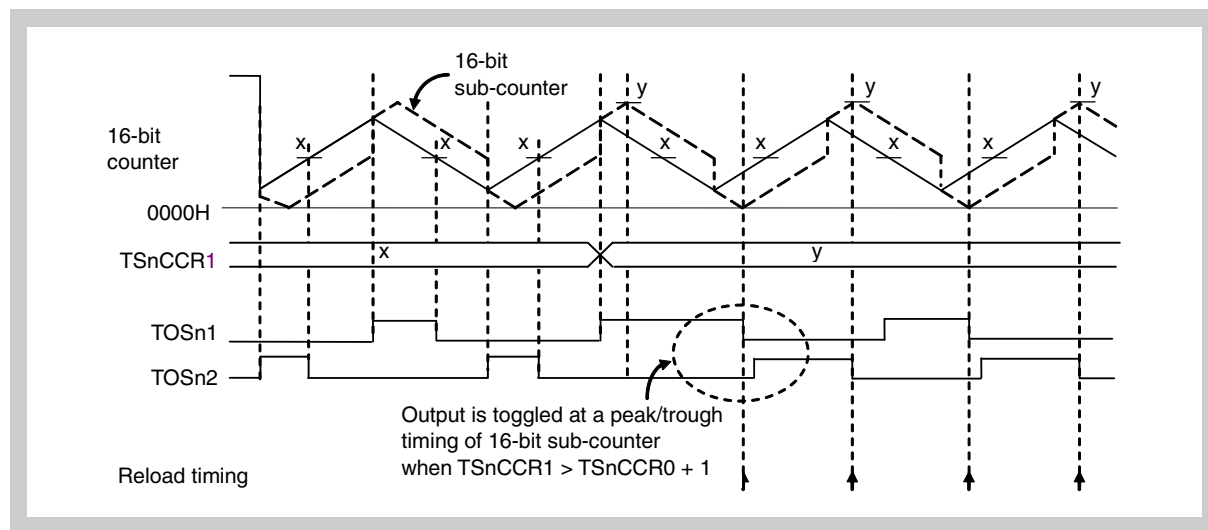
**Note** TSnCCR0 = 12, TSnDTC0 = 0, TSnDTC1 = 0

**(6) Caution on timer output in high-accuracy T-PWM mode**

There are cautions for TSnCCR1 to TSnCCR3 as follows when varying 6-phase PWM duty by using reload (batch rewrite).

**(a) In case of  $TSnCCR0 + 2 \leq TSnCCRM$  (Setting prohibited)**

Figure 17-82 on page 807 shows the case when the value of “TSnCCR0 + 2 or more” is set to the TSnCCR1 register. When the TSnCCR1 register setting is changed like this, a match between the 16-bit counter and TSnCCR1 register does not occur thereafter. Therefore, the TOSn1 pin output level is forcibly changed to inactive level at the following 16-bit sub-counter trough timing. Output will be switched at 16-bit sub-counter peak/trough timing after that.



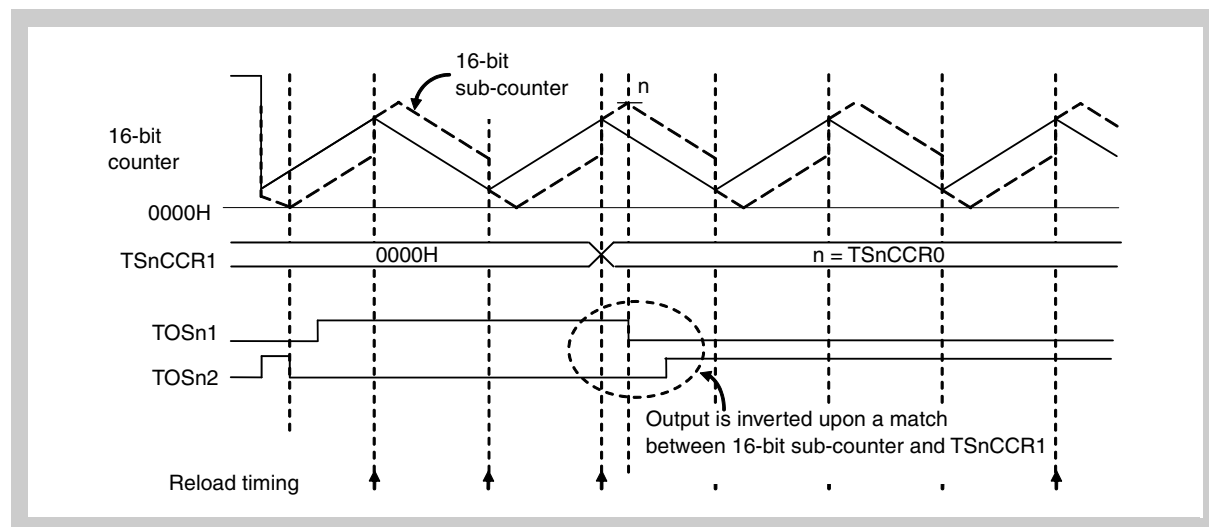
**Figure 17-82 Output when  $TSnCCR0 + 2 \leq TSnCCR1$**

**Note**  $m = 1$  to 3

**(b) In case of rewriting from  $\text{TSnCCRm} = 0000\text{H}$  to  $\text{TSnCCRm} = \text{TSnCCR0}$** 

Figure 17-83 shows the output waveform where the  $\text{TSnCCR1}$  register setting is changed from 100% output to 0% output.

The  $\text{TOSn1}$  pin output is inverted upon a match between the  $\text{TSnCCR1}$  register and 16-bit sub-counter, and the  $\text{TOSn2}$  pin output is inverted after the dead time count.

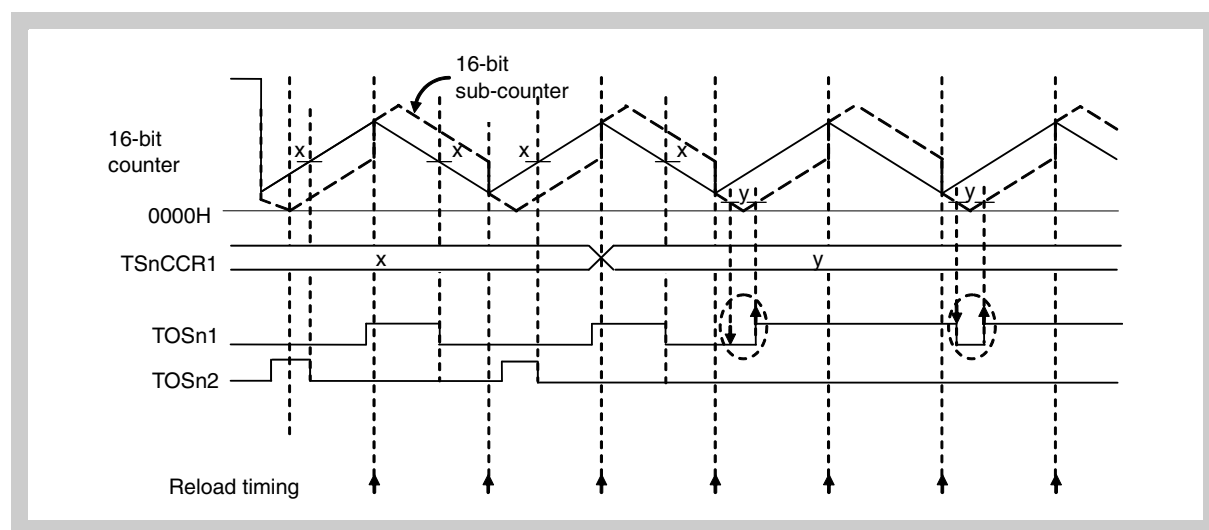


**Figure 17-83** Output when rewriting from  $\text{TSnCCR1} = 0000\text{H}$  to  $\text{TSnCCR1} = \text{TSnCCR0}$

**(c) In case of rewriting from “ $\text{TSnDTC0} + \text{TSnDTC1} < \text{TSnCCRm} < \text{TSnCCR0} - \text{TSnDTC0} - \text{TSnDTC1}$ ” to “ $\text{TSnCCRm} < \text{TSnDTC0} + \text{TSnDTC1}$ ”**

Figure 17-84 shows the output waveform when rewriting the  $\text{TSnCCR1}$  register from  $x$  ( $\text{TSnDTC0} + \text{TSnDTC1} < x < \text{TSnCCR0} - \text{TSnDTC0} - \text{TSnDTC1}$ ) to  $y$  ( $y < \text{TSnDTC0} + \text{TSnDTC1}$ ).

In this case, the  $\text{TOSn1}$  pin output becomes active when the  $\text{TOSn1}$  pin set condition occurs upon a match between the 16-bit counter (or 16-bit sub-counter) and the  $\text{TSnCCR1}$  register immediately after reload (batch rewrite).

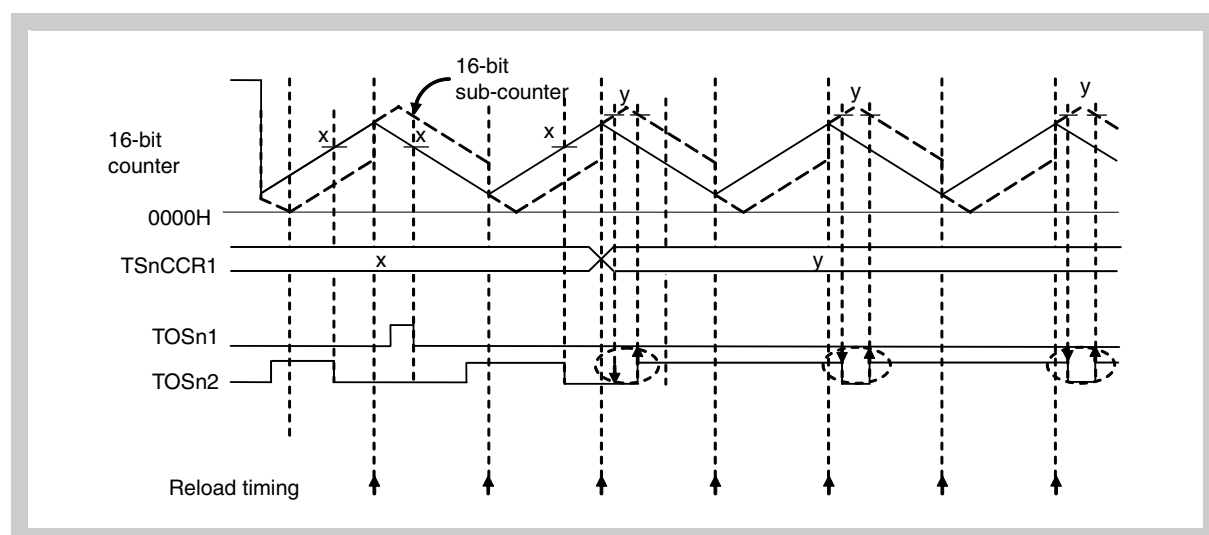


**Figure 17-84** Output when rewriting from  $\text{TSnDTC0} + \text{TSnDTC1} < \text{TSnCCR1} < \text{TSnCCR0} - \text{TSnDTC0} - \text{TSnDTC1}$  to  $\text{TSnCCR1} < \text{TSnDTC0} + \text{TSnDTC1}$

**Note**  $m = 1$  to 3

- (d) In case of rewriting from “ $\text{TSnDTC0} + \text{TSnDTC1} < \text{TSnCCRm} < \text{TSnCCR0} - \text{TSnDTC0} - \text{TSnDTC1}$ ” to “ $\text{TSnCCR0} - \text{TSnDTC1} + 1 < \text{TSnCCRm} < \text{TSnCCR0}$ ”

Figure 17-85 shows the output waveform when rewriting the TSnCCR1 register from  $x$  ( $\text{TSnDTC0} + \text{TSnDTC1} < x < \text{TSnCCR0} - \text{TSnDTC0} - \text{TSnDTC1}$ ) to  $y$  ( $\text{TSnCCR0} - \text{TSnDTC0} - \text{TSnDTC1} < \text{TSnDTC0} < \text{TSnCCR0}$ ). In this case, the TOSn2 pin output becomes inactive (high level) when the TOSn2 pin set condition occurs upon a match between the 16-bit counter (or 16-bit sub-counter) and TSnCCRm register immediately after batch rewrite.



**Figure 17-85** Output when rewriting from “ $\text{TSnDTC0} + \text{TSnDTC1} < \text{TSnCCR1} < \text{TSnCCR0} - \text{TSnDTC0} - \text{TSnDTC1}$ ” to “ $\text{TSnCCR0} - \text{TSnDTC1} + 1 < \text{TSnCCR1} < \text{TSnCCR0}$ ”

**Note**  $m = 1$  to 3

**(7) Timer output change after compare register updating**

Timer output is affected when the compare register value is updated during reload execution. The timer output level is changed at any timing listed in *Table 17-33 on page 782* and *Table 17-34 on page 810*.

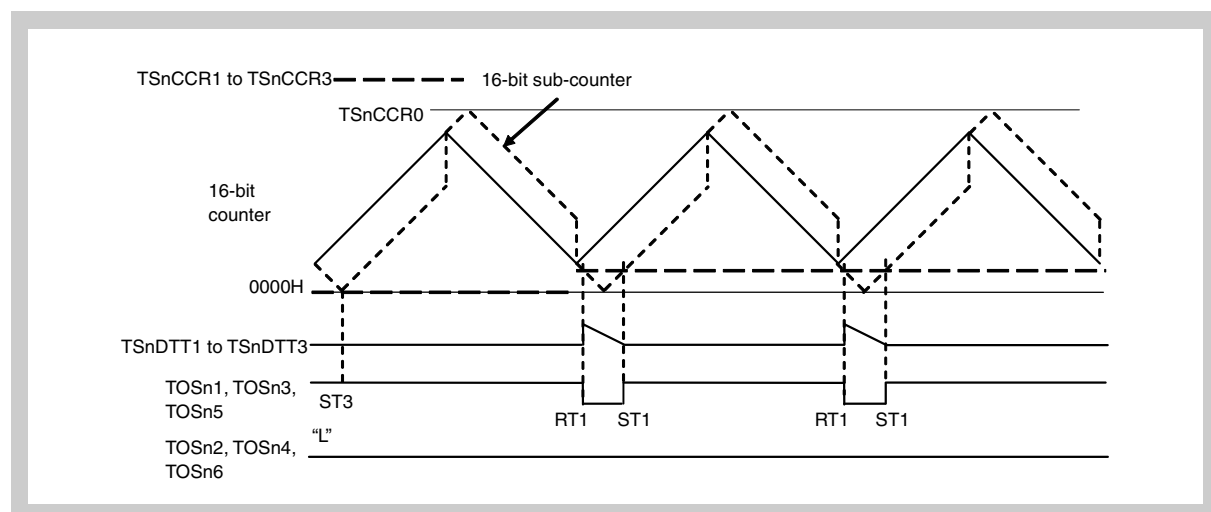
**Table 17-34 Positive phase operation condition list**

Operation	Symbol	Condition
Set	ST1	Match between counting up near the 16-bit sub-counter trough and compare register values
Clear	RT1	Match between counting down near the 16-bit sub-counter trough and compare register values
Set	ST2	At completion of dead time counter (TSnDTC0) operation
Clear	RT2	When 16-bit counter value matches with compare register value during count-down operation
Set	ST3	100% output for PWM duty
Clear	RT3	When no match occurs until 16-bit sub-counter counts down to 0000H
Clear	RT4	TSnCCR0 and TSnDTC0 settings are changed at a reload timing. Though neither a match (nor a match interrupt) occurs between TSnCCR0 and TSnDTC0, the operation is cleared by special processing.
Clear	RT5	The operation is cleared upon a match between peripheral 16-bit sub-counter peak and compare register values in positive phase active level.

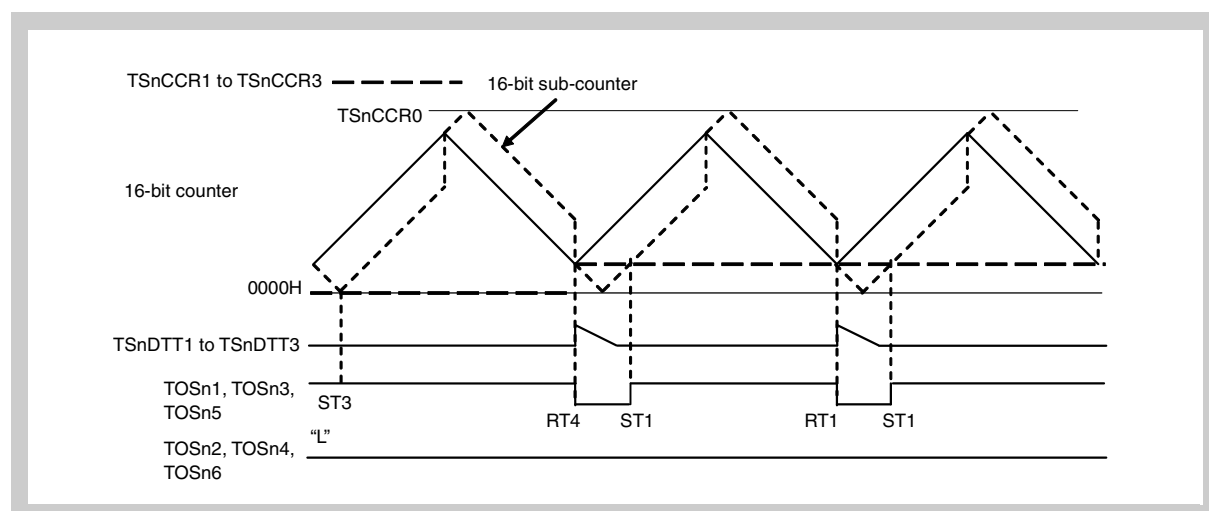
**Table 17-35 Negative phase operation condition list**

Operation	Symbol	Condition
Set	SB1	Match between counting down near the 16-bit sub-counter peak and compare register values
Clear	RB1	Match between counting up near the 16-bit sub-counter peak and compare register values
Set	SB2	At completion of dead time counter (TSnDTC1) operation
Clear	RB2	When 16-bit counter value matches with compare register value during count-up operation
Set	SB3	100% output for PWM duty
Clear	RB3	When no match occurs until 16-bit sub-counter counts up to TSnCCR0
Clear	RB4	TSnCCR0 and TSnDTC0 settings are changed at a reload timing. Though neither a match (nor a match interrupt) occurs between TSnCCR0 and TSnDTC1, the operation is cleared by special processing.
Clear	RB5	The operation is cleared upon a match between peripheral 16-bit sub-counter trough and compare register values in negative phase active level.

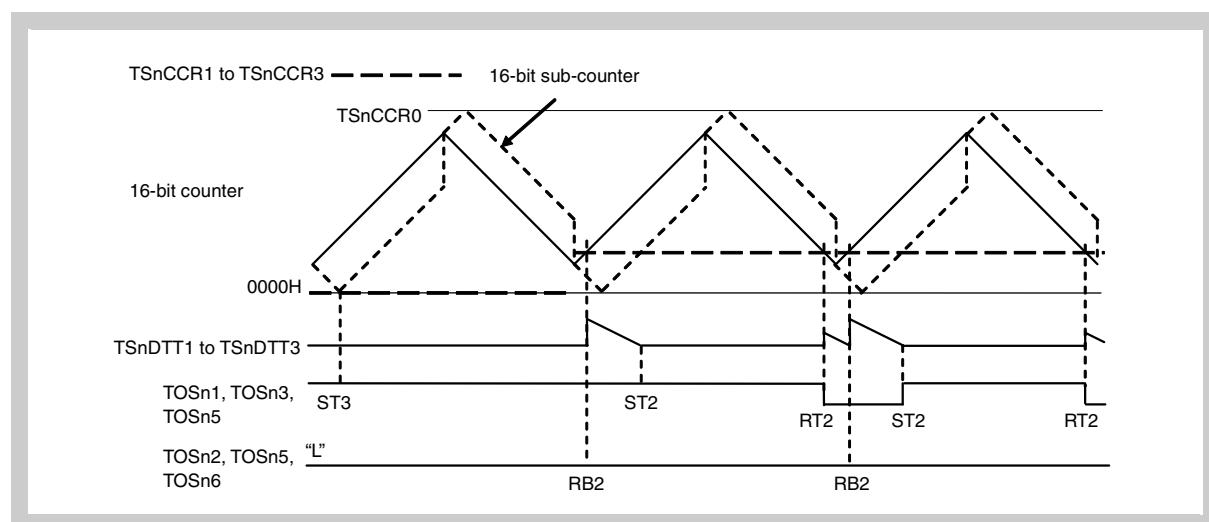
Compare register value immediately before trough reload	Compare register value after trough reload (TSnDTC0 < TSnDTC1)	Figure No.
0000H	0000H < TSnCCR1 to TSnCCR3 < TSnDTC0	Figure 17-86 on page 811
	TSnCCR1 to TSnCCR3 = 0000H, TSnDTC0 + 1	Figure 17-87 on page 812
	TSnDTC0 + 1 < TSnCCR1 to TSnCCR3 ≤ TSnDTC0 × 2	Figure 17-88 on page 812
	TSnDTC0 × 2 < TSnCCR1 to TSnCCR3 < TSnCCR0 – TSnDTC0 – TSnDTC1	Figure 17-89 on page 813
	TSnCCR0 – TSnDTC0 – TSnDTC1 ≤ TSnCCR1 to TSnCCR3 < TSnCCR0 – TSnDTC1	Figure 17-90 on page 813
	TSnCCR0 – TSnDTC1 ≤ TSnCCR1 to TSnCCR3 < TSnCCR0	Figure 17-91 on page 813
	TSnCCR1 to TSnCCR3 = TSnCCR0	Figure 17-92 on page 814



**Figure 17-86**    **TSnCCR1 to TSnCCR3 = 0000H → 0000H < TSnCCR1 to TSnCCR3 < TSnDTC0**



**Figure 17-87**  $\text{TSnCCR1 to TSnCCR3} = 0000\text{H} \rightarrow \text{TSnCCR1 to TSnCCR3} = \text{TSnDTC0}, \text{TSnDTC0} + 1$

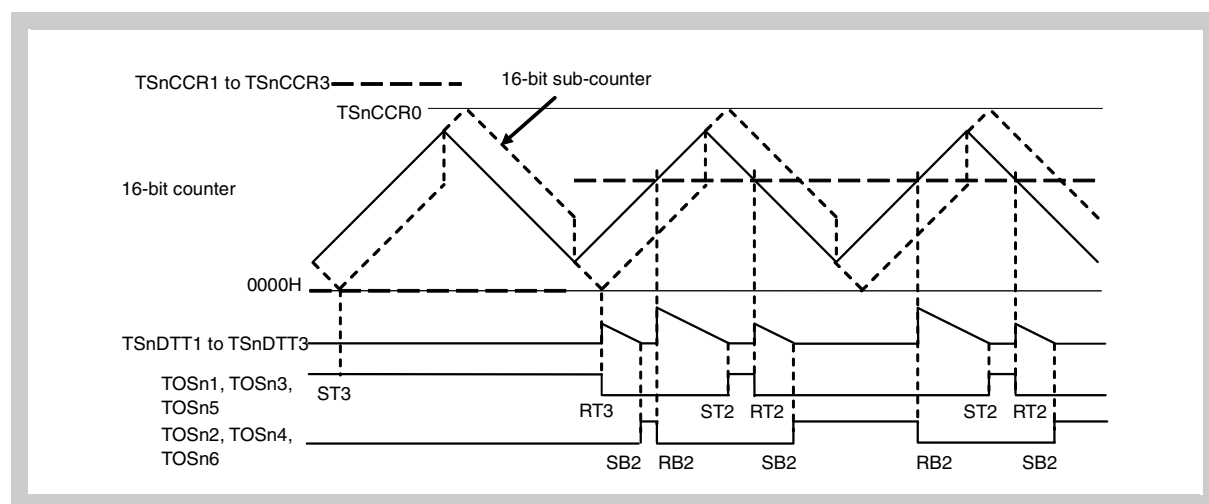


**Figure 17-88**  $\text{TSnCCR1 to TSnCCR3} = 0000\text{H} \rightarrow \text{TSnDTC0} < \text{TSnCCR1 to TSnCCR3} < \text{TSnDTC0} \times 2$

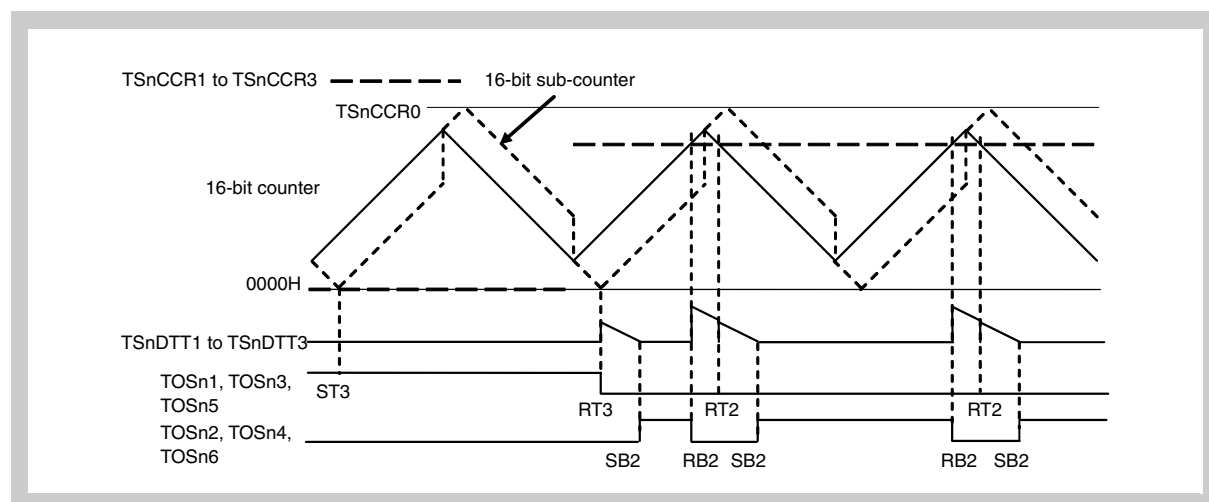
When the values of TSnCCR1 to TSnCCR3 are changed from " $0000\text{H} \leq \text{TSnCCR1 to TSnCCR3} < \text{TSnDTC0}$ " to " $\text{TSnDTC0} < \text{TSnCCR1 to TSnCCR3} < \text{TSnDTC0} \times 2$ ", the positive phase will be 100% output for one cycle, as shown in Figure 17-88.

To prevent this phenomenon, change " $0000\text{H} \leq \text{TSnCCR1 to TSnCCR3} < \text{TSnDTC0}$ " to " $\text{TSnDTC0} < \text{TSnCCR1 to TSnCCR3} < \text{TSnDTC0} \times 2$ " through TSnDTC0, or directly change " $0000\text{H} \leq \text{TSnCCR1 to TSnCCR3} < \text{TSnDTC0}$ " to " $\text{TSnDTC0} \times 2 \leq \text{TSnCCR1 to TSnCCR3}$ ".

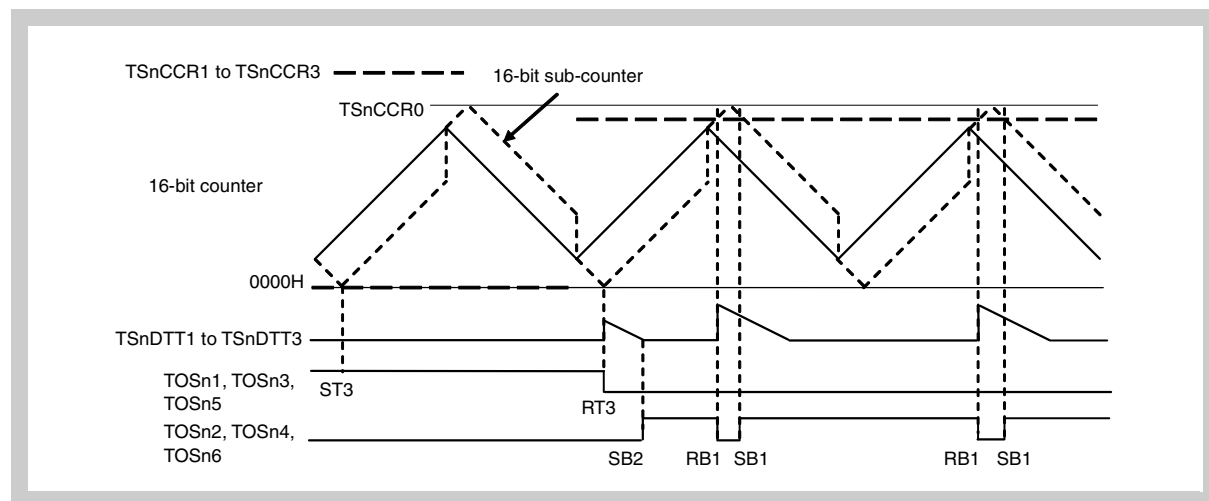




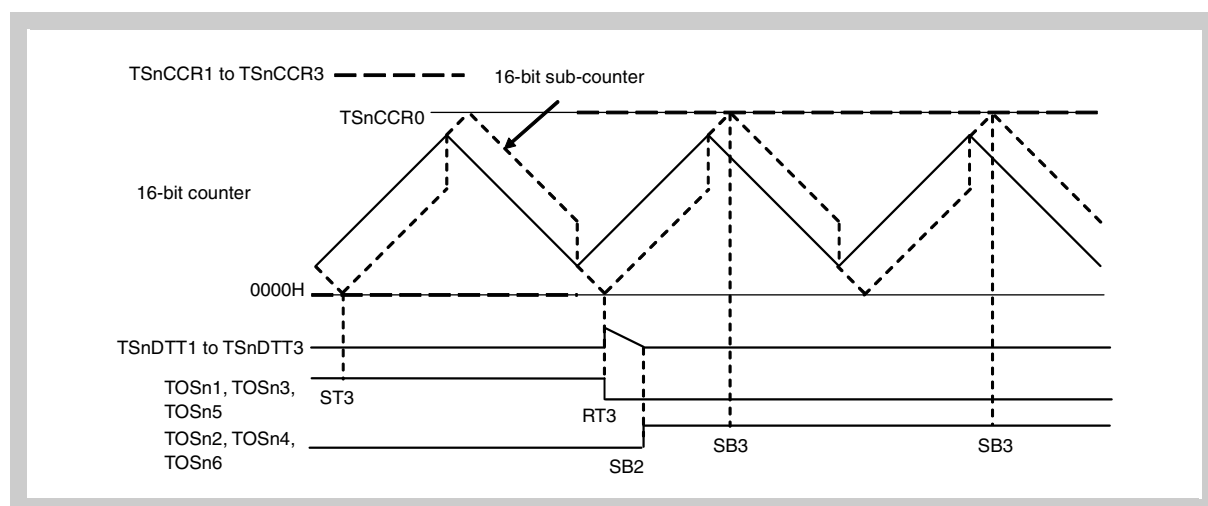
**Figure 17-89**  $TSnCCR1$  to  $TSnCCR3 = 0000H \rightarrow TSnDTC0 \times 2 < TSnCCR1$  to  $TSnCCR3 < TSnCCR0 - TSnDTC1 - TSnDTC0$



**Figure 17-90**  $TSnCCR1$  to  $TSnCCR3 = 0000H \rightarrow TSnCCR0 - TSnDTC1 - TSnDTC0 < TSnCCR1$  to  $TSnCCR3 < TSnCCR0 - TSnDTC1$

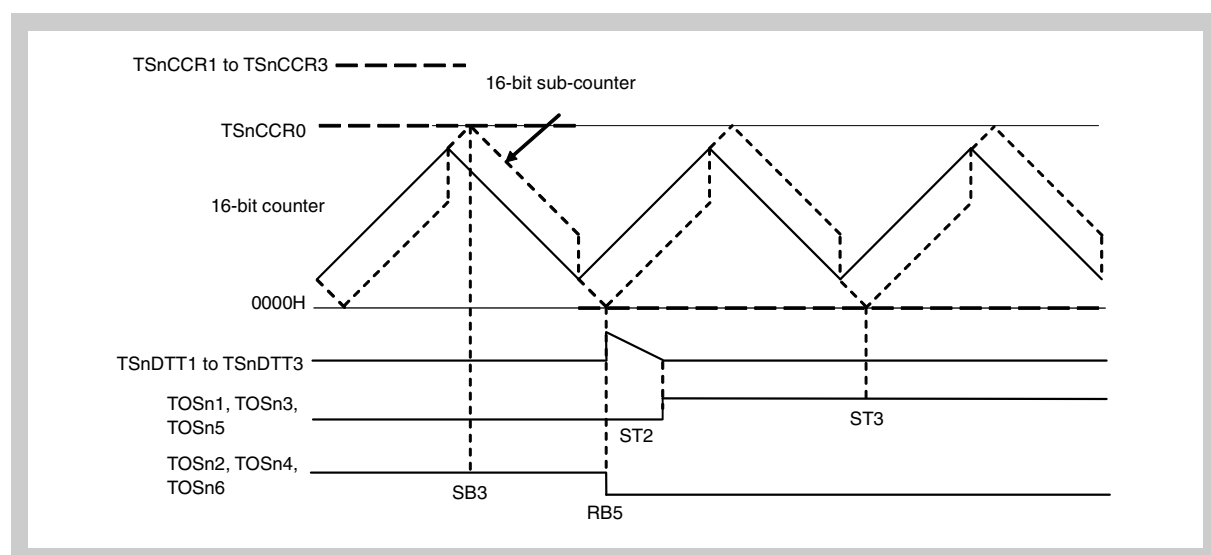


**Figure 17-91**  $TSnCCR1$  to  $TSnCCR3 = 0000H \rightarrow TSnCCR0 - TSnDTC1 < TSnCCR1$  to  $TSnCCR3 < TSnCCR0$

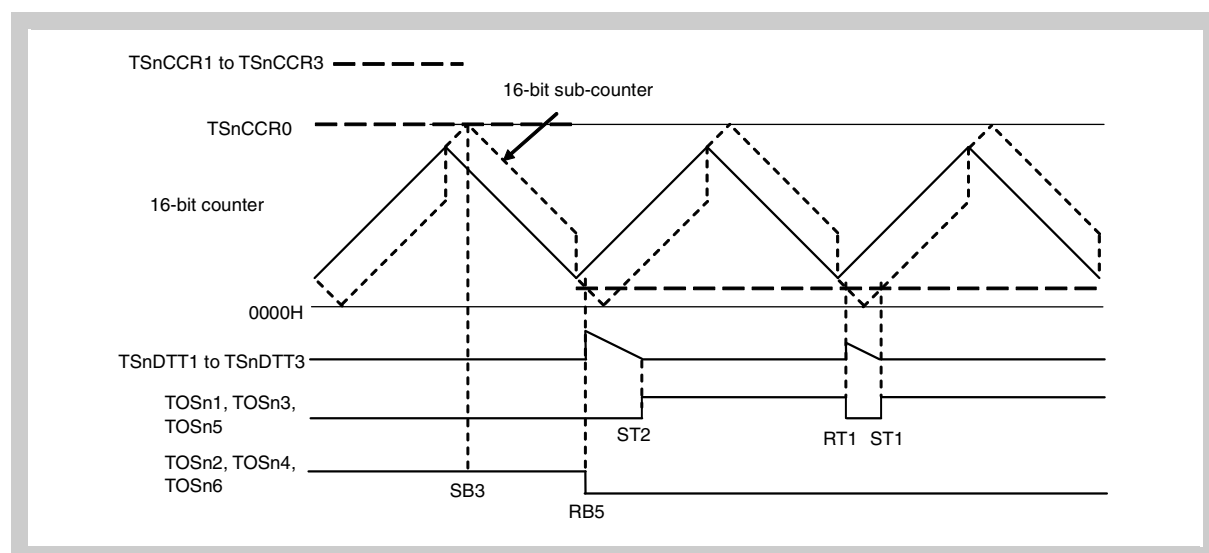


**Figure 17-92**  $\text{TSnCCR1 to TSnCCR3} = 0000\text{H} \rightarrow \text{TSnCCR1 to TSnCCR3} < \text{TSnCCR0}$

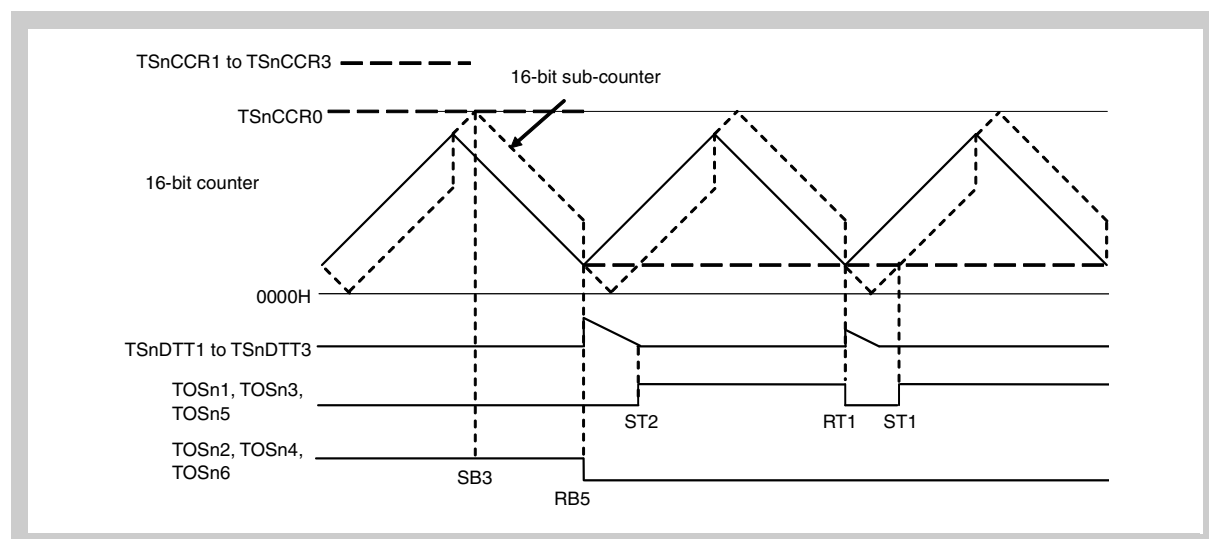
Compare register value immediately before trough reload	Compare register value after trough reload	Figure No.
TSnCCR0	TSnCCR1 to TSnCCR3 = 0000H	Figure 17-93 on page 815
	0000H < TSnCCR1 to TSnCCR3 < TSnDTC0	Figure 17-94 on page 816
	TSnCCR1 to TSnCCR3 = TSnDTC0, TSnDTC0 + 1	Figure 17-95 on page 816
	TSnDTC0 + 1 < TSnCCR1 to TSnCCR3 < TSnDTC0 + TSnDTC1	Figure 17-96 on page 817
	TSnDTC0 + TSnDTC1 < TSnCCR1 to TSnCCR3 < TSnCCR0 – TSnDTC0 – TSnDTC1	Figure 17-97 on page 817
	TSnCCR0 – TSnDTC0 – TSnDTC1 ≤ TSnCCR1 to TSnCCR3 < TSnCCR0 – TSnDTC1	Figure 17-98 on page 818
	TSnCCR0 – TSnDTC1 ≤ TSnCCR1 to TSnCCR3 < TSnCCR0	Figure 17-99 on page 818



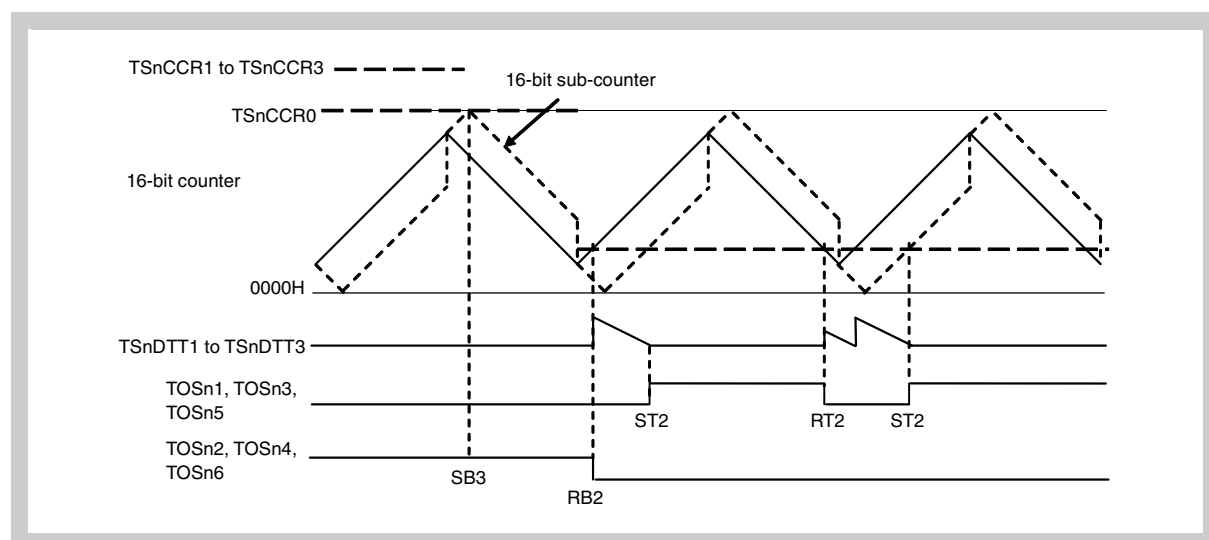
**Figure 17-93** TSnCCR1 to TSnCCR3 = TSnCCR0 → TSnCCR1 to TSnCCR3 = 0000H



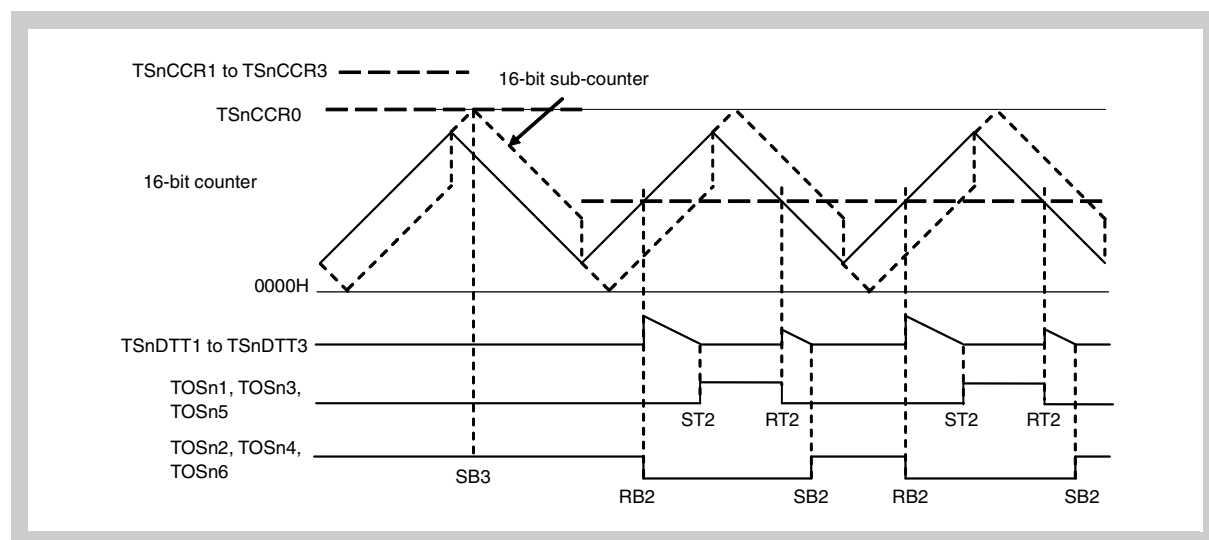
**Figure 17-94**  $\text{TSnCCR1 to TSnCCR3} = \text{TSnCCR0} \rightarrow 0000\text{H} < \text{TSnCCR1 to TSnCCR3} < \text{TSnDTC0}$



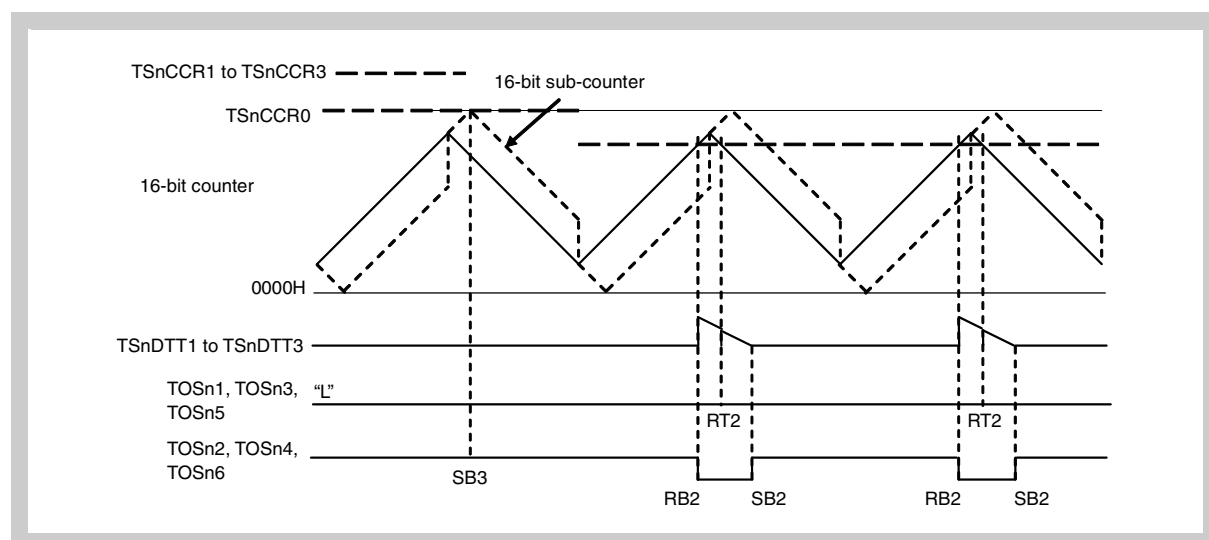
**Figure 17-95**  $\text{TSnCCR1 to TSnCCR3} = \text{TSnCCR0} \rightarrow \text{TSnCCR1 to TSnCCR3} = \text{TSnDTC0, TSnDTC0} + 1$



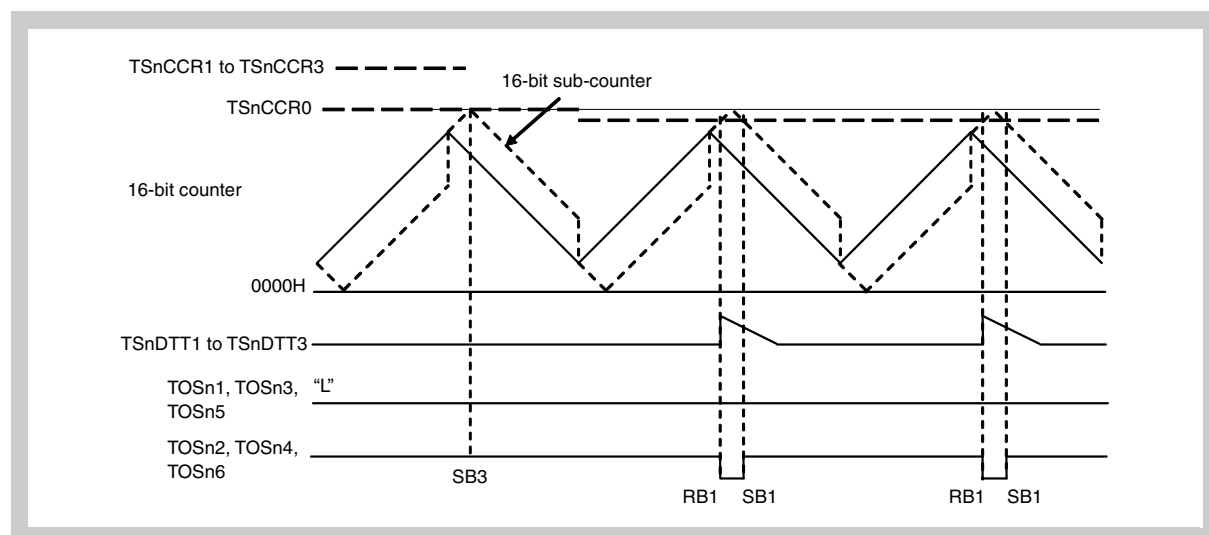
**Figure 17-96**  $TSnCCR1 \text{ to } TSnCCR3 = TSnCCR0 \rightarrow TSnDTC0 + 1 < TSnCCR1 \text{ to } TSnCCR3 \leq TSnDTC0 + TSnDTC1$



**Figure 17-97**  $TSnCCR1 \text{ to } TSnCCR3 = TSnCCR0 \rightarrow TSnDTC0 + TSnDTC1 < TSnCCR1 \text{ to } TSnCCR3 < TSnCCR0 - TSnDTC1 - TSnDTC0$

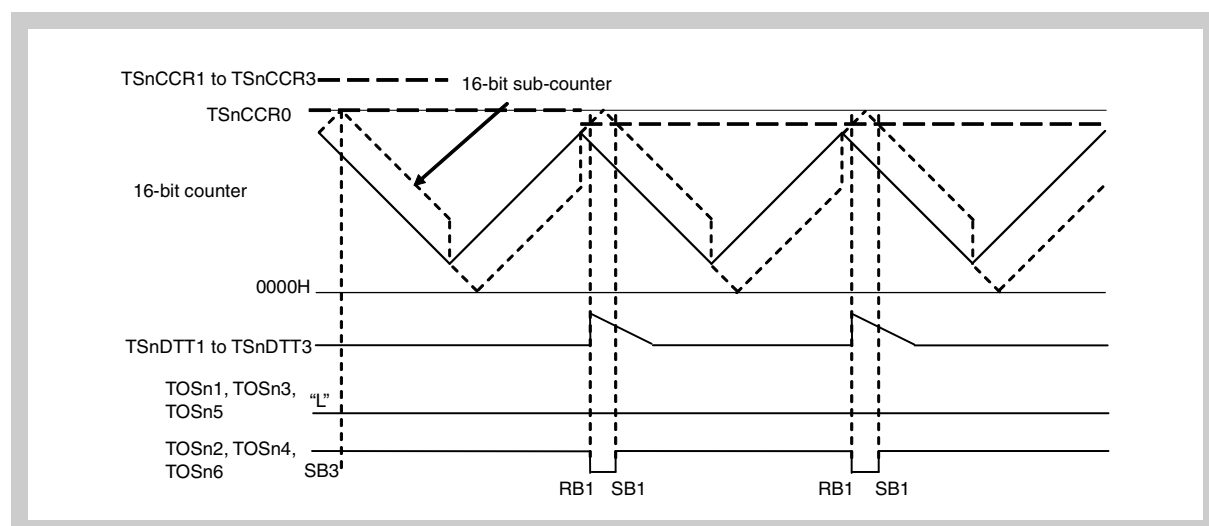


**Figure 17-98**  $TSnCCR1 \text{ to } TSnCCR3 = TSnCCR0 \rightarrow TSnCCR0 - TSnDTC1 - TSnDTC0 < TSnCCR1 \text{ to } TSnCCR3 < TSnCCR0 - TSnDTC1$

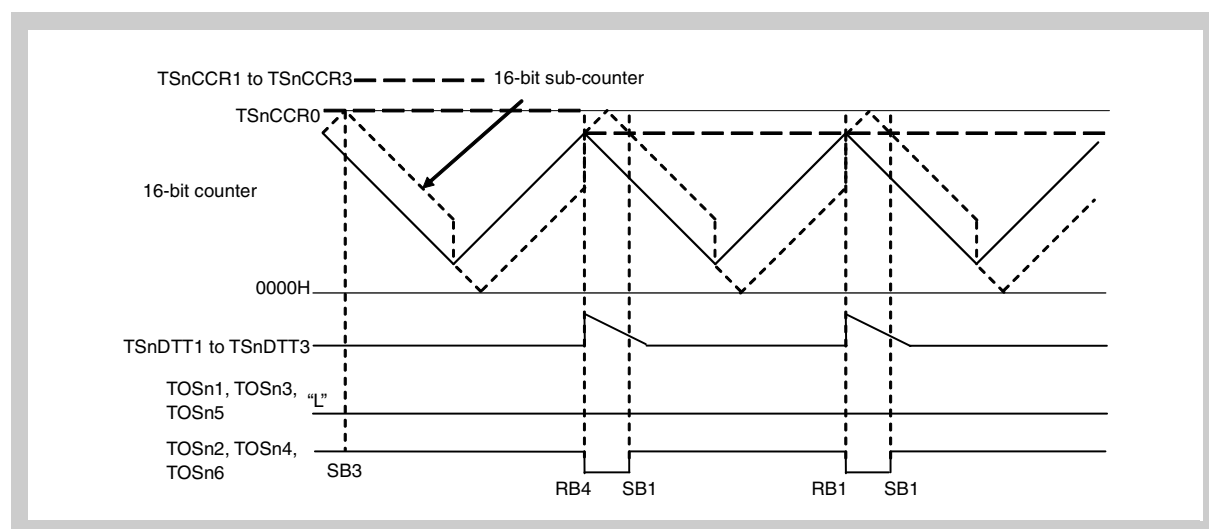


**Figure 17-99**  $TSnCCR1 \text{ to } TSnCCR3 = TSnCCR0 \rightarrow TSnCCR0 - TSnDTC1 \leq TSnCCR1 \text{ to } TSnCCR3 < TSnCCR0$

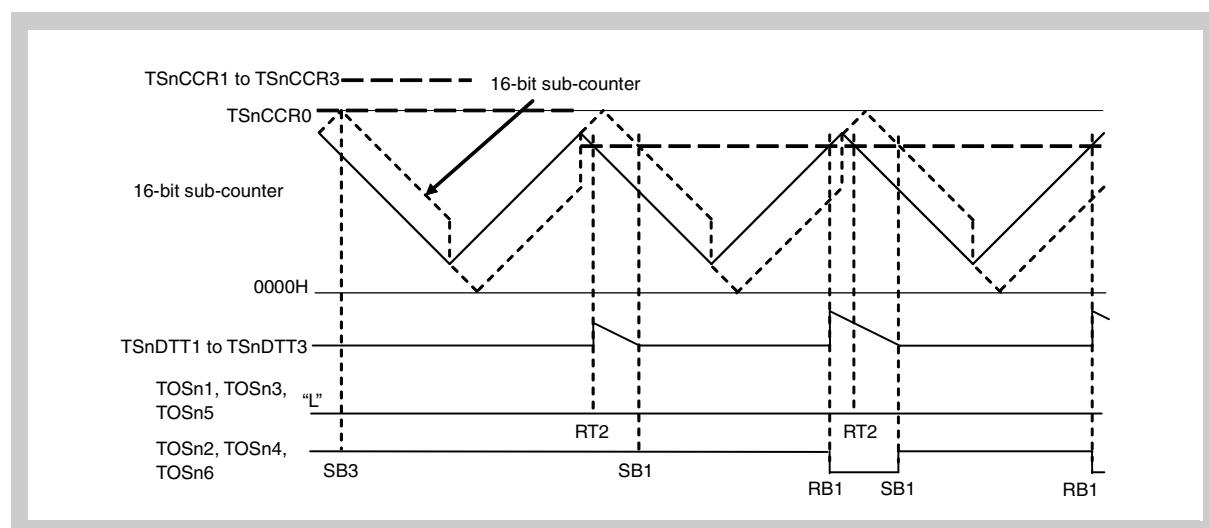
Compare register value immediately before peak reload	Compare register value after peak reload (TSnDTC1 < TSnDTC0)	Figure No.
TSnCCR0	$TSnCCR0 - TSnDTC1 \leq TSnCCR1$ to $TSnCCR3 < TSnCCR0$	Figure 17-100 on page 819
	$TSnCCR1$ to $TSnCCR3 = TSnCCR0 - TSnDTC1$	Figure 17-101 on page 820
	$TSnCCR0 - TSnDTC1 \times 2 \leq TSnCCR1$ to $TSnCCR3 < TSnCCR0 - TSnDTC1$	Figure 17-102 on page 820
	$TSnDTC0 + TSnDTC1 < TSnCCR1$ to $TSnCCR3 < TSnCCR0 - TSnDTC1 \times 2$	Figure 17-103 on page 821
	$TSnDTC0 + 1 < TSnCCR1$ to $TSnCCR3 < TSnDTC0 + TSnDTC1$	Figure 17-104 on page 821
	$0000H < TSnCCR1$ to $TSnCCR3 \leq TSnDTC0 + 1$	Figure 17-105 on page 822
	$TSnCCR1$ to $TSnCCR3 = 0000H$	Figure 17-106 on page 822



**Figure 17-100**  $TSnCCR1$  to  $TSnCCR3 = TSnCCR0 \rightarrow TSnCCR0 - TSnDTC1 < TSnCCR1$  to  $TSnCCR3 < TSnCCR0$



**Figure 17-101**  $TSnCCR1$  to  $TSnCCR3 = TSnCCR0 \rightarrow TSnCCR1$  to  $TSnCCR3 = TSnDTC0 - TSnDTC1$

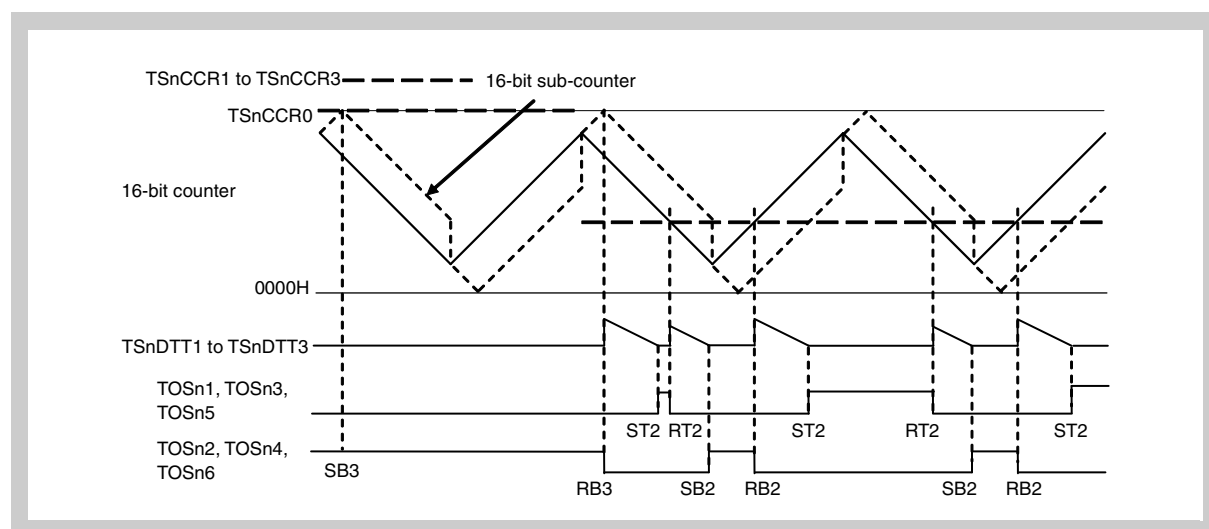


**Figure 17-102**  $TSnCCR1$  to  $TSnCCR3 = TSnCCR0 \rightarrow TSnCCR0 - TSnDTC1 \times 2 < TSnCCR1$  to  $TSnCCR3 < TSnCCR0 - TSnDTC1$

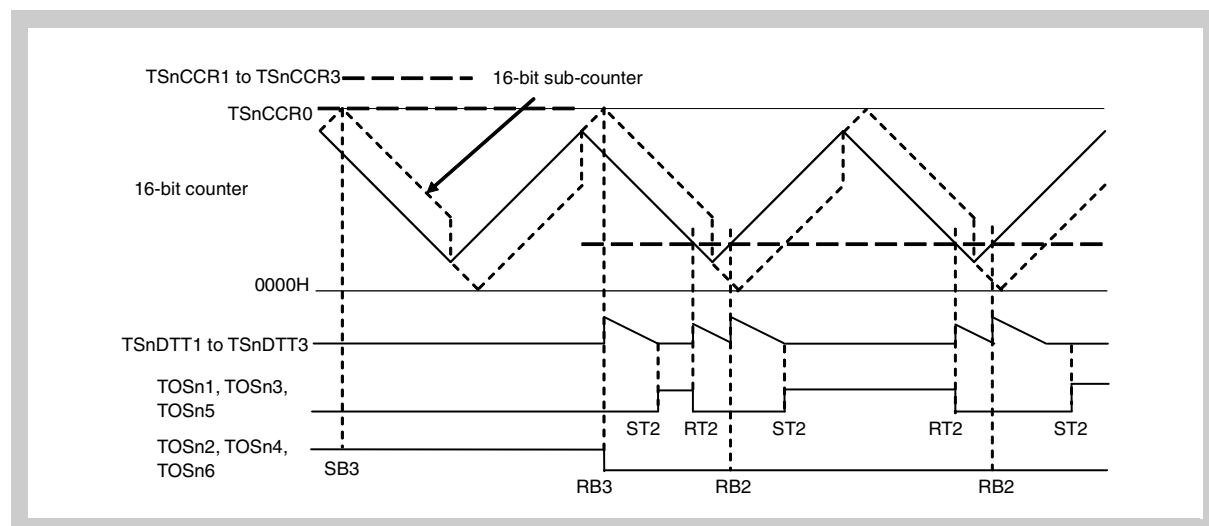
When the values of  $TSnCCR1$  to  $TSnCCR3$  are changed from " $TSnCCR0 - TSnDTC1 < TSnCCR1$  to  $TSnCCR3 \leq TSnCCR0$ " to " $TSnCCR0 - TSnDTC1 \times 2 < TSnCCR1$  to  $TSnCCR3 < TSnCCR0 - TSnDTC1$ ", the negative phase will be 100% output for one cycle, as shown in *Figure 17-102*.

To prevent this phenomenon, change " $TSnCCR0 - TSnDTC1 < TSnCCR1$  to  $TSnCCR3 \leq TSnCCR0$ " to " $TSnDTC0 < TSnCCR1$  to  $TSnCCR3 < TSnDTC1 \times 2$ " through " $TSnCCR0 - TSnDTC1$ ", or directly change " $TSnCCR0 - TSnDTC1 < TSnCCR1$  to  $TSnCCR3 < TSnCCR0$ " to " $TSnCCR1$  to  $TSnCCR3 \leq TSnCCR0 - TSnDTC1 \times 2$ ".

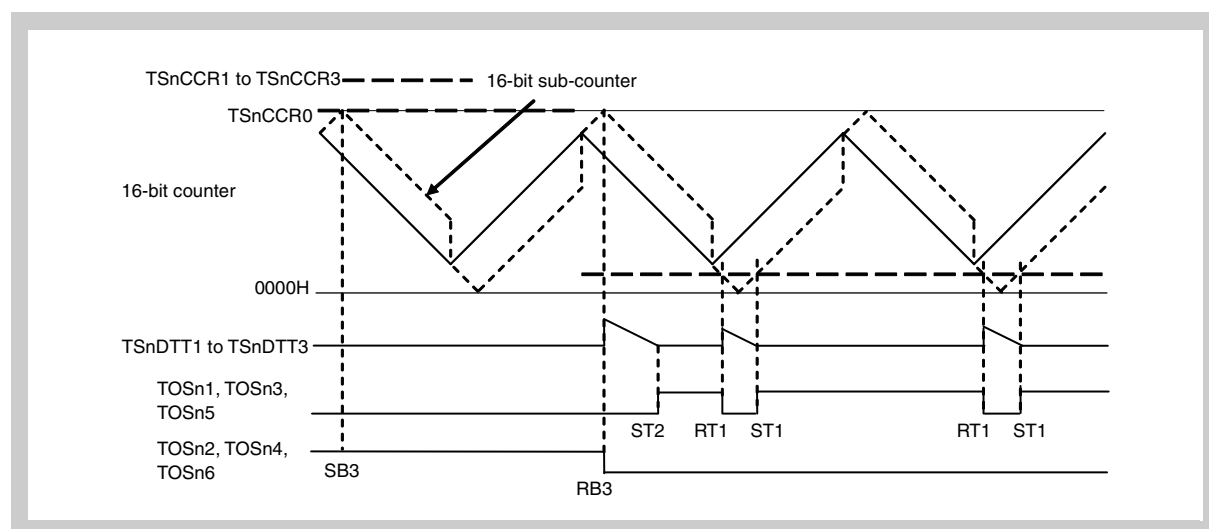




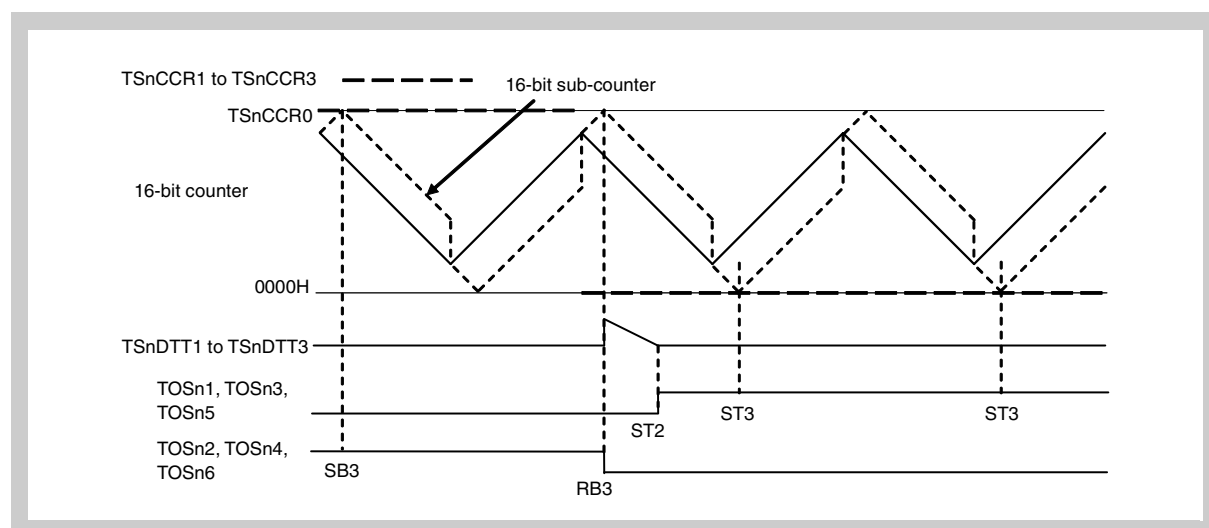
**Figure 17-103**  $TSnCCR1 \text{ to } TSnCCR3 = TSnCCR0 \rightarrow TSnDTC0 + TSnDTC1 < TSnCCR1 \text{ to } TSnCCR3 < TSnCCR0 - TSnDTC1 \times 2$



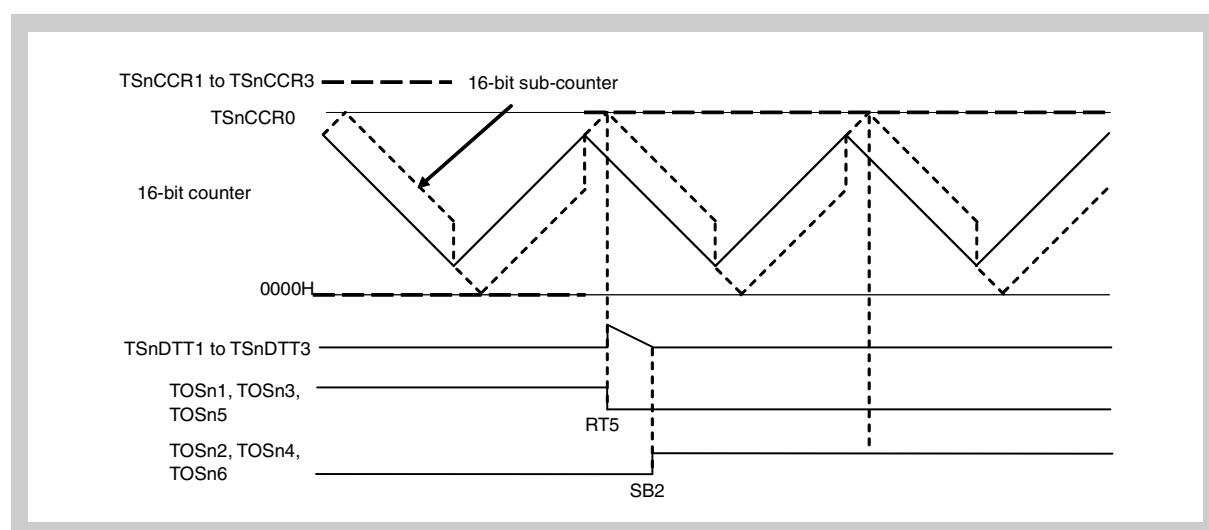
**Figure 17-104**  $TSnCCR1 \text{ to } TSnCCR3 = TSnCCR0 \rightarrow TSnDTC0 + 1 < TSnCCR1 \text{ to } TSnCCR3 \leq TSnDTC0 + TSnDTC1$



**Figure 17-105**  $\text{TSnCCR1 to TSnCCR3} = \text{TSnCCR0} \rightarrow 0000\text{H} < \text{TSnCCR1 to TSnCCR3} \leq \text{TSnDTC0} + 1$

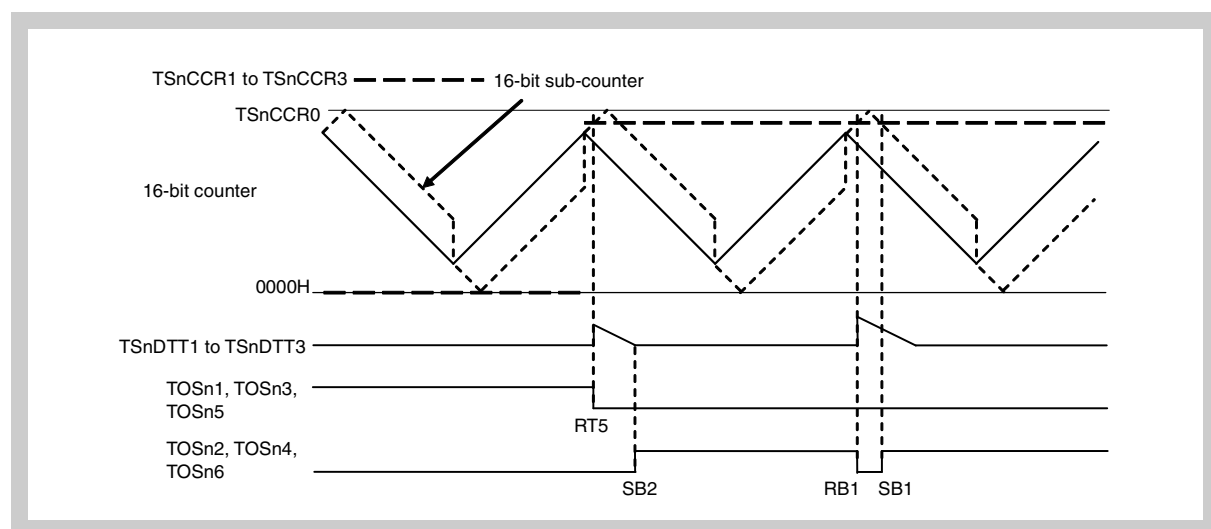


**Figure 17-106**  $\text{TSnCCR1 to TSnCCR3} = \text{TSnCCR0} \rightarrow \text{TSnCCR1 to TSnCCR3} = 0000\text{H}$

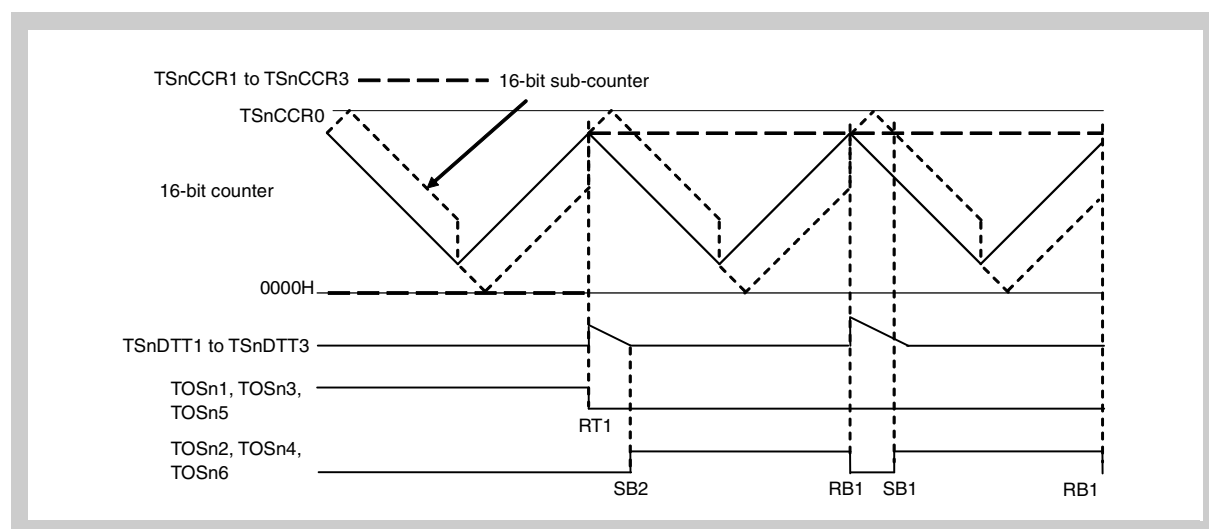


**Figure 17-107**  $\text{TSnCCR1 to TSnCCR3} = 0000\text{H} \rightarrow \text{TSnCCR1 to TSnCCR3} = \text{TSnCCR0}$

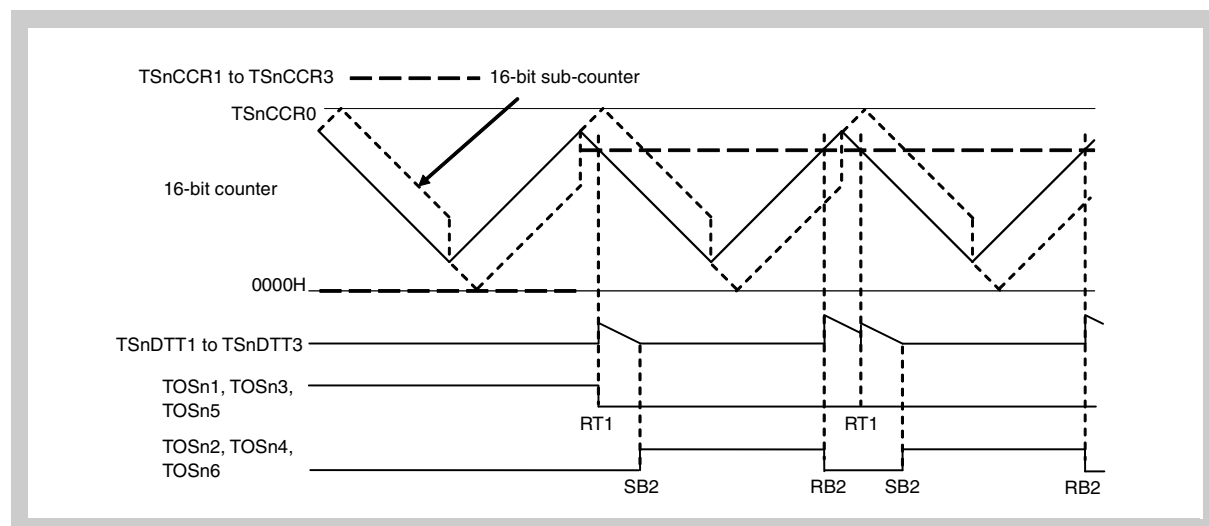
Compare register value immediately before peak reload	Compare register value after trough reload	Figure No.
0000H	$\text{TSnCCR1 to TSnCCR3} = \text{TSnCCR0}$	Figure 17-107 on page 822
	$\text{TSnCCR0} - \text{TSnDTC1} < \text{TSnCCR1 to TSnCCR3} < \text{TSnCCR0}$	Figure 17-108 on page 823
	$\text{TSnCCR1 to TSnCCR3} = \text{TSnCCR0} - \text{TSnDTC1}$	Figure 17-109 on page 824
	$\text{TSnCCR0} - \text{TSnDTC0} - \text{TSnDTC1} \leq \text{TSnCCR1 to TSnCCR3} < \text{TSnCCR0} - \text{TSnDTC1}$	Figure 17-110 on page 824
	$\text{TSnDTC0} + \text{TSnDTC1} < \text{TSnCCR1 to TSnCCR3} < \text{TSnCCR0} - \text{TSnDTC0} - \text{TSnDTC1}$	Figure 17-111 on page 825
	$\text{TSnDTC0} + 1 < \text{TSnCCR1 to TSnCCR3} \leq \text{TSnDTC0} + \text{TSnDTC1}$	Figure 17-112 on page 825
	$0000\text{H} < \text{TSnCCR1 to TSnCCR3} \leq \text{TSnDTC0} + 1$	Figure 17-113 on page 825



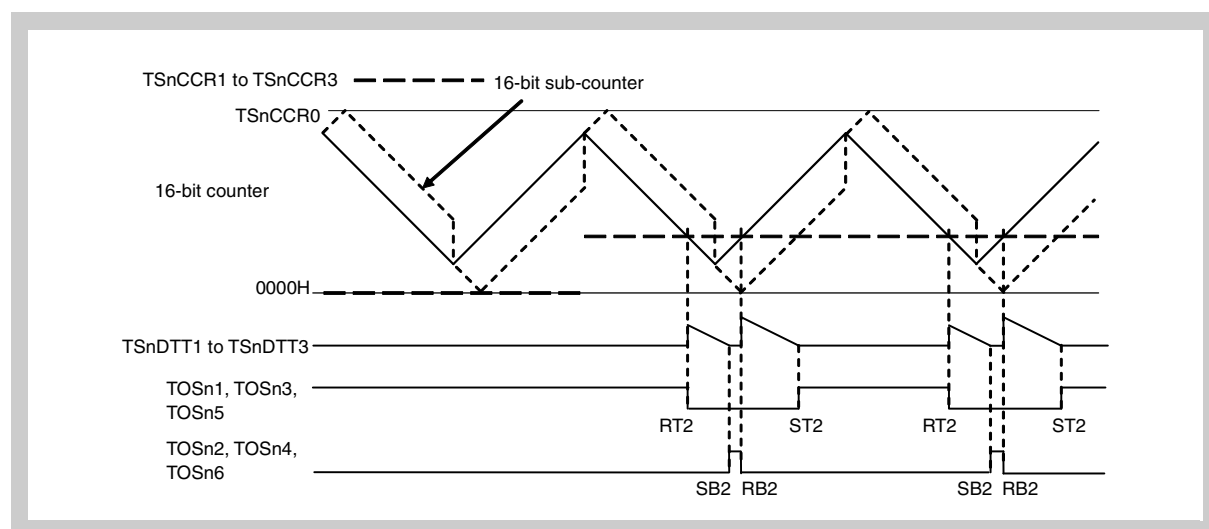
**Figure 17-108**  $\text{TSnCCR1 to TSnCCR3} = 0000\text{H} \rightarrow \text{TSnCCR0} - \text{TSnDTC1} < \text{TSnCCR1 to TSnCCR3} < \text{TSnCCR0}$



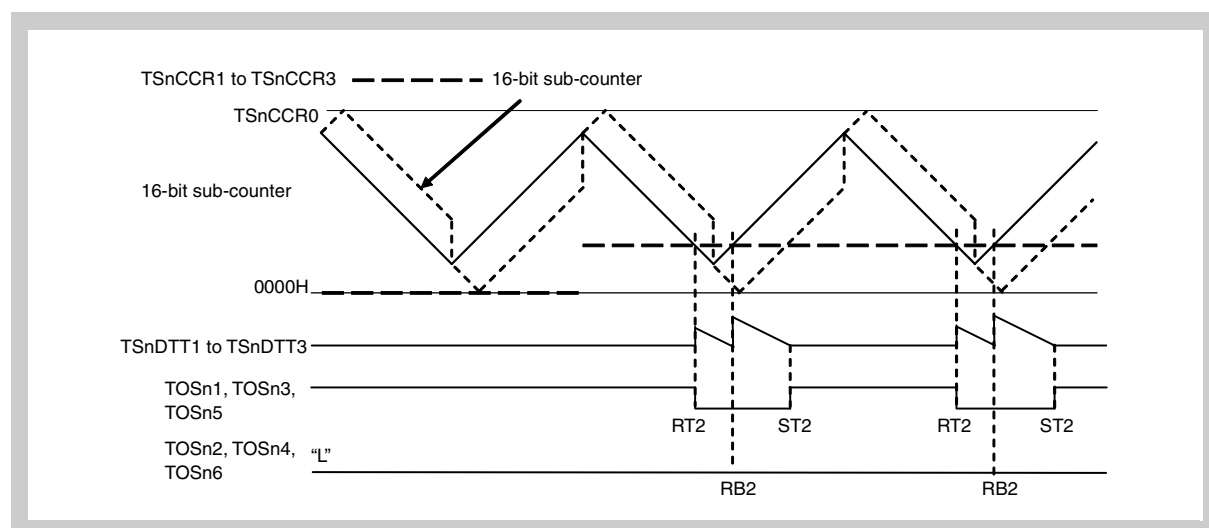
**Figure 17-109**  $\text{TSnCCR1 to TSnCCR3} = 0000\text{H} \rightarrow \text{TSnCCR1 to TSnCCR3} = \text{TSnCCR0} - \text{TSnDTC1}$



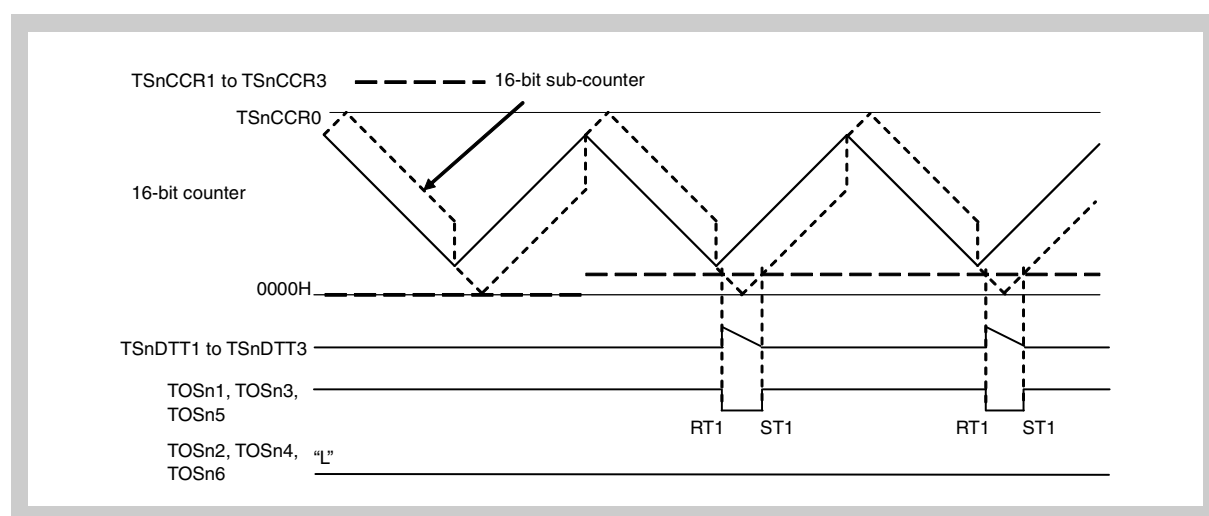
**Figure 17-110**  $\text{TSnCCR1 to TSnCCR3} = 0000\text{H} \rightarrow \text{TSnCCR0} - \text{TSnDTC0} - \text{TSnDTC1} < \text{TSnCCR1 to TSnCCR3} < \text{TSnCCR0} - \text{TSnDTC1}$



**Figure 17-111**  $\text{TSnCCR1 to TSnCCR3} = 0000\text{H} \rightarrow \text{TSnDTC0} + \text{TSnDTC1} < \text{TSnCCR1 to TSnCCR3} \leq \text{TSnCCR0} - \text{TSnDTC0} - \text{TSnDTC1}$



**Figure 17-112**  $\text{TSnCCR1 to TSnCCR3} = 0000\text{H} \rightarrow \text{TSnDTC0} + 1 < \text{TSnCCR1 to TSnCCR3} < \text{TSnDTC0} + \text{TSnDTC1}$



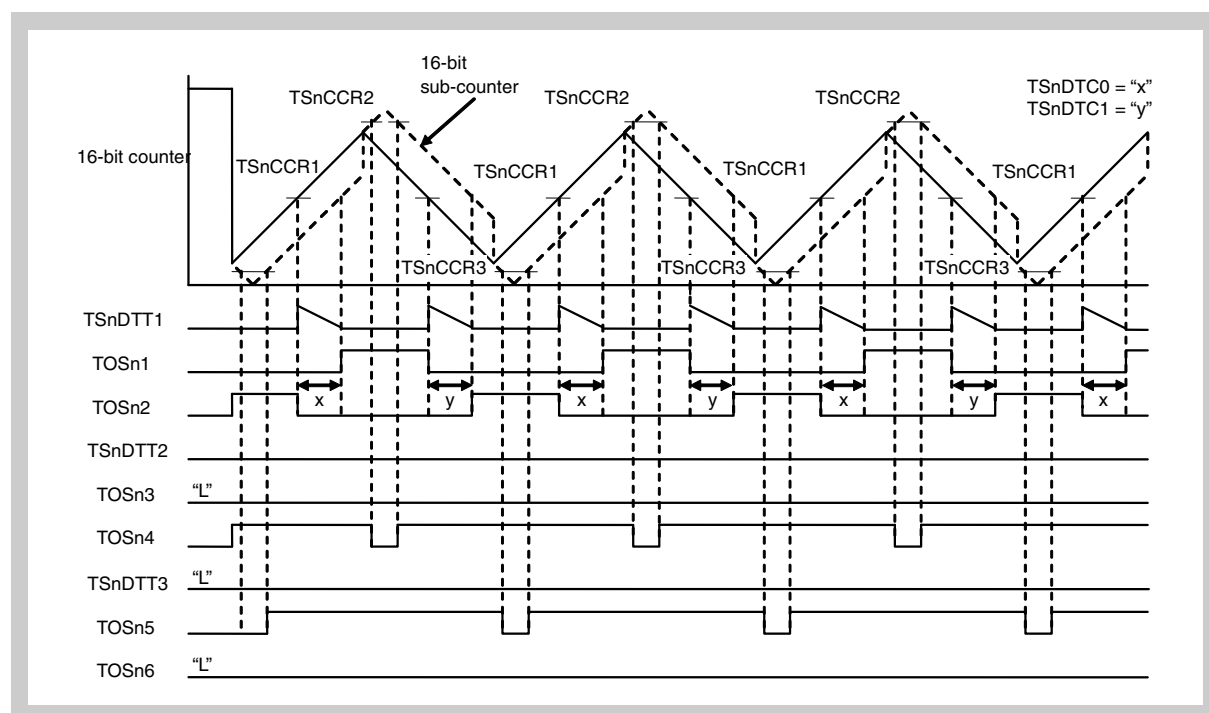
**Figure 17-113**  $\text{TSnCCR1 to TSnCCR3} = 0000\text{H} \rightarrow 0000\text{H} < \text{TSnCCR1 to TSnCCR3} \leq \text{TSnDTC0} + 1$

**(8) Dead time control in high-accuracy T-PWM mode**

In the high-accuracy T-PWM mode, the TSnCCR1 to TSnCCR3 registers are used for duty setting and the TSnCCR0 register is used for cycle setting. By using these four registers, duty variable type 6-phase PWM waveform can be output. To implement dead time control, there are three 10-bit down-counters that synchronously operate with the count clock of the 16-bit counter, and two dead time setting registers (TSnDTC0, TSnDTC1).

The TSnDTC0 register is used to set the dead time from when a negative phase changes to inactive until a positive phase changes to active. The TSnDTC1 register is used to set the dead time from when a positive phase changes to inactive until a negative phase changes to active.

The output waveform in case of TSnDTC0 = x, TSnDTC1 = y is shown below.

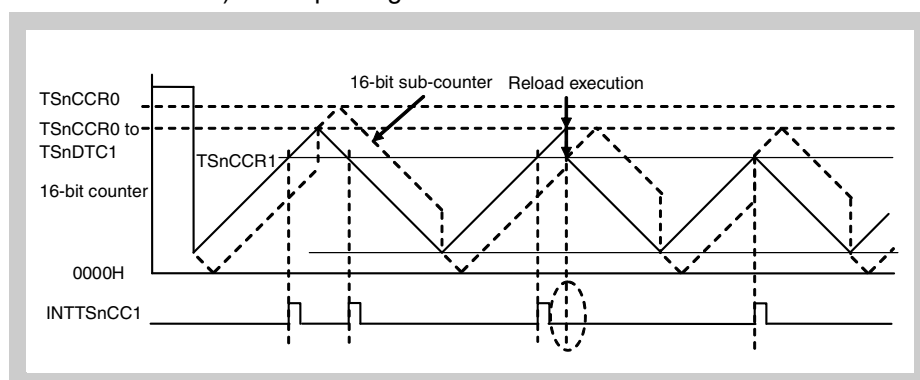


**Figure 17-114** Output waveform example when dead time is set

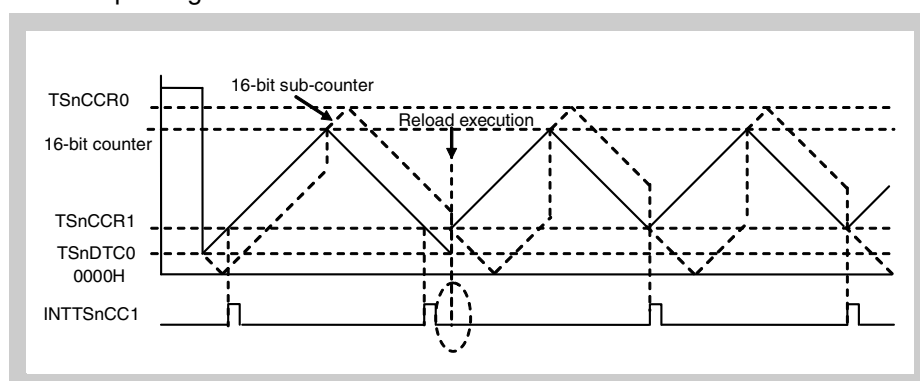
**(9) Cautions on dead time control in high-accuracy T-PWM mode****(a) Rewriting of TSnDTC0 and TSnDTC1 registers**

The setting of the dead time in the TSnDTC0, TSnDTC1 registers can be rewritten during operation. Note the following cautions when rewriting the dead time setting during operation.

- Caution**
1. Rewrite the TSnDTC0 and TSnDTC1 registers when using the reload function (TSnCMS = 0).
  2. When the TSnDTC0 and TSnDTC1 registers are rewritten, carrier-wave cycles will be changed. In cases where carrier-wave cycles should not be changed, rewrite the TSnCCR0 register value at the same time as changing the TSnDTC0 and TSnDTC1 registers.
  3. Rewriting is prohibited when TSnCMS = 1.
  4. In case of changing TSnCCR0 and TSnCCR1 at a 16-bit counter peak: Match interrupts (INTTSnCC1 to INTTSnCC5) will not occur immediately after reload execution if the values set in the TSnCCR1 to TSnCCR5 register matches with and TSnCCR0 – TSnDTC1 (the new maximum value of main counter) after updating.



5. In case of changing TS0DTC0 at a 16-bit counter trough: Match interrupts (INTTSnCC1 to INTTSnCC5) will not occur immediately after reload execution if the values set in the TSnCCR1 to TSnCCR5 register match with TS0DTC0 (the new minimum value of main counter) after updating.



**(10) Caution on rewriting cycles in high-accuracy T-PWM mode**

In high-accuracy T-PWM mode, setting conditions for the TSnCCR0, TSnDTC0, and TSnDTC1 registers are as follows.

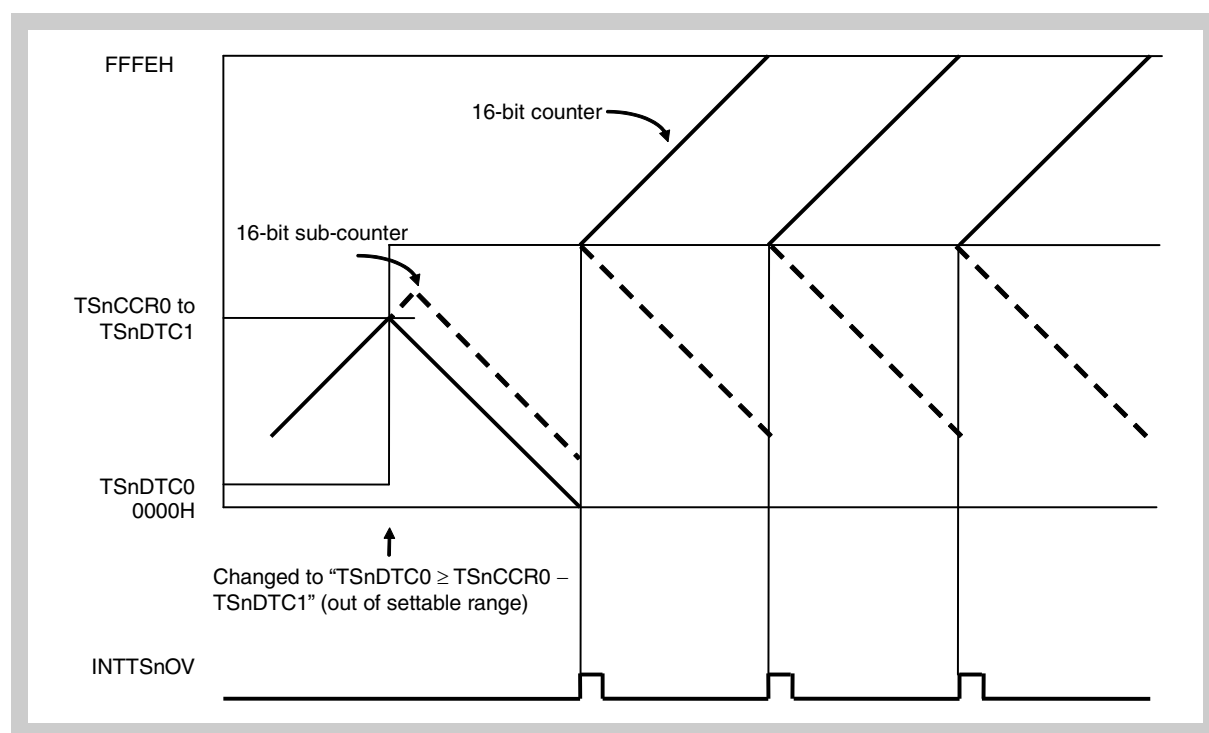
$$3 \times \text{MAX}(\text{TSnDTC0}, \text{TSnDTC1}) + \text{MIN}(\text{TSnDTC0}, \text{TSnDTC1}) < \text{TSnCCR0} \\ 0002\text{H} < \text{TSnCCR0} \leq \text{FFFEH}$$

MAX (A, B) indicates the greater value of A and B, and MIN (A, B) indicates the smaller value of A and B.

Figure 17-115 on page 828 shows an operation example when the setting range is exceeded.

This example shows the case where the TSnDTC0 register is set out of the range “TSnDTC0 ≥ TSnCCR0 – TSnDTC1”. Though the 16-bit counter executes count-down operation, the count-down operation is executed from 0000H because no match occurs. In this case, the count operation continues by loading the TSnDTC0 register setting value. However, no match with TSnCCR0 – TSnDTC1 occurs in the count-up operation, thus the 16-bit counter overflows. In this case, the count operation continues by loading the TSnDTC0 register setting value again.

An overflow interrupt (INTTSnOV) occurs when the 16-bit counter loads the TSnDTC0 register setting value from 0000H or when an overflow occurs at FFFEH, and then the TSnOVF flag is set. An overflow interrupt (INTTSnOV) does not occur if the TSnCCR0, TSnDTC0, and TSnDTC1 registers are set correctly, so this can be used for detecting incorrect settings.

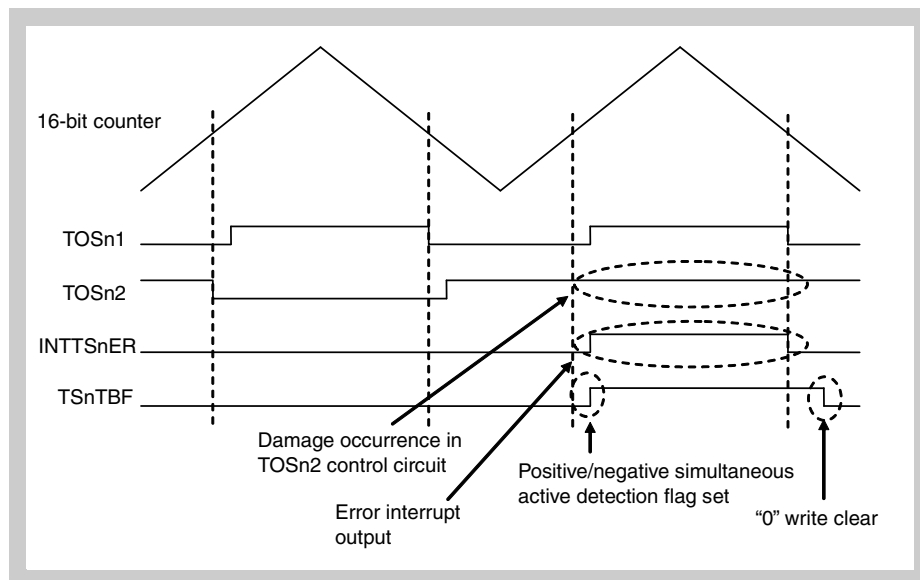


**Figure 17-115** Operation example setting is out of range



**(11) Error interrupt (INTTSnER) in high-accuracy T-PWM mode**

The positive/negative simultaneous active detection function can be used in the high-accuracy T-PWM mode. Error interrupts (INTTSnER) do not occur in the high-accuracy T-PWM mode. In case of occurrence, the internal circuits may be damaged.



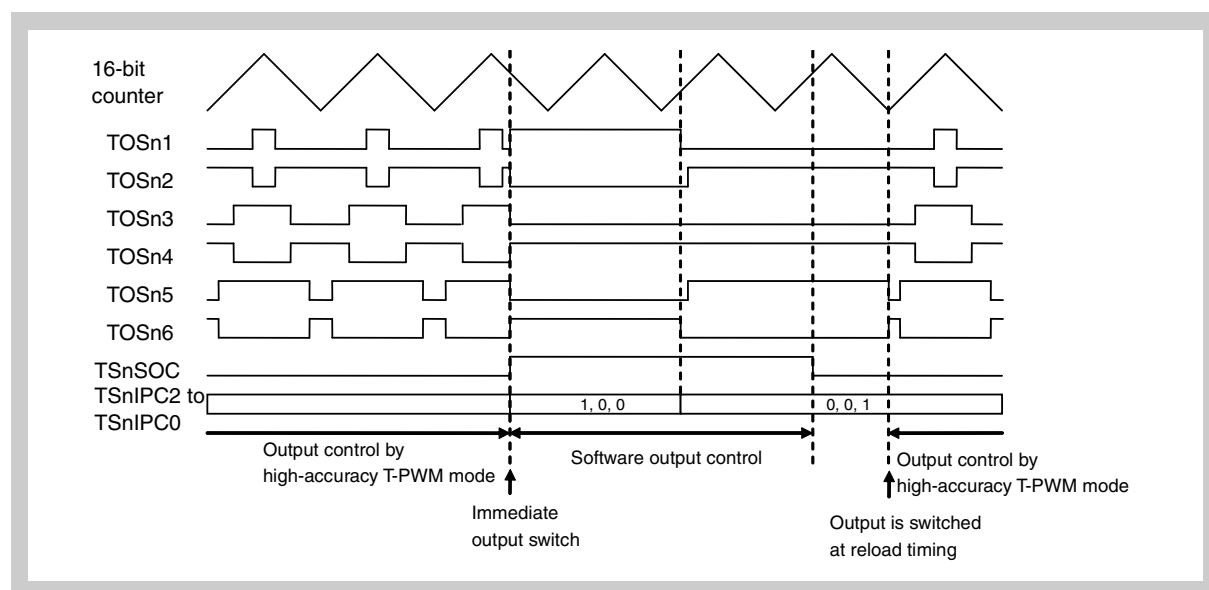
**Figure 17-116 Error interrupt operation example**

**(12) Software output control function in high-accuracy T-PWM mode**

In the high-accuracy T-PWM mode, timer output control can be performed through software control by using the TSnSOC and TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register, and the TSnIDC bit of the TSnOPT7 register.

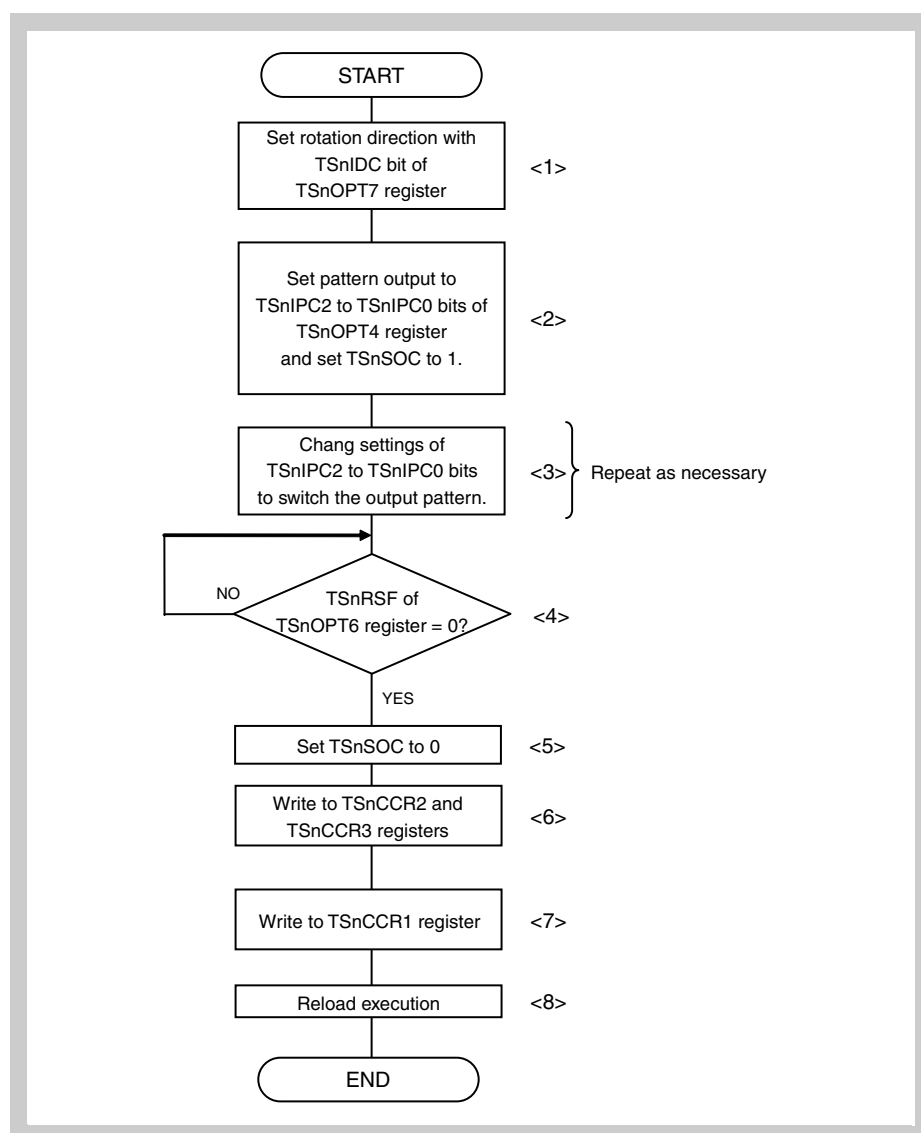
Output is switched immediately when TSnSOC is set to 1, as shown in *Figure 17-117 on page 830*. The dead time period is assured as long as the dead time is set. Output is retained until TSnSOC is cleared to 0. Then the function is switched to the output control by the high-accuracy T-PWM mode at the reload timing.

Refer to “Software output function” on page 936 for details on the software output control function.



**Figure 17-117** Software output control function switched from high-accuracy T-PWM mode

**Caution** Use the software output control function in the reload (batch rewrite) mode (TSnCMS of TSnOPT register = 0).

**(a) Software output control procedure****Figure 17-118 Software output control function processing flow (1/2)**

The procedure for software output control processing is as follows.

- <1> Set TSnIDC to determine the rotation direction. There is a 180° phase difference in the timer output between when TSnIDC = 0 and TSnIDC = 1. With the software output control function, the timer output is not changed only by rewriting of the bit.
- <2> Set the output pattern so as to output to TSnIPC2 to TSnIPC0 and also set TSnSOC to 1 to provide software output.
- <3> Change the output pattern setting of TSnIPC2 to TSnIPC0 to change the timer output. The settings of the following registers can be changed during software control.  
TSnCE bit of TSnCTL0 register, TSnOPT1 to TSnOPT4 registers, TSnIDC and TSnTOS bits of TSnOPT7 register, TSnCCR0 to TSnCCR5 registers, TSnDTC0 register, and TSnDTC1 register
- <4> Check that the reload request flag (TSnRSF) is 0. If TSnRSF = 1, do not proceed to the next step before TSnRSF is cleared to 0.

- <5> Software control release is started by clearing TSnSOC to 0 (Not yet released in this step).
- <6> Set the compare register required after the release of software output control. Proceed to the next step if changing is not needed. Change the register that has the reload function in this step, if necessary.
- <7> Write to the TSnCCR1 register to start reload.
- <8> Reload is executed and software output is released.

- 
- Caution**
1. Do not change TSnIPC2 to TSnIPC0 once after clearing TSnSOC to 0.
  2. Be sure to execute reload after execution of procedures <4>, <5>, <6> and <7>. Software output cannot be released if reload cannot be executed.
-

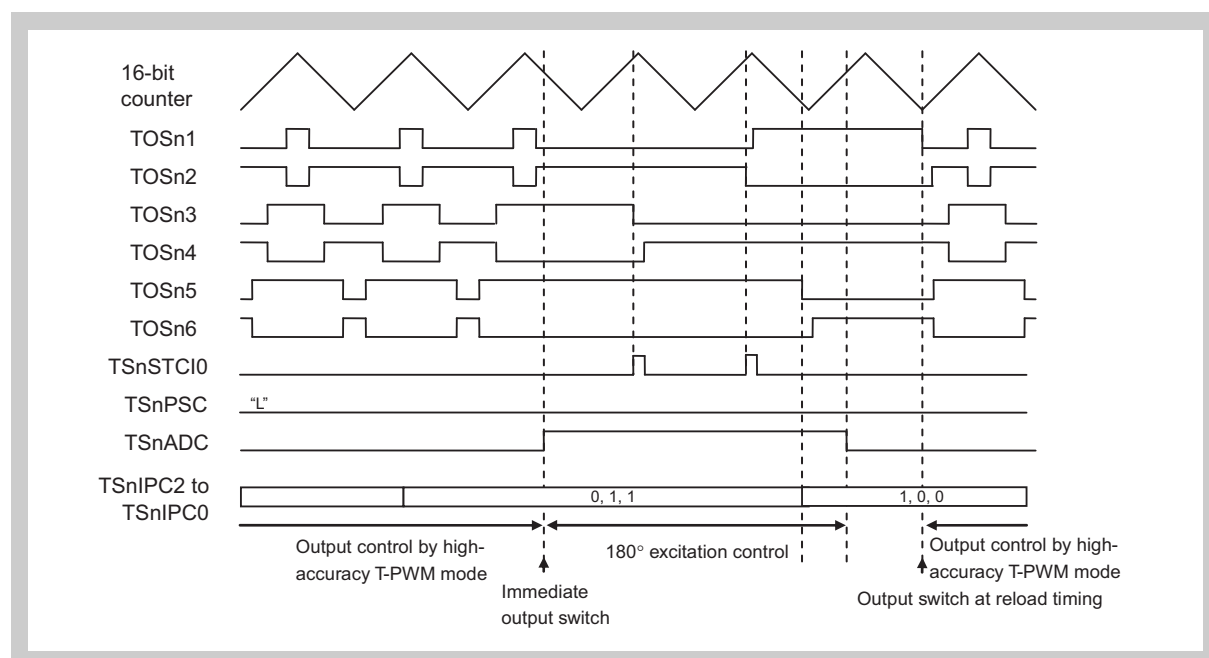
**(13) 180° excitation control in high-accuracy T-PWM mode**

In the high-accuracy T-PWM mode, the 180° excitation control function can be used by using the TSnADC and TSnPOT bits of the TSnOPT5 register, the TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register and the TSnIDC bit of the TSnOPT7 register.

Output is switched immediately when TSnADC is set to 1, as shown in *Figure 17-119 on page 833*. The dead time period is assured as long as the dead time is set. Then, when a trigger is input to the output pattern switch trigger (TSnSTCI1 and TSnSTCI0 signals or TAPTSn2 to TAPTSn0 pins), the output is changed in accordance with the setting of the TSnPSC bit of the TSnOPT4 register. Output is retained until TSnADC is cleared to 0. Then the function is switched to the output control by the high-accuracy T-PWM mode, at the reload timing.

Be sure to use the 180° excitation control function in reload (batch rewrite) mode (TSnCMS of TSnOPT0 register = 0).

Refer to “180° excitation control function” on page 938 for details on 180° excitation control.



**Figure 17-119 Example of switching 180° excitation control function in high-accuracy T-PWM mode**

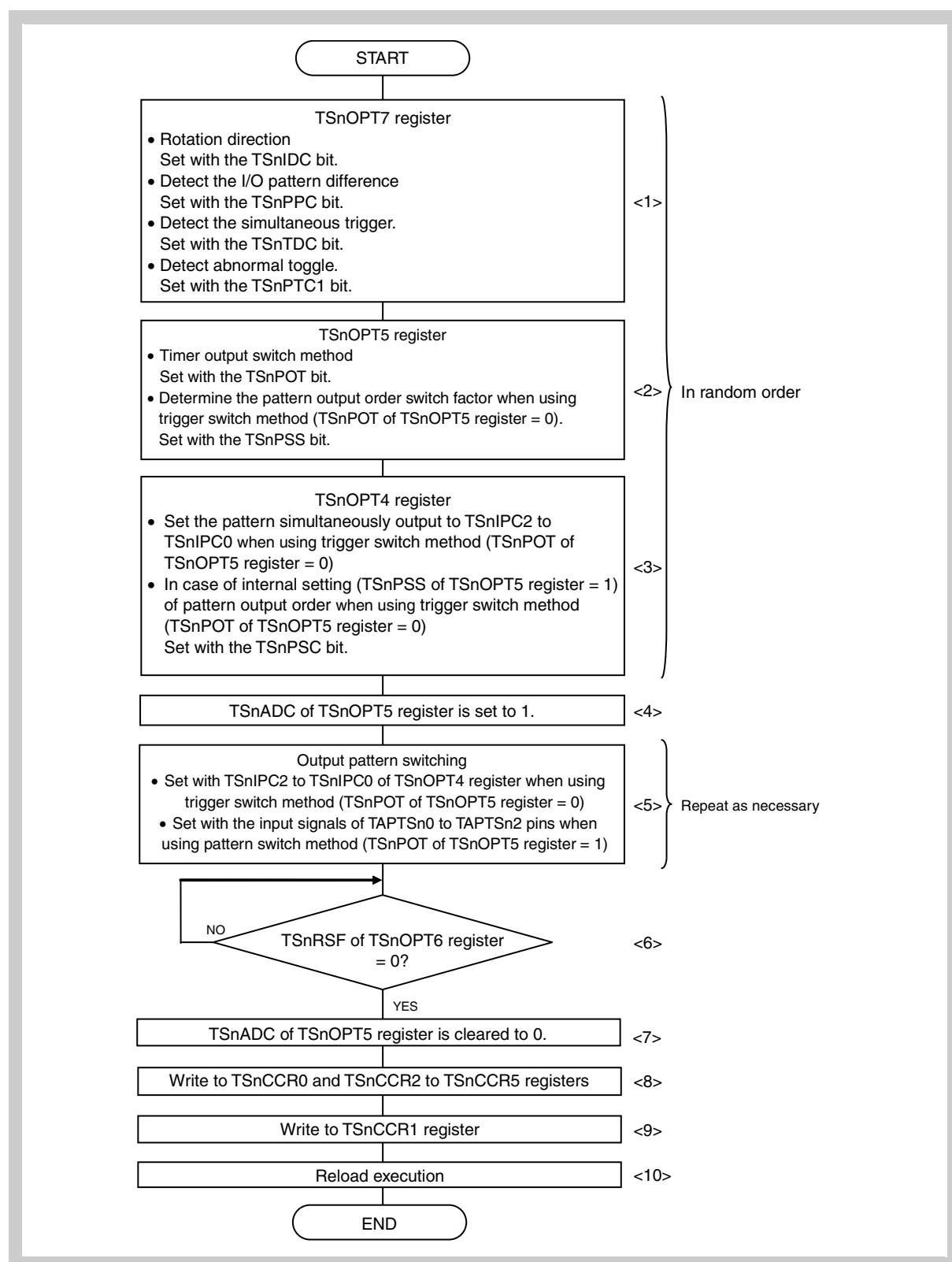
**(a) 180° excitation control processing procedure**

Figure 17-120 180° excitation control function processing flow

The procedure for 180° excitation control is as follows.

- <1> TSnOPT7 register setting
  - Set a value to the TSnIDC bit to select the rotation direction.
  - Set a value to the TSnPPC bit to select enable (1) or disable (0) of the pattern phase difference detection flag (TSnPPF) of input (TAPTSn2 to TAPTSn0 pins) or output (TSnOPF2 to TSnOPF0 flags) with the trigger switch method (TSnPOT of TSnOPT5 register = 1).
  - Set a value to the TSnTDC bit to select enable (1) or disable (0) of the TSnSTCI0, TSnSTCI1 signal simultaneous trigger detection flag (TSnTDF) with the trigger switch method.
  - Set values to the TSnPTC1 and TSnPTC0 bits to select enable (1) or disable (0) of the TAPTSn2 to TAPTSn0 pin abnormal toggle detection flag (TSnPTF).
- <2> TSnOPT5 register setting
  - Set a value to the TSnPOT bit to select the pattern output trigger. The pattern switch method with which the output is switched with the input pattern of the TAPTSn2 to TAPTSn0 pins can be enabled by clearing TSnPOT to 0, and the trigger switch method with which the output is switched at a rising edge of the TSnSTCI0 and TSnSTCI1 signals can be enabled by setting TSnPOT to 1.
  - Set a value to the TSnPSS bit to select the pattern output order switch factor.
- <3> TSnOPT4 register setting
  - Set a value to the TSnPSC bit to select the pattern output order for when TSnPSS of the TSnOPT5 register is 1 by using the trigger switch method.
  - Set values to the TSnIPC2 to TSnIPC0 to select the initial pattern for the trigger switch method.
- <4> Start the 180° excitation control function by setting TSnADC to 1.
- <5> Output pattern switching
  - Pattern switch method
    - With the pattern switch method, the output is switched with the input pattern of TAPTSn2 to TAPTSn0 pins. However, the previous output level is held until [1, 1, 1]0 or [0, 0, 0] is input to the TAPTSn2 to TAPTSn0 pins.
  - Trigger switch method
    - With the trigger switch method, the output is switched at a rising edge of the TSnSTCI0 and TSnSTCI1 signals. Set the appropriate value to TSnIPC2 to TSnIPC0 to forcibly change the timer output.
    - Writing to the TSnIPC2 to TSnIPC0 is prior to the rising edge of the TSnSTCI0 and TSnSTCI1 signals if both occur simultaneously.
  - Switch method can be changed during operation.
  - The registers that can be rewritten during 180° excitation control function operation are as follows.  
TSnCE bit of TSnCTL0 register, TSnOPT1 to TSnOPT5 registers, TSnOPT7 register, TSnCCR0 to TSnCCR5 registers, TSnDTC0 register, TSnDTC1 register

---

**Caution** Setting TSnSOC of the TSnOPT4 register to 1 and TSnADC of TSnOPT5 register to 1 is prohibited.

---

**Note** <1> to <3> are in random order.

- 
- <6> Check that the reload request flag (TSnRSF) is 0. If TSnRSF = 1, do not proceed to the next step before TSnRSF is cleared to 0.
  - <7> Release of 180° excitation control is started by clearing TSnADC to 0 (Not yet released in this step).  
After TSnADC is changed to 0, output is switched at the switch timing of the pattern/trigger switch method until reload is executed. The output level can be changed forcibly by writing to TSnIPC2 to TSnIPC0.
  - <8> Set the compare register required after the release of 180° excitation control. Proceed to the next step if changing is not needed. Change the register that has the reload function in this step if necessary.
  - <9> Write to the TSnCCR1 register to start reload.
  - <10> Reload is executed and 180° excitation control is released.

---

**Caution** Be sure to execute reload after execution of steps <6>, <7>, <8> and <9>. 180° excitation control cannot be released if reload cannot be executed.

---



### (b) 180° excitation control switch timing from high-accuracy T-PWM mode

Output is switched immediately when switching from the high-accuracy T-PWM mode to 180° excitation control. Dead time control by hardware is inserted if the on-going output level is reversed. The output level is held until the reload is performed when switching from 180° excitation control to the high-accuracy T-PWM mode.

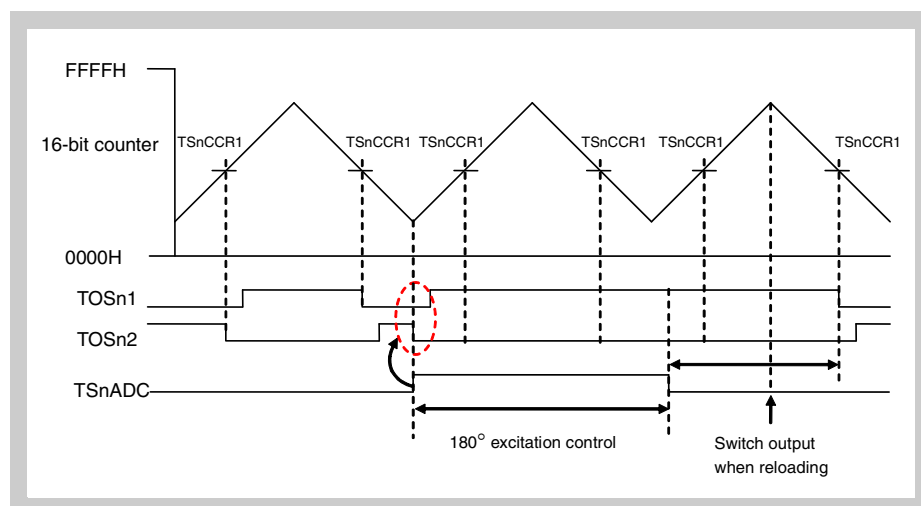


Figure 17-121 Output when switching between high-accuracy T-PWM Mode and 180° excitation control (in case of output reverse)

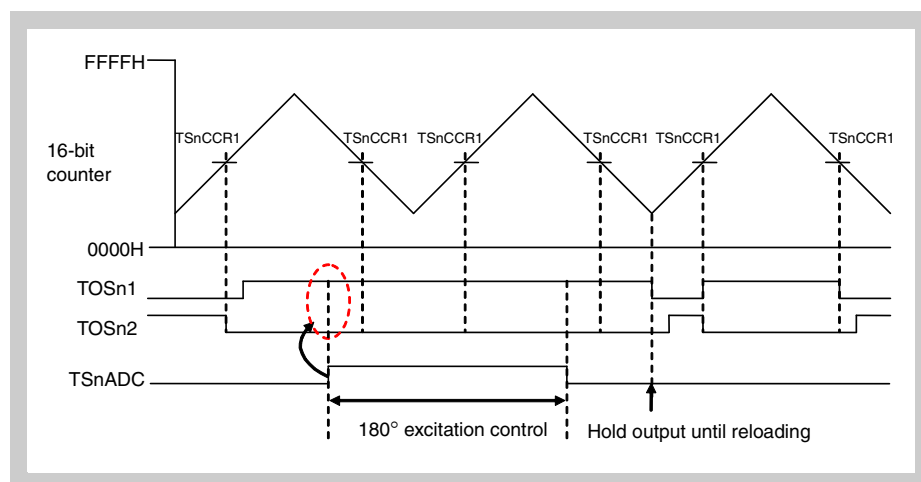
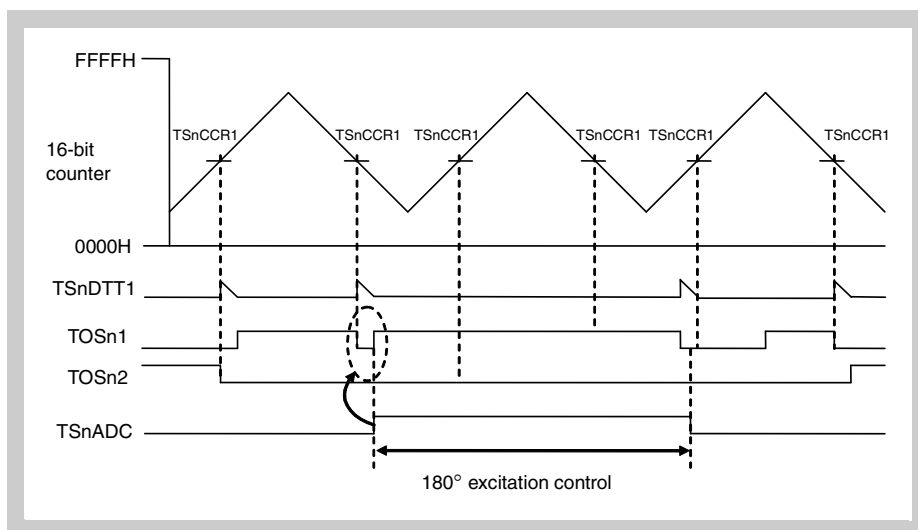


Figure 17-122 Output when switching between high-accuracy T-PWM Mode and 180° excitation control (in case of output not reversed)

The output switched during the dead time count is set without waiting for the dead time count to finish in case of the same phase, or is set after the dead time count finishes in case of the negative phase.

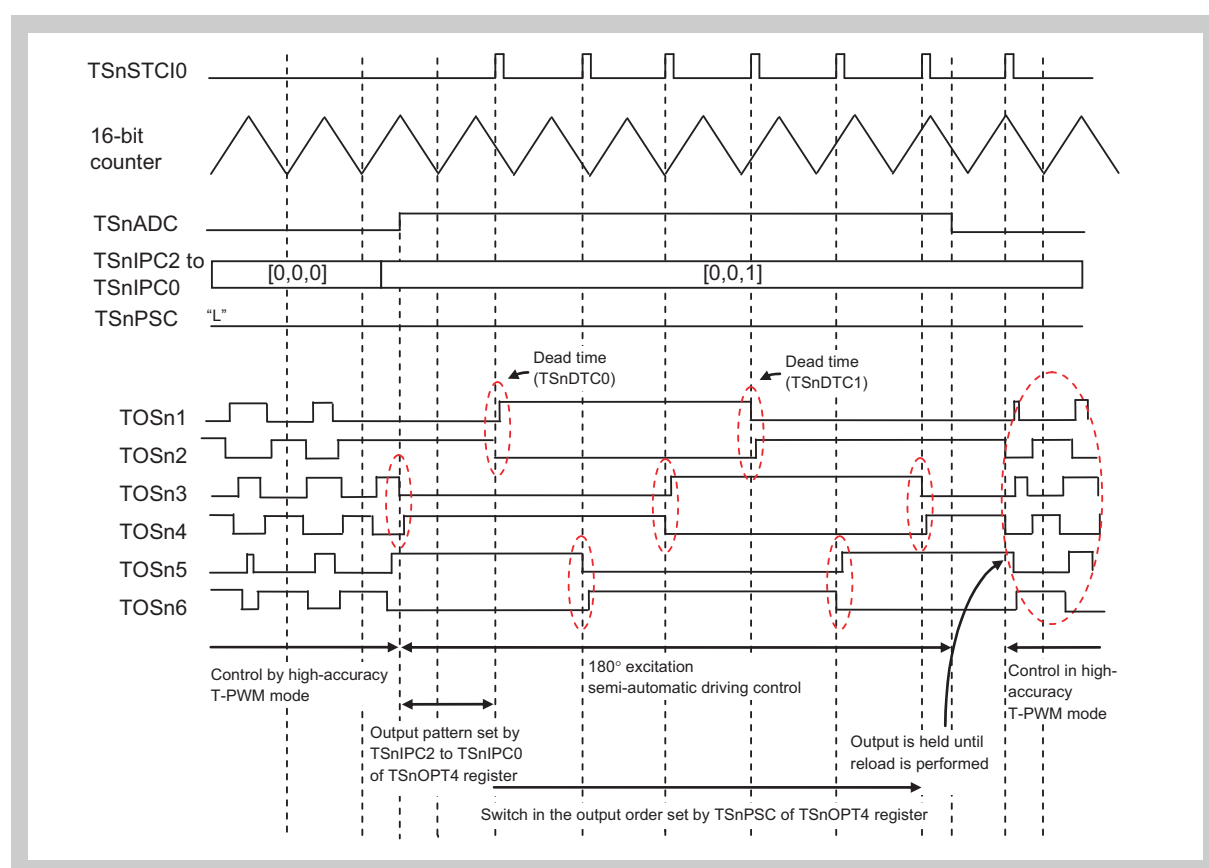


**Figure 17-123** Output when switching between high-accuracy T-PWM mode and 180° excitation control (during dead time count)

**(14) Semi-automatic driving system using high-accuracy T-PWM mode**

This section explains the system in which the semi-automatic driving function is used in the high-accuracy T-PWM mode. *Figure 17-124 on page 839* shows the timing. The 180° excitation control function is automatically enabled by setting TSnADC of the TSnOPT5 register to 1. As shown here, the output pattern is switched by the TSnSTCI0 signal as a trigger in the 180° excitation control function, and the dead time set with the TSnDTC0 and TSnDTC1 registers is always added when switching.

Output is switched immediately when switching from the high-accuracy T-PWM mode to 180° excitation control, but control is switched after waiting for the reload timing when switching from 180° excitation control to the high-accuracy T-PWM mode.



**Figure 17-124** Output example of semi-automatic driving system using high-accuracy T-PWM mode

**Note** Setting example

TSnCTL1 register: TSnMD3 to 0 = [1, 0, 0, 0], TSnIOC0 register:  
 TSnOE6 to 1 = [1, 1, 1, 1, 1, 1, 1, 1], TSnOL6 to 1 = [0, 0, 0, 0, 0, 0, 0, 0],  
 TSnOPT4 register: TSnIPC2 to 0 = [0, 0, 1], TSnPSC = 0, TSnOPT5 register:  
 TSnADC = 1

### 17.10.9 PWM mode with dead time

#### (1) Overview of PWM mode with dead time

In the PWM mode with dead time, 6-phase PWM is generated to output from the TOSn1 to OSn6 pins by using the saw-tooth wave operation of the 16-bit counter and four 16-bit compare registers. The maximum value of the 16-bit counter is set with the TSnCCR0 register. The duty of the U, V, and W-phase voltage data signal is set with the TSnCCR1 to TSnCCR3 registers. Dead time is set with the TSnDTC0 and TSnDTC1 registers. The dead time from the negative phase to the positive phase and the dead time from the positive phase to the negative phase can be set with TSnDTC0 and TSnDTC1, respectively. The 16-bit counter counts up by setting 0000H as the minimum value. When matching with the maximum value (cycle), the 16-bit counter is cleared (0000H), and continues counting up.

The 10-bit dead time counter (TSnDTT1 to TSnDTT3) reloads the setting value of the TSnDTC0 and TSnDTC1 registers upon a match between the values of the 16-bit counter and the TSnCCR1 to TSnCCR3 registers to count down. Compare match interrupts (INTTSnCC0 to INTTSnCC3) occur upon respective match between the values of the 16-bit counter and TSnCCR0 to TSnCCR3 registers.

**Note** PWM mode with dead time is valid when TSnMD3 to TSnMD0 of the TSnCTL1 register are set to [1, 0, 0, 1].

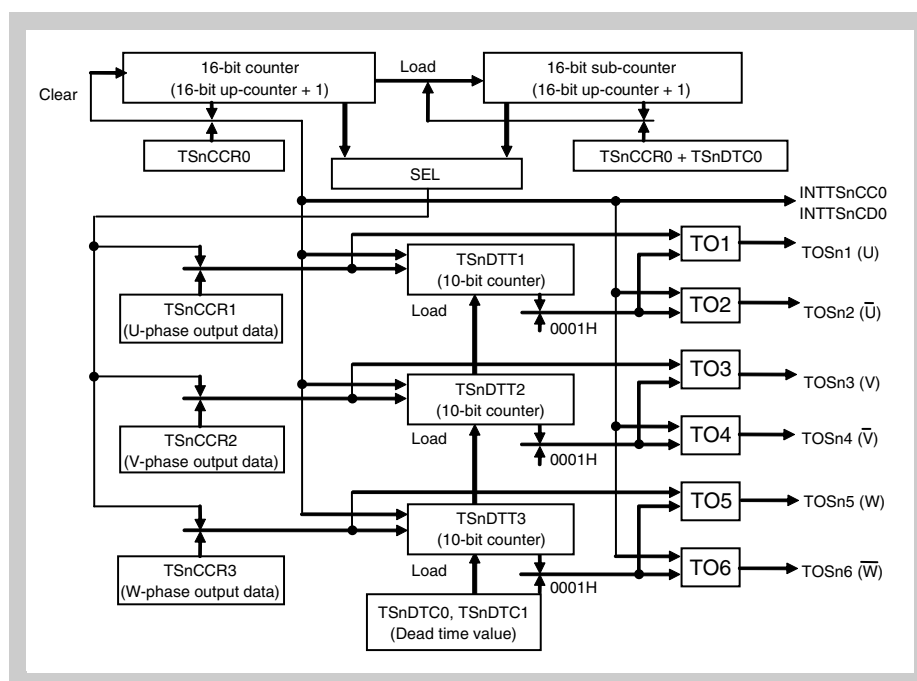
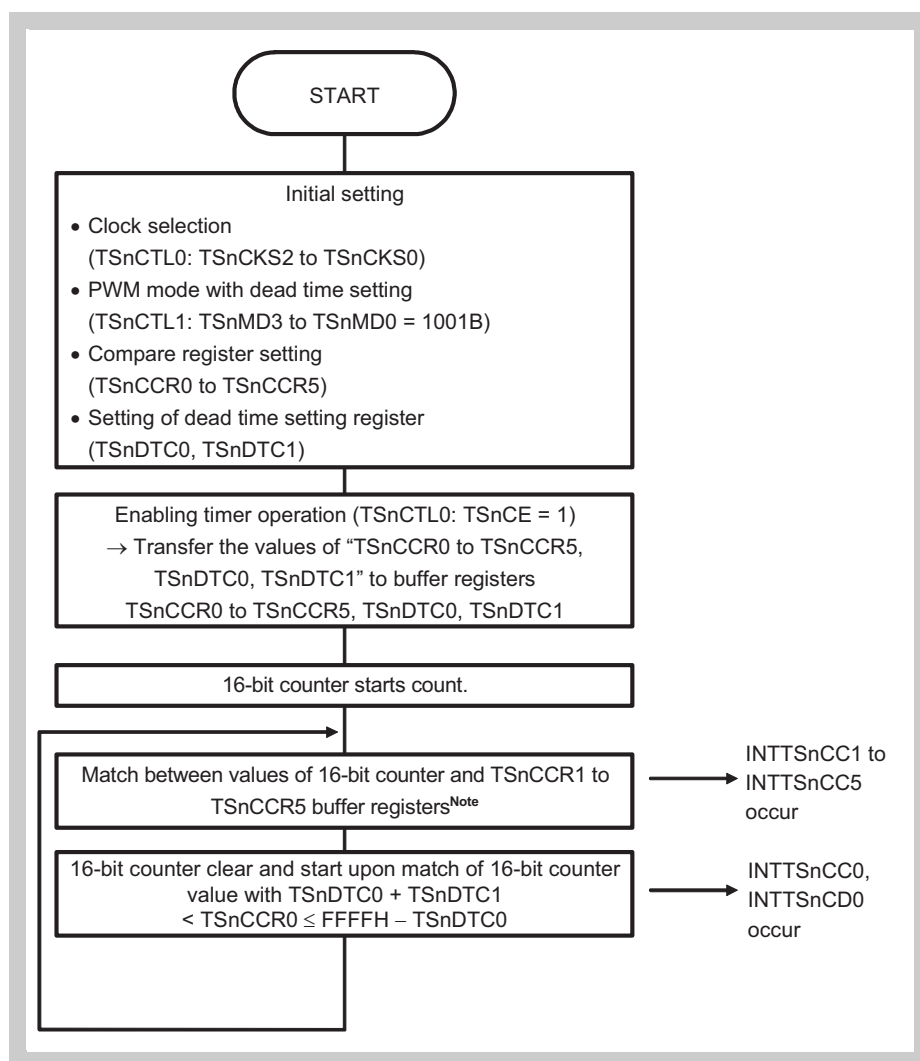


Figure 17-125 Block diagram in PWM mode with dead time



**Figure 17-126 Basic operation flow in PWM mode with dead time**

**Note** The 16-bit counter is not cleared upon a match between the values of the 16-bit counter and the TSnCCR1 to TSnCCR5 buffer registers.

**(2) PWM Mode with Dead Time Operation List****(a) Register rewriting**

Register	Rewriting method	Rewriting during operation	Function
TSnCCR0	Reload	Possible	Cycle
TSnCCR1 to TSnCCR3	Reload	Possible	PWM duty
TSnCCR4, TSnCCR5	Reload	Possible	PWM duty (Selectable as A/D conversion trigger)
TSnDTC0, TSnDTC1	Reload	Conditionally possible <sup>Note</sup>	Cycle, dead time

**Note** Refer to “Rewriting of TSnDTC0 and TSnDTC1 registers” on page 827 for details.

**(b) Input pin**

Pin	Function
TTRGSn	-
TEVTSn	-

**(c) Output pin**

Pin	Function
TOSn0	Toggle output by TSnCCR0 compare match
TOSn1	PWM output (with dead time) by TSnCCR1 compare match
TOSn2	Negative phase output (with dead time) to TOSn1
TOSn3	PWM output (with dead time) by TSnCCR2 compare match
TOSn4	Negative phase output (with dead time) to TOSn3
TOSn5	PWM output (with dead time) by TSnCCR3 compare match
TOSn6	Negative phase output (with dead time) to TOSn5
TOSn7	Pulse output by A/D conversion trigger

**(d) Interrupt**

Interrupt	Function
INTTSnCC0 to INTTSnCC5	Compare match of TSnCCR0 to TSnCCR5 registers
INTTSnOV	-
INTTSnER	Error
INTTSnOD	-
INTTSnCD0	Peak interrupt (occurs at the same timing with INTTSnCC0)
INTTSnWN	-

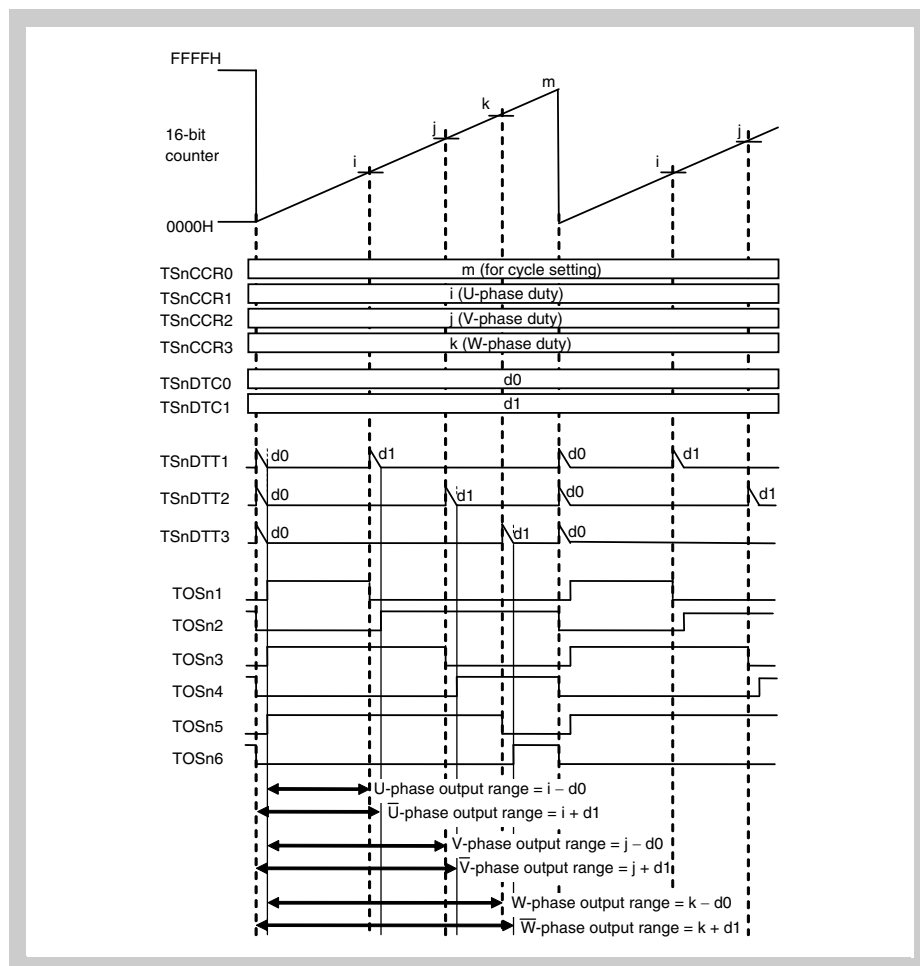
**(e) Compare match timing**

Compare match	Timing
TSnCCR0	When 16-bit counter switches from TSnCCR0 to 0000H
TSnCCR1 to TSnCCR5	After detection of match between 16-bit counter and TSnCCR1 to TSnCCR5

**(f) Output condition**

Output	Condition
Duty 0% output	TSnCCRm = 0000H
Duty 100% output	TSnCCR0 + TSnDTC0 + 1 = TSnCCRm
Dead time check	TSnDTC0, TSnDTC1

- Note**
1. “–” indicates an unused function in the PWM mode with dead time.
  2.  $m = 1$  to 3,  $n = 0, 1$



**Figure 17-127** Output waveform example in PWM mode with dead time

- Caution**
1. The maximum value settable to the TSnCCR1 to TSnCCR3 registers is “TSnCCR0 + TSnDTC0 + 1”.
  2. PWM output is provided with duty 0% when 0000H is set to the TSnCCR1 to TSnCCR3 registers.

3. PWM output is provided with duty 100% when “TSnCCR0 + TSnDTC0 + 1” is set to the TSnCCR1 to TSnCCR3 registers.
  4. Set the TSnCCR0 register to satisfy the condition “TSnDTC0 + TSnDTC1 < TSnCCR0 ≤ FFFFH – TSnDTC0”.
- 

### (3) Setting of PWM mode with dead time

#### (a) Mode setting

The PWM mode with dead time is set by setting TSnMD4 to TSnMD0 of the TSnCTL1 register to [1, 0, 0, 1].

#### (b) Output level/output enable setting

Output level/output enable is set by setting the TSnOL0 to TSnOL7 and TSnOE0 to TSnOE7 bits of the TSnIOC0 register.

Toggle output is provided from the TOSn0 pin upon the cycle match (match between the 16-bit counter and TSnCCR0 register).

Pulse output by A/D conversion trigger is provided from the TOSn7 pin. Set this pin as needed.

#### (c) Error interrupt occurrence enabling

Error interrupt (INTTSnER) occurrence is enabled by setting the TSnEOC bit of the TSnIOC4 register to 1 when the simultaneous positive/negative phase state is detected.

A pin that detects simultaneous active state is set by setting the TSnTBA2 to TSnTBA0 bits of the TSnIOC4 register. In the PWM mode with dead time, positive and negative phases do not become active even if any value is set to the TSnCCR0 to TSnCCR3 registers.

#### (d) Interrupt and thinning out function setting

A peak interrupt (INTTSnCD0) occurs upon a match between the TSnCCR0 register and 16-bit counter. (Trough interrupts do not occur). Set TSnICE of the TSnOPT1 register to 1 to output peak interrupts. To use the thinning out function for peak interrupt, execute setting with the TSnID4 to TSnID0 bits of the TSnOPT1 register.

#### (e) Reload thinning out function setting

Set TSnRDE of the TSnOPT2 register to 1 to set reload timing to the same timing as interrupt timing. Reload is performed when TSnRTE of the TSnOPT1 register is set to 1.



**(f) A/D conversion trigger output setting**

A/D conversion trigger 0 (TSnADTRG0 signal) is set by setting the TSnAT04, TSnAT02, TSnAT01 bits of the TSnOPT2 register. The TSnAT04, TSnAT02, and TSnAT01 bits specify whether to enable/disable A/D conversion trigger output upon match with the TSnCCR5 register, upon match with the TSnCCR4 register, and at the peak interrupt (INTTSnCD0) of the 16-bit counter or 16-bit sub-counter.

A/D conversion trigger 1 (TSnADTRG1 signal) is set by setting the TSnAT14, TSnAT12, and TSnAT11 bits of the TSnOPT3 register. The TSnAT14, TSnAT12, and TSnAT11 bits specify whether to enable/disable A/D conversion trigger output upon match with the TSnCCR5 register, upon match with the TSnCCR4 register, and at the peak interrupt (INTTSnCD0) of the 16-bit counter or 16-bit sub-counter.

Compare values are set with the TSnCCR4 and TSnCCR5 registers. Thinning out function is enabled for the TSnADTRG0 and TSnADTRG1 signals.

No thinning out, 1-thinning out, 3-thinning out and 7-thinning out setting can be executed with the TSnACC01 and TSnACC00 bits of the TSnOPT2 register, and the TSnACC11 and TSnACC10 bits of the TSnOPT3 register.

- 
- Caution**
1. Setting TSnAT07, TSnAT06, TSnAT05, TSnAT03, TSnAT00 of the TSnOPT2 register to 1 is prohibited.
  2. Setting TSnAT17, TSnAT16, TSnAT15, TSnAT13, TSnAT10 of the TSnOPT3 register to 1 is prohibited.
- 

**(g) Dead time setting**

Dead time is set with the TSnDTC0 and TSnDTC1 registers.

Dead time can be calculated with the following equation.

16-bit counter operation clock cycle × TSnDTC0

16-bit counter operation clock cycle × TSnDTC1

The time for the change from the inactive state of the TOSn2, TOSn4, and TOSn6 pins to the active state of the TOSn1, TOSn3, and TOSn5 pins can be set with the TSnDTC0 register.

The time for the change from the inactive state of the TOSn1, TOSn3, and TOSn5 pins to the active state of the TOSn2, TOSn4, and TOSn6 pins can be set with the TSnDTC1 register.

**(h) Duty (PWM width) setting**

U, V and W-phase duties are set with the TSnCCR1 to TSnCCR3 registers respectively. The setting range of the TSnCCR1 to TSnCCR3 registers is as follows.

$$0000H \leq \text{TSnCCR1} - \text{TSnCCR3} \leq \text{TSnCCR0} + \text{TSnDTC0} + 1$$

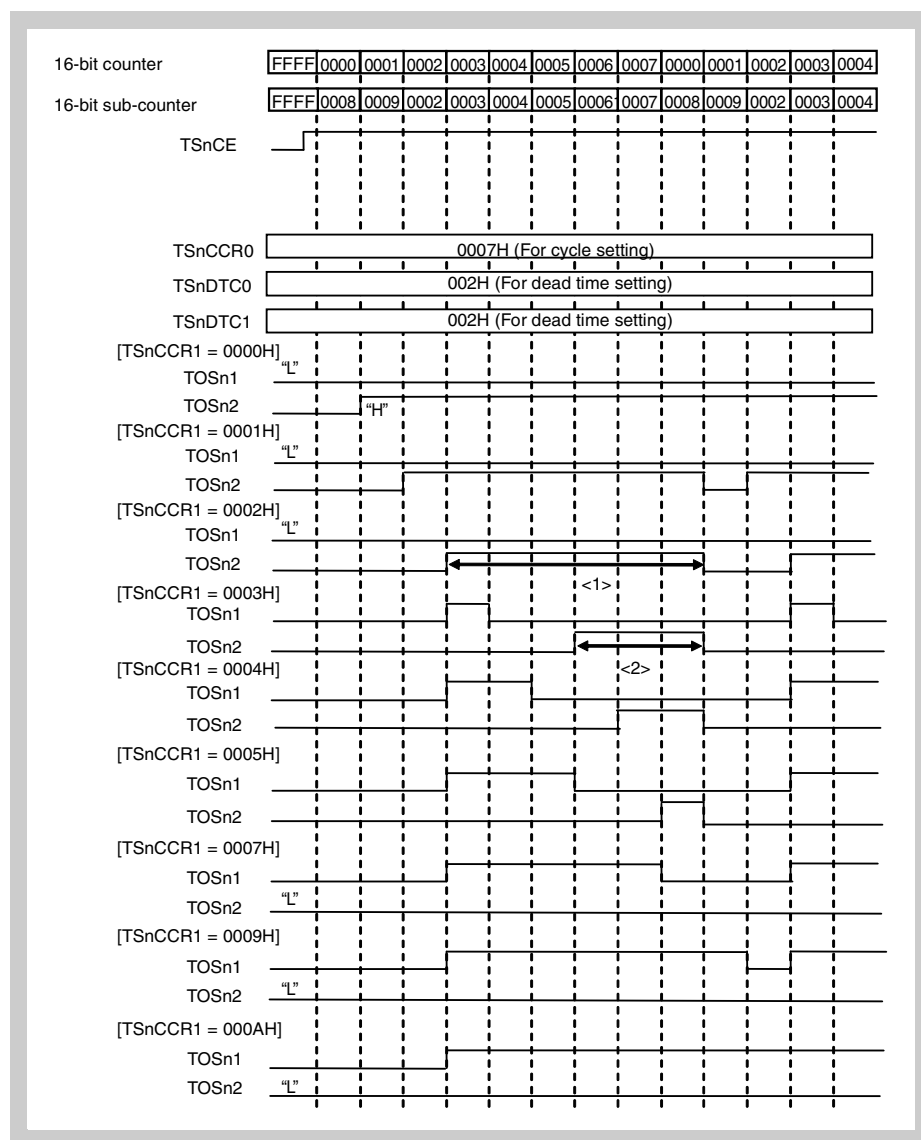
Set the TSnCCR0 register to satisfy the condition “TSnDTC0 + TSnDTC1 < TSnCCR0 ≤ FFFFH – TSnDTC0”.

**(4) Operation in PWM mode with dead time**

Figure 17-128 on page 846 shows the timing when setting  $\text{TSnCCR0} = 0007\text{H}$ ,  $\text{TSnDTC0} = 0002\text{H}$ ,  $\text{TSnDTC1} = 0002\text{H}$ , and setting the  $\text{TSnCCR0}$  register to  $0000\text{H}$  to  $0007\text{H}$  (in part).

The PWM width increases/decreases by 1 count clock if the compare value of the  $\text{TSnCCR1}$  register increases/decreases. Arrows indicate the increase/decrease by the  $\text{TSnDTC1} + 1$  count clock in Figure 17-128 on page 846.

This occurs when the value is rewritten to " $\text{TSnCCR1} + 0001\text{H}$ " which will be " $\text{TSnDTC1} < \text{TSnCCR1}$ " because the dead time control is valid.



**Figure 17-128** Timer output example in case of setting  $\text{TSnCE} = 1$  (initial setting in PWM mode with dead time)

<1> PWM width without dead time

<2> PWM width decreases by  $\text{TSnDTC1} + 1$  because the PWM width with dead time is " $\text{TSnCCR1} = \text{TSnDTC0} + 0001\text{H}$ ".

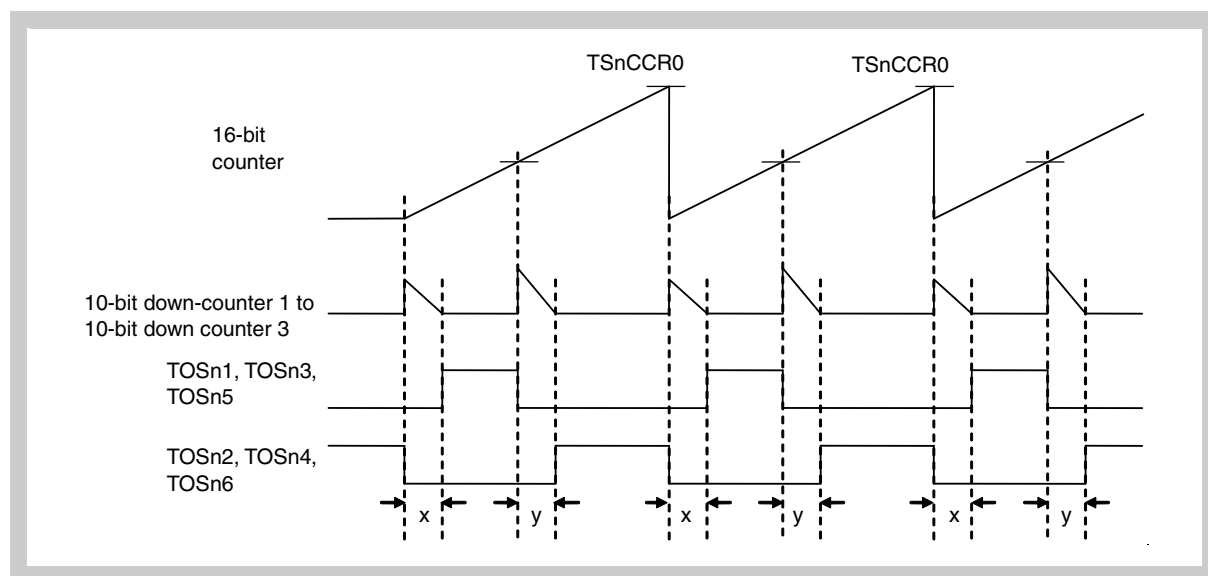
**(5) Dead time control in PWM mode with dead time**

In the PWM mode with dead time, the duty setting registers are TSnCCR1 to TSnCCR3, and the cycle setting register is TSnCCR0.

Duty variable type 6-phase PWM waveform output is provided by using these four registers.

There are three 10-bit down-counters that synchronously operate with the count clock of the 16-bit counter, and dead time setting registers (TSnDTC0, TSnDTC1) in order to achieve dead time control. Dead time from the negative phase changes to inactive until the positive phase changes to active is set by the TSnDTC0 register, and the dead time from the positive phase changes to inactive until the negative phase changes to active is set by the TSnDTC1 register.

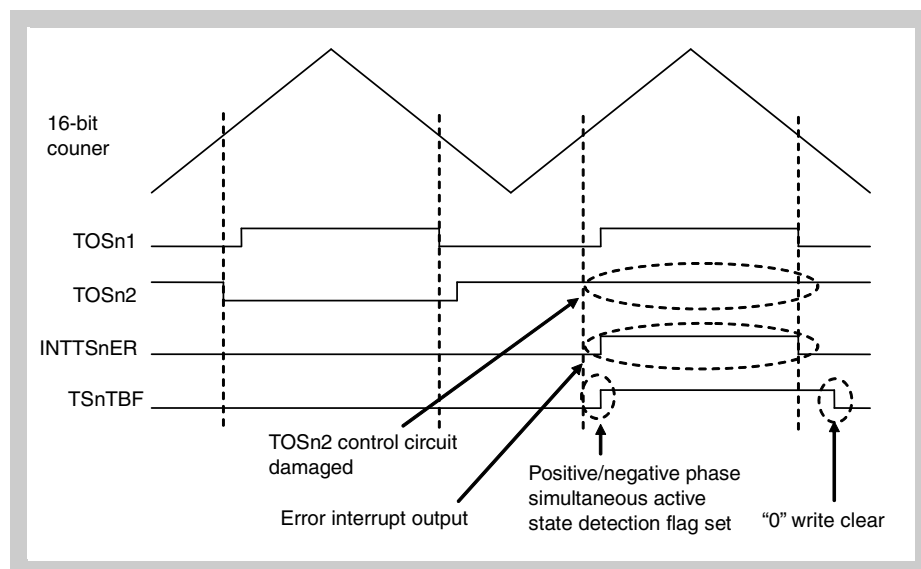
Then, the output waveform in case of TSnDTC0 = x, TSnDTC1 = y is shown in Figure 17-129 on page 847.



**Figure 17-129** Output waveform example in PWM mode with dead time

**(6) Error interrupt (INTTSnER) in T-PWM mode with dead time**

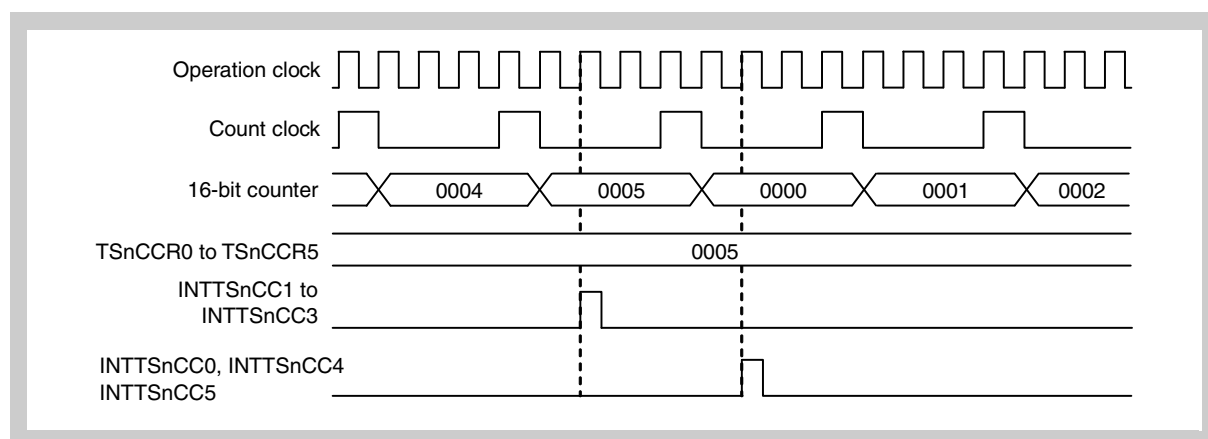
The positive/negative phase simultaneous active state detection function is enabled in the T-PWM mode with dead time. Error interrupts (INTTSnER) do not occur in the high-accuracy T-PWM mode. In case of occurrence, the internal circuit may be damaged.



**Figure 17-130 Error interrupt operation example**

**(7) Caution about interrupt timing in T-PWM mode with dead time**

The interrupt output timing differs between "INTTSnCC0, INTTSnCC4, INTTSnCC5" and "INTTSnCC1, INTTSnCC2, INTTSnCC3". Therefore, interrupt timing will not be the same even if the same value is set.



**Figure 17-131 Interrupt timing example in T-PWM mode with dead time**

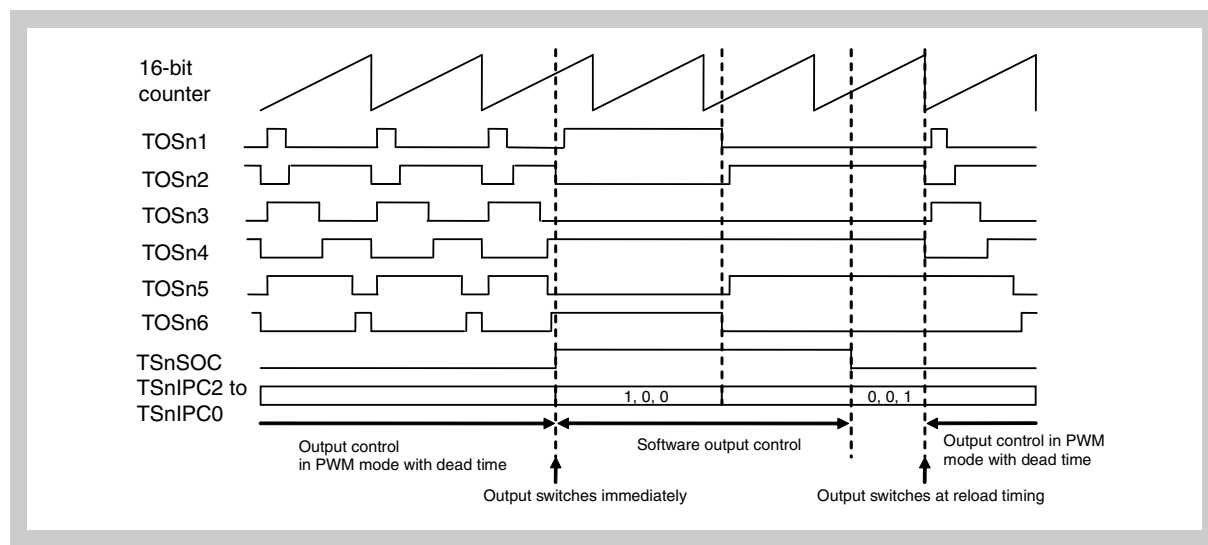
**(8) Software output control function in PWM mode with dead time**

Timer output control can be executed with software control by using the TSnSOC and TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register, and the TSnIDC bit of the TSnOPT7 register.

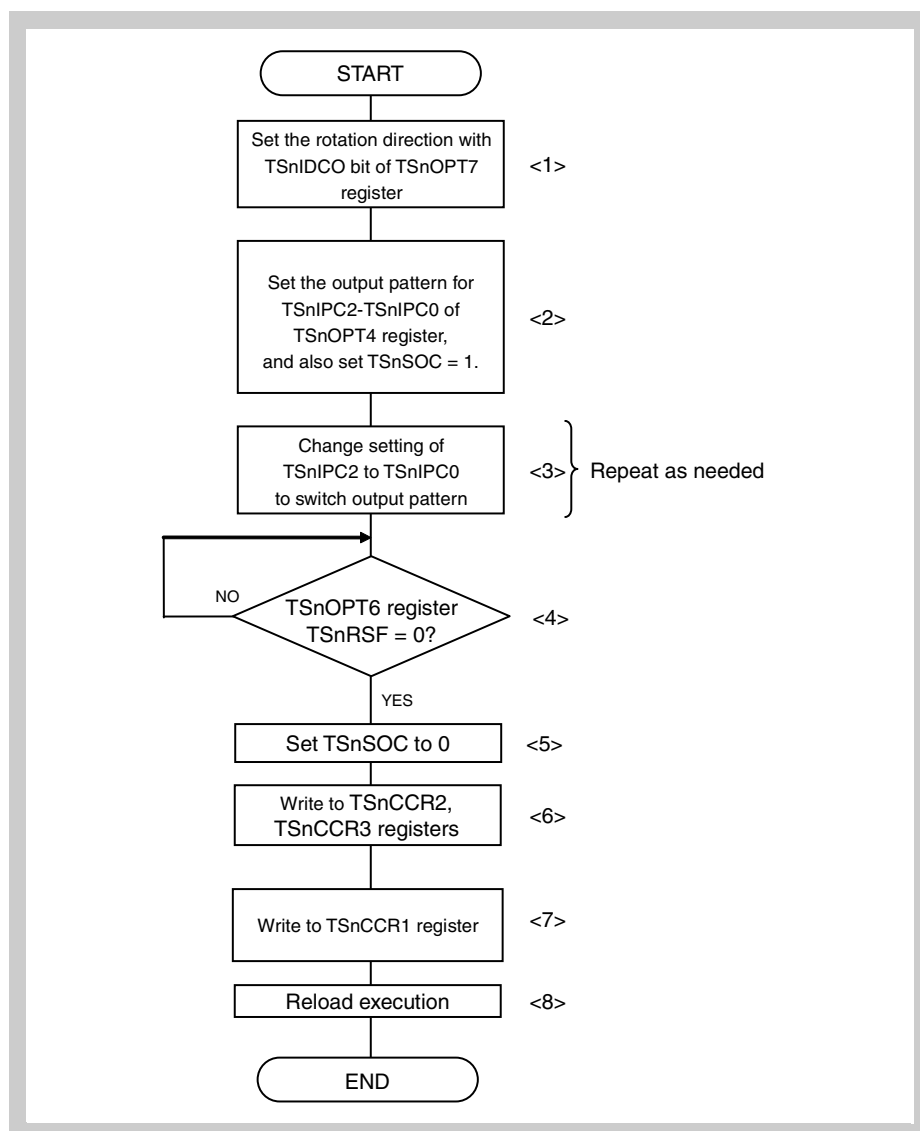
Output is switched immediately at the time of setting TSnSOC = 1 as shown in *Figure 17-132*. Dead time period is assured by setting the dead time.

Output is held at the time of setting TSnSOC = 0. Then the function is switched to the output control in the PWM mode with dead time at reload timing occurrence.

Refer to “Software output function” on page 936 for details on the software output control function.



**Figure 17-132 Software output control function switch example from PWM mode with dead time**

**(a) Software output control procedure****Figure 17-133 Processing flow in software output control**

The software output control processing procedure is described below.

- <1> Set TSnIDC in order to determine rotation direction. There is a 180° phase difference in the timer output between when TSnIDC = 0 and TSnIDC = 1. In the software output control function, the timer output is not changed only by rewriting the bit.
- <2> Set the pattern to be output to TSnIPC2 to TSnIPC0 and also set TSnSOC to 1 in order to provide software output.
- <3> Change the output pattern setting of TSnIPC2 to TSnIPC0 to change the timer output.  
The registers that can be changed during the software control are as follows.  
TSnCE bit of the TSnCTL0 register, TSnOPT1 to TSnOPT4 registers, TSnIDC, TSnTOS bits of TSnOPT7 register, TSnCCR0 to TSnCCR5 registers, TSnDTC0 register, TSnDTC1 register
- <4> Check that the reload request flag (TSnRSF) is set to 0. In the case of TSnRSF = 1, do not move to the next procedure until the setting of TSnRSF = 0.

- 
- <5> Software control release is started by setting TSnSOC = 0 (Not yet released in this step).
  - <6> Set the compare register which is necessary after the release of software output control. Move to the next procedure when any change is not needed. Change the register with the reload function in this step if necessary.
  - <7> Write to the TSnCCR1 register to activate reload.
  - <8> Execute reload to release the software output.

- 
- Caution**
1. Any change of TSnIPC2 to TSnIPC0 is prohibited after setting TSnSOC to 0.
  2. Be sure to execute reloading after the execution of procedures <4>, <5>, <6> and <7>. Software output cannot be released if reload cannot be executed.
-

### 17.10.10 120° excitation mode

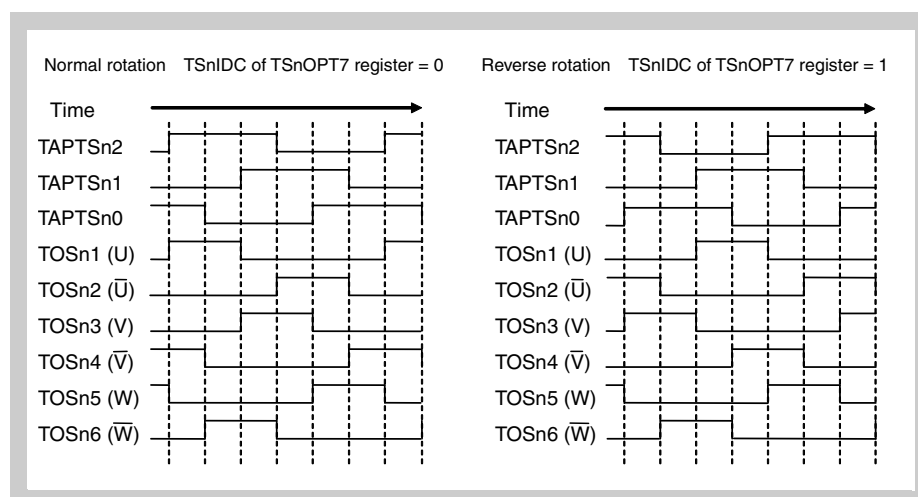
#### (1) Overview of 120° excitation mode

In the 120° excitation mode, 120° excitation control is performed by detecting the detection of external input (pattern: TAPTSn2 to TAPTSn0 pins, trigger: TSnSTCI0 signal) and by the output control of timer output (pins TOSn1 to TOSn6).

Motor systems that can be controlled in the 120° excitation mode are as follows.

- System where a hall sensor is connected at 120° intervals against the motor rotation
- System where normal/reverse rotation is possible by outputting, responding to the hall sensor input

A concrete output example of normal/reverse rotation is shown in *Figure 17-134 on page 852*.



**Figure 17-134** Normal/Reverse rotation example

- Note**
1. TSnOPT4 register: TSnSOC = 0,  
TSnOPT5 register: TSnADC, TSnPOT = 0
  2. The 120° excitation mode is valid when TSnMD3 to TSnMD0 of the TSnCTL1 register are set to [1, 0, 1, 0].



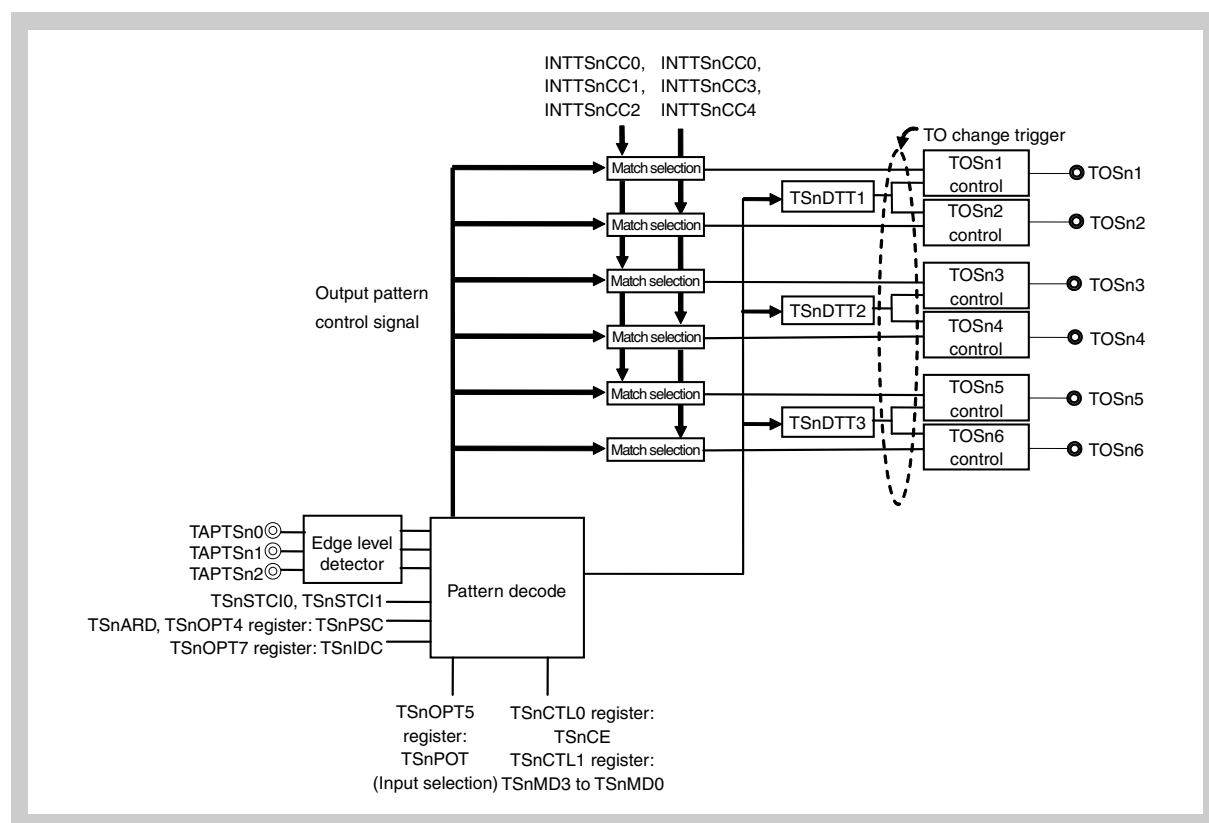


Figure 17-135 Configuration diagram in 120° excitation mode

Table 17-36 Output patten of TAPTSn2 to TAPTSn0 pins (normal rotation)

Output Pin	TAPTSn2 to TAPTSn0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	PWM1	PWM2	INACT	INACT	INACT	INACT	Note 1	Note 1
TOSn2	INACT	INACT	INACT	PWM3	PWM4	INACT	Note 1	Note 1
TOSn3	INACT	INACT	PWM1	PWM2	INACT	INACT	Note 1	Note 1
TOSn4	PWM4	INACT	INACT	INACT	INACT	PWM3	Note 1	Note 1
TOSn5	INACT	INACT	INACT	INACT	PWM1	PWM2	Note 1	Note 1
TOSn6	INACT	PWM3	PWM4	INACT	INACT	INACT	Note 1	Note 1

- Note**
1. The output level of the TOSn0 to TOSn6 pins remains the level before change when changed to [0, 0, 0] or [1, 1, 1].
  2. PWM1 to PWM4 indicate the PWM output of the TSnCCR1 to TSnCCR4 registers.
  3. NACT indicates the inactive level output.

There are the following two methods according to the external input used in the 120° excitation mode.

- Patten switch method: 3 patterns of input (hall sensor, etc.)
- Trigger input switch method: Trigger input (encoder circuit, offset, etc.)

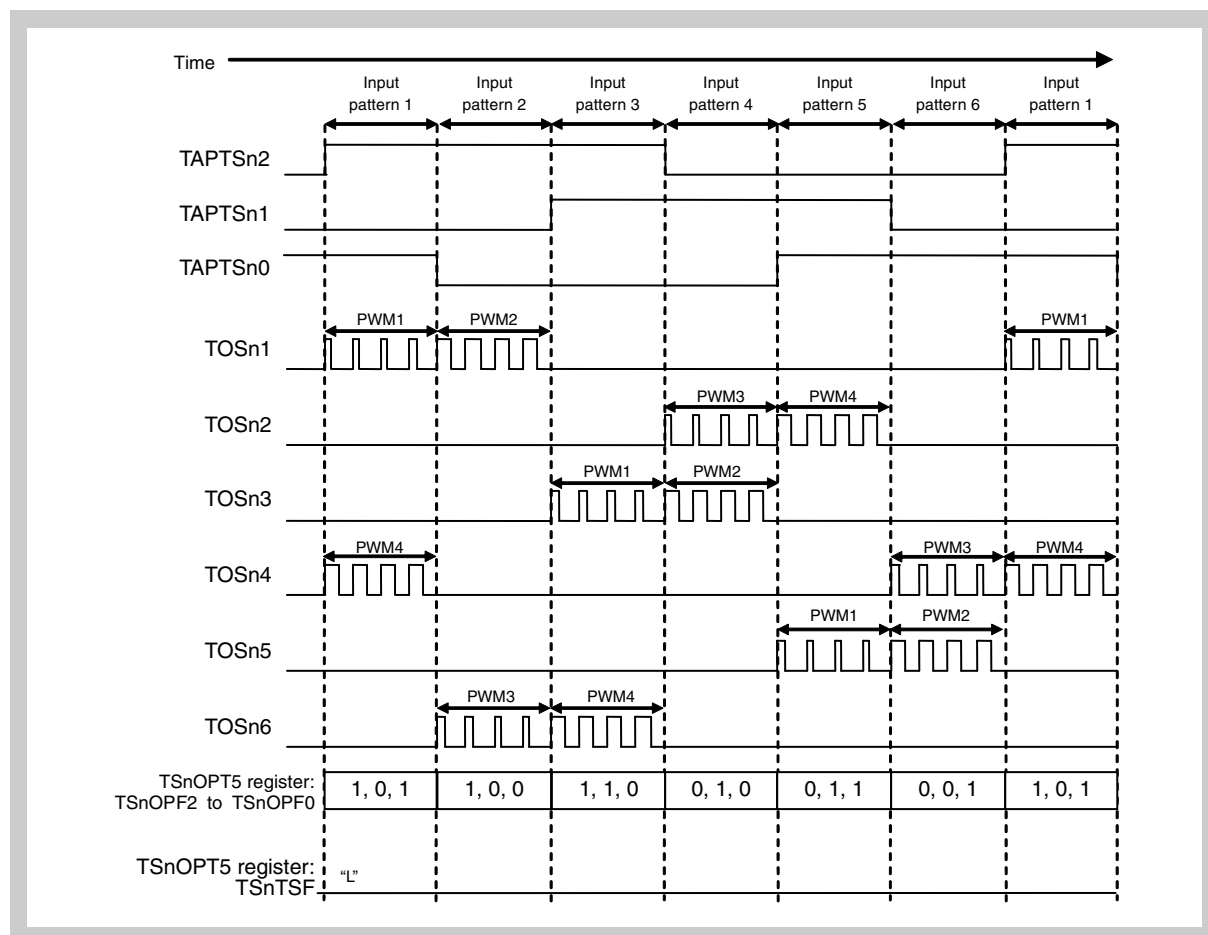
**(a) Pattern switch method (TSnPOT of TSnOPT5 register = 0)**

Level detection is performed for the TAPTSn2 to TAPTSn0 pins (three patterns of input from hall sensor) to decode the signals after the level detection. PWM output (which is to be determined with the values of the TSnCCR1 to TSnCCR4 registers) of the TOSn1 to TOSn6 pins is selected by judging the decode result. PWM output is generated by decoded signals and compare match interrupts (INTTSnCC1 to INTTSnCC4) to control the output. Dead time control is performed by the operation of the dead time counter at respective phase signal fall timing to ensure dead time insertion.

With the pattern switch method, the decoded pattern is output by using the data from input pattern (TAPTSn2 to TAPTSn0 pins), current direction control bit (TSnIDC of the TSnOPT7 register), TAPTSn2 to TAPTSn0 order detection flag (TSnTSF of the TSnOPT5 register). *Figure 17-136 on page 854 and Figure 17-137 on page 855* show the timer output when the TAPTSn2 to TAPTSn0 pins are changed. The output pattern is switched to the pattern corresponding to the input if any error occurs in the input pattern when switching from the input pattern 1 to the input pattern 4.

The pattern set with the input level of the TAPTSn2 to TAPTSn0 pins, the TSnIDC and TSnARD signals, (TSnPSS of TSnOPT5 register = 0), and the TSnPSC bit of the TSnOPT4 register (TSnPSS of TSnOPT5 register = 1) is output. Output pattern is determined with the TSnTSF flag instead of the TSnPSC bit if the value of the TSnTSF flag is determined.

TSnOPT4 register: TSnSOC = 0, TSnPSC = 0, TSnOPT5 register: TSnPOT = 0, and TSnOPT7 register: TSnIDC = 0.



**Figure 17-136 120° excitation mode operation example in use of pattern switch method (normal rotation)**

TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,

TSnOPT5 register: TSnPOT = 0, and TSnOPT7 register: TSnIDC = 1.

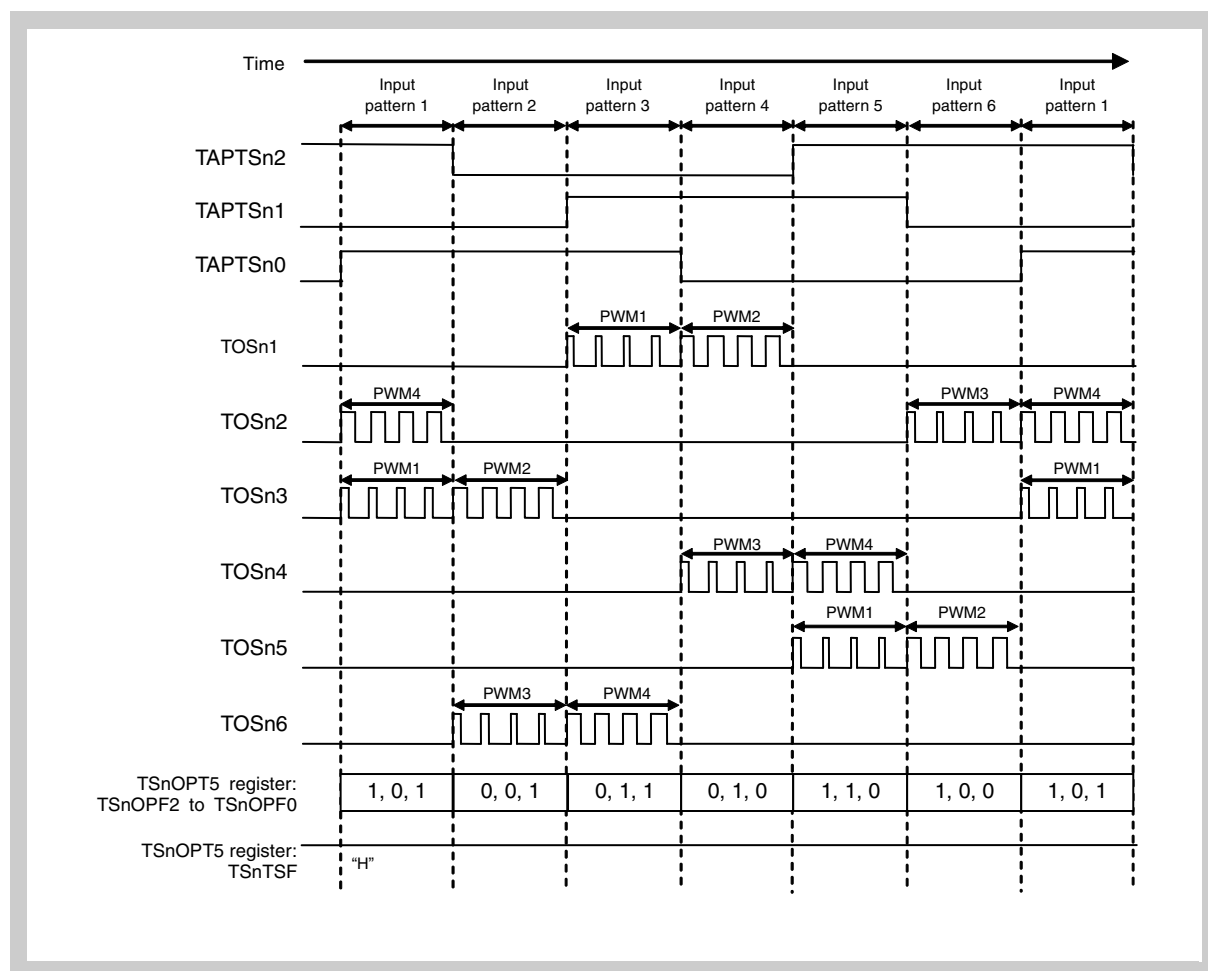


Figure 17-137 120° excitation mode operation example in use of pattern switch method (reverse rotation)

**(b) Trigger switch method**

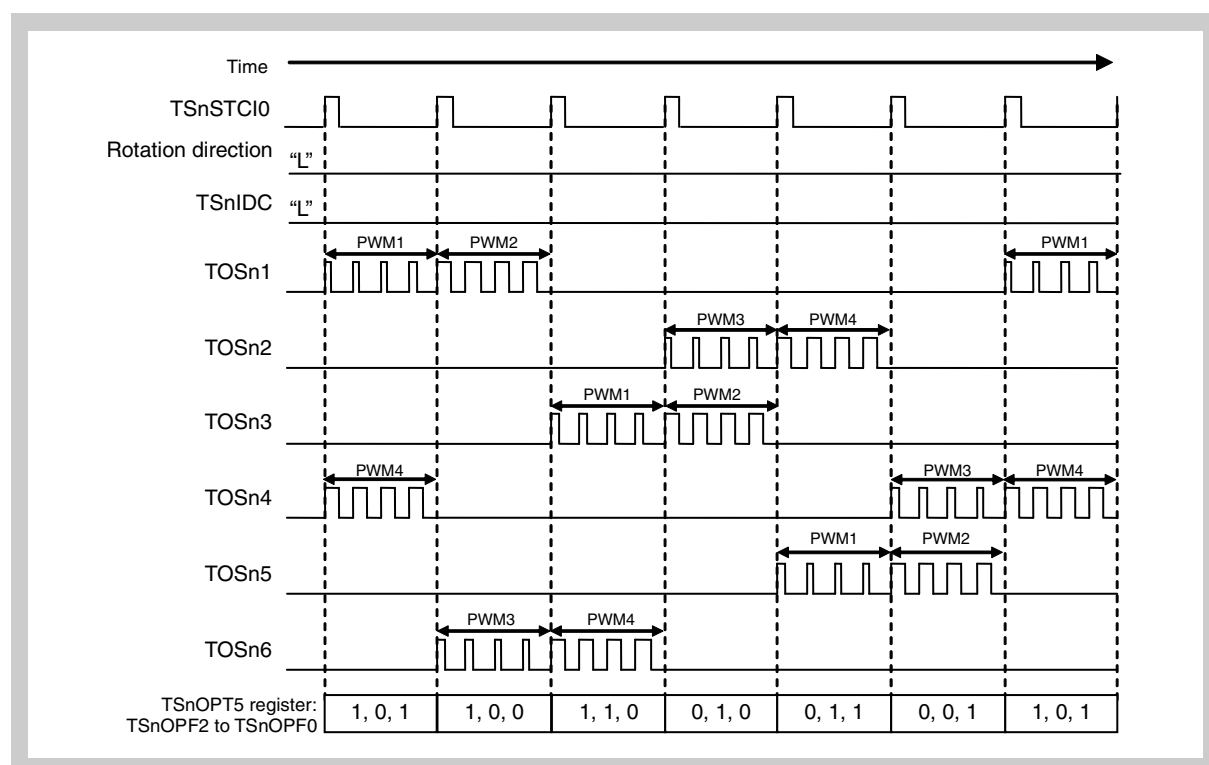
With the trigger input switch method, rises of the TSnSTCI0 and TSnSTCI1 signals are detected to generate the output switch timing.

The timer output initial pattern is set with the TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register. The output patterns after the initial pattern are determined with the TSnIPC2 to TSnIPC0 bits, rotation direction or the TSnIDC bit of the TSnOPT7 register. There are four patterns which can be output in combination of rotation direction and the TSnIDC bit.

TSnOPT5 register		Rotation direction
TSnPOT	TSnPSS	
1	0	Determine rotation direction by using input signal to TSnARD
	1	Determine rotation direction by using TSnPSC bit

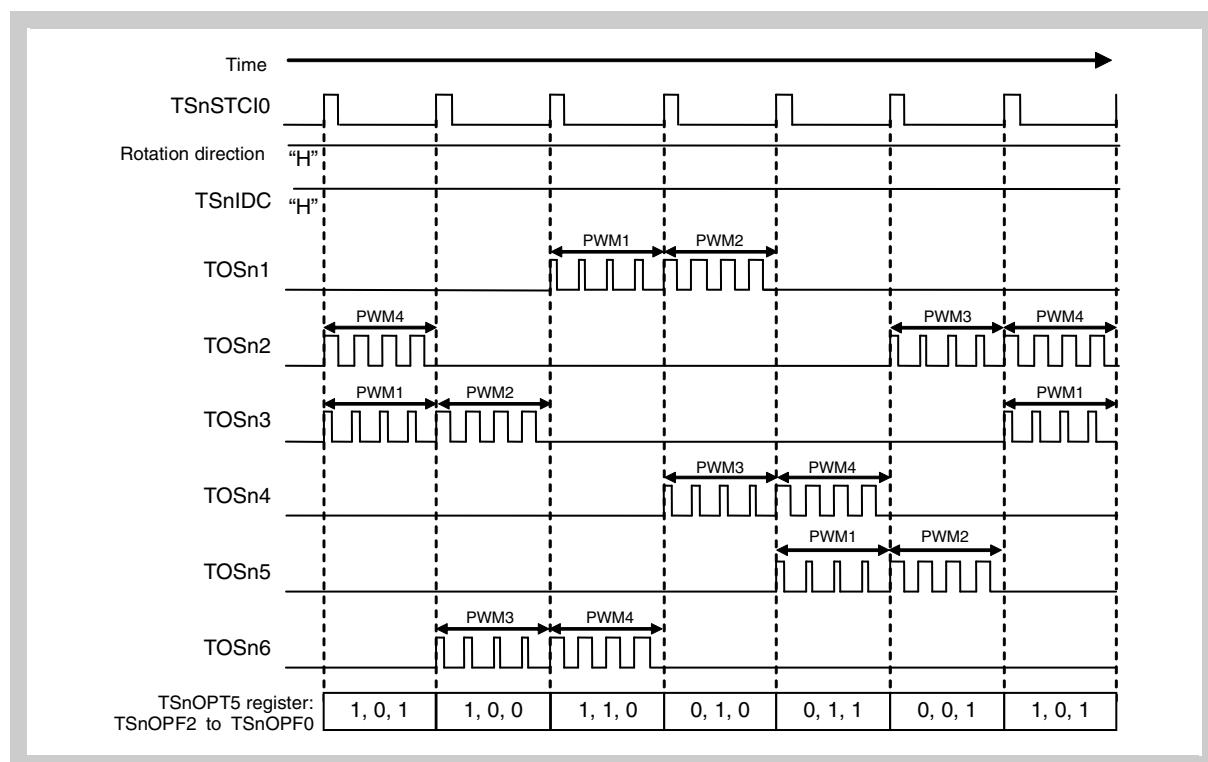
TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,

TSnOPT5 register: TSnPOT = 1, TSnPSS = 1, TSnOPT7 register: TSnIDC = 0

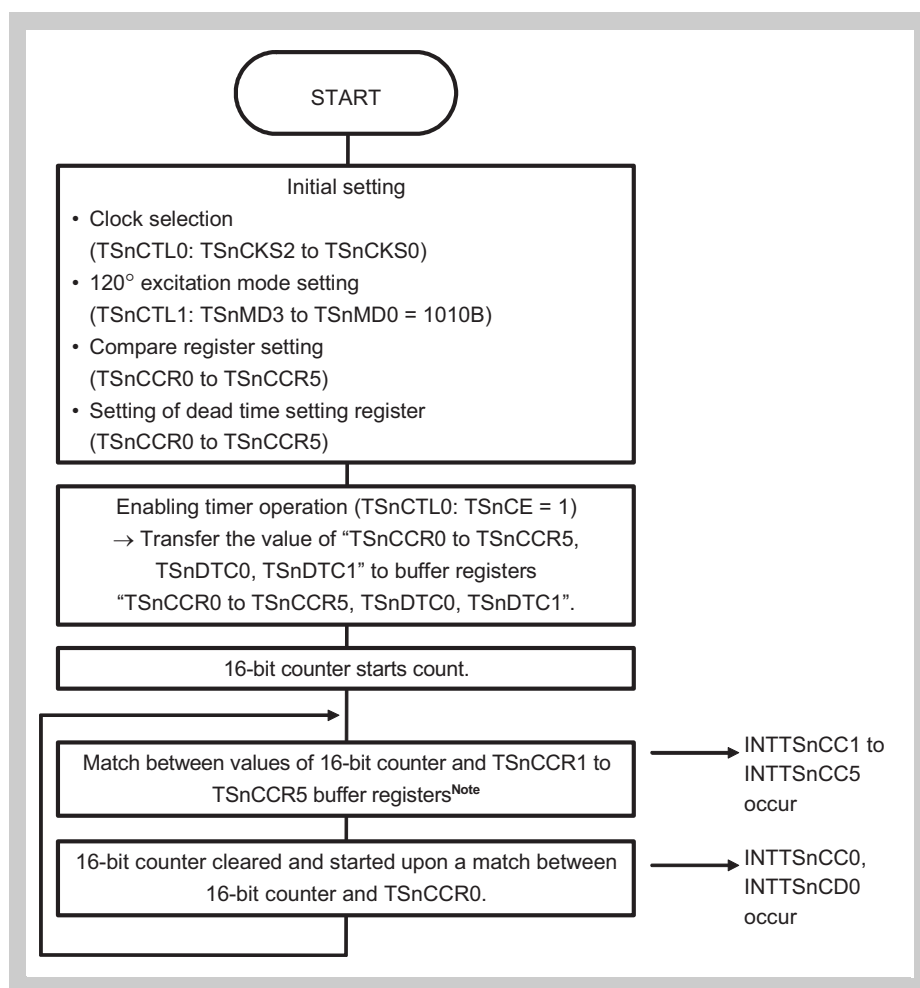


**Figure 17-138** 120° excitation mode operation example in use of trigger input switch method (normal rotation)

TSnOPT4 register: TSnSOC = 0, TSnPSC = 1, TSnOPT5 register: TSnPOT = 1, TSnPSS = 1, TSnOPT7 register: TSnIDC = 1



**Figure 17-139** 120° excitation mode operation example in use of trigger input switch method (reverse rotation)



**Figure 17-140 Basic operation flow in 120° excitation mode**

**Note** The 16-bit counter is not cleared upon a match between the values of the 16-bit counter and the TSnCCR1 to TSnCCR5 buffer registers.

**(2) 120° excitation mode operation list****(a) Register rewriting**

Register	Rewriting method	Rewriting During Operation	Function
TSnCCR0	Reload	Possible	Cycle
TSnCCR1 to TSnCCR3	Reload	Possible	PWM duty
TSnCCR4	Reload	Possible	PWM duty (selectable as A/D conversion trigger)
TSnCCR5	Reload	Possible	Selectable as A/D conversion trigger
TSnDTC0, TSnDTC1	Reload	Conditionally possible <sup>Note</sup>	Dead time

**Note** Refer to “Rewriting of TSnDTC0 and TSnDTC1 registers” on page 827 for details.

**(b) Input pin**

Pin	Function
TTRGSn	-
TEVTSn	-
TAPTSn2 to TAPTSn0	Pattern input
TSnSTCI0, TSnSTCI1 signals	Trigger input
TSnARD signal	Pattern change order input

**(c) Output pin**

Pin	Function
TOSn0	Toggle output by TSnCCR0 compare match
TOSn1	PWM output (with dead time) by TSnCCR1, TSnCCR2 compare match
TOSn2	PWM output (with dead time) by TSnCCR3, TSnCCR4 compare match
TOSn3	PWM output (with dead time) by TSnCCR1, TSnCCR2 compare match
TOSn4	PWM output (with dead time) by TSnCCR3, TSnCCR4 compare match
TOSn5	PWM output (with dead time) by TSnCCR1, TSnCCR2 compare match
TOSn6	PWM output (with dead time) by TSnCCR3, TSnCCR4 compare match
TOSn7	Pulse output by A/D conversion trigger

**(d) Interrupt**

Interrupt	Function
INTTSnCC0 to INTTSnCC5	Compare match of INTTSnCC0 to INTTSnCC5 registers
INTTSnOV	-
INTTSnER	Error
INTTSnOD	-
INTTSnCD0	Peak interrupt (occurs at the same timing with INTTSnCC0).
INTTSnWN	Warning

**(e) Compare match timing**

Compare match	Timing
TSnCCR0	When 16-bit counter switches from TSnCCR0 to 0000H
TSnCCR1 to TSnCCR5	After detection of match between values of 16-bit counter and TSnCCR1 to TSnCCR5

**Note** “-” indicates an unused function in the 120° excitation mode.



**(3) Setting of 120° excitation mode****(a) Mode setting**

The 120° excitation mode is set by setting TSnMD3 to TSnMD0 of the TSnCTL1 register to [1, 0, 1, 0].

**(b) Output level/output enable setting**

Output level/output enable is set by setting the TSnOL0 to TSnOL7 and TSnOE0 to TSnOE7 bits of the TSnIOC0 register.

Toggle output is provided from the TOSn0 pin with cycle match (match between the 16-bit counter and TSnCCR0 register).

The TOSn7 pin is for A/D conversion output. Enable the output according to the need.

**(c) Error interrupt/warning interrupt occurrence enabling**

Error interrupt (INTTSnER) occurrence is enabled by setting the TSnEOC bit of the TSnIOC4 register to 1 when the positive/negative phase simultaneous active state is detected.

A warning interrupt (INTTSnWN) is enabled by setting TSnWOC of the TSnIOC4 register to 1.

**(d) Interrupt and thinning out function setting**

A peak interrupt (INTTSnCD0) occurs upon a match between the TSnCCR0 register and 16-bit counter. (A trough interrupt does not occur).

Set TSnICE of the TSnOPT1 register to 1 to output a peak interrupt.

To use the thinning out function for peak interrupt, set the TSnID4 to TSnID0 bits of the TSnOPT1 register.

**(e) Reload thinning out function setting**

Set TSnRDE of the TSnOPT1 register to 1 to set reload timing to the same timing with the interrupt timing.

**(f) A/D conversion trigger output setting**

To set A/D conversion trigger 0 (TSnADTRG0 signal), set whether to enable/disable at the match timing with the TSnCCR5 register, at the match timing with the TSnCCR4 register, and at the peak interrupt (INTTSnCD0) by using the TSnAT04, TSnAT02, and TSnAT01 bits of the TSnOPT2 register.

To set A/D conversion trigger 1 (TSnADTRG1 signal), set whether to enable/disable at match timing with TSnCCR5 register, at match timing with the TSnCCR4 register, and at peak interrupt (INTTSnCD0) by using the TSnAT14, TSnAT12, and TSnAT11 bits of the TSnOPT3 register.

No thinning out, 1-thinning out, 3-thinning out, or 7-thinning out can be set with the TSnACC01 and TSnACC00 bits of the TSnOPT2 register, and the TSnACC11 and TSnACC10 bits of the TSnOPT3 register.

- Caution**
1. When using the TOSn7 pin, set the TSnOPT2 and TSnOPT3 registers and TSnCCR4 and TSnCCR5 registers appropriately according to the system.
  2. Trough interrupts (INTTSnOD) do not occur in the 120° excitation mode. Be sure to set TSnAT00, 10 bits of the TSnOPT2 and TSnOPT3 registers to 0.
  3. The 16-bit sub-counter does not operate in the 120° excitation mode. Be sure to set the TSnAT07, TSnAT06, TSnAT17, and TSnAT16 bits of the TSnOPT2, TSnOPT3 registers to 0.
  4. The 16-bit counter does not enter the count-down status in the 120° excitation mode. Be sure to set the TSnAT05, TSnAT03, TSnAT15, and TSnAT13 bits of the TSnOPT2 and TSnOPT3 registers to 0.
  5. Both PWM duty and A/D0 conversion trigger are set with the TSnCCR4 register in the 120° excitation mode. Note that both the PWM duty and A/D0 conversion trigger timing will be changed if the TSnCCR4 register setting value is changed.

#### (g) Dead time setting

Dead time is set with the TSnDTC0 and TSnDTC1 registers.

Dead time can be calculated with the following equation.

16-bit counter operation clock cycle × TSnDTC0

16-bit counter operation clock cycle × TSnDTC1

The time for the change from the inactive state of TOSn2, TOSn4, TOSn6 pins to the active state of TOSn1, TOSn3, TOSn5 pins can be set with the TSnDTC0 register.

The time for the change from the inactive state of TOSn1, TOSn3, TOSn5 pins to the active state of TOSn2, TOSn4, TOSn6 pins can be set with the TSnDTC1 register.

#### (h) Output PWM setting

In the 120° control, the output of the TOSn1, TOSn3, and TOSn5 pins is controlled with the TSnCCR1 and TSnCCR2 registers, and the output the TOSn2, TOSn4, and TOSn6 pins is controlled with the TSnCCR3 and TSnCCR4 registers. The duty ratio can be set with the TSnCCR1 to TSnCCR4 registers for the PWM cycle (TSnCCR0 register). 0000H must be set to the TSnCCR1 to TSnCCR4 registers to set the duty ratio of 0%, and “TSnCCR0 + 1” should be set to the TSnCCR1 to TSnCCR4 registers to set the duty ratio of 100%. These settings enable chopping output control and rectangular output control.

#### (i) Pattern output switch timing, pattern output order, and initial output pattern setting

[Pattern switch method]

The pattern switch method is enabled by setting TSnPOT of the TSnOPT5 register to 0.

Output of the TOSn1 to TOSn6 pins is switched when the TAPTSn2 to TAPTSn0 pins change.

Output order at operation start is set with the TSnIDC bit of the TSnOPT7 register. Initial output pattern is set with the TSnPSC bit of the TSnOPT4 register. However, setting of the TSnPSC bit is invalid after rotation direction is determined (after a value is set in the TSnTSF bit of the TSnOPT5 register).

[Trigger switch method]

Trigger switch method is enabled by setting TSnPOT of the TSnOPT5 register to 1. Output of the TOSn1 to TOSn6 pins is switched with the rise of the external input (TSnSTCI1 and TSnSTCI0 signals).

Output order is set with the combination of TSnARD signal, TSnPSC bit of the TSnOPT4 register, or the TSnIDC bit of the TSnOPT7 register. Selection of the TSnARD signal or the TSnPSC bit of the TSnOPT4 register is set with the TSnPSS bit of the TSnOPT4 register. The output order is controlled with the TSnARD input signal when TSnPSS is set to 0, and controlled with the TSnPSC bit when TSnPSS is set to 1. Refer to *“Timer output pattern in each mode and function” on page 671* for the pattern output order.

The initial output pattern can be controlled with the TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register. The Initial pattern is output at the timer Sn operation start setting timing (when TSnCE of the TSnCTL0 register to 1) following setting with the TSnIPC2 to TSnIPC0 bit. Refer to *“Timer output initial pattern setting in each mode and function” on page 663* for details.

---

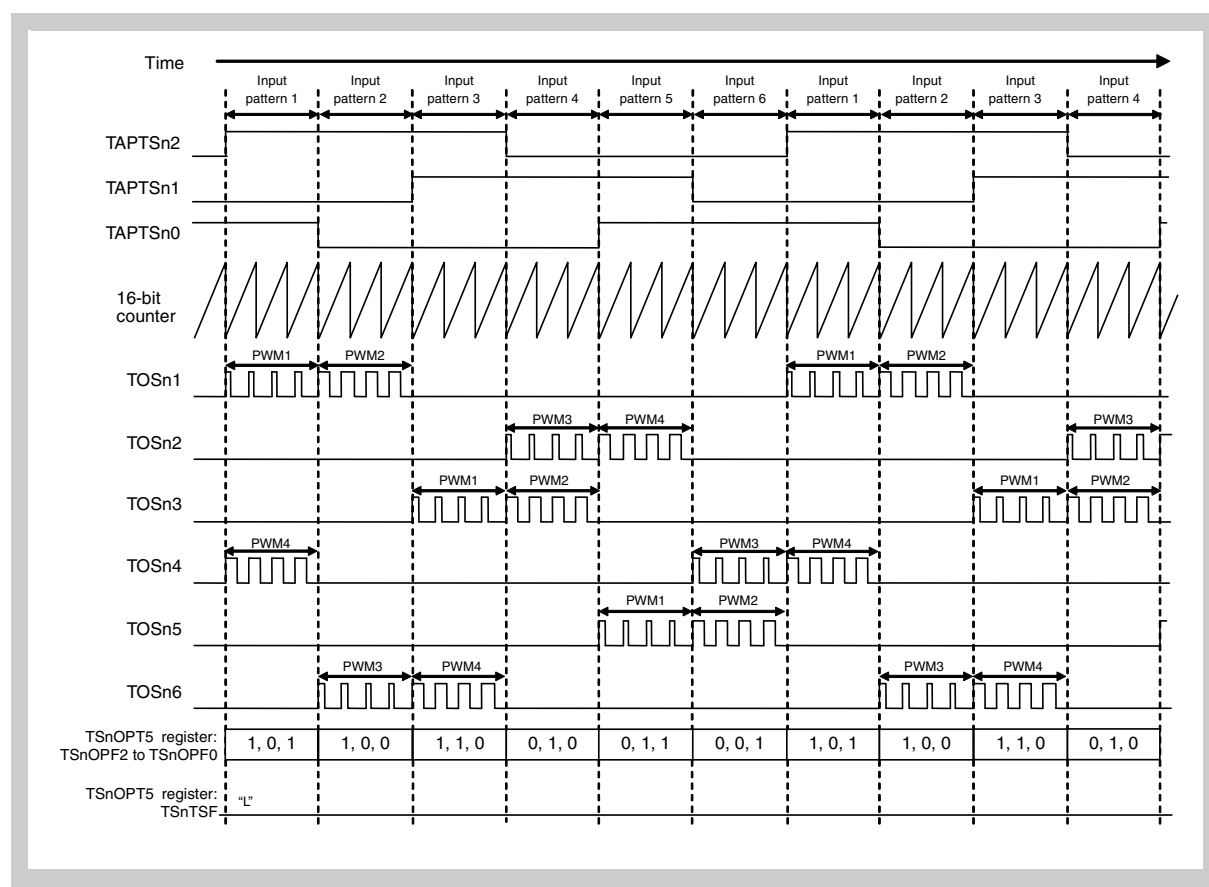
**Caution** Set the initial pattern by reading the input level of the port connected to the TAPTSn2 to TAPTSn0 pins.

---

**(4) Operation in 120° excitation mode**

Figure 17-141 on page 864 and Figure 17-142 on page 865 show operation examples in the 120° excitation mode. TOSn1 to TOSn6 detect input level change timing of the TAPTSn2 to TAPTSn0 pins to change output pattern. The 16-bit counter operation is saw-tooth wave operation to provide PWM output by the SnCCR0 to TSnCCR4 registers. The 16-bit counter is not cleared to 0000H until the match with the TSnCCR0 register value in the PWM cycle even if change of the TAPTSn2 to TAPTSn0 pins is detected. The timer output pattern is switched from the cycle following the PWM cycle where the change of the TAPTSn2 to TAPTSn0 pins is detected.

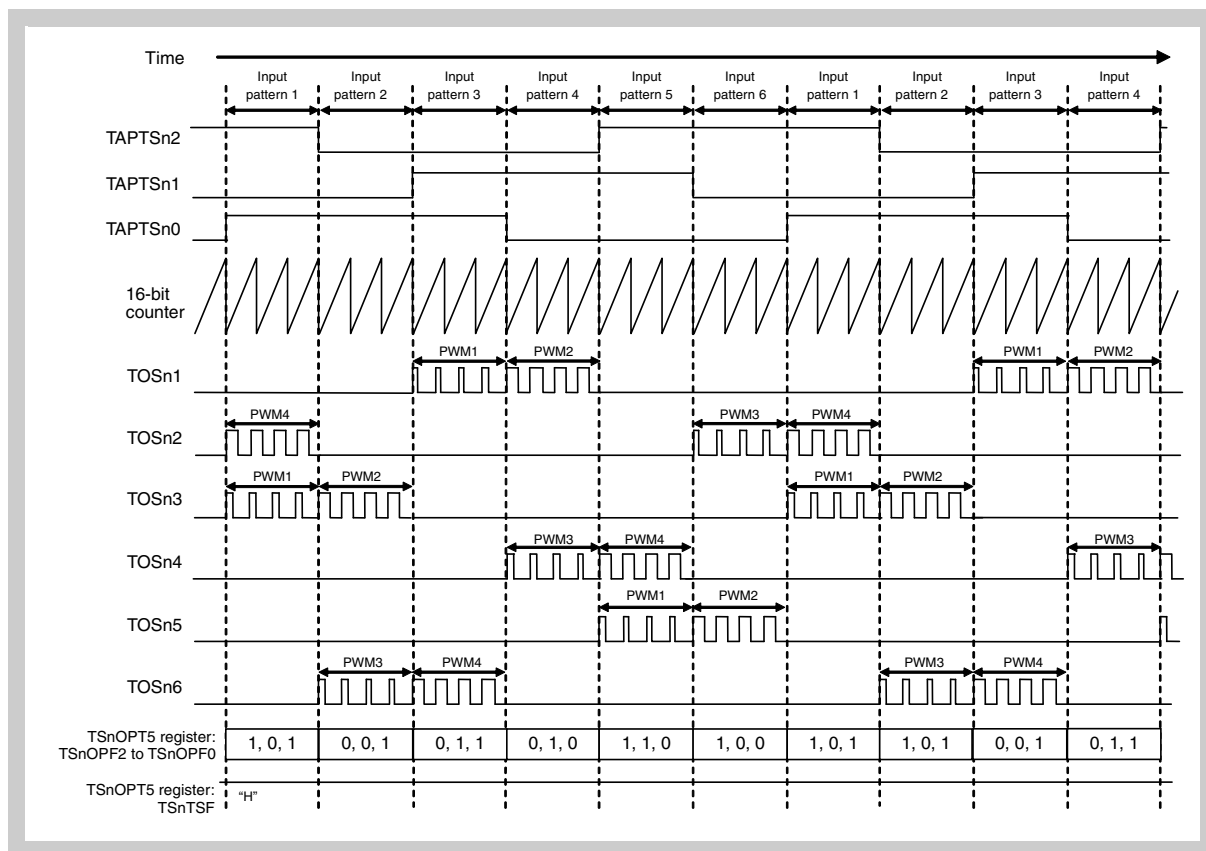
TSnOPT4 register: TSnSOC = 0, TSnPSC = 0, TSnOPT5 register: TSnADC = 0, TSnPOT = 0, TSnOPT7 register: TSnIDC = 0



**Figure 17-141** 120° excitation mode basic operation timing example (pattern switch method, normal rotation)

**Note** PWM1 to PWM4 indicate the PWM operation set with the TSnCCR1 to TSnCCR4 registers.

TSnOPT4 register: TSnSOC = 0, TSnPSC = 1, TSnOPT5 register: TSnADC = 0, TSnPOT = 0, TSnOPT7 register: TSnIDC = 1



**Figure 17-142** 120° excitation mode basic operation timing example (pattern switch method, reverse rotation)

**Note** PWM1 to PWM4 indicate the PWM operation set with the TSnCCR1 to TSnCCR4 registers.

**(5) Output pattern list in 120° excitation mode**

In the 120° excitation mode, the output pattern is determined by the rotation direction and the TSnIDC bit of the TSnOPT7 register.

TSnOPT5 register		Rotation direction
TSnPOT	TSnPSS	
0	-	TSnTSF flag
1	0	Input signal to TSnARD
1	1	TSnPSC bit

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,  
TSnOPT5 register: TSnADC = 0, TSnPOT = 1, TSnPSS = 1,  
and TSnOPT7 register: TSnIDC = 0

→ Pattern switch order

Output pin	TSnIPC2 to TSnIPC0 of TSnOPT4 register <sup>Note 1</sup> /TSnOPF2 to TSnOPF0 of TSnOPT5 register							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	PWM1	PWM2	INACT	INACT	INACT	INACT	Note 2	Note 3
TOSn2	INACT	INACT	INACT	PWM3	PWM4	INACT	Note 2	Note 3
TOSn3	INACT	INACT	PWM1	PWM2	INACT	INACT	Note 2	Note 3
TOSn4	PWM4	INACT	INACT	INACT	INACT	PWM3	Note 2	Note 3
TOSn5	INACT	INACT	INACT	INACT	PWM1	PWM2	Note 2	Note 3
TOSn6	INACT	PWM3	PWM4	INACT	INACT	INACT	Note 2	Note 3

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,  
TSnOPT5 register: TSnADC = 0, TSnPOT = 1, TSnPSS = 1,  
and TSnOPT7 register: TSnIDC = 1

→ Pattern switch order

Output pin	TSnIPC2 to TSnIPC0 of TSnOPT4 register <sup>Note 1</sup> /TSnOPF2 to TSnOPF0 of TSnOPT5 mode							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	INACT	PWM1	PWM2	INACT	Note 2	Note 3
TOSn2	PWM3	PWM4	INACT	INACT	INACT	INACT	Note 2	Note 3
TOSn3	PWM2	INACT	INACT	INACT	INACT	PWM1	Note 2	Note 3
TOSn4	INACT	INACT	PWM3	PWM4	INACT	INACT	Note 2	Note 3
TOSn5	INACT	PWM1	PWM2	INACT	INACT	INACT	Note 2	Note 3
TOSn6	INACT	INACT	INACT	INACT	PWM3	PWM4	Note 2	Note 3

- Note**
1. The output pattern of TSnIPC2 to TSnIPC0 is changed by writing when TSnPOT = 1. Then, output is switched according to the pattern switch order when the pattern switch trigger is generated by the rise of TSnSTCI0 and TSnSTCI1. In this case, TSnIPC2 to TSnIPC0 do not change even if the output pattern is switched.
  2. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], the output remains the level before the change.
  3. Output remains the level before the change.
  4. PWM1 to PWM4: PWM output of the TSnCCR1 to TSnCCR4 registers

## 5. INACT: Inactive level output

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,  
 TSnOPT5 register: TSnADC = 0, TSnPOT = 1, TSnPSS = 1,  
 and TSnOPT7 register: TSnIDC = 0

← Pattern switch order

Output Pin	TSnIPC2 to TSnIPC0 of TSnOPT4 Register <sup>Note 1</sup> /TSnOPF2 to TSnOPF0 of TSnOPT5 Register							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	PWM2	PWM1	INACT	INACT	INACT	INACT	Note 2	Note 3
TOSn2	INACT	INACT	INACT	PWM4	PWM3	INACT	Note 2	Note 3
TOSn3	INACT	INACT	PWM2	PWM1	INACT	INACT	Note 2	Note 3
TOSn4	PWM3	INACT	INACT	INACT	INACT	PWM4	Note 2	Note 3
TOSn5	INACT	INACT	INACT	INACT	PWM2	PWM1	Note 2	Note 3
TOSn6	INACT	PWM4	PWM3	INACT	INACT	INACT	Note 2	Note 3

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,  
 TSnOPT5 register: TSnADC = 0, TSnPOT = 1, TSnPSS = 1,  
 and TSnOPT7 register: TSnIDC = 1

← Pattern switch order

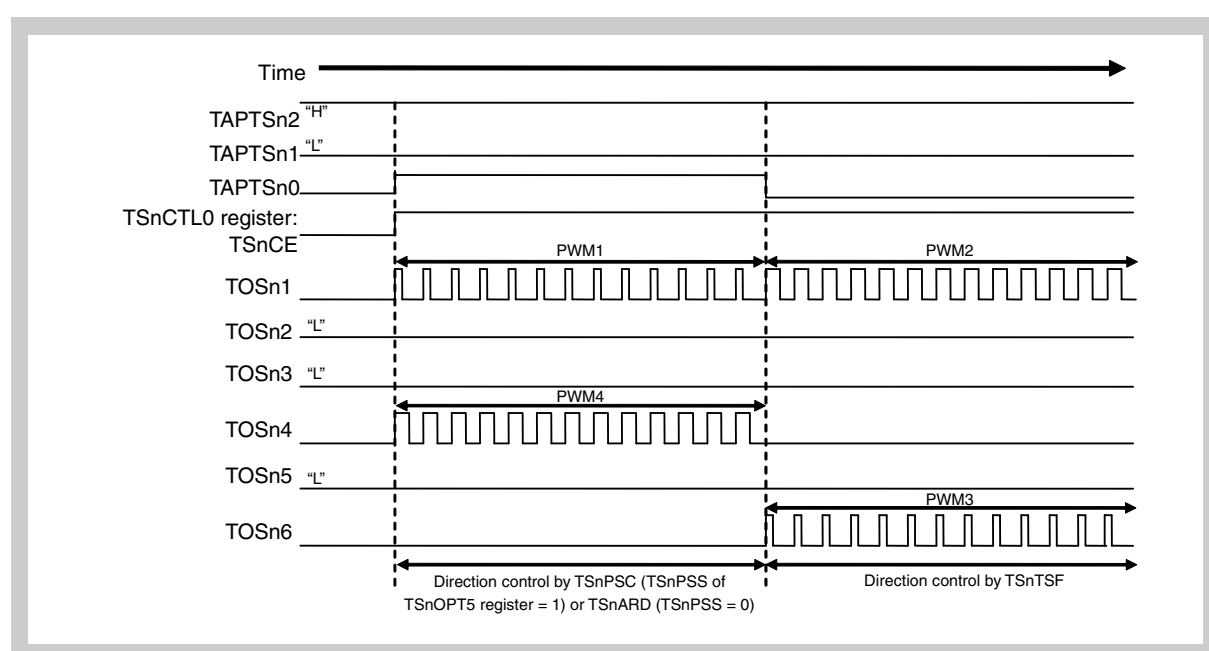
Output Pin	TSnIPC2 to TSnIPC0 of TSnOPT4 Register <sup>Note 1</sup> /TSnOPF2 to TSnOPF0 of TSnOPT5 Register							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	INACT	PWM2	PWM1	INACT	Note 2	Note 3
TOSn2	PWM4	PWM3	INACT	INACT	INACT	INACT	Note 2	Note 3
TOSn3	PWM1	INACT	INACT	INACT	INACT	PWM2	Note 2	Note 3
TOSn4	INACT	INACT	PWM4	PWM3	INACT	INACT	Note 2	Note 3
TOSn5	INACT	PWM2	PWM1	INACT	INACT	INACT	Note 2	Note 3
TOSn6	INACT	INACT	INACT	INACT	PWM4	PWM3	Note 2	Note 3

- Note**
1. The output pattern of TSnIPC2 to TSnIPC0 is changed by writing when TSnPOT = 1. Then, output is switched according to the pattern switch order when the pattern switch trigger is generated by the rise of TSnSTC10 and TSnSTC11. In this case, TSnIPC2 to TSnIPC0 do not change even if the output pattern is switched.
  2. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], the output remains the level before the change.
  3. Output remains the level before the change.
  4. PWM1 to PWM4: PWM output of the TSnCCR1 to TSnCCR4 registers
  5. INACT: Inactive level output

**(6) Operation start timing in 120° excitation mode**

In the case of the trigger switch control in the 120° excitation mode, the pattern that is set with the TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register, rotation direction, and the TSnIDC bit of the TSnOPT7 register can be output. In the case of the pattern switch control (TSnPOT = 0), however, the pattern of the TAPTSn2 to TAPTSn0 pins can be detected, but the rotation direction (TSnTSF bit of the TSnOPT5 register) cannot be determined. Therefore, the initial pattern is determined with TSnPSC of the TSnOPT4 register (TSnPSS = 1 of the TSnOPT5 register) or the TSnARD signal level (TSnPSS = 0 of the TSnOPT5 register) until the rotation direction is determined.

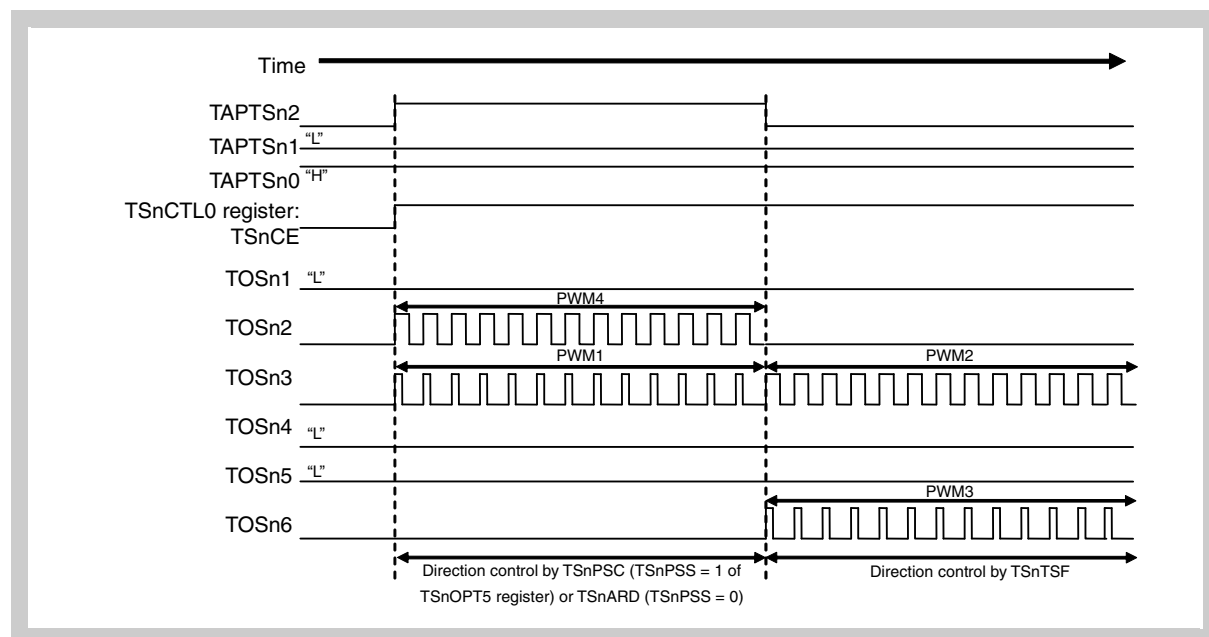
TSnOPT4 register: TSnSOC = 0, TSnPSC = 0 (TSnARD = 0),  
 TSnOPT5 register: TSnPOT = 0,  
 and TSnOPT7 register: TSnIDC = 0.



**Figure 17-143** Control at timer output start during normal rotation (when correct pattern is input)

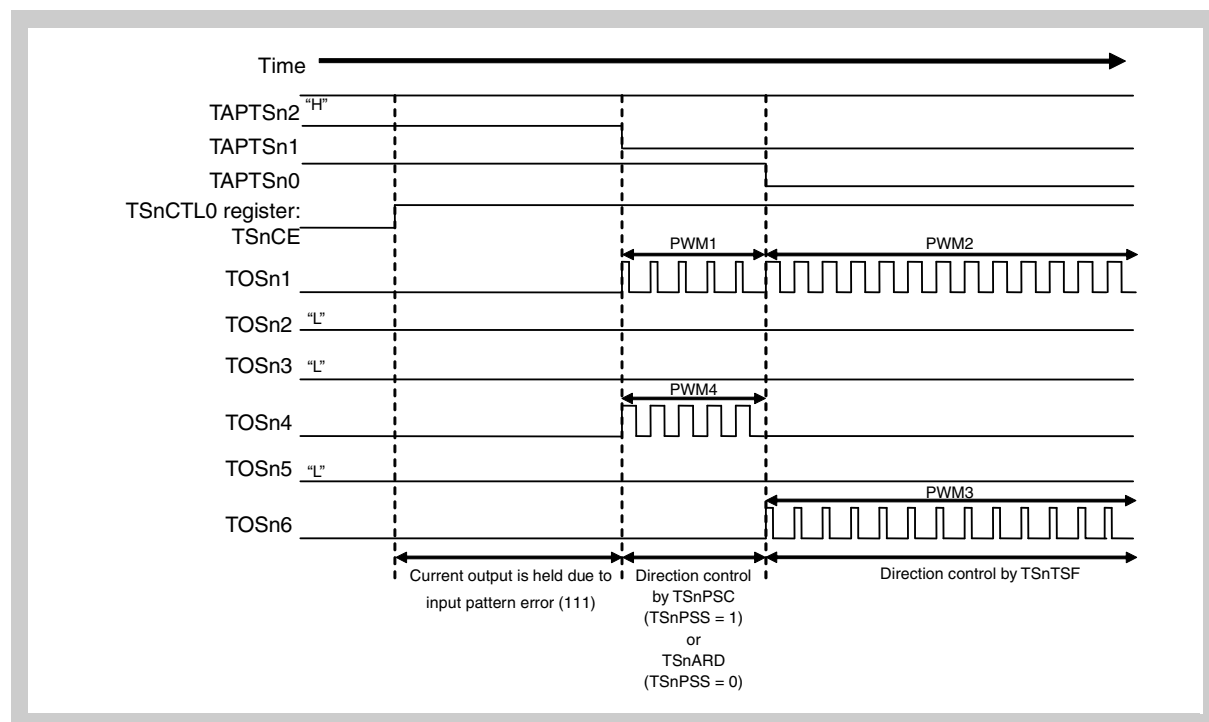


TSnOPT4 register: TSnSOC = 0, TSnPSC = 1 (TSnARD = 1),  
 TSnOPT5 register: TSnPOT = 0,  
 and TSnOPT7 register: TSnIDC = 0.



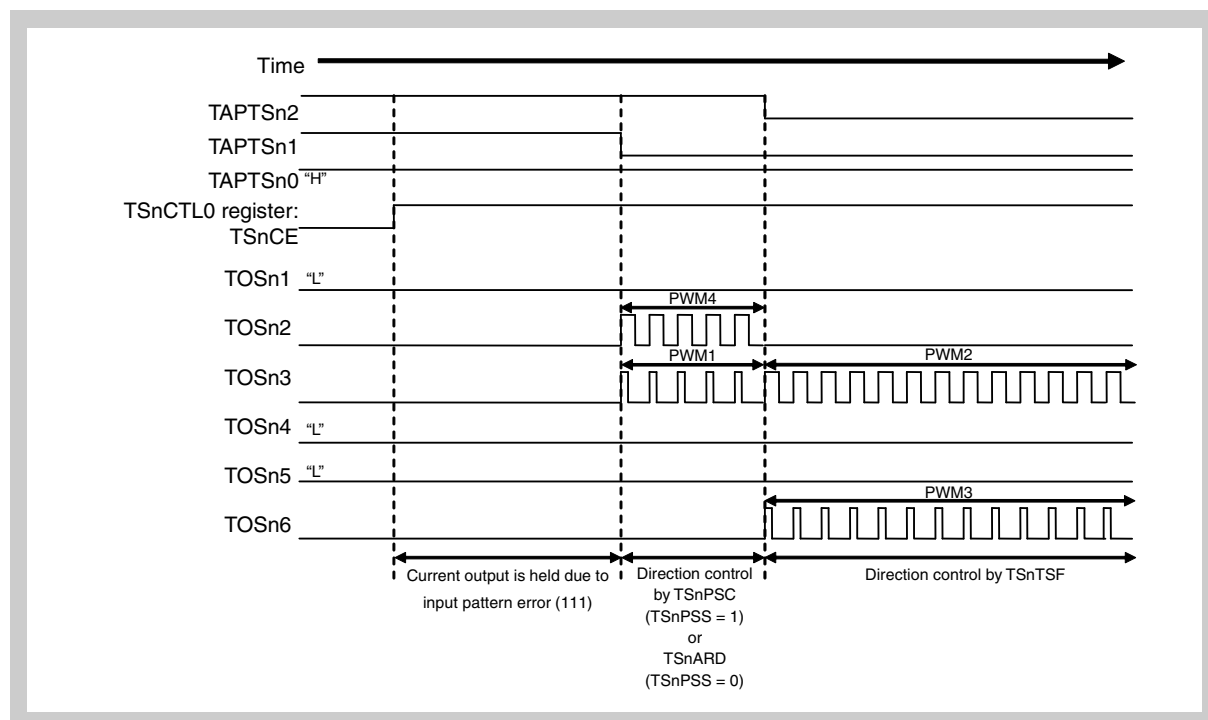
**Figure 17-144** Control at timer output start during reverse rotation (when correct pattern is input)

TSnOPT4 register: TSnSOC = 0, TSnPSC = 0 (TSnARD = 0),  
 TSnOPT5 register: TSnPOT = 0,  
 and TSnOPT7 register: TSnIDC = 0.



**Figure 17-145** Control at timer output start during normal rotation (when error pattern is input)

TSnOPT4 register: TSnSOC = 0, TSnPSC = 1 (TSnARD = 1),  
 TSnOPT5 register: TSnPOT = 0,  
 and TSnOPT7 register: TSnIDC = 0.



**Figure 17-146** Control at timer output start during reverse rotation (when error pattern is input)

### (7) Output switch timing in 120° excitation mode

In the 120° excitation mode, the external output pattern switch timing (TSnSTCI0 and TSnSTCI1 signals, TAPTSn2 to TAPTSn0 pins) is input regardless of the 16-bit counter operation. The output switch timing is from the cycle following the cycle of external input switch timing. Therefore, PWM output is held until the current cycle is completed even if the switch timing is input from the outside.

When the TAPTSn2 to TAPTSn0 pins are changed serially within the 1st cycle in the pattern switch method, the value edge-detected just before cycle match is used and reflected to PWM output from the following cycle. If the TSnSTCI0 and TSnSTCI1 signal triggers are input several times within a cycle in the trigger switch method, the pattern is switched to the adjacent output pattern due to only once acceptance.

Rewriting of TSnIPC2 to TSnIPC0 is prior to the TSnSTCI0 and TSnSTCI1 triggers in case of both occurrences within a cycle in the trigger switch method. The value rewritten just before cycle match is reflected if TSnIPC2 to TSnIPC0 are rewritten several times in a cycle.

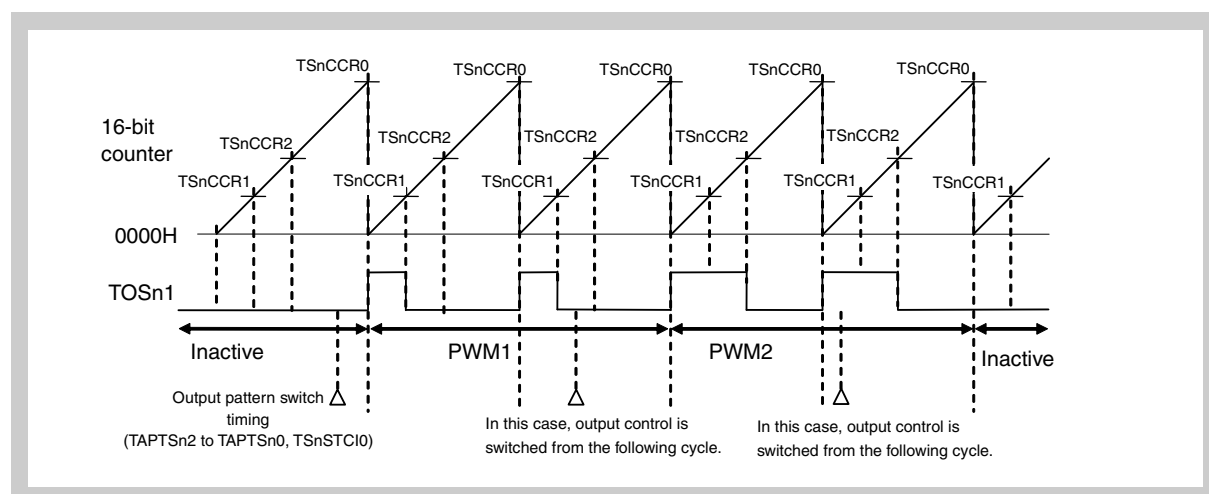


Figure 17-147 Output switch example (TAPTSn2 to TAPTSn0 Pins, TSnSTCI0, TSnSTCI1 signal trigger input)

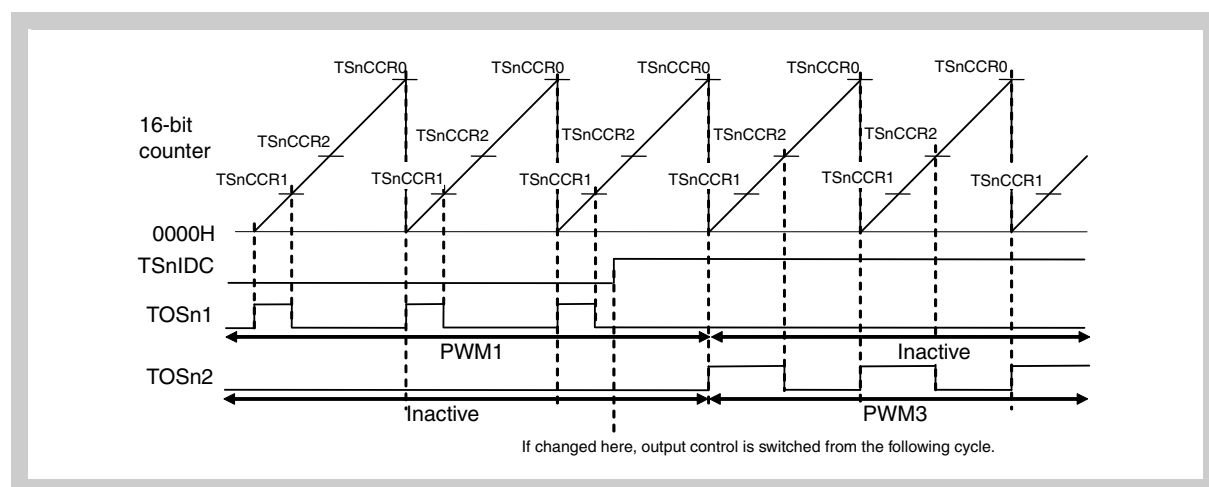
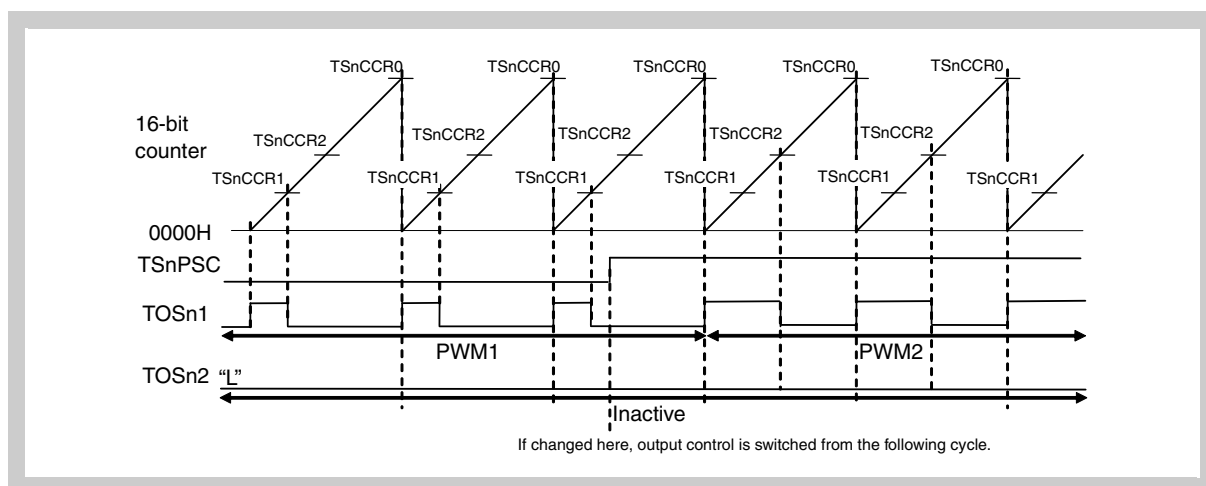


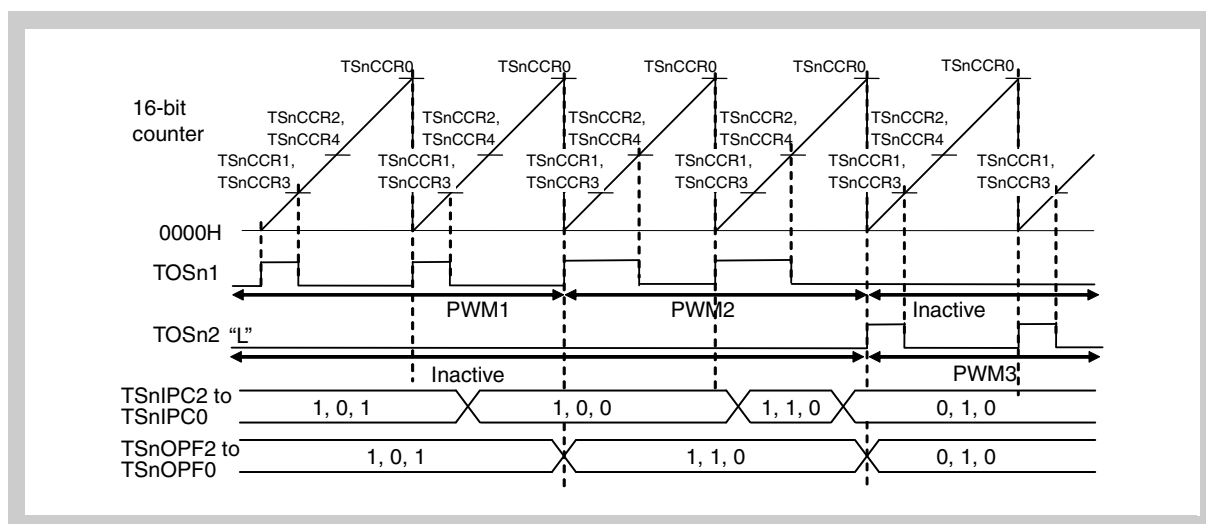
Figure 17-148 Output switch example (switch with TSnIDC Bit of TSnOPT7 register)

(TSnOPT5 register: TSnPOT bit = 1)



**Figure 17-149** Output switch example (switch with TSnPSC Bit of TSnOPT4 register)

TSnOPT4 register: TSnSOC = 0, TSnOPT5 register: TSnPOT = 1

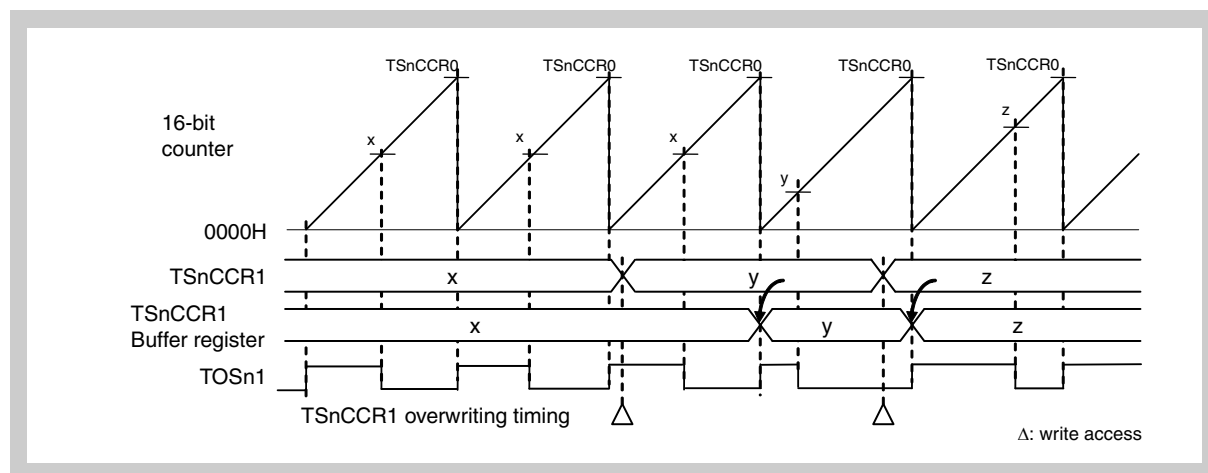


**Figure 17-150** Output switch example (switch with TSnIPC2 to TSnIPC0 bits of TSnOPT4 register)

**(8) Compare register rewrite timing in 120° excitation mode**

The operation example when the TSnCCR1 register is reloaded (rewritten) is described below.

Figure 17-151 on page 873 shows an output example in case of rewriting the TSnCCR1 register. After change of the TSnCCR1 register, data is not transferred to the TSnCCR1 buffer register until the next reload timing (the changed data will not be valid), so that the output waveform can be obtained as set. However, rewriting the TSnCCR1 register again is prohibited during the reload is kept pending (in the period from the change of the TSnCCR1 register to the reload overwriting). Be sure to read the reload request flag (TSnRSF) to confirm 0 status before writing to the TSnCCR1 register.



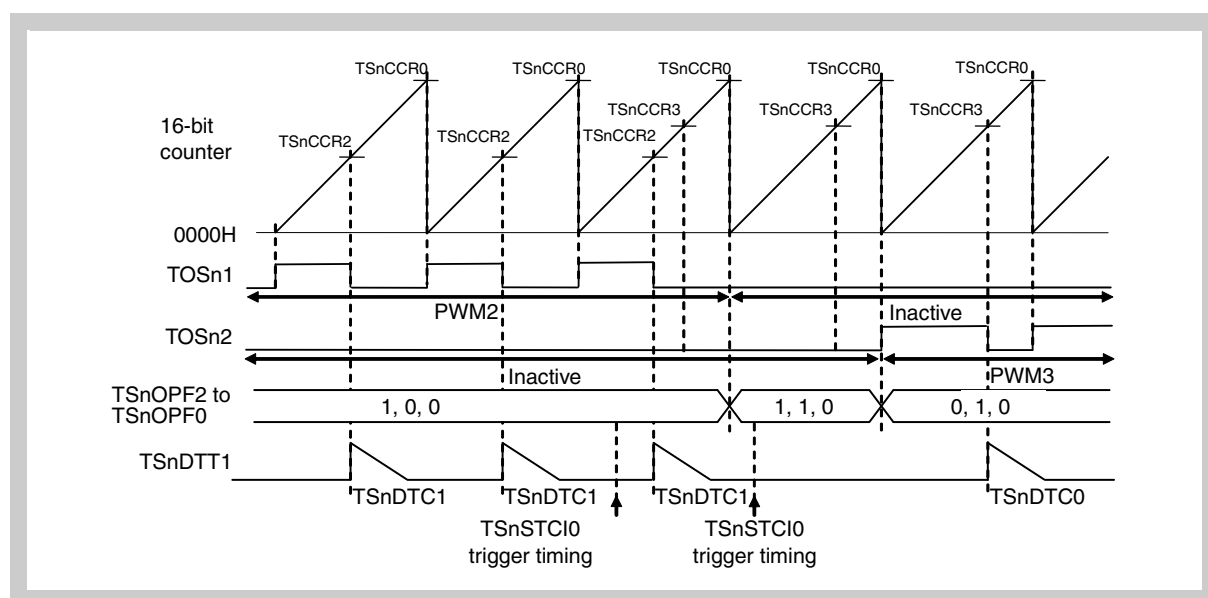
**Figure 17-151** Output example in case of rewriting of TSnCCR1 register

**(9) Dead time control in 120° excitation mode**

In the 120° excitation mode, dead time is added by the dead time control operation at the fall of each phase.

The dead time that is set to the TSnDTC1 register is inserted at the fall of the positive phase, and the dead time that is set to the TSnDTC0 register is inserted at the fall of the negative phase.

(TSnOPT5 register: TSnPOT bit = 1)



**Figure 17-152** Output switch example (switch with TSnIPC2 to TSnIPC0 bits of TSnOPT4 register)

**Caution** Dead time control has little effect on the timer output during normal operation. However, dead time control may affect the timer output under the following conditions.

- When noise occurs in input pattern during use of the pattern switch method
- When input pattern changes earlier than PWM cycle during use of the pattern switch method
- When the output pattern is forcibly changed by changing the TSnIPC2 to TSnIPC0 bit of the TSnOPT4 register in the trigger switch method
- When the switch method is changed
- When the current direction control bit (TSnIDC bit of TSnOPT7 register) is changed
- When software output control function is used
- When 180° excitation control is used

---

**(10) Output switch in 120° excitation mode**

In the 120° excitation mode, the output pattern can be controlled by writing the value to the TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register when the trigger switch method is used (TSnPOT of TSnOPT5 register is set to 1) and also the TSnADC bit of the TSnOPT5 register is set to 0 to control the output patterns. Dead time is secured by using hardware at the switch timing.

- 
- Caution**
1. The output level of the TOSn1 to TOSn6 pins before writing is kept when [1, 1, 1] or [0, 0, 0] is written to the TSnIPC2 to TSnIPC0 bits.
  2. Do not rewrite more than once to the TSnIPC2 to TSnIPC0 bits in a PWM cycle. If rewritten twice or more, the value just before PWM cycle match occurrence is reflected.
-

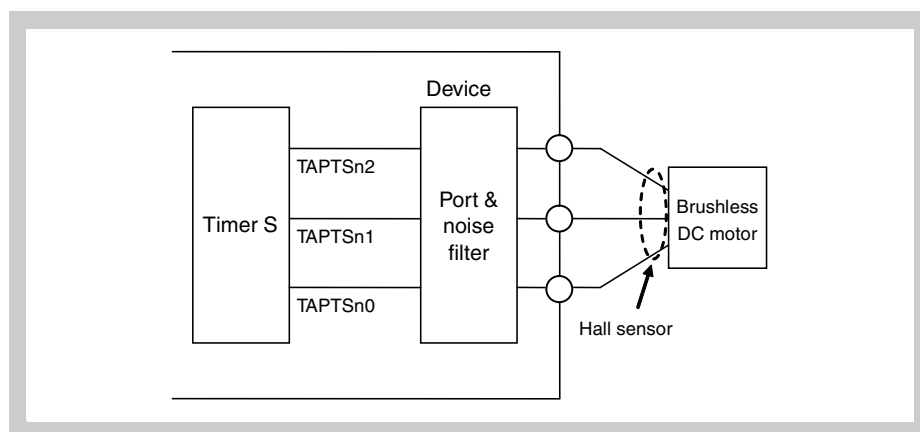
**(11) Operation when noise is generated in TAPTSn2 to TAPTSn0 pins in 120° excitation mode**

Brushless DC motor hall sensor input is assumed for the TAPTSn2 to TAPTSn0 pins.

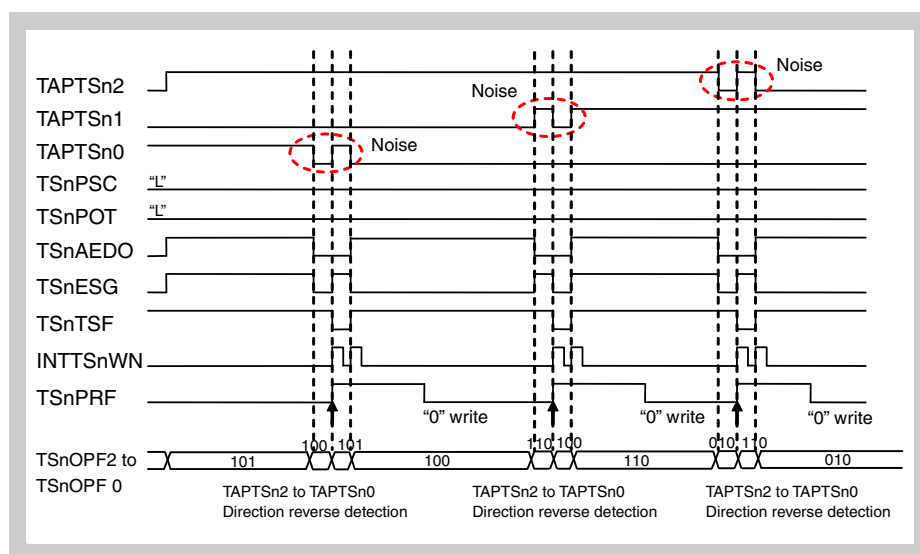
Noise may occur in the TAPTSn2 to TAPTSn0 pins in some systems. The operation in case of noise occurrence is described here.

Be sure to insert the noise filter circuit between the hall sensor and the TAPTSn2 to TAPTSn0 pins when designing system products.

Figure 17-154 on page 876 illustrates the case when noise is generated in the TAPTSn2 to TAPTSn0 pins during operation in the pattern switch method.



**Figure 17-153 Noise filter circuit connection example**



**Figure 17-154 Noise occurrence example upon level change of TAPTSn2 to TAPTSn0 pins (pattern switch method)**



**(a) Change timing of input pattern change detection signal (TSnAEDO)**

- TSnAEDO toggles at the input pattern (TAPTSn2 to TAPTSn0) change timing

**Caution** Be sure to specify the rotation direction with the TSnARD signal (TSnPSS of TSnOPT5 register set to 0) or the TSnPSC bit of the TSnOPT4 register (TSnPSS of TSnOPT5 register is set to 1).

(When TSnPSC = 0, or TSnARD = 0)

TAPTSn2 to TAPTSn0 after change									
		[0, 0, 0]	[1, 1, 1]	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]
Current TAPTSn2 to TAPTSn0	[0, 0, 0]	-	-	-	-	-	-	-	-
	[1, 1, 1]	-	-	-	-	-	-	-	-
	[1, 0, 1]	-	-	-	Toggle	-	-	-	-
	[1, 0, 0]	-	-	-	-	Toggle	-	-	-
	[1, 1, 0]	-	-	-	-	-	Toggle	-	-
	[0, 1, 0]	-	-	-	-	-	-	Toggle	-
	[0, 1, 1]	-	-	-	-	-	-	-	Toggle
	[0, 0, 1]	-	-	Toggle	-	-	-	-	-

(When TSnPSC = 1, or TSnARD = 1)

TAPTSn2 to TAPTSn0 after change									
		[0, 0, 0]	[1, 1, 1]	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]
Current TAPTSn2 to TAPTSn0	[0, 0, 0]	-	-	-	-	-	-	-	-
	[1, 1, 1]	-	-	-	-	-	-	-	-
	[1, 0, 1]	-	-	-	-	-	-	-	Toggle
	[1, 0, 0]	-	-	Toggle	-	-	-	-	-
	[1, 1, 0]	-	-	-	Toggle	-	-	-	-
	[0, 1, 0]	-	-	-	-	Toggle	-	-	-
	[0, 1, 1]	-	-	-	-	-	Toggle	-	-
	[0, 0, 1]	-	-	-	-	-	-	Toggle	-

**(b) 3-phase encode signal (TSnESG) change timing**

- TSnESG toggles at the input pattern (TAPTSn2 to TAPTSn0) change timing

TAPTSn2 to TAPTSn0 after change									
		[0, 0, 0]	[1, 1, 1]	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]
Current TAPTSn2 to TAPTSn0	[0, 0, 0]	-	-	-	-	-	-	-	-
	[1, 1, 1]	-	-	-	-	-	-	-	-
	[1, 0, 1]	-	-	-	Toggle	-	-	-	Toggle
	[1, 0, 0]	-	-	Toggle	-	Toggle	-	-	-
	[1, 1, 0]	-	-	-	Toggle	-	Toggle	-	-
	[0, 1, 0]	-	-	-	-	Toggle	-	Toggle	-
	[0, 1, 1]	-	-	-	-	-	Toggle	-	Toggle
	[0, 0, 1]	-	-	Toggle	-	-	-	Toggle	-

**(c) Change timing of TOSn1 to TOSn6 pins**

- In the case of the pattern switch method, the output pattern is changed when the input signal of the TAPTSn2 to TAPTSn0 pins<sup>Note</sup> is changed. The output switches in case of the simultaneous change of two or more pins.
- Output pattern changes at the TSnSTCI0 and TSnSTCI1 rising edges in the trigger switch method. Output also changes when writing to the TSnIPC2 to TSnIPC0<sup>Note</sup> bits of the TSnOPT4 register.

**Note** The output level of the TOSn0 to TOSn6 pins remains the level before change when changed to [0, 0, 0] or [1, 1, 1].

**(d) TSnTSF flag change timing**

- TSnTSF toggles at the input pattern (TAPTSn2 to TAPTSn0) change timing.

TAPTSn2 to TAPTSn0 after change									
		[0, 0, 0]	[1, 1, 1]	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]
Current TAPTSn2 to TAPTSn0	[0, 0, 0]	-	-	-	-	-	-	-	-
	[1, 1, 1]	-	-	-	-	-	-	-	-
	[1, 0, 1]	-	-	-	0	-	-	-	1
	[1, 0, 0]	-	-	1	-	0	-	-	-
	[1, 1, 0]	-	-	-	1	-	0	-	-
	[0, 1, 0]	-	-	-	-	1	-	0	-
	[0, 1, 1]	-	-	-	-	-	1	-	0
	[0, 0, 1]	-	-	0	-	-	-	1	-

**(e) TSnNDF flag setting timing**

- The setting timing is when two or more pins of TAPTSn2 to TAPTSn0 change simultaneously. The setting is cleared when writing 0.

**(f) TSnPRF flag setting timing**

- The setting timing is when TSnTSF changes. The setting is cleared when writing 0.

**(g) TSnPEF flag setting timing**

- The setting timing is when the value [0, 0, 0] or [1, 1, 1] is input to the TAPTSn2 to TAPTSn0 pins. The setting is cleared when writing 0.

**(12) Basic control flow in 120° excitation mode**

The 120° excitation mode includes 8 control statuses shown in *Table 17-37 on page 880*.

120° excitation control with the pattern switch method is enabled when setting TSnPOT of the TSnOPT5 register to 0.

This is defined as the phase fix control. In the phase fix control, delay from hall sensor or delay from sensor level detection to timer output needs to be considered. However, acceleration/deceleration is possible only by PWM duty change.

The 120° excitation control with the trigger switch method is used when TSnPOT = 1.

This is defined as phase variable control. In phase variable control, timer output pattern is set prior to hall sensor and acceleration/deceleration can be controlled by using the phase difference. However, phase variable control is more complex to control than phase fix control because the offset width for the hall sensor and the predicted value for the hall sensor must be considered. Refer to “*Timer Interconnection*” on page 1020 for details.

The motor rotation direction can be set with the TSnPSC bit of the TSnOPT4 register when TSnPOT = 1, TSnPSS = 1 of TSnOPT5 register. Set TSnPSC = 0 in normal rotation, and TSnPSC = 1 in reverse rotation.

Control direction (acceleration/deceleration) is set with the TSnIDC bit of the TSnOPT7 register. Acceleration control is performed when the setting value is the same as the motor rotation direction (TSnPSC setting value), and deceleration control is performed when the setting value is different from the motor rotation direction.

**Table 17-37 Timer Control Status**

Status	TSnOPT4 register: TSnPSC	TSnOPT5 Register: TSnTSF	TSnOPT7 register: TSnIDC	TSnOPT5 register: TSnPOT	Control
A	-	0	0	0	Normal rotation/acceleration/phase fix
B	0	-	0	1	Normal rotation/acceleration/phase variable
C	0	-	1	1	Normal rotation/deceleration/phase variable
D	-	0	1	0	Normal rotation/deceleration/phase fix
E	-	1	1	0	Reverse rotation/acceleration/phase fix
F	1	-	1	1	Reverse rotation/acceleration/phase variable
G	1	-	0	1	Reverse rotation/deceleration/phase variable
H	-	1	0	0	Reverse rotation/deceleration/phase fix

Normally, control starts under the start status in which motor rotation stops. First, rotate the motor by using phase fix control to rotate from the stop status. Second, switch to the phase variable control to accelerate up to the high speed rotation. In the phase variable control, change the timer output earlier than the hall sensor change point in combination with timer T.

When decelerating from high speed rotation, change the control to the deceleration control by rewriting only the TSnIDC bit of the TSnOPT7 register. After decelerating revolution to the low-speed rotation, make the PWM duty smaller up to the stop status. This is the overall motor control flow.

*Figure 17-155 and Figure 17-156 show the status changes.*

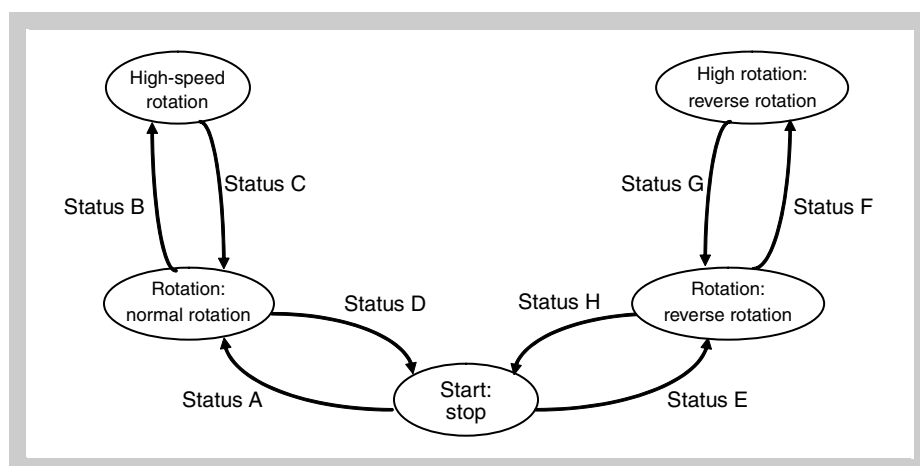


Figure 17-155 Status transition diagram

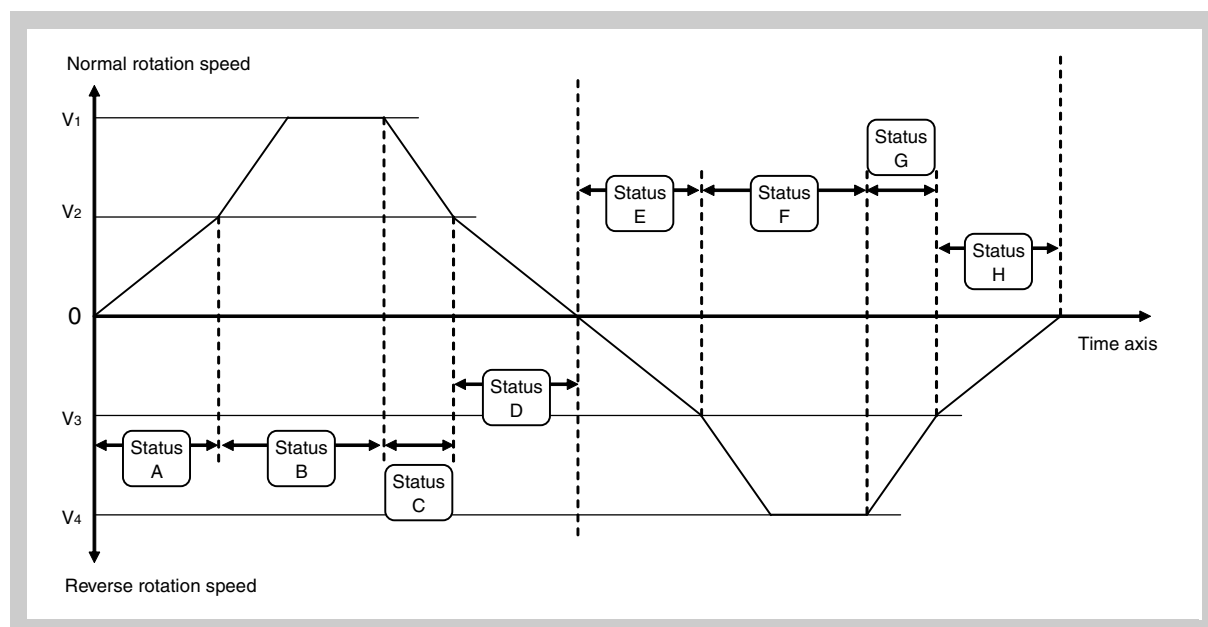


Figure 17-156 Relationship between status transition and motor rotation speed

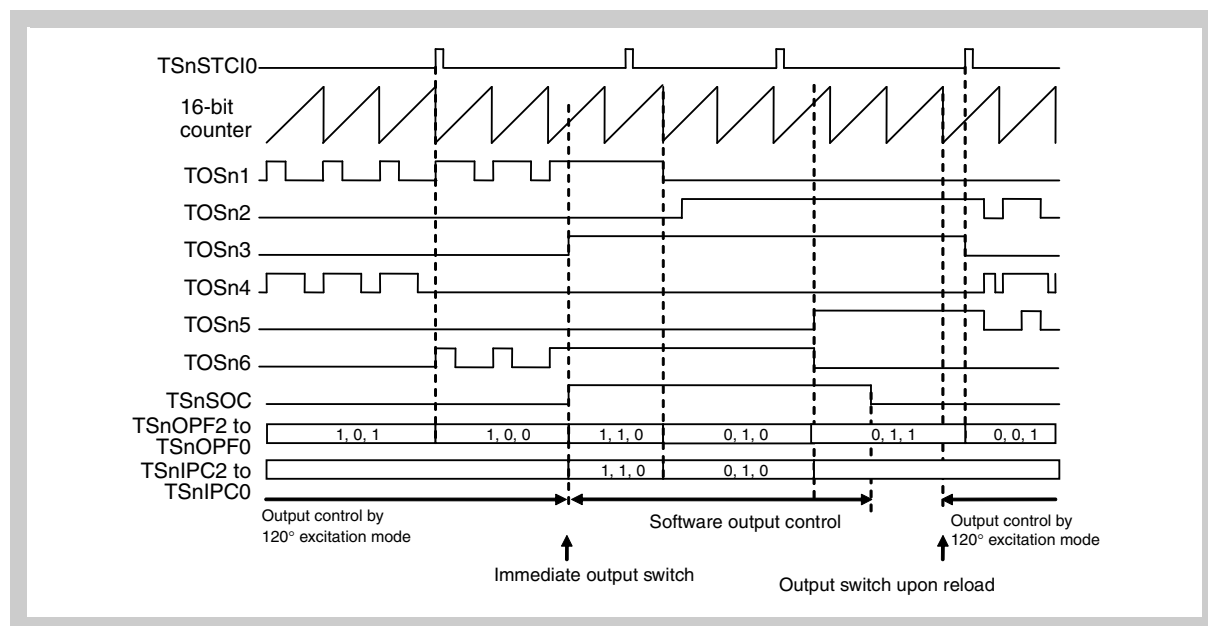
**Note**  $V_1$ ,  $V_4$ : Normal rotation/reverse rotation high-speed rotational speed  
 $V_2$ ,  $V_3$ : Normal rotation/reverse rotation low-speed rotational speed

**(13) Software output control function in 120° excitation mode**

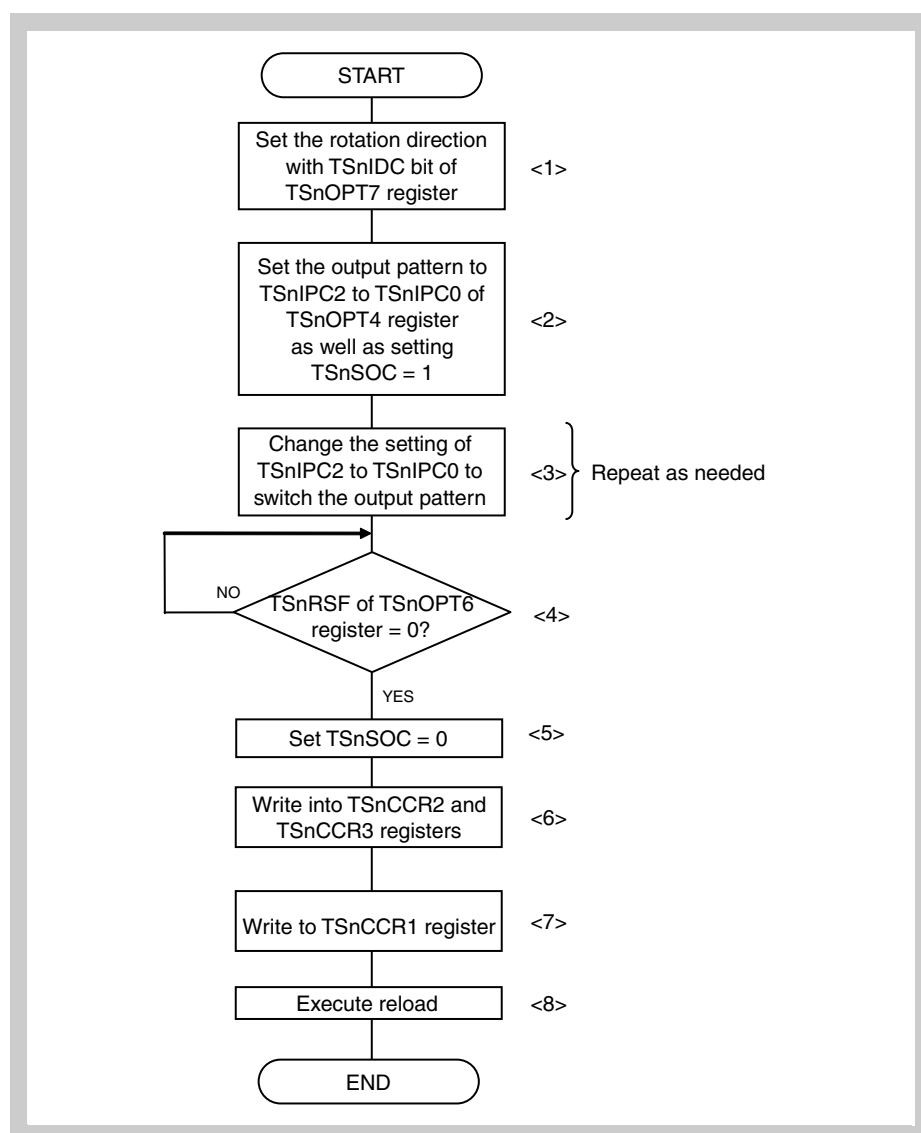
In the 120° excitation mode, timer output is controlled with software control by using the TSnSOC and TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register, and the TSnIDC bit of the TSnOPT7 register.

Output is switched immediately when setting TSnSOC = 1 as shown in *Figure 17-157*. The dead time period is assured by setting the dead time. Then, set TSnSOC to 0 in order to switch from the software output control to 120° excitation control. Output is held at this timing and is switched to be controlled by using the 120° excitation mode when reloading.

Refer to “Software output function” on page 936 for details on the software output control function.



**Figure 17-157** Example of software output control function switched from 120° excitation mode

**(a) Software output control procedure****Figure 17-158 Processing flow in software output control**

The software output control processing procedure is described below.

- <1> Set TSnIDC to determine the rotation direction. There is a 180° phase difference in the timer output between when TSnIDC = 0 and TSnIDC = 1. In the software output control function, timer output is not changed only by rewriting the bit. However, 120° excitation control output pattern is changed if a cycle match occurs before <2>, so prevent cycle match before <2> in scheduling.
- <2> Set the pattern to be output to TSnIPC2 to TSnIPC0 and also set TSnSOC to 1 in order to provide software output.
- <3> Change the output pattern setting of TSnIPC2 to TSnIPC0 to change the timer output.  
The registers that can be changed during software control are as follows.  
TSnCTL0 register: TSnCE bit, TSnOPT1 to TSnOPT4 registers,  
TSnOPT7 register: TSnIDC, TSnTOS bits, TSnCCR0 to TSnCCR5 registers, TSnDTC0 register, TSnDTC1 register
- <4> Check that the reload request flag (TSnRSF) is set to 0. In case of TSnRSF = 1, do not proceed to the next step before setting TSnRSF = 0.

- 
- <5> Software control release is started by setting TSnSOC = 0 (Not yet released in this step).
  - <6> Set the compare register which is necessary after the release of software output control. Proceed to the next step when there is no need to change. Change the register with the reload function in this step if necessary.
  - <7> Write to the TSnCCR1 register to perform reloading.
  - <8> Reload is executed and software output is released.

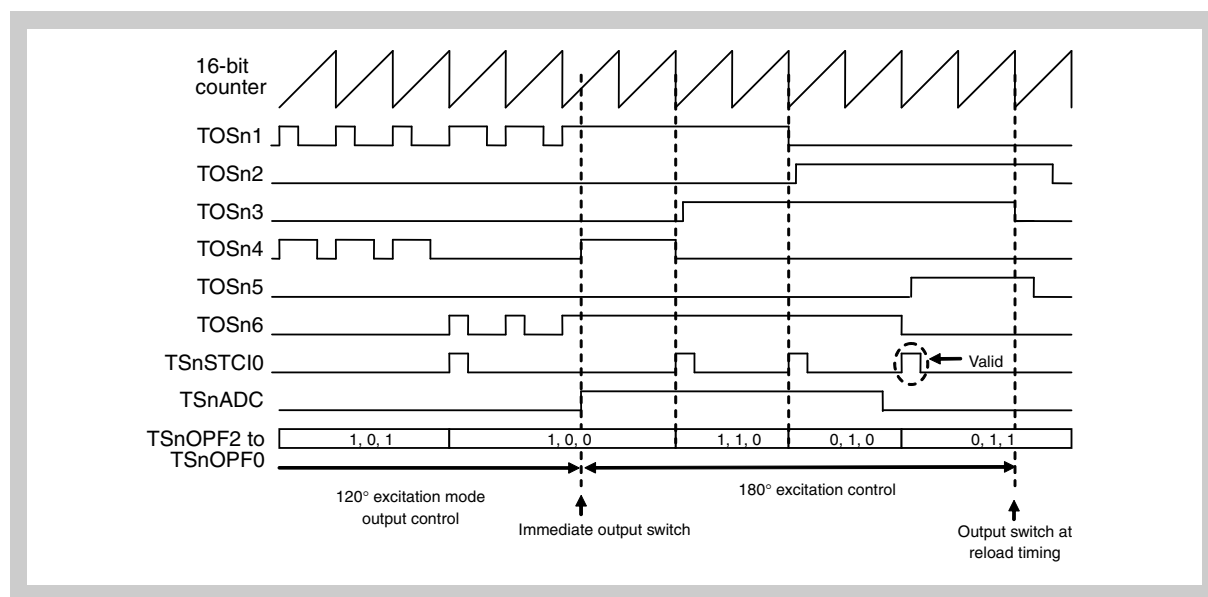
- 
- Caution**
1. Any change of TSnIPC2 to TSnIPC0 is prohibited after setting TSnSOC to 0.
  2. Be sure to perform reloading after the execution of steps <4>, <5>, <6> and <7>. Software output cannot be released if reload cannot be performed.
-



**(14) 180° excitation control function in 120° excitation mode**

In the 120° excitation mode, the 180° excitation control function can be used with TSnADC and TSnPOT of the TSnOPT5 register, the TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register, and the TSnIDC bit of the TSnOPT7 register. Output is switched immediately when setting TSnADC = 1 as shown in *Figure 17-159 on page 885*. The dead time period is assured by setting the dead time. Then, trigger input to the output pattern switch trigger (TSnSTCI1 and TSnSTCI0 signals or TAPTSn2 to TAPTSn0 pins) leads to the output change according to the TSnPSC bit setting of the TSnOPT4 register. Output is held when clearing TSnADC to 0. Then the function is switched to the 120° excitation mode output control upon reload.

Refer to “180° excitation control function” on page 938 for details on 180° excitation control.



**Figure 17-159** 180° excitation control function switch example in 120° excitation mode

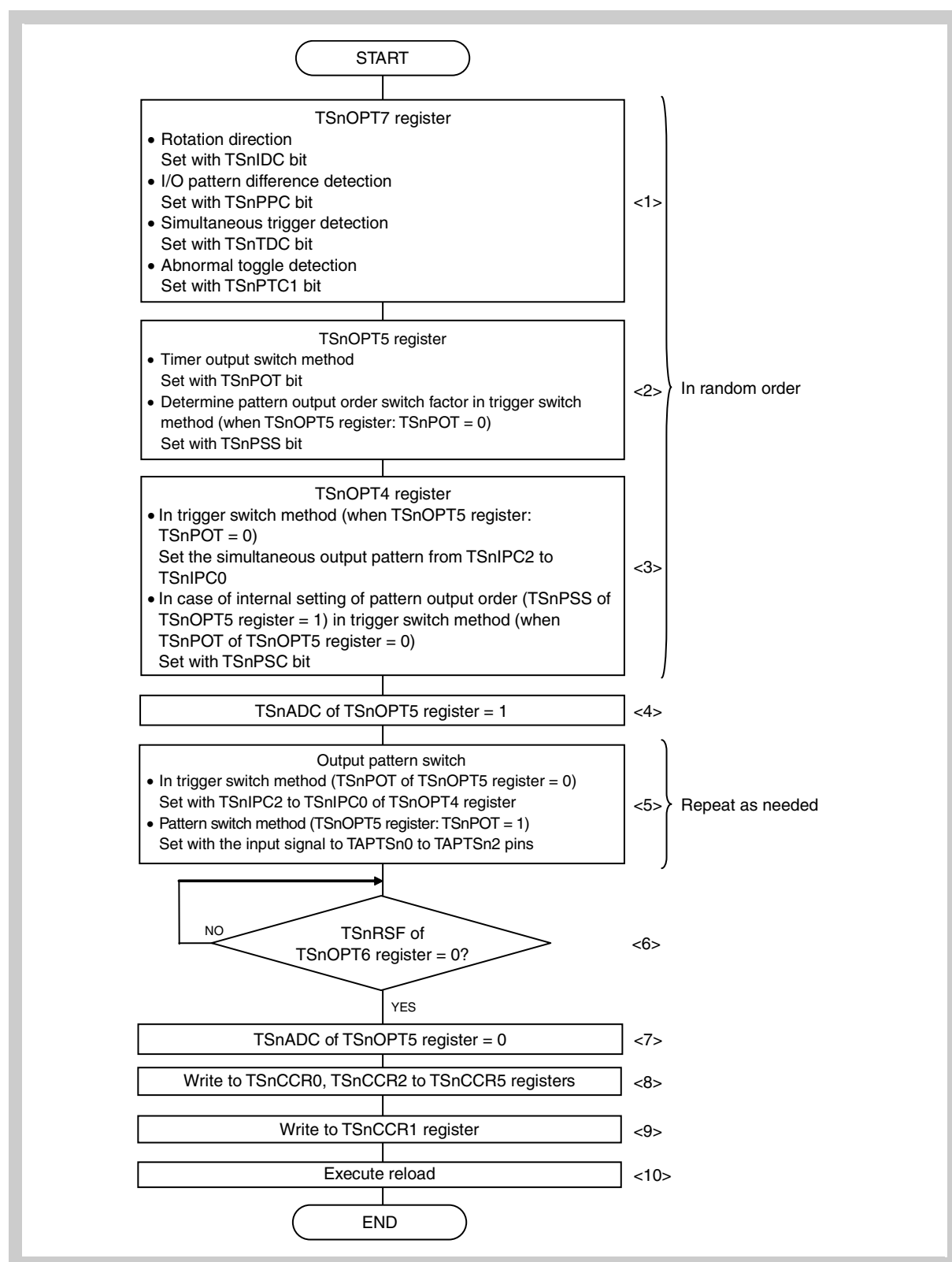
**(a) 180° excitation control processing procedure**

Figure 17-160 180° excitation control function processing flow

The following is the procedure for transition to 180° excitation control.

- <1> TSnOPT7 register setting
  - Set a value to the TSnIDC bit to select the rotation direction.
  - Set a value to the TSnPPC bit to select to enable (1) or disable (0) the pattern phase difference detection flag (TSnPPF) of input (TAPTSn2 to TAPTSn0 pins) or output (TSnOPF2 to TSnOPF0 flags) with the trigger switch method (TSnPOT of TSnOPT5 register = 1).
  - Set a value to the TSnTDC bit to select either enable (1) or disable (0) the TSnSTCI0, TSnSTCI1 signal simultaneous trigger detection flag (TSnTDF) with the trigger switch method.
  - Set a value to the TSnPTC1, TSnPTC0 bits to select either enable (1) or disable (0) for the TAPTSn2 to TAPTSn0 pin abnormal toggle detection flag (TSnPTE).
- <2> TSnOPT5 register setting
  - Set a value to the TSnPOT bit to select the pattern output trigger. The pattern switch method where output is switched with the input pattern of TAPTSn2 to TAPTSn0 pins can be enabled by setting TSnPOT = 0, and the trigger switch method where output is switched with rising edge of TSnSTCI0, TSnSTCI1 signals can be enabled by setting TSnPOT = 1.
  - Set a value to the TSnPSS bit to select the pattern output order switch factor.
- <3> TSnOPT4 register setting
  - Set a value to the TSnPSC bit to select the pattern output order for TSnPSS = 1 of the TSnOPT5 register in the trigger switch method.
  - Set a value to TSnIPC2 to TSnIPC0 to select the initial pattern for trigger switch method.
- <4> Start the 180° excitation control function by setting TSnADC = 1.
- <5> Output pattern switch
  - Pattern switch method
    - In the pattern switch method, output is switched with the input pattern of the TAPTSn2 to TAPTSn0 pins. However, the previous output level is held if [1, 1, 1] or [0, 0, 0] is input to the TAPTSn2 to TAPTSn0 pins.
  - Trigger switch method
    - In the trigger switch method, output is switched with the rising edge of the TSnSTCI0 and TSnSTCI1 signals.
    - Set the appropriate value to TSnIPC2 to TSnIPC0 if forcible change is needed for timer output. Writing to TSnIPC2 to TSnIPC0 is prior to the rising edge of the TSnSTCI0, TSnSTCI1 signals in case of these simultaneous occurrences.
  - Switch method can be changed during operation.
    - The registers that can be rewritten during the 180° excitation control function operation are as follows.
    - TSnCTL0 register: TSnCE bit, TSnOPT1 to TSnOPT5 registers, TSnOPT7 register, TSnCCR0 to TSnCCR5 registers, TSnDTC0 register, TSnDTC1 register

---

**Caution** Setting of TSnSOC of the TSnOPT4 register = 1 and TSnADC of TSnOPT5 register = 1 is prohibited.

---

**Note** <1> to <3> can be performed in random order.

- 
- <6> Check that the reload request flag (TSnRSF) is set to 0. If TSnRSF = 1, do not proceed to the next step before setting TSnRSF = 0.
  - <7> Start the 180° excitation control function by setting TSnADC = 0. (Not yet released in this step).  
After the change of TSnADC to 0, the output is switched upon the switching using the pattern/trigger switch method until the reload is performed. The output can be changed by writing to TSnIPC2 to TSnIPC0.
  - <8> Set the compare register that is necessary after the release of 180° excitation control. Proceed to the next step if there is no need for change. Change the register with the reload function in this step if necessary.
  - <9> Write to the TSnCCR1 register to perform reloading.
  - <10> Reload is executed and software output is released.

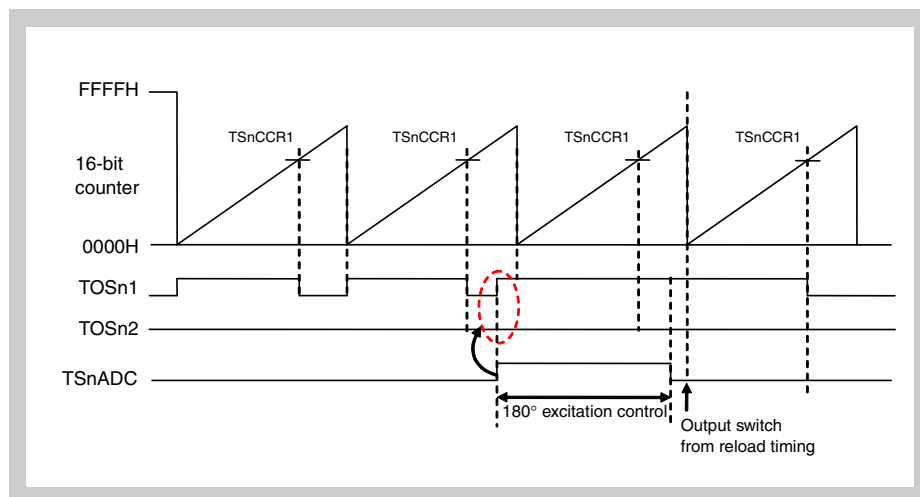
---

**Caution** Be sure to execute reload after execution of steps <6>, <7>, <8> and <9>. Software output cannot be released if reload cannot be executed.

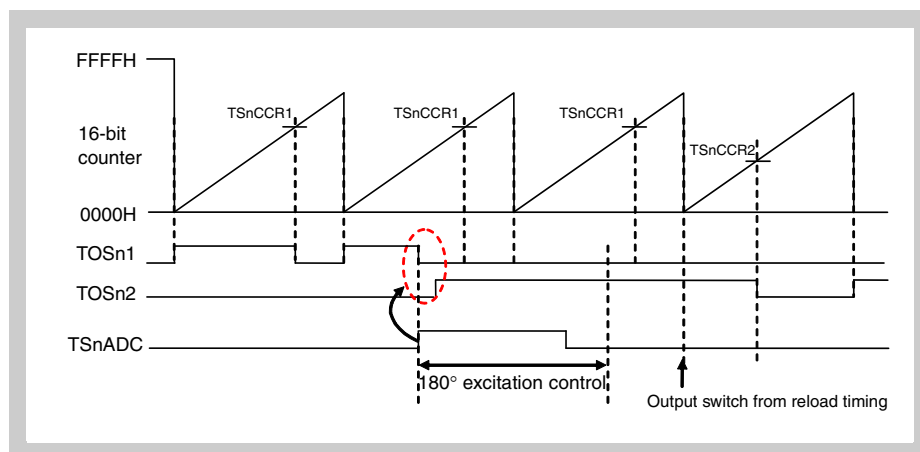
---

**(b) 180° excitation control switch timing from 120° excitation mode**

Output is switched immediately when switching from the 120° excitation mode to 180° excitation control. Dead time control by hardware is inserted if the output is reverse to the current output level. The output level is held until the reload timing when switching from 180° excitation control to the 120° excitation mode.

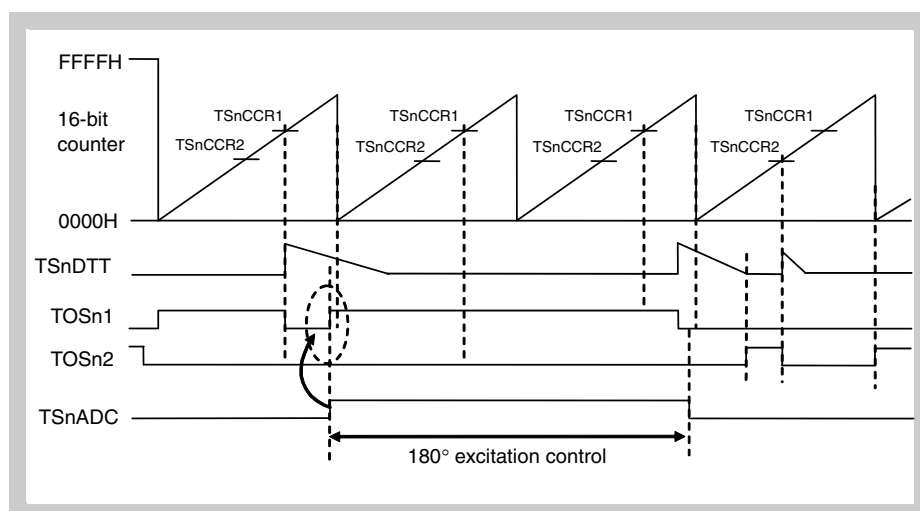


**Figure 17-161** Output at switch timing between 120° excitation mode and 180° excitation control (in case of output reverse)



**Figure 17-162** Output at switch timing between 120° excitation mode and 180° excitation control (in case of output not reversed)

The output switched during dead time count is set without waiting for the dead time count finish in case of the same phase, or is set after the dead time count finish in case of negative phase.



**Figure 17-163** Output at switch timing between 120° excitation mode and 180° excitation control (during dead time count)

### 17.10.11 Special 120° excitation mode

#### (1) Overview of special 120° excitation mode

In the special 120° excitation mode, 120° excitation control is performed by detecting the external input (pattern: TAPTSn2 to TAPTSn0 pins, trigger: TSnSTCI0 signal) and by switching the timer output (TOSn1 to TOSn6 pins). Motor systems that can be controlled in the special 120° excitation mode are as follows.

- System where a hall sensor is connected at 120° intervals against the motor rotation
- System where normal/reverse rotation is possible by outputting as follows, in response to the hall sensor input

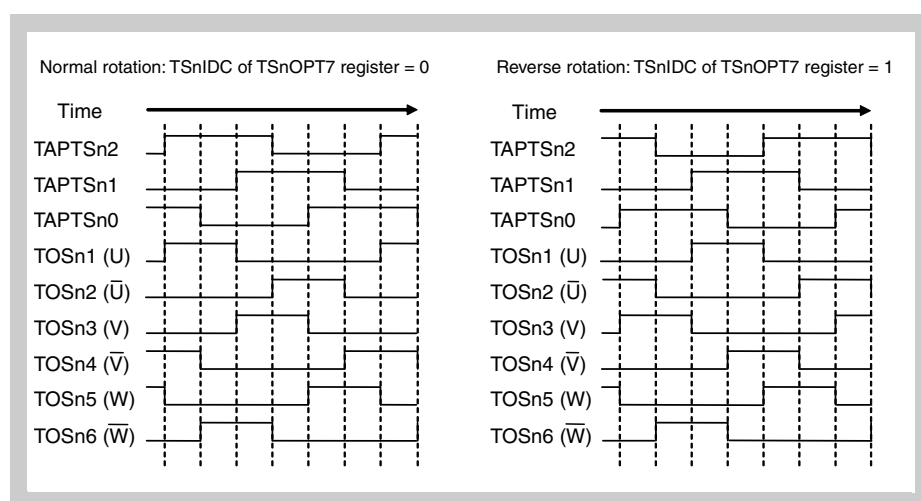


Figure 17-164 Normal/reverse rotation example

- Note**
1. TSnOPT4 register: TSnSOC = 0, TSnOPT5 register: TSnPOT = 0
  2. The special 120° excitation mode is valid when TSnMD3 to TSnMD0 of the TSnCTL1 register are set to [1, 0, 1, 1].

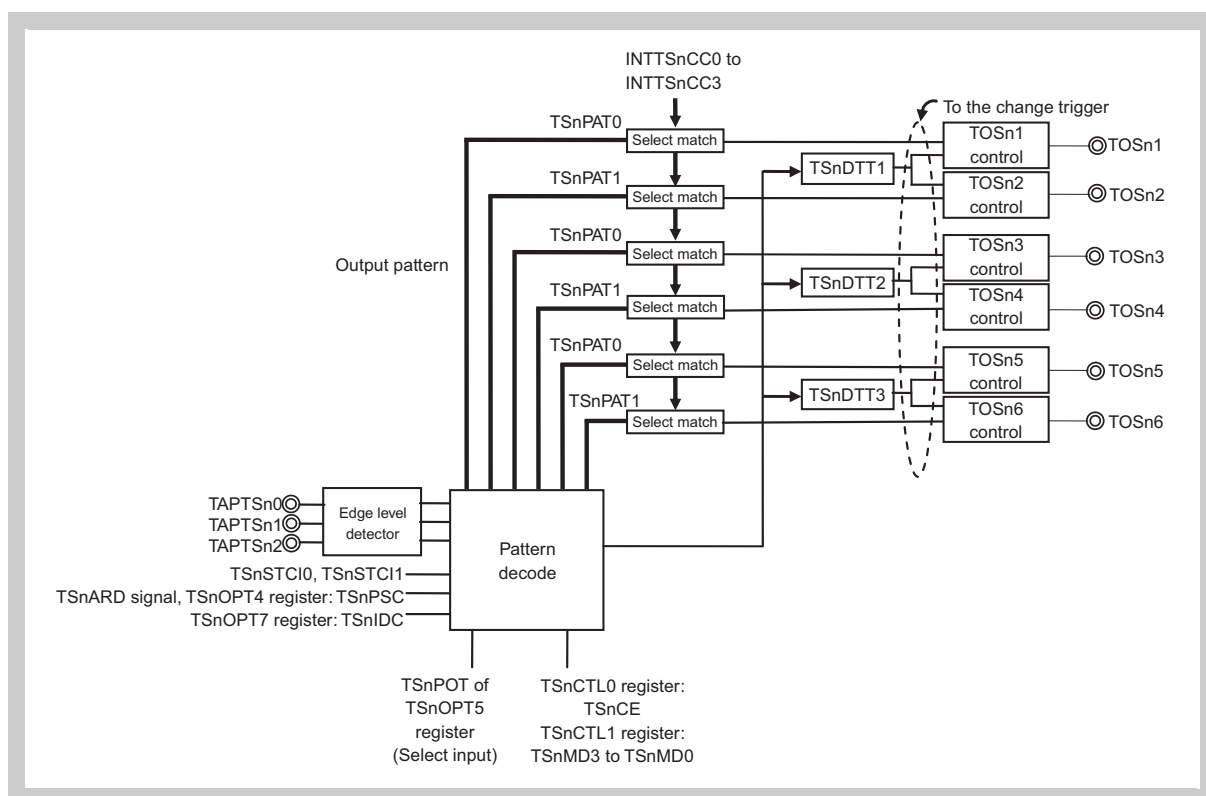


Figure 17-165 Configuration diagram in special 120° excitation mode

Table 17-38 Output Pattern of TAPTSn2 to TAPTSn0 Pins (Normal Rotation)

Output pin	TAPTSn2 to TAPTSn0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	Special pattern 1	Special pattern 2	Special pattern 3	INACT	INACT	Special pattern 0	Note	Note
TOSn2	INACT	INACT	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	Note	Note
TOSn3	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Special pattern 3	INACT	Note	Note
TOSn4	Special pattern 6	Special pattern 7	INACT	INACT	Special pattern 4	Special pattern 5	Note	Note
TOSn5	Special pattern 3	INACT	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Note	Note
TOSn6	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	INACT	INACT	Note	Note

**Note** The output level of the TOSn0 to TOSn6 pins remains the level before change when changed to [0, 0, 0] or [1, 1, 1].



- 
- Caution** Special pattern 3 and special pattern 4 cannot be set to be active simultaneously. Special pattern 0 and special pattern 7 cannot be set to be active simultaneously either. The simultaneous active setting results in the following operations.
1. When TOSn1, TOSn3 and TOSn5 are set to be active prior to TOSn2, TOSn4, TOSn6  
Only TOSn1, TOSn3 and TOSn5 are active in the simultaneous active period.
  2. When TOSn2, TOSn4, and TOSn6 are set to be active prior to TOSn1, TOSn3 and TOSn5  
Only TOSn2, TOSn4, and TOSn6 are active in the simultaneous active period.
  3. When “TOSn1, TOSn3 and TOSn5” and “TOSn2, TOSn4, and TOSn6” are set to be active simultaneously  
Only TOSn1, TOSn3 and TOSn5 are active in the simultaneous active period.
- 

There are two methods based on the external input in the special 120° excitation mode.

- Pattern switch method: Pattern input of 3-input (hall sensor, etc.)
- Trigger switch method: Trigger input (encoder circuit, offset, etc.)

#### (a) Pattern switch method (TSnPOT of TSnOPT5 register = 0)

Level detection is performed for the TAPTSn2 to TAPTSn0 pins (three patterns of input from hall sensor) to decode the signals after level detection. Pattern output of the TOSn1 to TOSn6 pins (the output determined with the values of the TSnCCR1 to TSnCCR3, TSnPAT0, and TSnPAT1 registers) is selected by judging the decode result. Dead time control is performed by the operation of the dead time counter at respective phase signal fall timing to ensure dead time insertion.

With the pattern switch method, the decoded pattern is output provided by using data from the input pattern (TAPTSn2 to TAPTSn0 pins), current direction control bit (TSnIDC of the TSnOPT7 register), and TAPTSn2 to TAPTSn0 order detection flag (TSnTSF of the TSnOPT5 register). *Figure 17-166 on page 894 and Figure 17-167 on page 895* show the timer output when the TAPTSn2 to TAPTSn0 pins are changed. The output pattern is switched to the pattern corresponding to the input if any error occurs in the input pattern when switching from input pattern 1 to input pattern 4.

The pattern set with the input level of the TAPTSn2 to TAPTSn0 pins, the TSnIDC and TSnARD signals, (TSnPSS of TSnOPT5 register = 0), and the TSnPSC bit of the TSnOPT4 register (TSnPSS of TSnOPT5 register = 1) are output immediately after the operation starts (TSnCE of TSnCTL0 register = 1/TSnCE = 1 of TSnCTL0 register). The output pattern is determined with the TSnTSF flag instead of the TSnPSC bit after the value of the TSnTSF flag is determined.

TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,  
 TSnOPT5 register: TSnPOT = 0,  
 and TSnOPT7 register: TSnIDC = 0.

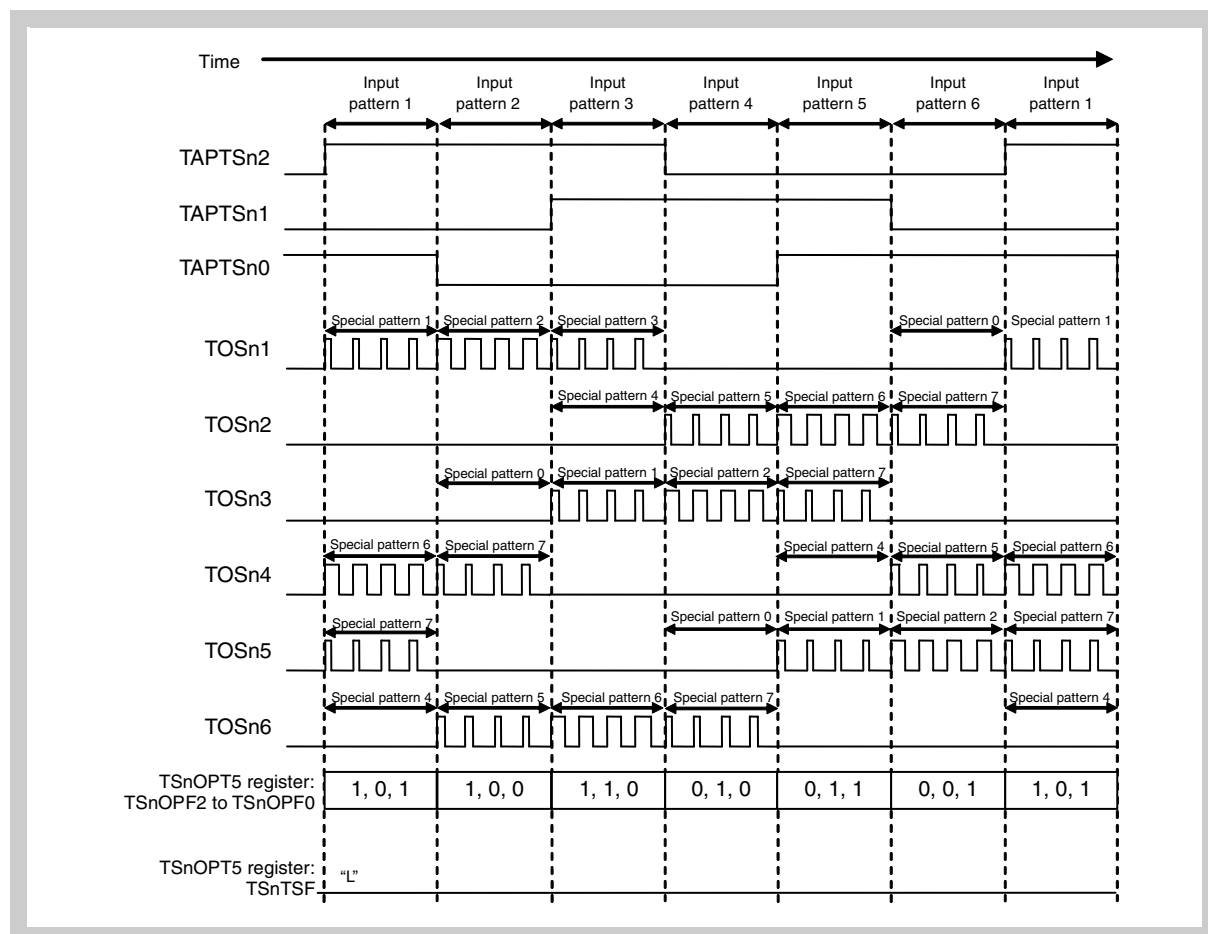


Figure 17-166 Special 120° excitation mode operation example in use of pattern switch method (normal rotation)

TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,  
 TSnOPT5 register: TSnPOT = 0,  
 and TSnOPT7 register: TSnIDC = 1.

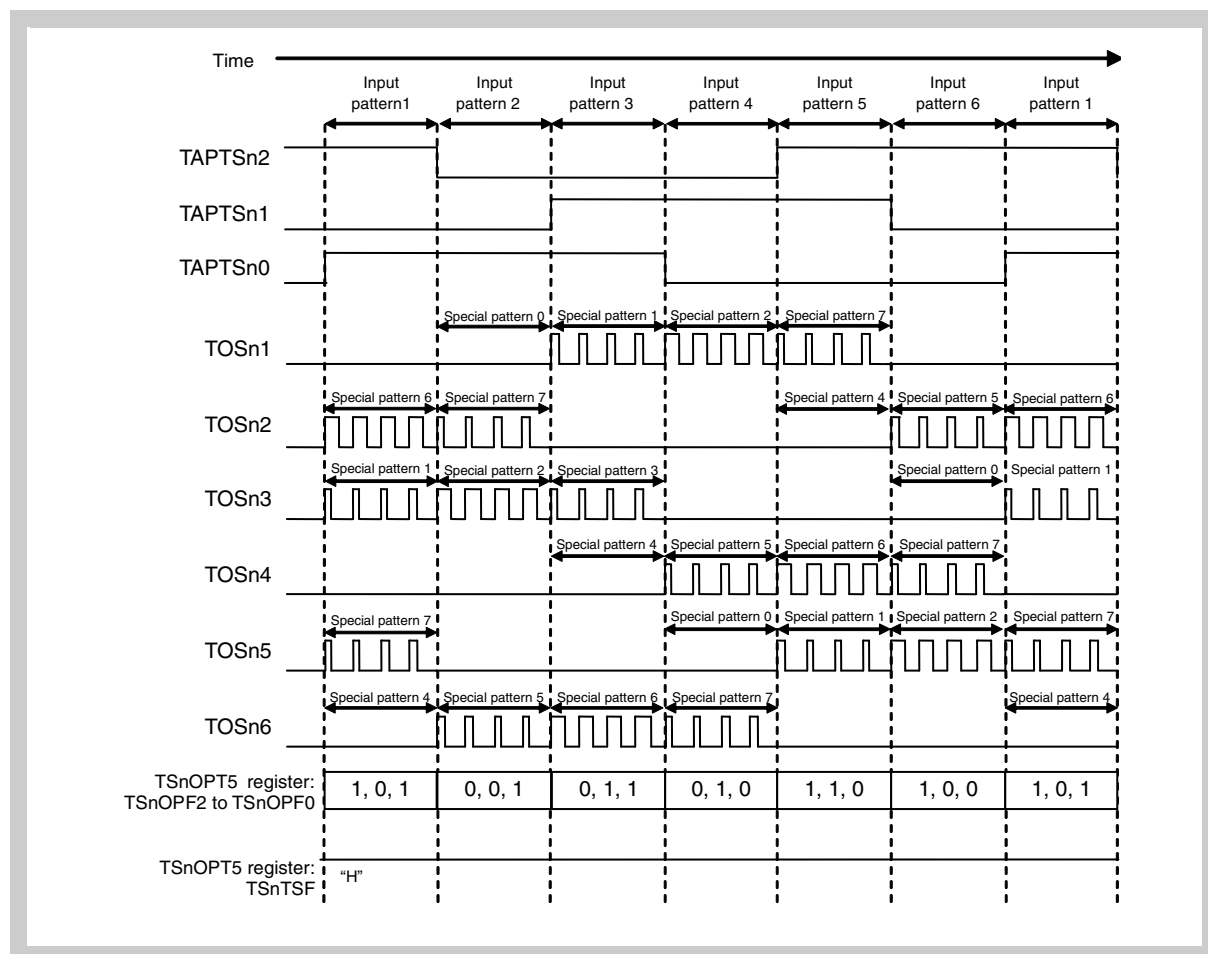


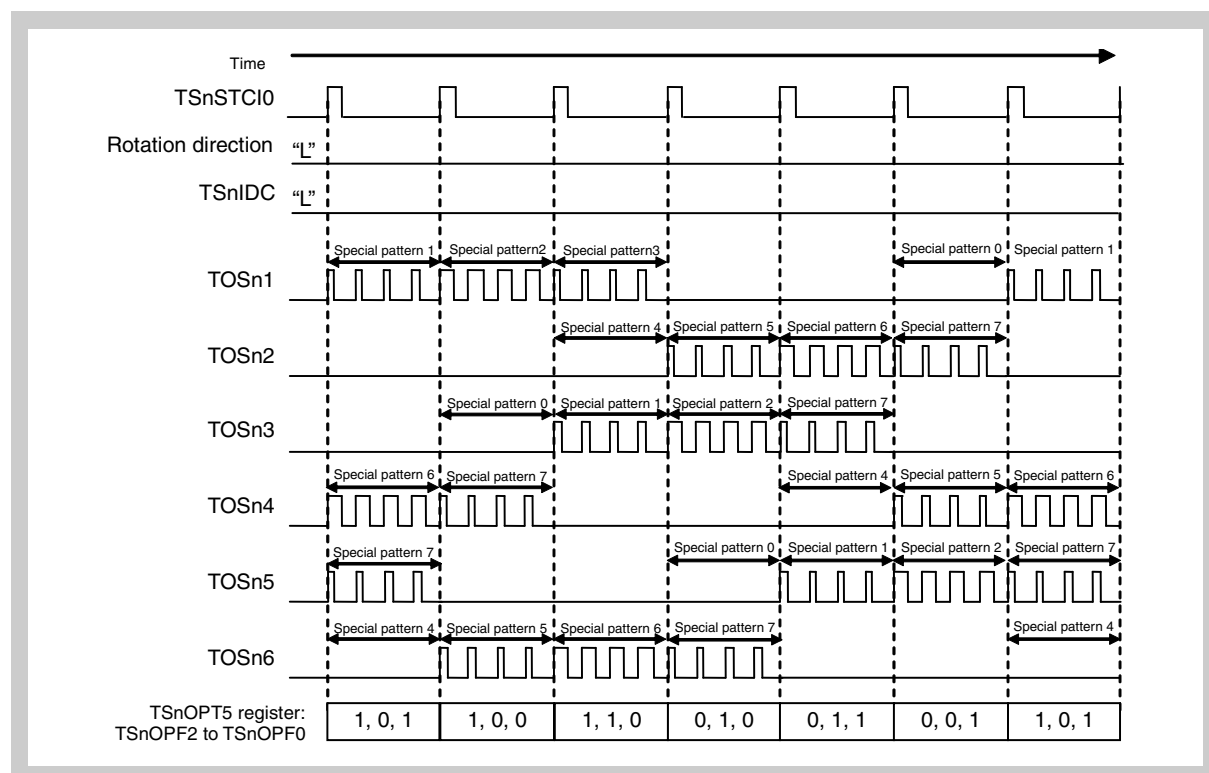
Figure 17-167 Special 120° excitation mode operation example in use of pattern switch method (reverse rotation)

**(b) Trigger input switch method (TSnOPT of TSnOPT5 register = 1)**

With the trigger input switch method, rise of the TSnSTCI0 and TSnSTCI1 signals is detected to generate the output switch timing. Timer output initial pattern is set with the TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register. The output patterns after the initial pattern are determined with the TSnIPC2 to TSnIPC0 bits, rotation direction or the TSnIDC bit of the TSnOPT7 register. There are four patterns that can be output according to the combination of rotation direction and the TSnIDC bit.

TSnOPT5 register		Rotation direction
TSnPOT	TSnPSS	
1	0	Determine rotation direction by using input signal to TSnARD
	1	Determine rotation direction by using TSnPSC bit

TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,  
 TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
 TSnOPT7 register: TSnIDC = 0

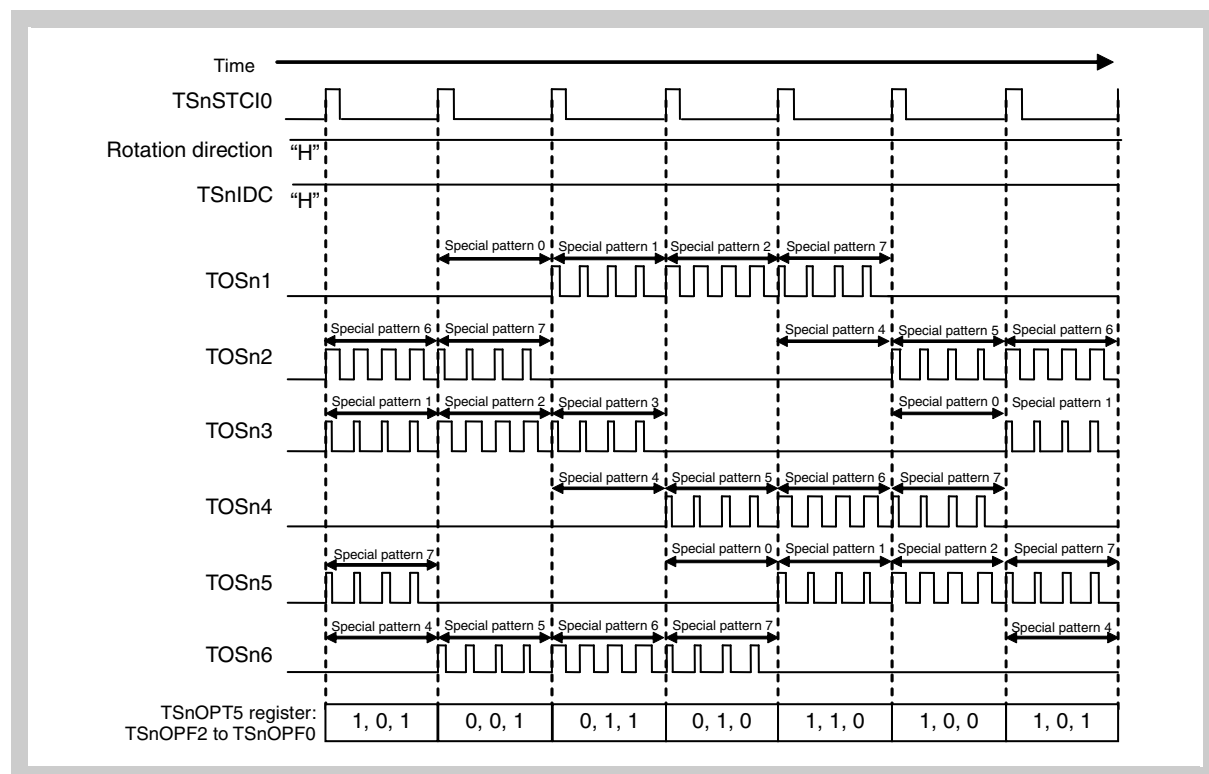


**Figure 17-168** Special 120° excitation mode operation example in use of trigger input switch method (normal rotation)

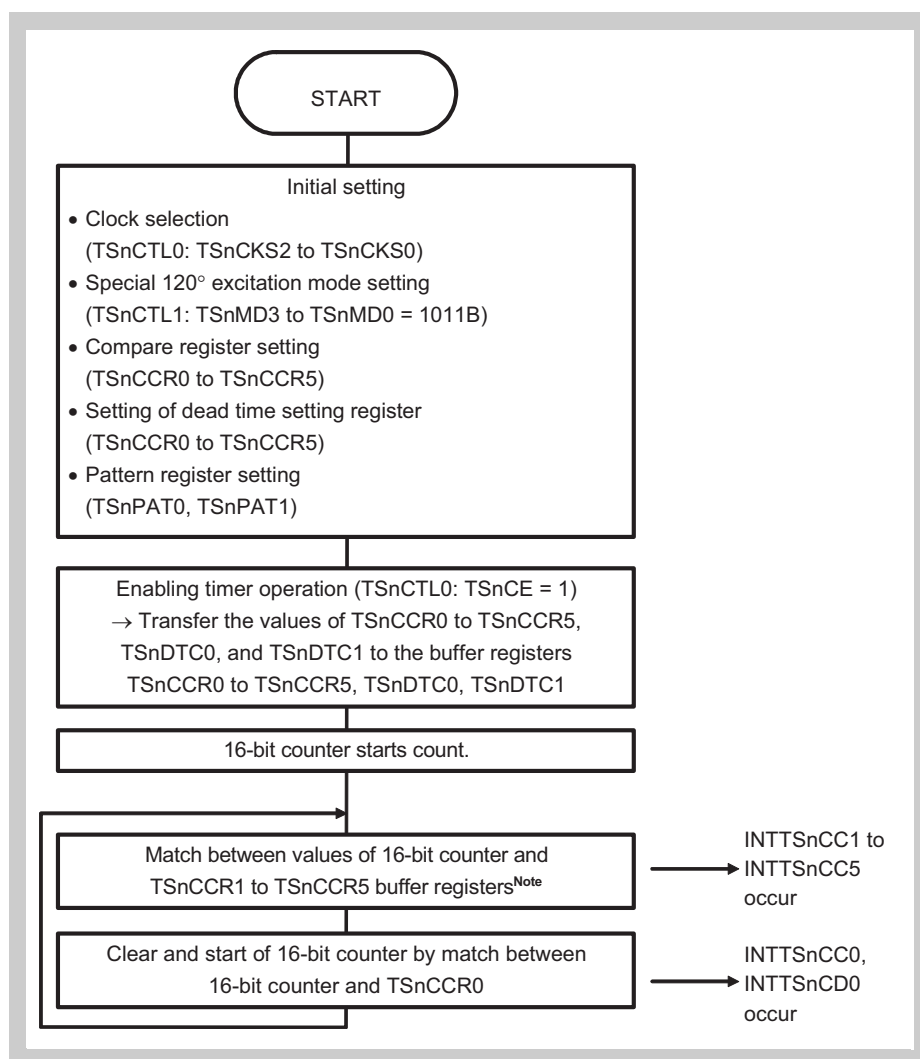
TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,

TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,

TSnOPT7 register: TSnIDC = 1



**Figure 17-169** Special 120° excitation mode operation example in use of trigger input switch method (reverse rotation)



**Figure 17-170 Basic operation flow in special 120° excitation mode**

**Note** The 16-bit counter is not cleared upon a match between the values of the 16-bit counter and the TSnCCR1 to TSnCCR5 buffer registers.

**(2) Special 120° excitation mode operation List****(a) Register rewriting**

Register	Rewriting method	Rewriting during operation	Function
TSnCCR0	Reload	Possible	Cycle
TSnCCR1 to TSnCCR3	Reload	Possible	PWM duty
TSnCCR4, TSnCCR5	Reload	Possible	Possible to select as A/D trigger
TSnDTC0, TSnDTC1	Reload	Conditionally possible <sup>Note</sup>	Dead time
TSnPAT0, TSnPAT1	Reload	Possible	Special pattern setting

**Note** Refer to “Rewriting of TSnDTC0 and TSnDTC1 registers” on page 827 for details.

**(b) Input pin**

Pin	Function
TTRGSn	-
TEVTSn	-
TAPTSn2 to TAPTSn0	Pattern input
TSnSTCI0, TSnSTCI1 signal	Trigger input
TSnARD signal	Pattern change order input

**(c) Output pin**

Pin	Function
TOSn0	Toggle output by TSnCCR0 compare match
TOSn1	Special pattern output (with dead time) by TSnCCR1 to TSnCCR3 compare match and TSnPAT0
TOSn2	Special pattern output (with dead time) by TSnCCR1 to TSnCCR3 compare match and TSnPAT1
TOSn3	Special pattern output (with dead time) by TSnCCR1 to TSnCCR3 compare match and TSnPAT0
TOSn4	Special pattern output (with dead time) by TSnCCR1 to TSnCCR3 compare match and TSnPAT1
TOSn5	Special pattern output (with dead time) by TSnCCR1 to TSnCCR3 compare match and TSnPAT0
TOSn6	Special pattern output (with dead time) by TSnCCR1 to TSnCCR3 compare match and TSnPAT1
TOSn7	Pulse output by A/D conversion trigger

**(d) Interrupt**

Interrupt	Function
INTTSnCC0 to INTTSnCC5	Compare match of INTTSnCC0 to INTTSnCC5 registers
INTTSnOV	-
INTTSnER	Error
INTTSnOD	-
INTTSnCD0	Peak interrupt (occurs at the same timing with INTTSnCC0)
INTTSnWN	Warning

**(e) Compare match timing**

Compare match	Timing
TSnCCR0	Timing of switching 16-bit counter from TSnCCR0 to 0000H
TSnCCR1 to TSnCCR5	Timing of switching 16-bit counter from “TSnCCR1 to TSnCCR5” to “TSnCCR1 to TSnCCR5 + 1”

**Note** “-” indicates an unused function in the special 120° excitation mode.



### (3) Setting of special 120° excitation mode

#### (a) Mode setting

The special 120° excitation mode is set by setting TSnMD3 to TSnMD0 of the TSnCTL1 register to [1, 0, 1, 1].

#### (b) Output level/output enable setting

Output level/output enabling is performed by setting the TSnOL0 to TSnOL7, TSnOE0 to TSnOE7 bits of the TSnLOC0 register.

Toggle output is performed from the TOSn0 pin with the cycle match (match between 16-bit counter and TSnCCR0 register).

The TOSn7 pin output is for A/D conversion. Enable the output if necessary.

#### (c) Error interrupt/warning interrupt occurrence enabling

Error interrupt (INTTSnER) occurrence is enabled by setting the TSnEOC bit of the TSnLOC4 register to 1 when positive/negative phase simultaneous active state is detected.

A warning interrupt (INTTSnWN) is enabled by setting TSnWOC of the TSnLOC4 register to 1.

#### (d) Interrupt and thinning out function setting

A peak interrupt (INTTSnCD0) occurs upon a match between the TSnCCR0 register and 16-bit counter. (Control of the TSnIOE bit of the TSnOPT1 register is invalid). Set TSnICE of the TSnOPT1 register to 1 to output a peak interrupt.

To use the thinning out function or peak interrupt, execute setting with the TSnID4 to TSnID0 bits of the TSnOPT1 register.

#### (e) Reload thinning out function setting

Set TSnRDE of the TSnOPT1 register to 1 to set reload timing to the same timing with the interrupt timing.

#### (f) A/D conversion trigger output setting

To set A/D conversion trigger 0 (TSnADTRG0 signal), set whether to enable/disable upon match with the TSnCCR5 register, upon match with the TSnCCR4 register, and at peak interrupt (INTTSnCD0) by using the TSnAT04, TSnAT02, and the TSnAT01 bits of the TSnOPT2 register.

To set A/D conversion trigger 1 (TSnADTRG1 signal), set whether to enable/disable upon the match with the TSnCCR5 register, upon match with the TSnCCR4 register, and at peak interrupt (INTTSnCD0) by using the TSnAT14, TSnAT12, and TSnAT11 bits of the TSnOPT3 register.

Set the compare value setting for the TSnCCR4 and TSnCCR5 registers. Thinning out function is enabled for the TSnADTRG0 and TSnADTRG1 signals.

No thinning out, 1-thinning out, 3-thinning out and 7-thinning out can be set with the TSnACC01 and TSnACC00 bits of the TSnOPT2 register, and the TSnACC11 and TSnACC10 bits of the TSnOPT3 register.

- Caution**
1. When using the TOSn7 pin, set the TSnOPT2 and TSnOPT3 registers and the TSnCCR4 and TSnCCR5 registers appropriately according to the system.
  2. Trough interrupts (INTTSnOD) do not occur in the special 120° excitation mode. Be sure to set TSnAT00, 10 bits of TSnOPT2 and TSnOPT3 registers to 0.
  3. The 16-bit sub-counter does not operate in the special 120° excitation mode.  
Be sure to set TSnAT07, 06, TSnAT17, 16 bits of TSnOPT2, TSnOPT3 registers to 0.
  4. Count-down status of the 16-bit counter does not occur in the special 120° excitation mode. Be sure to set TSnAT05, 03, TSnAT15, 13 bits of TSnOPT2, TSnOPT3 registers to 0.

#### (g) Dead time setting

Dead time is set with the TSnDTC0 and TSnDTC1 registers.

Dead time can be calculated with the following equation.

16-bit counter operation clock cycle × TSnDTC0

16-bit counter operation clock cycle × TSnDTC1

The time for the change from the inactive state of the TOSn2, TOSn4, TOSn6 pins to the active state of the TOSn1, TOSn3, TOSn5 pins can be set with the TSnDTC0 register.

The time for the change from the inactive state of the TOSn1, TOSn3, TOSn5 pins to the active state of the TOSn2, TOSn4, TOSn6 pins can be set with TSnDTC1 register.

#### (h) Special pattern output setting

In the special 120° control, the output of the TOSn1, TOSn3, and TOSn5 pins is controlled with TSnCCR0 to TSnCCR3 registers, and the output of the TOSn2, TOSn4, and TOSn6 pins is controlled with TSnCCR0 to TSnCCR3, and TSnPAT1 registers. The setting enables special pattern output control.

#### (i) Pattern output switch timing, pattern output order, and initial output pattern setting

[Pattern switch method]

The pattern switch method is enabled by setting TSnPOT of the TSnOPT5 register to 0. Output of pins TOSn1 to TOSn6 is switched at the change timing of the TAPTSn2 to TAPTSn0 pins.

The output order at operation start is set with the TSnIDC bit of the TSnOPT7 register. The initial output pattern is set with the TSnPSC bit of the TSnOPT4 register. However, setting with the TSnPSC bit is invalid after the rotation direction is determined (any value is set in the TSnTSF bit of the TSnOPT5 register).

[Trigger switch method]

Trigger switch method will be enabled when setting TSnPOT of TSnOPT5 register to 1.

Output of the TOSn1 to TOSn6 pins is switched with the rise of the external input (TSnSTCI1, TSnSTCI0 signals).

Output order is set in combination of TSnARD signal, TSnPSC bit of the

TSnOPT4 register, or TSnIDC bit of TSnOPT7 register.

Selection of TSnARD signal or TSnPSC bit of the TSnOPT4 register shall be set with the TSnPSS bit of the TSnOPT4 register.

Output order is controlled with the TSnARD input signal when TSnPSS is set to 0, and controlled with the TSnPSC bit when TSnPSS is set to 1. Refer to *“Timer output pattern in each mode and function” on page 671* for the pattern output order.

Initial output pattern can be controlled with the TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register.

Initial pattern is output upon the timer Sn operation start setting (when TSnCE of TSnCTL0 register is set to 1) after setting with the TSnIPC2 to TSnIPC0 bit. Refer to *“Timer output initial pattern setting in each mode and function” on page 663* for details.

---

**Caution** Set the initial pattern by reading the input level of the port connected to the TAPTSn2 to TAPTSn0 pins.

---

#### (4) Operation in special 120° excitation mode

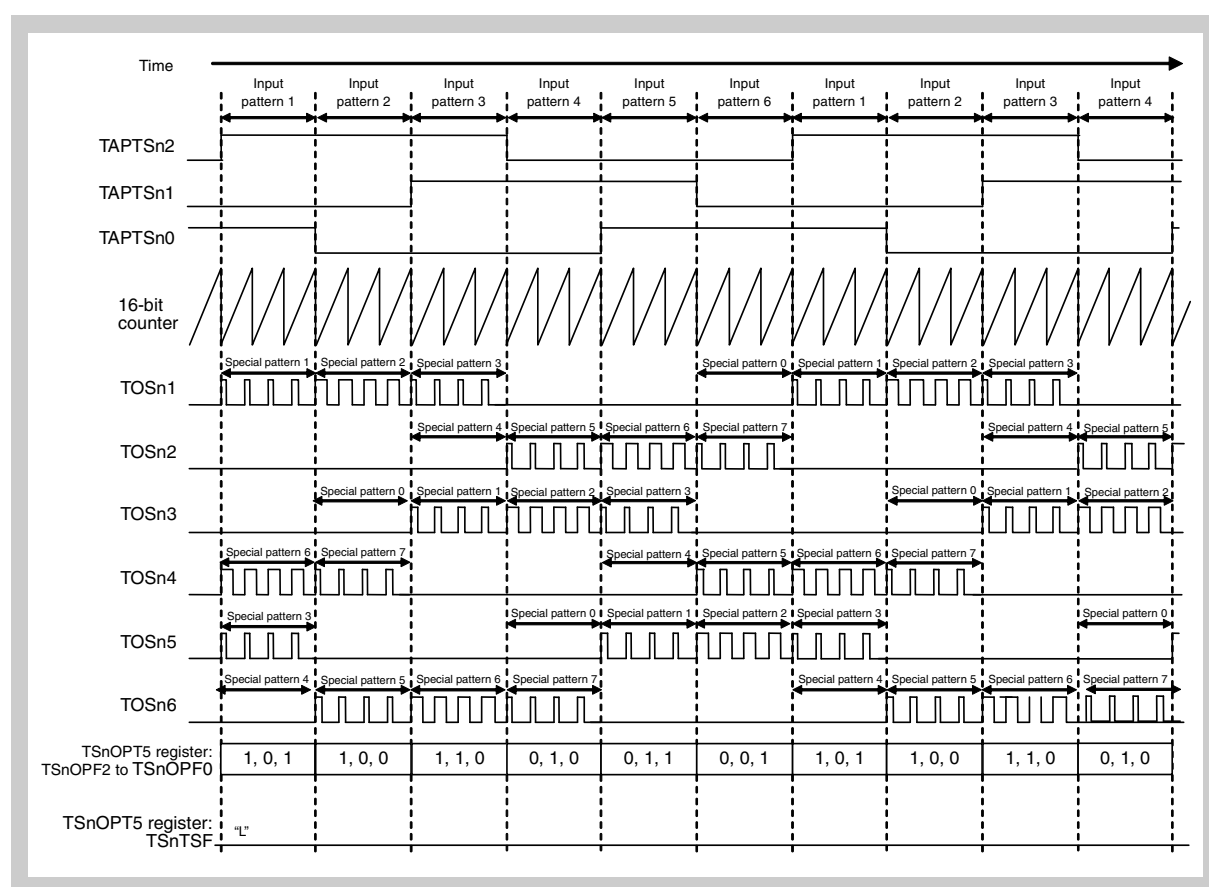
Figure 17-171 on page 904 and Figure 17-172 on page 905 show operation examples in the special 120° excitation mode. TOSn1 to TOSn6 detect input level change timing of the TAPTSn2 to TAPTSn0 pins to change the output pattern.

The 16-bit counter operation is a saw-tooth wave operation to output a special pattern set with the TSnCCR0 to TSnCCR3, TSnPAT0, and TSnPAT1 registers. The 16-bit counter is not cleared to 0000H until matched with PWM cycle TSnCCR0 register value even if the change of the TAPTSn2 to TAPTSn0 pins is detected. Special output pattern is switched from the following PWM cycle where the change of the TAPTSn2 to TAPTSn0 pins is detected.

TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,

TSnOPT5 register: TSnPOT = 0, TSnPSS = 1,

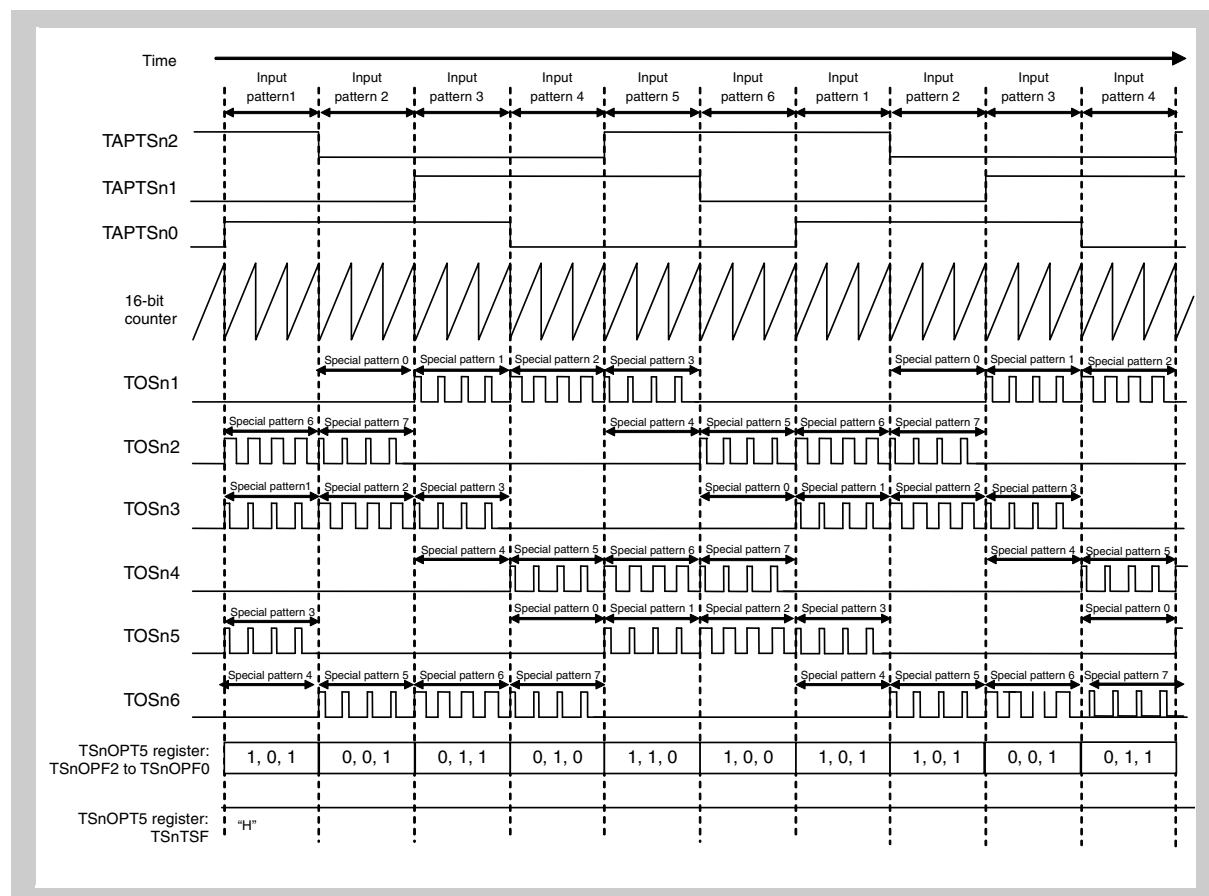
TSnOPT7 register: TSnIDC = 0



**Figure 17-171** Special 120° excitation mode basic operation timing example (pattern switch method, normal rotation)

**Note** Refer to "Description of special pattern of special 120° excitation mode" on page 909 for details on the special pattern.

TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,  
 TSnOPT5 register: TSnPOT = 0, TSnPSS = 1,  
 TSnOPT7 register: TSnIDC = 1



**Figure 17-172** Special 120° excitation mode basic operation timing example (pattern switch method, reverse rotation)

**Note** Refer to "Description of special pattern of special 120° excitation mode" on page 909 for details on the special pattern.

**(5) Output pattern list in special 120° excitation mode**

In the special 120° excitation mode, the output pattern is determined by rotation direction and the TSnIDC bit of the TSnOPT7 register.

TSnOPT5 register		Rotation direction
TSnPOT	TSnPSS	
0	-	TSnTSF flag
1	0	Input signal to TSnARD
1	1	TSnPSC bit

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,  
 TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
 TSnOPT7 register: TSnIDC = 0

→ Pattern switch order

Output pin	TSnOPT4 register TSnIPC2 to TSnIPC0 <sup>Note 1</sup> /TSnOPT5 register TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	Special pattern 1	Special pattern 2	Special pattern 3	INACT	INACT	Special pattern 0	Note 1	Note 2
TOSn2	INACT	INACT	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	Note 1	Note 2
TOSn3	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Special pattern 3	INACT	Note 1	Note 2
TOSn4	Special pattern 6	Special pattern 7	INACT	INACT	Special pattern 4	Special pattern 5	Note 1	Note 2
TOSn5	Special pattern 3	INACT	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Note 1	Note 2
TOSn6	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	INACT	INACT	Note 1	Note 2

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,  
 TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
 TSnOPT7 register: TSnIDC = 1

→ Pattern switch order

Output pin	TSnOPT4 register TSnIPC2 to TSnIPC0 <sup>Note 1</sup> /TSnOPT5 register TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Special pattern 3	Note 1	Note 2
TOSn2	Special pattern 5	Special pattern 6	Special pattern 7	INACT	INACT	Special pattern 4	Note 1	Note 2
TOSn3	Special pattern 2	Special pattern 3	INACT	INACT	Special pattern 0	Special pattern 1	Note 1	Note 2
TOSn4	INACT	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	INACT	Note 1	Note 2
TOSn5	Special pattern 0	Special pattern 1	Special pattern 2	Special pattern 3	INACT	INACT	Note 1	Note 2
TOSn6	Special pattern 7	INACT	INACT	Special pattern 4	Special pattern 5	Special pattern 6	Note 1	Note 2

- Note**
1. The output pattern of TSnIPC2 to TSnIPC0 is changed by writing when TSnPOT = 1. Then, output is switched in the pattern switch order when pattern switch trigger is generated by the rise of TSnSTCI0 and TSnSTCI1. In this case, TSnIPC2 to TSnIPC0 are not changed even if the output pattern is switched.
  2. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], the output remains the level before the change.
  3. Output remains the level before the change.

- Caution** Special pattern 3 and special pattern 4 cannot be active simultaneously. Special pattern 0 and special pattern 7 cannot be active simultaneously either. The simultaneous active setting results in the following operations.
1. When TOSn1, TOSn3 and TOSn5 are set active prior to TOSn2, TOSn4, and TOSn6  
Only TOSn1, TOSn3 and TOSn5 are active in the simultaneous active period.
  2. When TOSn2, TOSn4, and TOSn6 are set active prior to TOSn1, TOSn3 and TOSn5  
Only TOSn2, TOSn4, and TOSn6 are active in the simultaneous active period.
  3. When “TOSn1, TOSn3 and TOSn5” and “TOSn2, TOSn4, and TOSn6” are set active simultaneously  
Only TOSn1, TOSn3 and TOSn5 are active in the simultaneous active period.

- Note**
1. Special pattern 0 to special pattern 7 indicate the special pattern output by TSnCCR0 to TSnCCR3, TSnPAT0, and TSnPAT1 registers.
  2. INACT indicates the inactive level output.

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,  
TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
TSnOPT7 register: TSnIDC = 0

← Pattern switch order

Output pin	TSnOPT4 register TSnIPC2 to TSnIPC0 <sup>Note 1</sup> /TSnOPT5 register TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	Special pattern 2	Special pattern 1	Special pattern 0	INACT	INACT	Special pattern 3	Note 1	Note 2
TOSn2	INACT	INACT	Special pattern 7	Special pattern 6	Special pattern 5	Special pattern 4	Note 1	Note 2
TOSn3	INACT	Special pattern 3	Special pattern 2	Special pattern 1	Special pattern 0	INACT	Note 1	Note 2
TOSn4	Special pattern 5	Special pattern 4	INACT	INACT	Special pattern 7	Special pattern 6	Note 1	Note 2
TOSn5	Special pattern 0	INACT	INACT	Special pattern 3	Special pattern 2	Special pattern 1	Note 1	Note 2
TOSn6	Special pattern 7	Special pattern 6	Special pattern 5	Special pattern 4	INACT	INACT	Note 1	Note 2

Reverse rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 1,  
 TSnOPT5 register: TSnPOT = 1, TSnPSS = 1,  
 TSnOPT7 register: TSnIDC = 1

← Pattern switch order

Output pin	TSnOPT4 register TSnIPC2 to TSnIPC0 <sup>Note 1</sup> /TSnOPT5 register TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	Special pattern 3	Special pattern 2	Special pattern 1	Special pattern 0	Note 1	Note 2
TOSn2	Special pattern 6	Special pattern 5	Special pattern 4	INACT	INACT	Special pattern 7	Note 1	Note 2
TOSn3	Special pattern 1	Special pattern 0	INACT	INACT	Special pattern 3	Special pattern 2	Note 1	Note 2
TOSn4	INACT	Special pattern 7	Special pattern 6	Special pattern 5	Special pattern 4	INACT	Note 1	Note 2
TOSn5	Special pattern 3	Special pattern 2	Special pattern 1	Special pattern 0	INACT	INACT	Note 1	Note 2
TOSn6	Special pattern 4	INACT	INACT	Special pattern 7	Special pattern 6	Special pattern 5	Note 1	Note 2

- Note**
1. Output pattern of TSnIPC2 to TSnIPC0 is changed by writing when TSnPOT = 1.  
 Then, output is switched in the pattern switch order when a pattern switch trigger is generated by the rise of TSnSTCI0 and TSnSTCI1. In this case, TSnIPC2 to TSnIPC0 are not changed even if the output pattern is switched.
  2. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], the output remains the level before the change.
  3. Output remains the level before the change.

- Caution** Special pattern 3 and special pattern 4 cannot be set active. Special pattern 0 and special pattern 7 cannot be set active, either. The simultaneous active setting leads to the following operations.
1. When TOSn1, TOSn3 and TOSn5 are set active prior to TOSn2, TOSn4, and TOSn6  
 Only TOSn1, TOSn3 and TOSn5 are active in the simultaneous active period.
  2. When TOSn2, TOSn4, and TOSn6 are set active prior to TOSn1, TOSn3 and TOSn5  
 Only TOSn2, TOSn4, and TOSn6 are active in the simultaneous active period.
  3. When “TOSn1, TOSn3 and TOSn5” and “TOSn2, TOSn4, and TOSn6” are set active simultaneously  
 Only TOSn1, TOSn3 and TOSn5 are active in the simultaneous active period.

- Note**
1. Special patterns 0 to 7 indicate the special pattern output by the TSnCCR0 to TSnCCR3, TSnPAT0, and TSnPAT1 registers.
  2. INACT indicates the inactive level output.



**(6) Description of special pattern of special 120° excitation mode**

Special pattern is set with the TSnCCR0 to TSnCCR3 registers and the TSnPAT0 and TSnPAT1 registers.

The functions of the special pattern registers (TSnPAT0, TSnPAT1) are classified in 4-bit units. A special pattern is set with each 4-bit and the 8 special patterns can be executed as follows.

- Special pattern 0: Output pattern from TSnCCR0 to TSnCCR3 registers and TSnPAT0[3:0]
- Special pattern 1: Output pattern from TSnCCR0 to TSnCCR3 registers and TSnPAT0[7:4]
- Special pattern 2: Output pattern from TSnCCR0 to TSnCCR3 registers and TSnPAT0[11:8]
- Special pattern 3: Output pattern from TSnCCR0 to TSnCCR3 registers and TSnPAT0[15:12]
- Special pattern 4: Output pattern from TSnCCR0 to TSnCCR3 registers and TSnPAT1[3:0]
- Special pattern 5: Output pattern from TSnCCR0 to TSnCCR3 registers and TSnPAT1[7:4]
- Special pattern 6: Output pattern from TSnCCR0 to TSnCCR3 registers and TSnPAT1[11:8]
- Special pattern 7: Output pattern from TSnCCR0 to TSnCCR3 registers and TSnPAT1[15:12]

In the special 120° excitation mode, output of 4 patterns can be set in the positive phase with TSnPAT0 and output of 4 patterns settable in negative phase with TSnPAT1 are possible.

In the special 120° excitation mode, priority is as follows when the same value is set to the TSnCCR0 to TSnCCR3 registers.

Output setting after TSnCCR3 match > output setting after TSnCCR2 match > output setting after TSnCCR1 match > output setting after TSnCCR0 match

Priority of every special pattern is as follows.

- Special pattern 0: TSnPAT0[3] > TSnPAT0[2] > TSnPAT0[1] > TSnPAT0[0]
- Special pattern 1: TSnPAT0[7] > TSnPAT0[6] > TSnPAT0[5] > TSnPAT0[4]
- Special pattern 2: TSnPAT0[11] > TSnPAT0[10] > TSnPAT0[9] > TSnPAT0[8]
- Special pattern 3: TSnPAT0[15] > TSnPAT0[14] > TSnPAT0[13] > TSnPAT0[12]
- Special pattern 4: TSnPAT1[3] > TSnPAT1[2] > TSnPAT1[1] > TSnPAT1[0]
- Special pattern 5: TSnPAT1[7] > TSnPAT1[6] > TSnPAT1[5] > TSnPAT1[4]
- Special pattern 6: TSnPAT1[11] > TSnPAT1[10] > TSnPAT1[9] > TSnPAT1[8]
- Special pattern 7: TSnPAT1[15] > TSnPAT1[14] > TSnPAT1[13] > TSnPAT1[12]

Setting and output examples are shown below.

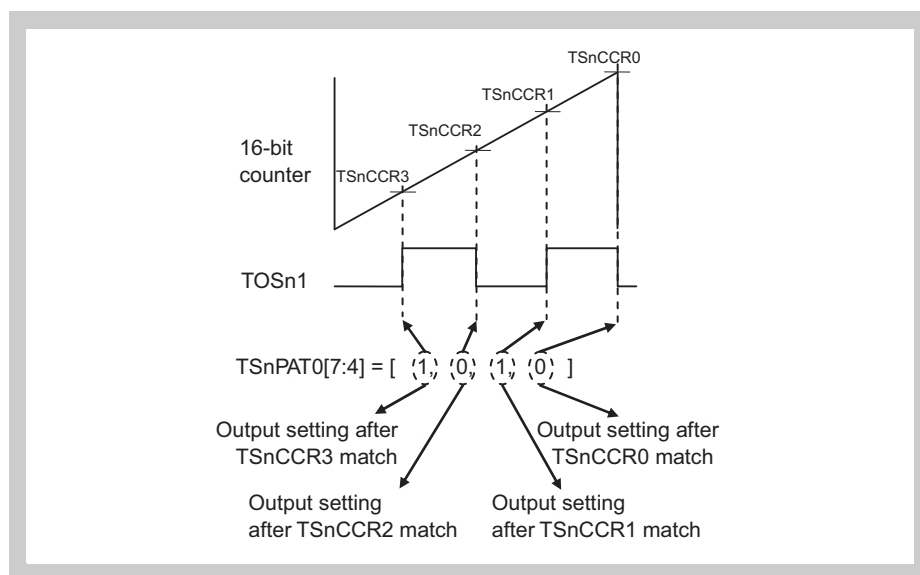


Figure 17-173 Special pattern setting example (1/2)

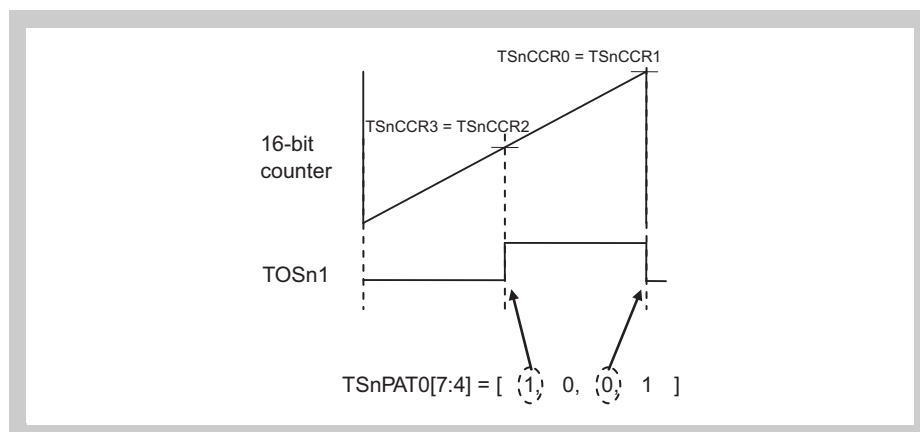


Figure 17-174 Special pattern setting example (2/2)

If the compare setting values of TSnPAT0[7] and TSnPAT0[6] are the same, TSnPAT0[7] has higher priority according to the priority order (TSnPAT0[7] > TSnPAT0[6] > TSnPAT0[5] > TSnPAT0[4]).

This applies to TSnPAT0[5] and TSnPAT0[4] as well.

**(7) Output switch timing in special 120° excitation mode**

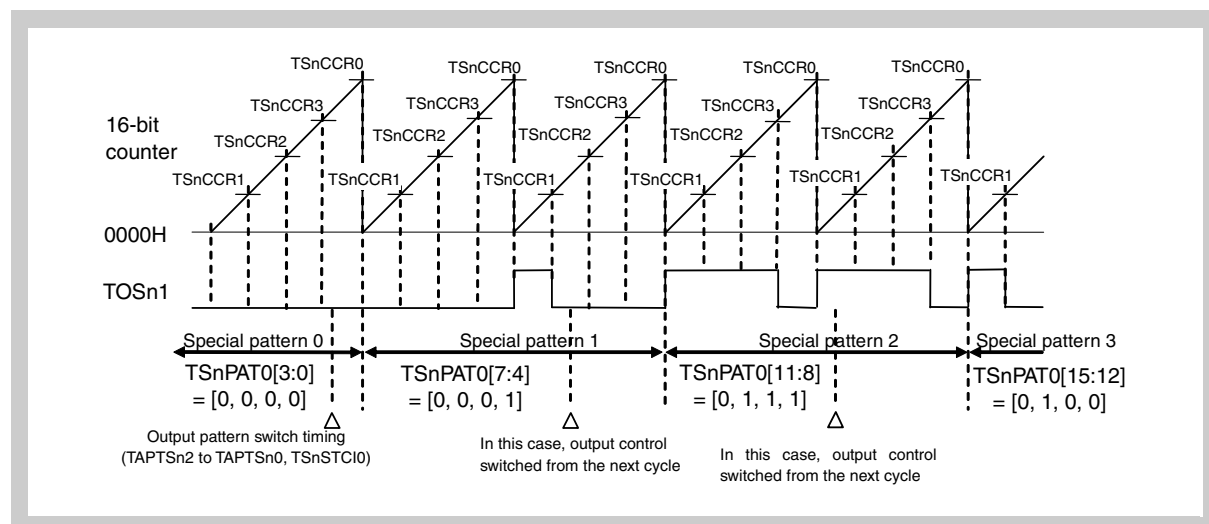
In the special 120° excitation mode, external output pattern switch timing (TSnSTCI0 and TSnSTCI1 signals, TAPTSn2 to TAPTSn0 pins) is input regardless of the 16-bit counter operation, as shown in Figure 17-175 on page 911, Figure 17-176 on page 912, Figure 17-177 on page 912, and Figure 17-178 on page 913. The output switch timing starts from the following cycle of external input switch timing. Therefore, special pattern output will be held until the current cycle is completed even if the switch timing is input from the outside.

When the TAPTSn2 to TAPTSn0 pins are changed serially within the first cycle in the pattern switch method, the edge-detection value just before the cycle match is used and reflected to the special pattern output from the following cycle.

If the TSnSTCI0 and TSnSTCI1 signal triggers are input several times within a cycle in the trigger switch method, the pattern is switched to adjacent output pattern due to only once acceptance.

Rewriting of TSnIPC2 to TSnIPC0 is performed prior to the TSnSTCI0, TSnSTCI1 trigger if both occur within a cycle in the trigger switch method.

The value rewritten just before cycle match is reflected if TSnIPC2 to TSnIPC0 are rewritten several times in a cycle.



**Figure 17-175** Output switch example (TAPTSn2 to TAPTSn0 pins, TSnSTCI0, TSnSTCI1 signal trigger input)

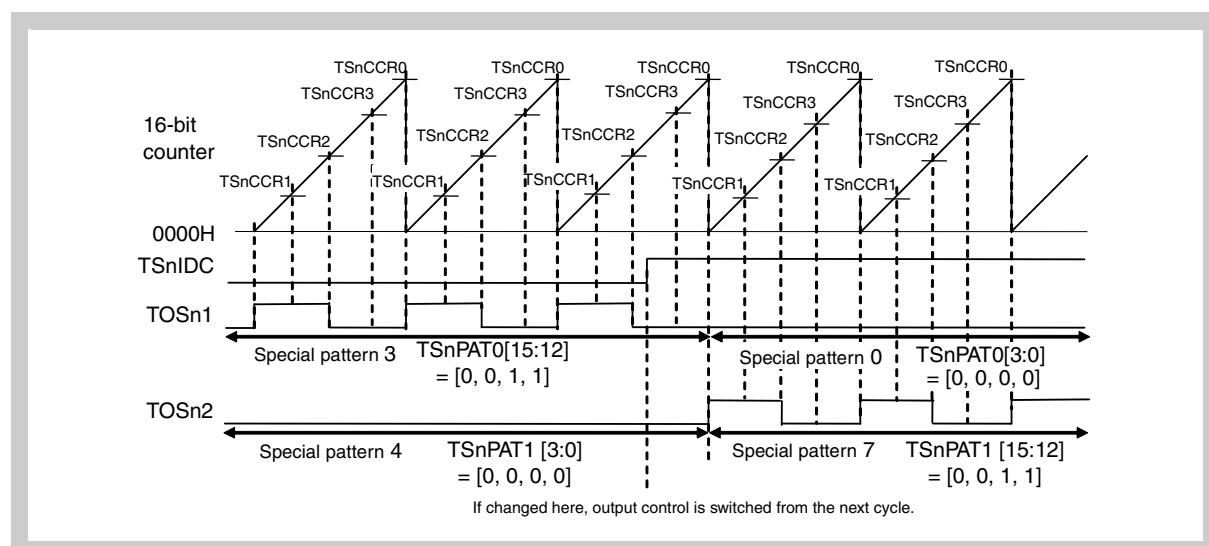


Figure 17-176 Output switch example (switch with TSnIDC bit of TSnOPT7 register)

(TSnOPT5 register: TSnPOT bit = 1)

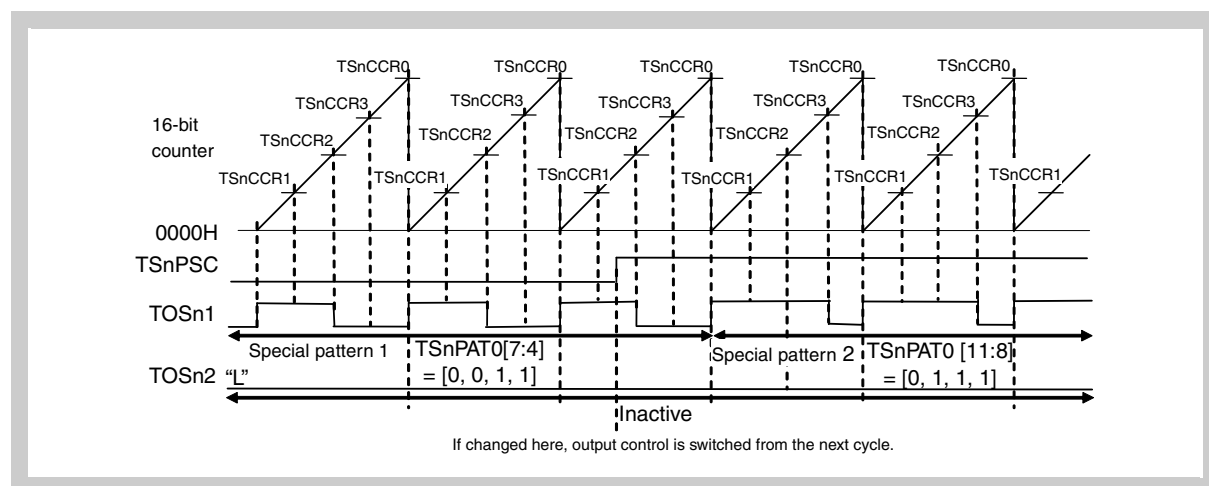


Figure 17-177 Output switch example (switch with TSnPSC bit of TSnOPT4 register)

TSnOPT4 register: TSnPSC = 0, TSnOPT5 register:  
TSnPOT = 1, TSnPSS = 1

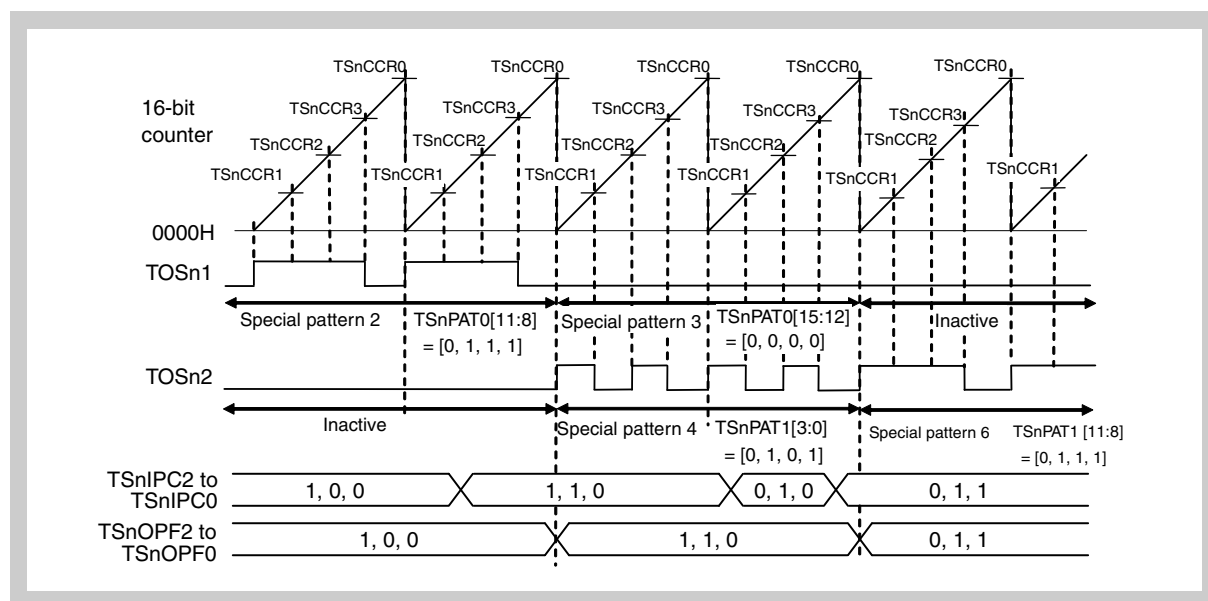


Figure 17-178 Output switch example (switch with TSnIPC2 to TSnIPC0 bits of TSnOPT4 register)

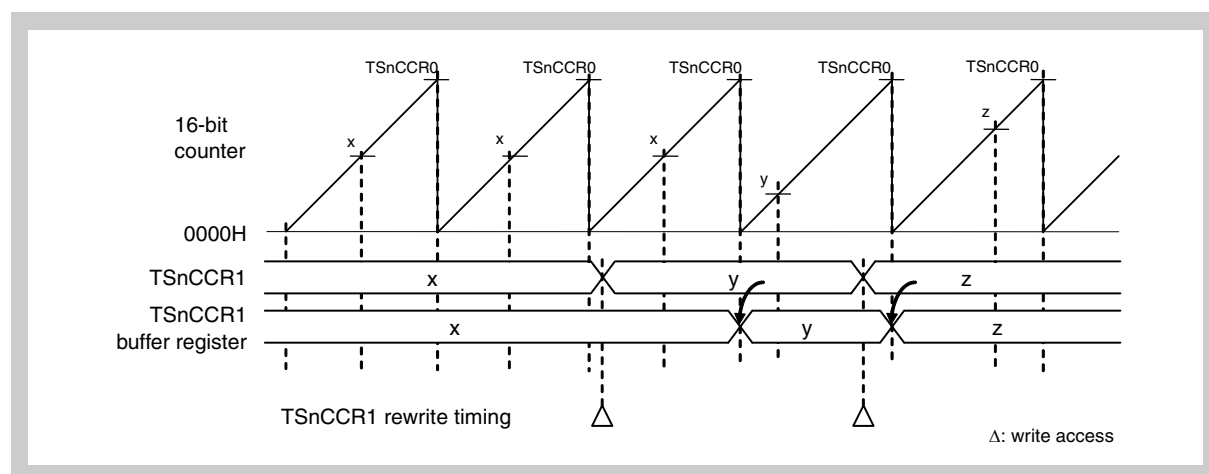
**(8) Compare register rewrite timing in special 120° excitation mode**

An operation example when the TSnCCR1 register is reloaded (batch rewriting) is described below.

Figure 17-179 on page 914 shows an output example in case of rewriting of the TSnCCR1 register.

After change of the TSnCCR1 register, data is not transferred to the TSnCCR1 buffer register until the next reload timing (the changed data will not be valid), so that the output waveform can be obtained as set. However, rewriting the TSnCCR1 register again is prohibited during the reload pending (in the period from changing the TSnCCR1 register to the batch rewriting).

Be sure to read the reload request flag (TSnRSF) to confirm 0 status when writing to the TSnCCR1 register.

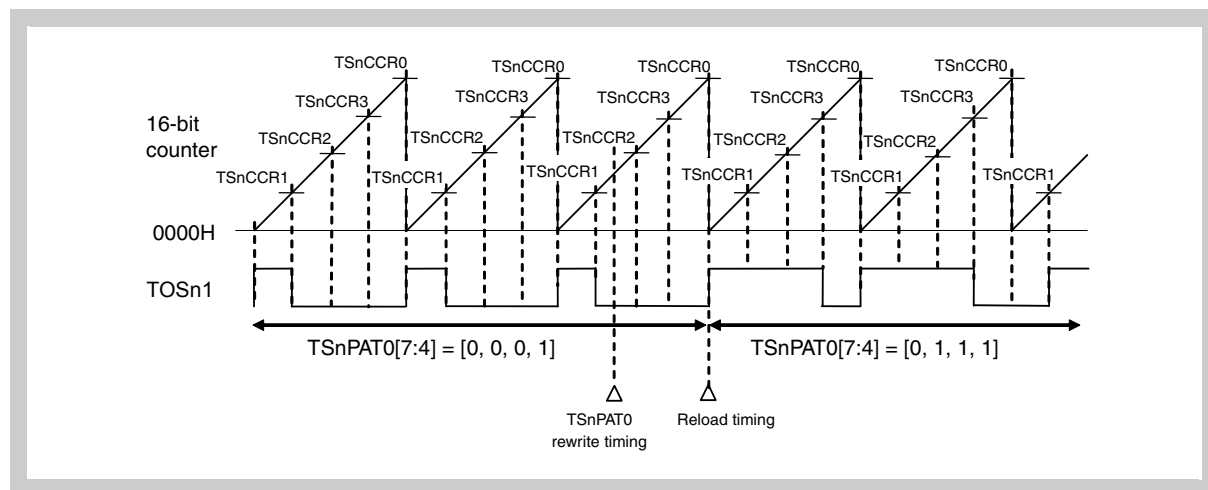


**Figure 17-179** Output example in case of rewriting of TSnCCR1 register

**(9) Pattern rewrite timing in special 120° excitation mode**

The operation example in case of rewriting of the TSnPAT0 register is described below.

TSnPAT0 register setting value is transferred to the buffer register at reload timing after rewriting (at TSnCCR0 register match timing after writing to TSnCCR1 register). The setting is valid at the timing when the 16-bit counter is cleared. Then, the expected pattern can be output.



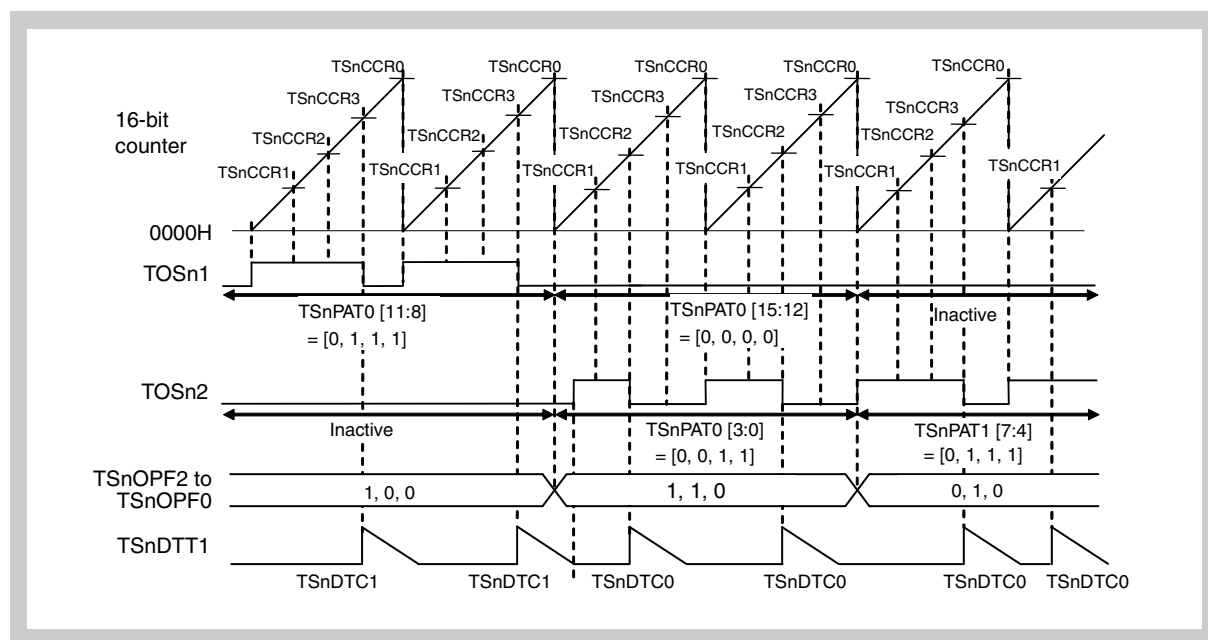
**Figure 17-180 Pattern rewrite timing example**

### (10) Dead time control in special 120° excitation mode

In the special 120° excitation mode, dead time is added by the dead time control operation at the fall of each phase.

The dead time that is set to the TSnDTC1 register is inserted at the fall of the positive phase, and the dead time that is set to the TSnDTC0 register is inserted at the fall of the negative phase.

TSnOPT4 register: TSnSOC = 0, TSnOPT5 register: TSnPOT = 1



**Figure 17-181** Dead time control example

### Caution

Dead time control has little effect on the timer output during normal operation. However, dead time control may affect the timer output under the following conditions.

- When noise occurs in input pattern during use of the pattern switch method
- When the input pattern changes earlier than PWM cycle during use of the pattern switch method
- When output pattern is forcibly changed by changing the TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register in the trigger switch method
- When the switch method is changed
- When the current direction control bit (TSnIDC bit of TSnOPT7 register) is changed
- When software output control function is used
- When 180° excitation control is used
- When special pattern 3 and special pattern 4 are set active simultaneously
- When special pattern 0 and special pattern 7 are set active simultaneously



**(11) Output switch in special 120° excitation mode**

In the special 120° excitation mode, the output pattern can be controlled by writing the value to the TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register when the trigger switch method is used (TSnPOT of TSnOPT5 register is set to 1) and also the TSnADC bit of the TSnOPT5 register is set to 0. Dead time is secured by using hardware at the switch timing.

- 
- Caution**
1. The output level of TOSn1 to TOSn6 pins is kept when [1, 1, 1] or [0, 0, 0] is written to TSnIPC2 to TSnIPC0.
  2. Do not rewrite more than once the TSnIPC2 to TSnIPC0 bits in a PWM cycle. If rewritten twice or more, the value just before PWM cycle match occurrence is reflected.
- 

**(12) Software output control function in special 120° excitation mode**

In the special 120° excitation mode, timer output control can be executed with software control by using the TSnSOC, TSnIPC2, TSnIPC1, and TSnIPC0 bits of the TSnOPT4 register, and the TSnIDC bit of the TSnOPT7 register.

Refer to “*Software output control function in 120° excitation mode*” on page 882 for the operation and setting procedure.

Refer to “*Software output function*” on page 936 for details on the software output control function.

**(13) 180° excitation control function in special 120° excitation mode**

In the special 120° excitation mode, 180° excitation control function is enabled by using the TSnADC, TSnIPC2, TSnIPC1, and TSnIPC0 bits of the TSnOPT5 register, and the TSnIDC bit of the TSnOPT7 register.

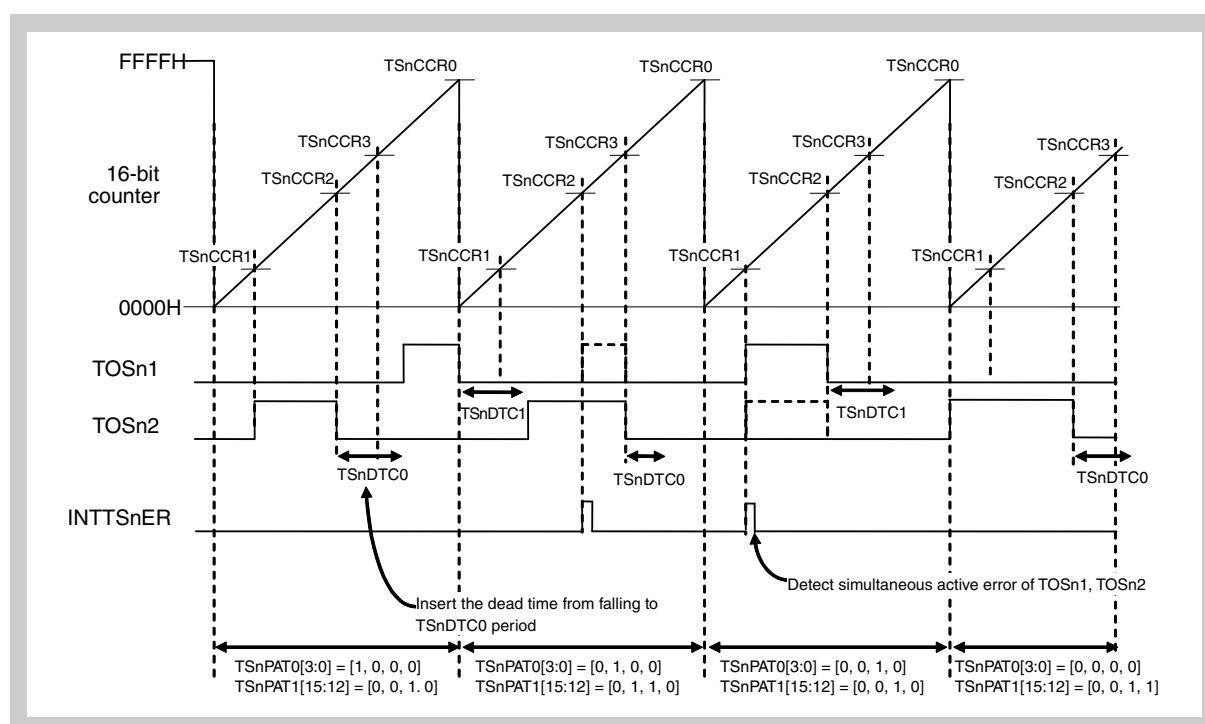
Refer to “*180° excitation control function in 120° excitation mode*” on page 885 for the operation and setting procedure.

Refer to “*180° excitation control function*” on page 938 for details on the 180° excitation control.

**(14) Cautions about setting of TSnPAT0, TSnPAT1 registers in special 120° excitation mode**

The setting where the simultaneous active level is output to positive and negative phases is possible in the special 120° excitation mode. Practically, this means the case when bits 3 to 0 of the TSnPAT0 register and bits 15 to 12 of the TSnPAT1 register are set to any value other than [0, 0, 0, 0], or the case when bits 15 to 12 of the TSnPAT0 register and bits 3 to 0 of the TSnPAT1 register are set to any value other than [0, 0, 0, 0]. However, the following circuit operation is used because it is risky to be simultaneous active level of positive and negative phases, taking the inverter control into account.

Figure 17-182 on page 918 shows an example in case of the setting to make the TOSn1 pin and TOSn2 pin active simultaneously. In this case, the TOSn1 pin is set by priority at the TOSn1, TOSn2 pins simultaneous rise timing. The TOSn3 pin is set by priority between the TOSn3 and TOSn4 pins, and the TOSn5 pin is set by priority between the TOSn5 and TOSn6 pins.



**Figure 17-182 Output Control at Simultaneous Active Occurrence**

**(15) Difference between 120° excitation mode and special 120° excitation mode**

The description of difference between the 120° excitation mode and the special 120° excitation mode are described below.

**(a) Timer output difference**

120° excitation mode

Positive phase: PWM1, PWM2 output  
Negative phase: PWM3, PWM4 output

Special 120° excitation mode

Positive phase: Special pattern 0, special pattern 1, special pattern 2, special pattern 3  
Negative phase: Special pattern 4 special pattern 5, special pattern 6, special pattern 7

**(b) Difference of electric angle that can be excited**

120° range of electric angle can be output during the PWM output period in the 120° excitation mode.

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnOPT5 register:  
TSnPOT = 0  
TSnOPT7 register: TSnIDC = 0

→ Pattern switch order

Output pin	TSnOPT4 register TSnIPC2 to TSnIPC0 <sup>Note 1</sup> /TSnOPT5 register TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	PWM1	PWM2	INACT	INACT	INACT	INACT	Note 2	Note 3
TOSn2	INACT	INACT	INACT	PWM3	PWM4	INACT	Note 2	Note 3
TOSn3	INACT	INACT	PWM1	PWM2	INACT	INACT	Note 2	Note 3
TOSn4	PWM4	INACT	INACT	INACT	INACT	PWM3	Note 2	Note 3
TOSn5	INACT	INACT	INACT	INACT	PWM1	PWM2	Note 2	Note 3
TOSn6	INACT	PWM3	PWM4	INACT	INACT	INACT	Note 2	Note 3

In the special 120° excitation mode, output range can be increased by 60° forward/backward respectively for the electric angle with 120° excitation control output.

Normal rotation: TSnOPT4 register: TSnSOC = 0, TSnPSC = 0,  
 TSnOPT5 register: TSnPOT = 0  
 TSnOPT7 register: TSnIDC = 0

→ Pattern switch order

Output pin	TSnOPT4 register TSnIPC2 to TSnIPC0 <sup>Note 1</sup> /TSnOPT5 register TSnOPF2 to TSnOPF0							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	Special pattern 1	Special pattern 2	Special pattern 3	INACT	INACT	Special pattern 0	Note 1	Note 2
TOSn2	INACT	INACT	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	Note 1	Note 2
TOSn3	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Special pattern 3	INACT	Note 1	Note 2
TOSn4	Special pattern 6	Special pattern 7	INACT	INACT	Special pattern 4	Special pattern 5	Note 1	Note 2
TOSn5	Special pattern 3	INACT	INACT	Special pattern 0	Special pattern 1	Special pattern 2	Note 1	Note 2
TOSn6	Special pattern 4	Special pattern 5	Special pattern 6	Special pattern 7	INACT	INACT	Note 1	Note 2

- Note**
1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], the output remains the level before the change.
  2. Output remains the level before the change.

- Caution** Special pattern 3 and special pattern 4 cannot be active simultaneously. Special pattern 0 and special pattern 7 cannot be active simultaneously, either. The simultaneous active setting leads to the following operations.
1. When TOSn1, TOSn3 and TOSn5 are set active prior to TOSn2, TOSn4, and TOSn6  
 Only TOSn1, TOSn3 and TOSn5 are active in the simultaneous active period.
  2. When TOSn2, TOSn4, and TOSn6 are set active prior to TOSn1, TOSn3 and TOSn5  
 Only TOSn2, TOSn4, and TOSn6 are active in the simultaneous active period.
  3. When “TOSn1, TOSn3 and TOSn5” and “TOSn2, TOSn4, and TOSn6” are set active simultaneously  
 Only TOSn1, TOSn3 and TOSn5 are active in the simultaneous active period.

- Note**
1. Each column indicates 60° of electric angle.
  2. PWM1 to PWM4 indicate PWM output by the TSnCCR1 to TSnCCR4 registers.
  3. Special pattern 0 to special pattern 7 indicate the special pattern output by the TSnCCR0 to TSnCCR3, TSnPAT0, and TSnPAT1 registers.
  4. INACT indicates the inactive level output.

### 17.10.12 Special pattern output mode

#### (1) Special pattern output mode overview

16-bit counter saw-tooth wave operation and four 16-bit compare registers are used in special pattern output mode. The value that is set by the TSnPAT0 and TSnPAT1 registers is output to the TOSn0 to TOSn7 pins upon a match between the 16-bit counter and TSnCCR0 to TSnCCR3 register values.

TSnPAT0[3:0]	TOSn0 pin output control
TSnPAT0[7:4]	TOSn1 pin output control
TSnPAT0[11:8]	TOSn2 pin output control
TSnPAT0[15:12]	TOSn3 pin output control
TSnPAT1[3:0]	TOSn4 pin output control
TSnPAT1[7:4]	TOSn5 pin output control
TSnPAT1[11:8]	TOSn6 pin output control
TSnPAT1[15:12]	TOSn7 pin output control

**Note** Special pattern output mode is valid when TSnMD3 to TSnMD0 of TSnCTL1 register is set to [1, 1, 0, 0].

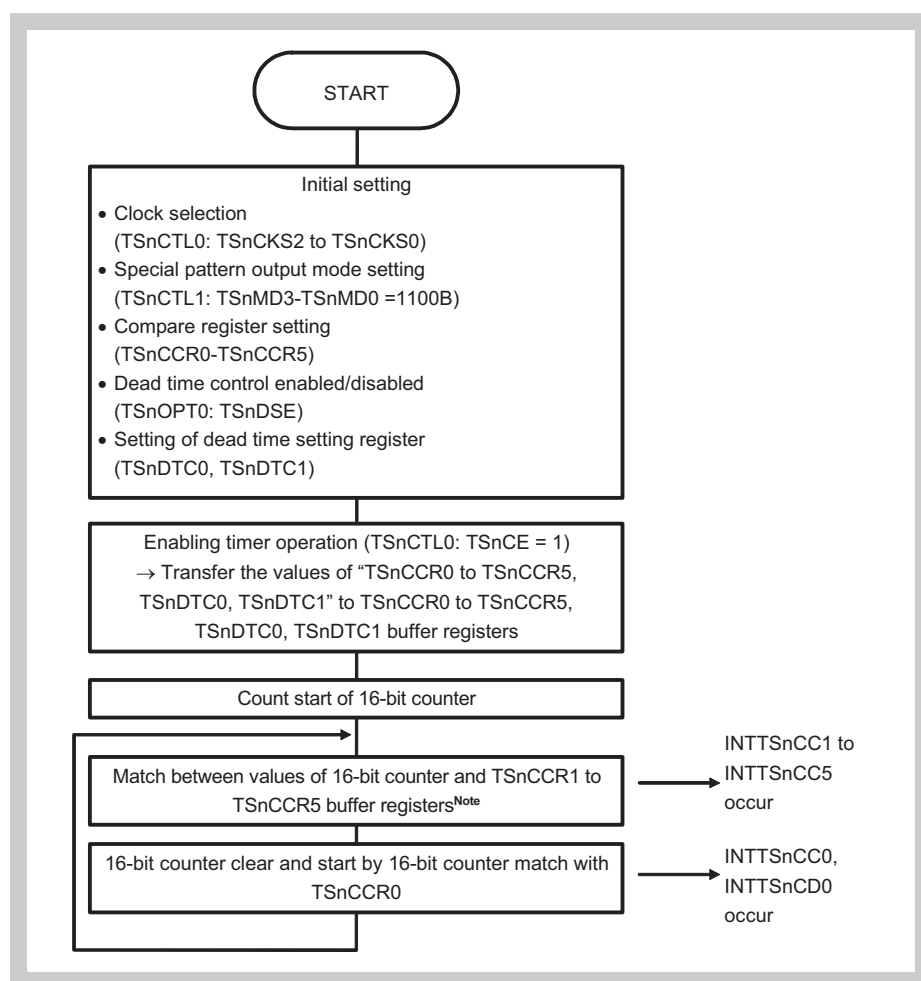


Figure 17-183 Basic operation flow in special pattern output mode

**Note** The 16-bit counter is not cleared upon a match between the values of the 16-bit counter and the TSnCCR1 to TSnCCR5 buffer registers.

**(2) Special pattern output mode operation list****(a) Register rewriting**

Register	Rewriting method	Rewriting during operation	Function
TSnCCR0	Reload	Possible	Cycle/compare value
TSnCCR1 to TSnCCR3	Reload	Possible	Compare value
TSnCCR4, TSnCCR5	Reload	Possible	Compare value (Selectable as A/D conversion trigger)
TSnDTC0, TSnDTC1	Reload	Conditionally possible <sup>Note</sup>	Compare value
TSnPAT0, TSnPAT1	Reload	Possible	Special pattern setting

**Note** Refer to “Rewriting of TSnDTC0 and TSnDTC1 registers” on page 827 for details.

**(b) Input pin**

Pin	Function
TTRGSn	-
TEVTSn	-

**(c) Output pin**

Pin	Function
TOSn0	Pattern output from TSnPAT0[3:0]
TOSn1	Pattern output from TSnPAT0[7:4]
TOSn2	Pattern output from TSnPAT0[11:8]
TOSn3	Pattern output from TSnPAT0[15:12]
TOSn4	Pattern output from TSnPAT1[3:0]
TOSn5	Pattern output from TSnPAT1[7:4]
TOSn6	Pattern output from TSnPAT1[11:8]
TOSn7	Pattern output from TSnPAT1[15:12]

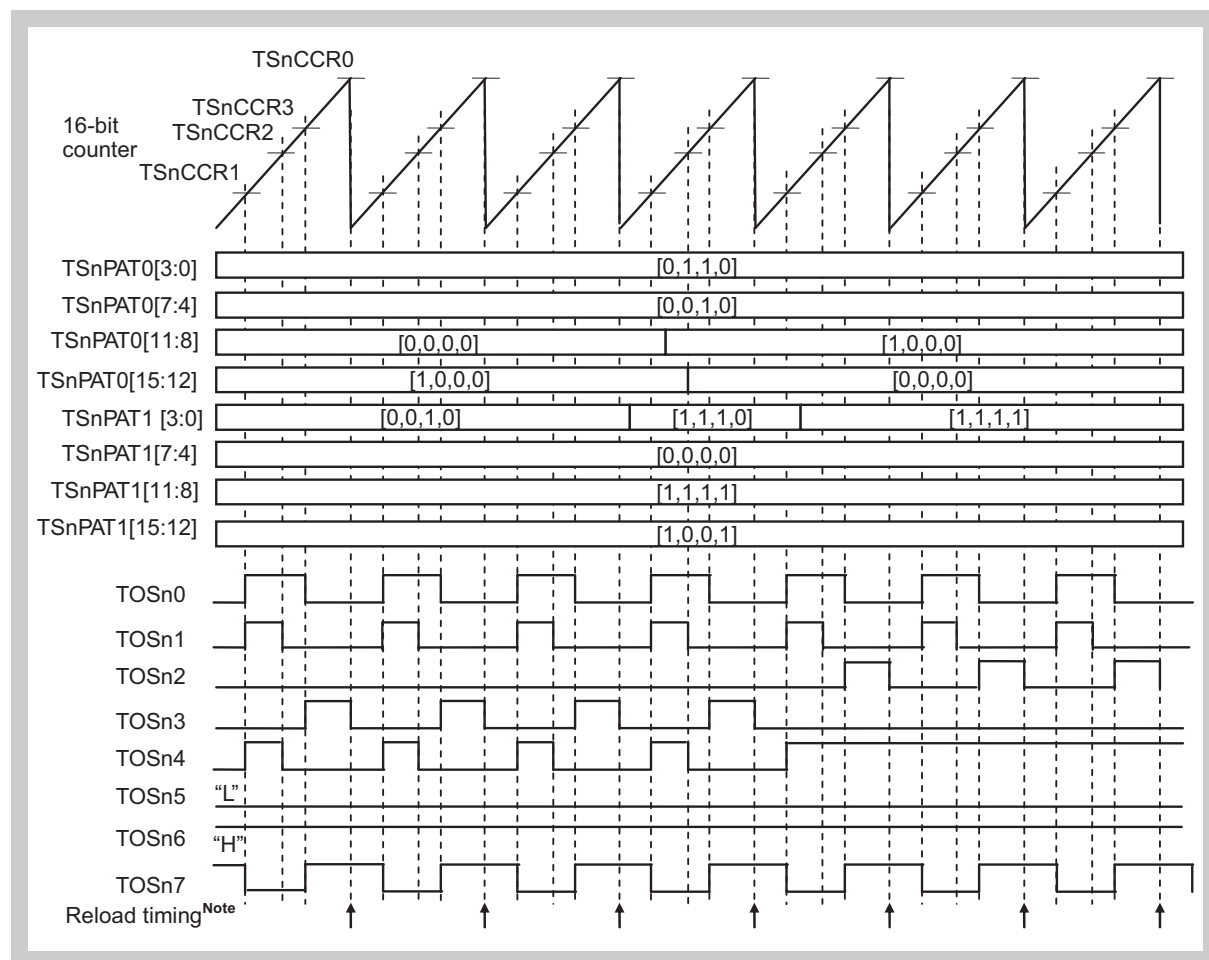
**(d) Interrupt**

Interrupt	Function
INTTSnCC0 to INTTSnCC5	Compare match of TSnCCR0 to TSnCCR5 registers
INTTSnOV	-
INTTSnER	Error
INTTSnOD	-
INTTSnCD0	Peak interrupt (occurs at the same timing as INTTSnCC0 register)
INTTSnWN	-

**(e) Compare match timing**

Compare match	Timing
TSnCCR0	Timing of switching 16-bit counter from TSnCCR0 to 0000H
TSnCCR1 to TSnCCR5	Timing of switching 16-bit counter from TSnCCR1 to TSnCCR5 to TSnCCR1 to TSnCCR5 +1

**Note** “–” indicates an unused function in the special pattern mode.



**Figure 17-184** Special pattern output mode operation example

**Note** Write access to the TSnCCR1 register is necessary.

### (3) Special pattern output mode setting

#### (a) Mode setting

The special pattern output mode is set by setting TSnMD4 to TSnMD0 of TSnCTL1 register to [1, 1, 0, 0].

#### (b) Output level/output enable setting

Output level/output enabling is performed by setting the TSnOL0 to TSnOL7 and TSnOE0 to TSnOE7 bits of the TSnIOC0 register. Pattern is output from the TOSn0 to TOSn7 pins.

#### (c) Error interrupt enabling

Error interrupt (INTTSnER) occurrence is enabled by setting the TSnEOC0 bit of the TSnIOC4 register to 1 when the positive/negative phase simultaneous active state is detected.

#### (d) Interrupt and thinning out function setting

Peak interrupt (INTTSnCD0) occurs by match between the TSnCCR0 register and 16-bit counter. (Trough interrupt does not occur). Set TSnICE of TSnOPT1 register to 1 if the peak interrupt is necessary. To use the thinning out function for peak interrupt, execute setting with the TSnID4 to TSnID0 bits of the TSnOPT1 register.

#### (e) Reload thinning out function setting

Set reload timing to the same timing as interrupt timing by setting the TSnRDE bit of the TSnOPT1 register to 1. Reload timing is generated by setting TSnRTE of TSnOPT1 register to 1.

---

**Caution** Write access to the TSnCCR1 register is required for reload execution.

---

#### (f) A/D conversion trigger output setting

To set A/D conversion trigger 0 (TSnADTRG0 signal), set enabling/disabling of match timing with the TSnCCR5 register, match timing with the TSnCCR4 register, and peak interrupt (INTTSnCD0) by using TSnAT04, TSnAT02, TSnAT01 bits of the TSnOPT2 register.

To set A/D conversion trigger 1 (TSnADTRG1 signal), set enabling/disabling of match timing with the TSnCCR5 register, match timing with the TSnCCR4 register, and peak interrupt (INTTSnCD0) by using the TSnAT14, TSnAT12, TSnAT11 bits of the TSnOPT3 register. Set the compare value of the TSnCCR4 and TSnCCR5 registers.

The thinning out function is enabled for the TSnADTRG0 and TSnADTRG1 signals. No thinning out, 1-thinning out, 3-thinning out and 7-thinning out can be set with the TSnACC01 and TSnACC00 bits of the TSnOPT2 register, and the TSnACC11 and TSnACC10 bits of the TSnOPT3 register.



- 
- Caution**
1. When using the TOSn7 pin, set the TSnOPT2 and TSnOPT3 registers and TSnCCR4 and TSnCCR5 registers appropriately according to the system.
  2. Trough interrupts (INTTSnOD) do not occur in special pattern output mode. Be sure to set TSnAT00, 10 bits of the TSnOPT2 and TSnOPT3 registers to 0.
  3. Do not use the peaks/troughs of the 16-bit sub-counter in special pattern output mode. Be sure to set TSnAT07, TSnAT06, TSnAT17, and TSnAT16 bits of the TSnOPT2 and TSnOPT3 registers to 0.
  4. Count-down status of the 16-bit counter does not occur in special pattern output mode. Be sure to set TSnAT05, TSnAT03, TSnAT15, TSnAT13 bits of the TSnOPT2 and TSnOPT3 registers to 0.
- 

### (g) Output pattern setting

In the output pattern, cycle setting is possible with the TSnCCR0 register, and duty setting with the TSnCCR1 to TSnCCR3 registers.

Timer output level after compare match (INTTSnCC0 to INTTSnCC3 occurrence) can be set with the TSnPAT0 and TSnPAT1 registers. In the special pattern output mode, the output pattern is controlled as follows.

TSnPAT0[3:0]:	TOSn0 pin output pattern setting
TSnPAT0[7:4]:	TOSn1 pin output pattern setting
TSnPAT0[11:8]:	TOSn2 pin output pattern setting
TSnPAT0[15:12]:	TOSn3 pin output pattern setting
TSnPAT1[3:0]:	TOSn4 pin output pattern setting
TSnPAT1[7:4]:	TOSn5 pin output pattern setting
TSnPAT1[11:8]:	TOSn6 pin output pattern setting
TSnPAT1[15:12]:	TOSn7 pin output pattern setting

### (h) Dead time setting

Dead time is set with the TSnDTC0 and TSnDTC1 registers.

Dead time can be calculated with the following equation.

16-bit counter operation clock cycle  $\times$  TSnDTC0

16-bit counter operation clock cycle  $\times$  TSnDTC1

The time for the change from the inactive state of TOSn2, TOSn4, TOSn6 pins to the active state of the TOSn1, TOSn3, TOSn5 pins can be set with the TSnDTC0 register.

The time for the change from the inactive state of TOSn1, TOSn3, TOSn5 pins to the active state of the TOSn2, TOSn4, TOSn6 pins can be set with the TSnDTC1 register.

- 
- Caution**
- TOSn1, TOSn3 and TOSn5 rise by priority in the setting of simultaneous rise of “TOSn1 and TOSn2”, “TOSn3 and TOSn4”, and “TOSn5 and TOSn6”.
-

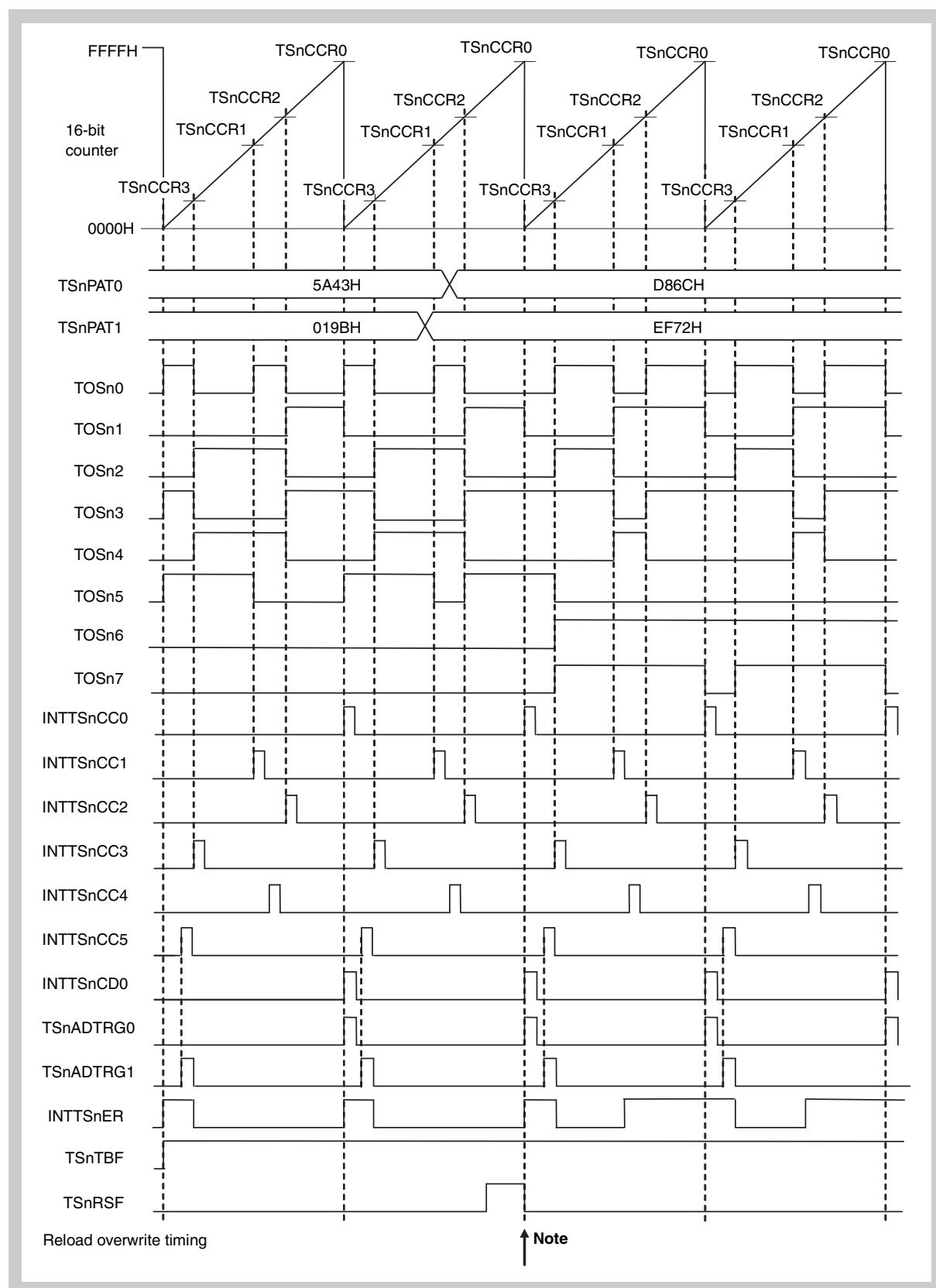
**(4) Special pattern output mode operation**

*Figure 17-185 on page 927 and Figure 17-186 on page 928* show an operation example in the special pattern output mode. The TSnPAT0 and TSnPAT1 register values are rewritten by using the reload mode (batch rewrite function) in *Figure 17-185 on page 927 and Figure 17-186 on page 928*. The output level is output as set in the TSnPAT0 and TSnPAT1 registers at match timing between the values of the TSnCCR0 to TSnCCR3 registers and the 16-bit counter.

Waveforms as expected are output because the batch rewrite is executed.

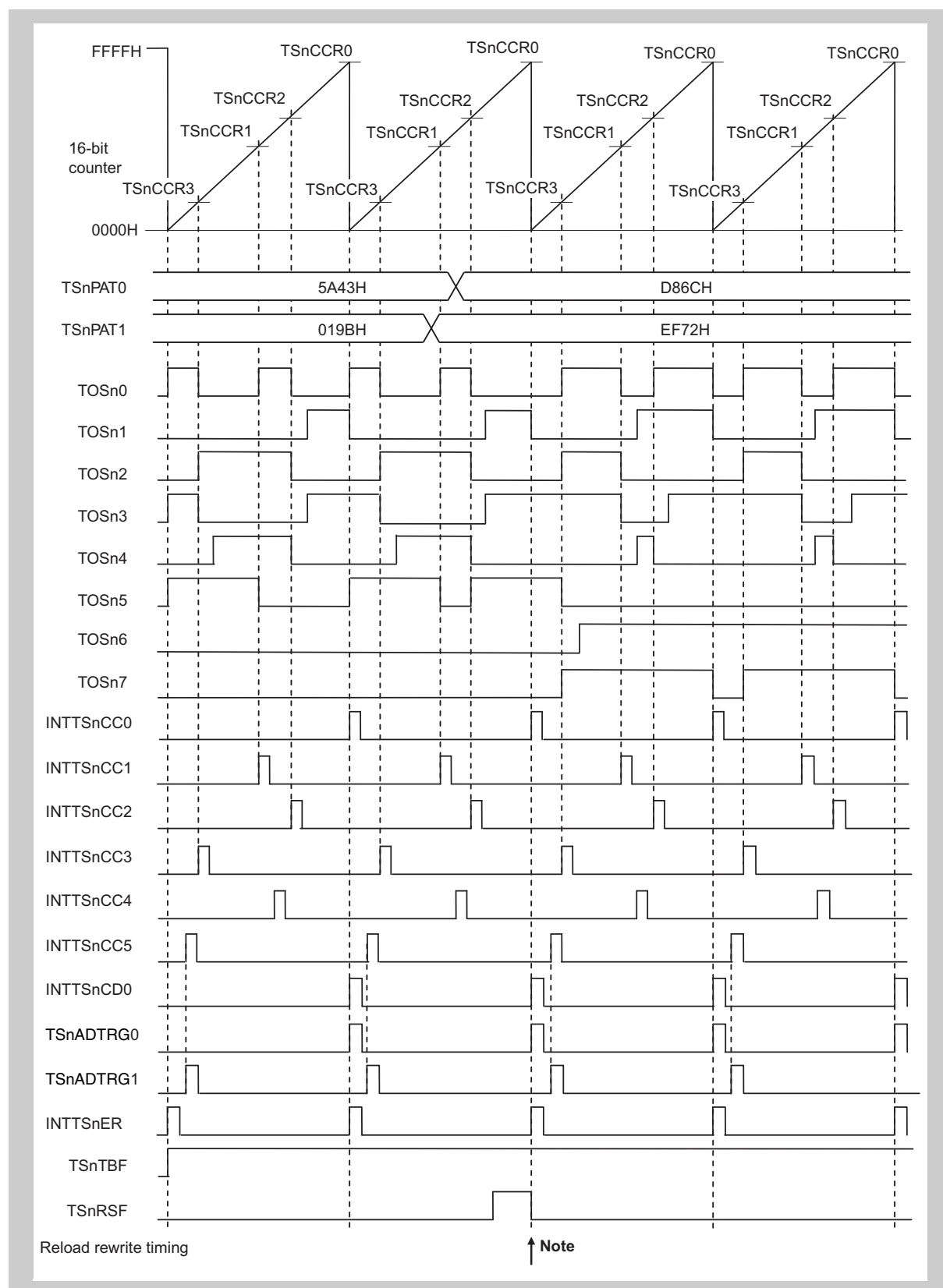
In the special pattern output mode, error interrupt (INTTSnER) occurs by setting TSnEOC of the TSnIOC4 register to 1 to detect the positive/negative simultaneous active state.

To generate a reload timing, write operation is necessary for the TSnCCR1 register. Therefore, write the same value to the TSnCCR1 register at last when rewriting the TSnPAT0 and TSnPAT1 registers.



**Figure 17-185** Special pattern output mode operation example (without dead time)

**Note** Write access to the TSnCCR1 register is required.



**Figure 17-186** Special pattern output mode operation example (with dead time)

**Note** Write access to the TSnCCR1 register is required.

**(5) Dead time control in special pattern output mode**

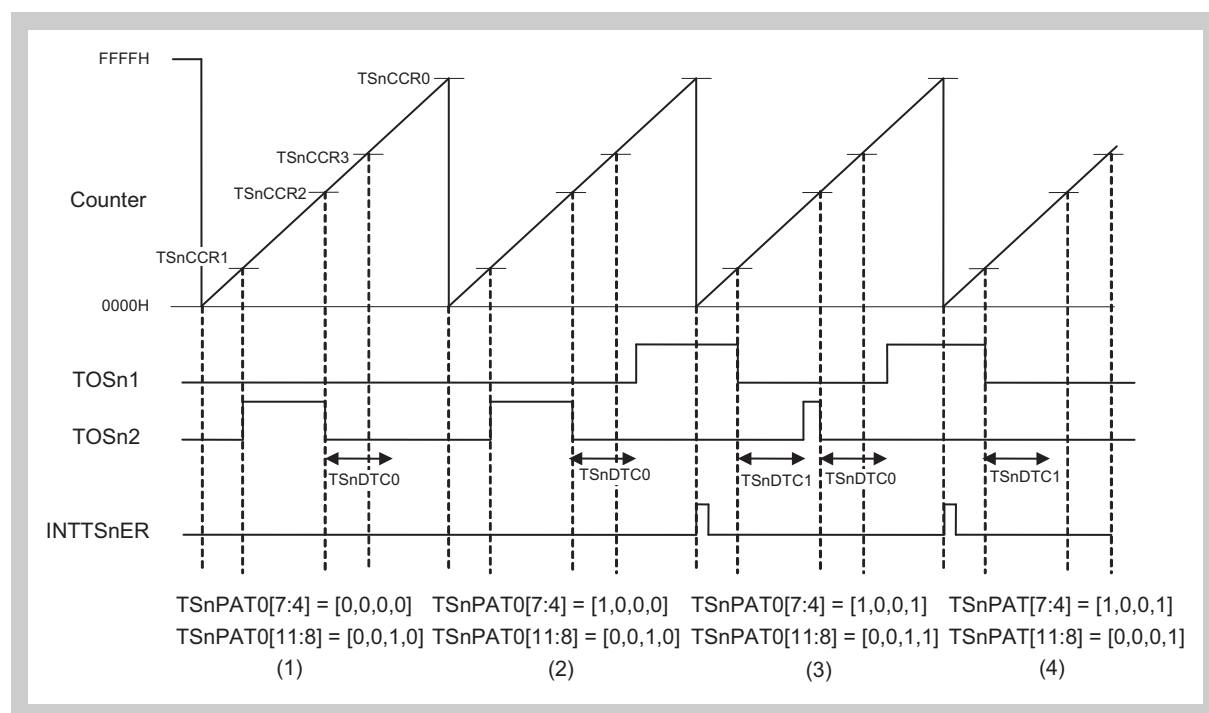
In the special pattern output mode, dead time is added by the dead time control operation at the fall of each phase when TSNDSE of the TSnOPT0 register is set to 1.

The dead time that is set to the TSNDTC1 register is inserted at the fall of the positive phase, and the dead time that is set to the TSNDTC0 register is inserted at the fall of the negative phase.

**Caution** Dead time control of the TOSn0 and TOS7 pins is impossible.

[Control of TOSn1, TOSn2 pins]

1. Dead time from the fall of the TOSn1 pin to the rise of the TOSn2 pin is assured.
2. The TOSn2 pin holds the inactive level when the TOSn1 pin is active level, and the TOSn1 pin holds the inactive level when the TOSn2 pin is active level.
3. Set the TOSn1 pin by priority if the TOSn1 and TOSn2 pins rise simultaneously.
4. The TOSn1 pin holds inactive level even if the TOSn1 pin setting condition is generated during the TOSn2 pin active level continued.



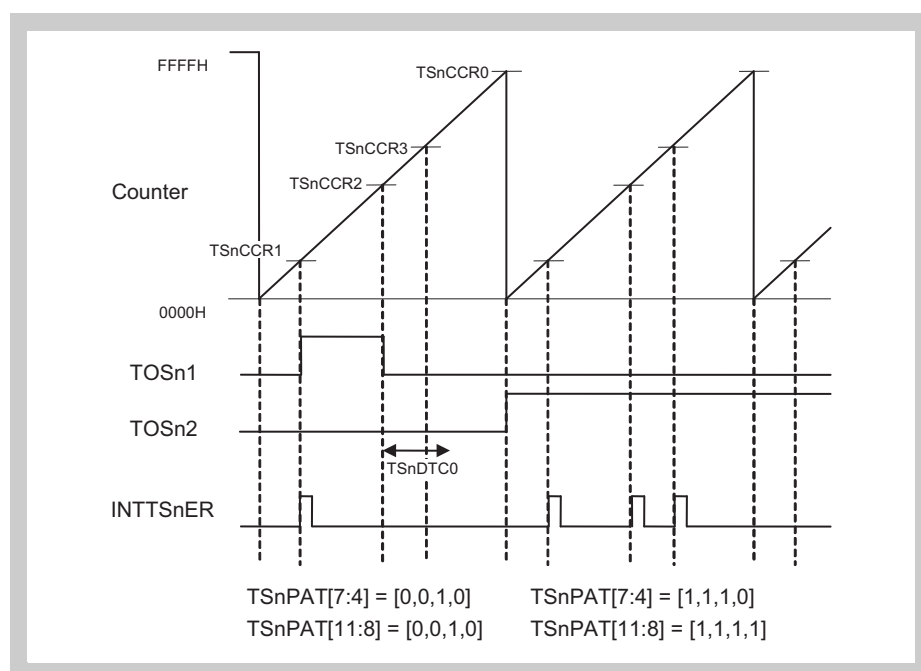
In cycle (1), dead time counter starts operation (with TSNDTC0 register value) from the fall of the TOSn2 pin, and controls the TOSn1 pin setting condition. In this example, there is nothing to occur because of no setting condition occurrence in the dead time period.

In cycle (2), dead time counter operates from the fall of the TOSn2 pin (with the TSNDTC0 register value). In this case, the TOSn1 pin setting condition is generated by the match with the TSnCCR3 register. However, the dead time counter is under operation, so the TOSn1 pin is set after the count completion of the dead time counter.

In cycle (3), setting conditions of the TOSn1 pin and TOSn2 pin are generated simultaneously at the start of the cycle. In this case, the TOSn1 pin is set by priority, the setting condition of TOSn2 pin is retained and an error interrupt (INTTSnER) occurs. The TOSn1 pin is cleared and the TOSn2 pin is set by the match between the TSnCCR1 register and counter. The TOSn2 pin is set after the dead time counter stops because dead time control is operated.

In cycle (4), the setting conditions of the TOSn1 pin and TOSn2 pin are generated simultaneously at the start of the cycle. In this case, the TOSn1 pin is set by priority, the setting condition of the TOSn2 pin is retained and an error interrupt (INTTSnER) occurs.

Clear conditions of the TOSn1 and TOSn2 pins are generated by match with the TSnCCR1 register, and then the TOSn1 pin is cleared. This time, the retained setting condition of TOSn2 pin is cleared.



When the condition in which TOSn1 and TOSn2 pins rise simultaneously, the TOSn1 pin becomes active. However, the TOSn1 pin retains the inactive level in case of the condition occurrence in which the TOSn1 and TOSn2 pins rise simultaneously during the active level output from the TOSn2 pin. Also, an error interrupt (INTTSnER) occurs at the same time.

#### [Control of TOSn3 and TOSn4 pins]

The control logic is same as the relation between TOSn1 and TOSn2. When the setting conditions for the TOSn3 and TOSn4 pins occur simultaneously, the TOSn3 pin is set by priority.

#### [Control of TOSn5 and TOSn6 pins]

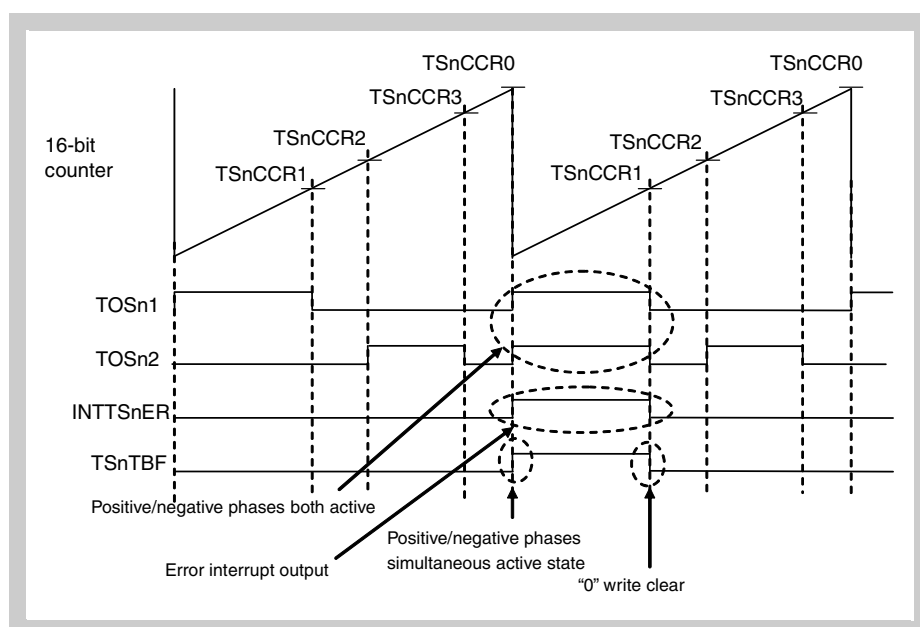
The control logic is the same as the relation between TOSn1 and TOSn2. When the setting conditions for the TOSn5 and TOSn6 pins occur simultaneously, the TOSn5 pin is set by priority.

**(6) Error interrupt in special pattern output mode (INTTSnER)**

In the special pattern output mode, the positive/negative phase simultaneous active state detection function can be used by setting the TSnTBA2 to TSnTBA0 bits of TSnIOC4 register. There are the TSnTBF bit of TSnOPT6 register, or INTTSnER for detection method. The INTTSnER operation differs according to the setting of the dead time control enabling/disabling (the TSnDSE bit of TSnOPT0 register).

**Table 17-39 Detection Method List**

TSnDSE of TSnOPT0 register	TSnTBF bit of TSnOPT6 register	INTTSnER
0 (Dead time control enabling)	Set with the timer output simultaneous active state	Error interrupt is output while active output is simultaneously provided.
1 (Dead time control disabling)	Set with the timer output simultaneous active state	Error interrupt is output at the timing when timer output simultaneous setting condition occurs.

**Figure 17-187 Error detection example (TSnDSE = 0) (1/2)**

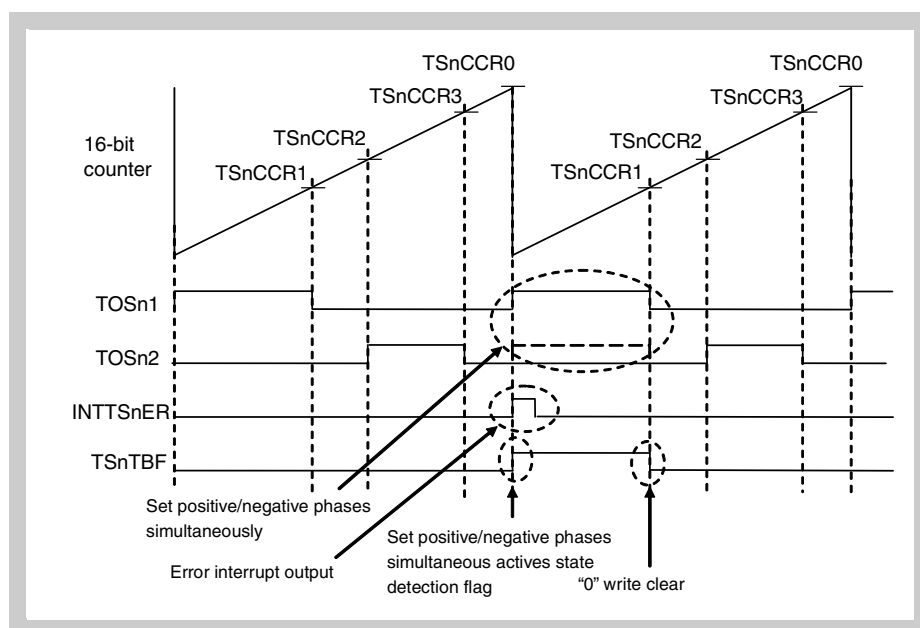


Figure 17-188 Error detection example (TSnDSE = 1) (2/2)

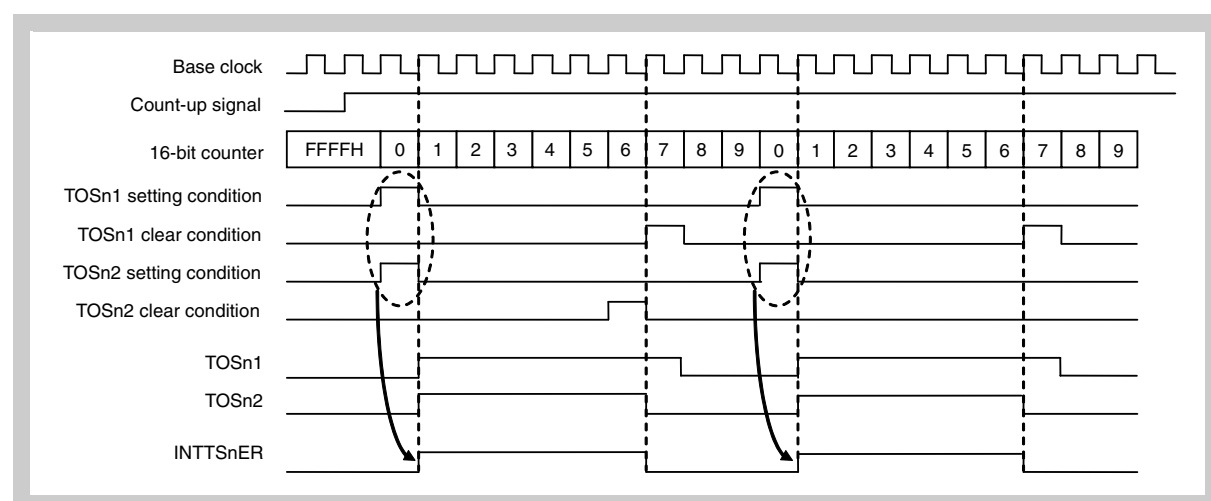
#### (a) Caution about error interrupt during dead time control

When the TSnDSE bit of the TSnOPT0 register is set to 1, or the TSnTBA0, TSnTBA1 and TSnTBA2 bits of the TSnIOC4 register are set to 1, positive/negative phase simultaneous setting condition occurrence of the TOSn1, TOSn3 and TOSn5 pins and the TOSn2, TOSn4, and TOSn6 pins is the error interrupt occurrence condition. Therefore, error interrupt occurs by occurrence of positive/negative phase simultaneous setting condition even when TSnOE1, TSnOE2 of the TSnIOC0 register is set to 0 and the TOSn1, TOSn2 pins are fixed in inactive level. The positive/negative phase simultaneous active state detection flag is also set. Set TSnTBA0, TSnTBA1 and TSnTBA2 to 0 in order to prevent the above situation.

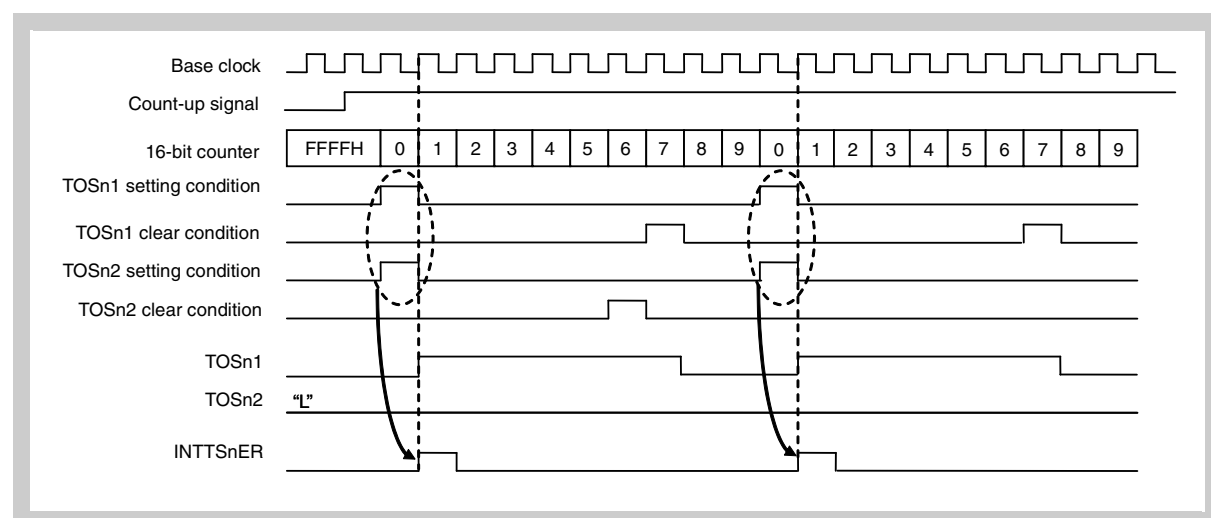
#### (b) Error interrupt detail timing

Figure 17-189 on page 933 and Figure 17-190 on page 933 show the INTTSnER trigger output timing. In case of TSnDSE = 1 of the TSnOPT0 register, 1 clock trigger is output to INTTSnER with the following operation clock when the setting condition of timer output 2 occurs. TSnTBF = 1 of the TSnOPT6 register is set simultaneously.





**Figure 17-189 Error interrupt timing (TSnDSE = 0)**

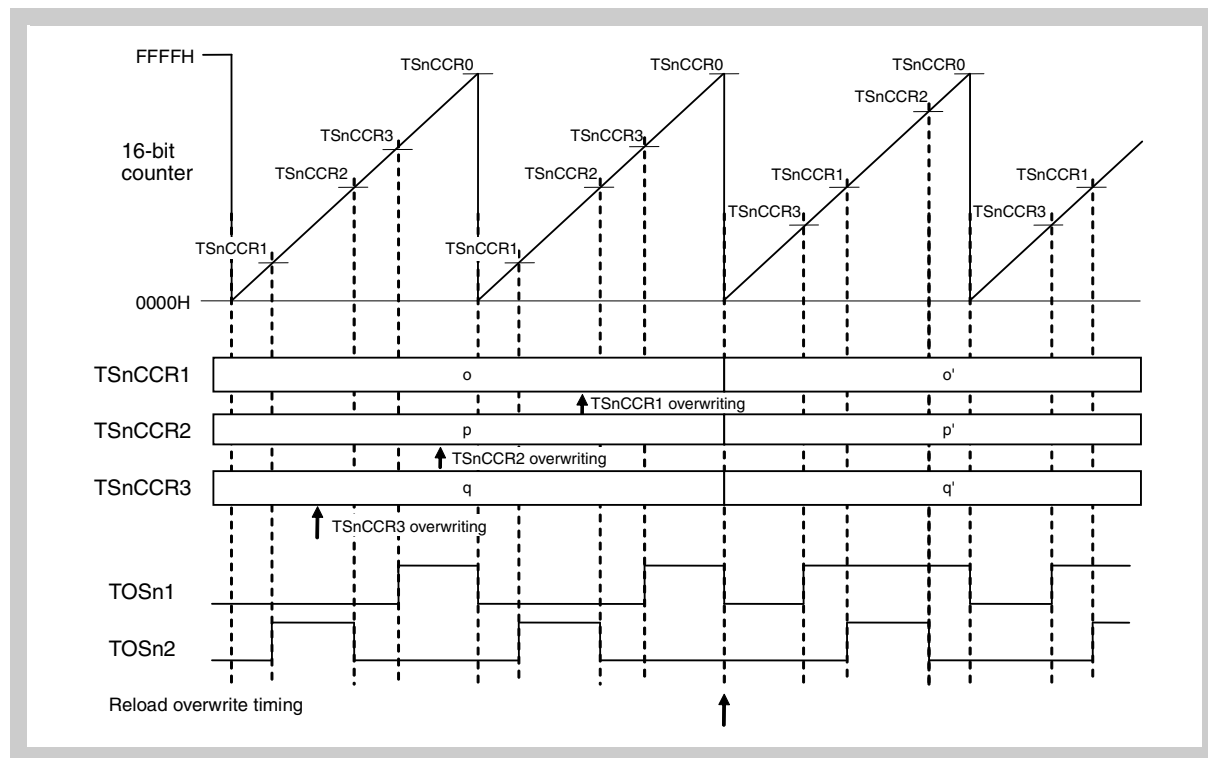


**Figure 17-190 Error interrupt timing (TSnDSE = 1)**

**Note** TOSn3 and TOSn4 and TOSn5 and TOSn6 are the same as well.

**(7) Compare register rewriting operation in special pattern output mode**

The compare register (TSnCCR0 to TSnCCR3) can be changed in the special pattern output mode. The change timing is the reload timing (the reload thinning out function is also enabled), so that the new setting is valid from the next reload timing after the TSnCCR1 register change.

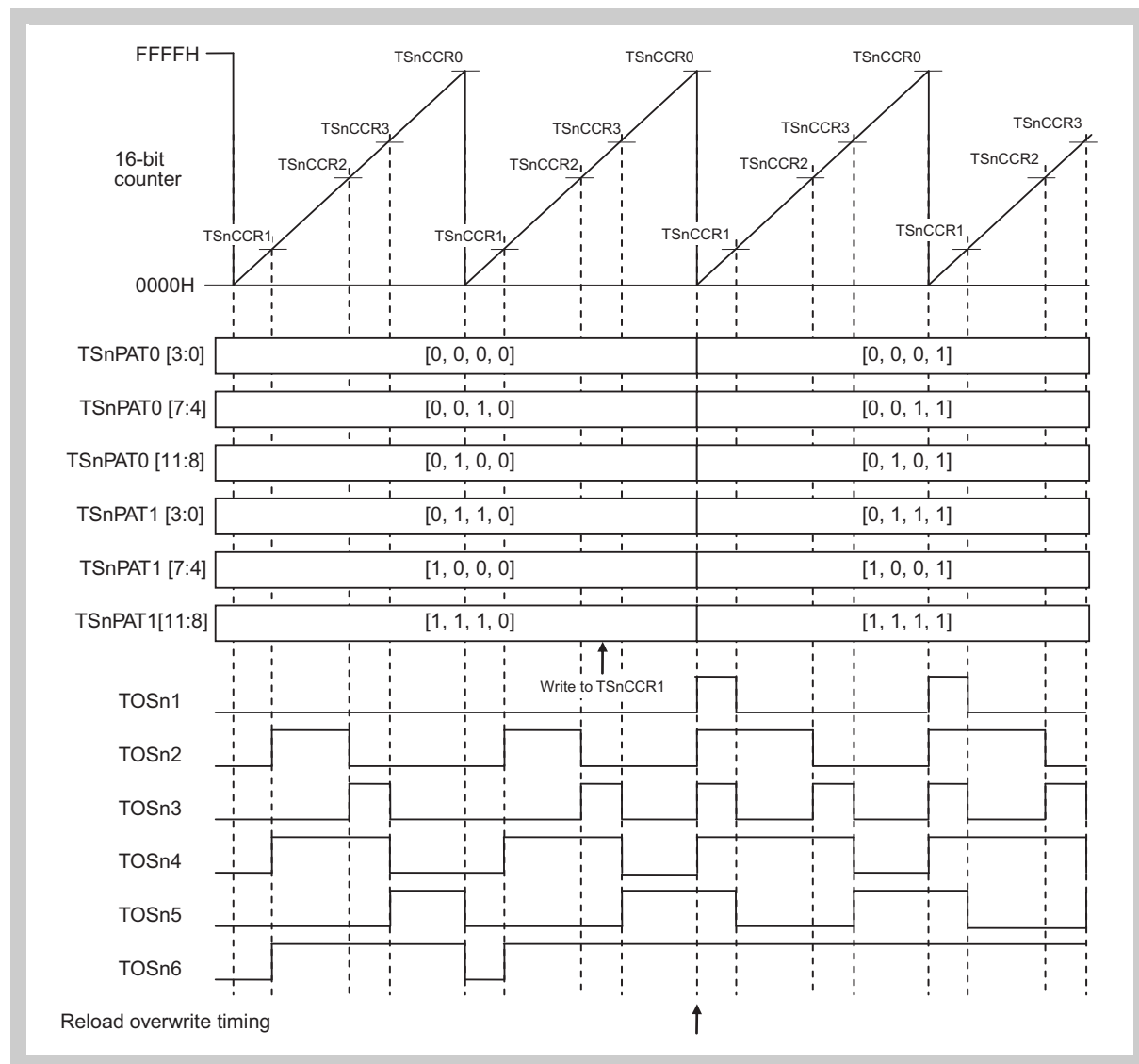


**Figure 17-191** Compare register rewriting operation in special pattern output mode

**Note** TSnPAT0[7:4] = [1, 0, 0, 0]  
 TSnPAT0[11:8] = [0, 0, 1, 0]

**(8) Pattern register rewriting operation in special pattern output mode**

In the special pattern output mode, pattern setting registers (TSnPAT0, TSnPAT1) can be rewritten dynamically. The change valid timing is the reload timing (the reload thinning out function is also enabled), so that writing to the TSnCCR1 register is required in order to generate reload timing.



**Figure 17-192** Pattern register rewriting operation in special pattern output mode

### 17.10.13 Software output function

The software output control function can be used in the high-accuracy T-PWM mode, PWM mode with dead time, 120° excitation mode and special 120° excitation mode. The function enables the TOSn1 to TOSn6 pins to switch 6 patterns of output with software by using the TSnSOC bit of the TSnOPT4 register, TSnIPC2 to TSnIPC0 bits, and the TSnIDC bit of the TSnOPT7 register.

The TOSn1 to TOSn6 pins are immediately switched to the software output control by switching from TSnSOC = 0 to TSnSOC = 1.

Conversely, software output control is released at the reload timing if switched from TSnSOC = 1 to TSnSOC = 0.

**Table 17-40 Register Description of Software Output Control Function**

Register	Operation
TSnSOC	TSnSOC = 1
TSnIDC	Set the output pattern
TSnPOT, TSnPSS, TSnPSC	Setting invalid
TSnIPC0 to TSnIPC2	Set the output pattern shown in <i>Table 17-41 on page 936</i> and <i>Table 17-42 on page 936</i>

**Table 17-41 Software output control output pattern (TSnIDC of TSnOPT7 register = 0)**

TSnOPT4 register: TSnSOC = 1, TSnOPT7 register: TSnIDC = 0

Output pin	TSnOPF2 to TSnOPF0 of TSnOPT5 Register							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	ACT	ACT	ACT	INACT	INACT	INACT	Note 1	Note 2
TOSn2	INACT	INACT	INACT	ACT	ACT	ACT	Note 1	Note 2
TOSn3	INACT	INACT	ACT	ACT	ACT	INACT	Note 1	Note 2
TOSn4	ACT	ACT	INACT	INACT	INACT	ACT	Note 1	Note 2
TOSn5	ACT	INACT	INACT	INACT	ACT	ACT	Note 1	Note 2
TOSn6	INACT	ACT	ACT	ACT	INACT	INACT	Note 1	Note 2

- Note**
1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], the output remains the level before the change.
  2. Output remains the level before the change.
  3. ACT:Active level output
  4. INACT:Inactive level output

**Table 17-42 Software output control output pattern (TSnIDC = 1 of TSnOPT7 register)**

TSnOPT4 register: TSnSOC = 1, TSnOPT7 register: TSnIDC = 1

Output pin	TSnOPF2 to TSnOPF0 of TSnOPT5 register							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	INACT	ACT	ACT	ACT	Note 1	Note 2
TOSn2	ACT	ACT	ACT	INACT	INACT	INACT	Note 1	Note 2
TOSn3	ACT	ACT	INACT	INACT	INACT	ACT	Note 1	Note 2
TOSn4	INACT	INACT	ACT	ACT	ACT	INACT	Note 1	Note 2

Output pin	TSnOPF2 to TSnOPF0 of TSnOPT5 register							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn5	INACT	ACT	ACT	ACT	INACT	INACT	Note 1	Note 2
TOSn6	ACT	INACT	INACT	INACT	ACT	ACT	Note 1	Note 2

- Note**
1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], the output remains the level before the change.
  2. Output remains the level before the change.
  3. ACT:Active level output
  4. INACT:Inactive level output

### 17.10.14 180° excitation control function

The 180° excitation control function is enabled in the high-accuracy T-PWM mode, 120° excitation mode and special 120° excitation mode.

The function provides 6 patterns of output to the TOSn1 to TOSn6 pins by using the TSnADC bit of the TSnOPT5 register, the TSnIPC2 to TSnIPC0 bits of the TSnOPT4 register, the TSnIDC bit of the TSnOPT7 register and external trigger (TSnSTCI0, TSnSTCI1 signals or TAPTSn2 to TAPTSn0 pins).

The TOSn1 to TOSn6 pins are immediately switched to 180° excitation control by switch from “TSnADC = 0” to “TSnADC = 1”.

Conversely, 180° excitation control is released at the reload timing if switched from TSnADC = 1 to TSnADC = 0.

**Table 17-43 Register description of 180° excitation control function**

Register	Operation
TSnSOC of TSnOPT4 register	TSnSOC = 0
TSnIPC2 to TSnIPC0 of TSnOPT4 register	Set the output pattern shown in the following <i>Table 17-44 on page 938</i> and <i>Table 17-45 on page 939</i>
TSnPSC of TSnOPT4 register	Setting invalid
TSnPSS, TSnTSF of TSnOPT5 register	
TSnPOT of TSnOPT5 register	Selection of pattern switch method or trigger switch method
TSnADC of TSnOPT5 register	180° excitation control function is enabled with TSnADC = 1.
TSnIDC of TSnOPT7 register	Set the output pattern (rotation direction).
TSnARD pin	Setting invalid

**Table 17-44 Software output control output pattern (TSnIDC = 0 of TSnOPT7 register)**

TSnOPT4 register: TSnSOC = 0,  
 TSnOPT5 register: TSnADC = 1, TSnPOT = 1,  
 TSnOPT7 register: TSnIDC = 0

Output pin	TSnOPF2 to TSnOPF0 of TSnOPT5 register							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	ACT	ACT	ACT	INACT	INACT	INACT	Note 1	Note 2
TOSn2	INACT	INACT	INACT	ACT	ACT	ACT	Note 1	Note 2
TOSn3	INACT	INACT	ACT	ACT	ACT	INACT	Note 1	Note 2
TOSn4	ACT	ACT	INACT	INACT	INACT	ACT	Note 1	Note 2
TOSn5	ACT	INACT	INACT	INACT	ACT	ACT	Note 1	Note 2
TOSn6	INACT	ACT	ACT	ACT	INACT	INACT	Note 1	Note 2

- Note**
1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], the output remains the level before the change.
  2. Output remains the level before the change.
  3. Active level output
  4. INACT:Inactive level output

**Table 17-45 Software Output Control Output Pattern (TSnIDC = 1 of TSnOPT7 Register)**

TSnOPT4 register: TSnSOC = 0,  
 TSnOPT5 register: TSnADC = 1, TSnPOT = 1,  
 TSnOPT7 register: TSnIDC = 1

Output pin	TSnOPF2 to TSnOPF0 of TSnOPT5 register							
	[1, 0, 1]	[1, 0, 0]	[1, 1, 0]	[0, 1, 0]	[0, 1, 1]	[0, 0, 1]	[0, 0, 0]	[1, 1, 1]
TOSn1	INACT	INACT	INACT	ACT	ACT	ACT	Note 1	Note 2
TOSn2	ACT	ACT	ACT	INACT	INACT	INACT	Note 1	Note 2
TOSn3	ACT	ACT	INACT	INACT	INACT	ACT	Note 1	Note 2
TOSn4	INACT	INACT	ACT	ACT	ACT	INACT	Note 1	Note 2
TOSn5	INACT	ACT	ACT	ACT	INACT	INACT	Note 1	Note 2
TOSn6	ACT	INACT	INACT	INACT	ACT	ACT	Note 1	Note 2

- Note**
1. INACT if no change from the initial value. If the value is changed from any value to [0, 0, 0], the output remains the level before the change.
  2. Output remains the level before the change.
  3. ACT:Active level output
  4. INACT:Inactive level output

# Chapter 18 16-bit Timer/Event Counter T (TMT)

This microcontroller has two instances of the Timer T (TMT), TMT0 to TMT1.

**Note** Throughout this chapter, the individual instances of Timer T are identified by “n” (n = 0 or 1), for example TTnCTL1 for the Timer Tn control register 1.

## 18.1 Features

Timer T (TMT) is a 16-bit timer/event counter that provides general-purpose functions.

Timer T can perform the following operations:

- Interval timer function
- External event count function
- One-shot pulse output function
- External trigger pulse function
- 16-bit accuracy PWM output function
- Free-running function
- Pulse width measurement function
- 2-phase encoder function
- Triangular wave PWM output function
- Offset trigger generation function

## 18.2 Function Outline

- Capture trigger input signal × 2
- Encoder input signal × 2
- Encoder clear signal × 1
- External trigger input signal × 1
- External event input × 1
- Readable counter × 1
- Count write buffer × 1
- Capture/compare reload register × 2
- Capture/compare match interrupt × 2
- Timer Output (TO) × 2
- Capture/compare match signal × 2
- Overflow interrupt × 1
- Encoder clear interrupt × 1



## 18.3 Configuration

Timer T is configured of the following hardware.

**Table 18-1 Timer T configuration**

Item	Configuration
Counter	16-bit counter
Registers	TMTn capture/compare registers 0, 1 (TTnCCR0, TTnCCR1) TMTn counter read buffer register (TTnCNT) TMTn counter write buffer register (TTnTCW) TMTn capture/compare buffer registers 0, 1
Timer input pins	7 (TITn0, TITn1, TEVTTn, TTRGTn, TENCTn0, TENCTn1, TECRTn)
Timer output pins	2 (TOTn0, TOTn1)
Timer input signals	TTTTI0, TTTTI1, TTTENC0, TTTENC1, TTTTRG, TTTEVT, TTTECR
Timer output signals	TTTTO0, TTTTO1, TTTICC0, TTTICC1, TTnEQC0, TTnEQC1
Control registers	TMTn control registers 0, 1 (TTnCTL0 to TTnCTL2) TMTn I/O control registers 0 to 3 (TTnIOC0 to TTnIOC3) TMTn option registers 0 to 2 (TTnOPT0 to TTnOPT2) TMTn input selection register (TAAIC2, See "TAAIC2 - Timer input control register 2" on page 551.)
Interrupts	Compare match interrupt (INTTTnCC0, INTTTnCC1) Overflow interrupt (INTTTnOV) Encoder clear interrupt (INTTTnEC)

**Table 18-2 List of Timer T registers (1/2)**

Address	Register name	Symbol	R/W	Manipulable Bit Units			After Reset
				1	8	16	
FFFFF690H	TMT0 control register 0	TT0CTL0	R/W	×	×		00H
FFFFF691H	TMT0 control register 1	TT0CTL1	R/W	×	×		00H
FFFFF692H	TMT0 control register 2	TT0CTL2	R/W	×	×		00H
FFFFF693H	TMT0 I/O control register 0	TT0IOC0	R/W	×	×		00H
FFFFF694H	TMT0 I/O control register 1	TT0IOC1	R/W	×	×		00H
FFFFF695H	TMT0 I/O control register 2	TT0IOC2	R/W	×	×		00H
FFFFF696H	TMT0 I/O control register 3	TT0IOC3	R/W	×	×		00H
FFFFF697H	TMT0 option register 0	TT0OPT0	R/W	×	×		00H
FFFFF698H	TMT0 option register 1	TT0OPT1	R/W	×	×		00H
FFFFF699H	TMT0 option register 2	TT0OPT2	R/W	×	×		00H
FFFFF69AH	TMT0 capture/compare register 0	TT0CCR0	R/W			×	0000H
FFFFF69CH	TMT0 capture/compare register 1	TT0CCR1	R/W			×	0000H
FFFFF69EH	TMT0 counter read buffer register	TT0CNT	R			×	0000H <sup>Note</sup>
FFFFF990H	TMT0 counter write buffer register	TT0TCW	R/W			×	0000H
FFFFF6A0H	TMT1 control register 0	TT1CTL0	R/W	×	×		00H
FFFFF6A1H	TMT1 control register 1	TT1CTL1	R/W	×	×		00H
FFFFF6A2H	TMT1 control register 2	TT1CTL2	R/W	×	×		00H
FFFFF6A3H	TMT1 I/O control register 0	TT1IOC0	R/W	×	×		00H

**Table 18-2 List of Timer T registers (2/2)**

Address	Register name	Symbol	R/W	Manipulable Bit Units			After Reset
				1	8	16	
FFFFF6A4H	TMT1 I/O control register 1	TT1IOC1	R/W	×	×		00H
FFFFF6A5H	TMT1 I/O control register 2	TT1IOC2	R/W	×	×		00H
FFFFF6A6H	TMT1 I/O control register 3	TT1IOC3	R/W	×	×		00H
FFFFF6A7H	TMT1 option register 0	TT1OPT0	R/W	×	×		00H
FFFFF6A8H	TMT1 option register 1	TT1OPT1	R/W	×	×		00H
FFFFF6A9H	TMT1 option register 2	TT1OPT2	R/W	×	×		00H
FFFFF6AAH	TMT1 capture/compare register 0	TT1CCR0	R/W			×	0000H
FFFFF6ACH	TMT1 capture/compare register 1	TT1CCR1	R/W			×	0000H
FFFFF6AEH	TMT1 counter read buffer register	TT1CNT	R			×	0000H <sup>Note</sup>
FFFFF9A0H	TMT1 counter write buffer register	TT1TCW	R/W			×	0000H

**Note** When TTnCE = 0

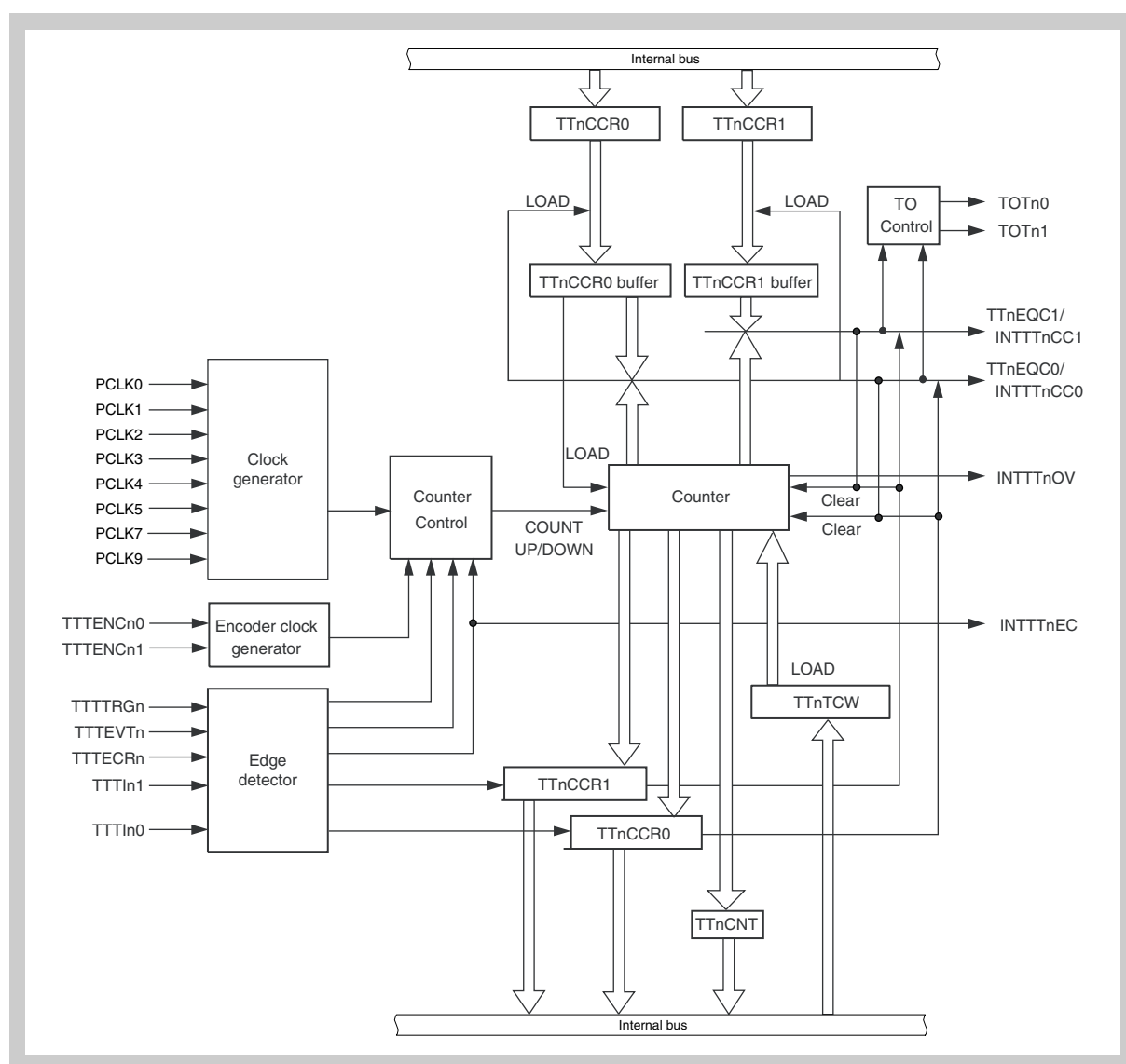


Figure 18-1 Block diagram of Timer T

**Note** 1.  $m = 0, 1$

2. PCLKn: peripheral clocks (refer to "Clock Generator" on page 148):

PCLK	0	1	2	3	4	5	7	9
f [MHz]	32	16	8	4	2	1	0.25	0.0625

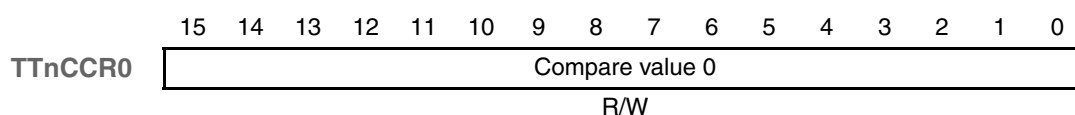
**(1) TTnCCR0 - TMTn capture/compare register 0**

The TTnCCR0 register is a 16-bit register that functions both as a capture register and as a compare register.

**Access** This registers can be read/written in 16-bit units.

**Address** TT0CCR0: FFFFF69A<sub>H</sub>  
TT1CCR0: FFFFF6AA<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. These registers are cleared by any reset.



The capture and compare functions are as follows in each mode.

**Table 18-3 Capture/compare functions in each mode**

Operation mode	Capture/compare setting of TTnCCR0 register	Rewriting method during compare	Counter clear function
Interval mode	Compare only	Anytime write	Compare match
External event count mode	Compare only	Anytime write	Compare match
External trigger pulse output mode	Compare only	Batch write (Reload)	Compare match
One-shot pulse mode	Compare only	Anytime write	Compare match
PWM mode	Compare only	Batch write (Reload)	Compare match
Free-running mode	Capture/compare selectable	Anytime write	-
Pulse width measurement mode	Capture only	-	External input (TITn0 pin)
Triangular wave PWM mode	Compare only	Batch write (Reload) <sup>Note 1</sup>	Compare match
Encoder compare mode	Compare only	Anytime write	Depends on set condition <sup>Note 2</sup>
Encoder capture mode	Capture only	-	-
Encoder capture compare mode	Compare only	Anytime write	Depends on set condition <sup>Note 2</sup>
Offset trigger generation mode	Capture only	-	External input (TITn0 pin)

**Note** 1. The batch write reload timing is the counter underflow timing only.

2. The condition is set with the TTnECM0 and TTnECM1 bits of the TTnCTL2 register.

**Use as compare register** When TTnCE = 1, the TTnCCR0 register rewrite method differs according to the operation mode. Refer to *Table 18-3 on page 944*.  
(For details about the compare register rewrite operation, refer to “Method for writing to compare register” on page 966.)

**Use as capture register** The counter value is saved to the TTnCCR0 register upon TITn0 pin input edge detection. The function to clear counters following capture differs according to the operation mode. Refer to *Table 18-3 on page 944*.

**(2) TTnCCR1 - TMTn capture/compare register 1**

The TTnCCR1 register is a 16-bit register that functions both as a capture register and a compare register.

**Access** This registers can be read/written in 16-bit units.

**Address** TT0CCR1: FFFFF69C<sub>H</sub>  
TT1CCR1: FFFFF6AC<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. These registers are cleared by any reset.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>TTnCCR1</b>	Compare value 0															
	R/W															

The capture/compare functions in each operation mode are as follows.

**Table 18-4 Capture/compare functions in each mode**

Operation mode	Capture/compare setting of TTnCCR1 register	Rewriting method during compare	Counter clear function
Interval mode	Compare only	Anytime write	-
External event count mode	Compare only	Anytime write	-
External trigger pulse output mode	Compare only	Batch write (Reload)	-
One-shot pulse mode	Compare only	Anytime write	-
PWM mode	Compare only	Batch write (Reload)	-
Free-running mode	Capture/compare selectable	Anytime write	-
Pulse width measurement mode	Capture only	-	External input (TITn1 pin)
Triangular wave PWM mode	Compare only	Batch write (Reload) <sup>Note 1</sup>	-
Encoder compare mode	Compare only	Anytime write	Depends on set conditions <sup>Note 2</sup>
Encoder capture mode	Capture only	-	-
Encoder capture compare mode	Capture only	-	-
Offset trigger generation mode	Compare only	Batch write (Reload) <sup>Note 3</sup>	-

- Note**
1. The batch write reload timing is the counter underflow occurrence timing only.
  2. The conditions are set with bits TTnECM0 and TTnECM1 of TTnCTL2 register.
  3. The batch write reload timing is the counter's 0000H clear timing only.

**Use as compare register** When TTnCE = 1, the write method of register TTnCCR1 differs according to the operation mode. Refer to *Table 18-4 on page 945*.  
(For details about the compare register rewrite operation, refer to “Method for writing to compare register” on page 966.)

**Use as capture register** The counter value upon TITn1 pin input edge detection is saved to the TTnCCR1 register. The function to clear the counter following capture also differs according to the mode. Refer to *Table 18-4 on page 945*.

**(3) TTnCNT - TMTn counter read buffer register**

The TTnCNT register is a read buffer register that can read the counter value.

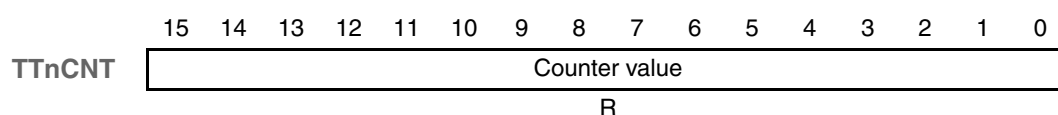
**Access** This register can be read only in 16-bit units.

**Address** TT0CNT: FFFFF69E<sub>H</sub>  
TT1CNT: FFFFF6AE<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. This register is cleared by any reset.

**Note** When, in the encoder compare mode, encoder capture mode, and encoder capture/compare mode, the value of the TTnCE bit is changed from “1” to “0”, the value that can be read by the TTnCNT register differs according to the following conditions.

- When bit TTnECC of the TTnCTL2 register = 0, 0000<sub>H</sub> can be read.
- When bit TTnECC = 1, the value held when bit TTnCE was cleared to “0” can be read.

**(4) TTnTCW - TMTn counter write buffer register**

The TTnTCW register is a write buffer register that can write the counter value.

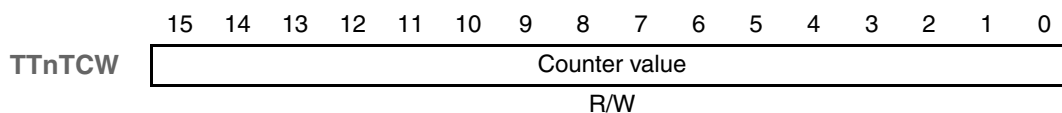
The setting value is valid only in the encoder compare mode, encoder capture mode, and encoder capture/compare mode. (In all other modes, the setting value is invalid.)

**Access** This registers can be read/written in 16-bit units.

**Address** TT0TCW: FFFFF990<sub>H</sub>  
TT1TCW: FFFFF9A0<sub>H</sub>

**Initial Value** 0000<sub>H</sub>. This register is cleared by any reset.

**Note** When TTnECC of register TTnCTL2 = 0, the setting value is loaded to the counter when the TTnCE bit is set (to 1). (When TTnECC = 1, the counter holds its value, so it is not reloaded.)



## 18.4 Control Registers

### (1) TTnCTL0 - TMTn control register 0

TTnCTL0 is an 8-bit register that controls the operation of TMTn.

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

**Address** TT0CTL0: FFFFF690<sub>H</sub>

TT1CTL0: FFFFF6A0<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
TTnCTL0	TTnCE	0	0	0	0	TTnCKS2	TTnCKS1	TTnCKS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Caution** When TTnCE = 1, only the TTnCE bit of the TTnCTL0 register can be changed. Perform write access to the other bits using the same values.

**Table 18-5 TTnCTL0 register contents (1/2)**

Bit position	Bit name	Function
7	TTnCE	<p>Controls the timer TMTn operation.</p> <p>0: Internal operating clock operation disabled (TMTn reset asynchronously)</p> <p>1: Internal operating clock operation enabled</p> <p>When bit TTnCE is set to “0”, the internal operation clock of TMTn stops (fixed to low level), and TMTn is reset asynchronously.</p> <p>When bit TTnCE is set to “1”, the internal operation of TMTn is enabled from when bit TTnCE was set to “1” and count-up is performed. The time until count-up is as listed in Table TMTn Count Clock and Time Until Count-Up.</p> <p><b>Note:</b></p> <ol style="list-style-type: none"> <li>In the encoder compare mode, encoder capture mode, and encoder capture/compare mode, the functions that are reset when TTnCE = 0 and TTnECC = 1 are as follows. <ul style="list-style-type: none"> <li>Compare match detector (interrupt output low level)</li> <li>Timer output (Output inactive level)</li> <li>Edge detector for other than pins TENCTn0, TENCTn1, and TECRTn</li> </ul> </li> <li>The following functions are not reset. <ul style="list-style-type: none"> <li>Counter</li> <li>Flags in TTnOPT1 register</li> <li>TTnCCR0 buffer, TTnCCR1 buffer register, counter read buffer register</li> <li>TENCTn0, TENCTn1, TECRTn pin edge detector</li> </ul> </li> <li>In modes other than the above, (in which TTnECC is fixed to 0), the functions that are reset by TTnCE = 0 are as follows. <ul style="list-style-type: none"> <li>Internal registers other than registers that can be written from the CPU, and internal latch circuits</li> </ul> </li> </ol>

**Table 18-5 TTnCTL0 register contents (2/2)**

Bit position	Bit name	Function																																				
2 to 0	TTnCKS2 TTnCKS1 TTnCKS0	Selects the count clock of timer TMTn.																																				
		<table><tr><th>TTnCKS2</th><th>TTnCKS1</th><th>TTnCKS0</th><th>Internal count clock selection</th></tr><tr><td>0</td><td>0</td><td>0</td><td>PCLK0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>PCLK1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>PCLK2</td></tr><tr><td>0</td><td>1</td><td>1</td><td>PCLK3</td></tr><tr><td>1</td><td>0</td><td>0</td><td>PCLK4</td></tr><tr><td>1</td><td>0</td><td>1</td><td>PCLK5</td></tr><tr><td>1</td><td>1</td><td>0</td><td>PCLK7</td></tr><tr><td>1</td><td>1</td><td>1</td><td>PCLK9</td></tr></table>	TTnCKS2	TTnCKS1	TTnCKS0	Internal count clock selection	0	0	0	PCLK0	0	0	1	PCLK1	0	1	0	PCLK2	0	1	1	PCLK3	1	0	0	PCLK4	1	0	1	PCLK5	1	1	0	PCLK7	1	1	1	PCLK9
		TTnCKS2	TTnCKS1	TTnCKS0	Internal count clock selection																																	
		0	0	0	PCLK0																																	
		0	0	1	PCLK1																																	
		0	1	0	PCLK2																																	
		0	1	1	PCLK3																																	
		1	0	0	PCLK4																																	
		1	0	1	PCLK5																																	
		1	1	0	PCLK7																																	
1	1	1	PCLK9																																			
<b>Note:</b> 1. Set the TTnCKS[2:0] bits while TTnCE = 0. The TTnCKS[2:0] bits can be changed simultaneously when the setting of TTnCE is changed from 0 to 1.																																						
2. The TTnCKS[2:0] bits are valid only when TTnCTL1.TTnEEE is set to 0.																																						

**Table 18-6 TMTn Count Clock and Count Delay**

Count clocks	TTnCKS2	TTnCKS1	TTnCKS0	Count delay	
				Minimum	Maximum
PCLK0	0	0	0	3 base clocks	4 base clocks
PCLK1	0	0	1		
PCLK2	0	1	0		
PCLK3	0	1	1	4 base clocks	5 base clocks + 1 count clock
PCLK4	1	0	0		
PCLK5	1	0	1		
PCLK7	1	1	0		
PCLK9	1	1	1		

**Note**  $f_{TMTn}$ : Base clock of TMTn ( $f_{TMTn} = f_{PCLK0}$ )



**(2) TTnCTL1 - TMTn control register 1**

The TTnCTL1 register is an 8-bit register that controls the operation of TMTn.

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

**Address** TT0CTL1: FFFFF691<sub>H</sub>  
TT1CTL1: FFFFF6A1<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
TTnCTL1	0	TTnEST	TTnEEE	0	TTnMD3	TTnMD2	TTnMD1	TTnMD0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Caution** Set the TTnCTL1 register when TTnCTL0.TTnCE = 0. When TTnCE = 1, bits other than bit TTnEST (TTnEEE, TTnMD3 to TTnMD0, TTnSYE) can be written using the same value.

**Table 18-7 TTnCTL1 register contents (1/2)**

Bit position	Bit name	Function
6	TTnEST	<p>Sets to enable/disable software trigger control.</p> <p>0: No operation/Disables software trigger control.</p> <p>1: Enables software trigger control.</p> <ul style="list-style-type: none"> <li>In one-shot pulse mode (One-shot pulse software trigger) Can be made to function as a software trigger by setting TTnETS to 1 when TTnCE = 1. Always write TTnEST = 1 when TTnCE = 1.</li> <li>In external trigger pulse output mode (Pulse output software trigger)</li> </ul> <hr/> <p><b>Caution:</b> The function operates as a software trigger by setting TTnEST to 1 during the timer TMTn operation (when TTnCTL0.TTnCE = 1). The function operates as a software trigger by setting TTnCTL0.TTnCE = 1 when the value of TTnEST is 1 while TTnCTL0.TTnCE = 0.</p> <hr/> <p><b>Note:</b> The TTnEST bit is always read as "0"</p>
5	TTnEEE	<p>Specifies count clock input.</p> <p>0: Use the clock selected with the TTnCTL0.TTnCKS[2:0] bits.</p> <p>1: Use the external clock (TEVTTn pin input edge).</p> <p>Specification of the valid edge when TTnEEE = 1 (external clock: TEVTTn pin) is set with bits TTnEES1 and TTnEES0 of TTnIOC2 register.)</p> <p><b>Note:</b> The setting of bit TTnEEE is invalid in the external event count mode, encoder compare mode, encoder capture mode, encoder capture/compare mode.</p> <hr/> <p><b>Caution:</b> Rewrite the TTnEEE bit only when TTnCE = 0. (The same value can be written when TTnCE = 1.) The operation is not guaranteed if rewriting is performed when TTnCE = 1. If rewriting was mistakenly performed, set TTnCE = 0 and then set the bit again.</p> <hr/>

Table 18-7 TTnCTL1 register contents (2/2)

Bit position	Bit name	Function				
3 to 0	TTnMD3 TTnMD2 TTnMD1 TTnMD0	Selects the timer mode..				
		TTnMD3	TTnMD2	TTnMD1	TTnMD0	Internal count clock selection
		0	0	0	0	Interval 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 mode
		0	1	0	0	PWM mode
		0	1	0	1	Free-running mode
		0	1	1	0	Pulse width measurement mode
		0	1	1	1	Triangular wave PWM mode
		1	0	0	0	Encoder compare mode
		1	0	0	1	Encoder capture mode
		1	0	1	0	Encoder capture compare mode
		1	1	0	0	Offset trigger generation mode
		Other than above				Setting prohibited
		<b>Caution:</b> Rewrite the TTnMD3 to TTnMD0 bits only when TTnCE = 0. (The same value can be written when TTnCE = 1.) The operation is not guaranteed if rewriting is performed when TTnCE = 1. If rewriting was mistakenly performed, set TTnCE = 0.				

**(3) TTnCTL2 - TMTn control register 2**

The TTnCTL2 register is an 8-bit register that controls the operation of TMTn.

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

**Address** TT0CTL2: FFFFF692<sub>H</sub>  
TT1CTL2: FFFFF6A2<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>TTnCTL2</b>	TTnECC	0	0	TTnLDE	TTnECM1	TTnECM0	TTnUDS1	TTnUDS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Caution**
1. The settings of the TTnCTL2 register are valid only in the encoder compare mode, encoder capture mode, and encoder capture/compare mode. (The settings of this register are invalid in all other modes.)
  2. Set the TTnCTL2 register when TTnCTL0.TTnCE = 0. When TTnCE = 1, write access to the TTnCTL2 register can be performed with the same value.

**Table 18-8 TTnCTL2 register contents (1/2)**

Bit position	Bit name	Function
7	TTnECC	<p>Selection of initialization/hold of counter value.</p> <p>0: Initialize counter value when TTnCTL0.TTnCE = 0.</p> <p>1: Hold counter value when TTnCTL0.TTnCE = 0.</p> <ul style="list-style-type: none"> <li>• When TTnECC = 0, setting TnCTL0.TTnCE = 0 causes the counter to be reset to FFFFH, the capture registers (TTnCCR0/TTnCCR1) to be reset to 0000H, and the encoder dedicated flags (TTnEOF/TTnEUF/TTnESF) to be reset to 0. When TTnECC = 0, the value of the TTnTCW register is loaded to the counter when TTnCE is set from 0 to 1.</li> <li>• When TTnECC = 1, setting TnCTL0.TTnCE = 0 causes the values of the counter, capture registers (TTnCCR0/TTnCCR1), and encoder dedicated flags (TTnEOF/TTnEUF/TTnESF) to be held. When TTnECC = 1, the value of the TTnTCW register is not loaded to the counter.</li> </ul> <p><b>Note:</b> The setting of bit TTnECC is valid in the encoder compare mode, encoder capture mode, and encoder capture/compare mode. (In all other modes, it is invalid.)</p>
4	TTnLDE	<p>Enables encoder load.</p> <p>0: Disable transfer of compare setting value to counter.</p> <p>1: Enable transfer of compare setting value (TTnCCR0) to counter when underflow occurs.</p> <p><b>Note:</b> The setting of bit TTnLDE is valid in the encoder compare mode and the encoder capture mode and bits TTnECM1 and TTnECM0 are set as follows.</p> <ul style="list-style-type: none"> <li>• TTnECM1 = 0, TTnECM0 = 0 or 1</li> </ul>
3	TTnECM1	<p>Enables encoder clear mode on match of counter and TTnCCR1 register.</p> <p>0: No clear condition.</p> <p>1: When the counter and TTnCCR1 register match, clear the counter if the next count is a down count (TTnESF = 1)</p> <p><b>Note:</b> The setting of bit TTnECM1 is valid in the encoder compare mode. (It is invalid in the encoder capture mode and encoder capture compare mode.)</p>

**Table 18-8 TTnCTL2 register contents (2/2)**

Bit position	Bit name	Function															
2	TTnECM0	<p>Enables encoder clear mode on match of counter and TTnCCR0 register.</p> <p>0: No clear condition.</p> <p>1: When the counter and TTnCCR0 register match, clear the counter if the next count is a down count (TTnESF = 1)</p> <p><b>Note:</b> The setting of bit TTnECM0 is valid in the encoder compare mode. (It is invalid in the encoder capture mode and encoder capture compare mode.)</p>															
3 to 0	TTnUDS1 TTnUDS0	<p>Selects the encoder operation mode..</p> <table border="1"> <thead> <tr> <th>TTnUDS1</th><th>TTnUDS0</th><th>Encoder operation mode</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td> <p>Upon detection of the valid edge of the A phase of encoder input (TENCTn0 pin), the following count operation is performed in the B phase of encoder input.</p> <ul style="list-style-type: none"> <li>When "high", count down.</li> <li>When "low", count up.</li> </ul> </td></tr> <tr> <td>0</td><td>1</td><td> <p>Count up upon detection of valid edge of A phase of encoder input (TENCTn0 pin).</p> <p>Count down upon detection of valid edge of B phase of encoder input (TENCTn1 pin).</p> </td></tr> <tr> <td>1</td><td>0</td><td> <p>Count up at rising edge of A phase of encoder input (TENCTn0 pin). Count down at falling edge of A phase of encoder input.</p> <p>However, count operation is performed only when B phase of encoder input (TENCTn1 pin) is "low".</p> </td></tr> <tr> <td>1</td><td>1</td><td> <p>Detection of both edges of phase A of encoder input (TENCTn0 pin)/phase B of encoder input (TENCTn1 pin).</p> <p>Judgment of count operation based on combination of detection edge and input level.</p> </td></tr> </tbody> </table> <p><b>Note:</b> When TTnUDS[1:0] bits are set to 10B or 11B, the settings of bits TTnIOC3.TTnEIS[1:0] are invalid, and the function is fixed to the setting for detection of both edges.</p>	TTnUDS1	TTnUDS0	Encoder operation mode	0	0	<p>Upon detection of the valid edge of the A phase of encoder input (TENCTn0 pin), the following count operation is performed in the B phase of encoder input.</p> <ul style="list-style-type: none"> <li>When "high", count down.</li> <li>When "low", count up.</li> </ul>	0	1	<p>Count up upon detection of valid edge of A phase of encoder input (TENCTn0 pin).</p> <p>Count down upon detection of valid edge of B phase of encoder input (TENCTn1 pin).</p>	1	0	<p>Count up at rising edge of A phase of encoder input (TENCTn0 pin). Count down at falling edge of A phase of encoder input.</p> <p>However, count operation is performed only when B phase of encoder input (TENCTn1 pin) is "low".</p>	1	1	<p>Detection of both edges of phase A of encoder input (TENCTn0 pin)/phase B of encoder input (TENCTn1 pin).</p> <p>Judgment of count operation based on combination of detection edge and input level.</p>
TTnUDS1	TTnUDS0	Encoder operation mode															
0	0	<p>Upon detection of the valid edge of the A phase of encoder input (TENCTn0 pin), the following count operation is performed in the B phase of encoder input.</p> <ul style="list-style-type: none"> <li>When "high", count down.</li> <li>When "low", count up.</li> </ul>															
0	1	<p>Count up upon detection of valid edge of A phase of encoder input (TENCTn0 pin).</p> <p>Count down upon detection of valid edge of B phase of encoder input (TENCTn1 pin).</p>															
1	0	<p>Count up at rising edge of A phase of encoder input (TENCTn0 pin). Count down at falling edge of A phase of encoder input.</p> <p>However, count operation is performed only when B phase of encoder input (TENCTn1 pin) is "low".</p>															
1	1	<p>Detection of both edges of phase A of encoder input (TENCTn0 pin)/phase B of encoder input (TENCTn1 pin).</p> <p>Judgment of count operation based on combination of detection edge and input level.</p>															

**(4) TTnIOC0 - TMTn I/O control register 0**

The TTnIOC0 register is an 8-bit register that controls timer output (TOTn0 and TOTn1 pins).

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

**Address** TT0IOC0: FFFFF693<sub>H</sub>  
TT1IOC0: FFFFF6A3<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
TTnIOC0	0	0	0	0	TTnOL1	TTnOE1	TTnOL0	TTnOE0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Caution** Set the TTnIOC0 register when TTnCTL0.TTnCE = 0. When TTnCE = 1, write access to the TTnIOC0 register can be performed with the same value.

**Table 18-9 TTnIOC0 register contents**

Bit position	Bit name	Function
3	TTnOL1	TOTn1 pin output level setting: 0: Normal output (Low level, when output is inactive.) 1: Inverted output (High level, when output is inactive.)
2	TTnOE1	TOTn1 pin output setting: 0: Timer output disable (TOTn1 pin output is fixed to inactive level.) 1: Timer output enabled (A pulse can be output from the TOTn1 pin.)
1	TTnOL0	TOTn0 pin output level setting: 0: Normal output (Low level, when output is inactive.) 1: Inverted output (High level, when output is inactive.)
0	TTnOE0	TOTn0 pin output setting: 0: Timer output disable (TOTn0 pin output is fixed to inactive level.) 1: Timer output enabled (A pulse can be output from the TOTn0 pin.)

**(5) TTnIOC1 - TMTn I/O control register 1**

The TTnIOC1 register is an 8-bit register that controls the valid edge of capture input (TITn1 and TITn0 pins).

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

**Address** TT0IOC1: FFFFF694<sub>H</sub>  
TT1IOC1: FFFFF6A4<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>TTnIOC1</b>	0	0	0	0	TTnIS3	TTnIS2	TTnIS1	TTnIS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Caution** Set the TTnIOC1 register when TTnCTL0.TTnCE = 0. When TTnCE = 1, write access to the TTnIOC1 register can be performed with the same value.

**Table 18-10 TTnIOC1 register contents**

Bit position	Bit name	Function															
3, 2	TTnIS3 TTnIS2	Specifies the capture input (TITn1) valid edge.															
		<table><tr><th>TTnIS3</th><th>TTnIS2</th><th>Capture input (TITn1) valid edge setting</th></tr><tr><td>0</td><td>0</td><td>No edge detection (capture operation invalid)</td></tr><tr><td>0</td><td>1</td><td>Rising edge detection</td></tr><tr><td>1</td><td>0</td><td>Falling edge detection</td></tr><tr><td>1</td><td>1</td><td>Both, rising and falling edge detection</td></tr></table>	TTnIS3	TTnIS2	Capture input (TITn1) valid edge setting	0	0	No edge detection (capture operation invalid)	0	1	Rising edge detection	1	0	Falling edge detection	1	1	Both, rising and falling edge detection
		TTnIS3	TTnIS2	Capture input (TITn1) valid edge setting													
		0	0	No edge detection (capture operation invalid)													
		0	1	Rising edge detection													
		1	0	Falling edge detection													
		1	1	Both, rising and falling edge detection													
Capture operation is performed and capture interrupt (INTTTnCC1) is output upon edge detection.																	
<b>Note:</b> The setting of bits TTnIS3 and TTnIS2 are valid in the free-running mode, pulse width measurement mode, encoder capture mode, and encoder capture compare mode.																	
1, 0	TTnIS1 TTnIS0	Specifies the capture input (TITn0) valid edge.															
		<table><tr><th>TTnIS1</th><th>TTnIS0</th><th>Capture input (TITn0) valid edge setting</th></tr><tr><td>0</td><td>0</td><td>No edge detection (capture operation invalid)</td></tr><tr><td>0</td><td>1</td><td>Rising edge detection</td></tr><tr><td>1</td><td>0</td><td>Falling edge detection</td></tr><tr><td>1</td><td>1</td><td>Both, rising and falling edge detection</td></tr></table>	TTnIS1	TTnIS0	Capture input (TITn0) valid edge setting	0	0	No edge detection (capture operation invalid)	0	1	Rising edge detection	1	0	Falling edge detection	1	1	Both, rising and falling edge detection
		TTnIS1	TTnIS0	Capture input (TITn0) valid edge setting													
		0	0	No edge detection (capture operation invalid)													
		0	1	Rising edge detection													
		1	0	Falling edge detection													
		1	1	Both, rising and falling edge detection													
Capture operation is performed and capture interrupt (INTTTnCC0) is output upon edge detection.																	
<b>Note:</b> The setting of bits TTnIS1 and TTnIS0 are valid in the free-running mode, pulse width measurement mode, encoder capture mode, and encoder capture compare mode.																	

**(6) TTnIOC2 - TMTn I/O control register 2**

The TTnIOC2 register is an 8-bit register that controls the valid edge of external event count input (TEVTTn pin) and external trigger input (TTRGTn pin).

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

**Address** TT0IOC2: FFFFF695<sub>H</sub>  
TT1IOC2: FFFFF6A5<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
TTnIOC2	0	0	0	0	TTnEES1	TTnEES0	TTnETS1	TTnETS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Caution** Set the TTnIOC2 register when TTnCTL0.TTnCE = 0. When TTnCE = 1, write access to the TTnIOC2 register can be performed with the same value.

**Table 18-11 TTnIOC2 register contents**

Bit position	Bit name	Function															
3, 2	TTnEES1 TTnEES0	Specifies the external event counter input (TEVTTn) valid edge.															
		<table><tr><th>TTnEES1</th><th>TTnEES0</th><th>External event counter input (TEVTTn) valid edge setting</th></tr><tr><td>0</td><td>0</td><td>No edge detection (capture operation invalid)</td></tr><tr><td>0</td><td>1</td><td>Rising edge detection</td></tr><tr><td>1</td><td>0</td><td>Falling edge detection</td></tr><tr><td>1</td><td>1</td><td>Both, rising and falling edge detection</td></tr></table>	TTnEES1	TTnEES0	External event counter input (TEVTTn) valid edge setting	0	0	No edge detection (capture operation invalid)	0	1	Rising edge detection	1	0	Falling edge detection	1	1	Both, rising and falling edge detection
		TTnEES1	TTnEES0	External event counter input (TEVTTn) valid edge setting													
		0	0	No edge detection (capture operation invalid)													
		0	1	Rising edge detection													
		1	0	Falling edge detection													
		1	1	Both, rising and falling edge detection													
<b>Note:</b> The settings of bits TTnEES1 and TTnEES0 are valid in the external event count mode, or when bit TTnCTL1.TTnEEE = 1.																	
1, 0	TTnETS1 TTnETS0	Specifies the external trigger input (TTRGTn) valid edge.															
		<table><tr><th>TTnETS1</th><th>TTnETS0</th><th>External trigger input (TTRGTn) valid edge setting</th></tr><tr><td>0</td><td>0</td><td>No edge detection (capture operation invalid)</td></tr><tr><td>0</td><td>1</td><td>Rising edge detection</td></tr><tr><td>1</td><td>0</td><td>Falling edge detection</td></tr><tr><td>1</td><td>1</td><td>Both, rising and falling edge detection</td></tr></table>	TTnETS1	TTnETS0	External trigger input (TTRGTn) valid edge setting	0	0	No edge detection (capture operation invalid)	0	1	Rising edge detection	1	0	Falling edge detection	1	1	Both, rising and falling edge detection
		TTnETS1	TTnETS0	External trigger input (TTRGTn) valid edge setting													
		0	0	No edge detection (capture operation invalid)													
		0	1	Rising edge detection													
		1	0	Falling edge detection													
		1	1	Both, rising and falling edge detection													
<b>Note:</b> The settings of bits TTnETS1 and TTnETS0 are valid in the external trigger pulse output mode and the one-shot pulse mode.																	

**(7) TTnIOC3 - TMTn I/O control register 3**

The TTnIOC3 register is an 8-bit register that controls the valid edge of encoder clear input (TECRTn pin) and encoder input (TENCTn1 and TENCTn0 pins).

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

**Address** TT0IOC3: FFFFF696<sub>H</sub>  
TT1IOC3: FFFFF6A6<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>TTnIOC3</b>	TTnSCE	TTnZCL	TTnBCL	TTnACL	TTnECS1	TTnECS0	TTnEIS1	TTnEIS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Caution** Set the TTnIOC3 register when TTnCTL0.TTnCE = 0. When TTnCE = 1, write access to the TTnIOC3 register can be performed with the same value.

**Table 18-12 TTnIOC3 register contents (1/2)**

Bit position	Bit name	Function
7	TTnSCE	<p>Selects the encoder counter clear method.</p> <p>0: Clear upon detection of edge of TECRTn pin 1: Clear upon match of clear condition level</p> <ul style="list-style-type: none"> <li>When TTnSCE = 1, the counter is cleared to 0000H if all the conditions set with bits TTnZCL, TTnBCL, and TTnACL are matched.</li> <li>When TTnSCE = 1, the settings of bits TTnECS1 and TTnECS0 are invalid, so no encoder clear interrupt (INTTTnEC) is output.</li> <li>When TTnSCE = 0, the settings of bits TTnZCL, TTnBCL, and TTnACL are invalid. The settings of bits TTnECS1 and TTnECS0 become valid, and the encoder clear interrupt (INTTTnEC) is output.</li> </ul> <p><b>Caution:</b> When TTnSCE = 1, be sure to set bits TTnUDS1, and TTnUDS0 of the TTnCTL2 register to 10<sub>B</sub> or 11<sub>B</sub>.</p>
6	TTnZCL	<p>Specifies the clear level for the Z phase of encoder input (TECRTn pin)</p> <p>0: Clear condition = Low level 1: Clear condition = High level</p> <p><b>Note:</b> The TTnZCL bit is valid when TTnSCE = 1.</p>
5	TTnBCL	<p>Specifies the clear level for the B phase of encoder input (TENCTn1 pin)</p> <p>0: Clear condition = Low level 1: Clear condition = High level</p> <p><b>Note:</b> The TTnBCL bit is valid when TTnSCE = 1.</p>
4	TTnACL	<p>Specifies the clear level for the Aphase of encoder input (TENCTn0 pin)</p> <p>0: Clear condition = Low level 1: Clear condition = High level</p> <p><b>Note:</b> The TTnACL bit is valid when TTnSCE = 1.</p>



**Table 18-12 TTnIOC3 register contents (2/2)**

Bit position	Bit name	Function															
3, 2	TTnECS1 TTnECS0	Specifies the encoder clear input (TECRTn pin) valid edge.															
		<table><tr><th>TTnECS1</th><th>TTnECS0</th><th>Valid edge of encoder clear input (TECRTn pin)</th></tr><tr><td>0</td><td>0</td><td>No edge detection</td></tr><tr><td>0</td><td>1</td><td>Rising edge detection</td></tr><tr><td>1</td><td>0</td><td>Falling edge detection</td></tr><tr><td>1</td><td>1</td><td>Both rising and falling edge detection</td></tr></table>	TTnECS1	TTnECS0	Valid edge of encoder clear input (TECRTn pin)	0	0	No edge detection	0	1	Rising edge detection	1	0	Falling edge detection	1	1	Both rising and falling edge detection
		TTnECS1	TTnECS0	Valid edge of encoder clear input (TECRTn pin)													
		0	0	No edge detection													
		0	1	Rising edge detection													
		1	0	Falling edge detection													
		1	1	Both rising and falling edge detection													
		The encoder clear interrupt (INTTTnEC) is output upon detection of the valid edge set with bits TTnECS1, TTnECS0.															
		<b>Caution:</b> When TTnSCE = 1, the encoder clear interrupt (INTTTnEC) is not output.															
		<b>Note:</b> Bits TTnECS1 and TTnECS0 are valid in the encoder compare mode, encoder capture mode, and encoder capture/compare mode and when TTnSCE = 0.															
1, 0	TTnEIS1 TTnEIS0	Specifies the encoder input signals (TENCTn1/TENCTn0 pins) valid edge.															
		<table><tr><th>TTnEIS1</th><th>TTnEIS0</th><th>Valid edge of the encoder input signal (TENCTn1/TENCTn0 pins)</th></tr><tr><td>0</td><td>0</td><td>No edge detection (capture operation invalid)</td></tr><tr><td>0</td><td>1</td><td>Rising edge detection</td></tr><tr><td>1</td><td>0</td><td>Falling edge detection</td></tr><tr><td>1</td><td>1</td><td>Both, rising and falling edge detection</td></tr></table>	TTnEIS1	TTnEIS0	Valid edge of the encoder input signal (TENCTn1/TENCTn0 pins)	0	0	No edge detection (capture operation invalid)	0	1	Rising edge detection	1	0	Falling edge detection	1	1	Both, rising and falling edge detection
		TTnEIS1	TTnEIS0	Valid edge of the encoder input signal (TENCTn1/TENCTn0 pins)													
		0	0	No edge detection (capture operation invalid)													
		0	1	Rising edge detection													
		1	0	Falling edge detection													
		1	1	Both, rising and falling edge detection													
		<b>Note:</b> Bits TTnEIS1 and TTnEIS0 are valid when bits TTnCTL2.TTnUDS[1:0] are 00 <sub>B</sub> or 01 <sub>B</sub> .															

**(8) TTnOPT0 - TMTn option register 0**

The TTnOPT0 register is an 8-bit register that sets the capture/compare operation and detects overflow.

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

**Address** TT0OPT0: FFFFF697<sub>H</sub>  
TT1OPT0: FFFFF6A7<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
TTnOPT0	0	0	TTnCCS1	TTnCCS0	0	0	0	TTnOVF
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Caution** Set the TTnOPT0 register when TTnCTL0.TTnCE = 0. When TTnCE = 1, write access to the TTnOPT0 register can be performed with the same value.

**Table 18-13 TTnOPT0 register contents**

Bit position	Bit name	Function
5	TTnCCS1	Specifies the operation mode of register TTnCCR1 0: Operation as compare register 1: Operation as capture register <b>Note:</b> The setting of bit TTnCCS1 is valid in the free-running mode only.
4	TTnCCS0	Specifies the operation mode of register TTnCCR0 0: Operation as compare register 1: Operation as capture register <b>Note:</b> The setting of bit TTnCCS0 is valid in the free-running mode only.
0	TTnOVF	Indicates TMTn overflow 0: No overflow occurrence after timer restart or flag reset 1: Overflow occurrence  In the free-running mode, pulse width measurement mode, and offset trigger generation mode, if the counter value is counted up from FFFFH, overflow occurs, the TTnOVF flag is set (1), and the counter is cleared to 0000H. The counter is also cleared by writing 0. At the same time that the TTnOVF flag is set (1), an overflow interrupt (INTTTnOV) occurs. If 0 is written to the TTnOVF flag, or if TTnECC = 0 and TTnCE = 0 are set, the counter is cleared. <b>Note:</b> Overflow does not occur during compare match & clear operation for counter value FFFFH and compare value FFFFH.  <b>Caution:</b> 1. If overflow occurs in the encoder compare mode, encoder capture mode, or encoder capture compare mode, the encoder-dedicated overflow flag (TTnEOF) is set, and the overflow flag (TTnOVF) is not set. At this time, the overflow interrupt (INTTTnOV) is output. 2. When TTnOVF = 1, the TTnOVF flag is not cleared even if the TTnOVF flag and TTnOPT0 register are read. 3. The TTnOVF flag can be read and written, but even if 1 is written to the TTnOVF flag from the CPU, this is invalid.

**(9) TTnOPT1 - TMTn option register 1**

The TTnOPT1 register is an 8-bit register that detects encoder-dedicated underflow, overflow, and counter up/down operation.

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

**Address** TT0OPT1: FFFFF698<sub>H</sub>  
TT1OPT1: FFFFF6A8<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>TTnOPT1</b>	0	0	0	0	0	TTnEUF	TTnEOF	TTnESF
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Note** The setting of the TTnOPT1 register is valid only in the encoder compare mode, encoder capture mode, and encoder capture/compare mode. (In all other modes, the setting value is invalid.)

**Table 18-14 TTnOPT1 register contents (1/2)**

Bit position	Bit name	Function
2	TTnEUF	<p>Indicates encoder underflow 0: No underflow indicated 1: Indicates counter underflow in the encoder compare mode, encoder capture mode, and encoder capture/compare mode.</p> <p>If the counter value is counted down from 0000H, underflow occurs, the OVF flag is set (to 1), and the counter is set to FFFFH. When the TTnEUF flag is set (to 1), an overflow interrupt (INTTTnOV) occurs at the same time.</p> <p>The TTnEUF flag is cleared (to 0) under the following conditions.</p> <ul style="list-style-type: none"> <li>• When 0 is written by CPU instruction</li> <li>• When TTnCTL0.TTnCE = 0 is set while TTnCTL2.TTnECC = 0</li> </ul> <p><b>Note:</b> When TTnCTL2.TTnECC bit is 1, the flag status is held even if the value of TTnCTL0.TTnCE bit is changed from 1 to 0.</p> <hr/> <p><b>Caution:</b> 1. The TTnEUF flag is not cleared even if it is read. 2. The TTnEUF flag can be read and written, but even if 1 is written to the TTnEUF flag, this is invalid.</p> <hr/>

Table 18-14 TTnOPT1 register contents (2/2)

Bit position	Bit name	Function
1	TTnEOF	<p>Indicates encoder overflow            0: No overflow indicated            1: Indicates counter overflow in the encoder compare mode, encoder capture mode, and encoder capture/compare mode.</p> <p>If the counter value is counted up from FFFFH, overflow occurs, the OVF flag is set (1), and the counter is cleared to 0000H. At the same time that the TTnEOF flag is set (1), an overflow interrupt (INTTTnOV) occurs. However, the TTnOVF flag is not set (to 1).</p> <p>The TTnEOF flag is cleared (0) under the following conditions.</p> <ul style="list-style-type: none"> <li>• When 0 is written by CPU instruction</li> <li>• When TTnCTL0.TTnCE = 0 is set while TTnCTL2.TTnECC = 0</li> </ul> <p><b>Note:</b> When TTnCTL2.TTnECC bit is 1, the flag status is held even if the value of TTnCTL0.TTnCE bit is changed from 1 to 0.</p> <hr/> <p><b>Caution:</b> 1. The TTnEOF flag is not cleared even if it is read.            2. The TTnEOF flag can be read and written, but even if 1 is written to the TTnEOF flag from the CPU, this is invalid.</p> <hr/>
0	TTnESF	<p>Indicates encoder count direction            0: Indicates the up count operation of the counter in the encoder compare mode, encoder capture mode, and encoder capture/compare mode.            1: Indicates the down count operation of the counter in the encoder compare mode, encoder capture mode, and encoder capture/compare mode.</p> <p>The TTnESF flag is cleared (to 0) under the following conditions.</p> <ul style="list-style-type: none"> <li>• When TTnCTL0.TTnCE = 0 is set while TTnCTL2.TTnECC = 0</li> </ul> <p><b>Note:</b> When TTnCTL2.TTnECC bit is 1, the flag status is held even if the value of TTnCTL0.TTnCE bit is changed from 1 to 0.</p>

**(10) TTnOPT2 - TMTn option register 2**

The TTnOPT2 register is an 8-bit register that indicates the reload request status when performing write access to compare registers using the reload method.

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

**Address** TT0OPT1: FFFFF699<sub>H</sub>  
TT1OPT1: FFFFF6A9<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
<b>TTnOPT2</b>	0	0	0	0	0	0	0	TTnRSF
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Note** The read contents of the TTnOPT2 register are valid only in the external trigger pulse mode, PWM mode, and offset trigger generation using the reload method. In all other modes, the read contents are 0.

**Table 18-15 TTnOPT2 register contents**

Bit position	Bit name	Function
2	TTnRSF	<p>Indicates the reload status</p> <p>0: No reload request, or reload completed</p> <p>1: Reload request was output</p> <p>This flag indicates that the data to be transferred next is held pending in the TTnCCR0 and TTnCCR1 registers.</p> <p>The TTnRSF flag is set (1) by writing to the TTnCCR1 register, and it is cleared (0) upon reload completion.</p> <hr/> <p><b>Caution:</b> When TTnRSF = 1, do not perform write access to the TTnCCR0 and TTnCCR1 registers.</p> <hr/>

## 18.5 Basic Operation

### 18.5.1 Basic counter operation

This section describes the basic operation of the counter. For details, refer to chapter “Operation in Each Mode” on page 970.

#### (1) Counter start operation

##### (a) Encoder compare mode, encoder capture mode, encoder capture compare mode

The count operation is controlled by the phases of pins TENCTn0 and TENCTn1.

When TTnCE = 0 and TTnECC = 0, the counter is initialized by the TTnTCW register and the count operation is started. (The setting value of the TTnTCW register is loaded to the counter at the timing when TTnCE changes from 0 to 1.)

##### (b) Triangular wave PWM MODE

The counter starts counting from initial value FFFFH.

It counts up FFFFH, 0000H, 0001H, 0002H, 0003H...

Following count up operation, the counter counts down upon a match with the TTnCCR0 register.

##### (c) Modes other than the above

The counter starts counting from initial value FFFFH.

It counts up FFFFH, 0000H, 0001H, 0002H, 0003H...

#### (2) Counter clear operation

There are the following five counter clear causes.

- Clear through match between counter value and compare setting value.
- Capture and clear through capture input
- Counter clear through encoder clear input (TECRTn pin)
- Counter clear through match with clear condition level
- Clear through clear signal input (TTnSYCI) for synchronization function during slave operation

**Table 18-16 Counter Clear Operation (1/2)**

Operation mode	Clear cause		
	TTnCCR0	TTnCCR1	Other
Interval mode	Compare match	-	-
External event count mode	Compare match	-	-
External trigger pulse output mode	Compare match	-	External trigger (TTRGTn pin)
One-shot pulse mode	Compare match	-	-
PWM mode	Compare match	-	-

**Table 18-16 Counter Clear Operation (2/2)**

Operation mode	Clear cause		
	TTnCCR0	TTnCCR1	Other
Free-running mode	-	-	-
Pulse width measurement mode	-	-	External input (TITn0 and TITn1 pins)
Triangular wave PWM mode	Compare match	-	-
Encoder compare mode	Depends on set conditions <sup>Note</sup>	Depends on set conditions <sup>Note</sup>	Pin TECRTn, clear condition level match
Encoder capture mode	-	-	Pin TECRTn, clear condition level match
Encoder capture compare mode	Depends on set conditions <sup>Note</sup>	-	Pin TECRTn, clear condition level match
Offset trigger generation mode	-	-	External input (TITn0 pin)

**Note** Conditions are set with bits TTnECM0 and TTnECM1 of the TTnCTL2 register.

### (3) Counter reset and hold operations

In the encoder compare mode, encoder capture mode, and encoder capture/compare mode, counter value hold is controlled with bit TTnECC of the TTnCTL2 register.

If TTnCE = 0 is set when TTnECC = 0, the counter is reset to 0000H. The setting value of the TTnTCW register is loaded to the counter when TTnCE = 1 is set next.

If TTnCE = 0 is set when TTnECC = 1, the counter value is held as is. Counting resumes from the held value when TTnCE = 1 is set next.

### (4) Counter read operation during counter operation

In TMT, the counter value can be read during count operation using the TTnCNT register.

**(5) Overflow operation**

Counter overflow occurs in the free-running mode, pulse width measurement mode, encoder compare mode, encoder capture mode, encoder capture/compare mode, and offset trigger generation mode.

Overflow occurs when the counter value changes from FFFFH to 0000H.

In the free-running mode, pulse width measurement mode, offset trigger generation mode, the overflow flag (TTnOVF) is set to 1 and an overflow interrupt (INTTTnOV) is output. At this time, the TTnEOF flag is not set.

In the encoder compare mode, encoder capture mode, and encoder capture/compare mode, the encoder dedicated overflow flag (TTnEOF) is set to 1 and an overflow interrupt (INTTTnOV) occurs. At this time, the TTnOVF flag is not set.

Under the following conditions, overflow does not occur.

- When the counter value changes from initial setting FFFFH to 0000H immediately after counting start
- When FFFFH is set to the compare register, and the counter is cleared to 0000H upon a match between the counter value and the compare setting value.
- When, in the pulse width measurement mode and offset trigger generation mode, capture operation is performed for counter value FFFFH, and the counter is cleared to 0000H.

**(6) Underflow operation**

Counter underflow occurs in the triangular wave PWM Mode, encoder compare mode, encoder capture mode, and encoder capture/compare mode.

Underflow occurs when the counter value changes from 0000H to FFFFH.

When underflow occurs in the triangular wave PWM mode, an overflow interrupt (INTTTnOV) occurs. At this time, the TTnOVF flag is not set.

In the encoder compare mode, encoder capture mode, and encoder capture/compare mode, the encoder dedicated underflow flag (TTnEUF) is set to 1, and an overflow interrupt (INTTTnOV) occurs.

Underflow does not occur during count down immediately following counter start.



**(7) Description of interrupt signal operation**

In TMT, the following interrupt signals are output.

Name	Occurrence cause
INTTTnCC0	<ul style="list-style-type: none"> <li>Match between counter and setting value of TTnCCR0 register</li> <li>Capture to TTnCCR0 register due to TITn0 pin input</li> </ul>
INTTTnCC1	<ul style="list-style-type: none"> <li>Match between counter and setting value of TTnCCR1 register</li> <li>Capture to TTnCCR1 register due to TITn1 pin input</li> </ul>
INTTTnOV	Overflow and underflow occurrence
INTTTnEC <sup>Note</sup>	Counter clearing through TECRTn pin

**Note** In the encoder compare mode, encoder capture mode, and encoder capture/compare mode, when TTnSCE = 0, an encoder clear interrupt (INTTTnEC) is output.

## 18.5.2 Method for writing to compare register

The TTnCCR0 and TTnCCR1 registers can be rewritten during timer operation (TTnCE = 1). There are two write modes (anytime write, reload), depending on the mode.

### (1) Anytime rewrite method

When the TTnCCR0 and TTnCCR1 registers are written during timer operation, the write value is immediately transferred to the TTnCCR0 buffer register and TTnCCR1 buffer register and is used as the value to be compared with the counter.

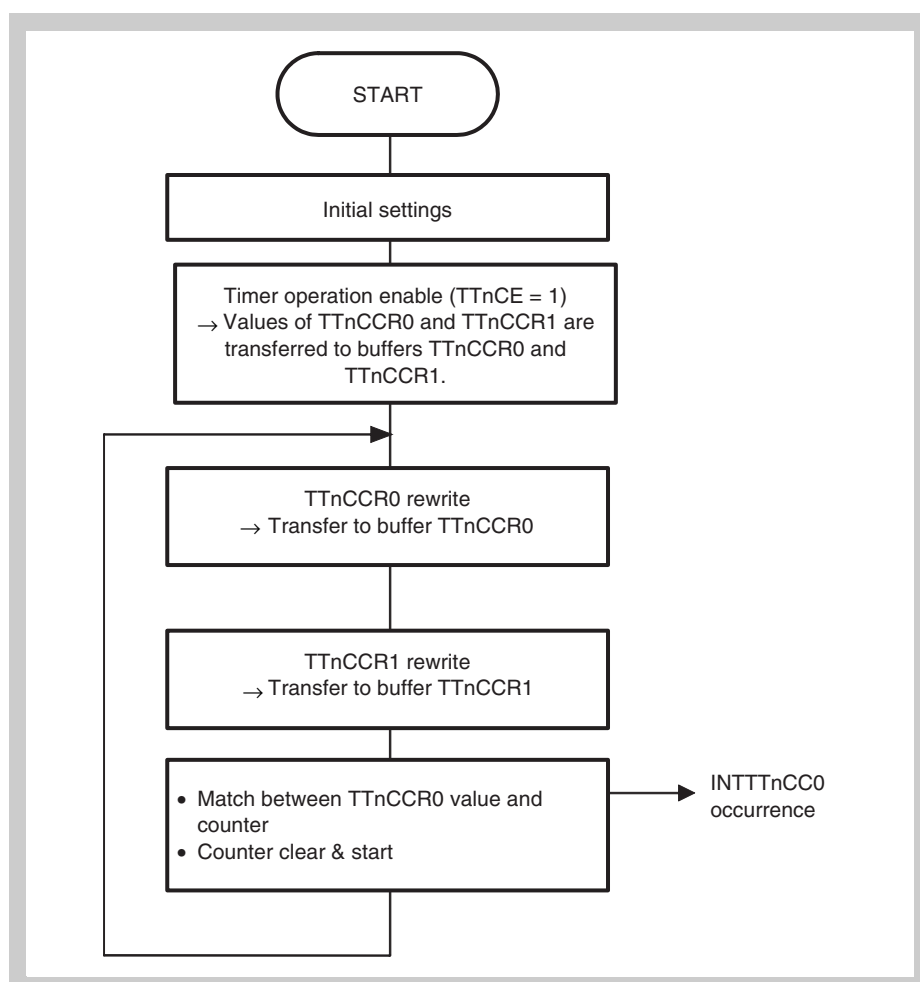
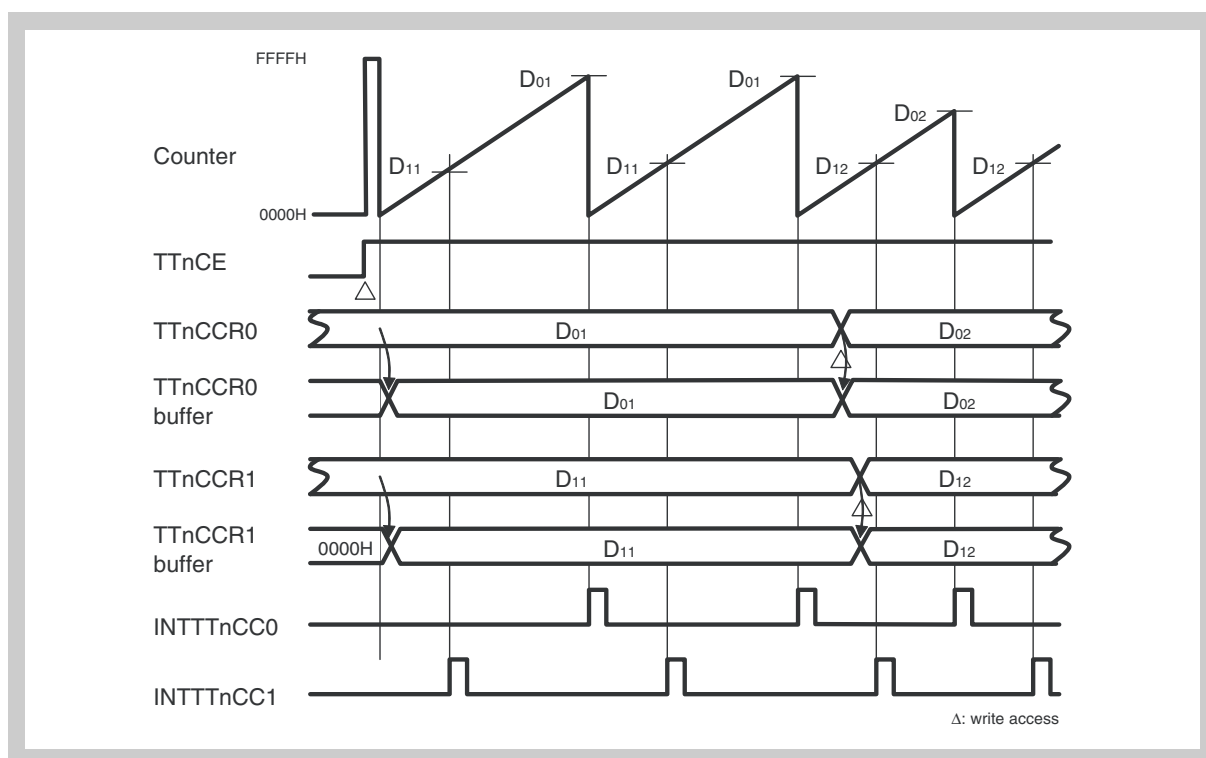


Figure 18-2 Basic operation flow for anytime rewrite

**Note** The interval mode is used as an example.



**Figure 18-3 Basic anytime rewrite operation timing**

- Note**
1. D<sub>01</sub>, D<sub>02</sub>: Setting values of TTnCCR0 register (0000H to FFFFH)  
D<sub>11</sub>, D<sub>12</sub>: Setting values of TTnCCR1 register (0000H to FFFFH)
  2. The interval mode is used as an example.

**(2) Reload method (Batch rewrite)**

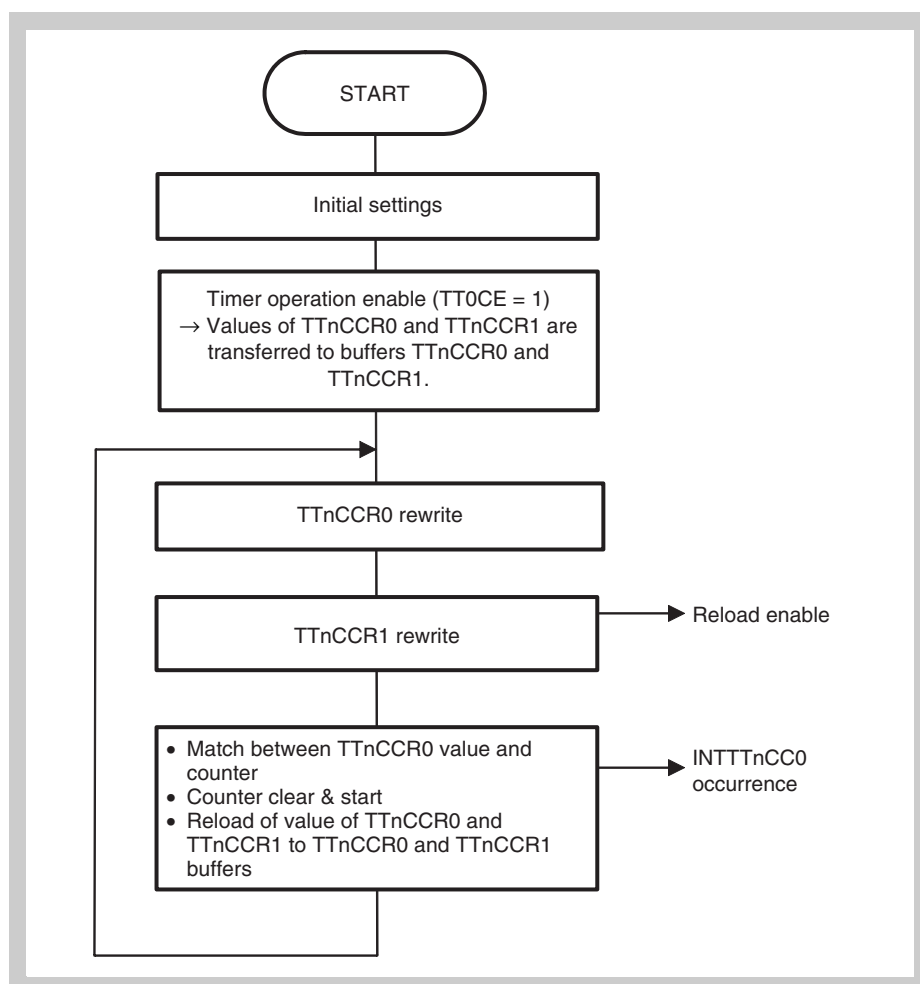
When TTnCCR0, TTnCCR1 register write is performed during timer operation, the written value is used as the comparison value for the counter via the TTnCCR0 and TTnCCR1 buffer registers.

Under the reload method, rewrite the TTnCCR0 register before the TTnCCR0 register value is matched, and next, write to the TTnCCR1 register.

Then, when the TTnCCR0 register is matched or the counter is cleared to 0000H through external input, the values of the TTnCCR0 register and TTnCCR1 register are reloaded.

By writing to the TTnCCR1 register, the value becomes valid at the next reload timing.

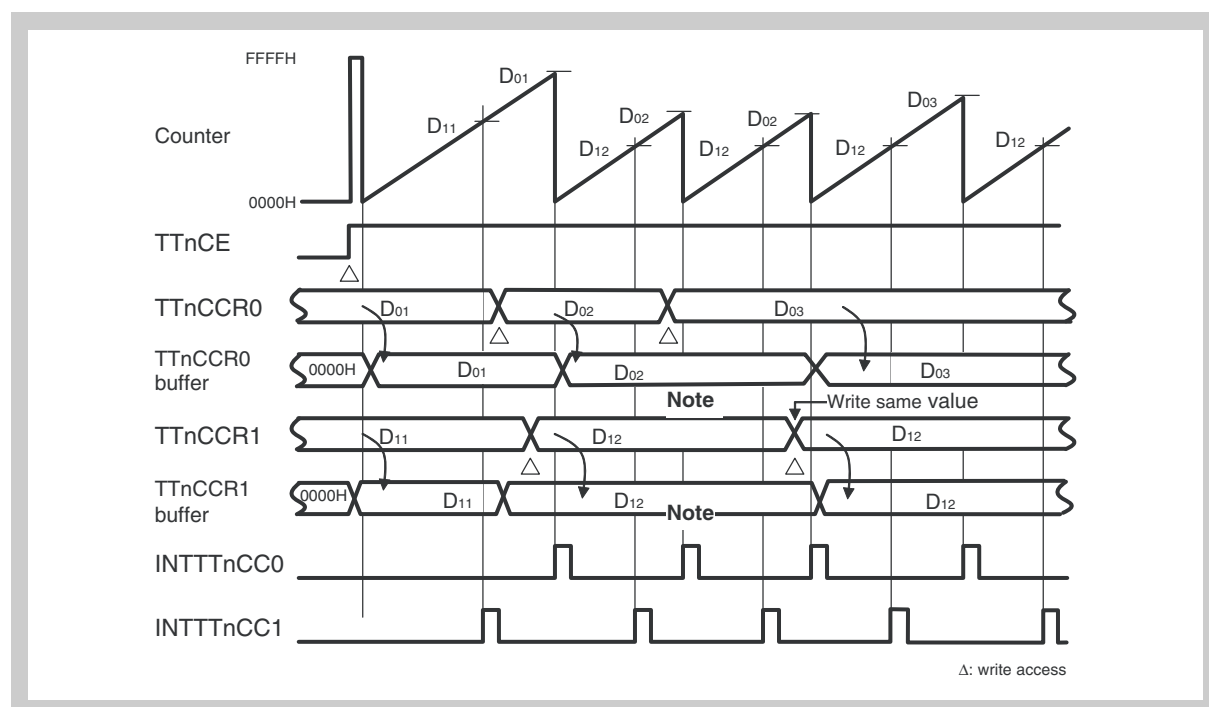
Therefore, even if wishing to rewrite only the value of the TTnCCR0, rewrite the same value to the TTnCCR1 register to make the next reload valid.



**Figure 18-4 Basic operation flow for reload (batch rewrite)**

**Caution** Rewrite to the TTnCCR1 register includes enabling reload. Therefore, rewrite the TTnCCR1 register after rewriting the TTnCCR0 register.

**Note** The PWM mode is used as an example.



**Figure 18-5 Basic reload operation timing**

- Note**
1. Since the TTnCCR1 register is not written to, reloading is not performed even if TTnCCR0 is rewritten.
  2. D<sub>01</sub>, D<sub>02</sub>, D<sub>03</sub>: Setting values of TTnCCR0 register (0000H to FFFFH)  
D<sub>11</sub>, D<sub>12</sub>: Setting values of TTnCCR1 register (0000H to FFFFH)
  3. The PWM mode is used as an example.

**Table 18-17 Capture/compare rewrite methods in each mode**

Operation mode	Capture/compare rewrite method	
	TTnCCR0	TTnCCR1
Interval mode	Compare only (Anytime write type)	
External event count mode		
External trigger pulse output mode	Compare only (Reload type)	
One-shot pulse mode	Compare only (Anytime write type)	
PWM mode	Compare only (Reload type)	
Free-running mode	Capture/compare selectable (When compare is selected, anytime write type)	
Pulse width measurement mode	Capture only	
Triangular wave PWM mode	Compare only (Reload type)	
Encoder compare mode	Compare only (Anytime write type)	
Encoder capture mode	Capture only	
Encoder capture compare mode	Compare only (Anytime write type)	Capture only
Offset trigger generation mode	Capture only	Compare only (Reload type)

## 18.6 Operation in Each Mode

### 18.6.1 Interval timer mode

In the interval timer mode, a compare match interrupt (INTTTnCC0) occurs and the counter is cleared upon a match between the setting value of the TTnCCR0 register and the counter value. The occurrence interval for this counter and TTnCCR0 register match interrupt becomes the interval time.

In the interval timer mode, the counter is cleared only upon a match between the counter and the value of the TTnCCR0 register. Counter clearing using the TTnCCR1 register is not performed.

However, the setting value of the TTnCCR1 is compared to the counter value transferred to the TTnCCR1 buffer register and a compare match interrupt (INTTTnCC1) is output.

The TTnCCR0 and TTnCCR1 registers can be rewritten using the anytime write method, regardless of the value of bit TTnCE.

Pins TOTn0 and TOTn1 are toggle output controlled when bits TTnOE0 and TTnOE1 are set to 1.

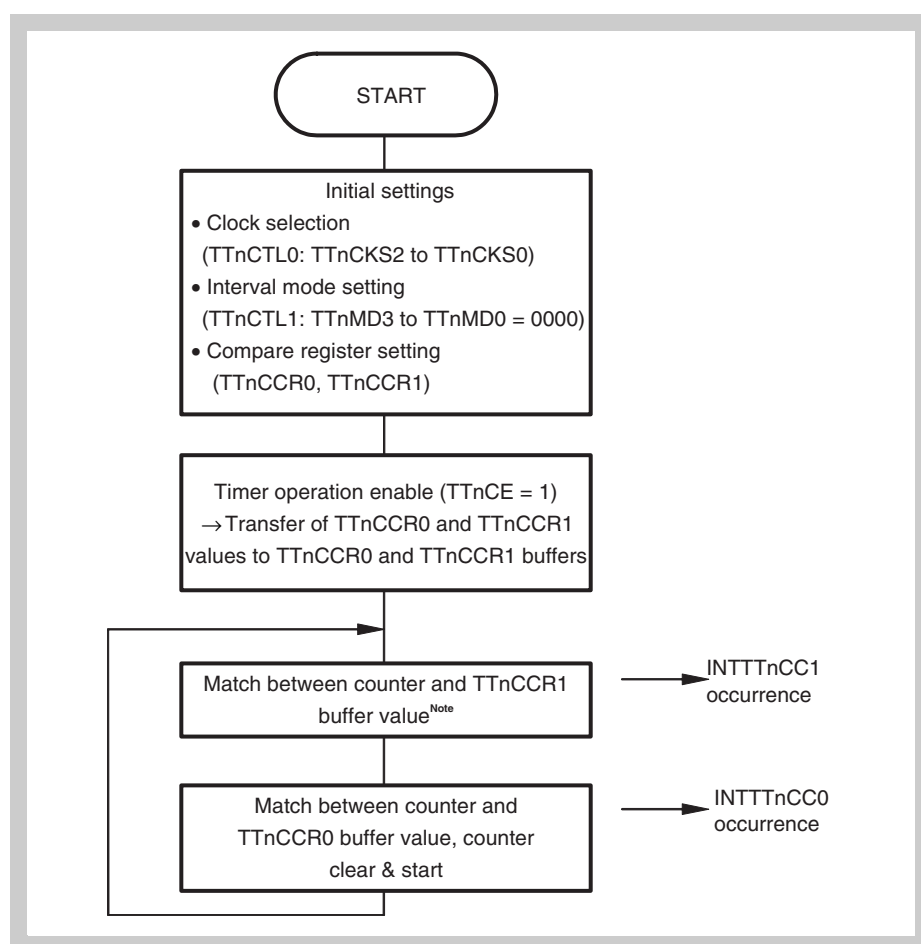
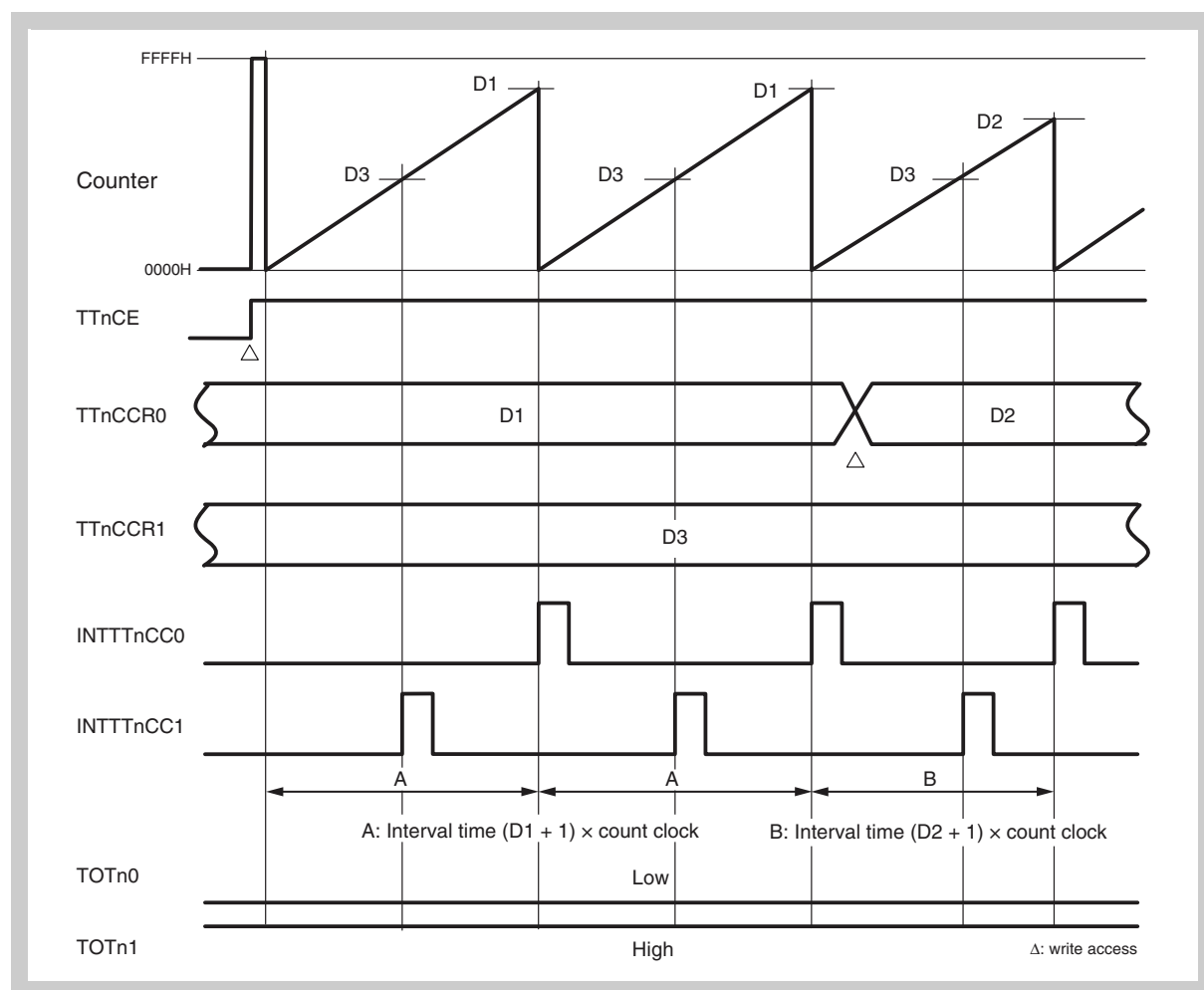


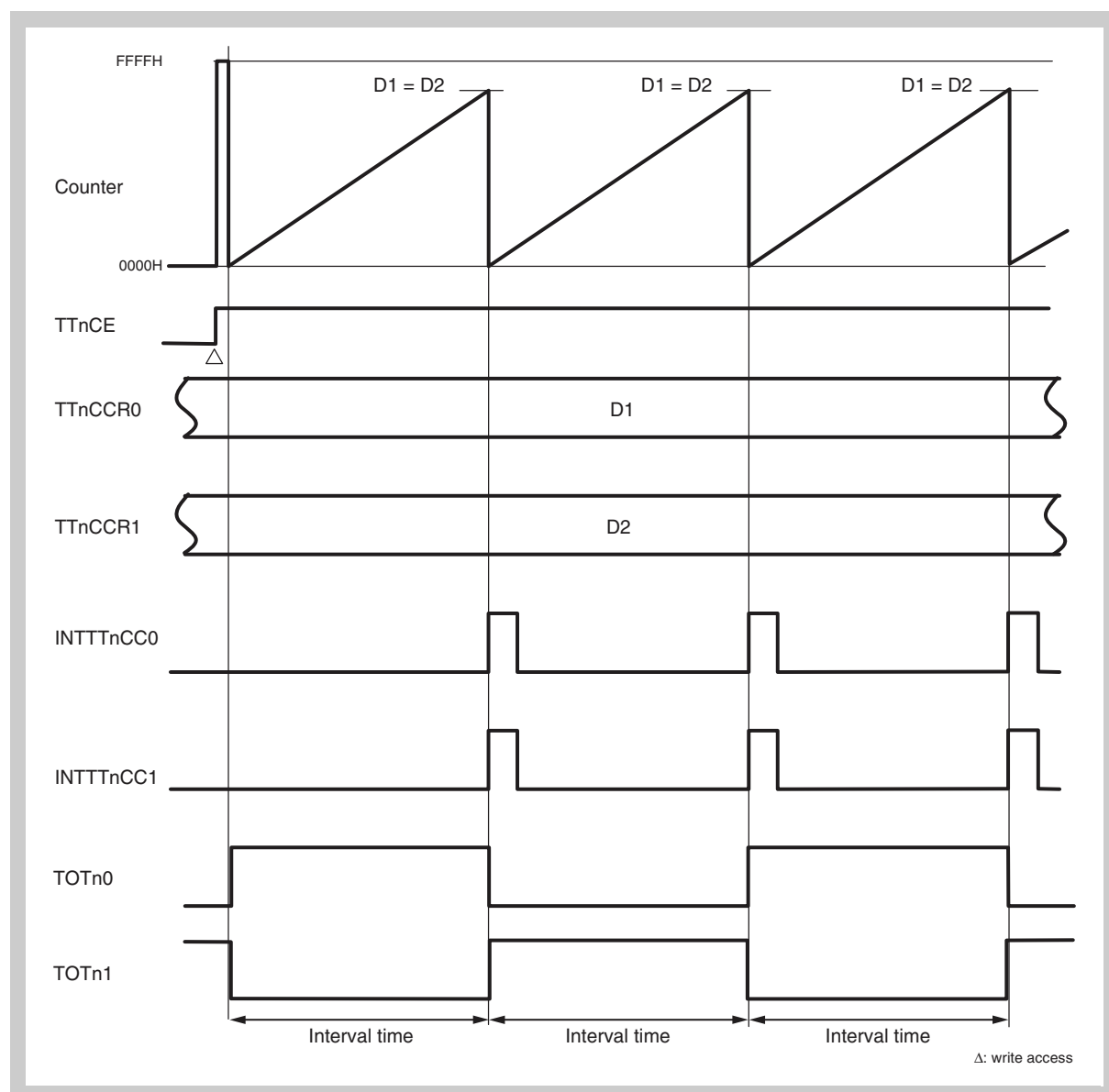
Figure 18-6 Basic operation flow in interval timer mode

**Note** In the case of a match between the counter and TTnCCR1 register, the counter is not cleared.



**Figure 18-7 Basic timing in interval timer mode (1/2)**  
 When  $D1 > D2 > D3$ , only value of TTnCCR0 register is rewritten, TOTn0 and TOTn1 are not output (TTnOE0, 1 = 0, TTnOL0 = 0, TTnOL1 = 1)

- Note**
1. D1, D2: Setting values of TTnCCR0 register (0000H to FFFFH)  
 D3: Setting values of TTnCCR1 register (0000H to FFFFH)
  2. Interval time =  $(Dm + 1) \times (\text{count clock cycle})$
  3.  $m = 1 \text{ to } 3$



**Figure 18-8 Basic timing in interval timer mode (2/2)**  
 When  $D1 = D2$ , values of  $TTnCCR0$  and  $TTnCCR1$  registers not rewritten,  
 $TOTn1$  output performed ( $TTnOE0, 1 = 1, TTnOL0 = 0, TTnOL1 = 1$ )

- Note**
1.  $D1$ : Setting value of  $TTnCCR0$  register (0000H to FFFFH)  
 $D2$ : Setting value of  $TTnCCR1$  register (0000H to FFFFH)
  2. Interval time =  $(Dm + 1) \times (\text{count clock cycle})$
  3.  $TOTn0, TOTn1$  toggle time =  $(Dm + 1) \times (\text{count clock cycle})$
  4.  $m = 1, 2$



### 18.6.2 External event count mode

In the external event count mode, count up starts upon external event input (TEVTTn pin). (The external event input (TEVTTn) is used as the count clock, regardless of bit TTnEEE of the TTnCTL1 register.)

In the external event count mode, the counter is cleared only upon a match between the counter and the value of the TTnCCR0 register. Counter clearing using the TTnCCR1 register does not work.

However, the value of the TTnCCR1 register is transferred to the TTnCCR1 buffer register, compared to the counter value, and a compare match interrupt (INTTTnCC1) is output.

The TTnCCR0 and TTnCCR1 registers can be rewritten with the anytime write method, regardless of the value of bit TTnCE.

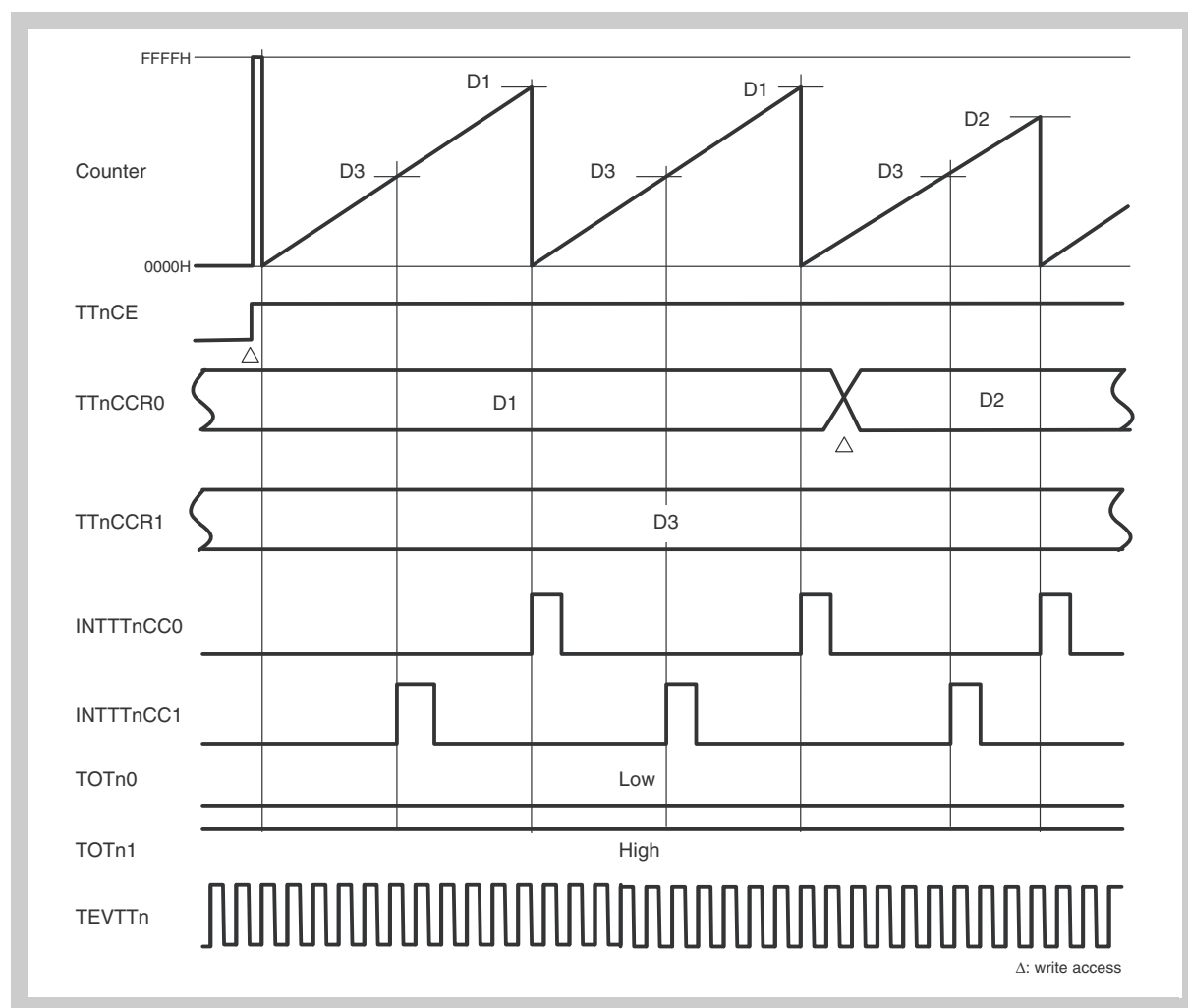
Pins TOTn0 and TOTn1 are toggle output controlled when bits TTnOE0 and TTnOE1 are set to 1.

When using only one compare register channel, it is recommended to set the TTnCCR1 register to FFFFH.

[External event count operation flow]

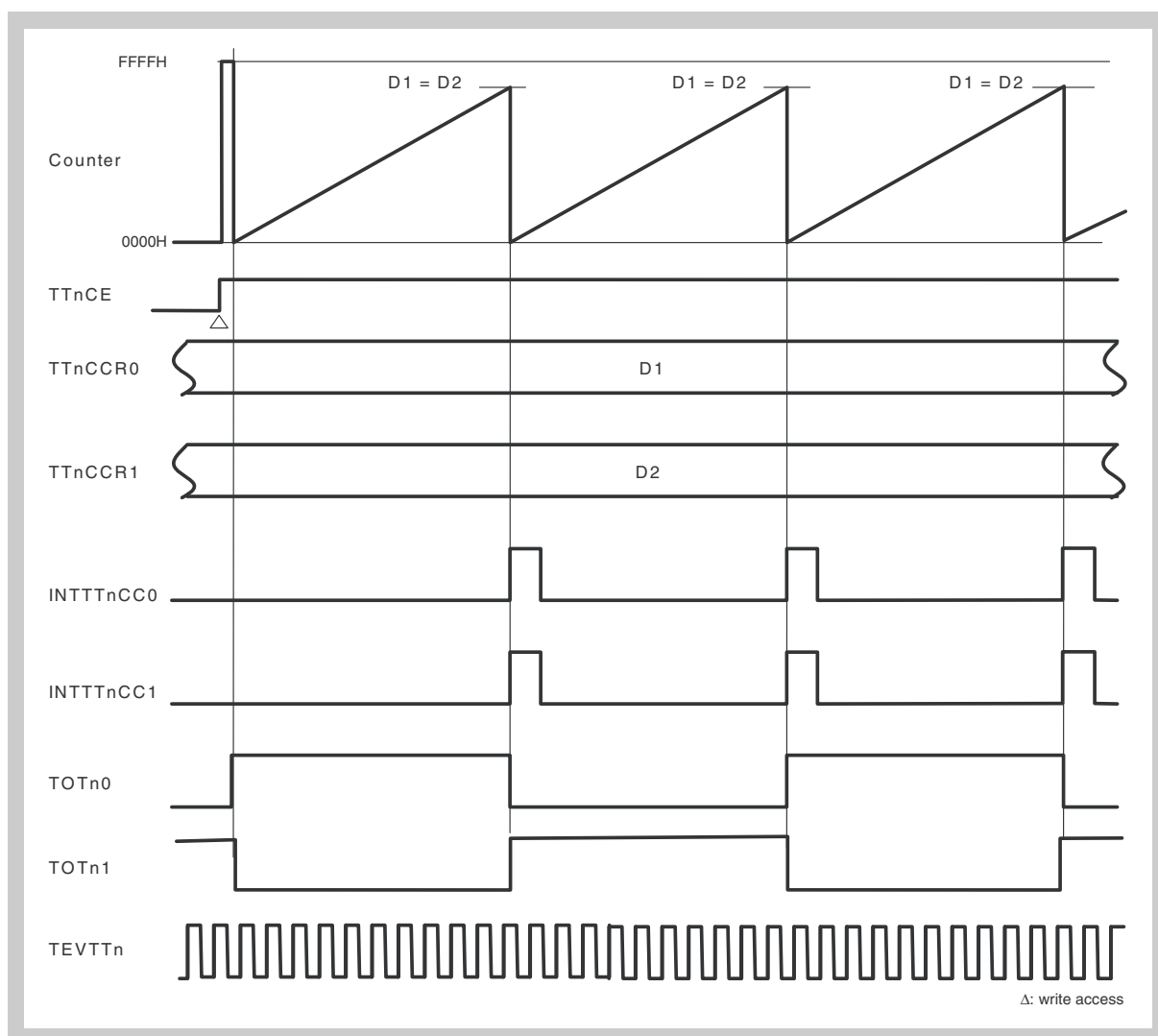
- <1> TTnCTL1 register bits TTnMD3 to TTnMD0 = 0001B (mode setting)  
Edge detection set with TTnIOC2 register bits TTnEES1 and TTnEES0 (TTnEES1, TTnEES0 = setting other than 01B)
- <2> TTnCTL0 register bit TTnCE = 1 (count enable)
- <3> TEVTTn pin input edge detection (count-up start)

- 
- Caution**
1. In external event count mode, when the setting value of the TTnCCR0 register is set to m, the number of TEVTTn pin input edge detection times is m+1.
  2. In external event count mode, do not send the TTnCCR0 register to 0000H.
-



**Figure 18-9 Basic operation timing in external event count mode (1/4)**  
 When  $D1 > D2 > D3$ , only value of TTnCCR0 register is rewritten, TOTn0 and TOTn1 are not output. The signal input from TEVTTn and internally synchronized is counted as the count clock (TTnOE0, 1 = 0, TTnOL0 = 0, TTnOL1 = 1)

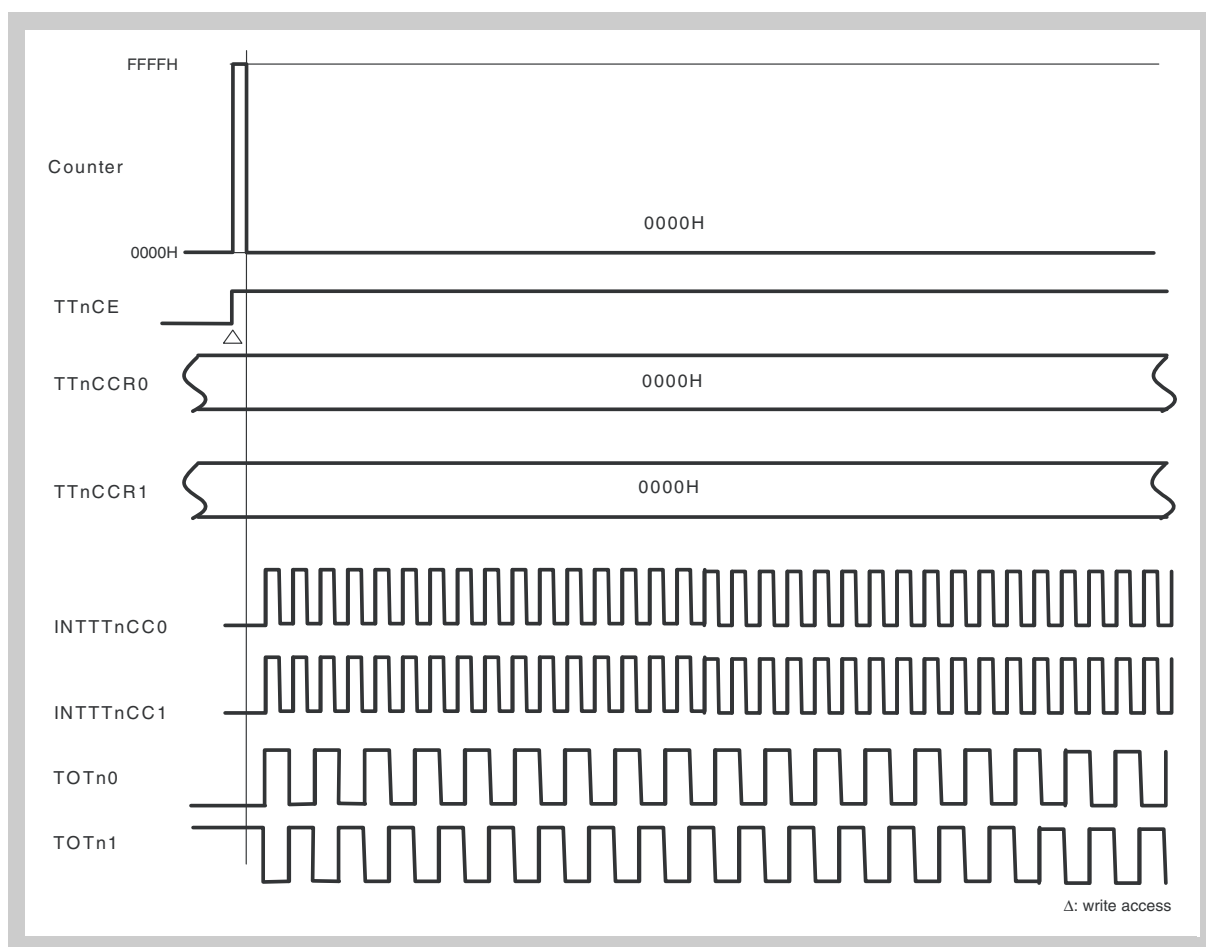
- Note**
1. D1, D2: Setting values of TTnCCR0 register (0000H to FFFFH)  
 D3: Setting value of TTnCCR1 register (0000H to FFFFH)
  2. Number of event counts =  $(Dm + 1)$  ( $m = 1, 2$ )



**Figure 18-10** Operation timing in external event count mode (2/4)

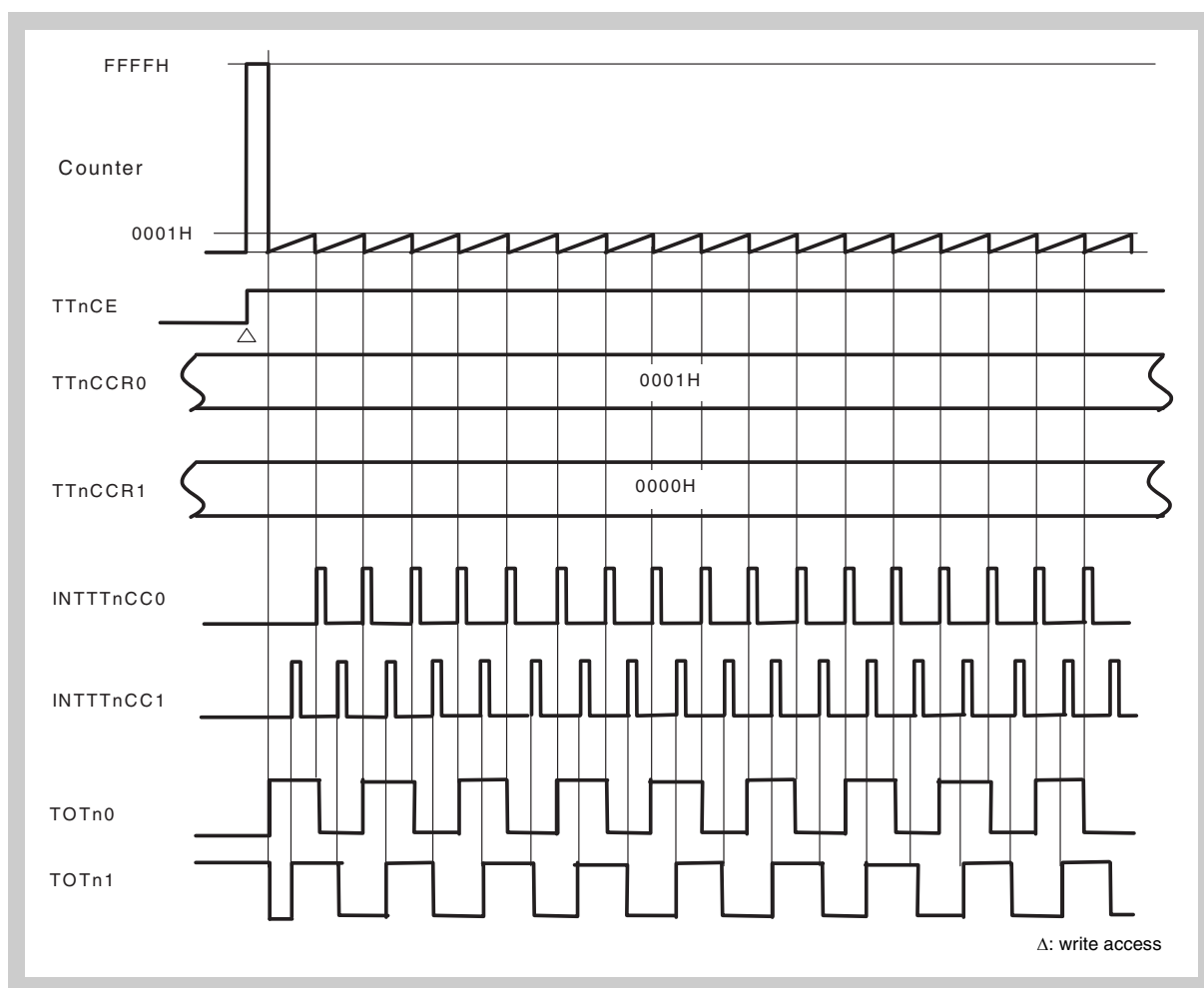
When  $D1 = D2$ , TTnCCR0 and TTnCCR1 register values are not rewritten, TOTn0 and TOTn1 are output (TTnOE0, 1 = 1, TTnOL0 = 0, TTnOL1 = 1)

- Note**
1. D1: Setting value of TTnCCR0 register (0000H to FFFFH)  
D2: Setting value of TTnCCR1 register (0000H to FFFFH)
  2. Number of event counts =  $(Dm + 1)$  ( $m = 1, 2$ )



**Figure 18-11** Operation timing in external event count mode (3/4)  
 When D1 = D2, TTnCCR0 and TTnCCR1 register values are not rewritten,  
 TOTn0 and TOTn1 are output (TTnOE0, 1 = 1, TTnOL0 = 0, TTnOL1 = 1)

- Note**
1. D1: Setting value of TTnCCR0 register (0000H)  
 D2: Setting value of TTnCCR1 register (0000H)
  2. Number of event counts = (Dm + 1) (m = 1, 2)



**Figure 18-12 Basic operation timing in external event count mode (4/4)**  
 When D1 = D2, TTnCCR0, TTnCCR1 register values are not rewritten,  
 TOTn0 and TOTn1 are output (TTnOE0, 1 = 1, TTnOL0 = 0, TTnOL1 = 1)

- Note**
1. D1: Setting value of TTnCCR0 register (0001H)  
 D2: Setting value of TTnCCR1 register (0000H)
  2. Number of event counts = (Dm + 1) (m = 1, 2)

### 18.6.3 External trigger pulse output mode

When, in the external trigger pulse mode, the duty is set to the TTnCCR1 register, the cycle is set to the TTnCCR0 register, and TTnCE = 1 is set, external trigger input (TTRGTn pin) wait results, with the counter remaining stopped at FFFFH. Upon detection of the valid edge of external trigger input (TTRGTn pin), or when the TTnEST bit of the TTnCTL1 register is set, count up starts.

An external trigger pulse is output from pin TOTn1, and toggle output is performed from pin TOTn0 upon a match with the TTnCCR0 register.

Moreover, during the count operation, upon a match between the counter and the TTnCCR0 register, a compare match interrupt (INTTTnCC0) is output, and upon a match between the counter and TTnCCR1 register, a compare match interrupt (INTTTnCC1) is output.

The TTnCCR0 and TTnCCR1 registers can be rewritten during count operation. Compare register reload is performed at the timing when the counter value and the TTnCCR0 register match.

However, when write access to the TTnCCR1 register is performed, the next reload timing becomes valid, so that even if wishing to rewrite only the value of the TTnCCR0 register, write the same value to the TTnCCR1 register. In this case, reload is not performed even if only the TTnCCR0 register is rewritten.

If, during operation in the external trigger pulse output mode, the external trigger (TTRGTn pin) edge is detected several times, or if the TTnEST bit of the TTnCTL1 register is set (to 1), the counter is cleared and count up is resumed.

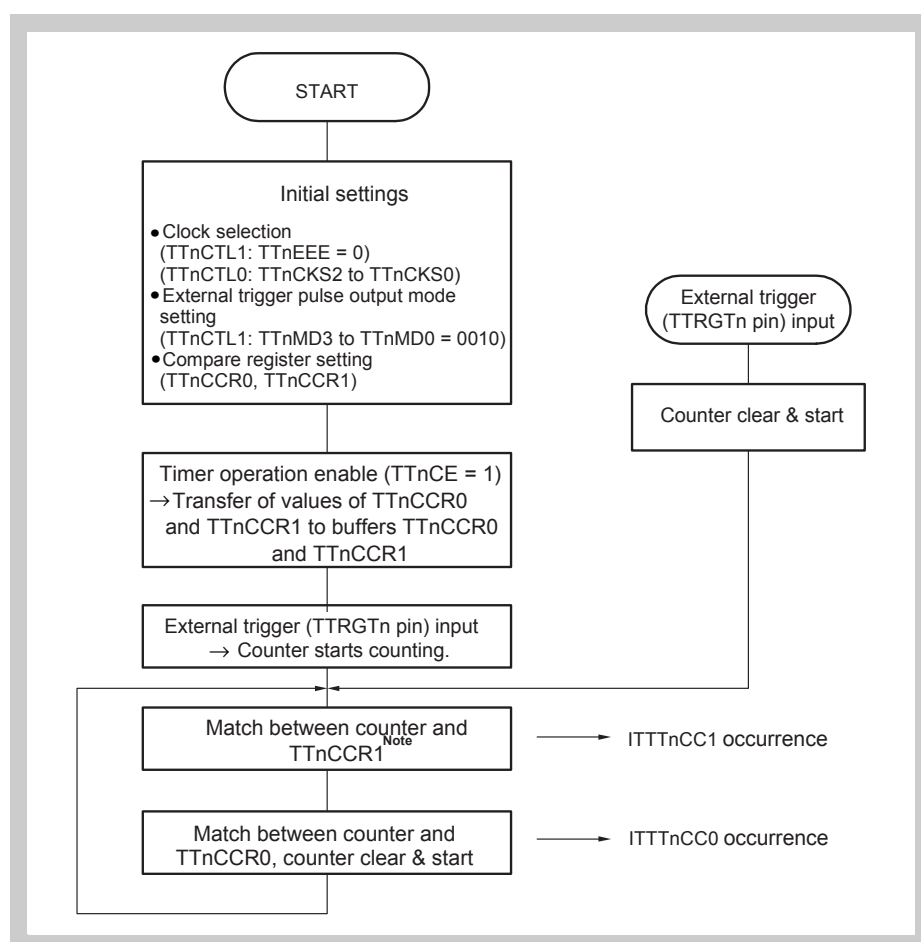
Moreover, if at this time, the TOTn1 pin is in the low level status, the TOTn1 pin output becomes high level when an external trigger is input. If the TOTn1 pin is in the high level status, it remains high level even if external trigger input occurs.

In the external trigger pulse output mode, the TTnCCR0 and TTnCCR1 registers have their function fixed as compare registers, so the capture function cannot be used.

---

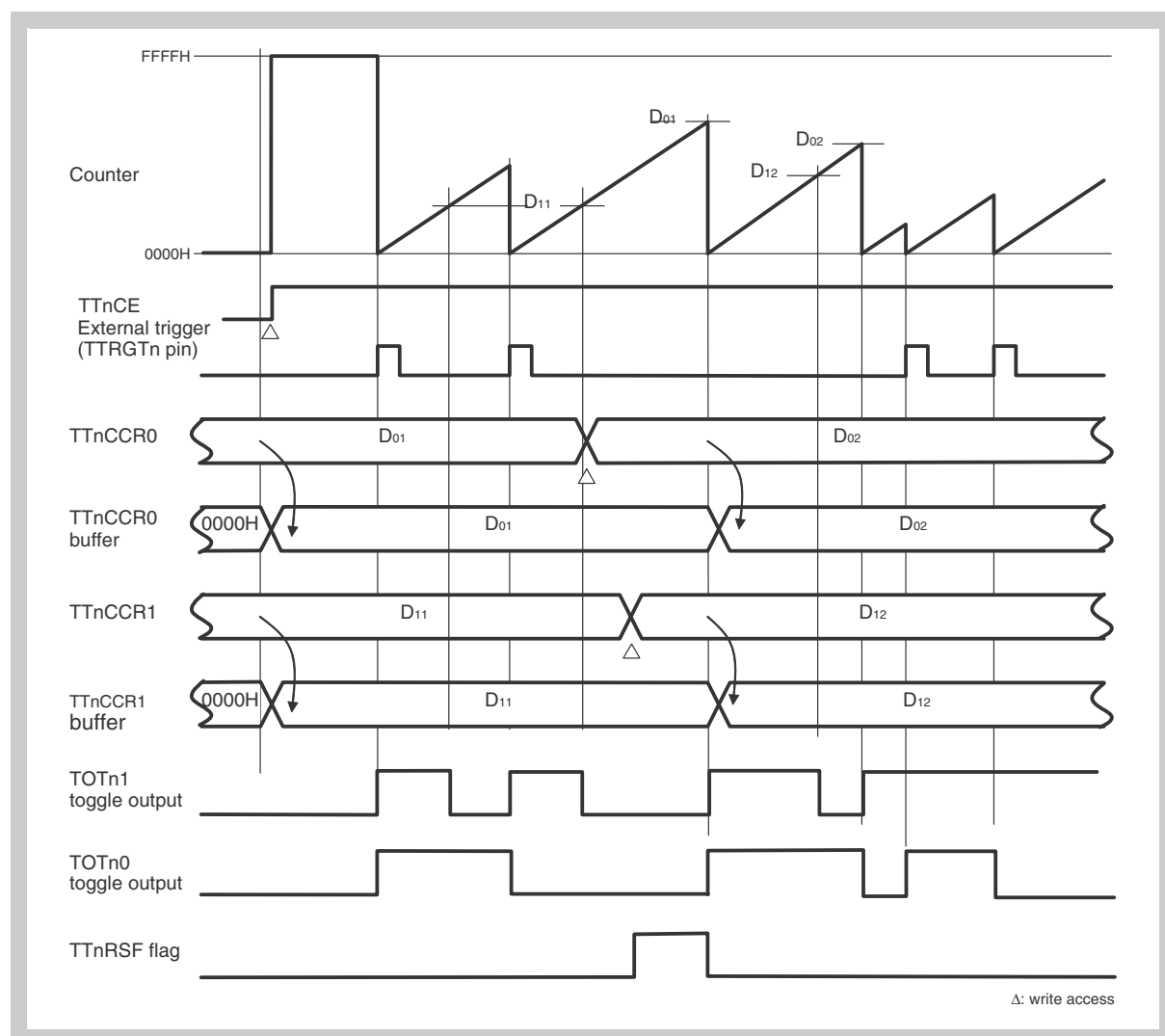
**Caution** In the external trigger pulse mode, set bit TTnEEE of the TTnCTL1 register to 0.

---



**Figure 18-13 Basic operation flow in external trigger pulse output mode**

**Note** The counter is not cleared upon a match between the counter and the TTnCCR1 buffer register.



**Figure 18-14 Basic operation timing in external trigger pulse output mode**  
 When values of TnCCR0 and TnCCR1 registers are rewritten, TOTn0 and TOTn1 are output (TnOE0, 1 = 1, TnOL0, 1 = 0)

- Note**
1. D<sub>01</sub>, D<sub>02</sub>: Setting values of TnCCR0 register (0000H to FFFFH)  
 D<sub>11</sub>, D<sub>12</sub>: Setting values of TnCCR1 register (0000H to FFFFH)
  2. TOTn1 (PWM) duty = (setting value of TnCCR1 register) × (count clock cycle)  
 TOTn1 (PWM) cycle = (setting value of TnCCR0 register + 1) × (count clock cycle)
  3. Pin TOTn0 is toggled when the counter is cleared immediately following count start.



### 18.6.4 One-shot pulse mode

When, in the one-shot pulse mode, the duty is set to the TTnCCR0 register, the output duty delay value is set to the TTnCCR1 register, and bit TTnCE of the TTnCTL0 register is set to 1, external trigger input (TTRGTn pin) wait results, with the counter remaining stopped at FFFFH. Upon detection of the valid edge of external trigger input (TTRGTn pin), or when bit TTnEST of the TTnCTL0 register is set to 1, count up starts. The TOTn1 pin becomes high level upon a match between the counter and TTnCCR1 register.

Moreover, upon a match between the counter and TTnCCR0 register, the TOTn1 pin becomes low level, and the counter is cleared to 0000H and then stops. The TOTn0 pin performs toggle output during the count operation upon a match between the counter and the TTnCCR0 buffer register.

Moreover, upon a match between the counter and TTnCCR0 register during count operation, a compare match interrupt (INTTTnCC0) is output, and upon a match between the counter and TTnCCR1 buffer register, a compare match interrupt (INTTTnCC1) is output.

The TTnCCR0 and TTnCCR1 registers can be rewritten using the anytime write method, regardless of the value of bit TTnCE.

Even if a trigger is input during the counter operation, it is ignored. Be sure to input the second trigger when the counter is stopped at 0000H.

In the one-shot pulse mode, registers TTnCCR0 and TTnCCR1 have their function fixed as compare registers, so the capture function cannot be used.

[One-shot pulse operation flow]

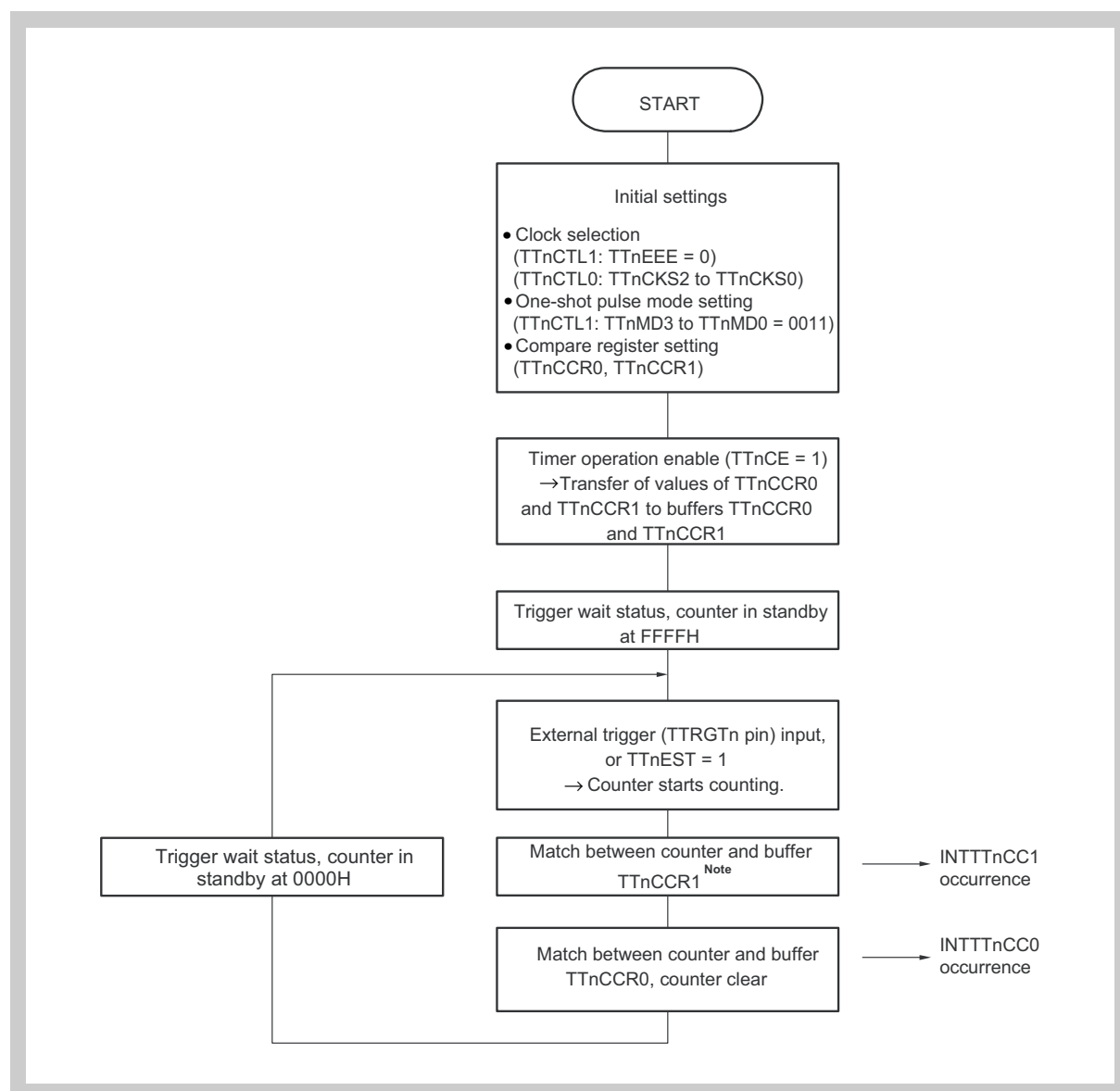
- <1> TTnCTL1 register bits TTnMD3 to TTnMD0 = 0011B (One-shot pulse mode)
- <2> TTnCCR0 register setting (duty setting),  
TTnLOC0 register bit TTnOE1 = 1 (TOTn1 pin output enable)
- <3> TTnCTL0 register bit TTnCE = 1 (counter operation enable):  
TOTn1 = Low-level output
- <4> TTnCTL1 register bit TTnEST = 1 or TTRGTn pin edge detection (count-up start):  
TOTn1 = Low-level output
- <5> Match between counter value and TTnCCR1 buffer register:  
TOTn1 = High-level output
- <6> Match between counter value and TTnCCR0 buffer register:  
TOTn1 = Low-level output, count clear
- <7> Count stop: TOTn1 = Low-level output
- <8> TTnCE = 0 (operation reset)

<1> to <2> can be in any order.

---

**Caution** In the one-shot pulse mode, set bit TTnEEE of the TTnCTL1 register to 0.

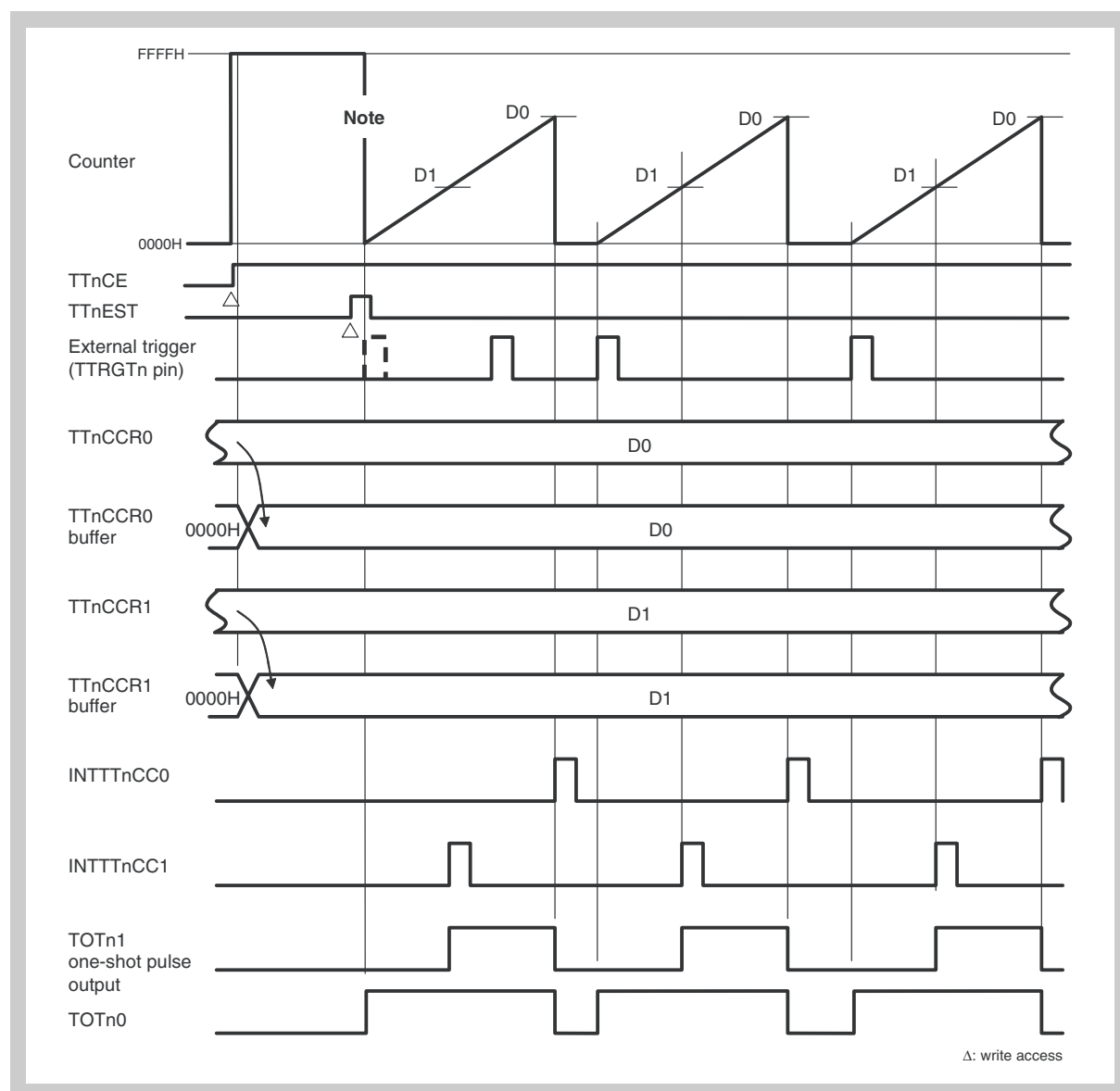
---



**Figure 18-15 Basic operation flow in one-shot pulse mode**

**Note** The counter is not cleared upon a match between the counter and the TTnCCR1 buffer register.

**Caution** The counter is not cleared even if trigger input is realized while the counter counts up, and the trigger input is ignored.



**Figure 18-16 Basic operation timing in one-shot pulse mode**  
(TTnOE0, 1 = 1, TTnOL0, 1 = 0)

- Note**
1. Count up starts when the value of TTnEST becomes 1 or TTRGTn is input.
  2. D0: Setting value of TTnCCR0 register (0000H to FFFFH)  
D1: Setting value of TTnCCR1 register (0000H to FFFFH)
  3.  $TOTn1 \text{ (output delay)} = (\text{setting value of TTnCCR1 register}) \times (\text{count clock cycle})$   
 $TOTn1 \text{ (output pulse width)} = \{(\text{setting value of TTnCCR0 register} + 1) - (\text{setting value of TTnCCR1 register})\} \times (\text{count clock cycle})$

### 18.6.5 PWM mode

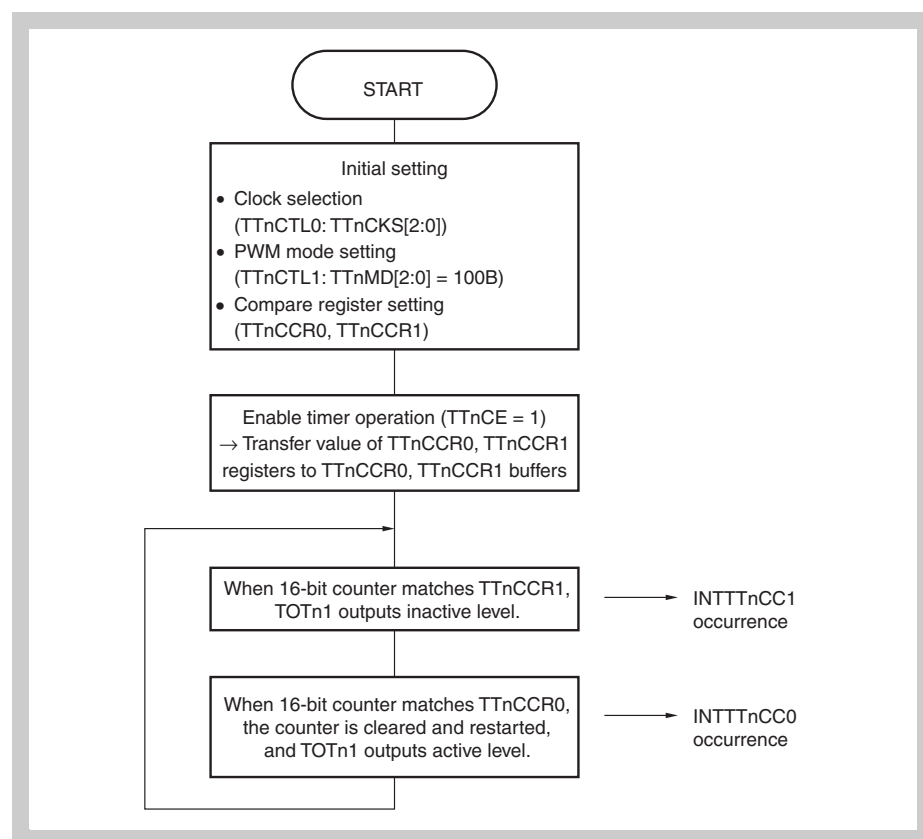
When, in the PWM mode, the duty is set to the TTnCCR1 register, the cycle is set to the TTnCCR0 register, and TTnCE = 1 is set, variable duty PWM output is performed from pin TOTn1.

Simultaneously with the start of count up operation, pin TOTn1 becomes active level, and upon a match between the counter and the TTnCCR1 register, becomes inactive level. Next, the TOTn1 pin becomes active level upon a match with the TTnCCR0 register. The TOTn0 pin performs toggle output upon a match with the TTnCCR0 buffer register.

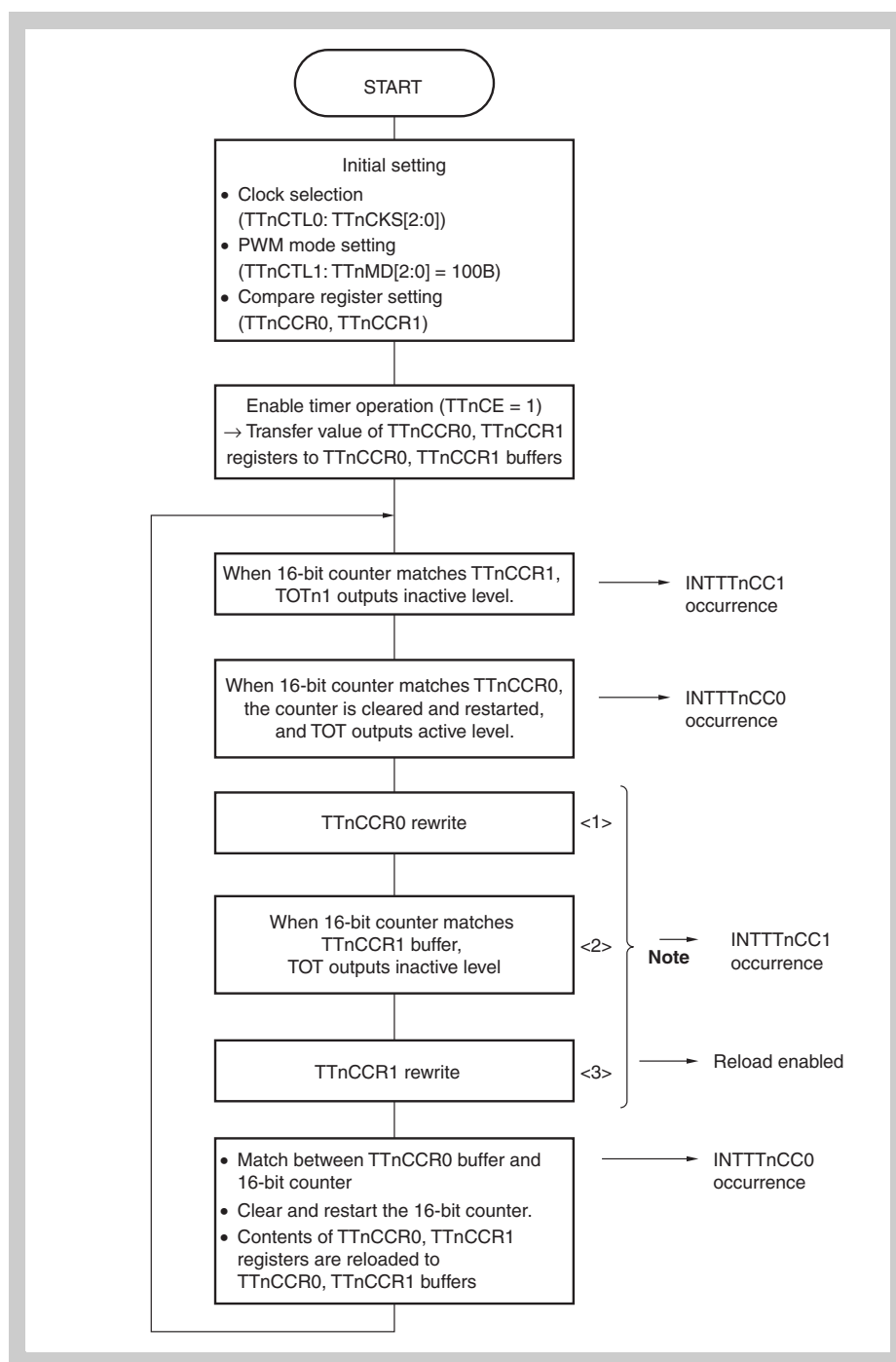
During count operation, a compare match interrupt (INTTTnCC0) is output upon a match between the counter and TTnCCR0 register, and a compare match interrupt (INTTTnCC1) is output upon a match between the counter and TTnCCR1 register.

The TTnCCR0 and TTnCCR1 registers can be rewritten during count operation. Compare register reload occurs upon a match between the counter value and the TTnCCR0 buffer register. However, since the next reload timing becomes valid when the TTnCCR1 register is written to, write the same value to the TTnCCR1 register even when wishing to rewrite only the value of the TTnCCR0 register. Reloading is not performed if only the TTnCCR0 register is rewritten.

In the PWM mode, the TTnCCR0 and TTnCCR1 registers have their function fixed as compare registers, so the capture function cannot be used.

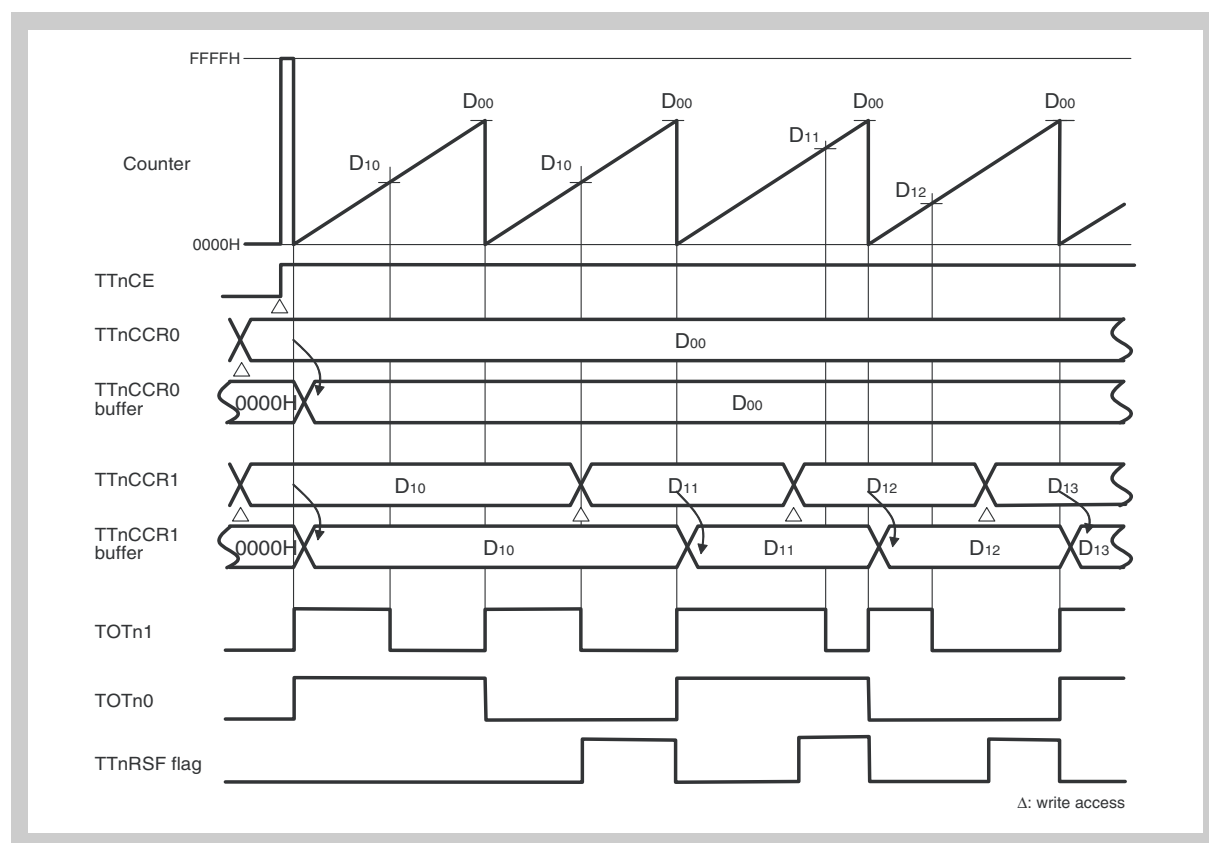


**Figure 18-17 Basic operation mode in PWM mode (1/2)**  
When values of TTnCCR0 and TTnCCR1 registers are not rewritten during timer operation



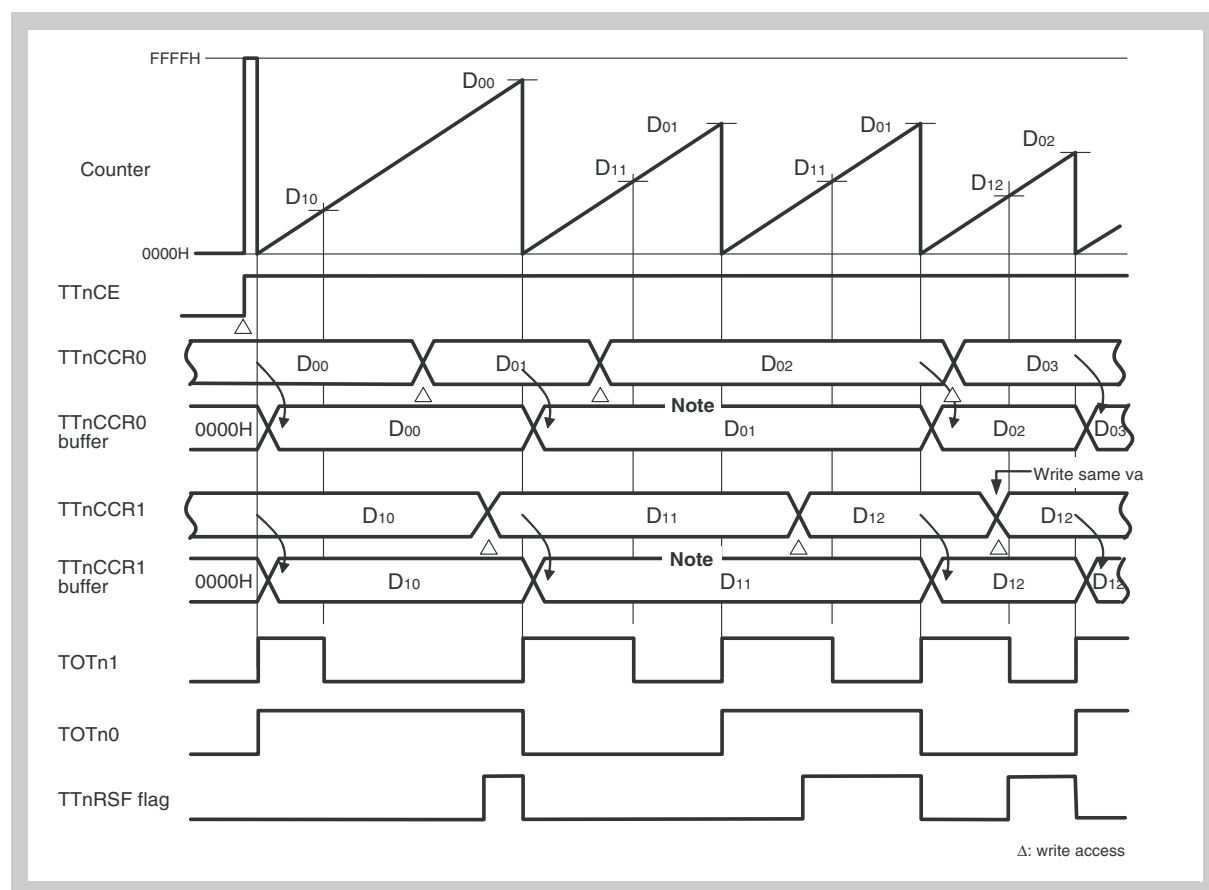
**Figure 18-18 Basic operation flow in PWM mode (2/2)**  
When values of TTnCCR0 and TTnCCR1 registers are rewritten during timer operation

**Note** Regarding the sequence, the timing of <2> may differ depending on the <1> or <3> rewrite timing, the value of the TTnCCR1 register, etc., but of <1> and <3>, always make <3> the last.



**Figure 18-19 Basic operation timing in PWM mode (1/2)**  
When only value of TTNCCR1 is rewritten, and TOTn0 and TOTn1 are output (TTnOE0, 1 = 1, TTNOL0, 1 = 0)

- Note**
1. D<sub>00</sub>: Setting value of TTnCCR0 register (0000H to FFFFH)  
D<sub>10</sub>, D<sub>11</sub>, D<sub>12</sub>, D<sub>13</sub>: Setting values of TTnCCR1 register (0000H to FFFFH)
  2. TOTn1 (PWM) duty = (setting value of TTnCCR1 register) × (count clock cycle)  
TOTn1 (PWM) cycle = (setting value of TTnCCR0 register + 1) × (count clock cycle)
  3. TOTn0 is toggled immediately following counter start and at (setting value of TTnCCR0 register + 1) × (count clock cycle)



**Figure 18-20 Basic operation timing in PWM mode (2/2)**  
 When values of TTnCCR0 and TTnCCR1 register are rewritten, TOTn0 and TOTn1 are output (TTnOE0, 1 = 1, TTnOL0, 1 = 0)

- Note**
1. The TTnCCR1 register was not written to, so transfer to the TTnCCR0 buffer register was not performed. Held until the next reload timing.
  2. D<sub>00</sub>, D<sub>01</sub>, D<sub>02</sub>, D<sub>03</sub>: Setting values of TTnCCR0 register (0000H to FFFFH)  
 D<sub>10</sub>, D<sub>11</sub>, D<sub>12</sub>, D<sub>13</sub>: Setting values of TTnCCR1 register (0000H to FFFFH)
  3. The TOTn0 and TOTn1 pins become high level at timer count start.

### 18.6.6 Free-running mode

The operation flow in free-running mode depends on the setting of the TTnOPT0.TTnCCS[1:0] bits, and is shown below.

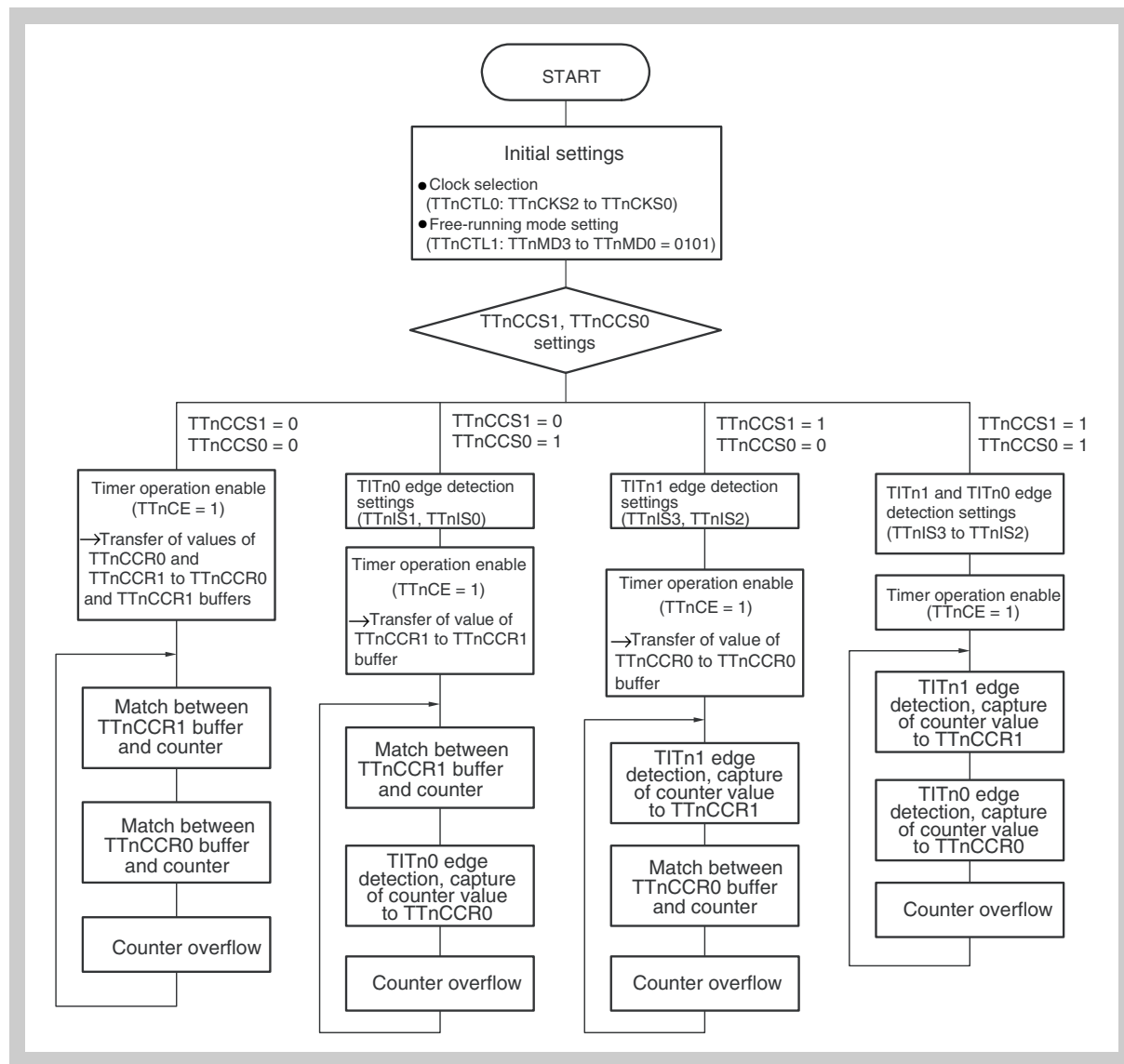


Figure 18-21 Basic operation flow in free-running mode

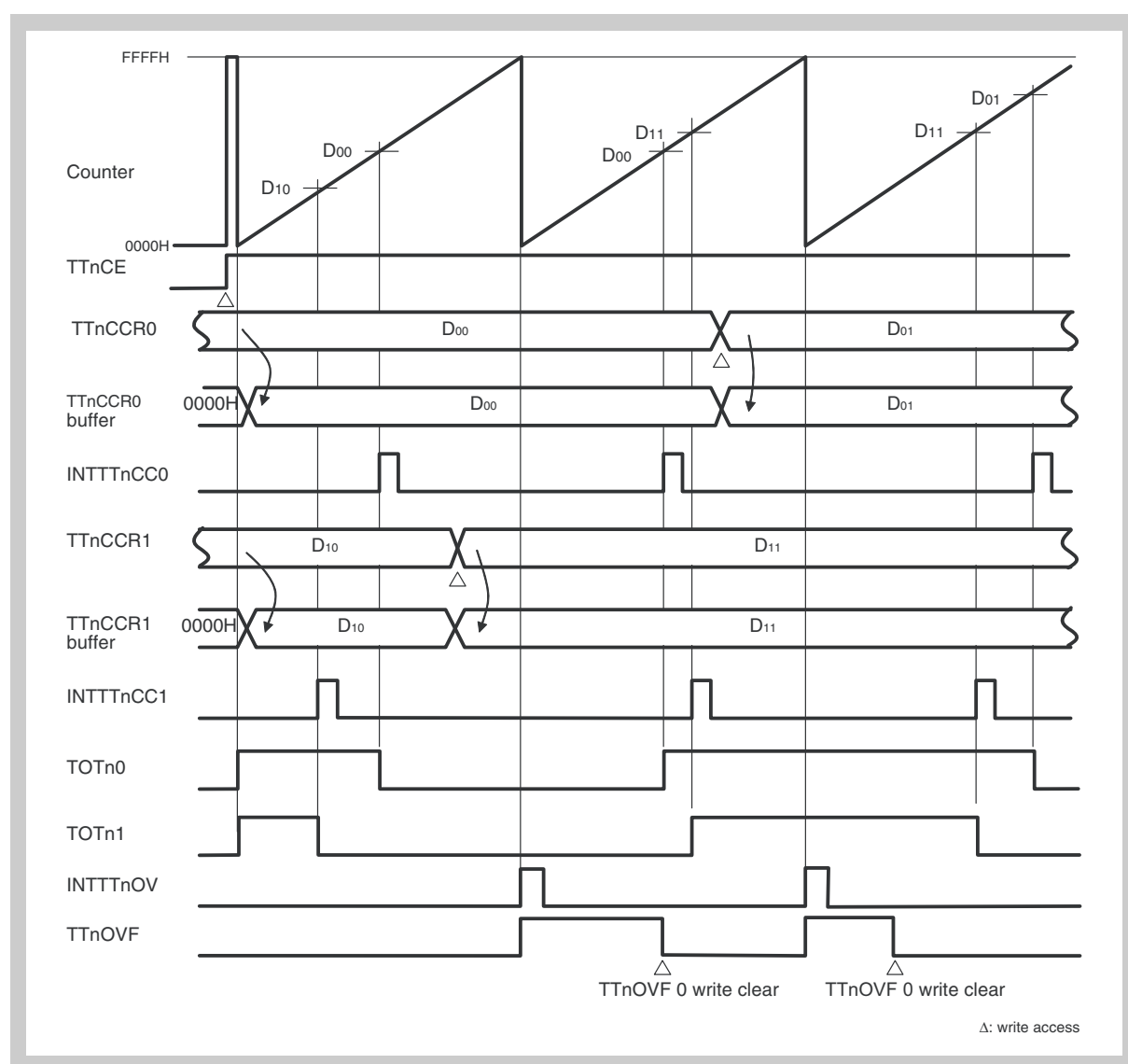


**(1) Compare function (TTnCCS1 = 0, TTnCCS0 = 0)**

When TTnCTL0 register bit TTnCE is set to 1, the counter counts from 0000H to FFFFH. An overflow interrupt (INTTTnOV) is output when the counter value changes from FFFFH to 0000H, and the counter is cleared. The count operation is performed in the free-running mode until TTnCE = 0 is set. Moreover, during count operation, a compare match interrupt (INTTTnCC0) is output upon a match between the counter and TTnCCR0 buffer register, and a compare match interrupt (INTTTnCC1) is output upon a match between the counter and TTnCCR1 buffer register.

The TTnCCR0 and TTnCCR1 registers can be rewritten using the anytime write method, regardless of the value of the TTnCE bit.

The TOTn0 and TOTn1 pins are toggle output controlled when bits register TTnOE0 and TTnOE1 of the TTnLOC0 register are set to 1.



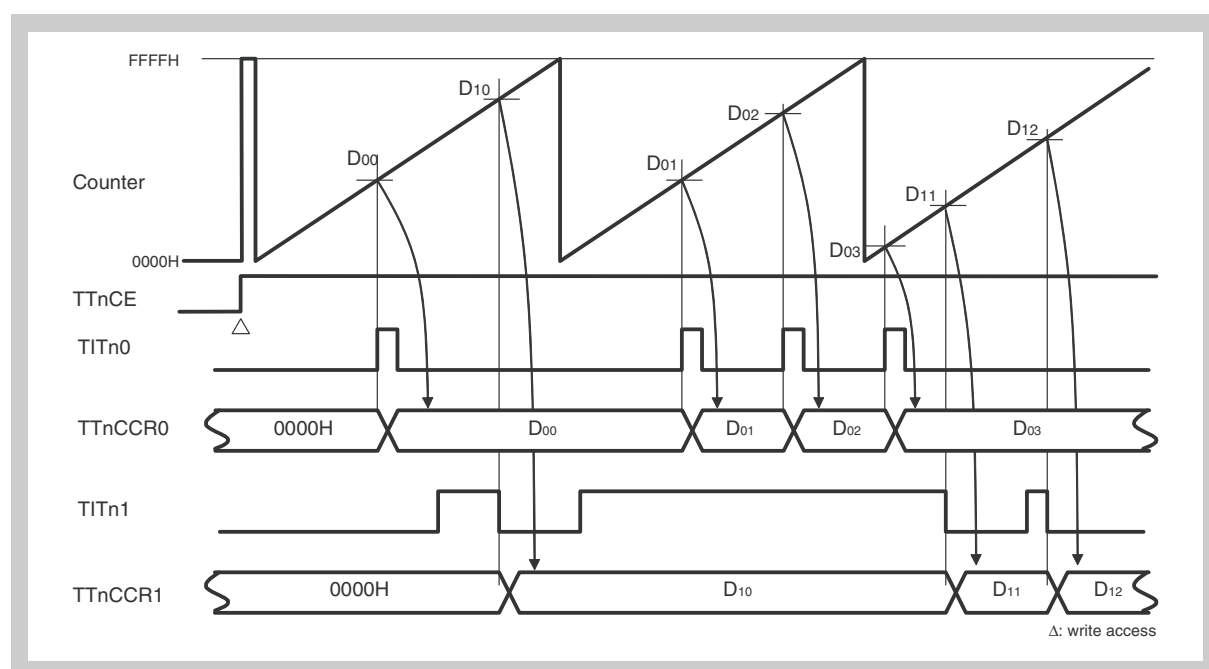
**Figure 18-22 Basic operation timing in free-running mode (compare function)**  
When values of TTnCCR0 and TTnCCR1 registers are rewritten, TOTn0, TOTn1 are output (TTnOE0, 1 = 1, TTnOL0, 1 = 0)

- Note**
1. D<sub>00</sub>, D<sub>01</sub>: Setting values of TTnCCR0 register (0000H to FFFFH)  
D<sub>10</sub>, D<sub>11</sub>: Setting values of TTnCCR1 register (0000H to FFFFH)
  2. TOTn0 (toggle) width = (setting value of TTnCCR0 register + 1) × (count clock cycle)
  3. TOTn1 (toggle) width = (setting value of TTnCCR1 register + 1) × (count clock cycle)
  4. Pins TOTn0 and TOTn1 become high level at count start.

**(2) Capture function (TTnCCS1 = 1, TTnCCS0 = 1)**

When TTnCTL0 register bit TTnCE is set to 1, the counter counts from 0000H to FFFFH. An overflow interrupt (INTTTnOV) is output when the value of the counter changes from FFFFH to 0000H, and the counter is cleared. The count operation is performed in the free-running mode until TTnCE = 0 is set. When, during count operation, the counter value is captured to the TTnCCR0 and TTnCCR1 registers through detection of the valid edge of capture input (TITn1, TITn0), a capture interrupt (INTTTnCC0, INTTTnCC1) is output.

Regarding capture in the vicinity of overflow (FFFFH), judgment is possible with the overflow flag (TTnOVF). However, judgment with the TTnOVF flag is not possible when the capture trigger interval is such that it includes two overflow occurrences (2 or more free-running cycles).

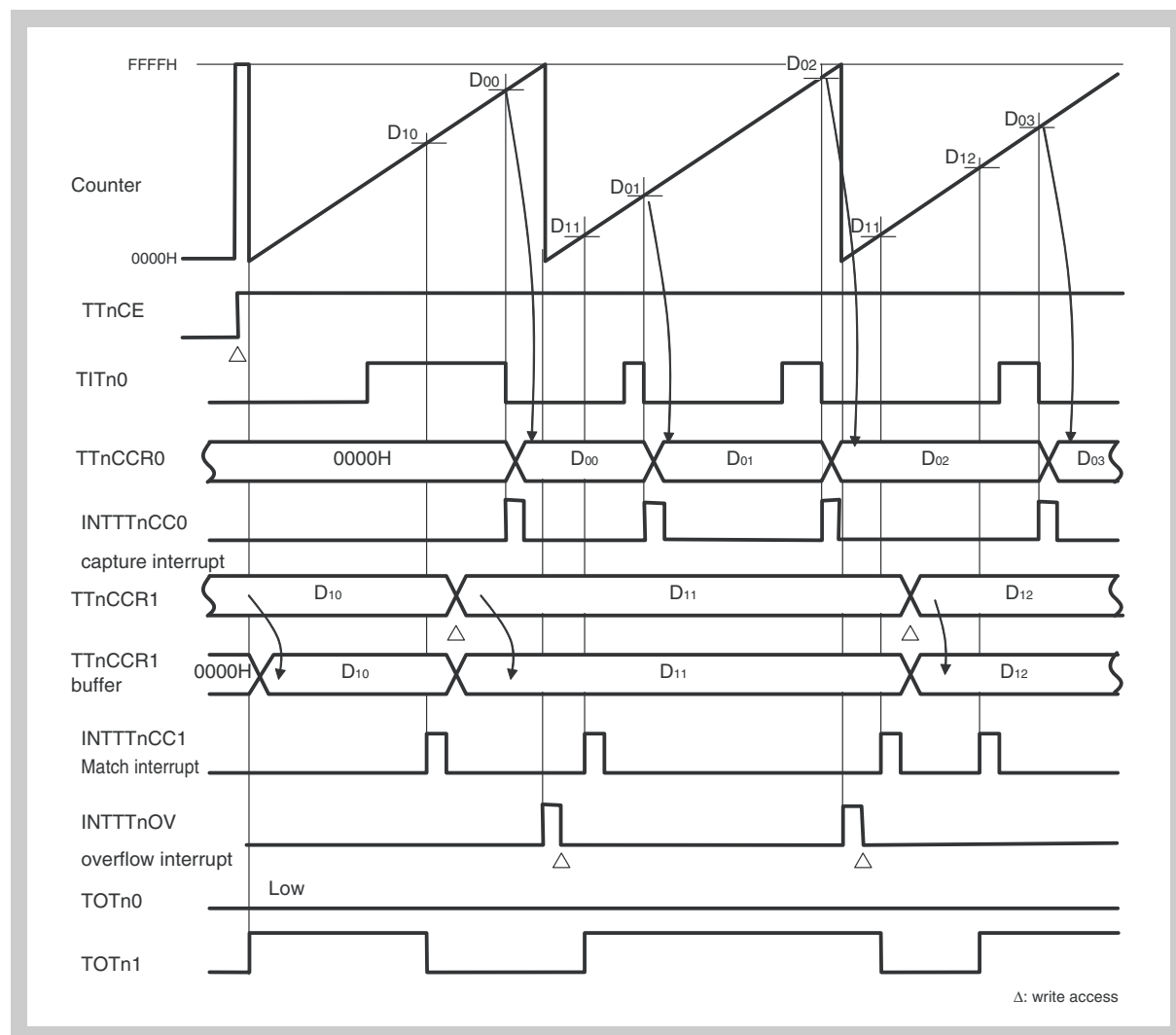


**Figure 18-23 Basic operation timing in free-running mode (capture function)**  
When TOTn0, TOTn1 are not output (TTnOE0, 1 = 0, TTnOL0, 1 = 0)

- Note**
1. D<sub>00</sub>, D<sub>01</sub>: Values captured to TTnCCR0 register (0000H to FFFFH)  
D<sub>10</sub>, D<sub>11</sub>: Values captured to TTnCCR1 register (0000H to FFFFH)
  2. TITn0: Setting to rising edge detection (TTnIOC1 register bits TTnIS1, TTnIS0 = 01)  
TITn1: Setting to falling edge detection (TTnIOC1 register bits TTnIS3, TTnIS2 = 10)

**(3) Compare/capture function (TTnCCS1 = 0, TTnCCS0 = 1)**

When TTnCTL0 register bit TTnCE is set to 1, the counter counts from 0000H to FFFFH, an overflow interrupt (INTTTnOV) is output when the value of the counter changes from FFFFH to 0000H, and the counter is cleared. The count operation is performed in the free-running mode until TTnCE = 0 is set. The TTnCCR1 register is used as a compare register, and as the interval function upon a match between the counter and TTnCCR1 register, a compare match interrupt (INTTTnCC1) is output. Since the TTnCCR0 register is set to the capture function, the TOTn0 pin cannot be controlled even when TTnIOC0 register bit TTnOE0 is set to 1.



**Figure 18-24 Basic operation timing in free-running mode (compare/capture function)**  
**When value of TTnCCR1 is rewritten, TOTn0, TOTn1 are output**  
**(TTnOE0, 1 = 1, TTnOL0, 1 = 0)**

- Note**
1. D<sub>00</sub>, D<sub>01</sub>: Setting values of TTnCCR1 register (0000H to FFFFH)  
D<sub>10</sub>, D<sub>11</sub>, D<sub>12</sub>, D<sub>13</sub>, D<sub>14</sub>, D<sub>15</sub>: Values captured to TTnCCR0 register (0000H to FFFFH)
  2. TITn0: Setting to rising edge detection (TTnIOC1 register bits TtnIS1, TtnIS0 = 11)

**(4) Overflow flag**

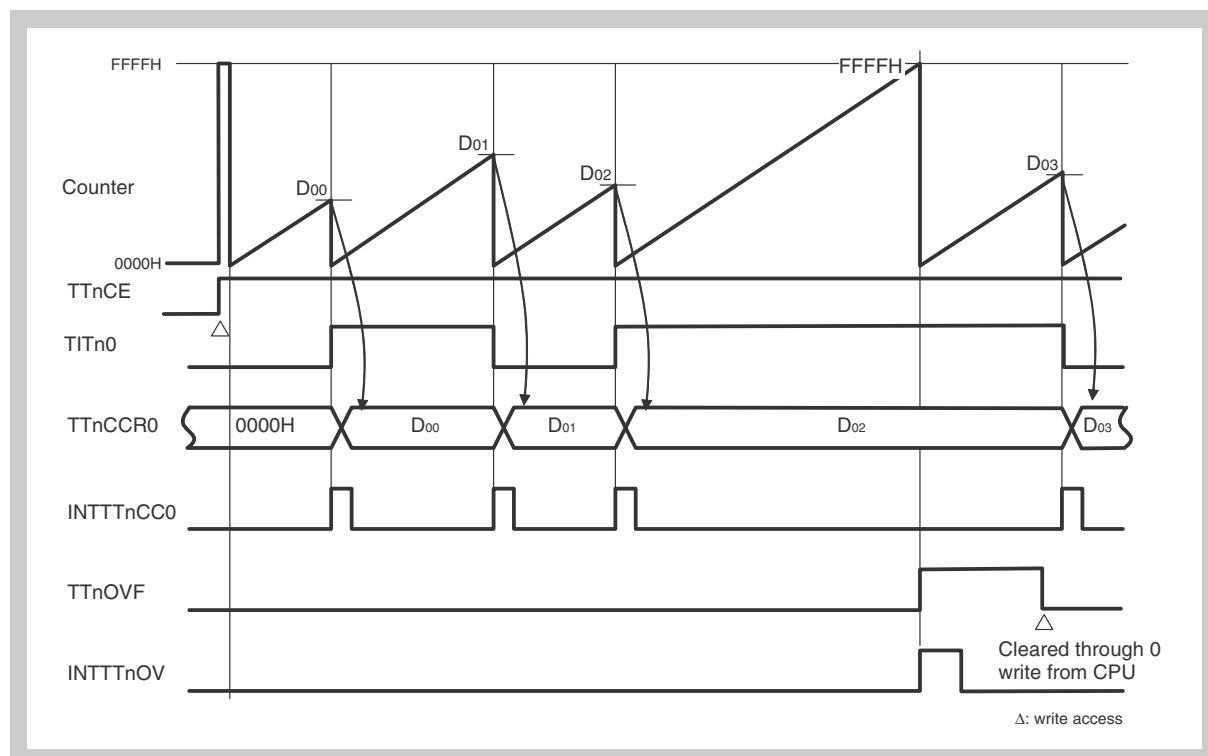
When, in the free-running mode, the counter overflows from FFFFH to 0000H, the overflow flag (TTnOVF) is set to "1", and an overflow interrupt (INTTTnOV) is output.

The overflow flag is cleared through 0 write from the CPU.

(The overflow flag is not cleared by just being read.)

**18.6.7 Pulse width measurement mode**

In the pulse width measurement mode, counting is performed in the free-running mode. The counter value is saved to the TTnCCR0 register, and the counter is cleared to 0000H. As a result, the external input pulse width can be measured. However, when measuring a long pulse width that exceeds counter overflow, perform judgment with the overflow flag. Measurement of pulses during which overflow occurs twice or more is not possible, so adjust the counter's operating frequency. Even in the case of TITn1 pin edge detection, pulse width measurement can be similarly performed by using the TTnCCR1 register.



**Figure 18-25 Basic operation timing in pulse width measurement mode**  
(TTnOE0, 1 = 0, TTnOL0, 1 = 0)

- Note**
1. D00, D01, D02, D03: Values captured to TTnCCR0 register (0000H to FFFFH)
  2. TITn0: Setting to rising edge/falling edge (both edges) detection (TTnIOC1 register bits TtnIS1, TtnIS0 = 11)

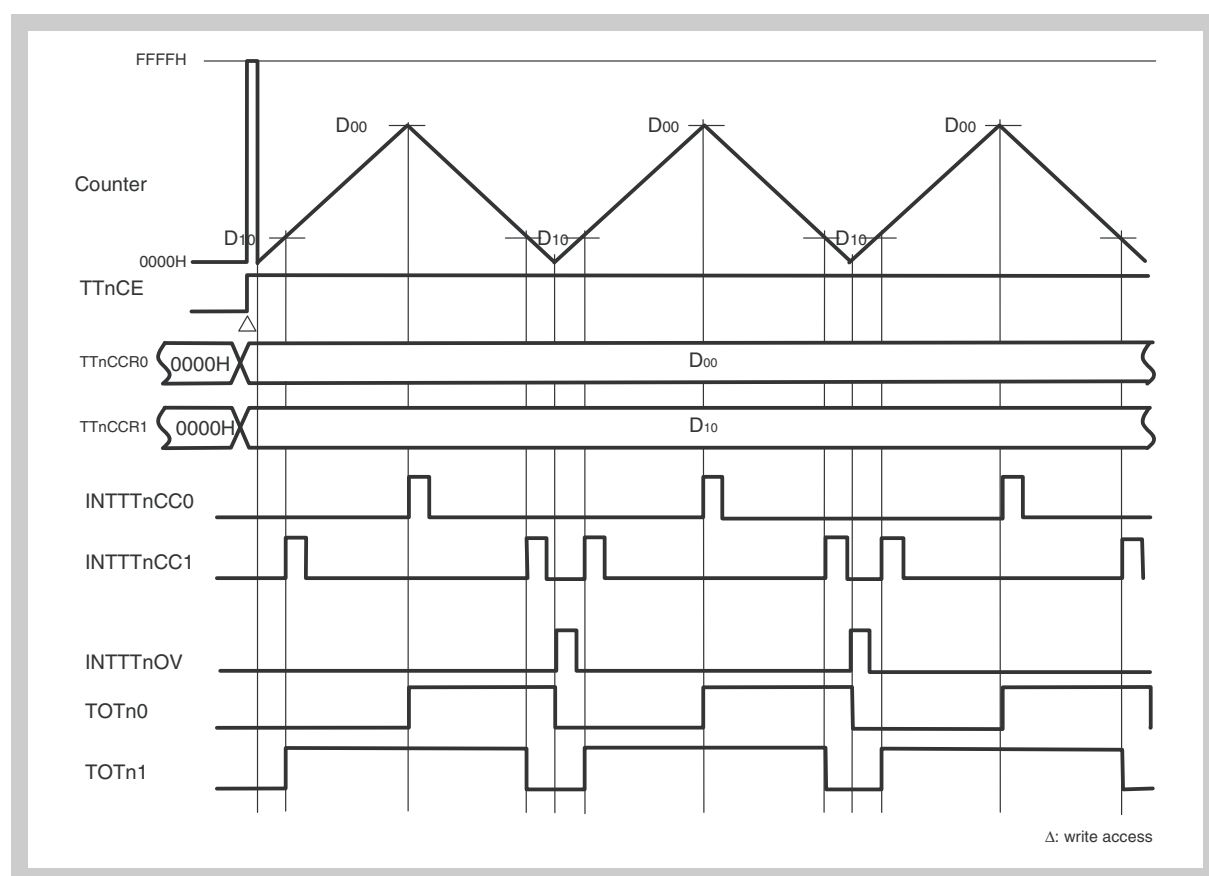
### 18.6.8 Triangular wave PWM mode

In the triangular wave PWM mode, similarly to in the PWM mode, when the duty is set to the TTnCCR1 register, the cycle is set to the TTnCCR0 register, and TTnCE = 1 is set, variable duty and cycle type triangular wave PWM output is performed from pin TOTn1. The TOTn0 pin is toggle output upon a match with the TTnCCR0 buffer register and upon counter underflow. Upon a match between the counter and TTnCCR0 register during count operation, a compare match interrupt (INTTTnCC0) is output, and upon a match between the counter and TTnCCR1 register, a compare match interrupt (INTTTnCC1) is output. Moreover, upon counter underflow, an overflow interrupt (INTTTnOV) is output.

The TTnCCR0 and TTnCCR1 registers can be rewritten during count operation. Compare register reload occurs upon a match between the counter value and the TTnCCR0 buffer register. However, since the next reload timing becomes valid when the TTnCCR1 register is written to, write the same value to the TTnCCR1 register even when wishing to rewrite only the value of the TTnCCR0 register. Reloading is not performed if only the TTnCCR0 register is rewritten. The reload timing is the underflow timing.

In the triangular wave PWM mode, the TTnCCR0 and TTnCCR1 registers have their function fixed as compare registers, so the capture function cannot be used.

**Note** In the triangular wave PWM mode, set the TTnCCR0 register to a value of  $0 \leq \text{TTnCCR0} \leq \text{FFFFH}$ .



**Figure 18-26 Basic operation timing in triangular wave pwm mode**  
When TOTn0, TOTn1 are output (TTnOE0, 1 = 1, TTnOL0, 1 = 0)

### 18.6.9 Encoder count function

Three encoder count function modes are provided, one for each capture compare function.

Mode	TTnCCR0 register	TTnCCR1 register
Encoder compare mode	Compare only	Compare only
Encoder capture mode	Capture only	Capture only
Encoder capture compare mode	Compare only	Capture only

#### (1) Counter up/down control

Counter up/down control is performed and the counter is operated according to the phase of signals TENCTn0 and TENCTn1 from the encoder and the set conditions of bits TTnUDS1 and TTnUDS0 of the TTnCTL2 register.

#### (2) Basic operation

To use the TTnCCR0 and TTnCCR1 registers are compare-only registers, enable rewrite during timer operation.

The rewrite method is anytime write.

A compare match interrupt (INTTTnCC0) is output upon a match between the counter and TTnCCR0 register. A compare match interrupt (INTTTnCC1) is output upon a match between the counter and TTnCCR1 register.

To use the TTnCCR0 and TTnCCR1 registers are capture-only registers, save the counter value to the capture registers (TTnCCR0, TTnCCR1) through detection of the edges of pins TITn0 and TITn1. Specification of the detection of the edges of pins TITn0 and TITn1 is done with bits TTnIS3 to TTnIS0 of the TTnIOC1 register. Pin TOTn0 outputs the count status.

If the TTnCCR1 register is a compare-only register, pin TOTn1 is toggled upon a match between the counter and the TTnCCR1 register.

If the TTnCCR1 register is a capture-only register, pin TOTn1 becomes the level set with bit TTnOL1 of the TTnIOC0 register.

If TTnOL1 = 0, pin TOTn1 outputs a low level, and if TTnOL1 = 1, pin TOTn1 outputs a high level.

#### (3) Counter clear operation

Clearing of the counter to 0000H is performed under the following conditions in each mode.

Clear condition	Encoder compare	Encoder capture	Encoder capture compare
Method whereby counter is cleared to 0000H upon match with compare register (setting of TTnCTL2 register bits TTnECM1, TTnECM0)	√	-	√ (TTnECM0 <sup>Note</sup> )

Clear condition	Encoder compare	Encoder capture	Encoder capture compare
Method whereby counter is cleared to 0000H upon detection of edge of pin TECR0 (setting of bits TTnECS1, TTnECS0 when TTnIOC3 register bit TTnSCE = 0)	√	√	√
Method whereby counter is cleared to 0000H by special clear function of encoder (setting of bits TTnZCL, TTnBCL, TTnACL when TTnIOC3 register bit TTnSCE = 1)	√	√	√

**Note** Since the TTnCCR1 register is a capture-only register, the setting of TTnCTL2 register bit TTnECM1 are invalid.



**(4) Control through TTnCTL2 register**

The settings of the TTnCTL2 register in the encoder compare mode, encoder capture mode, and encoder capture/compare mode (TTnMD3 to TTnMD0 = 1000B, 1001B, 1010B) are as follows.

TTnMD3 to 0	TTnUDS1 to 0	TTnECM1	TTnECM0	TTnLDE	Clear	Load
1000B	All settings possible 00B 01B 10B 11B	0	0	0	-	—
				1		√
			1	0	TTnCCR0	—
				1		●
		1	0	Invalid	TTnCCR1	—
			1	Invalid	TTnCCR0 TTnCCR1	—
1001B		Invalid			—	—
1010B		0	0	0	—	—
				1		√
		1	0	TTnCCR0	-	
			1		●	
		1	0	Invalid	—	—
	1		Invalid	TTnCCR0	—	

- In the case of bits TTnUDS1 and TTnUDS0, up/down judgment control is performed for the phase input from pins TENCTn0 and TENCTn1.
- In the case of bits TTnECM1 and TTnECM0, counter clear control is performed upon a match between the counter value and the compare setting value.

Bits TTnECM1 and TTnECM0 are valid in modes where the TTnCCR0 or TTnCCR1 register is used as a compare-only register.  
These bits are invalid in modes where the TTnCCR0 or TTnCCR1 register is used as a capture-only register.

- The TTnLDE bit controls the function to load to the counter the setting value of the TTnCCR0 register upon occurrence of counter underflow.  
Bit TTnLDE is valid only when the TTnECm bit setting is 00B, 01B, in a mode where the TTnCCR0 or TTnCCR1 register is used as a compare-only register.

In the case of all other settings, bit TTnLDE is invalid even if manipulated.

As an example of the use of the encoder count function, counter operation becomes possible between the setting values of registers 0000H to TTnCCR0 by using the counter load functions (TTnLDE = 1) indicated with “•” in the table, and the function for clearing the counter to 0000H in case the count operation following a match with the TTnCCR0 buffer register is up count (TTnECM0 = 1). (Refer to “Counter load function for TTnCCR0 register setting value upon underflow (bit TTnLDE of register TTnCTL2))” on page 1004).

**(a) Up/down count selection specification (TTnCTL2 register bits TTnUDS1, TTnUDS0)**

Counter up/down is judged according to the settings of bits TTnUDS1 and TTnUDS0, and the phases input from pins TENCTn0 and TENCTn1.

Bits TTnUDS1 and TTnUDS0 are valid only in the encoder compare mode, encoder capture mode, and encoder capture/compare mode.

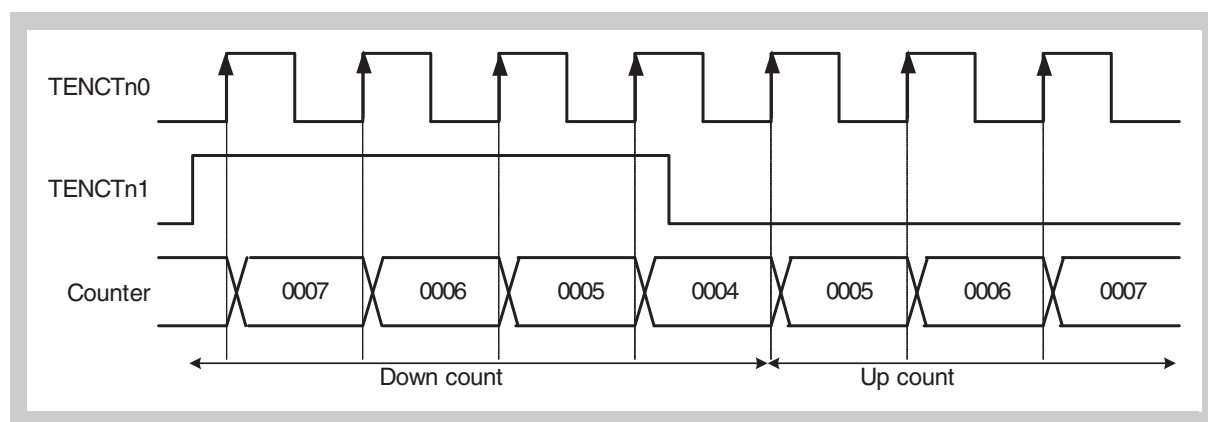
1. TTnCTL2: TTnUDS1, 0 = 00B (count judgment mode 1)

A phase (pin TENCTn0)	B phase (pin TENCTn1)	Count
Rising edge	High level	Down
Falling edge		
Both edges		
Rising edge	Low level	Up
Falling edge		
Both edges		

Operation example:

TTnIOC3: TENCTn1 pin input Edge detection specification invalid  
TTnEIS3 to 2

TTnIOC3: TENCTn0 pin input Rising edge detection  
TTnEIS1 to 0 = 10B



**Figure 18-27 Encoder count function up/down count selection specification timings (1/6) - timing 1**

**Note** Counting is performed when the edges of the TENCTn0/TENCTn1 pin inputs overlap.

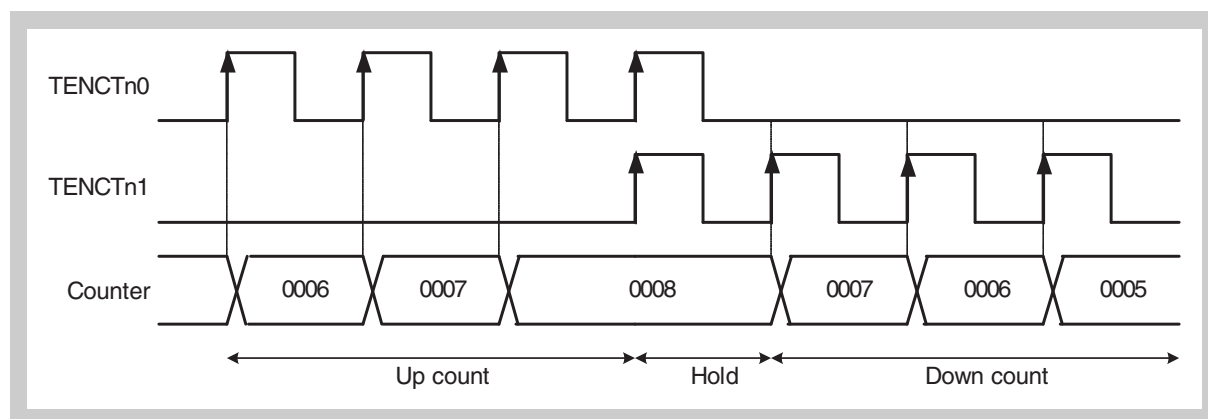
2. TTnCTL2: TTnUDS1, 0 = 01B (count judgment mode 2)

A phase (pin TENCTn0)	B phase (pin TENCTn1)	Count
Low level	Rising edge	Down
	Falling edge	
	Both edges	
High level	Rising edge	
	Falling edge	
	Both edges	
Rising edge	Low level	Up
Falling edge		
Both edges		
Rising edge	High level	
Falling edge		
Both edges		
Simultaneous pin TENCTn0/TENCTn1 inputs		Hold

Operation example:

TTnIOC3: TTnEIS3, TENCTn1 pin input Rising edge detection  
2 = 10B

TTnIOC3: TTnEIS1, TENCTn0 pin input Rising edge detection  
0 = 10B



**Figure 18-28** Encoder count function up/down count selection specification timings (2/6) - timing 2

**Note** The count value is held when the edges of the TENCTn0/TENCTn1 pin inputs overlap.

3. TTnCTL2: TTnUDS1, 0 = 10B (count judgment mode 3)

A phase (pin TENCTn0)	B phase (pin TENCTn1)	Count
Low level	Falling edge	Hold
Rising edge	Low level	Down
High level	Rising edge	Hold
Falling edge	High level	
Rising edge	High level	
High level	Falling edge	
Falling edge	Low level	Up
Low level	Rising edge	Hold
Rising edge	Rising edge	Hold
Falling edge	Rising edge	
Rising edge	Falling edge	Down
Falling edge	Falling edge	Up

Operation example:

TTnIOC3: Pins TENCTn1, Edge detection specification invalid  
TTnEIS3 to 0 TENCTn0

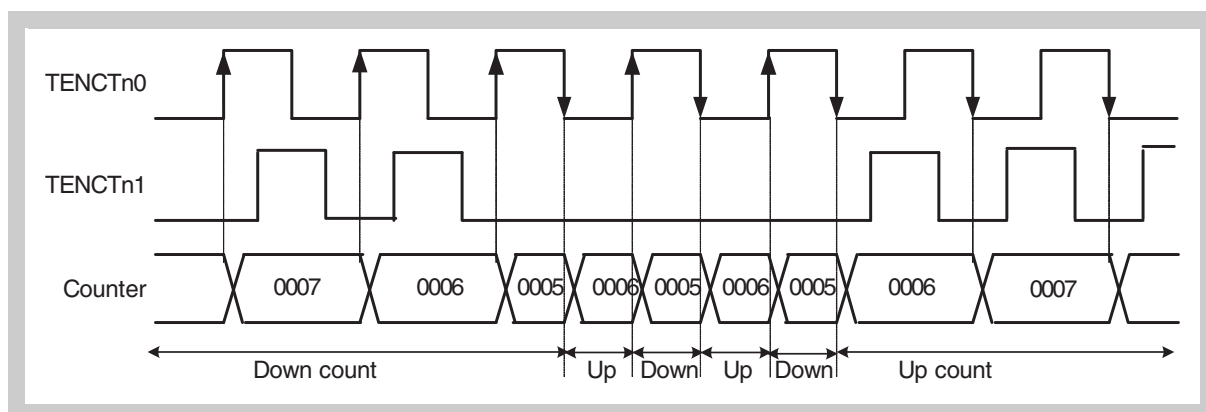


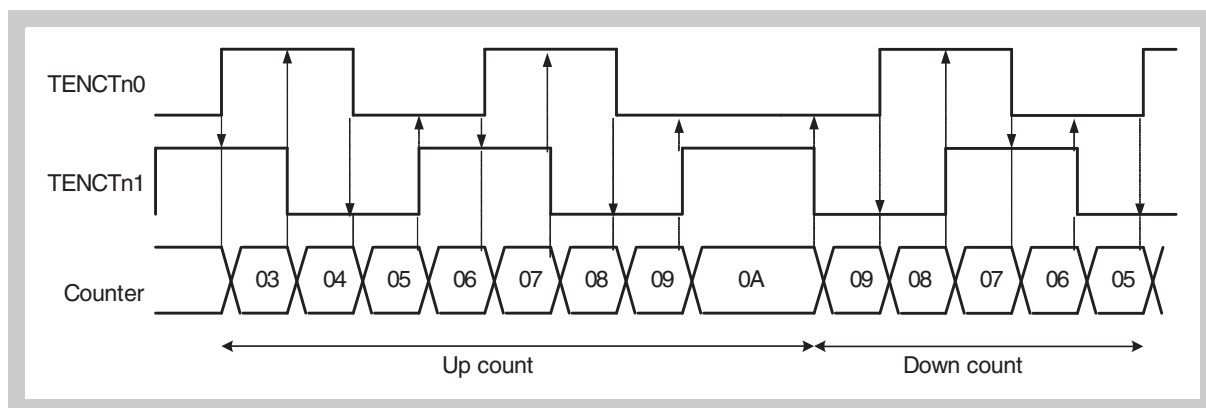
Figure 18-29 Encoder count function up/down count selection specification timings (3/6) - timing 3

4. TTnCTL2: TTnUDS1, 0 = 11B (count judgment mode 4)

A phase (pin TENCTn0)	B phase (pin TENCTn1)	Count
Low level	Falling edge	Down
Rising edge	Low level	
High level	Rising edge	
Falling edge	High level	
Rising edge	High level	Up
High level	Falling edge	
Falling edge	Low level	
Low level	Rising edge	
Simultaneous pin TENCTn0/TENCTn1 inputs		Hold

## Operation example 1:

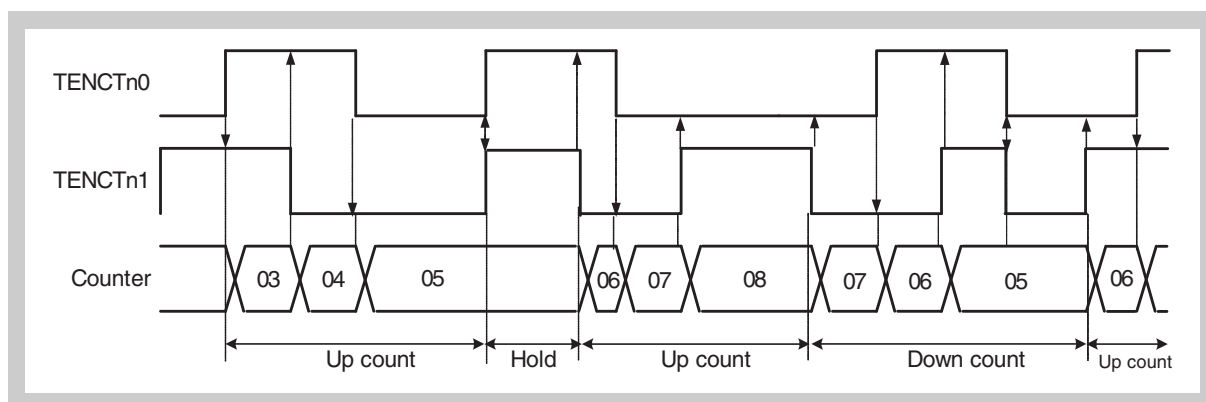
TTnIOC2: Pins TENCTn1, Edge detection specification invalid  
 TTnEIS3 to 0 TENCTn0



**Figure 18-30 Encoder count function up/down count selection specification timings (4/6) - timing 4**

## Operation example 2:

TTnIOC2: Pins TENCTn1, Edge detection specification invalid  
 TTnEIS3 to 0 TENCTn0



**Figure 18-31 Encoder count function up/down count selection specification timings (5/6) - timing 5**

**Note** The count value is held when the edges of the TENCTn0/TENCTn1 pin inputs overlap.

**(b) Counter clear condition setting upon match between counter value and compare setting value (TTnCTL2 register bits TTnECM1, TTnECM0)**

Counter operation is performed according to the setting values of these bits upon a match between the counter value and the compare setting value.

1. TTnECM1, 0 = 00B

Counter clear is not performed upon a match between the counter and compare values.

2. TTnECM1, 0 = 01B

Counter clear is performed upon a match between the counter and the TTnCCR0 register.

Next count operation	Description
Up count	Clear counter to 0000H.
Down count	Down count the counter value.

3. TTnECM1, 0 = 10B

Operation is performed under the following conditions upon a match between the counter and TTnCCR1 register.

Next Count operation	Description
Up count	Up count the counter value.
Down count	Clear counter to 0000H.

4. TTnECM1, 0 = 11B

- Operation is performed under the following conditions upon a match between the counter and TTnCCR0 register.

Next count operation	Description
Up count	Clear counter to 0000H.
Down count	Down count the counter value.

- Operation is performed under the following conditions upon a match between the counter and TTnCCR1 register.

Next count operation	Description
Up count	Up count the counter value.
Down count	Clear counter to 0000H.

---

**Caution** In encoder compare mode (TTnMD3 to TTnMD0 bits = 1000B), or encoder capture-compare mode (TTnMD3 to TTnMD0 bits = 1010B), if the compare registers (TTnCCR0, TTnCCR1) are set to the same value of TTnTCW register when TTnECC bit = 0, the timer cannot perform the comparison with the compare registers (TTnCCR0, TTnCCR1) and TTnTCW register (which is the start value of TTnCNT). In this case the “encoder clear mode on match of counter and compare register” does not work at the start timing (TTnECM0 = 1, and/or TTnECM1 = 1).

---

**(c) Counter load function for TTnCCR0 register setting value upon underflow (bit TTnLDE of register TTnCTL2))**

The setting value of the TTnCCR0 register can be loaded to the counter upon counter underflow, by setting TTnLDE = 1.

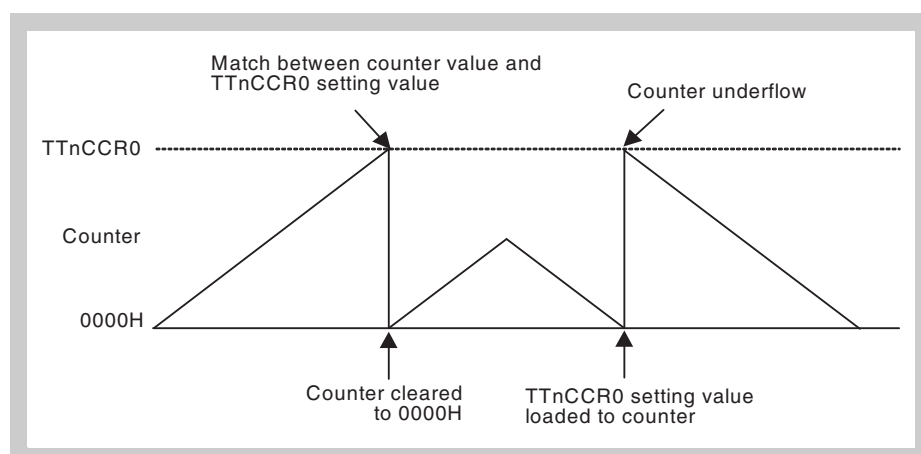
Bit TTnLDE is valid in the encoder compare mode and encoder capture compare mode.

Count operation between 0000H and setting value of TTnCCR0 register setting:

Set TTnLDE = 1, TTnECM1, 0 = 01B and perform count operation. When TTnECM0 = 1, the counter is cleared to 0000H if the next count following a match between the counter and TTnCCR0 register is up count.

When TTnLDE = 1, the setting value of the TTnCCR0 register is loaded to the counter upon underflow.

Therefore, the setting value of the TTnCCR0 register is used as the maximum count value and count operation can be realized within "0000H-TTnCCR0 register setting values.



**Figure 18-32 Encoder count function up/down count selection specification timings (6/6) - timing 6**



**(5) Counter clearing to 0000H through encoder clear input (pin TECRTn) (TTnIOC3 register bits TTnSCE, TTnECS1, TTnECS0)**

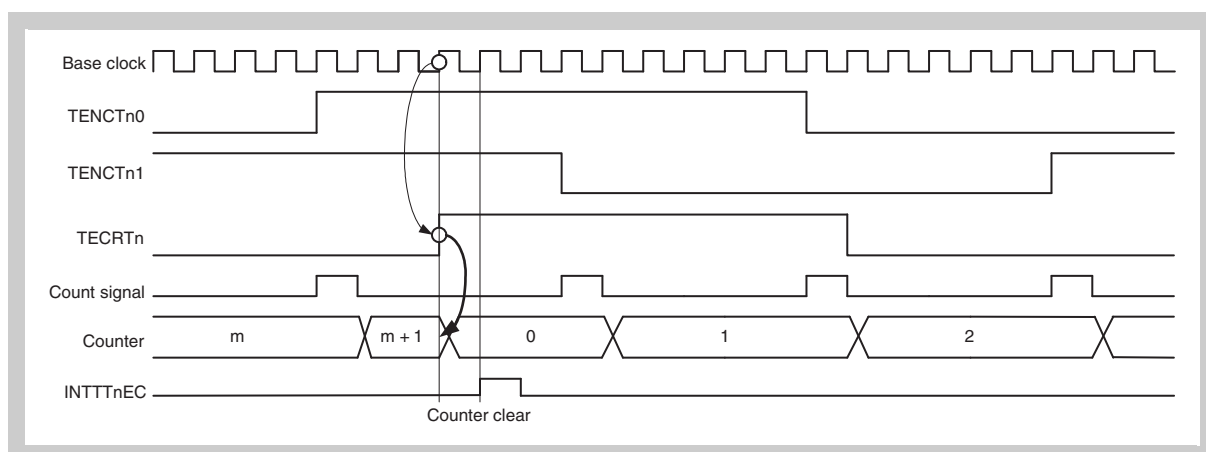
There are two methods to clear the counter to 0000H through TECRTn pin input, and encoder clear input is controlled by bit TTnSCE. Bits TTnZCL, TTnBCL, TTnACL, TTnECS1, and TTnECS0 are controlled by the setting of bit TTnSCE.

These clear methods are valid in the encoder compare mode, encoder capture mode, and encoder capture/compare mode.

TTnSCE	TTnZCL	TTnBCL	TTnACL	TTnECS1, 0	Method
0	Invalid	Invalid	Invalid	√	<1>
1	√	√	√	Invalid	<2>

**1. Method to clear counter to 0000H through detection of valid edge of TECRTn pin input (TTnSCE = 0)**

When TTnSCE = 0, the counter is cleared to 0000H in synchronization with the internal operation clock upon detection of the valid edge set through TECRTn pin input edge detection specification. At this time, an encoder clear interrupt (INTTTnEC) is output. When TTnSCE = 0, the setting of bits TTnZCL, TTnBCL, and TTnACL are invalid.



**Figure 18-33 Counter clearing to 0000h through encoder clear input (pin tecrtn) timings (1/4)**

When TTnSCE = 0, TTnECS1, 0 = 01B, TTnUDS = 11B are Set

2. Method to clear counter to 0000H through detection of level clear condition (TTnSCE = 1)

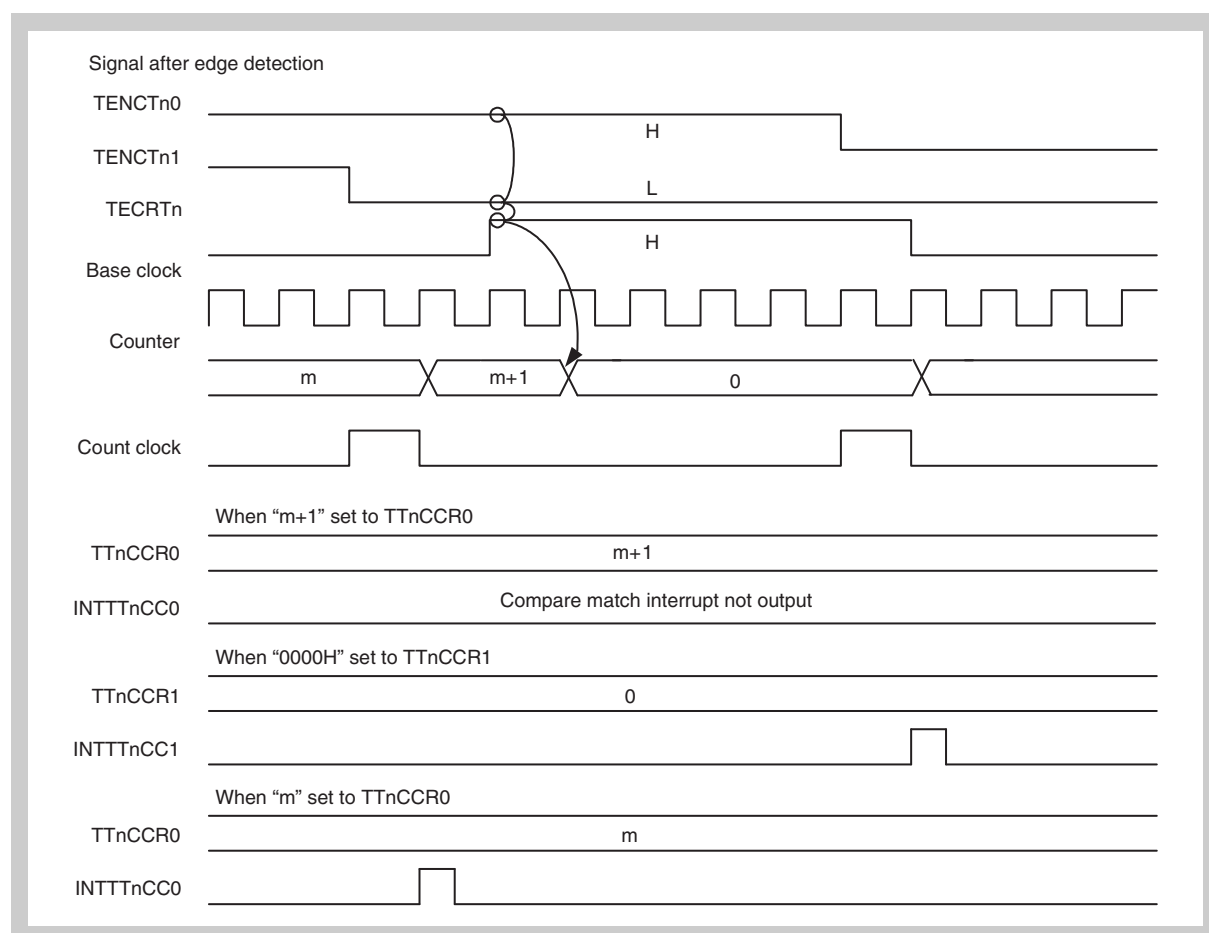
When TTnSCE = 1, the counter is cleared to 0000H according to the clear condition level of pins TECRTn, TENCTn1, and TENCTn0 set with bits TTnZCL, TTnBCL, and TTnACL. At this time, no encoder clear interrupt (INTTTnEC) is output. When TTnSCE = 1, the settings of bits TTnECS1 and TTnECS0 are invalid.

Operation example:

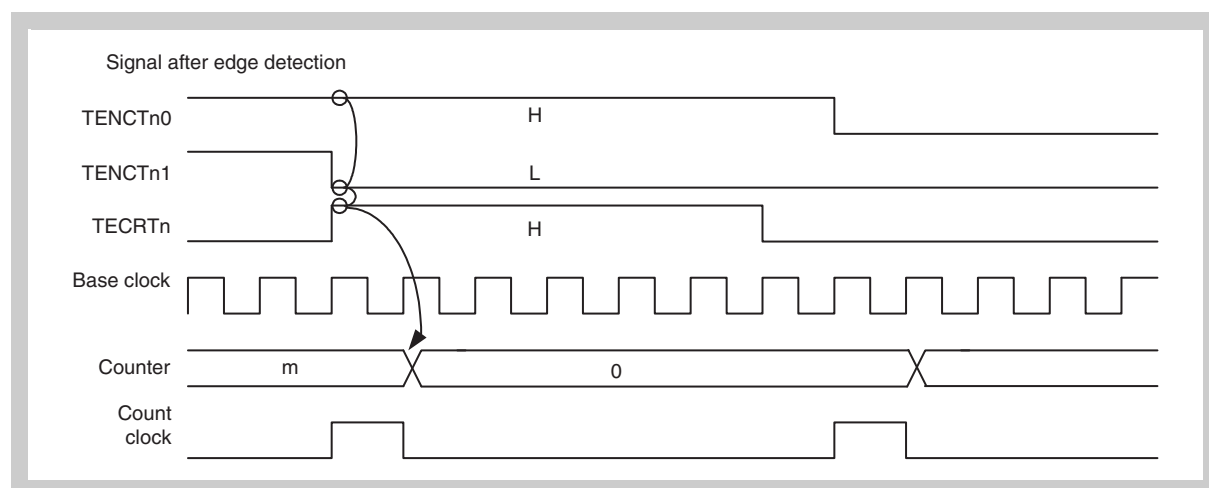
When TTnSCE = 1, TTnCLA = 1, TTnCLB = 0, TTnCLZ = 1, TTnUDS = 11B are set

<Clear condition level>

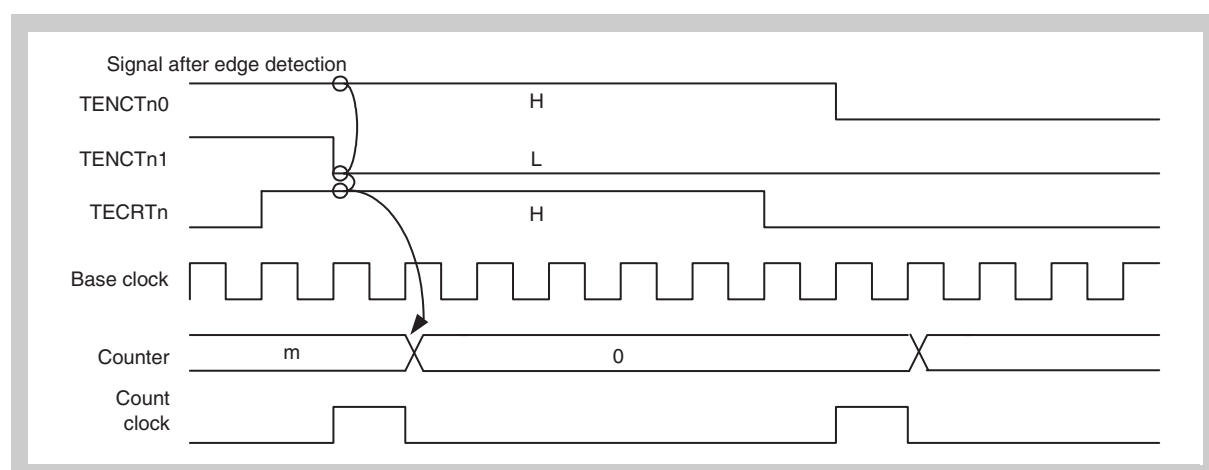
TECRTn pin: High level, TENCTn1 pin: Low level, TENCTn0 pin: High level



**Figure 18-34** Counter clearing to 0000H through encoder clear input (pin TECRTn) timings (2/4)  
When TECRTn pin input is delayed from TENCTn1 pin input during up count

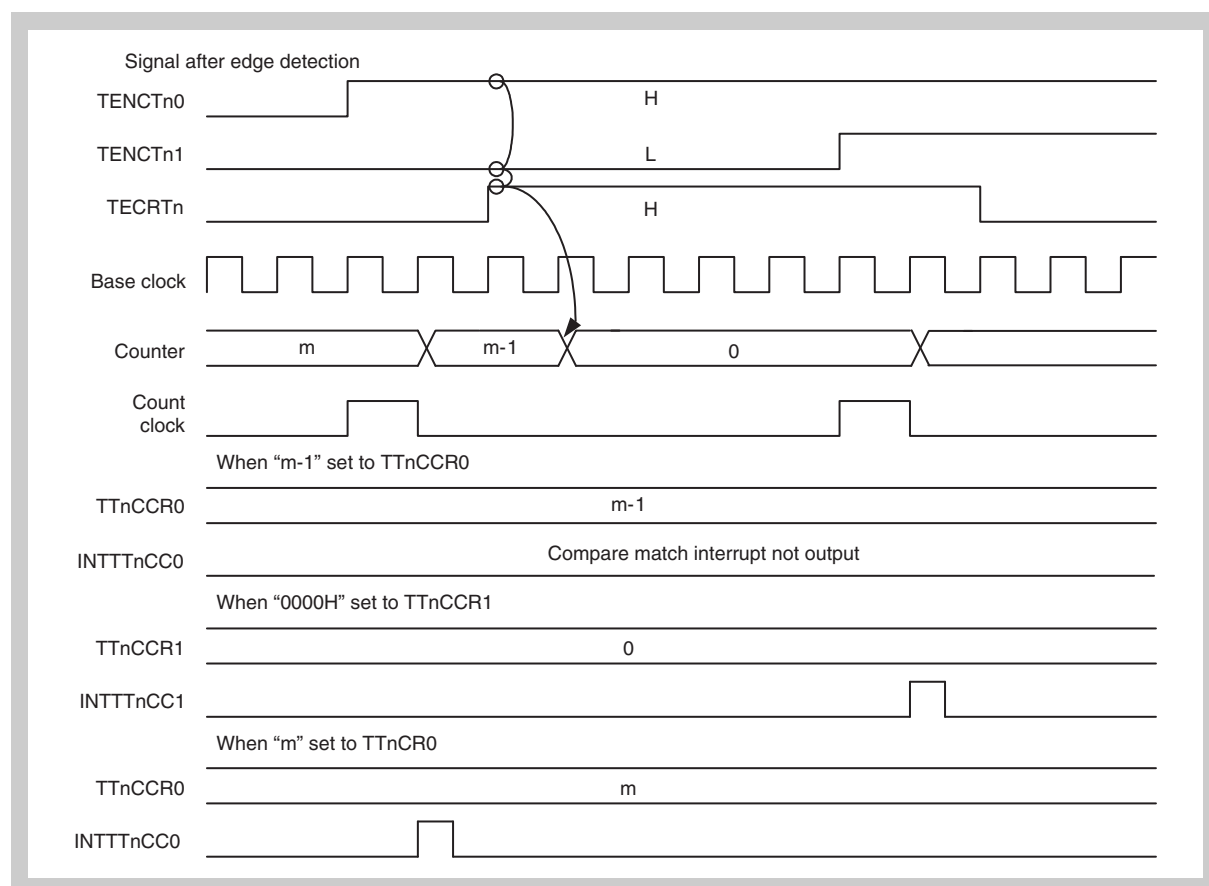


**Figure 18-35** Counter clearing to 0000H through encoder clear input (pin TECRTn) timings (3/4)  
 When TECRTn pin input and TENCTn1 pin input occur simultaneously during up count



**Figure 18-36** Counter clearing to 0000H through Encoder clear input (pin TECRTn) timings (3/4)  
 When TECRTn pin input occurs earlier than TENCTn1 pin input during up count

No miscount occurs due to TECRTn pin input delay because the clear condition is set according to the levels of pins TENCTn0, TENCTn1 and TECRTn, and the counter is cleared to 0000H upon clear condition detection.



**Figure 18-37 Counter clearing to 0000H through encoder clear input (pin TECRTn) timings (4/4)**  
**When TECRTn pin input occurs later than TENCTn1 pin input during down count**

No miscount occurs due to the TECRTn pin input delay during down count, similarly to during up count.

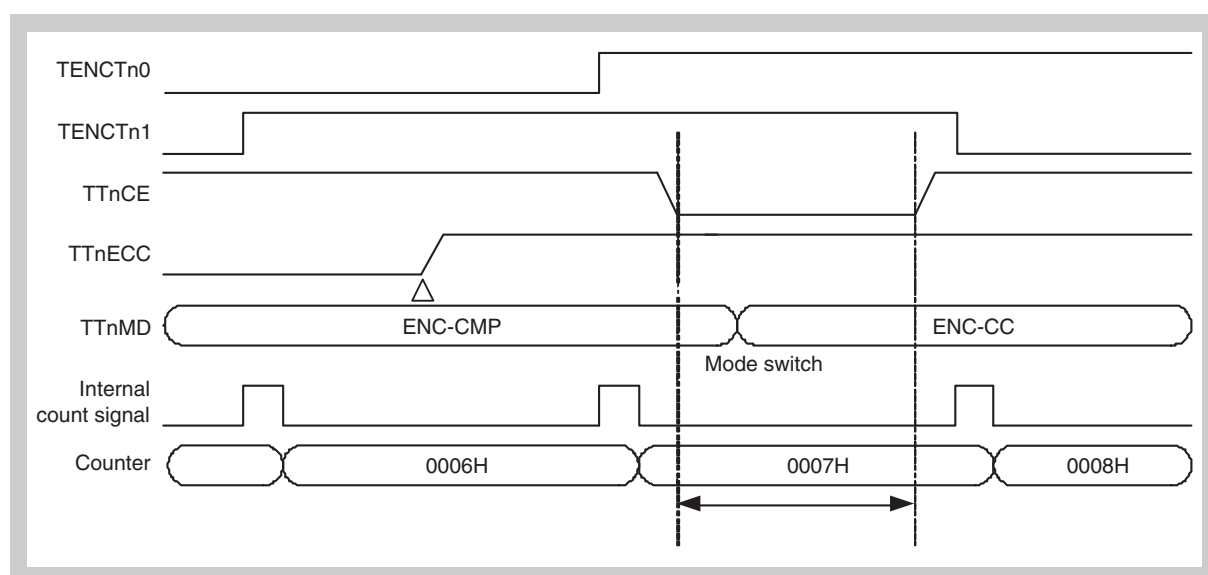
**(6) Counter hold through bit TTnECC**

By setting TTnCTL2 register bit TTnECC to 1, it is possible to switch to an encoder mode other than the currently operating mode during timer operation in the encoder compare mode, encoder capture mode, and encoder capture/compare mode. The capture function/compare function of the TTnCCR0 and TTnCCR1 registers can be switched by switching the operation mode.

**(a) Mode switching using bit TTnECC**

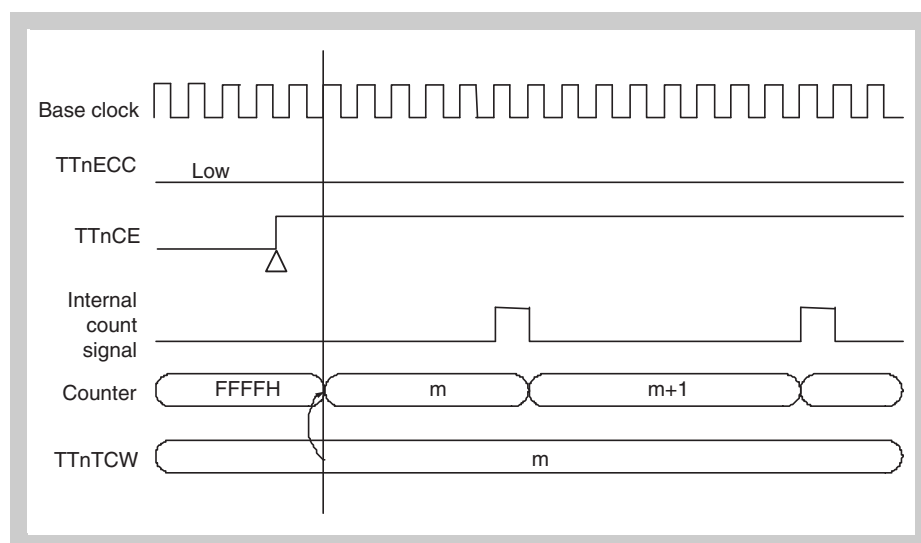
To switch the encoder mode while holding the counter value, be sure to set TTnECC = 1 before setting TTnCE = 0.

Since the counter gets reset if TTnCE = 0 is set with TTnECC = 0 left unchanged, the counter value cannot be held in this case.



**Figure 18-38 Counter hold through bit TTnECC timings (1/8)**  
**Mode switching using bit TTnECC**

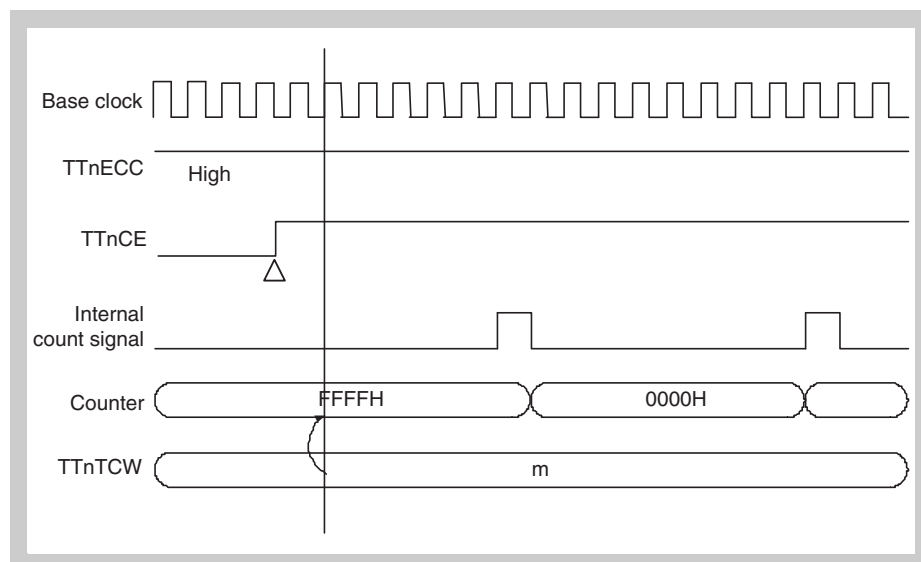
**Caution** To change the mode setting TTnCE = 0 once while the encoder itself is operating, perform initial setting processing within 1 cycle of the count clock. If switching is performed through bit TTnECC at the count signal output timing, miscount will occur.

**(b) Initial counter operation through bit TTnECC setting**

**Figure 18-39 Counter hold through bit TTnECC timings (2/8)**  
**Count operation when TTnECC = 0 is set**

The setting value of the TTnTCW register is loaded to the counter and count operation is performed from the setting value of the TTnTCW register.

(Initial value 0000H of TTnTCW register)



**Figure 18-40 Counter hold through bit TTnECC timings (2/8)**  
**Count operation when TTnECC = 1 is set**

Since the setting value of the TTnTCW register is not loaded to the counter, the count operation is performed from initial value FFFFH.

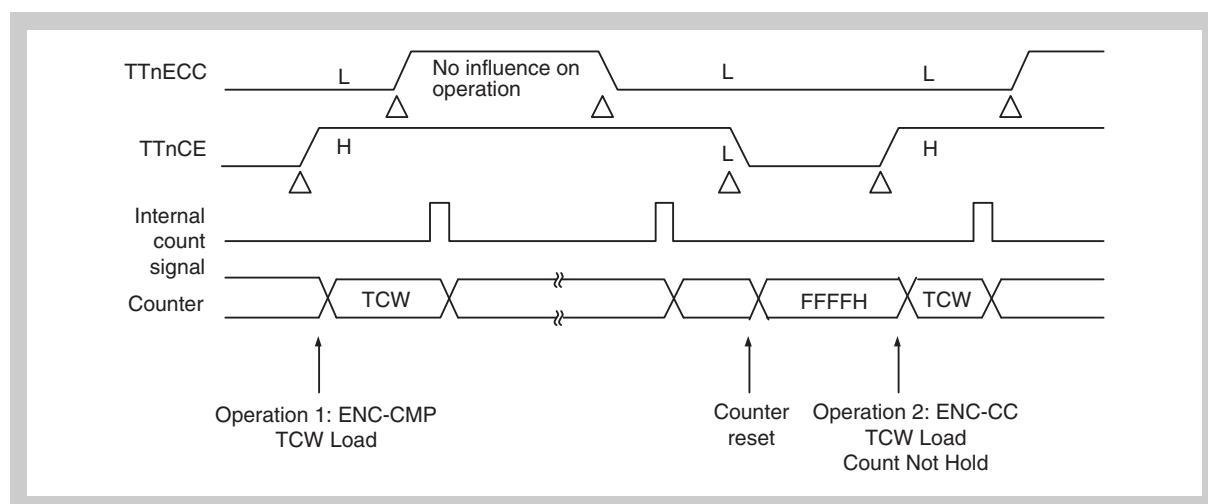
As the initial operation, it is recommended to set TTnECC = 0 and load to the counter the value set to the TTnTCW register, then start the count operation.

**(c) Bit TTnECC rewrite timing and its influence on counter**

1. When setting value of bit TTnECC is rewritten 0 → 1 → 0 when TTnCE = 1:  
Even if bit TTnECC rewrite is performed while TTnCE = 1, this has no influence on the counter operation.

Judgment as whether to hold or reset the counter value is performed while TTnCE = 0.

Moreover, judgment as to whether to load the setting value of the TTnTCW register to the counter is performed at the timing when the value of bit TTnCE changes from 0 to 1.

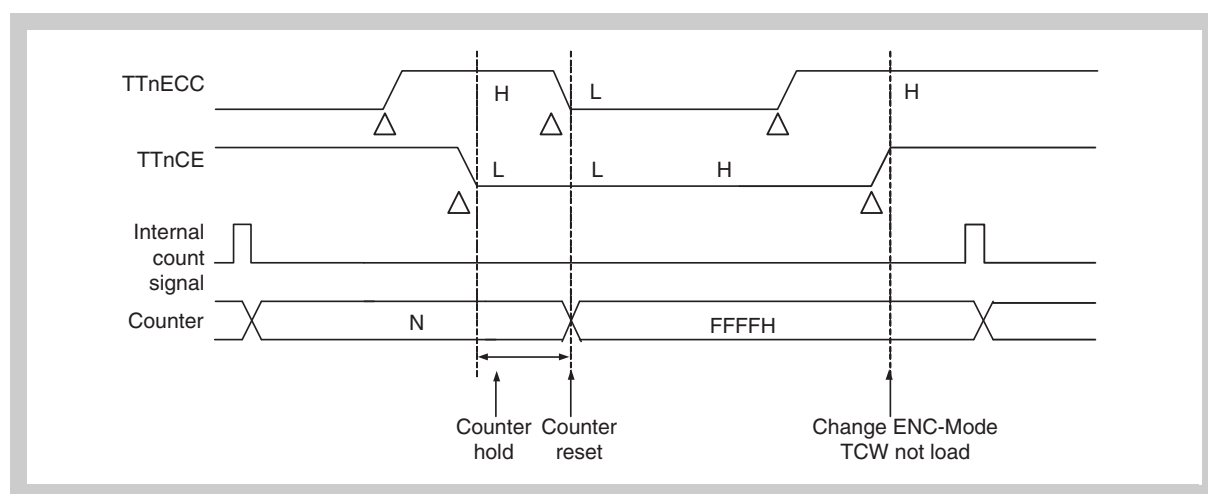


**Figure 18-41 Counter hold through bit TTnECC timings (3/8)**

When setting value of bit TTnECC is rewritten 0 → 1 → 0 when TTnCE = 1

2. When setting value of bit TTnECC is rewritten 1 → 0 → 1 while TTnCE = 0:  
The counter is reset when the setting value of bit TTnECC is changed from 1 to 0 while TTnCE = 0

Then, when TTnECC = 1 is set again and the value of bit TTnCE is changed from 0 to 1, counting restarts from the counter's initial value FFFFH, without the setting value of the TTnTCW being loaded to the counter.



**Figure 18-42 Counter hold through bit TTnECC timings (3/8)**

When setting value of bit TTnECC is rewritten 1 → 0 → 1 while TTnCE = 0

**(d) Rewrite timing of bit TTnECC**

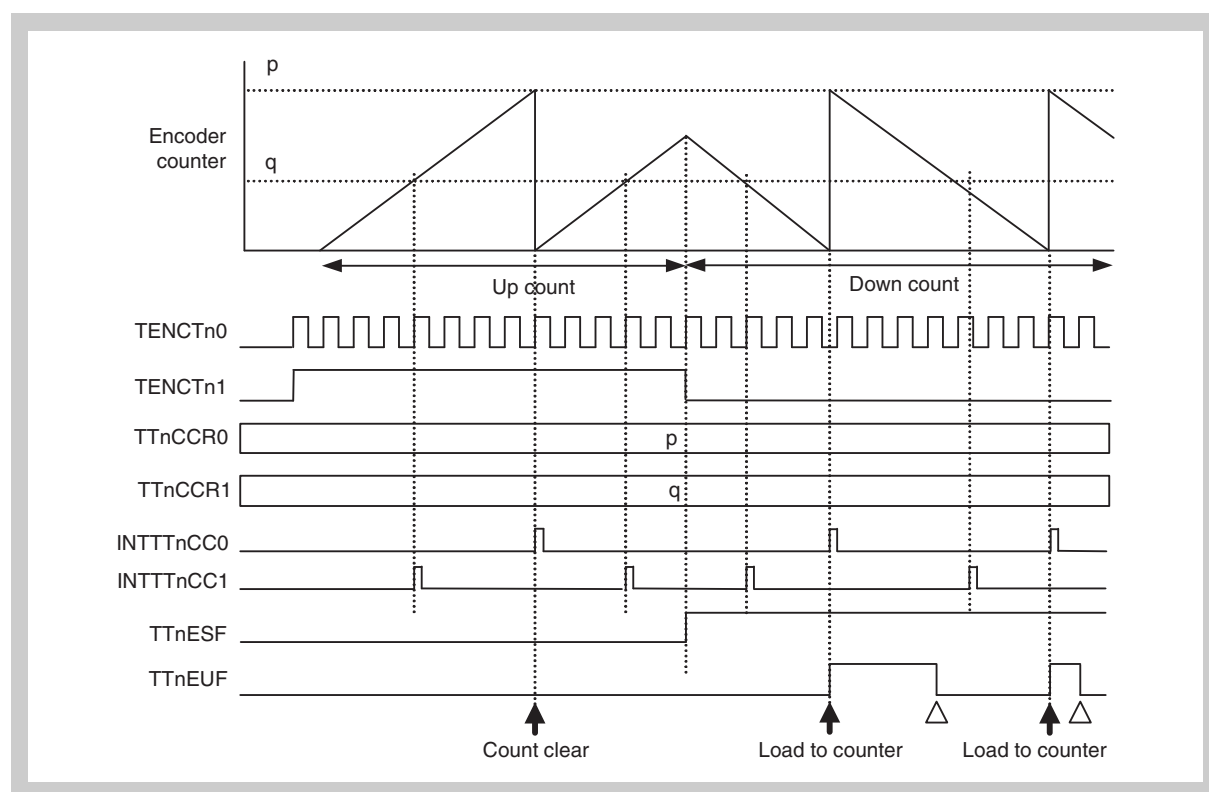
When  $TTnCE = 0$  and  $TTnECC = 0$ , setting  $TTnCE = 1$  causes the setting value of the  $TTnTCW$  register to be loaded to the counter.

Perform rewrite of the  $TTnECC$  bit after the operation clock has become valid (after several clocks: TBD), following setting of  $TTnCE = 1$ .

If bit  $TTnECC$  is rewritten before the operation clock becomes valid, counting starts from  $FFFFH$  without loading the setting value of the  $TTnTCW$  register to the counter.

## &lt; Register setting conditions &gt;

$TTnCTL0$ : $TTnMD3$ to $0 = 1000B$	Encoder compare mode
$TTnCTL1$ : $TTnUDS1, 0 = 00B$	Judgment of up/down count with count judgment mode 1
$TTnCTL1$ : $TTnECM1, 0 = 01B$	Counter clear upon match between counter value and $TTnCCR0$ buffer register
$TTnCTL1$ : $TTnLDE = 1$	Loading of setting value of $TTnCCR0$ register (p) upon underflow occurrence
$TTnIOC3$ : $TTnEIS1, 0 = 01B$	Detection of rising edge of $TENCTn0$ and $TENCTn1$ pin
$TTnIOC3$ : $TTnSCE = 0$ , $TTnECS1-0 = 00B$	Valid edge detection clear (no edge specified)



**Figure 18-43 Counter hold through bit  $TTnECC$  timings (4/8)**  
**Basic timing in encoder compare mode (1)**



Since TTnUDS1, 0 and TTnEIS1, 0 that control the count operation are set to 00B and 01B (rising edge detection), respectively, the counter is operated through detection of the phase of pin TENCTn1 upon detection of the rising edge of TENCTn0 pin input.

A compare match interrupt (INTTTnCC0) is output upon a match between the counter value and the TTnCCR0 compare register (p).

At this time, the counter is cleared to 0000H if the next count operation is up count.

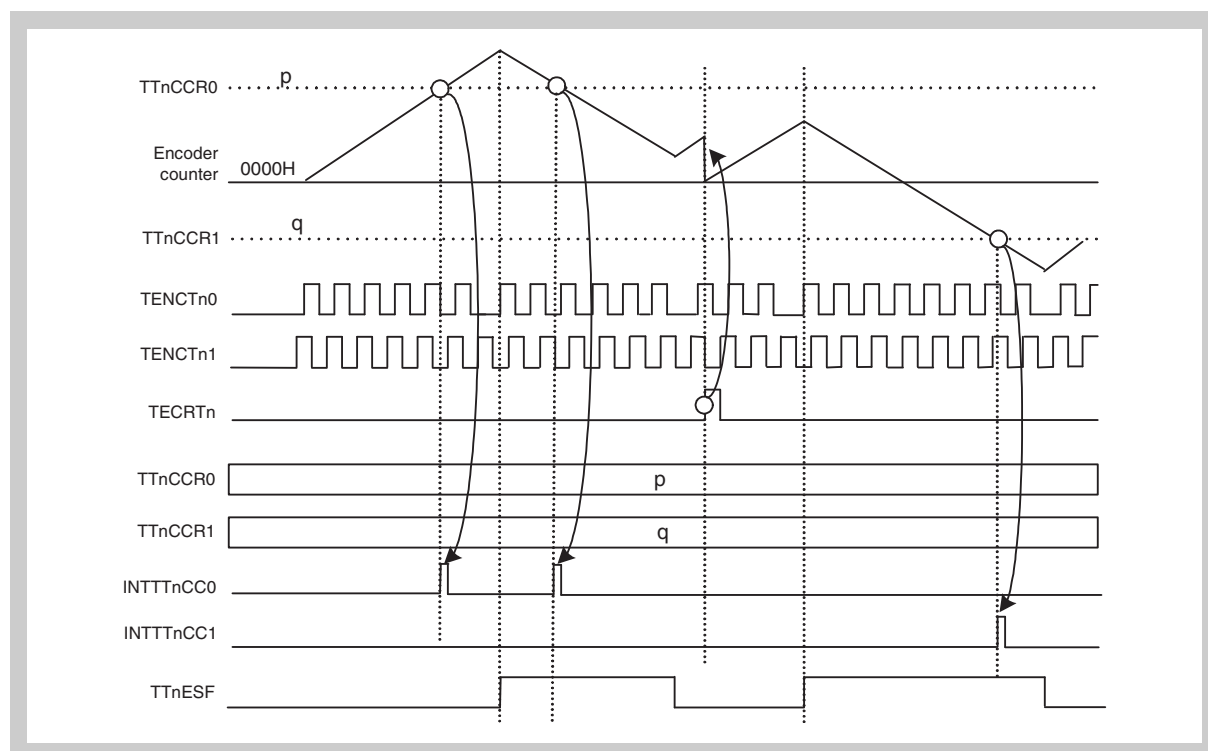
A compare match interrupt (INTTTnCC1) is output upon a match between the counter value and the TTnCCR1 buffer register (q).

The counter is not cleared upon a match between the counter value and the TTnCCR1 register.

If underflow occurs when TTnLDE = 1 is set, the setting value of the TTnCCR0 buffer register (m) is loaded to the counter. A count operation is possible between 0000H and the setting value of the TTnCCR0 register by setting TTnLDE = 1 and TTnECM0 = 1.

## &lt;Setting conditions&gt;

TTnCTL0:	Encoder compare mode
TTnMD3 to 0 = 1000B	
TTnCTL1:	Judgment of up/down count with count judgment mode 4
TTnUDS1, 0 = 11B	
TTnCTL1:	No clear operation upon match between counter value and compare
TTnECM1, 0 = 00B	
TTnCTL1:	No loading of setting value of TTnCCR0 register (p) to counter
TTnLDE = 0	
TTnIOC3:	Valid edge detection clear (rising edge specified)
TTnSCE = 0,	
TTnECS1-0 = 01B	



**Figure 18-44 Counter hold through bitTTnECC timings (5/8)**  
**Basic timing in encoder compare mode (2)**

Since TTnUDS1, 0 that control the count operation are set to 11B, the counter is operated through detection of the phase of pins TENCTn0 and TENCTn1.

A compare match interrupt (INTTTnCC0) is output upon a match between the counter value and the TTnCCR0 buffer register (p).

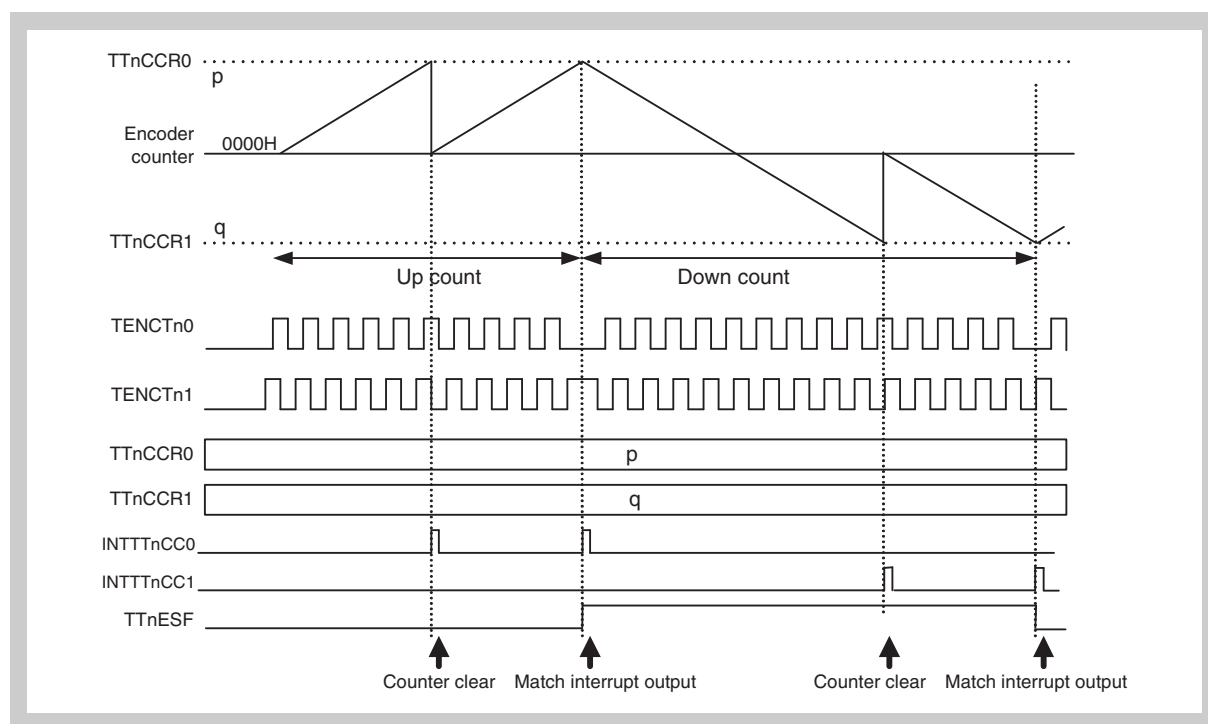
A compare match interrupt (INTTTnCC1) is output upon a match between the counter value and the TTnCCR1 buffer register (q).

The counter is not cleared upon a match with the TTnCCR0 register or the TTnCCR1 register.

Clearing of the counter to 0000H is done upon detection of the valid edge of the encoder clear input (pin TECRTn) when TTnSCE = 0. When TTnECS = 01B is set, the counter is cleared to 0000H in synchronization with the operation clock, following detection of the rising edge of the TECRTn pin input.

## &lt;Setting conditions&gt;

TTnCTL0:	Encoder compare mode
TTnMD3 to 0 = 1000B	
TTnCTL1:	Judgment of up/down count with count judgment mode 4
TTnUDS1, 0 = 11B	
TTnCTL1:	No clear operation upon match between counter value and
TTnECM1, 0 = 11B	and compare
TTnCTL1:	Counter clear upon match between counter value and
TTnLDE = 0	TTnCCR0 buffer register
	Counter clear upon match between counter value and
	TTnCCR1 buffer register
	(Since TTnCTL1: TTnECM1 to 0 = 11B, the setting of bit
	TTnLDE is invalid.)
TTnIOC3:	Valid edge detection clear (no edge specified)
TTnSCE = 0,	
TTnECS1-0 = 00B	



**Figure 18-45 Counter hold through bit TTnECC timings (6/8)**  
**Basic timing in encoder compare mode (3)**

Since TTnUDS1, 0 that control the count operation are set to 11B, the counter is operated through detection of the phase of pins TENCTn0 and TENCTn1.

A compare match interrupt (INTTnCC0) is output upon a match between the counter value and the TTnCCR0 buffer (p).

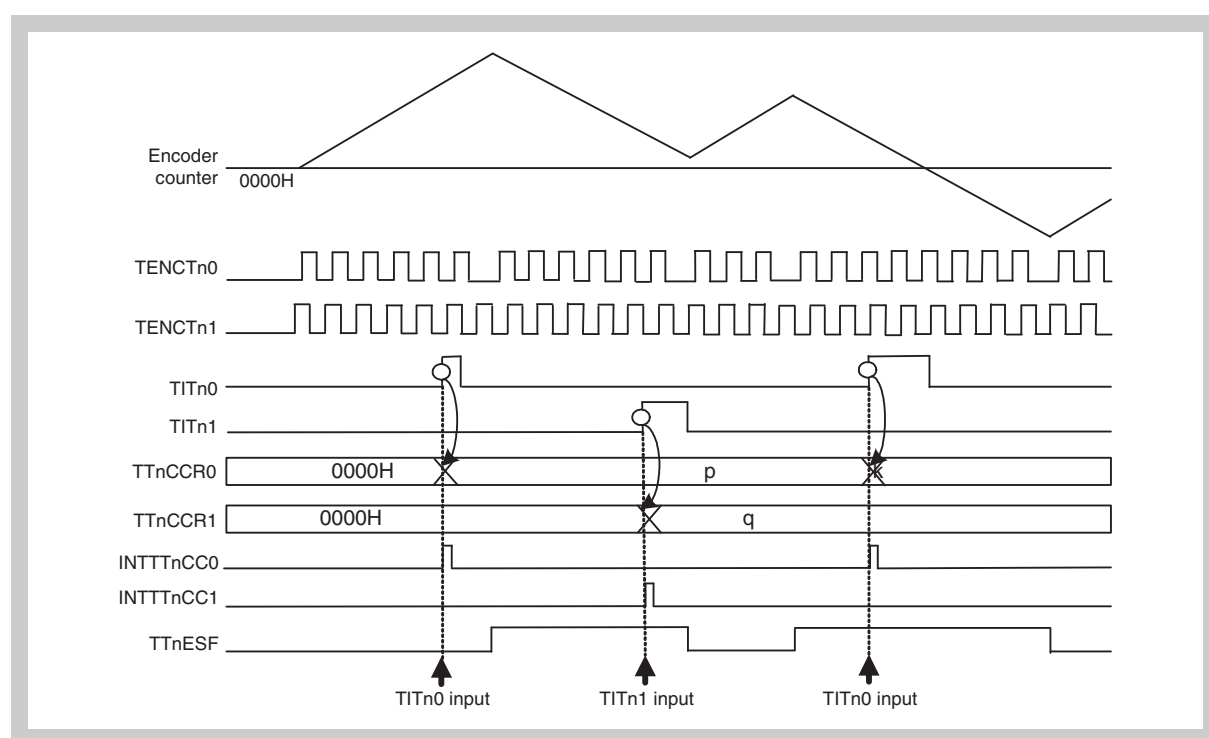
At this time, the counter is cleared to 0000H if the next count operation is up count.

A compare match interrupt (INTTnCC1) is output upon a match between the counter value and the TTnCCR1 buffer (q).

At this time, the counter is cleared to 0000H if the next count operation is down count.

## &lt;Setting conditions&gt;

TTnCTL0: TTnMD3 to 0 = 1001B	Encoder compare mode
TTnCTL1: TTnUDS1, 0 = 11B	Judgment of up/down count using count judgment method 4 (Since TTnCTL0: TTnMD3 to 0 = 1001B, the setting values of bits TTnECM1, TTnECM0, and TTnLDE are invalid.)
TTnIOC3: TTnSCE = 0, TTnECS1, 0 = 00B	Valid edge detection clear (no edge specified)
TTnOPT1: TTnIS1, 0 = 01B	Detection of rising edge of TITn0 pin input
TTnOPT1: TTnIS3, 2 = 01B	Detection of rising edge of TITn1 pin input



**Figure 18-46 Counter hold through bit TTnECC timings (7/8)  
Basic timing in encoder capture mode (1)**

Since TTnUDS1, 0 that control the count operation are set to 11B, the counter is operated through detection of the phase of pins TENCt0 and TENCt1.

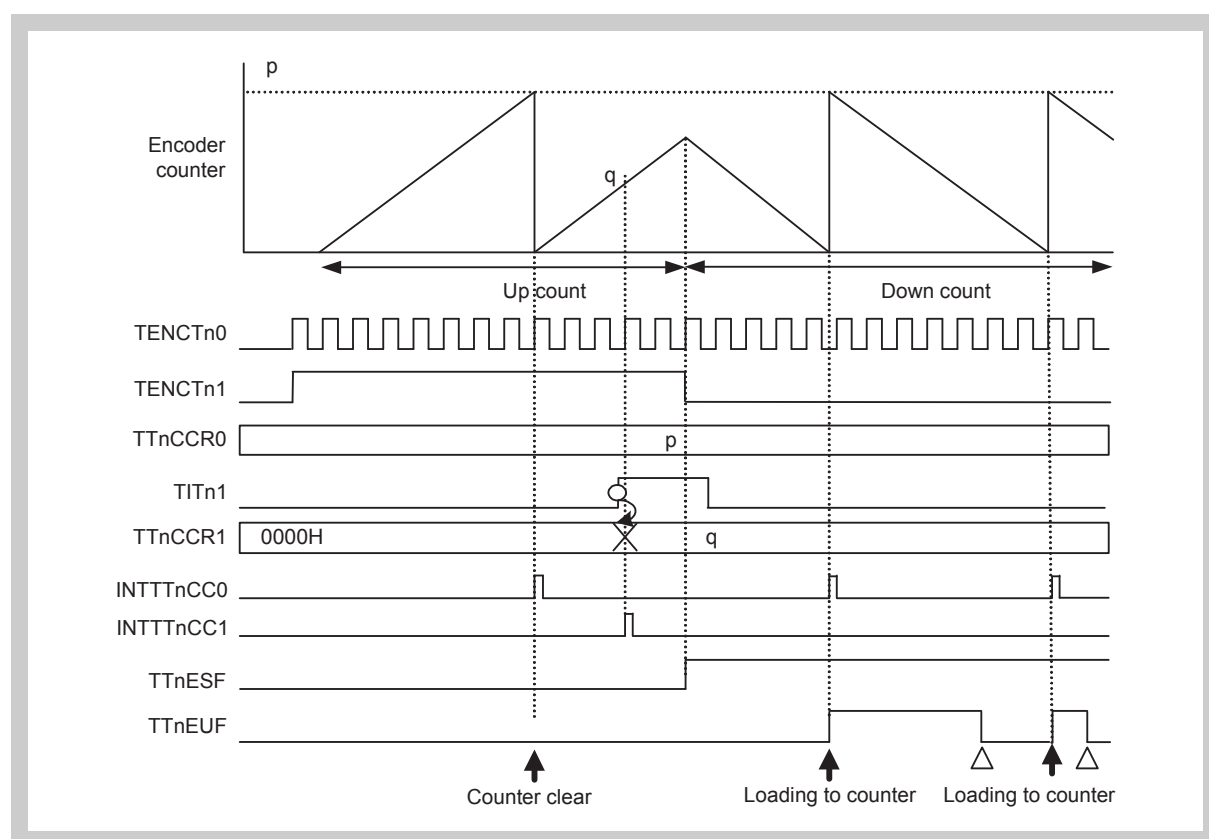
Upon detection of the edge of pin TITn0, the counter value is saved to the capture register (TTnCCR0), and a capture interrupt (INTTnCC0) is output.

Upon detection of the edge of TITn1, the counter value is saved to the capture register (TTnCCR1), and a capture interrupt (INTTnCC1) is output.

#### <Setting conditions>

TTnCTL0: TTnMD3 to 0 = 1010B	Encoder compare mode
TTnCTL1: TTnUDS1, 0 = 00B	Judgment of up/down count using count judgment mode 1

TTnCTL1: TTnECM1, 0 = 01B	Counter clear upon match between counter value and TTnCCR0 buffer register
TTnCTL1: TTnLDE = 1	Loading of setting value of TTnCCR0 register to counter upon occurrence of underflow
TTnIOC3: TTnEIS1, 0 = 01B	Detection of rising edge of TENCTn0 and TENCTn1 pin inputs
TTnIOC3: TTnSCE = 0, TTnECS1, 0 = 00B	Valid edge detection clear (no edge specified)
TTnOPT1: TTnIS3, 2 = 01B	Detection of rising edge of TITn1 pin input



**Figure 18-47 Counter hold through bit TTnECC timings (8/8)**  
**Basic timing in encoder capture compare mode (1)**

Since TTnUDS1, 0 and TTnEIS1, 0 that controls the count operation are set to 00B and 01B (rising edge detection), respectively, the counter operates through detection of the rising edge of the TENCTn0 pin input and detection of the phase of the TENCTn1 pin.

A compare match interrupt (INTTTnCC0) is output upon a match between the counter value and the TTnCCR0 buffer register (m). At this time, the counter is cleared to 000H if the next count operation is up count.

Upon detection of the edge of the TITn1 pin input, the counter value is saved to the capture register (TTnCCR1), and a capture interrupt (INTTTnCC1) is output.

If an underflow occurs due to the TTnLDE = 1 setting, the setting value of the TTnCCR0 buffer register is loaded to the counter. The count operation is possible between 0000H and the setting value of the TTnCCR0 register by setting TTnLDE = 1 and TTnECM0 = 1.

### 18.6.10 Offset trigger generation mode

In the offset trigger generation mode, the count value is saved to the capture register (TTnCCR0) upon detection of the valid edge of the TITn0 pin, and a capture interrupt (INTTTnCC0) is output. The counter is cleared to 0000H by capture input. (Counter clear operation is not performed using the TTnCCR1 register.)

The TTnCCR0 register and the TTnCCR1 register have their functions fixed as a capture register and a compare register, respectively. The TTnCCR1 register can be rewritten during count operation. Regarding compare register reload, the capture & clear timing upon detection of TITn0 pin input serves as the reload timing.

During count operation, a capture interrupt (INTTTnCC0) is output upon capture to the TTnCCR0 register through TITn0 pin input, and a compare match interrupt (INTTTnCC1) is output upon a match between the counter and the TTnCCR1 register.

The TOTn0 pin becomes the level set with bit TTnOL0. If TTnOL0 = 0, a low level is output and if TTnOL0 = 1, a high level is output.

The TOTn1 pin is reset upon a match between the counter and the TTnCCR1 register, and is set when the counter is cleared to 0000H.

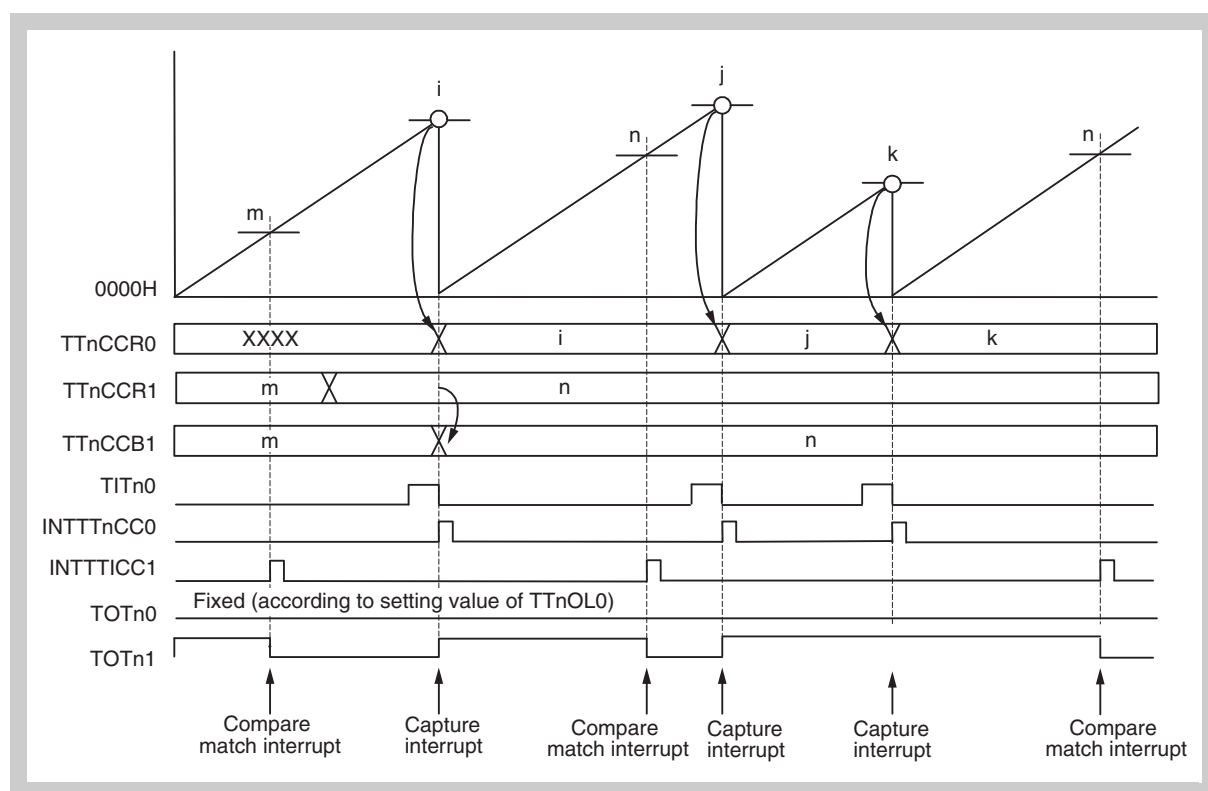


Figure 18-48 Basic timing in offset trigger generation mode

In the offset trigger generation mode, the setting value of the TTnCCR1 register is reloaded to the TTnCCR1 buffer register upon detection of the valid edge of pin TITn0. Until the edge of the TITn0 pin input is detected, the value of the TTnCCR1 register is not reloaded to the TTnCCR1 buffer register, even if this value is changed.

Pin TOTn1 is set when the counter is cleared to 0000H upon detection of the valid edge of pin TITn0, and it is reset upon a match between the counter value and the TTnCCR1 register.

Therefore, pin TOTn1 remains high level if the valid edge of the TITn0 pin input is detected before a match with the TTnCCR1 register occurs.

## 18.7 Cautions

### 18.7.1 TMT 'Encoder load enable'-mode limitation

**Description** If TMT operates

- in encoder compare mode (TTnCTL1.TTnMD[3:0] = 1000<sub>B</sub>)
- or in encoder capture/compare mode (TTnCTL1.TTnMD[3:0] = 1010<sub>B</sub>),
- and the 'Encoder load enable'-mode is set (TTnCTL2.TTnLDE = 1)

while the TMT operation is enabled (TTnCTL0.TTnCE = 1), an undefined value might be loaded into the counter (TTnCNT) when the TTnCCR0 register is rewritten during counter underflow.

**Workaround** If the TMT 'Encoder load enable'-mode is set (TTnCTL2.TTnLDE = 1) during TMT operation in encoder compare mode (TTnCTL1.TTnMD[3:0] = 1000<sub>B</sub>) or in encoder capture/compare mode (TTnCTL1.TTnMD[3:0] = 1010<sub>B</sub>) apply either of the following workarounds:

- Disable the TMT (TTnCTL0.TTnCE = 0) before rewriting the TTnCCR0 register.
- Ensure that the TTnCCR0 register is not rewritten during a possible underflow of the counter (TTnCNT).

## Chapter 19 Timer Interconnection

Various functions can be implemented by connecting the timer I/O and the internal signal between timers.:

- 2-phase encoder function
- 3-phase pulse input control function
- 3-phase encoder function
- TMT as Filter for TMAA

The timer I/O control register (TAAIC3) is used for signal connection between timers. The following shows the timer I/O control registers according to their functions.





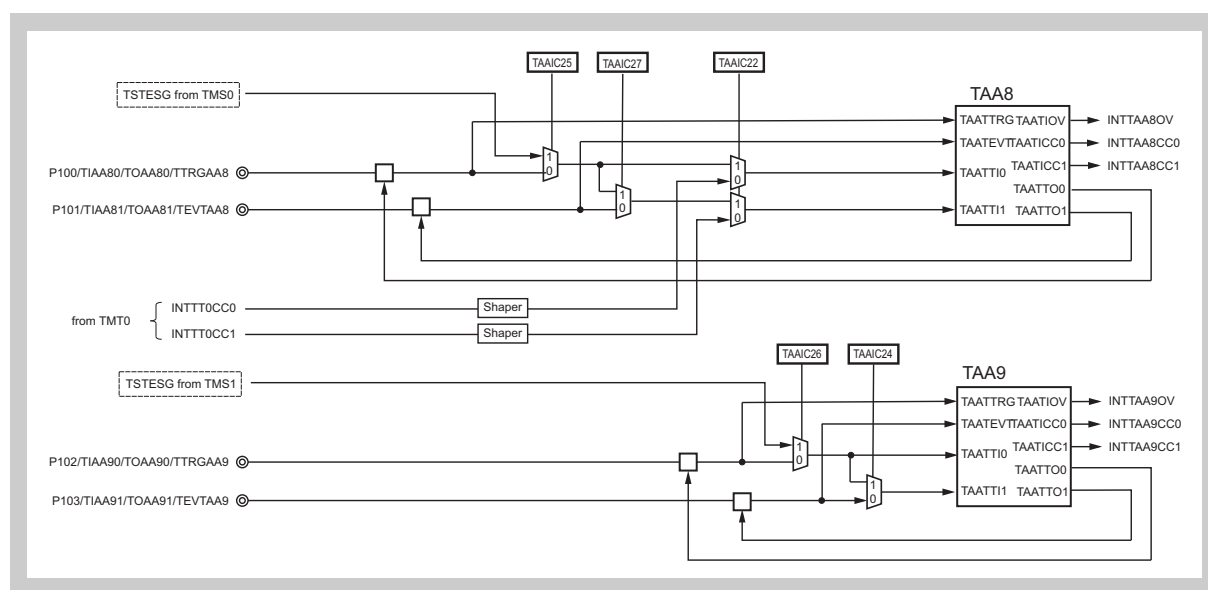


Figure 19-2 Timer AA configuration

## 19.1 Control register

### (1) TAAIC3 - Timer I/O control register 3

The TAAIC3 register selects the timer input signals for enhanced timer control.

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

**Address** FFFF F6F3<sub>H</sub>

**Initial Value** 00<sub>H</sub>

7	6	5	4	3	2	1	0
TAAIC37	TAAIC36	TAAIC35	TAAIC34	TAAIC33	TAAIC32	TAAIC31	TAAIC30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 19-1 TAAIC3 register contents (1/2)**

Bit position	Bit name	Function
Bit position	Bit name	Function
7	TAAIC37	Selection of TMS1 TSTSTIC0/TSTSTIC1 input signal 0: "L" signal level TMS1 TSTSTIC1 TMS1 TSTSTIC0: input selected by TAAIC33 bit 1: TMS1 TSTSTIC0: TMT0 TTTEQC1 TMS1 TSTSTIC1: TMT1 TTTEQC1
6	TAAIC36	Selection of TMS0 TSTSTIC0/TSTSTIC1 input signal 0: "L" signal level TMS0 TSTSTIC1 TMS0 TSTSTIC0: input selected by TAAIC31 bit 1: TMS0 TSTSTIC0: TMT0 TTTEQC1 TMS0 TSTSTIC1: TMT1 TTTEQC1
5	TAAIC35	Signal selection input to selector controlled by TAAIC34 bit 0: TMS0 TSTAEDO 1: TMS1 TSTAEDO
4	TAAIC34	Selection of TMTn (n = 0 - 1) TTTECR/TTTTI0 input signal 0: "L" level of "L" level of TMS1 0: TMT0 TTTECR:TECRT0 TMT1 TTTECR:TECRT1 TMT0 TTTTTI0:TIT00 TMT1 TTTTTI0:TIT10 1: TMT1 TTTTTI0: selected by TAAIC35 bitt TMT0 TTTTTI0: selected by TAAIC35 bitt TMT1 TTTECR: selected by TAAIC35 bitt TMT0 TTTECR: selected by TAAIC35 bitt
3	TAAIC33	Selection of TMS1 TSTARD 120 degree energizing pattern switch sequence control signal and Signal selection input to selector controlled by TAAIC37 bit. 0: "L" level of "L" level at TMS1 TSTARD input TAAIC37 selector input (for TMS1 TSTSTIC1) 1: TMS1 TSTARD:TTTTO0 (from TMT0) TAAIC37 selector (for TMS1 TSTSTIC0): TTTEQC0 or TTTEQC1 from TMT1, selected it with TTTTTO0
2	TAAIC32	Selection of TMT1 TTTENC0/TTTENC1 encoder input signal 0: TMT1 TTTENC0:TIT11/TEVTT0/TENCT10 TMT1 TTTENC1:TIT10/TTRGT0/TENCT11 1: TMT1 TTTENC0:TSTESG (from TMS1) TMT1 TTTENC1:TSTTSF(from TMS1)

**Table 19-1 TAAIC3 register contents (2/2)**

Bit position	Bit name	Function
1	TAAIC31	Selection of TMS0 TSTARD 120 degree energizing pattern switch sequence control signal and Signal selection input to selector controlled by TAAIC36 bit 0: "L" level of "L" level at TMS0 TSTARD input TAAIC36 selector input (for TMS0 TSTSTCI1) 1: TMS0 TSTARD:TTTTO0 (from TMT0) TAAIC36 selector (for TMS0 TSTSTCI0): TTTEQC0 or TTTEQC1 from TMT0 selected with TTTTO0
0	TAAIC30	Selection of TMT0 TTTENC0/TTTENC1 encoder input signal 0: TMT0 TTTENC0:TIT00/TEVTT1/TENCT00 TMT0 TTTENC1:TIT01/TTRGT1/TENCT01 1: TMT0 TTTENC0:TSTESG(from TMS0) TMT0 TTTENC1:TSTTSF(from TMS0)

## 19.2 2-Phase Encoder Control Function

### 19.2.1 Function overview

The function enables shifting of the timer S output pattern, such as 120° conducting control by use of the 2-phase encoder.

**Table 19-2 Control methods for 2-phase input control**

Method	Encoder counter count cycle	Encoder counter clear method
Control method 1	1 motor rotation	Clear by Z-phase input
Control method 2	Arbitrary cycle	Arbitrary value
Control method 3	Count value to shift timer S pattern	Clear by compare value (count value to shift pattern)

The internal signals used in the 2-phase control function are shown below.

**Table 19-3 Internal signals used in 2-phase encoder control function**

Timer	Signal name	I/O	Function
Timer S	TSnARD	I	Controls the TOSn1 to TOSn6 pin output pattern order. Low level: Shifts the output pattern in normal rotation order. High level: Shifts the output pattern in reverse rotation order.
	TSnSTCI0	I	Output pattern shift trigger
	TSnSTCI1	I	Low level should be input in the 2-phase encoder control function.
	TSnADTRG0	O	Conversion start trigger signal for A/D converter 0. When the A/D converter is used in timer trigger mode, the A/D conversion start trigger is set regardless of the signal connection setting.
	TSnADTRG1	O	Conversion start trigger signal for A/D converter 0. When the A/D converter is used in timer trigger mode, the A/D conversion start trigger is set regardless of the signal connection setting.
Timer T	TTmENC0	I	Encoder A-phase input signal
	TTmENC1	I	Encoder B-phase input signal
	TTmECR	I	Encoder Z-phase input signal
	TTmTI0	I	Cycle measurement capture input signal
	TTmTI1	I	Position detection capture input signal
	INTTTmCC0	O	Timer Tm TTmCCR0 compare match interrupt request signal
	INTTTmCC1	O	Timer Tm TTmCCR1 compare match interrupt request signal

**Note** n = 0, 1, m = 0 to 5

## 19.2.2 Configuration

2-phase input control/position detection

### (1) Control method 1

The following three types of timer configurations enable 2-phase input control/position detection control.

**Table 19-4 Control Method 1 Timer Configuration (1)**

Timer	Function	Block diagram	Mode
Timer T0	Encoder position detection		<ul style="list-style-type: none"> <li>•Encoder capture compare mode</li> </ul>
Timer S0	Motor control (position detection)		<p>Any of the following modes</p> <ul style="list-style-type: none"> <li>•High-accuracy T-PWM mode and semi-automatic drive 180° conduction control function</li> <li>•120° conduction mode</li> <li>Special 120° conduction mode</li> </ul>

Table 19-5 Control Method 1 Timer Configuration (2)

Timer	Function	Block diagram	Mode
Timer T4	Encoder position detection		<ul style="list-style-type: none"> <li>Encoder capture compare mode</li> </ul>
Timer S0	Motor control (position detection)		<p>Any of the following modes</p> <ul style="list-style-type: none"> <li>High-accuracy T-PWM mode and semi-automatic drive 180° conduction control function</li> <li>120° conduction mode</li> <li>Special 120° conduction mode</li> </ul>

## 19.3 3-Phase Pulse Input Control Function

### 19.3.1 Function overview

Phase variable control for timer S pattern output is enabled by combining the 3-phase pulse input (TAPTSn2 to TAPTSn0 pin input) of the timer S0/S1 and timer S connection mode of timer T ( $n = 0, 1$ ).

- Phase fix control: Outputs the pattern fixed at a certain rotation angle.
- Phase variable control: Outputs the pattern where the phase of angle (time) discretionary to the rotation angle is varied.

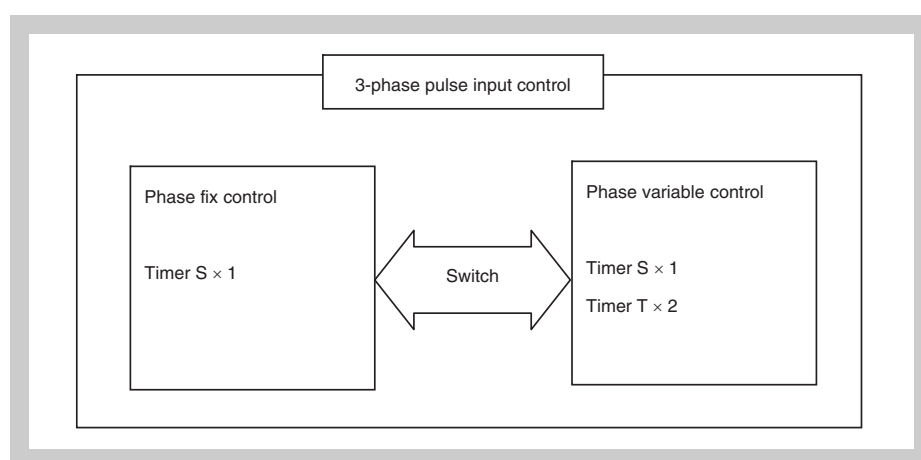


Figure 19-3 Phase fix control and phase variable control

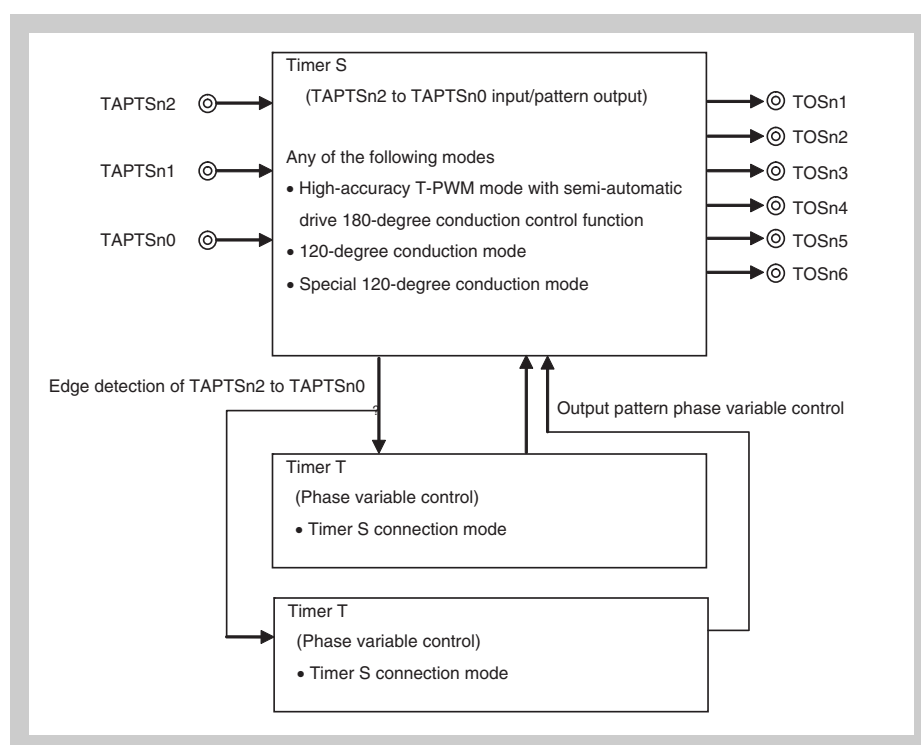


Figure 19-4 Block diagram of 3-phase pulse input control



## 19.3.2 Configuration

### (1) 3-phase pulse input control 1

Table 19-6 3-phase pulse input control 1 timer configuration

Timer	Function	Block diagram	Setting	
			Synchronous function	Operation mode
Timer T0	Offset control		Simultaneous start (master)	Timer S connection mode
Timer T1			Simultaneous start (slave)	
Timer S0	Pattern output		Simultaneous start (slave)	Any of the following modes <ul style="list-style-type: none"> <li>• High-accuracy T-PWM mode and semi-automatic drive 180° conduction control function</li> <li>• 120° conduction mode</li> <li>• Special 120° conduction mode</li> </ul>

### (2) 3-phase pulse input control 2

Table 19-7 3-phase pulse input control 2 timer configuration

Timer	Function	Block diagram	Setting	
			Synchronous function	Operation mode
Timer T0	Offset control		Simultaneous start (master)	Timer S connection mode
Timer T1			Simultaneous start (slave)	
Timer S1	Pattern output		Simultaneous start (slave)	Any of the following modes <ul style="list-style-type: none"> <li>• High-accuracy T-PWM mode and semi-automatic drive 180° conduction control function</li> <li>• 120° conduction mode</li> <li>• Special 120° conduction mode</li> </ul>

## 19.4 3-Phase Encoder Function

### 19.4.1 Function overview

The edges of the signals input to the TAPTS02 to TAPTS00 pins can be connected to any one of timers T0 to T3 to be counted.

### 19.4.2 Configuration

Table 19-8 Block diagram of 3-phase encoder (1)

Timer	Function	Block diagram	Mode
Timer T0 Timer AA8	Encoder		Any of the following modes <ul style="list-style-type: none"> <li>Encoder compare mode</li> <li>Encoder capture mode</li> <li>Encoder capture compare mode</li> </ul>
Timer S0	TAPTS02 to TAPTS00 input		Any of the following modes <ul style="list-style-type: none"> <li>High-accuracy T-PWM mode and semi-automatic drive 180° conduction control function</li> <li>120° conduction mode</li> <li>Special 120° conduction mode</li> </ul>

**Table 19-9 Block diagram of 3-phase encoder (2)**

Timer	Function	Block diagram	Mode
Timer T0 Timer AA9	Encoder		Any of the following modes <ul style="list-style-type: none"> <li>Encoder compare mode</li> <li>Encoder capture mode</li> <li>Encoder capture compare mode</li> </ul>
Timer S1	TAPTS02 to TAPTS00 input		Any of the following modes <ul style="list-style-type: none"> <li>High-accuracy T-PWM mode and semi-automatic drive 180° conduction control function</li> <li>120° conduction mode</li> <li>Special 120° conduction mode</li> </ul>

---

## 19.5 TMT as Filter for TMAA

To support an extended filtering function (i.e. jittering input signals) for the input of TMAA8, there is a connectivity possible between TMT0 and TMAA8 .

Via this connectivity the INTTnCC0/1 interrupt signals can be chosen as capture-inputs for the CCR0/1 register of TMAA8 (refer also to the description of TAAIC2.TAAIC22 in *“Input Selection Registers” on page 549*).

# Chapter 20 On-Chip Debug Unit

The microcontroller includes an on-chip debug unit. By connecting an N-Wire emulator, on-chip debugging can be executed.

## 20.1 Functional Outline

### 20.1.1 Debug functions

**(1) Debug interface**

Communication with the host machine is established by using the  $\overline{\text{DRST}}$ , DCK, DMS, DDI, and DDO signals via an N-Wire emulator. The communication specifications of N-Wire are used for the interface.

**(2) On-chip debug**

On-chip debugging can be executed by preparing wiring and a connector for on-chip debugging on the target system. An N-Wire emulator is used to connect the host PC to the on-chip debug unit.

**(3) Forced reset function**

The microcontroller can be forcibly reset.

**(4) Break reset function**

The CPU can be started in the debug mode immediately after reset of the CPU is released.

**(5) Forced break function**

Execution of the user program can be forcibly aborted.

**(6) Hardware break function**

Two breakpoints for instruction and data access can be used. The instruction breakpoint can abort program execution at any address. The access breakpoint can abort program execution by data access to any address.

**(7) Software break function**

Up to eight software breakpoints can be set in the internal flash memory area. The number of software breakpoints that can be set in the RAM area differs depending on the debugger to be used.

**(8) Debug monitor function**

A memory space for debugging that is different from the user memory space is used during debugging (background monitor mode). The user program can be executed starting from any address.

While execution of the user program is aborted, the user resources (such as memory and I/O) can be read and written, and the user program can be downloaded.

**(9) Mask function**

Each of the following signals can be masked. That means these signals will not be effective during debugging.

The correspondence with the mask functions of the debugger (ID850NWC) for the N-Wire emulator (IE-V850E1-CD-NW) of Renesas Electronics is shown below.

- NMI0 mask function: NMI pin
- $\overline{\text{RESET}}$  mask function: external  $\overline{\text{RESET}}$  pin
- $\overline{\text{WAIT}}$  mask function: external  $\overline{\text{WAIT}}$  pin

**(10) Timer function**

The execution time of the user program can be measured.

## 20.1.2 Security function

This microcontroller has a N-Wire security function, that demands the user to input an ID code upon start of the debugger. The ID code is compared to a predefined ID code, written in advance to the internal flash memory by an external flash programmer. This function prevents unauthorized persons to operate the microcontroller in N-Wire debug mode and to read the internal flash memory area.

The ID code in the internal flash memory can only be written by an external flash programmer. It can't be changed in self-programming mode and therefore also not in N-Wire debugging mode.

**ID code** Be sure to write an ID code when writing a program to the internal flash memory.

The area of the ID code is 10 bytes wide and in the range of addresses 0000 0070<sub>H</sub> to 0000 0079<sub>H</sub>.

The ID code when the memory is erased is shown below.

Address	ID code
0000 0079 <sub>H</sub>	FF <sub>H</sub>
0000 0078 <sub>H</sub>	FF <sub>H</sub>
0000 0077 <sub>H</sub>	FF <sub>H</sub>
0000 0076 <sub>H</sub>	FF <sub>H</sub>
0000 0075 <sub>H</sub>	FF <sub>H</sub>
0000 0074 <sub>H</sub>	FF <sub>H</sub>
0000 0073 <sub>H</sub>	FF <sub>H</sub>
0000 0072 <sub>H</sub>	FF <sub>H</sub>
0000 0071 <sub>H</sub>	FF <sub>H</sub>
0000 0070 <sub>H</sub>	FF <sub>H</sub>

**Security bit** Bit 7 of address 0000 0079<sub>H</sub> enables or disables use of the N-Wire emulator.

- Bit 7 of address 0000 0079<sub>H</sub>
  - 0: disabled N-Wire emulator cannot connect to the on-chip debug unit.
  - 1: enabled N-Wire emulator can connect to the on-chip debug unit if the 10-byte ID code input matches the ID code stored in the flash memory

This security bit can only be modified by programming the flash memory via an external flash programmer. It is not possible to modify the security bit in self-programming mode, and therefore also not in N-Wire debugging mode.

After reset, the entire ID code area is set to FF<sub>H</sub>. This means that

- N-Wire debugging is generally enabled
- the ID code is FF<sub>H</sub> for all ID code bytes

Consequently, controller access is possible without any restriction.

---

**Caution** If access via the N-Wire interface should be disabled "block erase disabled" should be configured as well. Otherwise the flash memory blocks containing the ID code could be erased and N-Wire access could be enabled.

---



## 20.2 Connection to N-Wire Emulator

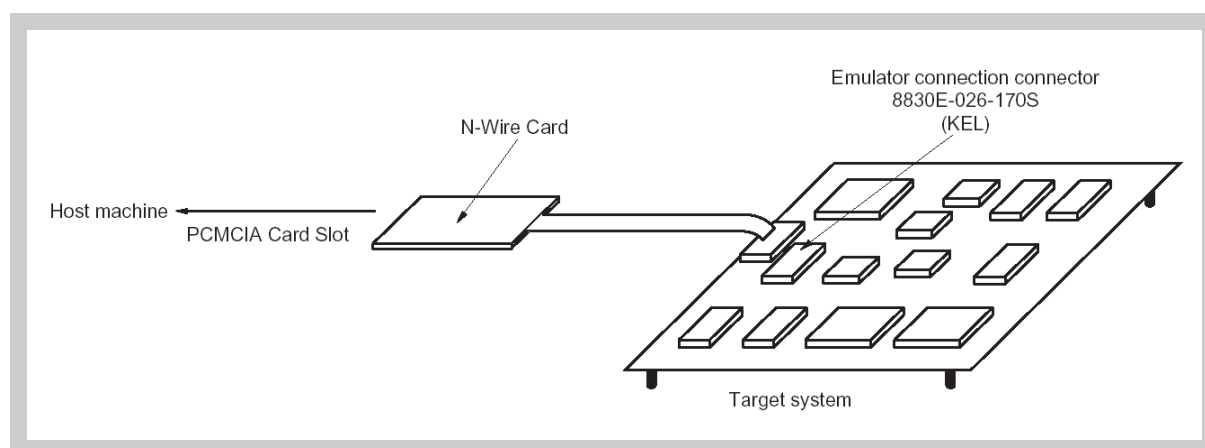
To connect the N-Wire emulator, a connector for emulator connection and a connection circuit must be mounted on the target system.

As a connector example the KEL connector is described in more detail. Other connectors, like for instance MICTOR connector (product name: 2-767004-2, Tyco Electronics AMP K.K.), are available as well. For the mechanical and electrical specification of these connectors refer to user's manual of the emulator to be used.

### 20.2.1 KEL connector

KEL connector product names:

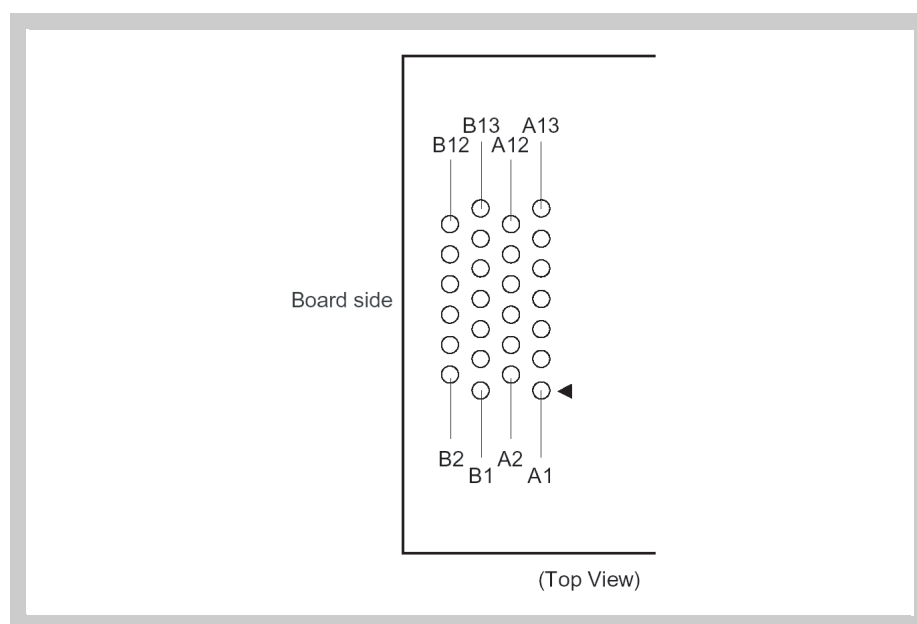
- 8830E-026-170S (KEL): straight type
- 8830E-026-170L (KEL): right-angle type



**Figure 20-1** Connection to N-Wire emulator (Renesas Electronics IE-V850E1-CD-NW: N-Wire Card)

**(1) Pin configuration**

Figure 20-2 shows the pin configuration of the connector for emulator connection (target system side), and Table 20-1 on page 1039 shows the pin functions.



**Figure 20-2** Pin configuration of connector for emulator connection (target system side)

**Caution** Evaluate the dimensions of the connector when actually mounting the connector on the target board.

**(2) Pin functions**

The following table shows the pin functions of the connector for emulator connection (target system side). "I/O" indicates the direction viewed from the device.

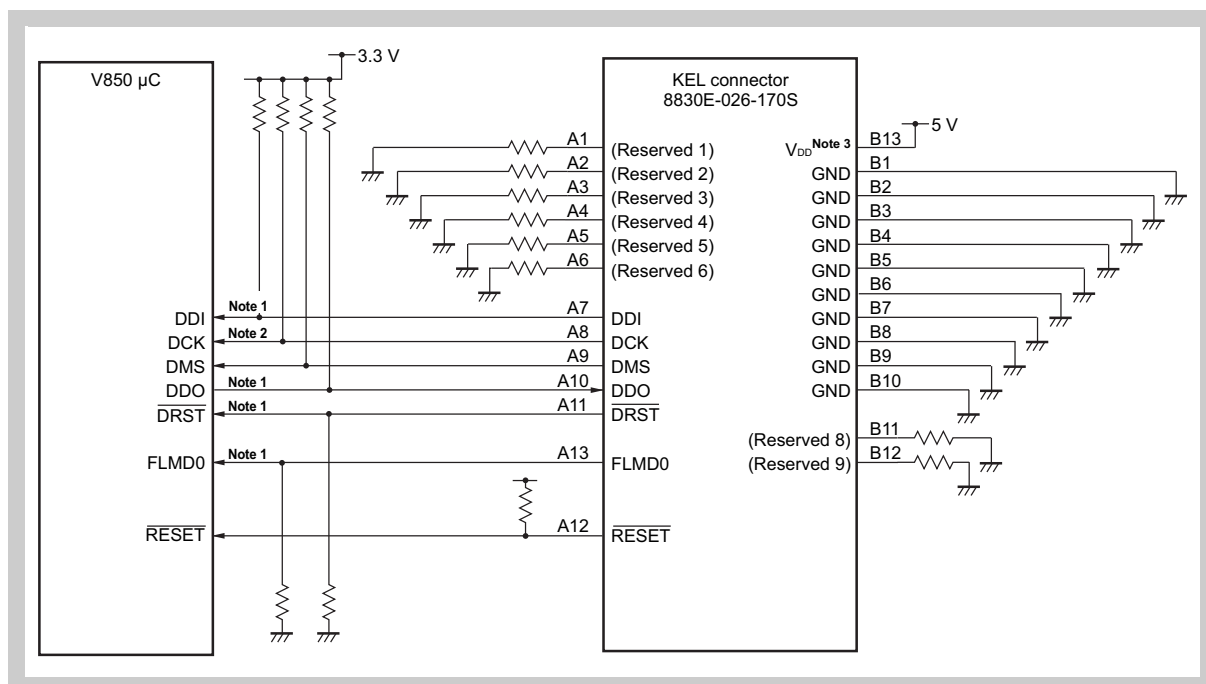
**Table 20-1 Pin functions of connector for emulator connection (target system side)**

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	Input	Data input for N-Wire interface
A8	DCK	Input	Clock input for N-Wire interface
A9	DMS	Input	Transfer mode select input for N-Wire interface
A10	DDO	Output	Data output for N-Wire interface
A11	$\overline{\text{DRST}}$	Input	On-chip debug unit reset input
A12	$\overline{\text{RESET}}$	Input	Reset input.
A13	FLMD0	Input	Control signal for flash download (flash memory versions only)
B1	GND	—	—
B2	GND	—	—
B3	GND	—	—
B4	GND	—	—
B5	GND	—	—
B6	GND	—	—
B7	GND	—	—
B8	GND	—	—
B9	GND	—	—
B10	GND	—	—
B11	(Reserved 8)	—	(Connect to GND)
B12	(Reserved 9)	—	(Connect to GND)
B13	V <sub>DD</sub>	—	3.3 V input (for monitoring power supply to target)

- Caution**
1. The connection of the pins not supported by the microcontroller is dependent upon the emulator to be used.
  2. The pattern of the target board must satisfy the following conditions.
    - The pattern length must be 100 mm or less.
    - The clock signal must be shielded by GND.

### (3) Example of recommended circuit

An example of the recommended circuit of the connector for emulator connection (target system side) is shown below.



**Figure 20-3** Example of recommended emulator connection circuit

- Note**
1. The pattern length must be 100 mm or less.
  2. Shield the DCK signal by enclosing it with GND.
  3. This pin is used to detect power to the target board. Connect the voltage of the N-Wire interface to this pin.

**Caution** The N-Wire emulator may not support a 5 V interface and may require a level shifter. Refer to the user's manual of the emulator to be used.

## 20.3 Restrictions and Cautions on On-Chip Debug Function

- Do not mount a device that was used for debugging on a mass-produced product (this is because the flash memory was rewritten during debugging and the number of rewrites of the flash memory cannot be guaranteed).
- The  $\overline{\text{RESET}}$  signal input is masked during a break.
- The on-chip debugging unit uses the exception vector address  $60_{\text{H}}$  for software breakpoint (DBTRAP, refer to “*Interrupt Controller (INTC)*” on page 99). Thus the debugger takes over control when one of the following exceptions occur:
  - debug trap (DBTRAP)
  - illegal op-code detection (ILGOP)
  - ROM Correction (if supported by the microcontroller)

The debugger executes its own exception handler. Therefore, the user's exception handler at address  $60_{\text{H}}$  will not be executed.

# Chapter 21 Auxiliary Frequency Output Function (AFO)

This microcontroller has an Auxiliary Frequency Output Function (AFO).

## 21.1 Features

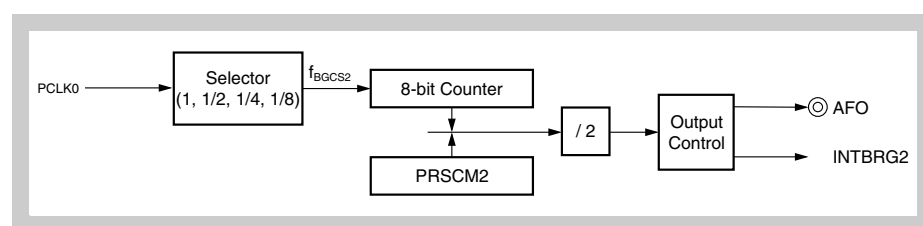
- Frequency up to 8 Mbps
- Programmable frequency output
- Interval timer function
- Interrupt request signal (INTBRG2)

## 21.2 Configuration

The AFO function includes the following hardware.

**Table 21-1 AFO Configuration**

Item	Configuration
Control registers	Prescaler mode register 2 (PRSM2)
	Prescaler compare register 2 (PRSCM2)



**Figure 21-1 Block diagram of Auxiliary Frequency Output Function**

**Note** PCLK0: 32 MHz peripheral clock (refer to “Clock Generator” on page 148)

## 21.3 Control Registers

### (1) PRSM2 - Prescaler mode register 2

The PRSM2 register controls generation of a baud rate signal for the AFO function.

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

**Address** FFFF FDE0<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is initialized by  $\overline{\text{RESET}}$ .

Reset input clears this register to 00<sub>H</sub>.

7	6	5	4	3	2	1	0
0	0	0	BGCE2	0	TODIS2	BGCS21	BGCS20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 21-2 PRSM2 register contents**

Bit position	Bit name	Function																									
4, 2	BGCE2, TODIS2	Sets the baud rate generator output																									
		<table><tr><th>BGCE</th><th>TODIS2</th><th>AFO</th><th>INTBRG2</th><th>Counter</th></tr><tr><td>0</td><td>0</td><td>fixed on 0</td><td>fixed on 0</td><td>fixed on 01<sub>H</sub></td></tr><tr><td>0</td><td>1</td><td>fixed on 0</td><td>fixed on 0</td><td>fixed on 01<sub>H</sub></td></tr><tr><td>1</td><td>0</td><td>active</td><td>active</td><td>active</td></tr><tr><td>1</td><td>1</td><td>fixed on 0</td><td>active</td><td>active</td></tr></table>	BGCE	TODIS2	AFO	INTBRG2	Counter	0	0	fixed on 0	fixed on 0	fixed on 01 <sub>H</sub>	0	1	fixed on 0	fixed on 0	fixed on 01 <sub>H</sub>	1	0	active	active	active	1	1	fixed on 0	active	active
		BGCE	TODIS2	AFO	INTBRG2	Counter																					
		0	0	fixed on 0	fixed on 0	fixed on 01 <sub>H</sub>																					
		0	1	fixed on 0	fixed on 0	fixed on 01 <sub>H</sub>																					
		1	0	active	active	active																					
1	1	fixed on 0	active	active																							
1 to 0	BGCS2[1:0]	Sets the baud rate generator clock																									
		<table><tr><th>BGCS21</th><th>BGCS20</th><th>Baud Rate Generator Clock Selection (f<sub>BGCS2</sub>)</th><th>Setting Value (k)</th></tr><tr><td>0</td><td>0</td><td>f<sub>PCLK0</sub> / 1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>f<sub>PCLK0</sub> / 2</td><td>1</td></tr><tr><td>1</td><td>0</td><td>f<sub>PCLK0</sub> / 4</td><td>2</td></tr><tr><td>1</td><td>1</td><td>f<sub>PCLK0</sub> / 8</td><td>3</td></tr></table>	BGCS21	BGCS20	Baud Rate Generator Clock Selection (f <sub>BGCS2</sub> )	Setting Value (k)	0	0	f <sub>PCLK0</sub> / 1	0	0	1	f <sub>PCLK0</sub> / 2	1	1	0	f <sub>PCLK0</sub> / 4	2	1	1	f <sub>PCLK0</sub> / 8	3					
		BGCS21	BGCS20	Baud Rate Generator Clock Selection (f <sub>BGCS2</sub> )	Setting Value (k)																						
		0	0	f <sub>PCLK0</sub> / 1	0																						
		0	1	f <sub>PCLK0</sub> / 2	1																						
		1	0	f <sub>PCLK0</sub> / 4	2																						
1	1	f <sub>PCLK0</sub> / 8	3																								

- Caution**
1. Do not rewrite the PRSM2 register during operation.
  2. Set bits BGCS21, BGCS20 and TODIS2 before setting the BGCE2 bit to 1.

**(2) PRSCM2 - Prescaler compare registers 2**

The PRSCM2 register is an 8-bit compare register. Together with the clock selected in the PRSM2 register, this register sets the clock of the AFO.

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

**Address** FFFF FDE1<sub>H</sub>

**Initial Value** 00<sub>H</sub>. This register is initialized by  $\overline{\text{RESET}}$ .

7	6	5	4	3	2	1	0
PRSCM27	PRSCM26	PRSCM25	PRSCM24	PRSCM23	PRSCM22	PRSCM21	PRSCM20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 21-3 PRSM2 register contents**

Bit position	Bit name	Function																																																																															
7 to 0	PRSCM2 [7:0]	Sets the AFO clock																																																																															
		<table><tr><th>PRSCM27</th><th>PRSCM26</th><th>...</th><th>PRSCM22</th><th>PRSCM21</th><th>PRSCM20</th><th>AFO clock</th><th>N</th></tr><tr><td>0</td><td>0</td><td>...</td><td>0</td><td>0</td><td>0</td><td>f<sub>BGCS2</sub>/256</td><td>256</td></tr><tr><td>0</td><td>0</td><td>...</td><td>0</td><td>0</td><td>1</td><td>f<sub>BGCS2</sub></td><td>1</td></tr><tr><td>0</td><td>0</td><td>...</td><td>0</td><td>1</td><td>0</td><td>f<sub>BGCS2</sub>/2</td><td>2</td></tr><tr><td>0</td><td>0</td><td>...</td><td>0</td><td>1</td><td>1</td><td>f<sub>BGCS2</sub>/3</td><td>3</td></tr><tr><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td></td></tr><tr><td>1</td><td>1</td><td>...</td><td>1</td><td>0</td><td>0</td><td>f<sub>BGCS2</sub>/252</td><td>252</td></tr><tr><td>1</td><td>1</td><td>...</td><td>1</td><td>0</td><td>1</td><td>f<sub>BGCS2</sub>/253</td><td>253</td></tr><tr><td>1</td><td>1</td><td>...</td><td>1</td><td>1</td><td>0</td><td>f<sub>BGCS2</sub>/254</td><td>254</td></tr><tr><td>1</td><td>1</td><td>...</td><td>1</td><td>1</td><td>1</td><td>f<sub>BGCS2</sub>/255</td><td>255</td></tr></table>	PRSCM27	PRSCM26	...	PRSCM22	PRSCM21	PRSCM20	AFO clock	N	0	0	...	0	0	0	f <sub>BGCS2</sub> /256	256	0	0	...	0	0	1	f <sub>BGCS2</sub>	1	0	0	...	0	1	0	f <sub>BGCS2</sub> /2	2	0	0	...	0	1	1	f <sub>BGCS2</sub> /3	3	...	...	...	...	...	...		1	1	...	1	0	0	f <sub>BGCS2</sub> /252	252	1	1	...	1	0	1	f <sub>BGCS2</sub> /253	253	1	1	...	1	1	0	f <sub>BGCS2</sub> /254	254	1	1	...	1	1	1	f <sub>BGCS2</sub> /255	255
		PRSCM27	PRSCM26	...	PRSCM22	PRSCM21	PRSCM20	AFO clock	N																																																																								
		0	0	...	0	0	0	f <sub>BGCS2</sub> /256	256																																																																								
		0	0	...	0	0	1	f <sub>BGCS2</sub>	1																																																																								
		0	0	...	0	1	0	f <sub>BGCS2</sub> /2	2																																																																								
		0	0	...	0	1	1	f <sub>BGCS2</sub> /3	3																																																																								
		...	...	...	...	...	...																																																																										
		1	1	...	1	0	0	f <sub>BGCS2</sub> /252	252																																																																								
		1	1	...	1	0	1	f <sub>BGCS2</sub> /253	253																																																																								
		1	1	...	1	1	0	f <sub>BGCS2</sub> /254	254																																																																								
1	1	...	1	1	1	f <sub>BGCS2</sub> /255	255																																																																										

- Caution**
1. Do not rewrite the PRSCM2 register during operation.
  2. Set the PRSCM2 register before setting the PRSM2.BGCE2 bit to 1.
  3. Do not set the AFO clock to a frequency higher than 8 MHz.

**Note**  $f_{\text{BGCS2}}$ : Clock frequency selected by bits PRSM2.BGCS2[1:0].



## 21.4 Operation

### 21.4.1 Auxiliary frequency generation

The auxiliary frequency output clock is generated by dividing the main clock. The baud rate generated from the main clock is obtained by the following equation.

$$f_{AFO} = \frac{f_{BGCS2}}{N \times 2} = \frac{f_{XX}}{2^{(k+2)} \times N \times 2}$$

$f_{AFO}$	AFO clock
$f_{BGCS2}$	Clock frequency selected by bits PRSM2.BGCS21 and PRSM2.BGCS20
$f_{XX}$	Main clock oscillation frequency
k	PRSM2 register setting value ( $0 \leq k \leq 3$ )
N	<ul style="list-style-type: none"> <li>if PRSCM2 = 01<sub>H</sub> to FF<sub>H</sub>: N = PRSCM2 register value (1 to 255)</li> <li>if PRSCM2 = 00<sub>H</sub>: N = 256</li> </ul>

### 21.4.2 Interval timer function

The AFO function can be used as interval timer regardless whether the auxiliary frequency output is used or not. For this purpose an interrupt request signal (INTBRG2) is assigned, which can be handled like any maskable interrupt.

# Chapter 22 Reset Function

## 22.1 Features

- Reset function by  $\overline{\text{RESET}}$  input  
Analog filter eliminates noise from the  $\overline{\text{RESET}}$  pin
- Forced reset function by DCU (refer to “On-Chip Debug Unit” on page 1034)
- Clock Monitor reset CLMRES
- Reset function by DDI = L while  $\overline{\text{DRST}} = \text{L}$  (N-Wire interface)

## 22.2 Configuration

During a system reset, most pins enter the high-impedance state.

Therefore, if an external device always requires a defined input level (e.g. external memory) a pull-up (or pull-down) resistor must be connected to each concerned output pin to prevent signal lines from floating. If no resistor is connected, the external device may be destroyed when these pins enter the high-impedance state.

For a list of all pin states after reset, see “Pin Functions during and after Reset” on page 67.

### (1) Hardware status

With reset the hardware is initialized. When the reset status is released, program execution is started.

The following table describes the status of the clocks during reset and after reset release. Note that the clock status "operates" does not inevitably mean that any function using this clock source operates as well. The function may additionally require to be enabled by other means.

**Table 22-1 Hardware status during and after reset**

Item	During reset	After reset
Main oscillator	Operation continues	Operation continues
Internal oscillator	Operation stopped	Starts oscillation
PLL clocks	Operation stopped	Starts operation
CPU system clock	Operation stopped	Starts operation after PLL stabilization time
CPU	Operation stopped	Initialized and program execution starts after PLL stabilization time
Peripheral clocks	Operation stopped	Start after PLL stabilization time
On-chip peripheral functions	Operation stopped	Initialized and can be operated after PLL stabilization time
I/O pins (port/alternative function pins)	See chapter “Pin Functions” on page 23 for a description.	

**(2) Register status**

With each reset the registers of the CPU, internal RAM, and on-chip peripherals are initialized. After a reset, make sure to set the registers to the values needed within your program.

**Table 22-2 Initial values of CPU and internal RAM after reset**

On-chip hardware		Register name	Initial value after Reset
CPU	Program registers	General-purpose register (r0)	0000 0000 <sub>H</sub>
		General-purpose registers (r1 to r31)	Undefined
	System registers	Program counter (PC)	0000 0000 <sub>H</sub>
		Program status word (PSW)	0000 0020 <sub>H</sub>
		Status saving registers during maskable interrupt (EIPC, EIPSW)	Undefined
		Status saving registers during non-maskable interrupt (NMI) (FEPC, FEPSW)	Undefined
		Status saving registers during CALLT execution (CTPC, CTPSW)	Undefined
		Status saving registers during exception/debug trap (DBPC, DBPSW)	Undefined
		Interrupt source register (ECR)	0000 0000 <sub>H</sub>
		CALLT base pointer (CTBP)	Undefined
		Floating-point arithmetic control register (ECT)	0000 0000 <sub>H</sub>
		Floating-point arithmetic status register (EFG)	0000 0000 <sub>H</sub>
Internal RAM		Undefined	
Peripherals	Macro internal registers	The reset values of the various registers are given in the chapters of the peripheral functions	

## 22.3 Operation

When a low-level signal is input to the  $\overline{\text{RESET}}$  pin, a system reset is effected and each on-chip hardware is initialized.

When the oscillation stabilization time has elapsed, the external  $\overline{\text{RESET}}$  can be released. The internal PLL stabilization time counter (PSTC) is started after some delay, induced by the analog noise rejection filter. The PSTC counts  $2^{14}$  oscillator clocks ( $f_x$ ) and determines the PLL lock state. The internal system reset is released, the CPU is supplied with the PLL output clock and the user's software is started up.

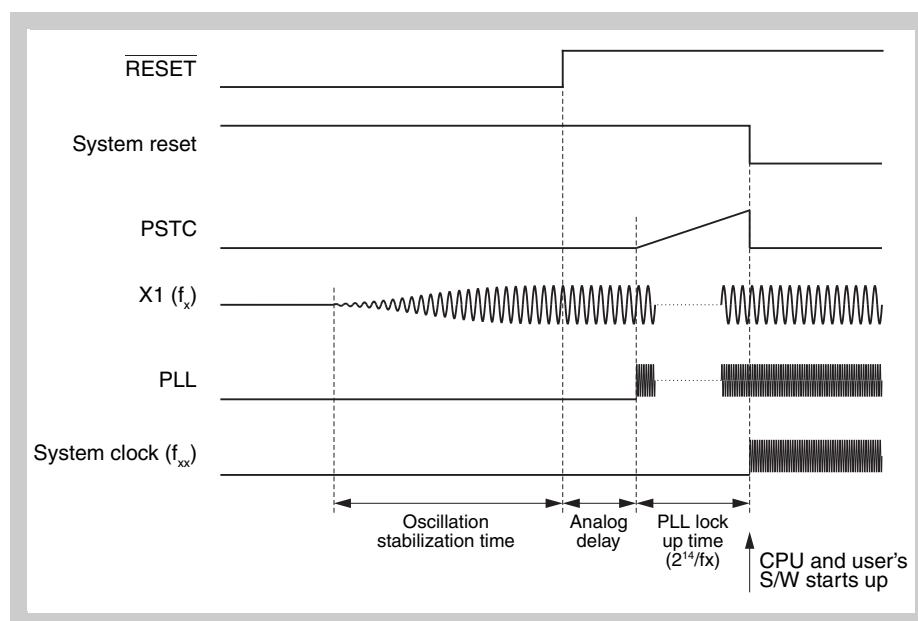


Figure 22-1 Reset timing

- Note**
1. If no clock is supplied (i.e. the oscillator does not work) the internal system reset will not be released independently from input level of the external  $\overline{\text{RESET}}$  pin.
  2. During  $\overline{\text{RESET}}$  the PLL is stopped. Therefore the PLL lock-up time is always necessary after  $\overline{\text{RESET}}$  release.
  3. The on-chip debug function can force the activation of the system reset independently from input level of the  $\overline{\text{RESET}}$  pin.

# Appendix A Special Function Registers

The following tables list all registers that are accessed via the NPB (Peripheral bus). The registers are called “special function registers” (SFR).

Table A-3 lists all CAN special function registers.

## A.1 CAN Registers

The CAN registers are accessible via the programmable peripheral area. The addresses are given as offsets to the programmable peripheral base address (refer to “CAN module register and message buffer addresses” on page 393.)

**Table A-3 CAN special function registers (1/3)**

Address offset	Register name	Shortcut	1	8	16	32	Initial value
0x000	CAN0 Global Macro Control register	C0GMCTRL	-	-	R/W	-	0x0000
0x000	CAN0 Global Macro Control register low byte	C0GMCTRLLL	R/W	R/W	-	-	0x00
0x001	CAN0 Global Macro Control register high byte	C0GMCTRLH	R/W	R/W	-	-	0x00
0x002	CAN0 Global Macro Clock Selection register	C0GMCS	R/W	R/W	-	-	0x0F
0x006	CAN0 Global Macro Automatic Block Transmission register	C0GMABT	-	-	R/W	-	0x0000
0x006	CAN0 Global Macro Automatic Block Transmission register low byte	C0GMABTLL	R/W	R/W	-	-	0x00
0x007	CAN0 Global Macro Automatic Block Transmission register high byte	C0GMABTH	R/W	R/W	-	-	0x00
0x008	CAN0 Global Macro Automatic Block Transmission Delay register	C0GMABTD	R/W	R/W	-	-	0x00
0x040	CAN0 Module Mask 1 register lower half word	C0MASK1L	-	-	R/W	-	undefined
0x042	CAN0 Module Mask 1 register upper half word	C0MASK1H	-	-	R/W	-	undefined
0x044	CAN0 Module Mask 2 register lower half word	C0MASK2L	-	-	R/W	-	undefined
0x046	CAN0 Module Mask 2 register upper half word	C0MASK2H	-	-	R/W	-	undefined
0x048	CAN0 Module Mask 3 register lower half word	C0MASK3L	-	-	R/W	-	undefined
0x04A	CAN0 Module Mask 3 register upper half word	C0MASK3H	-	-	R/W	-	undefined
0x04C	CAN0 Module Mask 4 register lower half word	C0MASK4L	-	-	R/W	-	undefined
0x04E	CAN0 Module Mask 4 register upper half word	C0MASK4H	-	-	R/W	-	undefined
0x050	CAN0 Module Control register	C0CTRL	-	-	R/W	-	0x0000
0x052	CAN0 Module Last Error Code register	C0LEC	R/W	R/W	-	-	0x00
0x053	CAN0 Module Information register	C0INFO	R	R	-	-	0x00
0x054	CAN0 Module Error Counter	C0ERC	-	-	R/W	-	0x0000
0x056	CAN0 Module Interrupt Enable register	C0IE	-	-	R/W	-	0x0000
0x056	CAN0 Module Interrupt Enable register low byte	C0IEL	R/W	R/W	-	-	0x00
0x057	CAN0 Module Interrupt Enable register high byte	C0IEH	R/W	R/W	-	-	0x00

**Table A-3 CAN special function registers (2/3)**

Address offset	Register name	Shortcut	1	8	16	32	Initial value
0x058	CAN0 Module Interrupt Status register	C0INTS	-	-	R/W	-	0x0000
0x058	CAN0 Module Interrupt Status register low byte	C0INTSL	R/W	R/W	-	-	0x00
0x05A	CAN0 Module Bit-Rate Prescaler register	C0BRP	R/W	R/W	-	-	0xFF
0x05C	CAN0 Bit Rate register	C0BTR	-	-	R/W	-	0x370F
0x05E	CAN0 Module Last In-Pointer register	C0LIPT	-	R/W	-	-	undefined
0x060	CAN0 Module Receive History List Get Pointer register	C0RGPT	-	-	R/W	-	0x??02 (undefined)
0x060	CAN0 Module Receive History List Get Pointer register low byte	C0RGPTL	R/W	R/W	-	-	0x02
0x062	CAN0 Module Last Out-Pointer register	C0LOPT	-	R	-	-	undefined
0x064	CAN0 Module Transmit History List Get Pointer register	C0TGPT	-	-	R/W	-	0x??02 (undefined)
0x064	CAN0 Module Transmit History List Get Pointer register low byte	C0TGPTL	R/W	R/W	-	-	0x02
0x066	CAN0 Module Time Stamp register	C0TS	-	-	R/W	-	0x0000
0x066	CAN0 Module Time Stamp register low byte	C0TSL	R/W	R/W	-	-	0x00
0x067	CAN0 Module Time Stamp register high byte	C0TSH	R/W	R/W	-	-	0x00
0x100 to 0x4EF	CAN0 Message Buffer registers, see <i>Table 14-20 on page 396</i> .						
0x600	CAN1 Global Macro Control register	C1GMCTRL	-	-	R/W	-	0x0000
0x600	CAN1 Global Macro Control register low byte	C1GMCTRLLL	R/W	R/W	-	-	0x00
0x601	CAN1 Global Macro Control register high byte	C1GMCTRLH	R/W	R/W	-	-	0x00
0x602	CAN1 Global Macro Clock Selection register	C1GMCS	R/W	R/W	-	-	0x0F
0x606	CAN1 Global Macro Automatic Block Transmission register	C1GMABT	-	-	R/W	-	0x0000
0x606	CAN1 Global Macro Automatic Block Transmission register low byte	C1GMABTL	R/W	R/W	-	-	0x00
0x607	CAN1 Global Macro Automatic Block Transmission register high byte	C1GMABTH	R/W	R/W	-	-	0x00
0x608	CAN1 Global Macro Automatic Block Transmission Delay register	C1GMABTD	R/W	R/W	-	-	0x00
0x640	CAN1 Module Mask 1 register lower half word	C1MASK1L	-	-	R/W	-	undefined
0x642	CAN1 Module Mask 1 register upper half word	C1MASK1H	-	-	R/W	-	undefined
0x644	CAN1 Module Mask 2 register lower half word	C1MASK2L	-	-	R/W	-	undefined
0x646	CAN1 Module Mask 2 register upper half word	C1MASK2H	-	-	R/W	-	undefined
0x648	CAN1 Module Mask 3 register lower half word	C1MASK3L	-	-	R/W	-	undefined
0x64A	CAN1 Module Mask 3 register upper half word	C1MASK3H	-	-	R/W	-	undefined
0x64C	CAN1 Module Mask 4 register lower half word	C1MASK4L	-	-	R/W	-	undefined
0x64E	CAN1 Module Mask 4 register upper half word	C1MASK4H	-	-	R/W	-	undefined
0x650	CAN1 Module Control register	C1CTRL	-	-	R/W	-	0x0000
0x652	CAN1 Module Last Error Code register	C1LEC	R/W	R/W	-	-	0x00
0x653	CAN1 Module Information register	C1INFO	R	R	-	-	0x00
0x654	CAN1 Module Error Counter	C1ERC	-	-	R/W	-	0x0000
0x656	CAN1 Module Interrupt Enable register	C1IE	-	-	R/W	-	0x0000

**Table A-3 CAN special function registers (3/3)**

Address offset	Register name	Shortcut	1	8	16	32	Initial value
0x656	CAN1 Module Interrupt Enable register low byte	C1IEL	R/W	R/W	-	-	0x00
0x657	CAN1 Module Interrupt Enable register high byte	C1IEH	R/W	R/W	-	-	0x00
0x658	CAN1 Module Interrupt Status register	C1INTS	-	-	R/W	-	0x0000
0x658	CAN1 Module Interrupt Status register low byte	C1INTSL	R/W	R/W	-	-	0x00
0x65A	CAN1 Module Bit-Rate Prescaler register	C1BRP	R/W	R/W	-	-	0xFF
0x65C	CAN1 Bit Rate register	C1BTR	-	-	R/W	-	0x370F
0x65E	CAN1 Module Last In-Pointer register	C1LIPT	-	R/W	-	-	undefined
0x660	CAN1 Module Receive History List Get Pointer register	C1RGPT	-	-	R/W	-	0x??02 (undefined)
0x660	CAN1 Module Receive History List Get Pointer register low byte	C1RGPTL	R/W	R/W	-	-	0x02
0x662	CAN1 Module Last Out-Pointer register	C1LOPT	-	R	-	-	undefined
0x664	CAN1 Module Transmit History List Get Pointer register	C1TGPT	-	-	R/W	-	0x??02 (undefined)
0x664	CAN1 Module Transmit History List Get Pointer register low byte	C1TGPTL	R/W	R/W	-	-	0x02
0x666	CAN1 Module Time Stamp register	C1TS	-	-	R/W	-	0x0000
0x666	CAN1 Module Time Stamp register low byte	C1TSL	R/W	R/W	-	-	0x00
0x667	CAN1 Module Time Stamp register high byte	C1TSH	R/W	R/W	-	-	0x00
0x700 to 0xAEF	CAN1 Message Buffer registers, see <i>Table 14-20 on page 396</i> .						

## A.2 Other Special Function Registers

**Table A-4 Other special function registers (1/15)**

Address	Register name	Shortcut	1	8	16	32	Initial value
0xFFFFF064	Peripheral area selection control register	BPC	-	-	R/W	-	0x0000
0xFFFFF06E	Internal peripheral function wait control register	VSWC	R/W	R/W	-	-	0x77
0xFFFFF100	Interrupt mask control register 0	IMR0	-	-	R/W	-	0xFFFF
0xFFFFF100	Interrupt mask control register 0L	IMR0L	R/W	R/W	-	-	0xFF
0xFFFFF101	Interrupt mask control register 0H	IMR0H	R/W	R/W	-	-	0xFF
0xFFFFF102	Interrupt mask control register 1	IMR1	-	-	R/W	-	0xFFFF
0xFFFFF102	Interrupt mask control register 1L	IMR1L	R/W	R/W	-	-	0xFF
0xFFFFF103	Interrupt mask control register 1H	IMR1H	R/W	R/W	-	-	0xFF
0xFFFFF104	Interrupt mask control register 2	IMR2	-	-	R/W	-	0xFFFF
0xFFFFF104	Interrupt mask control register 2L	IMR2L	R/W	R/W	-	-	0xFF
0xFFFFF105	Interrupt mask control register 2H	IMR2H	R/W	R/W	-	-	0xFF
0xFFFFF106	Interrupt mask control register 3	IMR3	-	-	R/W	-	0xFFFF
0xFFFFF106	Interrupt mask control register 3L	IMR3L	R/W	R/W	-	-	0xFF
0xFFFFF107	Interrupt mask control register 3H	IMR3H	R/W	R/W	-	-	0xFF
0xFFFFF108	Interrupt mask control register 4	IMR4	-	-	R/W	-	0xFFFF
0xFFFFF108	Interrupt mask control register 4L	IMR4L	R/W	R/W	-	-	0xFF
0xFFFFF109	Interrupt mask control register 4H	IMR4H	R/W	R/W	-	-	0xFF
0xFFFFF10A	Interrupt mask control register 5	IMR5	-	-	R/W	-	0xFFFF
0xFFFFF10A	Interrupt mask control register 5L	IMR5L	R/W	R/W	-	-	0xFF
0xFFFFF10B	Interrupt mask control register 5H	IMR5H	R/W	R/W	-	-	0xFF
0xFFFFF10C	Interrupt mask control register 6	IMR6	-	-	R/W	-	0xFFFF
0xFFFFF10C	Interrupt mask control register 6L	IMR6L	R/W	R/W	-	-	0xFF
0xFFFFF10D	Interrupt mask control register 6H	IMR6H	R/W	R/W	-	-	0xFF
0xFFFFF10E	Interrupt mask control register 7	IMR7	-	-	R/W	-	0x001F
0xFFFFF10E	Interrupt mask control register 7L	IMR7L	R/W	R/W	-	-	0x1F
0xFFFFF10F	Interrupt mask control register 7H	IMR7H	R/W	R/W	-	-	0x00
0xFFFFF110	Interrupt control register	ERRIC	R/W	R/W	-	-	0x47
0xFFFFF112	Interrupt control register	OSDIC	R/W	R/W	-	-	0x47
0xFFFFF114	Interrupt control register	PIC0	R/W	R/W	-	-	0x47
0xFFFFF116	Interrupt control register	PIC1	R/W	R/W	-	-	0x47
0xFFFFF118	Interrupt control register	PIC2	R/W	R/W	-	-	0x47
0xFFFFF11A	Interrupt control register	PIC3	R/W	R/W	-	-	0x47
0xFFFFF11C	Interrupt control register	PIC4	R/W	R/W	-	-	0x47
0xFFFFF11E	Interrupt control register	PIC5	R/W	R/W	-	-	0x47
0xFFFFF120	Interrupt control register	PIC6	R/W	R/W	-	-	0x47
0xFFFFF122	Interrupt control register	PIC7	R/W	R/W	-	-	0x47
0xFFFFF124	Interrupt control register	PIC8	R/W	R/W	-	-	0x47
0xFFFFF126	Interrupt control register	PIC9	R/W	R/W	-	-	0x47



**Table A-4 Other special function registers (2/15)**

Address	Register name	Shortcut	1	8	16	32	Initial value
0xFFFF128	Interrupt control register	PIC10	R/W	R/W	-	-	0x47
0xFFFF12A	Interrupt control register	PIC11	R/W	R/W	-	-	0x47
0xFFFF12C	Interrupt control register	PIC12	R/W	R/W	-	-	0x47
0xFFFF12E	Interrupt control register	TS0OVIC	R/W	R/W	-	-	0x47
0xFFFF130	Interrupt control register	TS0CCIC0	R/W	R/W	-	-	0x47
0xFFFF132	Interrupt control register	TS0CCIC1	R/W	R/W	-	-	0x47
0xFFFF134	Interrupt control register	TS0CCIC2	R/W	R/W	-	-	0x47
0xFFFF136	Interrupt control register	TS0CCIC3	R/W	R/W	-	-	0x47
0xFFFF138	Interrupt control register	TS0CCIC4	R/W	R/W	-	-	0x47
0xFFFF13A	Interrupt control register	TS0CCIC5	R/W	R/W	-	-	0x47
0xFFFF13C	Interrupt control register	TS0CDIC0	R/W	R/W	-	-	0x47
0xFFFF13E	Interrupt control register	TS0ODIC	R/W	R/W	-	-	0x47
0xFFFF140	Interrupt control register	TS0ERIC	R/W	R/W	-	-	0x47
0xFFFF142	Interrupt control register	TS0WNIC	R/W	R/W	-	-	0x47
0xFFFF144	Interrupt control register	TS1OVIC	R/W	R/W	-	-	0x47
0xFFFF146	Interrupt control register	TS1CCIC0	R/W	R/W	-	-	0x47
0xFFFF148	Interrupt control register	TS1CCIC1	R/W	R/W	-	-	0x47
0xFFFF14A	Interrupt control register	TS1CCIC2	R/W	R/W	-	-	0x47
0xFFFF14C	Interrupt control register	TS1CCIC3	R/W	R/W	-	-	0x47
0xFFFF14E	Interrupt control register	TS1CCIC4	R/W	R/W	-	-	0x47
0xFFFF150	Interrupt control register	TS1CCIC5	R/W	R/W	-	-	0x47
0xFFFF152	Interrupt control register	TS1CDIC0	R/W	R/W	-	-	0x47
0xFFFF154	Interrupt control register	TS1ODIC	R/W	R/W	-	-	0x47
0xFFFF156	Interrupt control register	TS1ERIC	R/W	R/W	-	-	0x47
0xFFFF158	Interrupt control register	TS1WNIC	R/W	R/W	-	-	0x47
0xFFFF15A	Interrupt control register	TT0OVIC	R/W	R/W	-	-	0x47
0xFFFF15C	Interrupt control register	TT0CCIC0	R/W	R/W	-	-	0x47
0xFFFF15E	Interrupt control register	TT0CCIC1	R/W	R/W	-	-	0x47
0xFFFF160	Interrupt control register	TT0ECIC	R/W	R/W	-	-	0x47
0xFFFF162	Interrupt control register	TT1OVIC	R/W	R/W	-	-	0x47
0xFFFF164	Interrupt control register	TT1CCIC0	R/W	R/W	-	-	0x47
0xFFFF166	Interrupt control register	TT1CCIC1	R/W	R/W	-	-	0x47
0xFFFF168	Interrupt control register	TT1ECIC	R/W	R/W	-	-	0x47
0xFFFF16A	Interrupt control register	TAA0OVIC	R/W	R/W	-	-	0x47
0xFFFF16C	Interrupt control register	TAA0CCIC0	R/W	R/W	-	-	0x47
0xFFFF16E	Interrupt control register	TAA0CCIC1	R/W	R/W	-	-	0x47
0xFFFF170	Interrupt control register	TAA1OVIC	R/W	R/W	-	-	0x47
0xFFFF172	Interrupt control register	TAA1CCIC0	R/W	R/W	-	-	0x47
0xFFFF174	Interrupt control register	TAA1CCIC1	R/W	R/W	-	-	0x47
0xFFFF176	Interrupt control register	TAA2OVIC	R/W	R/W	-	-	0x47
0xFFFF178	Interrupt control register	TAA2CCIC0	R/W	R/W	-	-	0x47

**Table A-4 Other special function registers (3/15)**

Address	Register name	Shortcut	1	8	16	32	Initial value
0xFFFF17A	Interrupt control register	TAA2CCIC1	R/W	R/W	-	-	0x47
0xFFFF17C	Interrupt control register	TAA3OVIC	R/W	R/W	-	-	0x47
0xFFFF17E	Interrupt control register	TAA3CCIC0	R/W	R/W	-	-	0x47
0xFFFF180	Interrupt control register	TAA3CCIC1	R/W	R/W	-	-	0x47
0xFFFF182	Interrupt control register	TAA4OVIC	R/W	R/W	-	-	0x47
0xFFFF184	Interrupt control register	TAA4CCIC0	R/W	R/W	-	-	0x47
0xFFFF186	Interrupt control register	TAA4CCIC1	R/W	R/W	-	-	0x47
0xFFFF188	Interrupt control register	TAA5OVIC	R/W	R/W	-	-	0x47
0xFFFF18A	Interrupt control register	TAA5CCIC0	R/W	R/W	-	-	0x47
0xFFFF18C	Interrupt control register	TAA5CCIC1	R/W	R/W	-	-	0x47
0xFFFF18E	Interrupt control register	TAA6OVIC	R/W	R/W	-	-	0x47
0xFFFF190	Interrupt control register	TAA6CCIC0	R/W	R/W	-	-	0x47
0xFFFF192	Interrupt control register	TAA6CCIC1	R/W	R/W	-	-	0x47
0xFFFF194	Interrupt control register	TAA7OVIC	R/W	R/W	-	-	0x47
0xFFFF196	Interrupt control register	TAA7CCIC0	R/W	R/W	-	-	0x47
0xFFFF198	Interrupt control register	TAA7CCIC1	R/W	R/W	-	-	0x47
0xFFFF19A	Interrupt control register	TAA8OVIC	R/W	R/W	-	-	0x47
0xFFFF19C	Interrupt control register	TAA8CCIC0	R/W	R/W	-	-	0x47
0xFFFF19E	Interrupt control register	TAA8CCIC1	R/W	R/W	-	-	0x47
0xFFFF1A0	Interrupt control register	TAA9OVIC	R/W	R/W	-	-	0x47
0xFFFF1A2	Interrupt control register	TAA9CCIC0	R/W	R/W	-	-	0x47
0xFFFF1A4	Interrupt control register	TAA9CCIC1	R/W	R/W	-	-	0x47
0xFFFF1A6	Interrupt control register	BRGIC	R/W	R/W	-	-	0x47
0xFFFF1A8	Interrupt control register	BRGIC2	R/W	R/W	-	-	0x47
0xFFFF1B6	Interrupt control register	C0ERRIC	R/W	R/W	-	-	0x47
0xFFFF1B8	Interrupt control register	C0WUPIC	R/W	R/W	-	-	0x47
0xFFFF1BA	Interrupt control register	C0RECIC	R/W	R/W	-	-	0x47
0xFFFF1BC	Interrupt control register	C0TRXIC	R/W	R/W	-	-	0x47
0xFFFF1BE	Interrupt control register	C1ERRIC	R/W	R/W	-	-	0x47
0xFFFF1C0	Interrupt control register	C1WUPIC	R/W	R/W	-	-	0x47
0xFFFF1C2	Interrupt control register	C1RECIC	R/W	R/W	-	-	0x47
0xFFFF1C4	Interrupt control register	C1TRXIC	R/W	R/W	-	-	0x47
0xFFFF1D6	Interrupt control register	CB0TIC	R/W	R/W	-	-	0x47
0xFFFF1D8	Interrupt control register	CB0RIC	R/W	R/W	-	-	0x47
0xFFFF1DA	Interrupt control register	CB0REIC	R/W	R/W	-	-	0x47
0xFFFF1DC	Interrupt control register	CB1TIC	R/W	R/W	-	-	0x47
0xFFFF1DE	Interrupt control register	CB1RIC	R/W	R/W	-	-	0x47
0xFFFF1E0	Interrupt control register	CB1REIC	R/W	R/W	-	-	0x47
0xFFFF1E2	Interrupt control register	CE0OFIC	R/W	R/W	-	-	0x47
0xFFFF1E4	Interrupt control register	CE0CIC	R/W	R/W	-	-	0x47
0xFFFF1E6	Interrupt control register	CE1OFIC	R/W	R/W	-	-	0x47

**Table A-4 Other special function registers (4/15)**

Address	Register name	Shortcut	1	8	16	32	Initial value
0xFFFFF1E8	Interrupt control register	CE1CIC	R/W	R/W	-	-	0x47
0xFFFFF1EA	Interrupt control register	UC0REIC	R/W	R/W	-	-	0x47
0xFFFFF1EC	Interrupt control register	UC0TIC	R/W	R/W	-	-	0x47
0xFFFFF1EE	Interrupt control register	UC1REIC	R/W	R/W	-	-	0x47
0xFFFFF1F0	Interrupt control register	UC1TIC	R/W	R/W	-	-	0x47
0xFFFFF1F2	Interrupt control register	ADIC0	R/W	R/W	-	-	0x47
0xFFFFF1F4	Interrupt control register	ADIC1	R/W	R/W	-	-	0x47
0xFFFFF1F6	Interrupt control register	DMAIC2	R/W	R/W	-	-	0x47
0xFFFFF1F8	Interrupt control register	DMAIC3	R/W	R/W	-	-	0x47
0xFFFFF1FA	In-service priority register	ISPR	R	R	-	-	0x00
0xFFFFF1FC	Command register	PRCMD	-	W	-	-	undefined
0xFFFFF200	A/D Converter 0 mode register 0	ADM00	R/W	R/W	-	-	0x00
0xFFFFF201	A/D Converter 0 mode register 1	ADM01	R/W	R/W	-	-	0x00
0xFFFFF202	A/D Converter 0 mode register 2	ADM02	R/W	R/W	-	-	0x00
0xFFFFF210	A/D conversion result register 00	ADCR00	-	-	R	-	undefined
0xFFFFF211	A/D conversion result register 00H	ADCR00H	-	R	-	-	undefined
0xFFFFF212	A/D conversion result register 01	ADCR01	-	-	R	-	undefined
0xFFFFF213	A/D conversion result register 01H	ADCR01H	-	R	-	-	undefined
0xFFFFF214	A/D conversion result register 02	ADCR02	-	-	R	-	undefined
0xFFFFF215	A/D conversion result register 02H	ADCR02H	-	R	-	-	undefined
0xFFFFF216	A/D conversion result register 03	ADCR03	-	-	R	-	undefined
0xFFFFF217	A/D conversion result register 03H	ADCR03H	-	R	-	-	undefined
0xFFFFF218	A/D conversion result register 04	ADCR04	-	-	R	-	undefined
0xFFFFF219	A/D conversion result register 04H	ADCR04H	-	R	-	-	undefined
0xFFFFF21A	A/D conversion result register 05	ADCR05	-	-	R	-	undefined
0xFFFFF21B	A/D conversion result register 05H	ADCR05H	-	R	-	-	undefined
0xFFFFF21C	A/D conversion result register 06	ADCR06	-	-	R	-	undefined
0xFFFFF21D	A/D conversion result register 06H	ADCR06H	-	R	-	-	undefined
0xFFFFF21E	A/D conversion result register 07	ADCR07	-	-	R	-	undefined
0xFFFFF21F	A/D conversion result register 07H	ADCR07H	-	R	-	-	undefined
0xFFFFF220	A/D conversion result register 08	ADCR08	-	-	R	-	undefined
0xFFFFF221	A/D conversion result register 08H	ADCR08H	-	R	-	-	undefined
0xFFFFF222	A/D conversion result register 09	ADCR09	-	-	R	-	undefined
0xFFFFF223	A/D conversion result register 09H	ADCR09H	-	R	-	-	undefined
0xFFFFF22E	A/D conversion result register 0 for DMA	ADDMA0	-	-	R	-	undefined
0xFFFFF240	A/D Converter 1 mode register 0	ADM10	R/W	R/W	-	-	0x00
0xFFFFF241	A/D Converter 1 mode register 1	ADM11	R/W	R/W	-	-	0x00
0xFFFFF242	A/D Converter 1 mode register 2	ADM12	R/W	R/W	-	-	0x00
0xFFFFF250	A/D conversion result register 10	ADCR10	-	-	R	-	undefined
0xFFFFF251	A/D conversion result register 10H	ADCR10H	-	R	-	-	undefined
0xFFFFF252	A/D conversion result register 11	ADCR11	-	-	R	-	undefined

**Table A-4 Other special function registers (5/15)**

Address	Register name	Shortcut	1	8	16	32	Initial value
0xFFFFF253	A/D conversion result register 11H	ADCR11H	-	R	-	-	undefined
0xFFFFF254	A/D conversion result register 12	ADCR12	-	-	R	-	undefined
0xFFFFF255	A/D conversion result register 12H	ADCR12H	-	R	-	-	undefined
0xFFFFF256	A/D conversion result register 13	ADCR13	-	-	R	-	undefined
0xFFFFF257	A/D conversion result register 13H	ADCR13H	-	R	-	-	undefined
0xFFFFF258	A/D conversion result register 14	ADCR14	-	-	R	-	undefined
0xFFFFF259	A/D conversion result register 14H	ADCR14H	-	R	-	-	undefined
0xFFFFF25A	A/D conversion result register 15	ADCR15	-	-	R	-	undefined
0xFFFFF25B	A/D conversion result register 15H	ADCR15H	-	R	-	-	undefined
0xFFFFF25C	A/D conversion result register 16	ADCR16	-	-	R	-	undefined
0xFFFFF25D	A/D conversion result register 16H	ADCR16H	-	R	-	-	undefined
0xFFFFF25E	A/D conversion result register 17	ADCR17	-	-	R	-	undefined
0xFFFFF25F	A/D conversion result register 17H	ADCR17H	-	R	-	-	undefined
0xFFFFF260	A/D conversion result register 18	ADCR18	-	-	R	-	undefined
0xFFFFF261	A/D conversion result register 18H	ADCR18H	-	R	-	-	undefined
0xFFFFF262	A/D conversion result register 19	ADCR19	-	-	R	-	undefined
0xFFFFF263	A/D conversion result register 19H	ADCR19H	-	R	-	-	undefined
0xFFFFF26E	A/D conversion result register 1 for DMA	ADDMA1	-	-	R	-	undefined
0xFFFFF270	A/D Converter 0 trigger source select register	ADTRSEL0	R/W	R/W	-	-	0x00
0xFFFFF272	A/D Converter 1 trigger source select register	ADTRSEL1	R/W	R/W	-	-	0x00
0xFFFFF300	DMA transfer memory start address register 0	MAR0	-	-	R/W	-	0x0000
0xFFFFF302	DMA transfer memory start address register 1	MAR1	-	-	R/W	-	0x0000
0xFFFFF304	DMA transfer memory start address register 2	MAR2	-	-	R/W	-	0x0000
0xFFFFF306	DMA transfer memory start address register 3	MAR3	-	-	R/W	-	0x0000
0xFFFFF308	DMA transfer memory start address register 4	MAR4	-	-	R/W	-	0x0000
0xFFFFF30A	DMA transfer memory start address register 5	MAR5	-	-	R/W	-	0x0000
0xFFFFF30C	DMA transfer memory start address register 6	MAR6	-	-	R/W	-	0x0000
0xFFFFF30E	DMA transfer memory start address register 7	MAR7	-	-	R/W	-	0x0000
0xFFFFF324	DMA transfer SFR start address register 2	SAR2	-	R/W	-	-	0x00
0xFFFFF326	DMA transfer SFR start address register 3	SAR3	-	R/W	-	-	0x00
0xFFFFF340	DMA transfer count register 0	DTCR0	-	R/W	-	-	0x00
0xFFFFF342	DMA transfer count register 1	DTCR1	-	R/W	-	-	0x00
0xFFFFF344	DMA transfer count register 2	DTCR2	-	R/W	-	-	0x00
0xFFFFF346	DMA transfer count register 3	DTCR3	-	R/W	-	-	0x00
0xFFFFF348	DMA transfer count register 4	DTCR4	-	R/W	-	-	0x00
0xFFFFF34A	DMA transfer count register 5	DTCR5	-	R/W	-	-	0x00
0xFFFFF34C	DMA transfer count register 6	DTCR6	-	R/W	-	-	0x00
0xFFFFF34E	DMA transfer count register 7	DTCR7	-	R/W	-	-	0x00
0xFFFFF360	DMA mode control register	DMAMCL	R/W	R/W	-	-	0x00
0xFFFFF362	DMA status register	DMASL	R/W	R/W	-	-	0x00
0xFFFFF364	DMA data size control register	DMADSCL	R/W	R/W	-	-	0x00

**Table A-4 Other special function registers (6/15)**

Address	Register name	Shortcut	1	8	16	32	Initial value
0xFFFFF388	DMA trigger factor register 4	DTFR4	-	R/W	-	-	0x00
0xFFFFF38A	DMA trigger factor register 5	DTFR5	-	R/W	-	-	0x00
0xFFFFF38C	DMA trigger factor register 6	DTFR6	-	R/W	-	-	0x00
0xFFFFF38E	DMA trigger factor register 7	DTFR7	-	R/W	-	-	0x00
0xFFFFF400	Port 0	P0	R	R	-	-	undefined
0xFFFFF402	Port 1	P1	R/W	R/W	-	-	undefined
0xFFFFF404	Port 2	P2	R/W	R/W	-	-	undefined
0xFFFFF406	Port 3	P3	R/W	R/W	-	-	undefined
0xFFFFF408	Port 4	P4	R/W	R/W	-	-	undefined
0xFFFFF40A	Port 5	P5	R/W	R/W	-	-	undefined
0xFFFFF40C	Port 6	P6	R/W	R/W	-	-	undefined
0xFFFFF40E	Port 7	P7	R/W	R/W	-	-	undefined
0xFFFFF410	Port 8	P8	R/W	R/W	-	-	undefined
0xFFFFF412	Port 9	P9	R/W	R/W	-	-	undefined
0xFFFFF414	Port 10	P10	R/W	R/W	-	-	undefined
0xFFFFF416	Port 11	P11	R/W	R/W	-	-	undefined
0xFFFFF422	Port 1 mode	PM1	R/W	R/W	-	-	0xFF
0xFFFFF424	Port 2 mode	PM2	R/W	R/W	-	-	0xFF
0xFFFFF426	Port 3 mode	PM3	R/W	R/W	-	-	0xFF
0xFFFFF428	Port 4 mode	PM4	R/W	R/W	-	-	0xFF
0xFFFFF42A	Port 5 mode	PM5	R/W	R/W	-	-	0xFF
0xFFFFF42C	Port 6 mode	PM6	R/W	R/W	-	-	0xFF
0xFFFFF42E	Port 7 mode	PM7	R/W	R/W	-	-	0xFF
0xFFFFF430	Port 8 mode	PM8	R/W	R/W	-	-	0xFF
0xFFFFF432	Port 9 mode	PM9	R/W	R/W	-	-	0xFF
0xFFFFF434	Port 10 mode	PM10	R/W	R/W	-	-	0xFF
0xFFFFF436	Port 11 mode	PM11	R/W	R/W	-	-	0xFF
0xFFFFF442	Port 1 mode control	PMC1	R/W	R/W	-	-	0x00
0xFFFFF444	Port 2 mode control	PMC2	R/W	R/W	-	-	0x00
0xFFFFF446	Port 3 mode control	PMC3	R/W	R/W	-	-	0x00
0xFFFFF448	Port 4 mode control	PMC4	R/W	R/W	-	-	0x00
0xFFFFF44A	Port 5 mode control	PMC5	R/W	R/W	-	-	0x00
0xFFFFF44C	Port 6 mode control	PMC6	R/W	R/W	-	-	0x00
0xFFFFF44E	Port 7 mode control	PMC7	R/W	R/W	-	-	0x00
0xFFFFF450	Port 8 mode control	PMC8	R/W	R/W	-	-	0x00
0xFFFFF452	Port 9 mode control	PMC9	R/W	R/W	-	-	0x00
0xFFFFF454	Port 10 mode control	PMC10	R/W	R/W	-	-	0x00
0xFFFFF456	Port 11 mode control	PMC11	R/W	R/W	-	-	0x00
0xFFFFF462	Port 1 function control register	PFC1	R/W	R/W	-	-	0x00
0xFFFFF464	Port 2 function control register	PFC2	R/W	R/W	-	-	0x00
0xFFFFF46E	Port 7 function control register	PFC7	R/W	R/W	-	-	0x00

**Table A-4 Other special function registers (7/15)**

Address	Register name	Shortcut	1	8	16	32	Initial value
0xFFFFF470	Port 8 function control register	PFC8	R/W	R/W	-	-	0x00
0xFFFFF472	Port 9 function control register	PFC9	R/W	R/W	-	-	0x00
0xFFFFF474	Port 10 function control register	PFC10	R/W	R/W	-	-	0x00
0xFFFFF480	Bus cycle configuration register 0	BCT0	-	-	R/W	-	0x4444
0xFFFFF482	Bus cycle configuration register 1	BCT1	-	-	R/W	-	0x4444
0xFFFFF484	Data wait control register 0	DWC0	-	-	R/W	-	0x7777
0xFFFFF486	Data wait control register 1	DWC1	-	-	R/W	-	0x7777
0xFFFFF488	Address wait control register	AWC	-	-	R/W	-	0xFFFF
0xFFFFF48A	Bus cycle control register	BCC	-	-	R/W	-	0xAAAA
0xFFFFF48E	Bus clock dividing frequency control register	DVC	-	R/W	-	-	0x00
0xFFFFF580	TMS0 control register 0	TS0CTL0	R/W	R/W	-	-	0x00
0xFFFFF581	TMS0 control register 1	TS0CTL1	R/W	R/W	-	-	0x00
0xFFFFF582	TMS0 I/O control register 0	TS0IOC0	-	-	R/W	-	0x0000
0xFFFFF582	TMS0 I/O control register 0L	TS0IOC0L	R/W	R/W	-	-	0x00
0xFFFFF583	TMS0 I/O control register 0H	TS0IOC0H	R/W	R/W	-	-	0x00
0xFFFFF584	TMS0 I/O control register 0	TS0IOC2	R/W	R/W	-	-	0x00
0xFFFFF585	TMS0 I/O control register 0	TS0IOC4	R/W	R/W	-	-	0x00
0xFFFFF586	TMS0 option register 0	TS0OPT0	R/W	R/W	-	-	0x00
0xFFFFF587	TMS0 option register 7	TS0OPT7	R/W	R/W	-	-	0x00
0xFFFFF588	TMS0 option register 4	TS0OPT4	R/W	R/W	-	-	0x00
0xFFFFF589	TMS0 option register 5	TS0OPT5	R/W	R/W	-	-	0x00
0xFFFFF58A	TMS0 option register 1	TS0OPT1	-	-	R/W	-	0x0000
0xFFFFF58A	TMS0 option register 1L	TS0OPT1L	R/W	R/W	-	-	0x00
0xFFFFF58B	TMS0 option register 1H	TS0OPT1H	R/W	R/W	-	-	0x00
0xFFFFF58C	TMS0 dead time setting register 1	TS0DTC1	-	-	R/W	-	0x0000
0xFFFFF58E	TMS0 dead time setting register 0	TS0DTC0	-	-	R/W	-	0x0000
0xFFFFF590	TMS0 compare register 5	TS0CCR5	-	-	R/W	-	0x0000
0xFFFFF592	TMS0 compare register 4	TS0CCR4	-	-	R/W	-	0x0000
0xFFFFF594	TMS0 pattern register 1	TS0PAT1	-	-	R/W	-	0x0000
0xFFFFF596	TMS0 pattern register 0	TS0PAT0	-	-	R/W	-	0x0000
0xFFFFF598	TMS0 compare register 0	TS0CCR0	-	-	R/W	-	0x0000
0xFFFFF59A	TMS0 compare register 3	TS0CCR3	-	-	R/W	-	0x0000
0xFFFFF59C	TMS0 compare register 2	TS0CCR2	-	-	R/W	-	0x0000
0xFFFFF59E	TMS0 compare register 1	TS0CCR1	-	-	R/W	-	0x0000
0xFFFFF5A0	TMS0 option register 2	TS0OPT2	-	-	R/W	-	0x0000
0xFFFFF5A0	TMS0 option register 2L	TS0OPT2L	R/W	R/W	-	-	0x00
0xFFFFF5A1	TMS0 option register 2H	TS0OPT2H	R/W	R/W	-	-	0x00
0xFFFFF5A2	TMS0 option register 3	TS0OPT3	-	-	R/W	-	0x0000
0xFFFFF5A2	TMS0 option register 3L	TS0OPT3L	R/W	R/W	-	-	0x00
0xFFFFF5A3	TMS0 option register 3H	TS0OPT3H	R/W	R/W	-	-	0x00
0xFFFFF5A4	TMS0 option register 6	TS0OPT6	-	-	R/W	-	0x0000



**Table A-4 Other special function registers (8/15)**

Address	Register name	Shortcut	1	8	16	32	Initial value
0xFFFFF5A4	TMS0 option register 6L	TS0OPT6L	R/W	R/W	-	-	0x00
0xFFFFF5A5	TMS0 option register 6H	TS0OPT6H	R/W	R/W	-	-	0x00
0xFFFFF5A6	TMS0 counter read register	TS0CNT	-	-	R	-	0x0000
0xFFFFF5A8	TMS0 sub-counter read register	TS0SBC	-	-	R	-	0x0000
0xFFFFF5C0	TMS1 control register 0	TS1CTL0	R/W	R/W	-	-	0x00
0xFFFFF5C1	TMS1 control register 1	TS1CTL1	R/W	R/W	-	-	0x00
0xFFFFF5C2	TMS1 I/O control register 0	TS1IOC0	-	-	R/W	-	0x0000
0xFFFFF5C2	TMS1 I/O control register 0L	TS1IOC0L	R/W	R/W	-	-	0x00
0xFFFFF5C3	TMS1 I/O control register 0H	TS1IOC0H	R/W	R/W	-	-	0x00
0xFFFFF5C4	TMS1 I/O control register 0	TS1IOC2	R/W	R/W	-	-	0x00
0xFFFFF5C5	TMS1 I/O control register 0	TS1IOC4	R/W	R/W	-	-	0x00
0xFFFFF5C6	TMS1 option register 0	TS1OPT0	R/W	R/W	-	-	0x00
0xFFFFF5C7	TMS1 option register 7	TS1OPT7	R/W	R/W	-	-	0x00
0xFFFFF5C8	TMS1 option register 4	TS1OPT4	R/W	R/W	-	-	0x00
0xFFFFF5C9	TMS1 option register 5	TS1OPT5	R/W	R/W	-	-	0x00
0xFFFFF5CA	TMS1 option register 1	TS1OPT1	-	-	R/W	-	0x0000
0xFFFFF5CA	TMS1 option register 1L	TS1OPT1L	R/W	R/W	-	-	0x00
0xFFFFF5CB	TMS1 option register 1H	TS1OPT1H	R/W	R/W	-	-	0x00
0xFFFFF5CC	TMS1 dead time setting register 1	TS1DTC1	-	-	R/W	-	0x0000
0xFFFFF5CE	TMS1 dead time setting register 0	TS1DTC0	-	-	R/W	-	0x0000
0xFFFFF5D0	TMS1 compare register 5	TS1CCR5	-	-	R/W	-	0x0000
0xFFFFF5D2	TMS1 compare register 4	TS1CCR4	-	-	R/W	-	0x0000
0xFFFFF5D4	TMS1 pattern register 1	TS1PAT1	-	-	R/W	-	0x0000
0xFFFFF5D6	TMS1 pattern register 0	TS1PAT0	-	-	R/W	-	0x0000
0xFFFFF5D8	TMS1 compare register 0	TS1CCR0	-	-	R/W	-	0x0000
0xFFFFF5DA	TMS1 compare register 3	TS1CCR3	-	-	R/W	-	0x0000
0xFFFFF5DC	TMS1 compare register 2	TS1CCR2	-	-	R/W	-	0x0000
0xFFFFF5DE	TMS1 compare register 1	TS1CCR1	-	-	R/W	-	0x0000
0xFFFFF5E0	TMS1 option register 2	TS1OPT2	-	-	R/W	-	0x0000
0xFFFFF5E0	TMS1 option register 2L	TS1OPT2L	R/W	R/W	-	-	0x00
0xFFFFF5E1	TMS1 option register 2H	TS1OPT2H	R/W	R/W	-	-	0x00
0xFFFFF5E2	TMS1 option register 3	TS1OPT3	-	-	R/W	-	0x0000
0xFFFFF5E2	TMS1 option register 3L	TS1OPT3L	R/W	R/W	-	-	0x00
0xFFFFF5E3	TMS1 option register 3H	TS1OPT3H	R/W	R/W	-	-	0x00
0xFFFFF5E4	TMS1 option register 6	TS1OPT6	-	-	R/W	-	0x0000
0xFFFFF5E4	TMS1 option register 6L	TS1OPT6L	R/W	R/W	-	-	0x00
0xFFFFF5E5	TMS1 option register 6H	TS1OPT6H	R/W	R/W	-	-	0x00
0xFFFFF5E6	TMS1 counter read register	TS1CNT	-	-	R	-	0x0000
0xFFFFF5E8	TMS1 sub-counter read register	TS1SBC	-	-	R	-	0x0000
0xFFFFF600	TAA0 control register 0	TAA0CTL0	R/W	R/W	-	-	0x00
0xFFFFF601	TAA0 control register 1	TAA0CTL1	R/W	R/W	-	-	0x00

**Table A-4 Other special function registers (9/15)**

Address	Register name	Shortcut	1	8	16	32	Initial value
0xFFFFF602	TAA0 I/O control register 0	TAA0IOC0	R/W	R/W	-	-	0x00
0xFFFFF603	TAA0 I/O control register 1	TAA0IOC1	R/W	R/W	-	-	0x00
0xFFFFF604	TAA0 I/O control register 2	TAA0IOC2	R/W	R/W	-	-	0x00
0xFFFFF605	TAA0 option register 0	TAA0OPT0	R/W	R/W	-	-	0x00
0xFFFFF606	TAA0 capture/compare register 0	TAA0CCR0	-	-	R/W	-	0x0000
0xFFFFF608	TAA0 capture/compare register 1	TAA0CCR1	-	-	R/W	-	0x0000
0xFFFFF60A	TAA0 counter read buffer register	TAA0CNT	-	-	R	-	0x0000
0xFFFFF60C	TAA0 I/O control register 4	TAA0IOC4	R/W	R/W	-	-	0x00
0xFFFFF60E	TAA0 specific overflow value control register	TAA0SVC	R/W	R/W	-	-	0x00
0xFFFFF610	TAA1 control register 0	TAA1CTL0	R/W	R/W	-	-	0x00
0xFFFFF611	TAA1 control register 1	TAA1CTL1	R/W	R/W	-	-	0x00
0xFFFFF612	TAA1 I/O control register 0	TAA1IOC0	R/W	R/W	-	-	0x00
0xFFFFF613	TAA1 I/O control register 1	TAA1IOC1	R/W	R/W	-	-	0x00
0xFFFFF614	TAA1 I/O control register 2	TAA1IOC2	R/W	R/W	-	-	0x00
0xFFFFF615	TAA1 option register 0	TAA1OPT0	R/W	R/W	-	-	0x00
0xFFFFF616	TAA1 capture/compare register 0	TAA1CCR0	-	-	R/W	-	0x0000
0xFFFFF618	TAA1 capture/compare register 1	TAA1CCR1	-	-	R/W	-	0x0000
0xFFFFF61A	TAA1 counter read buffer register	TAA1CNT	-	-	R	-	0x0000
0xFFFFF61C	TAA1 I/O control register 4	TAA1IOC4	R/W	R/W	-	-	0x00
0xFFFFF61D	TAA1 option register 1	TAA1OPT1	R/W	R/W	-	-	0x00
0xFFFFF61E	TAA1 specific overflow value control register	TAA1SVC	R/W	R/W	-	-	0x00
0xFFFFF620	TAA2 control register 0	TAA2CTL0	R/W	R/W	-	-	0x00
0xFFFFF621	TAA2 control register 1	TAA2CTL1	R/W	R/W	-	-	0x00
0xFFFFF622	TAA2 I/O control register 0	TAA2IOC0	R/W	R/W	-	-	0x00
0xFFFFF623	TAA2 I/O control register 1	TAA2IOC1	R/W	R/W	-	-	0x00
0xFFFFF624	TAA2 I/O control register 2	TAA2IOC2	R/W	R/W	-	-	0x00
0xFFFFF625	TAA2 option register 0	TAA2OPT0	R/W	R/W	-	-	0x00
0xFFFFF626	TAA2 capture/compare register 0	TAA2CCR0	-	-	R/W	-	0x0000
0xFFFFF628	TAA2 capture/compare register 1	TAA2CCR1	-	-	R/W	-	0x0000
0xFFFFF62A	TAA2 counter read buffer register	TAA2CNT	-	-	R	-	0x0000
0xFFFFF62C	TAA2 I/O control register 4	TAA2IOC4	R/W	R/W	-	-	0x00
0xFFFFF62D	TAA2 option register 1	TAA2OPT1	R/W	R/W	-	-	0x00
0xFFFFF62E	TAA2 specific overflow value control register	TAA2SVC	R/W	R/W	-	-	0x00
0xFFFFF630	TAA3 control register 0	TAA3CTL0	R/W	R/W	-	-	0x00
0xFFFFF631	TAA3 control register 1	TAA3CTL1	R/W	R/W	-	-	0x00
0xFFFFF632	TAA3 I/O control register 0	TAA3IOC0	R/W	R/W	-	-	0x00
0xFFFFF633	TAA3 I/O control register 1	TAA3IOC1	R/W	R/W	-	-	0x00
0xFFFFF634	TAA3 I/O control register 2	TAA3IOC2	R/W	R/W	-	-	0x00
0xFFFFF635	TAA3 option register 0	TAA3OPT0	R/W	R/W	-	-	0x00
0xFFFFF636	TAA3 capture/compare register 0	TAA3CCR0	-	-	R/W	-	0x0000
0xFFFFF638	TAA3 capture/compare register 1	TAA3CCR1	-	-	R/W	-	0x0000



**Table A-4 Other special function registers (10/15)**

Address	Register name	Shortcut	1	8	16	32	Initial value
0xFFFF63A	TAA3 counter read buffer register	TAA3CNT	-	-	R	-	0x0000
0xFFFF63C	TAA3 I/O control register 4	TAA3IOC4	R/W	R/W	-	-	0x00
0xFFFF63D	TAA3 option register 1	TAA3OPT1	R/W	R/W	-	-	0x00
0xFFFF63E	TAA3 specific overflow value control register	TAA3SVC	R/W	R/W	-	-	0x00
0xFFFF640	TAA4 control register 0	TAA4CTL0	R/W	R/W	-	-	0x00
0xFFFF641	TAA4 control register 1	TAA4CTL1	R/W	R/W	-	-	0x00
0xFFFF642	TAA4 I/O control register 0	TAA4IOC0	R/W	R/W	-	-	0x00
0xFFFF643	TAA4 I/O control register 1	TAA4IOC1	R/W	R/W	-	-	0x00
0xFFFF644	TAA4 I/O control register 2	TAA4IOC2	R/W	R/W	-	-	0x00
0xFFFF645	TAA4 option register 0	TAA4OPT0	R/W	R/W	-	-	0x00
0xFFFF646	TAA4 capture/compare register 0	TAA4CCR0	-	-	R/W	-	0x0000
0xFFFF648	TAA4 capture/compare register 1	TAA4CCR1	-	-	R/W	-	0x0000
0xFFFF64A	TAA4 counter read buffer register	TAA4CNT	-	-	R	-	0x0000
0xFFFF64C	TAA4 I/O control register 4	TAA4IOC4	R/W	R/W	-	-	0x00
0xFFFF64E	TAA4 specific overflow value control register	TAA4SVC	R/W	R/W	-	-	0x00
0xFFFF650	TAA5 control register 0	TAA5CTL0	R/W	R/W	-	-	0x00
0xFFFF651	TAA5 control register 1	TAA5CTL1	R/W	R/W	-	-	0x00
0xFFFF652	TAA5 I/O control register 0	TAA5IOC0	R/W	R/W	-	-	0x00
0xFFFF653	TAA5 I/O control register 1	TAA5IOC1	R/W	R/W	-	-	0x00
0xFFFF654	TAA5 I/O control register 2	TAA5IOC2	R/W	R/W	-	-	0x00
0xFFFF655	TAA5 option register 0	TAA5OPT0	R/W	R/W	-	-	0x00
0xFFFF656	TAA5 capture/compare register 0	TAA5CCR0	-	-	R/W	-	0x0000
0xFFFF658	TAA5 capture/compare register 1	TAA5CCR1	-	-	R/W	-	0x0000
0xFFFF65A	TAA5 counter read buffer register	TAA5CNT	-	-	R	-	0x0000
0xFFFF65C	TAA5 I/O control register 4	TAA5IOC4	R/W	R/W	-	-	0x00
0xFFFF65D	TAA5 option register 1	TAA5OPT1	R/W	R/W	-	-	0x00
0xFFFF65E	TAA5 specific overflow value control register	TAA5SVC	R/W	R/W	-	-	0x00
0xFFFF660	TAA6 control register 0	TAA6CTL0	R/W	R/W	-	-	0x00
0xFFFF661	TAA6 control register 1	TAA6CTL1	R/W	R/W	-	-	0x00
0xFFFF662	TAA6 I/O control register 0	TAA6IOC0	R/W	R/W	-	-	0x00
0xFFFF663	TAA6 I/O control register 1	TAA6IOC1	R/W	R/W	-	-	0x00
0xFFFF664	TAA6 I/O control register 2	TAA6IOC2	R/W	R/W	-	-	0x00
0xFFFF665	TAA6 option register 0	TAA6OPT0	R/W	R/W	-	-	0x00
0xFFFF666	TAA6 capture/compare register 0	TAA6CCR0	-	-	R/W	-	0x0000
0xFFFF668	TAA6 capture/compare register 1	TAA6CCR1	-	-	R/W	-	0x0000
0xFFFF66A	TAA6 counter read buffer register	TAA6CNT	-	-	R	-	0x0000
0xFFFF66C	TAA6 I/O control register 4	TAA6IOC4	R/W	R/W	-	-	0x00
0xFFFF66D	TAA6 option register 1	TAA6OPT1	R/W	R/W	-	-	0x00
0xFFFF66E	TAA6 specific overflow value control register	TAA6SVC	R/W	R/W	-	-	0x00
0xFFFF670	TAA7 control register 0	TAA7CTL0	R/W	R/W	-	-	0x00
0xFFFF671	TAA7 control register 1	TAA7CTL1	R/W	R/W	-	-	0x00

**Table A-4 Other special function registers (11/15)**

Address	Register name	Shortcut	1	8	16	32	Initial value
0xFFFFF672	TAA7 I/O control register 0	TAA7IOC0	R/W	R/W	-	-	0x00
0xFFFFF673	TAA7 I/O control register 1	TAA7IOC1	R/W	R/W	-	-	0x00
0xFFFFF674	TAA7 I/O control register 2	TAA7IOC2	R/W	R/W	-	-	0x00
0xFFFFF675	TAA7 option register 0	TAA7OPT0	R/W	R/W	-	-	0x00
0xFFFFF676	TAA7 capture/compare register 0	TAA7CCR0	-	-	R/W	-	0x0000
0xFFFFF678	TAA7 capture/compare register 1	TAA7CCR1	-	-	R/W	-	0x0000
0xFFFFF67A	TAA7 counter read buffer register	TAA7CNT	-	-	R	-	0x0000
0xFFFFF67C	TAA7 I/O control register 4	TAA7IOC4	R/W	R/W	-	-	0x00
0xFFFFF67D	TAA7 option register 1	TAA7OPT1	R/W	R/W	-	-	0x00
0xFFFFF67E	TAA7 specific overflow value control register	TAA7SVC	R/W	R/W	-	-	0x00
0xFFFFF680	TAA8 control register 0	TAA8CTL0	R/W	R/W	-	-	0x00
0xFFFFF681	TAA8 control register 1	TAA8CTL1	R/W	R/W	-	-	0x00
0xFFFFF682	TAA8 I/O control register 0	TAA8IOC0	R/W	R/W	-	-	0x00
0xFFFFF683	TAA8 I/O control register 1	TAA8IOC1	R/W	R/W	-	-	0x00
0xFFFFF684	TAA8 I/O control register 2	TAA8IOC2	R/W	R/W	-	-	0x00
0xFFFFF685	TAA8 option register 0	TAA8OPT0	R/W	R/W	-	-	0x00
0xFFFFF686	TAA8 capture/compare register 0	TAA8CCR0	-	-	R/W	-	0x0000
0xFFFFF688	TAA8 capture/compare register 1	TAA8CCR1	-	-	R/W	-	0x0000
0xFFFFF68A	TAA8 counter read buffer register	TAA8CNT	-	-	R	-	0x0000
0xFFFFF68C	TAA8 I/O control register 4	TAA8IOC4	R/W	R/W	-	-	0x00
0xFFFFF68E	TAA8 specific overflow value control register	TAA8SVC	R/W	R/W	-	-	0x00
0xFFFFF690	TMT0 control register 0	TT0CTL0	R/W	R/W	-	-	0x00
0xFFFFF691	TMT0 control register 1	TT0CTL1	R/W	R/W	-	-	0x00
0xFFFFF692	TMT0 control register 2	TT0CTL2	R/W	R/W	-	-	0x00
0xFFFFF693	TMT0 I/O control register 0	TT0IOC0	R/W	R/W	-	-	0x00
0xFFFFF694	TMT0 I/O control register 1	TT0IOC1	R/W	R/W	-	-	0x00
0xFFFFF695	TMT0 I/O control register 2	TT0IOC2	R/W	R/W	-	-	0x00
0xFFFFF696	TMT0 I/O control register 3	TT0IOC3	R/W	R/W	-	-	0x00
0xFFFFF697	TMT0 option register 0	TT0OPT0	R/W	R/W	-	-	0x00
0xFFFFF698	TMT0 option register 1	TT0OPT1	R/W	R/W	-	-	0x00
0xFFFFF699	TMT0 option register 2	TT0OPT2	R/W	R/W	-	-	0x00
0xFFFFF69A	TMT0 capture/compare register 0	TT0CCR0	-	-	R/W	-	0x0000
0xFFFFF69C	TMT0 capture/compare register 1	TT0CCR1	-	-	R/W	-	0x0000
0xFFFFF69E	TMT0 counter read buffer register	TT0CNT	-	-	R	-	0x0000
0xFFFFF6A0	TMT1 control register 0	TT1CTL0	R/W	R/W	-	-	0x00
0xFFFFF6A1	TMT1 control register 1	TT1CTL1	R/W	R/W	-	-	0x00
0xFFFFF6A2	TMT1 control register 2	TT1CTL2	R/W	R/W	-	-	0x00
0xFFFFF6A3	TMT1 I/O control register 0	TT1IOC0	R/W	R/W	-	-	0x00
0xFFFFF6A4	TMT1 I/O control register 1	TT1IOC1	R/W	R/W	-	-	0x00
0xFFFFF6A5	TMT1 I/O control register 2	TT1IOC2	R/W	R/W	-	-	0x00
0xFFFFF6A6	TMT1 I/O control register 3	TT1IOC3	R/W	R/W	-	-	0x00

**Table A-4 Other special function registers (12/15)**

Address	Register name	Shortcut	1	8	16	32	Initial value
0xFFFF6A7	TMT1 option register 0	TT1OPT0	R/W	R/W	-	-	0x00
0xFFFF6A8	TMT1 option register 1	TT1OPT1	R/W	R/W	-	-	0x00
0xFFFF6A9	TMT1 option register 2	TT1OPT2	R/W	R/W	-	-	0x00
0xFFFF6AA	TMT1 capture/compare register 0	TT1CCR0	-	-	R/W	-	0x0000
0xFFFF6AC	TMT1 capture/compare register 1	TT1CCR1	-	-	R/W	-	0x0000
0xFFFF6AE	TMT1 counter read buffer register	TT1CNT	-	-	R	-	0x0000
0xFFFF6B0	TAA9 control register 0	TAA9CTL0	R/W	R/W	-	-	0x00
0xFFFF6B1	TAA9 control register 1	TAA9CTL1	R/W	R/W	-	-	0x00
0xFFFF6B2	TAA9 I/O control register 0	TAA9IOC0	R/W	R/W	-	-	0x00
0xFFFF6B3	TAA9 I/O control register 1	TAA9IOC1	R/W	R/W	-	-	0x00
0xFFFF6B4	TAA9 I/O control register 2	TAA9IOC2	R/W	R/W	-	-	0x00
0xFFFF6B5	TAA9 option register 0	TAA9OPT0	R/W	R/W	-	-	0x00
0xFFFF6B6	TAA9 capture/compare register 0	TAA9CCR0	-	-	R/W	-	0x0000
0xFFFF6B8	TAA9 capture/compare register 1	TAA9CCR1	-	-	R/W	-	0x0000
0xFFFF6BA	TAA9 counter read buffer register	TAA9CNT	-	-	R	-	0x0000
0xFFFF6BC	TAA9 I/O control register 3	TAA9IOC4	R/W	R/W	-	-	0x00
0xFFFF6BD	TAA9 option register 1	TAA9OPT1	R/W	R/W	-	-	0x00
0xFFFF6BE	TAA9 specific overflow value control register	TAA9SVC	R/W	R/W	-	-	0x00
0xFFFF6D0	Interrupt share enable register	INTSEL	R/W	R/W	-	-	0x00
0xFFFF6D2	Interrupt source flag register	INTERRF	R/W	R/W	-	-	0x00
0xFFFF6F0	Timer input control register 0	TAAIC0	R/W	R/W	-	-	0x00
0xFFFF6F1	Timer input control register 1	TAAIC1	R/W	R/W	-	-	0x00
0xFFFF6F2	Timer input control register 2	TAAIC2	R/W	R/W	-	-	0x00
0xFFFF6F3	Timer input control register 3	TAAIC3	R/W	R/W	-	-	0x00
0xFFFF7A0	Noise rejection time control register	NRC	R/W	R/W	-	-	0x00
0xFFFF802	Peripheral status register	PHS	R/W	R/W	-	-	0x00
0xFFFF860	CAN clock selection register	CANCKSEL	R/W	R/W	-	-	0x00
0xFFFF880	External interrupt edge specification register	INTM0	R/W	R/W	-	-	0x00
0xFFFF882	External interrupt edge specification register	INTM1	R/W	R/W	-	-	0x00
0xFFFF884	External interrupt edge specification register	INTM2	R/W	R/W	-	-	0x00
0xFFFF886	External interrupt edge specification register	INTM3	R/W	R/W	-	-	0x00
0xFFFF888	Emergency shut-off control register 5	PESC5	R/W	R/W	-	-	0x00
0xFFFF88A	Emergency shut-off status register 5	ESOST5	R/W	R/W	-	-	0x00
0xFFFF88C	Emergency shut-off control register 6	PESC6	R/W	R/W	-	-	0x00
0xFFFF88E	Emergency shut-off status register 6	ESOST6	R/W	R/W	-	-	0x00
0xFFFF890	Emergency shut-off mask register 5	PESMK5	R/W	R/W	-	-	0x00
0xFFFF892	Emergency shut-off mask register 6	PESMK6	R/W	R/W	-	-	0x00
0xFFFF8A0	Flash ROM ECC error register	ROMEAD	-	-	-	R	0x00000000
0xFFFF8B0	RAM ECC error register	RAMEAD	-	-	R	-	0x0000
0xFFFF8B0	RAM ECC error address register	RAMEAD	-	-	R	-	0x0000
0xFFFF8B2	RAM ECC data location register	RAMEDLR	-	R	-	-	0x00

**Table A-4 Other special function registers (13/15)**

Address	Register name	Shortcut	1	8	16	32	Initial value
0xFFFFF8B4	RAM ECC control register	RAMECC	R/W	R/W	-	-	0x00
0xFFFFF990	TMT0 counter write buffer register	TT0TCW	-	-	R/W	-	0x0000
0xFFFFF9A0	TMT1 counter write buffer register	TT1TCW	-	-	R/W	-	0x0000
0xFFFFFA00	UARTC0 control register 0	UC0CTL0	R/W	R/W	-	-	0x10
0xFFFFFA01	UARTC0 control register 1	UC0CTL1	-	R/W	-	-	0x00
0xFFFFFA02	UARTC0 control register 2	UC0CTL2	-	R/W	-	-	0xFF
0xFFFFFA03	UARTC0 option control register 0	UC0OPT0	R/W	R/W	-	-	0x14
0xFFFFFA04	UARTC0 status register	UC0STR	R/W	R/W	-	-	0x00
0xFFFFFA06	UARTC0 receive data register	UC0RX	-	-	R	-	0x01FF
0xFFFFFA06	UARTC0 receive data register L	UC0RXL	-	R	-	-	0xFF
0xFFFFFA08	UARTC0 transmit data register	UC0TX	-	-	R/W	-	0x01FF
0xFFFFFA08	UARTC0 transmit data register L	UC0TXL	-	R/W	-	-	0xFF
0xFFFFFA0A	UARTC0 option control register 1	UC0OPT1	R/W	R/W	-	-	0x00
0xFFFFFA0B	UARTC0 option control register 2	UC0OPT2	R	R	-	-	0x00
0xFFFFFA20	UARTC1 control register 0	UC1CTL0	R/W	R/W	-	-	0x10
0xFFFFFA21	UARTC1 control register 1	UC1CTL1	-	R/W	-	-	0x00
0xFFFFFA22	UARTC1 control register 2	UC1CTL2	-	R/W	-	-	0xFF
0xFFFFFA23	UARTC1 option control register 0	UC1OPT0	R/W	R/W	-	-	0x14
0xFFFFFA24	UARTC1 status register	UC1STR	R/W	R/W	-	-	0x00
0xFFFFFA26	UARTC1 receive data register	UC1RX	-	-	R	-	0x01FF
0xFFFFFA26	UARTC1 receive data register L	UC1RXL	-	R	-	-	0xFF
0xFFFFFA28	UARTC1 transmit data register	UC1TX	-	-	R/W	-	0x01FF
0xFFFFFA28	UARTC1 transmit data register L	UC1TXL	-	R/W	-	-	0xFF
0xFFFFFA2A	UARTC1 option control register 1	UC1OPT1	R/W	R/W	-	-	0x00
0xFFFFFA2B	UARTC1 option control register 2	UC1OPT2	R	R	-	-	0x00
0xFFFFFB40	TMS0 option register 1A	TS0OPT1A	-	-	R/W	-	0x0000
0xFFFFFB40	TMS0 option register 1AL	TS0OPT1AL	R/W	R/W	-	-	0x00
0xFFFFFB41	TMS0 option register 1AH	TS0OPT1AH	R/W	R/W	-	-	0x00
0xFFFFFB42	TMS0 dead time setting register 1A	TS0DTC1A	-	-	R/W	-	0x0000
0xFFFFFB44	TMS0 dead time setting register 0A	TS0DTC0A	-	-	R/W	-	0x0000
0xFFFFFB46	TMS0 compare register 5A	TS0CCR5A	-	-	R/W	-	0x0000
0xFFFFFB48	TMS0 compare register 4A	TS0CCR4A	-	-	R/W	-	0x0000
0xFFFFFB4A	TMS0 compare register 3B	TS0CCR3B	-	-	R/W	-	0x0000
0xFFFFFB4C	TMS0 compare register 2B	TS0CCR2B	-	-	R/W	-	0x0000
0xFFFFFB4E	TMS0 compare register 1B	TS0CCR1B	-	-	R/W	-	0x0000
0xFFFFFB50	TMS0 compare register 0A	TS0CCR0A	-	-	R/W	-	0x0000
0xFFFFFB52	TMS0 compare register 3A	TS0CCR3A	-	-	R/W	-	0x0000
0xFFFFFB54	TMS0 compare register 2A	TS0CCR2A	-	-	R/W	-	0x0000
0xFFFFFB56	TMS0 compare register 1A	TS0CCR1A	-	-	R/W	-	0x0000
0xFFFFFBC0	TMS1 option register 1A	TS1OPT1A	-	-	R/W	-	0x0000
0xFFFFFBC0	TMS1 option register 1AL	TS1OPT1AL	R/W	R/W	-	-	0x00

**Table A-4 Other special function registers (14/15)**

Address	Register name	Shortcut	1	8	16	32	Initial value
0xFFFFFBC1	TMS1 option register 1AH	TS1OPT1AH	R/W	R/W	-	-	0x00
0xFFFFFBC2	TMS1 dead time setting register 1A	TS1DTC1A	-	-	R/W	-	0x0000
0xFFFFFBC4	TMS1 dead time setting register 0A	TS1DTC0A	-	-	R/W	-	0x0000
0xFFFFFBC6	TMS1 compare register 5A	TS1CCR5A	-	-	R/W	-	0x0000
0xFFFFFBC8	TMS1 compare register 4A	TS1CCR4A	-	-	R/W	-	0x0000
0xFFFFFBCA	TMS1 compare register 3B	TS1CCR3B	-	-	R/W	-	0x0000
0xFFFFFBCC	TMS1 compare register 2B	TS1CCR2B	-	-	R/W	-	0x0000
0xFFFFFBCE	TMS1 compare register 1B	TS1CCR1B	-	-	R/W	-	0x0000
0xFFFFFBD0	TMS1 compare register 0A	TS1CCR0A	-	-	R/W	-	0x0000
0xFFFFFBD2	TMS1 compare register 3A	TS1CCR3A	-	-	R/W	-	0x0000
0xFFFFFBD4	TMS1 compare register 2A	TS1CCR2A	-	-	R/W	-	0x0000
0xFFFFFBD6	TMS1 compare register 1A	TS1CCR1A	-	-	R/W	-	0x0000
0xFFFFFC32	Port 9 function control expansion register	PFCE9	R/W	R/W	-	-	0x00
0xFFFFFCA0	Clock monitor mode register	CLM	R/W	R/W	-	-	0x00
0xFFFFFCF8	Data flash control register	DFLCTL	R/W	R/W	-	-	0x00
0xFFFFFD00	CSIB0 control register 0	CB0CTL0	R/W	R/W	-	-	0x01
0xFFFFFD01	CSIB0 control register 1	CB0CTL1	R/W	R/W	-	-	0x00
0xFFFFFD02	CSIB0 control register 2	CB0CTL2	-	R/W	-	-	0x00
0xFFFFFD03	CSIB0 status register	CB0STR	R/W	R/W	-	-	0x00
0xFFFFFD04	CSIB0 receive data register	CB0RX0	-	-	R	-	0x0000
0xFFFFFD04	CSIB0 receive data register L	CB0RX0L	-	R	-	-	0x00
0xFFFFFD06	CSIB0 transmit data register	CB0TX0	-	-	R/W	-	0x0000
0xFFFFFD06	CSIB0 transmit data register L	CB0TX0L	-	R/W	-	-	0x00
0xFFFFFD20	CSIB1 control register 0	CB1CTL0	R/W	R/W	-	-	0x01
0xFFFFFD21	CSIB1 control register 1	CB1CTL1	R/W	R/W	-	-	0x00
0xFFFFFD22	CSIB1 control register 2	CB1CTL2	-	R/W	-	-	0x00
0xFFFFFD23	CSIB1 status register	CB1STR	R/W	R/W	-	-	0x00
0xFFFFFD24	CSIB1 receive data register	CB1RX0	-	-	R	-	0x0000
0xFFFFFD24	CSIB1 receive data register L	CB1RX0L	-	R	-	-	0x00
0xFFFFFD26	CSIB1 transmit data register	CB1TX0	-	-	R/W	-	0x0000
0xFFFFFD26	CSIB1 transmit data register L	CB1TX0L	-	R/W	-	-	0x00
0xFFFFFD40	Queued CSI0 control register 0	CE0CTL0	R/W	R/W	-	-	0x00
0xFFFFFD41	Queued CSI0 control register 1	CE0CTL1	R/W	R/W	-	-	0x07
0xFFFFFD42	CSIE0 receive data register	CE0RX0	-	-	R	-	0x0000
0xFFFFFD42	CSIE0 receive data register L	CE0RX0L	-	R	-	-	0x00
0xFFFFFD43	CSIE0 receive data register H	CE0RXH0	-	R	-	-	0x00
0xFFFFFD44	CSIE0 chip select data buffer register	CE0CS	-	R/W	R/W	-	0xFFFF
0xFFFFFD46	CSIE0 transmission data buffer register	CE0TX0	-	-	R/W	-	0x0000
0xFFFFFD46	CSIE0 transmission data buffer register L	CE0TX0L	-	R/W	-	-	0x00
0xFFFFFD47	CSIE0 transmission data buffer register H	CE0TXH0	-	R/W	-	-	0x00
0xFFFFFD48	Queued CSI0 status register	CE0STR	R/W	R/W	-	-	0x20

**Table A-4 Other special function registers (15/15)**

Address	Register name	Shortcut	1	8	16	32	Initial value
0xFFFFFD49	Queued CSI0 control register 2	CE0CTL2	R/W	R/W	-	-	0x00
0xFFFFFD4C	Queued CSI0 control register 3	CE0CTL3	R/W	R/W	-	-	0x00
0xFFFFFD4D	Queued CSI0 control register 4	CE0CTL4	R/W	R/W	-	-	0x00
0xFFFFFD50	Queued CSI0 enhanced timing register 0	CE0OPT0	-	-	R/W	-	0x0002
0xFFFFFD52	Queued CSI0 enhanced timing register 1	CE0OPT1	-	-	R/W	-	0x0002
0xFFFFFD54	Queued CSI0 enhanced timing register 2	CE0OPT2	-	-	R/W	-	0x0002
0xFFFFFD56	Queued CSI0 enhanced timing register 3	CE0OPT3	-	-	R/W	-	0x0002
0xFFFFFD58	Queued CSI0 enhanced timing register 4	CE0OPT4	-	-	R/W	-	0x0002
0xFFFFFD5A	Queued CSI0 enhanced timing register 5	CE0OPT5	-	-	R/W	-	0x0002
0xFFFFFD5C	Queued CSI0 enhanced timing register 6	CE0OPT6	-	-	R/W	-	0x0002
0xFFFFFD5E	Queued CSI0 enhanced timing register 7	CE0OPT7	-	-	R/W	-	0x0002
0xFFFFFD80	Queued CSI1 control register 0	CE1CTL0	R/W	R/W	-	-	0x00
0xFFFFFD81	Queued CSI1 control register 1	CE1CTL1	R/W	R/W	-	-	0x07
0xFFFFFD82	CSIE1 receive data register	CE1RX0	-	-	R	-	0x0000
0xFFFFFD82	CSIE1 receive data register L	CE1RXL0	-	R	-	-	0x00
0xFFFFFD83	CSIE1 receive data register H	CE1RXH0	-	R	-	-	0x00
0xFFFFFD84	CSIE1 chip select data buffer register	CE1CS	-	R/W	R/W	-	0xFFFF
0xFFFFFD86	CSIE1 transmission data buffer register	CE1TX0	-	-	R/W	-	0x0000
0xFFFFFD86	CSIE1 transmission data buffer register L	CE1TXL0	-	R/W	-	-	0x00
0xFFFFFD87	CSIE1 transmission data buffer register H	CE1TXH0	-	R/W	-	-	0x00
0xFFFFFD88	Queued CSI1 status register	CE1STR	R/W	R/W	-	-	0x20
0xFFFFFD89	Queued CSI1 control register 2	CE1CTL2	R/W	R/W	-	-	0x00
0xFFFFFD8C	Queued CSI1 control register 3	CE1CTL3	R/W	R/W	-	-	0x00
0xFFFFFD8D	Queued CSI1 control register 4	CE1CTL4	R/W	R/W	-	-	0x00
0xFFFFFD90	Queued CSI1 enhanced timing register 0	CE1OPT0	-	-	R/W	-	0x0002
0xFFFFFD92	Queued CSI1 enhanced timing register 1	CE1OPT1	-	-	R/W	-	0x0002
0xFFFFFD94	Queued CSI1 enhanced timing register 2	CE1OPT2	-	-	R/W	-	0x0002
0xFFFFFD96	Queued CSI1 enhanced timing register 3	CE1OPT3	-	-	R/W	-	0x0002
0xFFFFFDC0	Prescaler mode register 0	PRSM0	R/W	R/W	-	-	0x00
0xFFFFFDC1	Prescaler compare registers 0	PRSCM0	R/W	R/W	-	-	0x00
0xFFFFFDD0	Prescaler mode register 1	PRSM1	R/W	R/W	-	-	0x00
0xFFFFFDD1	Prescaler compare registers 1	PRSCM1	R/W	R/W	-	-	0x00
0xFFFFFDE0	Prescaler mode register 2	PRSM2	R/W	R/W	-	-	0x00
0xFFFFFDE1	Prescaler compare registers 2	PRSCM2	R/W	R/W	-	-	0x00
0xFFFFFE00	DMA wait control register 0	DMAWC0	-	R/W	-	-	0x37
0xFFFFFE02	DMA wait control register 1	DMAWC1	-	R/W	-	-	0x07



## Revision History

The following revision list shows all functional changes compared to the previous manual version R01UH0120ED0200 (published on September 24, 2010).

Chapter	Page	Description
4	103	unused interrupt INTTS0CC1B removed
4	103	unused interrupt INTTS0CC2B removed
4	103	unused interrupt INTTS0CC3B removed
4	103	unused interrupt INTTS1CC1B removed
4	103	unused interrupt INTTS1CC2B removed
4	103	unused interrupt INTTS1CC3B removed
4	120	interrupt control registers (C2ERRIC, C2WUPIC, C2ERRIC, C3ERRIC, C3WUPIC, C2ERRIC) removed for unused interrupts INTTS0CC1B, INTTS0CC2B, INTTS0CC3B, INTTS1CC1B, INTTS1CC2B, and INTTS1CC3B
7	185	caution for boot block cluster protection added
11	247	bit name of SBF transmission trigger bit corrected
11	247	bit name of transfer status flag corrected
12	280	access direction of CBnTX0L register corrected to read/write
12	288	misleading reference replaced by dedicated configuration of CBnCKP and CBnDAP bits
12	290	
12	291	
12	293	
12	294	
12	295	
12	297	
12	296	CSIBn transmit data register corrected to CBnTX0
12	301	CSIBn transmit data register corrected in figure to CBnTX0
12	302	CSIBn receive data register corrected in figure to CBnRX0
12	303	CSIBn transmit and receive data register, respectively, corrected in figure to CBnTX0 and CBnRX0
12	304	CSIBn transmit data register corrected in figure to CBnTX0
12	305	CSIBn receive data register corrected in figure to CBnRX0
12	306	CSIBn transmit and receive data register, respectively, corrected in figure to CBnTX0 and CBnRX0
13	326	caution of write to CEnCS register corrected
13	329	caution of write to CEnTX0 register corrected
13	332	access direction for CEnOPTm register corrected
14	370	term "memory access controller" replaced by "message control module" according to block diagram of CAN module
16	563	incorrect item OVFW in figure corrected to OVFV
16	585	formula of PWM duty factor corrected
16	586	
16	585	formula of PWM cycle corrected
16	586	
17	619	description of TSnCCR1B changed

Chapter	Page	Description
17	620	description of TSnCCR2B changed
17	621	description of TSnCCR3B changed
17	647	name of bit 8 of TSnOPT3 register changed to TSnACC2
17	648	
17	802	marking in figure corrected from "TSnCCR0 to TSnDTC1" to "TSnCCR0 - TSnDTC1"
A	1065	reset value and access size of CE0CS register corrected
A	1066	reset value and access size of CE1CS register corrected

The following revision list shows all functional changes compared to the previous manual version R01UH0120ED0210 (published on November 13, 2012)

Chapter	Page	Description
5	143	CPU-CRC removed in figure title
7	170	PG-FP4 replaced by PG-FP5 programmer in all following figures of this chapter
7	173	PG-FP4 replaced by PG-FP5 throughout this chapter
7	186	URL updated
9	193	unintended block diagram of Bus and Memory Control with Flexray controller and external bus interface removed
18	948	several bit identifiers corrected to TTn...
18	962	table "Counter Clear Option" moved to correct location
18	971	notation in figure corrected
18	984	high level setting replaced by active level setting throughout the subchapter
18	984	low level setting replaced by inactive level setting throughout the subchapter
18	984	subtitel of figure corrected (... not rewritten ...)
18	988	indication of operation flow in free-running mode corrected
18	1018	mixed up compare and capture interrupt notation in figure corrected



## Index

### A

- A/D conversion result register for DMA (ADDMAn) 518
- A/D conversion result registers (ADCRnm, ADCRnmH) 515
- A/D Converter
  - Operation in A/D trigger mode 525
  - Operation in external trigger mode 534
  - Operation in timer trigger mode 529
- A/D Converter mode register 0 (ADMn0) 510
- A/D Converter mode register 1 (ADMn1) 511
- A/D Converter mode register 2 (ADMn2) 513
- A/D Converter trigger source select register (ADTRSELn) 514
- AA5OVIC 102
- Access to
  - External devices (initialization) 202
- ADCRnm 515
- ADCRnmH 515
- ADDMAn 518
- Address setup wait control register (AWC) 207
- Address space 85
  - CPU 85
  - Images 85
  - Physical 85
- ADIC0 104, 120
- ADIC1 104, 120
- ADMn0 510
- ADMn1 511
- ADMn2 513
- ADTRSELn 514
- AFO (Auxiliary Frequency Output) 1042
- Analog filtered inputs 63
- Anytime write
  - TMT 966
- Asynchronous Serial Interface
  - see UARTC
- Auxiliary Frequency Output 1042
- AWC 207

### B

- Batch rewrite
  - TMT 968
- Baud rate generator
  - CSIB 307
  - UARTC 270
- BCC 209
- BCTn 206
- BCU

Registers 204

- BCU (Bus Control Unit) 192
- Boundary operation conditions 201
- BPC 204
- BRGIC 103, 119
- BRGIC2 103, 119
- BSC 206
- Bus and memory control 192
  - Registers 203
- Bus clock dividing frequency control register (DVC) 210
- Bus cycle configuration register (BCTn) 206
- Bus cycle control register (BCC) 209
- Bus size configuration register (BSC) 206

### C

- C0ERRIC 103, 119
- C0RECIC 103, 120
- C0TRXIC 103, 120
- C0WUPIC 103, 120
- C1ERRIC 103, 120
- C1RECIC 103, 120
- C1TRXIC 103, 120
- C1WUPIC 103, 120
- CALLT base pointer (CTBP) 81
- CAN (Controller area network) 368
- CAN clock selection register (CANCKSEL) 151
- CAN Controller 368
  - Baud rate settings 469
  - Bit set/clear function 400
  - Configuration 370
  - Connection with target system 392
  - Control registers 402
  - Diagnosis functions 464
  - Functions 381
  - Initialization 438
  - Internal registers 393
  - Interrupt function 463
  - Message reception 442
  - Message transmission 450
  - Operation 477
  - Overview of functions 369
  - Power saving modes 458
  - Register access type 395
  - Register bit configuration 397
  - Special operational modes 464
  - Time stamp function 468
  - Transition from initialization mode to operation mode 440
- CAN protocol 371
- CANCKSEL 151

---

CANn global automatic block transmission control register (CnGMABT) 405	CBnSTR 287
CANn global automatic block transmission delay register (CnGMABTD) 407	CBnTX 145, 280
CANn global clock selection register (CnGMCS) 404	CBnTX0 280
CANn global control register (CnGMCTRL) 402	CBnTX0L 280
CANn message configuration register m (CnMCONFm) 432	CBnTXL 280
CANn message control register m (CnMCTRLm) 435	CCRCINn (CPU-CRC input register) 145
CANn message data byte register (CnMDATAxm) 429	CE0CIC 104, 120
CANn message data length register m (CnMDLCm) 431	CE0OFIC 104, 120
CANn message ID register m (CnMIDLm, CnMIDHm) 434	CE1CIC 104, 120
CANn module bit rate prescaler register (CnBRP) 420	CE1OFIC 104, 120
CANn module bit rate register (CnBTR) 421	CEnCS 326
CANn module control register (CnCTRL) 410	CEnCTL0 316
CANn module error counter register (CnERC) 416	CEnCTL1 318
CANn module information register (CnINFO) 415	CEnCTL2 321
CANn module interrupt enable register (CnIE) 417	CEnCTL3 322
CANn module interrupt status register (CnINTS) 419	CEnCTL4 323
CANn module last error information register (CnLEC) 413	CEnOPTm 332
CANn module last in-pointer register (CnLIPT) 422	CEnRX0 325
CANn module last out-pointer register (CnLOPT) 424	CEnSTR 330
CANn module mask control register (CnMASKaL, CnMASKaH) 408	CEnTX0 329
CANn module receive history list register (CnRGPT) 423	Chip area select control register (CSCn) 206
CANn module time stamp register (CnTS) 427	Chip select data buffer registers (CEnCS) 326
CANn module transmit history list register (CnTGPT) 425	Chip select signals 194
CB0REIC 103, 120	Clock Generator 148
CB0RIC 103, 120	Clock Monitor 152
CB0TIC 103, 120	Clocked Serial Interface
CB1REIC 104, 120	see CSIB
CB1RIC 104, 120	CnBRP 420
CB1TIC 104, 120	CnBTR 421
CBnCTL0 281	CnCTRL 410
CBnCTL1 283	CnERC 416
CBnCTL2 285	CnGMABT 405
CBnRX0 280	CnGMABTD 407
CBnRX0L 280	CnGMCS 404
	CnGMCTRL 402
	CnIE 417
	CnINFO 415
	CnINTS 419
	CnLEC 413
	CnLIPT 422
	CnLOPT 424
	CnMASKaH 408
	CnMASKaL 408
	CnMCONFm 432
	CnMCTRLm 435
	CnMDATAxm 429
	CnMDLCm 431
	CnMIDHm 434
	CnMIDLm 434
	CnRGPT 423

---

---

CnTGPT 425  
 CnTS 427  
 Command register (PRCMD) 38  
 CPU  
     Address space 85  
     Functions 71  
     Register set 73  
 CPU-CRC 142  
 CRC  
     base address 200  
     CCRCINn (CPU-CRC input register) 145  
     CPU-CRC 142  
     CRCC (CRC control register) 146  
     DATA-CRC 142  
     DCRCIN 145  
     Registers 144  
 CRC (Cyclic Redundancy Check) 142  
 CRCC (CRC control register) 146  
 CSCn 206  
 CSIB  
     Baud rate generator 307  
     Control registers 281  
     Operation 288  
     Operation flow 301  
     Output pins 300  
 CSIB (Clocked Serial Interface) 278  
 CSIB transmit data register (CBnTX) 145  
 CSIBn control register 0 (CBnCTL0) 281  
 CSIBn control register 1 (CBnCTL1) 283  
 CSIBn control register 2 (CBnCTL2) 285  
 CSIBn receive data register (CBnRX0) 280  
 CSIBn status register (CBnSTR) 287  
 CSIBn transmit data register (CBnTX) 280  
 CSIBn transmit data register (CBnTX0) 280  
 CTBP 81  
 CTPC 76  
 CTPSW 79  
 Cyclic Redundancy Check (CRC) 142

**D**

Data address space  
     Recommended use 96  
 Data flash 199  
     base address 199  
 Data flash control register (DFLCTL) 169  
 Data space 87  
 Data wait control register (DWCn) 208  
 DATA-CRC 142  
 DBPC 76  
 DBPSW 79

DCRCIN 145  
 Debug function (on-chip) 1034  
     Code protection 187  
     Restrictions and Cautions 1041  
 Debug trap 136  
 DFLCTL 169  
 Digitally filtered inputs 64  
 DMA (direct memory access) 211  
 DMA Controller 211  
 DMA data size control register (DMDSCL) 222  
 DMA mode control register (DMAMCL) 221  
 DMA status register (DMASL) 220  
 DMA transfer  
     Forcible termination 212  
     Serial data reception 232  
     Serial data transmission 236  
     Timer AA capture register 240  
 DMA transfer count registers (DTCRn) 219  
 DMA transfer memory start address registers (MARn) 217  
 DMA transfer SFR start address registers (SARm) 216  
 DMA trigger factor registers (DTFRn) 218  
 DMA wait control registers (DMAWCm) 215  
 DMAIC2 104, 120  
 DMAIC3 104, 120  
 DMAMCL 221  
 DMASL 220  
 DMAWCm 215  
 DMDSCL 222  
 DTCRn 219  
 DTFRn 218  
 DVC 210  
 DWCn 208

**E**

ECR 80  
 ECT 83  
 EFG 82  
 EIPC 76  
 EIPSW 79  
 Element pointer 74  
 Emergency shut off control register (PESCn) 34  
 Emergency shut-off mask register (PESMKn) 37  
 Emergency shut-off status register (ESOSTn) 36  
 ERRIC 100, 118  
 ESOSTn 36  
 Exception status flag (EP) 133  
 Exception trap 134

---

---

External bus properties 201

External devices

- Initialization for access 202

External interrupt edge specification register (INTM0) 124

External interrupt edge specification register (INTM1) 125

External interrupt edge specification register (INTM2) 126

External interrupt edge specification register (INTM3) 126

External memory area 90

**F**

FEPC 76

FEPSW 79

Fixed peripheral I/O area 197

Flash area 89

Flash memory 159

- Self-programming 181

Flash programmer

- Communication mode 171
- Pin connection 173

Flash programming

- Mode 84

Flash ROM ECC error register (ROMEAD) 166

Floating point arithmetic control register (ECT) 83

Floating point arithmetic flag register (EFG) 82

Floating point arithmetic register set 82

**G**

General purpose registers (r0 to r31) 74

Global pointer 74

**H**

HALT mode 156, 157

- release 157

**I**

ID code 1036

Idle pins

- Recommended connection 68

Images in address space 85

IMRn 120

initial value of Pn registers changed to "undefined" 32

Initialization for access to external devices 202

In-service priority register (ISPR) 122

INTC (Interrupt Controller) 99

Internal flash area 89

Internal peripheral function wait control register (VSWC) 205

Internal RAM area 89

INTERRF 130

Interrupt

- Maskable 110
- Non-maskable 105
- Processing (multiple interrupts) 138
- Response time 140

Interrupt Controller 99

- Debug trap 136
- Edge and level detection configuration 124
- Exception trap 134
- Periods in which interrupts are not acknowledged 141
- Software exception 131

Interrupt mask registers IMRn 120

Interrupt share enable register (INTSEL) 129

Interrupt source flag register (INTERRF) 130

Interrupt/exception source register (ECR) 80

INTM0 124

INTM1 125

INTM2 126

INTM3 126

INTSEL 129

ISPR 122

**L**

Link pointer 74

**M**

MARn 217

Maskable interrupt status flag (ID) 123

Maskable interrupts 110

Maskable Interrupts Control Register (xxICn) 117

MEMC (Memory Controller) 192

Memory 89

- Access configuration 210
- Areas 89
- Blocks 194
- Controller registers 206

MODE pins 84

**N**

Noise elimination

- Pin input 63

Noise rejection time control register (NRC) 66

Non-maskable interrupts 105

Normal operation mode 84

NRC 66

N-Wire

---

Code protection 187	Port function control register (PFCn) 31
Connection to emulator 1037	Port groups 24
Emulator 1034	Configuration 40
ID code 1036	Configuration registers 27
Security function 1036	List 40
<b>O</b>	Port mode control register (PMCn) 29
on-chip debug cautions revised 1041	Port mode register (PMn) 30
Operation modes 84	Port register (Pn) 32
Flash programming mode 84	power save function 156
Normal operation mode 84	PPA (programmable peripheral I/O area) 197
OSDIC 100, 118	PRCMD 38
<b>P</b>	Prescaler compare registers (PRSCMn) 309
Package pins assignment 69	Prescaler compare registers 2 (PRSCM2) 1044
PC 76	Prescaler mode register 2 (PRSM2) 1043
PC saving registers 76	Prescaler mode registers (PRSMn) 308
Peripheral area selection control register (BPC) 204	Program counter (PC) 76
Peripheral I/O area	Program space 87
fixed 197	Program status word (PSW) 77
programmable 90, 197	Programmable peripheral I/O area 90, 197
Peripheral status register (PHS) 39	PRSCM2 1044
PESCn 34	PRSCMn 309
PESMKn 37	PRSM2 1043
PFCEn 31	PRSMn 308
PFCn 31	PSW 77
PHS 39	PSW saving registers 79
Physical address space 85	<b>Q</b>
PIC0 100, 118	Queued CSI control register 0 (CEnCTL0) 316
PIC1 100, 118	Queued CSI control register 1 (CEnCTL1) 318
PIC10 100, 118	Queued CSI control register 2 (CEnCTL2) 321
PIC11 100, 118	Queued CSI control register 3 (CEnCTL3) 322
PIC12 100, 118	Queued CSI control register 4 (CEnCTL4) 323
PIC2 100, 118	Queued CSI enhanced timing registers (CEnOPTm) 332
PIC3 100, 118	Queued-CSI status registers (CEnSTR) 330
PIC4 100, 118	<b>R</b>
PIC5 100, 118	RAM area 89
PIC6 100, 118	RAM ECC control register (RAMECC) 94
PIC7 100, 118	RAM ECC error address register (RAMEAD) 92
PIC8 100, 118	RAM ECC error data location register (RAMEDLR) 93
PIC9 100, 118	RAMEAD 92
Pin functions 23	RAMECC 94
After reset/in stand-by modes 67	RAMEDLR 93
List 45	Receive data buffer registers (CEnRX0) 325
Unused pins 68	regID (system register number) 75
PMCn 29	Reload
PMn 30	TMT 968
Pn 32	Reset
Port function control expansion register (PFCEn) 31	Register status after reset 1047

---

ROM	TAA5OVIC 119
see Internal flash area	TAA6CCIC0 102, 119
ROMEAD 166	TAA6CCIC1 102, 119
<b>S</b>	TAA6OVIC 102, 119
SARm 216	TAA7CCIC0 103, 119
SARn register read/write access of 1-bit units added 216	TAA7CCIC1 103, 119
Saturated operation instructions 78	TAA7OVIC 103, 119
SFR (special function register) 1049	TAA8CCIC0 103, 119
Software exception 131	TAA8CCIC1 103, 119
Special function registers (list) 1049	TAA8OVIC 103, 119
Stack pointer 74	TAA9CCIC0 103, 119
System register set 75	TAA9CCIC1 103, 119
<b>T</b>	TAA9OVIC 103, 119
TAA capture/compare register 0 (TAAAnCCR0) 546	TAAIC0 549
TAA capture/compare register 1 (TAAAnCCR1) 547	TAAIC1 550
TAA control register 0 (TAAAnCTL0) 552	TAAIC2 551
TAA counter read buffer register (TAAAnCNT) 548	TAAIC3 1023
TAA dedicated I/O control register 0 (TAAAnIOC0) 555	TAAAnCCR0 546
TAA dedicated I/O control register 1 (TAAAnIOC1) 556	TAAAnCCR1 547
TAA I/O control register 2 (TAAAnIOC2) 557	TAAAnCNT 548
TAA I/O control register 3 (TAAAnIOC3) 558	TAAAnCTL0 552
TAA option register 0 (TAAAnOPT0) 559	TAAAnCTL1 553
TAA option register 1 (TAAAnOPT1) 560	TAAAnIOC0 555
TAA specific overflow value control register 0 (TAAAnSVC) 561	TAAAnIOC1 556
TAA timer control register 1 (TAAAnCTL1) 553	TAAAnIOC2 557
TAA0CCIC0 102, 119	TAAAnIOC3 558
TAA0CCIC1 102, 119	TAAAnOPT0 559
TAA0OVIC 102, 119	TAAAnOPT1 560
TAA1CCIC0 102, 119	TAAAnSVC 561
TAA1CCIC1 102, 119	Text pointer 74
TAA1OVIC 102, 119	Timer I/O control register 3 (TAAIC3) 1023
TAA2CCIC 119	Timer input control register 0 (TAAIC0) 549
TAA2CCIC0 102	Timer input control register 1 (TAAIC1) 550
TAA2CCIC1 102, 119	Timer input control register 2 (TAAIC2) 551
TAA2OVIC 102, 119	Timer S 613
TAA3CCIC0 102, 119	A/D conversion trigger function 746
TAA3CCIC1 102, 119	Basic operation 689
TAA3OVIC 102, 119	Configuration 615
TAA4CCIC0 102, 119	Control register 627
TAA4CCIC1 102, 119	Error/warning interrupt 752
TAA4OVIC 102, 119	Features 613
TAA5CCIC0 102, 119	Flags 716
TAA5CCIC1 102, 119	Interrupt thinning out function 735
	Match interrupt 713
	Operation in each mode 756
	Timer Sn compare register 0 (TSnCCR0) 618
	Timer Sn compare register 0A (TSnCCR0A) 618
	Timer Sn compare register 1 (TSnCCR1) 619
	Timer Sn compare register 1A

---



---

(TSnCCR1A) 619	Anytime write 966
Timer Sn compare register 1B (TSnCCR1B) 619	Batch rewrite 968
Timer Sn compare register 2 (TSnCCR2) 620	Reload 968
Timer Sn compare register 2A (TSnCCR2A) 620	TMTn capture/compare register 0 (TTnCCR0) 944
Timer Sn compare register 2B (TSnCCR2B) 620	TMTn capture/compare register 1 (TTnCCR1) 945
Timer Sn compare register 3 (TSnCCR3) 621	TMTn control register 0 (TTnCTL0) 947
Timer Sn compare register 3A (TSnCCR3A) 621	TMTn control register 1 (TTnCTL1) 949
Timer Sn compare register 3B (TSnCCR3B) 621	TMTn control register 2 (TTnCTL2) 951
Timer Sn compare register 4 (TSnCCR4) 622	TMTn counter read buffer register (TTnCNT) 946
Timer Sn compare register 4A (TSnCCR4A) 622	TMTn counter write buffer register (TTnTCW) 946
Timer Sn compare register 5 (TSnCCR5) 623	TMTn I/O control register 0 (TTnLOC0) 953
Timer Sn compare register 5A (TSnCCR5A) 623	TMTn I/O control register 1 (TTnLOC1) 954
Timer Sn control register 0 (TSnCTL0) 627	TMTn I/O control register 2 (TTnLOC2) 955
Timer Sn control register 1 (TSnCTL1) 629	TMTn I/O control register 3 (TTnLOC3) 956
Timer Sn counter read register (TSnCNT) 624	TMTn option register 0 (TTnOPT0) 958
Timer Sn dead time setting register 0 (TSnDTC0) 625	TMTn option register 1 (TTnOPT1) 959
Timer Sn dead time setting register 0A (TSnDTC0A) 625	TMTn option register 2 (TTnOPT2) 961
Timer Sn dead time setting register 1 (TSnDTC1) 626	Transmission data buffer registers (CEnTX0) 329
Timer Sn dead time setting register 1A (TSnDTC1A) 626	TS0CCIC 101
Timer Sn I/O control register 0 (TSnLOC0) 631	TS0CCIC0 118
Timer Sn I/O control register 2 (TSnLOC2) 633	TS0CCIC1 101, 118
Timer Sn I/O control register 4 (TSnLOC4) 634	TS0CCIC2 101, 118
Timer Sn option register (TSnOPT6) 680	TS0CCIC3 101, 118
Timer Sn option register 0 (TSnOPT0) 636	TS0CCIC4 101, 118
Timer Sn option register 4 (TSnOPT4) 651	TS0CCIC5 101, 118
Timer Sn option register 5 (TSnOPT5) 669	TS0CDIC0 101, 118
Timer Sn option register 7 (TSnOPT7) 683	TS0ERIC 101, 118
Timer Sn option registers 1, 1H, 1L (TSnOPT1, TSnOPT1H, TSnOPT1L) 638	TS0ODIC 101, 118
Timer Sn option registers 1A, 1AH, 1AL (TSnOPT1A, TSnOPT1AH, TSnOPT1AL) 638	TS0OVIC 101, 118
Timer Sn option registers 2, 2H, 2L (TSnOPT2, TSnOPT2H, TSnOPT2L) 643	TS0WNIC 101, 118
Timer Sn option registers 3, 3H, 3L (TSnOPT3, TSnOPT3H, TSnOPT3L) 647	TS1CCIC0 101, 118
Timer Sn pattern register 0 (TSnPAT0) 685	TS1CCIC1 101, 118
Timer Sn pattern register 1 (TSnPAT1) 687	TS1CCIC2 101, 118
Timer Sn sub-counter read register (TSnSBC) 624	TS1CCIC3 101, 118
TMT	TS1CCIC4 101, 118
	TS1CCIC5 101, 118
	TS1CDIC0 101, 118
	TS1ERIC 101, 118
	TS1ODIC 101, 118
	TS1OVIC 101, 118
	TS1WNIC 101, 118
	TSnCCR0 618
	TSnCCR0A 618
	TSnCCR1 619
	TSnCCR1A 619
	TSnCCR1B 619

---

TSnCCR2 620  
 TSnCCR2A 620  
 TSnCCR2B 620  
 TSnCCR3 621  
 TSnCCR3A 621  
 TSnCCR3B 621  
 TSnCCR4 622  
 TSnCCR4A 622  
 TSnCCR5 623  
 TSnCCR5A 623  
 TSnCNT 624  
 TSnCTL0 627  
 TSnCTL1 629  
 TSnDTC0 625  
 TSnDTC0A 625  
 TSnDTC1 626  
 TSnDTC1A 626  
 TSnIOC0 631  
 TSnIOC2 633  
 TSnIOC4 634  
 TSnOPT0 636  
 TSnOPT1, TSnOPT1H, TSnOPT1L 638  
 TSnOPT1A, TSnOPT1AH, TSnOPT1AL 638  
 TSnOPT2, TSnOPT2H, TSnOPT2L 643  
 TSnOPT3, TSnOPT3H, TSnOPT3L 647  
 TSnOPT4 651  
 TSnOPT5 669  
 TSnOPT6 680  
 TSnOPT7 683  
 TSnPAT0 685  
 TSnPAT1 687  
 TSnSBC 624  
 TT0CCIC0 101, 119  
 TT0CCIC1 101, 119  
 TT0ECIC 101, 119  
 TT0OVIC 101, 118  
 TT1CCIC0 102, 119  
 TT1CCIC1 102, 119  
 TT1ECIC 102, 119  
 TT1OVIC 102, 119  
 TTnCCR0 944  
 TTnCCR1 945  
 TTnCNT 946  
 TTnCTL0 947  
 TTnCTL1 949  
 TTnCTL2 951  
 TTnIOC0 953  
 TTnIOC1 954  
 TTnIOC2 955  
 TTnIOC3 956

TTnOPT0 958  
 TTnOPT1 959  
 TTnOPT2 961  
 TTnTCW 946

## U

### UARTC

Cautions 277  
 Dedicated baud rate generator 270  
 Interrupt Request Signals 255  
 Operation 256  
 UARTCn control register 0 (UCnCTL0) 245  
 UARTCn control register 1 (UCnCTL1) 271  
 UARTCn control register 2 (UCnCTL2) 272  
 UARTCn option control register 0  
 (UCnOPT0) 247  
 UARTCn option control register 1  
 (UCnOPT1) 249  
 UARTCn option control register 2  
 (UCnOPT2) 250  
 UARTCn receive data register (UCnRX) 253  
 UARTCn receive shift register 243  
 UARTCn status register (UCnSTR) 251  
 UARTCn transmit data register (UCnTX) 254  
 UARTCn transmit shift register 244  
 UC0REIC 104, 120  
 UC0TIC 104, 120  
 UC1REIC 104, 120  
 UC1TIC 104, 120  
 UCnCTL0 245  
 UCnCTL1 271  
 UCnCTL2 272  
 UCnOPT0 247  
 UCnOPT1 249  
 UCnOPT2 250  
 UCnRX 253  
 UCnSTR 251  
 UCnTX 254

## V

VSWC 205

## W

Write protected registers 97

## Z

Zero register 74



---

V850E/PHC3 User Manual

Publication Date: Rev. 2.20 December 12, 2013

Published by: Renesas Electronics Corporation

---

**SALES OFFICES****Renesas Electronics Corporation**<http://www.renesas.com>Refer to "<http://www.renesas.com/>" for the latest and detailed information.**Renesas Electronics America Inc.**2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.  
Tel: +1-408-588-6000, Fax: +1-408-588-6130**Renesas Electronics Canada Limited**1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada  
Tel: +1-905-898-5441, Fax: +1-905-898-3220**Renesas Electronics Europe Limited**Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.  
Tel: +44-1628-585-100, Fax: +44-1628-585-900**Renesas Electronics Europe GmbH**Arcadiastrasse 10, 40472 Düsseldorf, Germany  
Tel: +49-211-65030, Fax: +49-211-6503-1327**Renesas Electronics (China) Co., Ltd.**7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China  
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679**Renesas Electronics (Shanghai) Co., Ltd.**Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China  
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898**Renesas Electronics Hong Kong Limited**Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong  
Tel: +852-2886-9318, Fax: +852 2886-9022/9044**Renesas Electronics Taiwan Co., Ltd.**7F, No. 363 Fu Shing North Road Taipei, Taiwan  
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670**Renesas Electronics Singapore Pte. Ltd.**1 harbourFront Avenue, #06-10, Keppel Bay Tower, Singapore 098632  
Tel: +65-6213-0200, Fax: +65-6278-8001**Renesas Electronics Malaysia Sdn.Bhd.**Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia  
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510**Renesas Electronics Korea Co., Ltd.**11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea  
Tel: +82-2-558-3737, Fax: +82-2-558-5141

V850E/PHC3