User Manual

32

Hardware

V850E/PH2

32-bit Single-Chip Microcontroller

µPD70F3187
µPD70F3447

Renesas Electronics
www.renesas.com

R01UH0439ED0400, Rev. 4.00
Apr 30, 2013

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.

1. 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.

2. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.

3. 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.

4. 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.

5. 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.

6. 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.

7. 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.
8. 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.
9. 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.
10. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
11. 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

Readers This manual is intended for users who want to understand the functions of the concerned microcontrollers.

Purpose This manual presents the hardware manual for the concerned microcontrollers.

Organization This system specification describes the following sections:

- Pin function
- CPU function
- Internal peripheral function

Module instances 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.

Legend Symbols and notation are used as follows:

- Weight in data notation: Left is high order column, right is low order column
- Active low notation: $\overline{xxx}$ (pin or signal name is over-scored) or /xxx (slash before signal name)
- Memory map address: High order at high stage and low order at low stage

Note Additional remark or tip

Caution Item deserving extra attention

Numeric notation:

- Binary: xxxx or xxx_B
- Decimal: xxxx
- Hexadecimal: $xxxx_H$ or 0x xxxx

Prefixes representing powers of 2 (address space, memory capacity):

- K (kilo): $2^{10} = 1024$
- M (mega): $2^{20} = 1024^2 = 1,048,576$
- G (giga): $2^{30} = 1024^3 = 1,073,741,824$

Register contents: X, x = don’t care

Diagrams 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.

Further Information For further information see *http://www.renesas.eu/*.

SuperFlash(R)

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

This product uses SuperFlash(R) technology licensed from Silicon Technology, Inc.

Table of Contents

Chapter 1 Introduction . . . 18

1.1 General . . . 18
1.2 Features Summary . . . 19
1.3 Ordering Information . . . 21
1.4 Pin Configuration (Top View) . . . 21
1.4.1 **256-pin plastic BGA (21 × 21)** . . . 21
1.4.2 **Pin Identification** . . . 26
1.5 Function Blocks . . . 27
1.5.1 Internal block diagrams . . . 27
1.5.2 On-chip units . . . 29

Chapter 2 Pin Functions . . . 33

2.1 List of Pin Functions . . . 33
2.2 Pin Status . . . 43
2.3 Description of Pin Functions . . . 44
2.3.1 P00 to P04 (Port 0) ... Input . . . 44
2.3.2 P10 to P17 (Port 1) ... Input/Output . . . 45
2.3.3 P20 to P27 (Port 2) ... Input/Output . . . 46
2.3.4 P30 to P37 (Port 3) ... Input/Output . . . 47
2.3.5 P40 to P45 (Port 10) ... Input/Output . . . 48
2.3.6 P50 to P57 (Port 5) ... Input/Output . . . 49
2.3.7 P60 to P67 (Port 6) ... Input/Output . . . 50
2.3.8 P70 to P75 (Port 7) ... Input/Output . . . 51
2.3.9 P80 to P86 (Port 8) ... Input/Output . . . 52
2.3.10 P90 to P96 (Port 9) ... Input/Output . . . 53
2.3.11 P100 to P102 (Port 10) ... Input/Output . . . 54
2.3.12 PAL0 to PAL15 (Port AL) ... I/O . . . 55
2.3.13 PAH0 to PAH5 (Port AH) ... I/O . . . 55
2.3.14 PDL0 to PDL15 (Port DL) ... I/O . . . 56
2.3.15 PDH0 to PDH15 (Port DH) ... I/O . . . 56
2.3.16 PCD2 to PCD5 (Port CD) ... I/O . . . 57
2.3.17 PCM0, PCM1, PCM6, PCM7 (Port CM) ... I/O . . . 57
2.3.18 PCS0, PCS1, PCS3, PCS4 (Port CS) ... I/O . . . 58
2.3.19 PCT4, PCT5 (Port CT) ... I/O . . . 59
2.3.20 DCK (Debug clock) ... Input . . . 59
2.3.21 DDI (Debug data input) ... Input . . . 59
2.3.22 DDO (Debug data output) ... Output . . . 59
2.3.23 DMS (Debug mode select) ... Input . . . 60
2.3.24 $\overline{\text{DRST}}$ (Debug reset) ... Input . . . 60
2.3.25 MODE0 to MODE2 (Mode) ... Input . . . 60
2.3.26 FLMD0, FLMD1 (flash programming mode) . . . 60
2.3.27 $\overline{\text{RESET}}$ (Reset) ... Input . . . 60
2.3.28 X1, X2 (Crystal) . . . 60
2.3.29 ANI00 to ANI09, ANI10 to ANI19 (Analog input) ... Input . . . 60
2.3.30 AV_{REF0}, AV_{REF1} (Analog reference voltage) ... Input . . . 60
2.3.31 AV_{DD} (Analog power supply) . . . 61

2.3.32 AV_{SS} (Analog ground) . . . 61
2.3.33 CV_{DD} (Power supply for clock generator) . . . 61
2.3.34 CV_{SS} (Ground for clock oscillator) . . . 61
2.3.35 V_{DD10} to V_{DD15} (Power supply) . . . 61
2.3.36 V_{DD30} to V_{DD37} (Power supply) . . . 61
2.3.37 V_{SS10} to V_{SS15} (Ground) . . . 61
2.3.38 V_{SS30} to V_{SS37} (Ground) . . . 61
2.4 Pin I/O Circuits and Recommended Connection of Unused Pins . . . 62
2.5 Noise Suppression . . . 66

Chapter 3 CPU Functions . . . 68

3.1 Overview . . . 68
3.1.1 CPU system structure . . . 69
3.2 CPU Register Set . . . 70
3.2.1 Program register set . . . 71
3.2.2 System register set . . . 72
3.2.3 Floating point arithmetic unit register set . . . 78
3.3 Operating Modes . . . 80
3.3.1 Operating modes outline . . . 80
3.3.2 Operation mode specification . . . 81
3.4 Address Space . . . 82
3.4.1 CPU address space . . . 82
3.4.2 Images . . . 83
3.4.3 Wrap-around of CPU address space . . . 84
3.4.4 Memory map . . . 86
3.4.5 Areas . . . 88
3.4.6 Programmable peripheral I/O area . . . 91
3.4.7 Specific registers . . . 93
3.4.8 System wait control register (VSWC) . . . 96
3.4.9 DMA wait control registers 0 and 1 (DMAWC0, DMAWC1) . . . 96
3.4.10 Cautions . . . 97

Chapter 4 Bus Control Function (BCU) . . . 98

4.1 Features . . . 98
4.2 Bus Control Pins . . . 98
4.3 Memory Block Function . . . 99
4.3.1 Chip select control function . . . 100
4.4 Bus Cycle Type Control Function . . . 102
4.5 Bus Access . . . 103
4.5.1 Number of access clocks . . . 103
4.5.2 Bus sizing function . . . 103
4.5.3 Endian control function . . . 104
4.6 Data Bus Access Order . . . 106
4.6.1 8-bit data bus access . . . 106
4.6.2 16-bit data bus access . . . 112
4.6.3 32-bit data bus . . . 118
4.7 Wait Function . . . 126
4.7.1 Programmable wait function . . . 126
4.8 Idle State Insertion Function . . . 129

4.9 Bus Priority Order . . . 132
4.10 Boundary Operation Conditions . . . 133
4.10.1 Program space . . . 133
4.10.2 Data space . . . 133

Chapter 5 Memory Access Control Function . . . 134
5.1 SRAM, External ROM, External I/O Interface . . . 134
5.1.1 Features . . . 134
5.1.2 SRAM connection . . . 135
5.1.3 SRAM, external ROM, external I/O access . . . 137

Chapter 6 DMA Functions (DMA Controller) . . . 145
6.1 Features . . . 145
6.2 Control Registers . . . 146
6.3 DMA Channel Priorities . . . 152
6.4 DMA Operation . . . 152
6.4.1 DMA transfer of A/D converter result registers (ADC0, ADC1) . . . 152
6.4.2 DMA transfer of PWM timer reload (TMR0, TMR1) . . . 156
6.4.3 DMA transfer of serial interfaces . . . 160
6.4.4 Forcible termination of DMA transfer . . . 168
6.5 DMA Interrupt Function . . . 169

Chapter 7 Interrupt/Exception Processing Function . . . 171
7.1Features171
7.2 Non-maskable Interrupt . . . 176
7.2.1 Operation . . . 177
7.2.2 Restore . . . 179
7.2.3 Non-maskable interrupt status flag (NP) . . . 180
7.2.4 Edge Detection Function . . . 180
7.3 Maskable Interrupts . . . 181
7.3.1 Operation . . . 181
7.3.2 Restore . . . 183
7.3.3 Priorities of maskable interrupts . . . 184
7.3.4 Interrupt control register (PICn) . . . 188
7.3.5 Interrupt mask registers 0 to 6 (IMR0 to IMR6) . . . 192
7.3.6 In-service priority register (ISPR) . . . 194
7.3.7 Maskable interrupt status flag (ID) . . . 195
7.3.8 Interrupt trigger mode selection . . . 196
7.4 Software Exception . . . 202
7.4.1 Operation . . . 202
7.4.2 Restore . . . 203
7.4.3 Exception status flag (EP) . . . 204
7.5 Exception Trap . . . 205
7.5.1 Illegal opcode definition . . . 205
7.6 Periods in Which CPU Does Not Acknowledge Interrupts . . . 207

Chapter 8 Clock Generator . . . 208
8.1 Features . . . 208
8.2 Configuration . . . 208

8.3 Power Save Control . . . 209
8.3.1 Overview . . . 209
8.3.2 HALT mode . . . 210

Chapter 9 16-Bit Timer/Event Counter P (TMP) . . . 212

9.1 Features . . . 212
9.2 Function Outline . . . 212
9.3 Configuration . . . 213
9.4 Control Registers . . . 218
9.5 Operation . . . 228
9.5.1 Anytime rewrite and reload . . . 229
9.5.2 Interval timer mode (TPnMD[2:0] = 000B) . . . 234
9.5.3 External event count mode (TPnMD[2:0] = 001B) . . . 237
9.5.4 External trigger pulse output mode (TPnMD[2:0] = 010B) . . . 241
9.5.5 One-shot pulse mode (TPnMD[2:0] = 011B) . . . 244
9.5.6 PWM mode (TPnMD[2:0] = 100B) . . . 247
9.5.7 Free-running mode (TPnMD[2:0] = 101B) . . . 252
9.5.8 Pulse width measurement mode (TPnMD[2:0] = 110B) . . . 259
9.5.9 Counter synchronous operation function . . . 266
9.6 Cautions . . . 267

Chapter 10 16-bit Inverter Timer/Counter R (TMR) . . . 268

10.1 Features . . . 269
10.2 Configuration . . . 270
10.3 Control Registers . . . 281
10.4 Basic Operation . . . 301
10.4.1 Basic counter operation . . . 301
10.4.2 Compare register rewrite operation . . . 303
10.4.3 List of outputs in each mode . . . 316
10.5 Match Interrupts . . . 322
10.5.1 Compare match interrupt related cautions . . . 324
10.6 Flags . . . 326
10.6.1 Up count flags . . . 326
10.6.2 Normal phase/inverted phase simultaneous active detection flag . . . 327
10.6.3 Reload hold flag . . . 328
10.7 Interrupt Thinning Out Function . . . 329
10.7.1 Operation of interrupt thinning out function . . . 330
10.7.2 Operation examples when peak interrupts and valley interrupts occur alternately 332
10.7.3 Interrupt thinning out function during counter saw tooth wave operation . . . 333
10.8 A/D Conversion Trigger Function . . . 334
10.8.1 A/D conversion trigger operation . . . 335
10.9 Error Interrupts . . . 338
10.9.1 Error interrupt and error signal output functions . . . 338
10.10 Operation in Each Mode . . . 342
10.10.1 Interval timer mode . . . 342
10.10.2 External event count mode (TMR1 only) . . . 346
10.10.3 External trigger pulse output mode (TMR1 only) . . . 351
10.10.4 One-shot pulse mode . . . 355
10.10.5 PWM mode . . . 359

10.10.6 Free-running mode . . . 365
10.10.7 Pulse width measurement mode (TMR1 only) . . . 373
10.10.8 Triangular wave PWM mode . . . 375
10.10.9 High-accuracy T-PWM mode . . . 378
10.10.10 PWM mode with dead time . . . 412
10.11 Cautions . . . 420

Chapter 11 16-bit Timer/Event Counter T (TMT) . . . 421

11.1 Features . . . 421
11.2 Function Outline . . . 421
11.3 Configuration . . . 422
11.4 Control Registers . . . 428
11.5 Basic Operation . . . 443
11.5.1 Basic counter operation . . . 443
11.5.2 Method for writing to compare register . . . 447
11.6 Operation in Each Mode . . . 451
11.6.1 Interval timer mode . . . 451
11.6.2 External event count mode . . . 454
11.6.3 External trigger pulse output mode . . . 458
11.6.4 One-shot pulse mode . . . 461
11.6.5 PWM mode . . . 464
11.6.6 Free-running mode . . . 468
11.6.7 Pulse width measurement mode . . . 474
11.6.8 Triangular wave PWM mode . . . 475
11.6.9 Encoder count function . . . 476
11.6.10 Offset trigger generation mode . . . 499
11.7 Cautions . . . 501

Chapter 12 16-bit 2-Phase Encoder Input Up/Down Counter/General Purpose Timer (TMENC10) 503

12.1 Features . . . 503
12.2 Function Outline . . . 504
12.3 Basic Configuration . . . 505
12.4 Control Registers . . . 511
12.5 Operation . . . 520
12.5.1 Basic operation . . . 520
12.5.2 Operation in general-purpose timer mode . . . 521
12.5.3 Operation in UDC mode . . . 524
12.6 Supplementary Description of Internal Operation . . . 531
12.6.1 Clearing of count value in UDC mode B . . . 531
12.6.2 Clearing of count value upon occurrence of compare match . . . 532
12.6.3 Transfer operation . . . 533
12.6.4 Interrupt signal output upon compare match . . . 534
12.6.5 TM1UBD flag operation . . . 534

Chapter 13 Auxiliary Frequency Output Function (AFO) . . . 535

13.1 Features . . . 535
13.2 Configuration . . . 535
13.3 Control Registers . . . 536

13.4 Operation . . . 538
13.4.1 Auxiliary frequency output . . . 538
13.4.2 Auxiliary frequency generation . . . 538
13.4.3 Interval timer function . . . 538

Chapter 14 A/D Converter . . . 539

14.1 Features . . . 539
14.2 Configuration . . . 540
14.3 Control Registers . . . 543
14.4 Operation . . . 552
14.4.1 Basic operation . . . 552
14.4.2 Operation mode and trigger mode . . . 553
14.5 Operation in A/D Trigger Mode . . . 558
14.5.1 Select mode operation . . . 558
14.5.2 Scan mode operations . . . 560
14.6 Operation in Timer Trigger Mode . . . 561
14.6.1 Select mode operation . . . 561
14.6.2 Scan mode operation . . . 565
14.7 Operation in External Trigger Mode . . . 567
14.7.1 Select mode operations . . . 567
14.7.2 Scan mode operation . . . 570
14.8 Precautions . . . 572

Chapter 15 Asynchronous Serial Interface (UARTC) . . . 574

15.1 Features . . . 574
15.2 Configuration . . . 575
15.3 Control Registers . . . 578
15.4 Interrupt Request Signals . . . 587
15.5 Operation . . . 588
15.5.1 Data format . . . 588
15.5.2 SBF transmission/reception format . . . 590
15.5.3 SBF transmission . . . 592
15.5.4 SBF reception . . . 592
15.5.5 UART transmission . . . 594
15.5.6 Continuous transmission procedure . . . 595
15.5.7 UART reception . . . 597
15.5.8 Reception errors . . . 599
15.5.9 Parity types and operations . . . 599
15.5.10 Receive data noise filter . . . 601
15.6 Baud Rate Generator . . . 602
15.6.1 Baud rate generator configuration . . . 602
15.6.2 Control registers . . . 603
15.6.3 Baud rate . . . 605
15.6.4 Baud rate error . . . 605
15.6.5 Baud rate setting example . . . 605
15.6.6 Allowable baud rate range during reception . . . 606
15.6.7 Baud rate during continuous transmission . . . 608
15.7 Cautions . . . 609

Chapter 16 Clocked Serial Interface B (CSIB) 610

16.1 Features 610
16.2 Configuration 611
16.3 CSIB Control Registers 612
16.4 Operation 622
16.4.1 Single transfer mode (master mode, transmission/reception mode) 622
16.4.2 Single transfer mode (master mode, reception mode) 624
16.4.3 Continuous mode (master mode, transmission/reception mode) 625
16.4.4 Continuous mode (master mode, reception mode) 626
16.4.5 Continuous reception mode (error) 627
16.4.6 Continuous mode (slave mode, transmission/reception mode) 629
16.4.7 Continuous mode (slave mode, reception mode) 631
16.4.8 Clock timing 632
16.5 Output Pins 634
16.6 Operation Flow 635
16.7 Baud Rate Generator 641
16.7.1 Overview 641
16.7.2 Baud rate generator registers 642
16.7.3 Baud rate generation 643
16.8 Cautions 644

Chapter 17 Clocked Serial Interface 3 (CSI3) 645

17.1 Features 645
17.2 Configuration 646
17.3 Control Registers 648
17.4 Dedicated Baud Rate Generator 3n (BRG3n) 662
17.5 Operation 664
17.5.1 Operation modes 664
17.5.2 Function of CSI data buffer register (CSIBUFn) 665
17.5.3 Data transfer direction specification function 667
17.5.4 Transfer data length changing function 669
17.5.5 Serial clock and data phase selection function 670
17.5.6 Master mode 671
17.5.7 Slave mode 672
17.5.8 Transfer clock selection function 673
17.5.9 Single mode 673
17.5.10 Consecutive mode 675
17.5.11 Transmission mode 677
17.5.12 Reception mode 677
17.5.13 Transmission/reception mode 677
17.5.14 Delay control of transmission/reception completion interrupt (INTC3n) 678
17.5.15 Transfer wait function 679
17.5.16 Output pins 681
17.5.17 CSIBUFn overflow interrupt signal (INTC3nOVF) 682
17.6 Operating Procedures 683
17.6.1 Single mode (master mode, transmission mode) 683
17.6.2 Single mode (master mode, reception mode) 684
17.6.3 Single mode (master mode, transmission/reception mode) 686
17.6.4 Single mode (slave mode, transmission mode) 688

17.6.5 Single mode (slave mode, reception mode) . . . 690
17.6.6 Single mode (slave mode, transmission/reception mode) . . . 692
17.6.7 Consecutive mode (master mode, transmission mode) . . . 694
17.6.8 Consecutive mode (master mode, reception mode) . . . 696
17.6.9 Consecutive mode (master mode, transmission/reception mode) . . . 698
17.6.10 Consecutive mode (slave mode, transmission mode) . . . 700
17.6.11 Consecutive mode (slave mode, reception mode) . . . 702
17.6.12 Consecutive mode (in slave mode and transmission/reception mode) . . . 704
17.7 Cautions . . . 706

Chapter 18 CAN Controller (CAN) . . . 707

18.1 Features . . . 708
18.1.1 Overview of functions . . . 709
18.1.2 Configuration . . . 710
18.2 CAN Protocol . . . 711
18.2.1 Frame format . . . 711
18.2.2 Frame types . . . 712
18.2.3 Data frame and remote frame . . . 712
18.2.4 Error frame . . . 719
18.2.5 Overload frame . . . 720
18.3 Functions . . . 721
18.3.1 Determining bus priority . . . 721
18.3.2 Bit stuffing . . . 721
18.3.3 Multi masters . . . 722
18.3.4 Multi cast . . . 722
18.3.5 CAN sleep mode/CAN stop mode function . . . 722
18.3.6 Error control function . . . 722
18.3.7 Baud rate control function . . . 729
18.4 Connection with Target System . . . 732
18.5 Internal Registers of CAN Controller . . . 733
18.5.1 CAN module register and message buffer addresses . . . 733
18.5.2 CAN Controller configuration . . . 734
18.5.3 CAN registers overview . . . 735
18.5.4 Register bit configuration . . . 737
18.6 Bit Set/Clear Function . . . 740
18.7 Control Registers . . . 742
18.8 CAN Controller Initialization . . . 778
18.8.1 Initialization of CAN module . . . 778
18.8.2 Initialization of message buffer . . . 778
18.8.3 Redefinition of message buffer . . . 778
18.8.4 Transition from initialization mode to operation mode . . . 780
18.8.5 Resetting error counter CnERC of CAN module . . . 781
18.9 Message Reception . . . 782
18.9.1 Message reception . . . 782
18.9.2 Receive data read . . . 783
18.9.3 Receive history list function . . . 784
18.9.4 Mask function . . . 786
18.9.5 Multi buffer receive block function . . . 788
18.9.6 Remote frame reception . . . 789
18.10 Message Transmission . . . 790

18.10.1 Message transmission . . . 790
18.10.2 Transmit history list function . . . 792
18.10.3 Automatic block transmission (ABT) . . . 794
18.10.4 Transmission abort process . . . 796
18.10.5 Remote frame transmission . . . 797
18.11 Power Saving Modes . . . 798
18.11.1 CAN sleep mode . . . 798
18.11.2 CAN stop mode . . . 801
18.11.3 Example of using power saving modes . . . 802
18.12 Interrupt Function . . . 803
18.13 Diagnosis Functions and Special Operational Modes . . . 804
18.13.1 Receive-only mode . . . 804
18.13.2 Single-shot mode . . . 805
18.13.3 Self-test mode . . . 806
18.13.4 Receive/transmit operation in each operation mode . . . 807
18.14 Time Stamp Function . . . 808
18.14.1 Time stamp function . . . 808
18.15 Baud Rate Settings . . . 809
18.15.1 Baud rate setting conditions . . . 809
18.15.2 Representative examples of baud rate settings . . . 813
18.16 Operation of CAN Controller . . . 817

Chapter 19 Random Number Generator (RNG) . . . 844
19.1 Features . . . 844
19.2 Configuration . . . 844
19.3 Operation . . . 845
19.3.1 Access timing . . . 845

Chapter 20 Port Functions . . . 846
20.1 Features . . . 846
20.2 Port Configuration . . . 847
20.2.1 Function of each port . . . 848
20.2.2 Port types . . . 849
20.2.3 Peripheral registers of I/O ports . . . 867
20.2.4 Peripheral registers of valid edge control . . . 869
20.3 Port Pin Functions . . . 870
20.3.1 Port 0 . . . 870
20.3.2 Port 1 . . . 871
20.3.3 Port 2 . . . 875
20.3.4 Port 3 . . . 879
20.3.5 Port 4 . . . 883
20.3.6 Port 5 . . . 886
20.3.7 Port 6 . . . 891
20.3.8 Port 7 . . . 897
20.3.9 Port 8 . . . 901
20.3.10 Port 9 . . . 905
20.3.11 Port 10 . . . 909
20.3.12 Port AL . . . 912
20.3.13 Port AH . . . 916

20.3.14 Port DL . . . 919
20.3.15 Port DH . . . 923
20.3.16 Port CS . . . 927
20.3.17 Port CT . . . 930
20.3.18 Port CM . . . 933
20.3.19 Port CD . . . 936
20.4 Noise Elimination . . . 939

Chapter 21 Reset Function . . . 943
21.1 Features . . . 943
21.2 Configuration . . . 943
21.3 Operation . . . 944

Chapter 22 Internal RAM Parity Check Function . . . 945
22.1 Features . . . 945
22.2 Operation . . . 945
22.3 Control Registers . . . 946

Chapter 23 On-Chip Debug Function (OCD) . . . 948
23.1 Function Overview . . . 948
23.1.1 On-chip debug unit type . . . 948
23.1.2 Debug function . . . 948
23.2 Connection with N-Wire Type Emulator . . . 950
23.2.1 KEL connector . . . 950
23.3 Precautions . . . 954

Chapter 24 Flash Memory . . . 955
24.1 Features . . . 955
24.2 Memory Configuration . . . 956
24.3 Functional Outline . . . 958
24.4 Rewriting by Dedicated Flash Programmer . . . 960
24.4.1 Programming environment . . . 960
24.4.2 Communication mode . . . 961
24.4.3 Pin connections with flash programmer PG-FP5 . . . 963
24.4.4 Flash memory control . . . 964
24.4.5 Selection of communication mode . . . 965
24.4.6 Communication commands . . . 966
24.4.7 Pin connection . . . 967
24.5 Rewriting by Self Programming . . . 971
24.5.1 Overview . . . 971
24.5.2 Features . . . 972

Chapter 25 Electrical Specifications . . . 974
25.1 Absolute Maximum Ratings . . . 974
25.2 General Characteristics . . . 976
25.2.1 Capacitance . . . 976
25.2.2 Operating conditions . . . 976
25.2.3 Oscillator characteristics . . . 977
25.3 DC Characteristics . . . 978

25.4 AC Characteristics 980
25.4.1 External asynchronous memory access read timing 981
25.4.2 External asynchronous memory access write timing 982
25.4.3 Reset Timing (Power Up/Down Sequence) 983
25.4.4 Interrupt timing 984
25.5 Peripheral Characteristics 985
25.5.1 Timer characteristics 985
25.5.2 Serial interface characteristics 987
25.5.3 A/D converter characteristics 994
25.6 Flash Programming Characteristics 995

Chapter 26 Package Drawings 997

Chapter 27 Recommended Soldering Conditions 998

Appendix A Special Function Registers 999
A.1 CAN Registers 999
A.2 Peripheral I/O registers list 1002

Revision History 1016

Index 1020

Chapter 1 Introduction

The V850E/PH2 is a product of the Renesas Electronics V850 family of single-chip microcontrollers designed for automotive applications.

1.1 General

The V850E/PH2 single-chip microcontroller devices make the performance gains attainable with 32-bit RISC-based controllers available for embedded control applications.

The V850E/PH2 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/PH2 product is ideally suited for automotive control and electric power steering (EPS) applications. It is also an excellent choice for other applications where a combination of sophisticated peripheral functions and CAN network support is required.

(1) V850E1 CPU

The V850E1 CPU supports a RISC instruction set that enhances the performance of the V850 CPU, which is the CPU core integrated in the V850 Series, and has added instructions supporting high-level languages, such as C-language switch statement processing, table look-up branching, stack frame creation/deletion, and data conversion. This enhances the performance of both data processing and control. It is possible to use the software resources of the V850 CPU integrated system since the instruction codes of the V850E1 are upwardly compatible at the object code level with those of the V850 CPU.

In addition, the V850E1 CPU of this product incorporates a single-precision floating point unit, which supports high speed floating point arithmetic operations.

(2) External memory interface function

The V850E/PH2 microcontroller features n on-chip external memory interface including separately configured address (22 bits) and data (32 bits) buses. SRAM and ROM can be connected.

(3) On-chip flash memory

The V850E/PH2 microcontroller has a quickly accessible flash memory on-chip, that can shorten system development time since it is possible to rewrite a program with the V850E/PH2 microcontroller mounted in an application system. Moreover, it can greatly improve maintain ability after system ships.

(4) A full range of development environment products

A development environment system that includes an optimized C compiler, debugger, in-circuit emulator, simulator, system performance analyser, and other elements is also available.

1.2 Features Summary

- 96 instructions
- 32 general-purpose registers (each of 32 bits width)
- Instruction set:
  - V850E1 (compatible with V850, dditional powerful instructions for reducing code and increasing execution speed)
  - Signed multiplication (16 bits × 16 bits → 32 bits or 32 bits × 32 bits → 64 bits) in 1 to 2 clocks
  - Saturated operation instructions (with overflow/underflow detection)
  - Floating-point arithmetic instructions (single precision, 32 bits, according to IEEE 754-85 standard)
  - 32-bit shift instructions in 1 clock cycle
  - Bit manipulation instructions
  - Load/store instructions with long/short format
  - Signed load instructions
- 64 MB linear address space

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

Table 1-1 V850E/PH2 features (1/2)

Features		V850E/PH2
CPU		V850E1 (32-bit RISC)
		Floating-point arithmetic unit (single precision, according to IEEE 754-85 standard)
Internal memory	Code Flash	• µPD70F3447: 384 KB • µPD70F3187: 512 KB
	Mask ROM	-
	RAM	• µPD70F3447: 24 KB • µPD70F3187: 32 KB
External memory interface	Address bus	22 bit
	Data bus	sizeable 32, 16, 8 bit
	Features	• Four programmable chip select areas • Wait state insertion function • Idle state insertion function • Endian switch function
Operating clocks	CPU frequency	max. 64 MHz
	MainOSC	operates on 16 MHz crystal
	PLL ratio	x 4
Interrupts	Non-maskable	1 ch
	Maskable int.	85 ch
	Maskable ext.	14 ch

Table 1-1 V850E/PH2 features (2/2)

Features		V850E/PH2
I/O lines	I/O ports	137
	Input ports	5
Timers	TMP	9 ch (16-bit general purpose timer/counter, cascadable for 32-bit operations)
	TMR	2 ch (16-bit timer/counter with Motor Control Functions)
	TMT	2 ch (16-bit general purpose timer/counter with PWM functions and encoder function)
	TMENC10	1 ch (16-bit general purpose timer/counter for 4-quadrant emcoding) [a]
A/D converters	Analog inputs	2 × 10
	Resolution	10 bit
	Type	Successive approximation
Serial interfaces	UARTC	2 ch
	CSIB	2 ch clocked cerial interface [b]
	CSI3	2 ch queued CSI [b]
	CAN	2 ch CAN [b] (32 message buffers for each channel)
DMA		8 ch
Other functions	Power save modes	HALT
	RNG	Random Number Generator provided [a]
	Aux. frequency output	Programmable baud rate generator provided
	On-chip debug	Connection of an external N-Wire emulator provided
Power supply		• 3.3 V ± 0.3 V (external I/O buffer, A/D converter) • 1.5V ± 0.15 V (internal power supply, clock generator) (refer to Data Sheet)
Package		256 pin BGA (1.0 mm ball pitch)

a) Not available on µPD70F3447
b) Only one channel available on µPD70F3447

1.3 Ordering Information

Table 1-2 Ordering information

Part Number	Package	Flash/ RAM
µPD70F3187F1(A2)-64-JN4	256-pin plastic BGA (21 × 21)	512 KB/ 32 KB
µPD70F3447F1(A2)-64-JN4	256-pin plastic BGA (21 × 21)	384 KB/ 24 KB

1.4 Pin Configuration (Top View)

1.4.1 256-pin plastic BGA (21 × 21)

µPD70F3187F1(A2)-JN4

µPD70F3447F1(A2)-JN4

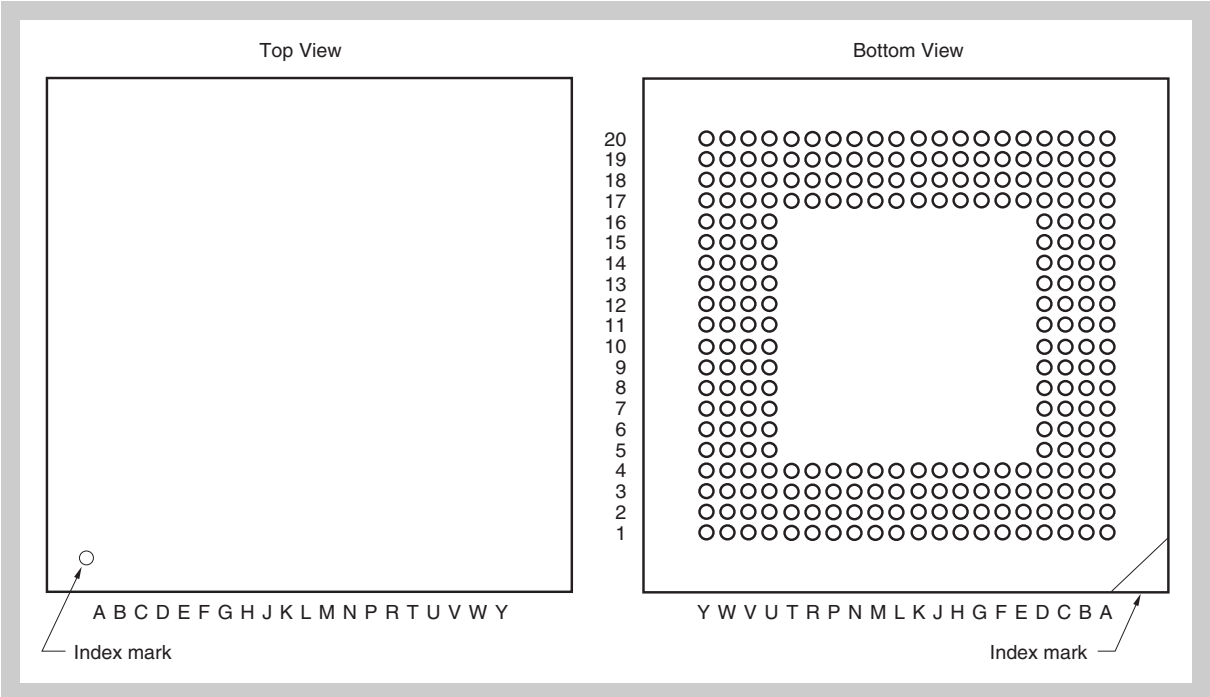


Figure 1-1 Pin Configuration 256-pin Plastic BGA (21 × 21)

Table 1-3 Pin Assignment of 256-pin Plastic BGA (1/4)

Pin No	Pin Function (Name)		Pin No	Pin Function (Name)	
	μPD70F3187	μPD70F3447		μPD70F3187	μPD70F3447
A1	NC	NC	B1	NC	NC
A2	NC	NC	B2	NC	NC
A3	PCT4/$\overline{RD}$	PCT4	B3	PCD5/$\overline{BEN3}$	PCD5
A4	PCT5/$\overline{WR}$	PCT5	B4	PCD2/$\overline{BEN0}$	PCD2
A5	PDH15/D31	PDH15	B5	PCM6	PCM6
A6	PDH13/D29	PDH13	B6	PCM1	PCM1
A7	PDH11/D27	PDH11	B7	PDH14/D30	PDH14
A8	PDH9/D25	PDH9	B8	PDH12/D28	PDH12
A9	PDH8/D24	PDH8	B9	PDH10/D26	PDH10
A10	PDH6/D22	PDH6	B10	PDH7/D23	PDH7
A11	PDH3/D19	PDH3	B11	PDH5/D21	PDH5
A12	PDH0/D16	PDH0	B12	PDH1/D17	PDH1
A13	PDL15/D15	PDL15	B13	PDL13/D13	PDL13
A14	PDL14/D14	PDL14	B14	PDL12/D12	PDL12
A15	PDL9/D9	PDL9	B15	PDL8/D8	PDL8
A16	PDL5/D5	PDL5	B16	PDL4/D4	PDL4
A17	PDL1/D1	PDL1	B17	PDL3/D3	PDL3
A18	PDL0/D0	PDL0	B18	PCS4/$\overline{CS4}$	PCS4
A19	NC	NC	B19	NC	NC
A20	NC	NC	B20	NC	NC
C1	AV_{SS0}	AV_{SS0}	D1	AV_{SS0}	AV_{SS0}
C2	MODE1	MODE1	D2	AV_{SS0}	AV_{SS0}
C3	MODE2	MODE2	D3	AV_{SS0}	AV_{SS0}
C4	PCD4/$\overline{BEN2}$	PCD4	D4	PCM0/$\overline{WAIT}$	PCM0
C5	PCM7	PCM7	D5	PCD3/$\overline{BEN1}$	PCD3
C6	V_{SS37}	V_{SS37}	D6	V_{DD37}	V_{DD37}
C7	V_{SS37}	V_{SS37}	D7	V_{DD37}	V_{DD37}
C8	V_{SS15}	V_{SS15}	D8	V_{DD15}	V_{DD15}
C9	V_{SS15}	V_{SS15}	D9	V_{DD15}	V_{DD15}
C10	PDH4/D20	PDH4	D10	PDH2/D18	PDH2
C11	V_{SS36}	V_{SS36}	D11	V_{DD36}	V_{DD36}
C12	V_{SS36}	V_{SS36}	D12	V_{DD36}	V_{DD36}
C13	PDL11/D11	PDL11	D13	PDL10/D10	PDL10
C14	V_{SS14}	V_{SS14}	D14	V_{DD14}	V_{DD14}
C15	PDL7/D7	PDL7	D15	PDL6/D6	PDL6
C16	V_{SS35}	V_{SS35}	D16	V_{DD35}	V_{DD35}
C17	V_{SS35}	V_{SS35}	D17	V_{DD35}	V_{DD35}
C18	PDL2/D2	PDL2	D18	PAH4/A20	PAH4
C19	PAH5/A21	PAH5	D19	PAH3/A19	PAH3
C20	NC	NC	D20	PAL14/A14	PAL14

Table 1-3 Pin Assignment of 256-pin Plastic BGA (2/4)

Pin No	Pin Function (Name)		Pin No	Pin Function (Name)	
	μPD70F3187	μPD70F3447		μPD70F3187	μPD70F3447
E1	ANI00	ANI00	F1	ANI03	ANI03
E2	ANI02	ANI02	F2	ANI06	ANI06
E3	ANI01	ANI01	F3	ANI05	ANI05
E4	AV_{SS0}	AV_{SS0}	F4	ANI04	ANI04
E17	PAH0/A16	PAH0	F17	PAL12/A12	PAL12
E18	PAH2/A18	PAH2	F18	PAL15/A15	PAL15
E19	PAH1/A17	PAH1	F19	PAL13/A13	PAL13
E20	PCS3/$\overline{CS3}$	PCS3	F20	PAL11/A11	PAL11
G1	ANI07	ANI07	H1	ANI18	ANI18
G2	ANI09	ANI09	H2	ANI19	ANI19
G3	ANI08	ANI08	H3	AV_{DD}	AV_{DD}
G4	AV_{REF0}	AV_{REF0}	H4	AV_{REF1}	AV_{REF1}
G17	V_{DD34}	V_{DD34}	H17	V_{DD34}	V_{DD34}
G18	V_{SS34}	V_{SS34}	H18	V_{SS34}	V_{SS34}
G19	PCS1/$\overline{CS1}$	PCS1	H19	PAL10/A10	PAL10
G20	PAL9/A9	PAL9	H20	PAL6/A6	PAL6
J1	ANI17	ANI17	K1	ANI13	ANI13
J2	ANI14	ANI14	K2	ANI10	ANI10
J3	ANI15	ANI15	K3	ANI11	ANI11
J4	ANI16	ANI16	K4	ANI12	ANI12
J17	PAL5/A5	PAL5	K17	V_{DD13}	V_{DD13}
J18	PAL8/A8	PAL8	K18	V_{SS13}	V_{SS13}
J19	PAL7/A7	PAL7	K19	PAL4/A4	PAL4
J20	MODE0	MODE0	K20	PAL2/A2	PAL2
L1	AV_{SS1}	AV_{SS1}	M1	P01/INTP0/ESO0	P01/INTP0/ESO0
L2	AV_{SS1}	AV_{SS1}	M2	P00/NMI	P00/NMI
L3	AV_{SS1}	AV_{SS1}	M3	V_{SS10}	V_{SS10}
L4	AV_{SS1}	AV_{SS1}	M4	V_{DD10}	V_{DD10}
L17	V_{DD13}	V_{DD13}	M17	P95/SCS312/INTP11	P95/INTP11
L18	V_{SS13}	V_{SS13}	M18	PAL0	PAL0
L19	PAL3/A3	PAL3	M19	PCS0	PCS0
L20	PAL1/A1	PAL1	M20	P42/$\overline{SCKB0}$	P42/$\overline{SCKB0}$
N1	P02/INTP1/ESO1	P02/INTP1/ESO1	P1	P04/INTP3/ADTRG1	P04/INTP3/ADTRG1
N2	P03/INTP2/ADTRG0	P03/INTP2/ADTRG0	P2	P10/TIP00/TEVTP1/ TOP00	P10/TIP00/TEVTP1/ TOP00
N3	V_{SS10}	V_{SS10}	P3	V_{SS30}	V_{SS30}
N4	V_{DD10}	V_{DD10}	P4	V_{DD30}	V_{DD30}
N17	V_{DD33}	V_{DD33}	P17	V_{DD33}	V_{DD33}
N18	V_{SS33}	V_{SS33}	P18	V_{SS33}	V_{SS33}
N19	P41/SOB0	P41/SOB0	P19	P96/SCS313/$\overline{SSB1}$	P96
N20	P40/SIB0	P40/SIB0	P20	P94/SCS311/INTP10	P94/INTP10

Table 1-3 Pin Assignment of 256-pin Plastic BGA (3/4)

Pin No	Pin Function (Name)		Pin No	Pin Function (Name)	
	μPD70F3187	μPD70F3447		μPD70F3187	μPD70F3447
R1	P11/TIP01/TTRGP1/TOP01	P11/TIP01/TTRGP1/TOP01	T1	P13/TIP11/TEVTP0/TOP11	P13/TIP11/TEVTP0/TOP11
R2	P12/TIP10/TTRGP0/TOP10	P12/TIP10/TTRGP0/TOP10	T2	P14/TIP20/TEVTP3/TOP20	P14/TIP20/TEVTP3/TOP20
R3	V_{SS30}	V_{SS30}	T3	P16/TIP30/TTRGP2/TOP30	P16/TIP30/TTRGP2/TOP30
R4	V_{DD30}	V_{DD30}	T4	P21/TIP41/TTRGP5/TOP41	P21/TIP41/TTRGP5/TOP41
R17	P83/SCS300/INTP6	P83/SCS300/INTP6	T17	P80/SI30	P80/SI30
R18	P86/SCS303/$\overline{SSB0}$	P86/SCS303/$\overline{SSB0}$	T18	P84/SCS301/INTP7	P84/SCS301/INTP7
R19	P93/SCS310/INTP9	P93/INTP9	T19	P90/SI31	P90
R20	P85/SCS302/INTP8	P85/SCS302/INTP8	T20	P91/SO31	P91
U1	P15/TIP21/TTRGP3/TOP21	P15/TIP21/TTRGP3/TOP21	V1	P20/TIP40/TEVTP5/TOP40	P20/TIP40/TEVTP5/TOP40
U2	P17/TIP31/TEVTP2/TOP31	P17/TIP31/TEVTP2/TOP31	V2	P23/TIP51/TEVTP4/TOP51	P23/TIP51/TEVTP4/TOP51
U3	P22/TIP50/TTRGP4/TOP50	P22/TIP50/TTRGP4/TOP50	V3	P24/TIP60/TEVTP7/TOP60	P24/TIP60/TEVTP7/TOP60
U4	P25/TIP61/TTRGP7/TOP61	P25/TIP61/TTRGP7/TOP61	V4	P70/TIT00/TEVTT1/TOT00/TENCT00	P70/TIT00/TEVTT1/TOT00/TENCT00
U5	P71/TIT01/TTRGT1/TOT01/TENCT01	P71/TIT01/TTRGT1/TOT01/TENCT01	V5	P74/TIT11/TEVTT0/TOT11/TENCT11	P74/TIT11/TEVTT0/TOT11/TENCT11
U6	P75	P75	V6	P102/TIUD1/TO1	P102
U7	V_{DD11}	V_{DD11}	V7	V_{SS11}	V_{SS11}
U8	V_{DD31}	V_{DD31}	V8	V_{SS31}	V_{SS31}
U9	P62/TOR12/TIR11	P62/TOR12/TIR11	V9	P61/TOR11/TIR10	P61/TOR11/TIR10
U10	V_{SS31}	V_{SS31}	V10	V_{SS31}	V_{SS31}
U11	DCK	DCK	V11	$\overline{RESET}$	$\overline{RESET}$
U12	V_{DD12}	V_{DD12}	V12	V_{SS12}	V_{SS12}
U13	V_{DD12}	V_{DD12}	V13	V_{SS12}	V_{SS12}
U14	V_{DD32}	V_{DD32}	V14	V_{SS32}	V_{SS32}
U15	V_{DD32}	V_{DD32}	V15	V_{SS32}	V_{SS32}
U16	P32/RXDC1/INTP5	P32/INTP5	V16	P67/TOR17/TEVTR1	P67/TOR17/TEVTR1
U17	P81/SO30	P81/SO30	V17	P31/TXDC0	P31/TXDC0
U18	P82/$\overline{SCK30}$	P82/$\overline{SCK30}$	V18	P30/RXDC0/INTP4	P30/RXDC0/INTP4
U19	P44/SOB1	P44	V19	P43/SIB1	P43
U20	P92/$\overline{SCK31}$	P92	V20	P45/$\overline{SCKB1}$	P45
W1	NC	NC	Y1	NC	NC
W2	P26	P26	Y2	NC	NC
W3	P27/TIP71/TEVTP6/TOP71	P27/TIP71/TEVTP6/TOP71	Y3	P72/TECRT0/INTP12	P72/INTP12
W4	P73/TIT10/TTRGT0/TOT10/TENCT10	P73/TIT10/TTRGT0/TOT10/TENCT10	Y4	P100/TCLR1/TICC10/TOP81	P100/TOP81

Table 1-3 Pin Assignment of 256-pin Plastic BGA (4/4)

Pin No	Pin Function (Name)		Pin No	Pin Function (Name)	
	μPD70F3187	μPD70F3447		μPD70F3187	μPD70F3447
W5	P101/TCUD1/TICC11	P101	Y5	P50/TOR00	P50/TOR00
W6	P51/TOR01	P51/TOR01	Y6	P52/TOR02	P52/TOR02
W7	P53/TOR03	P53/TOR03	Y7	P54/TOR04	P54/TOR04
W8	P55/TOR05	P55/TOR05	Y8	P56/TOR06	P56/TOR06
W9	P57/TOR07	P57/TOR07	Y9	P60/TOR10/TTRGR1	P60/TOR10/TTRGR1
W10	V_{SS31}	V_{SS31}	Y10	V_{SS31}	V_{SS31}
W11	X2	X2	Y11	CV_{SS}	CV_{SS}
W12	X1	X1	Y12	CV_{DD}	CV_{DD}
W13	DMS	DMS	Y13	$\overline{DRST}$	$\overline{DRST}$
W14	DDO	DDO	Y14	DDI	DDI
W15	P65/TOR15	P65/TOR15	Y15	P63/TOR13/TIR12	P63/TOR13/TIR12
W16	P66/TOR16	P66/TOR16	Y16	P64/TOR14/TIR13	P64/TOR14/TIR13
W17	P34/FCRXD0	P34/FCRXD0	Y17	P35/FCTXD0	P35/FCTXD0
W18	P36/FCRXD1	P36	Y18	P37/FCTXD1	P37
W19	P33/TXDC1	P33	Y19	NC	NC
W20	NC	NC	Y20	NC	NC

1.4.2 Pin Identification

A0 to A21:	Address bus
ADTRG0, ADTRG1:	A/D trigger input
AFO:	Auxiliary frequency output
ANI00 to ANI09, ANI10 to ANI19:	Analog input
AV_{DD}:	Analog power supply
AV_{REF0}, AV_{REF1}:	Analog reference voltage
AV_{SS0}, AV_{SS1}:	Analog ground
$\overline{BEN0}$ to $\overline{BEN3}$:	Byte enable
$\overline{CS0}$, $\overline{CS1}$, $\overline{CS3}$, $\overline{CS4}$:	Chip select
CV_{DD}:	Power supply for oscillator
CV_{SS}:	Oscillator ground
D0 to D31:	Data bus
DCK:	Debug clock input
DDI:	Debug data input
DDO:	Debug data output
DMS:	Debug mode select
$\overline{DRST}$:	Debug reset
ESO0, ESO1:	Emergency shut-off
FCRXD0, FCRXD1:	FCAN receive data input
FCTXD0, FCTXD1:	FCAN transmit data output
INTP0 to INTP12:	External interrupt request
MODE0 to MODE2:	Mode
NMI:	Non-maskable interrupt request
NC:	Not connected
P00 to P04:	Port 0
P10 to P17:	Port 1
P20 to P27:	Port 2
P30 to P37:	Port 3
P40 to P45:	Port 4
P50 to P57:	Port 5
P60 to P67:	Port 6
P70 to P75:	Port 7
P80 to P86:	Port 8
P90 to P96:	Port 9
P100 to P102:	Port 10
PAL0 to PAL15:	Port AL
PAH0 to PAH5:	Port AH
PCD2 to PCD5:	Port CD
PCM0, PCM1, PCM6, PCM7:	Port CM
PCS0, PCS1, PCS3, PCS4:	Port CS
PCT4, PCT5:	Port CT
PDL0 to PDL15:	Port DH
PDH0 to PDH15:	Port DL
$\overline{RD}$:	Read strobe
$\overline{RESET}$:	Reset
$\overline{SCK30}$, $\overline{SCK31}$, $\overline{SCKB0}$, $\overline{SCKB1}$:	Serial clock
SCS300 to SCS303, SCS310 to SCS313:	Serial chip select
SI30, SI31, SIB0, SIB1:	Serial data input
SO30, SO31, SOB0, SOB1:	Serial data output
SSB0, SSB1:	Serial slave select input
TCLR1:	Timer clear
TCUD1:	Timer control pulse input
TECRT0, TECRT1:	Timer external clear
TENCT00, TENCT01, TENCT10, TENCT11:	Timer encoder input
TEVTP0 to TEVTP7, TEVTR1, TEVTT0, TEVTT1:	Timer event input
TICC10, TICC11 TIP00, TIP01, TIP10, TIP11, TIP20, TIP21, TIP30, TIP31, TIP40, TIP41, TIP50, TIP51, TIP60, TIP61, TIP70, TIP71, TIR10 to TIR13, TIT00, TIT01, TIT10, TIT11:	Timer input
TIUD1:	Timer count pulse input
TO1, TOP00, TOP01, TOP10, TOP11, TOP20, TOP21, TOP30, TOP31, TOP40, TOP41, TOP50, TOP51, TOP60, TOP61, TOP70, TOP71, TOP81, TOR00 to TOR07, TOR10 to TOR17, TOT00, TOT01, TOT10, TOT11:	Timer output
TTRGP0 to TTRGP7, TTRGR1, TTRGT0, TTRGT1:	Timer trigger input
TXDC0, TXDC1:	Transmit data output
V_{DD10} to V_{DD15}:	Power supply for CPU
V_{DD30} to V_{DD37}:	I/O buffers power supply
V_{SS10} to V_{SS15}:	CPU Ground
V_{SS30} to V_{SS37}:	I/O buffers ground
$\overline{WAIT}$:	Wait
$\overline{WR}$:	Write strobe
X1, X2:	Crystal

1.5 Function Blocks

1.5.1 Internal block diagrams

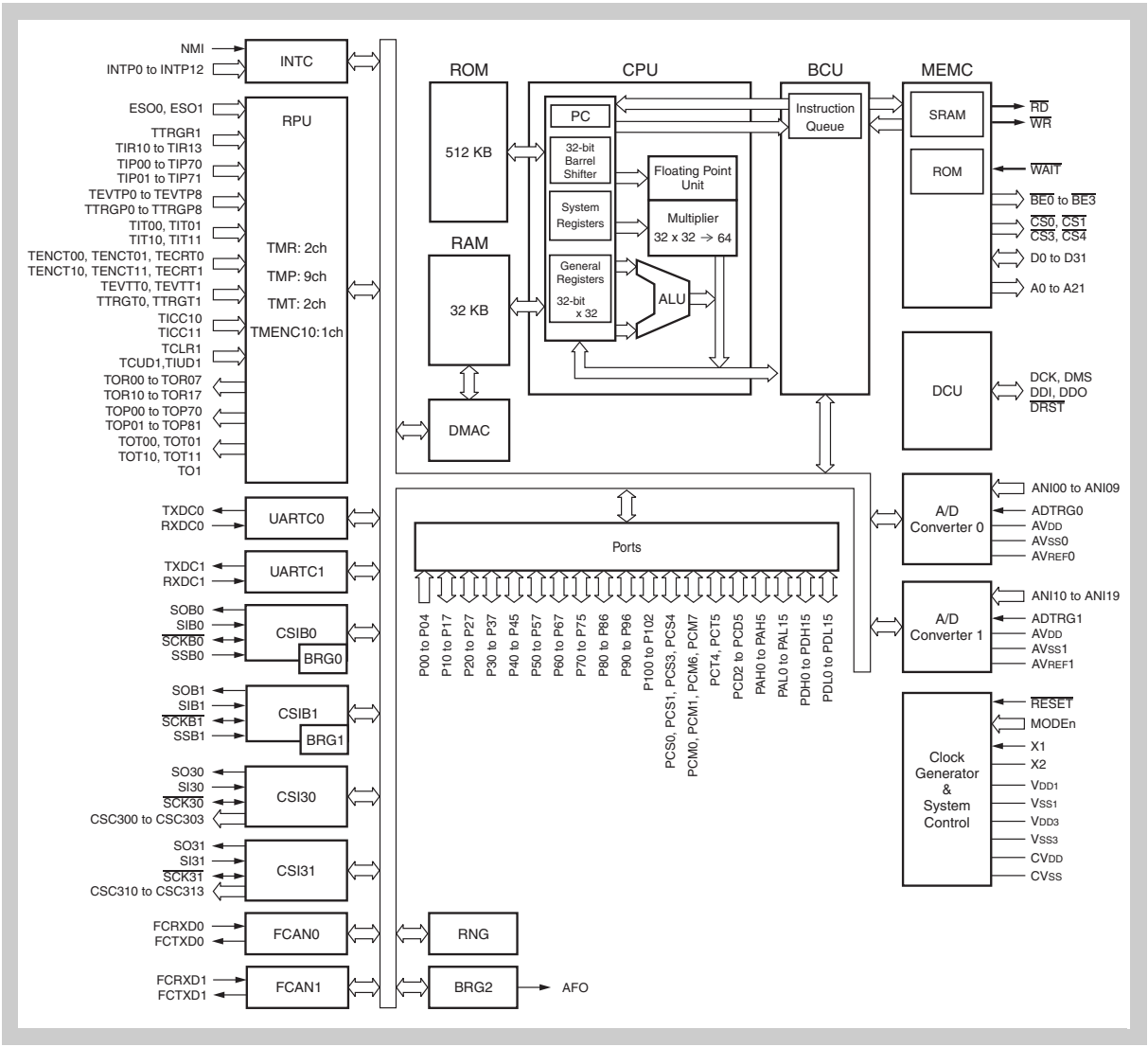


Figure 1-2 Internal Block Diagram of µPD70F3187

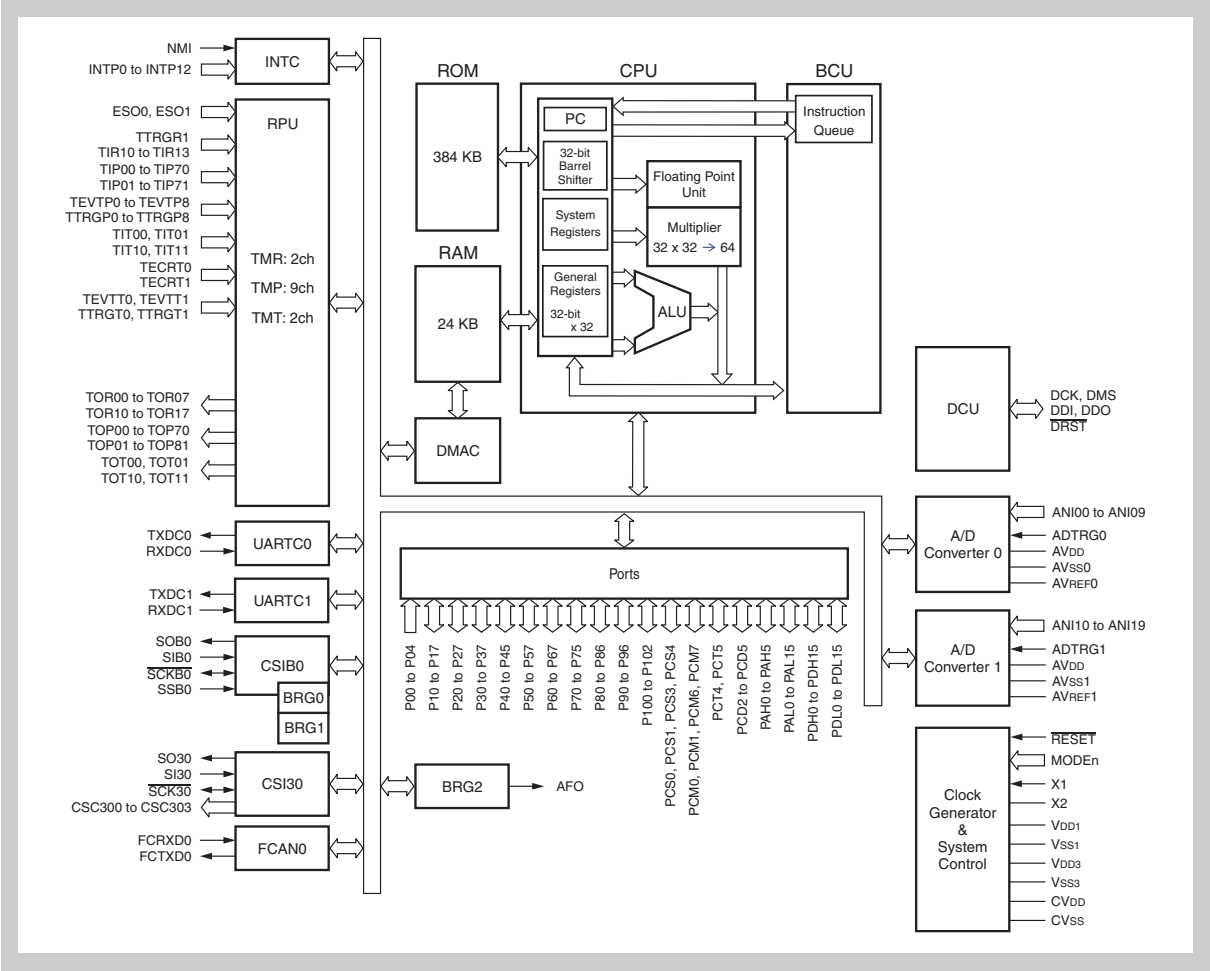


Figure 1-3 Internal Block Diagram of µPD70F3447

1.5.2 On-chip units

(1) CPU

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

Other dedicated on-chip hardware, such as a multiplier (16 bits × 16 bits → 32 bits or 32 bits × 32 bits → 64 bits) and a barrel shifter (32 bits), help accelerate processing of complex instructions.

(2) Bus control unit (BCU)

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

The BCU controls a memory controller (MEMC) and DMA controller (DMAC) and performs external memory access and DMA transfer.

(a) Memory controller (MEMC)[Note2]

The MEMC controls SRAM, ROM, and various I/O for external memory expansion.

– SRAM, external ROM, external I/O interface

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

(b) DMA controller (DMAC)

The DMAC performs data transfers b/w internal on-chip RAM and peripheral I/O. For this purpose eight DMA channels are provided for particular transfer functions of serial I/O interfaces, real-time pulse unit (TMR), and A/D converter.

(3) ROM

There is on-chip flash memory of 512 KB provided in the μPD70F3187, and 384 KB in the μPD70F3447.

On an instruction fetch, the ROM can be accessed by the CPU in one clock.

When single-chip mode 0 or flash memory programming mode is set, ROM is mapped starting from address 00000000_H.

When single-chip mode 1[Note2] is set, it is mapped starting from address 00100000_H.

ROM cannot be accessed if ROM-less mode[Note2] is set.

(4) RAM

There is on-chip RAM of 32 KB provided in the μPD70F3187, and 24 KB in the μPD70F3447. On-chip RAM is mapped starting from address $03FF0000_H$ for both, μPD70F3187 and μPD70F3447.

It can be accessed by the CPU in one clock on an instruction fetch or data access.

(5) Interrupt controller (INTC)

The INTC services hardware interrupt requests from on-chip peripheral I/O and external sources (NMI, INTP0 to INTP12). Eight levels of interrupt priorities can be specified for these interrupt requests, and multiple-interrupt servicing control can be performed for interrupt sources

(6) Clock generator (CG)

The CG provides a frequency that is 4 times the input clock (f_X) (using the on-chip PLL) as the internal system clock (f_{CPU}). As the input clock, connect an external crystal or resonator to pins X1 and X2 or input an external clock from the X1 pin.

(7) Real-time pulse unit (RPU)

The RPU incorporates a 2-channel 16-bit timer (TMR) for 3/6-phase sine wave PWM inverter control, an 1-channel 16-bit up/down counter (TMENC10), µPD70F3187 only and a 2-channel 16-bit up/down counter (TMT) that can be used for 2-phase encoder input or as a general-purpose timer, a 9-channel 16-bit general-purpose timer unit (TMP).

The RPU can measure pulse interval or frequency and can output programmable pulses.

(8) Serial interface (SIO)

The serial interfaces consist of 2 channels asynchronous serial interface C (UARTC), up to 2 channels clocked serial interface B (CSIB), up to 2 channels clocked serial interface 3 (CSI3), and up to 2 channels FCAN interface (AFCAN).

The UARTC performs data transfer using pins TXDCn and RXDCn (n = 0, 1).

The CSIB performs data transfer using pins SOBn, SIBn, $\overline{\text{SCKBn}}$, SSIn, and SSOn**Note1**.

The CSI3 performs data transfer using pins SO3n, SI3n, $\overline{\text{SCK3n}}$, SCS3n0 to SCS3**Note1**.

The AFCAN performs data transfer using pins FCTXDn and FCRXDn**Note1**.

(9) Baud rate generator (BRG)

The baud rate generator comprises 3 channels of 8-bit counters and comparators that can be used for clock supply of serial interfaces (CSIB), auxiliary frequency output (AFO) or interval timer.

(10) A/D converter (ADC)

The two units of high-speed, high-resolution 10-bit A/D converter include 10 analog input pins for each unit. Conversion is performed using the successive approximation method.

(11) Random number generator (RNG)

For encryption purpose a random number generator is provided.

(12) Debug control unit (DCU)

On-chip debugging can be performed via a debug control unit (n-wire interface).

Note 1. n = 0, 1 for μPD70F3187
n = 0 for μPD70F3447

2. Not available on μPD70F3447

(13) Ports

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

Table 1-1:

Port	I/O	Control Function	
		μPD70F3187	μPD70F3447
Port 0	5-bit input	NMI input, external interrupt input, A/D converter external trigger input, emergency shut-off input	
Port 1	8-bit I/O	Real-time pulse unit I/O	
Port 2	8-bit I/O	Real-time pulse unit I/O	
Port 3	8-bit I/O	Serial interface I/O, external interrupt input	
Port 4	6-bit I/O	Serial interface I/O	
Port 5	8-bit I/O	Real-time pulse unit I/O	
Port 6	8-bit I/O	Real-time pulse unit I/O	
Port 7	6-bit I/O	Real-time pulse unit I/O, external interrupt input	
Port 8	7-bit I/O	Serial interface I/O, external interrupt input	
Port 9	7-bit I/O	Serial interface I/O, external interrupt input	External interrupt input
Port 10	3-bit I/O	Real-time pulse unit I/O	
Port AL	16-bit I/O	External address bus	None
Port AH	6-bit I/O	External address bus	None
Port DL	16-bit I/O	External data bus	None
Port DH	16-bit I/O	External data bus	None
Port CD	4-bit I/O	External bus interface control signal output	None
Port CM	4-bit I/O	Wait insertion signal input	None
Port CS	4-bit I/O	External bus interface control signal output	None
Port CT	2-bit I/O	External bus interface control signal output	None

Chapter 2 Pin Functions

2.1 List of Pin Functions

The names and functions of the V850E/PH2 microcontroller pins are listed below. These pins can be divided into port pins and non-port pins according to their functions.

(1) Port pins

Table 2-1 List of port pins (1/5)

Pin Name	I/O	Function	Alternate Function	
			µPD70F3187	µPD70F3447
P00	I	Port 0 5-bit input-only port	NMI	
P01			INTP0, ESO0	
P02			INTP1, ESO1	
P03			INTP2, ADTRG0	
P04			INTP3, ADTRG1	
P10	I/O	Port 1 8-bit I/O port Input or output direction can be specified in 1-bit units	TIP00, TEVTP1, TOP00	
P11			TIP01, TTRGP1, TOP01	
P12			TIP10, TTRGP0, TOP10	
P13			TIP11, TEVTP0, TOP11	
P14			TIP20, TEVTP3, TOP20	
P15			TIP21, TTRGP3, TOP21	
P16			TIP30, TTRGP2, TOP30	
P17			TIP31, TEVTP2, TOP31	
P20	I/O	Port 2 8-bit I/O port Input or output direction can be specified in 1-bit units	TIP40, TEVTP5, TOP40	
P21			TIP41, TTRGP5, TOP41	
P22			TIP50, TTRGP4, TOP50	
P23			TIP51, TEVTP4, TOP51	
P24			TIP60, TEVTP7, TOP60	
P25			TIP61, TTRGP7, TOP61	
P26			TIP70, TTRGP6, TOP70	
P27			TIP71, TEVTP6, TOP71	
P30	I/O	Port 3 8-bit I/O port Input or output direction can be specified in 1-bit units	RXDC0, INTP4	
P31			TXDC0	
P32			RXDC1, INTP5	
P33			TXDC1	
P34			FCRXD0	
P35			FCTXD0	
P36			FCRXD1	–
P37			FCTXD1	–

Table 2-1 List of port pins (2/5)

Pin Name	I/O	Function	Alternate Function	
			µPD70F3187	µPD70F3447
P40	I/O	Port 4 6-bit I/O port Input or output direction can be specified in 1-bit units	SIB0	
P41			SOB0	
P42			$\overline{\text{SCKB0}}$	
P43			SIB1	–
P44			SOB1	–
P45			$\overline{\text{SCKB1}}$	–
P50	I/O	Port 5 8-bit I/O port Input or output direction can be specified in 1-bit units	TOR00	
P51			TOR01	
P52			TOR02	
P53			TOR03	
P54			TOR04	
P55			TOR05	
P56			TOR06	
P57			TOR07	
P60	I/O	Port 6 8-bit I/O port Input or output direction can be specified in 1-bit units	TOR10, TTRGR1	
P61			TOR11, TIR10	
P62			TOR12, TIR11	
P63			TOR13, TIR12	
P64			TOR14, TIR13	
P65			TOR15	
P66			TOR16	
P67			TOR17, TEVTR1	
P70	I/O	Port 7 6-bit I/O port Input or output direction can be specified in 1-bit units	TIT00, TEVTT1, TOT00, TENCT00	
P71			TIT01, TTRGT1, TOT01, TENCT01	
P72			TECRT0, INTP12	
P73			TIT10, TTRGT0, TOT10, TENCT10	
P74			TIT11, TEVTT0, TOT11, TENCT11	
P75			TECRT1, AFO	
P80	I/O	Port 8 7-bit I/O port Input or output direction can be specified in 1-bit units	SI30	
P81			SO30	
P82			$\overline{\text{SCK30}}$	
P83			SCS300, INTP6	
P84			SCS301, INTP7	
P85			SCS302, INTP8	
P86			SCS303, $\overline{\text{SSB0}}$	

Table 2-1 List of port pins (3/5)

Pin Name	I/O	Function	Alternate Function	
			µPD70F3187	µPD70F3447
P90	I/O	Port 9 7-bit I/O port Input or output direction can be specified in 1-bit units	SI31	–
P91			SO31	–
P92			$\overline{SCK31}$	–
P93			SCS310, INTP9	INTP9
P94			SCS311, INTP10	INTP10
P95			SCS312, INTP11	INTP11
P96			SCS313, $\overline{SSB1}$	–
P100	I/O	Port 10 3-bit I/O port Input or output direction can be specified in 1-bit units	TCLR1, TICC10, TOP81	TOP81
P101			TCUD1, TICC11	–
P102			TIUD1, TO1	–
PAL0	I/O	Port AL 16-bit I/O port Input or output direction can be specified in 1-bit units	A0	–
PAL1			A1	
PAL2			A2	
PAL3			A3	
PAL4			A4	
PAL5			A5	
PAL6			A6	
PAL7			A7	
PAL8			A8	
PAL9			A9	
PAL10			A10	
PAL11			A11	
PAL12			A12	
PAL13			A13	
PAL14			A14	
PAL15			A15	
PAH0	I/O	Port AH 6-bit I/O port Input or output direction can be specified in 1-bit units	A16	–
PAH1			A17	
PAH2			A18	
PAH3			A19	
PAH4			A20	
PAH5			A21	

Table 2-1 List of port pins (4/5)

Pin Name	I/O	Function	Alternate Function	
			µPD70F3187	µPD70F3447
PDL0	I/O	Port DL 16-bit I/O port Input or output direction can be specified in 1-bit units	D0	–
PDL1			D1	
PDL2			D2	
PDL3			D3	
PDL4			D4	
PDL5			D5	
PDL6			D6	
PDL7			D7	
PDL8			D8	
PDL9			D9	
PDL10			D10	
PDL11			D11	
PDL12			D12	
PDL13			D13	
PDL14			D14	
PDL15			D15	
PDH0	I/O	Port DH 16-bit I/O port Input or output direction can be specified in 1-bit units	D16	–
PDH1			D17	
PDH2			D18	
PDH3			D19	
PDH4			D20	
PDH5			D21	
PDH6			D22	
PDH7			D23	
PDH8			D24	
PDH9			D25	
PDH10			D26	
PDH11			D27	
PDH12			D28	
PDH13			D29	
PDH14			D30	
PDH15			D31	
PCD2	I/O	Port CD 4-bit I/O port Input or output direction can be specified in 1-bit units	$\overline{\text{BEN0}}$	–
PCD3			$\overline{\text{BEN1}}$	
PCD4			$\overline{\text{BEN2}}$	
PCD5			$\overline{\text{BEN3}}$	
PCM0	I/O	Port CM 4-bit I/O port Input or output direction can be specified in 1-bit units	$\overline{\text{WAIT}}$	–
PCM1			–	
PCM6			–	
PCM7			–	

Table 2-1 List of port pins (5/5)

Pin Name	I/O	Function	Alternate Function	
			µPD70F3187	µPD70F3447
PCS0	I/O	Port CS 4-bit I/O port Input or output direction can be specified in 1-bit units	$\overline{CS0}$	–
PCS1			$\overline{CS1}$	
PCS3			$\overline{CS3}$	
PCS4			$\overline{CS4}$	
PCT4	I/O	Port CT 2-bit I/O port Input or output direction can be specified in 1-bit units	$\overline{RD}$	–
PCT5			$\overline{WR}$	–

(2) Non-port pins

Table 2-2 List of non-port pins (1/6)

Pin Name	I/O	Function	Alternate Function	
			µPD70F3187	µPD70F3447
A0 to A15	O	22-bit external address bus[a]	PAL0 to PAL15	
A16 to A21			PAH0 to PAH5	
ADTRG0	I	A/D conversion start trigger (ADC0)	P03, INTP2	
ADTRG1	I	A/D conversion start trigger (ADC1)	P04, INTP3	
AFO	O	Auxiliary frequency output	P75, TECRT1	
ANI00 to ANI09	I	Analog input channels (ADC0)	–	
ANI10 to ANI19	I	Analog input channels (ADC1)	–	
AV_{DD}	–	Positive power supply (3.3 V) (ADC0, ADC1)	–	
AV_{REF0}	I	Reference voltage input (ADC0)	–	
AV_{REF1}	I	Reference voltage input (ADC1)	–	
AV_{SS0}	–	Power supply ground (ADC0)	–	
AV_{SS1}	–	Power supply ground (ADC1)	–	
$\overline{BEN0}$	O	External byte enable output[a]	PCD2	
$\overline{BEN1}$			PCD3	
$\overline{BEN2}$			PCD4	
$\overline{BEN3}$			PCD5	
$\overline{CS0}$	O	Chip select signal output[a]	PCS0	
$\overline{CS1}$			PCS1	
$\overline{CS3}$			PCS3	
$\overline{CS4}$			PCS4	
CV_{DD}	–	Oscillator power supply (1.5 V)	–	
CV_{SS}	–	Oscillator power supply ground	–	

Table 2-2 List of non-port pins (2/6)

Pin Name	I/O	Function	Alternate Function	
			µPD70F3187	µPD70F3447
D0 to D15	I/O	32-bit external data bus[a]	PDL0 to PDL15	
D16 to D31			PDH0 to PDH15	
DCK	I	N-wire interface clock	–	
DDI	I	N-wire data input and reset mode selection	–	
DDO	O	N-wire data output	–	
DMS	I	N-wire mode select	–	
$\overline{\text{DRST}}$	I	N-wire interface reset	–	
ESO0	I	Emergency shut off input (TMR0)	INTP0, P01	
ESO1	I	Emergency shut off input (TMR1)	INTP1, P02	
FCRXD0	I	Receive input (AFCAN0)	P34	
FCRXD1	I	Receive input (AFCAN1)[a]	P36	
FCTXD0	O	Transmit output (AFCAN0)	P35	
FCTXD1	O	Transmit output (AFCAN1)[a]	P37	
FLMD0	I	Flash programming mode selection	MODE0	
FLMD1			MODE1	
INTP0	I	External maskable interrupt request input	P01, ESO0	
INTP1			P02, ESO1	
INTP2			P03, ADTRG0	
INTP3			P04, ADTRG1	
INTP4			P30, RXDC0	
INTP5			P32, RXDC1	P32
INTP6			P83, SCS300	
INTP7			P84, SCS301	
INTP8			P85, SCS302	
INTP9			P93, SCS310	P93
INTP10			P94, SCS311	P94
INTP11			P95, SCS312	P95
INTP12			P72, TECRT0	
MODE0	I	Device operating mode selection	FLMD0	
MODE1			FLMD1	
MODE2			–	
NMI	I	Non-maskable interrupt request input	P00	
$\overline{\text{RD}}$	O	Read strobe signal output[a]	PCT4	
$\overline{\text{RESET}}$	I	System reset input	–	
RXDC0	I	Receive input (UARTC0)	P30, INTP4	
RXDC1	I	Receive input (UARTC1)	P32, INTP5	
$\overline{\text{SCK30}}$	I/O	Serial shift clock I/O (CSI30)	P82	
$\overline{\text{SCK31}}$	I/O	Serial shift clock I/O (CSI31)[a]	P92	
$\overline{\text{SCKB0}}$	I/O	Serial shift clock I/O (CSIB0)	P42	
$\overline{\text{SCKB1}}$	I/O	Serial shift clock I/O (CSIB1)[a]	P45	

Table 2-2 List of non-port pins (3/6)

Pin Name	I/O	Function	Alternate Function	
			µPD70F3187	µPD70F3447
SCS300	O	Serial peripheral chip select (CSI30)	P83, INTP7	
SCS301			P84, INTP8	
SCS302			P85, INTP9	
SCS303			P86, $\overline{SSB0}$	
SCS310	O	Serial peripheral chip select (CSI31)[a]	P93, INTP10	
SCS311			P94, INTP10	
SCS312			P95, INTP11	
SCS313			P96, $\overline{SSB1}$	P96
SI30	I	Serial data input (CSI30)	P80	
SI31	I	Serial data input (CSI31)[a]	P90	
SIB0	I	Serial data input (CSIB0)	P40	
SIB1	I	Serial data input (CSIB1)[a]	P43	
SO30	O	Serial data output (CSI30)	P81	
SO31	O	Serial data output (CSI31)	P91	
SOB0	O	Serial data output (CSIB0)	P41	
SOB1	O	Serial data output (CSIB1)	P44	
$\overline{SSB0}$	I	Serial slave select input (CSIB0)	P86, SCS303	
$\overline{SSB1}$	I	Serial slave select input (CSIB1)[a]	P96, SCS313	P96
TCLR1	I	Timer clear input (TMENC10)[a]	P100, TICC10, TOP81	P100, TOP81
TCUD1	I	Count up/down direction control input (TMENC10)[a]	P101, TICC11	P101
TECRT0	I	Timer clear input (TMT0)	P72, INTP12	
TECRT1	I	Timer clear input (TMT1)	P75, AFO	
TENCT00	I	Timer encoder input (TMT0)	P70, TIT00, TEVTT1, TOT00	
TENCT01	I		P71, TIT01, TTRGT1, TOT01	
TENCT10	I	Timer encoder input (TMT1)	P73, TIT10, TTRGT0, TOT10	
TENCT11	I		P74, TIT11, TEVTT0, TOT11	
TEVTP0	I	Timer event input (TMP0)	P13, TIP11, TOP11	
TEVTP1	I	Timer event input (TMP1)	P10, TIP00, TOP00	
TEVTP2	I	Timer event input (TMP2)	P17, TIP31, TOP31	
TEVTP3	I	Timer event input (TMP3)	P14, TIP20, TOP20	
TEVTP4	I	Timer event input (TMP4)	P23, TIP51, TOP51	
TEVTP5	I	Timer event input (TMP5)	P20, TIP40, TOP40	
TEVTP6	I	Timer event input (TMP6)	P27, TIP71, TOP71	
TEVTP7	I	Timer event input (TMP7)	P24, TIP60, TOP60	
TEVTR1	I	Timer event input (TMR1)	P67, TOR17	
TEVTT0	I	Timer event input (TMT0)	P74,TIT11, TOT11, TENCT11	
TEVTT1	I	Timer event input (TMT1)	P70, TIT00, TOT00, TENCT00	

Table 2-2 List of non-port pins (4/6)

Pin Name	I/O	Function	Alternate Function	
			µPD70F3187	µPD70F3447
TICC10	I	TMENC10 capture trigger input[a]	P100, TCLR1, TOP81	P100, TOP81
TICC11			P101, TCUD1	P101
TIP00	I	Capture trigger input (TMP0)	P10, TEVTP1, TOP00	
TIP01			P11, TTRGP1, TOP01	
TIP10	I	Capture trigger input (TMP1)	P12, TTRGP0, TOP10	
TIP11			P13, TEVTP0, TOP11	
TIP20	I	Capture trigger input (TMP2)	P14, TEVTP3, TOP20	
TIP21			P15, TTRGP3, TOP21	
TIP30	I	Capture trigger input (TMP3)	P16, TTRGP2, TOP30	
TIP31			P17, TEVTP2, TOP31	
TIP40	I	Capture trigger input (TMP4)	P20, TEVTP5, TOP40	
TIP41			P21, TTRGP5, TOP41	
TIP50	I	Capture trigger input (TMP5)	P22, TTRGP4, TOP50	
TIP51			P23, TEVTP4, TOP51	
TIP60	I	Capture trigger input (TMP6)	P24, TEVTP7, TOP60	
TIP61			P25, TTRGP7, TOP61	
TIP70	I	Capture trigger input (TMP7)	P26, TTRGP6, TOP70	
TIP71			P27, TEVTP6, TOP71	
TIR10	I	Capture trigger input (TMR1)	P61, TOR11	
TIR11			P62, TOR12	
TIR12			P63, TOR13	
TIR13			P64, TOR14	
TIT00	I	Capture trigger input (TMT0)	P70, TEVTT1, TOT00, TENCT00	
TIT01			P71, TTRGT1, TOT01, TENCT01	
TIT10	I	Capture trigger input (TMT0)	P73, TTRGT0, TOT10, TENCT10	
TIT11			P74,TEVTT0, TOT11, TENCT11	
TIUD1	I	External count clock input (TMENC10)[a]	P102, TO1	P102
TO1	O	Pulse signal output (TMENC10)[a]	P102, TIUD1	P102
TOP00	O	Pulse signal output (TMP0)	P10, TIP00, TEVTP1	
TOP01			P11, TIP01, TTRGP1	
TOP10	O	Pulse signal output (TMP1)	P12, TIP10, TTRGP0	
TOP11			P13, TIP11, TEVTP0	
TOP20	O	Pulse signal output (TMP2)	P14, TIP20, TEVTP3	
TOP21			P15, TIP21, TTRGP3	
TOP30	O	Pulse signal output (TMP3)	P16, TIP30, TTRGP2	
TOP31			P17, TIP31, TEVTP2	
TOP40	O	Pulse signal output (TMP4)	P20, TIP40, TEVTP5	
TOP41			P21, TIP41, TTRGP5	
TOP50	O	Pulse signal output (TMP5)	P22, TIP50, TTRGP4	
TOP51			P23, TIP51, TEVTP4	

Table 2-2 List of non-port pins (5/6)

Pin Name	I/O	Function	Alternate Function	
			µPD70F3187	µPD70F3447
TOP60	O	Pulse signal output (TMP6)	P24, TIP60, TEVTP7	
TOP61			P25, TIP61, TTRGP7	
TOP70	O	Pulse signal output (TMP7)	P26, TIP70, TTRGP6	
TOP71			P27, TIP71, TEVTP6	
TOP81	O	Pulse signal output (TMP8)	P100, TCLR1, TICC10	P100
TOR00	O	Pulse signal output (TMR0)	P50	
TOR01			P51	
TOR02			P52	
TOR03			P53	
TOR04			P54	
TOR05			P55	
TOR06			P56	
TOR07			P57	
TOR10	O	Pulse signal output (TMR1)	P60, TTRGR1	
TOR11			P61, TIR10	
TOR12			P62, TIR11	
TOR13			P63, TIR12	
TOR14			P64, TIR13	
TOR15			P65	
TOR16			P66	
TOR17			P67, TEVTR1	
TOT00	O	Pulse signal output (TMT0)	P70, TIT00, TEVTT1, TENCT00	
TOT01			P71, TIT01, TTRGT1, TENCT01	
TOT10	O	Pulse signal output (TMT1)	P73, TIT10, TTRGT0, TENCT10	
TOT11			P74,TIT11, TEVTT0, TENCT11	
TTRGP0	I	Timer trigger input (TMP0)	P12, TIP10, TOP10	
TTRGP1		Timer trigger input (TMP1)	P11, TIP01, TOP01	
TTRGP2		Timer trigger input (TMP2)	P16, TIP30, TOP30	
TTRGP3		Timer trigger input (TMP3)	P15, TIP21, TOP21	
TTRGP4		Timer trigger input (TMP4)	P22, TIP50, TOP50	
TTRGP5		Timer trigger input (TMP5)	P21, TIP41, TOP41	
TTRGP6		Timer trigger input (TMP6)	P26, TIP70, TOP70	
TTRGP7		Timer trigger input (TMP7)	P25, TIP61, TOP61	
TTRGR1	I	Timer trigger input (TMR1)	P60, TOR10	
TTRGT0	I	Timer trigger input (TMT0)	P73, TIT10, TOT10, TENCT10	
TTRGT1	I	Timer trigger input (TMT1)	P71, TIT01, TOT01, TENCT01	
TXDC0	O	Transmit output (UARTC0)	P31	
TXDC1	O	Transmit output (UARTC1)	P33	

Table 2-2 List of non-port pins (6/6)

Pin Name	I/O	Function	Alternate Function	
			µPD70F3187	µPD70F3447
V_{DD10} to V_{DD15}	–	Positive power supply for internal CPU (1.5 V)	–	
V_{DD30} to V_{DD37}	–	Positive power supply for peripheral interface (3.3 V)	–	
V_{SS10} to V_{SS15}	–	Power supply ground for internal CPU	–	
V_{SS30} to V_{SS37}	–	Power supply ground for peripheral interface	–	
$\overline{WAIT}$	I	External wait control signal input[a]	PCM0	
$\overline{WR}$	O	Write strobe signal output[a]	PCT5	
X1	I	Crystal connection	–	
X2	–		–	

a) not available on µPD70F3447

2.2 Pin Status

Table 2-3 Pin status during operating states

Pin	Operating Status				
	During reset	After reset release			HALT Mode
		Single-chip Mode 0	Single-chip Mode 1[a]	ROM-less Mode[a]	
A0 to A15 (PAL0 to PAL15)	Hi-Z	Hi-Z	Operating		Operating
A16 to A21 (PAH0 to PAH5)	Hi-Z	Hi-Z	Operating		Operating
D0 to D15 (PDL0 to PDL15)	Hi-Z	Hi-Z	Operating		Operating
D16 to D31 (PDH0 to PDH15)	Hi-Z	Hi-Z	Operating		Operating
$\overline{BEN0}$ to $\overline{BEN3}$ (PCD2 to PCD5)	Hi-Z	Hi-Z	Operating		Operating
$\overline{CS0}$ (PCS0)	Hi-Z	Hi-Z	Operating		Operating
$\overline{CS1}$ (PCS1)	Hi-Z	Hi-Z	Operating		Operating
$\overline{CS3}$ (PCS3)	Hi-Z	Hi-Z	Operating		Operating
$\overline{CS4}$ (PCS4)	Hi-Z	Hi-Z	Operating		Operating
$\overline{RD}$ (PCT4)	Hi-Z	Hi-Z	Operating		Operating
$\overline{WR}$ (PCT5)	Hi-Z	Hi-Z	Operating		Operating
$\overline{WAIT}$ (PCM0)	Hi-Z	Hi-Z	Operating		Operating
PCM1, PCM6, PCM7	Hi-Z	Hi-Z	Hi-Z		Operating
DCK	Operating	Operating	Operating		Operating
DDI	Operating	Operating	Operating		Operating
DDO	Operating	Operating	Operating		Operating
DMS	Operating	Operating	Operating		Operating
$\overline{DRST}$	Operating	Operating	Operating		Operating
INTP0 to INTP3 (P01 to P04)	–	Input	Input		Operating
INTP4 (P30)	–	Input	Input		Operating
INTP5 (P32)	–	Input	Input		Operating
INTP6 to INTP8 (P83 to P85)	–	Input	Input		Operating
INTP9 to INTP11 (P93 to P95)	–	Input	Input		Operating
NMI (P00)	–	Input	Input		Operating
Peripheral input pin other than above	Hi-Z	Hi-Z	Hi-Z		Operating
Peripheral output pin other than above	×	×	×		Operating
Port input pin other than above	Hi-Z	Hi-Z	Hi-Z		–
Port output pin other than above	×	×	×		Hold

a) not available on μPD70F3447

Remark Hi-Z: High Impedance
–: Input data is not sampled
×: No function selected at reset

2.3 Description of Pin Functions

2.3.1 P00 to P04 (Port 0) … Input

Port 0 is an 8-bit input-only port in which all pins are fixed for input.

Besides functioning as a port, in control mode, P00 to P04 operate as NMI input, external interrupt request signal, real-time pulse unit (RPU) emergency shut off signal input, and A/D converter (ADC) external trigger input. Normally, if function pins also serve as ports, one mode or the other is selected using a port mode control register. However, there is no such register for P00 to P04. Therefore, the input port cannot be switched with the NMI input pin, external interrupt request input pin, RPU emergency shut off signal input pin, and A/D converter (ADC) external trigger input pin. Read the status of each pin by reading the port.

(1) Port mode

P00 to P04 are input-only.

(2) Control mode

P00 to P04 also serve as NMI, INTP0 to INTP3, ESO0, ESO1, ADTRG0, and ADTRG1 pins, but the control function cannot be disabled.

(a) NMI (Non-maskable interrupt request) … Input

This is non-maskable interrupt request input.

(b) INTP0 to INTP3 (Interrupt request from peripherals) … Input

These are external interrupt request input pins.

(c) ESO0, ESO1 (Emergency shut off) … Input

These pins input timer TMR0 and timer TMR1 emergency shut off signals.

(d) ADTRG0, ADTRG1 (A/D trigger input) … Input

These are A/D converter external trigger input pins.

2.3.2 P10 to P17 (Port 1) … Input/Output

Port 1 is an 8-bit I/O port in which input or output can be set for each port pin individually.

Besides functioning as an I/O port, in control mode, P10 to P17 operate as RPU input or output.

The operation mode can be specified by the port 1 mode control register (PMC1) to port or control mode for each port pin individually.

(1) Port mode

P10 to P17 can be set to input or output in 1-bit units using the port 1 mode register (PM1).

(2) Control mode

P10 to P17 can be set to port or control mode in 1-bit units using the PMC1 register.

(a) TIP00, TIP01, TIP10, TIP11, TIP20, TIP21, TIP30, TIP31 (Timer capture input) … Input

These are timer TMP0 to TMP3 capture trigger input pins.

(b) TEVTP0, TEVTP1, TEVTP2, TEVTP3 (Timer event input) … Input

These are timer TMP0 to TMP3 external event counter input pins.

(c) TTRGP0, TTRGP1, TTRGP2, TTRGP3 (Timer trigger) … Input

These are timer TMP0 to TMP3 external trigger input pins.

(d) TOP00, TOP01, TOP10, TOP11, TOP20, TOP21, TOP30, TOP31 (Timer output) … Output

These pins output timer TMP0 to TMP3 pulse signals.

2.3.3 P20 to P27 (Port 2) … Input/Output

Port 2 is an 8-bit I/O port in which input or output can be set for each port pin individually.

Besides functioning as an I/O port, in control mode, P20 to P27 operate as RPU input or output.

The operation mode can be specified by the port 2 mode control register (PMC2) to port or control mode for each port pin individually.

(1) Port mode

P20 to P27 can be set to input or output in 1-bit units using the port 2 mode register (PM2).

(2) Control mode

P20 to P27 can be set to port or control mode in 1-bit units using the PMC2 register.

(a) TIP40, TIP41, TIP50, TIP51, TIP60, TIP61, TIP70, TIP71 (Timer capture input) … Input

These are timer TMP4 to TMP7 capture trigger input pins.

(b) TEVTP4, TEVTP5, TEVTP6, TEVTP7 (Timer event input) … Input

These are timer TMP4 to TMP7 external event counter input pins.

(c) TTRGP4, TTRGP5, TTRGP6, TTRGP7 (Timer trigger) … Input

These are timer TMP4 to TMP7 external trigger input pins.

(d) TOP40, TOP41, TOP50, TOP51, TOP60, TOP61, TOP70, TOIP71 (Timer output) … Output

These pins output timer TMP4 to TMP7 pulse signals.

2.3.4 P30 to P37 (Port 3) … Input/Output

Port 3 is an 8-bit I/O port in which input or output can be set for each port pin individually.

Besides functioning as an I/O port, in control mode, P30 to P37 operate as serial interface (UARTC0, UARTC1, AFCAN0, AFCAN1[Note]). Additionally external interrupt request signal inputs are available in port input mode.

The operation mode can be specified by the port 3 mode control register (PMC3) to port or control mode for each port pin individually.

(1) Port mode

P30 to P37 can be set to input or output in 1-bit units using the port 3 mode register (PM3).

(a) INTP4, INTP5 (Interrupt request from peripherals) … Input

These are external interrupt request input pins, which are simultaneously enabled in port input mode.

(2) Control mode

P30 to P37 can be set to port or control mode in 1-bit units using the PMC3 register.

(a) TXDC0, TXDC1 (Transmit data) … Output

These pins output serial transmit data of UARTC0 and UARTC1.

(b) RXDC0, RXDC1 (Receive data) … Input

These pins input serial receive data of UARTC0 and UARTC1.

(c) FCTXD0, FCTXD1[Note] (Transmit data for controller area network) … Output

These pins output AFCAN0 and AFCAN1[Note] serial transmit data.

(d) FCRXD 0, FCRXD1[Note] (Receive data for controller area network) … Input

These pins input AFCAN0 and AFCAN1[Note] serial receive data.

Note not available on µPD70F3447

2.3.5 P40 to P45 (Port 10) … Input/Output

Port 4 is a 6-bit I/O port in which input or output can be set for each port pin individually.

Besides functioning as an I/O port, in control mode, P40 to P45 operate as serial interface (CSIB0, CSIB1[Note]).

The operation mode can be specified by the port 4 mode control register (PMC4) to port or control mode for each port pin individually.

(1) Port mode

P40 to P45 can be set to input or output in 1-bit units using the port 4 mode register (PM4).

(2) Control mode

P40 to P45 can be set to port or control mode in 1-bit units using the PMC4 register.

(a) SOB0, SOB1[Note] (Serial output) … Output

These pins output CSIB0 and CSIB1[Note] serial transmit data.

(b) SIB0, SIB1[Note] (Serial input) … Input

These pins input CSIB0 and CSIB1[Note] serial receive data.

(c) $\overline{\text{SCKB0}}$, $\overline{\text{SCKB1}}$ [Note](Serial clock) … I/O

These are the CSIB0 and CSIB1[Note] serial clock I/O pins.

Note not available on µPD70F3447

2.3.6 P50 to P57 (Port 5) … Input/Output

Port 5 is an 8-bit I/O port in which input or output can be set for each port pin individually.

Besides functioning as an I/O port, in control mode, P50 to P57 operate as RPU input or output.

The operation mode can be specified by the port 5 mode control register (PMC5) to port or control mode for each port pin individually.

(1) Port mode

P50 to P57 can be set to input or output in 1-bit units using the port 5 mode register (PM5).

(2) Control mode

P50 to P57 can be set to port or control mode in 1-bit units using the PMC5 register.

(a) TOR00, TOR01, TOR02, TOR03, TOR04 (Timer output) … Output

These pins output timer TMR0 pulse signals.

2.3.7 P60 to P67 (Port 6) … Input/Output

Port 6 is an 8-bit I/O port in which input or output can be set for each port pin individually.

Besides functioning as an I/O port, in control mode, P60 to P67 operate as RPU input or output.

The operation mode can be specified by the port 6 mode control register (PMC6) to port or control mode for each port pin individually.

(1) Port mode

P60 to P67 can be set to input or output in 1-bit units using the port 6 mode register (PM6).

(2) Control mode

P60 to P67 can be set to port or control mode in 1-bit units using the PMC6 register.

(a) TIR10, TIR11, TIR12, TIR13 (Timer capture input) … Input

These are timer TMR1 capture trigger input pins.

(b) TEVTR1 (Timer event input) … Input

This is a timer TMR1 external event counter input pin.

(c) TTRGR1 (Timer trigger) … Input

This is a timer TMR1 external trigger input pin.

(d) TOR10, TOR11, TOR12, TOR13, TOR14 (Timer output) … Output

These pins output timer TMR1 pulse signals.

2.3.8 P70 to P75 (Port 7) … Input/Output

Port 7 is a 6-bit I/O port in which input or output can be set for each port pin individually.

Besides functioning as an I/O port, in control mode, P70 to P75 operate as RPU input or output, and auxiliary frequency output. Additionally an external interrupt request signal input is available in port input mode.

The operation mode can be specified by the port 7 mode control register (PMC7) to port or control mode for each port pin individually.

(1) Port mode

P70 to P75 can be set to input or output in 1-bit units using the port 7 mode register (PM7).

(a) INTP12 (Interrupt request from peripherals) … Input

This is an external interrupt request input pin, which is simultaneously enabled in port input mode.

(2) Control mode

P70 to P75 can be set to port or control mode in 1-bit units using the PMC7 register.

(a) TIT00, TIT01, TIT10, TIT11 (Timer capture input) … Input

These are timer TMT0 and TMT1 capture trigger input pins.

(b) TEVTT0, TEVTT1 (Timer event input) … Input

These are timer TMT0 and TMT1 external event counter input pins.

(c) TTRGT0, TTRGT1 (Timer trigger) … Input

These are timer TMT0 and TMT1 external trigger input pins.

(d) TECRT0, TECRT1 (Timer clear) … Input

These are timer TMT0 and TMT1 external clear input pins.

(e) TENCT00, TENCT01, TENCT10, TENCT11 (Timer encoder input … Input

These are timer TMT0 and TMT1 encoder input pins.

(f) TOT00, TOT01, TOT10, TOT11 (Timer output) … Output

These pins output timer TMT0 and TMT1 pulse signals.

(g) AFO (Auxiliary frequency) … Output

This is an auxiliary frequency output signal pin of baudrate generator BGR2.

2.3.9 P80 to P86 (Port 8) … Input/Output

Port 8 is a 7-bit I/O port in which input or output can be set for each port pin individually.

Besides functioning as an I/O port, in control mode, P80 to P86 operate as serial interface (CSI30, CSIB0). Additionally external interrupt request signal inputs are available in port input mode.

The operation mode can be specified by the port 8 mode control register (PMC8) to port or control mode for each port pin individually.

(1) Port mode

P80 to P86 can be set to input or output in 1-bit units using the port 8 mode register (PM8).

(a) INTP6, INTP7, INTP8 (Interrupt request from peripherals) … Input

These are external interrupt request input pins, which are simultaneously enabled in port input mode.

(2) Control mode

P80 to P86 can be set to port or control mode in 1-bit units using the PMC8 register.

(a) SO30 (Serial output) … Output

This pin outputs CSI30 serial transmit data.

(b) SI30 (Serial input) … Input

This pin inputs CSI30 serial receive data.

(c) $\overline{\text{SCK30}}$ (Serial clock) … I/O

This is the CSI30 serial clock I/O pin.

(d) SCS300 to SCS303 (Serial chip select) … Output

These pins output CSI30 serial chip select signals.

(e) $\overline{\text{SSB0}}$ (Serial slave select signal) … Input

This pin inputs CSIB0 slave select signal.

2.3.10 P90 to P96 (Port 9) … Input/Output

Port 9 is a 7-bit I/O port in which input or output can be set for each port pin individually.

Besides functioning as an I/O port, in control mode, P90 to P96 operate as serial interface (CSI31[Note], CSIB1[Note]). Additionally external interrupt request signal inputs are available in port input mode.

The operation mode can be specified by the port 9 mode control register (PMC9) to port or control mode for each port pin individually.

(1) Port mode

P90 to P96 can be set to input or output in 1-bit units using the port 9 mode register (PM9).

(a) INTP9, INTP10, INTP11 (Interrupt request from peripherals) … Input

These are external interrupt request input pins, which are simultaneously enabled in port input mode.

(2) Control mode

P90 to P96 can be set to port or control mode in 1-bit units using the PMC9 register.

(a) SO31 (Serial output) … Output[Note]

This pin outputs CSI31 serial transmit data.

(b) SI31 (Serial input) … Input[Note]

This pin inputs CSI31 serial receive data.

(c) $\overline{\text{SCK31}}$ (Serial clock) … I/O[Note]

This is the CSI31 serial clock I/O pin.

(d) SCS310 to SCS313 (Serial chip select) … Output[Note]

These pins output CSI31 serial chip select signals.

(e) $\overline{\text{SSB1}}$ (Serial slave select input) … Input[Note]

This pin inputs CSIB1 slave select signal.

Note not available on µPD70F3447

2.3.11 P100 to P102 (Port 10) … Input/Output

Port 10 is a 3-bit I/O port in which input or output can be set for each port pin individually.

Besides functioning as an I/O port, in control mode, P100 to P102 operate as RPU input or output

The operation mode can be specified by the port 10 mode control register (PMC10) to port or control mode for each port pin individually.

(1) Port mode

P100 to P102 can be set to input or output in 1-bit units using the port 10 mode register (PM10).

(2) Control mode

P100 to P102 can be set to port or control mode in 1-bit units using the PMC4 register.

(a) TIUD1 (Timer count pulse input) … Input[Note]

This is an external count clock input pin to the up/down counter (TMENC10).

(b) TCUD1 (Timer control pulse input) … Input[Note]

This is an input count operation switching signal to the up/down counter (TMENC10).

(c) TCLR1 (Timer clear) … Input[Note]

This is a clear signal input pin to the up/down counter (TMENC10).

(d) TICC10, TICC11 (Timer capture input) … Input[Note]

These are timer TMENC10 external capture trigger input pins.

(e) TO1 (Timer output) … Output[Note]

This pin outputs timer TMENC10 pulse signals.

(f) TOP80 (Timer output) … Output

This pin outputs timer TMP8 pulse signals.

Note not available on µPD70F3447

2.3.12 PAL0 to PAL15 (Port AL) … I/O

Port AL is an 8-bit or a 16-bit I/O port in which input or output can be set for each port pin individually.

Besides functioning as a port, in control mode, these pins operate as the address bus (A0 to A15) when memory is expanded externally.

The operation mode can be specified by the port AL mode control register (PMCAL) to port or control mode for each port pin individually.

(1) Port mode

PAL0 to PAL15 can be set to input or output in 1-bit units using the port AL mode register (PMAL).

(2) Control mode

PAL0 to PAL15 can be set to port or control mode in 1-bit units using the PMCAL register.

(a) A0 to A15 (Address bus) … 3-state output[Note]

These are the address output pins of the lower 16 bits of the 22-bit address bus when the external memory is accessed.

2.3.13 PAH0 to PAH5 (Port AH) … I/O

Port AH is a 6-bit I/O port in which input or output can be set for each port pin individually.

Besides functioning as a port, in control mode, these pins operate as the address bus (A16 to A21) when memory is expanded externally.

The operation mode can be specified by the port AH mode control register (PMCAH) to port or control mode for each port pin individually.

(1) Port mode

PAH0 to PAH5 can be set to input or output in 1-bit units using the port AH mode register (PMAH).

(2) Control mode

PAH0 to PAH6 can be set to port or control mode in 1-bit units using the PMCAH register.

(a) A16 to A21 (Address bus) … 3-state output[Note]

These are the address output pins of the higher 6 bits of the 22-bit address bus when the external memory is accessed.

Note not available on µPD70F3447

2.3.14 PDL0 to PDL15 (Port DL) … I/O

Port DL is an 8-bit or a 16-bit I/O port in which input or output can be set for each port pin individually.

Besides functioning as a port, in control mode, these pins operate as the data bus (D0 to D15) when memory is expanded externally.

The operation mode can be specified by the port DL mode control register (PMCDL) to port or control mode for each port pin individually.

(1) Port mode

PDL0 to PDL15 can be set to input or output in 1-bit units using the port DL mode register (PMDL).

(2) Control mode

PDL0 to PDL15 can be set to port or control mode in 1-bit units using the PMCDL register.

(a) D0 to D15 (Address bus) … 3-state I/O[Note]

These are the data I/O pins of the lower 16 bits of the 32-bit data bus when the external memory is accessed.

2.3.15 PDH0 to PDH15 (Port DH) … I/O

Port DH is an 8-bit or a 16-bit I/O port in which input or output can be set for each port pin individually.

Besides functioning as a port, in control mode, these pins operate as the data bus[Note] (D16 to D31) when memory is expanded externally.

The operation mode can be specified by the port DH mode control register (PMCDH) to port or control mode for each port pin individually.

(1) Port mode

PDH0 to PDH15 can be set to input or output in 1-bit units using the port DH mode register (PMDH).

(2) Control mode

PDH0 to PDH15 can be set to port or control mode in 1-bit units using the PMCDH register.

(a) D16 to D31 (Address bus) … 3-state I/O[Note]

These are the data I/O pins of the higher 16 bits of the 32-bit data bus when the external memory is accessed.

Note not available on µPD70F3447

2.3.16 PCD2 to PCD5 (Port CD) … I/O

Port CD is a 4-bit I/O port in which input or output can be set for each port pin individually.

Besides functioning as a port, in control mode, these pins operate as control signal outputs[Note] when memory is expanded externally.

The operation mode can be specified by the port CD mode control register (PMCCD) to port or control mode for each port pin individually.

(1) Port mode

PCD2 to PCD5 can be set to input or output in 1-bit units using the port CD mode register (PMCD).

(2) Control mode

PCD2 to PCD5 can be set to port or control mode in 1-bit units using the PMCCD register.

(a) $\overline{\text{BEN0}}$ to $\overline{\text{BEN3}}$ (Byte enable) … 3-state output[Note]

These are the byte enable control signal pins, which indicate the validity of the corresponding byte on the 32-bit data bus.

2.3.17 PCM0, PCM1, PCM6, PCM7 (Port CM) … I/O

Port CM is a 4-bit I/O port in which input or output can be set for each port pin individually.

Besides functioning as a port, in control mode, these pins operate as control signal input[Note] when memory is expanded externally.

The operation mode can be specified by the port CM mode control register (PMCCM) to port or control mode for each port pin individually.

(1) Port mode

PCM0, PCM1, PCM6, and PCM7 can be set to input or output in 1-bit units using the port CM mode register (PMCM).

(2) Control mode

PCM0 can be set to port or control mode in 1-bit units using the PMCCM register.

(a) $\overline{\text{WAIT}}$ (Wait) … Input[Note]

This is the control signal input pin at which an external data wait is inserted into the bus cycle. The $\overline{\text{WAIT}}$ signal can be input asynchronously, and is sampled at the falling edge of the $\overline{\text{BCLK}}$ signal. When the setup or hold time is terminated within the sampling timing, wait insertion may not be executed.

Note not available on μPD70F3447

2.3.18 PCS0, PCS1, PCS3, PCS4 (Port CS) … I/O

Port CS is a 4-bit I/O port in which input or output can be set for each port pin individually.

Besides functioning as a port, in control mode, these pins operate as control signal outputs[Note] when memory is expanded externally.

The operation mode can be specified by the port CS mode control register (PMCCS) to port or control mode for each port pin individually.

(1) Port mode

PCS0, PCS1, PCS3, and PCS4 can be set to input or output in 1-bit units using the port CS mode register (PMCS).

(2) Control mode

PCS0, PCS1, PCS3, and PCS4 can be set to port or control mode in 1-bit units using the PMCCS register.

(a) $\overline{CS0}$, $\overline{CS1}$, $\overline{CS3}$, $\overline{CS4}$ (Chip select) … 3-state output[Note]

These are the chip select signal output pins for the external memory or peripheral I/O extension areas.The $\overline{CSn}$ signal is assigned to the memory block n (n = 0, 1, 3, 4).It becomes active while the bus cycle that accesses the corresponding memory block is activated. In the idle state (TI), it becomes inactive.

Note not available on µPD70F3447

2.3.19 PCT4, PCT5 (Port CT) … I/O

Port CT is a 2-bit I/O port in which input or output can be set for each port pin individually.

Besides functioning as a port, in control mode, these pins operate as control signal outputs[Note] when memory is expanded externally.

The operation mode can be specified by the port CT mode control register (PMCCT) to port or control mode for each port pin individually.

(1) Port mode

PCT4 and PCT5 can be set to input or output in 1-bit units using the port CT mode register (PMCT).

(2) Control mode

PCT4 and PCT5 can be set to port or control mode in 1-bit units using the PMCCT register.

(a) $\overline{RD}$ (Read strobe) … 3-state output[Note]

This is a strobe signal output pin that shows whether the bus cycle currently being executed is a read cycle for the external memory or peripheral I/O extension area. In the idle state (TI), it becomes inactive.

(b) $\overline{WR}$ (Write strobe) … 3-state output[Note]

This is a strobe signal output pin that shows whether the bus cycle currently being executed is a write cycle for the external memory or peripheral I/O extension area.

2.3.20 DCK (Debug clock) … Input

This pin inputs a debug clock. At the rising edge of the DCK signal, the DMS and DDI signals are sampled, and data is output from the DDO pin at the falling edge of the DCK signal. Keep this pin high when the debug function is not used.

2.3.21 DDI (Debug data input) … Input

This pin inputs debug data, which is sampled at the rising edge of the DCK signal when the debug serial interface is in the shift state. Data is input with the LSB first. Keep this pin high when the debug function is not used.

2.3.22 DDO (Debug data output) … Output

This pin outputs debug data at the falling edge of the DCK signal when the debug serial interface is in the shift state. Data is output with the LSB first.

Note not available on µPD70F34477

2.3.23 DMS (Debug mode select) … Input

This input pin selects a debug mode. Depending on the level of the DMS signal, the state machine of the debug serial interface changes. This pin is sampled at the rising edge of the DCK signal. Keep this pin high when the debug function is not used.

2.3.24 $\overline{\text{DRST}}$ (Debug reset) … Input

This pin inputs a debug reset signal that is a negative-logic signal to initialize the DCU asynchronously.

When this signal goes low, the DCU is reset/invalidated. Keep this pin low when the debug function is not used.

2.3.25 MODE0 to MODE2 (Mode) … Input

These are input pins used to specify the operating mode.

2.3.26 FLMD0, FLMD1 (flash programming mode)

These are input pins used to specify the flash programming mode.

2.3.27 $\overline{\text{RESET}}$ (Reset) … Input

$\overline{\text{RESET}}$ is a signal that is input asynchronously and that has a constant low level width regardless of the operating clock's status. When this signal is input, a system reset is executed as the first priority ahead of all other operations.

In addition to being used for ordinary initialization/start operations, this pin can also be used to release a standby mode (HALT).

2.3.28 X1, X2 (Crystal)

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

2.3.29 ANI00 to ANI09, ANI10 to ANI19 (Analog input) … Input

These are analog input pins of the corresponding A/D converter (ADC0, ADC1).

2.3.30 AV_{REF0}, AV_{REF1} (Analog reference voltage) … Input

These are reference voltage supply pins for the corresponding A/D converter (ADC0, ADC1).

2.3.31 AV_{DD} (Analog power supply)

This is the positive power supply pin for the A/D converters.

2.3.32 AV_{SS} (Analog ground)

This is the analog ground pin for the A/D converters.

2.3.33 CV_{DD} (Power supply for clock generator)

This is the positive power supply pin for the clock generator.

2.3.34 CV_{SS} (Ground for clock oscillator)

This is the ground pin for the clock generator.

2.3.35 V_{DD10} to V_{DD15} (Power supply)

These are the positive power supply pins for the internal CPU.

2.3.36 V_{DD30} to V_{DD37} (Power supply)

These are the positive power supply pins for the peripheral interface.

2.3.37 V_{SS10} to V_{SS15} (Ground)

These are the ground pins for the internal CPU.

2.3.38 V_{SS30} to V_{SS37} (Ground)

These are the ground pins for the peripheral interface.

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

Table 2-4 Recommended connection of unused pins (1/3)

Terminal		I/O circuit type	Recommended termination
μPD70F3187	μPD70F3447		
P00/NMI		2	Connect independently to V_{SS3} via a resistor
P01/INTP0/ESO0			
P02/INTP1/ESO1			
P03INTP2/ADTRG0			
P04INTP3/ADTRG1			
P10/TIP00/TEVTP1/TOP00		5-K	Input: Connect independently to V_{DD3} or V_{SS3} via a resistor Output: leave open
P11/TIP01/TTRGP1/TOP01			
P12/TIP10/TTRGP0/TOP10			
P13/TIP11/TEVTP0/TOP11			
P14/TIP20/TEVTP3/TOP20			
P15/TIP21/TTRGP3/TOP21			
P16/TIP30/TTRGP2/TOP30			
P17/TIP31/TEVTP2/TOP31			
P20/TIP40/TEVTP5/TOP40			
P21/TIP41/TTRGP5/TOP41			
P22/TIP50/TTRGP4/TOP50			
P23/TIP51/TEVTP4/TOP51			
P24/TIP60/TEVTP7/TOP60			
P25/TIP61/TTRGP7/TOP61			
P26/TIP70/TTRGP6/TOP70			
P27/TIP71/TEVTP6/TOP71			
P30/RXDC0/INTP4			
P31/TXDC0			
P32/RXDC1/INTP5			
P33/TXDC1			
P34/FCRXD0			
P35/FCTXD0			
P36/FCRXD1	P36		
P37/FCTXD1	P37		
P40/SIB0			
P41/SOB0			
P42/$\overline{\text{SCKB0}}$			
P43/SIB1	P43		
P44/SOB1	P44		
P45/$\overline{\text{SCKB1}}$	P45		
P50/TOR00 to P57/TOR07			
P60/TOR10/TTRGR1			

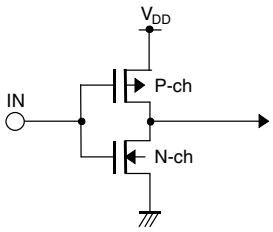
Table 2-4 Recommended connection of unused pins (2/3)

Terminal		I/O circuit type	Recommended termination
μPD70F3187	μPD70F3447		
P61/TOR11/TIR10		5-K	Input: Connect independently to V_{DD3} or V_{SS3} via a resistor Output: leave open
P62/TOR12/TIR11			
P63/TOR13/TIR12			
P64/TOR14/TIR13			
P65/TOR15			
P66/TOR16			
P67/TOR17/TEVTR1			
P70/TIT00/TEVTT1/TOT00/TENCT00			
P71/TIT01/TTRGT1/TOT01/TENCT01			
P72/TECRT0/INTP12			
P73/TIT10/TTRGT0/TOT10/TENCT10			
P74/TIT11/TEVTT0/TOT11/TENCT11			
P75/TECRT1/AFO			
P80/SI30			
P81/SO30			
P82/$\overline{\text{SCK30}}$			
P83/SCS300/INTP6			
P84/SCS301/INTP7			
P85/SCS302/INTP8			
P86/SCS303/$\overline{\text{SSB0}}$			
P90/SI31	P90		
P91/SO31	P91		
P92/$\overline{\text{SCK31}}$	P92		
P93/SCS310/ INTP9	P93/INTP9		
P94/SCS311/ INTP10	P94/INTP10		
P95/SCS312 /INTP11	P95/INTP11		
P96/SCS313/ $\overline{\text{SSB1}}$	P96		
P100/TCLR1/ TICC10/TOP80	P100/TOP80		
P101/TCUD1/ TICC11	P101		
P102/TIUD1/TO1	P102		
PAH0/A16 to PAH5/A21	PAH0 to PAH5	5	Input: Connect independently to V_{DD3} or V_{SS3} via a resistor Output: leave open
PAL0/A0 to PAL15/A15	PAL0 to PAL15		
PDH0/D16 to PDH15/D31	PDH0 to PDH15		

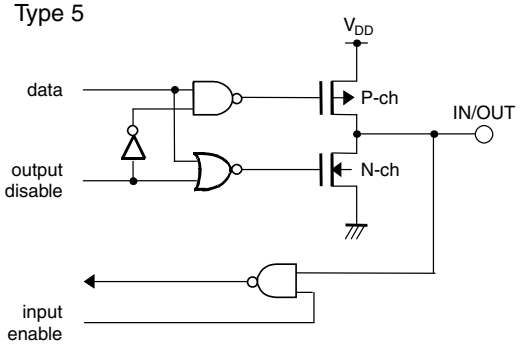
Table 2-4 Recommended connection of unused pins (3/3)

Terminal		I/O circuit type	Recommended termination
μPD70F3187	μPD70F3447		
PDL0/D0 to PDL15/D15	PDL0 to PDL15	5	Input: Connect independently to V_{DD3} or V_{SS3} via a resistor Output: leave open
PCS0/$\overline{CS0}$	PCS0		
PCS1/$\overline{CS1}$	PCS1		
PCS3/$\overline{CS3}$	PCS3		
PCS4/$\overline{CS4}$	PCS4		
PCD2/$\overline{BEN0}$ to PCD5/$\overline{BEN3}$	PCD2 to PCD5		
PCT4/$\overline{RD}$	PCT4	5	Input: Connect independently to V_{DD3} or V_{SS3} via a resistor Output: leave open
PCT5/$\overline{WR}$	PCT5		
PCM0/$\overline{WAIT}$	PCM0		
PCM1	PCM1		
PCM6	PCM6		
PCM7	PCM7		
$\overline{RESET}$		2	Pin must be used in the intended way
X1		–	
X2		–	
MODE0/FLMD0		2	
MODE1/FLMD1		2	
MODE2		2	
DCK		1	Connect independently to V_{DD3} via a resistor
$\overline{DRST}$		2-I	Leave open (on-chip pull-down resistor
DMS		1	Connect independently to V_{DD3} via a resistor
DDI			
DDO		3	Leave open (always level output during reset)
ANI00 to ANI09		7	Connect independently to AV_{DD} or AV_{SS} via a resistor
ANI10 to ANI19			
AV_{REF0}		–	Connect independently to AV_{SS} via a resistor
AV_{REF1}			
AV_{DD}		–	Pin must be used in the intended way
AV_{SS0}			
AV_{SS1}			
V_{DD10} to V_{DD15}			
V_{SS10} to V_{SS15}			
V_{DD30} to V_{DD37}			
V_{SS30} to V_{SS37}			
CV_{DD}			
CV_{SS}			

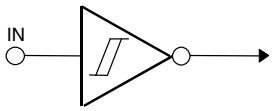
Type 1



Type 5

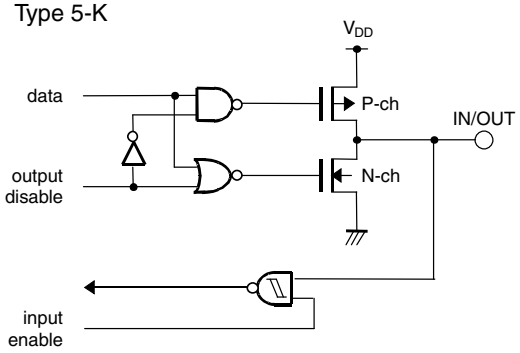


Type 2

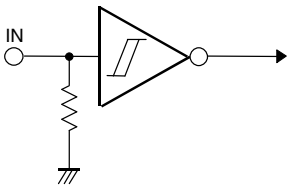


Schmitt trigger input with hysteresis characteristics

Type 5-K

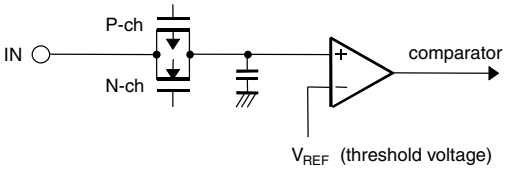


Type 2-I



Schmitt trigger input with hysteresis characteristics

Type 7



Type 3

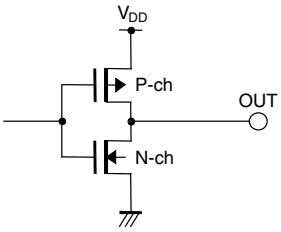


Figure 2-1 Pin I/O Circuits

2.5 Noise Suppression

The V850E/PH2 has a digital or analog delay circuits for noise suppression on all edge sensitive inputs.

The digital delay circuit suppresses input pulses shorter than the internally generated edge detection signal to assure the hold time for these signals. The noise suppression is only effective on alternate pin functions, and it is not effective when the port input function is selected.

Table 2-5 Noise Suppression Timing

Pin Function	Noise removal time	Clock Source
NMI	4 to 5 clocks	f_{XX}/16 or f_{XX}/64 (set by NCR0 bit of NRC register)
INTP0, INTP1, ESO0, ESO1	Analog delay (60ns to 200ns)	
INTP2 to INTP11, ADTRG0, ADTRG1	4 to 5 clocks	f_{XX}/16 or f_{XX}/64 (set by NCR1 bit of NRC register)
INTP12, TICC00[a], TICC01[a], TCLR0[a], TCUD0[a], TIUD0[a], TIT00, TIT01, TIT10, TIT11, TECRT0, TECRT1, TEVTT0, TEVTT1, TTRGT0, TTRGT1, TENCT00, TENCT01, TENCT10, TENCT11	4 to 5 clocks	f_{XX}/16 or f_{XX}/64 (set by NCR2 bit of NRC register)
TIP00, TIP01, TIP10, TIP11, TEVTP0, TEVTP1, TTRGP0, TTRGP1	4 to 5 clocks	f_{XX}/16 or f_{XX}/64 (set by NCR3 bit of NRC register)
TIP20, TIP21, TIP30, TIP31, TEVTP2, TEVTP3, TTRGP2, TTRGP3	4 to 5 clocks	f_{XX}/16 or f_{XX}/64 (set by NCR4 bit of NRC register)
TIP40, TIP41, TIP50, TIP51, TEVTP4, TEVTP5, TTRGP4, TTRGP5	4 to 5 clocks	f_{XX}/16 or f_{XX}/64 (set by NCR5 bit of NRC register)
TIP60, TIP61, TIP70, TIP71, TEVTP6, TEVTP7, TTRGP6, TTRGP7	4 to 5 clocks	f_{XX}/16 or f_{XX}/64 (set by NCR6 bit of NRC register)
TIR10 to TIR13, TEVTR1, TTRGR1	4 to 5 clocks	f_{XX}/16 or f_{XX}/64 (set by NCR7 bit of NRC register)

a) not available on µPD70F3447

(1) Noise removal time control register (NRC)

The NRC register specifies the noise removal clock setting for different edge sensitive inputs.

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

Address $FFFFF7A0_H$

Initial Value 00_H. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
NRC	NCR7	NCR6	NCR5	NCR4	NCR3	NCR2	NCR1	NCR0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2-6 NRC register contents

Bit position	Bit name	Function
7	NCR7	Noise removal clock setting for input pins TIR10 to TIR13, TEVTR1, TTRGR1 0: $f_{XX}/16$ 1: $f_{XX}/64$
6	NCR6	Noise removal clock setting for input pins TIP60, TIP61, TIP70, TIP71, TEVTP6, TEVTP7, TTRGP6, TTRGP7 0: $f_{XX}/16$ 1: $f_{XX}/64$
5	NCR5	Noise removal clock setting for input pins TIP40, TIP41, TIP50, TIP51, TEVTP4, TEVTP5, TTRGP4, TTRGP5 0: $f_{XX}/16$ 1: $f_{XX}/64$
4	NCR4	Noise removal clock setting for input pins TIP20, TIP21, TIP30, TIP31, TEVTP2, TEVTP3, TTRGP2, TTRGP3 0: $f_{XX}/16$ 1: $f_{XX}/64$
3	NCR3	Noise removal clock setting for input pins TIP00, TIP01, TIP10, TIP11, TEVTP0, TEVTP1, TTRGP0, TTRGP1 0: $f_{XX}/16$ 1: $f_{XX}/64$
2	NCR2	Noise removal clock setting for input pins INTP12, TICC00, TICC01, TCLR0, TCUD0, TIUD0, TIT00, TIT01, TIT10, TIT11, TECRT0, TECRT1, TEVTT0, TEVTT1, TTRGT0, TTRGT1, TENCT00, TENCT01, TENCT10, TENCT11 0: $f_{XX}/16$ 1: $f_{XX}/64$ **Note:** Input pins TIC00, TIC01, TCRL0, TCUD0, and TIUD0 are not available on μPD70F3447
1	NCR1	Noise removal clock setting for input pins INTP2 to INTP11, ADTRG0, ADTRG1 0: $f_{XX}/16$ 1: $f_{XX}/64$
0	NCR0	Noise removal clock setting for NMI input pin 0: $f_{XX}/16$ 1: $f_{XX}/64$

Chapter 3 CPU Functions

The CPU of the V850E/PH2 microcontroller is based on the RISC architecture and executes most instructions in one clock cycle by using a 5-stage pipeline control.

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 fivestage 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,
  e.g. 32 bits × 32 bits → 64 bits in 1 to 2 clock cycles
- Saturated operation instructions
- Floating point arithmetic unit
  (single precision, 32 bits, IEEE754-85 standard)
- Barrel shifter (32-bit shift in one clock cycle)
- Load/store instruction with long/short format
- Four types of bit manipulation instructions: set, clear, not, test

3.1.1 CPU system structure

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.

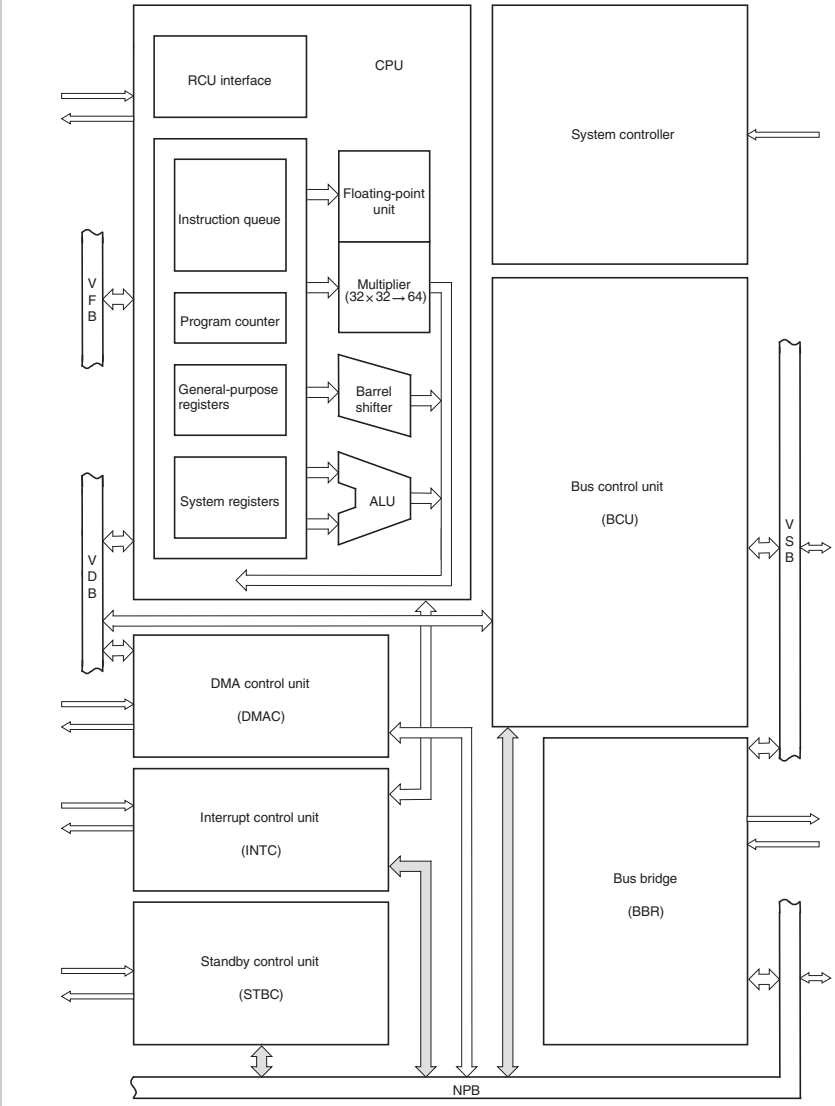


Figure 3-1 CPU system

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.

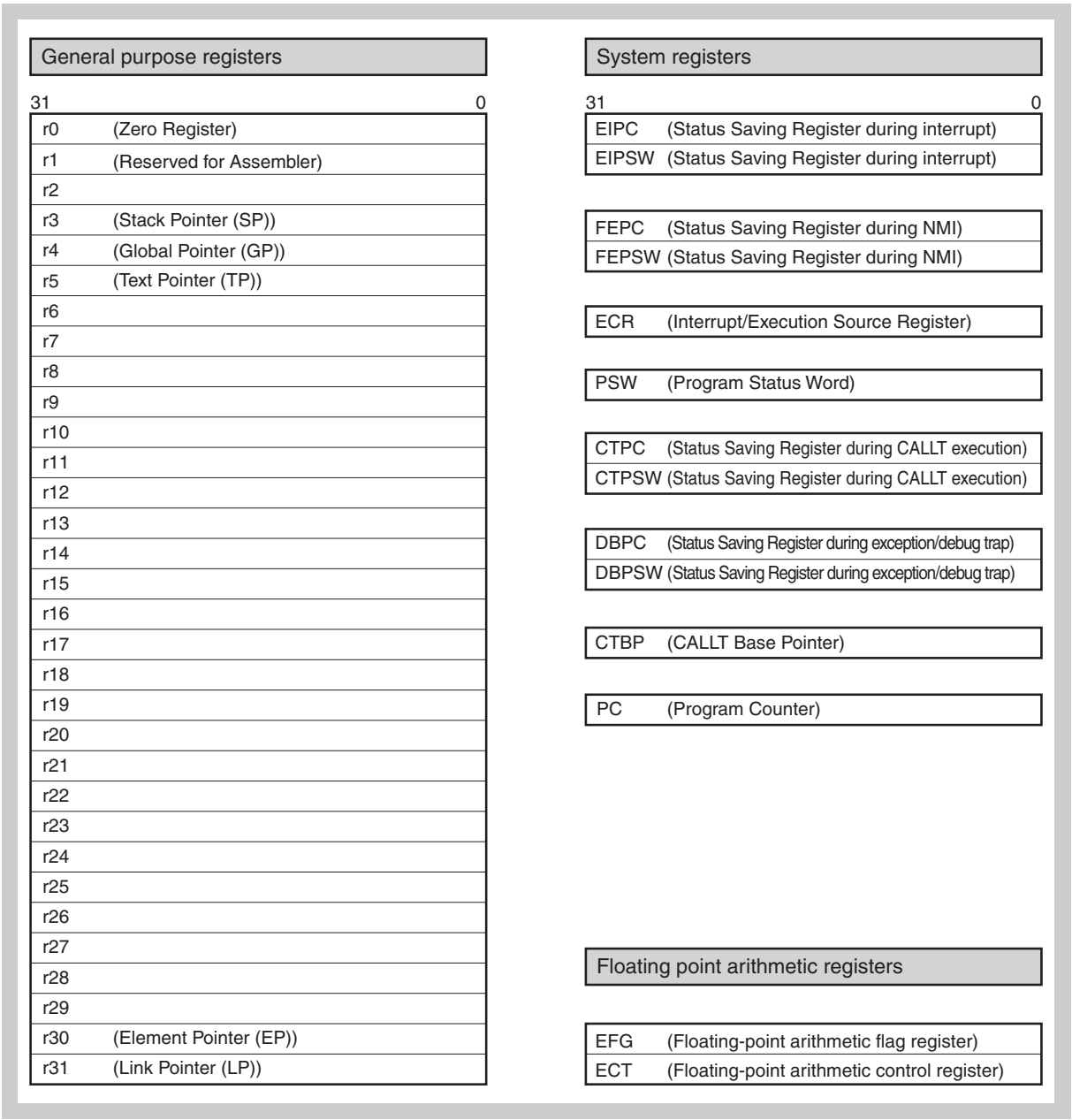


Figure 3-2 CPU Register Set

3.2.1 Program register set

The program register set includes general-purpose registers and a program counter.

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

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

However, r0 and r30 are implicitly used by instructions and care must be exercised when using these registers. r0 always holds 0 and is used for operations that use 0 or offset 0 addressing. r30 is used as a base pointer when performing memory access with the SLD and SST short instructions.

Also, r1, r3 to r5, and r31 are implicitly used by the assembler and C compiler. Therefore, before using these registers, their contents must be saved so that they are not lost, and they must be restored to the registers after use. There are cases when r2 is used by the real-time OS. If r2 is not used by the real-time OS, r2 can be used as a variable register.

Table 3-2 Program Registers

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

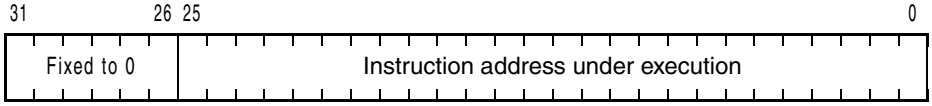
(2) Program counter (PC)

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

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

Initial value $0000\ 0000_H$

PC



3.2.2 System register set

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

Read from and write to a system register is performed by setting the system register number (regID) shown below within the system register load/store instructions (LDSR, STSR instructions).

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 (×) for the register or not (–).

Caution Even if bit 0 of EIPC, FEPC, or CTPC is set to (1) by the LDSR instruction, bit 0 is ignored during return with the RETI instruction following interrupt servicing (because bit 0 of PC is fixed to 0). If setting a value to EIPC, FEPC, and CTPC, set an even number (bit 0 = 0).

Table 3-3 System Register Numbers

System Register			Operand Specification Enabled for instruction	
regID	Name	Function	LDSR	STSR
0	EIPC	PC value at Interrupt handler entry [a]	×	×
1	EIPSW	PSW value at Interrupt handler entry [a]	×	×
2	FEPC	PC value at NMI handler entry	×	×
3	FEPSW	PSW value at NMI handler entry	×	×
4	ECR	Exception Cause Register	–	×
5	PSW	Program status word	×	×
6 to 15	-	Reserved numbers for future function expansion (The operation is not guaranteed if accessed.)	–	–
16	CTPC	PC value at CALLT subroutine entry [a]	×	×
17	CTPSW	PSW value at CALLT subroutine entry [a]	×	×
18	DBPC	PC value at exception/debug trap entry	×[b]	×
19	DBPSW	PSW value at exception/debug trap entry	×[b]	×
20	CTBP	CALLT base pointer	×	×
21 to 31	-	Reserved numbers for future function expansion (The operation is not guaranteed if accessed.)	–	–

a) Since only one set of registers is available, the contents of these registers must be saved by the program when multiple interrupt servicing is enabled.

b) Reading from this register is only enabled between a DBTRAP exception (exception handler address $0000\ 0060_H$) and the exception handler terminating DBRET instruction. DBTRAP exceptions are generated upon ILGOP detections (refer to *"Exception Trap" on page 205*).

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

There are two context saving registers, EIPC and EIPSW.

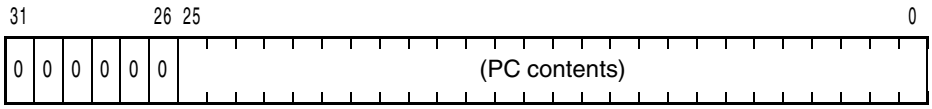
Upon occurrence of a software exception or a maskable interrupt, the content of the program counter (PC) is saved to EIPC and the content of the program status word (PSW) is saved to EIPSW (upon occurrence of a non-maskable interrupt (NMI), the contents are saved to the NMI status saving registers (FEPC, FEPSW)). The address of the next instruction following the instruction executed when a software exception or maskable interrupt occurs is saved to EIPC, except for the DIVH instruction (refer to *"Interrupt/Exception Processing Function" on page 171*).

The values of EIPC and EIPSW are restored to PC and PSW during execution of a RETI instruction.

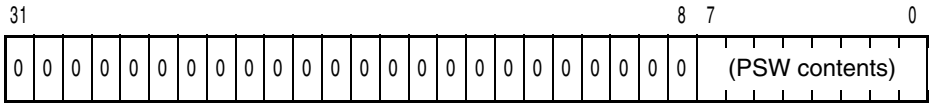
Bits 31 to 26 of EIPC and bits 31 to 8 of EIPSW are reserved (fixed to 0) for future function expansion.

Initial value
EIPC: Undefined.
EIPSW: Undefined

EIPC



EIPSW



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

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

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

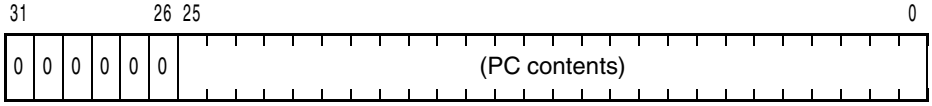
Upon occurrence of a non-maskable interrupt (NMI), the content of the program counter (PC) is saved to FEPC and the content of the program status word (PSW) is saved to FEPSW. The address of the next instruction following the instruction executed when a non-maskable interrupt occurs is saved to FEPC, except for the DIVH instruction.

The values of FEPC and FEPSW are restored to PC and PSW during execution of a RETI instruction.

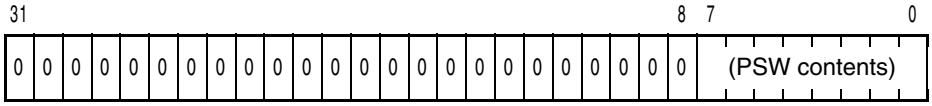
Bits 31 to 26 of FEPC and bits 31 to 8 of FEPSW are reserved (fixed to 0) for future function expansion.

Initial value FEPC: Undefined.
FEPSW: Undefined

FEPC



FEPSW



(3) Exception cause register (ECR)

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

Initail value $0000\ 0000_H$

ECR

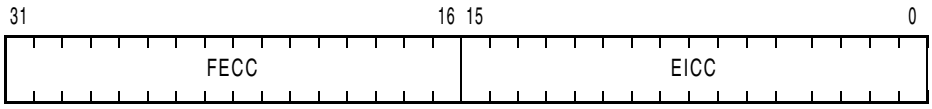


Table 3-4 ECR register contents

Bit position	Bit name	Description
31 to 16	FECC	Non-maskable interrupt (NMI) exception code
15 to 0	EICC	Exception, maskable interrupt exception code

The list of exception codes is tabulated in *Table 7-1, “Interrupt/exception source list,” on page 172*.

(4) Program status word (PSW)

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

When the contents of this register are changed using the LDSR instruction, the new contents become valid immediately following completion of the LDSR instruction execution. However, if the ID flag is set to 1, interrupt request acknowledgement during LDSR instruction execution is prohibited.

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

Initial value $0000\ 0020_H$

PSW

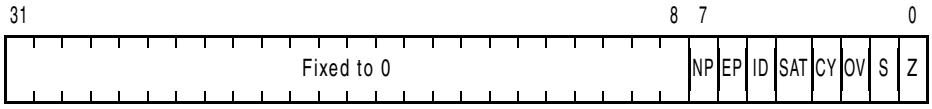


Table 3-5 PSW register contents

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

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

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 Saturated operation results

Operation result status	Flag status			Saturated operation result
	SAT	OV	S	
Maximum positive value exceeded	1	1	0	7FFF FFFF$_H$
Maximum negative value exceeded	1	1	1	8000 0000$_H$
Positive (maximum value not exceeded)	Retains value before operation	0	0	Actual operation result
Negative (maximum value not exceeded)			1	

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

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

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

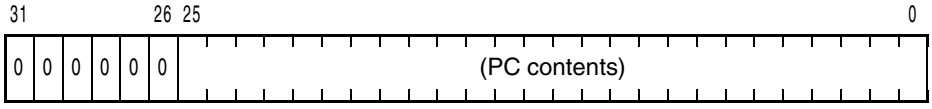
The values of CTPC and CTPSW are restored to PC and PSW during execution of the CTRET instruction.

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

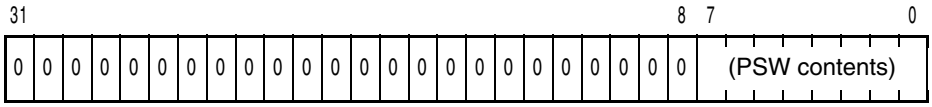
Initial value

CTPC: Undefined.
CTPSW: Undefined

CTPC



CTPSW



(6) CALLT base pointer (CTBP)

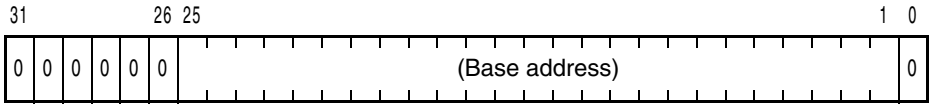
The CALLT base pointer (CTBP) is used to specify CALLT table start address and generate target addresses (bit 0 is fixed to 0).

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

Initial value

Undefined.

CTBP



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

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

Upon occurrence of an exception trap or debug trap, the contents of the program counter (PC) are saved to DBPC, and the program status word (PSW) contents are saved to DBPSW. The contents saved to DBPC consist of the address of the next instruction after the instruction executed when an exception trap or debug trap occurs.

The values of DBPC and DBPSW are restored to PC and PSW during execution of the DBRET instruction.

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

Initial value DBPC: Undefined.
DBPSW: Undefined

DBPC

31	30	29	28	27	26	25 ... 0
0	0	0	0	0	0	(PC contents)

DBPSW

31 ... 8	7 ... 0
0	(PSW contents)

3.2.3 Floating point arithmetic unit register set

The floating point arithmetic unit is provided with one flag register and one control register.

Table 3-7 Floating Point Arithmetic Unit Registers

Name	Usage	Operation
ECT	Control register	Sets the operation of the EFG register
EFG	Flag register	Holds the status of the FPU

(1) Floating point arithmetic control register (ECT)

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 the ECT register is a mask bit for one condition.

Initial value $0000\ 0000_H$

ECT

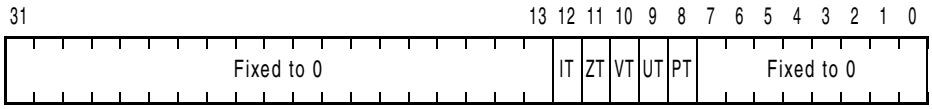


Table 3-8 ECT register contents

Bit position	Bit name	Description
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

(2) Floating point arithmetic status register (EFG)

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

Initial value $0000\ 0000_H$

EFG

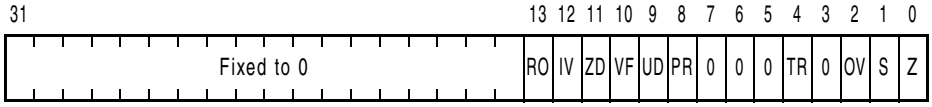


Table 3-9 EFG register contents

Bit position	Bit name	Description
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 13 to 8 is set. The setting conditions of this flag depends on the ECT register value.
2	OV	Indicates whether an overflow occurred during floating point to integer conversion 0: no overflow generated 1: overflow generated
1	S	Indicates whether floating point operation result is negative. 0: Operation result is not negative. 1: Operation result is negative.
0	Z	Indicates whether floating point operation result is 0. 0: Operation result is not 0. 1: Operation result is 0.

3.3 Operating Modes

The V850E/PH2 has the following operating modes.

3.3.1 Operating modes outline

(1) Normal operating mode

(a) Single-chip mode 0

Access to the internal ROM is enabled.

In single-chip mode 0, after the system reset is released, each pin related to the bus interface enters the port mode, program execution branches to the reset entry address of the internal ROM, and instruction processing starts. By setting the PMCDH, PMCDL, PMCCS, PMCCT, and PMCCM registers to control mode by instruction, an external device can be connected to the external memory area.

(b) Single-chip mode 1 (µPD70F3187 only)

In single-chip mode 1, after the system reset is released, each pin related to the bus interface enters the control mode, program execution branches to the external device's (memory) reset entry address, and instruction processing starts. The internal ROM area is mapped from address 100000H.

Note Single-chip mode 1 is not available on µPD70F3447.

(c) ROM-less mode (µPD70F3187 only)

After the system reset is released, each pin related to the bus interface enters the control mode, program execution branches to the external device's (memory) reset entry address, and
instruction processing starts. Fetching of instructions and data access for internal ROM becomes impossible.

In ROM-less mode the data bus width is 32 bits.

Note ROM-less mode 1 is not available on µPD70F3447.

(2) Flash memory programming mode

In this mode the internal flash memory can be written or erased with an external flash writer, using the CSIB0 or UARTC0 as serial interface.

3.3.2 Operation mode specification

The operation mode is specified according to the status of pins MODE0 to MODE2. In an
application system fix the specification of these pins and do not change them during operation. Operation is not guaranteed if these pins are changed during operation.

Table 3-10 Operation Mode Specification

MODE2	MODE1	MODE0	Mode	Remark
L	L	L	Single chip mode 0	Internal ROM area is allocated from address 00000000H.
L	L	H	Flash memory programming mode	CSIB0/IUARTC0 selected by MODE0 pin toggling.
L	H	L	ROM-less mode [a]	External 32-bit data bus
L	H	H	Single chip mode 1 [a]	Internal ROM area is allocated from address 00100000H. External 32-bit data bus
other value than above			Setting prohibited	

a) not available on µPD70F3447

3.4 Address Space

3.4.1 CPU address space

The CPU of the V850E/PH2 uses a 32-bit architecture and supports up to 4 GB of linear address space (data space) during operand addressing (data access). When addressing instructions, a linear address space (program space) of up to 64 MB is supported. However, both the program and data spaces include areas whose use is prohibited.

For details, refer to *Figure 3-4, "Address Space Image," on page 83*.

Figure 3-3 shows the CPU address space.

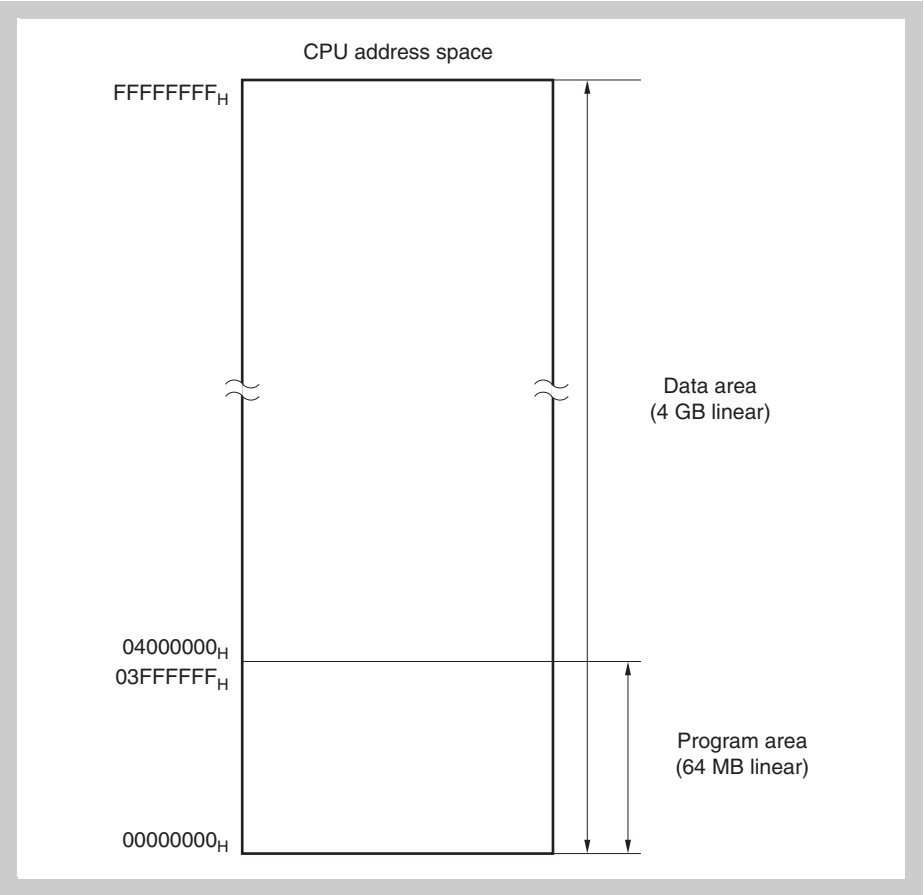


Figure 3-3 CPU Address Space

3.4.2 Images

When addressing an instruction address, up to 64 MB of linear address space (program space) and Internal RAM area are supported.

For operand addressing (data access), up to 4 GB of linear address space (data area) is supported. On

this 4 GB address space, however, 256 MB physical address spaces can be seen as an image. Therefore, whatever the values of bits 31 to 29 of an address may be, a physical address space of the same 256 MB is accessed.

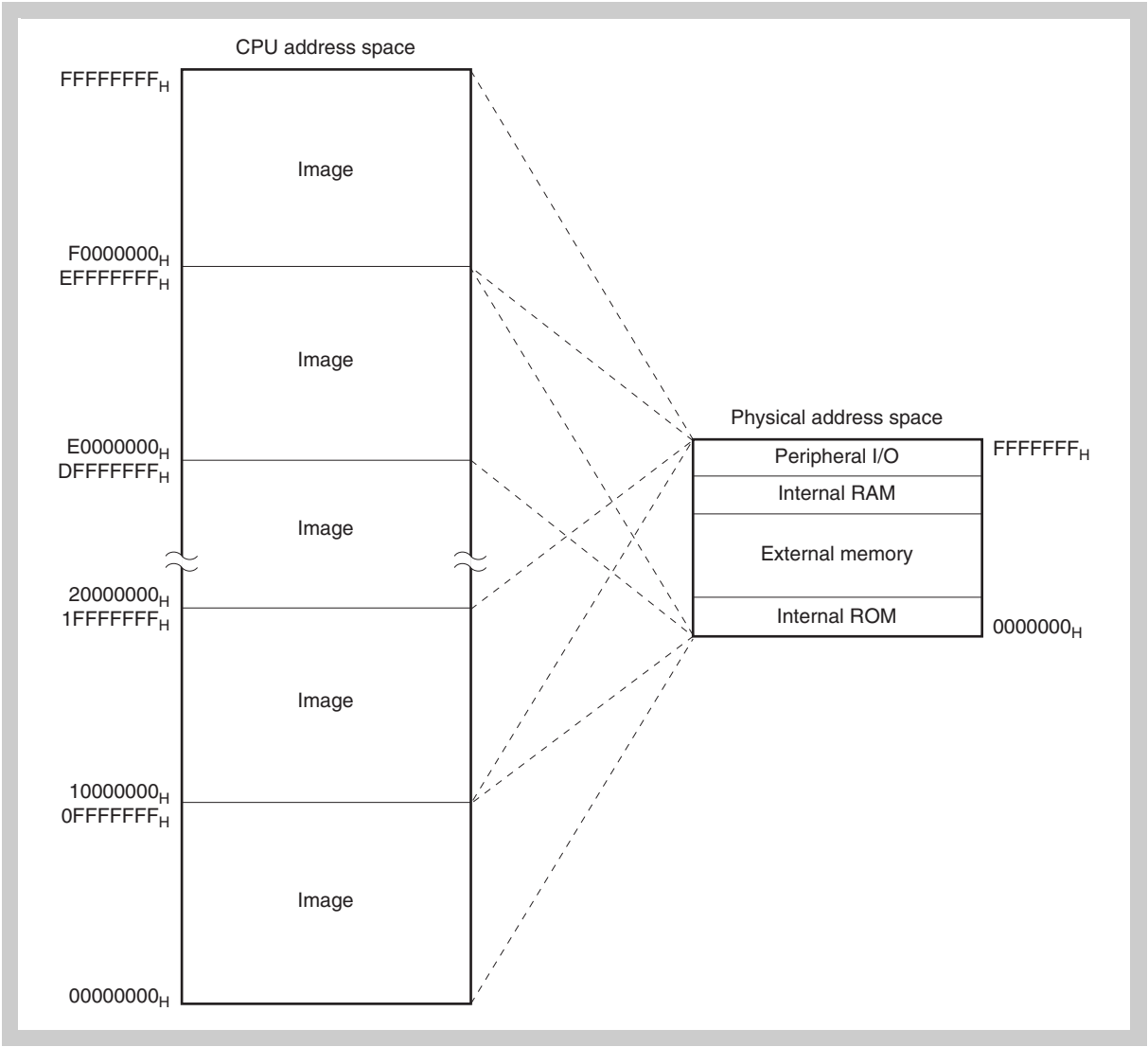


Figure 3-4 Address Space Image

3.4.3 Wrap-around of CPU address space

(1) Program space

Of the 32 bits of the program counter (PC), the higher 6 bits are fixed to 0 and only the lower 26 bits are valid. Even if a carry or borrow occurs from bit 25 to bit 26 as a result of branch address calculation, the higher 6 bits ignore this and remain 0.

Therefore, the lower-limit address of the program space, 00000000_H, and the upper-limit address, $03FFFFFF_H$, are contiguous addresses, and the program space is wrapped around at the boundary of these addresses.

Caution No instructions can be fetched from the 4 KB area of $03FFF000_H$ to $03FFFFFF_H$ because this area is a peripheral I/O area. Therefore, do not execute any branch
operation instructions in which the destination address will reside in any part of this area.

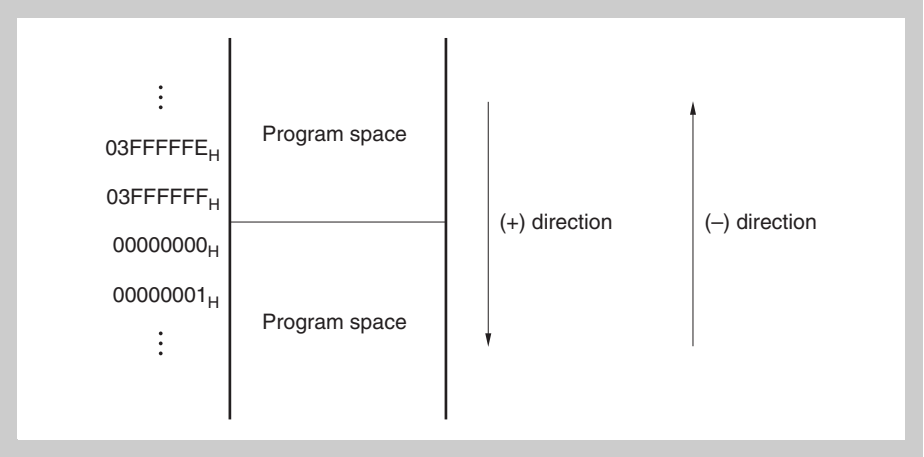


Figure 3-5 Program Space

(2) Data space

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

Therefore, the lower-limit address of the data space, address 00000000_H, and the upper-limit address, $FFFFFFFF_H$, are contiguous addresses, and the data space is wrapped around at the boundary of these addresses.

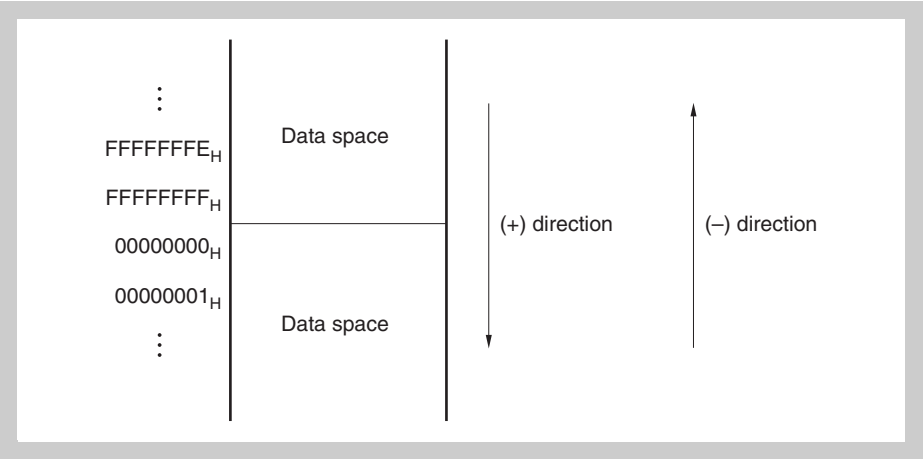


Figure 3-6 Data Space

3.4.4 Memory map

Areas are reserved in V850E/PH2 as shown in Figure 3-7. Each mode is specified by the MODE0 to MODE2 pins.

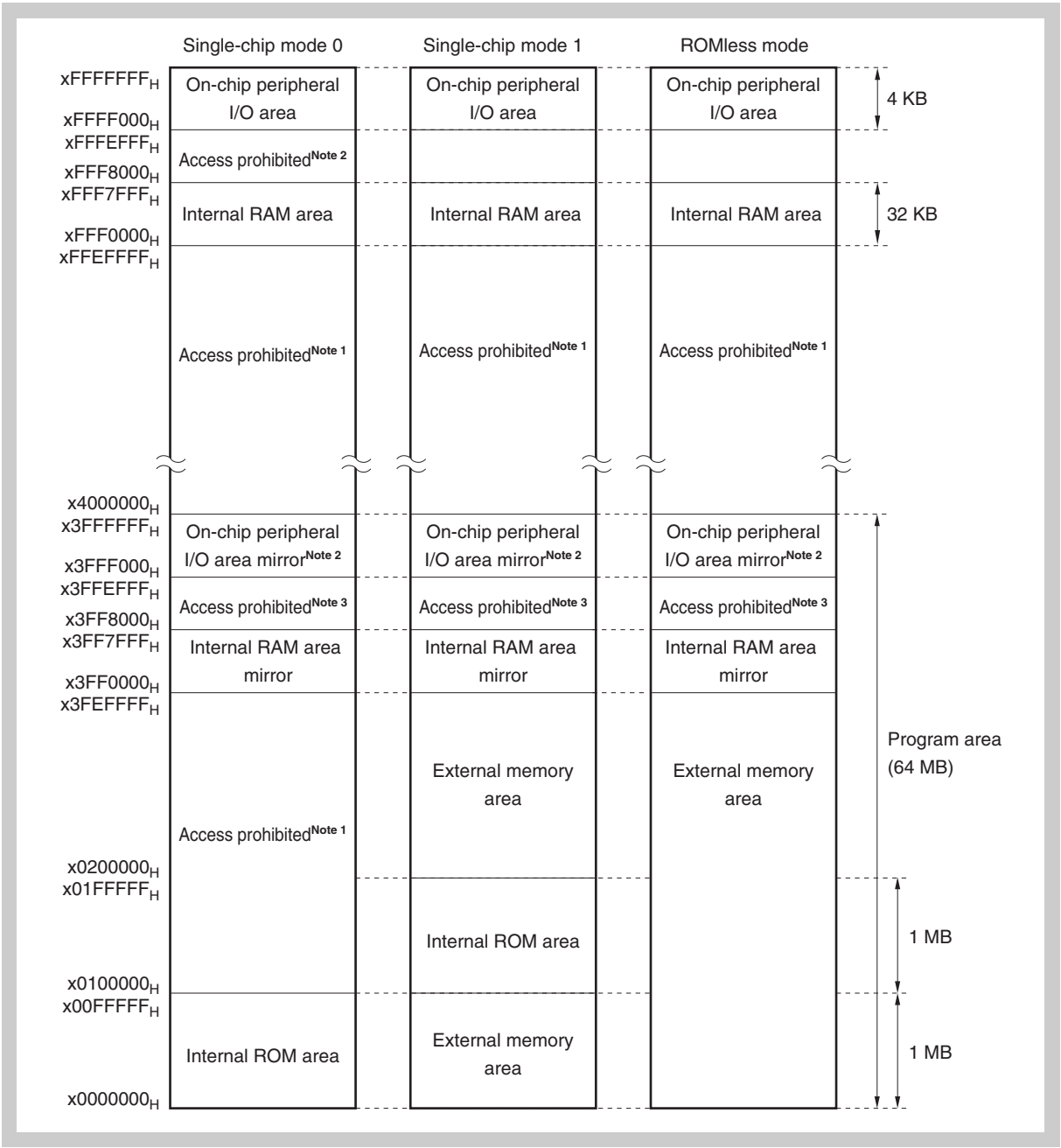


Figure 3-7 Memory Map of µPD70F3187

Note 1. By setting the PMCAL, PMCAH, PMCDL, PMCDH, PMCCS, PMCCT, and PMCCD port mode control registers to control mode, this area can be used as external memory area.

2. Accessing addresses $3FFF000_H$ to $3FFFFFF_H$ is prohibited. Specify addresses $FFFF000_H$ to $FFFFFFF_H$ to access the on-chip peripheral I/O.

3. The operation is not guaranteed if an access-prohibited area is accessed.

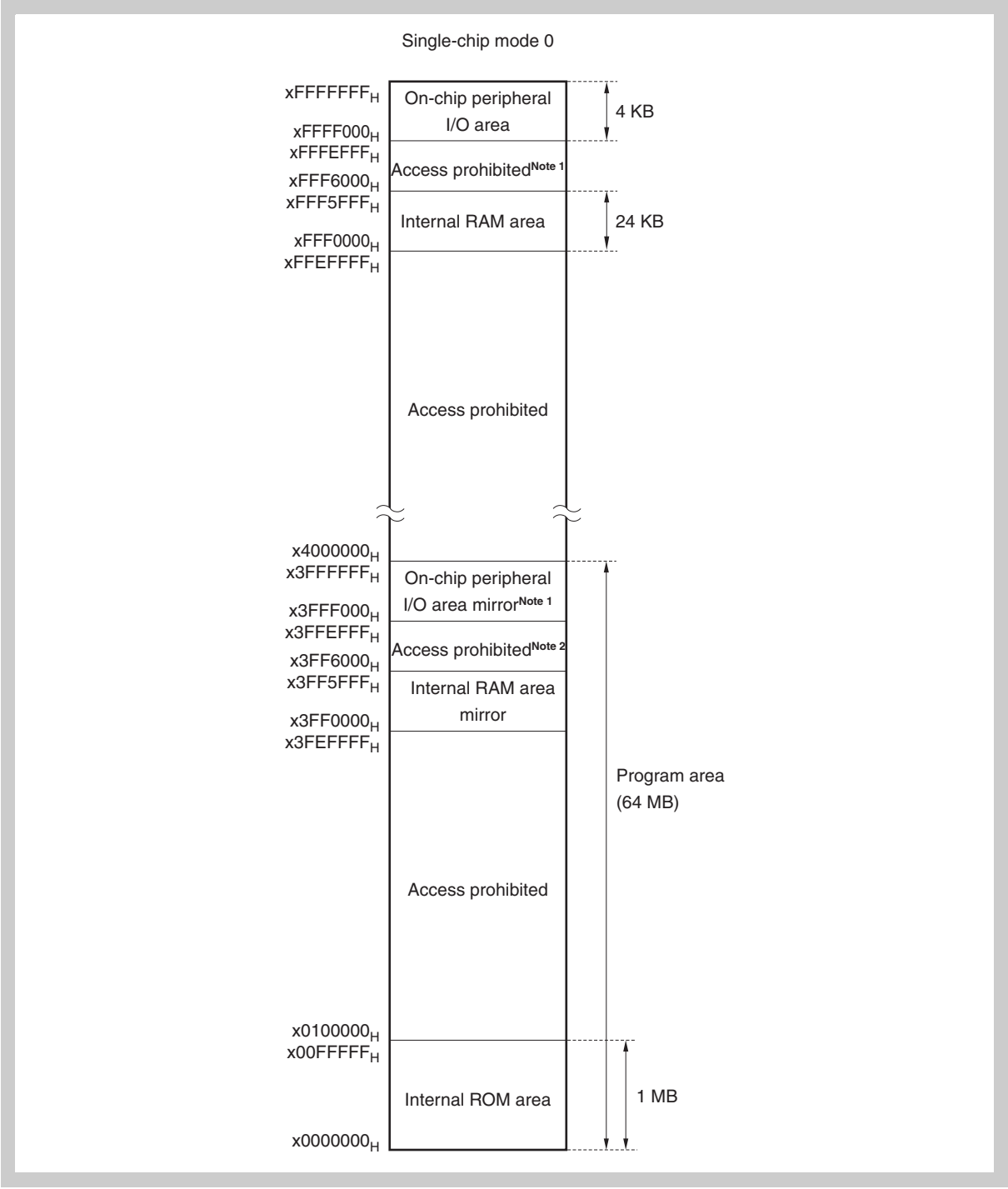


Figure 3-8 Memory Map of µPD70F3447

Note 1. Accessing addresses $3FFF000_H$ to $3FFFFFF_H$ is prohibited. Specify addresses $FFFF000_H$ to $FFFFFFF_H$ to access the on-chip peripheral I/O.

2. The operation is not guaranteed if an access-prohibited area is accessed.

3.4.5 Areas

(1) Internal ROM area

(a) Memory map

1MB of the internal area is reserved for the physical internal ROM (flash memory).

In case of µPD70F3187 internal flash memory of 512 KB are physically provided in the following addresses as internal ROM (flash memory).

- In single-chip mode 0: Addresses 000000_H to $07FFFF_H$ (addresses 080000_H to $0FFFFF_H$ are undefined)
- In single-chip mode 1: Addresses 0100000_H to $017FFF_H$ (addresses 0180000_H to $01FFFFF_H$ are undefined)

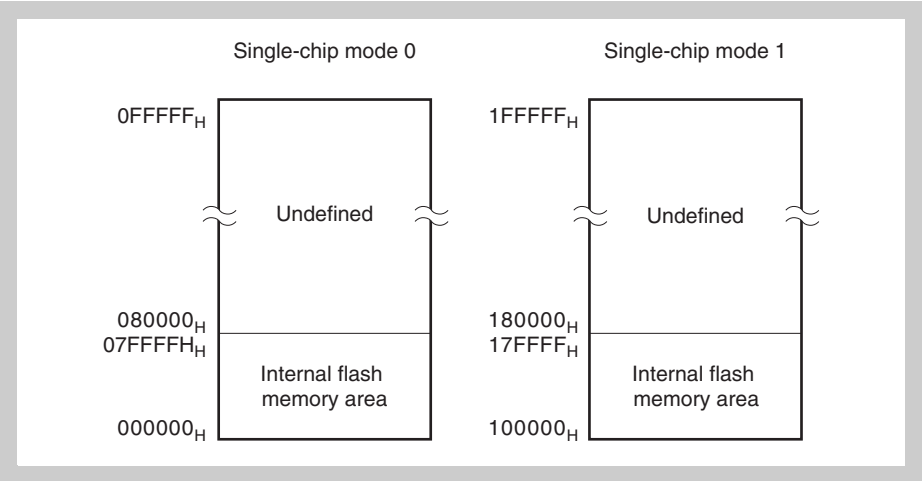


Figure 3-9 Internal ROM / Internal Flash Memory Area of µPD70F3187

In case of µPD70F3447 internal flash memory of 384 KB are physically provided in the following addresses as internal ROM (flash memory).

- In single-chip mode 0: Addresses 000000H to $05FFFF_H$ (addresses 060000H to $0FFFFF_H$ are undefined)

Remark Single-chip mode 1 is not supported on µPD70F3447.

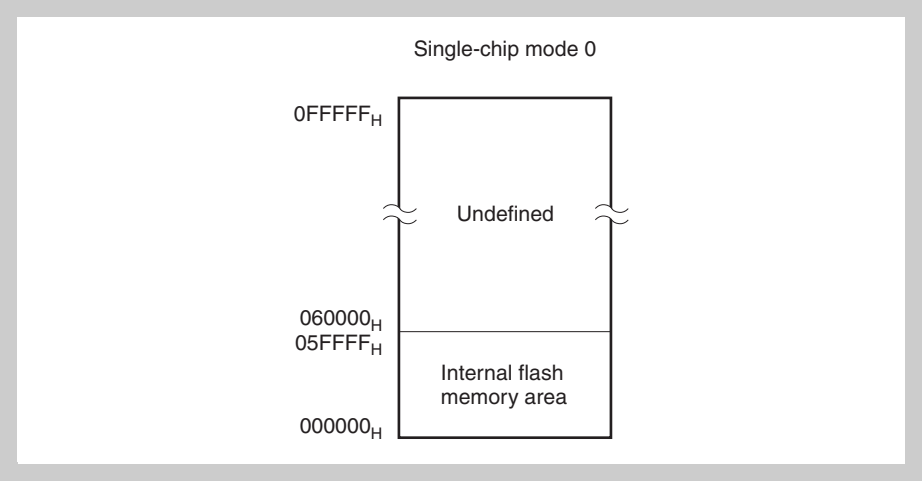


Figure 3-10 Internal ROM / Internal Flash Memory Area of µPD70F3447

(b) Interrupt/exception table

The V850E/PH2 increases the interrupt response speed by assigning handler addresses corresponding to each interrupt/exception.

This group of handler addresses is called an interrupt/exception table, which is located in the internal ROM area. When an interrupt/exception request is acknowledged, execution jumps to the handler address and the program written in that memory is executed.

For detailed list of the interrupt/exception sources and the corresponding handler addresses, refer to *Table 7-1, "Interrupt/exception source list," on page 172*.

(2) Internal RAM area

An area of 60 KB from FFF0000$_H$ to FFFEFFF$_H$ is reserved for the internal RAM area.

In case of µPD70F3187 internal RAM of 32 KB are physically provided at addresses FFF0000$_H$ to FFF7FFF$_H$ as internal RAM. The 32 KB area of 3FF0000$_H$ to 3FF7FFF$_H$ can be seen as an image of FFF0000$_H$ to FFF7FFF$_H$.

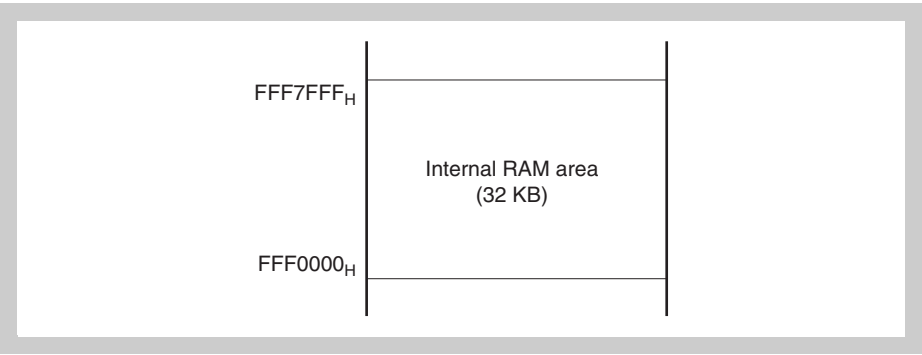


Figure 3-11 Internal RAM Area of µPD70F3187

In case of μPD70F3447 internal RAM of 24 KB are physically provided at addresses $FFF0000_H$ to $FFF5FFF_H$ as internal RAM. The 32 KB area of $3FF0000_H$ to $3FF5FFF_H$ can be seen as an image of $FFF0000_H$ to $FFF5FFF_H$.

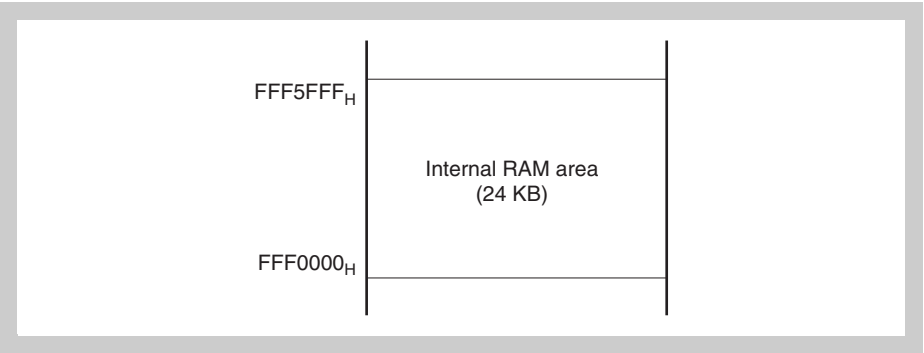


Figure 3-12 Internal RAM Area of μPD70F3447

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

A 4 KB area from $FFFF000_H$ to $FFFFFFF_H$ is provided as the on-chip peripheral I/O area. An image of addresses $FFFF000_H$ to $FFFFFFF_H$ can be seen at addresses $3FFF000_H$ to $3FFFFFF_H$

Note Addresses $3FFF000_H$ to $3FFFFFF_H$ are access-prohibited. To access the on-chip peripheral I/O, specify addresses $FFFF000_H$ to $FFFFFFF_H$.

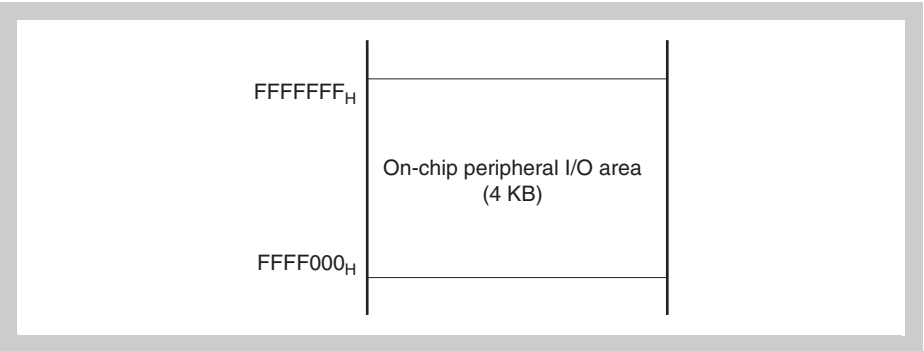


Figure 3-13 On-Chip Peripheral I/O Area

Peripheral I/O registers assigned with functions such as on-chip peripheral I/O operation mode specification and state monitoring are mapped to the on-chip peripheral I/O area. Program fetches are not allowed in this area.

Caution

1. For registers in which byte access is possible, if half-word access is executed, the higher 8 bits become undefined during a read operation, and the lower 8 bits of data are written to the register during a write operation. Do not access an 8-bit register in half-word units.
2. Addresses that are not defined as registers are reserved for future expansion. If these addresses are accessed, the operation is undefined and not guaranteed.

3.4.6 Programmable peripheral I/O area

In the V850E/PH2, the 16 KB area of $x0000_H$ to $x3FFF_H$ is provided as a programmable peripheral I/O area. In this area, the area between $x0000_H$ and $x08FF_H$ is used exclusively for the CAN controllers (CAN0, CAN1[Note]).

The internal bus of the V850E/PH2 becomes active when the on-chip peripheral I/O register area ($FFFF000_H$ to $FFFFFFF_H$) or the programmable peripheral I/O register area ($xxxxm000_H$ to $xxxxnFFF_H$) is accessed (m = xx00B, n= xx11B). However, the on-chip peripheral I/O area is allocated to the last 4 KB of the programmable peripheral I/O register area. Note that when data is written to this area, the written contents are reflected on the on-chip peripheral I/O area. Therefore, access to this area is prohibited. To access the on-chip peripheral I/O area, be sure to specify addresses $FFFF000_H$ to $FFFFFFF_H$.

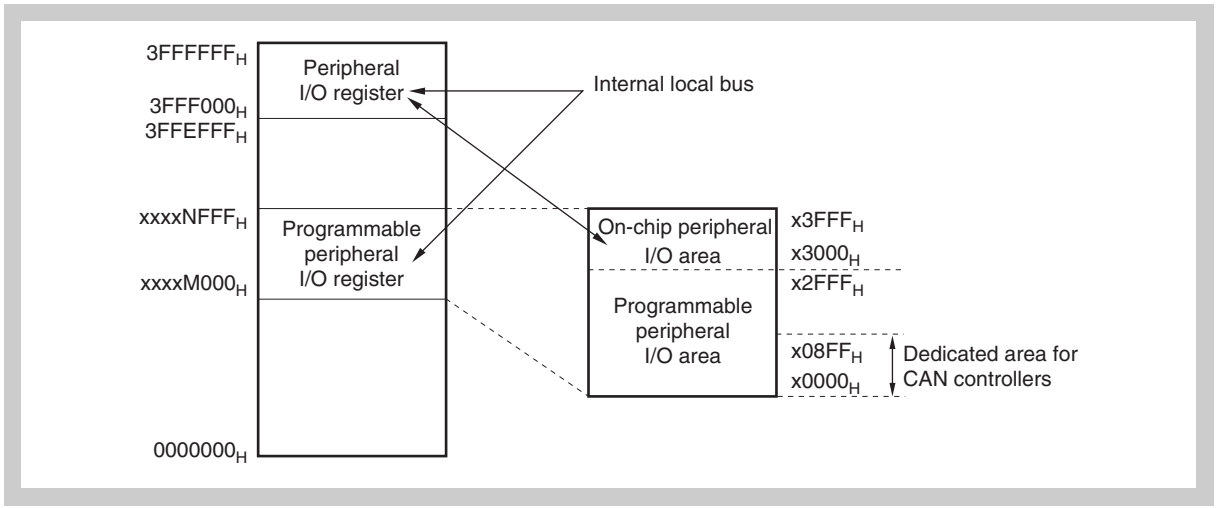


Figure 3-14 Programmable Peripheral I/O Area (Outline)

Remark M = xx00B, N = M + 11B, P= M + 10B

Caution
1. It is recommended to locate the programmable peripheral area in the first 32 Mbyte of the physical adress range.
2. The programmable peripheral area is not allowed to overlap the ROM or RAM areas. Therefore BPC must be initialized with a value in the range 0040_H to $0FFB_H$.

Note CAN1 not available for µPD70F3447.

(1) Peripheral area selection control register (BPC)

The peripheral area selection control register (BPC) is used to select a programmable peripheral I/O register area where the registers of the CAN controller are allocated.

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

Address $FFFFF064_H$

Initial value 0000_H. This register is cleared by any reset.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BPC	PA15	0	PA13	PA12	PA11	PA10	PA9	PA8	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
	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 3-11 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 recommended value of the BPC register to enable the programmable peripheral I/O area is $87FF_H$. This setting assigns the programmable peripheral I/O area to addresses from $1FFC000_H$ to $1FFFFFF_H$.

3.4.7 Specific registers

Specific registers are registers that prevent invalid data from being written if an inadvertent program behaviour occurs.

The V850E/PH2 has the following specific registers:

- Port registers 5 and 6 (P5, P6)
- Port mode registers 5 and 6 (PM5, PM6)
- Port mode control registers 5 and 6 (PMC5, PMC6)
- Port emergency shut off control registers 5 and 6 (PESC5, PESC6)
- Port emergency shut off status registers 5 and 6 (ESOST5, ESOST6)

Moreover, there is also a command register (PRCMD), which is a protection register against write operations to the specific registers. Write access to the specific registers is performed with a special sequence and illegal store operations are notified to the system status register (PHS).

This section of the manual describes the access method to these specific registers, rather than the values that can be written to these registers. For details on these register values, please refer to sections *"Port 5" on page 886* and *"Port 6" on page 891*.

(1) Setting data to specific registers

Setting data to a specific registers is done in the following sequence.

<1> Prepare the data to be set to the special register in a general-purpose register.

<2> Write the data prepared in <1> to the command register (PRCMD).

<3> Write the data to the specific register (using the following instructions).

- Store instruction (ST/SST instruction)
- Bit manipulation instruction (SET1/CLR1/NOT1 instruction)

Example

```
<1>  MOV 0x02, r10              ; Prepare data in r10
<2>  ST.B r10, PRCMD[r0]        ; Write PRCMD register
<3>  ST.B r10, P5               ; Set P5 register
(next instruction)
```

Caution

1. Interrupts are not acknowledged when executing the store instruction to the PRCMD register.
   If another instruction is placed between steps <2> and <3>, the correct sequence may not be realized if an interrupt is acknowledged for that instruction, resulting in the writing to the protected register to be not done, and an error to be stored in the PRERR bit of the PHS register.
2. If there is a possibility of an active DMA register before <2> and <3>, the specific register may not be written. In this case, ensure that no DMA register is active during the sequence <2> to <3>, or repeat the sequencer <2> to <3> as long as the PRERR bit of the PHS register is set to <1>.

(2) Processor command register (PRCMD)

The PRCMD register is an 8-bit register used to prevent data from being written to registers that may have a large influence on the system, possibly causing the application system to unexpectedly stop. Only the first write operation to a specific register following the execution of a write operation to the PRCMD register, is valid.

As a result, register values can be overwritten only using a preset sequence, preventing invalid write operations.

PRCMD register must be written with store instruction execution by CPU only (not with DMA transfer). If an illegal store operation to a command or specific register takes place, it is reported by the PRERR flag of the system status register (PHS).

Access This register can be written in 8-bit units only. Undefined data is read from this register.

Address FFFFF1FCH

Initial value Undefined

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

(3) System status register (PHS)

The PHS register is an 8-bit register to which the PRERR flag showing the generation of protection errors is assigned.

If a write operation to a specific register has not been executed in the correct sequence including the access to the command register (PRCMD), the write operation to the intended register is not executed, a protection error is generated and the PRERR flag is set to 1.

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

Address FFFFF802H

Initial value 00H. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
PHS	0	0	0	0	0	0	0	PRERR
	R	R	R	R	R	R	R	R/W

Table 3-12 PHS register contents

Bit Position	Bit Name	Function
0	PRERR	Write error status. 0: Protection error did not occur 1: Protection error occurred **Note:** This bit can be cleared by writing 0 to it. Setting this bit to 1 by software is not possible.

Caution

1. If 0 is written to the PRERR bit of the PHS register (that is not a specific register) immediately following write to the PRCMD register, the PRERR bit becomes 0 (write priority).
2. If data is written to the PRCMD register (that is not a specific registers) immediately following write to the PRCMD register, the PRERR bit becomes 1.

The PRERR flag operates under the following conditions.

(a) Setting condition (PRERR flag = 1)

- When a write operation is not performed on the PRCMD register and an operation to write a specific register is performed (when <4> in the example **3.4.8 (1) Setting data to specific registers** is executed without <3>).
- If a write operation (including a bit manipulation instruction) is performed on an on-chip peripheral I/O register other than a specific register after a write operation to the PRCMD register (when <4> in the example **3.4.8 (1) Setting data to specific registers** is not performed for a specific register).

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

(b) Clearing condition (PREER flag = 0)

- When 0 is written to the PRERR flag of the PHS register.
- When system reset is executed.

3.4.8 System wait control register (VSWC)

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

Access to on-chip peripheral I/O registers is made in 3 clocks (without wait), however, wait states may be required depending on the operation frequency. After $\overline{\text{RESET}}$ the maximum number wait states are set, therefore intialize the the VSWC register during the start up procedure with value according to *Table 3-13* below.

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

Address FFFF F06EH

Initial value 77H.

Table 3-13 Initialization of VSWC register

Register Name	Operating frequency	Set Value
VSWC	64 MHz	13H

3.4.9 DMA wait control registers 0 and 1 (DMAWC0, DMAWC1)

The DMA wait control registers 0 and 1 (DMAWC0, DMAWC1) are a registers that control the bus access wait and signal timing for DMA transfers.

After $\overline{\text{RESET}}$ the maximum number wait states are set, therefore intialize the the DMAWC0 and DMAWC1 registers during the start up procedure with values according to the operating frequency in *Table 3-14* below

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

Address DMAWC0: FFFF FE00H
DMAWC1: FFFF FE00H

Initial value DMAWC0: 37H
DMAWC1: 07H

Table 3-14 TInitialization of DMAWC0 and DMAWC1 registers

Register Name	Operating frequency	Set Value
DMAWC0	64 MHz	13H[a]
DMAWC1		04H

a) same value as VSWC register

3.4.10 Cautions

Initialize the following registers immediately after reset signal release in the following sequence:

- System wait control register (VSWC)
  (refer to *“System wait control register (VSWC)” on page 96*))
- DMA wait control registers 0 and 1 (DMAWC0,DMAWC1)
  (refer to *“DMA wait control registers 0 and 1 (DMAWC0, DMAWC1)” on page 96*))

Chapter 4 Bus Control Function (BCU)

The µPD70F3187 is provided with an external bus interface function by which external memories, such as ROM and SRAM, and external I/O can be connected.

Note The external bus interface function is not available on µPD70F3447.

4.1 Features

- 32-bit/16-bit/8-bit data bus sizing function
- 8 chip areas select function
- 4 chip area select signals externally available ($\overline{CS0}$, $\overline{CS1}$, $\overline{CS3}$ and $\overline{CS4}$)
- Wait function
- Programmable wait function, capable of inserting up to 7 wait states for each memory block
- External wait function via $\overline{WAIT}$ pin
- Idle state insertion function
- External device connection can be enabled via bus control/port alternate function pins
- Programmable Endian format (Little Endian/Big Endian)

4.2 Bus Control Pins

The following pins are used for connecting to external devices.

Table 4-1 External bus control pins and corresponding port pins

Bus Control Pin (Function when in Control Mode)	Function when in Port Mode	Register for Port/ Control Mode Switching[a]
Data bus (D0 to D15)	PDL0 to PDL15 (Port DL)	PMCDL
Data bus (D16 to D31)	PDH0 to PDH15 (Port DH)	PMCDH
Address bus (A0 to A15)	PAL0 to PAL15 (Port AL)	PMCAL
Address bus (A16 to A21)	PAH0 to PAH5 (Port AH)	PMCAH
Chip select ($\overline{CS0}$, $\overline{CS1}$, $\overline{CS3}$ and $\overline{CS4}$)	PCS0, PCS1, PCS3 and PCS4 (Port CS)	PMCCS
Read/write control ($\overline{RD}$,$\overline{WR}$)	PCT4, PCT5 (Port CT)	PMCCT
Byte enable control ($\overline{BE0}$ to $\overline{BE3}$)	PCD2 to PCD5 (Port CD)	PMCCD
External wait control ($\overline{WAIT}$)	PCM0 (Port CM)	PMCCM

a) Even if the port mode control registers for the µPD70F3447 exist, it is prohibited to write other values to these registers than the reset values.

4.3 Memory Block Function

The 64 MB memory space is divided into memory blocks of 2 MB, 4 MB, and 8 MB units.

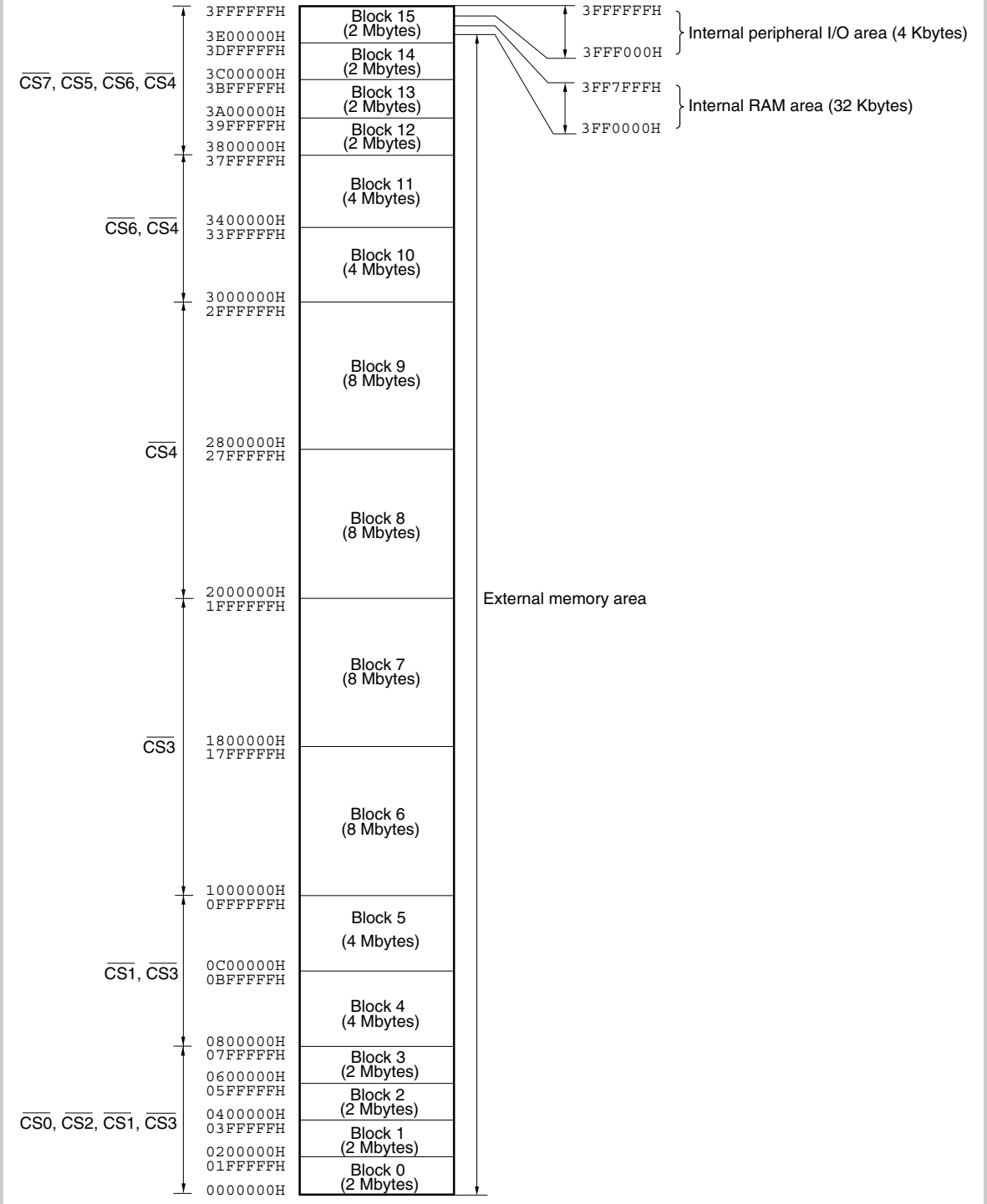


Figure 4-1 Memory block function

4.3.1 Chip select control function

The 64 MB memory area can be divided into 2 MB, 4 MB and 8 MB memory blocks by the chip area selection control registers 0 and 1 (CSC0, CSC1) to control the chip select signals.

The memory area can be effectively used by dividing the memory area into memory blocks using the chip select control function.

Note If different chip area select signals are set to the same block, the priority order is controlled as follows.

CSC0: Peripheral I/O area > $\overline{CS0}$ > $\overline{CS2}$ > $\overline{CS1}$ > $\overline{CS3}$

CSC1: Peripheral I/O area > $\overline{CS7}$ > $\overline{CS5}$ > $\overline{CS6}$ > $\overline{CS4}$

(1) Chip area selection control registers 0, 1 (CSC0, CSC1)

Access These registers can be read/written in 16-bit units.

Address CSC0: FFFF F060H
CSC1: FFFF F062H

Initial Value 2C11H

CSC0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CS33	CS32	CS31	CS30	CS23	CS22	CS21	CS20	CS13	CS12	CS11	CS10	CS03	CS02	CS01	CS00
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
$\overline{CS3}$				$\overline{CS2}$				$\overline{CS1}$				$\overline{CS0}$			

CSC1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CS43	CS42	CS41	CS40	CS53	CS52	CS51	CS50	CS63	CS62	CS61	CS60	CS73	CS72	CS71	CS70
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
$\overline{CS4}$				$\overline{CS5}$				$\overline{CS6}$				$\overline{CS7}$			

Caution Not all the chip area select signals are externally available on output pins. Even so, enabling chip area select signals other than $\overline{CS0}$, $\overline{CS1}$, $\overline{CS3}$ or $\overline{CS4}$, the setting for the corresponding memory blocks will be effective too, regardless of an external chip select output pin.

Table 4-2 CSC0 register contents

Bit Position	Bit Name	Function	Chip Select Signal
15	CS33	$\overline{CS3}$ active during block 7 access.	$\overline{CS3}$
14	CS32	$\overline{CS3}$ active during block 6 access.	
13	CS31	$\overline{CS3}$ active during block 4 or 5 access.	
12	CS30	$\overline{CS3}$ active during block 0, 1, 2, or 3 access.	
11	CS23	$\overline{CS2}$ active during block 3 access.	$\overline{CS2}$
10	CS22	$\overline{CS2}$ active during block 2 access.	
9	CS21	$\overline{CS2}$ active during block 1 access.	
8	CS20	$\overline{CS2}$ active during block 0 access.	
7	CS13	$\overline{CS1}$ active during block 5 access.	$\overline{CS1}$
6	CS12	$\overline{CS1}$ active during block 4 access.	
5	CS11	$\overline{CS1}$ active during block 2 or 3 access.	
4	CS10	$\overline{CS1}$ active during block 0 or 1 access.	
3	CS03	$\overline{CS0}$ active during block 3 access.	$\overline{CS0}$
2	CS02	$\overline{CS0}$ active during block 2 access.	
1	CS01	$\overline{CS0}$ active during block 1 access.	
0	CS00	$\overline{CS0}$ active during block 0 access	

Table 4-3 CSC1 register contents

Bit Position	Bit Name	Function	Chip Select Signal
	CS43	$\overline{CS4}$ active during block 8 access.	$\overline{CS4}$
	CS42	$\overline{CS4}$ active during block 9 access.	
	CS41	$\overline{CS4}$ active during block 10 or 11 access.	
	CS40	$\overline{CS4}$ active during block 12, 13, 14, or 15 access.	
	CS53	$\overline{CS5}$ active during block 12 access.	$\overline{CS5}$
	CS52	$\overline{CS5}$ active during block 13 access.	
	CS51	$\overline{CS5}$ active during block 14 access.	
	CS50	$\overline{CS5}$ active during block 15 access.	
	CS63	$\overline{CS6}$ active during block 10 access.	$\overline{CS6}$
	CS62	CS6 active during block 11 access.	
	CS61	$\overline{CS6}$ active during block 12 or 13 access.	
	CS60	$\overline{CS6}$ active during block 14 or 15 access.	
	CS73	$\overline{CS7}$ active during block 12 access.	$\overline{CS7}$
	CS72	$\overline{CS7}$ active during block 13 access.	
	CS71	$\overline{CS7}$ active during block 14 access.	
	CS70	$\overline{CS7}$ active during block 15 access.	

Note Dedicated chip select operation is enabled when corresponding CSnm bit is set (1), and disabled when CSnm is cleared (0) (n = 0 to 7, m = 0 to 3)

4.4 Bus Cycle Type Control Function

In the V850E/PH2, the following external devices can be connected directly to each memory block.

- SRAM, external ROM, external I/O

Connected external devices are specified by the bus cycle type configuration registers 0, 1 (BCT0, BCT1).

(1) Bus cycle configuration registers 0, 1 (BCTn, BCT1)

Access These registers can be read/written in 16-bit units.

Address BCT0: FFFF F480H
BCT1: FFFF F482H

Initial Value CCCCH

BCT0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ME3	1	0	0	ME2	1	0	0	ME1	1	0	0	ME0	1	0	0
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
			$\overline{CS3}$				$\overline{CS2}$				$\overline{CS1}$				$\overline{CS0}$

BCT1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ME7	1	0	0	ME6	1	0	0	ME5	1	0	0	ME4	1	0	0
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
			$\overline{CS7}$				$\overline{CS6}$				$\overline{CS5}$				$\overline{CS4}$

Table 4-4 BCT0, BCT1 registers contents

Bit position	Bit name	Function
15, 11, 7, 3	MEn	Enables/disables Memory Controller operation for chip select area n ($\overline{CSn}$). 0: Operation disabled 1: Operation enabled

Caution

1. Write to the BCT0 and BCT1 registers after reset, and then do not change the set value. Also, do not access an external memory area other than that for this initialization routine until initial setting of the BCT0 and BCT1 registers is finished. However, it is possible to access external memory areas whose initialization has been finished.
2. The bits marked as 0 and 1 are reserved. The values of these bits must not be changed. Otherwise the operation of the external bus interface cannot be ensured.

4.5 Bus Access

4.5.1 Number of access clocks

The number of basic clocks necessary for accessing each resource is as follows.

Table 4-5 Number of bus access clocks

Resources (Bus width) / Bus Cycle Configuration		Internal RAM (32 bits)	Peripheral I/O (16 bits)	External memory (16 bits)
Instruction fetch	Normal access	1[a]	-	2[b]
	Branch	1	-	2[b]
Operand data access		1	3[b]	2[b]

a) The instruction fetch becomes 2 clocks, in case of contention with data access.
b) This is the minimum value.

4.5.2 Bus sizing function

The bus sizing function controls data bus width for each CS area. The data bus width is specified by using the bus size configuration register (BSC).

(1) Bus size configuration register (BSC)

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

Address FFFF F066H

Initial Value AAAAH

Caution Write to the BSC register after reset, and then do not change the set value. Also, do not access an external memory area other than that for this initialization routine until initial setting of the BSC register is finished. However, it is possible to access external memory areas whose initialization has been finished.

BSC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS71	BS70	BS61	BS60	BS51	BS50	BS41	BS40	BS31	BS30	BS21	BS20	BS11	BS10	BS01	BS00
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
		$\overline{CS7}$		$\overline{CS6}$		$\overline{CS5}$		$\overline{CS4}$		$\overline{CS3}$		$\overline{CS2}$		$\overline{CS1}$		$\overline{CS0}$	

Table 4-6 BSC register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>15 to 0</td><td>BEn[1:0]</td><td>Specifies the data bus width of chip select area n ($\overline{CSn}$)
<table>
<tr><th>BEn1</th><th>BEn0</th><th>Data Bus Width of $\overline{CSn}$ Area</th></tr>
<tr><td>0</td><td>0</td><td>8 bits</td></tr>
<tr><td>0</td><td>1</td><td>16 bits</td></tr>
<tr><td>1</td><td>0</td><td>32 bits</td></tr>
<tr><td>1</td><td>1</td><td>Setting prohibited</td></tr>
</table>
</td></tr>
</table>

4.5.3 Endian control function

The Endian control function can be used to set processing of word data in memory either by the Big Endian method or the Little Endian method for each CS area selected with the chip select signal ($\overline{CS0}$ to $\overline{CS7}$). Switching of the Endian method is specified with the Endian configuration register (BEC).

"Big endian" means that the high-order byte of the word is stored in memory at the lowest address, and the low-order byte at the highest address. Therefore, the base address of the word addresses the high-order byte:

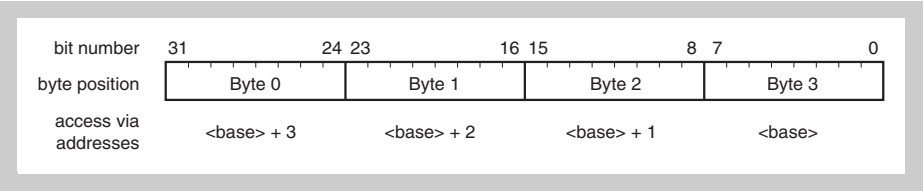


Figure 4-2 Big endian addresses within word

"Little Endian" means that the low-order byte of the word is stored in memory at the lowest address, and the high-order byte at the highest address. Therefore, the base address of the word addresses the low-order byte:

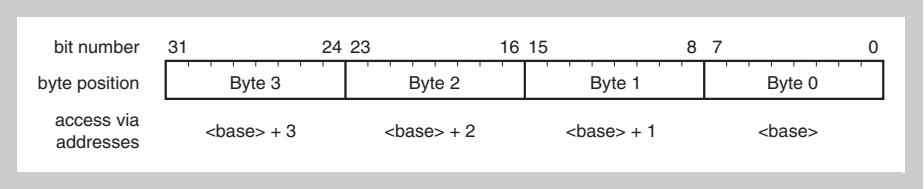


Figure 4-3 Little endian addresses within word

(1) Endian configuration register (BEC)

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

Address FFFF F068H

Initial Value 0000H

Caution

1. Bits 15, 13, 11, 9, 7, 5, 3, and 1 of the BEC register must be cleared (0). If these bits are set to 1, the operation is not guaranteed.
2. Set the $\overline{CSn}$ area specified as the programmable peripheral I/O area to Little Endian format.
3. In the following areas, the data processing method is fixed to Little Endian method. Any setting of Big Endian method for these areas according to the BEC register is invalid.
   - On-chip peripheral I/O area
   - Internal RAM area
   - Fetch area of external memory

BEC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	BE70	0	BE60	0	BE50	0	BE40	0	BE30	0	BE20	0	BE10	0	BE00
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
	$\overline{CS7}$		$\overline{CS6}$		$\overline{CS5}$		$\overline{CS4}$		$\overline{CS3}$		$\overline{CS2}$		$\overline{CS1}$		$\overline{CS0}$

Table 4-7 BEC registers contents

Bit position	Bit name	Function
14, 12, 10, 8, 6, 4, 2, 0	MEn	Controls the endian method of chip select area n ($\overline{CSn}$). 0: Little Endian method 1: Big Endian method

4.6 Data Bus Access Order

The V850E/PH2 accesses peripheral I/O and external memory in 8-bit, 16-bit, or 32-bit units. The following shows the operation for each type of access.

4.6.1 8-bit data bus access

(1) Byte access (8 bits)

(a) Little endian

Figure 4-4 Byte acces to 8-bit data bus (little endian)

(b) Big endian

Figure 4-5 Byte access to 8-bit data bus (big endian)

(2) Halfword access (16 bits)

(a) Little endian

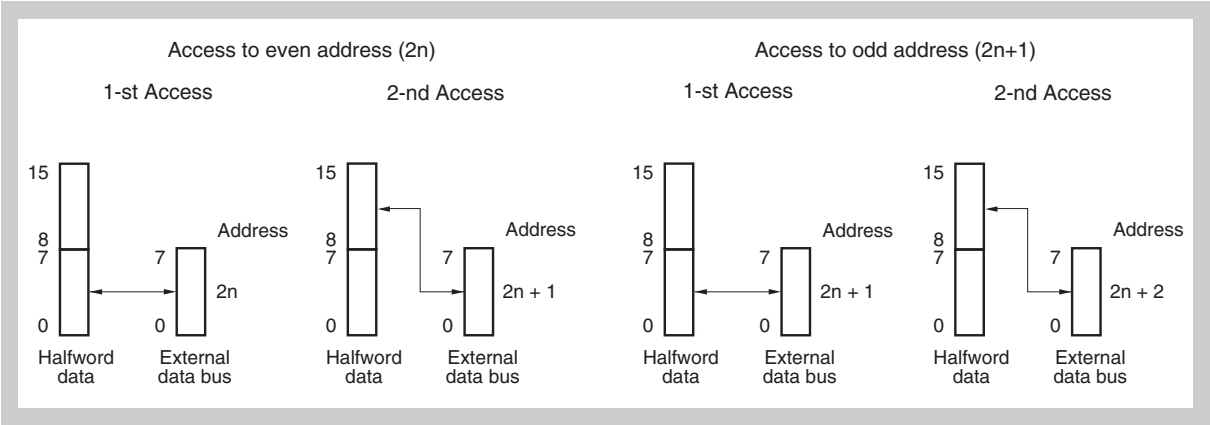


Figure 4-6 Halfword access to 8-bit data bus (little endian)

(b) Big endian

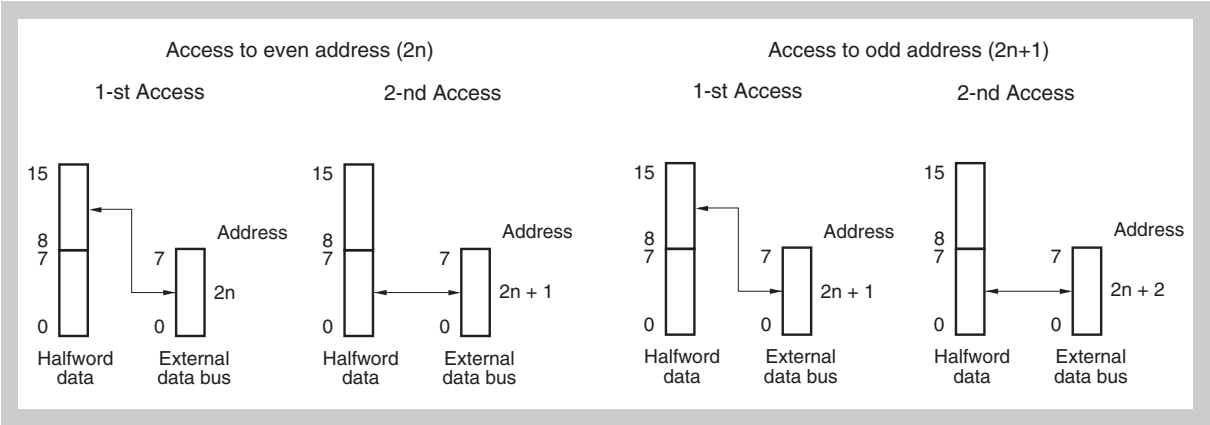


Figure 4-7 Halfword access to 8-bit data bus (big endian)

(3) Word access (32 bits)

(a) Little endian

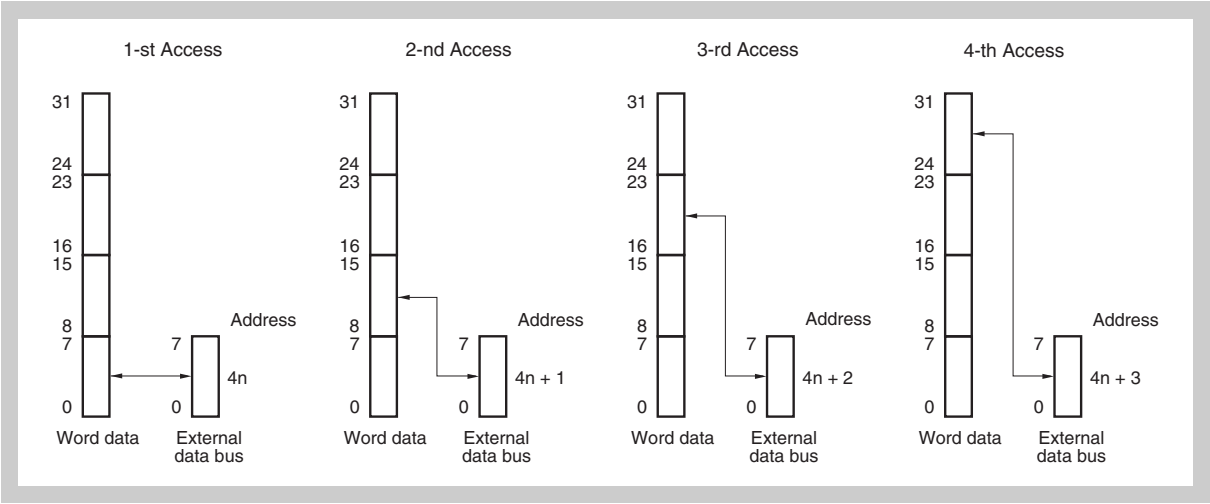


Figure 4-8 Word access to 8-bit data bus at address 4n (little endian)

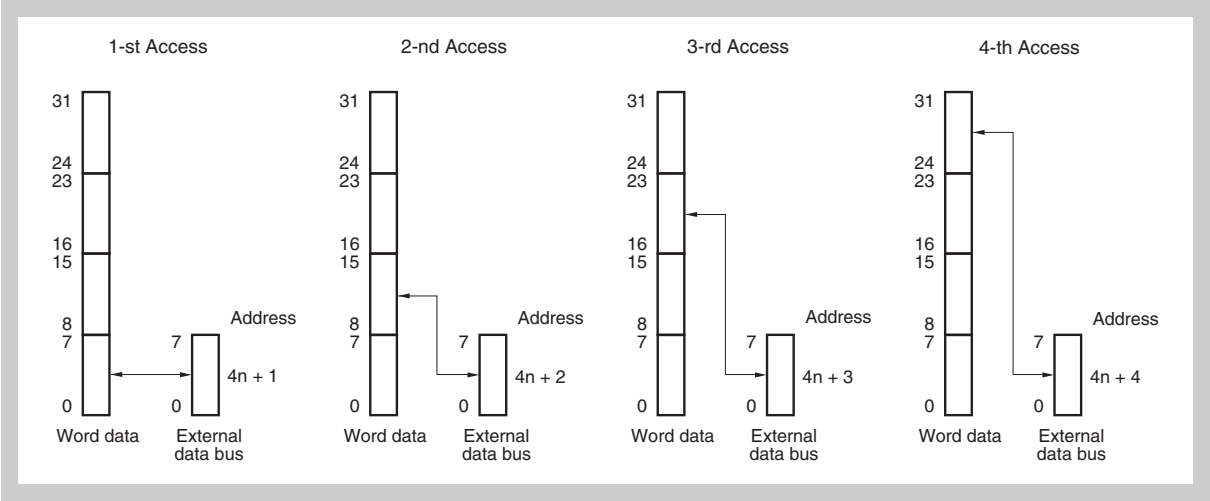


Figure 4-9 Word access to 8-bit data bus at address 4n+1 (little endian)

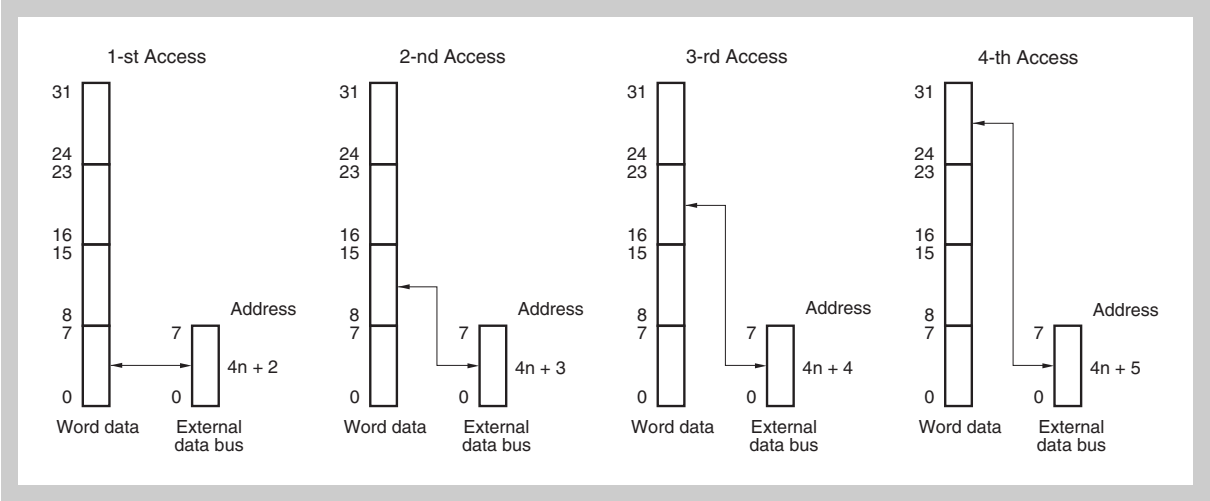


Figure 4-10 Word access to 8-bit data bus at address 4n+2 (little endian)

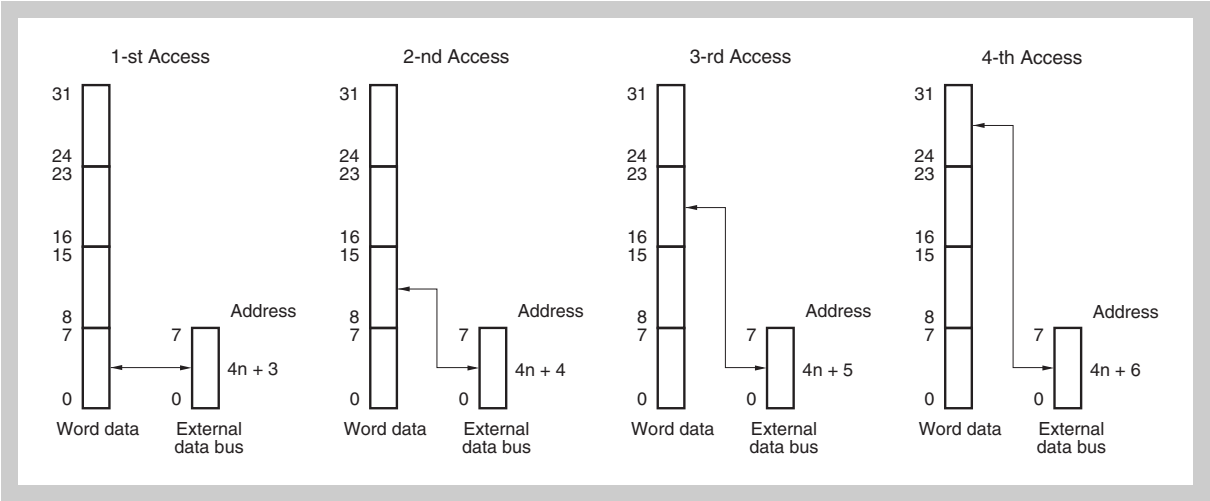


Figure 4-11 Word access to 8-bit data bus at address 4n+3 (little endian)

(b) Big endian

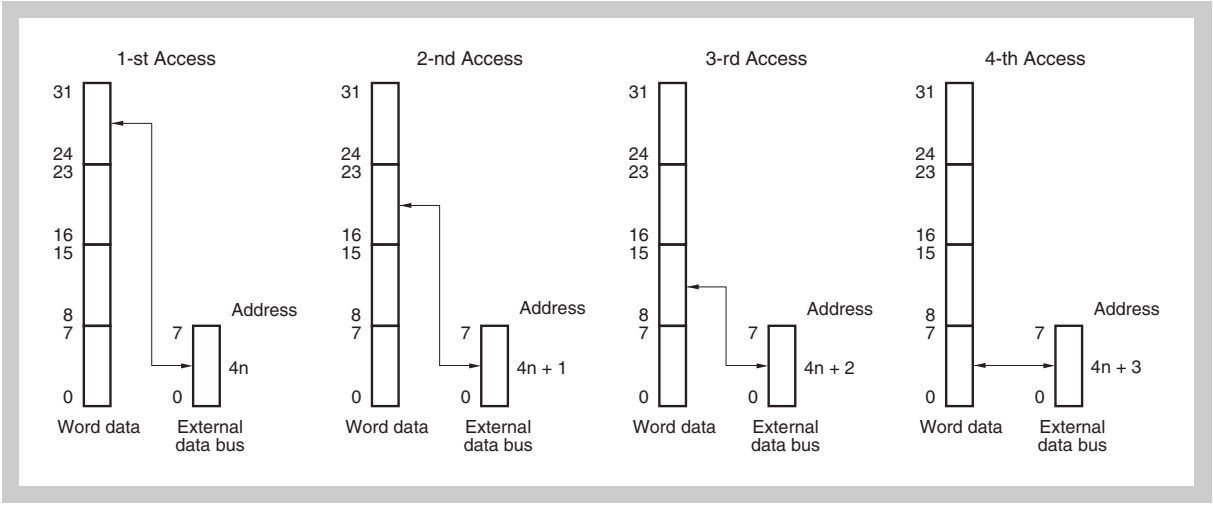


Figure 4-12 Word access to 8-bit data bus at address 4n (big endian)

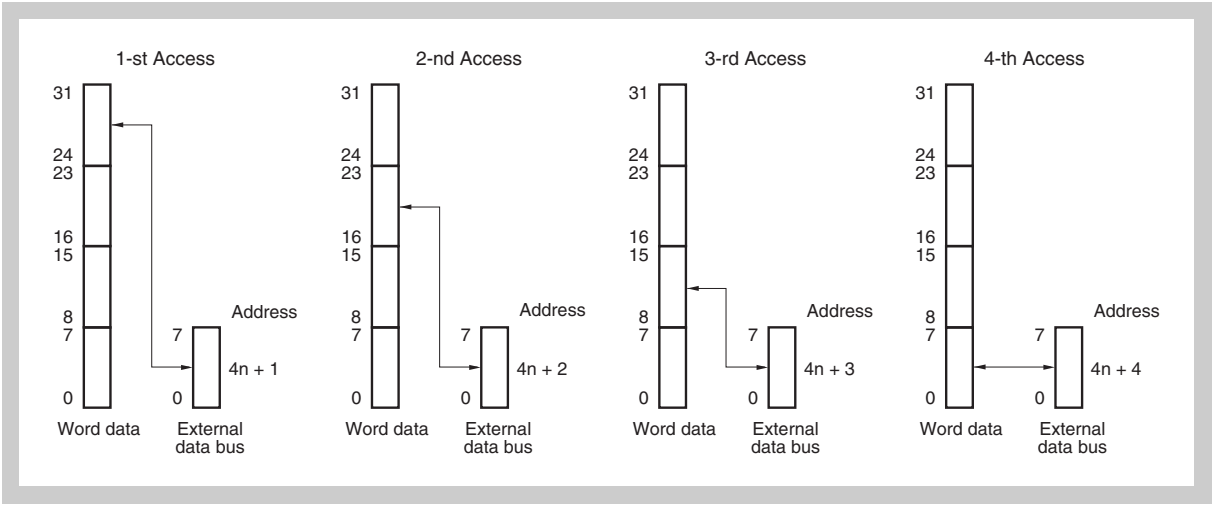


Figure 4-13 Word access to 8-bit data bus at address 4n+1 (big endian)

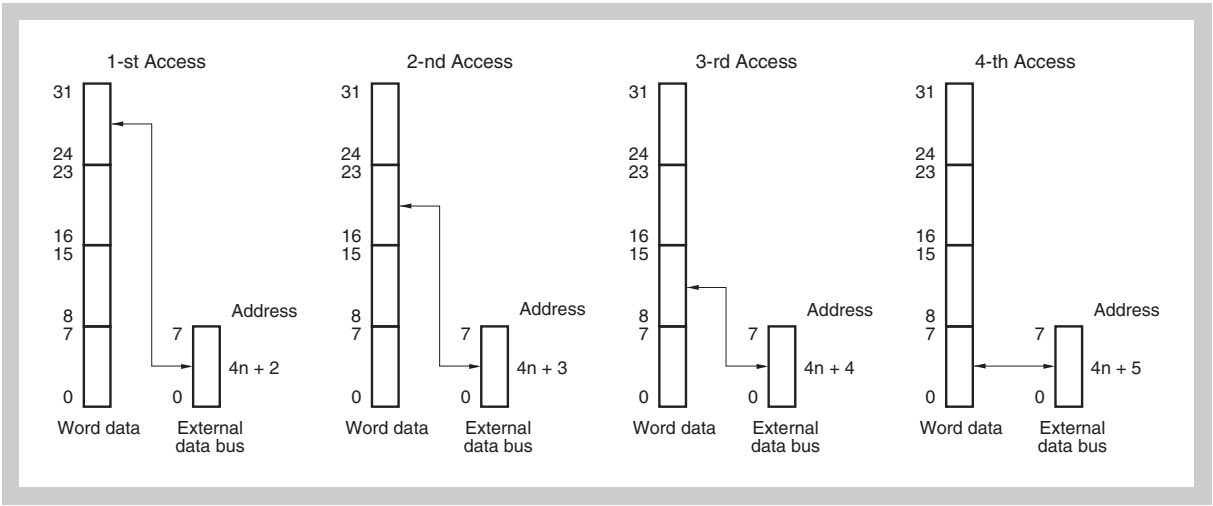


Figure 4-14 Word access to 8-bit data bus at address 4n+2 (big endian)

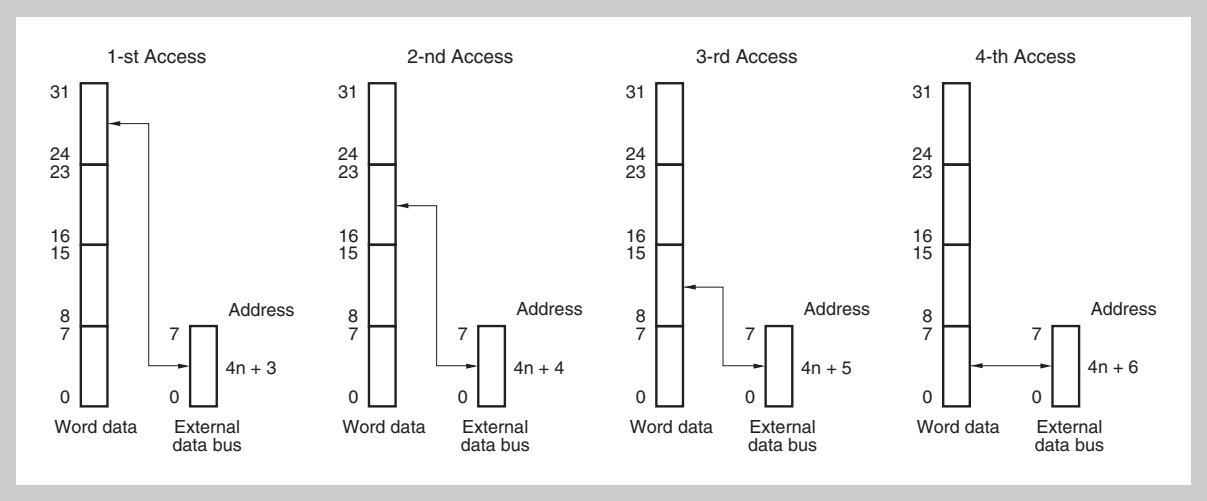


Figure 4-15 Word access to 8-bit data bus at address 4n+3 (big endian)

4.6.2 16-bit data bus access

(1) Byte access (8 bits)

(a) Little endian

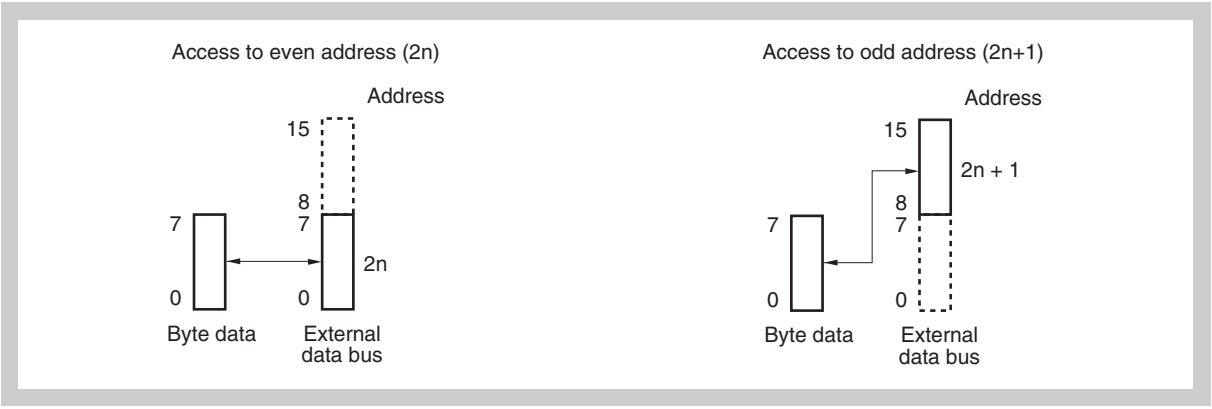


Figure 4-16 Byte access to 16-bit data bus (little endian)

(b) Big endian

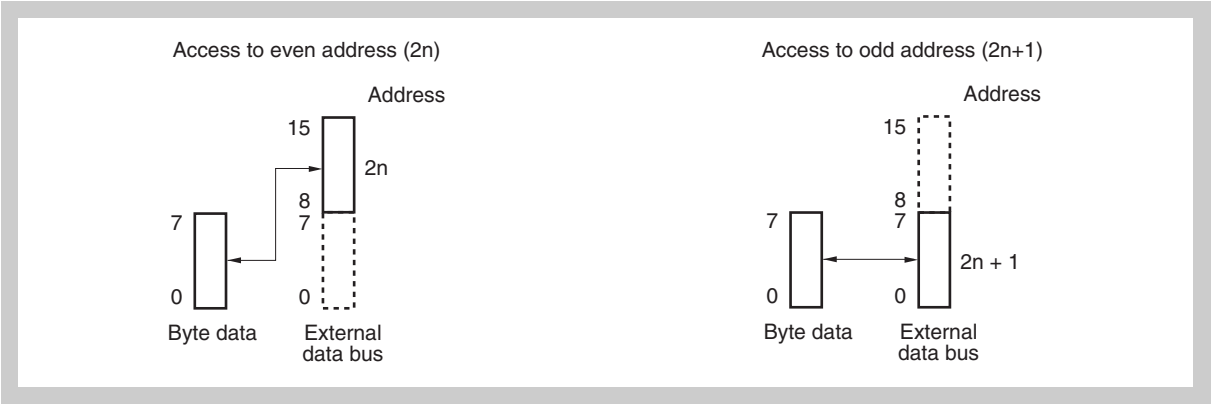


Figure 4-17 Byte access to 16-bit data bus (big endian)

(2) Halfword access (16 bits)

(a) Little endian

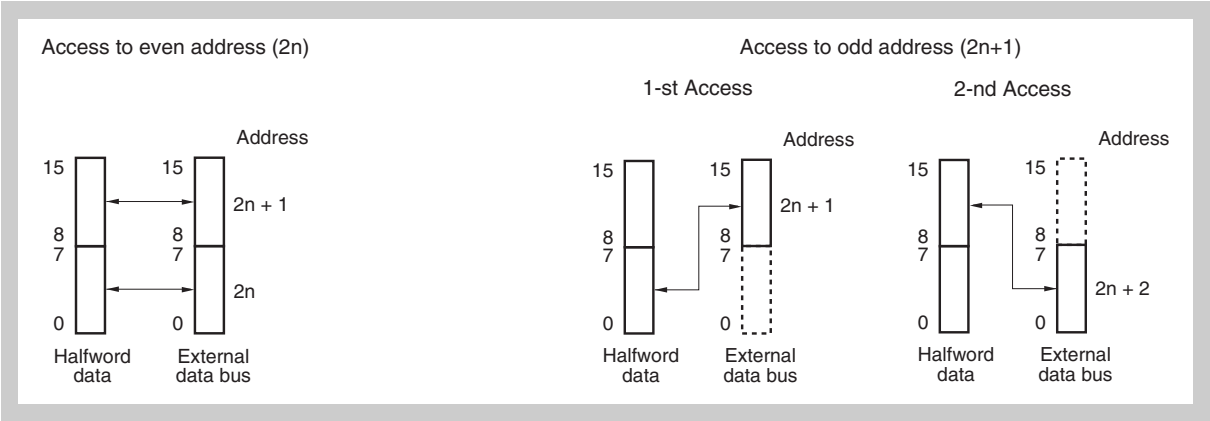


Figure 4-18 Halfword access to 16-bit data bus (little endian)

(b) Big endian

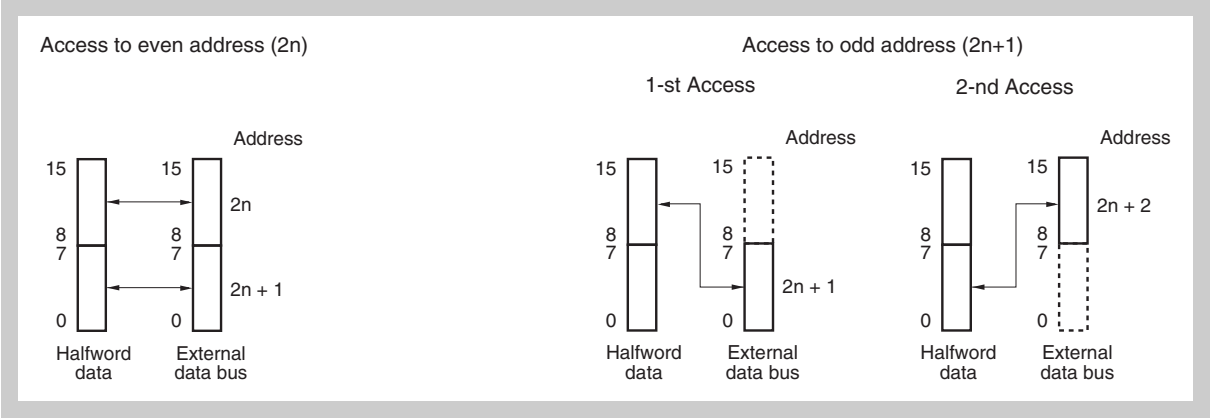


Figure 4-19 Halfword access to 16-bit data bus (big endian)

(3) Word access (32 bits)

(a) Little endian

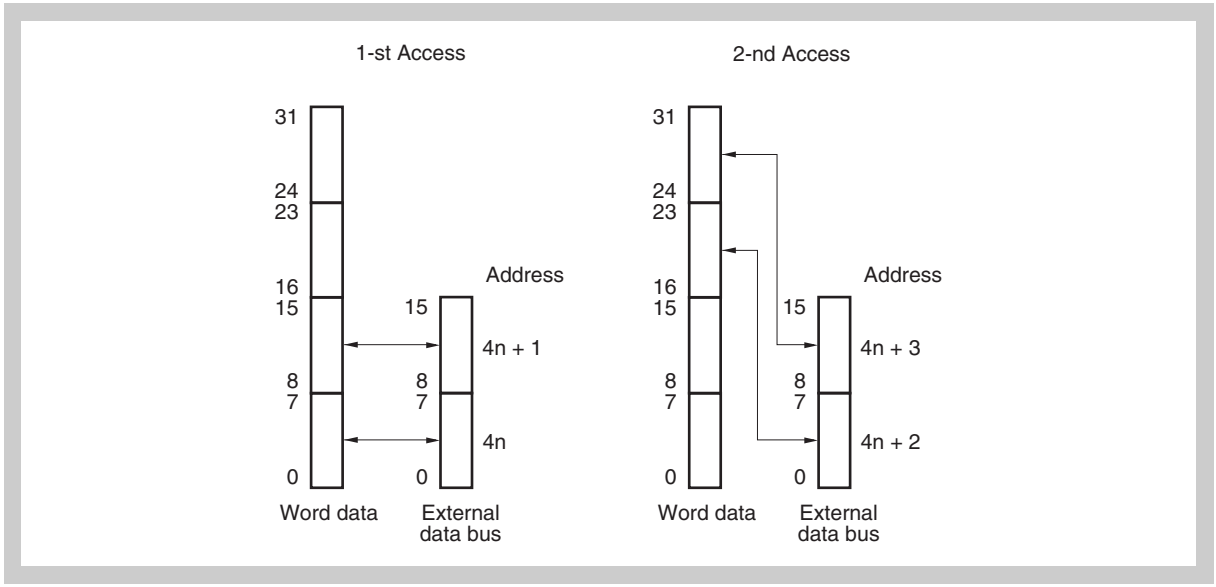


Figure 4-20 Word access to 16-bit data bus at address 4n (little endian)

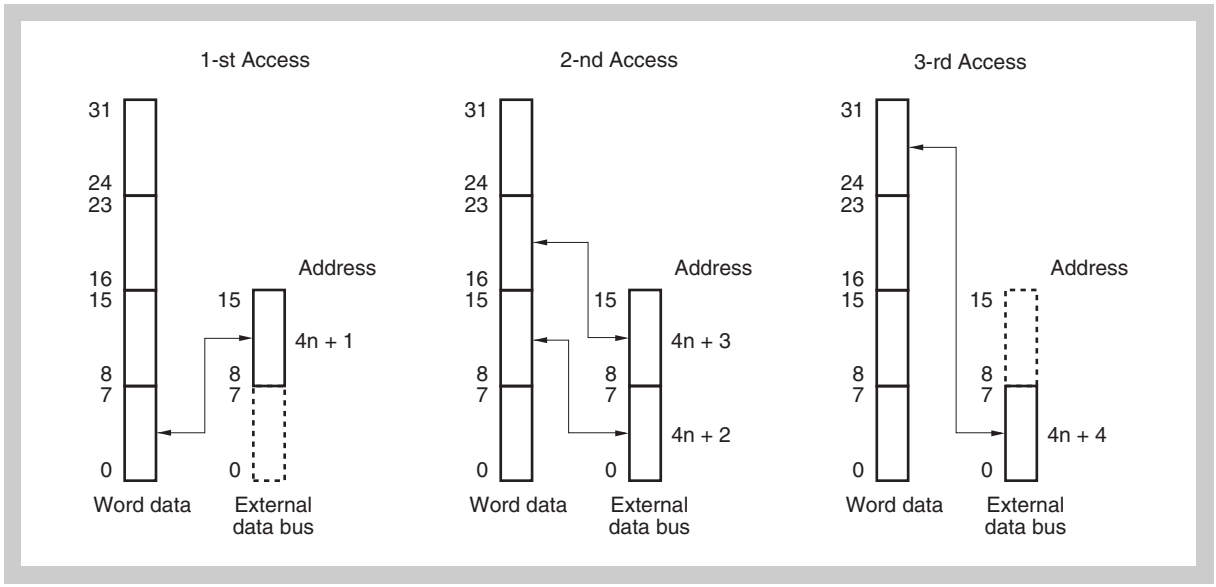


Figure 4-21 Word access to 16-bit data bus at address 4n+1 (little endian)

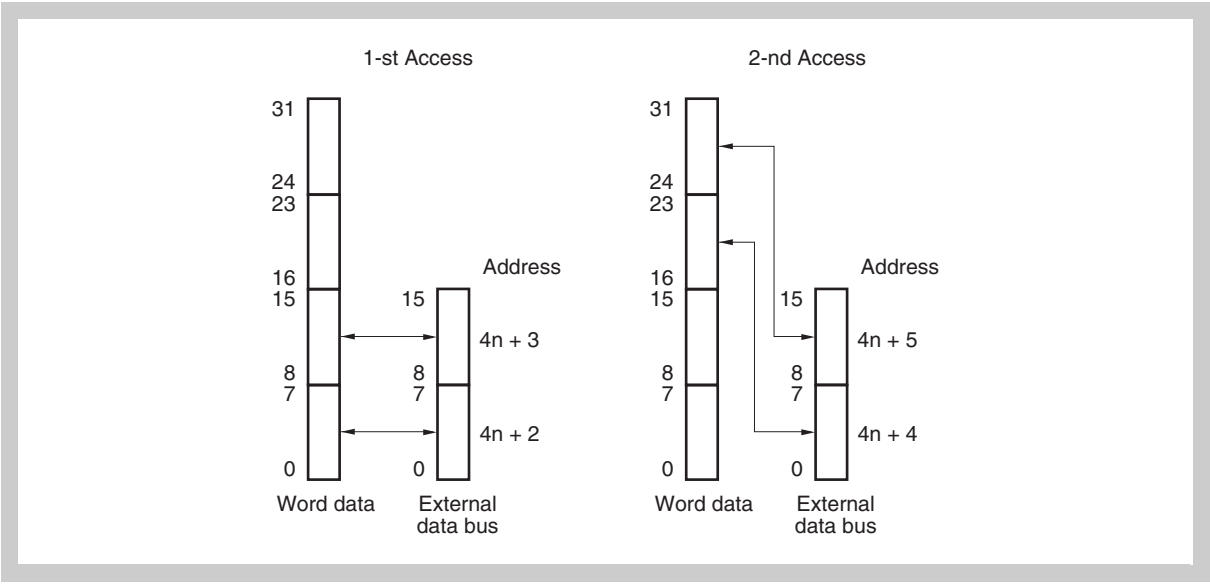


Figure 4-22 Word access to 16-bit data bus at address 4n+2 (little endian)

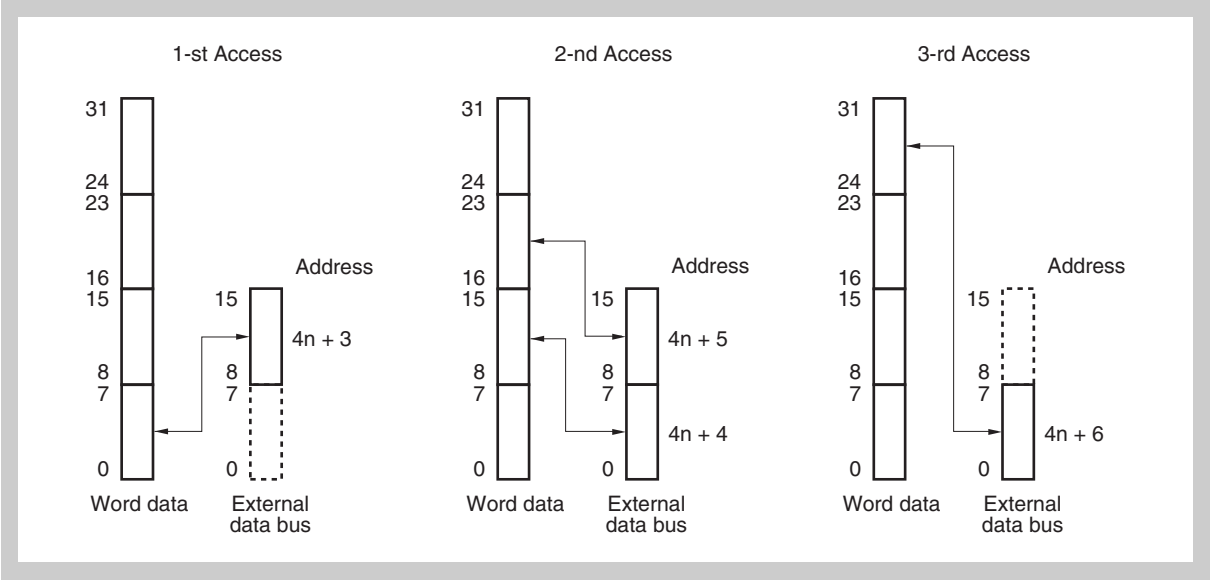


Figure 4-23 Word access to 16-bit data bus at address 4n+3 (little endian)

(b) Big endian

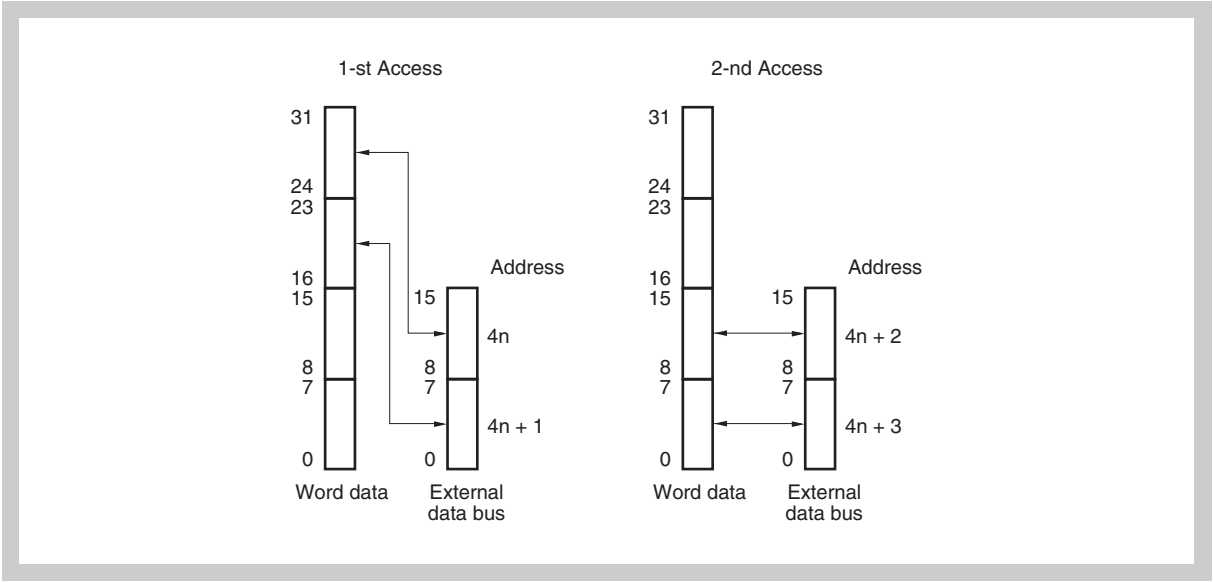


Figure 4-24 Word access to 16-bit data bus at address 4n (big endian)

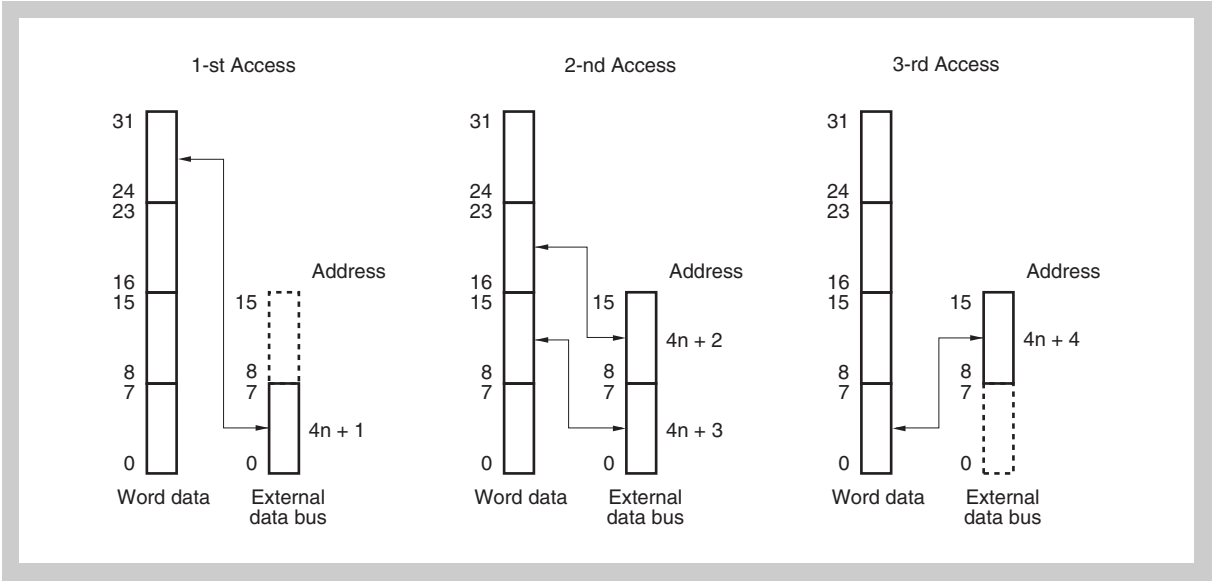


Figure 4-25 Word access to 16-bit data bus at address 4n+1 (big endian)

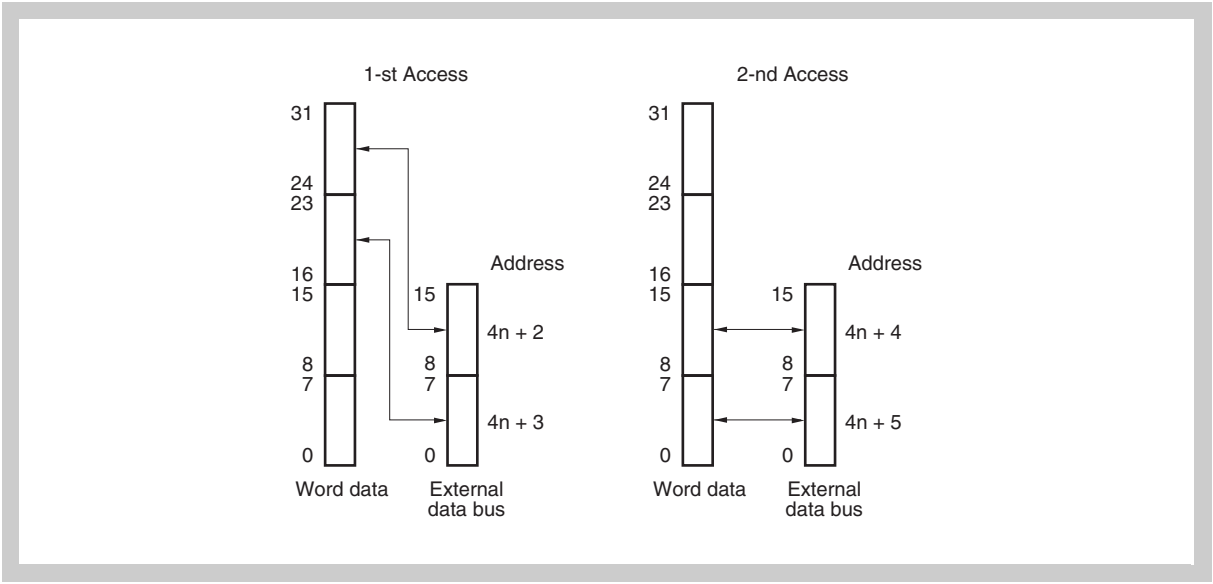


Figure 4-26 Word access to 16-bit data bus at address 4n+2 (big endian)

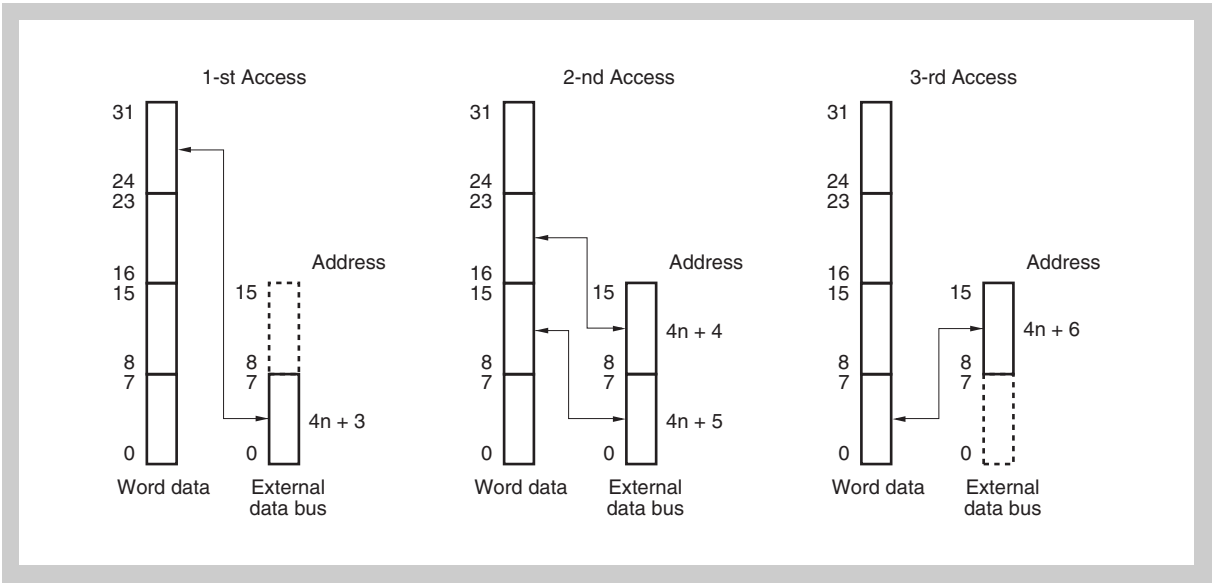


Figure 4-27 Word access to 16-bit data bus at address 4n+3 (big endian)

4.6.3 32-bit data bus

(1) Byte access (8 bits)

(a) Little endian

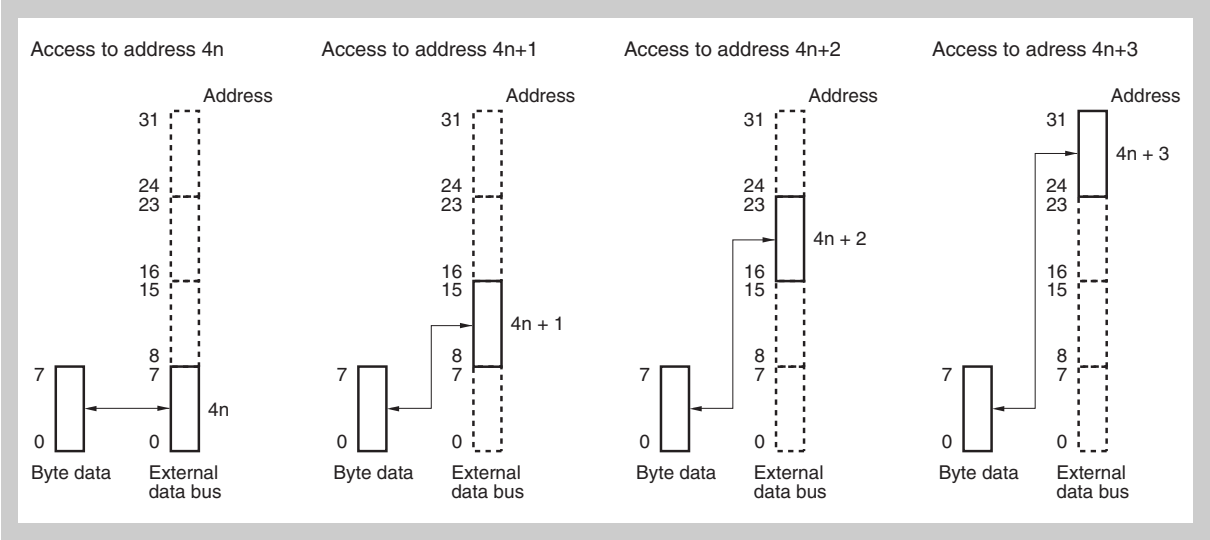


Figure 4-28 Byte access to 32-bit data bus (little endian)

(b) Big endian

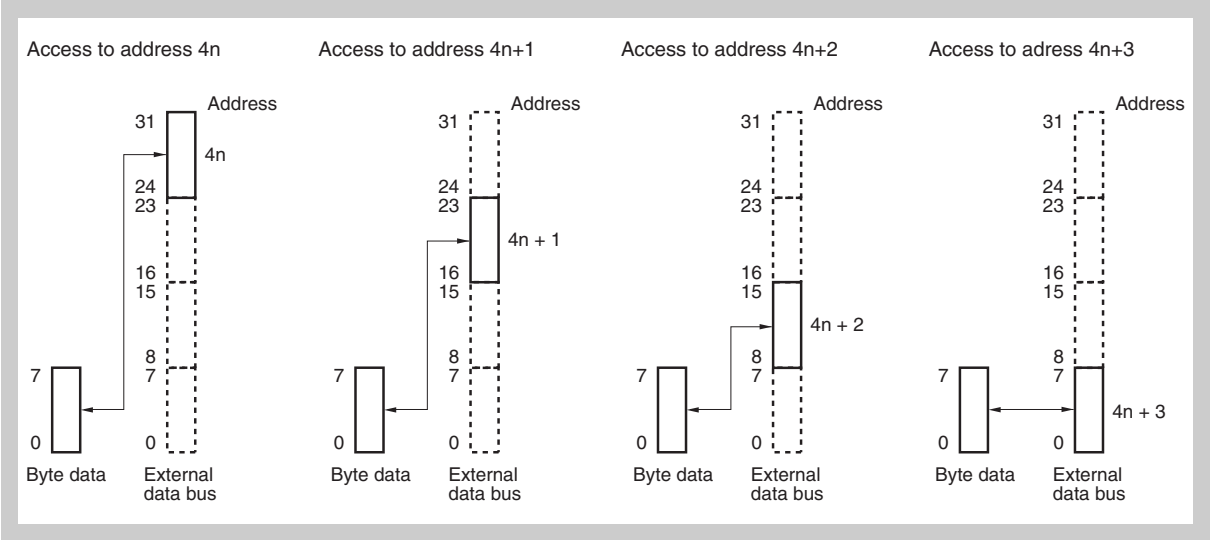


Figure 4-29 Byte access to 32-bit data bus (big endian)

(2) Halfword access (16 bits)

(a) Little endian

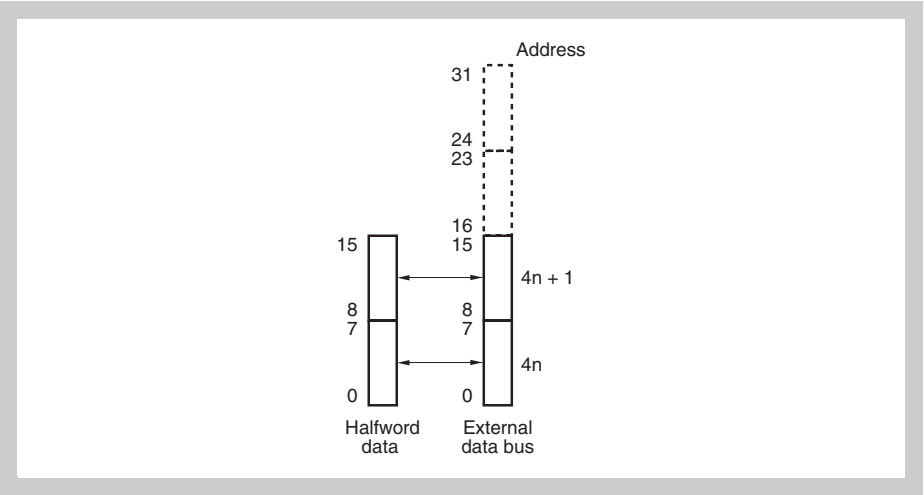


Figure 4-30 Halfword access to 32-bit data bus at address 4n (little endian)

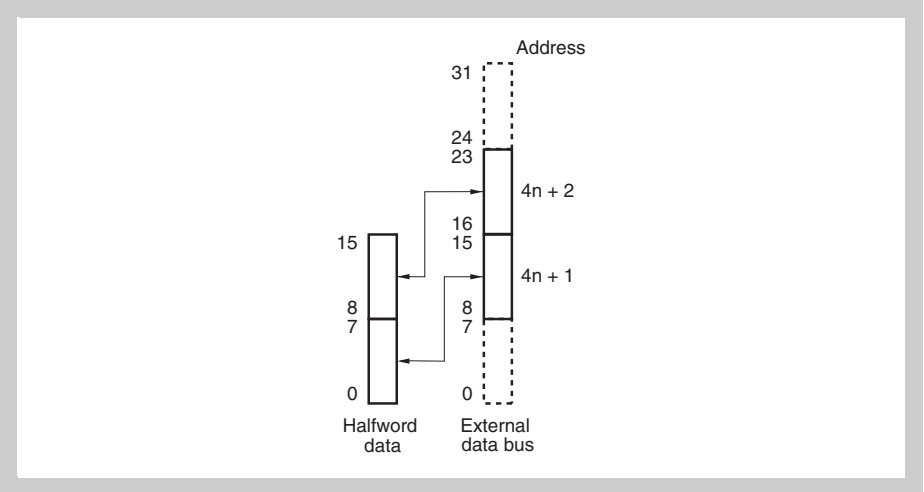


Figure 4-31 Halfword access to 32-bit data bus at address 4n+1 (little endian)

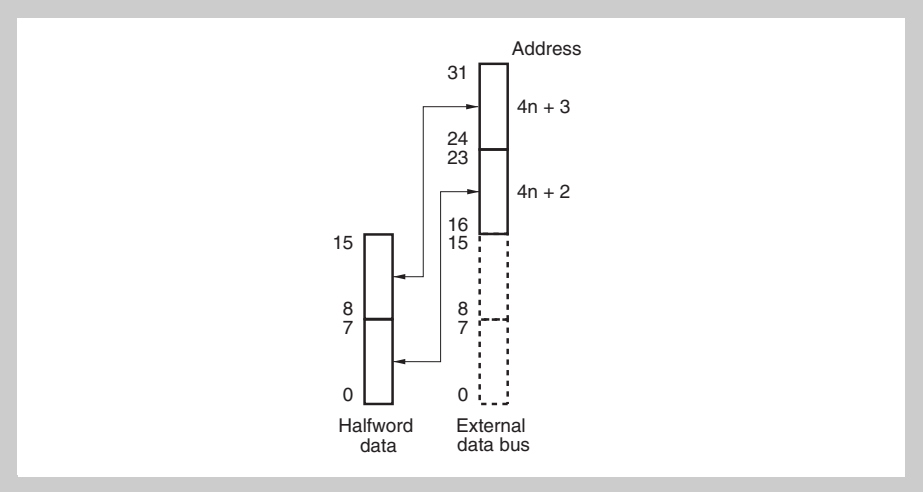


Figure 4-32 Halfword access to 32-bit data bus at address 4n+2 (little endian)

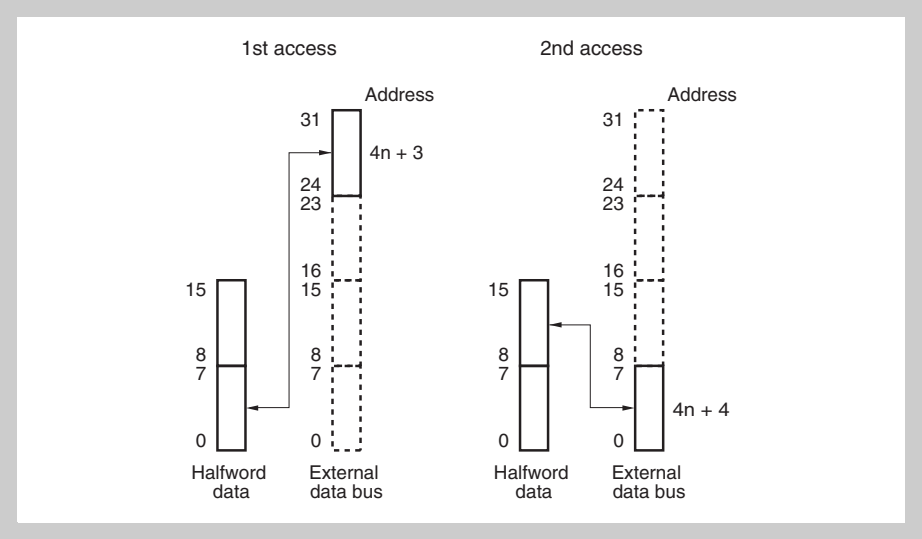


Figure 4-33 Halfword access to 32-bit data bus at address 4n+3 (little endian)

(b) Big endian

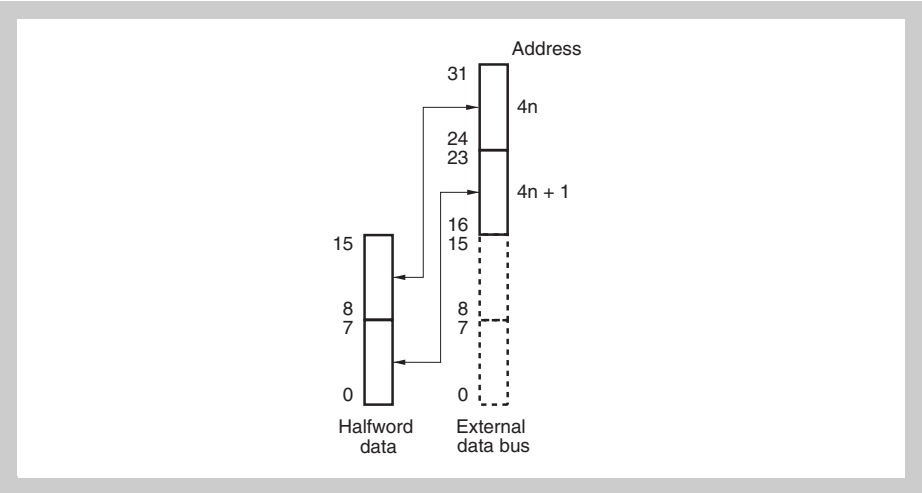


Figure 4-34 Halfword access to 32-bit data bus at address 4n (big endian)

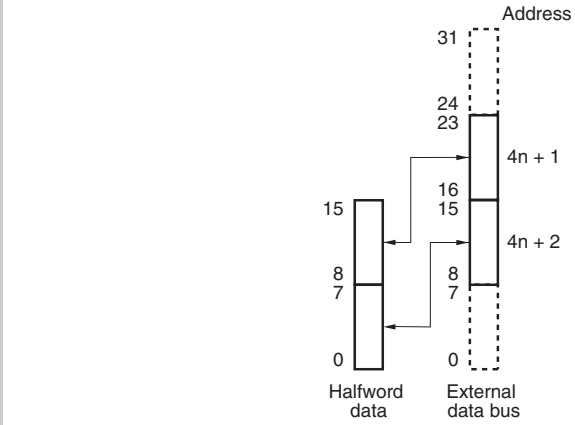


Figure 4-35 Halfword access to 32-bit data bus at address 4n+1 (big endian)

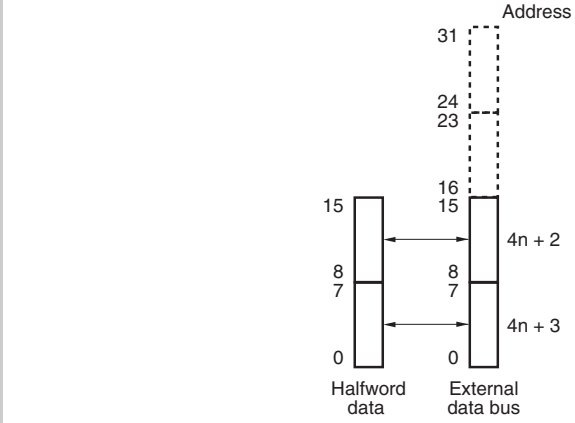


Figure 4-36 Halfword access to 32-bit data bus at address 4n+2 (big endian)

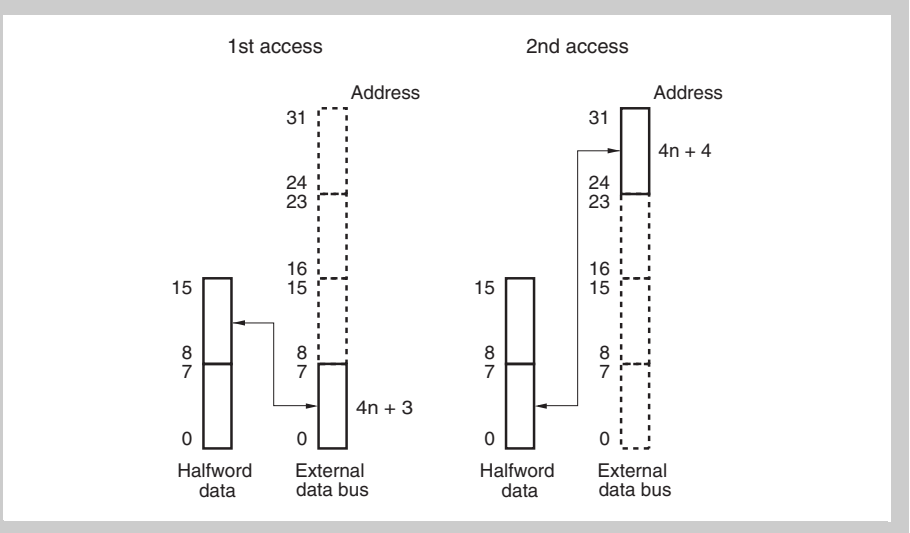


Figure 4-37 Halfword access to 32-bit data bus at address 4n+3 (big endian)

(3) Word access (32 bits)

(a) Little endian

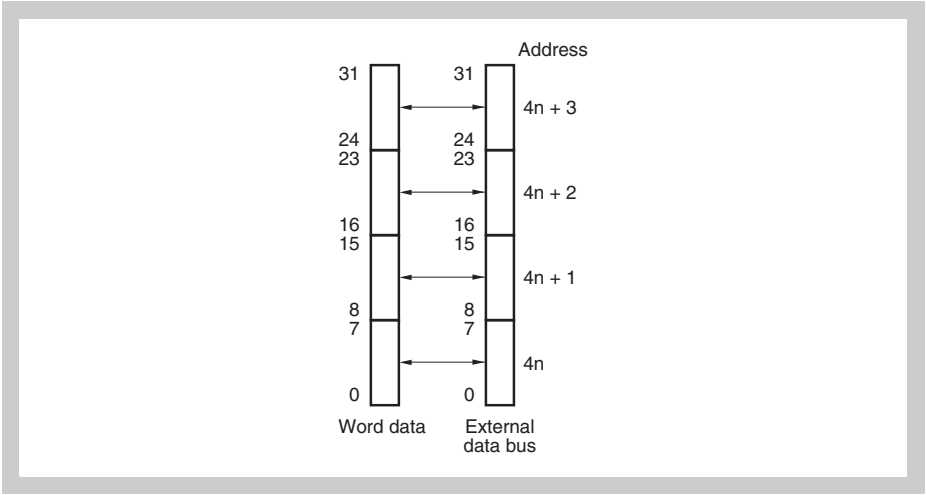


Figure 4-38 Word access to 32-bit data bus at address 4n (little endian)

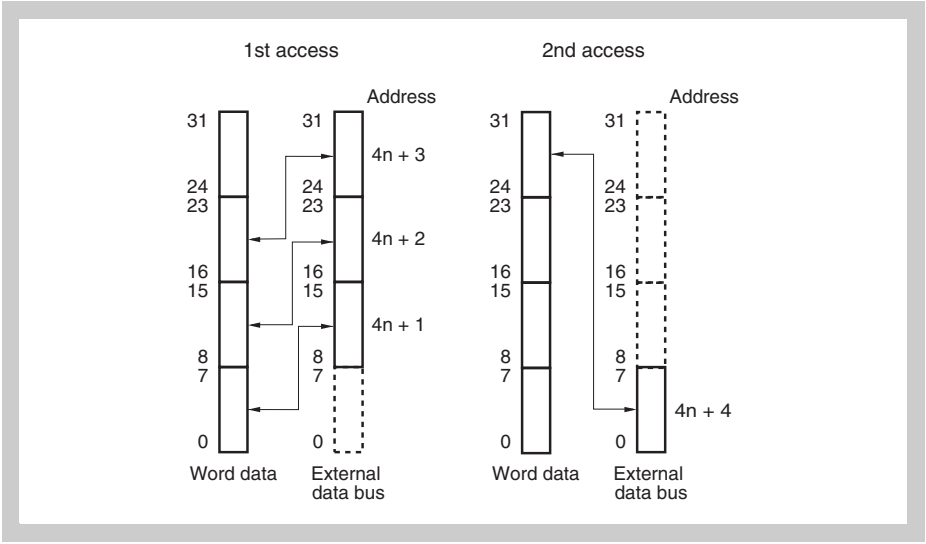


Figure 4-39 Word access to 32-bit data bus at address 4n+1 (little endian)

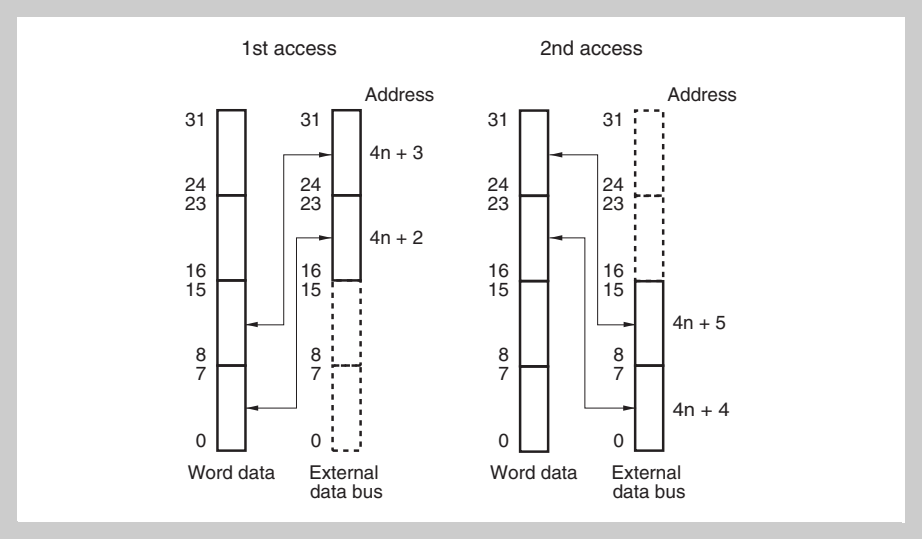


Figure 4-40 Word access to 32-bit data bus at address 4n+2 (little endian)

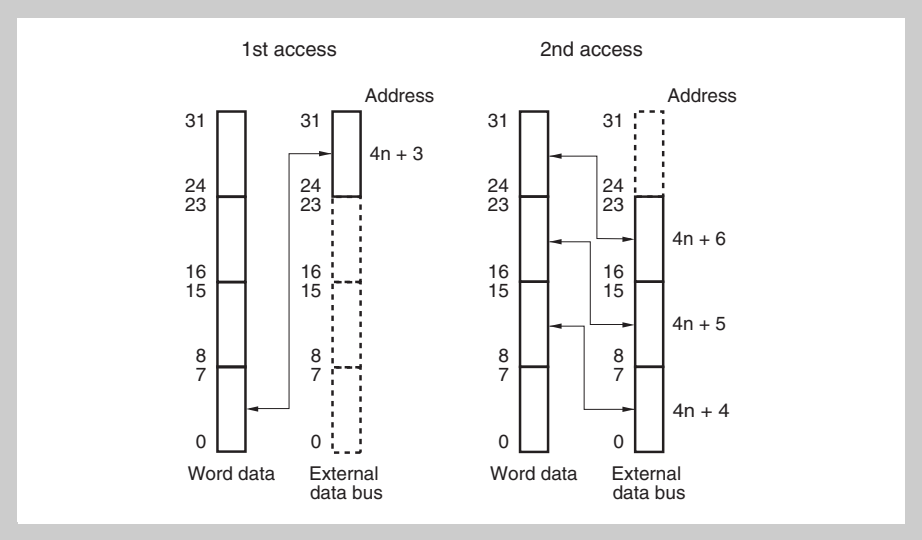


Figure 4-41 Word access to 32-bit data bus at address 4n+3 (little endian)

(b) Big endian

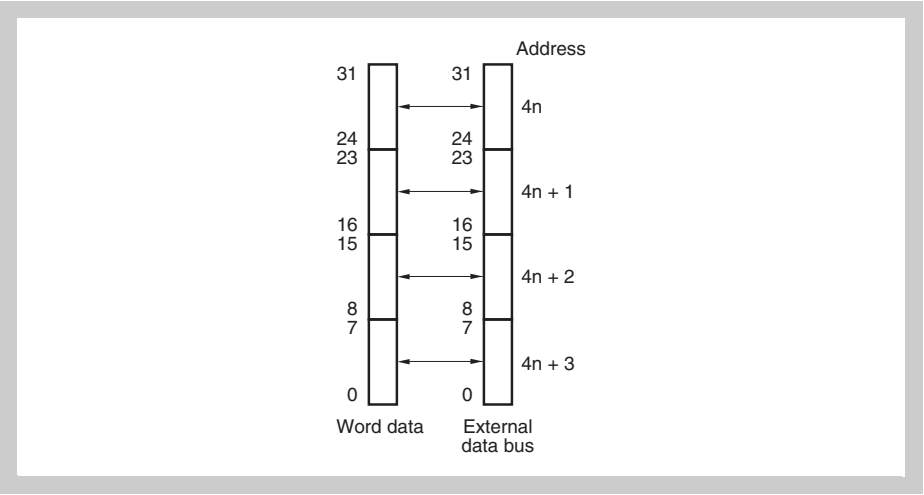


Figure 4-42 Word access to 32-bit data bus at address 4n (big endian)

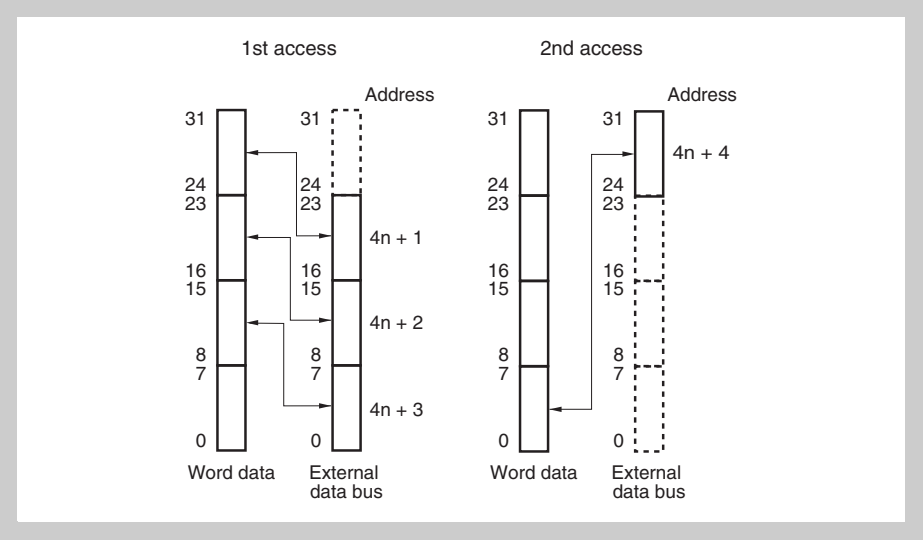


Figure 4-43 Word access to 32-bit data bus at address 4n+1 (big endian)

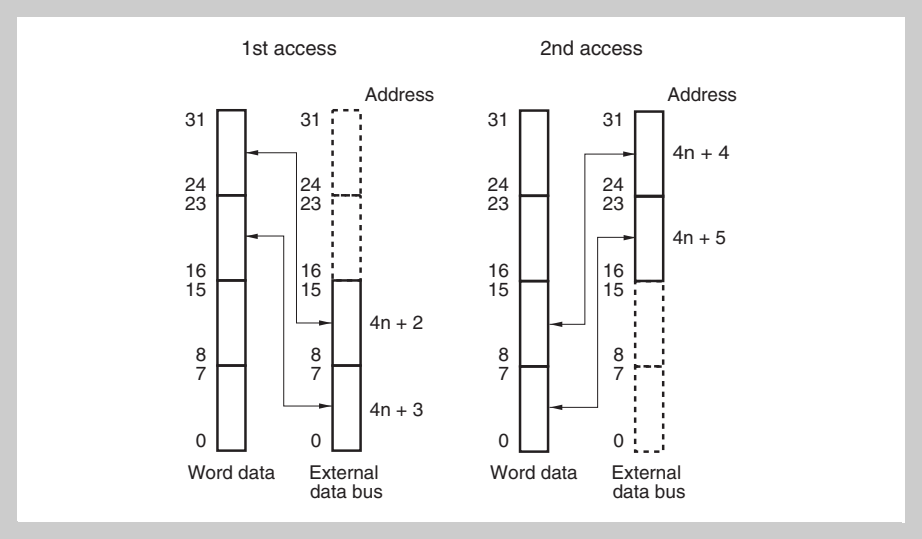


Figure 4-44 Word access to 32-bit data bus at address 4n+2 (big endian)

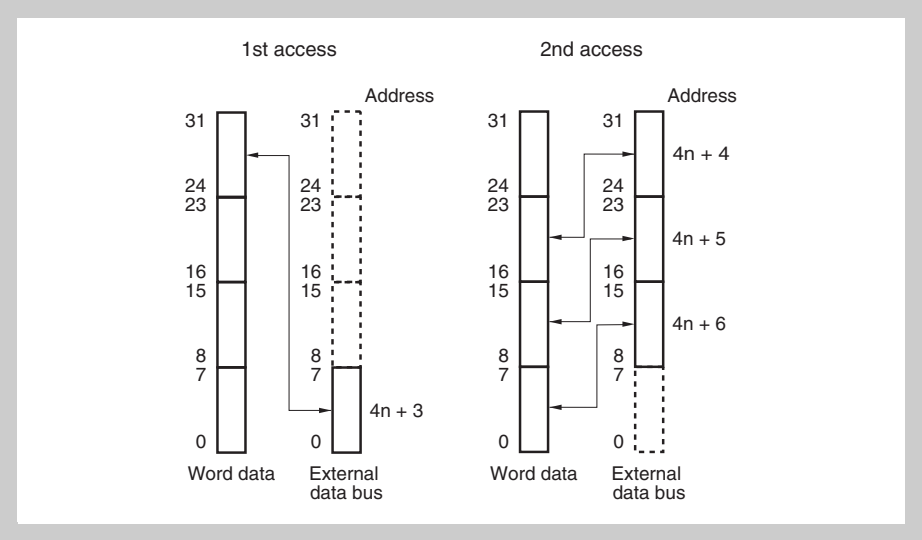


Figure 4-45 Word access to 32-bit data bus at address 4n+3 (big endian)

4.7 Wait Function

4.7.1 Programmable wait function

(1) Data wait control registers 0, 1 (DWC0, DWC1)

To facilitate interfacing with low-speed memory or with I/Os, it is possible to insert up to 7 data wait states with respect to the starting bus cycle for each CS area.

The number of wait states can be specified by data wait control registers 0 and 1 (DWC0, DWC1). Just after system reset, all blocks have 7 data wait states inserted.

Access These registers can be read/written in 16-bit units.

Address
BCT0: FFFF F484H
BCT1: FFFF F486H

Initial Value 7777H

Caution
1. The internal ROM area (flash memory) and the internal RAM area are not subject to programmable waits and ordinarily no wait access is carried out. The internal peripheral I/O area is also not subject to programmable wait states, with wait control performed only by each peripheral function.
2. Write to the DWC0 and DWC1 registers after reset, and then do not change the set values. Also, do not access an external memory area other than that for this initialization routine until initial setting of the DWC0 and DWC1 registers is finished. However, it is possible to access external memory areas whose initialization has been finished.

DWC0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	DW32	DW31	DW30	0	DW22	DW21	DW20	0	DW12	DW11	DW10	0	DW02	DW01	DW00
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
		$\overline{CS3}$				$\overline{CS2}$				$\overline{CS1}$				$\overline{CS0}$	

DWC1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	DW72	DW71	DW70	0	DW62	DW61	DW60	0	DW52	DW51	DW50	0	DW42	DW41	DW40
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
		$\overline{CS7}$				$\overline{CS6}$				$\overline{CS5}$				$\overline{CS4}$	

Table 4-8 DWC0, DWC1 register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>14 to 12,
10 to 8,
6 to 4,
2 to 0</td><td>DWCn[2:0]</td><td>Specifies the inserted data wait states during access of chip select area n ($\overline{CSn}$)
<table><tr><th>DWCn2</th><th>DWCn1</th><th>DWCn0</th><th>Number of inserted Data Wait States</th></tr><tr><td>0</td><td>0</td><td>0</td><td>No wait states inserted</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>2</td></tr><tr><td>0</td><td>1</td><td>1</td><td>3</td></tr><tr><td>1</td><td>0</td><td>0</td><td>4</td></tr><tr><td>1</td><td>0</td><td>1</td><td>5</td></tr><tr><td>1</td><td>1</td><td>0</td><td>6</td></tr><tr><td>1</td><td>1</td><td>1</td><td>7</td></tr></table></td></tr>
</table>

(2) Address wait control register (AWC)

The V850E/PH2 allows insertion of address setup wait and address hold wait states before and after the T1 cycle.

The address setup wait and address hold wait states can be set with the AWC register for each CS area.

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

Address FFFF F488H

Initial Value 0000H

Caution

1. The internal ROM area (flash memory) and the internal RAM area are not subject to programmable waits and ordinarily no wait access is carried out. The internal peripheral I/O area is also not subject to programmable wait states, with wait control performed only by each peripheral function.
2. Write to the AWC registers after reset, and then do not change the set values. Also, do not access an external memory area other than that for this initialization routine until initial setting of the AWC registers is finished. However, it is possible to access external memory areas whose initialization has been finished.

AWC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AHW7	ASW7	AHW6	ASW6	AHW5	ASW5	AHW4	ASW4	AHW3	ASW3	AHW2	ASW2	AHW1	ASW1	AHW0	ASW0
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
	$\overline{CS7}$		$\overline{CS6}$		$\overline{CS5}$		$\overline{CS5}$		$\overline{CS3}$		$\overline{CS2}$		$\overline{CS1}$		$\overline{CS0}$

Table 4-9 AWC register contents

Bit position	Bit name	Function
15, 13, 11, 9, 7, 5, 3, 1	AHWn	Controls the address hold wait insertion during access of chip select area n ($\overline{CSn}$). 0: No insertion 1: Address hold wait state insertion after T1 bus cycle
14, 12, 10, 8, 6, 4, 2, 0	ASWn	Controls the address setup wait insertion during access of chip select area n ($\overline{CSn}$). 0: No insertion 1: Address setup wait state insertion before T1 bus cycle

4.8 Idle State Insertion Function

To facilitate interfacing with low-speed memory devices, an idle state (TI) can be inserted into the current bus cycle after the T2 state to meet the data output float delay time (tdf) on memory read access for each CS space. The bus cycle following the T2 state starts after the idle state is inserted.

An idle state is inserted after read/write cycles for SRAM, external I/O, or external ROM.

In the following cases, an idle state is inserted in the timing.

- after read/write cycles for SRAM, external I/O, or external ROM

The idle state insertion setting can be specified by program using the bus cycle control register (BCC) and the bus clock dividing control register (DVC).

Immediately after the system reset, idle state insertion is automatically programmed for all memory blocks on read access.

(1) Bus cycle control register (BCC)

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

Address FFFF F48AH

Initial Value AAAAH

Caution

1. Idle states cannot be inserted in internal ROM, internal RAM, on-chip peripheral I/O, or programmable peripheral I/O areas.
2. Write to the BCC register after reset, and then do not change the set value. Also, do not access an external memory area other than that for this initialization routine until initial setting of the BCC register is finished. However, it is possible to access external memory areas whose initialization has been finished.
3. Do not change the settings of bits that are 0 after reset. Otherwise the operation of the external bus interface cannot be ensured.

BCC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BC71	0	BC61	0	BC51	0	BC41	0	BC31	0	BC21	0	BC11	0	BC01	0
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 4-10 BCC register contents

Bit position	Bit name	Function
15, 13, 11, 9, 7, 5, 3, 1	BCn1	Controls the idle state insertion during access of chip select area n ($\overline{CSn}$). 0: No insertion 1: Idle state insertion **Note:** When bit BCn1 bit is set to “1”, an idle state will be inserted after any read access. If an idle state after write access is necessary, the BCWI bit of the DVC register has to be set additionally.

(2) Bus clock dividing control register (DVC)

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

Reset input changes the value of this register to initial setting 01H.

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

Address FFFF F48EH

Initial Value 01H

Caution

1. Idle states cannot be inserted in internal ROM, internal RAM, on-chip peripheral I/O, or programmable peripheral I/O areas.
2. Write to the DVC register after reset, and then do not change the set value. Also, do not access an external memory area other than that for this initialization
routine until initial setting of the DVC register is finished. However, it is possible to access external memory areas whose initialization has been finished.
3. Do not change the settings of bits 0 to 6. Otherwise the operation of the external bus interface cannot be ensured.

DVC

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

Table 4-11 DVC register contents

Bit position	Bit name	Function
7	BCWI	Controls the idle state insertion after write cycle. 0: No insertion of idle state after write access 1: Idle state inserted after write access **Note:** BCWI bit setting is only valid when BCn1 bit of the BCC register, corresponding to the $\overline{CSn}$ area for which the write access will be performed, is set to “1”.

4.9 Bus Priority Order

There are two external bus cycles: operand data access and instruction fetch.

As for the priority order, the highest priority has the instruction fetch than operand data access.

An instruction fetch may be inserted between read access and write access during read modify write access.

Also, an instruction fetch may be inserted between bus access and bus access during CPU bus clock.

Table 4-12 Bus priority prder

Priority order	External bus cycle	Bus master
↕ High	Operand data access	CPU
	Instruction fetch	CPU

4.10 Boundary Operation Conditions

4.10.1 Program space

Branching to the on-chip peripheral I/O area is prohibited. If the above is performed, undefined data is fetched, and fetching from the external memory is not performed.

4.10.2 Data space

The V850E/PH2 is provided with an address misalign function.

Through this function, regardless of the data format (word, halfword or byte), data can be allocated to all addresses. However, in the case of word data and halfword data, if the data is not subject to boundary alignment, the bus cycle will be generated at least 2 times and bus efficiency will drop.

(1) External bus width: 16 bits

(a) In the case of halfword-length data access

When the address's LSB is 1, a byte-length bus cycle will be generated 2 times.

(b) In the case of word-length data access

- When the address's LSB is 1, bus cycles will be generated in the order of byte-length bus cycle, halfword-length bus cycle, and byte-length bus cycle.
- When the address's lower 2 bits are 10B, a halfword-length bus cycle will be generated 2 times.

(2) External bus width: 32 bits

(a) In the case of halfword-length data access

When the address's lower 2 bits are 11B, a byte-length bus cycle will be generated 2 times.

(b) In the case of word-length data access

When the address's lower 2 bits are 10B, a halfword-length bus cycle will be generated 2 times.

Chapter 5 Memory Access Control Function

5.1 SRAM, External ROM, External I/O Interface

5.1.1 Features

- SRAM is accessed in a minimum of 2 states.
- Up to 7 states of programmable data waits can be inserted by setting the DWC0 and DWC1 registers.
- Data wait can be controlled via $\overline{\text{WAIT}}$ pin input.
- An idle state can be inserted after a read/write cycle by setting the BCC and DVC registers.
- An address setup wait state and an address hold state can be inserted by setting the ASC register.

Note The memory access control function is not available on μPD70F3447.

5.1.2 SRAM connection

Examples of connection to SRAM are shown below.

(1) Examples of connections with SRAM of 8 bit data size

(a) When data bus width is 32 bits and data size of SRAM is 8 bits

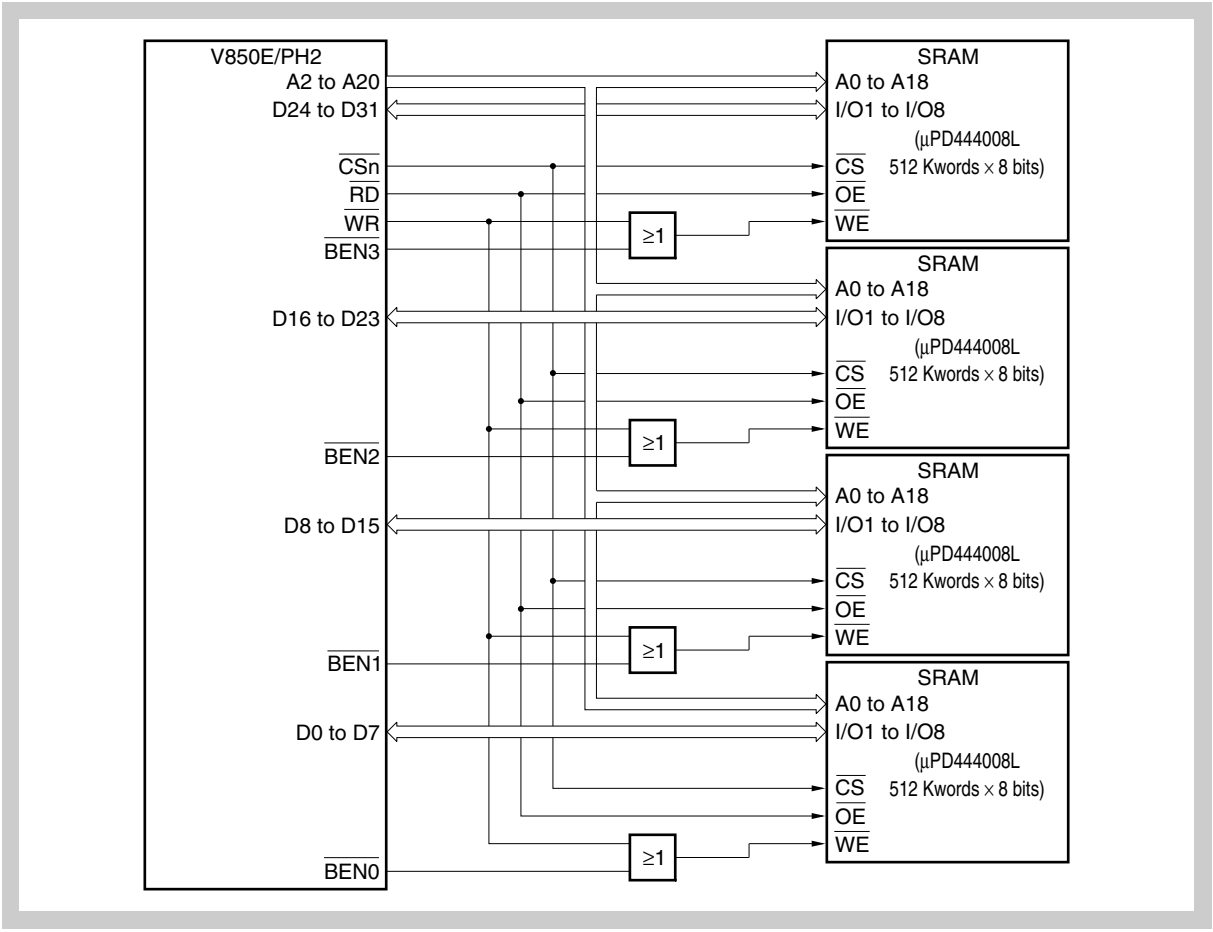


Figure 5-1 Exemplary connection between 8 bit SRAM and 32 bit data bus

(b) When data bus width is 8 bits and data size of SRAM is 8 bits

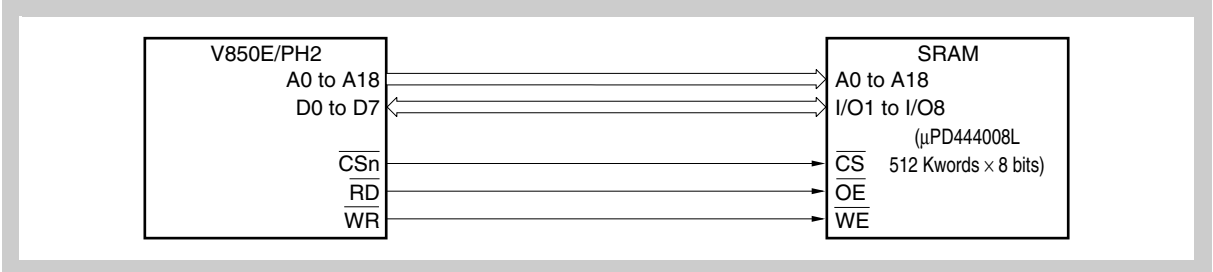


Figure 5-2 Exemplary connection between 8 bit SRAM and 8 bit data bus

(2) Examples of connections with SRAM of 16 bit data size

(a) When data bus width is 32 bits and data size of SRAM is 16 bits

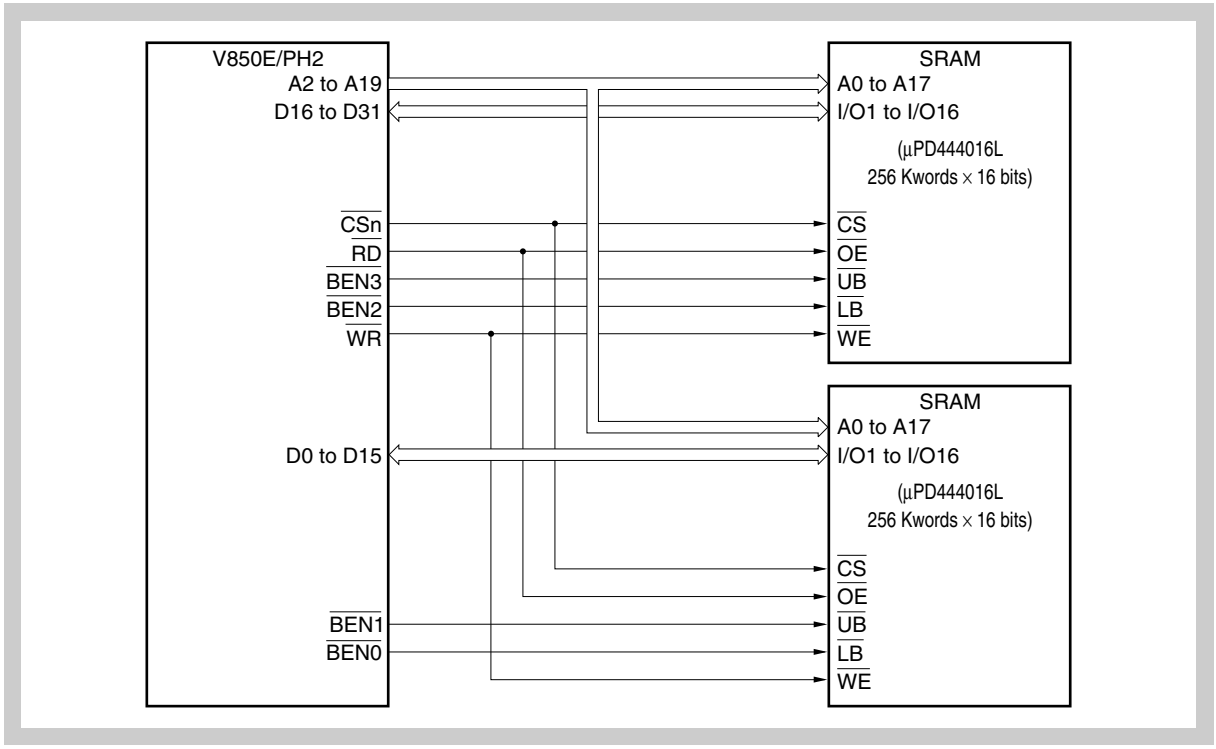


Figure 5-3 Exemplary connection between 16 bit SRAM and 32 bit data bus

(b) When data bus width is 16 bits and data size of SRAM is 16 bits

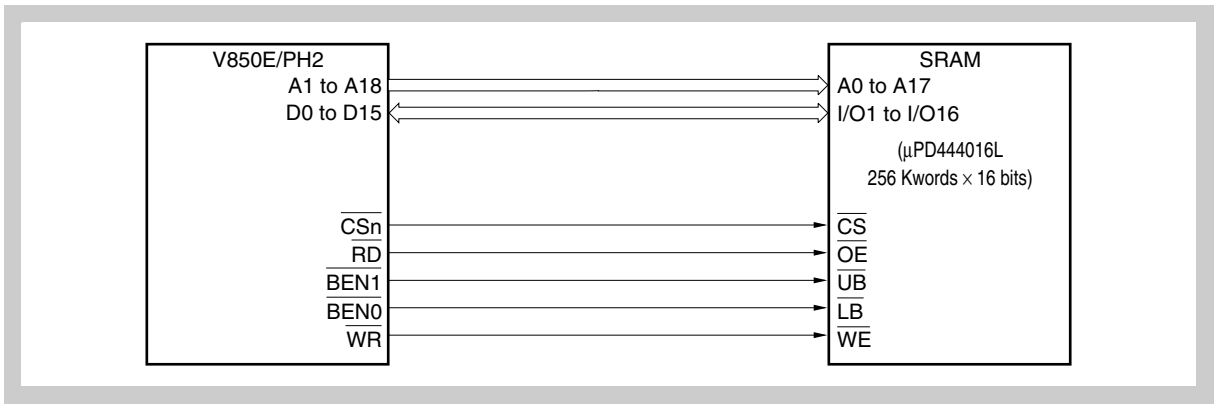


Figure 5-4 Exemplary connection between 16 Bit SRAM and 16 bit data bus

5.1.3 SRAM, external ROM, external I/O access

(1) Read Access

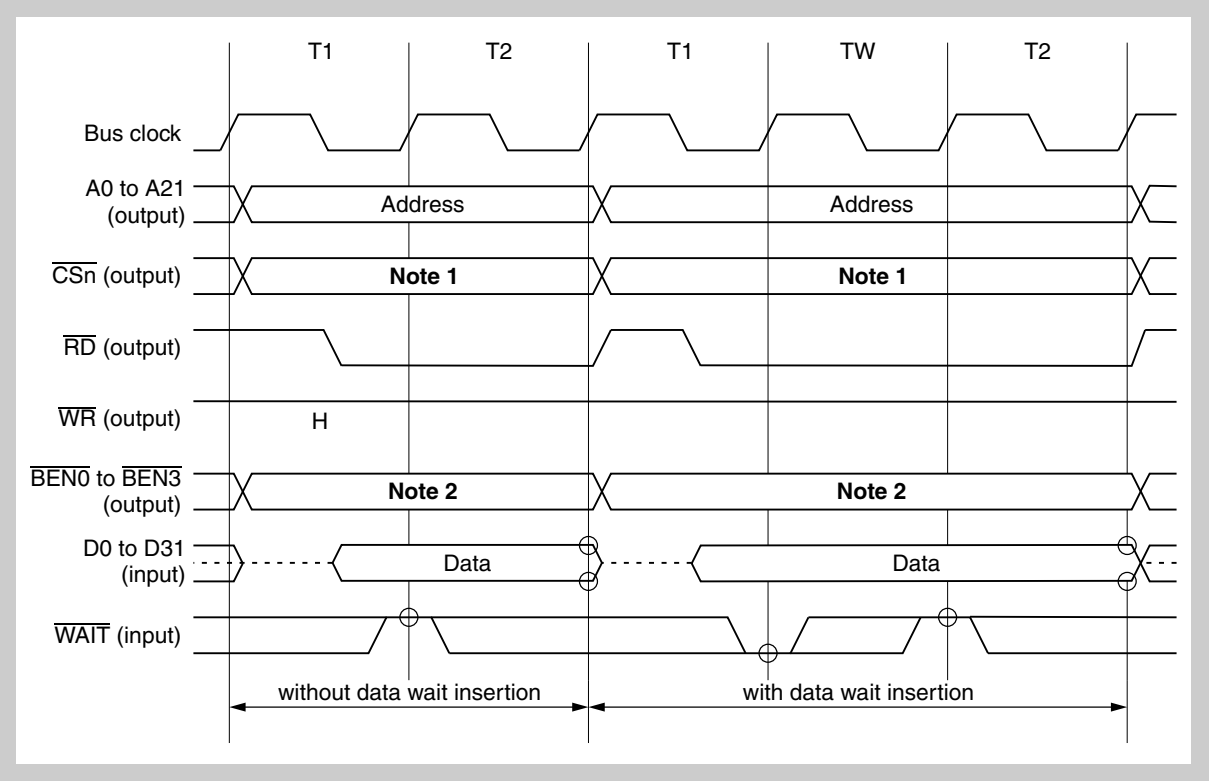


Figure 5-5 Read timing of SRAM, external ROM, external I/O access

Note
1. $\overline{CSn}$ output levels depend on the accessed area when enabled by BCT0 and BCT1 registers.
2. $\overline{BEN0}$ to $\overline{BEN3}$ output levels depend on the accessed type (byte, half-word, or word) and the external bus size (8, 16, or 32 bits) specified by the BSC register

Remarks
1. n = 0, 1, 3, 4
2. Bus clock = $f_{XX}/2$
3. The circle indicates the sampling timing.
4. The dashed line indicates the high impedance state.

(2) Read Access (Idle State Inserted)

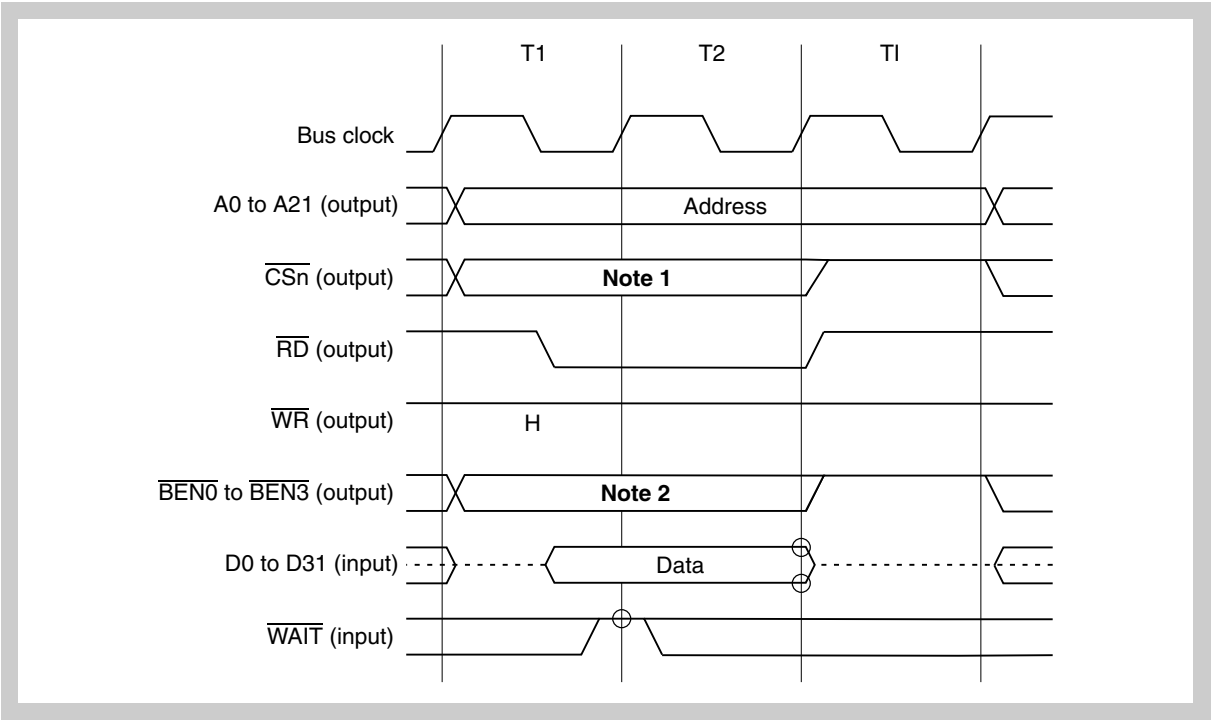


Figure 5-6 Read timing of SRAM, external ROM, external I/O access (idle state inserted)

Note
1. $\overline{CSn}$ output levels depend on the accessed area when enabled by BCT0 and BCT1 registers.
2. $\overline{BEN0}$ to $\overline{BEN3}$ output levels depend on the accessed type (byte, half-word, or word) and the external bus size (8, 16, or 32 bits) specified by the BSC register

Remarks
1. n = 0, 1, 3, 4
2. Bus clock = $f_{XX}/2$
3. The circle indicates the sampling timing.
4. The dashed line indicates the high impedance state.

(3) Read Access (Data Wait, Idle State Inserted)

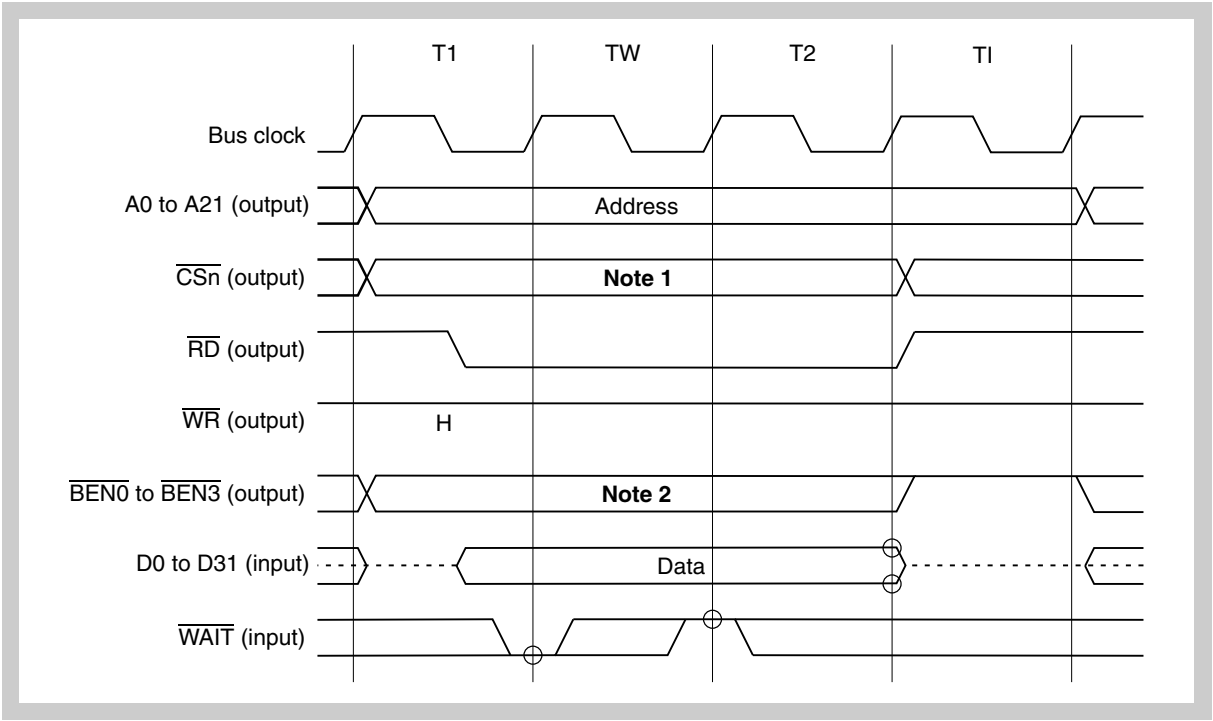


Figure 5-7 Read timing of SRAM, external ROM, external I/O access (data wait, idle state inserted)

Note
1. $\overline{CSn}$ output levels depend on the accessed area when enabled by BCT0 and BCT1 registers.
2. $\overline{BEN0}$ to $\overline{BEN3}$ output levels depend on the accessed type (byte, half-word, or word) and the external bus size (8, 16, or 32 bits) specified by the BSC register

Remarks
1. n = 0, 1, 3, 4
2. Bus clock = $f_{XX}/2$
3. The circle indicates the sampling timing.
4. The dashed line indicates the high impedance state.

(4) Read Access (Address Setup Wait and Address Hold Wait State Inserted)

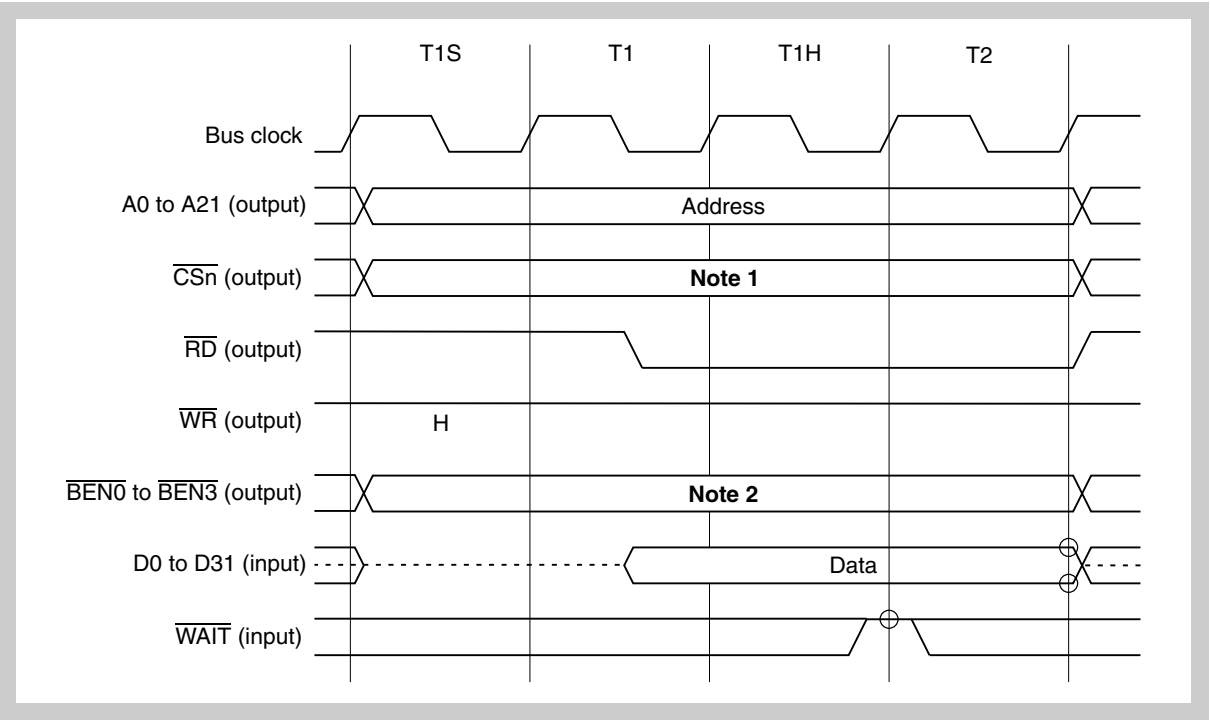


Figure 5-8 Read timing of SRAM, external ROM, external I/O access (address setup wait and address hold wait state inserted)

Note
1. $\overline{CSn}$ output levels depend on the accessed area when enabled by BCT0 and BCT1 registers.
2. $\overline{BEN0}$ to $\overline{BEN3}$ output levels depend on the accessed type (byte, half-word, or word) and the external bus size (8, 16, or 32 bits) specified by the BSC register

Remarks
1. n = 0, 1, 3, 4
2. Bus clock = $f_{XX}/2$
3. The circle indicates the sampling timing.
4. The dashed line indicates the high impedance state.

(5) Write Access

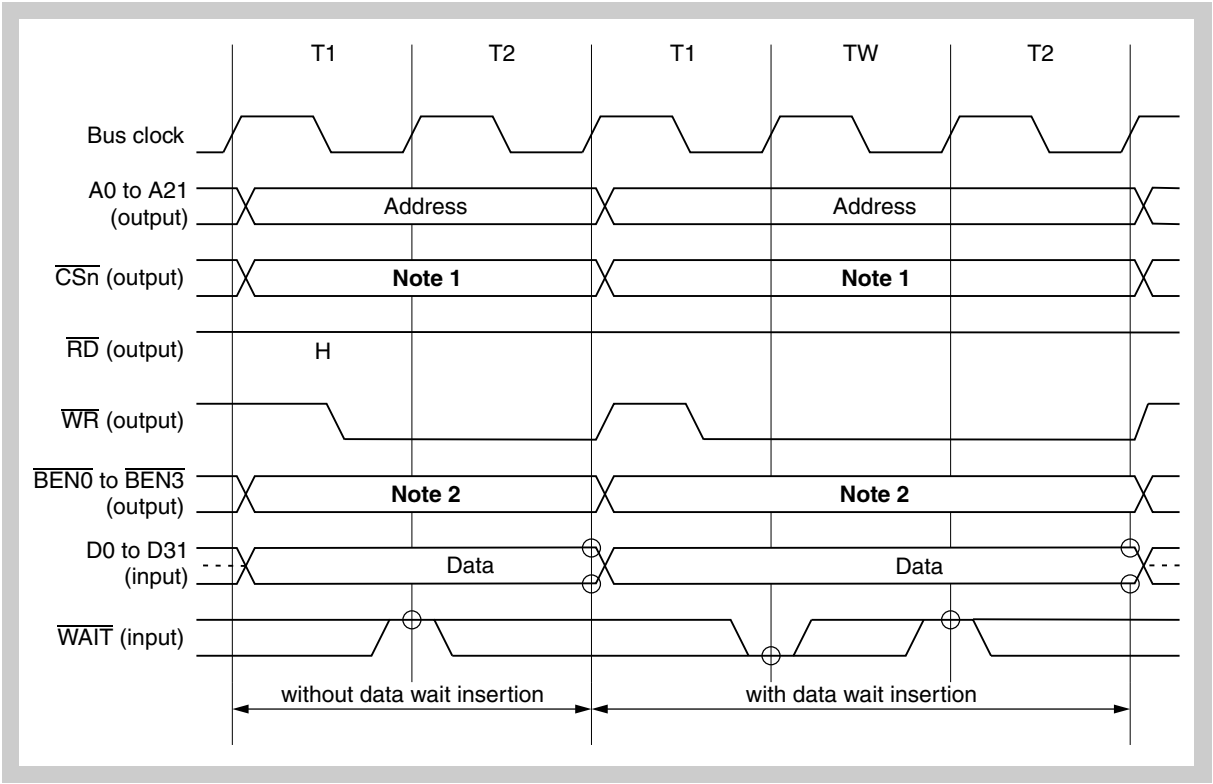


Figure 5-9 Write timing of SRAM, external ROM, external I/O access

Note
1. $\overline{CSn}$ output levels depend on the accessed area when enabled by BCT0 and BCT1 registers.
2. $\overline{BEN0}$ to $\overline{BEN3}$ output levels depend on the accessed type (byte, half-word, or word) and the external bus size (8, 16, or 32 bits) specified by the BSC register

Remarks
1. n = 0, 1, 3, 4
2. Bus clock = $f_{XX}/2$
3. The circle indicates the sampling timing.
4. The dashed line indicates the high impedance state.

(6) Write Access (Idle State Inserted)

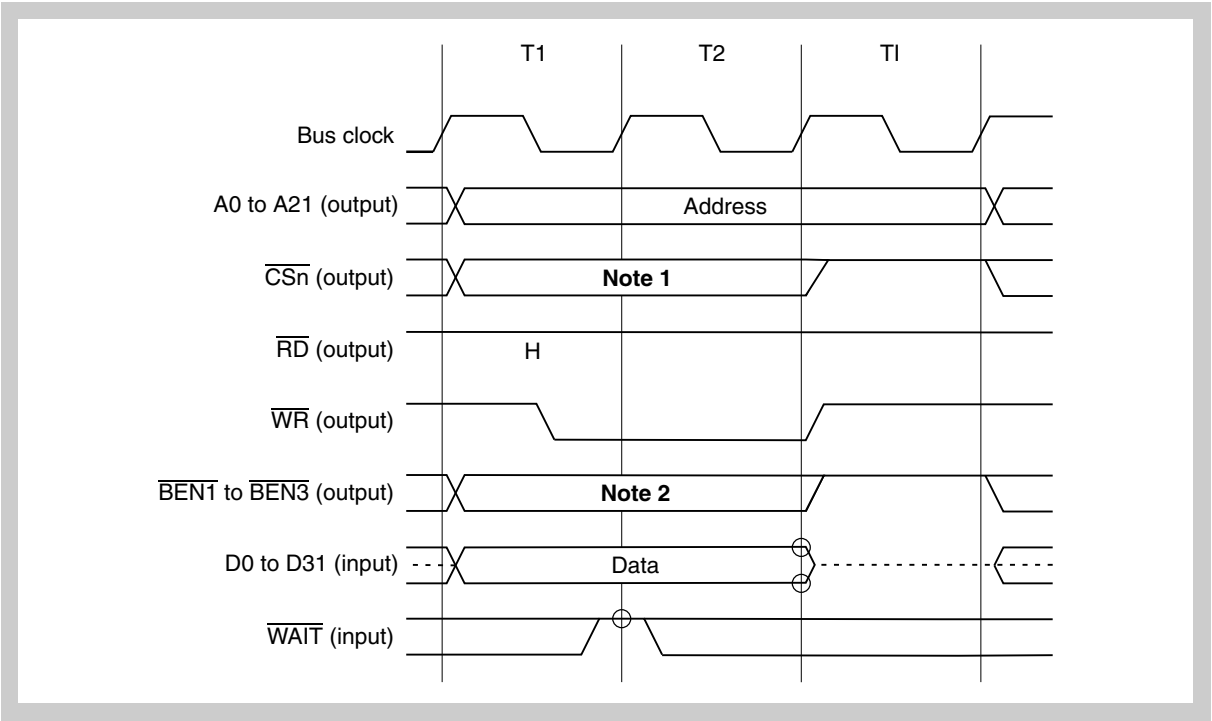


Figure 5-10 Write timing of SRAM, external ROM, external I/O access (idle state inserted)

Note
1. $\overline{CSn}$ output levels depend on the accessed area when enabled by BCT0 and BCT1 registers.
2. $\overline{BEN0}$ to $\overline{BEN3}$ output levels depend on the accessed type (byte, half-word, or word) and the external bus size (8, 16, or 32 bits) specified by the BSC register

Remarks
1. n = 0, 1, 3, 4
2. Bus clock = $f_{XX}/2$
3. The circle indicates the sampling timing.
4. The dashed line indicates the high impedance state.

(7) Write Access (Data Wait, Idle State Inserted)

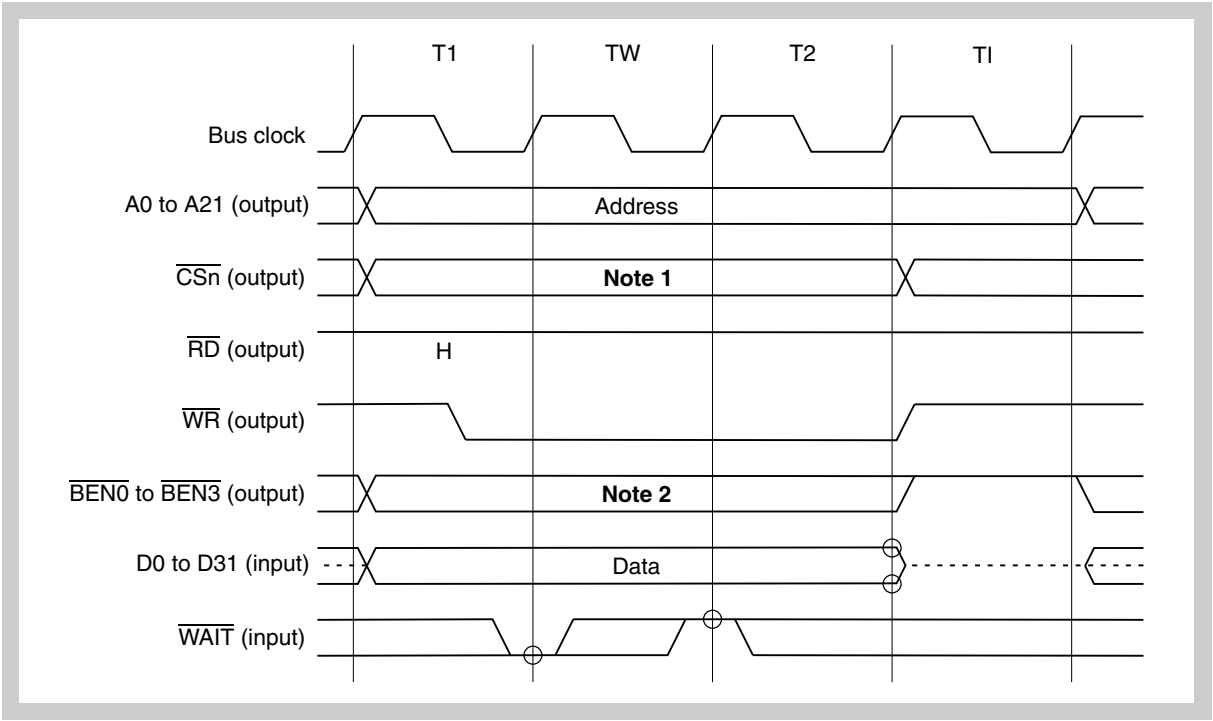


Figure 5-11 Write timing of SRAM, external ROM, external I/O access (data wait, idle state inserted)

Note
1. $\overline{CSn}$ output levels depend on the accessed area when enabled by BCT0 and BCT1 registers.
2. $\overline{BEN0}$ to $\overline{BEN3}$ output levels depend on the accessed type (byte, half-word, or word) and the external bus size (8, 16, or 32 bits) specified by the BSC register

Remarks
1. n = 0, 1, 3, 4
2. Bus clock = $f_{XX}/2$
3. The circle indicates the sampling timing.
4. The dashed line indicates the high impedance state.

(8) Read (Address Setup Wait and Address Hold Wait State Inserted)

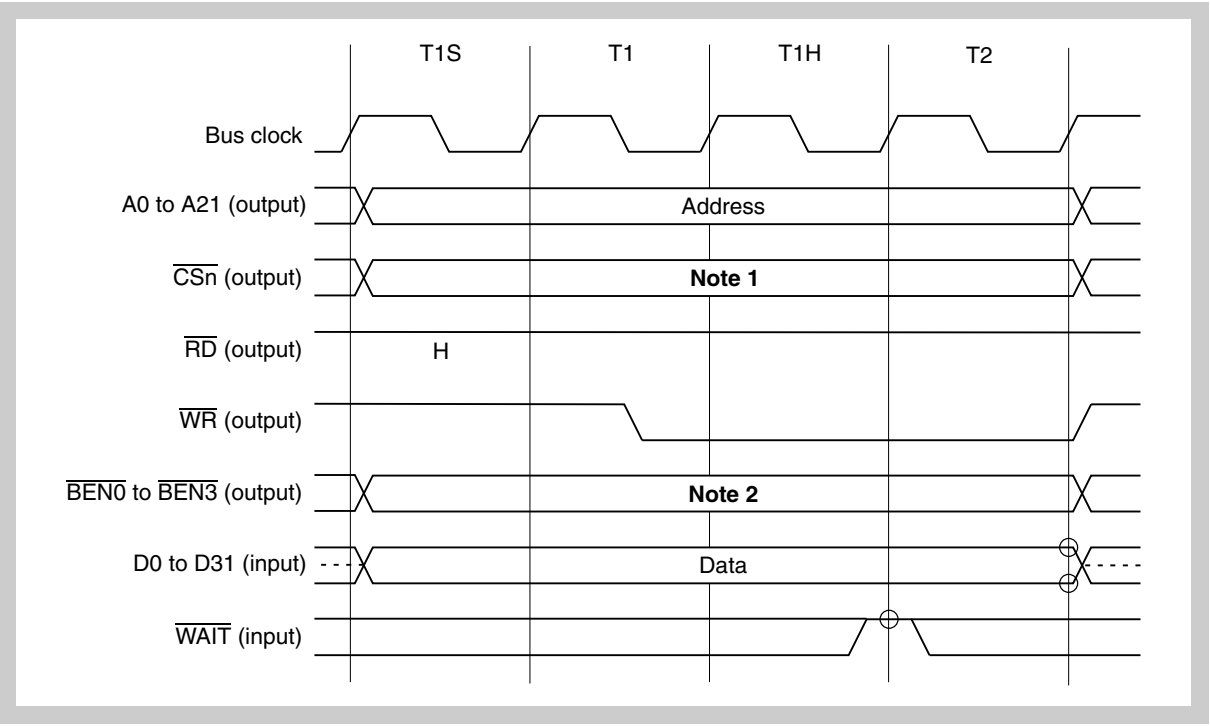


Figure 5-12 Write timing of SRAM, external ROM, external I/O access (address setup wait and address hold wait state inserted)

Note
1. $\overline{CSn}$ output levels depend on the accessed area when enabled by BCT0 and BCT1 registers.
2. $\overline{BEN0}$ to $\overline{BEN3}$ output levels depend on the accessed type (byte, half-word, or word) and the external bus size (8, 16, or 32 bits) specified by the BSC register

Remarks
1. n = 0, 1, 3, 4
2. Bus clock = $f_{XX}/2$
3. The circle indicates the sampling timing.
4. The dashed line indicates the high impedance state.

Chapter 6 DMA Functions (DMA Controller)

6.1 Features

The V850E/PH2 includes a direct memory access (DMA) controller (DMAC) that executes and controls DMA transfer.

The DMAC controls data transfer between internal RAM (iRAM) and peripheral I/O registers, based on DMA requests issued by the on-chip peripheral I/O (A/D converters, inverter timers, and serial interfaces), with the following features.

- 2 channels for DMA transfer from A/D converter (ADC0, ADC1)
  - Transfer object: I/O → iRAM
  - Transfer size: 16 bits
  - Dedicated transfer channels for ADC0 and ADC1

- 2 channels for DMA transfer to PWM timer (TMR0, TMR1)
  - Transfer object: iRAM → I/O
  - Transfer size: 16 bits
  - Dedicated transfer channels for TMR0 and TMR1

- 2 channels for DMA transfer from serial interfaces on reception completion
  - Transfer object: I/O → iRAM
  - Transfer size: 8 or 16 bits
  - DMA request for each channel selectable Clocked serial interfaces: CSIB0, CSIB1, CSI30, CSI31
    Asynchronous serial interface: UARTC0, UARTC1

- 2 channels for DMA transfer to serial interfaces on transmission repetition
  - Transfer object: iRAM → I/O
  - Transfer size: 8 or 16 bits
  - DMA request for each channel selectable
    Clocked serial interfaces: CSIB0, CSIB1, CS30, CSI31
    Asynchronous serial interface: UARTC0, UARTC1

- Up to 256 transfer counts for each channel

6.2 Control Registers

(1) DMA transfer memory start address registers 0 to 7 (MAR0 to MAR7)

The MARn register specifies the subordinated 16 bits of the DMA transfer start address within the internal RAM area for the DMA channel n (n = 0 to 7).

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

Address

MAR0:	$FFFFF300_H$	MAR1:	$FFFFF302_H$
MAR2:	$FFFFF304_H$	MAR3:	$FFFFF306_H$
MAR4:	$FFFFF308_H$	MAR5:	$FFFFF30A_H$
MAR6:	$FFFFF30C_H$	MAR7:	$FFFFF30E_H$

Initial Value Undefined

Caution

1. Since the internal RAM area is mapped between $3FF0000_H$ and $3FF7FFF_H$ the value written to the MARn register has to be in the range from 0000_H to $7FFF_H$.
2. The value set to the MARn register is increased by each DMA transfer of channels. It does not keep the initial value after the DMA transfer ends.

MARn

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

(2) DMA transfer SFR start address registers 2, 3 (SAR2, SAR3)

The SARn register specifies the start address of the TMR register for which the DMA transfer is started on DMA channel n (n = 2, 3).

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

Address SAR2: $FFFFF314_H$
SAR3: $FFFFF316_H$

Initial value Undefined

Caution During DMA transfer (DMAMC.DEn = 1) the contents of the SARn register may change.
After each DMA transfer the contents is incremented by 1 until the final value (07H) is reached. When the SARn register contents becomes 07_H, the initial set value is reloaded.

SARn

7	6	5	4	3	2	1	0
0	0	0	0	0	SARn2	SARn1	SARn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 6-1 SARn register contents

Bit position	Bit name	Function
2 to 0	SARn[2:0]	Specifies the start address of the TMR register for which the DMA transfer is started on the corresponding DMA channel n.

SARn2	SARn1	SARn0	Start Address of TMR Reload Register			
			n = 2		n = 3	
			Register	Address	Register	Address
0	0	0	TR0CCR5	$FFFFF590_H$	TR1CCR5	$FFFFF5D0_H$
0	0	1	TR0CCR4	$FFFFF592_H$	TR1CCR4	$FFFFF5D2_H$
0	1	0	N/A[a]	$FFFFF594_H$	N/A[a]	$FFFFF5D4_H$
0	1	1	N/A[b]	$FFFFF596_H$	N/A[b]	$FFFFF5D6_H$
1	0	0	TR0CCR0	$FFFFF598_H$	TR1CCR0	$FFFFF5D8_H$
1	0	1	TR0CCR3	$FFFFF59A_H$	TR1CCR3	$FFFFF5DA_H$
1	1	0	TR0CCR2	$FFFFF59C_H$	TR1CCR2	$FFFFF5DC_H$
1	1	1	TR0CCR1	$FFFFF59E_H$	TR1CCR1	$FFFFF5DE_H$

a) Although the register address is meaningless, a transfer to this address is always performed when SARn[2:0] bits are equal to 010_B or less.

b) Although the register address is meaningless, a transfer to this address is always performed when SARn[2:0] bits are equal to 011_B or less.

(3) DMA transfer count registers 0 to 7 (DTCR0 to DTCR7)

The DTCRn register is an 8-bit register that sets the transfer count for DMA channel n and stores the remaining transfer count during DMA transfer (n = 0 to 7).

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

Address

DTCR0:	$FFFFF320_H$	DTCR1:	$FFFFF322_H$
DTCR2:	$FFFFF324_H$	DTCR3:	$FFFFF326_H$
DTCR4:	$FFFFF328_H$	DTCR5:	$FFFFF32A_H$
DTCR6:	$FFFFF32C_H$	DTCR7:	$FFFFF32E_H$

Initial value Undefined

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 values becomes 00H.
2. A DMA request becomes only effective after the DTCRn register was written. Even if 00_H (means a transfer count of 256) is the initial value, the DTRCn register must be rewritten in order to enable a new DMA transfer.

DTCRn

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 6-2 DTCRn register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7 to 0</td><td>DTCRn[7:0]</td><td>Specifies the transfer count for the corresponding DMA channel and stores the remaining transfer count during DMA transfer.
<table>
<tr><th>DTCRn7</th><th>DTCRn6</th><th>DTCRn5</th><th>DTCRn4</th><th>DTCRn3</th><th>DTCRn2</th><th>DTCRn1</th><th>DTCRn0</th><th>Remaining DMA Transfer Counts</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>256</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td colspan="8">.
.
.</td><td>.
.
.</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>254</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>255</td></tr>
</table>
</td></tr>
</table>

(4) DMA mode control register (DMAMC)

The DMAMC register is an 8-bit register that controls the operation of the DMA channels.

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

Address $FFFFF330_H$

Initial value 00_H

Caution Writing of the DE1 and DE0 bits is prohibited if the corresponding A/D converter is operating.

	7	6	5	4	3	2	1	0
DMAMC	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 6-3 DMAMC register contents

Bit position	Bit name	Function
7 to 0	DEn	Controls the operation of DMA channel n. 0: DMA transfer operation of channel n disabled. 1: DMA transfer operation of channel n enabled.

(5) DMA status register (DMAS)

The DMAS register is an 8-bit register that displays the transfer status of the DMA channels.

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

Address $FFFFF332_H$

Initial value 00_H

	7	6	5	4	3	2	1	0
DMAS	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 6-4 DMAS register contents

Bit position	Bit name	Function
7 to 0	DMASn	Status of DMA channel n. 0: DMA transfer of channel n is idle or in progress. 1: DMA transfer of channel n 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.

(6) DMA data size control register (DMADSC)

The DMADSC register is an 8-bit register that controls the transfer data size of DMA channels 4 to 7. The data size of DMA channels 0 to 3 is fixed, and therefore not selectable.

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

Address $FFFFF334_H$

Initial value 00_H

	7	6	5	4	3	2	1	0
DMADSC	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 6-5 DMADSC register contents

Bit position	Bit name	Function
7 to 4	DMADSCn	Specifies the transfer data size of DMA channel n. 0: 8 bit 1: 16 bit

(7) TDMA trigger factor registers 4 to 7 (DTFR4 to DTFR7)

The DTFRn register is an 8-bit register that controls the DMA transfer start trigger of DMA channel n via interrupt requests from on-chip peripheral I/O (n = 4 to 7).

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

Address
DTFR4: $FFFFF348_H$
DTFR5: $FFFFF34A_H$
DTFR6: $FFFFF34C_H$
DTFR7: $FFFFF34E_H$

Initial value 00_H

Caution
1. Do not set the same transfer start factor by different DTFRn registers.
2. Do not rewrite the DTFRn register until a started DMA transfer ends (corresponding DTCRn register value is 00_H).
3. Write the DTFRn register before setting the corresponding DTCRn register. According to the present transfer start factor in the DTFRn register a DMA might be started when the DTCRn register is written previously.

DTFRn

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

Table 6-6 DTFRn register contents

Bit position	Bit name	Function
2 to 0	IFCn[2:0]	Controls the DMA transfer start trigger of DMA channel n.

IFCn2	IFCn1	IFCn0	DMA Transfer Start Factor	
			n = 4, 5	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[a]	INTCB1T[a]
1	0	1	INTC30	INTC30
1	1	0	INTC31[a]	INTC31[a]
1	1	1	Setting prohibited	

a) not available on µPD70F3447

6.3 DMA Channel Priorities

The DMA channel priorities are fixed as follows.

DMA channel 0 > DMA channel 1 > DMA channel 2 >... > DMA channel 7

6.4 DMA Operation

6.4.1 DMA transfer of A/D converter result registers (ADC0, ADC1)

The DMAC has two dedicated channels to support DMA transfer for both A/D converters independently, DMA channel 0 for A/D converter 0 and DMA channel 1 for A/D converter 1. As DMA trigger factor, which requests and starts the DMA transfer, the end of conversion interrupt signal of the corresponding A/D converter is pre-defined (INTADn) (n = 0, 1).

For each DMA trigger the data will be transferred from the A/D conversion result register for DMA (ADDMAn) into the internal RAM specified as destination. While the source transfer address is fixed to the ADDMAn register of the corresponding A/D converter (ADCn), the destination start address can be set up to any even address in the internal RAM.

When the DMA transfer count of a DMA channel terminates, the DMA transfer is stopped and a termination interrupt is generated. The maximum DMA transfer count is 256.

Since the DMA transfer is performed for each finished A/D conversion, it is possible to transfer more than conversion results of one A/D converter scan sequence. However, the user has to take care that the number of transfer counts complies with the product of A/D converter scan area size and the number of A/D converter start triggers.

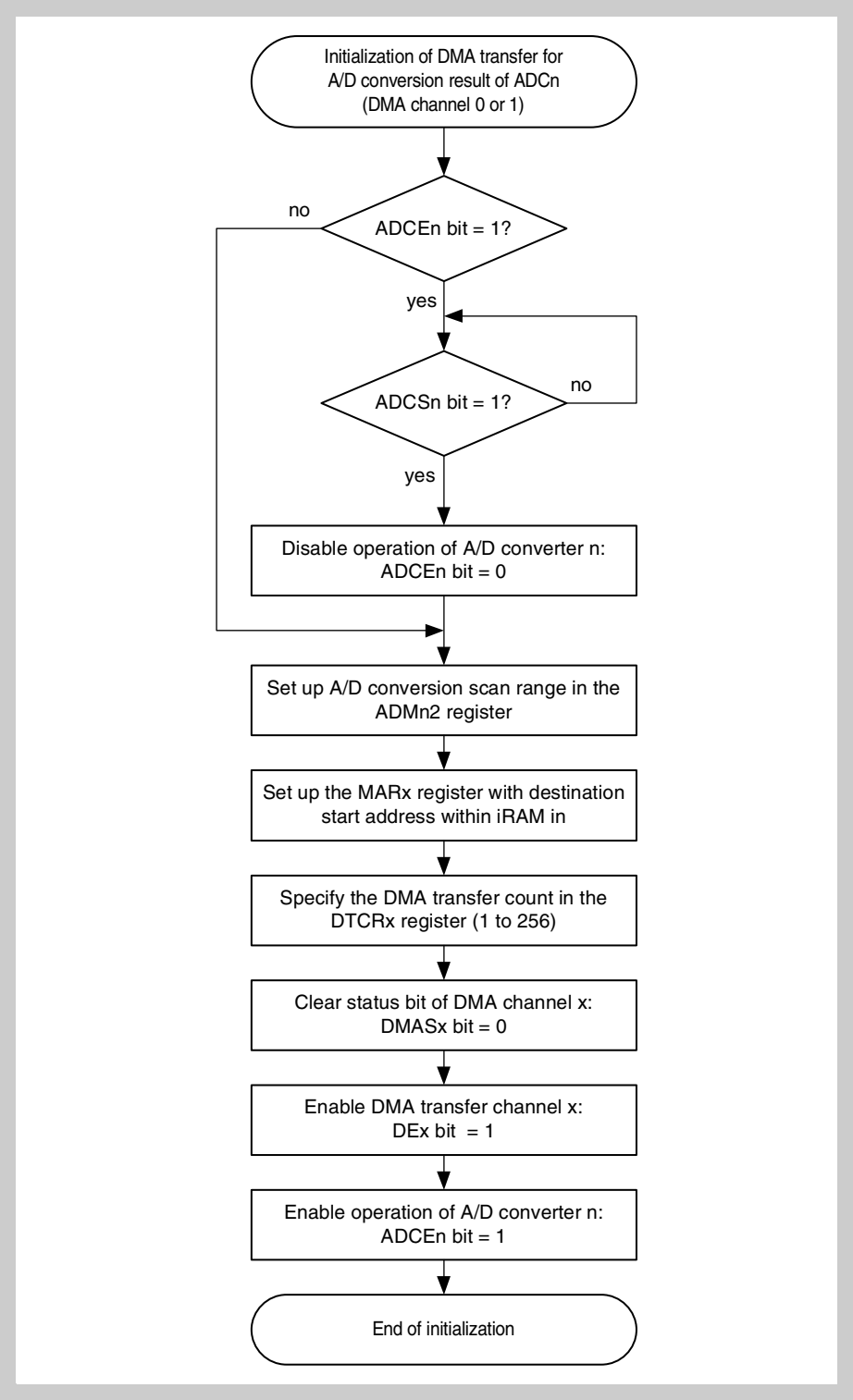


Figure 6-1 Initialization of DMA transfer for A/D conversion result

Remark n = 0, 1 (number of ADC channel)
x = n (number of corresponding DMA channel)

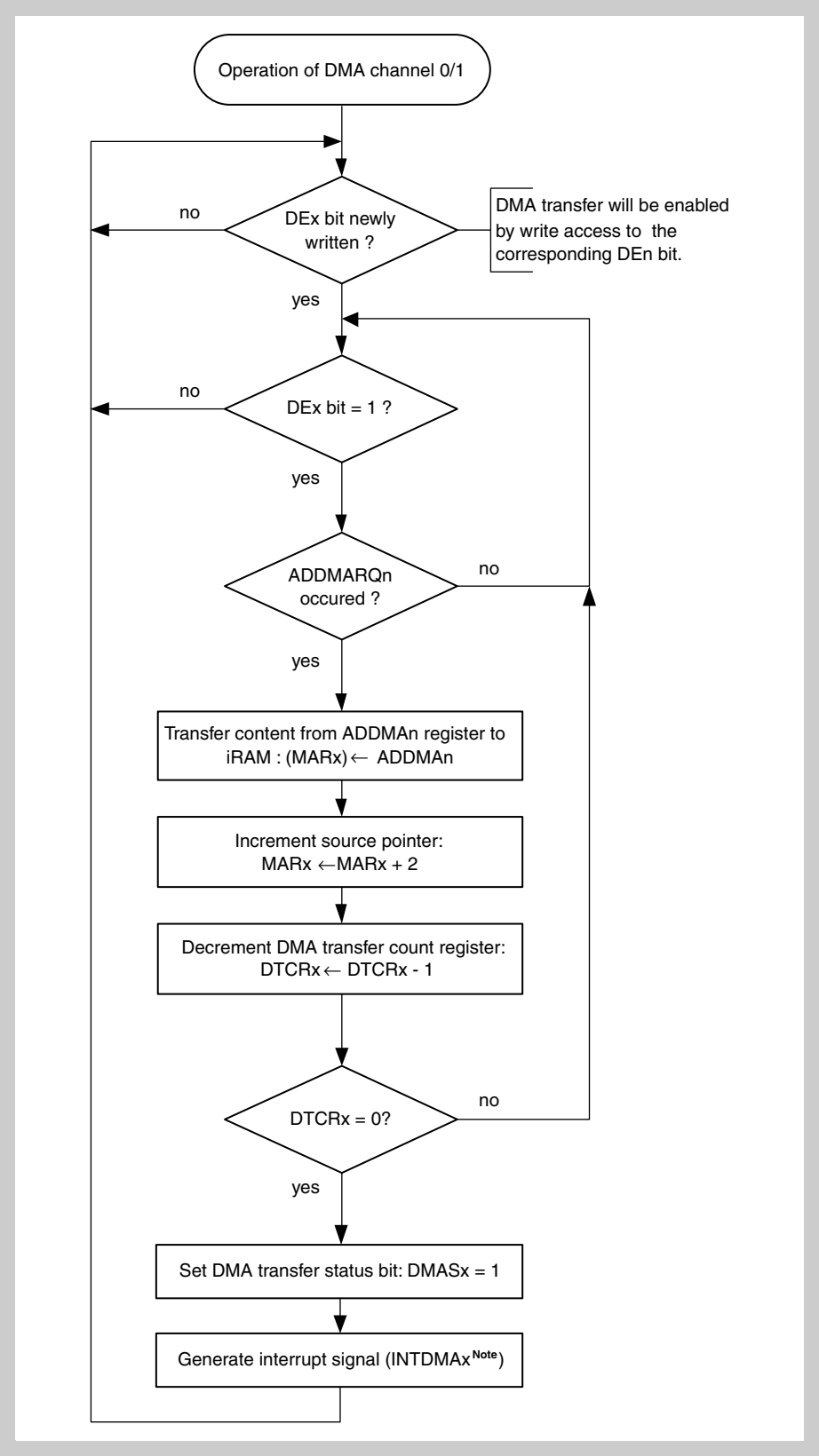


Figure 6-2 Operation of DMA channel 0/1

Note DMA transfer completion interrupt has the same interrupt vector address as the corresponding A/D conversion completion interrupt (INTADn), and replaces that interrupt.

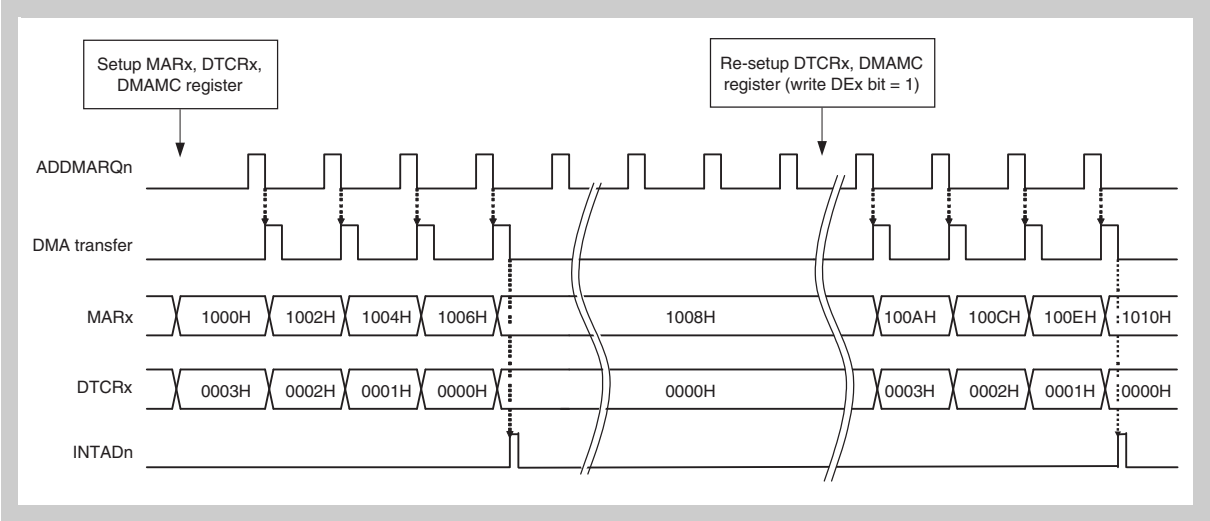


Figure 6-3 DMA channel 0 and 1 trigger signal timing

Remarks

1. The DMA request by ADDMARQ is disregarded after INTDMA is generated, and the DMA transfer is not restarted automatically. Write “1” in the corresponding DEx bit of the DMAMC register again to enable the next transfer of DMA channel x. The DEx bit is not cleared by hardware.

2. n = 0, 1 (number of A/D converter channel)
   x = n (number of corresponding DMA channel)

6.4.2 DMA transfer of PWM timer reload (TMR0, TMR1)

The DMAC has two dedicated channels to support DMA transfer for both PWM timers TMRn independently, DMA channel 2 for TMR0 and DMA channel 3 for TMR1. As DMA trigger factor, which requests and starts the DMA transfer, two corresponding timer interrupt signals are pre-defined (INTTRnOD or INTTRnCD). These are the same signals as for reloading the internal buffer compare registers by the contents of the capture/compare registers TRnCCRm (n = 0, 1)(m = 0 to 5).

For each DMA trigger data will be transferred from internal RAM to the capture/compare registers of corresponding timer TMRn. The destination start address of the TMRn register (TRnCC0, TRnCC2 to TRnCC5) can be set up by the SARx register, as well as the source start address in the internal RAM by the MARx register. The destination end address is always fixed to TRnCC1 register, which also enables the buffer reload in the timer TMRn period (refer to *Table 6-7*).

The DMA transfer count is defined by the destination start and end address. However, an additionally DMA trigger count is available, which can be specified in the DTCRx register from 1 to 256. After decrementing the DTCRx register the DMAC will be prepared for a new DMA transfer from internal RAM to the timer TMRn registers until the DMA trigger count terminates (DTCRx register = 0).

Table 6-7 Timer TMR address mapping for DMA transfer

DMA Transfer Destination Address			DMA Transfer Source Address
TMRn registers	Address Offset		
TRnCCR5	00H	Selectable as start address	Any even address in internal RAM area
TRnCCR4	02H		
TRnCCR0	08H		
TRnCCR3	0AH		
TRnCCR2	0CH		
TRnCCR1	0EH	Always end address	

Remark

n = 0, 1	(number of TMR channel)
m = 0 to 5	(number of TMR capture/compare register)
x = n + 2	(number of corresponding DMA channel)

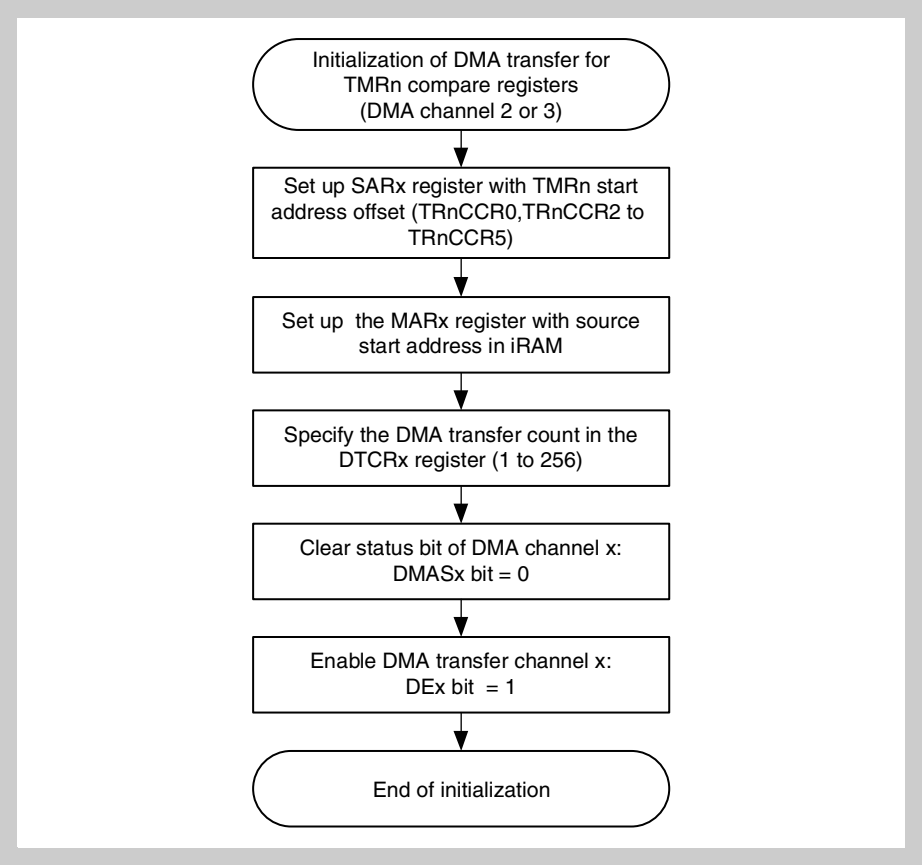


Figure 6-4 Initialization of DMA transfer for TMRn compare registers

Remark n = 0, 1 (number TMR channel)
x = n + 2 (number corresponding DMA channel)

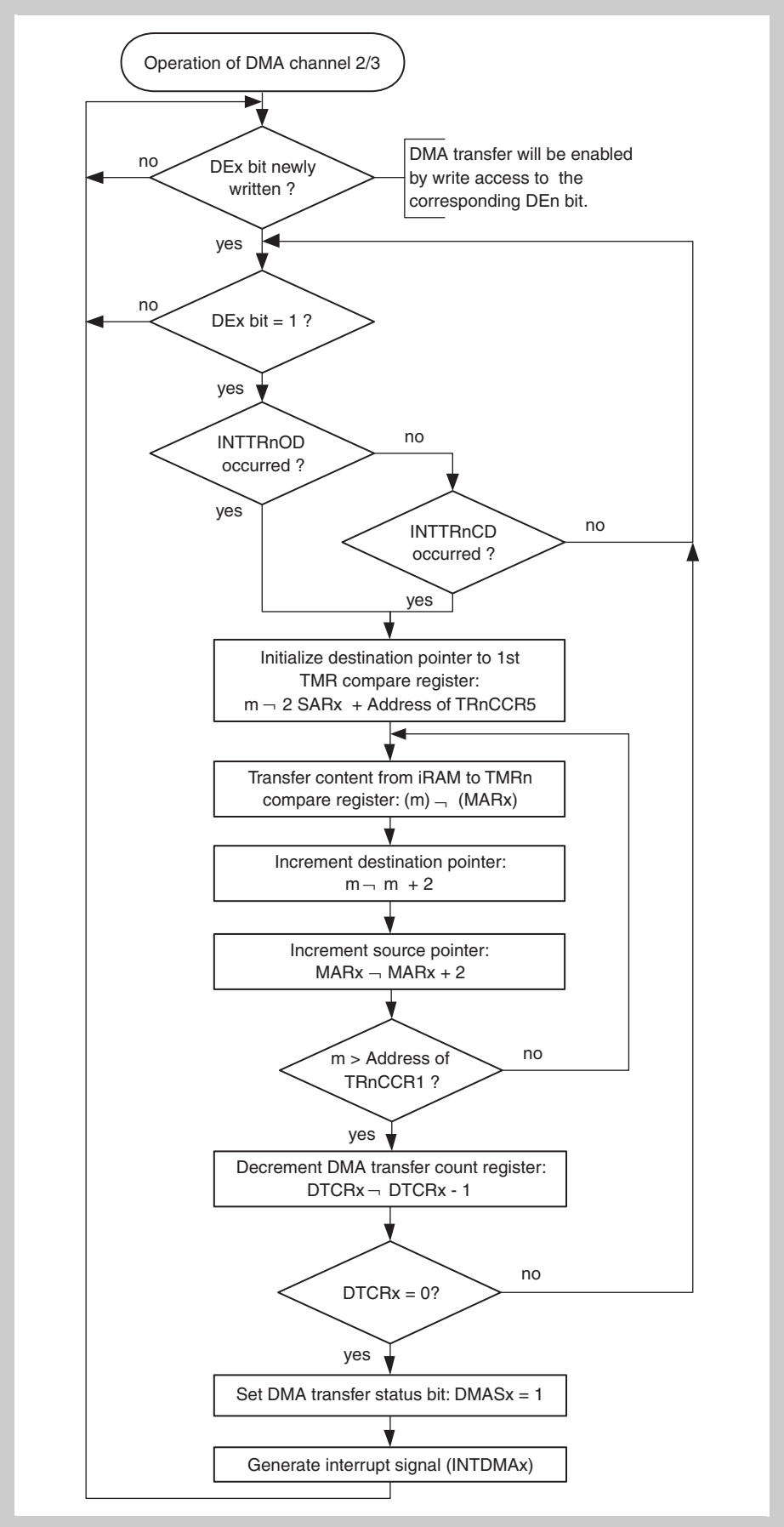


Figure 6-5 Operation of DMA channel 2/3

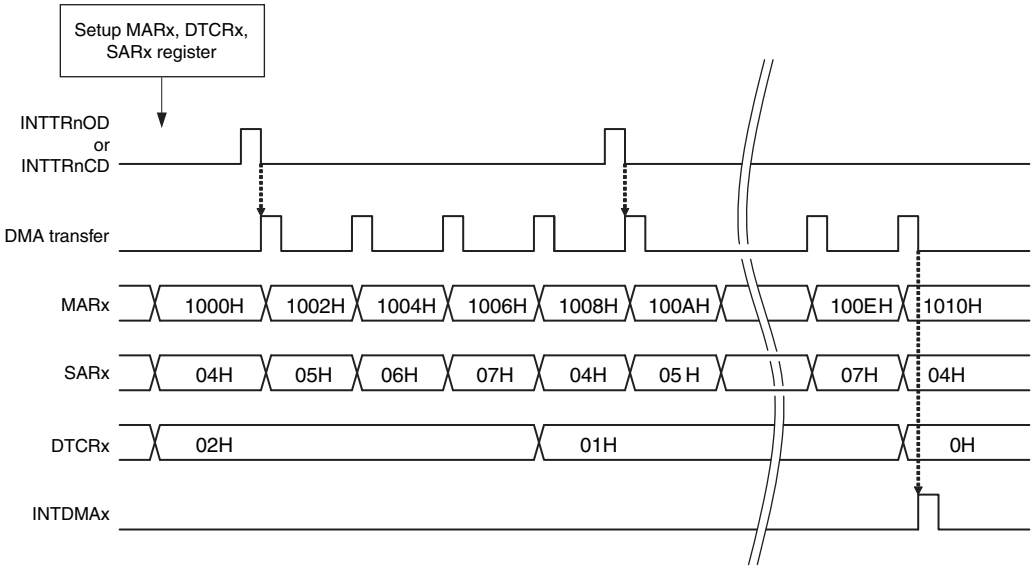


Figure 6-6 DMA Channel 2 and 3 trigger signal timing

Remarks

1. The DMA request by INTTRnOD or INTTRnCD is disregarded after INTDMAx is generated, and the DMA transfer is not restarted automatically. Write “1” in the corresponding DEx bit of the DMAMC register again to enable the next transfer of DMA channel x. The DEx bit is not cleared by hardware.

2. n = 0, 1 (number of TMR channel)
   x = n+2 (number of corresponding DMA channel)

6.4.3 DMA transfer of serial interfaces

(1) Serial data reception with DMA transfer

The DMAC has two dedicated channels (4 and 5) to support the serial data reception. Each of both channels can be assigned to a serial interface (CSI30, CSI31, CSIB0, CSIB1, UARTC0, UARTC1). As DMA trigger factor, which requests and starts the DMA transfer, the corresponding interrupt signal at the end of reception is pre-defined (refer to *Table 6-8*).

For each DMA trigger the data will be transferred from the corresponding serial reception register to internal RAM. Depending on the serial interface the transfer data size can be set to 8 or 16 bits (refer to *Table 6-8*).

In case of 8 bits transfer data size, the destination address is incremented by 1 for each occurrence of DMA trigger. When selecting 16 bits transfer data size the destination address must be even, and is incremented by 2 for each DMA trigger.

When the DMA transfer count of a DMA channel terminates, the DMA transfer is stopped and a DMA completion interrupt is generated. The maximum DMA transfer count is 256.

Note Serial interfaces CSI31 and CSIB1 are not available on µPD70F3447.

Table 6-8 DMA configuration of serial data reception

Serial Interface	DMA Trigger Factor	Transfer Data Size	Source	Destination
CSI30	INTC30	8 bits	SIRB0L	Any iRAM address
		16 bits	SIRB0	Any even iRAM address
CSI31[a]	INTC31	8 bits	SIRB1L	Any iRAM address
		16 bits	SIRB1	Any even iRAM address
CSIB0	INTCB0T	8 bits	CB0RXL	Any iRAM address
		16 bits	CB0RX	Any even iRAM address
CSIB1[a]	INTCB1T	8 bits	CB1RXL	Any iRAM address
		16 bits	CB1RX	Any even iRAM address
UARTC0	INTUC0T	8 bits	UC0RX	Any iRAM address
		16 bits	Setting prohibited	
UARTC1	INTUC1T	8 bits	UC1RX	Any iRAM address
		16 bits	Setting prohibited	

a) not available on µPD70F3447

The initialization procedure of the DMA transfer in case of serial data reception is shown in *Figure 6-7*.

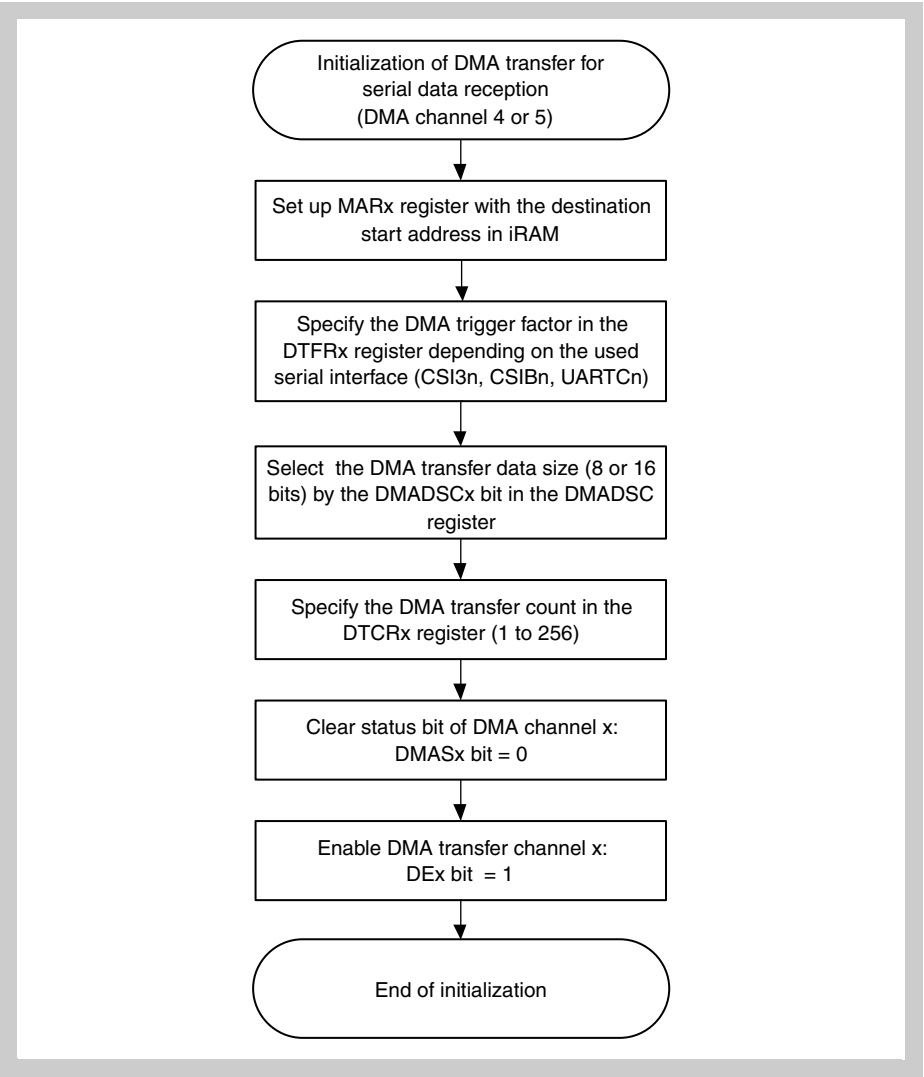


Figure 6-7 Initialization of DMA transfer for serial data reception

Remark n = 0, 1 (number of serial interface channel)
x = 4, 5 (number corresponding DMA channel)

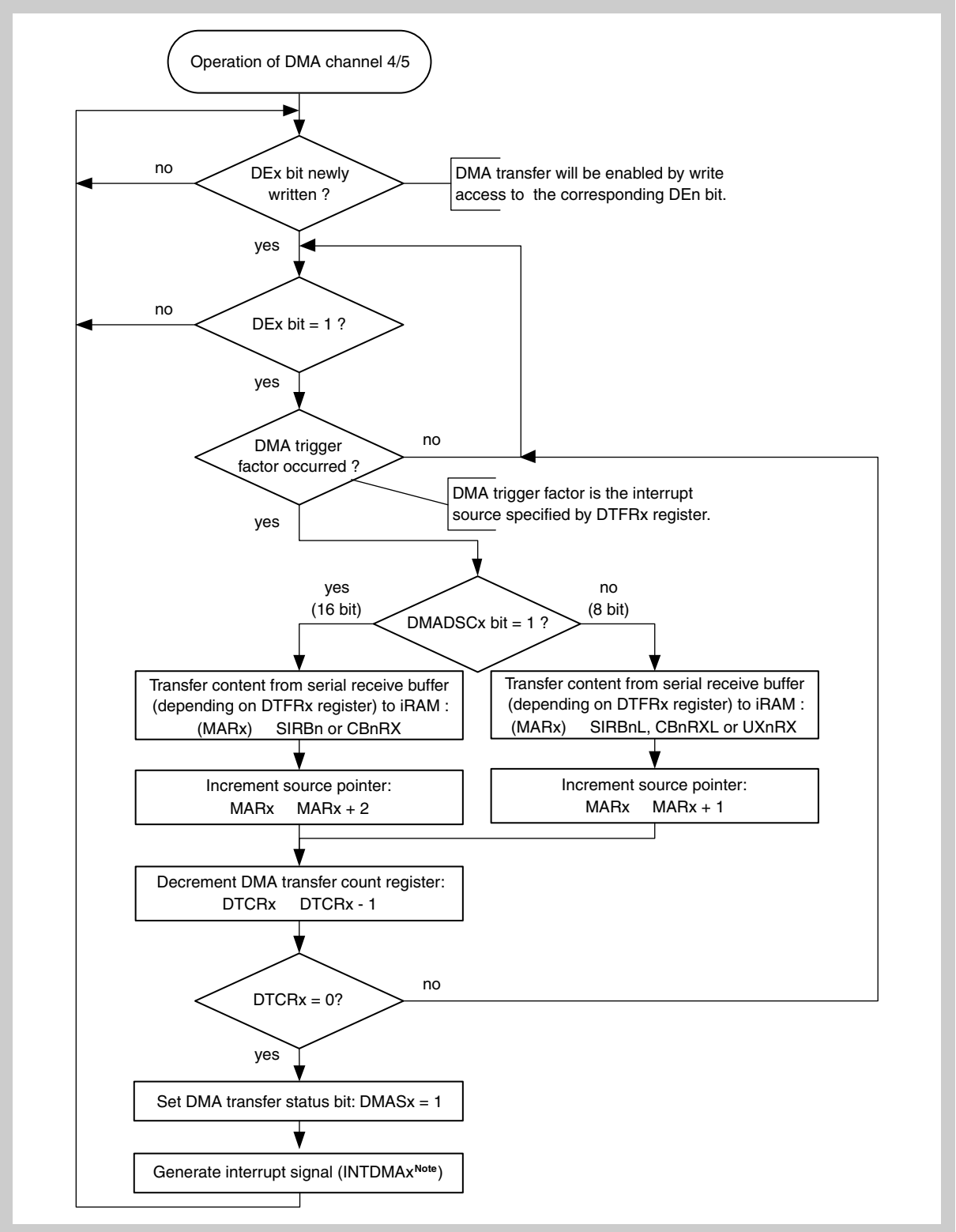


Figure 6-8 Operation of DMA channel 4/5

Note DMA transfer completion interrupt has the same interrupt vector address as the corresponding reception completion interrupt specified by DTFRx register, and replaces that interrupt.

Remark n = 0, 1 (number of serial interface channel)
x = 4, 5 (number of corresponding DMA channel)

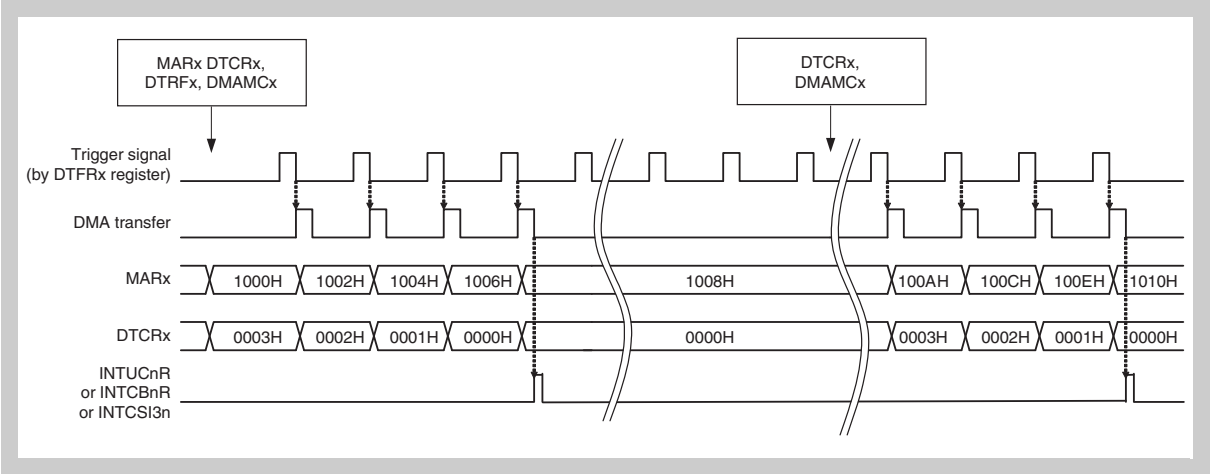


Figure 6-9 DMA Channel 4 and 5 trigger signal timing

Remark n = 0, 1 (number of serial interface channel)
x = 4, 5 (number of corresponding DMA transfer channel)

(2) Serial data transmission with DMA transfer

The DMAC has two dedicated channels (6 and 7) to support the serial data transmission. Each of both channels can be assigned to a serial interface (CSI30, CSI31, CSIB0, CSIB1, UARTC0, UARTC1). As DMA trigger factor, which requests and starts the DMA transfer, the corresponding transmission enable interrupt signal is pre-defined (refer to *Table 6-9*).

For each DMA trigger the data will be transferred from internal RAM to the corresponding serial transmit register. Depending on the serial interface the transfer data size can be set to 8 or 16 bits (refer to *Table 6-9*).

In case of 8 bits transfer data size, the source address is incremented by 1 for each occurrence of DMA trigger. When selecting 16 bits transfer data size the source address must be even, and is incremented by 2 for each DMA trigger.

When the DMA transfer count of a DMA channel terminates, the DMA transfer is stopped and a DMA completion interrupt is generated. The maximum DMA transfer count is 256.

Note Serial interfaces CSI31 and CSIB1 are not available on µPD70F3447.

Table 6-9 DMA Configuration of Serial Data Transmission

Serial Interface	DMA Trigger Factor	Transfer Data Size	Source	Destination
CSI30 [a]	INTC30	8 bits	Any iRAM address	SFDB0L
		16 bits	Any even iRAM address	SFDB0
CSI31 [a, b]	INTC31	8 bits	Any iRAM address	SFDB1L
		16 bits	Any even iRAM address	SFDB1
CSIB0	INTCB0T	8 bits	Any iRAM address	CB0TXL
		16 bits	Any even iRAM address	CB0TX
CSIB1 [b]	INTCB1T	8 bits	Any iRAM address	CB1TXL
		16 bits	Any even iRAM address	CB1TX
UARTC0	INTUC0T	8 bits	Any iRAM address	UC0TX
		16 bits	Setting prohibited	
UARTC1	INTUC1T	8 bits	Any iRAM address	UC1TX
		16 bits	Setting prohibited	

a) The serial peripheral chip select lines SCS0 to SCS3 are not supported by DMA transfer.
b) not available on µPD70F3447

The initialization procedure of the DMA transfer in case of serial data transmission is shown in *Figure 6-10*.

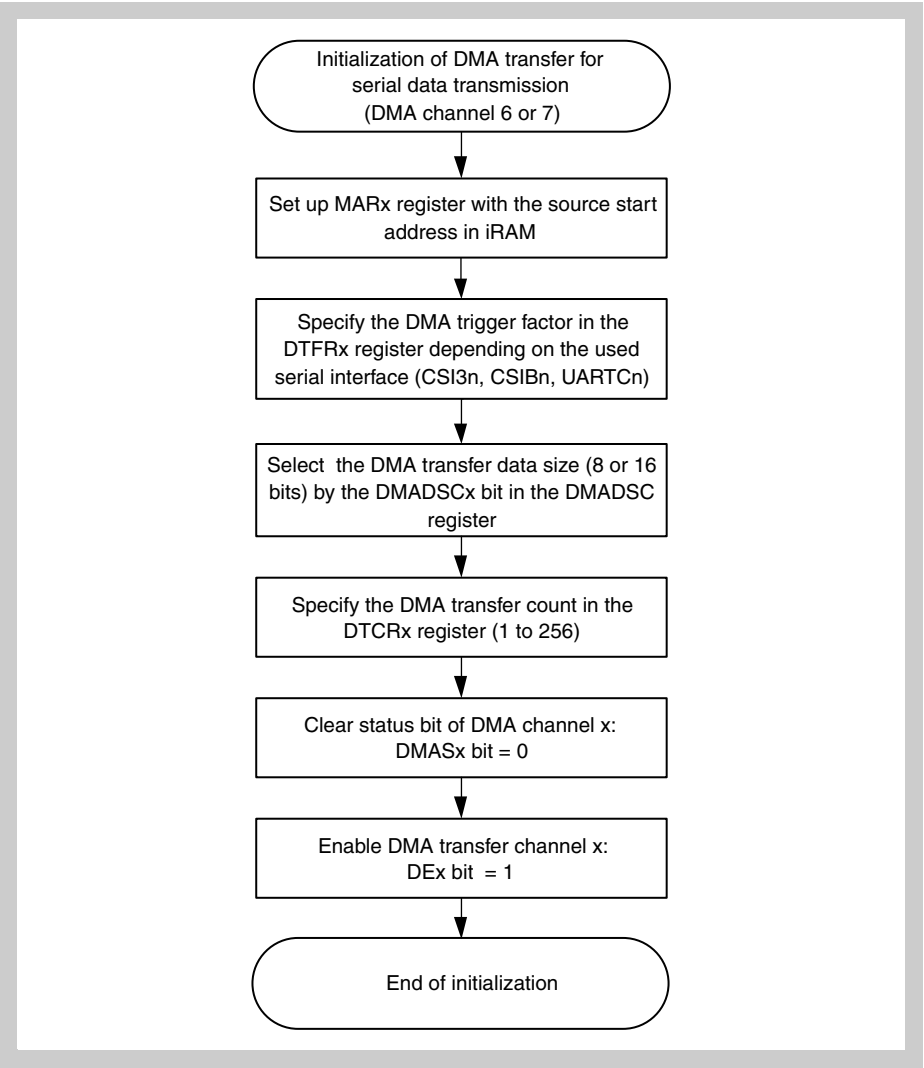


Figure 6-10 Initialization of DMA transfer for serial data transmission

Remark n = 0, 1 (number of serial interface channel)
x = 6, 7 (number of corresponding DMA channel)

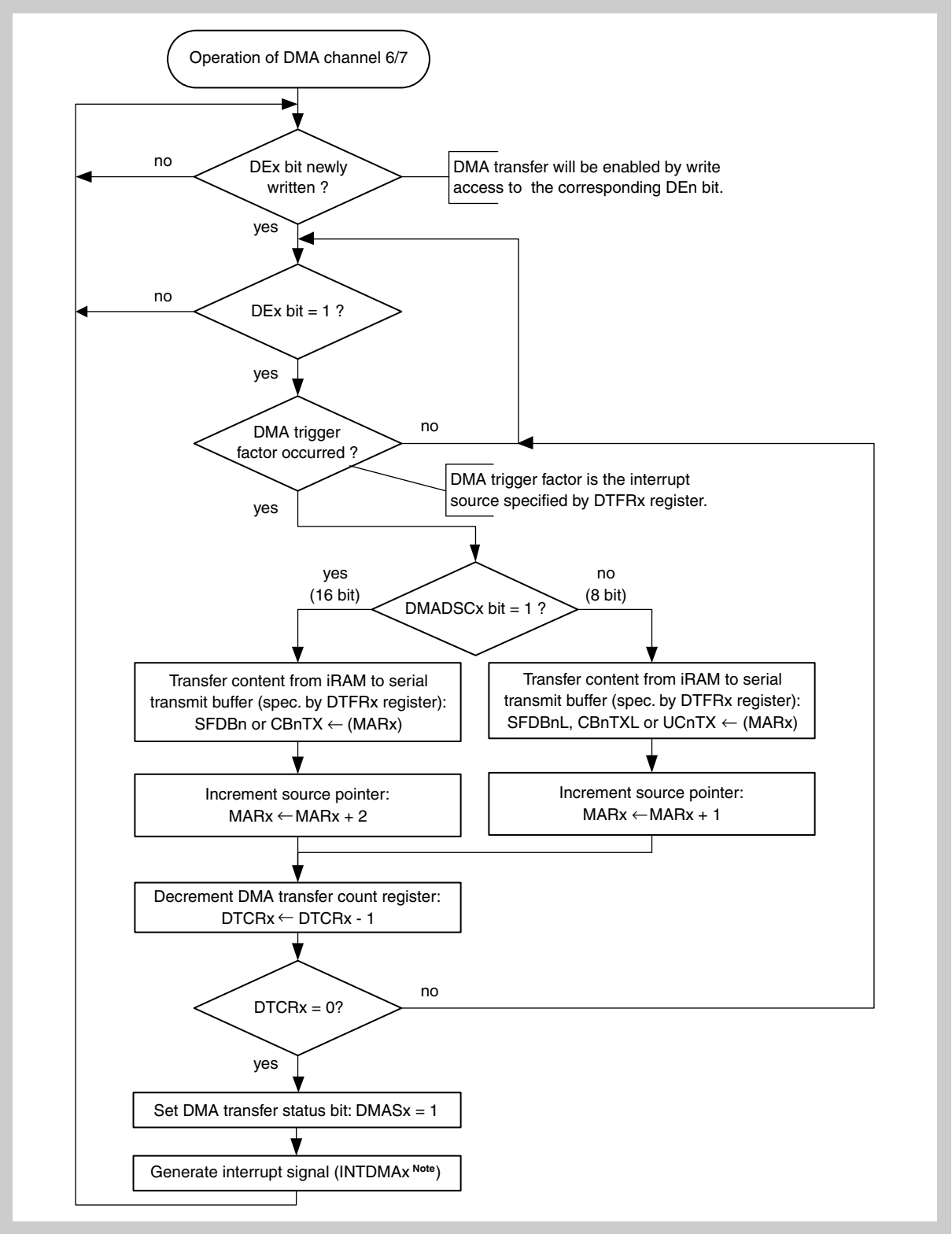


Figure 6-11 Operation of DMA channel 6/7

Note DMA transfer completion interrupt has the same interrupt vector address as the corresponding transmission start interrupt specified by DTFRx register, and replaces that interrupt.

Remark n = 0, 1 (number of serial interface channel)
x = 6, 7 (number of corresponding DMA transfer channel)

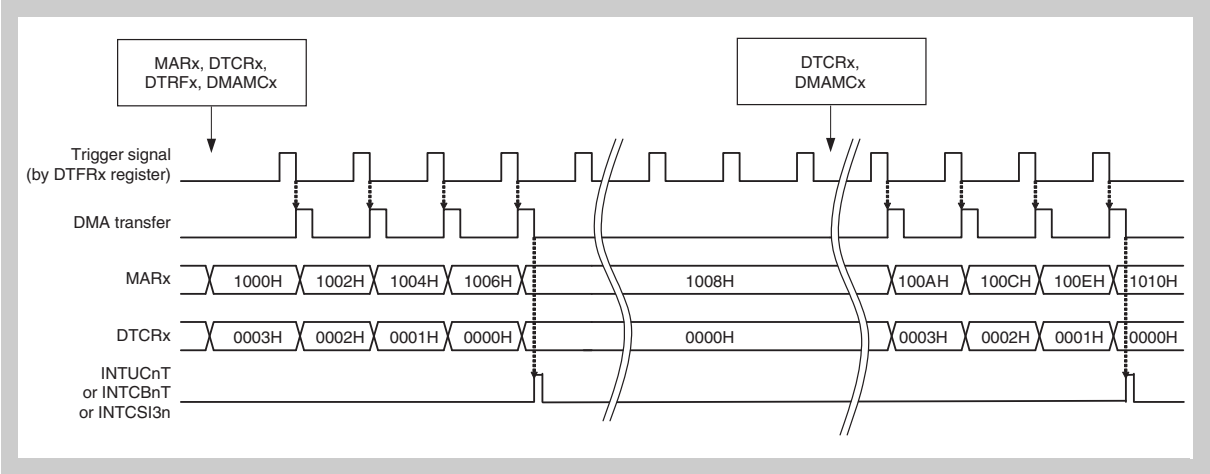


Figure 6-12 DMA Channel 6 and 7 trigger signal timing

Remark n = 0, 1 (number of serial interface channel)
x = 6, 7 (number of corresponding DMA transfer channel)

6.4.4 Forcible termination of DMA transfer

A once started DMA transfer can be forcible terminated when the corresponding DEn bit in the DMAMC register is cleared (0). However, if the DEn bit is cleared while DMA transferring, an once started data transfer is stopped first after it has been finished (refer to *Figure 6-13*).

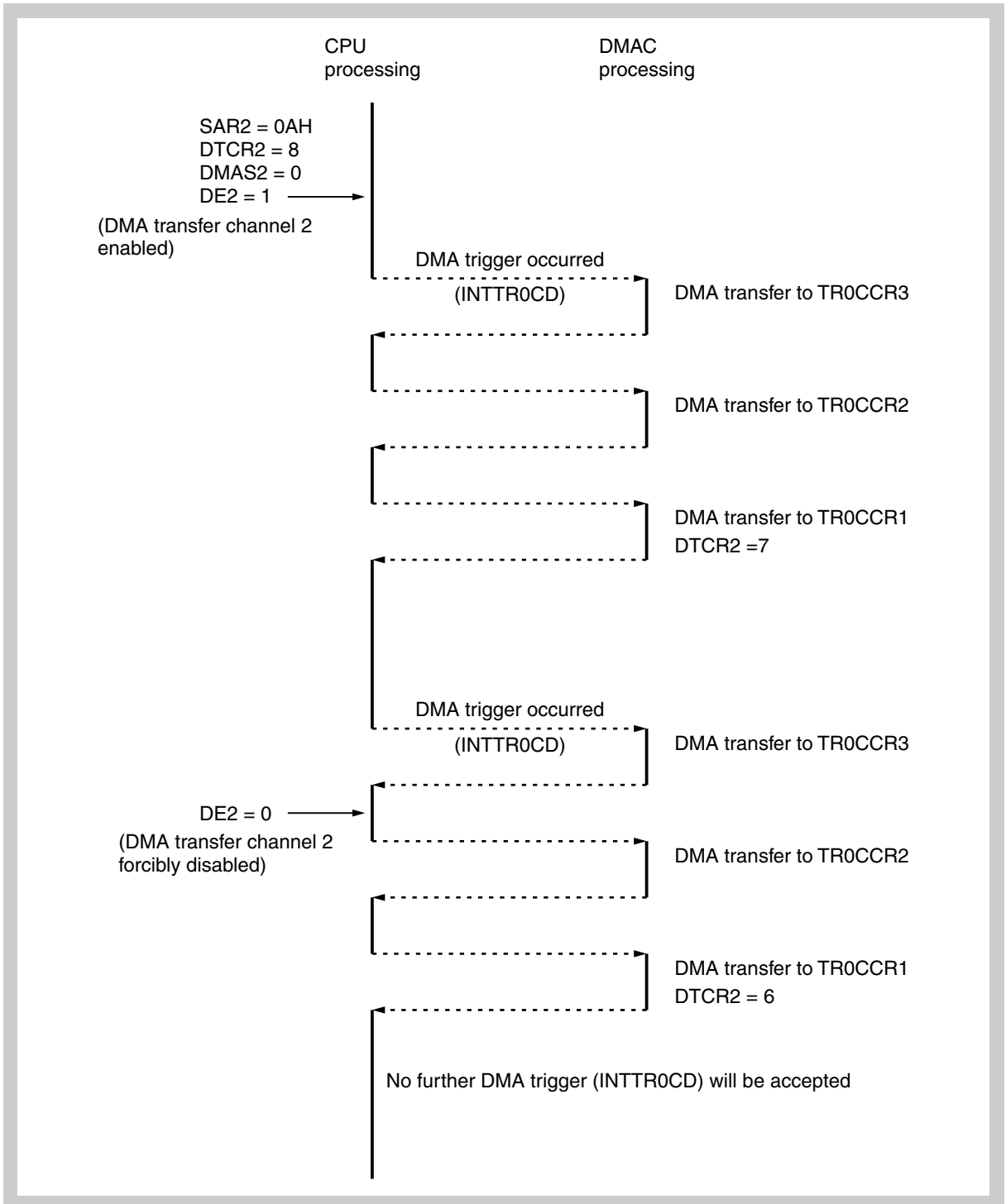


Figure 6-13 CPU and DMA controller processing of DMA transfer termination (example)

6.5 DMA Interrupt Function

The peripheral I/O interrupts of the A/D converters and the serial interfaces, which serve as DMA trigger factors, are shared with the DMA transfer completion interrupt of the corresponding channel n (INTDMAn) (n = 0, 1, 4 to 7). When a DMA channel is enabled the specified peripheral I/O interrupt is no longer applied to the interrupt controller. Instead of it the corresponding DMA transfer completion interrupt is applied to the appropriate interrupt handler address.

In opposite to the other interrupts serving as DMA trigger factors, the TMR0 interrupts INTTR0OD and INTTR0CD, and the TMR1 interrupts INTTR1OD and INTTR1CD respectively, are not shared with DMA transfer completion interrupt of channel 2 (INTDMA2) and channel 3 (INTDMA3) respectively. These DMA completion interrupts have dedicated entries in the interrupt source list (refer to *"Interrupt/exception source list" on page 172*).

Table 6-10 shows the relations between DMA trigger factors and DMA completion interrupts.

Table 6-10 Relations Between DMA Trigger Factors and DMA Completion Interrupts

DMA channel	DMA trigger factor	DMA completion interrupt			Remark
		Name	Entry	Handler Address	
0	INTAD0	INTDMA0	INTAD0	00000670_H	a
1	INTAD1	INTDMA1	INTAD1	00000680_H	a
2	INTTR0CD or INTR0OD	INTDMA2	INTDMA2	$000006F0_H$	
3	INTTR1CD or INTR1OD	INTDMA3	INTDMA3	00000700_H	
4, 5	INTC30	INTDMA4, INTDMA5	INTC30	$000005E0_H$	a
	INTC31 b		INTC31 b	00000600_H	a
	INTCB0R		INTCB0R	00000580_H	a
	INTCB1R b		INTCB1R b	$000005B0_H$	a
	INTUC0R		INTUC0R	00000620_H	a
	INTUC1R		INTUC1R	00000650_H	a
6, 7	INTC30	INTDMA6, INTDMA7	INTC30	$000005E0_H$	a
	INTC31 b		INTC31 b	00000600_H	a
	INTCB0T		INTCB0T	00000570_H	a
	INTCB1T b		INTCIB1T b	$000005A0_H$	a
	INTUC0T		INTUC0T	00000630_H	a
	INTUC1T		INTUC1T	00000660_H	a

a) An interrupt request is not generated for a signal, which serves as DMA trigger factor. Instead of this the defined DMA completion interrupt request is executed on the same interrupt entry address of the DMA trigger factor.

b) not available on µPD70F3447

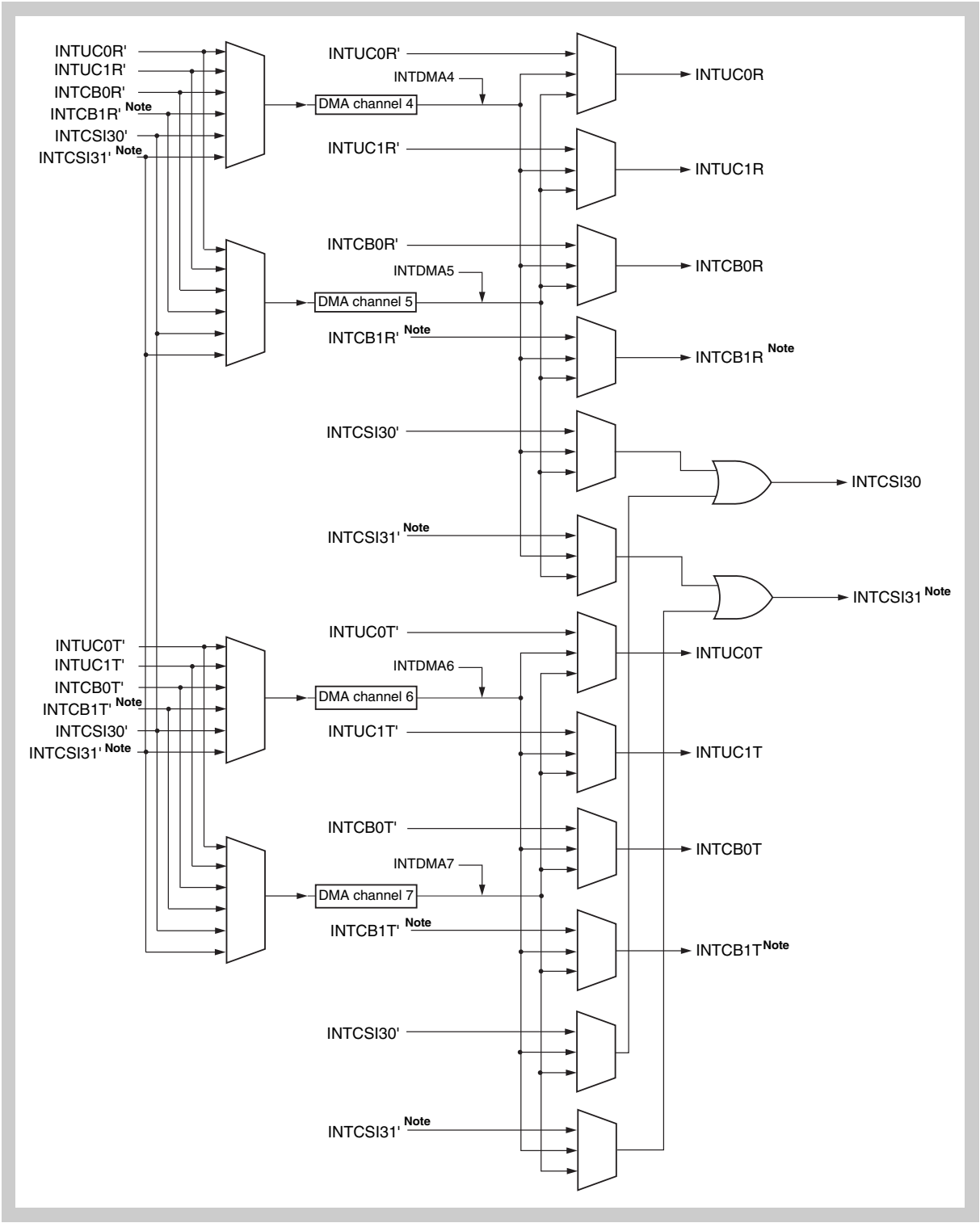


Figure 6-14 Correlation between serial I/O interface interrupts and DMA completion interrupts

Note Not available on μPD70F3447.

Remark Interrupt signals with quote mark (') are signals, which are directly connected from the corresponding serial interface.
Interrupt signals without quote mark are provided to the interrupt controller.

Chapter 7 Interrupt/Exception Processing Function

The V850E/PH2 microcontroller is provided with a dedicated interrupt controller (INTC) for interrupt servicing, which realizes a high-performance interrupt function that can service interrupt requests from a total of up to 107 sources.

An interrupt is an event that occurs asynchronously (independently of program execution), and an exception is an event that occurs synchronously (dependently on program execution). Generally, an exception takes precedence over an interrupt.

The V850E/PH2 microcontroller can process interrupt requests from the internal peripheral hardware and external sources. Moreover, exception processing can be started (exception trap) by the TRAP instruction (software exception) or by generation of an exception event (fetching of an illegal op code).

7.1 Features

- Interrupts
  - Non-maskable interrupt: 1 source
  - Maskable interrupt:
  - 106 sources (μPD70F3187)
  - 91 sources (μPD70F3447)
  - 8 levels programmable priorities
  - Mask specification for the interrupt request according to priority
  - Mask can be specified to each maskable interrupt request.
  - Valid edge for detection of external interrupt request signal can be specified.

- Exceptions
  - Software exceptions: 32 sources
  - Exception trap: 1 source (illegal op code exception)

Interrupt/exception sources are listed in *Table 7-1*.

Table 7-1 Interrupt/exception source list (1/4)

Type	Classification	Interrupt/Exception Source				Default Priority[a]	Exception Code	Handler Address
		Name	Control Register	Generating Source	Gener. Unit			
Reset	Interrupt	RESET	–	$\overline{\text{RESET}}$ input	Pin	–	0000H	00000000H
Non-maskable	Interrupt	NMI	–	NMI input	Pin	–	0010H	00000010H
Software exception	Exception	TRAP0n[b]	–	TRAP instruction	–	–	004nH[b]	00000040H
	Exception	TRAP1n[b]	–	TRAP instruction	–	–	005nH[b]	00000050H
Exception trap	Exception	ILGOP[c]/ DBTRAP	–	Illegal opcode/ DBTRAP instruction	–	–	0060H	00000060H
Maskable	Interrupt	INTP0	PIC0	INTP0 valid edge input	Pin	0	0080H	00000080H
	Interrupt	INTP1	PIC1	INTP1 valid edge input	Pin	1	0090H	00000090H
	Interrupt	INTP2	PIC2	INTP2 valid edge input	Pin	2	00A0H	000000A0H
	Interrupt	INTP3	PIC3	INTP3 valid edge input	Pin	3	00B0H	000000B0H
	Interrupt	INTP4	PIC4	INTP4 valid edge input	Pin	4	00C0H	000000C0H
	Interrupt	INTP5	PIC5	INTP5 valid edge input	Pin	5	00D0H	000000D0H
	Interrupt	INTP6	PIC6	INTP6 valid edge input	Pin	6	00E0H	000000E0H
	Interrupt	INTP7	PIC7	INTP7 valid edge input	Pin	7	00F0H	000000F0H
	Interrupt	INTP8	PIC8	INTP8 valid edge input	Pin	8	0100H	00000100H
	Interrupt	INTP9	PIC9	INTP9 valid edge input	Pin	9	0110H	00000110H
	Interrupt	INTP10	PIC10	INTP10 valid edge input	Pin	10	0120H	00000120H
	Interrupt	INTP11	PIC11	INTP11 valid edge input	Pin	11	0130H	00000130H
	Interrupt	INTP12	PIC12	INTP12 valid edge input	Pin	12	0140H	00000140H
	Interrupt	INTTR0OV	PIC13	TR0CNT overflow	TMR0	13	0150H	00000150H
	Interrupt	INTTR0CC0	PIC14	TR0CCR0 match	TMR0	14	0160H	00000160H
	Interrupt	INTTR0CC1	PIC15	TR0CCR1 match	TMR0	15	0170H	00000170H
	Interrupt	INTTR0CC2	PIC16	TR0CCR2 match	TMR0	16	0180H	00000180H
	Interrupt	INTTR0CC3	PIC17	TR0CCR3 match	TMR0	17	0190H	00000190H
	Interrupt	INTTR0CC4	PIC18	TR0CCR4 match	TMR0	18	01A0H	000001A0H
	Interrupt	INTTR0CC5	PIC19	TR0CCR5 match	TMR0	19	01B0H	000001B0H
	Interrupt	INTTR0CD	PIC20	TR0CNT top reversal	TMR0	20	01C0H	000001C0H
	Interrupt	INTTR0OD	PIC21	TR0CNT bottom reversal	TMR0	21	01D0H	000001D0H
	Interrupt	INTTR0ER	PIC22	TMR0 error detection	TMR0	22	01E0H	000001E0H
	Interrupt	INTTR1OV	PIC23	TR1CNT overflow	TMR1	23	01F0H	000001F0H
	Interrupt	INTTR1CC0	PIC24	TIR10 capture input/ TR1CCR0 match	TMR1	24	0200H	00000200H
	Interrupt	INTTR1CC1	PIC25	TIR11 capture input/ TR1CCR1 match	TMR1	25	0210H	00000210H
	Interrupt	INTTR1CC2	PIC26	TIR12 capture input/ TR1CCR2 match	TMR1	26	0220H	00000220H
	Interrupt	INTTR1CC3	PIC27	TIR13 capture input/ TR1CCR3 match	TMR1	27	0230H	00000230H
	Interrupt	INTTR1CC4	PIC28	TR1CCR4 match	TMR1	28	0240H	00000240H
	Interrupt	INTTR1CC5	PIC29	TR1CCR5 match	TMR1	29	0250H	00000250H
	Interrupt	INTTR1CD	PIC30	TR1CNT top reversal	TMR1	30	0260H	00000260H
	Interrupt	INTTR1OD	PIC31	TR1CNT bottom reversal	TMR1	31	0270H	00000270H
	Interrupt	INTTR1ER	PIC32	TMR1 error detection	TMR1	32	0280H	00000280H

Table 7-1 Interrupt/exception source list (2/4)

Type	Classification	Interrupt/Exception Source				Default Priority[a]	Exception Code	Handler Address
		Name	Control Register	Generating Source	Gener. Unit			
Maskable	Interrupt	INTT0OV	PIC33	TMT0 overflow	TMT0	33	0290H	00000290H
	Interrupt	INTT0CC0	PIC34	TIT00 capture input/ TT0CCR0 match	TMT0	34	02A0H	000002A0H
	Interrupt	INTT0CC1	PIC35	TIT01 capture input/ TT0CCR1 match	TMT0	35	02B0H	000002B0H
	Interrupt	INTT0EC	PIC36	TMT0 encoder clear	TMT0	36	02C0H	000002C0H
	Interrupt	INTT1OV	PIC37	TMT1 overflow	TMT1	37	02D0H	000002D0H
	Interrupt	INTT1CC0	PIC38	TIT10 capture input/ TT1CCR0 match	TMT1	38	02E0H	000002E0H
	Interrupt	INTT1CC1	PIC39	TIT11 capture input/ TT1CCR1 match	TMT1	39	02F0H	000002F0H
	Interrupt	INTT1EC	PIC40	TMT1 encoder clear	TMT1	40	0300H	00000300H
	Interrupt	INTP0OV	PIC41	TMP0 overflow	TMP0	41	0310H	00000310H
	Interrupt	INTP0CC0	PIC42	TIP00 capture input/ TP0CCR0 match	TMP0	42	0320H	00000320H
	Interrupt	INTP0CC1	PIC43	TIP01 capture input/ TP0CCR1 match	TMP0	43	0330H	00000330H
	Interrupt	INTP1OV	PIC44	TMP1 overflow	TMP1	44	0340H	00000340H
	Interrupt	INTP1CC0	PIC45	TIP10 pin/ TP1CCR0 match	TMP1	45	0350H	00000350H
	Interrupt	INTP1CC1	PIC46	TIP11 capture input/ TP1CCR1 match	TMP1	46	0360H	00000360H
	Interrupt	INTP2OV	PIC47	TMP2 overflow	TMP2	47	0370H	00000370H
	Interrupt	INTP2CC0	PIC48	TIP20 capture input/ TP2CCR0 match	TMP2	48	0380H	00000380H
	Interrupt	INTP2CC1	PIC49	TIP21capture input/ TP2CCR1 match	TMP2	49	0390H	00000390H
	Interrupt	INTP3OV	PIC50	TMP3 overflow	TMP3	50	03A0H	000003A0H
	Interrupt	INTP3CC0	PIC51	TIP30 capture input/ TP3CCR0 match	TMP3	51	03B0H	000003B0H
	Interrupt	INTP3CC1	PIC52	TIP31 capture input/ TP3CCR1 match	TMP3	52	03C0H	000003C0H
	Interrupt	INTP4OV	PIC53	TMP4 overflow	TMP4	53	03D0H	000003D0H
	Interrupt	INTP4CC0	PIC54	TIP40 capture input/ TP4CCR0 match	TMP4	54	03E0H	000003E0H
	Interrupt	INTP4CC1	PIC55	TIP41 capture input/ TP4CCR1 match	TMP4	55	03F0H	000003F0H
	Interrupt	INTP5OV	PIC56	TMP5overflow	TMP5	56	0400H	00000400H
	Interrupt	INTP5CC0	PIC57	TIP50 capture input/ TP5CCR0 match	TMP5	57	0410H	00000410H
	Interrupt	INTP5CC1	PIC58	TIP51 capture input/ TP5CCR1 match	TMP5	58	0420H	00000420H
	Interrupt	INTP6OV	PIC59	TMP6 overflow	TMP6	59	0430H	00000430H
	Interrupt	INTP6CC0	PIC60	TIP60 capture input/ TP6CCR0 match	TMP6	60	0440H	00000440H
	Interrupt	INTP6CC1	PIC61	TIP61 capture input/ TP6CCR1 match	TMP6	61	0450H	00000450H

Table 7-1 Interrupt/exception source list (3/4)

Type	Classification	Interrupt/Exception Source				Default Priority[a]	Exception Code	Handler Address
		Name	Control Register	Generating Source	Gener. Unit			
Maskable	Interrupt	INTP7OV	PIC62	TMP7 overflow	TMP7	62	0460H	00000460H
	Interrupt	INTP7CC0	PIC63	TIP70 capture input/ TP7CCR0 match	TMP7	63	0470H	00000470H
	Interrupt	INTP7CC1	PIC64	TIP71 capture input/ TP7CCR1 match	TMP7	64	0480H	00000480H
	Interrupt	INTP8OV	PIC65	TMP8 overflow	TMP8	65	0490H	00000490H
	Interrupt	INTP8CC0	PIC66	TP8CCR0 match	TMP8	66	04A0H	000004A0H
	Interrupt	INTP8CC1	PIC67	TP8CCR1 match	TMP8	67	04B0H	000004B0H
	Interrupt	INTBRG0	PIC68	BRG0 match	BRG0	68	04C0H	000004C0H
	Interrupt	INTBRG1	PIC69	BRG1 match	BRG1	69	04D0H	000004D0H
	Interrupt	INTBRG2	PIC70	BRG2 match	AFO	70	04E0H	000004E0H
	Interrupt	INTC0ERR	PIC71	FCAN0 error	FCAN0	71	04F0H	000004F0H
	Interrupt	INTC0WUP	PIC72	FCAN0 wake up	FCAN0	72	0500H	00000500H
	Interrupt	INTC0REC	PIC73	FCAN0 bus reception	FCAN0	73	0510H	00000510H
	Interrupt	INTC0TRX	PIC74	FCAN0 bus transmission	FCAN0	74	0520H	00000520H
	Interrupt	INTC1ERR	PIC75[d]	FCAN1 error	FCAN1[d]	75	0530H	00000530H
	Interrupt	INTC1WUP	PIC76[d]	FCAN1 wake up	FCAN1[d]	76	0540H	00000540H
	Interrupt	INTC1REC	PIC77[d]	FCAN1 bus reception	FCAN1[d]	77	0550H	00000550H
	Interrupt	INTC1TRX	PIC78[d]	FCAN1 bus transmission	FCAN1[d]	78	0560H	00000560H
	Interrupt	INTCB0T	PIC79	CSIB0 transmission enable/ DMA transfer completion	CSIB0/ DMAC	79	0570H	00000570H
	Interrupt	INTCB0R	PIC80	CSIB0 reception completion/ DMA transfer completion	CSIB0/ DMAC	80	0580H	00000580H
	Interrupt	INTCB0RE	PIC81	CSIB0 receive error	CSIB0	81	0590H	00000590H
	Interrupt	INTCB1T	PIC82[d]	CSIB1 transmission enable/ DMA transfer completion	CSIB1[d]/ DMAC	82	05A0H	000005A0H
	Interrupt	INTCB1R	PIC83[d]	CSIB1 reception completion/ DMA transfer completion	CSIB1[d]/ DMAC	83	05B0H	000005B0H
	Interrupt	INTCB1RE	PIC84[d]	CSIB1 receive error	CSIB1[d]	84	05C0H	000005C0H
	Interrupt	INTC30OVF	PIC85	CSI30 overrun	CSI30	85	05D0H	000005D0H
	Interrupt	INTC30	PIC86	CSI30 transmission enable/ DMA transfer completion	CSI30/ DMAC	86	05E0H	000005E0H
	Interrupt	INTC31OVF	PIC87[d]	CSI31 overrun	CSI31[d]	87	05F0H	000005F0H
	Interrupt	INTC31	PIC88[d]	CSI31 transmission enable/ DMA transfer completion	CSI31[d]/ DMAC	88	0600H	00000600H
	Interrupt	INTUC0RE	PIC89	UARTC0 receive error	UARTC0	89	0610H	00000610H
	Interrupt	INTUC0R	PIC90	UARTC0 reception completion/ DMA transfer completion	UARTC0/ DMAC	90	0620H	00000620H
	Interrupt	INTUC0T	PIC91	UARTC0 transmission enable/ DMA transfer completion	UARTC0/ DMAC	91	0630H	00000630H
	Interrupt	INTUC1RE	PIC92	UARTC1receive error	UARTC1	92	0640H	00000640H
	Interrupt	INTUC1R	PIC93	UARTC1 reception completion/ DMA transfer completion	UARTC1/ DMAC	93	0650H	00000650H

Table 7-1 Interrupt/exception source list (4/4)

Type	Classification	Interrupt/Exception Source				Default Priority[a]	Exception Code	Handler Address
		Name	Control Register	Generating Source	Gener. Unit			
Maskable	Interrupt	INTUC1T	PIC94	UARTC1 transmission enable/ DMA transfer completion	UARTC1/ DMAC	94	0660H	00000660H
	Interrupt	INTAD0	PIC95	ADC0 conversion completion/ DMA transfer completion	ADC0/ DMAC	95	0670H	00000670H
	Interrupt	INTAD1	PIC96	ADC1 conversion completion/ DMA transfer completion	ADC1/ DMAC	96	0680H	00000680H
	Interrupt	INTCC10	PIC97[d]	CC10 capture input/ compare match	TMENC1[d]	97	0690H	00000690H
	Interrupt	INTCC11	PIC98[d]	CC11capture input/ compare match	TMENC1[d]	98	06A0H	000006A0H
	Interrupt	INTCM10	PIC99[d]	CM10 compare match	TMENC1[d]	99	06B0H	000006B0H
	Interrupt	INTCM11	PIC100[d]	CM10 compare match	TMENC1[d]	100	06C0H	000006C0H
	Interrupt	INTOVF	PIC101[d]	TMENC1 overflow	TMENC1[d]	101	06D0H	000006D0H
	Interrupt	INTUDF	PIC102[d]	TMENC1 underflow	TMENC1[d]	102	06E0H	000006E0H
	Interrupt	INTDMA2	PIC103	DMA channel 2 transfer completion	DMAC	103	06F0H	000006F0H
	Interrupt	INTDMA3	PIC104	DMA channel 3 transfer completion	DMAC	104	0700H	00000700H
	Interrupt	INTPERR	PIC105	Internal RAM parity error	iRAM	105	0710H	00000710H

a) Default Priority: The priority order that takes precedence when two or more maskable interrupt requests at the same software priority level are present at the same time. The highest priority is 0.

b) n = 0 to FH

c) The execution address of the illegal instruction when an illegal opcode exception occurs is calculated by (Restored PC - 4).

d) not available on µPD70F3447

7.2 Non-maskable Interrupt

A non-maskable interrupt request is acknowledged unconditionally, even when interrupts are in the interrupt disabled (DI) status. A NMI is not subject to priority control and takes precedence over all the other interrupts.

A non-maskable interrupt request is input from the NMI pin. When the valid edge specified by ESN0, ESN1 bits of the interrupt mode register 0 (INTM0) is detected at the NMI pin, the interrupt occurs.

While the service program of the non-maskable interrupt is being executed (PSW.NP = 1), the
acknowledgment of another non-maskable interrupt request is held pending. The pending NMI is acknowledged after the original service program of the non-maskable interrupt under execution has been terminated (by the RETI instruction). Note that if two or more NMI requests are input during the execution of the service program for a NMI, the number of NMIs that will be acknowledged after PSW.NP is cleared to 0 is only one.

Remark PSW.NP: The NP bit of the PSW register.

7.2.1 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 half-word (FECC) of ECR.

(4) Sets the NP and ID bits of the PSW and clears the EP bit.

(5) Sets the handler address (00000010H) corresponding to the non-maskable interrupt to the PC, and transfers control.

The processing configuration of a non-maskable interrupt is shown in *Figure 7-1*.

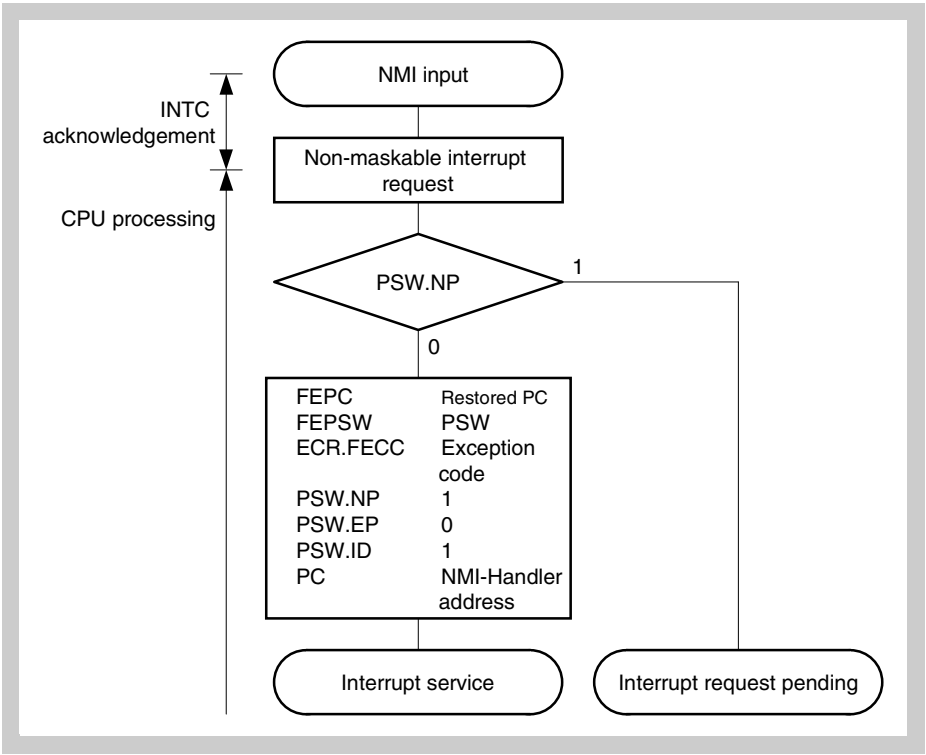


Figure 7-1 Processing configuration of non-maskable interrupt

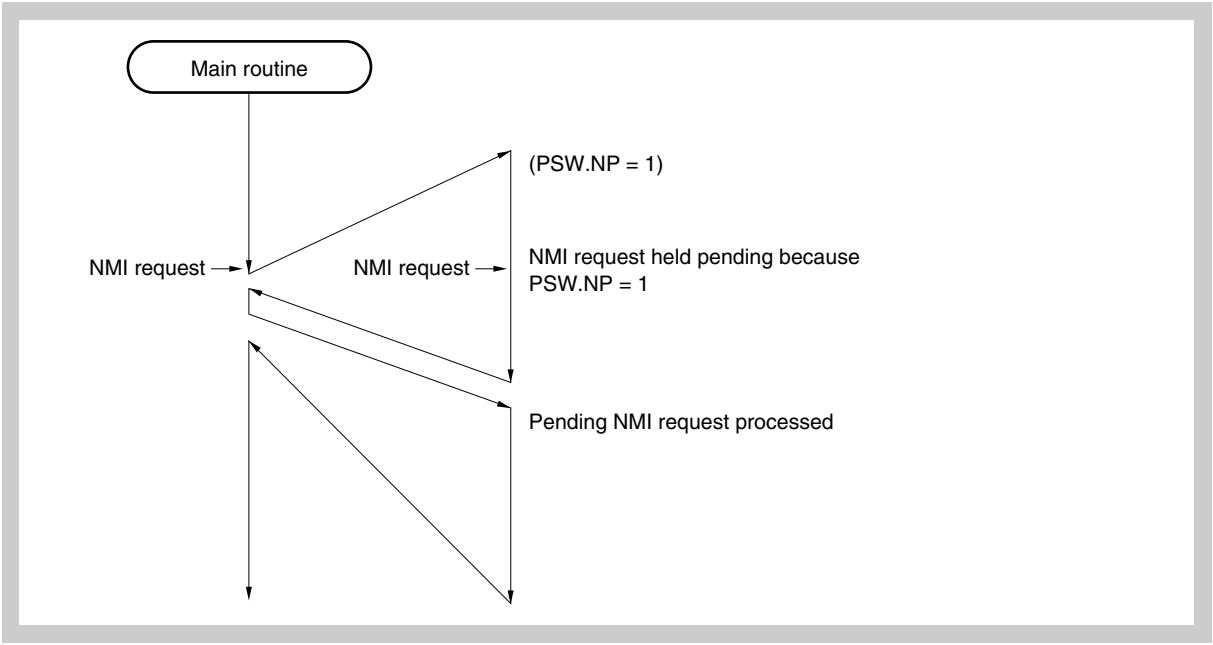


Figure 7-2 Acknowledging non-maskable interrupt request

(a) If a new NMI request is generated while a NMI service program is being executed

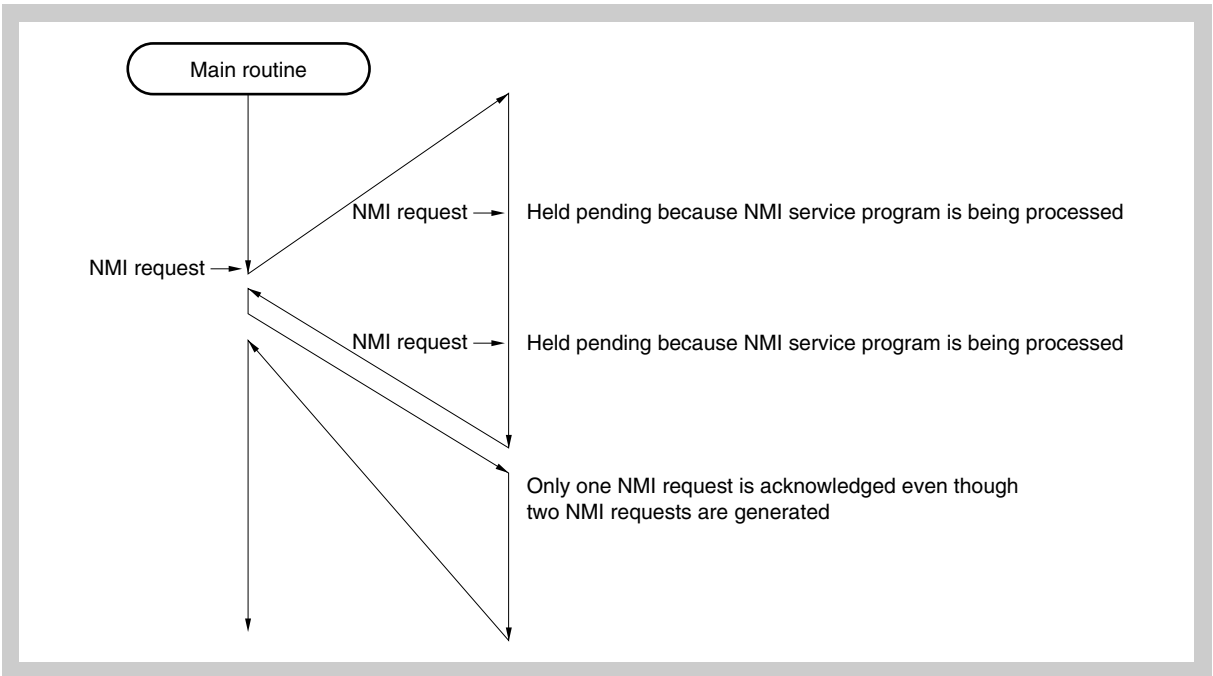


Figure 7-2 Acknowledging non-maskable interruptrequest

(b) If a new NMI request is generated twice while a NMI service program is being executed

7.2.2 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 7-3 illustrates how the RETI instruction is processed.

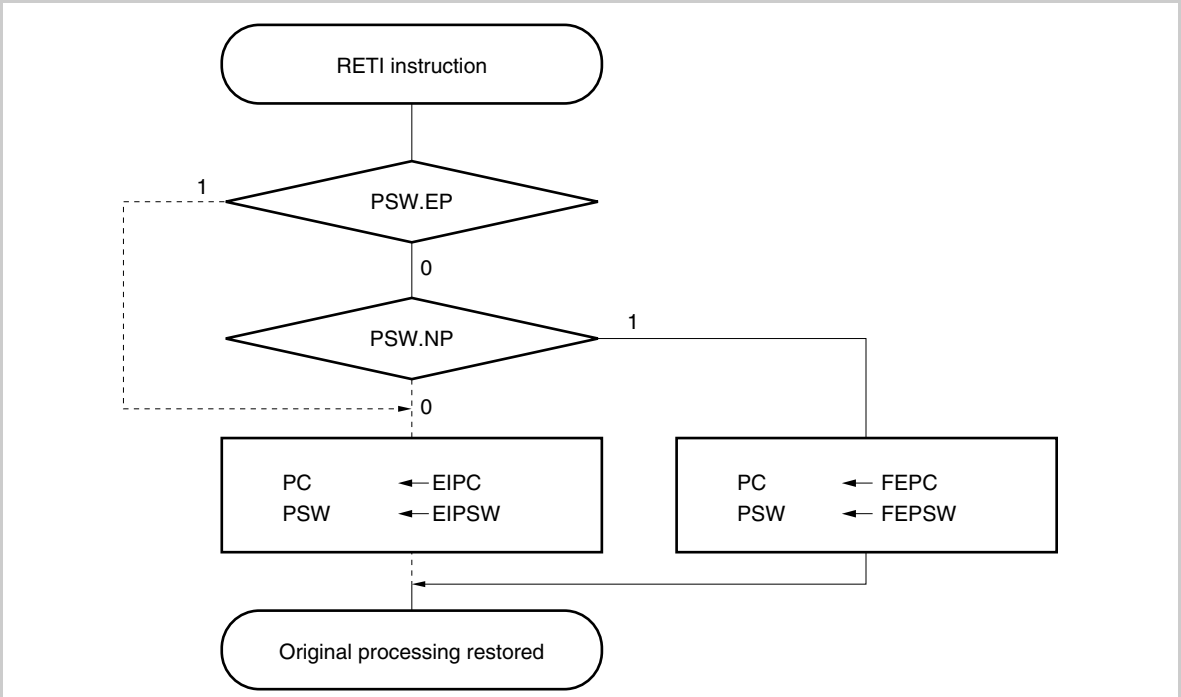


Figure 7-3 RETI instruction processing

Remark The solid line indicates the CPU processing flow.

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.

7.2.3 Non-maskable interrupt status flag (NP)

The NP flag is a status flag that indicates that non-maskable interrupt (NMI) processing is under execution.

This flag is set when a NMI interrupt has been acknowledged, and masks all interrupt requests and exceptions to prohibit multiple interrupts from being acknowledged.

Initial Value 00000020_H

	31	8	7	6	5	4	3	2	1	0
PSW			**NP**	EP	ID	SAT	CY	OV	S	Z

Table 7-2 Non-maskable interrupt status flag (NP)

Bit position	Bit name	Function
7	NP	NMI Servicing Status 0: No NMI interrupt servicing 1: NMI interrupt currently servicing

7.2.4 Edge Detection Function

The valid edge of the external NMI pin input can be specified by the ESN[1:0] bits of the interrupt mode register 0 (INTM0).

For further information refer to *“Interrupt mode register 0 (INTM0)” on page 196*.

7.3 Maskable Interrupts

Maskable interrupt requests can be masked by interrupt control registers. The μPD70F3187 has 106 maskable interrupt sources and the μPD70F3447 has 91 maskable interrupt sources.

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).

7.3.1 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 half-word 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 7-4*.

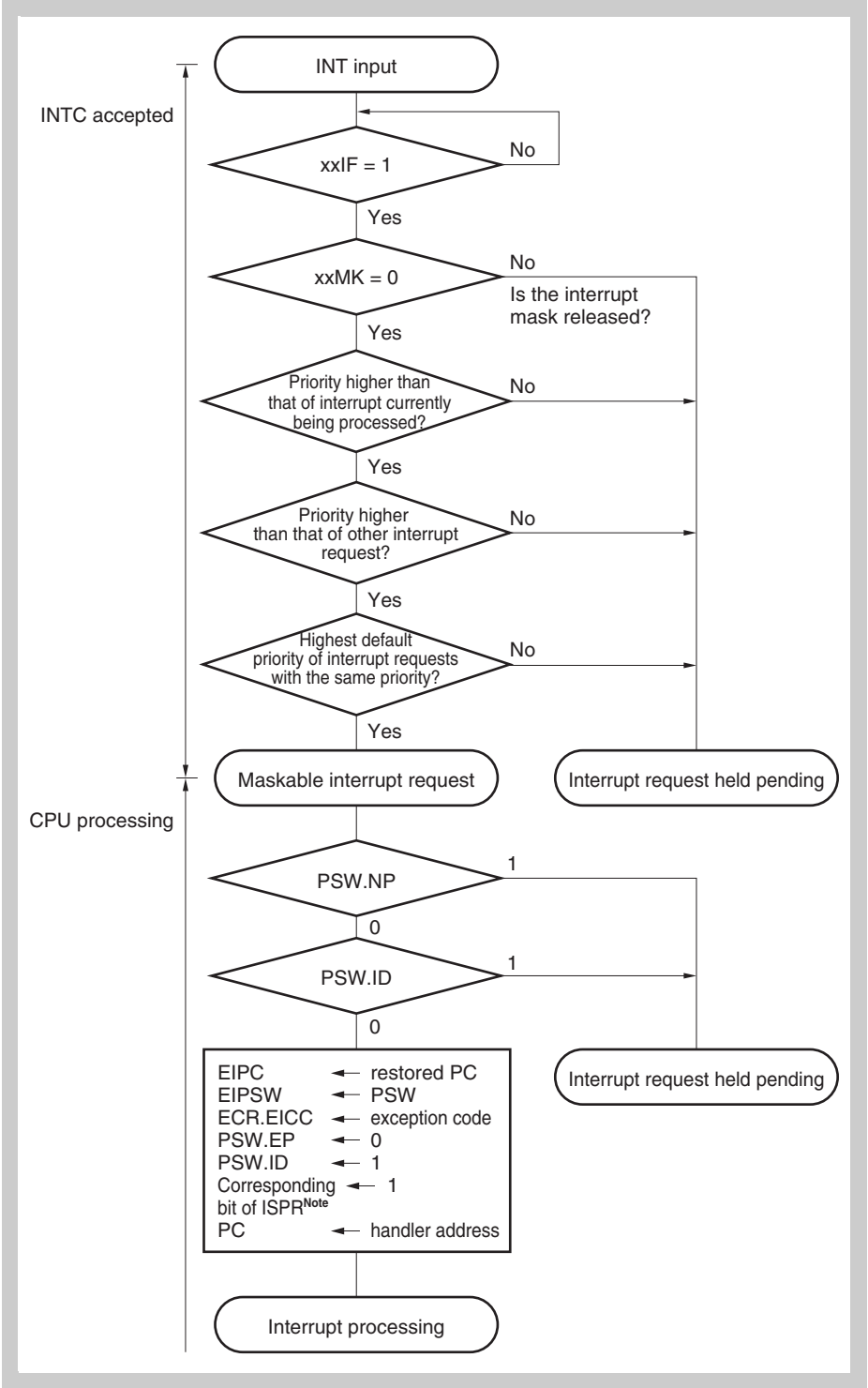


Figure 7-4 Maskable Interrupt Processing

Note For details refer to *"In-service priority register (ISPR)" on page 194*.

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.

7.3.2 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 7-5 illustrates the processing of the RETI instruction.

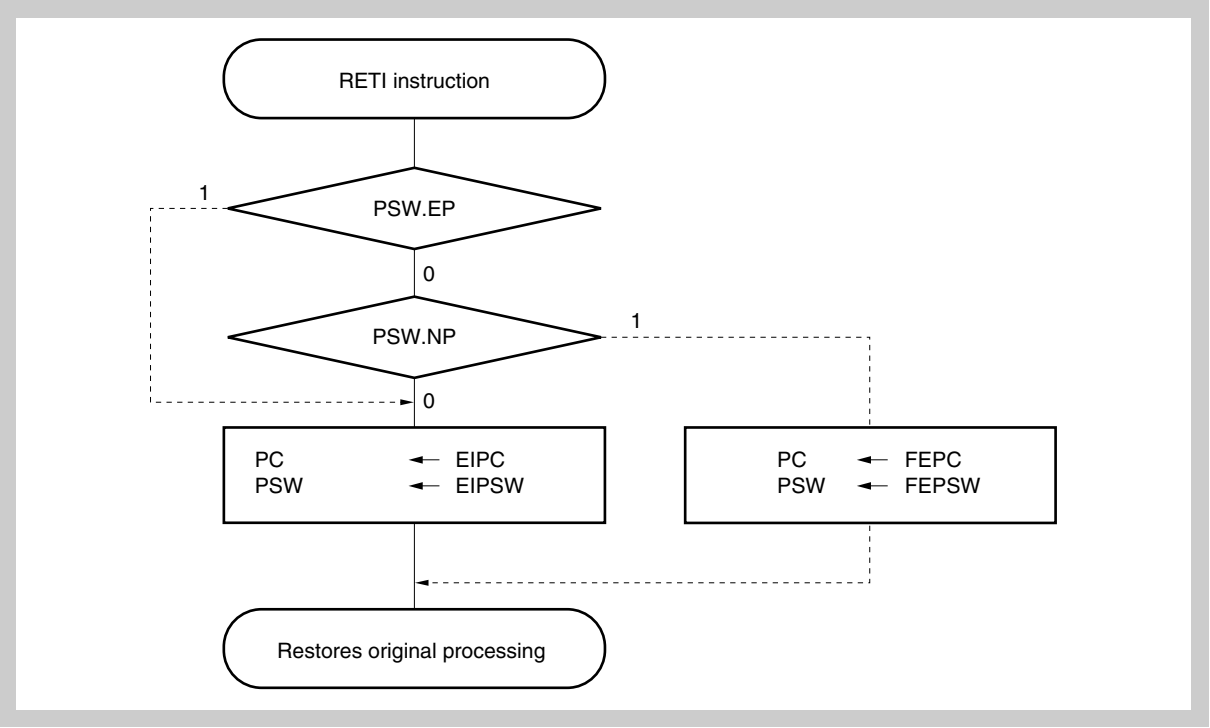


Figure 7-5 RETI Instruction Processing

Remark 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.

7.3.3 Priorities of maskable interrupts

The V850E/PH2 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 (PRn) of the interrupt control register (PICn). When two or more interrupts having the same priority level specified by the PRn 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 *Table 7-1, "Interrupt/exception source list," on page 172*. The programmable priority control customizes interrupt requests into eight levels by setting the priority level specification flag.

Note 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.

Remark n = 0 to 105 (number of interrupt)

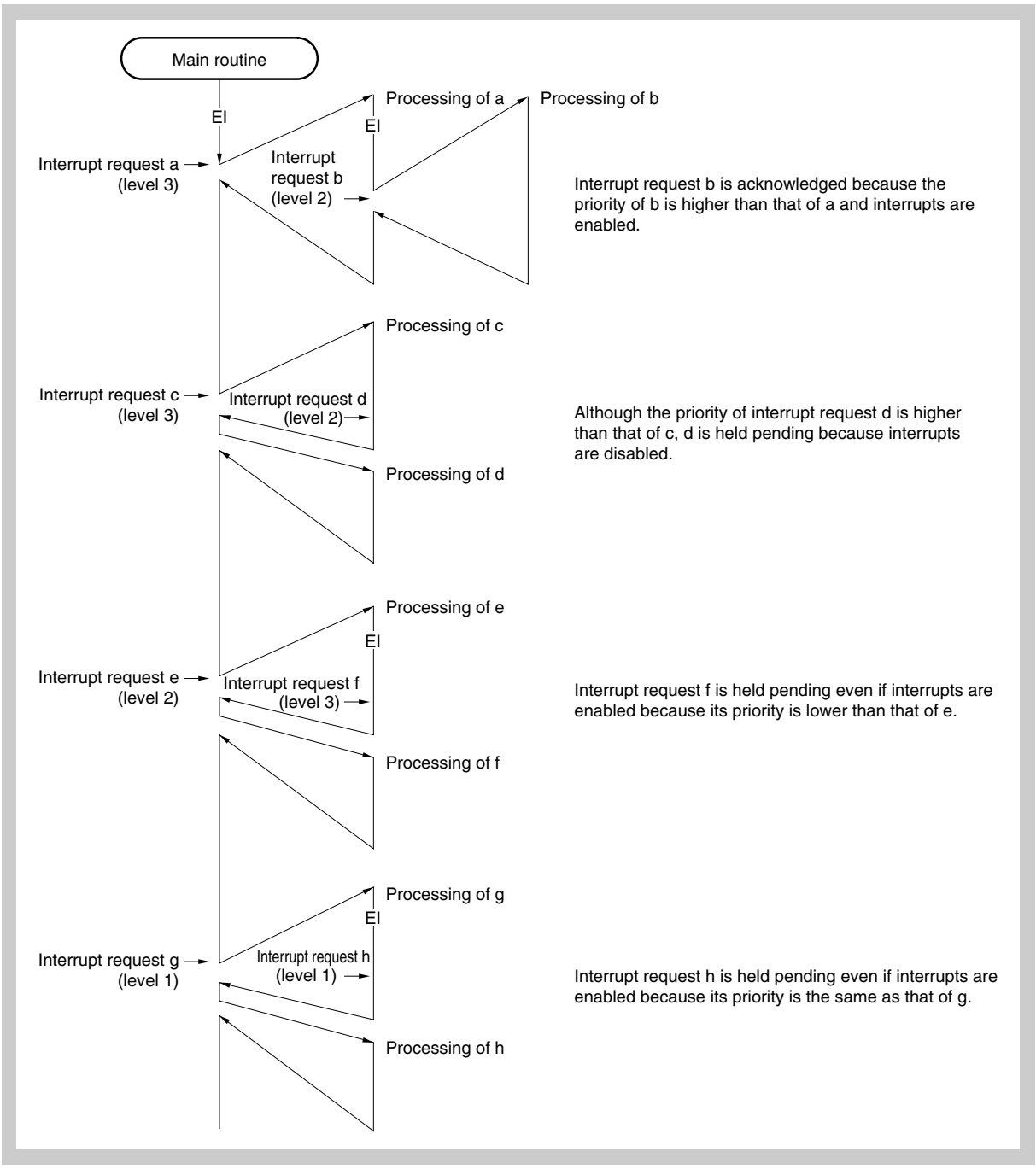


Figure 7-6 Example of processing in which another interrupt request is issued while an interrupt is being processed (1/2)

Note
1. **a** to **u** in the figure 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.

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.

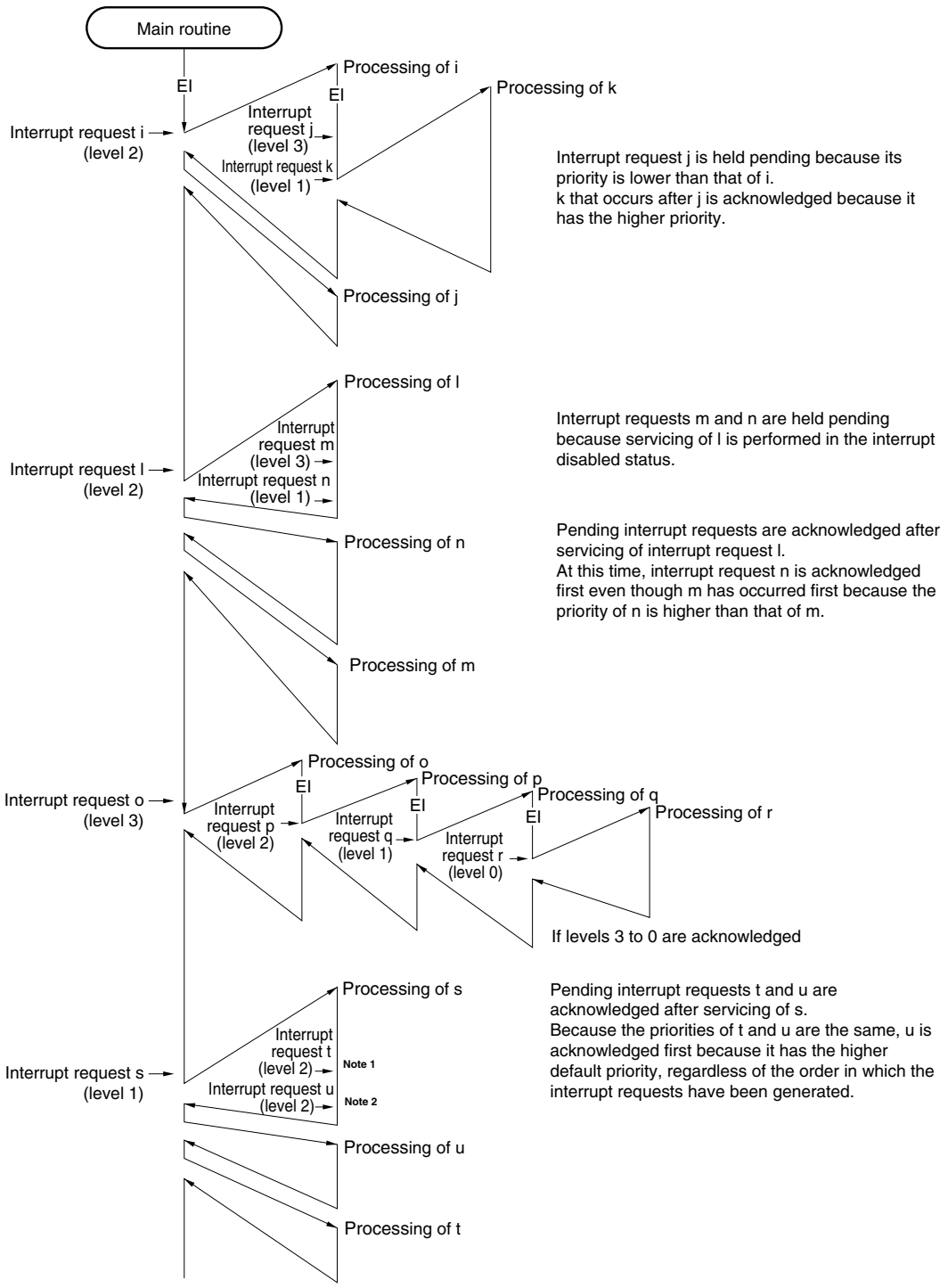


Figure 7-6 Example of processing in which another interrupt request is issued while an interrupt is being processed (2/2)

Note 1. Lower default priority

2. Higher default priority

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.

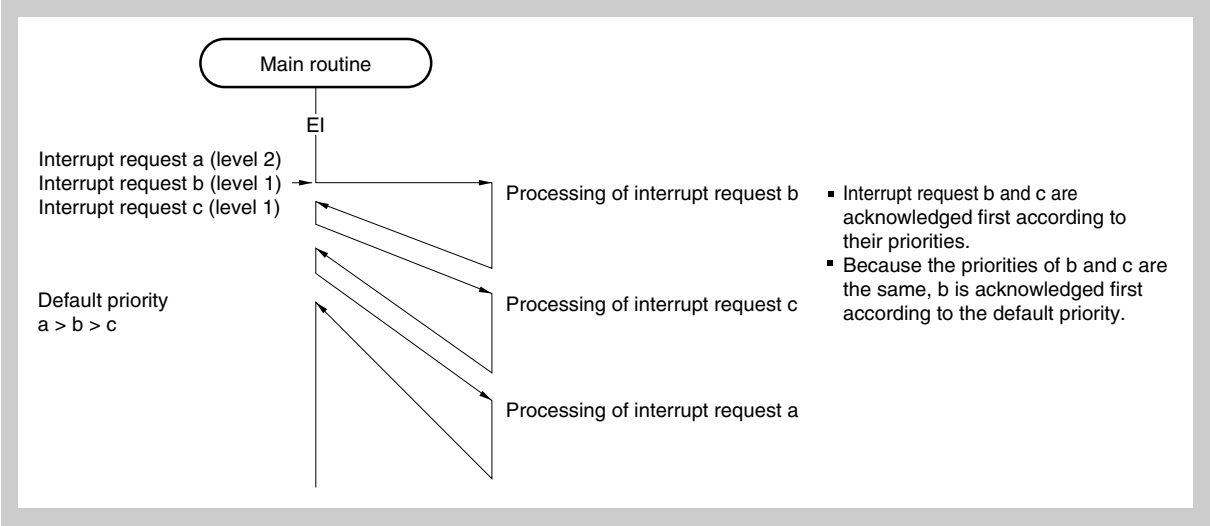


Figure 7-7 Example of processing interrupt requests simultaneously generated

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.

7.3.4 Interrupt control register (PICn)

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

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

Address PICn: n = 0 to 105, refer to *Table 7-4 on page 189*

Initial Value 47_H. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
PICn	IFn	MKn	0	0	0	PRn2	PRn1	PRn0
	R/W	R/W	R	R	R	R/W	R/W	R/W

Table 7-3 Interrupt control register (PICn) contents

Bit position	Bit name	Function
7	IFn	Interrupt Request Flag n 0: Interrupt request is not issued. 1: Interrupt request issued. **Note:** The IFn flag is automatically reset by hardware when an interrupt request is acknowledged.
6	MKn	Interrupt Mask Flag n 0: Interrupt servicing enabled. 1: Interrupt servicing disabled. (IFn flag hold pending.)
2 to 0	PRn[2:0]	Specifies the priority of the corresponding maskable interrupt.

PRn2	PRn1	PRn0	Interrupt Priority Specification n
0	0	0	Priority level 0 (highest priority)
0	0	1	Priority level 1
0	1	0	Priority level 2
0	1	1	Priority level 3
1	0	0	Priority level 4
1	0	1	Priority level 5
1	1	0	Priority level 6
1	1	1	Priority level 7 (lowest priority)

Table 7-4 Adresses and bits of interrupt control registers

Address	Register	Bit								Associated Interrupt
		7	6	5	4	3	2	1	0	
FFFFF110$_H$	PIC0	IF0	MK0	0	0	0	PR02	PR01	PR00	INTP0
FFFFF112$_H$	PIC1	IF1	MK1	0	0	0	PR12	PR11	PR10	INTP1
FFFFF114$_H$	PIC2	IF2	MK2	0	0	0	PR22	PR21	PR20	INTP2
FFFFF116$_H$	PIC3	IF3	MK3	0	0	0	PR32	PR31	PR30	INTP3
FFFFF118$_H$	PIC4	IF4	MK4	0	0	0	PR42	PR41	PR40	INTP4
FFFFF11A$_H$	PIC5	IF5	MK5	0	0	0	PR52	PR51	PR50	INTP5
FFFFF11C$_H$	PIC6	IF6	MK6	0	0	0	PR62	PR61	PR60	INTP6
FFFFF11E$_H$	PIC7	IF7	MK7	0	0	0	PR72	PR71	PR70	INTP7
FFFFF120$_H$	PIC8	IF8	MK8	0	0	0	PR82	PR81	PR80	INTP8
FFFFF122$_H$	PIC9	IF9	MK9	0	0	0	PR92	PR91	PR90	INTP9
FFFFF124$_H$	PIC10	IF10	MK10	0	0	0	PR102	PR101	PR100	INTP10
FFFFF126$_H$	PIC11	IF11	MK11	0	0	0	PR112	PR111	PR110	INTP11
FFFFF128$_H$	PIC12	IF12	MK12	0	0	0	PR122	PR121	PR120	INTP12
FFFFF12A$_H$	PIC13	IF13	MK13	0	0	0	PR132	PR131	PR130	INTTR0OV
FFFFF12C$_H$	PIC14	IF14	MK14	0	0	0	PR142	PR141	PR140	INTTR0CC0
FFFFF12E$_H$	PIC15	IF15	MK15	0	0	0	PR152	PR151	PR150	INTTR0CC1
FFFFF130$_H$	PIC16	IF16	MK16	0	0	0	PR162	PR161	PR160	INTTR0CC2
FFFFF132$_H$	PIC17	IF17	MK17	0	0	0	PR172	PR171	PR170	INTTR0CC3
FFFFF134$_H$	PIC18	IF18	MK18	0	0	0	PR182	PR181	PR180	INTTR0CC4
FFFFF136$_H$	PIC19	IF19	MK19	0	0	0	PR192	PR191	PR190	INTTR0CC5
FFFFF138$_H$	PIC20	IF20	MK20	0	0	0	PR202	PR201	PR200	INTTR0CD
FFFFF13A$_H$	PIC21	IF21	MK21	0	0	0	PR212	PR211	PR210	INTTR0OD
FFFFF13C$_H$	PIC22	IF22	MK22	0	0	0	PR222	PR221	PR220	INTTR0ER
FFFFF13E$_H$	PIC23	IF23	MK23	0	0	0	PR232	PR231	PR230	INTTR1OV
FFFFF140$_H$	PIC24	IF24	MK24	0	0	0	PR242	PR241	PR240	INTTR1CC0
FFFFF142$_H$	PIC25	IF25	MK25	0	0	0	PR252	PR251	PR250	INTTR1CC1
FFFFF144$_H$	PIC26	IF26	MK26	0	0	0	PR262	PR261	PR260	INTTR1CC2
FFFFF146$_H$	PIC27	IF27	MK27	0	0	0	PR272	PR271	PR270	INTTR1CC3
FFFFF148$_H$	PIC28	IF28	MK28	0	0	0	PR282	PR281	PR280	INTTR1CC4
FFFFF14A$_H$	PIC29	IF29	MK29	0	0	0	PR292	PR291	PR290	INTTR1CC5
FFFFF14C$_H$	PIC30	IF30	MK30	0	0	0	PR302	PR301	PR300	INTTR1CD
FFFFF14E$_H$	PIC31	IF31	MK31	0	0	0	PR312	PR311	PR310	INTTR1OD
FFFFF150$_H$	PIC32	IF32	MK32	0	0	0	PR322	PR321	PR320	INTTR1ER
FFFFF152$_H$	PIC33	IF33	MK33	0	0	0	PR332	PR331	PR330	INTT0OV
FFFFF154$_H$	PIC34	IF34	MK34	0	0	0	PR342	PR341	PR340	INTT0CC0
FFFFF156$_H$	PIC35	IF35	MK35	0	0	0	PR352	PR351	PR350	INTT0CC1
FFFFF158$_H$	PIC36	IF36	MK36	0	0	0	PR362	PR361	PR360	INTT0EC
FFFFF15A$_H$	PIC37	IF37	MK37	0	0	0	PR372	PR371	PR370	INTT1OV
FFFFF15C$_H$	PIC38	IF38	MK38	0	0	0	PR382	PR381	PR380	INTT1CC0
FFFFF15E$_H$	PIC39	IF39	MK39	0	0	0	PR392	PR391	PR390	INTT1CC1

Table 7-4 Adresses and bits of interrupt control registers

Address	Register	Bit								Associated Interrupt
		7	6	5	4	3	2	1	0	
FFFFF160$_H$	PIC40	IF40	MK40	0	0	0	PR402	PR401	PR400	INTT1EC
FFFFF162$_H$	PIC41	IF41	MK41	0	0	0	PR412	PR411	PR410	INTP0OV
FFFFF164$_H$	PIC42	IF42	MK42	0	0	0	PR422	PR421	PR420	INTP0CC0
FFFFF166$_H$	PIC43	IF43	MK43	0	0	0	PR432	PR431	PR430	INTP0CC1
FFFFF168$_H$	PIC44	IF44	MK44	0	0	0	PR442	PR441	PR440	INTP1OV
FFFFF16A$_H$	PIC45	IF45	MK45	0	0	0	PR452	PR451	PR450	INTP1CC0
FFFFF16C$_H$	PIC46	IF46	MK46	0	0	0	PR462	PR461	PR460	INTP1CC1
FFFFF16E$_H$	PIC47	IF47	MK47	0	0	0	PR472	PR471	PR470	INTP2OV
FFFFF170$_H$	PIC48	IF48	MK48	0	0	0	PR482	PR481	PR480	INTP2CC0
FFFFF172$_H$	PIC49	IF49	MK49	0	0	0	PR492	PR491	PR490	INTP2CC1
FFFFF174$_H$	PIC50	IF50	MK50	0	0	0	PR502	PR501	PR500	INTP3OV
FFFFF176$_H$	PIC51	IF51	MK51	0	0	0	PR512	PR511	PR510	INTP3CC0
FFFFF178$_H$	PIC52	IF52	MK52	0	0	0	PR522	PR521	PR520	INTP3CC1
FFFFF17A$_H$	PIC53	IF53	MK53	0	0	0	PR532	PR531	PR530	INTP4OV
FFFFF17C$_H$	PIC54	IF54	MK54	0	0	0	PR542	PR541	PR540	INTP4CC0
FFFFF17E$_H$	PIC55	IF55	MK55	0	0	0	PR552	PR551	PR550	INTP4CC1
FFFFF180$_H$	PIC56	IF56	MK56	0	0	0	PR562	PR561	PR560	INTP5OV
FFFFF182$_H$	PIC57	IF57	MK57	0	0	0	PR572	PR571	PR570	INTP5CC0
FFFFF184$_H$	PIC58	IF58	MK58	0	0	0	PR582	PR581	PR580	INTP5CC1
FFFFF186$_H$	PIC59	IF59	MK59	0	0	0	PR592	PR591	PR590	INTP6OV
FFFFF188$_H$	PIC60	IF60	MK60	0	0	0	PR602	PR601	PR600	INTP6CC0
FFFFF18A$_H$	PIC61	IF61	MK61	0	0	0	PR612	PR611	PR610	INTP6CC1
FFFFF18C$_H$	PIC62	IF62	MK62	0	0	0	PR622	PR621	PR620	INTP7OV
FFFFF18E$_H$	PIC63	IF63	MK63	0	0	0	PR632	PR631	PR630	INTP7CC0
FFFFF190$_H$	PIC64	IF64	MK64	0	0	0	PR642	PR641	PR640	INTP7CC1
FFFFF192$_H$	PIC65	IF65	MK65	0	0	0	PR652	PR651	PR650	INTP8OV
FFFFF194$_H$	PIC66	IF66	MK66	0	0	0	PR662	PR661	PR660	INTP8CC0
FFFFF196$_H$	PIC67	IF67	MK67	0	0	0	PR672	PR671	PR670	INTP8CC1
FFFFF198$_H$	PIC68	IF68	MK68	0	0	0	PR682	PR681	PR680	INTBRG0
FFFFF19A$_H$	PIC69	IF69	MK69	0	0	0	PR692	PR691	PR690	INTBRG1
FFFFF19C$_H$	PIC70	IF70	MK70	0	0	0	PR702	PR701	PR700	INTBRG2
FFFFF19E$_H$	PIC71	IF71	MK71	0	0	0	PR712	PR711	PR710	INTC0ERR
FFFFF1A0$_H$	PIC72	IF72	MK72	0	0	0	PR722	PR721	PR720	INTC0WUP
FFFFF1A2$_H$	PIC73	IF73	MK73	0	0	0	PR732	PR731	PR730	INTC0REC
FFFFF1A4$_H$	PIC74	IF74	MK74	0	0	0	PR742	PR741	PR740	INTC0TRX
FFFFF1A6$_H$	PIC75[a]	IF75	MK75	0	0	0	PR752	PR751	PR750	INTC1ERR[a]
FFFFF1A8$_H$	PIC76[a]	IF76	MK76	0	0	0	PR762	PR761	PR760	INTC1WUP[a]
FFFFF1AA$_H$	PIC77[a]	IF77	MK77	0	0	0	PR772	PR771	PR770	INTC1REC[a]
FFFFF1AC$_H$	PIC78[a]	IF78	MK78	0	0	0	PR782	PR781	PR780	INTC1TRX[a]
FFFFF1AE$_H$	PIC79	IF79	MK79	0	0	0	PR792	PR791	PR790	INTCB0T

Table 7-4 Adresses and bits of interrupt control registers

Address	Register	Bit								Associated Interrupt
		7	6	5	4	3	2	1	0	
FFFFF1B0$_H$	PIC80	IF80	MK80	0	0	0	PR802	PR801	PR800	INTCB0R
FFFFF1B2$_H$	PIC81	IF81	MK81	0	0	0	PR812	PR811	PR810	INTCB0RE
FFFFF1B4$_H$	PIC82[a]	IF82	MK82	0	0	0	PR822	PR821	PR820	INTCB1T[a]
FFFFF1B6$_H$	PIC83[a]	IF83	MK83	0	0	0	PR832	PR831	PR830	INTCB1R[a]
FFFFF1B8$_H$	PIC84[a]	IF84	MK84	0	0	0	PR842	PR841	PR840	INTCB1RE[a]
FFFFF1BA$_H$	PIC85	IF85	MK85	0	0	0	PR852	PR851	PR850	INTC30OVF
FFFFF1BC$_H$	PIC86	IF86	MK86	0	0	0	PR862	PR861	PR860	INTC30
FFFFF1BE$_H$	PIC87[a]	IF87	MK87	0	0	0	PR872	PR871	PR870	INTC31OVF[a]
FFFFF1C0$_H$	PIC88[a]	IF88	MK88	0	0	0	PR882	PR881	PR880	INTC31[a]
FFFFF1C2$_H$	PIC89	IF89	MK89	0	0	0	PR892	PR891	PR890	INTUC0RE
FFFFF1C4$_H$	PIC90	IF90	MK90	0	0	0	PR902	PR901	PR900	INTUC0R
FFFFF1C6$_H$	PIC91	IF91	MK91	0	0	0	PR912	PR911	PR910	INTUC0T
FFFFF1C8$_H$	PIC92	IF92	MK92	0	0	0	PR922	PR921	PR920	INTUC1RE
FFFFF1CA$_H$	PIC93	IF93	MK93	0	0	0	PR932	PR931	PR930	INTUC1R
FFFFF1CC$_H$	PIC94	IF94	MK94	0	0	0	PR942	PR941	PR940	INTUC1T
FFFFF1CE$_H$	PIC95	IF95	MK95	0	0	0	PR952	PR951	PR950	INTAD0
FFFFF1D0$_H$	PIC96	IF96	MK96	0	0	0	PR962	PR961	PR960	INTAD1
FFFFF1D2$_H$	PIC97[a]	IF97	MK97	0	0	0	PR972	PR971	PR970	INTCC10[a]
FFFFF1D4$_H$	PIC98[a]	IF98	MK98	0	0	0	PR982	PR981	PR980	INTCC11[a]
FFFFF1D6$_H$	PIC99[a]	IF99	MK99	0	0	0	PR992	PR991	PR990	INTCM10[a]
FFFFF1D8$_H$	PIC100[a]	IF100	MK100	0	0	0	PR1002	PR1001	PR1000	INTCM11[a]
FFFFF1DA$_H$	PIC101[a]	IF101	MK101	0	0	0	PR1012	PR1011	PR1010	INTOVF[a]
FFFFF1DC$_H$	PIC102[a]	IF102	MK102	0	0	0	PR1022	PR1021	PR1020	INTUDF[a]
FFFFF1DE$_H$	PIC103	IF103	MK103	0	0	0	PR1032	PR1031	PR1030	INTDMA2
FFFFF1E0$_H$	PIC104	IF104	MK104	0	0	0	PR1042	PR1041	PR1040	INTDMA3
FFFFF1E2$_H$	PIC105	IF105	MK105	0	0	0	PR1052	PR1051	PR1050	INTPERR

a) Not available on µPD70F3447.

7.3.5 Interrupt mask registers 0 to 6 (IMR0 to IMR6)

The IMR0 to IMR6 registers set the interrupt mask state for the maskable interrupts. The IMK0 to IMK104 bits are equivalent to the MKn bit in the corresponding PICn register.

Access These registers can be read/written in 16-bit units.
If the higher 8 bits of the IMRm register are used as the IMRmH register and the lower 8 bits as the IMRmL register, these registers can be read or written in 8-bit or 1-bit units.

Address

IMR0, IMR0L:	$FFFFF100_H$	IMR0H:	$FFFFF101_H$
IMR1, IMR1L:	$FFFFF102_H$	IMR1H:	$FFFFF103_H$
IMR2, IMR2L:	$FFFFF104_H$	IMR2H:	$FFFFF105_H$
IMR3, IMR3L:	$FFFFF106_H$	IMR3H:	$FFFFF107_H$
IMR4, IMR4L:	$FFFFF108_H$	IMR4H:	$FFFFF109_H$
IMR5, IMR5L:	$FFFFF10A_H$	IMR5H:	$FFFFF10B_H$
IMR6, IMR6L:	$FFFFF10C_H$	IMR6H:	$FFFFF10D_H$

Initial Value $FFFF_H$. These registers are cleared by any reset.

Caution

1. Bits 15 to 9 of the IMR6 register (bits 7 to 1 of the IMR6H register) are fixed to 1. If these bits are not 1, the operation cannot be guaranteed.
2. The device file defines the MKn bits as a reserved word. If a bit is manipulated using the name of MKn, the contents of the PICn register, instead of the IMRm register, are rewritten (as a result, the contents of the IMRm register are also rewritten).

IMR0

	15	14	13	12	11	10	9	8
IMR0H	MK15	MK14	MK13	MK12	MK11	MK10	MK9	MK8
	R/W	R/W	R	R	R	R/W	R/W	R/W
	7	6	5	4	3	2	1	0
IMR0L	MK7	MK6	MK5	MK4	MK3	MK2	MK1	MK0
	R/W	R/W	R	R	R	R/W	R/W	R/W

IMR1

	15	14	13	12	11	10	9	8
IMR1H	MK31	MK30	MK29	MK28	MK27	MK26	MK25	MK24
	R/W	R/W	R	R	R	R/W	R/W	R/W
	7	6	5	4	3	2	1	0
IMR1L	MK23	MK22	MK21	MK20	MK19	MK18	MK17	MK16
	R/W	R/W	R	R	R	R/W	R/W	R/W

IMR2, IMR2H

	15	14	13	12	11	10	9	8
IMR2, IMR2H	MK47	MK46	MK45	MK44	MK43	MK42	MK41	MK40
	R/W	R/W	R	R	R	R/W	R/W	R/W
	7	6	5	4	3	2	1	0
IMR2L	MK39	MK38	MK37	MK36	MK35	MK34	MK33	MK32
	R/W	R/W	R	R	R	R/W	R/W	R/W

	15	14	13	12	11	10	9	8
IMR3, IMR3H	MK63	MK62	MK61	MK60	MK59	MK58	MK57	MK56
	R/W	R/W	R	R	R	R/W	R/W	R/W
	7	6	5	4	3	2	1	0
IMR3L	MK55	MK54	MK53	MK52	MK51	MK50	MK49	MK48
	R/W	R/W	R	R	R	R/W	R/W	R/W

	15	14	13	12	11	10	9	8
IMR4, IMR4H	MK79	MK78Note	MK77Note	MK76Note	MK75Note	MK74	MK73	MK72
	R/W	R/W	R	R	R	R/W	R/W	R/W
	7	6	5	4	3	2	1	0
IMR4L	MK71	MK70	MK69	MK68	MK67	MK66	MK65	MK64
	R/W	R/W	R	R	R	R/W	R/W	R/W

	15	14	13	12	11	10	9	8
IMR5, IMR5H	MK95	MK94	MK93	MK92	MK91	MK90	MK89	MK88Note
	R/W	R/W	R	R	R	R/W	R/W	R/W
	7	6	5	4	3	2	1	0
IMR5L	MK87Note	MK86	MK85	MK84Note	MK83Note	MK82Note	MK81	MK80
	R/W	R/W	R	R	R	R/W	R/W	R/W

	15	14	13	12	11	10	9	8
IMR6, IMR6H	1	1	1	1	1	1	MK105	MK104
	R/W	R/W	R	R	R	R/W	R/W	R/W
	7	6	5	4	3	2	1	0
IMR6L	MK39	MK38	MK37	MK36	MK35	MK34	MK103	MK102Note
	R/W	R/W	R	R	R	R/W	R/W	R/W

Note Since these interrupt entries are not specified for μPD70F3447, the reset value (1) of these mask bits should be kept for μPD70F3447 at any time.

Table 7-5 Interrupt mask register (IMR0 to IMR6) contents

Bit position	Bit name	Function
x	MKn[a]	Interrupt Mask Flag n 0: Interrupt servicing enabled. 1: Interrupt servicing disabled (pending).

a) n = 0 to 105, according to the default priority of maskable interrupts, refer to *Table 7-1*

7.3.6 In-service priority register (ISPR)

The ISPR 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.

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

Address $FFFFF1FA_H$

Initial Value 007_H. This register is cleared by any reset.

Caution In the interrupt enabled (EI) state, if an interrupt is acknowledged during the reading of the ISPR register, the value of the ISPR register may be read after the bit is set (1) by this interrupt acknowledgment. To read the value of the ISPR register properly before interrupt acknowledgment, read it in the interrupt disabled (DI) state.

ISPR

7	6	5	4	3	2	1	0
ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0
R	R	R	R	R	R	R	R

Table 7-6 ISPR register contents

Bit position	Bit name	Function
0 to 7	ISPR[7:0]	Priority n (n = 0 to 7) of interrupt currently being acknowledged 0: Interrupt request with priority n is not acknowledged. 1: Interrupt request with priority n is being acknowledged.

7.3.7 Maskable interrupt status flag (ID)

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

Initial Value 00000020_H

	31	8	7	6	5	4	3	2	1	0
PSW			NP	EP	**ID**	SAT	CY	OV	S	Z

Table 7-7 Maskable interrupt status flag (ID)

Bit position	Bit name	Function
5	ID	Disables/enables maskable interrupt servicing. 0: Maskable interrupt request acknowledgment enabled. 1: Maskable interrupt request acknowledgment disabled (pending). **Note:** Interrupt disable flag (ID) function • This flag 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 the PSW. • Non-maskable interrupt and exceptions are acknowledged regardless of this flag. When a maskable interrupt is acknowledged, the ID flag is automatically set to 1 by hardware. • The interrupt request generated during the acknowledgement disabled period (ID = 1) can be acknowledged when the IFn bit of the interrupt control register PICn is set to 1, and the ID flag is reset to 0.

7.3.8 Interrupt trigger mode selection

The valid edge of the maskable external interrupt input pin (INTPn) can be selected by software (n = 0 to 12).

Following valid edges and combinations can be specified by interrupt mode registers 0 to 3 (INTM0 to INTM3:

- Rising edge
- Falling edge
- Both, the rising and falling edges

The edge-detected INTPn signal becomes an interrupt source.

(1) Interrupt mode register 0 (INTM0)

The behaviour of the external interrupt input pins INTP0 to INTP2, as well as NMI, can be specified by the interrupt mode register 0 (INTM0).

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

Address $FFFFF880_H$

Initial Value 00_H. This register is cleared by any reset.

Caution Changing the state of interrupt mode configuration bits ESn[0:1] may trigger an unintended interrupt event for the corresponding interrupt channel. Be sure to mask the corresponding interrupt channel and clear the interrupt status flag after changing the bits ESn[0:1] (n = 0 to 2).

INTM0

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

Table 7-8 INTM0 register contents (1/2)

Bit position	Bit name	Function
7, 6	ES21 ES20	Specifies the INTP2 pin input valid edge.

ES21	ES20	Valid Edge Specification of INTP2 pin input
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both, rising and falling edges

Table 7-8 INTM0 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>5, 4</td><td>ES11
ES10</td><td>Specifies the INTP1 pin input valid edge.
<table>
<tr><th>ES11</th><th>ES10</th><th>Valid Edge Specification of INTP1 pin input</th></tr>
<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>Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>Both, rising and falling edges</td></tr>
</table></td></tr>
<tr><td>3, 2</td><td>ES01
ES00</td><td>Specifies the INTP0 pin input valid edge.
<table>
<tr><th>ES01</th><th>ES00</th><th>Valid Edge Specification of INTP0 pin input</th></tr>
<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>Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>Both, rising and falling edges</td></tr>
</table></td></tr>
<tr><td>1, 0</td><td>ESN1
ESN0</td><td>Specifies the NMI pin input valid edge.
<table>
<tr><th>ESN1</th><th>ESN0</th><th>Valid Edge Specification of NMI pin input</th></tr>
<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>Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>Both, rising and falling edges</td></tr>
</table></td></tr>
</table>

(2) Interrupt mode register 1 (INTM1)

The behaviour of the external interrupt input pins INTP3 to INTP6 can be specified by the interrupt mode register 1 (INTM1).

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

Address $FFFFF882_H$

Initial Value 00_H. This register is cleared by any reset.

Caution Changing the state of interrupt mode configuration bits ESn[0:1] may trigger an unintended interrupt event for the corresponding interrupt channel. Be sure to mask the corresponding interrupt channel and clear the interrupt status flag after changing the bits ESn[0:1] (n = 3 to 6).

INTM1

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

Table 7-9 INTM1 register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7, 6</td><td>ES61
ES60</td><td>Specifies the INTP6 pin input valid edge.
<table>
<tr><th>ES61</th><th>ES60</th><th>Valid Edge Specification of INTP6 pin input</th></tr>
<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>Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>Both, rising and falling edges</td></tr>
</table></td></tr>
<tr><td>5, 4</td><td>ES51
ES50</td><td>Specifies the INTP5 pin input valid edge.
<table>
<tr><th>ES51</th><th>ES50</th><th>Valid Edge Specification of INTP5 pin input</th></tr>
<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>Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>Both, rising and falling edges</td></tr>
</table></td></tr>
<tr><td>3, 2</td><td>ES41
ES40</td><td>Specifies the INTP4 pin input valid edge.
<table>
<tr><th>ES41</th><th>ES40</th><th>Valid Edge Specification of INTP4 pin input</th></tr>
<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>Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>Both, rising and falling edges</td></tr>
</table></td></tr>
<tr><td>1, 0</td><td>ES31
ES30</td><td>Specifies the INTP3 pin input valid edge.
<table>
<tr><th>ES31</th><th>ES30</th><th>Valid Edge Specification of INTP3 pin input</th></tr>
<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>Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>Both, rising and falling edges</td></tr>
</table></td></tr>
</table>

(3) Interrupt mode register 2 (INTM2)

The behaviour of the external interrupt input pins INTP7 to INTP10 can be specified by the interrupt mode register 2 (INTM2).

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

Address $FFFFF884_H$

Initial Value 00_H. This register is cleared by any reset.

Caution Changing the state of interrupt mode configuration bits ESn[0:1] may trigger an unintended interrupt event for the corresponding interrupt channel. Be sure to mask the corresponding interrupt channel and clear the interrupt status flag after changing the bits ESn[0:1] (n = 7 to 10).

INTM2

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

Table 7-10 INTM2 register contents (1/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7, 6</td><td>ES101
ES100</td><td>Specifies the INTP10 pin input valid edge.
<table>
<tr><th>ES101</th><th>ES100</th><th>Valid Edge Specification of INTP10 pin input</th></tr>
<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>Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>Both, rising and falling edges</td></tr>
</table></td></tr>
<tr><td>5, 4</td><td>ES91
ES90</td><td>Specifies the INTP9 pin input valid edge.
<table>
<tr><th>ES91</th><th>ES90</th><th>Valid Edge Specification of INTP9 pin input</th></tr>
<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>Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>Both, rising and falling edges</td></tr>
</table></td></tr>
<tr><td>3, 2</td><td>ES81
ES80</td><td>Specifies the INTP8 pin input valid edge.
<table>
<tr><th>ES81</th><th>ES80</th><th>Valid Edge Specification of INTP8 pin input</th></tr>
<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>Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>Both, rising and falling edges</td></tr>
</table></td></tr>
</table>

Table 7-10 INTM2 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>1, 0</td><td>ES71
ES70</td><td>Specifies the INTP7 pin input valid edge.
<table>
<tr><th>ES71</th><th>ES70</th><th>Valid Edge Specification of INTP7 pin input</th></tr>
<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>Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>Both, rising and falling edges</td></tr>
</table>
</td></tr>
</table>

(4) Interrupt mode register 3 (INTM3)

The behaviour of the external interrupt input pins INTP11 and INTP12 can be specified by the interrupt mode register 3 (INTM3).

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

Address $FFFFF886_H$

Initial Value 00_H. This register is cleared by any reset.

Caution Changing the state of interrupt mode configuration bits ESn[0:1] may trigger an unintended interrupt event for the corresponding interrupt channel. Be sure to mask the corresponding interrupt channel and clear the interrupt status flag after changing the bits ESn[0:1] (n = 11, 12).

	7	6	5	4	3	2	1	0
INTM3	0	0	0	0	ES121	ES120	ES111	ES110
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7-11 INTM3 register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7, 6</td><td>ES121
ES120</td><td>Specifies the INTP12 pin input valid edge.
<table>
<tr><th>ES121</th><th>ES120</th><th>Valid Edge Specification of INTP12 pin input</th></tr>
<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>Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>Both, rising and falling edges</td></tr>
</table></td></tr>
<tr><td>7, 6</td><td>ES111
ES110</td><td>Specifies the INTP11 pin input valid edge.
<table>
<tr><th>ES111</th><th>ES110</th><th>Valid Edge Specification of INTP11 pin input</th></tr>
<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>Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>Both, rising and falling edges</td></tr>
</table></td></tr>
</table>

7.4 Software Exception

A software exception is generated when the CPU executes the TRAP instruction, and is always accepted.

For details of the instruction function, refer to the *V850 Family User's Manual Architecture*.

7.4.1 Operation

If a software exception occurs, the CPU performs the following processing, and transfers control to the handler routine:

<1> Saves the current 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 PSW.
<5> Loads the handler address (00000040_H or 00000050_H) of the software exception routine in the PC, and transfers control.

The processing of a software exception is shown below.

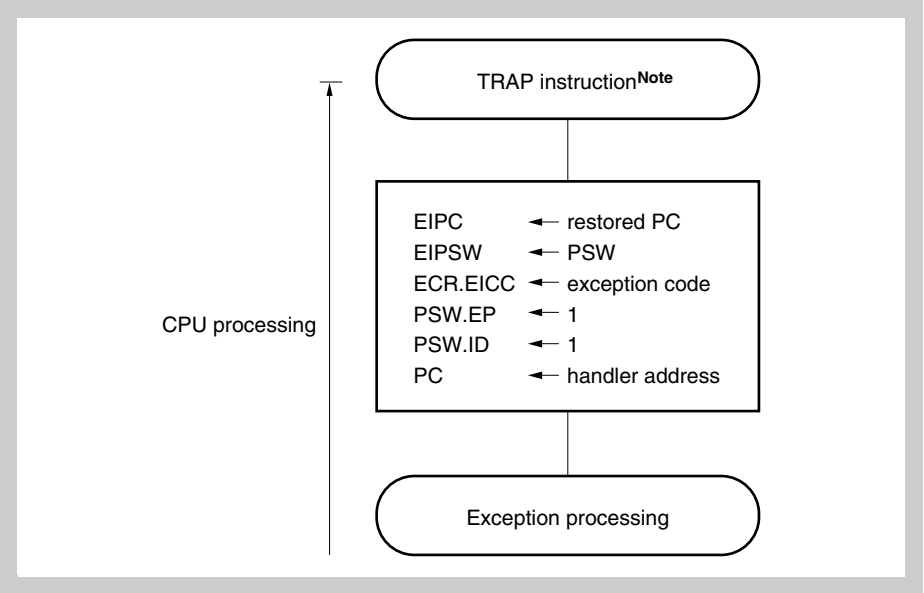


Figure 7-8 Software exception processing

Note TRAP instruction format: TRAP vector (the vector is a value from 0 to $1F_H$.)

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

7.4.2 Restore

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

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

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

The processing of the RETI instruction is shown below.

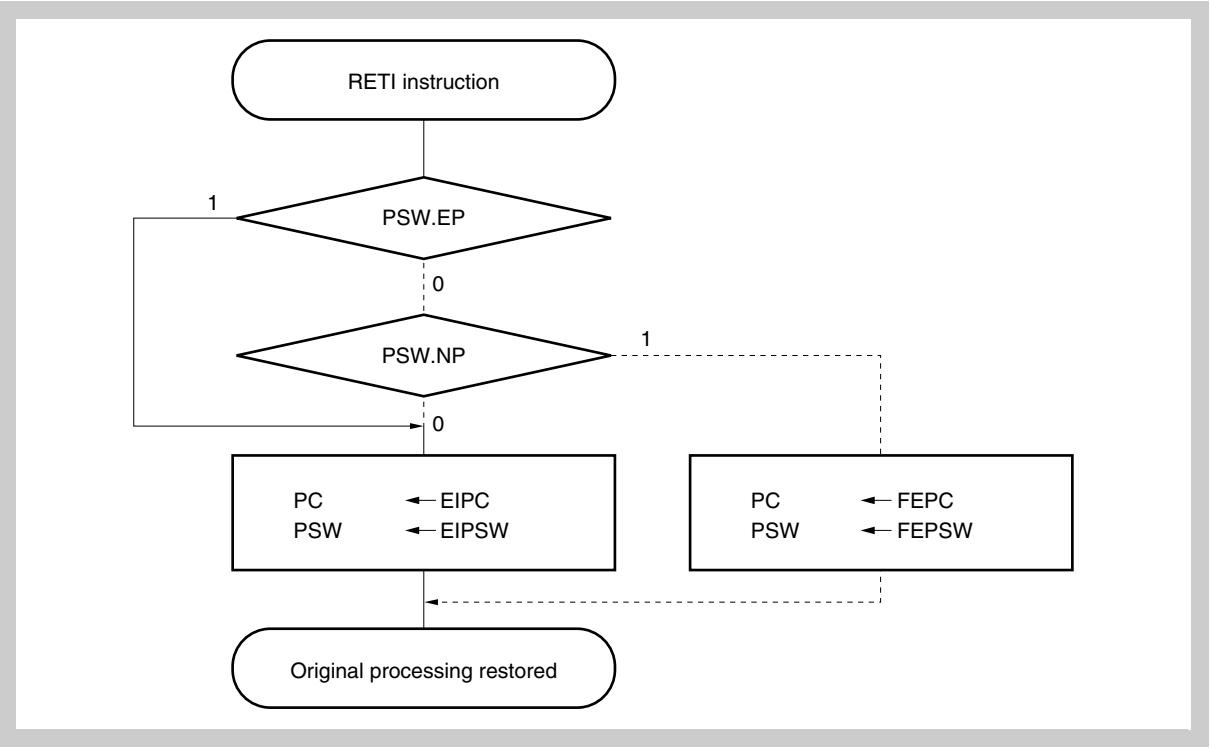


Figure 7-9 RETI instruction processing

Remark The solid line shows the CPU processing flow.

Caution When the PSW.EP bit and the PSW.NP bit are changed by the LDSR instruction
during the software exception process, 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 using the LDSR instruction immediately before the RETI instruction.

7.4.3 Exception status flag (EP)

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

Initial Value 00000020_H

PSW

31 … 8	7	6	5	4	3	2	1	0
	NP	**EP**	ID	SAT	CY	OV	S	Z

Table 7-12 Exception status flag (EP)

Bit position	Bit name	Function
5	EP	Exception processing status 0: Exception processing not in progress. 1: Exception processing in progress.

7.5 Exception Trap

An exception trap is an interrupt that is requested when the illegal execution of an instruction takes place. In the V850E/PH2, an illegal opcode trap (ILGOP: Illegal Opcode Trap) is considered as an exception trap.

7.5.1 Illegal opcode definition

The illegal instruction has an opcode (bits 10 to 5) of 111111_B, a sub-opcode (bits 26 to 23) of 1000_B to 1111_B, and a sub-opcode (bit 16) of 0. An exception trap is generated when an instruction applicable to this illegal instruction is executed.

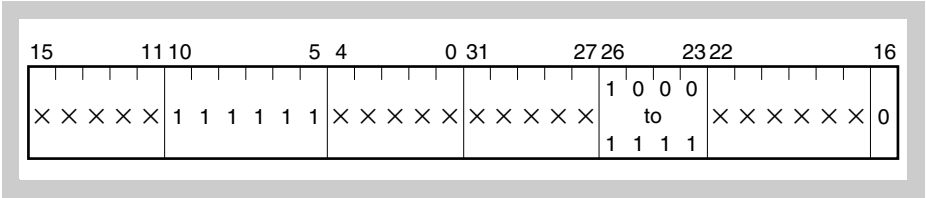


Figure 7-10 Illegal opcode

Remark x: don't care

Caution Caution Since it is possible that this instruction may be assigned to an illegal opcode in the future, it is recommended that it not be used.

(1) Operation

If an exception trap occurs, the CPU performs the following processing, and transfers control to the handler routine.

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

Figure 7-11 illustrates the processing of the exception trap.

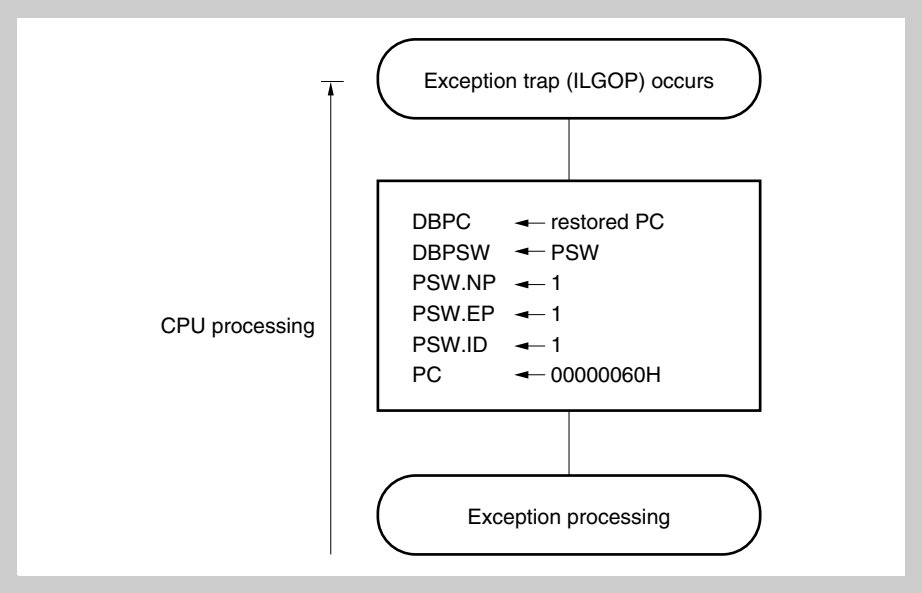


Figure 7-11 Exception trap processing

(2) Restore

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

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

Figure 7-12 illustrates the restore processing from an exception trap.

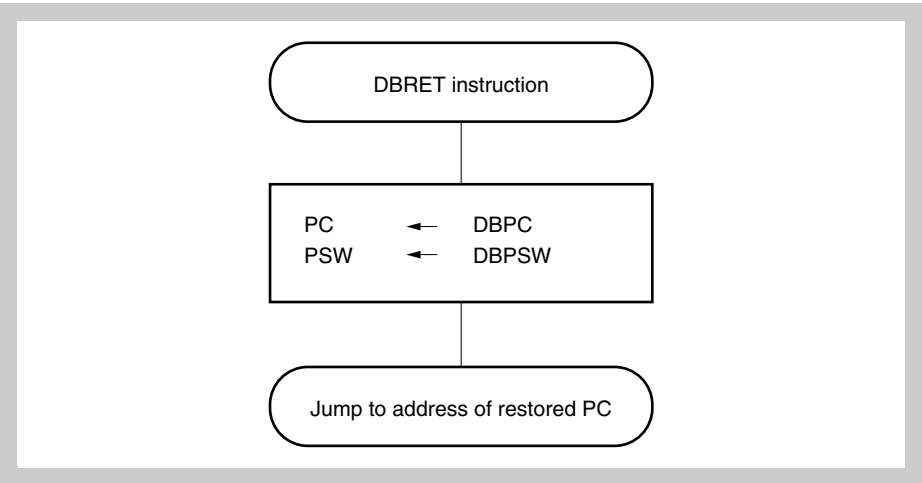


Figure 7-12 Restore processing from exception trap

7.6 Periods in Which CPU Does Not Acknowledge Interrupts

The CPU acknowledges an interrupt while an instruction is being executed. However, no interrupt will be acknowledged between an interrupt request non-sample instruction and the next instruction (interrupt is held pending).

The interrupt request non-sample instructions are as follows.

- EI instruction
- DI instruction
- LDSR reg2, 0x5 instruction (for PSW)
- The store instruction for the command register (PRCMD)
- The store, or bit manipulation instructions excluding the TST1 instruction for the following interrupt-related registers:
- Interrupt control registers (PICn)**Note**
- Interrupt mask registers 0 to 3 (IMR0 to IMR3)

Note n = 0 to 105, refer to *Table 7-4, “Adresses and bits of interrupt control registers,” on page 189*)

Chapter 8 Clock Generator

The clock generator (CG) generates and controls the internal system clock (f_{XX}) that is supplied to each internal unit, such as the CPU.

8.1 Features

- Multiplier function using a phase locked loop (PLL) synthesizer ($f_{XX} = 4 \times f_X$)
  - Crystal frequency: f_X = 16 MHz
  - Internal system clock: f_{XX} = 64 MHz
- Power saving mode: HALT mode

8.2 Configuration

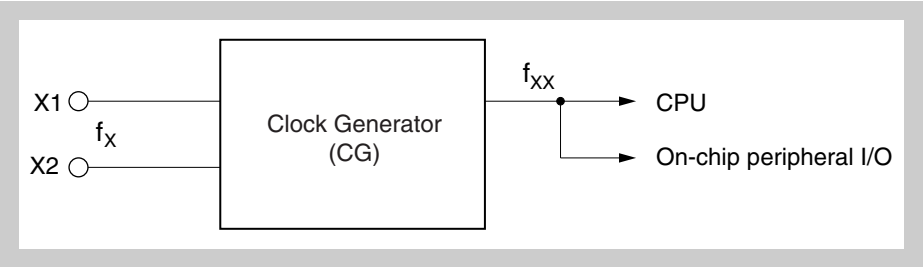


Figure 8-1 Clock generator

Remark f_X: External resonator or external clock frequency
f_{XX}: Internal system clock

An external resonator or crystal is connected to X1 and X2 pins, whose frequency is multiplied by the PLL synthesizer. By this an internal system clock (f_{XX}) is generated that is 4 times the frequency (f_X) of the external resonator or crystal.

The clock controller enables PLL automatically and starts clock supply to the system after oscillation stabilization time has passed.

Internal System Clock Frequency (f_{XX})	External Resonator or Crystal Frequency (f_X)
64.000 MHz	16.0000 MHz

8.3 Power Save Control

8.3.1 Overview

The power save function of V850E/PH2 supports the HALT mode only. In this mode, the clock generator (oscillator and PLL synthesizer) continues to operate, but the CPU's operation clock stops. Since the supply of clocks to on-chip peripheral functions other than the CPU continues, operation continues. The power consumption of the overall system can be reduced by intermittent operation that is achieved due to a combination of HALT mode and normal operation mode.

The system is switched to HALT mode by a specific instruction (the HALT instruction).

Figure 8-2 shows the operation of the clock generator in normal operation mode and HALT mode.

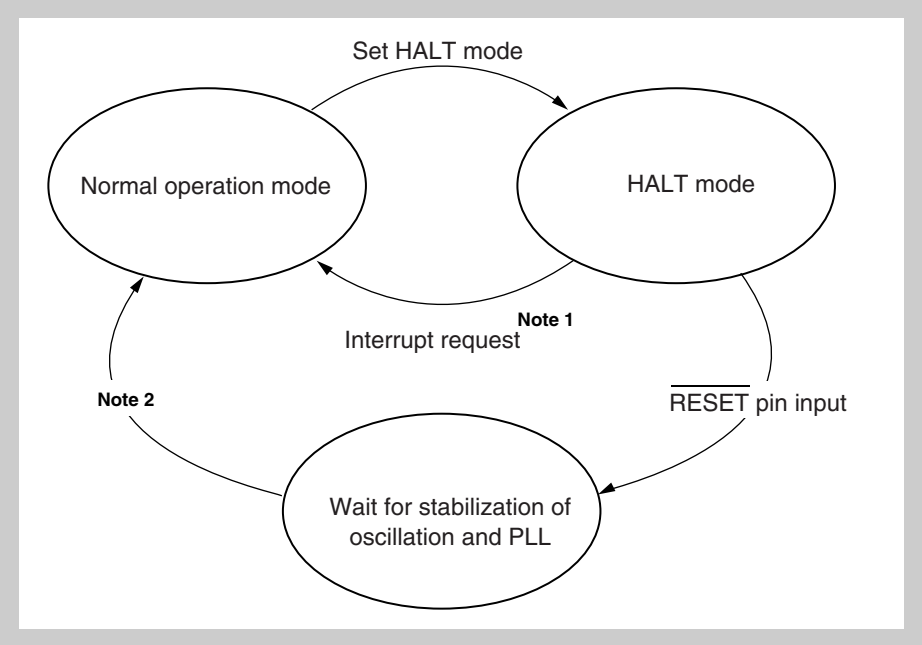


Figure 8-2 Power save mode state transition diagram

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 default.

8.3.2 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 18-3 shows the operation status in the HALT mode.

The average power consumption of the system can be reduced by using the HALT mode in combination with the normal operation mode for intermittent operation.

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.

Table 8-1 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.
A0 to A21	Operating
D0 to D31	
$\overline{RD}$	
$\overline{WR}$	
$\overline{BEN0}$ to $\overline{BEN3}$	
$\overline{CS0}$, $\overline{CS1}$, $\overline{CS3}$, $\overline{CS4}$	
$\overline{WAIT}$	

(2) Releasing HALT mode

The HALT mode is released by a non-maskable interrupt request signal (NMI), an unmasked maskable interrupt request signal, or RESET pin input.

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 (INTWDT) or an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request. If the HALT mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is serviced as follows.

- 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 8-2 Operation after releasing HALT mode by interrupt request signal

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request signal	Execution branches to the handler address	
Unmasked maskable interrupt request signal	Execution branches to the handler address or the next instruction is executed	The next instruction is executed

(b) Releasing HALT mode by RESET pin input or WDTRES signal generation

The same operation as the normal reset operation is performed.

Chapter 9 16-Bit Timer/Event Counter P (TMP)

This microcontroller has nine instances of this 16-bit timer/event counter P, TMP0 to TMP8.

Throughout this chapter, the individual instances of Timer P are identified by “n” (n = 0 to 8), for example, TMPnCTL0 for the TMPn control register 0.

9.1 Features

Timer P (TMP) is a 16-bit timer/event counter provided with general-purpose functions.

TMP 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

9.2 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 (TOPn0, TOPn1) × 2

9.3 Configuration

TMP includes the following hardware.

Table 9-1 Timer TMP registers and external connections

Item	Configuration
Timer registers	• 16-bit counter • TMPn read buffer register (TPnCNT) • TMPn capture/compare registers 0, 1 (TPnCCR0, TPnCCR1) • Internal CCR buffer register 0, 1 (CCR0, CCR1)
Timer input[a]	TIPn0, TIPn1, TTRGPn, TEVTPn
Timer output	TOPn0[b], TOPn1
Timer control registers	• TMPn control registers 0, 1 (TPnCTL0, TPnCTL1) • TMPn I/O control registers 0 to 2 (TPnIOC0 to TPnIOC2) • TMPn option registers 0, 1 (TPnOPT0, TPnOPT1)
Input selection registers	TMPn input selection registers (TPIC0 to TPIC2)

a) Timer inputs not available for TMP8.
b) Timer output TOP80 not availabel for TMP8.

Note Timer P (TMP) pins are alternate functions of port pins. For how to set the alternate function, refer to the corresponding description of the registers in chapter*"Port Pin Functions" on page 870*.

The block diagram of the timer TMP is shown below.

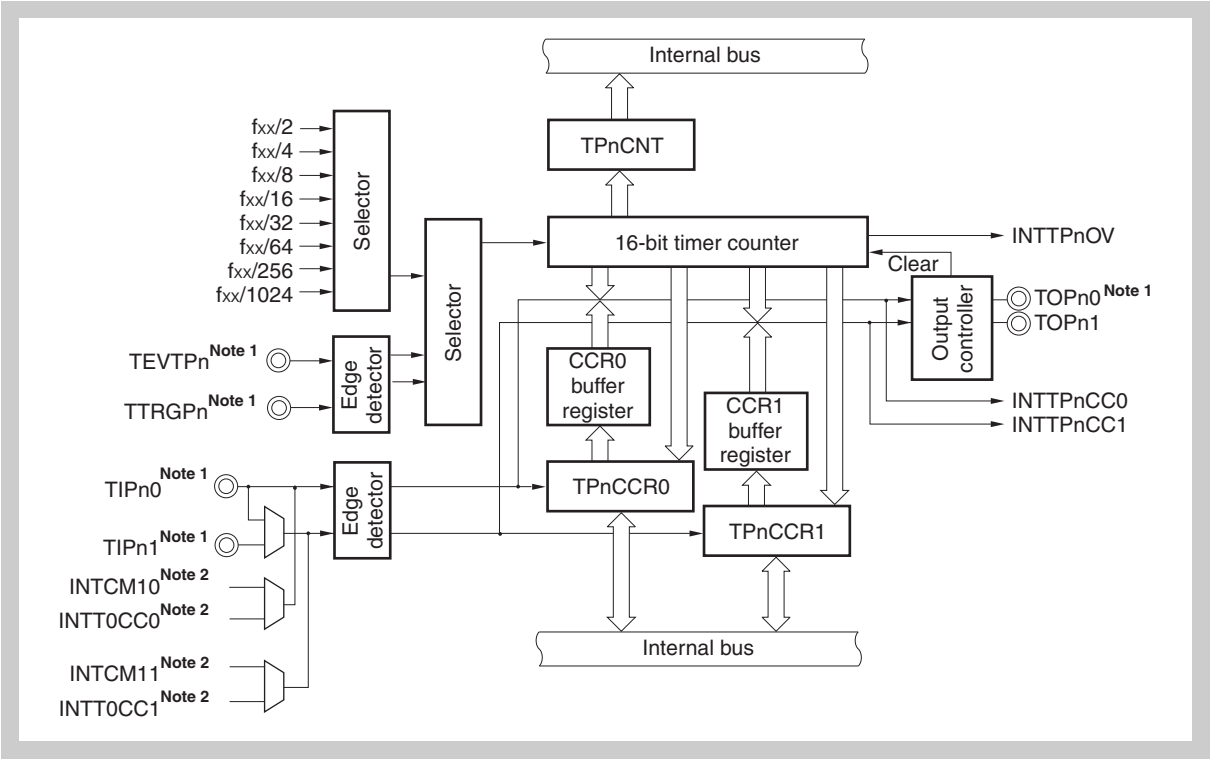


Figure 9-1 Block diagram of timer P

Note 1. External pin is not available for TMP8.

2. Internal signal inputs (INTTT0CC0 and INTTT0CC1 of TMT0, or INTCM10 and INTCM11 of TMENC1) available on TMP8 only. (refer to *"TMP input control register 2 (TPIC2)" on page 227*).

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

The TPnCCR0 register is a 16-bit register that functions both as a capture register and as a compare register.

Whether this register functions as a capture register or as a compare register can be controlled with the TPnCCS0 bit of the TPnOPT0 register, but only in the free-running mode.

In the pulse width measurement mode, this register can be used as a dedicated capture register (the compare function cannot be used.)

In modes other than the free-running mode and pulse width measurement mode, this register is used as a dedicated compare register.

In the initial setting, the TPnCCR0 register is a compare register.

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

Address

TP0CCR0:	$FFFFF606_H$,	TP1CCR0:	$FFFFF616_H$,
TP2CCR0:	$FFFFF626_H$,	TP3CCR0:	$FFFFF636_H$,
TP4CCR0:	$FFFFF646_H$,	TP5CCR0:	$FFFFF656_H$,
TP6CCR0:	$FFFFF666_H$,	TP7CCR0:	$FFFFF676_H$,
TP8CCR0:	$FFFFF686_H$		

Initial Value 0000_H. This register is cleared by any reset.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPnCCR0	Compare value 0															
	R/W															

(a) Use as compare register

When used as a compare register, TPnCCR0 can be rewritten when TMPnCE = 1, as shown below:

TMP Operation Mode	Method of Writing TPnCCR0 Register
PWM mode External trigger pulse output mode	Reload
Free-running mode External event count mode One-shot pulse output mode Interval timer mode	Anytime write
Pulse width measurement mode	Not applicable (used as capture register)

(b) Use as capture register

- **TMP0 to TMP7**
  The counter value is saved to TPnCCR0 upon capture trigger (TIPn0) input edge detection.
- **TMP8**
  Since TMP8 has no external input pin, the capture function can only be used internally for capturing the interrupt signal (INTTT0CC0 of TMT0, or INTCM10 of TMENC1) specified by the TPIC22 bit of TPIC2 register (refer to *"TMP input control register 2 (TPIC2)" on page 227*).

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

The TPnCCR1 register is a 16-bit register that functions both as a capture register and as a compare register.

Whether this register functions as a capture register or as a compare register can be controlled with the TPnCCS1 bit of the TPnOPT0 register, but only in the free-running mode.

In the pulse width measurement mode, this register can be used as a dedicated capture register (the compare function cannot be used.)

In modes other than the free-running mode and pulse width measurement mode, this register is used as a dedicated compare register.

In the initial setting, the TPnCCR1 register is a reload register.

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

Address
TP0CCR1: $FFFFF608_H$, TP1CCR1: $FFFFF618_H$,
TP2CCR1: $FFFFF628_H$, TP3CCR1: $FFFFF638_H$,
TP4CCR1: $FFFFF648_H$, TP5CCR1: $FFFFF658_H$,
TP6CCR1: $FFFFF668_H$, TP7CCR1: $FFFFF678_H$,
TP8CCR1: $FFFFF688_H$

Initial value 0000_H. This register is cleared by any reset.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPnCCR1	Compare value 1															
	R/W															

(a) Use as compare register

When used as a compare register, TPnCCR1 can be rewritten when TMPnCE = 1, as shown below:

TMP Operation Mode	Method of Writing TPnCCR0 Register
PWM mode, external trigger pulse output mode	Reload
Free-running mode, external event count mode, one-shot pulse output mode, interval timer mode	Anytime write
Pulse width measurement mode	Cannot be used because dedicated capture register

(b) Use as capture register

- **TMP0 to TMP7**
  The counter value is saved to TPnCCR1 upon capture trigger (TIPn1) input edge detection.
- **TMP8**
  Since TMP8 has no external input pin, the capture function can only be used internally for capturing the interrupt signal (INTTT0CC1 of TMT0, or INTCM11 of TMENC1) specified by the TPIC22 bit of TPIC2 register (refer to *"TMP input control register 2 (TPIC2)" on page 227*).

(3) TMPn counter register (TPnCNT)

The TPnCNT register is a read buffer register that can read 16-bit counter values.

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

Address

TP0CNT:	$FFFFF60A_H$,	TP1CNT:	$FFFFF61A_H$,
TP2CNT:	$FFFFF62A_H$,	TP3CNT:	$FFFFF63A_H$,
TP4CNT:	$FFFFF64A_H$,	TP5CNT:	$FFFFF65A_H$,
TP6CNT:	$FFFFF66A_H$,	TP7CNT:	$FFFFF67A_H$,
TP8CNT:	$FFFFF68A_H$		

Initial value 0000_H. This register is cleared by any reset or when TPnCTL0.TPnCE is set to 0.
If the TPnCNT register is read when TPnCTL0.TPnCE = 0, the value of the 16-bit counter ($FFFF_H$) is not read, but 0000_H instead.

TPnCNT

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

9.4 Control Registers

(1) TMPn control register 0 (TPnCTL0)

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

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

Address
TP0CTL0: $FFFFF600_H$, TP1CTL0: $FFFFF610_H$,
TP2CTL0: $FFFFF620_H$, TP3CTL0: $FFFFF630_H$,
TP4CTL0: $FFFFF640_H$, TP5CTL0: $FFFFF650_H$,
TP6CTL0: $FFFFF660_H$, TP7CTL0: $FFFFF670_H$,
TP8CTL0: $FFFFF680_H$

Initial value 00_H. This register is cleared by any reset

TPnCTL0

7	6	5	4	3	2	1	0
TPnCE	0	0	0	0	TPnCKS2	TPnCKS1	TPnCKS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Caution Only the TPnCE bit of the TPnCTL0 register may be changed during operation of timer P (when TPnCE is 1). Do not rewrite any other bit than the TPnCE bit.

Table 9-2 TPnCTL0 register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7</td><td>TPnCE</td><td>Specifies the timer Pn operation.
0: Internal operating clock disabled (TMPn is reset asynchronously)
1: Internal operating clock enabled

TPnCE bit controls internal operating clock of TMPn. When the TPnCE bit is cleared to 0, the internal operating clock iof TMPn stops (fixed to low level) and TMPn counter is reset asynchronously.
When the TPnCE bit is set to 1, the internal operating clock is enabled and count-up operation starts within 2 clock cycles.</td></tr>
<tr><td>2 to 0</td><td>TPnCKS2
TPnCKS1
TPnCKS0</td><td>Selects the count clock of timer Pn.
<table>
<tr><th>TPnCKS2</th><th>TPnCKS1</th><th>TPnCKS0</th><th>Internal Count Clock Selection</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>$f_{XX}/2$</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>$f_{XX}/4$</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>$f_{XX}/8$</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>$f_{XX}/16$</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>$f_{XX}/32$</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>$f_{XX}/64$</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>$f_{XX}/256$</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>$f_{XX}/1024$</td></tr>
</table>
Caution: Set the TPnCKS2 to TPnCKS0 bits when TPnCE = 0. When the value of the TPnCE bit is changed from 0 to 1, the TPnCKS2 to TPnCKS0 bits can be set simultaneously.</td></tr>
</table>

(2) TMPn control register 1 (TPnCTL1)

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

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

Address

TP0CTL1:	$FFFFF601_H$,	TP1CTL1:	$FFFFF611_H$,
TP2CTL1:	$FFFFF621_H$,	TP3CTL1:	$FFFFF631_H$,
TP4CTL1:	$FFFFF641_H$,	TP5CTL1:	$FFFFF651_H$,
TP6CTL1:	$FFFFF661_H$,	TP7CTL1:	$FFFFF671_H$,
TP8CTL1:	$FFFFF681_H$		

Initial value 00_H. This register is cleared by any reset.

TPnCTL1

7	6	5	4	3	2	1	0
TPnSYE	TPnEST	TPnEEE	0	0	TPnMD2	TPnMD1	TPnMD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 9-3 TPnCTL1 register contents (1/2)

Bit position	Bit name	Function
7	TPnSYE	Selects single or synchronous mode operation of timer Pn. 0: Timer Pn operates in single operation mode 1: Timer Pn operates in synchronous operation mode • This bit supports synchronous operation of two or more timer P. • Two groups of timers exist, which can be synchronized: TMP0 to TMP3 with TMP0 as master, and TMP4 to TMP7 with TMP4 as master. **Caution:** 1. The TPnSYE bit must not be set to 1 for master timers (TMP0, TMP4). Therefore always keep TP0SYE = TP4SYE = 0. 2. Synchronous operation mode is not available for TMP8. Thus do not operate TMP8 in synchronous mode and clear TP8SYE bit to 0. **Note:** Synchronous operation mode is not available for TMP8 (n = 8)
6	TPnEST	Enable/disable software trigger control of timer Pn. 0: Disables software trigger control. 1: Enables software trigger control. • In one-shot pulse mode: One-shot pulse software trigger • In external trigger pulse output mode: Pulse output software trigger **Caution:** The TPnEST bit functions as a software trigger in the one-shot pulse mode or the external trigger pulse output mode, if it is set to 1 when TPnCE = 1. Therefore, be sure to set TPnEST to 1 after setting TPnCE to 1. **Note:** 1. The read value of the TPnEST bit is always 0. 2. The TRnEST bit is invalid even if it is controlled in any other mode.

Table 9-3 TPnCTL1 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>5</td><td>TPnEEE</td><td>Specifies count clock.
0: Use the clock selected with the TPnCKS2 to TPnCKS0 bits.
1: Use the external clock (TEVTPn pin input edge).
The valid edge during TSnEEE = 1 (external clock: TEVTPn pin) is specified by the TPnEES[1:0] bits of the TPnIOC2 register.

Caution: Set TP8EEE bit of the TR0CTL1 register always to 0, since TMP8 does not incorporate an external clock input. In case of TP8EEE = 1 operation of TMP8 is not guaranteed.

Note: The TSnEEE bit is ignored in the external event count mode.</td></tr>
<tr><td>2 to 0</td><td>TPnCKS2
TPnCKS1
TPnCKS0</td><td>Selects the mode of timer Pn.
<table>
<tr><th>TPnMD2</th><th>TPnMD1</th><th>TPnMD0</th><th>Timer Mode Selection</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Interval timer mode[a b]</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>External event count mode[a b c]</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>External trigger pulse output mode[b c]</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>One-shot pulse mode[b]</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>PWM mode[b]</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[a b]</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Setting prohibited</td></tr>
</table>
a) Setting prohibited for TMP0 and TMP4, when synchronous operation function is enabled (TPnSYE = 1).
b) Setting prohibited for TMP1 to TMP3, and TMP5 to TMP7, when synchronous operation function is enabled (TPnSYE = 1).
c) Setting prohibited for TMP8.

Caution: Rewrite the TPnEEE and TPnMD2 to TPnMD0 bits only when TPnCE = 0. (The same value can be written when TPnCE = 1.) The operation is not guaranteed if rewriting is performed when TPnCE = 1. If rewriting was mistakenly performed, set TPnCE = 0 and then set the bits again.</td></tr>
</table>

(3) TMPn I/O control register 0 (TPnIOC0)

The TPnIOC0 register is an 8-bit register that controls the timer output (TOPn0, TOPn1).

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

Address

TP0IOC0:	$FFFFF602_H$,	TP1IOC0:	$FFFFF612_H$,
TP2IOC0:	$FFFFF622_H$,	TP3IOC0:	$FFFFF632_H$,
TP4IOC0:	$FFFFF642_H$,	TP5IOC0:	$FFFFF652_H$,
TP6IOC0:	$FFFFF662_H$,	TP7IOC0:	$FFFFF672_H$,
TP8IOC0:	$FFFFF682_H$		

Initial value 00_H. This register is cleared by any reset.

TPnIOC0

7	6	5	4	3	2	1	0
0	0	0	0	TPnOL1	TPnOE1	TPnOL0	TPnOE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Caution Rewrite the TPnOL1, TPnOE1, TPnOl0, and TPnOE0 bits only when TPnCE = 0. (The same value can be written when TPnCE = 1.) If rewriting was mistakenly performed, set TPnCE = 0 and then set the bits again.

Table 9-4 TPnIOC0 register contents

Bit position	Bit name	Function
3	TPnOL1	Sets the TOPn1 output level. 0: Active level is high level. 1: Active level is low level.
2	TPnOE1	Enables/disables the TOPn1 pin output. 0: Timer output disabled (Output is inactive level of the TPnOL1 bit). 1: Timer output enabled
1	TPnOL0	Sets the TOPn0 output level. 0: Active level is high level. 1: Active level is low level.
0	TPnOE0	Enables/disables the TOPn0 pin output. 0: Timer output disabled (Output is inactive level of the TPnOL0 bit). 1: Timer output enabled **Note:** Timer output TOPn0 not available for TMP8.

(4) TMPn I/O control register 1 (TPnIOC1)

The TPnIOC1 register is an 8-bit register that controls the valid edge for the external input signals (TIPn0, TIPn1).

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

Address

TP0IOC1:	$FFFFF603_H$,	TP1IOC1:	$FFFFF613_H$,
TP2IOC1:	$FFFFF623_H$,	TP3IOC1:	$FFFFF633_H$,
TP4IOC1:	$FFFFF643_H$,	TP5IOC1:	$FFFFF653_H$,
TP6IOC1:	$FFFFF663_H$,	TP7IOC1:	$FFFFF673_H$,
TP8IOC1:	$FFFFF683_H$		

Initial value 00_H. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
TPnIOC1	0	0	0	0	TPnIS3	TPnIS2	TPnIS1	TPnIS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Caution

1. Rewrite the TPnIS3 to TPnIS0 bits only when TPnCE = 0. (The same value can be written when TPnCE = 1.) If rewriting was mistakenly performed, set TPnCE = 0 and then set the bits again.
2. The TPnIS3 to TPnIS0 bits are valid only in the free-running mode and the pulse width measurement mode. In all other modes, a capture operation is not possible.

Table 9-5 TPnIOC1 register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3, 2</td><td>TPnIS3, TPnIS2</td><td>Sets the valid edge of the TIPn1 capture input [a]
<table>
<tr><th>TPnIS3</th><th>TPnIS2</th><th>Capture Input (TIPn1) 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></td></tr>
<tr><td>1, 0</td><td>TPnIS1, TPnIS0</td><td>Sets the valid edge of the TIPn0 capture input [a]
<table>
<tr><th>TPnIS1</th><th>TPnIS0</th><th>Capture Input (TIPn0) 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></td></tr>
</table>

a) TIPn0 and TIPn1 input pins are not available for TMP8. These inputs are only connected internally to capture the interrupt signals INTTT0CC0 and INTT0CC1 of TMT0, or INTCM10 and INTCM11 of TMENC1, specified by the TPIC22 bit of TPIC2 register (ref. to *"TMP input control register 2 (TPIC2)" on page 227*).

(5) TMPn I/O control register 2 (TPnIOC2)

The TPnIOC2 register is an 8-bit register that controls the valid edge of the external event count input signal (TEVTPn) and external trigger input signal (TTRGPn).

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

Address
TP0IOC2: $FFFFF604_H$, TP1IOC2: $FFFFF614_H$,
TP2IOC2: $FFFFF624_H$, TP3IOC2: $FFFFF634_H$,
TP4IOC2: $FFFFF644_H$, TP5IOC2: $FFFFF654_H$,
TP6IOC2: $FFFFF664_H$, TP7IOC2: $FFFFF674_H$,

Initial value 00_H. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
TPnIOC2	0	0	0	0	TPnEES1	TPnEES0	TPnETS1	TPnETS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Caution
1. Rewrite the TPnEES1, TPnEES0, TPnEST1, and TPnEST0 bits only when TPnCE = 0. (The same value can be written when TPnCE = 1.) If rewriting was mistakenly performed, set TPnCE = 0 and then set the bits again.
2. The TPnEES1 and TPnEES0 bits are valid only when TPnEEE = 1 or when the external event count mode (TPnMD2 to TPnMD0 = 001B of the TPnCTL1 register) has been set.

Table 9-6 TPnIOC2register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3, 2</td><td>TPnEES1,
TPnEES0</td><td>Sets the valid edge of the TIPn1 capture input
<table>
<tr><th>TPnEES1</th><th>TPnEES0</th><th>External Event Counter Input (TEVTPn) 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></td></tr>
<tr><td>1, 0</td><td>TPnETS1,
TPnETS0</td><td>Sets the valid edge of the TIPn0 capture input
<table>
<tr><th>TPnETS1</th><th>TPnETS0</th><th>External Trigger Input (TTRGPn) 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></td></tr>
</table>

Note External event count intput (TEVTP8) and external trigger input (TTRGP8) are not available for TMP8.

(6) TMPn option register 0 (TPnOPT0)

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

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

Address

TP0OPT0:	$FFFFF605_H$,	TP1OPT0:	$FFFFF615_H$,
TP2OPT0:	$FFFFF625_H$,	TP3OPT0:	$FFFFF635_H$,
TP4OPT0:	$FFFFF645_H$,	TP5OPT0:	$FFFFF655_H$,
TP6OPT0:	$FFFFF665_H$,	TP7OPT0:	$FFFFF675_H$,
TP8OPT0:	$FFFFF685_H$		

Initial value 00_H. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
TPnOPT0	0	0	TPnCCS1	TPnCCS0	0	0	0	TPnOVF
	R	R	R	R	R/W	R/W	R/W	R/W

Table 9-7 TPnOPT0 register contents

Bit position	Bit name	Function
5	TPnCCS1	Selects the TPnCCR1 register capture/compare function 0: Compare register selection 1: Capture register selection **Caution:** Rewrite the TPnCCS1 bit only when TPnCE = 0. (The same value can be written when TPnCE = 1.) If rewriting was mistakenly performed, set TPnCE = 0 and then set the bits again. **Note:** The TPnCCS1 bit setting is valid only in the free-running mode.
4	TPnCCS0	Selects the TPnCCR0 register capture/compare function 0: Compare register selection 1: Capture register selection **Caution:** Rewrite the TPnCCS0 bit only when TPnCE = 0. (The same value can be written when TPnCE = 1.) If rewriting was mistakenly performed, set TPnCE = 0 and then set the bits again. **Note:** The TPnCCS0 bit setting is valid only in the free-running mode.
0	TPnOVF	This is a flag indicating the timer Pn overflow. It is set to 1 when the 16-bit counter value overflows from $FFFF_H$ to 0000_H. This flag is cleared by writing 0 or setting TPnCE to 0. An overflow interrupt (INTTPnOV) occurs at the same time that the TPnOVF bit is set to 1. **Caution:** 1. Overflow occurs only in the free-running mode or the pulse measurement mode. 2. The TPnOVF bit is not cleared even if the TPnOVF bit and the TPnOPT0 register are read while TPnOVF = 1. 3. Reading or writing is possible for the TPnOVF bit, but writing 1 to the TPnOVF bit is ignored.

(7) TMP input control register 0 (TPIC0)

The TPIC0 register is an 8-bit register that controls the external input pin source of the capture register 1 of TMP0 to TMP3.

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

Address $FFFFF6F0_H$

Initial value 00_H. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
TPIC0	0	0	0	0	TPIC03	TPIC02	TPIC01	TPIC00
	R	R	R	R	R/W	R/W	R/W	R/W

Table 9-8 TPIC0 register contents

Bit position	Bit name	Function
3	TPIC03	Selects the capture input for TP3CCR1 register. 0: Capture source input is pin P17/TIP31 1: Capture source input is pin P16/TIP30
2	TPIC02	Selects the capture input for TP2CCR1 register. 0: Capture source input is pin P15/TIP21 1: Capture source input is pin P14/TIP20
1	TPIC01	Selects the capture input for TP1CCR1 register. 0: Capture source input is pin P13/TIP11 1: Capture source input is pin P12/TIP10
0	TPIC00	Selects the capture input for TP3CCR1 register. 0: Capture source input is pin P11/TIP01 1: Capture source input is pin P10/TIP00

(8) TMP input control register 1 (TPIC1)

The TPIC1 register is an 8-bit register that controls the external input pin source of the capture register 1 of TMP4 to TMP7, as well as the internal time trigger source from the AFCAN controllers of both capture registers 0 and 1 of TMP7.

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

Address $FFFFF6F2_H$

Initial value 00_H. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
TPIC1	0	0	TPIC15	TPIC14	TPIC13	TPIC12	TPIC11	TPIC10
	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 9-9 TPIC1 register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>5, 3</td><td>TPIC15, TPIC13</td><td>Selects the capture input for TP7CCR1 register.
<table>
<tr><th>TPIC15</th><th>TPIC13</th><th>Capture source input selection of TP7CCR1</th></tr>
<tr><td>0</td><td>0</td><td>Pin P27/TIP71</td></tr>
<tr><td>0</td><td>1</td><td>Pin P26/TIP70</td></tr>
<tr><td>1</td><td>×</td><td>AFCAN1 timer trigger</td></tr>
</table>
Caution: Setting of TPIC15 to 1 is prohibited for µPD70F3447, since the AFCAN1 controller is not available.</td></tr>
<tr><td>4</td><td>TPIC14</td><td>Selects the capture input for TP7CCR0 register.
0: Capture source input is pin P26/TIP70
1: Capture source input is AFCAN0 time trigger</td></tr>
<tr><td>2</td><td>TPIC12</td><td>Selects the capture input for TP6CCR1 register.
0: Capture source input is pin P25/TIP61
1: Capture source input is pin P24/TIP60</td></tr>
<tr><td>1</td><td>TPIC11</td><td>Selects the capture input for TP5CCR1 register.
0: Capture source input is pin P23/TIP51
1: Capture source input is pin P22/TIP50</td></tr>
<tr><td>0</td><td>TPIC10</td><td>Selects the capture input for TP4CCR1 register.
0: Capture source input is pin P21/TIP41
1: Capture source input is pin P20/TIP40</td></tr>
</table>

(9) TMP input control register 2 (TPIC2)

The TPIC2 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 TMP8.

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

Address FFFFF6F4$_H$

Initial value 00$_H$. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
TPIC2	0	0	0	0	0	TPIC22	TPIC21	TPIC20
	R	R	R	R	R	R/W	R/W	R/W

Table 9-10 TPIC2 register contents

Bit position	Bit name	Function
2	TPIC22	Selects the capture inputs for TP8CCR1 and TP8CCR0 registers. **TPIC22** / **Capture source input selection of TP8CCR1** / **TP8CCR0** 0 / INTT0CC1 signal of TMT0 / INTT0CC0 signal of TMT0 1 / INTCM11signal of TMENC1 / INTCM10 signalof TMENC1 **Caution:** Setting of TPIC22 to 1 is prohibited for µPD70F3447, since the timer TMENC1 is not available.
1	TPIC21	Selects the capture input for TT1CCR1 register. 0: Capture source input is pin P74/TIT11 1:Capture source input is pin P73/TIT10
0	TPIC20	Selects the capture input for TT0CCR1 register. 0: Capture source input is pin P71/TIT01 1: Capture source input is pin P70/TIT00

9.5 Operation

Timer P can perform the following operations.

Operation	TPnEST (Software Trigger Bit)	TTRGPn0 (External Trigger Input)	Capture/Compare Mode	Compare Register Rewriting Method
Interval timer mode	Invalid	Invalid	Compare only	Anytime rewrite
External event count mode[a]	Invalid	Invalid	Compare only	Anytime rewrite
External trigger pulse output mode[b]	Valid	Valid	Compare only	Reload
One-shot pulse output mode[b]	Valid	Valid	Compare only	Anytime rewrite
PWM mode	Invalid	Invalid	Compare only	Reload
Free-running mode	Invalid	Invalid	Capture/compare selectable	Anytime rewrite
Pulse width measurement mode[b]	Invalid	Invalid	Capture only	Not applicable

a) To use the external event count function, specify that the edge of the capture input TIPn1 or TIPn0 respectively, shared with event input TEVTPn is not detected (by clearing the TPnIOC1.TPISn[3:2] or TPnIOC1.TPISn[1:0] respectively to “00B”)

b) When using the external trigger pulse output mode, one-shot pulse mode, and pulse width measurement mode, select a count clock (by clearing the TPnCTL1.TPnEEE bit to 0).

9.5.1 Anytime rewrite and reload

TPnCCR0 and TPnCCR1 register rewrite is possible for timer P during timer operation (TPnCE = 1), but the write method (anytime rewrite, reload) differs depending on the mode.

(1) Anytime rewrite

When the TPnCCRm register is written during timer operation, the write data is transferred at that time to the CCRm buffer register and used as the 16-bit counter comparison value.

The following flowchart illustrates an example of the operation in the interval timer mode.

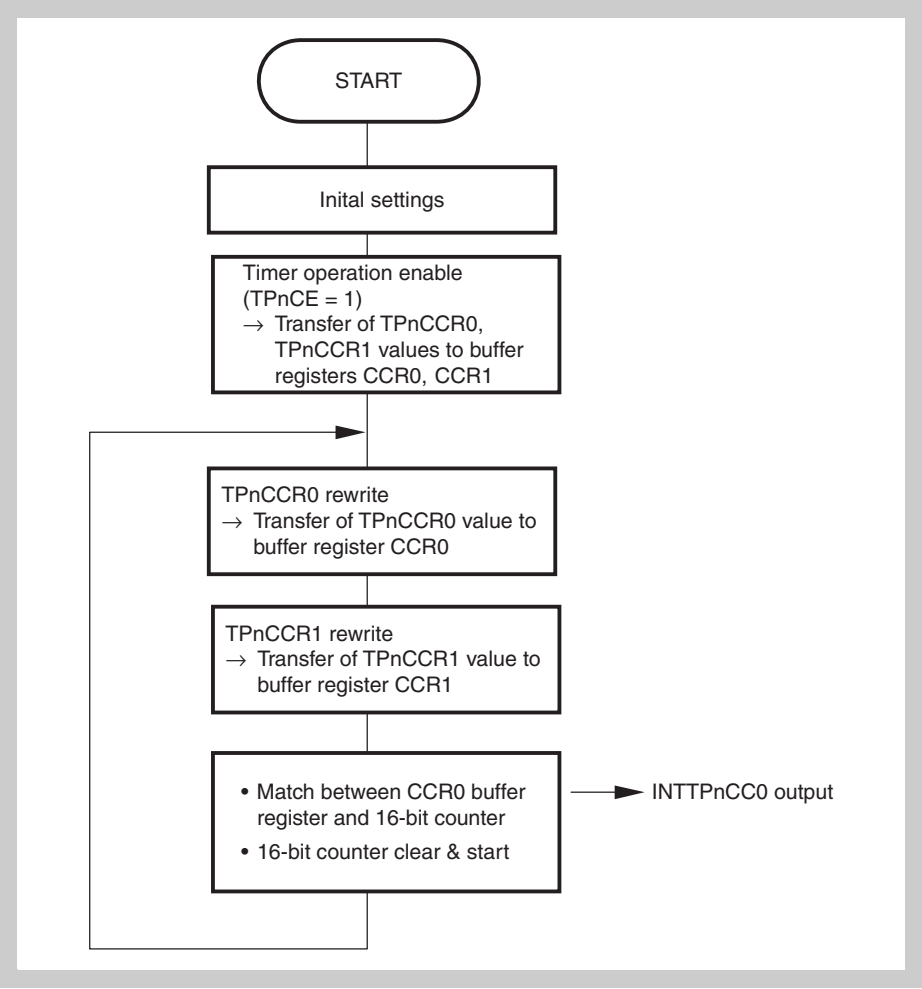


Figure 9-2 Basic operation flow for anytime write

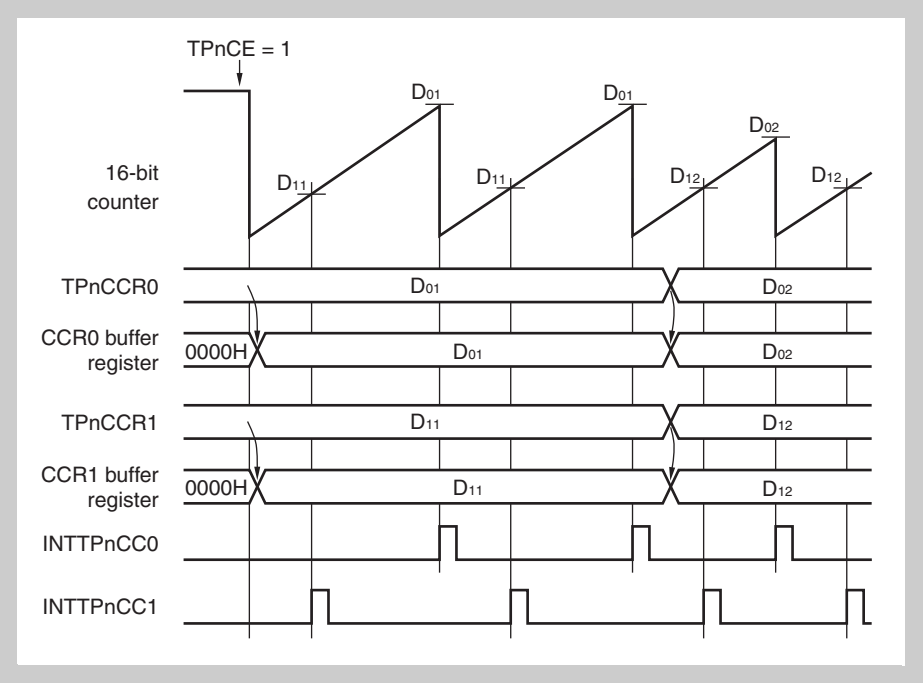


Figure 9-3 Timing diagram of anytime write

Remark D_{01}, D_{02}: Setting values of TPnCCR0 register (0000_H to $FFFF_H$)
D_{11}, D_{12}: Setting values of TPnCCR1 register (0000_H to $FFFF_H$)

(2) Reload method (Batch Rewrite)

When the TPnCCR0 and TPnCCR1 registers are written during timer operation via the CCRm buffer register, the write data is used as the 16-bit counter comparison value. The TPnCCR0 register and the TPnCCR1 register can be rewritten when TPnCE = 1.

In order for the setting value when the TPnCCR0 register and the TPnCCR1 register are rewritten to become the 16-bit counter comparison value (in other words, in order for this value to be reloaded to the CCRm buffer register), it is necessary to rewrite TPnCCR0 and then write to the TPnCCR1 register before the 16-bit counter value and the TPnCCR0 register value match. Thereafter, the values of the TPnCCR0 and the TPnCCR1 register are reloaded upon TPnCCR0 register match.

Whether to enable or disable the next reload timing is controlled by writing to the TPnCCR1 register. Thus even when wishing only to rewrite the value of the TPnCCR0 register, also write the same value to the TPnCCR1 register.

Caution Writing to the TPnCCR1 register includes enabling of reload. Thus, rewrite the TPnCCR1 register after rewriting the TPnCCR0 register.

The following flowchart illustrates an example of the operation in the PWM mode.

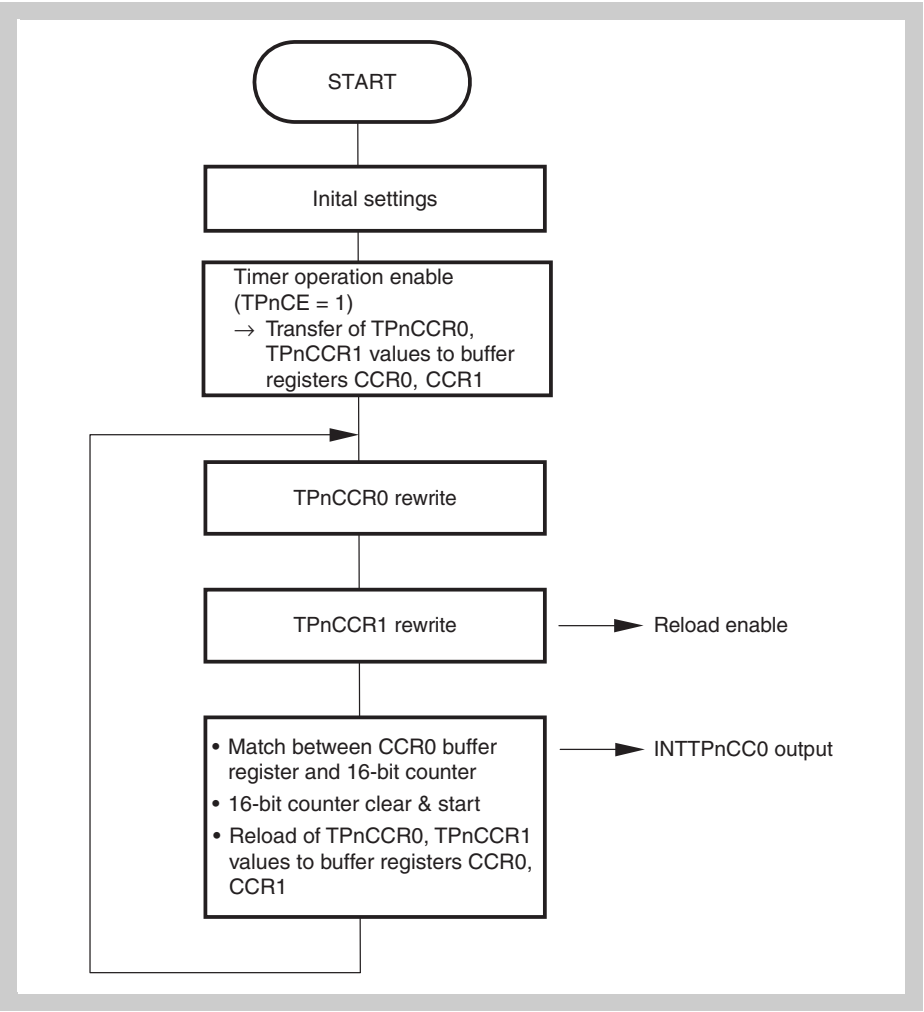


Figure 9-4 Basic operation flow for reload (batch rewrite)

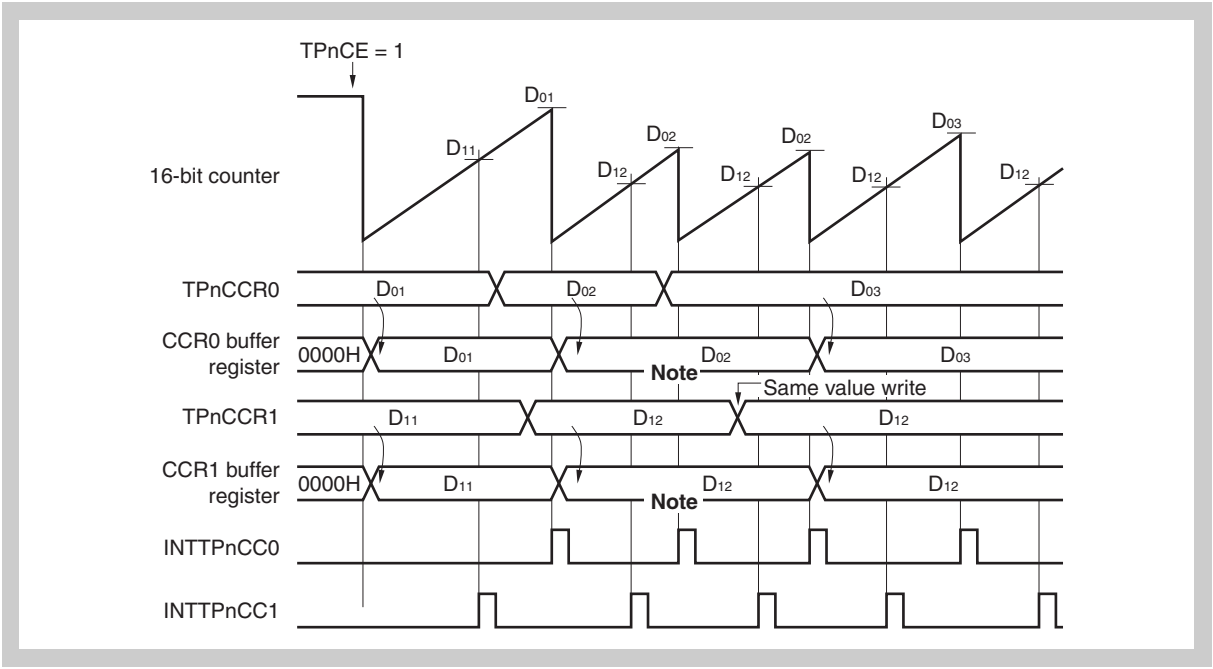


Figure 9-5 Timing diagram of reload

Note Reload is not performed because the TPnCCR1 register was not rewritten.

Remark D_{01}, D_{02}, D_{03}: Setting value of TPnCCR0 register (0000_H to $FFFF_H$)
D_{11}, D_{12}: Setting value of TPnCCR1 register (0000_H to $FFFF_H$)

9.5.2 Interval timer mode (TPnMD[2:0] = 000B)

In the interval timer mode, an interrupt request signal (INTTPnCC0) is output upon a match between the setting value of the TPnCCR0 register and the value of the 16-bit counter, and the 16-bit counter is cleared. The TPnCCR0 register can be rewritten when TPnCE = 1, and when a value is set to the TPnCCR0 register with a write instruction from the CPU, it is transferred to the CCR0 buffer register through anytime write, and is used as the value for comparison 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 TPnCCR1 register is not performed. However, the setting value of the TPnCCR1 register is transferred to the CCR1 buffer register and compared with the value of the 16-bit counter, and an interrupt request (INTTPnCC1) is output if these values match.

Moreover, TOPnm pin output is also possible by setting the TPnOEm bit to 1.

When the TPnCCR1 register is not used, it is recommended to set $FFFF_H$ as the setting value for the TPnCCR1 register.

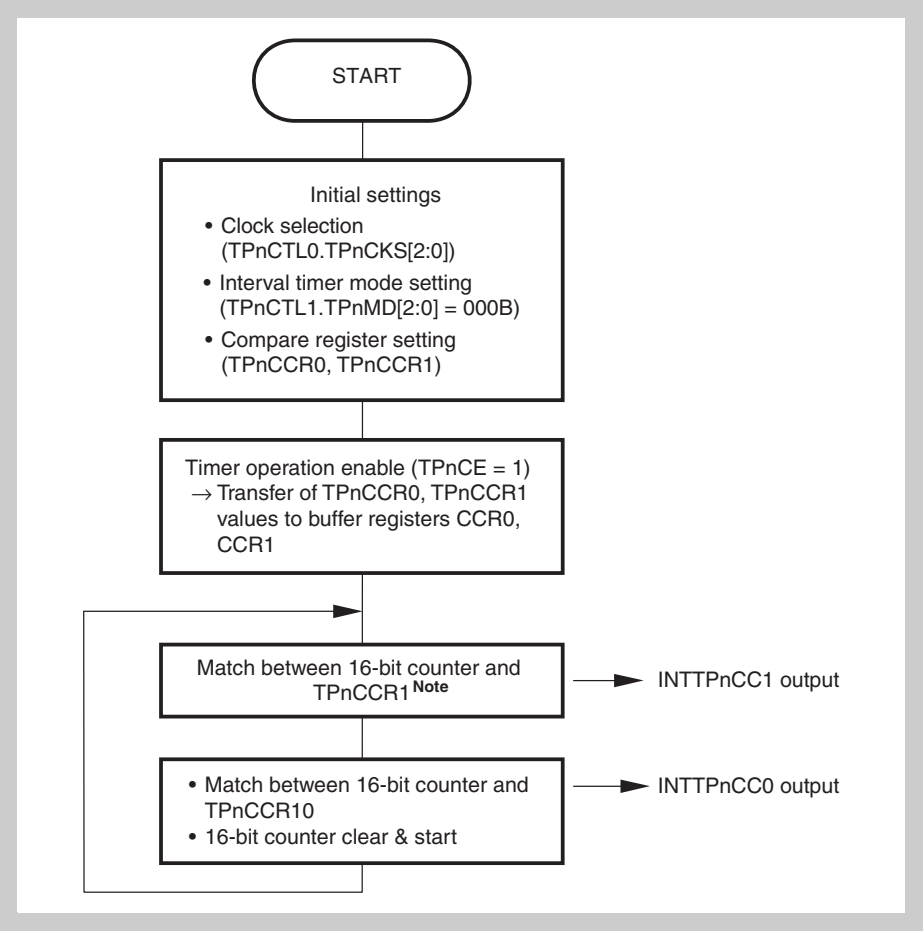


Figure 9-6 Flowchart of basic operation in interval timer mode

Note The 16-bit counter is not cleared upon a match between the 16-bit counter and TPnCCR1.

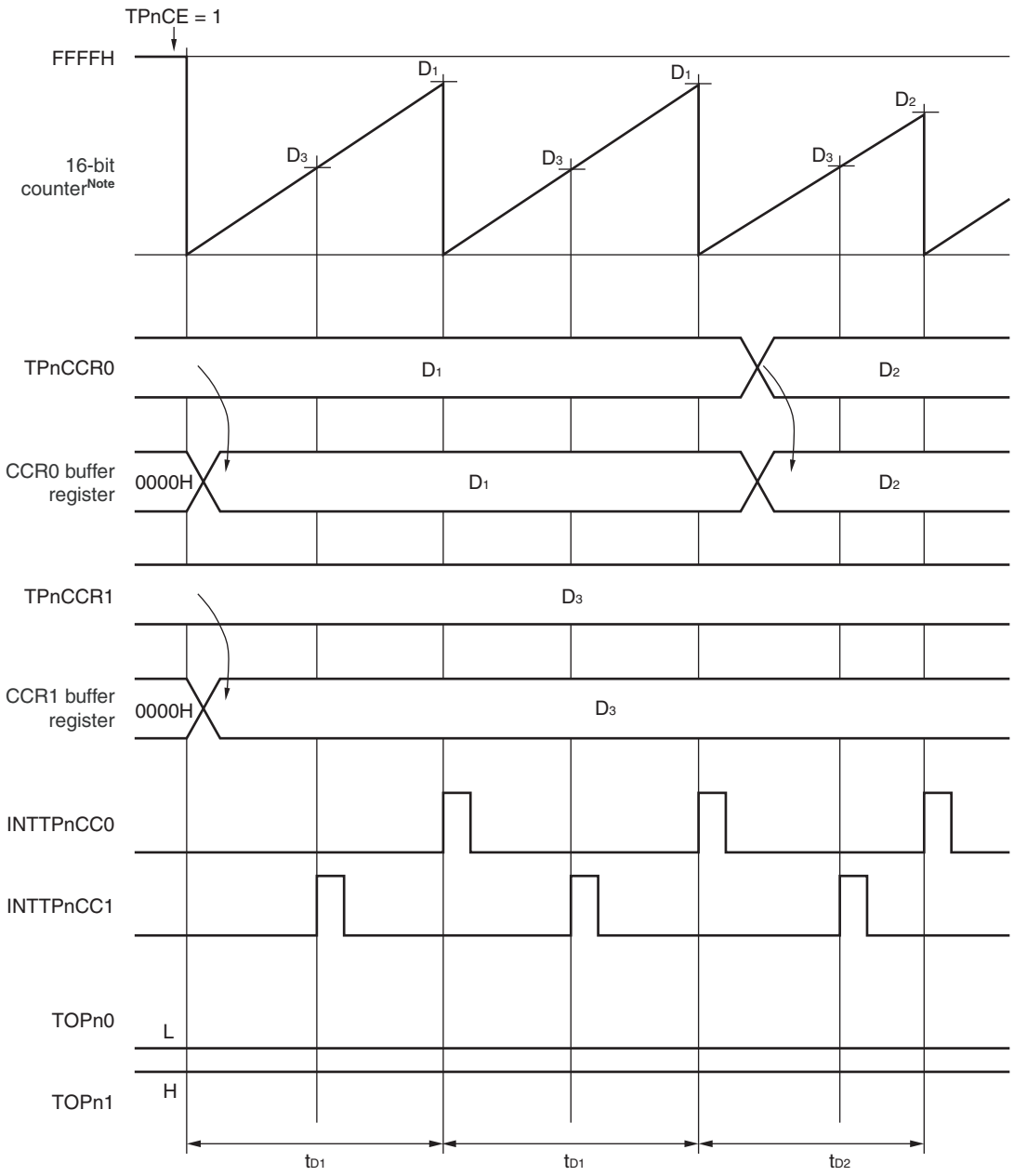


Figure 9-7 Basic operation timing in interval timer mode (1/2)

(a) $D_1 > D_2 > D_3$; rewrite of TPnCCR0 register only; TOPn0, TOPn1 are not outputs (TPnOE0/TPnOE1 = 0, TPnOL0 = 0, TPnOL = 1)

Note The 16-bit counter is not cleared when its value matches the value of TPnCCR1.

Remarks

1. D_1, D_2: Setting values of TPnCCR0 register (0000_H to $FFFF_H$)
   D_3: Setting value of TPnCCR1 register (0000_H to $FFFF_H$)
2. Interval time ($t_{Dn)}$ = (Dn + 1) × (count clock cycle)

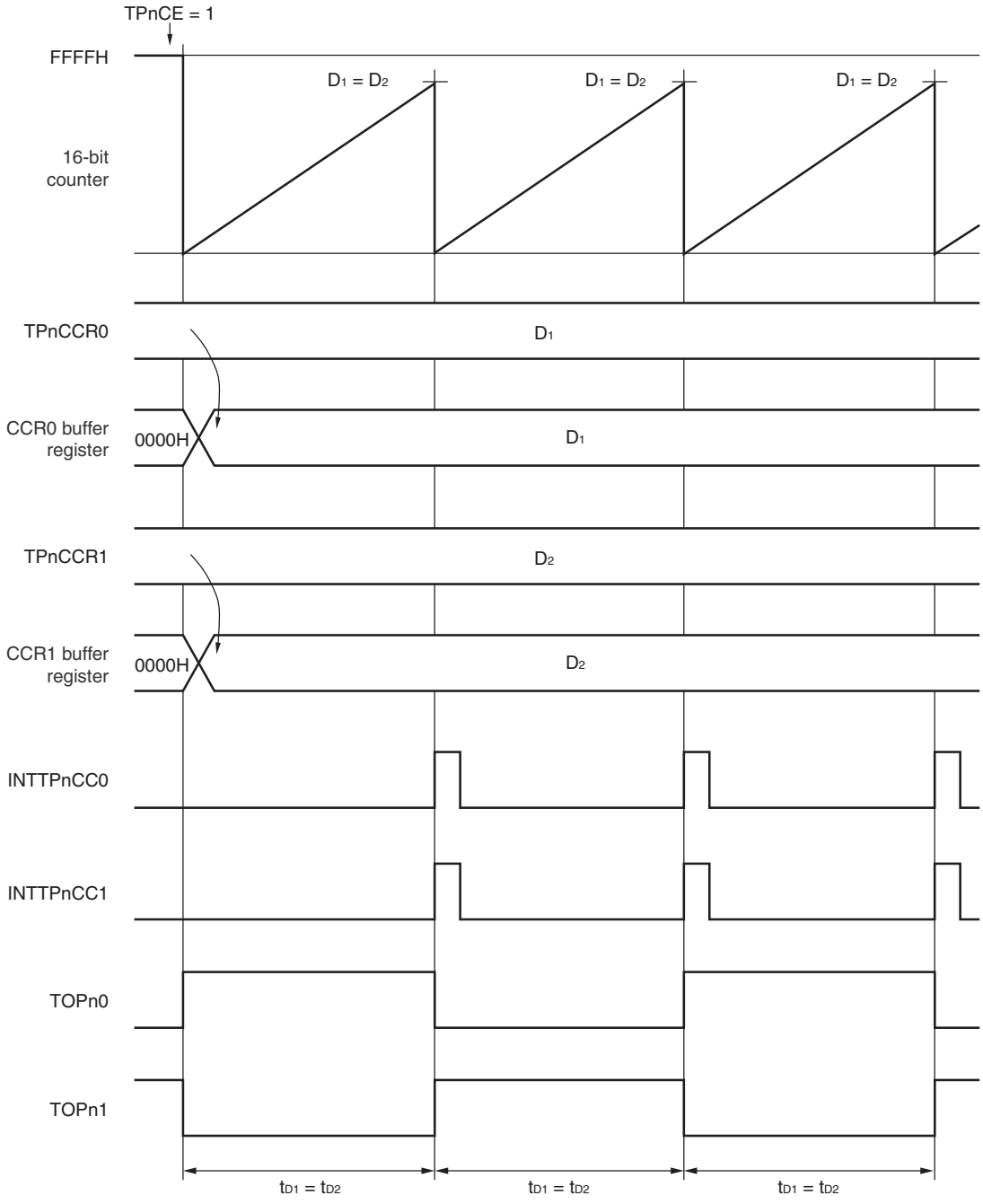


Figure 9-7 Basic operation timing in interval timer mode (2/2)

(b) $D_1 = D_2$; no TPnCCR0, TPnCCR1 rewrite; TOPn0 and TOPn1 are outputs (TPnOE0/TPnOE1 = 1, TPnOL0 = 0, TPnOL1 = 1)

Remarks

1. D_1: Setting value of TPnCCR0 register (0000_H to $FFFF_H$)
   D_2: Setting value of TPnCCR1 register (0000_H to $FFFF_H$)
2. Interval time ($t_{Dn)}$ = (Dn + 1) × (count clock cycle)

9.5.3 External event count mode (TPnMD[2:0] = 001B)

In the external event count mode, external event count input (TEVTPn pin input) is used as a count-up signal. When the external event count mode is set, count-up is performed using external event count input (TEVTPn pin input), regardless of the setting of the TPnEEE bit of the TPnCTL0 register.

In the external event count mode, a match interrupt request (INTTPnCC0) is output upon a match between the setting value of the TPnCCR0 register and the value of the 16-bit counter, and the 16-bit counter is cleared.

When a value is set to the TPnCCR0 register with a write instruction from the CPU, it is transferred to the CCR0 buffer register through anytime write, and is used as the value for comparison with the 16-bit counter value.

In the external event count 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 TPnCCR1 register is not performed. However, the setting value of the TPnCCR1 register is transferred to the CCR1 buffer register and compared with the value of the 16-bit counter, and an interrupt request (INTTPnCC1) is output if these values match.

Moreover, TOPn1 pin output is also possible by setting the TPnOE1 bit to 1.

The TPnCCR0 register can be rewritten when TPnCE = 1. When the TPnCCR1 register is not used, it is recommended to set TPnCCR1 to $FFFF_H$.

Caution

1. In external event count mode, when the content of the TRnCCR0 register is set to m, the number of TEVTPn pin input edge detection times is m+1.
2. In external event count mode, do not set TPnCCR0 register to 0000_H.
3. When the TPnCCR1 register value is set to 0000_H in external event count mode the corresponding interrupt (INTTPnCC1) does not occur immediately after start, but after the first overflow of the timer ($FFFF_H$ to 0000_H).
4. TOPn0 pin output cannot be used in external event count mode. Alternatively use the interval timer mode (refer to section *9.5.2 on page 234*) and set TPnEEE = 1 in conjunction with TOPn0 pin output.

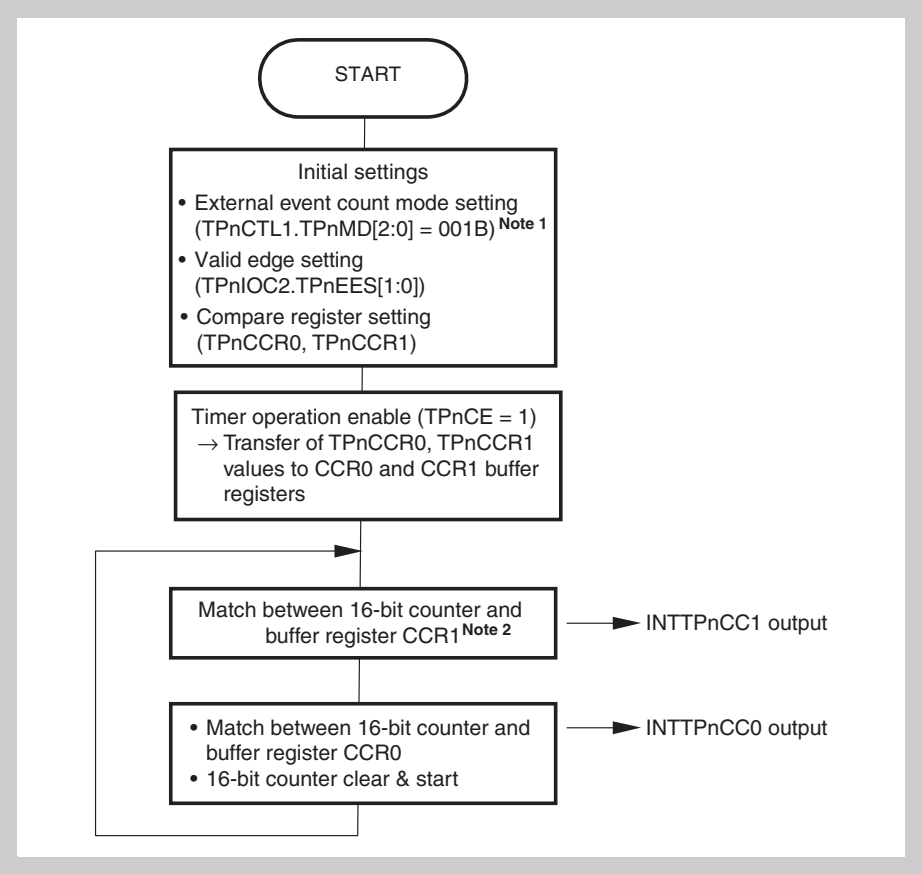


Figure 9-8 Flowchart of basic operation in external event count mode

Note 1. Selection of the TPnEEE 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.

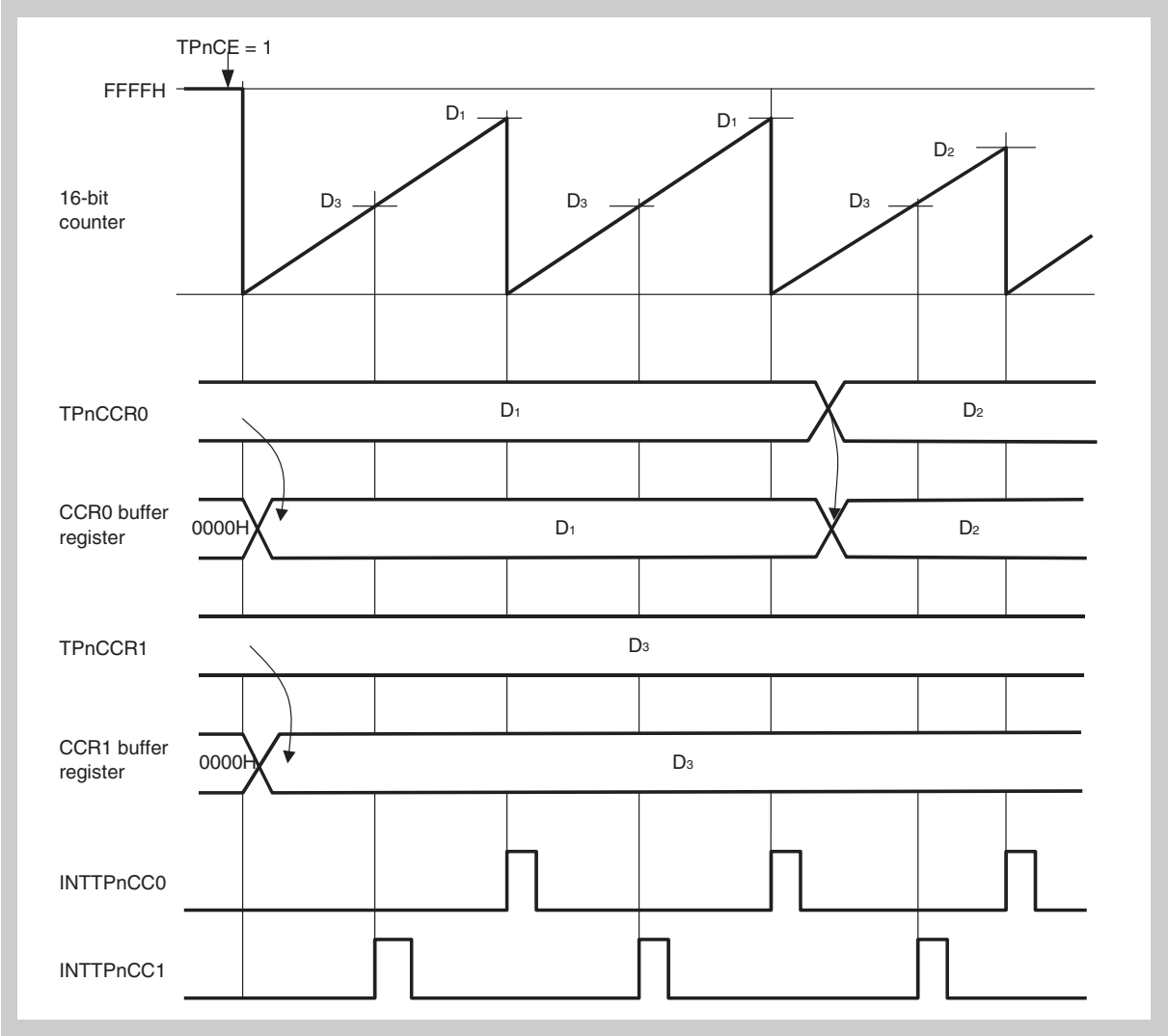


Figure 9-9 Basic operation timing in external event count mode (1/2)

(a) D1 > D2 > D3; rewrite of TPnCCR0 only; no TOPn1 output

Remarks

1. D_1, D_2: Setting values of TPnCCR0 register (0001_H to $FFFF_H$)
   D_3: Setting value of TPnCCR1 register (0000_H to $FFFF_H$)
2. Event count = (Dn + 1)

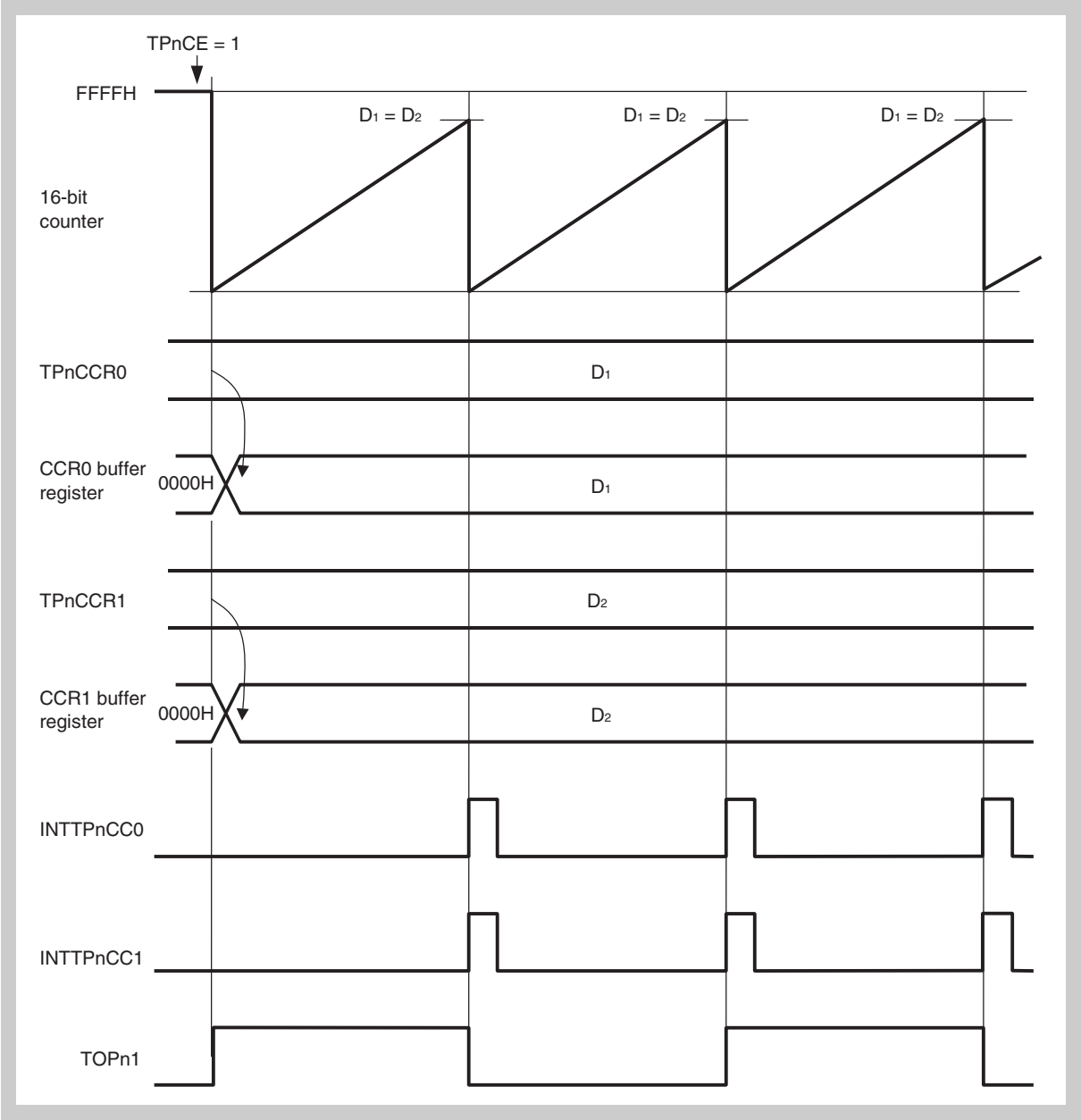


Figure 9-9 Basic operation timing in external event count mode (2/2)

(b) D1 = D2; no TPnCCR0, TPnCCR1 rewrite; TOPn1 output

Remarks

1. D1: Setting value of TPnCCR0 register (0001_H to $FFFF_H$)
   D2: Setting value of TPnCCR1 register (0000_H to $FFFF_H$)
2. Event count = (Dn + 1)

9.5.4 External trigger pulse output mode (TPnMD[2:0] = 010B)

In the external trigger pulse output mode, setting TPnCE = 1 causes external trigger input (TTRGPn pin input) wait with the 16-bit counter stopped at $FFFF_H$. The count-up operation starts upon detection of the external trigger input (TTRGPn pin input) edge.

Regarding TOPn1 output control, the reload register (TPnCCR1) is used as the duty setting register and the compare register (TPnCCR0) is used as the cycle setting register.

The TPnCCR0 register and the TPnCCR1 register can be rewritten when TPnCE = 1.

In order for the setting value when the TPnCCR0 register and the TPnCCR1 register are rewritten to become the 16-bit counter comparison value (in other words, in order for this value to be reloaded to the CCRm buffer register), it is necessary to rewrite TPnCCR0 and then write to the TPnCCR1 register before the 16-bit counter value and the TPnCCR0 register value match. Thereafter, the values of the TPnCCR0 and the TPnCCR1 register are reloaded upon a TPnCCR0 register match.

Whether to enable or disable the next reload timing is controlled by writing to the TPnCCR1 register. Thus even when wishing only to rewrite the value of the TPnCCR0 register, also write the same value to the TPnCCR1 register.

Reload is disabled even when only the TPnCCR0 register is rewritten. To stop timer P, set TPnCE = 0. If the external trigger (TTRGPn pin input) edge is detected several times in the external trigger pulse mode, the 16-bit counter is cleared at the edge detection timing and count-up starts.

To realize the same function (software trigger pulse mode) as external trigger pulse mode using a software trigger instead of external trigger input (TTRGPn pin input), set the TPnCTL1.TPnEST bit to 1 so that the software trigger is output. The external trigger pulse waveform is output from TOPn1. The TOPn0 pin performs toggle output upon a match between the TPnCCR0 register and the 16-bit counter.

Since the TPnCCR0 register and the TPnCCR1 register have their function fixed to that of a compare register in the external trigger pulse mode, they cannot be used for capture operation in this mode.

Caution In the external trigger pulse output mode, the external event clock input (TEVTPn) is prohibited (TPnCTL1.TPnEEE = 0).

Note For the reload operation when TPnCCR0 and TPnCCR1 are rewritten during timer operation, refer to *"PWM mode (TPnMD[2:0] = 100B)" on page 247*.

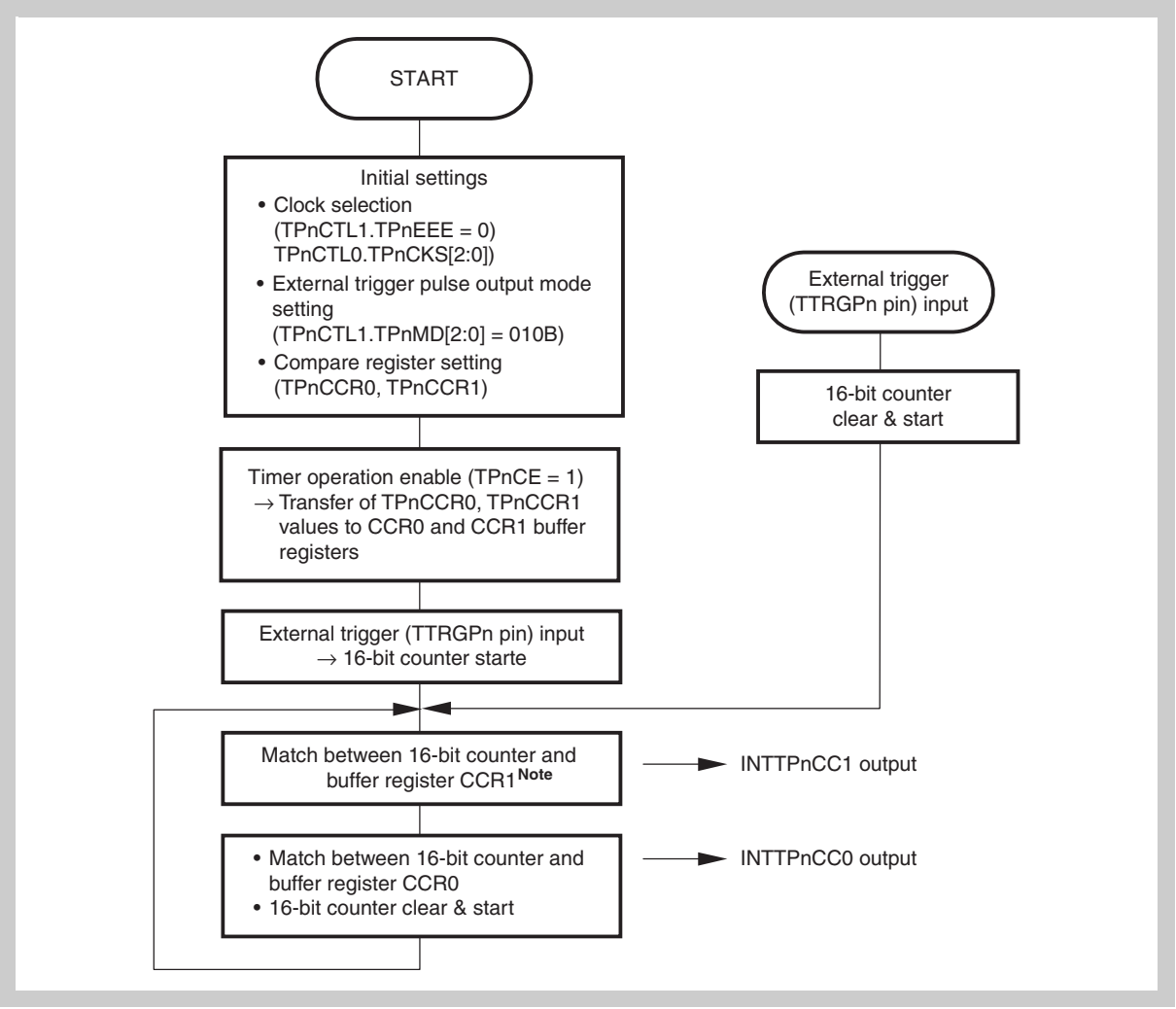


Figure 9-10 Flowchart of basic operation in external trigger pulse output mode

Note The 16-bit counter is not cleared upon a match between the 16-bit counter and the CCR1 buffer register.

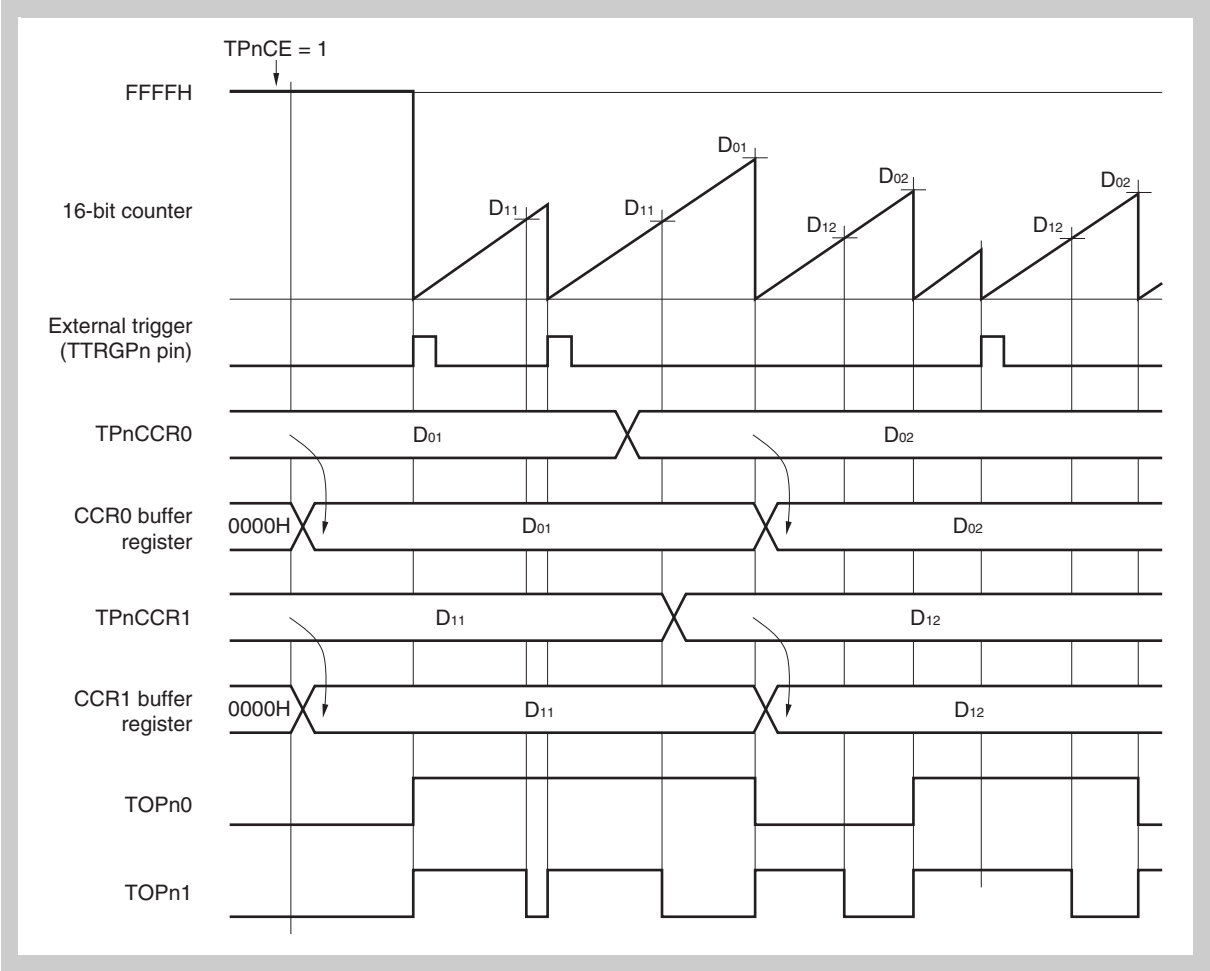


Figure 9-11 Basic operation timing in external trigger pulse output mode

Remarks

1. D01, D02: Setting value of TPnCCR0 register (0000_H to $FFFF_H$)
   D11, D12: Setting value of TPnCCR1 register (0000_H to $FFFF_H$)

2. TOPn1 output duty = (Setting value of TPnCCR1 register) / (Setting value of TP0CCR0 register)
   TOPn1 output cycle = (Setting value of TPnCCR0 register) × (Count clock cycle)

9.5.5 One-shot pulse mode (TPnMD[2:0] = 011B)

In the one-shot pulse mode, setting TPnCE = 1 causes waiting on TPnEST bit setting (1) or TTRGPn pin edge detection trigger with the 16-bit counter held at $FFFF_H$. The 16-bit counter starts
counting up upon trigger input, and upon a match between the value of the 16-bit counter and the value of the CCR1 buffer register transferred from the TPnCR1 register, TOPn1 becomes high level; Upon a match between the value of the 16-bit counter and the value of the CCR0 register transferred from the TPnCCR0 register, TOPn1 becomes low level and the 16-bit counter is cleared to 0000_H and stops. Any trigger input past the first one during 16-bit counter operation is ignored. Be sure to input the
second and subsequent triggers when the 16-bit counter has stopped at 0000_H. In the one-shot pulse mode, the TPnCCR0 and TPnCCR1 registers can be rewritten when TPnCE = 1. The setting values rewritten to the TPnCCR0 and TPnCCR1 registers become valid following execution of a write instruction from the CPU, at which time they are transferred to the CCR0 buffer register and the CCR0 buffer register through anytime write, and become the values for comparison with the 16-bit counter value. The one-shot pulse waveform is output from the TOPn1 pin. The TOPn0 pin performs toggle output upon a match between the 16-bit counter and the TPnCCR0 register.

Since the TPnCCR0 and TPnCCR1 registers have their function fixed to that of a compare register in the one-shot pulse mode, they cannot be used for capture operation in this mode.

Caution In the one-shot pulse mode, the external event clock input (TEVTPn) is prohibited (TPnCTL1.TPnEEE = 0).

Note External trigger input pin (TTRGP8) and output pin (TOP80) are not available for TMP8.

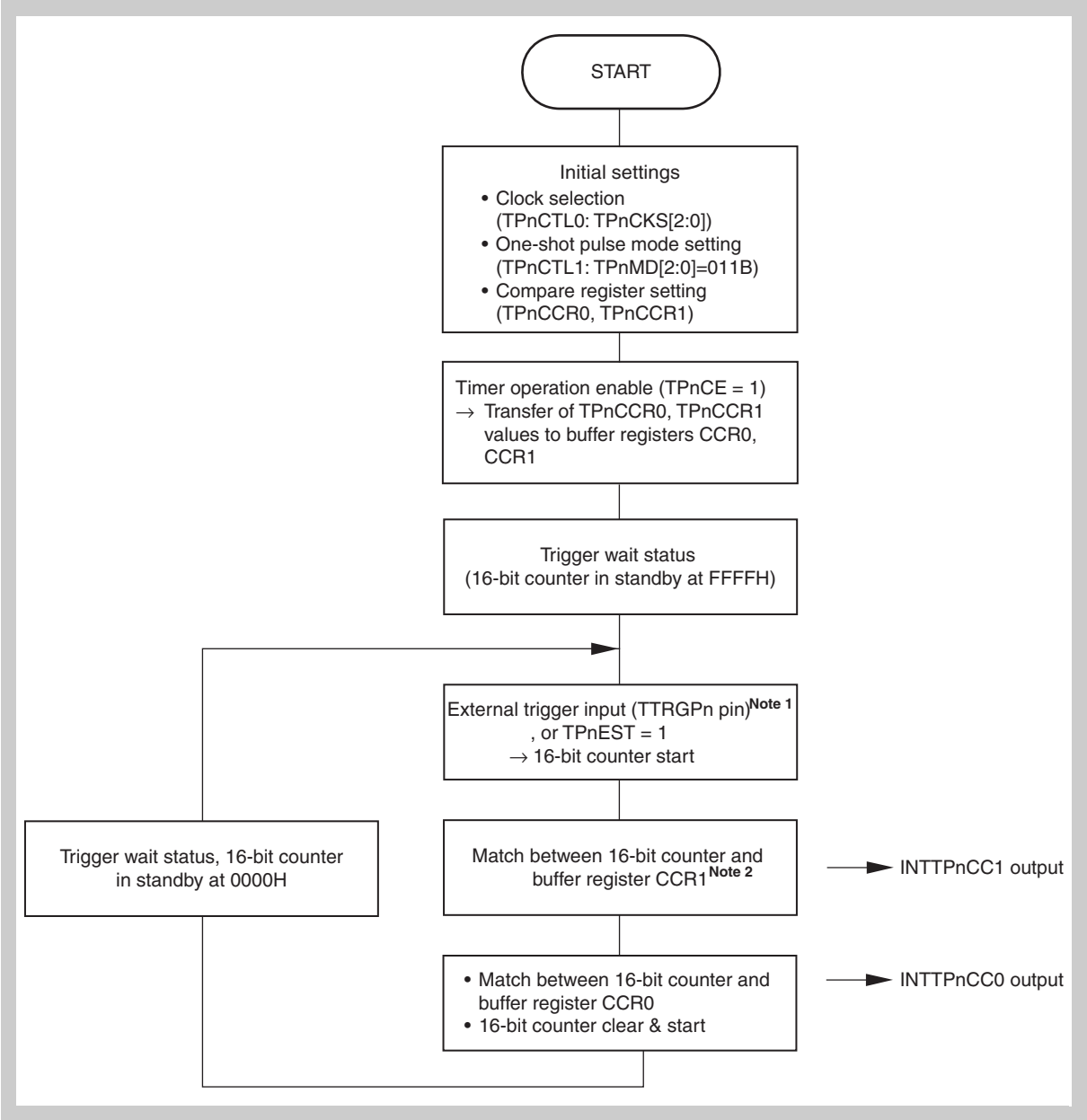


Figure 9-12 Flowchart of basic operation in one-shot pulse mode

Note
1. External trigger input (TTRGPn) is not available for TMP8.
2. 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 and trigger input is ignored even if trigger input is performed during the count-up operation of the 16-bit counter.

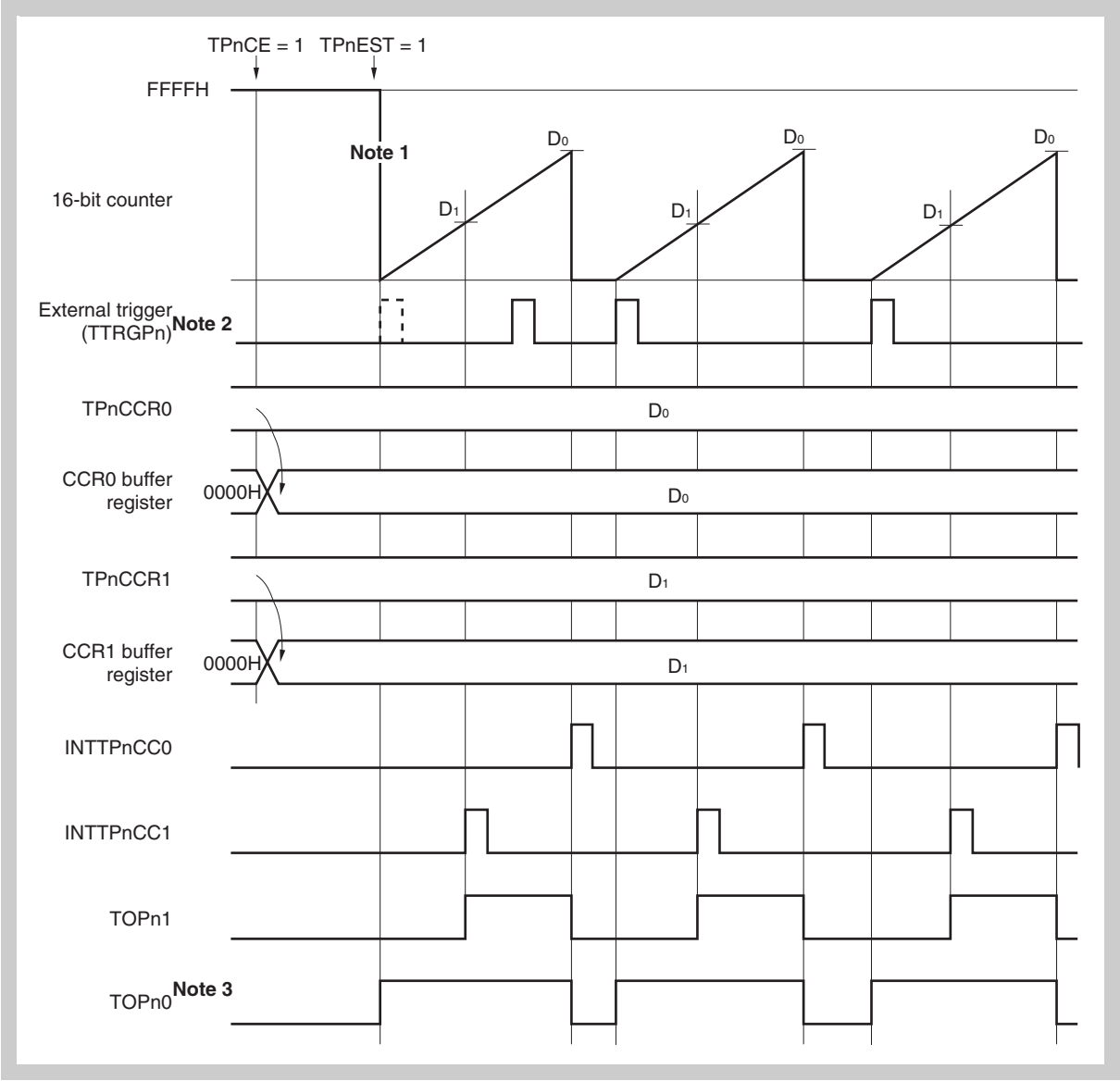


Figure 9-13 Timing of basic operation in one-shot pulse mode

Note

1. The 16-bit counter starts counting up when either TPnEST = 1 is set or TEVTPn is input.
2. External trigger input pin (TTRGPn) is not available for TMP8.
3. Output pin (TOPn0) is not available for TMP8.

Remarks

1. D0: Setting value of TPnCCR0 register (0000_H to $FFFF_H$)
   D1: Setting value of TPnCCR1 register (0000_H to $FFFF_H$)
2. Delay time of one-shot pulse output (TOPn1) when external pin edge detection trigger is used:
   (TPnCCR1 value + 1) (Selected count clock) + $2/(f_{XX})$ + (TTRGPn input filter delay)

9.5.6 PWM mode (TPnMD[2:0] = 100B)

In the PWM mode, TMPn capture/compare register 1 (TPnCCR1) is used as the duty setting register and TMPn capture/compare register 0 (TPnCCR0) is used as the cycle setting register.

Variable duty PWM is output by setting these two registers and operating the timer.

The TPnCCR0 register and the TPnCCR1 register can be rewritten when TPnCE = 1.

In order for the setting value when the TPnCCR0 register and the TPnCCR1 register are rewritten to become the 16-bit counter comparison value (in other words, in order for this value to be reloaded to CCR0 buffer register or CCR1 buffer register), it is necessary to rewrite TPnCCR0 and then write to the TPnCCR1 register before the 16-bit counter value and the TPnCCR0 register value match. Thereafter, the values of the TPnCCR0 register and the TPnCCR1 register are reloaded upon a TPnCCR0 register match.

Whether to enable or disable the next reload timing is controlled by writing to the TPnCCR1 register. Thus even when wishing only to rewrite the value of the TPnCCR0 register, also write the same value to the TPnCCR1 register.

Reload is disabled even when only the TPnCCR0 register is rewritten. To stop timer P, set TPnCE = 0. PWM waveform output is performed from the TOPn1 pin. The TOPn0 pin[Note] performs toggle output upon a match between the 16-bit counter and the TPnCCR0 register.

Since the TPnCCR0 and TPnCCR1 registers have their function fixed that of a compare register in the PWM mode, they cannot be used for capture operation in this mode.

Note TOPn0 output pin is not available for TMP8.

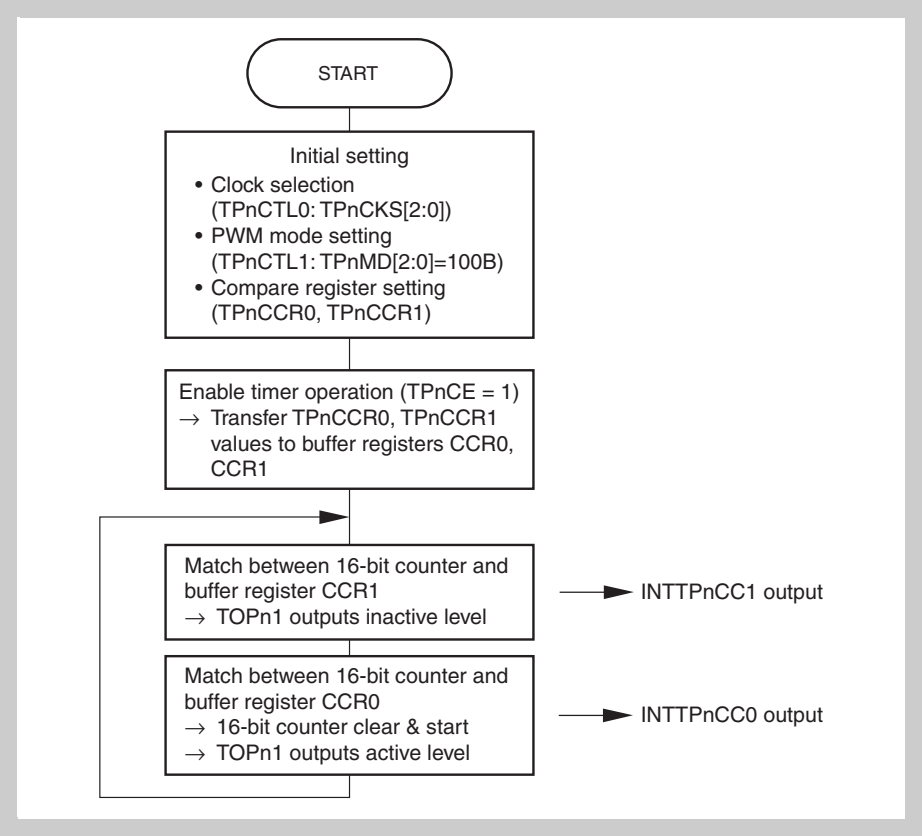


Figure 9-14 Flowchart of basic operation in PWM mode (1/2)

(a) Values of TPnCCR0, TPnCCR1 registers not rewritten during timer operation

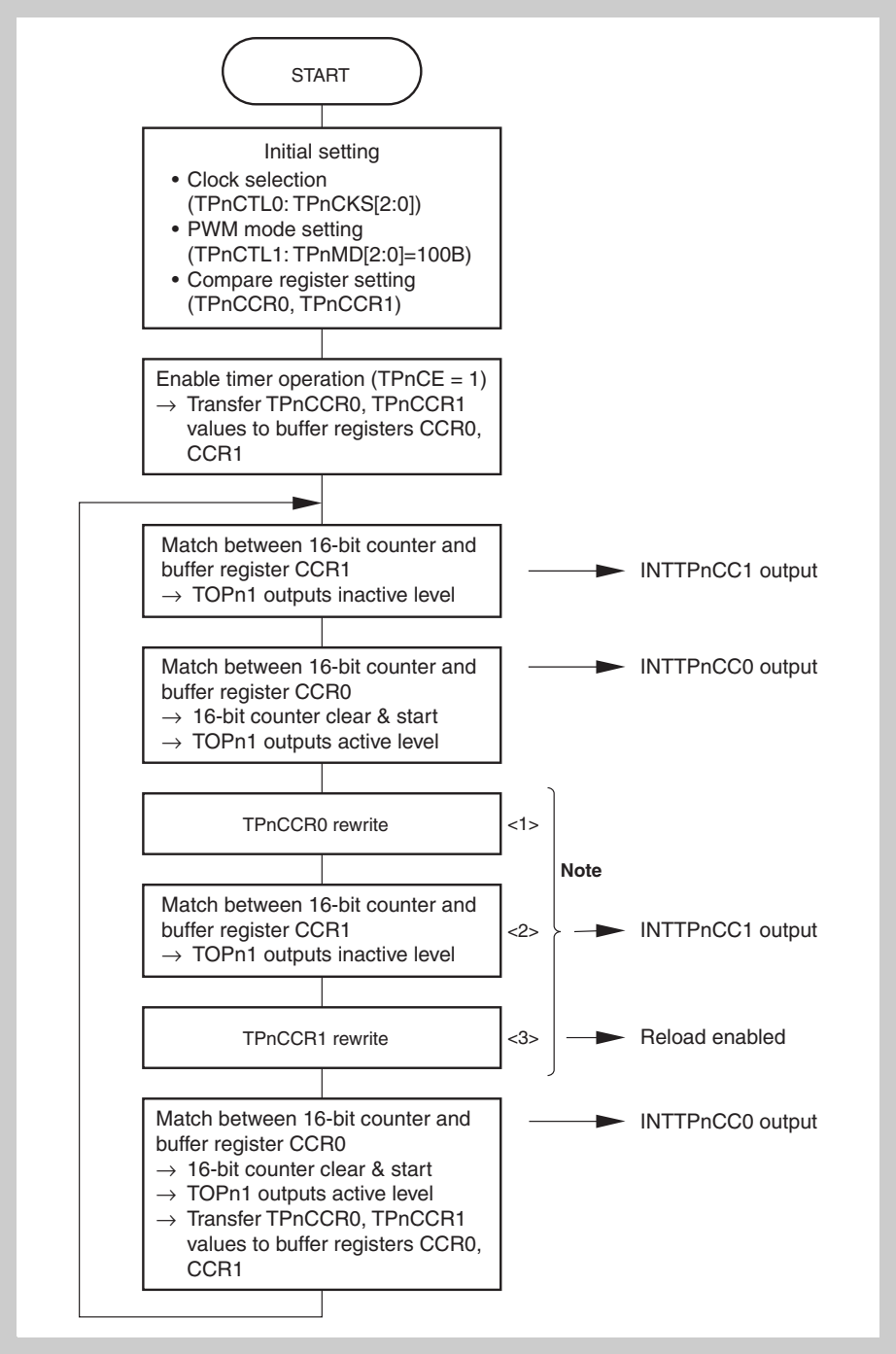


Figure 9-14 Flowchart of basic operation in PWM mode (2/2)

(b) Values of TPnCCR0, TPnCCR1 registers rewritten during timer operation

Note The timing of <2> in the above flowchart may differ depending on the rewrite timing of steps <1> and <3> and the value of TPnCCR1, but make sure that step <3> comes after step <1>.

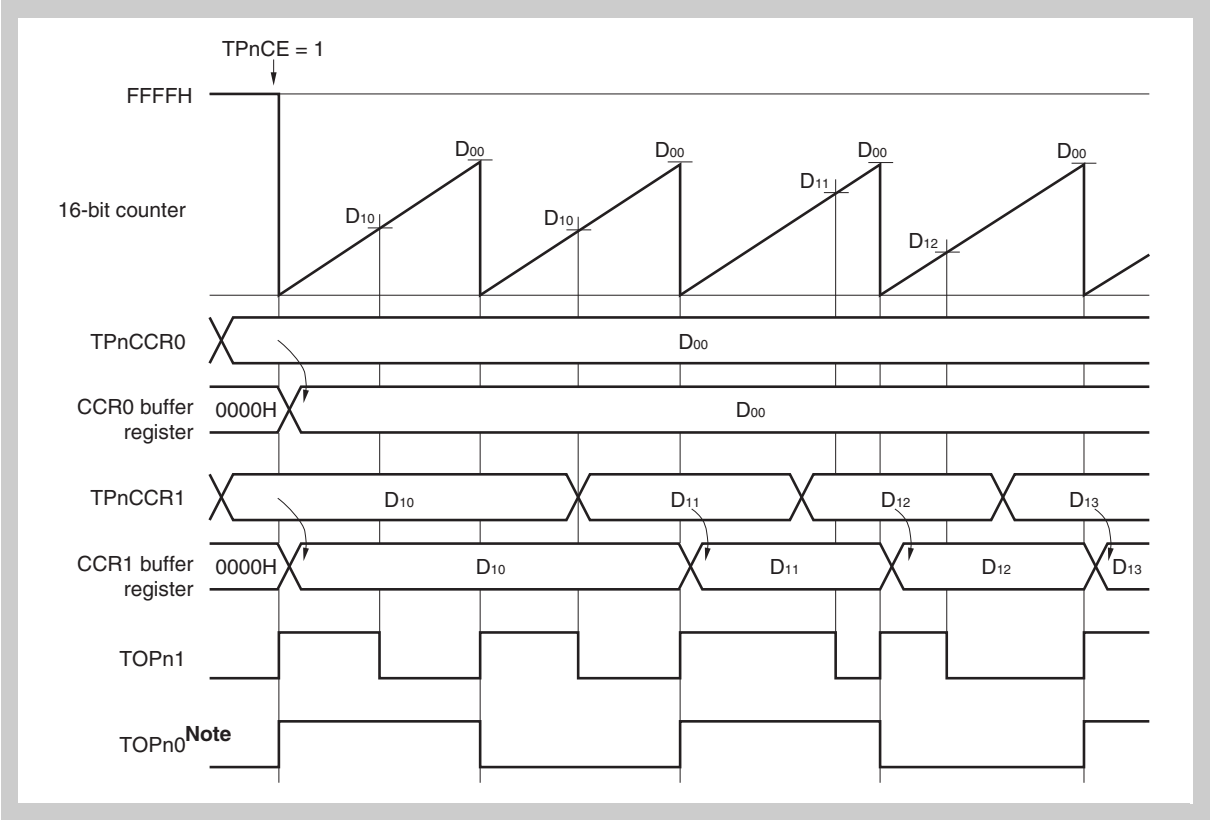


Figure 9-15 Basic operation timing in PWM mode (1/2)

(a) TPnCCR1 value rewritten

Note TOPn0 output pin is not available for TMP8.

Remarks **1.** D_{00}: Setting value of TPnCCR0 register (0000_H to $FFFF_H$)
D_{10}, D_{11}, D_{12}, D_{13}: Setting values of TPnCCR1 register (0000_H to $FFFF_H$)

2. TOPn1 output duty factor = (Setting value of TPnCCR1 register) / (Setting value of TP0CCR0 register + 1)
TOPn1 output cycle = (Setting value of TPnCCR0 register + 1) × (Count clock cycle)
TOPn0 output toggle width = (Setting value of TPnCCR0 register + 1) × (Count clock cycle)

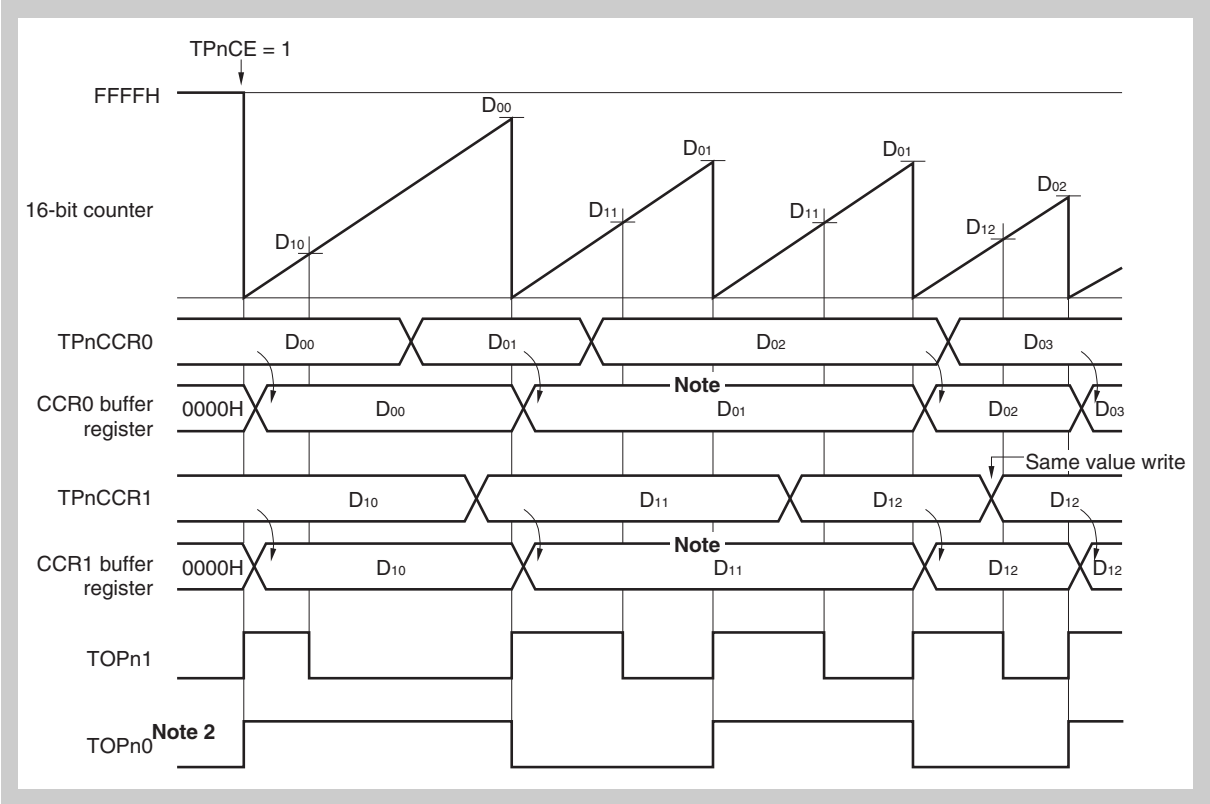


Figure 9-15 Basic operation timing in PWM mode (2/2)

(b) TPnCCR0, TPnCCR1 values rewritten

Note 1. Reload is not performed because the TPnCCR1 register was not rewritten.

2. TOPn0 output pin is not available for TMP8.

Remarks 1. D_{00}, D_{01}, D_{02}, D_{03}: Setting values of TPnCCR0 register (0000_H to $FFFF_H$)
D_{10}, D_{11}, D_{12}, D_{13}: Setting values of TPnCCR1 register (0000_H to $FFFF_H$)

2. TOPn1 output duty factor = (Setting value of TPnCCR1 register) / (Setting value of TP0CCR0 register + 1)
TOPn1 output cycle = (Setting value of TPnCCR0 register + 1) × (Count clock cycle)
TOPn0 output toggle width = (Setting value of TPnCCR0 register + 1) × (Count clock cycle)

9.5.7 Free-running mode (TPnMD[2:0] = 101B)

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 TPnCCS1 and TPnCCS0 bits.

The settings of the TPnCCS1 and TPnCCS0 bits of the TPnOPT0 register are valid only in the free-running mode.

TPnCCS1	Operation
0	Use TPnCCR1 register as compare register
1	Use TPnCCR1 register as capture register

TPnCCS0	Operation
0	Use TPnCCR0 register as compare register
1	Use TPnCCR0 register as capture register

Caution

1. In free-running mode the external event clock input (TEVTPn) is prohibited (TPnCTL1.TPnEEE = 0).
2. When an internal count clock ≤ f_{XX}/16 (TPnCTL0.TPnCKS2-0) is selected in free-running mode, and TPnCCR0 and/or TPnCCR1 are used as capture registers, the a value of $FFFF_H$ will be captured if a valid signal edge is input before the first count up.

(a) Using TPnCCR1 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 anytime write. (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 (TOPn1) has been enabled, TOPn1 performs toggle output upon a match between the 16-bit counter and the CCR1 buffer register.

(b) Using TPnCCR1 register as capture register

The value of the 16-bit counter is saved to the TPnCCR1 register upon TIPn1 pin edge detection.

Note Since TMP8 has no external input pin, the capture function can only be used internally for capturing the interrupt signal INTTT0CC1 of TMT0, or INTCM11 of TMENC1 into the TP8CCR1 register respectively, which is specified by the TPIC2.TPIC22 (refer to *“TMP input control register 2 (TPIC2)” on page 227*).

(c) Using TPnCCR0 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 anytime rewrite.

When timer output (TOPn0) has been enabled, TOPn0 performs toggle output upon a match between the 16-bit counter and the CCR0 buffer register.

Note TOPn0 output pin is not available for TMP8.

(d) Using TPnCCR0 register as capture register

The value of the 16-bit counter is saved to the TPnCCR0 register upon TIPn0 pin edge detection.

Note Since TMP8 has no external input pin, the capture function can only be used internally for capturing the interrupt signal INTTT0CC0 of TMT0, or INTCM10 of TMENC1 into the TP8CCR0 register, respectively, which is specified by the TPIC2.TPIC22 bit (refer to *"TMP input control register 2 (TPIC2)" on page 227*).

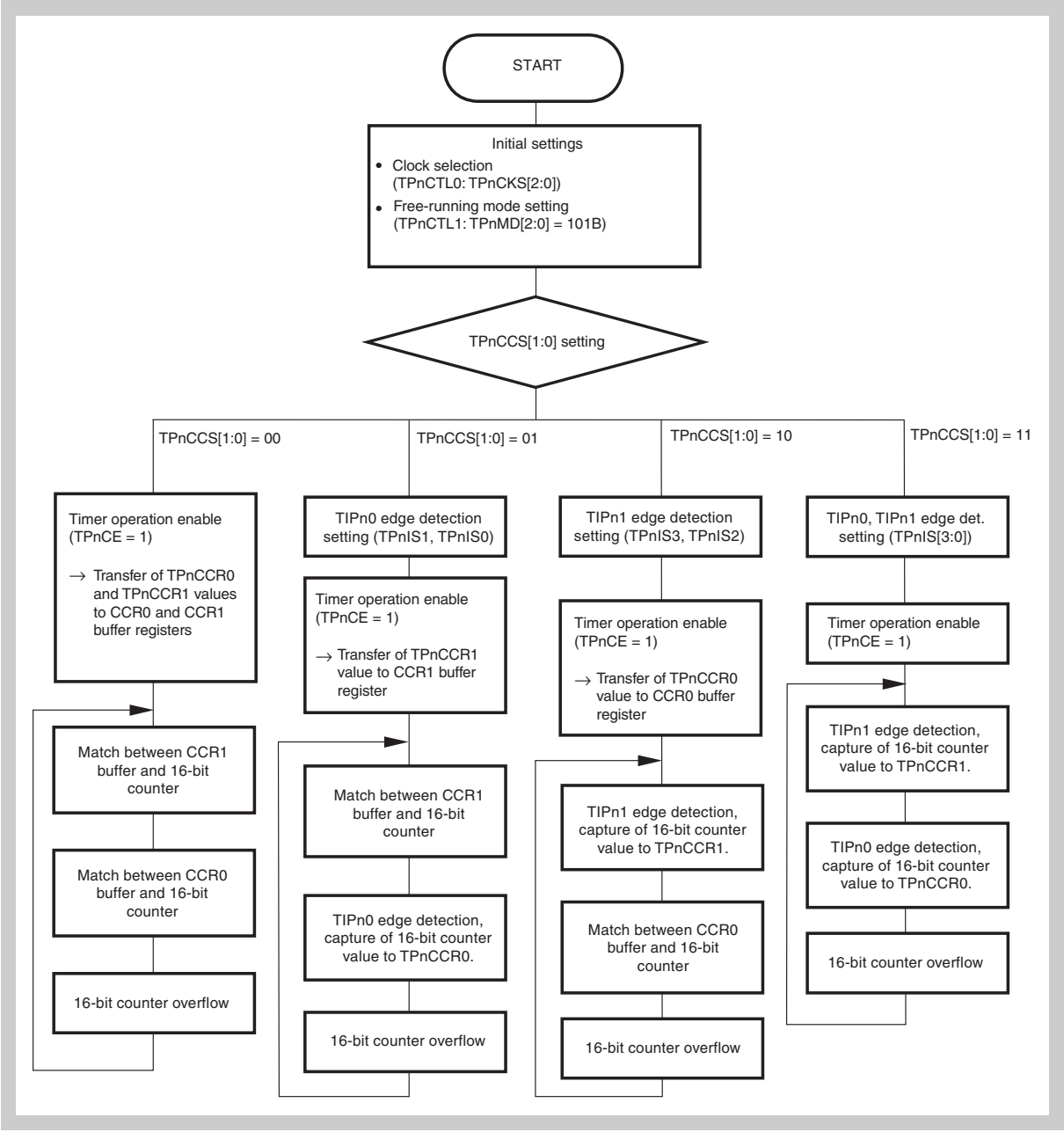


Figure 9-16 Flowchart of basic operation in free-running mode

(1) TPnCCS1 = 0, TPnCCS0 = 0 settings (interval function description)

When TPnCE = 1 is set, the 16-bit counter counts from 0000_H to $FFFF_H$ and the free-running count-up operation continues until TPnCE = 0 is set. In this mode, when a value is written to the TPnCCR0 and TPnCCR1 registers, they are transferred to the CCR0 buffer register and the CCR1 buffer register (anytime write). In this mode, no one-shot pulse is output even when an one-shot pulse trigger is input. Moreover, when TPnOEm = 1 is set, TOPnm performs toggle output upon a match between the 16-bit counter and the CCRm buffer register.

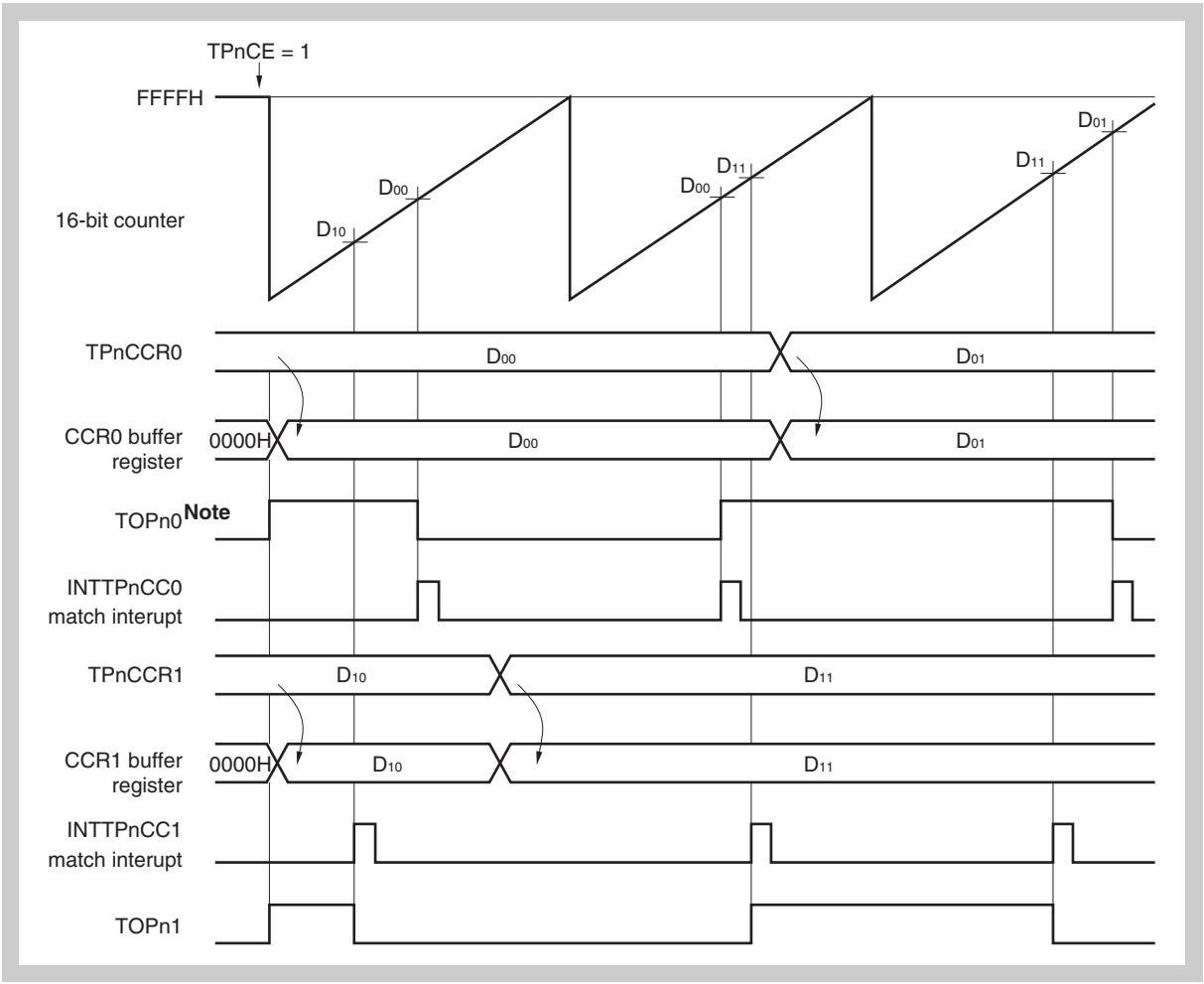


Figure 9-17 Basic operation timing in free-running mode (TPnCCS1 = 0, TPnCCS0 = 0)

Note TOPn0 output pin is not available for TMP8.

Remarks 1. D_{00}, D_{01}: Setting values of TPnCCR0 register (0000_H to $FFFF_H$)
D_{10}, D_{11}: Setting values of TPnCCR1 register (0000_H to $FFFF_H$)

2. TOPnm output rises to the high level when counting is started.

(2) TPnCCS1 = 1, TPnCCS0 = 1 settings (capture function description)

When TPnCE = 1, the 16-bit counter counts from 0000_H to $FFFF_H$ and free-running count-up operation continues until TPnCE = 0 is set. During this time, values are captured by capture trigger operation and are written to the TPnCCR0 and TPnCCR1 registers.

Regarding capture in the vicinity of overflow ($FFFF_H$), judgment is made using the overflow flag (TPnOVF). However, if overflow occurs twice (2 or more free-running cycles), the capture trigger
interval cannot be judged with the TPnOVF flag. In this case, the system should be revised.

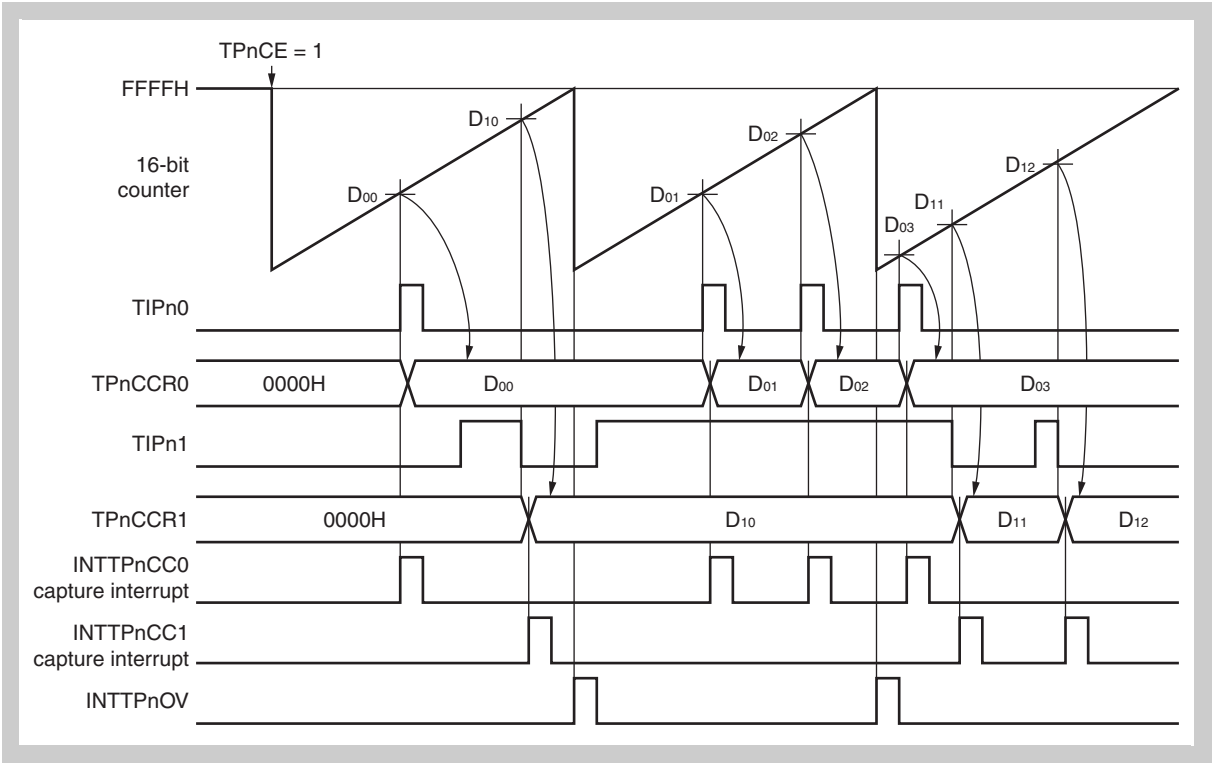


Figure 9-18 Basic operation timing in free-running mode (TPnCCS1 = 1, TPnCCS0 = 1)

Remarks

1. D_{00}, D_{01}: Values captured to TPnCCR0 register (0000_H to $FFFF_H$)
   D_{10}, D_{11}: Values captured to TPnCCR1 register (0000_H to $FFFF_H$)
2. TIPn0: Set to rising edge detection (TPnIS[1:0] = 01B)
   TIPn1: Set to falling edge detection (TPnIS[3:2] = 10B)

(3) TPnCCS1 = 1, TPnCCS0 = 0 settings

When TPnCE = 1 is set, the counter counts from 0000_H to $FFFF_H$ and free-running count-up operation continues until TPnCE = 0 is set. The TPnCCR0 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 TPnCCR0 register as an interval function. Even if TPnOE1 = 1 is set to realize the capture function, the TPnCCR1 register cannot control TOPn1.

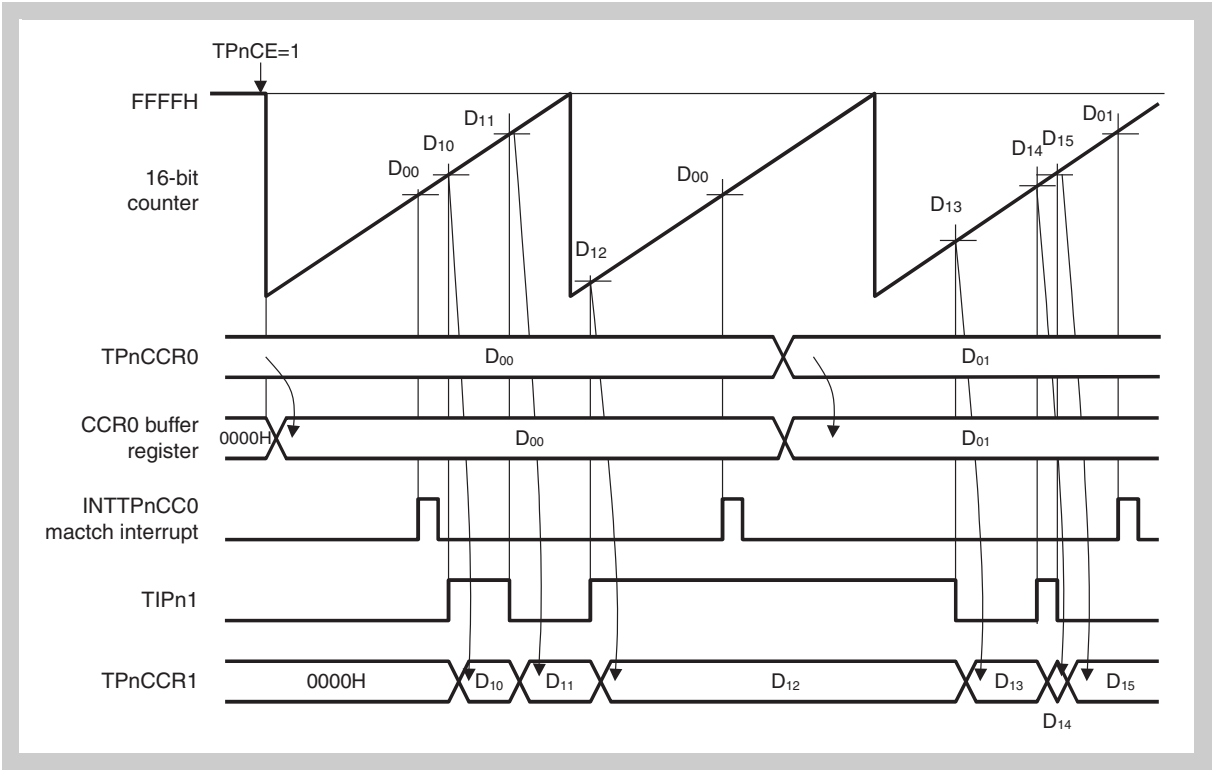


Figure 9-19 Basic operation timing in free-running mode (TPnCCS1 = 1, TPnCCS0 = 0)

Remarks

1. D00, D01: Setting values of TPnCCR0 register (0000_H to $FFFF_H$)
   D10, D11, D12, D13, D14, D15: Values captured to TPnCCR1 register (0000_H to $FFFF_H$)
2. TIPn1: Set to detection of both rising and falling edges (TPnIS[3:2] = 11B)

(4) TPnCCS1 = 0, TPnCCS0 = 1 settings

When TPnCE is set to 1, the 16-bit counter counts from 0000_H to $FFFF_H$ and free-running
count-up operation continues until TPnCE = 0 is set. The TPnCCR1 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 TPnCCR1 register as an interval function. When TPnOE1 = 1 is set, TOPn1 performs toggle output upon mach between the value of the 16-bit counter and the setting value of the TPnCCR1 register.

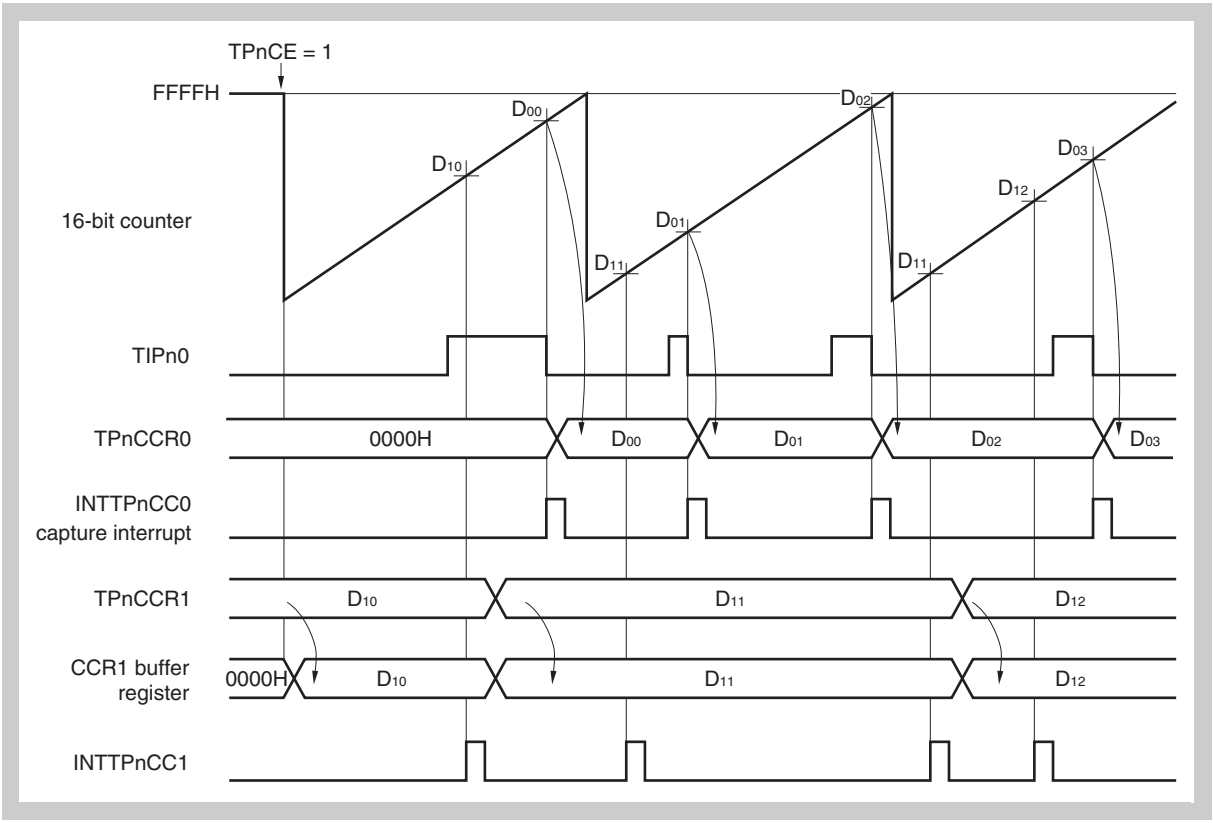


Figure 9-20 Basic operation timing in free-running mode (TPnCCS1 = 0, TPnCCS0 = 1)

Remarks

1. D_{00}, D_{01}, D_{02}, D_{03}: Values captured to TPnCCR0 register (0000_H to $FFFF_H$)
   D_{10}, D_{11}, D_{12}: Setting value of TPnCCR1 register (0000_H to $FFFF_H$)
2. TIPn0: Set to falling edge detection (TPnIS[1:0] = 10B)

(5) Overflow flag

When the counter overflows from $FFFF_H$ to 0000_H in the free-running mode, the overflow flag (TPnOVF) is set to 1 and an overflow interrupt (INTTPnOV) is output.

Be sure to confirm that the overflow flag (TPnOVF) is set to “1” when the overflow interrupt (INTTPnOV) has occurred.

The overflow flag is cleared by writing 0 from the CPU.

9.5.8 Pulse width measurement mode (TPnMD[2:0] = 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 (TPnCCR0), or capture register 1 (TPnCCR1) respectively, and the 16-bit counter is cleared upon edge detection of the TIPn0 pin, or TIPn1 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

1. In the pulse width measurement mode, the external event clock input (TEVTPn) is prohibited (TPnCTL1.TPnEEE = 0).
2. When an internal count clock ≤ f_{XX}/16 (TPnCTL0.TPnCKS2-0) is selected in pulse width measurement mode, and a valid signal edge is input before the first count up, the a value of $FFFF_H$ will be captured in the corresponding TPnCCR0 or TPnCCR1 register.
3. Pulse width measurement cannot be performed by TMP8.

(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 TIPn0 and TIPn1 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 (TPnCCR0 or TPnCCR1), and the timer is cleared and restarts counting.

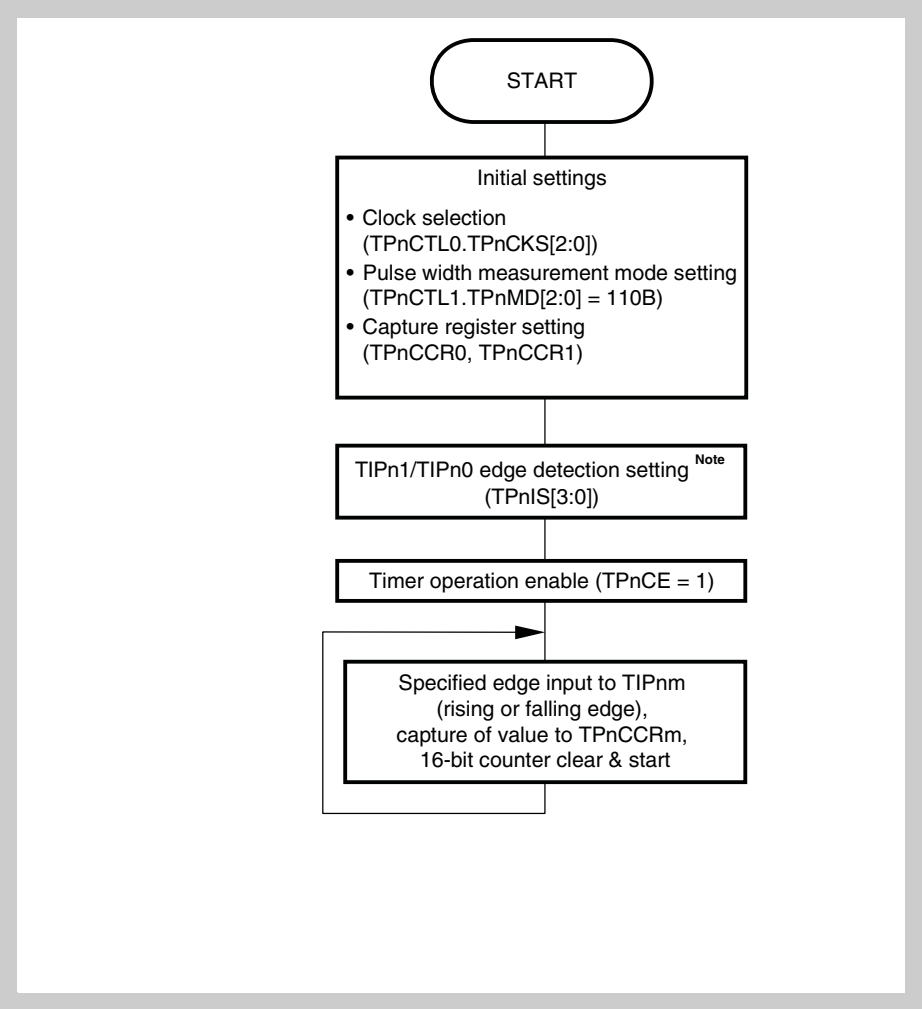


Figure 9-21 Flowchart of pulse period measurement

Note External pulse input is possible for both TIPn0 and TIPn1, 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”.

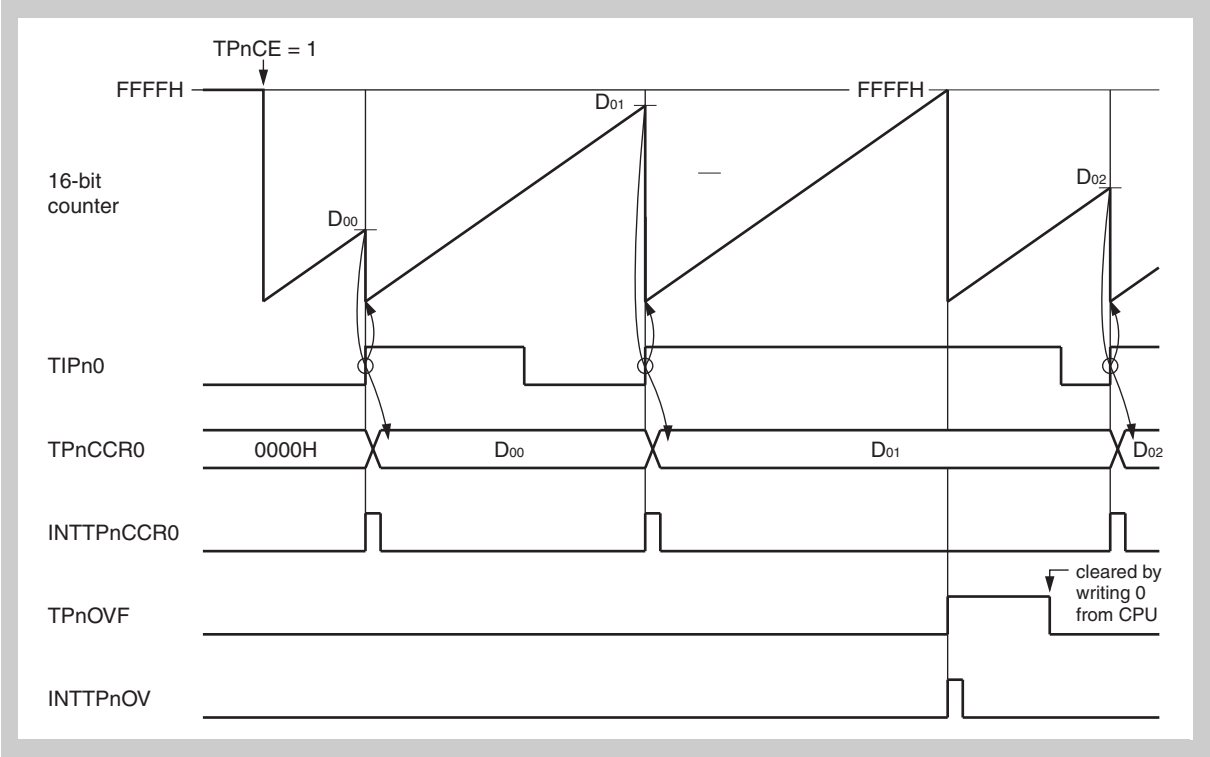


Figure 9-22 Basic operation timing of pulse period measurement

Remarks

1. D_{00}, D_{01}, D_{02}: Values captured to TPnCCR0 register (0000_H to $FFFF_H$)
2. TIPn0: Set to detection of rising edge (TPnIS[1:0] = 01B)
3. TIPn1: Set to no edge detection (TPnIS[3:2] = 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 TIPn0 and TIPn1 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 (TPnCCR0 or TPnCCR1), and the timer is cleared and restarts counting.

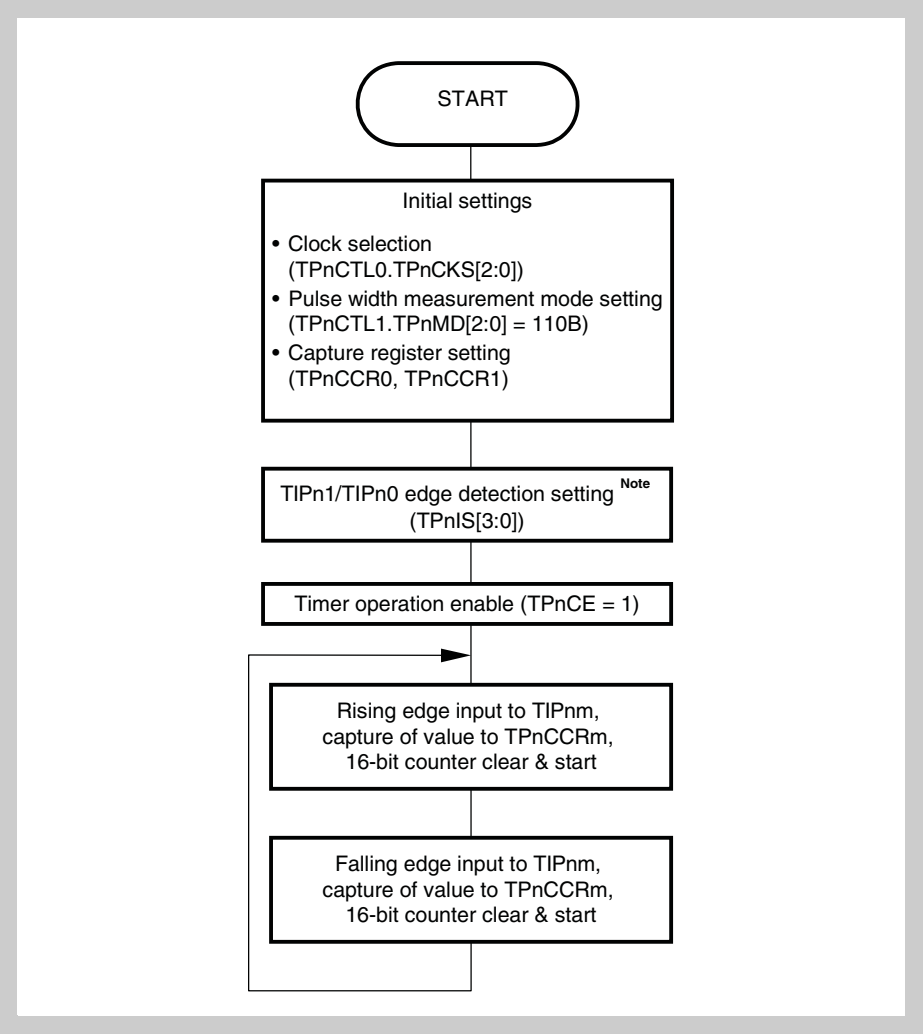


Figure 9-23 Flowchart of alternating pulse width and pulse space measurement

Note External pulse input is possible for both TIPn0 and TIPn1, 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”.

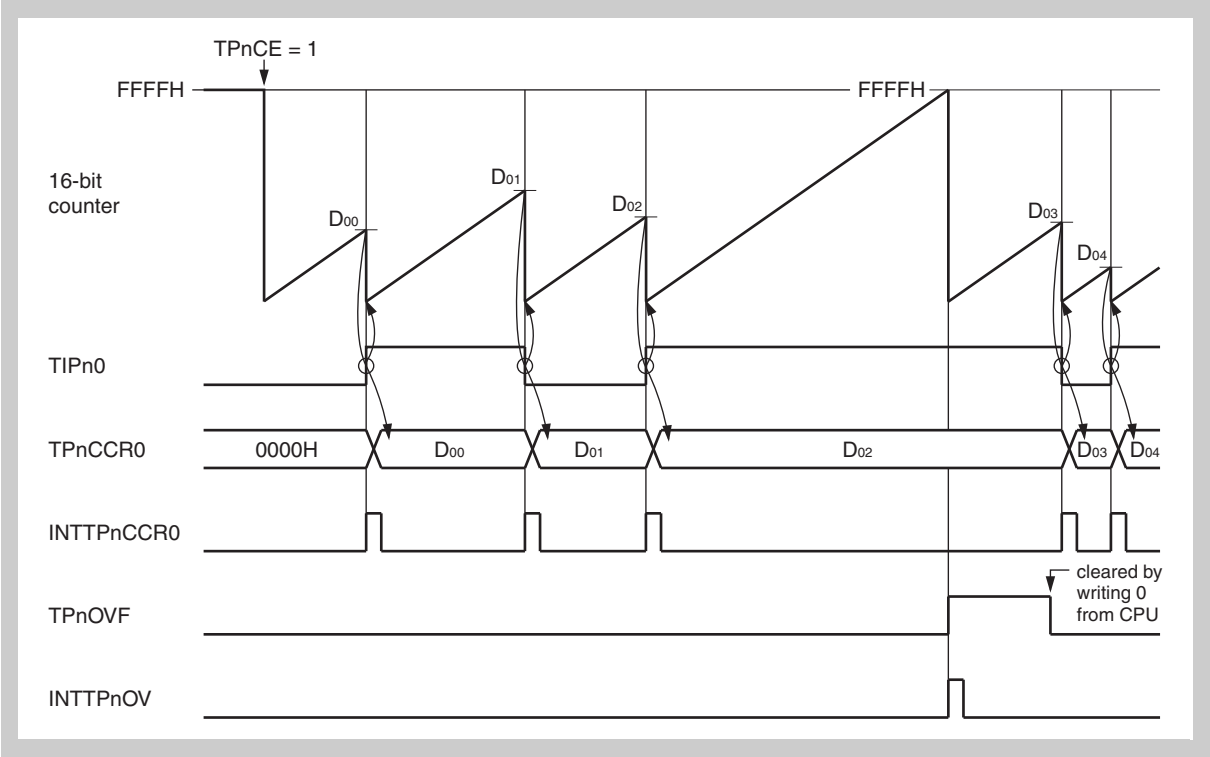


Figure 9-24 Basic operation timing of alternating pulse width and pulse space measurement

Remarks

1. D_{00}, D_{01}, D_{02}, D_{03}, D_{04}: Values captured to TPnCCR0 register (0000_H to $FFFF_H$)
2. TIPn0: Set to detection of both rising and falling edges (TPnIS[1:0] = 11B)
3. TIPn1: Set to no edge detection (TPnIS[3:2] = 00B)

(3) Simultaneous pulse width and pulse space measurement

Pulse width and pulse space can be measure simultaneously in the pulse width measurement mode, when the signal is input to both inputs TIPn0 and TIPn1, where both inputs detect opposite edges. Alternatively the signal can be input to TIPn0 only, when the capture source input selection for capture register 1 is used (refer to *"TMP input control register 0 (TPIC0)" on page 225* and *"TMP input control register 1 (TPIC1)" on page 226*).

By detection of the specified edge the resulting values of pulse width or pulse space are captured in the corresponding capture registers (TPnCCR0, TPnCCR1), and the timer is cleared and restarts counting.

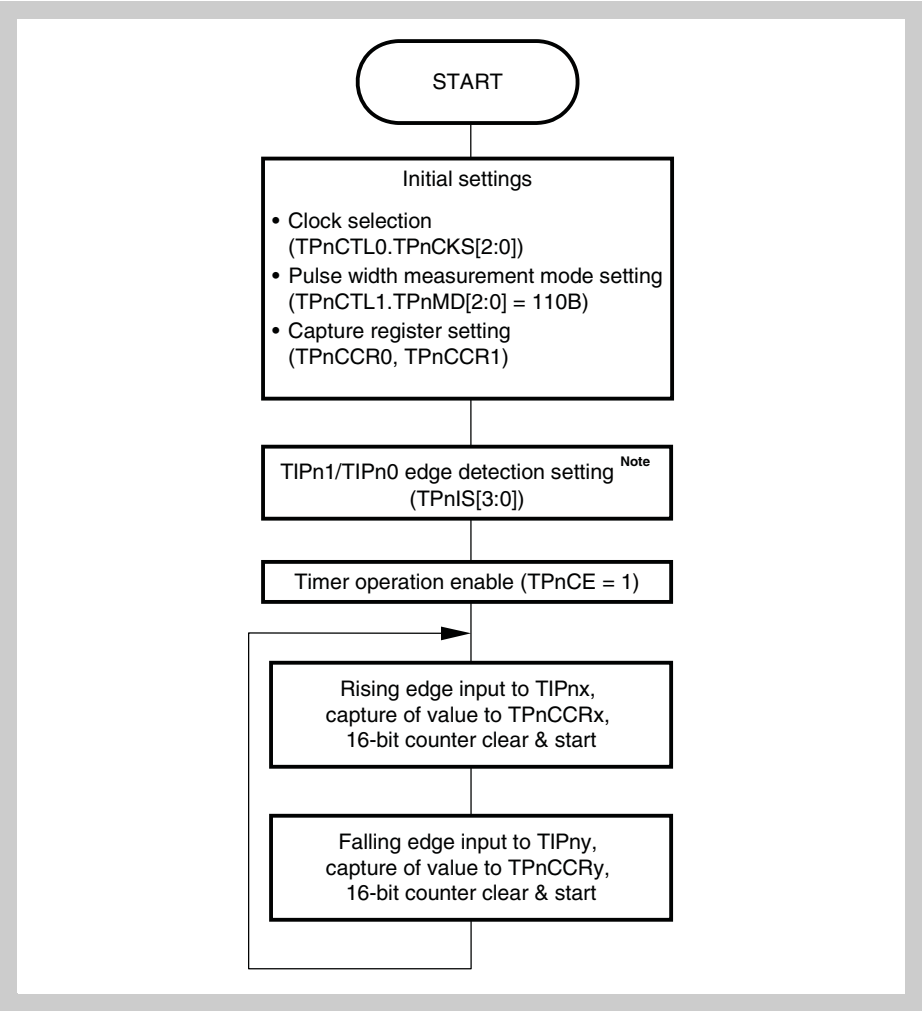


Figure 9-25 Flowchart of simultaneous pulse width and pulse space measurement

Note External pulse input must be input to both TIPn0 and TIPn1, or to TIPn0 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.

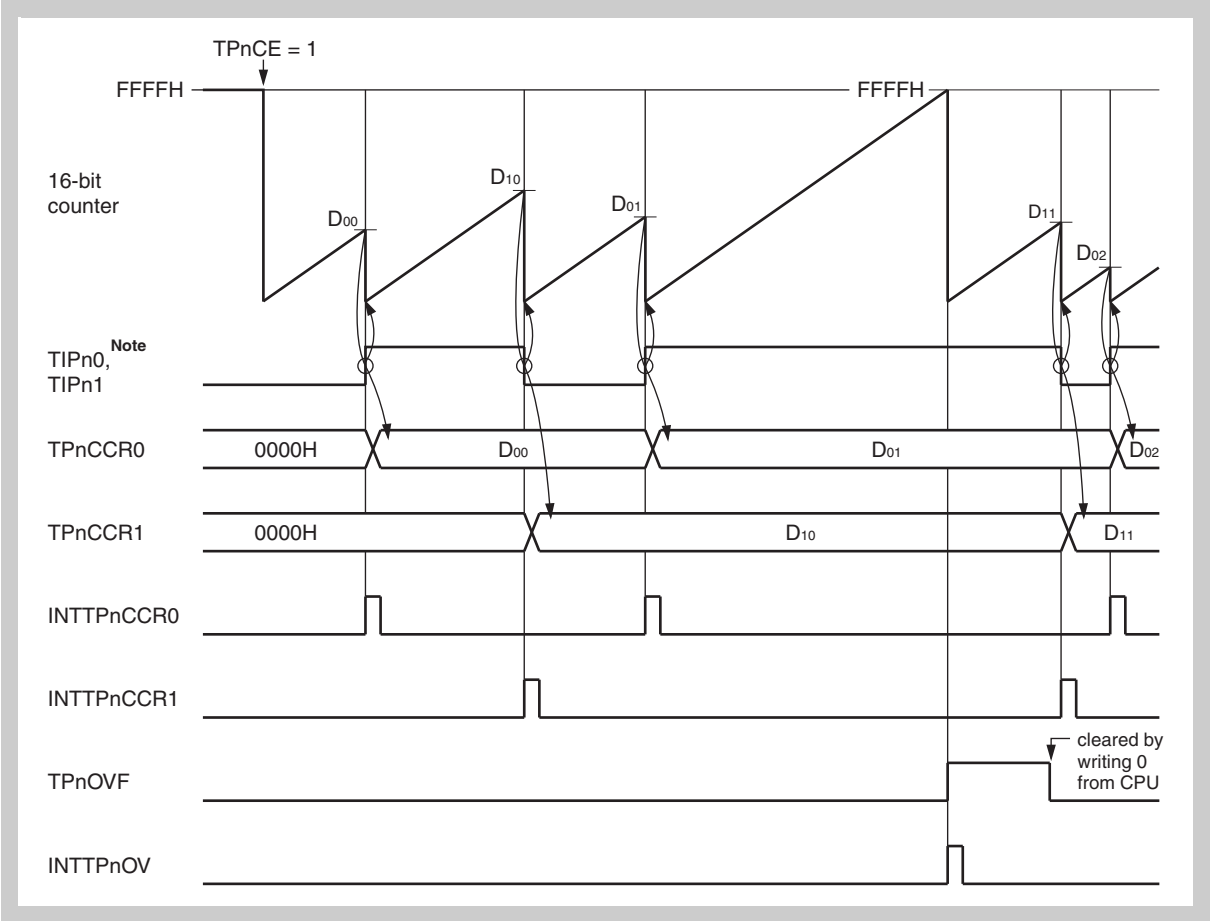


Figure 9-26 Basic operation timing of simultaneous pulse width and pulse space measurement

Note The signal to measure has to be assigned to both inputs, TIPn0 and TIPn1. This can be done either by external pin connection, or internally when selecting TIPn1 input on TIPn0 pin. In case of internal connection the signal has to be input on TIPn0 pin.

Remarks

1. D_{00}, D_{01}, D_{02}: Values captured to TPnCCR0 register (0000_H to $FFFF_H$)
2. D_{10}, D_{11}: Values captured to TPnCCR1 register (0000_H to $FFFF_H$)
3. TIPn0: Set detection to rising edge (TPnIS[1:0] = 01B)
4. TIPn1: Set detection to falling edge (TPnIS[3:2] = 10B)

9.5.9 Counter synchronous operation function

Timer P supports a function to start several timers P simultaneously. For this purpose two timer groups are defined, TMP0 to TMP3, as well as TMP4 to TMP7. For each timer group the counting of one to three slave counters (TMP1 to TMP3, or TMP5 to TMP7) can be synchronized with the corresponding master counter (TMP0 or TMP4). The synchronous operation function is enabled for each incorporated timer by the TPnSYE bit in the TPnCTL1 register (refer to *"TMPn control register 1 (TPnCTL1)" on page 219*).

When enabling the synchronous operation function, observe the following procedure:

<1> Clear the synchronous mode selection bit TPmSYE of the master counter TMPm to 0.
<2> Disable the count operation of the master counter TMPm (TPmCE = 0).
<3> Enable the synchronous operation for each of the incorporated slave counters TMPs (TPsSYE = 1).
<4> Enable the operation of the master counter TMPm (TPmCE = 1).

Master and incorporated slave counters of that group start and clock synchronously. When the master counter is cleared, the slave counters are cleared synchronously too.

Caution

1. In synchronous operation mode, the master counter can be used only in PWM mode (TPmMD = 100B), external trigger pulse output mode (TPmMD[2:0] = 010B), one-shot pulse output mode (TPmMD[2:0] = 011B), and free-running mode (TPmMD[2:0] = 101B).
2. In synchronous operation mode, the slave counters can be used in free-running mode only (TPsMD[2:0] = 101B).

Remark Maste index: m = 0, 4
Slave index: s = 1 to 3, wenn m = 0
s = 5 to 7, wenn m = 4

9.6 Cautions

(1) Capture operation of illegal data before first counting up

In free-running mode (TPnMD[3:0] = 0101_B), pulse width measurement mode (TPnMD[3:0] = 0110_B), and offset trigger generation mode (TPnMD[3:0] = 1100_B, TMT only), when a lower count clock (TPnEEE = 0, TPnCKS[2:0] = 011_B to 111_B) or an external clock (TPnEEE = 1) is selected, the timer captures the value of FFFFH and outputs a capture interrupt signal (INTTPnCCm), if a capture trigger signal (TIPnm) is enabled and input before first counting up. This captured data and the corresponding interrupt might be useless.

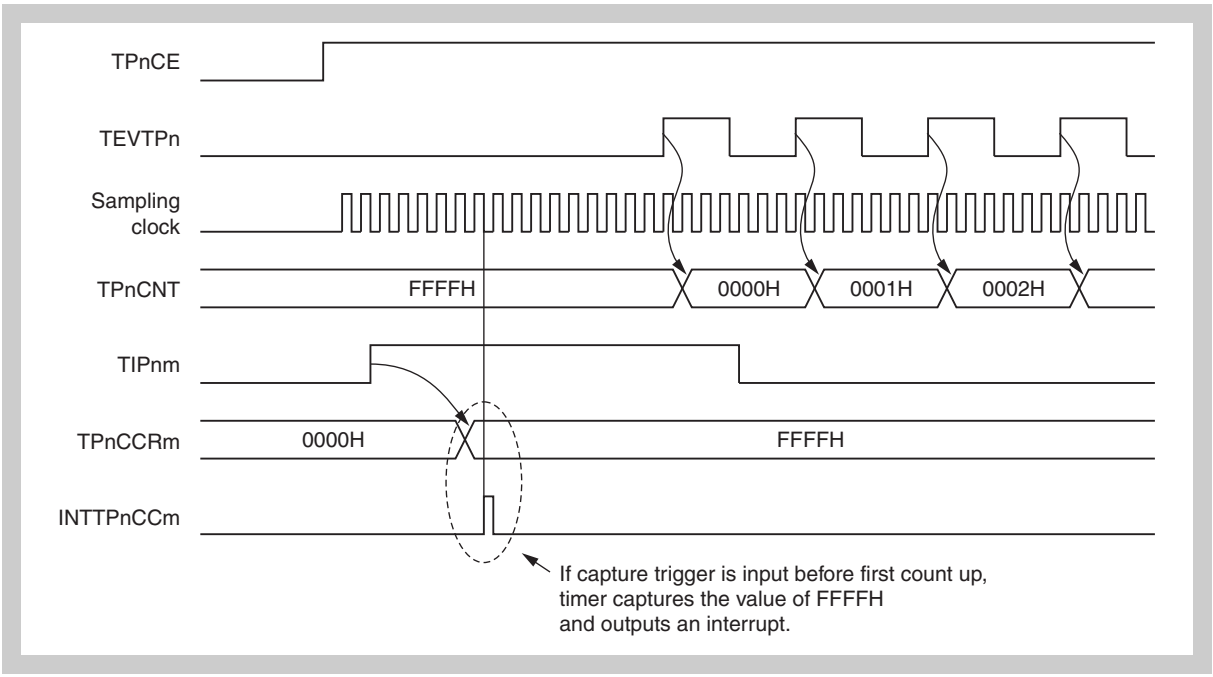


Figure 9-27 Capture operation of illegal data before first counting up

Chapter 10 16-bit Inverter Timer/Counter R (TMR)

This microcontroller has two instances of the Timer R (TMR), TMR0 and TMR1.

Note Throughout this chapter, the individual instances of Timer R are identified by “n” (n = 0 or 1), for example TRnCTL1 for the Timer Rn control register 1.

10.1 Features

Timer R is a 16-bit timer/counter that provides various motor control functions.

- Count clock resolution: 31.25 ns min. (when using 32 MHz count clock)
- General-purpose timer and operation mode supporting various motor control methods
- Compare registers with reload buffers
- 10-bit dead time counter
  - Dead time value independently settable through normal phase → inverted phase → normal phase
- A/D conversion trigger signal generation
  - Generation of A/D conversion trigger with 2 compare registers, TRnCCR4 and TRnCCR5
  - Dedicated output pin (TORn7) set with the TRnADTRG0 signal and reset with the TRnADTRG1 signal
- Interrupt thinning out function
  - Thinning out rates of 1/1 to 1/32
- Forced output stop function: Emergency Shut-off (ESO)
  - High-impedance output of pins TORn0 to TORn7 possible during ESOn input
- Compare value setting
  - Reload (batch rewrite)/anytime rewrite mode selectable **Note**
- Reload mode
  - Reload enabled by writing to TRnCCR1 register last, multiple registers simultaneity maintained
  - Peak/valley/peak and valley reload, transfer possible at reload timing **Note**
  - Provision of reload request flag TRnRSF
  - DMA transferable register address placement
- High-accuracy T-PWM mode
  - 0 to 100% duty PWM output possible, including dead time reduction
  - Increased output resolution without software load, because presence/absence of added pulse to PWM output on up-count side can be controlled with LSB of compare register
- 8 selectable count clocks: $\phi/2$, $\phi/4$, $\phi/8$, $\phi/16$, $\phi/32$, $\phi/64$, $\phi/256$, $\phi/1024$
- Active level of output pins TORn0 to TORn7 settable for each pin
- Fail-safe function (error interrupt output possible)
  - Simultaneous active output detection function in normal phase/inverted phase

Note High-accuracy T-PWM mode

10.2 Configuration

Timer R is configured of the following hardware.

Table 10-1 Timer R Configuration

Item	Configuration
Counters	16-bit counter × 1 16-bit sub-counter × 1 10-bit dead time counter × 3
Registers	Timer Rn counter read register (TRnCNT) Timer Rn sub-counter read register (TRnSBC) Timer Rn dead time setting registers 0, 1 (TRnDTC0, TRnDTC1) Timer Rn capture/compare registers 0 to 3 (TRnCCR0-TRnCCR3) Timer Rn compare registers 4, 5 (TRnCCR4, TRnCCR5) TRnCCR0 to TRnCCR5 buffer registers TRnDTC0, TRnDTC1 buffer registers
Timer input pins	3 (TIR10 to TIR13, TTRGR1, TEVTR1, ESOn)[a]
Timer output pins	8 (TORn0 to TORn7)[a]
Timer input signal	-
Timer output signal	TRnADTRG0, TRnADTRG1
Control registers	Timer Rn control registers 0, 1 (TRnCTL0, TRnCTL1) Timer Rn I/O control registers 0 to 4 (TRnIOC0 to TRnIOC4) Timer Rn option registers 0 to 3, 6, 7 (TRnOPT0 to TRnOPT3, TRnOPT6, TRnOPT7)
Interrupt requests	Compare match interrupts (INTTRnCC0 to INTTRnCC5) Peak interrupt (INTTRnCD) Valley interrupt (INTTRnOD) Overflow interrupt (INTTRnOV) Error interrupt (INTTRnER)

a) alternate-function pins

Table 10-2 List of Timer R registers

Address	Register name	Symbol	R/W	Manipulable Bit Units			After Reset
				1	8	16	
FFFFF581$_H$	TMR0 control register 1	TR0CTL1	R/W	×	×		00$_H$
FFFFF582$_H$	TMR0 I/O control register 0	TR0IOC0	R/W	×	×		00$_H$
FFFFF585$_H$	TMR0 I/O control register 3	TR0IOC3	R/W	×	×		00$_H$
FFFFF586$_H$	TMR0 I/O control register 4	TR0IOC4	R/W	×	×		00$_H$
FFFFF587$_H$	TMR0 option register 0	TR0OPT0	R/W	×	×		00$_H$
FFFFF588$_H$	TMR0 option register 2	TR0OPT2	R/W	×	×		00$_H$
FFFFF589$_H$	TMR0 option register 3	TR0OPT3	R/W	×	×		00$_H$
FFFFF58C$_H$	TMR0 option register 6	TR0OPT6	R/W	×	×		00$_H$
FFFFF58D$_H$	TMR0 option register 7	TR0OPT7	R/W	×	×		00$_H$
FFFFF58E$_H$	TMR0 option register 1	TR0OPT1	R/W	×	×		00$_H$
FFFFF590$_H$	TMR0 capture/compare register 5	TR0CCR5	R/W			×	0000$_H$
FFFFF592$_H$	TMR0 capture/compare register 4	TR0CCR4	R/W			×	0000$_H$
FFFFF598$_H$	TMR0 capture/compare register 0	TR0CCR0	R/W			×	0000$_H$
FFFFF59A$_H$	TMR0 capture/compare register 3	TR0CCR3	R/W			×	0000$_H$
FFFFF59C$_H$	TMR0 capture/compare register 2	TR0CCR2	R/W			×	0000$_H$

Table 10-2 List of Timer R registers

Address	Register name	Symbol	R/W	Manipulable Bit Units			After Reset
				1	8	16	
$FFFFF59E_H$	TMR0 capture/compare register 1	TR0CCR1	R/W			×	0000_H
$FFFFF5A0_H$	TMR0 dead time set register 0	TR0DTC0	R/W			×	0000_H
$FFFFF5A2_H$	TMR0 dead time set register 1	TR0DTC1	R/W			×	0000_H
$FFFFF5A4_H$	TMR0 timer counter read register	TR0CNT	R/W			×	0000_H
$FFFFF5A6_H$	TMR0 timer sub-counter read register	TR0SBC	R/W			×	0000_H
$FFFFF5C0_H$	TMR1 control register 0	TR1CTL0	R/W	×	×		00_H
$FFFFF5C1_H$	TMR1 control register 1	TR1CTL1	R/W	×	×		00_H
$FFFFF5C2_H$	TMR1 I/O control register 0	TR1IOC0	R/W	×	×		00_H
$FFFFF5C3_H$	TMR1 I/O control register 1	TR1IOC1	R/W	×	×		00_H
$FFFFF5C4_H$	TMR1 I/O control register 2	TR1IOC2	R/W	×	×		00_H
$FFFFF5C5_H$	TMR1 I/O control register 3	TR1IOC3	R/W	×	×		00_H
$FFFFF5C6_H$	TMR1 I/O control register 4	TR1IOC4	R/W	×	×		00_H
$FFFFF5C7_H$	TMR1 option register 0	TR1OPT0	R/W	×	×		00_H
$FFFFF5C8_H$	TMR1 option register 2	TR1OPT2	R/W	×	×		00_H
$FFFFF5C9_H$	TMR1 option register 3	TR1OPT3	R/W	×	×		00_H
$FFFFF5CC_H$	TMR1 option register 6	TR1OPT6	R/W	×	×		00_H
$FFFFF5CD_H$	TMR1 option register 7	TR1OPT7	R/W	×	×		00_H
$FFFFF5CE_H$	TMR1 option register 1	TR1OPT1	R/W	×	×		00_H
$FFFFF5D0_H$	TMR1 capture/compare register 5	TR1CCR5	R/W			×	0000_H
$FFFFF5D2_H$	TMR1 capture/compare register 4	TR1CCR4	R/W			×	0000_H
$FFFFF5D8_H$	TMR1 capture/compare register 0	TR1CCR0	R/W			×	0000_H
$FFFFF5DA_H$	TMR1 capture/compare register 3	TR1CCR3	R/W			×	0000_H
$FFFFF5DC_H$	TMR1 capture/compare register 2	TR1CCR2	R/W			×	0000_H
$FFFFF5DE_H$	TMR1 capture/compare register 1	TR1CCR1	R/W			×	0000_H
$FFFFF5E0_H$	TMR1 dead time set register 0	TR1DTC0	R/W			×	0000_H
$FFFFF5E2_H$	TMR1 dead time set register 1	TR1DTC1	R/W			×	0000_H
$FFFFF5E4_H$	TMR1 timer counter read register	TR1CNT	R			R	0000_H
$FFFFF5E6_H$	TMR1 timer sub-counter read register	TR1SBC	R			R	0000_H

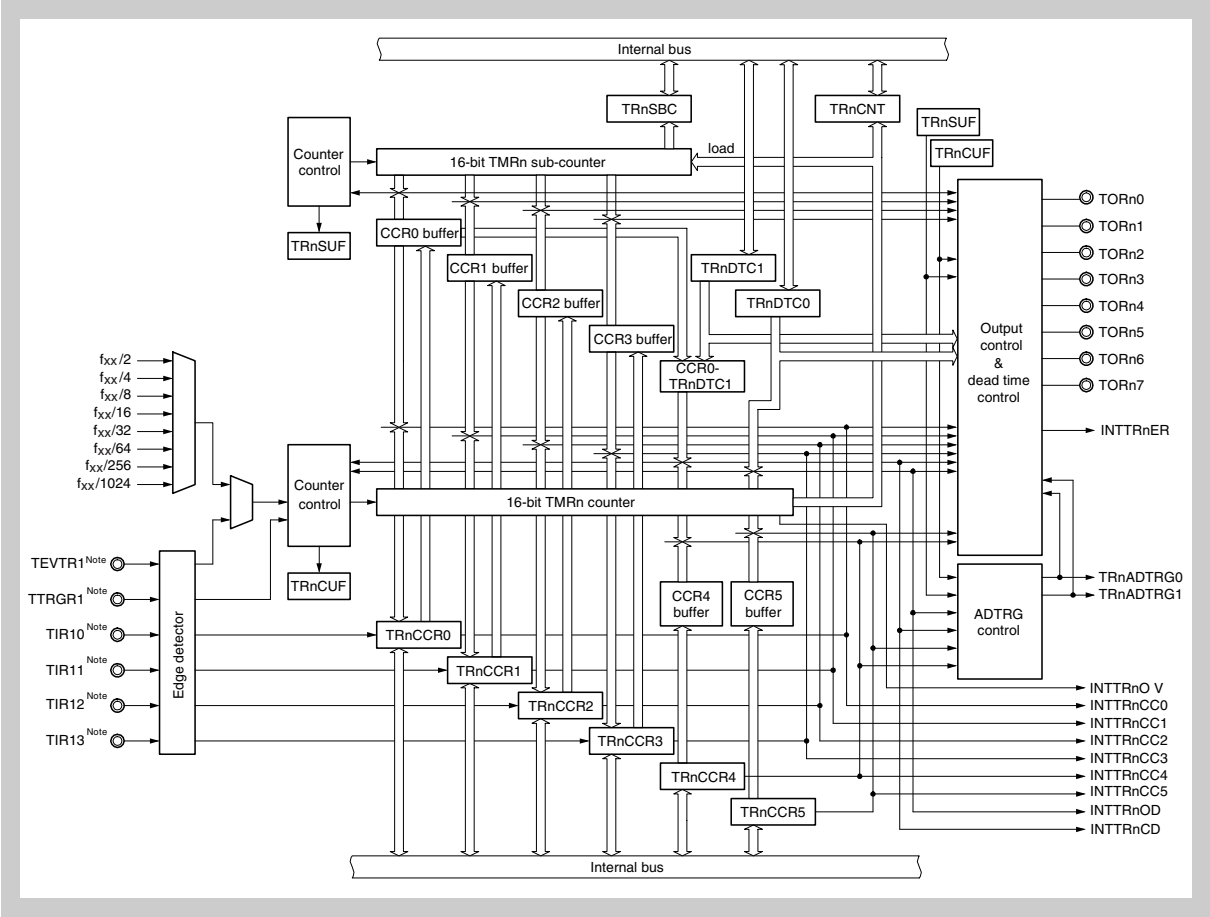


Figure 10-1 Block diagram of timer R

Note Timer inputs are only available in TMR1 (n = 1). The TIR10 to TIR13 capture inputs are shared with TOR11 to TOR14. External trigger input TTRGR1 is shared with TIR10 output, and external event input TEVTR1 is shared with TIR17 output.

Remark f_{XX}: Internal system clock

(1) TMRn capture/compare register 0 (TRnCCR0)

The TRnCCR0 register is a 16-bit register provided with a capture function and a compare function.

In case of free-running mode only, the TRnOPT0.TRnCCS0 bit specifies the usage of the TRnCCR0 register as a capture register or as a compare register.

In pulse width measurement mode, the TRnCCR0 register is always used as a capture register.

In modes other than the free-running mode or the pulse width measurement mode, the TRnCCR0 register is used as a compare register.

Access This registers can be read/written in 16-bit units.

Address TR0CCR0: $FFFFF598_H$
TR1CCR0: $FFFFF5D8_H$

Initial Value 0000_H. This register is cleared by any reset.

Note In high-accuracy T-PWM mode, writing to bit 0 of the TRnCCR0 register is ignored. Moreover, bit 0 is always read as 0.

TRnCCR0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Capture/compare value 0															
R/W															

(a) Use as compare register

When TRnCTL0.TRnCE = 1, the TRnCCR0 register write access method is as follows. For details about the compare register rewrite operation, refer to *"Compare register rewrite operation" on page 303*.

Table 10-3 TRnCCR0 register write access mode

Timer Rn Operation Mode	TRnCCR0 Register Write Access Mode
PWM mode, external trigger pulse output mode, triangular wave PWM mode, PWM mode with dead time	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 switchable

Caution To set the carrier frequency in the high-accuracy T-PWM mode, set the TRnCCR0 register as follows:
Number of count clocks of carrier frequency + TRnDTC0 register value + TRnDTC1 register value.
For details about the carrier wave and dead time settings, refer to *"High-accuracy T-PWM mode" on page 378*.

(b) Use as capture register

The counter value is saved to the TR1CCR0 register upon detection of the edge of the capture trigger (TIR10) input.

Note The capture function is provided only for TMR1.

(2) TMRn capture/compare register 1 (TRnCCR1)

The TRnCCR1 register is a 16-bit register that functions both as a capture register and a compare register.

When a compare register is rewritten in the reload mode, the reload request flag (TRnOPT6.TRnRSF) becomes 1 when write access is performed to the TRnCCR1 register, and all the registers are rewritten at the same time at the next reload timing.

Only in free-running mode, the TRnOPT0.TRnCCS1 bit specifies whether the TRnCCR1 register is used as a capture register or as a compare register.

In pulse width measurement mode, the TRnCCR1 register is always used as a capture register.

In modes other than the free-running mode and the pulse width measurement mode, the TRnCCR1 register is used as a compare registers.

Access This registers can be read/written in 16-bit units.

Address TR0CCR1: $FFFFF59E_H$
TR1CCR1: $FFFFF5DE_H$

Initial Value 0000_H. This register is cleared by any reset.

Note In high-accuracy T-PWM mode, when bit 0 is set to 1, the additional pulse control function is engaged. (For details about the additional pulse control function, refer to *"Additional pulse control in high-accuracy T-PWM mode" on page 386*.)

TRnCCR1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Capture/compare value 1															
R/W															

(a) Use as compare register

When TRnCTL0.TRnCE = 1, the TRnCCR1 register write access method is as follows. For details about the compare register rewrite operation, refer to *"Compare register rewrite operation" on page 303*.

Table 10-4 TRnCCR1 register write access mode

Timer Rn Operation Mode	TRnCCR1 Register Write Access Mode
PWM mode, external trigger pulse output mode, triangular wave PWM mode, PWM mode with dead time	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 switchable

(b) Use as capture register

The counter value is saved to the TR1CCR1 register upon detection of the edge of the capture trigger (TIR11) input.

Note The capture function is provided only for TMR1.

(3) TMRn capture/compare register 2 (TRnCCR2)

The TRnCCR2 register is a 16-bit register that functions both as a capture register and compare register.

Only in free-running mode, the TRnOPT0.TRnCCS2 bit specifies whether the TRnCCR2 register is used as a capture register or a compare register.

In the pulse width measurement mode, the TRnCCR2 register is always used as a capture register.

In modes other than the free-running mode and the pulse width measurement mode, the TRnCCR2 register is used as a compare registers.

Access This registers can be read/written in 16-bit units.

Address TR0CCR2: $FFFFF59C_H$
TR1CCR2: $FFFFF5DC_H$

Initial Value 0000_H. This register is cleared by any reset.

Note In high-accuracy T-PWM mode, when bit 0 is set to 1, the additional pulse control function is engaged. (For details about the additional pulse control function, refer to *"Additional pulse control in high-accuracy T-PWM mode" on page 386*.)

TRnCCR2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Capture/compare value 2															
R/W															

(a) Use as compare register

When TRnCTL0.TRnCE = 1, the TRnCCR2 register write access method is as follows. For details about the compare register rewrite operation, refer to *"Compare register rewrite operation" on page 303*.

Table 10-5 TRnCCR2 register write access mode

Timer Rn Operation Mode	TRnCCR2 Register Write Access Mode
PWM mode, external trigger pulse output mode, triangular wave PWM mode, PWM mode with dead time	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 switchable

(b) Use as capture register

The counter value is saved to the TRnCCR2 register upon detection of the edge of the capture trigger (TIR12) input.

Note The capture function is provided only for TMR1.

(4) TMRn capture/compare register 3 (TRnCCR3)

The TRnCCR3 register is a 16-bit register that functions both as a capture register and a compare register.

Only in free-running mode, the TRnOPT0.TRnCCS3 bit specifies whether the TRnCCR3 register is used as a capture register or a compare register.

In the pulse width measurement mode, the TRnCCR3 register is always used as a capture register.

In modes other than the free-running mode and the pulse width measurement mode, the TRnCCR3 register is used as a compare registers.

Access This registers can be read/written in 16-bit units.

Address TR0CCR3: $FFFFF59A_H$
TR1CCR3: $FFFFF5DA_H$

Initial Value 0000_H. This register is cleared by any reset.

Note In high-accuracy T-PWM mode, when bit 0 is set to 1, the additional pulse control function is engaged. (For details about the additional pulse control function, refer to *"Additional pulse control in high-accuracy T-PWM mode" on page 386*.)

TRnCCR3

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Capture/compare value 3															
R/W															

(a) Use as compare register

When TRnCTL0.TRnCE = 1, the TRnCCR3 register write access method is as follows. For details about the compare register rewrite operation, refer to *"Compare register rewrite operation" on page 303*.

Table 10-6 TRnCCR3 register write access mode

Timer Rn Operation Mode	TRnCCR3 Register Write Access Mode
PWM mode, external trigger pulse output mode, triangular wave PWM mode, PWM mode with dead time	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 switchable

(b) Use as capture register

The counter value is saved to the TR1CCR3 register upon detection of the edge of the capture trigger (TIR13) input.

Note The capture function is provided only for TMR1.

(5) TMRn compare register 4 (TRnCCR4)

The TRnCCR4 register is a 16-bit register compare register.

In high-accuracy T-PWM mode and PWM mode with dead time, the interrupt (INTTRnCC4) for matches between the counter and the TRnCCR4 register can be selected as the timing for A/D conversion trigger input.

Access This registers can be read/written in 16-bit units.

Address TR0CCR4: $FFFFF592_H$
TR1CCR4: $FFFFF5D2_H$

Initial Value 0000_H. This register is cleared by any reset.

Note In high-accuracy T-PWM mode, bit 0 of the TRnCCR4 register is ignored.

TRnCCR4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Capture/compare value 4															
R/W															

When TRnCTL0.TRnCE = 1, the TRnCCR4 register write access method is as follows. For details about the compare register rewrite operation, refer to *"Compare register rewrite operation" on page 303*.

Table 10-7 TRnCCR4 register write access mode

Timer Rn Operation Mode	TRnCCR4 Register Write Access Mode
PWM mode, external trigger pulse output mode, triangular wave PWM mode, PWM mode with dead time	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 switchable

(6) TMRn compare register 5 (TRnCCR5)

The TRnCCR5 register is a 16-bit compare register.

In high-accuracy T-PWM mode and PWM mode with dead time, the interrupt (INTTRnCC5) for matches between the counter and the TRnCCR5 register can be selected as the timing for A/D conversion trigger input.

Access This registers can be read/written in 16-bit units.

Address TR0CCR5: $FFFFF590_H$
TR1CCR5: $FFFFF5D0_H$

Initial Value 0000_H. This register is cleared by any reset.

Note In high-accuracy T-PWM mode, bit 0 of the TRnCCR5 register is ignored.

TRnCCR5

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Capture/compare value 4															
R/W															

When TRnCTL0.TRnCE = 1, the TRnCCR5 register write access method is as follows. For details about the compare register rewrite operation, refer to *"Compare register rewrite operation" on page 303*.

Table 10-8 TRnCCR5 register write access mode

Timer Rn Operation Mode	TRnCCR5 Register Write Access Mode
PWM mode, external trigger pulse output mode, triangular wave PWM mode, PWM mode with dead time	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 switchable

(7) TMRn counter read register (TRnCNT)

The TRnCNT register is a timer read register that can read the current value of the 16-bit counter.

During the interval from when CE = 1 until count up, the value of the TRnCNT register is FFFFH.

Access This registers can only be read in 16-bit units.

Address TR0CNT: $FFFFF5A4_H$
TR1CNT: $FFFFF5E4_H$

Initial Value 0000_H. This register is cleared by any reset, or when TRnCTL0.TRnCE = 0 is set.

Caution During the interval from when TRnCTL0.TRnCE = 1 until count up, the value of the TRnCNT register is FFFFH.

TRnCNT

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Counter value															**Note**
R															

Note In high-accuracy T-PWM mode, bit 0 is read as 0.

(8) TMRn sub-counter read register (TRnSBC)

The TRnSBC register is a timer read register that can read the current value of the 16-bit sub-counter.

Access This registers can only be read in 16-bit units.

Address TR0SBC: $FFFFF5A6_H$
TR1SBC: $FFFFF5E6_H$

Initial Value 0000_H. This register is cleared by any reset, or when TRnCTL0.TRnCE = 0 is set.

Note This register is used only in high-accuracy T-PWM mode and PWM mode with dead time. In all other modes it has no meaning.

TRnSBC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Sub-counter value															**Note**
R															

Note In high-accuracy T-PWM mode, bit 0 is read as 0.

(9) TMRn dead time setting register 0 (TRnDTC0)

The TRnDTC0 register is a 10-bit register that specifies the dead time value 0.

Access This registers can be read/written in 16-bit units.

Address TR0DTC0: $FFFFF5A0_H$
TR1DTC0: $FFFFF5E0_H$

Initial Value 0000_H. This register is cleared by any reset.

Caution

1. The TRnDTC0 register must not be written with a different value during TMRn operation (TRnCTL0.TRnCE = 1).
2. When the TRnDTC0 register is set to 0000H, dead time is not inserted.
3. Bits 0 and 10 to 15 are fixed to 0.
4. The dead time counter operates only in high-accuracy T-PWM mode and PWM mode with dead time. In all other modes, the TRnDTC0 register must be set to 0000H.

TRnDTC0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	Dead time value 0									0
R/W															

(10) TMRn dead time setting register 1 (TRnDTC1)

The TRnDTC1 register is a 10-bit register that specifies the dead time value 1.

Access This registers can be read/written in 16-bit units.

Address TR0DTC1: $FFFFF5A2_H$
TR1DTC1: $FFFFF5E2_H$

Initial Value 0000_H. This register is cleared by any reset.

Caution

1. The TRnDTC1 register must not be written with a different value during TMRn operation (TRnCTL0.TRnCE = 1).
2. When the TRnDTC1 register is set to 0000H, dead time is not inserted.
3. Bits 0 and 10 to 15 are fixed to 0.
4. The dead time counter operates only in high-accuracy T-PWM mode and PWM mode with dead time. In all other modes, the TRnDTC1 register must be set to 0000H.

TRnDTC1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	Dead time value 1									0
R/W															

10.3 Control Registers

(1) TMRn control register 0 (TRnCTL0)

The TRnCTL0 register is an 8-bit register that controls the operation of TMRn.

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

Address TR0CTL0: $FFFFF580_H$
TR1CTL0: $FFFFF5C0_H$

Initial Value 00_H. This register is cleared by any reset.

TRnCTL0

7	6	5	4	3	2	1	0
TRnCE	0	0	0	0	TRnCKS2	TRnCKS1	TRnCKS0
R/W	R	R	R	R	R/W	R/W	R/W

Caution When TRnCE = 1, only the TRnCE bit of the TRnCTL0 register can be changed. Perform write access to the other bits using the same values.

Table 10-9 TRnCTL0 register contents (1/2)

Bit position	Bit name	Function
7	TRnCE	Controls the timer TMRn operation. 0: Internal operating clock operation disabled (TMRn reset asynchronously) 1: Internal operating clock operation enabled When bit TRnCE is set to “0”, the internal operation clock of TMRn stops (fixed to low level), and TMRn is set asynchronously. When bit TRnCE is set to “1”, the internal operation of TMRn is enabled from when bit TRnCE was set to “1” and count-up is performed. The time until count-up is as listed in *Table 10-10, “TMRn Count Clock and Count Delay,” on page 282*. **Remark:** By setting TRnCE = 0 following functions of TMRn are reset. • Internal registers and internal latch circuits other than registers that can be written to/from the CPU • TRnOPT0.TRnOVF flag and flags in TRnOPT6 register • Counter, sub-counter, dead time counter, counter read register, sub-counter read register • TRnCCR0 to TRnCCR5 buffer registers, TRnDTC0 buffer register, and TRnDTC1 buffer register • Timer output (inactive level output)

Table 10-9 TRnCTL0 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>2 to 0</td><td>TRnCKS2
TRnCKS1
TRnCKS0</td><td>Selects the count clock of timer TMRn.
<table>
<tr><th>TRnCKS2</th><th>TRnCKS1</th><th>TRnCKS0</th><th>Internal Count Clock Selection</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>$f_{XX}/2$</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>$f_{XX}/4$</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>$f_{XX}/8$</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>$f_{XX}/16$</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>$f_{XX}/32$</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>$f_{XX}/64$</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>$f_{XX}/256$</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>$f_{XX}/1024$</td></tr>
</table>
Caution: Set bits TRnCKS[2:0] when TRnCE = 0.
When bit TRnCE is set from 0 to 1, bits TRnCKS[2:0] can be simultaneously set.</td></tr>
</table>

Table 10-10 TMRn Count Clock and Count Delay

Count clocks	TRnCKS2	TRnCKS1	TRnCKS0	Count delay	
				Minimum	Maximum
$f_{XX}/2$	0	0	0	3 base clocks	4 base clocks
$f_{XX}/4$	0	0	1		
$f_{XX}/8$	0	1	0		
$f_{XX}/16$	0	1	1	4 base clocks	5 base clocks + 1 count clock
$f_{XX}/32$	1	0	0		
$f_{XX}/64$	1	0	1		
$f_{XX}/256$	1	1	0		
$f_{XX}/1024$	1	1	1		

Note 1. f_{XX}: System clock

2. f_{TMRn}: Base clock of TMRn ($f_{TMRn} = f_{XX}/2$)

(2) TMRn control register 1 (TRnCTL1)

The TRnCTL1 register is an 8-bit register that controls the operation of TMRn.

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

Address TR0CTL1: $FFFFF581_H$
TR1CTL1: $FFFFF5C1_H$

Initial Value 00_H. This register is cleared by any reset.

TRnCTL1

7	6	5	4	3	2	1	0
0	TRnEST	TRnEEE	0	TRnMD3	TRnMD2	TRnMD1	TRnMD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Caution Set the TRnCTL1 register when TRnCTL0.TTnCE = 0. When TRnCE = 1, TRnEEE and TRnMD[3:0] bits can be written using the same value.

Table 10-11 TRnCTL1 register contents (1/2)

Bit position	Bit name	Function
6	TRnEST	Sets to enable/disable software trigger control. 0: No operation/Disables software trigger control. 1: Enables software trigger control. - In one-shot pulse mode: One-shot pulse software trigger) - In external trigger pulse output mode :Pulse output software trigger **Caution:** The TRnEST bit operates as a software trigger by it to 1 during TMRn operation (when TRnCTL0.TRnCE = 1). Do not write TRnEST = 1 when TRnCTL0.TRnCE = 0. **Note:** The TRnEST bit is always read as “0”.
5	TRnEEE	Specifies count clock input. 0: Use the internal clock selected with the TRnCTL0.TRnCKS[2:0] bits. 1: Use the external clock input (TEVTRn pin input edge). **Note:** External clock input pin is available for TMR1 only (TEVTR1). When TR1EEE = 1 (external clock input TEVTR1), the valid edge is specified by TRnIOC2.TR1EES[1:0] bits.

Table 10-11 TRnCTL1 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3 to 0</td><td>TRnMD3
TRnMD2
TRnMD1
TRnMD0</td><td>Selects the timer mode..
<table>
<tr><th>TTnMD3</th><th>TTnMD2</th><th>TTnMD1</th><th>TTnMD0</th><th>Internal count clock selection</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Interval timer mode</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>External event count mode[a]</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>External trigger pulse output mode[b]</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>One-shot pulse mode</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>PWM mode</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>Free-running mode</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>Pulse width measurement mode[a]</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>Triangular wave PWM mode</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>High accuracy T-PWM mode</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>PWM mode with dead time</td></tr>
<tr><td colspan="4">Other than above</td><td>Setting prohibited</td></tr>
</table>
a) Setting prohibited for TMR0.
b) For TMR0 an output pulse can be triggered only by software trigger (TR0EST = 1).</td></tr>
</table>

(3) TMRn I/O control register 0 (TRnIOC0)

The TRnIOC0 register is an 8-bit register that controls the timer output (pins TORn0 to TORn3).

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

Address TR0IOC0: $FFFFF582_H$
TR1IOC0: $FFFFF5C2_H$

Initial Value 00_H. This register is cleared by any reset.

TRnIOC0

7	6	5	4	3	2	1	0
TRnOL3	TRnOE3	TRnOL2	TRnOE2	TRnOL1	TRnOE1	TRnOL0	TRnOE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Caution If dead time cannot be secured or if spikes (noise) may occur on the output pin, set the TRnIOC0 register when TRnCTL0.TRnCE = 0. When TRnCE = 1, the TRnIOC0 register can be written using the same value.

Table 10-12 TRnIOC0 register contents

Bit position	Bit name	Function
7, 5, 3, 1	TRnOLm	TORnm pin output level setting: 0: Normal output (Low level, when output is inactive.) 1: Inverted output (High level, when output is inactive.)
6, 4, 2, 0	TRnOEm	TORnm pin output setting: 0: Timer output disable (TORnm pin output is fixed to inactive level.) 1: Timer output enabled (A pulse can be output from the TORnm pin.)

Remark m = 0 to 3

(4) TMR1 I/O control register 1 (TR1IOC1)

The TR1IOC1 register is an 8-bit register that controls the valid edge of external signal inputs (pins TIR10 to TIR13) of TMR1.

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

Address $FFFFF5C3_H$

Initial Value 00_H. This register is cleared by any reset.

TR1IOC1

7	6	5	4	3	2	1	0
TR1IS7	TR1IS6	TR1IS5	TR1IS4	TR1IS3	TR1IS2	TR1IS1	TR1IS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Caution

1. Set the TR1IOC1 register when TR1CTL0.TRnCE = 0. When TR1CE = 1, write access to the TR1IOC1 register can be performed with the same value.
2. The TR1IOC1 register is valid only in free-running mode and pulse width measurement mode. In all other modes, capture operation is not performed.

Table 10-13 TR1IOC1 register contents (1/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7, 6</td><td>TR1IS7
TR1IS6</td><td>Specifies the capture input (TIR13) valid edge.
<table>
<tr><th>TR1IS7</th><th>TR1IS6</th><th>Capture input (TIR13) 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>
Capture operation is performed and capture interrupt (INTTR1CC3) is output upon edge detection.</td></tr>
<tr><td>5, 4</td><td>TR1IS5
TR1IS4</td><td>Specifies the capture input (TIR12) valid edge.
<table>
<tr><th>TR1IS5</th><th>TR1IS4</th><th>Capture input (TIR12) 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>
Capture operation is performed and capture interrupt (INTTR1CC2) is output upon edge detection.</td></tr>
</table>

Table 10-13 TR1IOC1 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3, 2</td><td>TR1IS3
TR1IS2</td><td>Specifies the capture input (TIR11) valid edge.
<table>
<tr><th>TR1IS3</th><th>TR1IS2</th><th>Capture input (TIR11) 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>
Capture operation is performed and capture interrupt (INTTR1CC1) is output upon edge detection.</td></tr>
<tr><td>1, 0</td><td>TR1IS1
TR1IS0</td><td>Specifies the capture input (TIR10) valid edge.
<table>
<tr><th>TR1IS1</th><th>TR1IS0</th><th>Capture input (TIR10) 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>
Capture operation is performed and capture interrupt (INTTR1CC0) is output upon edge detection.</td></tr>
</table>

(5) TMR1 I/O control register 2 (TR1IOC2)

The TR1IOC2 register is an 8-bit register that controls the valid edge of external event count input (pin TEVTR1) and external trigger input (pin TTRGR1) of TMR1.

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

Address $FFFFF5C4_H$

Initial Value 00_H. This register is cleared by any reset.

TR1IOC2

7	6	5	4	3	2	1	0
0	0	0	0	TR1EES1	TR1EES0	TR1ETS1	TR1ETS0
R	R	R	R	R/W	R/W	R/W	R/W

Caution Set the TR1IOC2 register when TR1CTL0.TR1CE = 0. When TR1CE = 1, write access to the TR1IOC2 register can be performed with the same value.

Table 10-14 TTnIOC2 register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3, 2</td><td>TR1EES1
TR1EES0</td><td>Specifies the external event counter input (TEVTR1) valid edge.
<table>
<tr><th>TR1EES1</th><th>TR1EES0</th><th>External event counter input (TEVTR1) 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>
Note: The settings of bits TR1EES1 and TR1EES0 are valid in the external event count mode (TR1CTL1.TR1MD0 = 0001_B), or when bit TR1CTL1.TR1EEE = 1.</td></tr>
<tr><td>1, 0</td><td>TR1ETS1
TR1ETS0</td><td>Specifies the external trigger input (TTRGR1) valid edge.
<table>
<tr><th>TR1ETS1</th><th>TR1ETS0</th><th>External trigger input (TTRGR1) 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>
Note: The settings of bits TR1ETS1 and TR1ETS0 are valid in the external trigger pulse output mode and the one-shot pulse mode (TR1CTL1.TR1MD0 = 0010_B or 0011_B).</td></tr>
</table>

(6) TMRn I/O control register 3 (TRnIOC3)

The TRnIOC3 register is an 8-bit register that controls timer output (pins TORn4 to TORn7).

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

Address TR0IOC3: $FFFFF585_H$
TR1IOC3: $FFFFF5C5_H$

Initial Value 00_H. This register is cleared by any reset.

TRnIOC3

7	6	5	4	3	2	1	0
TRnOL7	TRnOE7	TRnOL6	TRnOE6	TRnOL5	TRnOE5	TRnOL4	TRnOE4
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Caution If dead time cannot be secured or if spikes (noise) may occur on the output pin, set the TRnIOC3 register when TRnCTL0.TRnCE = 0. When TRnCE = 1, the TRnIOC3 register can be written using the same value.

Table 10-15 TRnIOC3 register contents

Bit position	Bit name	Function
7, 5, 3, 1	TRnOLm	TORnm pin output level setting: 0: Normal output (Low level, when output is inactive.) 1: Inverted output (High level, when output is inactive.)
6, 4, 2, 0	TRnOEm	TORnm pin output setting: 0: Timer output disable (TORnm pin output is fixed to inactive level.) 1: Timer output enabled (A pulse can be output from the TORnm pin.)

Remark m = 4 to 7

(7) TMRn I/O control register 4 (TRnIOC4)

The TRnIOC4 register is an 8-bit register that controls timer output error detection.

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

Address TR0IOC4: $FFFFF586_H$
TR1IOC4: $FFFFF5C6_H$

Initial Value 00_H. This register is cleared by any reset.

TRnIOC4

7	6	5	4	3	2	1	0
0	TRnTBA2	TRnTBA1	TRnTBA0	0	0	0	TRnEOC
R	R/W	R/W	R/W	R	R/W	R	R/W

Caution Set the TRnIOC4 register when TRnCTL0.TRnCE = 0. When TRnCE = 1, write access to the TRnIOC4 register can be performed using the same value.

Table 10-16 TRnIOC4 register contents

Bit position	Bit name	Function
6	TRnTBA2	Controls true bar active detection of timer outputs (TORn5/TORn6). 0: No detection of simultaneous active state of pins TORn5 and TORn6 1: Detection of simultaneous active state of pins TORn5 and TORn6 If simultaneous active state is detected when TRnTBA2 = 1, the TRnOPT6.TRnTBF flag is set (1), and an error interrupt (INTTRnER) is output.
5	TRnTBA1	Controls true bar active detection of timer outputs (TORn3/TORn4). 0: No detection of simultaneous active state of pins TORn3 and TORn4 1: Detection of simultaneous active state of pins TORn3 and TORn4 If simultaneous active state is detected when TRnTBA1 = 1, the TRnOPT6.TRnTBF flag is set (1), and an error interrupt (INTTRnER) is output.
4	TRnTBA0	Controls true bar active detection of timer outputs (TORn1/TORn2). 0: No detection of simultaneous active state of pins TORn1 and TORn2 1: Detection of simultaneous active state of pins TORn1 and TORn2 If simultaneous active state is detected when TRnTBA0 = 1, the TRnOPT6.TRnTBF flag is set (1), and an error interrupt (INTTRnER) is output.
0	TRnEOC	Controls the error interrupt output 0: Disable output of error interrupt (INTTRnER) 1: Enable output of error interrupt (INTTRnER) For details about error interrupt control, refer to *"Error Interrupts" on page 338*

(8) TMRn option register 0 (TRnOPT0)

The TRnOPT0 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: $FFFFF587_H$
TT1OPT0: $FFFFF5C7_H$

Initial Value 00_H. This register is cleared by any reset.

TR0OPT0

7	6	5	4	3	2	1	0
0	0	0	0	0	TR0CMS	TR0CUF	TR0OVF
R/W	R/W	R/W	R/W	R	R/W	R	R/W

TR1OPT0

7	6	5	4	3	2	1	0
TR1CCS3	TR1CCS2	TR1CCS1	TR1CCS0	0	TR1CMS	TR1CUF	TR1OVF
R/W	R/W	R/W	R/W	R	R/W	R	R/W

Caution Do not rewrite TR1CC3[3:0] bits during TMR1 operation (TRnCTL0.TRnCE = 1).

Table 10-17 TTnOPT0 register contents (1/2)

Bit position	Bit name	Function
/	TR1CCS3	Specifies the operation mode of register TR1CCR3 0: Operation as compare register 1: Operation as capture register **Note:** The setting of bit TR1CCS3 is valid in free-running mode only.
6	TR1CCS2	Specifies the operation mode of register TR1CCR2 0: Operation as compare register 1: Operation as capture register **Note:** The setting of bit TR1CCS2 is valid in free-running mode only.
5	TR1CCS1	Specifies the operation mode of register TR1CCR1 0: Operation as compare register 1: Operation as capture register **Note:** The setting of bit TR1CCS1 is valid in free-running mode only.
4	TR1CCS0	Specifies the operation mode of register TR1CCR0 0: Operation as compare register 1: Operation as capture register **Note:** The setting of bit TR1CCS0 is valid in free-running mode only.
2	TRnCMS	Specifies the compare register transfer timing mode. 0: Reload mode (batch rewrite): When the TRnCCR1 register is written to, all the registers are updated at the next reload timing (reload). Even if registers other than the TRnCCR1 register are written, reload is not executed. 1: Anytime rewrite mode: Each register is updated independently, and when write access is performed to a compare register, the register is updated to the value used during anytime write access. Several clocks are required until the value is transferred to the register following write. (Refer to *"Compare register rewrite operation" on page 303*). **Note:** The TRnCMS bit is valid only in the high-accuracy T-PWM mode, In all other modes it is invalid and must be cleared (TRnCMS = 0).

Table 10-17 TTnOPT0 register contents (2/2)

Bit position	Bit name	Function
1	TRnCUF	Indicates the TMRn count direction. 0: The timer counter is in up count state. 1: The timer counter is in down count state. **Note:** The TRnCUF bit is valid only in the high-accuracy T-PWM mode and triangular wave PWM mode. In all other modes, it is invalid (TRnCUF = 0).
0	TRnOVF	Indicates TMRn overflow 0: No overflow occurrence after timer restart or flag reset 1: Overflow occurrence **Remarks: 1.** The TRnOVF bit is set (1) when the 16-bit counter value overflows from FFFFH to 0000H. **2.** The TRnOVF bit is cleared (0) when either 0 is written to it, or TRnCTL0.TRnCE = 0 is set. **3.** When TRnOVF bit is set (1), an overflow interrupt (INTTRnOV) is simultaneously output. **Caution: 1.** Overflow can only occur in the free-running mode and the T-PWM mode. If, in the high-accuracy T-PWM mode, the set conditions for the TRnDTC0 and TRnDTC1 registers are incorrect, the TRnOVF bit may be set (1). **2.** When TRnOVF = 1, even if the TRnOVF bit and the TRnOPT0 register are read, the TRnOVF bit is not cleared. **3.** The TRnOVF bit can be read and written, but even if “1” is written to TRnOVF bit from the CPU, this is ignored.

(9) TMRn option register 1 (TRnOPT1)

The TRnOPT1 register is an 8-bit register used to enable/disable peak/valley interrupts and set interrupt thinning out.

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

Address TR0OPT1: $FFFFF58E_H$
TR1OPT1: $FFFFF5CE_H$

Initial Value 00_H. This register is cleared by any reset.

TRnOPT1

7	6	5	4	3	2	1	0
TRnICE	TRnIOE	TRnRDE	TRnID4	TRnID3	TRnID2	TRnID1	TRnID0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Caution

1. The TRnOPT1 register write method is as follows.
   - In high-accuracy T-PWM mode: Anytime write, or reload write
   - In mode other than high-accuracy T-PWM mode: Reload write
2. Do not set TRnICE = 0 and TRnIOE = 0.
   Since reload does not occur when TRnICE = TRnIOE = 0, the TRnOPT1 register, which is a reload write register, stops being updated.

Table 10-18 TRnOPT1 register contents (1/2)

Bit position	Bit name	Function
/	TRnICE	Controls the peak interrupt (INTTRnCD) 0: Disable peak interrupt (INTTRnCD) output in the counter's peak timing Interrupt thinning out is not performed. Reload operation is disabled in the counter's peak timing. 1: Enable peak interrupt (INTTRnCD) in the counter's peak timing Interrupt thinning out is performed. Reload operation is enabled in the counter's peak timing. **Note:** Bit TRnICE is valid only in the PWM mode, high-accuracy T-PWM mode, and PWM mode with dead time.
6	TRnIOE	Controls the valley interrupt (INTTRnOD) 0: Disable valley interrupt (INTTRnOD) output in the counter's valley timing Reload operation is disabled in the counter's valley timing. 1: Enable valley interrupt (INTTRnOD) output in the counter's valley timing Reload operation is enabled in the counter's valley timing. **Note:** Bit TRnIOE is valid only in the high-accuracy T-PWM mode and triangular wave PWM mode.
5	TRnRDE	Controls reload timing thinning out 0: Don't perform reload thinning out Reload timing occurs at each peak/valley. 1: Perform reload thinning out Reload timing occurs at the same interval as interrupt thinning out. **Note:** Bit TRnRDE is valid only in the PWM mode, high-accuracy T-PWM mode, triangular wave PWM output mode, and PWM mode with dead time.

Table 10-18 TRnOPT1 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>4 to 0</td><td>TRnID[4:0]</td><td>Specifies the operation mode of register TR1CCR0
<table>
<tr><th>TRnID4</th><th>TRnID3</th><th>TRnID2</th><th>TRnID1</th><th>TRnID0</th><th>Interrupt Thinning Out Rate</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>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>
Caution: When during TMRn operation (TRnCTL0.TRnCE = 1) the TRnOPT1 register is written (including same value to bits TRnID[4:0]), the interrupt thinning out counter is cleared.

Note: Bits TRnID0 to TRnID4 are valid only in the PWM mode, high-accuracy T-PWM mode, triangular wave PWM mode, and PWM mode with dead time.</td></tr>
</table>

(10) TMRn option register 2 (TRnOPT2)

The TRnOPT2 register is an 8-bit register that controls A/D conversion trigger output (TRnADTRG0 signal).

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

Address TR0OPT2: $FFFFF588_H$
TR1OPT2: $FFFFF5C8_H$

Initial Value 00_H. This register is cleared by any reset.

TRnOPT2

7	6	5	4	3	2	1	0
0	0	TRnAT05	TRnAT04	TRnAT03	TRnAT02	TRnAT01	TRnAT00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Caution The settings of the TRnCCR5 and TRnCCR4 registers have an influence on the PWM output of pins TORn5 and TORn4 at the same time as the TRnADTRG0 signal output. Therefore, if setting bits TRnAT05 to TRnAT02, it is recommended to set the TRnOPT3 register as follows.

- In the triangular wave PWM mode, when setting TRnAT05 = 1, set TRnIOC3.TRnOE5 = 0.
- In the PWM mode and triangular wave PWM mode, when setting TRnAT04 = 1, set TRnIOC3.TRnOE5 = 0.
- In the triangular wave PWM mode, when setting TRnAT03 = 1, set TRnIOC3.TRnOE4 = 0
- In the PWM mode and the triangular wave PWM mode, when setting TRnAT02 = 1, set TRnIOC3.TRnOE4 = 0.

Table 10-19 TRnOPT2 register contents (1/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>5, 4</td><td>TRnAT05
TRnAT04</td><td>Controls the A/D converter trigger Signal (TRnADTRG0) generation with occurrence of the compare match interrupt (INTTRnCCR5).
<table>
<tr><th>TRnAT05</th><th>TRnAT04</th><th>A/D Converter Trigger Signal (TRnADTRG0) generation</th></tr>
<tr><td>0</td><td>0</td><td>No trigger signal is generated when INTTRnCCR5 occurs.</td></tr>
<tr><td>0</td><td>1</td><td>Trigger signal is generated, when INTTRnCCR5 occurs and TMRn is counting up.</td></tr>
<tr><td>1</td><td>0</td><td>Trigger signal is generated, when INTTRnCCR5 occurs and TMRn is counting down.</td></tr>
<tr><td>1</td><td>1</td><td>Trigger signal is generated, when INTTRnCCR5 occurs in any state (TMRn is counting up or down)</td></tr>
</table>
Caution: 1. Bit TRnAT05 can be set to 1 only in the triangular wave PWM mode and high-accuracy T-PWM mode. In all other modes, be sure to set this bit to 0.
2. Bit TRnAT04 can be set to 1 only in the PWM mode, triangular wave PWM mode, high-accuracy T-PWM mode, and PWM mode with dead time. In all other modes, be sure to set this bit to 0.</td></tr>
</table>

Table 10-19 TRnOPT2 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3, 2</td><td>TRnAT03
TRnAT02</td><td>Controls the A/D converter trigger Signal (TRnADTRG0) generation with occurrence of the compare match interrupt (INTTRnCCR4).
<table>
<tr><th>TRnAT03</th><th>TRnAT02</th><th>A/D Converter Trigger Signal (TRnADTRG0) generation</th></tr>
<tr><td>0</td><td>0</td><td>No trigger signal is generated when INTTRnCCR4 occurs.</td></tr>
<tr><td>0</td><td>1</td><td>Trigger signal is generated, when INTTRnCCR4 occurs and TMRn is counting up.</td></tr>
<tr><td>1</td><td>0</td><td>Trigger signal is generated, when INTTRnCCR4 occurs and TMRn is counting down.</td></tr>
<tr><td>1</td><td>1</td><td>Trigger signal is generated, when INTTRnCCR4 occurs in any state (TMRn is counting up or down)</td></tr>
</table>
Caution: 1. Bit TRnAT03 can be set to 1 only in the triangular wave PWM mode and high-accuracy T-PWM mode. In all other modes, be sure to set this bit to 0.
2. Bit TRnAT02 can be set to 1 only in the PWM mode, triangular wave PWM mode, high-accuracy T-PWM mode, and PWM mode with dead time. In all other modes, be sure to set this bit to 0.</td></tr>
<tr><td>1</td><td>TRnAT01</td><td>Controls the A/D converter trigger Signal (TRnADTRG0) generation with occurrence of the peak interrupt (INTTRnCD).
0: No trigger signal is generated when peak interrupt (INTTRnCD) occurs.
1: Trigger signal is generated when peak interrupt (INTTRnCD) occurs after thinning out.
Caution: Bit TRnAT01 can be set to 1 only in the PWM mode, high-accuracy T-PWM mode, and PWM mode with dead time. In all other modes, be sure to set this bit to 0.
Note: When bit TRnAT01 is set (1) the trigger signal coincides with the peak interrupt (INTTRnCD) controlled by the TRnOPT1 register (including thinning out).</td></tr>
<tr><td>0</td><td>TRnAT00</td><td>Controls the A/D converter trigger Signal (TRnADTRG0) generation with occurrence of the valley interrupt (INTTRnOD).
0: No trigger signal is generated when peak interrupt (INTTRnOD) occurs.
1: Trigger signal is generated when peak interrupt (INTTRnOD) occurs after thinning out.
Caution: Bit TRnAT00 can be set to 1 only in the high-accuracy T-PWM mode and triangular wave PWM mode. In all other modes, be sure to set this bit to 0.
Note: When bit TRnAT00 is set (1) the trigger signal coincides with the valley interrupt (INTTRnOD) controlled by the TRnOPT1 register (including thinning out).</td></tr>
</table>

(11) TMRn option register 3 (TRnOPT3)

The TRnOPT3 register is an 8-bit register that controls A/D conversion trigger output (signal TRnADTRG1).

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

Address TR0OPT3: $FFFFF589_H$
TR1OPT3: $FFFFF5C9_H$

Initial Value 00_H. This register is cleared by any reset.

TRnOPT3

7	6	5	4	3	2	1	0
0	0	TRnAT15	TRnAT14	TRnAT13	TRnAT12	TRnAT11	TRnAT10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Caution The settings of the TRnCCR5 and TRnCCR4 registers have an influence on the PWM outputs of pins TORn5, TORn4 at the same time as the TRnADTRG0 signal output. Therefore, if setting bits TRnAT15 to TRnAT12, it is recommended to set the TRnOPT3 register as follows.

- In the triangular wave PWM mode, when setting TRnAT15 = 1, set TRnIOC3.TRnOE5 = 0.
- In the PWM mode and triangular wave PWM mode, when setting TRnAT14 = 1, set TRnIOC3.TRnOE5 = 0.
- In the triangular wave PWM mode, when setting TRnAT13 = 1, set TRnIOC3.TRnOE4 = 0
- In the PWM mode and the triangular wave PWM mode, when setting TRnAT12 = 1, set TRnIOC3.TRnOE4 = 0.

Table 10-20 TRnOPT3 register contents (1/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>5, 4</td><td>TRnAT15
TRnAT14</td><td>Controls the A/D converter trigger Signal (TRNADTRG1) generation with occurrence of the compare match interrupt (INTTRnCCR5).
<table>
<tr><th>TRnAT15</th><th>TRnAT14</th><th>A/D Converter Trigger Signal (TRNADTRG1) generation</th></tr>
<tr><td>0</td><td>0</td><td>No trigger signal is generated when INTTRnCCR5 occurs.</td></tr>
<tr><td>0</td><td>1</td><td>Trigger signal is generated, when INTTRnCCR5 occurs and TMRn is counting up.</td></tr>
<tr><td>1</td><td>0</td><td>Trigger signal is generated, when INTTRnCCR5 occurs and TMRn is counting down.</td></tr>
<tr><td>1</td><td>1</td><td>Trigger signal is generated, when INTTRnCCR5 occurs in any state (TMRn is counting up or down)</td></tr>
</table>
Caution: 1. Bit TRnAT15 can be set to 1 only in the triangular wave PWM mode and high-accuracy T-PWM mode. In all other modes, be sure to set this bit to 0.
2. Bit TRnAT14 can be set to 1 only in the PWM mode, triangular wave PWM mode, high-accuracy T-PWM mode, and PWM mode with dead time. In all other modes, be sure to set this bit to 0.</td></tr>
</table>

Table 10-20 TRnOPT3 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3, 2</td><td>TRnAT13
TRnAT12</td><td>Controls the A/D converter trigger Signal (TRNADTRG1) generation with occurrence of the compare match interrupt (INTTRnCCR4).
<table>
<tr><th>TRnAT13</th><th>TRnAT12</th><th>A/D Converter Trigger Signal (TRNADTRG1) generation</th></tr>
<tr><td>0</td><td>0</td><td>No trigger signal is generated when INTTRnCCR4 occurs.</td></tr>
<tr><td>0</td><td>1</td><td>Trigger signal is generated, when INTTRnCCR4 occurs and TMRn is counting up.</td></tr>
<tr><td>1</td><td>0</td><td>Trigger signal is generated, when INTTRnCCR4 occurs and TMRn is counting down.</td></tr>
<tr><td>1</td><td>1</td><td>Trigger signal is generated, when INTTRnCCR4 occurs in any state (TMRn is counting up or down)</td></tr>
</table>
Caution: 1. Bit TRnAT13 can be set to 1 only in the triangular wave PWM mode and high-accuracy T-PWM mode. In all other modes, be sure to set this bit to 0.
2. Bit TRnAT12 can be set to 1 only in the PWM mode, triangular wave PWM mode, high-accuracy T-PWM mode, and PWM mode with dead time. In all other modes, be sure to set this bit to 0.</td></tr>
<tr><td>1</td><td>TRnAT11</td><td>Controls the A/D converter trigger Signal (TRNADTRG1) generation with occurrence of the peak interrupt (INTTRnCD).
0: No trigger signal is generated when peak interrupt (INTTRnCD) occurs.
1: Trigger signal is generated when peak interrupt (INTTRnCD) occurs after thinning out.
Caution: Bit TRnAT11 can be set to 1 only in the PWM mode, high-accuracy T-PWM mode, and PWM mode with dead time. In all other modes, be sure to set this bit to 0.
Note: When bit TRnAT11 is set (1) the trigger signal coincides with the peak interrupt (INTTRnCD) controlled by the TRnOPT1 register (including thinning out).</td></tr>
<tr><td>0</td><td>TRnAT10</td><td>Controls the A/D converter trigger Signal (TRNADTRG1) generation with occurrence of the valley interrupt (INTTRnOD).
0: No trigger signal is generated when peak interrupt (INTTRnOD) occurs.
1: Trigger signal is generated when peak interrupt (INTTRnOD) occurs after thinning out.
Caution: Bit TRnAT10 can be set to 1 only in the high-accuracy T-PWM mode and triangular wave PWM mode. In all other modes, be sure to set this bit to 0.
Note: When bit TRnAT10 is set (1) the trigger signal coincides with the valley interrupt (INTTRnOD) controlled by the TRnOPT1 register (including thinning out).</td></tr>
</table>

(12) TMRn option register 6 (TRnOPT6)

The TRnOPT6 register is an 8-bit register that controls the various flags of timer Rn. For a detailed description of the various flag functions, refer to *"Flags" on page 326*.

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

Address TR0OPT6: $FFFFF58C_H$
TR1OPT6: $FFFFF5CC_H$

Initial Value 00_H. This register is cleared by any reset, or when TRnCTL0.TRnCE = 0 is set.

	7	6	5	4	3	2	1	0
TRnOPT6	0	0	0	0	0	TRnTBF	TRnSUF	TRnRSF
	R	R	R/W	R/W	R/W	R/W	R	R/W

Table 10-21 TRnOPT6 register contents

Bit position	Bit name	Function
2	TRnTBF1	Indicates the true bar active detection. 0: Normal phase and inverted phase are not simultaneously active. 1: Normal phase and inverted phase are simultaneously active. This flag detects when the normal phase and inverted phase are simultaneously active, while any of the TRnIOC4.TRnTBA[2:0] bits is 1. When bits TRnTBA[2:0] = 000_B, a simultaneous active state is not detected. **Note:** The TRnTBF flag is set (1) upon detection that any of the normal phases (TORn1, TORn3, TORn5) and inverted phases (TORn2, TORn4, TORn6) are simultaneously active, and an error interrupt (INTTRnER) is output at such time.
1	TRnSUF	Indicates the TMRn sub-count direction. 0: Sub-counter is counting up. 1: Sub-counter is counting down. The TRnSUF flag detects sub-counter counting from 0000_H until (TRnCCR0 register value - 2) as up count, and counting from TRnCCR0 register value until 0002_H as down count. **Note:** **1.** The TRnSUF flag is a read-only flag. **2.** The TRnSUF flag is valid only in the high-accuracy T-PWM mode.
0	TRnRSF	Indicates the reload suspension. 0: Write access to TRnCCR0 to TRnCCR5 and TRnOPT1 registers is enabled (no reload request, or completion of reload). 1: Write access to TRnCCR0 to TRnCCR5 and TRnOPT1 registers is disabled (reload request was output). The TRnRSF flag indicates output of a reload request. It indicates that the data to be transferred next will be held in the TRnCCR0 to TRnCCR5 and TRnOPT1 registers. The TRnRSF flag is set (1) upon write to the TRnCCR1 register, and cleared (0) upon reload completion.

(13) TMRn option register 7 (TRnOPT7)

The TRnOPT7 register is an 8-bit register that controls timer output TORn0 switching.

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

Address TR0OPT7: $FFFFF58D_H$
TR1OPT7: $FFFFF5CD_H$

Initial Value 00_H. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
TRnOPT7	0	0	0	0	0	0	0	TRnTOS
	R	R	R	R	R	R	R	R/W

Table 10-22 TRnOPT7 register contents

Bit position	Bit name	Function
0	TRnRSF	Controls timer output (TORn0) switching. 0: Output up/down count state of the counter TRnCN to TORn0 pin. 1: Output up/down count state of the sub-counter TRnSBC to TORn0 pin. When TRnTOS = 0, the status of TRnOPT0.TRnCUF bit is output to pin TORn0 pin. When TRnTOS = 1, the status of TRnOPT6.TRnSUF bit is output to the TORn0 pin. **Note:** The TRnTOS bit is valid only in the high-accuracy T-PWM mode.

10.4 Basic Operation

10.4.1 Basic counter operation

This section describes the basic operation of the 16-bit counter. For details, refer to the description of the operation of each mode.

(1) Count start operation

The 16-bit counter of timer R starts counting from initial value FFFFH in all the modes except the high-accuracy T-PWM mode.

The counter counts up FFFFH, 0000H, 0001H, 0002H, 0003H, …

For details on the count operation refer to *"High-accuracy T-PWM mode" on page 378*.

(2) Clear operation

The 16-bit counter is cleared to 0000H upon a match between the 16-bit counter and the compare register. Counting immediately following the start of count operation and counting from FFFFH to 0000H in the case of overflow are not detected as clear operations.

(3) Overflow operation

16-bit counter overflow occurs when the value of the 16-bit counter changes from FFFFH to 0000H. When overflow occurs, bit TRnOVF of the TRnOPT0 register is set (to 1), and an interrupt (INTTRnOV) is output. No overflow interrupt (INTTRnOV) is output under the following conditions.

- Immediately after count operation start
- When compare value is matched and cleared at FFFFH

Caution Be sure to check that the overflow flag (TRnOVF) is set to 1 following output of the overflow interrupt (INTTRnOV).

(4) Counter read operation during count operation

In the case of timer R, the value of the 16-bit counter can be read by the TRnCNT register during count operation.

(5) Interrupt operation

In the case of timer R, the following interrupts are output.

- INTTRnCC0: Functions as TRnCCRn0 buffer register match interrupt.
- INTTRnCC1: Functions as TRnCCRn1 buffer register match interrupt.
- INTTRnCC2: Functions as TRnCCRn2 buffer register match interrupt.
- INTTRnCC3: Functions as TRnCCRn3 buffer register match interrupt.
- INTTRnCC4: Functions as TRnCCRn4 buffer register match interrupt.
- INTTRnCC5: Functions as TRnCCRn5 buffer register match interrupt.
- INTTRnCD: Functions as a peak interrupt at the timing when the counter switches from down count to up count.
- INTTRnOD: Functions as a valley interrupt at the timing when the counter switches from up count to down count.
- INTTRnOV: Functions as an overflow interrupt.
- INTTRnER: Functions as an normal phase/inverted phase simultaneous active detection interrupt.

10.4.2 Compare register rewrite operation

In the PWM mode, high-accuracy T-PWM mode, PWM mode with dead time, external trigger pulse output mode, and triangular wave PWM mode, the reload function is valid. (In all other modes, reload-related settings are invalid.)

The compare/control registers with the reload function are listed below.

– TRnCCR0 to TRnCCR5

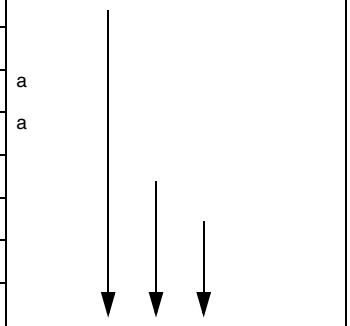
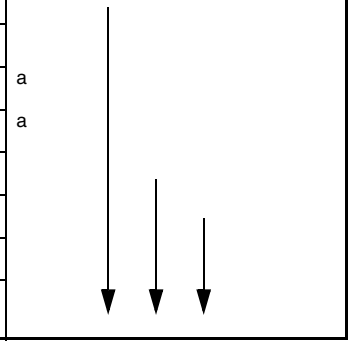
– TRnOPT1

Compare registers with the reload function can be rewritten in the following modes.

- **Anytime rewrite mode**
  In this mode, each compare register is updated independently, and when a compare register is written to, the register is updated to the value written during anytime write access.
- **Reload mode (batch rewrite)**
  When the TRnCCR1 register is written to, all the registers are updated at the next reload timing (reload). Reload does not occur even if a register other than the TRnCCR1 register is written to. A reload request flag (TRnRSF) is provided.

The compare register can be rewritten using DMA transfer. DMA transfer is performed as follows.

Table 10-23 DMA transfer of TMRn register in reload mode

Address	Register Name	DMA Transfer Sequence
FFFFF590H	TR0CCR5	
FFFFF592H	TR0CCR4	
FFFFF594H	-	a
FFFFF596H	-	a
FFFFF598H	TR0CCR0	
FFFFF59AH	TR0CCR3	
FFFFF59CH	TR0CCR2	
FFFFF59EH	TR0CCR1	
FFFFF5D0H	TR1CCR5	
FFFFF5D2H	TR1CCR4	
FFFFF5D4H	-	a
FFFFF5D6H	-	a
FFFFF5D8H	TR1CCR0	
FFFFF5DAH	TR1CCR3	
FFFFF5DCH	TR1CCR2	
FFFFF5DEH	TR1CCR1	

a) Dummy data transfer

For details about the interrupt thinning out function specified by setting the TRnOPT1 register, refer to *“Interrupt Thinning Out Function” on page 329*.

Table 10-24 Rewrite timing in each mode

Mode	Rewrite Timing
Interval mode	Anytime rewrite
External event count mode	Anytime rewrite
External trigger pulse output mode	Reload
One-shot pulse mode	Anytime rewrite
PWM mode	Reload
Free-running mode	Anytime rewrite
Pulse width measurement mode	Reload
Triangular wave PWM mode	Reload[a]
High-accuracy T-PWM mode	Anytime rewrite, Reload[b]
PWM mode with dead time	Reload

a) Rewrite is performed upon valley interrupt.
b) Set with TRnOPT0.TRnCMS = 0 and TRnOPT1.TRnRDE = 0.

(1) Anytime rewrite

Anytime rewrite is selected by setting bit TRnOPT0.TRnCMS = 1. The TRnOPT1.TRnRDE bit setting is ignored.

In this mode, the value written to each compare register is immediately transferred to the internal buffer register and compared to the counter value.

Following write to a compare register (TRnCCR0 register, etc.), the value is transferred to the internal buffer register after the delay of 4 clocks (f_{TMRn}). However, since only the TRnCCR1 register has a 2-stage configuration, the actual transfer timing is after the delay of 5 clocks (f_{TMRn}).

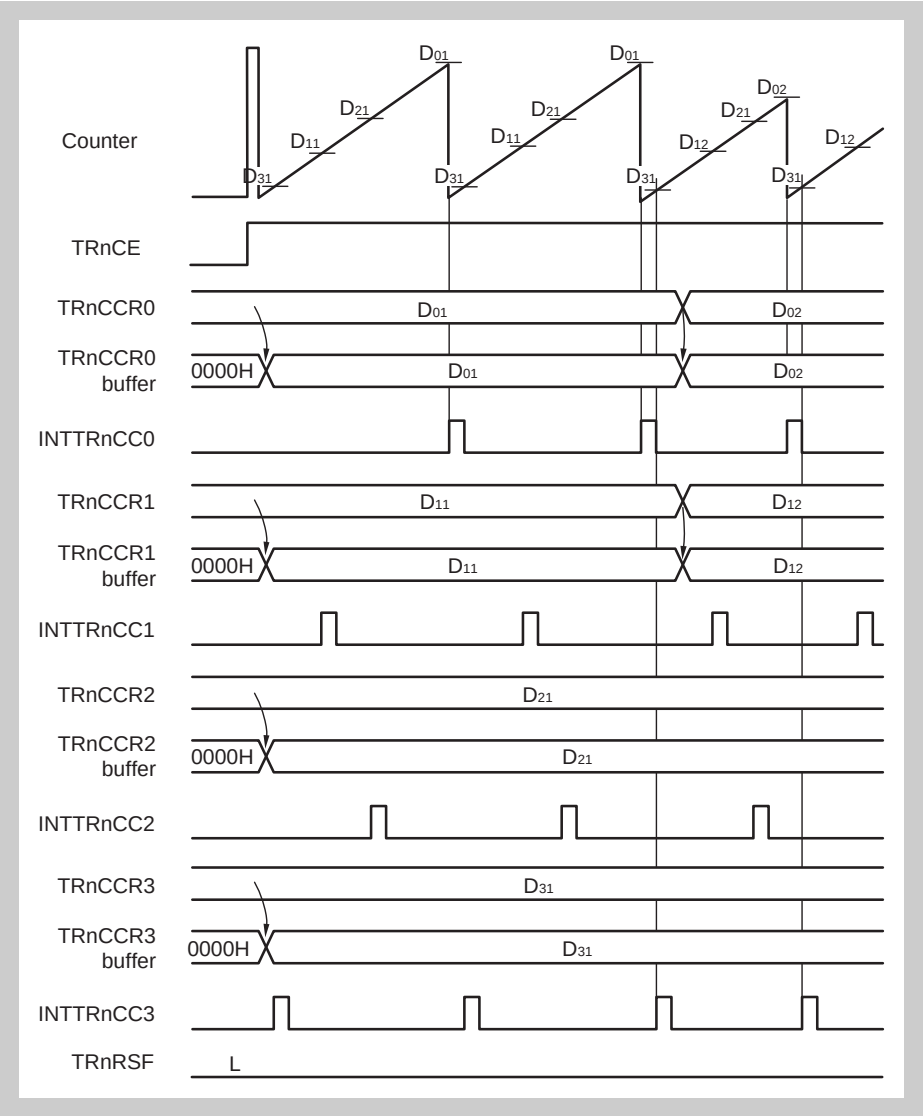


Figure 10-2 Anytime rewrite timing

Remarks

1. D01, D02: TRnCCR0 register setting value (0000H to FFFFH)
   D11, D12: TRnCCR1 register setting value (0000H to FFFFH)
   D21: TRnCCR2 register setting value (0000H to FFFFH)
   D31: TRnCCR3 register setting value (0000H to FFFFH)
2. Timing chart using interval timer mode as an example

(a) Cautions related to rewriting TRnCCR0 register in high-accuracy T-PWM mode

When the TRnCCR0 register is rewritten during operation using the anytime rewrite function, anytime transfer of the value to the TRnCCR0 buffer register is not performed. The timing is shown below.

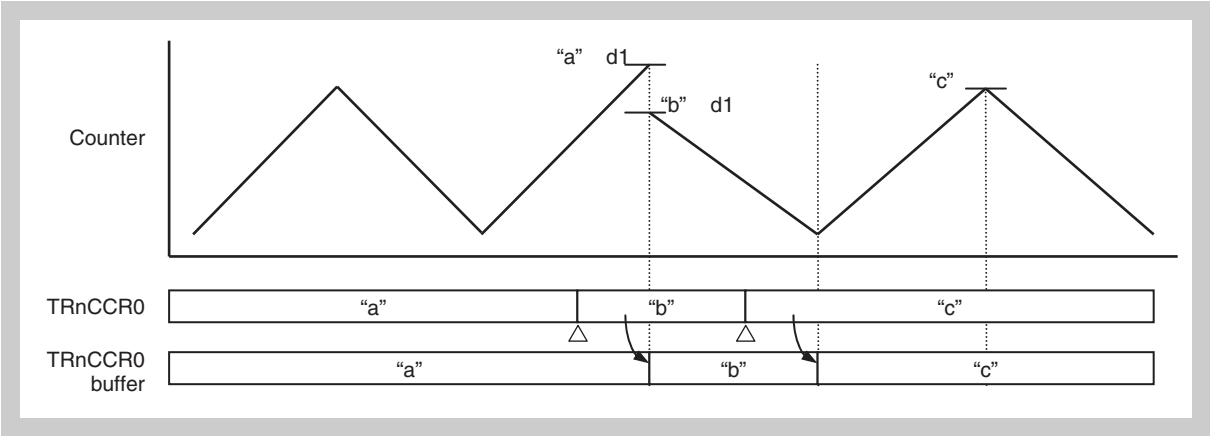


Figure 10-3 Anytime rewrite timing of TRnCCR0 in high-accurracy T-PWM mode

Remark d1: TRnDTC1 setting value

Following write to the TRnCCR0 register, the value of the TRnCCR0 register is transferred to the TRnCCR0 buffer register at the next peak or at the valley timing. Since TRnCMS = 1 (anytime rewrite), the settings of bits TRnIOE, TRnICE, TRnRDE, and TRnID4 to TRnID0 have no influence.

(b) Cautions related to rewriting of TRnCCR1 to TRnCCR3 registers

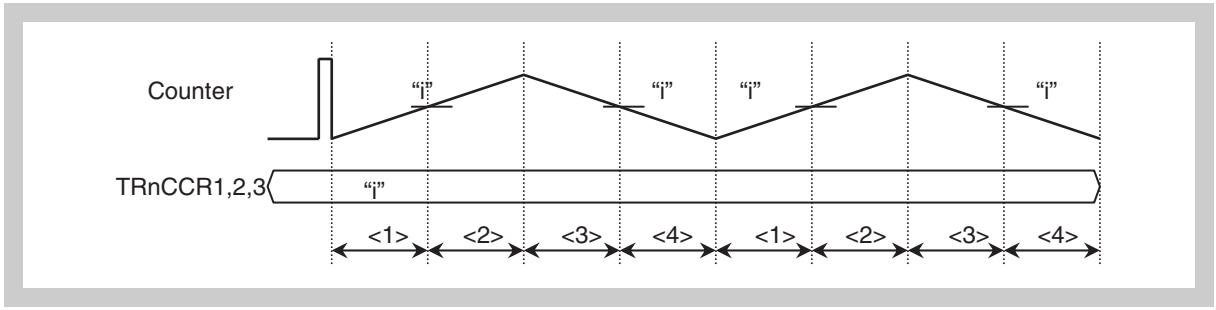


Figure 10-4 Anytime rewrite timing of TRnCCR1 to TRnCCR3 registers

Rewrite in <1> interval (rewrite before match occurrence)

In the case of rewrite before a match between the TRnCCR1 to TRnCCR3 registers and the counter occurs, a match with the counter occurs following rewrite and the rewrite value is instantly reflected.

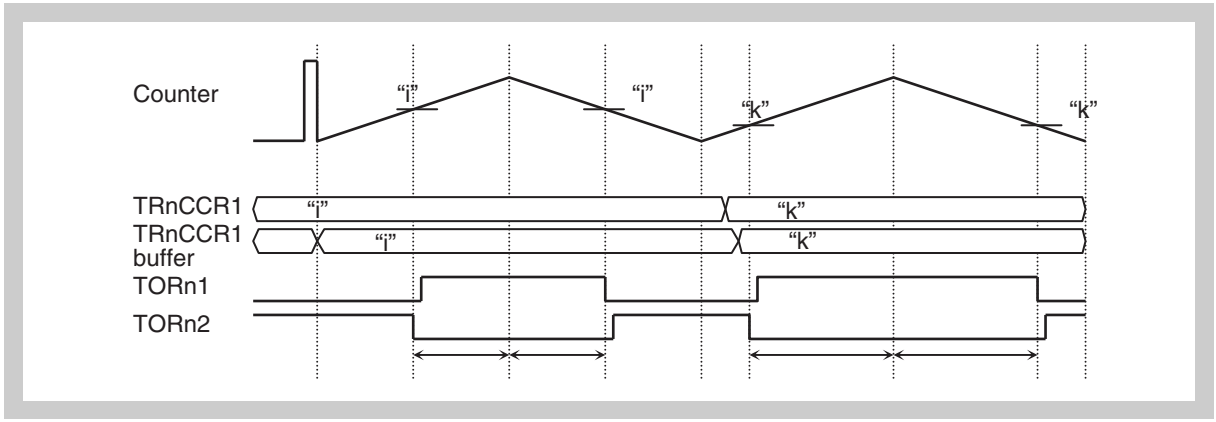


Figure 10-5 Anytime rewrite timing in <1> interval (rewrite before match occurrence) and (counter value < rewrite value)

If a value smaller than the counter value is written before match occurrence, no match occurs, so the following output wave results.

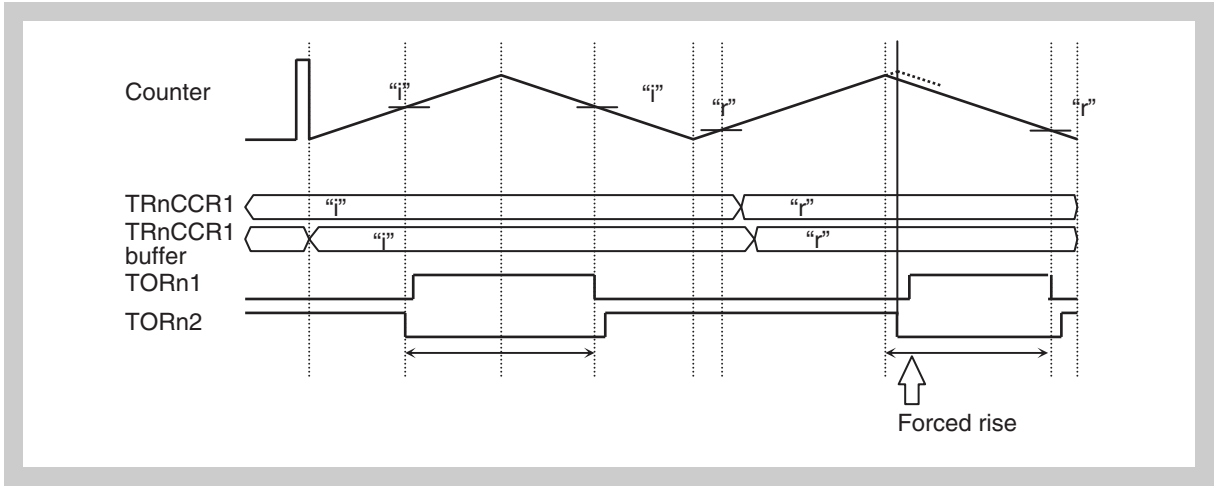


Figure 10-6 Anytime rewrite timing in <1> interval (rewrite before match occurrence) and (counter value > rewrite value)

If no match occurs, the timer output remains unchanged.

However, even if a match does not occur the timer output is forcibly changed to normal phase active level at peaks.

Rewrite in <2> interval (rewrite after match occurrence)

In the case of rewrite after a match between the TRnCCR1 to TRnCCR3 registers and the counter occurs, further match occurrences are ignored, so the rewrite value is not reflected.

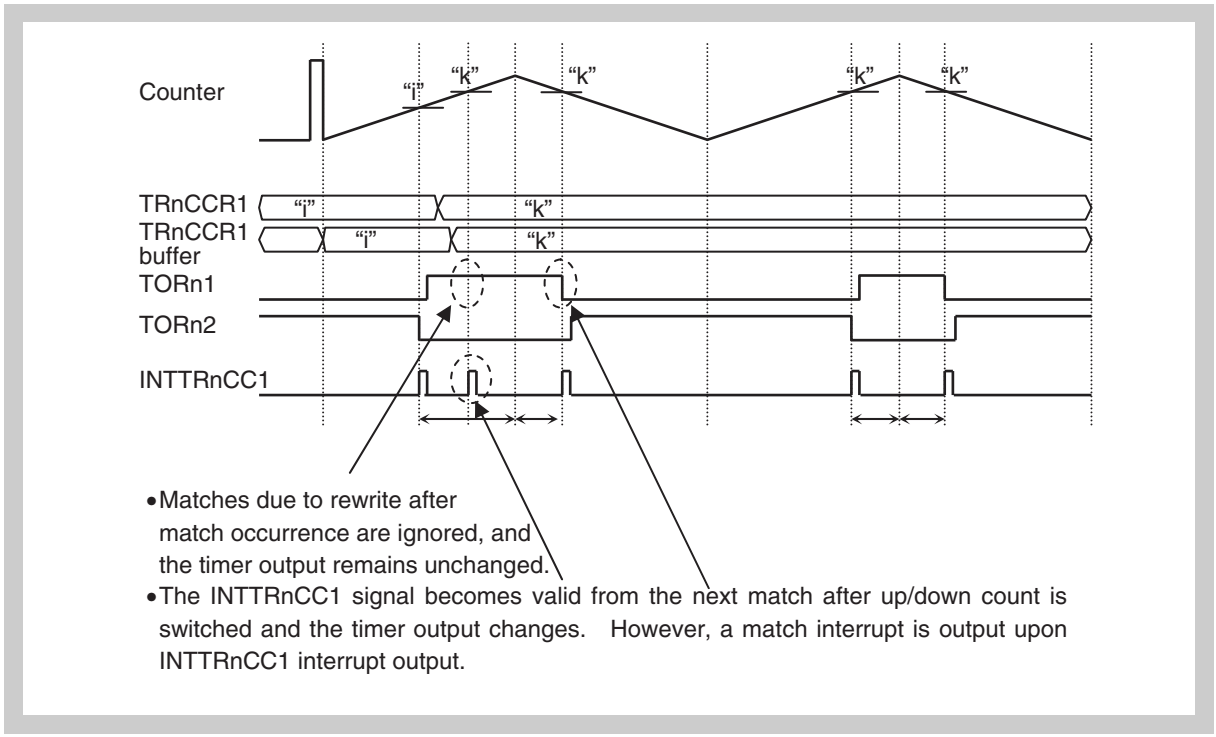


Figure 10-7 Anytime rewrite timing in <2> interval (rewrite after match occurrence)

Rewrite in <3> interval (rewrite before match occurrence)

In the case of rewrite before a match between the TRnCCR1 to TRnCCR3 registers and the counter occurs, a match with the counter occurs following rewrite and the rewrite value is instantly reflected.

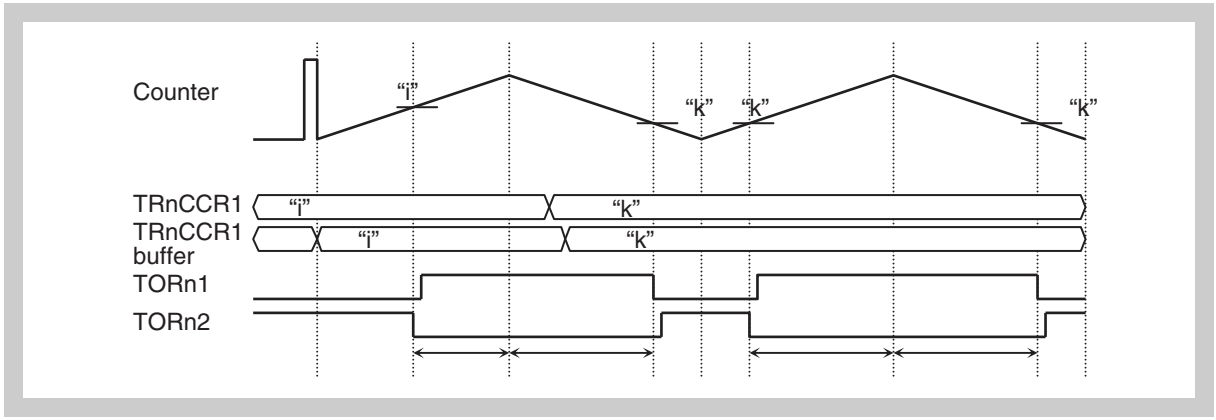


Figure 10-8 Anytime rewrite timing in <3> interval (rewrite before match occurrence) and (counter value > rewrite value)

If a value larger than the counter value is written before match occurrence, no match occurs, so the following output wave results.

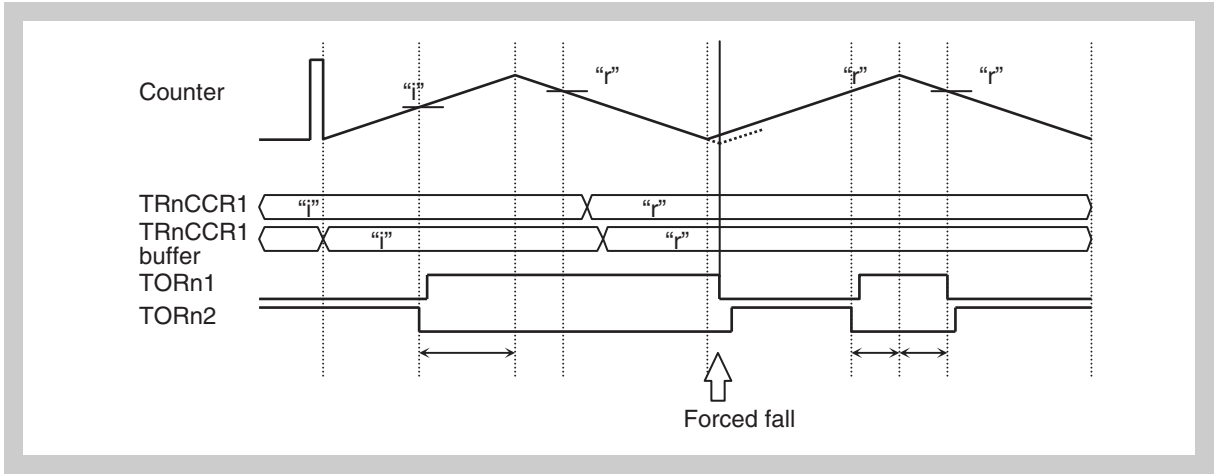


Figure 10-9 Anytime rewrite timing in <3> interval (rewrite before match occurrence) and (counter value < rewrite value)

If no match occurs, the timer output remains unchanged.

However, even if a match occurs, the timer output is forcibly changed to normal phase inactive level at valleys.

Rewrite in <4> interval (rewrite after match occurrence)

In the case of rewrite after a match between the TRnCCR1 to TRnCCR3 registers and the counter occurs, further match occurrences are ignored, so the rewrite value is not reflected.

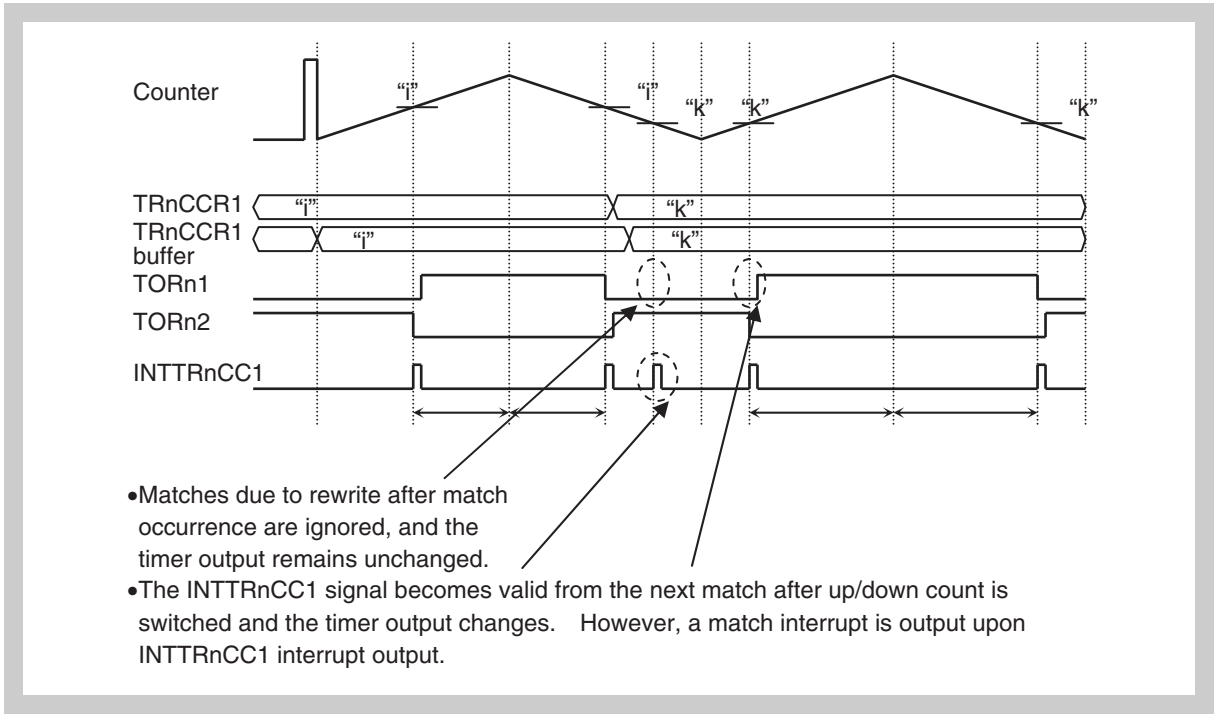


Figure 10-10 Anytime rewrite timing in <4> interval (rewrite after match occurrence)

(c) Cautions related to rewriting TRnOPT1

Since the internal interrupt thinning out counter is cleared when the TRnOPT1 register is written to, the interrupt output interval may temporarily become longer.

(2) Batch rewrite (reload mode)

Batch rewrite is selected by setting bits TRnOPT0.TRnCMS = 0, TRnOPT1.TRnRDE = 0, TRnOPT1.TRnICE = 1 (reload enabled at peaks), and TRnOPT1.TRnIOE = 1 (reload enabled at valleys).

In this mode, the values written to the various compare registers are all transferred at the same time to the respective buffer registers at the reload timing.

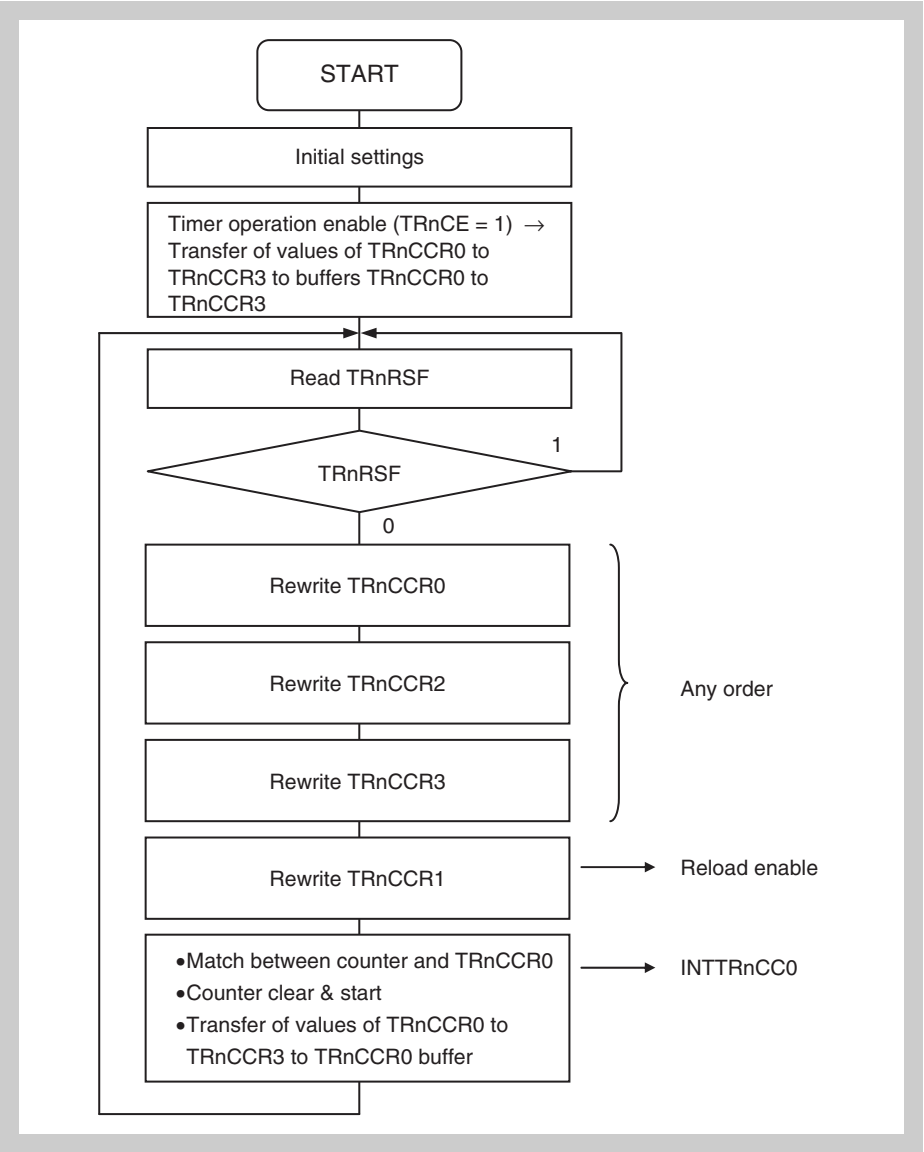


Figure 10-11 Basic operation flow during batch rewrite

Caution Write access to the TRnCCR1 register includes also the reload enable operation. Therefore, rewrite the TRnCCR1 register after rewriting the other TRnCCR registers.

Remark In the sample flow chart in *Figure 10-11* the PWM mode is assumed.

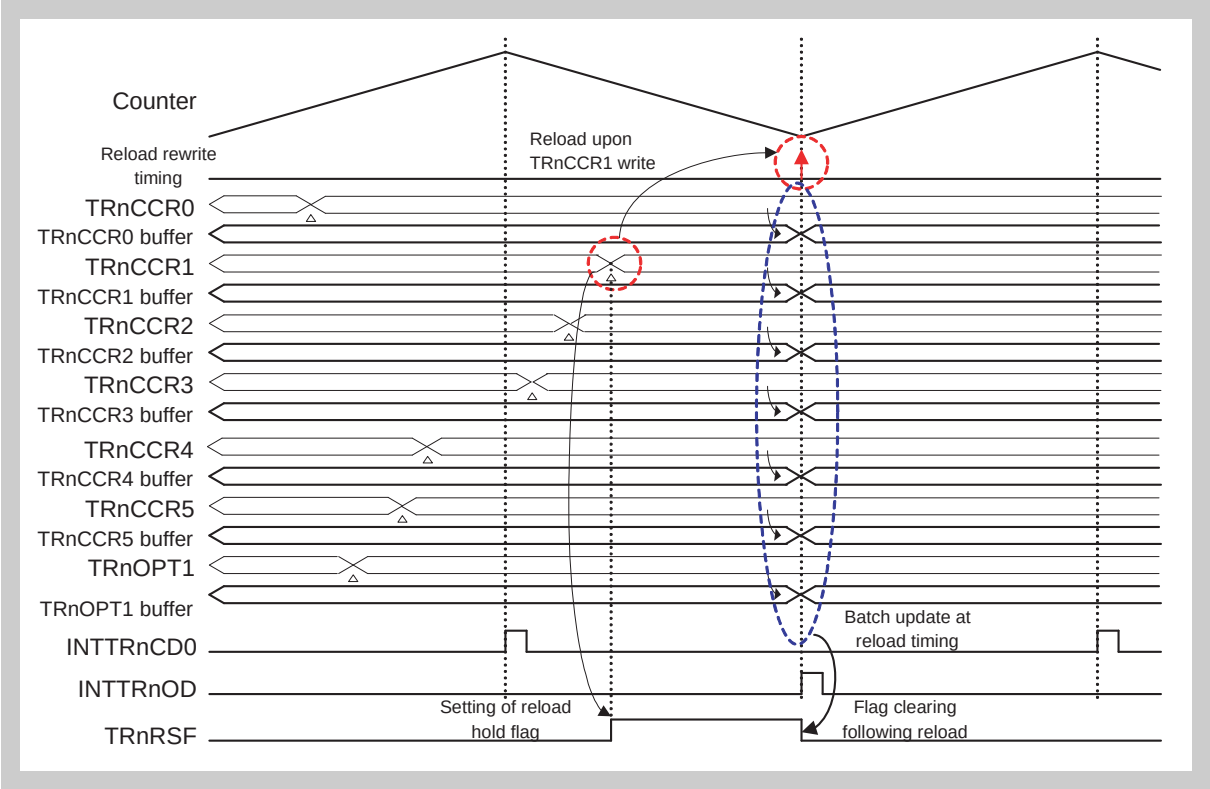


Figure 10-12 Batch rewrite timing (1/2)

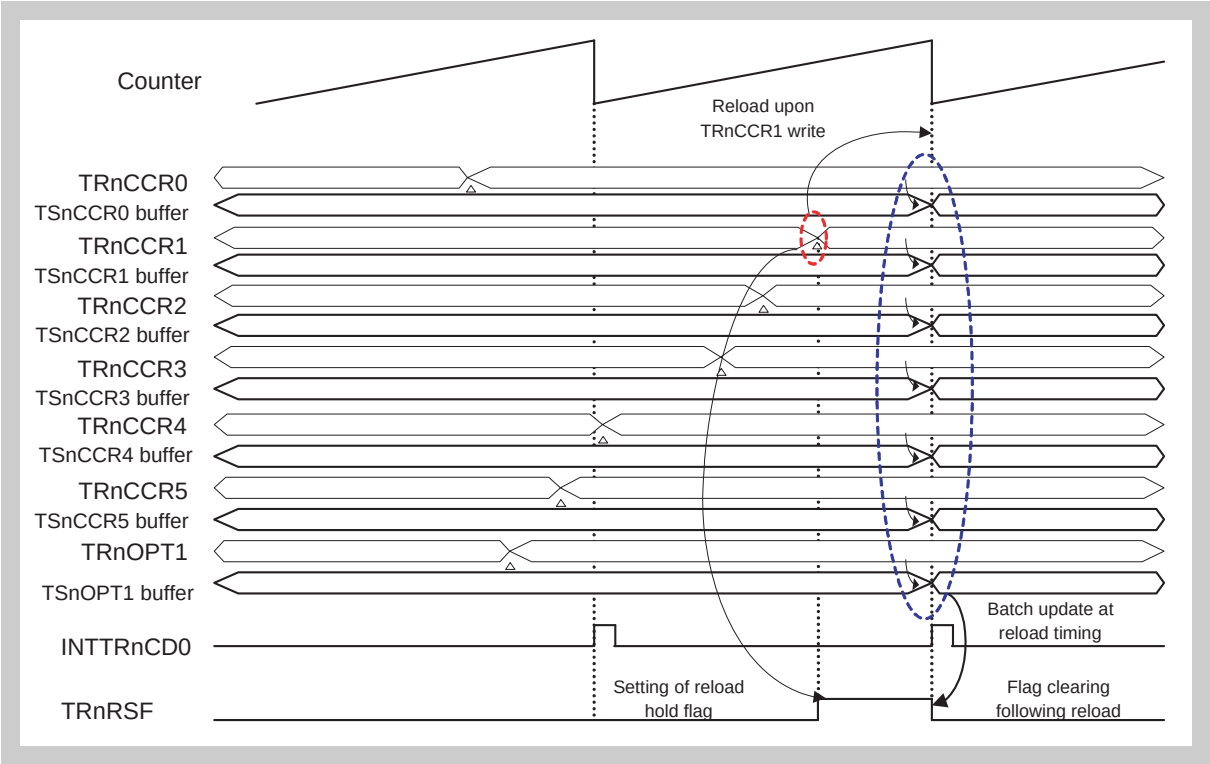


Figure 10-13 Batch rewrite timing (2/2)

(a) TRnCCR0 register rewrite operation in high-accuracy T-PWM mode

When rewriting the TRnCCR0 register in the batch rewrite mode, the output waveform changes according to whether reload occurs at a peak or at a valley (TRnICE = 1, TRnIOE = 1 settings).

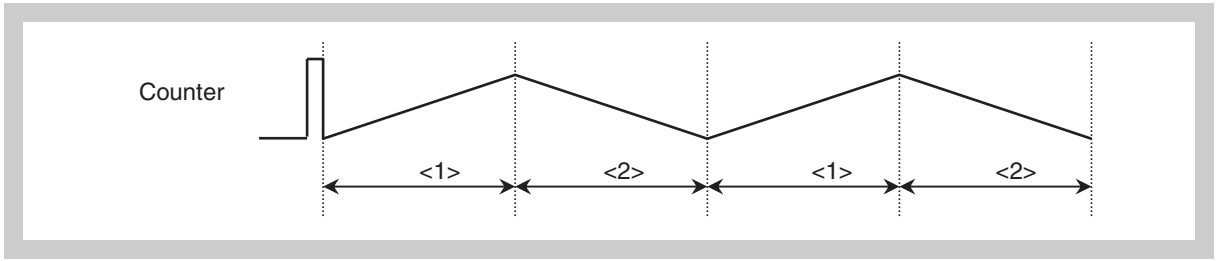


Figure 10-14 Batch rewrite timing of TRnCCR0 register in high-accuracy T-PWM mode

Rewrite in <1> interval (rewrite during up count)

Since the next reload timing becomes the peak point, the cycle on the down count side changes and an asymmetrical triangular waveform is output. Also, since the cycle changes, reset the duty value as necessary.

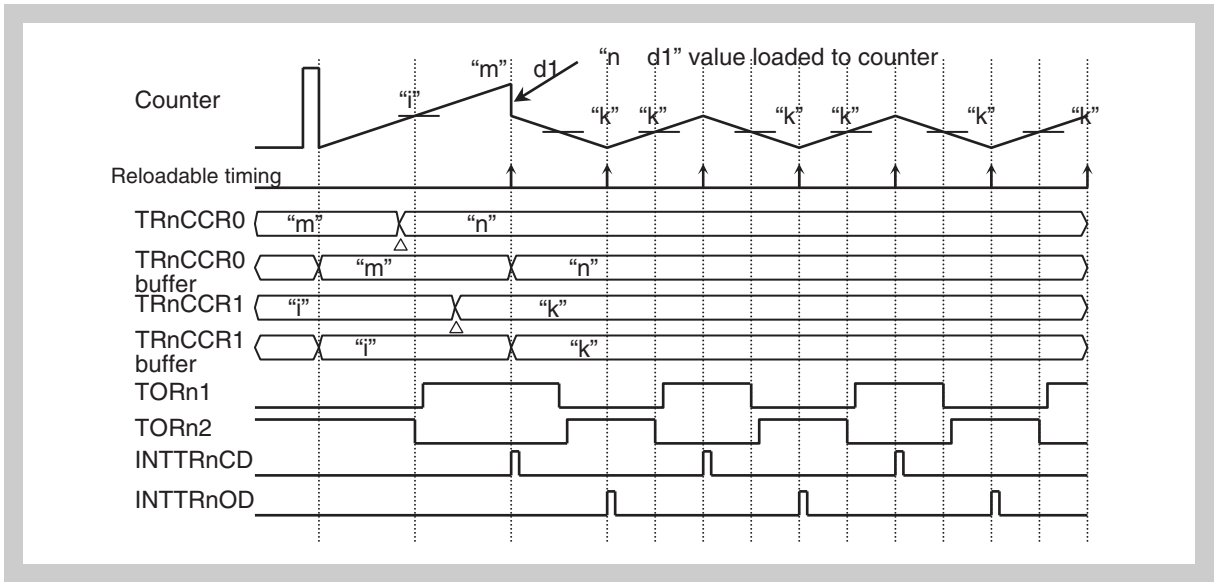


Figure 10-15 Batch rewrite timing in <1> interval (rewrite during up count) (1/2)

Remark d1: TRnDTC1 setting value

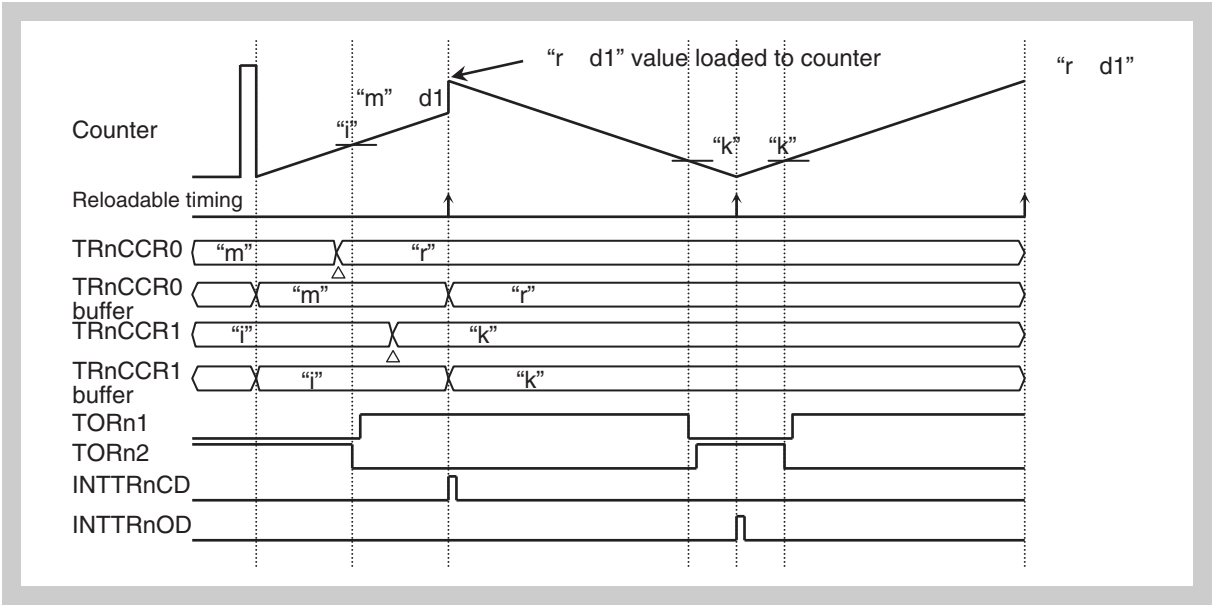


Figure 10-16 Batch rewrite timing in <1> interval (rewrite during up count) (2/2)

Remark d1: TRnDTC1 setting value

The counter loads the TRnCCR0 value minus "d1" upon occurrence of reload in the high-accuracy T-PWM mode. As a result, the expected waveform can be output even if the cycle value is changed at the peak reload timing.

Rewrite in <2> interval (rewrite during down count)

Since the next reload timing becomes the valley point, the cycle value changes from the next cycle and the asymmetrical triangular waveform output is held. Since the cycle changes, be sure to set again the duty value as required.

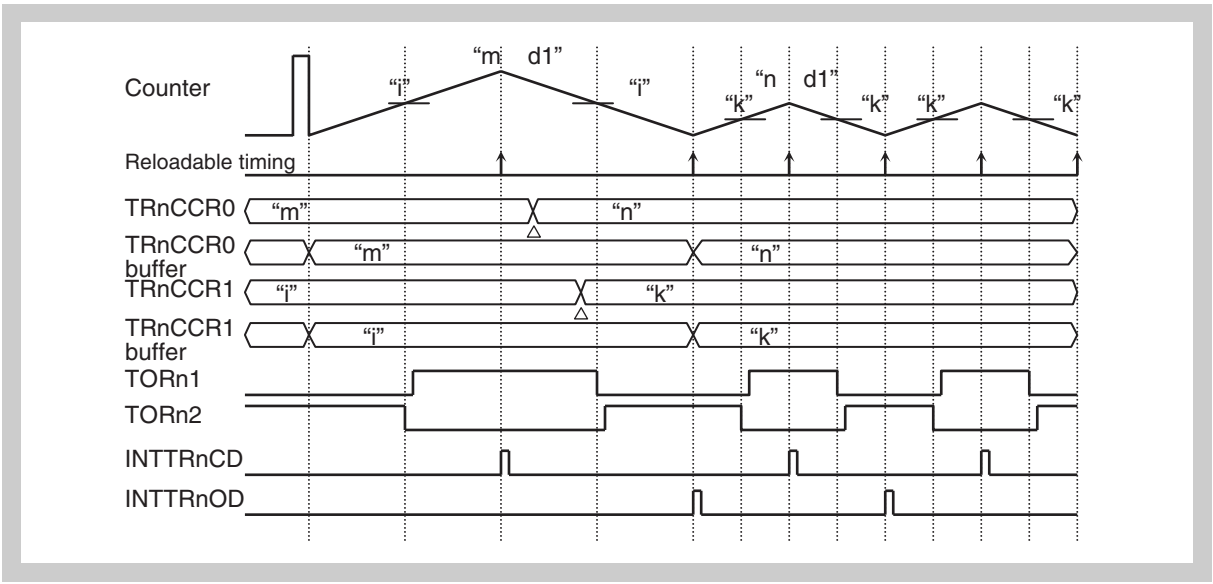


Figure 10-17 Batch rewrite timing in <2> interval (rewrite during down count)

Remark d1: TRnDTC1 setting value

(b) TRnCCR1 to TRnCCR3 register rewrite operation in high-accuracy T-PWM mode

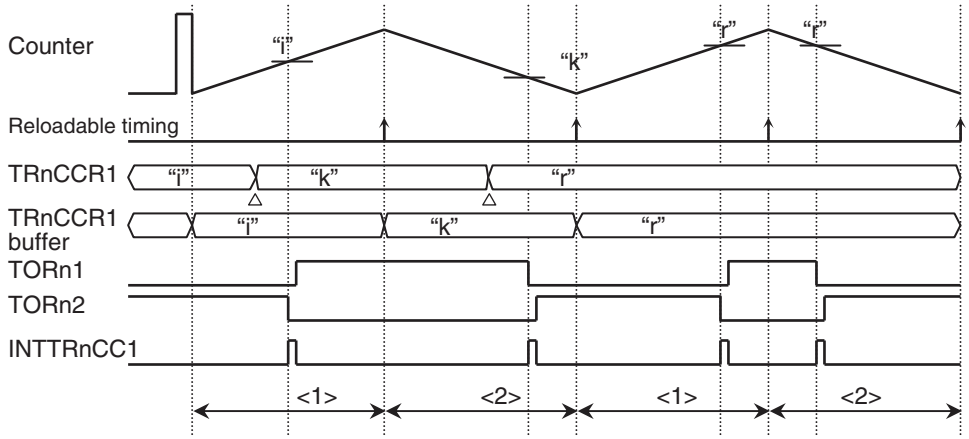


Figure 10-18 Batch rewrite timing of TRnCCR1 to TRnCCR3 registers in high-accuracy T-PWM mode

Remark TRnDTC0 = 0, TRnDTC1 = 0

Rewrite in <1> interval (rewrite during up count)

Since reload is performed at the peak interrupt timing, an asymmetric triangular waveform is output.

Rewrite in <2> interval (rewrite during down count)

Since reload is performed at the valley interrupt timing, an asymmetric triangular waveform is output.

10.4.3 List of outputs in each mode

(1) Timer outputs in each mode

The timer outputs (pins TORn0 to TORn7) in each mode are listed below.

Table 10-25 List of Timer Outputs in Each Mode (1/2)

Operation Mode	TORn0	TORn1	TORn2	TORn3
Interval mode	Toggle output upon TRnCCR0 compare match	Toggle output upon TRnCCR1 compare match	Toggle output upon TRnCCR2 compare match	Toggle output upon TRnCCR3 compare match
External event count mode	Toggle output upon TRnCCR0 compare match	Toggle output upon TRnCCR1 compare match	Toggle output upon TRnCCR2 compare match	Toggle output upon TRnCCR3 compare match
External trigger pulse output mode	Toggle output upon CCR0 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 upon TRnCCR0 match.	Active upon TRnCCR1 match. Inactive upon TRnCCR0 match.	Active upon TRnCCR2 match. Inactive upon TRnCCR0 match.	Active upon TRnCCR3 match. Inactive upon TRnCCR0 match.
PWM mode	Toggle output upon TRnCCR0 compare match	PWM output upon TRnCCR1 compare match	PWM output upon TRnCCR2 compare match	PWM output upon TRnCCR3 compare match
Free-running mode	Toggle output upon TRnCCR0 compare match	Toggle output upon TRnCCR1 compare match	Toggle output upon TRnCCR2 compare match	Toggle output upon TRnCCR2 compare match
Pulse width measurement mode	-	-	-	-
Triangular wave PWM mode	Inactive during up count. Active during down count.	PWM output upon TRnCCR1 compare match	PWM output upon TRnCCR2 compare match	PWM output upon TRnCCR3 compare match
High-accuracy T-PWM mode	Inactive during counter or sub-counter up count. Active during down count.	PWM output (with dead time) upon TRnCCR1 compare match	Inverted phase output to TORn1	PWM output (with dead time) upon TRnCCR2 compare match
PWM mode with dead time	Toggle output upon TRnCCR0 compare match	PWM output (with dead time) upon TRnCCR1 compare match	Inverted phase output to TORn1	PWM output (with dead time) upon TRnCCR2 compare match

Table 10-25: List of Timer Outputs in Each Mode (2/2)

Operation Mode	TORn4	TORn5	TORn6	TORn7
Interval mode	Toggle output upon TRnCCR4 compare match	Toggle output upon TRnCCR5 compare match	-	-
External event count mode	Toggle output upon TRnCCR4 compare match	Toggle output upon TRnCCR5 compare match	-	-
External trigger pulse output mode	External trigger pulse waveform output	External trigger pulse waveform output	-	-
One-shot pulse mode	High upon TRnCCR4 match. Inactive upon TRnCCR0 match.	High upon TRnCCR5 match. Inactive upon TRnCCR0 match.	-	-
PWM mode	PWM output upon TRnCCR4 compare match	PWM output upon TRnCCR5 compare match	-	Pulse output upon A/D conversion trigger[a]
Free-running mode	Toggle output upon TRnCCR4 compare match	Toggle output upon TRnCCR5 compare match	-	-
Pulse width measurement mode	-	-	-	-
Triangular wave PWM mode	PWM output upon TRnCCR4 compare match	PWM output upon TRnCCR5 compare match	-	Pulse output upon A/D conversion trigger[a]
High-accuracy T-PWM mode	Inverted phase output to TORn3	PWM output (with dead time) upon TRnCCR3 compare match	Inverted phase output to TORn5	Pulse output upon A/D conversion trigger[a]
PWM mode with dead time	Inverted phase output to TORn3	PWM output (with dead time) upon TRnCCR3 compare match	Inverted phase output to TORn5	Pulse output upon A/D conversion trigger[a]

a) For details on TORn7, refer to *"TORn7 pin output control" on page 318*.

(a) TORn7 pin output control

The A/D conversion signals can be output to pin TORn7. Pin TORn7 is set (to 1) by the TRnADTRG0 signal trigger, and it is reset (to 0) by the TRnADTRG1 signal trigger. If the TRnADTRG0 trigger occurs while pin TORn7 is set (to 1), its set (1) status is maintained. If the TRnADTRG1 trigger occurs while pin TORn7 is reset (0), the (0) status is maintained. If the TRnADTRG0 and TRnADTRG1 signal triggers occur simultaneously, pin TORn7 is reset (to 0).

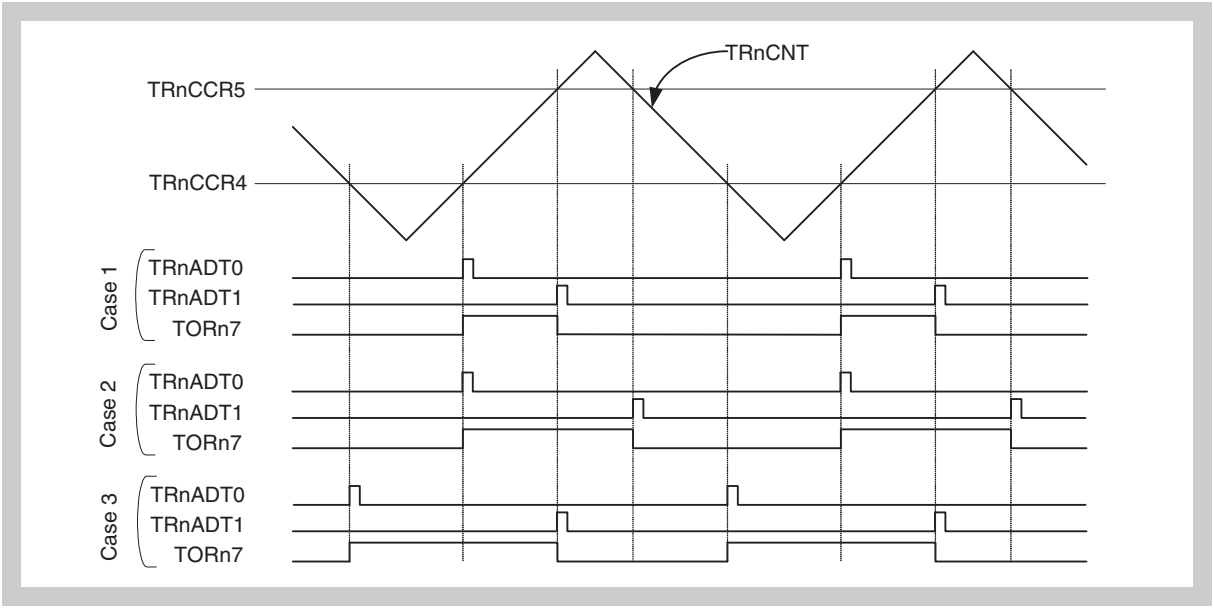


Figure 10-19 TORn7 pin output timing

Remark Case 1: When TRnCCR4 < TRnCCR5, TRnOPT2 = 04H, TRnOPT3 = 10H
Case 2: When TRnCCR4 < TRnCCR5, TRnOPT2 = 04H, TRnOPT3 = 20H
Case 3: When TRnCCR4 < TRnCCR5, TRnOPT2 = 08H, TRnOPT3 = 10H

(2) Interrupts in each mode

The interrupts in each mode (INTTRnCC0 to INTTRnCC5, INTTRnOV, INTTRnER) are listed below.

Table 10-26 List of Interrupts in Each Mode (1/2)

Operation Mode	INTTRnCC0	INTTRnCC1	INTTRnCC2	INTTRnCC3
Interval mode	TRnCCR0 compare match interrupt	TRnCCR1 compare match interrupt	TRnCCR2 compare match interrupt	TRnCCR3 compare match interrupt
External event count mode	TRnCCR0 compare match interrupt	TRnCCR1 compare match interrupt	TRnCCR2 compare match interrupt	TRnCCR3 compare match interrupt
External trigger pulse output mode	TRnCCR0 compare match interrupt	TRnCCR1 compare match interrupt	TRnCCR2 compare match interrupt	TRnCCR3 compare match interrupt
One-shot pulse mode	TRnCCR0 compare match interrupt	TRnCCR1 compare match interrupt	TRnCCR2 compare match interrupt	TRnCCR3 compare match interrupt
PWM mode	TRnCCR0 compare match interrupt	TRnCCR1 compare match interrupt	TRnCCR2 compare match interrupt	TRnCCR3 compare match interrupt
Free-running mode	TRnCCR0 compare match interrupt	TRnCCR1 compare match interrupt	TRnCCR2 compare match interrupt	TRnCCR3 compare match interrupt
Pulse width measurement mode	TIR10 capture interrupt	TIR11 capture interrupt	TIR12 capture interrupt	TIR13 capture interrupt
Triangular wave PWM mode	TIR10 capture interrupt	TIR11 capture interrupt	TIR12 capture interrupt	TIR13 capture interrupt
High-accuracy T-PWM mode	TRnCCR0 compare match interrupt[a]	TRnCCR1 compare match interrupt[b]	TRnCCR2 compare match interrupt[b]	TRnCCR3 compare match interrupt[b]
PWM mode with dead time	TRnCCR0 compare match interrupt	TRnCCR1 compare match interrupt[c]	TRnCCR2 compare match interrupt[c]	TRnCCR3 compare match interrupt[c]

a) A compare match interrupt is output when the TRnDTC1 register is set to 000_H. INTTRnCD can be used as the peak interrupt.

b) If set in the range of $0000_H \leq$ TRnCCRm $<$ TRnDTC0, (TRnCCR0 - TRnDTC1)$<$ TRnCCRm $\leq$ TRnCCR0 (m = 1 to 5), no compare match interrupt is output.

c) If set to TRnCCR0 $<$ TRnCCRm (m = 1 to 5), no compare match interrupt is output.

Table 10-26: List of Interrupts in Each Mode (2/2)

Operation Mode	INTTRnCC4	INTTRnCC5	INTTRnOV	INTTRnER
Interval mode	TRnCCR4 compare match interrupt	TRnCCR5 compare match interrupt	-	-
External event count mode	TRnCCR4 compare match interrupt	TRnCCR5 compare match interrupt	-	-
External trigger pulse output mode	TRnCCR4 compare match interrupt	TRnCCR5 compare match interrupt	-	-
One-shot pulse mode	TRnCCR4 compare match interrupt	TRnCCR5 compare match interrupt	-	-
PWM mode	TRnCCR4 compare match interrupt	TRnCCR5 compare match interrupt	-	Error interrupt
Free-running mode	TRnCCR4 compare match interrupt	TRnCCR5 compare match interrupt	Overflow interrupt	-
Pulse width measurement mode	-	-	Overflow interrupt	-
Triangular wave PWM mode	TRnCCR4 compare match interrupt	TRnCCR5 compare match interrupt	-	Error interrupt
High-accuracy T-PWM mode	TRnCCR4 compare match interrupt[a]	TRnCCR5 compare match interrupt[a]	Overflow interrupt[b]	Error interrupt
PWM mode with dead time	TRnCCR4 compare match interrupt[c]	TRnCCR5 compare match interrupt [c]	-	Error interrupt

a) If set in the range of $0000_H \leq$ TRnCCRm $<$ TRnDTC0, (TRnCCR0 - TRnDTC1) $<$ TRnCCRm $\leq$ TRnCCR0 (m = 1 to 5), no compare match interrupt is output.

b) If a setting error has been made for TRnCCR0, TRnDTC0, TRnDTC1, an overflow interrupt (INTTRnOV) is output.

c) If set to TRnCCR0 $<$ TRnCCRm (m = 1 to 5), no compare match interrupt is output.

Remark "-" in the table indicates inactive level output.

(3) A/D conversion triggers, peak interrupts, and valley interrupts in each mode

The A/D conversion triggers, peak interrupts, and valley interrupts in each mode are listed below.

Table 10-27 List of A/D Conversion Triggers, Peak Interrupts and Valley Interrupts in Each Mode

Operation Mode	TRnADTRG0	TRnADTRG1	INTTRnCD	INTTRnOD
Interval mode	-	-	-	-
External event count mode	-	-	-	-
External trigger pulse output mode	-	-	-	-
One-shot pulse mode	-	-	-	-
PWM mode	Select from interrupts INTTRnCD, INTTRnCC4, INTTRnCC5	Select from interrupts INTTRnCD, INTTRnCC4, INTTRnCC5	Peak interrupt at same timing as INTTRnCC0 interrupt	-
Free-running mode	-	-	-	-
Pulse width measurement mode	-	-	-	-
Triangular wave PWM mode	Select from interrupts INTTRnCD, INTTRnCC4, INTTRnCC5	Select from interrupts INTTRnCD, INTTRnCC4, INTTRnCC5	-	Valley interrupt at counter valley (upon switching from down to up count)
High-accuracy T-PWM mode	Select from interrupts INTTRnCD, INTTRnCC4, INTTRnCC5	Select from interrupts INTTRnCD, INTTRnCC4, INTTRnCC5	Peak interrupt	Valley interrupt at counter valley (upon switching from down to up count)
PWM mode with dead time	Select from interrupts INTTRnCD, INTTRnCC4, INTTRnCC5	Select from interrupts INTTRnCD, INTTRnCC4, INTTRnCC5	Peak interrupt at same timing as INTTRnCC0 interrupt	-

Remark The INTTRnCD interrupt and INTTRnOD interrupt are the occurrence conditions following interrupt thinning out.

10.5 Match Interrupts

Match interrupts consist of compare match interrupts (INTTRnCC0 to INTTRnCC5), peak interrupts (INTTRnCD), and valley interrupts (INTTRnOD). For details about error interrupts, refer to *"Error Interrupts" on page 338*.

Compare match interrupts (INTTRnCC0 to INTTRnCC5) are interrupts that occur following a match between the TRnCCR0 to TRnCCR5 registers and the counter, and are output in all modes (no operation mode restrictions).

Peak interrupts (INTTRnCD) are output in the PWM mode, triangular wave PWM mode, high-accuracy T-PWM mode, and PWM mode with dead time. If the counter is a triangular wave operation mode (triangular wave PWM mode, high-accuracy PWM mode), a peak interrupt is output when the counter switches from up count to down count. If the counter is in a saw tooth wave operation mode (PWM mode, PWM mode with dead time), a peak interrupt occurs upon a match between the counter and the TRnCCR0 register (same timing as INTTRnCC0 interrupt).

Valley interrupts occur when the counter switches from down count to up count in the triangular wave PWM mode and high-accuracy T-PWM mode.

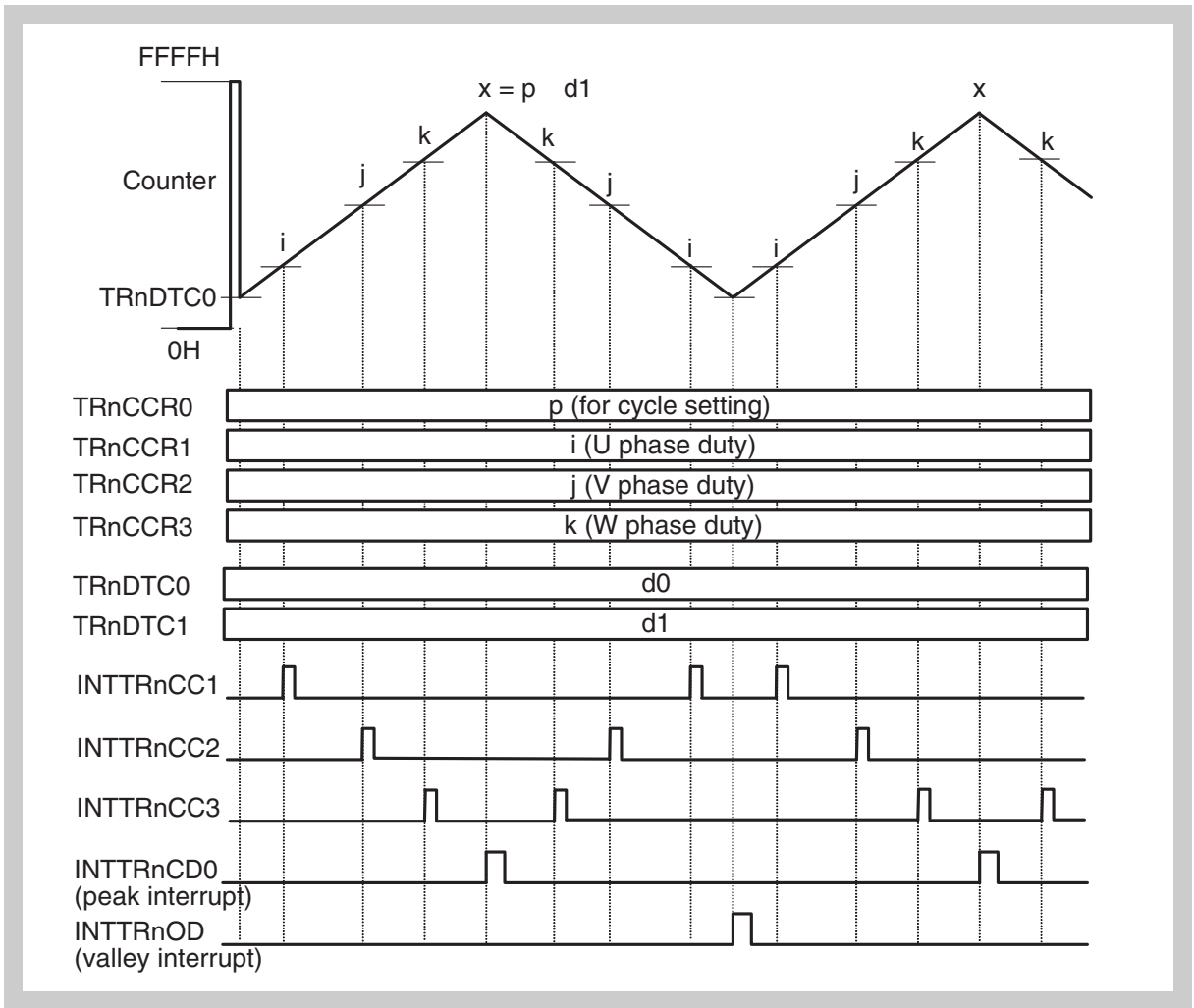


Figure 10-20 Interrupt signal output example (1/2)

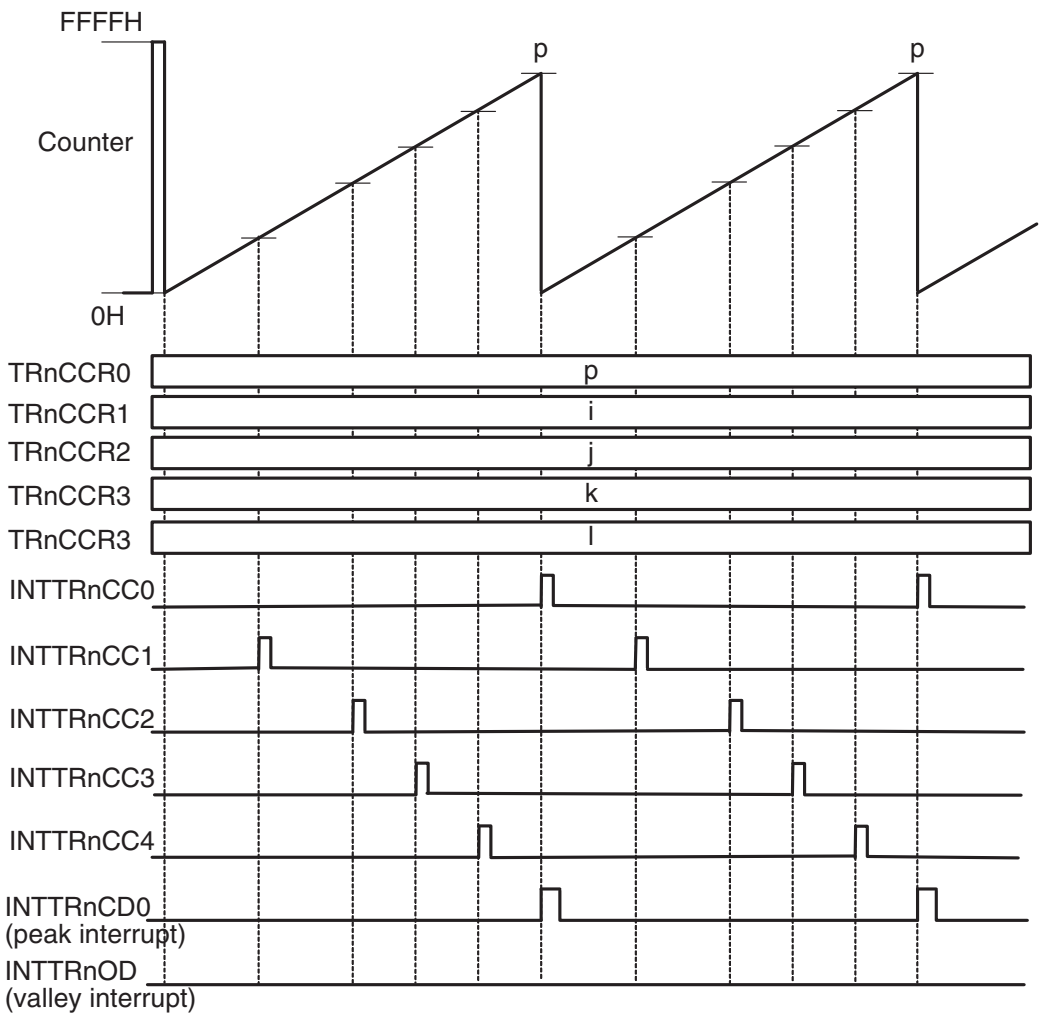


Figure 10-21 Interrupt signal output example (2/2)

10.5.1 Compare match interrupt related cautions

(1) Cautions in high-accuracy T-PWM mode

Compare match interrupts occur upon a match between the counter and a compare register (TRnCCR0 to TRnCCR5). However, in the high-accuracy T-PWM mode, the compare register can be set exceeding the counter's count operation range. Therefore, under the following conditions, no compare interrupt is output.

(a) Restrictions related to compare match interrupt with TRnCCR0 register

(INTTRnCC0) In the high-accuracy T-PWM mode, when TRnDTC1 ≠ 000H, no compare match interrupt (INTTRnCC0) is output.
(Use INTTRnOD (valley interrupt) and INTTRnCD (peak interrupt) as the cycle interrupts.)

(b) Restrictions related to compare match interrupt with TRnCCR1 to TRnCCR3 register

In the high-accuracy T-PWM mode, if set in the range of 0000H ≤ TRnCCRm < TRnDTC0, (TRnCCR0 - TRnDTC1) < TRnCCRm ≤ TRnCCR0, no interrupt occurs upon a match between the compare value and the counter.

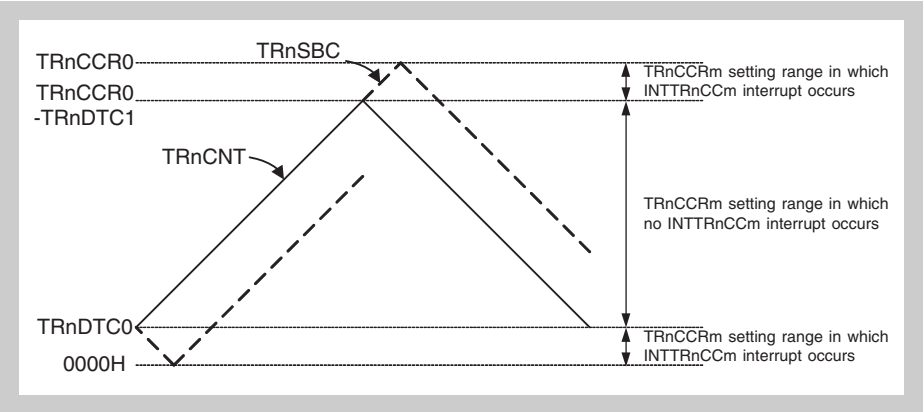


Figure 10-22 Compare match interrupt occurrence related to TRnCCR1 to TRnCCR3 in high-accuracy T-PWM mode

Remark m = 1 to 3

(c) Restrictions related to compare match interrupt with TRnCCR4 and TRnCCR5 registers

In the high-accuracy T-PWM mode, if set in the range of 0000H ≤ TRnCCR4, TRnCCR5 < TRnDTC0, (TRnCCR0 - TRnDTC1) < TRnCCR4, TRnCCR5 ≤ TRnCCR0, no compare match interrupt is output since no match between the compare value and counter occurs.
When TRnCCR4 and TRnCCR5 registers are used as trigger causes for A/D triggers, perform setting in the range of TRnDTC0 ≤ TRnCCR4, TRnCCR5 ≤ (TRnCCR0 - TRnDTC1).

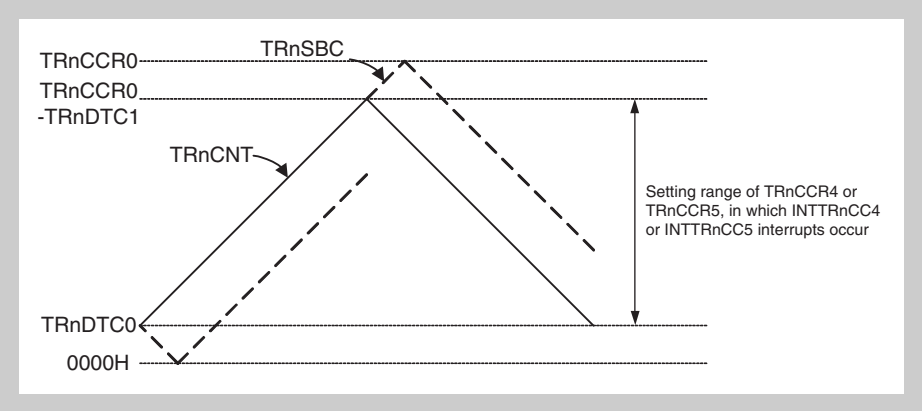


Figure 10-23 Compare match interrupt occurrence related to TRnCC4 and TRnCCR5 in high-accuracy T-PWM mode

(2) Cautions in PWM mode with dead time

Compare match interrupts are output upon a match between the counter and compare registers (TRnCCR0 to TRnCCR5). However, in the high-accuracy T-PWM mode, the compare register can be set exceeding the counter's count operation range. Therefore, under the following conditions, no compare interrupt is output.

Restrictions related to TRnCCRm

In the PWM mode with dead time, if setting is performed in the following range, no match between the compare value and counter occurs, and no compare match interrupt is output:
When TRnCCR0 < TRnCCRm ≤ (TRnCCR0 + TRnDTC0), TRnCCR4, TRnCCR5 registers are used as trigger causes for A/D triggers, perform settings with TRnCCR4, TRnCCR5 ≤ TRnCCR0.

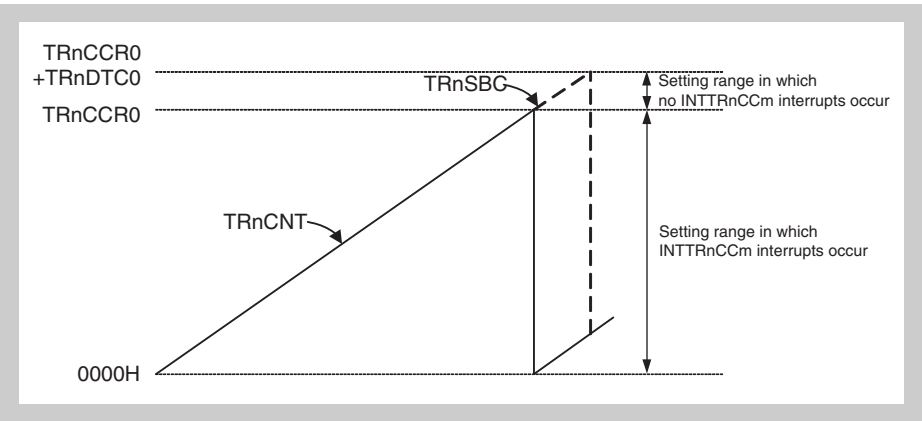


Figure 10-24 Compare match interrupt occurrence related to TRnCCR1 to TRnCCR5 in PWM mode with dead time

Remark m = 1 to 5

10.6 Flags

10.6.1 Up count flags

Timer Rn has two counters, a counter and a sub-counter.

TRnCUF is the counter's up/down status flag (refer to *"TMRn option register 0 (TRnOPT0)" on page 291*). It operates in the triangular wave PWM mode and high-accuracy T-PWM mode, and is fixed to 0 in all other modes.

TRnSUF is the sub-counter's up/down status flag (refer to *"TMRn option register 6 (TRnOPT6)" on page 299*). It operates in the high-accuracy T-PWM mode, and is fixed to 0 in all other modes.

For both TRnCUF and TRnSUF, 0 indicates the up count status, and 1 indicates the down count status.

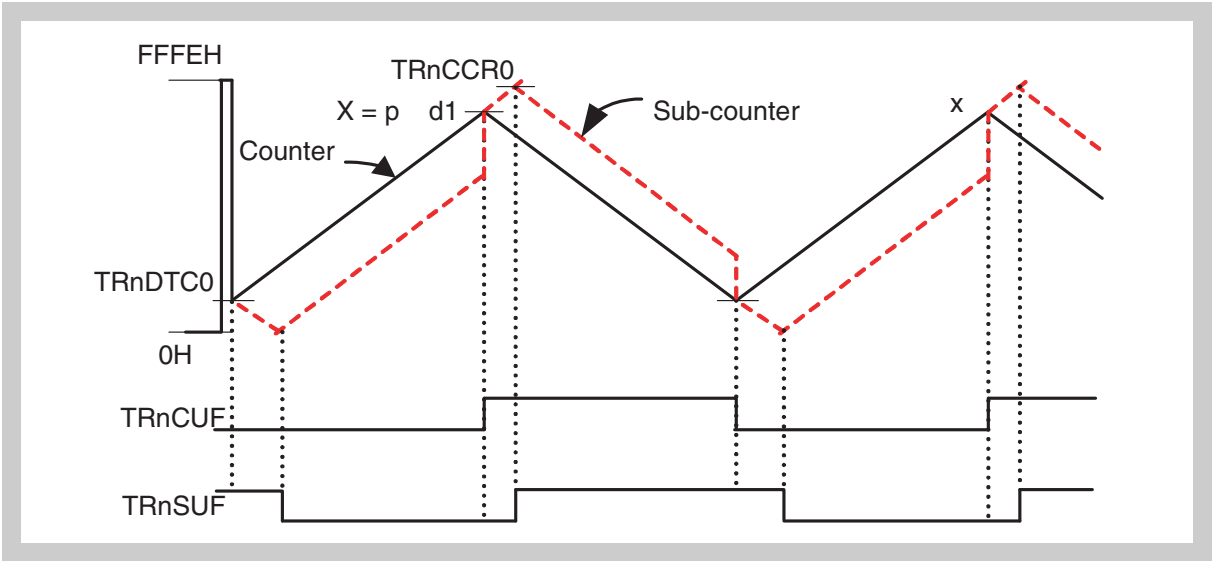


Figure 10-25 Up count flags timings (1/2)

In the triangular wave PWM mode, the values of TRnCUF are as follows.

[TRnCUF]
0 ≤ counter < TRnCCR0+1 … 0 (up count)
TRnCCR0+1 ≥ counter > 0 … 1 (down count)

In the high-accuracy T-PWM mode, the values of TRnCUF/TRnSUF are as follows.

[TRnCUF]
TRnDTC0 ≤ counter < (TRnCCR0 - TRnDTC1) … 0 (up count)
TRnCCR0-TRnDTC1 ≥ counter > TRnDTC0 … 1 (down count)

[TRnSUF]
0 ≤ sub-counter < TRnCCR0 … 0 (up count)
TRnCCR0 ≥ sub-counter > 0 … 1 (down count)

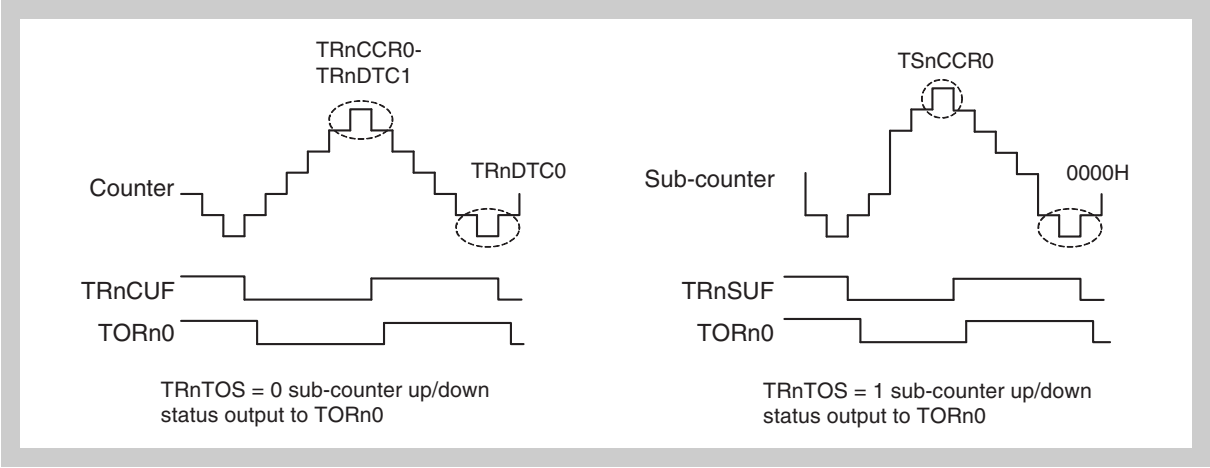


Figure 10-26 Up count flags timings (2/2)

10.6.2 Normal phase/inverted phase simultaneous active detection flag

Timer Rn has a flag (TRnTBF) that detects normal phase/inverted phase simultaneous active states (refer to *"TMRn option register 6 (TRnOPT6)" on page 299*). The TRnTBF flag is valid in the PWM mode, triangular wave PWM mode, high-accuracy T-PWM mode, and PWM mode with dead time.

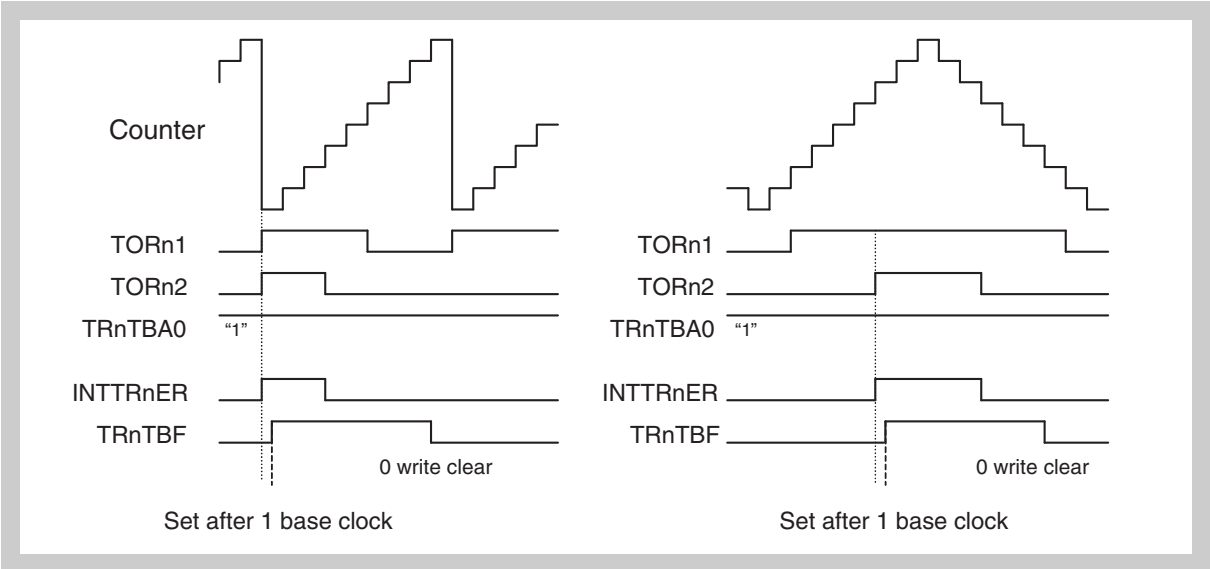


Figure 10-27 Normal phase/inverted phase simultaneous active detection flag timing

10.6.3 Reload hold flag

In the case of timer Rn, the reload hold flag (TRnRSF, refer to *"TMRn option register 6 (TRnOPT6)" on page 299*) is set to "1" upon occurrence of a reload request (when the TRnCCR1 register is written to). When reload occurs and the values are transferred to all the buffer registers, the reload hold flag is cleared to "0". The TRnRSF flag is valid in the following operation modes.

- External trigger pulse output mode
- PWM mode
- Triangular wave PWM mode
- High-accuracy T-PWM mode (TRnCMS = 0)
- PWM mode with dead time

Caution The TRnRSF flag is set to "1" by a delay of 4 base clocks after TRnCCR1 register write completion.

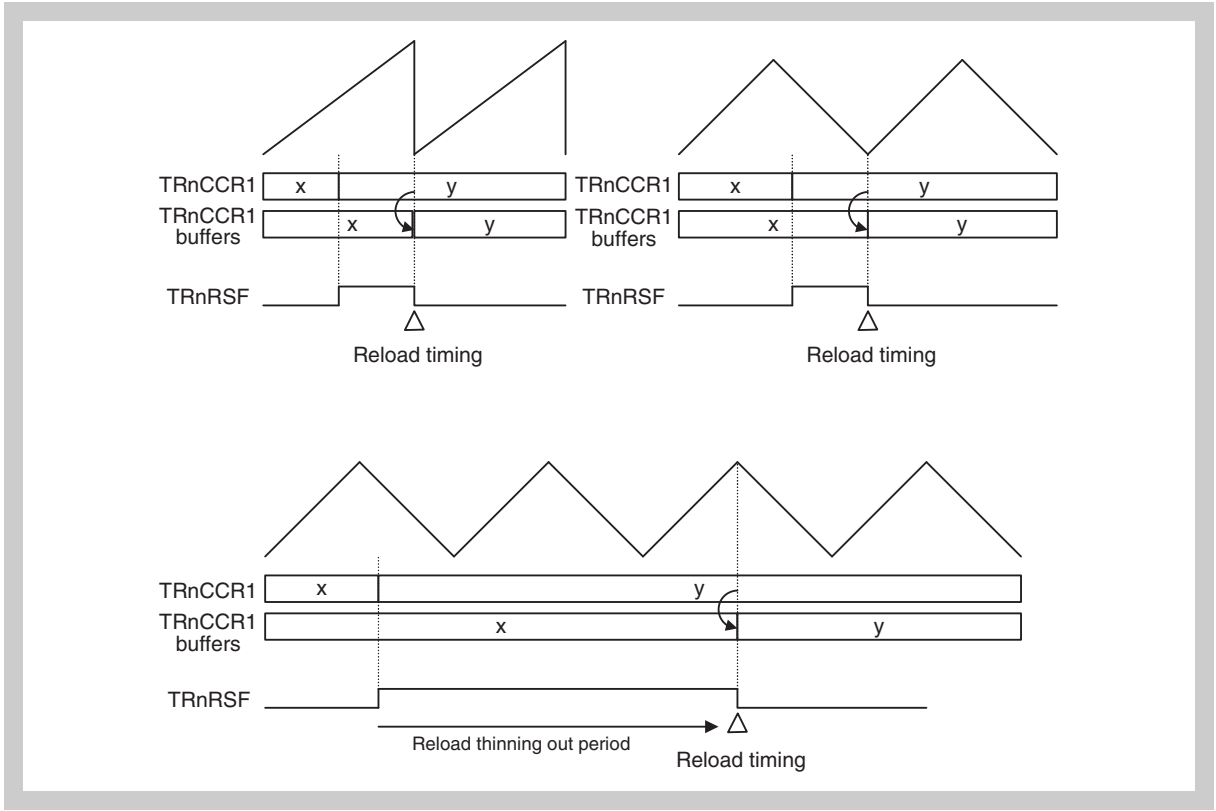


Figure 10-28 Reload hold flag timings

10.7 Interrupt Thinning Out Function

The operations related to the interrupt thinning out function are indicated below.

- The interrupts subject to thinning out are INTTRnCD (peak interrupt) and INTTRnOD (valley interrupt).
- TRnOPT1.TRnICE bit is used to enable INTTRnCD interrupt output and to specify thinning out count targets.
- TRnOPT1.TRnIOE bit is used to enable INTTRnOD interrupt output and to specify thinning out count targets.
- TRnOPT1.TRnRDE bit is used to enable reload thinning out.
- If thinning out is enabled, reload is executed at the same timing as interrupt output following thinning out.
- If thinning out is disabled, reload is executed at the reload timing after write access to the TRnCCR1 register.
- The reload/anytime rewrite method can be specified with TRnOPT0.TRnCMS bit.
- When TRnCMS = 0, the register value is updated in synchronization with reload, but when TRnCMS = 1, the register value is updated immediately after write access.

Caution When write access is performed to the TRnOPT1 register, the internal thinning out counter is cleared when the register value is updated. Therefore, the interrupt interval may temporarily become longer than expected.
To prevent this, it is recommended to set TRnCSM = 0 and TRnRDE = 1, and to change the interrupt thinning out count with the reloaded setting according to interrupt thinning out. Using this method, the interrupt interval is kept the same as the setting value.

10.7.1 Operation of interrupt thinning out function

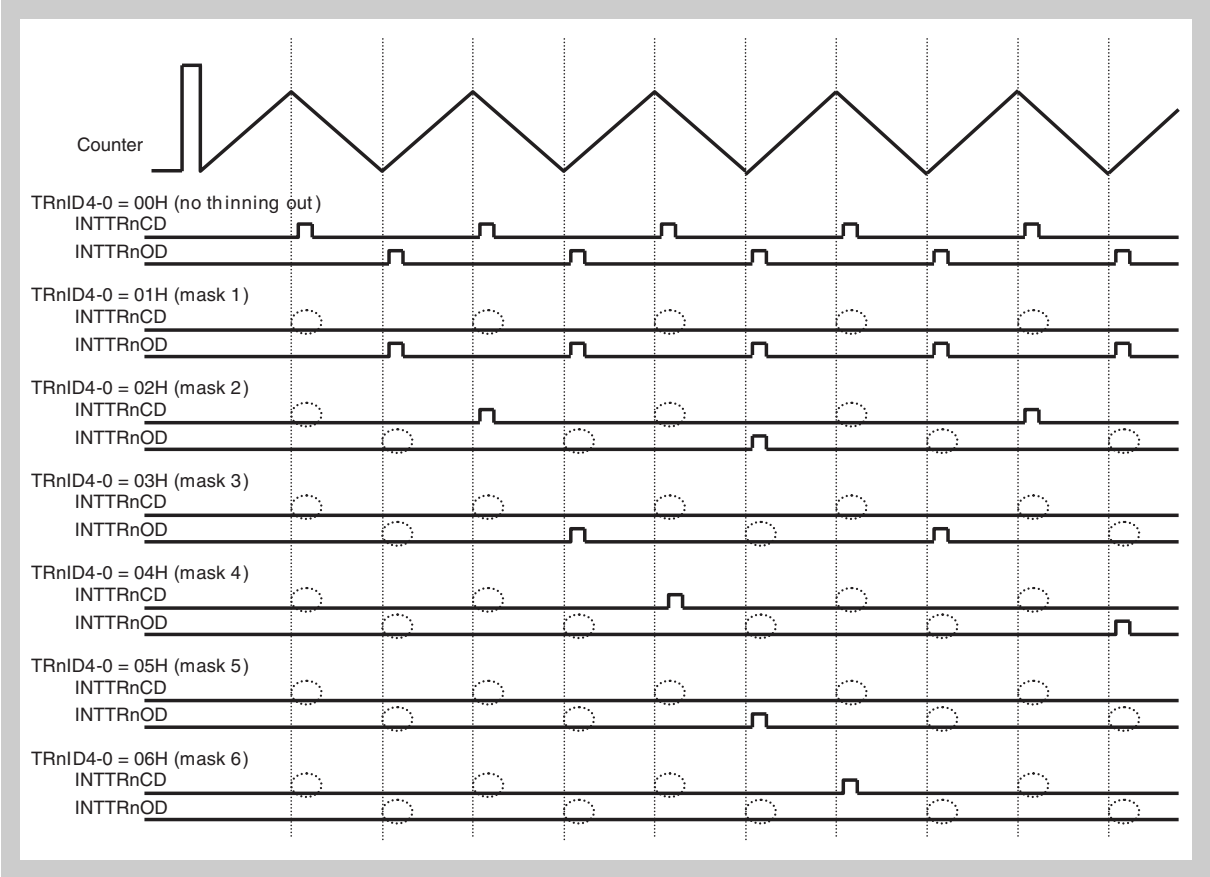


Figure 10-29 Interrupt thinning out operations when TRnICE = 1, TRnIOE = 1 (peak/valley interrupt output)

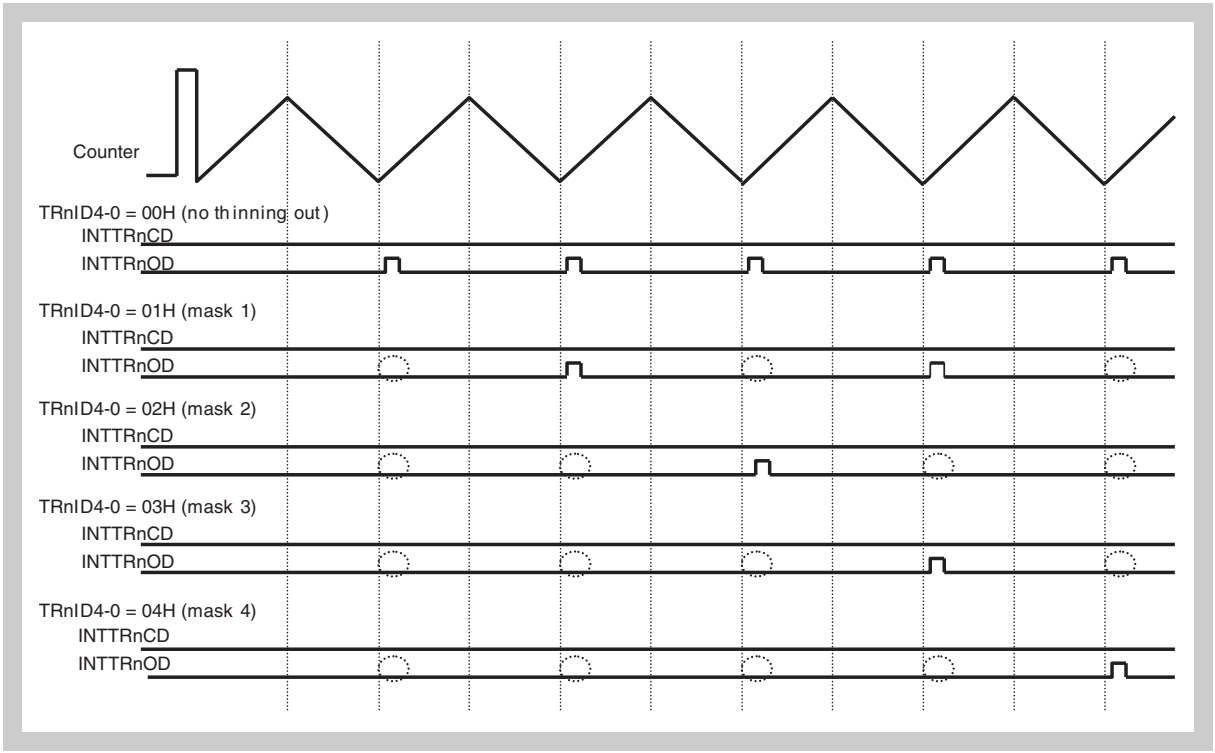


Figure 10-30 Interrupt thinning out operations when TRnICE = 1, TRnIOE = 0 (peak interrupt only output)

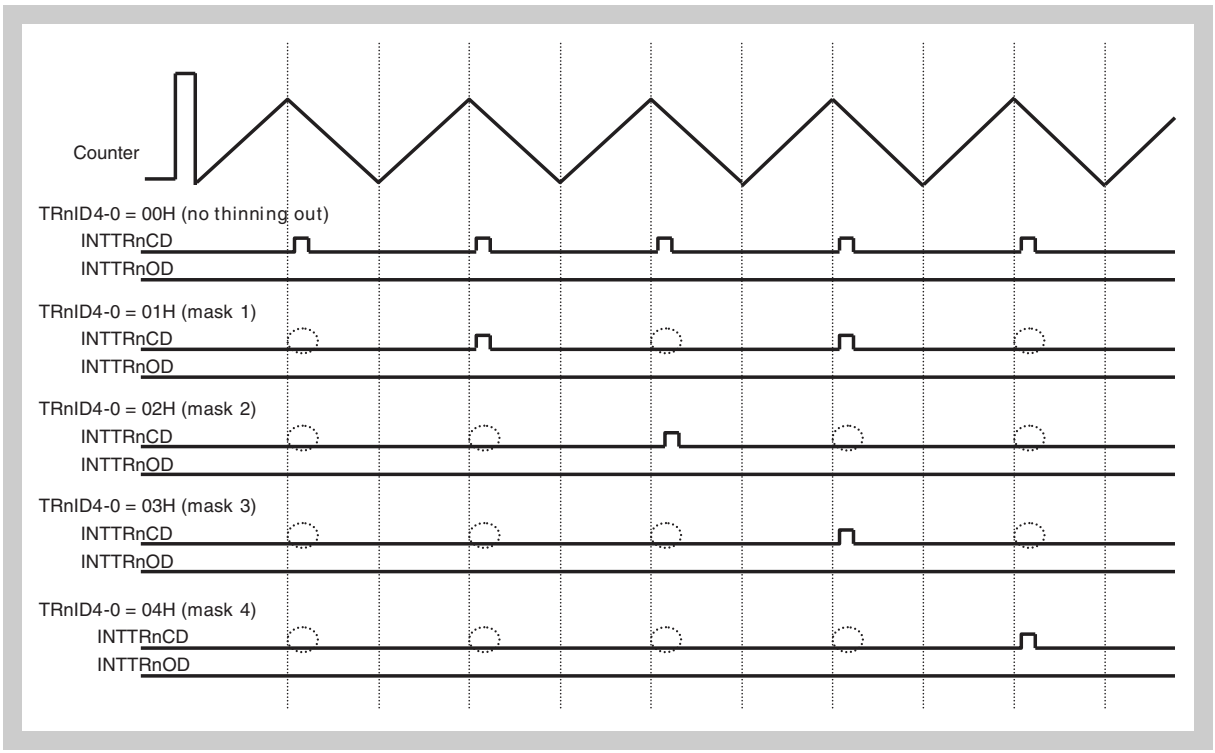


Figure 10-31 Interrupt thinning out operations when TRnICE = 0, TRnIOE = 1 (valley interrupt only output)

10.7.2 Operation examples when peak interrupts and valley interrupts occur alternately

(1) Register settings

Set both TRnOPT1.TRnICE bit and TRnOPT1.TRnIOE bit to 1.

(2) Operation examples

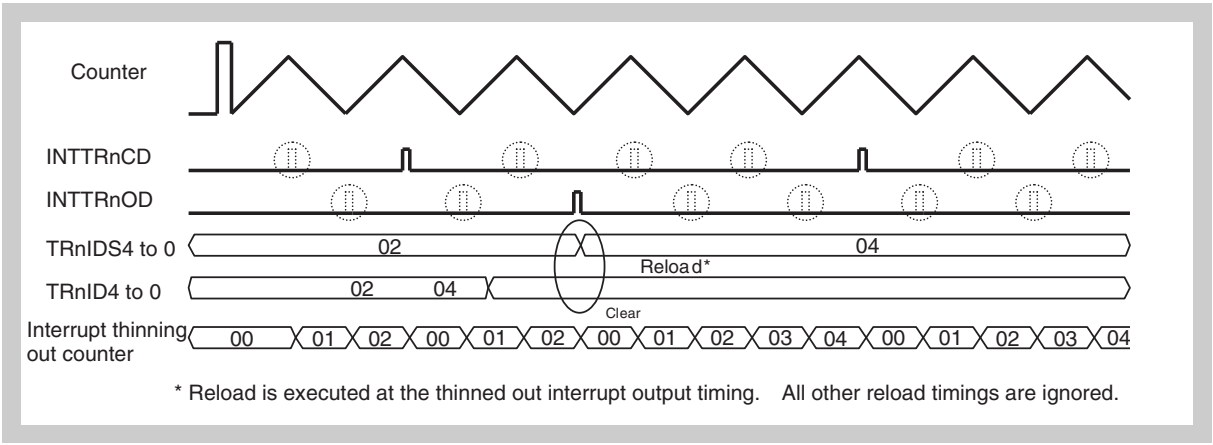


Figure 10-32 Example when peak interrupts and valley interrupts occur alternately when TRnCMS = 0, TRnRDE = 1 (reload thinning out control) (recommended settings)

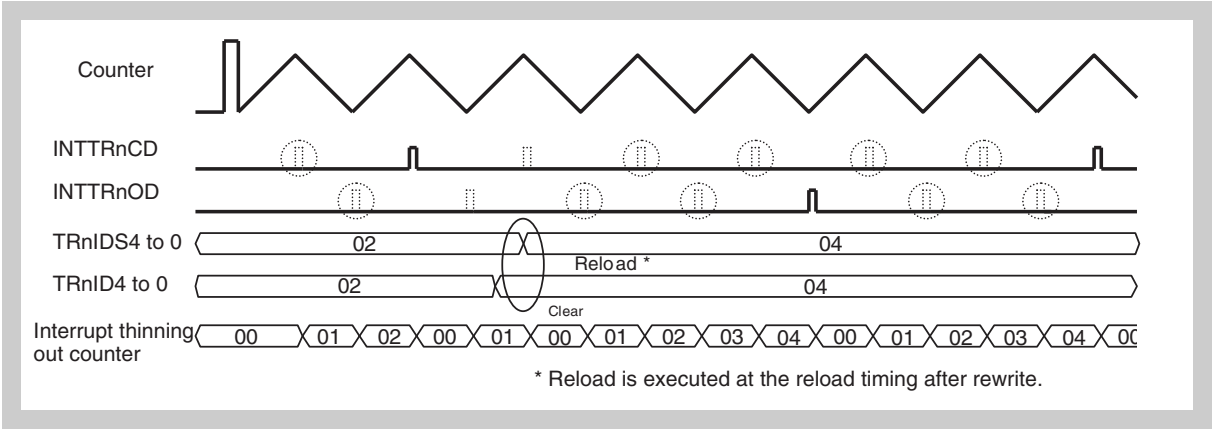


Figure 10-33 Example when peak interrupts and valley interrupts occur alternately when TRnCMS = 0, TRnRDE = 0 (no reload control)

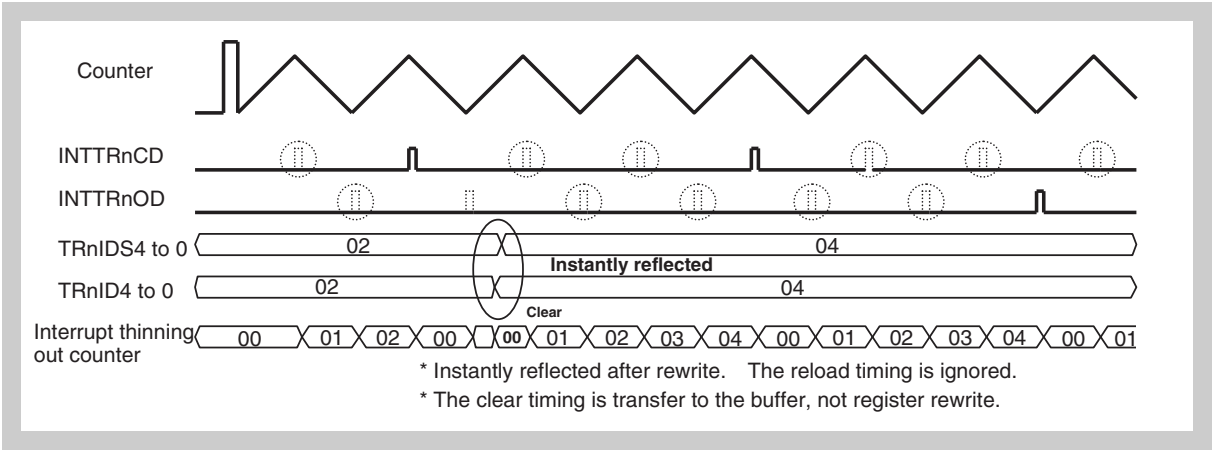


Figure 10-34 Example when peak interrupts and valley interrupts occur alternately when TRnCMS = 1, TRnRDE = x (anytime rewrite)

10.7.3 Interrupt thinning out function during counter saw tooth wave operation

The operations related to the interrupt thinning out function during counter saw tooth wave operation (PWM mode, PWM mode with dead time) are indicated below.

- The interrupt subject to thinning out is INTTRnCD (peak interrupt). The saw tooth wave operation occurs upon a match between the TRnCCR0 register and counter occurs.
- TRnOPT1.TRnICE bit is used to enable INTTRnCD interrupt output and to specify thinning out count targets.
- TRnOPT1.TRnIOE bit setting is invalid. INTTRnOD interrupt output is prohibited.
- TRnOPT1.TRnRDE bit is used to enable reload thinning out.
- If thinning out is enabled, reload is executed at the same timing as interrupt output following thinning out.
- If thinning out is disabled, reload is executed at the reload timing after write access to the TRnCCR1 register.

Caution When write access is performed to the TRnOPT1 register, the internal thinning out counter is cleared when the register value is updated. Therefore, the interrupt interval may temporarily become longer than expected.
To prevent this, it is recommended to set TRnCSM = 0 and TRnRDE = 1, and to change the interrupt thinning out count with the reloaded setting according to interrupt thinning out. Using this method, the interrupt interval is kept the same as the setting value.

10.8 A/D Conversion Trigger Function

This section describes the operation of the A/D conversion triggers output in the PWM mode, triangular wave PWM mode, high-accuracy T-PWM mode, and PWM mode with dead time. In these modes, the TRnCCR4 and TRnCCR5 registers are used as match interrupts and for the A/D conversion trigger function, with no influence on timer outputs in terms of the compare operation. For the A/D conversion triggers that can be output in each mode, refer to *"A/D conversion triggers, peak interrupts, and valley interrupts in each mode" on page 321*.

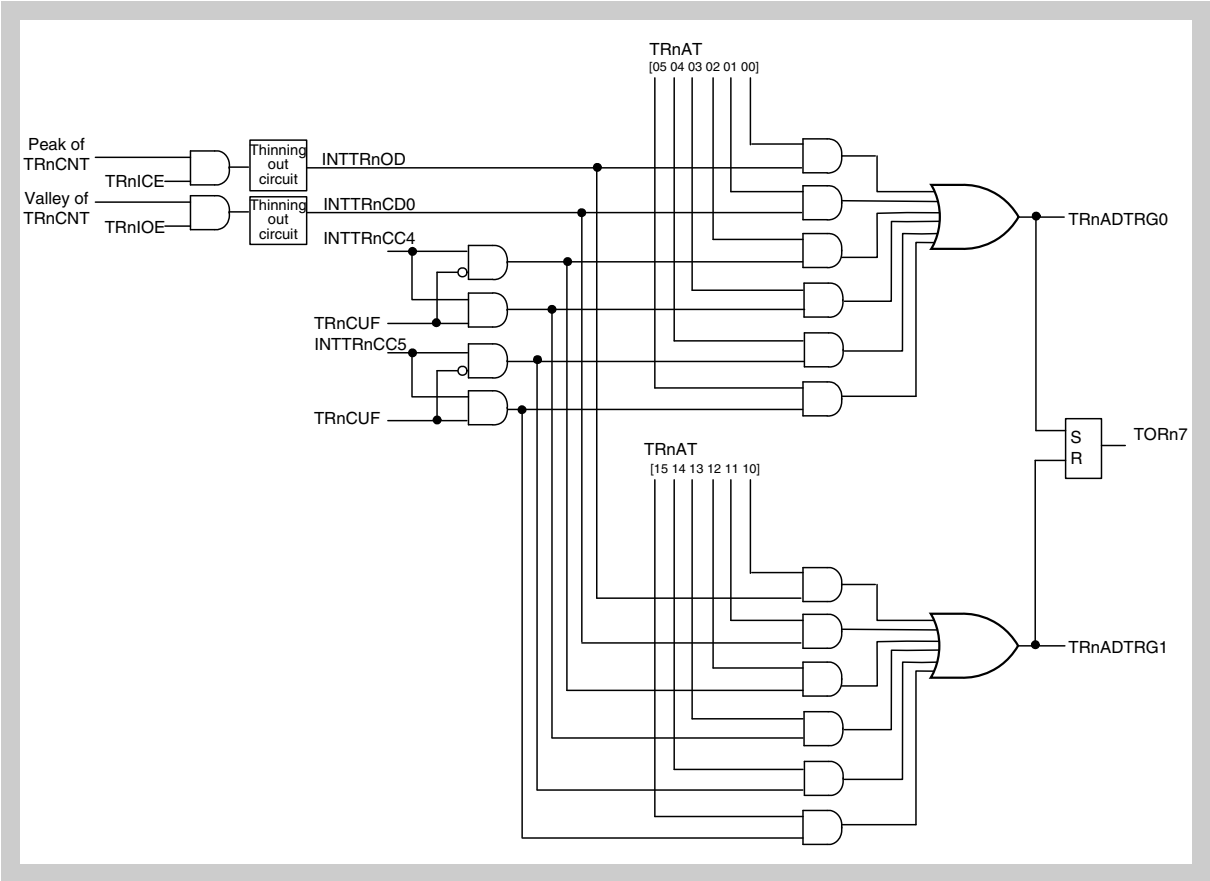


Figure 10-35 A/D conversion trigger output controller

The above figure shows the A/D conversion trigger controller. As shown in this figure, it is possible to select and perform OR output of compare match interrupts (INTTRnCC5, INTTRnCC4) and peak interrupts (INTTRnCD), valley interrupts (INTTRnOD) interrupt signals, sub-counter peak timing, and sub-counter valley timing.

In the case of timer R, there are two identical A/D conversion trigger controllers, and each one can be controlled independently.

10.8.1 A/D conversion trigger operation

Timer R has a function for generating A/D conversion start triggers (TRnADTRG0, TRnADTRG1 signals), freely selecting 4 trigger sources. The following 4 triggers sources are provided, which can be specified with TRnOPT2.TRnAT0[5:0] bits and TRnOPT3.TRnAT1[5:0] bits. Here, control of the TRnADTRG0 using TRnOPT2.TRnAT0[5:0] bits is described. The same type of control can be achieved for the TRnADTRG1 signal with control bits TRnOPT3.TRnAT1[5:0]

(1) TRnADTRG0 signal output control

- TRnOPT2.TRnAT00 = 1: Output of A/D conversion trigger upon valley interrupt (INTTRnOD) output
- TRnOPT2.TRnAT01 = 1: Output of A/D conversion trigger upon peak interrupt (INTTRnCD) output
- TRnOPT2.TRnAT02 = 1: A/D conversion trigger outputtable upon compare match interrupt (INTTRnCC4) during counter up count
- TRnOPT2.TRnAT03 = 1: A/D conversion trigger outputtable upon compare match interrupt (INTTRnCC4) during counter down count
- TRnOPT2 .TRnAT04 = 1: A/D conversion trigger outputtable upon compare match interrupt (INTTRnCC5) during counter up count
- TRnOPT2.TRnAT05 = 1: A/D conversion trigger outputtable upon compare match interrupt (INTTRnCC5) during counter down count

The A/D conversion start trigger signals selected with bits TRnOPT2.TRnAT0[5:0] are all “ORed” and output to the TRnADTRG0 pin.

The peak and valley interrupts (INTTRnOD, INTTRnCD) selected with bits TRnOPT2.TRnAT00 and TRnOPT2.TRnAT01 are the signals after interrupt thinning out. Therefore, when interrupt output is enabled (TRnOPT1.TRnICE and TRnOPT1.TRnIOE) they are output at the timing when interrupt thinning out control is received.

Moreover, TRnOPT2.TRnAT0[5:0] can be rewritten during operation.

When the A/D conversion start trigger setting bit is rewritten during operation, this is instantly reflected to the output status of the A/D conversion start trigger.

These control bits do not have a reload function and are write accessed only in anytime write mode.

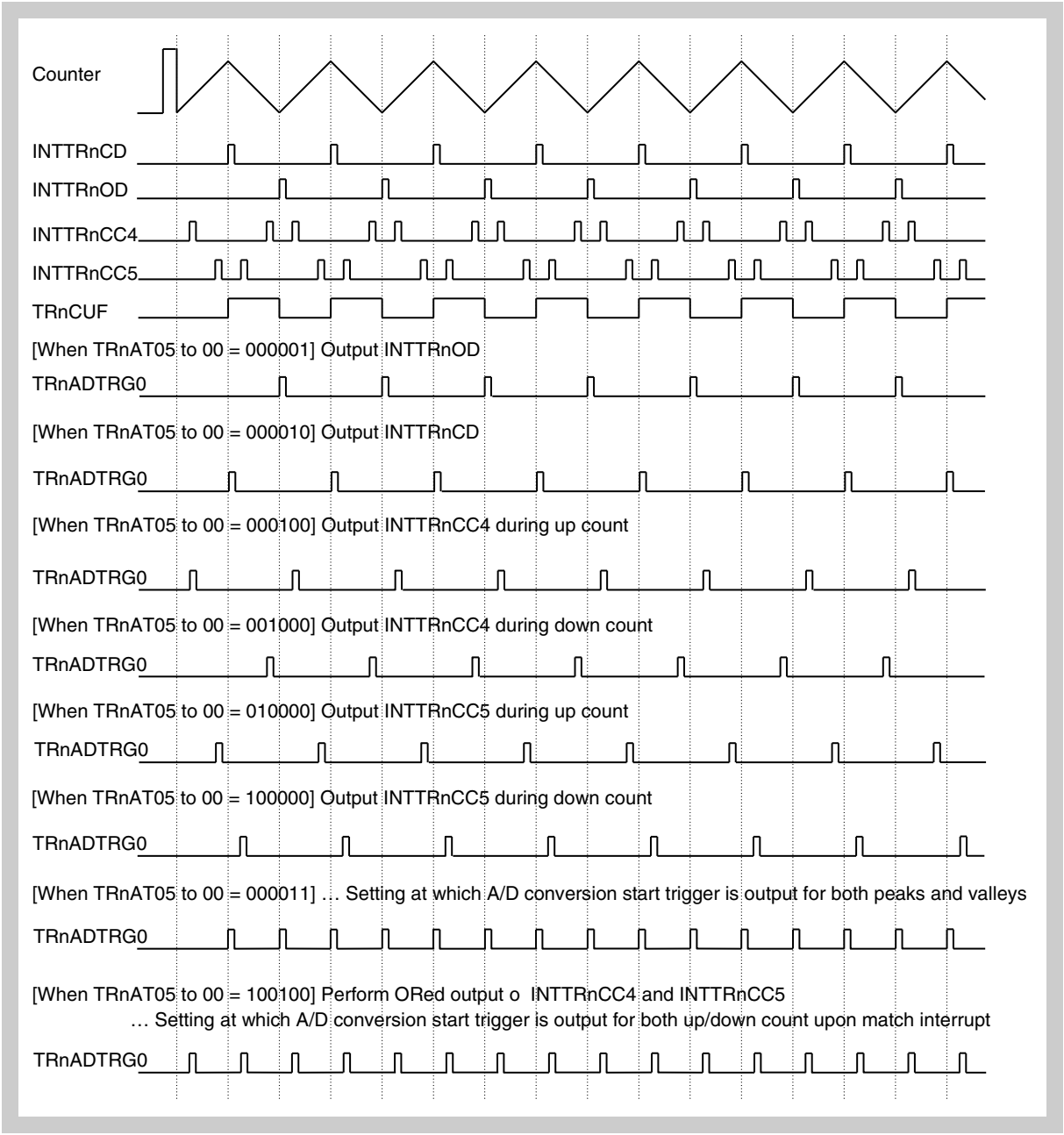


Figure 10-36 A/D conversion trigger timings when INTTRnCD and INTTRnOD are selected and not thinned out (TRnICE = 1, TRnIOE = 1, TRnID[4:0] = 00_H)

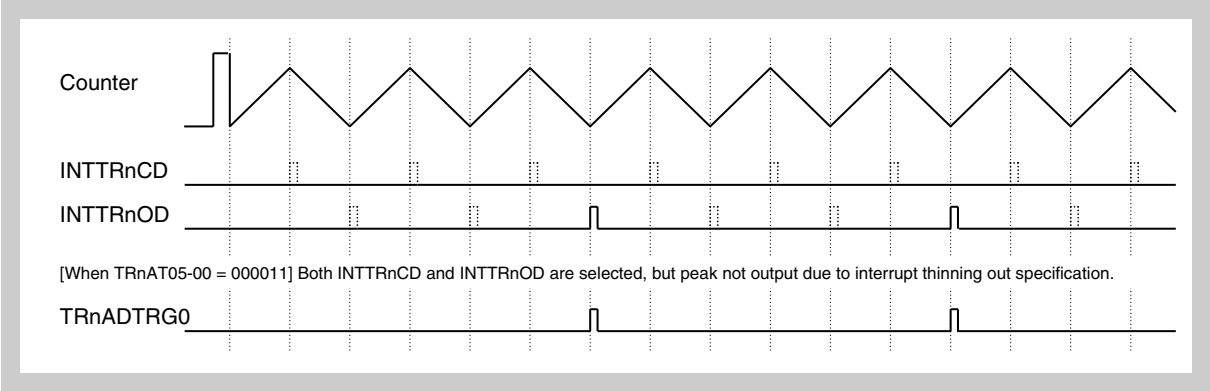


Figure 10-37 A/D conversion trigger timings when INTCTRnOD is selected and thinned out (TRnICE = 0, TRnIOE = 1, TRnID[4:0] = 02_H)

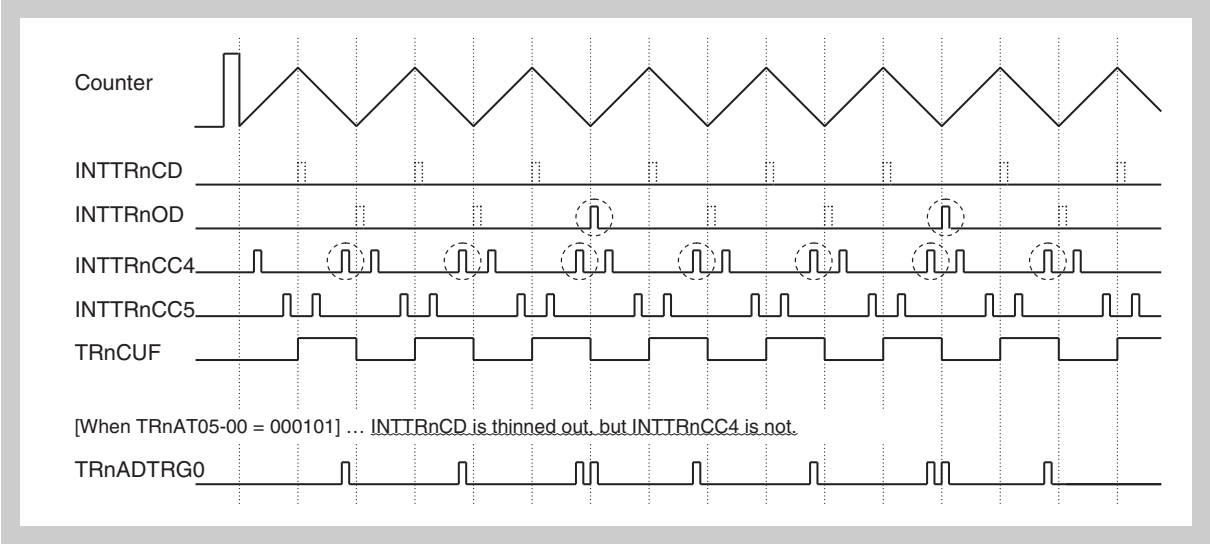


Figure 10-38 A/D conversion trigger timings when INTTRnCC4 is selected and INTTRnOD is selected and thinned out (TRnICE = 0, TRnIOE = 1, TRnID[4:0] = 02_H)

(2) Cautions related to A/D conversion triggers

In PWM mode and PWM mode with dead time, no valley interrupt (INTTRnOD) is output. Only peak interrupts (INTTRnCD) are valid.

10.9 Error Interrupts

10.9.1 Error interrupt and error signal output functions

Timer R has an error interrupt (INTTRnER) and an error signal output (TRnER).

As the errors detected with timer R, normal phase/inverted phase simultaneous active (fault of dead time controller) are detected as errors in the high-accuracy T-PWM mode, PWM mode with dead time, and PWM mode. Regarding normal phase/inverted phase simultaneous active errors, following error occurrence, the error occurrence can be confirmed by reading bit TRnOPT6 .TRnTBF.

Moreover, detection ON/OFF switching control in each phase (TORn1/TORn2, TORn3/TORn4, TORn5/TORn6) is possible using bits TRnIOC4.TRnTBA[2:0].

The possibility of normal phase/inverted phase simultaneous active error detection in each mode is indicated in the table below.

Table 10-28 Error detection in each mode

Mode	Normal Phase/Inverted Phase Simultaneous Active Detection
Interval mode	×
External event count mode	×
External trigger pulse output mode	×
One-shot pulse mode	×
PWM mode	√
Free-running mode	×
Pulse width measurement mode	×
Triangular wave PWM mode	√
High-accuracy T-PWM mode	√
PWM mode with dead time	√

Remark √:Error detection possible
×:Error detection not possible

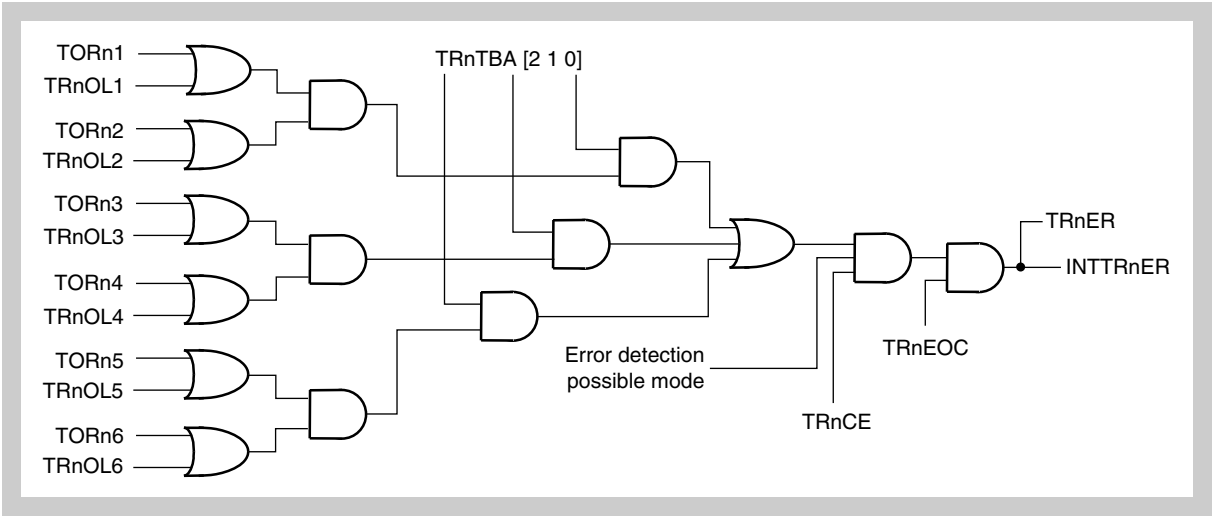


Figure 10-39 Error interrupt (INTTRnER) and error signal (TRnER) output controller

Output of the error signal (TRnER) due to normal phase/inverted phase simultaneous active error is active level during detection of normal phase/inverted phase simultaneous active.

(1) In PWM mode

The case of normal phase/inverted phase simultaneous active in the PWM mode is described below.

As shown in the figure below, an error interrupt (INTTRnER) is output when the TRnCCR1 and TRnCCR2 registers are set so that pins TORn1 and TORn2 simultaneously output “H”. Similarly, an error interrupt (INTTRnER) is output when the TRnCCR3 and TRnCCR4 registers are set so that pins TORn3 and TORn4 simultaneously output “H”.

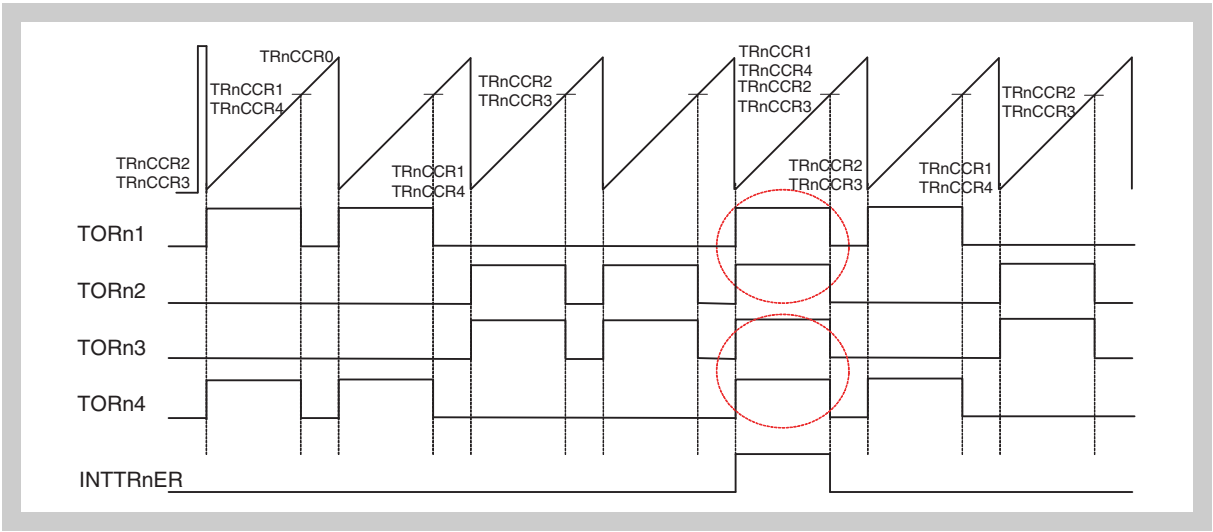


Figure 10-40 Error interrupt timing in PWM mode

If the output active level is switched by manipulating the bits TRnIOC0.TRnOL1 and TRnIOC0.TRnOL2, the following results.

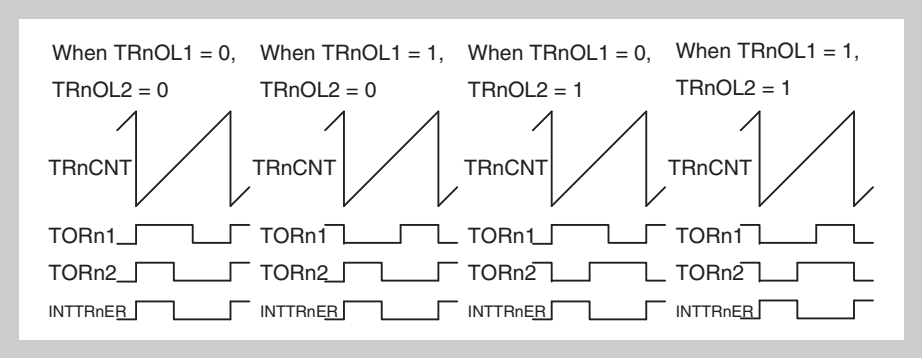


Figure 10-41 Dependency of error interrupt on output active level setting

(2) In triangular wave PWM mode

The case of normal phase/inverted phase simultaneous active in the triangular wave PWM mode is described below.

As shown in the figure below, an error output (INTTRnER) is output when the TRnCCR0 and TRnCCR1 registers are set so that pins TORn1 and TORn2 simultaneously output “H”. Similarly, an error interrupt (INTTRnER) is output when the TRnCCR3 and TRnCCR4 registers are set so that pins TORn3 and TORn4 simultaneously output “H”.

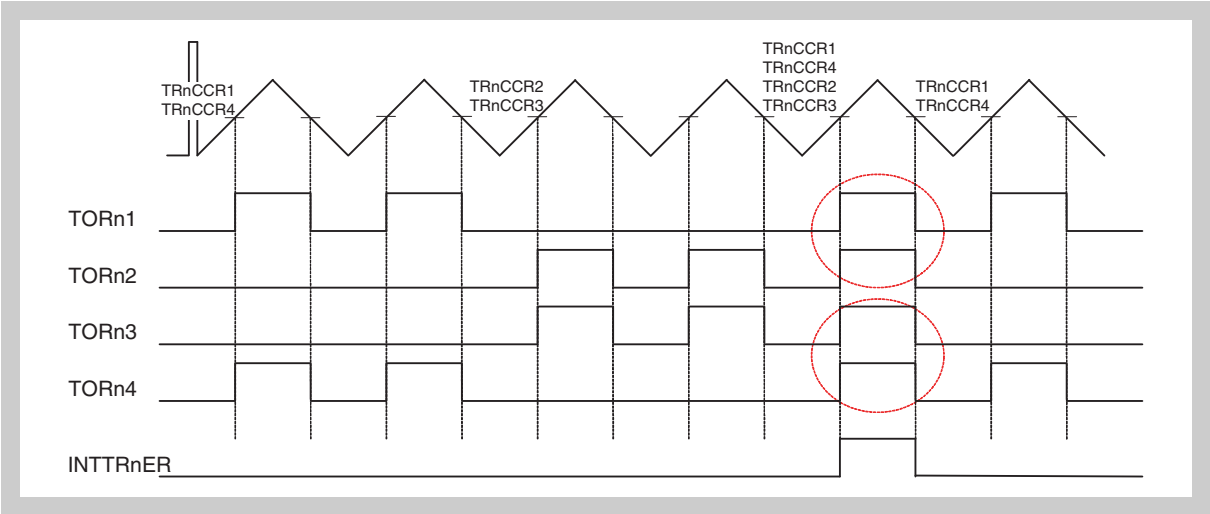


Figure 10-42 Error interrupt timing in triangular wave PWM mode

(3) In high-accuracy T-PWM mode/PWM mode with dead time

In the high-accuracy T-PWM mode and PWM mode with dead time, no error occurs except when the dead time setting is “0”. If an error occurs, this is likely due to an internal circuit fault.

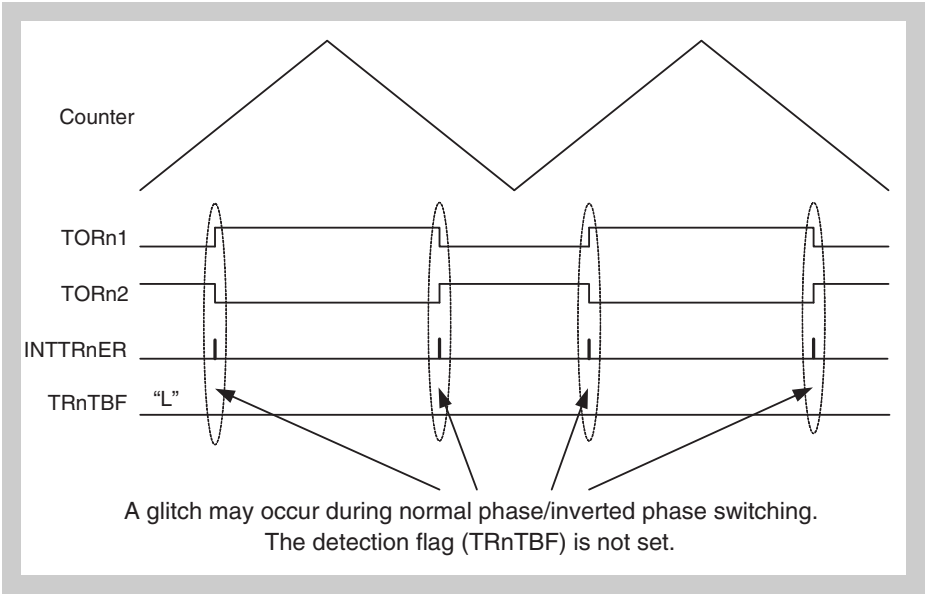


Figure 10-43 Error interrupt timing in high-accuracy T-PWM mode

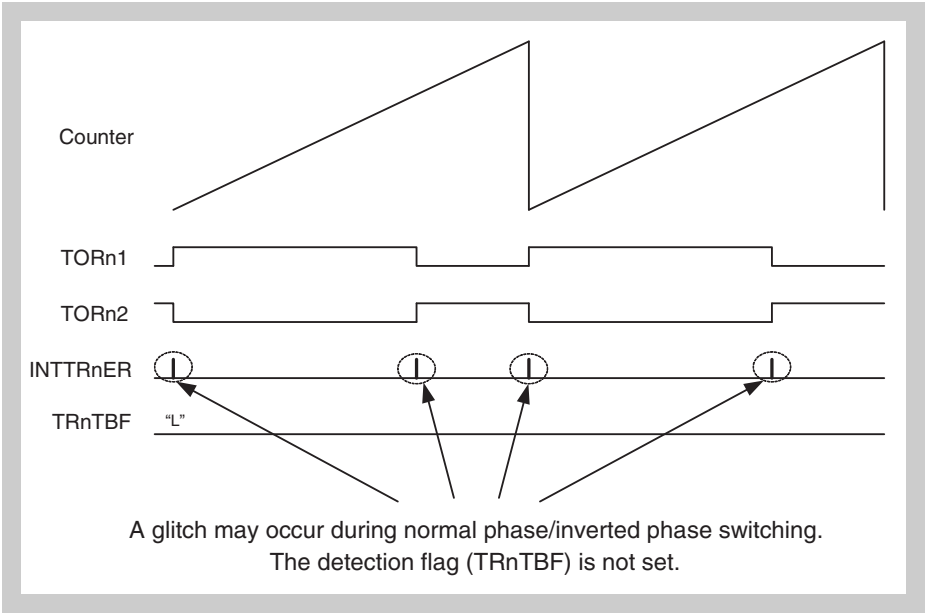


Figure 10-44 Error interrupt timing in PWM Mode with dead time

10.10 Operation in Each Mode

10.10.1 Interval timer mode

(1) Outline of interval timer mode

In the interval timer mode, a compare match interrupt (INTTRnCC0) occurs and the counter is cleared upon a match between the setting value of the TRnCCR0 register and the counter value. The occurrence interval for this counter and TRnCCR0 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 TRnCCR0 register. Counter clearing using the TRnCCR1 to TRnCCR5 registers is not performed.

However, the setting values of the TRnCCR1 to TRnCCR5 registers are compared to the counter values transferred to the TRnCCR1 to TRnCCR5 buffer registers and compare match interrupts (INTTRnCC1 to INTTRnCCR5) are output.

The TRnCCR0 to TRnCCR5 registers can be rewritten using the anytime write method, regardless of the value of bit TRnCE.

Pins TORn0 to TORn5 are toggle output controlled when bits TRnOE0 to TRnOE5 are set to 1.

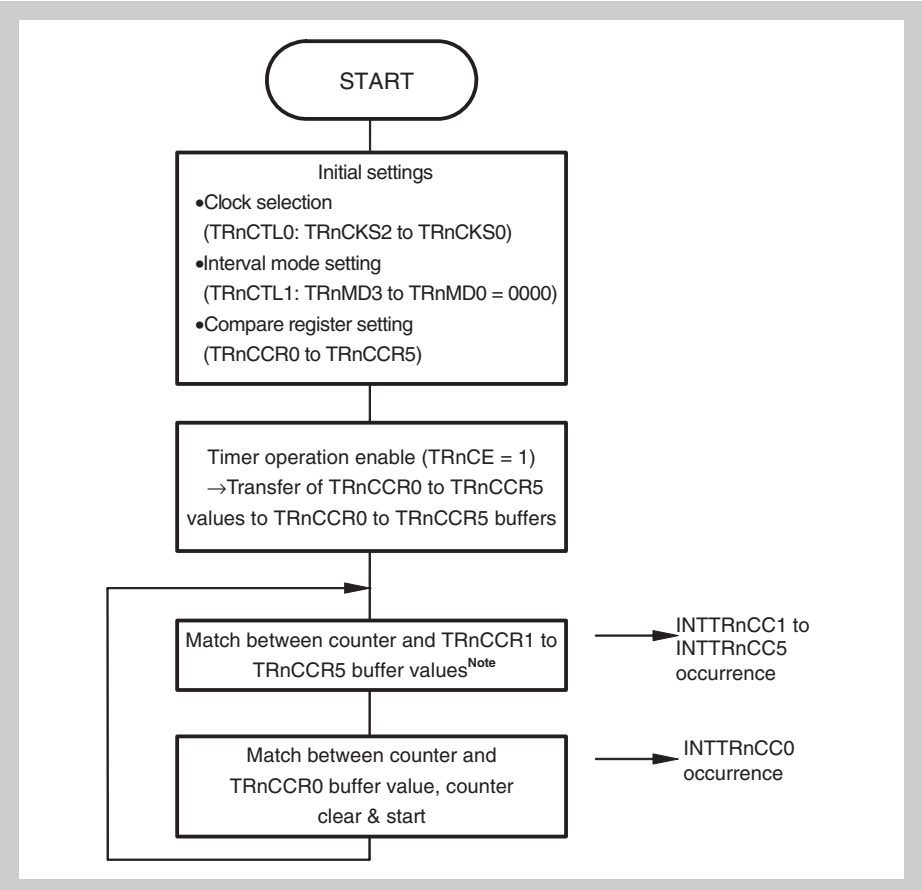


Figure 10-45 Basic operation flow in interval timer mode

Note In the case of a match between the counter and TRnCCR1 to TRnCCR5 registers, the counter is not cleared.

(2) Interval timer mode operation list

(a) Compare registers

Register	Rewriting method	Rewriting during operation	Function
TRnCCR0	Anytime rewrite	Possible	Compare and clear value (interval time)
TRnCCR1 to TRnCCR3	Anytime rewrite	Possible	Compare value
TRnCCR4, TRnCCR5	Anytime rewrite	Possible	Compare value

(b) Input pins

Pin	Function
TIR10 to TIR13	–
TTRGR1	–
TEVTR1	–

(c) Output pins

Pin	Function
TORn0 to TORn5	Toggle output upon TRnCCRm register compare match (m = 0 to 5)
TORn6, TORn7	–

(d) Interrupts

Interrupt	Function
INTTRnCCm	TRnCCRm register compare match (m = 0 to 5)
INTTRnOV	–
INTTRnER	–

(3) Operation timing in interval timer mode

The timing example below assumes that D1 > D2 > D3, only the value of TRnCCR0 register is rewritten, TORn0 and TORn1 are not output (TRnOE0, 1 = 0, TRnOL0 = 0, TRnOL1 = 1).

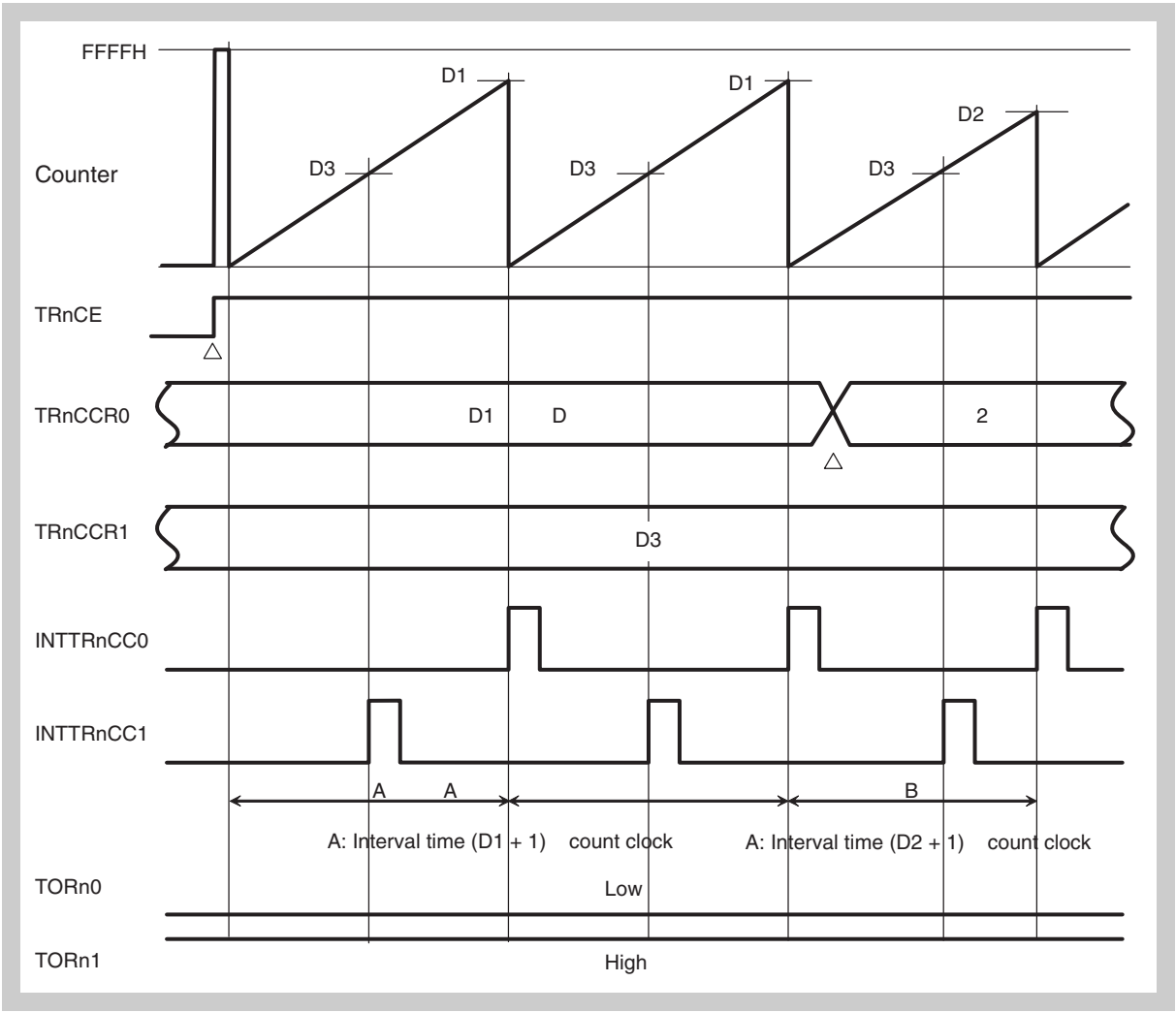


Figure 10-46 Basic operation timing in interval timer mode (1/2)

Remarks

1. D1, D2: Setting values of TRnCCR0 register (0000_H to $FFFF_H$)
   D3: Setting values of TRnCCR1 register (0000_H to $FFFF_H$)
2. Interval time = (Dm + 1) × (count clock cycle)
3. m = 1 to 3

The timing example below assumes that D1 = D2, the values of TRnCCR0 and TRnCCR1 registers not rewritten, TORn0 and TORn1 are output (TRnOE0 = TRnOE1 = 1, TRnOL0 = 0, TRnOL1 = 1)

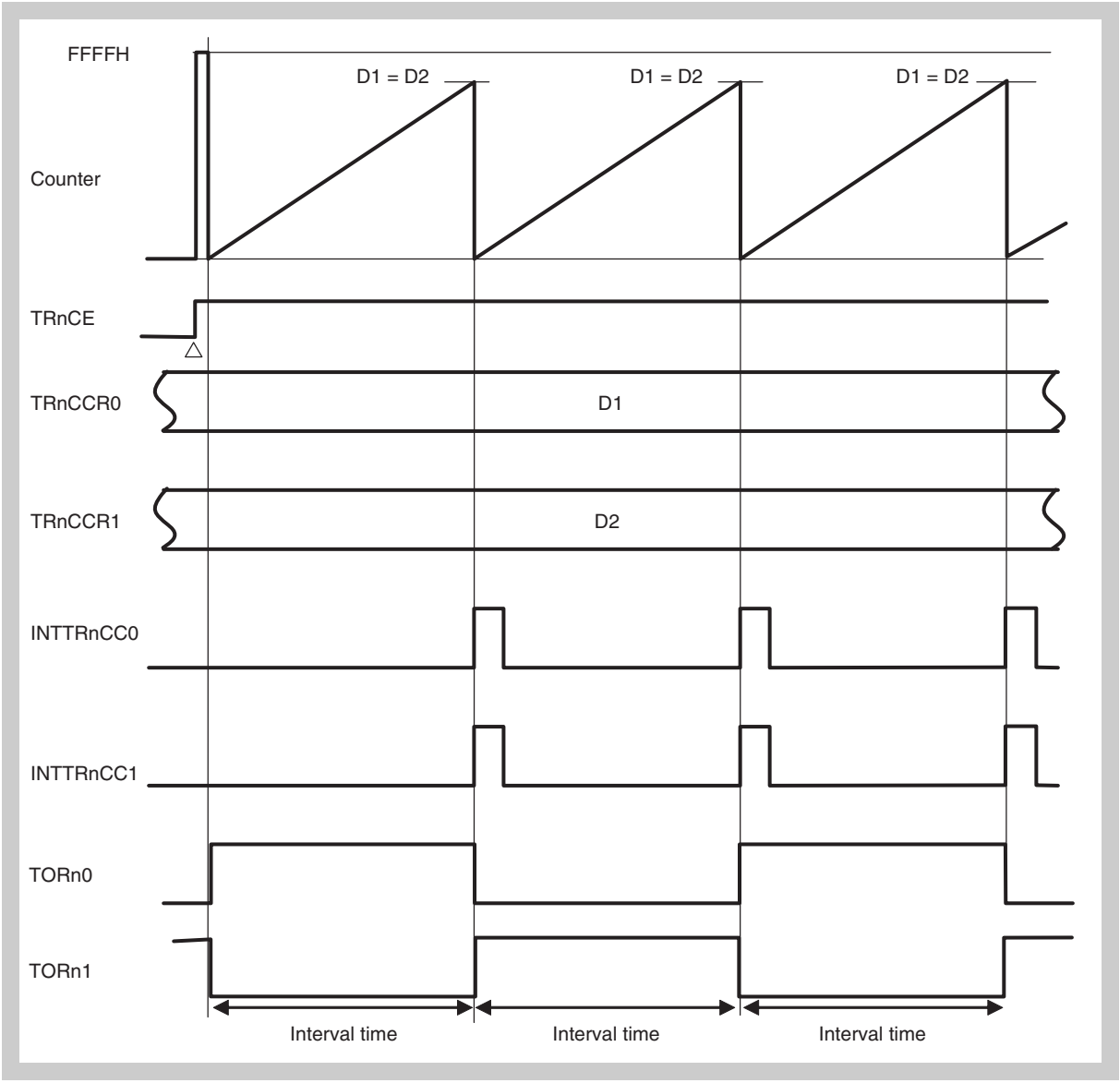


Figure 10-47 Basic operation timing in interval timer mode (2/2)

Remarks

1. D1: Setting value of TRnCCR0 register (0000_H to $FFFF_H$)
   D2: Setting value of TRnCCR1 register (0000_H to $FFFF_H$)
2. Interval time = (Dm + 1) × (count clock cycle)
3. TORn0, TORn1 toggle time = (Dm + 1) × (count clock cycle)
4. m = 1, 2

10.10.2 External event count mode (TMR1 only)

(1) Outline of external event count mode

In the external event count mode, count up starts upon external event input (TEVTR1 pin). The external event input (TEVTR1) is used as the count clock, regardless of TRnCTL1.TR1EEE bit.

In the external event count mode, the counter is cleared only upon a match between the counter and the value of the TR1CCR0 register. Counter clearing using the TR1CCR1 to TR1CCR5 registers is not performed.

However, the values of the TR1CCR1 to TR1CCR5 registers are transferred to the TR1CCR1 to TR1CCR5 buffer registers, compared to the counter value, and compare match interrupts (INTTR1CC1to INTR1CCR5) are output.

The TR1CCR0 to TR1CCR5 registers can be rewritten with the anytime write method, regardless of the value of bit TR1CE.

Pins TOR10 to TOR15 are toggle output controlled when bits TR1OE0 to TR1OE5 are set to 1.

When a compare register TR1CCR0 to TR1CCR5 is not used, it is recommended to set it contents to $FFFF_H$.

[External event count operation flow]

<1> TR1CTL1.TR1MD[3:0] = 0001_B (mode setting)
Edge detection set with TRnIOC2.TRnEES[1:0] other than 00_B
<2> TR1CTL0.TR1CE = 1 (count enable)
<3> TEVTR1n pin input edge detection (count-up start)

Caution

1. In case of the external event count mode, when the content of the TR1CCR0 register is set to m, the number of TEVTR1 pin input edge detection times is m+1.
2. Do not set the value of the TR1CCR0 register to 0000_H in external event count mode.
3. When a TR1CCR1 to TR1CCR5 register value is set to 0000_H in external event count mode the corresponding interrupt (INTTR1CC1 to INTTR1CC5) does not occur immediately after start, but after the first overflow of the timer ($FFFF_H$ to 0000_H).

(2) External event count mode operation list

(a) Compare registers

Register	Rewriting method	Rewriting during operation	Function
TR1CCR0	Anytime rewrite	Possible	Compare & clear value (Interval time)
TR1CCR1 to TR1CCR3	Anytime rewrite	Possible	Compare value
TR1CCR4, TR1CCR5	Anytime rewrite	Possible	Compare value

(b) Input pins

Pin	Function
TIR10 to TIR13	–
TTRGR1	–
TEVTR1	External event count input pin

(c) Output pins

Pin	Function
TOR10 to TOR15	Toggle output upon TRnCCRm register compare match (m = 0 to 5)
TOR16, TOR17	–

(d) Interrupts

Interrupt	Function
INTTR1CCm	TRnCCRm register compare match (m = 0 to 5)
INTTR1OV	–
INTTR1ER	–

(3) Operation timing in external event count mode

The timing example below assumes that D1 > D2 > D3, only value of TRnCCR0 register is rewritten, TOR10 and TOR11 are not output (TR1OE0 = TR1OE1 = 0, TR1OL0 = 0, TR1OL1 = 1).
The signal input from TEVTRn is internally synchronized and counted as the count clock.

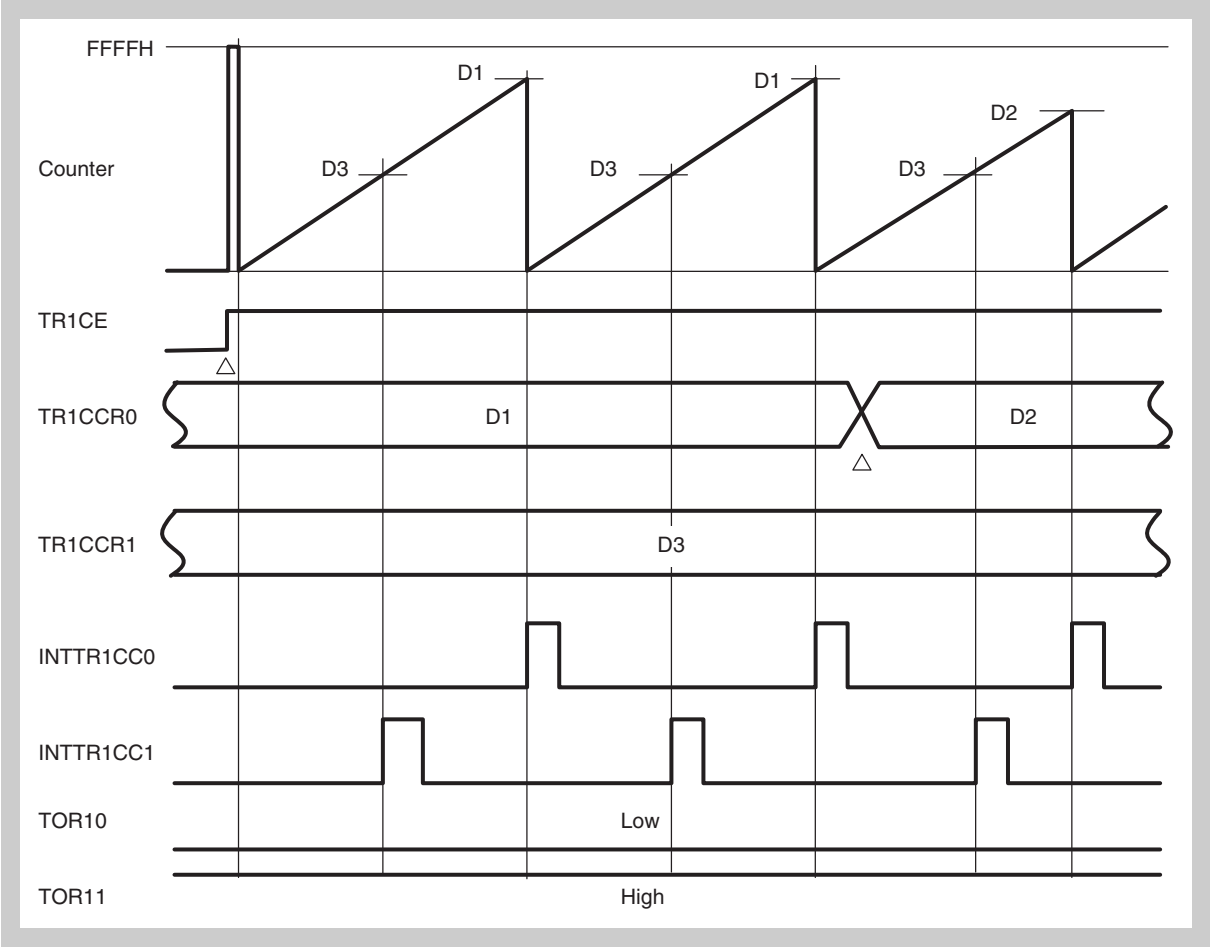


Figure 10-48 Basic operation timing in external event count mode (1/3)

Remarks
1. D1, D2: Setting values of TR1CCR0 register (0001_H to $FFFF_H$)
   D3: Setting value of TR1CCR1 register (0000_H to $FFFF_H$)
2. Number of event counts = (Dm + 1) (m = 1, 2)

The timing example below assumes that D1 = D2, TR1CCR0 and TR1CCR1 register values are not rewritten, TOR10 and TOR11 are output (TR1OE0 = TR1OE1 = 1, TR1OL0 = 0, TR1OL1 = 1).

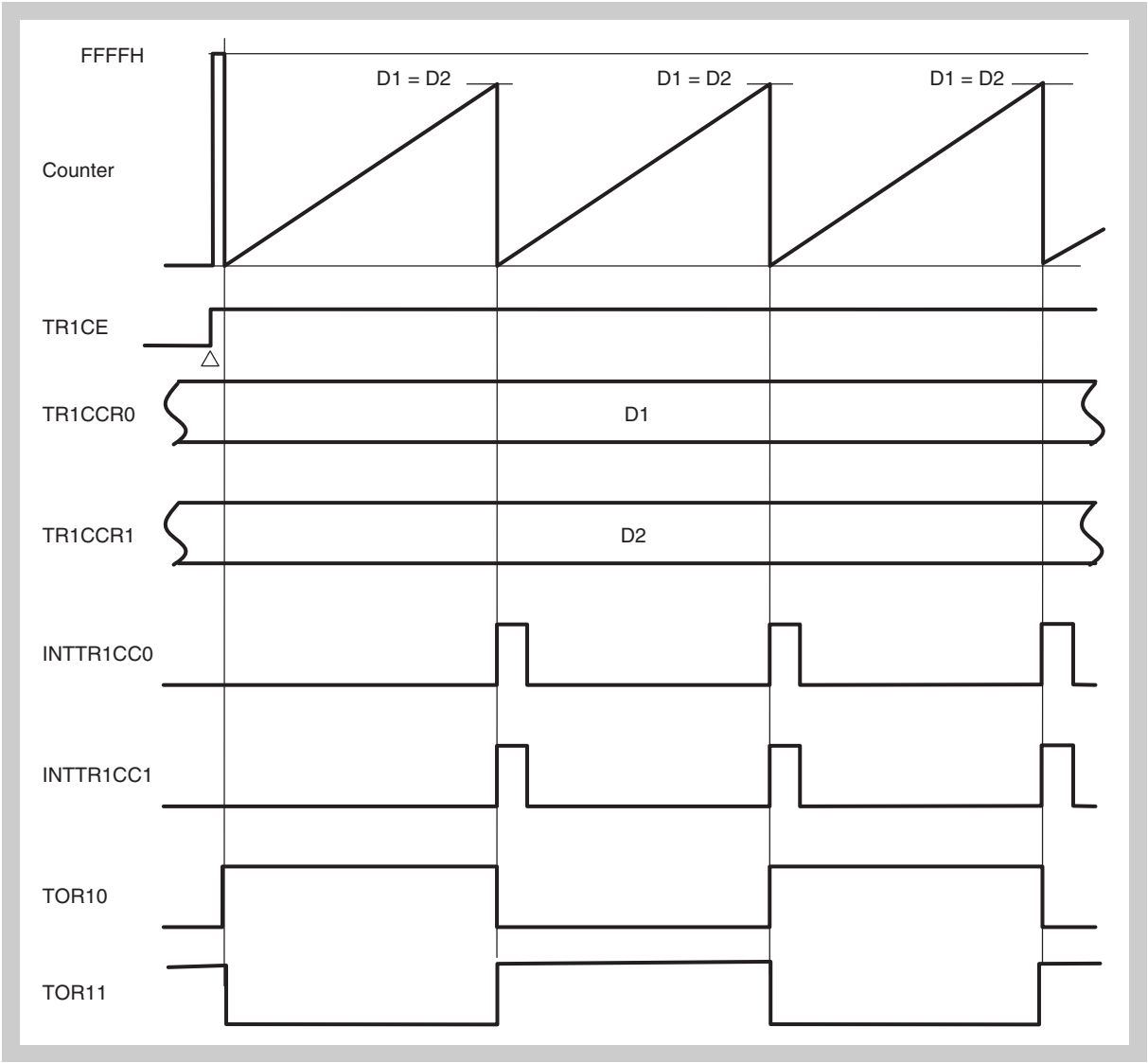


Figure 10-49 Basic operation timing in external event count mode (2/3)

Remarks

1. D1: Setting value of TR1CCR0 register (0001_H to $FFFF_H$)
   D2: Setting value of TR1CCR1 register (0000_H to $FFFF_H$)
2. Number of event counts = (Dm + 1) (m = 1, 2)

The timing example below assumes that D1 = D2, TR1CCR0 and TR1CCR1 register values are not rewritten, TOR10 and TOR11 are output (TR1OE0 = TR1OE1 = 1, TR1OL0 = 0, TR1OL1 = 1).

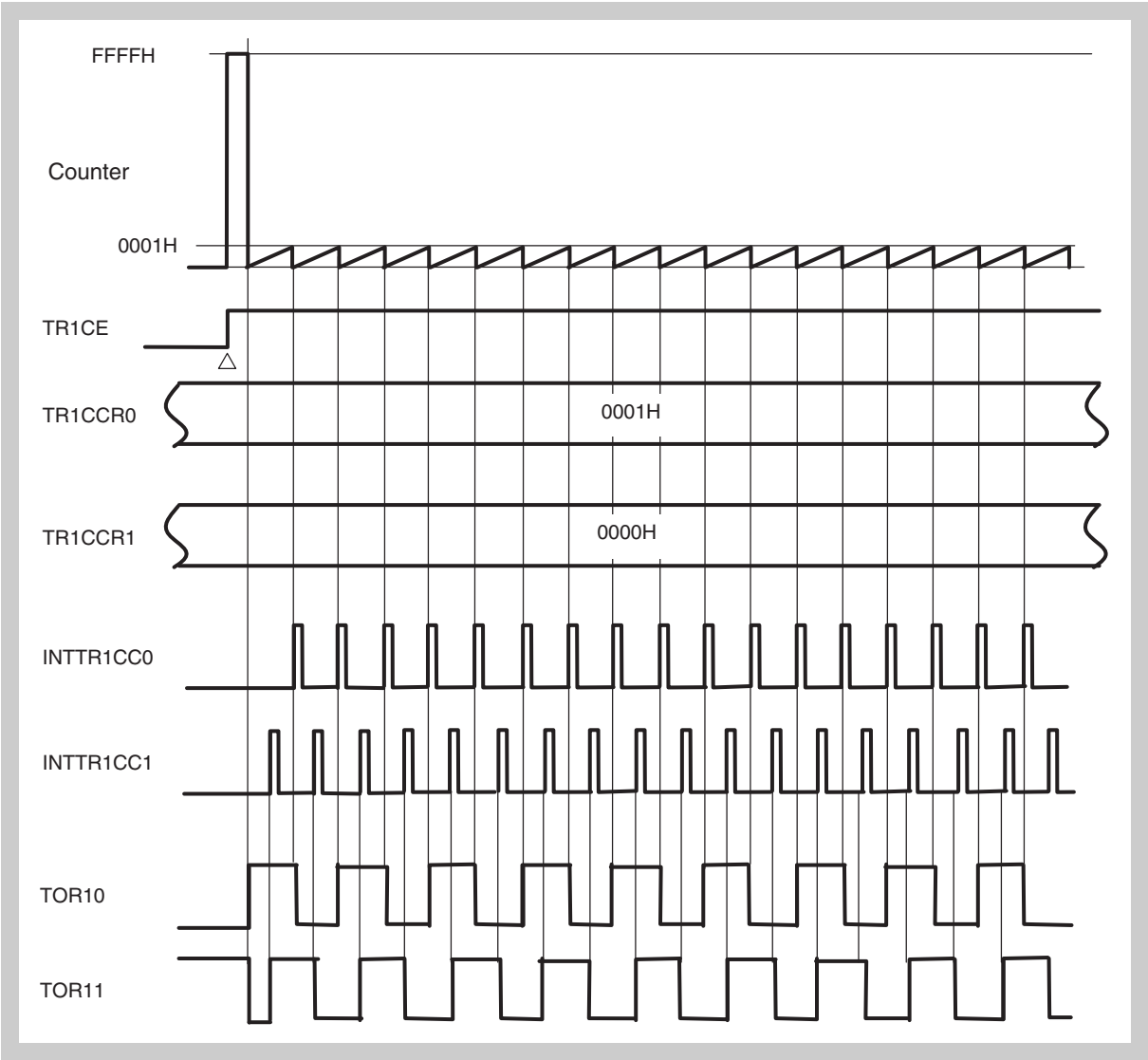


Figure 10-50 Basic operation timing in external event count mode (3/3)

The example above assumes that D1 = D2, TR1CCR0, TR1CCR1 register values are not rewritten, TOR10 and TOR11 are output (TR1OE0 = TR1OE1 = 1, TR1OL0 = 0, TR1OL1 = 1)

Remarks

1. D1: Setting value of TRnCCR0 register (0001_H)
   D2: Setting value of TRnCCR1 register (0000_H)
2. Number of event counts = (Dm + 1) (m = 1, 2)

10.10.3 External trigger pulse output mode (TMR1 only)

(1) Outline of external trigger pulse output mode

When, in external trigger pulse mode, the duty is set to the TR1CCR1 to TR1CCR5 registers, the cycle is set to the TR1CCR0 register, and bit TR1CTL0.TR1CE = 1 is set, external trigger input (TTRGR1 pin) wait results, with the counter remaining stopped at $FFFF_H$. Upon detection of the valid edge of external trigger input (TTRGR1 pin), or when the TR1CTL1.TR1EST bit is set, count up starts. An external trigger pulse is output from pins TOR11 to TOR15, and toggle output is performed from pin TOR10 upon a match with the TR1CCR0 register. Moreover, during the count operation, upon a match between the counter and the TR1CCR0 register, a compare match interrupt (INTTR1CC0) is output, and upon a match between the counter and TR1CCR1 to TR1CCR5 registers, compare match interrupts (INTTR1CC1 to INTTR1CC5) are output.

The TR1CCR0 to TR1CCR5 registers can be rewritten during count operation. Compare register reload is performed at the timing when the counter value and the TR1CCR0 register match. However, when write access to the TR1CCR1 register is performed, the next reload timing becomes valid, so that even if wishing to rewrite only the value of the TR1CCR0 register, write the same value to the TR1CCR1 register. In this case, reload is not performed even if only the TR1CCR0 register is rewritten.

If, during operation in the external trigger pulse output mode, the external trigger (TTRGR1 pin) edge is detected several times, or if the TR1CTL1.TR1EST bit is set (1), the counter is cleared and count up is resumed. Moreover, if at this time, the TOR11 to TOR15 pins are in the low level status, the TOR11 to TOR15 pin outputs become high level when an external trigger is input. If the TOR11 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 TR1CCR0 to TR1CCR3 registers have their function fixed as compare registers, so the capture function cannot be used.

Caution In the external trigger pulse mode, the external event clock input (TEVTR1) is prohibited (TR1CTL1.TR1EEE = 0).

(2) External trigger pulse output mode operation list

(a) Compare registers

Register	Rewriting method	Rewriting during operation	Function
TR1CCR0	Anytime rewrite	Possible	Compare & clear value (Cycle)
TR1CCR1 to TR1CCR3	Anytime rewrite	Possible	Compare value (Duty)
TR1CCR4, TR1CCR5	Anytime rewrite	Possible	Compare value (Duty)

(b) Input pins

Pin	Function
TIR10 to TIR13	–
TTRGR1	Counter clear & start through external trigger input
TEVTR1	–

(c) Output pins

Pin	Function
TOR10	Toggle output upon TR1CCR0 register compare match or external trigger input
TOR11 to TOR15	External trigger pulse waveform output
TOR16, TOR17	–

(d) Interrupts

Interrupt	Function
INTTR1CCm	TR1CCRm register compare match (m = 0 to 5)
INTTR1OV	–
INTTR1ER	–

(3) Operation flow in external trigger pulse output

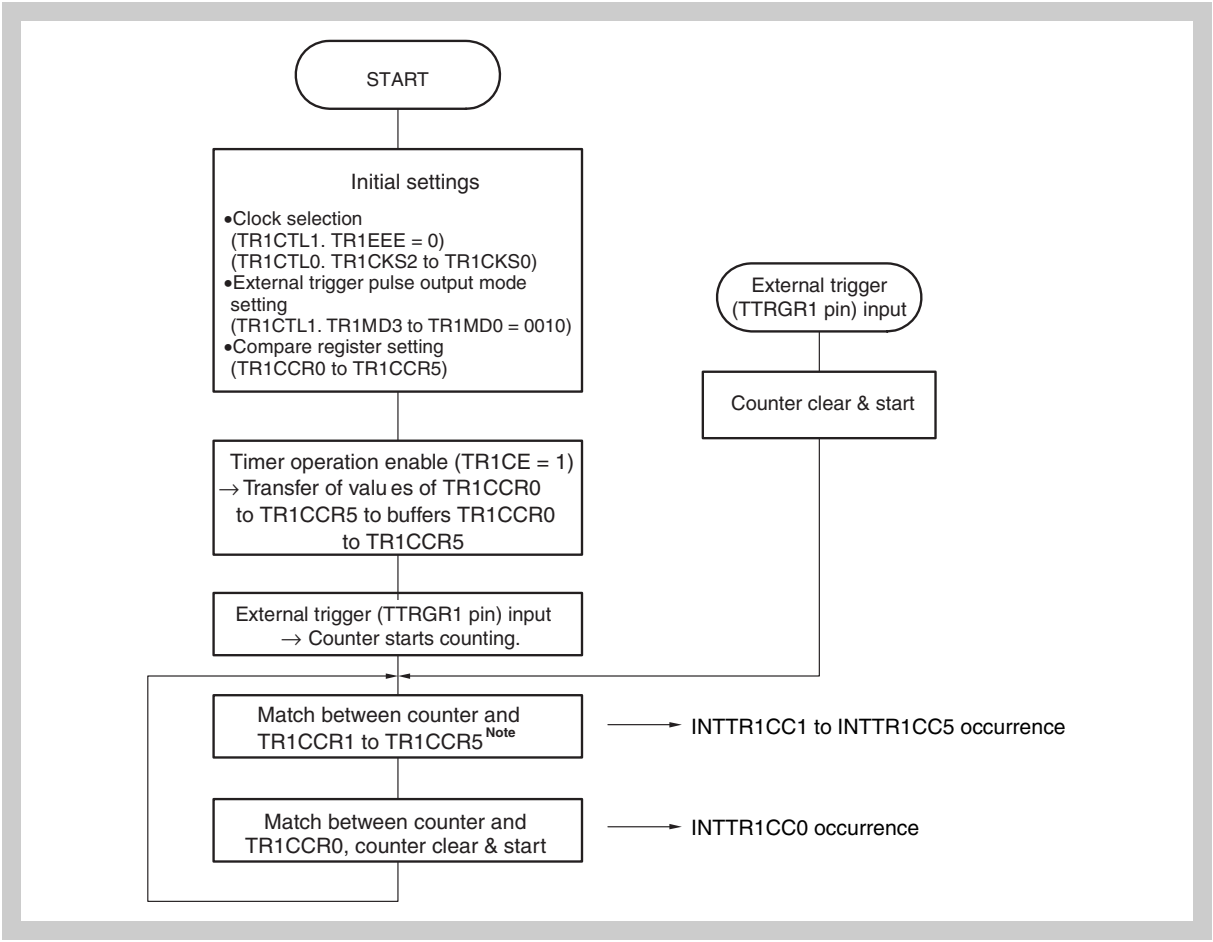


Figure 10-51 Basic operation flow in external trigger pulse output mode

Note The counter is not cleared upon a match between the counter and the TR1CCR1 to TR1CCR5 buffer register.

(4) Operation timing in external trigger pulse output

In the timing example below the values of TR1CCR0 and TR1CCR1 registers are rewritten, TOR10 and TOR11 are set to output with active H-level (TRnOE0 = TRnOE1 = 1, TRnOL0 = TRnOL1 = 0)

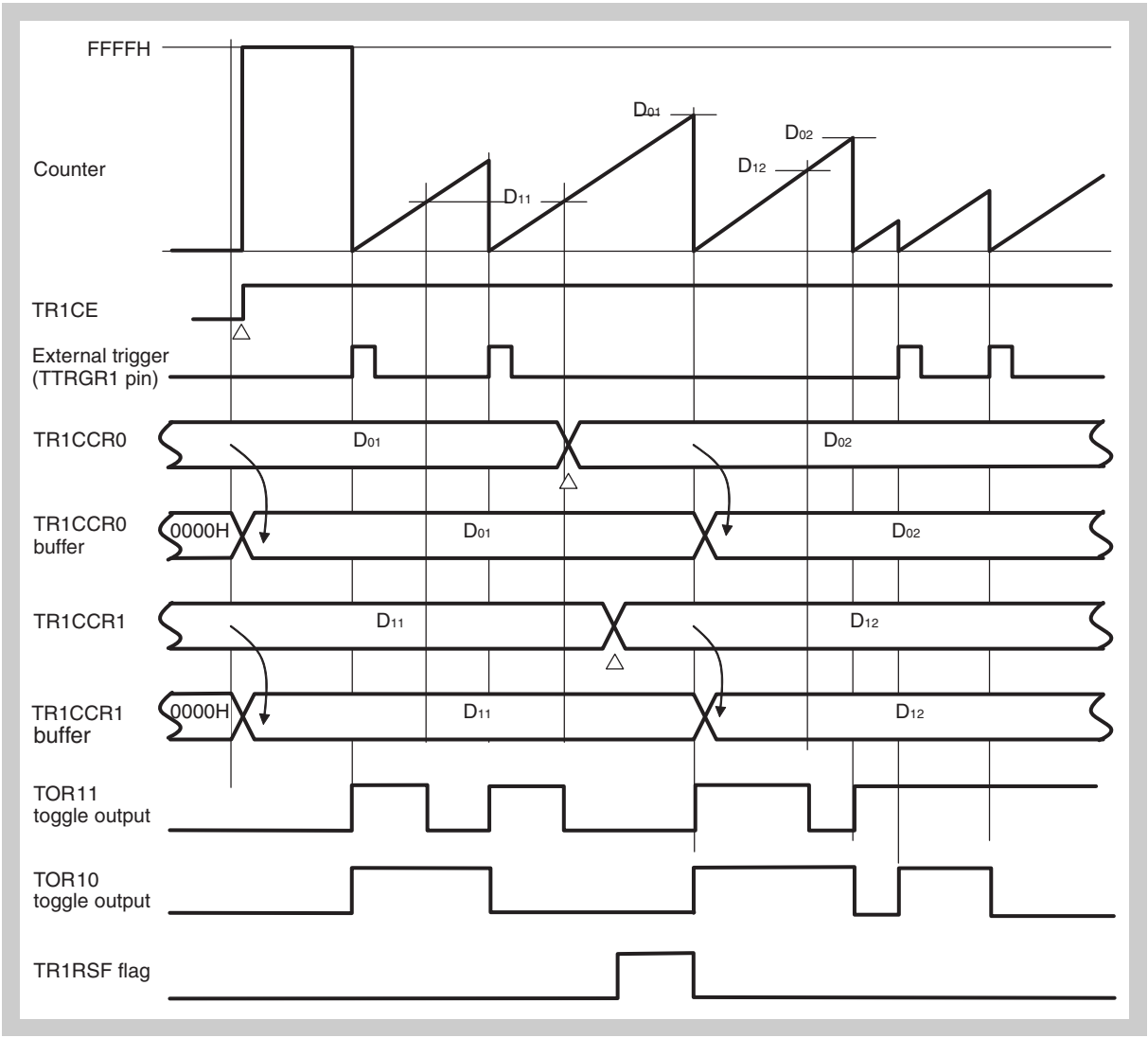


Figure 10-52 Basic operation timing in external trigger pulse output mode

The example above assume that values of TR1CCR0 and TR1CCR1 registers are rewritten, TOR10 and TOR11 are output (TR1OE0 = TR1OE1 = 1, TR1OL0 = TR1OL1 = 0)

Remarks

1. D01, D02: Setting values of TR1CCR0 register (0000_H to $FFFF_H$)
   D11, D12: Setting values of TR1CCR1 register (0000_H to $FFFF_H$)
2. TOR11 (PWM) duty =
   (setting value of TR1CCR1 register) × (count clock cycle)
3. TOR11 (PWM) cycle =
   (setting value of TR1CCR0 register + 1) × (count clock cycle)
4. Pin TOR10 is toggled when the counter is cleared immediately following count start.

10.10.4 One-shot pulse mode

(1) Outline of one-shot pulse mode

When, in the one-shot pulse mode, the duty is set to the TRnCCR0 register, the output duty delay value is set to the TRnCCR1 to TRnCCR5 registers, and bit TRnCTL0.TRnCE is set to 1, external trigger input (TTRGR1 pin of TMR1) wait results, with the counter remaining stopped at $FFFF_H$. Upon detection of the valid edge of external trigger input (TTRGR1 pin of TMR1), or when bit TRnCTL0.TRnEST is set to 1, count up starts. The TORn1 to TORn5 pins become high level upon a match between the counter and TRnCCR1 to TRnCCR5 registers. Moreover, upon a match between the counter and TRnCCR0 register, the TORn1 to TORn5 pins become low level, and the counter is cleared to 0000H and then stops. The TORn0 pin performs toggle output and a compare match interrupt (INTTRnCC0) is output during the count operation. Upon a match between the counter and TRnCCR1 to TRnCCR5 buffer registers, the corresponding compare match interrupts (INTTRnCC1 to INTTRnCCR5) are output.

The TRnCCR0 and TRnCCR1 registers can be rewritten using the anytime write method, regardless of the value of bit TTRnCTL0.RnCE.

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 one-shot pulse mode, registers TRnCCR0 to TRnCCR3 have their function fixed as compare registers, so the capture function cannot be used.

(2) One-shot pulse mode operation list

(a) Compare registers

Register	Rewriting method	Rewriting during operation	Function
TRnCCR0	Anytime rewrite	Possible	Compare & clear value (Cycle)
TRnCCR1 to TRnCCR3	Anytime rewrite	Possible	Compare value (Output delay value)
TRnCCR4, TRnCCR5	Anytime rewrite	Possible	Compare value (Output delay value)

(b) Input pins

Pin	Function
TIR10 to TIR13	–
TTRGR1	Counter start through external trigger input
TEVTR1	–

(c) Output pins

Pin	Function
TORn0	Active at count start, inactive upon TRnCCR0 register match
TORn1 to TORn5	Active upon TRnCCRm register match, inactive upon TRnCCR0 register match (m = 1 to 5)
TORn6, TORn7	–

(d) Interrupts

Interrupt	Function
INTTR1CCm	TRnCCRm register compare match (m = 0 to 5)
INTTR1OV	–
INTTR1ER	–

(3) Operation Flow in one-shot pulse mode

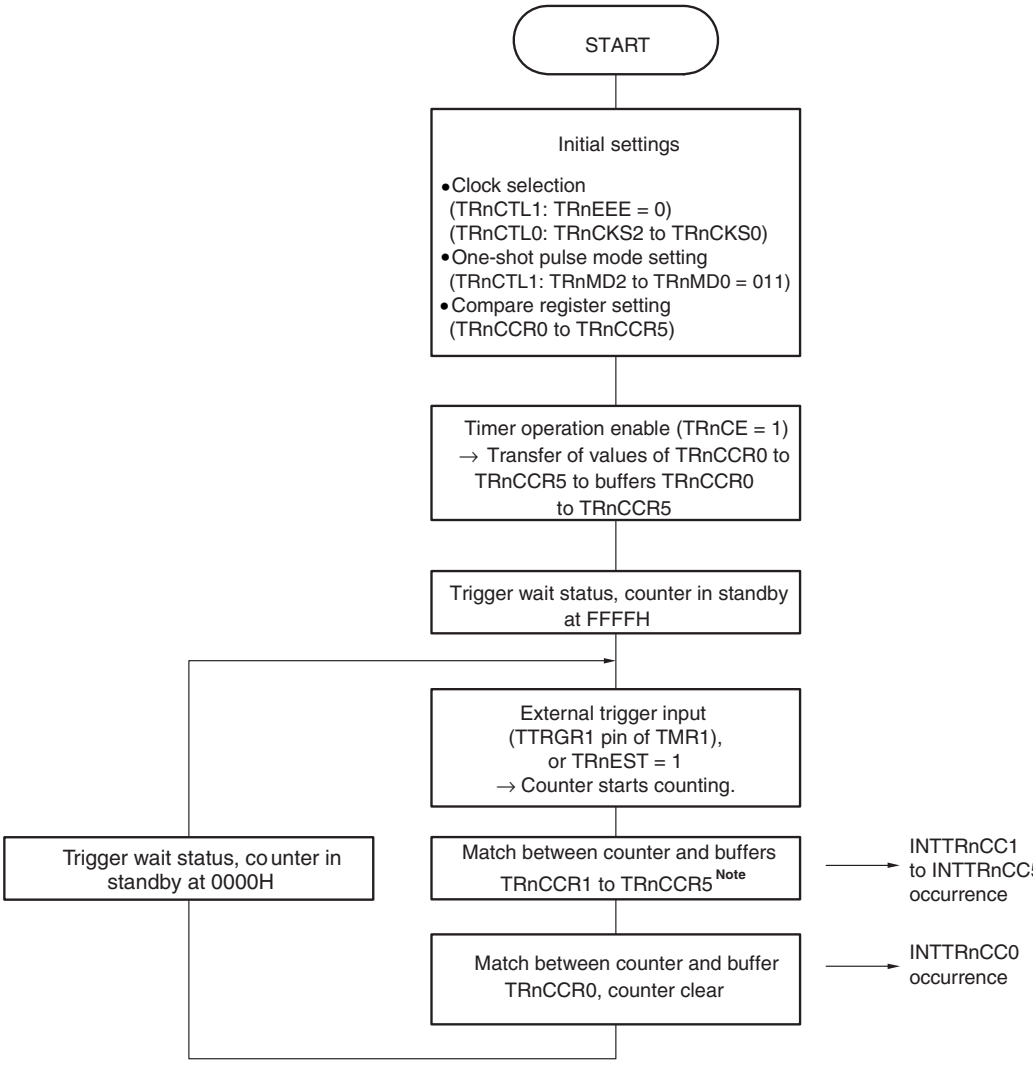


Figure 10-53 Basic operation flow in one-shot pulse mode

Note The counter is not cleared upon a match between the counter and the TRnCCR1 to TRnCCR5 buffer registers.

Caution
1. In one-shot pulse mode, ensure that TRnCTL1.TRnEEE bit is set to 0.
2. Once the counter is started and counts up, the trigger input is ignored. The counter is cleared only by match with the TRnCCR0 register.

(4) Operation timing in one-shot pulse mode

The timing example below assumes that TOR10 and TOR11 are set to output with active H-level (TRnOE0 = TRnOE1 = 1, TRnOL0 = TRnOL1 = 0)

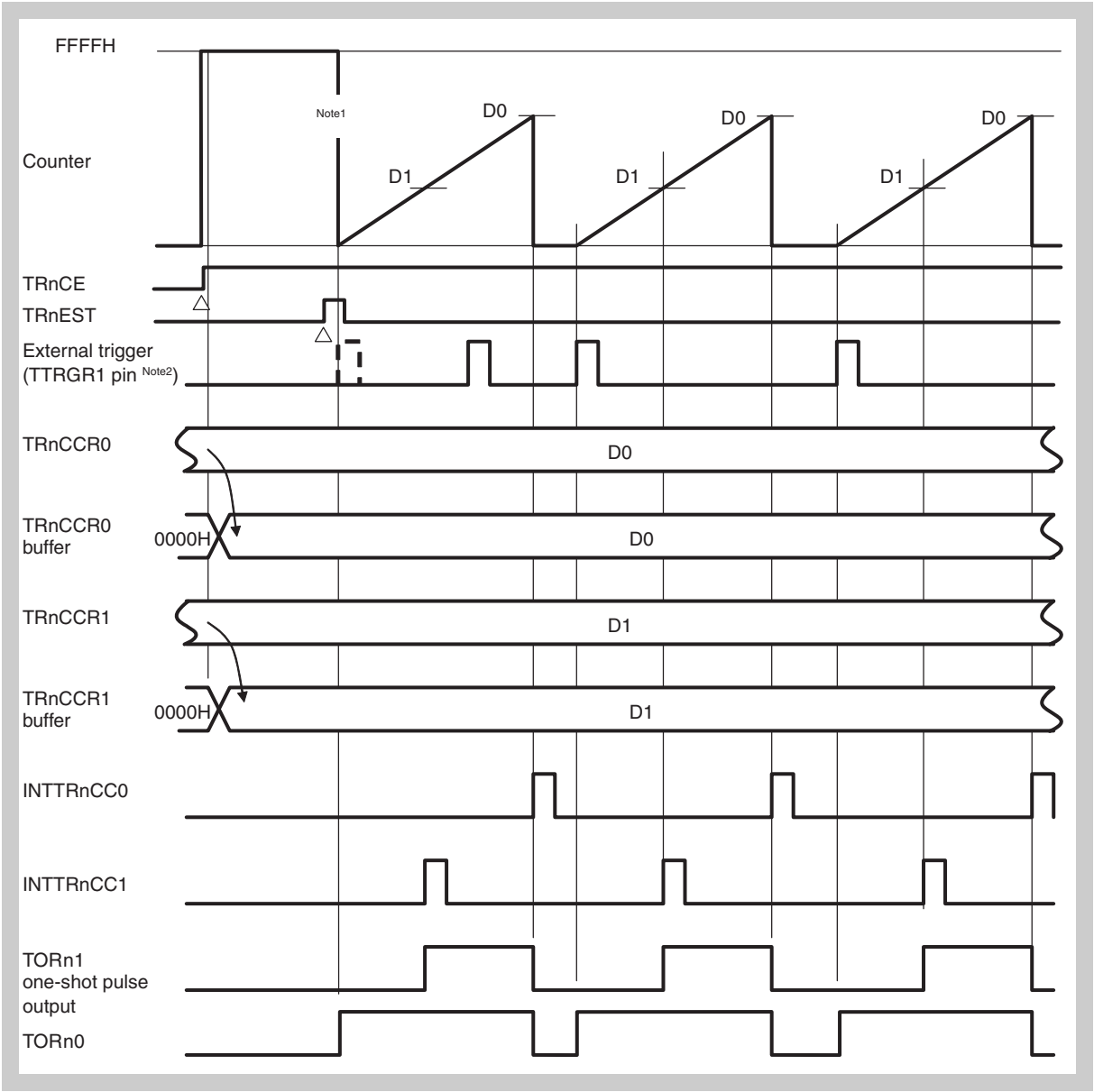


Figure 10-54 Basic operation timing in one-shot pulse mode

Note
1. Count up starts when TRnEST bit is set to 1 or TTRGRn is input.
2. Trigger input available for TMR1 only.

Remarks
1. D0: Setting value of TRnCCR0 register (0000H to FFFFH)
   D1: Setting value of TRnCCR1 register (0000H to FFFFH)
2. TORn1 (output delay) =
   (setting value of TRnCCR1 register) × (count clock cycle)
3. TORn1 (output pulse width) =
   {(setting value of TRnCCR0 register +1)
   - (setting value of TRnCCR1 register)} × (count clock cycle)

10.10.5 PWM mode

(1) Outline of PWM mode

When, in the PWM mode, the duty is set to the TRnCCR1 to TRnCCR5 registers, the cycle is set to the TRnCCR0 register, and TRnCTL0.TRnCE = 1 is set, variable duty PWM output is performed from pins TORn1 to TORn5.

Simultaneously with the start of count up operation, pins TORn1 to TORn5 become lactive level, and upon a match between the counter and the TRnCCR1 to TRnCCR5 registers, the corresponding output becomes inactive level. On each match with the TRnCCR0 register, the TORn1 to TORn5 pins become active level. The TORn0 pin performs toggle output upon a match with the TRnCCR0 buffer register.

During count operation, a compare match interrupt (INTTRnCC0) is output upon a match between the counter and TRnCCR0 register, and compare match interrupts (INTTRnCC1 to INTTRnCC5) are output upon a match between the counter and the corresponding TRnCCR1 to TRnCCR5 registers.

The TRnCCR0 to TRnCCR5 registers can be rewritten during count operation. Compare register reload occurs upon a match between the counter value and the TRnCCR0 buffer register. However, since the next reload timing becomes valid when the TRnCCR1 register is rewritten, write even the same value to the TRnCCR1 register when a reload of any other register should be performed. Reloading is not performed if only the TRnCCR0 register is rewritten.

In the PWM mode, the TRnCCR0 to TRnCCR3 registers have their function fixed as compare registers, so the capture function cannot be used.

(2) PWM mode operation list

(a) Compare registers

Register	Rewriting method	Rewriting during operation	Function
TRnCCR0	Anytime rewrite	Possible	Compare & clear value (Cycle)
TRnCCR1 to TRnCCR3	Anytime rewrite	Possible	Compare value (Duty)
TRnCCR4, TRnCCR5	Anytime rewrite	Possible	Compare value (Duty)

(b) Input pins

Pin	Function
TIR10 to TIR13	–
TTRGR1	–
TEVTR1	–

(c) Output pins

Pin	Function
TORn0	Toggle output upon TRnCCR0 register compare match
TORn1 to TORn5	PWM output upon TRnCCRm register compare match (m = 1 to 5)
TORn6	–
TORn7	Pulse output through A/D conversion trigger

(d) Interrupts

Interrupt	Function
INTTR1CCm	TRnCCRm register compare match (m = 0 to 5)
INTTR1OV	–
INTTR1ER	Error

(3) Operation flow in PWM mode

In the operation flow below the values of TRnCCR0 to TRnCCR5 registers are rewritten not during timer operation.

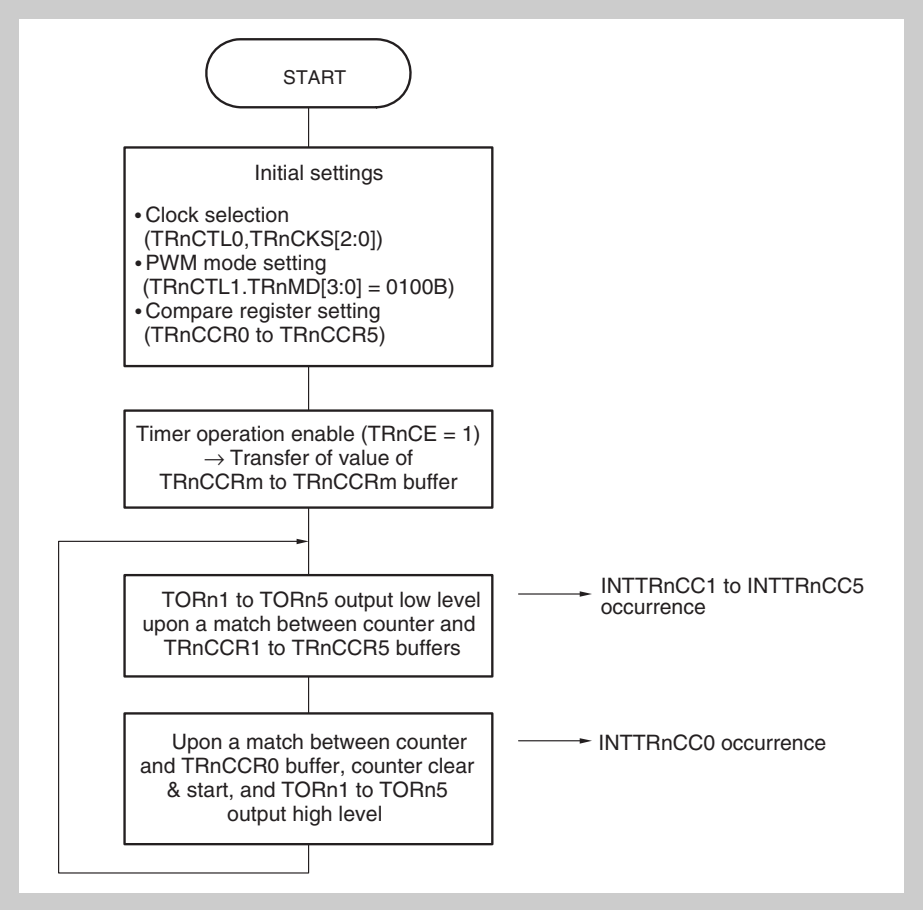


Figure 10-55 Basic operation flow in PWM mode (1/2)

In the operation flow below the values of TRnCCR0 to TRnCCR5 registers are rewritten during timer operation.

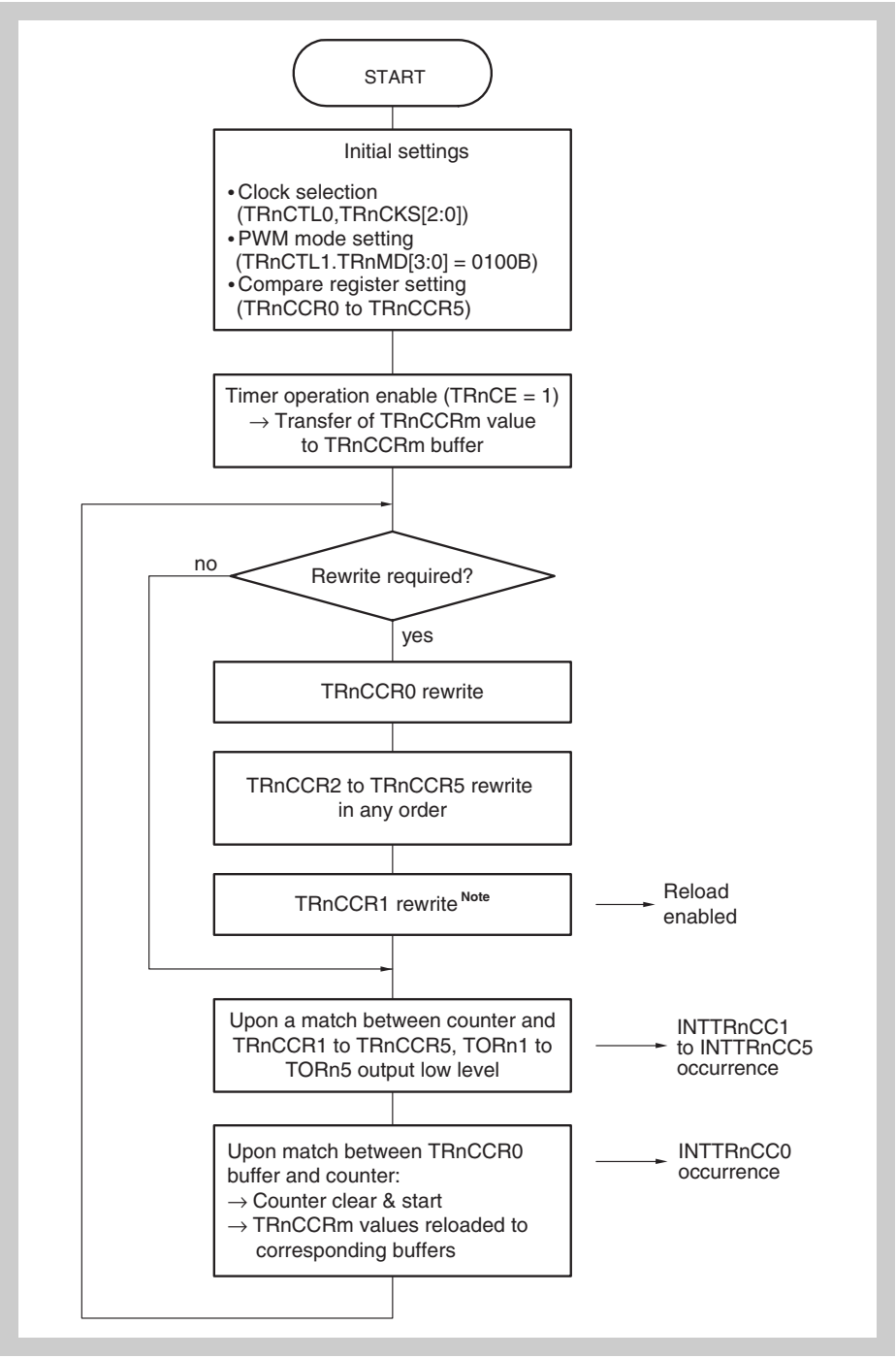


Figure 10-56 Basic operation flow in PWM mode (2/2)

Note Rewrite TRnCCR1 register always as last to enable the reload.

Remark m = 0 to 5

(4) Operation timing in PWM mode

In the timing example below only the value of TRnCCR1 is rewritten, and TORn0 and TORn1 are output (TRnOE0, 1 = 1, TRnOL0, 1 = 0).

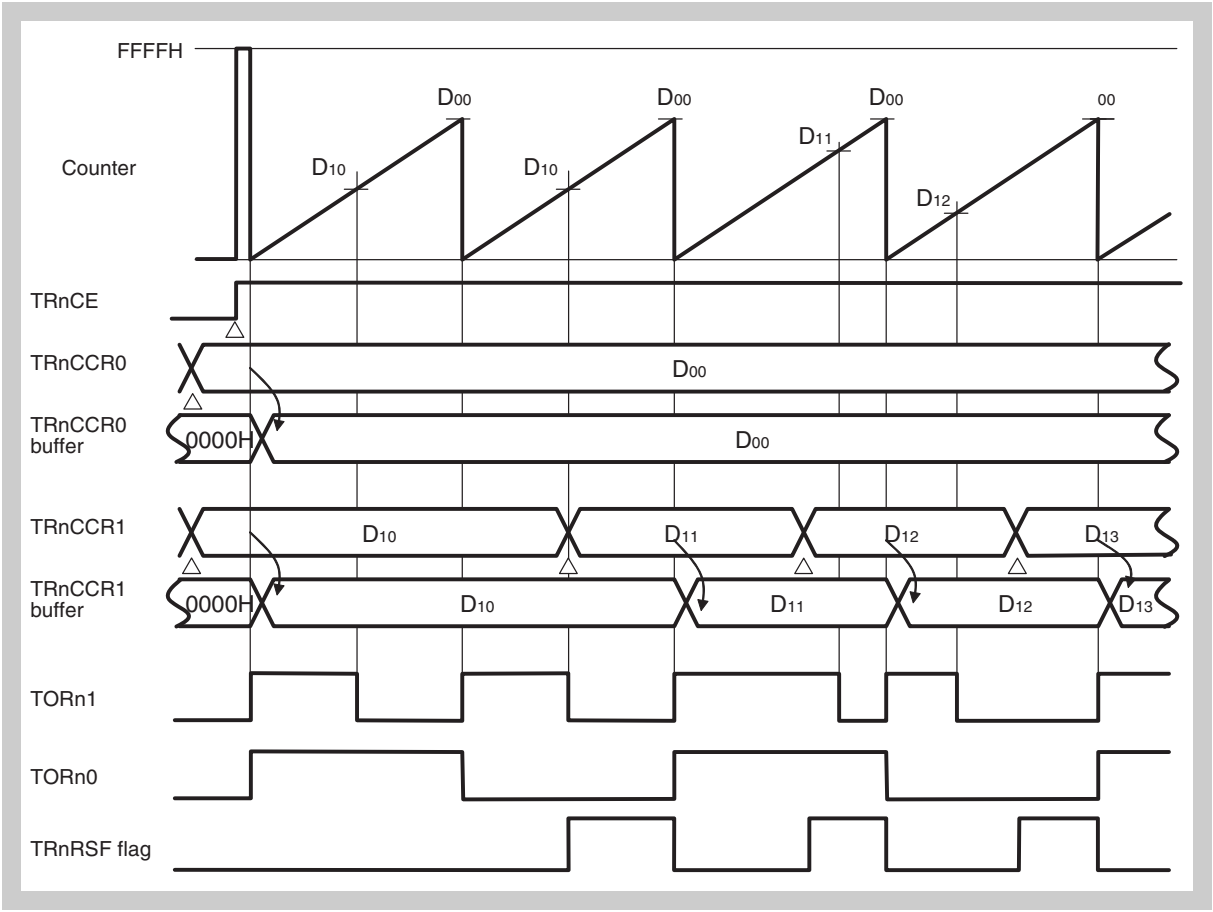


Figure 10-57 Basic operation timing in PWM mode (1/2)

Remarks

1. D00: Setting value of TRnCCR0 register (0000_H to $FFFF_H$)
   D10, D11, D12, D13: Setting values of TRnCCR1 register (0000_H to $FFFF_H$)
2. TORn1 (PWM) duty =
   (setting value of TRnCCR1 register) × (count clock cycle)
3. TORn1 (PWM) cycle =
   (setting value of TRnCCR0 register + 1) × (count clock cycle)
4. TORn0 is toggled immediately following counter start and at
   (setting value of TRnCCR0 register + 1) × (count clock cycle)

In the timing example below the values of TRnCCR0 and TRnCCR1 register are rewritten, TORn0 and TORn1 are output (TRnOE0 = TRnOE1 = 1, TRnOL0 = TRnOL1 = 0)

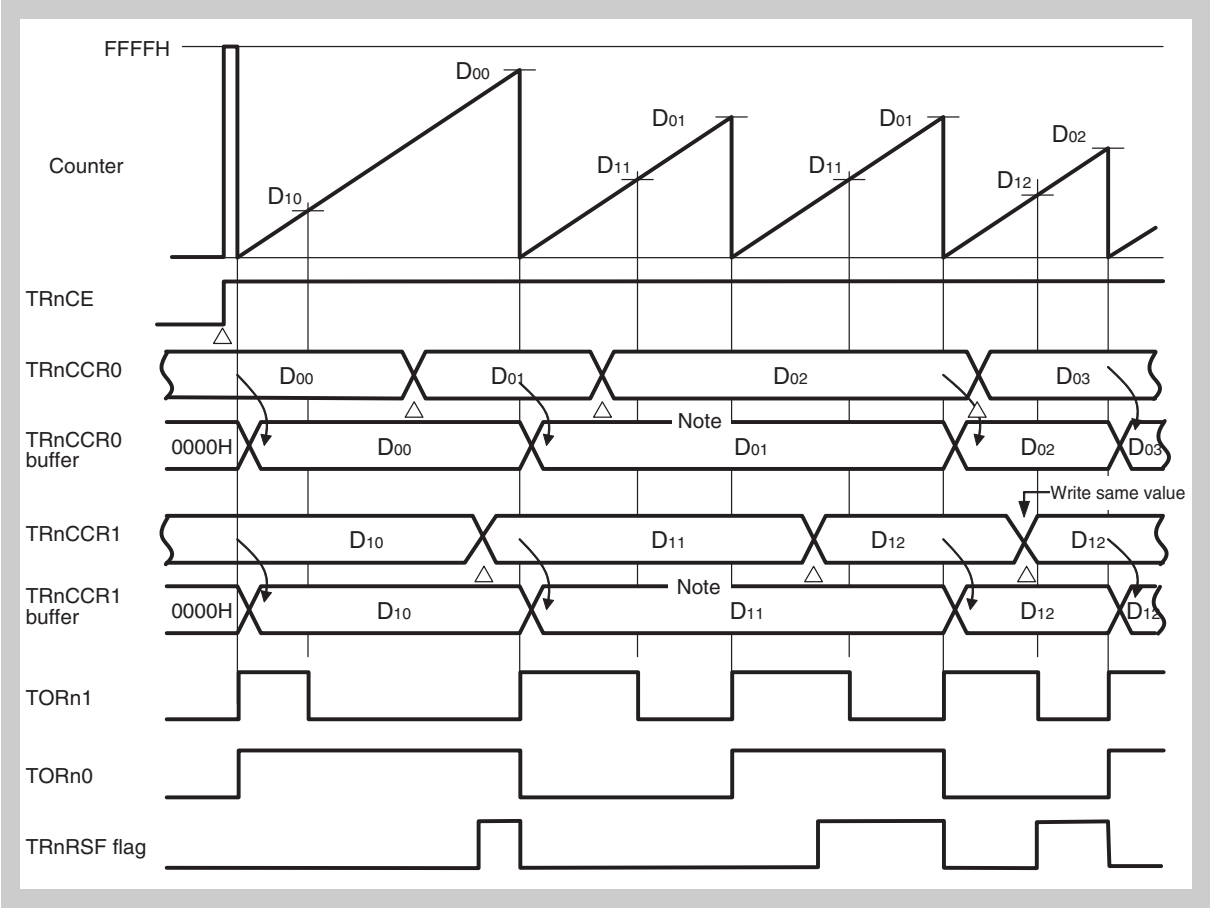


Figure 10-58 Basic operation timing in PWM mode (2/2)

Note The TRnCCR1 register was not written to, so transfer to the TRnCCR0 buffer register was not performed. Held until the next reload timing.

Remarks

1. D00, D01, D02, D03: Setting values of TRnCCR0 register (0000_H to $FFFF_H$)
   D10, D11, D12, D13: Setting values of TRnCCR1 register (0000_H to $FFFF_H$)
2. The TORn0 and TORn1 pins become high level at timer count start.

10.10.6 Free-running mode

(1) Outline of free-running mode

The operation in free-running mode depends TRnOPT0.TRnCCS[3:0] bits.

For details on the the free-running mode refer to *"Operation Flow in Free-Running Mode" on page 366*.

(2) Free-running mode operation list

(a) Compare registers

Register	Rewriting method	Rewriting during operation	Function
TRnCCR0	Anytime rewrite[a]	Possible[a]	Capture or compare value
TRnCCR1 to TRnCCR3	Anytime rewrite[a]	Possible[a]	Capture or compare value
TRnCCR4, TRnCCR5	Anytime rewrite[a]	Possible[a]	Compare value

(b) Input pins

Pin	Function
TIR10 to TIR13	Input capture trigger, transfer counter value to TR1CCRm register (m = 0 to 3)[b]
TTRGR1	–
TEVTR1	–

(c) Output pins

Pin	Function
TORn0 to TORn5	Toggle output upon TRnCCRm register compare match (m = 0 to 5)[a]
TORn6, TORn7	–

(d) Interrupts

Interrupt	Function
INTTR1CCm	TRnCCRm register compare match (m = 0 to 5)[a], or occurrence of TIR1m capture input signal (m = 0 to 3)[b]
INTTR1OV	Overflow
INTTR1ER	–

a) 1When compare function is selected.
TMR0: compare function is permitted only for TR0CCRm registers (m = 0 to 3).
TMR1: compare function is permitted for any TR1CCRm register (m = 0 to 5).

b) 2When capture function is selected for the corresponding TR1CCRm register of TMR1 (m = 0 to 3). The capture function is not available for TMR0.

(3) Operation Flow in Free-Running Mode

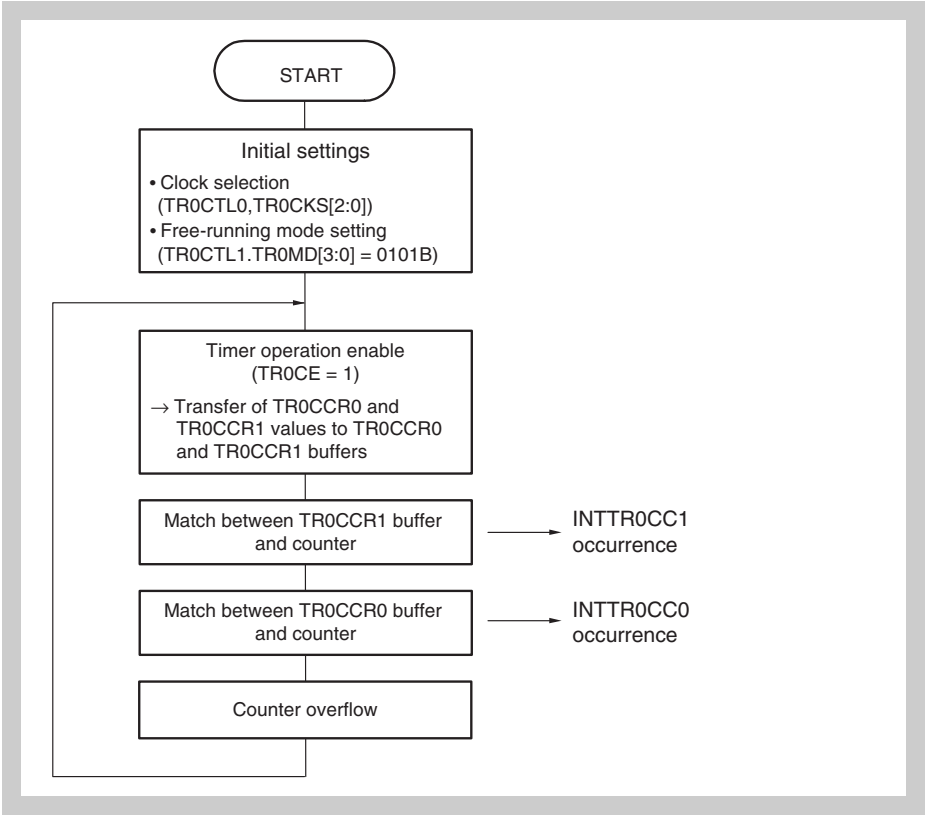


Figure 10-59 Operation flow in free-running mode of TMR0

Remark This is an example when using the TRnCCR0 and TRnCCR1 registers. The TRnCCR2 and TRnCCR3 registers can be applied in the same way.

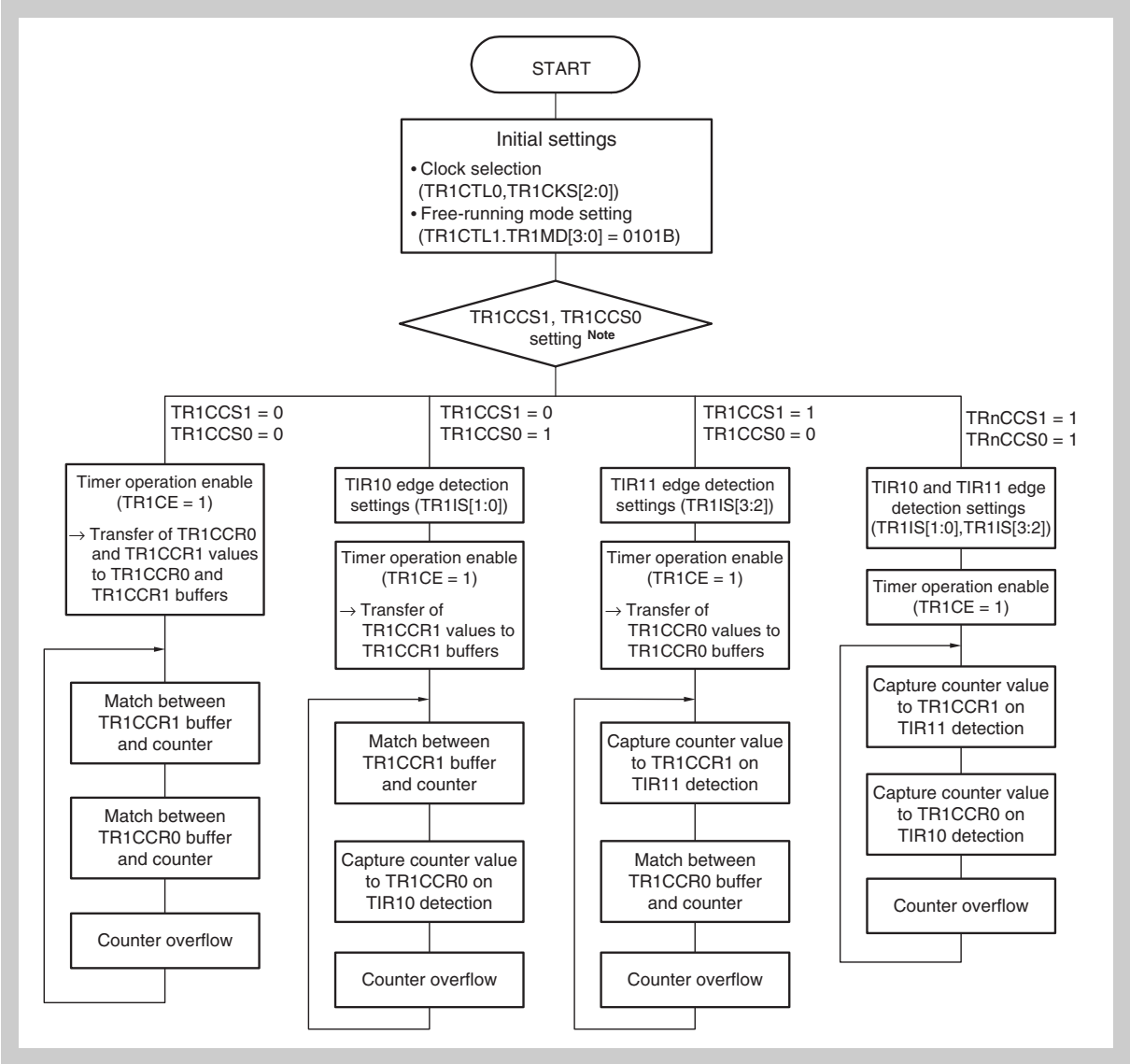


Figure 10-60 Operation flow in free-running mode of TMR1

Remark This is an example when using the TRnCCR0 and TRnCCR1 registers. When using the TRnCCR2 and TRnCCR3 registers, the operation is controlled in the same manner via bits TRnCCS3 and TRnCCS2.

(4) Compare function (TRnCCS1 = 0, TRnCCS0 = 0)

When TRnCTL0 register bit TRnCE is set to 1, the counter counts from 0000_H to $FFFF_H$. An overflow interrupt (INTTRnOV) is output when the counter value changes from $FFFF_H$ to 0000_H, and the counter is cleared. The count operation is performed in the free-running mode until TRnCE = 0 is set. Moreover, during count operation, a compare match interrupt (INTTRnCC0) is output upon a match between the counter and TRnCCR0 buffer register, and a compare match interrupt (INTTRnCC1) is output upon a match between the counter and TRnCCR1 buffer register.

The TRnCCR0 and TRnCCR1 registers can be rewritten using the anytime write method, regardless of the value of the TRnCE bit.

The TORn0 and TORn1 pins are toggle output controlled when bits register TRnOE0 and TRnOE1 of the TRnIOC0 register are set to 1.

The timing example below assumes that values of TRnCCR0 and TRnCCR1 registers are rewritten, TORn0 and TORn1 are output
(TRnOE0 = TRnOE1 = 1, TRnOL0 = TRnOL1 = 0)

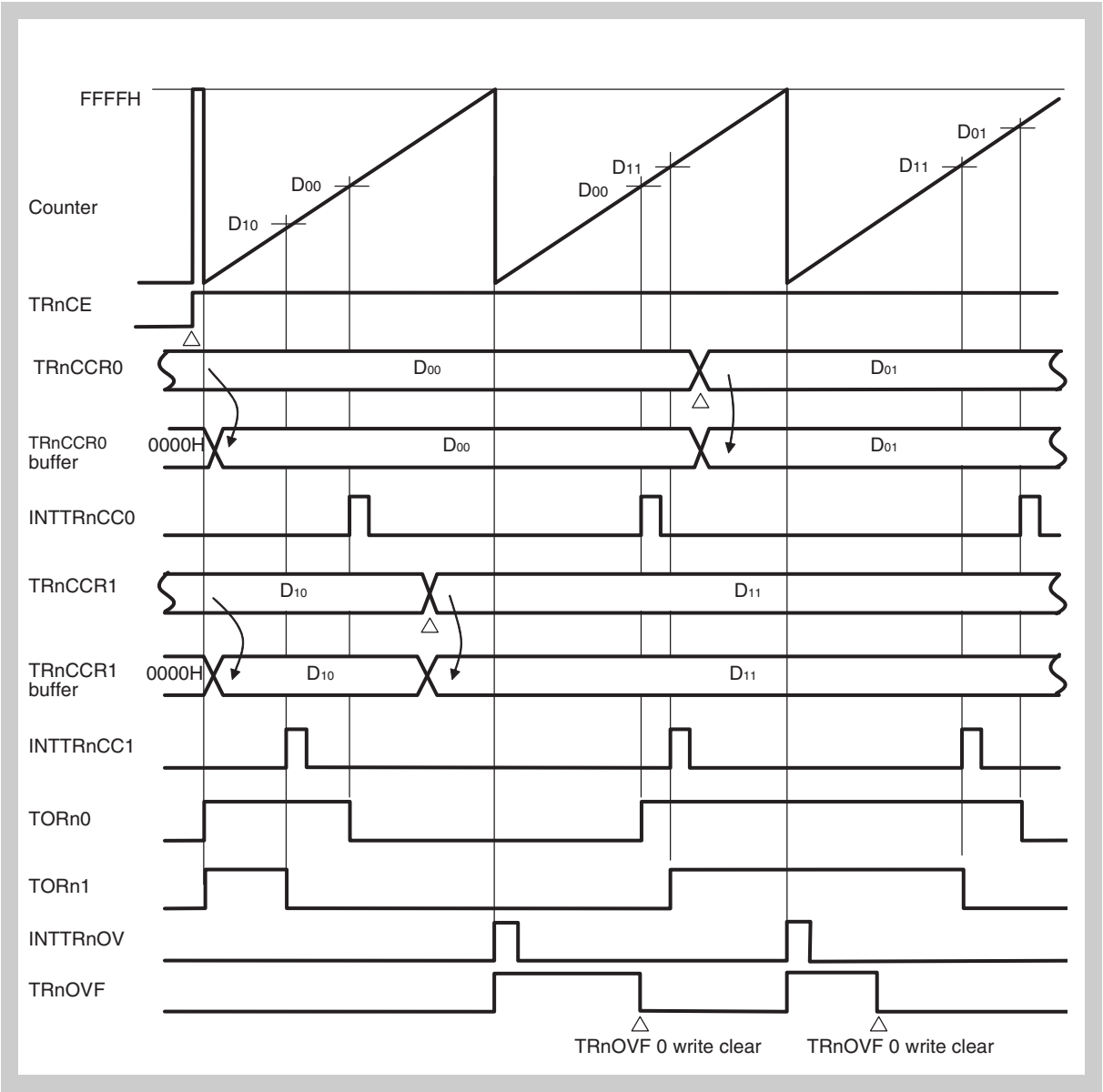


Figure 10-61 Basic operation timing in free-running mode (compare function)

Remarks

1. D00, D01: Setting values of TRnCCR0 register (0000_H to $FFFF_H$)
   D10, D11: Setting values of TRnCCR1 register (0000_H to $FFFF_H$)
2. TORn0 (toggle) width =
   (setting value of TRnCCR0 register + 1) × (count clock cycle)
3. TORn1 (toggle) width =
   (setting value of TRnCCR1 register + 1) × (count clock cycle)
4. Pins TORn0 and TORn1 become high level at count start.

(5) Capture function (TRnCCS1 = 1, TRnCCS0 = 1)

When TRnCTL0 register bit TRnCE is set to 1, the counter counts from 0000_H to $FFFF_H$. An overflow interrupt (INTTRnOV) is output when the value of the counter changes from $FFFF_H$ to 0000_H, and the counter is cleared. The count operation is performed in the free-running mode until TRnCE = 0 is set. When, during count operation, the counter value is captured to the TRnCCR0 and TRnCCR1 registers through detection of the valid edge of capture input (TIRn1, TIRn0), a capture interrupt (INTTRnCC0, INTTRnCC1) is output.

Regarding capture in the vicinity of overflow ($FFFF_H$), judgment is possible with the overflow flag (TRnOVF). However, judgment with the TRnOVF flag is not possible when the capture trigger interval is such that it includes two overflow occurrences (2 or more free-running cycles).

Caution

1. In free-running mode the external event clock input (TEVTR1) is prohibited (TR1CTL1.TR1EEE = 0).
2. When an internal count clock ≤ $f_{XX}/16$ (TRnCTL0.TRnCKS2-0) is selected in free-running mode, the TRnCCR0 and TRnCCR1 registers are used as capture registers, the a value of $FFFF_H$ will be captured if a valid signal edge is input before the first count up.

The timing example below assumes that TORn0 and TORn1 are not output (TRnOE0 = TRnOE1 = 0, TRnOL0 = TRnOL1 = 0)

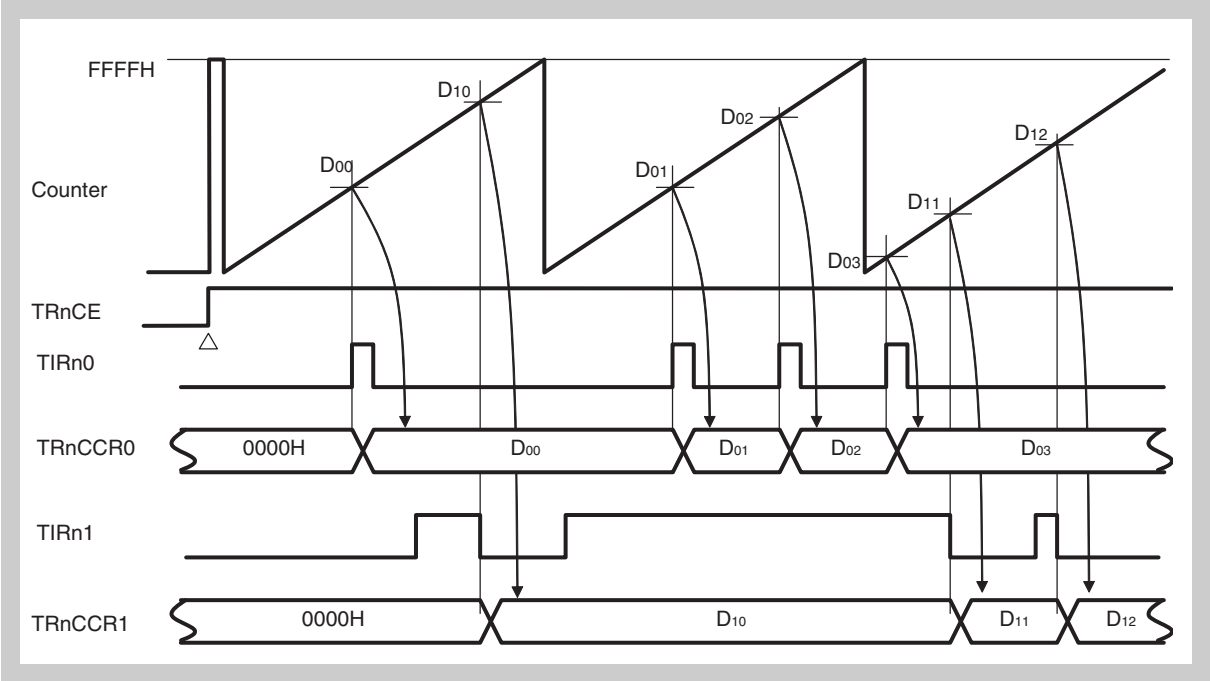


Figure 10-62 Basic operation timing in free-running mode (capture function)

Remarks

1. D00, D01: Values captured to TRnCCR0 register (0000_H to $FFFF_H$)
   D10, D11: Values captured to TRnCCR1 register (0000_H to $FFFF_H$)
2. TIRn0: Setting to rising edge detection (TRnIOC1.TRnIS[1:0] = 01_B)
   TIRn1: Setting to falling edge detection (TRnIOC1.TRnIS[3:2] = 10_B)

(6) Compare/capture function (TRnCCS1 = 0, TRnCCS0 = 1)

When bit TRnCTL0.TRnCE is set to 1, the counter counts from 0000_H to $FFFF_H$, an overflow interrupt (INTTRnOV) is output when the value of the counter changes from $FFFF_H$ to 0000_H, and the counter is cleared. The count operation is performed in the free-running mode until TRnCE = 0 is set. The TRnCCR1 register is used as a compare register, and as the interval function upon a match between the counter and TRnCCR1 register, a compare match interrupt (INTTRnCC1) is output. Since the TRnCCR0 register is set to the capture function, the TORn0 pin cannot be controlled even when bit TRnIOC0.TRnOE0 is set to 1.

Caution

1. In free-running mode the external event clock input (TEVTR1) is prohibited (TR1CTL1.TR1EEE = 0).
2. When an internal count clock ≤ f_{XX}/16 (TRnCTL0.TRnCKS2-0) is selected in free-running mode, and TRnCCR0 register is used as capture register, the a value of FFFFH will be captured if a valid signal edge is input before the first count up.

The timing example below assumes that TRnCCR1 is rewritten, TORn0 and TORn1 are output (TRnOE0 = TRnOE1 = 1, TRnOL0 = TRnOL1 = 0)

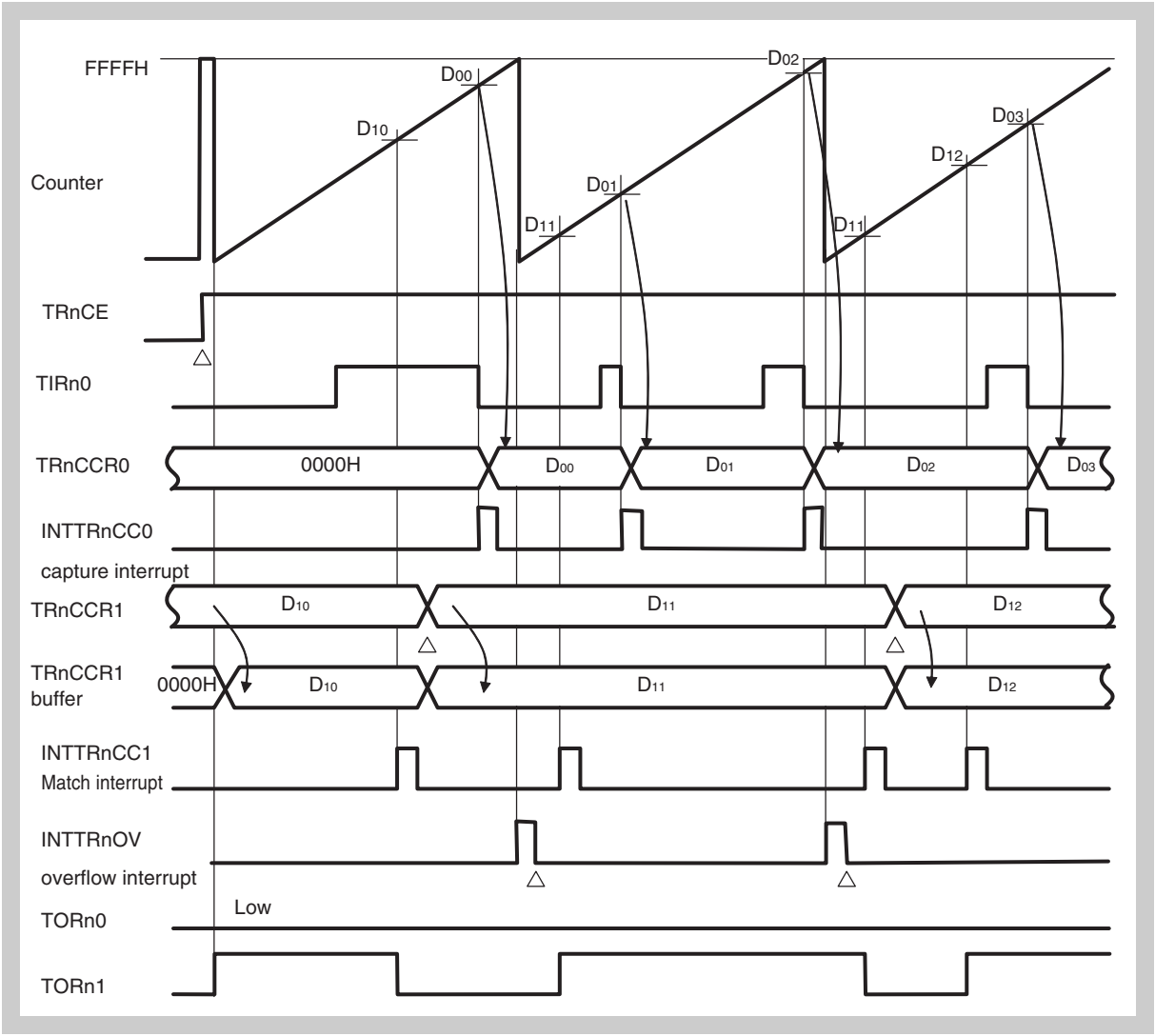


Figure 10-63 Basic operation timing in free-running mode (compare/capture function)

Remarks

1. D00, D01: Setting values of TRnCCR1 register (0000_H to $FFFF_H$)
   D10, D11, D12, D13, D14, D15: Values captured to TRnCCR0 register (0000_H to $FFFF_H$)
2. TIRn0: Setting to rising edge detection (TRnIOC1.TRnIS[1:0] = 11_B)

(7) Overflow flag

When, in the free-running mode, the counter overflows from $FFFF_H$ to 0000_H, the overflow flag (TRnOVF) is set to “1”, and an overflow interrupt (INTTRnOV) is output.

The overflow flag is cleared through 0 write by the CPU.

Note The overflow flag is not cleared by just being read.

10.10.7 Pulse width measurement mode (TMR1 only)

(1) Outline of pulse width measurement mode

In the pulse width measurement mode, counting is performed in free-running mode. The counter value is saved to the TR1CCR0 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 case of TIR11 to TIR13 pin edge detection, pulse width measurement can be similarly performed by using the TR1CCR1 to TR1CCR3 registers.

Caution
1. In the pulse width measurement mode the external event clock input (TEVTR1) is prohibited (TR1CTL1.TR1EEE = 0).
2. When an internal count clock ≤ f_{XX}/16 (TP1CTL0.TP1CKS[2:0]) is selected in pulse width measurement mode, and a valid signal edge is input before the first count up, the a value of $FFFF_H$ will be captured in the corresponding TP1CCR0 or TP1CCR1 register.

(2) Pulse width measurement mode operation list

(a) Compare registers

Register	Rewriting method	Rewriting during operation	Function
TR1CCR0	–	–	Capture value
TR1CCR1 to TR1CCR3	–	–	Capture value
TR1CCR4, TR1CCR5	–	–	–

(b) Input pins

Pin	Function
TIR10 to TIR13	Input capture trigger, transfer counter value to TR1CCRm register (m = 0 to 3)
TTRGR1	–t
TEVTR1	–

(c) Output pins

Pin	Function
TOR10 to TOR15	–
TOR16, TOR17	–

(d) Interrupts

Interrupt	Function
INTTR1CCm	TIR1m capture (m = 0 to 3)
INTTR1CC4, INTTR1CC5	–
INTTR1OV	Overflow
INTTR1ER	–

(3) Operation timing in pulse width measurement mode

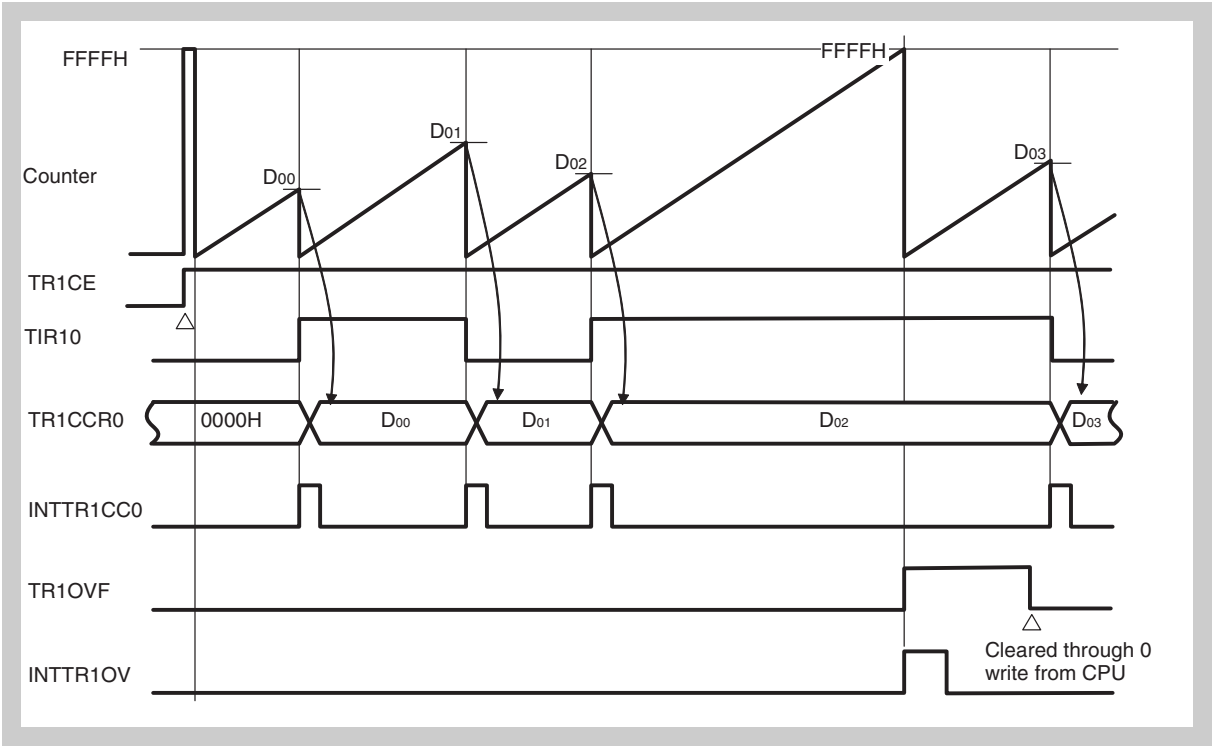


Figure 10-64 Basic operation timing in pulse width measurement mode

Remarks
1. D00, D01, D02, D03: Values captured to TR1CCR0 register (0000_H to $FFFF_H$)
2. TIR10: Setting to rising edge/falling edge (both edges) detection (TRnIOC1.TR1IS[1:0] = 11_B)

10.10.8 Triangular wave PWM mode

(1) Outline of triangular wave PWM mode

In the triangular wave PWM mode, similarly to in the PWM mode, when the duty is set to the TRnCCR1 to TRnCCR5 registers, the cycle is set to the TRnCCR0 register, and TRnCE = 1 is set, variable duty and cycle type triangular wave PWM output is performed from pins TORn1 to TORn5. The TORn0 pin is toggle output upon a match with the TRnCCR0 buffer register and upon counter underflow. Upon a match between the counter and TRnCCR0 register during count operation, compare match interrupts (INTTRnCC0 to INTTRnCC5) are output, and upon a match between the counter and TRnCCR1 to TRnCCR5 registers, a compare match interrupt (INTTRnCC1) is output. Moreover, upon counter underflow, an overflow interrupt (INTTRnOV) is output.

The TRnCCR0 to TRnCCR5 registers can be rewritten during count operation. Compare register reload occurs upon a match between the counter value and the TRnCCR0 buffer register. However, since the next reload timing becomes valid when the TRnCCR1 register is written to, write the same value to the TRnCCR1 register even when wishing to rewrite only the value of the TRnCCR0 register. Reloading is not performed if only the TRnCCR0 register is rewritten. The reload timing is the underflow timing.

In the triangular wave PWM mode, the TRnCCR0 to TRnCCR3 registers have their function fixed as compare registers, so the capture function cannot be used.

Note In the triangular wave PWM mode, set the TRnCCR0 register to a value of $0000_H \leq$ TRnCCR0 $\leq$ $FFFE_H$.

(2) Triangular wave PWM mode operation list

(a) Compare registers

Register	Rewriting method	Rewriting during operation	Function
TRnCCR0	Reload	Possible	1/2 of cycle
TRnCCR1 to TRnCCR3	Reload	Possible	1/2 of cycle
TRnCCR4, TRnCCR5	Reload	Possible	1/2 of cycle

(b) Input pins

Pin	Function
TIR10 to TIR13	–
TTRGR1	–
TEVTR1	–

(c) Output pins

Pin	Function
TORn0	Inactive level during counter up count, active level during down count
TORn1 to TORn5	PWM output upon TRnCCRm register compare match (m = 0 to 5)
TORn6	–
TORn7	Pulse output through A/D conversion trigger

(d) Interrupts

Interrupt	Function
INTTR1CCm	TRnCCRm register compare match (m = 0 to 5)
INTTR1OV	–
INTTR1ER	Error

(3) Operation timing in triangular wave PWM mode

The timing example below assumes that TORn0 and TORn1 are output (TRnOE0 = TRnOE1 = 1, TRnOL0 = TRnOL1 = 0)

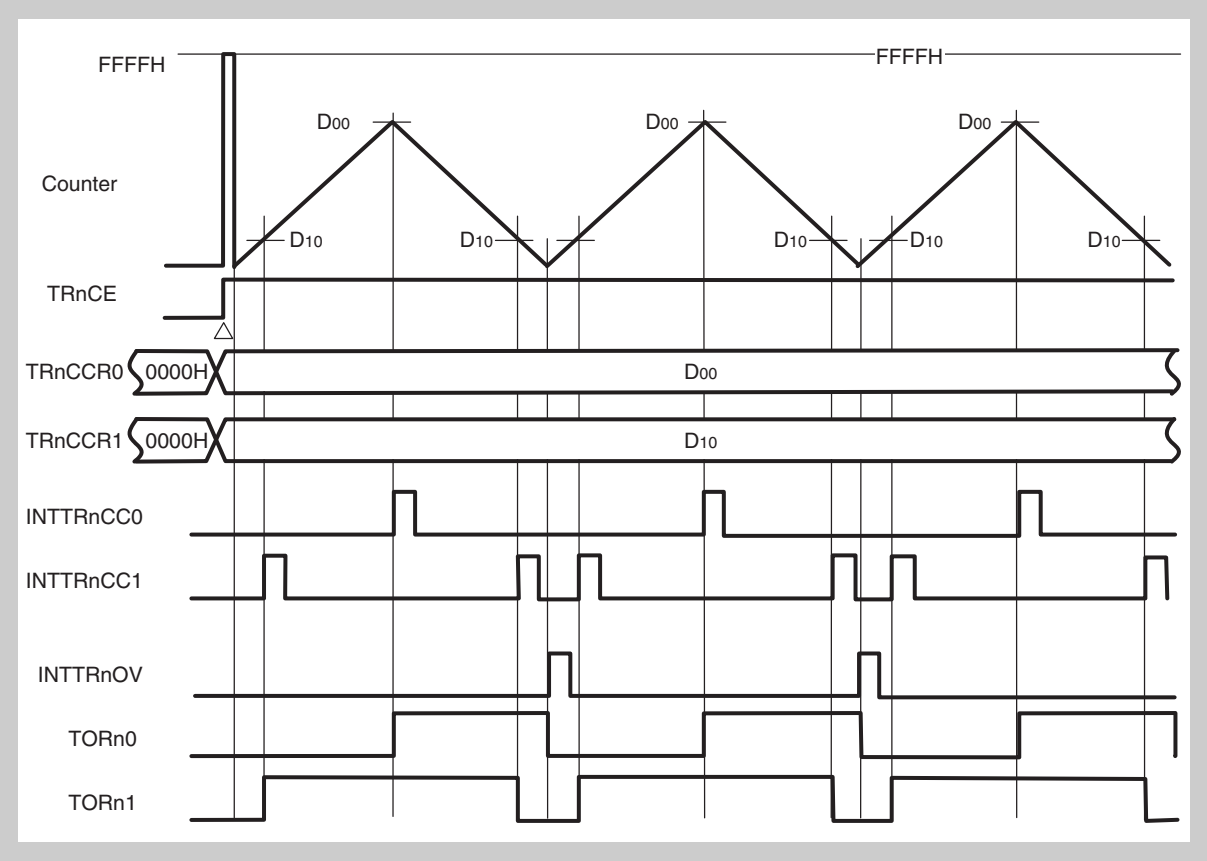


Figure 10-65 Basic operation timing in triangular wave PWM mode

10.10.9 High-accuracy T-PWM mode

(1) Outline of high-accuracy T-PWM mode

The high-accuracy T-PWM mode generates 6-phase PWM using four 16-bit counters (up/down, ±2 counts, 15 real bits) and 16-bit compare registers (LSB = additional pulse control).

The carrier wave cycle calculated with “TRnCCR0-TRnDTC0-TRnDTC1” is set to the TRnCCR0 register. The duty of the U phase, V phase, and W phase voltage data signal is set with the TRnCCR1 to TRnCCR3 registers. The dead time is set with the TRnDTC0 and TRnDTC1 registers. The TRnDTC0 register can be used to set the inverted phase (OFF) → normal phase (ON) dead time, while the TRnDTC1 register can be used to set the normal phase (OFF) → inverted phase (ON) dead time.

The counter operation consists of an up count phase with the TRnDTC0 register value as the minimum value, and a down count phase upon a match with the maximum value indicated by “TRnCCR0-TRnDTC1”.

The 10-bit counters for dead time generation (TRnDTT1 to TRnDTT3) load the setting values of the TRnDTC0 and TRnDTC1 registers upon a match between the counter and the TRnCCR1 to TRnCCR3 registers, and perform down-count.

Upon a match between the 16-bit counter and the TRnCCR1 to TRnCCR3 registers, INTTRnCC1 to INTTRnCC3, which are used as the respective compare match interrupt signals, are output. (In the 0% output vicinity and 100% output vicinity, no interrupt signal may be output.)

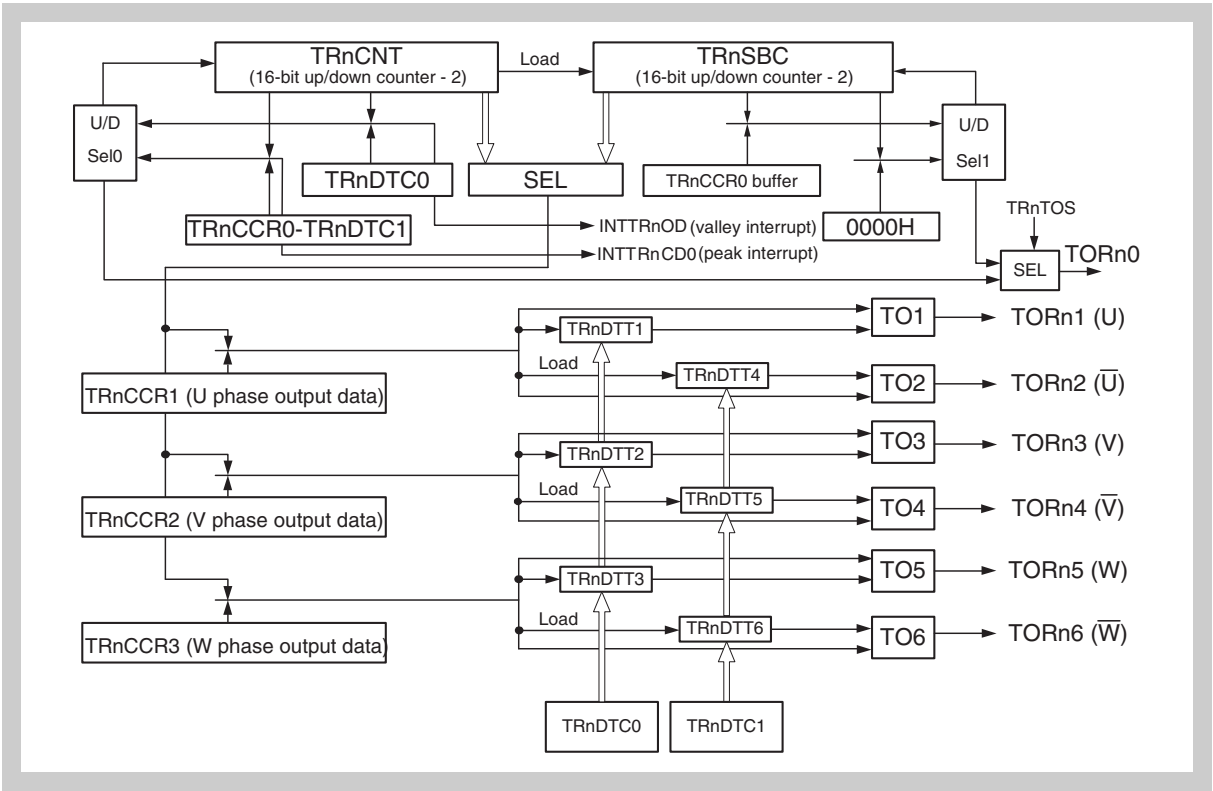


Figure 10-66 High-accuracy T-PWM mode block diagram

(2) High-accuracy T-PWM mode operation list

(a) Compare registers

Register	Rewriting method	Rewriting during operation	Function
TRnCCR0	Reload or Anywrite method	Possible	PWM cycle
TRnCCR1 to TRnCCR3	Reload or Anywrite method	Possible	PWM duty
TRnCCR4, TRnCCR5	Reload or Anywrite method	Possible	PWM duty (selectable as A/D conversion trigger)

(b) Input pins

Pin	Function
TIR10 to TIR13	–
TTRGR1	–
TEVTR1	–

(c) Output pins

Pin	Function
TORn0	Inactive level during counter or sub-counter up count, active level during down count
TORn1	PWM output upon TRnCCR1 compare match (with dead time)
TORn2	Inverted output to TORn1 (with dead time)
TORn3	PWM output upon TRnCCR2 compare match (with dead time)
TORn4	Inverted output to TORn3 (with dead time)
TORn5	PWM output upon TRnCCR3 compare match (with dead time)
TORn6	Inverted output to TORn5 (with dead time)
TORn7	Pulse output through A/D conversion trigger

(d) Interrupts

Interrupt	Function
INTTRnCCR0	TRnCCR0 compare match[a]
INTTRnCC1 to INTTRnCC5	TRnCCR1 to TRnCCR5 compare match
INTTRnOV	Overflow[b]
INTTRnER	Error
INTTRnOD	Through interrupt
INTTRnCD	Peak interrupt

a) Only when TRnDTC1 = 000_H.
b) When TRnCCR0, TRnDTC0, and TRnDTC1 registers are incorrectly set.

(3) High-accuracy T-PWM mode settings

(a) Mode settings

The high-accuracy T-PWM mode is selected by setting bits TRnCTL1.TRnMD[4:0] = 1000B.

(b) Output level/output enable settings

Set bits TRnOL0 to TRnOL7 and TRnOE0 to TRnOE7 of the TRnIOC0 and TRnIOC3 registers, to enable output level/output enable.

Pin TORn0 indicates the counter's and sub-counter's up count/down count status. The counter/sub-counter can be switched with TRnOPT7.TRnTOR bit.

Pin TORn7 is the external A/D conversion output pin. Set this pin as required.

(c) Error interrupt output enable

Set error interrupt output enable upon detection of normal phase/inverted phase simultaneous active. Error interrupt output is enabled by setting bit TRnIOC4.TRnEOC to "1". In the high-accuracy T-PWM mode, when the dead time setting is other than "000_H", the error interrupt (INTTRnER) never goes active, regardless of which value the TRnCCR0 to TRnCCR3 registers are set.

However, an error may be detected upon the occurrence of a timer Rn internal circuit fault. If the dead time setting is "000_H", a glitch may occur upon occurrence of an error interrupt (INTTRnER) at the normal phase and inverted phase switch timing.

(d) Rewrite timing for registers with reload function

Batch rewrite/anytime rewrite can be set for registers with the reload function. This setting is performed with bit TRnOPT0.TRnCMS, where the default is "0" for batch rewrite. To perform batch rewrite, be sure to set TRnOPT1 register bits TRnICE or TRnIOE.

Caution If bit TRnICE and bit TRnIOE are both "0", the reload timing does not occur.)

If anytime rewrite is selected, unintended output may occur depending on the rewrite timing.

When using the anytime rewrite function, refer to cautions in sub chapter *"Anytime rewrite" on page 305*.)

(e) Interrupt and thinning out function settings

The interrupt and thinning out function settings are performed with the TRnOPT1 register. If a peak interrupt (INTTRnCD) is required, set bit TRnICE to 1. If a valley interrupt (INTTRnOD) is required, set bit TRnIOE to 1. To use the thinning out function for peak/valley interrupts, perform settings with the TRnID[4:0] bits.

(f) Reload thinning out function setting

To set the reload timing to the same timing as the interrupt timing, set bit TRnOPT1.TRnRDE to 1.

(g) A/D conversion trigger output settings

To configure the A/D conversion trigger 0 (TRnADTRG0 signal), set bits TRnAT0[5:0] in the TRnOPT2 register.

With bits TRnAT0[5:0], peak interrupt (INTTRnCD) and valley interrupt (INTTRnOD) enable/disable is performed at the TRnCCR5 register match timing (counter up count/down count), and the TRnCCR4 register match timing (counter up count/down count).

To configure the A/D conversion trigger 1 (TRnADTRG1 signal), set bits TRnAT1[5:0] in the TRnOPT3 register.

With bits TRnAT1[5:0], peak interrupt (INTTRnCD) and valley interrupt (INTTRnOD) enable/disable is performed at the TRnCCR5 register match timing (counter up count/down count), and TRnCCR4 register match timing (counter up count/down count).

Caution To use the TORn7 pin correctly configure the TRnOPT2 and TRnOPT3 register and the TRnCCR4 and TRnCCR5 register settings accordingly.

(h) Dead time settings

The dead time settings are performed with the TRnDTC0 and TRnDTC1 registers.

The dead time can be obtained with counter operation clock cycle × TRnDTC0, TRnDTC1.

The time until TORn2, TORn4, TORn6 pin inactive level change → TORn1, TORn3, TORn5 pin active level change can be set with the TRnDTC0 register.

The time until TORn1, TORn3, TORn5 pin inactive level change → TORn2, TORn4, TORn6 pin active level change can be set with the TRnDTC1 register.

(i) Carrier wave cycle

For the carrier wave cycle, set the TRnCCR0 register using the following equation.

TRnCCR0 =
(carrier wave cycle/ counter operation clock cycle) + TRnDTC1 + TRnDTC0

For the setting value of the TRnCCR0 register, meet the following conditions keeping in mind the dead time.

TRnCCR0 > 3 × MAX (TRnDTC0, TRnDTC1) + MIN (TRnDTC0, TRnDTC1)

TRnCCR0 ≤ FFFEH

Note MAX(A,B) indicates the larger value of A and B, and MIN(A,B) indicates the smaller value of A and B.

(j) Duty (PWM width) setting

For the duty setting, perform the U phase, V phase, and W phase settings with the TRnCCR1 to TRnCCR3 registers.

The setting range of the TRnCCR1 to TRnCCR3 registers is $0000_H \leq$ TRnCCR1, TRnCCR2, TRnCCR3 $\leq$ TRnCCR0 + 1.

Do not set TRnCCR0 + 2 < TRnCCR1, TRnCCR2, TRnCCR3.

LSB (Least Significant Bit) of the TRnCCR1 to TRnCCR3 registers means the additional pulse setting. For example, if TRnCCR1 = 0003H is set, compared to when TRnCCR1 = 0002H is set, the inverted phase (pin TORn2) change is an 1-count clock delay (during counter up count).

(4) Counter operation in high-accuracy T-PWM mode

At initial value FFFEH, the TRnDTC0 value is loaded to the counter immediately after TRnCE = 1 is set, and the counter counts up in +2 steps. Then, upon a match with TRnCCR0 to TRnDTC1, the counter counts down in -2 steps. The counter operation is as follows.

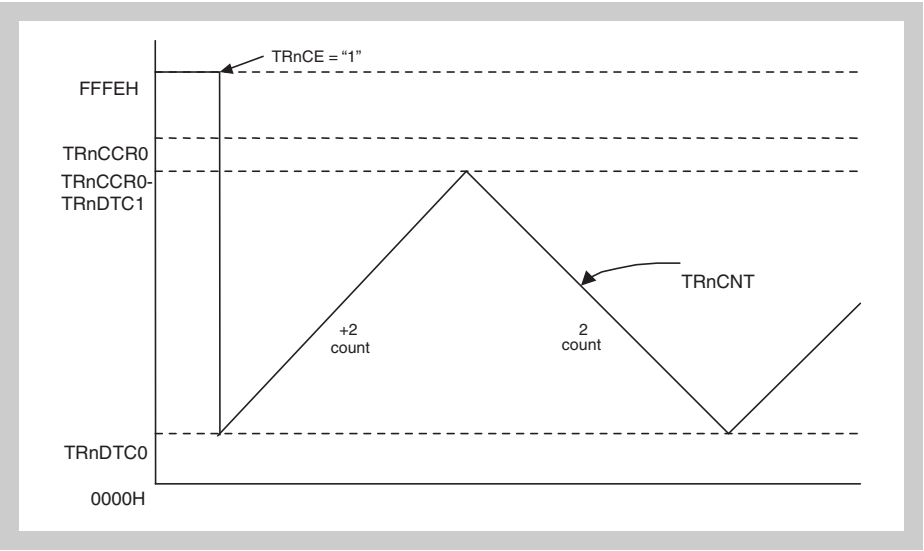


Figure 10-67 Counter operation in high-accuracy T-PWM mode

Remark Minimum counter value: TRnDTC0
Maximum counter value: TRnCCR0 - TRnDTC1
Carrier wave cycle: (TRnCCR0-TRnDTC0-TRnDTC1) × count clock cycle

At initial value $FFFE_H$, the value of TRnDTC0 register is loaded to the sub-counter immediately after TRnCE = 1 is set. Then, until a match with 0000_H, the sub-counter counts down in -2 steps, and the counter value is loaded to the sub-counter at the counter's up count → down count switch timing. The TRnDTC0 register goes on counting up, and upon a match with the TRnCCR0 register, starts counting down in -2 steps. At the same time, upon a match between the counter and the TRnDTC0 register, the counter value is loaded and down count is continued.

The sub-counter operation is shown in *Figure 10-68*.

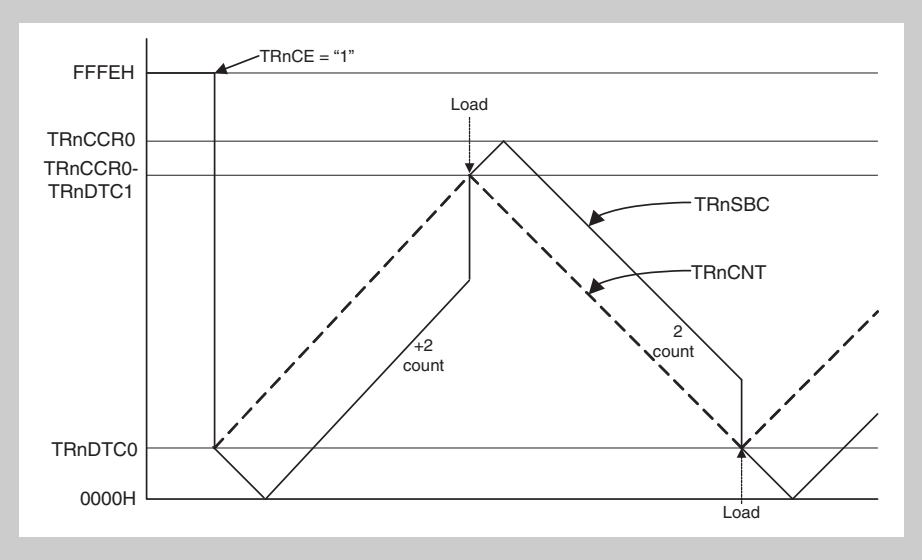


Figure 10-68 Sub-counter operation in high-accuracy T-PWM mode

(5) Basic operation in high-accuracy T-PWM mode

The *Figure 10-69* shows the timing chart when TRnCCR0 = 0010H, TRnDTC0 = 0002_H, TRnDTC1 = 0004_H, and the TRnCCR1 register is set from 0000_H to 0010_H (one part only shown). In this example the outputs are set to active high level (TRnOLm = 0, m = 1 to 6).

If TRnCCR1 > TRnDTC0, pin TORn2 changes with the following compare match.

Since TRnCCR1 = (TRnDTC0-0001_H) is an additional pulse, compared to when TRnCCR1 = (TRnDTC0 - 0002_H), pin TORn2 changes with an 1 count clock delay.

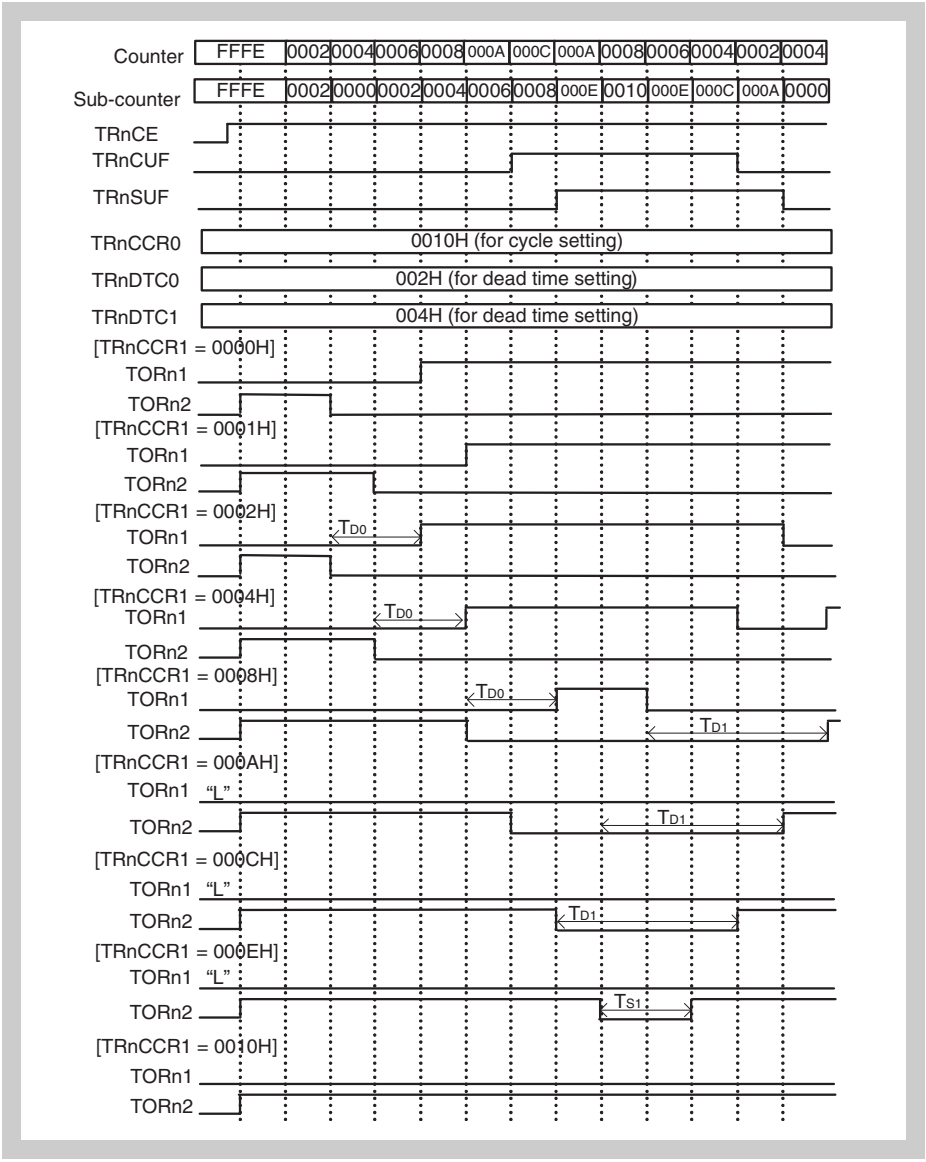


Figure 10-69 Timer output example when TRnCE = 1 is set (initial) (high-accuracy T-PWM mode)

Remarks

1. TRnCCR0 = 0010_H, TRnDTC0 = 0002_H, TRnDTC1 = 0004_H
2. TD0: Time depends on dead time setting of TRnDTC0 register
   TD1: Time depends on dead time setting of TRnDTC1 register
   TS1: Time is determined through sub-counter compare, when sub-counter value > counter value

The *Figure 10-70* shows the timing chart when TRnCCR0 = 0010_H, TRnDTC0 = 0002_H, TRnDTC1 = 0004_H, and the TRnCCR1 register is set from 0000_H to 0010_H (one part only shown). In this example the outputs are set to active high level (TRnOLm = 0, m = 1 to 6).

As can be seen in this figure, a normal phase (pin TORn1) that is active (high level) is output when $0000_H \leq$ TRnCCR1 $\leq$ (TRnCCR0 - TRnDTC0 + 0001_H).

Also, the inverted phase (pin TORn2) that is active (low level) is output when (TRnDTC0 + TRnDTC1) < TRnCCR1 $\leq$ TRnCCR0.

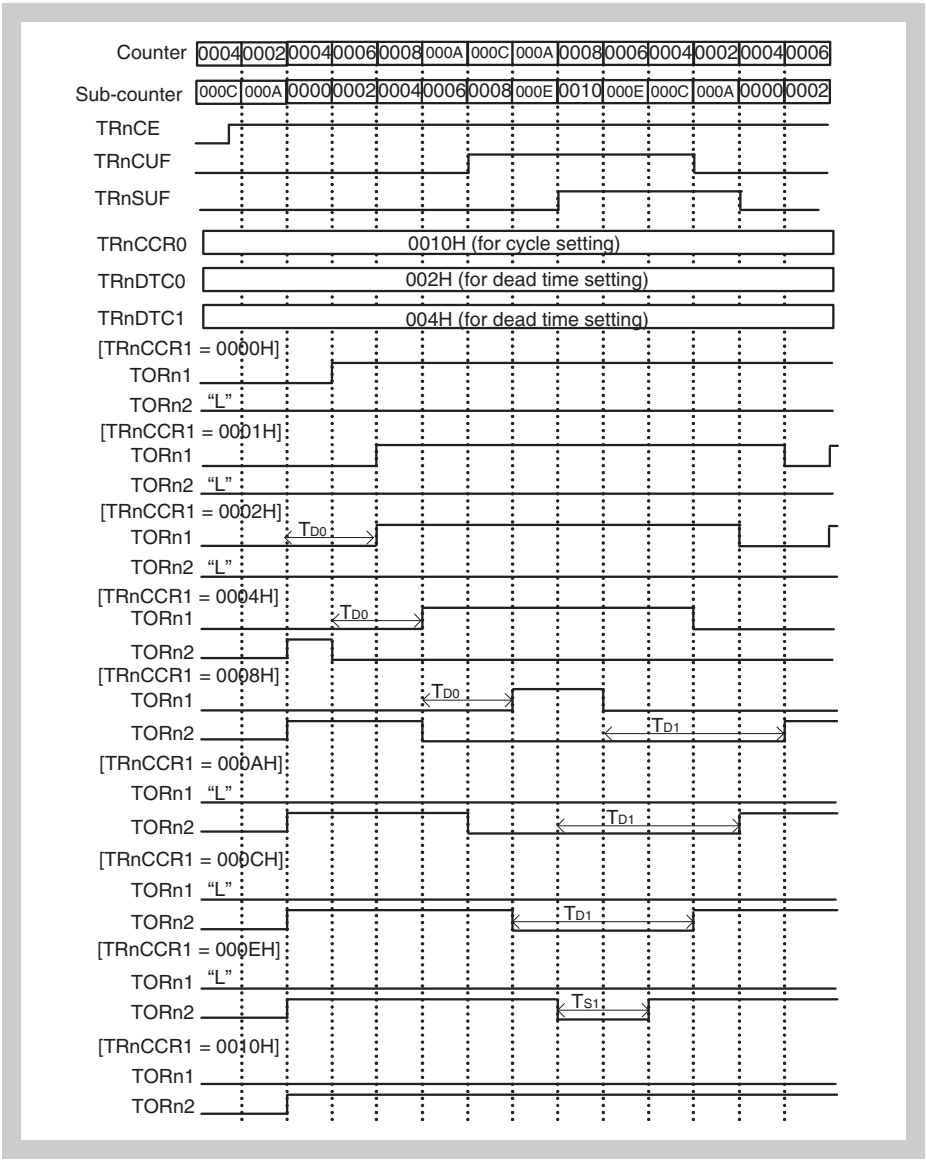


Figure 10-70 Timer output example during operation (high-accuracy T-PWM mode)

Remarks

1. TRnCCR0 = 0010_H, TRnDTC0 = 0002_H, TRnDTC1 = 0004_H
2. TD0: Time depends on dead time setting of TRnDTC0 register
   TD1: Time depends on dead time setting of TRnDTC1 register
   TS0: Time is determined through sub-counter compare, when sub-counter value < counter value
   TS1: Time is determined through sub-counter compare, when sub-counter value > counter value

(6) Additional pulse control in high-accuracy T-PWM mode

In the high-accuracy T-PWM mode, additional pulse can be set by setting the LSB of the duty setting registers (TRnCCR1 to TRnCCR3) to “1”. With the additional pulse control function, finer duty control can be performed (higher accuracy).

TORn1 pin output examples are provided below for when additional pulse control is and is not performed. The settings used here are TRnCCR = 12, TRnDTC0, and TRnDTC1 = 0.

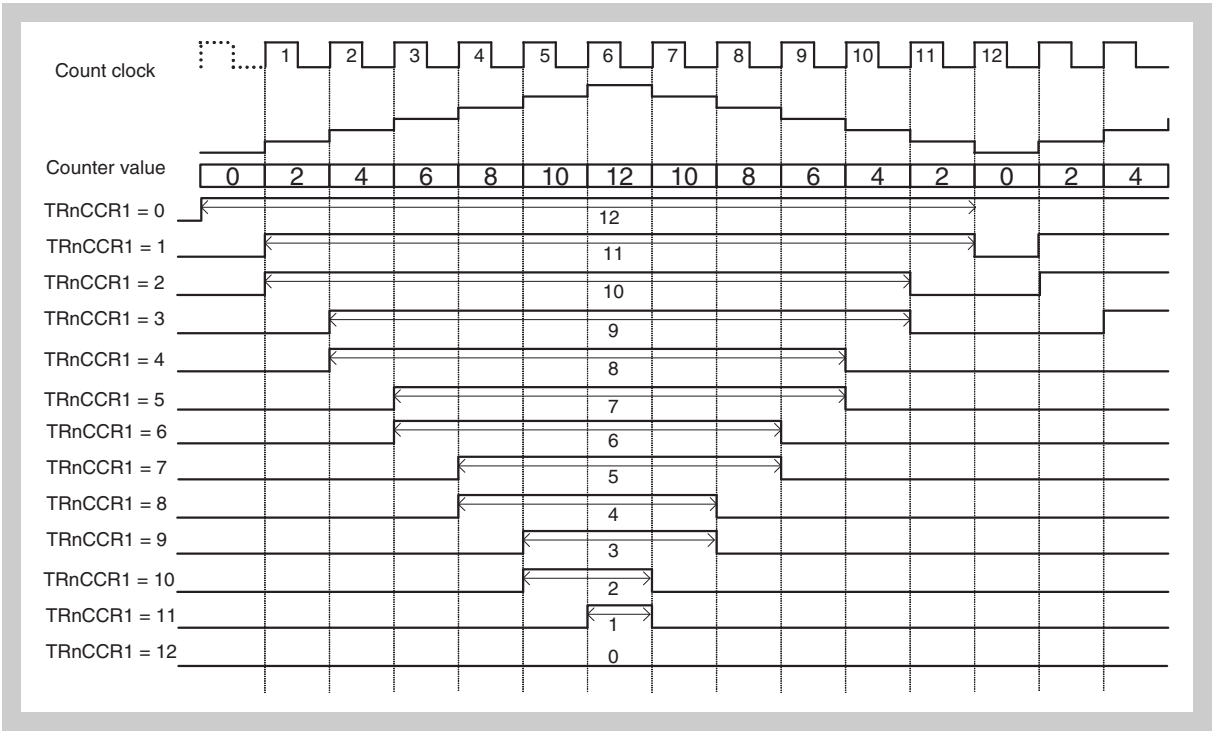


Figure 10-71 TORn1 pin output example when additional pulse control is performed

Remark TRnCCR0 = 12, TRnDTC0 = 0, TRnDTC1 = 0

The locations where additional pulse control is performed are when an odd value has been set to the TRnCCR1 register.

In the above figure, the arrows and numbers indicate the duty width of the TORn1 pin output within 1 cycle.

As can be seen in the above figure, when additional pulse control is performed, the output width (duty ratio) of pin TORn1 can be controlled in 1 count clock steps from 12 clocks to 0 clocks.

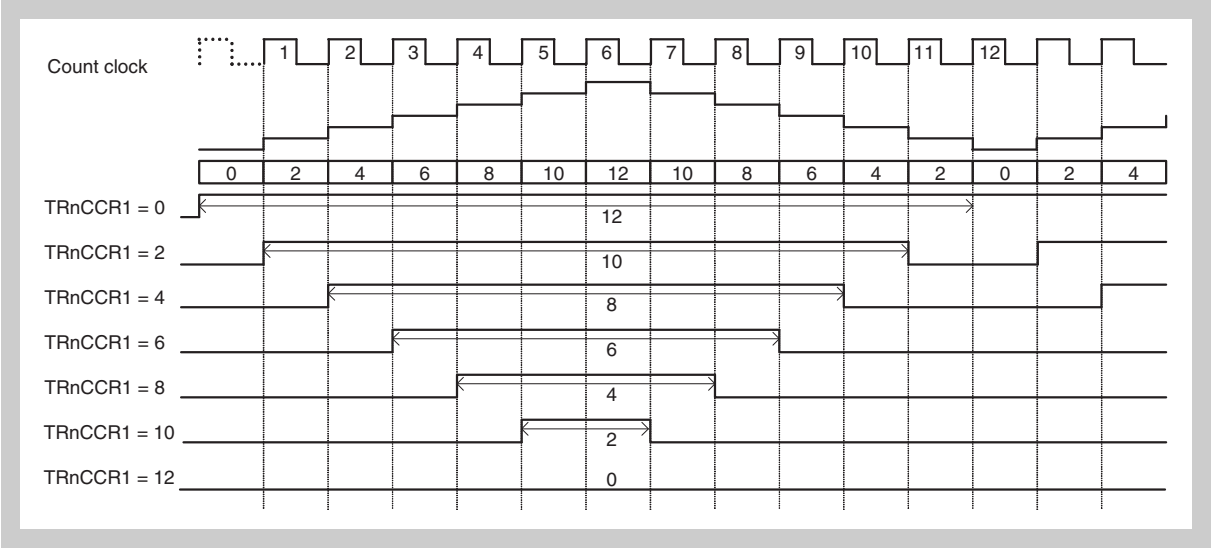


Figure 10-72 TORn1 pin output example when additional pulse control is not performed

Remark TRnCCR0 = 12, TRnDTC0 = 0, TRnDTC1 = 0

The figure above is an example when additional pulse control is not performed.

In the above figure, the arrows and numbers indicate the duty width of the TORn1 pin output within 1 cycle.

When additional pulse control is not performed, the output width of pin TORn1 can be controlled in 2 count clock steps from 12 clocks to 0 clocks. In this case, the duty change amount is larger compared to when additional pulse control is performed.

(7) Caution on timer output in high-accuracy T-PWM mode

There are cautions for TRnCCR1 to TRnCCR3 as follows when varying 6-phase PWM duty by using reload (batch rewrite).

(a) In case of TRnCCR0 + 2 ≤ TRnCCRm (Setting prohibited)

Figure 10-73 shows the case when the value of "TRnCCR0 + 2 or more" is set to the TRnCCR1 register. When the TRnCCR1 register setting is changed like this, a match between the 16-bit counter and TRnCCR1 register does not occur thereafter. Therefore, the TORn1 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.

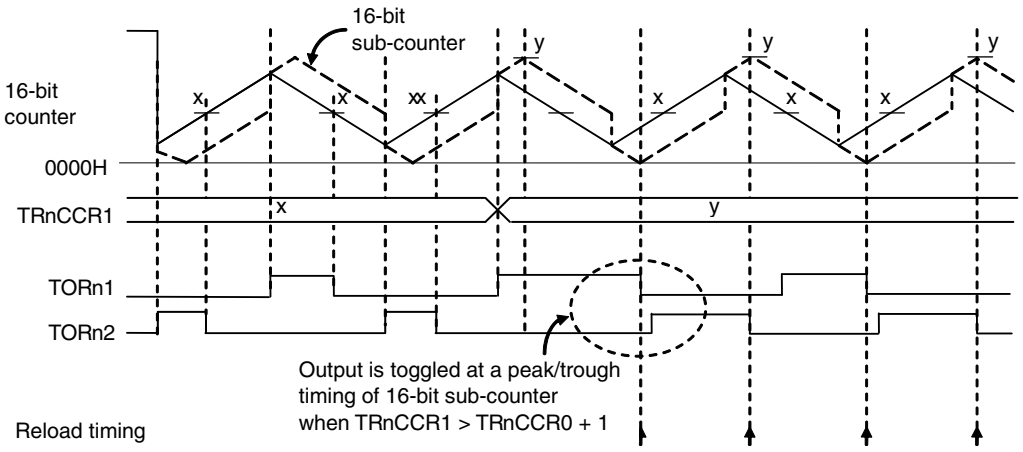


Figure 10-73 Timing of timer output in high-accuracy T-PWM mode when TRnCCR0 + 2 ≤ TRnCCR1

Remark m = 1 to 3

(b) In case of rewriting from "TRnCCRm = 0000_H" to "TRnCCRm = TRnCCR0"

Figure 10-74 shows the output waveform where the TRnCCR1 register setting is changed from 100% output to 0% output. The TORn1 pin output is inverted upon a match between the TRnCCR1 register and 16-bit sub-counter, and the TORn2 pin output is inverted after the dead time count.

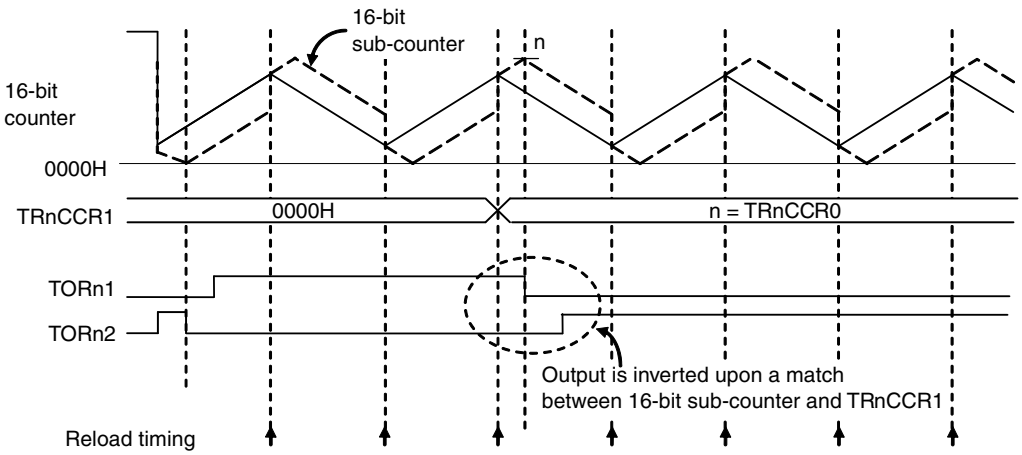


Figure 10-74 Timing of timer output in high-accuracy T-PWM mode when rewriting from "TRnCCR1 = 0000_H" to TRnCCR1 = TRnCCR0

Remark m = 1 to 3

(c) In case of rewriting from "(TRnDTC0 + TRnDTC1) < TRnCCRm < (TRnCCR0 − TRnDTC0 − TRnDTC1)" to "TRnCCRm < (TRnDTC0 + TRnDTC1)"

Figure 10-75 shows the output waveform when rewriting the TRnCCR1 register from x (TRnDTC0 + TRnDTC1 < x < TRnCCR0 − TRnDTC0 − TRnDTC1) to y (y < TRnDTC0 + TRnDTC1). In this case, the TORn1 pin output becomes active when the TORn1 pin set condition occurs upon a match between the 16-bit counter (or 16-bit sub-counter) and the TRnCCR1 register immediately after reload (batch rewrite).

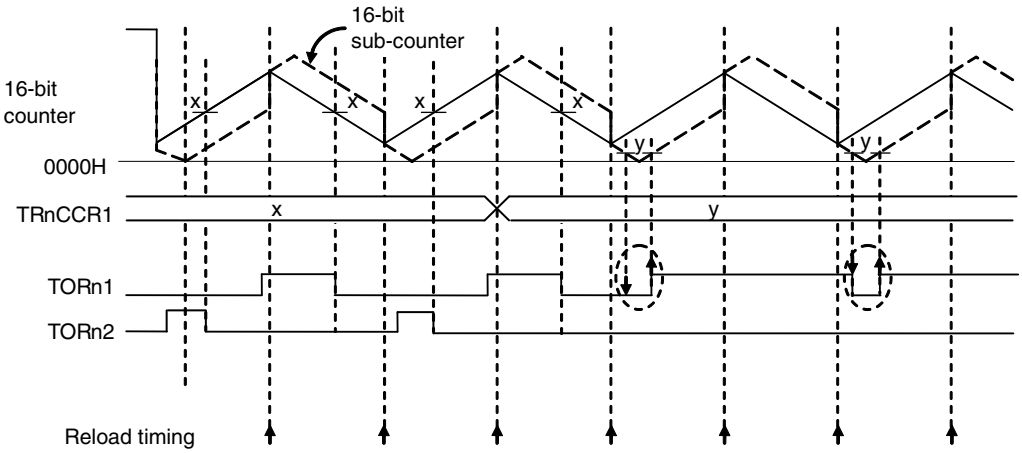


Figure 10-75 Timing of timer output in high-accuracy T-PWM mode when rewriting from "(TRnDTC0 + TRnDTC1) < TRnCCR1 < (TRnCCR0 - TRnDTC0 - TRnDTC1)" to "TRnCCR1 < (TRnDTC0 + TRnDTC1)"

Remark m = 1 to 3

(d) In case of rewriting from "(TRnDTC0 + TRnDTC1) < TRnCCRm < (TRnCCR0 – TRnDTC0 – TRnDTC1)" to "(TRnCCR0 – TRnDTC1 + 1) < TRnCCRm < TRnCCR0"

Figure 10-76 shows the output waveform when rewriting the TRnCCR1 register from x (TRnDTC0 + TRnDTC1 < x < TRnCCR0 – TRnDTC0 – TRnDTC1) to y (TRnCCR0 – TRnDTC0 – TRnDTC1 < TRnDTC0 < TRnCCR0). In this case, the TORn2 pin output becomes inactive (high level) when the TORn2 pin set condition occurs upon a match between the 16-bit counter (or 16-bit sub-counter) and TRnCCRm register immediately after batch rewrite.

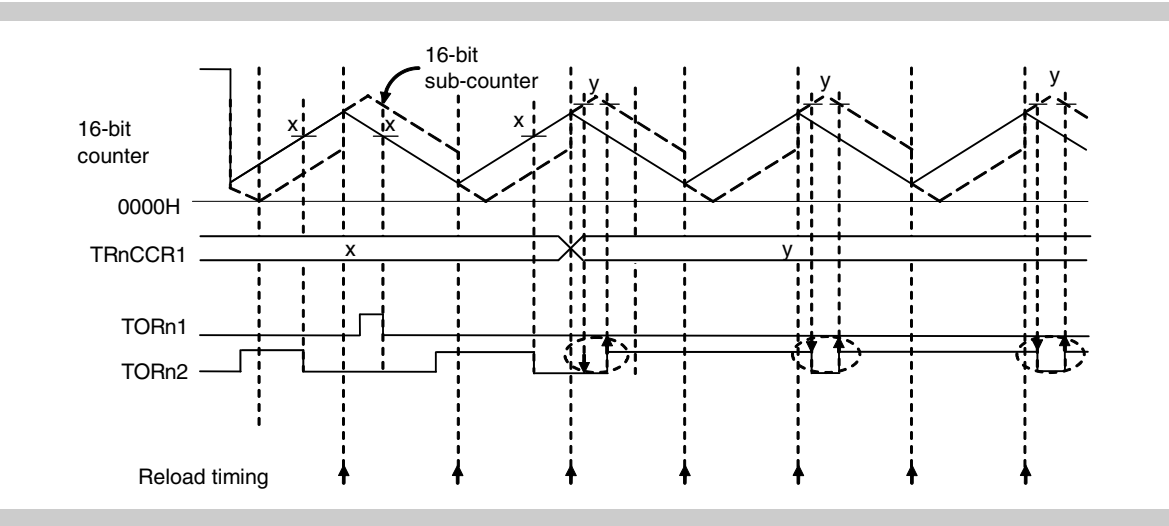


Figure 10-76 Timing of timer output in high-accuracy T-PWM mode when rewriting from "(TRnDTC0 + TRnDTC1) < TRnCCR1 < (TRnCCR0 - TRnDTC0 - TRnDTC)1" to "(TRnCCR0 - TRnDTC1 + 1) < TRnCCR1 < TRnCCR0"

Remark m = 1 to 3

(8) 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 10-29* and *Table 10-30*.

Table 10-29 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 (< TRnDTC0)
Clear	RT1	Match between counting down near the 16-bit sub-counter trough and compare register values (< TRnDTC0)
Set	ST2	At completion of dead time counter (TRnDTC0) 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	TRnCCR0 and TRnDTC0 settings are changed at a reload timing. Though neither a match (nor a match interrupt) occurs between TRnCCR0 and TRnDTC0, 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 10-30 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 (> TRnCCR0 - TRnDTC1)
Clear	RB1	Match between counting up near the 16-bit sub-counter peak and compare register values (> TRnCCR0 - TRnDTC1)
Set	SB2	At completion of dead time counter (TRnDTC1) 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 TRnCCR0
Clear	RB4	TRnCCR0 and TRnDTC0 settings are changed at a reload timing. Though neither a match (nor a match interrupt) occurs between TRnCCR0 and TRnDTC1, 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.

Table 10-31 Timer output change after compare register updating by trough reload when origin compare value is 0000_H and (TRnDTC0 < TRnDTC1)

Compare register value		Reference figure
immediately before trough reload	after trough reload (TRnDTC0 < TRnDTC1)	
0000H	0000H < TRnCCR1 to TRnCCR3 < TRnDTC0	*Figure 10-77*
	TRnCCR1 to TRnCCR3 = 0000H, TRnDTC0 + 1	*Figure 10-78*
	TRnDTC0 + 1 < TRnCCR1 to TRnCCR3 ≤ TRnDTC0 × 2	*Figure 10-79*
	TRnDTC0 × 2 < TRnCCR1 to TRnCCR3 < TRnCCR0 − TRnDTC0 − TRnDTC1	*Figure 10-80*
	TRnCCR0 − TRnDTC0 − TRnDTC1 ≤ TRnCCR1 to TRnCCR3 < TRnCCR0 − TRnDTC1	*Figure 10-81*
	TRnCCR0 − TRnDTC1 ≤ TRnCCR1 to TRnCCR3 < TRnCCR0	*Figure 10-82*
	TRnCCR1 to TRnCCR3 = TRnCCR0	*Figure 10-83*

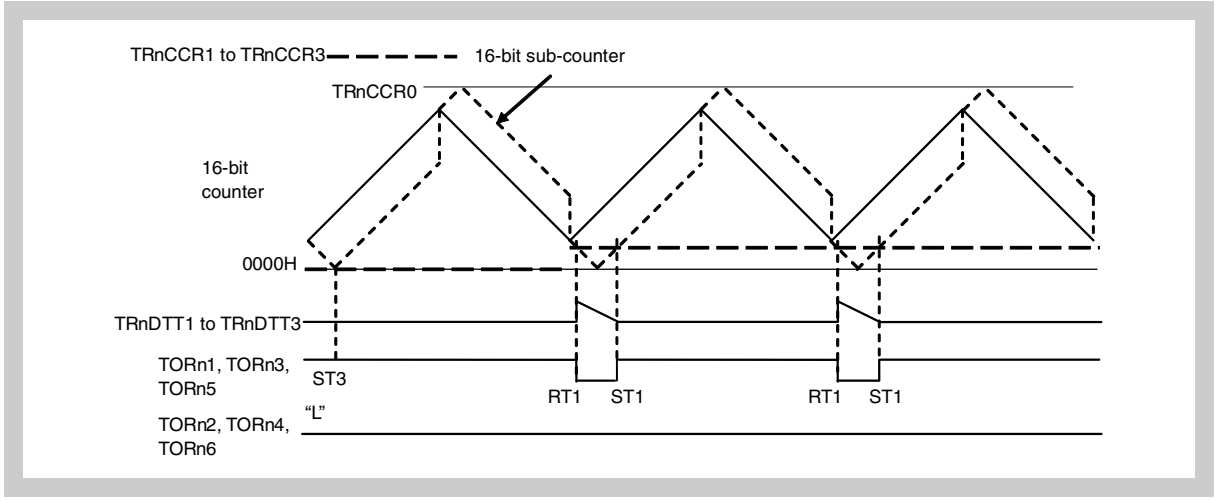


Figure 10-77 Timer output change after compare register updating by trough reload when "TRnCCR1 to TRnCCR3 = 0000_H"→ "0000_H < TRnCCR1 to TRnCCR3 < TRnDTC0"

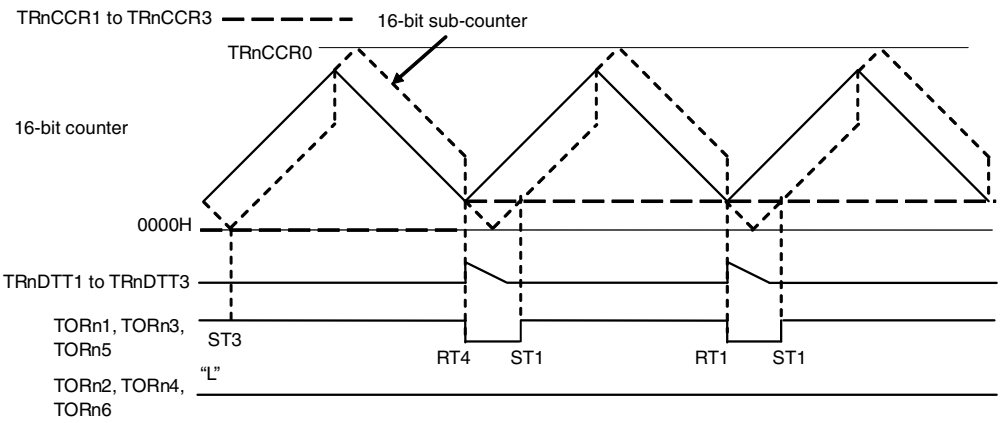


Figure 10-78 Timer output change after compare register updating by trough reload when "TRnCCR1 to TRnCCR3 = 0000_H" → "TRnCCR1 to TRnCCR3 = TRnDTC0, TRnDTC0 + 1"

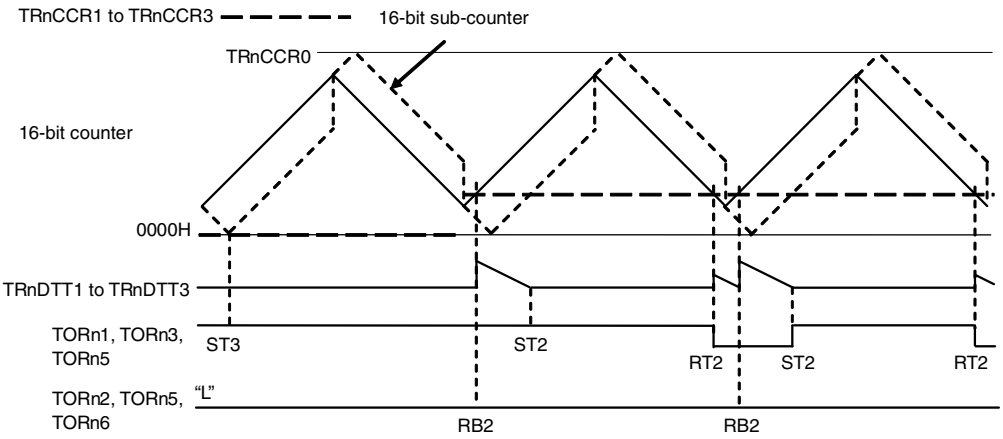


Figure 10-79 Timer output change after compare register updating by trough reload when "TRnCCR1 to TRnCCR3 = 0000_H" → "TRnDTC0 < TRnCCR1 to TRnCCR3 < TRnDTC0 × 2"

When the values of TRnCCR1 to TRnCCR3 are changed from "0000H ≤ TRnCCR1 to TRnCCR3 < TRnDTC0" to "TRnDTC0 < TRnCCR1 to TRnCCR3 < TRnDTC0 × 2", the positive phase will be 100% output for one cycle, as shown in *Figure 10-79*.
To prevent this phenomenon, change "0000H ≤ TRnCCR1 to TRnCCR3 < TRnDTC0" to "TRnDTC0 < TRnCCR1 to TRnCCR3 < TRnDTC × 2" through TRnDTC0, or directly change "0000H ≤ TRnCCR1 to TRnCCR3 < TRnDTC0" to "TRnDTC0 × 2 ≤ TRnCCR1 to TRnCCR3".

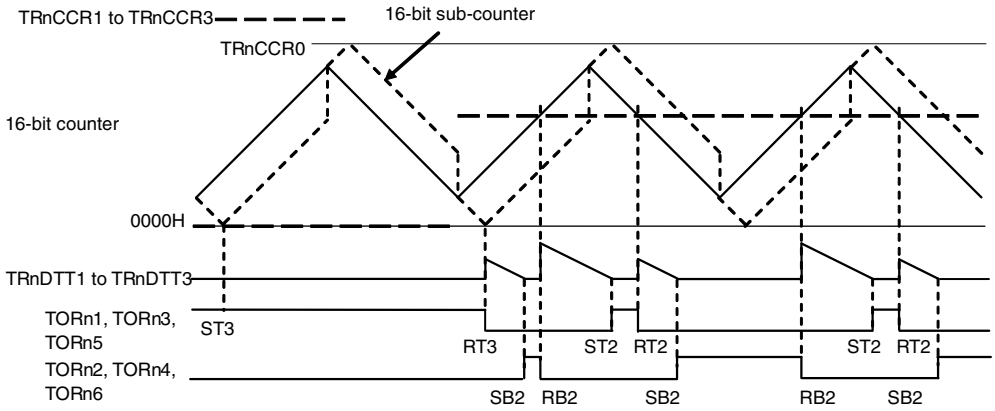


Figure 10-80 Timer output change after compare register updating by trough reload when "TRnCCR1 to TRnCCR3 = 0000_H" → "TRnDTC0 × 2 < TRnCCR1 to TRnCCR3 < (TRnCCR0 − TRnDTC1 − TRnDTC0)"

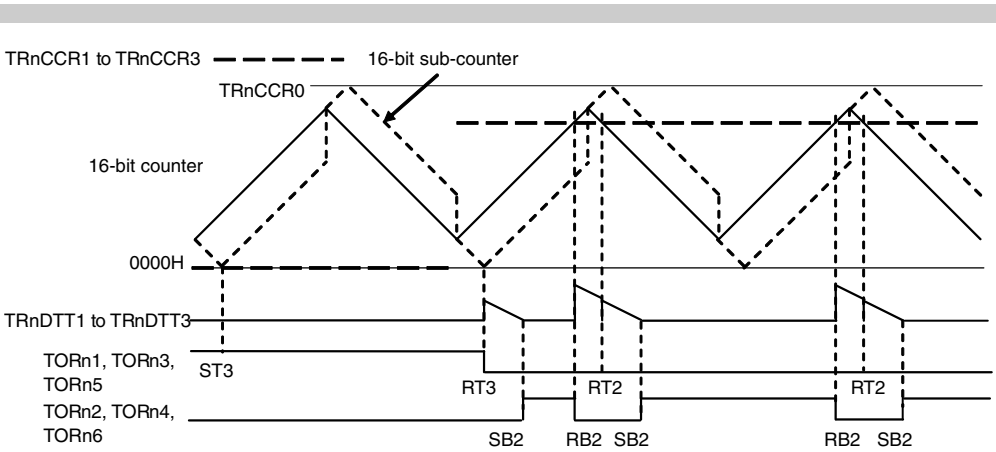


Figure 10-81 Timer output change after compare register updating by trough reload when "TRnCCR1 to TRnCCR3 = 0000_H" → "(TRnCCR0 − TRnDTC1 − TRnDTC0) < TRnCCR1 to TRnCCR3 < (TRnCCR0 − TRnDTC1)"

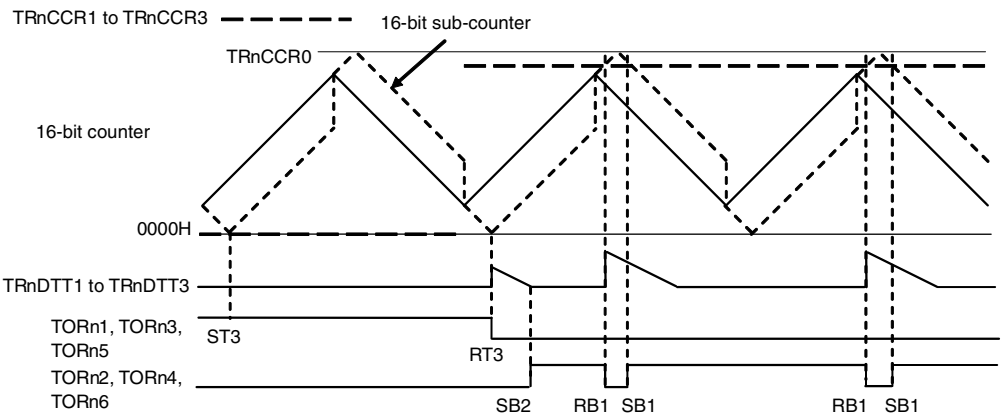


Figure 10-82 Timer output change after compare register updating by trough reload when "TRnCCR1 to TRnCCR3 = 0000_H" → "(TRnCCR0 − TRnDTC1) < TRnCCR1 to TRnCCR3 < TRnCCR0"

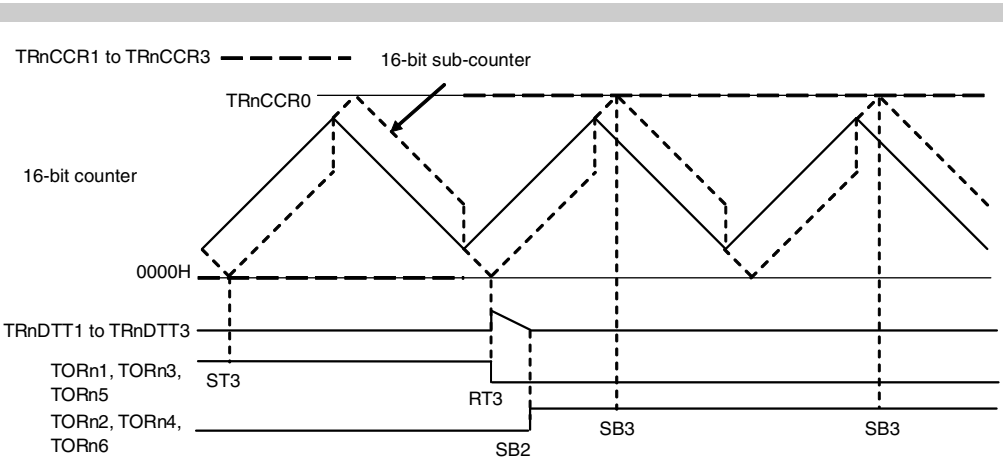


Figure 10-83 Timer output change after compare register updating by trough reload when "TRnCCR1 to TRnCCR3 = 0000_H" → "(TRnCCR0 − TRnDTC1) < TRnCCR1 to TRnCCR3 < TRnCCR0"

Table 10-32 Timer output change after compare register updating by trough reload when origin compare value = TRnCCR0

Compare register value		Reference figure
immediately before trough reload	after trough reload	
TRnCCR0	TRnCCR1 to TRnCCR3 = 0000H	*Figure 10-84*
	0000H < TRnCCR1 to TRnCCR3 < TRnDTC0	*Figure 10-85*
	TRnCCR1 to TRnCCR3 = TRnDTC0, TRnDTC0 + 1	*Figure 10-86*
	TRnDTC0 + 1 < TRnCCR1 to TRnCCR3 < TRnDTC0 + TRnDTC1	*Figure 10-87*
	TRnDTC0 + TRnDTC1 < TRnCCR1 to TRnCCR3 < TRnCCR0 – TRnDTC0 – TRnDTC1	*Figure 10-88*
	TRnCCR0 – TRnDTC0 – TRnDTC1 ≤ TRnCCR1 to TRnCCR3 < TRnCCR0 – TRnDTC1	*Figure 10-89*
	TRnCCR0 – TRnDTC1 ≤ TRnCCR1 to TRnCCR3 < TRnCCR0	*Figure 10-90*

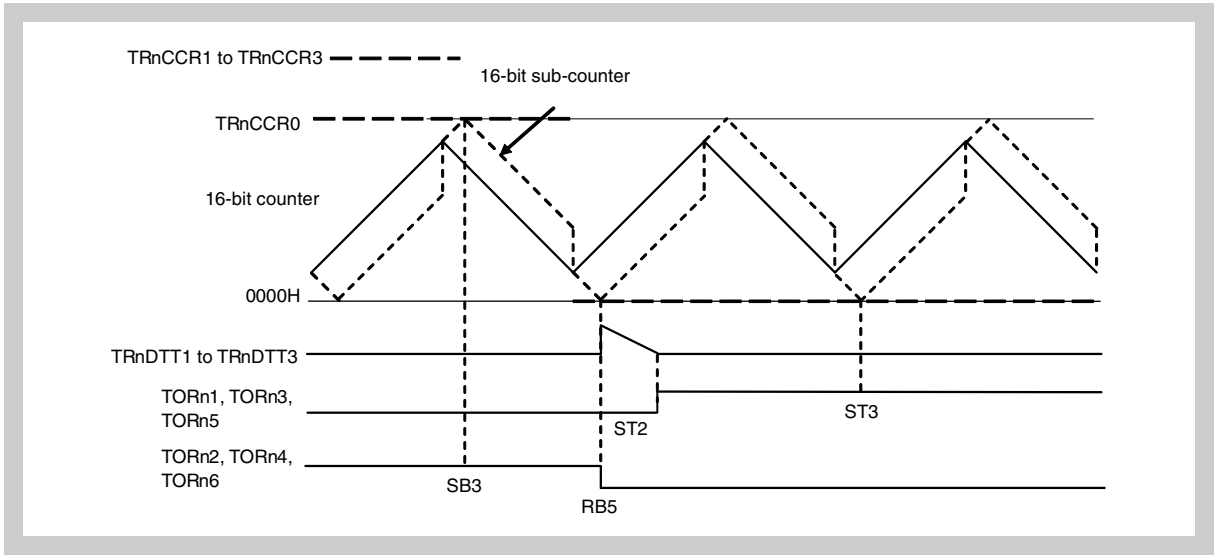


Figure 10-84 Timer output change after compare register updating by trough reload when "TRnCCR1 to TRnCCR3 = TRnCCR0" → "TRnCCR1 to TRnCCR3 = 0000_H"

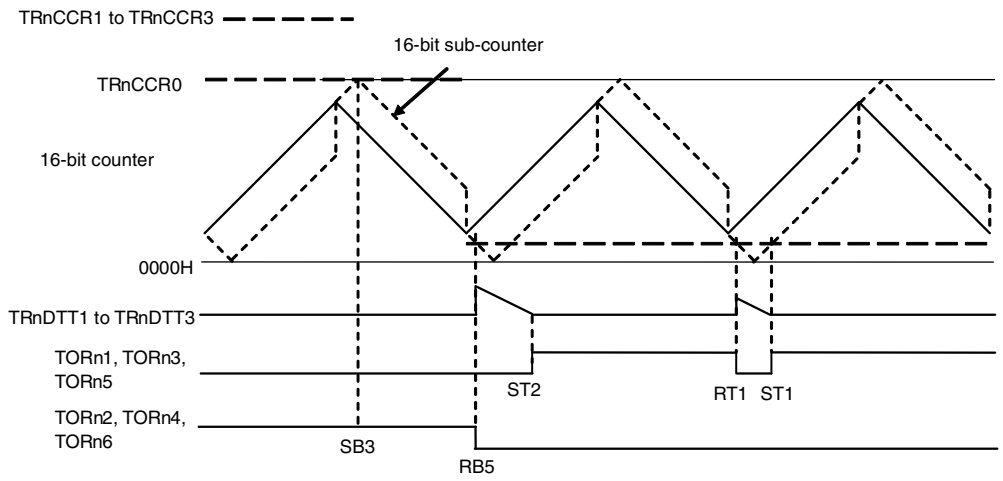


Figure 10-85 Timer output change after compare register updating by trough reload when "TRnCCR1 to TRnCCR3 = TRnCCR0" → "0000_H < TRnCCR1 to TRnCCR3 < TRnDTC0"

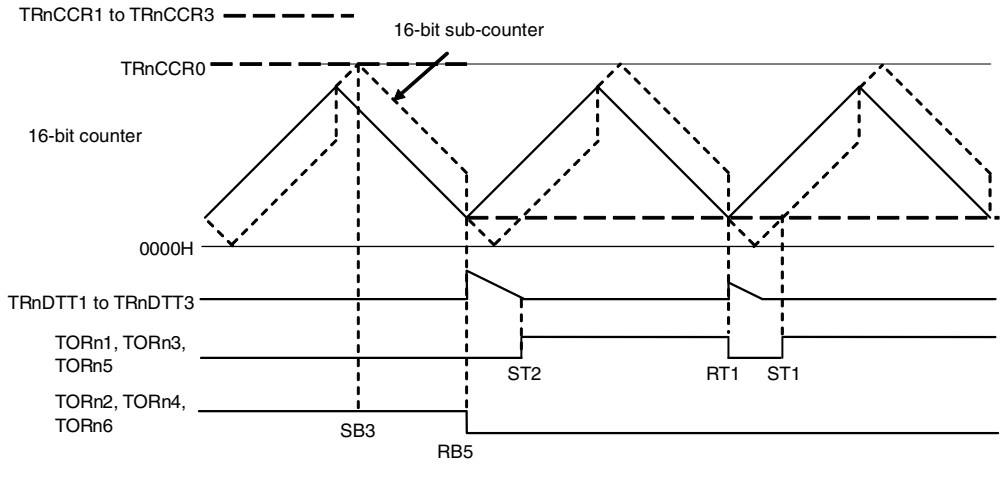


Figure 10-86 Timer output change after compare register updating by trough reload when "TRnCCR1 to TRnCCR3 = TRnCCR0" → "TRnCCR1 to TRnCCR3 = TRnDTC0, TRnDTC0 + 1"

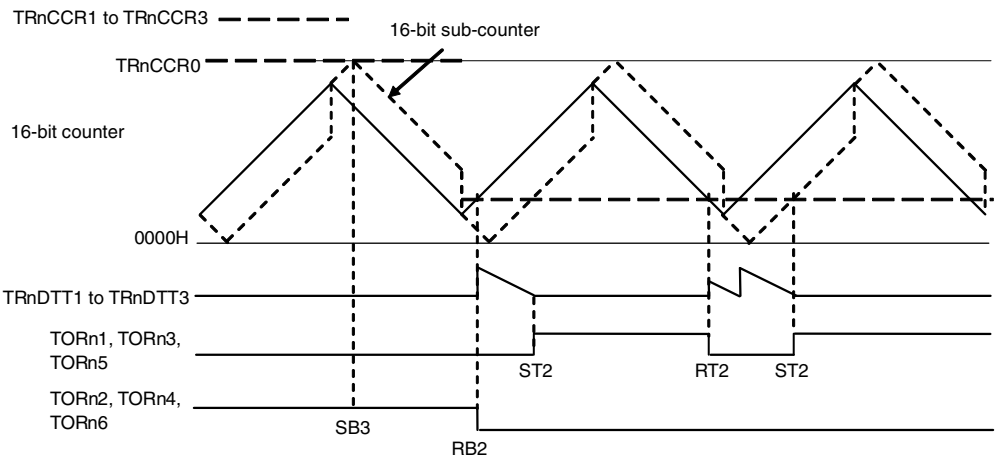


Figure 10-87 Timer output change after compare register updating by trough reload when "TRnCCR1 to TRnCCR3 = TRnCCR0" → "(TRnDTC0 + 1) < TRnCCR1 to TRnCCR3 ≤ (TRnDTC0 + TRnDTC1)"

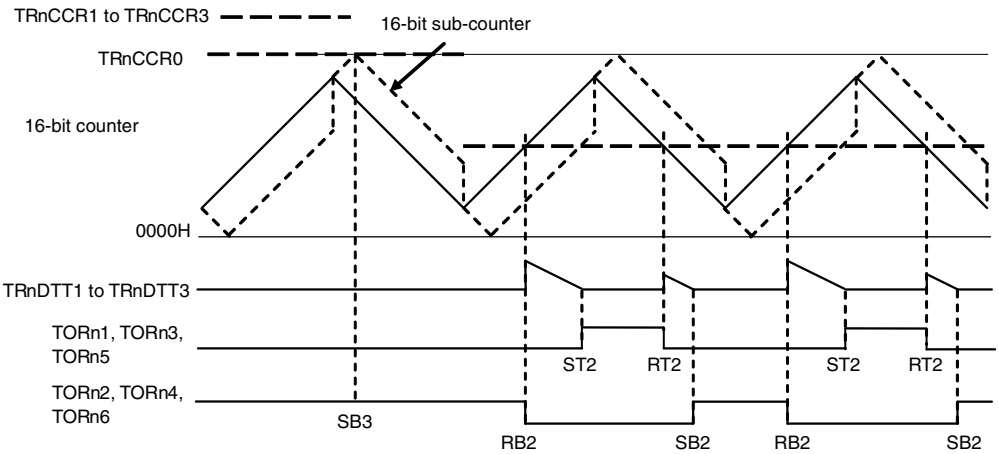


Figure 10-88 Timer output change after compare register updating by trough reload when "TRnCCR1 to TRnCCR3 = TRnCCR0" → "(TRnDTC0 + TRnDTC1) < TRnCCR1 to TRnCCR3 < (TRnCCR0 − TRnDTC1 − TRnDTC0)"

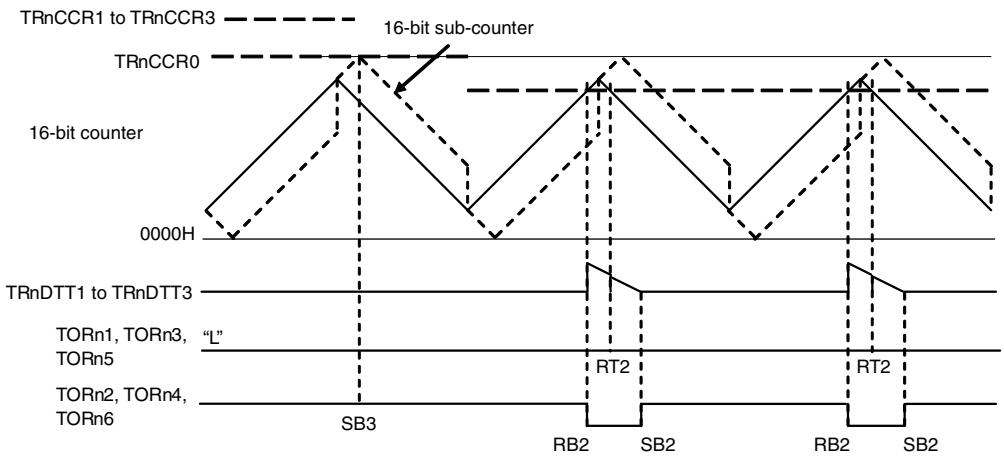


Figure 10-89 Timer output change after compare register updating by trough reload when "TRnCCR1 to TRnCCR3 = TRnCCR0" → "(TRnCCR0 − TRnDTC1 − TRnDTC0) < TRnCCR1 to TRnCCR3 < (TRnCCR0 − TRnDTC1)"

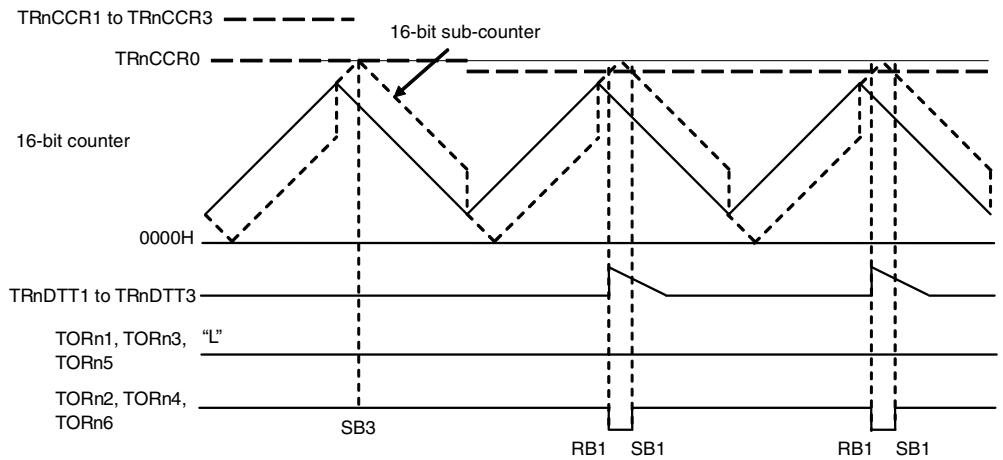


Figure 10-90 Timer output change after compare register updating by trough reload when "TRnCCR1 to TRnCCR3 = TRnCCR0" → "(TRnDTC0 − TRnDTC1) < TRnCCR1 to TRnCCR3 < TRnCCR0"

Table 10-33 Timer output change after compare register updating by peak reload when origin compare value = TRnCCR0 and (TRnDTC1 < TRnDTC0)

Compare register value		Reference figure
immediately before peak Reload	**after peak reload (TRnDTC1 < TRnDTC0)**	
TRnCCR0	(TRnCCR0 − TRnDTC1) ≤ TRnCCR1 to TRnCCR3 < TRnCCR0	*Figure 10-91*
	TRnCCR1 to TRnCCR3 = TRnCCR0 − TRnDTC1	*Figure 10-92*
	TRnCCR0 − TRnDTC1 × 2 ≤ TRnCCR1 to TRnCCR3 < TRnCCR0 − TRnDTC1	*Figure 10-93*
	TRnDTC0 + TRnDTC1 < TRnCCR1 to TRnCCR3 < TRnCCR0 − TRnDTC1 × 2	*Figure 10-94*
	TRnDTC0 + 1 < TRnCCR1 to TRnCCR3 < TRnDTC0 + TRnDTC1	*Figure 10-95*
	0000H < TRnCCR1 to TRnCCR3 ≤ TRnDTC0 + TRnDTC1	*Figure 10-96*
	TRnCCR1 to TRnCCR3 = 0000H	*Figure 10-97*

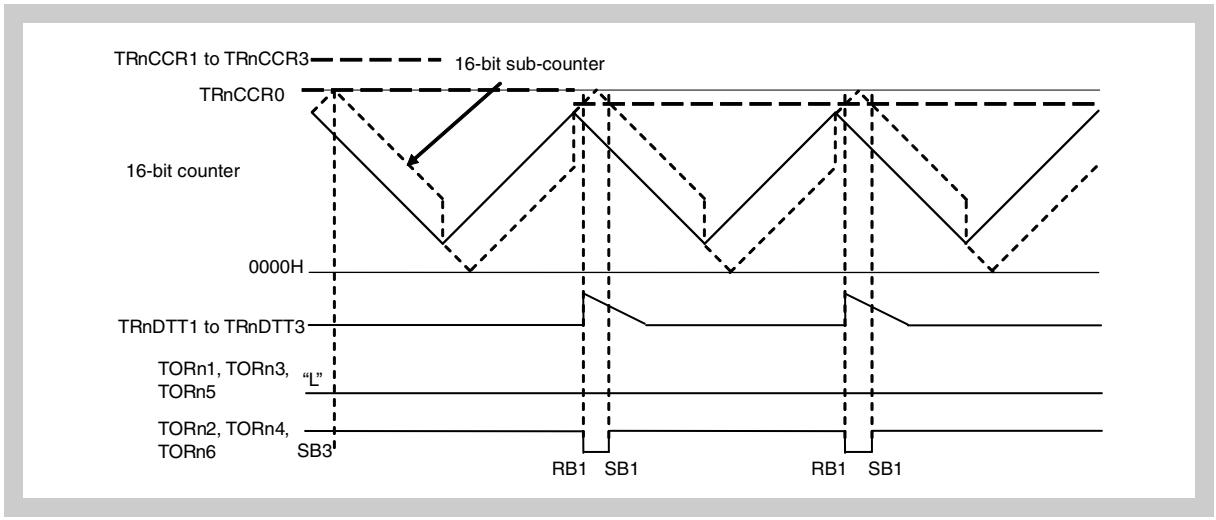


Figure 10-91 Timer output change after compare register updating by peak reload when "TRnCCR1 to TRnCCR3 = TRnCCR0" → "(TRnCCR0 − TRnDTC1) < TRnCCR1 to TRnCCR3 < TRnCCR0"

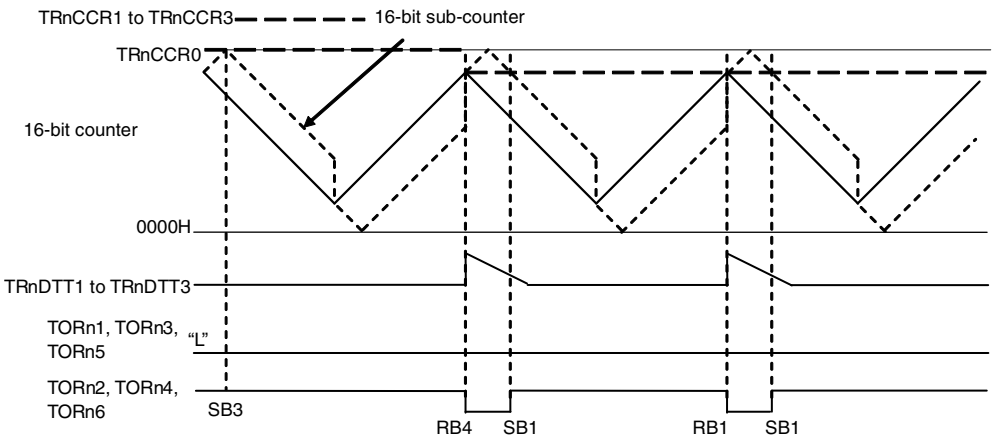


Figure 10-92 Timer output change after compare register updating by peak reload when "TRnCCR1 to TRnCCR3 = TRnCCR0" → "TRnCCR1 to TRnCCR3 = (TRnDTC0 − TRnDTC1)"

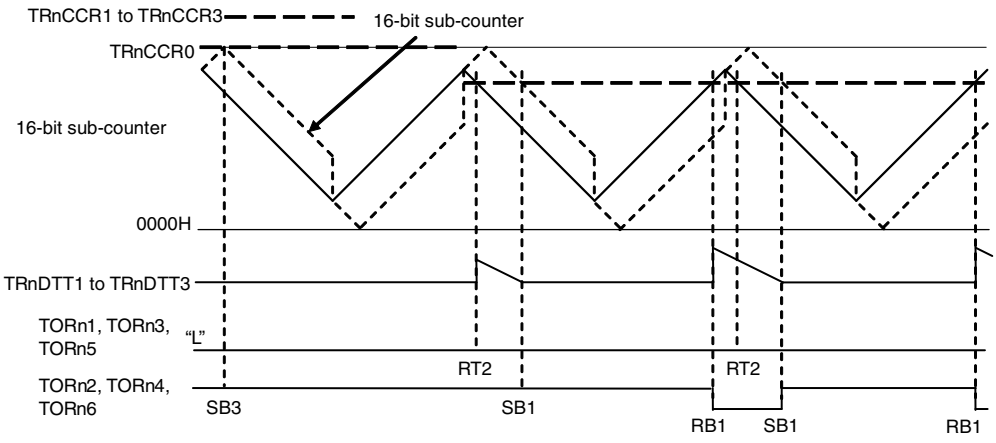


Figure 10-93 Timer output change after compare register updating by peak reload when "TRnCCR1 to TRnCCR3 = TRnCCR0" → "(TRnCCR0 − TRnDTC1 × 2) < TRnCCR1 to TRnCCR3 < (TRnCCR0 − TRnDTC1)"

When the values of TRnCCR1 to TRnCCR3 are changed from "TRnCCR0 − TRnDTC1 < TRnCCR1 to TRnCCR3 ≤ TRnCCR0" to "TRnCCR0 − TRnDTC1 × 2 < TRnCCR1 to TRnCCR3 < TRnCCR0 − TRnDTC1", the negative phase will be 100% output for one cycle, as shown in *Figure 10-93*.
To prevent this phenomenon, change "TRnCCR0 − TRnDTC1 < TRnCCR1 to TRnCCR3 ≤ TRnCCR0" to "TRnDTC0 < TRnCCR1 to TRnCCR3 < TRnDT1 × 2" through "TRnCCR0 − TRnDTC1", or directly change "TRnCCR0 − TRnDTC1 < TRnCCR1 to TRnCCR3 < TRnCCR0" to "TRnCCR1 to TRnCCR3 ≤ TRnCCR0 − TRnDTC1 × 2".

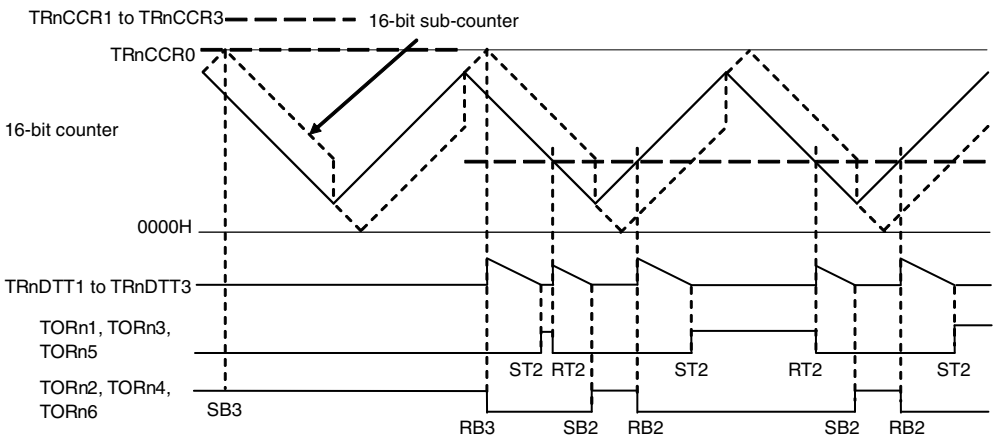


Figure 10-94 Timer output change after compare register updating by peak reload when "TRnCCR1 to TRnCCR3 = TRnCCR0" → "(TRnDTC0 + TRnDTC1) < TRnCCR1 to TRnCCR3 < (TRnCCR0 − TRnDTC1 × 2)"

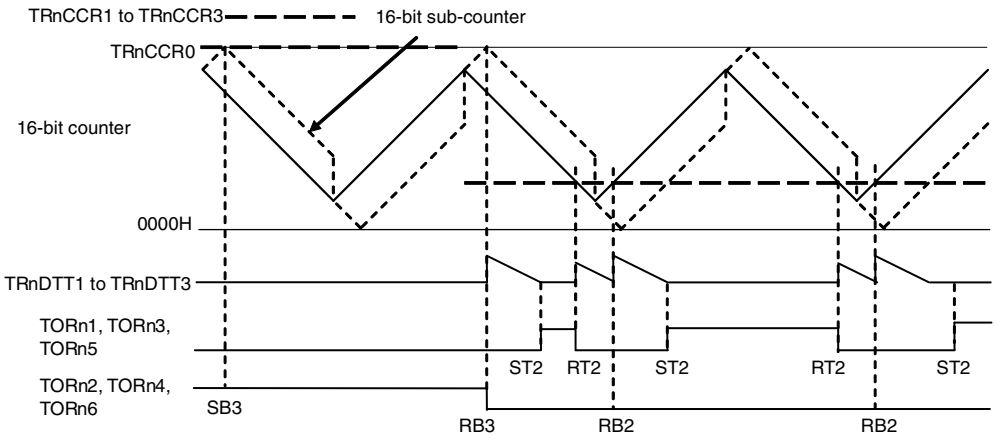


Figure 10-95 Timer output change after compare register updating by peak reload when "TRnCCR1 to TRnCCR3 = TRnCCR0" → "(TRnDTC0 + 1) < TRnCCR1 to TRnCCR3 ≤ (TRnDTC0 + TRnDTC1)

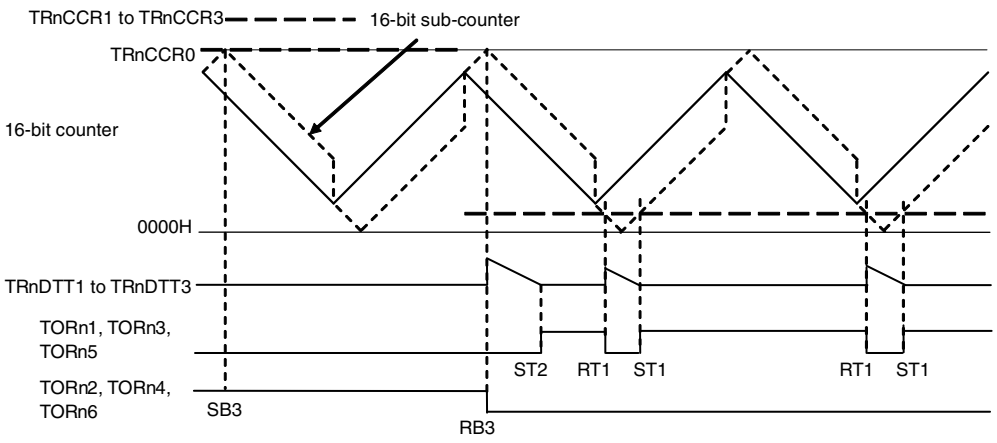


Figure 10-96 Timer output change after compare register updating by peak reload when "TRnCCR1 to TRnCCR3 = TRnCCR0" → "0000_H < TRnCCR1 to TRnCCR3 ≤ (TRnDTC0 + 1)"

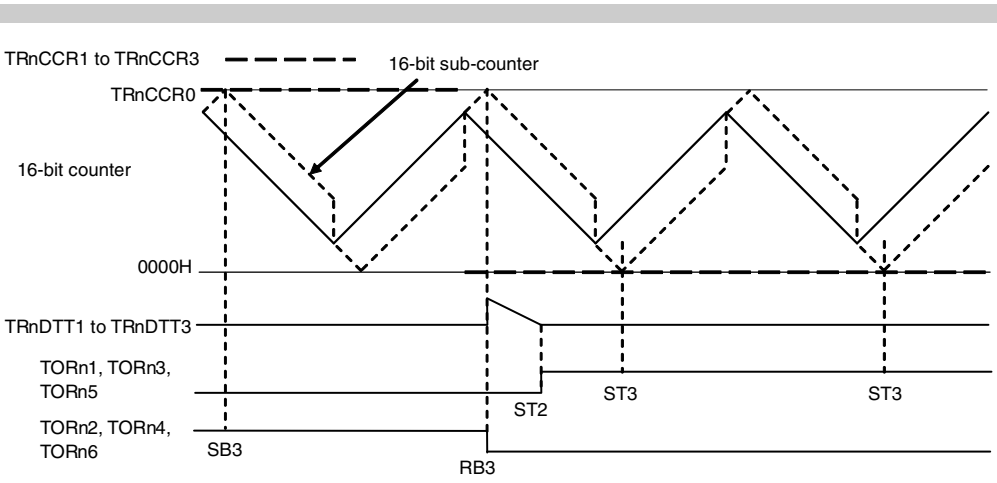


Figure 10-97 Timer output change after compare register updating by peak reload when "TRnCCR1 to TRnCCR3 = TRnCCR0" → "TRnCCR1 to TRnCCR3 = 0000_H"

Table 10-34 Timer output change after compare register updating by peak reload when origin compare value = 0000_H

Compare Register Value Immediately Before Peak Reload	Compare Register Value After Trough Reload	Figure No.
0000H	TRnCCR1 to TRnCCR3 = TRnCCR0	*Figure 10-98*
	TRnCCR0 – TRnDTC1 < TRnCCR1 to TRnCCR3 < TRnCCR0	*Figure 10-99*
	TRnCCR1 to TRnCCR3 = TRnCCR0 – TRnDTC1	*Figure 10-100*
	TRnCCR0 – TRnDTC0 – TRnDTC1 ≤ TRnCCR1 to TRnCCR3 < TRnCCR0 – TRnDTC1	*Figure 10-101*
	TRnDTC0 + TRnDTC1 < TRnCCR1 to TRnCCR3 < TRnCCR0 – TRnDTC0 – TRnDTC1	*Figure 10-102*
	TRnDTC0 + 1 < TRnCCR1 to TRnCCR3 ≤ TRnDTC0 + TRnDTC1	*Figure 10-103*
	0000H < TRnCCR1 to TRnCCR3 ≤ TRnDTC0 + 1	*Figure 10-104*

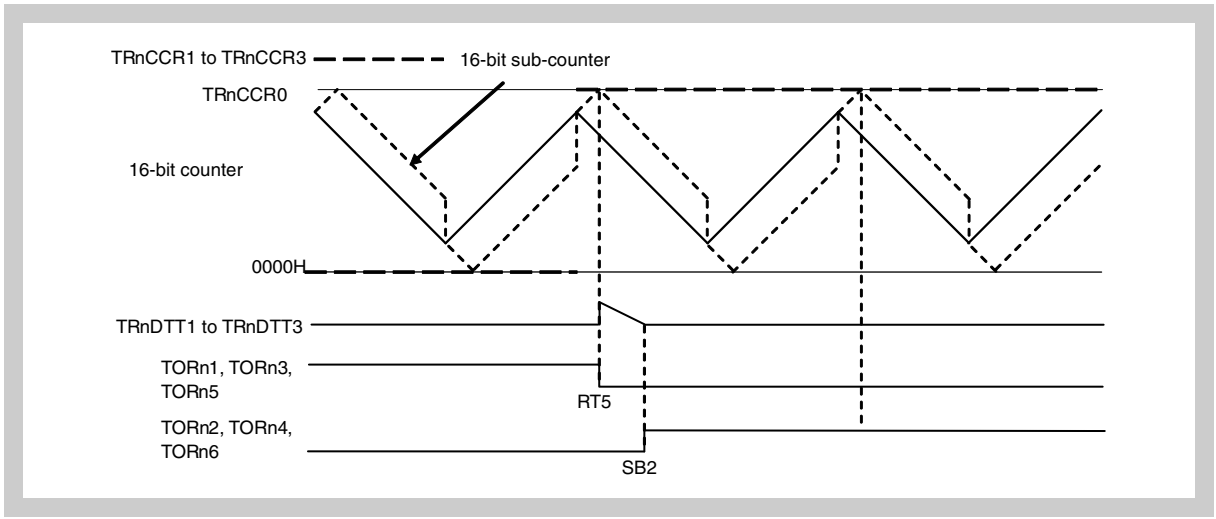


Figure 10-98 Timer output change after compare register updating by peak reload when "TRnCCR1 to TRnCCR3 = 0000_H" → "TRnCCR1 to TRnCCR3 = TRnCCR0"

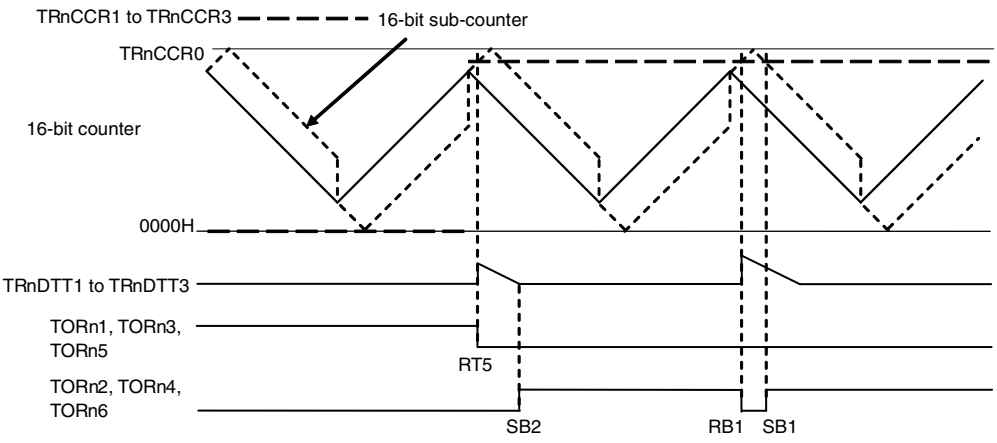


Figure 10-99 Timer output change after compare register updating by peak reload when "TRTRnCCR1 to TRnCCR3 = 0000_H" → "(TRnCCR0 − TRnDTC1) < TRTRnCCR1 to TRnCCR3 < TRnCCR0"

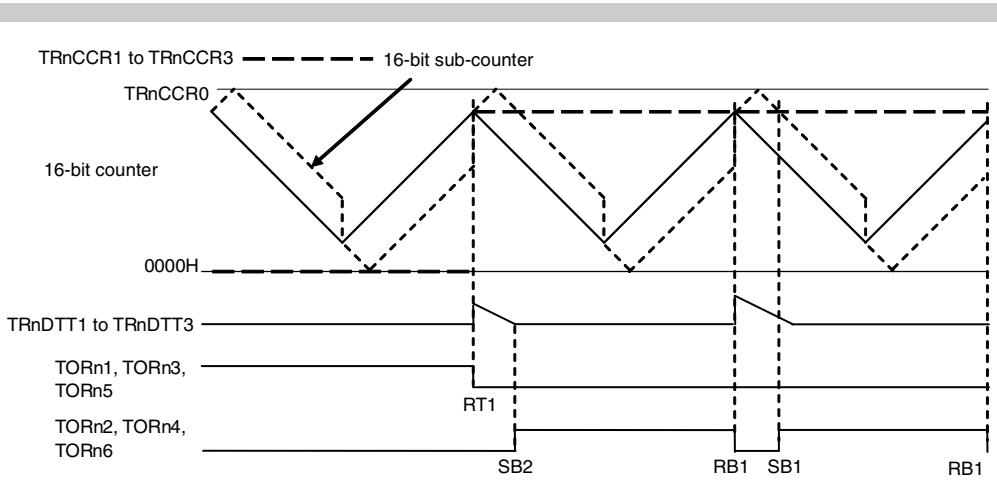


Figure 10-100 Timer output change after compare register updating by peak reload when "TRnCCR1 to TRnCCR3 = 0000_H"→ "TRnCCR1 to TRnCCR3 = (TRnCCR0 − TRnDTC1)"

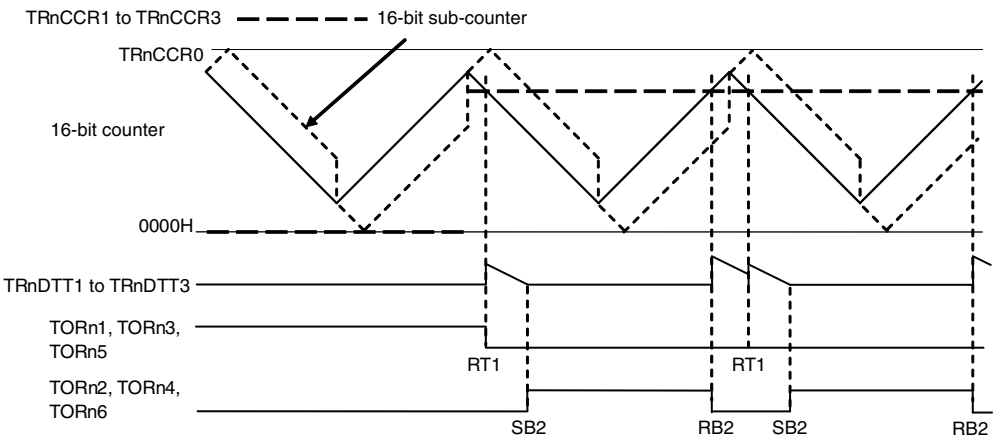


Figure 10-101 Timer output change after compare register updating by peak reload when "TRnCCR1 to TRnCCR3 = 0000_H" → "(TRnCCR0 – TRnDTC0 – TRnDTC1) < TRnCCR1 to TRnCCR3 < (TRnCCR0 – TRnDTC1)"

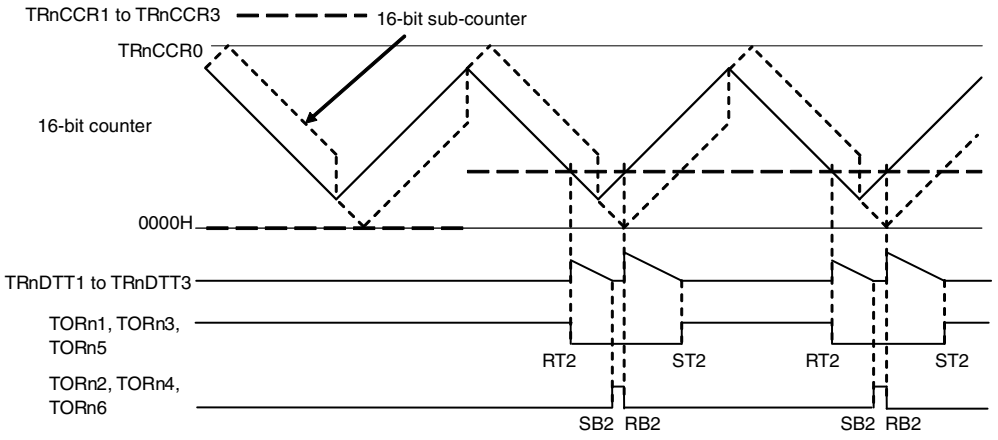


Figure 10-102 Timer output change after compare register updating by peak reload when "TRnCCR1 to TRnCCR3 = 0000_H" → "(TRnDTC0 + TRnDTC1) < TRnCCR1 to TRnCCR3 ≤ (TRnCCR0 – TRnDTC0 – TRnDTC1)"

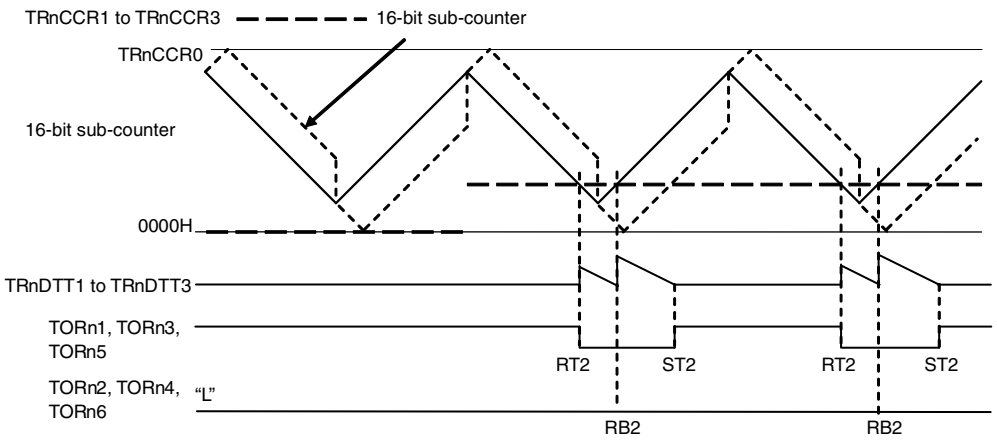


Figure 10-103 Timer output change after compare register updating by peak reload when "TRnCCR1 to TRnCCR3 = 0000_H" → "(TRnDTC0 + 1) < TRnCCR1 to TRnCCR3 < (TRnDTC0 + TRnDTC1)"

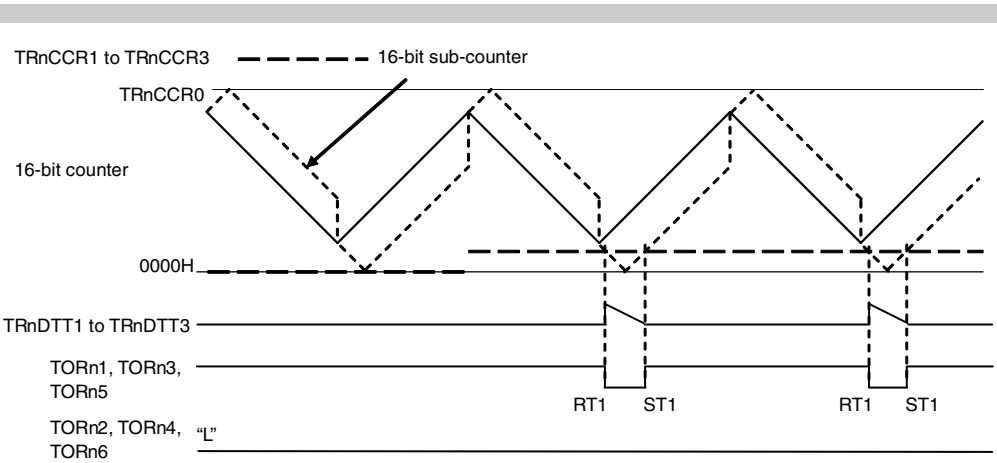


Figure 10-104 Timer output change after compare register updating by peak reload when "TRnCCR1 to TRnCCR3 = 0000_H" → "0000_H < TRnCCR1 to TRnCCR3 ≤ (TRnDTC0 + 1)"

(9) Dead time control in high-accuracy T-PWM mode

In the high-accuracy T-PWM mode, the TRnCCR1 to TRnCCR3 registers are used for duty setting and the TRnCCR0 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 (TRnDTC0, TRnDTC1).

The TRnDTC0 register is used to set the dead time from when a negative phase changes to inactive until a positive phase changes to active. The TRnDTC1 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 TRnDTC0 = x, TRnDTC1 = y is shown below.

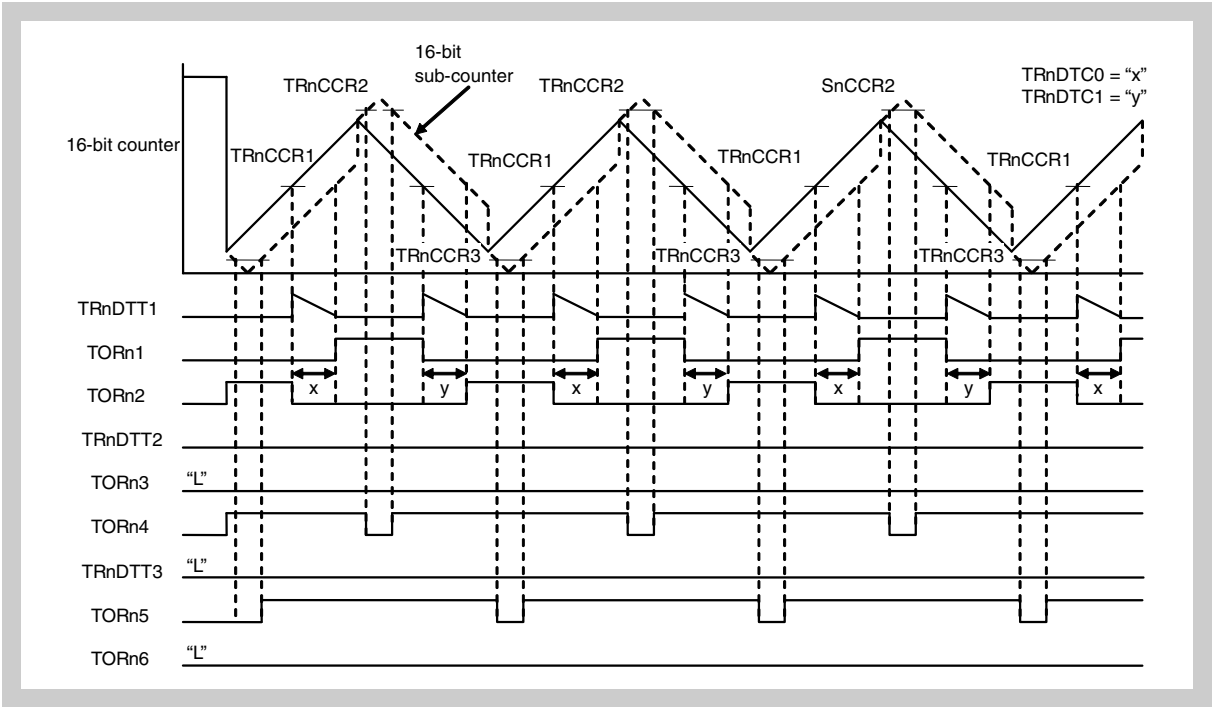


Figure 10-105 Output waveform example when dead time is set

(10) Cautions on dead time control in high-accuracy T-PWM mode

(a) Rewriting of TRnDTC0 and TRnDTC1 registers

The setting of the dead time in the TRnDTC0, TRnDTC1 registers can be rewritten during operation. Note the following cautions when rewriting the dead time setting during operation.

Caution

1. Rewrite the TRnDTC0 and TRnDTC1 registers when using the reload function (TRnCMS = 0).
2. When the TRnDTC0 and TRnDTC1 registers are rewritten, carrier-wave cycles will be changed. In cases where carrier-wave cycles should not be changed, rewrite the TRnCCR0 register value at the same time as changing the TRnDTC0 and TRnDTC1 registers.
3. Rewriting is prohibited when TRnCMS = 1.
4. In case of changing TRnCCR0 and TRnCCR1 at a 16-bit counter peak: Match interrupts (INTTRnCC1 to INTTRnCC5) will not occur immediately after reload execution if the values set in the TRnCCR1 to TRnCCR5 register matches with and TRnCCR0 − TRnDTC1 (the new maximum value of main counter) after updating.

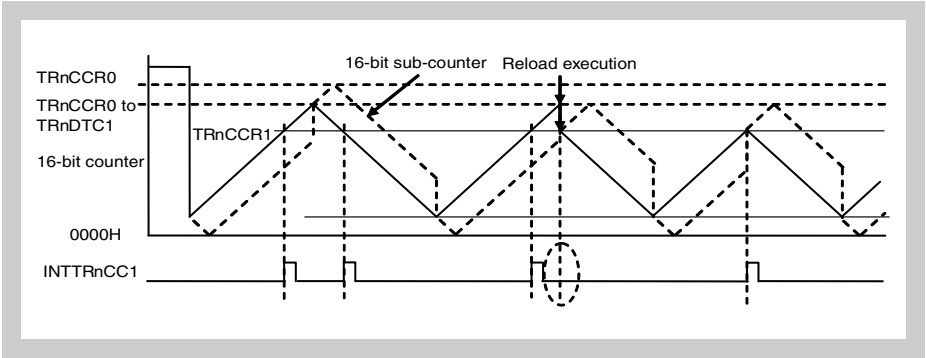


Figure 10-106 Interrupt occurence when changing TRnCCR0 and TRnCCR1 at a counter peak

5. In case of changing TR0DTC0 at a 16-bit counter trough: Match interrupts (INTTRnCC1 to INTTRnCC5) will not occur immediately after reload execution if the values set in the TRnCCR1 to TRnCCR5 register match with TR0DTC0 (the new minimum value of main counter) after updating.

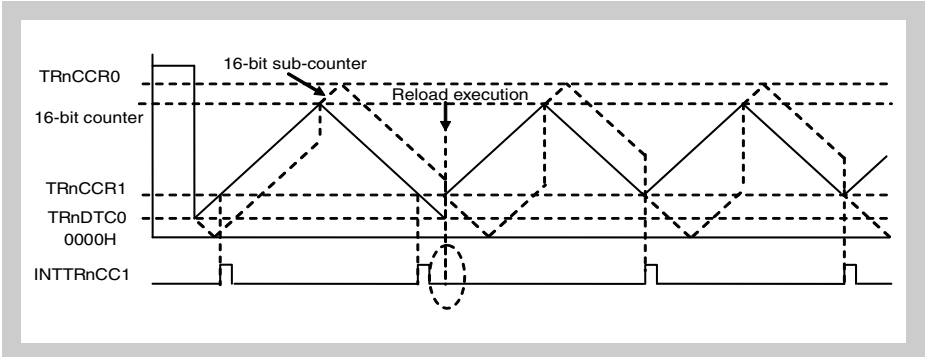


Figure 10-107 Interrupt occurence when changing TR0DTC0 at a counter trough

(11) Caution on rewriting cycles in high-accuracy T-PWM mode

In high-accuracy T-PWM mode, setting conditions for the TRnCCR0, TRnDTC0, and TRnDTC1 registers are as follows.

- $3 \times$ MAX (TRnDTC0, TRnDTC1) + MIN (TRnDTC0, TRnDTC1) < TRnCCR0
  0002_H < TRnCCR0 ≤ $FFFE_H$
- MAX (A, B) indicates the greater value of A and B, and
  MIN (A, B) indicates the smaller value of A and B.

Figure 10-108 shows an operation example when the setting range is exceeded.
This example shows the case where the TRnDTC0 register is set out of the range "TRnDTC0 ≥ TRnCCR0 − TRnDTC1". Though the 16-bit counter executes count-down operation, the count-down operation is executed from 0000_H because no match occurs. In this case, the count operation continues by loading the TRnDTC0 register setting value. However, no match with TRnCCR0 − TRnDTC1 occurs in the count-up operation, thus the 16-bit counter overflows. In this case, the count operation continues by loading the TRnDTC0 register setting value again.
An overflow interrupt (INTTRnOV) occurs when the 16-bit counter loads the TRnDTC0 register setting value from 0000_H or when an overflow occurs at FFFEH, and then the TRnOVF flag is set. An overflow interrupt (INTTRnOV) does not occur if the TRnCCR0, TRnDTC0, and TRnDTC1 registers are set correctly, so this can be used for detecting incorrect settings.

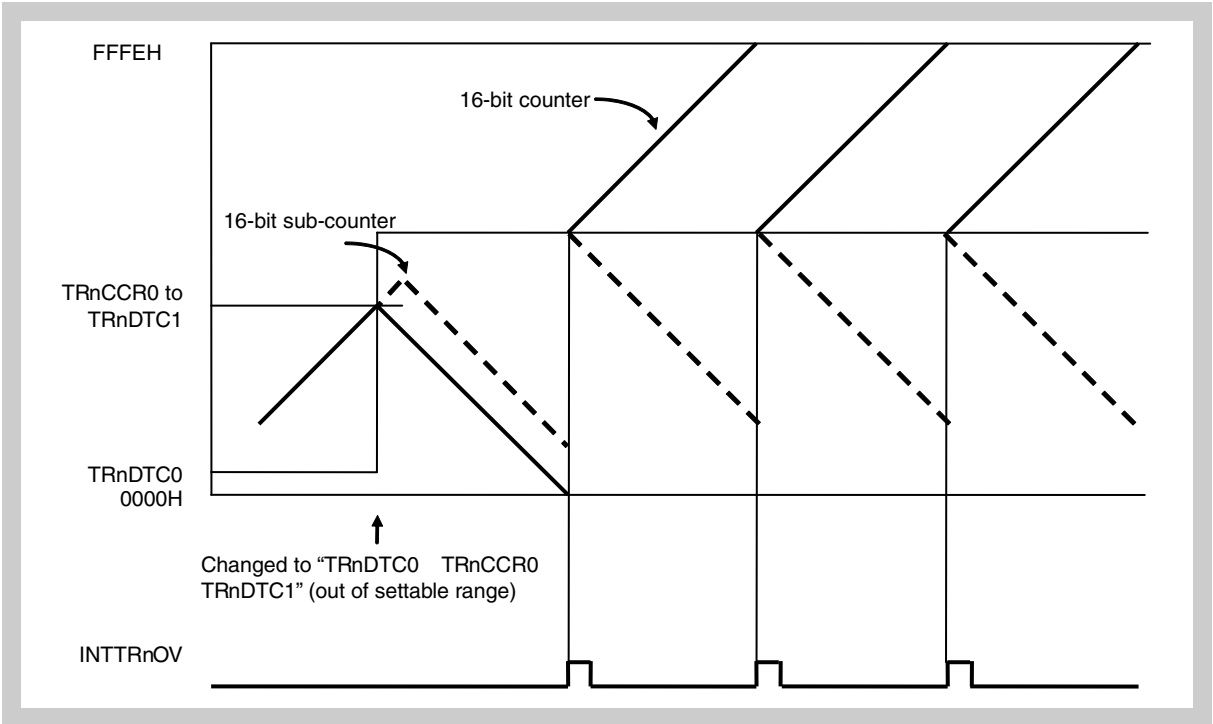


Figure 10-108 Operation example when setting is out of range

(12) Error interrupt (INTTRnER) 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 (INTTRnER) normally do not occur in high-accuracy T-PWM mode. In case of occurrence, the internal circuits may be damaged.

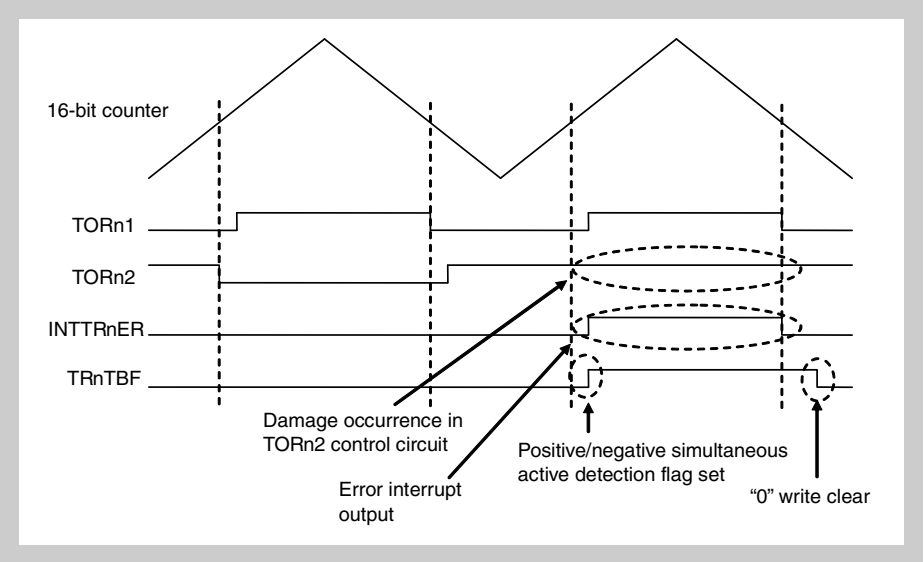


Figure 10-109 Error interrupt operation example

10.10.10 PWM mode with dead time

(1) Outline of PWM mode with dead time

In the PWM mode with dead time, 6-phase PWM is generated using the 16-bit counter's saw tooth wave operation and four 16-bit counters. The counter's maximum value is set with the TRnCCR0 register. The duties of the U phase, V phase, and W phase voltage data signals are set with the TRnCCR1 to TRnCCR3 registers. The dead time is set with the TRnDTC0 and TRnDTC1 registers, and the dead time for inverted phase → normal phase and the dead time for normal phase → inverted phase can be independently set with the TRnDTC0 register and TRnDTC1 register, respectively. The counter's operation consists in performing up count with 0000_H as the minimum value, and when the maximum value (cycle) indicated by the TRnCCR0 register is matched, the counter is cleared (0000_H), and the counter continues up-count operation.

The 10-bit dead time counters (TRnDTT1 to TRnDTT3) reload the setting value of the TRnDTC0 and TRnDTC1 registers upon a match between the counter and the TRnCCR1 to TRnCCR3 registers, and perform down count.

Upon a match between the 16-bit counter and the TRnCCR0 to TRnCCR3 registers, the corresponding compare match interrupts (INTTRnCC1 to INTTRnCC3) are output.

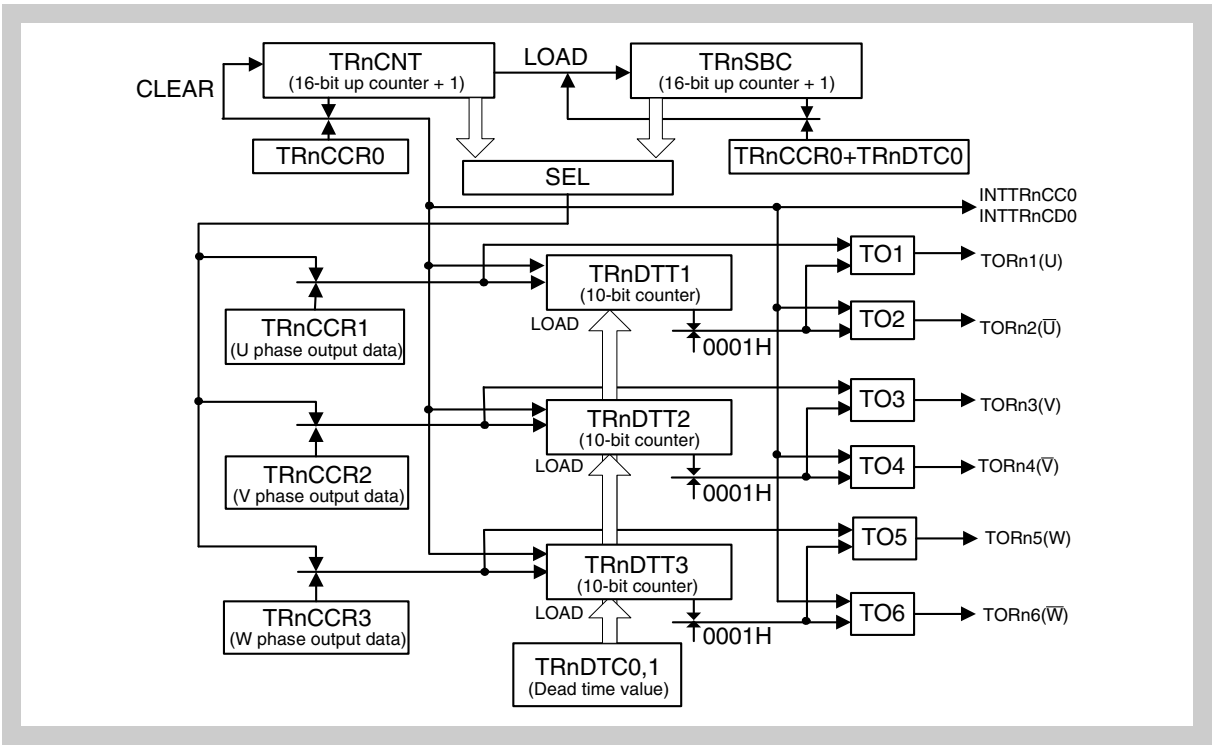


Figure 10-110 Block diagram in PWM mode with dead time

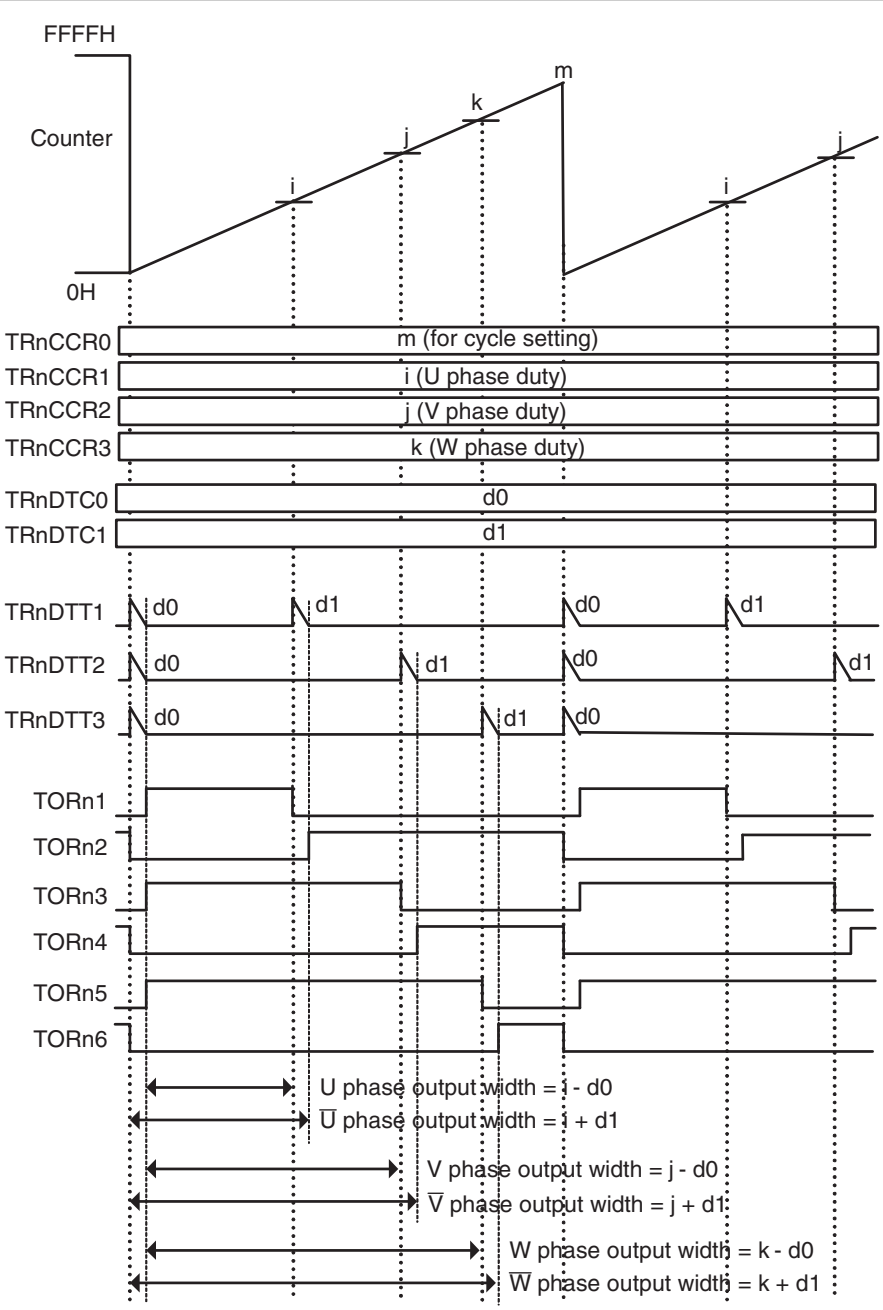


Figure 10-111 Output waveform example in PWM Mode with dead time

Remarks

1. The maximum value that can be set to the TRnCCR1 to TRnCCR3 registers is TRnCCR0 + TRnDTC0.
2. If "0000_H" is set to the TRnCCR1 to TRnCCR3 registers, PWM is output with 0% duty.
3. If TRnCCR0 + TRnDTC0 is set to the TRnCCR1 to TRnCCR3 registers, PWM is output with 100% duty.
4. The maximum value of the TRnCCR0 register is $FFFF_H$ - TRnDTC0.
5. Perform setting so as to satisfy condition $FFFF_H$ > TRnCCR0 + TRnDTC0.

(2) PWM mode with dead time operation list

(a) Compare registers

Register	Rewriting method	Rewriting during operation	Function
TRnCCR0	Reload	Possible	PWM cycle
TRnCCR1 to TRnCCR3	Reload	Possible	PWM duty
TRnCCR4, TRnCCR5	Reload	Possible	PWM duty

(b) Input pins

Pin	Function
TIR10 to TIR13	–
TTRGR1	–
TEVTR1	–

(c) Output pins

Pin	Function
TORn0	Toggle output upon TRnCCR0 register compare match
TORn1	PWM output (with dead time) upon TRnCCR1 register compare match
TORn2	Inverted phase output to TORn1
TORn3	PWM output (with dead time) upon TRnCCR2 register compare match
TORn4	Inverted phase output to TORn3
TORn5	PWM output (with dead time) upon TRnCCR3 register compare match
TORn6	Inverted phase output to TORn5
TORn7	Pulse output through A/D conversion trigger

(d) Interrupts

Interrupt	Function
INTTRnCC0 to INTTRnCC5	TRnCCR0 to TRnCCR5 compare match
INTTRnOV	–
INTTRnER	Error
INTTRnOD	–
INTTRnCD	Peak interrupt

(3) PWM mode with dead time settings

(a) Mode setting

The PWM mode with dead time is selected by setting TRnCTL1.TRnMD[4:0] = 1001_B.

(b) Output level/output enable settings

Output level/output enable is set by setting the TRnOL0 to TRnOL7 and TRnOE0 to TRnOE7 bits of the TRnIOC0 and TRnIOC3 registers.

Pin TORn0 performs toggle output upon cycle match (match between the counter and the TRnCCR0 register).

Pin TORn7 is the output for A/D conversion. Set this pin as required.

(c) Error output enable

Set error output enable when normal phase/inverted phase simultaneous active is detected. Error output is enabled by setting TRnIOC4.TRnEOC bit to 1. Moreover, the pin for detecting simultaneous active can also be set, by setting TRnIOC4.TRnTBA[2:0] bits. In the PWM mode with dead time, INTTRnER does not become active, regardless of which value the user sets to the TRnCCR0 to TRnCCR3 registers, except when the dead time setting is 0. When an error occurs, this indicates an internal circuit fault.

(d) Interrupt and thinning out function settings

A peak interrupt (INTTRnCD) occurs upon a match between the TRnCCR0 register and the counter (TRnOPT1.TRnIOE bit control is invalid). To output a peak interrupt, set TRnOPT1.TRnICE = 1. Use of the thinning out function for peak interrupts is done with the TRnOPT1.TRnID[4:0] bits.

(e) Reload thinning out function setting

To set the reload timing to the same timing as the interrupt timing, set TRnOPT1.TRnRDE bit to 1.

The reload timing occurs when TRnOPT1.TRnICE = 1.

(f) A/D conversion trigger output setting

A/D conversion trigger 0 (TRnADTRG0 signal) is set with TRnOPT2 register bits TRnAT04, TRnAT02, and TRnAT01. The TRnCCR5 register match timing, TRnCCR4 register match timing, and peak interrupt (INTTRnCD) enable/disable settings are performed with bits TRnAT04, TRnAT02, and TRnAT01.

Do not set TRnAT05, TRnAT03, and TRnAT00 to “1”.

A/D conversion trigger 1 (TRnADTRG1 signal) is set with TRnOPT3 register bits TRnAT14, TRnAT12, and TRnAT11. The TRnCCR5 register match timing, TRnCCR4 register match timing, and peak interrupt (INTTRnCD) enable/disable settings are performed with bits TRnAT14, TRnAT12, and TRnAT11.

Do not set bits TRnAT15, TRnAT13, and TRnAT10 to “1”.

Set the compare values of the TRnCCR4 and TRnCCR5 registers.

(g) Dead time settings

The dead time settings are performed with the TRnDTC0 and TRnDTC1 registers. The dead time can be obtained with count clock cycle × TRnDTC0,TRnDTC1.

The time until TORn2, TORn4, TORn6 pin inactive change → TORn1, TORn3, TORn5 pin active change can be set with the TRnDTC0 register.

The time until TORn1,TORn3,TORn5 pin inactive change → TORn2, TORn4, TORn6 pin active change can be set with the TRnDTC1 register.

(h) PWM cycle, duty (PWM width) setting

The duty is set with the TRnCCR1 to TRnCCR3 registers. The setting range of the TRnCCR1 to TRnCCR3 registers is

$$0000H \leq TRnCCRm \leq (TRnCCR0 + TRnDTC0)$$

The TRnCCR0 and TRnDTC0 registers must be set so as to satisfy

$$TRnCCR0 + TRnDTC0 < FFFFH.$$

Remark m = 1 to 3

(4) Operation in PWM mode with dead time

Figure 10-112 shows the timing chart when TRnCCR0 = 0007H, TRnDTC0 = 0002_H, TRnDTC1 = 0002_H, and the TRnCCR0 register is set to 0000_H to 0007_H (in part).

When the compare value of the TRnCCR1 register is incremented/decremented by 1 at a time, the PWM width is incremented/decremented 1 count clock at a time, but at the points indicated by arrows in *Figure 10-112*, incrementing/decrementing is done by TRnDTC1+1 count clock. This occurs when the TRnCCR1 register is rewritten from the setting value of the TRnDTC0 register to TRnDTC0+0001_H (because dead time control is required).

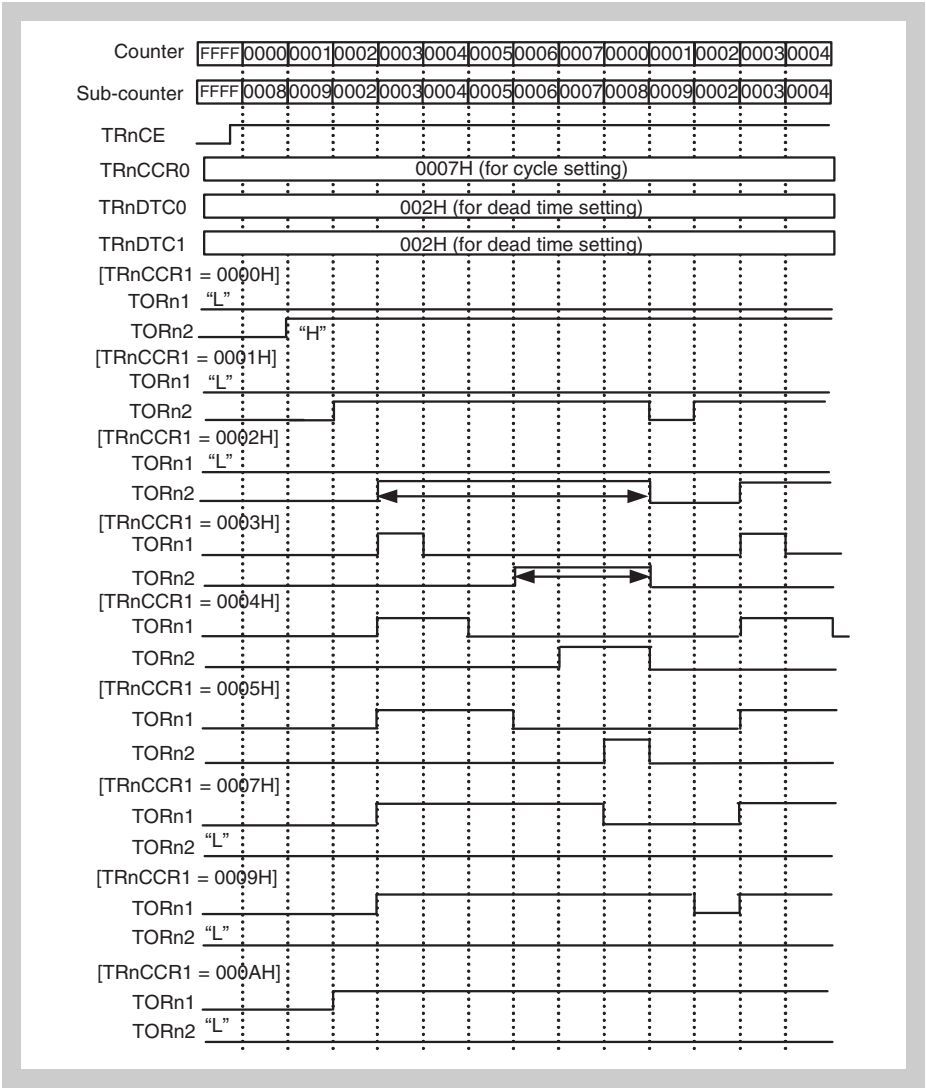


Figure 10-112 Timer output example when TRnCE = 1 is set (initial setting in PWM mode with dead time)

(5) Dead time control in PWM mode with dead time

In the PWM mode with dead time, compare registers (TRnCCR1 to TRnCCR3) are used as the duty setting registers, and another compare register (TRnCCR0) is used as the cycle setting register. Through the use of these four registers, a variable duty 6-phase PWM waveform is output. To realize dead time control, three 10-bit down counters that operate in synchronization with the counter's count clock, and dead time setting registers (TRnDTC0, TRnDTC1) are provided. The TRnDTC0 register is used to set the dead time from when the inverted phase becomes inactive to when the normal phase becomes active, and the TRnDTC1 register is used to set the dead time from when the normal phase becomes inactive to when the inverted phase becomes active.

The following figure shows an output example when TRnDTC0 = x, TRnDTC1 = y.

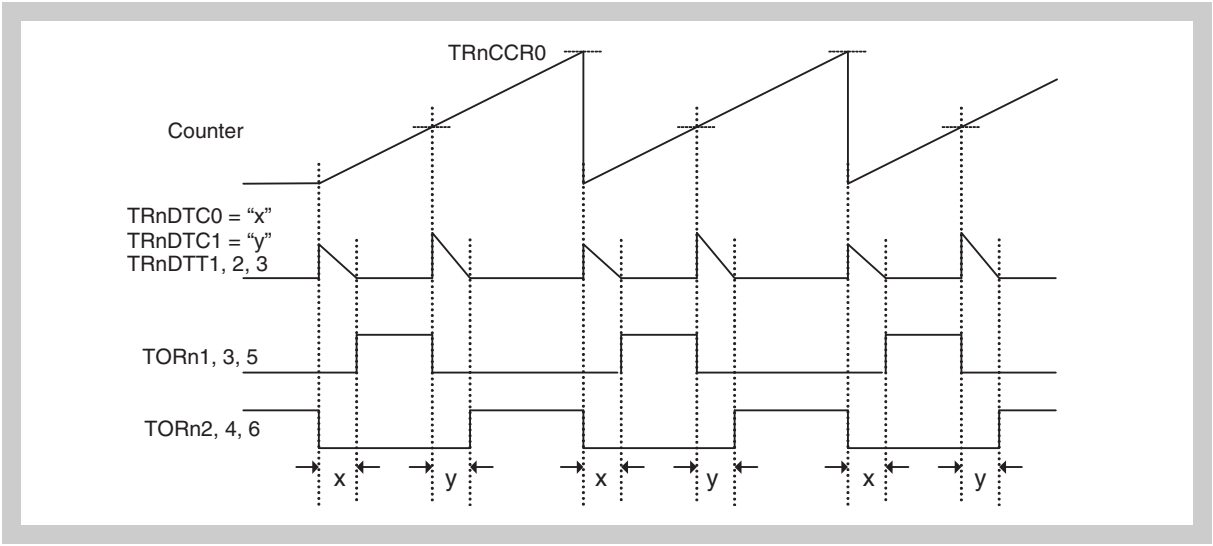


Figure 10-1 Output waveform example in PWM mode with dead time

(6) Error interrupt (INTTRnER) in PWM mode with dead time

In the PWM mode with dead time, the normal phase/inverted phase simultaneous active detection function can be used. When using the PWM mode with dead time, no error interrupt (INTTRnER) is output as long as no hardware fault occurs (except when TRnDTC0, TRnDTC1 = 0000_H is set).

Also, when TRnDTC0, TRnDTC1 = 000_H is set, glitches may occur upon error interrupt (INTTRnER) output. In this case, the occurrence of glitches during error interrupt (INTTRnER) output can be prevented by setting bit TRnEOC to 0.

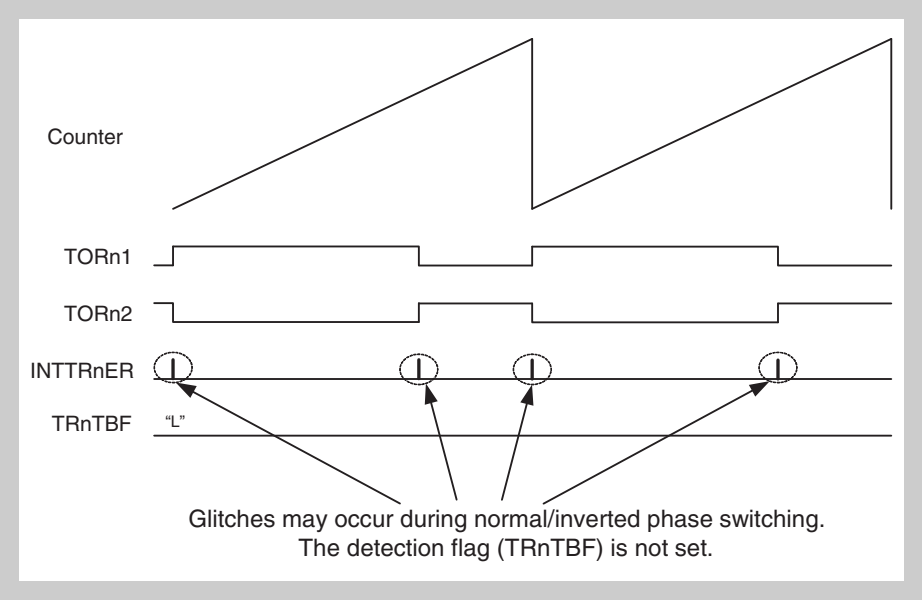


Figure 10-2 Error interrupt (INTTRnER) in PWM mode with dead time

10.11 Cautions

(1) Capture operation of illegal data before first counting up

In free-running mode (TR1MD[3:0] = 0101_B), and pulse width measurement mode (TR1MD[3:0] = 0110_B), when a lower count clock (TR1EEE = 0, TR1CKS[2:0] = 011_B to 111_B) or an external clock (TR1EEE = 1) is selected, the timer captures the value of $FFFF_H$ and outputs a capture interrupt signal (INTTR1CCm), if a capture trigger signal (TIR1m) is enabled and input before first counting up. This captured data and the corresponding interrupt might be useless.

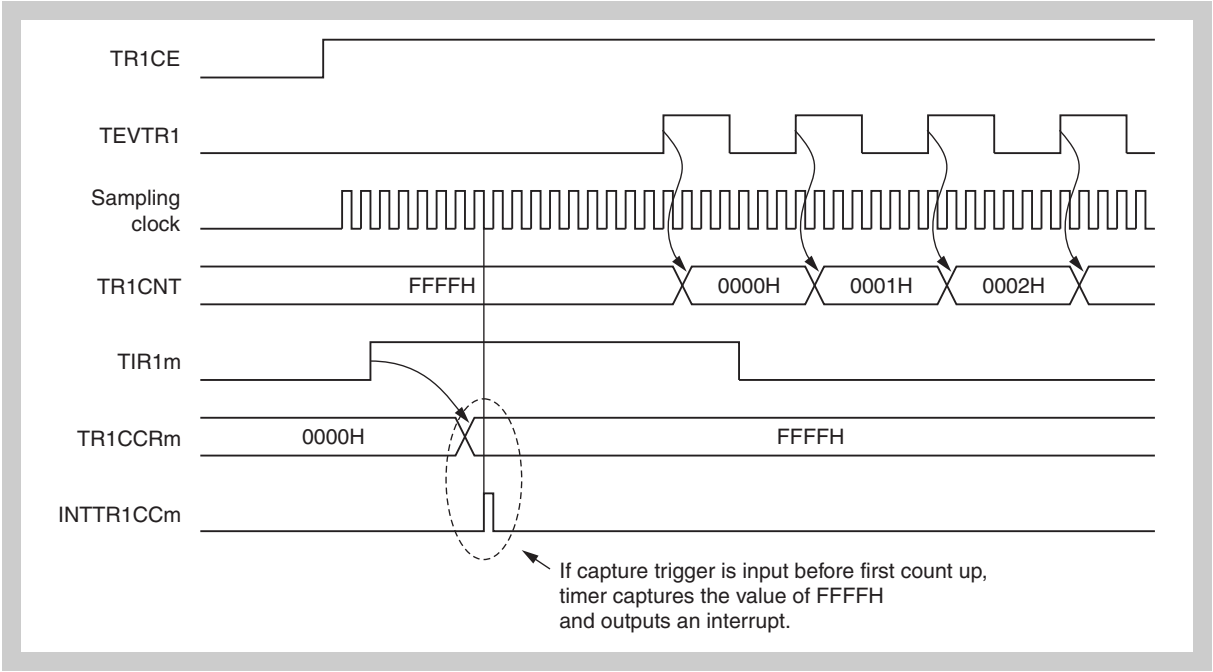


Figure 10-3 Capture operation of illegal data before first counting up

Chapter 11 16-bit Timer/Event Counter T (TMT)

This microcontroller has two instances of the Timer T (TMT), TMT0 and 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.

11.1 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

11.2 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

11.3 Configuration

Timer T is configured of the following hardware.

Table 11-1 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)[a]
Timer output pins	2 (TOTn0, TOTn1)[a]
Timer input signals	
Timer output signals	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) TMP input control register 2 (TAAIC2, refer to *"TMP input control register 2 (TPIC2)" on page 227*)
Interrupts	Compare match interrupt (INTTTnCC0, INTTTnCC1) Overflow interrupt (INTTTnOV) Encoder clear interrupt (INTTTnEC)

a) Alternate function pins

Table 11-2 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 [a]
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 11-2 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 [a]
FFFFF9A0H	TMT1 counter write buffer register	TT1TCW	R/W			×	0000H

a) when TTnCTL0.TTnCE = 0

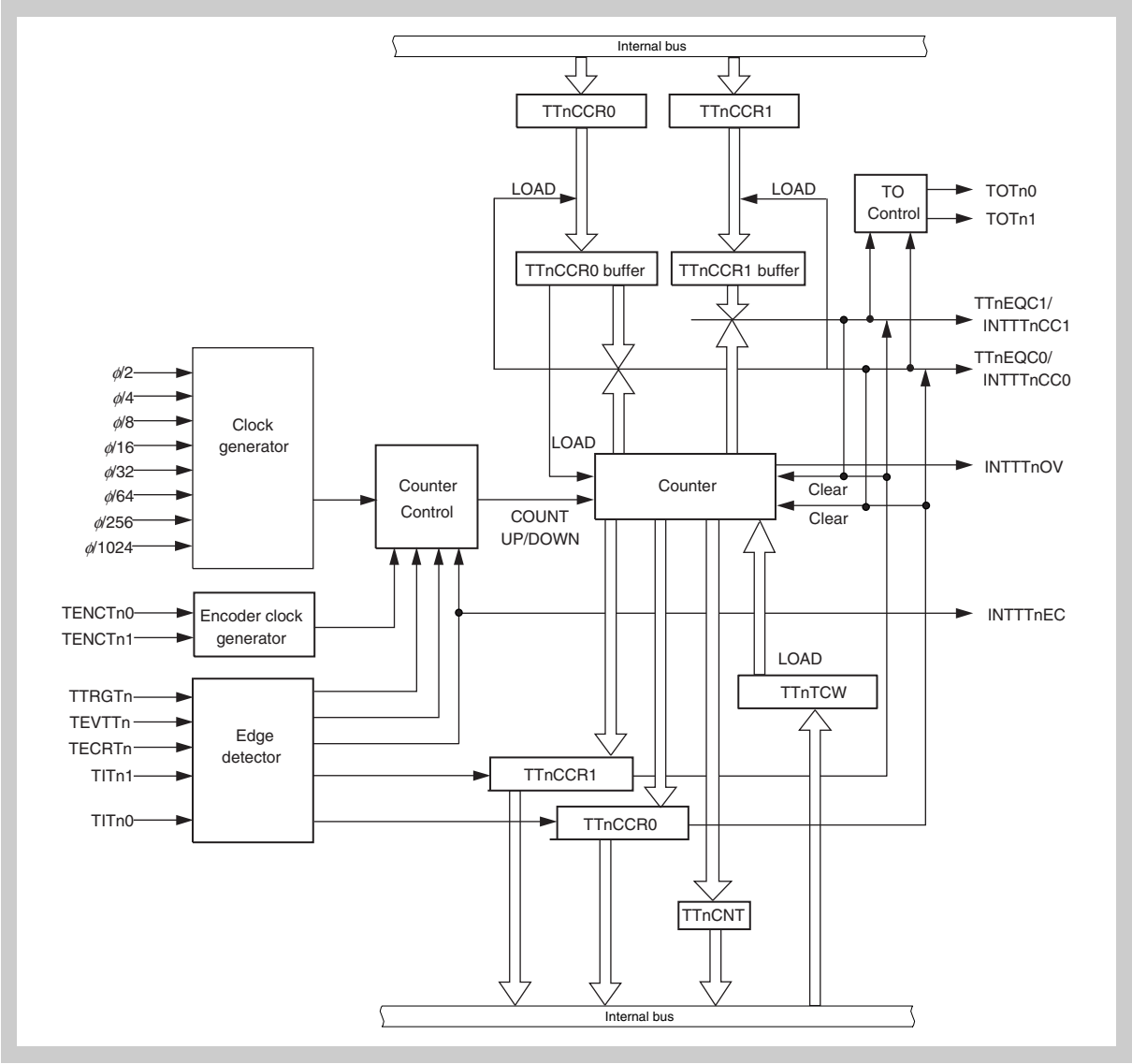


Figure 11-1 Block diagram of Timer T

(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: $FFFFF69A_H$
TT1CCR0: $FFFFF6AA_H$

Initial Value 0000_H. These registers are cleared by any reset.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTnCCR0	Capture/compare value 0															
	R/W															

The capture and compare functions are as follows in each mode.

Table 11-3 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)Note 1	Compare match
Encoder compare mode	Compare only	Anytime write	Depends on set conditionNote 2
Encoder capture mode	Capture only	-	-
Encoder capture compare mode	Compare only	Anytime write	Depends on set conditionNote 2
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 TTnCTL2.TTnECM[0:1] bits.

Use as compare register When TTnCE = 1, the TTnCCR0 register rewrite method differs according to the operation mode. Refer to *Table 11-3*.
(For details about the compare register rewrite operation, refer to *"Method for writing to compare register" on page 447*.)

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 11-3*.

(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: $FFFFF69C_H$
TT1CCR1: $FFFFF6AC_H$

Initial Value 0000_H. These registers are cleared by any reset.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTnCCR1	Capture/compare value 0															
	R/W															

The capture/compare functions in each operation mode are as follows.

Table 11-4 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)Note 1	-
Encoder compare mode	Compare only	Anytime write	Depends on set conditionsNote 2
Encoder capture mode	Capture only	-	-
Encoder capture compare mode	Capture only	-	-
Offset trigger generation mode	Compare only	Batch write (Reload)Note 3	-

Note
1. The batch write reload timing is the counter underflow occurrence timing only.
2. The conditions are set with TTnCTL2.TTnECM[0:1] bits.
3. The batch write reload timing is the counter's 0000_H 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 11-4*.
(For details about the compare register rewrite operation, refer to *"Method for writing to compare register" on page 447*.)

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 11-4*.

(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: $FFFFF69E_H$
TT1CNT: $FFFFF6AE_H$

Initial Value 0000_H. 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_H can be read.
- When bit TTnECC = 1, the value held when bit TTnCE was cleared to “0” can be read.

TTnCNT

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

(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: $FFFFF990_H$
TT1TCW: $FFFFF9A0_H$

Initial Value 0000_H. This register is cleared by any reset.

Note When TTnCTL2.TTnECC = 0, the setting value is loaded to the counter when the TTnCE bit is set (to 1). (When TTnCTL2.TTnECC = 1, the counter holds its value, so it is not reloaded.)

TTnTCW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Counter value															
R/W															

11.4 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: $FFFFF690_H$
TT1CTL0: $FFFFF6A0_H$

Initial Value 00_H. This register is cleared by any reset.

TTnCTL0

7	6	5	4	3	2	1	0
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 11-5 TTnCTL0 register contents (1/2)

Bit position	Bit name	Function
7	TTnCE	Controls the timer TMTn operation. 0: Internal operating clock operation disabled (TMTn reset asynchronously) 1: Internal operating clock operation enabled When bit TTnCE is set to “0”, the internal operation clock of TMTn stops (fixed to low level), and TMTn is reset asynchronously. 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 11-6, “TMTn Count Clock and Count Delay,” on page 429*. **Note: 1.** 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. • Compare match detector (interrupt output low level) • Timer output (Output inactive level) • Edge detector for other than pins TENCTn0, TENCTn1, and TECRTn **2.** The following functions are not reset. • Counter • Flags in TTnOPT1 register • TTnCCR0 buffer, TTnCCR1 buffer register, counter read buffer register • TENCTn0, TENCTn1, TECRTn pin edge detector **3.** In modes other than the above, (in which TTnECC is fixed to 0), the functions that are reset by TTnCE = 0 are as follows. • Internal registers other than registers that can be written from the CPU, and internal latch circuits

Table 11-5 TTnCTL0 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>2 to 0</td><td>TTnCKS2
TTnCKS1
TTnCKS0</td><td>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>$f_{XX}/2$</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>$f_{XX}/4$</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>$f_{XX}/8$</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>$f_{XX}/16$</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>$f_{XX}/32$</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>$f_{XX}/64$</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>$f_{XX}/256$</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>$f_{XX}/1024$</td></tr>
</table>
Note: 1. Set the TTnCKS[2:0] bits while TTnCE = 0.
The TTnCKS[2:0] bits can be changed simultaneously when the setting of TSnCE is changed from 0 to 1.
2. The TTnCKS[2:0] bits are valid only when TSnCTL1.TSnEEE is set to 0.</td></tr>
</table>

Table 11-6 TMTn Count Clock and Count Delay

Count clocks	TTnCKS2	TTnCKS1	TTnCKS0	Count delay	
				Minimum	Maximum
$f_{XX}/2$	0	0	0	3 base clocks	4 base clocks
$f_{XX}/4$	0	0	1		
$f_{XX}/8$	0	1	0		
$f_{XX}/16$	0	1	1	4 base clocks	5 base clocks + 1 count clock
$f_{XX}/32$	1	0	0		
$f_{XX}/64$	1	0	1		
$f_{XX}/256$	1	1	0		
$f_{XX}/1024$	1	1	1		

Note 1. f_{XX}: System clock

2. f_{TMTn}: Base clock of TMTn ($f_{TMTn} = f_{XX}/2$)

(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: $FFFFF691_H$
TT1CTL1: $FFFFF6A1_H$

Initial Value 00_H. 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, TTnEEE and TTnMD[3:0] bits can be written using the same value.

Table 11-7 TTnCTL1 register contents (1/2)

Bit position	Bit name	Function
6	TTnEST	Sets to enable/disable software trigger control. 0: No operation/Disables software trigger control. 1: Enables software trigger control. • 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. • In external trigger pulse output mode (Pulse output software trigger) **Caution:** 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. **Note:** The TTnEST bit is always read as “0”
5	TTnEEE	Specifies count clock input. 0: Use the clock selected with the TTnCTL0.TTnCKS[2:0] bits. 1: Use the external clock (TEVTTn pin input edge). Specification of the valid edge when TTnEEE = 1 (external clock: TEVTTn pin) is set with bits TTnEES1 and TTnEES0 of TTnIOC2 register.) **Note:** The setting of bit TTnEEE is invalid in the external event count mode, encoder compare mode, encoder capture mode, encoder capture/compare mode. **Caution:** 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.

Table 11-7 TTnCTL1 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3 to 0</td><td>TTnMD3
TTnMD2
TTnMD1
TTnMD0</td><td>Selects the timer mode..
<table>
<tr><th>TTnMD3</th><th>TTnMD2</th><th>TTnMD1</th><th>TTnMD0</th><th>Internal count clock selection</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Interval mode</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>External event count mode</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>External trigger pulse output mode</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>One-shot pulse mode</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>PWM mode</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>Free-running mode</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>Pulse width measurement mode</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>Triangular wave PWM mode</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>Encoder compare mode</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>Encoder capture mode</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>Encoder capture compare mode</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>Offset trigger generation mode</td></tr>
<tr><td colspan="4">Other than above</td><td>Setting prohibited</td></tr>
</table>
Caution: Rewrite the TTnMD[3:0] 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.</td></tr>
</table>

(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: $FFFFF692_H$
TT1CTL2: $FFFFF6A2_H$

Initial Value 00_H. This register is cleared by any reset.

TTnCTL2

7	6	5	4	3	2	1	0
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 11-8 TTnCTL2 register contents (1/2)

Bit position	Bit name	Function
7	TTnECC	Selection of initialization/hold of counter value. 0: Initialize counter value when TTnCTL0.TTnCE = 0. 1: Hold counter value when TTnCTL0.TTnCE = 0. • When TTnECC = 0, setting TnCTL0.TTnCE = 0 causes the counter to be reset to $FFFF_H$, the capture registers (TTnCCR0/TTnCCR1) to be reset to 0000_H, 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. • 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. **Note:** 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.)
4	TTnLDE	Enables encoder load. 0: Disable transfer of compare setting value to counter. 1: Enable transfer of compare setting value (TTnCCR0) to counter when underflow occurs. **Note:** 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. • TTnECM1 = 0, TTnECM0 = 0 or 1
3	TTnECM1	Enables encoder clear mode on match of counter and TTnCCR1 register. 0: No clear condition. 1: When the counter and TTnCCR1 register match, clear the counter if the next count is a down count (TTnESF = 1) **Note:** 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.)

Table 11-8 TTnCTL2 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>2</td><td>TTnECM0</td><td>Enables encoder clear mode on match of counter and TTnCCR0 register.
0: No clear condition.
1: When the counter and TTnCCR0 register match, clear the counter if the next count is a down count (TTnESF = 1)
Note: 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.)</td></tr>
<tr><td>1, 0</td><td>TTnUDS1
TTnUDS0</td><td>Selects the encoder operation mode..
<table>
<tr><th>TTnUDS1</th><th>TTnUDS0</th><th>Encoder operation mode</th></tr>
<tr><td>0</td><td>0</td><td>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.
• When "high", count down.
• When "low", count up.</td></tr>
<tr><td>0</td><td>1</td><td>Count up upon detection of valid edge of A phase of encoder input (TENCTn0 pin).
Count down upon detection of valid edge of B phase of encoder input (TENCTn1 pin).</td></tr>
<tr><td>1</td><td>0</td><td>Count up at rising edge of A phase of encoder input (TENCTn0 pin). Count down at falling edge of A phase of encoder input.
However, count operation is performed only when B phase of encoder input (TENCTn1 pin) is “low”.</td></tr>
<tr><td>1</td><td>1</td><td>Detection of both edges of phase A of encoder input (TENCTn0 pin)/phase B of encoder input (TENCTn1 pin).
Judgment of count operation based on combination of detection edge and input level.</td></tr>
</table>
Note: When TTnUDS[1:0] bits are set to 10_B or 11_B, the settings of bits TTnIOC3.TTnEIS[1:0] are invalid, and the function is fixed to the setting for detection of both edges.</td></tr>
</table>

(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: $FFFFF693_H$
TT1IOC0: $FFFFF6A3_H$

Initial Value 00_H. This register is cleared by any reset.

TTnIOC0

7	6	5	4	3	2	1	0
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 11-9 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: $FFFFF694_H$
TT1IOC1: $FFFFF6A4_H$

Initial Value 00_H. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
TTnIOC1	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
1. Set the TTnIOC1 register when TTnCTL0.TTnCE = 0. When TTnCE = 1, write access to the TTnIOC1 register can be performed with the same value.
2. The TT1IOC1 register setting is valid only in free-running mode, pulse width measurement mode, encoder capture mode, and encoder capture compare mode. In all other modes, capture operation is not performed.

Table 11-10 TTnIOC1 register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3, 2</td><td>TTnIS3
TTnIS2</td><td>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>
Capture operation is performed and capture interrupt (INTTTnCC1) is output upon edge detection.</td></tr>
<tr><td>1, 0</td><td>TTnIS1
TTnIS0</td><td>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>
Capture operation is performed and capture interrupt (INTTTnCC0) is output upon edge detection.</td></tr>
</table>

(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: $FFFFF695_H$
TT1IOC2: $FFFFF6A5_H$

Initial Value 00_H. 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 11-11 TTnIOC2 register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3, 2</td><td>TTnEES1
TTnEES0</td><td>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>
Note: The settings of bits TTnEES1 and TTnEES0 are valid in the external event count mode, or when bit TTnCTL1.TTnEEE = 1.</td></tr>
<tr><td>1, 0</td><td>TTnETS1
TTnETS0</td><td>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>
Note: The settings of bits TTnETS1 and TTnETS0 are valid in the external trigger pulse output mode and the one-shot pulse mode.</td></tr>
</table>

(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: $FFFFF696_H$
TT1IOC3: $FFFFF6A6_H$

Initial Value 00_H. This register is cleared by any reset.

TTnIOC3

7	6	5	4	3	2	1	0
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 11-12 TTnIOC3 register contents (1/2)

Bit position	Bit name	Function
7	TTnSCE	Selects the encoder counter clear method. 0: Clear upon detection of edge of TECRTn pin 1: Clear upon match of clear condition level • When TTnSCE = 1, the counter is cleared to 0000_H if all the conditions set with bits TTnZCL, TTnBCL, and TTnACL are matched. • When TTnSCE = 1, the settings of bits TTnECS1 and TTnECS0 are invalid, so no encoder clear interrupt (INTTTnEC) is output. • 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. **Caution:** When TTnSCE = 1, be sure to set bits TTnUDS1, and TTnUDS0 of the TTnCTL2 register to 10_B or 11_B.
6	TTnZCL	Specifies the clear level for the Z phase of encoder input (TECRTn pin) 0: Clear condition = Low level 1: Clear condition = High level **Note:** The TTnZCL bit is valid when TTnSCE = 1.
5	TTnBCL	Specifies the clear level for the B phase of encoder input (TENCTn1 pin) 0: Clear condition = Low level 1: Clear condition = High level **Note:** The TTnBCL bit is valid when TTnSCE = 1.
4	TTnACL	Specifies the clear level for the Aphase of encoder input (TENCTn0 pin) 0: Clear condition = Low level 1: Clear condition = High level **Note:** The TTnACL bit is valid when TTnSCE = 1.

Table 11-12 TTnIOC3 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3, 2</td><td>TTnECS1
TTnECS0</td><td>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>
The encoder clear interrupt (INTTTnEC) is output upon detection of the valid edge set with bits TTnECS1, TTnECS0.

Caution: When TTnSCE = 1, the encoder clear interrupt (INTTTnEC) is not output.

Note: Bits TTnECS1 and TTnECS0 are valid in the encoder compare mode, encoder capture mode, and encoder capture/compare mode and when TTnSCE = 0.</td></tr>
<tr><td>1, 0</td><td>TTnEIS1
TTnEIS0</td><td>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>
Note: Bits TTnEIS1 and TTnEIS0 are valid when bits TTnCTL2.TTnUDS[1:0] are 00_B or 01_B.</td></tr>
</table>

(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: $FFFFF697_H$
TT1OPT0: $FFFFF6A7_H$

Initial Value 00_H. This register is cleared by any reset.

TTnOPT0

7	6	5	4	3	2	1	0
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 11-13 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 **Note:** 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 **Note:** 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 $FFFF_H$, overflow occurs, the TTnOVF flag is set (1), and the counter is cleared to 0000_H. 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. **Note:** Overflow does not occur during compare match & clear operation for counter value $FFFF_H$ and compare value $FFFF_H$. **Caution:** 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: $FFFFF698_H$
TT1OPT1: $FFFFF6A8_H$

Initial Value 00_H. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
TTnOPT1	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 11-14 TTnOPT1 register contents

Bit position	Bit name	Function
2	TTnEUF	Indicates encoder underflow 0: No underflow indicated 1: Indicates counter underflow in the encoder compare mode, encoder capture mode, and encoder capture/compare mode. If the counter value is counted down from 0000_H, underflow occurs, the OVF flag is set (to 1), and the counter is set to $FFFF_H$. When the TTnEUF flag is set (to 1), an overflow interrupt (INTTTnOV) occurs at the same time. The TTnEUF flag is cleared (to 0) under the following conditions. • When 0 is written by CPU instruction • When TTnCTL0.TTnCE = 0 is set while TTnCTL2.TTnECC = 0 **Note:** 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. **Caution:** 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.
1	TTnEOF	Indicates encoder overflow 0: No overflow indicated 1: Indicates counter overflow in the encoder compare mode, encoder capture mode, and encoder capture/compare mode. If the counter value is counted up from $FFFF_H$, overflow occurs, the OVF flag is set (1), and the counter is cleared to 0000_H. 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). The TTnEOF flag is cleared (0) under the following conditions. • When 0 is written by CPU instruction • When TTnCTL0.TTnCE = 0 is set while TTnCTL2.TTnECC = 0 **Note:** 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. **Caution:** 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.
0	TTnESF	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. The TTnESF flag is cleared (to 0) under the following conditions. • When TTnCTL0.TTnCE = 0 is set while TTnCTL2.TTnECC = 0 **Note:** 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.

(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: $FFFFF699_H$
TT1OPT1: $FFFFF6A9_H$

Initial Value 00_H. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
TTnOPT2	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 11-15 TTnOPT2 register contents

Bit position	Bit name	Function
2	TTnRSF	Indicates the reload status 0: No reload request, or reload completed 1: Reload request was output This flag indicates that the data to be transferred next is held pending in the TTnCCR0 and TTnCCR1 registers. The TTnRSF flag is set (1) by writing to the TTnCCR1 register, and it is cleared (0) upon reload completion. **Caution:** When TTnRSF = 1, do not perform write access to the TTnCCR0 and TTnCCR1 registers.

(11) TPIC1 - TMP Input Control Register 2

Beside other, this register controls the assignment of external input sources of TMTn (TITn0, TITn1). For details refer to *"TMP input control register 2 (TPIC2)" on page 227*

11.5 Basic Operation

11.5.1 Basic counter operation

This section describes the basic operation of the counter. For details, refer to chapter *"Operation in Each Mode" on page 451*.

(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 $FFFF_H$.

It counts up $FFFF_H$, 0000_H, 0001_H, 0002_H, 0003_H…

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 $FFFF_H$.

It counts up $FFFF_H$, 0000_H, 0001_H, 0002_H, 0003_H…

(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 11-16 Counter Clear Operation

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	-	-
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[Note]	Depends on set conditions[Note]	Pin TECRTn, clear condition level match
Encoder capture mode	-	-	Pin TECRTn, clear condition level match
Encoder capture compare mode	Depends on set conditions[Note]	-	Pin TECRTn, clear condition level match
Offset trigger generation mode	-	-	External input (TITn0 pin)

Note Conditions are set with TTnCTL2.TTnECM[1:0] bits.

(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 TTnCTL2.TTnECC bit.

If TTnCE = 0 is set when TTnECC = 0, the counter is reset to 0000_H. 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 $FFFF_H$ to 0000_H.

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 $FFFF_H$ to 0000_H immediately after counting start
- When $FFFF_H$ is set to the compare register, and the counter is cleared to 0000_H 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 $FFFF_H$, and the counter is cleared to 0000_H.

(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 0000_H to $FFFF_H$.

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	• Match between counter and setting value of TTnCCR0 register • Capture to TTnCCR0 register due to TITn0 pin input
INTTTnCC1	• Match between counter and setting value of TTnCCR1 register • Capture to TTnCCR1 register due to TITn1 pin input
INTTTnOV	Overflow and underflow occurrence
INTTTnEC[Note]	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.

11.5.2 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.

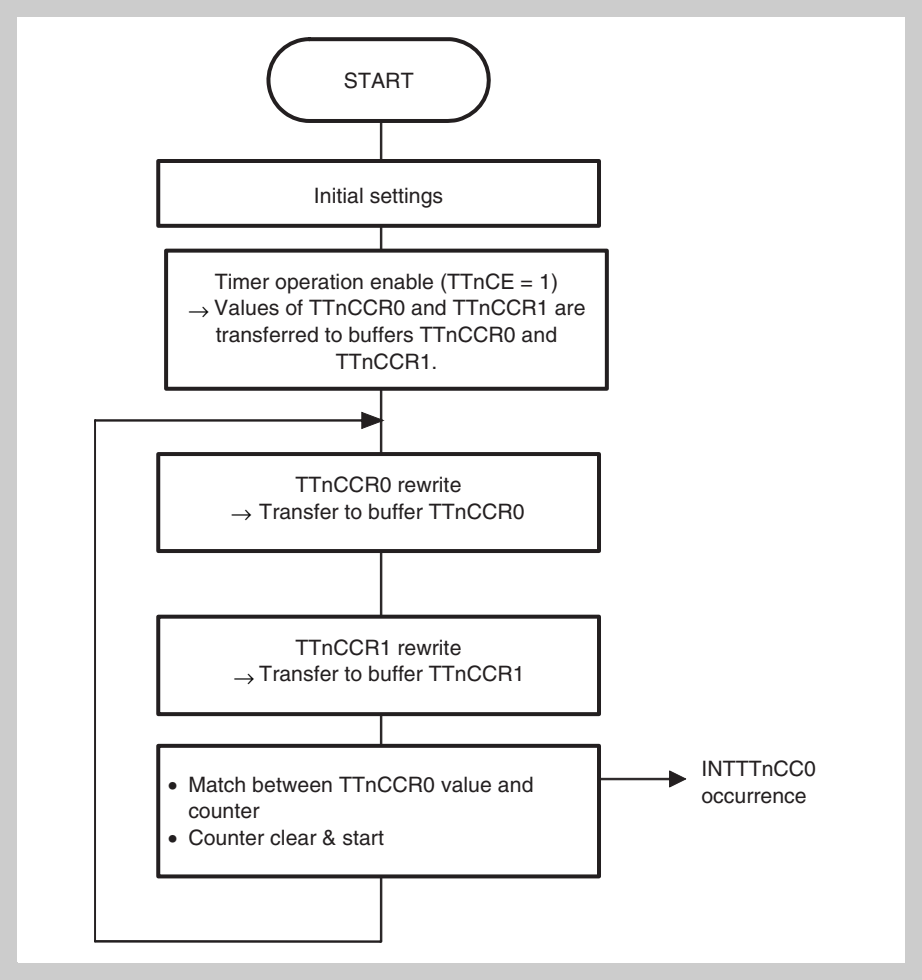


Figure 11-2 Basic operation flow for anytime rewrite

Note The interval mode is used as an example.

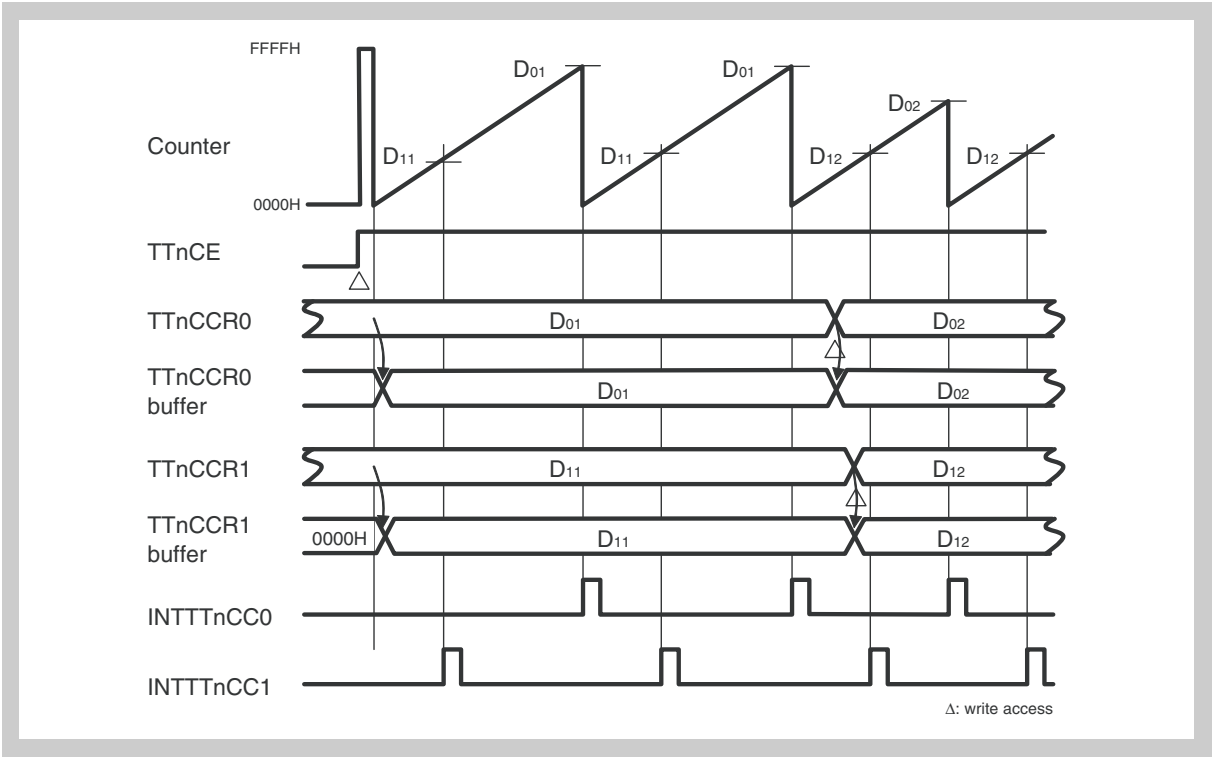


Figure 11-3 Basic anytime rewrite operation timing

Note 1. D_{01}, D_{02}: Setting values of TTnCCR0 register (0000_H to $FFFF_H$)
D_{11}, D_{12}: Setting values of TTnCCR1 register (0000_H to $FFFF_H$)

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 0000_H 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.

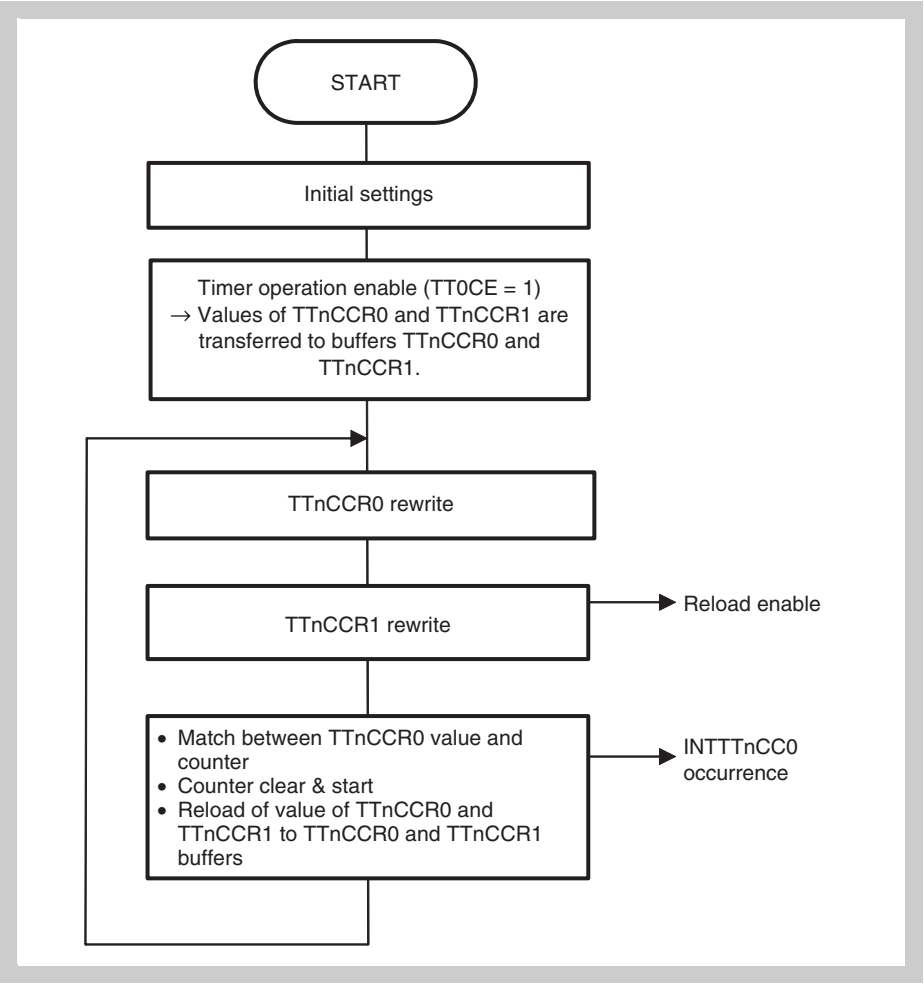


Figure 11-4 Basic operation flow for reload (batch rewrite)

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.

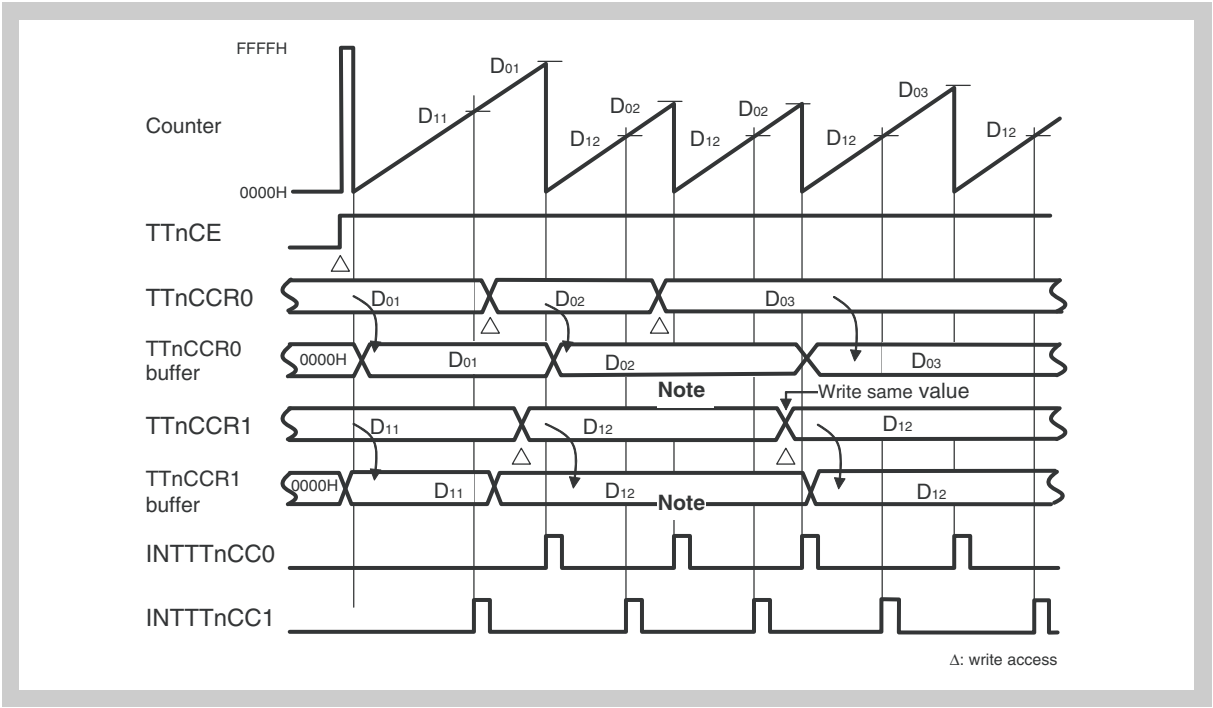


Figure 11-5 Basic reload operation timing

Note 1. Since the TTnCCR1 register is not written to, reloading is not performed even if TTnCCR0 is rewritten.

2. D_{01}, D_{02}, D_{03}: Setting values of TTnCCR0 register (0000_H to $FFFF_H$)
   D_{11}, D_{12}: Setting values of TTnCCR1 register (0000_H to $FFFF_H$)

3. The PWM mode is used as an example.

Table 11-17 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)

11.6 Operation in Each Mode

11.6.1 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.

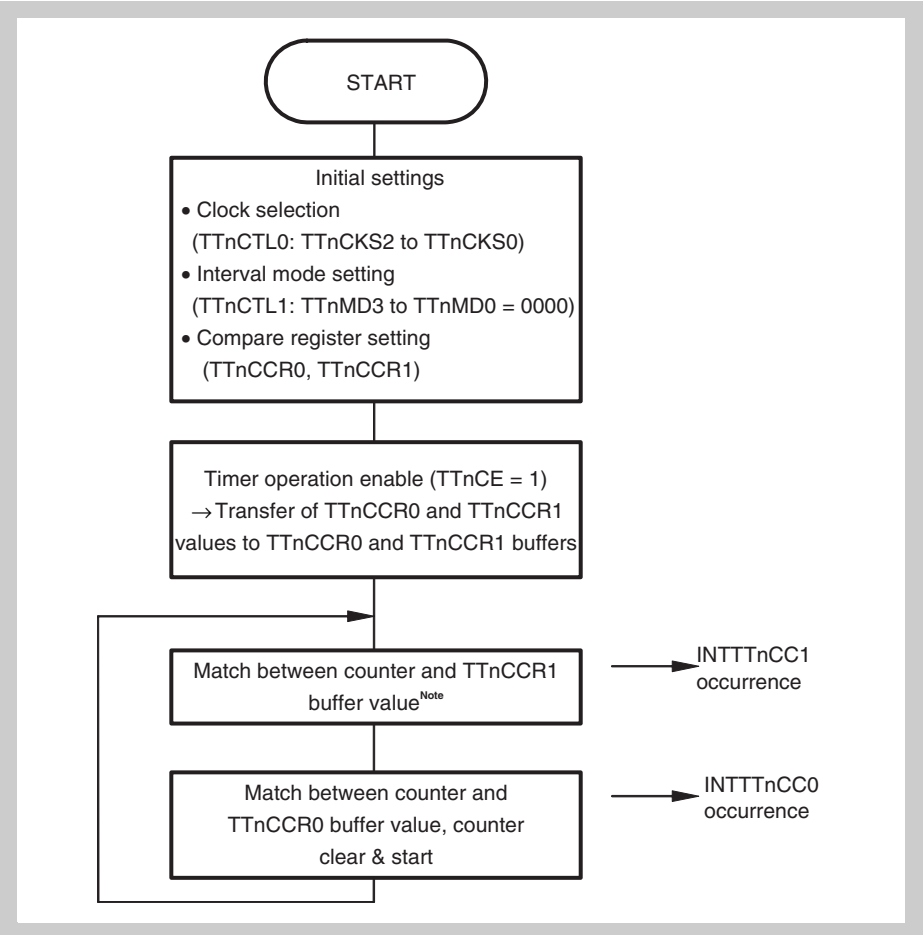


Figure 11-6 Basic operation flow in interval timer mode

Note In the case of a match between the counter and TTnCCR1 register, the counter is not cleared.

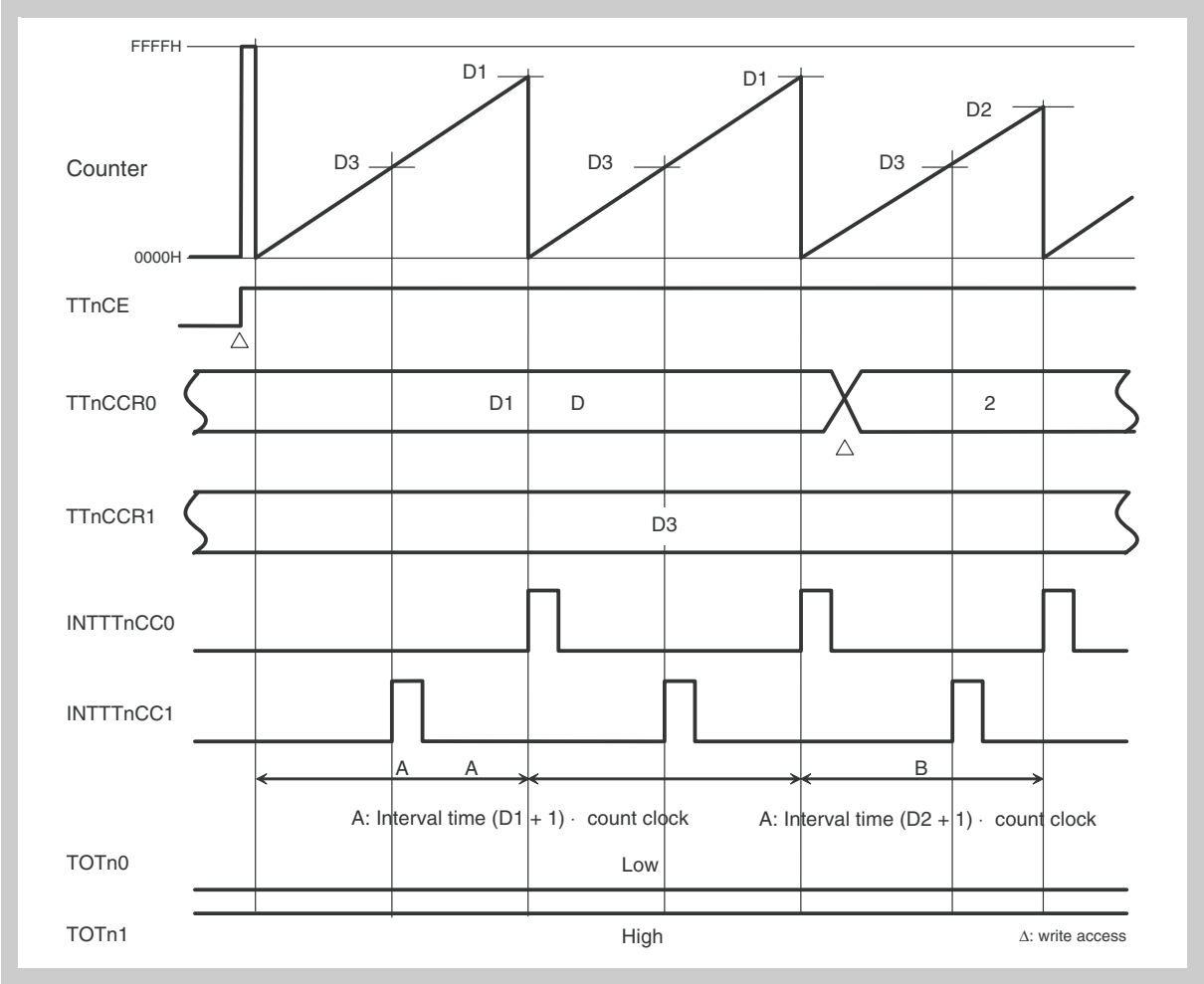


Figure 11-7 Basic timing in interval timer mode (1/2)
When D1 > D2 > D3, only value of TTnCCR0 register is rewritten, TOTn0 and TOTn1 are not output (TTnOE0, 1 = 0, TTnOL0 = 0, TTnOL1 = 1)

Note
1. D1, D2: Setting values of TTnCCR0 register (0000_H to $FFFF_H$)
   D3: Setting values of TTnCCR1 register (0000_H to $FFFF_H$)
2. Interval time = (Dm + 1) × (count clock cycle)
3. m = 1 to 3

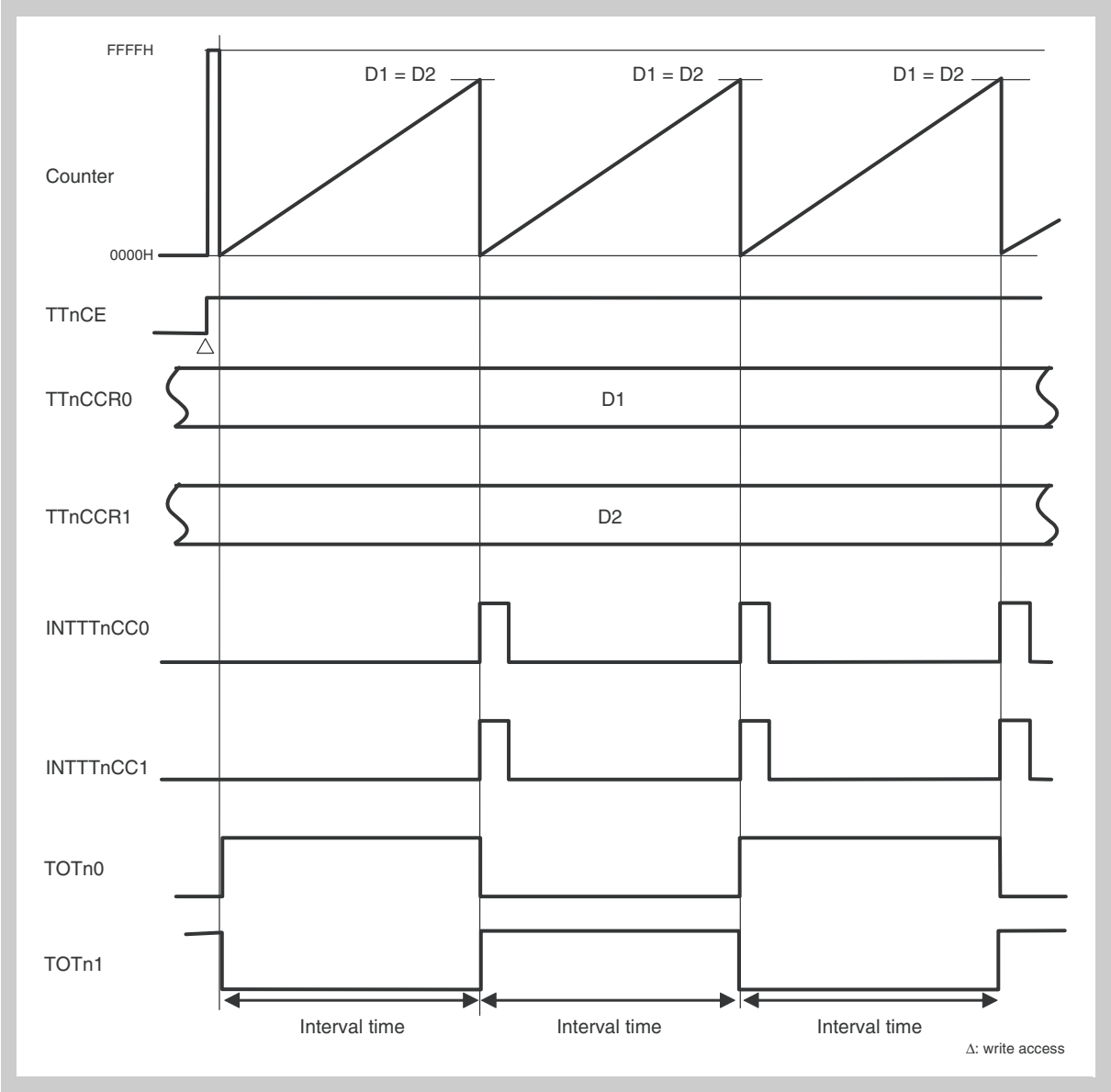


Figure 11-8 Basic timing in interval timer mode (2/2)
When D1 = D2, values of TTnCCR0 and TTnCCR1 registers not rewritten, TOTn1 output performed (TTnOE0, 1 = 1, TTnOL0 = 0, TTnOL1 = 1)

Note
1. D1: Setting value of TTnCCR0 register (0000_H to $FFFF_H$)
   D2: Setting value of TTnCCR1 register (0000_H to $FFFF_H$)
2. Interval time = (Dm + 1) × (count clock cycle)
3. TOTn0, TOTn1 toggle time = (Dm + 1) × (count clock cycle)
4. m = 1, 2

11.6.2 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 TTnCTL1.TTnEEE bit.)

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 $FFFF_H$.

[External event count operation flow]

<1> TTnCTL1.TTnMD[3:0] = 0001_B (mode setting)
Edge detection set with TTnIOC2.TTnEES[1:0] (TTnEES[1:0] = setting other than 01_B)

<2> TTnCTL0.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 set the TTnCCR0 register to 0000_H.
3. When TTnCCR1 register is set to 0000_H in external event count mode, the corresponding interrupt (INTTTnCC1) does not occur immediately after start, but after the first overflow of the timer ($FFFF_H$ to 0000_H).
4. TOTn0 pin output cannot be used during external event count mode. Alternatively use the interval timer mode (TTnCTL1.TTnMD[3:0] = 0000_B) and set bit TTnEEE = 1 in conjunction with TOTn0 pin output.

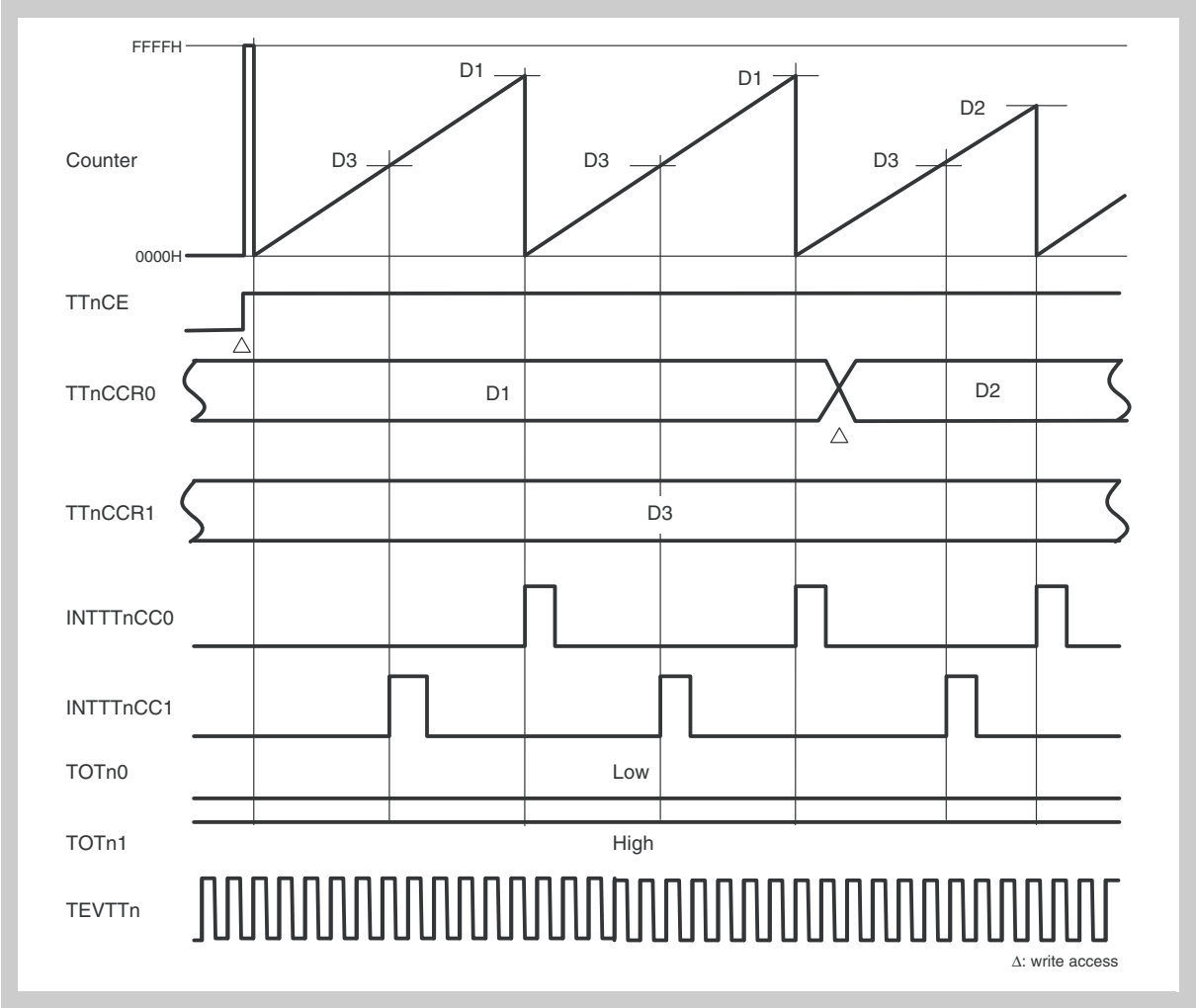


Figure 11-9 Basic operation timing in external event count mode (1/3)
When D1>D2>D3, only value of TTnCCR0 register is rewritten, TOTn0 and TOTn1 are not output. The signal input from TEVTTn and internally synchronized is counted as the count clock (TTnOE0, 1 = 0, TTnOL0 = 0, TTnOL1 = 1)

Note
1. D1, D2: Setting values of TTnCCR0 register (0001_H to $FFFF_H$)
   D3: Setting value of TTnCCR1 register (0000_H to $FFFF_H$)
2. Number of event counts = (Dm + 1) (m = 1, 2)

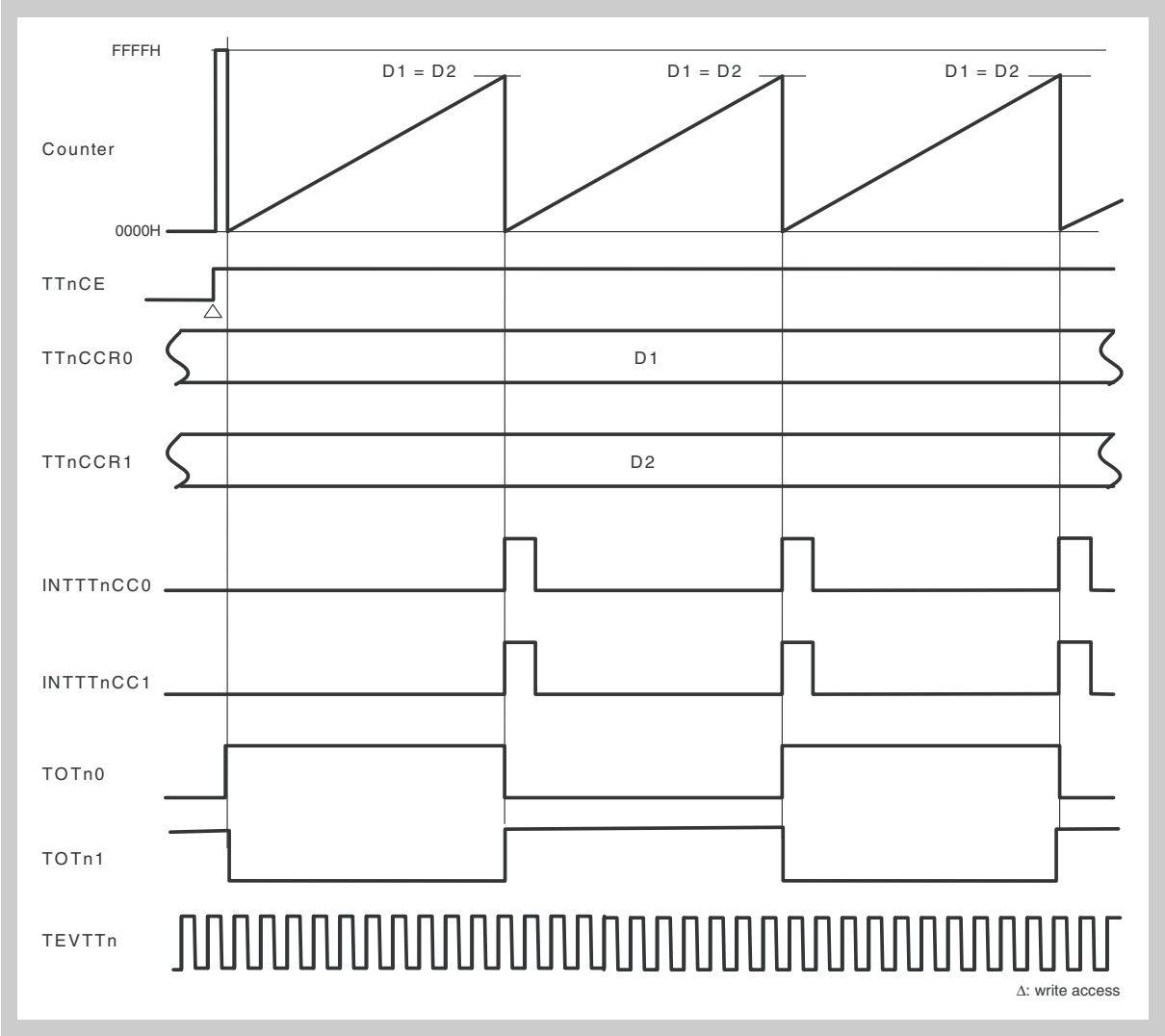


Figure 11-10 Operation timing in external event count mode (2/3)
When D1 = D2, TTnCCR0 and TTnCCR1 register values are not rewritten, TOTn0 and TOTn1 are output (TTnOE0, 1 = 1, TTnOL0 = 0, TTnOL1 = 1)

Note 1. D1: Setting value of TTnCCR0 register (0001_H to $FFFF_H$)
D2: Setting value of TTnCCR1 register (0000_H to $FFFF_H$)

2. Number of event counts = (Dm + 1) (m = 1, 2)

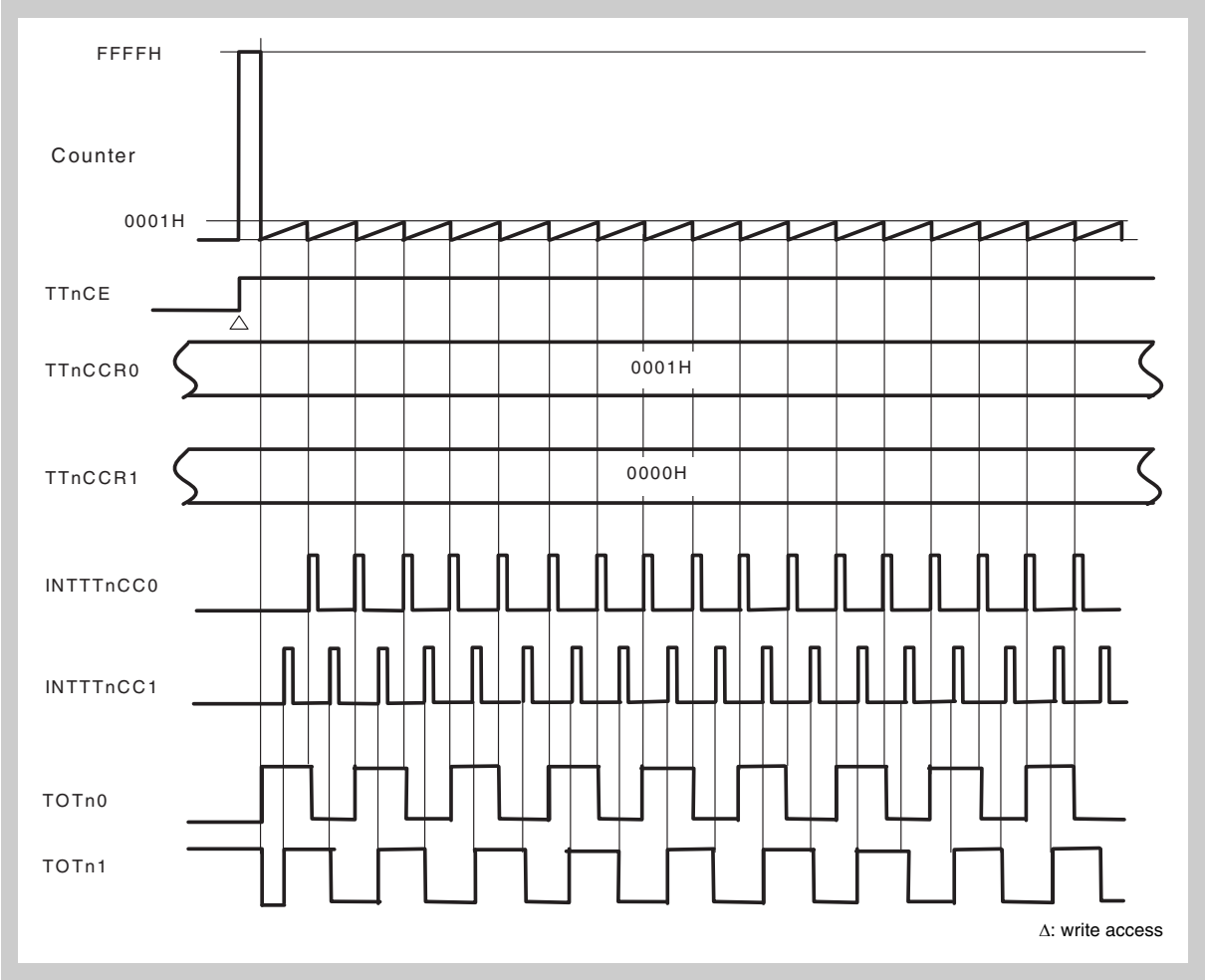


Figure 11-11 Basic operation timing in external event count mode (3/3)
When D1 = D2, TTnCCR0, TTnCCR1 register values are not rewritten,
TOTn0 and TOTn1 are output (TTnOE0, 1 = 1, TTnOL0 = 0, TTnOL1 = 1)

Note 1. D1: Setting value of TTnCCR0 register (0001_H)
D2: Setting value of TTnCCR1 register (0000_H)

2. Number of event counts = (Dm + 1) (m = 1, 2)

11.6.3 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 $FFFF_H$. Upon detection of the valid edge of external trigger input (TTRGTn pin), or when the TTnCTL1.TTnEST bit 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 TTnCTL1.TTnEEE bit to 0.

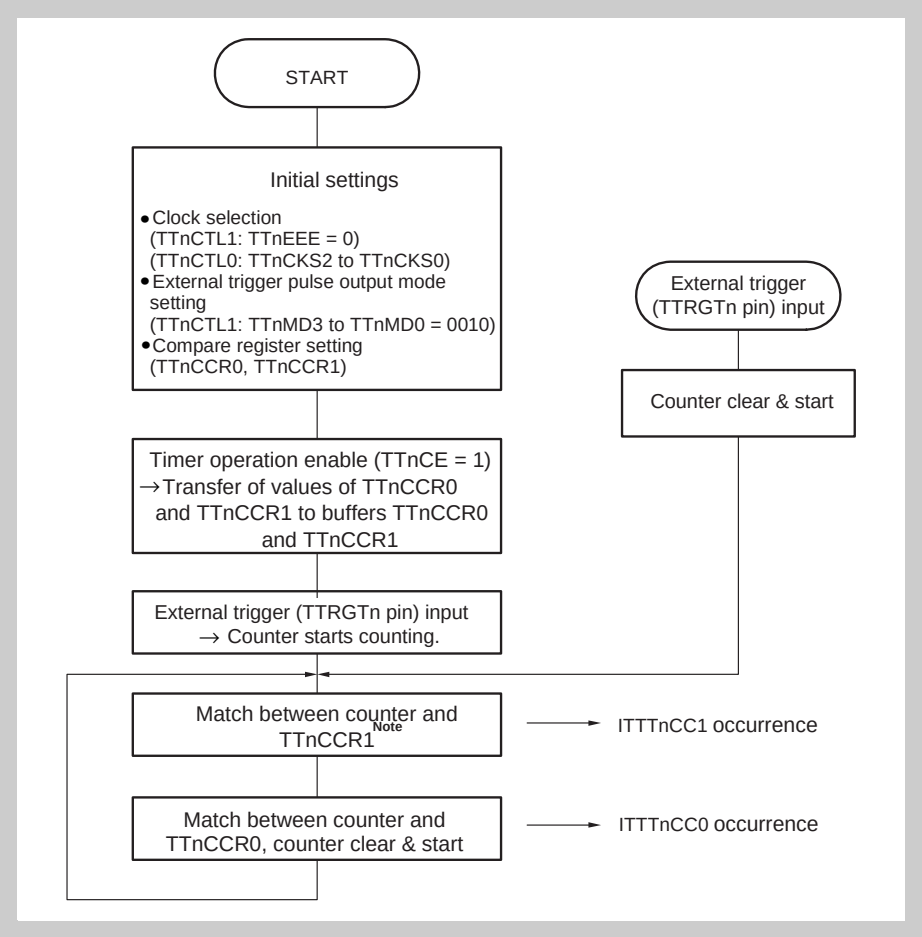


Figure 11-12 Basic operation flow in external trigger pulse output mode

Note The counter is not cleared upon a match between the counter and the TTnCCR1 buffer register.

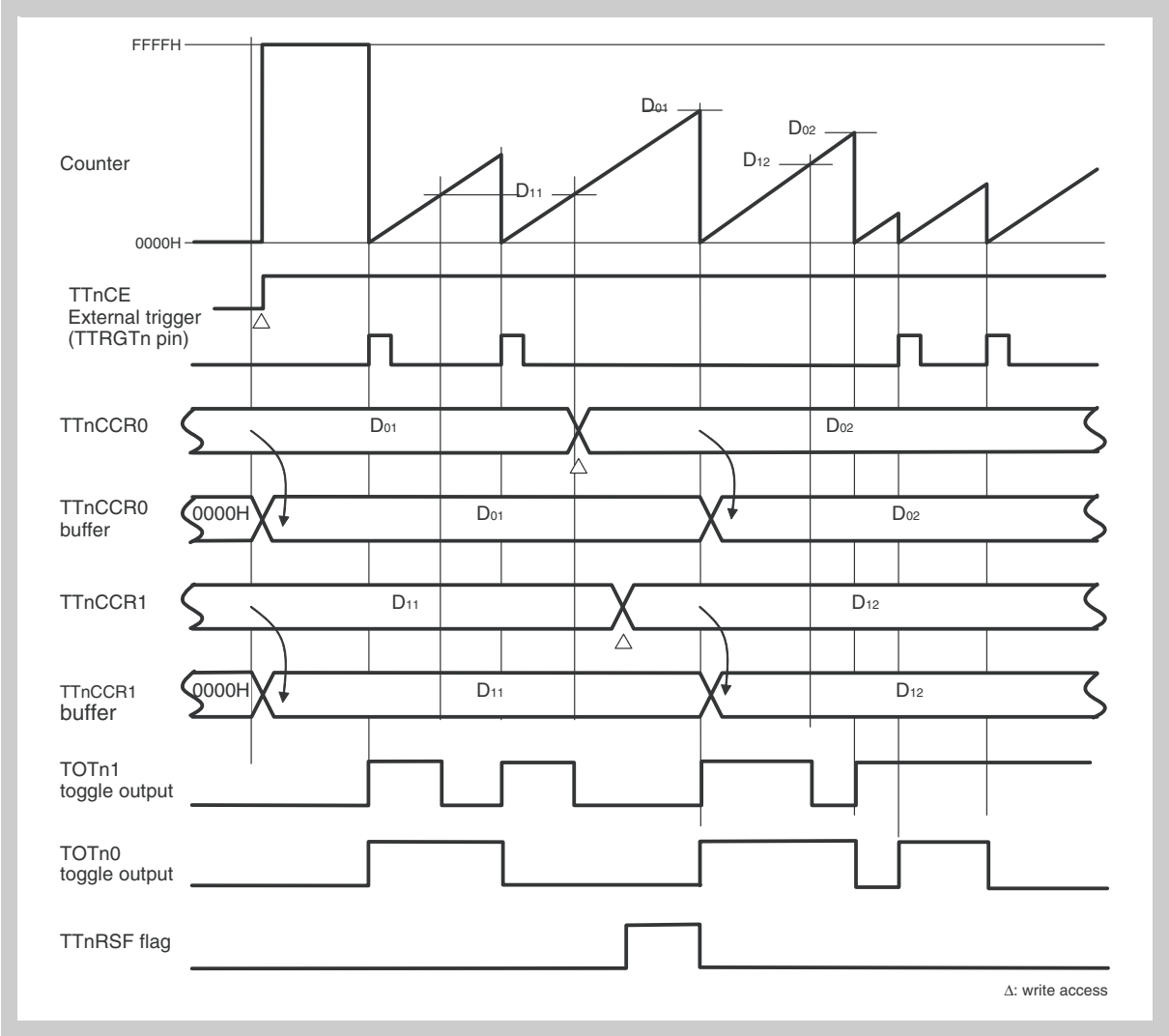


Figure 11-13 Basic operation timing in external trigger pulse output mode When values of TTnCCR0 and TTnCCR1 registers are rewritten, TOTn0 and TOTn1 are output (TTnOE0, 1 = 1, TTnOL0, 1 = 0)

Note

1. D_{01}, D_{02}: Setting values of TTnCCR0 register (0000_H to $FFFF_H$)
   D_{11}, D_{12}: Setting values of TTnCCR1 register (0000_H to $FFFF_H$)
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. Pin TOTn0 is toggled when the counter is cleared immediately following count start.

11.6.4 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 TTnCTL0.TTnCE bit is set to 1, external trigger input (TTRGTn pin) wait results, with the counter remaining stopped at $FFFF_H$. Upon detection of the valid edge of external trigger input (TTRGTn pin), or when TTnCTL0.TTnEST bit is set to 1, count up starts. The TOTn1 pin becomes high level upon a match between the counter and TTnCCR1 register and a compare match interrupt (INTTTnCC1) is output.
Moreover, upon a match between the counter and TTnCCR0 register, the TOTn1 pin becomes low level, and the counter is cleared to 0000_H and then stops. The TOTn0 pin performs toggle output during the count operation upon a match between the counter and the TTnCCR0 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.

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 0000_H.

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.TTnMD[3:0] = 0011_B (One-shot pulse mode)

<2> TTnCCR0 register setting (duty setting),
TTnIOC0.TTnOE1 = 1 (TOTn1 pin output enable)

<3> TTnCTL0.TTnCE = 1 (counter operation enable):
TOTn1 = Low-level output

<4> TTnCTL1.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
Counter cleared

<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 TTnCTL1.TTnEEE bit to 0.

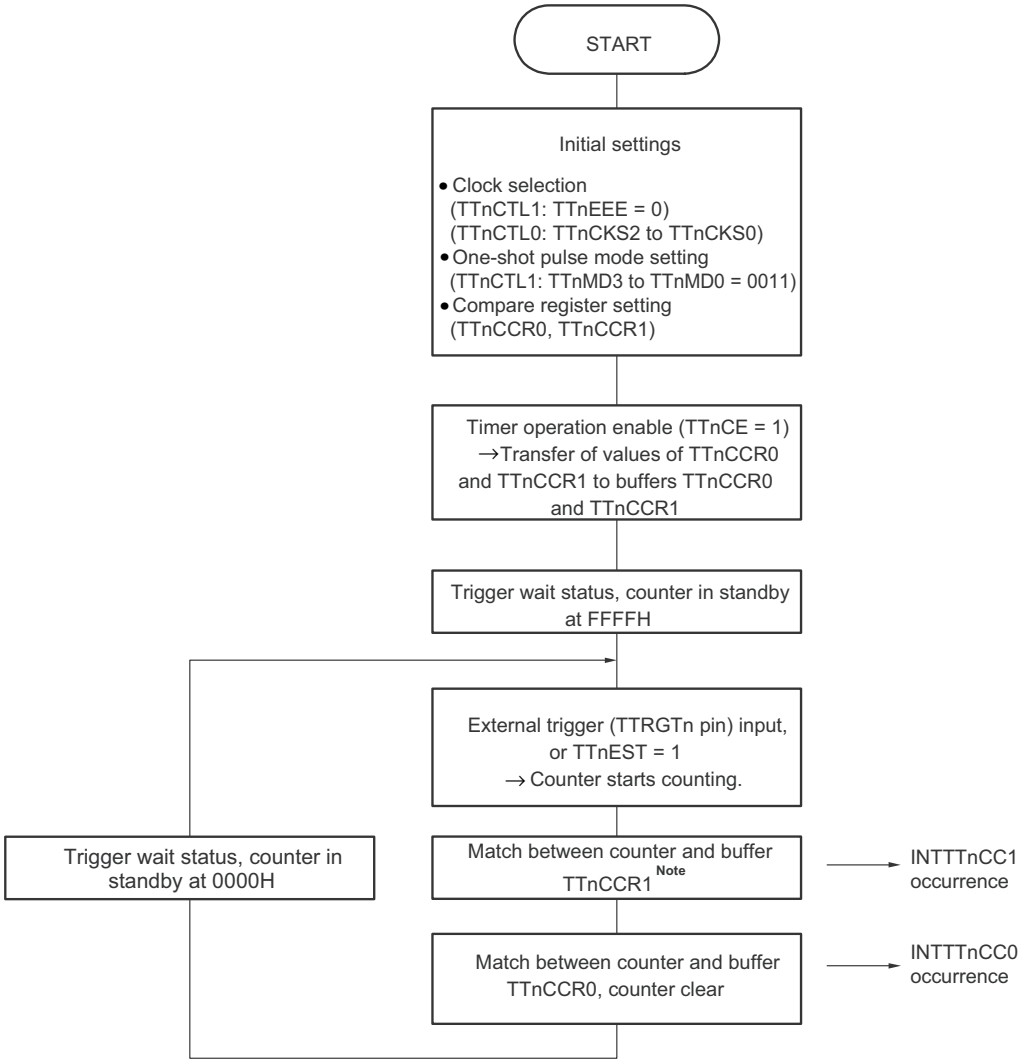


Figure 11-14 Basic operation flow in one-shot pulse mode

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.

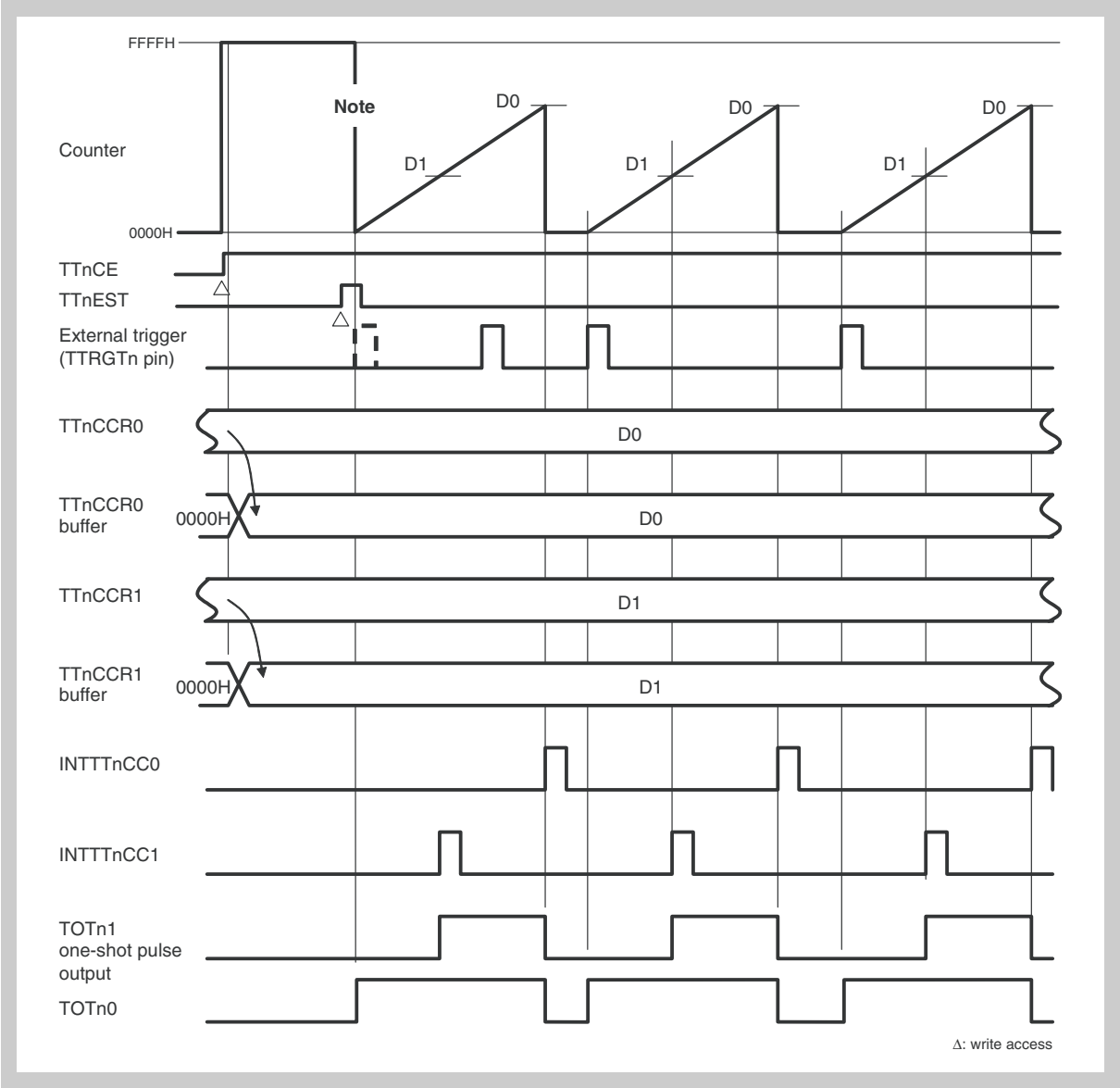


Figure 11-15 Basic operation timing in one-shot pulse mode (TTnOE0, 1 = 1, TTnOL0, 1 = 0)

Note 1. Count up starts when the value of TTnEST becomes 1 or TTRGTn is input.

2. D0: Setting value of TTnCCR0 register (0000_H to $FFFF_H$)
   D1: Setting value of TTnCCR1 register (0000_H to $FFFF_H$)

3. TOTn1 (output delay) = (setting value of TTnCCR1 register) × (count clock cycle) TOTn1 (output pulse width) = {(setting value of TTnCCR0 register +1) - (setting value of TTnCCR1 register)} × (count clock cycle)

11.6.5 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 high level, and upon a match between the counter and the TTnCCR1 register, becomes low level. Next, the TOTn1 pin becomes high 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.

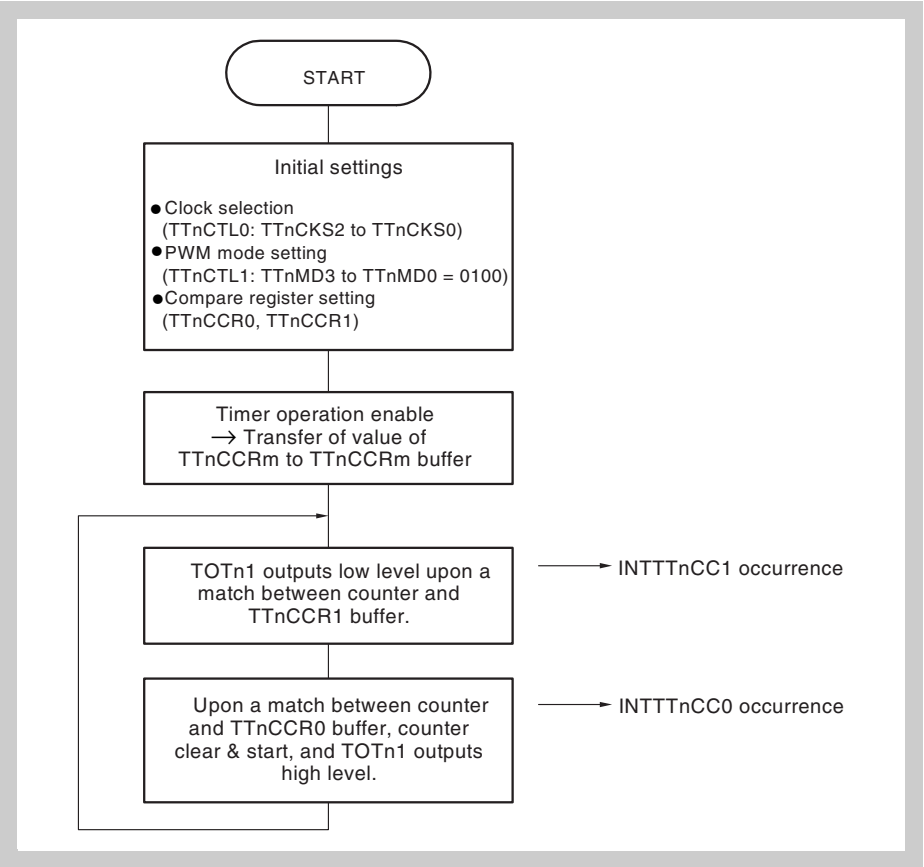


Figure 11-16 Basic operation mode in PWM mode (1/2)
when values of TTnCCR0 and TTnCCR1 registers are not rewritten during timer operation

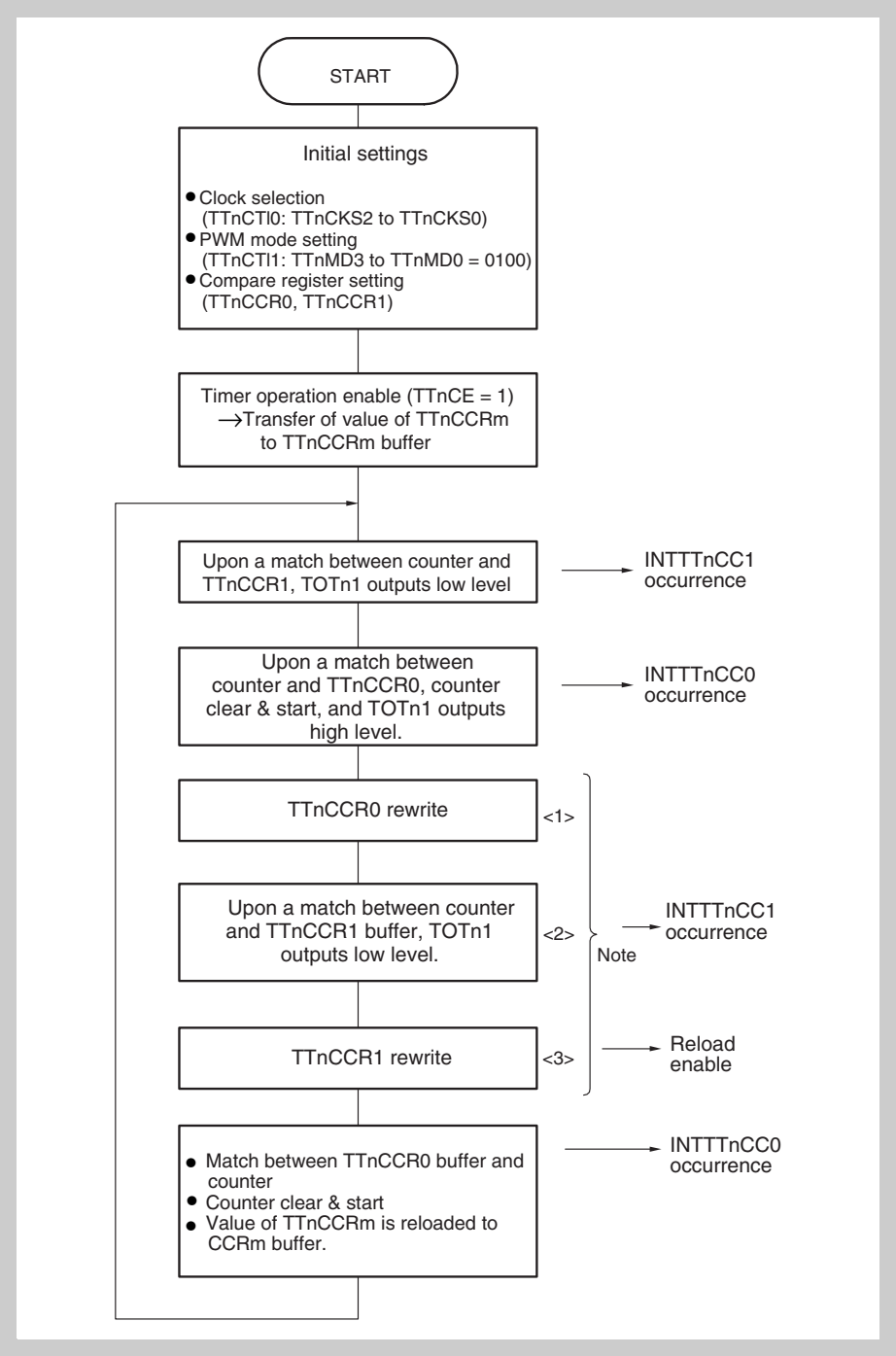


Figure 11-17 Basic operation flow in PWM mode (2/2) when values of TTnCCR0 and TTnCCR1 registers are rewritten during timer operation

Note 1. 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.

2. m = 0, 1

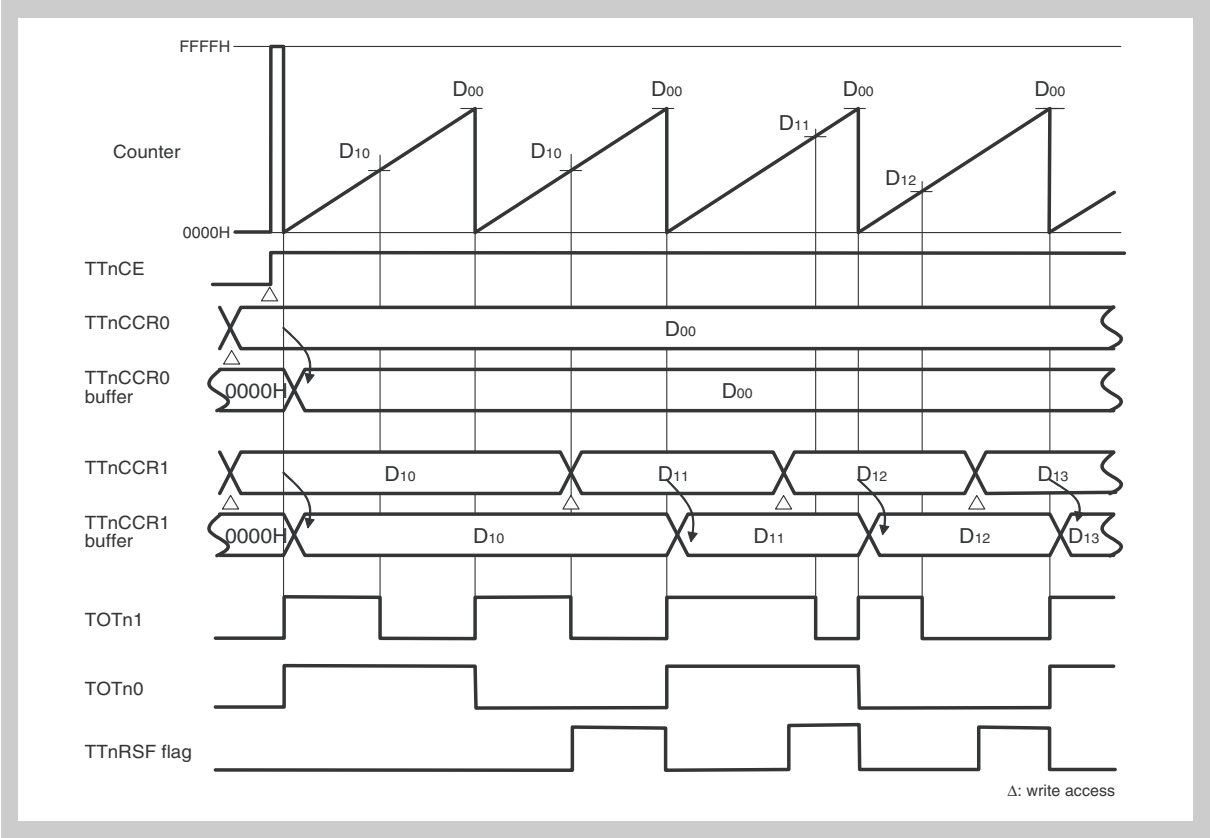


Figure 11-18 Basic operation timing in PWM mode (1/2) when only value of TTnCCR1 is rewritten, and TOTn0 and TOTn1 are output (TTnOE0, 1 = 1, TTnOL0, 1 = 0)

Note

1. D_{00}: Setting value of TTnCCR0 register (0000_H to $FFFF_H$)
   D_{10}, D_{11}, D_{12}, D_{13}: Setting values of TTnCCR1 register (0000_H to $FFFF_H$)
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)

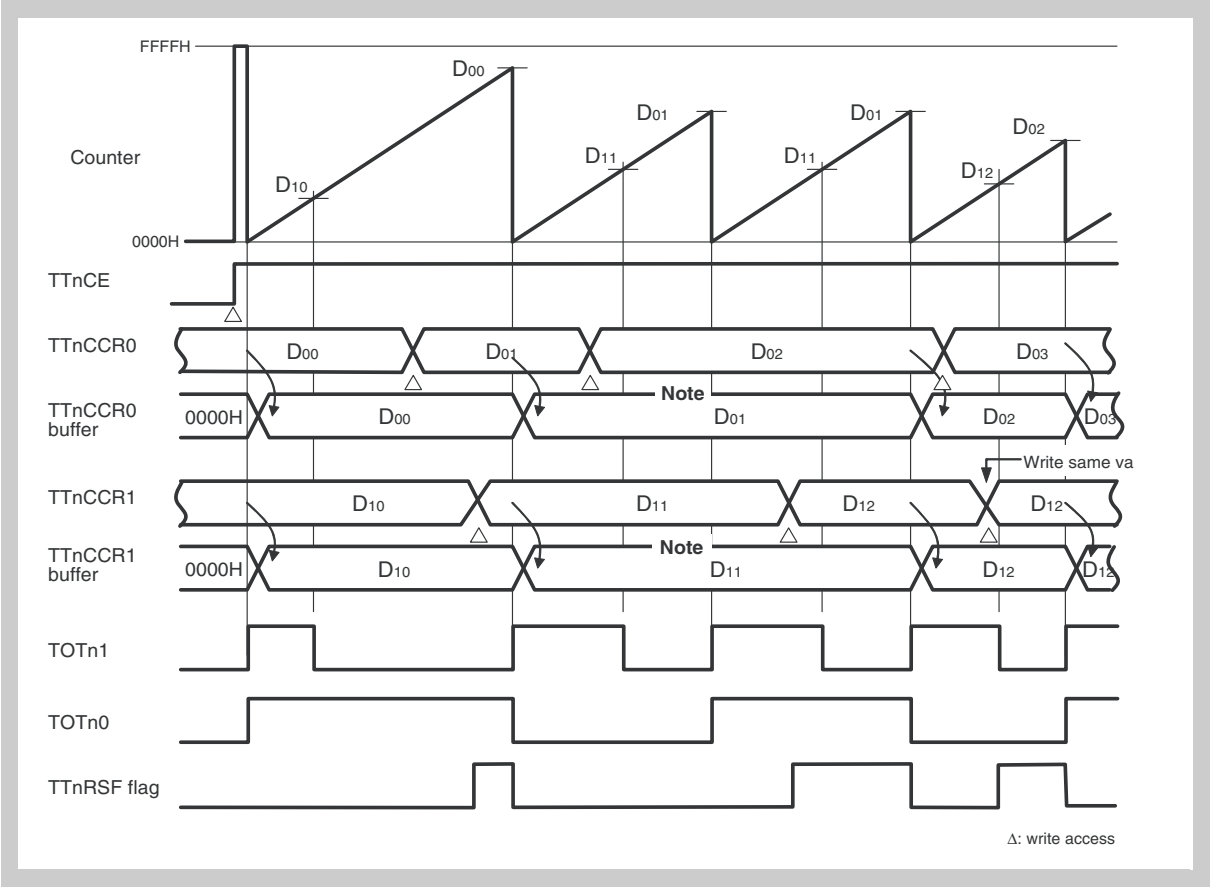


Figure 11-19 Basic operation timing in PWM mode (2/2) when values of TTnCCR0 and TTnCCR1 register are rewritten, TOTn0 and TOTn1 are output (TTnOE0, 1 = 1, TTnOL0, 1 = 0)

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_{00}, D_{01}, D_{02}, D_{03}: Setting values of TTnCCR0 register (0000_H to $FFFF_H$)
   D_{10}, D_{11}, D_{12}, D_{13}: Setting values of TTnCCR1 register (0000_H to $FFFF_H$)
3. The TOTn0 and TOTn1 pins become high level at timer count start.

11.6.6 Free-running mode

The operation flow of the free-running mode depends on the setting of TTnOPT0.TTnCCS[1:0] bits, and is shown below.

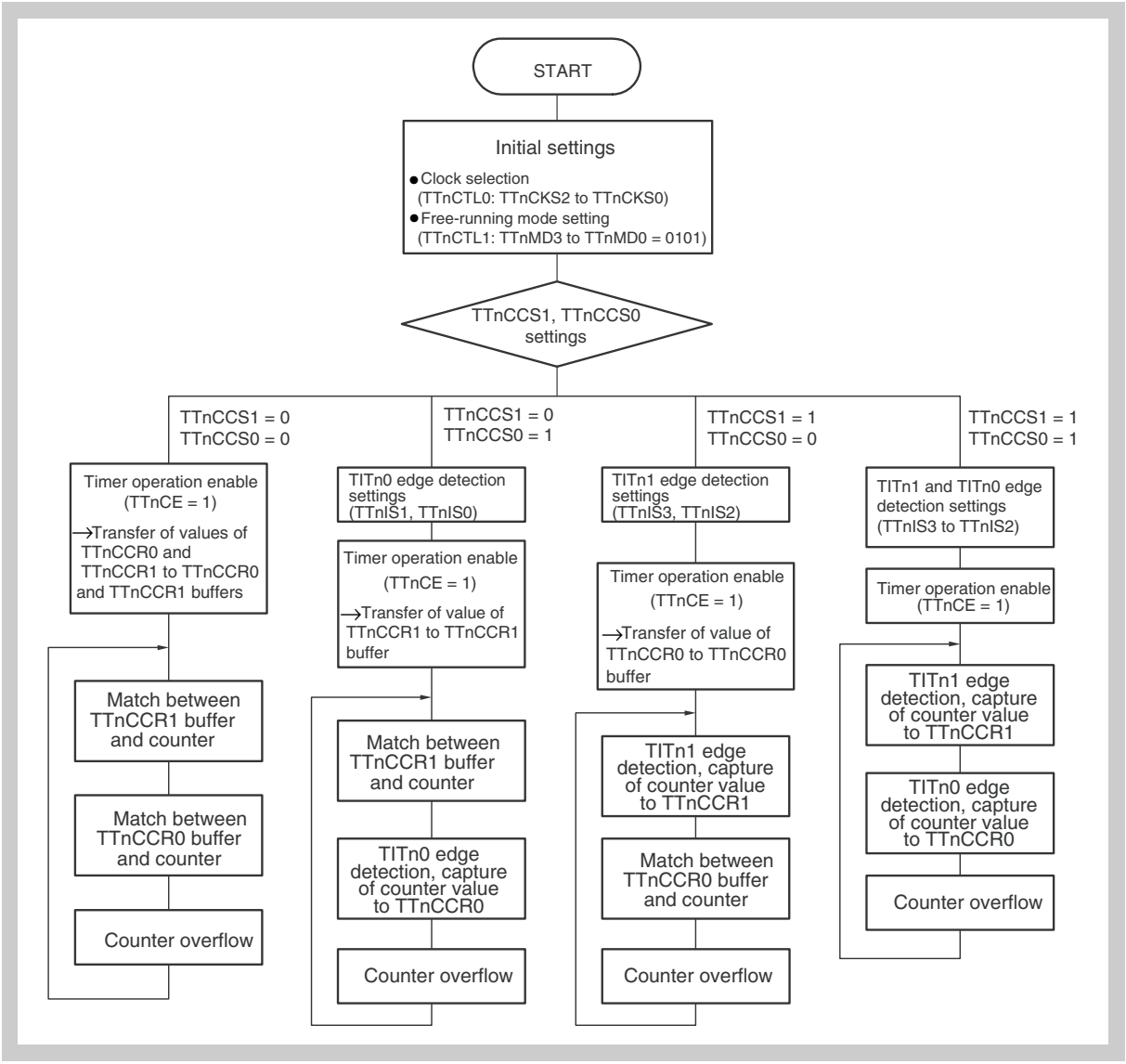


Figure 11-20 Basic operation flow in free-running mode

(1) Compare function (TTnCCS1 = 0, TTnCCS0 = 0)

When TTnCTL0.TTnCE is set to 1, the counter counts from 0000_H to $FFFF_H$. An overflow interrupt (INTTTnOV) is output when the counter value changes from $FFFF_H$ to 0000_H, 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 TTnIOC0 register are set to 1.

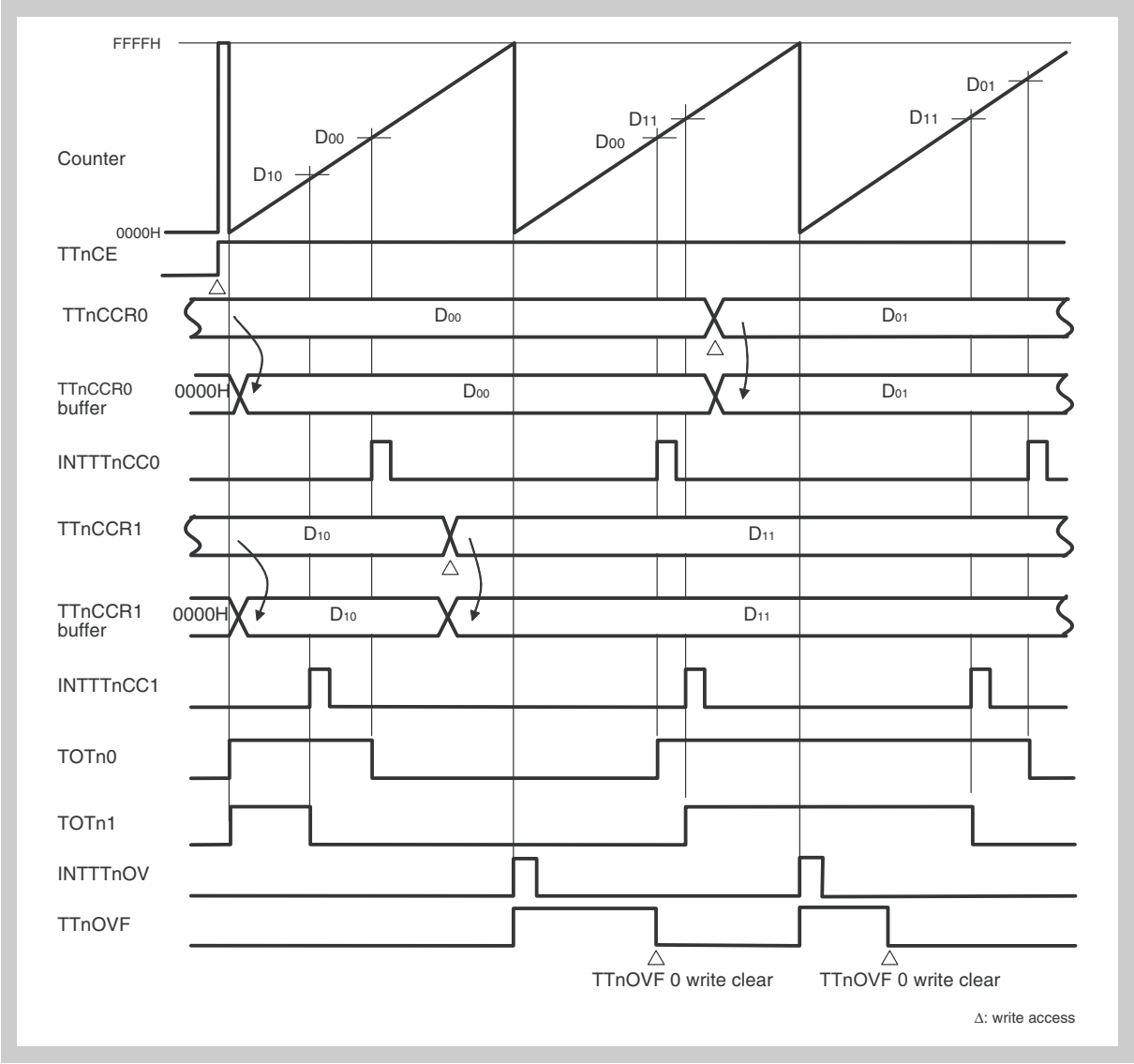


Figure 11-21 Basic operation timing in free-running mode (compare function) when values of TTnCCR0 and TTnCCR1 registers are rewritten, TOTn0, TOTn1 are output (TTnOE0, 1 = 1, TTnOL0, 1 = 0)

Note

1. D_{00}, D_{01}: Setting values of TTnCCR0 register (0000_H to $FFFF_H$)
   D_{10}, D_{11}: Setting values of TTnCCR1 register (0000_H to $FFFF_H$)
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.TTnCE is set to 1, the counter counts from 0000_H to $FFFF_H$. An overflow interrupt (INTTTnOV) is output when the value of the counter changes from $FFFF_H$ to 0000_H, 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 ($FFFF_H$), 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).

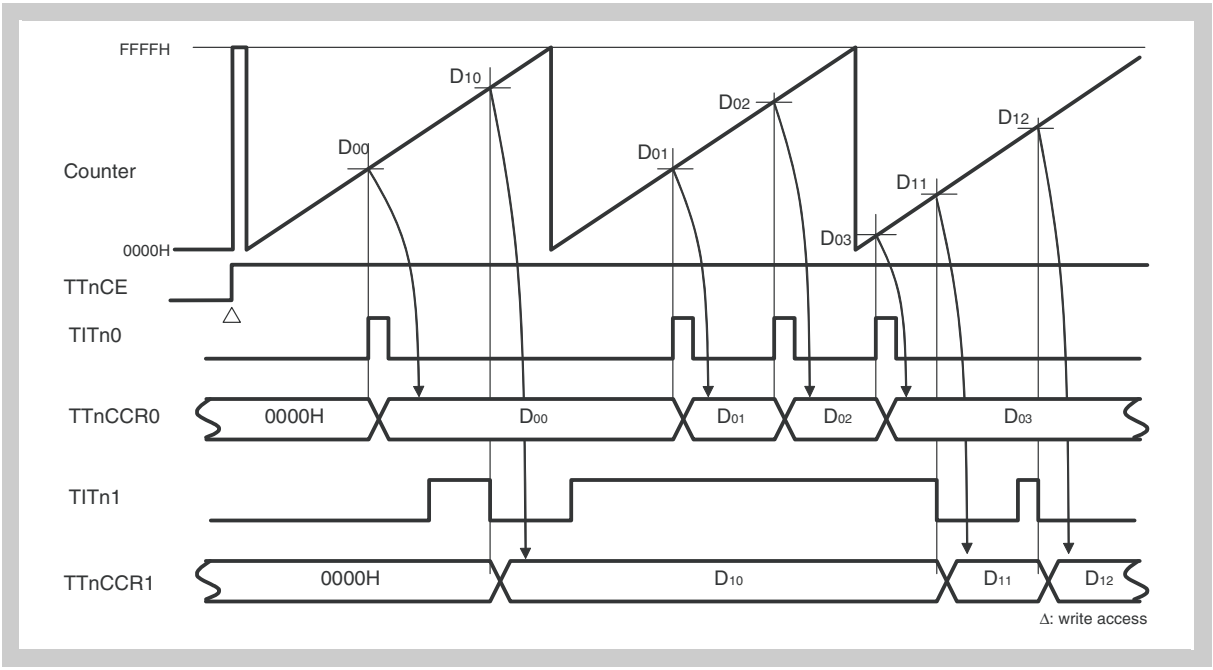


Figure 11-22 Basic operation timing in free-running mode (capture function) when TOTn0, TOTn1 are not output (TTnOE0, 1 = 0, TTnOL0, 1 = 0)

Note 1. D_{00}, D_{01}: Values captured to TTnCCR0 register (0000_H to $FFFF_H$)
D_{10}, D_{11}: Values captured to TTnCCR1 register (0000_H to $FFFF_H$)

2. TITn0: Setting to rising edge detection (TTnIOC1.TTnIS[1:0] = 01_B)
TITn1: Setting to falling edge detection (TTnIOC1.TTnIS[3:2] = 10_B)

(3) Compare/capture function (TTnCCS1 = 0, TTnCCS0 = 1)

When TTnCTL0.TTnCE bit is set to 1, the counter counts from 0000_H to $FFFF_H$, an overflow interrupt (INTTTnOV) is output when the value of the counter changes from $FFFF_H$ to 0000_H, 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.TTnOE0 bit is set to 1.

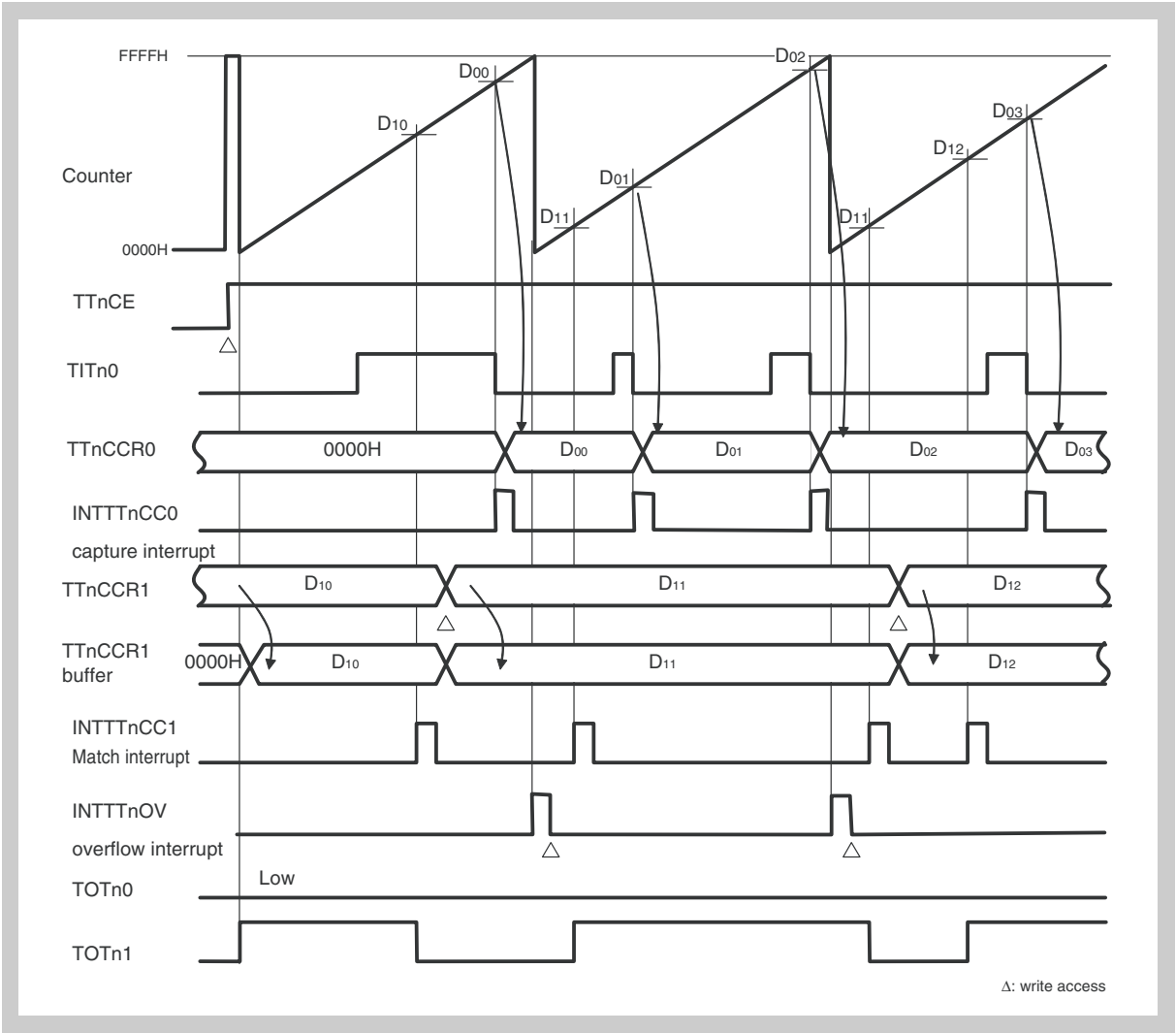


Figure 11-23 Basic operation timing in free-running mode (compare/capture function) when value of TTnCCR1 is rewritten, TOTn0, TOTn1 are output (TTnOE0, 1 = 1, TTnOL0, 1 = 0)

Note 1. D_{00}, D_{01}: Setting values of TTnCCR1 register (0000_H to $FFFF_H$)
D_{10}, D_{11}, D_{12}, D_{13}, D_{14}, D_{15}: Values captured to TTnCCR0 register (0000_H to $FFFF_H$)

2. TITn0: Setting to rising edge detection (TTnIOC1.TTnIS[1:0] = 11_B)

(4) Overflow flag

When, in the free-running mode, the counter overflows from $FFFF_H$ to 0000_H, the overflow flag (TTnOVF) is set to "1", and an overflow interrupt (INTTTnOV) is output.

The overflow flag is cleared through 0 write by the CPU. It is not just cleared by a read access.

11.6.7 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 0000_H. 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.

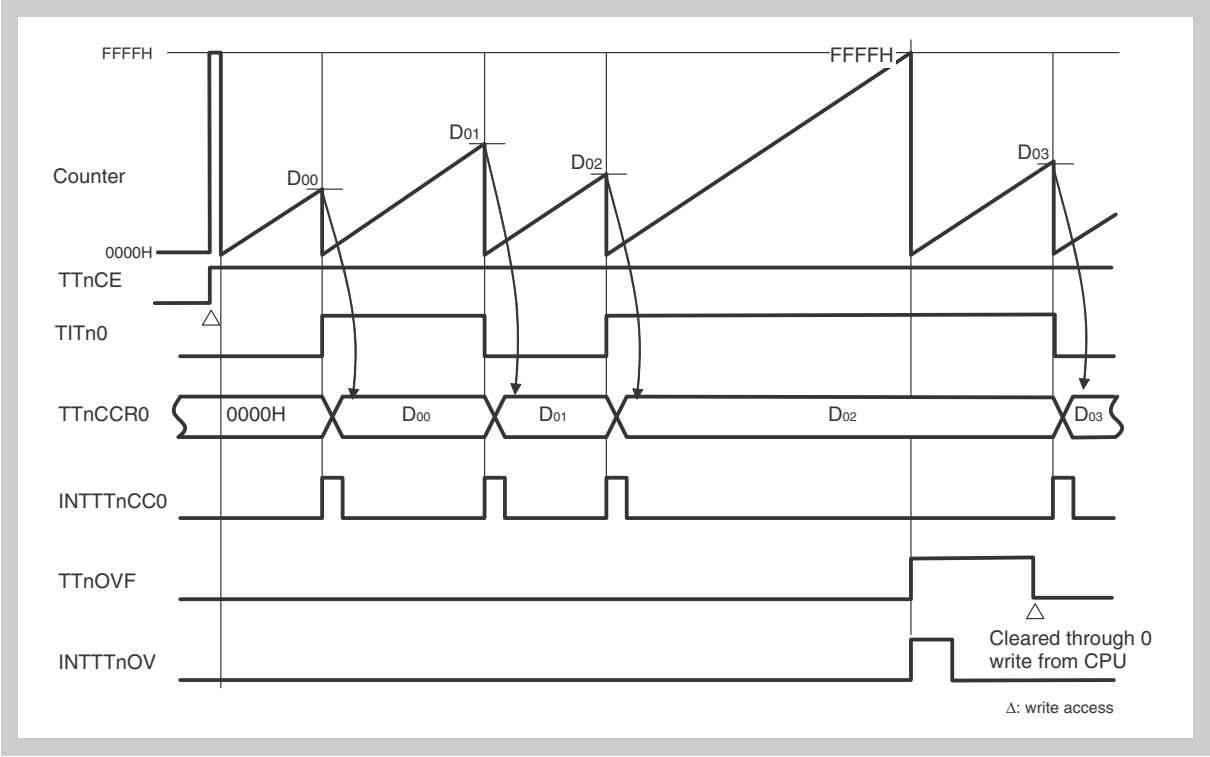


Figure 11-24 Basic operation timing in pulse width measurement mode (TTnOE0, 1 = 0, TTnOL0, 1 = 0)

Note 1. D_{00}, D_{01}, D_{02}, D_{03}: Values captured to TTnCCR0 register (0000_H to $FFFF_H$)

2. TITn0: Setting to rising edge/falling edge (both edges) detection ($TTnIOC1.TTnIS[1:0] = 11_B$)

11.6.8 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 $0 \leq TTnCCR0 \leq FFFE_H$.

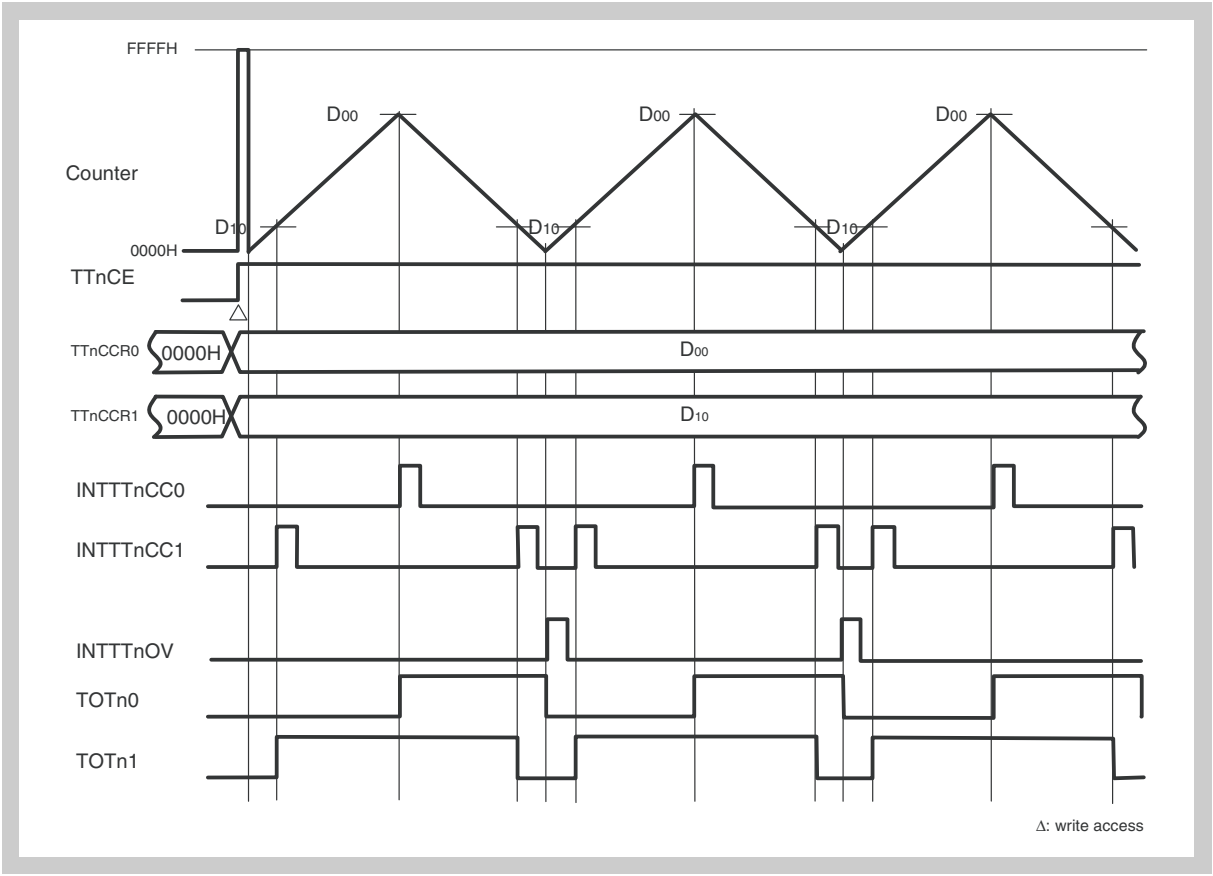


Figure 11-25 Basic operation timing in triangular wave PWM mode

When TOTn0, TOTn1 are output (TTnOE0, 1 = 1, TTnOL0, 1 = 0).

11.6.9 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 by the TTnIOC0.TTnOL1 bit.
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 0000_H is performed under the following conditions in each mode.

Table 11-18 Counter clear operation

Clear condition	Encoder compare	Encoder capture	Encoder capture compare
Method whereby counter is cleared to 0000_H upon match with compare register (setting of TTnCTL2 .TTnECM[1:0])	√	-	√ (TTnECM0[Note])
Method whereby counter is cleared to 0000_H upon detection of edge of pin TECR0 (setting of bits TTnECS[1:0] when TTnIOC3.TTnSCE = 0)	√	√	√
Method whereby counter is cleared to 0000_H by special clear function of encoder (setting of bits TTnZCL, TTnBCL, TTnACL when TTnIOC3.TTnSCE = 1)	√	√	√

Note Since the TTnCCR1 register is a capture-only register, the setting of TTnCTL2.TTnECM1 bit is 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 (TTnMD[3:0] = 1000_B, 1001_B, 1010_B) are as follows.

Table 11-19 Control through TTnCTL2 register

TTnMD[3:0]	TTnUDS[1:0]	TTnECM1	TTnECM0	TTnLDE	Clear	Load
1000_B	All settings possible 00_B 01_B 10_B 11_B	0	0	0	-	–
				1		√
			1	0	TTnCCR0	–
				1		•
		1	0	Invalid	TTnCCR1	–
			1	Invalid	TTnCCR0 TTnCCR1	–
1001_B		Invalid			–	–
1010_B		0	0	0	–	–
				1		√
			1	0	TTnCCR0	-
				1		•
		1	0	Invalid	–	–
			1	Invalid	TTnCCR0	–

- In the case of bits TTnUDS[1:0], up/down judgment control is performed for the phase input from pins TENCTn0 and TENCTn1.
- In the case of bits TTnECM[1:0], counter clear control is performed upon a match between the counter value and the compare setting value.

  Bits TTnECM[1:0] 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[1:0] bit setting is 00_B, 01_B, 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 0000_H to TTnCCR0 by using the counter load functions (TTnLDE = 1) indicated with "•" in *Table 11-19 "Control through TTnCTL2 register"*, and the function for clearing the counter to 0000_H 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 (TTnCTL2.TTnLDE bit)" on page 485*).

(a) Up/down count selection specification (TTnCTL2. TTnUDS[1:0] bits)

Counter up/down is judged according to the settings of bits TTnCTL2.TTnUDS[1:0], and the phases input from pins TENCTn0 and TENCTn1.

Bits TTnUDS[1:0] are valid only in the encoder compare mode, encoder capture mode, and encoder capture/compare mode.

1. TTnUDS[1:0] = 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 <Register Settings>

TTnIOC3.TTnEIS[3:2]	TENCTn1 pin input	Edge detection specification invalid
TTnIOC3.TTnEIS[1:0] = 10_B	TENCTn0 pin input	Rising edge detection

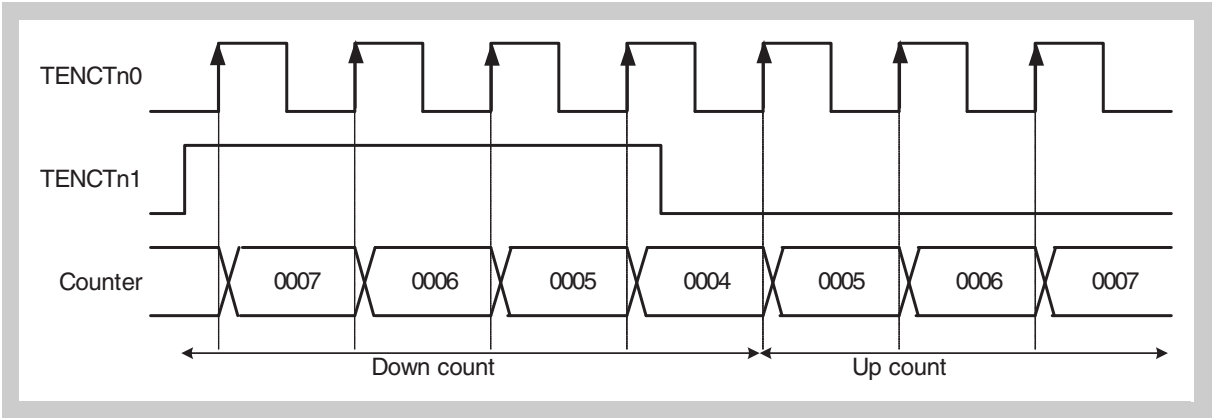


Figure 11-26 Encoder count function up/down count selection specification timing 1

Note Counting is performed when the edges of the TENCTn0/TENCTn1 pin inputs overlap.

2. TTnUDS[1:0] = 01_B (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 <Register Settings>

TTnIOC3.TTnEIS[3:2] = 10_B TENCTn1 pin input Rising edge detection

TTnIOC3.TTnEIS[1:0] = 10_B TENCTn0 pin input Rising edge detection

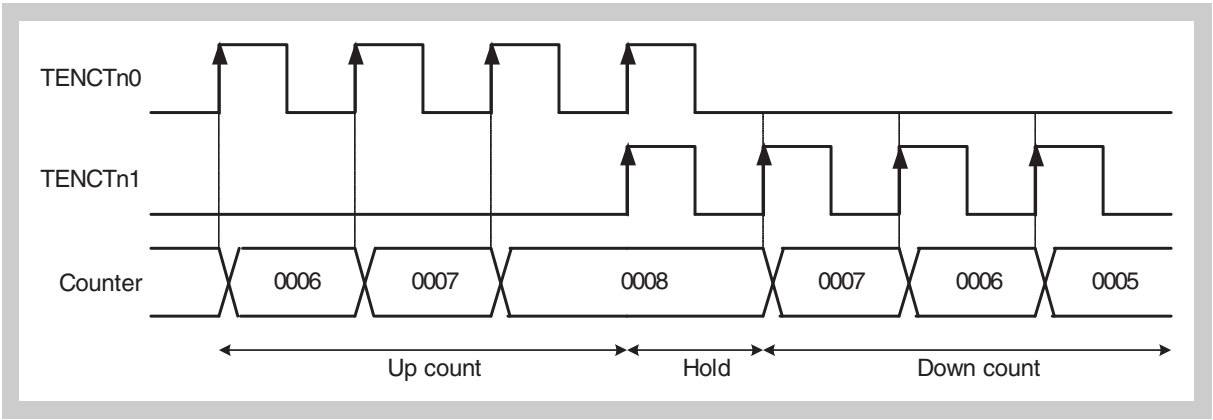


Figure 11-27 Encoder count function up/down count selection specification timing 2

Note The count value is held when the edges of the TENCTn0/TENCTn1 pin inputs overlap.

3. TTnUDS[1:0] = 10_B (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 <Register Setting>

TTnIOC3:TTnEIS[3:0]	Pins TENCTn1, TENCTn0	Edge detection specification invalid

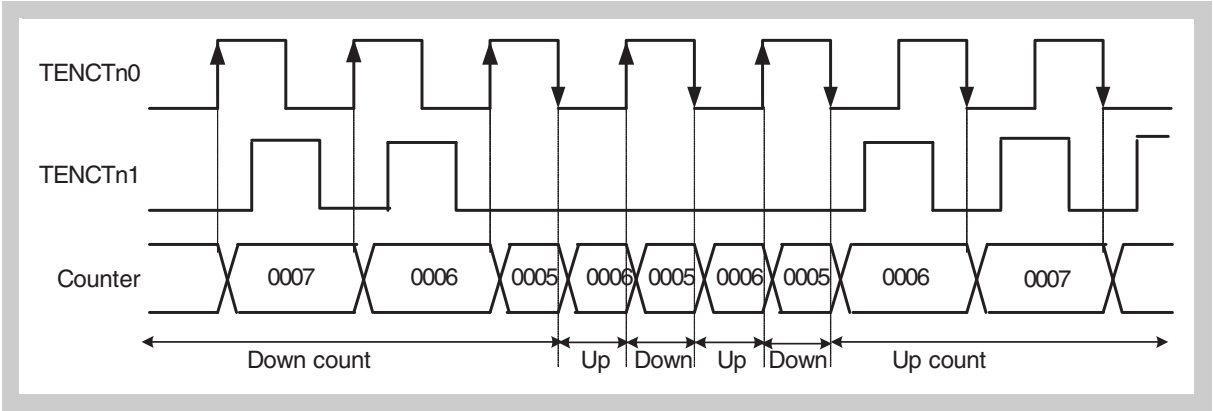


Figure 11-28 Encoder count function up/down count selection specification timing 3

4. TTnUDS[1:0] = 11_B (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

<Register Setting>

TTnIOC2.TTnEIS[3:0]	Pins TENCTn1, TENCTn0	Edge detection specification invalid

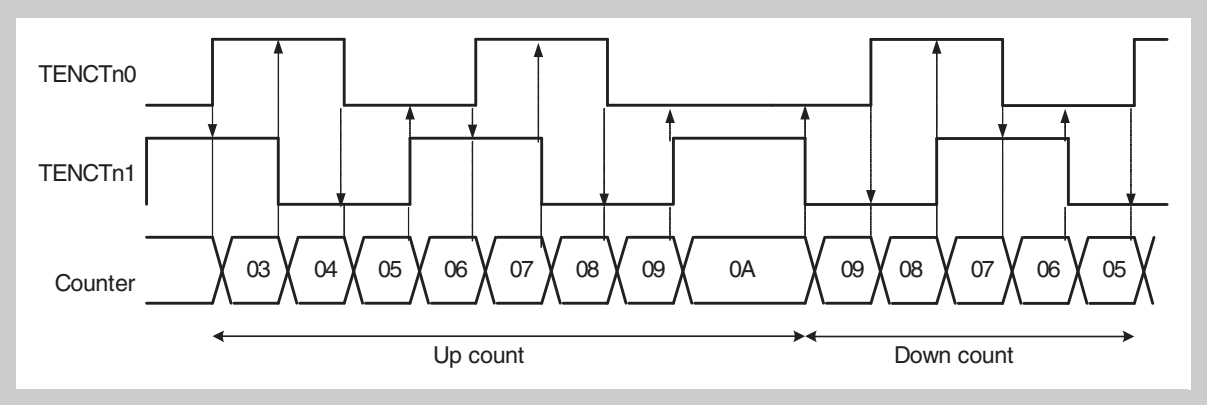


Figure 11-29 Encoder count function up/down count selection specification timing 4

Operation Example 2 <Register Setting>

TTnIOC2.TTnEIS[3:0]	Pins TENCTn1, TENCTn0	Edge detection specification invalid

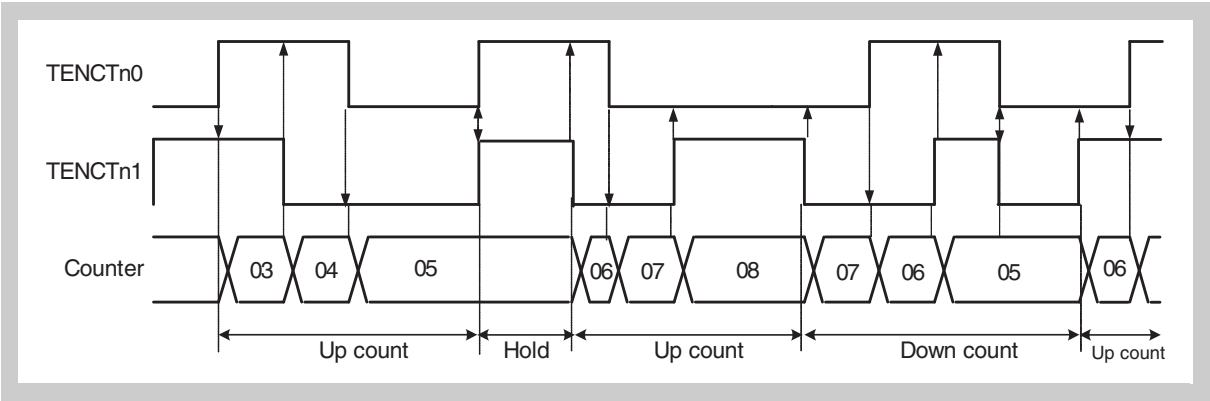


Figure 11-30 Encoder count function up/down count selection specification timing 5

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.TTnECM[1:0] bits)

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. TTnECM[1:0] = 00_B

Counter clear is not performed upon a match between the counter and compare values.

2. TTnECM[1:0] = 01_B

Counter clear is performed upon a match between the counter and the TTnCCR0 register.

Next count operation	Description
Up count	Clear counter to 0000_H.
Down count	Down count the counter value.

3. TTnECM[1:0] = 10_B

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 0000_H.

4. TTnECM[1:0] = 11_B

– 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 0000_H.
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 0000_H.

Caution In encoder compare mode (TTnMD[3:0] bits = 1000_B), or encoder capture-compare mode (TTnMD[3:0] bits = 1010_B), 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 (TTnCTL2.TTnLDE bit)

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 0000_H and setting value of TTnCCR0 register setting:

Set TTnLDE = 1, TTnECM[1:0] = 01_B and perform count operation. When TTnECM0 = 1, the counter is cleared to 0000_H 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 0000_H and TTnCCR0 register setting values.

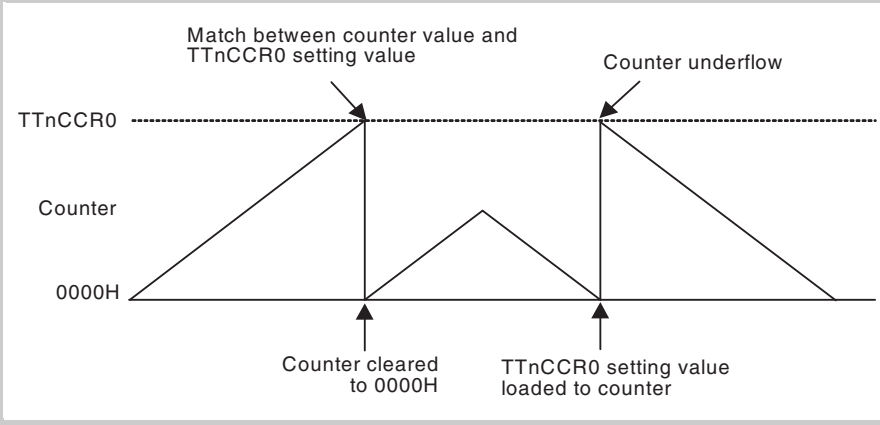


Figure 11-31 Encoder count function up/down count selection specification timing 6

(5) Counter clearing to 0000_H through encoder clear input (TECRTn pin) (TTnIOC3 register bits TTnSCE, TTnECS1, TTnECS0)

There are two methods to clear the counter to 0000_H 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 0000_H through detection of valid edge of TECRTn pin input (TTnSCE = 0)

When TTnSCE = 0, the counter is cleared to 0000_H 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.

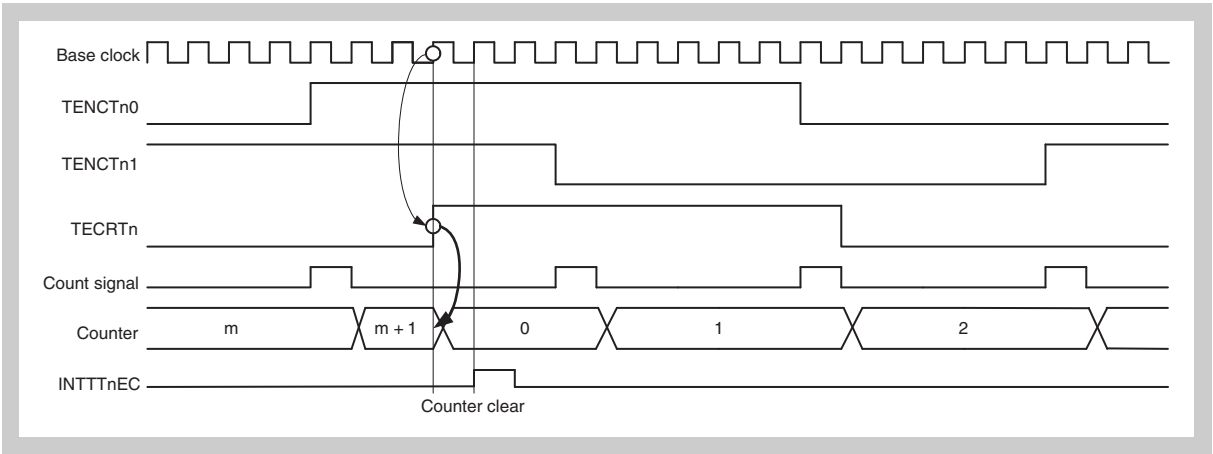


Figure 11-32 Counter clearing to 0000_H through encoder clear input (TECRTn pin) timings (1/5)
when TTnSCE = 0, TTnECS1, 0 = 01_B, TTnUDS = 11_B are set.

2. Method to clear counter to 0000_H through detection of level clear condition (TTnSCE = 1)

When TTnSCE = 1, the counter is cleared to 0000_H 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 TTnECS[1:0] are invalid.

Operation Example TTnSCE = 1, TTnCLA = 1, TTnCLB = 0, TTnCLZ = 1, TTnUDS[1:0] = 11_B

<Clear condition level>
TECRTn pin: High level
TENCTn1 pin: Low leve
TENCTn0 pin: High level

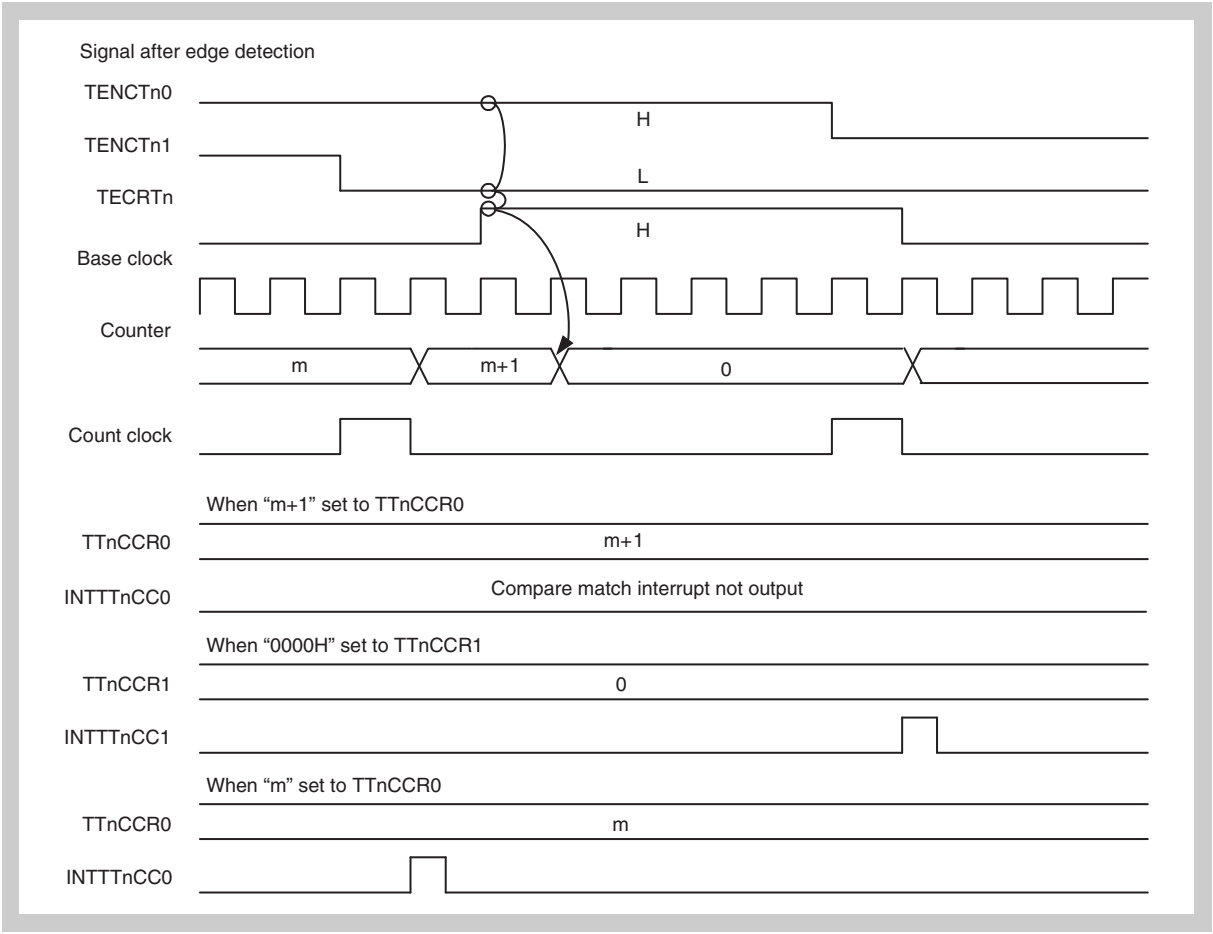


Figure 11-33 Counter clearing to 0000_H through encoder clear input (TECRTn pin) timings (2/5)
when TECRTn pin input is delayed from TENCTn1 pin input during up count.

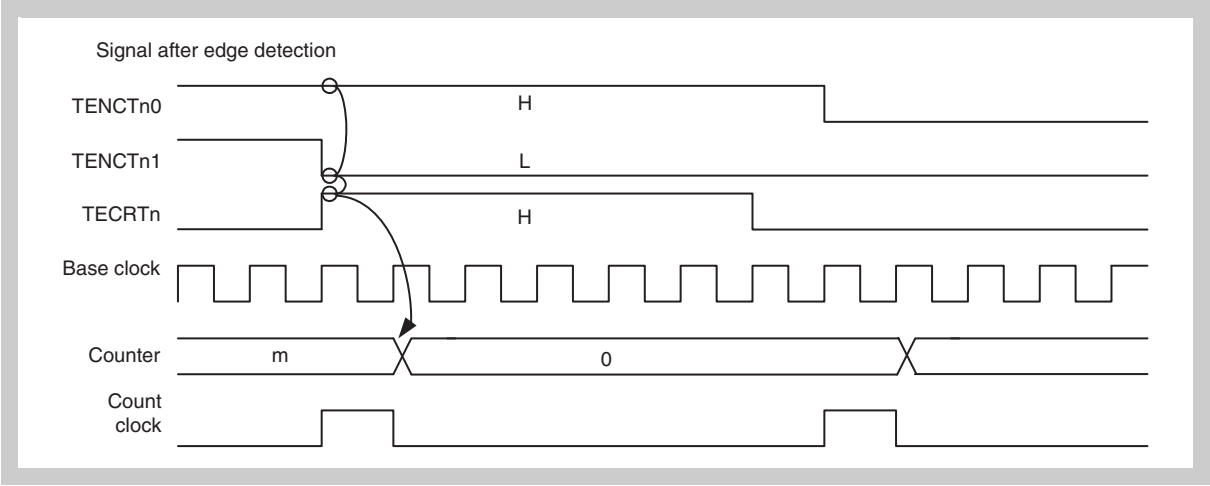


Figure 11-34 Counter clearing to 0000_H through encoder clear input (TECRTn pin) timings (3/5)
when TECRTn pin input and TENCTn1 pin input occur simultaneously during up count.

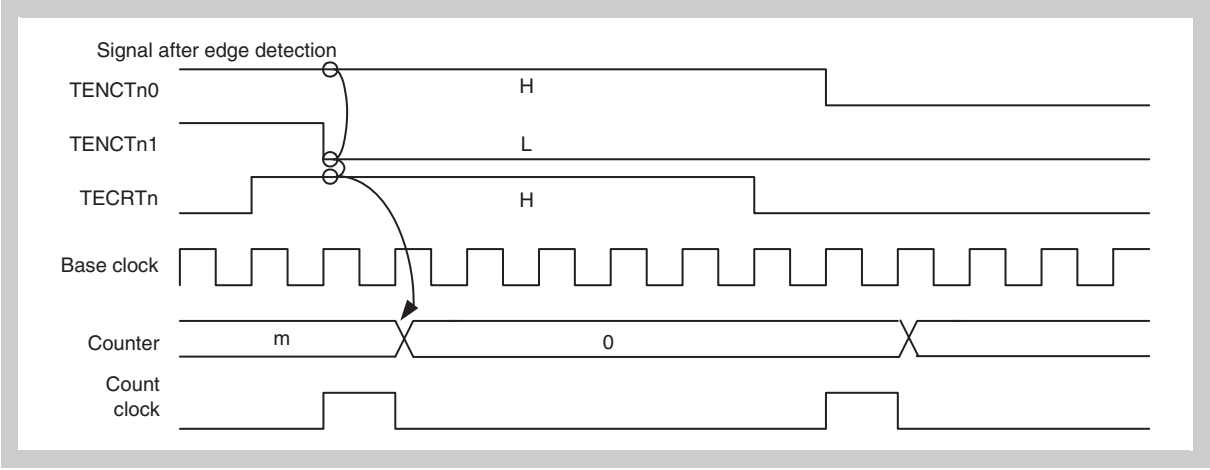


Figure 11-35 Counter clearing to 0000_H through Encoder clear input (TECRTn pin) timings (4/5)
when TECRTn pin input occurs earlier than TENCTn1 pin input during up count.

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 0000_H upon clear condition detection.

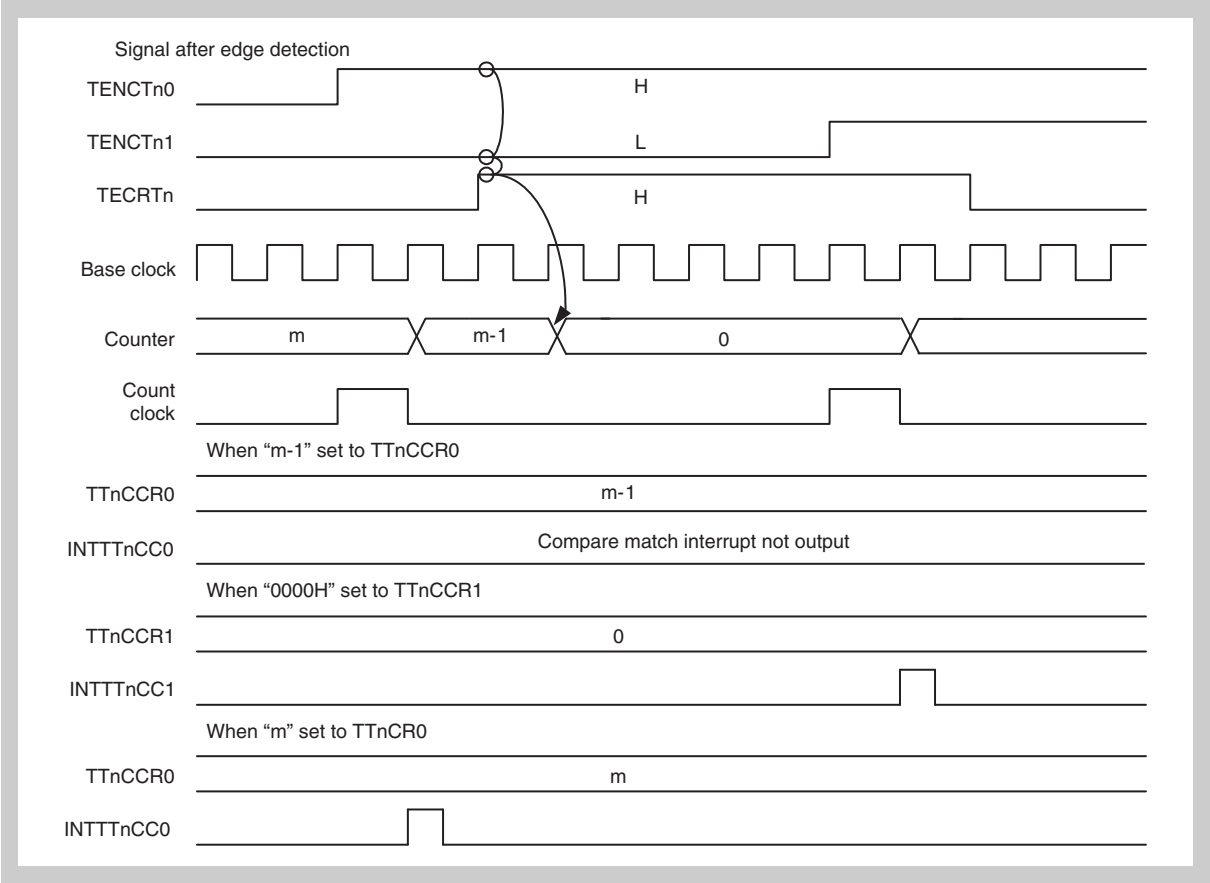


Figure 11-36 Counter clearing to 0000_H through encoder clear input (TECRTn pin) timings (5/5)
when TECRTn pin input occurs later than TENCTn1 pin input during down count.

No miscount occurs due to the TECRTn pin input delay during down count, similarly to during up count.

(6) Counter hold through TTnECC bit

By setting TTnCTL2.TTnECC bit 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 TTnECC bit

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.

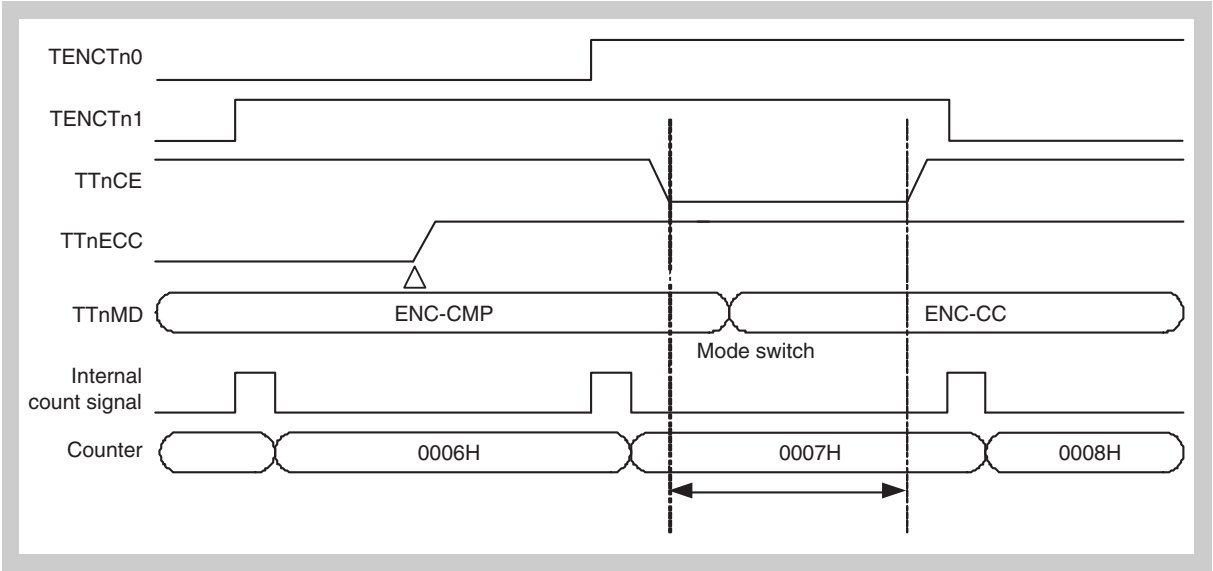


Figure 11-37 Counter hold through TTnECC bit timings (1/10) Mode switching using bit TTnECC

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 TTnECC bit setting

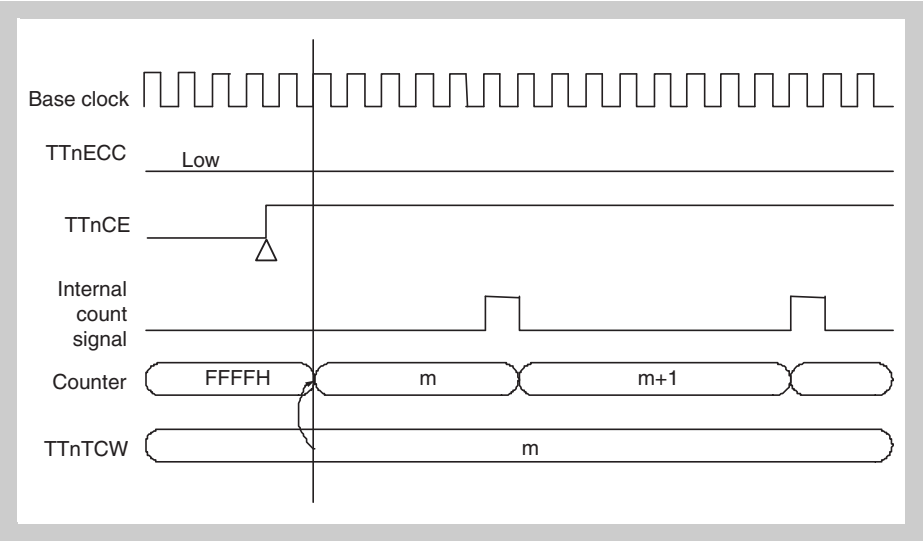


Figure 11-38 Counter hold through TTnECC bit timings (2/10)
Count operation when TTnECC = 0 is set

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 0000_H of TTnTCW register)

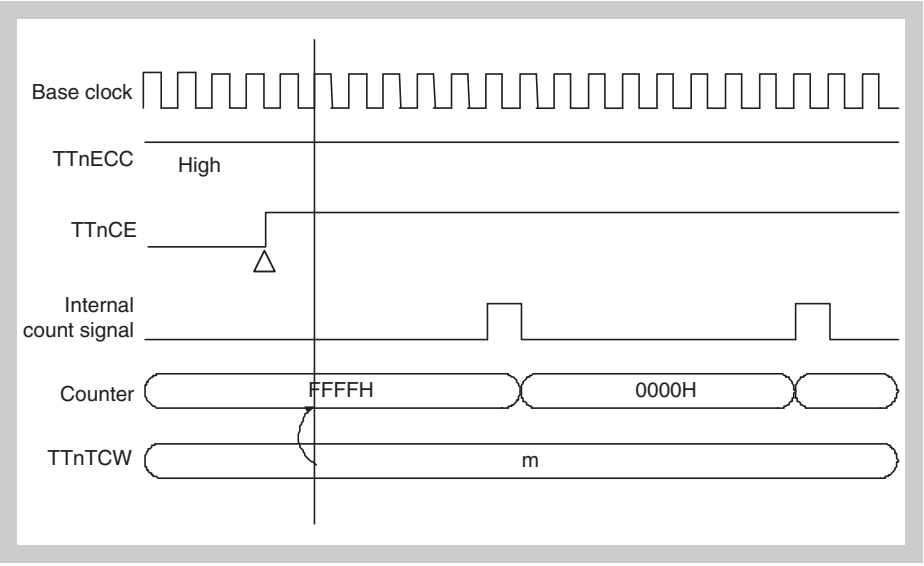


Figure 11-39 Counter hold through TTnECC bit timings (3/10)
Count operation when TTnECC = 1 is set

Since the setting value of the TTnTCW register is not loaded to the counter, the count operation is performed from initial value $FFFF_H$.

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) TTnECC bit rewrite timing and its influence on counter

1. When setting value of bit TTnECC is rewritten 0 → 1 → 0 if 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.

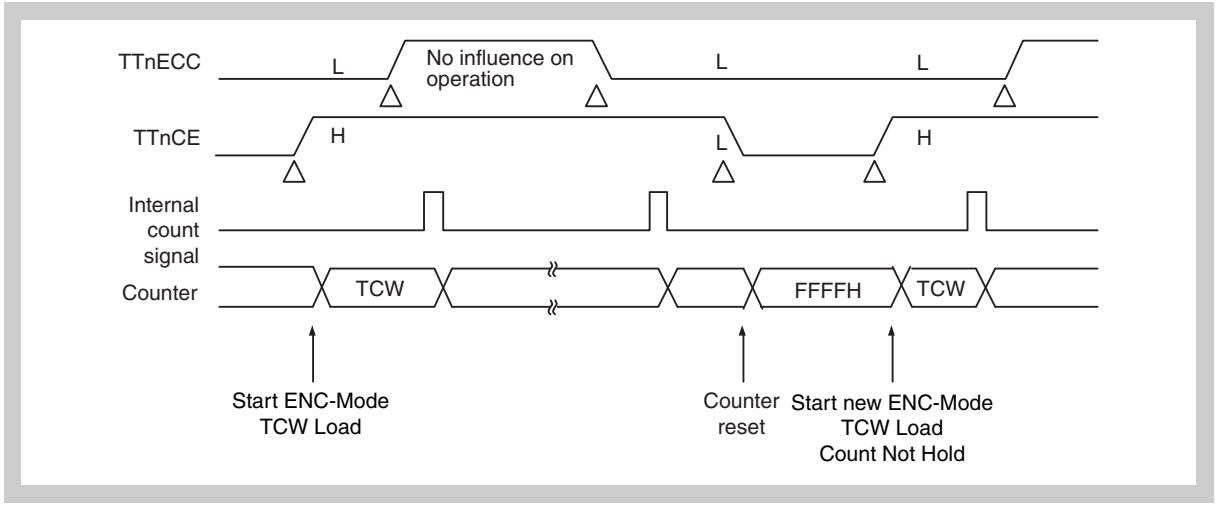


Figure 11-40 Counter hold through TTnECC bit timings (4/10) when setting value of TTnECC bit is rewritten 0 → 1 → 0 if TTnCE = 1

2. When setting value of bit TTnECC is rewritten 1 → 0 → 1 if 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 $FFFF_H$, without the setting value of the TTnTCW being loaded to the counter.

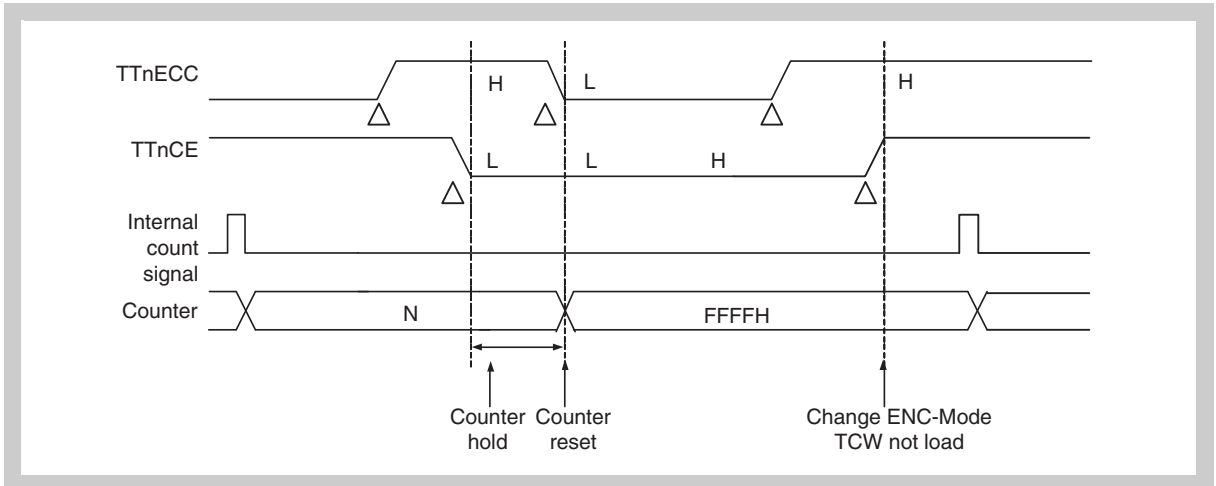


Figure 11-41 Counter hold through TTnECC bit timings (5/10) when setting value of TTnECC bit is rewritten 1 → 0 → 1 if TTnCE = 0

(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 $FFFF_H$ without loading the setting value of the TTnTCW register to the counter.

< Register setting conditions>

TTnCTL0.TTnMD[3:0] = 1000_B	Encoder compare mode
TTnCTL1.TTnUDS[1:0] = 00_B	Judgment of up/down count with count judgment mode 1
TTnCTL1.TTnECM[1:0] = 01_B	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.TTnEIS[1:0] = 01_B	Detection of rising edge of TENCTn0 and TENCTn1 pin
TTnIOC3.TTnSCE = 0, TTnIOC3.TTnECS[1:0] = 00_B	Valid edge detection clear (no edge specified)

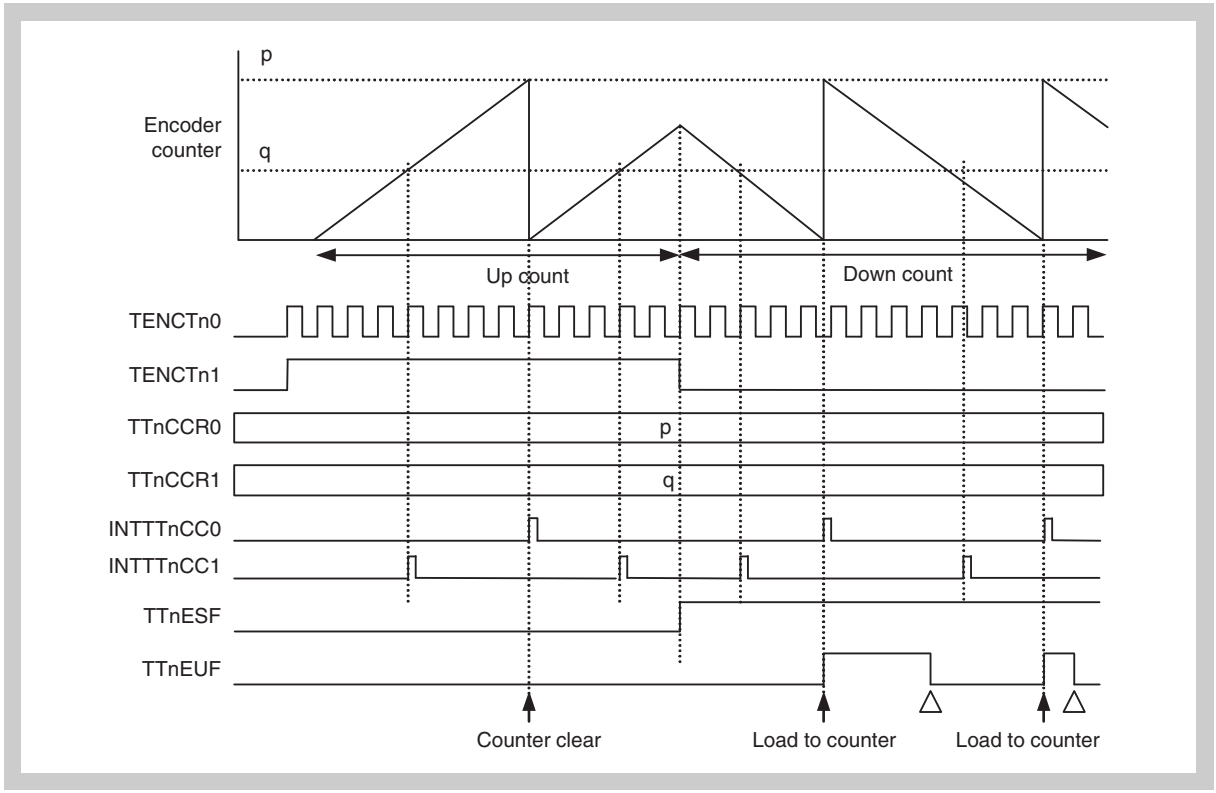


Figure 11-42 Counter hold through TTnECC bit timings (6/10)
Basic timing in encoder compare mode (1)

Since TTnUDS[1:0] and TTnEIS[1:0] that control the count operation are set to 00_B and 01_B (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 0000_H 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 0000_H and the setting value of the TTnCCR0 register by setting TTnLDE = 1 and TTnECM0 = 1.

<Register setting conditions>

TTnCTL0.TTnMD[3:0] = 1000_B	Encoder compare mode
TTnCTL1.TTnUDS[1:0] = 11_B	Judgment of up/down count with count judgment mode 4
TTnCTL1.TTnECM[1:0] = 00_B	No clear operation upon match between counter value and compare
TTnCTL1.TTnLDE = 0	No loading of setting value of TTnCCR0 register (p) to counter
TTnIOC3.TTnSCE = 0, TTnIOC3.TTnECS[1:0] = 01_B	Valid edge detection clear (rising edge specified)

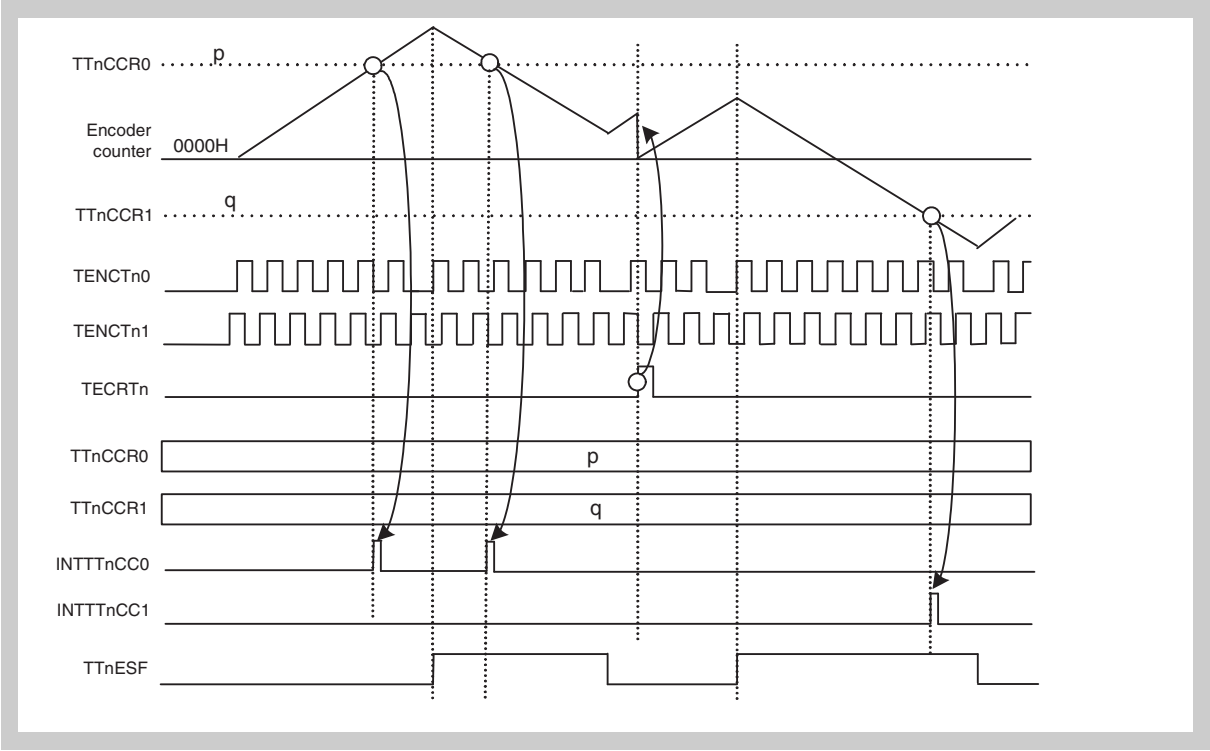


Figure 11-43 Counter hold through TTnECC bit timings (7/10)
Basic timing in encoder compare mode (2)

Since TTnUDS1, 0 that control the count operation are set to 11_B, 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 0000_H is done upon detection of the valid edge of the encoder clear input (pin TECRTn) when TTnIOC3.TTnSCE = 0. When TTnIOC3.TTnECS[0:1] = 01_B is set, the counter is cleared to 0000_H in synchronization with the operation clock, following detection of the rising edge of the TECRTn pin input.

<Register setting conditions>

TTnCTL0.TTnMD[3:0] = 1000_B	Encoder compare mode
TTnCTL1.TTnUDS[1:0] = 11_B	Judgment of up/down count with count judgment mode 4
TTnCTL1.TTnECM[1:0] = 11_B	No clear operation upon match between counter value and compare
TTnCTL1.TTnLDE = 0	Counter clear upon match between counter value and TTnCCR0 buffer register Counter clear upon match between counter value and TTnCCR1 buffer register (Since TTnCTL1.TTnECM[1:0] = 11_B, the setting of bit TTnLDE is invalid.)
TTnIOC3.TTnSCE = 0, TTnIOC3.TTnECS[1:0] = 00_B	Valid edge detection clear (no edge specified)

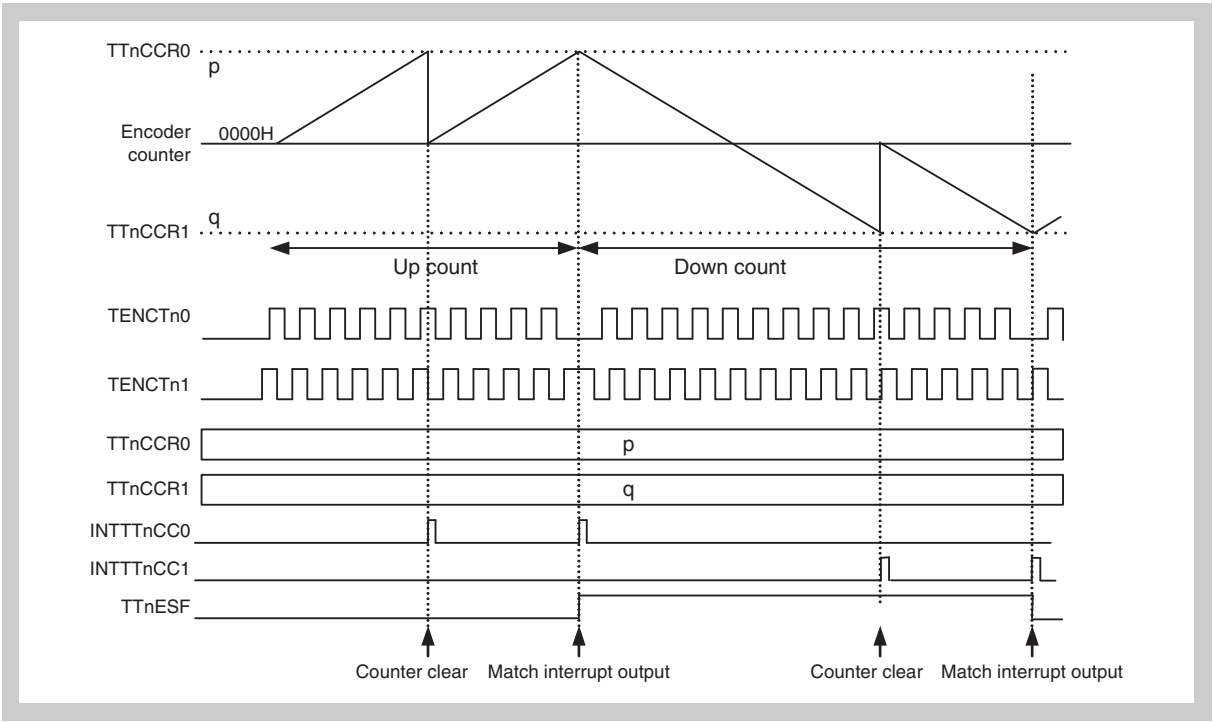


Figure 11-44 Counter hold through TTnECC bit timings (8/10)
Basic timing in encoder compare mode (3)

Since TTnUDS[1:0] that control the count operation are set to 11_B, 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 (p).

At this time, the counter is cleared to 0000_H 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 (q).

At this time, the counter is cleared to 0000_H if the next count operation is down count.

<Register setting conditions>

TTnCTL0.TTnMD[3:0] = 1001_B	Encoder compare mode
TTnCTL1.TTnUDS[1:0] = 11_B	Judgment of up/down count using count judgment method 4 (Since TTnCTL0: TTnMD3 to 0 = 1001_B, the setting values of bits TTnECM1, TTnECM0, and TTnLDE are invalid.)
TTnIOC3.TTnSCE = 0, TTnECS[1:0] = 00_B	Valid edge detection clear (no edge specified)
TTnOPT1.TTnIS[1:0] = 01_B	Detection of rising edge of TITn0 pin input
TTnOPT1.TTnIS[3:2] = 01_B	Detection of rising edge of TITn1 pin input

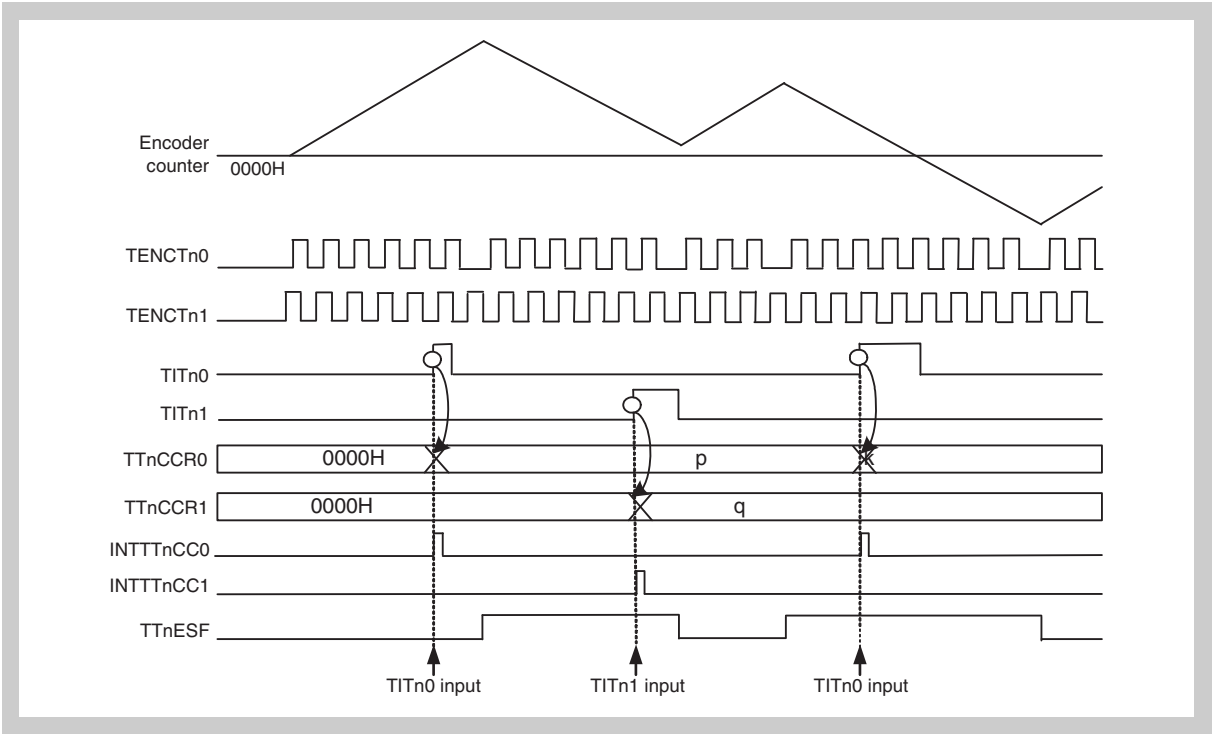


Figure 11-45 Counter hold through TTnECC bit timings (9/10)
Basic timing in encoder capture mode (1)

Since TTnUDS1, 0 that control the count operation are set to 11_B, the counter is operated through detection of the phase of pins TENCTn0 and TENCTn1.

Upon detection of the edge of pin TITn0, the counter value is saved to the capture register (TTnCCR0), and a capture interrupt (INTTTnCC0) is output.

Upon detection of the edge of TITn1, the counter value is saved to the capture register (TTnCCR1), and a capture interrupt (INTTTnCC1) is output.

<Register setting conditions>

TTnCTL0.TTnMD[3:0] = 1010_B	Encoder compare mode
TTnCTL1.TTnUDS[1:0] = 00_B	Judgment of up/down count using count judgment mode 1
TTnCTL1.TTnECM[1:0] = 01_B	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.TTnEIS[1:0] = 01_B	Detection of rising edge of TENCTn0 and TENCTn1 pin inputs
TTnIOC3.TTnSCE = 0, TTnIOC3.TTnECS[1:0] = 00_B	Valid edge detection clear (no edge specified)
TTnOPT1.TTnIS[3.2] = 01_B	Detection of rising edge of TITn1 pin input

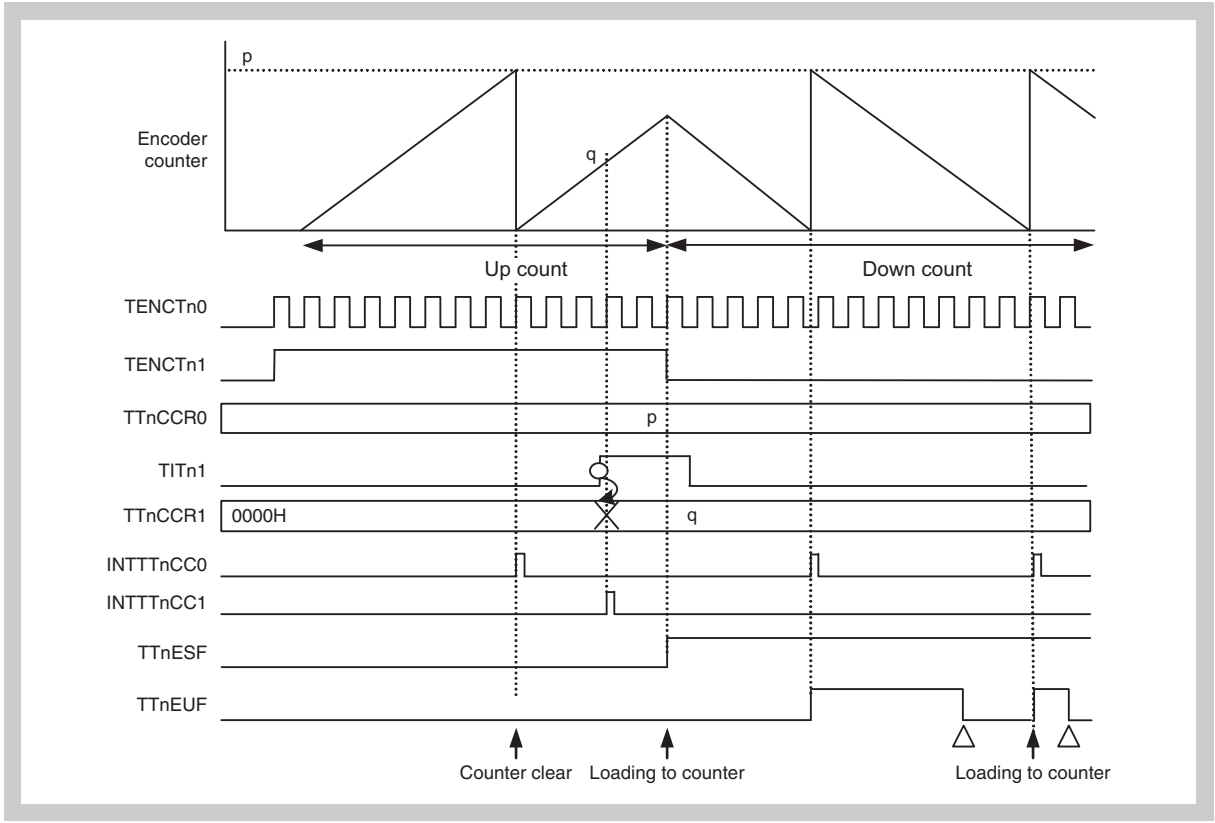


Figure 11-46 Counter hold through TTnECC bit timings (10/10)
Basic timing in encoder capture compare mode (2)

Since TTnUDS[1:0] and TTnEIS[1:0]that controls the count operation are set to 00_B and 01_B (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 000_H 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 0000_H and the setting value of the TTnCCR0 register by setting TTnLDE = 1 and TTnECM0 = 1.

11.6.10 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 0000_H 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 TTnOL0 bit. If TTnOL0 = 0, a low level is output a 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 0000_H.

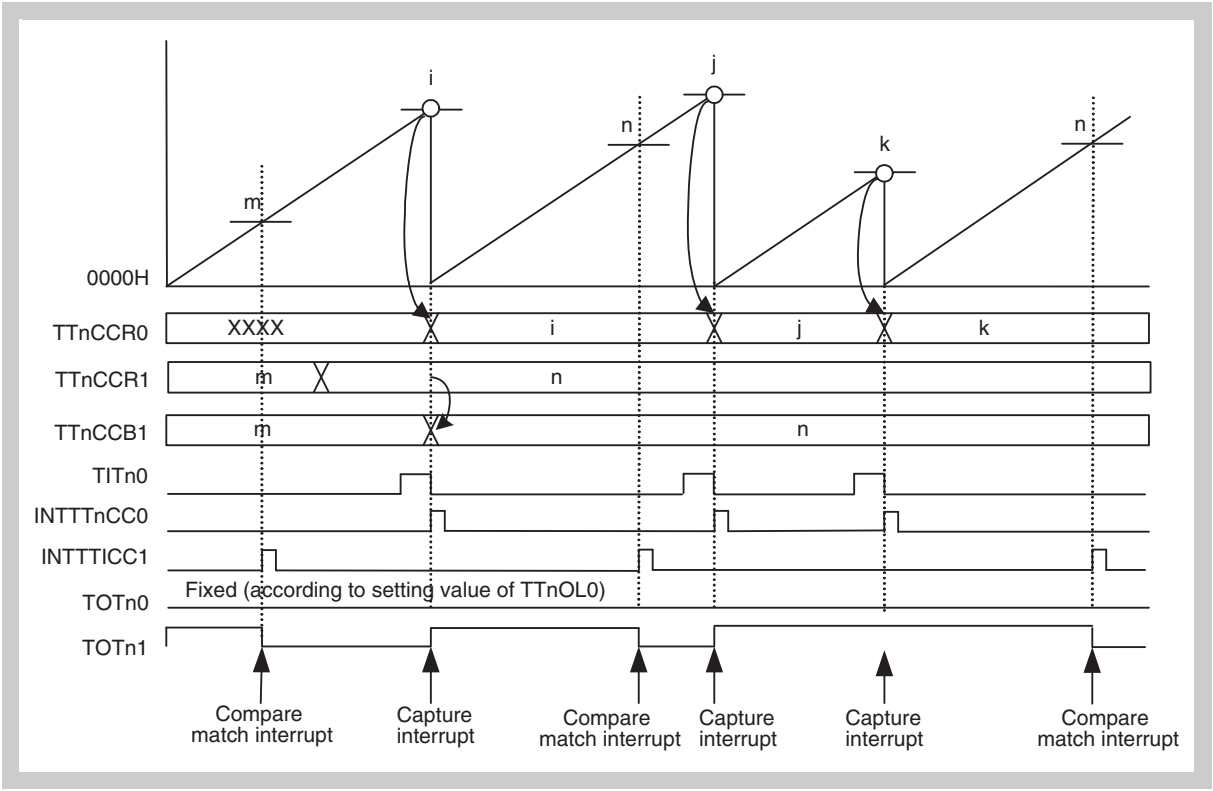


Figure 11-47 Basic timing in offset trigger generation mode

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 0000_H 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.

11.7 Cautions

(1) Encoder load mode setting during TMT operating

Description If TMT operates

- in encoder compare mode (TTnCTL1.TTnMD[3:0] = 1000_B)
- or in encoder capture/compare mode (TTnCTL1.TTnMD[3:0] = 1010_B),
- and the encoder load 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 encoder load mode is set (TTnCTL2.TTnLDE = 1) during TMT operation in encoder compare mode (TTnCTL1.TTnMD[3:0] = 1000_B) or in encoder capture/compare mode (TTnCTL1.TTnMD[3:0] = 1010_B) 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).

(2) Compare and clear function does not work at start timing in encoder mode

In encoder compare mode (TTnMD[3:0] = 1000_B), or encoder capture-compare mode (TTnMD[3:0] = 1010_B), if the compare registers (TTnCCR0, TTnCCR1) are set to the same value of the TTnTCW register when bit TTnECC = 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).

(3) Capture operation of illegal data before first counting up

In free-running mode (TTnMD[3:0] = 0101_B), pulse width measurement mode (TTnMD[3:0] = 0110_B), and offset trigger generation mode (TTnMD[3:0] = 1100_B, when a lower count clock (TTnEEE = 0, TTnCKS[2:0] = 011_B to 111_B) or an external clock (TTnEEE = 1) is selected, the timer captures the value of $FFFF_H$ and outputs a capture interrupt signal (INTTTnCCm), if a capture trigger signal (TITnm) is enabled and input before first counting up. This captured data and the corresponding interrupt might be useless.

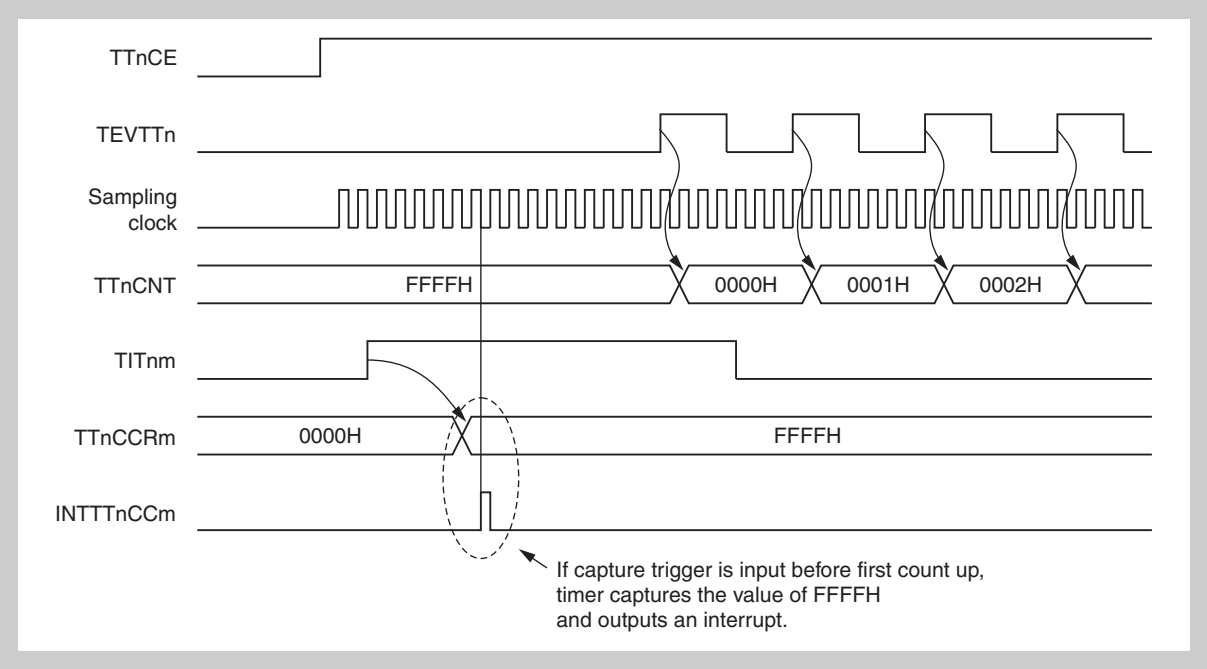


Figure 11-48 Capture operation of illegal data before first counting up

Chapter 12 16-bit 2-Phase Encoder Input Up/Down Counter/General Purpose Timer (TMENC10)

The V850E/PH2 microcontrollers have following number of channels of the 2-Phase Encoder Input Up/Down Counter/General Purpose Timer (TMENC10):

TMENC10	µPD70F3447	µPD70F3187
Instances	0	1
Names	–	TMENC10

Note TMENC10 is not available on µPD70F3447.

12.1 Features

Timer ENC10 (TMENC10) is a 16-bit up/down counter that performs the following operations.

- General-purpose timer mode
  - Free-running timer
  - PWM output
- Up/down counter mode
  - UDC mode A
  - UDC mode B

12.2 Function Outline

- Compare register × 2
- Capture/compare register × 2
- Interrupt request source
  - Capture/compare match interrupt × 2
  - Compare match interrupt × 2
  - Overflow interrupt × 1
  - Underflow interrupt × 1
- Capture request signal × 2
  The TMENC10 value can be latched using the valid edge of the TICC10, TICC11 pins corresponding to the capture/compare register as the capture trigger.
- Base clock (f_{CLK}) = f_{XX}/4
  (f_{CLK} = 16 MHz @ f_{XX} = 64 MHz)
- Count clocks selectable through division by prescaler
- 2-phase encoder input
  The 2-phase encoder signal from external is used as the count clock of the timer counter with the external clock input pins (TIUD1, TCUD1). The counter mode can be selected from among the four following modes.

  Mode 1: Counts the input pulses of the count pulse input pin. Up/down is specified by the level of one more input pin.

  Mode 2: Counts up/down using the respective input pulses of the up count pulse input pin and down count pulse input pin.

  Mode 3: Counts up/down using the phase relationship of the pulses input to 2 pins.

  Mode 4: Counts up/down using the phase relationship of the pulses input to 2 pins. Counting is done using the respective rising edges and the falling edges of the pulses.
- PWM output function
  In general-purpose timer mode, 16-bit resolution PWM output can be output from the TO1 pin.
- Timer clear
  - The following timer clear operations are performed according to the mode that is used.

    (a) General-purpose timer mode: Timer clear operation is possible upon occurrence of match with CM100 set value.

    (b) Up/down counter mode: The timer clear operation can be selected from among the following four conditions.
  - Timer clear performed upon occurrence of match with CM100 set value during TMENC10 up count operation, and timer clear performed upon occurrence of match with CM101 set value during TMENC10 down count operation.
  - Timer clear performed only by external input.
  - Timer clear performed upon occurrence of match between TMENC10 count value and CM100 set value.
  - Timer clear performed upon occurrence of external input and match between TMENC10 count value and CM100 set value.
- External pulse output (TO1) × 1

12.3 Basic Configuration

The basic configuration is shown below.

Table 12-1 Timer ENC10 Configuration List

Timer	Count Clock	Register	Read/Write	Generated Interrupt Signal	Capture Trigger
Timer ENC10	$f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, $f_{XX}/64$, $f_{XX}/128$, $f_{XX}/256$, $f_{XX}/512$	TMENC10	Read/write	INTOVF INTUDF	–
		CM100	Read/write	INTCM10	–
		CM101	Read/write	INTCM11	–
		CC100	Read/write	INTCC10	TICC10
		CC101	Read/write	INTCC11	TICC11

Figure 12-1 shows the block diagram of timer ENC10.

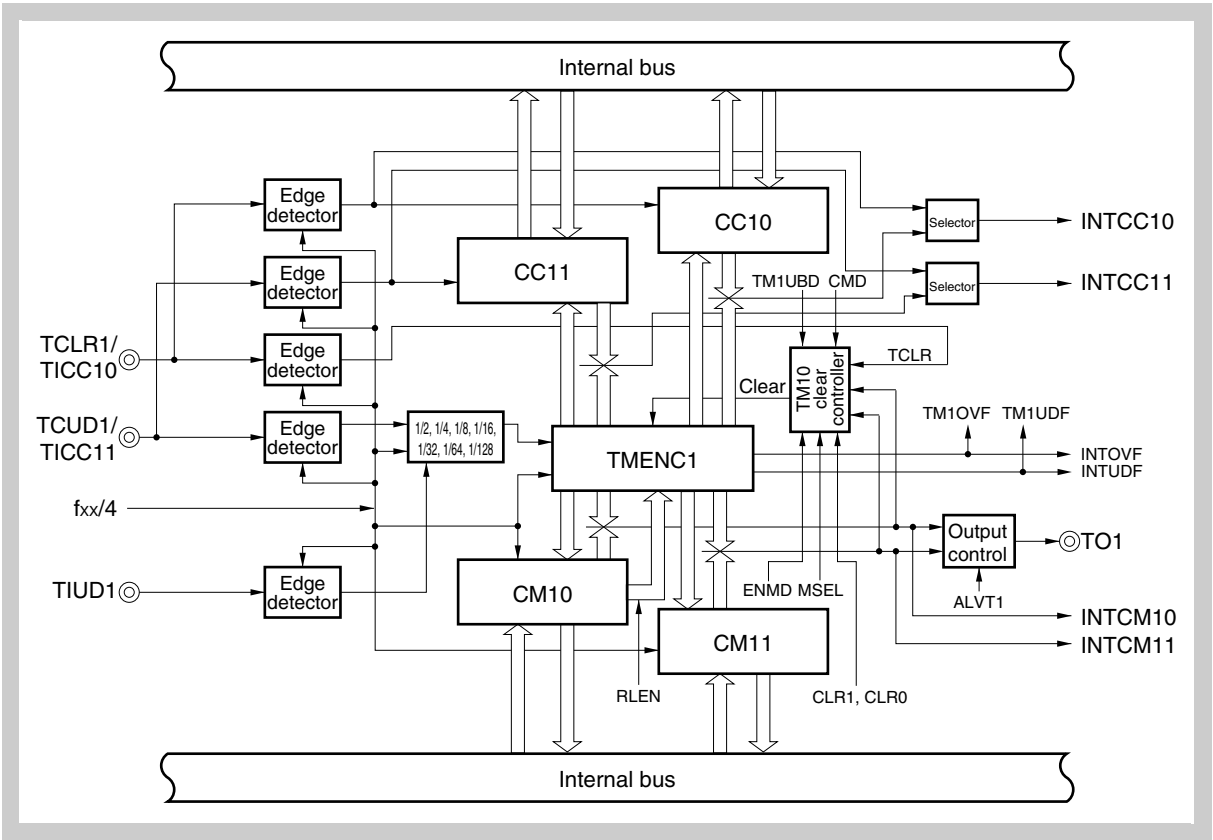


Figure 12-1 Block diagram of Timer ENC10 (TMENC10)

Note The TICC11 interrupt is the signal of the interrupt from the TICC11 pin or the interrupt from the TICC10 pin, selected by the CSL bit of the CSL1 register.

Remark f_{XX}: Internal system clock

(1) Timer ENC10 (TMENC10)

TMENC10 is a 2-phase encoder input up/down counter and general-purpose timer.

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

Address $FFFFF6B0_H$

Initial Value 0000_H. This register is cleared by any reset.

TMENC10

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Counter value															
R/W															

Caution

1. Write to TMENC10 is enabled only when the TMC10.TM1CE bit is “0” (count operation disabled).
2. It is prohibited to clear the TUM10.CMD bit (general-purpose timer mode) to 0 and to set the TUM10.MSEL bit (UDC mode B) to 1.
3. Continuous reading of TMENC10 is prohibited. If TMENC10 is continuously read, the second value read may differ from the actual value. If TMENC1n must be read twice, be sure to read another register between the first and the second read operation.
4. Writing the same value to the TMENC10, CC100, and CC101 registers, and the STATUS10 register is prohibited.
   Writing the same value to the CCR10, TUM10, TMC10, SESA10, and PRM10 registers, and CM100 and CM101 registers is permitted (writing the same value is guaranteed even during a count operation).

TMENC10 start and stop is controlled by the TMC10.TM1CE bit.

The TMENC10 operation consists of the following two modes.

(a) General-purpose timer mode

In the general-purpose timer mode, TMENC10 operates as a 16-bit interval timer, free-running timer, or for PWM output.

Counting is performed based on the clock selected by software.

Division by the prescaler can be selected for the count clock from among $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, $f_{XX}/64$, $f_{XX}/128$, $f_{XX}/256$, or $f_{XX}/512$ with bits PRM102 to PRM100 of prescaler mode register 10 (PRM10) (f_{XX}: internal system clock).

(b) Up/down counter mode (UDC mode)

In the UDC mode, TMENC10 functions as a 16-bit up/down counter, counting based on the TCUD1 and TIUD1 input signals.

Two operation modes can be set with the MSEL bit of the TUM register for this mode.

- **UDC mode A (when CMD bit = 1, MSEL bit = 0)**

  TMENC10 can be cleared by setting the CLR1 and CLR0 bits of the TMC10 register.

- **UDC mode B (when CMD bit = 1, MSEL bit = 1)**

  TMENC10 is cleared upon match with CM100 during TMENC10 up count operation.
  TMENC10 is cleared upon match with CM101 during TMENC10 down count operation.
  When the TM1CE bit of the TMC10 register is “1”, TMENC10 counts up when the operation mode is the general-purpose mode, and counts up/down when the operation mode is the UDC mode.

The conditions for clearing the TMENC10 are classified as follows depending on the operation mode.

Table 12-2 Timer ENC10 (TMENC10) Clear Conditions

Operation Mode	TUM10 Register		TMC10 Register			TMENC10 Clear
	CMD	MSEL	ENMD	CLR1	CLR0	
General-purpose timer mode	0	0	0	×	×	Clearing not performed
			1	×	×	Cleared upon match with CM100 set value
UDC mode A	1	0	×	0	0	Cleared only by TCLR1 input
			×	0	1	Cleared upon match with CM1n0 set value during up count operation
			×	1	0	Cleared by TCLR1 input or upon match with CM100 set value during up count operation
			×	1	1	Clearing not performed
UDC mode B	1	1	×	×	×	Cleared upon match with CM100 set value during up count operation or upon match with CM101 set value during down count operation
Settings other than the above						Setting prohibited

Remark ×: Indicates that the set value of that bit is ignored.

(2) Compare register 100 (CM100)

CM100 is a 16-bit register that always compares its value with the value of TMENC10. When the value of a compare register matches the value of TMENC10, an interrupt signal is generated. The interrupt generation timing in the various modes is described below.

- In general-purpose timer mode (TUM10.CMD bit = 0) and UDC mode A (TUM10.MSEL bit = 0), an interrupt signal (INTCM10) is always generated upon occurrence of a match.
- In UDC mode B (TUM10.MSEL bit = 1), an interrupt signal (INTCM10) is generated only upon occurrence of a match during up count operation.

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

Address $FFFFF6B2_H$

Initial Value 0000_H. This register is cleared by any reset.

CM100

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

Caution When the TMC10.TM1CE bit is set (1), it is prohibited to overwrite the value of the CM100 register.

(3) Compare register 101 (CM101)

CM101 is a 16-bit register that always compares its value with the value of TMENC10. When the value of a compare register matches the value of TMENC10, an interrupt signal is generated. The interrupt generation timing in the various modes is described below.

- In general-purpose timer mode (TUM10.CMD bit = 0) and UDC mode A (TUM10.MSEL bit = 0), an interrupt signal (INTCM11) is always generated upon occurrence of a match.
- In UDC mode B (TUM10.MSEL bit = 1), an interrupt signal (INTCM101) is generated only upon occurrence of a match during up count operation.

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

Address $FFFFF6B4_H$

Initial Value 0000_H. This register is cleared by any reset.

CM101

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

Caution When the TMC10.TM1CE bit is set (1), it is prohibited to overwrite the value of the CM101 register.

(4) Capture/compare register 100 (CC100)

CC100 is a 16-bit register. It can be used as a capture register or as a compare register through specification with capture/compare control register n (CCR).

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

Address $FFFFF6B6_H$

Initial Value 0000_H. This register is cleared by any reset.

CC100

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

Caution

1. When used as a capture register (CCR.CMS0 bit = 0), write access is prohibited.
2. When used as a compare register (CCR.CMS0 bit = 1) and the TMC10.TM1CE bit is set (1), overwriting the CC100 register values is prohibited.
3. When the TMC10.TM1CE bit is cleared (0), the capture trigger is disabled.
4. When the operation mode is changed from capture register to compare register, set a new compare value.
5. Continuous reading of CC100 is prohibited. If CC100 is continuously read, the second read value may differ from the actual value. If CC100 must be read twice, be sure to read another register between the first and the second read operation.

(a) When set as a capture register (CCR.CMS0 bit = 0)

When CC100 is set as a capture register, the valid edge of the corresponding external TICC10 signal is detected as the capture trigger. TMENC10 latches the count value in synchronization with the capture trigger (capture operation). The latched value is held in the capture register until the next capture operation.

The valid edge of external interrupts (rising edge, falling edge, both edges) is selected with signal edge selection register 10 (SESA10).

When the CC100 register is specified as a capture register, an INTCC10 interrupt is generated upon detection of the valid edge of the external TICC10 signal.

(b) When set as a compare register (CCR.CMS0 bit = 1)

When CC100 is set as a compare register, it always compares its own value with the value of TMENC10. If the value of CC100 matches the value of the TMENC10 counter, CC100 generates an interrupt signal (INTCC10).

(5) Capture/compare register 101 (CC101)

CC101 is a 16-bit register. It can be used as a capture register or as a compare register through specification with capture/compare control register (CCR).

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

Address $FFFFF6B8_H$

Initial Value 0000_H. This register is cleared by any reset.

CC101

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

Caution

1. When used as a capture register (CCR.CMS1 bit = 0), write access is prohibited.
2. When used as a compare register (CCR.CMS1 bit = 1) and the TMC10.TM1CE bit is set (1), overwriting the CC101 register values is prohibited.
3. When the TMC10.TM1CE bit is cleared (0), the capture trigger is disabled.
4. When the operation mode is changed from capture register to compare register, set a new compare value.
5. Continuous reading of CC101 is prohibited. If CC101 is continuously read, the second read value may differ from the actual value. If CC101 must be read twice, be sure to read another register between the first and the second read operation.

(a) When set as a capture register (CCR.CMS1 bit = 0)

When CC101 is set as a capture register, the valid edge of the corresponding external TICC11 signal is detected as the capture trigger. TMENC10 latches the count value in synchronization with the capture trigger (capture operation). The latched value is held in the capture register until the next capture operation.

The valid edge of external interrupts (rising edge, falling edge, both edges) is selected with signal edge selection register 10 (SESA10).

When the CC101 register is specified as a capture register, an INTCC11 interrupt is generated upon detection of the valid edge of the external TICC11 signal.

(b) When set as a compare register (CCR.CMS1 bit = 1)

When CC101 is set as a compare register, it always compares its own value with the value of TMENC10. If the value of CC101 matches the value of the TMENC10 counter, CC101 generates an interrupt signal (INTCC11).

12.4 Control Registers

(1) Timer unit mode register 10 (TUM10)

The TUM10 register is an 8-bit register used to specify the TMENC10 operation mode or to control the operation of the PWM output pin.

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

Address $FFFFF6BB_H$

Initial Value 00_H. This register is cleared by any reset.

TUM10

7	6	5	4	3	2	1	0
CMD	0	0	0	TOE	ALVT1	0	MSEL
R/W	R	R	R	R/W	R/W	R	R/W

Caution

1. Changing the value of the TUM10 register during TMENC10 operation (TMC.TM1CE bit = 1) is prohibited.
2. When the CMD bit = 0 (general-purpose timer mode), setting MSEL bit = 1 (UDC mode B) is prohibited.

Table 12-3 TUM10 register contents

Bit position	Bit name	Function
7	CMD	Specifies TMENC10 operation mode. 0: General-purpose timer mode (up count) 1: UDC mode (up/down count)
3	TOE	Controls timer output (TO1). 0: Timer output disabled. The output level is set to the inactive level. 1: Timer output enabled. **Note:** When CMD bit = 1 (UDC mode), timer output is disabled regardless of the TOE bit setting. At this time, the timer output adopts the inactive level according to the active level specified by the ALVT1 bit.
2	ALVT1	Specifies the active level of the timer output (TO1). 0: Active level is high level (H). 1: Active level is low level (L). **Note:** The inactive level is inverse to the active level. If active level is set to H-level the inactive level becomes L-level, and vice versa.
0	MSEL	Selects the sub mode in UDC Mode (CMD bit = 1). 0: UDC mode A. TMENC10 can be cleared by setting the TMC10.CLR[1:0] bits. 1: UDC mode B. TMENC10 is cleared in the following cases. - Upon match with CM100 during TMENC10 up count operation - Upon match with CM101 during TMENC10 down count operation **Note:** When UDC mode B is set, the ENMD, CLR1, and CLR0 bits of the TMC10 register become invalid.

(2) Timer control register 10 (TMC10)

The TMC10 register is used to enable/disable TMENC10 operation and to set transfer and timer clear operations.

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

Address $FFFFF6BC_H$

Initial Value 00_H. This register is cleared by any reset.

TMC10

7	6	5	4	3	2	1	0
0	TM1CE	0	0	RLEN	ENMD	CLR1	CLR0
R	R/W	R	R	R/W	R/W	R/W	R/W

Caution Changing the values of the TMC10 register bits other than the TM1CE bit during TMENC10 operation (TM1CE bit = 1) is prohibited.

Table 12-4 TMC10 register contents (1/2)

Bit position	Bit name	Function
6	TM1CE	Controls TMENC10 operation. 0: Count operation disabled. 1: Count operation enabled
3	RLEN	Controls reload operation in UDC Mode A. 0: Reload operation from CM100 register to TMENC10 disabled. 1: Reload operation from CM100 register to TMENC10 enabled. • When RLEN = 1, the value set to CM100 is transferred to TMENC10 upon occurrence of TMENC10 underflow. **Note:** When TUM10.CMD bit = 0 (general-purpose timer mode) or when TUM10.MSEL bit = 1 (UDC mode B), the RLEN bit setting is invalid, and a reload operation is not executed even if the RLEN bit is set (1).
2	ENMD	Controls clear operation in general purpose mode. 0: Clear operation is disabled (free-running mode). Clearing is not performed even when TMENC10 and CM100 values match. 1: Clear operation is enabled. Clearing is performed upon match of TMENC10 and CM100 values. **Note:** When TUM10.CMD bit = 1 (UDC mode), the ENMD bit setting becomes invalid.

Table 12-4 TMC10 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>1, 0</td><td>CLR[1:0]</td><td>Controls the clear operation in UDC mode A.
<table>
<tr><th>CLR1</th><th>CLR0</th><th>Clear Operation Control in UDC Mode A</th></tr>
<tr><td>0</td><td>0</td><td>Clear only by external input (TCLR1)</td></tr>
<tr><td>0</td><td>1</td><td>Clear upon match of TMENC10 count value and CM100 set value</td></tr>
<tr><td>1</td><td>0</td><td>Clear by TCLR1 input or upon match of TMENC10 count value and CM100 set value</td></tr>
<tr><td>1</td><td>1</td><td>No clearing</td></tr>
</table>
Note: 1. Clearing by match of the TMENC10 count value and CM100 set value is valid only during TMENC10 up count operation (TMENC10 is not cleared during TMENC10 down count operation).
2. When TUM10.CMD bit = 0 (general-purpose timer mode) or when TUM10.MSEL bit = 1 (UDC mode B), the CLR1 and CLR0 bit settings are invalid.
3. When clearing by TCLR1 pin has been enabled by bits CLR1 and CLR0, clearing is performed whether the value of the TM1CE bit is 1 or 0.</td></tr>
</table>

(3) Capture/compare control register 10 (CCR10)

The CCR10 register specifies the operation mode of the capture/compare registers (CC100, CC101).

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

Address $FFFFF6BA_H$

Initial Value 00_H. This register is cleared by any reset.

CCR10

7	6	5	4	3	2	1	0
0	0	0	0	0	0	CMS1	CMS0
R	R	R	R	R	R	R/W	R/W

Caution

1. Overwriting the CCR10 register during TMENC10 operation (TMC10.TM1CE = 1) is prohibited.
2. The TCUD1 pin is used for the UDC mode and shared with the external capture input pin TICC11. Therefore, in the UDC mode, the external capture function cannot be used.
3. The TCLR1 pin is used for the UDC mode and alternately shared with the external capture input pin TICC10. Therefore, when the TCLR1 input is used in UDC mode A, the external capture function cannot be used.

Table 12-5 CCR10 register contents

Bit position	Bit name	Function
1	CMS1	Specifies the operation mode of the CC101 register. 0: CC101 operates as capture register. 1: CC101 operates as compare register
0	CMS0	Specifies the operation mode of the CC100 register. 0: CC100 operates as capture register. 1: CC100 operates as compare register

(4) Signal edge selection register 10 (SESA10)

The SESA10 register specifies the valid edge of external interrupt requests from external pins (TICC10, TICC11, TCLR1).

The valid edge (rising edge, falling edge, or both edges) can be specified independently for each pin.

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

Address $FFFFF6BD_H$

Initial Value 00_H. This register is cleared by any reset.

SESA10

7	6	5	4	3	2	1	0
TESUD1	TESUD0	CESUD1	CESUD0	IES111	IES110	IES101	IES100
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
TIUD1, TCUD1 input		TCLR1 input		TICC11 capture trigger input		TICC10 capture trigger input	

Caution

1. Changing the values of the SESA10 register bits during TMENC10 operation (TMC10.TM1CE bit = 1) is prohibited.
2. Be sure to set (1) the TMC10.TM1CE bit even when TMENC10 is not used and the TICC10 and TICC11 pins are used as external interrupts INTCC10 and INTCC11 respectively.
3. Before setting the trigger mode of the TICC10, TICC11, and TCLR1n pins, set the PM10 and PMC10 registers. If the PM10 and PMC10 registers are set after the SESA10 register has been set, an illegal interrupt, incorrect counting, and incorrect clearing may occur, depending on the timing of setting the PM10 and PMC10 registers.

Table 12-6 SESA10 register contents (1/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7, 6</td><td>TESUD1, TESUD0</td><td>Specifies the valid edge of TIUD1 and TCUD1 input pins.
<table>
<tr><th>TESUD1</th><th>TESUD0</th><th>Valid edge specification of TIUD1 and TCUD1 input pins</th></tr>
<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>Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>Both, rising and falling edges</td></tr>
</table>
Note: 1. The set values of the TESUD[1:0] bits are only valid in UDC mode A and UDC mode B.
2. If mode 4 is specified as the operation mode of TMENC10 (specified with PRM10.PRM10[2:0] bits), the set values of TESUD[1:0] bits are invalid.</td></tr>
</table>

Table 12-6 SESA10 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>5, 4</td><td>CESUD1,
CESUD0</td><td>Specifies the valid edge and level of the TCLR1 input pin.

<table><tr><th>CESUD1</th><th>CESUD0</th><th>Valid edge and level specification of TCLR1 input pin</th></tr><tr><td>0</td><td>0</td><td>Falling edge (TMENC10 cleared after edge detection)</td></tr><tr><td>0</td><td>1</td><td>Rising edge (TMENC10 cleared after edge detection)</td></tr><tr><td>1</td><td>0</td><td>Low level (TMENC10 clear status held)</td></tr><tr><td>1</td><td>1</td><td>High level (TMENC10 clear status held)</td></tr></table>
Note: The set values of the CESUD[1:0] bits are valid only in UDC mode A.</td></tr>
<tr><td>3, 2</td><td>IES111,
IES110</td><td>Specifies the valid edge and level of the TICC11 input pin.

<table><tr><th>IES111</th><th>IES110</th><th>Valid edge specification of TICC11 capture trigger input pin</th></tr><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>Setting prohibited</td></tr><tr><td>1</td><td>1</td><td>Both, rising and falling edges</td></tr></table>
Note: A valid edge on the TICC11 pin triggers the capture register CC101. Simultaneously an interrupt (INTCC11) is generated.</td></tr>
<tr><td>1, 0</td><td>IES101,
IES100</td><td>Specifies the valid edge and level of the TICC10 input pin.

<table><tr><th>IES101</th><th>IES100</th><th>Valid edge specification of TICC10 capture trigger input pin</th></tr><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>Setting prohibited</td></tr><tr><td>1</td><td>1</td><td>Both, rising and falling edges</td></tr></table>
Note: A valid edge on the TICC10 pin triggers the capture register CC100. Simultaneously an interrupt (INTCC10) is generated.</td></tr>
</table>

(5) Prescaler mode register 10 (PRM10)

The PRM register is used to perform the following selections.

- Selection of count clock in the general-purpose timer mode (TUM10.CMD bit = 0)
- Selection of count operation mode in the UDC mode (TUM10.CMD bit = 1)

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

Address $FFFFF6BE_H$

Initial Value 07_H

PRM10

7	6	5	4	3	2	1	0
0	0	0	0	0	PRM102	PRM101	PRM100
R	R	R	R	R	R/W	R/W	R/W

Caution

1. Overwriting the PRM10 register during TMENC10 operation (TMC10.TM1CE bit = 1) is prohibited.
2. When TUM10.CMD bit = 1 (UDC mode), setting the values of the PRM[2:0] bits to 000_B, 001_B, 010_B, and 011_B is prohibited.
3. When TMENC10 is in mode 4, specification of the valid edge for the TIUD1 and TCUD1 pins is invalid.

Table 12-7 PRM10 register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>2 to 0</td><td>PRM10[2:0]</td><td>Selects the count clock or count clock operation depending on the TMENC10 operation mode (specified by TUM10.CMD bit).
<table>
<tr><th rowspan="2">PRM102</th><th rowspan="2">PRM101</th><th rowspan="2">PRM100</th><th>TUM10.CMD = 0</th><th colspan="2">TUM10.CMD = 1</th></tr>
<tr><th>Count Clock</th><th>Count Clock</th><th>UDC Mode</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Setting prohibited</td><td colspan="2" rowspan="4">Setting prohibited</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>f_{XX}/8</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>f_{XX}/16</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>f_{XX}/32</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>f_{XX}/64</td><td rowspan="4">TIUD1</td><td>Mode 1</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>f_{XX}/128</td><td>Mode 2</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>f_{XX}/256</td><td>Mode 3</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>f_{XX}/512</td><td>Mode 4</td></tr>
</table>
• In general-purpose timer mode (TUM10.CMD bit = 0)
The count clock is fixed to the internal clock. The clock rate of TMENC10 is specified by the PRM10[2:0] bits.

• In UDC mode (TUM10.CMD bit = 1)
The TMENC10 count triggers in the UDC mode are as follows.
<table>
<tr><th>Operation Mode</th><th>TMENC10 Operation</th></tr>
<tr><td>Mode 1</td><td>Down count when TCUD1 = high level
Up count when TCUD1 = low level</td></tr>
<tr><td>Mode 2</td><td>Up count upon detection of valid edge of TIUD1 input
Down count upon detection of valid edge of TCUD1 input</td></tr>
<tr><td>Mode 3</td><td>Automatic judgment by TCUD1 input level upon detection of valid edge of TIUD1 input</td></tr>
<tr><td>Mode 4</td><td>Automatic judgment upon detection of both edges of TIUD1 input and both edges of TCUD1 input</td></tr>
</table>
Remark: f_{XX}: Internal system clock.</td></tr>
</table>

(6) Status register 10 (STATUS10)

The STATUS10 register indicates the operating status of TMENC10.

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

Address $FFFFF6BF_H$

Initial Value 00_H. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
STATUS10	0	0	0	0	0	TM1UDF	TM1OVF	TM1UBD
	R	R	R	R	R	R	R	R

Caution Overwriting the STATUS10 register during TMENC10 operation (TMC10.TM1CE bit = 1) is prohibited.

Table 12-8 STATUS10 register contents

Bit position	Bit name	Function
2	TM1UDF	Indicates the TMENC10 underflow status. 0: No TMENC10 count underflow. 1: TMENC10 count underflow The TM1UDF bit is cleared (0) upon completion of read access to the STATUS10 register from the CPU.
1	TM1OVF	Indicates the TMENC10 overflow status. 0: No TMENC10 count overflow. 1: TMENC10 count overflow. The TM1OVF bit is cleared (0) upon completion of read access to the STATUS10 register from the CPU.
0	TM1UBD	Indicates the TMENC10 up/down counter operation status. 0: TMENC10 up count in progress. 1: TMENC10 down count in progress. The state of the TM1UBD bit differs according to the mode as follows. • The TM1UBD bit is fixed to 0 when the TUM10.CMD bit = 0 (general-purpose timer mode). • The TM1UBD bit indicates the TMENC10 up/down count status when the TUM10.CMD bit = 1 (UDC mode)

12.5 Operation

12.5.1 Basic operation

The following two operation modes can be selected for TMENC10.

(1) General-purpose timer mode (TUM10.CMD = 0)

In the general-purpose timer mode, the TMENC10 operates either as a 16-bit interval timer or as a PWM output timer (count operation is up count only).

The count clock to TMENC10 is selected by prescaler mode register 10 (PRM10).

(2) Up/down counter mode (UDC mode) (TUM10.CMD = 1)

In the UDC mode, TMENC10 operates as a 16-bit up/down counter.

External clock input (TIUD1, TCUD1 pins) set by PRM10 register setting is used as the TMENC10 count clock.

The UDC mode is further divided into two modes according to the TMENC10 clear conditions.

- **UDC mode A (TUM10.CMD = 1, TUM10.MSEL = 0)**

The TMENC10 clear source can be selected as external clear input (TCLR1), the internal signal indicating a match between the TMENC10 count value and the CM100 set value during an up count operation, or the logical sum (OR) of the two signals, using the CLR1 and CLR0 bits of the TMC10 register.

TMENC10 can transfer (reload) the value of CM100 upon occurrence of TMENC10 underflow, when the RLEN bit of the TMC10 register is set (1).

- **UDC mode B (TUM10.CMD = 1, TUM10.MSEL = 1)**

The status of TMENC10 after a match of the TMENC10 count value and CM100 set value is as follows.

<1> In case of an up count operation, TMENC10 is cleared (0000H), and the INTCM10 interrupt is generated.
<2> In case of a down count operation, the TMENC10 count value is decremented (-1).

The status of TMENC10 after a match of the TMENC10 count value and CM101 set value is as follows.

<1> In case of an up count operation, the TMENC10 count value is incremented (+1).
<2> In case of a down count operation, TMENC10 is cleared (0000H), and the INTCM11
interrupt is generated.

12.5.2 Operation in general-purpose timer mode

TMENC10 can perform the following operations in the general-purpose timer mode.

(1) Interval operation

TMENC10 and CM100 always compare their values and the INTCM10 interrupt is generated upon occurrence of a match. TMENC10 is cleared (0000_H) at the count clock following the match.

Furthermore, when one more count clock is input, TMENC10 counts up to 0001_H.

The interval time can be calculated by the following formula.

Interval time = (CM100 value + 1) × TMENC10 count clock rate

Note Interval operation can be selected by setting the TMC10.ENMD bit of to 1.

(2) Free-running operation

TMENC10 performs full count operation from 0000_H to $FFFF_H$, and after the TM1OVF bit of the STATUS10 register is set (1), TMENC10 is cleared and resumes counting.

The free-running cycle can be calculated by the following formula.

Free-running cycle = 65,536 × TMENC10 count clock rate

Note Free-running operation can be selected by setting the TMC10.ENMD bit to 0.

(3) Compare function

TMENC10 connects two compare register (CM100, CM101) channels and two capture/compare register (CC100, CC101) channels.

When the TMENC10 count value and the set value of one of the compare registers match, a match interrupt (INTCM10, INTCM11, INTCC10**Note**, INTCC11**Note**) is output. Particularly in the case of an interval operation, TMENC10 is cleared upon generation of the INTCM10 interrupt.

Note This match interrupt is generated when CC100 and CC101 are set to the compare register mode.

(4) Capture function

TMENC10 connects two capture/compare register (CC100, CC101) channels.

When CC100 and CC101 are set to the capture register mode, the value of TMENC10 is captured in synchronization with the corresponding capture trigger signal.

Furthermore, an interrupt request (INTCC10, INTCC11) is generated by the TICC10, TICC11 input signals.

Table 12-9 Capture Trigger Signal to 16-Bit Capture Register

Capture Register	Capture Trigger Signal
CC100	TICC10
CC101	TICC11

Remark CC100 and CC101 are capture/compare registers. Which of these registers is used is specified with capture/compare control register 1 (CCR10).

The valid edge of the capture trigger is specified by signal edge selection register 10 (SESA10). If both the rising edge and the falling edge are selected as the capture triggers, it is possible to measure the input pulse width from external. If a single edge is selected as the capture trigger, the input pulse cycle can be measured.

(5) PWM output operation

PWM output operation is performed from the TO1 pin by setting TMENC10 to the general-purpose timer mode (CMD bit of the TUM10 register = 0).

The resolution is 16 bits, and the count clock can be selected from among seven internal clocks ($f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, $f_{XX}/64$, $f_{XX}/128$, $f_{XX}/256$, $f_{XX}/512$).

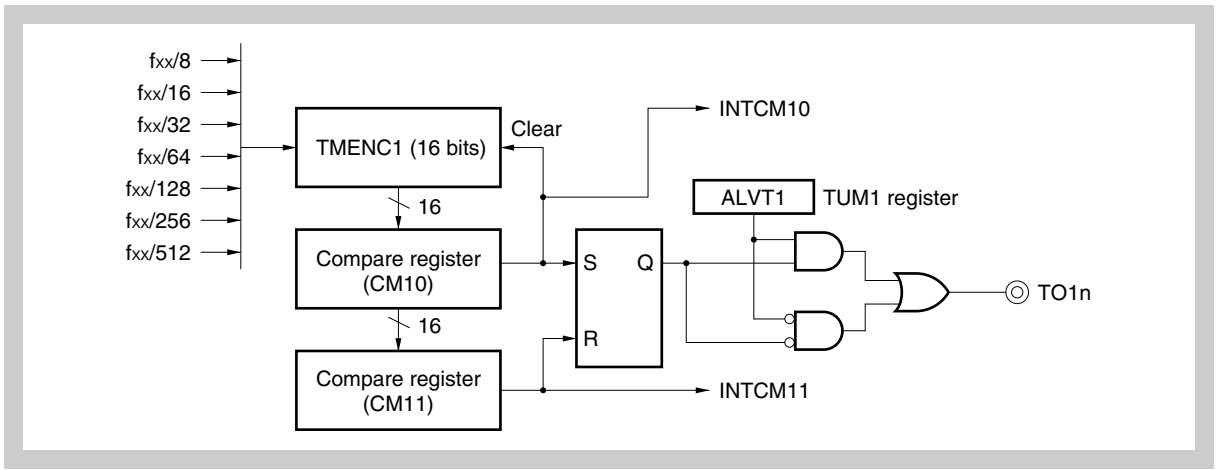


Figure 12-2 TMENC10 block diagram (during PWM output operation)

Remark f_{XX}: Internal system clock

- **Description of operation**

The PWM output cycle is specified by using the compare register CM100. When the value of this register matches the value of TMENC10, the INTCM10 interrupt is generated, and TMENC10 is cleared at the next count clock after the match.

The required PWM output duty is set by using the compare register CM101.

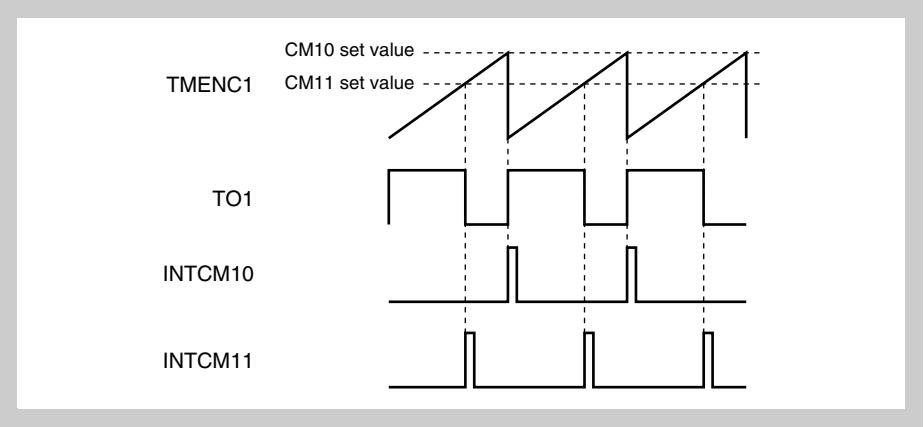


Figure 12-3 PWM signal output example (when ALVT10 bit = 0)

Caution

1. Changing the values of the CM100 and CM101 registers is prohibited during TMENC10 operation (TMC10.TM1CE bit = 1).
2. Changing the value of the TUM.ALVT1 bit is prohibited during TMENC10 operation.
3. PWM signal output is performed from the second PWM cycle after the TM1CE bit is set (1).

12.5.3 Operation in UDC mode

(1) Overview of operation in UDC mode

The count clock input to TMENC10 in the UDC mode (CMD bit of TUM10 register = 1) can only be externally input from the TIUD1 and TCUD1 pins. Up/down count judgment in the UDC mode is determined based on the phase difference of the TIUD1 and TCUD1 pin inputs according to the PRM10 register setting (there is a total of four choices).

Table 12-10 List of Count Operations in UDC Mode

PRM10 Register			Operation Mode	TM1n Operation
PRM102	PRM101	PRM100		
1	0	0	Mode 1	Down count when TCUD1 = high level Up count when TCUD1 = low level
1	0	1	Mode 2	Up count upon detection of valid edge of TIUD1 input Down count upon detection of valid edge of TCUD1 input
1	1	0	Mode 3	Automatic judgment in TCUD1 input level upon detection of valid edge of TIUD1 input
1	1	1	Mode 4	Automatic judgment upon detection of both edges of TIUD1 input and both edges of TCUD1 input

The UDC mode is further divided into two modes according to the TMENC10 clear conditions (count operation is performed only with TIUD1, TCUD1 input in both modes).

(a) UDC mode A (TUM10.CMD = 1, TUM10.MSEL = 0)

The TMENC10 clear source can be selected as only external clear input (TCLR1), a match signal between the TMENC10 count value and the CM100 set value during up count operation, or logical sum (OR) of the two signals, using bits CLR1 and CLR0 of the TMC10 register.

TMENC10 can transfer (reload) the value of CM100 upon occurrence of TMENC10 underflow, when the RLEN bit of the TMC10 register is set (1).

(b) UDC mode B (TUM10.CMD = 1, TUM10.MSEL = 1)

The status of TMENC10 after match of the TMENC10 count value and CM100 set value is as follows.

<1> In case of an up count operation, TMENC10 is cleared (0000_H), and the INTCM10 interrupt is generated.
<2> In case of a down count operation, the TMENC10 count value is decremented (-1).

The status of TMENC10 after match of the TMENC10 count value and CM101 set value is as follows.

<1> In case of an up count operation, the TMENC10 count value is incremented (+1).
<2> In case of a down count operation, TMENC10 is cleared (0000_H), and the INTCM11 interrupt is generated.

(2) Up/down count operation in UDC mode

TMENC10 up/down count judgment in the UDC mode is determined based on the phase difference of the TIUD1 and TCUD1 pin inputs according to the PRM register setting.

(a) Mode 1 (PRM10[2:0] = 100_B)

In mode 1, the following count operations are performed based on the level of the TCUD1 pin upon detection of the valid edge of the TIUD1 pin.

- TMENC10 down count operation when TCUD1 pin = high level
- TMENC10 up count operation when TCUD1 pin = low level

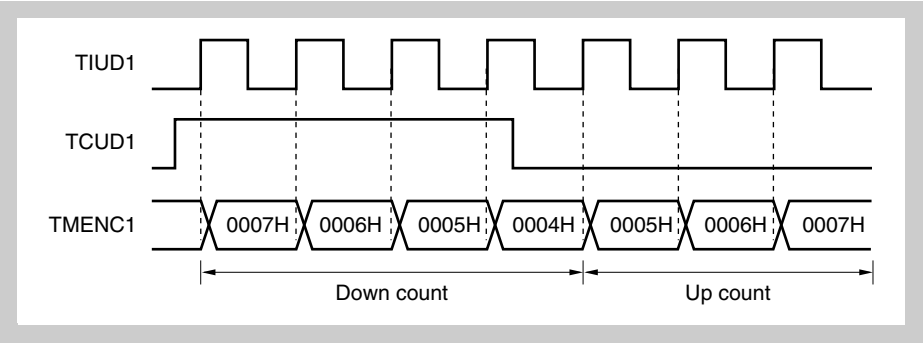


Figure 12-4 UDC operation in mode 1 (when rising edge is specified as valid edge of TIUD1 input pin)

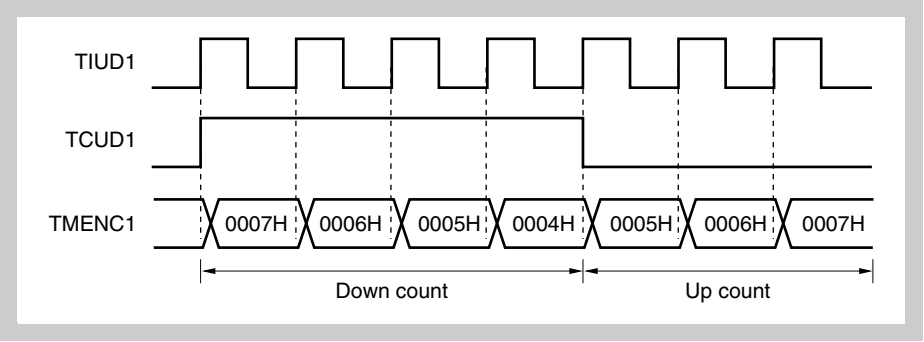


Figure 12-5 UDC operation in mode 1 (when rising edge is specified as valid edge of TIUD1 input pin): in case of simultaneous TIUD1, TCUD1 pin edge timing

(b) Mode 2 (PRM10[2:0] = 101_B)

The count conditions in mode 2 are as follows.

- TMENC10 up count upon detection of valid edge of TIUD1 pin
- TMENC10 down count upon detection of valid edge of TCUD1 pin

Caution If the count clock is simultaneously input to the TIUD1 pin and the TCUD1 pin, count operation is not performed and the immediately preceding value is held.

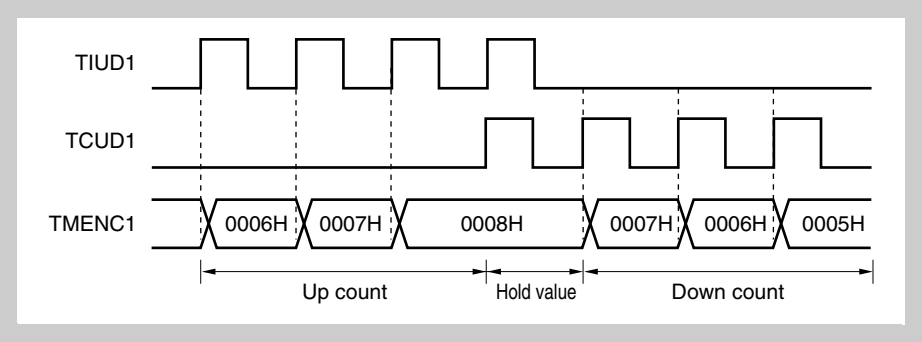


Figure 12-6 UDC operation in mode 2 (when rising edge is specified as valid edge of TIUD1, TCUD1 input pins)

(c) Mode 3 (PRM10[2:0] = 110_B)

In mode 3, when two signals 90 degrees out of phase are input to the TIUD1 and TCUD1 pins, the level of the TCUD1 pin is sampled at the timing of the valid edge of the TIUD1 pin (refer to *Figure 12-7*).

If the TCUD1 pin level sampled at the valid edge timing of the TIUD1 pin is low, TMENC10 counts down.

If the TCUD1 pin level sampled at the valid edge timing of the TIUD1 pin is high, TMENC10 counts up.

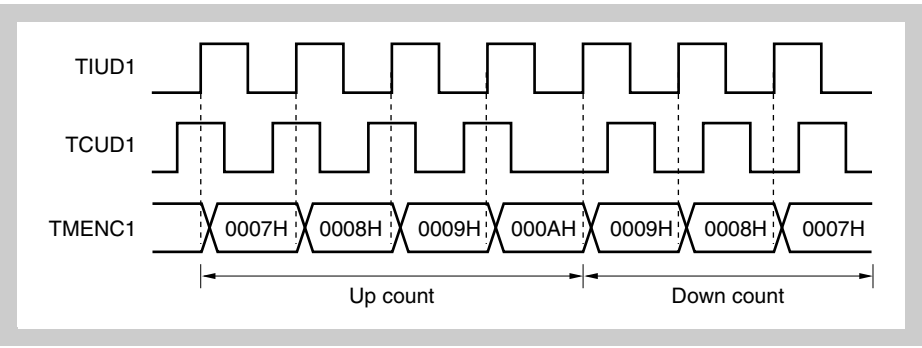


Figure 12-7 UDC operation in mode 3 (when rising edge is specified as valid edge of TIUD1 input pin)

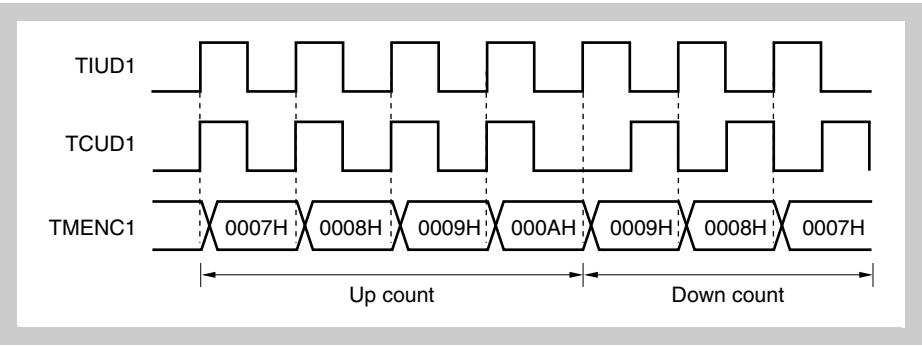


Figure 12-8 UDC operation in mode 3 (when rising edge is specified as valid edge of TIUD1 input pin): in case of simultaneous TIUD1, TCUD1 pin edge timing

(d) Mode 4 (PRM10[2:0] = 111_B)

In mode 4, when two signals out of phase are input to the TIUD1 and TCUD1 pins, up/down operation is automatically judged and counting is performed according to the timing shown in *Figure 12-9*.

In mode 4, counting is executed at both the rising and falling edges of the two signals input to the TIUD1 and TCUD1 pins. Therefore, TMENC10 counts four times per cycle of an input signal (× 4 count).

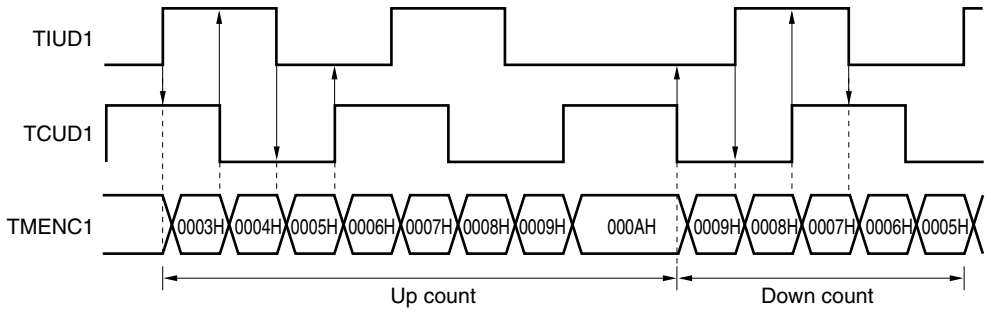


Figure 12-9 UDC operation in mode 4

Caution

1. When mode 4 is specified as the operation mode of TMENC10, the valid edge specifications for pins TIUD1 and TCUD1 are not valid.
2. If the TIUD1 pin edge and TCUD1 pin edge are input simultaneously in mode 4, TMENC10 continues the same count operation (up or down) it was performing immediately before the input.

(3) Operation in UDC mode A

(a) Interval operation

The operations at the count clock following a match of the TMENC10 count value and the CM100 set value are as follows.

- In case of up count operation: TMENC10 is cleared (0000_H) and the INTCM10 interrupt is generated.
- In case of down count operation: The TMENC10 count value is decremented (-1) and the INTCM10 interrupt is generated.

Note The interval operation can be combined with the transfer operation.

(b) Transfer operation

The operations at the next count clock after the count value of TMENC10 becomes 0000H during TMENC10 count down operation are as follows.

- In case of down count operation: The data held in CM100 is transferred.
- In case of up count operation: The TMENC10 count value is incremented (+1).

Note
1. Transfer enable/disable can be set with the TMC10.RLEN bit.
2. The transfer operation can be combined with the interval operation.

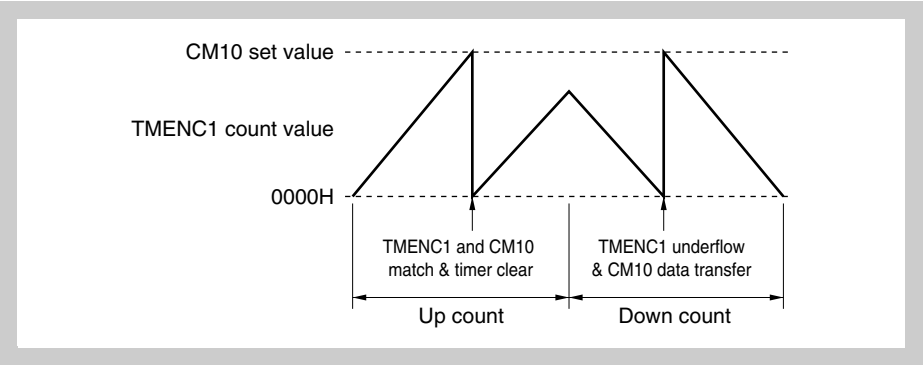


Figure 12-10 Example of TMENC10 operation when interval operation and transfer operation are combined

(c) Compare function

TM1n connects two compare register (CM100, CM101) channels and two capture/compare register (CC100, CC101) channels.

When the TMENC10 count value and the set value of one of the compare registers match, a match interrupt (INTCM10, INTCM11, INTCC10**Note**, INTCC11**Note**) is output.

Note This match interrupt is generated when CC100 and CC101 are set to the compare register mode.

(d) Capture function

TMENC10 connects two capture/compare register (CC100, CC101) channels.

When CC100 and CC101 are set to the capture register mode, the value of TMENC10 is captured in synchronization with the corresponding capture trigger signal.

When the TMENC10 is set to the capture register mode, a capture interrupt (INTCC10, INTCC11) is generated upon detection of the valid edge.

(4) Operation in UDC mode B

(a) Basic operation

The operations at the next count clock after the count value of TMENC10 and the CM100 set value match when TMENC10 is in UDC mode B are as follows.

- In case of up count operation: TMENC10 is cleared (0000H) and the INTCM10 interrupt is generated.
- In case of down count operation: The TMENC10 count value is decremented (-1).

The operations at the next count clock after the count value of TMENC10 and the CM101 set value match when TMENC10 is in UDC mode B are as follows.

- In case of up count operation: The TMENC10 count value is incremented (+1).
- In case of down count operation: TMENC10 is cleared (0000H) and the INTCM11 interrupt is generated.

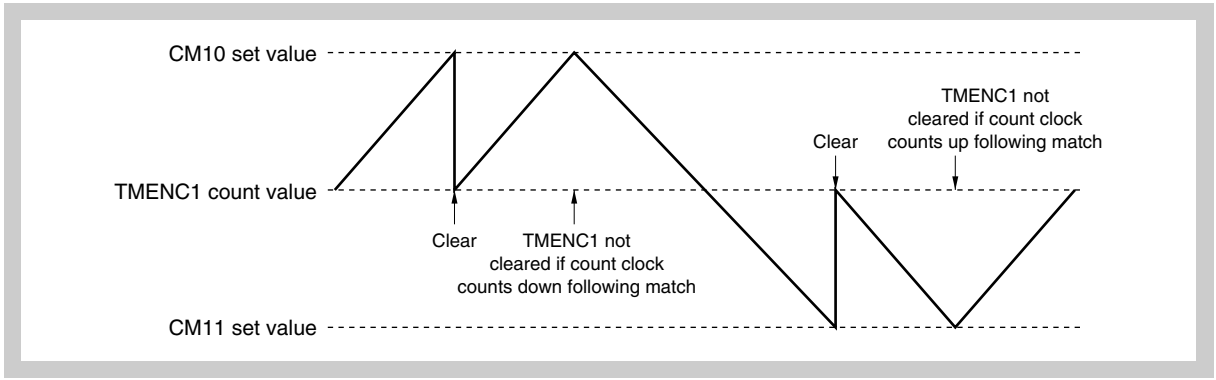


Figure 12-11 Example of TMENC10 operation in UDC mode

(b) Compare function

TMENC10 connects two compare register (CM100, CM101) channels and two capture/compare register (CC100, CC101) channels.

When the TMENC10 count value and the set value of one of the compare registers match, a match interrupt (INTCM10 (only during up count operation), INTCM11 (only during down count operation), INTCC10[Note], INTCC11[Note]) is output.

Note This match interrupt is generated when CC100 and CC101 are set to the compare register mode.

(c) Capture function

TMENC10 connects two capture/compare register (CC100, CC101) channels.

When CC100 and CC101 are set to the capture register mode, the value of TMENC10 is captured in synchronization with the corresponding capture trigger signal.

When the TMENC10 is set to the capture register mode, a capture interrupt (INTCC10, INTCC11) is generated upon detection of the valid edge.

12.6 Supplementary Description of Internal Operation

12.6.1 Clearing of count value in UDC mode B

When TMENC10 is in UDC mode B, the count value clear operation is as follows.

- In case of up count operation: TMENC10 is cleared (0000_H) upon match with CM100.
- In case of down count operation: TMENC10 is cleared (0000_H) upon match with CM101

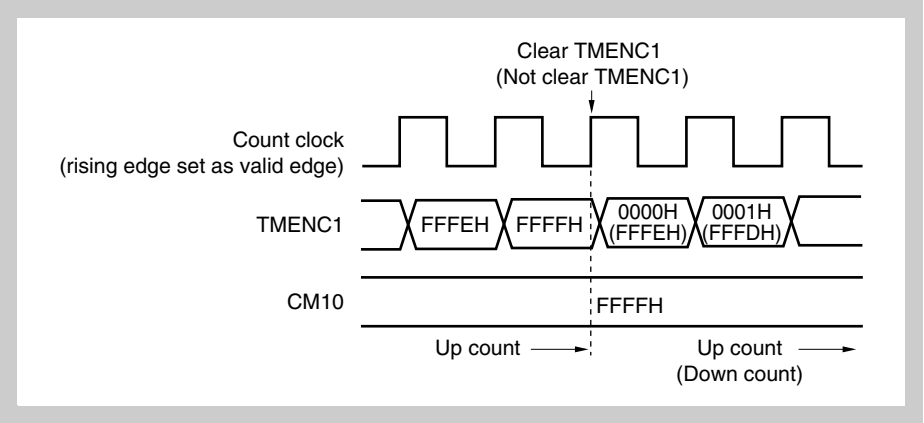


Figure 12-12 Clear operation upon match with CM100 during TMENC10 up count operation

Note Items between parentheses in the above figure apply to down count operation.

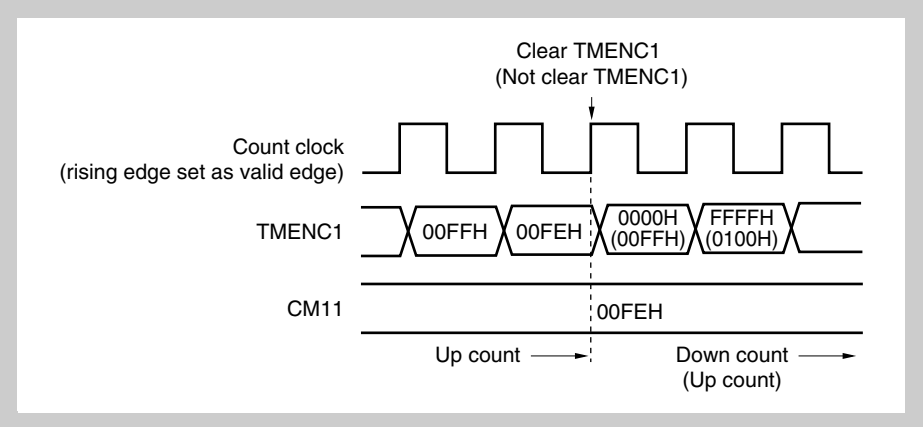


Figure 12-13 Clear operation upon match with CM100 during TMENC10 down count operation

Note Items between parentheses in the above figure apply to up count operation.

12.6.2 Clearing of count value upon occurrence of compare match

The internal operation during TMENC10 clear operation upon occurrence of a compare match is as follows.

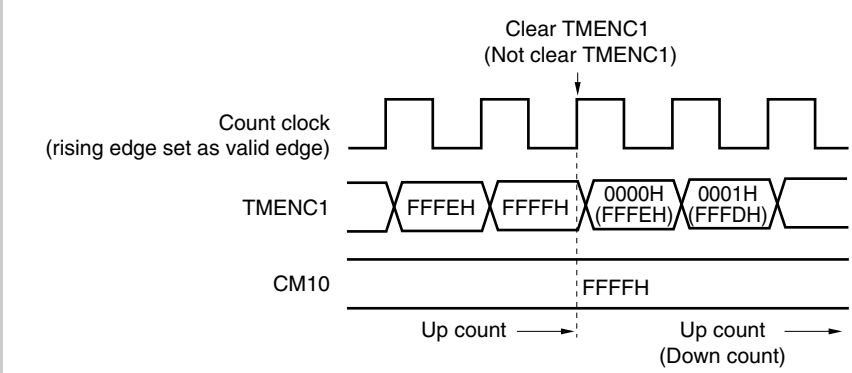


Figure 12-14 Count value clear operation upon compare match

Note Items between parentheses in the above figure apply to down count operation.

Caution The operations at the next count clock after the count value of TMENC10 and the CM100 set value match are as follows.

- In case of up count: Clear operation is performed.
- In case of down count: Clear operation is not performed.

12.6.3 Transfer operation

The internal operation during TMENC10 transfer operation is as follows.

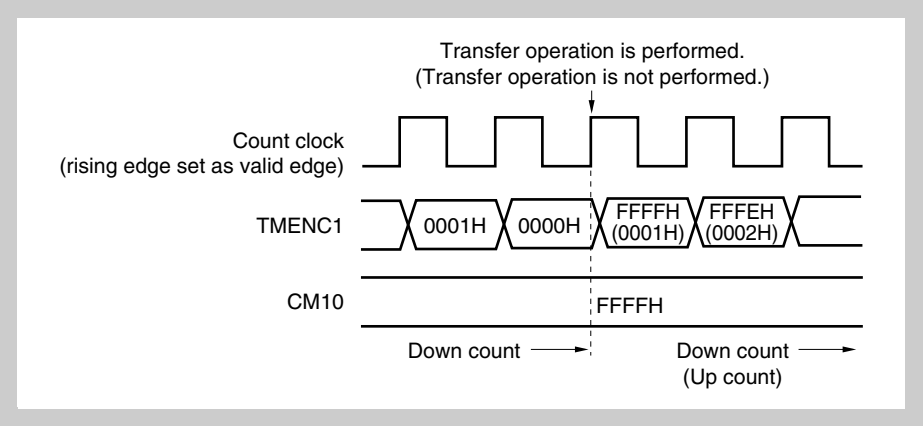


Figure 12-15 Internal operation during transfer operation

Note Items between parentheses in the above figure apply to up count operation.

Caution The count operations after the TMENC10 count value becomes 0000_H are as follows.

- In case of down count: Transfer operation is performed.
- In case of up count: Transfer operation is not performed.

12.6.4 Interrupt signal output upon compare match

An interrupt signal is output when the count value of TMENC10 matches the set value of the CM100, CM101, CC100[Note], or CC101[Note] register. The interrupt generation timing is as follows.

Note When CC100 and CC101 are set to the compare register mode.

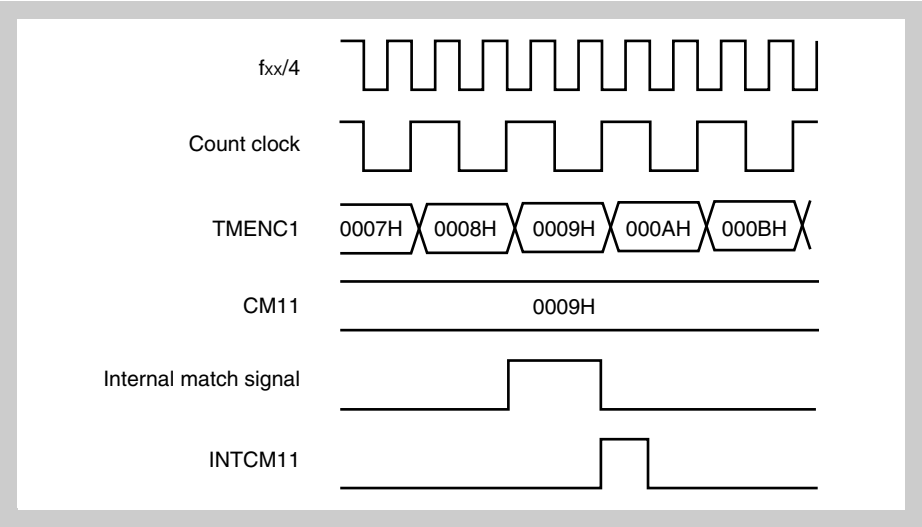


Figure 12-16 Interrupt output upon compare match (CM101 with operation in general-purpose timer mode and count clock set to f_{XX}/8)

An interrupt signal such as illustrated in *Figure 12-16* is output at the next count following match of the TMENC10 count value and the set value of a corresponding compare register.

12.6.5 TM1UBD flag operation

In the UDC mode (TUM10.CMD = 1), the STATUS1.TM1UBD flag changes as follows during TMENC10 up/down count operation at every internal operation clock.

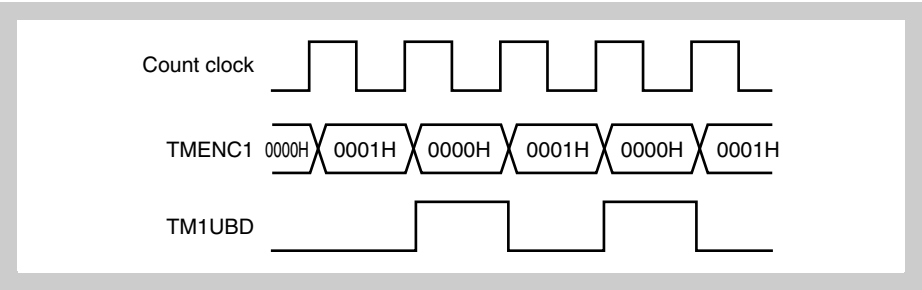


Figure 12-17 TM1UBDn flag operation

Chapter 13 Auxiliary Frequency Output Function (AFO)

13.1 Features

- Frequency up to 8 Mbps
- Programmable frequency output
- Interval timer function
- Interrupt request signal (INTBRG2)

13.2 Configuration

The AFO function includes the following hardware.

Table 13-1 AFO Configuration

Item	Configuration
Control registers	Prescaler mode registers 2 (PRSM2)
	Prescaler compare registers 2 (PRSCM2)

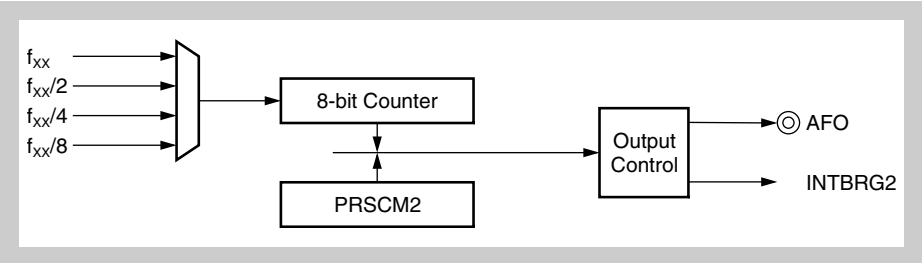


Figure 13-1 Block diagram of Auxiliary Frequency Output function

13.3 Control Registers

(1) Prescaler mode register 2 (PRSM2)

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 $FFFFFDE0_H$

Initial Value 00_H. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
PRSM2	0	0	0	CE2	0	0	BGCS21	BGCS20
	R	R	R	R/W	R	R	R/W	R/W

Caution
1. Do not rewrite the PRSM2 register during operation.
2. Set the BGCS21, BGCS20 bits before setting the CE2 bit to 1.

Table 13-2 PRSM2 register contents

Bit position	Bit name	Function
4	CE2	Controls the baud rate generator output. 0: Baud rate generator disabled. 1: Baud rate generator enabled.
1, 0	BGCS21, BGCS20	Selects the baud rate generator clock (f_{BGCS2}).. **BGCS21 / BGCS20 / Baud Rate Generator Clock Selection (f_{BGCS2}) / Setting Value (k)** 0 / 0 / f_{XX} / 0 0 / 1 / $f_{XX}/2$ / 1 1 / 0 / $f_{XX}/4$ / 2 1 / 1 / $f_{XX}/8$ / 3

(2) Prescaler compare registers 2 (PRSCM2)

The PRSCM2 register is an 8-bit compare register.

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

Address $FFFFFDE1_H$

Initial Value 00_H. This register is cleared by any reset.

PRSCM2

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

Caution

1. Do not rewrite the PRSCM2 register during operation.
2. Set the PRSCM2 register before setting the PRSM2.CE2 bit to 1.
3. Do not set the AFO clock to a higher frequency than 8 MHz.

Table 13-3 PRSM2 register contents

Bit position	Bit name	Function
7 to 0	PRSCM2[7:0]	Specifies the AFO clock (f_{AFO})..

PRSCM 27	PRSCM 26	PRSCM 25	PRSCM 24	PRSCM 23	PRSCM 22	PRSCM 21	PRSCM 20	AFO Clock (f_{AFO})	N
0	0	0	0	0	0	0	0	f_{BGSC2}/512	256
0	0	0	0	0	0	0	1	f_{BGSC2}/2	1
0	0	0	0	0	0	1	0	f_{BGSC2}/4	2
0	0	0	0	0	0	1	1	f_{BGSC2}/6	3
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	1	1	1	0	1	f_{BGSC2}/506	253
1	1	1	1	1	1	1	0	f_{BGSC2}/508	254
1	1	1	1	1	1	1	1	f_{BGSC2}/510	255

Note: f_{BGCS2}: Clock frequency selected by the PRSM2.BGCS2[1:0].

13.4 Operation

13.4.1 Auxiliary frequency output

The auxiliary frequency output (AFO) is enabled as soon as the shared port (P75) is set into control output mode by setting bit 5 of the PM7 register to 0 and bit 5 of the PMC7 register to 1.

13.4.2 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.

$$f_{AFO} = \frac{f_{BGCS2}}{N \times 2} = \frac{f_{XX}}{2^k \times N \times 2}$$

Remarks

1. f_{AFO}: AFO clock
2. f_{BGCS2}: Clock frequency selected by the BGCS21, BGCS20 bits of the PRSM2 register.
3. f_{XX}: Main clock oscillation frequency
4. k: PRSM2 register setting value ($0 \leq k \leq 3$)
5. N: PRSCMm register setting value
   N = 1 to 255, when PRSCM2 = 01H to FFH, or
   N = 256, when PRSCM2 = 00H.

13.4.3 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 14 A/D Converter

The microcontroller has two instances of this A/D converter, ADC0 and ADC1.

Throughout this chapter, the individual instances of ADC are identified by “n” (n = 0, 1), for example, ADMn0 for the A/D converter n mode register 0.

14.1 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

14.2 Configuration

The A/D converter of the V850E/PH2 adopts the successive approximation method, and uses A/D converter n mode registers 0, 1, 2 (ADMn0, ADMn1, ADMn2), and the A/D conversion result register (ADCRn0 to ADCRn9) to perform A/D conversion operations.

(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) (m = 0 to 9)

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).

RESET input makes this register undefined.

(6) A/D conversion result register for DMA transfer (ADDMAn)

ADDMAn is a 16-bit register that holds the last 10-bit A/D conversion result and an over rung flag for indicating a DMA transfer failure.

(7) ANIn0 to ANIn9 pins (n = 0, 1)

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 (n = 0, 1)

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 (n = 0, 1)

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).

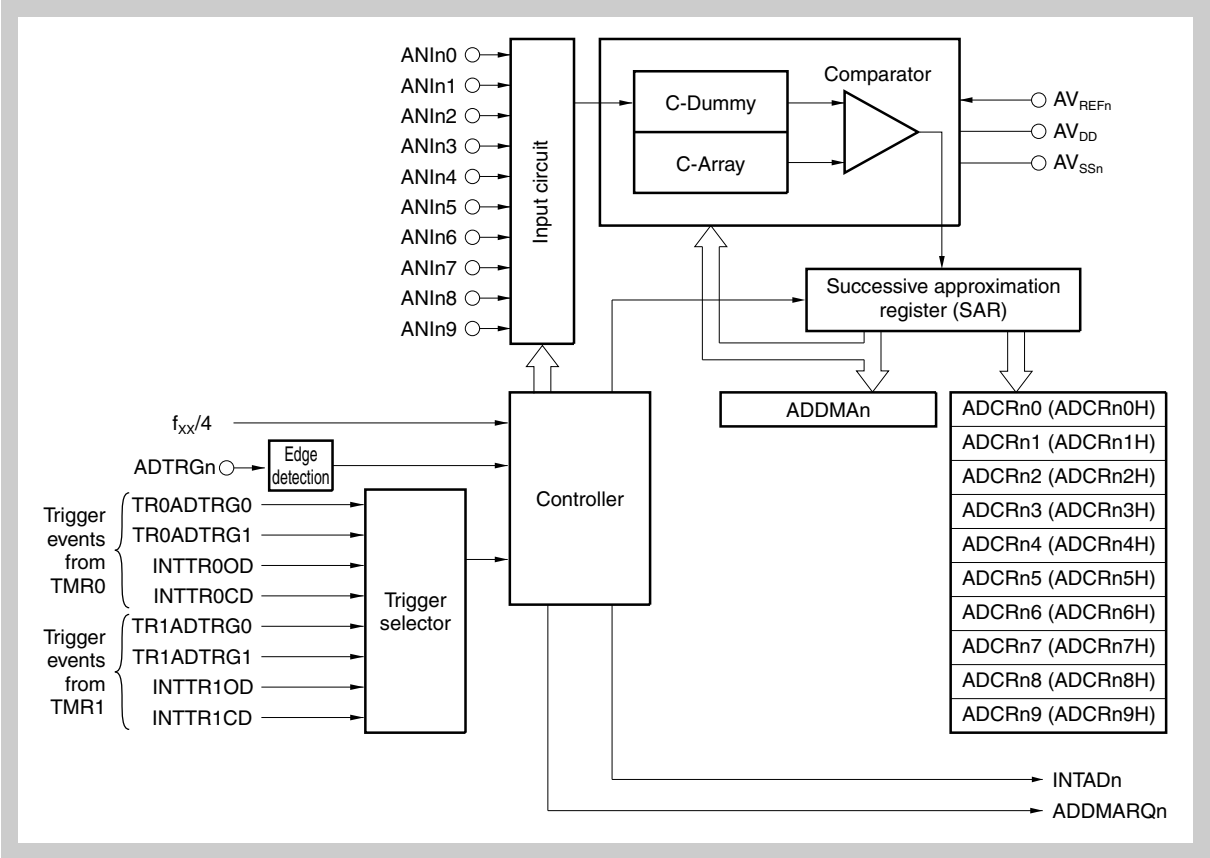


Figure 14-1 Block Diagram of A/D Converter (ADCn)

Remark f_{XX}: Main clock

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.

14.3 Control Registers

(1) A/D converter n mode register 0 (ADMn0)

The ADMn0 register is an 8-bit register that specifies the operation mode, and executes conversion operations.

Access This register can be read or written in 8-bit units.However, bit 6 can only be read. Writing this bit is ignored.

Address ADM00: FFFFF200H ADM10: FFFFF240H

Initial value 00H. This register is cleared by any reset.

Caution

1. 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 the A/D trigger mode, 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.
2. Changing the setting of the BSn and MSn bits is prohibited while A/D conversion is enabled (ADCEn = 1).
3. When data is written to the ADMn0 register during an A/D conversion operation, the conversion operation is initialized and conversion is executed from the beginning.

ADMn0

7	6	5	4	3	2	1	0
ADCEn	ADCSn	BSn	MSn	0	0	0	0
R/W	R	R/W	R/W	R	R	R	R

Table 14-1 ADMn0 register contents

Bit position	Bit name	Function
7	ADCEn	A/D Conversion Operation Control of ADCn 0: Disables A/D conversion operation of ADCn 1: Enables A/D conversion operation ADCn
6	ADCSn	A/D Conversion Status Flag of ADCn 0: A/D conversion of ADCn is stopped 1: A/D conversion of ADCn is operating
5	BSn	ADCn Buffer Mode Specification 0: 1-buffer mode 1: 4-buffer mode
4	MSn	ADCn Operation Mode Specification 0: Scan mode 1: Select mode

(2) A/D converter n mode register 1 (ADMn1)

The ADMn1 register is an 8-bit register that specifies the conversion operation time and trigger mode.

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

Address ADM01: FFFFF201H ADM11: FFFFF241H

Initial value 00H. This register is cleared by any reset.

Caution

1. Changing the setting of the EGAn1, EGAn0, and FRn3 to FRn0 bits is prohibited while A/D conversion is enabled (ADMn0.ADCEn = 1).
2. When data is written to the ADMn1 register during an A/D conversion operation, the conversion operation is initialized and conversion is executed from the beginning.
3. When the trigger mode (TRGn1 and TGRn0 bits) is changed midway, A/D conversion can be started immediately without having to secure the A/D stabilization time by re-setting the ADCE bit to 1.

ADMn1

7	6	5	4	3	2	1	0
EGAn1	EGAn0	TRGn1	TRGn0	FRn3	FRn2	FRn1	FRn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14-2 ADMn1 register contents (1/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7, 6</td><td>EGAn1,
EGAn0</td><td>Specifies the valid edge of the external trigger input (ADTRGn).
<table>
<tr><th>EGAn1</th><th>EGAn0</th><th>External Trigger Input (ADTRGn) Valid Edge Setting</th></tr>
<tr><td>0</td><td>0</td><td>No edge detected (does not operate as external trigger)</td></tr>
<tr><td>0</td><td>1</td><td>Falling edge detected</td></tr>
<tr><td>1</td><td>0</td><td>Rising edge detected</td></tr>
<tr><td>1</td><td>1</td><td>Both edges, falling and rising edge detected</td></tr>
</table></td></tr>
<tr><td>5, 4</td><td>TRGn1,
TRGn0</td><td>Specifies the ADCn trigger mode.
<table>
<tr><th>TRGn1</th><th>TRGn0</th><th>ADCn Trigger Mode</th></tr>
<tr><td>0</td><td>0</td><td>A/D trigger mode</td></tr>
<tr><td>0</td><td>1</td><td>Timer trigger mode</td></tr>
<tr><td>1</td><td>0</td><td>External trigger mode</td></tr>
<tr><td>1</td><td>1</td><td>Setting prohibited</td></tr>
</table></td></tr>
</table>

Table 14-2 ADMn1 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3 to 0</td><td>FRn3 to FRn0</td><td>Specifies the number of conversion clocks of ADCn.
<table>
<tr><th rowspan="2">FRn3</th><th rowspan="2">FRn2</th><th rowspan="2">FRn1</th><th rowspan="2">FRn0</th><th rowspan="2">Number of conversion clocks</th><th colspan="2">Conversion Operation Time[a]</th></tr>
<tr><th>f_{XX} = 64 MHz</th><th>A/D Stabilization Time[b]</th></tr>
<tr><td>0</td><td>0</td><td></td><td></td><td>128</td><td>2.0 μs</td><td>$64/f_{XX}$</td></tr>
<tr><td>0</td><td>1</td><td></td><td></td><td>256</td><td>4.0 μs</td><td>$128/f_{XX}$</td></tr>
<tr><td>1</td><td>0</td><td></td><td></td><td>384</td><td>6.0 μs</td><td>$160/f_{XX}$</td></tr>
<tr><td>1</td><td>1</td><td></td><td></td><td>512</td><td>8.0 μs</td><td>$160/f_{XX}$</td></tr>
<tr><td colspan="5">Others than above</td><td colspan="2">Setting prohibited</td></tr>
</table>
a) The conversion operation time has to be set within the range of 2 to 10 μs.
b) The 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.</td></tr>
</table>

(3) A/D converter n mode register 2 (ADMn2)

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 or written in 8-bit units.

Address ADM02: FFFFF202H ADM12: FFFFF242H

Initial value 00H. This register is cleared by any reset.

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. When data is written to the ADMn2 register during an A/D conversion operation, the conversion operation is initialized and conversion is executed from the beginning.

	7	6	5	4	3	2	1	0
ADMn2	0	0	0	0	ANISn3	ANISn2	ANISn1	ANISn0
	R	R	R	R	R/W	R/W	R/W	R/W

Table 14-3 ADMn1 register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3 to 0</td><td>ANISn3 to ANISn0</td><td>Specifies the analog input pins for conversion of ADCn.
<table>
<tr><th rowspan="2">ANISn3</th><th rowspan="2">ANISn2</th><th rowspan="2">ANISn1</th><th rowspan="2">ANISn0</th><th colspan="2">Analog Input Pins for A/D Conversion</th></tr>
<tr><th>Select Mode</th><th>Scan Mode</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>ANIn0</td><td>ANIn0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>ANIn1</td><td>ANIn0, ANIn1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>ANIn2</td><td>ANIn0 to ANIn2</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>ANIn3</td><td>ANIn0 to ANIn3</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>ANIn4</td><td>ANIn0 to ANIn4</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>ANIn5</td><td>ANIn0 to ANIn5</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>ANIn6</td><td>ANIn0 to ANIn6</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>ANIn7</td><td>ANIn0 to ANIn7</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>ANIn8</td><td>ANIn0 to ANIn8</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>ANIn9</td><td>ANIn0 to ANIn9</td></tr>
<tr><td colspan="4">Others than above</td><td colspan="2">Setting prohibited</td></tr>
</table>
</td></tr>
</table>

(4) A/D converter n trigger source select register (ADTRSELn)

The ADTRSELn register is an 8-bit register that specifies the timer trigger signal in the timer trigger mode (ADMn1.TRGn[1:0] = 01B).

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

Address ADTRSEL0: FFFFF270H ADTRSEL1: FFFFF272H

Initial value 00H. This register is cleared by any reset.

Caution Before changing the setting of the ADTRSELn register, stop the A/D conversion operation (by clearing the ADCEn bit of the ADMn0 register 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).

	7	6	5	4	3	2	1	0
ADTRSELn	0	0	0	0	TSELn3	TSELn2	TSELn1	TSELn0
	R	R	R	R	R/W	R/W	R/W	R/W

Table 14-4 ADTRSELn register contents

Bit position	Bit name	Function
3 to 0	ANISn3 to ANISn0	Selects the trigger source of ADCn in timer trigger mode.

TSELn3	TSELn2	TSELn1	TSELn0	Trigger Source in Timer Trigger Mode
0	0	0	0	None. All trigger sources are ignored.
0	0	0	1	TR0ADTRG0 signal (from TMR0)
0	0	1	0	TR0ADTRG1 signal (from TMR0)
0	0	1	1	TR1ADTRG0 signal (from TMR1)
0	1	0	0	TR1ADTRG1 signal (from TMR1)
0	1	0	1	INTTR0OD interrupt (from TMR0)
0	1	1	0	INTTR0CD interrupt (from TMR0)
0	1	1	1	INTTR1OD interrupt (from TMR1)
1	0	0	0	INTTR1CD interrupt (from TMR1)
Others than above				Setting prohibited

(5) A/D conversion result registers n0 to n9, n0H to n9H (ADCRn0 to ADCRn9, ADCRn0H to ADCRn9H)

The ADCRnm register is a 10-bit register holding the A/D conversion results (m = 0 to 9).

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 upper 8 bits of the conversion result is specified

When reading the 10-bit data of the A/D conversion results from the ADCRnm register, only the upper 10 bits are valid and the lower 6 bits are always read as 0.

Address

ADCR00:	FFFFF210H	ADCR00H:	FFFFF211H
ADCR01:	FFFFF212H	ADCR01H:	FFFFF213H
ADCR02:	FFFFF214H	ADCR02H:	FFFFF215H
ADCR03:	FFFFF216H	ADCR03H:	FFFFF217H
ADCR04:	FFFFF218H	ADCR04H:	FFFFF219H
ADCR05:	FFFFF21AH	ADCR05H:	FFFFF21BH
ADCR06:	FFFFF21CH	ADCR06H:	FFFFF21DH
ADCR07:	FFFFF21EH	ADCR07H:	FFFFF21FH
ADCR08:	FFFFF220H	ADCR08H:	FFFFF221H
ADCR09:	FFFFF222H	ADCR09H:	FFFFF223H
ADCR10:	FFFFF250H	ADCR10H:	FFFFF251H
ADCR11:	FFFFF252H	ADCR11H:	FFFFF253H
ADCR12:	FFFFF254H	ADCR12H:	FFFFF255H
ADCR13:	FFFFF256H	ADCR13H:	FFFFF257H
ADCR14:	FFFFF258H	ADCR14H:	FFFFF259H
ADCR15:	FFFFF25AH	ADCR15H:	FFFFF25BH
ADCR16:	FFFFF25CH	ADCR16H:	FFFFF25DH
ADCR17:	FFFFF25EH	ADCR17H:	FFFFF25FH
ADCR18:	FFFFF260H	ADCR18H:	FFFFF261H
ADCR19:	FFFFF262H	ADCR19H:	FFFFF263H

Initial value undefined

ADCRnm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

ADCRnmH

7	6	5	4	3	2	1	0
AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
R	R	R	R	R	R	R	R

The correspondence between each analog input pin and the ADCRnm register is shown in *Table 14-5* below.

Table 14-5 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	ADCRn8 to ADCRn9, ADCRn8H to ADCRn9H
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 (ADCRnm)) is as follows:

$$ADCR = INT\left(\frac{V_{IN}}{AV_{REF}} \times 1024 + 0{,}5\right)$$

or,

$$(ADCR - 0{,}5) \times \frac{AV_{REF}}{1024} \leq V_{IN} < (ADCR + 0{,}5) \times \frac{AV_{REF}}{1024}$$

INT( ): Function that returns the integer value

V_{IN}: Analog input voltage

AV_{REF}: AV_{REF} pin voltage

ADCR: Value of A/D conversion result register (ADCRnm)

Figure shows the relationship between the analog input voltage and the A/D conversion results.

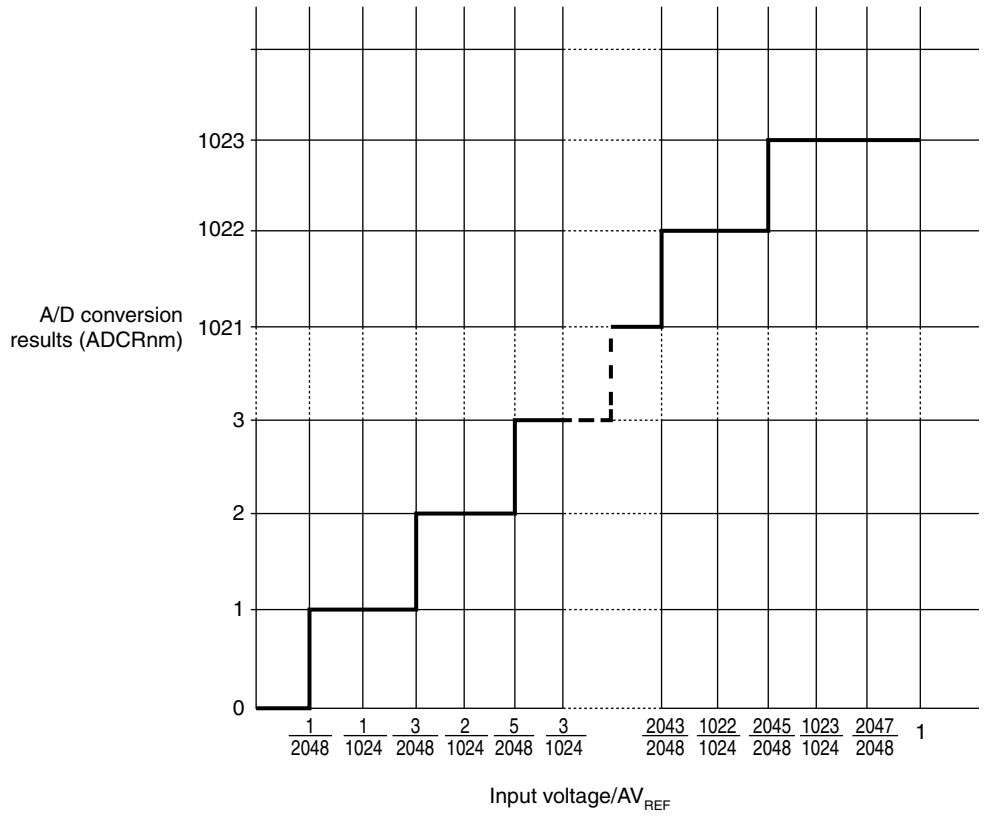


Figure 14-2 Relationship Between Analog Input Voltage and A/D Conversion Results

(6) A/D conversion result register n for DMA (ADDMAn)

The ADDMAn register is a 16-bit register holding the result of the latest A/D conversion operation, and is used for DMA transfer of ADCn results into the internal RAM. It has an overrun detection flag indicating an overrun situation of the DMA transfer mechanism.

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

Address ADDMA0: FFFFF224H ADDMA1: FFFFF264H

Initial value undefined

Caution Do not read the ADDMAn register by CPU during DMA transfer activities. If this register is read by CPU, overflow detection cannot be ensured.

ADDMAn

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDMAn9	ADDMAn8	ADDMAn7	ADDMAn6	ADDMAn5	ADDMAn4	ADDMAn3	ADDMAn2	ADDMAn1	ADDMAn0	0	0	0	0	0	ODFn
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 14-6 ADDMAn register contents

Bit position	Bit name	Function
15 to 6	ADDMAn9 to ADDMAn0	Latest A/D conversion result value (000H to 3FFH).
0	ODFn	Overrun Detection Flag The ODFn flag indicates a DMA transfer failure of the A/D conversion result. 0: No A/D conversion result overrun was detected. 1: At least one A/D conversion result was overrun since the last read of the ADDMAn register **Remark:** The ODFn flag is cleared (0), when the A/D conversion is stopped (ADMn0.ADCEn = 0).

14.4 Operation

14.4.1 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[Note 1] (n = 0, 1).
When the ADMn0.ADCEn bit 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[Note 2] 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 (n = 0, 1), (m = 0 to 9).

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 ADMn0.ADCEn bit is set to 1, the mode changes to the trigger standby state. The A/D conversion operation is started by the trigger signal (ADMn0.ADCSn = 1), and the trigger standby state (ADMn0.ADCSn = 0) is returned when the A/D conversion operation ends.

14.4.2 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 ADMn1registers.

The following table shows the relationship between the operation mode and trigger mode.

Table 14-7 Relationship Between Operation Mode and Trigger Mode

Trigger Mode	Operation Mode		Register Set Value	
			ADMn0	ADMn1
A/D trigger	Select	1 buffer	xx010000B	xx000xxxB
		4 buffers	xx110000B	
	Scan		xx000000B	
Timer trigger	Select	1 buffer	xx010000B	xx010xxxB
		4 buffers	xx110000B	
	Scan		xx000000B	
External trigger	Select	1 buffer	xx010000B	xx100xxxB
		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 ADMn0.TRGn[1:0] bits.

(a) A/D trigger mode

This mode starts the conversion timing of the analog input set to the ANIn0 to ANIn9 pins, and by setting ADMn0.ADCEn = 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.

(b) 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 R (TMR0, TMR1).

The ADTRSELn register specifies the analog input conversion timing by selecting either one of the A/D converter trigger signals (TR0ADTRG0, TR0ADTRG1, TR1ADTRG0, TR1ADTRG1) or one of the top and bottom reversal interrupts (INTTR0CD, INTR0OD, INTTR1CD, INTTR1OD) connected to the 16-bit inverter timer R (TMR0, TMR1).

If the ADMn0.ADCEn bit is set to 1, the A/D converter waits for an event input (TR0ADTRG0, TR0ADTRG1, TR1ADTRG0, TR1ADTRG1, INTTR0CD, INTR0OD, INTTR1CD, or INTTR1OD), and starts conversion when the event occurs (ADMn0.ADCSn = 1). When conversion has finished, the converter waits for an event input again (ADMn0.ADCSn = 0). If data is written to the ADMn0 to ADMn2 registers during conversion, conversion is stopped and then executed from the beginning again.

(c) 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 = 1). When the converter has finished its conversion operation, it waits for an 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 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 ADMn0.BSn and ADMn0.MSn bits.

(a) 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 (m = 0 to 9).

- **1-buffer mode**

In this mode, one analog input specified by the ADMn2 register is converted. The conversion results are stored in the ADCRnm register corresponding to the analog input (ANInm). The ANInm pin and ADCRnm register corresponds 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.

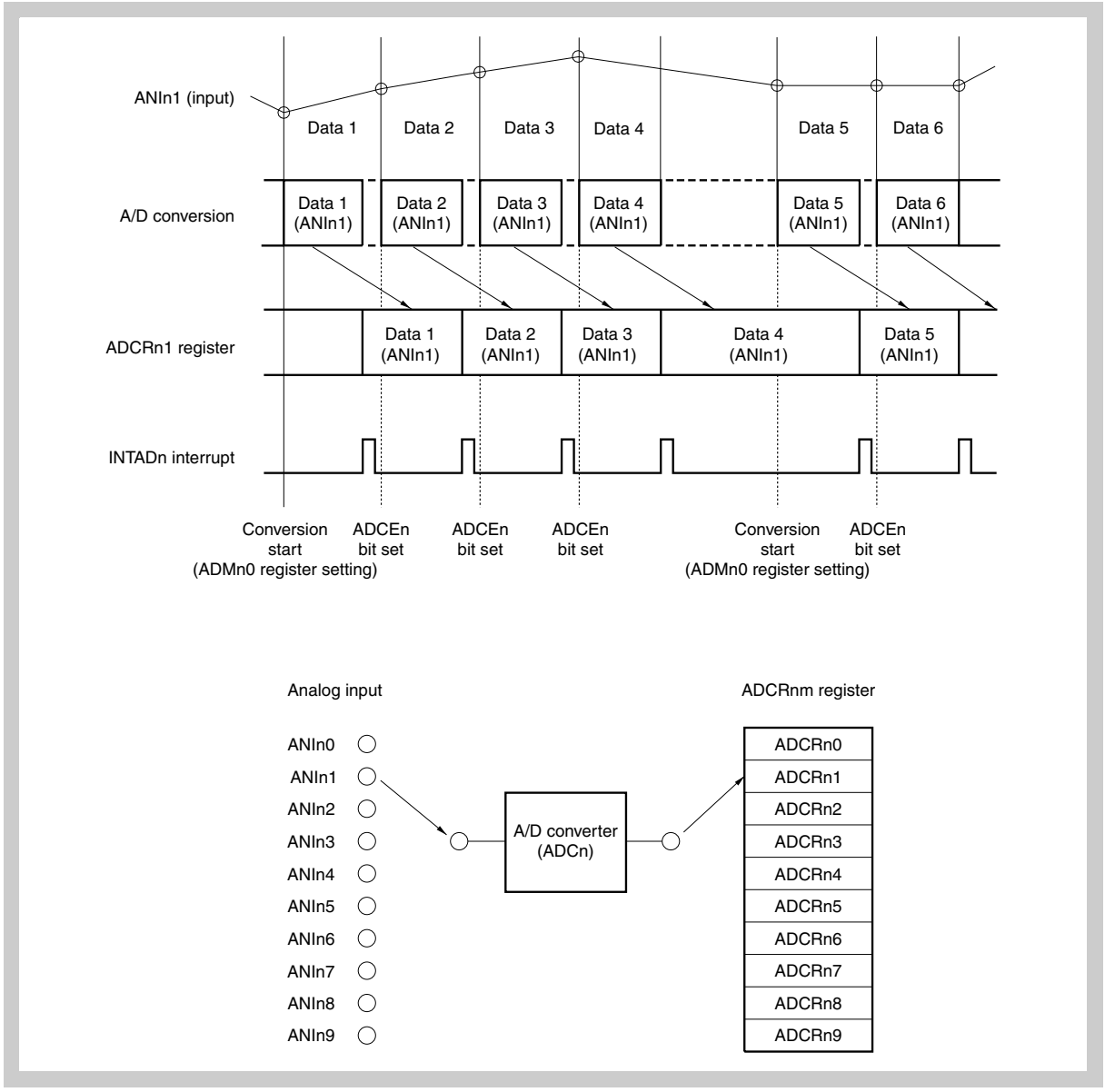


Figure 14-3 Select Mode Operation Timing: 1-Buffer Mode (ANIn1)

- **4-buffer mode**

In this mode, one analog input is A/D converted and the results are stored in the ADCRnm registers. The A/D conversion end interrupt (INTADn) is generated when the four A/D conversions end (m = 0 to 3 when one of the analog input channels ANIn0 to ANIn3 is specified, m = 4 to 7 when one of analog input channels ANIn4 to ANIn7 is specified, and m = 8 to 9 when one of the analog input channels ANIn8 or ANIn9 is specified).

After conversion has finished, the next conversion operation is repeated, unless the ADCEn bit of the ADM0 register is cleared to 0.

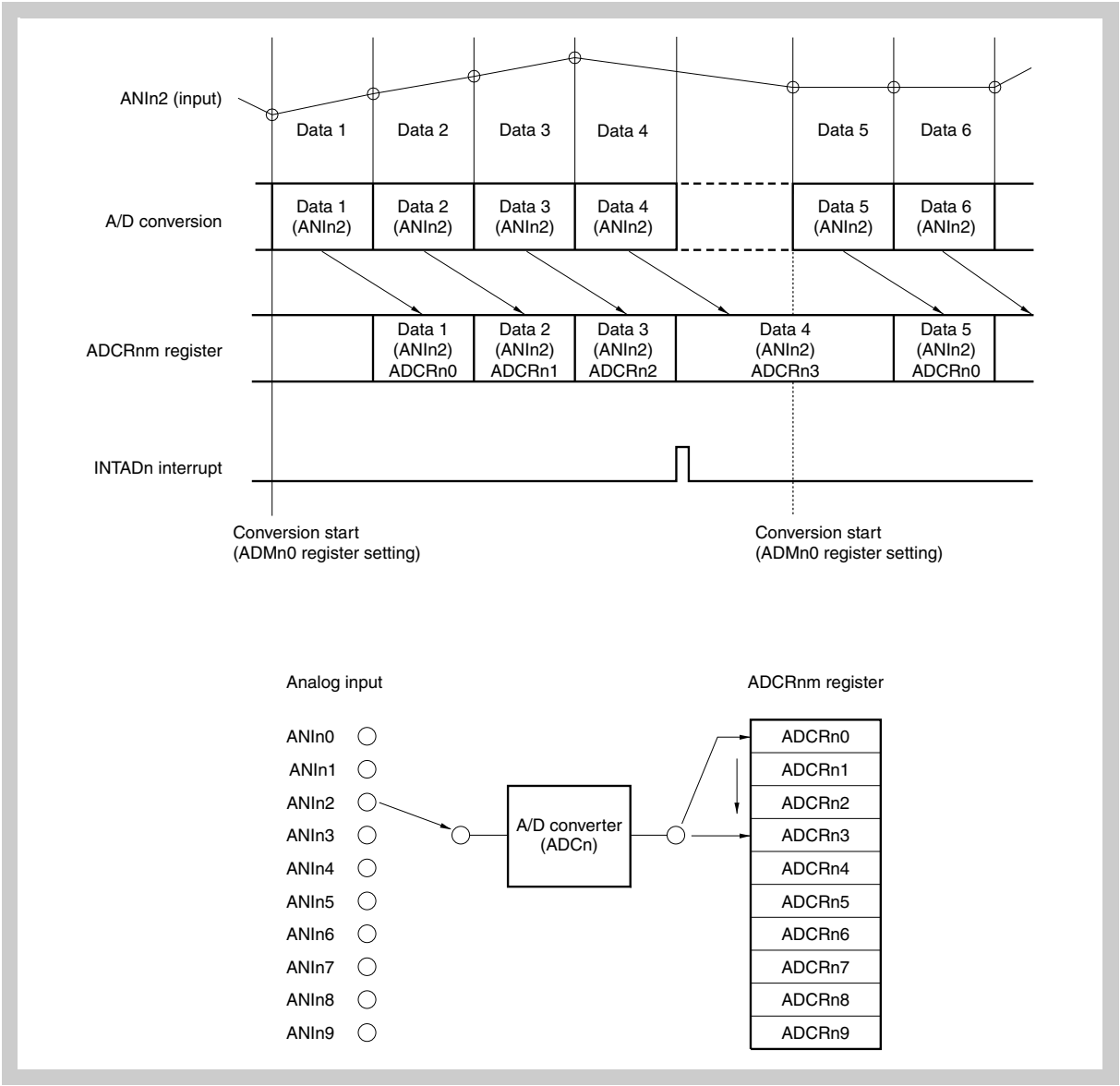


Figure 14-4 Select Mode Operation Timing: 4-Buffer Mode (ANIn2)

(b) 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 (ANInm). 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.

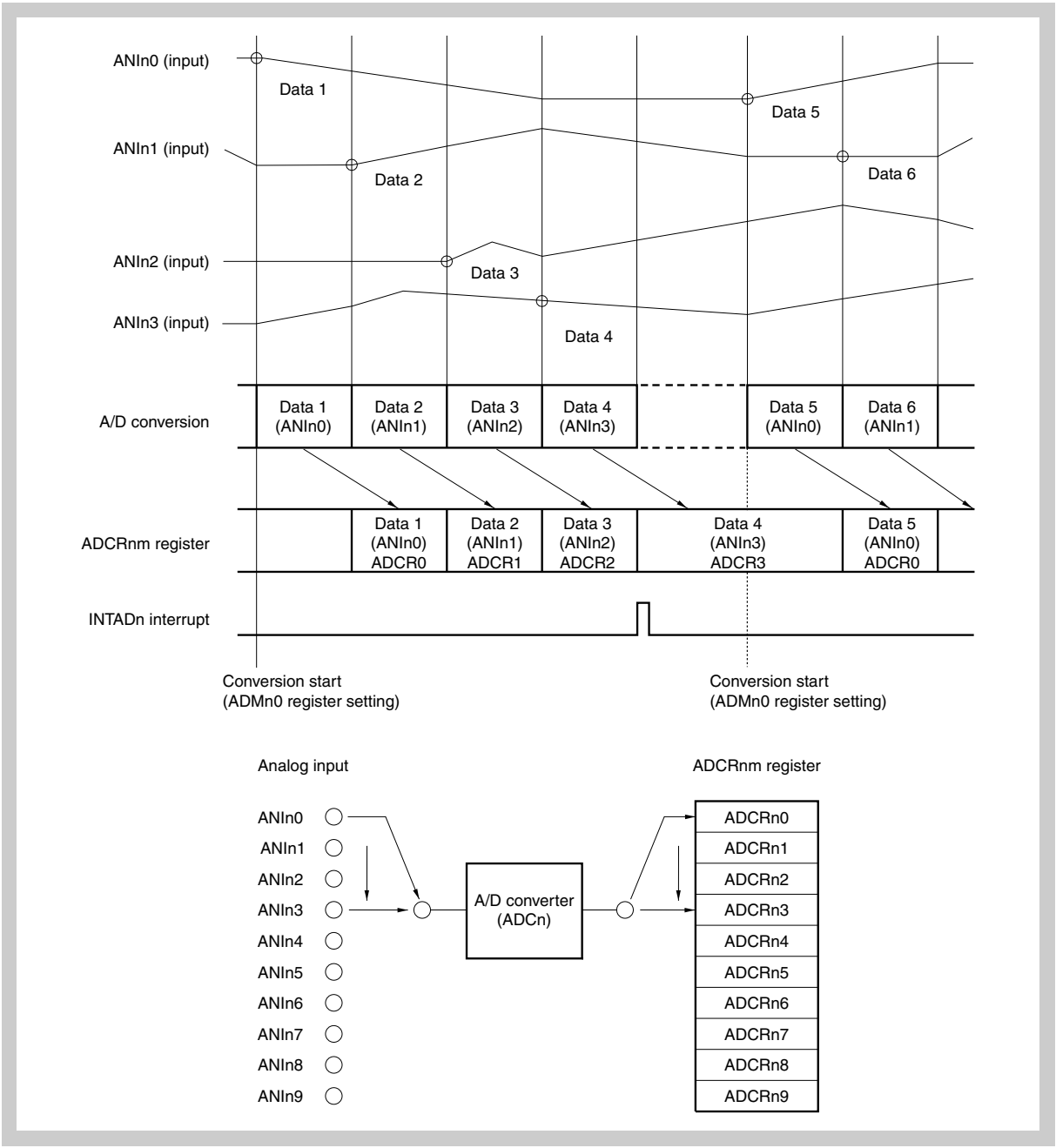


Figure 14-5 Scan Mode Operation Timing: 4-Channel Scan (ANI0 to ANI3)

14.5 Operation in A/D Trigger Mode

When the ADCEn bit of the ADMn0 register is set to 1, A/D conversion is started.

14.5.1 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 (ANInm) and ADCRnm 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 14-8 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.

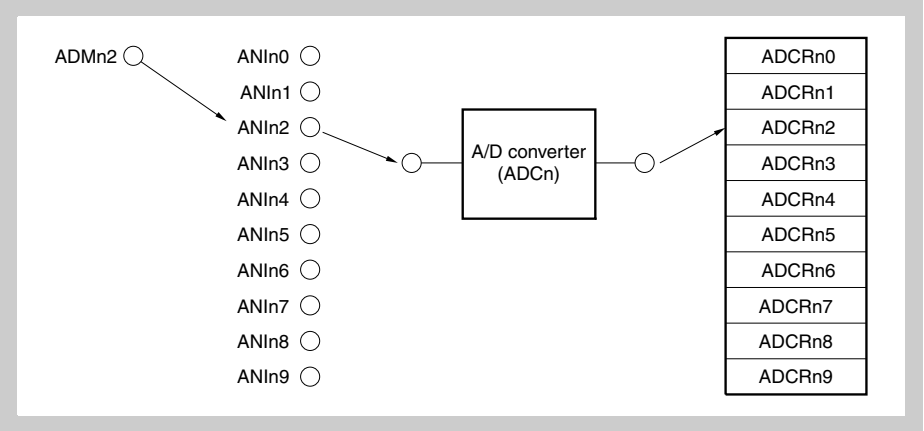


Figure 14-6 Example of 1-Buffer Mode Operation (A/D Trigger Select: 1 Buffer)

<1> The ADMn0.ADCEn bit is set to 1 (enabled).
<2> ANIn2 is 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 (two times for analog input ANIn8 or ANIn9) and the results are stored in the ADCRnm register. When the 4th A/D conversion ends, an A/D conversion end interrupt (INTADn) is generated and the A/D conversion is stopped. The next conversion operation is repeated, unless the ADmn0.ADCEn bit is cleared to 0.

Table 14-9 Correspondence Between Analog Input Pins and ADCRnm 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	ADCRn8 (1st time)
	ADCRn9 (2nd time)

This mode is suitable for applications in which the average of the A/D conversion results is calculated.

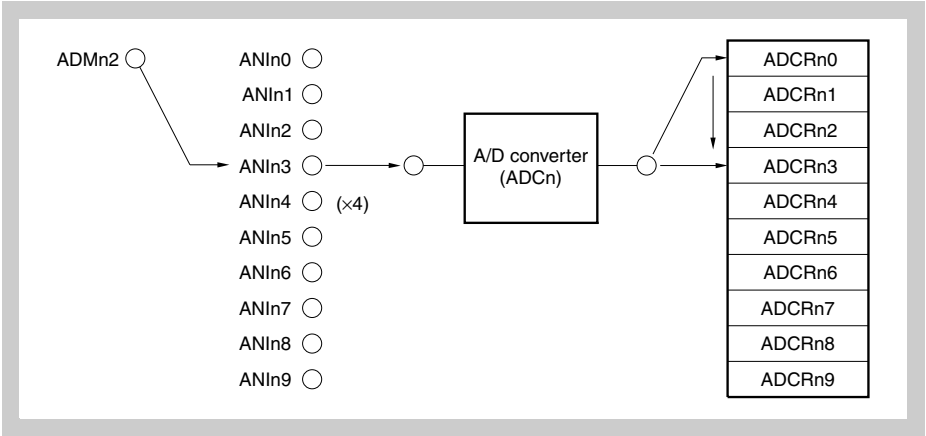


Figure 14-7 Example of 4-Buffer Mode Operation (A/D Trigger Select: 4 Buffers)

<1> The ADMn0.ADCEn bit is set to 1 (enabled).
<2> ANIn3 is converted.
<3> The conversion result is stored in ADCRn0 register.
<4> ANIn3 is converted.
<5> The conversion result is stored in ADCRn1 register.
<6> ANIn3 is converted.
<7> The conversion result is stored in ADCRn2 register.
<8> ANIn3 is converted.
<9> The conversion result is stored in ADCRn3 register.
<10> The INTAD interrupt is generated.

14.5.2 Scan mode operations

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 14-10 Correspondence Between Analog Input Pins and ADCRnm Register (A/D Trigger Scan)

Analog Input	A/D Conversion Result Register
ANIn0	ADCRn0
×	×
×	×
×	×
ANInm[a]	ADCRnm

a) Set by the ANISn3 to ANISn0 bits of the ADMn2 register.

This mode is most appropriate for applications in which multiple analog inputs are constantly monitored.

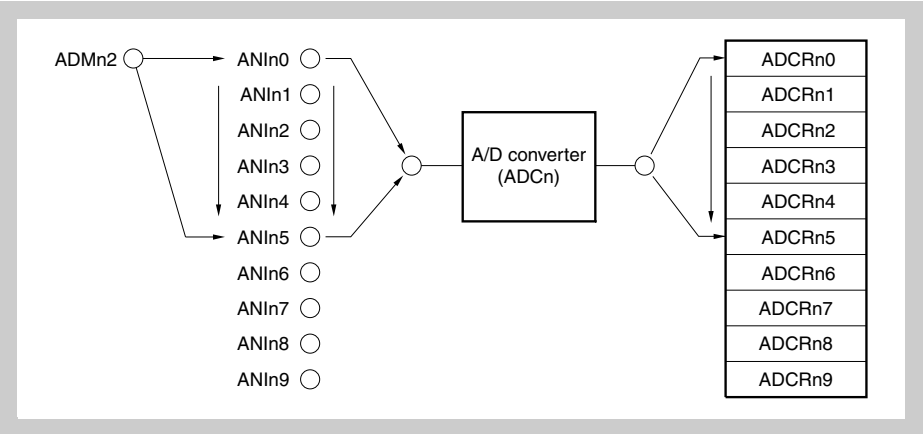


Figure 14-8 Example of Scan Mode Operation (A/D Trigger Scan)

<1> The ADMn0.ADCEn bit is set to 1 (enabled).
<2> ANIn0 is converted.
<3> The conversion result is stored in ADCRn0.
<4> ANIn1 is converted.
<5> The conversion result is stored in ADCRn1.
<6> ANIn2 is converted.
<7> The conversion result is stored in ADCRn2.
<8> ANIn3 is converted.
<9> The conversion result is stored in ADCRn3.
<10> ANIn4 is converted.
<11> The conversion result is stored in ADCRn4.
<12> ANIn5 is converted.
<13> The conversion result is stored in ADCRn5.
<14> The INTAD interrupt is generated.

14.6 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 (A/D converter trigger signal, or top and bottom reversal interrupt) of the inverter timers R0 and R1 (TMR0, TMR1).

The analog input conversion timing is generated when an A/D converter trigger signal from the timers (TR0ADTRG0, TR0ADTRG1, TR1ADTRG0, TR1ADTRG1), or a top or bottom reversal interrupt (INTTR0CD, INTR0OD, INTTR1CD, INTTR1OD) is generated by inverter timer R0 or R1 (TMR0 or TMR1).

When the ADMn0.ADCEn bit is set to 1, the A/D converter waits for the signal (TR0ADTRG0, TR0ADTRG1, TR1ADTRG0, TR1ADTRG1) or interrupt (INTTR0CD, INTR0OD, INTTR1CD, INTTR1OD), and starts conversion when the timer event occurs (ADMn0.ADCSn = 1). When conversion is finished (ADMn0.ADCSn = 0), the converter waits for a timer event signal again .

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.

14.6.1 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 converted once using the trigger of the timer event signals (TR0ADTRG0, TR0ADTRG1, TR1ADTRG0, TR1ADTRG1, INTTR0CD, INTR0OD, INTTR1CD, INTTR1OD) and the result is stored in the corresponding 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 14-11 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 (TR0ADTRG0, TR0ADTRG1, TR1ADTRG0, TR1ADTRG1, INTTR0CD, INTR0OD, INTTR1CD, INTTR1OD)	ANIn0	ADCRn0
	ANIn1	ADCRn1
	ANIn2	ADCRn2
	ANIn3	ADCRn3
	ANIn4	ADCRn4
	ANIn5	ADCRn5
	ANIn6	ADCRn6
	ANIn7	ADCRn7
	ANIn8	ADCRn8
	ANIn9	ADCRn9

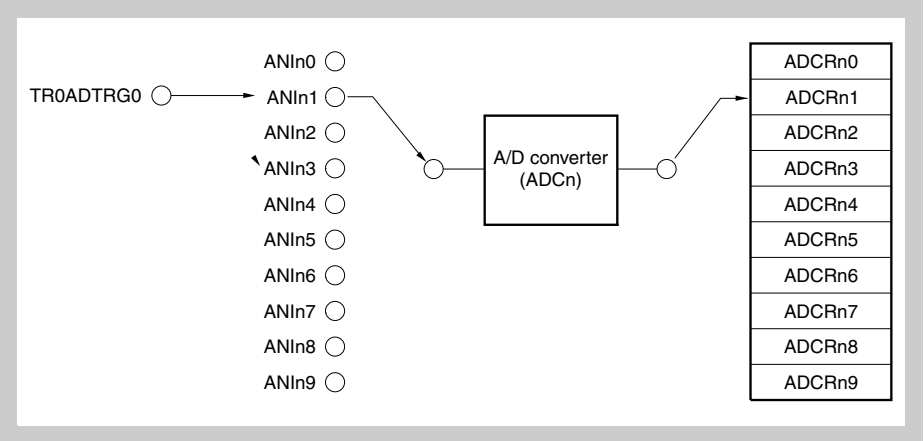


Figure 14-9 Example of 1-Buffer Mode Operation (Timer Trigger Select: 1 Buffer) (ANIn1)

<1> The ADMn0.ADCEn bit isset to 1(enabled).
<2> The TR0ADTRG0 signal is generated.
<3> ANIn1 is 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 ADCRnm register.

One analog input is A/D converted four times using the timer event signals (TR0ADTRG0, TR0ADTRG1, TR1ADTRG0, TR1ADTRG1, INTTR0CD, INTR0OD, INTTR1CD, INTTR1OD) as a trigger, and the results are stored in four ADCRnm registers. The A/D conversion end interrupt (INTADn) is generated when the fourth A/D conversion ends.

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 14-12 Correspondence Between Analog Input Pins and ADCRnm Register (4-Buffer Mode (Timer Trigger Select: 4 Buffers))

Trigger	Analog Input	A/D Conversion Result Register
Timer event signal (TR0ADTRG0, TR0ADTRG1, TR1ADTRG0, TR1ADTRG1, INTTR0CD, INTR0OD, INTTR1CD, INTTR1OD)	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	ADCRn8 (1st time)
		ADCRn9 (2nd time)

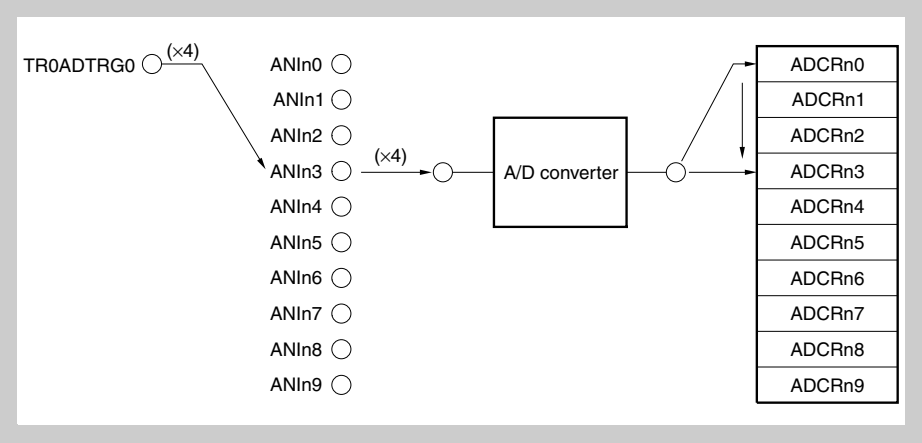


Figure 14-10 Example of 4-Buffer Mode Operation (Timer Trigger Select: 4 Buffers) (ANIn3)

<1> The ADMn0.ADCEn bit is set to 1 (enabled).
<2> The TR0ADTRG0 signal is generated.
<3> ANIn3 is converted
<4> The conversion result is stored in ADCR0.
<5> ANIn3 is converted.
<6> The conversion result is stored in ADCR1.
<7> ANIn3 is converted.
<8> The conversion result is stored in ADCR2.
<9> ANIn3 is converted.
<10> The conversion result is stored in ADCR3.
<11> The INTADn interrupt is generated.

14.6.2 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 ADCEn bit of the ADMn0 register 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 14-13 Correspondence Between Analog Input Pins and ADCRnm Register (Scan Mode (Timer Trigger Scan))

Trigger	Analog Input	A/D Conversion Result Register
Timer event signal (TR0ADTRG0, TR0ADTRG1, TR1ADTRG0, TR1ADTRG1, INTTR0CD, INTR0OD, INTTR1CD, INTTR1OD)	ANIn0	ADCRn0
	ANIn1	ADCRn1
	ANIn2	ADCRn2
	ANIn3	ADCRn3
	ANIn4	ADCRn4
	ANIn5	ADCRn5
	ANIn6	ADCRn6
	ANIn7	ADCRn7
	ANIn8	ADCRn8
	ANIn9	ADCRn9

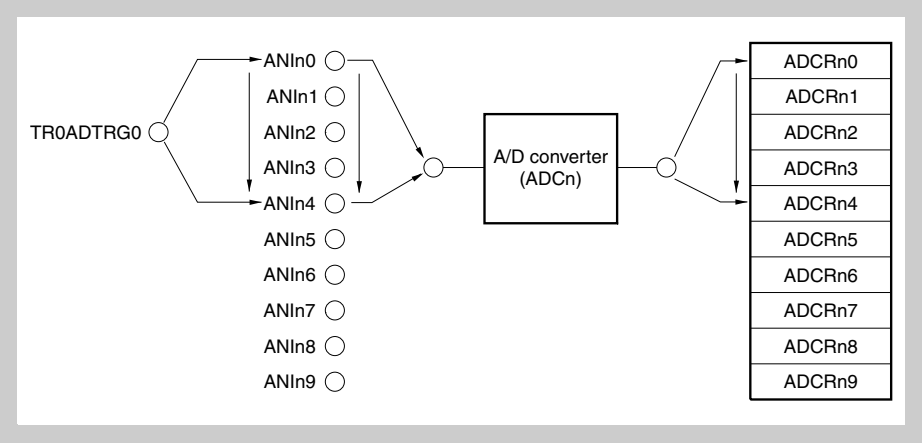


Figure 14-11 Example of Scan Mode Operation (Timer Trigger Scan) (ANIn0 to ANIn4)

<1> The ADMn0.ADCEn bit is set to 1 (enabled).
<2> The TR0ADTRG0 signal is generated.
<3> ANIn0 is converted.
<4> The conversion result is stored in ADCRn0.
<5> ANIn1 is converted.
<6> The conversion result is stored in ADCRn1.
<7> ANIn2 is converted.
<8> The conversion result is stored in ADCRn2.
<9> ANIn3 is converted.
<10> The conversion result is stored in ADCRn3.
<11> ANIn4 is converted.
<12> The conversion result is stored in ADCRn4.
<13> The INTADn interrupt is generated.

14.7 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 ADMn1.EGAn[1:0] bits.

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 (ADMn0.ADCSn = 0), it waits for the external trigger again .

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.

14.7.1 Select mode operations

In this mode, one analog input (ANIn0 to ANIn9) specified by the ADMn2 register is A/D converted. The conversion result is stored in the ADCRnm register corresponding to the analog input ANInm. In select mode, there are two sub-modes available: 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 result is stored in one ADCRnm register. The analog input and the A/D conversion result 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 14-14 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 to the ADTRGn pin.

This mode is most appropriate for applications in which the results are read after each A/D conversion.

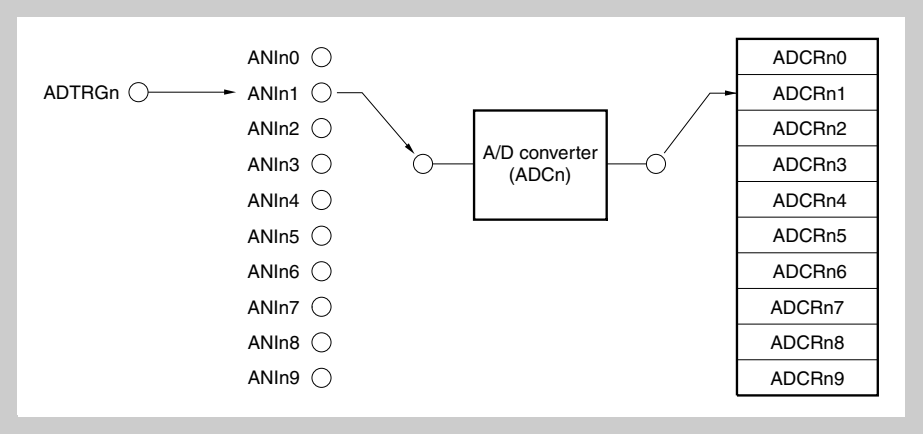


Figure 14-12 Example of 1-Buffer Mode Operation (External Trigger Select: 1 Buffer) (ANIn1)

<1> The ADMn0.ADCEn bit is set to 1 (enabled).
<2> The external trigger is generated.
<3> ANIn1 is 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 registers. The A/D conversion end interrupt (INTADn) is generated and A/D conversion is stopped after the 4th A/D conversion.

Table 14-15 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	ADCRn8 (1st time)
		ADCRn9 (2nd time)

While the ADMn0.ADCEn bit = 1, A/D conversion is started when 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.

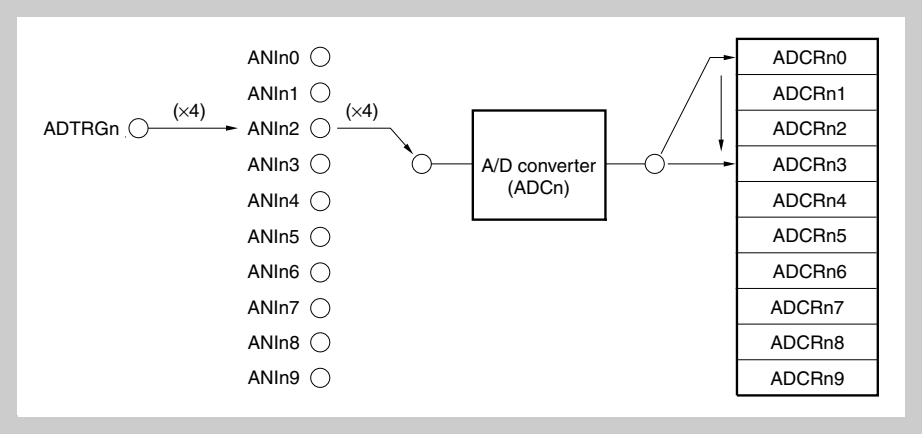


Figure 14-13 Example of 4-Buffer Mode Operation (External Trigger Select: 4 Buffers) (ANIn2)

<1> The ADMn0.ADCEn bit is set to 1 (enabled).
<2> The external trigger is generated.
<3> ANIn3 is converted.
<4> The conversion result is stored in ADCR0.
<5> ANIn3 is converted.
<6> The conversion result is stored in ADCR1.
<7> ANIn3 is converted.
<8> The conversion result is stored in ADCR2.
<9> ANIn3 is converted.
<10> The conversion result is stored in ADCR3.
<11> The INTADn interrupt is generated.

14.7.2 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.

Table 14-16 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

When a trigger is input to the ADTRGn pin while the ADMn0.ADCEn bit is 1, A/D conversion is started again.

This is most appropriate for applications in which multiple analog inputs are constantly monitored.

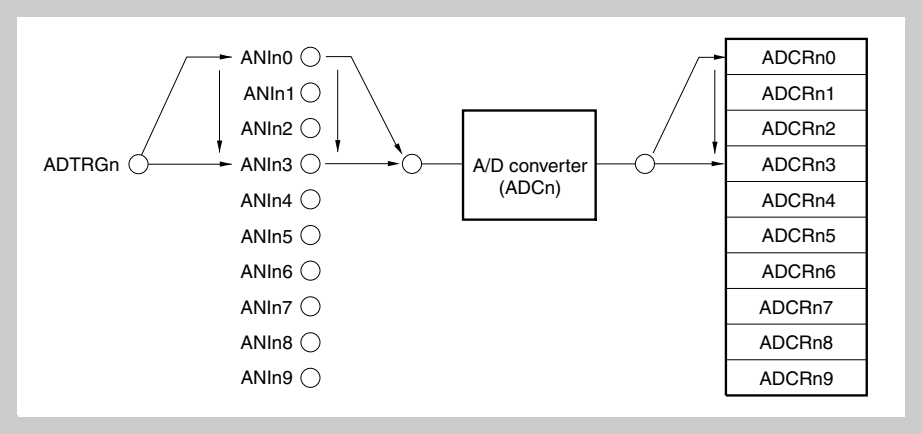


Figure 14-14 Example of Scan Mode Operation (External Trigger Scan) (ANIn0 to ANIn3)

<1> The ADCEn bit of ADMn0 register 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

14.8 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 FRn3 to FRn0 bits of the ADMn1 register.

When 0 < interval ≤ conversion operation time:

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 (see section *"HALT mode" on page 210*), 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

(a) 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.

(b) 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.

(c) 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.

Chapter 15 Asynchronous Serial Interface (UARTC)

This microcontroller has two instances of the universal Asynchronous Serial Interface UARTC.

Note Throughout this chapter, the individual instances of UARTCn are identified by “n” (n = 0 or1), for example UCnCTL0 for the UARTCn control register 0.

15.1 Features

- Transfer rate: 16 bps to 2 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

15.2 Configuration

The block diagram of the UARTCn is shown below.

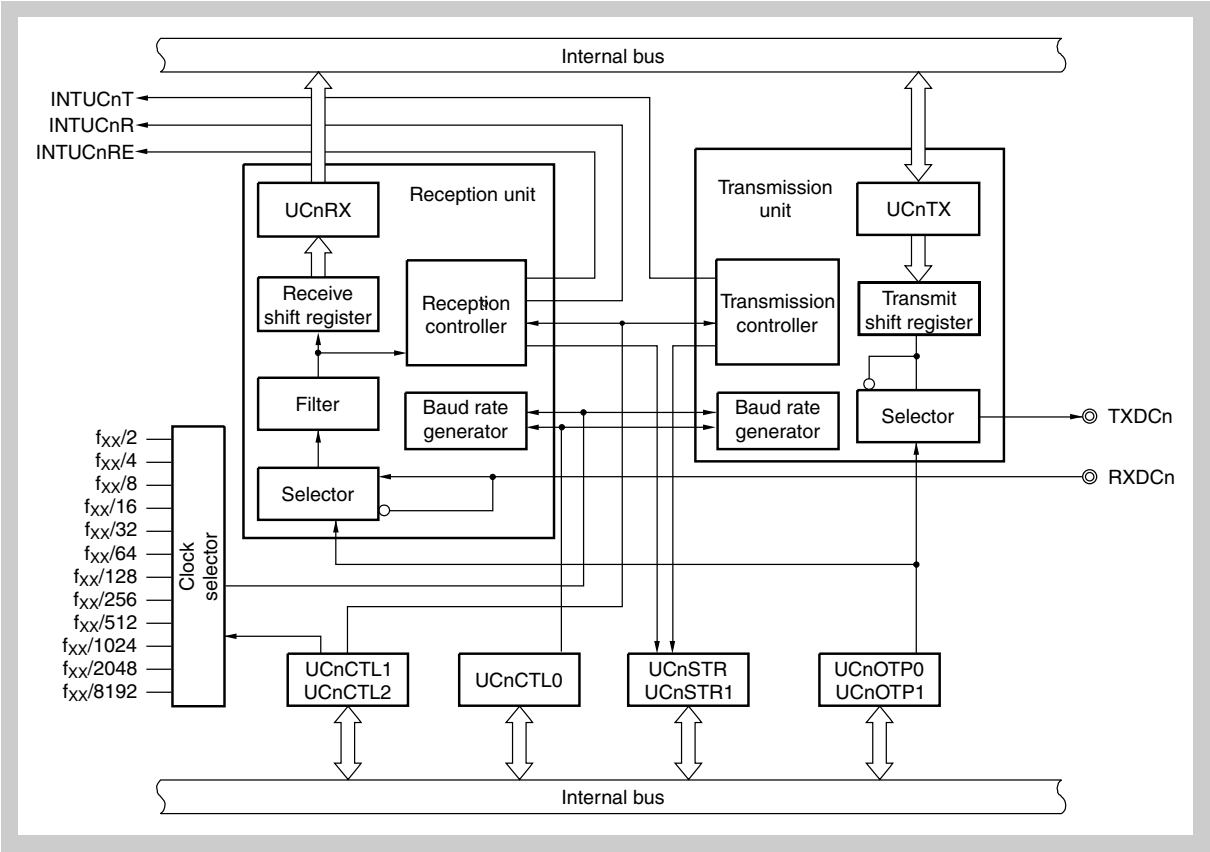


Figure 15-1 Block diagram of Asynchronous Serial Interface UARTCn

Note f_{XX}: Internal system clock

UARTCn consists of the following hardware units.

Table 15-1 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 status register (UCnSTR) UARTCn status register 1 (UCnSTR1) 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 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.

(7) UARTCn status register 1 (UCnSTR1)

The UCnSTR1 register is an 8-bit register indicates the operating status during a reception.

(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.

15.3 Control 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: $FFFFFA00_H$
UC1CTL0: $FFFFFA20_H$

Initial Value 10_H. This register is initialized by any reset.

UCnCTL0

7	6	5	4	3	2	1	0
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 UCnOPT0.UCnRDL bit = 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 15-2 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. **Note:** 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. **Note:** 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 **Note:** This bit can be rewritten only when UCnPWR = 0 or UCnTXE = UCnRXE = 0.

Table 15-2 UCnCTL0 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3
2</td><td>UCnPS1
UCnPS0</td><td>Parity Selection
<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>
For details of parity, see *“Parity types and operations” on page 599*.
Note: **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.
2. When transmission and reception are performed in the LIN format, set the UCnPS[1:0] bits to 00B.
3. This bit can be rewritten only when UCnPWR = 0 or UCnTXE = UCnRXE = 0.</td></tr>
<tr><td>1</td><td>UCnDL</td><td>Data Character Length Specification
0: 7 bits
1: 8 bits
Note: This bit can be rewritten only when UCnPWR = 0 or UCnTXE = UCnRXE = 0.</td></tr>
<tr><td>0</td><td>UCnSL</td><td>Stop Bit Length Specification
0: 1 bit
1: 2 bits
Note: This bit can be rewritten only when UCnPWR = 0 or UCnTXE = UCnRXE = 0.</td></tr>
</table>

(2) UCnCTL1 - UARTCn control register 1

This register controls the Baud Rate Generator. For details see *“UCnCTL1 - UARTCn control register 1” on page 603*.

(3) UCnCTL2 - UARTCn control register 2

This register controls the Baud Rate Generator. For details see *“UCnCTL2 - UARTCn control register 2” on page 604*.

(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: FFFFFA03$_H$
UC1OPT0: FFFFFA23$_H$

Initial Value 14$_H$. This register is initialized by any reset.

UCnOPT0

7	6	5	4	3	2	1	0
UCnSRF	UCnSRT	UCnSTT	UCnSLS2	UCnSLS1	UCnSLS0	UCnTDL	UCnRDL
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15-3 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 • SBF (Sync Brake Field) reception is judged during LIN communication. • The UCnSRF bit is held high when a SBF reception error occurs, and then SBF reception is started again.
6	UCnSRT	SBF Reception Trigger 0: – 1: SBF reception trigger • For SBF reception, set the UCnSRT bit (to 1) to enable reception. **Note: 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 • This bit triggers the SBF transmission during LIN communication. **Note: 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 15-3 UCnOPT0 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>4 to 2</td><td>UCnSLS[2:0]</td><td>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>
Note: Setting of the UCnSLS[2:0] bits is permitted only when UCnCTL0.UCnPWR = 0, or UCnCTL0.UCnTXE = 0.</td></tr>
<tr><td>1</td><td>UCnTDL</td><td>Transmit Data Level
0: Normal output of transfer data
1: Inverted output of transfer data
• The value of the TXDCn pin can be inverted using the UCnTDL bit.
Note: Setting of the UCnTDL bit is permitted only when UCnCTL0.UCnPWR = 0, or UCnCTL0.UCnTXE = 0.</td></tr>
<tr><td>0</td><td>UCnRDL</td><td>Receive Data Level
0: Normal input of transfer data
1: Inverted input of transfer data
• The value of the RXDCn pin can be inverted using the UCnRDL bit.
Note: Setting of the UCnRDL bit is permitted only when UCnCTL0.UCnPWR = 0 ,or UCnCTL0.UCnRXE = 0.</td></tr>
</table>

(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: FFFFFA0A$_H$
UC1OPT1: FFFFFA2A$_H$

Initial Value 00$_H$. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
UCnOPT1	0	0	0	0	0	0	0	UCnEBE
	R	R	R	R	R	R	R	R/W

Table 15-4 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. • During extension bit operation a 9-th data bit is sent or received instead of the parity bit. **Note:** 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 15-5 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 [a]	Stop	
			1	1	Data	Data	Data [a]	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) 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: $FFFFFA04_H$
UC1STR: $FFFFFA24_H$

Initial Value 00_H. This register is cleared by any reset, and when UCnCTL0.PWR = 0 is set.

UCnSTR

7	6	5	4	3	2	1	0
UCnTSF	0	0	0	0	UCnPE	UCnFE	UCnOVE
R	R	R	R	R	R/W	R/W	R/W

Table 15-6 UCnSTR register contents (1/2)

Bit position	Bit name	Function
7	UCnTSF	Transfer Status Flag 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. 1: When data to be transferred is written to UCnTX register, or when SBF transmission trigger bit is set (UCnSTT = 1). **Note:** 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.
2	UCnPE	Parity Error Flag 0: When UCnCTL0.UCnPWR = 0, or when UCnCTL0.UCnRXE = 0 has been set (reception disabled), or when 0 has been written 1: When parity of data and parity bit do not match during reception. **Note:** **1.** The operation of the UCnPE bit is controlled by the settings of the UCnCTL0.UCnPS[1:0] bits. **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.
1	UCnFE	Framing Error Flag 0: When UCnCTL0.UCnPWR = 0, or when UCnCTL0.UCnRXE = 0 has been set (reception disabled), or when 0 has been written 1: When no stop bit is detected during reception. **Note:** **1.** Only the first bit of the receive data stop bits is checked, regardless of the value of the UCnCTL0.UCnSL bit. **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.

Table 15-6 UCnSTR register contents (2/2)

Bit position	Bit name	Function
0	UCnOVE	Overrun Error Flag 0: When UCnCTL0.UCnPWR = 0, or when UCnCTL0.UCnRXE = 0 has been set (reception disabled), or when 0 has been written 1: When data has been received into the UCnRX register and the next receive operation is completed before that receive data has been read. **Note: 1.** When an overrun error occurs, the data is discarded without the next receive data being written to the receive buffer. **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.

(7) UARTCn status register 1 (UCnSTR1)

The UCnSTR1 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: $FFFFFA0B_H$
UC1OPT1: $FFFFFA2B_H$

Initial Value 00_H. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
UCnSTR1	0	0	0	0	0	0	0	UCnRSF
	R	R	R	R	R	R	R	R

Table 15-7 UCnSTR1 register contents

Bit position	Bit name	Function
0	UCnRSF	Receive Status Flag 0: When UCnCTL0.UCnPWR = 0 or UCnCTL0.UCnRXE = 0 has been set, or when the stop bit has been detected. 1: During reception, when the start bit has been detected. • The UCnRSF flag is set (1) by the start bit detection, and it is cleared (0) by detection of the first stop bit condition. **Note:** 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).

(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 overlayed 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 (UCnOPT1.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 (UCnSTR.UCnOVE bit = 1) occurs, the receive data at this time is not transferred to the UCnRX and UXnRXL 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: $FFFFFA06_H$
UC1RX, UC1RXL: $FFFFFA26_H$

Initial Value $1FF_H$ in case of UCnRX register, and
FF_H in case of UCnRXL register, respectively,
by any reset, and when UCnCTL0.PWR = 0 is set.

UCnRX

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Receive data															
								UCnRXL							
R															

UCnRXL

7	6	5	4	3	2	1	0
Receive data							
R							

(9) UCnTX - UARTCn transmit data register

The UCnTX register is a 16-bit buffer register used to set transmit data. It is overlayed 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 (UCnOPT1.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: $FFFFFA08_H$
UC1TX, UC1TXL: $FFFFFA28_H$

Initial Value $1FF_H$ in case of UCnTX register, and
FF_H in case of UCnTXL register, respectively,
by any reset.

UCnTX

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Receive data															
								UCnTXL							
R/W															

UCnTXL

7	6	5	4	3	2	1	0
Receive data							
R/W							

15.4 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.

15.5 Operation

15.5.1 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.

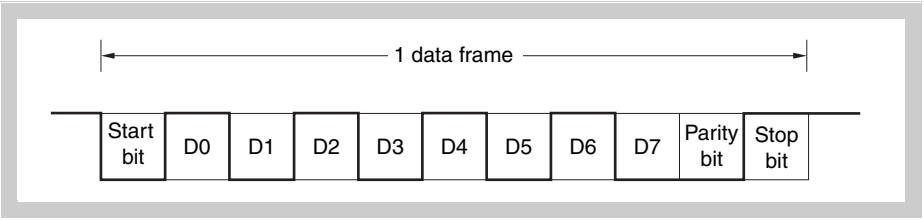
Moreover, control of UART output/inverted output for the TXDCn bit is performed using the UCnOPT0.UCnTDL bit.

- 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

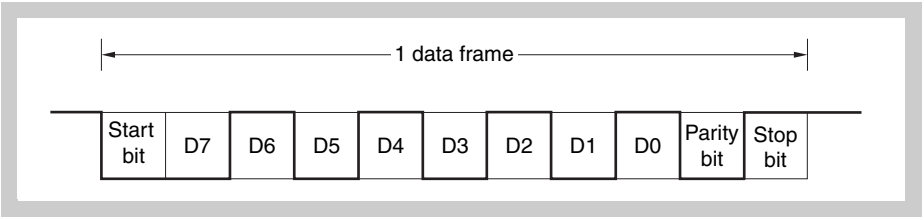
Note Extension bit operation presumes no parity setting.

(1) UARTC transmit/receive data format

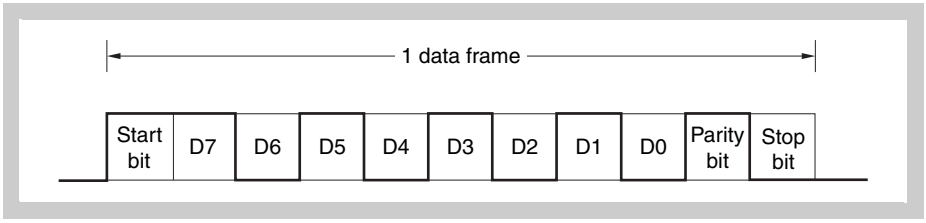
(a) 8-bit data length, LSB first, even parity, 1 stop bit, transfer data: 55H



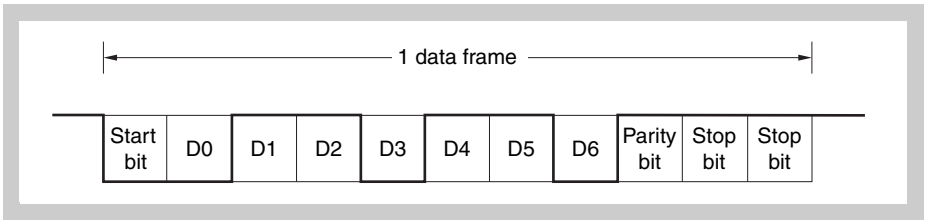
(b) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H



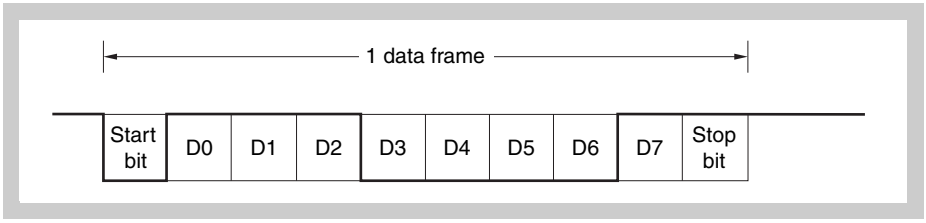
(c) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H, TXDCn inversion



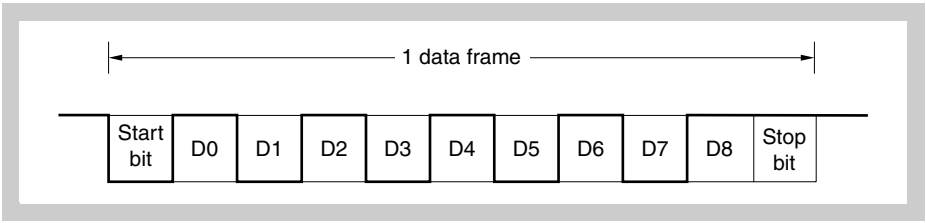
(d) 7-bit data length, LSB first, odd parity, 2 stop bits, transfer data: 36H



(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



15.5.2 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 ±15% or less.

Figure 15-2 and *Figure 15-3* outline the transmission and reception manipulations of LIN.

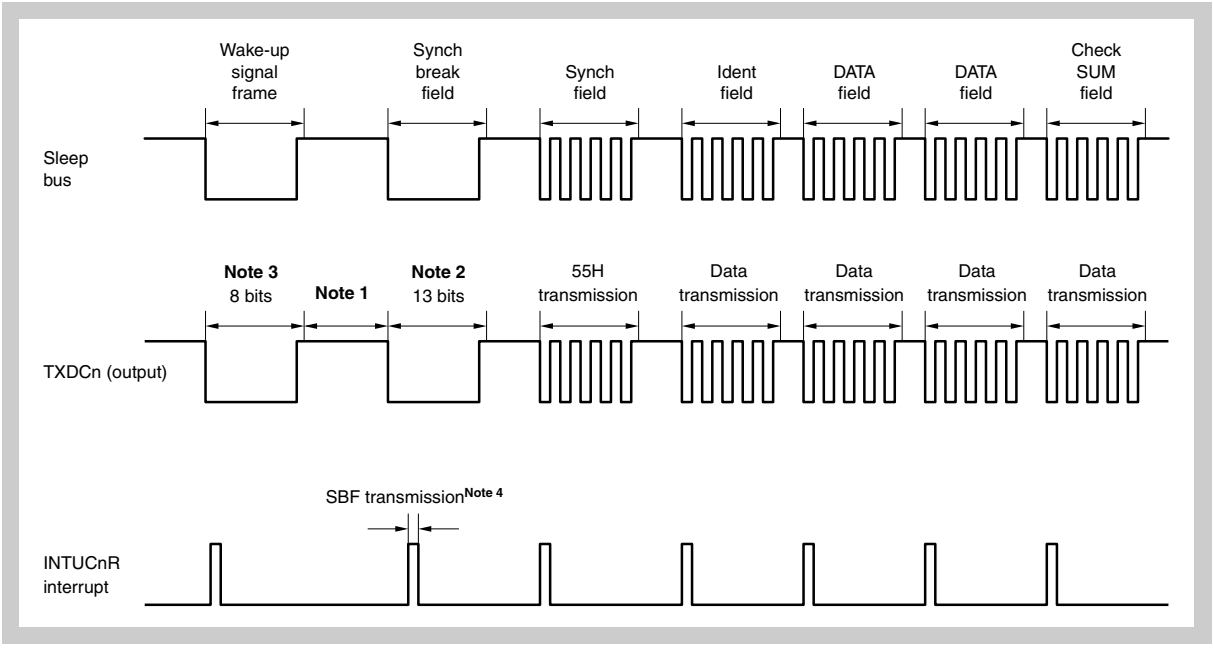


Figure 15-2 LIN transmission manipulation outline

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.

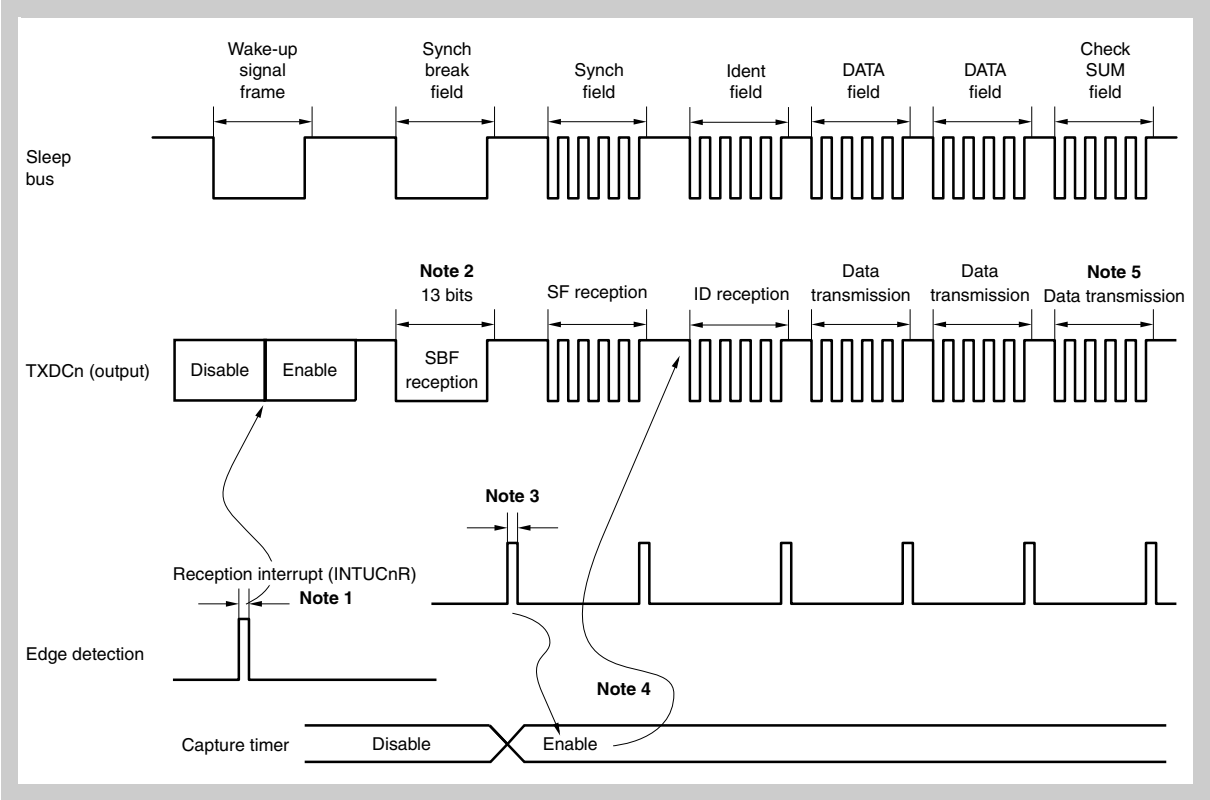


Figure 15-3 LIN reception manipulation outline

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.

15.5.3 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.

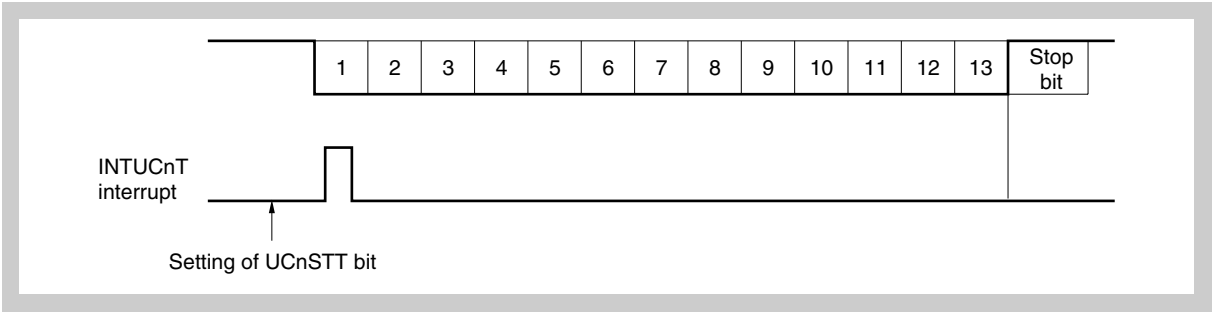


Figure 15-4 SBF transmission

15.5.4 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.

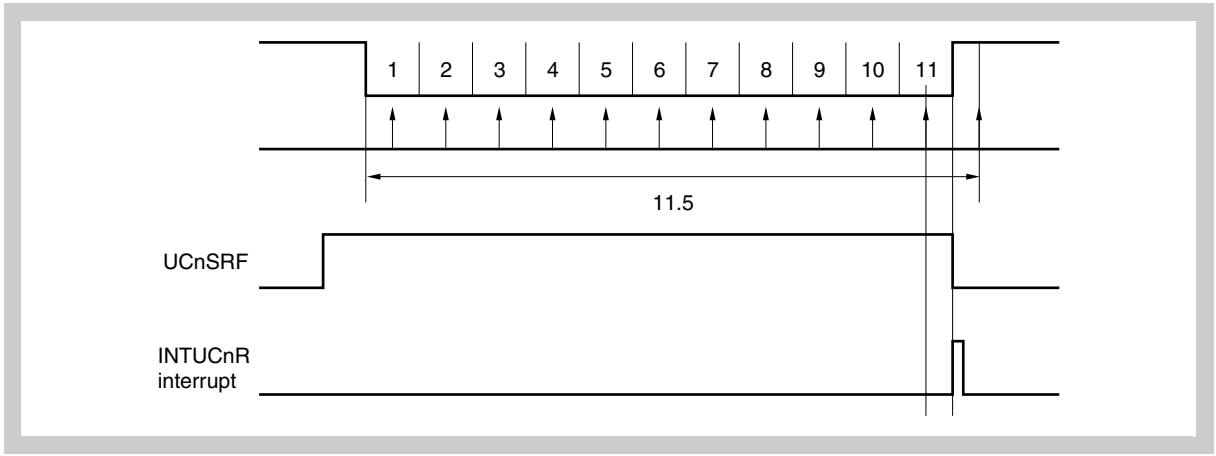
The SBF reception wait status is set by setting the SBF reception trigger (UCnOPT0.UCnSTR bit) to 1.

In the SBF reception wait status, similarly to the UART reception wait status, the RXDCn pin is monitored and start bit detection is performed.

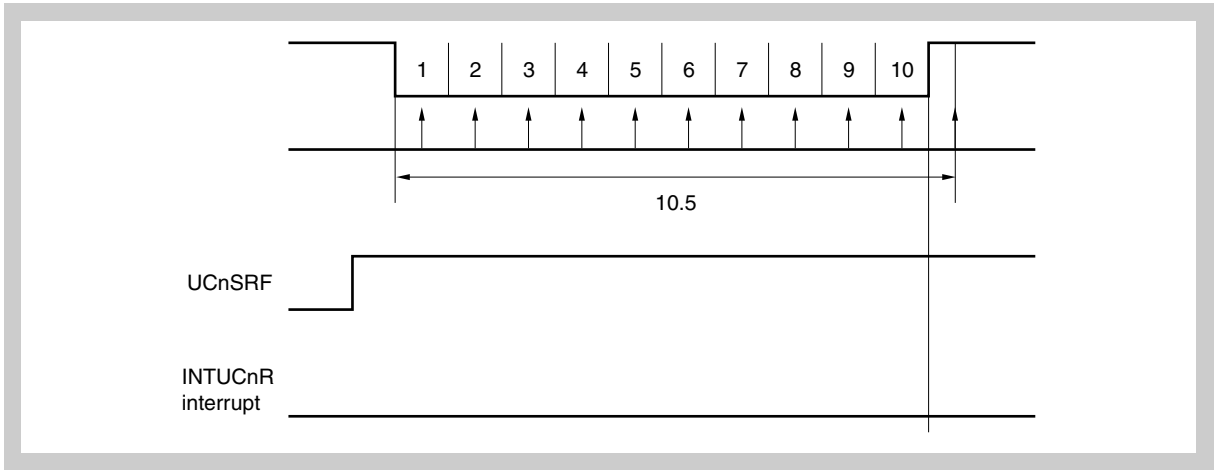
Following detection of the start bit, reception is started and the internal counter counts up according to the set baud rate.

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)



15.5.5 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.

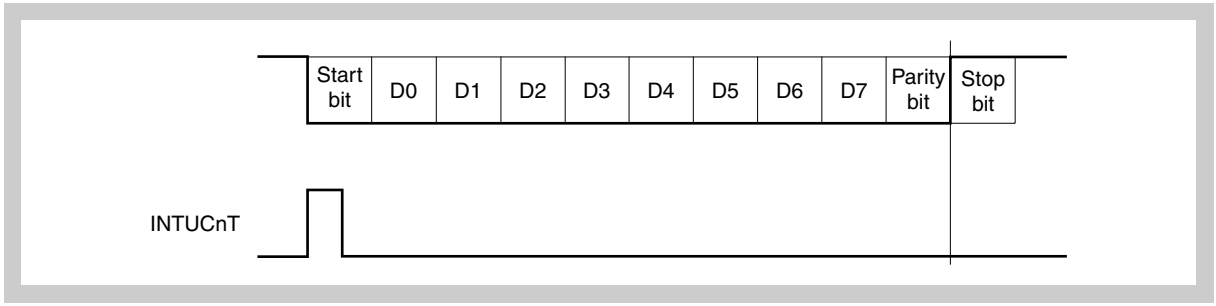


Figure 15-5 UART transmission

Note LSB first

15.5.6 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.

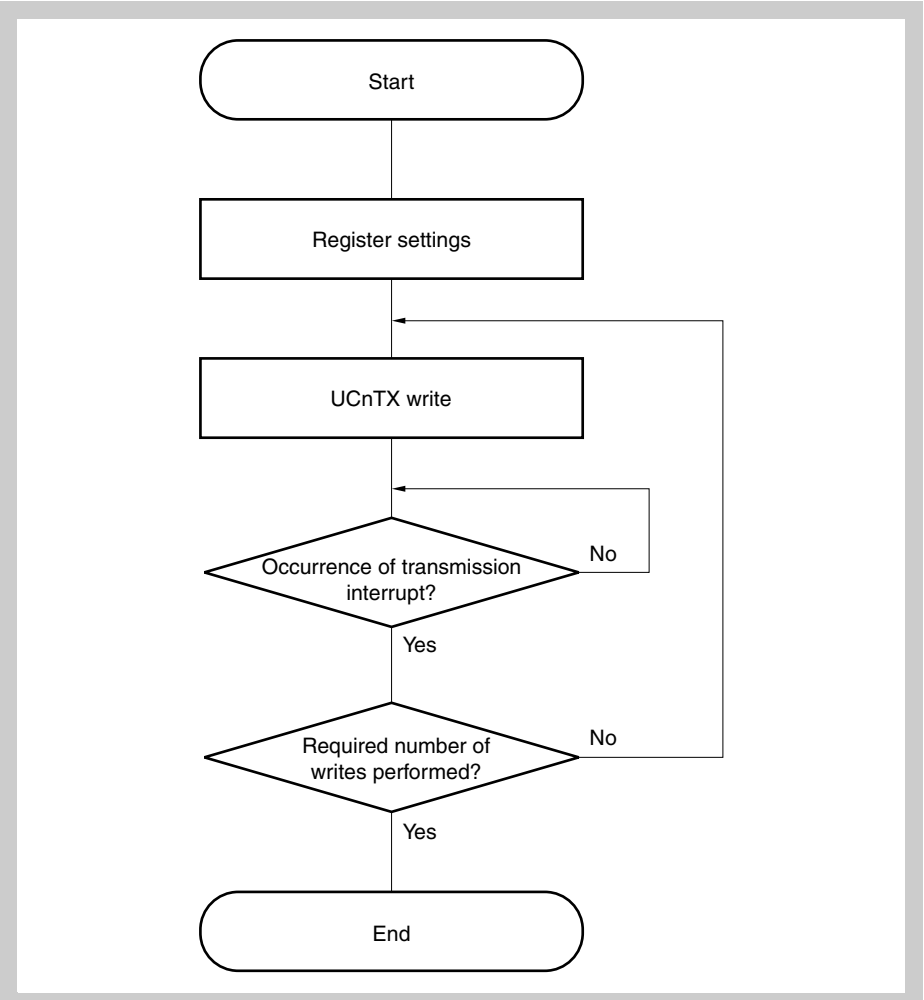


Figure 15-6 Continuous transmission processing flow

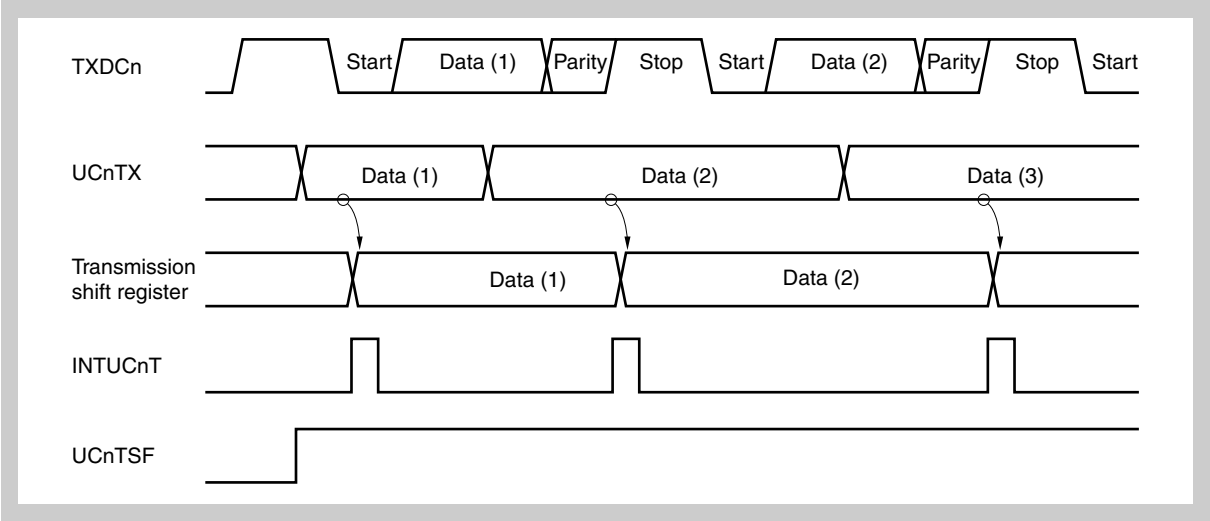


Figure 15-7 Continuous transmission operation timing —transmission start

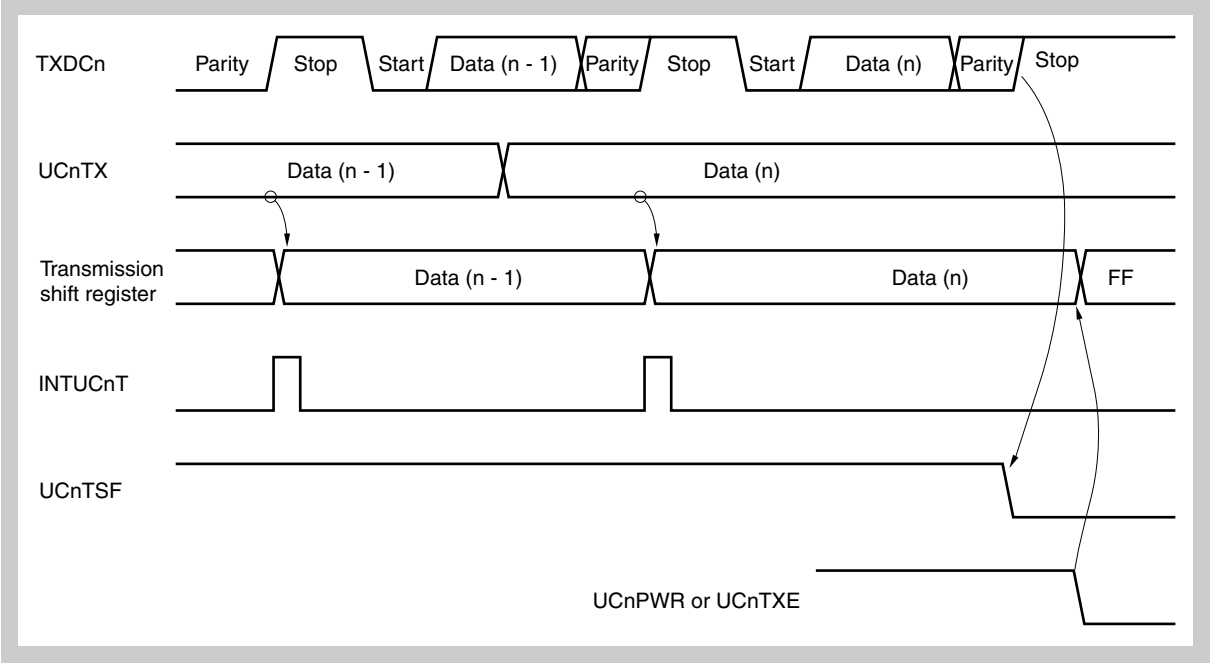


Figure 15-8 Continuous transmission operation timing—transmission end

15.5.7 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 UCnSTR1.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.

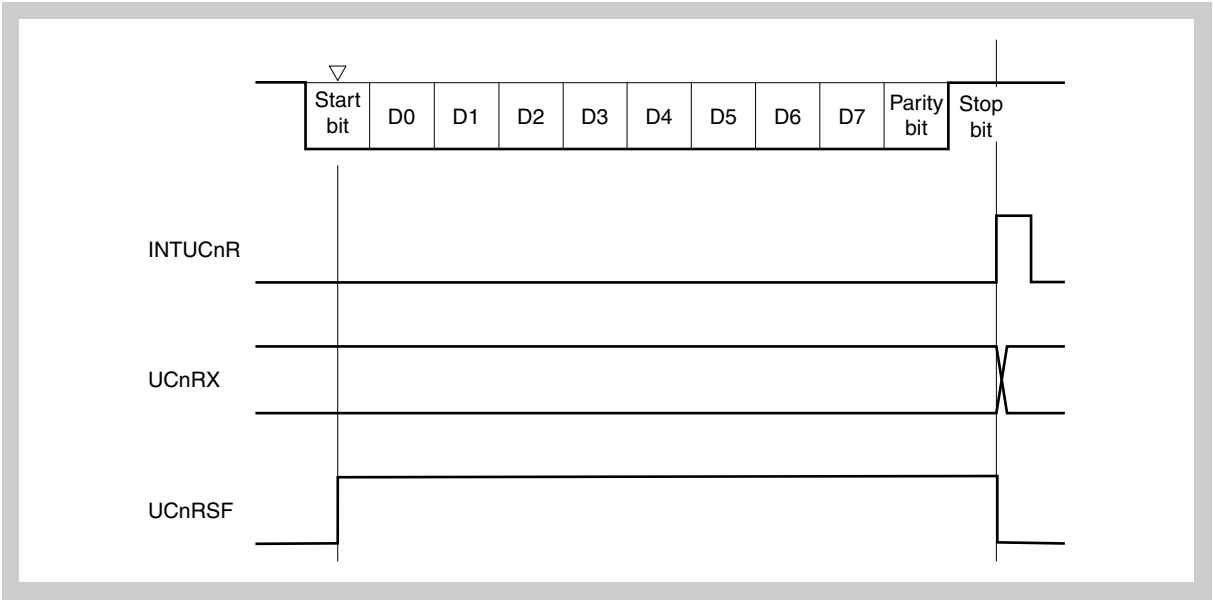


Figure 15-9 UART reception

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.

15.5.8 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 15-8 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.

15.5.9 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.

15.5.10 Receive data noise filter

This filter samples the RXDCn pin using the selected clock ($f_{XX}/2$).

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 15-10*).

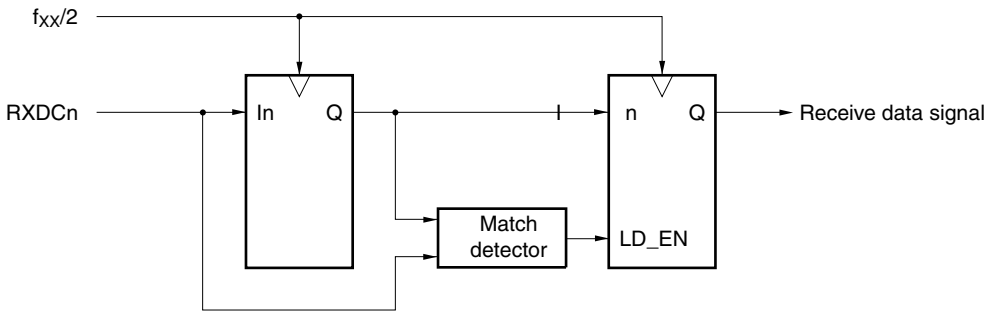


Figure 15-10 Noise filter circuit

15.6 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.

15.6.1 Baud rate generator configuration

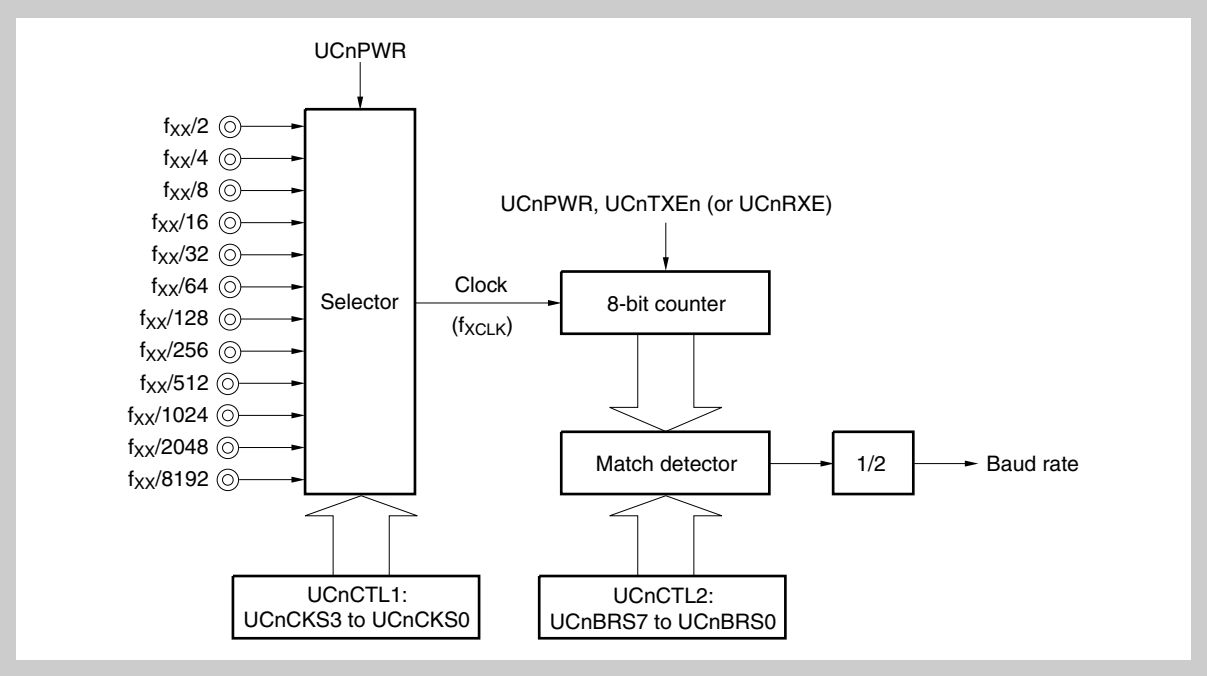


Figure 15-11 Configuration of baud rate generator

(1) 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 (f_{XCLK}). When the UCnPWR bit = 0, the clock is fixed to the low level.

(2) 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.

15.6.2 Control registers

(1) 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: $FFFFFA01_H$
UC1CTL1: $FFFFFA21_H$

Initial Value 00_H. This register is cleared by any reset.

UCnCTL1

7	6	5	4	3	2	1	0
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 15-9 UCnCTL1 register contents

Bit position	Bit name	Function
3 to 0	UCnCKS[3:0]	Base clock selection

UCnCKS3	UCnCKS2	UCnCKS1	UCnCKS0	Base clock (f_{XCLK}) selection
0	0	0	0	$f_{XX}/2$
0	0	0	1	$f_{XX}/4$
0	0	1	0	$f_{XX}/8$
0	0	1	1	$f_{XX}/16$
0	1	0	0	$f_{XX}/32$
0	1	0	1	$f_{XX}/64$
0	1	1	0	$f_{XX}/128$
0	1	1	1	$f_{XX}/256$
1	–	0	0	$f_{XX}/512$
1	–	0	1	$f_{XX}/1024$
1	–	0	0	$f_{XX}/2048$
1	–	0	1	$f_{XX}/8192$

Note: f_{XX}: Internal system clock

(2) 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: $FFFFFA02_H$
UC1CTL2: $FFFFFA22_H$

Initial Value FF_H. This register is cleared by any reset.

UCnCTL2

7	6	5	4	3	2	1	0
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 15-10 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 [a]
0	0	0	0	0	0	-	-	-	Setting prohibited
0	0	0	0	0	1	0	0	4	$f_{XCLK}/4$
0	0	0	0	0	1	0	1	5	$f_{XCLK}/5$
0	0	0	0	0	1	1	0	6	$f_{XCLK}/6$
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	0	0	252	$f_{XCLK}/252$
1	1	1	1	1	1	0	1	253	$f_{XCLK}/253$
1	1	1	1	1	1	1	0	254	$f_{XCLK}/254$
1	1	1	1	1	1	1	1	255	$f_{XCLK}/255$

a) f_{XCLK}: Clock frequency selected by UCnCTL1.UCnCKS[3:0]

15.6.3 Baud rate

The baud rate is obtained by the following equation.

$$\text{Baud rate} = \frac{f_{XCLK}}{2 \times k} \text{ [bps]}$$

f_{XCLK} = Clock frequency selected by UCnCTL1.UCnCKS[3:0].

k = Value set using the UCnCTL2.UCnBRS[7:0] bits
(k = 4, 5, 6, …, 255)

15.6.4 Baud rate error

The baud rate error is obtained by the following equation.

$$\text{Error (\%)} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Target baud rate (correct baud rate)}} - 1 \right) \times 100 \text{ [\%]}$$

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 606*.

15.6.5 Baud rate setting example

Example

- Setting value of
  - UCnCTL1.UDnCKS[3:0] = 0001_B: $f_{XCLK} = f_{XX}/4$ = 16 MHz
  - UCnCTL2.UDnBRS[7:0] = $0011\ 0100_B$: k = 52
- Target baud rate = 153,600 bps
- Actual Baud rate = 16 MHz / (2 × 52) = 153,846 [bps]
- Baud rate error = (153,846/153,600 – 1) × 100 = 0.160 [%]

Table 15-11 Baud rate generator setting data (1/2)

Target baud rate [bps]	UCnCTL1	f_{XCLK}	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

Table 15-11 Baud rate generator setting data (2/2)

Target baud rate [bps]	UCnCTL1	f_{XCLK}	UCnCTL2	k	Baud rate error [%]
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

15.6.6 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.

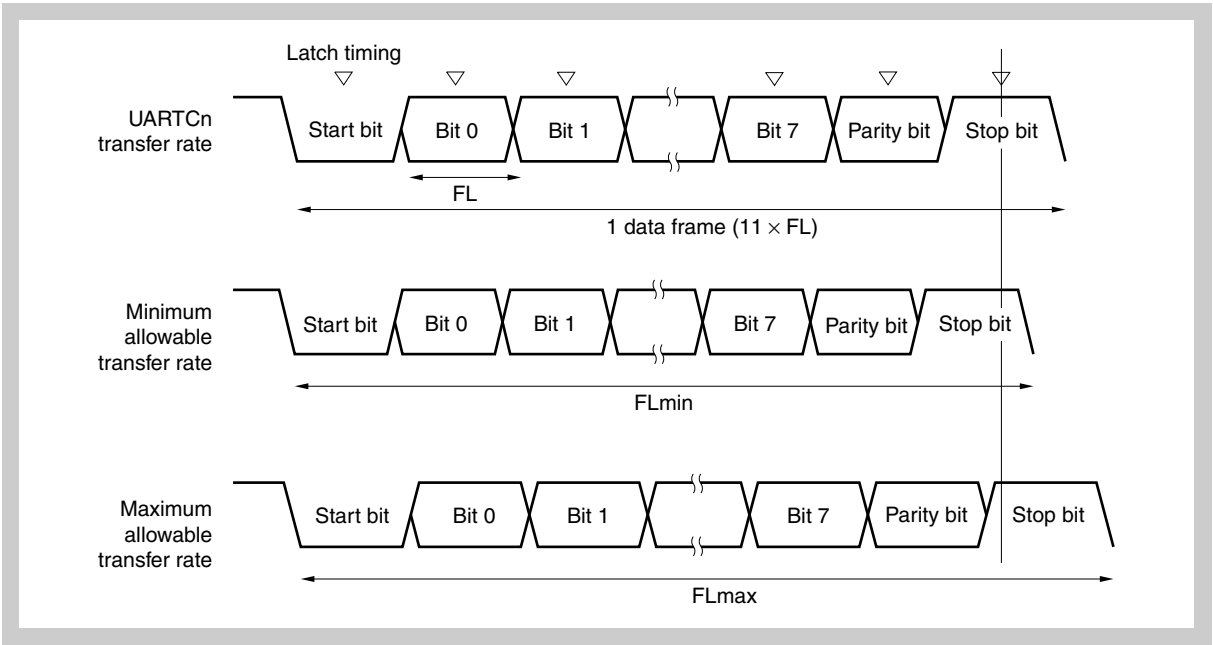


Figure 15-12 Allowable baud rate range during reception

As shown in *Figure 15-12*, 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.

$$FL = (Brate)^{-1}$$

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.

$$BR_{max} = (FL_{min}/11)^{-1} = \frac{22k}{21k+2} \times Brate$$

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.

$$BR_{min} = (FL_{max}/11)^{-1} = \frac{20k}{21k-2} \times Brate$$

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 15-12 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]

15.6.7 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.

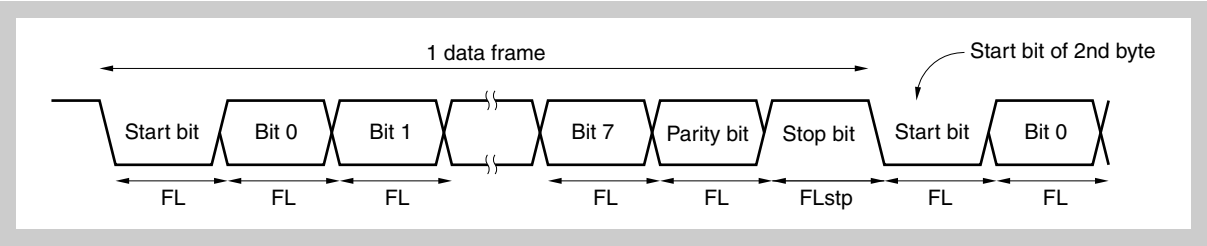


Figure 15-13 Transfer rate during continuous transfer

Assuming 1 bit data length: FL; stop bit length: FLstp; and base clock frequency: f_{XCLK}, we obtain the following equation.

$$FLstp = FL + 2/f_{XCLK}$$

Therefore, the transfer rate during continuous transmission is as follows.

$$\text{Transfer rate} = 11 \times FL + (2/f_{XCLK})$$

15.7 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 16 Clocked Serial Interface B (CSIB)

The V850E/PH2 microcontrollers have following number of channels of the clocked serial interface B (CSIB):

CSIB	µPD70F3447	µPD70F3187
Instances	1	2
Names	CSIB0	CSIB0 to CSIB1

Note Throughout this chapter, the individual instances of CSIB are identified by “n” (n = 0, 1).

16.1 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

16.2 Configuration

The following figure shows the block diagram of CSIBn.

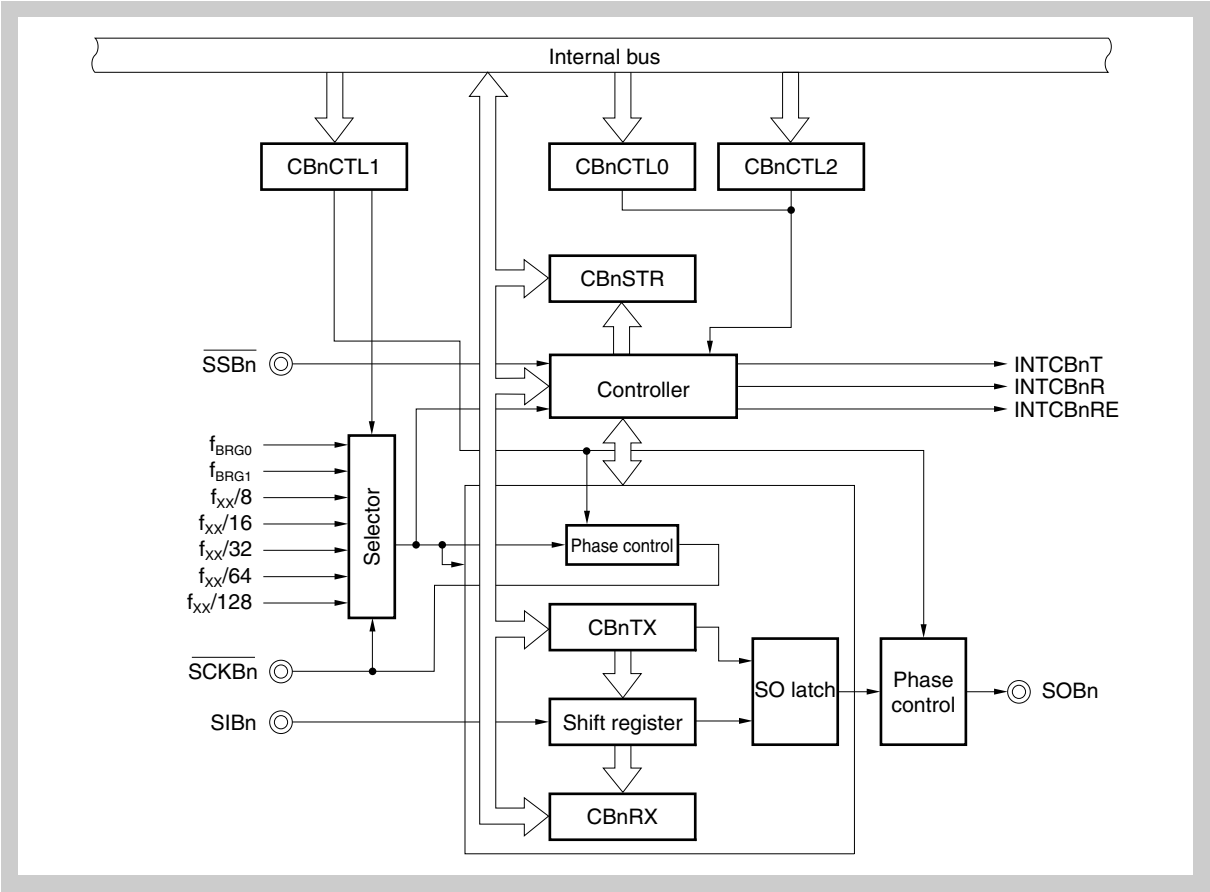


Figure 16-1 Block diagram of CSIBn

CSIBn includes the following hardware.

Table 16-1 Configuration of CSIBn

Item	Configuration
Registers	CSIBn receive data register (CBnRX) CSIBn transmit data register (CBnTX)
Control registers	CSIBn control register 0 (CBnCTL0) CSIBn control register 1 (CBnCTL1) CSIBn control register 2 (CBnCTL2) CSIBn status register (CBnSTR)

16.3 CSIB Control Registers

The clocked serial interfaces CSIBn are controlled and operated by means of the following registers:

Table 16-2 CSIBn registers overview

Register name	Shortcut	Address
CSIBn control register 0	CBnCTL0	<base>
CSIBn control register 1	CBnCTL1	<base> + 1_H
CSIBn control register 2	CBnCTL2	<base> + 2_H
CSIBn status register	CBnSTR	<base> + 3_H
CSIBn receive data register	CBnRX	<base> + 4_H
CSIBn transmit data register	CBnTX	<base> + 6_H

Table 16-3 CSIBn register base address

Timer	Base address
CSIB0	FFFF $FD00_H$
CSIB1	FFFF $FD10_H$

(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 <base>

Initial Value 01_H. This register is cleared by any reset.

CBnCTL0

7	6	5	4	3	2	1	0
CBnPWR	CBnTXE[a]	CBnRXE[a]	CBnDIR[a]	0	0	CBnTMS[a]	CBnSCE
R/W	R/W	R/W	R/W	R/W	R/W	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 16-4 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.
6	CBnTXE	Transmit operation disable/enable: 0: Disable transmit operation 1: Enable transmit operation The SOBn output is low level 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 (CBnRX 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 16-4 CBnCTL0 register contents (2/2)

Bit position	Bit name	Function
0	CBnSCE	Specification of start transfer disable/enable: 0: Communication start trigger invalid 1: Communication start trigger valid • In master mode This bit enables or disables the communication start trigger. (a) In single transmission or transmission/reception mode, or continuous transmission mode or continuous transmission/reception mode A communication operation can be started only when the CBnSCE bit = 1. (Set the CBnSCE bit to 1.) (b) In single reception mode Clear the CBnSCE bit to 0 before reading the receive data (CBnRX register). If the CBnSCE bit is read while it is 1, the next communication operation is started. (c) In continuous reception mode Clear the CBnSCE bit to 0 one communication clock before reception of the last data is completed. The CBnSCE bit is not cleared to 0 one communication clock before the completion of the last data reception, the next communication operation is automatically started. • In slave mode This bit enables or disables the communication start trigger. Set the CBnSCE bit to 1.

Note These bits can only be rewritten when the CBnPWR bit = 0. However, CBnPWR bit = 1 can also be set at the same time as rewriting these bits.

Note
1. These bits can only be rewritten when the CBnPWR bit = 0. However, the CBnPWR can be set to 1 at the same time as these bits are rewritten.
2. If the CBnSCE bit is read while it is 1, the next communication operation is started.
3. The CBnSCE bit is not cleared to 0 one communication clock before the completion of the last data reception, the next communication operation is automatically started.

(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 <base> + 1_H

Initial Value 00_H. This register is cleared by any reset.

Caution The CBnCTL1 register can be rewritten only when the CBnCTL0.CBnPWR bit = 0.

CBnCTL1

7	6	5	4	3	2	1	0
0	0	0	CBnCKP	CBnDAP	CBnCKS2	CBnCKS1	CBnCKS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

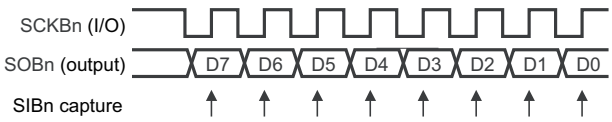
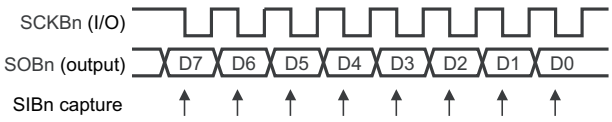
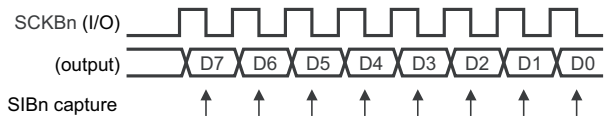
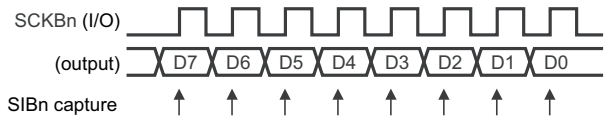
Table 16-5 CBnCTL1 register contents

Bit position	Bit name	Function
4, 3	CBnCKP CBnDAP	Specification of data transmission/reception timing in relation to SCKBn. Refer to *Table 16-6*.
2 to 0	CBnCKS[2:0]	Communication clock setting

CBnCKS2	CBnCKS1	CBnCKS0	Communication clock	Mode
0	0	0	f_{BRG0}	Master
0	0	1	f_{BRG1}	Master
0	1	0	$f_{XX}/8$	Master
0	1	1	$f_{XX}/16$	Master
1	0	0	$f_{XX}/32$	Master
1	0	1	$f_{XX}/64$	Master
1	1	0	$f_{XX}/64$	Master
1	1	1	External clock ($\overline{SCKBn}$)	Slave

Note For details on the baud rate generators refer to *16.7"Baud Rate Generator" on page 641*.

Table 16-6 Specification of data transmission/reception timing in relation to SCKBn

Communication type	CBnCKP	CBnDAP	SIBn/SOBN timing in relation to SCKBn
Communication type 1	0	0	
Communication type 2	0	1	
Communication type 3	1	0	
Communication type 4	1	1	

(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 <base> + 2_H

Initial Value 00_H. This register is cleared by any reset.

Caution The CBnCTL2 register can be rewritten only when the CBnCTL0.CBnPWR bit = 0 or when both the CBnTXE and CBnRXE bits = 0.

	7	6	5	4	3	2	1	0
CBnCTL2	0	0	0	0	CBnCL3	CBnCL2	CBnCL1	CBnCL0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16-7 CBnCTL2 register contents

Bit position	Bit name	Function
3 to 0	CBnCL[3:0]	Sepcifies the serial transfer 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 CBnTX and CBnRX registers.

(a) Transfer data length change function

The CSIBn transfer data length can be set in 1-bit units between 8 and 16 bits using the CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits.

When the transfer bit length is set to a value other than 16 bits, set the data to the CBnTX or CBnRX register starting from the LSB, regardless of whether the transfer start bit is the MSB or LSB. Any data can be set for the higher bits that are not used, but the receive data becomes 0 following serial transfer.

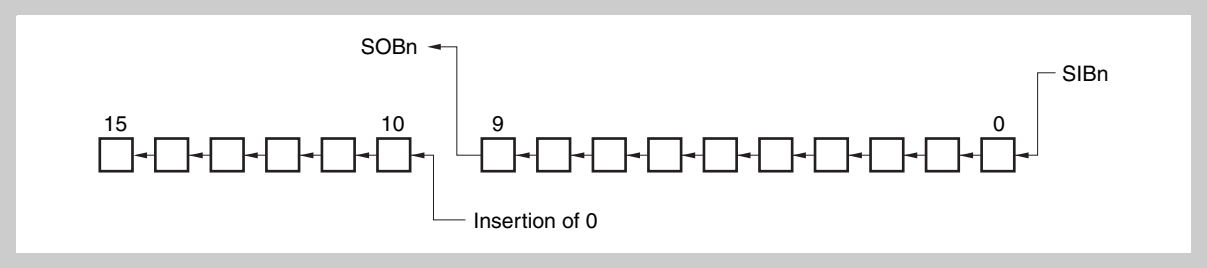


Figure 16-2 (i) Transfer bit length = 10 bits, MSB first

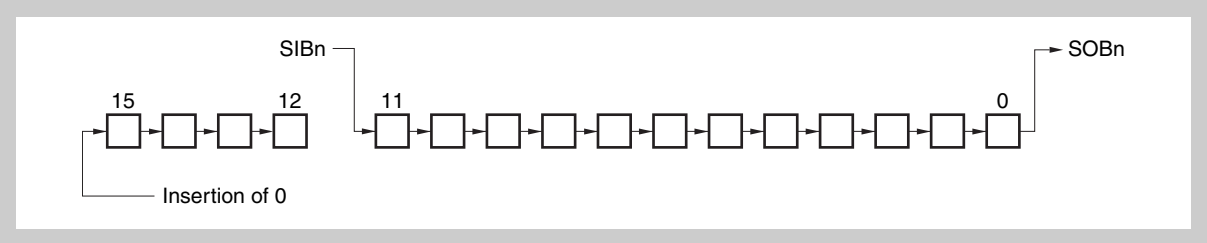


Figure 16-3 (ii) Transfer bit length = 12 bits, LSB first

(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 <base> + 3_H

Initial Value 00_H. 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.

CBnSTR

7	6	5	4	3	2	1	0
CBnTSF	0	0	0	0	0	0	CBnOVE
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16-8

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 overrrun 1: Overrun • An overrun error occurs when the next reception starts without performing a CPU read of the value of the receive buffer, upon completion of the receive operation. The CBnOVE flag displays the overrun error occurrence status in this case. • The CBnOVE flag is cleared by writing 0 to it. It cannot be set even by writing 1 to it.

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 CBnRX will be overwritten with the new data.

(5) CBnRX - CSIBn receive data register

The CBnRX register is a 16-bit buffer register that holds receive data.

Access This register is 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 CBnRXL register.

Address CBnRX, CBnRXL: <base> + 4_H

Initial value 0000H, or 00H respectively. This register is cleared by any reset.
In addition to reset input, the CBnRX register can be initialized by clearing the CBnCTL0.CBnPWR bit to 0.

CBnRX

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

CBnRXL

7	6	5	4	3	2	1	0
Receive data (lower 8 bits)							
R							

The receive operation is started by reading the CBnRX or CBnRXL register in the reception enabled status.

(6) CBnTX - CSIB transmit data register

The CBnTX register is a 16-bit buffer register used to write the CSIBn transfer data.

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 CBnTXL register.

Address CBnTX, CBnTXL: <base> + 6_H

Initial Value 0000_H. This register is cleared by any reset.
In addition to reset input, the CBnTX register can be initialized by clearing the CBnCTL0.CBnPWR bit to 0.

CBnTX

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Transmit data															
R/W															

CBnTXL

7	6	5	4	3	2	1	0
Transmit data (lower 8 bits)							
R/W							

The transmit operation is started by writing data to the CBnTX or CBnTXL register in the transmission enabled status.

Note The communication start conditions are shown below:

- Transmission mode (CBnTXE bit = 1, CBnRXE bit = 0):
  Write to CBnTX or CBnTXL register
- Transmission/reception mode (CBnTXE bit = 1, CBnRXE bit = 1):
  Write to CBnTX or CBnTXL register
- Reception mode (CBnTXE bit = 0, CBnRXE bit = 1):
  Read from CBnRX or CBnRXL register

16.4 Operation

16.4.1 Single transfer mode (master mode, transmission/reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (see 16.4 (2) CSIBn control register 1 (CBnCTL1), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0, 0, 0, 0)

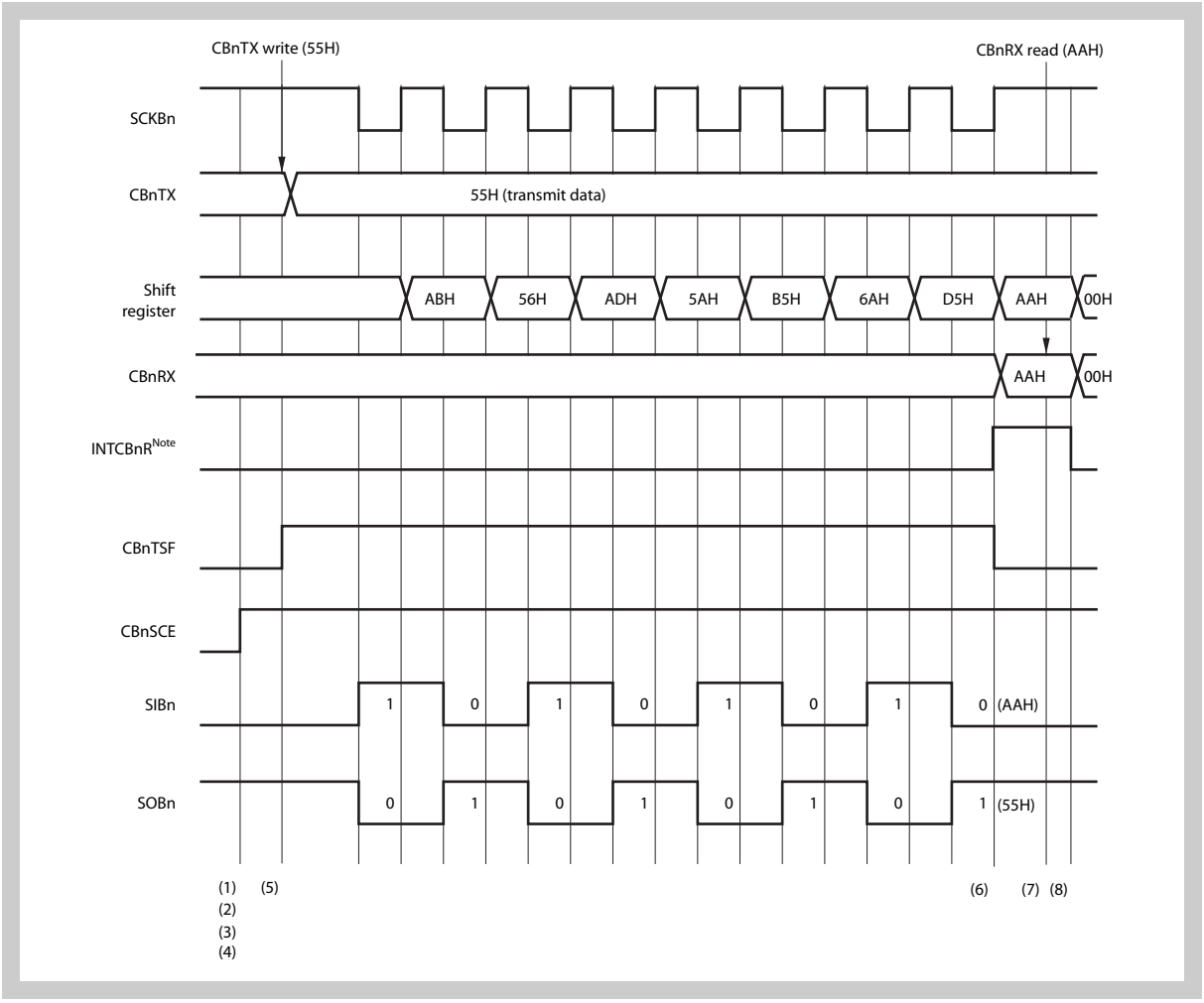


Figure 16-4 Timing diagram in Single transfer mode (master mode, transmission/reception mode)

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 CBnTX register (transmission/reception start).
6. The reception complete interrupt request signal (INTCBnR) is output.
7. Read the CBnRX register before clearing the CBnPWR bit to 0.
8. Check that the CBnSTR.CBnTSF bit = 0 and 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).

In transmission mode or transmission/reception mode, communication is not started by reading the CBnRX register.

Note
1. In single transmission or single transmission/reception mode, the INTCBnT signal is not generated. When communication is complete, the INTCBnR signal is generated.
2. 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 CBnTX 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 trasmitting 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 CBnTX register.

16.4.2 Single transfer mode (master mode, reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (see 16.4 (2) CSIBn control register 1 (CBnCTL1), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0, 0, 0, 0)

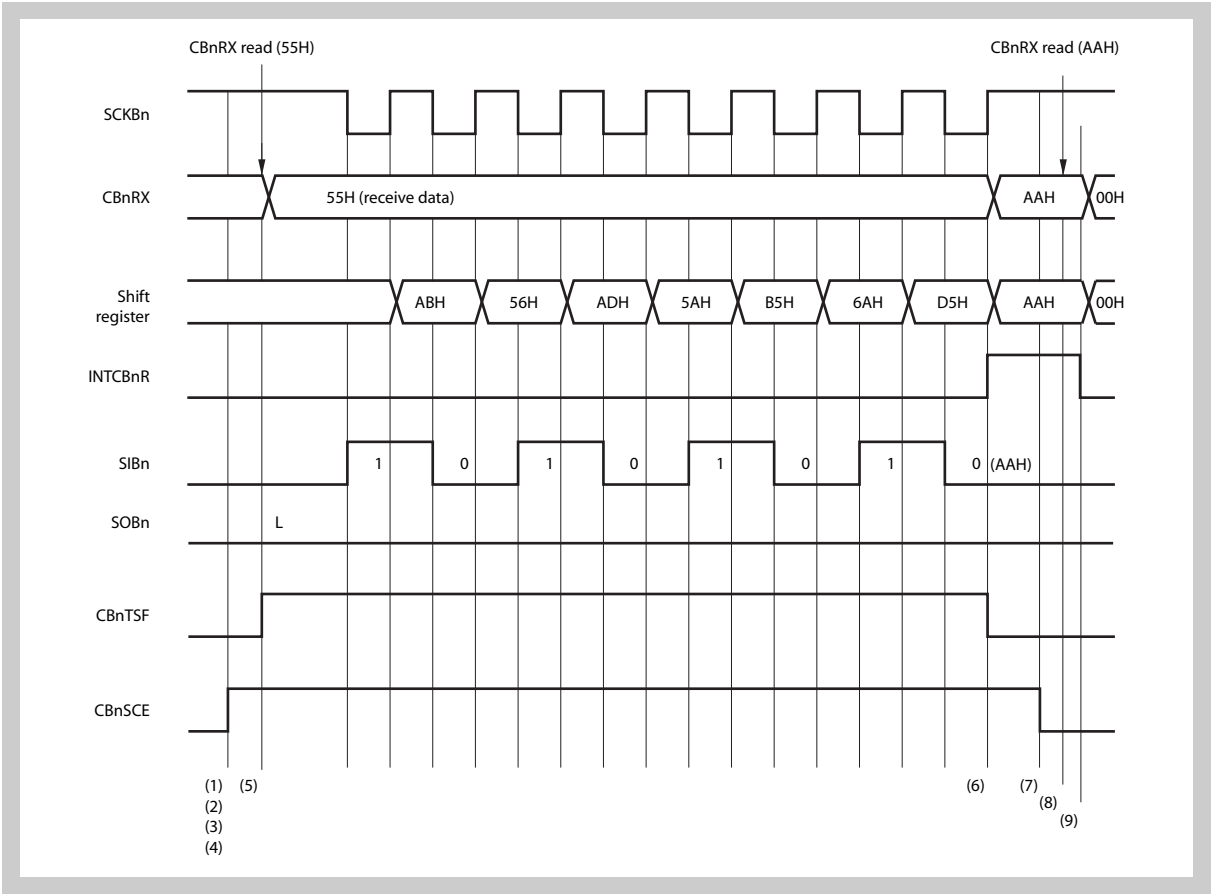


Figure 16-5 Timing diagram in single transfer mode (master mode, reception mode)

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 CBnRX 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 CBnRX 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.

16.4.3 Continuous mode (master mode, transmission/reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 3 (see 16.4 (2) CSIBn control register 1 (CBnCTL1)), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0, 0, 0, 0)

Figure 16-6

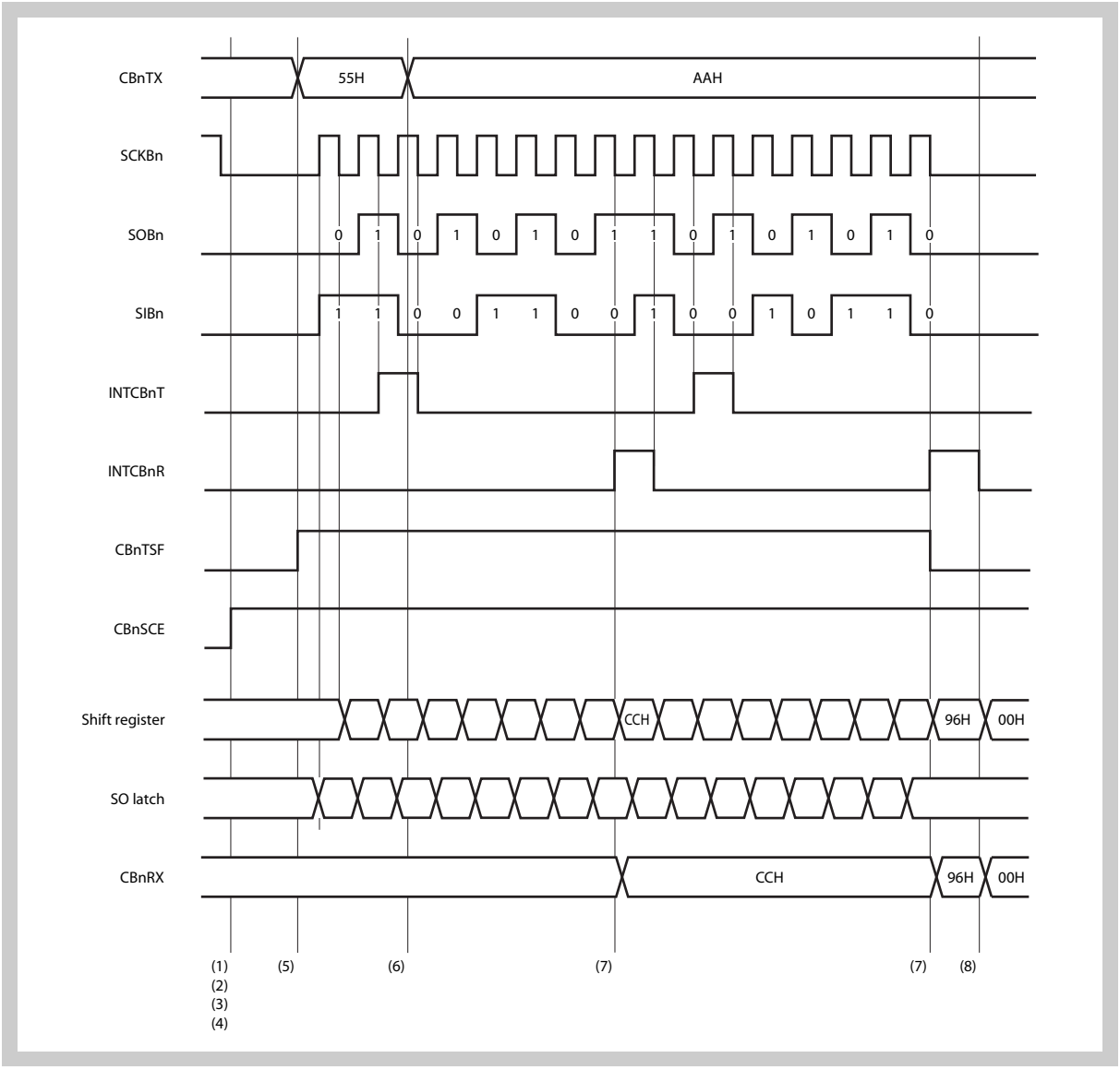


Figure 16-7 Timing diagram in continuous mode (master mode, transmission/reception mode)

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 CBnTX register (transmission/reception start).
6. The transmission enable interrupt request signal (INTCBnT) is received

and transfer data is written to the CBnTX register.

7. The reception complete interrupt request signal (INTCBnR) is output. Read the CBnRX register before the next receive data arrives or before the CBnPWR bit is cleared to 0.
8. Check that the CBnSTR.CBnTSF bit = 0 and 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 CBnRX register.

16.4.4 Continuous mode (master mode, reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 2 (see 16.4 (2) CSIBn control register 1 (CBnCTL1)), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0, 0, 0, 0)

Figure 16-8

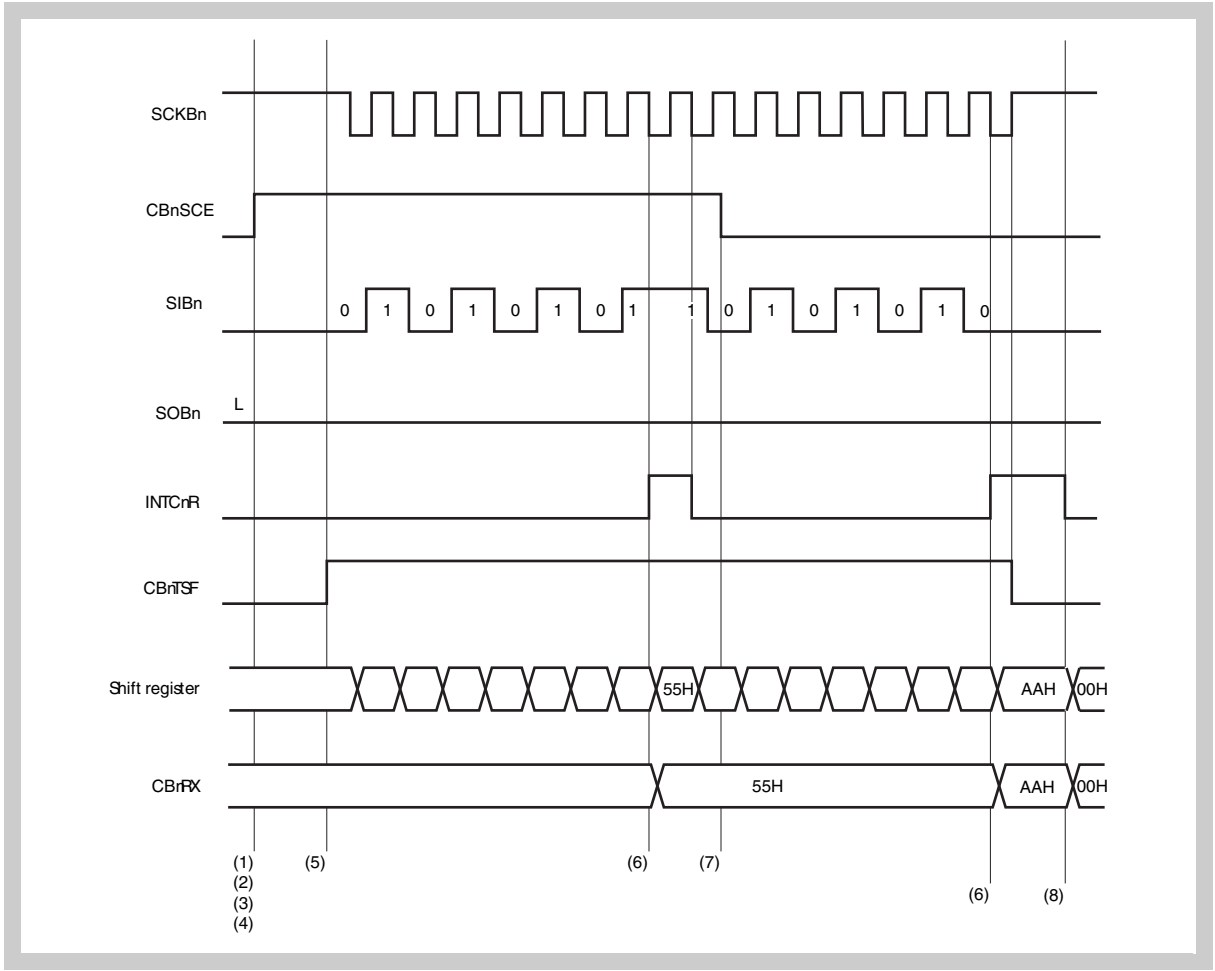


Figure 16-9 Timing diagram in continuous mode (master mode, reception mode)

1. Clear the CBnCTL0.CBnPWR bit to 0.
2. Set the CBnCTL1 and CBnCTL2 registers to specify the transfer mode.
3. Set the CBnCTL0.CBnRXE and CBnCTL0.CBnSCE bits to 1 at the same time as specifying the transfer mode using the CBnDIR bit, to set the

reception enabled status.

4. Set the CBnPWR bit to 1 to enable the CSIBn operation.
5. Perform a dummy read of the CBnRX register (reception start trigger).
6. The reception complete interrupt request signal (INTCBnR) is output. Read the CBnRX register 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).

16.4.5 Continuous reception mode (error)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 2 (see 16.4 (2) CSIBn control register 1 (CBnCTL1)), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0, 0, 0, 0)

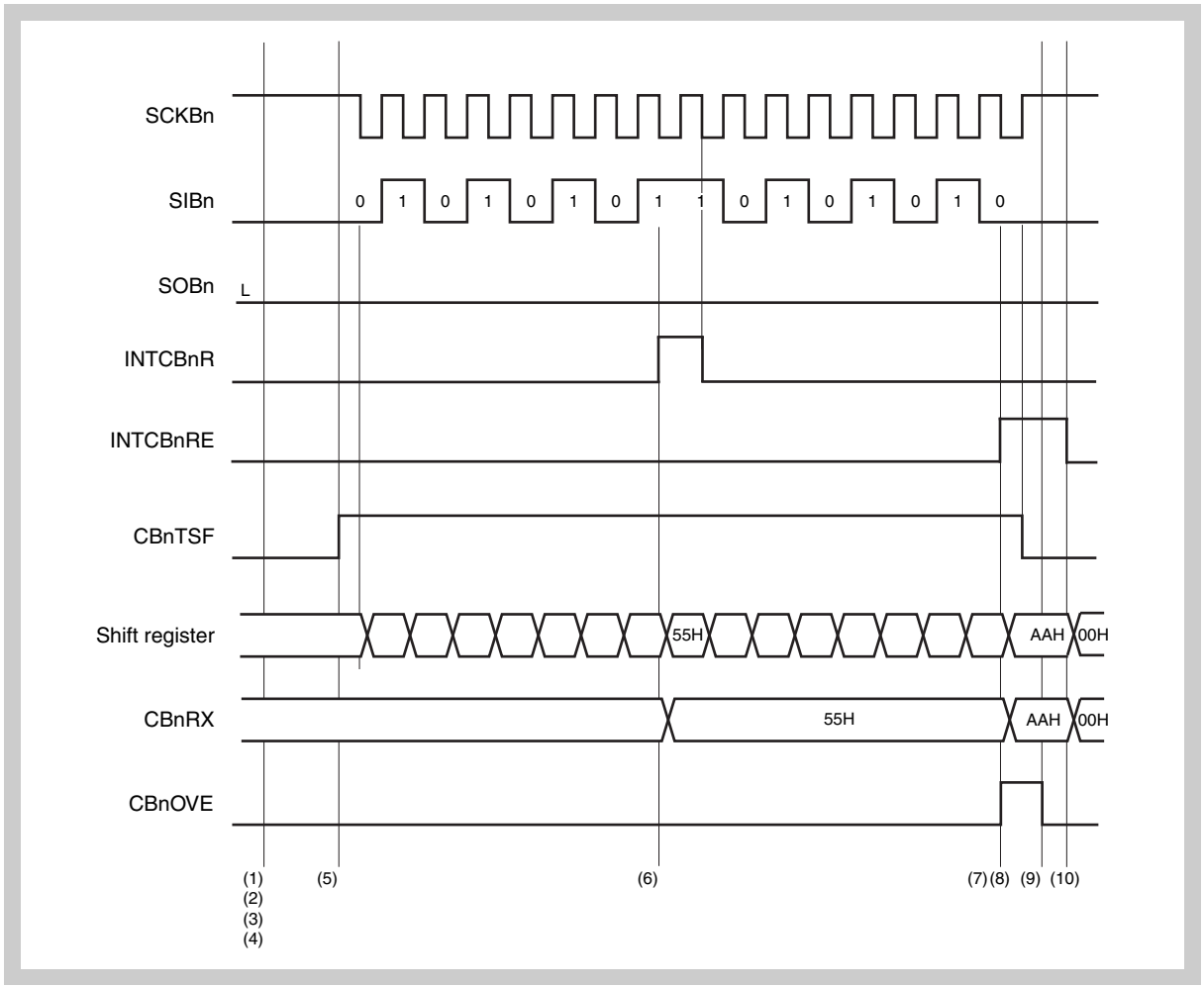


Figure 16-10 Timing diagram in continuous reception mode when overrun occurs

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 CBnRX register (reception start trigger).
6. The reception complete interrupt request signal (INTCBnR) is output.
7. If the data could not be read before the end of the next transfer, the CBnSTR.CBnOVE flag is set to 1 upon the end of reception and the INTCBnR signal is output.
8. Overrun error processing is performed after checking that the CBnOVE bit = 1 in the 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).

16.4.6 Continuous mode (slave mode, transmission/reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 2 (see 16.4 (2) CSIBn control register 1 (CBnCTL1)), transfer data length = 8 bits (CBnCTL2.CSnCL3 to CBnCTL2.CBnCL0 bits = 0, 0, 0, 0)

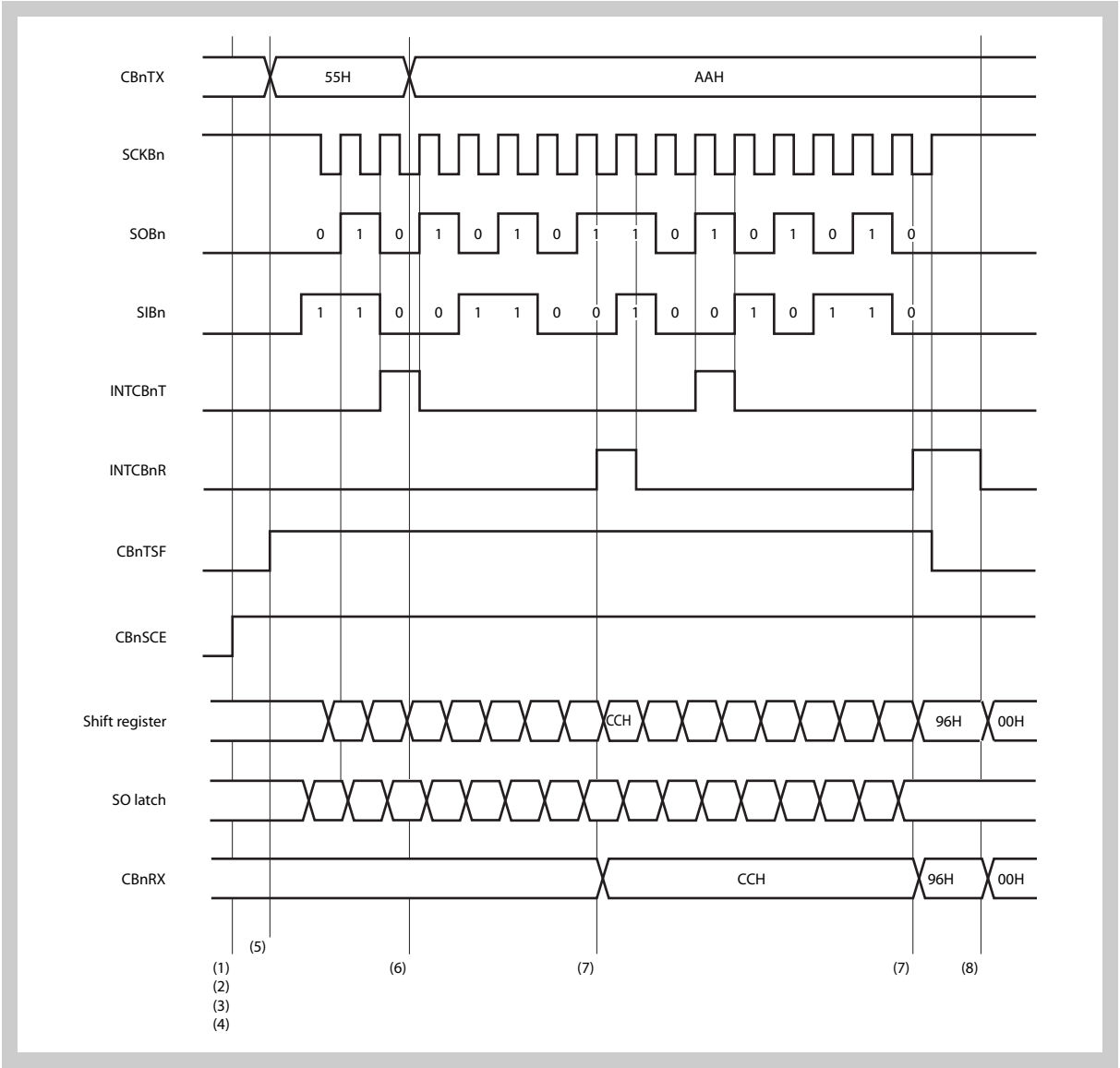


Figure 16-11 Timing diagram in continuous mode (slave mode, transmission/reception mode)

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 CBnTX register.
6. The transmission enable interrupt request signal (INTCBnT) is received and the transfer data is written to the CBnTX register.
7. The reception complete interrupt request signal (INTCBnR) is output.

Read the CBnRX register.

8. Check that the CBnSTR.CBnTSF bit = 0 and 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 (CBnCTL0.CBnTMS = 1, CBnCTL1.CBnCKS[2:0] = 111_B) and new data is not written to the CBnTX register the SOBn pin outputs the level of the last bit.

Figure 16-12 outlines this behaviour.

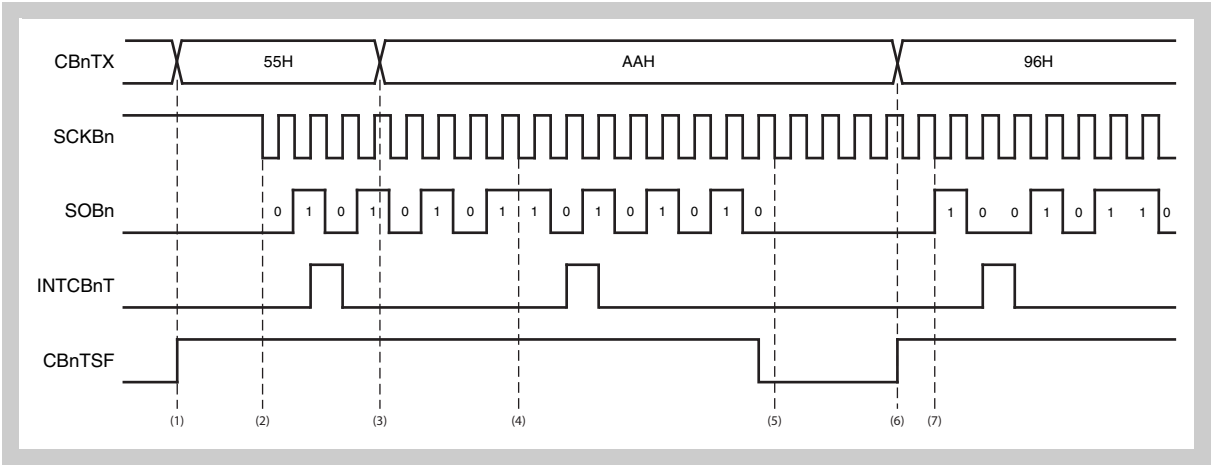


Figure 16-12 Discontinued slave transmission

The example shows the situation that two data bytes (55_H, AA_H) are transmitted correctly, but the third (96_H) fails.

(1) Data 55_H is written (by the CPU or DMA) to CBnTX.

(2) The master issues the clock SCKBn and transmission of 55_H starts.

(3) INTCBnT is generated and the next data AA_H is written to CBnTX promptly, i.e. before the first data has been transmitted completely.

(4) Transmission of the second data AA_H continues correctly and INTCBnT is generated. But this time the next data is not written to CBnTX in time.

(5) Since there is no new data available in CBnTX, but the master continuous to apply SCKBn clocks, SOBn remains at the level of the transmitted last bit.

(6) New data (96_H) is written to CBnTX.

(7) With the next SCKBn cycle transmission of the new data (96_H) 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.

16.4.7 Continuous mode (slave mode, reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (see 16.4 (2) CSIBn control register 1 (CBnCTL1)), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0, 0, 0, 0)

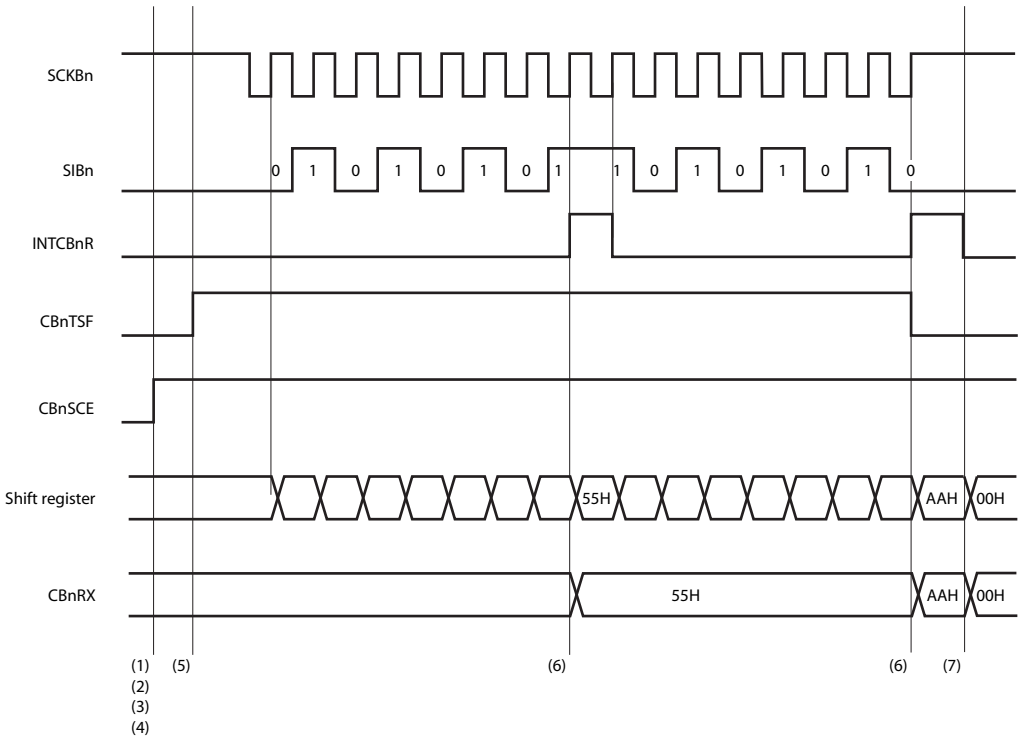


Figure 16-13 Timing diagram in continuous mode (slave mode, reception mode)

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 CBnRX register (reception start trigger).
6. The reception complete interrupt request signal (INTCBnR) is output. Read the CBnRX register.
7. Check that the CBnSTR.CBnTSF bit = 0 and 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).

16.4.8 Clock timing

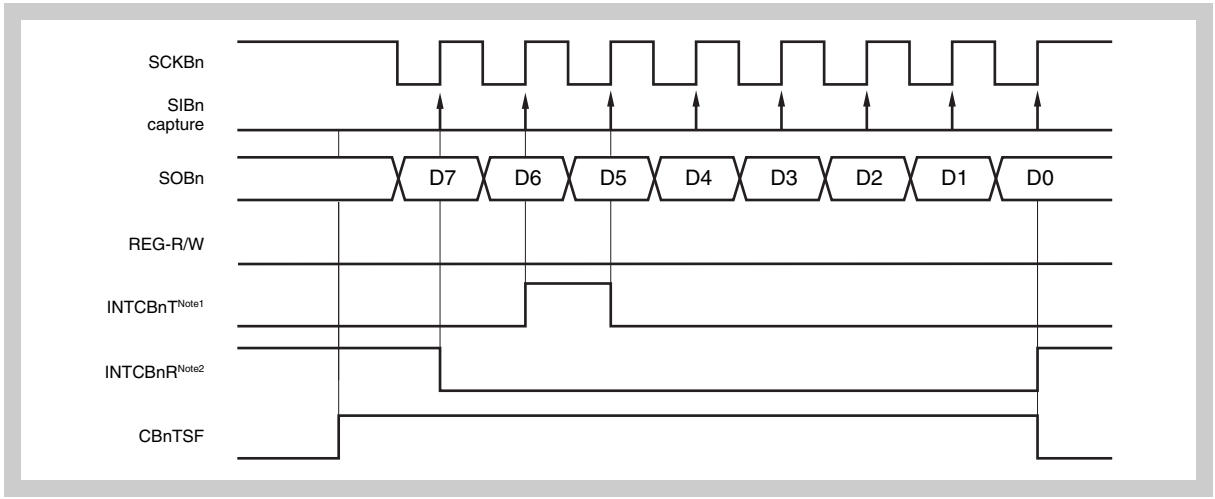


Figure 16-14 (i) Communication type 1 (CBnCKP = 0, CBnDAP = 0)

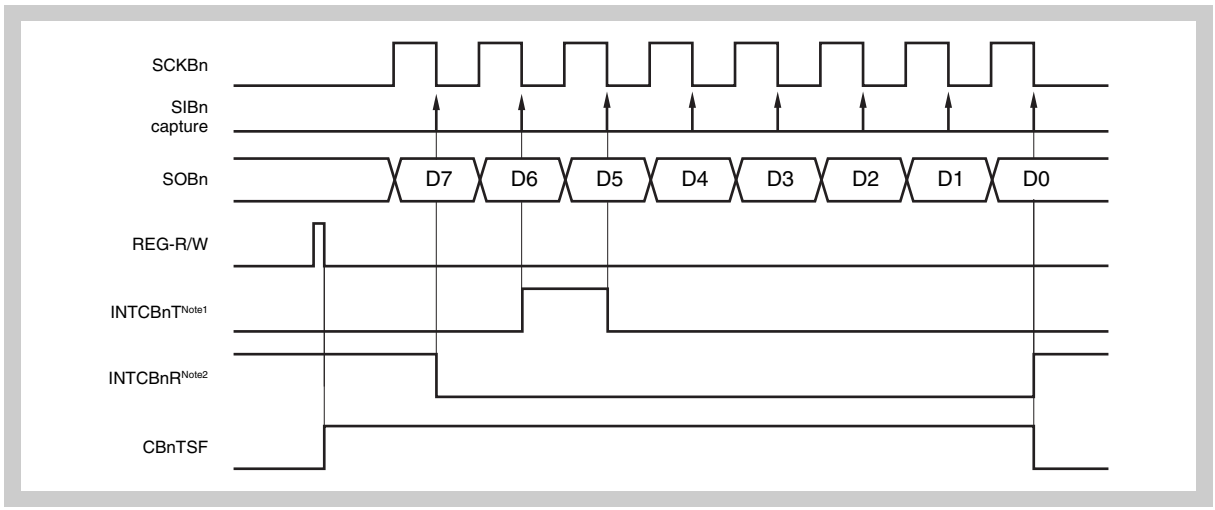


Figure 16-15 (ii) Communication type 3 (CBnCKP = 1, CBnDAP = 0)

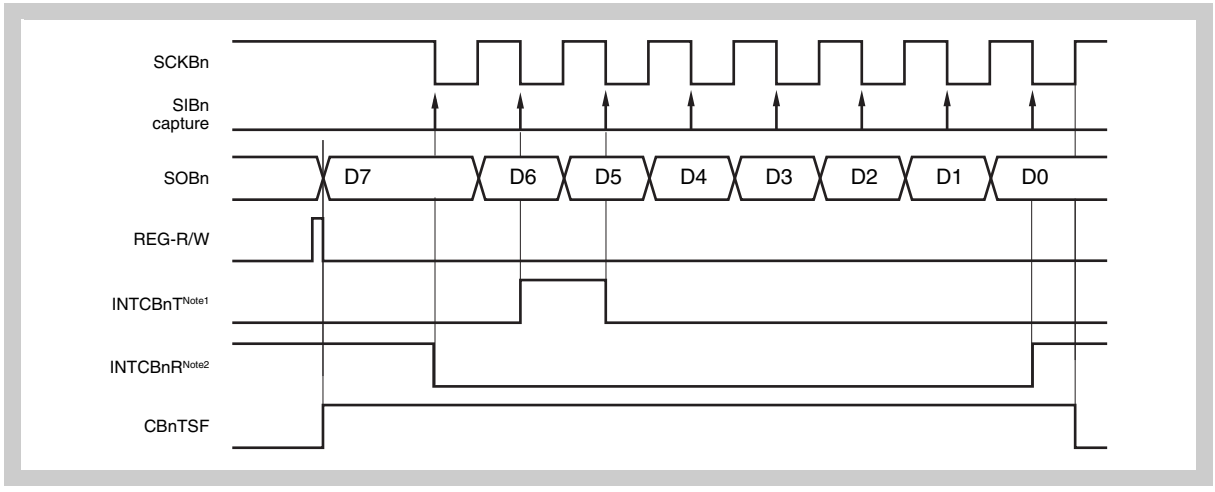


Figure 16-16 (iii) Communication type 2 (CBnCKP = 0, CBnDAP = 1)

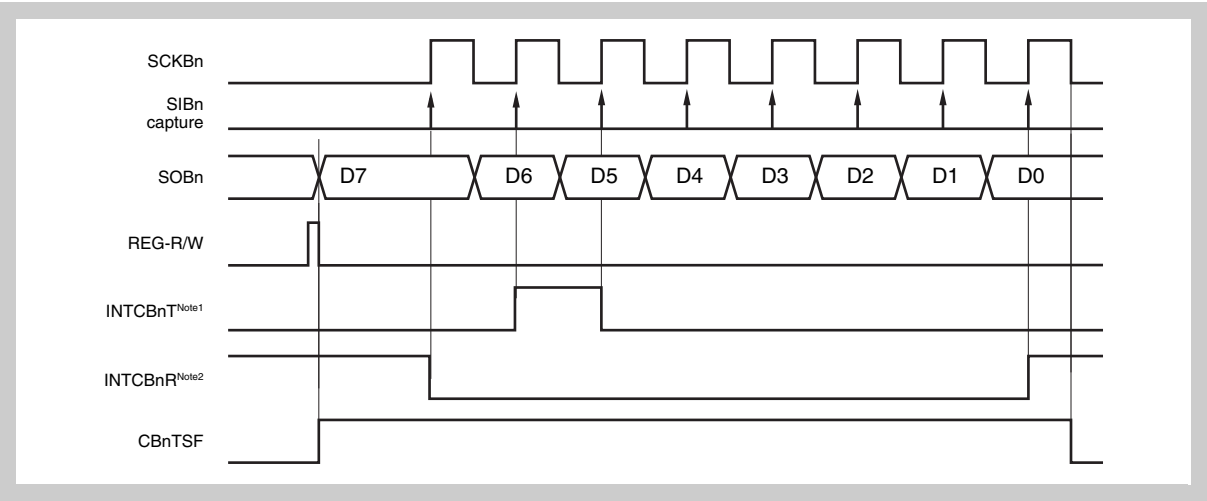


Figure 16-17 (iv) Communication type 4 (CBnCKP = 1, CBnDAP = 1)

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 CBnRX register while reception is enabled, and if an overrun error occurs. In the single mode, the INTCBnR interrupt request signal is generated even in the transmission mode, upon completion of communication.

16.5 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.

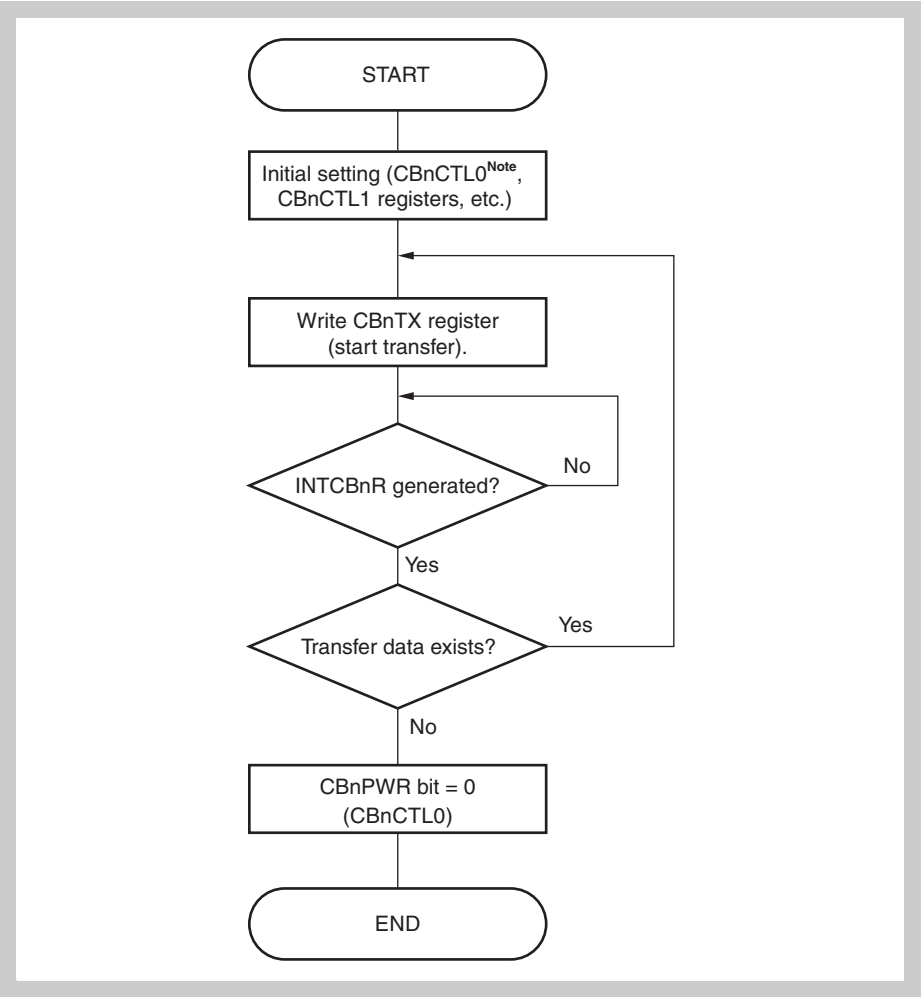
CBnTXE	CBnDAP	CBnDIR	SOBn pin output
0	×	×	Fixed to low level
1	0	×	SOBn latch value (low level)
	1	0	CBnTX value (MSB)
		1	CBnTX value (LSB)

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

16.6 Operation Flow

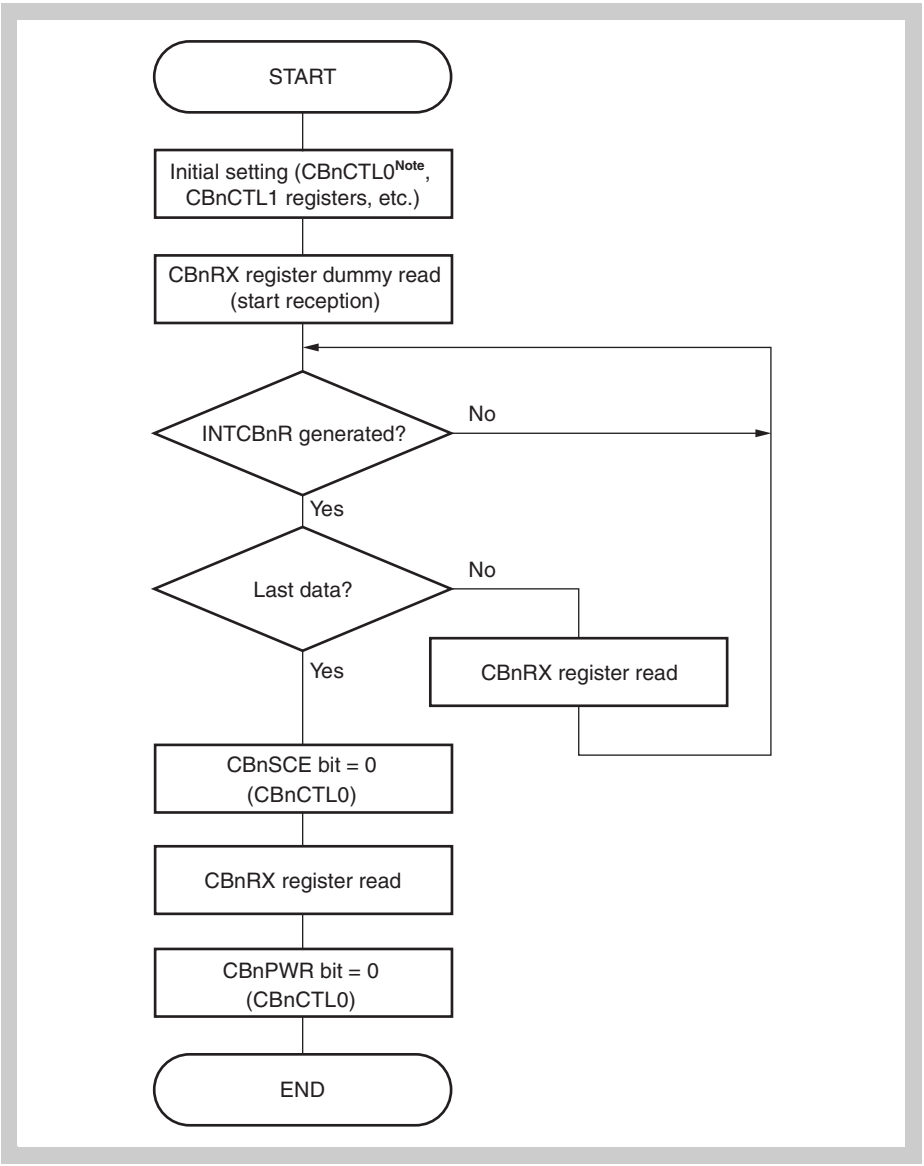
(1) Single transmission



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 CBnTX register is written.

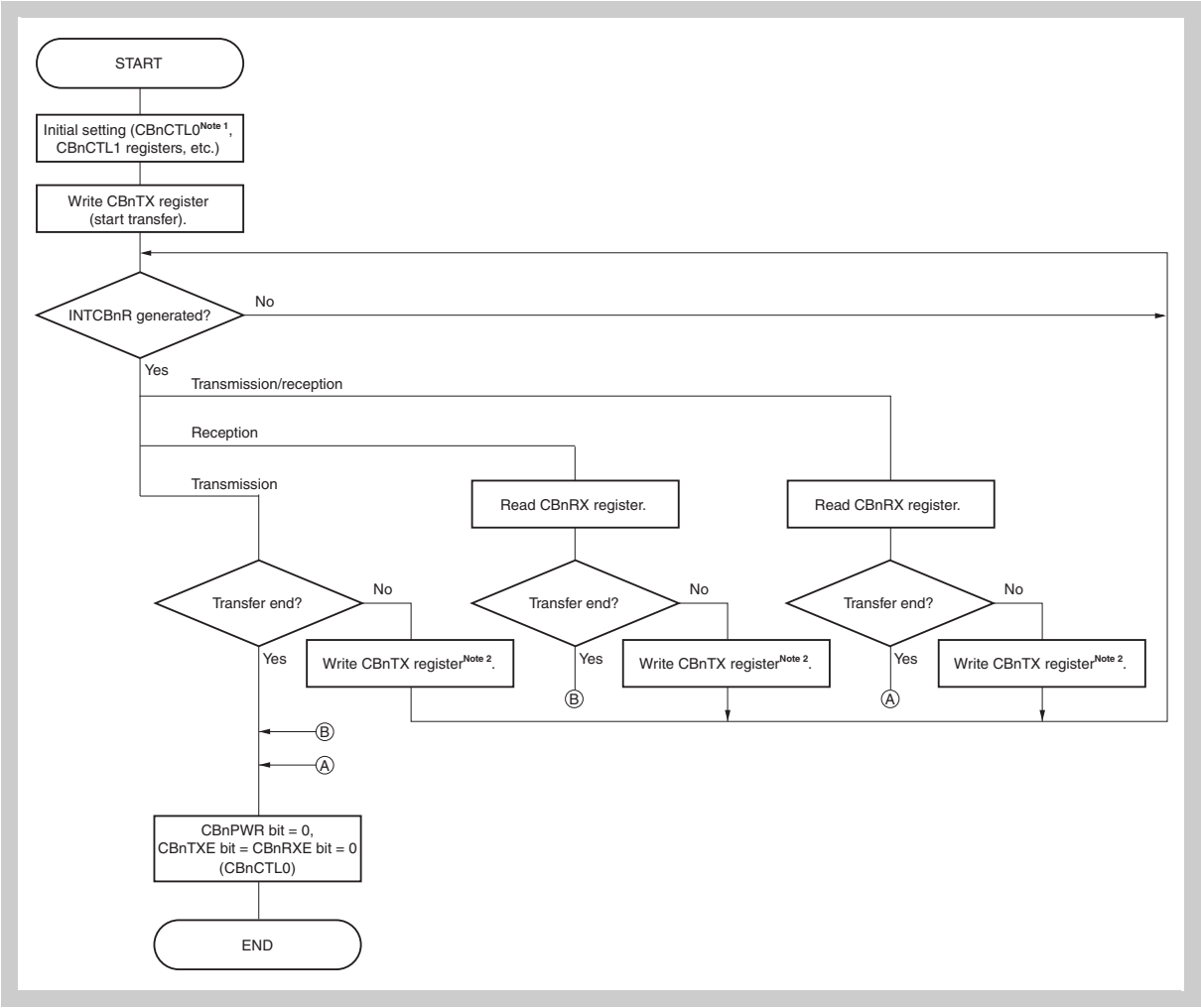
(2) Single reception



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 CBnRX register is read.

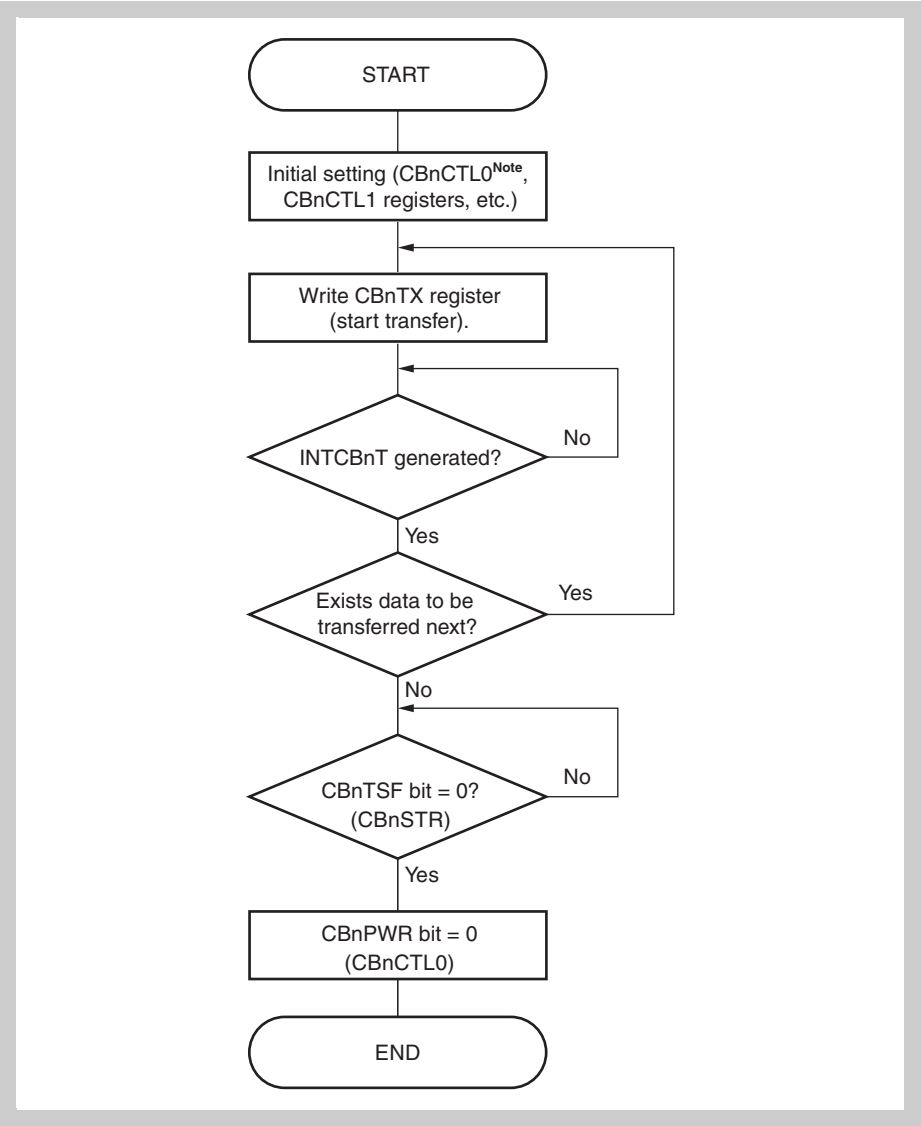
(3) Single transmission/reception



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 CBnTX register.

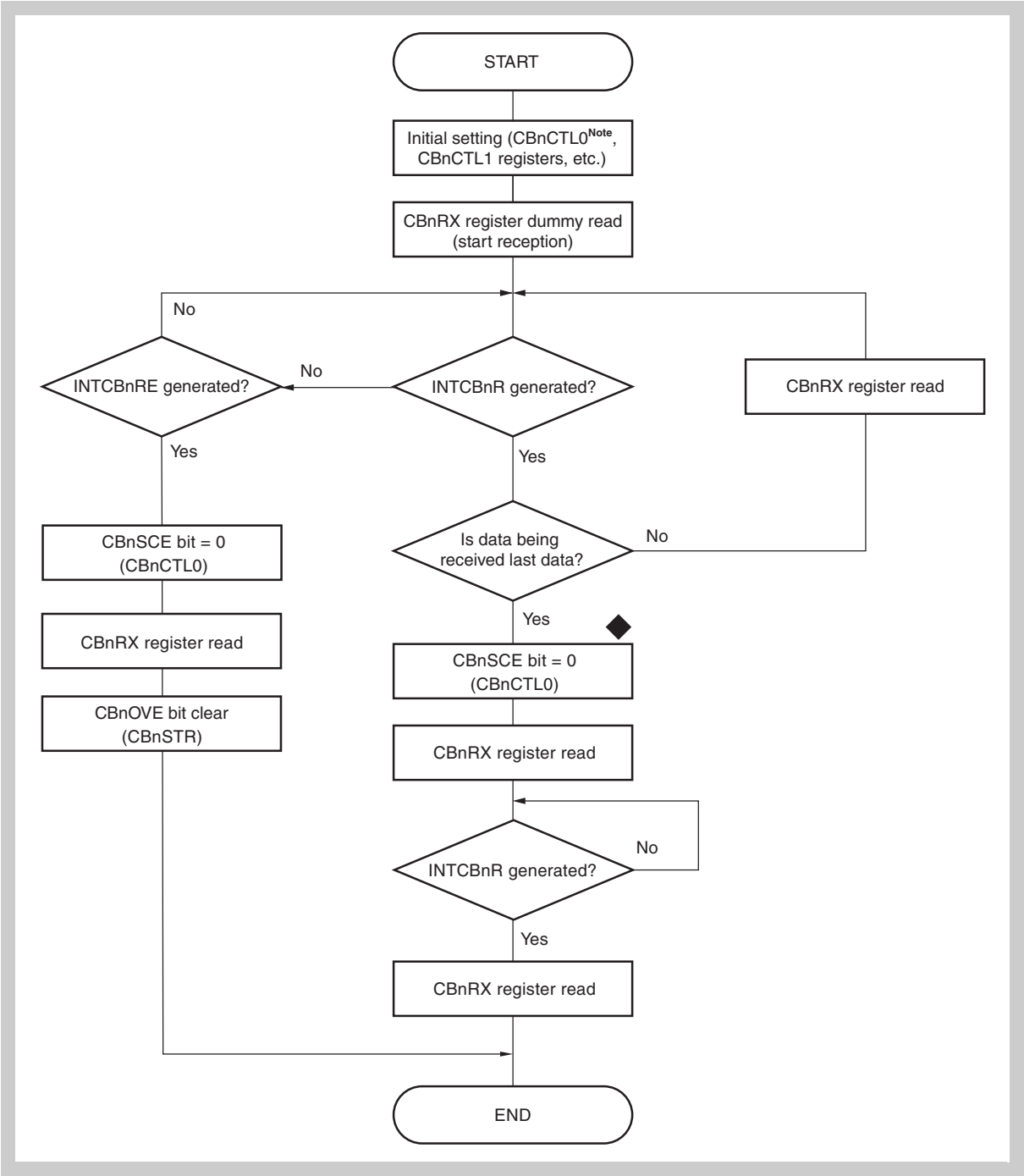
Caution Even in the single mode, the CBnSTR.CBnOVE flag is set to 1. If only transmission is used in the transmission/reception mode, therefore, programming without checking the CBnOVE flag is recommended.

(4) Continuous transmission



Note Set the CBnSCE bit to 1 in the initial setting.

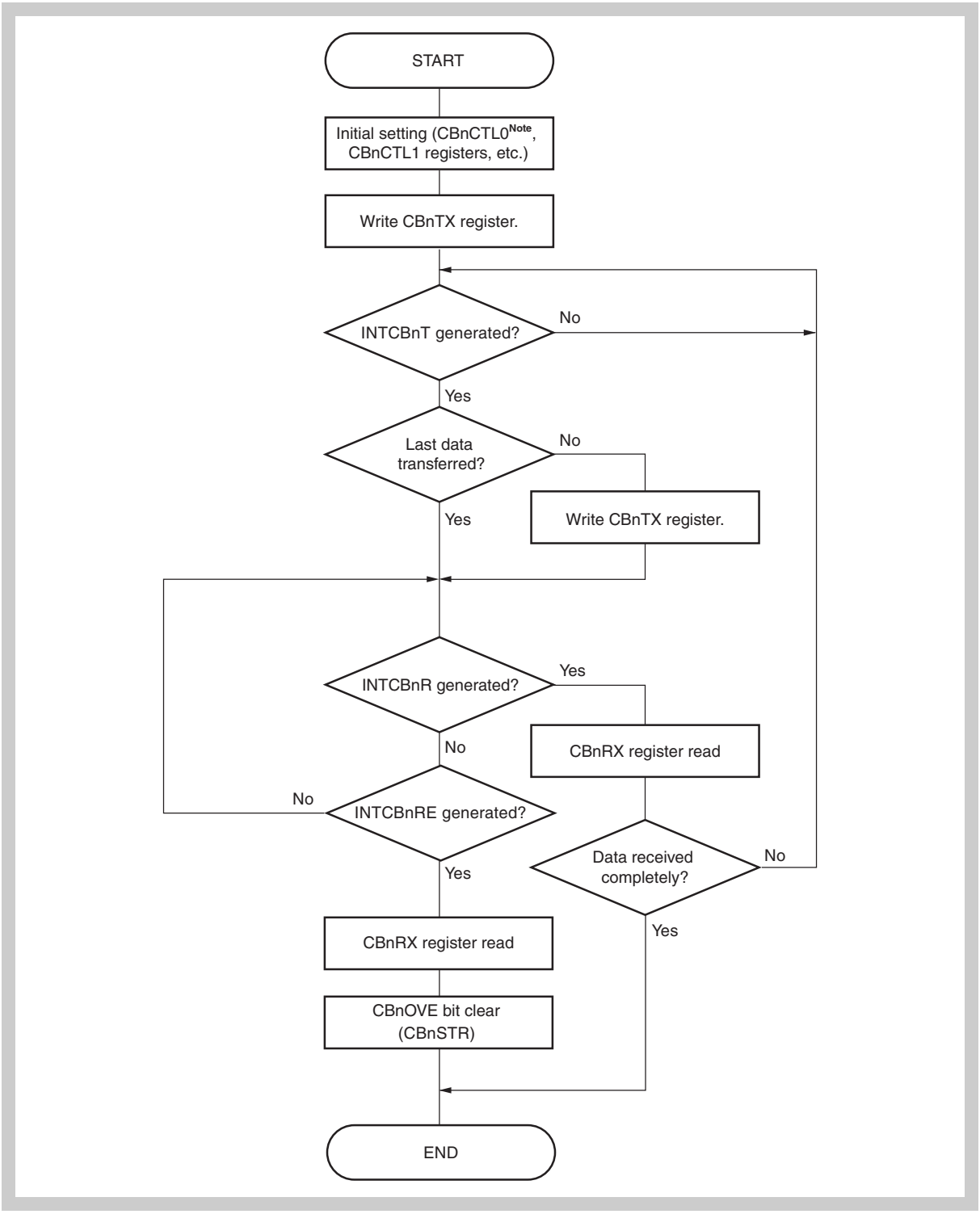
(5) Continuous reception



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 CBnRX register.

(6) Continuous transmission/reception



Note Set the CBnSCE bit to 1 in the initial setting.

16.7 Baud Rate Generator

16.7.1 Overview

Each CSIBSn interface is equipped with a dedicated baud rate generator.

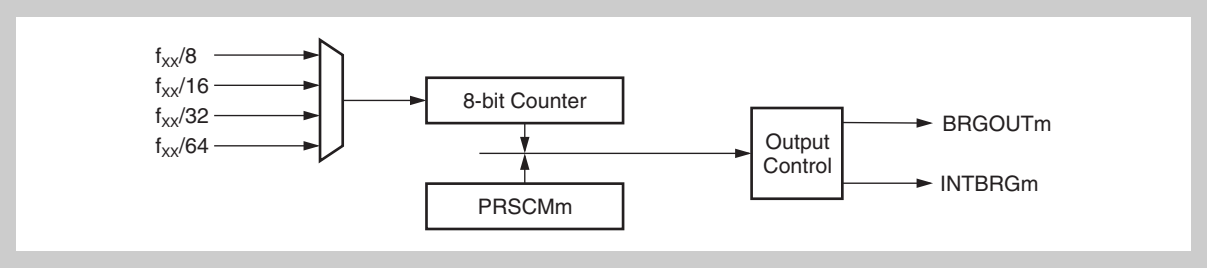


Figure 16-18 Block diagram of baud rate generator

The baud rate generators 0 and 1 (BRG0, BRG1) and CSIB0 and CSIB1 are connected as shown in the following block diagram.

Figure 16-19

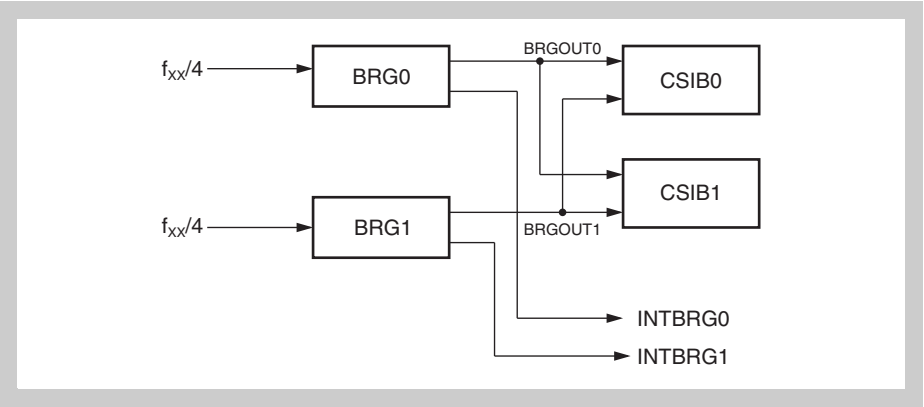


Figure 16-20 Interconnection of baud rate generators

Note 1. An unused baud rate generator (BRGm) can be employed as interval timer generating a dedicated interrupt request (INTBRGm).

2. CSIB1 is not available on µPD70F3447.

16.7.2 Baud rate generator registers

The Baud Rate Generators BRGn are controlled and operated by means of the following registers:

Table 16-9 BRGn registers overview

Register name	Shortcut	Address
BRGn prescaler mode register	PRSMn	<BRG_base>
BRGn prescaler compare register	PRSCMn	<BRG_base> + 1_H

Table 16-10 BRGn register base address

Timer	Base address <BRG_base>
BRG0	FFFF FDC0$_H$
BRG1	FFFF FDD0$_H$

(1) PRSMn - Prescaler mode registers

The PRSMn registers control generation of the baud rate signal for CSIB.

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

Address <BRG_base>

Initial Value 00_H. This register is cleared by any reset.

7	6	5	4	3	2	1	0
0	0	0	CEn	0	0	BGCSn1	BGCSn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16-11 PRSMn register contents

Bit position	Bit name	Function
4	CEn	Baud rate genrator output control 0: disabled 1: enabled
1 to 0	BGCSn[1:0]	Input clock selection

BGCSn1	BGCSn0	Input clock selection (f_{BGCSn})	Setting value k
0	0	$f_{XX}/4$	2
0	1	$f_{XX}/8$	3
1	0	$f_{XX}/16$	4
1	1	$f_{XX}/32$	5

Caution
1. Do not rewrite the PRSMn register during operation.
2. Set the BGCSn[1:0] bits before setting the CEn bit to 1.

(2) PRSCMn - Prescaler compare registers

The PRSCMn registers are 8-bit compare registers.

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

Address <BRG_base> + 1_H

Initial Value 00_H. This register is cleared by any reset.

7	6	5	4	3	2	1	0
PRSCMn7	PRSCMn6	PRSCMn5	PRSCMn4	PRSCMn3	PRSCMn2	PRSCMn1	PRSCMn0
R/W							

Caution
1. Do not rewrite the PRSCMn register during operation.
2. Set the PRSCMn register before setting the PRSMn.CEn bit to 1.

16.7.3 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.

$$f_{BRGn} = \frac{f_{BGCSn}}{N \times 2} = \frac{f_{XX}}{2^k \times N \times 2}$$

f_{BRGn}: BRGn output clock

f_{BGCSn}: Clock frequency selected by the BGCSn1, BGCSn0 bits of the PRSMn register.

f_{XX}: Main clock oscillation frequency

k: PRSMn.BGCSn[1:0] register setting value ($2 \leq k \leq 5$)

N: PRSCMn.PRSCMn[7:0] register value
if PRSCMn = 00H: N = 256

16.8 Cautions

(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 CBnTX register may trigger the aforementioned behaviour.
  Any write to the related CBnTX 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 CBnRX register may trigger the aforementioned behaviour.
  Any read from the related CBnRX 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 CBnTX 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 CBnRX register after a reception sequence is finished.

Chapter 17 Clocked Serial Interface 3 (CSI3)

The V850E/PH2 microcontrollers have following number of channels of the clocked cerial interface 3 (CSI3):

CSIB	µPD70F3447	µPD70F3187
Instances	1	2
Names	CSI30	CSI30 to CSI31

Note Throughout this chapter, the individual instances of CSI3 are identified by “n” (n = 0, 1).

17.1 Features

- Transfer rate: Maximum 8 Mbps
- Master mode and slave mode selectable
- Serial clock and data phase switchable
- Transmission data length: 8 to 16 bits (selectable in 1-bit units)
- Transfer data MSB-first/LSB-first switchable
- Transmission mode, reception mode, and transmission/reception mode selectable
- 3-wire serial interface
- SO3n: Serial data output
- SI3n: Serial data input
- $\overline{\text{SCK3n}}$: Serial clock I/O
- Four external chips select signal outputs (SCS3n0 to SCS3n3)
- Interrupt request signals × 2
- Transmission/reception completion interrupt (INTC3n)
- CSIBUFn overflow interrupt (INTC3nOVF)
- Sixteen on-chip 20-bit transmit/receive buffers (CSIBUFn)
- On-chip dedicated baud rate generator

17.2 Configuration

CSI3n is controlled by the clocked serial interface mode register 3n (CSIM3n).

(1) Clocked serial interface mode register 3n (CSIM3n)

The CSIM3n register is an 8-bit register for specifying the operation of CSI3n.

(2) Clocked serial interface clock select register 3n (CSIC30, CSIC31)

The CSIC3n register is an 8-bit register for controlling the operation clock and operating mode of CSI3n.

(3) Serial I/O shift register 3n (SIO3n)

The SIO3n register is an 8-bit register for converting between serial data and parallel data. SIO3n is used for both transmission and reception.

Data is shifted in (reception) or shifted out (transmission) beginning at either the MSB side or the LSB side.

(4) Receive data buffer register 3n (SIRB3n)

The SIRB3n register is a 16-bit buffer register that stores receive data. This register is also divided into two registers: the higher 8 bits (SIRB3nH) and lower 8 bits (SIRB3nL).

(5) Chip select CSI buffer register 3n (SFCS3n)

The SFCS3n register is a 16-bit buffer register that stores chip select data. The lower 8 bits can also be accessed by an 8-bit buffer register (SFCS3nL).

(6) Transmit data CSI buffer register 3n (SFDB3n)

The SFDB3n register is a 16-bit buffer register that stores transmit data. This register is also divided into two registers: the higher 8 bits (SFDB3nH) and lower 8 bits (SFDB3nL).

(7) CSIBUF status register 3n (SFA3n)

The SFA3n register is an 8-bit register that indicates the status of CSI data buffer register n (CSIBUFn) or the transfer status.

(8) Transfer data length select register 3n (CSIL3n)

The CSIL3n register is an 8-bit register that selects the CSI3n transfer data length.

(9) Transfer data number specification register 3n (SFN3n)

The SFN3n register is an 8-bit register that sets the number of CSI3n transfer data in consecutive mode.

(10) CSI data buffer register n (CSIBUFn)

By consecutively writing transmit data to the SFDB3n register from where it is transferred, the data can be stored in the CSIBUFn register while the CSIBUFn pointer for writing is automatically
incremented (CSIBUFn).

The CSIBUFn is a 16-bit buffer register.

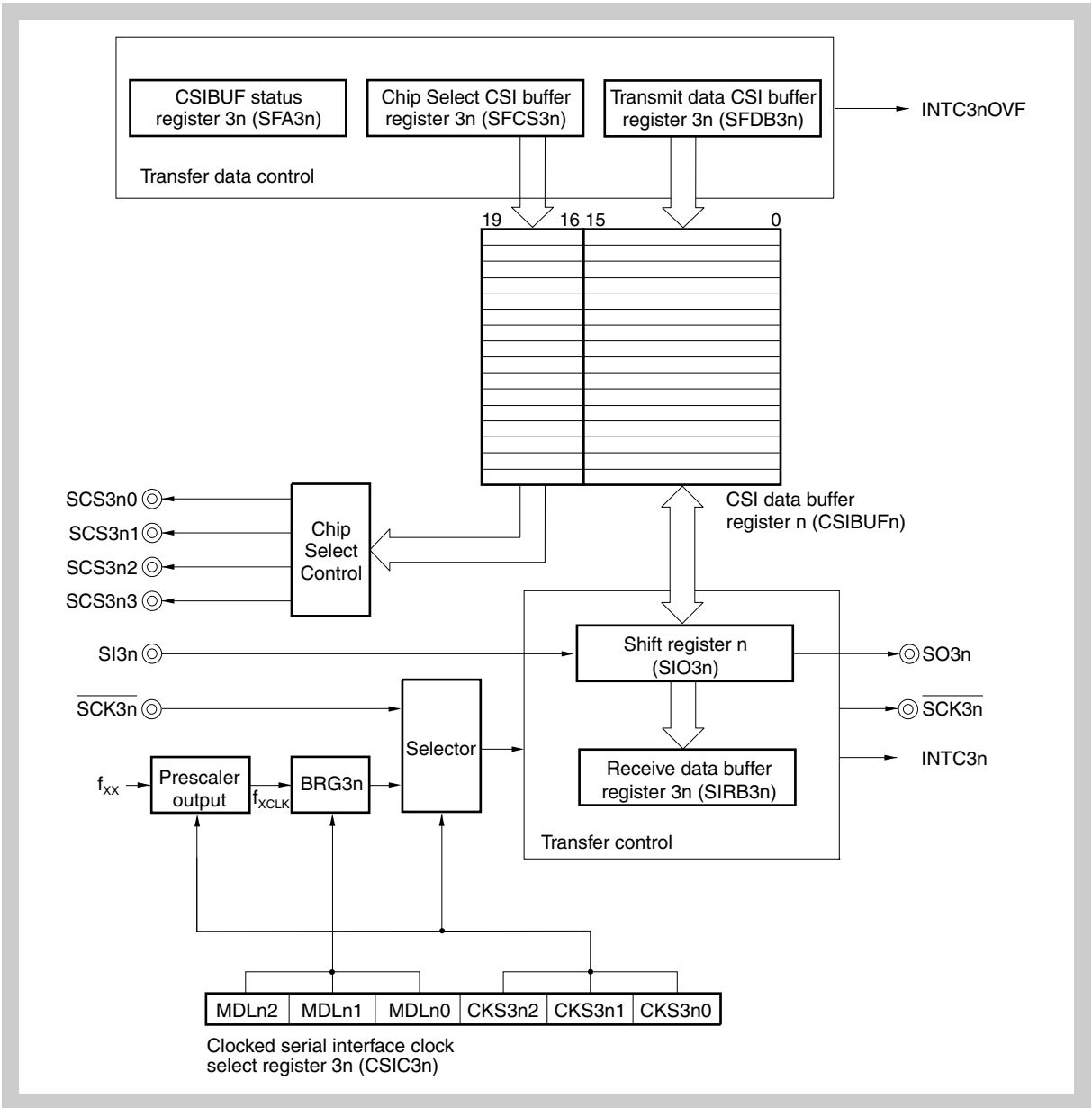


Figure 17-1 Block Diagram of Clocked Serial Interface 3n (CSI3n)

Remark f_{XX}: Main clock
f_{XCLK}:Basic clock specified by CSIC3n.CKS3n[2:0]

17.3 Control Registers

The clocked serial interfaces CSI3n are controlled and operated by means of the following registers:

Table 17-1 CSIBn registers overview

Register name	Shortcut	Address
CSI3n mode register	CSIM3n	<base>
CSI3n clock select register	CSIC3n	<base> + 1_H
CSI3n receive data buffer register	SIRB3n	<base> + 2_H
CSI3n chip select buffer egister	SFCS3n	<base> + 4_H
CSI3n transmit data buffer register	SFDB3n	<base> + 6_H
CSI3n buffer status register	SFA3n	<base> + 8_H
CSI3n transfer data length select register	CSIL3n	<base> + 9_H
CSI3n transfer data number specification register	SFN3n	<base> + C_H

Table 17-2 CSIBn register base address

Channel	Base address
CSI30	FFFF $FD40_H$
CSI31	FFFF $FD60_H$

(1) Clocked serial interface mode registers 3n (CSIM3n)

The CSIM3n register controls the operation of CSI3n.

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

Address <base>

Initial Value 00_H. This register is cleared by any reset.

Caution
1. Writing the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits is enabled only when CTXEn bit = 0 and CRXEn bit = 0.
2. To use CSI3n, be sure to set the external pins related to the CSI3n function to control the mode and set the CSIC3n register. Then set the CSICAEn bit to 1 before setting the other bits.

CSIM3n

7	6	5	4	3	2	1	0
CSICAEn	CTXEn	CRXEn	TRMDn	DIRn	CSITn	CSWEn	CSMDn
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17-3 CSIM3n register contents (1/2)

Bit position	Bit name	Function
7	CSICAEn	CSI3n operation clock control 0: Stops clock supply to CSI3n 1: Supplies clock to CSI3n **Caution:** 1. The CSI3n unit is reset when the CSICAEn bit = 0, and CSI3n is stopped. To operate CSI3n, first set the CSICAEn bit to 1. 2. When changing the CSICAEn bit from 0 to 1 or from 1 to 0, simultaneously rewriting of bits other than the CSICAEn bit is prohibited. When the CSICAEn bit = 0, rewriting of bits other than the CSICAEn bit, and the SFDB3n, SFDB3nL, and SFA3n registers is prohibited.
6	CTXEn	Transmission operation enable 0: Disables transmit operation 1: Enables transmit operation **Caution:** The CTXEn bit is reset when the CSICAEn bit is cleared to 0.
5	CRXEn	Receive operation enable 0: Disables receive operation 1: Enables receive operation **Caution:** The CRXEn bit is reset when the CSICAEn bit is cleared to 0.
4	TRMDn	Transfer mode specification 0: Single transfer mode 1: Consecutive mode Specifies the transfer direction when data is written from the SFDB3n register to the CSIBUFn register or read from the SIRB3n and CSIBUFn registers.

Table 17-3 CSIM3n register contents (2/2)

Bit position	Bit name	Function
3	DIRn	Transfer direction specification 0: MSB first transfer 1: LSB first transfer
2	CSITn	Transmission completion interrupt (INTC3n) control 0: No delay 1: Delay mode (The interrupt request signal is delayed by half a cycle.) **Caution:** 1. The delay mode (CSIT bit = 1) is valid only in the master mode (CSIC3n.CKS3n[2:0] other than 111_B). In the slave mode (CSIC3n.CKS3n[2:0] = 111_B), do not set the delay mode. If the delay mode is set, INTC3n is not affected by the CSITn bit. 2. If the CSITn bit is set to 1 in the consecutive mode (TRMDn = 1), the INTC3n interrupt is not output except when the last data set by the SFN3n.SFNn[3:0] bits is transferred, but a delay of half a clock can be inserted between each data transferred.
1	CSWEn	Transfer wait control 0: Disables transfer wait. 1: Enables transfer wait (1 wait cycle inserted on starting transfer). **Caution:** Inserting a transfer wait cycle (CSWEn = 1) is valid only in the master mode (CSIC3n.CKS3n[2:0] other than 111_B. In the slave mode (CSIC3n.CKS3n[2:0] = 111_B), do not insert a transfer wait cycle. If set, a transfer wait cycle is not inserted.
0	CSMDn	Chip Select Mode Specification 0: Disables inactive level setting of chip select outputs (SCS3n0 to SCS3n3) during transfer wait. 1: Enables inactive level setting of chip select outputs (SCS3n0 to SCS3n3) during transfer wait. **Caution:** The CSMDn bit setting is valid only when the transfer wait is enabled (CSWEn bit = 1) and the master mode is specified (CKS3n0 bits of the CSIC3n register other than 111_B). In all other cases the CSMDn bit setting is invalid and no inactive level setting of chip select outputs between two consecutive transfers takes place.

(2) Clocked serial interface clock select register 3n (CSIC3n)

The CSIC3n register is an 8-bit register that controls the operation clock and operating mode of CSI3n.

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

Address <base> + 1_H

Initial Value 07_H

Caution Data can be written to the CSIC3n register only when the CSIM3n.CTXEn = 0 and CSIM3n.CRXEn = 0.

CSIC3n

7	6	5	4	3	2	1	0
MDLn2	MDLn1	MDLn0	CKPn	DAPn	CKS3n2	CKS3n1	CKS3n0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

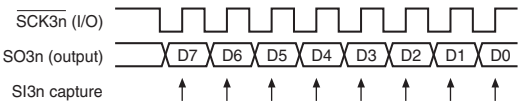
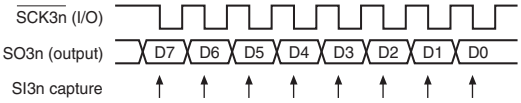
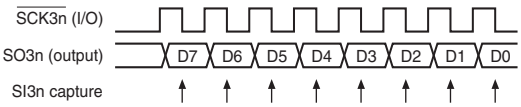
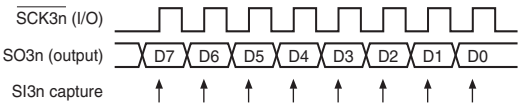
Table 17-4 CSIC3n register contents (1/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7 to 5</td><td>MDLn[2:0]</td><td>Transfer clock setting (BRG3 output signal)
<table>
<tr><th>MDLn2</th><th>MDLn1</th><th>MDLn0</th><th>Transfer clock</th><th>Set Value (N)</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>BRG3n stop mode (power save)</td><td>–</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>$f_{XCLK}/2$</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>$f_{XCLK}/4$</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>$f_{XCLK}/8$</td><td>3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>$f_{XCLK}/10$</td><td>4</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>$f_{XCLK}/12$</td><td>5</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>$f_{XCLK}/14$</td><td>6</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>External clock ($\overline{SCKBn}$)</td><td>7</td></tr>
</table>
Caution: In the slave mode (CKS3n[2:0] = 111_B), it is recommended to clear the MDLn[2:0] bits to 000_B (BRG3n stop mode).</td></tr>
<tr><td>4, 3</td><td>CKPn
DAPn</td><td>Specification of data transmission/reception timing in relation to $\overline{SCK3n}$.
Refer to Table 17-5.</td></tr>
</table>

Table 17-4 CSIC3n register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>2 to 0</td><td>CKS3n[2:0]</td><td>Basic clock setting (f_{XCLK})
<table>
<tr><th>CKS3n2</th><th>CKS3nS1</th><th>CKS3n0</th><th>Basic clock (f_{XCLK})</th><th>Mode</th><th>Set value (k)</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>f_{XX}</td><td>Master</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>$f_{XX}/2$</td><td>Master</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>$f_{XX}/4$</td><td>Master</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>$f_{XX}/8$</td><td>Master</td><td>3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>$f_{XX}/16$</td><td>Master</td><td>4</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>$f_{XX}/32$</td><td>Master</td><td>5</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>$f_{XX}/64$</td><td>Master</td><td>6</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>External clock ($\overline{SCK3n}$)</td><td>Slave</td><td>–</td></tr>
</table>
Remark: f_{XX}: Main clock</td></tr>
</table>

Table 17-5 Specification of data transmission/reception timing in relation to $\overline{SCK3n}$

CKPn	DAPn	Specification of Data Transmission/Reception Timing in Relation to Clock Phase
0	0	
0	1	
1 Note	0	
1 Note	1	

Note If the CKPn bit is set to 1 in the master mode (CKS3n[2:0] bits other than 111_B), the $\overline{SCK3n}$ pin outputs a low level when it is inactive. If the CTXEn bit of the CSIM3n register is cleared to 0 (disabling transmission) and CRXEn bit is cleared to 0 (disabling reception), the $\overline{SCK3n}$ pin outputs a high level.

Therefore, take the following measures to fix the $\overline{\text{SCK3n}}$ pin to low level when CSI3n is not used.

[$\overline{\text{SCK3n}}$ pin]

<1> Clear the corresponding port bit (P82 of the P8 register for CSI30, or P92 of the P9 register for CSI31) to 0:
The port output level is set to low.

<2> Clear the corresponding bit in the port mode register (PM82 of the PM8 register for CSI30, or PM92 of the PM9 register for CSI31) to 0:
The pin is set into output mode.

<3> Clear the corresponding bit in the port mode control register (PMC82 of the PMC8 register for CSI30, or PMC92 of the PMC9 register for CSI31) to 0:
The pin is set into port mode (fixed to low-level output).

<4> Clear bits CSIM3n.CTXEn and CSIM3n.CRXEn to 0:
Transmission and reception are disabled.

<5> Set bits CSIM3n.CTXEn and CSIM3n.CRXEn to 1:
Transmission or reception is enabled (both transmission and reception can also be enabled).

<6> Set the corresponding bit in the port mode control register (PMC82 of the PMC8 register for CSI30, or PMC92 of the PMC9 register for CSI31) to 1:
The pin is set in the control mode ($\overline{\text{SCK3n}}$ pin output).

Because the register set values <1> and <2> are retained, control can be performed only by <3> to <6> once they have been set.

(3) Receive data buffer register 3n (SIRB3n, SIRB3nL, SIRB3nH)

The SIRB3n register is a 16-bit buffer register that stores receive data. It is overlayed by an 8-bit buffer register SIRB3nL on the lower 8 bits, and an 8-bit buffer register SIRB3nH on the higher 8 bits.

By consecutively reading this register in the consecutive mode (CSIM3n.TRMDn = 1), the received data in the CSIBUFn register can be sequentially read while the CSIBUFn pointer for reading is incremented.

In the single mode (CSIM3n.TRMDn = 0), received data is read by reading the SIRB3n register and it is judged that the SIRB3n register has become empty.

Access The SIRB3n register is read-only, in 16-bit units.
The SIRB3nL and SIRB3nH registers are read-only, in 8-bit units.

Address SIRB3n, SIRB3nL: <base> + 2_H
SIRB3nH: <base> + 2_H

Initial value 0000H, or 00H respectively. This register is cleared by any reset.
In addition to reset input it can be initialized in single mode (CSIM3n.TRMDn = 0) by clearing the CSIM3n.CSICAEn bit to 0.
In consecutive mode (CSIM3n.TRMDn = 1) the inital value is undefined.

SIRB3n

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

SIRB3nL

7	6	5	4	3	2	1	0
Receive data (lower 8 bits)							
R							

SIRB3nH

7	6	5	4	3	2	1	0
Receive data (higher 8 bits)							
R							

(4) Chip select CSI buffer register 3n (SFCS3n, SFCS3nL)

The SFCS3n register is a 16-bit buffer register that stores transmit data. It is overlayed by an 8-bit buffer register SFCS3nL on the lower 8 bits.

When chip select data is written to the SFCS3n (SFCS3nL) register, the data is stored in the CSIBUFn register following the CSIBUFn pointer for writing. The store operation is executed after next write of the transmit data CSI buffer register SFDB3n (SFDB3nL).

When the data of this register is read, the value of the transmit data written last is read.

Access The SFCS3n register can be read/written in 16-bit units.
The SFCS3nL register can be read/written in 8-bit and 1-bit units.

Address <base> + 4_H

Initial value $FFFF_H$, or FF_H respectively. This register is cleared by any reset.

SFCS3n

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SFCSn15	SFCSn14	SFCSn13	SFCSn12	SFCSn11	SFCSn10	SFCSn9	SFCSn8	SFCSn7	SFCSn6	SFCSn5	SFCSn4	SFCSn3	SFCSn2	SFCSn1	SFCSn0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

SFCS3nL

7	6	5	4	3	2	1	0
SFCSn7	SFCSn6	SFCSn5	SFCSn4	SFCSn3	SFCSn2	SFCSn1	SFCSn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(5) Transmit data CSI buffer register 3n (SFDB3n, SFDB3nL, SFDB3nH)

The SFDB3n register is a 16-bit buffer register that stores transmit data. It is overlayed by an 8-bit buffer register SFDB3nL on the lower 8 bits, and an 8-bit buffer register SFDB3nH on the higher 8 bits.

When transmit data is written to the SFDB3n register, the data is sequentially stored in the CSIBUFn register while the CSIBUFn pointer for writing is incremented.

When the data of this register is read, the value of the transmit data written last is read.

Access The SFDB3n register can be read/written in 16-bit units.
The SFDB3nL and SFDB3nH registers can be read/written in 8-bit and 1-bit units.

Address SFDB3n, SFDB3nL: <base> + 6_H
SFDB3nH: <base> + 7_H

Initial value 0000_H, or 00_H respectively. This register is cleared by any reset.

SFDB3n

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SFDBn15	SFDBn14	SFDBn13	SFDBn12	SFDBn11	SFDBn10	SFDBn9	SFDBn8	SFDBn7	SFDBn6	SFDBn5	SFDBn4	SFDBn3	SFDBn2	SFDBn1	SFDBn0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

SFDB3nL

7	6	5	4	3	2	1	0
SFDBn7	SFDBn6	SFDBn5	SFDBn4	SFDBn3	SFDBn2	SFDBn1	SFDBn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SFDB3nH

7	6	5	4	3	2	1	0
SFDBn15	SFDBn14	SFDBn13	SFDBn12	SFDBn11	SFDBn10	SFDBn9	SFDBn8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(6) CSIBUF status register 3n (SFA3n)

The SFA3n register indicates the status of the CSIBUFn register or the transfer status.

Access This register can be read or written in 8-bit or 1-bit units.
(However, bits 6 to 0 can only be read. They will not change even if they are written).

Address <base> + 8_H

Initial value 20_H. This register is initalized by any reset.

Caution
1. Reading the SFA3n register is prohibited when the CSIM3n.CSICAEn bit is cleared (0).
2. Because the values of the SFFULn, SFEMPn, CSOTn, and SFPn3 to SFPn0 bits may change at any time during transfer, their values during transfer may differ from the actual values. Especially, use the CSOTn bit independently (do not use this bit in relation with the other bits). To detect the end of transfer by the SFA3n register, check to see if the SFEMPn bit is 1 after the data to be transferred has been written to the CSIBUFn register.
3. If the SFA3n register is read immediately after data has been written to the SFDB3n and SFDB3nL registers, the values of the SFFULn, SFEMPn, and SFPn3 to SFPn0 bits do not change in time.
4. If the SFA3n register is read before the SFFULn bit is set to 1 and the 17th data is written, the CSIBUFn overflow interrupt (INTC3nOVF) is generated.

SFA3n

7	6	5	4	3	2	1	0
FPCLRn	SFFULn	SFEMPn	CSOTn	SFPn3	SFPn2	SFPn1	SFPn0
R/W	R/W	R	R	R	R	R	R

Table 17-6 SFA3n register contents (1/3)

Bit position	Bit name	Function
7	FPCLRn	CSIBUFn pointer clear operation 0: No operation 1: Clear all CSIBUFn pointers **Caution:** 1. This bit is always 0 when it is read. 2. If 1 is written to the FPCLRn bit during transfer, transfer is aborted. Because all the CSIBUFn pointers are cleared to 0, the remaining data in the CSIBUFn register is ignored. If 1 is written to the FPCLRn bit, be sure to read the SFA3n register to ensure that all CSIBUFn pointers have been correctly cleared to 0 (SFFULn = 0, SFEMPn = 1, SFPn[3:0] = 0000_B). Writing 0 to the FPCLRn bit is ignored and nothing happens.

Table 17-6 SFA3n register contents (2/3)

Bit position	Bit name	Function
6	SFFULn	CSIBUFn full status flag 0: CSIBUFn register has a vacancy 1: CSIBUFn is full **Caution:** 1. This bit is cleared to 0 when the CSIM3n.CSICAEn bit is cleared to 0 and the FPCLR bit is set to 1. 2. If transfer of 16 data is specified in the consecutive mode (CSIM3n.TRMDn = 1, SFN3n:SFNn[3:0] = 0000_B), the SFFULn bit is set to 1 in the same way as in the single mode (CSIM3n.TRMDn = 0) when 16 data are in the CSIBUFn register. If even one of the data has been completely transferred, the SFFULn bit is cleared to 0. However, this does not mean that the CSIBUFn register has a vacancy.
5	SFEMPn	CSIBUFn empty status flag 0: Data is in CSIBUFn register 1: CSIBUFn is empty **Caution:** 1. This flag is cleared to 0 when the CSIM3n.CSICAEn is cleared to 0 and the FPCLR bit is set to 1. 2. If the data written to the CSIBUFn register has been transferred in the consecutive mode (CSIM3n.TRMDn = 1), the SFEMP bit is set to 1 in the same way as in the single mode (CSIM3n.TRMDn = 0) even if receive data is stored in the CSIBUF register.
4	CSOTn	Transfer status flag 0: Idle status 1: Transfer or transfer start processing in progress **Caution:** 1. This flag is cleared to 0 when the CSIM3n.CSICAEn bit is cleared to 0 and the FPCLRn bit is set to 1, or when the CSIM3n.CTXEn and CSIM3n.CRXEn bits are cleared to 0. 2. This flag is set (1) from when transfer is started until there is no more transfer data in the CSIBUFn register in the single mode (CSIM3n.TRMDn = 0) or until the specified number of data has been transferred in the consecutive mode (CSIM3n.TRMDn = 1).

Table 17-6 SFA3n register contents (3/3)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3 to 0</td><td>SFPn[3:0]</td><td>CSIBUFn pointer status
<table>
<tr><th>SFPn3</th><th>SFPn2</th><th>SFPnS1</th><th>SFPn0</th><th>CSIBUFn pointer status</th></tr>
<tr><td colspan="4">0_H to F_H
(0 to 15)</td><td>• In the single mode (CSIM3n.TRMDn = 0), the “number of transfer data remaining in CSIBUFn register (CSIBUFn pointer value for writing – CSIBUFn pointer value for SIO3n loading)” can be read.
• In the consecutive mode (CSIM3n.TRMDn = 1), the “number of data completely transferred (value of CSIBUFn pointer for SIO3n loading/storing)” can be read. If the SFPn3 to SFPn0 bits are 0H, however, the number of transferred data is as follows, depending on the setting of the SFEMPn bit.
When SFEMPn bit = 0:
Number of transferred data = 0
When SFEMPn bit = 1:
Number of transferred data = 16
or status before starting transfer
(before writing transfer data)</td></tr>
</table>
Caution: These bits are cleared to 0 in synchronization with the operating clock when the FPCLRn bit = 1. However, the values of these bits are held until the CSIM3n.CSICAEn bit is cleared to 0 or the FPCLRn bit is set to 1.</td></tr>
</table>

(7) Transfer data length select register 3n (CSIL3n)

The CSIL3n register is used to select the transfer data length of CSI3n.

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

Address <base> + 9_H

Initial value 00_H. This register is cleared by any reset.

Caution The CSIL3n register may be transferring data when the CSIM3n.CTXEn or CSIM3n.CRXEn bit is 1. Before writing data to the CSIL3n register, be sure to clear the CSIM3n.CTXEn and CSIM3n.CRXEn bits to 0.

CSIL3n

7	6	5	4	3	2	1	0
CSLVn3	CSLVn2	CSLVn1	CSLVn0	CCLn3	CCLn2	CCLn1	CCLn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17-7 CSIL3n register contents

Bit position	Bit name	Function
7 to 4	CSLVnm	Chip select output (SCS3nm) level setting 0: Active level of SCSnm output is low level. 1: Active level of SCSnm output is high level. **Note:** m = 0 to 3
3 to 0	SFPn[3:0]	Transfer data length

CCLn3	CCLn2	CCLn1	CCLn0	Transfer data length
0	0	0	0	16 bits
1	0	0	0	8 bits
1	0	0	1	9 bits
1	0	1	0	10 bits
1	0	1	1	11 bits
1	1	0	0	12 bits
1	1	0	1	13 bits
1	1	1	0	14 bits
1	1	1	1	15 bits
Other than above				Setting prohibited

Caution: If a transfer data length other than 16 bits is specified (CCLn[3:0] = 0000_B), an undefined value is read to the higher excess bits of the SIRB3n and CSIBUFn registers (srefer to *"Data transfer direction specification function" on page 667*).

(8) Transfer data number specification register 3n (SFN3n)

The SFN3n register is used to set the number of transfer data of CSI3n in the consecutive mode (TRMDn bit of the CSIM3n register = 1).

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

Address <base> + C_H

Initial value 00_H. This register is cleared by any reset.

	7	6	5	4	3	2	1	0
SFN3n	0	0	0	0	SFNn3	SFNn2	SFNn1	SFNn0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17-8 SFN3n register contents

Bit position	Bit name	Function
3 to 0	SFNn[3:0]	Number of transfer data in consecutive mode.

SFNn3	SFNn2	SFNn1	SFNn0	Number of Transfer Data
0	0	0	0	16
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

Caution: Writing data exceeding the value set by the SFNn3 to SFNn0 bits (number of CSI3n transfer data) to the CSIBUFn register is prohibited (data is ignored even if written).

17.4 Dedicated Baud Rate Generator 3n (BRG3n)

The transfer clock of CSI3n can be selected from the output of a dedicated baud rate generator (BRG3n) or external clock.

The serial clock source is specified by the CSIC3n register.

In the master mode (CSIC3.CKS3n[2:0] bits other than 111_B), BRG3n is selected as the clock source.

(1) Transfer clock

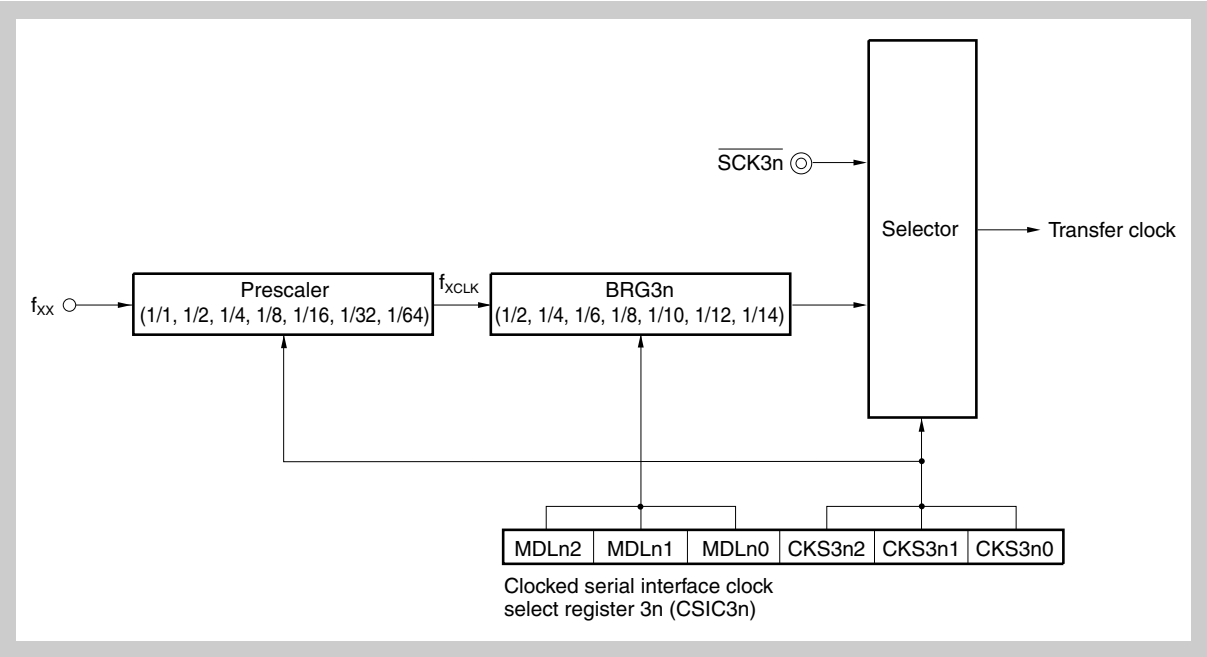


Figure 17-2 Transfer Clock of CSI3n

Remark f_{XX}: Main clock
f_{XCLK}:Basic clock selected by CSIC3n register

(2) Baud rate

The baud rate is calculated by the following expression.

$$\text{Baud rate} = \frac{f_{XX}}{N \times 2^{(k+1)}} \text{ [bps]}$$

Remarks

1. f_{XX}: Main clock
2. k: Value set by CSIC3.CKS3n[2:0] bits ($0 \leq k \leq 6$)
3. N: Value set by CSIC3.MDLn[2:0] bits ($1 \leq N \leq 7$)

Caution

1. If the CSIC3.CKS3n[2:0] bits are cleared to 000_B, setting the CSIC3.MDLn[2:0] bits to 001_B is prohibited.
2. Because the maximum transfer rate in the master mode (CSIC3.CKS3n[2:0] bits other than 111_B) is 8 Mbps, do not exceed this value.
   Example: When the main clock f_{XX} is 64 MHz, the maximum transfer rate is set when the CKS3n[2:0] bits = 000_B and the MDLn[2:0] bits = 100_B.

17.5 Operation

17.5.1 Operation modes

Table 17-9 Operation Modes

TRMDn Bit	CKS3n[2:0] Bits	CTXEn and CRXEn Bits	DIRn Bit	CSITn Bit	CSWEn Bit	CSMDn Bit
Single mode	Master mode	Transmission/ reception/ transmission and reception	MSB/LSB first	INTC3n delay mode enabled/ disabled	Transfer wait disabled	Intermediate inactive level of chip select outputs disabled
					Transfer wait enabled	Intermediate inactive level of chip select outputs enabled
	Slave mode			–	–	–
Consecutive mode	Master mode			INTC3n delay mode enabled/ disabled	Transfer wait disabled	Intermediate inactive level of chip select outputs disabled
					Transfer wait enabled	Intermediate inactive level of chip select outputs enabled
	Slave mode			–	–	

17.5.2 Function of CSI data buffer register (CSIBUFn)

By consecutively writing the transmit data to the SFCS3n register and the SFDB3n register from where it is transferred, the data can be stored in the CSIBUFn register while the CSIBUFn pointer for writing is automatically incremented (the CSIBUFn register size is 20 bits × 16) (n = 0, 1).

When the chip select outputs SCS3n0 to SCS3n3 are used, write SFCS3n register before the SFDB3n register. However, in slave mode the chip select outputs SCS3n0 to SCSS3n3 keep the inactive level and therefore writing to the SFCS3n register is not necessary.

The condition under which transfer is to be started (SFEMPn bit of the SFA3n register = 0) is satisfied when data is written to the lower 8 bits of the SFDB3n register (or SFDB3nL register). If a transfer data length of 9 bits or more is specified (CSIL3n.CCLn[3:0] bits = 0000_B, or 1001_B to 1111_B), data must be written to the SFDB3n register in 16-bit units or to the SFDB3nH and SFDB3nL registers, in that order, in 8-bit units. If the transfer data length is set to 8 bits (CCLn[3:0] bits = 1000_B), data must be written to the SFDB3nL register in 8-bit units or to the SFDB3n register in 16-bit units. If data is written to the SFDB3nL register in 16-bit units, however, the higher 8 bits of the data (of the SFDB3nH register) are ignored and not transferred.

The SFFULn bit of the SFA3n register is set to 1 when 16 data exist in the CSIBUFn register and outputs a CSIBUFn overflow interrupt (INTC3nOVF) when the SFFULn bit = 1 and when the 17th transfer data is written.

Sixteen data exist in the CSIBUFn register in the single mode (CSIM3n.TRMDn bit = 0) when “CSIBUFn pointer value for writing = CSIBUFn pointer value for SIO3n loading, and SFFULn bit = 1”. When the CSIBUFn pointer for SIO3n loading is incremented after completion of transfer, the CSIBUFn register has a vacancy of one data (in the consecutive mode (TRMDn bit = 1), the CSIBUFn register does not have a vacancy even if one data has been transferred).

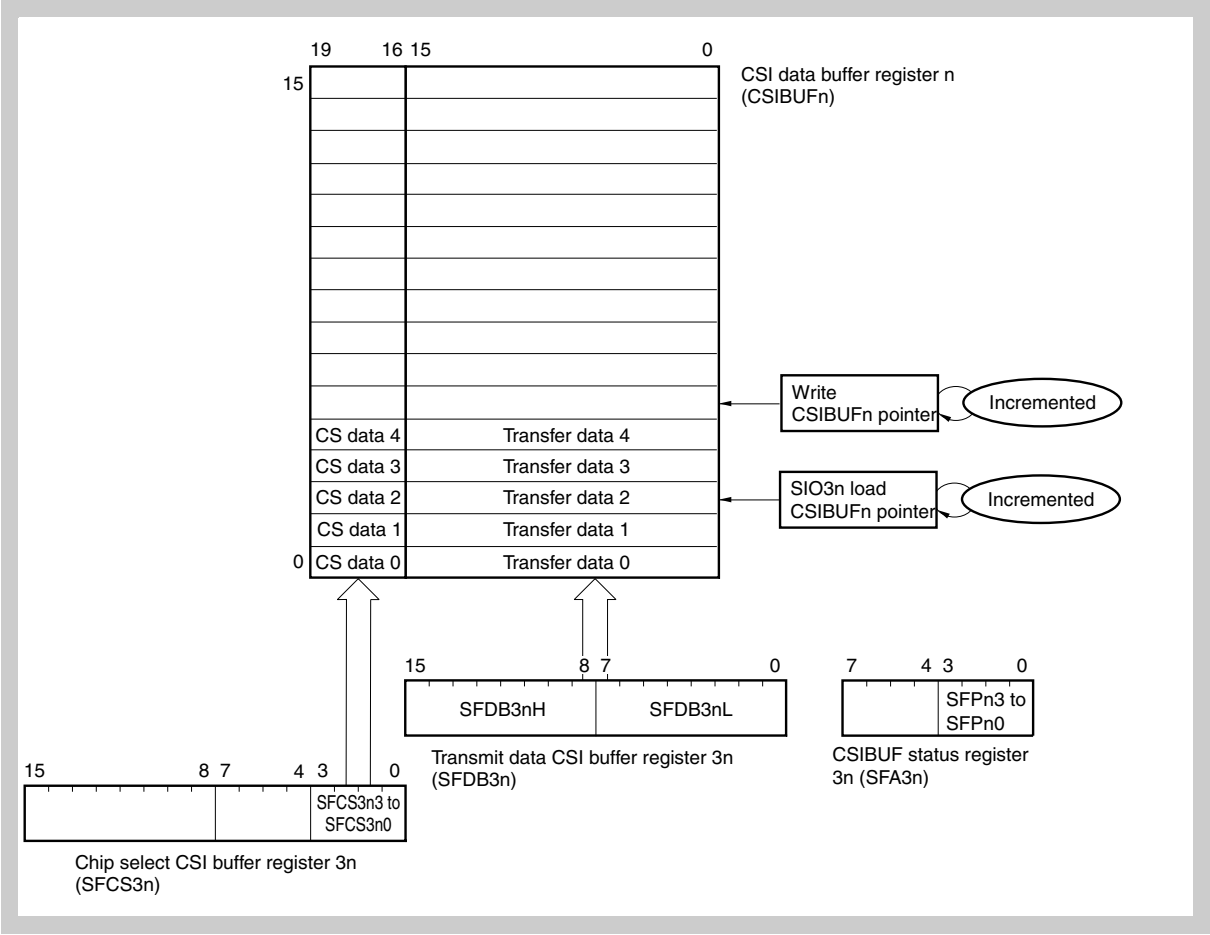


Figure 17-3 Function of CSI Data Buffer Register n (CSIBUFn)

17.5.3 Data transfer direction specification function

The data transfer direction can be changed by using the CSIM3n.DIRn bit.

(1) MSB first (CSIM3n.DIRn = 0)

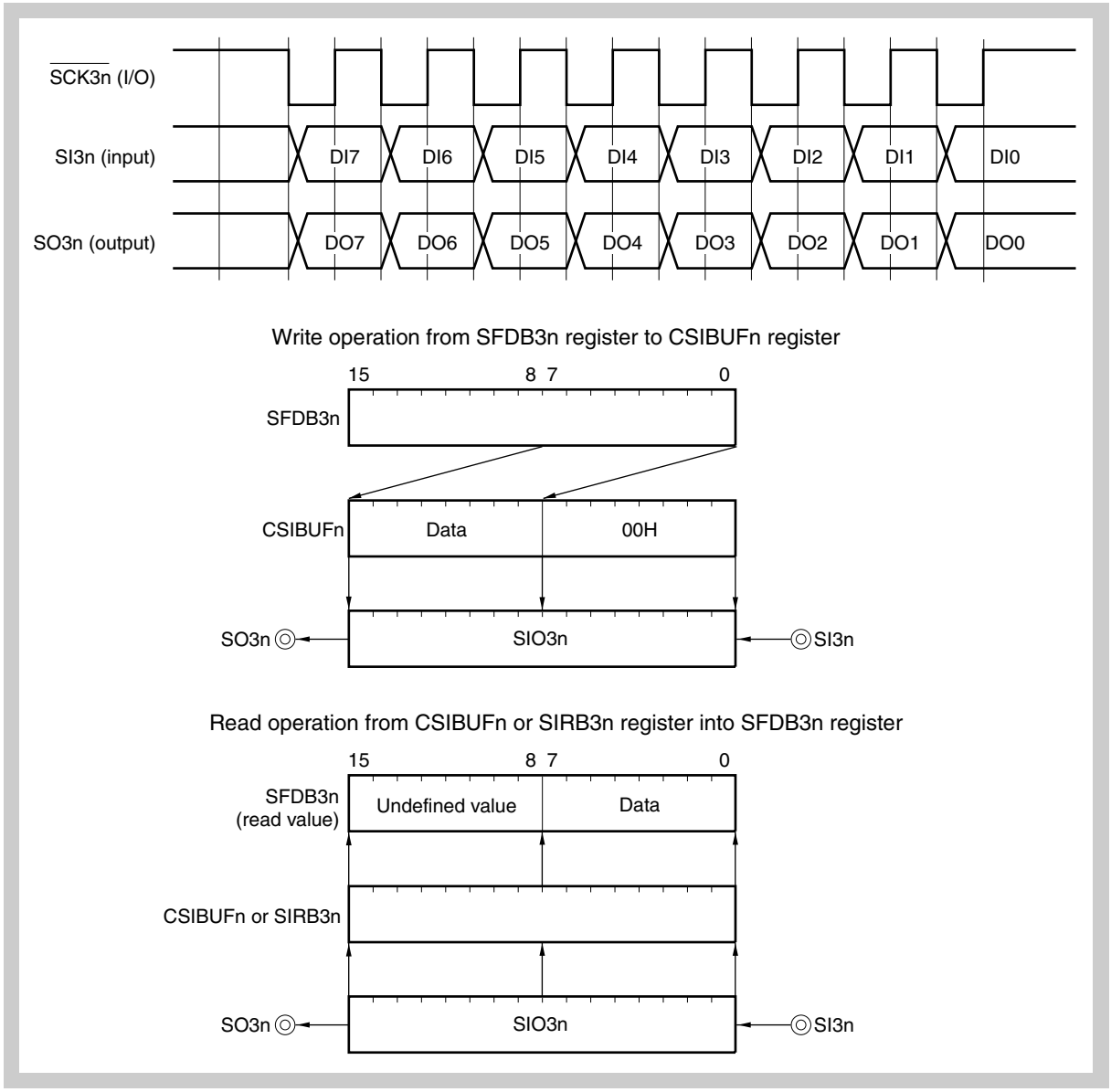


Figure 17-4 Data Transfer Direction Specification (MSB first)

(2) LSB first (CSIM3n.DIRn = 1)

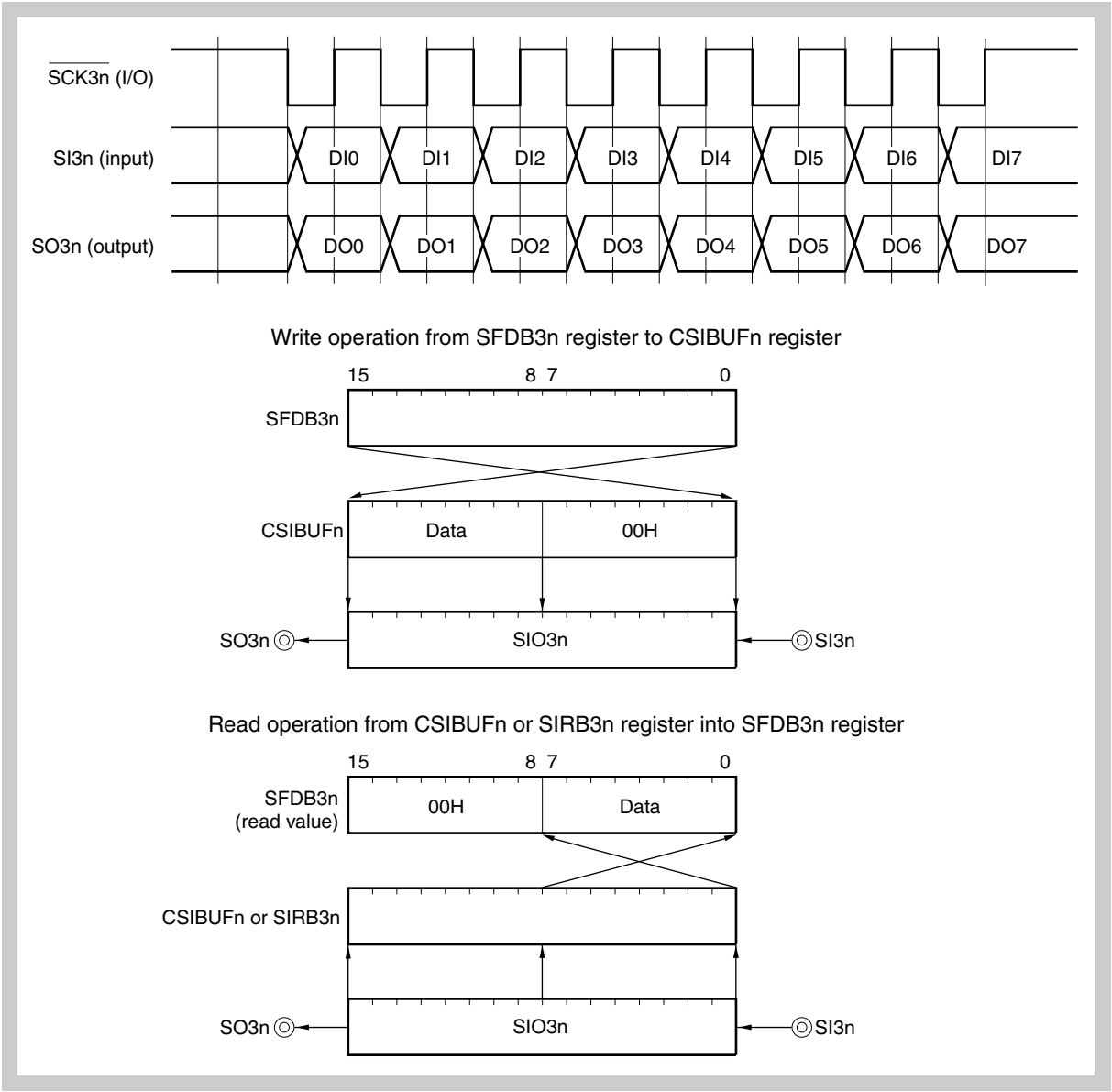


Figure 17-5 Data Transfer Direction Specification (LSB first)

17.5.4 Transfer data length changing function

The transfer data length can be set from 8 to 16 bits in 1-bit units by using the CSIL3n.CCLn[3:0] bits.

Example: Transfer Data Length: 16 Bits (CSIL3n.CCLn[3:0] = 0000_B),
Transfer Direction: MSB First (CSIM3n.DIRn = 0)

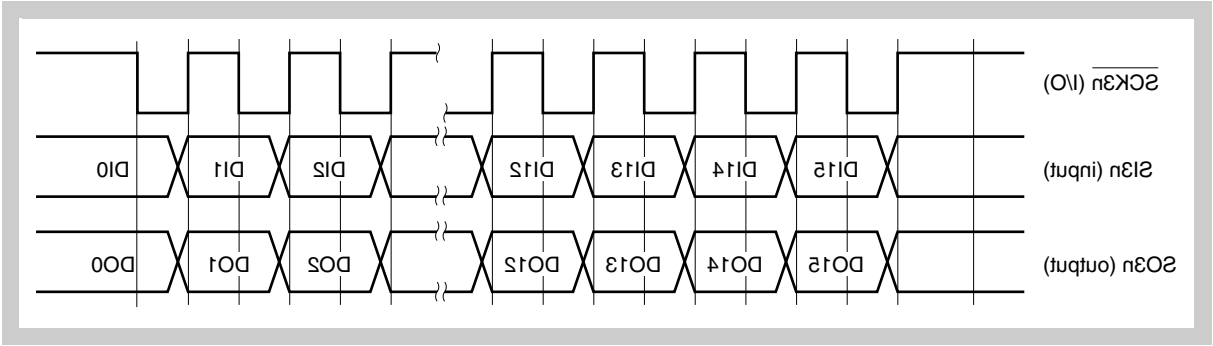


Figure 17-6 Transfer Data Length Changing Function

17.5.5 Serial clock and data phase selection function

The serial clock and data phase can be changed by using the CKPn and DAPn bits of the CSIC3n register.

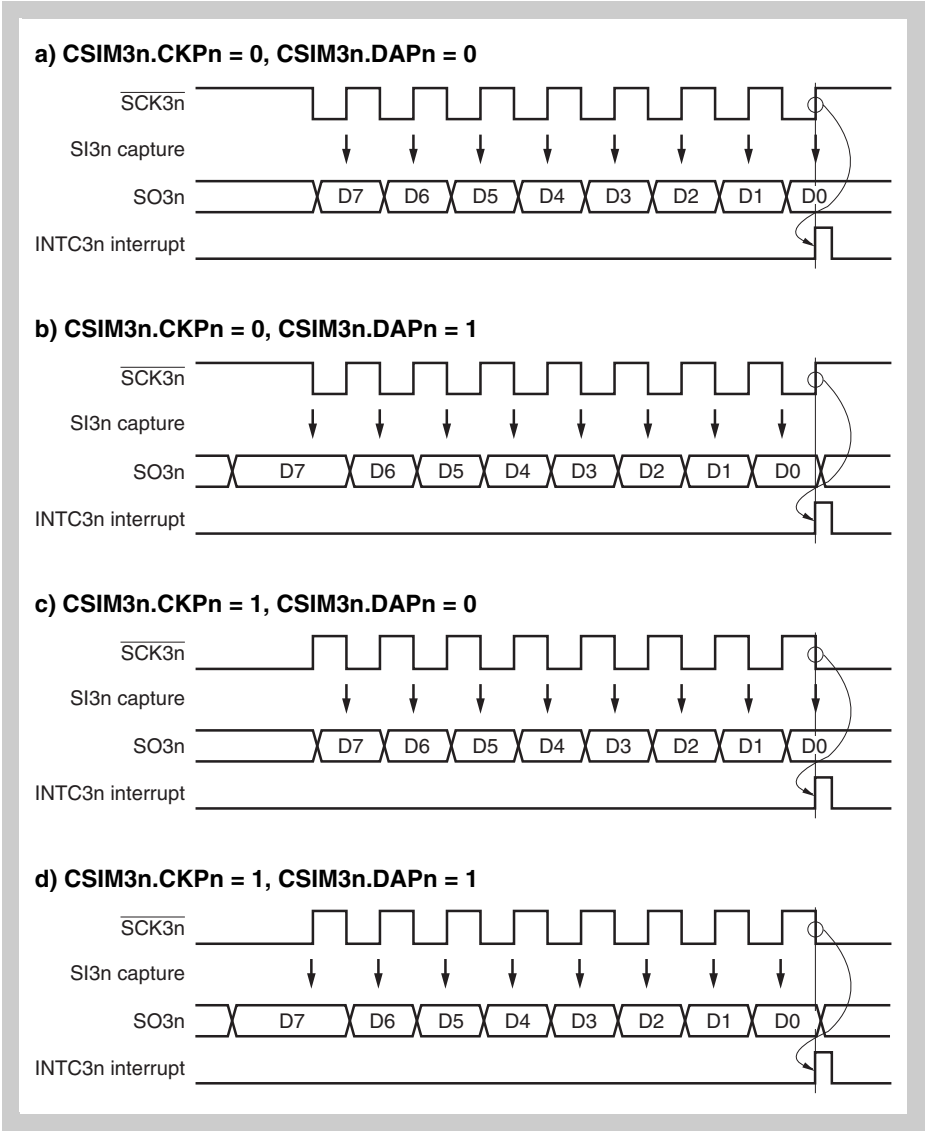


Figure 17-7 Clock Timing

17.5.6 Master mode

The master mode is set and data is transferred with the transfer clock output to the $\overline{SCK3n}$ pin when the CSIC3n.CKS3n[2:0] bits are set to a value other than 111_B ($\overline{SCK3n}$ pin input is invalid).

The default output level of the $\overline{SCK3n}$ pin is high when the CSIC3n.CKPn bit is 0, and low when the CSIC3n.CKPn bit is 1.

In master mode the chip select outputs (SCS3n0 to SCS3n3) are effective.

Example CSIC3n.CKPn = 0 and CSIC3n.DAPn = 0,
Active Level of CS Outputs: Low Level (CSIL3n.CSLVn[3:0] = 0000_B)
Transfer Data Length: 8 Bits (CSIL3n.CCLn[3:0] = 1000_B)

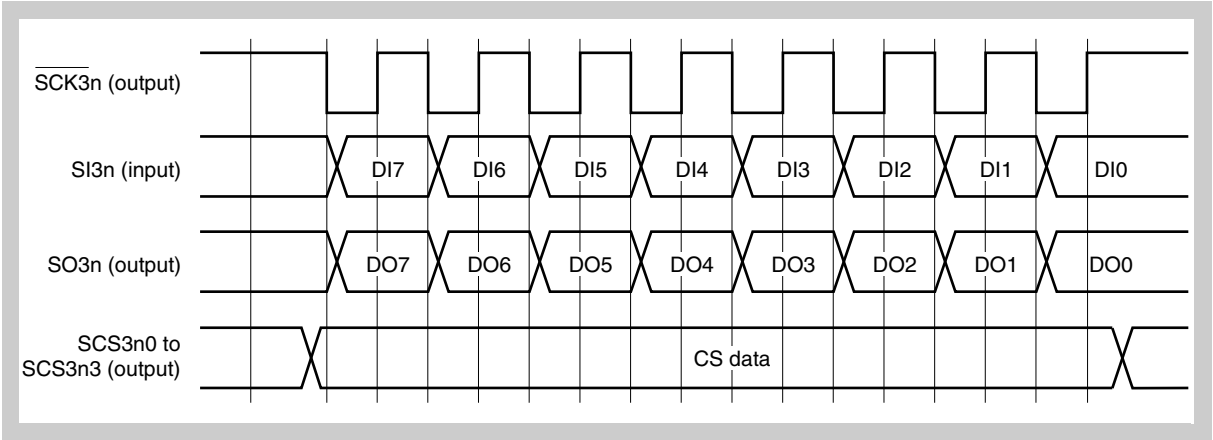


Figure 17-8 CSI3n Timing in Master Mode

17.5.7 Slave mode

The slave mode is set when the CSIC3n.CKS3n[2:0] bits are set to 111_B, and data is transferred with the transfer clock input to the $\overline{SCK3n}$ pin (in the slave mode, it is recommended to set the CSIC3n.MDLn[2:0] bits = 000_B and to set the BRGn into stop mode).

The chip select outputs (SCS3n0 to SCS3n3) are ineffective in slave mode, the output levels are fixed to inactive level (chip select outputs are effective in master mode only).

Example CSIC3n.CKPn = 0 and CSIC3n.DAPn = 0,
Active Level of CS Outputs: Low Level (CSIL3n.CSLVn[3:0] = 0000_B)
Transfer Data Length: 8 Bits (CSIL3n.CCLn[3:0] = 1000_B)

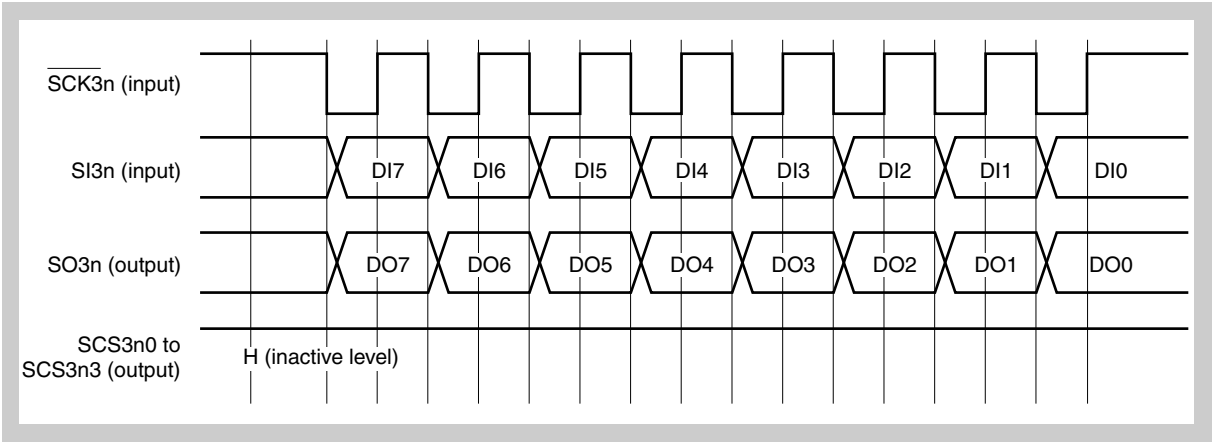


Figure 17-9 CSI3n Timing in Slave Mode

The conditions under which data can be transferred in the slave mode are listed in the table below.

Table 17-10 Conditions Under Which Data Can Be Transferred in Slave Mode

Transfer Mode		CTXEn Bit	CRXEn Bit	CSIBUFn Register	SIRB3n Register and SIO3n Register
Single mode	Transmission mode	1	0	Data is in CSIBUFn register (SFEMPn bit = 0).	–
	Reception mode	0	1	Dummy data is in CSIBUFn register (SFEMPn bit = 0).	SIRB3n register or SIO3n register is empty.
	Transmission/ reception mode	1	1	Data is in CSIBUFn register (SFEMPn bit = 0).	
Consecutive mode	Transmission mode	1	0	Data is in CSIBUFn register (SFEMPn bit = 0).	–
	Reception mode	0	1	Dummy data is in CSIBUFn register (SFEMPn bit = 0).	–
	Transmission/ reception mode	1	1	Data is in CSIBUFn register (SFEMPn bit = 0).	–

Remark CTXEn bit: Bit 6 of CSIM3n register
CRXEn bit: Bit 5 of CSIM3n register
SFEMPn bit: Bit 5 of SFA3n register

17.5.8 Transfer clock selection function

In the master mode (CSIC3n.CKS3n[2:0] bits other than 111_B), the bit transfer rate can be selected by setting the CSIC3n.CKS3n[2:0] and CSIC3n.MDLn[2:0] (ref. to **17.3 (2) Clocked serial interface clock select register 3n (CSIC30, CSIC31)**).

17.5.9 Single mode

The single mode is set when the CSIM3n.TRMDn bit is 0 .

In this mode, transfer is started when the CTXEn bit or CRXEn bit is set to 1 and when data is in the CSIBUFn register (SFA3n.SFEMPn bit = 0).

If no data is in the CSIBUFn register (SFA3n.SFEMPn bit = 1), transfer is kept waiting until a given start condition is satisfied.

When data is written to the CSIBUFn register while the CTXEn or CRXEn bit is 1, the SFA3n.CSOTn bit (transfer status flag) is set to 1, and the chip select data (CS data) corresponding to SIO3n load CSIBUFn pointer is transferred to the chip select output buffer. However, in slave mode (CSIC3n.CKS3n[2:0] = 111_B) the chip select outputs (SCS3n0 to SCS3n3) keep always the inactive level.

If transfer is not in the wait status, the transfer data indicated by the SIO3n load CSIBUFn pointer is loaded from the CSIBUFn register to the SIO3n register, and transfer processing is started.

If the SIRB3n register is empty when one data has been transferred in the reception mode or transmission/reception mode, the received data is stored from the SIO3n register to the SIRB3n register, the transmission/reception completion interrupt (INTC3n) is output, and the SIO3n load CSIBUFn pointer is incremented. If the SIRB3n register is not empty, the next transfer processing is started. However, storing the receive data in the SIRB3n register, outputting the INTC3n interrupt, and incrementing the SIO3n load CSIBUFn pointer are held pending, until the previously received data is read from the SIRB3n register and the SIRB3n register becomes empty.

In the transmission mode, the INTC3n interrupt is output and the SIO3n load pointer is incremented when transfer processing of one data has been completed (the SIRB3n register is always empty because no data is stored from the SIO3n register to the SIRB3n register).

In all modes (transmission, reception, and transmission/reception modes), if the CSIBUFn register is empty (write CSIBUFn pointer value = SIO3n load CSIBUFn pointer value) when transfer processing of one data has been completed, the CSOTn bit is cleared to 0. The value of the “number of remaining data in the CSIBUFn register (write CSIBUFn pointer – SIO3n load pointer)” can always be read from the SFA3n.SFPn[3:0] bits.

Caution When writing data to the SFDB3n register, be sure to confirm that the SFFULn bit of the SFA3n register is 0. Even if data is written to this register when SFFULn bit is 1, the CSIBUFn overflow interrupt (INTC3nOVF) is output, and the written data is ignored.

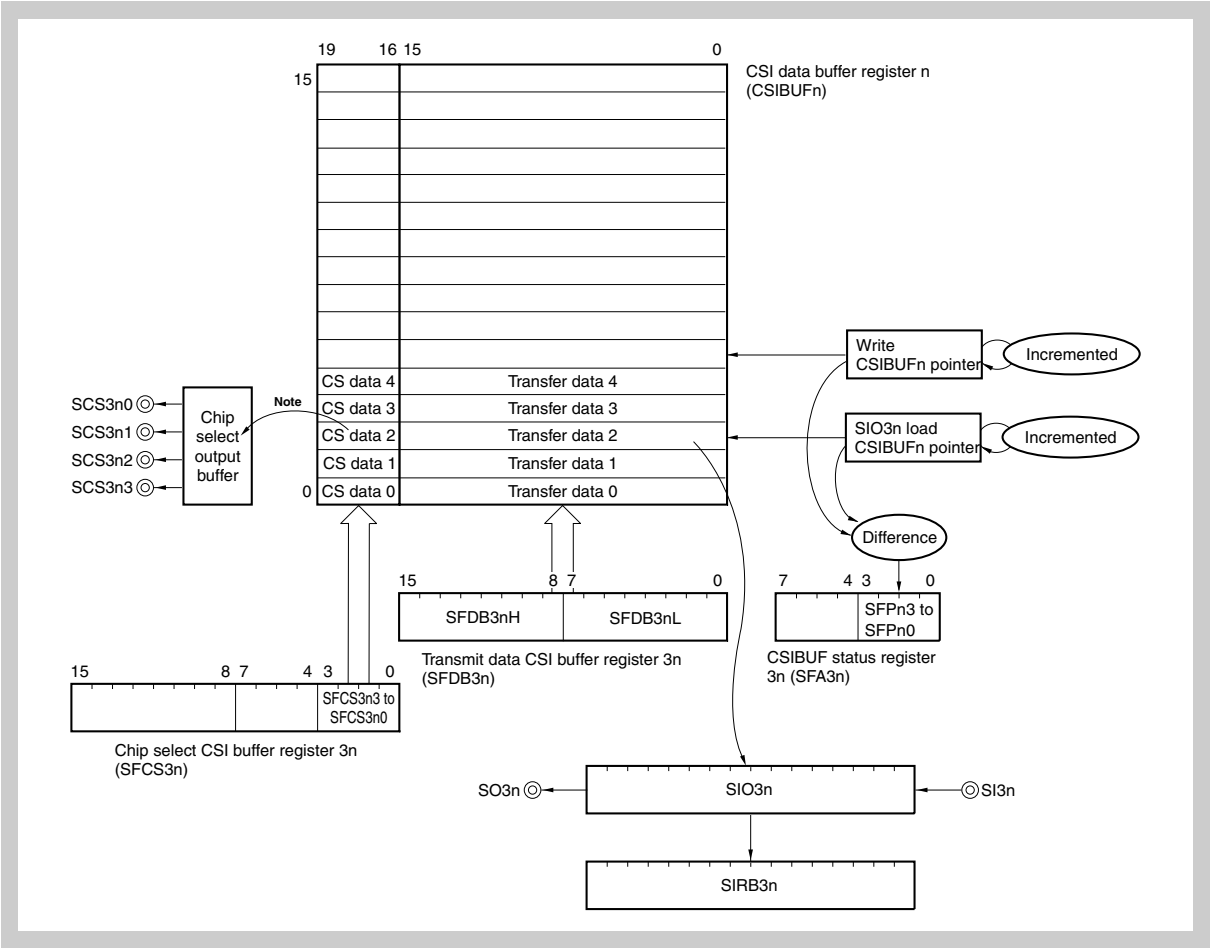


Figure 17-10 CSI3n in Single Mode Operation

Note Transfer of CS data will be performed in master mode only.

17.5.10 Consecutive mode

The consecutive mode is set when the CSIM3n.TRMDn bit is 1.

In this mode, transfer is started when the CTXEn bit or CRXEn bit is 1 and when data is in the CSIBUFn register (SFA3n.SFEMPn bit = 0). At this time, set the number of transfer data in advance by using the SFN3n.SFNn[3:0] bits. Seventeen or more transfer data cannot be set. If 17 or more transfer data are written to the CSIBUFn register, the excess data are ignored and not transferred. Do not write data exceeding the number of transfer data specified by the SFN3n.SFNn[3:0] bits to the CSIBUFn register.

If no data is in the CSIBUFn register (SFA3n.SFEMPn bit = 1), transfer is kept waiting until a given start condition is satisfied.

If data is written to the CSIBUFn register when the CTXEn or CRXEn bit is 1, the SFA3n.CSOTn bit (transfer status flag) is set to 1 and the chip select data (CS data) according to the SIO3n load/store CSIBUFn pointer is transferred to the chip select output buffer. However, in slave mode (CSIC3n.CKS3n[2:0] bits = 111_B) the chip select outputs (SCS3n0 to SCS3n3) keep always the inactive level.

If transfer is not in the wait status, the transfer data indicated by the SIO3n load/store CSIBUFn pointer is loaded from the CSIBUFn register to SIO3n register. Then transfer processing is started.

When transfer processing of one data is completed in the reception mode or transmission/reception mode, the received data is overwritten from the SIO3n register to the transfer data in the CSIBUFn register indicated by the SIO3n load/store CSIBUFn pointer, and then the pointer is incremented. By consecutively reading the transfer data from the SIRB3n register after all data in the CSIBUFn register have been transferred (when the INTC3n interrupt has occurred), the receive data can be sequentially read while the read CSIBUFn pointer is incremented.

In the transmission mode, the SIO3n load/store CSIBUFn pointer is incremented when transfer processing of one data has been completed.

In all modes (transmission, reception, and transmission/reception modes), when data has been transferred by the value set by the SFN3n.SFNn[3:0] bits, the CSOTn bit is cleared to 0 and the transmission/reception completion interrupt (INTC3n) is output.

To transfer the next data, be sure to write 1 to the SFA3n.FPCLRn bit and clear all the CSIBUFn pointers to 0.

The “number of transferred data (SIO3n load/store CSIBUFn pointer value)” can always be read from the SFA3n.SFPn[3:0] bits of the register.

Caution The SFA3n register is in the same status when transfer data is written (before start of transfer) after the CSIBUFn pointer is cleared (FPCLRn bit = 1) and when 16 data have been transferred (SFFULn bit = 0, SFEMPn bit = 1, SFPn[3:0] bits = 0000_B).

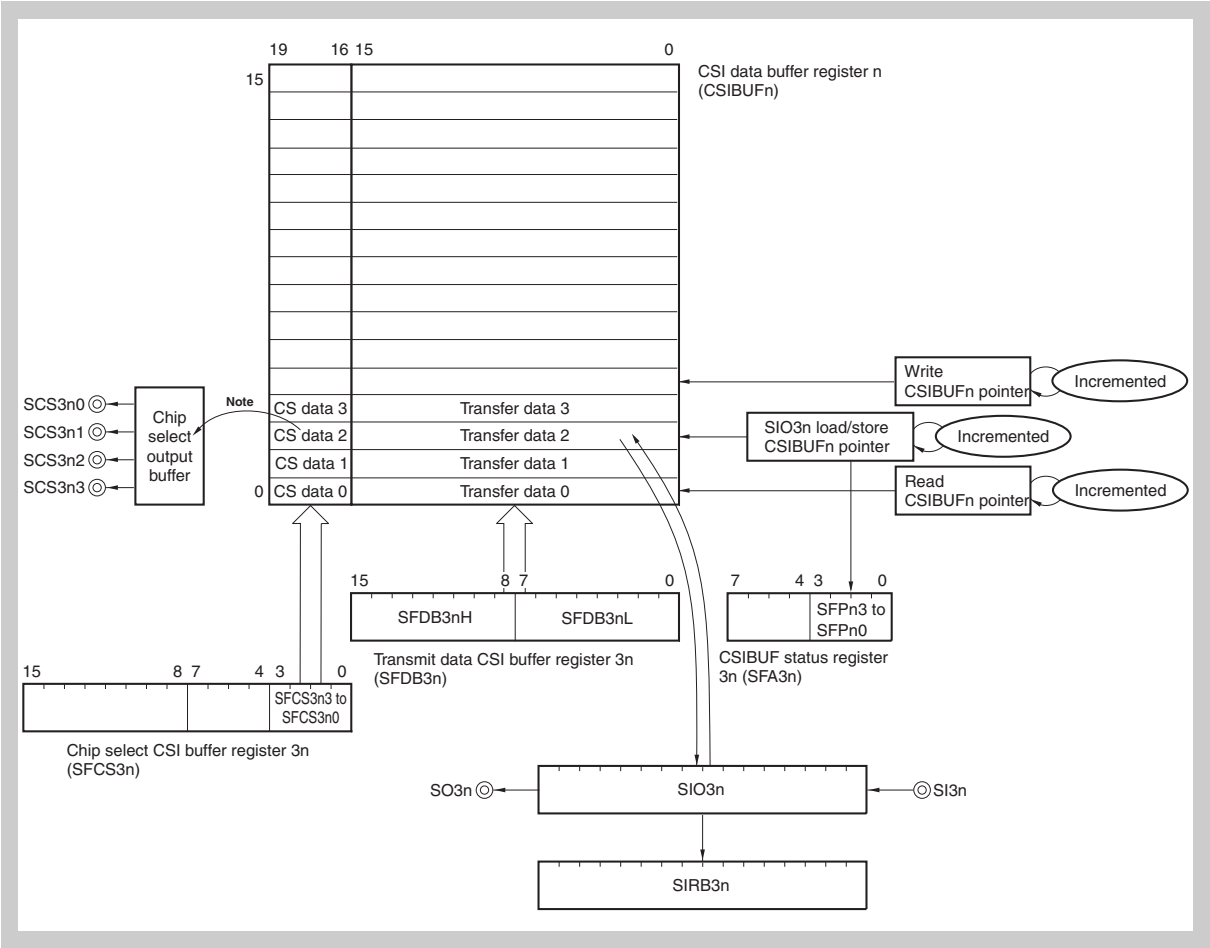


Figure 17-11 CSI3n in Consecutive Mode Operation

Note Transfer of CS data will be performed in master mode only.

17.5.11 Transmission mode

The transmission mode is set when the CSIM3n.CTXEn bit is set to 1 and the CSIM3n.CRXEn bit is cleared to 0. In this mode, transmission is started by a trigger that writes transmit data to the SFDB3n register or sets the CTXEn bit to 1 when transmit data is in the SFDB3n register. Even in the single mode (CSIM3n.TRMDn bit = 0), whether the SIRB3n or SIO3n register is empty has nothing to do with starting transmission. The value input to the SI3n pin during transmission is latched in the shift register (SIO3n) but is not transferred to the SIRB3n and CSIBUFn registers at the end of transmission.

The transmission/reception completion interrupt (INTC3n) occurs immediately after data is sent out from the SIO3n register.

17.5.12 Reception mode

The reception mode is set when the CSIM3n.CTXEn bit is cleared to 0 and CSIM3n.CRXEn bit is set to 1. In this mode, reception is started by using the processing of writing dummy data to the SFDB3n register as a trigger. In the single mode (CSIM3n.TRMDn bit = 0), however, the condition of starting reception includes that the SIRB3n or SIO3n register is empty. (If reception to the SIO3n register is completed when the previously received data is held in the SIRB3n register without being read, the previously received data is read from the SIRB3n register and the wait status continues until the SIRB3n register becomes empty.)

The SO3n pin outputs a low level.

The transmission/reception completion interrupt (INTC3n) occurs immediately after receive data is transferred from the SIO3n register to the SIRB3n register.

17.5.13 Transmission/reception mode

The transmission/reception mode is set when both the CSIM3n.CTXEn and CSIM3n.CRXEn bits are set to 1. In this mode, transmission/reception is started by using the processing to write transmit data to the SFDB3n register as a trigger (n = 0, 1). In the single mode (CSIM3n.TRMDn bit = 0), however, the condition of starting transmission/reception includes that the SIRB3n or SIO3n register is empty. (If reception to the SIO3n register is completed when the previously received data is held in the SIRB3n register without being read, the previously received data is read from the SIRB3n register and the wait status continues until the SIRB3n register becomes empty.)

17.5.14 Delay control of transmission/reception completion interrupt (INTC3n)

In the master mode (CSIC3n.CKS3n[2:0] bits other than 111_B), occurrence of the transmission/reception completion interrupt (INTC3n) can be delayed by half a clock (1/2 serial clock) when the CSIM3n.CSITn bit is set to 1. However, the CSITn bit is valid only in master mode. In slave mode (CSIC3n.CKS3n[2:0] bits = 111_B), setting the CSITn bit to 1 is prohibited (even if set, the INTC3n interrupt is not affected).

Caution If the CSIM3n.CSITn bit is set to 1 in the consecutive mode (CSIM3n.TRMDn bit = 1), the INTC3n interrupt is not output at the end of data other than the last data set by the SFN3n.SFNn[3:0] bits, but a delay of half a clock can be inserted between each data transfer.

Example CSIM3n.CSITn = 1,
CSIM3n.CSWEn = 0,
CSIC3n.CKPn = 0 and CSIC3n.DAPn = 0,
Transfer Data Length: 8 Bits (CSIL3n.CCLn[3:0] = 1000_B)

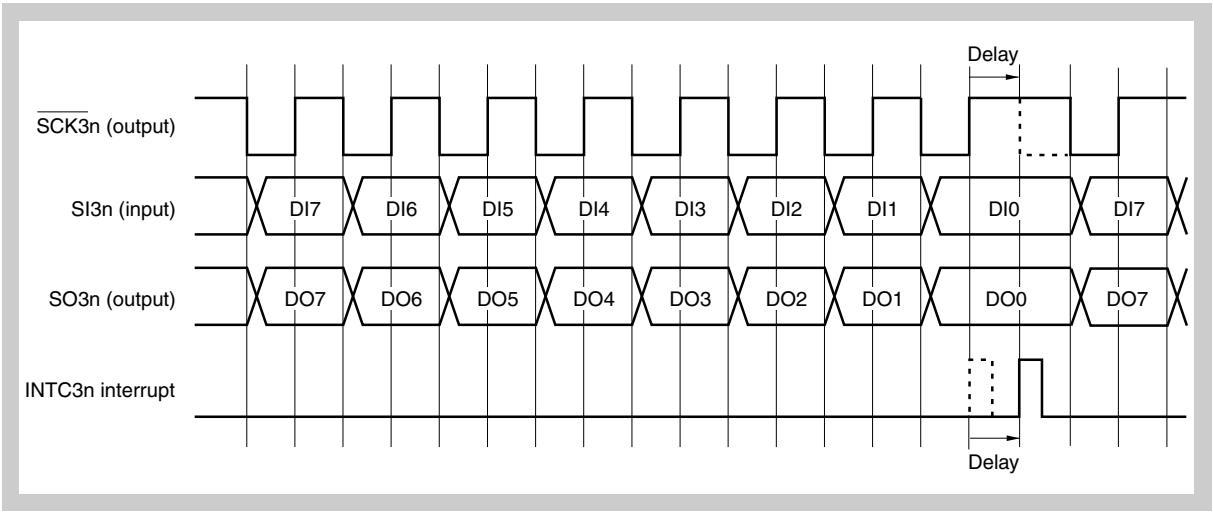


Figure 17-12 Delay Control of Transmission/Reception Completion Interrupt (INTC3n)

17.5.15 Transfer wait function

In the master mode (CSIC3n.CKS3n[2:0] bits other than 111_B), starting transfer can be delayed by one clock, when the CSIM3n.CSWEn bit is set to 1. The CSWEn bit is valid only in master mode. In slave mode (CSIC3n.CKS3n[2:0] bits = 111_B), setting the CSWEn bit to 1 is prohibited (even if set, transfer wait is not inserted).

During transfer wait (CSWE bit = 1) the chip select outputs (SCS3n0 to SCS3n3) can be configured for an intermediate inactive level output of half a clock period by setting the CSIM3n.CSMDn bit to 1.

Note Following figures prosume that CSIC3n.CKPn bit = 0 and CSIC3nDAPn bit = 0, and thet transfer data length is set to 8 bits (CSIL3n.CCLn[3:0] = 1000_B)

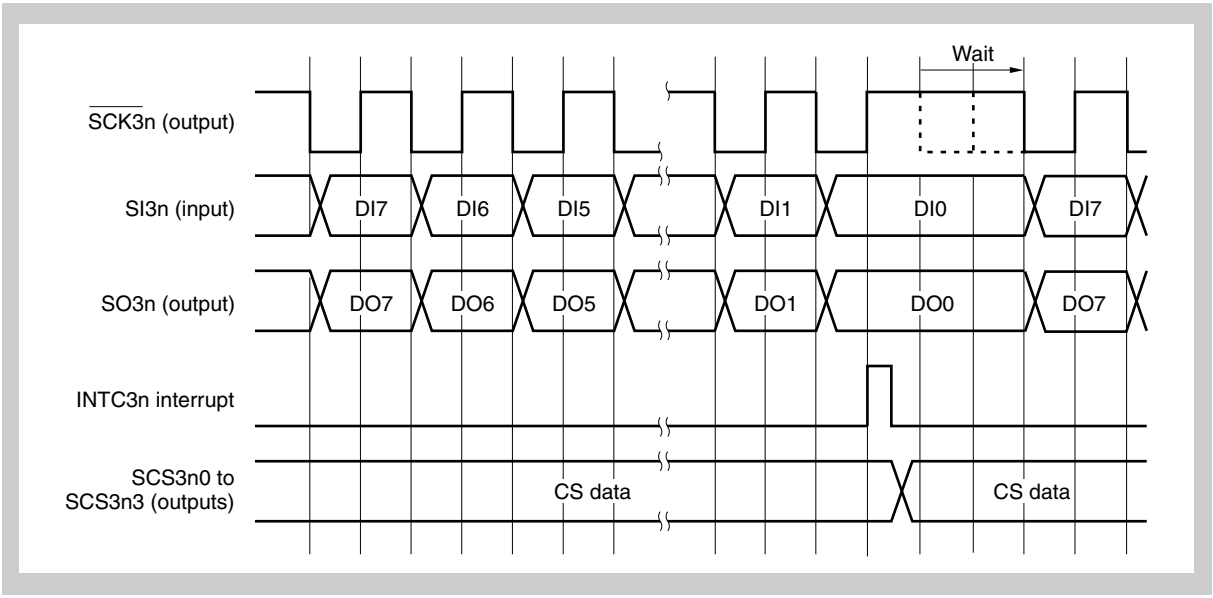


Figure 17-13 Transfer Wait Function (CSIM3n.CSWEn = 1) when INTC3 Delay disabled (CSIM3n.CSITn = 0) and Intermediate Inactive Chip Select Level disabled (CSIM3n.CSMDn = 0)

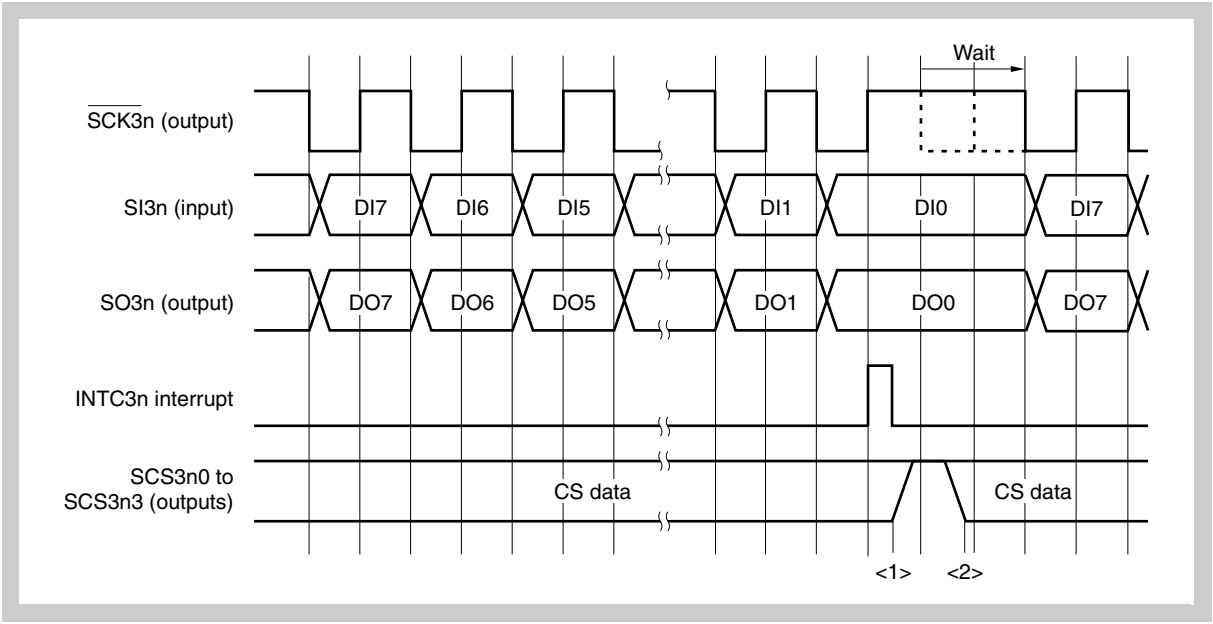


Figure 17-14 Transfer Wait Function (CSIM3n.CSWEn = 1) when INTC3 Delay disabled (CSIM3n.CSITn = 0) and Intermediate Inactive Chip Select Level enabled (CSIM3n.CSMDn = 1)

Remark When the CSIBUFn register is empty at the time of <1>, the chip select pins output an inactive level and maintain it.
When the CSIBUFn register is not empty at the time of <1>, the chip select pins output an inactive level up to the time of <2>, and output subsequently the succeeding chip select data
Moreover, in single mode (TRMDn bit of the CSIM3n register = 0) the chip select pins output an inactive level from the time <1> and held it pending until the previously receive data is read from the SIRB3n register and the SIRB3n register becomes empty.

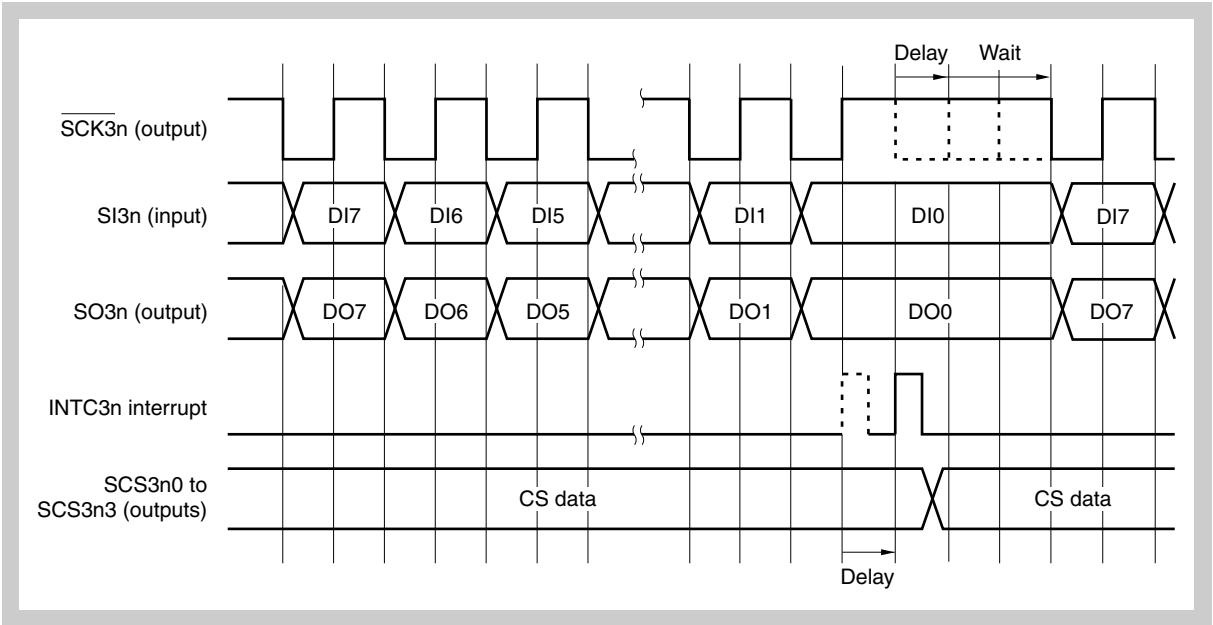


Figure 17-15 Transfer Wait Function (CSIM3n.CSWEn = 1) when INTC3 Delay enabled (CSIM3n.CSITn = 1) and Intermediate Inactive Chip Select Level disabled (CSIM3n.CSMDn = 0)

17.5.16 Output pins

(1) $\overline{\text{SCK3n}}$ pin

The $\overline{\text{SCK3n}}$ pin outputs a high level when both the CSIM3n.CTXEn and CSIM3n.CRXEn bits are 0.

In the master mode (CSIC3n.CKS3n[2:0] bits = other than 111_B), this pin outputs the default level when the SFA3n.FPCLRn bit is set to 1.

In the slave mode (CSIC3n.CKS3n[2:0] bits = 111_B), the default output level of the $\overline{\text{SCK3n}}$ pin is fixed to the high level.

Table 17-11 Default Output Level of $\overline{\text{SCK3n}}$ Pin

CSICn.CKPn	CSIC3n.CKS3n[2:0]	Default Output Level of $\overline{\text{SCK3n}}$ Pin
0	111_B (slave mode)	High level [a]
	Other than 111_B (master mode)	High level
1	111_B (slave mode)	High level
	Other than 111_B (master mode)	Low level

a) Default value after reset, or value when CSIM3n.CSICAEn bit is cleared to 0.

Remark The output of the $\overline{\text{SCK3n}}$ pin changes if the CKPn bit is rewritten in the master mode.

(2) SO3n pin

The SO3n pin outputs a low level when both the CSIM3n.CTXEn and CSIM3n.CRXEn bits are 0.

This pin outputs a low level when the SFA3n.FPCLRn bit is set to 1 (the previous value is retained only in the slave mode (CSIC3n.CKS3n[2:0] bits = 111_B) and when the CSIC3n.DAPn bit is 0).

Table 17-12 Default Output Level of SO3n Pin

Default Output Level of SO3n Pin
Low level [a]

a) Default value after reset, or value when CSIM3n.CSICAEn bit is cleared to 0.

(3) SCS3n0 to SCS3n3 pins

The SCS3n0 to SCS3n3 pins output the default level when both the CSIM3n.CTXEn and CSIM3n.CRXEn bits are 0, or when the CSIM3n.CSICAEn bit is cleared to 0.

These pins output the default level when the SFA3n.FPCLRn bit is set to 1. In slave mode these pins output always the default level (inactive level).

Table 17-13 Default Output Level of SCS3n0 to SCS3n3 Pins

CSIL3n.CSLVnm	Default Output Level of SCS3nm Pin[a]
0	High level [b]
1	Low level

a) m = 0 to 3
b) Default value after reset.

17.5.17 CSIBUFn overflow interrupt signal (INTC3nOVF)

The INTC3nOVF interrupt is output when 16 data exist in the CSIBUFn register and when the 17th data is written (to the SFDB3n or SFDB3nL register). The 17th data is not written but ignored.

In the single mode (CSIM3n.TRMDn bit = 0), 16 data exist in the CSIBUFn register when “write CSIBUFn pointer value = SIO3n load CSIBUFn pointer value” and SFA3n.SFFULn bit = 1. When transfer is completed and the SIO3n load CSIBUFn pointer is incremented, the CSIBUFn register has one vacancy (the CSIBUFn register has no vacancy even when transfer of one data has been completed in the consecutive mode (CSIM3n.TRMDn bit = 1)).

17.6 Operating Procedures

17.6.1 Single mode (master mode, transmission mode)

Example Conditions

MSB First (CSIM3n.DIRn bit = 0)
CSIC3n.CKPn bit = 0, CSIC3n.DAPn bit = 0
Transfer Data Length: 8 Bits (CSIL3n.CCLn[3:0] bits = 1000_B)
INTC3n Interrupt Not Delayed (CSIM3n.CSIT bit = 0),
Transfer Wait: Disabled (CSIM3n.CSWE bit = 0),
Chip Select Active Level: L-Level (CSIL3n.CSLVn[3:0] bits = 0000_B)

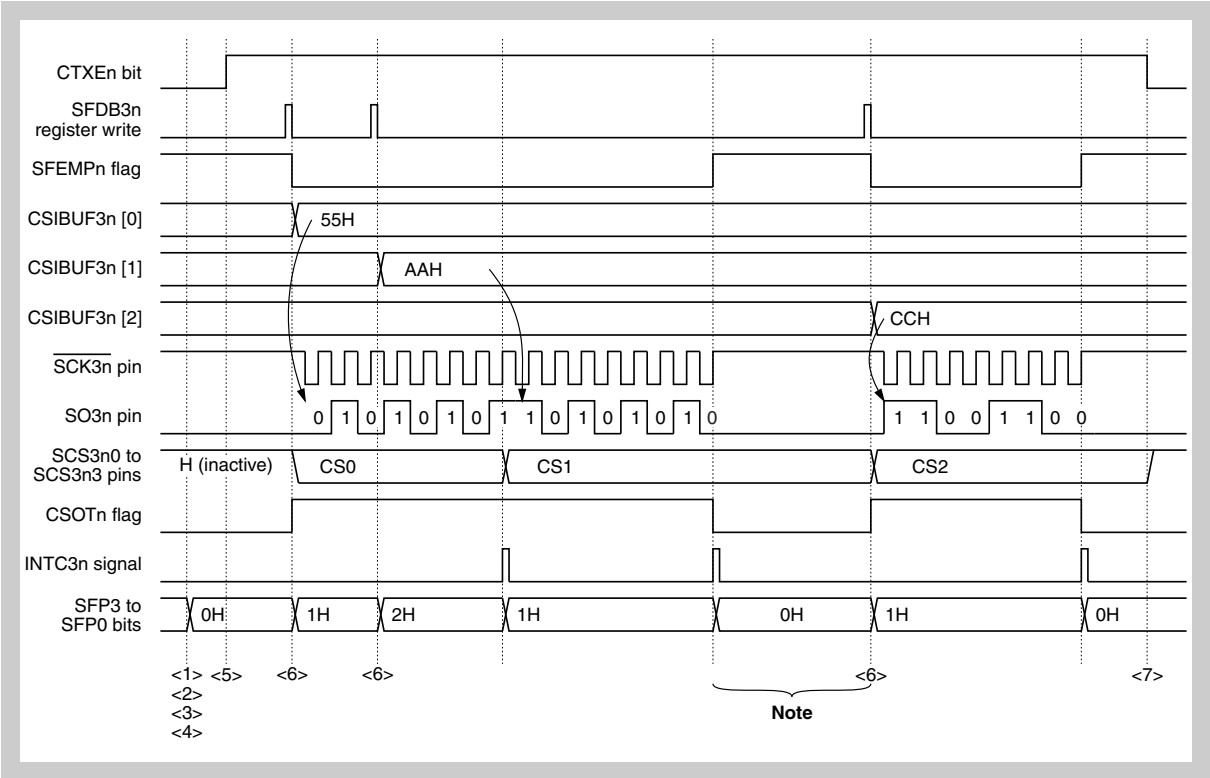


Figure 17-16 Single Mode (Master Mode, Transmission Mode)

Note During this period a reception from the slave is put on hold until at least one transmit data has been loaded to the CSIBUFn register by writing the SFDB3n register (SFA3n.SFEMPn flag = 0) in order to start the transfer.

<1> When the CSIM3n.CSICAEn bit is set to 1, operating clock supply is enabled.
<2> Specify the transfer mode by setting the CSIC3n and CSIL3n registers.
<3> Write 1 to the SFA3n.FPCLRn bit to clear all the CSIBUFn pointers to 0.
<4> Confirm that the SFA3n..SFFULn bit = 0, SFA3n.SFEMPn bit = 1, and SFA3n.SFPn[3:0] bits = 0000_B.
<5> Specify the transfer mode by using the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits of the CSIM3n register and, at the same time, enable transmission by setting the CTXEn bit to 1.
<6> Confirm that the SFA3n.SFFULn bit is 0, and then write first CS data to the SFCS3n register and subsequently write transfer data to the SFDB3n register.
If it is clearly known that the SFA3n.SFFULn bit is 0, because transfer

data is written to that bit by the interrupt servicing routine of INTC3n, it is not always necessary to confirm that the SFA3n.SFFULn bit is 0.

<7> Confirm that the INTC3n interrupt has occurred and the SFA3n.SFEMPn bit is 1, and disable transmission by clearing the CSIM3n.CTXEn bit to 0 (end of transmission).

Remark To execute a further transfer, repeat <6> before <7>.

17.6.2 Single mode (master mode, reception mode)

Example Conditions
MSB First (CSIM3n.DIRn bit = 0)
CSIC3n.CKPn bit = 1, CSIC3n.DAPn bit = 1
Transfer Data Length: 8 Bits (CSIL3n.CCLn[3:0] bits = 1000_B)
INTC3n Interrupt Not Delayed (CSIM3n.CSIT bit = 0),
Transfer Wait: Disabled (CSIM3n.CSWE bit = 0),
Chip Select Active Level: L-Level (CSIL3n.CSLVn[3:0] bits = 0000_B)

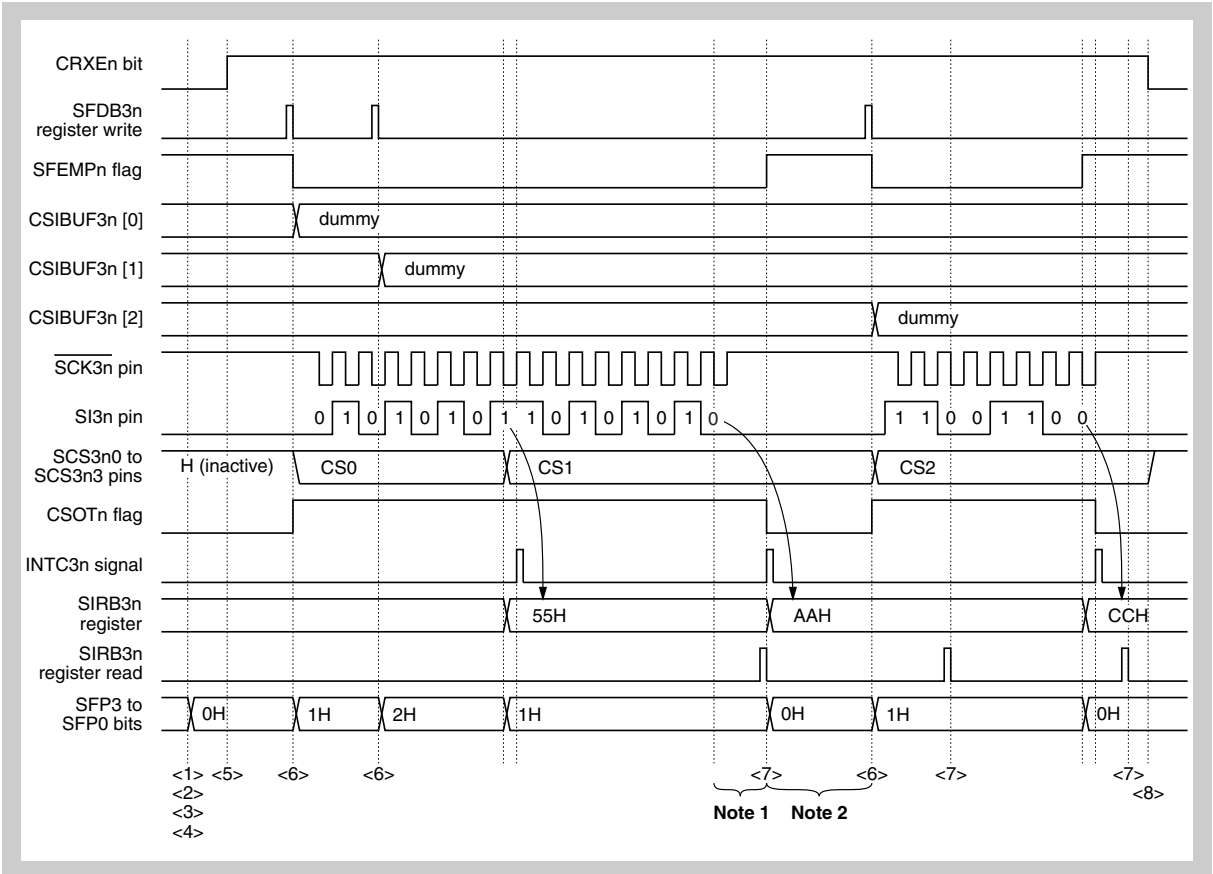


Figure 17-17 Single Mode (Master Mode, Reception Mode)

Note
1. While the SIRB3n register is full a new transfer start of reception from the slave is put on hold until the SIRB3n register is read.
2. During this period a reception from the slave is put on hold until at least one dummy transmit data has been loaded to the CSIBUFn register by writing the SFDB3n register (SFA3n.SFEMPn flag = 0) in order to start the transfer.

<1> When the CSIM3n.CSICAEn bit is set to 1, operating clock supply is enabled.
<2> Specify the transfer mode by setting the CSIC3n and CSIL3n registers.
<3> Write 1 to the SFA3n.FPCLRn bit to clear all the CSIBUFn pointers to 0.
<4> Confirm that the SFA3n.SFFULn bit = 0, SFA3n.SFEMPn bit = 1, and SFA3n.SFPn[3:0] bits = 0000_B.
<5> Specify the transfer mode by using the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits of the CSIM3n register and, at the same time, enable reception by setting the CRXEn bit to 1.
<6> Confirm that the SFA3n.SFFULn bit is 0, and then write first CS data to the SFCS3n register and subsequently write dummy transfer data to the SFDB3n register (reception start trigger).
If it is clearly known that the SFA3n.SFFULn bit is 0, because dummy transfer data is written to that bit by the interrupt servicing routine of INTC3n, it is not always necessary to confirm that the SFA3n.SFFULn bit is 0.
<7> Confirm that the INTC3n interrupt has occurred, and then read the SIRB3n register.
<8> Confirm that the INTC3n interrupt has occurred and the SFA3n.SFEMPn bit is 1, and disable reception by clearing the CSIM3n.CRXEn bit to 0 (end of reception).

Remarks
1. To execute a further transfer, repeat <6> and <7> before <8>.
Perform writing dummy transfer data in <6> every time transfer is executed.
2. The SO3n pin output is fixed to low level (default value).

17.6.3 Single mode (master mode, transmission/reception mode)

Example Conditions

MSB First (CSIM3n.DIRn bit = 0)
CSIC3n.CKPn bit = 1, CSIC3n.DAPn bit = 0
Transfer Data Length: 8 Bits (CSIL3n.CCLn[3:0] bits = 1000_B)
INTC3n Interrupt Not Delayed (CSIM3n.CSIT bit = 0),
Transfer Wait: Disabled (CSIM3n.CSWE bit = 0),
Chip Select Active Level: L-Level (CSIL3n.CSLVn[3:0] bits = 0000_B)

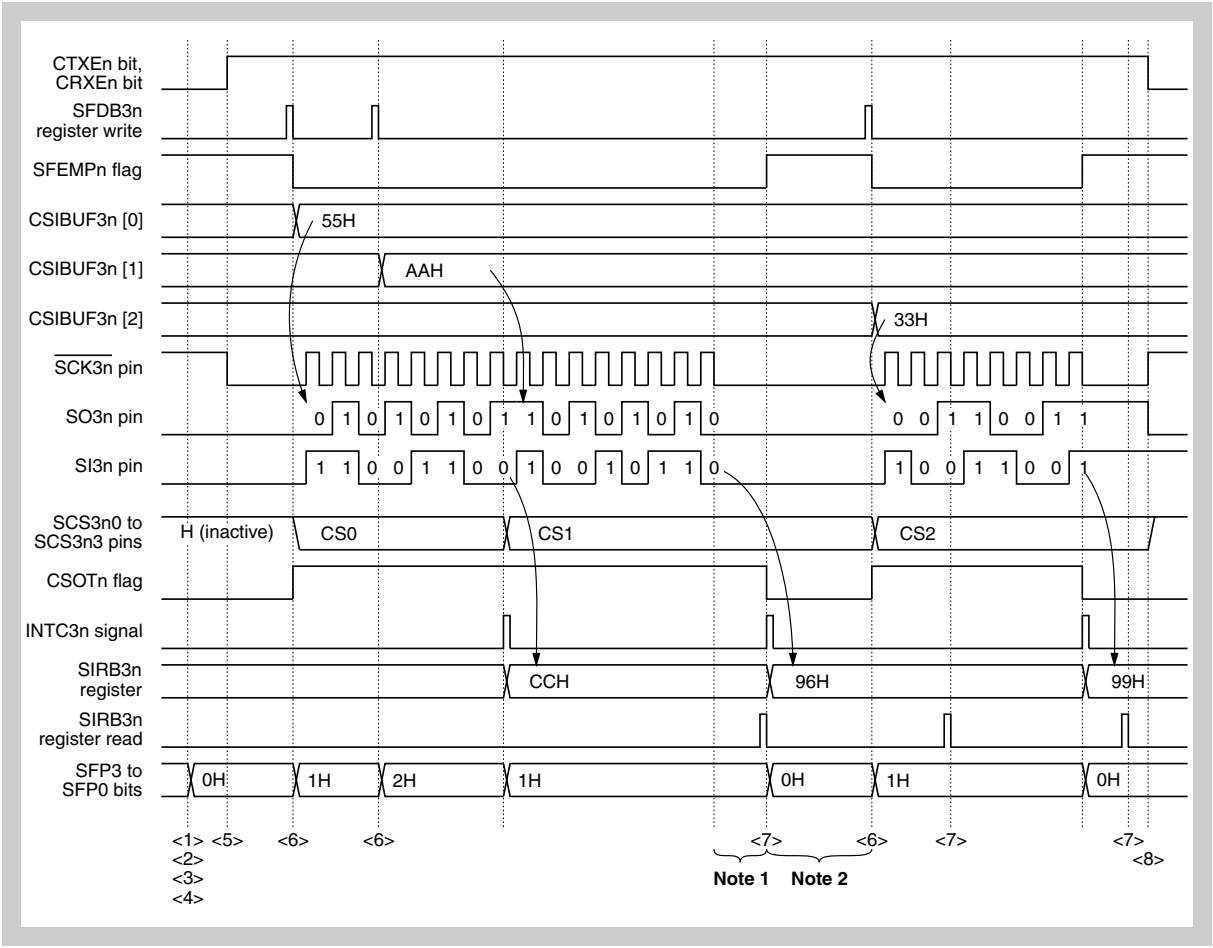


Figure 17-18 Single Mode (Master Mode, Transmission/Reception Mode)

Note

1. While the SIRB3n register is full a new transfer start of reception from the slave is put on hold until the SIRB3n register is read.
2. During this period a reception from the slave is put on hold until at least one transmit data has been loaded to the CSIBUFn register by writing the SFDB3n register (SFA3n.SFEMPn flag = 0) in order to start the transfer.

<1> When the CSIM3n.CSICAEn bit of the register is set to 1, operating clock supply is enabled.
<2> Specify the transfer mode by setting the CSIC3n and CSIL3n registers.
<3> Write 1 to the SFA3n.FPCLRn bit to clear all the CSIBUFn pointers to 0.
<4> Confirm that the SFA3n.SFFULn bit = 0, SFA3n.SFEMPn bit = 1, and SFA3n.SFPn[3:0] bits = 0000_B.
<5> Specify the transfer mode by using the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits of the CSIM3n register and, at the same time, enable transmission/reception by setting the CTXEn and CRXEn bits to 1.
<6> Confirm that the SFA3n.SFFULn bitis 0, and then write first CS data to the SFCS3n register and subsequently write transfer data to the SFDB3n register.
If it is clearly known that the SFA3n.SFFULn bit is 0, because transfer data is written to that bit by the interrupt servicing routine of INTC3n, it is not always necessary to confirm that the SFA3n.SFFULn bit is 0.
<7> Confirm that the INTC3n interrupt has occurred, and then read the SIRB3n register.
<8> Confirm that the INTC3n interrupt has occurred and the SFA3n.SFEMPn bit is 1, and disable transmission/reception by clearing the CSIM3n.CTXEn and CSIM3n.CRXEn bits to 0 (end of transmission/reception).

Remark To execute a further transfer, repeat <6> and <7> before <8>.

17.6.4 Single mode (slave mode, transmission mode)

Example Conditions

MSB First (CSIM3n.DIRn bit = 0)
CSIC3n.CKPn bit = 1, CSIC3n.DAPn bit = 1
Transfer Data Length: 8 Bits (CSIL3n.CCLn[3:0] bits = 1000_B)
INTC3n Interrupt Not Delayed (CSIM3n.CSIT bit = 0),
Transfer Wait: Disabled (CSIM3n.CSWE bit = 0),
Chip Select Active Level: L-Level (CSIL3n.CSLVn[3:0] bits = 0000_B)

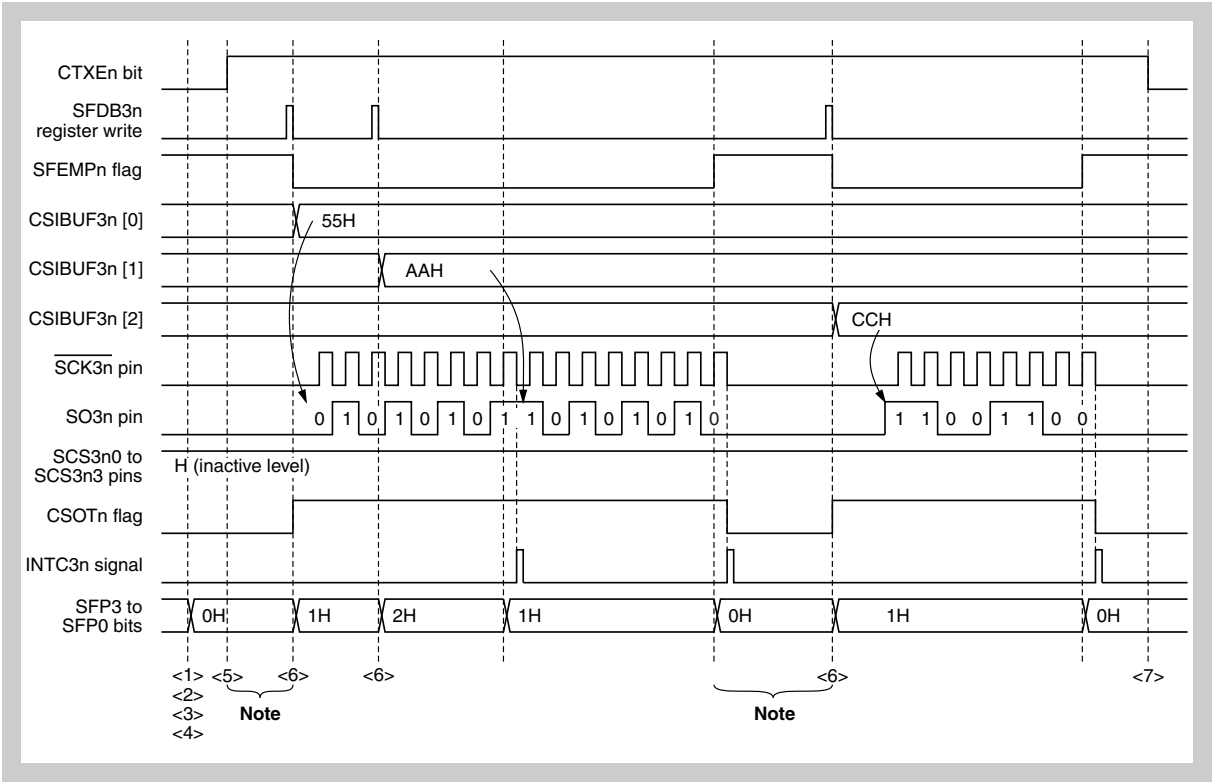


Figure 17-19 Single Mode (Slave Mode, Transmission Mode)

Note During this period a transmission to the master will be ignored until at least one transmit data is loaded to the CSIBUFn register by writing the SFDB3n register (SFA3n.SFEMPn flag = 0).

<1> When the CSIM3n.CSICAEn bit is set to 1, operating clock supply is enabled.
<2> Specify the transfer mode by setting the CSIC3n and CSIL3n registers.
<3> Write 1 to the SFA3n.FPCLRn bit to clear all the CSIBUFn pointers to 0.
<4> Confirm that the SFA3n.SFFULn bit = 0, SFA3n.SFEMPn bit = 1, and SFA3n.SFPn[3:0] bits = 0000_Br.
<5> Specify the transfer mode by using the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits of the CSIM3n register and, at the same time, enable transmission by setting the CTXEn bit to 1.
<6> Confirm that the SFA3n.SFFULn bit is 0, and then write transfer data to the SFDB3n register.
Since the chip select outputs (SCS3n0 to SCS3n3) are ineffective in the slave mode and always output the inactive level, writing of CS data to the SFCS3n register is not necessary.
If it is clearly known that the SFA3n.SFFULn bit is 0, because transfer data is written to that bit by the interrupt servicing routine of INTC3n, it is not always necessary to confirm that the SFA3n.SFFULn bit is 0.
<7> Confirm that the INTC3n interrupt has occurred and the SFA3n.SFEMPn bit is 1, and disable transmission by clearing the CSIM3n.CTXEn bit to 0 (end of transmission).

Remark To execute a further transfer, repeat <6> before <7>.

17.6.5 Single mode (slave mode, reception mode)

Example Conditions

MSB First (CSIM3n.DIRn bit = 0)
CSIC3n.CKPn bit = 0, CSIC3n.DAPn bit = 0
Transfer Data Length: 8 Bits (CSIL3n.CCLn[3:0] bits = 1000_B)
INTC3n Interrupt Not Delayed (CSIM3n.CSIT bit = 0),
Transfer Wait: Disabled (CSIM3n.CSWE bit = 0),
Chip Select Active Level: L-Level (CSIL3n.CSLVn[3:0] bits = 0000_B)

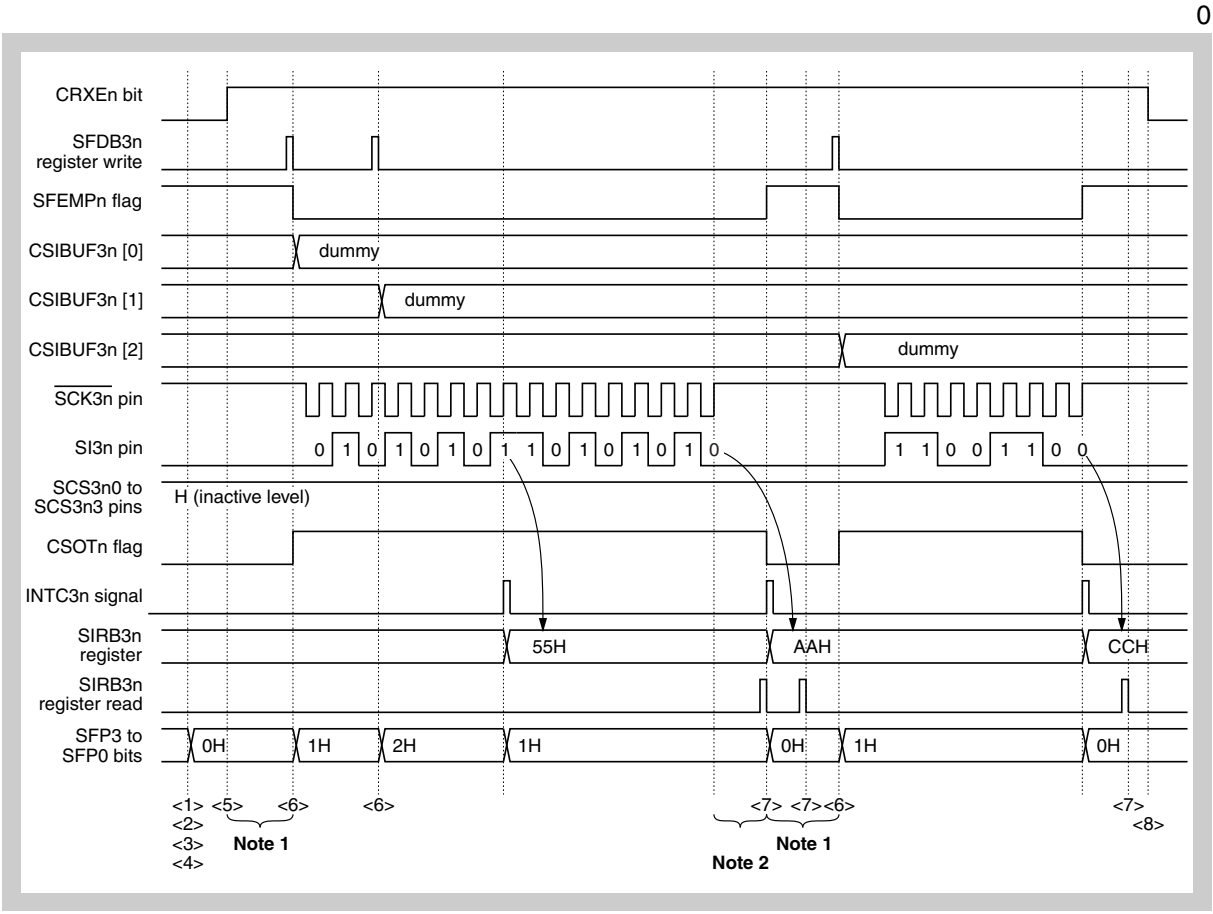


Figure 17-20 Single Mode (Slave Mode, Reception Mode)

Note

1. During this period a transmission/reception from the master will be ignored until at least one dummy transmit data is loaded to the CSIBUFn register by writing the SFDB3n register (SFA3n.SFEMPn flag = 0).
2. While the SIRB3n register is full a new reception from the master will be ignored until the SIRB3n register is read.

<1> When the CSIM3n.CSICAEn bit is set to 1, operating clock supply is enabled.
<2> Specify the transfer mode by setting the CSIC3n and CSIL3n registers.
<3> Write 1 to the SFA3n.FPCLRn bit to clear all the CSIBUFn pointers to 0.
<4> Confirm that the SFA3n.SFFULn bit = 0, SFA3n.SFEMPn bit = 1, and SFA3n.SFPn[3:0] bits = 0000_B.
<5> Specify the transfer mode by using the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits of the CSIM3n register and, at the same time, enable reception by setting the CRXEn bit to 1.
<6> Confirm that the SFA3n.SFFULn bit is 0, and then write dummy transfer data to the SFDB3n register (reception start trigger).
Since the chip select outputs (SCS3n0 to SCS3n3) are ineffective in the slave mode and always output the inactive level, writing of CS data to the SFCS3n register is not necessary.
If it is clearly known that the SFA3n.SFFULn bit is 0, because dummy transfer data is written to that bit by the interrupt servicing routine of INTC3n, it is not always necessary to confirm that the SFA3n.SFFULn bit is 0.
<7> Confirm that the INTC3n interrupt has occurred, and then read the SIRB3n register.
<8> Confirm that the INTC3n interrupt has occurred and the SFA3n.SFEMPn bit is 1, and disable reception by clearing the CSIM3n.CRXEn bit to 0 (end of reception).

Remarks
1. To execute a further transfer, repeat <6> and <7> before <8>.
Perform writing dummy transfer data in <6> every time transfer is executed.
2. The SO3n pin output is fixed to low level (default value).

17.6.6 Single mode (slave mode, transmission/reception mode)

Example Conditions

MSB First (CSIM3n.DIRn bit = 0)
CSIC3n.CKPn bit = 0, CSIC3n.DAPn bit = 1
Transfer Data Length: 8 Bits (CSIL3n.CCLn[3:0] bits = 1000_B)
INTC3n Interrupt Not Delayed (CSIM3n.CSIT bit = 0),
Transfer Wait: Disabled (CSIM3n.CSWE bit = 0),
Chip Select Active Level: L-Level (CSIL3n.CSLVn[3:0] bits = 0000_B)

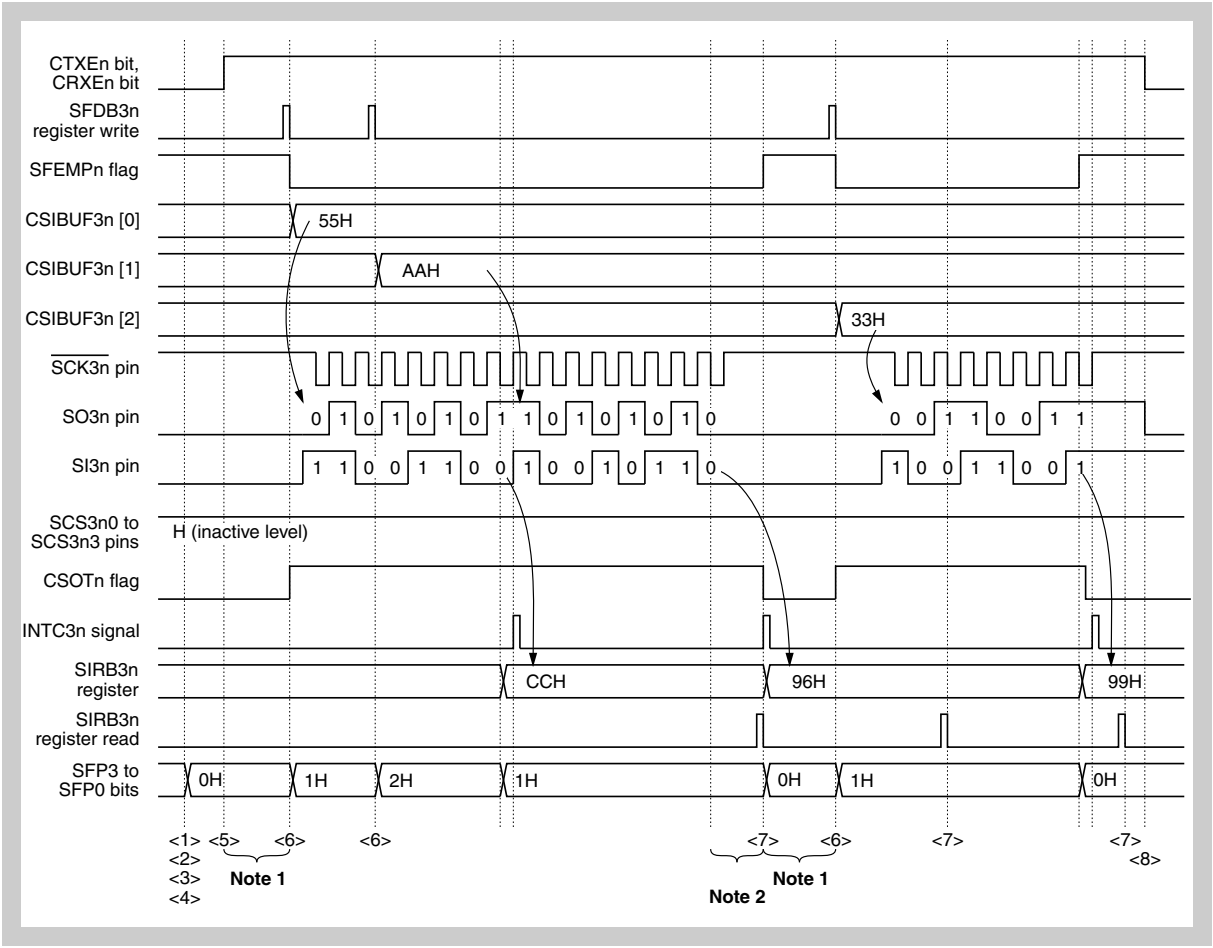


Figure 17-21 Single Mode (Slave Mode, Transmission/Reception Mode)

Note

1. During this period a transmission/reception from the master will be ignored until at least one transmit data is loaded to the CSIBUFn register by writing the SFDB3n register (SFA3n.SFEMPn flag = 0).
2. While the SIRB3n register is full a new transmission/reception from the master will be ignored until the SIRB3n register is read.

<1> When the CSIM3n.CSICAEn bit is set to 1, operating clock supply is enabled.
<2> Specify the transfer mode by setting the CSIC3n and CSIL3n registers.
<3> Write 1 to the SFA3n.FPCLRn bit to clear all the CSIBUFn pointers to 0.
<4> Confirm that the SFA3n.SFFULn bit = 0, SFA3n.SFEMPn bit = 1, and SFA3n.SFPn[3:0] bits = 0000_B.
<5> Specify the transfer mode by using the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits of the CSIM3n register and, at the same time, enable transmission/reception by setting the CTXEn and CRXEn bits to 1.
<6> Confirm that the SFA3n.SFFULn bit is 0, and then write transfer data to the SFDB3n register.
Since the chip select outputs (SCS3n0 to SCS3n3) are ineffective in the slave mode and always output the inactive level, writing of CS data to the SFCS3n register is not necessary.
If it is clearly known that the SFA3n.SFFULn bit is 0, because transfer data is written to that bit by the interrupt servicing routine of INTC3n, it is not always necessary to confirm that the SFA3n.SFFULn bit is 0.
<7> Confirm that the INTC3n interrupt has occurred, and then read the SIRB3n register.
<8> Confirm that the INTC3n interrupt has occurred and the SFA3n.SFEMPn bit is 1, and disable transmission/reception by clearing the CSIM3n.CTXEn and CSIM3n.CRXEn bits to 0 (end of transmission/reception).

Remark To execute a further transfer, repeat <6> and <7> before <8>.

17.6.7 Consecutive mode (master mode, transmission mode)

Example Conditions

MSB First (CSIM3n.DIRn bit = 0)
CSIC3n.CKPn bit = 0, CSIC3n.DAPn bit = 0
Transfer Data Length: 8 Bits (CSIL3n.CCLn[3:0] bits = 1000_B)
INTC3n Interrupt Not Delayed (CSIM3n.CSIT bit = 0),
Transfer Wait: Disabled (CSIM3n.CSWE bit = 0),
Chip Select Active Level: L-Level (CSIL3n.CSLVn[3:0] bits = 0000_B)

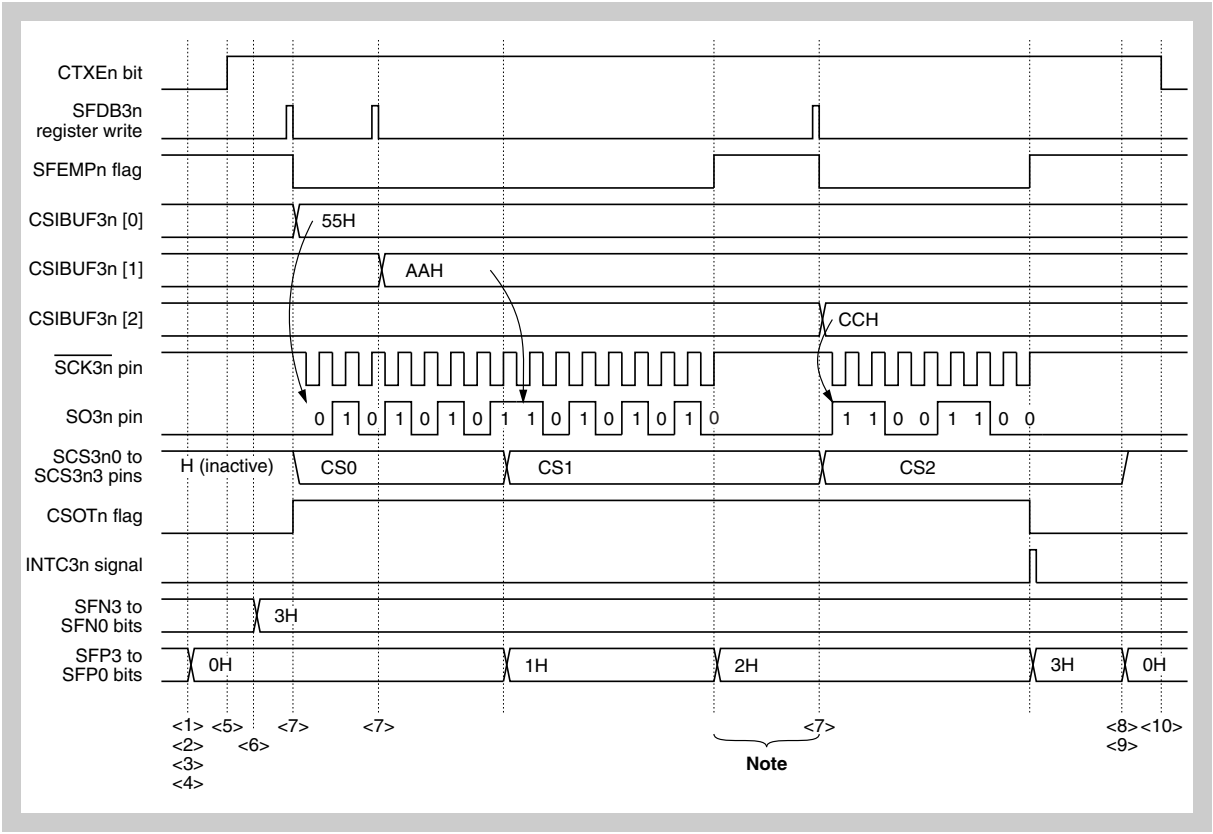


Figure 17-22 Consecutive Mode (Master Mode, Transmission Mode)

Note During this period a reception from the slave is put on hold until at least one transmit data has been loaded to the CSIBUFn register by writing the SFDB3n register (SFA3n.SFEMPn flag = 0) in order to start the transfer.

<1> When the CSIM3n.CSICAEn bit is set to 1, operating clock supply is enabled.
<2> Specify the transfer mode by setting the CSIC3n and CSIL3n registers.
<3> Write 1 to the SFA3n.FPCLRn bit to clear all the CSIBUFn pointers to 0.
<4> Confirm that the SFA3n.SFFULn bit = 0, SFA3n.SFEMPn bit = 1, and SFA3n.SFPn[3:0] bits = 0000_B.
<5> Specify the transfer mode by using the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits of the CSIM3n register and, at the same time, enable transmission by setting the CTXEn bit to 1.
<6> Set the number of data to be transmitted by using the SFN3n.SFNn[3:0] bits
<7> Write first CS data to the SFCS3n register and subsequently write transfer data to the SFDB3n register. Writing data exceeding the set value of the SFN3n register is prohibited.
<8> Confirm that the INTC3n interrupt has occurred and the SFA3n.SFEMPn bit is 1. Then write 1 to the SFA3n.FPCLRn bit , and clear all the CSIBUFn pointers to 0 in preparation for the next transfer.
<9> Confirm that the SFA3n.SFFULn bit = 0, SFA3n.SFEMPn bit = 1, and SFA3n.SFPn[3:0] bits = 0000_B.
<10> Disable transmission by clearing the CSIM3n.CTXEn bit to 0 (end of transmission).

Remark To execute a further transfer, repeat <6> to <9> before <10>.

17.6.8 Consecutive mode (master mode, reception mode)

Example Conditions

MSB First (CSIM3n.DIRn bit = 0)
CSIC3n.CKPn bit = 0, CSIC3n.DAPn bit = 1
Transfer Data Length: 8 Bits (CSIL3n.CCLn[3:0] bits = 1000_B)
INTC3n Interrupt Not Delayed (CSIM3n.CSIT bit = 0),
Transfer Wait: Disabled (CSIM3n.CSWE bit = 0),
Chip Select Active Level: L-Level (CSIL3n.CSLVn[3:0] bits = 0000_B)

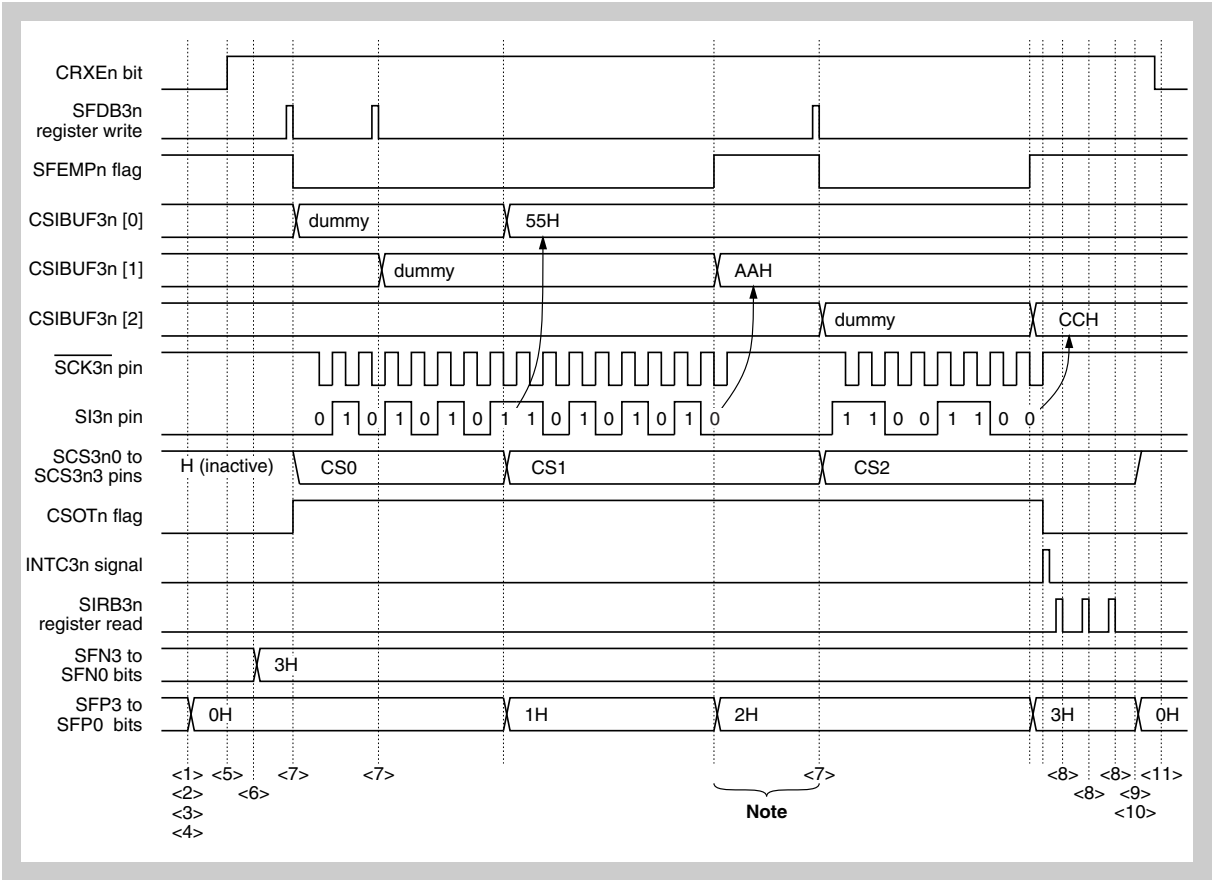


Figure 17-23 Consecutive Mode (Master Mode, Reception Mode)

Note During this period a reception from the slave is put on hold until at least one dummy transmit data has been loaded to the CSIBUFn register by writing the SFDB3n register (SFA3n.SFEMPn flag = 0) in order to start the transfer.

<1> When the CSIM3n.CSICAEn bit is set to 1, operating clock supply is enabled.
<2> Specify the transfer mode by setting the CSIC3n and CSIL3n registers.
<3> Write 1 to the SFA3n.FPCLRn bit to clear all the CSIBUFn pointers to 0.
<4> Confirm that the SFA3n.SFFULn bit = 0, SFA3n.SFEMPn bit = 1, and SFA3n.SFPn[3:0] bits = 0000_B.
<5> Specify the transfer mode by using the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits of the CSIM3n register and, at the same time, enable reception by setting the CRXEn bit to 1.
<6> Set the number of data to be received by using the SFN3n.SFNn[3:0] bits.
<7> Write first CS data to the SFCS3n register and subsequently write dummy transfer data to the SFDB3n register (reception start trigger). Writing dummy data exceeding the set value of the SFN3n register is prohibited.
<8> Confirm that the INTC3n interrupt has occurred and the SFA3n.SFEMPn bit is 1. Then read the SIRB3n register (sequentially read the receive data stored in the CSIBUFn register).
<9> Write 1 to the SFA3n.FPCLRn bit, and clear all the CSIBUFn pointers to 0 in preparation for the next transfer.
<10> Confirm that the SFA3n.SFFULn bit = 0, SFA3n.SFEMPn bit = 1, and SFA3n.SFPn[3:0] bits = 0000_B.
<11> Disable reception by clearing the CSIM3n.CRXEn bit to 0 (end of reception).

Remarks
1. To execute a further transfer, repeat <6> to <10> before <11>.
   Perform writing dummy transfer data in <7> every time transfer is executed.
2. The SO3n pin output is fixed to low level (default value).

17.6.9 Consecutive mode (master mode, transmission/reception mode)

Example Conditions

MSB First (CSIM3n.DIRn bit = 0)
CSIC3n.CKPn bit = 0, CSIC3n.DAPn bit = 1
Transfer Data Length: 8 Bits (CSIL3n.CCLn[3:0] bits = 1000_B)
INTC3n Interrupt Not Delayed (CSIM3n.CSIT bit = 0),
Transfer Wait: Disabled (CSIM3n.CSWE bit = 0),
Chip Select Active Level: L-Level (CSIL3n.CSLVn[3:0] bits = 0000_B)

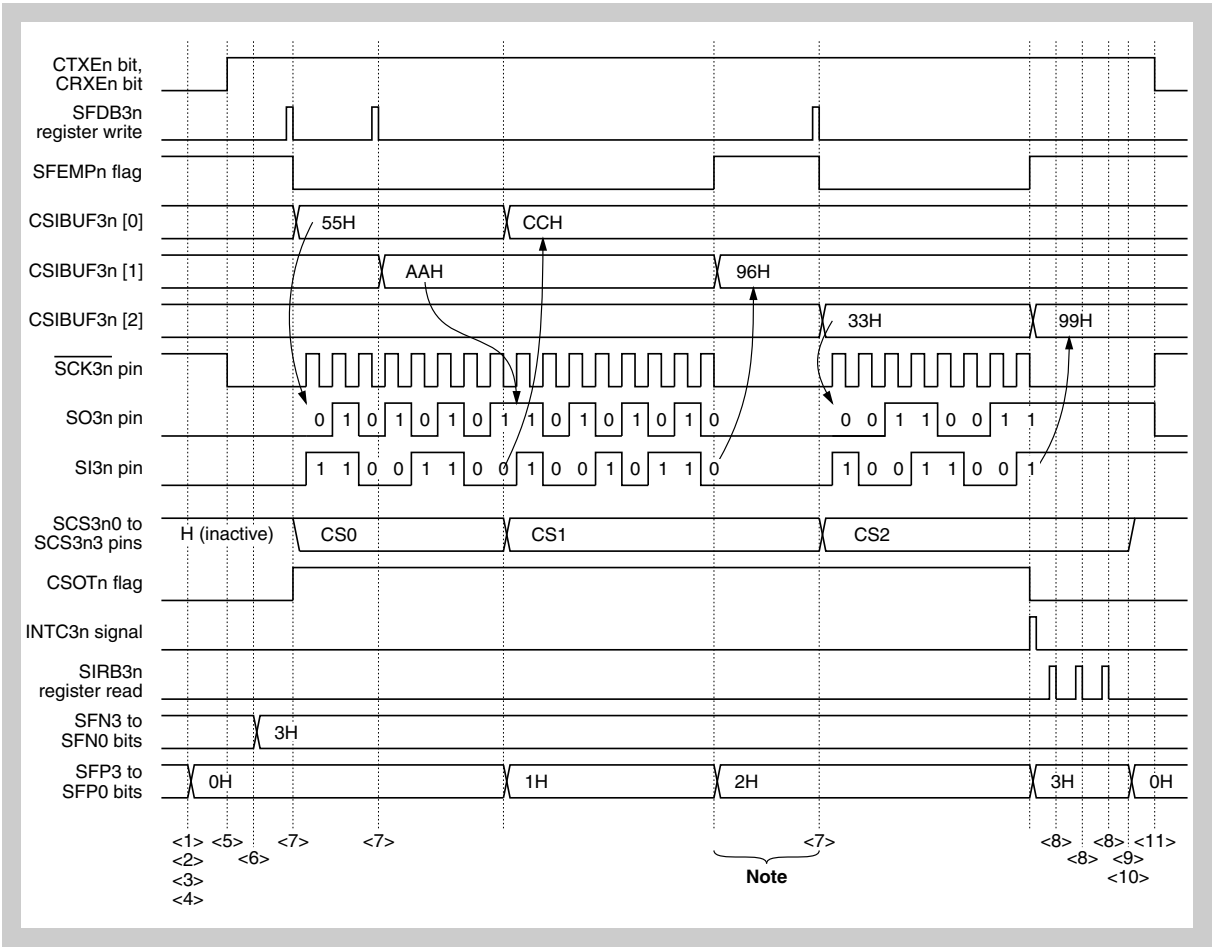


Figure 17-24 Consecutive Mode (Master Mode, Transmission/Reception Mode)

Note During this period a reception from the slave is put on hold until at least one transmit data has been loaded to the CSIBUFn register by writing the SFDB3n register (SFA3n.SFEMPn flag = 0) in order to start the transfer.

<1> When the CSIM3n.CSICAEn bit is set to 1, operating clock supply is enabled.
<2> Specify the transfer mode by setting the CSIC3n and CSIL3n registers.
<3> Write 1 to the SFA3n.FPCLRn bit to clear all the CSIBUFn pointers to 0.
<4> Confirm that the SFA3n.SFFULn bit = 0, SFA3n.SFEMPn bit = 1, and SFA3n.SFPn[3:0] bits = 0000_B.
<5> Specify the transfer mode by using the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits of the CSIM3n register and, at the same time, enable transmission/reception by setting both the CTXEn and CRXEn bits to 1.
<6> Set the number of data to be transmitted/received by using the SFN3n.SFNn[3:0] bits.
<7> Write first CS data to the SFCS3n register and subsequently write transfer data to the SFDB3n register. Writing data exceeding the set value of the SFN3n register is prohibited.
<8> Confirm that the INTC3n interrupt has occurred and the SFA3n.SFEMPn bit is 1. Then read the SIRB3n register (sequentially read the receive data stored in the CSIBUFn register).
<9> Write 1 to the SFA3n.FPCLRn bit , and clear all the CSIBUFn pointers to 0 in preparation for the next transfer.
<10> Confirm that the SFA3n.SFFULn bit = 0, SFA3n.SFEMPn bit = 1, and SFA3n.SFPn[3:0] bits = 0000_B.
<11> Disable transmission/reception by clearing the CSIM3n.CTXEn and CSIM3n.CRXEn bits to 0 (end of transmission/reception).

Remark To execute a further transfer, repeat <6> to <10> before <11>.

17.6.10 Consecutive mode (slave mode, transmission mode)

Example Conditions

MSB First (CSIM3n.DIRn bit = 0)
CSIC3n.CKPn bit = 1, CSIC3n.DAPn bit = 1
Transfer Data Length: 8 Bits (CSIL3n.CCLn[3:0] bits = 1000_B)
INTC3n Interrupt Not Delayed (CSIM3n.CSIT bit = 0),
Transfer Wait: Disabled (CSIM3n.CSWE bit = 0),
Chip Select Active Level: L-Level (CSIL3n.CSLVn[3:0] bits = 0000_B)

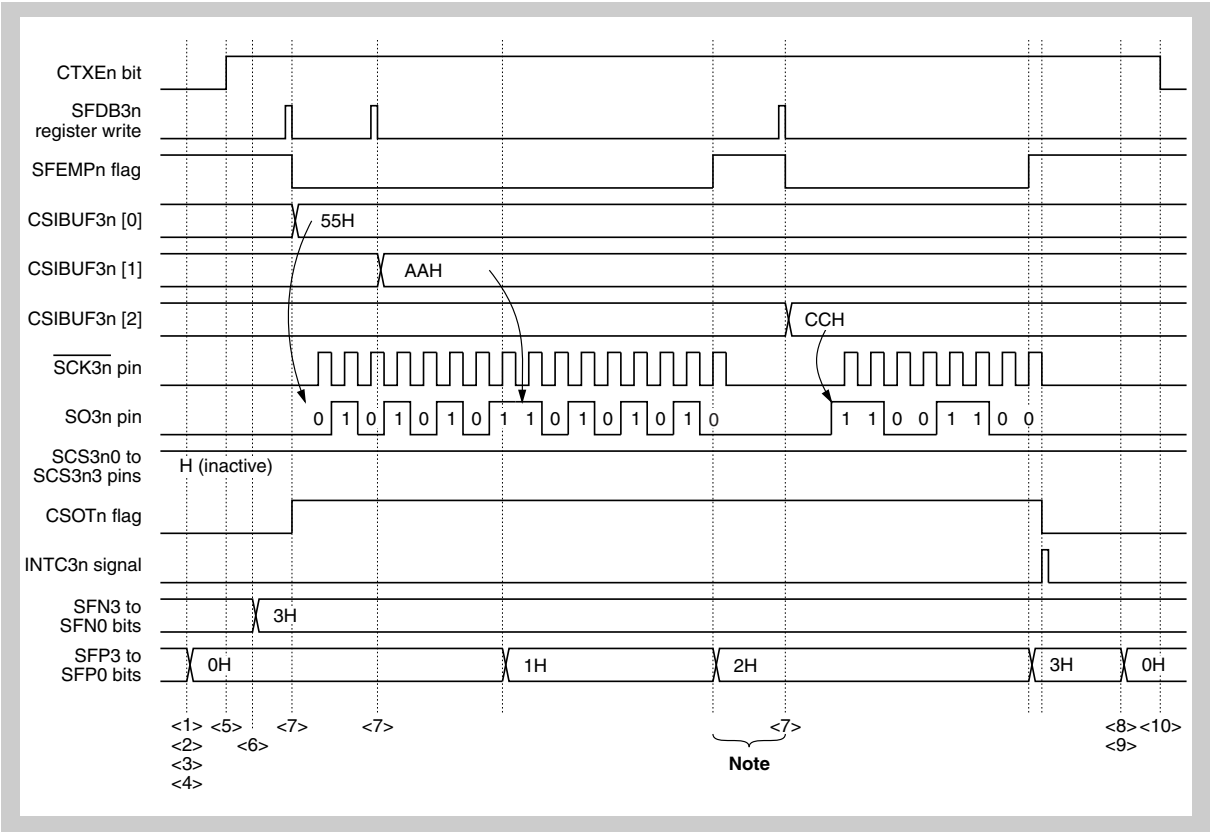


Figure 17-25 Consecutive Mode (Slave Mode, Transmission Mode)

Note During this period a reception request from the master will be ignored until at least one transmit data is loaded to the CSIBUFn register by writing the SFDB3n register (SFA3n.SFEMPn flag = 0).

<1> When the CSIM3n.CSICAEn bit is set to 1, operating clock supply is enabled.
<2> Specify the transfer mode by setting the CSIC3n and CSIL3n registers.
<3> Write 1 to the SFA3n.FPCLRn bit to clear all the CSIBUFn pointers to 0.
<4> Confirm that the SFA3n.SFFULn bit = 0, SFA3n.SFEMPn bit = 1, and SFA3n.SFPn[3:0] bits = 0000_B.
<5> Specify the transfer mode by using the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits of the CSIM3n register and, at the same time, enable transmission by setting the CTXEn bit to 1.
<6> Set the number of data to be transmitted by using the SFN3n.SFNn[3:0] bits.
<7> Write transfer data to the SFDB3n register. Writing data exceeding the set value of the SFN3n register is prohibited.
Since the chip select outputs (SCS3n0 to SCS3n3) are ineffective in the slave mode and always output the inactive level, writing of CS data to the SFCS3n register is not necessary.
<8> Confirm that the INTC3n interrupt has occurred and the SFA3n.SFEMPn bit is 1. Then write 1 to the SFA3n.FPCLRn bit , and clear all the CSIBUFn pointers to 0 in preparation for the next transfer.
<9> Confirm that the SFA3n.SFFULn bit = 0, SFA3n.SFEMPn bit = 1, and SFA3n.SFPn[3:0] bits = 0000_B.
<10> Disable transmission by clearing the CSIM3n.CTXEn bit to 0 (end of transmission).

Remarks **1.** To execute a further transfer, repeat <6> to <9> before <10>.

17.6.11 Consecutive mode (slave mode, reception mode)

Example Conditions

MSB First (CSIM3n.DIRn bit = 0)
CSIC3n.CKPn bit = 0, CSIC3n.DAPn bit = 0
Transfer Data Length: 8 Bits (CSIL3n.CCLn[3:0] bits = 1000_B)
INTC3n Interrupt Not Delayed (CSIM3n.CSIT bit = 0),
Transfer Wait: Disabled (CSIM3n.CSWE bit = 0),
Chip Select Active Level: L-Level (CSIL3n.CSLVn[3:0] bits = 0000_B)

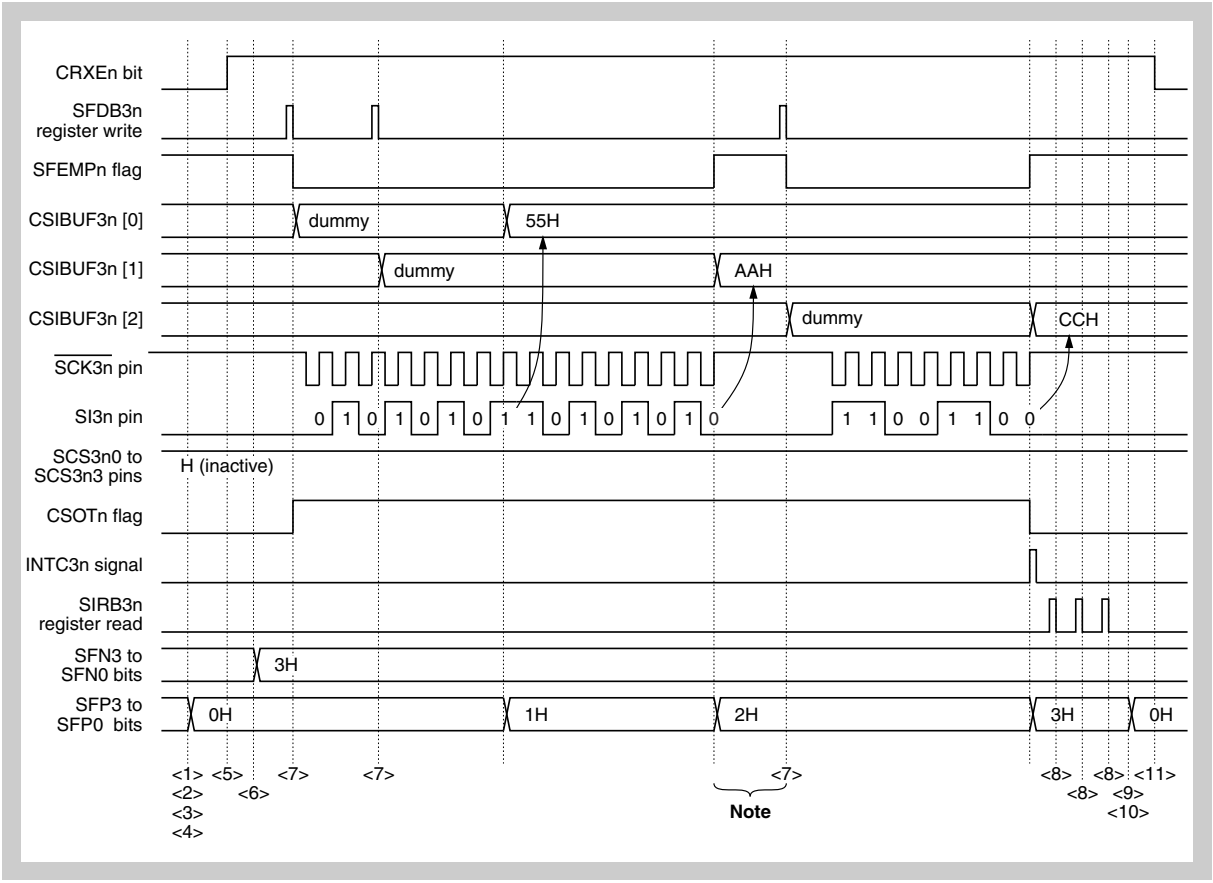


Figure 17-26 Consecutive Mode (Slave Mode, Reception Mode)

Note During this period a transmission from the master will be ignored until at least one transmit data is loaded to the CSIBUFn register by writing the SFDB3n register (SFA3n.SFEMPn flag = 0).

<1> When the CSIM3n.CSICAEn bit is set to 1, operating clock supply is enabled.
<2> Specify the transfer mode by setting the CSIC3n and CSIL3n registers.
<3> Write 1 to the SFA3n.FPCLRn bit to clear all the CSIBUFn pointers to 0.
<4> Confirm that the SFA3n.SFFULn bit = 0, SFA3n.SFEMPn bit = 1, and SFA3n.SFPn[3:0] bits = 0000_B.
<5> Specify the transfer mode by using the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits of the CSIM3n register and, at the same time, enable reception by setting the CRXEn bit to 1.
<6> Set the number of data to be received by using the SFN3n.SFNn[3:0] bits.
<7> Write dummy transfer data to the SFDB3n register (reception start trigger). Writing dummy data exceeding the set value of the SFN3n register is prohibited.
Since the chip select outputs (SCS3n0 to SCS3n3) are ineffective in the slave mode and always output the inactive level, writing of CS data to the SFCS3n register is not necessary.
<8> Confirm that the INTC3n interrupt has occurred and the SFA3n.SFEMPn bit is 1. Then read the SIRB3n register (sequentially read the receive data stored in the CSIBUFn register).
<9> Write 1 to the SFA3n.FPCLRn bit, and clear all the CSIBUFn pointers to 0 in preparation for the next transfer.
<10> Confirm that the SFA3n.SFFULn bit = 0, SFA3n.SFEMPn bit = 1, and SFA3n.SFPn[3:0] bits = 0000_B.
<11> Disable reception by clearing the CSIM3n.CRXEn bit to 0 (end of reception).

Remarks
1. To execute a further transfer, repeat <6> to <10> before <11>.
Perform writing dummy transfer data in <7> every time transfer is executed.
2. The SO3n pin output is fixed to low level (default value).

17.6.12 Consecutive mode (in slave mode and transmission/reception mode)

Example Conditions

MSB First (CSIM3n.DIRn bit = 0)
CSIC3n.CKPn bit = 0, CSIC3n.DAPn bit = 1
Transfer Data Length: 8 Bits (CSIL3n.CCLn[3:0] bits = 1000_B)
INTC3n Interrupt Not Delayed (CSIM3n.CSIT bit = 0),
Transfer Wait: Disabled (CSIM3n.CSWE bit = 0),
Chip Select Active Level: L-Level (CSIL3n.CSLVn[3:0] bits = 0000_B)

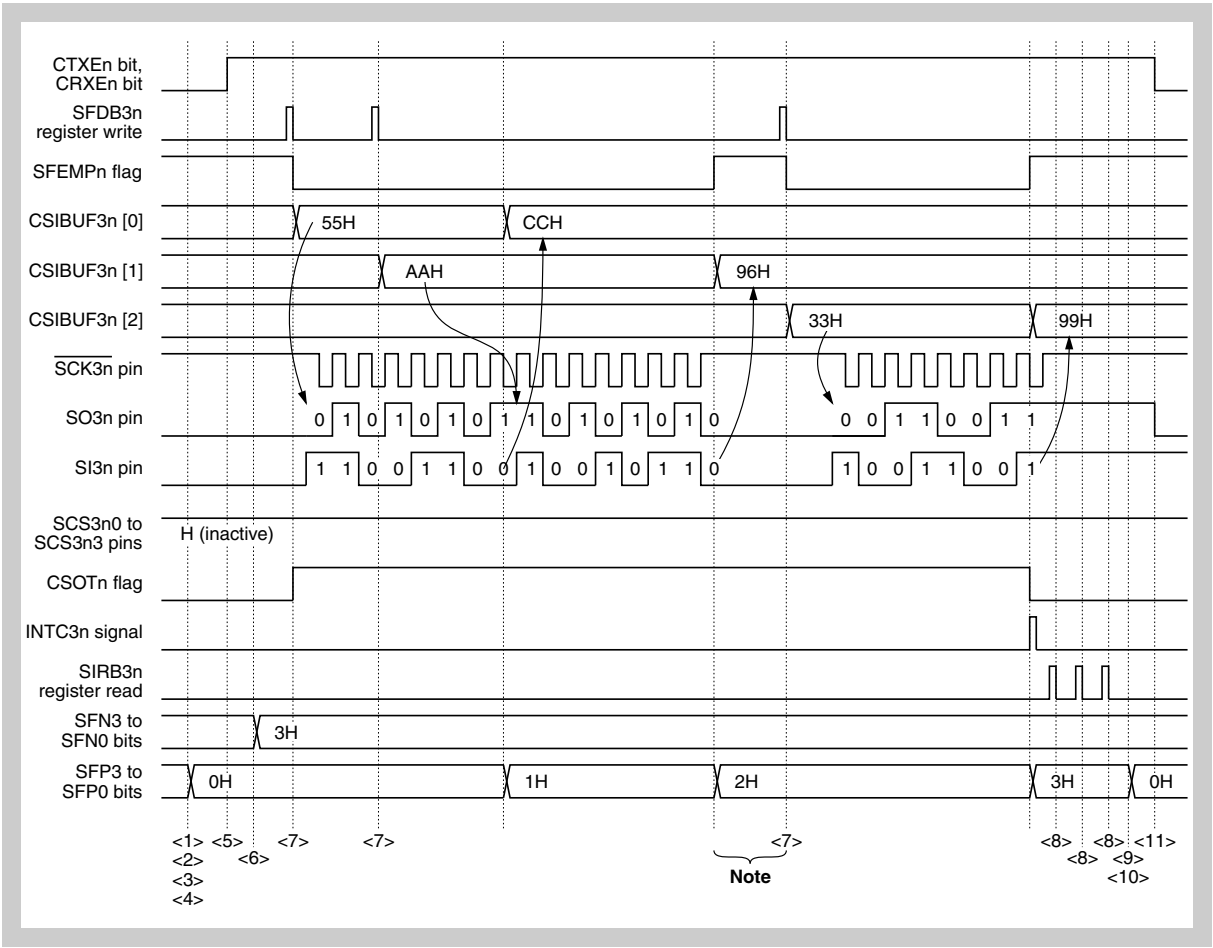


Figure 17-27 Consecutive Mode (Slave Mode, Transmission/Reception Mode)

Note During this period a transmission/reception from the master will be ignored until at least one transmit data is loaded to the CSIBUFn register by writing the SFDB3n register (SFA3n.SFEMPn flag = 0).

<1> When the CSIM3n.CSICAEn bit is set to 1, operating clock supply is enabled.
<2> Specify the transfer mode by setting the CSIC3n and CSIL3n registers.
<3> Write 1 to the SFA3n.FPCLRn bit to clear all the CSIBUFn pointers to 0.
<4> Confirm that the SFA3n.SFFULn bit = 0, SFA3n.SFEMPn bit = 1, and SFA3n.SFPn[3:0] bits = 0000_B.
<5> Specify the transfer mode by using the TRMDn, DIRn, CSITn, CSWEn, and CSMDn bits of the CSIM3n register and, at the same time, enable transmission/reception by setting both the CTXEn and CRXEn bits to 1.
<6> Set the number of data to be transmitted/received by using the SFN3n.SFNn[3:0] bits.
<7> Write transfer data to the SFDB3n register. Writing data exceeding the set value of the SFN3n register is prohibited.
Since the chip select outputs (SCS3n0 to SCS3n3) are ineffective in the slave mode and always output the inactive level, writing of CS data to the SFCS3n register is not necessary.
<8> Confirm that the INTC3n interrupt has occurred and the SFA3n.SFEMPn bit is 1. Then read the SIRB3n register (sequentially read the receive data stored in the CSIBUFn register).
<9> Write 1 to the SFA3n.FPCLRn bit, and clear all the CSIBUFn pointers to 0 in preparation for the next transfer.
<10> Confirm that the SFA3n.SFFULn bit = 0, SFA3n.SFEMPn bit = 1, and SFA3n.SFPn[3:0] bits = 0000_B.
<11> Disable transmission/reception by clearing the CSIM3n.CTXEn and CSIM3n.CRXEn bits to 0 (end of transmission/reception).

Remark To execute a further transfer, repeat <6> to <10> before <11>.

17.7 Cautions

The following points must be observed when using CSI3n.

(1) Starting and stopping of CSI3n

The CSI3n unit is reset and CSI3n is stopped when the CSIM3n.CSICAEn bit is cleared to 0. To operate CSI3n, first set the CSIM3n.CSICAEn bit to 1. Usually, before clearing the CSIM3n.CSICAEn bit to 0, clear both the CSIM3nCTXEn and CSIM3n.CRXEn bits to 0 (after the end of transfer).

(2) Clear all buffer pointers before transfer start

Be sure to write 1 to the SFA3n.FPCLRn bit to clear all the CSIBUFn pointers to 0 before enabling transfer by setting the CSIM3n.CTXEn or CSIM3n.CRXEn bit to 1. If the CSIM3n.CTXEn or CSIM3n.CRXEn bit is set to 1 without clearing the pointers, and if the previously transferred data remains in the CSIBUFn register, transferring that data is immediately started.
If transfer data is set to the CSIBUFn register before transfer is enabled, transfer is started as soon as the CSIM3n.CTXEn or CSIM3n.CRXEn bit is set to 1.

(3) Timing of status flags

If the SFA3n register is read immediately after data has been written to the SFDB3n and SFDB3nL registers, the SFFULn, SFEMPn, and SFPn[3:0] bits of the SFA3n register may not change their values in time.
If the SFA3n register is read before the SFFULn bit is set to 1 and a 17th data is written, the CSIBUFn overflow interrupt (INTC3nOVF) occurs.

(4) DMA transfer of CSI3n data

When using CSI3n in configuration with DMA transfer, observe that only single mode is permitted (CSIM3n.TRMDn bit = 0), and chip select CSI registers (SFCS3n, SFCS3nL) are not supported.

Chapter 18 CAN Controller (CAN)

The microcontroller features an on-chip n-channel CAN (Controller Area Network) controller that complies with the CAN protocol as standardized in ISO 11898.

The V850E/PH2 microcontrollers have following number of channels of the CAN controller:

CAN	µPD70F3447	µPD70F3187
Instances	1	2
Names	CAN0	CAN0, CAN1

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 C0MDATA4m for CAN0 message data byte 4 of message buffer register m.

18.1 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 ≥ 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

18.1.1 Overview of functions

Table 18-1 presents an overview of the CAN Controller functions.

Table 18-1 Overview of functions

Function	Details
Protocol	CAN protocol ISO 11898 (standard and extended frame transmission/reception)
Baud rate	Maximum 1 Mbps (CAN clock input ≥ 8 MHz)
Data storage	Storing messages in the CAN RAM
Number of messages	• 32 message buffers per channel • Each message buffer can be set to be either a transmit message buffer or a receive message buffer.
Message reception	• Unique ID can be set to each message buffer. • Mask setting of four patterns is possible for each channel. • A receive completion interrupt is generated each time a message is received and stored in a message buffer. • Two or more receive message buffers can be used as a FIFO receive buffer (multi-buffer receive block function). • Receive history list function
Message transmission	• Unique ID can be set to each message buffer. • Transmit completion interrupt for each message buffer • Message buffer number 0 to 7 specified as the transmit message buffer can be set for automatic block transfer. Message transmission interval is programmable (automatic block transmission function (hereafter referred to as “ABT”)). • Transmission history list function
Remote frame processing	Remote frame processing by transmit message buffer
Time stamp function	• The time stamp function can be set for a message reception when a 16-bit timer is used in combination. • Time stamp capture trigger can be selected (SOF or EOF in a CAN message frame can be detected.). • The time stamp function can be set for a transmit message.
Diagnostic function	• Readable error counters • “Valid protocol operation flag” for verification of bus connections • Receive-only mode • Single-shot mode • CAN protocol error type decoding • Self-test mode
Release from bus-off state	• Forced release from bus-off (by ignoring timing constraint) possible by software. • No automatic release from bus-off (software must re-enable).
Power save mode	• CAN Sleep mode (can be woken up by CAN bus) • CAN Stop mode (cannot be woken up by CAN bus)

18.1.2 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.
- MAC (Memory Access Controller)
  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.

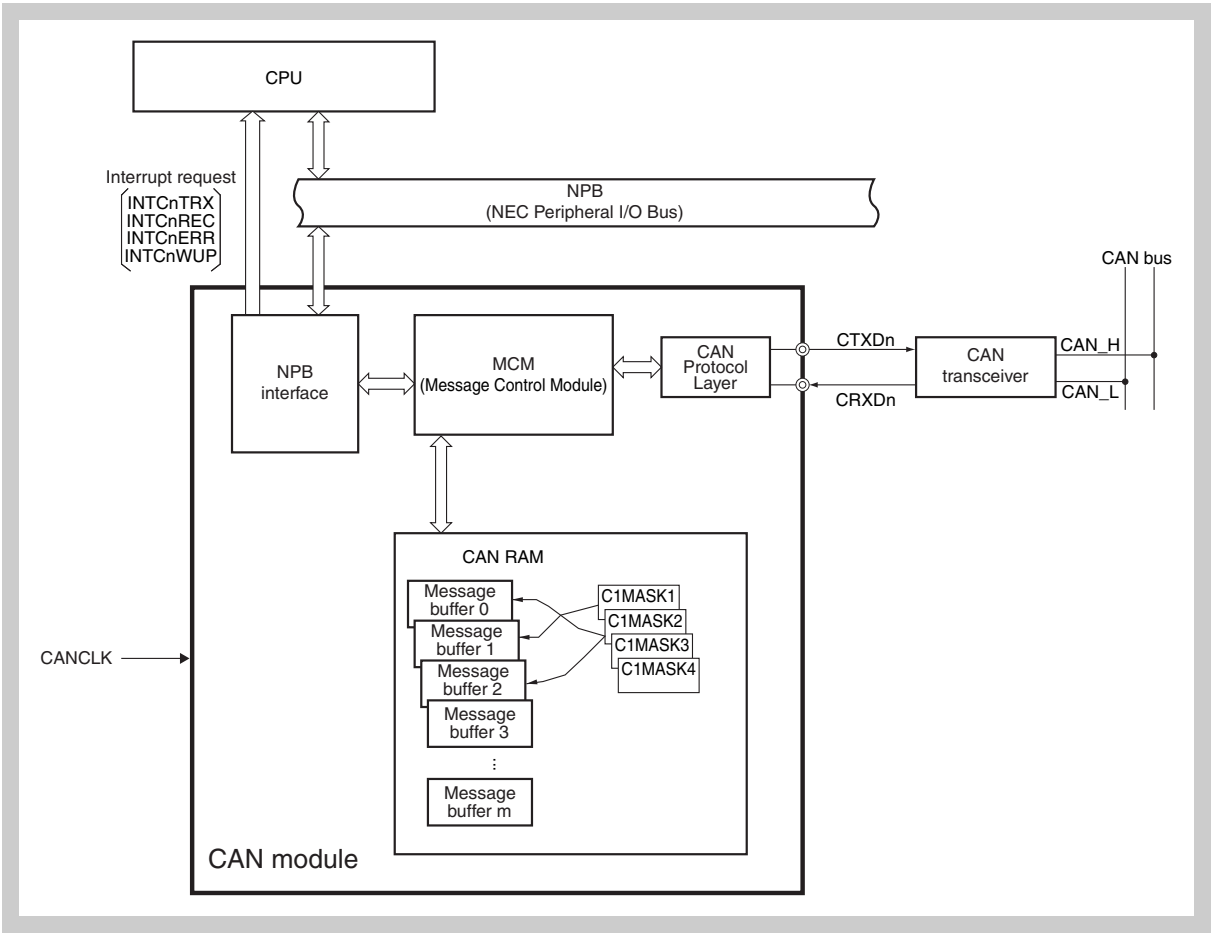


Figure 18-1 Block diagram of CAN module

18.2 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.

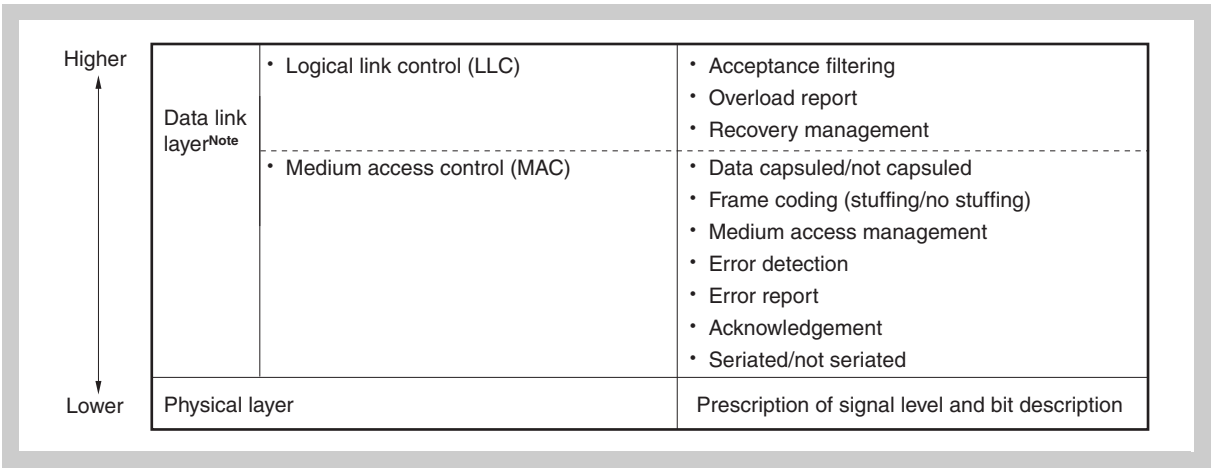


Figure 18-2 Composition of layers

Note CAN Controller specification

18.2.1 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.

18.2.2 Frame types

The following four types of frames are used in the CAN protocol.

Table 18-2 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.

18.2.3 Data frame and remote frame

(1) Data frame

A data frame is composed of seven fields.

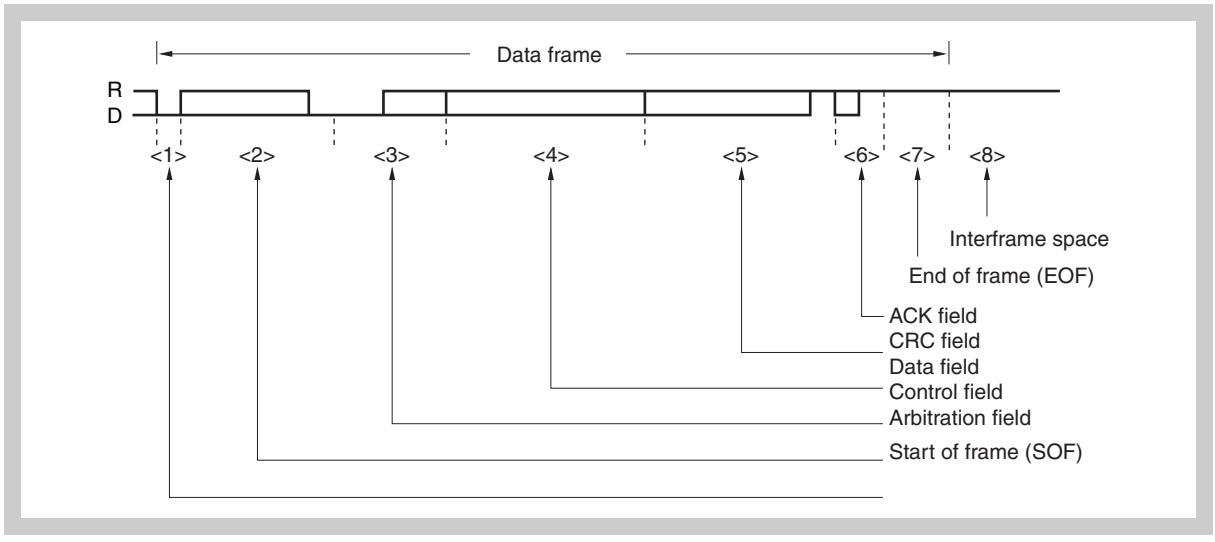


Figure 18-3 Data frame

Note D: Dominant = 0
R: Recessive = 1

(2) Remote frame

A remote frame is composed of six fields.

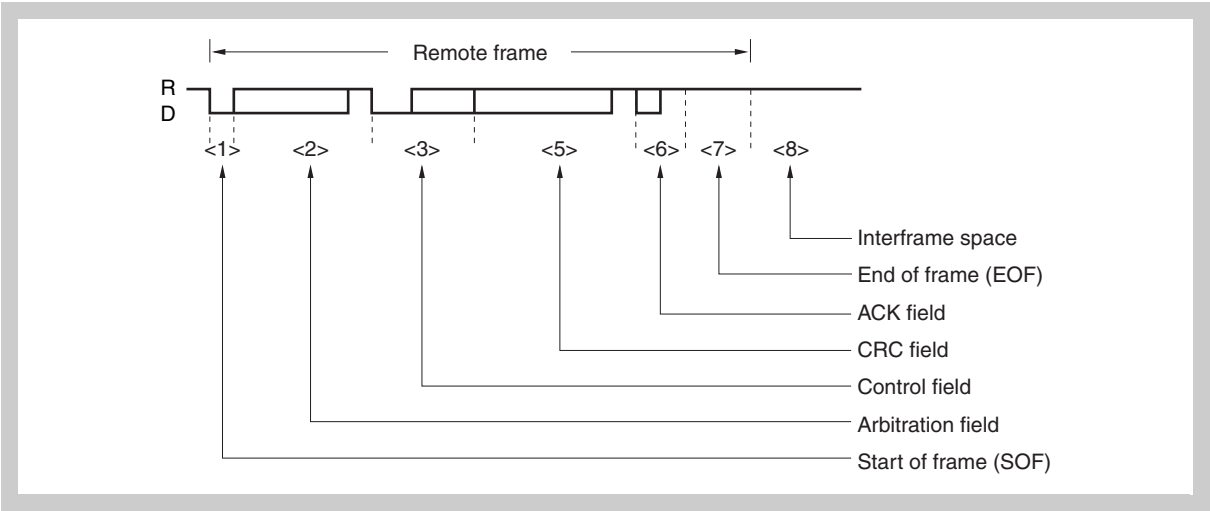


Figure 18-4 Remote frame

Note 1. The data field is not transferred even if the control field's data length code is not "0000_B".

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.

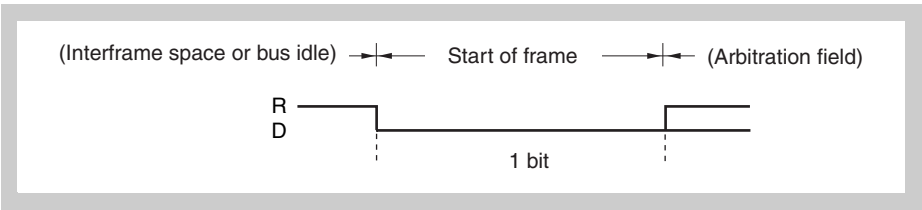


Figure 18-5 Start of frame (SOF)

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.

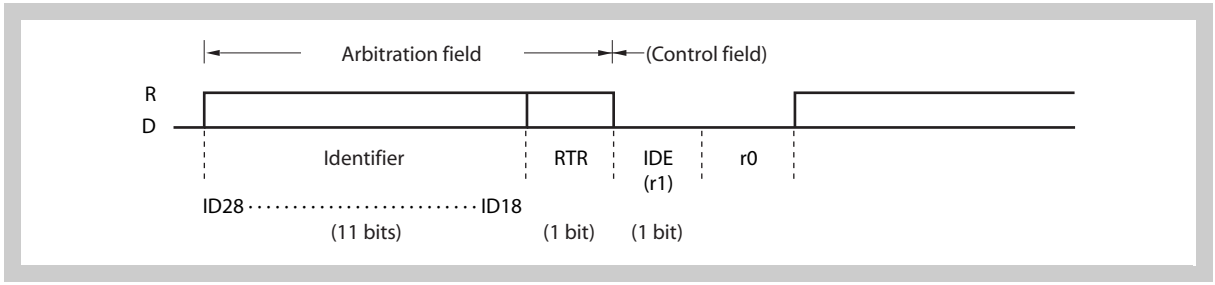


Figure 18-6 Arbitration field (in standard format mode)

Caution
1. ID28 to ID18 are identifiers.
2. An identifier is transmitted MSB first.

Note D: Dominant = 0
R: Recessive = 1

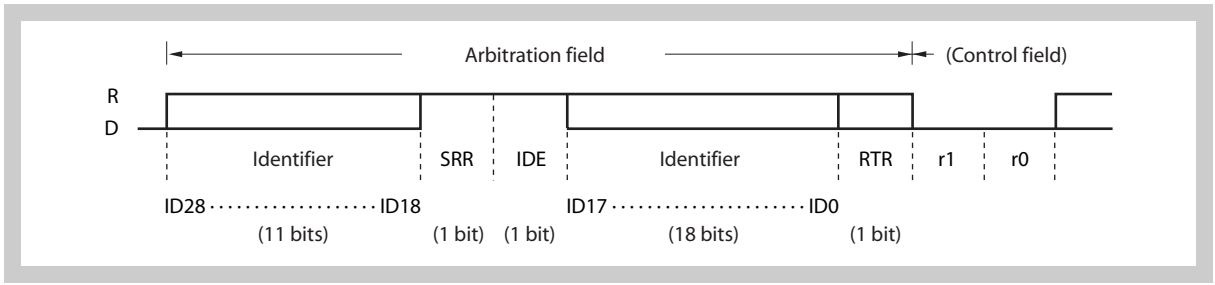


Figure 18-7 Arbitration field (in extended format mode)

Caution
1. ID28 to ID18 are identifiers.
2. An identifier is transmitted MSB first.

Note D: Dominant = 0
R: Recessive = 1

Table 18-3 RTR frame settings

Frame type	RTR bit
Data frame	0 (D)
Remote frame	1 (R)

Table 18-4 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).

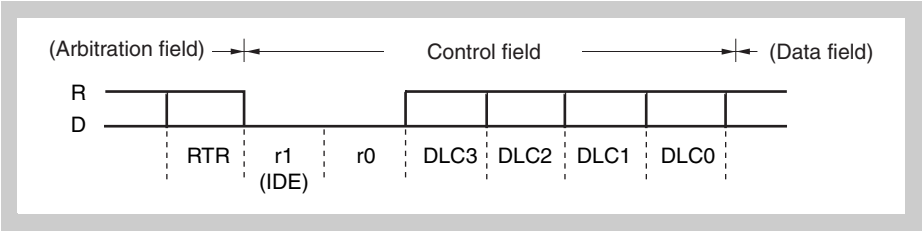


Figure 18-8 Control field

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 18-5 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_B.

(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.

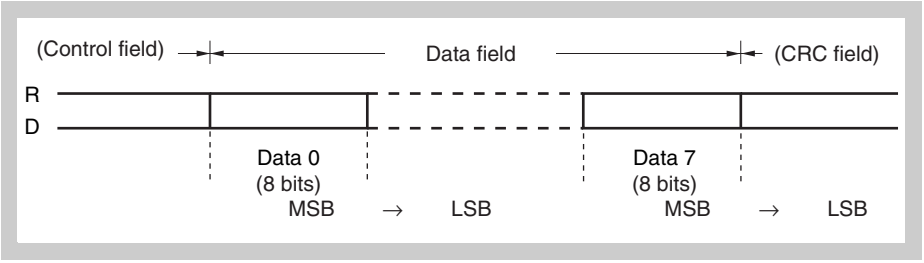


Figure 18-9 Data field

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.

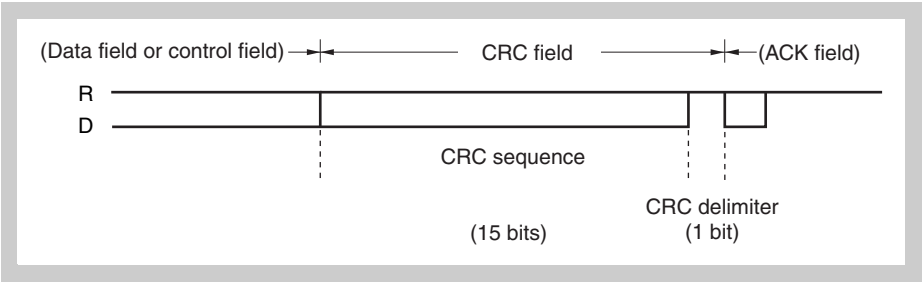


Figure 18-10 CRC field

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.

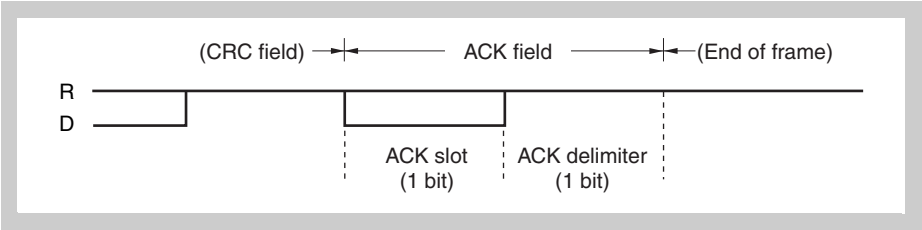


Figure 18-11 ACK field

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.

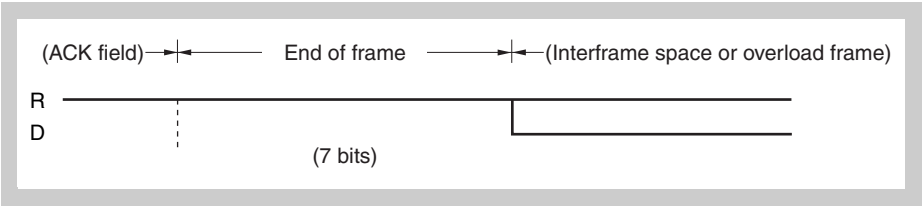


Figure 18-12 End of frame (EOF)

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.

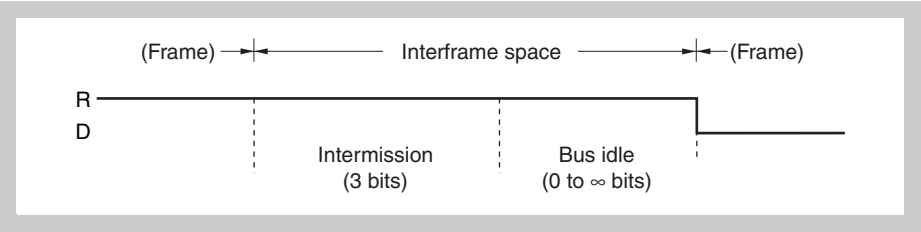


Figure 18-13 Interframe space (error active node)

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.

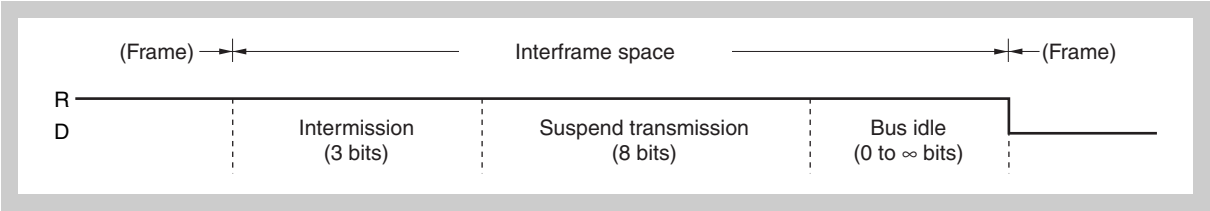


Figure 18-14 Interframe space (error passive node)

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 18-6 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.

18.2.4 Error frame

An error frame is output by a node that has detected an error.

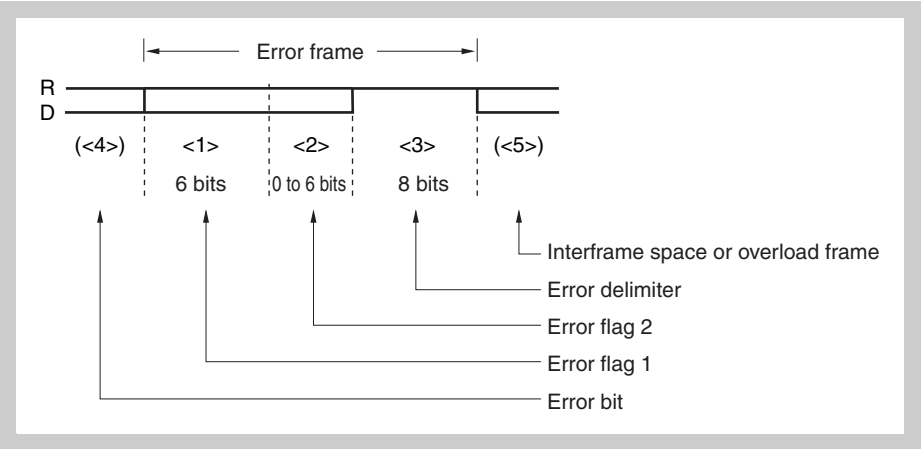


Figure 18-15 Error frame

Note D: Dominant = 0
R: Recessive = 1

Table 18-7 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.

18.2.5 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.

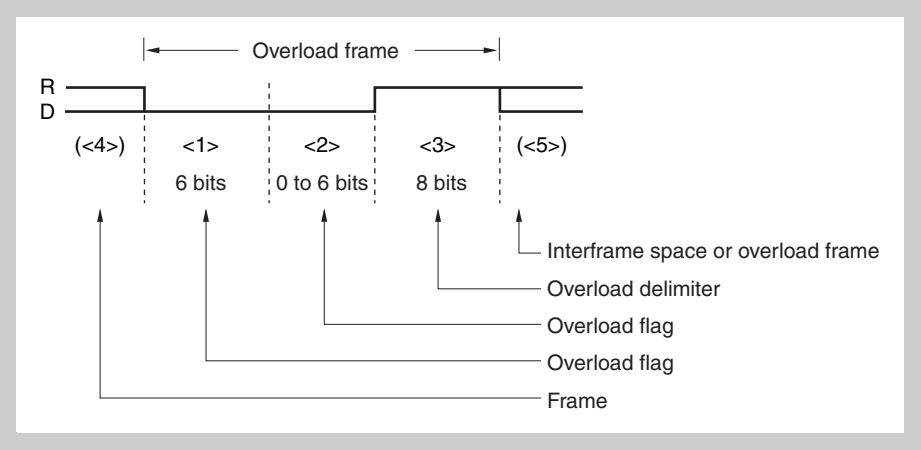


Figure 18-16 Overload frame

Note D: Dominant = 0
R: Recessive = 1

Table 18-8 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.

18.3 Functions

18.3.1 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 18-9 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.

18.3.2 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 18-10 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.

18.3.3 Multi masters

As the bus priority (a node acquiring transmit functions) is determined by the identifier, any node can be the bus master.

18.3.4 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.

18.3.5 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).

18.3.6 Error control function

(1) Error types

Table 18-11 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 18-12 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 18-13*.

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 18-13 Types of error states

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)Note	BOFF = 1, TECS1, TECS0 = 11	Communication is not possible. Messages are not stored when receiving frames, however, the following operations of <1>, <2>, and <3> are done. <1> TSOUT toggles. <2> REC is incremented/decremented. <3> VALID bit is set. If the CAN module is entered to the initialization mode and then transition request to any operation mode is made, and when 11 consecutive recessive-level bits are detected 128 times, the error counter is reset to 0 and the error active state can be restored.

Note The value of the transmission error counter (TEC) is invalid when the BOFF bit is set to 1. If an error that increments the value of the transmission error counter by +8 while the counter value is in a range of 248 to 255, the counter is not incremented and the bus-off state is assumed.

(b) Error counter

The error counter counts up when an error has occurred, and counts down upon successful transmission and reception. The error counter is updated immediately after error detection.

Table 18-14 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 (±0 if error counter = 0)	–1	No change
When the receiving node has completed reception without error	No change	• –1 (1 ≤ REC6 to REC0 ≤ 127, when REPS = 0) • ±0 (REC6 to REC0 = 0, when REPS = 0) • Value of 119 to 127 is set (when REPS = 1)

(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

(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 18-17 on page 727*). This request will be immediately acknowledged, and the OPMODE bits of the CnCTRL. register are cleared to 000_B. 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 18-17 on page 727*). 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 18-17 on page 727*), 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.

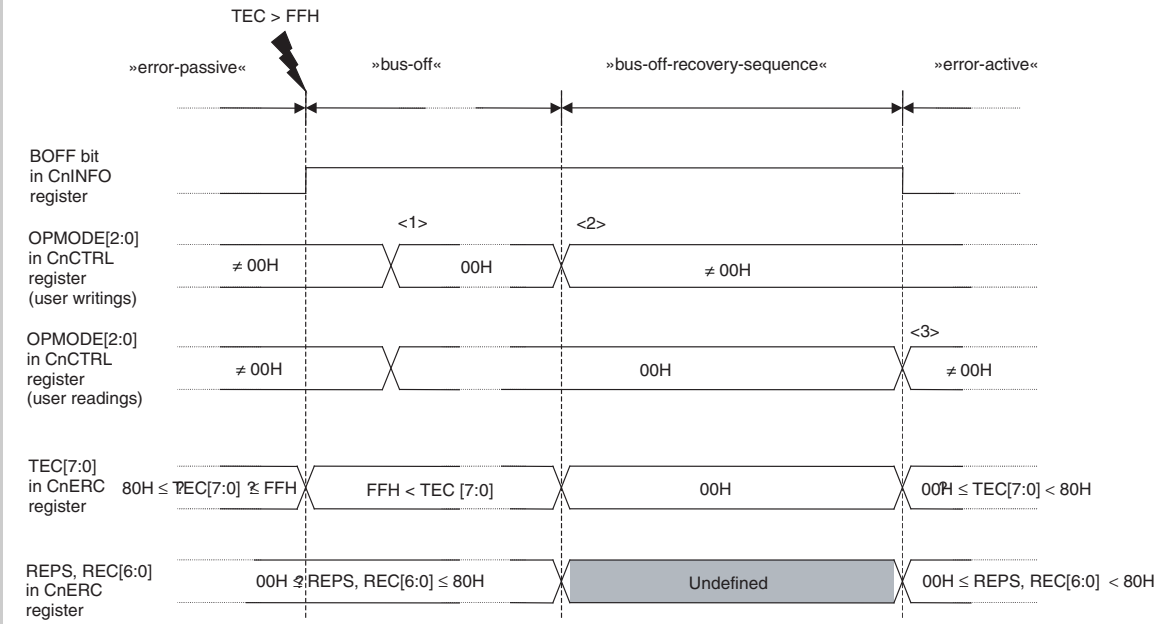


Figure 18-17 Recovery from bus-off state through normal recovery sequence

(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 726*.

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 18-55 on page 838*.

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.

18.3.7 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 760*).

(2) Data bit time (8 to 25 time quanta)

One data bit time is defined as shown in *Figure 18-18 on page 729*.

The CAN Controller sets time segment 1, time segment 2, and reSynchronization Jump Width (SJW) of data bit time, as shown in *Figure 18-18*. 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.

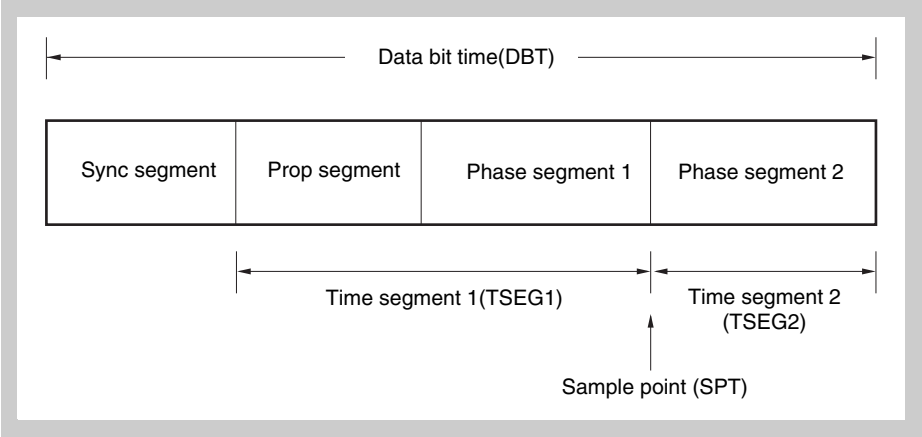


Figure 18-18 Segment setting

Table 18-15 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 18-19*.

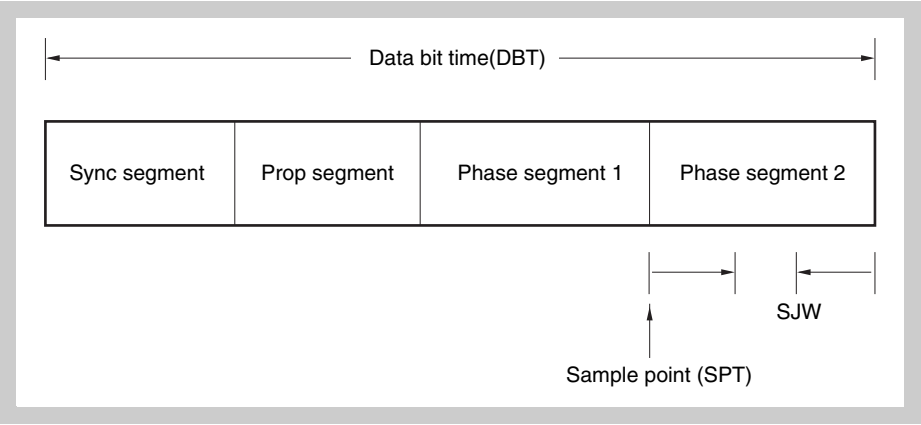


Figure 18-19 Configuration of data bit time defined by CAN specification

Table 18-16 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. The length of this segment is set so that ACK is returned before the start of phase segment 1. Time of prop segment ≥ (Delay of output buffer) + 2 × (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 1	Programmable to 1 to 8	
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.

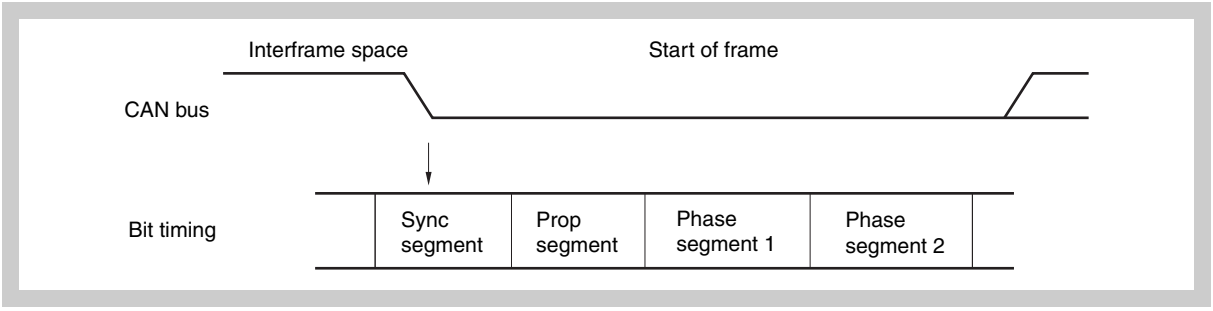


Figure 18-20 Adjusting synchronization of data bit

(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.

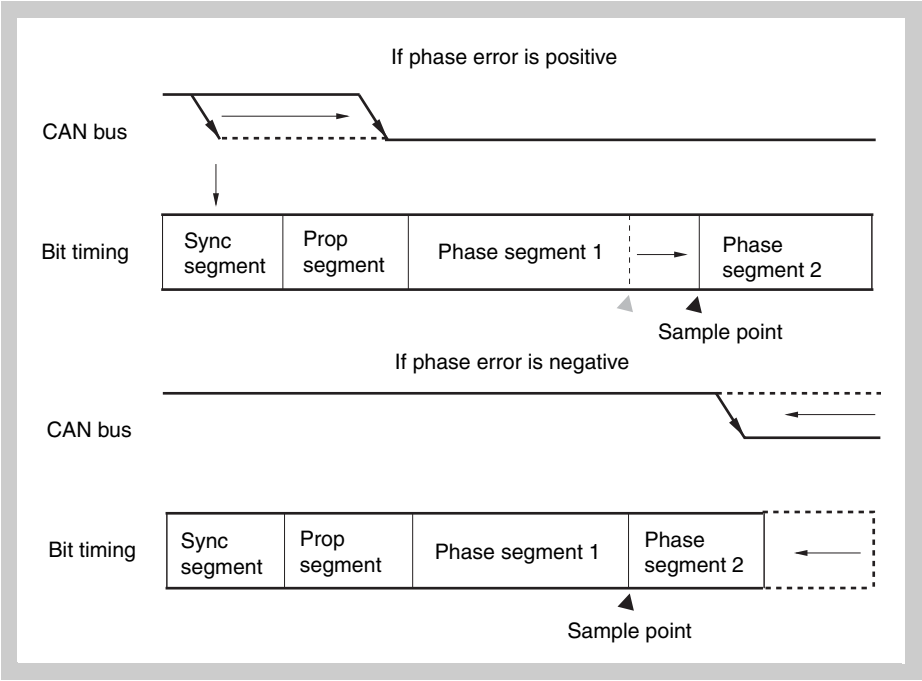


Figure 18-21 Resynchronization

18.4 Connection with Target System

The CAN module has to be connected to the CAN bus using an external transceiver.

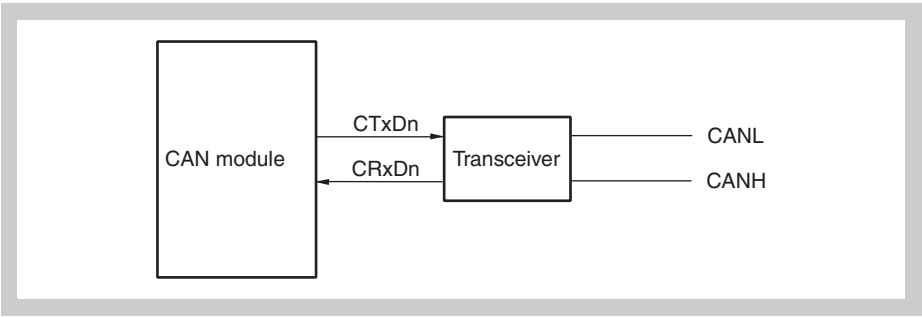


Figure 18-22 Connection to CAN bus

18.5 Internal Registers of CAN Controller

18.5.1 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 I/O area the bottom address is defined by the BPC register (refer to *"Programmable peripheral I/O area" on page 91*).

The addresses given in the following tables are offsets to the programmable peripheral I/O area base address PBA.

For the BPC setting $87FF_H$ is recommended. This setting defines the programmable peripheral I/O area base address

PBA = $01FF\ C000_H$

Table 18-17 lists all base addresses used throughout this chapter.

Table 18-17 CAN module base addresses

Base address name	Base address of	Address	Address for BPC =$B7FF_H$
C0RBaseAddr	CAN0 registers	PBA + 000_H	$01FF\ C000_H$
C0MBaseAddr	CAN0 message buffers	PBA + 100_H	$01FF\ C100_H$
C1RBaseAddr	CAN1 registers	PBA + 600_H	$01FF\ C600_H$
C1MBaseAddr	CAN1 message buffers	PBA + 700_H	$01FF\ C700_H$

In the following <CnRBaseAddr> respectively <CnMBaseAddr> are used for the base address names for CAN channel n.

18.5.2 CAN Controller configuration

Table 18-18 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)

18.5.3 CAN registers overview

(1) CANn global and module registers

The following table lists the address offsets to the CANn register base address: CnRBaseAddr.

Table 18-19 CANn global and module registers

Address offset	Register name	Symbol	R/W	Access			After reset
				1-bit	8-bit	16-bit	
000_H	CANn global control register	CnGMCTRL	R/W	–	–	√	0000_H
002_H	CANn global clock selection register	CnGMCS		–	√	–	$0F_H$
006_H	CANn global automatic block transmission register	CnGMABT		–	–	√	0000_H
008_H	CANn global automatic block transmission delay register	CnGMABTD		–	√	–	00_H
040_H	CANn module mask 1 register	CnMASK1L		–	–	√	Undefined
042_H		CnMASK1H		–	–	√	Undefined
044_H	CANn module mask 2 register	CnMASK2L		–	–	√	Undefined
046_H		CnMASK2H		–	–	√	Undefined
048_H	CANn module mask 3 register	CnMASK3L		–	–	√	Undefined
$04A_H$		CnMASK3H		–	–	√	Undefined
$04C_H$	CANn module mask 4 register	CnMASK4L		–	–	√	Undefined
$04E_H$		CnMASK4H		–	–	√	Undefined
050_H	CANn module control register	CnCTRL		–	–	√	0000_H
052_H	CANn module last error code register	CnLEC		–	√	–	00_H
053_H	CANn module information register	CnINFO	R	–	√	–	00_H
054_H	CANn module error counter register	CnERC		–	–	√	0000_H
056_H	CANn module interrupt enable register	CnIE	R/W	–	–	√	0000_H
058_H	CANn module interrupt status register	CnINTS		–	–	√	0000_H
$05A_H$	CANn module bit-rate prescaler register	CnBRP		–	√	–	FF_H
$05C_H$	CANn module bit-rate register	CnBTR		–	–	√	$370F_H$
$05E_H$	CANn module last in-pointer register	CnLIPT	R	–	√	–	Undefined
060_H	CANn module receive history list register	CnRGPT	R/W	–	–	√	$xx02_H$
062_H	CANn module last out-pointer register	CnLOPT	R	–	√	–	Undefined
064_H	CANn module transmit history list register	CnTGPT	R/W	–	–	√	$xx02_H$
066_H	CANn module time stamp register	CnTS		–	–	√	0000_H

(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 C0MDATA614 has the address E_H x 20_H + 6_H + C0MBaseAddr.

Note The message buffer register number m in the register symbols has 2 digits, for example, COMDATA01m = COMDATA0100 for m = 0.

Table 18-20 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 $xxxx_B$
$mx20_H + 9_H$	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_B

18.5.4 Register bit configuration

Table 18-21 CAN global register bit configuration

Address offset[a]	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_H	CnGMCTRL (W)	0	0	0	0	0	0	0	Clear GOM
01_H		0	0	0	0	0	0	Set EFSD	Set GOM
00_H	CnGMCTRL (R)	0	0	0	0	0	0	EFSD	GOM
01_H		MBON	0	0	0	0	0	0	0
02_H	CnGMCS	0	0	0	0	CCP3	CCP2	CCP1	CCP0
06_H	CnGMABT (W)	0	0	0	0	0	0	0	Clear ABTTRG
07_H		0	0	0	0	0	0	Set ABTCLR	Set ABTTRG
06_H	CnGMABT (R)	0	0	0	0	0	0	ABTCLR	ABTTRG
07_H		0	0	0	0	0	0	0	0
08_H	CnGMABTD	0	0	0	0	ABTD3	ABTD2	ABTD1	ABTD0

a) Base address: <CnRBaseAddr>

Table 18-22 CAN module register bit configuration (1/2)

Address offset[a]	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_H	CnMASK1L	CMID7 to CMID0							
41_H		CMID15 to CMID8							
42_H	CnMASK1H	CMID23 to CMID16							
43_H		0	0	0	CMID28 to CMID24				
44_H	CnMASK2L	CMID7 to CMID0							
45_H		CMID15 to CMID8							
46_H	CnMASK2H	CMID23 to CMID16							
47_H		0	0	0	CMID28 to CMID24				
48_H	CnMASK3L	CMID7 to CMID0							
49_H		CMID15 to CMID8							
$4A_H$	CnMASK3H	CMID23 to CMID16							
$4B_H$		0	0	0	CMID28 to CMID24				
$4C_H$	CnMASK4L	CMID7 to CMID0							
$4D_H$		CMID15 to CMID8							
$4E_H$	CnMASK4H	CMID23 to CMID16							
$4F_H$		0	0	0	CMID28 to CMID24				
50_H	CnCTRL (W)	0	Clear AL	Clear VALID	Clear PSMODE1	Clear PSMODE0	Clear OPMODE2	Clear OPMODE1	Clear OPMODE0
51_H		Set CCERC	Set AL	0	Set PSMODE1	Set PSMODE0	Set OPMODE2	Set OPMODE1	Set OPMODE0
50_H	CnCTRL (R)	CCERC	AL	VALID	PS MODE1	PS MODE0	OP MODE2	OP MODE1	OP MODE0
51_H		0	0	0	0	0	0	RSTAT	TSTAT

Table 18-22 CAN module register bit configuration (2/2)

Address offset[a]	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_H	CnLEC (W)	0	0	0	0	0	0	0	0
52_H	CnLEC (R)	0	0	0	0	0	LEC2	LEC1	LEC0
53_H	CnINFO	0	0	0	BOFF	TECS1	TECS0	RECS1	RECS0
54_H	CnERC	TEC7 to TEC0							
55_H		REC7 to REC0							
56_H	CnIE (W)	0	0	Clear CIE5	Clear CIE4	Clear CIE3	Clear CIE2	Clear CIE1	Clear CIE0
57_H		0	0	Set CIE5	Set CIE4	Set CIE3	Set CIE2	Set CIE1	Set CIE0
56_H	CnIE (R)	0	0	CIE5	CIE4	CIE3	CIE2	CIE1	CIE0
57_H		0	0	0	0	0	0	0	0
58_H	CnINTS (W)	0	0	Clear CINTS5	Clear CINTS4	Clear CINTS3	Clear CINTS2	Clear CINTS1	Clear CINTS0
59_H		0	0	0	0	0	0	0	0
58_H	CnINTS (R)	0	0	CINTS5	CINTS4	CINTS3	CINTS2	CINTS1	CINTS0
59_H		0	0	0	0	0	0	0	0
$5A_H$	CnBRP	TQPRS7 to TQPRS0							
$5C_H$	CnBTR	0	0	0	0	TSEG13 to TSEG10			
$5D_H$		0	0	SJW1, SJW0		0	TSEG22 to TSEG20		
$5E_H$	CnLIPT	LIPT7 to LIPT0							
60_H	CnRGPT (W)	0	0	0	0	0	0	0	Clear ROVF
61_H		0	0	0	0	0	0	0	0
60_H	CnRGPT (R)	0	0	0	0	0	0	RHPM	ROVF
61_H		RGPT7 to RGPT0							
$F62_H$	CnLOPT	LOPT7 to LOPT0							
64_H	CnTGPT (W)	0	0	0	0	0	0	0	Clear TOVF
65_H		0	0	0	0	0	0	0	0
64_H	CnTGPT (R)	0	0	0	0	0	0	THPM	TOVF
65_H		TGPT7 to TGPT0							
66_H	CnTS (W)	0	0	0	0	0	Clear TSLOCK	Clear TSSEL	Clear TSEN
67_H		0	0	0	0	0	Set TSLOCK	Set TSSEL	Set TSEN
66_H	CnTS (R)	0	0	0	0	0	TSLOCK	TSSEL	TSEN
67_H		0	0	0	0	0	0	0	0
68_H to FF_H	-	Access prohibited (reserved for future use)							

a) Base address: <CnRBaseAddr>

Table 18-23 Message buffer register bit configuration

Address offset[a]	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_H	CnMDATA01m	Message data (byte 0)							
1_H		Message data (byte 1)							
0_H	CnMDATA0m	Message data (byte 0)							
1_H	CnMDATA1m	Message data (byte 1)							
2_H	CnMDATA23m	Message data (byte 2)							
3_H		Message data (byte 3)							
2_H	CnMDATA2m	Message data (byte 2)							
3_H	CnMDATA3m	Message data (byte 3)							
4H	CnMDATA45m	Message data (byte 4)							
5_H		Message data (byte 5)							
4_H	CnMDATA4m	Message data (byte 4)							
5_H	CnMDATA5m	Message data (byte 5)							
6_H	CnMDATA67m	Message data (byte 6)							
7_H		Message data (byte 7)							
6_H	CnMDATA6m	Message data (byte 6)							
7_H	CnMDATA7m	Message data (byte 7)							
8_H	CnMDLCm	0				MDLC3	MDLC2	MDLC1	MDLC0
9_H	CnMCONFm	OWS	RTR	MT2	MT1	MT0	0	0	MA0
A_H	CnMIDLm	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
B_H		ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
C_H	CnMIDHm	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16
D_H		IDE	0	0	ID28	ID27	ID26	ID25	ID24
E_H	CnMCTRLm (W)	0	0	0	Clear MOW	Clear IE	Clear DN	Clear TRQ	Clear RDY
F_H		0	0	0	0	Set IE	0	Set TRQ	Set RDY
E_H	CnMCTRLm (R)	0	0	0	MOW	IE	DN	TRQ	RDY
F_H		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 735*.

18.6 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 18-23* 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 18-26*). *Figure 18-23* shows how the values of set bits or clear bits relate to set/clear/no change operations in the corresponding register.

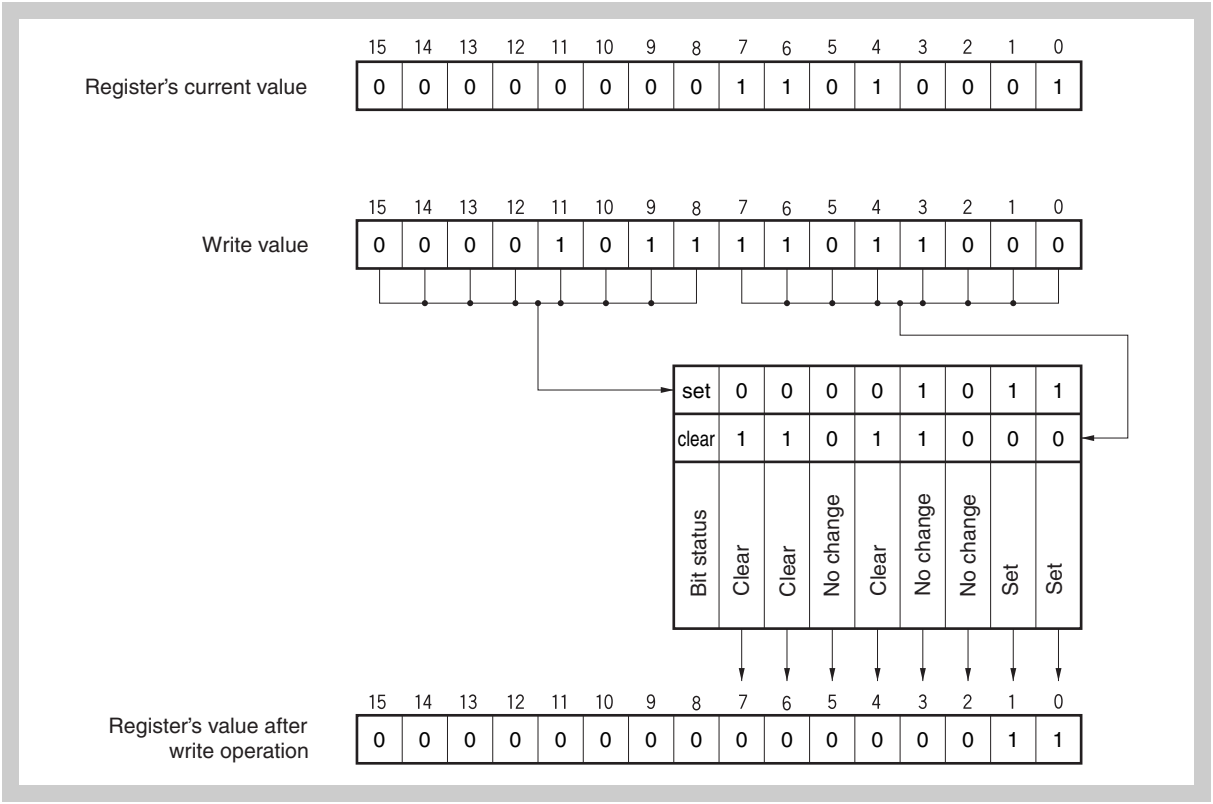


Figure 18-23 Example of bit setting/clearing operations

(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

18.7 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_H

Initial Value 0000_H. 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_H

Initial Value $0F_H$. 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_{CANMOD})
0	0	0	0	$f_{CAN}/1$
0	0	0	1	$f_{CAN}/2$
0	0	1	0	$f_{CAN}/3$
0	0	1	1	$f_{CAN}/4$
0	1	0	0	$f_{CAN}/5$
0	1	0	1	$f_{CAN}/6$
0	1	1	0	$f_{CAN}/7$
0	1	1	1	$f_{CAN}/8$
1	0	0	0	$f_{CAN}/9$
1	0	0	1	$f_{CAN}/10$
1	0	1	0	$f_{CAN}/11$
1	0	1	1	$f_{CAN}/12$
1	1	0	0	$f_{CAN}/13$
1	1	0	1	$f_{CAN}/14$
1	1	1	0	$f_{CAN}/15$
1	1	1	1	$f_{CAN}/16$ (default value)

Note f_{CAN} = 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_H

Initial Value 0000_H. 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_H) and confirm the CnGMABT register is surely initialized to the default value (0000_H).

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_H

Initial Value 00_H. 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[a]
0	0	0	0	0 DBT (default value)
0	0	0	1	2^5 DBT
0	0	1	0	2^6 DBT
0	0	1	1	2^7 DBT
0	1	0	0	2^8 DBT
0	1	0	1	2^9 DBT
0	1	1	0	2^{10} DBT
0	1	1	1	2^{11} DBT
1	0	0	0	2^{12} DBT
Other than above				Setting prohibited

a) 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_H
CnMASK1H: <CnRBaseAddr> + 042_H

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_H
CnMASK2H: <CnRBaseAddr> + 046_H

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_H
CnMASK3H: <CnRBaseAddr> + $04A_H$

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> + $04C_H$
CnMASK4H: <CnRBaseAddr> + $04E_H$

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_H

Initial Value 0000_H. 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
1. 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
2. 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
1. The TSTAT bit is set to 1 under the following conditions (timing)
   - The SOF bit of a transmit frame is detected
2. 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_H

Initial Value 00_H. 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_H 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_H

Initial Value 00_H. 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_H

Initial Value 0000_H. 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_B).

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_H

Initial Value 0000_H. 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_H

Initial Value 0000_H. 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[a]
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> + $05A_H$

Initial Value FF_H. 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)

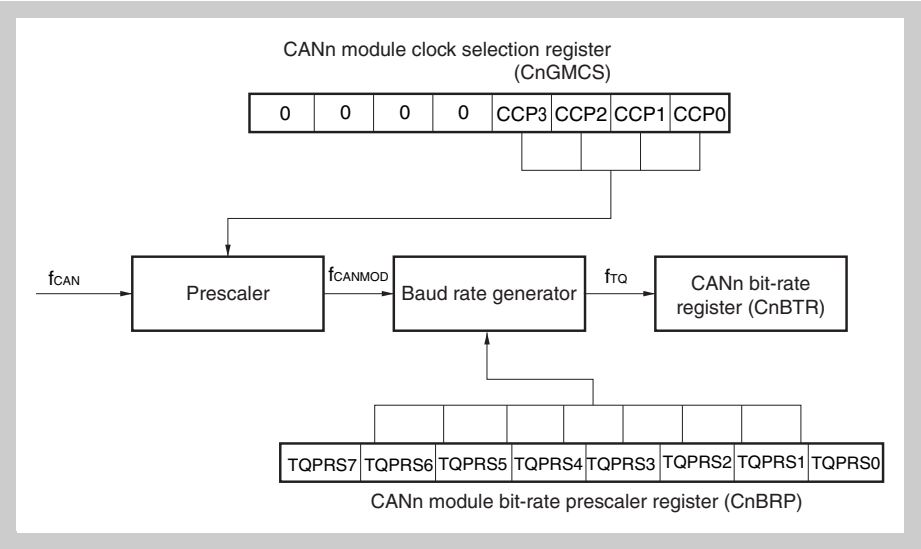


Figure 18-24 CAN module clock

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> + $05C_H$

Initial Value $370F_H$. 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

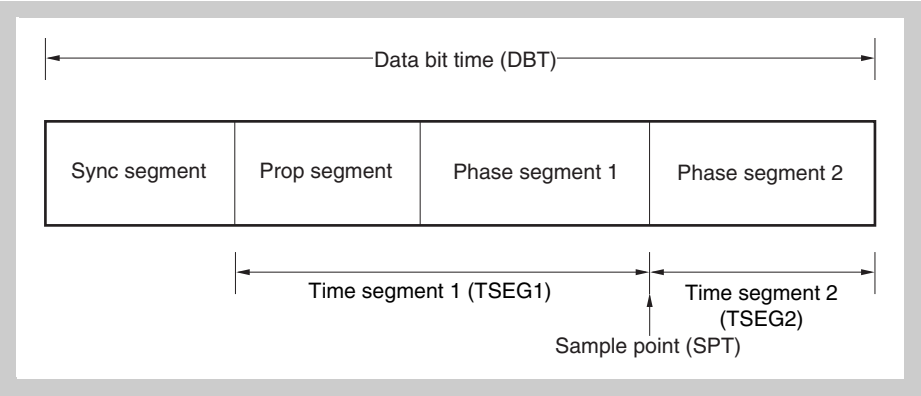


Figure 18-25 Data bit time

SJW1	SJW0	Length of synchronization jump width
0	0	$1T_Q$
0	1	$2T_Q$
1	0	$3T_Q$
1	1	$4T_Q$ (default value)

TSEG22	TSEG21	TSEG20	Length of time segment 2
0	0	0	$1T_Q$
0	0	1	$2T_Q$
0	1	0	$3T_Q$
0	1	1	$4T_Q$
1	0	0	$5T_Q$
1	0	1	$6T_Q$
1	1	0	$7T_Q$
1	1	1	$8T_Q$ (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 RHPM 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_H

Initial Value $xx02_H$. 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[a]	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[a]	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_H

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_H

Initial Value $xx02_H$. 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[a]	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[a]	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_H

Initial Value 0000_H. 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[a].

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 735*.

Initial Value Undefined.

CnMDATA01m

15	14	13	12	11	10	9	8
MDATA0115	MDATA0114	MDATA0113	MDATA0112	MDATA0111	MDATA0110	MDATA019	MDATA018
7	6	5	4	3	2	1	0
MDATA017	MDATA016	MDATA015	MDATA014	MDATA013	MDATA012	MDATA011	MDATA010

CnMDATA0m

7	6	5	4	3	2	1	0
MDATA07	MDATA06	MDATA05	MDATA04	MDATA03	MDATA02	MDATA01	MDATA00

CnMDATA1m

7	6	5	4	3	2	1	0
MDATA17	MDATA16	MDATA15	MDATA14	MDATA13	MDATA12	MDATA11	MDATA1

CnMDATA23m

15	14	13	12	11	10	9	8
MDATA2315	MDATA2314	MDATA2313	MDATA2312	MDATA2311	MDATA2310	MDATA239	MDATA238
7	6	5	4	3	2	1	0
MDATA237	MDATA236	MDATA235	MDATA234	MDATA233	MDATA232	MDATA231	MDATA230

CnMDATA2m

7	6	5	4	3	2	1	0
MDATA27	MDATA26	MDATA25	MDATA24	MDATA23	MDATA22	MDATA21	MDATA20

CnMDATA3m

7	6	5	4	3	2	1	0
MDATA37	MDATA36	MDATA35	MDATA34	MDATA33	MDATA32	MDATA31	MDATA30

CnMDATA45m

15	14	13	12	11	10	9	8
MDATA4515	MDATA4514	MDATA4513	MDATA4512	MDATA4511	MDATA4510	MDATA459	MDATA458
7	6	5	4	3	2	1	0
MDATA457	MDATA456	MDATA455	MDATA454	MDATA453	MDATA452	MDATA451	MDATA450

CnMDATA4m

7	6	5	4	3	2	1	0
MDATA47	MDATA46	MDATA45	MDATA44	MDATA43	MDATA42	MDATA41	MDATA40

CnMDATA5m

7	6	5	4	3	2	1	0
MDATA57	MDATA56	MDATA55	MDATA54	MDATA53	MDATA52	MDATA51	MDATA50

CnMDATA67m

15	14	13	12	11	10	9	8
MDATA6715	MDATA6714	MDATA6713	MDATA6712	MDATA6711	MDATA6710	MDATA679	MDATA678
7	6	5	4	3	2	1	0
MDATA677	MDATA676	MDATA675	MDATA674	MDATA673	MDATA672	MDATA671	MDATA670

CnMDATA6m

7	6	5	4	3	2	1	0
MDATA67	MDATA66	MDATA65	MDATA64	MDATA63	MDATA62	MDATA61	MDATA60

CnMDATA7m

7	6	5	4	3	2	1	0
MDATA77	MDATA76	MDATA75	MDATA74	MDATA73	MDATA72	MDATA71	MDATA70

(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 735.*

Initial Value $0000xxxx_B$. 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.)Note
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Note The data and DLC value actually transmitted to CAN bus are as follows.

Type of transmit frame	Length of transmit data	DLC transmitted
Data frame	Number of bytes specified by DLC (However, 8 bytes if DLC ≥ 8)	MDLC3 to MDLC0 bits
Remote frame	0 bytes	

Caution

1. Be sure to set bits 7 to 4 to 0000_B.
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 735.*

Initial Value Undefined.

7	6	5	4	3	2	1	0
OWS	RTR	MT2	MT1	MT0	0	0	MA0

OWS	Overwrite control bit
0	The message buffer that has already received a data frame[a] 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[a] 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 OWS 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[a]
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 735.*

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)[a]
1	Extended format mode (ID28 to ID0: 29 bits)

a) 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 735.*

Initial Value 00x0 0000 0000 0000_B. 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[a]	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[a]	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.

18.8 CAN Controller Initialization

18.8.1 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 817*.

18.8.2 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.

18.8.3 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 18-38*.

(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 796* and *"Transmission abort process except for ABT transmission in normal operation mode with automatic block transmission (ABT)" on page 796*). 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.

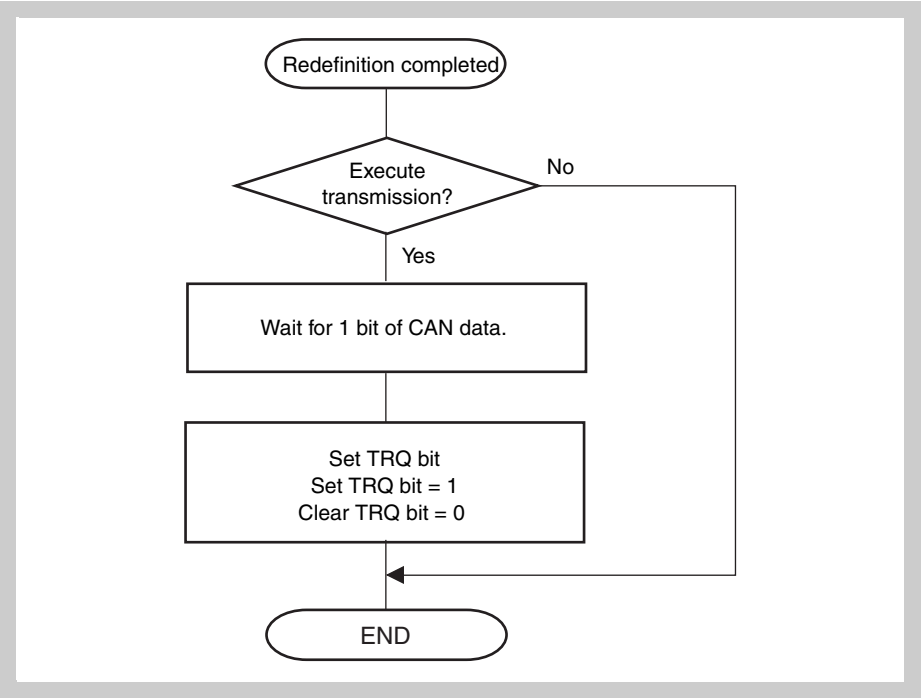


Figure 18-26 Setting transmission request (TRQ) to transmit message buffer after redefinition

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 18-38 on page 820* 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 18-26 on page 779* is not observed, a message with an ID not having the highest priority may be transmitted after redefinition.

18.8.4 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

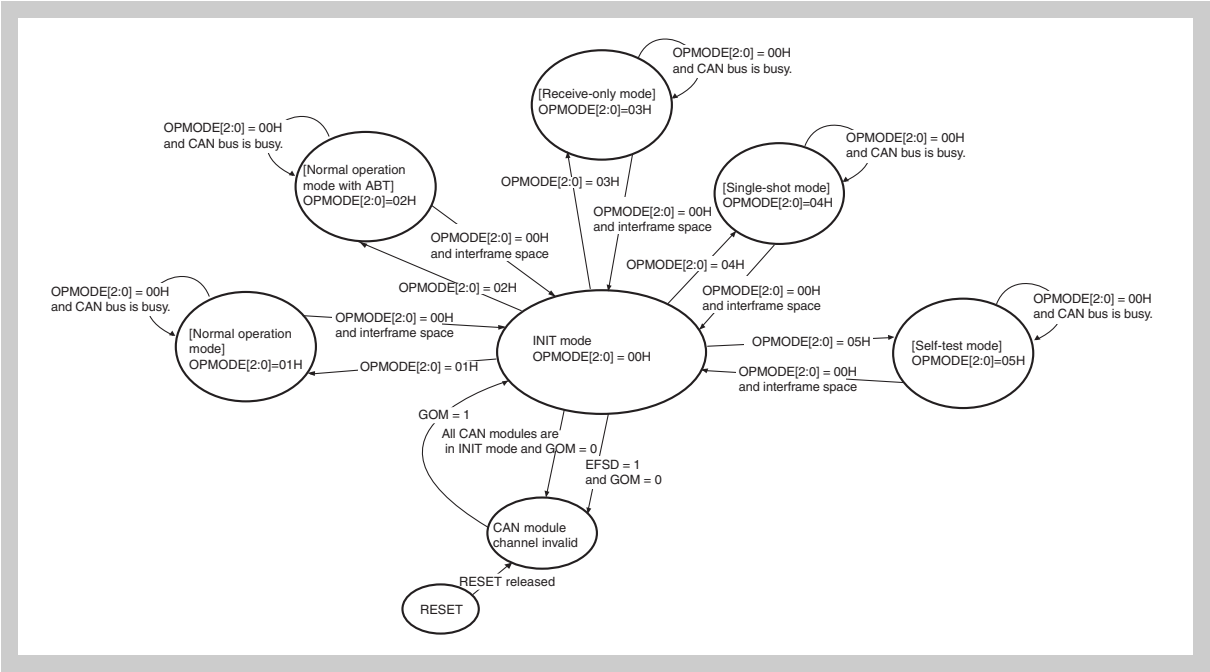


Figure 18-27 Transition to operation modes

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 $000_{B)}$. After issuing a request to change the mode to the initialization mode, read the OPMODE[2:0] bits until their value becomes 000_B to confirm that the module has entered the initialization mode (see *Figure 18-36 on page 818*).

18.8.5 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.

18.9 Message Reception

18.9.1 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_B, 010_B, 011_B, 100_B, or 101_B.)
- 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 18-24 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

18.9.2 Receive data read

To keep data consistency when reading CAN message buffers, perform the data reading according to *Figure 18-49 on page 831* to *Figure 18-52 on page 835*.

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 18-28 on page 783*.)

The receive history list is also updated just before the storgage 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.

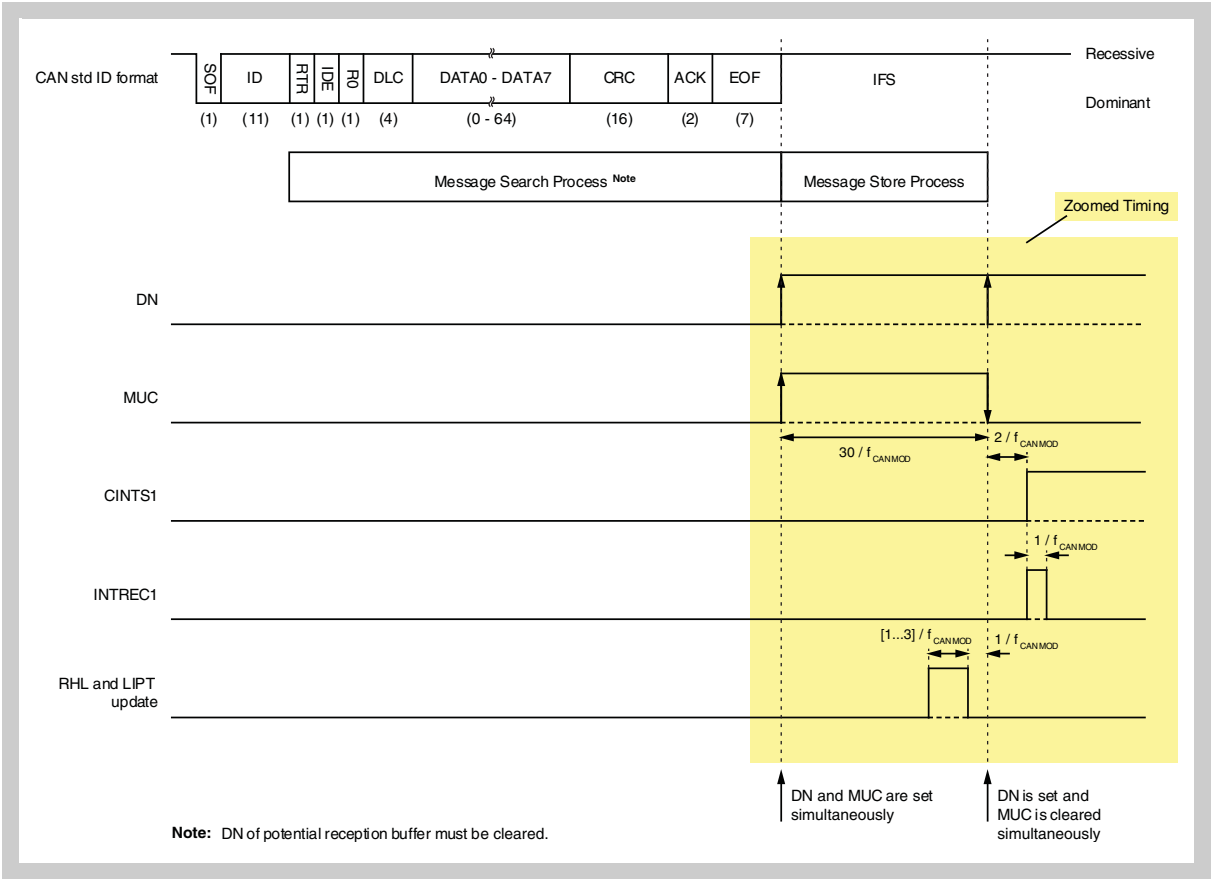


Figure 18-28 DN and MUC bit setting period (for standard ID format)

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.

18.9.3 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.

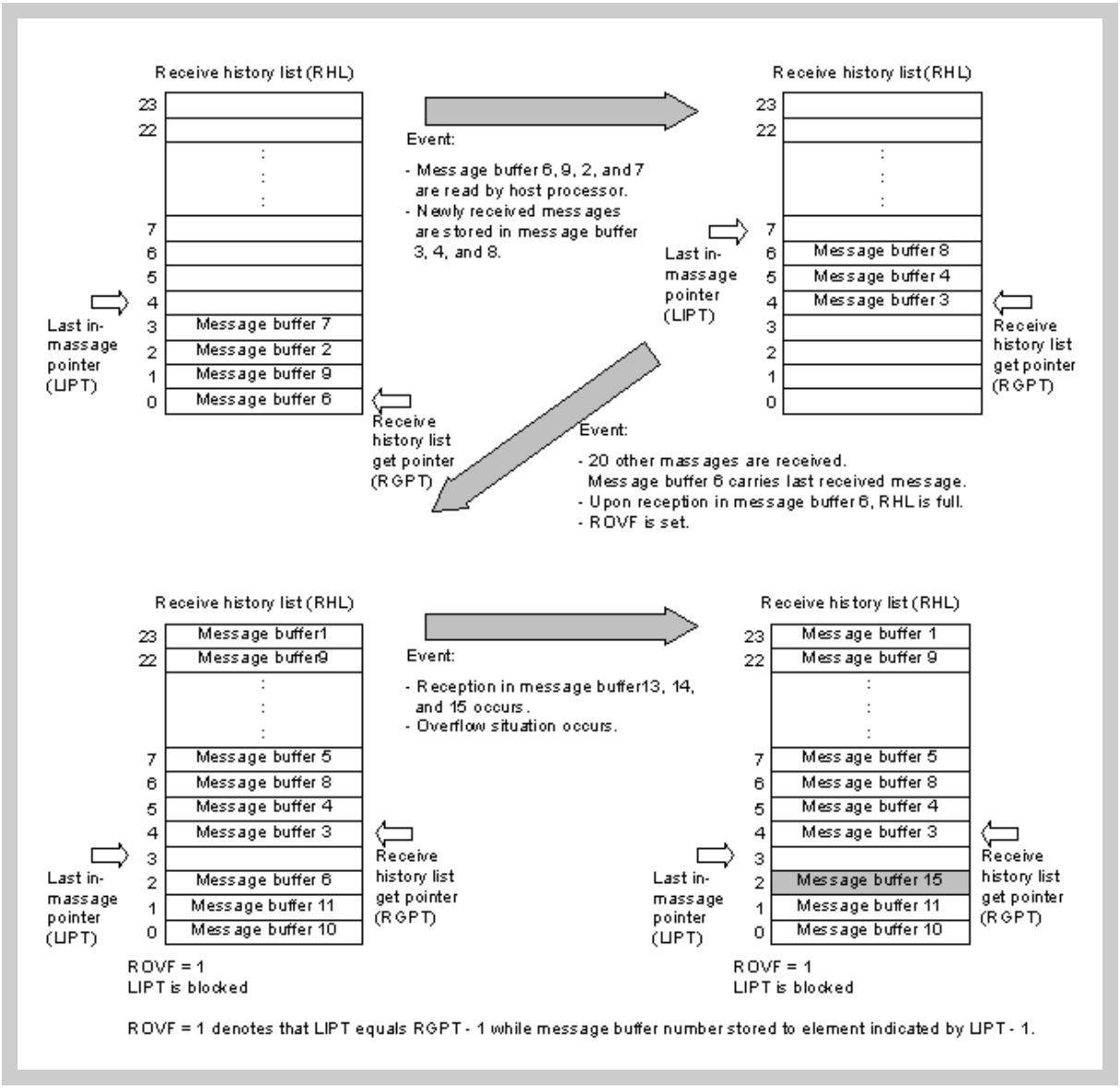


Figure 18-29 Receive history list

18.9.4 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_B).

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.

18.9.5 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 18-24*.

18.9.6 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_B)
- 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 CnMDLCm 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.

18.10 Message Transmission

18.10.1 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_B.)
- 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).

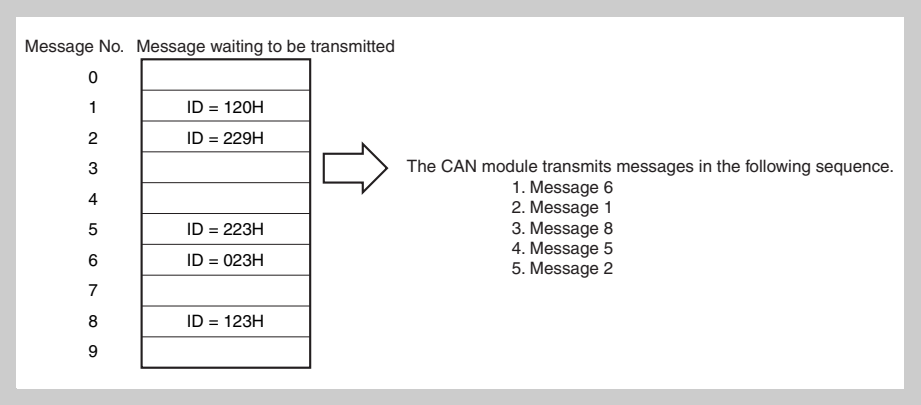


Figure 18-30 Message processing example

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.

18.10.2 Transmit history list function

The transmit history list (THL) function records in the transmit history list the number of the transmit message buffer from which data or remote frames have been were sent. The THL consists of storage elements equivalent to up to seven messages, the last out-message pointer (LOPT) with the corresponding 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).

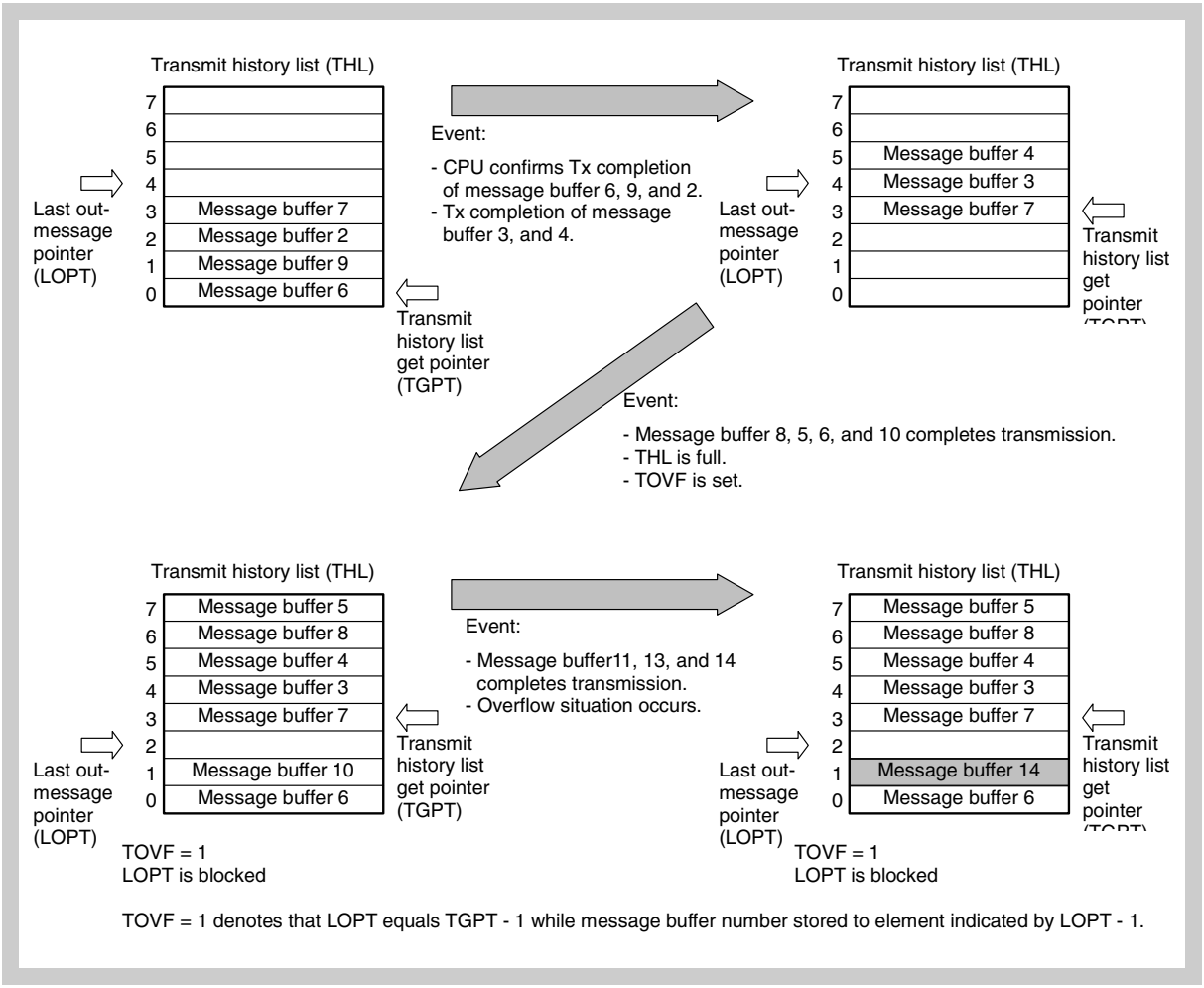


Figure 18-31 Transmit history list

18.10.3 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_B, “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_B. 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_H), 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_H.

18.10.4 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 18-45 on page 827*).

(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 18-46 on page 828*).

(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 18-47 on page 829*). 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 18-48 on page 830*).

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[a]	Same message buffer in the ABT area
Cleared (0)	Next message buffer in the ABT area[a]	Next message buffer in the ABT area[a]

a) 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.

18.10.5 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.

18.11 Power Saving Modes

18.11.1 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_B 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_B. When the module has entered the CAN sleep mode, the PSMODE[1:0] bits are set

to 01_B.

- 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_B 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 andPSMODE[1:0] will remain 01_B 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_B. 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_B, 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.

18.11.2 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_B 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_B 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_B, 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_B 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.

18.11.3 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_B). 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_B) 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_B).

18.12 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 18-25 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[a]	CnIE	INTCnTRX	Message frame successfully transmitted from message buffer m
2	CINTS1	CnINTS	CIE1[a]	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)

a) 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).

18.13 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.

18.13.1 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).

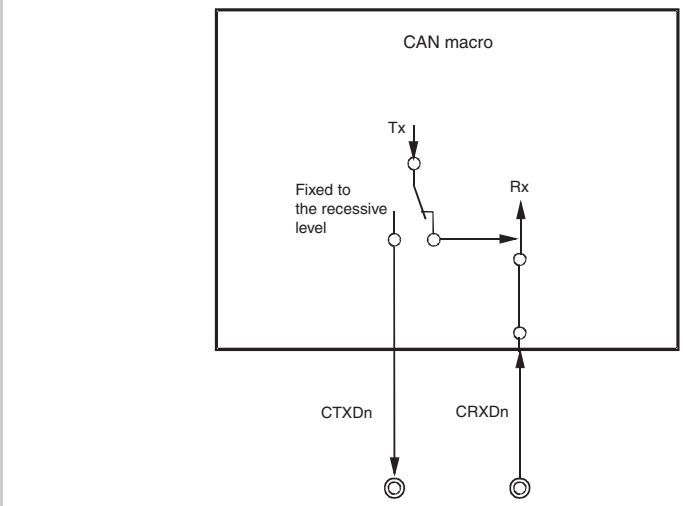


Figure 18-32 CAN module terminal connection in receive-only mode

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.

18.13.2 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.

18.13.3 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.

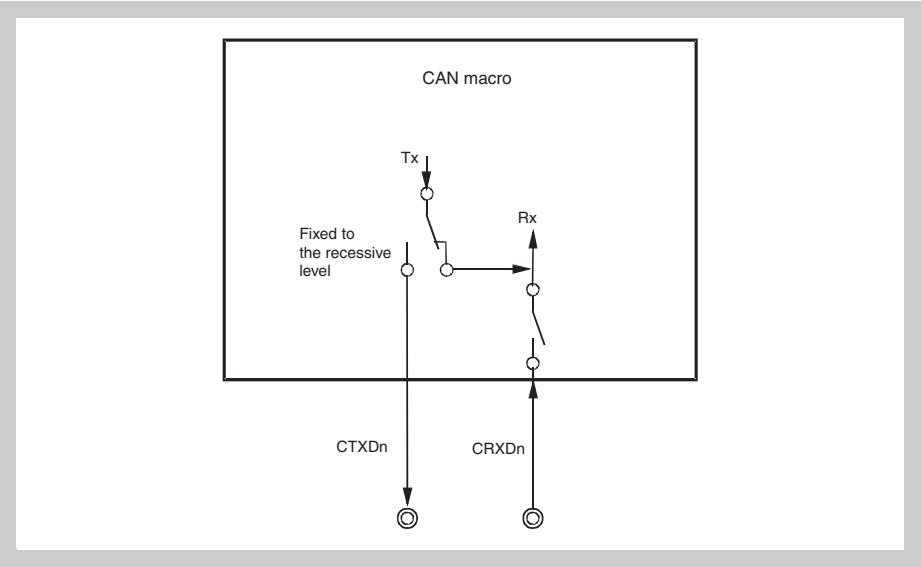


Figure 18-33 CAN module terminal connection in self-test mode

18.13.4 Receive/transmit operation in each operation mode

The following table shows outline of the receive/transmit operation in each operation mode.

Table 18-26 Outline of the receive/transmit in each operation mode

Operation mode	Transmission of data/remote frame	Transmission of ACK	Transmission of error/overload frame	Transmission retry	Automatic block transmission (ABT)	Set of VALID bit	Store data to message buffer
Initialization mode	No	No	No	No	No	No	No
Normal operation mode	Yes	Yes	Yes	Yes	No	Yes	Yes
Normal operation mode with ABT	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Receive only mode	No	No	No	No	No	Yes	Yes
Single-shot mode	Yes	Yes	Yes	No[a]	No	Yes	Yes
Self-test mode	Yes[b]	Yes[b]	Yes[b]	Yes[b]	No	Yes[b]	Yes[b]

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.

18.14 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.

18.14.1 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.

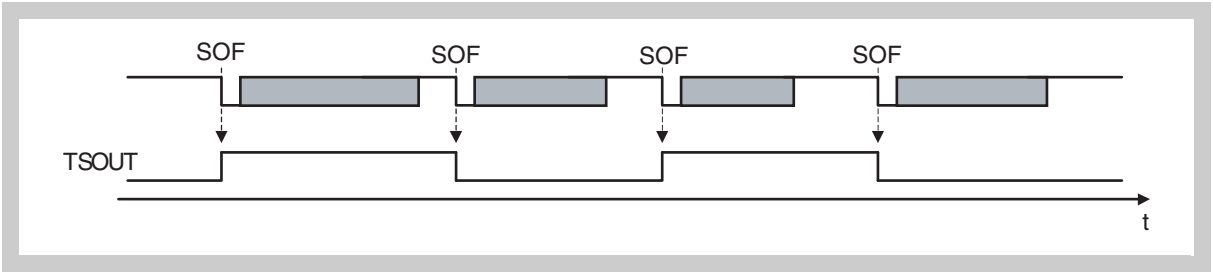


Figure 18-34 Timing diagram of capture signal TSOUT

The TSOUT signal toggles its level upon occurrence of the selected event during data frame reception (in *Figure 18-34*, 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.

18.15 Baud Rate Settings

18.15.1 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 ≤ SPT (sampling point) ≤ 17 TQ

  SPT = TSEG1 + 1
- 8 TQ ≤ DBT (data bit time) ≤ 25 TQ

  DBT = TSEG1 + TSEG2 + 1TQ = TSEG2 + SPT
- 1 TQ ≤ SJW (synchronization jump width) ≤ 4TQ

  SJW ≤ DBT – SPT
- 4 ≤ TSEG1 ≤ 16 [3 ≤ Setting value of TSEG1[3:0] ≤ 15]
- 1 ≤ TSEG2 ≤ 8 [0 ≤ Setting value of TSEG2[2:0] ≤ 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 18-27 shows the combinations of bit rates that satisfy the above conditions.

Table 18-27 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 18-27 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 18-27 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[a]	1	2	2	2	0011	001	71.4
7[a]	1	4	1	1	0100	000	85.7
6[a]	1	1	2	2	0010	001	66.7
6[a]	1	3	1	1	0011	000	83.3
5[a]	1	2	1	1	0010	000	80.0
4[a]	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_H.

Caution The values in *Table 18-27* 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.

18.15.2 Representative examples of baud rate settings

Table 18-28 and *Table 18-29* show representative examples of baud rate settings.

Table 18-28 Representative examples of baud rate settings (f_{CANMOD} = 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 SEGMENT1	PHASE SEGMENT2	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
100	5	00000100	16	1	9	3	3	1011	010	81.3

Table 18-28 Representative examples of baud rate settings (f_{CANMOD} = 8 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 SEGMENT1	PHASE SEGMENT2	TSEG1 [3:0]	TSEG2 [2:0]	
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 18-28* 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 18-29 Representative examples of baud rate settings (f_{CANMOD} = 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 SEGMENT1	PHASE SEGMENT2	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 18-29 Representative examples of baud rate settings (f_{CANMOD} = 16 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 SEGMENT1	PHASE SEGMENT2	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 18-29* 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.

18.16 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.

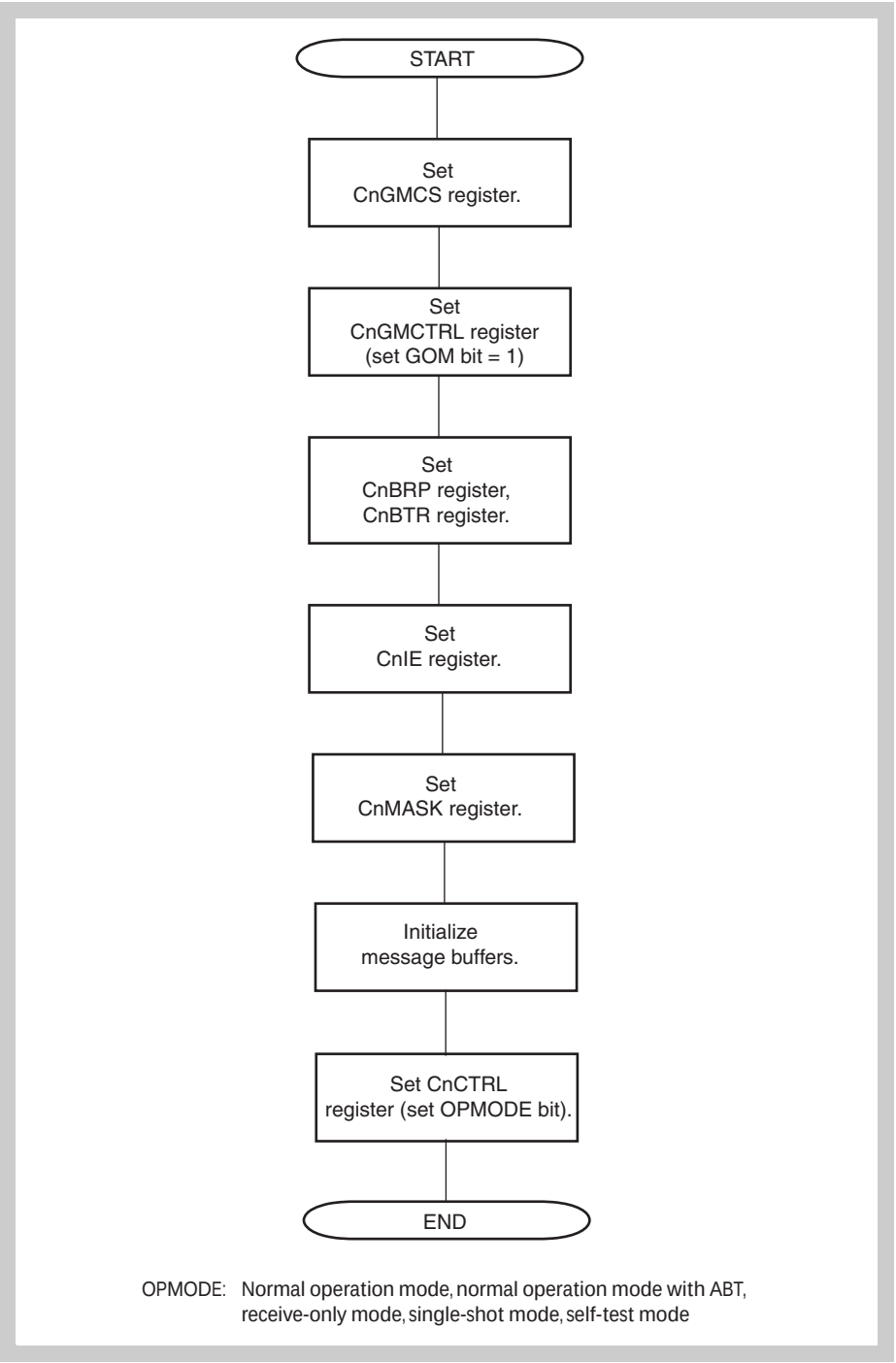


Figure 18-35 Initialization

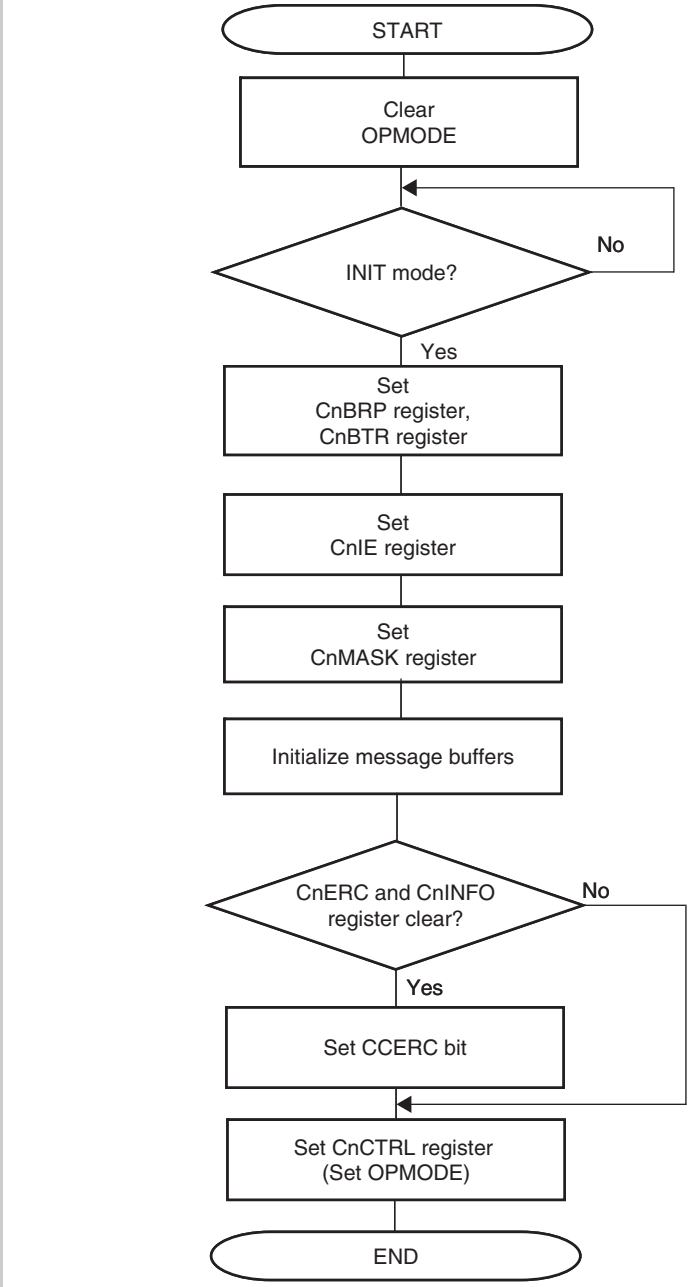


Figure 18-36 Re-initialization

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).

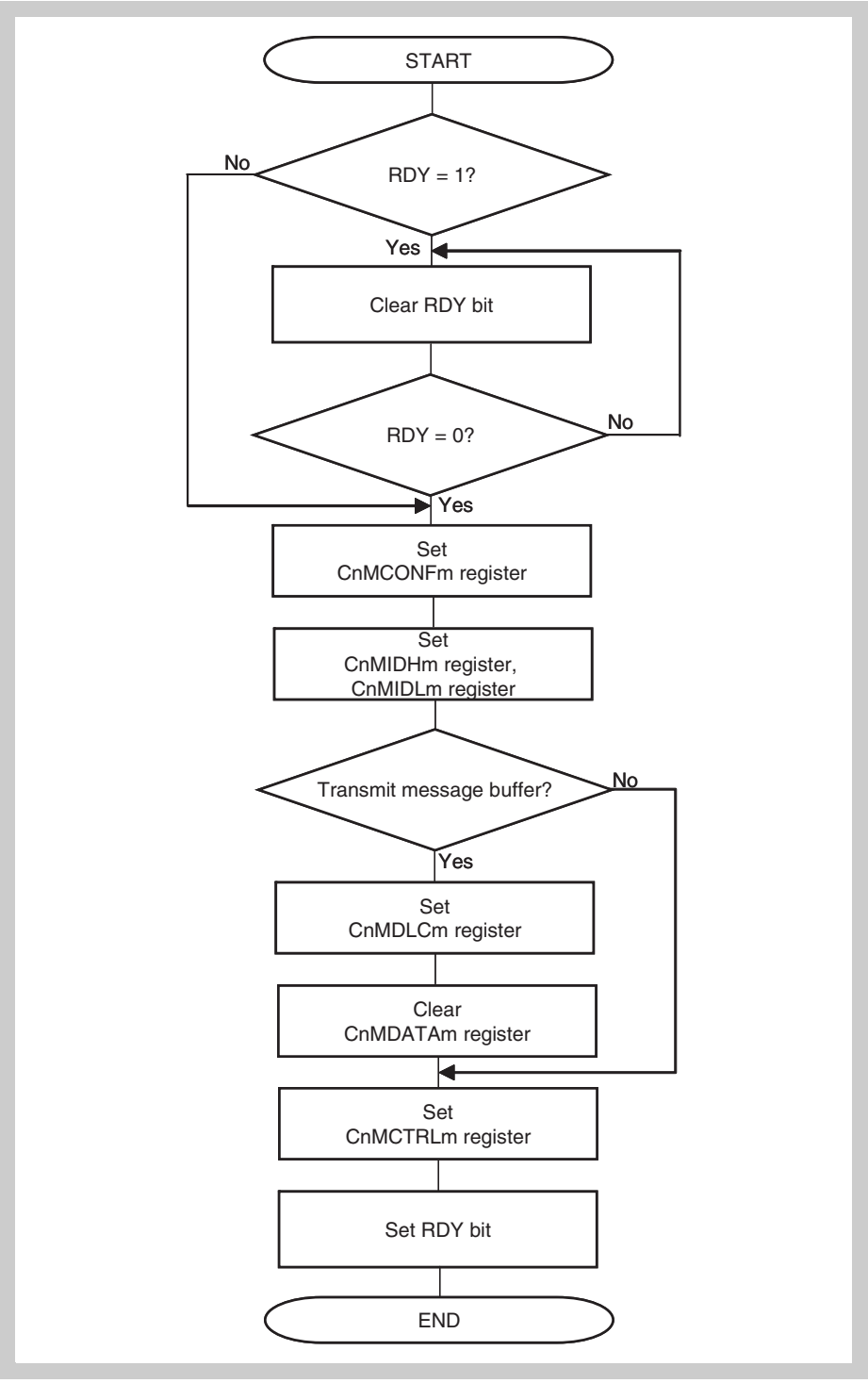


Figure 18-37 Message buffer initialization

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 18-38 shows the processing for a receive message buffer (MT[2:0] bits of CnMCONFm register = 001_B to 101_B).

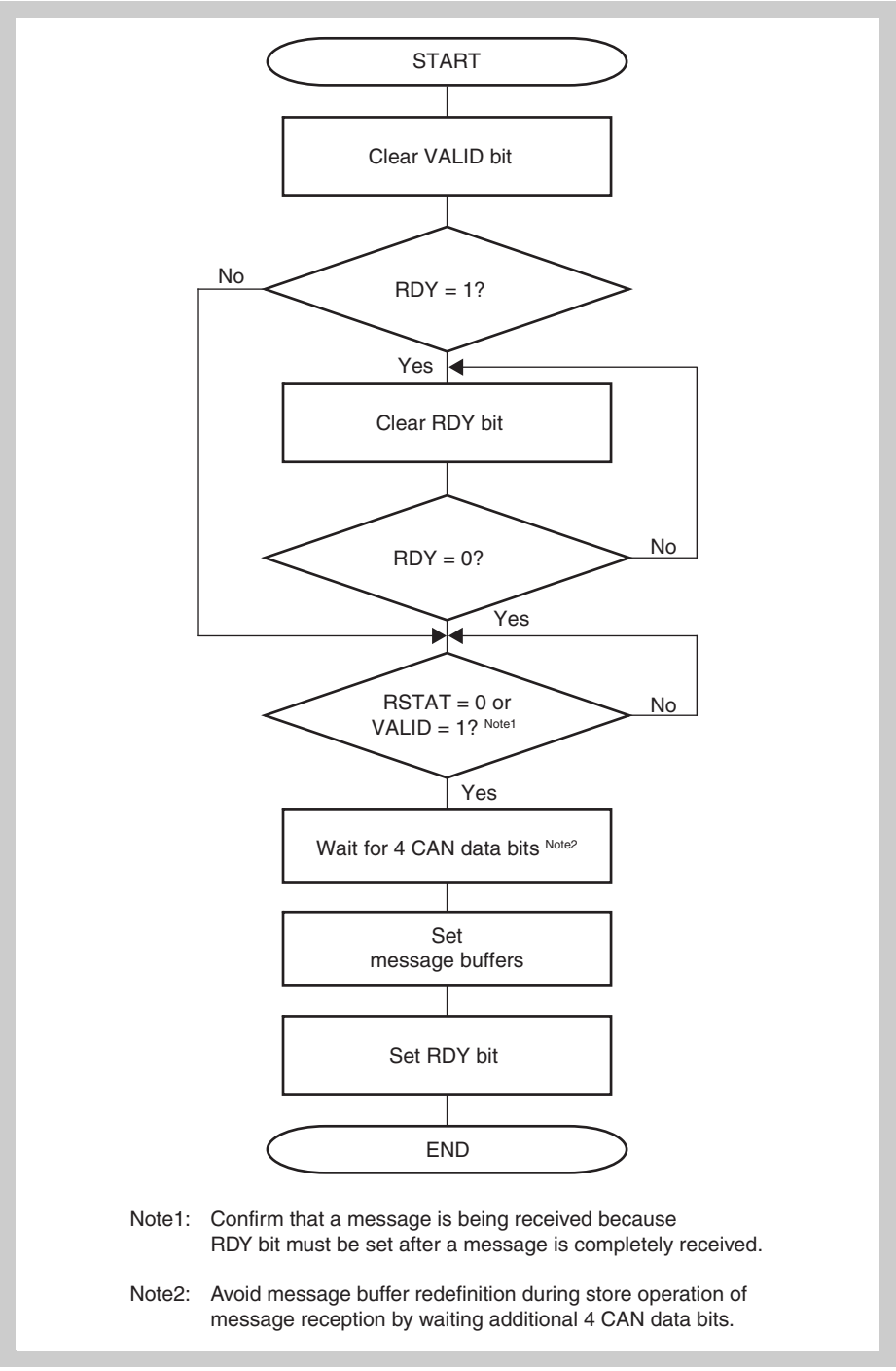


Figure 18-38 Message buffer redefinition

Figure 18-39 shows the processing for a transmit message buffer during transmission (MT[2:0] bits of CnMCONFm register = 000_B).

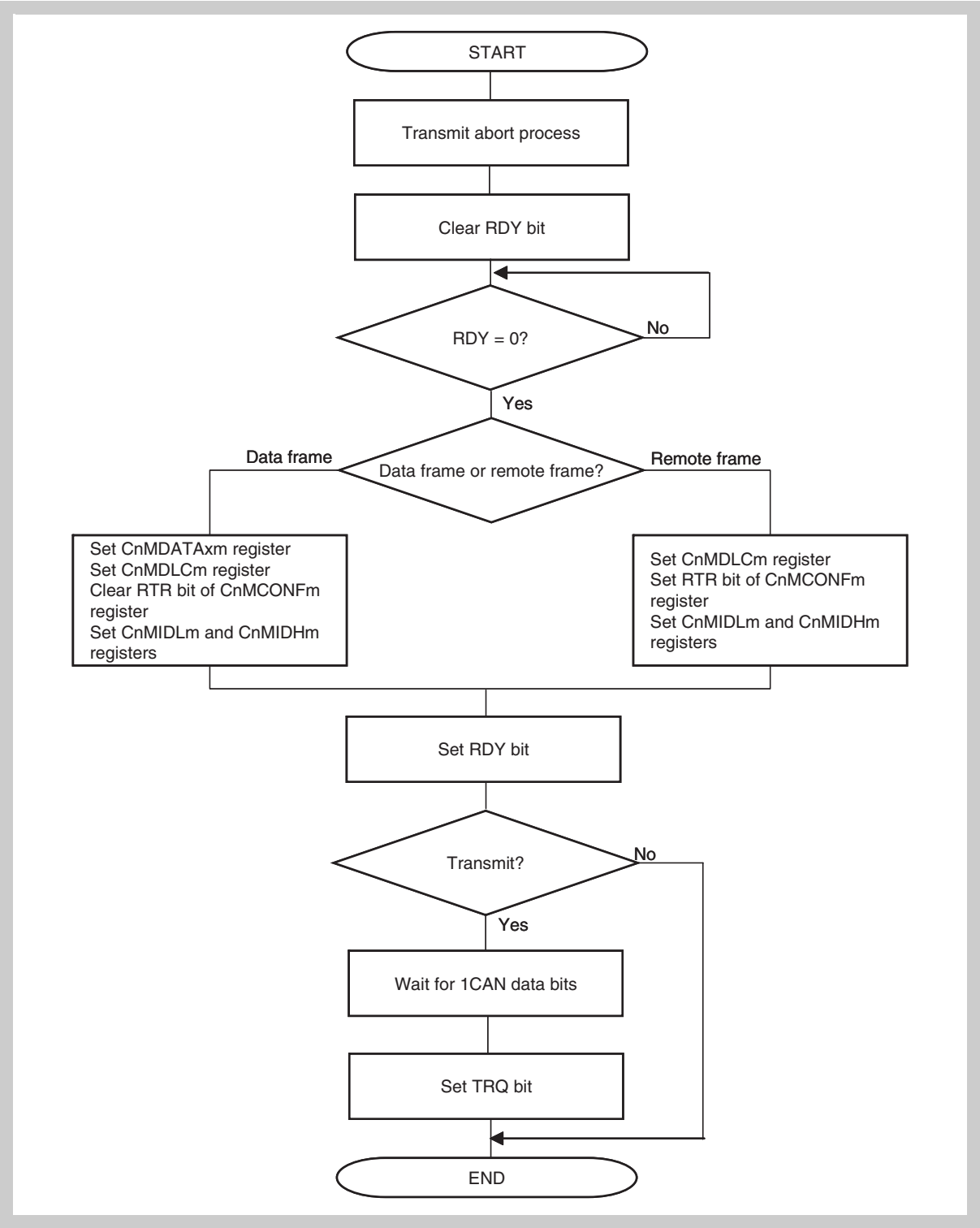


Figure 18-39 Message buffer redefinition during transmission

Figure 18-40 shows the processing for a transmit message buffer (MT[2:0] bits of CnMCONFm register = 000_B).

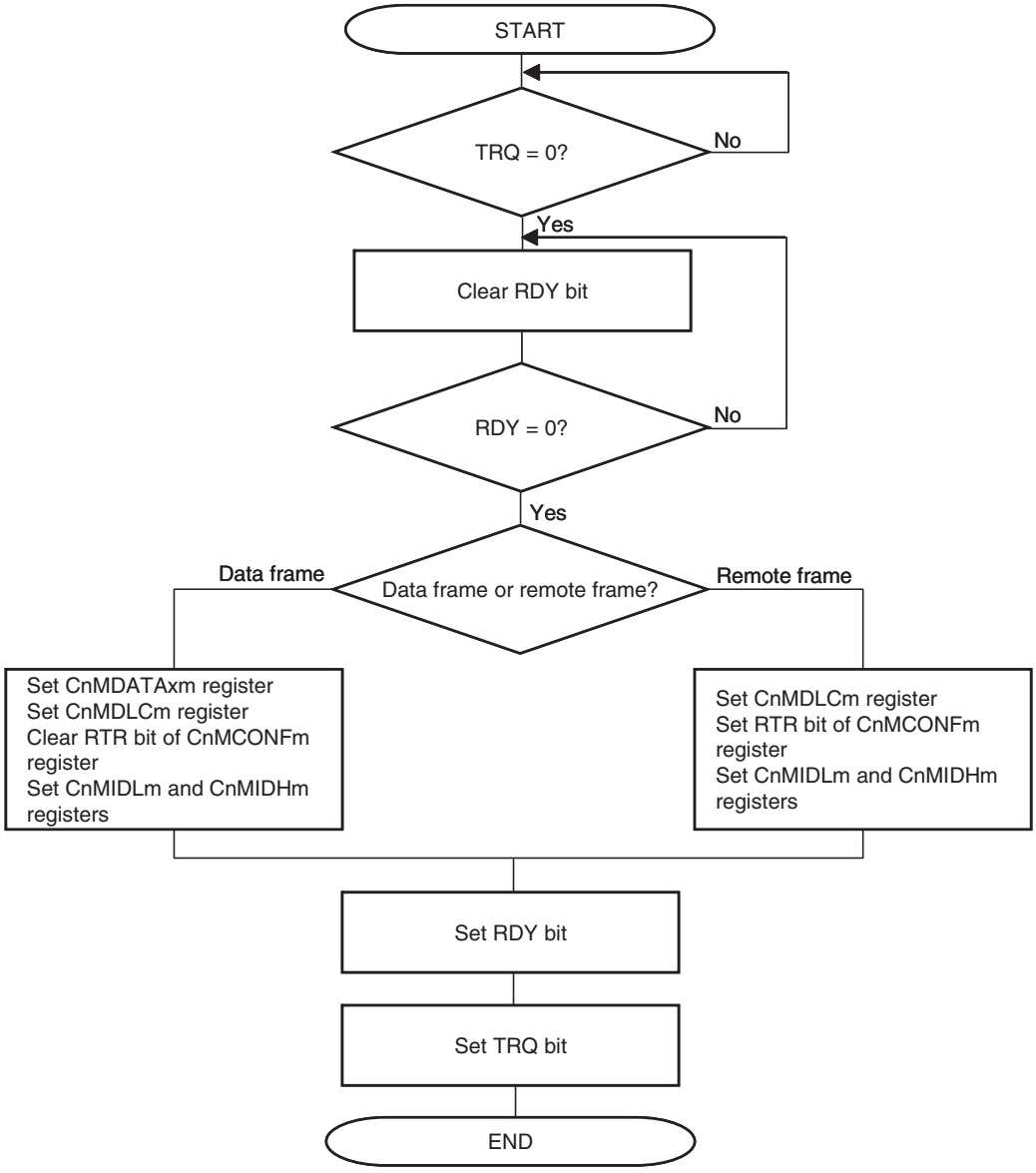


Figure 18-40 Message transmit processing

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 18-41 shows the processing for a transmit message buffer (MT[2:0] bits of CnMCONFm register = 000_B)

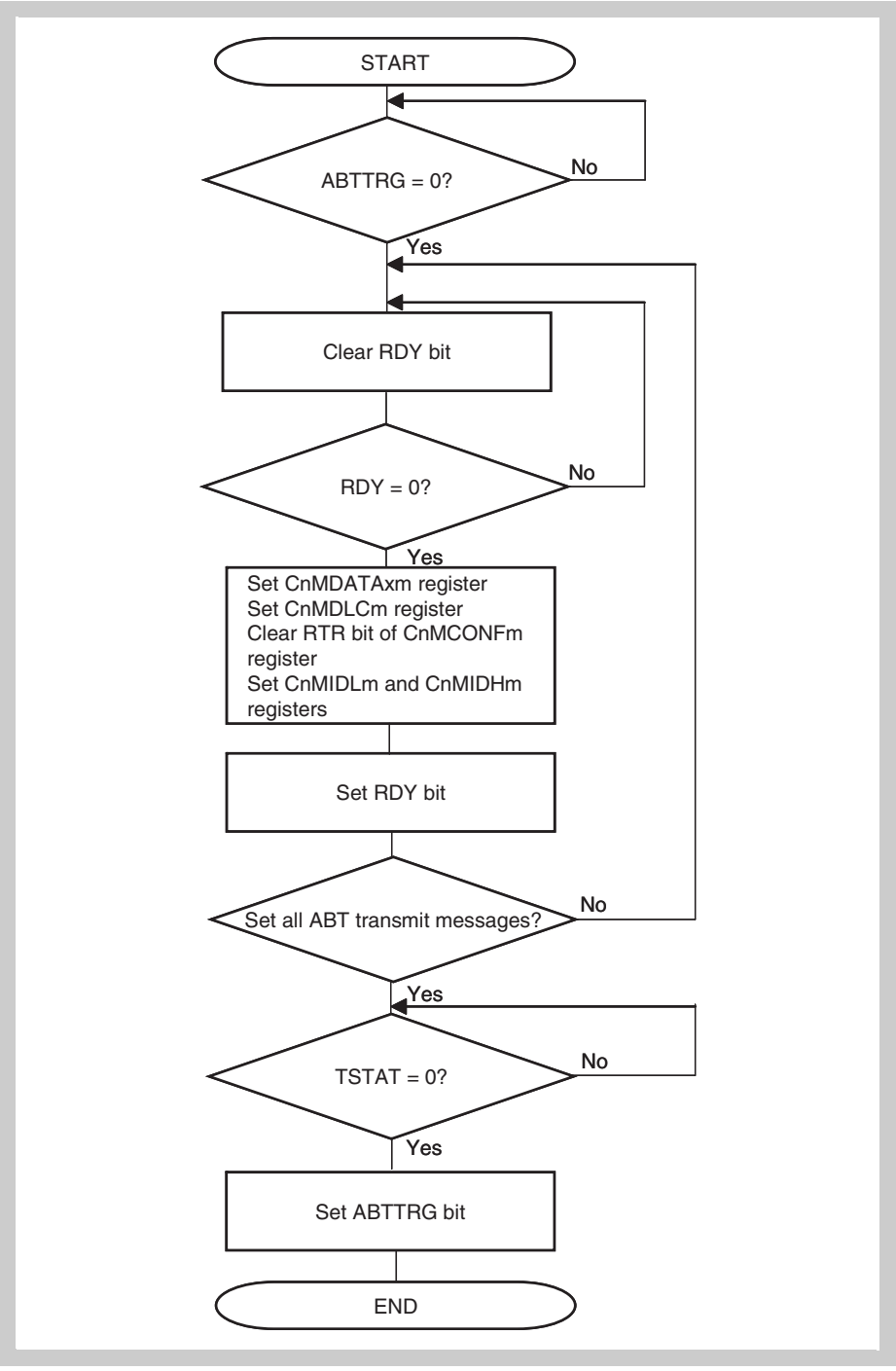


Figure 18-41 ABT message transmit processing

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 18-40 on page 822*.

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.

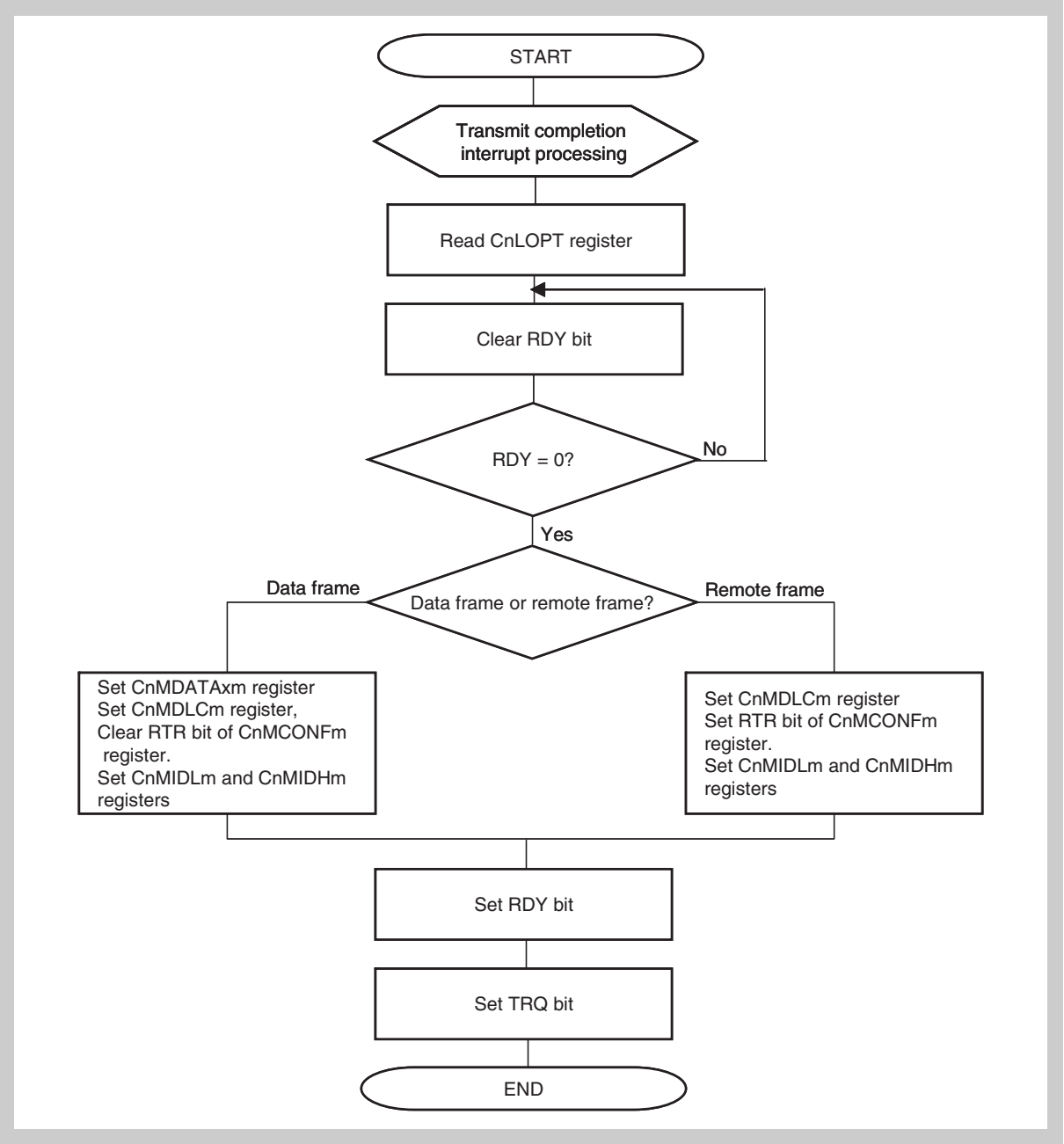


Figure 18-42 Transmission via interrupt (using CnLOPT register)

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.

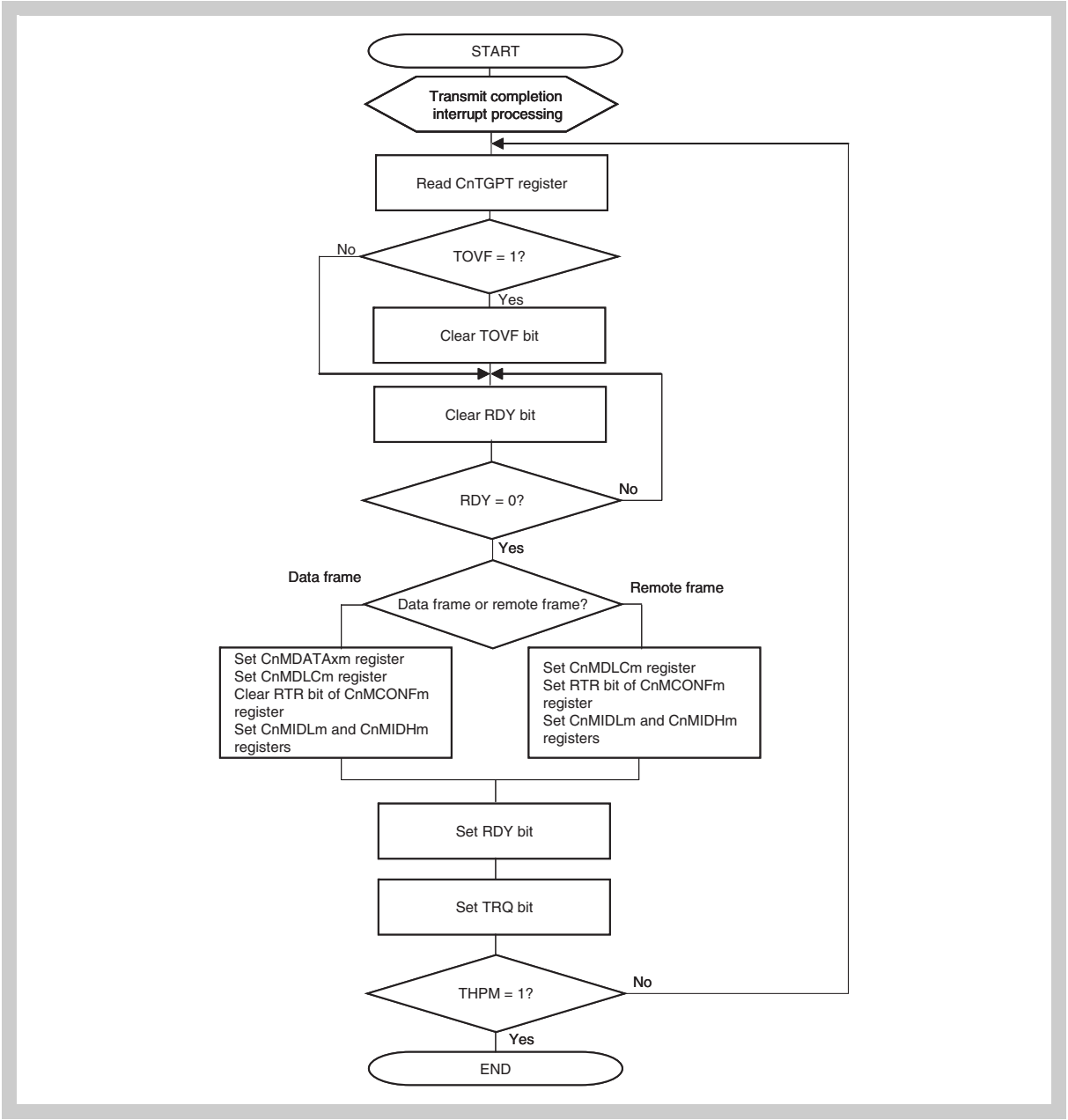


Figure 18-43 Transmission via interrupt (using CnTGPT register)

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.

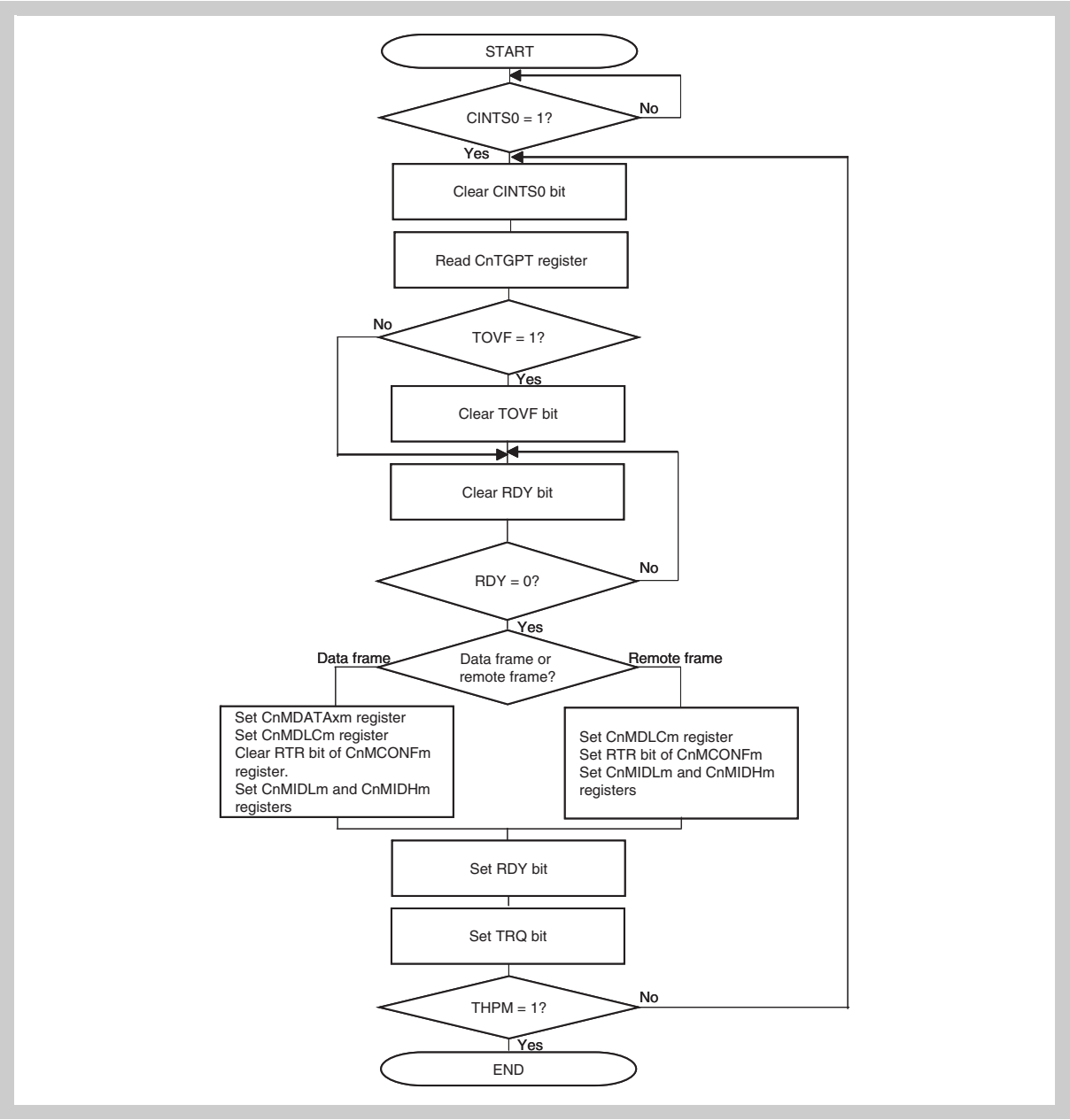


Figure 18-44 Transmission via software polling

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.

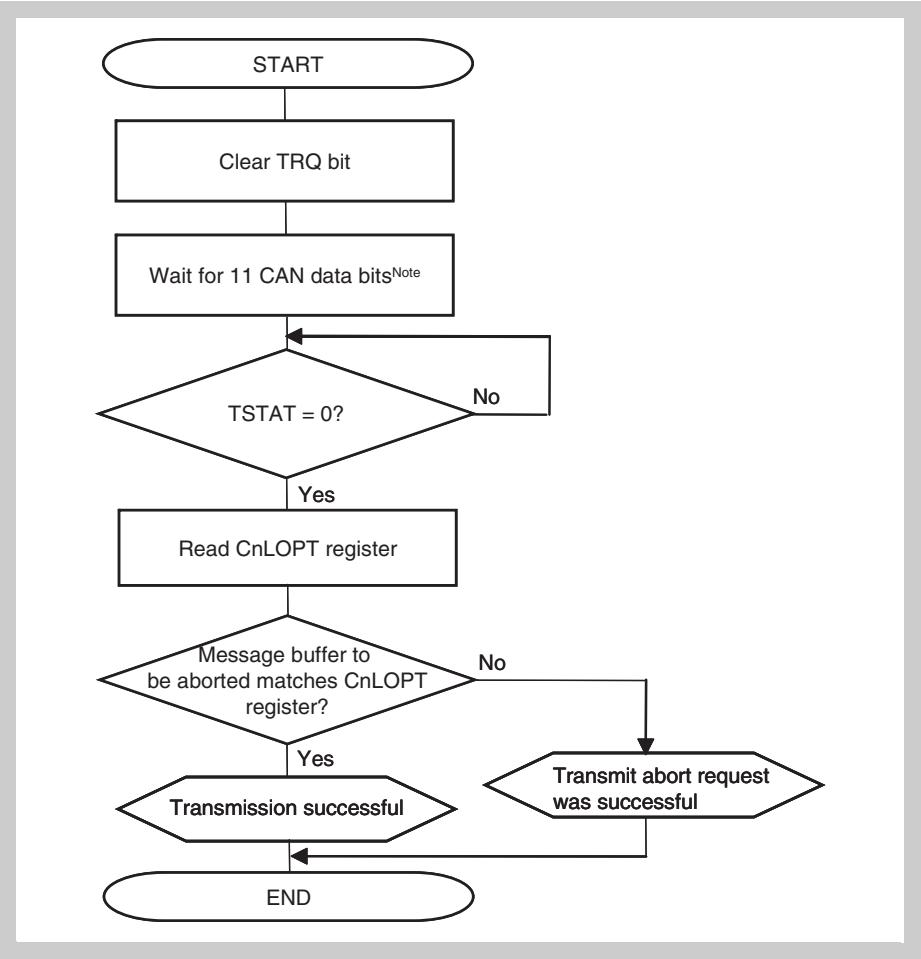


Figure 18-45 Transmission abort processing (except normal operation mode with ABT)

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.

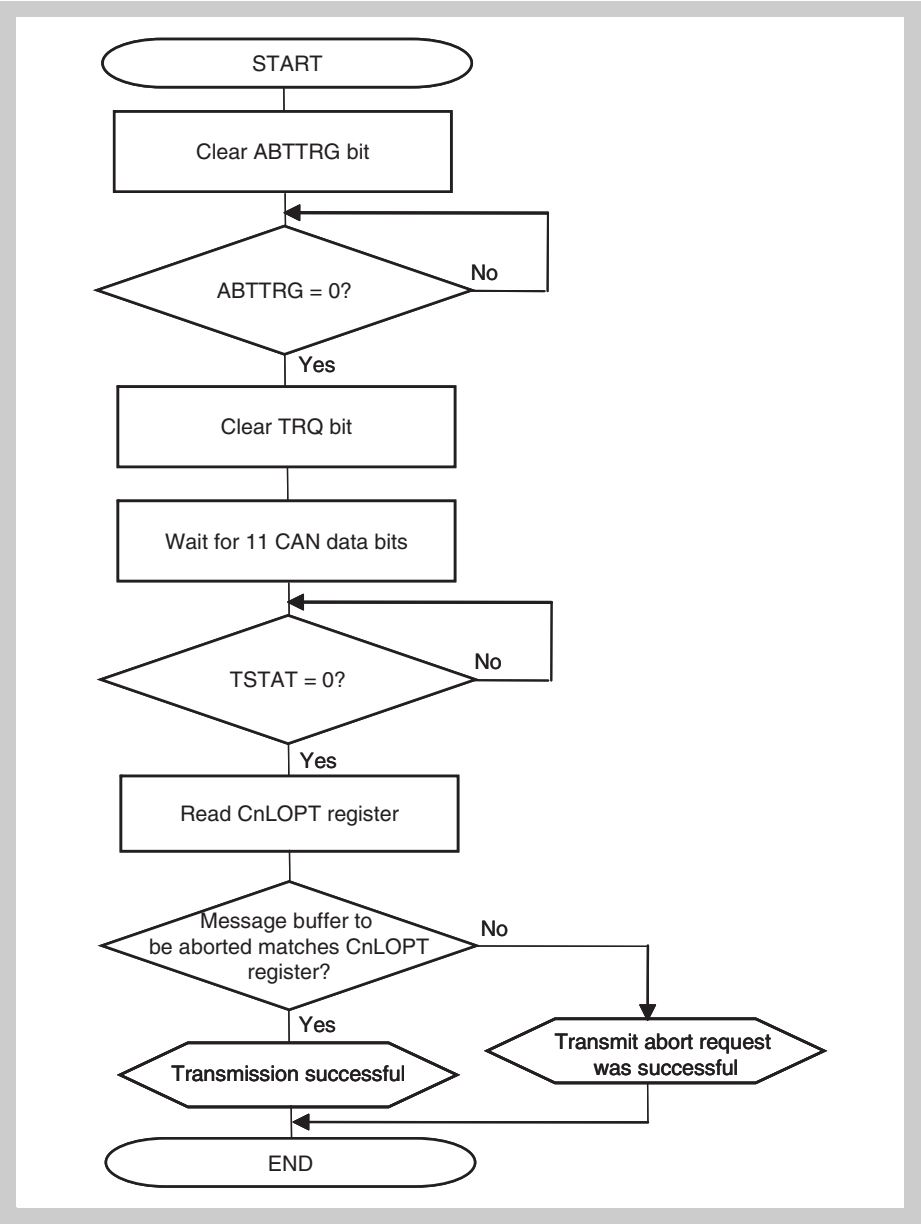


Figure 18-46 Transmission abort processing except for ABT transmission (normal operation mode with ABT)

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 18-47 shows the processing to skip resumption of transmitting a message that was stopped when transmission of an ABT message buffer was aborted.

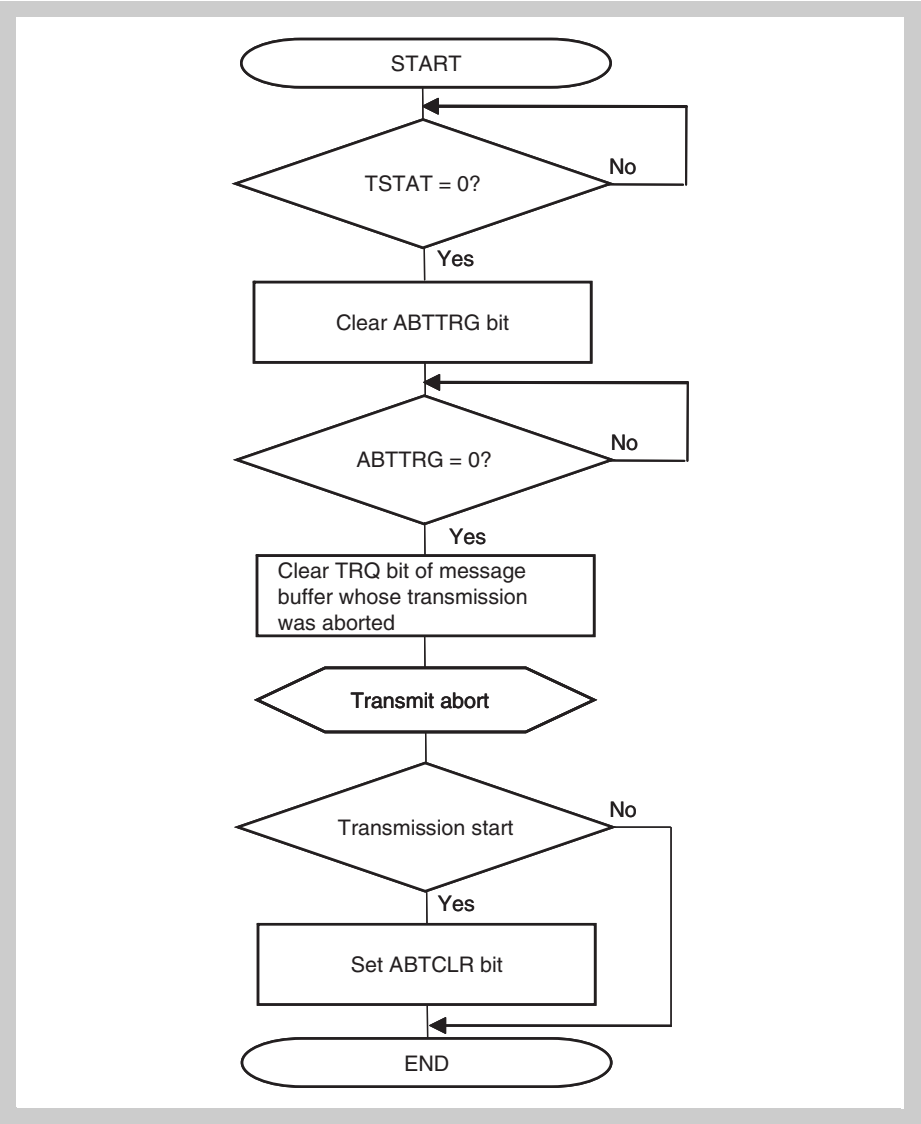


Figure 18-47 Transmission abort processing (normal operation mode with ABT)

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 18-47* or *Figure 18-48*. When clearing a transmission request in an area other than the ABT area, follow the procedure shown in *Figure 18-45 on page 827*.

Figure 18-48 shows the processing to not skip resumption of transmitting a message that was stopped when transmission of an ABT message buffer was aborted.

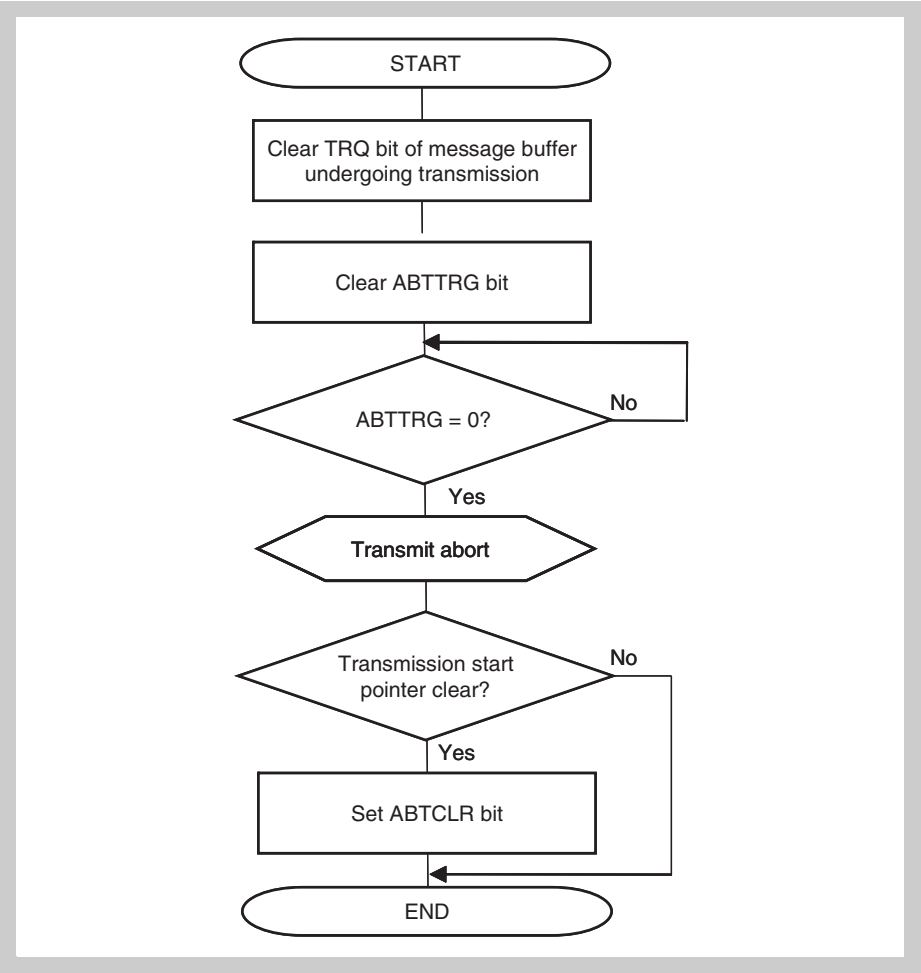


Figure 18-48 ABT transmission request abort processing (normal operation mode with ABT)

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 18-47* or *Figure 18-48*. When clearing a transmission request in an area other than the ABT area, follow the procedure shown in *Figure 18-45 on page 827*.

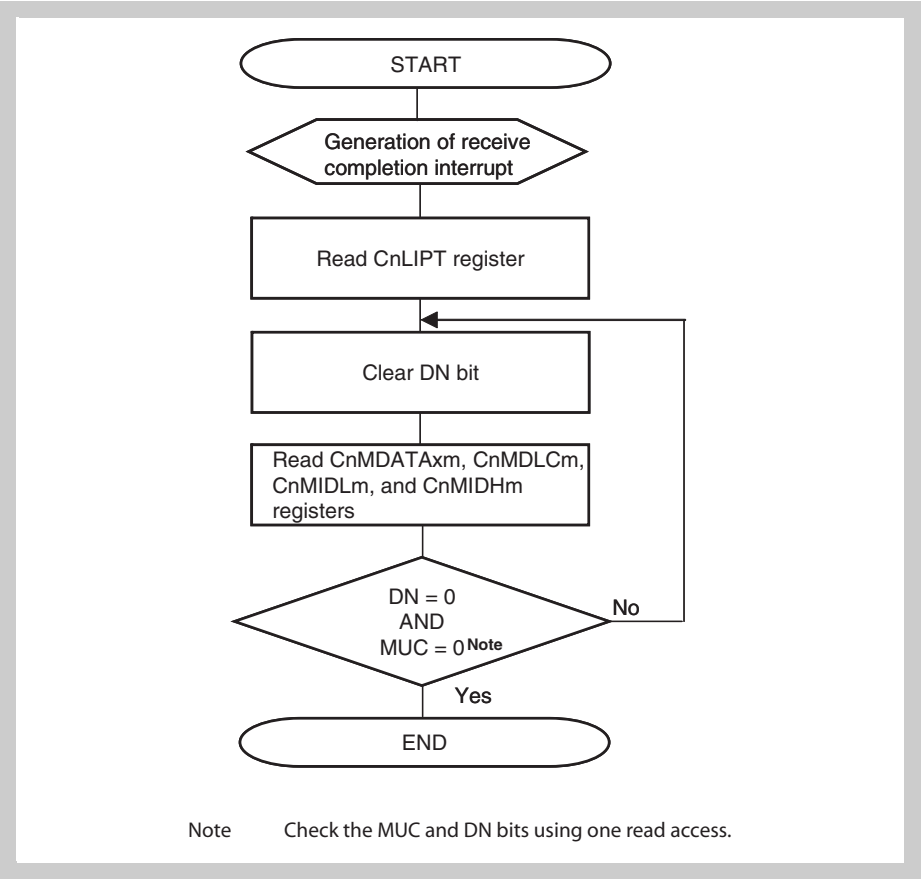


Figure 18-49 Reception via interrupt (using CnLIPT register)

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.

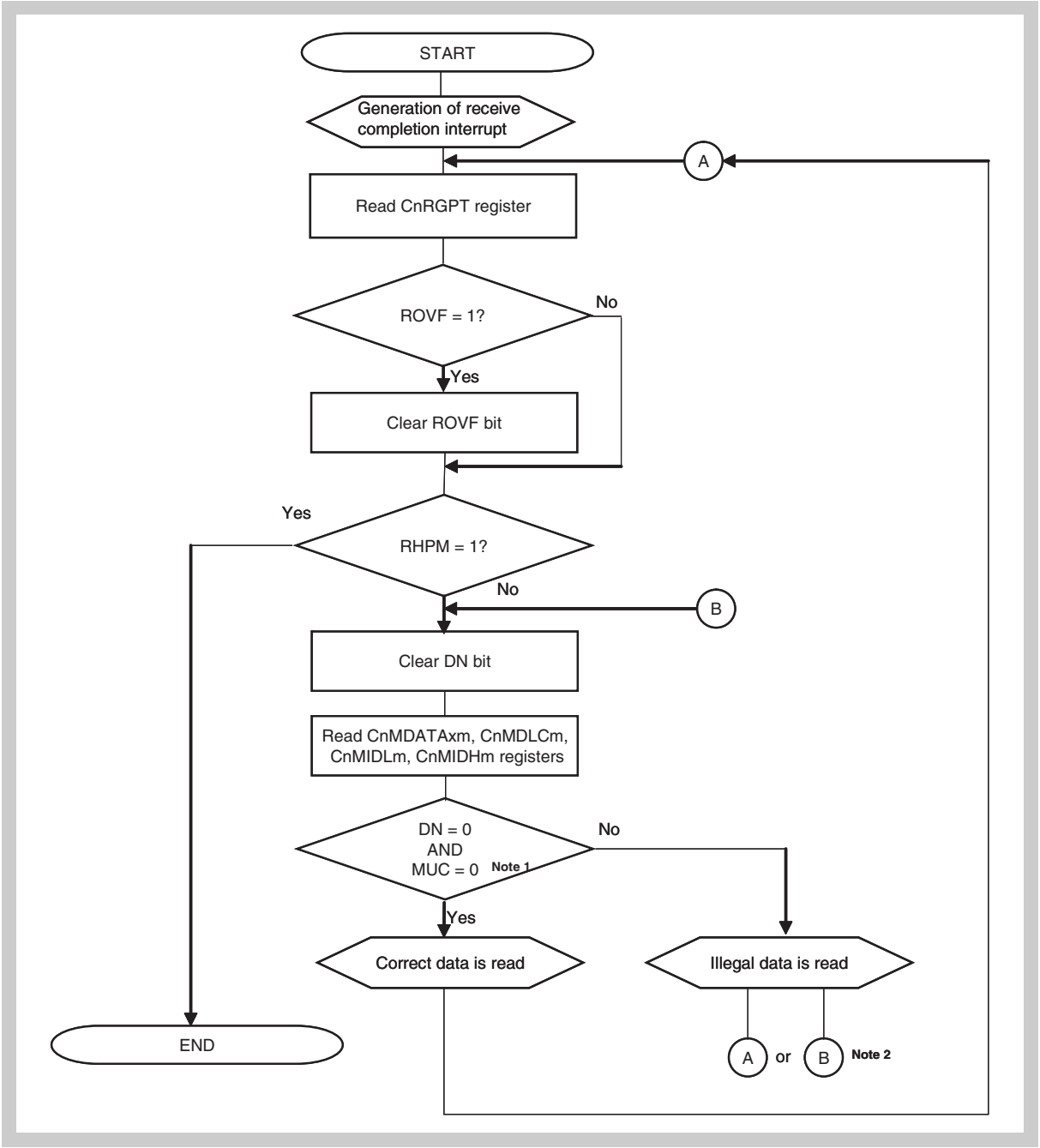


Figure 18-50 Reception via interrupt (using CnRGPT register)

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.

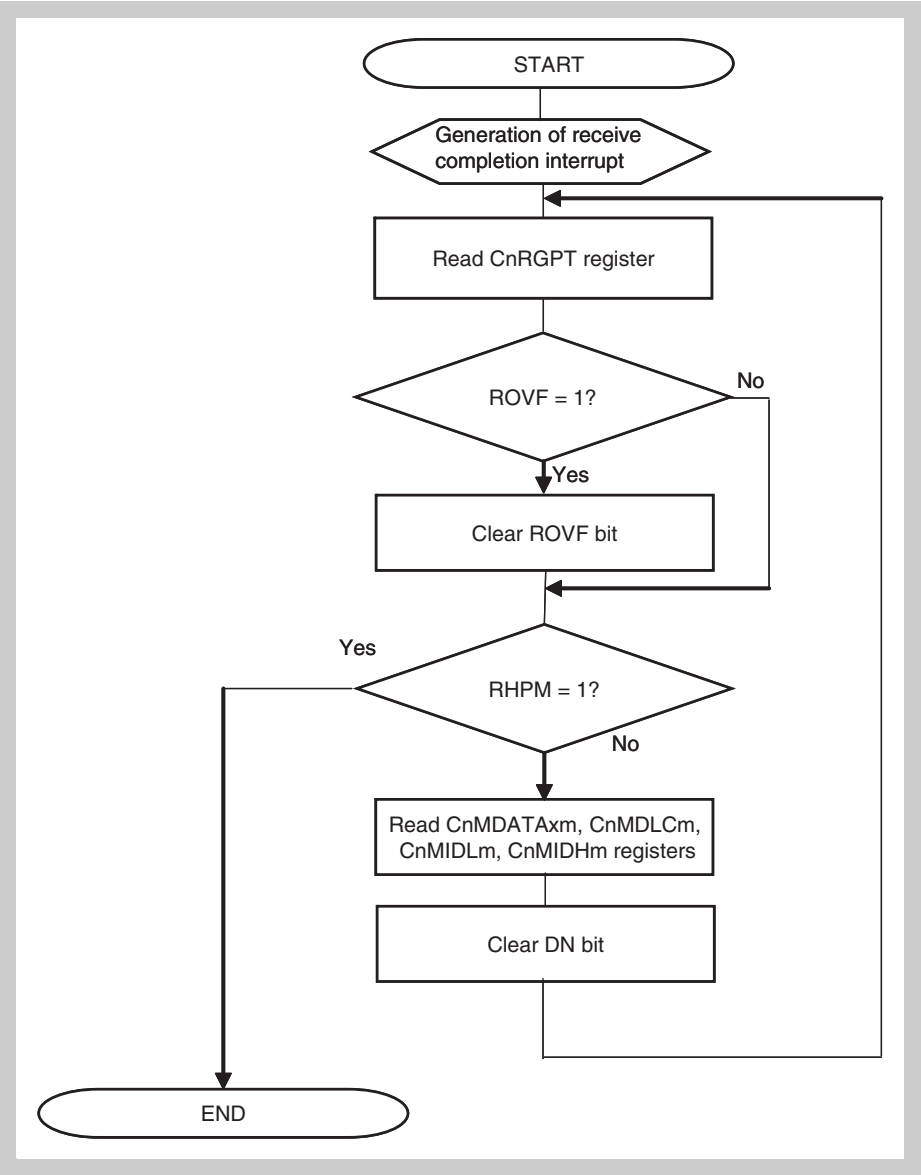


Figure 18-51 Reception via interrupt (using CnRGPT register), alternative way

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 18-50 on page 832*.

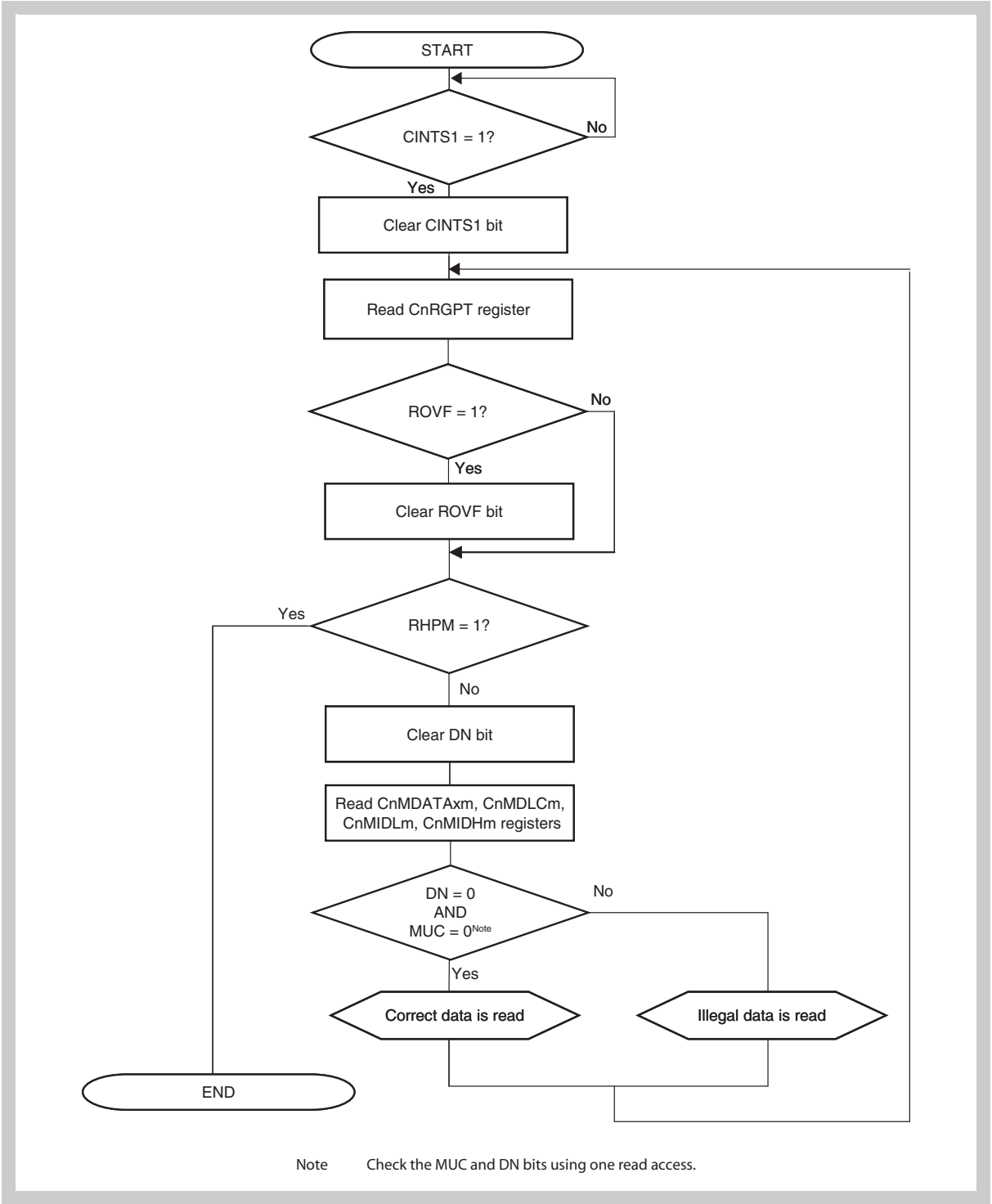


Figure 18-52 Reception via software polling

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.

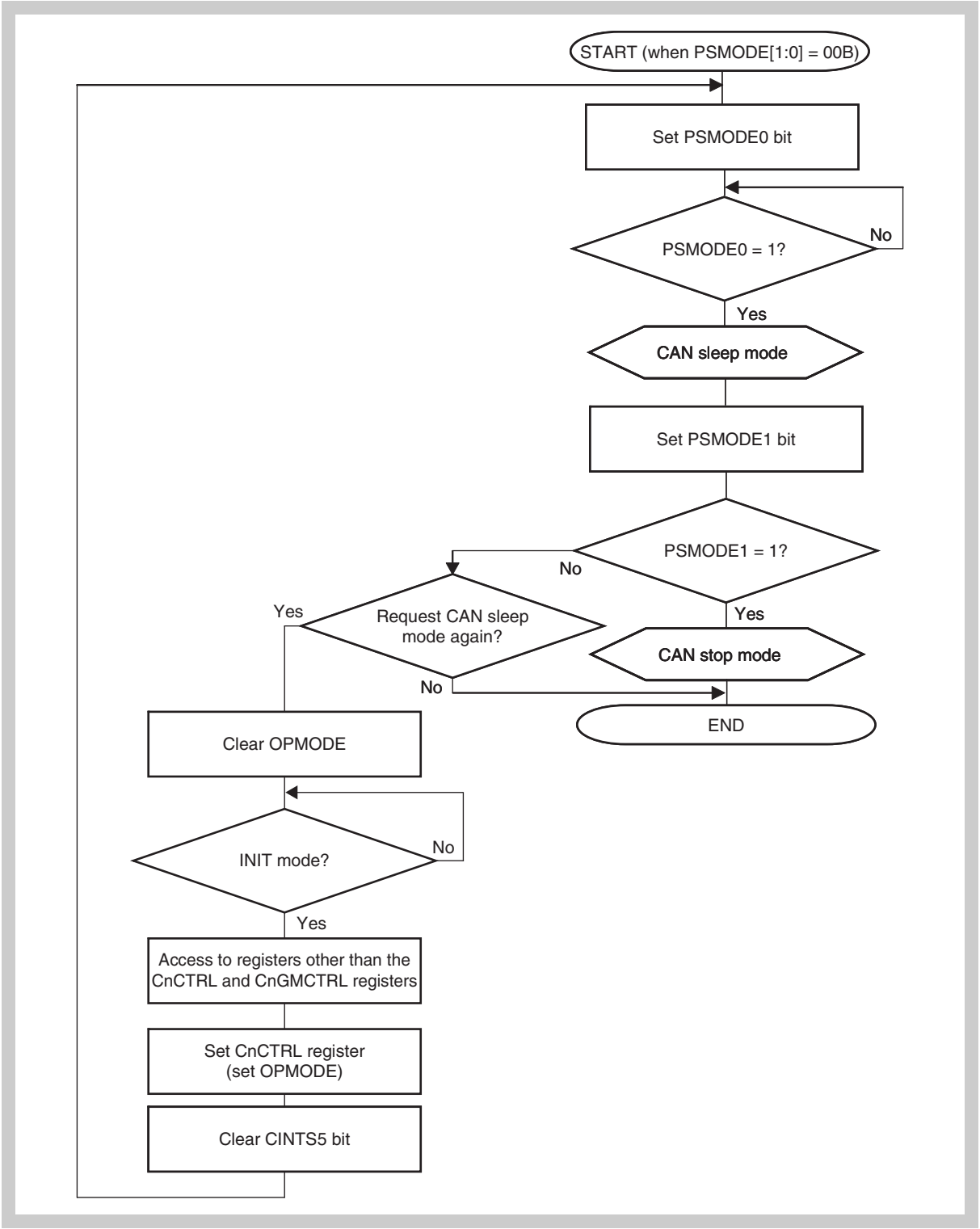


Figure 18-53 Setting CAN sleep mode/stop mode

Caution To abort transmission before making a request for the CAN sleep mode, perform processing according to *Figure 18-45 on page 827* and *Figure 18-47 on page 829*.

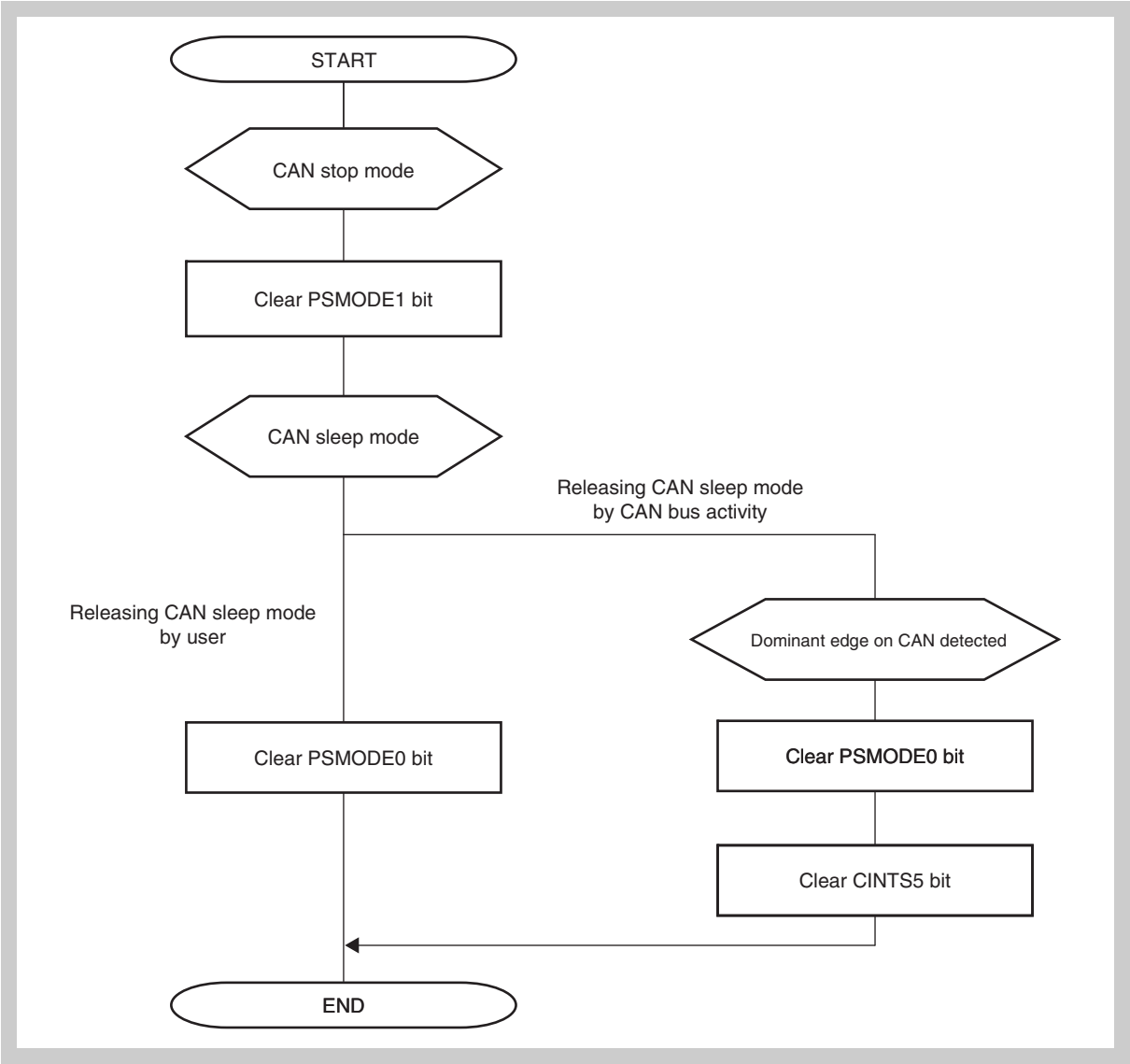


Figure 18-54 Clear CAN sleep/stop mode

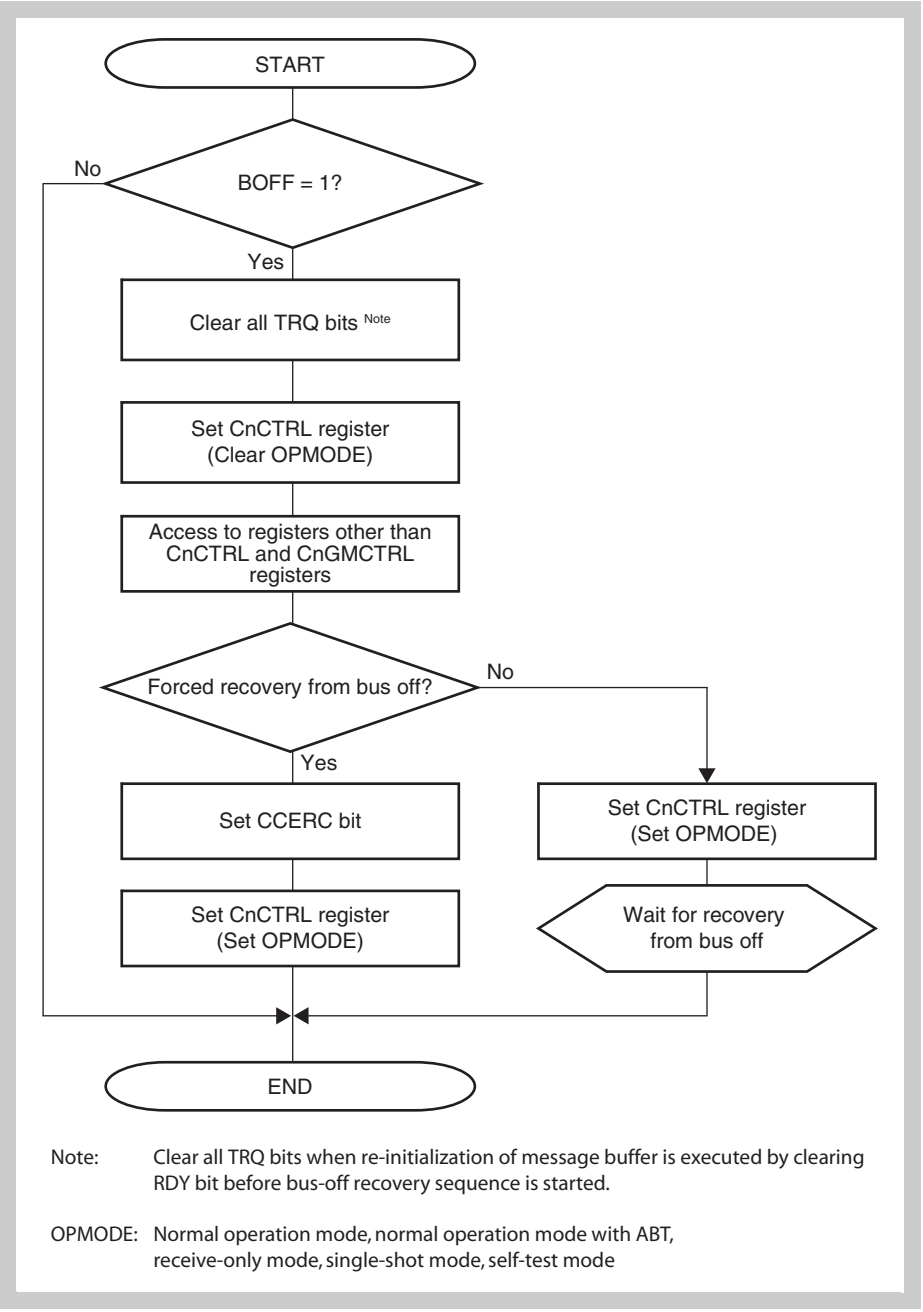


Figure 18-55 Bus-off recovery (except normal operation mode with ABT)

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.

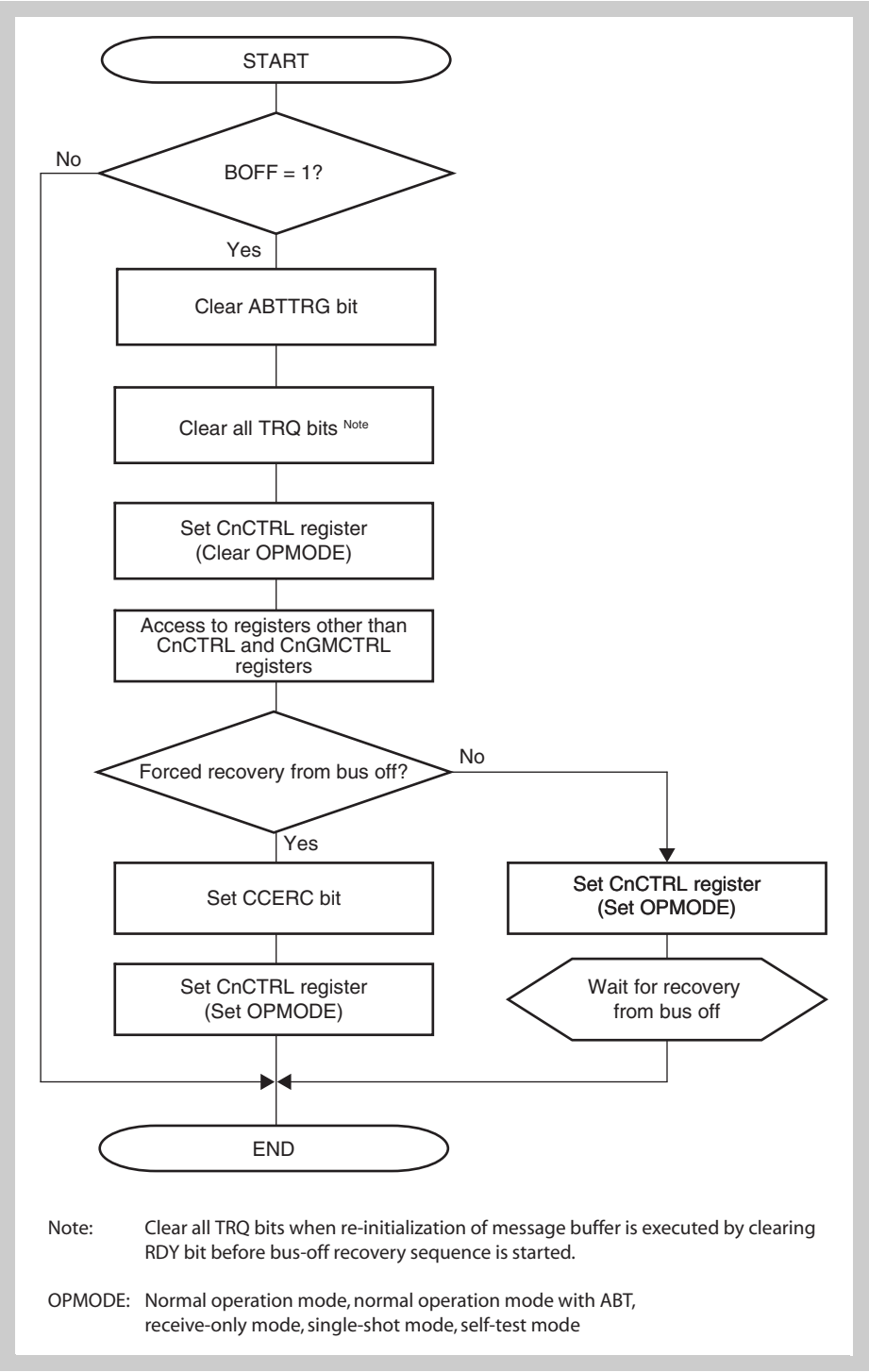


Figure 18-56 Bus-off recovery (Normal Operation Mode with ABT)

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.

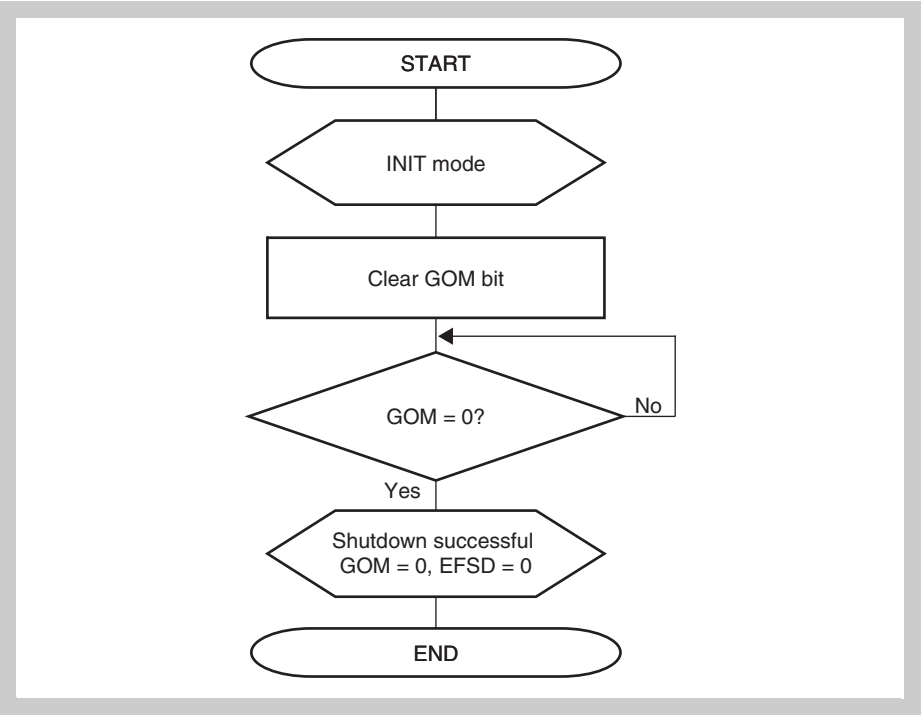


Figure 18-57 Normal shutdown process

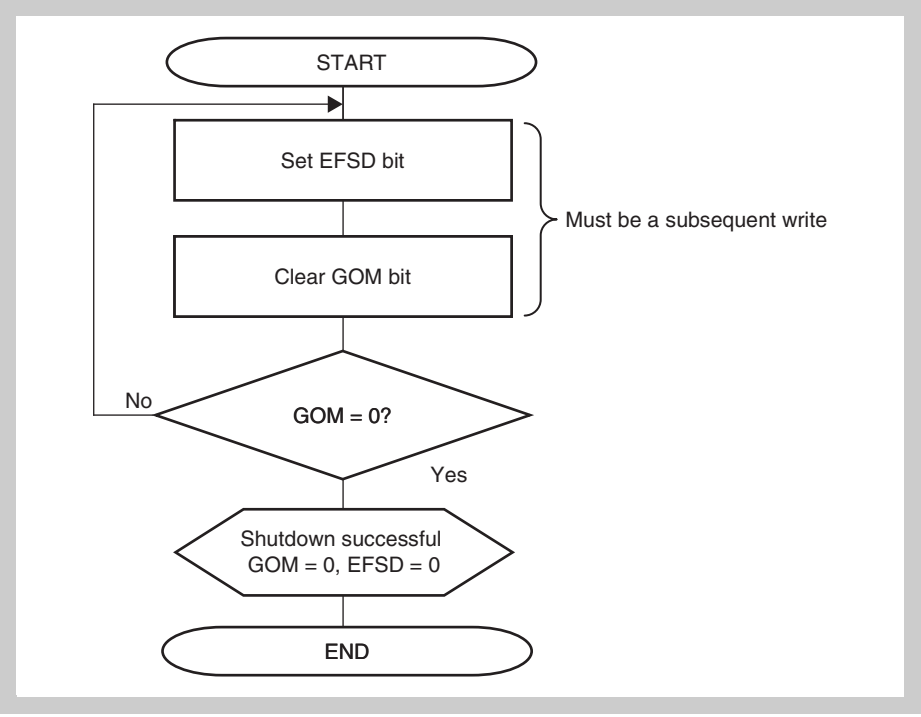


Figure 18-58 Forced shutdown process

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 *18.7 (1) "CnGMCTRL - CANn global control register"*)

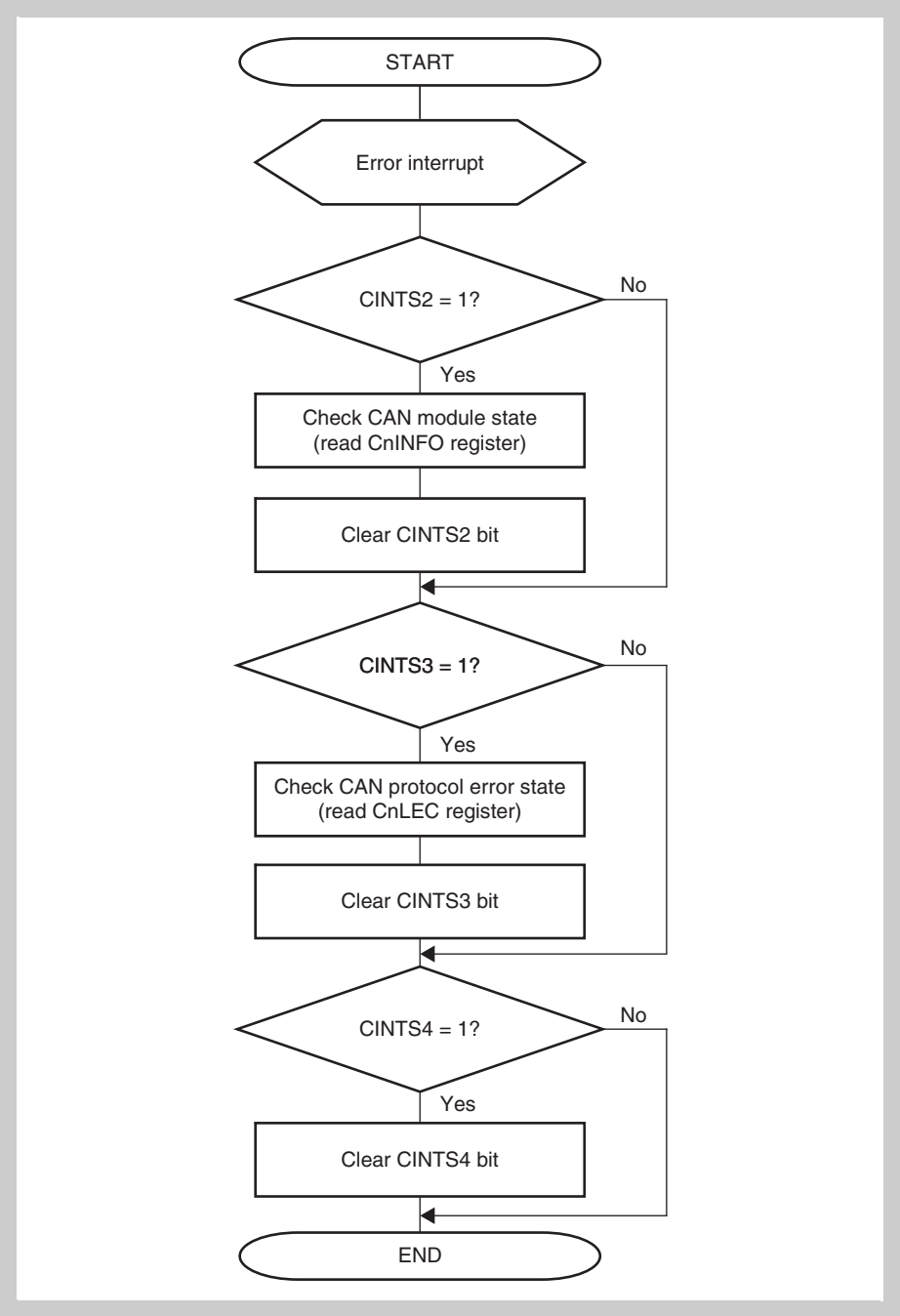


Figure 18-59 Error handling

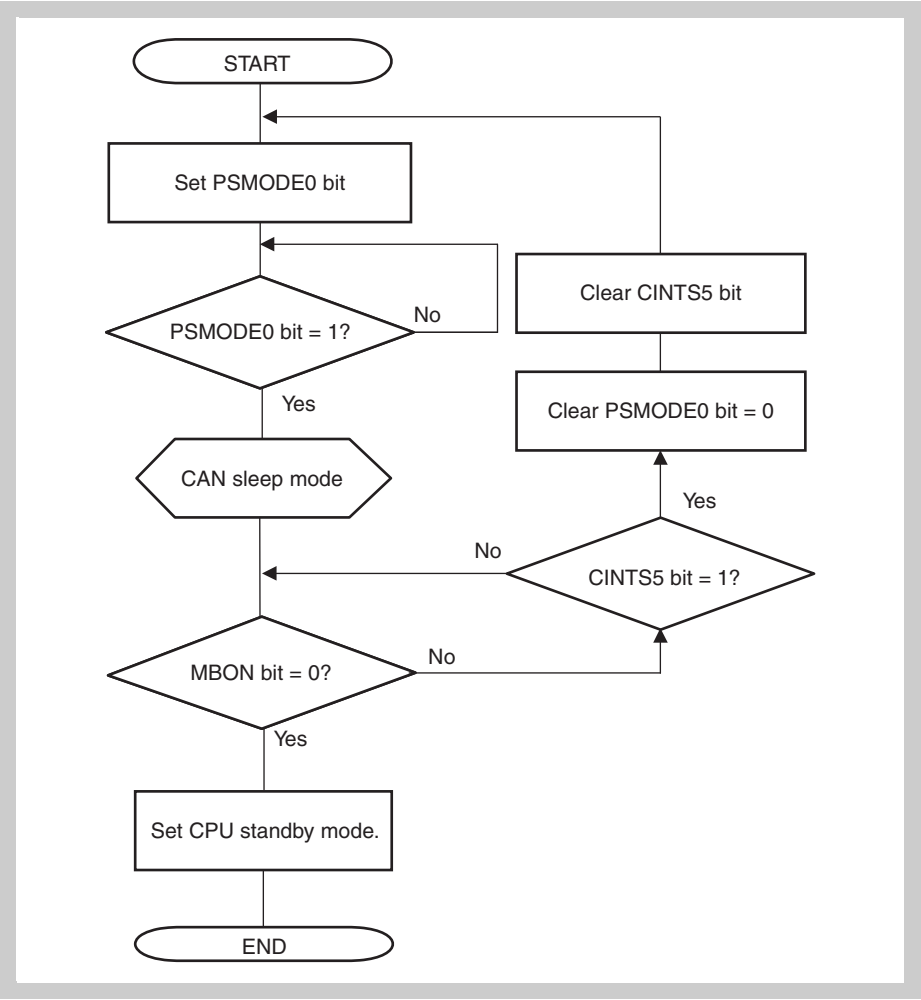


Figure 18-60 Setting CPU stand-by (from CAN sleep mode)

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.

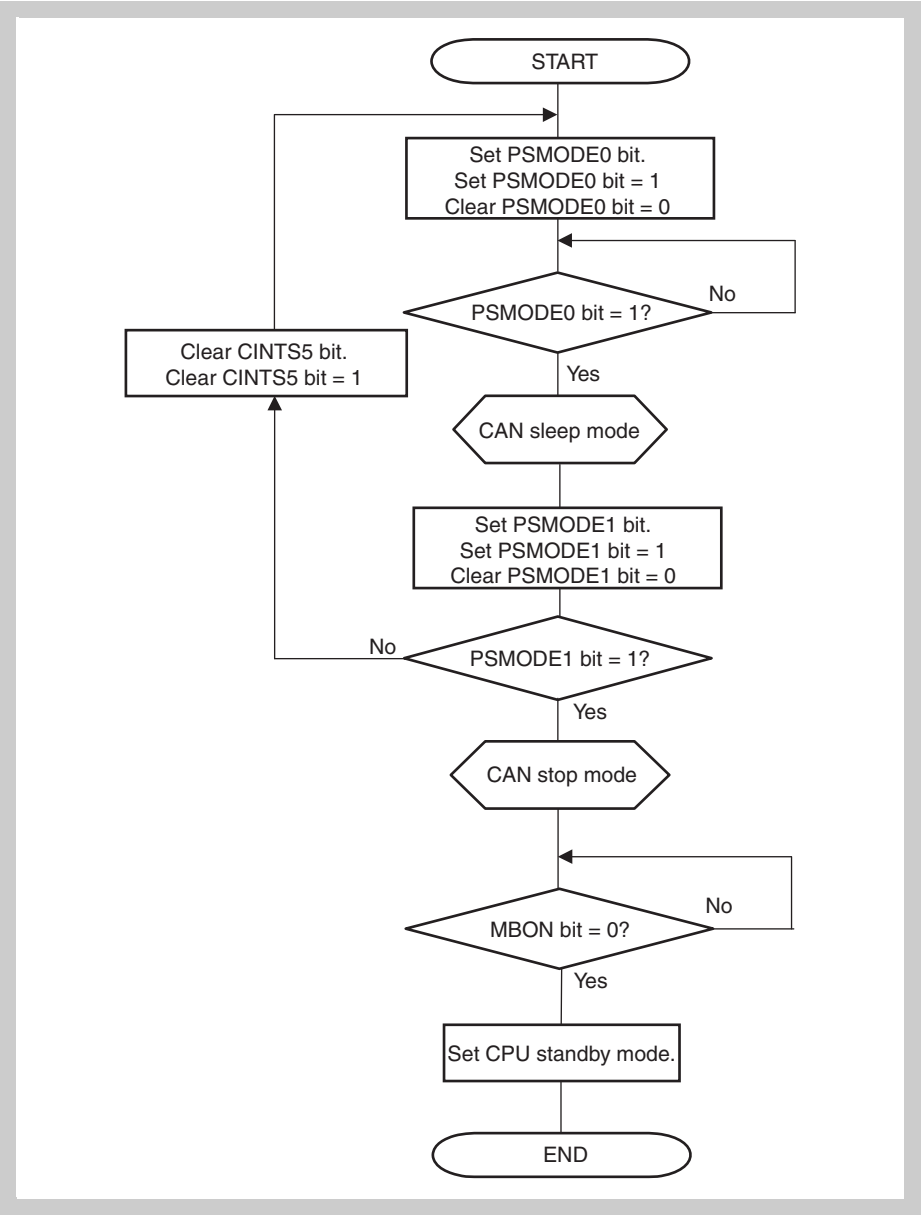


Figure 18-61 Setting CPU stand-by (from CAN stop mode)

Caution The CAN stop mode can only be released by writing 01_B to the PSMODE[1:0] bit of the CnCTRL register and not by a change in the CAN bus state.

Chapter 19 Random Number Generator (RNG)

The V850E/PH2 microcontrollers have following number of channels of a hardware random number generator (RNG):

RNG	µPD70F3447	µPD70F3187
Instances	0	1
Names	–	RNG

Note The random number generator is not supported on µPD70F3447.

19.1 Features

- Random number sequence passing FIPS and Maurer test
- Random number format: 16 bits
- Seed generated by hardware

19.2 Configuration

(1) Random number register (RNG)

The RNG register is a 16-bit register that holds the random number.

Access This registers can be read only in 16-bit units.

Address $FFFFF700_H$

Initial Value undefined

RNG

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Random number value															
R															

Note After read access to this register a certain time is required to generate the next random number. If a consecutive read access takes place before the new random number has been generated, the read access will be delayed.

19.3 Operation

19.3.1 Access timing

After read access to the RNG register it needs a certain time to generate the next random number. Moreover, when a consecutive read access takes place before the new random number has been generated, the read access will be delayed.

The access timing to the RNG register is as follows.

- Single read access to RNG register (when VSWC register = 13_H):

$$T_{single} = 102{,}5 \cdot f_{XX}^{-1}$$

- Consecutive read access to RNG register:

$$T_{consecutive} = T_{single} + (1024 \cdot f_{XX}^{-1})$$

Chapter 20 Port Functions

Note Throughout this sub-chapter, the individual instances of ports are identified by “m” (port number) and “n” (port bit number).

20.1 Features

- Input-only ports: 5
  I/O ports: 136
- Input/Output direction can be specified in 1.bit units
- Noise removal circuit provided for external interrupts and timer inputs
- Edge detect function for external interrupts (rising-, falling-, both edges)
- Security features for port 5 and 6 shared as 3-phase PWM timer outputs
- Emergency shut off feature
- Software protection feature

20.2 Port Configuration

The V850E/PH2 incorporates a total of 141 input/output ports (including 5 input-only ports) labelled port 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, AL, AH, DH, DL, CS, CM, CT, and CD. The port configuration is shown in *Figure 20-1* below.

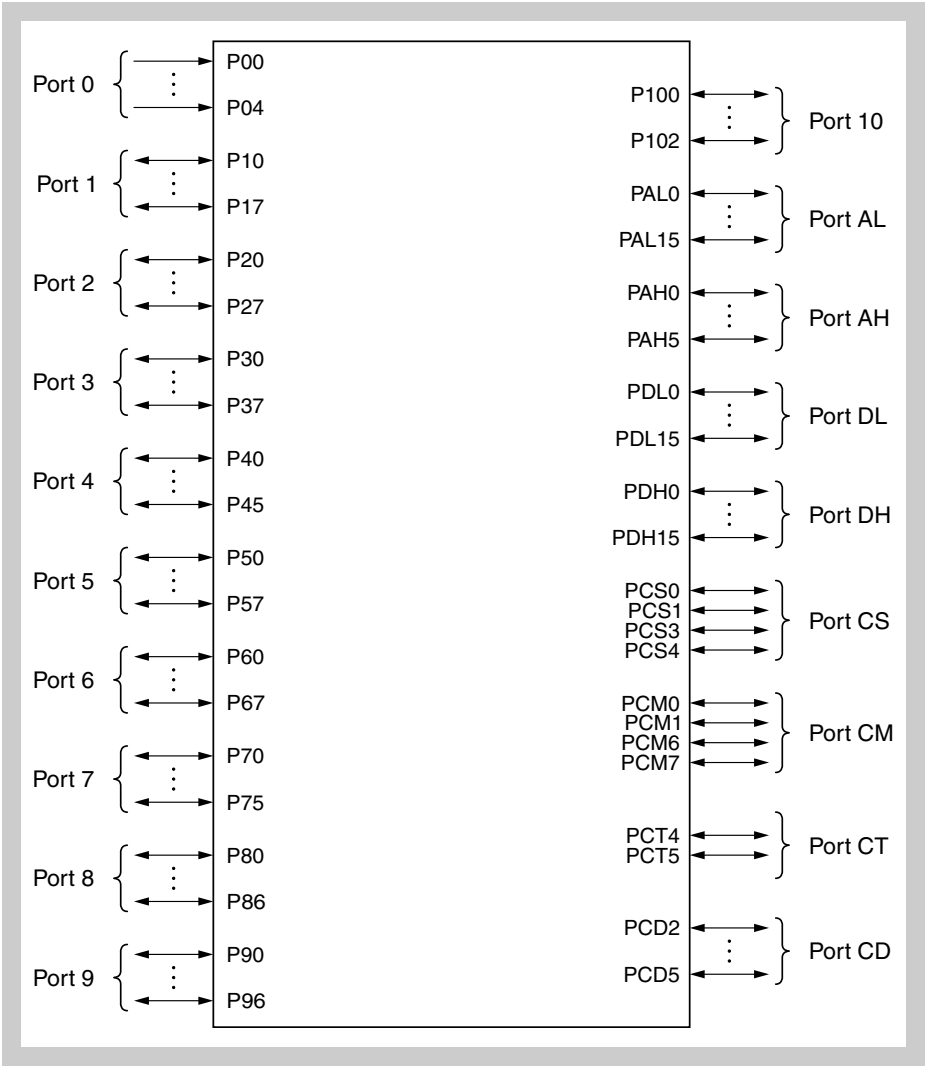


Figure 20-1 Port Configuration

20.2.1 Function of each port

The port functions of V850E/PH2 are shown in *Table 20-1* below.

The port type can vary for each individual bit of a port. In addition to their port functions, these pins are also shared with on-chip peripheral I/O pins in control mode.

Table 20-1 Port functions and types

Port Name	Pin Name	Port Function	Function in Control Mode	Port Type
Port 0	P00 to P04	5-bit input only	External interrupt input External A/D conversion start trigger input Emergency shut-off input	3, 15, 15A
Port 1	P10 to P17	8-bit I/O port	Timer I/O (TMP0, TMP1, TMP2, TMP3)	6
Port 2	P20 to P27	8-bit I/O port	Timer I/O (TMP4, TMP5, TMP6, TMP7)	6
Port 3	P30 to P37	8-bit I/O port	Serial interface I/O (UARTC0, UARTC1, AFCAN0, AFCAN1[a])	1S, 2, 9
Port 4	P40 to P45	6-bit I/O port	Serial interface I/O (CSIB0, CSIB1[a])	1E, 2, 4C
Port 5	P50 to P57	8-bit I/O port	Timer output (TMR0)	11, 13
Port 6	P60 to P67	8-bit I/O port	Timer I/O (TMR1)	12, 13, 14
Port 7	P70 to P75	6-bit I/O port	Timer I/O (TMT0, TMT1)	6, 8
Port 8	P80 to P86	7-bit I/O port	Serial interface I/O (CSI30)	1S, 2, 4, 5, 7
Port 9	P90 to P96	7-bit I/O port	Serial interface I/O (CSI31[a])	1S, 2, 4, 5, 7
Port 10	P100 to P102	3-bit I/O port	Timer I/O (TENC1[a], TMP8, TMR0, TMR1)	6, 10
Port AL	PAL0 to PAL15	16-bit I/O port	External address bus (A0-A15)[a]	1
Port AH	PAH0 to PAH5	6-bit I/O port	External address bus (A16-A21)[a]	1
Port DL	PDL0 to PDL15	16-bit /IO port	External data bus (D0-D15)[a]	4C
Port DH	PDH0 to PDH15	16-bit I/O port	External data bus (D16-D31)[a]	4C
Port CS	PCS0, PCS1, PCS3, PCS4	4-bit I/O port	External bus interface control signal output ($\overline{CS0}$, $\overline{CS1}$, $\overline{CS3}$, $\overline{CS4}$)[a]	1
Port CM	PCM0, PCM1, PCM6, PCM7	4-bit I/O port	External bus interface control signal I/O ($\overline{WAIT}$)[a]	1, 2C
Port CT	PCT4, PCT5	2-bit I/O port	External bus interface control signal output ($\overline{RD}$, $\overline{WR}$)[a]	1
Port CD	PCD2 to PCD5	4-bit I/O port	External bus interface control signal output ($\overline{BEN0}$-$\overline{BEN3}$)[a]	1

a) Alternate control function not available on µPD70F3447.

20.2.2 Port types

(1) Port type 1

Port type 1 provides a general purpose I/O port with peripheral output function.

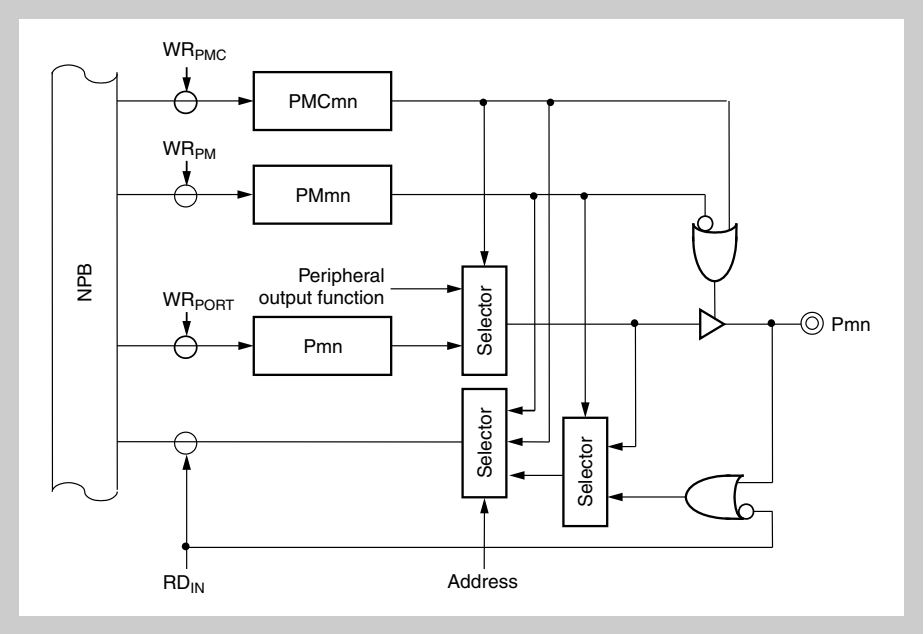


Figure 20-2 Port Type 1

(2) Port type 1S

Port type 1S provides a general purpose I/O port with peripheral output function. This type is similar to port type 1, but features a Schmitt trigger input buffer characteristic.

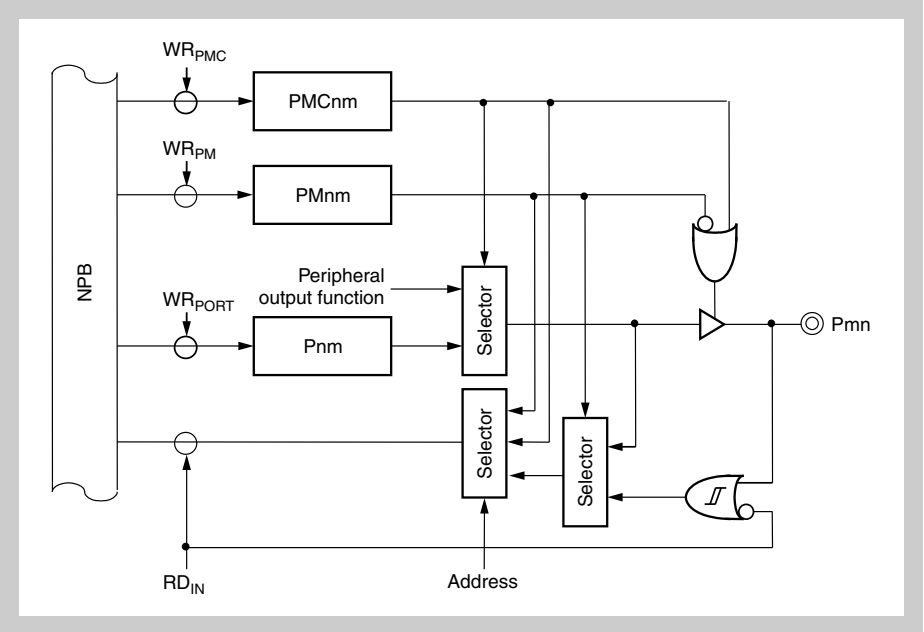


Figure 20-3 Port Type 1S

(3) Port type 1E

Port type 1E provides a general purpose I/O port with peripheral output function. In peripheral function mode a control signal is provided to enable or disable the output.

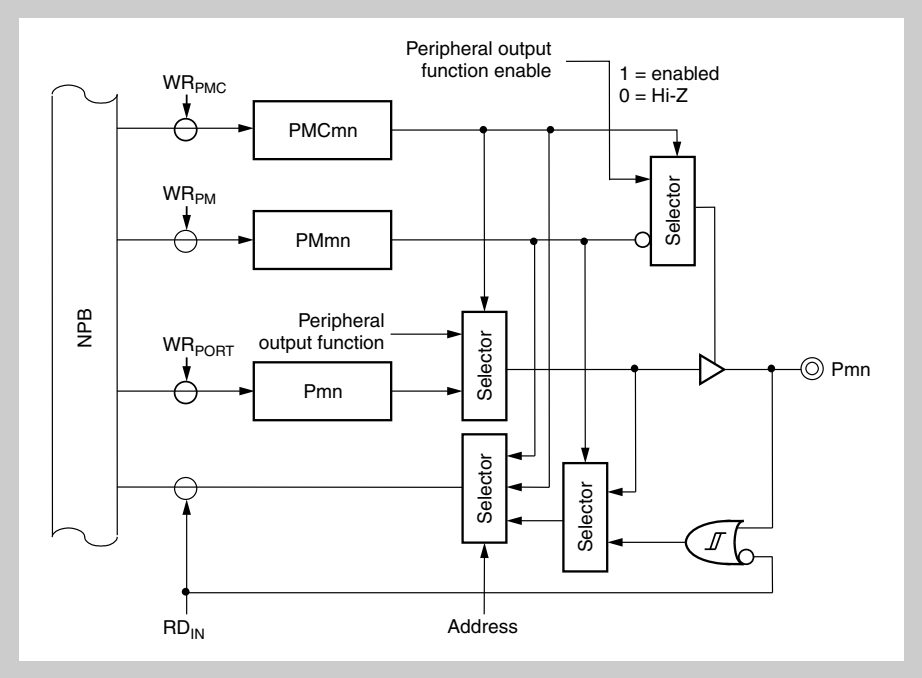


Figure 20-4 Port Type 1E

(4) Port type 2

Port type 2 provides a general purpose I/O port with peripheral input function.

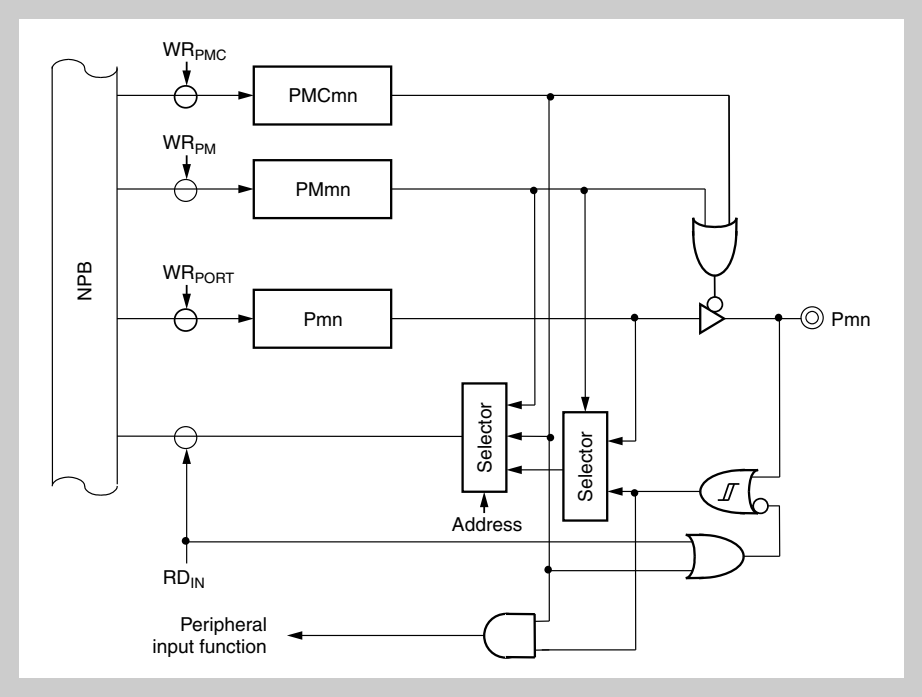


Figure 20-5 Port Type 2

(5) Port type 2A

Port type 2A provides a general purpose I/O port with peripheral input function. This type is similar as port type 2, but in port mode the peripheral input function is forced to high level.

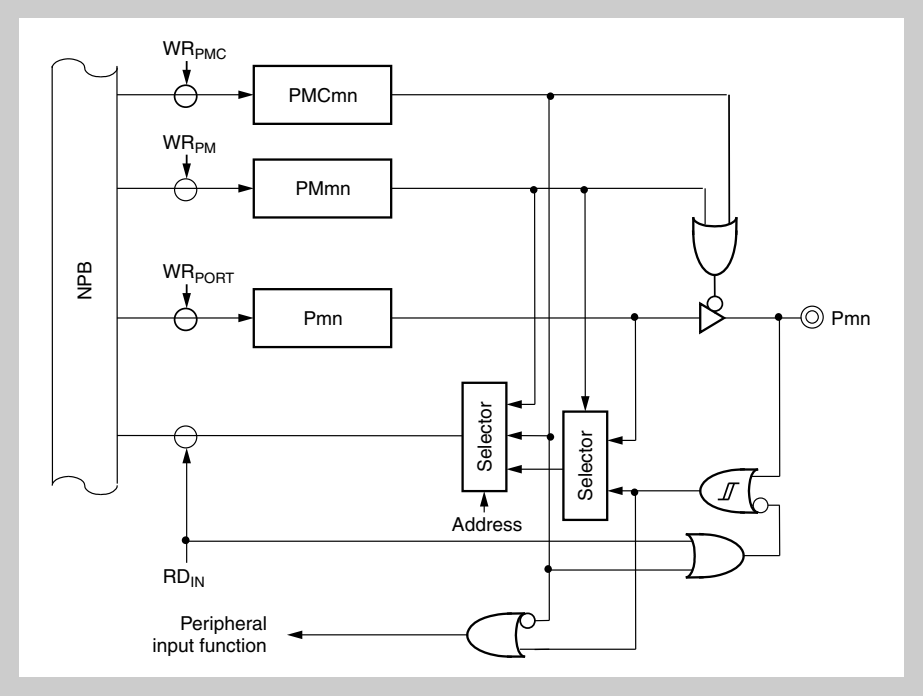


Figure 20-6 Port Type 2A

(6) Port type 2C

Port type 2C provides a general purpose I/O port with peripheral input function. This type is similar to type 2, but features CMOS input buffer characteristic.

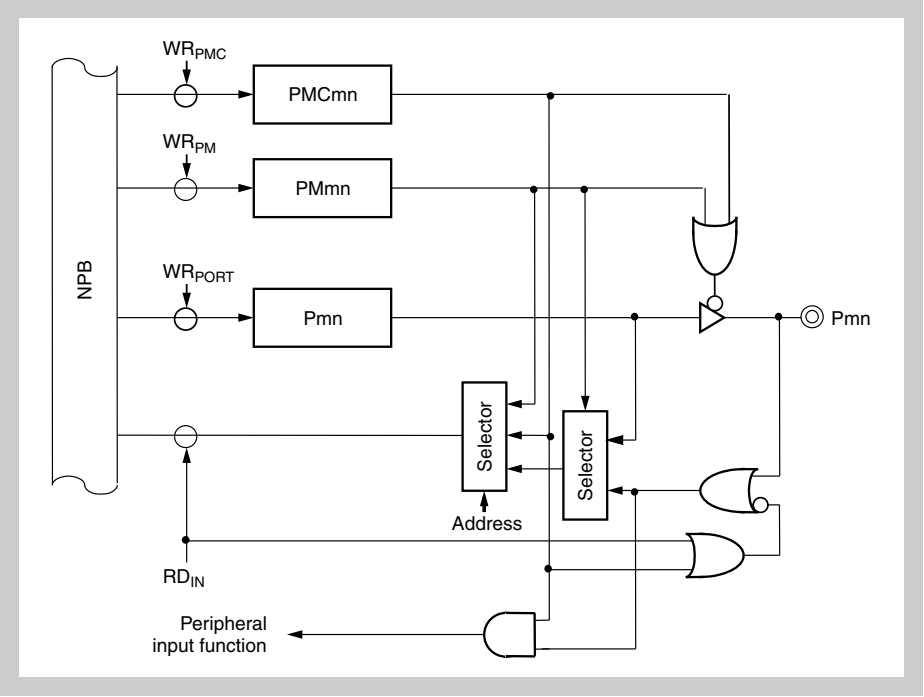


Figure 20-7 Port Type 2C

(7) Port type 3

Port type 3 provides a general purpose input port with NMI interrupt input function.

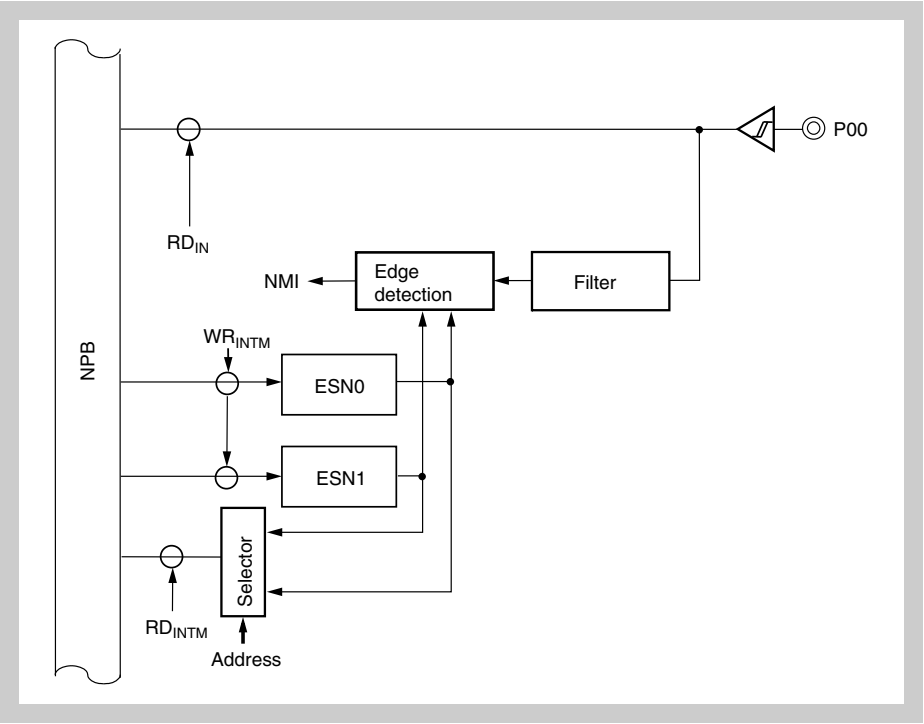


Figure 20-8 Port Type 3

(8) Port type 4

Port type 4 provides a general purpose I/O port with peripheral I/O function. Peripheral output enable is controlled by the corresponding peripheral function.

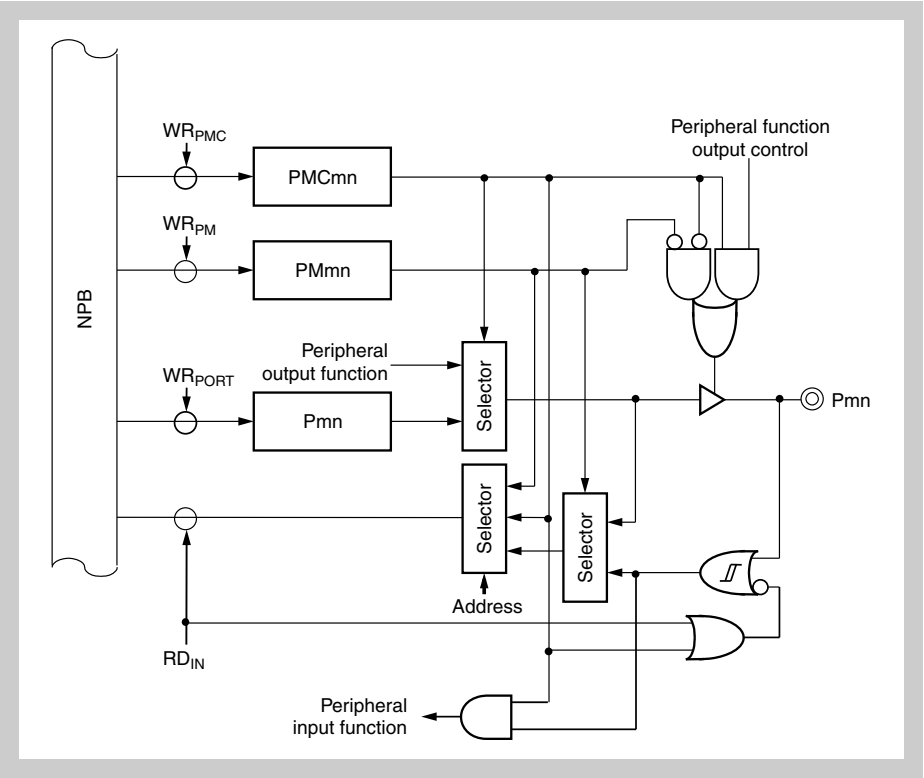


Figure 20-9 Port Type 4

(9) Port type 4C

Port type 4 provides a general purpose I/O port with peripheral I/O function. Peripheral output enable is controlled by the corresponding peripheral function.

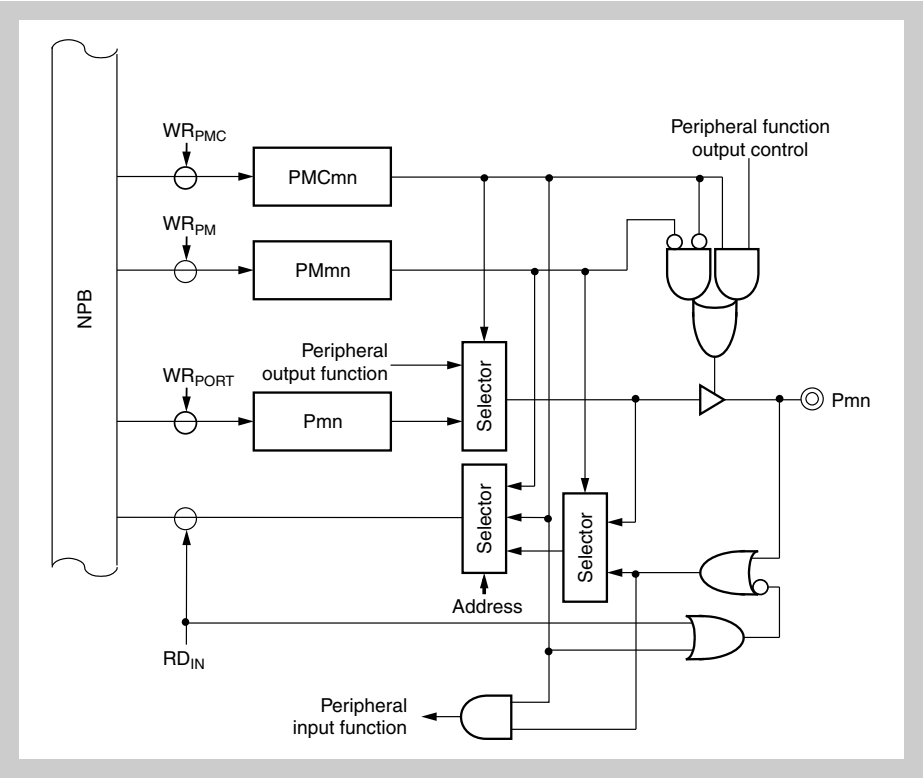


Figure 20-10 Port Type 4C

(10) Port type 5

Port type 5 provides a general purpose I/O port with peripheral I/O function. If the peripheral input function is disabled, the value of the peripheral input signal is fixed to low level.

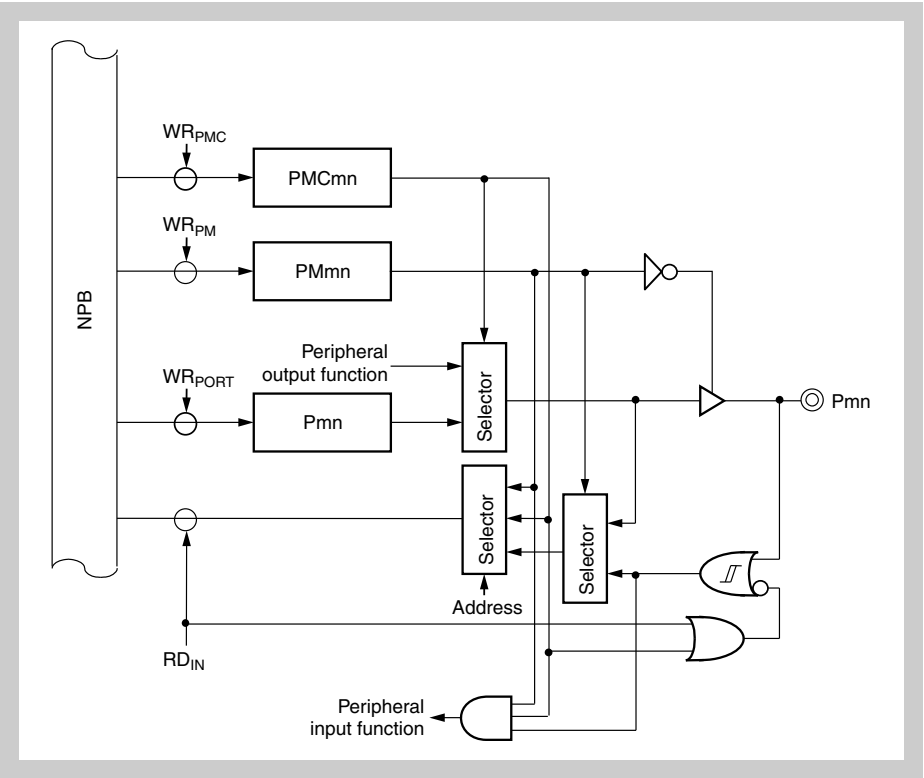


Figure 20-11 Port Type 5

(11) Port type 6

Port type 6 provides a general purpose I/O port with peripheral output function and digitally filtered peripheral input function.

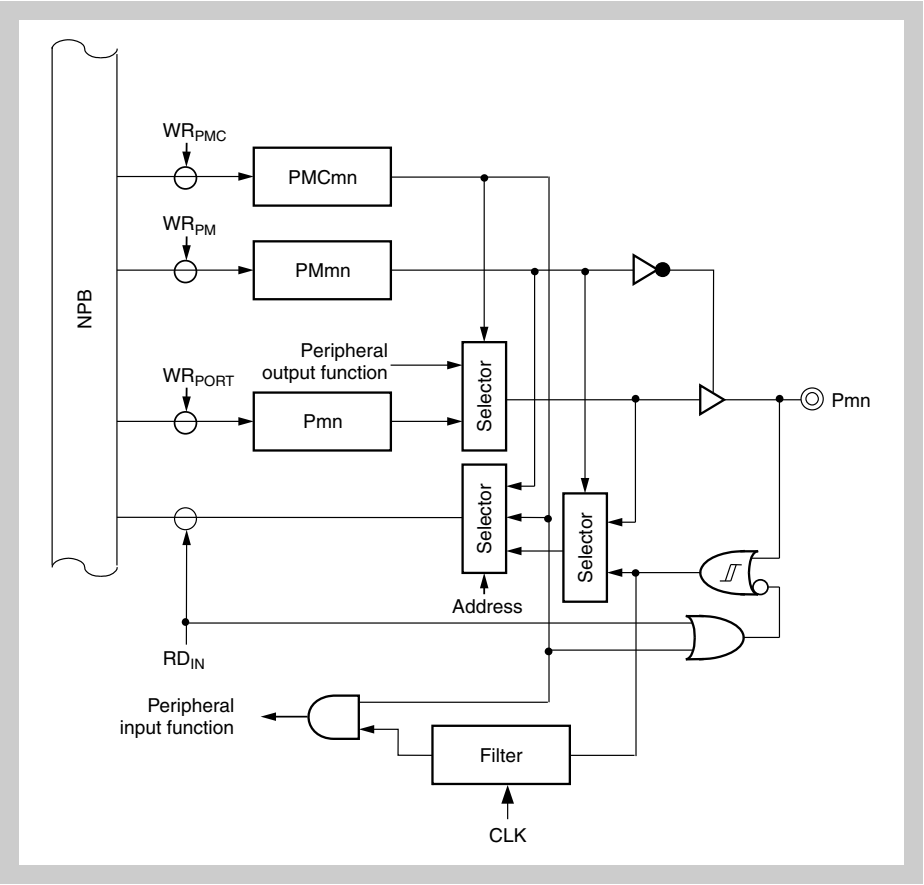


Figure 20-12 Port Type 6

(12) Port type 7

Port type 7 provides a general purpose I/O port with peripheral output function and external interrupt input capability.

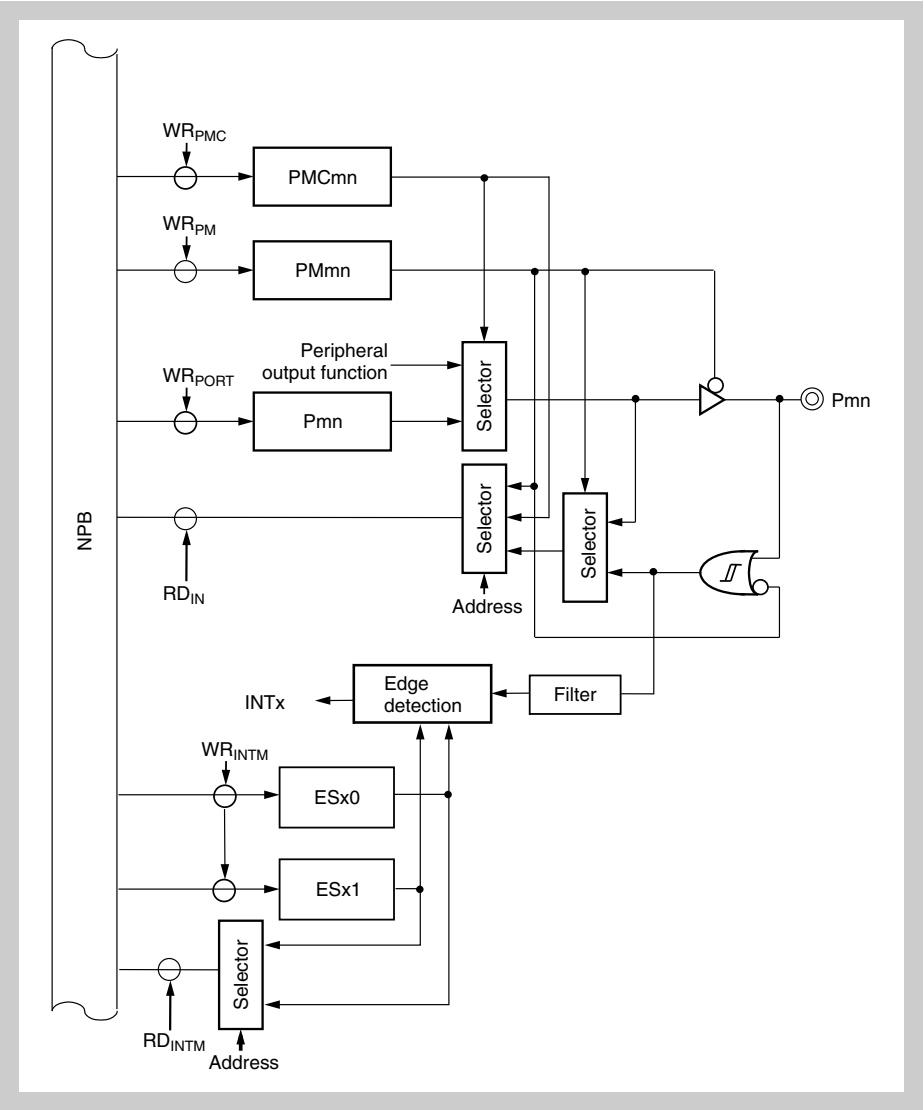


Figure 20-13 Port Type 7

(13) Port type 8

Port type 8 provides a general purpose I/O port with digitally filtered peripheral input function and external interrupt input capability.

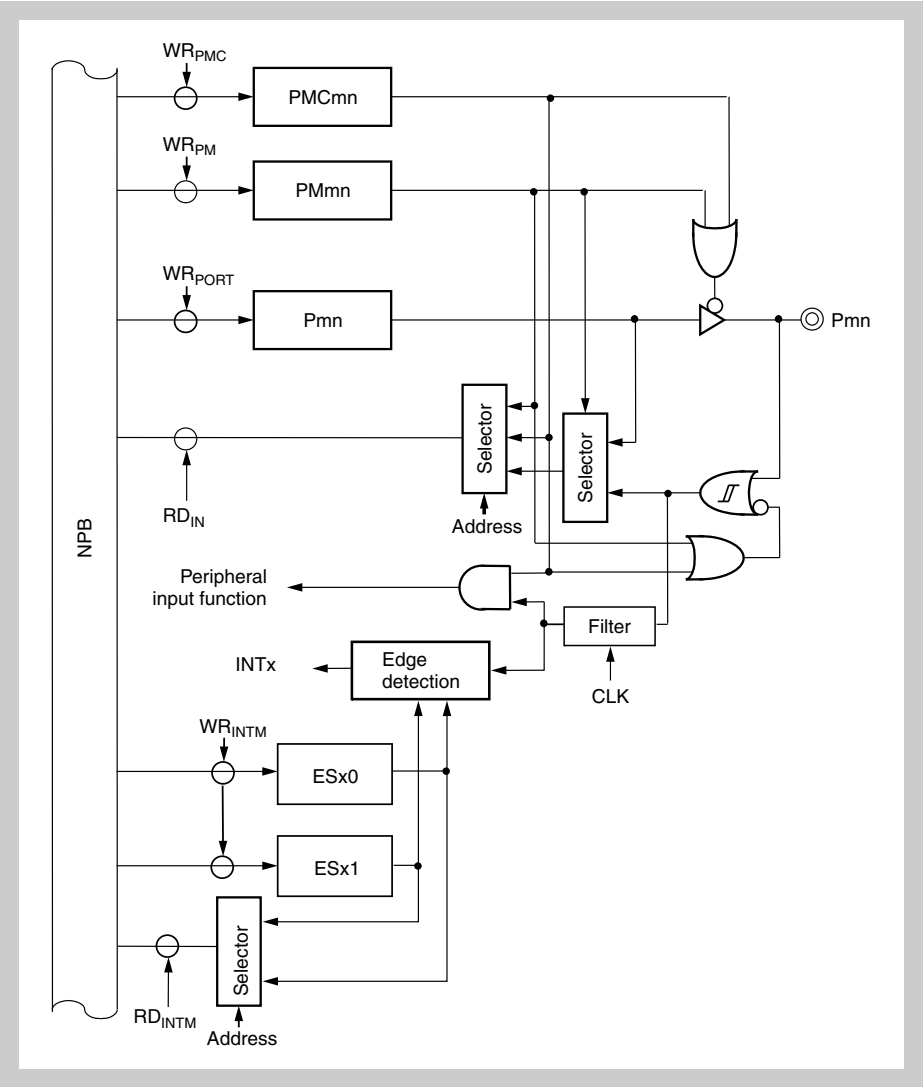


Figure 20-14 Port Type 8

Remark x: external interrupt number

(14) Port type 9

Port type 9 provides a general purpose I/O port with peripheral input function and external interrupt input capability. This type is similar to the port type 8, but input noise filter is bypassed for peripheral input function.

Note The peripheral input signal provided by port type 9 is fixed to high level, if peripheral input function is disabled.

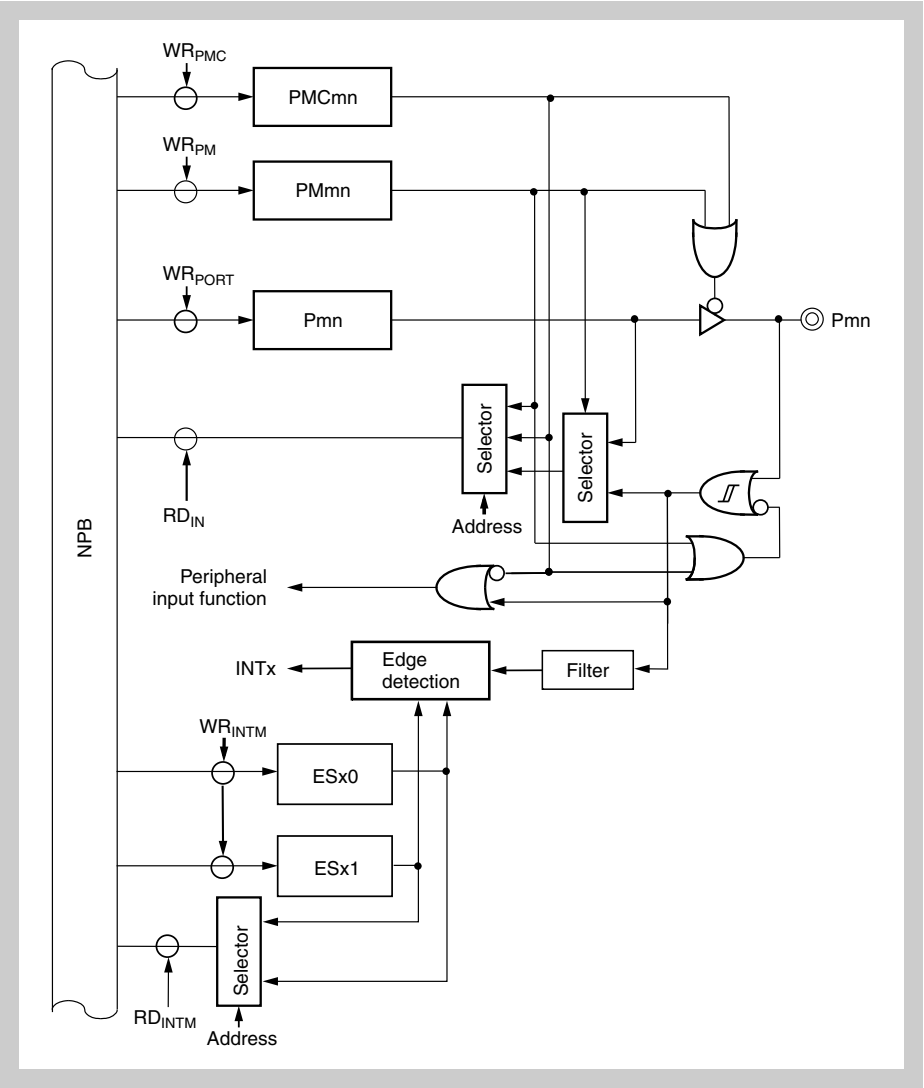


Figure 20-15 Port Type 9

Remark x: external interrupt number

(15) Port type 10

Port type10 provides a general purpose I/O port with digitally filtered peripheral input function.

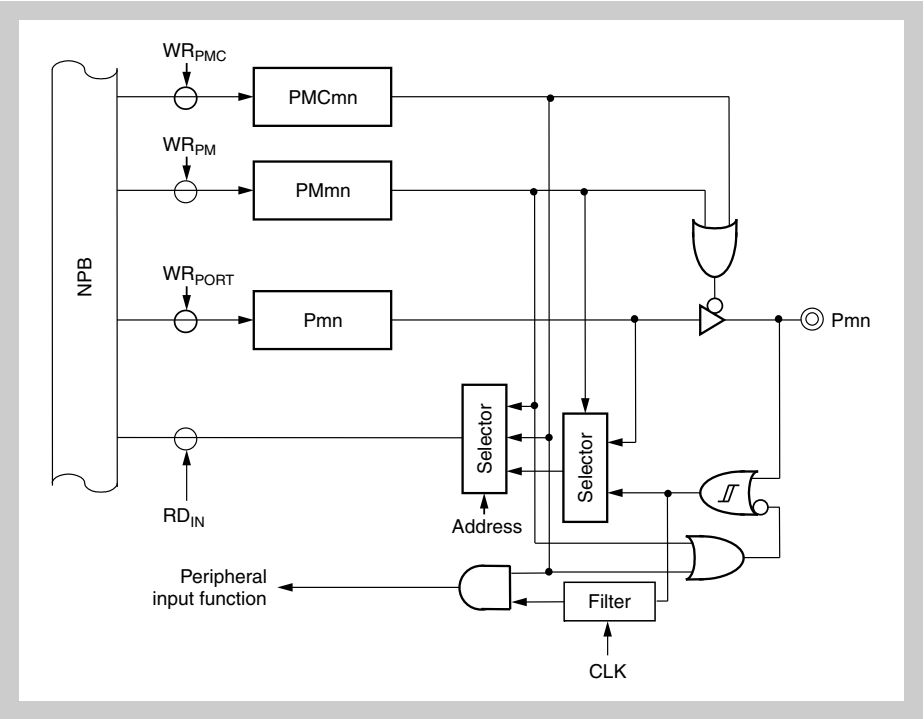


Figure 20-16 Port Type 10

(16) Port type 11

Port type 11 provides a general purpose I/O port with peripheral output function. This type is similar to the port type 6, but all port registers are write protected against unintended change due to system or software malfunction. Writing to the port registers of type 11 is only possible immediately after a write access to the PRCMD register.

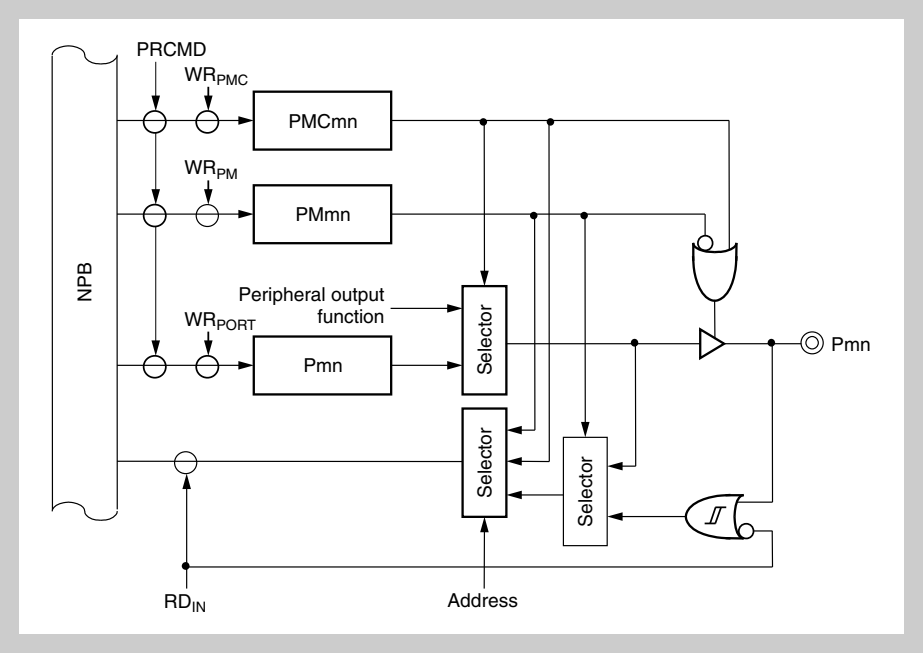


Figure 20-17 Port Type 11

(17) Port type 12

Port type 12 provides a general purpose I/O port with digitally filtered peripheral input function and peripheral output function. This type is similar to the port logic type 1S, but all port registers are write protected against unintended change due to system or software malfunction. Writing to the port registers of type 12 is only possible immediately after a write access to the PRCMD register.

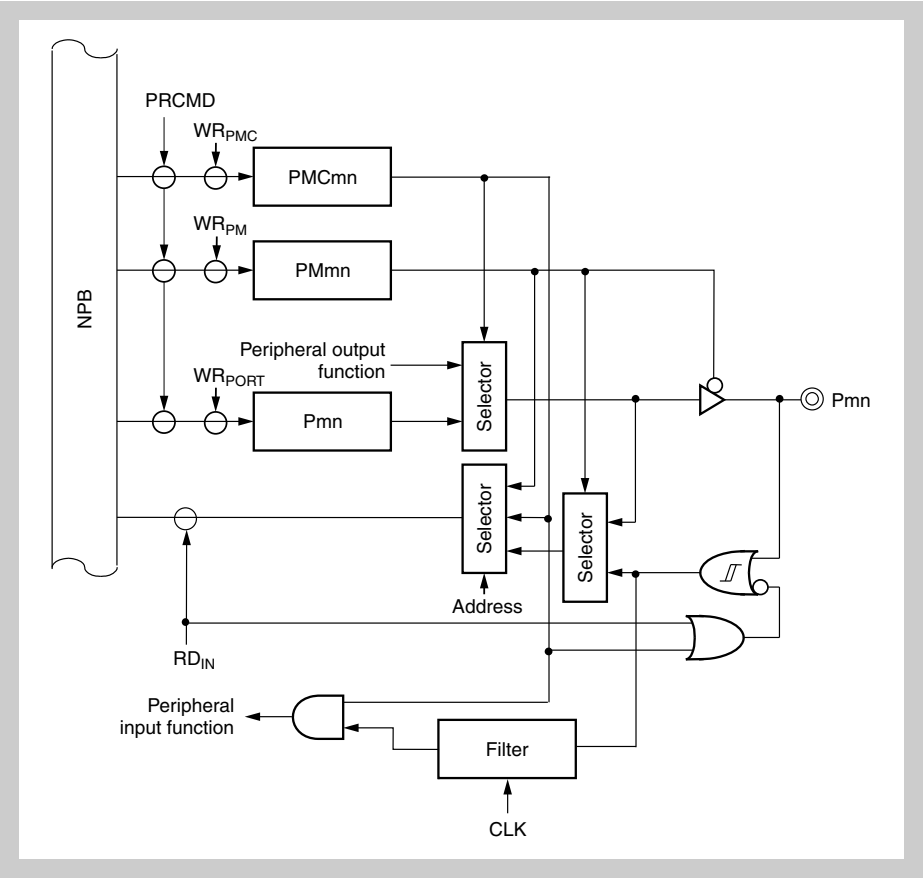


Figure 20-18 Port Type 12

(18) Port type 13

Port type 13 provides a general purpose I/O port with peripheral output function. This type is similar to the port logic type 11, but the output driver can be shut down immediately by the ESOx input signal (x = 0, 1). All port registers are write protected against unintended change due to system or software malfunction. Writing to the port registers of type 13 is only possible immediately after a write access to the PRCMD register.

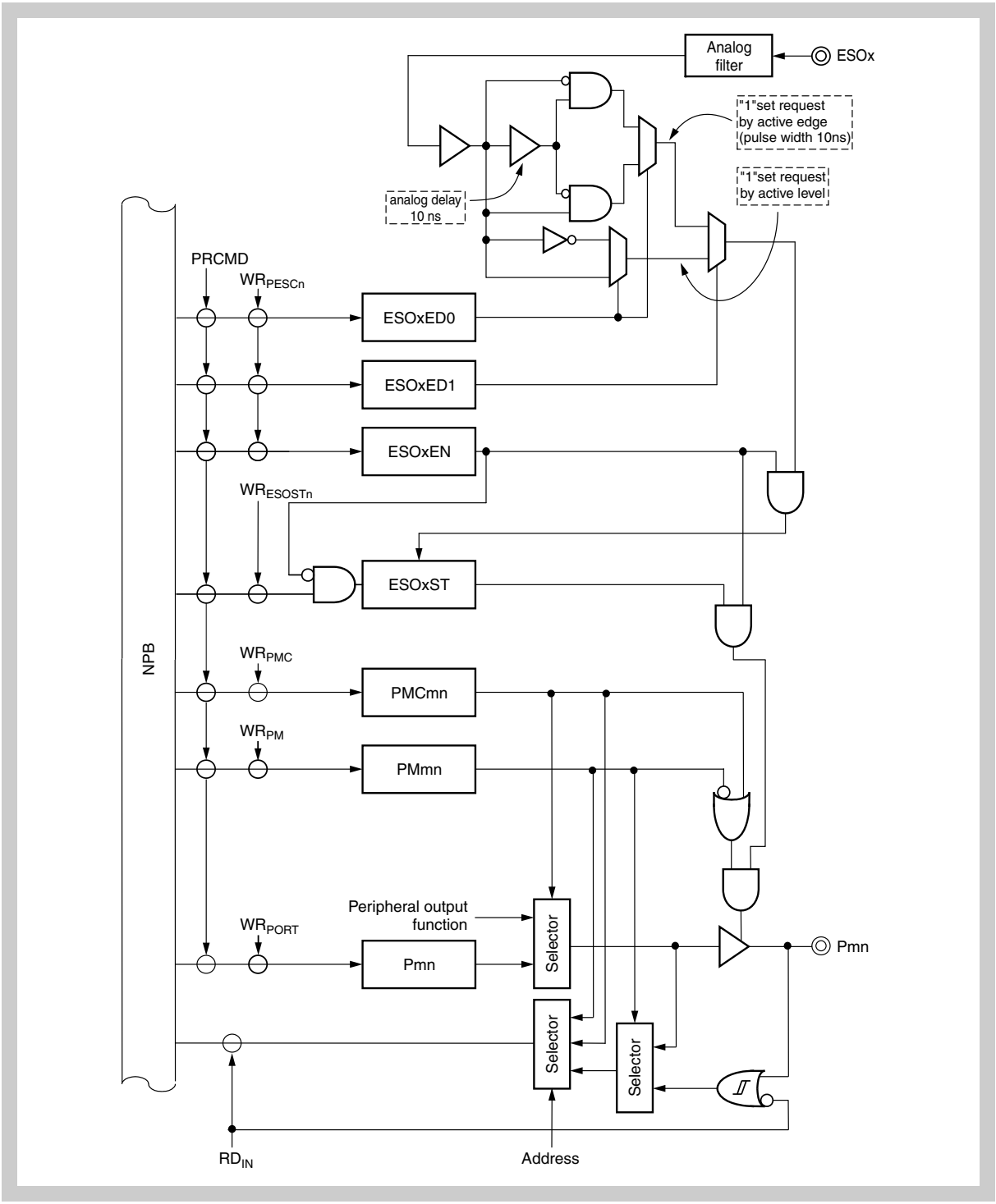


Figure 20-19 Port Type 13

(19) Port type 14

Port type 14 provides a general purpose I/O port with digitally filtered peripheral input function and peripheral output function. This type is similar to the port type 12, but the output driver can be shut down immediately by the ESOx input signal (x = 0, 1). All port registers are write protected against unintended change due to system or software malfunction. Writing to the port registers of type 13 is only possible immediately after a write access to the PRCMD register.

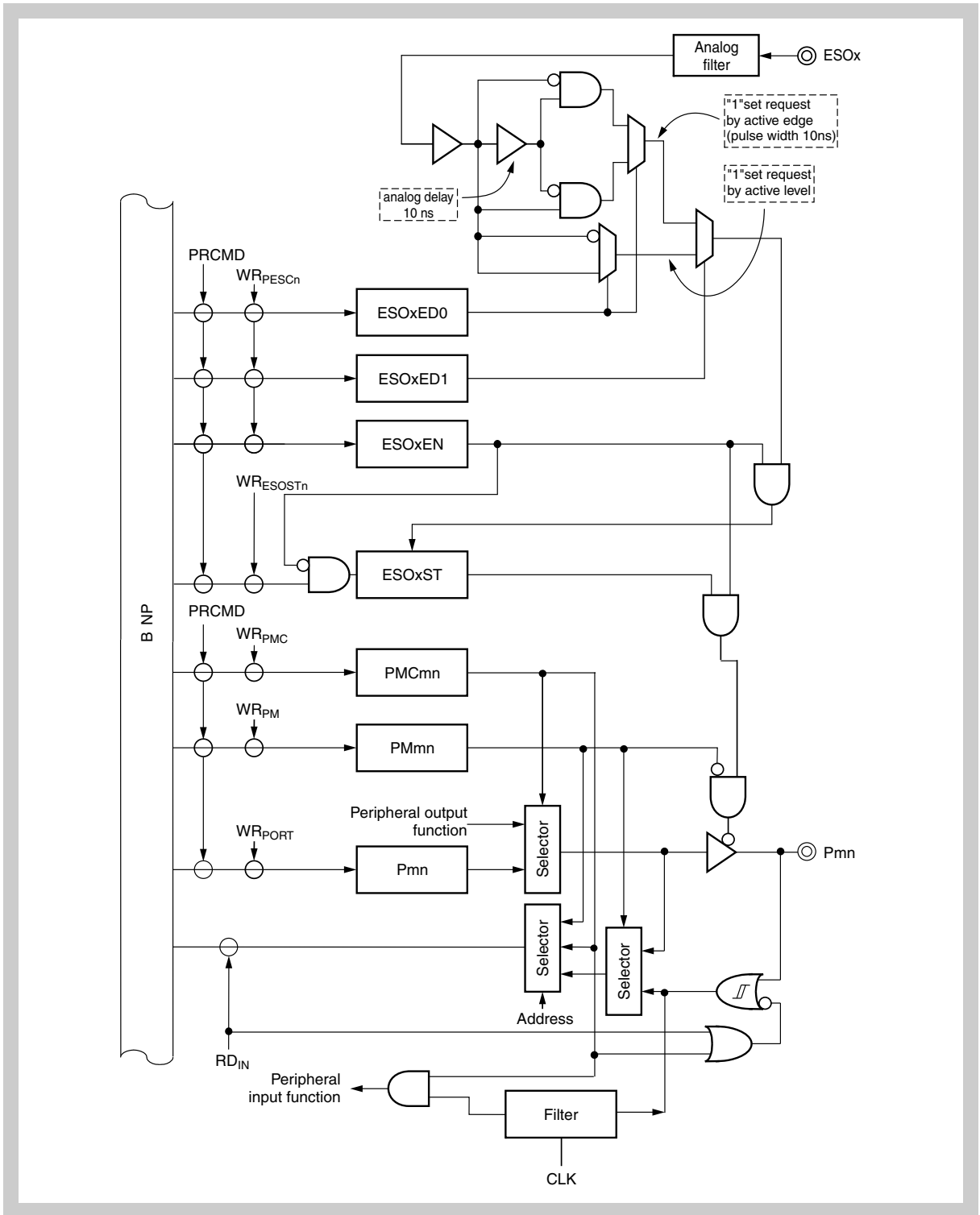


Figure 20-20 Port Type 14

(20) Port type 15

Port type 15 provides a general purpose input port with external interrupt input function. This type is similar as port type 3. Difference is the additional filtered peripheral input function support.

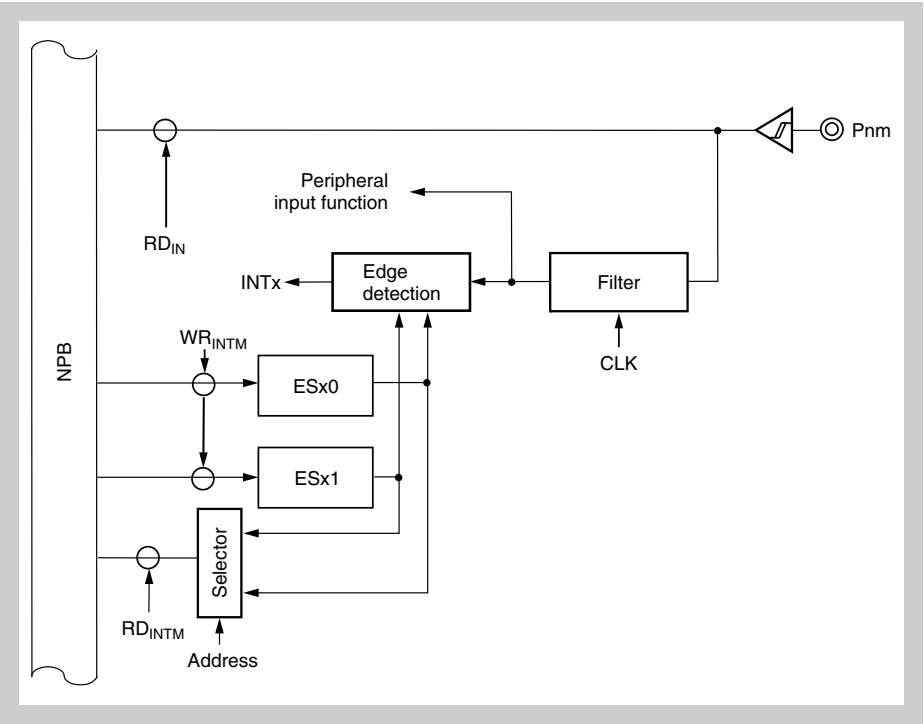


Figure 20-21 Port Type 15

Remark x: external interrupt number

(21) Port type 15A

Port type 15A provides a general purpose input port with external interrupt input function. This type is similar as port type 15. Difference is the analog filter instead of digital filter.

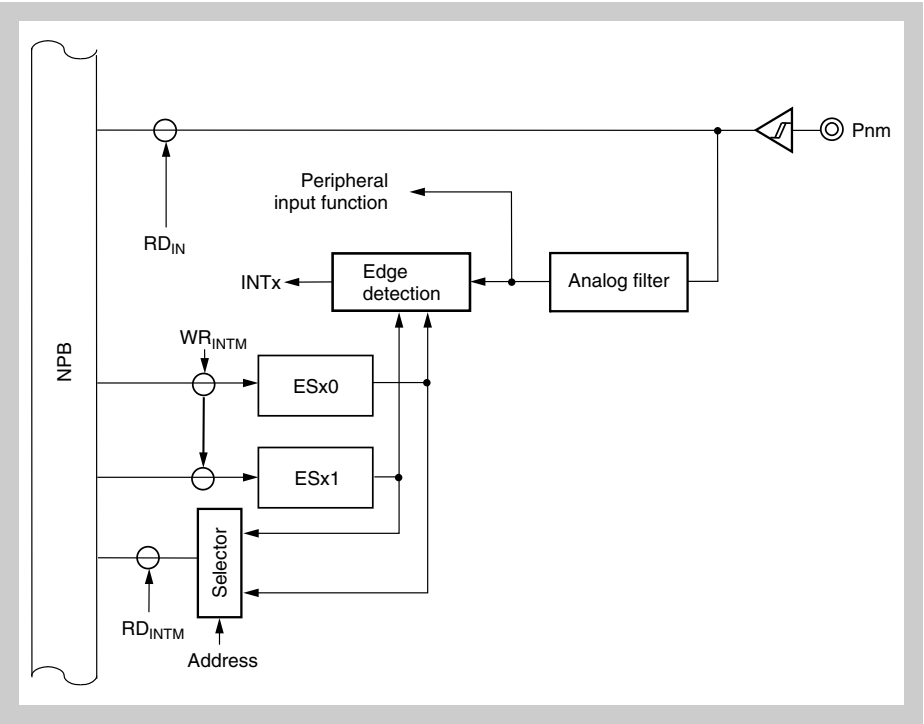


Figure 20-22 Port Type 15A

Remark x: external interrupt number

20.2.3 Peripheral registers of I/O ports

The following table lists the peripheral registers related to I/O ports.

Table 20-2 List of port registers (1/2)

Address	Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
$FFFFF000_H$	Port register port AL	PAL	R/W	-	-	×	0000_H
$FFFFF000_H$	Port register port AL low byte	PALL	R/W	×	×	-	00_H
$FFFFF001_H$	Port register port AL high byte	PALH	R/W	×	×	-	00_H
$FFFFF002_H$	Port register port AH	PAH	R/W	×	×	-	00_H
$FFFFF004_H$	Port register port DL	PDL	R/W	-	-	×	0000_H
$FFFFF004_H$	Port register port DL low byte	PDLL	R/W	×	×	-	00_H
$FFFFF005_H$	Port register port DL high byte	PDLH	R/W	×	×	-	00_H
$FFFFF006_H$	Port register port DH	PDH	R/W	-	-	×	0000_H
$FFFFF006_H$	Port register port DH low byte	PDHL	R/W	×	×	-	00_H
$FFFFF007_H$	Port register port DH high byte	PDHH	R/W	×	×	-	00_H
$FFFFF008_H$	Port register port CS	PCS	R/W	×	×	-	00_H
$FFFFF00A_H$	Port register port CT	PCT	R/W	×	×	-	00_H
$FFFFF00C_H$	Port register port CM	PCM	R/W	×	×	-	00_H
$FFFFF00E_H$	Port register port CD	PCD	R/W	×	×	-	00_H
$FFFFF020_H$	Port mode register AL	PMAL	R/W	-	-	×	$FFFF_H$
$FFFFF020_H$	Port mode register AL low byte	PMALL	R/W	×	×	-	FF_H
$FFFFF021_H$	Port mode register AL high byte	PMALH	R/W	×	×	-	FF_H
$FFFFF022_H$	Port mode register AH	PMAH	R/W	×	×	-	FF_H
$FFFFF024_H$	Port mode register DL	PMDL	R/W	-	-	×	$FFFF_H$
$FFFFF024_H$	Port mode register DL low byte	PMDLL	R/W	×	×	-	FF_H
$FFFFF025_H$	Port mode register DL high byte	PMDLH	R/W	×	×	-	FF_H
$FFFFF026_H$	Port mode register DH	PMDH	R/W	-	-	×	$FFFF_H$
$FFFFF026_H$	Port mode register DH low byte	PMDHL	R/W	×	×	-	FF_H
$FFFFF027_H$	Port mode register DH high byte	PMDHH	R/W	×	×	-	FF_H
$FFFFF028_H$	Port mode register Port mode CS	PMCS	R/W	×	×	-	FF_H
$FFFFF02A_H$	Port mode register Port mode CT	PMCT	R/W	×	×	-	FF_H
$FFFFF02C_H$	Port mode register Port mode CM	PMCM	R/W	×	×	-	FF_H
$FFFFF02E_H$	Port mode register Port mode CD	PMCD	R/W	×	×	-	FF_H
$FFFFF040_H$	Port mode control register AL	PMCAL	R/W	-	-	×	0000_H
$FFFFF040_H$	Port mode control register AL low byte	PMCALL	R/W	×	×	-	00_H
$FFFFF041_H$	Port mode control register AL high byte	PMCALH	R/W	×	×	-	00_H
$FFFFF042_H$	Port mode control register AH	PMCAH	R/W	×	×	-	00_H
$FFFFF044_H$	Port mode control register DL	PMCDL	R/W	-	-	×	0000_H
$FFFFF044_H$	Port mode control register DL low byte	PMCDLL	R/W	×	×	-	00_H
$FFFFF045_H$	Port mode control register DL high byte	PMCDLH	R/W	×	×	-	00_H

Table 20-2 List of port registers (2/2)

Address	Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
$FFFFF046_H$	Port mode control register DH	PMCDH	R/W	-	-	×	0000_H
$FFFFF046_H$	Port mode control register DH low byte	PMCDHL	R/W	×	×	-	00_H
$FFFFF047_H$	Port mode control register DH high byte	PMCDHH	R/W	×	×	-	00_H
$FFFFF048_H$	Port mode control register CS	PMCCS	R/W	×	×	-	00_H
$FFFFF04A_H$	Port mode control register CT	PMCCT	R/W	×	×	-	00_H
$FFFFF04C_H$	Port mode control register CM	PMCCM	R/W	×	×	-	00_H
$FFFFF04E_H$	Port mode control register CD	PMCCD	R/W	×	×	-	00_H
$FFFFF400_H$	Port register port 0	P0	R	×	×	-	undef.
$FFFFF402_H$	Port register port 1	P1	R/W	×	×	-	undef.
$FFFFF404_H$	Port register port 2	P2	R/W	×	×	-	undef.
$FFFFF406_H$	Port register port 3	P3	R/W	×	×	-	undef.
$FFFFF408_H$	Port register port 4	P4	R/W	×	×	-	undef.
$FFFFF40A_H$	Port register port 5	P5	R/W	×	×	-	undef.
$FFFFF40C_H$	Port register port 6	P6	R/W	×	×	-	undef.
$FFFFF40E_H$	Port register port 7	P7	R/W	×	×	-	undef.
$FFFFF410_H$	Port register port 8	P8	R/W	×	×	-	undef.
$FFFFF412_H$	Port register port 9	P9	R/W	×	×	-	undef.
$FFFFF414_H$	Port register port 10	P10	R/W	×	×	-	undef.
$FFFFF422_H$	Port mode register 1	PM1	R/W	×	×	-	FF_H
$FFFFF424_H$	Port mode register 2	PM2	R/W	×	×	-	FF_H
$FFFFF426_H$	Port mode register 3	PM3	R/W	×	×	-	FF_H
$FFFFF428_H$	Port mode register 4	PM4	R/W	×	×	-	FF_H
$FFFFF42A_H$	Port mode register 5	PM5	R/W	×	×	-	FF_H
$FFFFF42C_H$	Port mode register 6	PM6	R/W	×	×	-	FF_H
$FFFFF42E_H$	Port mode register 7	PM7	R/W	×	×	-	FF_H
$FFFFF430_H$	Port mode register 8	PM8	R/W	×	×	-	FF_H
$FFFFF432_H$	Port mode register 9	PM9	R/W	×	×	-	FF_H
$FFFFF434_H$	Port mode register 10	PM10	R/W	×	×	-	FF_H
$FFFFF442_H$	Port mode control register 1	PMC1	R/W	×	×	-	00_H
$FFFFF444_H$	Port mode control register 2	PMC2	R/W	×	×	-	00_H
$FFFFF446_H$	Port mode control register 3	PMC3	R/W	×	×	-	00_H
$FFFFF448_H$	Port mode control register 4	PMC4	R/W	×	×	-	00_H
$FFFFF44A_H$	Port mode control register 5	PMC5	R/W	×	×	-	00_H
$FFFFF44C_H$	Port mode control register 6	PMC6	R/W	×	×	-	00_H
$FFFFF44E_H$	Port mode control register 7	PMC7	R/W	×	×	-	00_H
$FFFFF450_H$	Port mode control register 8	PMC8	R/W	×	×	-	00_H
$FFFFF452_H$	Port mode control register 9	PMC9	R/W	×	×	-	00_H
$FFFFF454_H$	Port mode control register 10	PMC10	R/W	×	×	-	00_H

20.2.4 Peripheral registers of valid edge control

The following table lists the peripheral registers related to valid edge control.

Table 20-3 List of valid edge control registers

Address	Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 Bit	8 Bits	16 Bits	
$FFFFF880_H$	Interrupt mode register 0	INTM0	R/W	×	×	-	00_H
$FFFFF882_H$	Interrupt mode register 1	INTM1	R/W	×	×	-	00_H
$FFFFF884_H$	Interrupt mode register 2	INTM2	R/W	×	×	-	00_H
$FFFFF886_H$	Interrupt mode register 3	INTM3	R/W	×	×	-	00_H
$FFFFF888_H$	Port emergency shut off control register 5	PESC5	R/W	×	×	-	00_H
$FFFFF88A_H$	Port emergency shut off control register 5	ESOST5	R/W	×	×	-	00_H
$FFFFF88C_H$	Port emergency shut off status register 6	PESC6	R/W	×	×	-	00_H
$FFFFF88E_H$	Port emergency shut off status register 6	ESOST6	R/W	×	×	-	00_H

20.3 Port Pin Functions

20.3.1 Port 0

Port 0 is a 5-bit input only port.

(1) Functions

- Input data can be read in 1-bit units by using the port register 0 (P0).
- The alternate functions shared with the input port functionality of port 0 are always enabled.

Table 20-4 Alternate function pins and port types of port 0

Port		Alternate Function	Remark	Port Type
Port 0	P00	NMI	Non maskable interrupt	3
	P01	INTP0, ESO0	External interrupt request input, Emergency output shut off input (TMR0)	15A
	P02	INTP1, ESO1	External interrupt request input, Emergency output shut off input (TMR1)	
	P03	INTP2, ADTRG0	External interrupt request input, External A/D conversion start trigger (ADC0)	15
	P04	INTP3, ADTRG1	External interrupt request input, External A/D conversion start trigger (ADC1)	

(2) Port register 0 (P0)

The port register 0 (P0) is an 8-bit register that reflects the input levels of port pins P00 to P04.

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

Address $FFFFF400_H$

Initial Value Undefined

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

Table 20-5 P0 register contents

Bit position	Bit name	Function
4...0	P0[4:0]	Input data control of pin P0n. 0: Low level is input 1: High level is input **Note:** n = 0 to 4

20.3.2 Port 1

Port 1 is an 8-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register 1 (P1).
- Input or output mode can be set in 1-bit units by using the port mode register 1 (PM1).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register 1 (PMC1).

Table 20-6 Alternate function pins and port types of port 1

Port		Alternate Function	Remark	Port Type
Port 1	P10	TIP00, TEVTP1 TOP00	Timer input (TMP0/TMP1) Timer output (TMP0)	6
	P11	TIP01, TTRGP1 TOP01	Timer input (TMP0/TMP1) Timer output (TMP0)	
	P12	TIP10, TTRGP0 TOP10	Timer input (TMP0/TMP1) Timer output (TMP1)	
	P13	TIP11, TEVTP0 TOP11	Timer input (TMP0/TMP1) Timer output (TMP1)	
	P14	TIP20, TEVTP3 TOP20	Timer input (TMP2/TMP3) Timer output (TMP2)	
	P15	TIP21, TTRGP3 TOP21	Timer input (TMP2/TMP3) Timer output (TMP2)	
	P16	TIP30, TTRGP2 TOP30	Timer input (TMP2/TMP3) Timer output (TMP3)	
	P17	TIP31, TEVTP2 TOP31	Timer input (TMP2/TMP3) Timer output (TMP3)	

(2) Port register 1 (P1)

The P1 register 1 is an 8-bit register that controls reading the pin levels and writing the output levels of port pins P10 to P17.

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

Address $FFFFF402_H$

Initial Value Undefined

	7	6	5	4	3	2	1	0
P1	P17	P16	P15	P14	P13	P12	P11	P10
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-7 P1 register contents

Bit position	Bit name	Function
7...0	P1[7:0]	Input data control of pin P1n (in port mode PMC1n = 0). 0: Input mode (PM1n = 1): Low level is input Output mode (PM1n = 0): Low level is output 1: Input mode (PM1n = 1): High level is input Output mode (PM1n = 0): High level is output **Note:** n = 0 to 7

(3) Port mode register 1 (PM1)

The PM1 register is an 8-bit register that specifies the input or output mode.

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

Address $FFFFF422_H$

Initial Value FF_H

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

Table 20-8 PM1 register contents

Bit position	Bit name	Function
7...0	PM1[7:0]	Input/output mode control of pin P1n (in port mode PMC1n = 0). 0: Output mode 1: Input mode **Note:** n = 0 to 7

(4) Port mode control register 1 (PMC1)

The PMC1 register is an 8-bit register that specifies the port mode or control mode (alternate function).

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

Address $FFFFF442_H$

Initial Value 00_H

	7	6	5	4	3	2	1	0
PMC1	PMC17	PMC16	PMC15	PMC14	PMC13	PMC12	PMC11	PMC10
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-9 PMC1 register contents (1/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7</td><td>PMC17</td><td>Port mode/control mode specification of pin P17
<table>
<tr><th>PMC17</th><th>PM17</th><th colspan="2">Function of pin P17</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOP31 output mode</td></tr>
<tr><td>1</td><td>TIP31, TEVTP2 input mode</td></tr>
</table></td></tr>
<tr><td>6</td><td>PMC16</td><td>Port mode/control mode specification of pin P16
<table>
<tr><th>PMC16</th><th>PM16</th><th colspan="2">Function of pin P16</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOP30 output mode</td></tr>
<tr><td>1</td><td>TIP30, TTRGP2 input mode</td></tr>
</table></td></tr>
<tr><td>5</td><td>PMC15</td><td>Port mode/control mode specification of pin P15
<table>
<tr><th>PMC15</th><th>PM15</th><th colspan="2">Function of pin P15</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOP21 output mode</td></tr>
<tr><td>1</td><td>TIP21, TTRGP3 input mode</td></tr>
</table></td></tr>
<tr><td>4</td><td>PMC14</td><td>Port mode/control mode specification of pin P14
<table>
<tr><th>PMC14</th><th>PM14</th><th colspan="2">Function of pin P14</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOP20 output mode</td></tr>
<tr><td>1</td><td>TIP20, TEVTP3 input mode</td></tr>
</table></td></tr>
</table>

Table 20-9 PMC1 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3</td><td>PMC13</td><td>Port mode/control mode specification of pin P13
<table>
<tr><th>PMC13</th><th>PM13</th><th colspan="2">Function of pin P13</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOP11 output mode</td></tr>
<tr><td>1</td><td>TIP11, TEVTP0 input mode</td></tr>
</table></td></tr>
<tr><td>2</td><td>PMC12</td><td>Port mode/control mode specification of pin P12
<table>
<tr><th>PMC12</th><th>PM12</th><th colspan="2">Function of pin P12</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOP10 output mode</td></tr>
<tr><td>1</td><td>TIP10, TTRGP0 input mode</td></tr>
</table></td></tr>
<tr><td>1</td><td>PMC11</td><td>Port mode/control mode specification of pin P11
<table>
<tr><th>PMC12</th><th>PM12</th><th colspan="2">Function of pin P11</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOP01 output mode</td></tr>
<tr><td>1</td><td>TIP01, TTRGP1 input mode</td></tr>
</table></td></tr>
<tr><td>0</td><td>PMC10</td><td>Port mode/control mode specification of pin P10
<table>
<tr><th>PMC12</th><th>PM12</th><th colspan="2">Function of pin P10</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOP00 output mode</td></tr>
<tr><td>1</td><td>TIP00, TEVTP1 input mode</td></tr>
</table></td></tr>
</table>

20.3.3 Port 2

Port 2 is an 8-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register 2 (P2).
- Input or output mode can be set in 1-bit units by using the port mode register 2 (PM2).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register 2 (PMC2).

Table 20-10 Alternate function pins and port types of port 2

Port		Alternate Function	Remark	Port Type
Port 2	P20	TIP40, TEVTP5 TOP40	Timer input (TMP4/TMP5) Timer output (TMP4 output)	6
	P21	TIP41, TTRGP5 TOP41	Timer input (TMP4/TMP5) Timer output (TMP4 output)	
	P22	TIP50, TTRGP4 TOP50	Timer input (TMP4/TMP5) Timer output (TMP5 output)	
	P23	TIP51, TEVTP4 TOP51	Timer input (TMP4/TMP5) Timer output (TMP5 output)	
	P24	TIP60, TEVTP7 TOP60	Timer input (TMP6/TMP7) Timer output (TMP6 output)	
	P25	TIP61, TTRGP7 TOP61	Timer input (TMP6/TMP7) Timer output (TMP6 output)	
	P26	TIP70, TTRGP6 TOP70	Timer input (TMP6TMP7) Timer output (TMP7 output)	
	P27	TIP71, TEVTP6 TOP71	Timer input (TMP6/TMP7) Timer output (TMP7 output)	

(2) Port register 2 (P2)

The P2 register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins P20 to P27.

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

Address $FFFFF404_H$

Initial Value Undefined

	7	6	5	4	3	2	1	0
P2	P27	P26	P25	P24	P23	P22	P21	P20
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-11 P2 register contents

Bit position	Bit name	Function
7...0	P2[7:0]	Input data control of pin P2n (in port mode PMC2n = 0). 0: Input mode (PM2n = 1): Low level is input Output mode (PM2n = 0): Low level is output 1: Input mode (PM2n = 1): High level is input Output mode (PM2n = 0): High level is output **Note:** n = 0 to 7

(3) Port mode register 2 (PM2)

The PM2 register is an 8-bit register that specifies the input or output mode.

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

Address $FFFFF424_H$

Initial Value FF_H

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

Table 20-12 PM2 register contents

Bit position	Bit name	Function
7...0	PM2[7:0]	Input/output mode control of pin P2n (in port mode PMC2n = 0). 0: Output mode 1: Input mode **Note:** n = 0 to 7

(4) Port mode control register 2 (PMC2)

The PMC2 register is an 8-bit register that specifies the port mode or control mode (alternate function).

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

Reset input sets this register to 00H.

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

Address $FFFFF444_H$

Initial Value 00_H

	7	6	5	4	3	2	1	0
PMC2	PMC27	PMC26	PMC25	PMC24	PMC23	PMC22	PMC21	PMC20
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-13 PMC2 register contents (1/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7</td><td>PMC27</td><td>Port mode/control mode specification of pin P27
<table>
<tr><th>PMC27</th><th>PM27</th><th colspan="2">Function of pin P27</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOP71 output mode</td></tr>
<tr><td>1</td><td>TIP71, TEVTP6 input mode</td></tr>
</table></td></tr>
<tr><td>6</td><td>PMC26</td><td>Port mode/control mode specification of pin P26
<table>
<tr><th>PMC26</th><th>PM26</th><th colspan="2">Function of pin P26</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOP70 output mode</td></tr>
<tr><td>1</td><td>TIP70, TTRGP6 input mode</td></tr>
</table></td></tr>
<tr><td>5</td><td>PMC25</td><td>Port mode/control mode specification of pin P25
<table>
<tr><th>PMC25</th><th>PM25</th><th colspan="2">Function of pin P25</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOP61 output mode</td></tr>
<tr><td>1</td><td>TIP61, TTRGP7 input mode</td></tr>
</table></td></tr>
<tr><td>4</td><td>PMC24</td><td>Port mode/control mode specification of pin P24
<table>
<tr><th>PMC24</th><th>PM24</th><th colspan="2">Function of pin P24</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOP60 output mode</td></tr>
<tr><td>1</td><td>TIP60, TEVTP7 input mode</td></tr>
</table></td></tr>
</table>

Table 20-13 PMC2 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3</td><td>PMC23</td><td>Port mode/control mode specification of pin P23
<table>
<tr><th>PMC23</th><th>PM23</th><th colspan="2">Function of pin P23</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOP51 output mode</td></tr>
<tr><td>1</td><td>TIP51, TEVTP4 input mode</td></tr>
</table></td></tr>
<tr><td>2</td><td>PMC22</td><td>Port mode/control mode specification of pin P22
<table>
<tr><th>PMC22</th><th>PM22</th><th colspan="2">Function of pin P22</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOP50 output mode</td></tr>
<tr><td>1</td><td>TIP50, TTRGP4 input mode</td></tr>
</table></td></tr>
<tr><td>1</td><td>PMC21</td><td>Port mode/control mode specification of pin P21
<table>
<tr><th>PMC22</th><th>PM22</th><th colspan="2">Function of pin P21</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOP41 output mode</td></tr>
<tr><td>1</td><td>TIP41, TTRGP5 input mode</td></tr>
</table></td></tr>
<tr><td>0</td><td>PMC20</td><td>Port mode/control mode specification of pin P20
<table>
<tr><th>PMC22</th><th>PM22</th><th colspan="2">Function of pin P20</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOP40 output mode</td></tr>
<tr><td>1</td><td>TIP40, TEVTP5 input mode</td></tr>
</table></td></tr>
</table>

20.3.4 Port 3

Port 3 is an 8-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register 3 (P3).
- Input or output mode can be set in 1-bit units by using the port mode register 3 (PM3).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register 3 (PMC3).
- The external interrupt request inputs shared with the input port functionality of port 3 are always enabled in input port mode.

Table 20-14 Alternate function pins and port types of port 3

Port		Alternate Function	Remark	Port Type
Port 3	P30	RXDC0 INTP4	Serial interface (UARTC0) input External interrupt request input	9
	P31	TXDC0	Serial interface (UARTC0) output	1S
	P32	RXDC1 INTP5	Serial interface (UARTC1) input External interrupt request input	9
	P33	TXDC1	Serial interface (UARTC1) output	1S
	P34	FCRXD0	FCAN0 input	2A
	P35	FCTXD0	FCAN0 output	1S
	P36	FCRXD1 [a]	FCAN1 input [a]	2A
	P37	FCTXD1 [a]	FCAN1 output [a]	1S

a) Alternate function not available on µPD70F3447.

(2) Port register 3 (P3)

The P3 register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins P30 to P37.

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

Address $FFFFF406_H$

Initial Value Undefined

	7	6	5	4	3	2	1	0
P3	P37	P36	P35	P34	P33	P32	P31	P30
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-15 P3 register contents

Bit position	Bit name	Function
7...0	P3[7:0]	Input data control of pin P3n (in port mode PMC3n = 0). 0: Input mode (PM3n = 1): Low level is input Output mode (PM3n = 0): Low level is output 1: Input mode (PM3n = 1): High level is input Output mode (PM3n = 0): High level is output **Note:** n = 0 to 7

(3) Port mode register 3 (PM3)

The PM3 register is an 8-bit register that specifies the input or output mode.

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

Address $FFFFF426_H$

Initial Value FF_H

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

Table 20-16 PM3 register contents

Bit position	Bit name	Function
7...0	PM3[7:0]	Input/output mode control of pin P3n (in port mode PMC3n = 0). 0: Output mode 1: Input mode **Note:** n = 0 to 7

(4) Port mode control register 3 (PMC3)

The PMC3 register is an 8-bit register that specifies the port mode or control mode (alternate function).

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

Address $FFFFF446_H$

Initial Value 00_H

Caution On μPD70F3447 do not set bits PMC34 and PMC36 to 1, since the corresponding alternate function is not available.

PMC3

7	6	5	4	3	2	1	0
PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-17 PMC3 register contents (1/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7</td><td>PMC37</td><td>Port mode/control mode specification of pin P37
<table>
<tr><th>PMC37</th><th>PM37</th><th colspan="2">Function of pin P37</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td>1</td><td>×</td><td>Alternate function</td><td>FCTXD1 output mode</td></tr>
</table>
Note: Alternate function not available on μPD70F3447.</td></tr>
<tr><td>6</td><td>PMC36</td><td>Port mode/control mode specification of pin P36 [a]
<table>
<tr><th>PMC36</th><th>PM36</th><th colspan="2">Function of pin P36</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td>1</td><td>×</td><td>Alternate function</td><td>FCRXD1 input mode</td></tr>
</table>
Note: Alternate function not available on μPD70F3447.</td></tr>
<tr><td>5</td><td>PMC35</td><td>Port mode/control mode specification of pin P35
<table>
<tr><th>PMC35</th><th>PM35</th><th colspan="2">Function of pin P35</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td>1</td><td>×</td><td>Alternate function</td><td>FCTXD0 output mode</td></tr>
</table></td></tr>
<tr><td>4</td><td>PMC34</td><td>Port mode/control mode specification of pin P34 [a]
<table>
<tr><th>PMC34</th><th>PM34</th><th colspan="2">Function of pin P34</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td>1</td><td>×</td><td>Alternate function</td><td>FCRXD0 input mode</td></tr>
</table></td></tr>
</table>

Table 20-17 PMC3 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3</td><td>PMC33</td><td>Port mode/control mode specification of pin P33
<table>
<tr><th>PMC33</th><th>PM33</th><th colspan="2">Function of pin P33</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td>1</td><td>×</td><td>Alternate function</td><td>TXDC1 output mode</td></tr>
</table></td></tr>
<tr><td>2</td><td>PMC32</td><td>Port mode/control mode specification of pin P32 [a]
<table>
<tr><th>PMC32</th><th>PM32</th><th colspan="2">Function of pin P32</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td>Input port mode</td><td>External interrupt request input mode (INTP5)</td></tr>
<tr><td>1</td><td>×</td><td>Alternate function</td><td>RXDC1 input mode,
External interrupt request input mode (INTP5)</td></tr>
</table></td></tr>
<tr><td>1</td><td>PMC31</td><td>Port mode/control mode specification of pin P31
<table>
<tr><th>PMC31</th><th>PM31</th><th colspan="2">Function of pin P31</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td>1</td><td>×</td><td>Alternate function</td><td>TXDC0 output mode</td></tr>
</table></td></tr>
<tr><td>0</td><td>PMC30</td><td>Port mode/control mode specification of pin P30 [a]
<table>
<tr><th>PMC30</th><th>PM30</th><th colspan="2">Function of pin P30</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td>Input port mode</td><td>External interrupt request input mode (INTP4)</td></tr>
<tr><td>1</td><td>×</td><td>Alternate function</td><td>Input mode,
External interrupt request input mode (INTP4)</td></tr>
</table></td></tr>
</table>

a) If this pin is set to port mode, the corresponding peripheral input signal (alternate function) is forced to high level internally.

20.3.5 Port 4

Port 4 is a 6-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register 4 (P4).
- Input or output mode can be set in 1-bit units by using the port mode register 4 (PM4).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register 4 (PMC4).

Table 20-18 Alternate function pins and port types of port 4

Port		Alternate Function	Remark	Port Type
Port 4	P40	SIB0	Serial interface (CSIB0) input	2
	P41	SOB0	Serial interface (CSIB0) output	1E
	P42	SCKB0	Serial interface (CSIB0) I/O	4C
	P43	SIB1 [a]	Serial interface (CSIB1) input [a]	2
	P44	SOB1 [a]	Serial interface (CSIB1) output [a]	1E
	P45	SCKB1 [a]	Serial interface (CSIB1) I/O [a]	4C

a) Alternate function not available on μPD70F3447.

(2) Port register 4 (P4)

The P4 register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins P40 to P45.

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

Address $FFFFF408_H$

Initial Value Undefined

	7	6	5	4	3	2	1	0
P4	0	0	P45	P44	P43	P42	P41	P40
	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-19 P4 register contents

Bit position	Bit name	Function
5...0	P4[5:0]	Input data control of pin P4n (in port mode PMC4n = 0). 0: Input mode (PM4n = 1): Low level is input Output mode (PM4n = 0): Low level is output 1: Input mode (PM4n = 1): High level is input Output mode (PM4n = 0): High level is output **Note:** n = 0 to 5

(3) Port mode register 4 (PM4)

The PM4 register is an 8-bit register that specifies the input or output mode.

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

Address $FFFFF428_H$

Initial Value FF_H

	7	6	5	4	3	2	1	0
PM4	1	1	PM45	PM44	PM43	PM42	PM41	PM40
	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-20 PM4 register contents

Bit position	Bit name	Function
5...0	PM4[5:0]	Input/output mode control of pin P4n (in port mode PMC4n = 0). 0: Output mode 1: Input mode **Note:** n = 0 to 5

(4) Port mode control register 4 (PMC4)

The PMC4 register is an 8-bit register that specifies the port mode or control mode (alternate function).

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

Address $FFFFF448_H$

Initial Value 00_H

Caution On µPD70F3447 do not set bits PMC4[5:3] to 1, since the corresponding alternate function is not available.

	7	6	5	4	3	2	1	0
PMC4	0	0	PMC45	PMC44	PMC43	PMC42	PMC41	PMC40
	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-21 PMC4 register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>5</td><td>PMC45</td><td>Port mode/control mode specification of pin P45
<table>
<tr><th>PMC45</th><th>PM45</th><th colspan="2">Function of pin P45</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td>1</td><td>×</td><td>Alternate function</td><td>SCKB1 I/O mode (input or output mode controlled by CSIB1)</td></tr>
</table>
Note: Alternate function not available on μPD70F3447.</td></tr>
<tr><td>4</td><td>PMC44</td><td>Port mode/control mode specification of pin P44
<table>
<tr><th>PMC44</th><th>PM44</th><th colspan="2">Function of pin P44</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td>1</td><td>×</td><td>Alternate function</td><td>SOB1 output mode</td></tr>
</table>
Note: Alternate function not available on μPD70F3447.</td></tr>
<tr><td>3</td><td>PMC43</td><td>Port mode/control mode specification of pin P43
<table>
<tr><th>PMC43</th><th>PM43</th><th colspan="2">Function of pin P43</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td>1</td><td>×</td><td>Alternate function</td><td>SIB1input mode</td></tr>
</table>
Note: Alternate function not available on μPD70F3447.</td></tr>
<tr><td>2</td><td>PMC42</td><td>Port mode/control mode specification of pin P42
<table>
<tr><th>PMC42</th><th>PM42</th><th colspan="2">Function of pin P42</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td>1</td><td>×</td><td>Alternate function</td><td>SCKB0 I/O mode (input or output mode controlled by CSIB0)</td></tr>
</table></td></tr>
<tr><td>1</td><td>PMC41</td><td>Port mode/control mode specification of pin P41
<table>
<tr><th>PMC41</th><th>PM41</th><th colspan="2">Function of pin P41</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td>1</td><td>×</td><td>Alternate function</td><td>SOB0 output mode</td></tr>
</table></td></tr>
<tr><td>0</td><td>PMC40</td><td>Port mode/control mode specification of pin P40
<table>
<tr><th>PMC40</th><th>PM40</th><th colspan="2">Function of pin P40</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td>Input port mode</td><td>External interrupt request input mode (INTP4)</td></tr>
<tr><td>1</td><td>×</td><td>Alternate function</td><td>SIB0 input mode</td></tr>
</table></td></tr>
</table>

20.3.6 Port 5

Port 5 is an 8-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register 5 (P5).
- Input or output mode can be set in 1-bit units by using the port mode register 5 (PM5).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register 5 (PMC5).
- Emergency shut off by ES0 input signal of output buffers P51 to P56 can be controlled by port emergency shut off control register 5 (PESC5) and emergency shut off status register 5 (ESOST5).
- Security feature to protect the timer output signals of TMR0 from unintended CPU interference. Registers P5, PM5, PMC5, PESC5 and ESOST5 can only be written in a special sequence.

Table 20-22 Alternate function pins and port types of port 5

Port		Alternate Function	Remark	Port Type
Port 5	P50	TOR00	Timer output (TMR0)	11
	P51	TOR01		13
	P52	TOR02		
	P53	TOR03		
	P54	TOR04		
	P55	TOR05		
	P56	TOR06		
	P57	TOR07		11

(2) Port register 5 (P5)

The P5 register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins P50 to P57.

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

Address $FFFFF40A_H$

Initial Value Undefined

Note Writing to the P5 register is only possible in a specific sequence, where a write access to the command register (PRCMD) must be made before a write access to the P5 register is accepted. A read operation in between the two write operations is allowed, i.e. read-modify-write is possible on register P5. For details refer to section *"Specific registers" on page 93*.

	7	6	5	4	3	2	1	0
P5	P57	P56	P55	P54	P53	P52	P51	P50
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-23 P5 register contents

Bit position	Bit name	Function
7...0	P5[7:0]	Input data control of pin P5n (in port mode PMC5n = 0). 0: Input mode (PM5n = 1): Low level is input Output mode (PM5n = 0): Low level is output 1: Input mode (PM5n = 1): High level is input Output mode (PM5n = 0): High level is output **Note:** n = 0 to 7

(3) Port mode register 5 (PM5)

The PM5 register is an 8-bit register that specifies the input or output mode.

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

Address $FFFFF42A_H$

Initial Value FF_H

Note Writing to the PM5 register is only possible in a specific sequence, where a write access to the command register (PRCMD) must be made before a write access to the PM5 register is accepted. A read operation in between the two write operations is allowed, i.e. read-modify-write is possible on register PM5. For details refer to section *"Specific registers" on page 93*.

	7	6	5	4	3	2	1	0
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-24 PM5 register contents

Bit position	Bit name	Function
7...0	PM5[7:0]	Input/output mode control of pin P5n (in port mode PMC5n = 0). 0: Output mode 1: Input mode **Note:** n = 0 to 7

(4) Port mode control register 5 (PMC5)

The PMC5 register is an 8-bit register that specifies the port mode or control mode (alternate function).

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

Address $FFFFF44A_H$

Initial Value 00_H

Note Writing to the PMC5 register is only possible in a specific sequence, where a write access to the command register (PRCMD) must be made before a write access to the PMC5 register is accepted. A read operation in between the two write operations is allowed, i.e. read-modify-write is possible on register PMC5. For details refer to section *"Specific registers" on page 93*.

PMC5

7	6	5	4	3	2	1	0
PMC57	PMC56	PMC55	PMC54	PMC53	PMC52	PMC51	PMC50
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-25 PMC5 register contents

Bit position	Bit name	Function
7...0	PMC5]7:0]	Port mode/control mode specification of pin P5n **Note:** n = 0 to 7

PMC5n	PM5n	Function of pin P5n	
0	0	Output port mode	
	1	Input port mode	
1	×	Alternate function	TOR0n output mode

(5) Port emergency shut off control register 5 (PESC5)

The PESC5 register is an 8-bit register that controls the emergency shut off behaviour of output buffers of ports P51 to P56.

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

Address $FFFFF888_H$

Initial Value 00_H

Note Writing to the PESC5 register is only possible in a specific sequence, where a write access to the command register (PRCMD) must be made before a write access to the PESC5 register is accepted. A read operation in between the two write operations is allowed, i.e. read-modify-write is possible on register PESC5. For details refer to section *"Specific registers" on page 93*.

	7	6	5	4	3	2	1	0
PESC5	0	0	0	0	ESO0EN	0	ESO0ED1	ESO0ED0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-26 PESC5 register contents

Bit position	Bit name	Function
3	ESO0EN	Controls the emergency output shut off of P5[6:1] output buffers 0: Emergency shut off control by ESO0 input disabled. 1: Emergency shut off control by ESO0 input enabled **Note:** The output buffers of ports P5[6:1] are forcibly disabled as long as ESO0EN = 1 and ESOST5.ESO0ST = 1.
1, 0	ESO0ED1 ESO0ED0	Specifies the valid edge and level of emergency shut off input (ESO0)

ESO0ED1	ESO0ED0	Valid edge and level of ESO0 input
0	0	Falling edge
	1	Rising edge
1	×	Low level
		High level

Caution: State of the edge detection control bits ESO0ED1 and ESO0ED0 must not be changed while ESO0EN is set (1). Otherwise the output shut off function may be unintentionally triggered or a trigger event may be lost.

Note 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 ESO0EN bit to 0
<4> PRCMD write (write protect released)
<5> clear ESOST5.ESO0ST bit to 0
<6> PRCMD write (write protect released)
<7> set ESO0ED[1:0] bits
<8> PRCMD write (write protect released)
<9> set ESO0EN bit to 1

(6) Port emergency shut off status register 5 (ESOST5)

The ESOST5 register is an 8-bit register that indicates the emergency status control mode (alternate function).

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

Address $FFFFF88A_H$

Initial Value 00_H

Note Writing to the PESC5 register is only possible in a specific sequence, where a write access to the command register (PRCMD) must be made before a write access to the PESC5 register is accepted. A read operation in between the two write operations is allowed, i.e. read-modify-write is possible on register PESC5. For details refer to section *"Specific registers" on page 93*.

ESOST5

7	6	5	4	3	2	1	0
ESO0ST	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

Table 20-27 ESOST5 register contents

Bit position	Bit name	Function
7	ESO0ST	Indicates the status id the emergency output shut off function (ESO0) 0: No emergency shut off was triggered. 1: Emergency shut off of output ports P5[6:1] triggered by ESO0 input. **Note: 1.** Writing the ESO0ST bit is only possible, if the PESC5.ES0EN bit is cleared (0). **2.** The ESO0ST bit can only be cleared by CPU to 0. Setting the ESO0ST bit to 1 is not possible. **3.** The output buffers of ports P5[6:1] are forcibly disabled as long as ESO0ST = 1 and PESC5.ESO0EN = 1.

20.3.7 Port 6

Port 6 is an 8-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register 6 (P6).
- Input or output mode can be set in 1-bit units by using the port mode register 6 (PM6).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register 6 (PMC6).
- Emergency shut off by ES0 input signal of output buffers P61 to P66 can be controlled by port emergency shut off control register 6 (PESC6) and emergency shut off status register 6 (ESOST6).
- Security feature to protect the timer output signals of TMR0 from unintended CPU interference. Registers P6, PM6, PMC6, PESC6 and ESOST6 can only be written in a special sequence.

Table 20-28 Alternate function pins and port types of port 6

Port		Alternate Function	Remark	Port Type
Port 6	P60	TOR10, TTRGR1	Timer I/O (TMR1)	12
	P61	TOR11, TIR10		14
	P62	TOR12, TIR11		
	P63	TOR13, TIR12		
	P64	TOR14, TIR13		
	P65	TOR15	Timer output (TMR1)	13
	P66	TOR16		
	P67	TOR17, TEVTR1	Timer I/O (TMR1)	12

(2) Port register 6 (P6)

The P6 register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins P60 to P67.

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

Address $FFFFF40C_H$

Initial Value Undefined

Note Writing to the P6 register is only possible in a specific sequence, where a write access to the command register (PRCMD) must be made before a write access to the P6 register is accepted. A read operation in between the two write operations is allowed, i.e. read-modify-write is possible on register P6. For details refer to section *"Specific registers" on page 93*.

	7	6	5	4	3	2	1	0
P6	P67	P66	P65	P64	P63	P62	P61	P60
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-29 P6 register contents

Bit position	Bit name	Function
7...0	P6[7:0]	Input data control of pin P6n (in port mode PMC6n = 0). 0: Input mode (PM6n = 1): Low level is input Output mode (PM6n = 0): Low level is output 1: Input mode (PM6n = 1): High level is input Output mode (PM6n = 0): High level is output **Note:** n = 0 to 7

(3) Port mode register 6 (PM6)

The PM6 register is an 8-bit register that specifies the input or output mode.

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

Address $FFFFF42C_H$

Initial Value FF_H

Note Writing to the PM6 register is only possible in a specific sequence, where a write access to the command register (PRCMD) must be made before a write access to the PM6 register is accepted. A read operation in between the two write operations is allowed, i.e. read-modify-write is possible on register PM6. For details refer to section *"Specific registers" on page 93*.

	7	6	5	4	3	2	1	0
PM6	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-30 PM6 register contents

Bit position	Bit name	Function
7...0	PM6[7:0]	Input/output mode control of pin P6n (in port mode PMC6n = 0). 0: Output mode 1: Input mode **Note:** n = 0 to 7

(4) Port mode control register 6 (PMC6)

The PMC6 register is an 8-bit register that specifies the port mode or control mode (alternate function).

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

Address $FFFFF44C_H$

Initial Value 00_H

Note Writing to the PMC6 register is only possible in a specific sequence, where a write access to the command register (PRCMD) must be made before a write access to the PMC6 register is accepted. A read operation in between the two write operations is allowed, i.e. read-modify-write is possible on register PMC6. For details refer to section *"Specific registers" on page 93*.

PMC6

7	6	5	4	3	2	1	0
PMC67	PMC66	PMC65	PMC64	PMC63	PMC62	PMC61	PMC60
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-31 PMC6 register contents (1/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7</td><td>PMC67</td><td>Port mode/control mode specification of pin P67
<table>
<tr><th>PMC67</th><th>PM67</th><th colspan="2">Function of pin P67</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOR17 output mode</td></tr>
<tr><td>1</td><td>TTEVTR1 input mode</td></tr>
</table></td></tr>
<tr><td>6</td><td>PMC66</td><td>Port mode/control mode specification of pin P66
<table>
<tr><th>PMC66</th><th>PM66</th><th colspan="2">Function of pin P66</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td>1</td><td>×</td><td>Alternate function</td><td>TOR16 output mode</td></tr>
</table></td></tr>
</table>

Table 20-31 PMC6 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>5</td><td>PMC65</td><td>Port mode/control mode specification of pin P65
<table>
<tr><th>PMC65</th><th>PM65</th><th colspan="2">Function of pin P65</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td>1</td><td>×</td><td>Alternate function</td><td>TOR15 output mode</td></tr>
</table></td></tr>
<tr><td>4</td><td>PMC64</td><td>Port mode/control mode specification of pin P64
<table>
<tr><th>PMC64</th><th>PM64</th><th colspan="2">Function of pin P64</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOR14 output mode</td></tr>
<tr><td>1</td><td>TIR13 input mode</td></tr>
</table></td></tr>
<tr><td>3</td><td>PMC63</td><td>Port mode/control mode specification of pin P63
<table>
<tr><th>PMC63</th><th>PM63</th><th colspan="2">Function of pin P63</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOR13 output mode</td></tr>
<tr><td>1</td><td>TIR12 input mode</td></tr>
</table></td></tr>
<tr><td>2</td><td>PMC62</td><td>Port mode/control mode specification of pin P62
<table>
<tr><th>PMC62</th><th>PM62</th><th colspan="2">Function of pin P62</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOR12 output mode</td></tr>
<tr><td>1</td><td>TIR11 input mode</td></tr>
</table></td></tr>
<tr><td>1</td><td>PMC61</td><td>Port mode/control mode specification of pin P61
<table>
<tr><th>PMC61</th><th>PM61</th><th colspan="2">Function of pin P61</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOR11 output mode</td></tr>
<tr><td>1</td><td>TIR10 input mode</td></tr>
</table></td></tr>
<tr><td>3</td><td>PMC60</td><td>Port mode/control mode specification of pin P60
<table>
<tr><th>PMC60</th><th>PM60</th><th colspan="2">Function of pin P60</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOR10 output mode</td></tr>
<tr><td>1</td><td>TTRGR1 input mode</td></tr>
</table></td></tr>
</table>

(5) Port emergency shut off control register 6 (PESC6)

The PESC6 register is an 8-bit register that controls the emergency shut off behaviour of output buffers of ports P61 to P66.

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

Address $FFFFF88C_H$

Initial Value 00_H

Note Writing to the PESC6 register is only possible in a specific sequence, where a write access to the command register (PRCMD) must be made before a write access to the PESC6 register is accepted. A read operation in between the two write operations is allowed, i.e. read-modify-write is possible on register PESC6. For details refer to section *"Specific registers" on page 93*.

	7	6	5	4	3	2	1	0
PESC6	0	0	0	0	ESO1EN	0	ESO1ED1	ESO1ED0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-32 PESC6 register contents

Bit position	Bit name	Function
3	ESO1EN	Controls the emergency output shut off of P6[6:1] output buffers 0: Emergency shut off control by ESO1 input disabled. 1: Emergency shut off control by ESO1 input enabled **Note:** The output buffers of ports P6[6:1] are forcibly disabled as long as ESO1EN = 1 and ESOST6.ESO1ST = 1.
1, 0	ESO1ED1 ESO1ED0	Specifies the valid edge and level of emergency shut off input (ESO1)

ESO1ED1	ESO1ED0	Valid edge and level of ESO1 input
0	0	Falling edge
	1	Rising edge
1	×	Low level
		High level

Caution: State of the edge detection control bits ESO1ED1 and ESO1ED0 must not be changed while ESO1EN is set (1). Otherwise the output shut off function may be unintentionally triggered or a trigger event may be lost.

Note 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 ESO1EN bit to 0
<4> PRCMD write (write protect released)
<5> clear ESOST6.ESO1ST bit to 0
<6> PRCMD write (write protect released)
<7> set ESO1ED[1:0] bits
<8> PRCMD write (write protect released)
<9> set ESO1EN bit to 1

(6) Port emergency shut off status register 6 (ESOST6)

The ESOST6 register is an 8-bit register that indicates the emergency status control mode (alternate function).

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

Address $FFFFF88E_H$

Initial Value 00_H

Note Writing to the ESOST6 register is only possible in a specific sequence, where a write access to the command register (PRCMD) must be made before a write access to the ESOST6 register is accepted. A read operation in between the two write operations is allowed, i.e. read-modify-write is possible on register ESOST6. For details refer to section *"Specific registers" on page 93*.

ESOST6

7	6	5	4	3	2	1	0
ESO1ST	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

Table 20-33 ESOST6 register contents

Bit position	Bit name	Function
7	ESO1ST	Indicates the status of the emergency output shut off function (ESO1) 0: No emergency shut off was triggered. 1: Emergency shut off of output ports P6[6:1] triggered by ESO1 input. **Note: 1.** Writing the ESO1ST bit is only possible, if the PESC6.ES1EN bit is cleared (0). **2.** The ESO1ST bit can only be cleared by CPU to 0. Setting the ESO1ST bit to 1 is not possible. **3.** The output buffers of ports P5[6:1] are forcibly disabled as long as ESO1ST = 1 and PESC6.ESO1EN = 1.

20.3.8 Port 7

Port 7 is an 8-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register 7 (P7).
- Input or output mode can be set in 1-bit units by using the port mode register 7 (PM7).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register 7 (PMC7).
- The external interrupt request input shared with the input port functionality of port 7 is always enabled in input port mode.

Table 20-34 Alternate function pins and port types of port 7

Port		Alternate Function	Remark	Port Type
Port 7	P70	TIT00, TEVT1 TOT00	Timer input (TMT0, TMT1) Timer output (TMT0)	6
	P71	TIT01, TTRGT1 TOT01	Timer input (TMT0, TMT1) Timer output (TMT0)	
	P72	TECRT0 INTP12	Timer input (TMT0), External interrupt request input	8
	P73	TIT10, TTRGT0 TOT10	Timer input (TMT0, TMT1) Timer output (TMT1)	6
	P74	TIT11, TEVT0 TOT11	Timer input (TMT0, TMT1) Timer output (TMT1)	
	P75	TECRT1 AFO	Timer input (TMT1) Auxiliary frequency output	

(2) Port register 7 (P7)

The P7 register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins P70 to P75.

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

Address $FFFFF40E_H$

Initial Value Undefined

	7	6	5	4	3	2	1	0
P7	0	0	P75	P74	P73	P72	P71	P70
	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-35 P7 register contents

Bit position	Bit name	Function
5...0	P7[5:0]	Input data control of pin P7n (in port mode PMC7n = 0). 0: Input mode (PM7n = 1): Low level is input Output mode (PM7n = 0): Low level is output 1: Input mode (PM7n = 1): High level is input Output mode (PM7n = 0): High level is output **Note:** n = 0 to5

(3) Port mode register 7 (PM7)

The PM7 register is an 8-bit register that specifies the input or output mode.

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

Address $FFFFF42E_H$

Initial Value FF_H

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

Table 20-36 PM4 register contents

Bit position	Bit name	Function
5...0	PM7[5:0]	Input/output mode control of pin P7n (in port mode PMC7n = 0). 0: Output mode 1: Input mode **Note:** n = 0 to 5

(4) Port mode control register 7 (PMC7)

The PMC7 register is an 8-bit register that specifies the port mode or control mode (alternate function).

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

Address $FFFFF44E_H$

Initial Value 00_H

	7	6	5	4	3	2	1	0
PMC7	0	0	PMC75	PMC74	PMC73	PMC72	PMC71	PMC70
	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-37 PMC7 register contents (1/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>5</td><td>PMC75</td><td>Port mode/control mode specification of pin P75
<table>
<tr><th>PMC75</th><th>PM75</th><th colspan="2">Function of pin P75</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>AFO output mode</td></tr>
<tr><td>1</td><td>TECRT1 input mode</td></tr>
</table></td></tr>
<tr><td>4</td><td>PMC74</td><td>Port mode/control mode specification of pin P74
<table>
<tr><th>PMC74</th><th>PM74</th><th colspan="2">Function of pin P74</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOT11 output mode</td></tr>
<tr><td>1</td><td>TIT11 input mode,
TEVTT0 input mode</td></tr>
</table></td></tr>
<tr><td>3</td><td>PMC73</td><td>Port mode/control mode specification of pin P73
<table>
<tr><th>PMC73</th><th>PM73</th><th colspan="2">Function of pin P73</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOT10 output mode</td></tr>
<tr><td>1</td><td>TIT10 input mode,
TTRGT0 input mode</td></tr>
</table></td></tr>
</table>

Table 20-37 PMC7 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>2</td><td>PMC72</td><td>Port mode/control mode specification of pin P72
<table>
<tr><th>PMC72</th><th>PM72</th><th colspan="2">Function of pin P72</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode
External interrupt request input mode (INTP12)</td></tr>
<tr><td>1</td><td>×</td><td>Alternate function</td><td>TECRT0 input mode
External interrupt request input mode (INTP12)</td></tr>
</table></td></tr>
<tr><td>1</td><td>PMC71</td><td>Port mode/control mode specification of pin P71
<table>
<tr><th>PMC71</th><th>PM71</th><th colspan="2">Function of pin P71</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOT01 output mode</td></tr>
<tr><td>1</td><td>TIT01, TTRGT1 input mode</td></tr>
</table></td></tr>
<tr><td>0</td><td>PMC70</td><td>Port mode/control mode specification of pin P70
<table>
<tr><th>PMC70</th><th>PM70</th><th colspan="2">Function of pin P70</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>TOT00 output mode</td></tr>
<tr><td>1</td><td>TIT00, TEVTT1 input mode</td></tr>
</table></td></tr>
</table>

20.3.9 Port 8

Port 8 is a 7-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register 8 (P8).
- Input or output mode can be set in 1-bit units by using the port mode register 8 (PM8).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register 8 (PMC8).
- The external interrupt request inputs shared with the input port functionality of port 8 are always enabled in input port mode.

Table 20-38 Alternate function pins and port types of port 8

Port		Alternate Function	Remark	Port Type
Port 8	P80	SI30	Serial interface (CSI30) input	2
	P81	SO30	Serial interface (CSI30) output	1S
	P82	SCK30	Serial interface (CSI30) I/O	4
	P83	SCS300 INTP6	Serial interface (CSI30) output, External interrupt request input	7
	P84	SCS301 INTP7	Serial interface (CSI30) output, External interrupt request input	
	P85	SCS302 INTP8	Serial interface (CSI30) output, External interrupt request input	
	P86	SCS303 SSB0	Serial interface (CSI30) output, Serial interface (CSIB0) input	5

(2) Port register 8 (P8)

The P8 register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins P80 to P86.

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

Address $FFFFF410_H$

Initial Value Undefined

	7	6	5	4	3	2	1	0
P8	0	P86	P85	P84	P83	P82	P81	P80
	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-39 P8 register contents

Bit position	Bit name	Function
6...0	P8[6:0]	Input data control of pin P8n (in port mode PMC8n = 0). 0: Input mode (PM8n = 1): Low level is input Output mode (PM8n = 0): Low level is output 1: Input mode (PM8n = 1): High level is input Output mode (PM8n = 0): High level is output **Note:** n = 0 to6

(3) Port mode register 8 (PM8)

The PM8 register is an 8-bit register that specifies the input or output mode.

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

Address $FFFFF430_H$

Initial Value FF_H

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

Table 20-40 PM8 register contents

Bit position	Bit name	Function
6...0	PM8[6:0]	Input/output mode control of pin P8n (in port mode PMC8n = 0). 0: Output mode 1: Input mode **Note:** n = 0 to 6

Remark n = 0 to 6

(4) Port mode control register 8 (PMC8)

The PMC8 register is an 8-bit register that specifies the port mode or control mode (alternate function).

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

Address $FFFFF450_H$

Initial Value 00_H

	7	6	5	4	3	2	1	0
PMC8	0	PMC86	PMC85	PMC84	PMC83	PMC82	PMC81	PMC80
	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-41 PMC8 register contents (1/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>6</td><td>PMC86</td><td>Port mode/control mode specification of pin P86
<table>
<tr><th>PMC86</th><th>PM86</th><th colspan="2">Function of pin P86</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>SCS303 output mode</td></tr>
<tr><td>1</td><td>SSB0 input mode</td></tr>
</table></td></tr>
<tr><td>5</td><td>PMC85</td><td>Port mode/control mode specification of pin P85
<table>
<tr><th>PMC85</th><th>PM85</th><th colspan="2">Function of pin P85</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input mode,
External interrupt request input mode (INTP8)</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>SCS302 output mode</td></tr>
<tr><td>1</td><td>External interrupt request input mode (INTP8)</td></tr>
</table></td></tr>
<tr><td>4</td><td>PMC84</td><td>Port mode/control mode specification of pin P84
<table>
<tr><th>PMC84</th><th>PM84</th><th colspan="2">Function of pin P84</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode,
External interrupt request input mode (INTP7)</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>SCS301 output mode</td></tr>
<tr><td>1</td><td>External interrupt request input mode (INTP7)</td></tr>
</table></td></tr>
</table>

Table 20-41 PMC8 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3</td><td>PMC83</td><td>Port mode/control mode specification of pin P83
<table>
<tr><th>PMC83</th><th>PM83</th><th colspan="2">Function of pin P83</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode,
External interrupt request input mode (INTP6)</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>SCS300 output mode</td></tr>
<tr><td>1</td><td>External interrupt request input mode (INTP6)</td></tr>
</table></td></tr>
<tr><td>2</td><td>PMC82</td><td>Port mode/control mode specification of pin P82
<table>
<tr><th>PMC82</th><th>PM82</th><th colspan="2">Function of pin P82</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td>1</td><td>×</td><td>Alternate function</td><td>SCK30 I/O mode</td></tr>
</table></td></tr>
<tr><td>1</td><td>PMC81</td><td>Port mode/control mode specification of pin P81
<table>
<tr><th>PMC81</th><th>PM81</th><th colspan="2">Function of pin P81</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td>1</td><td>×</td><td>Alternate function</td><td>SO30 output mode</td></tr>
</table></td></tr>
<tr><td>0</td><td>PMC80</td><td>Port mode/control mode specification of pin P80
<table>
<tr><th>PMC80</th><th>PM80</th><th colspan="2">Function of pin P80</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td>1</td><td>×</td><td>Alternate function</td><td>SI30 input mode</td></tr>
</table></td></tr>
</table>

20.3.10 Port 9

Port 9 is a 7-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register 9 (P9).
- Input or output mode can be set in 1-bit units by using the port mode register 9 (PM9).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register 9 (PMC9).
- The external interrupt request inputs shared with the input port functionality of port 9 are always enabled in input port mode.

Table 20-42 Alternate function pins and port types of port 9

Port		Alternate Function	Remark	Port Type
Port 9	P90	SI31 [a]	Serial interface (CSI31) input [a]	2
	P91	SO31 [a]	Serial interface (CSI31) output [a]	1S
	P92	SCK31 [a]	Serial interface (CSI31) I/O [a]	4
	P93	SCS310 [a] INTP9	Serial interface (CSI31) outpu [a] External interrupt request input	7
	P94	SCS311 [a] INTP10	Serial interface (CSI31) output [a] External interrupt request input	
	P95	SCS312 [a] INTP11	Serial interface (CSI31) output [a], External interrupt request input	
	P96	SCS313 [a] SSB1 [a]	Serial interface (CSI31) output [a] Serial interface (CSIB1) input [a]	5

a) Alternate function not available on µPD70F3447.

(2) Port register 9 (P9)

The P9 register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins P90 to P96.

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

Address $FFFFF412_H$

Initial Value Undefined

	7	6	5	4	3	2	1	0
P9	0	P96	P95	P94	P93	P92	P91	P90
	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-43 P9 register contents

Bit position	Bit name	Function
6...0	P9[6:0]	Input data control of pin P9n (in port mode PMC9n = 0). 0: Input mode (PM9n = 1): Low level is input Output mode (PM9n = 0): Low level is output 1: Input mode (PM9n = 1): High level is input Output mode (PM9n = 0): High level is output **Note:** n = 0 to6

(3) Port mode register 9 (PM9)

The PM9 register is an 8-bit register that specifies the input or output mode.

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

Address $FFFFF432_H$

Initial Value FF_H

	7	6	5	4	3	2	1	0
PM9	1	PM96	PM95	PM94	PM93	PM92	PM91	PM90
	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-44 PM9 register contents

Bit position	Bit name	Function
6...0	PM9[6:0]	Input/output mode control of pin P9n (in port mode PMC9n = 0). 0: Output mode 1: Input mode **Note:** n = 0 to 6

(4) Port mode control register 9 (PMC9)

The PMC9 register is an 8-bit register that specifies the port mode or control mode (alternate function).

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

Address $FFFFF452_H$

Initial Value 00_H

Caution On μPD70F3447 do not set bits PMC9[6:0] to 1, since the corresponding alternate function is not available.

	7	6	5	4	3	2	1	0
PMC9	0	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90
	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-45 PMC9 register contents (1/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>6</td><td>PMC96</td><td>Port mode/control mode specification of pin P96
<table>
<tr><th>PMC96</th><th>PM96</th><th colspan="2">Function of pin P96</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input mode</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>SCS313 output mode</td></tr>
<tr><td>1</td><td>SSB1 input mode</td></tr>
</table></td></tr>
<tr><td>5</td><td>PMC95</td><td>Port mode/control mode specification of pin P95
<table>
<tr><th>PMC95</th><th>PM95</th><th colspan="2">Function of pin P95</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input mode,
External interrupt request input mode (INTP11)</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>SCS312 output mode</td></tr>
<tr><td>1</td><td>External interrupt request input mode (INTP11)</td></tr>
</table></td></tr>
<tr><td>4</td><td>PMC94</td><td>Port mode/control mode specification of pin P94
<table>
<tr><th>PMC94</th><th>PM94</th><th colspan="2">Function of pin P94</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode,
External interrupt request input mode (INTP10)</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>SCS311 output mode</td></tr>
<tr><td>1</td><td>External interrupt request input mode (INTP10)</td></tr>
</table></td></tr>
</table>

Table 20-45 PMC9 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3</td><td>PMC93</td><td>Port mode/control mode specification of pin P93
<table>
<tr><th>PMC93</th><th>PM93</th><th colspan="2">Function of pin P93</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode,
External interrupt request input mode (INTP9)</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td rowspan="2">Alternate function</td><td>SCS310 output mode</td></tr>
<tr><td>1</td><td>External interrupt request input mode (INTP9)</td></tr>
</table></td></tr>
<tr><td>2</td><td>PMC92</td><td>Port mode/control mode specification of pin P92
<table>
<tr><th>PMC92</th><th>PM92</th><th colspan="2">Function of pin P92</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td>1</td><td>×</td><td>Alternate function</td><td>SCK31 I/O mode</td></tr>
</table></td></tr>
<tr><td>1</td><td>PMC91</td><td>Port mode/control mode specification of pin P91
<table>
<tr><th>PMC91</th><th>PM91</th><th colspan="2">Function of pin P91</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td>1</td><td>×</td><td>Alternate function</td><td>SO31 output mode</td></tr>
</table></td></tr>
<tr><td>0</td><td>PMC90</td><td>Port mode/control mode specification of pin P90
<table>
<tr><th>PMC90</th><th>PM90</th><th colspan="2">Function of pin P90</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td colspan="2">Output port mode</td></tr>
<tr><td>1</td><td colspan="2">Input port mode</td></tr>
<tr><td>1</td><td>×</td><td>Alternate function</td><td>SI31 input mode</td></tr>
</table></td></tr>
</table>

20.3.11 Port 10

Port 10 is a 3-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register 10 (P10).
- Input or output mode can be set in 1-bit units by using the port mode register 10 (PM10).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register 10 (PMC10).

Table 20-46 Alternate function pins and port types of port 10

Port		Alternate Function	Remark	Port Type
Port 10	P100	TCLR0 [a], TICC00 [a] TOP81	Timer input (ITENC0) [a] Timer output (TMP8)	6
	P101	TCUD0 [a], TICC01 [a]	Timer input (ITENC0) [a]	10
	P102	TIUD0 [a] TO1 [a]	Timer input (ITENC0) [a] Timer output (ITENC0) [a]	6

a) Alternate function not available on µPD70F3447.

(2) Port register 10 (P10)

The P10 register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins P100 to P105.

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

Address $FFFFF414_H$

Initial Value Undefined

	7	6	5	4	3	2	1	0
P10	0	0	0	0	0	P102	P101	P100
	R	R	R	R	R	R/W	R/W	R/W

Table 20-47 P10 register contents

Bit position	Bit name	Function
2...0	P10[2:0]	Input data control of pin P10n (in port mode PMC10n = 0). 0: Input mode (PM10n = 1): Low level is input Output mode (PM10n = 0): Low level is output 1: Input mode (PM10n = 1): High level is input Output mode (PM10n = 0): High level is output **Note:** n = 0 to 2

(3) Port mode register 10 (PM10)

The PM10 register is an 8-bit register that specifies the input or output mode.

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

Address $FFFFF434_H$

Initial Value FF_H

	7	6	5	4	3	2	1	0
PM10	1	1	1	1	1	PM102	PM101	PM100
	R	R	R	R	R	R/W	R/W	R/W

Table 20-48 PM10 register contents

Bit position	Bit name	Function
2...0	PM10[2:0]	Input/output mode control of pin P10n (in port mode PMC10n = 0). 0: Output mode 1: Input mode **Note:** n = 0 to 2

(4) Port mode control register 10 (PMC10)

The PMC10 register is an 8-bit register that specifies the port mode or control mode (alternate function).

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

Address $FFFFF454_H$

Initial Value 00_H

Caution On µPD70F3447 do not set bits PMC100 and PMC101 to 1, since the corresponding alternate function is not available.
Further do not set PMC102 to 1, when PM102 is set (1).

	7	6	5	4	3	2	1	0
PMC10	0	0	0	0	0	PMC102	PMC101	PMC100
	R	R	R	R	R	R/W	R/W	R/W

Table 20-49 PMC10 register contents

Bit position	Bit name	Function
2	PMC102	Port mode/control mode specification of pin P102
1	PMC101	Port mode/control mode specification of pin P101
0	PMC100	Port mode/control mode specification of pin P100

PMC102 / PM102 / Function of pin P102

PMC102	PM102	Function of pin P102	
0	0	Output port mode	
	1	Input mode	
1	0	Alternate function	TOP81 output mode
	1		TCLR0 input mode[Note], TICC0 input mode[Note]

PMC101 / PM101 / Function of pin P101

PMC101	PM101	Function of pin P101	
0	0	Output port mode	
	1	Input port mode	
1	x	Alternate function	TCUD0 input mode[Note], TICC01 input mode[Note]

PMC100 / PM100 / Function of pin P100

PMC100	PM100	Function of pin P100	
0	0	Output port mode	
	1	Input port mode	
1	0	Alternate function	TO1 output mode
	1		TIUD0 input mode

20.3.12 Port AL

Port AL is a 16-bit I/O port that can be set to input or output mode in 1-bit units. When the higher 8 bits of port AL are used as port ALH (PALH) and the lower 8 bits as port ALL (PALL), port AL becomes two 8-bit ports that can be set in the input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register AL (PAL).
- Input or output mode can be set in 1-bit units by using the port mode register AL (PMAL).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register AL (PMCAL).

Table 20-50 Alternate function pins and port types of port AL

Port		Alternate Function	Remark	Port Type
Port AL	PAL0 to PAL15	A0 to A15 [a]	External address bus [a]	1

a) Alternate function not available on µPD70F3447.

Caution On the µPD70F3187, in single-chip mode 1 or in ROM-less mode, this port has external address bus function only. Reprogramming of port AL to port mode is not possible in these modes. Reading and writing of the port register PAL and port mode register PMAL is possible but has no effect. Reading of the port mode control register PMCAL is possible and the result is always $FFFF_H$. Writing of the port mode control register PMCAL is not possible.

(2) Port register AL (PAL)

The PAL register is a 16-bit register that controls reading the pin levels and writing the output levels of port pins PAL0 to PAL15.

Access This register can be read/written in 16-bit units.
If the higher 8 bits of the PAL register are used as PALH register, and the lower 8 bits as the PALL register, however, these registers can be read or written in 8-bit or 1-bit units.

Address PAL, PALL: $FFFFF000_H$
PALH: $FFFFF001_H$

Initial Value Undefined

PAL

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAL15	PAL14	PAL13	PAL12	PAL11	PAL10	PAL9	PAL8	PAL7	PAL6	PAL5	PAL4	PAL3	PAL2	PAL1	PAL0
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
PALH								PALL							

PALL

7	6	5	4	3	2	1	0
PAL7	PAL6	PAL5	PAL4	PAL3	PAL2	PAL1	PAL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PALH

7	6	5	4	3	2	1	0
PAL15	PAL14	PAL13	PAL12	PAL11	PAL10	PAL9	PAL8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-51 PAL register contents

Bit position	Bit name	Function
15...0	PAL[15:0]	Input data control of pin PALn (in port mode PMCALn = 0). 0: Input mode (PMALn = 1): Low level is input Output mode (PMALn = 0): Low level is output 1: Input mode (PMALn = 1): High level is input Output mode (PMALn = 0): High level is output **Note:** n = 0 to 15

(3) Port mode register AL (PMAL)

The PMAL register is a 16-bit register that specifies the input or output mode of port pins PAL0 to PAL15.

Access This register can be read/written in 16-bit units.
If the higher 8 bits of the PMAL register are used as PMALH register, and the lower 8 bits as the PMALL register, however, these registers can be read or written in 8-bit or 1-bit units.

Address PMAL, PMALL: $FFFFF020_H$
PMALH: $FFFFF021_H$

Initial Value $FFFF_H$

PMAL

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMAL15	PMAL14	PMAL13	PMAL12	PMAL11	PMAL10	PMAL9	PMAL8	PMAL7	PMAL6	PMAL5	PMAL4	PMAL3	PMAL2	PMAL1	PMAL0
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

PMALH PMALL

PMALL

7	6	5	4	3	2	1	0
PMAL7	PMAL6	PMAL5	PMAL4	PMAL3	PMAL2	PMAL1	PMAL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PMALH

7	6	5	4	3	2	1	0
PMAL15	PMAL14	PMAL13	PMAL12	PMAL11	PMAL10	PMAL9	PMAL8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-52 PMAL register contents

Bit position	Bit name	Function
15...0	PMAL[15:0]	Input/output mode control of pin PALn (in port mode PMCALn = 0). 0: Output mode 1: Input mode **Note:** n = 0 to 15

(4) Port AL mode control register (PMCAL)

The PMCAL register is a 16-bit read/write register that specifies the port mode or control mode (alternate function) of Port AL.

Access This register can be read/written in 16-bit units.
If the higher 8 bits of the PMCAL register are used as PMCALH register, and the lower 8 bits as the PMCALL register, however, these registers can be read or written in 8-bit or 1-bit units.

Address PMCAL, PMCALL: $FFFFF040_H$
PMCALH: $FFFFF041_H$

Initial Value Reset input sets this register to 0000_H in single-chip mode 0, and to $FFFF_H$ in ROM-less mode and single-chip mode 1.

Caution
1. On μPD70F3447 the alternate function is not available. Therefore do not set PMCALn bits to 1.
2. On μPD70F3187, in single-chip mode 1 or in ROM-less mode, this register can not be written. Reading is possible and returns $FFFF_H$.

PMCAL

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMCAL15	PMCAL14	PMCAL13	PMCAL12	PMCAL11	PMCAL10	PMCAL9	PMCAL8	PMCAL7	PMCAL6	PMCAL5	PMCAL4	PMCAL3	PMCAL2	PMCAL1	PMCAL0
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
PMCALH								PMCALL							

PMCALL

7	6	5	4	3	2	1	0
PMCAL7	PMCAL6	PMCAL5	PMCAL4	PMCAL3	PMCAL2	PMCAL1	PMCAL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PMCALH

7	6	5	4	3	2	1	0
PMCAL15	PMCAL14	PMCAL13	PMCAL12	PMCAL11	PMCAL10	PMCAL9	PMCAL8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-53 PMCAL register contents

Bit position	Bit name	Function
15...0	PMCAL [15:0]	Port mode/control mode specification of pin PALn

PMCALn	PMALn	Function of pin PALn
0	0	Output port mode
	1	Input port mode
1	×	External address bus output mode (An)

Note: n = 0 to 15

20.3.13 Port AH

Port AH is a 6-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register AH (PAH).
- Input or output mode can be set in 1-bit units by using the port mode register AH (PMAH).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register AH (PMCAH).

Table 20-54 Alternate function pins and port types of port AH

Port		Alternate Function	Remark	Port Type
Port AH	PAH0 to PAH5	A16 to A21 [a]	External address bus [a]	1

a) Alternate function not available on µPD70F3447.

Caution On the µPD70F3187, in single-chip mode 1 or in ROM-less mode, this port has external address bus function only. Reprogramming of port AH to port mode is not possible in these modes. Reading and writing of the port register PAH and port mode register PMAH is possible but has no effect. Reading of the port mode control register PMCAH is possible and the result is always $3F_H$. Writing of the port mode control register PMCAH is not possible.

(2) Port register AH (PAH)

The PAH register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins PAH0 to PAH5.

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

Address $FFFFF002_H$

Initial Value Undefined

	7	6	5	4	3	2	1	0
PAH	0	0	PAH5	PAH4	PAH3	PAH2	PAH1	PAH0
	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-55 PAH register contents

Bit position	Bit name	Function
5...0	PAH[5:0]	Input data control of pin PAHn (in port mode PMCAHn = 0). 0: Input mode (PMAHn = 1): Low level is input Output mode (PMAHn = 0): Low level is output 1: Input mode (PMAHn = 1): High level is input Output mode (PMAHn = 0): High level is output **Note:** n = 0 to 5

(3) Port mode register AH (PMAH)

The PMAH register is an 8-bit register that specifies the input or output mode of port pins PAL0 to PAL15.

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

Address $FFFFF022_H$

Initial Value FF_H

	7	6	5	4	3	2	1	0
PMAH	0	0	PMAH5	PMAH4	PMAH3	PMAH2	PMAH1	PMAH0
	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-56 PMAH register contents

Bit position	Bit name	Function
5...0	PMAH[5:0]	Input/output mode control of pin PAHn (in port mode PMCAHn = 0). 0: Output mode 1: Input mode **Note:** n = 0 to 5

(4) Port mode control register AH (PMCAH)

The PMCAH register is an 8-bit register that specifies the port mode or control mode (alternate function) of Port AH.

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

Address $FFFFF042_H$

Initial Value Reset input sets this register to 00_H in single-chip mode 0, and to $3F_H$ in ROM-less mode and single-chip mode 1.

Caution
1. On μPD70F3447 the alternate function is not available. Therefore do not set PMCAHn bits to 1.
2. On μPD70F3187, in single-chip mode 1 or in ROM-less mode, this register can not be written. Reading is possible and returns $3F_H$.

	7	6	5	4	3	2	1	0
PMCAH	0	0	PMCAH5	PMCAH4	PMCAH3	PMCAH2	PMCAH1	PMCAH0
	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-57 PMCAH register contents

Bit position	Bit name	Function
5...0	PMCAH [5:0]	Port mode/control mode specification of pin PAHn

PMCAHn	PMAHn	Function of pin PAHn
0	0	Output port mode
	1	Input port mode
1	×	External address bus output mode (A[n+16])

Note: n = 0 to 5

20.3.14 Port DL

Port DL is a 16-bit I/O port that can be set to input or output mode in 1-bit units. When the higher 8 bits of port DL are used as port DLH (PDLH) and the lower 8 bits as port DLL (PDLL), port DL becomes two 8-bit ports that can be set in the input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register DL (PDL).
- Input or output mode can be set in 1-bit units by using the port mode register DL (PMDL).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register DL (PMCDL).

Table 20-58 Alternate function pins and port types of port DL

Port		Alternate Function	Remark	Port Type
Port DL	PDL0 to PDL15	D0 to D15 [a]	External data bus [a]	4C

a) Alternate function not available on µPD70F3447.

Caution On the µPD70F3187, in single-chip mode 1 or in ROM-less mode, this port has external data bus function only. Reprogramming of port DL to port mode is not possible in these modes. Reading and writing of the port register PDL and port mode register PMDL is possible but has no effect. Reading of the port mode control register PMCDL is possible and the result is always $FFFF_H$. Writing of the port mode control register PMCDL is not possible.

(2) Port register DL (PDL)

The PDL register is a 16-bit register that controls reading the pin levels and writing the output levels of port pins PDL0 to PDL15.

Access This register can be read/written in 16-bit units.
If the higher 8 bits of the PDL register are used as PDLH register, and the lower 8 bits as the PDLL register, however, these registers can be read or written in 8-bit or 1-bit units.

Address PDL, PDLL: $FFFFF004_H$
PDLH: $FFFFF005_H$

Initial Value Undefined

PDL

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDL15	PDL14	PDL13	PDL12	PDL11	PDL10	PDL9	PDL8	PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0
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

PDLH PDLL

PDLL

7	6	5	4	3	2	1	0
PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDLH

7	6	5	4	3	2	1	0
PDL15	PDL14	PDL13	PDL12	PDL11	PDL10	PDL9	PDL8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-59 PDL register contents

Bit position	Bit name	Function
15...0	PDL[15:0]	Input data control of pin PDLn (in port mode PMCDLn = 0). 0: Input mode (PMDLn = 1): Low level is input Output mode (PMDLn = 0): Low level is output 1: Input mode (PMDLn = 1): High level is input Output mode (PMDLn = 0): High level is output **Note:** n = 0 to 15

(3) Port mode register DL (PMDL)

The PMDL register is a 16-bit register that specifies the input or output mode of port pins PDL0 to PDL15.

Access This register can be read/written in 16-bit units.
If the higher 8 bits of the PMDL register are used as PMDLH register, and the lower 8 bits as the PMDLL register, however, these registers can be read or written in 8-bit or 1-bit units.

Address PMDL, PMDLL: $FFFFF024_H$
PMDLH: $FFFFF025_H$

Initial Value $FFFF_H$

PMDL

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMDL15	PMDL14	PMDL13	PMDL12	PMDL11	PMDL10	PMDL9	PMDL8	PMDL7	PMDL6	PMDL5	PMDL4	PMDL3	PMDL2	PMDL1	PMDL0
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

PMDLH PMDLL

PMDLL

7	6	5	4	3	2	1	0
PMDL7	PMDL6	PMDL5	PMDL4	PMDL3	PMDL2	PMDL1	PMDL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PMDLH

7	6	5	4	3	2	1	0
PMDL15	PMDL14	PMDL13	PMDL12	PMDL11	PMDL10	PMDL9	PMDL8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-60 PMDL register contents

Bit position	Bit name	Function
15...0	PMDL[15:0]	Input/output mode control of pin PDLn (in port mode PMCDLn = 0). 0: Output mode 1: Input mode **Note:** n = 0 to 15

(4) Port mode control register DL (PMCDL)

The PMCDL register is a 16-bit read/write register that specifies the port mode or control mode (alternate function) of Port DL.

Access This register can be read/written in 16-bit units.
If the higher 8 bits of the PMCDL register are used as PMCDLH register, and the lower 8 bits as the PMCDLL register, however, these registers can be read or written in 8-bit or 1-bit units.

Address PMCDL, PMCDLL: $FFFFF044_H$
PMCDLH: $FFFFF045_H$

Initial Value Reset input sets this register to 0000_H in single-chip mode 0, and to $FFFF_H$ in ROM-less mode and single-chip mode 1.

Caution
1. On μPD70F3447 the alternate function is not available. Therefore do not set PMCDLn bits to 1.
2. On μPD70F3187, in single-chip mode 1 or in ROM-less mode, this register can not be written. Reading is possible and returns $FFFF_H$.

PMCDL

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMCDL15	PMCDL14	PMCDL13	PMCDL12	PMCDL11	PMCDL10	PMCDL9	PMCDL8	PMCDL7	PMCDL6	PMCDL5	PMCDL4	PMCDL3	PMCDL2	PMCDL1	PMCAL0
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
PMCDLH								PMCDLL							

PMCDLL

7	6	5	4	3	2	1	0
PMCDL7	PMCDL6	PMCDL5	PMCDL4	PMCDL3	PMCDL2	PMCDL1	PMCDL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PMCDLH

7	6	5	4	3	2	1	0
PMCDL15	PMCDL14	PMCDL13	PMCDL12	PMCDL11	PMCDL10	PMCDL9	PMCDL8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-61 PMCDL register contents

Bit position	Bit name	Function
15...0	PMCDL [15:0]	Port mode/control mode specification of pin PDLn

PMCDLn	PMDLn	Function of pin PDLn
0	0	Output port mode
	1	Input port mode
1	×	ExternDL data bus input/output mode (Dn)

Note: n = 0 to 15

20.3.15 Port DH

Port DH is a 16-bit I/O port that can be set to input or output mode in 1-bit units. When the higher 8 bits of port DH are used as port DHH (PDHH) and the lower 8 bits as port DHL (PDHL), port DH becomes two 8-bit ports that can be set in the input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register DH (PDH).
- Input or output mode can be set in 1-bit units by using the port mode register DH (PMDH).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register DH (PMCDH).

Table 20-62 Alternate function pins and port types of port DH

Port		Alternate Function	Remark	Port Type
Port DH	PDH0 to PDH15	D16 to D31 [a]	External data bus [a]	4C

a) Alternate function not available on µPD70F3447.

Caution On the µPD70F3187, in single-chip mode 1 or in ROM-less mode, this port has external data bus function only. Reprogramming of port DH to port mode is not possible in these modes. Reading and writing of the port register PDH and port mode register PMDH is possible but has no effect. Reading of the port mode control register PMCDH is possible and the result is always $FFFF_H$. Writing of the port mode control register PMCDH is not possible.

(2) Port register DH (PDH)

The PDH register is a 16-bit register that controls reading the pin levels and writing the output levels of port pins PDH0 to PDH15.

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

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

Reset input causes an undefined register content.

Access This register can be read/written in 16-bit units.
If the higher 8 bits of the PDH register are used as PDHH register, and the lower 8 bits as the PDHL register, however, these registers can be read or written in 8-bit or 1-bit units.

Address PDH, PDHL: $FFFFF006_H$
PDHH: $FFFFF007_H$

Initial Value Undefined

PDH

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDH15	PDH14	PDH13	PDH12	PDH11	PDH10	PDH9	PDH8	PDH7	PDH6	PDH5	PDH4	PDH3	PDL2	PDL1	PDL0
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
PDHH								PDHL							

PDHL

7	6	5	4	3	2	1	0
PDH7	PDH6	PDH5	PDH4	PDH3	PDH2	PDH1	PDH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDHH

7	6	5	4	3	2	1	0
PDH15	PDH14	PDH13	PDH12	PDH11	PDH10	PDH9	PDH8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-63 PDH register contents

Bit position	Bit name	Function
15...0	PDH[15:0]	Input data control of pin PDHn (in port mode PMCDHn = 0). 0: Input mode (PMDHn = 1): Low level is input Output mode (PMDHn = 0): Low level is output 1: Input mode (PMDHn = 1): High level is input Output mode (PMDHn = 0): High level is output **Note:** n = 0 to 15

(3) Port mode register DH (PMDH)

The PMDH register is a 16-bit register that specifies the input or output mode of port pins PDH0 to PDH15.

Access This register can be read/written in 16-bit units.
If the higher 8 bits of the PMDH register are used as PMDHH register, and the lower 8 bits as the PMDHL register, however, these registers can be read or written in 8-bit or 1-bit units.

Address PMDH, PMDHL: $FFFFF026_H$
PMDHH: $FFFFF027_H$

Initial Value $FFFF_H$

PMDH

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMDH15	PMDH14	PMDH13	PMDH12	PMDH11	PMDH10	PMDH9	PMDH8	PMDH7	PMDH6	PMDH5	PMDH4	PMDH3	PMDH2	PMDH1	PMDH0
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

PMDHH PMDHL

PMDHL

7	6	5	4	3	2	1	0
PMDH7	PMDH6	PMDH5	PMDH4	PMDH3	PMDH2	PMDH1	PMDH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PMDHH

7	6	5	4	3	2	1	0
PMDH15	PMDH14	PMDH13	PMDH12	PMDH11	PMDH10	PMDH9	PMDH8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-64 PMDH register contents

Bit position	Bit name	Function
15...0	PMDH[15:0]	Input/output mode control of pin PDHn (in port mode PMCDHn = 0). 0: Output mode 1: Input mode **Note:** n = 0 to 15

(4) Port mode control register DH (PMCDH)

The PMCDH register is a 16-bit read/write register that specifies the port mode or control mode (alternate function) of Port DH.

Access This register can be read/written in 16-bit units.
If the higher 8 bits of the PMCDH register are used as PMCDHH register, and the lower 8 bits as the PMCDHL register, however, these registers can be read or written in 8-bit or 1-bit units.

Address PMCDH, PMCDHL: $FFFFF046_H$
PMCDHH: $FFFFF047_H$

Initial Value Reset input sets this register to 0000_H in single-chip mode 0, and to $FFFF_H$ in ROM-less mode and single-chip mode 1.

Caution
1. On μPD70F3447 the alternate function is not available. Therefore do not set PMCDHn bits to 1.
2. On μPD70F3187, in single-chip mode 1 or in ROM-less mode, this register can not be written. Reading is possible and returns $FFFF_H$.

PMCDH

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMCDH15	PMCDH14	PMCDH13	PMCDH12	PMCDH11	PMCDH10	PMCDH9	PMCDH8	PMCDH7	PMCDH6	PMCDH5	PMCDH4	PMCDH3	PMCDH2	PMCDH1	PMCAL0
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
PMCDHH								PMCDHL							

PMCDHL

7	6	5	4	3	2	1	0
PMCDH7	PMCDH6	PMCDH5	PMCDH4	PMCDH3	PMCDH2	PMCDH1	PMCDH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PMCDHH

7	6	5	4	3	2	1	0
PMCDH15	PMCDH14	PMCDH13	PMCDH12	PMCDH11	PMCDH10	PMCDH9	PMCDH8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-65 PMCDH register contents

Bit position	Bit name	Function
15...0	PMCDH [15:0]	Port mode/control mode specification of pin PDHn

PMCDHn	PMDHn	Function of pin PDHn
0	0	Output port mode
	1	Input port mode
1	×	External data bus input/output mode (D[n+16])

Note: n = 0 to 15

20.3.16 Port CS

Port CS is a 4-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register CS (PCS).
- Input or output mode can be set in 1-bit units by using the port mode register CS (PMCS).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register CS (PMCCS).

Table 20-66 Alternate function pins and port types of port CS

Port		Alternate Function	Remark	Port Type
Port CS	PCS0	$\overline{CS0}$ [a]	Chip select signal output [a]	1
	PCS1	$\overline{CS1}$ [a]		
	PCS3	$\overline{CS3}$ [a]		
	PCS4	$\overline{CS4}$ [a]		

a) Alternate function not available on µPD70F3447.

Caution On the µPD70F3187, in single-chip mode 1 or in ROM-less mode, this port has external control bus function only. Reprogramming of port CS to port mode is not possible in these modes. Reading and writing of the port register PCS and port mode register PMCS is possible but has no effect. Reading of the port mode control register PMCCS is possible and the result is always $1B_H$. Writing of the port mode control register PMCCS is not possible.

(2) Port register CS (PCS)

The PCS register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins PCS0, PCS1, PCS3 and PCS4.

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

Address $FFFFF008_H$

Initial Value Undefined

	7	6	5	4	3	2	1	0
PCS	0	0	0	PCS4	PCS3	0	PCS1	PCS0
	R	R	R	R/W	R/W	R	R/W	R/W

Table 20-67 PCS register contents

Bit position	Bit name	Function
4, 3, 1, 0	PCS [4,3,1,0]	Input data control of pin PCSn (in port mode PMCCSn = 0). 0: Input mode (PMCSn = 1): Low level is input Output mode (PMCSn = 0): Low level is output 1: Input mode (PMCSn = 1): High level is input Output mode (PMCSn = 0): High level is output **Note:** n = 0, 1, 3, 4

(3) Port mode register CS (PMCS)

The PMCS register is an 8-bit register that specifies the input or output mode of port pins PCS0, PCS1, PCS3 and PCS4.

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

Address $FFFFF028_H$

Initial Value FF_H

	7	6	5	4	3	2	1	0
PMCS	1	1	1	PMCS4	PMCS3	1	PMCS1	PMCS0
	R	R	R	R/W	R/W	R	R/W	R/W

Table 20-68 PMCS register contents

Bit position	Bit name	Function
4, 3, 1, 0	PMCS [4,3,1,0]	Input/output mode control of pin PCSn (in port mode PMCCSn = 0). 0: Output mode 1: Input mode **Note:** n = 0, 1, 3, 4

(4) Port mode control register CS (PMCCS)

The PMCCSL register is an 8-bit register that specifies the port mode or control mode (alternate function) of port pins PCS0, PCS1, PCS3 and PCS4.

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

Address $FFFFF048_H$

Initial Value Reset input sets this register to 00_H in single-chip mode 0, and to $1B_H$ in ROM-less mode and single-chip mode 1.

Caution

1. On μPD70F3447 the alternate function is not available. Therefore do not set PMCCSn bits to 1.
2. On μPD70F3187, in single-chip mode 1 or in ROM-less mode, this register can not be written. Reading is possible and returns $1B_H$.

PMCCS

7	6	5	4	3	2	1	0
0	0	05	PMCCS4	PMCCS3	0	PMCCS1	PMCCS0
R	R	R	R/W	R/W	R	R/W	R/W

Table 20-69 PMCCS register contents

Bit position	Bit name	Function
4, 3, 1, 0	PMCCS [4,3,1,0]	Port mode/control mode specification of pin PCSn

PMCCSn	PMCSn	Function of pin PCSn
0	0	Output port mode
	1	Input port mode
1	×	Chip select signal output mode ($\overline{CSn}$)

Note: n = 0, 1, 3, 4

20.3.17 Port CT

Port CT is a 2-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register CT (PCT).
- Input or output mode can be set in 1-bit units by using the port mode register CT (PMCT).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register CT (PMCCT).

Table 20-70 Alternate function pins and port types of port CT

Port		Alternate Function	Remark	Port Type
Port CT	PCT4	$\overline{RD}$ [a]	Read strobe signal output [a]	1
	PCT5	$\overline{WR}$ [a]	Write strobe signal output [a]	

a) Alternate function not available on µPD70F3447.

Caution On the µPD70F3187, in single-chip mode 1 or in ROM-less mode, this port has external control bus function only. Reprogramming of port CT to port mode is not possible in these modes. Reading and writing of the port register PCT and port mode register PMCT is possible but has no effect. Reading of the port mode control register PMCCT is possible and the result is always 30_H. Writing of the port mode control register PMCCT is not possible.

(a) Port register CT (PCT)

The PCT register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins PCT4 and PCT5.

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

Address $FFFFF00A_H$

Initial Value Undefined

	7	6	5	4	3	2	1	0
PCT	0	0	PCT5	PCT4	0	0	0	0
	R	R	R/W	R/W	R	R	R	R

Table 20-71 PCT register contents

Bit position	Bit name	Function
5, 4	PCT[5:4]	Input data control of pin PCTn (in port mode PMCCTn = 0). 0: Input mode (PMCTn = 1): Low level is input Output mode (PMCTn = 0): Low level is output 1: Input mode (PMCTn = 1): High level is input Output mode (PMCTn = 0): High level is output **Note:** n = 4, 5

(2) Port mode register CT (PMCT)

The PMCT register is an 8-bit register that specifies the input or output mode of port pins PCT4 and PCT5.

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

Address $FFFFF02A_H$

Initial Value FF_H

	7	6	5	4	3	2	1	0
PMCT	1	1	PMCT5	PMCT4	1	1	1	1
	R	R	R/W	R/W	R	R	R	R

Table 20-72 PMCS register contents

Bit position	Bit name	Function
5, 4	PMCT[5:4]	Input/output mode control of pin PCTn (in port mode PMCCTn = 0). 0: Output mode 1: Input mode **Note:** n = 4, 5

(3) Port mode control register CT (PMCCT)

The PMCCTL register is an 8-bit register that specifies the port mode or control mode (alternate function) of port pins PCT4 and PCT5.

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

Address FFFFF04A$_H$

Initial Value Reset input sets this register to 00$_H$ in single-chip mode 0, and to 30$_H$ in ROM-less mode and single-chip mode 1.

Caution
1. On μPD70F3447 the alternate function is not available. Therefore do not set PMCCTn bits to 1.
2. On μPD70F3187, in single-chip mode 1 or in ROM-less mode, this register can not be written. Reading is possible and returns 30$_H$.

PMCCT

7	6	5	4	3	2	1	0
0	0	PMCCT5	PMCCT4	0	0	0	0
R	R	R/W	R/W	R	R	R	R

Table 20-73 PMCCT register contents

Bit position	Bit name	Function
5	PMCCT5	Port mode/control mode specification of pin PCT5
4	PMCCT4	Port mode/control mode specification of pin PCT4

Port mode/control mode specification of pin PCT5

PMCCT5	PMCT5	Function of pin PCT5
0	0	Output port mode
	1	Input port mode
1	×	Write strobe signal output mode ($\overline{WR}$)

Port mode/control mode specification of pin PCT4

PMCCT4	PMCT4	Function of pin PCT4
0	0	Output port mode
	1	Input port mode
1	×	Read strobe signal output mode ($\overline{RD}$)

20.3.18 Port CM

Port CM is a 4-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register CM (PCM).
- Input or output mode can be set in 1-bit units by using the port mode register CM (PMCM).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register CM (PMCCM).

Table 20-74 Alternate function pins and port types of port CM

Port		Alternate Function	Remark	Port Type
Port CM	PCM0	$\overline{\text{WAIT}}$ [a]	Wait insertion signal input [a]	2C
	PCM1	-		1
	PCM6	-		
	PCM7	-		

a) Alternate function not available on µPD70F3447.

Caution On the µPD70F3187, in single-chip mode 1 or in ROM-less mode, this port has external control bus function only. Reprogramming of port CM to port mode is not possible in these modes. Reading and writing of the port register PCM and port mode register PMCM is possible but has no effect. Reading of the port mode control register PMCCM is possible and the result is always 01_H. Writing of the port mode control register PMCCM is not possible.

(2) Port register CM (PCM)

The PCM register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins PCM0, PCM1, PCM6 and PCM7.

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

Address $FFFFF00C_H$

Initial Value Undefined

	7	6	5	4	3	2	1	0
PCM	PCM7	PCM6	0	0	0	0	PCM1	PCM0
	R/W	R/W	R	R	R	R	R/W	R/W

Table 20-75 PCM register contents

Bit position	Bit name	Function
7, 6, 1, 0	PCM [7,6,1,0]	Input data control of pin PCMn (in port mode PMCCMn = 0). 0: Input mode (PMCMn = 1): Low level is input Output mode (PMCMn = 0): Low level is output 1: Input mode (PMCMn = 1): High level is input Output mode (PMCMn = 0): High level is output **Note:** n = 0, 1, 6, 7

(3) Port mode register CM (PMCM)

The PMCM register is an 8-bit register that specifies the input or output mode of port pins PCM0, PCM1, PCM6 and PCM7.

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

Address $FFFFF02C_H$

Initial Value FF_H

	7	6	5	4	3	2	1	0
PMCM	PMCM7	PMCM6	1	1	1	1	PMCM1	PMCM0
	R/W	R/W	R	R	R	R	R/W	R/W

Table 20-76 PMCM register contents

Bit position	Bit name	Function
7, 6, 1, 0	PMCM [7,6,1,0]	Input/output mode control of pin PCMn (in port mode PMCCMn = 0). 0: Output mode 1: Input mode **Note:** n = 0, 1, 6, 7

(4) Port mode control register CM (PMCCM)

The PMCCML register is an 8-bit register that specifies the port mode or control mode (alternate function) of port pin PCM0.

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

Address FFFFF04C$_H$

Initial Value Reset input sets this register to 00$_H$ in single-chip mode 0, and to 01$_H$ in ROM-less mode and single-chip mode 1.

Caution

1. Do not set bits 1, 6, and 7 to 1, since the alternate function is not specified.
2. On µPD70F3447 the alternate function is not available. Therefore do not set PMCCM0 bit to 1.
3. On µPD70F3187, in single-chip mode 1 or in ROM-less mode, this register can not be written. Reading is possible and returns 01$_H$.

	7	6	5	4	3	2	1	0
PMCCM	0	0	0	0	0	0	0	PMCCM0
	R/W	R/W	R	R	R	R	R/W	R/W

Table 20-77 PMCCM register contents

Bit position	Bit name	Function
0	PMCCM0	Port mode/control mode specification of pin PCM0

PMCCM0	PMCM0	Function of pin PCM0
0	0	Output port mode
	1	Input port mode
1	×	Wait insertion signal input mode ($\overline{WAIT}$)

20.3.19 Port CD

Port CD is a 4-bit I/O port that can be set to input or output mode in 1-bit units.

(1) Functions

- Input/output data can be specified in 1-bit units by using the port register CD (PCD).
- Input or output mode can be set in 1-bit units by using the port mode register CD (PMCD).
- Port mode or control mode (for alternate function) can be specified in 1-bit units by using the port mode control register CD (PMCCD).

Table 20-78 Alternate function pins and port types of port CD

Port		Alternate Function	Remark	Port Type
Port CD	PCD2	$\overline{BEN0}$ [a]	Byte enable signal output [a]	1
	PCD3	$\overline{BEN1}$ [a]		
	PCD4	$\overline{BEN2}$ [a]		
	PCD5	$\overline{BEN3}$ [a]		

a) Alternate function not available on µPD70F3447.

Note Alternate function not available on µPD70F3447.

Caution On the µPD70F3187, in single-chip mode 1 or in ROM-less mode, this port has external control bus function only. Reprogramming of port CD to port mode is not possible in these modes. Reading and writing of the port register PCD and port mode register PMCD is possible but has no effect. Reading of the port mode control register PMCCD is possible and the result is always $3C_H$. Writing of the port mode control register PMCCD is not possible.

(2) Port register CD (PCD)

The PCD register is an 8-bit register that controls reading the pin levels and writing the output levels of port pins PCD2 to PCD5.

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

Address $FFFFF00E_H$

Initial Value Undefined

	7	6	5	4	3	2	1	0
PCD	0	0	PCD5	PCD4	PCD3	PCD2	0	0
	R	R	R/W	R/W	R/W	R/W	R	R

Table 20-79 PCM register contents

Bit position	Bit name	Function
5...2	PCD[5:2]	Input data control of pin PCDn (in port mode PMCCDn = 0). 0: Input mode (PMCDn = 1): Low level is input Output mode (PMCDn = 0): Low level is output 1: Input mode (PMCDn = 1): High level is input Output mode (PMCDn = 0): High level is output **Note:** n = 2 to 5

(3) Port mode register CD (PMCD)

The PMCD register is an 8-bit register that specifies the input or output mode of port pins PCD2 to PCD5.

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

Address $FFFFF02E_H$

Initial Value FF_H

	7	6	5	4	3	2	1	0
PMCD	1	1	PMCD5	PMCD4	PMCD3	PMCD2	1	1
	R	R	R/W	R/W	R/W	R/W	R	R

Table 20-80 PMCDS register contents

Bit position	Bit name	Function
5...2	PMCD[5:2]	Input/output mode control of pin PCDn (in port mode PMCCDn = 0). 0: Output mode 1: Input mode **Note:** n = 2 to 5

(4) Port mode control register CD (PMCCD)

The PMCCDL register is an 8-bit register that specifies the port mode or control mode (alternate function) of port pins PCD2 to PCD5.

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

Address $FFFFF04E_H$

Initial Value Reset input sets this register to 00_H in single-chip mode 0, and to $3C_H$ in ROM-less mode and single-chip mode 1.

Caution

1. On μPD70F3447 the alternate function is not available. Therefore do not set PMCCDn bits to 1.
2. On μPD70F3187, in single-chip mode 1 or in ROM-less mode, this register can not be written. Reading is possible and returns $3C_H$.

PMCCD

7	6	5	4	3	2	1	0
0	0	PMCCD5	PMCCD4	PMCCD3	PMCCD2	0	0
R	R	R/W	R/W	R/W	R/W	R	R

Table 20-81 PMCCD register contents

Bit position	Bit name	Function
5	PMCCD5	Port mode/control mode specification of pin PCD5
4	PMCCD4	Port mode/control mode specification of pin PCD4
3	PMCCD3	Port mode/control mode specification of pin PCD3
2	PMCCD2	Port mode/control mode specification of pin PCD2

PMCCD5	PMCD5	Function of pin PCD5
0	0	Output port mode
	1	Input port mode
1	×	Byte enable signal output mode ($\overline{BEN3}$)

PMCCD4	PMCD4	Function of pin PCD4
0	0	Output port mode
	1	Input port mode
1	×	Byte enable signal output mode ($\overline{BEN2}$)

PMCCD3	PMCD3	Function of pin PCD3
0	0	Output port mode
	1	Input port mode
1	×	Byte enable signal output mode ($\overline{BEN1}$)

PMCCD2	PMCD2	Function of pin PCD2
0	0	Output port mode
	1	Input port mode
1	×	Byte enable signal output mode ($\overline{BEN0}$)

20.4 Noise Elimination

A timing controller used to secure the noise elimination time is provided for the pins shown in *Table 20-82* below. Input signals that change within the noise elimination time are not internally acknowledged.

Table 20-82 Noise elimination (1/2)

Unit	Pin	Delay Type	Noise Elimination Time	Sampling Clock
Reset	$\overline{\text{RESET}}$	Analog Delay	Several 10 ns (typ.)	
On-chip debug	DRST			
Non-maskable interrupt	P00/NMI	Digital delay	4 to 5 clocks	f_{XX}/16 (250 ns @ f_{XX} = 64 MHz) or f_{XX}/64 (1 µs @ f_{XX} = 64 MHz)
• Maskable Interrupt • Forced output stop function (TMR) • A/D converter (ADC)	P01/INTP0/ESO0 P02/INTP1/ESO1	Analog Delay	60 ns to 200 ns	
	P03/INTP2/ADTRG0 P04/INTP3/ADTRG1	Digital delay	4 to 5 clocks	f_{XX}/16 (250 ns @ f_{XX} = 64 MHz) or f_{XX}/64 (1 µs @ f_{XX} = 64 MHz)
• Maskable Interrupt • Asynchronous serial Interface (UART C)	P30/RXDC0/INTP4 P32/RXDC1/INTP5			
• Maskable Interrupt • Clocked serial interface 3 (CSI3)	P83/SCS300/INTP6 P84/SCS301/INTP7 P85/SCS302/INTP8 P93/SCS310[a]/INTP9 P94/SCS311[a]/INTP10 P95/SCS312[a]/INTP11			
Timer ENC (TMNEC10) [a]	P100/TCLR0/TICC00/TOP81 [b] P101/TCUD0/TICC01 [b] P102/TIUD0/TO1 [b]			
Timer P (TMP)	P10/TIP00/TEVTP1/TOP00 P11/TIP01/TTRGP1/TOP01 P12/TIP10/TTRGP0/TOP10 P13/TIP11/TEVTP0/TOP11 P14/TIP20/TEVTP3/TOP20 P15/TIP21/TTRGP3/TOP21 P16/TIP30/TTRGP2/TOP30 P17/TIP31/TEVTP2/TOP31 P20/TIP40/TEVTP5/TOP40 P21/TIP41/TTRGP5/TOP41 P22/TIP50/TTRGP4/TOP50 P23/TIP51/TEVTP4/TOP51 P24/TIP60/TEVTP7/TOP60 P25/TIP61/TTRGP7/TOP61 P26/TIP70/TTRGP6/TOP70 P27/TIP71/TEVTP6/TOP71	Digital delay	4 to 5 clocks	f_{XX}/16 (250 ns @ f_{XX} = 64 MHz) or f_{XX}/64 (1 µs @ f_{XX} = 64 MHz)

Table 20-82 Noise elimination (2/2)

Unit	Pin	Delay Type	Noise Elimination Time	Sampling Clock
Timer R (TMR)	P60/TOR10/TTRGR1 P61/TOR11/TIR10 P62/TOR12/TIR11 P63/TOR13/TIR12 P64/TOR14/TIR13 P67/TOR17/TEVTR1	Digital delay	4 to 5 clocks	f_{XX}/16 (250 ns @ f_{XX} = 64 MHz) or f_{XX}/64 (1 µs @ f_{XX} = 64 MHz)
Timer T (TMT)	P70/TIT00/TEVTT1/TOT00 P71/TIT01/TTRGT1/TOT01 P72/TECRT0/INTP12 P73/TIT10/TTRGT0/TOT10 P74/TIT11/TEVTT0/TOT11 P75/TECRT1/AFO	Digital delay	4 to 5 clocks	f_{XX}/16 (250 ns @ f_{XX} = 64 MHz) or f_{XX}/64 (1 µs @ f_{XX} = 64 MHz)

a) Not available on µPD70F3447.
b) No noise elimination on µPD70F3447 available.

Caution The noise elimination function is valid only in control mode (alternate function).

(1) Noise elimination control register (NRC)

The NRC register is an 8-bit register that specifies the sampling clock that is used to eliminate digital noise of input pins.

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

Address $FFFFF7A0_H$

Initial Value 00_H

Caution

1. If the input pulse lasts for the duration of 4 to 5 clocks, it is undefined whether the pulse is detected as a valid edge or eliminated as noise. So that the pulse is actually detected as a valid edge, the same pulse level must be input for the duration of 5 clocks or more.
2. If noise is generated in synchronization with the sampling clock, eliminate the noise by attaching a filter to the input pin.
3. Noise is not eliminated if the corresponding pin is used as normal input port pin.

NRC

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 20-83 NRC register contents

Bit position	Bit name	Function
7	NRC7	Noise elimination clock setting for pin group 7 [a]. 0: f_{XX}/16 (250 ns @ f_{XX} = 64 MHz) 1: f_{XX}/64 (1 µs @ f_{XX} = 64 MHz)
6	NRC6	Noise elimination clock setting for pin group 6 [a]. 0: f_{XX}/16 (250 ns @ f_{XX} = 64 MHz) 1: f_{XX}/64 (1 µs @ f_{XX} = 64 MHz)
5	NRC5	Noise elimination clock setting for pin group 5 [a]. 0: f_{XX}/16 (250 ns @ f_{XX} = 64 MHz) 1: f_{XX}/64 (1 µs @ f_{XX} = 64 MHz)
4	NRC4	Noise elimination clock setting for pin group 4 [a]. 0: f_{XX}/16 (250 ns @ f_{XX} = 64 MHz) 1: f_{XX}/64 (1 µs @ f_{XX} = 64 MHz)
3	NRC3	Noise elimination clock setting for pin group 3 [a]. 0: f_{XX}/16 (250 ns @ f_{XX} = 64 MHz) 1: f_{XX}/64 (1 µs @ f_{XX} = 64 MHz)
2	NRC2	Noise elimination clock setting for pin group 2 [a]. 0: f_{XX}/16 (250 ns @ f_{XX} = 64 MHz) 1: f_{XX}/64 (1 µs @ f_{XX} = 64 MHz)
1	NRC1	Noise elimination clock setting for pin group 1 [a]. 0: f_{XX}/16 (250 ns @ f_{XX} = 64 MHz) 1: f_{XX}/64 (1 µs @ f_{XX} = 64 MHz)
0	NRC0	Noise elimination clock setting for P00/NMI pin. 0: f_{XX}/16 (250 ns @ f_{XX} = 64 MHz) 1: f_{XX}/64 (1 µs @ f_{XX} = 64 MHz)

a) refer to *Table 20-84, "Pin groups of noise elemination function," on page 942*

Table 20-84 Pin groups of noise elemination function

Pin group	Associated pins
1	(on µPD70F3187) P03/INTP2/ADTRG0, P04/INTP3/ADTRG1, P30/RXDC0/INTP4, P32/RXDC1/INTP5, P83/SCS300/INTP6, P84/SCS301/INTP7, P85/SCS302/INTP8, P93/SCS310/INTP9, P94/SCS310/INTP10, P95/SCS310/INTP11 (on µPD70F3447) P03/INTP2/ADTRG0, P04/INTP3/ADTRG1, P30/RXDC0/INTP4, P32/RXDC1/INTP5, P83/SCS300/INTP6, P84/SCS301/INTP7, P85/SCS302/INTP8, P93/INTP9, P94/INTP10, P95/INTP11
2	(on µPD70F3187) P100/TCLR0/TICC00/TOP81, P101/TCUD0/TICC01, P102/TIUD0/TO1, P70/TIT00/TEVTT1/TOT00, P71/TIT01/TTRGT1/TOT01, P72/TECRT0/INTP12, P73/TIT10/TTRGT0/TOT10, P74/TIT11/TEVTT0/TOT11, P75/TECRT1/AFO (on µPD70F3447) P70/TIT00/TEVTT1/TOT00, P71/TIT01/TTRGT1/TOT01, P72/TECRT0/INTP12, P73/TIT10/TTRGT0/TOT10, P74/TIT11/TEVTT0/TOT11, P75/TECRT1/AFO
3	P10/TIP00/TEVTP1/TOP00, P11/TIP01/TTRGP1/TOP01, P12/TIP10/TTRGP0/TOP10, P13/TIP11/TEVTP0/TOP11
4	P14/TIP20/TEVTP3/TOP20, P15/TIP21/TTRGP3/TOP21, P16/TIP30/TTRGP2/TOP30, P17/TIP31/TEVTP2/TOP31
5	P20/TIP40/TEVTP5/TOP40, P21/TIP41/TTRGP5/TOP41, P22/TIP50/TTRGP4/TOP50, P23/TIP51/TEVTP4/TOP51
6	P24/TIP60/TEVTP7/TOP60, P25/TIP61/TTRGP7/TOP61, P26/TIP70/TTRGP6/TOP70, P27/TIP71/TEVTP6/TOP71
7	P60/TOR10/TTRGR1, P61/TOR11/TIR10, P62/TOR12/TIR11, P63/TOR13/TIR12, P64/TOR14/TIR13, P67/TOR17/TEVTR1

Chapter 21 Reset Function

21.1 Features

- Reset function by $\overline{\text{RESET}}$ input
- Forced reset function by DCU (refer to chapter *"On-Chip Debug Function (OCD)" on page 948*)
- Reset generator (RG) eliminates noise from the $\overline{\text{RESET}}$ pin.

21.2 Configuration

During a system reset, most pins (all except the DCK, $\overline{\text{DRST}}$, DMS, DDI, DDO, $\overline{\text{RESET}}$, X2, V_{DD10} to V_{DD15}, V_{SS10} to V_{SS15}, V_{DD30} to V_{DD37}, V_{SS30} to V_{SS37}, CV_{DD}, CV_{SS}, AV_{DD}, AV_{REF0}, AV_{REF1}, AV_{SS0} and AV_{SS1} 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.

21.3 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 $\overline{\text{RESET}}$ pin level changes from low to high, the oscillation stabilization time counter (OSTC) is started after analog delay by the reset generator. At that time the OSTC elapses, the PLL circuit will be enabled and the internal PLL stabilization time counter (PSTC) is started using the oscillator output clock (f_X). After 2^{14} oscillator clocks (f_X) the PLL output clock becomes the system clock (f_{XX}) and the internal system reset is released synchronously to the system clock.

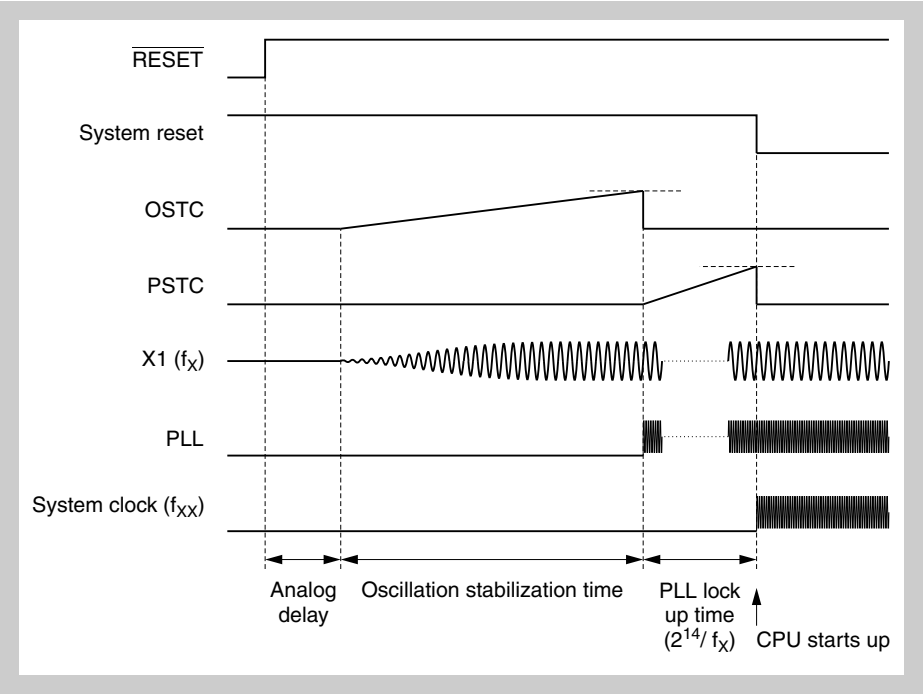


Figure 21-1 Reset timing

Remarks

1. If no clock is supplied to V850E/PH2 (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. The on-chip debug function can force the activation of the system reset independently from input level of the $\overline{\text{RESET}}$ pin.

Chapter 22 Internal RAM Parity Check Function

The V850E/PH2 microcontroller is provided with a parity check function for the internal RAM (iRAM).

22.1 Features

- Maskable interrupt (INTPERR) on detection of parity error
- Indication of internal RAM address of detected parity error
- Indication of erroneous byte within 32-bit word

22.2 Operation

Each byte of data stored in the internal RAM is checked by its parity bit. A maskable interrupt (INTPERR) is generated, if a parity mismatch is detected on iRAM read operation. In this case the address of the erroneous data is latched in the RAMPADD register and the erroneous byte(s) are indicated in the RAMERR register.

Caution 1. Ensure that all internal RAM data is initialized on a word (32-bit) base by a write operation before first read access is made. It is important to initialize the whole memory word, even if only byte or half word is used. Otherwise a parity fail may be indicated mistakenly.

22.3 Control Registers

(1) Internal RAM parity error status register (RAMERR)

The RAMERR register is an 8-bit register that reflects the parity error flags of the four bytes of one word (32 bits) in the internal RAM. The corresponding error flag (bits RAE0 to RAE3) is set and a maskable interrupt (INTPERR) is generated, if a parity error is detected during read access.

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

Address $FFFFF4C0_H$

Initial Value 00_H. This register is cleared by any reset.

Note The RAE[3:0] bits can be both, read and written, but only 0 can be written to these bits in order to cleared them. These bits cannot be set by writing 1 to them.

	7	6	5	4	3	2	1	0
RAMERR	0	0	0	0	RAE3	RAE2	RAE1	RAE0
	R	R	R	R	R/W	R/W	R/W	R/W

Table 22-1 RAMERR register contents

Bit position	Bit name	Function
3	RAE3	Indicates the internal RAM parity error for bits 24 to 31 of the concerned word. 0: No parity error detected in internal RAM. 1: Parity error caused by bits 24 to 31.
2	RAE2	Indicates the internal RAM parity error for 16 to 23 of the concerned word. 0: No parity error detected in internal RAM. 1: Parity error caused by bits 16 to 23.
1	RAE1	Indicates the internal RAM parity error for 8 to 15 of the concerned word. 0: No parity error detected in internal RAM. 1: Parity error caused by bits 8 to 15.
0	RAE0	Indicates the internal RAM parity error for bits 0 to 7 of the concerned word. 0: No parity error detected in internal RAM. 1: Parity error caused by bits 0 to 7.

(2) Internal RAM parity error address register (RAMPADD)

The RAMPADD register is a 16-bit register that latches the internal RAM address causing the first parity error after hardware reset was released or RAMERR register was cleared.

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

Address $FFFFF4C2_H$

Initial Value 8000_H. This register is cleared by any reset.

Caution

1. Do not read the RAMPADD register, when all internal RAM parity error flags RAMERR.RAE[3:0] are cleared. If a parity error is detected and the RAMPADD register is read before the respective RAE[3:0] flag is set, the read value might be invalid.
2. Bit 15 of the RAMPADD register is always 1. This does not reflect the correct address bit 15 of the internal RAM, which starts at location FFF0000H.

RAMPADD

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	RAMPADD[14:2]													0	0
R	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	R

Table 22-2 RAMERR register contents

Bit position	Bit name	Function
14 to 0	RAMPADD[14:0]	Internal RAM address of the 32-bit word causing the parity error. **Note:** Bits 0 and 1 of the RAMPADD register are always 0, because the parity check function is aligned on 32-bit words.

Chapter 23 On-Chip Debug Function (OCD)

An on-chip debug unit is provided in the V850E/PH2 microcontroller and realizes stand-alone on-chip debugging of the V850E/PH2 microcontroller by connecting a N-Wire type emulator.

Caution The debug function explained in this chapter is based on the usage of the Renesas IE-V850E1-CD-NW, or IE-V850MINI emulator.
When using a another N-Wire type emulator, refer to the manual of the used debugger.

23.1 Function Overview

23.1.1 On-chip debug unit type

The on-chip debug unit incorporated in the V850E/PH2 microcontroller is RCU1 (run control unit 1). The on-chip unit incorporated differs depending on the microcontroller, and also features different functions.

23.1.2 Debug function

For details of the debug function, refer to the corresponding debugger operation user's manual.

(1) Debug interface

This interface establishes communication with the host machine by using the $\overline{\text{DRST}}$, DCK, DMS, DDI, and DDO signals, via a N-Wire type emulator. The communication specifications of N-Wire are used for this interface. It does not support a boundary scan function.

(2) On-chip debug

On-chip debugging can be performed by providing wiring and connectors for debugging on the target system.

Connect a N-Wire type emulator to the emulator connector.

(3) Forced reset function

The V850E/PH2 can be forcibly reset.

(4) Break reset function

The CPU can be started in the debug mode immediately after resetting the CPU has been cleared.

(5) Forced break function

Execution of the user program can be forcibly stopped (however, the handler of the illegal instruction code exception (first address: 00000060_H) cannot be used).

(6) Hardware break function

Two common instruction fetch/access breakpoints can be used. By using the instruction breakpoint, program execution can be suspended at an arbitrary address. By using the access breakpoint, program execution can be suspended by data-accessing an arbitrary address.

(7) Software break function

In addition to the hardware break function, a software break function is available. Up to eight software breakpoints can be set in the internal ROM area. The number of software breakpoints that can be set in the internal RAM area differs depending on the debugger used.

(8) Debug monitor function

During debugging, a memory space for debugging that differs from the user memory space is used (background monitor format). The user program can be executed starting from any address.

While execution of the user program is stopped, the user resources (such as memory and I/O) can be read or written, and the user program can be downloaded.

(9) Mask function

$\overline{\text{RESET}}$, $\overline{\text{WAIT}}$, NMI and all maskable interrupt request signals can be masked.

(10) Timer function

The execution time of the user program can be measured.

23.2 Connection with N-Wire Type Emulator

To connect a N-Wire type emulator, it is necessary to mount an emulator connector and circuit for connection on the target system.

Select either the KEL connector, MICTOR connector (Part number: 2-767004-2, distributor: Tyco Electronics AMP K.K.), or 2.54 mm pitch 20-pin general-purpose connector as the emulator connector. Connectors other than the KEL connector may not be supported, depending on the emulator, so when using a connector, refer to the manual of the emulator used.

23.2.1 KEL connector

When the IE-V850E1-CD-NW is used, use of the following connector is recommended.

Part number

- 8830E-026-170S: Straight type
- 8830E-026-170L: Right-angle type

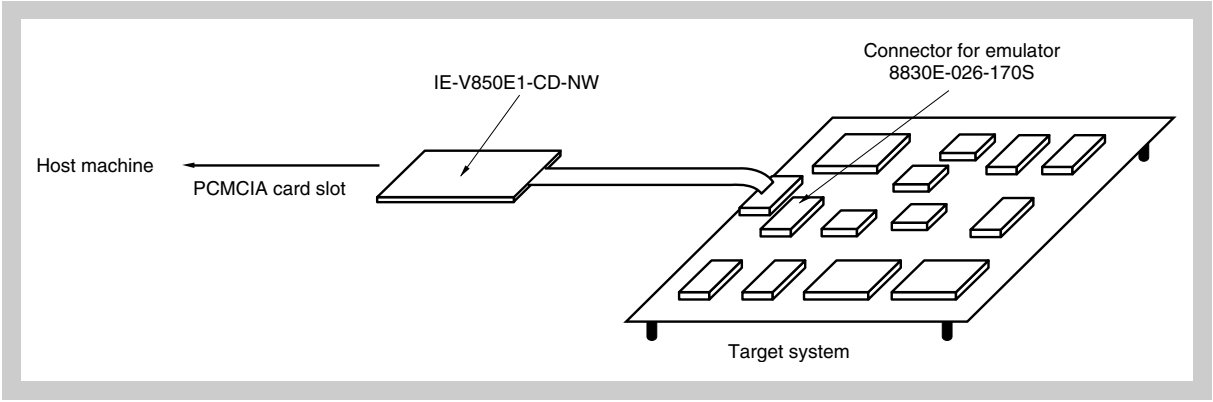


Figure 23-1 Connecting N-Wire type emulator (IE-V850E1-CD-NW)

(1) Pin configuration

Figure 23-2 shows the pin configuration of the emulator connector (target system side), and Table 23-1 shows the pin functions.

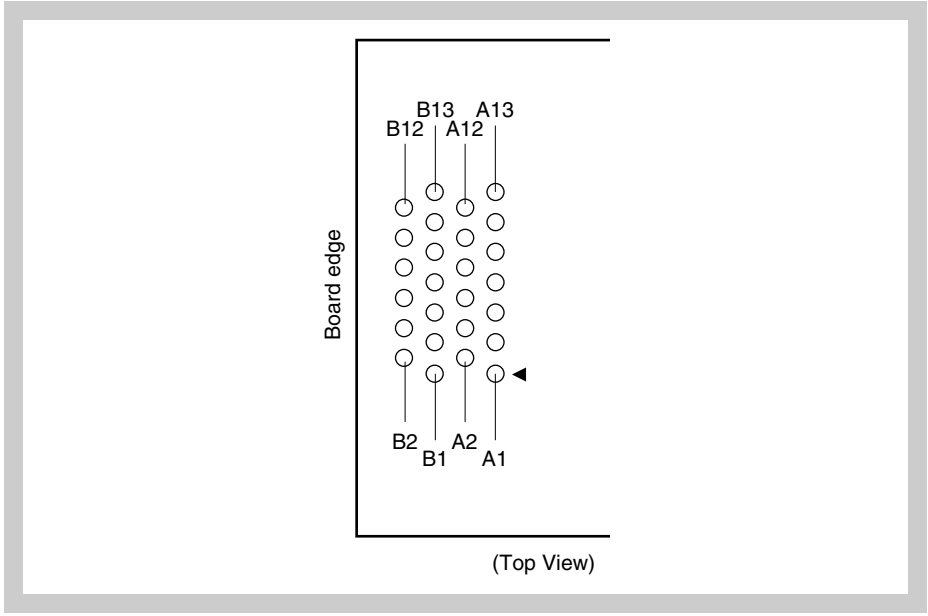


Figure 23-2 Pin configuration of emulator connector (on target system side)

Caution Design the board based on the dimensions of the connector when actually mounting the connector on the board.

(2) Pin functions

The following table shows the pin functions of the emulator connector (on the target system side).

Table 23-1 Pin functions of connector for IE-V850MINI (on target system side)

Pin No.	Pin Name	I/O [a]	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	$\overline{\text{RESET}}$	-	System reset input (leave open when not used)[b]
A13	FLMD0	Input	Control signal for flash memory downloading[b]
B1	GND	-	-
B2	GND	-	-
B3	GND	-	-
B4	GND	-	-
B5	GND	-	-
B6	GND	-	-
B7	GND	-	-
B8	GND	-	-
B9	GND	-	-
B10	GND	-	-
B11	PORT0_IN	-	(Connect to GND)
B12	PORT1_IN	-	(Connect to GND)
B13	VDD	-	3.3 V input (for monitoring power application to target)

a) Input/output is as viewed from the device side.
b) For connecting the signal refer to the manual of the emulator.

Caution

1. The processing of the pins not incorporated in the V850E/PH2 or unused pins depends on the emulator used.
2. The pattern on the target board must satisfy the following conditions.
   - Keep the pattern length to within 100 mm.
   - Shield the clock signal with GND.

(3) IRecommended circuit example

The following figure shows an example of the recommended circuit of the emulator connector (on the target system side).

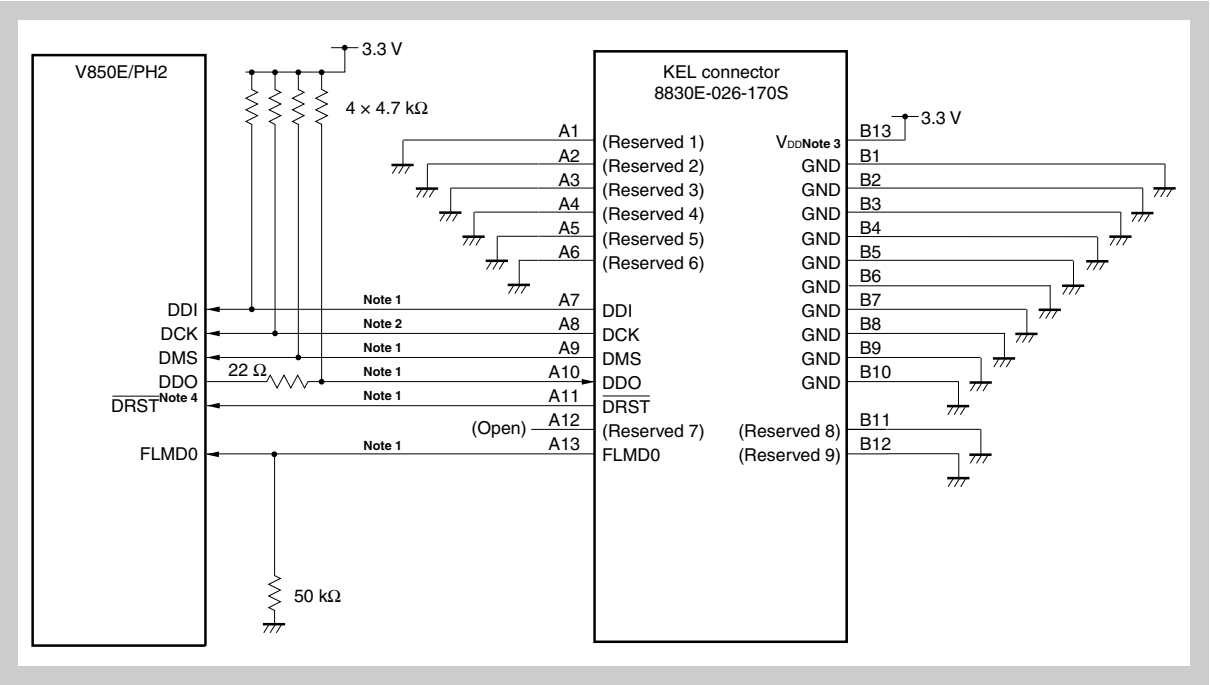


Figure 23-3 Example of recommended emulator connection of V850E/PH2

Note

1. Keep the pattern length to within 100 mm.
2. Shield the DCK signal with GND.
3. For detecting power supply to the target board. Connect to the N-Wire interface voltage.
4. When $\overline{\text{DRST}}$ pin is high level: On-chip debug mode
   When $\overline{\text{DRST}}$ pin is low level or open: Normal operation mode
   $\overline{\text{DRST}}$ pin is connected to V_{SS3} via an internal pull-down resistor

Caution

1. The DDO signal is 3.3 V output, and the input level of the DDI, DCK, DMS, and $\overline{\text{DRST}}$ signals is TTL level.
2. A 3.3 V interface may not be supported, so a level shifter may be required by some N-Wire type emulators. Refer to the manual of the emulator used. Note that the IE-V850E1-CD-NW supports a 5 V interface.

23.3 Precautions

(1) Number of flash memory rewrites

The flash memory of the device used in debugging is rewritten during debugging, so the number of flash memory rewrites cannot be guaranteed. Therefore, do not use the device used in debugging for a mass production product.

(2) Reset during program execution

If a reset ($\overline{\text{RESET}}$ signal input from the target system or reset input by an internal reset source) occurs during RUN (program execution), the break function may malfunction.

(3) Masked reset

Even if reset is masked by using the mask function, the I/O buffers (port pins, etc.) are set to the reset state when the $\overline{\text{RESET}}$ signal is input.

(4) Reset during break

$\overline{\text{RESET}}$ signal input during a break is masked.

Chapter 24 Flash Memory

The V850E/PH2 and has a 512 KB on-chip flash memory configured as 128 blocks of 4 KB block size.

24.1 Features

- 4-byte/1-clock access (when instruction is fetched)
- Capacity:
  - μPD70F3481: 512 KB
  - μPD70F3447: 384 KB
- Block size:
  - μPD70F3481: 128 blocks of 4 KB
  - μPD70F3447: 96 blocks of 4 KB
- Write voltage: Erase/write with single voltage
- Rewriting method
  - Rewriting by communication with dedicated flash programmer via serial interface (on-board/off-board programming)
  - Rewriting flash memory by user program (self-programming)
- 64 KB boot block cluster with write prohibit function supported (protection function)
- Interrupts can be acknowledged during self programming.

24.2 Memory Configuration

The internal flash memory area is divided into 4 KB blocks (128 blocks for μPD70F3187, and 96 blocks for μPD70F3447)and can be programmed/erased in block units. All the blocks can also be erased at once.

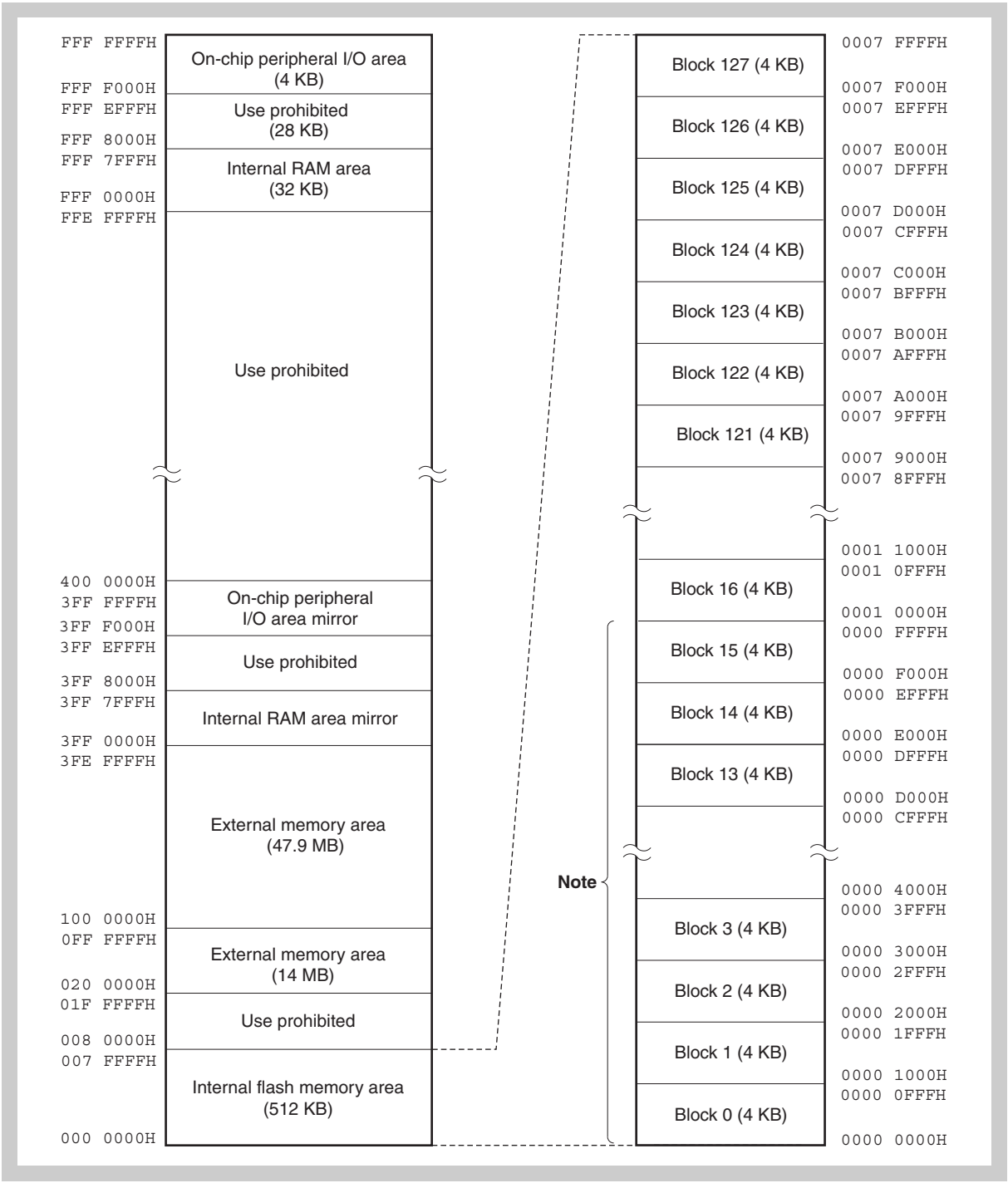


Figure 24-1 Flash memory mapping of μPD70F3187

Note Blocks 0 to 15 (64 KB): Boot block cluster

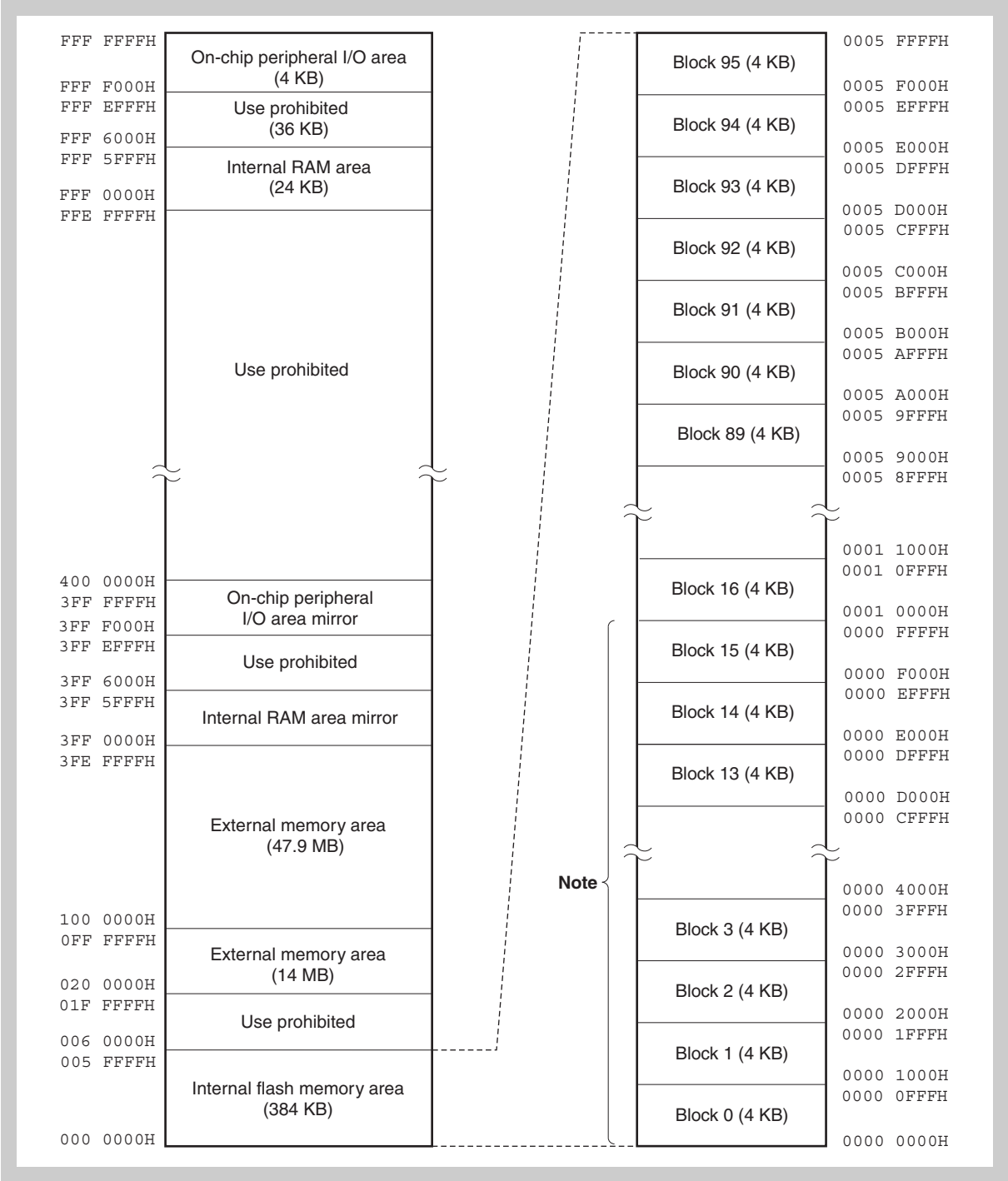


Figure 24-2 Flash memory mapping of µPD70F3447

Note Blocks 0 to 15 (64 KB): Boot block cluster

24.3 Functional Outline

The internal flash memory of the V850E/PH2 can be rewritten by using the rewrite function of the dedicated flash programmer, regardless of whether the V850E/PH2 has already been mounted on the target system or not (off-board/on-board programming).

In addition, different functions are implemented to protect unwanted Flash access via the programmer interface and to protect the boot area from any unwanted modification

The rewrite function using the user program (self programming) is ideal for an application where it is assumed that the program is changed or extended during or after production/shipment of the target system. Interrupt servicing is supported during self programming, so that the flash memory can be rewritten under various conditions, such as while communicating with an external device.

Table 24-1 Rewrite Method

Rewrite Method	Functional Outline	Operation Mode
Off-board programming	Flash memory can be rewritten before the device is mounted on the target system, by using a dedicated flash programmer and a dedicated program adapter board.	Flash memory programming mode
On-board programming	Flash memory can be rewritten after the device is mounted on the target system, by using a dedicated flash programmer.	
Self-programming	Flash memory can be rewritten by executing a user program that has been written to the flash memory in advance by means of off-board/on-board programming. (During self-programming, instructions cannot be fetched from or data access cannot be made to the internal flash memory area. Therefore, the rewrite program must be transferred to the internal RAM or external memory in advance.)	Normal operation mode

Table 24-2 Basic Functions (1/2)

Function	Functional Outline	Support by	
		On-Board/Off-Board Programming	Self Programming
Block erasure	The contents of specified memory blocks are erased.	yes	yes
Chip erasure	The contents of the entire memory area are erased all at once.	yes	no
Write	Writing to specified addresses, and a verify check to see if write level is secured are performed.	yes	yes
Verify/check sum	Data read from the flash memory is compared with data transferred from the flash programmer.	yes	no (Can be read by user program)

Table 24-2 Basic Functions (2/2)

Function	Functional Outline	Support by	
		On-Board/Off-Board Programming	Self Programming
Blank check	The erasure status of the entire memory is checked.	yes	yes
Read	The content of specified addresses can be read.	yes	yes
Protection setting	Setting of protection flags to prohibit programming interface commands (write, block erase, chip erase and read) and to prohibit boot block cluster modification via self-programming	yes	yes

The following *Table 24-3* lists the protection functions. After shipment no protection feature is set on the device. Furthermore, after chip erase by a dedicated programmer (PG-FP4) the protection is reset. Each protection function can be used in combination with the others at the same time.

Table 24-3 Protection Functions

Function	Functional Outline	Operation	
		On-Board/Off-Board Programming	Self-Programming
Block erase command prohibit	Execution of a block erase command on all blocks is prohibited. Setting of prohibition can be initialized by execution of a chip erase command.	Block erase command: × Chip erase command: ⊕ Program command: ⊕	Can always be read or rewritten regardless of protection function setting
Chip erase command prohibit	Execution of block erase and chip erase commands on all the blocks is prohibited. Once prohibition is set, setting of prohibition cannot be initialized because the chip erase command cannot be executed.	Block erase command: × Chip erase command: × Program command: ⊕	
Program command prohibit	Write and block erase commands on all the blocks are prohibited. Setting of prohibition can be initialized by execution of the chip erase command.	Block erase command: × Chip erase command: ⊕ Program command: ×	
Read command prohibit	Read of memory content is prohibited.	Read command: ×	

Remark ×: Operation prohibited
⊕: Operation executable

24.4 Rewriting by Dedicated Flash Programmer

The flash memory can be rewritten by using a dedicated flash programmer after the V850E/PH2 is mounted on the target system (on-board programming). The flash memory can also be rewritten before the device is mounted on the target system (off-board programming) by using a dedicated programming adapter.

24.4.1 Programming environment

The following shows the environment required for rewriting the flash memory of the V850E/PH2:

Figure 24-3 Environment Required for Writing Programs to Flash Memory

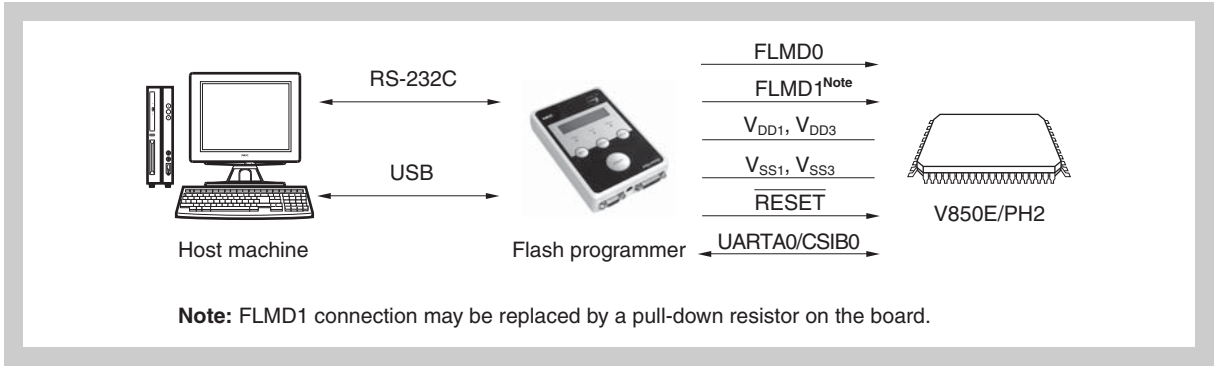


Figure 24-4 Environment required for rewriting the flash memory

A host machine is required for controlling the flash programmer.

UARTC0 or CSIB0 is used for the interface between the flash programmer and the V850E/PH2 to perform writing, erasing, etc. A dedicated program adapter is required for off-board writing.

24.4.2 Communication mode

Communication between the flash programmer and the V850E/PH2 is performed by serial communication using the UARTC0 or CSIB0 interfaces of the V850E/PH2.

(1) UARTC0

Transfer rate: 9,600 to 153,600 bps

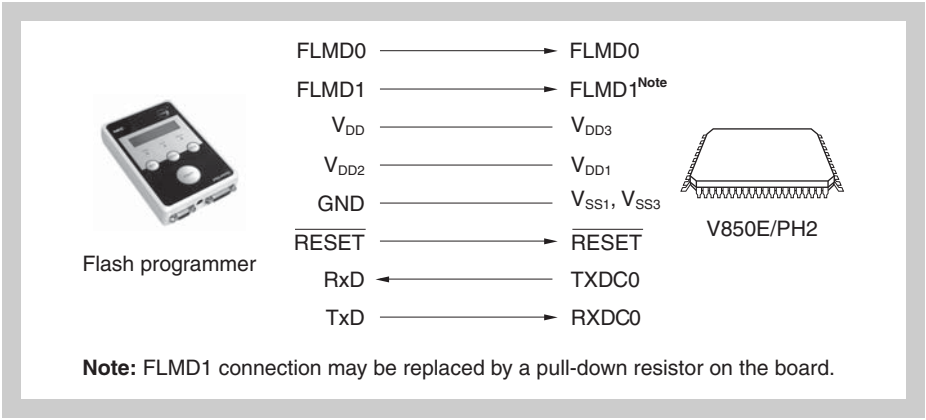


Figure 24-5 UARTC0 communication with flash programmer

(2) CSIB0

Serial clock: 2.4 kHz to 2.5 MHz (MSB first)

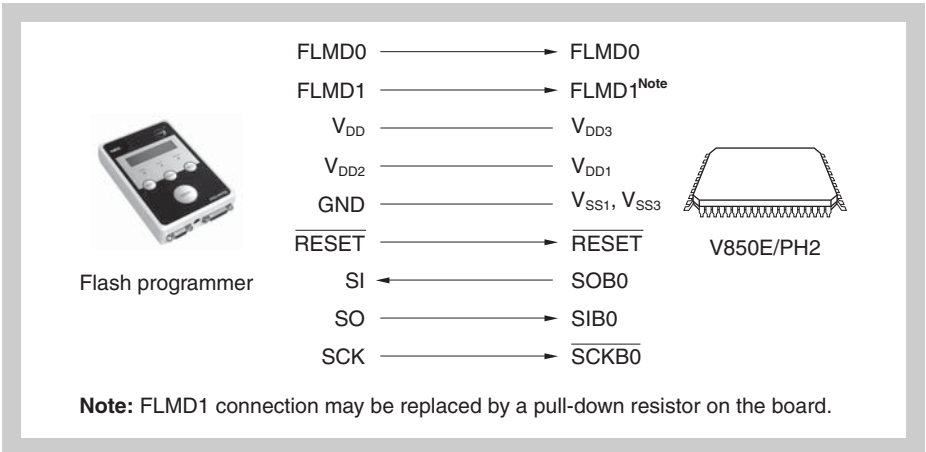
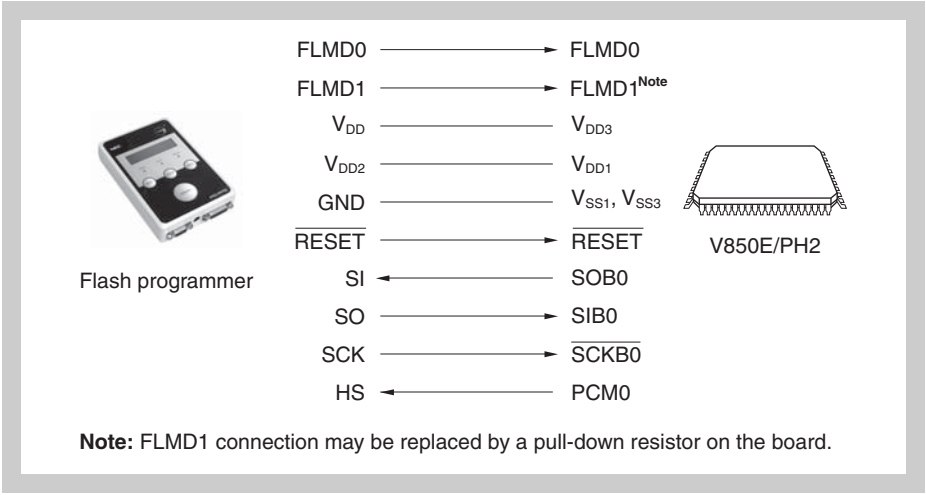


Figure 24-6 CSIB0 communication with flash programmer

(3) CSIB0 + HS (Handshake)

Serial clock: 2.4 kHz to 2.5 MHz (MSB first)



Note: FLMD1 connection may be replaced by a pull-down resistor on the board.

Figure 24-7 CSIB0 + HS communication with flash programmer

24.4.3 Pin connections 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.

When the PG-FP5 is used as the dedicated flash programmer, it generates the following signals to the V850E/PH2. For details, refer to the PG-FP5 User's Manual (R20UT0008EJ).

Table 24-4 Pin connections to signals of flash programmer (PG-FP5)

PG-FP5			V850E/PH2	Processing for Connection [a]		
Signal Name	**I/O**	**Pin Function**	**Pin Name**	**UARTC0**	**CSIB0**	**CSIB0 + HS**
FLMD0	Output	Write enable/disable	FLMD0	⊕	⊕	⊕
FLMD1	Output	Write enable/disable	FLMD1	⊕ [β]	⊕ [b]	⊕ [b]
VDD	–	V_{DD} voltage generation/ Voltage monitor	VDD3x	⊕ [c]	⊕ [c]	⊕ [c]
VDD2	–	V_{DD2} voltage generation	VDD1x	× [d]	× [d]	× [d]
GND	–	Ground	VSS1x, VSS3x	⊕	⊕	⊕
CLK	Output	Clock output	X1, X2	× [e]	× [e]	× [e]
$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$	⊕	⊕	⊕
SI/RXD	Input	Receive signal	SIB0/RXDC0	⊕	⊕	⊕
SO/TXD	Output	Transmit signal	SOB0/TXDC0	⊕	⊕	⊕
SCK	Output	Transfer clock	$\overline{\text{SCKB0}}$	×	⊕	⊕
H/S	Input	Handshake signal	PCM0	×	×	⊕

a) ⊕: Must be connected.
×: Do not need to be connected.
b) Connect to GND via pull-down resistor.
c) Connect these pins to supply power from the PG-FP5. When power is supplied externally to the target board, the voltage is monitored by PG-FP5.
d) Connect these pins to supply power from the PG-FP5, or supply power externally to the target board.
e) Clock supply must be provided by an oscillator on the target board. Clock supply from PG-FP5 is not supported for V850E/PH2.

24.4.4 Flash memory control

The following shows the procedure for manipulating the flash memory.

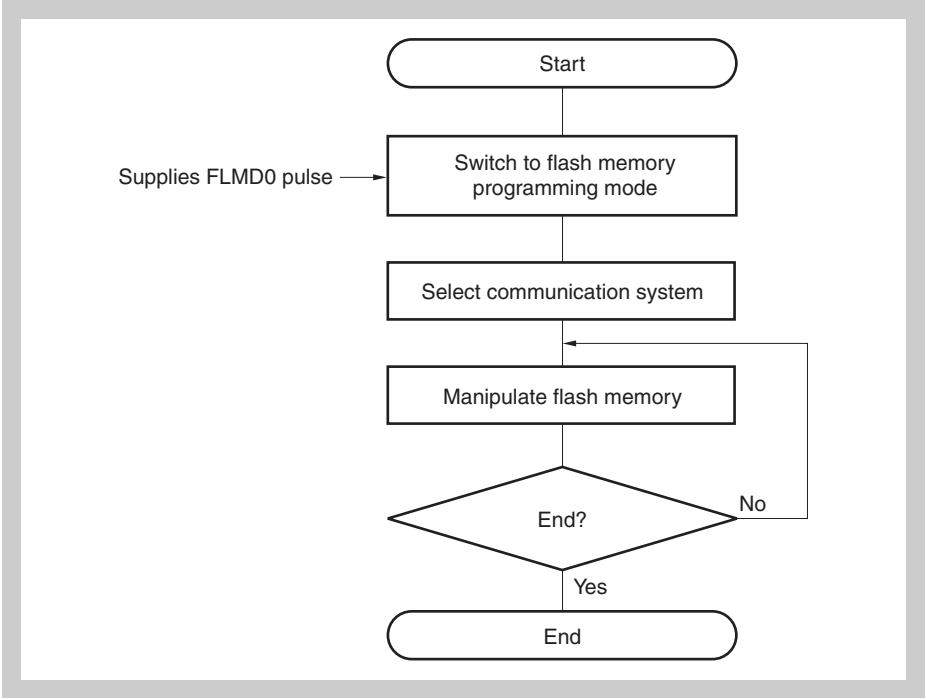


Figure 24-8 Procedure for manipulating flash memory

24.4.5 Selection of communication mode

In the V850E/PH2, the communication mode is selected by inputting pulses (11 pulses max.) to the FLMD0 pin after switching to the flash memory programming mode. The FLMD0 pulse is generated by the dedicated flash programmer.

The following shows the relationship between the number of pulses and the communication mode.

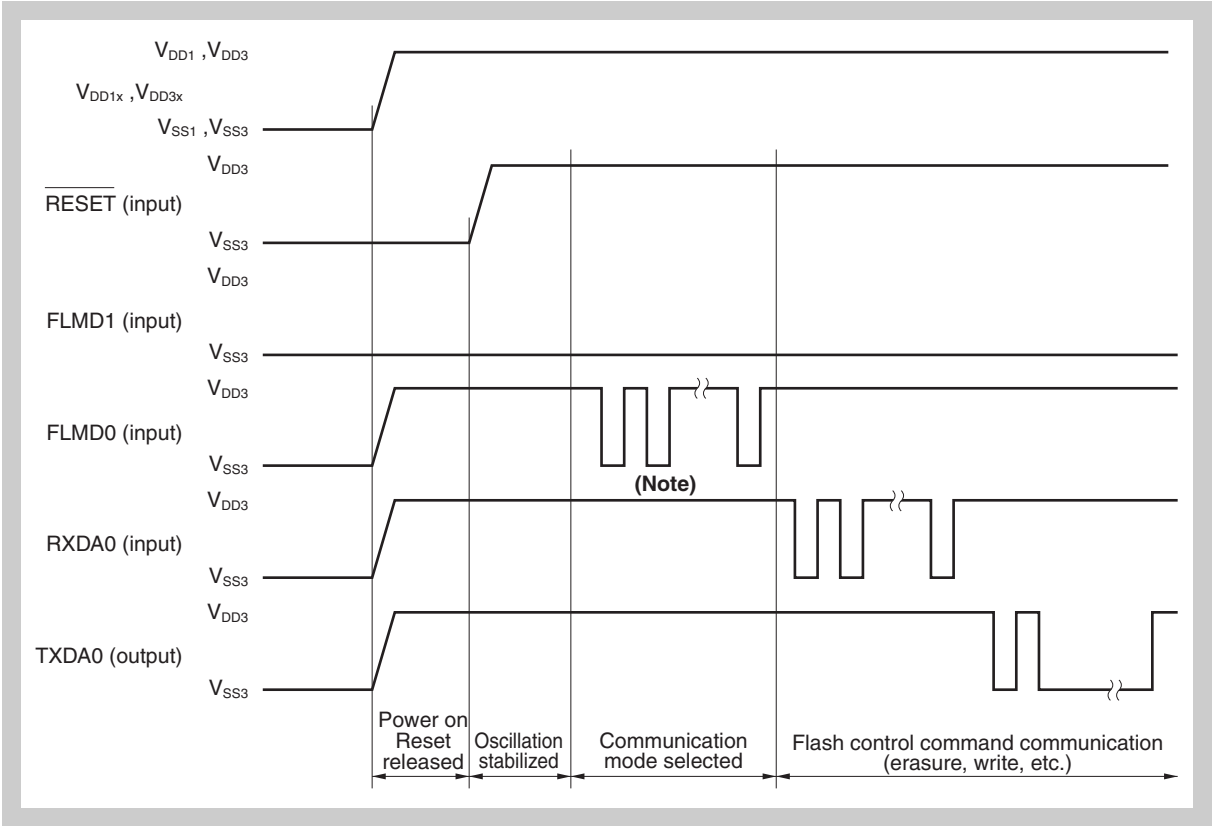


Figure 24-9 Selection of communication mode

Note The number of clocks is as follows depending on the communication mode.

Number of FLMD0 Pulses	Communication Mode	Remarks
0	UARTC0	Communication rate: 9,600 bps (after reset), LSB first
8	CSIB0	V850E/PH2 performs slave operation, MSB first
11	CSIB0 + HS	
Others	RFU	Setting prohibited

Caution When UARTC0 is selected, the receive clock is calculated based on the reset command sent from the flash programmer after receiving the FLMD0 pulse.

24.4.6 Communication commands

The V850E/PH2 communicates with the dedicated flash programmer by means of commands. The signals sent from the dedicated flash programmer to the V850E/PH2 are called “commands”. The response signals sent from the V850E/PH2 to the dedicated flash programmer are called “response commands”.

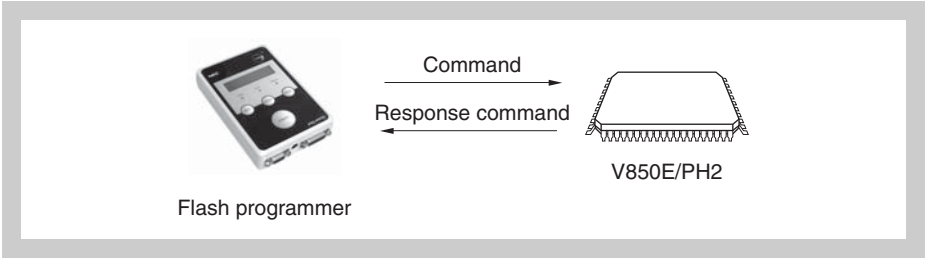


Figure 24-10 Communication commands

The following shows the commands for flash memory control in the V850E/PH2. All of these commands are issued from the dedicated flash programmer, and the V850E/PH2 performs the processing corresponding to the commands.

Table 24-5 Communication commands

Classification	Command Name	Support [a]			Function
		UARTC0	CSI0	CSI0 + HS	
Blank check	Block blank check command	⊕	⊕	⊕	Checks if the contents of the memory in the specified block have been correctly erased.
Erase	Chip erase command	⊕	⊕	⊕	Erases the contents of the entire memory.
	Block erase command	⊕	⊕	⊕	Erases the contents of the memory of the specified block.
Write	Write command	⊕	⊕	⊕	Writes the specified address range, and executes a contents verify check.
Verify	Verify command	⊕	⊕	⊕	Compares the contents of memory in the specified address range with data transferred from the flash programmer.
	Checksum command	⊕	⊕	⊕	Reads the checksum in the specified address range.
Read	Read command	⊕	⊕	⊕	Reads the specified address range.
System setting, control	Silicon signature command	⊕	⊕	⊕	Reads silicon signature information.
	protection setting command	⊕	⊕	⊕	Disables the chip erase command, enables the block erase command, and disables the write command.

a) ⊕: Operation supported
×: Operation not supported

24.4.7 Pin connection

When performing on-board writing, mount a connector on the target system to connect to the dedicated flash programmer. Also, incorporate a function on-board to switch from the normal operation mode to the flash memory programming mode.

In the flash memory programming mode, all the pins not used for flash memory programming become the same status as that immediately after reset. Therefore, pin handling is required when the external device does not acknowledge the status immediately after a reset.

(1) FLMD0 pin

In the normal operation mode, input a voltage of V_{SS3} level to the FLMD0 pin. In the flash memory programming mode, supply a write voltage of V_{DD3} level to the FLMD0 pin.

Because the FLMD0 pin serves as a write protection pin in the self programming mode, a voltage of V_{DD3} level must be supplied to the FLMD0 pin via port control, etc., before writing to the flash memory. For details, refer to the self-programming application note (U16929E).

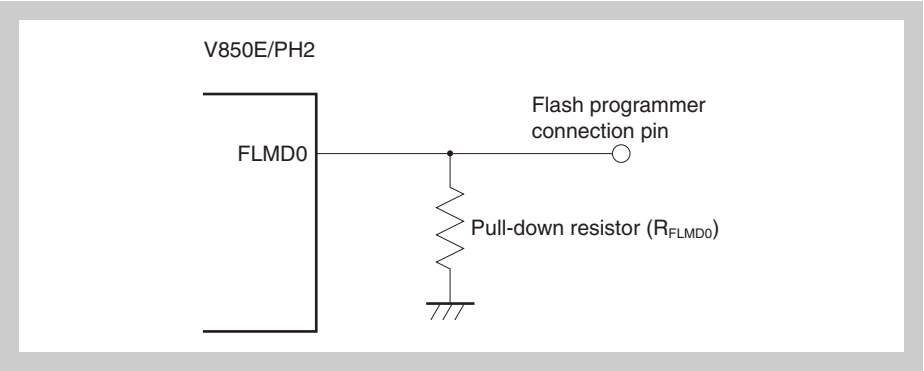


Figure 24-11 FLMD0 pin connection example

(2) FLMD1 pin

When 0 V is input to the FLMD0 pin, the FLMD1 pin does not function. When V_{DD3} is supplied to the FLMD0 pin, the flash memory programming mode is entered, so 0 V must be input to the FLMD1 pin. The following shows an example of the connection of the FLMD1 pin.

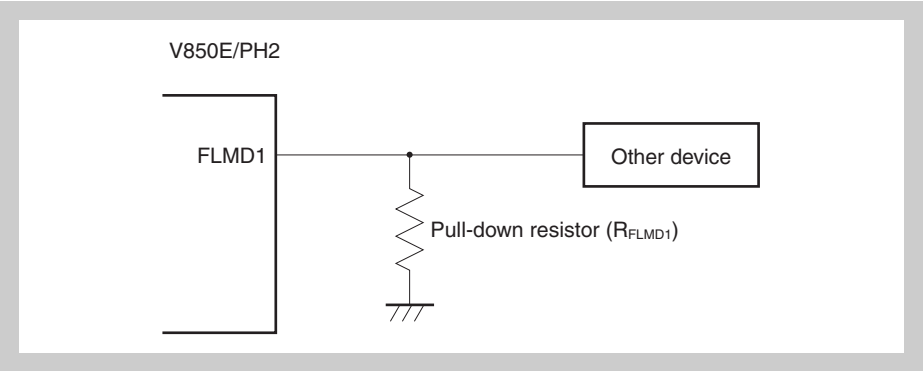


Figure 24-12 FLMD1 pin connection example

Caution If the V_{DD3} signal is input to the FLMD1 pin from another device during on-board writing and immediately after reset, isolate this signal.

Table 24-6 Relationship Between FLMD0 and FLMD1 Pins and Operation Mode when Reset is Released

FLMD0	FLMD1	Operation Mode
0	Don't care	Normal operation mode
V_{DD3}	0	Flash memory programming mode
V_{DD3}	V_{DD3}	Setting prohibited

(3) Serial interface pin

Table 24-7 shows the pins used by each serial interface.

Table 24-7 Pins Used by Serial Interfaces

Serial Interface	Used Pins
UARTC0	TXDC0, RXDC0
CSIB0	SOB0, SIB0, $\overline{\text{SCKB0}}$
CSIB0 + HS	SOB0, SIB0, $\overline{\text{SCKB0}}$, PCM0

When connecting a flash programmer to a serial interface pin that is connected to another device on-board, care should be taken to avoid conflict of signals and malfunction of the other device.

(a) Conflict of signals

When the dedicated flash programmer (output) is connected to a serial interface pin (input) that is connected to another device (output), a conflict of signals occurs. To avoid the conflict of signals, isolate the connection to the other device or set the other device to the output high-impedance status.

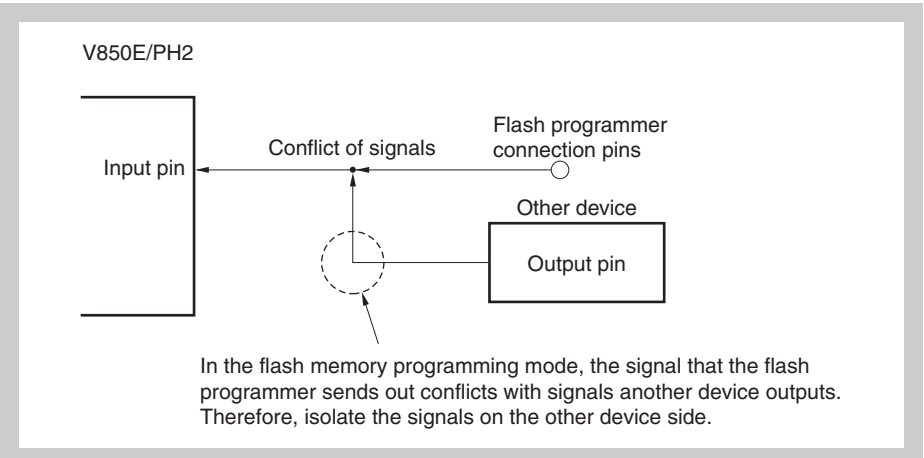


Figure 24-13 Conflict of signals (serial interface input pin)

(b) Malfunction of other device

When the flash programmer (output or input) is connected to a serial interface pin (input or output) that is connected to another device (input), the signal is output to the other device, causing the device to malfunction. To avoid this, isolate the connection to the other device.

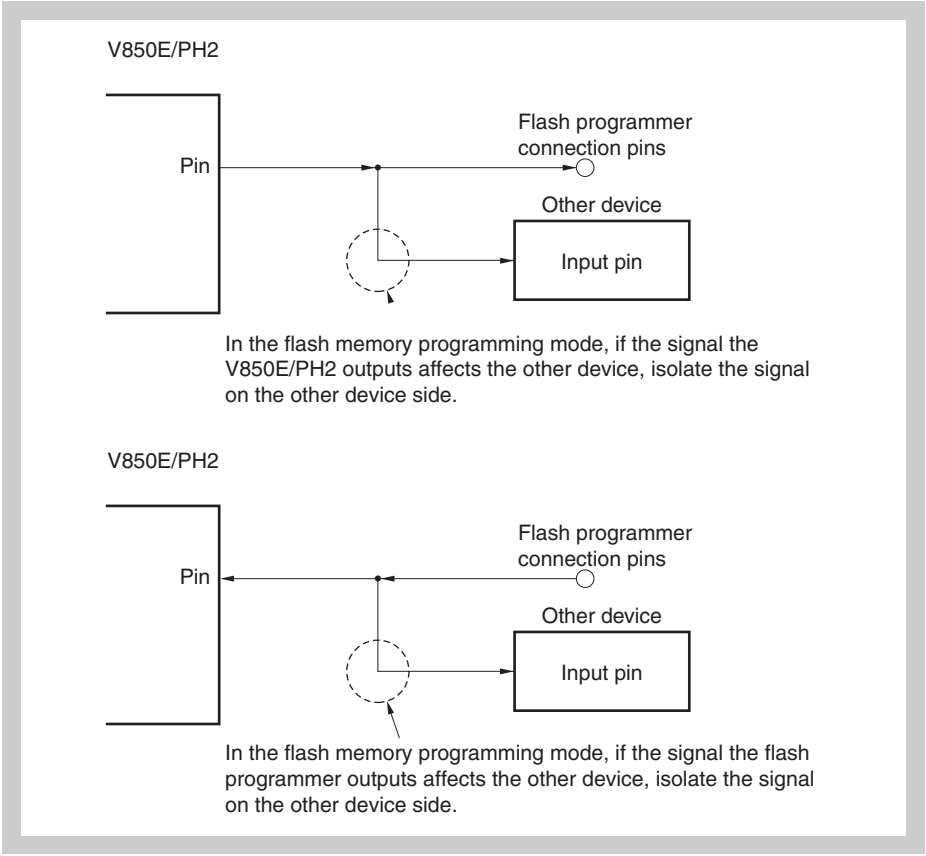


Figure 24-14 Malfunction of other device

(4) $\overline{\text{RESET}}$ pin

When the reset signals of the dedicated flash programmer are connected to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generator on-board, a conflict of signals occurs. To avoid the conflict of signals, isolate the connection to the reset signal generator.

When a reset signal is input from the user system in the flash memory programming mode, the programming operation will not be performed correctly. Therefore, do not input signals other than the reset signals from the dedicated flash programmer.

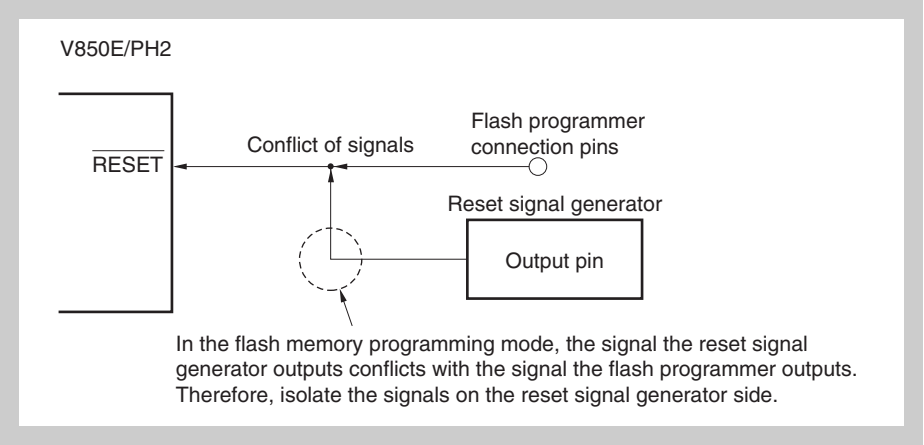


Figure 24-15 Conflict of signals ($\overline{\text{RESET}}$ pin)

(5) Port pins (including NMI)

When the system shifts to the flash memory programming mode, all the pins that are not used for flash memory programming are in the same status as that immediately after reset. If the external device connected to each port does not recognize the status of the port immediately after reset, pins require appropriate processing, such as connecting to V_{DD3} via a resistor or connecting to V_{SS3} via a resistor.

(6) Other signal pins

Connect X1 and X2 in the same status as that in the normal operation mode.

During flash memory programming, input a low level to the $\overline{\text{DRST}}$ pin or leave it open. Do not input a high level.

(7) Power supply

Supply the same power (V_{DD1}, V_{SS1}, V_{DD3}, V_{SS3}, CV_{DD}, CV_{SS}, AV_{DD}, AV_{SS}, AV_{REF0}, AV_{REF1}) as in normal operation mode.

24.5 Rewriting by Self Programming

24.5.1 Overview

The V850E/PH2 supports a flash macro service that allows the user program to rewrite the internal flash memory by itself. By using this interface and a self-programming library that is used to rewrite the flash memory with a user application program, the flash memory can be rewritten by a user application transferred in advance to the internal RAM or external memory. Consequently, the user program can be upgraded and constant data can be rewritten in the field.

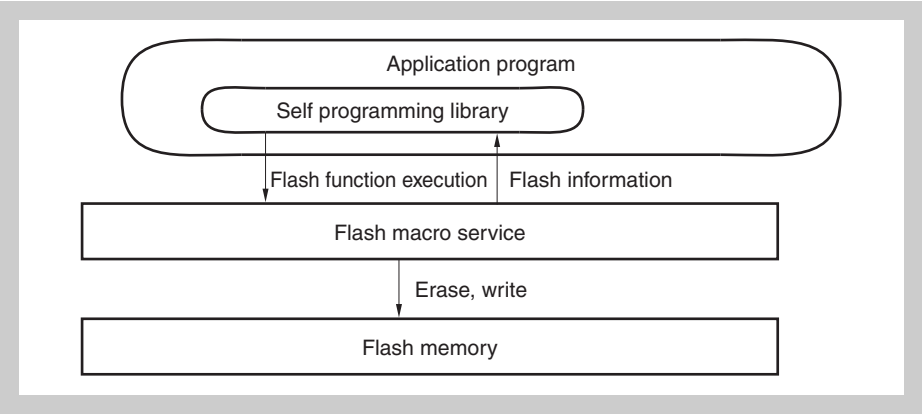


Figure 24-16 Concept of self programming

For further details refer to the application note “Self-Programming Library for embedded Single Voltage FLASH” (U16929EE), obtainable on the Renesas internet page:

http://www2.renesas.eu/products/micro/download/

24.5.2 Features

(1) Secure self-programming (boot swap function)

The V850E/PH2 supports a boot swap function that can exchange the physical memory of blocks 0 to 15 with the physical memory of blocks 16 to 31. By writing the start program to be rewritten to blocks 16 to 31 in advance and then swapping the physical memory, the entire area can be safely rewritten even if a power failure occurs during rewriting because a correct user program always exists in blocks 0 to 15. Since a program flow into and out of the 2nd boot block cluster (physical addresses: 10000_H to $1FFFF_H$) is prohibited, the boot block swap has to be done twice. With this second swap the logical address will be relocated to the physical address. For further information refer to the application note “Self-Programming Library for embedded Single Voltage FLASH” (U16929EE).

Caution Program flow into and out of the 2nd boot block cluster (physical addresses 10000_H to $1FFFF_H$) is prohibited. Thus, when boot swap function is used, the boot block cluster has to be swapped twice.
The user application has to ensure that the second swap will be performed even if the process was interrupted after the first swap. Therefore a dedicated mechanism has to be provided by the user application to check the boot swap status, e.g. by a signature.

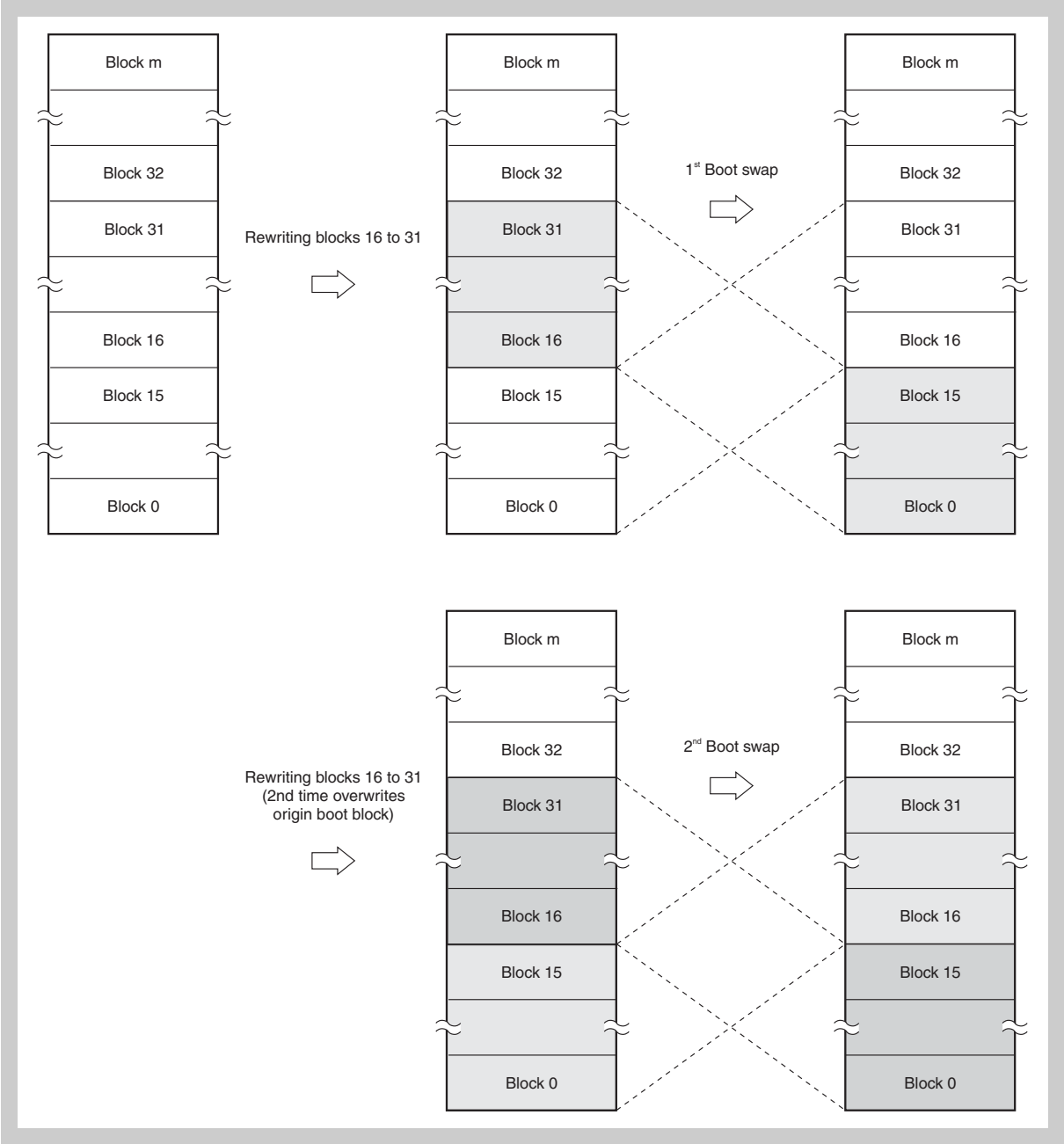


Figure 24-17 Rewriting entire memory area (boot swap)

Remark m = 127 for µPD70F3187
m = 95 for µPD70F3447

(2) Interrupt support

Instructions cannot be fetched from the flash memory during self programming. Conventionally, therefore, a user handler written to the flash memory could not be used even if an interrupt occurred. With the V850E/PH2, a user handler can be registered to an entry RAM area by using a library function, so that interrupt servicing can be performed by internal RAM or external memory execution.

Chapter 25 Electrical Specifications

25.1 Absolute Maximum Ratings

Caution

1. Do not directly connect output (or I/O) pins of IC products to each other, or to V_{DD}, V_{SS}, and GND.
2. Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded. The ratings and conditions shown below for DC characteristics and AC characteristics are within the range for normal operation and quality assurance.

Table 25-1 Absolute maximum ratings (1/2)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD1}	V_{DD1} applied to V_{DD10} to V_{DD15} pins. V_{SS1} = 0 V	-0.5 to +2.0	V
	V_{DD3}	V_{DD3} applied to V_{DD30} to V_{DD37} pins. V_{SS1} = 0 V	-0.5 to +4.6	V
	CV_{DD}	V_{SS1} = 0 V	-0.5 to +2.0	V
	AV_{DD}	V_{DD3} - 0.5 V < AV_{DD} < V_{DD3} + 0.5 V V_{SS1} = 0 V	-0.5 to +4.6	V
	V_{SS1}	V_{SS3} = 0 V	-0.5 to +0.5	V
	V_{SS3}	V_{SS3} applied to V_{SS30} to V_{SS37} pins. V_{SS1} = 0 V	-0.5 to +0.5	V
	CV_{SS}	V_{SS1} = 0 V	-0.5 to +0.5	V
	AV_{SS}	AV_{SS} applied to AV_{SS0} to AV_{SS1} pins. V_{SS1} = 0 V	-0.5 to +0.5	V
Input voltage	V_{I1}	All pins except X1 pin, ANI00 to ANI19 pins V_{I1} < V_{DD3} + 0.3 V	-0.5 to +4.6	V
	V_{I2}	X1 pin V_{I2} < CV_{DD} + 0.5 V	-0.5 to +2.0	V
Analog input voltage	V_{IAN}	ANI00 to ANI19 pins AV_{DD} = 3.0 V to 3.6 V	-0.3 to AV_{DD} + 0.3	V
Analog reference input voltage	AV_{REF0}, AV_{REF1}		-0.3 to AV_{DD} + 0.3	V
Output current, low	I_{OL}	Per pin	4.0	mA
		Total of all pins	100	mA
Output current, high	I_{OH}	Per pin	-4.0	mA
		Total of all pins	-100	mA

Table 25-1 Absolute maximum ratings (2/2)

Parameter	Symbol	Conditions	Ratings	Unit
Operating temperature	T_A	μPD70F3187	-40 to +85	°C
		μPD70F3187(A1)	-40 to +110	°C
		μPD70F3187(A2) μPD70F3447(A2)	-40 to +125	°C
Storage temperature	T_{stg}		-65 to +150	°C

25.2 General Characteristics

Unless specified otherwise, the following conditions are assumed for all characteristics in this chapter.

$V_{DD3x} = AV_{DD} = 3.0$ V to 3.6 V
$V_{DD1x} = CV_{DD} = 1.35$ V to 1.65 V
$V_{SS1x} = CV_{SS} = V_{SS3x} = AV_{SSx} = 0$ V
µPD70F3187: T_A = -40 °C to +85 °C
µPD70F3187(A1): T_A = -40 °C to +110 °C
µPD70F3187(A2), µPD70F3447(A2): T_A = -40 °C to +125 °C

25.2.1 Capacitance

Table 25-2 Capacitance [a]

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input capacitance	C_I	f_C = 1 MHz Un-measured pins returned to 0 V.			15	pF
Output capacitance	C_O				15	pF
I/O capacitance	C_{IO}				15	pF

a) T_A = 25°C
$V_{DD1x} = CV_{DD} = V_{DD3x} = AV_{DD} = V_{SS1x} = CV_{SS} = V_{SS3x} = AV_{SSx} = 0$ V

25.2.2 Operating conditions

Table 25-3 Operating conditions

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Internal system clock frequency	f_{XX}	f_{OSC} = 16 MHz		64		MHz

25.2.3 Oscillator characteristics

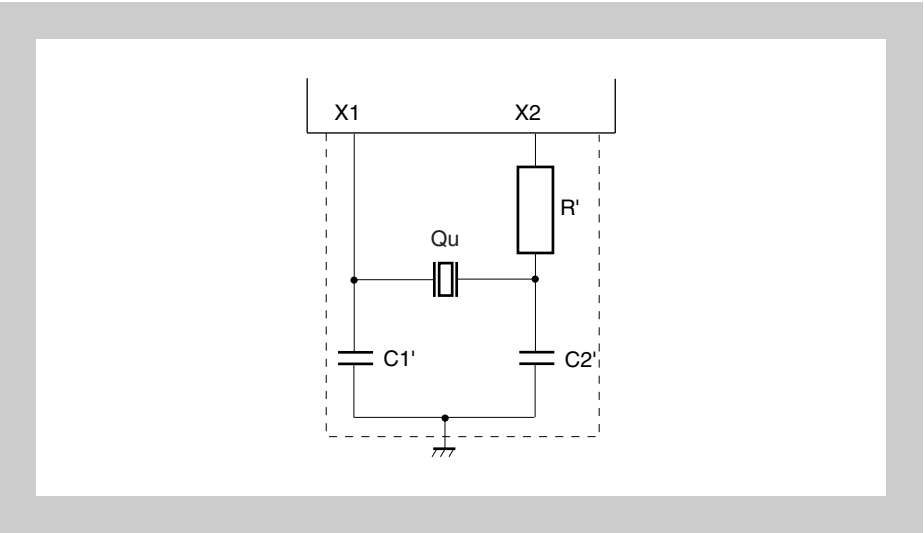


Figure 25-1 Oscillator recommendations

Note Values of capacitors C1', C2' and R' depend on used crystal or resonator and must be specified in cooperation with the manufacturer.

Caution

1. External clock input is prohibited.
2. Wire as follows in the area enclosed by the broken lines in the above figure to avoid an adverse effect from wiring capacitance.
   - Keep the wiring length as short as possible.
   - Do not cross the wiring with the other signal lines.
   - Do not route the wiring near a signal line through which a high fluctuating current flows.
   - Always make the ground point of the oscillator capacitor the same potential as CV_{SS}.
   - Do not ground the capacitor to a ground pattern through which a high current flows.
   - Do not fetch signals from the oscillator.

Table 25-4 Oscillator characteristics

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Oscillation frequency	f_{OSC}			16		MHz
Oscillation stabilization time	t_{OST}	f_{OSC} = 16 MHz			4.096	ms

25.3 DC Characteristics

Unless specified otherwise, the following conditions are assumed for all characteristics in this chapter.

$V_{DD3x} = AV_{DD}$ = 3.0 V to 3.6 V
$V_{DD1x} = CV_{DD}$ = 1.35 V to 1.65 V
$V_{SS1x} = CV_{SS} = V_{SS3x} = AV_{SSx}$ = 0 V
µPD70F3187: T_A = -40 °C to +85 °C
µPD70F3187(A1): T_A = -40 °C to +110 °C
µPD70F3187(A2), µPD70F3447(A2): T_A = -40 °C to +125 °C

Table 25-5 DC characteristics

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	PAL0 to PAL15, PAH0 to PAH5, PDL0 to PDL15, PDH0 to PDH15, PCS0, PCS1, PCS3, PCS4, PCD2 to PCD5, PCT4, PCT5, PCM0, PCM1, PCM6, PCM7, DCK, DMS, DDI, DDO	0.7 V_{DD3}		V_{DD3} + 0.3	V
	V_{IH3}	P00 to P04, P10 to P17, P20 to P27, P30 to P37, P40 to P45, P50 to P57, P60 to P67, P70 to P75, P80 to P86, P90 to P96, P100 to P102, $\overline{RESET}$, MODE0 to MODE2, $\overline{DRST}$				
Input voltage, low	V_{IL1}	PAL0 to PAL15, PAH0 to PAH5, PDL0 to PDL15, PDH0 to PDH15, PCS0, PCS1, PCS3, PCS4, PCD2 to PCD5, PCT4, PCT5, PCM0, PCM1, PCM6, PCM7, DCK, DMS, DDI, DDO	-0.5		0.3 V_{DD3}	V
	V_{IL3}	P00 to P04, P10 to P17, P20 to P27, P30 to P37, P40 to P45, P50 to P57, P60 to P67, P70 to P75, P80 to P86, P90 to P96, P100 to P102, $\overline{RESET}$, MODE0 to MODE2, $\overline{DRST}$				
Output voltage, high	V_{OH1}	I_{OH} = -2.5 mA	V_{DD3} - 1.0			V
		I_{OH} = -100 µA	V_{DD3} - 0.4			V
Output voltage, low	V_{OL1}	I_{OL} = 2.5 mA			0.8	V
		I_{OL} = 100 µA			0.4	V
Input leakage current, high	I_{LIH}	Pins other than analog input pins ANI00 to ANI19, $V_{IH} = V_{DD3}$			10	µA
		Analog input pins ANI00 to ANI19, $V_{IH} = AV_{DD}$			3	µA
Input leakage current, low	I_{LIL}	Pins other than analog input pins ANI00 to ANI19, V_{IL} = 0 V			-10	µA
		Analog input pins ANI00 to ANI19, V_{IL} = 0 V			-3	µA
Power supply current	I_{DD1}	$V_{DD1} + CV_{DD}$			200	mA
	I_{DD3}	V_{DD3} [a]			50	mA

a) No external loads considered (C_L = 0pF). External loads cause additional pin currents. Pin current for each pin can be calculated according to following formula:
I_{Pin} [µA] = 3.63 × C_L [pF] × F [MHz]
where CL is external load capacitance and F is the average pin toggle frequency.
Load dependent pin currents must be summed up and added to I_{DD3}.
Do not exceed the following supply current in total:
$I_{DD3} + \Sigma I_{Pin} \leq 100mA$

25.4 AC Characteristics

Unless specified otherwise, the following conditions are assumed for all characteristics in this chapter.

V_{DD3x} = AV_{DD} = 3.0 V to 3.6 V
V_{DD1x} = CV_{DD} = 1.35 V to 1.65 V
V_{SS1x} = CV_{SS} = V_{SS3x} = AV_{SSx} = 0 V
µPD70F3187: T_A = -40 °C to +85 °C
µPD70F3187(A1): T_A = -40 °C to +110 °C
µPD70F3187(A2), µPD70F3447(A2): T_A = -40 °C to +125 °C

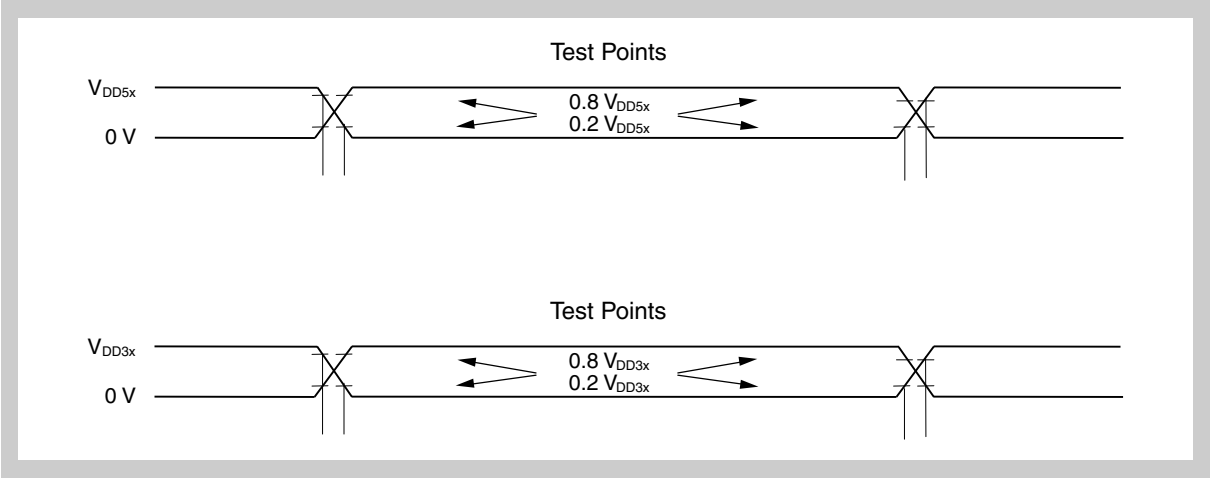


Figure 25-2 AC test input/output waveform

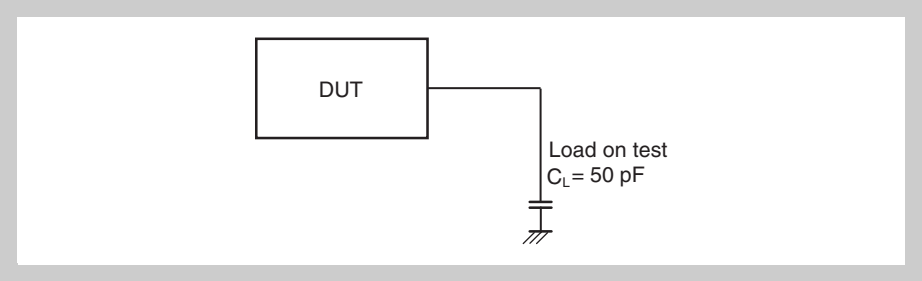


Figure 25-3 AC test load condition

25.4.1 External asynchronous memory access read timing

Table 25-6 External asynchronous memory access read timing

Parameter		Symbol	MIN. [a]	MAX. [a]	Unit
Data input set up time (vs. address)	<10>	t_{SAID}		(2 + w_{AS} + w_D + w) T [ns] - 30	ns
Data input set up time (vs. $\overline{RD}\downarrow$)	<11>	t_{SRDID}		(1.5 + w_D + w) T [ns] - 30	ns
$\overline{RD}$ Low level width	<12>	t_{WRDL}	(1.5 + w_D + w) T [ns] - 15		ns
$\overline{RD}$ High level width	<13>	t_{WRDH}	(0.5 + w_{AS} + i) T [ns] - 15		ns
Address, $\overline{CSn} \rightarrow \overline{RD}\downarrow$ delay time	<14>	t_{DARD}	(0.5 + w_{AS}) T [ns] - 20		ns
$\overline{RD}\uparrow \rightarrow$ address delay time	<15>	t_{DRDA}	i × T [ns] - 2		ns
Data input hold time (vs. $\overline{RD}\uparrow$)	<16>	t_{HRDID}	0		ns
$\overline{RD}\uparrow \rightarrow$ data output delay time	<17>	t_{DRDOD}	(1 + i) T [ns] - 15		ns
$\overline{WAIT}$ set up time (vs. address)	<31>	t_{SAW}		(1 + w_{AS}) T [ns] - 30	ns
$\overline{WAIT}$ high level width	<32>	t_{WWH}	T [ns] - 2		ns

a) T: $2 / f_{XX}$
i: Number of idle states specified by BCC register
w_{AS}: Number of waits specified by ASC register
w_D: Number of waits specified by DWC1, DWC2 register; $w_D \geq 1$
w: Number of waits due to external wait signal ($\overline{WAIT}$)

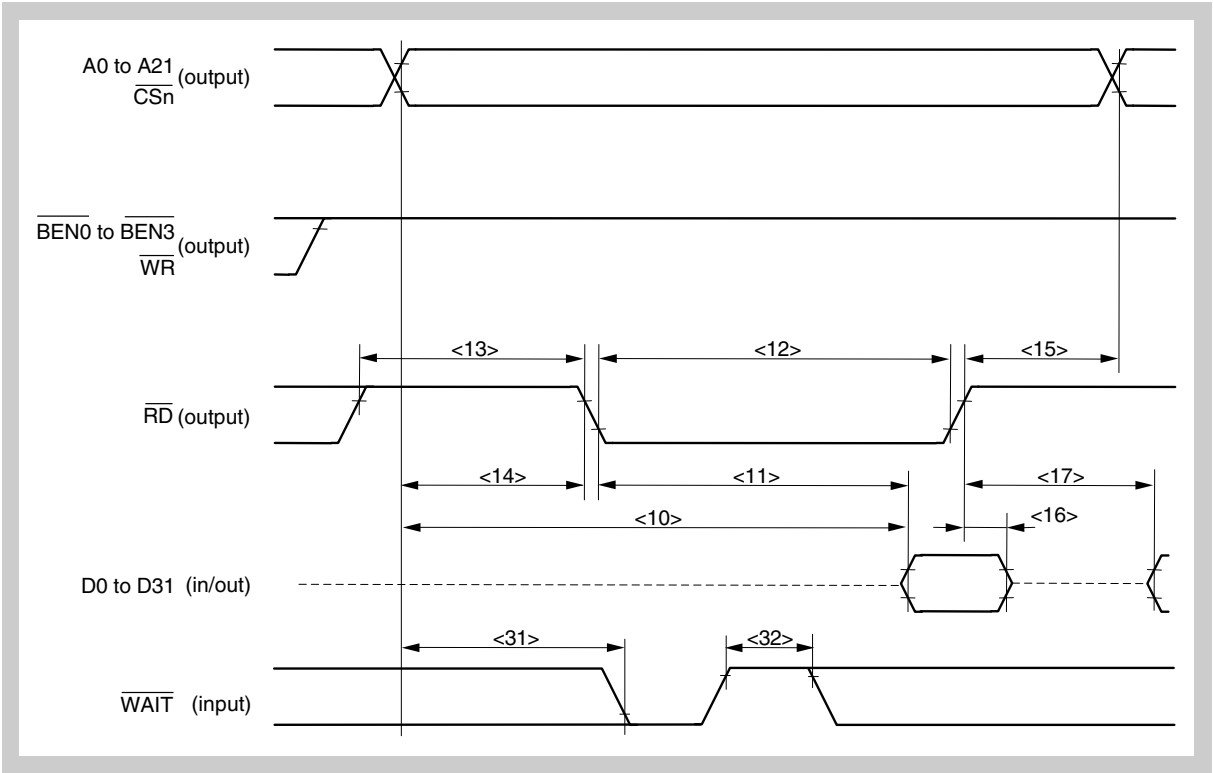


Figure 25-4 External asynchronous memory access read timing

25.4.2 External asynchronous memory access write timing

Table 25-7 External asynchronous memory access write timing

Parameter		Symbol	MIN. [a]	MAX. [a]	Unit
Address, $\overline{CSn} \rightarrow \overline{WR} \downarrow$ delay time	<20>	T_{DAWR}	$(1 + w_{AS})$ T [ns] - 20		ns
Address set up (vs. $\overline{WR}\uparrow$)	<21>	T_{SAWR}	$(1.5 + w_{AS} + w_D + w)$ T [ns] - 10		ns
$\overline{WR}\uparrow \rightarrow$ address delay time	<22>	T_{DWRA}	(0.5 + i) T [ns] - 5		ns
$\overline{WR}$ High level width	<23>	T_{WWRH}	$(1.5 + i + w_{AS})$ T [ns] - 15		ns
$\overline{WR}$ Low level width	<24>	T_{WWRL}	$(0.5 + w + w_D)$ T [ns] - 12		ns
Data output set up time (vs. $\overline{WR}\uparrow$)	<25>	T_{SODWR}	$(0.5 + w_{AS} + w_D + w)$ T [ns] - 15		ns
Data output hold time (vs. $\overline{WR}\uparrow$)	<26>	T_{HWROD}	(0.5 + i) T [ns] - 15		ns
$\overline{WAIT}$ set up time (vs. address)	<31>	T_{SAW}		$(1 + w_{AS})$ T [ns] - 30	ns
$\overline{WAIT}$ high level width	<32>	T_{WWH}	T [ns] - 2		ns

a) T: $2 / f_{XX}$
i: Number of idle states specified by BCC register
w_{AS}: Number of waits specified by ASC register
w_D: Number of waits specified by DWC1, DWC2 register; $w_D \geq 1$
w: Number of waits due to external wait signal ($\overline{WAIT}$)

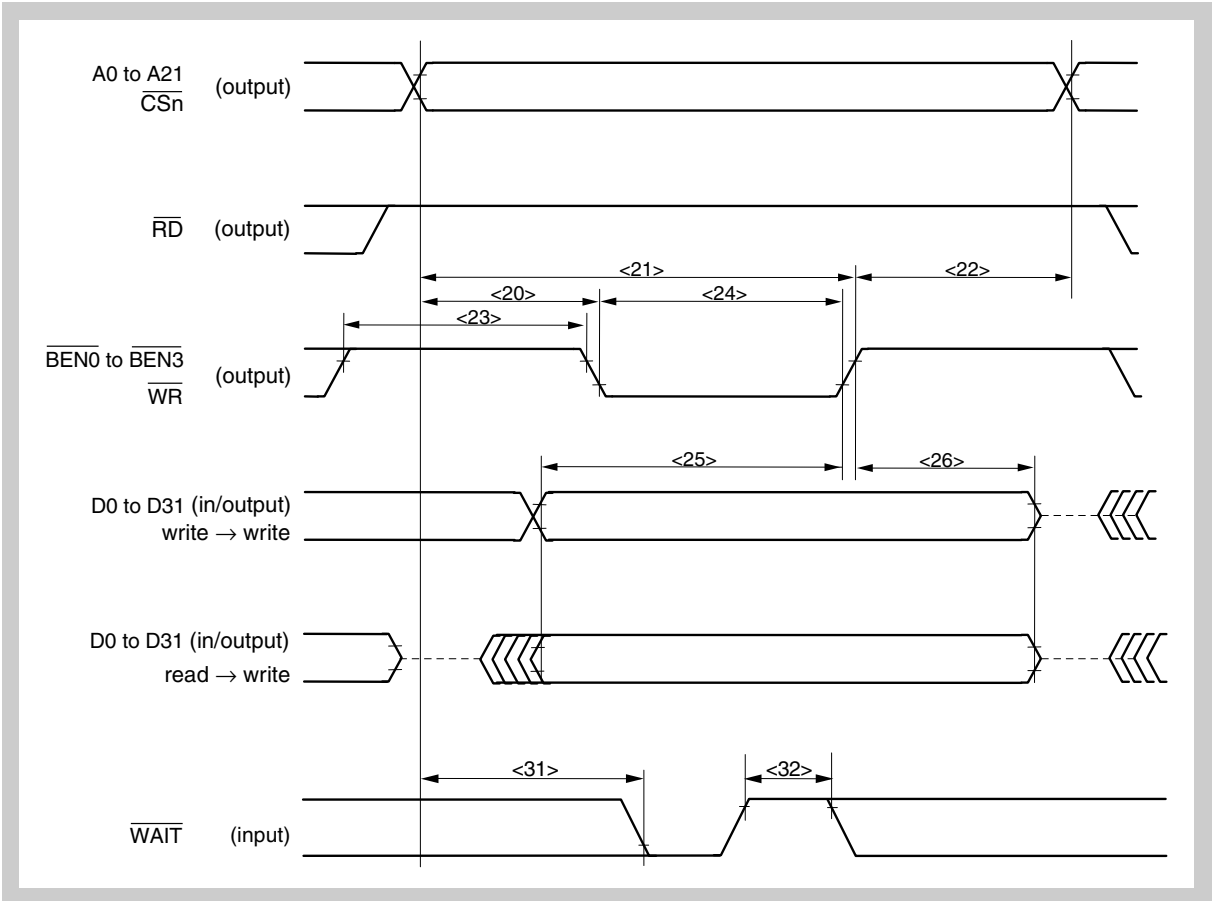


Figure 25-5 External asynchronous memory access write timing

25.4.3 Reset Timing (Power Up/Down Sequence)

Table 25-8 Reset timing

Parameter	Symbol	MIN.	MAX.	Unit
$\overline{\text{RESET}}$ high-level width	t_{WRSH}	500		ns
$\overline{\text{RESET}}$ low-level width	t_{WRSL}	500		ns
$V_{DD3x} \leftrightarrow V_{DD1x}$ power up delay	t_{DVR}	0		ns
$V_{DD3x} \leftrightarrow V_{DD1x}$ power down delay	t_{DVF}	0		ns
$\overline{\text{RESET}}$ hold time	t_{DVRR}	1		µs
$\overline{\text{RESET}}$ setup time	t_{DVRF}	0		ns

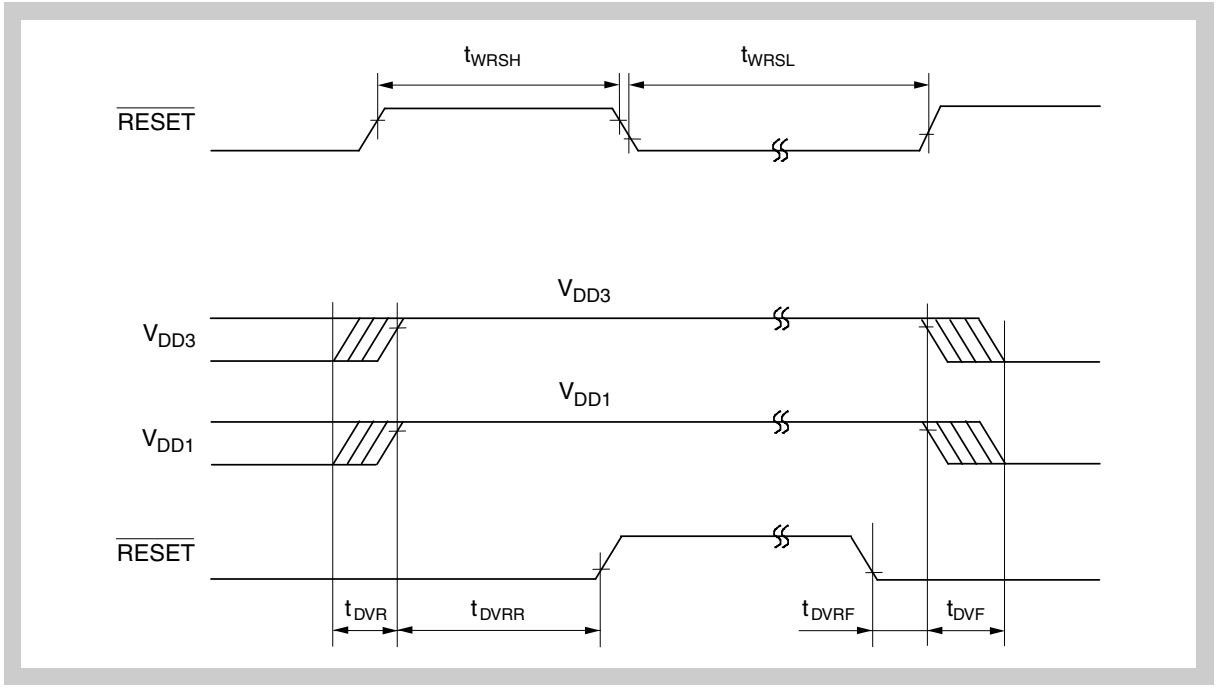


Figure 25-6 Reset Timing

Caution Ensure that a valid $\overline{\text{RESET}}$ signal (low active) is applied to the $\overline{\text{RESET}}$ pin at any time if the voltage power of V_{DD1x} is below its operating condition range.

25.4.4 Interrupt timing

Table 25-9 Interrupt timing

Parameter	Symbol	Condition	MIN.	MAX.	Unit
NMI high-level width	t_{WNIH}	NRC0 bit = 0	96 T [ns] + 10		ns
		NRC0 bit = 1	384 T [ns] + 10		ns
NMI low-level width	t_{WNIL}	NRC0 bit = 0	96 T [ns] + 10		ns
		NRC0 bit = 1	384 T [ns] + 10		ns
INTPx high-level width	t_{WITH}	NRC1 bit = 0	96 T [ns] + 10		ns
		NRC1 bit = 1	384 T [ns] + 10		ns
INTPx low-level width	t_{WITL}	NRC1 bit = 0	96 T [ns] + 10		ns
		NRC1 bit = 1	384 T [ns] + 10		ns
INTP0, INTP1 high-level width	t_{WTIH}		500		ns
INTP0, INTP1 low-level width	t_{WTIL}		500		ns

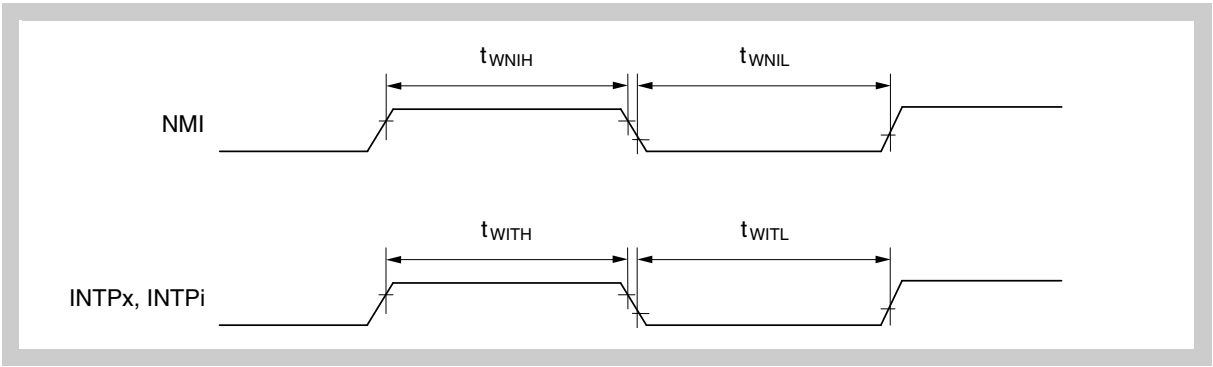


Figure 25-7 Interrupt timing

25.5 Peripheral Characteristics

Unless specified otherwise, the following conditions are assumed for all characteristics in this chapter.

$V_{DD3x} = AV_{DD} = 3.0$ V to 3.6 V
$V_{DD1x} = CV_{DD} = 1.35$ V to 1.65 V
$V_{SS1x} = CV_{SS} = V_{SS3x} = AV_{SSx} = 0$ V
µPD70F3187: $T_A = -40$ °C to +85 °C
µPD70F3187(A1): $T_A = -40$ °C to +110 °C
µPD70F3187(A2), µPD70F3447(A2): $T_A = -40$ °C to +125 °C

25.5.1 Timer characteristics

Table 25-10 Timer P characteristics

Parameter	Symbol	Condition	MIN.	MAX.	Unit
TIPmn input high-level width	t_{WTIPH}	NRCx bit[a] = 0	96 T [ns] + 10		ns
		NRCx bit[a] = 1	384 T [ns] + 10		ns
TIPmn input low-level width	t_{WTIPL}	NRCx bit[a] = 0	96 T [ns] + 10		ns
		NRCx bit[a] = 1	384 T [ns] + 10		ns

a) x = 3 to 6 (depending on the pin group the TIPmn belongs to, refer to *Table 20-82, "Noise elimination," on page 939*).

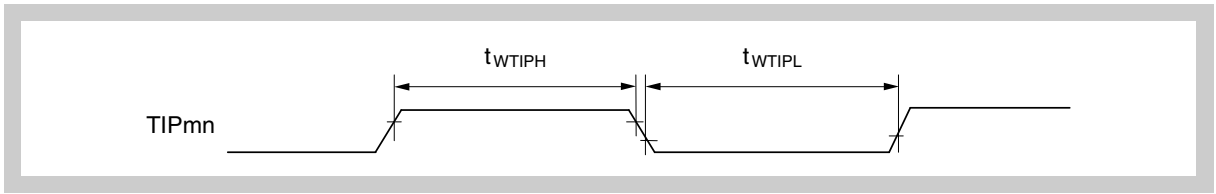


Figure 25-8 Timer P characteristics

Table 25-11 Timer R characteristics

Parameter	Symbol	Condition	MIN.	MAX.	Unit
TIR1n input high-level width	t_{WTIRH}	NRC7 bit = 0	96 T [ns] + 10		ns
		NRC7 bit = 1	384 T [ns] + 10		ns
TIR1n input low-level width	t_{WTIRL}	NRC7 bit = 0	96 T [ns] + 10		ns
		NRC7 bit = 1	384 T [ns] + 10		ns
TORnx to TORny output delay	$t_{DTORTOR}$			15	ns

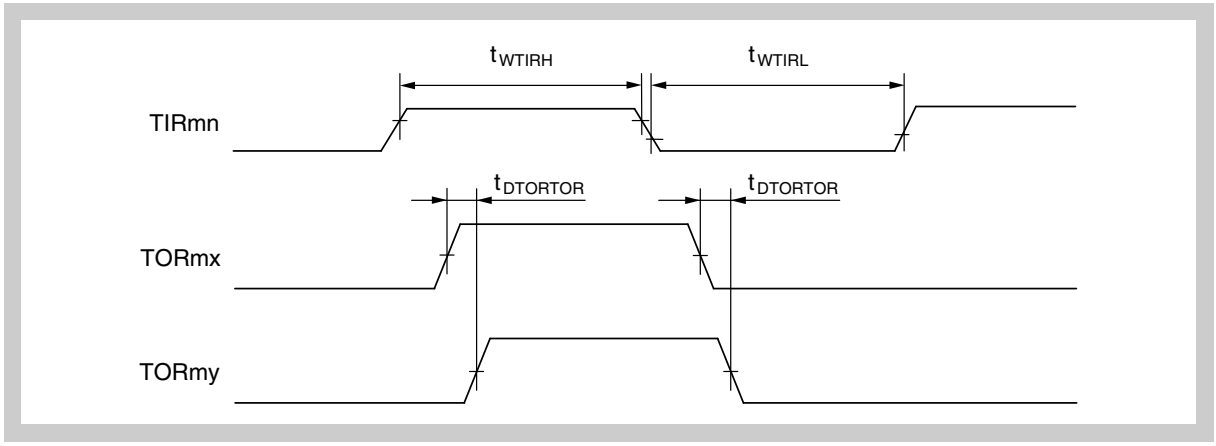


Figure 25-9 Timer R characteristics

Note x = 0 to 7, y = 0 to 7, x ≠ y

Table 25-12 Timer T characteristics

Parameter	Symbol	Condition	MIN.	MAX.	Unit
TITmn input high-level width	t_{WTITH}	NRC2 bit = 0	96 T [ns] + 10		ns
		NRC2 bit = 1	384 T [ns] + 10		ns
TITmn input low-level width	t_{WTITL}	NRC2 bit = 0	96 T [ns] + 10		ns
		NRC2 bit = 1	384 T [ns] + 10		ns

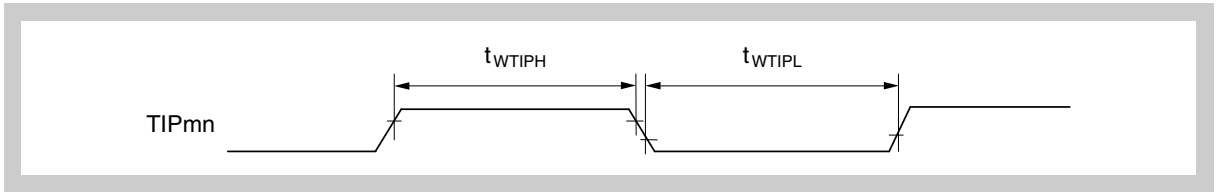


Figure 25-10 Timer T characteristics

25.5.2 Serial interface characteristics

(1) Clocked serial interface B (CSIB) characteristics

Table 25-13 CSIB characteristics (master mode)

Parameter	Symbol	MIN.	MAX.	Unit
$\overline{\text{SCKBn}}$ output clock cycle time	t_{CYSKM}	125		ns
$\overline{\text{SCKBn}}$ output high level width	t_{WSKHM}	0.5 t_{CYSKM} [ns] - 10		ns
$\overline{\text{SCKBn}}$ output low level width	t_{WSKLM}	0.5 t_{CYSKM} [ns] - 10		ns
SIBn input setup time (vs. $\overline{\text{SCKBn}}\uparrow$)	t_{SSISKM}	20		ns
SIBn input hold time (vs. $\overline{\text{SCKBn}}\uparrow$)	t_{HSKSIM}	10		ns
SOBn output delay (vs. $\overline{\text{SCKBn}}\downarrow$)	t_{DSKSOM}		10	ns
SOBn output hold time (vs. $\overline{\text{SCKBn}}\uparrow$)	t_{HSKSOM}	0.5 t_{CYSKM} [ns] - 10		ns

Table 25-14 CSIB characteristics (slave mode)

Parameter	Symbol	MIN.	MAX.	Unit
$\overline{\text{SCKBn}}$ input clock cycle time	t_{CYSKS}	125		ns
$\overline{\text{SCKBn}}$ input high level width	t_{WSKHS}	0.5 t_{CYSKS} [ns] - 10		ns
$\overline{\text{SCKBn}}$ input low level width	t_{WSKLS}	0.5 t_{CYSKS} [ns] - 10		ns
SIBn input setup time (vs. $\overline{\text{SCKBn}}\uparrow$)	t_{SSISKS}	5		ns
SIBn input hold time (vs. $\overline{\text{SCKBn}}\uparrow$)	t_{HSKSIS}	10		ns
SOBn output delay (vs. $\overline{\text{SCKBn}}\downarrow$)	t_{DSKSOS}		25	ns
SOBn output hold time (vs. $\overline{\text{SCKBn}}\uparrow$)	t_{HSKSOS}	0.5 t_{CYSKS} [ns] - 10		ns

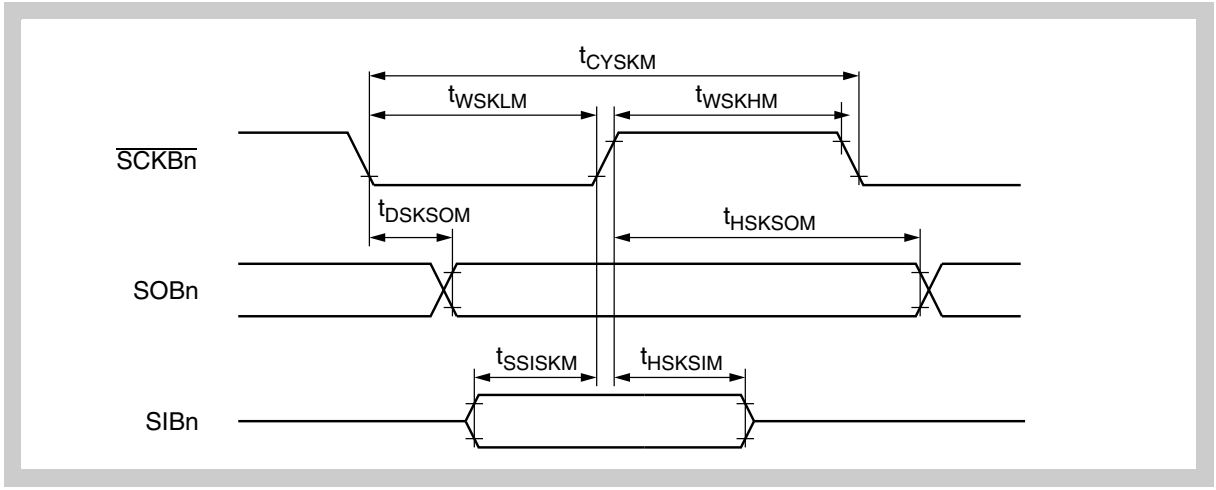


Figure 25-11 CSIB timing in master mode (CKP, DAP bits = 00_B or 11_B)

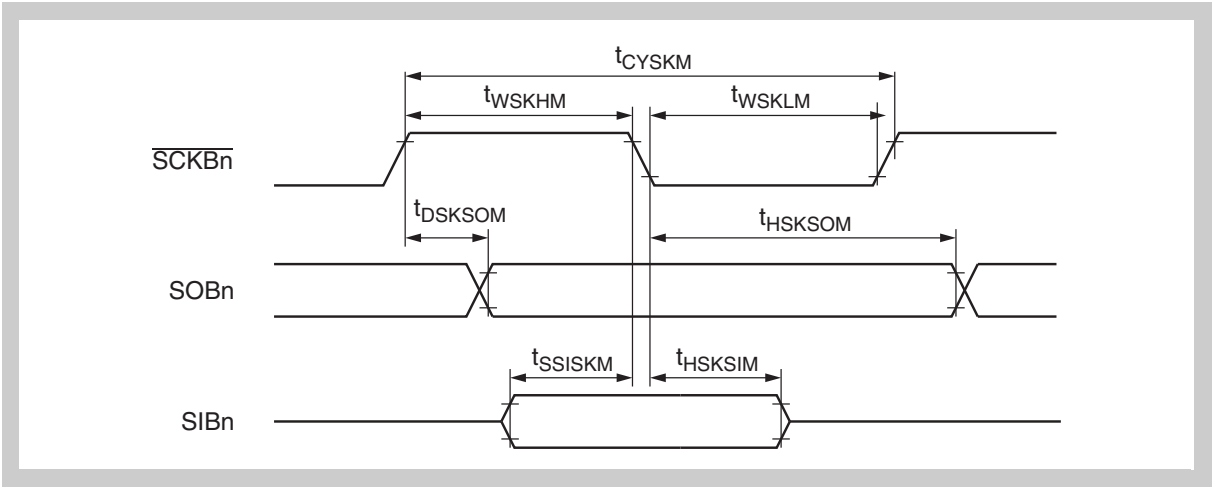


Figure 25-12 CSIB Timing in master mode (CKP, DAP bits = 01_B or 10_B)

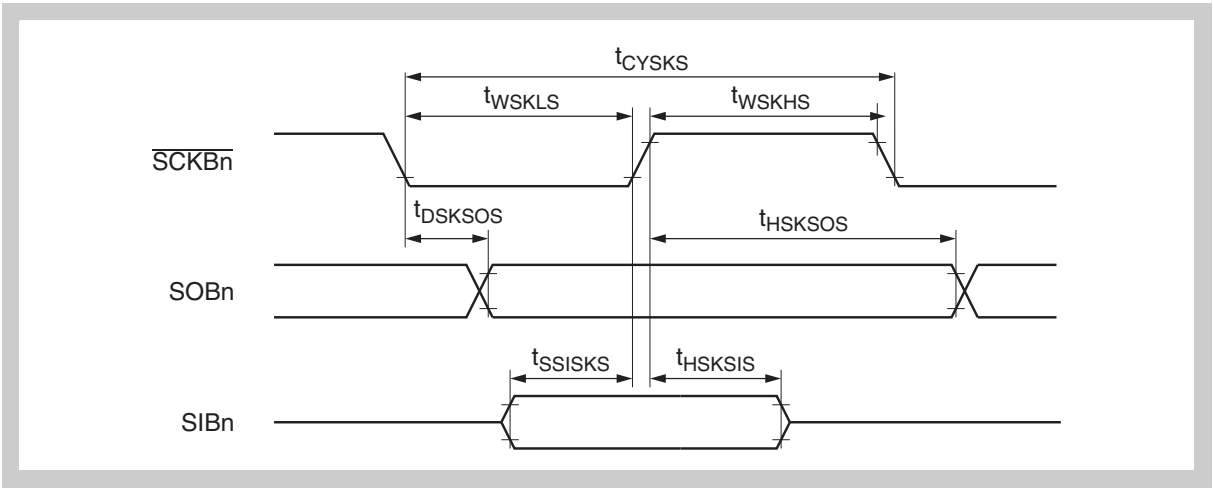


Figure 25-13 CSIB timing in slave mode (CKP, DAP bits = 00_B or 11_B)

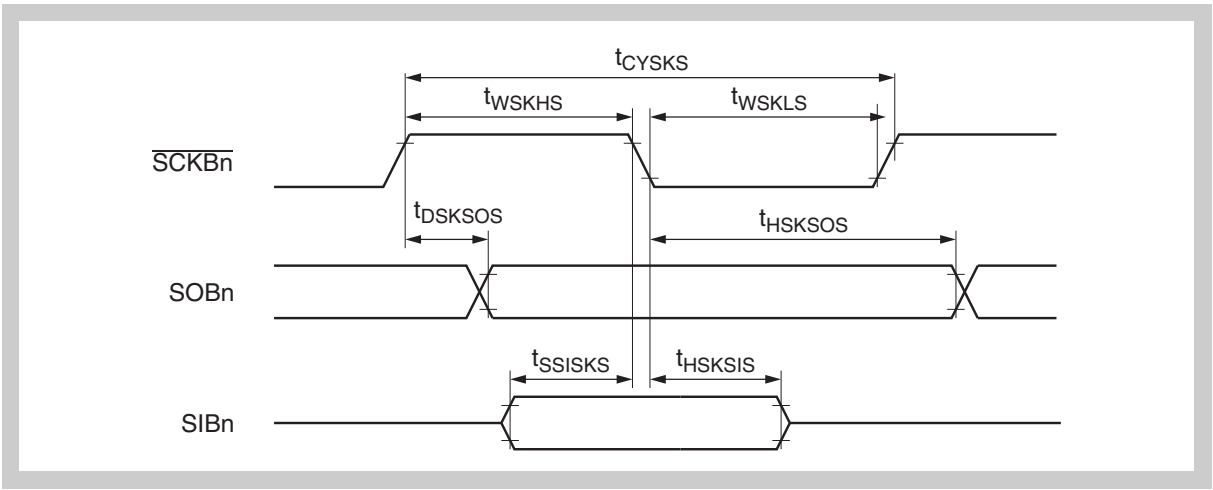


Figure 25-14 CSIB timing in slave mode (CKP, DAP bits = 01_B or 10_B)

(2) Clocked serial interface 3 (CSI3) timing

Table 25-15 CSI3 characteristics (master mode)

Parameter	Symbol	MIN.	MAX.	Unit
CSI3 operation clock cycle time	t_{CYK}	15.625		ns
$\overline{SCK3n}$ clock cycle time	t_{CYSKM}	125		ns
$\overline{SCK3n}$ high level width	t_{WSKHM}	0.5 t_{CYSKM} [ns] - 10		ns
$\overline{SCK3n}$ low level width	t_{WSKLM}	0.5 t_{CYSKM} [ns] - 10		ns
SI3n setup time (vs. $\overline{SCK3n}$↑)	t_{SSISKM}	20		ns
SI3n hold time (vs. $\overline{SCK3n}$↑)	t_{HSKSIM}	10		ns
SO3n output delay (vs. $\overline{SCK3n}$↓)	t_{DSKSOM}		10	ns
SO3n output hold time (vs. $\overline{SCK3n}$↑)	t_{HSKSOM}	0.5 t_{CYSKM} [ns] - 10		ns
SCS3nm inactive width	t_{WSKCSB}	0.5 t_{CYSKM} [ns] - 10		ns
SCS3nm setup time (vs. $\overline{SCK3n}$↓)	$t_{SCSZCK0}$	t_{CYK} [ns] - 10		ns
	$t_{SCSZCK1}$	t_{CYSKM} [ns] + t_{CYK} [ns] - 10		ns
	$t_{SCSZCK2}$	t_{CYSKM} [ns] - t_{CYK} [ns] - 10		ns
SCS3nm hold time (vs. $\overline{SCK3n}$↑)	$t_{HSKCSZ0}$	t_{CYK} [ns] - 10		ns
	$t_{HSKCSZ1}$	0.5 t_{CYSKM} [ns] - 10		ns

Table 25-16 CSI3 characteristics (slave mode)

Parameter	Symbol	MIN.	MAX.	Unit
CSI3 operation clock cycle time	t_{CYK}	15.625		ns
$\overline{SCK3n}$ clock cycle time	t_{CYSKS}	125		ns
$\overline{SCK3n}$ high level width	t_{WSKHS}	0.5 t_{CYSKS} [ns] - 10		ns
$\overline{SCK3n}$ low level width	t_{WSKLS}	0.5 t_{CYSKS} [ns] - 10		ns
SI3n setup time (vs. $\overline{SCK3n}$↑)	t_{SSISKS}	5		ns
SI3n hold time (vs. $\overline{SCK3n}$↑)	t_{HSKSIS}	1.5 t_{CYK} [ns] + 10		ns
SO3n output delay (vs. $\overline{SCK3n}$↓)	t_{DSKSOS}		25	ns
SO3n output hold time (vs. $\overline{SCK3n}$↑)	t_{HSKSOS}	0.5 t_{CYSKS} [ns] - 10		ns

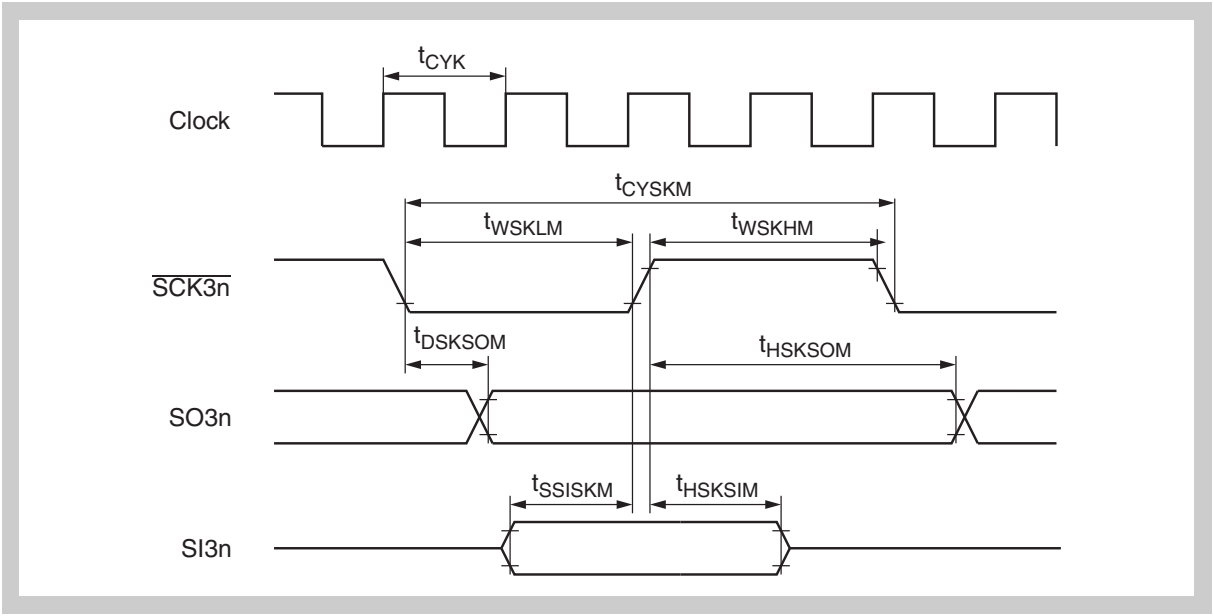


Figure 25-15 CSI3 timing in master mode (CKP, DAP bits = 00_B or 11_B)

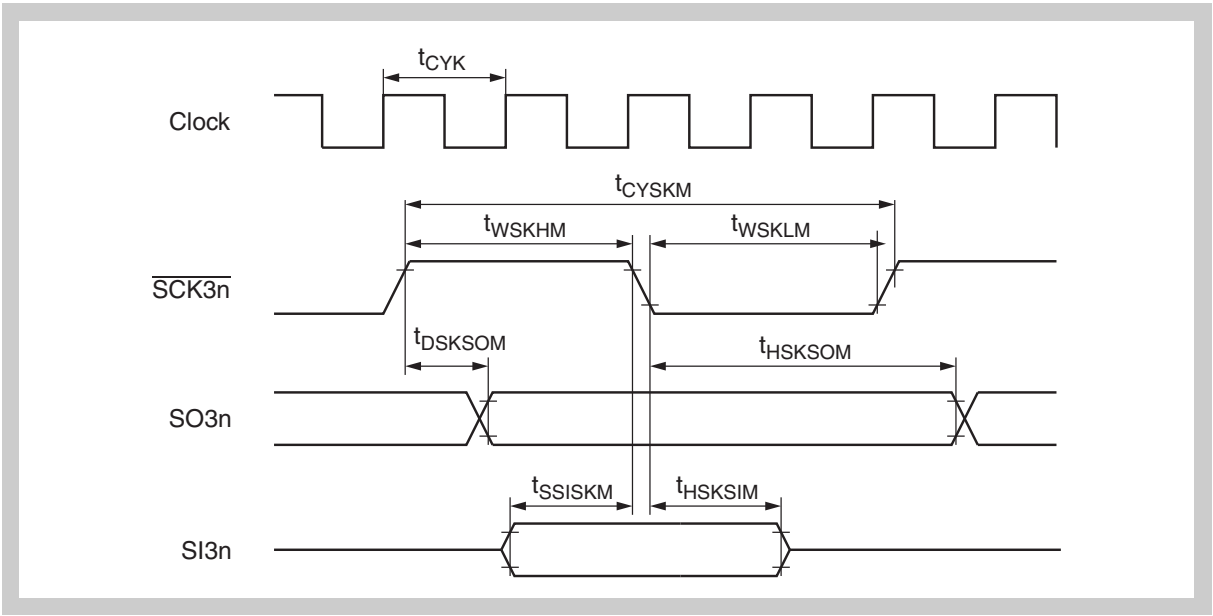


Figure 25-16 CSI3 timing in master mode (CKP, DAP bits = 01_B or 10_B)

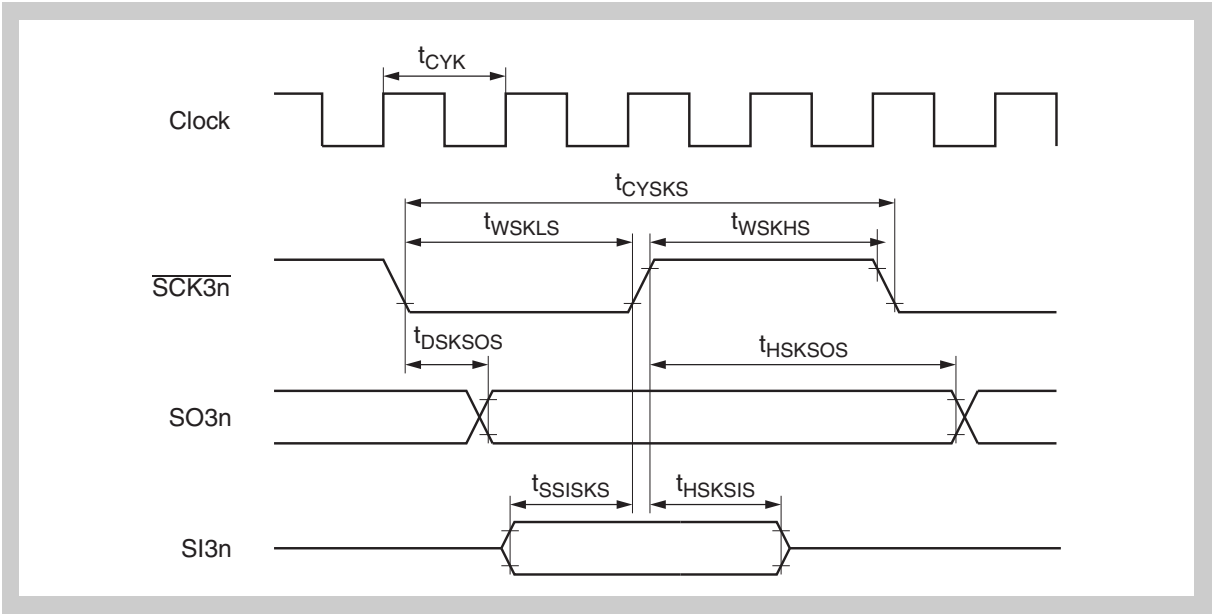


Figure 25-17 CSI3 timing in slave mode (CKP, DAP bits = 00_B or 11_B)

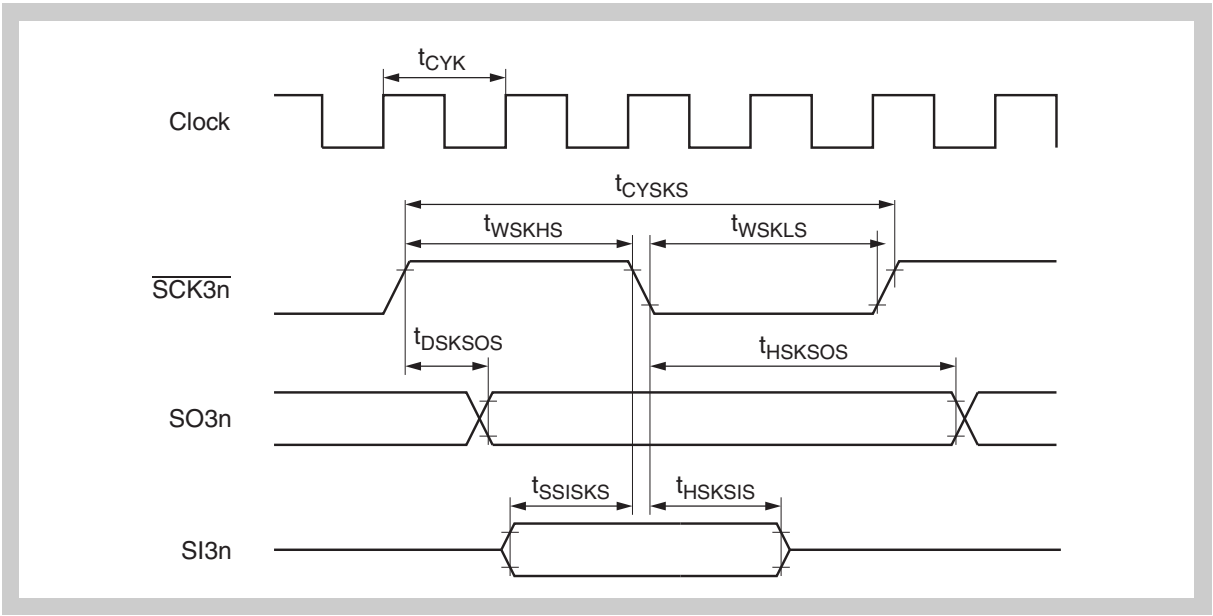


Figure 25-18 CSI3 timing in slave mode (CKP, DAP bits = 01_B or 10_B)

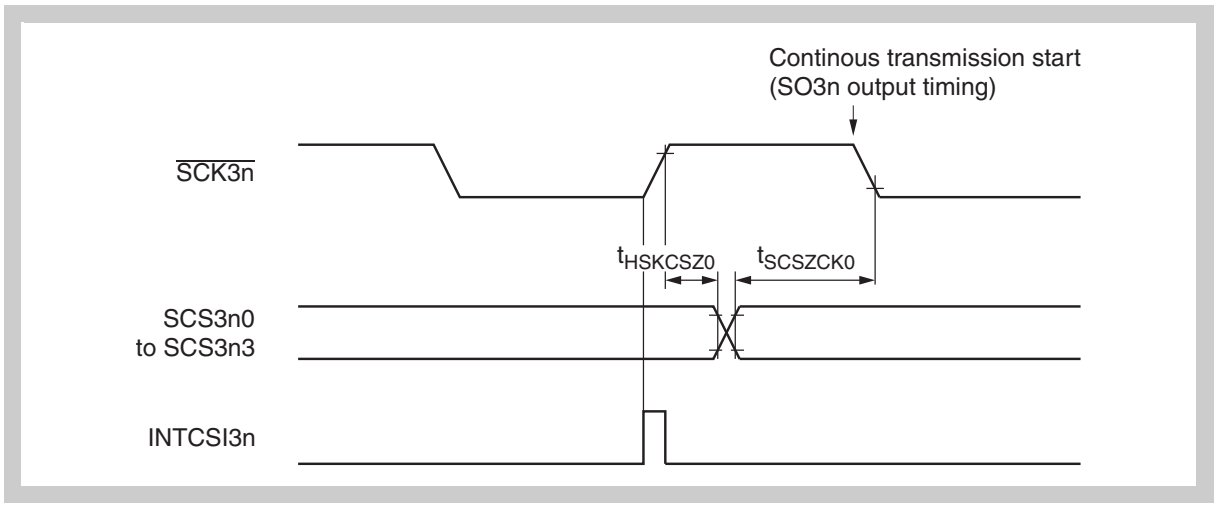


Figure 25-19 CSI3 chip select timing (master mode only) (CSIT = 0, CSWE = 0, CSMD = 0)

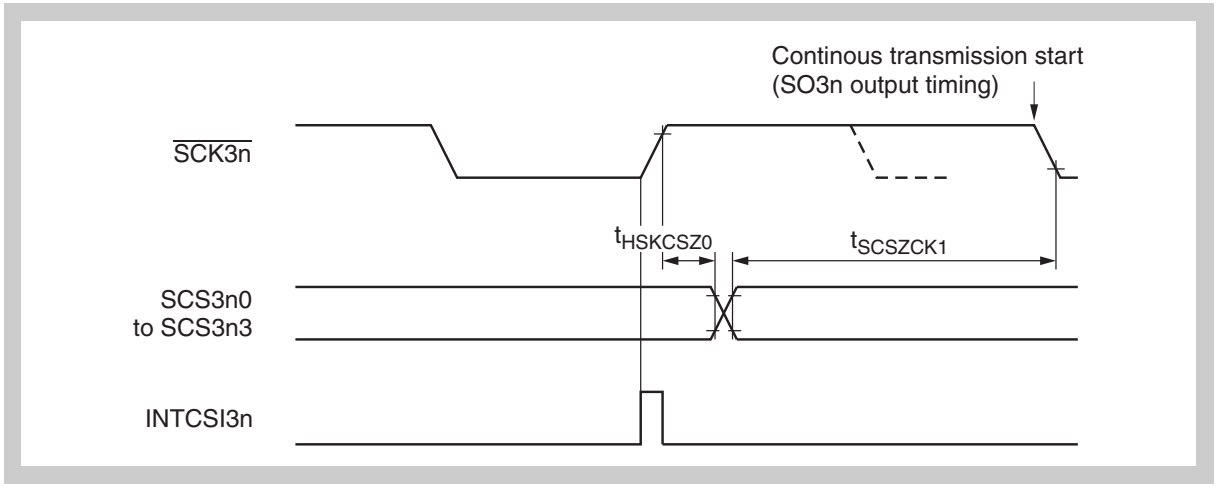


Figure 25-20 CSI3 chip select timing (master mode only) (CSIT = 0, CSWE = 1, CSMD = 0)

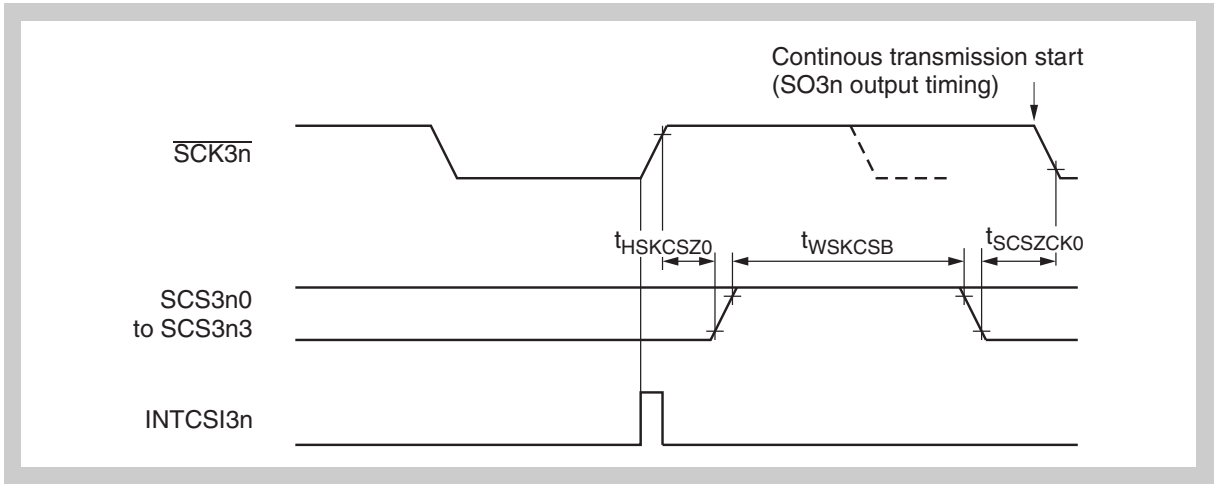


Figure 25-21 CSI3 chip select timing (master mode only) (CSIT = 0, CSWE = 1, CSMD = 1)

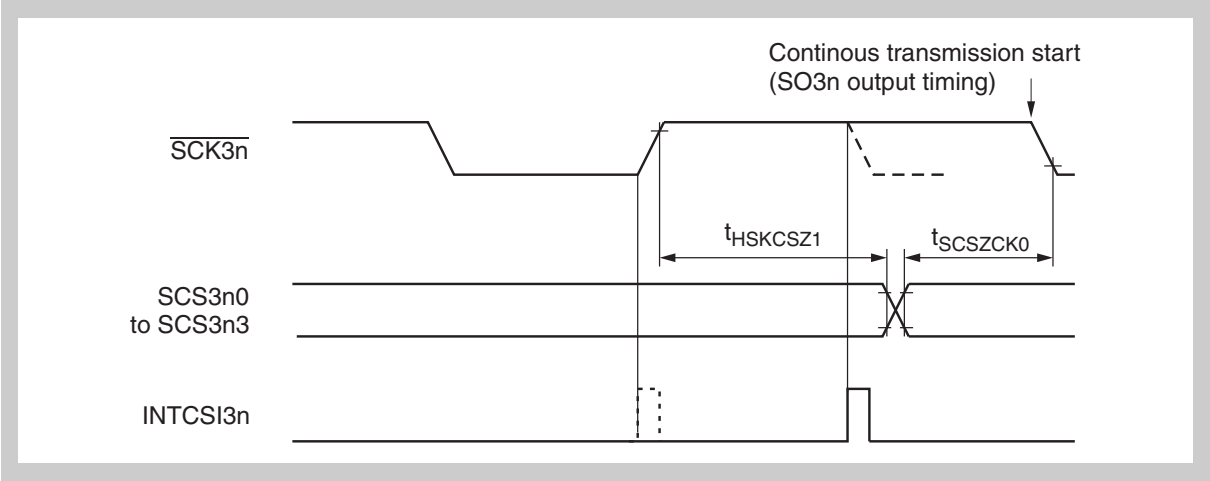


Figure 25-22 CSI3 chip select timing (master mode only) (CSIT = 1, CSWE = 0, CSMD = 0)

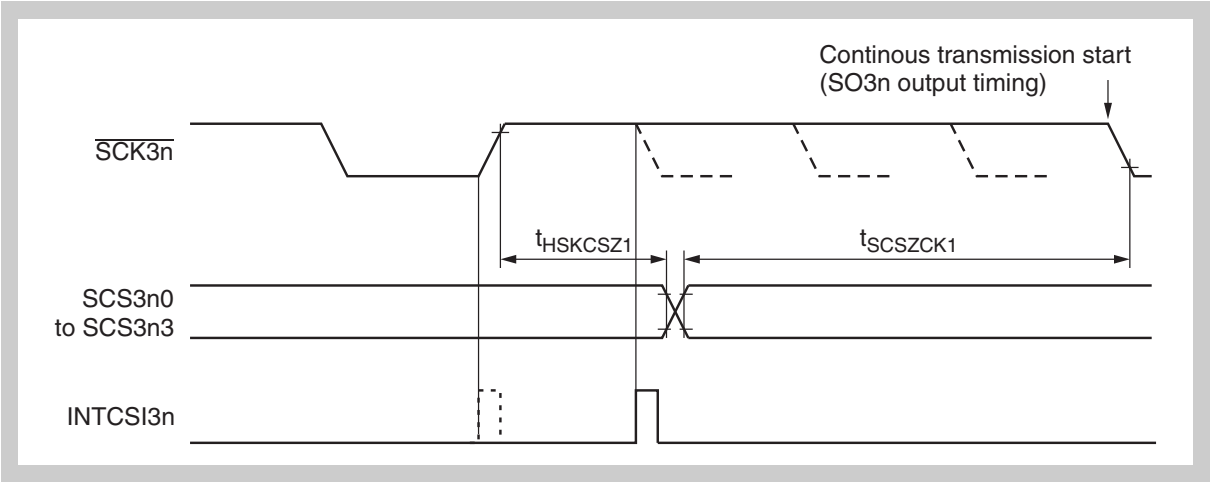


Figure 25-23 CSI3 chip select timing (master mode only) (CSIT = 1, CSWE = 1, CSMD = 0)

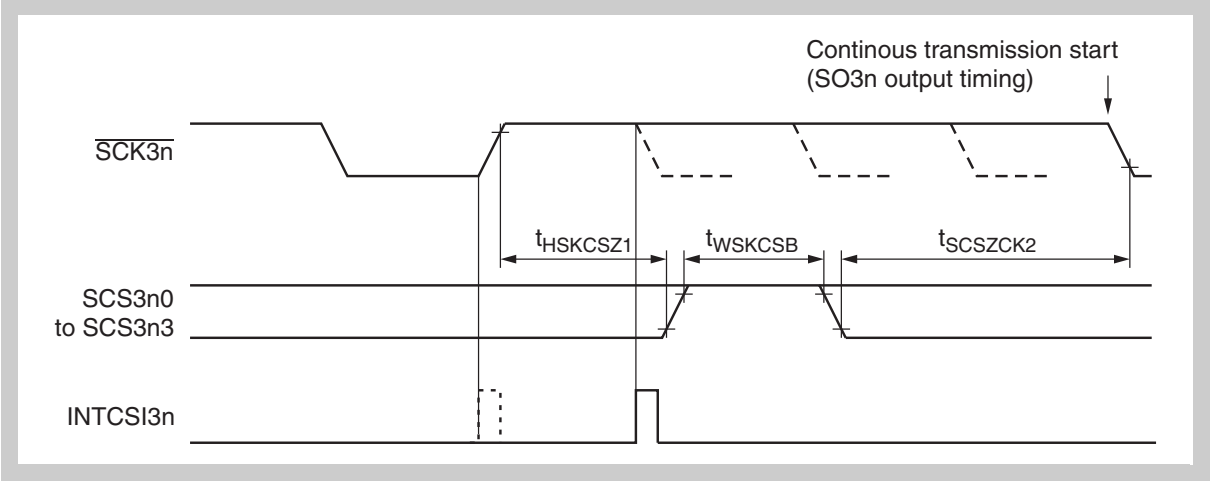


Figure 25-24 CSI3 chip select timing (master mode only) (CSIT = 1, CSWE = 1, CSMD = 1)

25.5.3 A/D converter characteristics

Table 25-17 A/D converter characteristics

Parameter	Symbol	MIN.	TYP.	MAX.	Unit
Resolution	-		10		Bit
Overall error	-			±4	LSB
Conversion time	T_{CONV}	2		8	µs
Sampling time	T_{SAM}	0.375		1.5	µs
Analog input voltage	V_{IAN}	AV_{SS}		AV_{DD}	V
Analog supply current	I_{AVDD}			2.4	mA
Reference voltage	AV_{REF}	AV_{DD}		AV_{DD}	V

Table 25-18 Analog input characteristics

Parameter	Symbol	MIN.	TYP.	MAX.	Unit
Equivalent circuit parameters	R1			600	Ω
	R2			160	Ω
	C1			15	pF
	C2			3.5	pF
	C3			5.8	pF

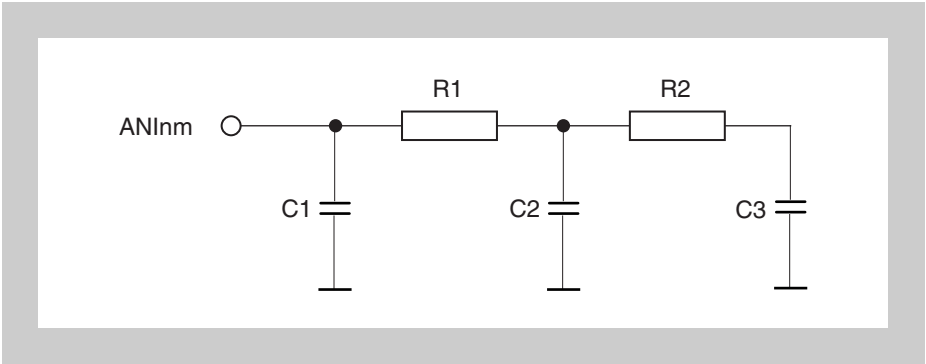


Figure 25-25 Equivalent circuit of analog inputs

25.6 Flash Programming Characteristics

Unless specified otherwise, the following conditions are assumed for all characteristics in this chapter.

V_{DD3x} = AV_{DD} = 3.0 V to 3.6 V
V_{DD1x} = CV_{DD} = 1.35 V to 1.65 V
V_{SS1x} = CV_{SS} = V_{SS3x} = AV_{SSx} = 0 V

Table 25-19 Flash memory basic characteristics

Parameter	Condition	Symbol	MIN.	TYP.	MAX.	Unit
Number of rewrites		C_{WRT}			100	times/ block
Ambient programming temperature		T_{APRG}	-40		+100	°C
Data retention time	6000 h key-on time		15			years

Table 25-20 Flash memory programming characteristics

Parameter	Symbol	MIN.	TYP.	MAX.	Unit
Write time			30	300	µs/ word
Erase time			0.2	2	s

Table 25-21 Serial write operation characteristics

Parameter	Symbol	MIN.	TYP.	MAX.	Unit
V_{DD}↑ setup time to FLMD0↑	t_{DRPSR}	0			ns
V_{DD}↑ setup time to $\overline{\text{RESET}}$↑	t_{DRRR}	2			ms
FLMD0 setup time to $\overline{\text{RESET}}$↑	t_{PSRRF}	2			ms
FLMD0 count start time from $\overline{\text{RESET}}$↑	t_{RFCF}	10			ms
FLMD0 count time	t_{COUNT}			10	ms
FLMD0 counter high-level width/low-level width	t_{CH}/t_{CL}	10			µs

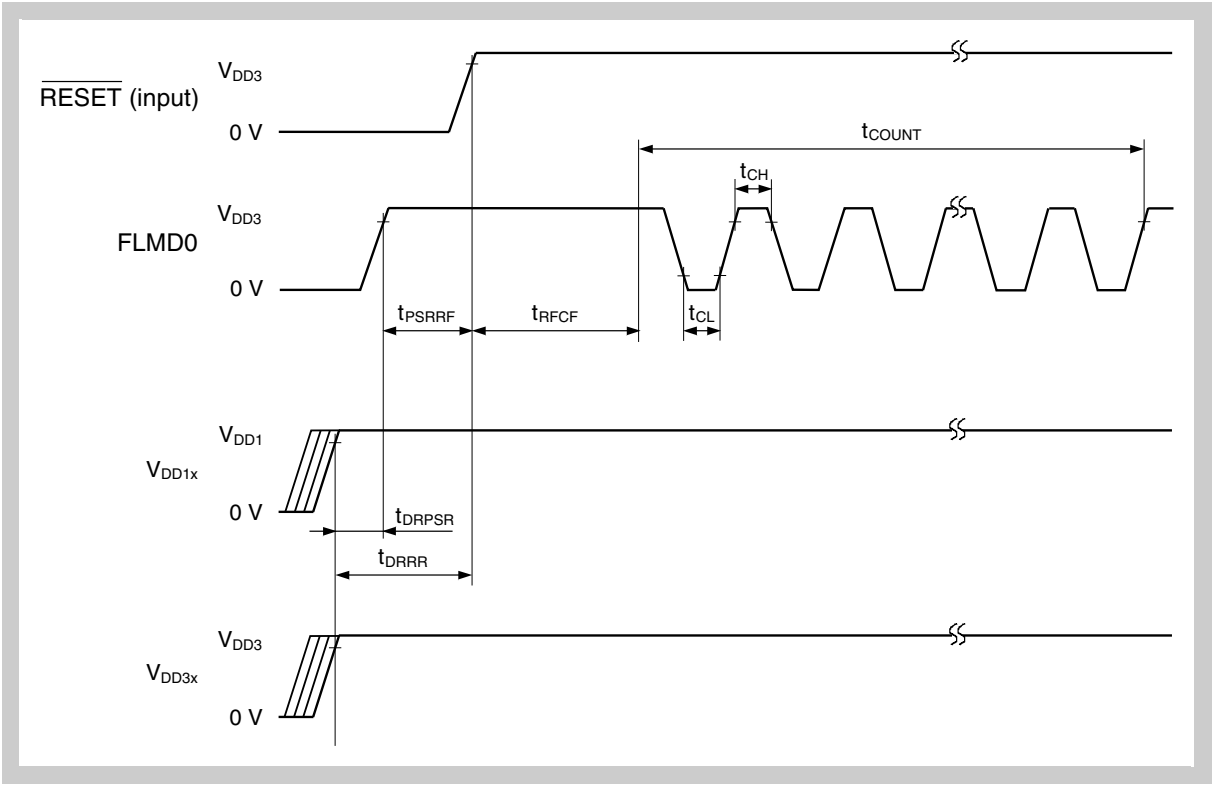
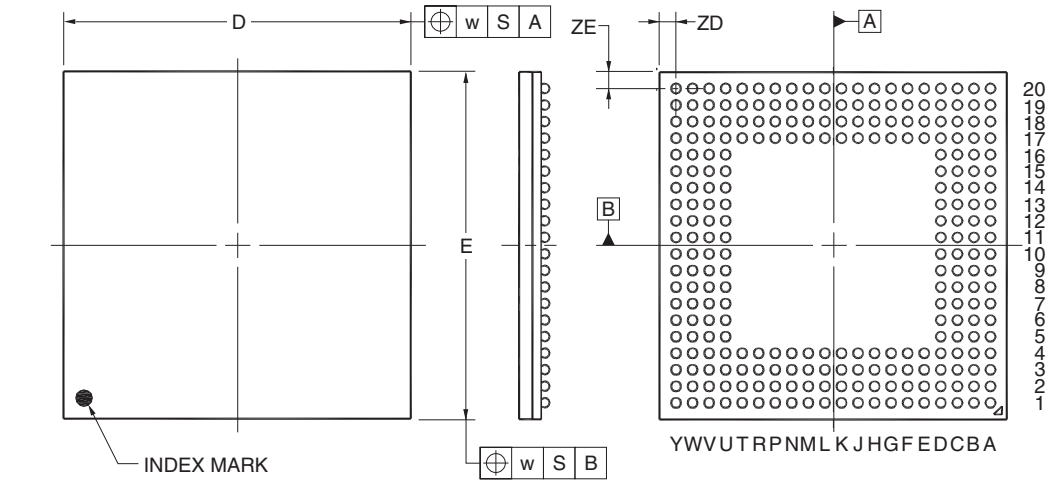
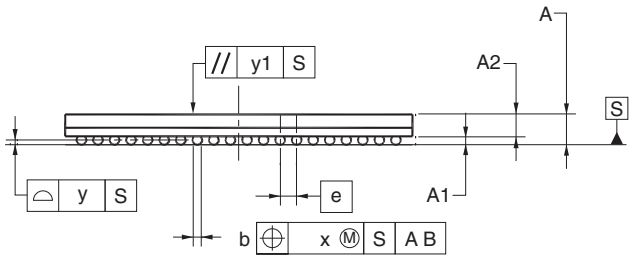


Figure 25-26 Serial Write Operation Characteristics

Chapter 26 Package Drawings

256-PIN PLASTIC BGA (21x21)





(UNIT:mm)

ITEM	DIMENSIONS
D	21.00±0.10
E	21.00±0.10
w	0.30
e	1.00
A	1.83±0.17
A1	0.50±0.10
A2	1.33
b	0.60±0.10
x	0.15
y	0.15
y1	0.35
ZD	1.00
ZE	1.00

P256F1-100-JN4

Figure 26-1 256-Pin Plastic BGA (Fine Pitch) (21 x 21)

Chapter 27 Recommended Soldering Conditions

For general information of soldering conditions, refert to the document “Renesas Semiconductor Package Mount Manual” (R50ZZ0003EJ) available on the Renesas internet page:

http://www.renesas.eu/products/package/manual/index.jsp

For particular soldering methods and conditions please consult your Renesas sales representative.

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-1 lists all CAN special function registers.

Table A-2 lists all other special function registers.

A.1 CAN Registers

The CAN registers are accessible via the programmable peripheral I/O area. The addresses are given as offsets to the programmable peripheral base address (refer to *“CAN module register and message buffer addresses” on page 733*.)

Table A-1 CAN special function registers (1/3)

Address offset	Register name	Symbol	1	8	16	32	Initial value
0x000	CAN0 Global Macro Control register	C0GMCTRL	-	-	R/W	-	0x0000
0x000	CAN0 Global Macro Control register low byte	C0GMCTRLL	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	C0GMABTL	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

Table A-1 CAN special function registers (2/3)

Address offset	Register name	Symbol	1	8	16	32	Initial value
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
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, refer to *Table 18-20 on page 736*.						
0x600	CAN1 Global Macro Control register	C1GMCTRL	-	-	R/W	-	0x0000
0x600	CAN1 Global Macro Control register low byte	C1GMCTRLL	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

Table A-1 CAN special function registers (3/3)

Address offset	Register name	Symbol	1	8	16	32	Initial value
0x654	CAN1 Module Error Counter	C1ERC	-	-	R/W	-	0x0000
0x656	CAN1 Module Interrupt Enable register	C1IE	-	-	R/W	-	0x0000
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, refer to *Table 18-20 on page 736*.						

A.2 Peripheral I/O registers list

Table A-2 Peripheral I/O Registers (1/14)

Address	Register Name	Symbol	1	8	16	Initial value
0xFFFFF000	Port register AL	PAL			R/W	undefined
0xFFFFF000	Port register ALL	PALL	R/W	R/W		undefined
0xFFFFF001	Port register ALH	PALH	R/W	R/W		undefined
0xFFFFF002	Port register AH	PAH	R/W	R/W		undefined
0xFFFFF004	Port register DL	PDL			R/W	undefined
0xFFFFF004	Port register DLL	PDLL	R/W	R/W		undefined
0xFFFFF005	Port register DLH	PDLH	R/W	R/W		undefined
0xFFFFF006	Port register DH	PDH			R/W	undefined
0xFFFFF006	Port register DHL	PDHL	R/W	R/W		undefined
0xFFFFF007	Port register DHH	PDHH	R/W	R/W		undefined
0xFFFFF008	Port register CS	PCS	R/W	R/W		undefined
0xFFFFF00A	Port register CT	PCT	R/W	R/W		undefined
0xFFFFF00C	Port register CM	PCM	R/W	R/W		undefined
0xFFFFF00E	Port register CD	PCD	R/W	R/W		undefined
0xFFFFF020	Port mode register AL	PMAL			R/W	0xFFFF
0xFFFFF020	Port mode register ALL	PMALL	R/W	R/W		0xFF
0xFFFFF021	Port mode register ALH	PMALH	R/W	R/W		0xFF
0xFFFFF022	Port mode register AH	PMAH	R/W	R/W		0xFF
0xFFFFF024	Port mode register DL	PMDL			R/W	0xFFFF
0xFFFFF024	Port mode register DLL	PMDLL	R/W	R/W		0xFF
0xFFFFF025	Port mode register DLH	PMDLH	R/W	R/W		0xFF
0xFFFFF026	Port mode register DH	PMDH			R/W	0xFFFF
0xFFFFF026	Port mode register DHL	PMDHL	R/W	R/W		0xFF
0xFFFFF027	Port mode register DHH	PMDHH	R/W	R/W		0xFF
0xFFFFF028	Port mode register CS	PMCS	R/W	R/W		0xFF
0xFFFFF02A	Port mode register CT	PMCT	R/W	R/W		0xFF
0xFFFFF02C	Port mode register CM	PMCM	R/W	R/W		0xFF
0xFFFFF02E	Port mode register CD	PMCD	R/W	R/W		0xFF
0xFFFFF040	Port mode control register AL	PMCAL			R/W	0x0000/ 0xFFFF [a]
0xFFFFF040	Port mode control register ALL [b]	PMCALL	R/W	R/W		0x00/ 0xFF [a]
0xFFFFF041	Port mode control register ALH [b]	PMCALH	R/W	R/W		0x00/ 0xFF [a]
0xFFFFF042	Port mode control register AH [b]	PMCAH	R/W	R/W		0x00/ 0x3F [a]
0xFFFFF044	Port mode control register DL [b]	PMCDL			R/W	0x0000/ 0xFFFF [a]
0xFFFFF044	Port mode control register DLL [b]	PMCDLL	R/W	R/W		0x00/ 0xFF [a]

Table A-2 Peripheral I/O Registers (2/14)

Address	Register Name	Symbol	1	8	16	Initial value
0xFFFFF045	Port mode control register DLH [b]	PMCDLH	R/W	R/W		0x00/ 0xFF [a]
0xFFFFF046	Port mode control register DH [b]	PMCDH			R/W	0x0000/ 0xFFFF [a]
0xFFFFF046	Port mode control register DHL [b]	PMCDHL	R/W	R/W		0x00/ 0xFF [a]
0xFFFFF047	Port mode control register DHH [b]	PMCDHH	R/W	R/W		0x00/ 0xFF [a]
0xFFFFF048	Port mode control register CS [b]	PMCCS	R/W	R/W		0x00/ 0x1B [a]
0xFFFFF04A	Port mode control register CT [b]	PMCCT	R/W	R/W		0x00/ 0x30 [a]
0xFFFFF04C	Port mode control register CM [b]	PMCCM	R/W	R/W		0x00/ 0x01 [a]
0xFFFFF04E	Port mode control register CD [b]	PMCCD	R/W	R/W		0x00/ 0x3F [a]
0xFFFFF060	Chip area select control register 0 [c]	CSC0			R/W	0x2C11
0xFFFFF062	Chip area select control register 1 [c]	CSC1			R/W	0x2C11
0xFFFFF064	Peripheral area select control register	BPC			R/W	0x0FFF
0xFFFFF066	Bus size configuration register [c]	BSC			R/W	0xAAAA
0xFFFFF068	Endian configuration register [c]	BEC			R/W	0x0000
0xFFFFF06E	System wait control register	VSWC	R/W	R/W		0x77
0xFFFFF100	Interrupt mask register 0	IMR0			R/W	0xFFFF
0xFFFFF100	Interrupt mask register 0L	IMR0L	R/W	R/W		0xFF
0xFFFFF101	Interrupt mask register 0H	IMR0H	R/W	R/W		0xFF
0xFFFFF102	Interrupt mask register 1	IMR1			R/W	0xFFFF
0xFFFFF102	Interrupt mask register 1L	IMR1L	R/W	R/W		0xFF
0xFFFFF103	Interrupt mask register 1H	IMR1H	R/W	R/W		0xFF
0xFFFFF104	Interrupt mask register 2	IMR2			R/W	0xFFFF
0xFFFFF104	Interrupt mask register 2L	IMR2L	R/W	R/W		0xFF
0xFFFFF105	Interrupt mask register 2H	IMR2H	R/W	R/W		0xFF
0xFFFFF106	Interrupt mask register 3	IMR3			R/W	0xFFFF
0xFFFFF106	Interrupt mask register 3L	IMR3L	R/W	R/W		0xFF
0xFFFFF107	Interrupt mask register 3H	IMR3H	R/W	R/W		0xFF
0xFFFFF108	Interrupt mask register 4	IMR4			R/W	0xFFFF
0xFFFFF108	Interrupt mask register 4L	IMR4L	R/W	R/W		0xFF
0xFFFFF109	Interrupt mask register 4H	IMR4H	R/W	R/W		0xFF
0xFFFFF10A	Interrupt mask register 5	IMR5			R/W	0xFFFF
0xFFFFF10A	Interrupt mask register 5L	IMR5L	R/W	R/W		0xFF
0xFFFFF10B	Interrupt mask register 5H	IMR5H	R/W	R/W		0xFF
0xFFFFF10C	Interrupt mask register 6	IMR6			R/W	0xFFFF
0xFFFFF10C	Interrupt mask register 6L	IMR6L	R/W	R/W		0xFF
0xFFFFF10D	Interrupt mask register 6H	IMR6H	R/W	R/W		0xFF

Table A-2 Peripheral I/O Registers (3/14)

Address	Register Name	Symbol	1	8	16	Initial value
0xFFFFF10E	Interrupt mask register 7	IMR7			R/W	0xFFFF
0xFFFFF10E	Interrupt mask register 7L	IMR7L	R/W	R/W		0xFF
0xFFFFF10F	Interrupt mask register 7H	IMR7H	R/W	R/W		0xFF
0xFFFFF110	Interrupt control register 0	PIC0	R/W	R/W		0x47
0xFFFFF112	Interrupt control register 1	PIC1	R/W	R/W		0x47
0xFFFFF114	Interrupt control register 2	PIC2	R/W	R/W		0x47
0xFFFFF116	Interrupt control register 3	PIC3	R/W	R/W		0x47
0xFFFFF118	Interrupt control register 4	PIC4	R/W	R/W		0x47
0xFFFFF11A	Interrupt control register 5	PIC5	R/W	R/W		0x47
0xFFFFF11C	Interrupt control register 6	PIC6	R/W	R/W		0x47
0xFFFFF11E	Interrupt control register 7	PIC7	R/W	R/W		0x47
0xFFFFF120	Interrupt control register 8	PIC8	R/W	R/W		0x47
0xFFFFF122	Interrupt control register 9	PIC9	R/W	R/W		0x47
0xFFFFF124	Interrupt control register 10	PIC10	R/W	R/W		0x47
0xFFFFF126	Interrupt control register 11	PIC11	R/W	R/W		0x47
0xFFFFF128	Interrupt control register 12	PIC12	R/W	R/W		0x47
0xFFFFF12A	Interrupt control register 13	PIC13	R/W	R/W		0x47
0xFFFFF12C	Interrupt control register 14	PIC14	R/W	R/W		0x47
0xFFFFF12E	Interrupt control register 15	PIC15	R/W	R/W		0x47
0xFFFFF130	Interrupt control register 16	PIC16	R/W	R/W		0x47
0xFFFFF132	Interrupt control register 17	PIC17	R/W	R/W		0x47
0xFFFFF134	Interrupt control register 18	PIC18	R/W	R/W		0x47
0xFFFFF136	Interrupt control register 19	PIC19	R/W	R/W		0x47
0xFFFFF138	Interrupt control register 20	PIC20	R/W	R/W		0x47
0xFFFFF13A	Interrupt control register 21	PIC21	R/W	R/W		0x47
0xFFFFF13C	Interrupt control register 22	PIC22	R/W	R/W		0x47
0xFFFFF13E	Interrupt control register 23	PIC23	R/W	R/W		0x47
0xFFFFF140	Interrupt control register 24	PIC24	R/W	R/W		0x47
0xFFFFF142	Interrupt control register 25	PIC25	R/W	R/W		0x47
0xFFFFF144	Interrupt control register 26	PIC26	R/W	R/W		0x47
0xFFFFF146	Interrupt control register 27	PIC27	R/W	R/W		0x47
0xFFFFF148	Interrupt control register 28	PIC28	R/W	R/W		0x47
0xFFFFF14A	Interrupt control register 29	PIC29	R/W	R/W		0x47
0xFFFFF14C	Interrupt control register 30	PIC30	R/W	R/W		0x47
0xFFFFF14E	Interrupt control register 31	PIC31	R/W	R/W		0x47
0xFFFFF150	Interrupt control register 32	PIC32	R/W	R/W		0x47
0xFFFFF152	Interrupt control register 33	PIC33	R/W	R/W		0x47
0xFFFFF154	Interrupt control register 34	PIC34	R/W	R/W		0x47
0xFFFFF156	Interrupt control register 35	PIC35	R/W	R/W		0x47
0xFFFFF158	Interrupt control register 36	PIC36	R/W	R/W		0x47
0xFFFFF15A	Interrupt control register 37	PIC37	R/W	R/W		0x47

Table A-2 Peripheral I/O Registers (4/14)

Address	Register Name	Symbol	1	8	16	Initial value
0xFFFFF15C	Interrupt control register 38	PIC38	R/W	R/W		0x47
0xFFFFF15E	Interrupt control register 39	PIC39	R/W	R/W		0x47
0xFFFFF160	Interrupt control register 40	PIC40	R/W	R/W		0x47
0xFFFFF162	Interrupt control register 41	PIC41	R/W	R/W		0x47
0xFFFFF164	Interrupt control register 42	PIC42	R/W	R/W		0x47
0xFFFFF166	Interrupt control register 43	PIC43	R/W	R/W		0x47
0xFFFFF168	Interrupt control register 44	PIC44	R/W	R/W		0x47
0xFFFFF16A	Interrupt control register 45	PIC45	R/W	R/W		0x47
0xFFFFF16C	Interrupt control register 46	PIC46	R/W	R/W		0x47
0xFFFFF16E	Interrupt control register 47	PIC47	R/W	R/W		0x47
0xFFFFF170	Interrupt control register 48	PIC48	R/W	R/W		0x47
0xFFFFF172	Interrupt control register 49	PIC49	R/W	R/W		0x47
0xFFFFF174	Interrupt control register 50	PIC50	R/W	R/W		0x47
0xFFFFF176	Interrupt control register 51	PIC51	R/W	R/W		0x47
0xFFFFF178	Interrupt control register 52	PIC52	R/W	R/W		0x47
0xFFFFF17A	Interrupt control register 53	PIC53	R/W	R/W		0x47
0xFFFFF17C	Interrupt control register 54	PIC54	R/W	R/W		0x47
0xFFFFF17E	Interrupt control register 55	PIC55	R/W	R/W		0x47
0xFFFFF180	Interrupt control register 56	PIC56	R/W	R/W		0x47
0xFFFFF182	Interrupt control register 57	PIC57	R/W	R/W		0x47
0xFFFFF184	Interrupt control register 58	PIC58	R/W	R/W		0x47
0xFFFFF186	Interrupt control register 59	PIC59	R/W	R/W		0x47
0xFFFFF188	Interrupt control register 60	PIC60	R/W	R/W		0x47
0xFFFFF18A	Interrupt control register 61	PIC61	R/W	R/W		0x47
0xFFFFF18C	Interrupt control register 62	PIC62	R/W	R/W		0x47
0xFFFFF18E	Interrupt control register 63	PIC63	R/W	R/W		0x47
0xFFFFF190	Interrupt control register 64	PIC64	R/W	R/W		0x47
0xFFFFF192	Interrupt control register 65	PIC65	R/W	R/W		0x47
0xFFFFF194	Interrupt control register 66	PIC66	R/W	R/W		0x47
0xFFFFF196	Interrupt control register 67	PIC67	R/W	R/W		0x47
0xFFFFF198	Interrupt control register 68	PIC68	R/W	R/W		0x47
0xFFFFF19A	Interrupt control register 69	PIC69	R/W	R/W		0x47
0xFFFFF19C	Interrupt control register 70	PIC70	R/W	R/W		0x47
0xFFFFF19E	Interrupt control register 71	PIC71	R/W	R/W		0x47
0xFFFFF1A0	Interrupt control register 72	PIC72	R/W	R/W		0x47
0xFFFFF1A2	Interrupt control register 73	PIC73	R/W	R/W		0x47
0xFFFFF1A4	Interrupt control register 74	PIC74	R/W	R/W		0x47
0xFFFFF1A6	Interrupt control register 75 [c]	PIC75	R/W	R/W		0x47
0xFFFFF1A8	Interrupt control register 76 [c]	PIC76	R/W	R/W		0x47
0xFFFFF1AA	Interrupt control register 77 [c]	PIC77	R/W	R/W		0x47
0xFFFFF1AC	Interrupt control register 78 [c]	PIC78	R/W	R/W		0x47

Table A-2 Peripheral I/O Registers (5/14)

Address	Register Name	Symbol	1	8	16	Initial value
0xFFFFF1AE	Interrupt control register 79	PIC79	R/W	R/W		0x47
0xFFFFF1B0	Interrupt control register 80	PIC80	R/W	R/W		0x47
0xFFFFF1B2	Interrupt control register 81	PIC81	R/W	R/W		0x47
0xFFFFF1B4	Interrupt control register 82 [c]	PIC82	R/W	R/W		0x47
0xFFFFF1B6	Interrupt control register 83 [c]	PIC83	R/W	R/W		0x47
0xFFFFF1B8	Interrupt control register 84 [c]	PIC84	R/W	R/W		0x47
0xFFFFF1BA	Interrupt control register 85	PIC85	R/W	R/W		0x47
0xFFFFF1BC	Interrupt control register 86	PIC86	R/W	R/W		0x47
0xFFFFF1BE	Interrupt control register 87 [c]	PIC87	R/W	R/W		0x47
0xFFFFF1C0	Interrupt control register 88 [c]	PIC88	R/W	R/W		0x47
0xFFFFF1C2	Interrupt control register 89	PIC89	R/W	R/W		0x47
0xFFFFF1C4	Interrupt control register 90	PIC90	R/W	R/W		0x47
0xFFFFF1C6	Interrupt control register 91	PIC91	R/W	R/W		0x47
0xFFFFF1C8	Interrupt control register 92	PIC92	R/W	R/W		0x47
0xFFFFF1CA	Interrupt control register 93	PIC93	R/W	R/W		0x47
0xFFFFF1CC	Interrupt control register 94	PIC94	R/W	R/W		0x47
0xFFFFF1CE	Interrupt control register 95	PIC95	R/W	R/W		0x47
0xFFFFF1D0	Interrupt control register 96	PIC96	R/W	R/W		0x47
0xFFFFF1D2	Interrupt control register 97 [c]	PIC97	R/W	R/W		0x47
0xFFFFF1D4	Interrupt control register 98 [c]	PIC98	R/W	R/W		0x47
0xFFFFF1D6	Interrupt control register 99 [c]	PIC99	R/W	R/W		0x47
0xFFFFF1D8	Interrupt control register 100 [c]	PIC100	R/W	R/W		0x47
0xFFFFF1DA	Interrupt control register 101 [c]	PIC101	R/W	R/W		0x47
0xFFFFF1DC	Interrupt control register 102 [c]	PIC102	R/W	R/W		0x47
0xFFFFF1DE	Interrupt control register 103	PIC103	R/W	R/W		0x47
0xFFFFF1E0	Interrupt control register 104	PIC104	R/W	R/W		0x47
0xFFFFF1E2	Interrupt control register 105	PIC105	R/W	R/W		0x47
0xFFFFF1FA	Interrupt 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

Table A-2 Peripheral I/O Registers (6/14)

Address	Register Name	Symbol	1	8	16	Initial value
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
0xFFFFF253	A/D conversion result register 11H	ADCR11H		R		undefined
0xFFFFF254	A/D conversion result register 12	ADCR112			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 trigger select register 0	ADTRSEL0	R/W	R/W		0x00
0xFFFFF272	A/D trigger select register 1	ADTRSEL1	R/W	R/W		0x00
0xFFFFF300	Memory transfer start address register 0	MAR0			R/W	undefined
0xFFFFF302	Memory transfer start address register 1	MAR1			R/W	undefined
0xFFFFF304	Memory transfer start address register 2	MAR2			R/W	undefined

Table A-2 Peripheral I/O Registers (7/14)

Address	Register Name	Symbol	1	8	16	Initial value
0xFFFFF306	Memory transfer start address register 3	MAR3			R/W	undefined
0xFFFFF308	Memory transfer start address register 4	MAR4			R/W	undefined
0xFFFFF30A	Memory transfer start address register 5	MAR5			R/W	undefined
0xFFFFF30C	Memory transfer start address register 6	MAR6			R/W	undefined
0xFFFFF30E	Memory transfer start address register 7	MAR7			R/W	undefined
0xFFFFF314	SFR transfer start address register 2	SAR2	R/W	R/W		undefined
0xFFFFF316	SFR transfer start address register 3	SAR3	R/W	R/W		undefined
0xFFFFF320	DMA transfer count register 0	DTCR0	R/W	R/W		undefined
0xFFFFF322	DMA transfer count register 1	DTCR1	R/W	R/W		undefined
0xFFFFF324	DMA transfer count register 2	DTCR2	R/W	R/W		undefined
0xFFFFF326	DMA transfer count register 3	DTCR3	R/W	R/W		undefined
0xFFFFF328	DMA transfer count register 4	DTCR4	R/W	R/W		undefined
0xFFFFF32A	DMA transfer count register 5	DTCR5	R/W	R/W		undefined
0xFFFFF32C	DMA transfer count register 6	DTCR6	R/W	R/W		undefined
0xFFFFF32E	DMA transfer count register 7	DTCR7	R/W	R/W		undefined
0xFFFFF330	DMA mode control register	DMAMC	R/W	R/W		0x00
0xFFFFF332	DMA status register	DMAS	R/W	R/W		0x00
0xFFFFF334	DMA data size control register	DMADSC	R/W	R/W		0x00
0xFFFFF348	DMA trigger factor register 4	DTFR4	R/W	R/W		0x00
0xFFFFF34A	DMA trigger factor register 5	DTFR5	R/W	R/W		0x00
0xFFFFF34C	DMA trigger factor register 6	DTFR6	R/W	R/W		0x00
0xFFFFF34E	DMA trigger factor register 7	DTFR7	R/W	R/W		0x00
0xFFFFF400	Port register 0	P0	R	R		undefined
0xFFFFF402	Port register 1	P1	R/W	R/W		undefined
0xFFFFF404	Port register 2	P2	R/W	R/W		undefined
0xFFFFF406	Port register 3	P3	R/W	R/W		undefined
0xFFFFF408	Port register 4	P4	R/W	R/W		undefined
0xFFFFF40A	Port register 5	P5	R/W	R/W		undefined
0xFFFFF40C	Port register 6	P6	R/W	R/W		undefined
0xFFFFF40E	Port register 7	P7	R/W	R/W		undefined
0xFFFFF410	Port register 8	P8	R/W	R/W		undefined
0xFFFFF412	Port register 9	P9	R/W	R/W		undefined
0xFFFFF414	Port register 10	P10	R/W	R/W		undefined
0xFFFFF422	Port mode register 1	PM1	R/W	R/W		0xFF
0xFFFFF424	Port mode register 2	PM2	R/W	R/W		0xFF
0xFFFFF426	Port mode register 3	PM3	R/W	R/W		0xFF
0xFFFFF428	Port mode register 4	PM4	R/W	R/W		0xFF
0xFFFFF42A	Port mode register 5	PM5	R/W	R/W		0xFF
0xFFFFF42C	Port mode register 6	PM6	R/W	R/W		0xFF
0xFFFFF42E	Port mode register 7	PM7	R/W	R/W		0xFF
0xFFFFF430	Port mode register 8	PM8	R/W	R/W		0xFF

Table A-2 Peripheral I/O Registers (8/14)

Address	Register Name	Symbol	1	8	16	Initial value
0xFFFFF432	Port mode register 9	PM9	R/W	R/W		0xFF
0xFFFFF434	Port mode register 10	PM10	R/W	R/W		0xFF
0xFFFFF442	Port mode control register 1	PMC1	R/W	R/W		0x00
0xFFFFF444	Port mode control register 2	PMC2	R/W	R/W		0x00
0xFFFFF446	Port mode control register 3	PMC3	R/W	R/W		0x00
0xFFFFF448	Port mode control register 4	PMC4	R/W	R/W		0x00
0xFFFFF44A	Port mode control register 5	PMC5	R/W	R/W		0x00
0xFFFFF44C	Port mode control register 6	PMC6	R/W	R/W		0x00
0xFFFFF44E	Port mode control register 7	PMC7	R/W	R/W		0x00
0xFFFFF450	Port mode control register 8	PMC8	R/W	R/W		0x00
0xFFFFF452	Port mode control register 9	PMC9	R/W	R/W		0x00
0xFFFFF454	Port mode control register 10	PMC10	R/W	R/W		0x00
0xFFFFF480	Bus cycle type configuration register 0 [c]	BCT0			R/W	0xCCCC
0xFFFFF482	Bus cycle type configuration register 1 [c]	BCT1			R/W	0xCCCC
0xFFFFF484	Data wait control register 0 [c]	DWC0			R/W	0x7777
0xFFFFF486	Data wait control register 1 [c]	DWC1			R/W	0x7777
0xFFFFF488	Address wait control register [c]	AWC			R/W	0x0000
0xFFFFF48A	Bus and cycle control register [c]	BCC			R/W	0xAAAA
0xFFFFF48E	Bus clock dividing control register	DVC		R/W		0x01
0xFFFFF4C0	iRAM parity error flag register [c]	RAMERR	R/W	R/W		0x00
0xFFFFF4C2	iRAM parity error address register	RAMPADD			R/W	800x00
0xFFFFF580	TMR0 control register 0	TR0CTL0	R/W	R/W		0x00
0xFFFFF581	TMR0 control register 1	TR0CTL1	R/W	R/W		0x00
0xFFFFF582	TMR0 I/O control register 0	TR0IOC0	R/W	R/W		0x00
0xFFFFF585	TMR0 I/O control register 3	TR0IOC3	R/W	R/W		0x00
0xFFFFF586	TMR0 I/O control register 4	TR0IOC4	R/W	R/W		0x00
0xFFFFF587	TMR0 option register 0	TR0OPT0	R/W	R/W		0x00
0xFFFFF588	TMR0 option register 2	TR0OPT2	R/W	R/W		0x00
0xFFFFF589	TMR0 option register 3	TR0OPT3	R/W	R/W		0x00
0xFFFFF58C	TMR0 option register 6	TR0OPT6	R/W	R/W		0x00
0xFFFFF58D	TMR0 option register 7	TR0OPT7	R/W	R/W		0x00
0xFFFFF58E	TMR0 option register 1	TR0OPT1	R/W	R/W		0x00
0xFFFFF590	TMR0 capture/compare register 5	TR0CCR5			R/W	0x0000
0xFFFFF592	TMR0 capture/compare register 4	TR0CCR4			R/W	0x0000
0xFFFFF598	TMR0 capture/compare register 0	TR0CCR0			R/W	0x0000
0xFFFFF59A	TMR0 capture/compare register 3	TR0CCR3			R/W	0x0000
0xFFFFF59C	TMR0 capture/compare register 2	TR0CCR2			R/W	0x0000
0xFFFFF59E	TMR0 capture/compare register 1	TR0CCR1			R/W	0x0000
0xFFFFF5A0	TMR0 dead time set register 0	TR0DTC0			R/W	0x0000
0xFFFFF5A2	TMR0 dead time set register 1	TR0DTC1			R/W	0x0000
0xFFFFF5A4	TMR0 timer counter read register	TR0CNT			R	0x0000

Table A-2 Peripheral I/O Registers (9/14)

Address	Register Name	Symbol	1	8	16	Initial value
0xFFFFF5A6	TMR0 timer sub-counter read register	TR0SBC			R	0x0000
0xFFFFF5C0	TMR1 control register 0	TR1CTL0	R/W	R/W		0x00
0xFFFFF5C1	TMR1 control register 1	TR1CTL1	R/W	R/W		0x00
0xFFFFF5C2	TMR1 I/O control register 0	TR1IOC0	R/W	R/W		0x00
0xFFFFF5C3	TMR1 I/O control register 1	TR1IOC1	R/W	R/W		0x00
0xFFFFF5C4	TMR1 I/O control register 2	TR1IOC2	R/W	R/W		0x00
0xFFFFF5C5	TMR1 I/O control register 3	TR1IOC3	R/W	R/W		0x00
0xFFFFF5C6	TMR1 I/O control register 4	TR1IOC4	R/W	R/W		0x00
0xFFFFF5C7	TMR1 option register 0	TR1OPT0	R/W	R/W		0x00
0xFFFFF5C8	TMR1 option register 2	TR1OPT2	R/W	R/W		0x00
0xFFFFF5C9	TMR1 option register 3	TR1OPT3	R/W	R/W		0x00
0xFFFFF5CC	TMR1 option register 6	TR1OPT6	R/W	R/W		0x00
0xFFFFF5CD	TMR1 option register 7	TR1OPT7	R/W	R/W		0x00
0xFFFFF5CE	TMR1 option register 1	TR1OPT1	R/W	R/W		0x00
0xFFFFF5D0	TMR1 capture/compare register 5	TR1CCR5			R/W	0x0000
0xFFFFF5D2	TMR1 capture/compare register 4	TR1CCR4			R/W	0x0000
0xFFFFF5D8	TMR1 capture/compare register 0	TR1CCR0			R/W	0x0000
0xFFFFF5DA	TMR1 capture/compare register 3	TR1CCR3			R/W	0x0000
0xFFFFF5DC	TMR1 capture/compare register 2	TR1CCR2			R/W	0x0000
0xFFFFF5DE	TMR1 capture/compare register 1	TR1CCR1			R/W	0x0000
0xFFFFF5E0	TMR1 dead time set register 0	TR1DTC0			R/W	0x0000
0xFFFFF5E2	TMR1 dead time set register 1	TR1DTC1			R/W	0x0000
0xFFFFF5E4	TMR1 timer counter read register	TR1CNT			R	0x0000
0xFFFFF5E6	TMR1 timer sub-counter read register	TR1SBC			R	0x0000
0xFFFFF600	TMP0 timer control register 0	TP0CTL0	R/W	R/W		0x00
0xFFFFF601	TMP0 timer control register 1	TP0CTL1	R/W	R/W		0x00
0xFFFFF602	TMP0 I/O control register 0	TP0IOC0	R/W	R/W		0x00
0xFFFFF603	TMP0 I/O control register 1	TP0IOC1	R/W	R/W		0x00
0xFFFFF604	TMP0 I/O control register 2	TP0IOC2	R/W	R/W		0x00
0xFFFFF605	TMP0 option register	TP0OPT0	R/W	R/W		0x00
0xFFFFF606	TMP0 capture/compare register 0	TP0CCR0			R/W	0x0000
0xFFFFF608	TMP0 capture/compare register 1	TP0CCR1			R/W	0x0000
0xFFFFF60A	TMP0 count register	TP0CNT			R	0x0000
0xFFFFF610	TMP1 timer control register 0	TP1CTL0	R/W	R/W		0x00
0xFFFFF611	TMP1 timer control register 1	TP1CTL1	R/W	R/W		0x00
0xFFFFF612	TMP1 I/O control register 0	TP1IOC0	R/W	R/W		0x00
0xFFFFF613	TMP1 I/O control register 1	TP1IOC1	R/W	R/W		0x00
0xFFFFF614	TMP1 I/O control register 2	TP1IOC2	R/W	R/W		0x00
0xFFFFF615	TMP1 option register	TP1OPT0	R/W	R/W		0x00
0xFFFFF616	TMP1 capture/compare register 0	TP1CCR0			R/W	0x0000
0xFFFFF618	TMP1 capture/compare register 1	TP1CCR1			R/W	0x0000

Table A-2 Peripheral I/O Registers (10/14)

Address	Register Name	Symbol	1	8	16	Initial value
0xFFFFF61A	TMP1 count register	TP1CNT			R	0x0000
0xFFFFF620	TMP2 timer control register 0	TP2CTL0	R/W	R/W		0x00
0xFFFFF621	TMP2 timer control register 1	TP2CTL1	R/W	R/W		0x00
0xFFFFF622	TMP2 I/O control register 0	TP2IOC0	R/W	R/W		0x00
0xFFFFF623	TMP2 I/O control register 1	TP2IOC1	R/W	R/W		0x00
0xFFFFF624	TMP2 I/O control register 2	TP2IOC2	R/W	R/W		0x00
0xFFFFF625	TMP2 option register	TP2OPT0	R/W	R/W		0x00
0xFFFFF626	TMP2 capture/compare register 0	TP2CCR0			R/W	0x0000
0xFFFFF628	TMP2 capture/compare register 1	TP2CCR1			R/W	0x0000
0xFFFFF62A	TMP2 count register	TP2CNT			R	0x0000
0xFFFFF630	TMP3 timer control register 0	TP3CTL0	R/W	R/W		0x00
0xFFFFF631	TMP3 timer control register 1	TP3CTL1	R/W	R/W		0x00
0xFFFFF632	TMP3 I/O control register 0	TP3IOC0	R/W	R/W		0x00
0xFFFFF633	TMP3 I/O control register 1	TP3IOC1	R/W	R/W		0x00
0xFFFFF634	TMP3 I/O control register 2	TP3IOC2	R/W	R/W		0x00
0xFFFFF635	TMP3 option register	TP3OPT0	R/W	R/W		0x00
0xFFFFF636	TMP3 capture/compare register 0	TP3CCR0			R/W	0x0000
0xFFFFF638	TMP3 capture/compare register 1	TP3CCR1			R/W	0x0000
0xFFFFF63A	TMP3 count register	TP3CNT			R	0x0000
0xFFFFF640	TMP4 timer control register 0	TP4CTL0	R/W	R/W		0x00
0xFFFFF641	TMP4 timer control register 1	TP4CTL1	R/W	R/W		0x00
0xFFFFF642	TMP4 I/O control register 0	TP4IOC0	R/W	R/W		0x00
0xFFFFF643	TMP4 I/O control register 1	TP4IOC1	R/W	R/W		0x00
0xFFFFF644	TMP4 I/O control register 2	TP4IOC2	R/W	R/W		0x00
0xFFFFF645	TMP4 option register	TP4OPT0	R/W	R/W		0x00
0xFFFFF646	TMP4 capture/compare register 0	TP4CCR0			R/W	0x0000
0xFFFFF648	TMP4 capture/compare register 1	TP4CCR1			R/W	0x0000
0xFFFFF64A	TMP4 count register	TP4CNT			R	0x0000
0xFFFFF650	TMP5 timer control register 0	TP5CTL0	R/W	R/W		0x00
0xFFFFF651	TMP5 timer control register 1	TP5CTL1	R/W	R/W		0x00
0xFFFFF652	TMP5 I/O control register 0	TP5IOC0	R/W	R/W		0x00
0xFFFFF653	TMP5 I/O control register 1	TP5IOC1	R/W	R/W		0x00
0xFFFFF654	TMP5 I/O control register 2	TP5IOC2	R/W	R/W		0x00
0xFFFFF655	TMP5 option register	TP5OPT0	R/W	R/W		0x00
0xFFFFF656	TMP5 capture/compare register 0	TP5CCR0			R/W	0x0000
0xFFFFF658	TMP5 capture/compare register 1	TP5CCR1			R/W	0x0000
0xFFFFF65A	TMP5 count register	TP5CNT			R	0x0000
0xFFFFF660	TMP6 timer control register 0	TP6CTL0	R/W	R/W		0x00
0xFFFFF661	TMP6 timer control register 1	TP6CTL1	R/W	R/W		0x00
0xFFFFF662	TMP6 I/O control register 0	TP6IOC0	R/W	R/W		0x00
0xFFFFF663	TMP6 I/O control register 1	TP6IOC1	R/W	R/W		0x00

Table A-2 Peripheral I/O Registers (11/14)

Address	Register Name	Symbol	1	8	16	Initial value
0xFFFFF664	TMP6 I/O control register 2	TP6IOC2	R/W	R/W		0x00
0xFFFFF665	TMP6 option register	TP6OPT0	R/W	R/W		0x00
0xFFFFF666	TMP6 capture/compare register 0	TP6CCR0			R/W	0x0000
0xFFFFF668	TMP6 capture/compare register 1	TP6CCR1			R/W	0x0000
0xFFFFF66A	TMP6 count register	TP6CNT			R	0x0000
0xFFFFF670	TMP7 timer control register 0	TP7CTL0	R/W	R/W		0x00
0xFFFFF671	TMP7 timer control register 1	TP7CTL1	R/W	R/W		0x00
0xFFFFF672	TMP7 I/O control register 0	TP7IOC0	R/W	R/W		0x00
0xFFFFF673	TMP7 I/O control register 1	TP7IOC1	R/W	R/W		0x00
0xFFFFF674	TMP7 I/O control register 2	TP7IOC2	R/W	R/W		0x00
0xFFFFF675	TMP7 option register	TP7OPT0	R/W	R/W		0x00
0xFFFFF676	TMP7 capture/compare register 0	TP7CCR0			R/W	0x0000
0xFFFFF678	TMP7 capture/compare register 1	TP7CCR1			R/W	0x0000
0xFFFFF67A	TMP7 count register	TP7CNT			R	0x0000
0xFFFFF680	TMP8 timer control register 0	TP8CTL0	R/W	R/W		0x00
0xFFFFF681	TMP8 timer control register 1	TP8CTL1	R/W	R/W		0x00
0xFFFFF682	TMP8 I/O control register 0	TP8IOC0	R/W	R/W		0x00
0xFFFFF683	TMP8 I/O control register 1	TP8IOC1	R/W	R/W		0x00
0xFFFFF684	TMP8 I/O control register 2	TP8IOC2	R/W	R/W		0x00
0xFFFFF685	TMP8 option register	TP8OPT0	R/W	R/W		0x00
0xFFFFF686	TMP8 capture/compare register 0	TP8CCR0			R/W	0x0000
0xFFFFF688	TMP8 capture/compare register 1	TP8CCR1			R/W	0x0000
0xFFFFF68A	TMP8 count register	TP8CNT			R	0x0000
0xFFFFF690	TMT0 timer control register 0	TT0CTL0	R/W	R/W		0x00
0xFFFFF691	TMT0 timer control register 1	TT0CTL1	R/W	R/W		0x00
0xFFFFF692	TMT0 timer 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 register	TT0CNT			R	0x0000
0xFFFFF6A0	TMT1 timer control register 0	TT1CTL0	R/W	R/W		0x00
0xFFFFF6A1	TMT1 timer control register 1	TT1CTL1	R/W	R/W		0x00
0xFFFFF6A2	TMT1 timer 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

Table A-2 Peripheral I/O Registers (12/14)

Address	Register Name	Symbol	1	8	16	Initial value
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
0xFFFFF6A7	TMT1 option register 0	TT1OPT0	R/W	R/W		0x00
0xFFFFF6A8	TMT1 option register 1	TT1OPT1	R/W	R/W		0x00
0xFFFFF6A9	TMT1 option register 2	TT1OPT2	R/W	R/W		0x00
0xFFFFF6AA	TMT1 capture/compare register 0	TT1CCR0			R/W	0x0000
0xFFFFF6AC	TMT1 capture/compare register 1	TT1CCR1			R/W	0x0000
0xFFFFF6AE	TMT1 counter read register	TT1CNT			R	0x0000
0xFFFFF6B0	Timer ENC10 count register [c]	TMENC10			R/W	0x0000
0xFFFFF6B2	Compare register 100 [c]	CM100			R/W	0x0000
0xFFFFF6B4	Compare register 101 [c]	CM101			R/W	0x0000
0xFFFFF6B6	Capture/Compare register 100 [c]	CC100			R/W	0x0000
0xFFFFF6B8	Capture/Compare register 101 [c]	CC101			R/W	0x0000
0xFFFFF6BA	Capture/Compare control register 10 [c]	CCR10	R/W	R/W		0x00
0xFFFFF6BB	Timer unit mode register 10 [c]	TUM10	R/W	R/W		0x00
0xFFFFF6BC	Timer control register 10 [c]	TMC10	R/W	R/W		0x00
0xFFFFF6BD	Signal edge selection register 10 [c]	SESA10	R/W	R/W		0x00
0xFFFFF6BE	Prescaler mode register 10 [c]	PRM10	R/W	R/W		0x07
0xFFFFF6BF	Status register 10 [c]	STATUS10	R	R		0x00
0xFFFFF6F0	TMP input source control register 0	TPIC0	R/W	R/W		0x00
0xFFFFF6F2	TMP input source control register 1	TPIC1	R/W	R/W		0x00
0xFFFFF6F4	TMP input source control register 2	TPIC2	R/W	R/W		0x00
0xFFFFF700	Random number register	RNG			R	undefined
0xFFFFF7A0	Noise removal time control register	NRC	R/W	R/W		0x00
0xFFFFF802	Peripheral status register	PHS	R/W	R/W		0x00
0xFFFFF880	Interrupt mode register 0	INTM0	R/W	R/W		0x00
0xFFFFF882	Interrupt mode register 1	INTM1	R/W	R/W		0x00
0xFFFFF884	Interrupt mode register 2	INTM2	R/W	R/W		0x00
0xFFFFF886	Interrupt mode register 3	INTM3	R/W	R/W		0x00
0xFFFFF888	Port emergency shut off control register 5	PESC5	R/W	R/W		0x00
0xFFFFF88A	Port emergency shut off status register 5	ESOST5	R/W	R/W		0x00
0xFFFFF88C	Port emergency shut off control register 6	PESC6	R/W	R/W		0x00
0xFFFFF88E	Port emergency shut off status register 6	ESOST6	R/W	R/W		0x00
0xFFFFF990	Timer T0 counter write buffer register	TT0TCW			R/W	0x0000
0xFFFFF9A0	Timer T1 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		0x00
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

Table A-2 Peripheral I/O Registers (13/14)

Address	Register Name	Symbol	1	8	16	Initial value
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 status register 1	UC0STR1	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		0x00
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 status register 1	UC1STR1	R	R		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 state 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 L	CB0TX0L		R/W		0x00
0xFFFFFD06	CSIB0 transmit data register	CB0TX0			R/W	0x0000
0xFFFFFD20	CSIB1 control register 0 [c]	CB1CTL0	R/W	R/W		0x01
0xFFFFFD21	CSIB1 control register 1 [c]	CB1CTL1	R/W	R/W		0x00
0xFFFFFD22	CSIB1 control register 2 [c]	CB1CTL2		R/W		0x00
0xFFFFFD23	CSIB1 state register [c]	CB1STR	R/W	R/W		0x00
0xFFFFFD24	CSIB1 receive data register [c]	CB1RX0			R	0x0000
0xFFFFFD24	CSIB1 receive data register L [c]	CB1RX0L		R		0x00
0xFFFFFD26	CSIB1 transmit data register L [c]	CB1TX0L		R/W		0x00
0xFFFFFD26	CSIB1 transmit data register [c]	CB1TX0			R/W	0x0000
0xFFFFFD40	CSI30 operation mode register	CSIM30	R/W	R/W		0x00
0xFFFFFD41	CSI30 clock selection register	CSIC30	R/W	R/W		0x07
0xFFFFFD42	CSI30 receive data buffer register	SIRB30			R	0x0000
0xFFFFFD42	CSI30 receive data buffer register L	SIRB30L		R		0x00
0xFFFFFD43	CSI30 receive data buffer register H	SIRB30H		R		0x00
0xFFFFFD44	CSI30 chip selection CSI buffer register L	SFCS30L	R/W	R/W		0xFF
0xFFFFFD44	CSI30 chip selection CSI buffer register	SFCS30			R/W	0xFFFF
0xFFFFFD45	CSI30 chip selection CSI buffer register H	SFCS30H	R	R		0xFF
0xFFFFFD46	CSI30 transmit data CSI buffer register L	SFDB30L		R/W		0x00

Table A-2 Peripheral I/O Registers (14/14)

Address	Register Name	Symbol	1	8	16	Initial value
0xFFFFFD46	CSI30 transmit data CSI buffer register	SFDB30			R/W	0x0000
0xFFFFFD47	CSI30 transmit data CSI buffer register H	SFDB30H		R/W		0x00
0xFFFFFD48	CSI30 SIBUF state register	SFA30	R/W	R/W		0x20
0xFFFFFD49	CSI30 transfer data length select register	CSIL30	R/W	R/W		0x00
0xFFFFFD4C	CSI30 transfer data number specification register	SFN30	R/W	R/W		0x00
0xFFFFFD60	CSI31 operation mode register [c]	CSIM31	R/W	R/W		0x00
0xFFFFFD61	CSI31 clock selection register [c]	CSIC31	R/W	R/W		0x07
0xFFFFFD62	CSI31 receive data buffer register [c]	SIRB31			R	0x0000
0xFFFFFD62	CSI31 receive data buffer register L [c]	SIRB31L		R		0x00
0xFFFFFD63	CSI31 receive data buffer register H [c]	SIRB31H		R		0x00
0xFFFFFD64	CSI31 chip selection CSI buffer register L [c]	SFCS31L	R/W	R/W		0xFF
0xFFFFFD64	CSI31 chip selection CSI buffer register [c]	SFCS31			R/W	0xFFFF
0xFFFFFD65	CSI31 chip selection CSI buffer register H [c]	SFCS31H	R	R		0xFF
0xFFFFFD66	CSI31 transmit data CSI buffer register L [c]	SFDB31L		R/W		0x00
0xFFFFFD66	CSI31 transmit data CSI buffer register [c]	SFDB31			R/W	0x0000
0xFFFFFD67	CSI31 transmit data CSI buffer register H [c]	SFDB31H		R/W		0x00
0xFFFFFD68	CSI31 SIBUF state register [c]	SFA31	R/W	R/W		0x20
0xFFFFFD69	CSI31 transfer data length select register [c]	CSIL31	R/W	R/W		0x00
0xFFFFFD6C	CSI31 transfer data number specification register [c]	SFN31	R/W	R/W		0x00
0xFFFFFDC0	Prescaler mode register 0	PRSM0	R/W	R/W		0x00
0xFFFFFDC1	Prescaler compare register 0	PRSCM0	R/W	R/W		0x00
0xFFFFFDD0	Prescaler mode register 1	PRSM1	R/W	R/W		0x00
0xFFFFFDD1	Prescaler compare register 1	PRSCM1	R/W	R/W		0x00
0xFFFFFDE0	Prescaler mode register 2	PRSM2	R/W	R/W		0x00
0xFFFFFDE1	Prescaler compare register 2	PRSCM2	R/W	R/W		0x00
0xFFFFFE00	DMA wait control register 0	DMAWC0	R/W	R/W		0x37
0xFFFFFE02	DMA wait control register 1	DMAWC1	R/W	R/W		0x07

a) Initial value depends on the selected operating mode (refer to corresponding register description).
b) Only writing of the reset value is permitted for this register on µPD70F3447.
c) Register not available on µPD70F3447.

Revision History

The following revision list shows all functional changes of this document R01UH0439ED0400 compared to the previous user manual version U16580EE3V1UD00

Chapter	Page	Description
–	2	disclaimer changed for Renesas Electronics
1	20	208-pin LQFP package removed
1	21	
1	21	ordering information for 208-pin LQFP removed
3	69	new sub-chapter of CPU system structure added
3	90	peripheral I/O registers list moved to appendix
3	92	programmable peripheral I/O register list moved to appendix
4	104	figure of big endian format changed
4	104	figure of little endian format changed
4	106	chapter "bus width" to "data bus access order" renamed and structure of chapter changed
6	147	1-bit access added for SAR2, SAR3 registers
6	148	1-bit access added for DCTR0 to DTCR7 registers
6	149	1-bit access added for DMAMC register
6	150	1-bit access added for DMAS register
6	150	register name corrected to DMADSC
6	150	1-bit access added for DMADSC register
6	151	1-bit access added for DTFR4 to DTFR7 registesr
6	158	end address corrected to TRnCCR1 in flowchart of DMA operation of channel 2/3
7	174	generating unit of INTBRG2 corrected to AFO
7	175	missing INTAD0 added to list
7	201	name of INTM3 register corrected
9	223	note inserted for TMP8 usage
9	242	external trigger input corrected to TTRGPn pin in flowchart of basic operation in external trigger pulse output mode
9	243	external trigger input corrected to TTRGPn pin in basic operation timing in external trigger pulse output mode
9	245	external trigger input corrected to TTRGPn pin in flowchart of basic operation in one-shot pulse mode
9	267	new subchapter of cautions added
10	293	access size of TRnOPT1 register corrected to 8-bit or 1-bit
10	324	interrupt name in figure corrected to INTTRnCCm
10	329	register of TRnRDE bit corrected
10	342	applicable output pins in interval timer mode corrected to TORn0 to TORn5
10	343	rewriting method corrected in interal timer mode operation list
10	343	description of TRnCCR0 register in interval timer mode refined
10	346	limitation (TMR1 only) for external event count mode added
10	346	external event count mode only available for TMR1, therefore all placeholders "n" replaced by "1" in the sub-chapter
10	346	applicable output pins in external event count mode corrected to TOR10 to TOR15

Chapter	Page	Description
10	346	caution for TORn0 pin removed
10	347	description of TR1CCR0 register in external event count mode refined
10	347	description of TEVTR1 pin added for external event count mode
10	349	figure of timing in event count mode with TR1CCR0 = TR1CCR1 = 0000H removed
10	352	description of TR1CCR0 register in external trigger pulse output mode refined
10	352	function of TEVTR1 pin removed for external trigger pulse output mode (according to previous caution)
10	356	description of TR1CCR0 register in one-shot trigger pulse output mode refined
10	359	output level setting on match with TRnCCR0 in PWM mode corrected
10	360	description of TR1CCR0 register in external trigger pulse output mode refined
10	361	comment for access to TRnCCR0 to TRnCCR5 registers during PWM mode operation corrected to "not rewritten"
10	362	rewrite processing during timer operation in flow chart corrected
10	366	new figure added for operation flow in free-running mode of TMR0 according to the specification
10	367	TR1CCS[1:0] conditions in decision tree of operation flow corrected
10	385	pin name corrected
10	385	active level corrected
10	399	compare value range corrected
10	400	reload timing corrected
10	400	relation corrected in term
10	404	reload timing corrected
10	414	INTTRnOD and INTTRnCD to operation list of PWM mode with dead time added
10	420	new subchapter of cautions added
11	422	TMP input control register 2 and corresponding cross-reference added
11	442	
11	456	figure of even count mode with TTnCCR0 = TTnCCR1 = 0000H removed
11	465	timer write operation in subtitle of figure corrected
11	478	bit name corrected to TTnECM[1:0]
11	501	caution added for compare an clear function at start timing
11	502	caution added for capture operation of illegal data before first counting up
13	536	name of BGCE2 bit in PRSM2 register changed to CE2 (according to device file definition)
13	537	AFO clock divider value corrected
13	538	range of PRSM2 register setting value corrected
14	546	ADMn2 addresses corrected
14	548	addresse of ADCRnmH registers corrected
15	580	bit name of SBF transmission trigger bit corrected
15	580	note added
15	609	subchapter of cautions added
16	636	flow chart of CSIB single reception combined for separate flow charts of single reception in master and slave mode
16	637	flow chart of CSIB single transmission/reception added

Chapter	Page	Description
16	638	flow chart of CSIB continuous transmission extended by query of transfer status ("CBnTSF bit = 0?")
16	639	flow chart of CSIB continuous reception replaced for separate flow charts of master and slave mode
16	640	flow chart of CSIB continuous transmission/reception added
16	642	bit name changed from BGCEn to CEn (according to device header file)
17	661	SFN3n register address corrected
17	682	bit name for chip select output level setting corrected
18	733	BPC setting corrected to 87FFH
18	733	programmable peripheral I/O area base address (PBA) corrected according to the BPC setting
18	743	EFSD bit description more precisely stated
18	743	NMI processing and DMAC operation added to caution of EFSD bit
18	743	new caution for EFSD bit added
18	783	figure of DN and MUC bit setting period changed
18	783	note added for receive data read
18	827	new caution added to "transmission abort processing"
18	840	caution of EFSD bit setting expanded
18	842	flowchart "setting CPU stand-by (from CAN sleep mode)" modified
20	867	few register names in table corrected
20	895	bit names corrected to ESO1EN, ESO1ED1, and ESO1ED0
20	929	initial value of PMCCS register in ROM-less and single-chip mode 1 corrected
20	932	initial value of PMCCT register in ROM-less and single-chip mode 1 corrected
23	952	table of conector pin functions changed to IE-V850MINI emulator
24	961	interface name corrected to UARTC0
24	963	new sub chapter inserted
24	963	PG-FP4 replaced by new flash programmer PG-FP5 throughout the chapter
25	975	μPD70F3447 added
25	976	
25	978	
25	979	max. supply current on VDD3 added when external loads are regarded
25	980	μPD70F3447 added
25	985	
26	997	208-pin plastic LQFP package removed
27	998	refrence document updated by new "Renesas Seminconductor Package Mount Manual"

Chapter	Page	Description
A	1002	initial value of PAL register corrected
A	1002	initial value of PAH register corrected
A	1002	initial value of PDL register corrected
A	1002	initial value of PDH register corrected
A	1002	initial value of PCS register corrected
A	1002	initial value of PCT register corrected
A	1002	initial value of PCM register corrected
A	1002	initial value of PCD register corrected
A	1009	r/w access of TR0CNT register in list of SFR's corrected to read-only
A	1010	r/w access of TR0SBC register in list of SFR's corrected to read-only

Index

A

A/D conversion result register n for DMA . . 551
A/D conversion result registers n0 to n9 . . 548
A/D conversion result registers n0H to n9H 548
A/D converter 30
Operation in A/D trigger mode 558
Operation in external trigger mode . . 567
Operation in timer trigger mode 561
A/D converter n mode register 0 543
A/D converter n mode register 1 544
A/D converter n mode register 2 546
A/D converter n trigger source select register 547
ADCRn0 to ADCRn9 548
ADCRn0H to ADCRn9H 548
ADDMAn 551
Address wait control register 128
ADMn0 543
ADMn1 544
ADMn2 546
ADTRSELn 547
Anytime rewrite
TMP 229
TMR 303, 305
Anytime write
TMT 447
Asynchronous Serial Interface
see UARTC
AWC 128

B

Batch rewrite
TMP 231
TMR 303, 311
TMT 449
Baud rate generator 30
CSIB 641
UARTC 602
Baud rate generator 3n 662
BCC 130
BCT0 102
BCT1 102
BEC 105
BPC 92
BRG3n 662
BSC 103
Bus clock dividing control register 131
Bus control unit 29, 98
Bus cycle configuration registers 0, 1 102
Bus cycle control register 130
Bus size configuration register 103

C

CALLT base pointer 76
CALLT execution status saving registers . . 76
CAN (Controller area network) 707
CAN Controller 707
Baud rate settings 809
Bit set/clear function 740
Configuration 710
Connection with target system 732
Control registers 742
Diagnosis functions 804
Functions 721
Initialization 778
Internal registers 733
Interrupt function 803
Message reception 782
Message transmission 790
Operation 817
Overview of functions 709
Power saving modes 798
Register access type 735
Register bit configuration 737
Special operational modes 804
Time stamp function 808
Transition from initialization mode to operation mode 780
CAN protocol 711
CANn global automatic block transmission control register (CnGMABT) 745
CANn global automatic block transmission delay register (CnGMABTD) 747
CANn global clock selection register (CnGMCS) 744
CANn global control register (CnGMCTRL) 742
CANn message configuration register m (CnMCONFm) 772
CANn message control register m (CnMCTRLm) 775
CANn message data byte register (CnMDATAxm) 769
CANn message data length register m (CnMDLCm) 771
CANn message ID register m (CnMIDLm, CnMIDHm) 774
CANn module bit rate prescaler register (CnBRP) 760
CANn module bit rate register (CnBTR) . . 761
CANn module control register (CnCTRL) . 750

CANn module error counter register (CnERC) 756
CANn module information register (CnINFO) 755
CANn module interrupt enable register (CnIE) 757
CANn module interrupt status register (CnINTS) 759
CANn module last error information register (CnLEC) 754
CANn module last in-pointer register (CnLIPT) 762
CANn module last out-pointer register (CnLOPT) 764
CANn module mask control register (CnMASKaL, CnMASKaH) 748
CANn module receive history list register (CnRGPT) 763
CANn module time stamp register (CnTS) 767
CANn module transmit history list register (CnTGPT) 765
Capture/compare control register 10 514
Capture/compare register 100 509
Capture/compare register 101 510
CBnCTL0 613
CBnCTL1 615
CBnCTL2 617
CBnRX 620
CBnRX0 612
CBnRXL 620
CBnSTR 619
CBnTX 621
CBnTX0 612
CBnTXL 621
CC100 509
CC101 510
CCR10 514
CG 208
Chip area selection control registers 0, 1 . . 100
Chip select CSI buffer register 3n 655
Clock generator 30, 208
Clocked Serial Interface
    see CSIB
Clocked serial interface clock select register 3n 651
Clocked serial interface mode registers 3n 649
CM100 508
CM101 508
CnBRP 760
CnBTR 761
CnCTRL 750
CnERC 756
CnGMABT 745
CnGMABTD 747
CnGMCS 744
CnGMCTRL 742
CnIE 757
CnINFO 755
CnINTS 759
CnLEC 754
CnLIPT 762
CnLOPT 764
CnMASKaH 748
CnMASKaL 748
CnMCONFm 772
CnMCTRLm 775
CnMDATAxm 769
CnMDLCm 771
CnMIDHm 774
CnMIDLm 774
CnRGPT 763
CnTGPT 765
CnTS 767
Compare register 100 508
Compare register 101 508
CPU 29, 68
CPU address space 82
CPU register set 70
CSC0 100
CSC1 100
CSIB
    Baud rate generator 641
    Control registers 612
    Operation 622
    Operation flow 635
    Output pins 634
CSIB (Clocked Serial Interface) 610
CSIB transmit data register (CBnTX) 621
CSIB transmit data register (CBnTX0) . . . 612
CSIBn control register 0 (CBnCTL0) 613
CSIBn control register 1 (CBnCTL1) 615
CSIBn control register 2 (CBnCTL2) 617
CSIBn receive data register (CBnRX) 620
CSIBn receive data register (CBnRX0) . . . 612
CSIBn status register (CBnSTR) 619
CSIBUF status register 3n 657
CSIC3n 651
CSIL3n 660
CSIM3n 649
CTBP 76
CTPC 76
CTPSW 76

D

Data space 85
Data wait control registers 0, 1 126
DBPC 77
DBPSW 77
Debug control unit 31
DMA Controller 145
DMA controller 29
DMA data size control register 150
DMA mode control register 149
DMA status register 150
DMA transfer
A/D converter result registers 152
Forcible termination 168
PWM timer reload 156
Serial data reception 160
Serial data transmission 164
DMA transfer count registers 0 to 7 148
DMA transfer memory start address registers 0 to 7 146
DMA transfer SFR start address registers 2, 3 147
DMA trigger factor registers 4 to 7 151
DMA wait control registers 0 and 1 96
DMAC 145
DMADSC 150
DMAMC 149
DMAS 150
DMAWC0 96
DMAWC1 96
DTCR0 to DTCR7 148
DTFR4 to DTFR7 151
DVC 131
DWC0 126
DWC1 126

E

ECR 74
ECT 78
Edge detection 180
EFG 79
EIPC 73
EIPSW 73
Endian configuration register 105
EP 204
Exception cause register 74
Exception status flag 204
Exception trap 205
Exception/debug trap status saving registers 77
External bus interface 98

F

FEPC 74
FEPSW 74
Flash memory programming mode 80
Floating point arithmetic control register . . . 78
Floating point arithmetic status register 79
Floating point arithmetic unit register set . . . 78
floating-point arithmetic unit added 19

G

General-purpose registers (r0 to r31) 71

H

HALT mode 210
release 211

I

ID 195
IMR0 to IMR6 192
In-service priority register 194
INTC 171
Internal RAM area 89
Internal ROM area 88
Interrupt control register 188
Interrupt controller 30, 171
Interrupt mask registers 0 to 6 192
Interrupt mode register 0 196
Interrupt mode register 1 197
Interrupt mode register 2 199
Interrupt mode register 3 201
Interrupt status saving registers 73
INTM0 196
INTM1 197
INTM2 199
INTM3 201
ISPR 194

M

MAR0 to MAR7 146
Maskable interrupt status flag 195
Maskable interrupts 181
Priorities 184
Restore 183
Memory controller 29
Memory map 86

N

NMI status saving registers 74
Noise removal time control register 67
Non-maskable interrupt 176
Restore 179

Non-maskable interrupt status flag 180
Non-port pins 37
Normal operating mode 80
NP 180
NRC 67

O

On-chip peripheral I/O area 90
Operating modes 80

P

PC 71
Peripheral area selection control register . . 92
PHS 95
PICn 188
Pin configuration 21
Pin functions 33, 44
Pin identification 26
Port pins 33
Ports 32
PRCMD 94
Prescaler compare registers (PRSCMn) . . 643
Prescaler compare registers 2 537
Prescaler mode register 10 517
Prescaler mode register 2 536
Prescaler mode registers (PRSMn) 642
PRM10 517
Processor command register 94
Program counter 71
Program register set 71
Program space 84
Program status word 75
Programmable peripheral I/O area 91
PRSCM2 537
PRSCMn 643
PRSM2 536
PRSMn 642
PSW 75

R

RAM 29
Random number generator 30
Real-time pulse unit 30
Receive data buffer register 3n 654
Reload
    TMP 231
    TMT 449
Reload mode
    TMR 303
ROM 29
ROM-less mode 80

S

SAR2 147
SAR3 147
Serial interface 30
SESA10 515
SFA3n 657
SFCS3n 655
SFCS3nL 655
SFDB3n 656
SFDB3nH 656
SFDB3nL 656
SFN3n 661
SFR (special function register) 999
SFR area 90
Signal edge selection register 10 515
Single-chip modes 0, 1 80
SIRB3n 654
SIRB3nH 654
SIRB3nL 654
Software exception 202
Specific registers 93
SRAM connection 135
Status register 10 519
STATUS10 519
System register set 72
System status register 95
System wait control register 96

T

Timer control register 10 512
Timer ENC10 506
Timer unit mode register 10 511
TMC10 512
TMENC10 506
TMP 225
TMP input control register 0 225
TMP input control register 1 226
TMP input control register 2 227
TMP input control register 2 (TPIC2) 442
TMPn capture/compare register 0 215
TMPn capture/compare register 1 216
TMPn control register 0 218
TMPn control register 1 219
TMPn counter register 217
TMPn I/O control register 0 221
TMPn I/O control register 1 222
TMPn I/O control register 2 223
TMPn option register 0 224
TMR1 I/O control register 1 286
TMR1 I/O control register 2 288

TMRn capture/compare register 0 273
TMRn capture/compare register 1 274
TMRn capture/compare register 2 275
TMRn capture/compare register 3 276
TMRn compare register 4 277
TMRn compare register 5 278
TMRn control register 0 281
TMRn control register 1 283
TMRn counter read register 279
TMRn dead time setting register 0 280
TMRn dead time setting register 1 280
TMRn I/O control register 0 285
TMRn I/O control register 3 289
TMRn I/O control register 4 290
TMRn option register 0 291
TMRn option register 1 293
TMRn option register 2 295
TMRn option register 3 297
TMRn option register 6 299
TMRn option register 7 300
TMRn sub-counter read register 279
TMT
Anytime write 447
Batch rewrite 449
Reload 449
TMTn capture/compare register 0 (TTnCCR0) 425
TMTn capture/compare register 1 (TTnCCR1) 426
TMTn control register 0 (TTnCTL0) 428
TMTn control register 1 (TTnCTL1) 430
TMTn control register 2 (TTnCTL2) 432
TMTn counter read buffer register (TTnCNT) 427
TMTn counter write buffer register (TTnTCW) 427
TMTn I/O control register 0 (TTnIOC0) . . . 434
TMTn I/O control register 1 (TTnIOC1) . . . 435
TMTn I/O control register 2 (TTnIOC2) . . . 436
TMTn I/O control register 3 (TTnIOC3) . . . 437
TMTn option register 0 (TTnOPT0) 439
TMTn option register 1 (TTnOPT1) 440
TMTn option register 2 (TTnOPT2) 442
TPIC0 225
TPIC1 226
TPIC2 . . . . . 227, 442
TPnCCR0 215
TPnCCR1 216
TPnCNT 217
TPnCTL0 218
TPnCTL1 219
TPnIOC0 221
TPnIOC1 222
TPnIOC2 223
TPnOPT0 224
TR1IOC1 286
TR1IOC2 288
Transfer data length select register 3n . . . 660
Transfer data number specification register 3n 661
Transmit data CSI buffer register 3n 656
TRnCCR0 273
TRnCCR1 274
TRnCCR2 275
TRnCCR3 276
TRnCCR4 277
TRnCCR5 278
TRnCNT 279
TRnCTL0 281
TRnCTL1 283
TRnDTC0 280
TRnDTC1 280
TRnIOC0 285
TRnIOC3 289
TRnIOC4 290
TRnOPT0 291
TRnOPT1 293
TRnOPT2 295
TRnOPT3 297
TRnOPT6 299
TRnOPT7 300
TRnSBC 279
TTnCCR0 425
TTnCCR1 426
TTnCNT 427
TTnCTL0 428
TTnCTL1 430
TTnCTL2 432
TTnIOC0 434
TTnIOC1 435
TTnIOC2 436
TTnIOC3 437
TTnOPT0 439
TTnOPT1 440
TTnOPT2 442
TTnTCW 427
TUM10 511

U

UARTC
Cautions 609

Dedicated baud rate generator 602
Interrupt Request Signals 587
Operation 588
UARTCn control register 0 (UCnCTL0) . . . 578
UARTCn control register 1 (UCnCTL1) . . . 603
UARTCn control register 2 (UCnCTL2) . . . 604
UARTCn option control register 0 (UCnOPT0) 580
UARTCn option control register 1 (UCnOPT1) 582
UARTCn receive data register (UCnRX) . . 585
UARTCn receive shift register 576
UARTCn status register (UCnSTR) 583
UARTCn status register 1 (UCnSTR1) . . . 584
UARTCn transmit data register (UCnTX) . . 586
UARTCn transmit shift register 577
UCnCTL0 578
UCnCTL1 603
UCnCTL2 604
UCnOPT0 580
UCnOPT1 582
UCnRX 585
UCnSTR 583
UCnSTR1 584
UCnTX 586

V

VSWC 96

V850E/PH2 User Manual

Publication Date: Rev. 4.00 Apr 30, 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

Colophon 1.0

V850E/PH2

Renesas Electronics Corporation
R01UH0439ED0400, Rev. 4.00